SS-CPS&IoT'2023

4th Summer School on Cyber-Physical Systems and Internet-of-Things

Budva, Montenegro, June 06-10, 2023

Proceedings of the 4th Summer School on Cyber-Physical Systems and Internet-of-Things

Vol. IV

Editors:

Lech Jóźwiak
Eindhoven University of Technology, The Netherlands

Radovan Stojanović
University of Montenegro, MECOnet, Montenegro

Christos Antonopoulos
University of Peloponnese, Greece



MECOnet, MANT, SMART4ALL
Montenegro, June 2023

Disclaimer

The responsibility for content in presentations and other contributions in this publication rests solely with their authors.

Official website of the event:

https://mecoconference.me/ss-cpsiot2023/

Proceedings of the 4th Summer School on Cyber-Physical Systems and Internet-of-Things, Vol. IV, June 2023

Edited by Lech Jozwiak and Radovan Stojanovic

Technical editors:

Prof. dr Budimir Lutovac, University of Montenegro

Jovan Djurkovic, MECOnet

Contributors:

Andreas Burger, Lech Jóźwiak, Sakir Sezer, Kevin Mika, Piotr Zierhoffer, Aizea Lojo, Juan Besg, Iñaki Paz, Gianluca Brilli, Paolo Burgio, Martin Matschnig, Jürgen Becker, Mario Diaz Nava, Tobias Dörr, Florian Schade, Alexander Ahlbrecht, Dominique Blouin, Anish Bhobe, Rakshit Mittal, Etienne Borde, Hector Posadas, Jose Luis Vazquez, Daniel Suarez, Naim Dahnoun, Dimitris Karadimas, Morayo Adedjouma, Luis Palacios, Réda Nouacer, Zakaria Chihani, Christoph Schmittner, Radovan Stojanović, Ramiro Samano Robles, Andrej Škraba, Christos Antonopoulos

DOI: Assigned in Zenodo and ResearchGate

Cite as:

Author/s, “Title of contribution-presentation”, in Proceedings of the 4th Summer School on Cyber-Physical Systems and Internet-of Things, Editors: Lech Jozwiak, Radovan Stojanovic and Christos Antonopoulos, Vol. IV, June 2023, pp. xx-yy, DOI: Assigned in Zenodo and ResearchGate

Copyright @ 2023 MECOnet

All rights are reserved according to the code of intellectual property acts of Montenegro and EU

Published by Consortium of MECOnet d.o.o. , MANT Association and SMART4ALL Project

For publisher: MECOnet, www.meconet.me

Filipa Lainovica 19, Podgorica, Montenegro

Message from the Chairs,

The Summer School on Cyber-Physical Systems and Internet of Things (SS-CPS&IoT'2023) is the fourth school in a series, organized in Charming Budva, a Mediterranean pearl.

In line with the practice established last year in response to the COVID-19 situation, we are pleased to continue the tradition of hosting the fourth Summer School in a hybrid format, accommodating both online and in-person participation.

SS-CPS&IoT'2023 has set forth several important objectives:

- To provide **advanced training** for industrial and academic researchers, developers, engineers, decision-makers, educators, Ph.D. and M.Sc. students, entrepreneurs, investors, research funding agents, and policymakers seeking to enhance their understanding of CPS and IoT engineering.
- To facilitate the **dissemination, exchange, and discussion** of cutting-edge knowledge and project outcomes derived from numerous European R&D initiatives in the field of CPS and IoT.
- To **promote and facilitate international connections and collaborations** among individuals working or interested in Embedded Computing, CPS, and IoT domains.

The School is open to everybody, but previous knowledge or equivalent practical experience at least at the Bachelor level in engineering (e.g. system, computer, electronic, electrical, automotive, aviation, mechanical, or industrial engineering), computer science, informatics, applied physics or similar is recommended. Industry participation is encouraged.

SS-CPS&IoT is not only to follow courses and learn new knowledge on Embedded Systems, CPS and IoT from top professionals, but to meet people, interact and discuss with outstanding researchers, developers, academic lecturers, advanced students, and other participants, collaborate or start collaborations, and meet many talented people who may become employees of your companies as well.

Distinguishing features of this advanced traditional Summer School are that its lectures, demonstrations, and practical hands-on sessions are given by top European and Worldwide specialists in particular CPS and IoT fields from industry and academia, delivering very fresh advanced knowledge. They are based on results from numerous currently running or recently finished European R&D projects in CPS and IoT, what gives an excellent opportunity to get acquainted with issues and challenges of CPS and IoT development; actual industrial problems, designs and case studies; and new concepts, advanced knowledge and modern design methods and tools created in the European R&D projects.

This year, we had the honor to invite outstanding lecturers from and outside Europe. Part of the students and lecturers came from the **H2020 project SMART4ALL**, "Self-sustained customized cyber physical system experiments for capacity building among European stakeholders", so it can be said that it was a Joint School of our community with this significant project.

SS-CPS&IoT'2023 is collocated with CPSIoT'2023, 11th International Conference on Cyber-Physical Systems and Internet-of-Things and MECO'2023, 12th Mediterranean Conference on Embedded Computing. The Summer School participants were encouraged to submit their papers to CPSIoT'2023 and MECO'2023, and thus gain additional experience of presenting work in one of the TOP conferences in computing.

The CPS&IoT'2023 Summer School Program was composed of four days of lectures, demonstrations, practical hands-on sessions, and discussions, as well as free participation in MECO'2023 and CPSIoT'2023 sessions. The topics of the lectures, demonstrations, and practical hands-on sessions cover major CPS fields and applications. We had about 60 lecturers and students, coming from over 20 countries, WorldWide. We worked for four days in a 32-hour capacity, that is equivalent to an academic workload of **3 ECTS credits**. Detailed list of the presentations including the names of their authors and presenters is provided in this Proceedings. This Proceedings, in addition to its research and educational component, serves as a supplement to the diplomas awarded to School's participants, after testing their activities and knowledge.

SS-CPS&IoT'2023 taken place in Hotel Budva, Montenegro. Budva is a 3500 years old town located at the Adriatic Sea coast of Montenegro. It is a popular touristic destination, with its charming Old Town, beautiful natural environment, 35 clean sandy beaches, and proximity to many famous touristic attractions as Kotor, Boka Kotorska, Sveti Stefan, Dubrovnik, and several national parks. It is an excellent place to have a summer school in a relaxed and friendly atmosphere.

The Chairs of the SS-CPS&IoT'2023 express their thanks to all authors and presenters as well as, to all other people who contributed to the success of the Summer School. We are especially proud on 3rd generation of students who successfully finished School and showed an enviable level of knowledge and interest.

We are very grateful to Professor **Budimir Lutovac**, Publication Chair of CPSIoT'2023 and MECO'2023 helping us to compose these Proceedings, which represents only part of the results carried out by SS-CPSIoT'2023.

We hope to see you again next year, mostly, in-venue, in good health and friendly atmosphere.

Yours,

Lech Jóźwiak
Eindhoven University of Technology, The Netherlands

Radovan Stojanović
University of Montenegro, Montenegro

Christos Antonopoulos
University of Peloponnese, Coordinator of SMART4ALL Project, Greece

Contents

Disclaimer

Lech Jóźwiak, Radovan Stojanović
Introduction to the CPS&IoT'2023 Summer School 1
Lech Jóźwiak
Green CPS and IoT for Green World 4
Sakir Sezer
DPU Technology: Redefining next generation Datacenter Security 106
Kevin Mika, Piotr Zierhoffer
VEDLIoT – Accelerated AIoT 134
Aizea Lojo, Juan Besga, Iñaki Paz, Gianluca Brilli, Paolo Burgio, Martin Matschnig
FRACTAL: A Cognitive Computing Platform Node for the Edge 155
Jürgen Becker
Embedded reliable HPC enabling Automotive & IoT Intelligence 232
Mario Diaz Nava
ANDANTE: AI for New devices and Technologies at the Edge 233
Tobias Dörr, Florian Schade, Alexander Ahlbrecht
Simulation-based development of networked avionics systems using the XANDAR toolchain 279
Dominique Blouin, Anish Bhobe, Rakshit Mittal, Etienne Borde
Embedded Systems Modeling, Analysis and Automatic Code Generation with AADL and RAMSES (hands-on) 316
Hector Posadas, Jose Luis Vazquez, Daniel Suarez
Offloading and Reconfiguration of Functional Components in Embedded Hardware Resources: Application to AI Components 368
Dimitris Karadimas
XANDAR: CI/CD automations for complex pipelines with hardware in the loop 422
Morayo Adedjouma, Luis Palacios, Réda Nouacer
From Knowledge-based to Model-based Engineering of embedded intelligence and trust in CPS 513
Zakaria Chihani, Réda Nouacer
Towards Characterizing AI Trustworthiness 541
Radovan Stojanović
Design of performance and energy efficient nodes for smart systems 731
Ramiro Samano Robles
Evolution and evaluation of wireless technologies and standards towards the convergence of 5G/6G and IoT 767
Andrej Škraba
Overview of Several CPS&IoT Prototypes: Wheelchair, Group Heart Rate Monitoring, PIDDCMotorControl 862
Christos Antonopoulos
SMART4ALL project: Customized Low-Energy Computing for CPS 906

Schedule-CPS&IoT'2023 Summer School on Cyber-Physical Systems and Internet-of-Things..... *958*

4th Summer School on Cyber Physical Systems and Internet of Things - SS-CPSIoT'2023 4th Generation (Students and Teachers)..... *959*

Certificate of Attendance..... *960*

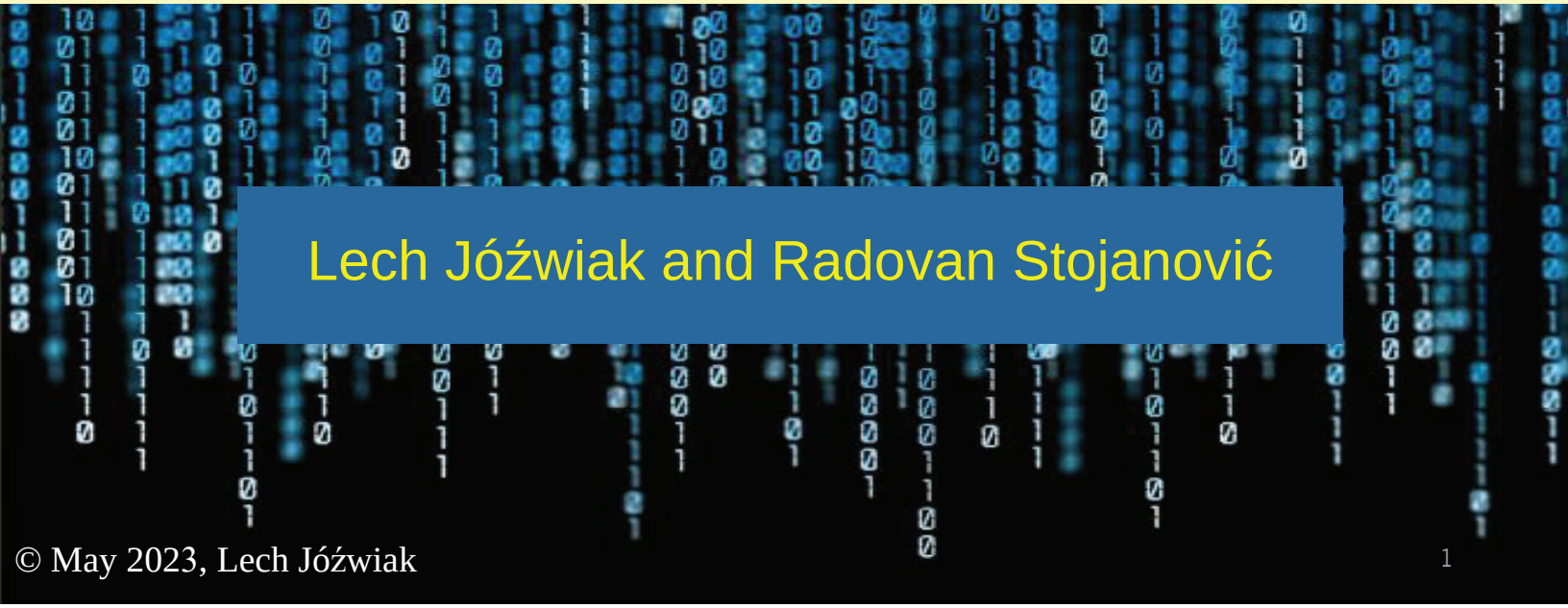
Author Index..... *961*

Photo gallery..... *962*

CPS&IoT'2023 Summer School

Budva, Montenegro
June 6-10, 2023

Introduction



Lech Jóźwiak and Radovan Stojanović

© May 2023, Lech Jóźwiak

Introduction

- Systemic drawbacks of the traditional economy and cumulation of bad decisions driven by the short-term profit and made without adequately accounting for long-term consequences resulted in the **huge global environmental disaster**
- Innovations exploiting modern CPS and IoT technologies have a high potential to significantly improve systems used by us or that we are part of
- To recover from the environmental disaster and further develop:
  - *a model of a well regulated and controlled effective and efficient system should be applied to all kinds of systems, collaboration chains and related flows*
  - *modern CPS and IoT technologies should be used to much better control and optimize the social, physical and life systems than till now*
  - *methodologies of circular regenerative economy and quality-driven design should be used to design the systems*
- In this CPS&IoT Summer School you will have a unique occasion to be informed on and to discuss the most recent European R&D developments in CPS and IoT

2

Outline of the CPS&IoT'2023 Summer School

1. Introduction to CPS&IoT 2023 Summer School
2. Green CPS and IoT for Green World
3. Computing and communication technologies for CPS and IoT
4. Machine Learning and Edge Computing
5. Modeling, design and implementation of CPS and IoT
6. Trustworthy CPS and IoT: reliability, security and safety
7. Energy-efficient computing for CPS and IoT
8. Closing of the CPS&IoT2023 Summer School

CPS&IoT

CPS&IoT'2023 Summer School
Budva, Montenegro, June 6-10, 2023

SMART4ALL

Green CPS and IoT for Green World



Lech Jóźwiak

L.Jozwiak@tue.nl

© May 2023, Lech Jóźwiak

1

Outline

1. Introduction
2. Modern cyber-physical systems (CPS)
3. Importance of modern CPS and IoT
4. Challenges of advanced CPS development
5. Computing technology for advanced CPS
6. Environmental crisis and environmental footprint of CPS and IoT
7. Importance of advanced green CPS and IoT for environmental recovery
8. IoT for advanced green CPS
9. Quality-driven design of advanced green CPS
10. Conclusion

Introduction: Aims of this tutorial

- **The two main aims of this tutorial are the following:**
  - ***to make the participants aware of the necessity of green CPS and IoT***
  - ***to prepare the ground for the whole CPS&IoT'2021 Summer School***
- This means in particular:
  - to introduce several basic definitions related to CPS
  - to explain the necessity of green CPS and IoT
  - to sketch the CPS scene, what includes:
    - introduction to modern CPS and IoT, their importance, their ongoing revolution, and challenges of their development, and
    - explanation of the necessity of their holistic multi-objective quality-driven design
  - to introduce the methodology of quality-driven green system design

Introduction: Further reading for this tutorial

- L. Jóźwiak: Advanced Mobile and Wearable Systems, Microprocessors and Microsystems, Elsevier, Vol. 50, May 2017, pp. 202–221
- L. Jóźwiak: Quality-driven Design in the System-on-a-Chip Era: Why and how?, Journal of Systems Architecture, vol. 47, no. 3-4, Apr. 2001, pp. 201-224
- L. Jóźwiak: Life-inspired Systems and Their Quality-driven Design, Lecture Notes in Computer Science, Vol. 3894, 2006, Springer, pp. 1-16
- Jóźwiak, L.; Lindwer, M.; Corvino, R.; Meloni, P.; Micconi, L.; Madsen, J.; Diken, E.; Gangadharan, D.; Jordans, R.; Pomata, S.; Pop, P.; Tuveri, G.; Raffo, L. and Notarangelo, G.: ASAM: Automatic Architecture Synthesis and Application Mapping, Microprocessors and Microsystems journal, Vol.37, No 8, pp. 1002-1019, 2013
- Jóźwiak, L. and Jan, Y.: Design of Massively Parallel Hardware Multi-Processors for Highly-Demanding Embedded Applications. Microprocessors and Microsystems, Volume 37, Issue 8, November 2013, pp. 1155–1172.
- L. Jóźwiak and S.-A. Ong: Quality-driven Model-based Architecture Synthesis for Real-time Embedded SoCs, Journal of Systems Architecture, Elsevier Science, Amsterdam, The Netherlands, ISSN 1383-7621, Vol. 54, No 3-4, March-April 2008, pp. 349-368
- Many other papers of myself and my former Ph.D. students; many of them referenced in the above papers

Introduction: What is a system?

- A **system** is a ***complex whole composed of interrelated, interdependent and/or interacting items*** (parts or elements of a system) ***that are so intimately connected that they appear and operate as a single unit in relation to the external world*** (to other systems)
- **Three basic types of systems:**
  - ***unorganized system*** **-** a mechanical unsystematic conglomerate of objects
  - ***organized system*** **-** a systematic, relatively stable and law-governed composition of parts which properties cannot be reduced to the simple sum of the properties of its parts, but involve some new emerging properties resulting from complex composition of the parts' properties (e. g. a molecule, crystal, circuit, computer, machine), and
  - ***organic system*** **-** formed not as a composition of some ready-made parts, but being an ***integral whole*** with distinguishable parts that originate, develop and die together with the whole, and cannot preserve and demonstrate their complete quality without the whole (e. g. life organisms); the characteristic features of the organic systems are the self-development and self-reproduction
- In this presentation **organized systems** will be considered

5

Introduction: System organization and structure

- The **system organization** (composition) appropriately:
  - defines its parts
  - arranges the parts in relation to each other and to the whole, and
  - interconnects them to form the whole
- The term **system structure** designates the ***parts of a system arranged into a proper relation and appropriately interconnected*** according to a certain set of laws and/or rules in order to form a whole
- We will consider material systems
- **Since matter is active** and is in constant change, **the material systems are in constant change**, with only some relative and transient stability conditions
- Compositions of interrelated, interdependent or interacting single changes (transformations, actions) form **processes**
- **Process** is a relatively ***isolated composition of interrelated interdependent or interacting actions*** (transformations, changes)

6

Introduction: System = process © structure

- A given process can only perform (take place, occur) in particular relatively stabile conditions
- These conditions that make the process possible are created and guaranteed by the system **structure**
- The **system structure** is a relatively isolated, stable and slowly changing (in relation to the process) part of the universe in which a particular process (or a collection of co-operating processes) can take place
- A **system** is a ***unity of a process and structure*** in which this process takes place
- **System design** is an activity of ***defining an appropriate composition of the system process and structure***

7

Introduction: What are cyber-physical systems?

- **Cyber** comes from Greek adjective ***kyberneticos*** (***cybernetic***) that means skilled in steering or governing
- Already in ancient times people constructed various systems: the oldest known artificial automatically controlled system is probably a water clock invented by Ktesibios (285–222 BC) in Alexandria
- Form those times, the construction of machines (physical systems) and their controllers (cyber systems) continued and developed through the centuries
- Until the end of 19th century the controllers (cyber systems) were implemented as mechanical, hydraulic and pneumatic systems
- In the 20th century they started to be gradually replaced by the electric controllers, and later by the electronic controllers
- **Physical systems** are systems in which matter or energy acquisition, processing and transfer take place according to the lows of physics
- **Cyber systems** are ***(parts of) control systems***, i. e. information collecting, processing and communicating systems

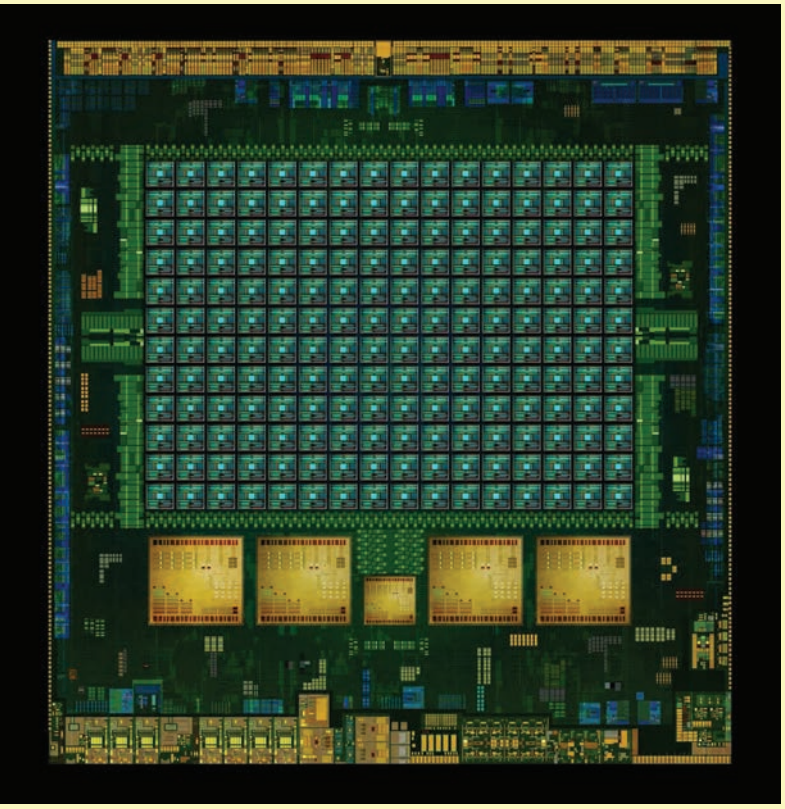
Introduction : What are cyber-physical systems?

- **Cyber-physical system** (**CPS**) is a compound system engineered through integration of cyber and physical sub-systems or components and/or pre-existing component cyber-physical systems, so that it appears and operates as a single unit in relation to the external world (to other systems)

- Introduction of the transistor and integrated circuit technologies in the years 1950s and 1960s, correspondingly, enabled the ***ongoing microelectronics and information technology revolution*** that is till now progressing according to the Moore's low

- The recent revolutionary progress in computing platforms, communication, networking, sensors and actuators enables:
  - much more effective and efficient CPS for traditional applications, and
  - "smart", sophisticated and affordable CPS for numerous new applications, e.g. smart robots, homes, cars, wearable and implantable medical devices, etc.

Introduction: very complex MPSoCs



Source: ANANDTECH
(http://www.anandtech.com/show/7622/nvidia-tegra-k1)

- *Modern nano-dimension semiconductor technology enables implementation of a* ***very complex multiprocessor system on a single chip (MPSoC)***
- **This facilitates a rapid progress in:**
  - ***global networking***
  - ***(mobile) wire-less communication***
  - ***(mobile autonomous) embedded computing***

NVIDIA Tegra K1 massively parallel MPSoC for mobile applications

CPU: (4+1) Cortex-A15 cores

Kepler GPU: 192 CUDA GPU cores

10

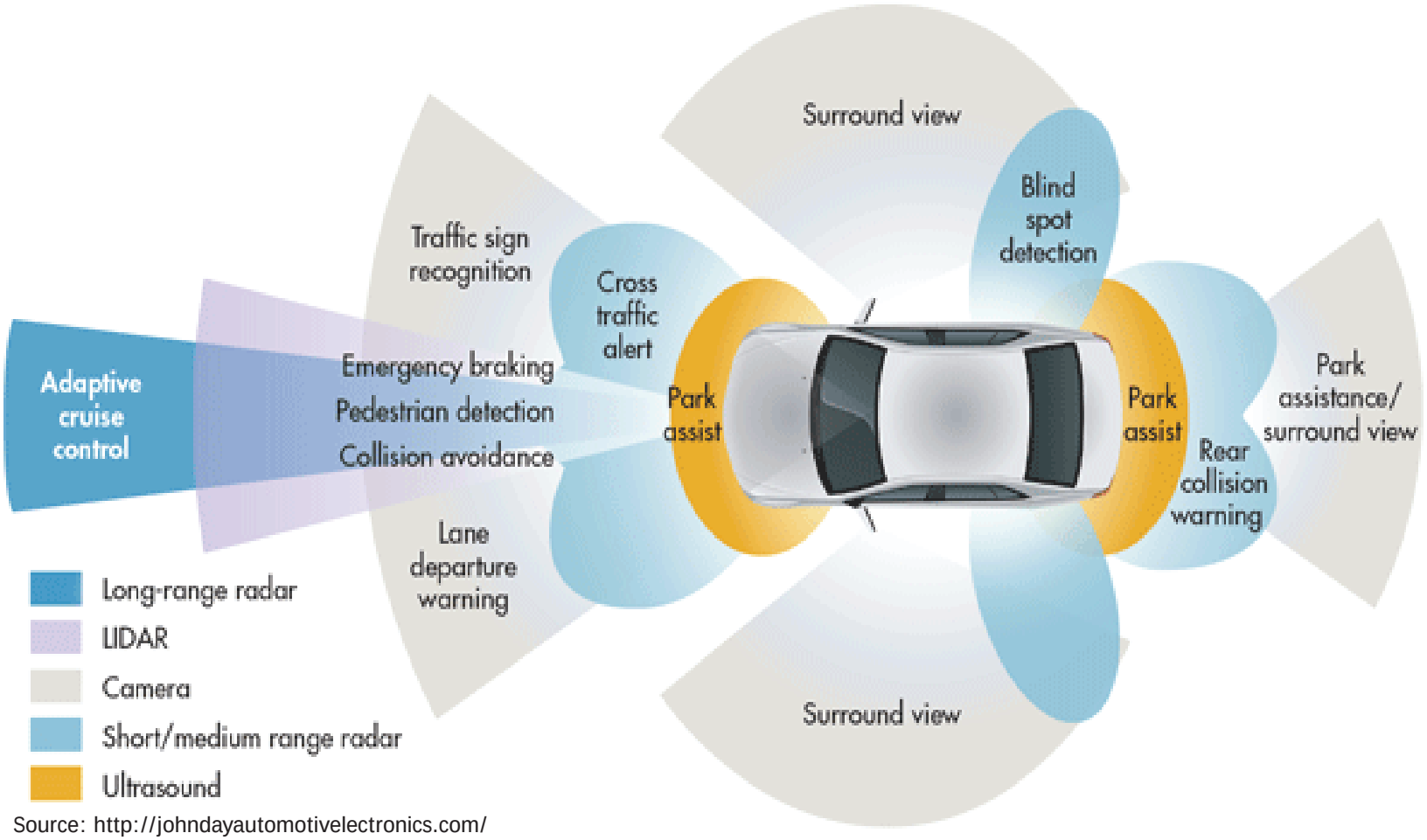
Introduction: cyber-physical technology revolution

- **The recent rapid developments in:**
  - system-on-a-chip technology
  - common global networking
  - wire-less communication
  - mobile and autonomous computing
  - miniaturized sensors and actuators
  - material technology

  created a **large discrepancy between what is possible and what is used nowadays**
- This discrepancy:
  - causes both a **very strong technology push** and **market pull** to create new or modified products and services, and
  - results in the ***cyber-physical technology revolution***
- Recently, a revolutionary transition has been started from the **internet of computers** to the **internet of smart (mobile) cyber-physical systems (CPS)**, called **Internet of Things (IoT)**

11

Examples of modern CPS: autonomously-driving cars



Source: http://johndayautomotivelectronics.com/

Examples of modern CPS: smart wearables













Examples of CPS: wearable virtual and augmented reality





Source: https://www.oculus.com



Source: http://www.technodo.com/

14

Examples of modern CPS: smart miniaturized implants and pill-size medical devices





modern 10 times smaller pace-makers

A new wave of the information technology revolution has arrived that creates much more coherent and fit to use CPS and connects them to form the IoT

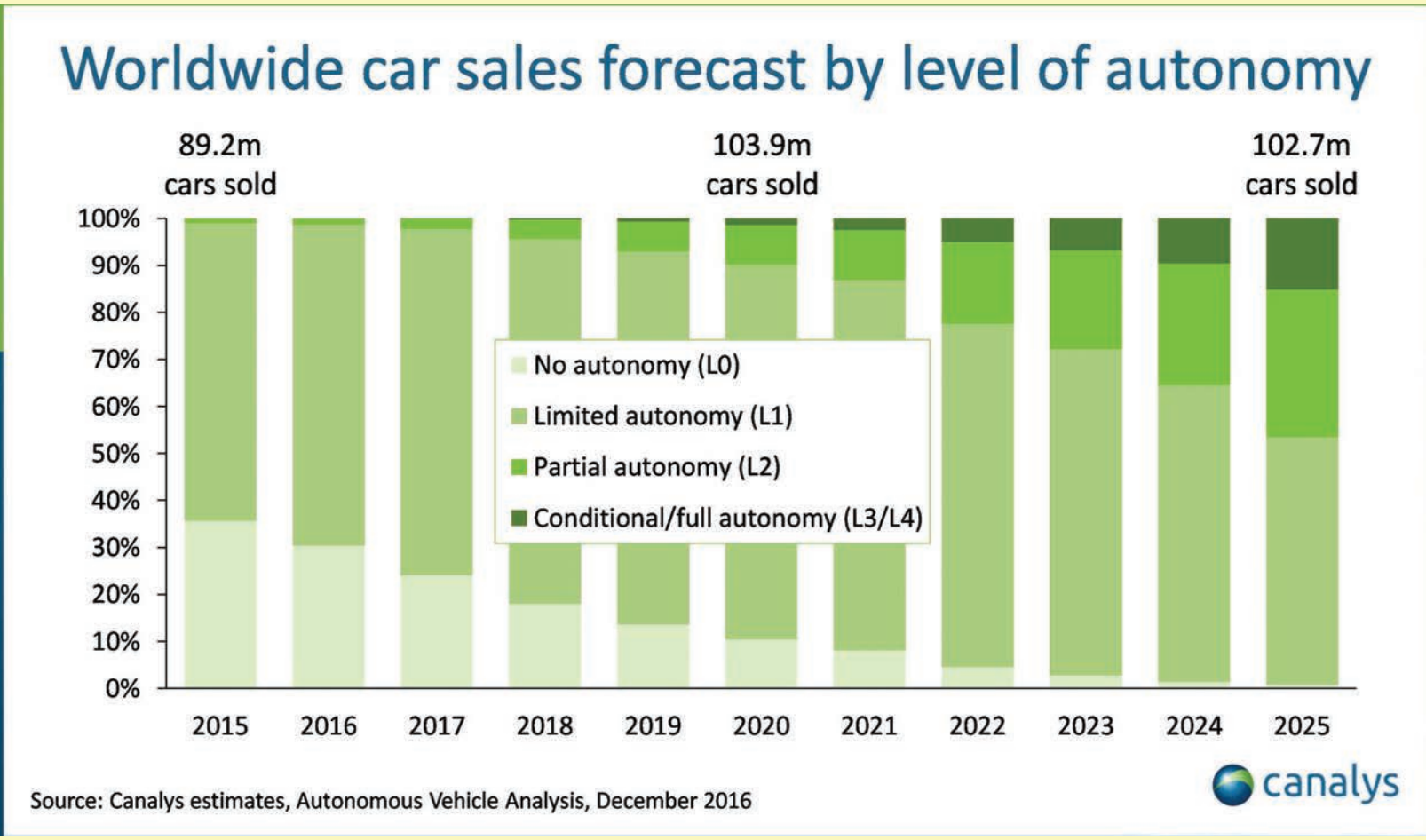
15

Importance of modern CPS

- **Application areas of mobile CPS** cover ***virtually all socially important application sectors***, including:
  - *consummer applications* , e.g. mobile computing, communication, localization, navigation, gaming, entertainment, fashion, etc.
  - *extension or replacement of human capabilities*, e.g. tele-operation, personal assistance, artificial limbs, implants, etc.
  - *social systems*, e.g. smart health-care and other numerous health-care applications, assisted leaving, law enforcement, public safety, military, etc.
  - *transportation and automotive*, e.g. traffic control, navigation, tracking, communication, mobile fares and personalized customer service, assisted/autonomous driving, etc.
  - *industrial, safety, security and military applications* , e.g. mobile real-time in-the-field surveillance, monitoring, inspection, repair, robotics, instruction, assistance, etc.
  - *commercial applications*, e.g. mobile inventory tracking and customer service, wearable augmented reality and other systems for touristic applications, and many others
- **The economic and societal importance of mobile CPS is very high and rapidly increases**

16

Rapid growth of the modern mobile CPS and IoT markets



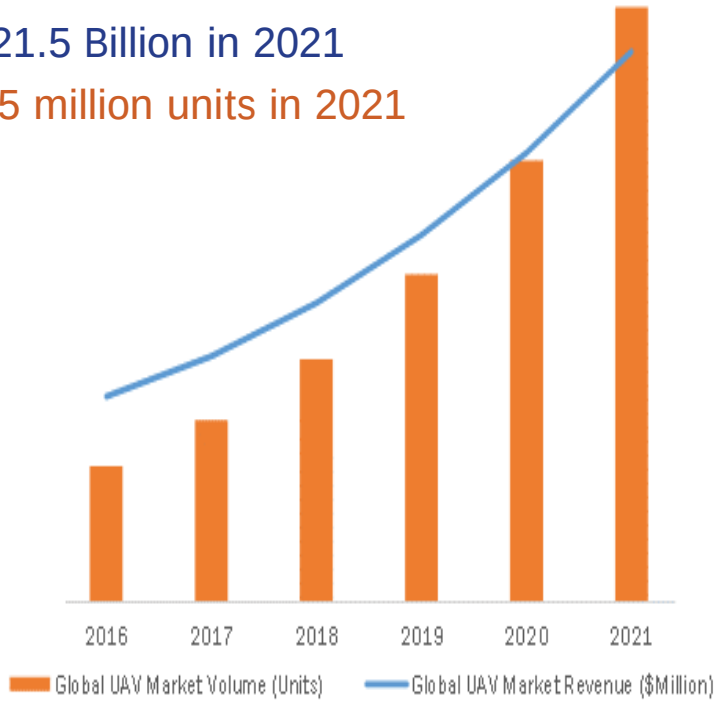
Rapid growth of the modern mobile CPS and IoT markets

Global unmanned aerial vehicle (UAV) market

$8 billion in 2016

$21.5 Billion in 2021

>5 million units in 2021



Source: BIS Research, January 2018

- **The fastest growing market** of all mobile sectors is this **of smart wearable devices**:
  - $14 billion and 123 million devices in 2016
  - $34 billion and 411 million devices in 2020

  (CCS Insight, February 2016)

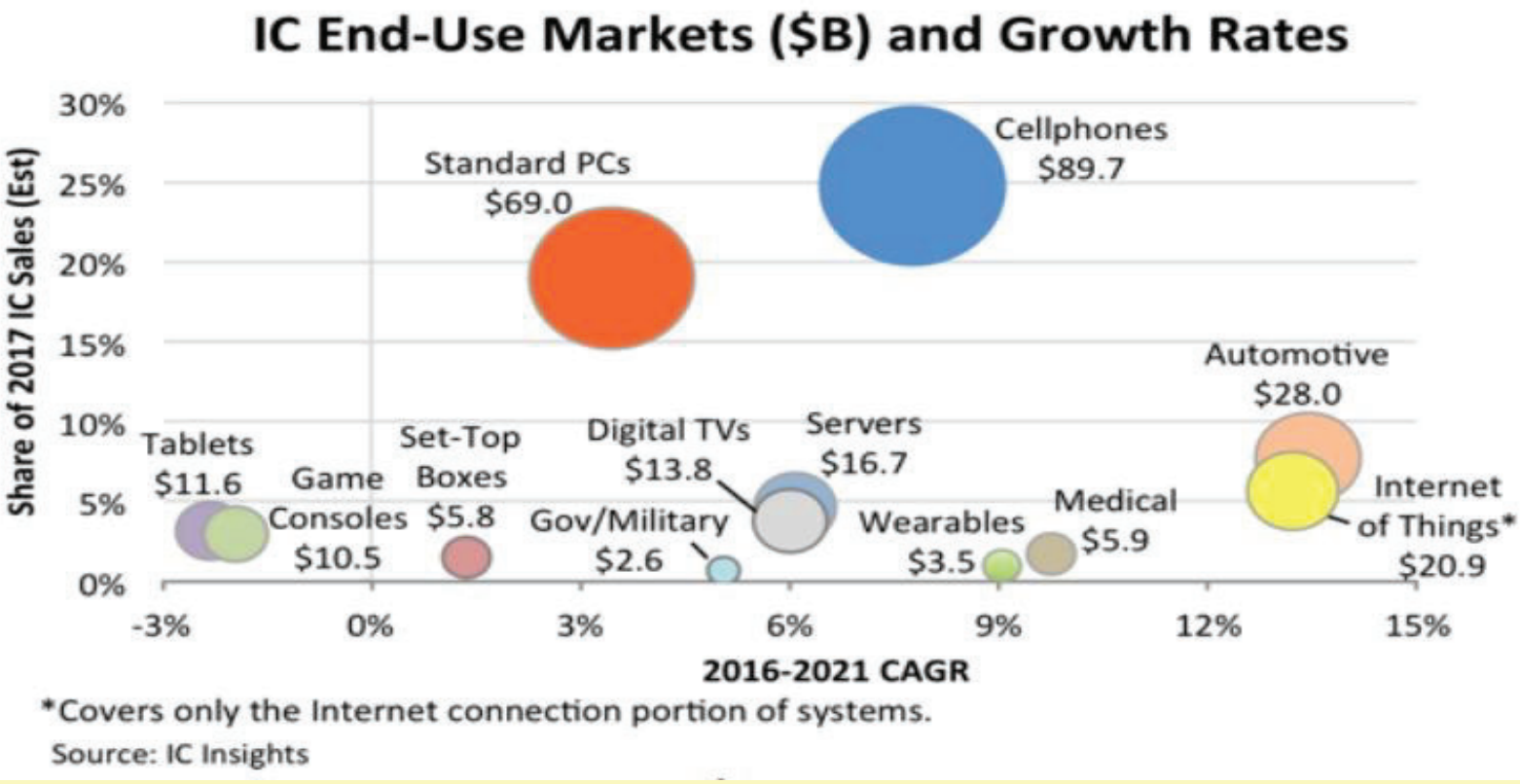
18

Rapid growth of the CPS and IoT markets

- The number of connected IoT devices was 11.3 billion in 2020
- IoT Analytics forecasts that there will be 27.1 billion connected IoT devices by 2025, and Cisco 14.6 billion machine-to-machine IoT connections by 2022.
- Allied Market Research finds that the global IoT industry generated sales of $740.5 billion in 2020, and is estimated to reach $4,421.6 billion by 2030
- This corresponds to the growth rate at a CAGR of 19.6% between 2021 and 2030
- In 2020 the strongest contributor to the global IoT market was the industrial manufacturing segment, accounting for more than 25% of the market, and it is expected to maintain its leadership during the forecast period
- However, the healthcare segment is expected to grow at the highest CAGR of 26.2% between 2021 and 2030
- The ESD Alliance reported that the electronic (embedded) system design (ESD) industry had the total market revenue of $13.2 billion in 2021, with the revenue grow rate of 15.8% comparing to 2020, with the fastest growing computer-aided engineering, printed circuit board and multi-chip module, semiconductor intellectual property, and services

19

Rapid growth of the **chip market** for CPS and IoT

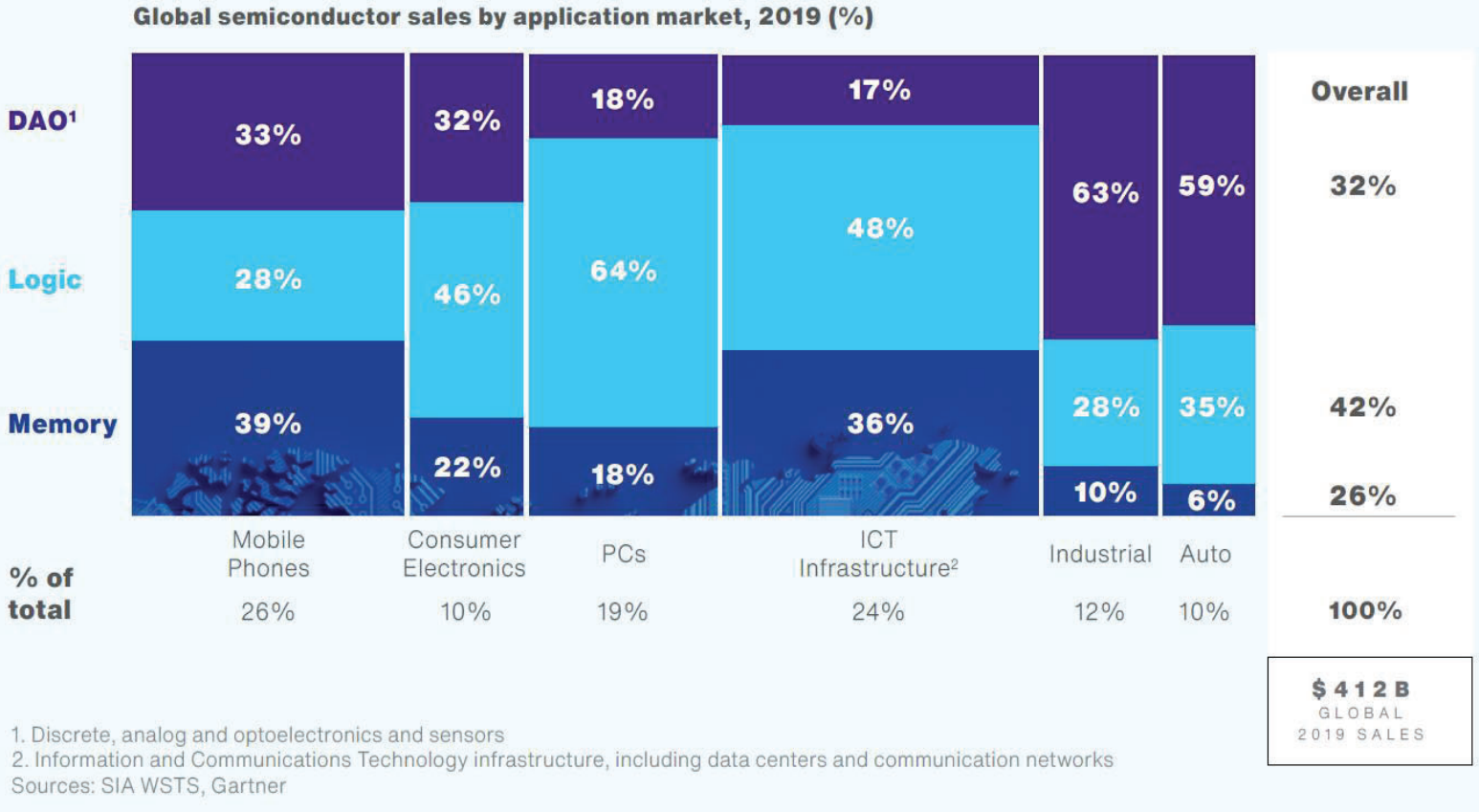


Source: IC Insights

- The fastest-growing chip markets are automotive, IoT, medical and wearables

20

Semiconductor market related to CPS and IoT in 2019



Source: SIA WSTS and Gartner

- PCs account for only 19%, while a large majority of the rest is related to CPS and IoT

21

Semiconductor market related to CPS/IoT in 2021/2022

- According to Semiconductor Industry Association (SIA), the global semiconductor industry sales in 2021 increased by 26.2% compared to the 2020 to the highest-ever annual value of $556 billion
- A record number of 1.15 trillion semiconductor units were shiped in 2021
- Sales in categories related to CPS and IoT were as follows:
  - micro-ICs (including microprocessors) increased 15.1% to $80.2 billion
  - logic increased by 30.8% to $154.8 billion
  - memories incresed 30.9% to $153.8 billion
  - analog semiconductors (commonly used in vehicles, consumer goods, and computers) increased 33.1% to $74 billion
  - automotive ICs increased 34.3% to a record high of $26.4 billion
- According to World Semiconductor Trade Statistics (WSTS), in 2022 the global semiconductor market is expected to increase by 10.4%, which corresponds to annual sales of US$ 613.5 billion
- The growth will be mainly driven by sensors 17.2%, logic with 17.1% and analog with 14.1%

22

Challenges: unusual complexity and ultra-high demands

- The huge and rapidly developing markets of sophisticated mobile CPS represent **great opportunities**
- These opportunities come with a price of:
  - **unusual system complexity** and **heterogeneity**, resulting from ***convergence and combination of various applications and technologies*** in one system or even on one chip, and
  - **stringent and difficult to satisfy requirements** of modern applications
- **Smart cars, drones and various wearable systems**:
  - involve **big instant data** from multiple complex sensors (e.g. camera, radar, lidar, ultrasonic, sensor network tissues, etc.) and from other systems, used for mobile vision, imaging, virtual or augmented reality, etc.
  - are required to provide **continuous autonomous service in a long time**
  - are **safety-critical**
- In consequence, they demand a **guaranteed (ultra-)high performance** and/or **(ultra-)low energy consumption**, while requiring a **high reliability, safety and security**

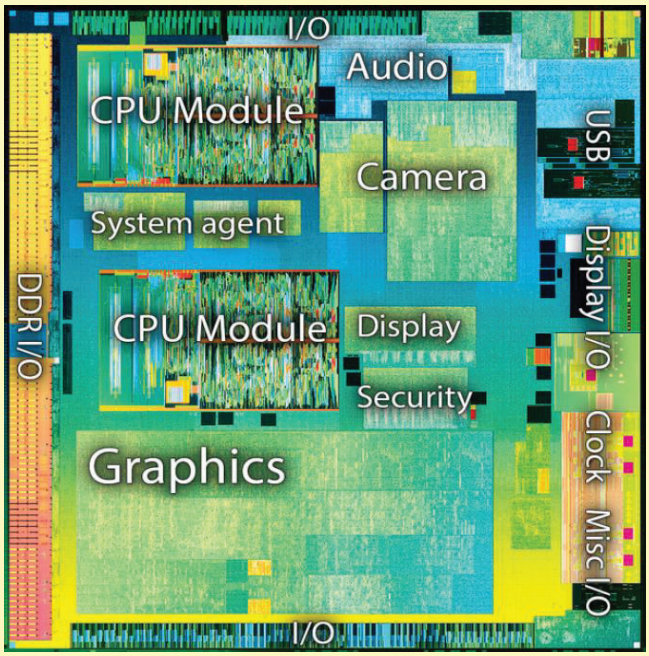
23

Challenges: application parallelism and heterogeneity

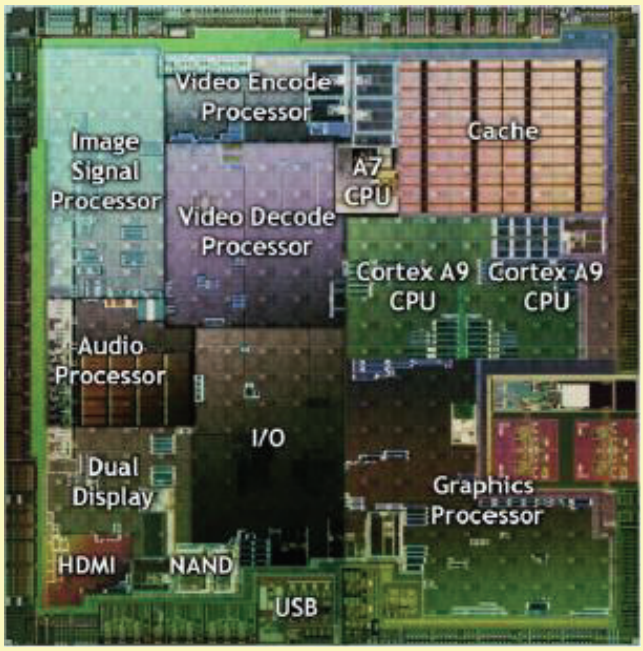
- The modern complex applications that require ultra-high performance and/or ultra-low energy consumption:
  - are from their very nature **heterogeneous**
  - include numerous different algorithms involving **various kinds of massive parallelism**: data parallelism, and task-level, instruction-level and operation-level functional parallelism
- To adequately serve these applications:
  - **heterogeneous computation platforms** have to be exploited
  - processing engines with **parallel multi-processor macro-architectures** and **parallel processor micro-architectures** have to be constructed
  - different parts of complex applications involving different kinds of parallelism have to be implemented with corresponding different application-part specific parallel hardware
  - multiple different or identical processors, each operating on a (partly) different data sub-set, have to work concurrently to realize the ultra-high throughput and ultra-low energy consumption

Challenges: application complexity, parallelism and heterogeneity

To implement the highly-demanding complex heterogeneous CPS applications ***complex heterogeneous MPSoCs*** *are needed*



Intel Atom Z3770*

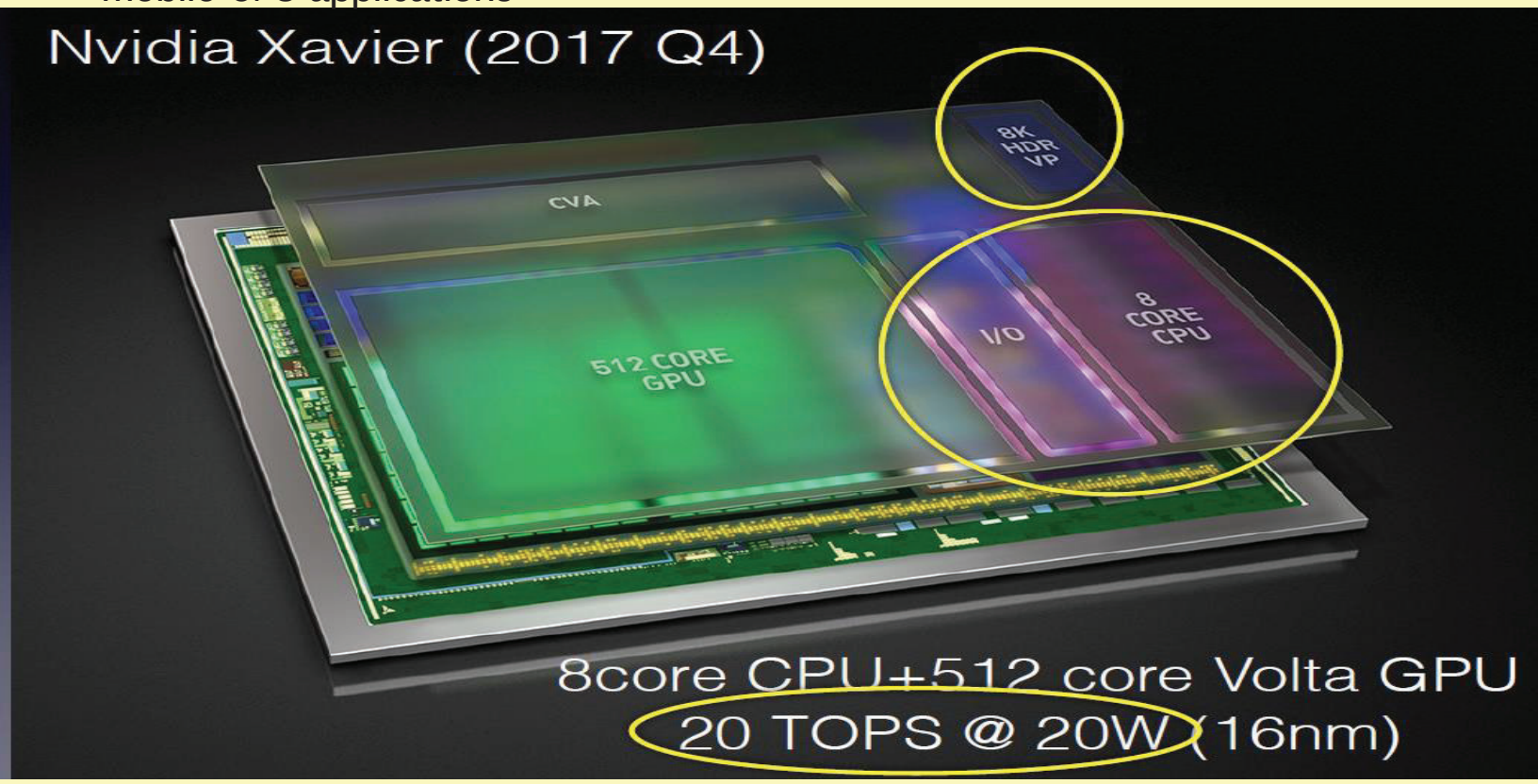


Nvidia Tegra 2+

*Source: http://tweakers.net/reviews/3162/2/intels-atom-bay-trail-de-eerstenieuwe-atom-in-vijf-jaar-zes-verschillende-bay-trails.html
+Source: http://www.anandtech.com/show/4144/lg-optimus-2x-nvidia-tegra-2-reviewthe-first-dual-core-smartphone/3

Challenges: application complexity, parallelism and heterogeneity

NVIDIA's advanced massively parallel heterogeneous MPSoC for ADAS and similar mobile CPS applications



Source: Albert Y.C. Chen, Viscovery

26

Challenges: criticality of applications

- Cyber-physical systems influence our life to a higher and higher degree
- Therefore, the society expectations regarding them grow rapidly
- Due to CPS common usage in various kinds of technical, social and biological applications, and their growing influence, **we and the life on the Earth more and more depend and rely on these systems**:
  - their ***quality*** is becoming ***more and more critical***
  - many ***applications considered previously as non-critical are becoming critical***
- Due to the rapidly growing share of the highly demanding embedded and CPS applications, ***higher demands are becoming much more common***
- Due to the multiple reasons just discussed, and specifically, due to the rapidly growing system and silicon complexity and diversity, it will be ***more and more difficult to guarantee the systems' quality***
- This is a **new difficult situation** that cannot be adequately addressed without an **adequate design methodology** and **electronic design automation**

27

Quality-driven Model-based Design

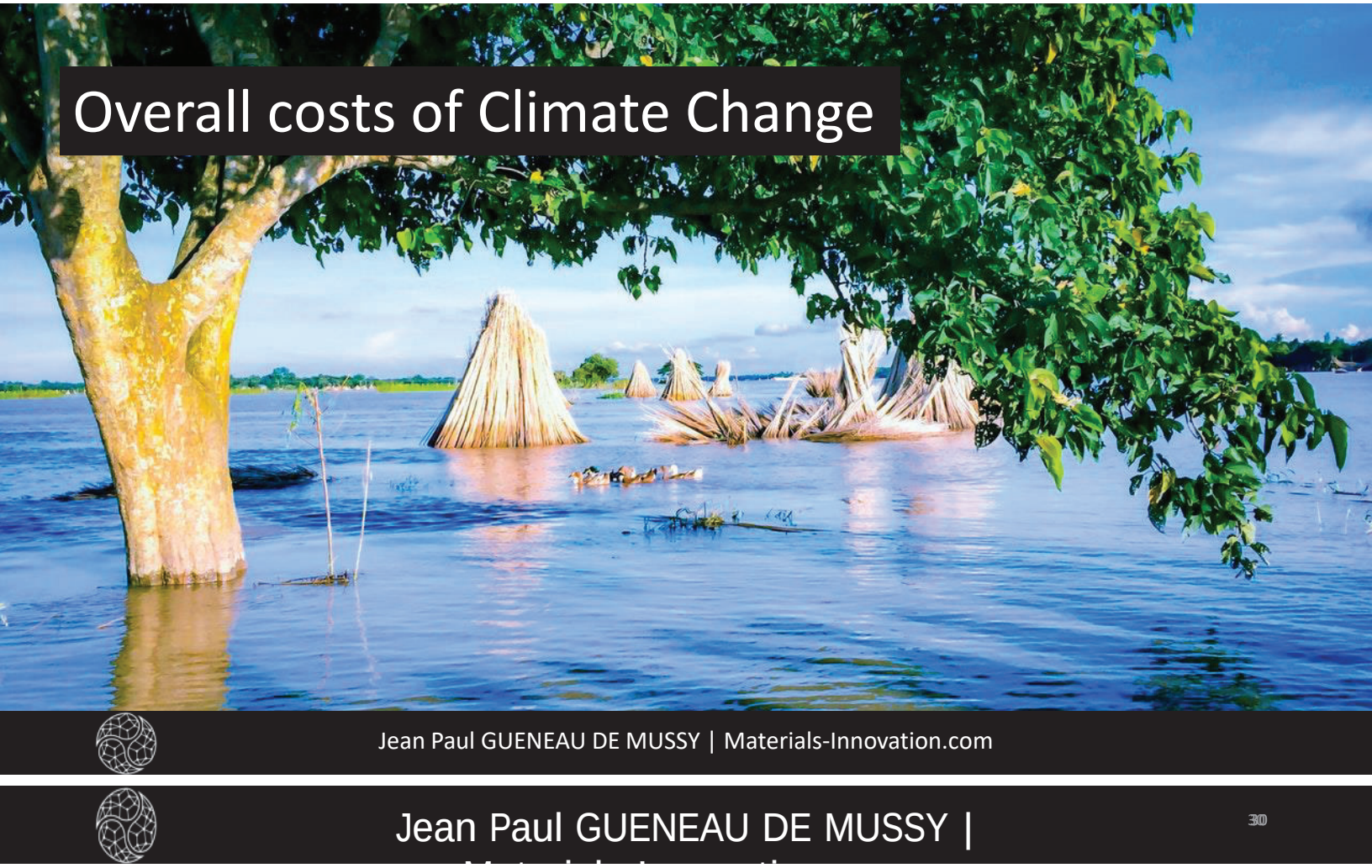
- When considering a **system and design methodology adaptation** to the situation in the field of modern CPS, we have first to ask: *what general system approach and design approach seem to be adequate to solve the listed problems and overcome the challenges*?
- **Predicting the current situation,** more than 20 years ago I proposed such **system paradigm** and **design paradigm**, i.e. the paradigms of:
  - **life-inspired systems** and **quality-driven design**, and
  - the **methodology of quality-driven model-based system design** based on them
- From that time my research team and our industrial and academic collaborators were researching the **application of this methodology** to the design and design automation of embedded processors, MPSoCs and CPS, and this **research confirmed the adequacy of the quality-driven design methodology**
- For "Outstanding Achievements and Contributions to Quality of Electronic Design" I was awarded the Honorary Fellow Award by the International Society for Quality Electronic Design (San Jose, CA, USA, 2008)

28

Quality-driven Design, CPS and IoT for making high-quality systems

- When using the quality-driven design methodology to develop the modern high-quality collaborating cyber-physical systems, in which the sophisticated cyber systems (controllers) are tightly integrated with the controlled by them physical, social and life systems, we have a great chance to much better control and optimize the social, physical and life systems than we did it till now
- ***With modern CPS and IoT technology we have a great chance to significantly improve most systems used by us or that we are part of***
- **We also have no chance to not do this**
- ***Our social, physical and life systems have to be significantly and immediately improved***
- **Why?**
- Please watch the following few slides that I got from my friend Jean Paul Gueneau de Mussy, Sustainability and Innovation Expert, CEO of Materials and Systems Innovation Company, https://materials-innovation.com/

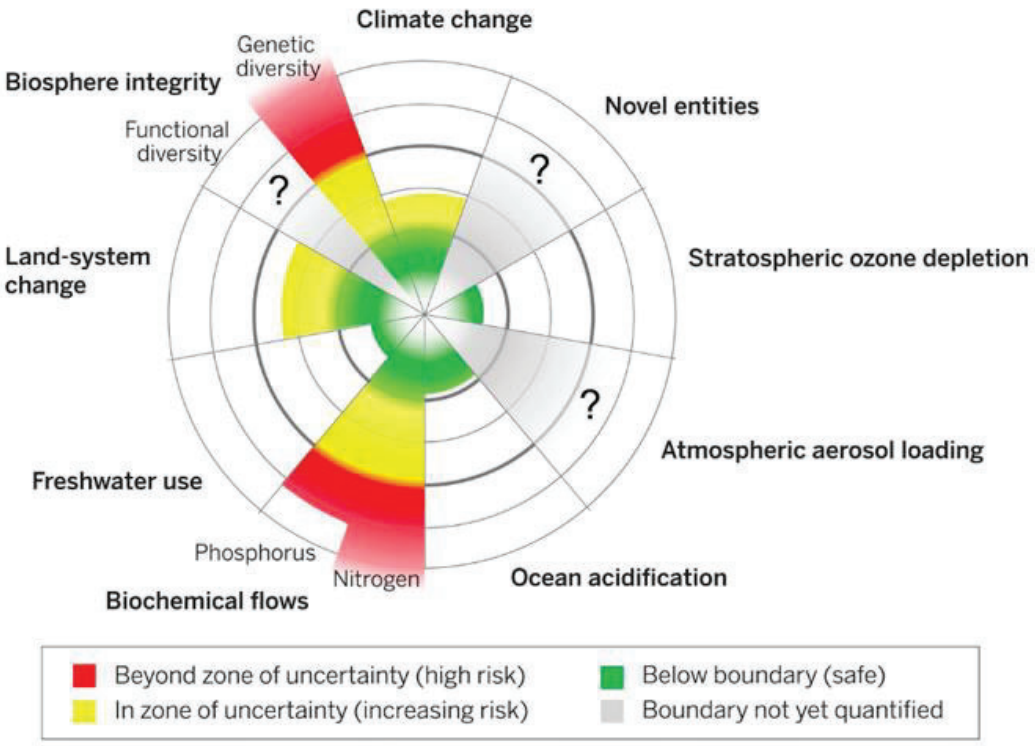
29


Overall costs of Climate Change
Jean Paul GUENEAU DE MUSSY | Materials-Innovation.com
Jean Paul GUENEAU DE MUSSY |
30


Biodiversity loss


Massive use of Resources

Planetary Boundaries



Johan Rockström et all, February 2017, Volume 46, Issue 1, pp 4–17

Huge destruction, chaos, no care for long-term consequences

- These were only a few examples of what was done wrong for a long time with our economic, social, technical and life systems on a global scale, and what resulted in a **huge destruction on a global scale**
- This huge destruction is a result of systemic drawbacks of the traditional economy and very many bad decisions made by numerous governments and companies for a short-term profit only, without accounting for long-term consequences
- Example: the wild chaotic globalization, without carefully designed interfaces and collaboration between very different economic/political systems in different parts of the World and between companies from the very different systems
- Globalization is unavoidable, but the actual costs of the wild globalization were not pay by those who profited, but by the poverty of others and destruction of the World
- The not well regulated and controlled inefficient collaboration chains and related material, product and waste flows of the wild globalization resulted in inefficient use of resources, environment destruction and pollution, climate change, bio-diversity loss, etc.

33

Huge destruction, chaos, no care for long-term consequences

- Covid-19 pandemics demonstrated the problems sharply
- Example: Due to globalization multiple supply chains became very complicated and very long, often crossing borders of several countries; due to Covid-19 pandemics, protectionism, etc. many chains were broken or function inefficiently
- For instance, current chip shortages for 5G, automotive, industrial machinery, electrical equipment, servers, etc. highlighted the supply competition among different countries and industries, and the necessity of making the critical supply chains less complicated, shorter, better controlled and more resilient
- The manufacturing of the global chip supply chains is mainly concentrated in East Asia, and manufacturing in the most advanced nodes below 10nm in Taiwan and South Korea.
- The decisions on the concentration of the critical manufacturing in one or two countries were almost only based on profit, without accounting for the fact that East Asia is a region of political conflicts and natural disasters
- The only-profit-driven wild globalization and chaotic resource exploitation results in a rapidly increasing fierce competition among different countries and industries for scarce resources, environment destruction and pollution

34

Broader context of the destruction

- Without understanding the broader context of the destruction we will not be able to effectively recover from it
- **The world is in constant war**: of **evil** against **good**.
- This war is "**eternal**" and has different phases of:
  - "**cold**" war, in the sense of moral, political, economic, etc., war

  and
  - "**hot**" war, in the sense of military conflict, revolution, and other types of enslavement and exploitation of people or destruction and looting of nature and all what humans created.
- Now this war between good and evil is a war between:
  - the world of civilization achievements being humanistic and ecological values, moral and social norms such as: human rights, democracy, self-governance, fair division of welfare, nature protection, etc.,
  - and
  - the backward old-fashioned world, negating humanistic and ecological values, negating moral and social norms such as: human rights, democracy, self-governance, fair division of welfare, nature protection, etc.

35

Broader context of the destruction

- Now this war between good and evil is a war between:
  - the world based on the state of law build on humanistic and ecological values, in which all are equal, and which protects everyone, a world where the government elected by the whole society in free and democratic elections acts for the social good within the law, and everyone has free access to information,

  and
  - the world of lawlessness of a totalitarian regime, negating humanistic and ecological values, denying and destroying moral and social norms, destroying or enslaving people, destructing and looting nature and all what humans created, and where society does not have free access to information and is manipulated by totalitarian propaganda.

Broader context of the destruction

- **Where is the front line between good and evil in this war**?

- Some say that this war between good and evil is between:
  - the world of the "West" build on a socially advanced civilization based on humanistic and ecological values, and social norms such as: human rights, democracy, self-governance, nature protection, etc.

  and
  - some other parts of the world where these rights and norms are not actually accepted and not followed by rulers and influential people.

- Is this the (whole) truth ???

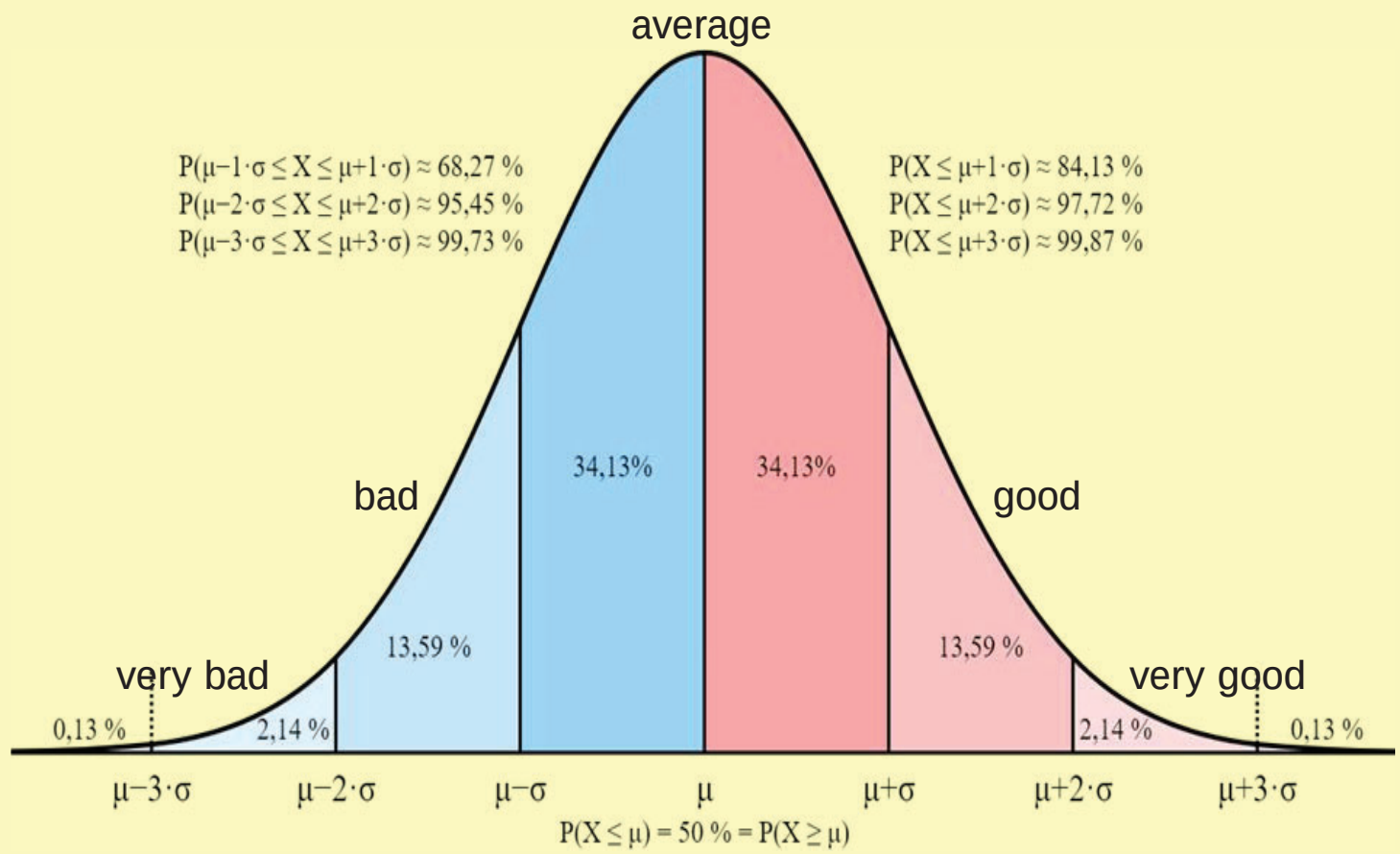
- **Definitely not !!!**

37

Broader context of the destruction

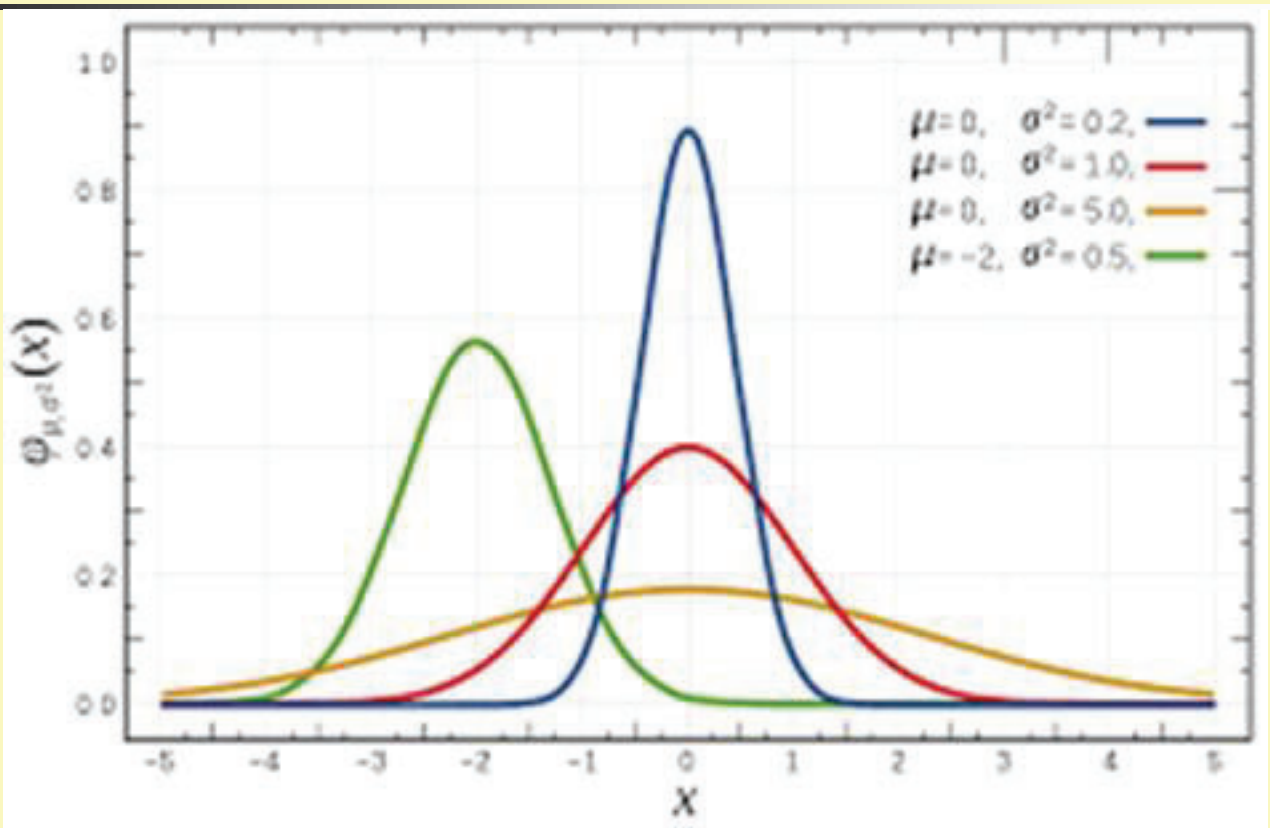
- In terms of a hot war, the front line between evil and good runs often between a totalitarian regime ruling a certain country and free nation of a neighbouring country
- The front line between evil and good is often between a totalitarian regime and a part of the society ruled by the totalitarian regime
- The front line between evil and good is often between a company owner not respecting people and environment, and the exploited company employees and destructed environment
- In general:
- **The war between good and evil is taking place all over the world, in every country and in every society**
- In each country and society, the distribution of characteristic features of people can be well modelled by a normal distribution
- In each country and society, there are "good" and "bad" people, but there are the most "average" people, less "good" and "bad", and only a small number of "very good" and "very bad" people.
- Likewise, with "smart" and "dumb" people

Broader context of the destruction



Free Public Licence received from https://creativecommons.org/licenses/by-sa/4.0/

Broader context of the destruction



The parameters of the normal distribution can be different for each country and for each society

Source: Wikipedia

40

Broader context of the destruction

- Observe that "bad" and "stupid" people are in every country and in every society, but in different countries can be in different proportions
- In particular, the stronger totalitarian and longer-lasting totalitarian a country is, the more heavily manipulated the society of that country is and the more the mean value of the normal distribution shifts towards "evil" and "stupidity"
- Actual supporters of totalitarian regimes are usually people who are “bad" or bemused by totalitarian propaganda
- It is a common knowledge that one can influence people, their thinking and their characteristics
- Let us observe:
  - how important the role of free access to information and "real" education is, and
  - how disastrous is the lack of free access to information and propaganda instead of "real" education and information

41

Let us act on the side of good

- As people belonging to the best educated part of our societies, let us not only be well educated, but also "good" and "wise".
- **Let us be on the side of good** in this war between good and evil.
- Let's not wait for someone to win this war for us.
- Let us actively work against evil and do good in all the most effective and efficient ways available to us.
- Let us work for respecting the humanistic and ecological values, and for human rights, democracy, self-governance, fair division of welfare, nature protection, etc.
- Let us inform and educate people.
- As scientists and engineers: let us create "green" cyber-physical systems.
- **How to recover from the environmental disaster**?

EUROPE Recognizes the CLIMATE and POLUTION CRISIS and starts to take serious measures

EU President **Ursula von der Leyen** unveiled Europe's "**Green Deal**" plan to fight the crises on Dec. 11, 2019



It represents a stepwise incremental approach to solve the problems

43

How to recover from the disaster?

- The agreed in July 2020 Next Generation EU fund of €750 billion to recover from the crisis caused by the COVID-19 pandemics will be added to the regular EU budget for 2021–2027 to result in approximately €1824.3 billion
- As much as 30% of the total amount will be devoted to the climate and environment in compliance with the Paris Climate Agreement
- US also came back to the Paris Climate Agreement and devoted substantial funds to the climate and environment, and many other countries follow
- To recover from the disaster, ***a model of a well regulated and controlled effective and efficient system has to be applied to all kinds of systems, collaboration chains and related flows, implementing***:
  - **regenerative, circular and more local economy**

  and
  - **global ecology**
- In particular, ***this applies to collaboration chains and related material and information flows in CPS and IoT***
- ***What is circular regenerative economy?***

44

Traditional versus Circular Regenerative economy

- Traditional economy is characterised by assumption of unlimited growth; competition; intensive exploitation of and fighting for non-renewable scarce resources; and short-term profit maximalization, without taking care of the negative long-term economic, social and ecological consequences
- Traditional economy uses linear model: take scarce resources – make – use – dispose waste; it did not pay the actual costs of inefficient resource usage and of the pollution and destruction it made
- Circular regenerative economy is a systemic approach that aims to benefit all: business, society and environment, through:
  - quality-based growth, collaboration and partnership;
  - increasing use of renewable resources, resource sharing and gradually limiting the use of finite resources;
  - introducing biological cycles to regenerate living systems and technical cycles implementing product repair, reuse, sharing, remake, and recycling; and this way minimizing the use of scarce resources and regenerating the environment

45

Innovate applying circular economy and quality-driven design

- The principles of the circular regenerative economy are derived from the same source as the principles of my paradigms of life-inspired systems and quality-driven design
- They are derived from the observation of nature, and especially of structures and operations of living organisms, their populations and ecosystems that have demonstrated to effectively, efficiently and robustly work for many millions of years, and are a great source of inspiration
- Therefore, in relation to technical systems the principles of the circular regenerative economy repeat the main principles of the paradigms of life-inspired systems and quality-driven design proposed by me more than 20 years ago
- Implementation of the circular regenerative economy will require **many breakthrough innovations of processes and products**
- All those innovations will have to be designed and implemented
- ***When designing and implementing the innovative processes and products the methodologies of circular regenerative economy and quality-driven design should be used***

46

We have to recover from this disaster ASAP

- Innovations exploiting modern CPS and IoT technologies, circular regenerative economy and quality-driven design can significantly improve systems used by us or that we are part of
- Significantly improve does not mean to completely solve the environmental crises
- For this, the unnecessary and inefficient consumption has to be eliminated and all social systems have to be re-organized and made much more efficient
- The principles of circular regenerative economy and the quality-driven design methodology should be used to develop high-quality collaborating cyber-physical systems
- In these systems the sophisticated intelligent cyber systems (controllers) will be tightly integrated with the intelligently controlled and optimized physical, social and life systems
- This way, we have a great chance to much better control and optimize the social, physical and life systems than we did it till now
- This way, we can create green cyber-physical systems

47

Environmental footprint of cyber systems

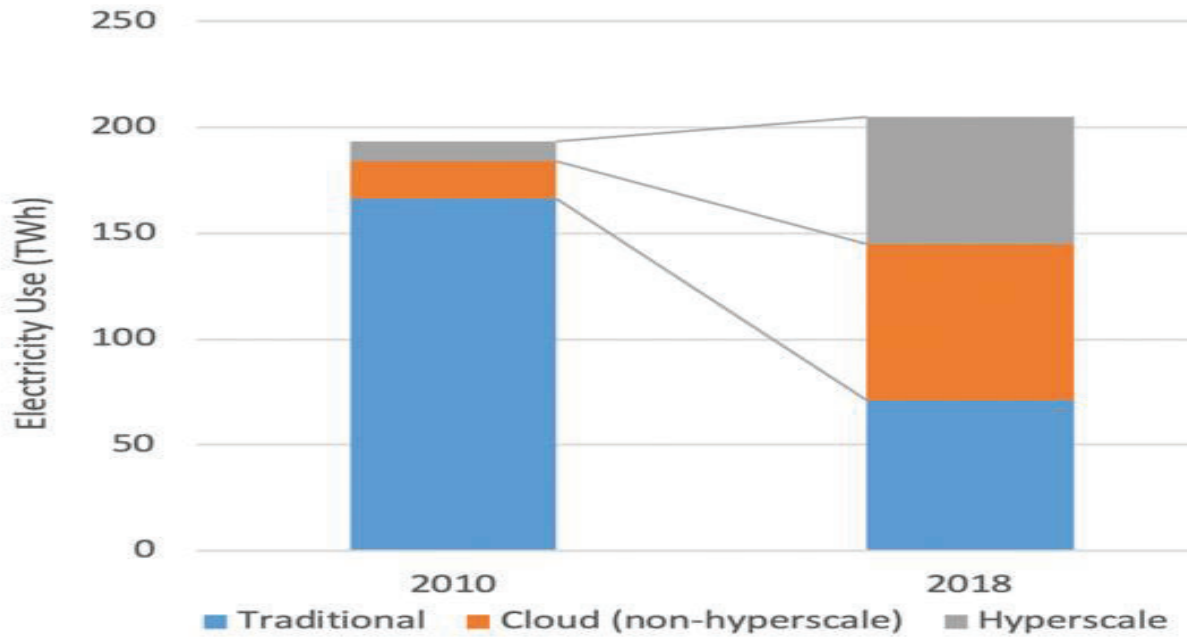
- According to https://www.energuide.be, the average energy consumption and CO_2 footprint of a contemporary computer are the following:
  - desktop (basic peripherals included): 200 W/hour in work mode; used for 8h a day *consumes 600 kWh and emits 175 kg of CO_2 per year*;
  - laptop: 50 and 100 W/hour in work mode; used for 8h a day *consumes between 150 and 300 kWh and emits between 44 and 88 kg of CO_2 per year*;
  - in stand-by mode: the consumption/emission of both decrease to a third of the above.
- For microcontrollers (MCUs) and MPSoCs used in CPS, the story is much more complicated
- For them, the actual energy consumed depends on very many factors
- It is difficult to speak about an average energy consumption even for a given single MCU or MPSoC, because the energy consumption very much depends on the actual use and working conditions
- The power consumed by MCU or MPSoC grows with operating frequency, temperature, supply voltage and signal activity

48

Environmental footprint of cyber systems

- Moreover, modern MCUs and MPSoCs often have several different active and energy saving modes (e. g. sleep, deep sleep, standby, etc.) and use the frequency and voltage scaling
- Finally, different MCUs and MPSoCs may have very different energy consumption characteristics, dependent on their architectures and implementation technologies, which in turn depend on the purposes/application fields which a given MCU or MPSoC is supposed to serve
- A simple ultra-low-power MCU for wearables can run in its active mode at much under 1W
- A complex MPSoC for automotive may use hundreds of Watts
- However, this is only a small part of the whole story
- The environmental footprint of cyber systems in CPS depends not only the embedded processors and their use, but on the usage of fog and cloud computers, and of the communication among all the computers as well

Environmental footprint of cyber systems



Source: https://energyinnovation.org/2020/03/17/

Figure 2. Estimated global data electricity use by data center type, 2010 and 2018. Source: Masanet et al. 2020.

- In 2018 global data centers consumed approximately 205TWh, what is more than the electric energy consumption of a medium country
- It represents 1% of global electric energy use and 0.3% of global CO_2 emission

50

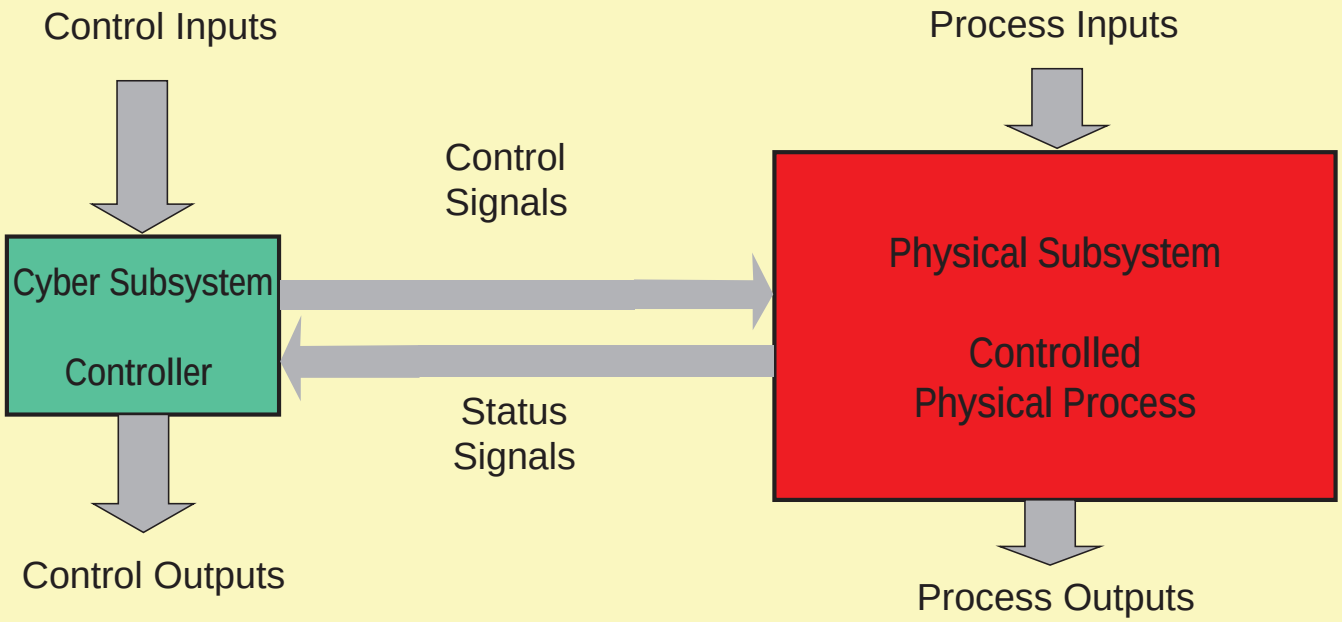
Environmental footprint of cyber systems

- Similarly, in 2019 global data transmission networks consumed around 250 TWh or somewhat more than 1% of global electric energy use, what corresponds to more than 0.3% of global CO_2 emission
- The demand for data center and network services is exponentially increasing.
- Between the 2019 and 2025, the number of IoT connections is expected to grow from 12 billion to 25 billion (https://www.gsma.com/mobileeconomy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf)
- To manage the environmental footprint of the CPS cyber systems, the exponential growth of CPS and IoT has to be compensated by efficient IoT organization and continuous energy efficiency improvements of embedded processors and MPSoCs, servers and storage devices, network processors and their software
- However, this is still only a small part of the whole story
- The environmental footprint of cyber systems depends not only on their use, but on their whole life cycle, including design, manufacturing, usage and disposal

51

Environmental footprint of cyber-physical systems

General Model of Cyber-Physical System



Usually:

Low Energy

High Energy

52

Environmental footprint of CPS

- The physical subsystem of CPS (implementing the controlled physical process) usually involves much larger material structures and flows, and several times more energy than the cyber subsystem (controller)
- The environmental and other effects are usually much larger from usage of the modern CPS and IoT technology to intelligently control and optimize the physical, social and life systems than from making green only the cyber systems
- We should make green the physical, social and life systems, as well as the cyber systems controlling them and the IoT connecting the collaborating CPS
- The environmental footprint of CPS and IoT depends on the whole CPS and IoT life cycle involving the CPS and IoT design, manufacturing, usage and disposal
- *Manufacturing* usually includes installation, testing and validation
- *Usage* often involves maintenance, repair and enhancement
- Let's start with IoT

Distribution of intelligence, computing resources, services and workloads in the IoT chierarchy

- To transform the big data from multiple sensors to the information being directly used for decisions, while satisfying the stringent requirements of the modern mobile systems, a careful distribution of information delivery and computation services among the different layers of IoT is needed
- For many reasons of primary importance, as:
  - real-time availability of local information
  - guaranteed real-time reaction
  - privacy, security, safety, reliability
  - minimization of energy used, communication traffic, costs, etc.

  a majority of computing and decision making related to advanced CPS should be performed locally in the IoT edge devices, in collaboration among various local IoT edge devices or just above the edge nodes, and not in the higher levels of fog or in cloud
- The higher levels of fog and cloud should only be asked for services if:
  - necessary information or computing resources are not available locally, and
  - reaction-time, security, safety, etc. allow for this

54

Distribution of intelligence, computing resources, services and workloads in the IoT chierarchy

- This requires implementation of advanced intelligent computations and sophisticated powerful embedded computing technology:
  - directly in the IoT edge devices related to the (complex) sensors and actuators, or
  - just above the edge nodes, where the information from different sensors can be combined and based on the combined information the control decisions can be taken and subsequently actuated
- Sophisticated and powerful edge computing has to be used requiring advanced intelligence, processing power and communication capabilities to be pushed towards the edge-nodes of IoT, where the data originate and information is used (i. e. to sensors, controllers and actuators)
- A very good example of the edge computing necessity is the **local** vehicle-to-vehicle and -infrastructure communication and collaboration necessary for autonomous driving
- In consequence, the **IoT for advanced CPS will be substantially different than Internet for other traditional targets**

55

Edge Computing, Intelligent Sensors, Edge AI and Edge ML

- This is the reason why Edge Computing, and specifically, intelligent sensors and actuators, as well as edge Artificial Intelligence (edge AI) and edge Machine Learning (edge ML) became very relevant and hot R&D topics recently
- Artificial intelligence (AI) is intelligence demonstrated by organized systems (e.g. machines), in contrast to "natural" intelligence demonstrated by organic systems (e.g. humans or animals)
- An intelligent system is a system that shows a goal-directed behavior
- AI system is a system that analyses the problem, and based on the analysis results, takes actions that maximize the chances of success to achieve the goal
- Machine learning (ML) is a learning implemented in machines through developing methods and algorithms that can "learn"', in the sense of being trained on some set of data, discovering the structure in data, or optimizing own performance for some set of problems through interacting with environment and processing feedback from the environment

Edge AI and Edge ML

- Usually, based on the training data machine learning methods/algorithms build/train a model which is then used to process additional data to make decisions or predictions
- Machine learning system is an organized system that implements one or more machine learning methods/algorithms
- Several types of learning approaches and models are known
- Depending on the nature of the input data and feedback used for learning the following three main machine learning approaches can be distinguished: supervised learning, unsupervised learning and reinforcement learning
- Supervised Learning:
  - the training data consists of sample inputs and the desired output for each sample input
  - the learning goal is to build/train a model that implements a general rule (function) that maps inputs to outputs

57

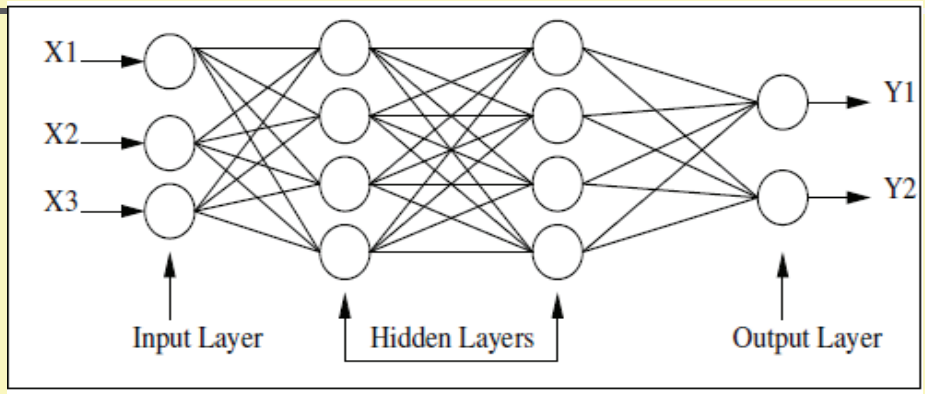
Edge AI and Edge ML

- Unsupervised Learning:
  - the training data consists of only inputs and no information on the desired outputs is given to the learning algorithm
  - the learning goal is to discover structure/patterns in the data (such as clustering, partitioning or other grouping) through finding and reacting to commonalities and differences in the data in order to react to new data based on the discovered structure and commonalities/differences
- Reinforcement Learning:
  - it does not need explicit training data (as e.g. some sample inputs and the desired outputs for the inputs);
  - it learns the optimal behaviour through interacting with its environment in which it has to achieve a certain goal (e.g. autonomously driving a car, planning an efficient and safe path for a robot, etc.) and observing how the environment responds to its actions;
  - it does it through rewarding desired actions and punishing non-desired once to discover the sequence of actions that maximize the cumulative reward that is usually expressed by a value function that defines the cumulative reward as a sum of the award of being in a certain state and the expected future award to be collected from that state further

Edge AI and Edge ML

- Although reinforcement learning methods/algorithms can be model-based or model-free (i.e. they can work without building an explicit model of the environment), a vast majority of ML methods/algorithms use various models, such as: artificial neural networks, support-vector machines, decision trees, belief networks, etc.
- In CPS and IoT, Machine Learning is used for a wide variety of important tasks, such as:
  - video and image processing
  - computer vision
  - speech processing and recognition
  - object (e.g. human, animal, car etc.) motion prediction
  - robot or vehicle path planning
  - and many more
- Machine Learning (ML) can be seen as a part of Artificial Intelligence (AI), although some researchers argue that they only have a large common part

Edge ML and Artificial Neural Networks (ANNs)



- ANN is a ML model involving nodes called neurons which are connected with edges called synapses that serve to transfer signals between neurons
- A neuron processes the received signals, when computing a non-linear function of the sum of its inputs, and then sends a signal to neurons to which its output is connected
- Neurons are organized into layers that may perform different transformations on their inputs
- Neurons and edges have weights that adjust during the learning process, and this way change the strength of the signals at a corresponding connections
- Neurons may also have a threshold, so that the signal is only sent from a given neuron if the aggregate signal of the neuron is higher than the threshold value

60

Edge ML and Deep Learning (DL)

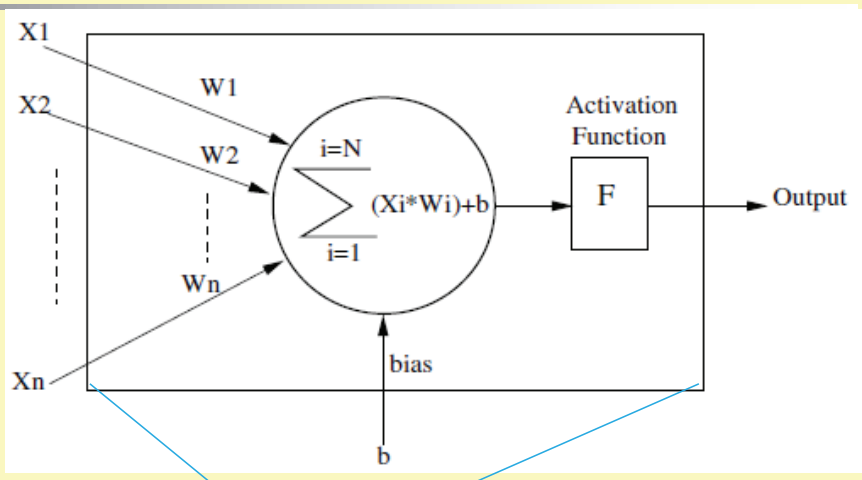
- The spectacular progress in the nano-dimension semiconductor technologies in the recent 10 years resulted in:
  - ability to implement much larger and more complex neural networks on a chip

  and
  - availability of powerful processing technology that makes it possible to efficiently use large data sets to train the networks
- In the recent 10 years the neural networks and related fields are undergoing a reincarnation in the form of the so called "deep learning"
- Deep learning refers to machine learning using complex ANNs with "multiple layers of nonlinear processing units and supervised or unsupervised learning of feature representations in each layer, with the layers forming a hierarchy from low-level to high-level features." (Wikipedia)
- Specifically, in 2015 several specific computing platforms for multi-layer Convolutional Neural Networks (CNNs) called Deep Neural Networks (DNN) have been developed, by Cognivue: G2-APEX IP - High Performance Image Cognition Processor Core, Qualcomm: Zeroth cognitive processor, Nvidia: Drive PX, Synopsys: DesignWare EV Processors, and by some other companies

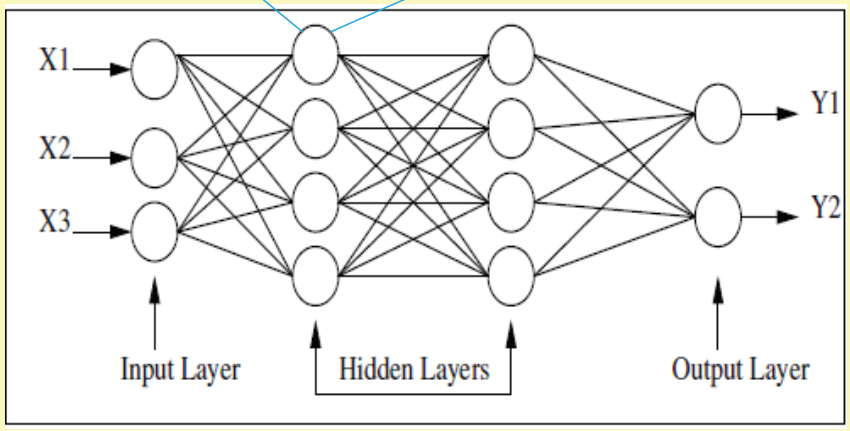
61

Convolutional Neural Networks (CNNs) and Deep Learning (DL)

- CNN is type of ANN in which neurons mainly compute convolutions on their input data



- CNN with several hidden layers is called a deep CNN (DNN)



62

New Edge Computing Platforms for ML and AI

- The interest in Machine (Deep) Learning and Artificial Intelligence is rapidly increasing (Research and Markets predicts that AI in IoT will reach a value of $14,8 billion by 2026)
- ML and AI technologies belong to main contributors to modern CPS and IoT, but they are also expected to substantially contribute to the solution of the environmental crises
- In the last two years many different new Edge computing platforms and accelerators for Deep Learning, other learning and other AI have been developed
- Synaptics developed Katana ultra-low power Edge AI SoC for a wide range of energy constrained IoT applications (e.g. sensors and edge devices in offices, factories, warehouses, robotics, farms, smart homes and cities, etc.)
- Katana has a heterogenous hexa-core architecture, with each core optimized for specific set of tasks, such as ANN image processing, audio/voice processing, control, etc. Its power efficiency mainly results from the combination of efficient specialized cores, sophisticated dynamic voltage and frequency scaling mechanism, and efficient (model) development and optimization tools.

63

New Edge Computing Platforms for ML and AI

- Silicon Labs implemented a low power AI/ML accelerator in the BG24 and MG24 Gecko wireless SoCs for IoT applications, such as such as medical, industrial, and smart home and town.
- It can speed up IoT AI/ML workloads up to four times with a resulting six-fold power savings compared to Cortex-M33, what makes BG24 and MG24 suitable for battery-operating IoT devices
- GreenWaves developed Gap9 ultra-low power neural network Edge processor suitable for battery-powered devices and optimized for advanced audio. The total power consumption for Gap9 can be as low as 1.8 mW
- quadric developed q16 processor implementing "Quadric architecture" being a general-purpose neural processing unit (GPNPU). Quadric's GPNPU architecture is claimed to be appropriate for a wide range of processing (including ML, general-purpose control, and signal processing) and delivering a high performance for both ML inference and conventional signal processing, as e.g. for computer vision
- Quadric announced that it starts marketing its edge AI processor architecture as a licensable intellectual property (IP).

64

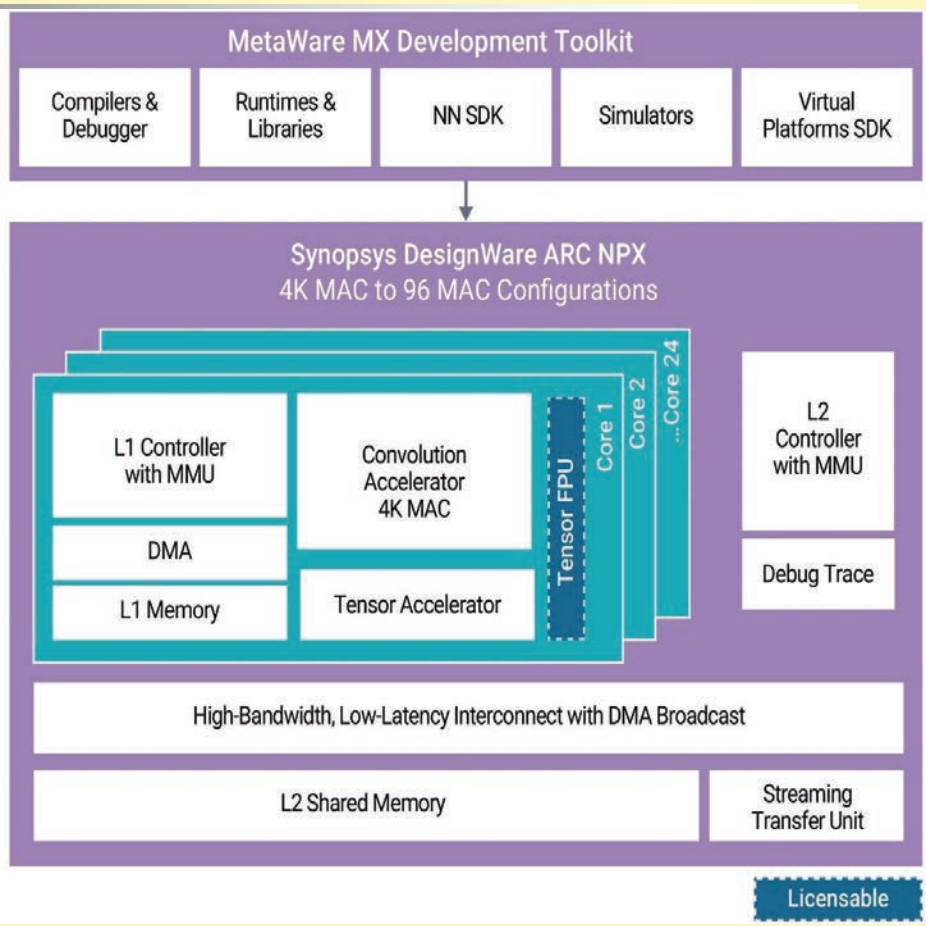
New Edge Computing Platforms for ML and AI

- Intuitive developed NU4000 high-performance heterogeneous multi-core SOC for robots, drones, VR/AR, etc. that involves a. o. a high-quality 3D depth engine, SLAM accelerators, strong Computer Vision engine and deep-learning CNN processor
- It is a powerful imaging, vision and AI computing platform with total performance exceeding 8 TOPS
- The CNN processor provides above 2 Terra OPS, and it enables processing of large deep CNNs (e.g. VGG16) at the rate of 40 frames (ROIs) per seconds at ~10 times less power than equivalent GPU, DSP or FPGA based processing
- For deep-learning with advanced CNN models the Synopsys DesignWare® ARC® EV processor IP is used in NU4000
- Synopsys DesignWare ARC NPX Neural Processor IP family provides a high-performance, power- and area-efficient IP solution for applications requiring AI supported computer vision (e.g. for object detection, scene segmentation, image quality improvement), and for broader range of applications (e.g. audio, natural language processing, etc.)

65

New Edge Computing Platforms for ML and AI

- The NPX6 NPU IP architecture is based on individual cores that can scale from 4K MACs to 96K MACs for a single AI engine performance of over 250 TOPS and over 440 TOPS with sparsity
- The IP includes hardware and software support for multi-NPU clusters of up to 8 NPUs achieving 3500 TOPS with sparsity
- An optional tensor floating point unit is available for applications benefiting from BF16 or FP16 inside the neural network.



Source: Synopsys

66

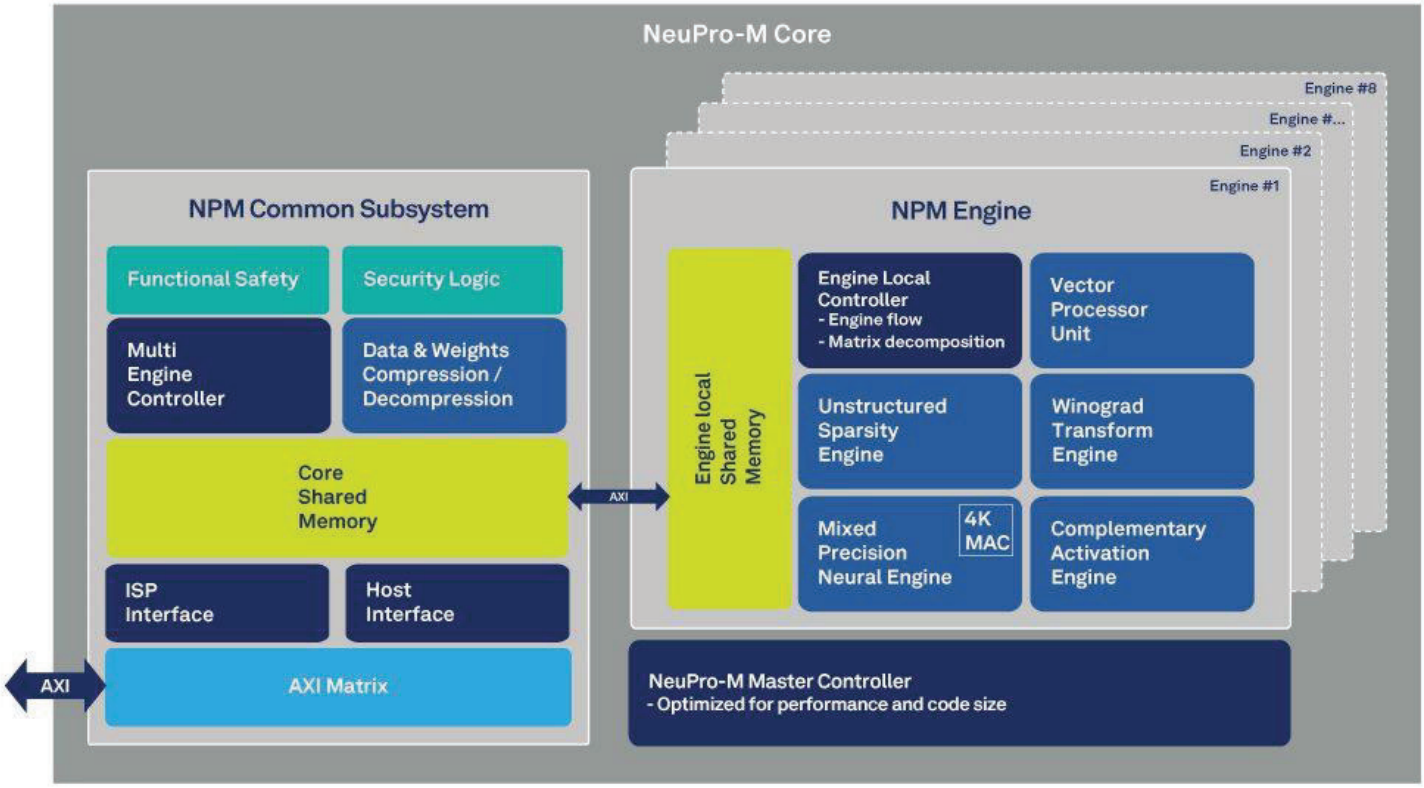
New Edge Computing Platforms for ML and AI

- CEVA's new NeuPro-M scalable heterogeneous high-performance AI processor architecture IP is focused on edge AI/ML applications where both the high-performance and power efficiency are of importance, such as automotive and other mobile systems, robotics, industrial IoT, etc.
- It achieves from 8 TOPS (with a single NPM engine) up to 160 TOPS (with 8 NPM engines) per core, and can reach above 1200 TOPS with multi-core instantiations, with a remarkable power efficiency of up to 24 TOPS/Watt
- It is accompanied by and CEVA Deep Neural Network (CDNN) development software including:
  - NeuPro-M system architecture planner for ANN deployment over NeuPro-M
  - ANN training optimizer

  and
  - CDNN compiler

67

New Edge Computing Platforms for ML and AI



Source: CEVA

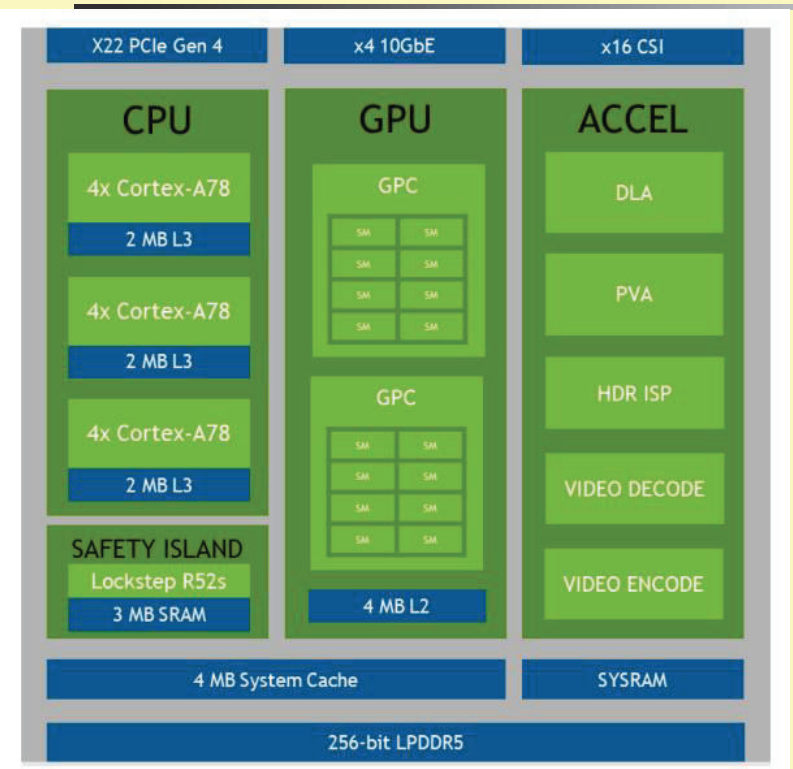
68

New Edge Computing Platforms for ML and AI

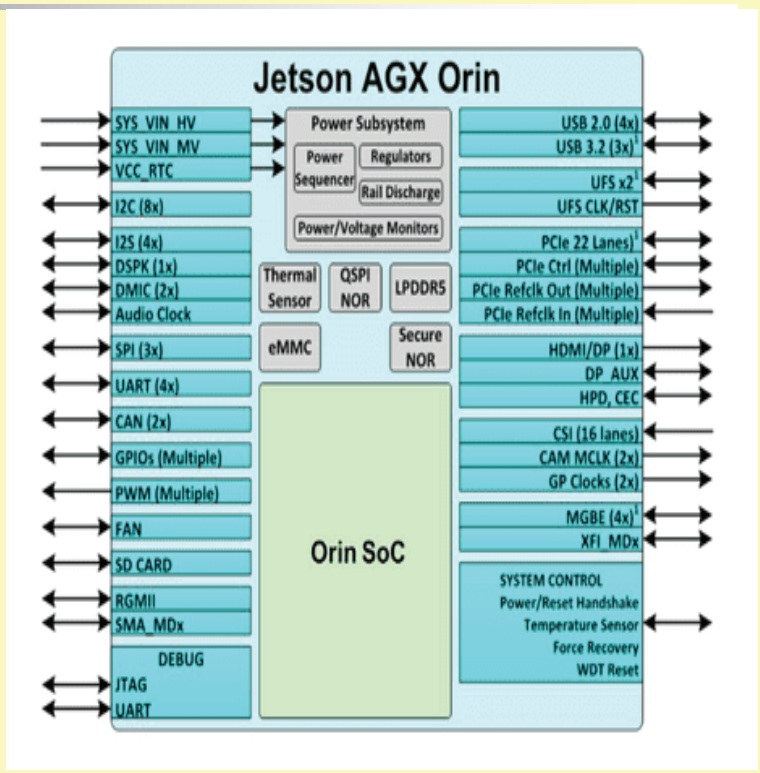
- NVIDIA introduces new Jetson AGX Orin System-on-Module (SoM) for powerful high-performance and energy-efficient AI/ML at the edge
- It is aimed at the most advanced applications requiring powerful embedded computing at the edge in such sectors as advanced medical devices, autonomous cars, autonomous delivery, logistics and factory robots, advanced UAVs, and other advanced autonomous systems, for highly demanding tasks of multi-sensor fusion, computer vision, motion prediction, path planning, natural language understanding, etc.
- Jetson AGX Orin delivers up to 200 TOPS AI performance, which is comparable to the performance of a GPU-based server, but has a size of only 100mm x 87mm and uses much less power (15 – 40 W)
- Jetson AGX Orin SoM is built around Orin SoC with Nvidia's GPU Ampere architecture with 1792 NVIDIA® CUDA® cores and 56 Tensor Cores in two Graphic Processing Clusters (GPCs), 8-core ARM Cortex-A78AE CPU, powerful HW deep learning accelerator (DLA) and vision accelerator (PVA), video encoder and video decoder
- High speed I/O and memory with 204 GB/s bandwidth make it possible to run multiple concurrent AI pipelines

69

New Edge Computing Platforms for ML and AI



Orin SoC Block Diagram



Jetson AGX Orin System-on-Module

Source: NVIDIA

70

New Edge Computing Platforms for ML and AI

- Mobileye introduces its EyeQ Ultra high-performance and low-power SoC aimed at autonomous vehicles and similar advanced applications
- EyeQ Ultra is fabricated in 5-nm process and delivers AI performance up to 176 TOPS at less than 100 W
- It has a very heterogeneous architecture involving several different types of cores tuned to different tasks involved in an L4 autonomous car, including:
  - 12 RISC-V CPU cores,
  - Arm GPU and VPU,
  - 4 types of Mobileye's proprietary accelerators involving 16 CNN accelerators, 8 CGRA-based cores, 16 VLIW/SIMD cores, and 24 barrel-threaded CPU cores,
  - video encoding/decoding cores, safety/security subsystem, two separate sensor subsystems: one camera-only, and the other one for radar and lidar, etc.
- Each of the two separate sensor subsystems can support a full operation, and this redundancy results in a more robust overall system.

71

New Edge Computing Platforms for ML and AI



Source: Mobileye, an Intel Company

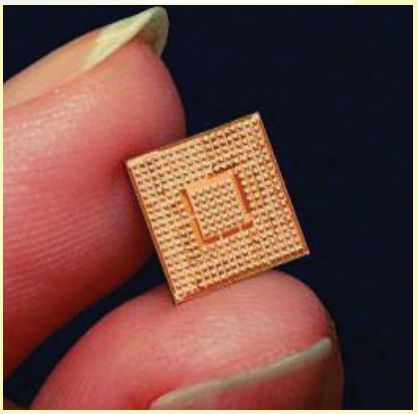
72

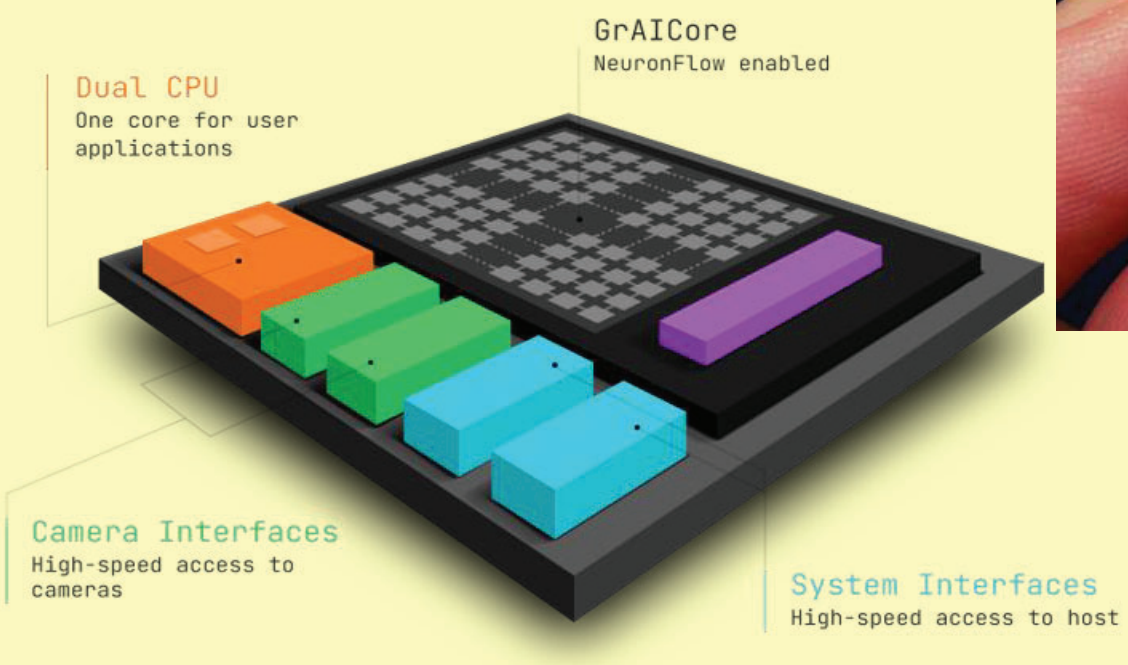
New Edge Computing Platforms for ML and AI

- GrAI Matter Labs (Eindhoven, Paris, San Jose) is introducing GrAI VIP Edge AI SoC for high-performance and energy-efficient AI/ML at the edge, aimed at near-sensor AI/ML based solutions in robotics, industrial automation, AR/VR, Smart Homes, Infotainment in automobiles, etc.
- GrAI VIP SoC is based on GrAICore™ neuron AI engine, and involves embedded ARM processors, and interfaces to be connected to a multitude of sensors (e.g. vision, sound, pressure, etc.) to enable Life-Ready AI
- GrAI VIP SoC is manufactured in 12nm TSMC process and has a 8mmx8mm compact package with memory included
- It can can execute complex AI applications based on advanced DNNs, such as ResNet-50, EfficientNet, SSD, Yolo, Unet, etc. with very low inference latencies (only few ms for ResNet-50) and low-power (< 0,5 W for ResNet-50)
- GrAIFlow™ SDK provides a complete toolkit to conceptualize, visualize, test, debug, and iterate user applications

73

New Edge Computing Platforms for ML and AI





GrAI VIP SoC

Source: GrAI Matter Labs

74

New Edge Computing Platforms for ML and AI: **Summary**

- Very many new Edge Computing Platforms for ML/AI have been developed recently, and the above discussed constitute only a subset of them
- There is no one computing platform that fits all applications
- Different AI/ML computing platforms with different performance, energy usage, size etc. are needed for different CPS and IoT applications having their different characteristics and different requirements
- For Edge Computing it is necessary to realize the required computations when guaranteeing the required performance at the minimum possible energy used
- Nevertheless, the platforms have much in common
- All the computing platforms for Edge AI/ML are heterogeneous and implement massive parallelism

75

What can be the future in Computing Platforms for ML and AI?

- Companies will introduce their next-generation more powerful platforms
- For example: Nvidia is introducing its Orin SoC, but it already announced its next-generation much more powerful Drive Atlan that is expected to process AI/ML loads with 1000 TOPS.
- European Processor Initiative (EPI) will perhaps propose some solutions, as EPI aims a.o. at Accelerator Processors and Edge Systems?
- More energy efficient implementation technologies will be used
- For example: The StorAIge project led by STMicroelectronics aims to develop and industrialize the FD-SOI 28nm process and a very efficient embedded Phase Change Memory (ePCM), what will enable high-performance and ultra-low power SoCs for Edge AI (the tearget is 10 TOPS/W)
- Other examples can be analog implementation and in-memory computing (e.g. DIANA chip from Dutch AI startup Axelera, Eindhoven), use of photonic IC technology, etc.
- Development and use of a more efficient "neuromorphic" computing that will much better mimic the biological brain processes than the currently used neural networks
- etc.

76

Main IoT Networking Technologies and Standards

- As earlier explained: the IoT for advanced CPS will be substantially different than Internet for other traditional targets
- Specifically, due to different application requirements in relation to connectivity (data rate, latency, etc.), deployment area, number of connected devices, energy consumption, safety, security, reliability, cost, etc. different networking technologies, standards and protocols will be used
- The following two kinds of IoT applications are distinguished in relation to two distinct areas of the requirement spectrum: Massive IoT and Critical IoT
- Massive IoT refers to applications that require a huge number (from thousands to milliards) of low-cost and low-energy devices often in remote locations, each generating a small number of (regularly) reported data, and that have relatively low throughput and latency requirements:
  - Aim: to efficiently transmit small amounts of data from the huge number of devices
  - Key requirements: sufficient network capacity, scalability, security and availability, wide and strong coverage, (ultra) low-power/energy, low cost
  - Example Applications: smart metering, smart building/city, smart grid, asset tracking, fleet management, wearables and part of e-health, process monitoring and optimization in indystry, environmental monitoring, climate monitoring and livestock tracking in agriculture, etc.

77

Main IoT Networking Technologies and Standards

- Critical IoT refers to time- and safety-critical applications that demand data delivery within a specified time and with required guarantees, and that usually involve fewer (up to thousands) complex costly devices, each generating/receiving large amount of data with high throughput and low latency requirements, and that have to withstand harsh/remote environments, as well as security threats and attacks:
  - Aim: to guarantee efficient transmission of large amount of data with high throughput and low latency in harsh environment and while facing security threats and attacks
  - Key Requirements: guaranteed high-bandwidth, low-latency, and very high security, safety, reliability, and availability, at low energy and acceptable cost
  - Example Applications: Autonomous Vehicles and V2X, UAVs, Robotics, Industry 4.0, telemedicine, VR/AR/MR applications, traffic and flight control and safety, critical part of smart city, etc.
- For massive IoT applications requiring:
  - low-power, wide area connectivity, security and availability, cellular network standards LTE-M and NB-IoT can be used
  - very low power from the device to send/receive data, very many connected devices/large area and lower cost, some LPWANs, as LoRa or Sigfox, can be used

78

Main IoT Networking Technologies and Standards

- For home appliances and similar consumer devices and applications WiFi, Bluetooth, Thread or Zigbee can be a satisfactory and low-cost solution, and the recently introduced Matter uses a combination of WiFi, Bluetooth Low Energy and Thread to enable devices and applications interoperability
- From the above it is clear that 5G is not always required and not always the best option for IoT
- However, 5G is indispensable for Critical IoT, as it provides Network Slicing, and much higher bandwidth, lower latency, lower power consumption, and higher safety, security and reliability than 4G
- Using Network Slicing the service provider can devote a part of the 5G radio spectrum to run a separate private wireless network for a company, or an NB-IoT massive service connecting thousands of sensors, or to enable higher bandwidth and lower latency for some highly demanding applications as autonomous vehicles or UAVs
- Allied Market Research reported that the global market of 5G infrastructure industry was $2.06 billion in 2020, and the market will grow to $83.62 billion by 2030, at a CAGR of 45.3 percent between 2021 and 2030

Quality-driven design approach

- To develop green collaborating CPS the principles of circular regenerative economy and the quality-driven design methodology should be used
- **System design is a *definition of the required quality***, i. e. a satisfactory answer to the following two questions:
  - **What new** (or modified) **quality is required**?
    and
  - **How can it be achieved**?
- Intuitively we feel that **quality** is here used in the sense of ***the totality of the (important) features the system has***
- So, **system design should define**:
  - **What is the required totality of the (important) system features?**
    and
  - **How to realize a system that has these all features**?
- In other words:
  - What process must be realized in a certain system and what structural and parametric features must have the system?
  - How can we build a system that will be able to realize this process and will have the required structural and parametric features?

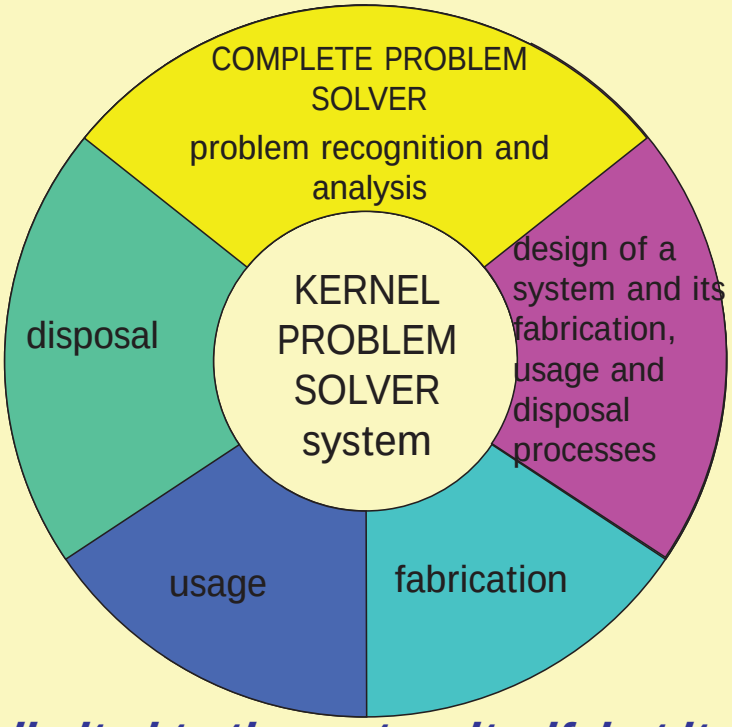
80

Quality

- Actually, **what is quality?**
- The most used and cited definitions of quality:
  - fitness for use (*Juran*)
  - conformance to requirements (*Crosby*)
  - quality is meeting the customers' expectations at a price they can afford (*Deming*)
  - the loss of quality is the loss a product causes to society after being shipped, other than any losses caused by its intrinsic functions (*Taguchi*)
  - the totality of features and characteristics of a product or service that bear on its ability to satisfy given needs (*American Society for Quality Control*)
  - the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs (*ISO8402: Quality Vocabulary Part 1)*

Problems with the existing definitions of quality

they focus exclusively on a product being designed, while the original problem is solved by designing, fabrication, usage and disposing of the system



Quality cannot be limited to the system itself, but it must account for the complete problem solution, related to complete system life-cycle

82

Problems with the existing definitions of quality

- none of these definitions is precise enough to enable the systematic consideration, measurement and comparison of quality
- the assumption of perfectly known and inviolable customer's requirements is not acceptable, because the customer may specify the requirements poorly and such requirements may result in system which will create danger, damage environment or squander scarce resources
- **engineered systems** solve certain real-life problems, serve certain purposes – they are **purposive systems**
- quality of a purposive system can only be defined in relation to its purpose

83

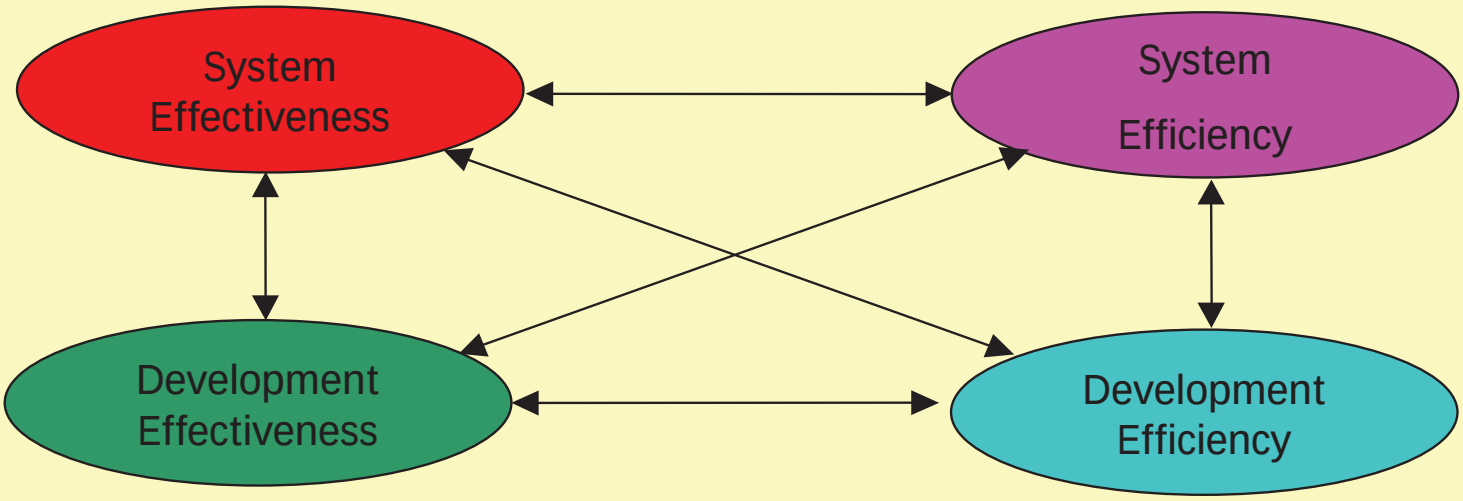
New quality definition proposed by me 20 years ago

__Quality__ of a purposive systemic solution is
its __total effectiveness and efficiency__
in solving of the real-life problem that defines the solution's purpose

- **Effectiveness** = the degree to which a solution attains its goals
- **Efficiency** = the degree to which a solution uses resources in order to realize its aims
- ***Effectiveness and efficiency of a systemic solution together decide its grade of excellence*** - **their aggregation expresses quality**
- Effectiveness and efficiency can be expressed in terms of measurable parameters, and in this way, quality can be modeled and measured
- In particular, the quality can be modeled in the form of *multi-objective decision models* involving measurable design parameters
- *The multi-objective decision models* and *design parameter estimators* enable application of the *multi-objective decision methods* for construction, improvement and selection of the most promising solutions

84

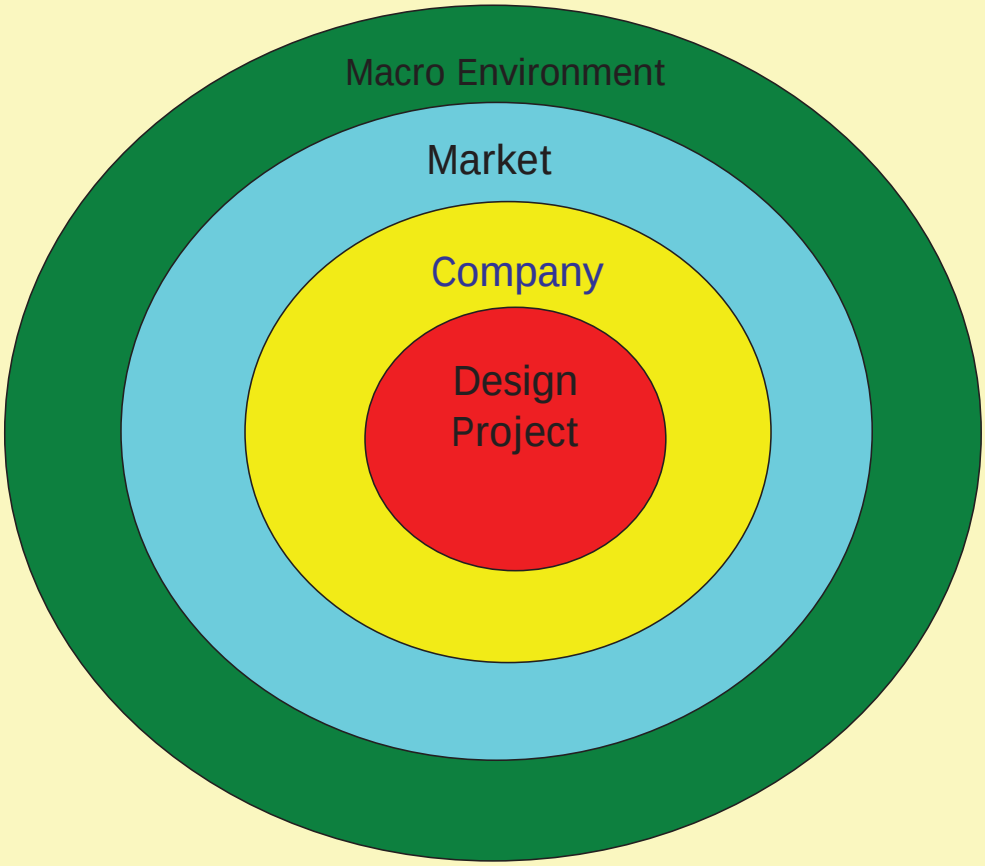
Quality-driven Design - Difficulties



Interactions and trade-offs between various parts and aspects of the total systemic solution

85

Quality-driven Design - Difficulties



Interactions of a design project with its context

86

Quality-driven Design - Difficulties

- Design does not concern the reality as it is, but as it will possibly be realized
- Quality recognition and formulation, i.e. recognition of the problem, as well as of the nature of its solution are ***subjective*** to a high degree
- The **contemporary system design problems** are ***complex***, ***multi-aspectual***, ***dynamic***, and ***ill-structured***:
  - there is no definitive formulation of the problem,
  - any problem formulation may be inconsistent,
  - formulations of the problem are solution dependent,
  - proposing and considering solutions is a means for understanding the problem, and
  - there is no definitive solution to the problem

87

Quality-driven Design - Difficulties

- The complex design problems are ill-defined
- It is very difficult to find precise relations between various aspects of the system effectiveness and between the different forms of energy and matter used to attain the system's aim, and even more difficult to express them as one uniform measure
- There are trade-offs as well between effectiveness and efficiency as among different their aspects
- The required quality or its perception can change in time



quality cannot be well defined,
but it can and should be modelled

Quality-driven Design - Design models

- *Well-structured models of the required/delivered quality* can serve to:
  - conceptualize, denote, analyse and communicate the customer's and designer's ideas
  - show that the requirements and designs are meaningful and correct
  - guide the design process
  - enable the explicit and well-organized design decision making
  - enable design automation
  - etc.

Quality-driven Design: Design problem-solving using models

- Since the system design problems are:
  - complex;
  - multi-aspect;
  - ill-defined,

  to solve them, ***all human concepts for dealing with complexity, diversity and ill-structure have to be applied***:
  - abstraction;
  - separation of concerns;
  - decomposition and composition;
  - generalization and specialization;
  - modelling;
  - simulation;
  - prototyping;
  - .....
- ***A design problem has to be converted into a system of simpler sub-problems***
- The solution to the original problem can then be achieved by solving the sub-problems and composing the sub-problem solutions into an aggregate solution

90

Quality-driven Design: Design problem-solving using models

- The problem decomposition and design modelling are to some degree subjective
- The design decision processes are also to some degree subjective, as they are influenced by the designers' value systems, feelings, believes, intuition etc.
- The design problem solving activity is performed under uncertainty, inaccuracy, imprecision and risk conditions, and in a dynamic environment



- ***System design has to be an evolutionary process*** in which analysis and modelling of problems; proposing their solutions; analysis, testing and validation of the proposals; learning and adapting are very important

91

Main concepts of the quality-driven design

- Designing ***top-quality systems is the aim*** of a design process
- ***Quality is modelled and measured*** (in particular, in the form of the multi-objective decision models) to enable invention and selection of the best alternatives and quality improvement
- ***Quality models are considered to be heuristics for setting and controlling the course of design***
- ***The design process is evolutionary*** and it basically **consists of**:
  - constructing the tentative quality models,
  - using them for constructing, improving and selecting of the tentative solutions,
  - analysing and estimating them directly and through analysis of the resulting solutions,
  - improving the models, and using them again to get improved solutions, etc.

92

Quality-driven Design: Limiting the design subjectivity

- **One of the main aims** of using the well-defined quality models in design is:

  Limiting the scope of subjective design decision making and ***enlarging the scope of reasoning-based decision making with clear and well-defined rational procedures*** which can be ***computerized***

- Too much subjectivity in design may result in solutions that either do not solve the actual real-life problem or do not do it in a satisfactory manner

- Limiting the design subjectivity in an appropriate manner, when enabling the creativity exploitation at the same time, *is necessary to arrive at the high-quality designs*

- The **main means for limiting the design subjectivity** is the ***design space exploration (DSE) with usage of the well-structured quality models***

93

Quality-driven Design: Limiting the design subjectivity

- **Exploration** of the abstract models of the required quality and more concrete solutions obtained with these models:
  - *gives much and more objective information* on the design problem, its possible and preferred solutions, and various models used in this process
  - *enhances exploitation of the designer's imagination, creativity, knowledge and experience*
- **Other important means for limiting the design subjectivity** and for **increasing quality** this way include:
  - appropriately organised team-work
  - benchmarking and comparison with both own previous designs and designs of competition
  - design analysis and validation
  - design reuse
  - government and branch **regulations and standards**

Quality-driven Design: Government regulations and standards

- ***Adequate government and branch regulations and standards are of primary importance for bringing into effect the green systems and green economy***
- Regulations and standards specify what is allowed or standard, and what is not
- They constitute general constraints for the industry and system designers that have to be satisfied by their designs, products and services
- Of course, particular systemic solutions satisfying these general constraints can still be very different, better or worse for the environment, but ***all systemic solutions have to satisfy the minimum required by the regulations and standards***
- Remember that the decisions made by companies and governments that caused the environmental destruction were mainly driven by short-term profit, without accounting for long-term consequences
- It would be naïve to expect that all companies and individuals will suddenly become environment-friendly without adequate regulations pressing them to do so

95

Quality-driven Design - Design requirements

- The general model of the required system's quality is represented by the ***system (design) requirements***
- System requirements can only be treated as *a non-perfect and tentative model of the required quality*
- Requirements and solutions obtained with their use are *subject to design and change*
- They should be confronted with the actual up-to-date needs many times during the design process, and replaced or modified, if necessary
- Design requirements model the design problem at a hand through *imposition of constraints and objectives in relation to the acceptable or preferred problem solutions*
- It is possible to distinguish **three sorts of requirements:**
  - ***functional***,
  - ***structural***, and
  - ***parametric***

Quality-driven Design - Design requirements

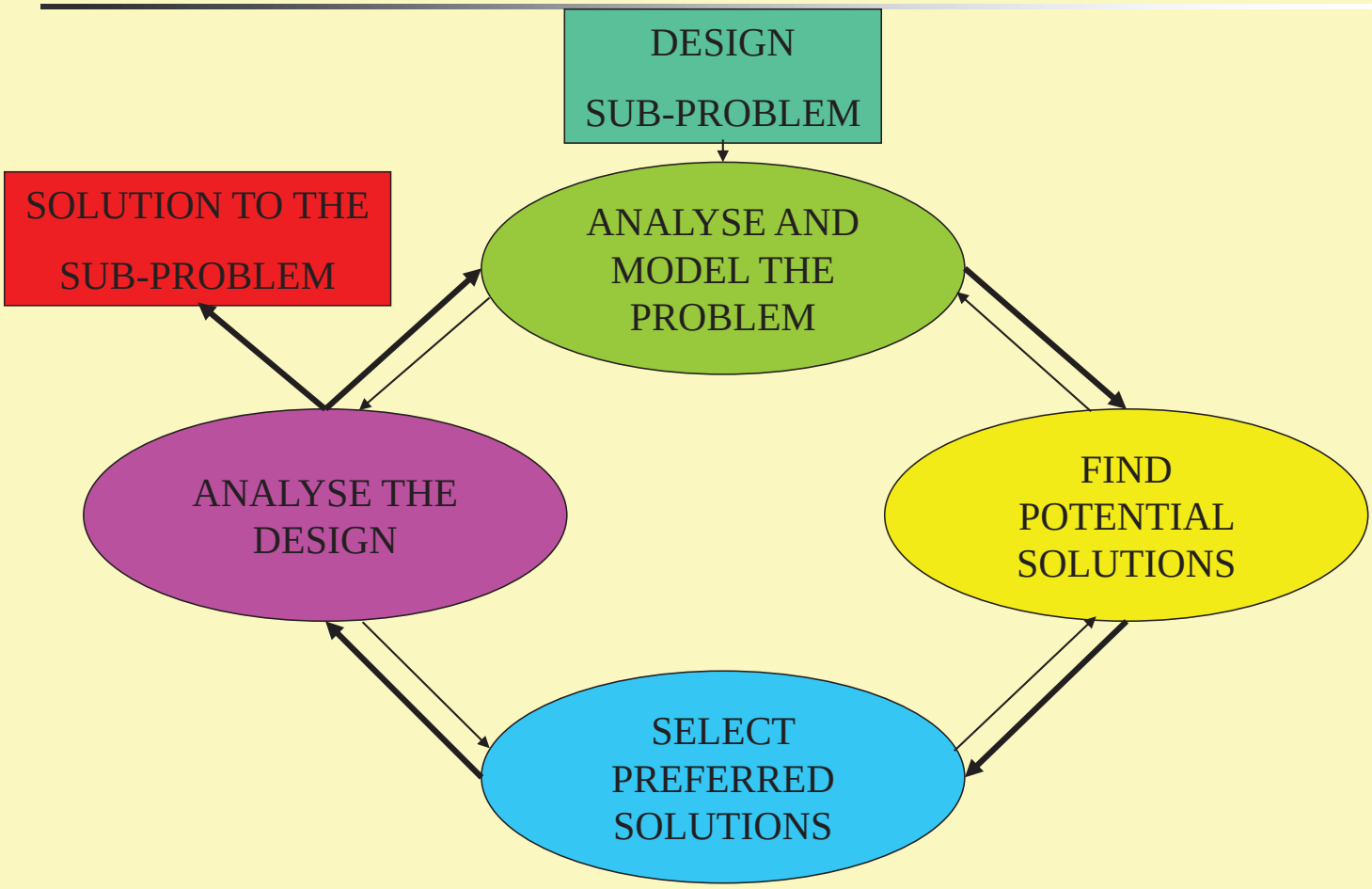
- All the three sorts of **requirements impose *limits on the structure of a required solution***, but they do it in different ways
- The ***structural requirements*** define the acceptable or preferred solution structures directly, by limiting them to a certain class or imposing a preference relation on them
- The ***parametric requirements*** define the structures indirectly, by requiring that the structure has such physical, economic or other properties (described by values of some parameters) as fulfil given constraints and satisfy stated objectives
- The ***functional requirements*** also define the structures indirectly, by requiring the structure to expose a certain externally observable behaviour that realizes the required behaviour

Quality-driven design space exploration (DSE)

- ***System design is an evolutionary quality engineering process*** in which the concepts of analysing and modelling problems, proposing their solutions, analysing and testing the proposals, learning and adapting are very important
- It **starts** with an ***abstract***, and possibly *incomplete, imprecise,* and *contradictory*, ***initial quality model*** (initial requirements)
- It tries to **transform** the initial model into a ***concrete, precise, complete, coherent and directly implementable final quality model***
- Usually, the initial abstract model mostly involves some *behavioural and parametric characteristics* and to a lesser extend the structure definition
- The **final model** defines the ***system's structure explicitly***
- This structure supports the system's required behaviour and satisfies the parametric requirements

98

Generic model of the quality-driven design space exploration

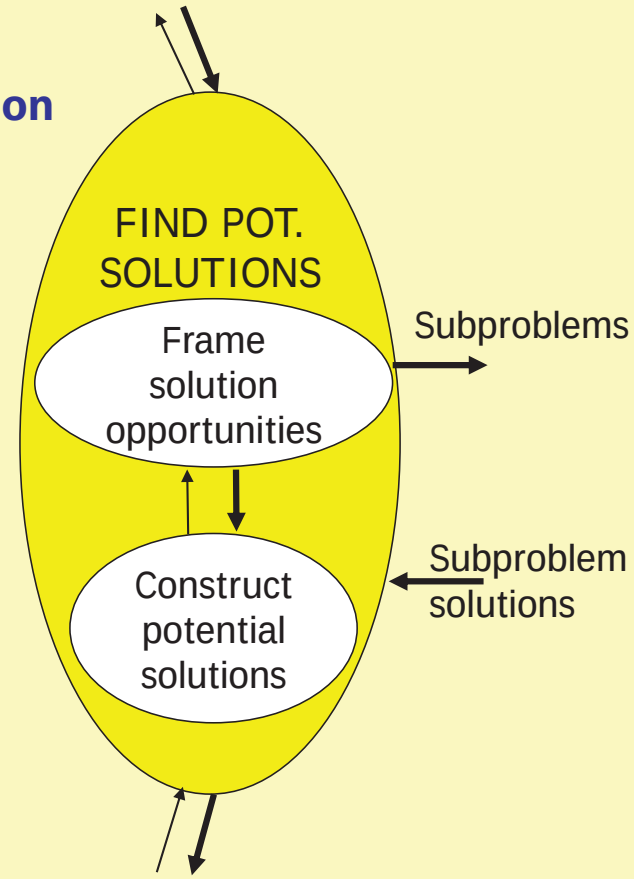


99

Generic model of the quality-driven design space exploration

- The **quality-driven design space exploration** basically consists of the alternating phases of:
  - ***exploration of the space of abstract models of the required quality***

    and
  - ***exploration of the space of the more concrete issue's solutions*** obtained with the chosen quality models



100

Quality-driven design space exploration

- In result of the design space exploration, the considered system is defined as an appropriate *decomposition into a network of sub-systems*
- Each sub-system solves a certain sub-problem
- All *sub-systems cooperating together solve the system design problem* by exposing the external *aggregate behaviour and characteristics* which *match the required behaviour and characteristics*
- The design process breaks down *a complex system* defined in *abstract and non-precise terms* into *a structure of cooperating sub-systems* defined in *more concrete and precise terms*, which are in turn further broken down to the ***simpler sub-systems that can be directly implemented with the elements and sub-systems at the designer's disposal***

101

Conclusion

- Systemic drawbacks of the traditional economy and cumulation of bad decisions made by numerous governments and companies without accounting for long-term consequences resulted in the **huge global environmental disaster**
- To recover from the environmental disaster and further develop:
  - *a model of a well regulated and controlled effective and efficient system should be applied to all kinds of systems, collaboration chains and related flows*
  - *modern CPS and IoT technologies should be used to much better control and optimize the social, physical and life systems than till now*
  - *methodologies of circular regenerative economy and quality-driven design should be used to design the systems*
- Innovations exploiting modern CPS and IoT technologies, circular regenerative economy and quality-driven design can significantly improve systems used by us or that we are part of
- In this CPS&IoT Summer School you will have a unique occasion to be informed on and to discuss the most recent European R&D developments in CPS and IoT

102


CSIT
CENTRE
FOR SECURE
INFORMATION
TECHNOLOGIES
DPU Technology
Redefining next generation
Datacenter Security
QUEEN'S
UNIVERSITY
BELFAST
Prof. Sakir Sezer
CPS&IoT'2023 Summer School

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

Overview

- Evolution of Data Center Server Architecture
- SmartNIC - Definition and Architecture
- Challenges of enabling defensive cybersecurity within the cloud
- Enabling Smarter Security on DPUs
- DPU enabled Security use-cases = classical IPS and AI enabled
- Challenges and opportunities

Queen's University Belfast

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES





Queen's University Chancellor
Secretary Hillary Clinton

3

Nvidia Belfast

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

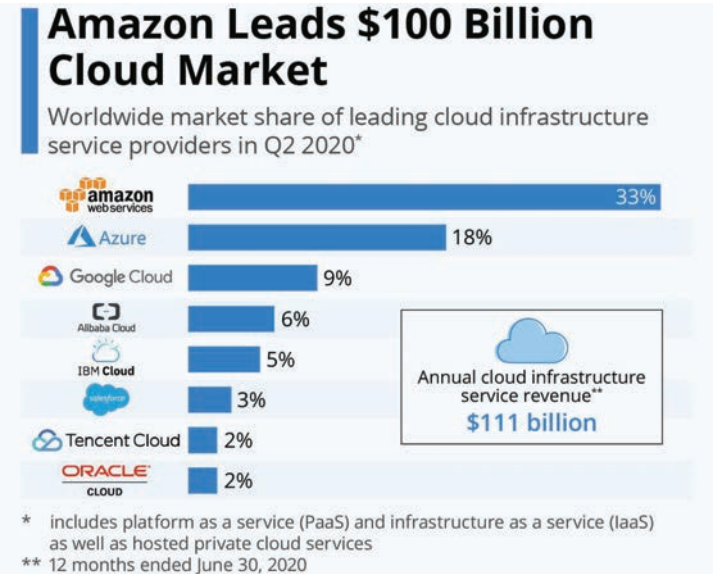


UK Minister visit Steve Baker (MP)

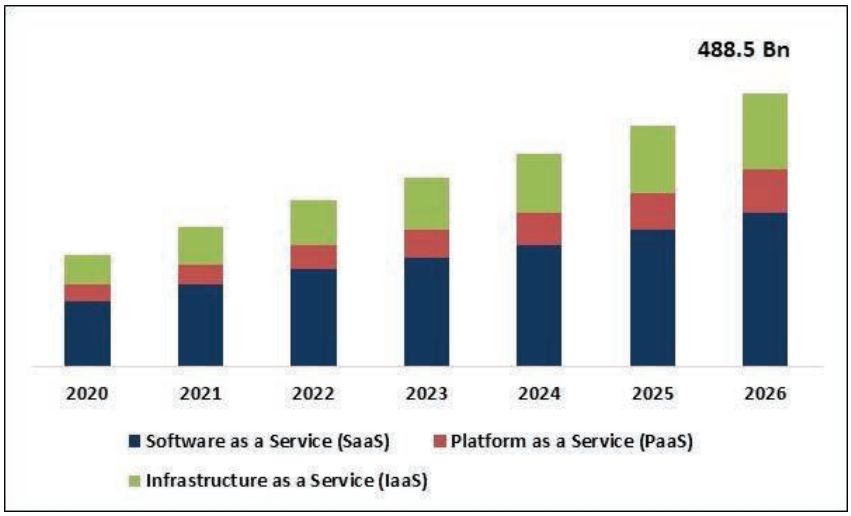
4

Cloud and SaaS and IaaS Market

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES



Source: Statista, Synergy Research Group

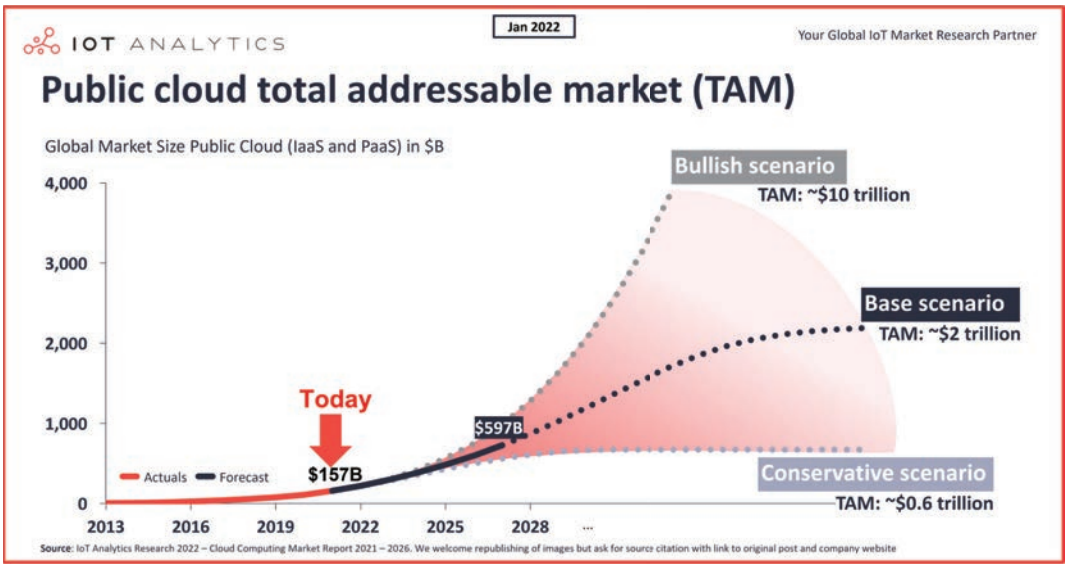


Source: KBV Research (https://www.kbvresearch.com)

5

Public Cloud
Total Addressable Market (TAM)

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

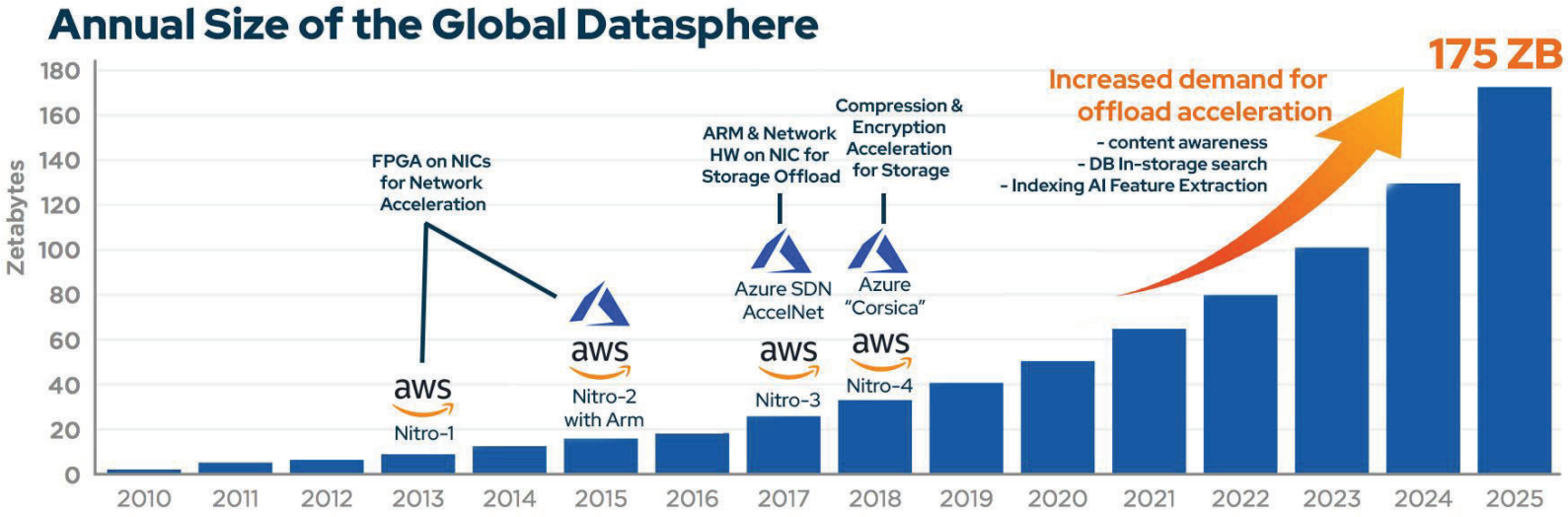


Source: IOT Analytics - https://iot-analytics.com/cloud-market/

6

Evolution of Data Center Architectures

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES



7

Key Features Defining DPU/SmartNIC Technologies

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

- **Managed high-bandwidth low-latency communication**
  Hyperscaler specific communication protocol, network and resource management support
- **Elastic and compute/storage resources management**
  Enabling efficient virtualization
- **Smart disaggregation of computing and storge resources**
  Enabling effective utilisation of data center resources
- **On-device processing of upper-layer functions**
  Enabling semi-autonomous decision-making
- **On-device acceleration**
  Offloading heavy-duty tasks such as encryption, switching, inspection etc.

Common SmartNIC and DPU Architectures

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES



Xilinx ALVEO: Fully logic programmable SmartNICs. Enables the customization of all network and application layer functions to achieve the best performance for a given use-case.



Mellanox Innova II: Semi-programmable SmartNICs. Combines highly efficient standard NIC technology with programmable logic for customization of critical network and application layer functions.



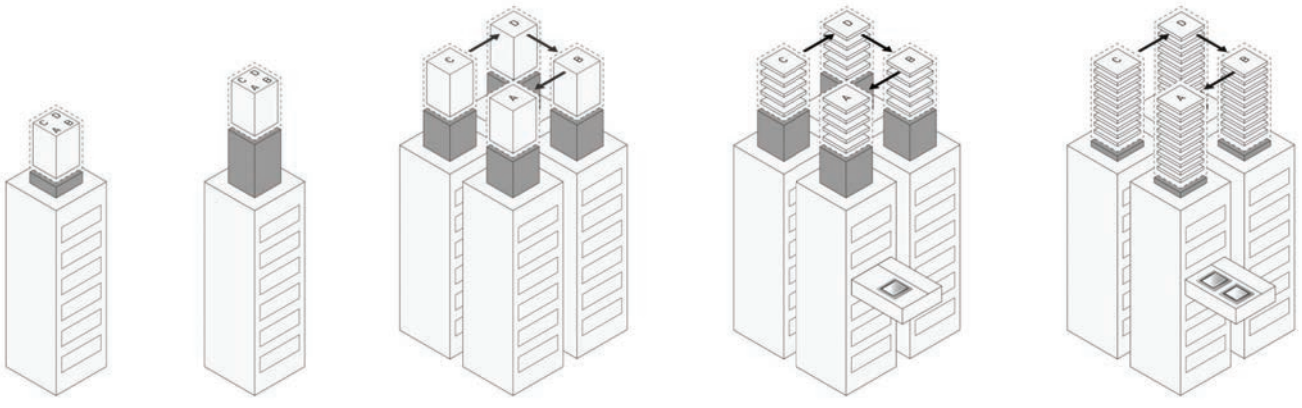
Mellanox BlueField-1: Software programmable SmartNICs. Combines embedded high-performance 64-bit processors (8 to 16 x 64-bit ARM cores) and performance optimized offload accelerators for network and application layer functions.



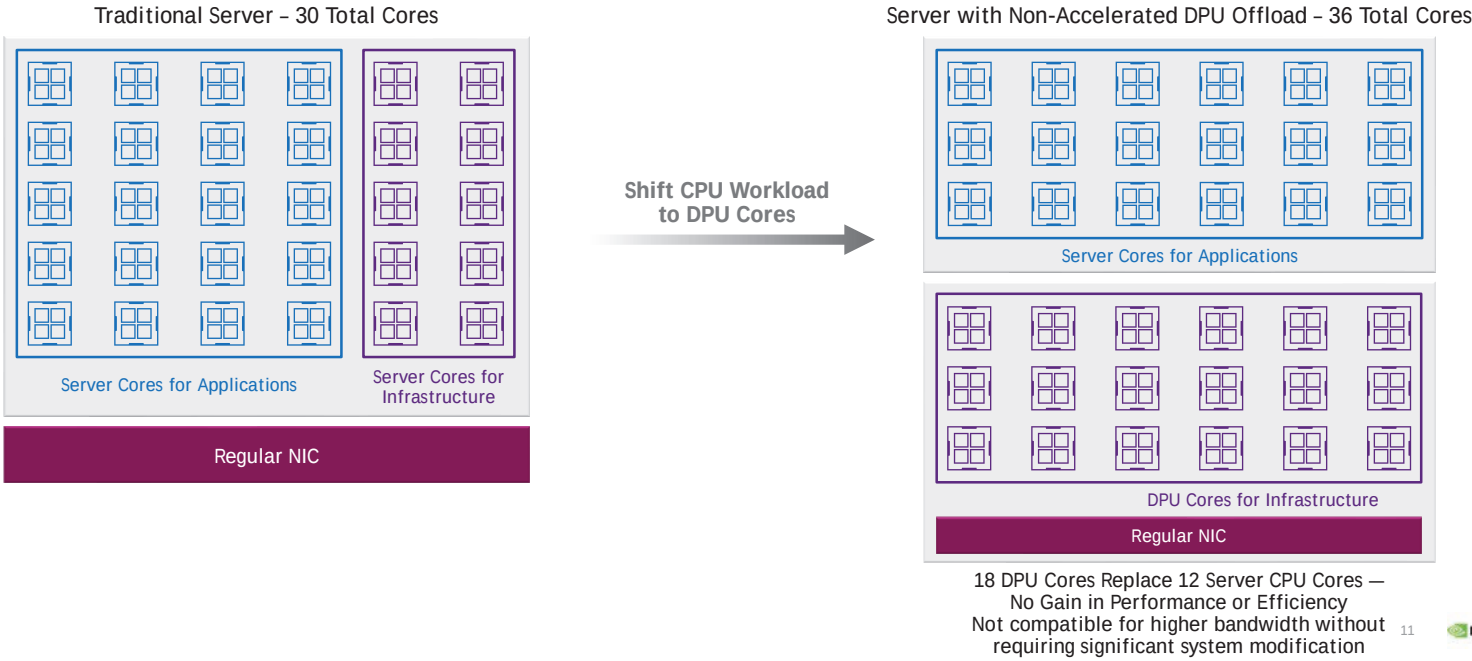
The Era of DPU Emerges - Nvidia DPU (BF3), Intel IPU, AMD (Pensando) DPU, Marvell DPU, Broadcom Stingray, Fungible DPU, …….. Combines embedded high-performance 64-bit processors (16 to 32 x 64-bit ARM cores) and performance optimized parallel processing cores and applications specific accelerators.

9

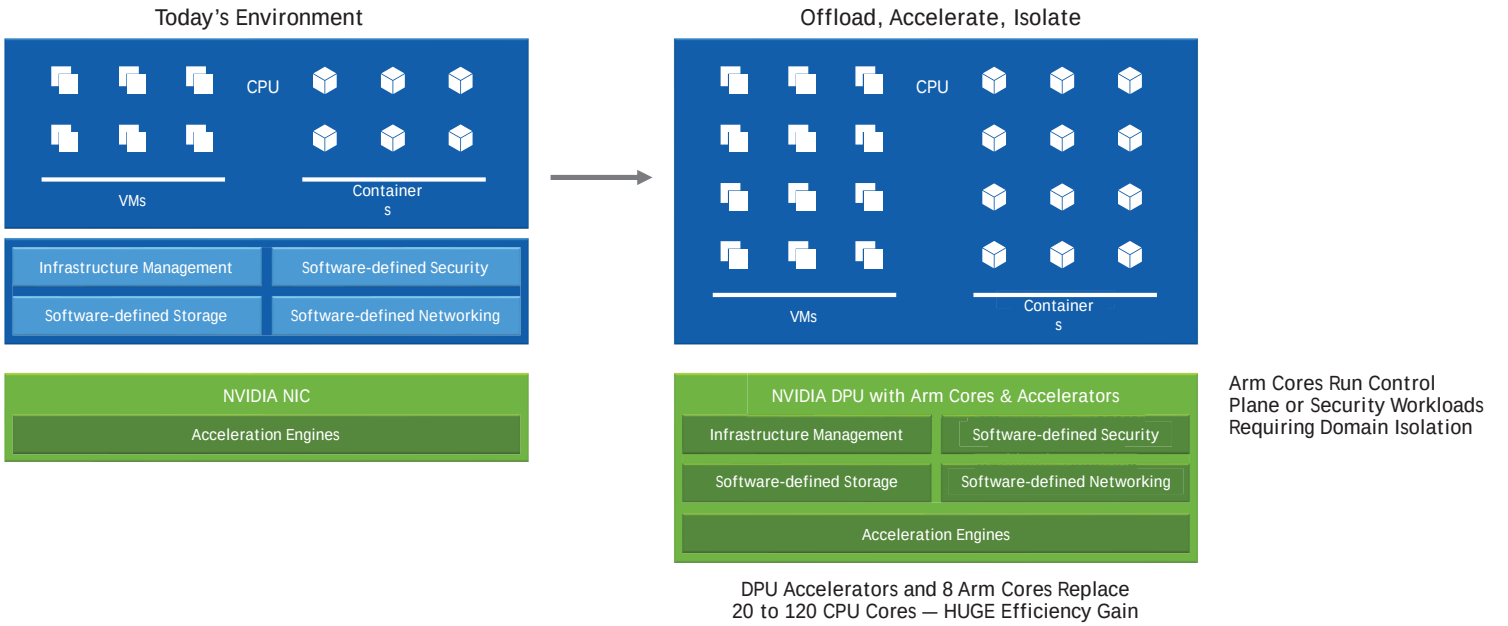
DATA CENTER IS THE NEW UNIT OF COMPUTING



NAIVELY MOVING WORKLOADS TO NIC CPUS DOESN'T WORK



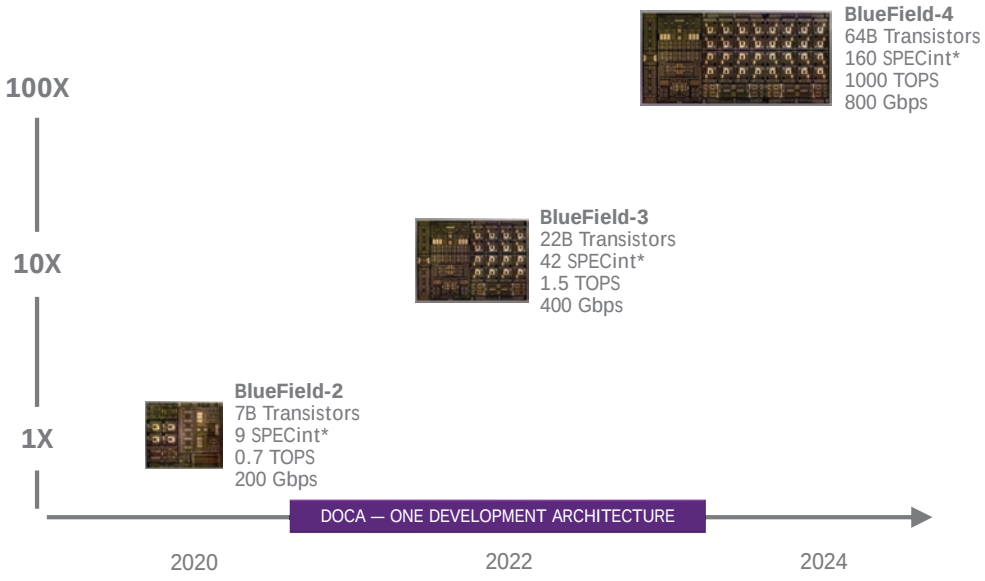
DPU MUST INCLUDE HARDWARE ACCELERATION



12 NVIDIA

NVIDIA DPU ROADMAP

Exponential Growth in Data Center Infrastructure Processing



** SPECint2k17-rate*

13 NVIDIA

NVIDIA BLUEFIELD-3 DPU

First 400Gb/s Data Processing Unit

22 Billion Transistors

400Gb/s Ethernet & InfiniBand Connectivity (1-4 Ports)

PCIe Switch Gen 3/4/5 x32+x4

400Gb/s Crypto / Security Acceleration

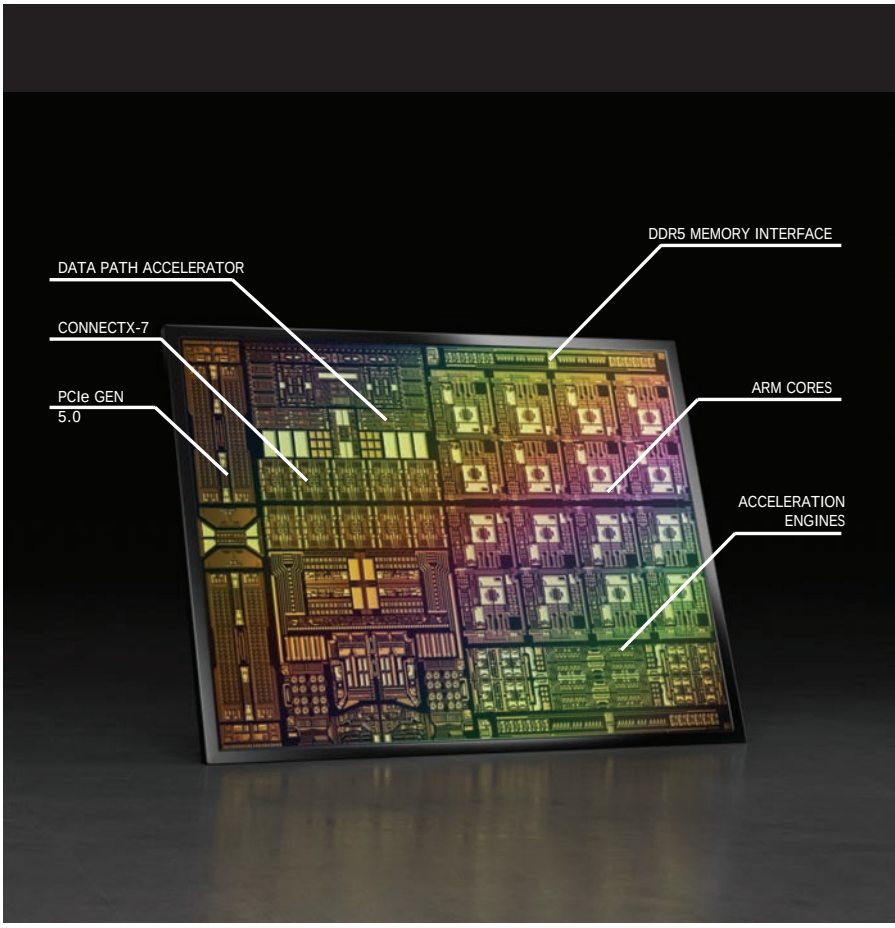
2x370M PPS, 2x40M PPS at scale of millions of flows

18M IOP/s Elastic Block Storage

300 Equivalent x86 Cores

16C A78 ARM 42 SPECINT2k17-rate

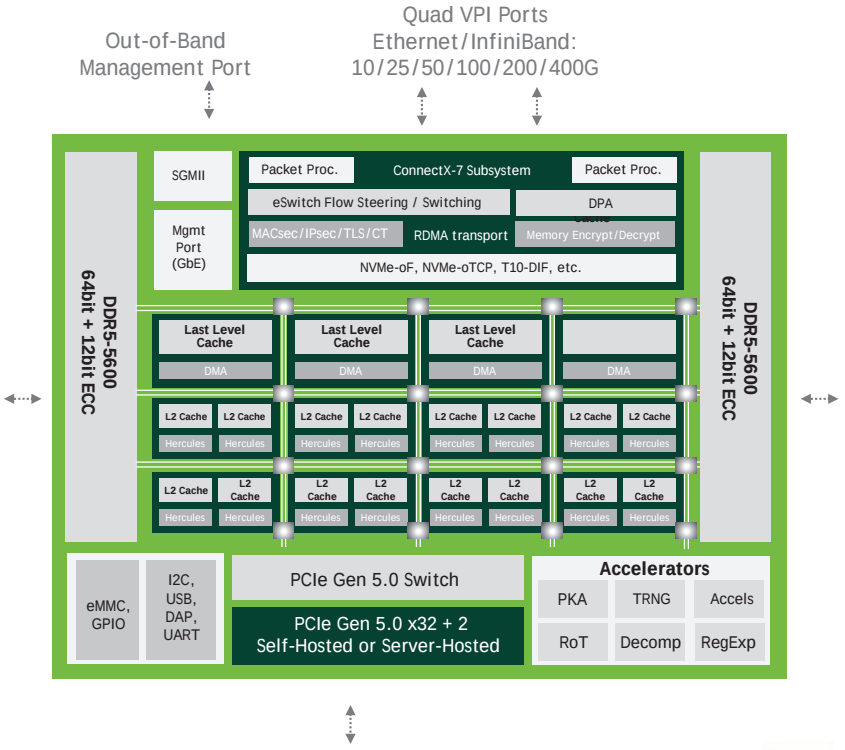
128b DDR5-5600



BLUEFIELD-3 DPU

Technical Overview

- ConnectX-7 inside
- I/O
  - 1x400Gb/s, 4x100Gb/s Ethernet/InfiniBand
  - 100G PAM4 SerDes
  - 400Gb/s total bandwidth
  - Integrated PCIe switch
  - Gen5.0 x32+2
  - Multi-host – 8 hosts
- Compute sub-system
  - 16 Arm®A78 v8.2+ @2.3GHz
  - SkyMesh™ fully coherent low-latency interconnect
  - 8MB L2 Cache, 16MB LLC System Cache
- Datapath Programmable Accelerator (DPA)
- Advanced Memory sub-system
  - Dual DDR5-5600 interfaces (128bit)
  - NVDIMM-N Support
  - DDR memory encryption
- 1GbE Out-Of-Band management port
- Self-hosted or Server-hosted



NVIDIA DOCA

Enabling Broad BlueField Partner Ecosystem

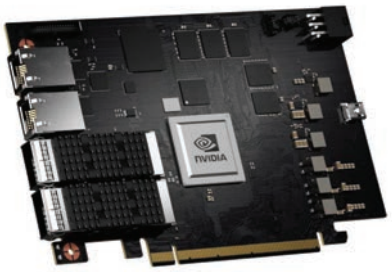
Software Development Framework for BlueField DPUs

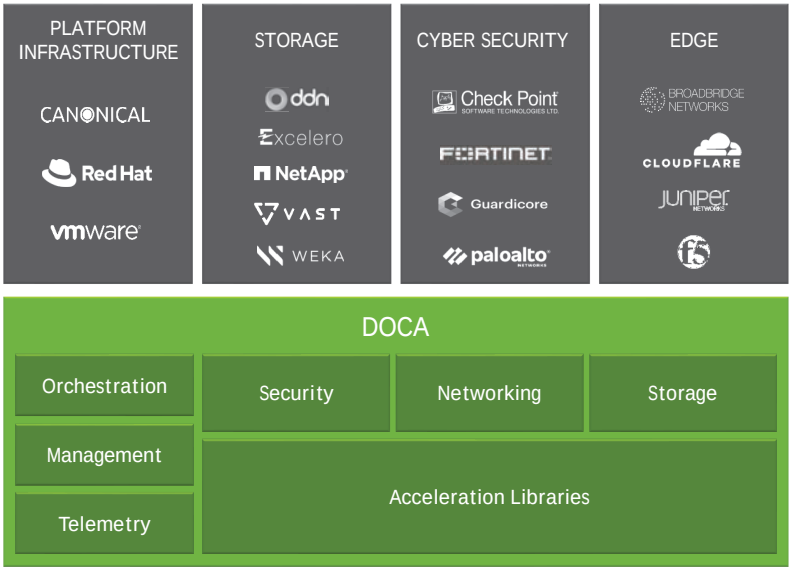
Software Compatibility for Generations of BlueField DPUs

Offload, Accelerate, and Isolate Infrastructure Processing

Support for Hyperscale, Enterprise, Supercomputing and Hyperconverged Infrastructure

DOCA is for DPUs what CUDA is for GPUs

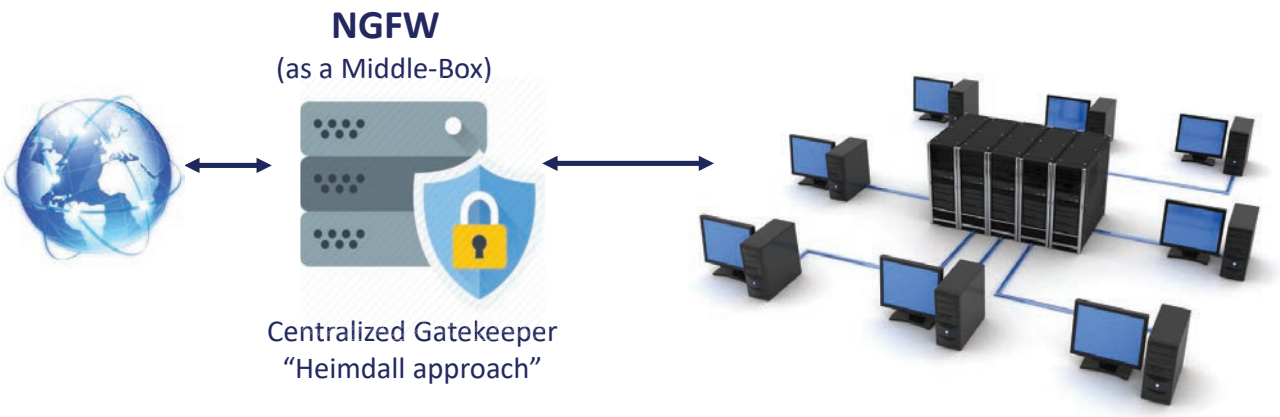




16 NVIDIA

Traditional Network Security

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES



- Centralized, difficult to scale
- Locked to one specific vendor
- Vulnerable to vendor specific DDoS attacks
- Cannot be easily extended into the cloud
- Single point of failure

17

Cloud Enabling Network & App Security

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

Virtual Security Appliance
AWS Instance + AWS Marketplace Apps



Instance + Apps

TREND MICRO

elastic

MySQL

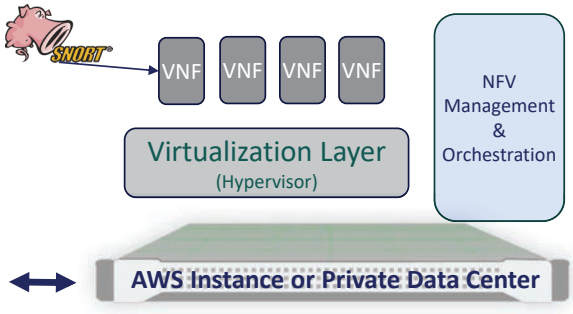
CentOS

SaaS Client
(Enterprise, SME)

Security becomes a heavyweight inefficient software-based virtual appliance



Virtual Security Appliance
Cloud-based Security Middle-box Model

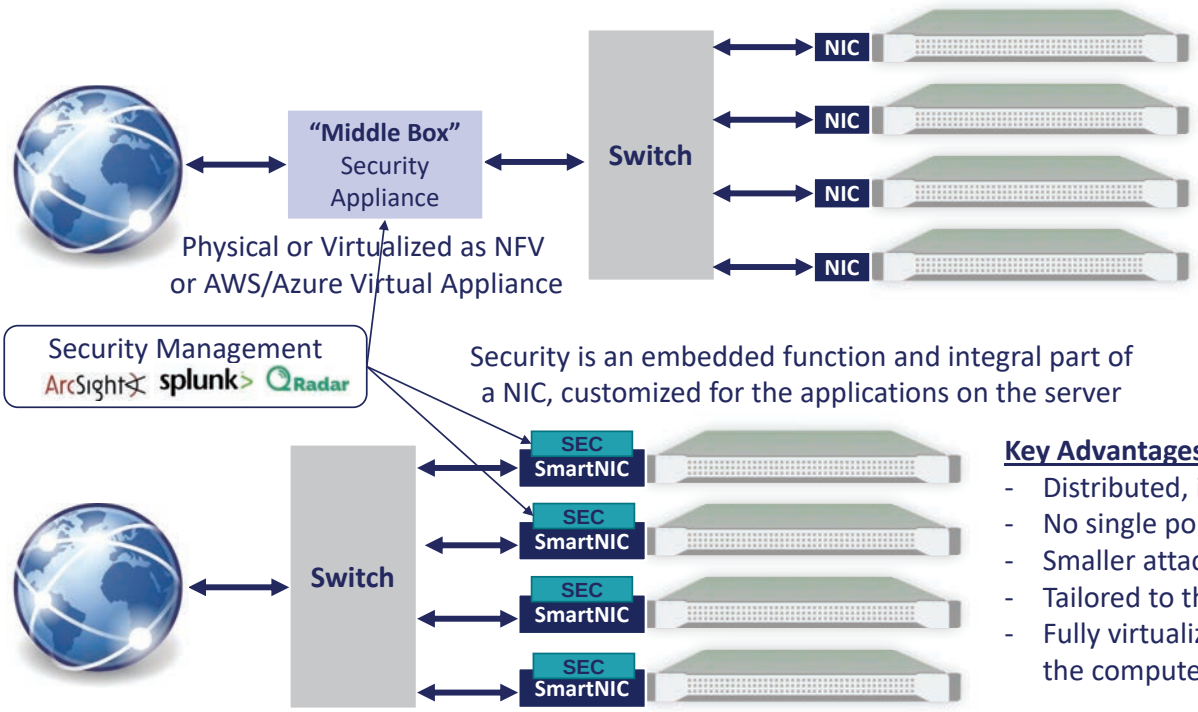


Network Function Virtualization
Virtualized Security Middle-box Model

18

Centralized vs DPU Network Security

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES



Key Advantages

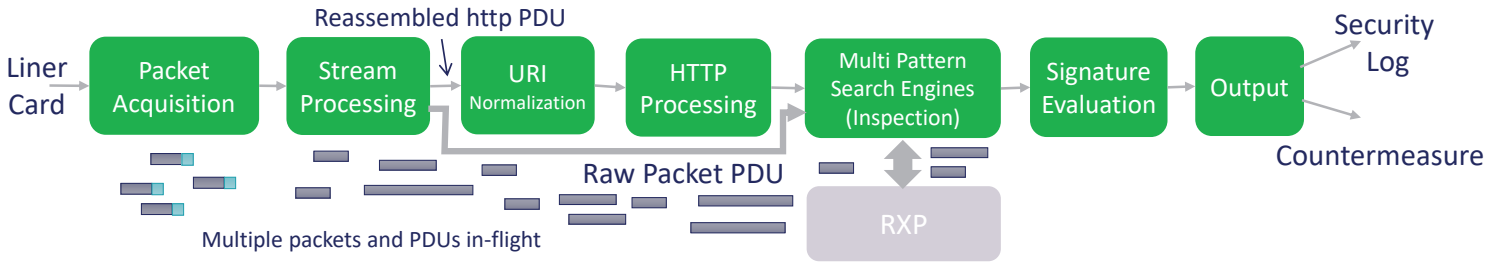
- Distributed, inherently resilient
- No single point of failure
- Smaller attack surface
- Tailored to the application
- Fully virtualizable without the compute overhead (Advanced NFV)

19

Use Case: Snort Network IPS

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

- Snort 3.0: - Open source Network Intrusion Detection/Prevention System
- Optimized for real-time detection and prevention of network centric attacks and issues: - buffer overflows, stealth port scans, semantic URL attacks, CGI attacks, etc.
- Snort 3.0 operation can be subdivided into 7 phases
- http processing is stateful and inspection targets reassembled PDUs
- Multiple Snort instances (on multiple cores) can offload many PDUs in-flight



Use Case: Snort Network IPS

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 1978
(msg:"APP-DETECT Apple OSX Remote Mouse usage";
flow:to_server,established;

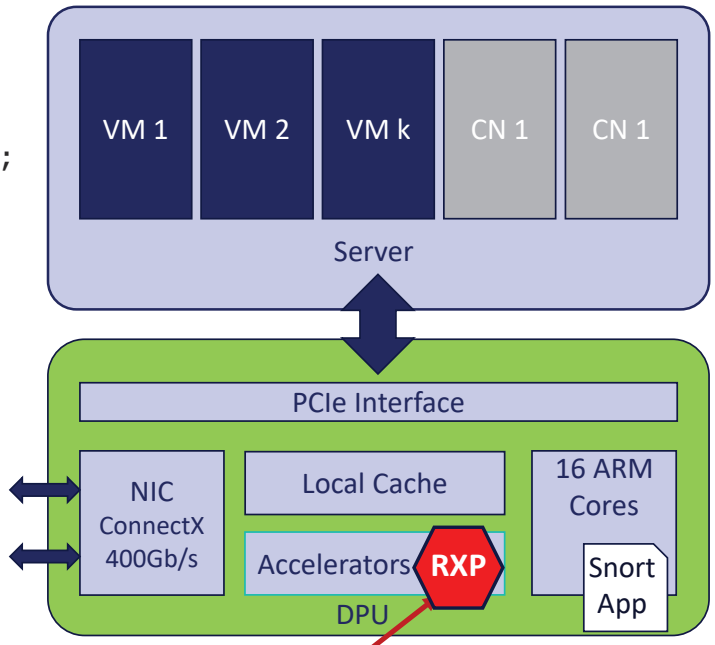
content:"mos "; fast_pattern:only;
pcre:"/mos\s{2}\dm\s\d/";
reference:url,pastebin.com/F81NCiYE;
classtype:policy-violation; sid:20443; rev:2;)
```

Packet Header Rule converted into RegEx

```
/^\xB7\x11\x3A\x5
1\x8F\x75\x45\x53
  .{2}\x07\xBA/
```

Fast Pattern Rules

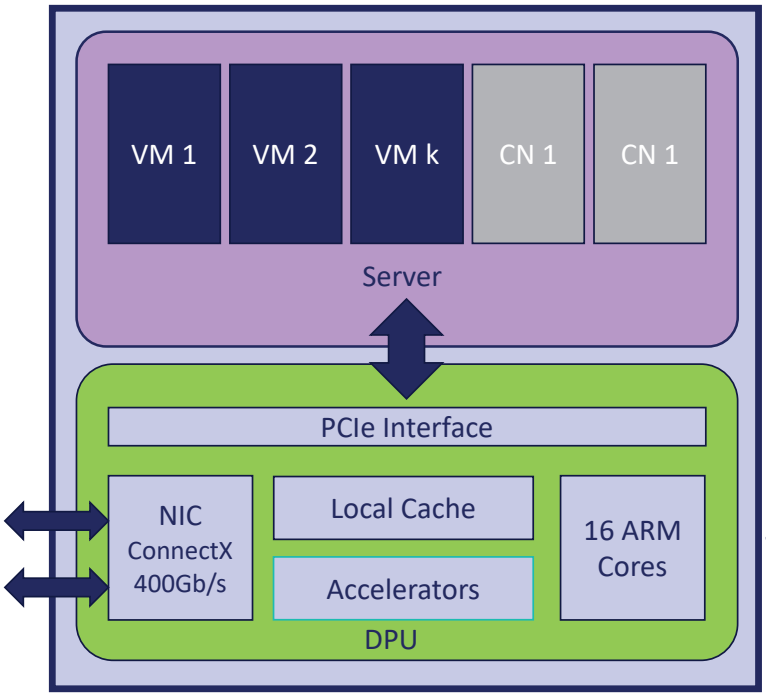
PCRE Rule



NVIDIA Bluefield 3

ML - based Network IPS

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES



Alarms, Warnings, Logs and Countermeasures

ML Inference cost and latency is critical

Threat Classification

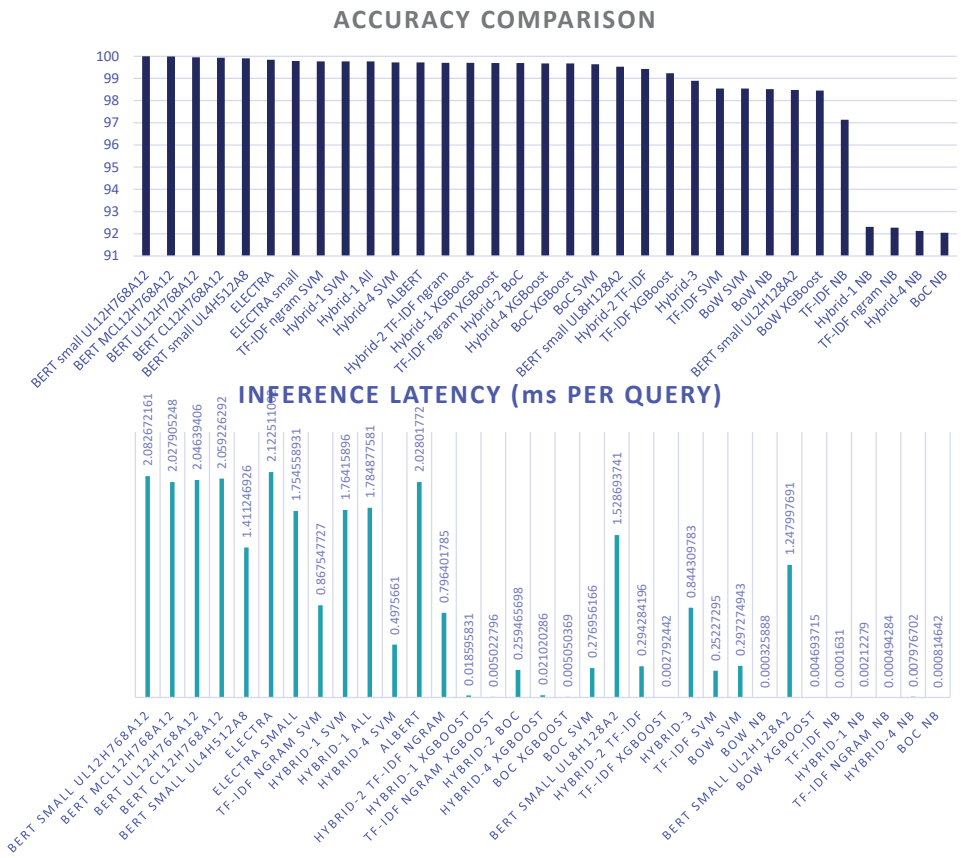
L4, L5, L7 Feature Extraction / Feature Reduction

TLS Proxy (decrypting https/http2/http3)

L1-L3 Feature Extraction (traffic statistics)

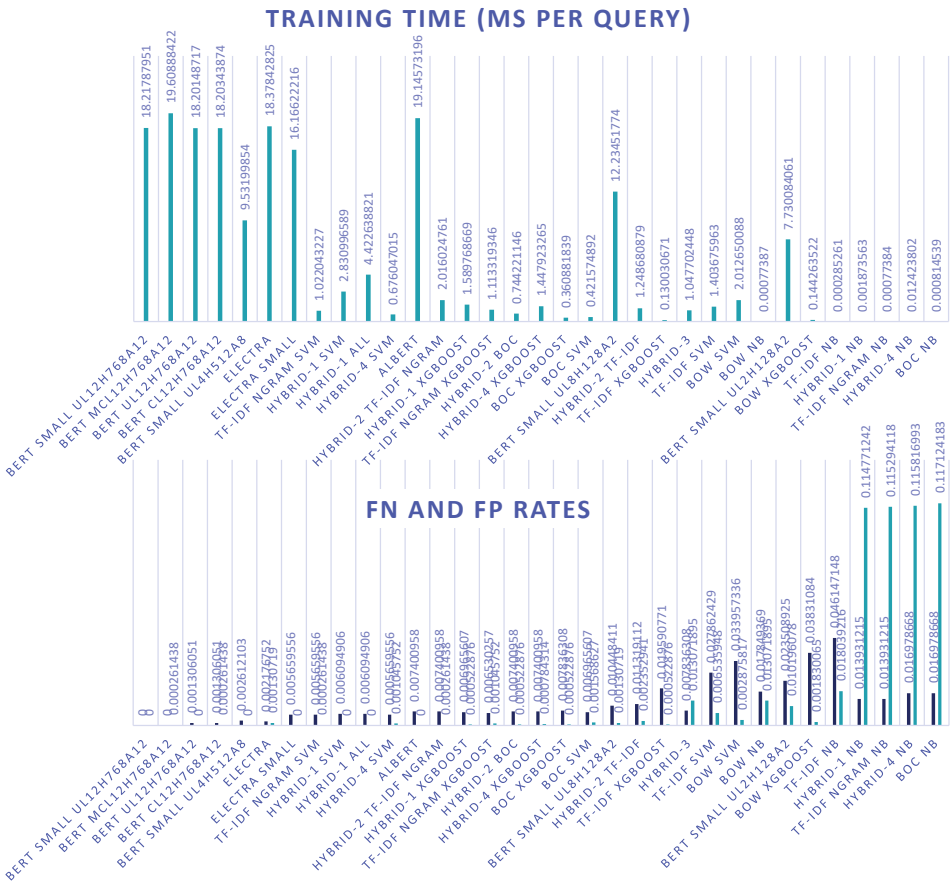
Needed for traditional IPS solutions

22



CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

Preliminary Analysis
Detection Accuracy
Inference Cost



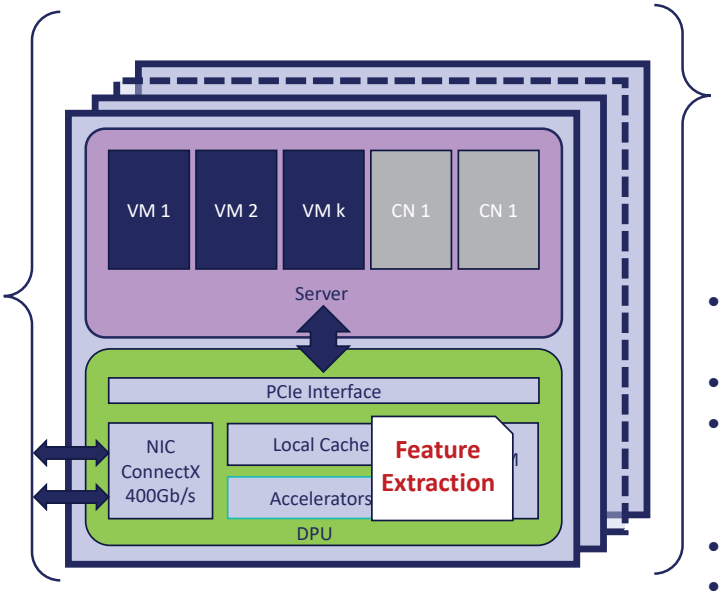
Preliminary Analysis Training time False Negative vs Positive Rates

AI - based Threat Intelligence

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

Datacenter





Nvidia DGX
(Morpheus)
Large AI models for
Threat Prediction

- **Classification of complex behaviors**
  - ➢ **e.g. Reconnaissance**
- **Rec. historic malicious behaviors**
- **Predication of potential threats**
  - **type of threat**
  - **time / execution method**
- **Automated counter response**
- **Visualization (SOC)**
- **AI assisted situation awareness (SOC)**

25

Challenges and Opportunities

CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

Current Challenges:

- Lack of common approach for User Application Integration
- Lack of SmartNIC User Application Orchestration
- Lack of SmartNIC Native Applications
- Lack of standard interface support for efficient offload acceleration

Opportunities :

- SmartNIC "Open Data-Plane" Framework with offload acceleration
- NVIDIA® DOCA™ – Programming framework for DPUs
- VNF based SmartNIC open framework for third-party application
  - OVS, NGFW, IPS, WAF, VPN, vRouter, etc.

Academic Research:

- Defining programming paradigm for DPU Technologies
- Benchmarking and exploring limitations of emerging DPU planforms
- **AI Research** - ***understanding the problem that needs to be resolved and the impact of your AI related research will achieve.***

In Summary

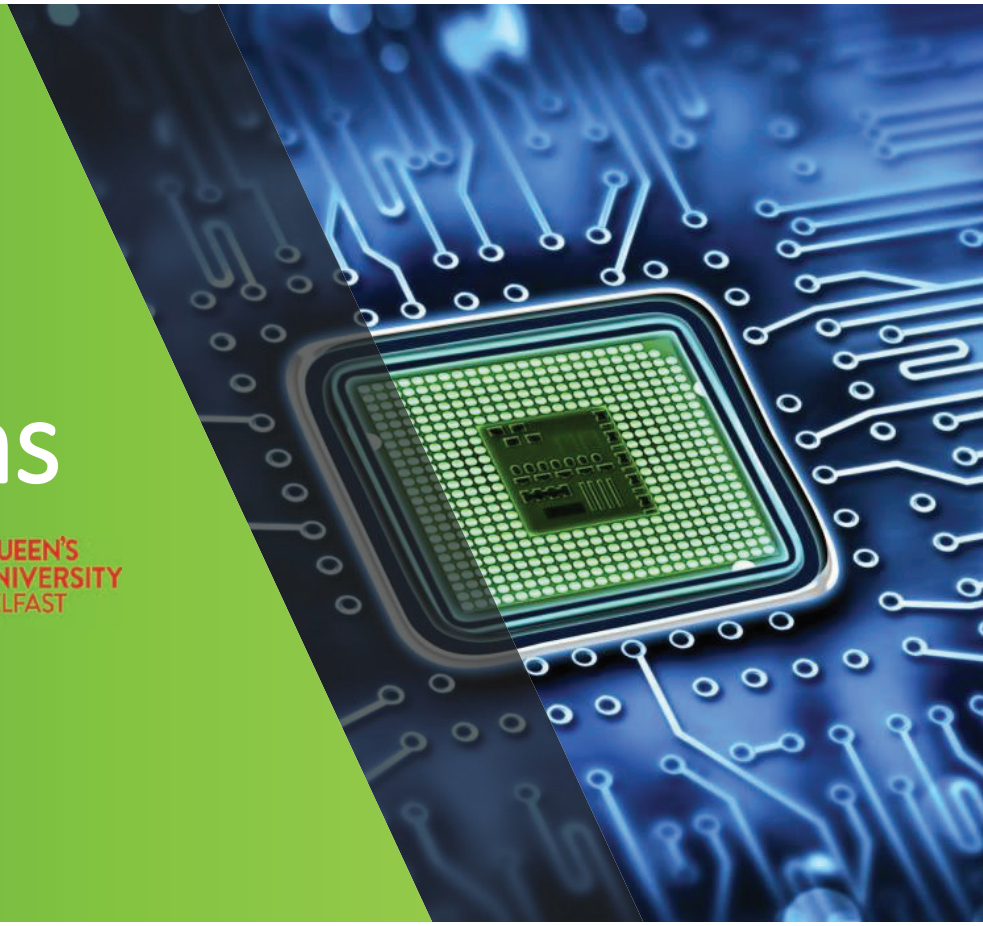
CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES

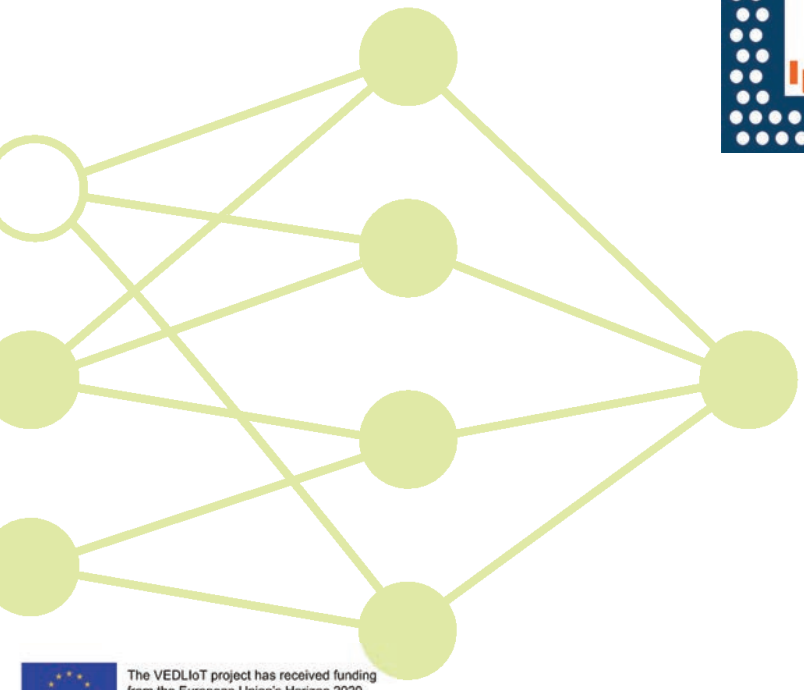
- Evolution of data center server architectures postulates the need for scalable disaggregation and security within the server infrastructure
- The exponential increase of global datasphere and data-centric services require heterogenous computing including:
  - Computational storage
  - Accelerated custom-purpose computing (CPU, GPU / AI-TPU / DPU)
- **Security is an indispensable service** underpinning the fabric of hyperscale data center and is redefined by SmartNIC / DPU technology
- SmartNIC / DPU technologies enable opportunities for emerging critical and unforeseen services
- Further research (**academic and industrial**) is vital for secure and AI accelerated services within future Cloud Platforms

Questions

NVIDIA

QUEEN'S UNIVERSITY BELFAST





The VEDLIoT project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957197

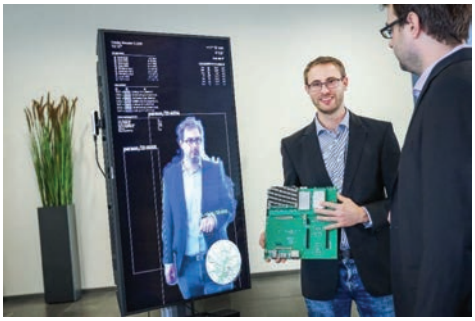
VEDLIoT – Accelerated AIoT

Kevin Mika
Bielefeld University

Very Efficient Deep Learning for IoT – VEDLIoT

VEDLIoT
Very Efficient Deep Learning in IoT

- **Platform**
  - Hardware: Scalable, heterogeneous, distributed
  - Accelerators: Efficiency boost by FPGA and ASIC technology
  - Toolchain: Optimizing Deep Learning for IoT
- **Use cases**
  - Industrial IoT
  - Automotive
  - Smart Home
- **Open call**
  - 10 projects covering a wide range of AIoT applications
  - Early use and evaluation of VEDLIoT technology



- **Call:** H2020-ICT2020-1
- **Topic:** ICT-56-2020 Next Generation Internet of Things
- **Duration:** 1. November 2020 – 31. Oktober 2023
- **Coordinator:** Bielefeld University (Germany)
- **Overall budget:** 7 996 646.25 €
- **Consortium:** 12 partners from 4 EU countries (Germany, Poland, Portugal and Sweden) and one associated country (Switzerland).

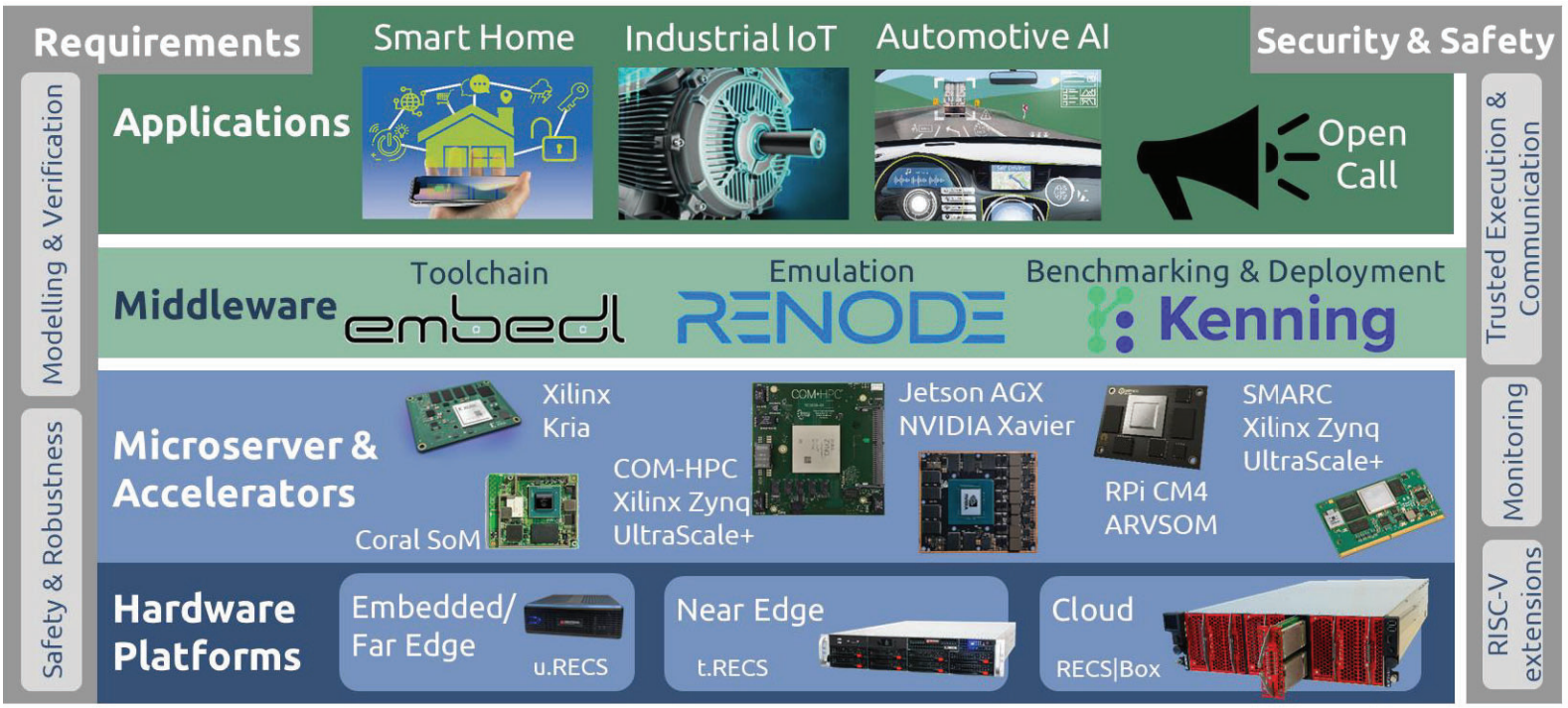
More info:

⇒ https://www.vedliot.eu/
⇒ https://twitter.com/VEDLIoT
⇒ https://www.linkedin.com/company/vedliot/

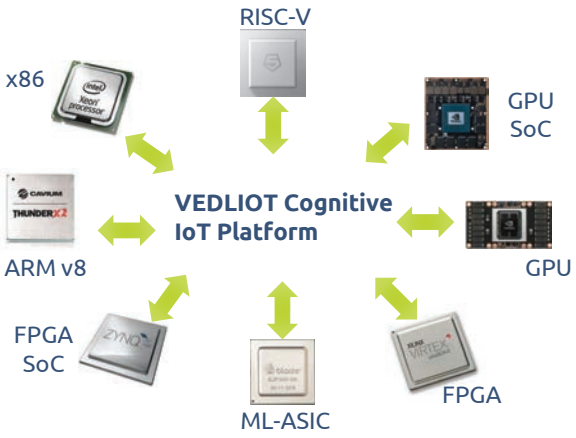
2

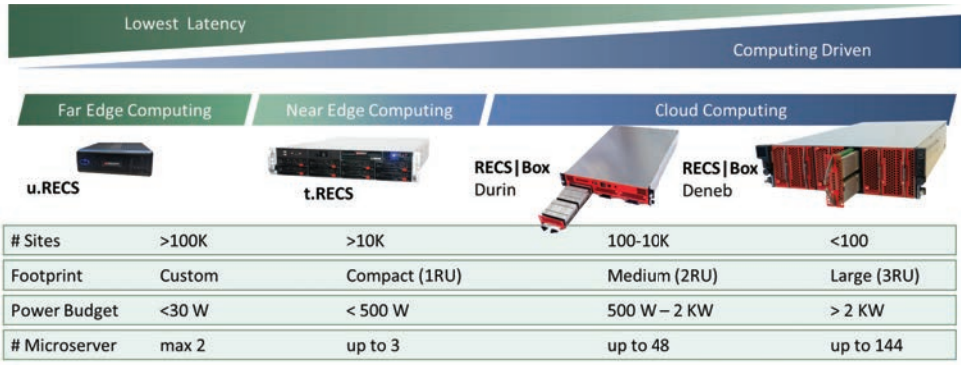
Big Picture

VEDLIoT
Very Efficient Deep Learning in IoT



VEDLIoT Hardware Platform





# Sites	>100K	>10K	100-10K	<100
Footprint	Custom	Compact (1RU)	Medium (2RU)	Large (3RU)
Power Budget	<30 W	< 500 W	500 W – 2 KW	> 2 KW
# Microserver	max 2	up to 3	up to 48	up to 144

- Heterogeneous, modular, scalable microserver system
- Supporting the full spectrum of IoT **from embedded over the edge towards the cloud**
- Different technology concepts for improving
  - Performance
  - Cost-effectiveness
  - Maintainability
  - Reliability
  - Energy-Efficiency
  - Safety

4

RECS Architecture – RECS|BOX



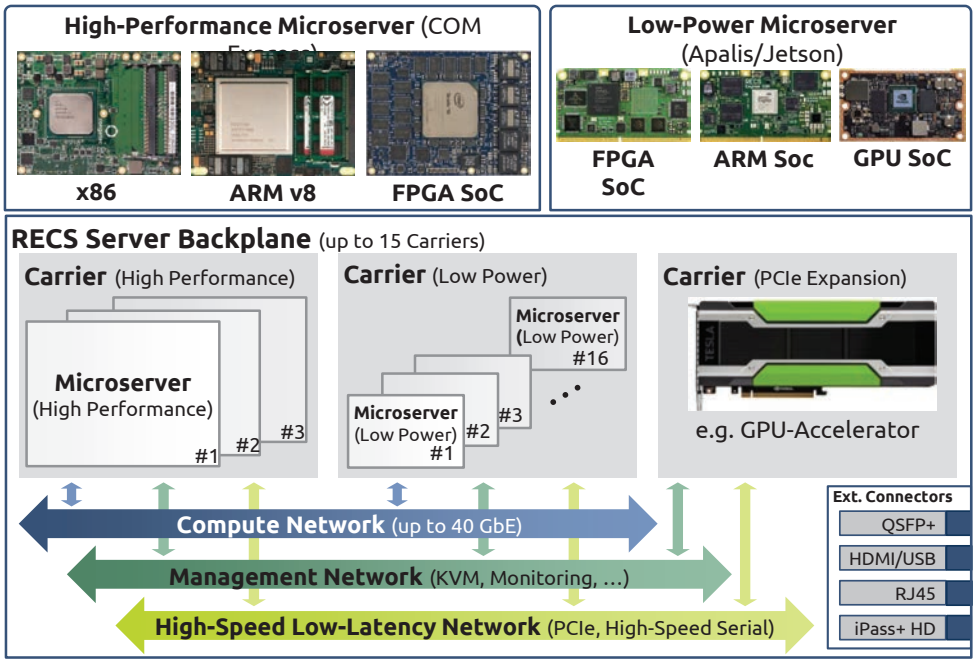
VEDLIoT
Very Efficient Deep Learning in IoT

High-Performance Carrier
(up to 3 microservers)



Low-Power Carrier
(up to 16 microservers)





5

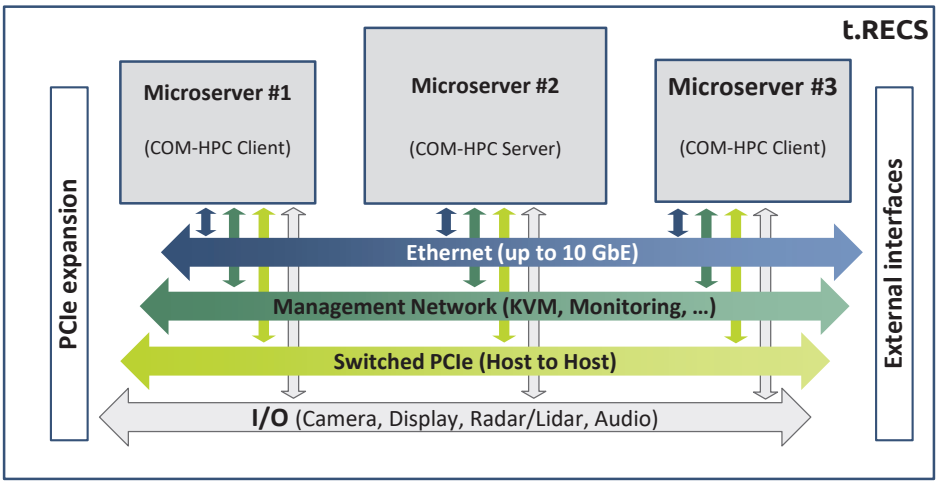
RECS Architecture – t.RECS

VEDLIoT
Very Efficient Deep Learning in IoT

t.RECS Edge Server

- Optimized platform for local / edge applications
- Provide interfaces for
  - Video
  - Camera
  - Peripheral input (USB)
- Combine FPGA and GPU acceleration
- Compact dimensions
  1 RU, E-ATX form factor
  (2 RU/ 3 RU for special cases)





6

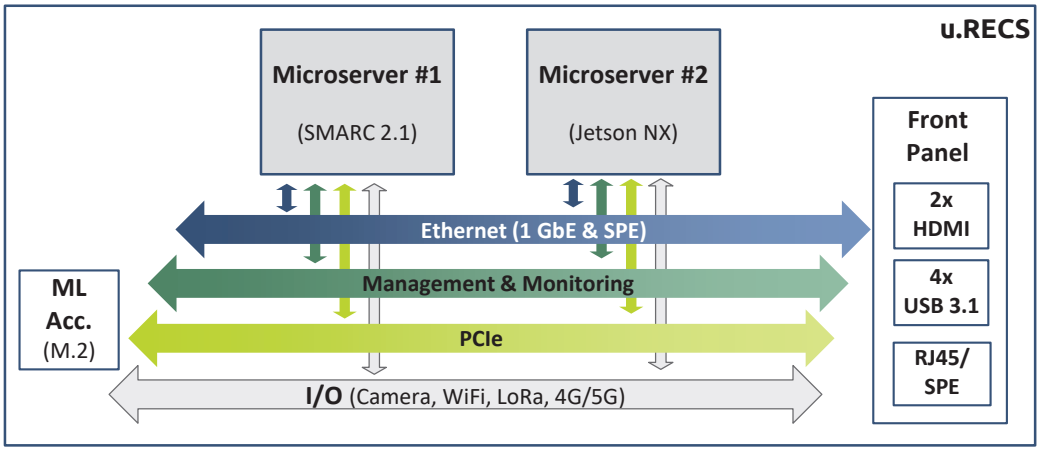
RECS Architecture – u.RECS

VEDLIoT
Very Efficient Deep Learning in IoT

u.RECS AIoT Server

- Supports ML acceleration
  - FPGA
  - ASIC
- Communication interfaces
  - Wired (CAN, Ethernet, CSI)
  - Wireless (WLAN, LoRa, 5G)
- Sensors
  - Camera
  - Environment (Temp./Hum.)
  - Housekeeping
- Embedded Device
  (~ 20x20x6 cm)

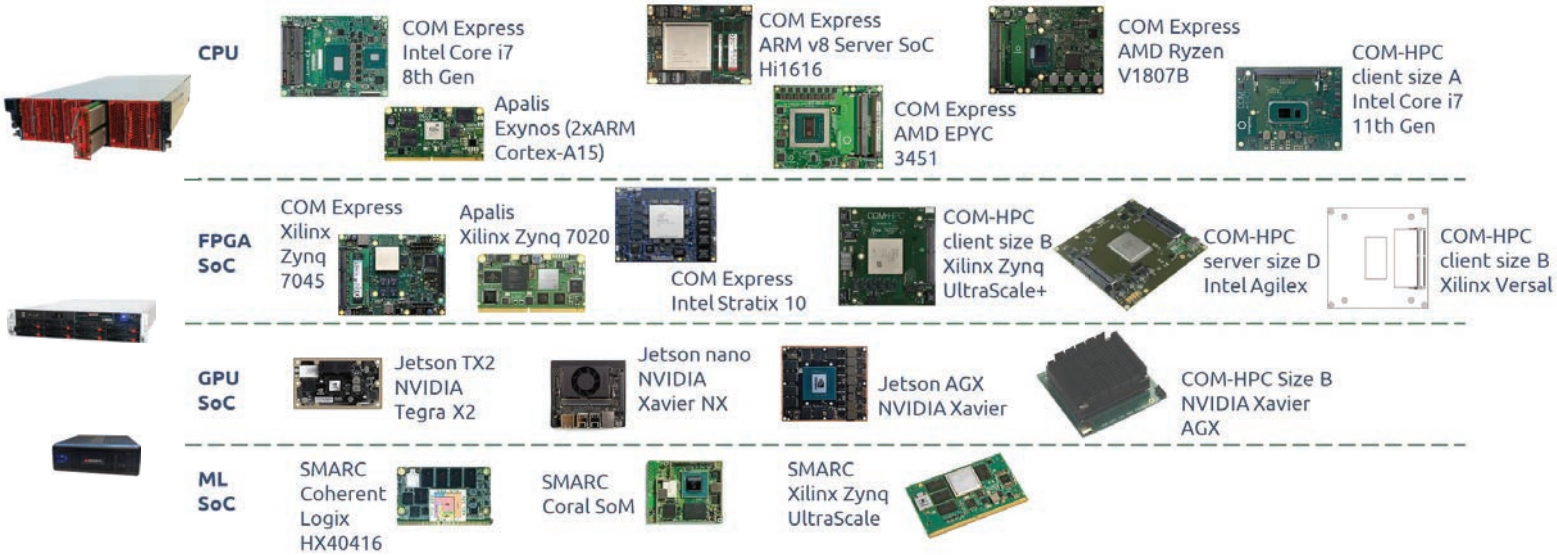




7

Microserver overview

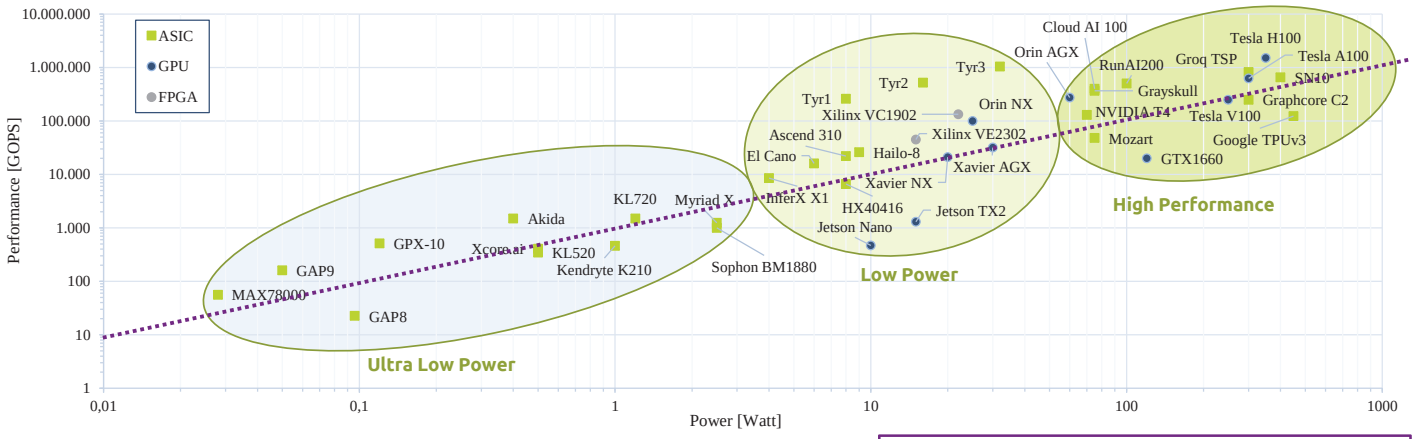
VEDLIoT
Very Efficient Deep Learning in IoT



Peak Performance of DL Accelerators

VEDLIoT
Very Efficient Deep Learning in IoT

- Peak performance values of specialized accelerators, provided by the vendors (precisions varying from INT8 to FP32)

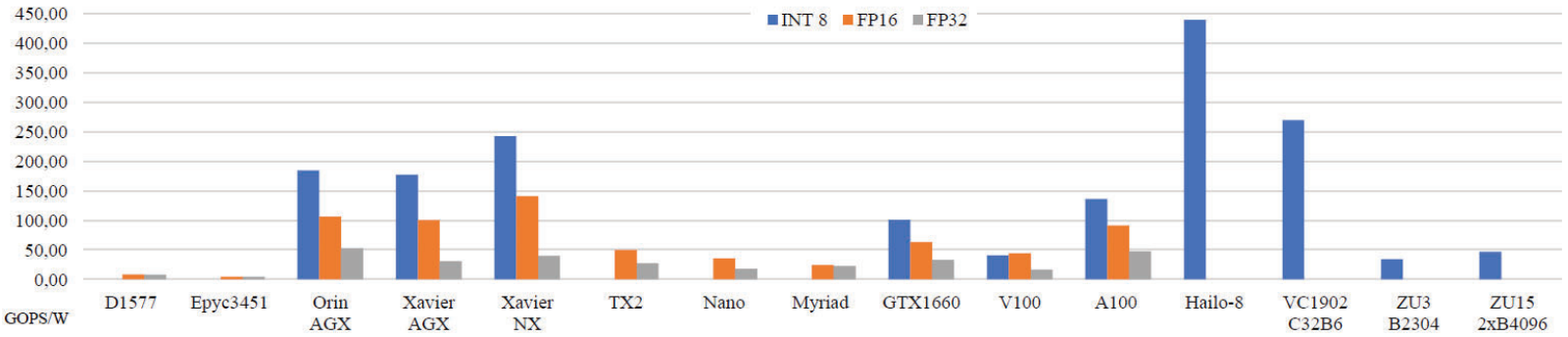


Average efficiency at 1000 GOPS /W

Yolo v4 accelerator performance

VEDLIoT
Very Efficient Deep Learning in IoT

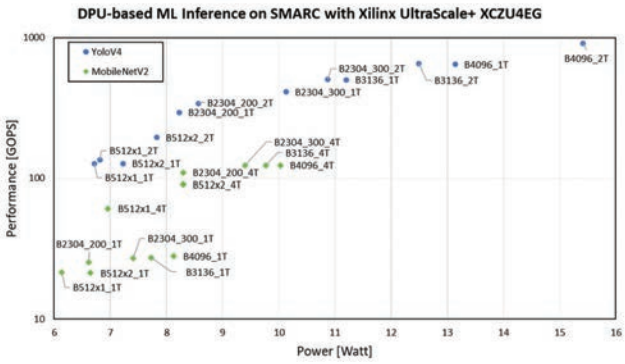
- Performance of Yolo v4 for different hardware platform has been evaluated
- Performance measurement for other networks (Resnet, EfficientNet) available as well

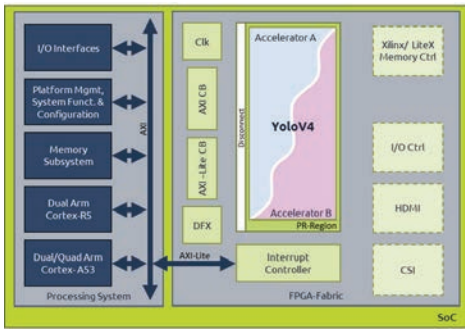


Reconfigurable DL accelerators

VEDLIoT
Very Efficient Deep Learning in IoT

- VEDLIoT accelerators support a large variety of reconfigurable architectures
  - From small embedded FPGAs to large ACAPs
- Large design space for FPGA-based accelerators
- Dynamic hardware reconfiguration
  - Adapt to changing requirements at run-time
  - Change characteristics of DL-accelerator
  - Trade-off between power and performance, power and accuracy, etc.
- Inference and training on FPGA
  - Supports quantization from int8 to float32
  - DL and Deep Reinforcement Learning





DL accelerator co-design

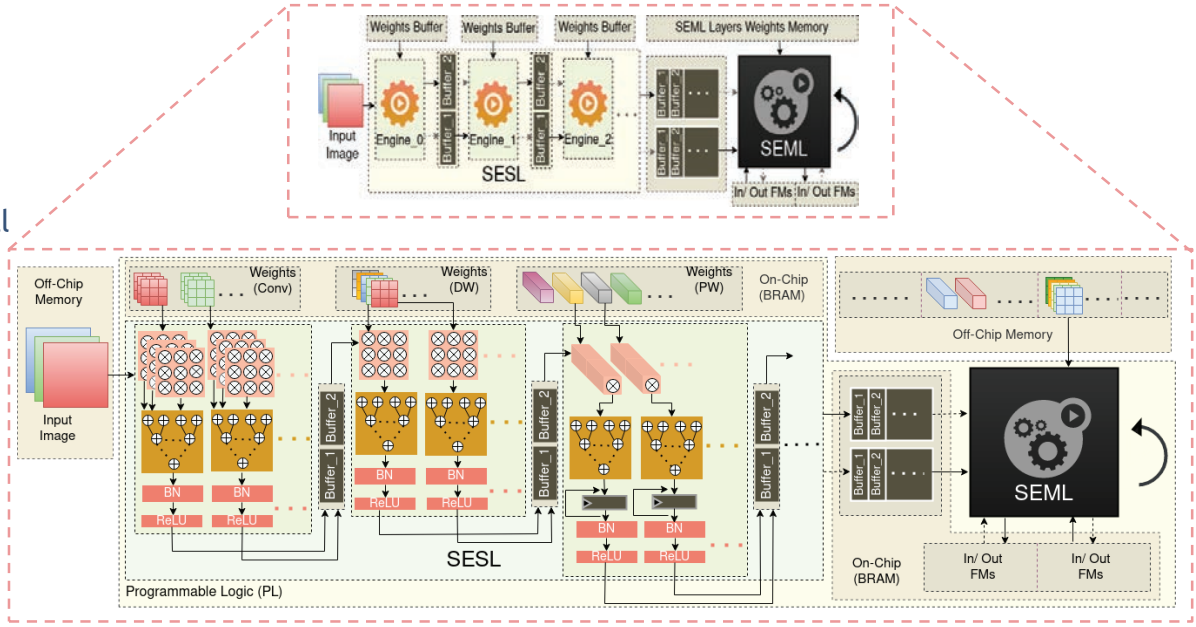
VEDLIoT
Very Efficient Deep Learning in IoT

Monolithic design
- One engine computes all the core layers
- E.g. TPU

SEML
- One engine computes all layers of the same type
- PW engine, DW engine

SESL
- One engine per layer
- E.g. FINN

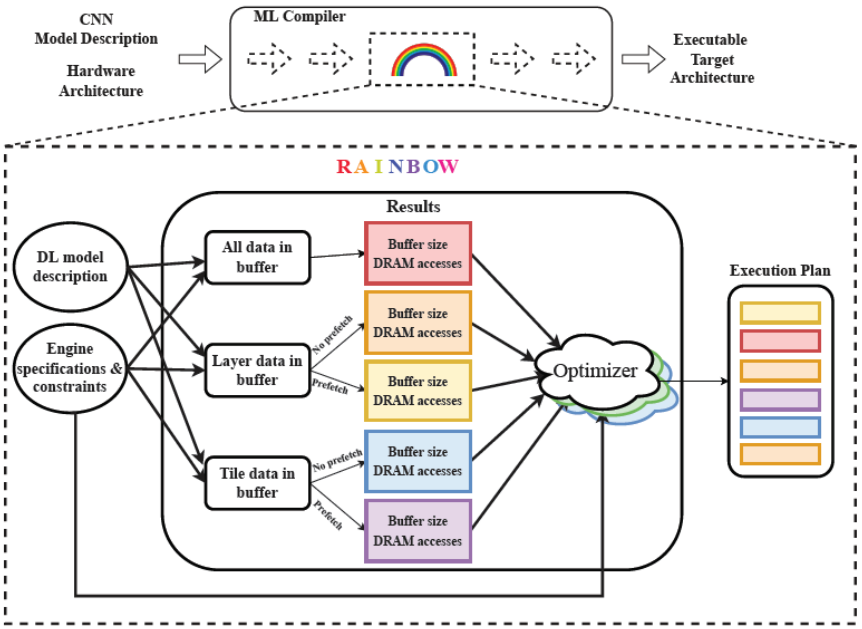
FiBHA
- SESL + SEML



"FiBHA: Fixed Budget Hybrid CNN Accelerator", Fareed Qararyah, Muhammad Waqar Azhar, Pedro Trancoso, IEEE 34th International Symposium on Computer Architecture and High-Performance Computing (SBAC-PAD 2022), Bordeaux, France, November 2–5 2022

Memory management for DL accelerators

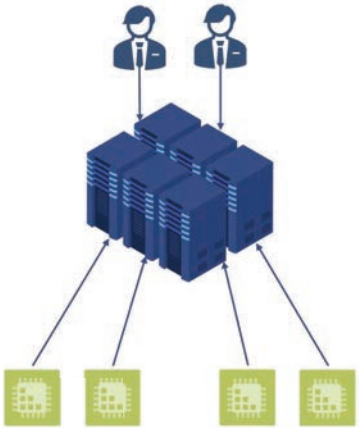
- RAINBOW tool
  - Different types of memory buffer strategies
  - Different types of optimizers
  - Layer-by-layer on chip memory analysis
  - Requirements determine best heterogeneous execution plan (combination of different strategies)

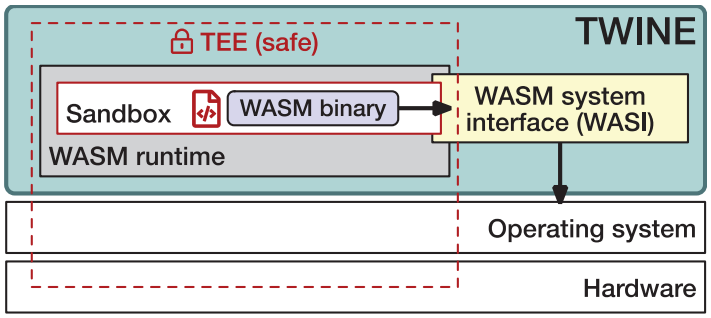


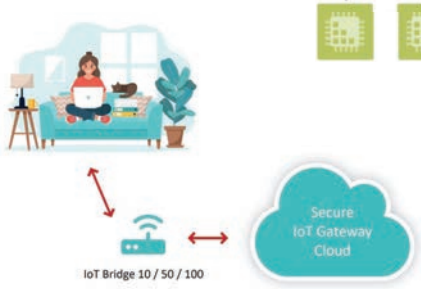
Security

VEDLIoT
Very Efficient Deep Learning in IoT

- Common environment for running distributed applications
  - WebAssembly runtime + Trusted Execution Environment
  - Security for edge (and cloud) devices
- Advances on attestation
  - Better support for edge devices
  - Distributed (Byzantine fault-tolerant) attestation and configuration service
- Secure IoT Gateway







Safety and Robustness

VEDLIoT
Very Efficient Deep Learning in IoT

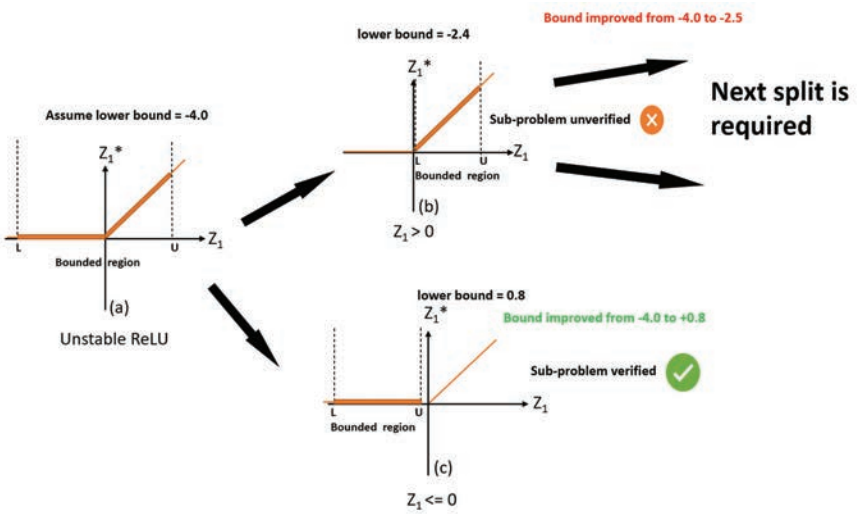
RENODE™

Simulation platform for ML accelerators

- RISC-V SoCs and Custom Function Units
- Improve test and verification
- Co-simulate Verilog blocks
- Used in Google's CFU Playground
- Continuous integration based in Gitlab and Google Cloud Platform

Robustness verification on DL models

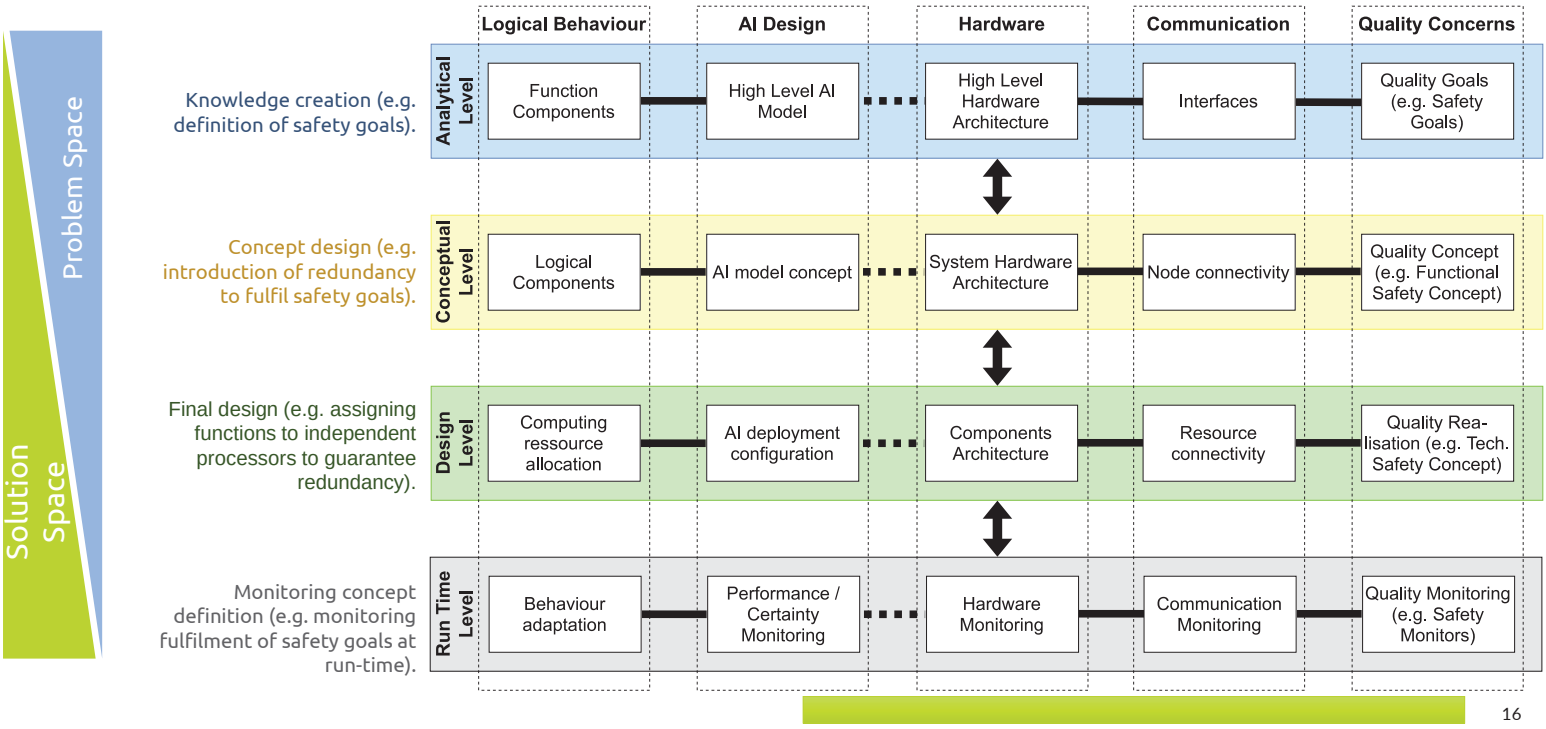
- Tuning hyperparameters



15

A compositional architecture framework for AIoT

VEDLIoT
Very Efficient Deep Learning in IoT



Use case: Automotive

VEDLIoT
Very Efficient Deep Learning in IoT









- Focus on collision detection/avoidance scenario
- Improve performance/cost ratio – AI processing hardware distributed over the entire chain

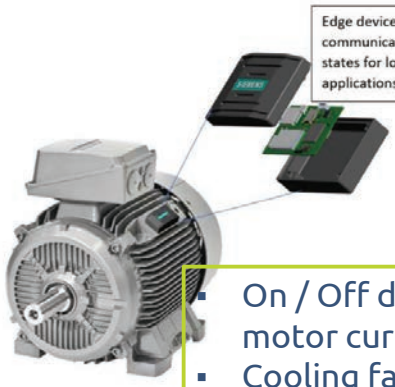
Challenge: Distribution of work

Use case: Industrial IoT – drive condition classification

VEDLIoT
Very Efficient Deep Learning in IoT

- Control applications need DL-based condition classification
  - On the edge device for low power consumption
  - Suggestions for control and maintenance
- DL methods on all communication layers
  - DL in a distributed architecture
  - Dynamically configured systems
- Sensored testbench with 2 motors
  - Acceleration, Magnetic field, Temperature, IR-Cam (temperature), Current-Sensors, Torque

Challenge: Low-power / Efficiency

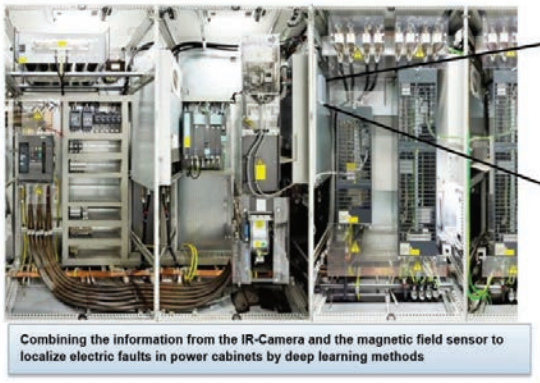


- On / Off detection without motor current or voltage
- Cooling fault detection
- Bearing fault detection

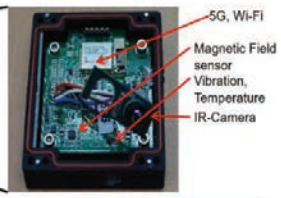
Use case: Industrial IoT – Arc detection

VEDLIoT
Very Efficient Deep Learning in IoT

- AI based pattern recognition for different local sensor data
  - current, magnetic field, vibration, temperature, low resolution infrared picture
- Safety critical nature
  - response time should be <10ms
  - AI based or AI supported decision made by the sensor node itself or by a local part of the sensor network



Combining the information from the IR-Camera and the magnetic field sensor to localize electric faults in power cabinets by deep learning methods



Specifications:
- Industrial temperature range (-20°C ... +85°C)
- Industrial batteries (rechargeable for ID-Tag)
- IP65 protection
- RoHS and IEC 61850-3 complaint
- Pre-certified wireless transceivers
- Target price: 100€ (ID-Tag)
- SIM on Chip*

Challenge: Accuracy

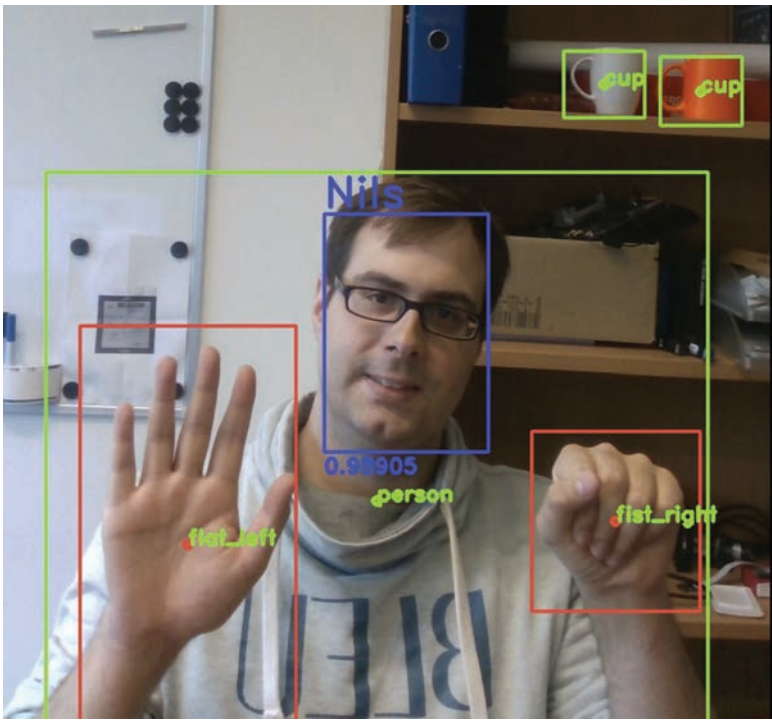
Use case: Smart Mirror – Neural Networks

VEDLIoT
Very Efficient Deep Learning in IoT

- Face recognition
  - Mobilenet SSD trained on WIDERFACE dataset
- Object detection
  - YoloV3, Efficient-Net, yoloV4-tiny
- Gesture detection
  - YoloV4-tiny with 3 Yolo layers (usually: 2 layers)
- Speech recognition
  - Mozilla DeepSpeech

- AI Art: Style-Gan trained on works of arts
- Collect usage data in situation memory

Challenge: Data privacy, Efficiency



20

VEDLIoT
Very Efficient Deep Learning in IoT



Thank you for your attention.

The VEDLIoT project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957197

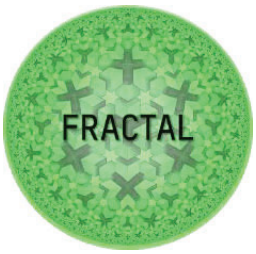
UNIVERSITÄT BIELEFELD · embedl · CHALMERS · SIEMENS · CHRISTMANN · unine · veoneer · UNIVERSITY OF GOTHENBURG · RI.SE · Ciências ULisboa · FCiências · antmicro

Contact
Jens Hagemeyer, Carola Haumann
Bielefeld University, Germany
chaumann@cor-lab.uni-bielefeld.de
jhagemey@cit-ec.uni-bielefeld.de

FRACTAL

A Cognitive Fractal and Secure EDGE based on an unique Open-Safe-Reliable-Low Power Hardware Platform Node

Aizea Lojo (IKERLAN)
Juan Manuel Besga (IKERLAN)
Iñaki Paz (LKS Next)
Gianluca Brilli (Università degli studi di Modena e Reggio Emilia)
Paolo Burgio (Università degli studi di Modena e Reggio Emilia)



CPS&IoT'2023 – Summer School on Cyber-Physical Systems and Internet-of-Things
Budva, Montenegro, June 6, 2023

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056

Co-funded by the Horizon 2020 Programme of the European Union under grant agreement No 877056.

www.fractal-project.eu



Index

1. Project overview
2. FRACTAL System
3. FRACTAL Features & Building Process
4. Use Case - Intelligent Totem

© 2023 FRACTAL. All rights reserved



PROJECT OVERVIEW



FRACTAL GIST

The OBJECTIVE of FRACTAL project is to create a COMPUTING NODE as the building block of scalable Internet of Things

The two main general characteristics of our node would be: COGNITIVE + FRACTALITY

© 2023 FRACTAL. All rights reserved

FRACTAL GIST

FRACTAL **NODE** has to be **COGNITIVE**



adaptive to external changes

interactive to other nodes

© 2023 FRACTAL. All rights reserved



FRACTAL GIST

OUR NODE has to be **FRACTAL**

FRACTALITY is when an apparently irregular structure
is repeated in multiple and different scales of the same structure

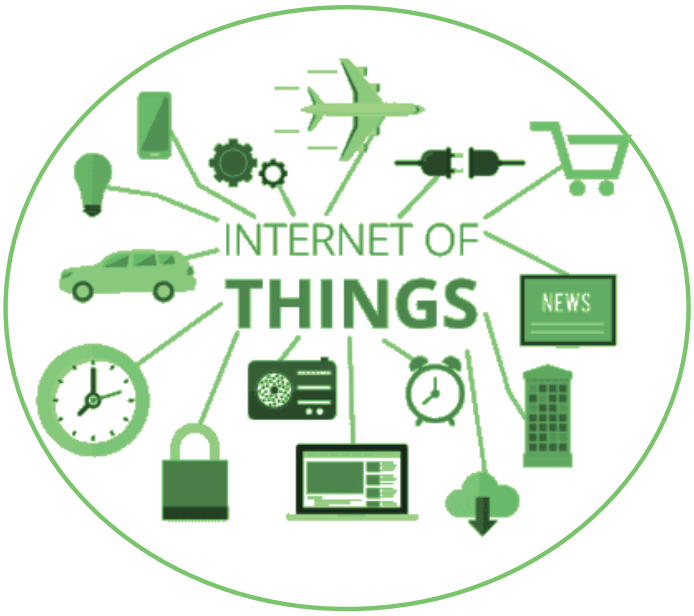


- Is a journey from chaos to order
- Is a connection between the small world and the big world

© 2023 FRACTAL. All rights reserved

FRACTAL GIST

FRACTALITY in the IoT



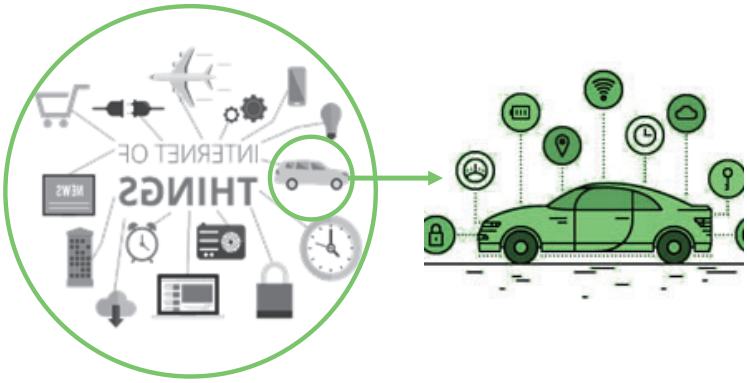
The **IoT** as a whole:

- Is learning from its context
- Is adapting to the changes
- Is dynamically connected
- Is processing data
- …

© 2023 FRACTAL. All rights reserved

FRACTAL GIST

FRACTALITY in the IoT



The **CAR**:

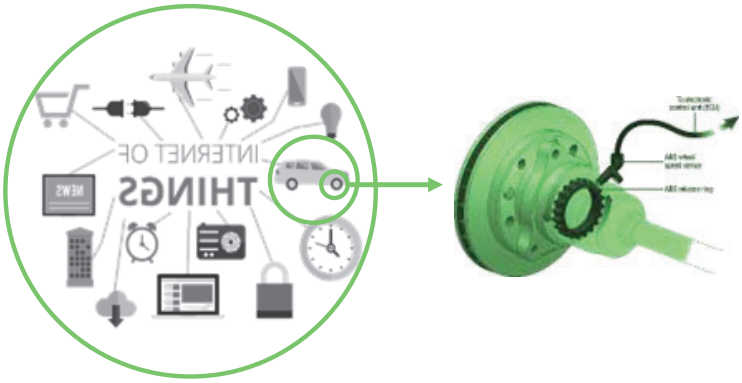
- Is learning from its context
- Is adapting to the changes
- Is dynamically connected
- Is processing data
- …

© 2023 FRACTAL. All rights reserved



FRACTAL GIST

FRACTALITY in the IoT

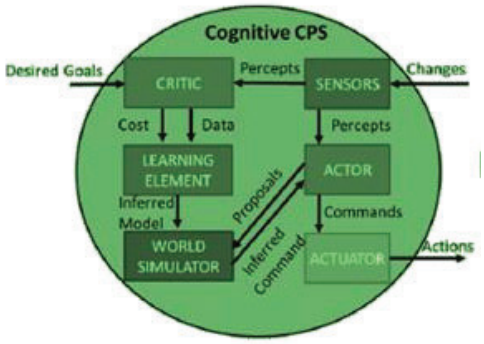


The **WHEEL** in the car:

- Is learning from its context
- Is adapting to the changes
- Is dynamically connected
- Is processing data
- …

© 2023 FRACTAL. All rights reserved

FRACTAL GIST



Therefore our **FRACTAL COGNITIVE NODE** will:

- Be learning from its context
- Be adapting to the changes
- Be dynamically connected
- Be processing data
- …

Our FRACTAL COGNITIVE NODE will be the **building block** of scalable decentralized Internet of Things (ranging from Smart Low-Energy Computing Systems to High-Performance Computing Edge Nodes).

© 2023 FRACTAL. All rights reserved



Project Description

The FRACTAL project aims to create **a reliable Edge Computing node**.

It will suit a range of computing systems, from smart low-energy to high-performance ones. Such flexibility is what will make it a building block of scalable and intelligent Internet-of-Things.

> *The project will build an Artificial Intelligence-powered computing edge node – the **"FRACTAL node"**. Cognitive, secure, adaptive and scalable, it will provide **a new approach to reliable Edge Computing**. It will be the building block of scalable decentralized IoT.*



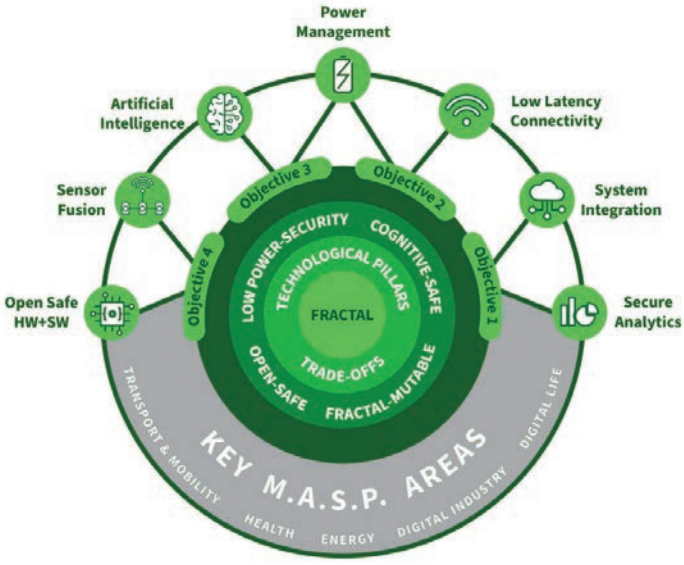
© 2023 FRACTAL. All rights reserved

Technology Pillars

The project objectives are held under four technology pillars. They represent crucial trade-offs between all specifications and characteristics of the node's hardware and software that need to be in place.

The project objectives are held under four technology pillars:

- **Open-safe-reliable and low power architecture**
- **Low power, secure** and **high performance** capabilities
- **Cognitivity** and **autonomy**, together with **safety**
- **Mutable** and **fractal** communication and storage capabilities, to allow scalability.



© 2023 FRACTAL. All rights reserved



Project Overview

Strategic Objectives

FRACTAL's ambition extends across hardware and software. It covers aspects related to AI, safety, security, low power, and edge integration.

1 To design and implement an open-safe-reliable hardware platform. It will be used for building the cognitive edge nodes of variable complexity

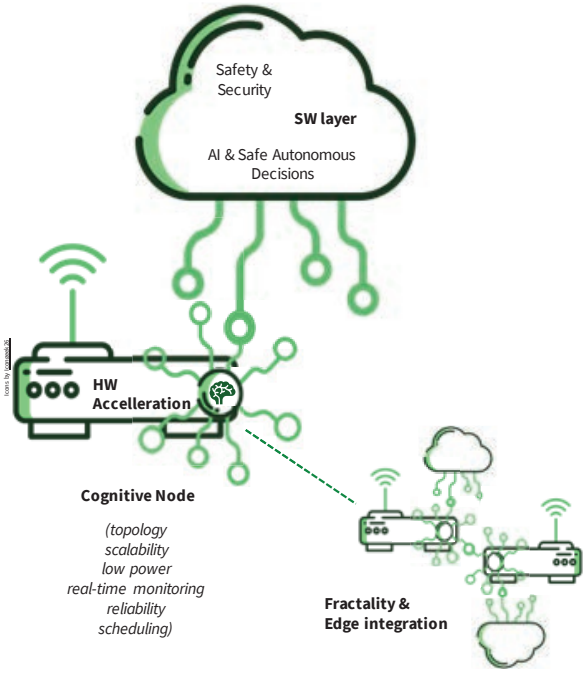
2 To guarantee extra-functional properties of FRACTAL nodes (dependability, security, timeliness and energy-efficiency)

3 To evaluate and validate data analytics with AI. To identify the largest set of working conditions, while preserving safe and secure operations

4 To integrate fractal communication and remote management features into the nodes.

© 2023 FRACTAL. All rights reserved

Project Overview



System Architecture & Node Integration

FRACTAL project puts in place a new Smart System paradigm. The cognitive node it creates, will bring the intelligence to the edge of the network.

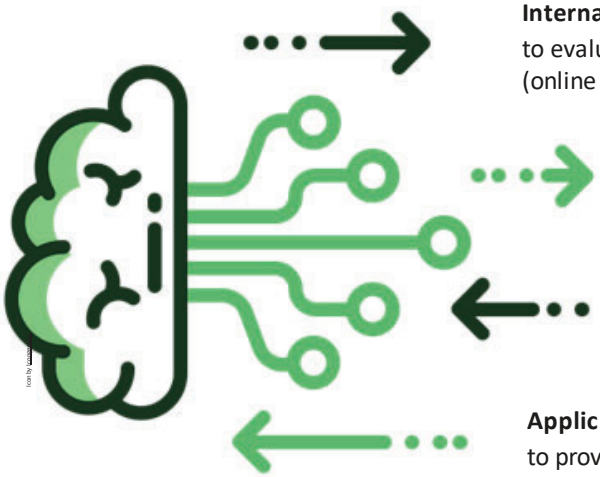
As a result, the FRACTAL node will pave the way to a new generation of smart systems. In such systems, the cloud will only need to manage and control the edge nodes.

© 2023 FRACTAL. All rights reserved



Project Overview

Features that will enable FRACTAL Node's Cognitivity:



Internal performance metrics supervision
to evaluate the fulfilment of the desired goals (online monitoring).

Operating environment modelling
to provide context-awareness for the Cognitive System (advanced sensing & computer vision).

Integration of computational capacity
to process data, models, and AI algorithms (low-power multicores and hardware acceleration).

Application of AI methods
to provide Smart Systems with learning, prediction and autonomy (artificial intelligence).

© 2023 FRACTAL. All rights reserved



Project Overview

Two different platforms offering different tradeoffs

- Commertial node

  VERSAL platform

  combines the low power and high-performance capabilities of hardened processors and other computational efficient cores with the flexibility of FPGAs by means of flexible network-on-chip architecture.

- Customizable node

  PULP platform

  provides a series of RISC-V cores (both 32 and 64-bit), interconnect solutions, and complete processing platforms from single-core microcontrollers suitable for Edge nodes, to cluster-based processing systems that can work as a gateway node all the way up to compute-intensive heterogeneous compute nodes comprising of multiple clusters and tens/hundreds of cores and accelerators.

© 2023 FRACTAL. All rights reserved



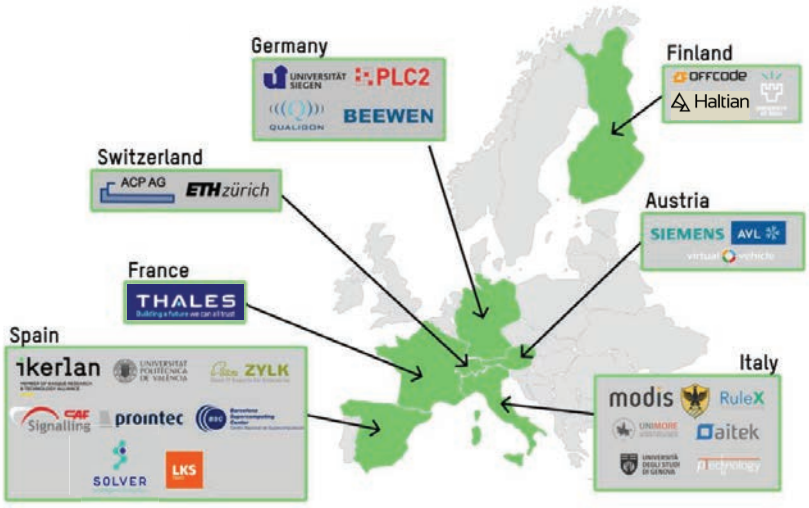
Demostrator

To validate the results:

- **8 use cases** have been proposed.
- To **maximize generalization** and scalability, the use cases have been chosen **in different application fields**:
  - Transport: Autonomous vehicles
  - Digital Life: Engineering
  - Digital Industry: Industry 4.0
  - Energy: Energy Consumption and Environmental
- The approach will be however general in order to guarantee the maximum scalability. Hence, the use cases share some **needs that are closely related to the objectives of FRACTAL**
  - Innovative methods for low power and high-performance edge architecture.
  - Advanced AI techniques for safe and reliable autonomous decisions.
  - Innovative methods for ensuring safe communication and storage.
  - Taking autonomous decisions quickly and without many computational resources.
  - Ensure security in data communication and storage in compliance with EU regulations.

© 2023 FRACTAL. All rights reserved

Project Overview



Project Details Summary

Partners
28

Person Months total
1 949.50

Number of Work Packages
9

Number of Tasks
34

Number of Deliverables
48

Number of Use Cases
8

Project Number
877056

Project Acronym
FRACTAL

Call (part)
H2020-ECSEL-2019-2-RIA

Starting Date
1.9.2020

Duration in months
36

Website:
www.fractal-project.eu

Topic
ECSEL-RIA-2019-2-Special-Topic-2 EDGE COMPUTING

Countries by name and partners per country
Spain (8), Italy (7), Austria (3), Germany (4), France (1), Switzerland (2), Finland (3)

Number of EU programs funded in this call special topic
2

Free keywords
Edge Computing, Open Source Software/Hardware, Time- critical, Reliability, Security, Complexity

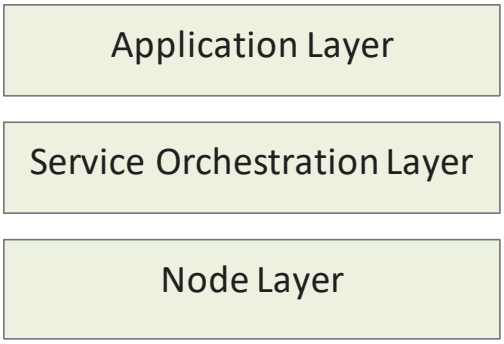
Coordinator
ikerlan

© 2023 FRACTAL. All rights reserved



FRACTAL SYSTEM

FRACTAL High Level Architecture



The FRACTAL system, from a high-level point of view, can be seen as a simple layered architecture with three layers:

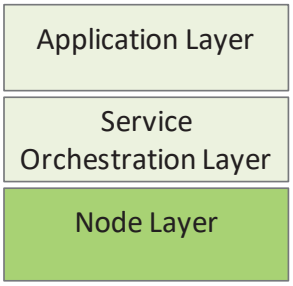
- A node layer that refers to the element that provides certain low-level characteristics such as AI Accelerators or computing power.
- Over the node layer, a service orchestration layer is built that enables to manage the services that run over the node.
- Finally, the application layer describes the business logic for the different applications / use case.

© 2020 FRACTAL. All rights reserved



FRACTAL High Level Architecture

Node Layer



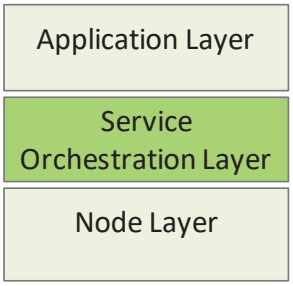
Node layer refers to Infrastructure:

- Cloud Node (Infrastructure)
- Fractal Edge HW Platform
  - Low Level SW Platform
    - OS
    - Drivers
    - Libraries
    - …
  - Low Level HW Platform
    - Versal / Noel-V / Pulp / CVA6
    - AI Accelerators
    - …

© 2020 FRACTAL. All rights reserved

FRACTAL High Level Architecture

Service Orchestration Layer



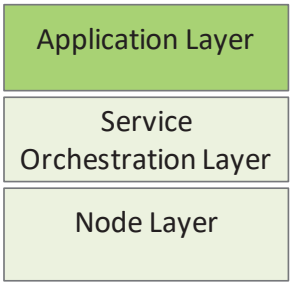
Over Node Layer, FRACTAL places a Service Orchestration Layer

- Executed over Cloud or Edge Infrastructure
- Standardizes development (application logic) and deployment of FRACTAL components as containers
- Manages the FRACTAL Distributed System
  - How nodes are added/removed in FRACTAL
  - How services are added / removed / configured in nodes
  - …

© 2020 FRACTAL. All rights reserved

FRACTAL High Level Architecture

Application Layer



Over the Service Orchestration Layer, application logic is added

- Specific functionalities (services) are built as microservices in containers and deployed over nodes
- Services make use of specific Node Layer offerings if needed (e.g. HW acceleration, diverse redundancy support, etc.)
- Services include:
  - Data ingestion
  - Data storage
  - Data preprocessing
  - Data analysis
  - Edge ML API
  - ML/AI Tools
  - …

© 2020 FRACTAL. All rights reserved



From High Level Architecture to Big Picture

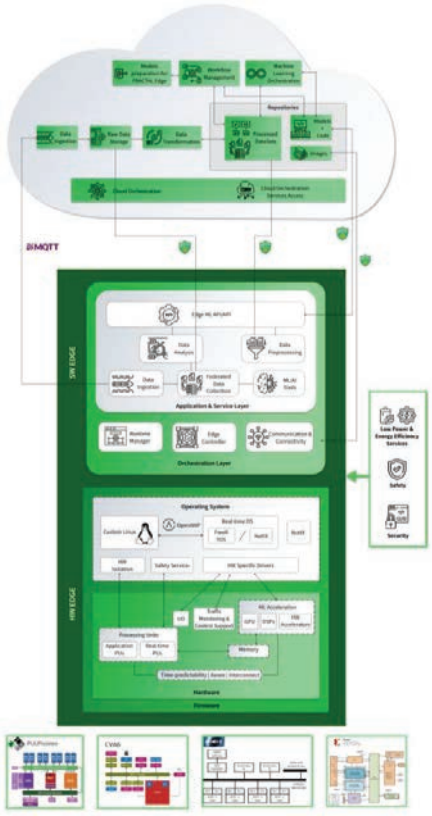
A FRACTAL application:

- Requires one or more FRACTAL Edge Nodes
- Might require Cloud Node

- Both cloud or edge nodes:
  - provide specific capabilities (e.g. AI Acceleration)
  - contain an orchestration layer
  - execute services or components required for application logic

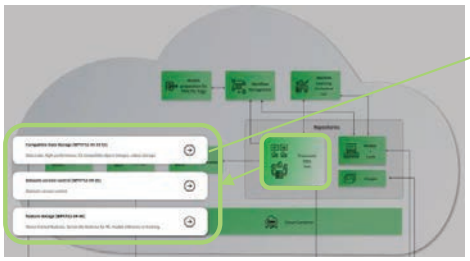
© 2020 FRACTAL. All rights reserved

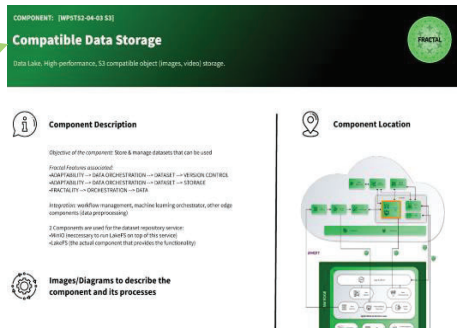
FRACTAL Big Picture



FRACTAL is a modular system. A set of components has been developed, for the different layers of the architectures. For different applications, a different set of components will be selected depending on the characteristics of the application.

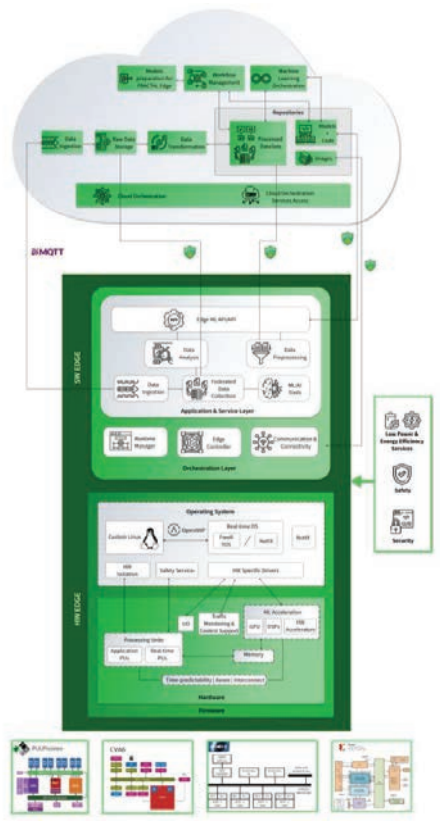
The Big Picture is a holistic representation of the FRACTAL System, reflecting all the distinct layers and components developed





© 2020 FRACTAL. All rights reserved

FRACTAL Big Picture



CLOUD SW PLATFORM

EDGE SW PLATFORM

LOW LEVEL SW PLATFORM

HW PLATFORM

EDGE HW PLATFORM

CLOUD NODE

EDGE NODE

FRACTAL SYSTEM

© 2020 FRACTAL. All rights reserved

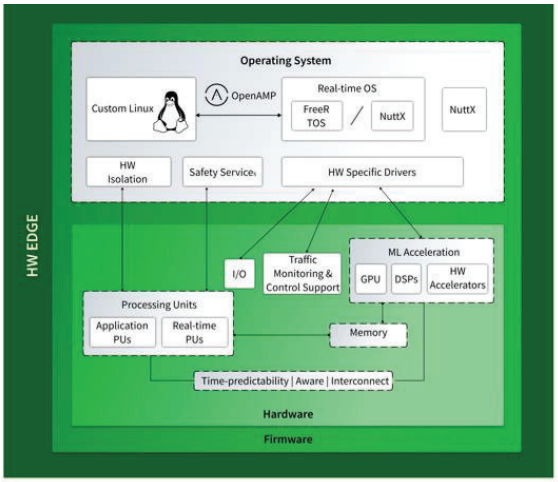
Edge HW Platform

The Edge HW Platform and its low-level SW are located in the Node Layer.

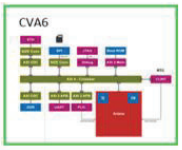
Application Layer

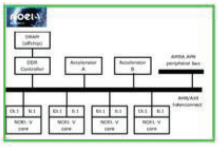
Service Orchestration Layer

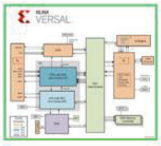
Node Layer





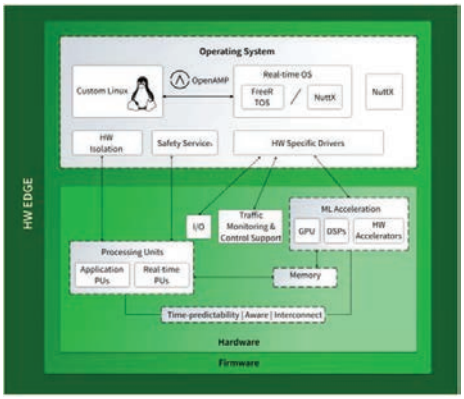






© 2020 FRACTAL. All rights reserved

Edge HW Platform

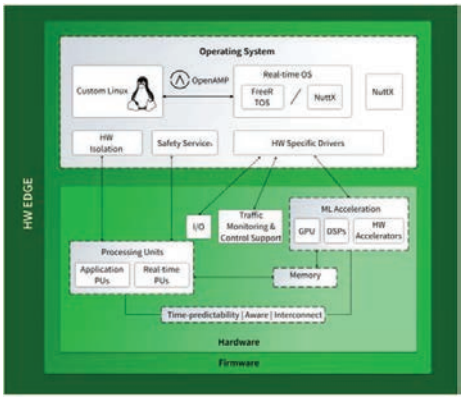


Components

- Hardware
  - AI Accelerator. Hardware and Software Support
    - **WP3T32-10 - VERSAL accelerator building-blocks:** Development of building-blocks for accelerators for VERSAL
    - **WP3T33-01 - Ariane RISC-V for ZCU102:** Porting Ariane 64bit to Xilinx ZCU 102 board
  - CPU and OS Support (HW Platform)
    - **WP3T32-02 - PULPissimo platform for IoT:** Base RISC-V platform that can be used to implement FRACTAL specific **components**
    - **WP3T32-02b - Ariane for Linux capable RISC-V platform:** Base RISC-V platform that can be used to implement FRACTAL specific functionalities
    - **WP3T32-03 - PULP trainings:** Training videos for using PULP
  - Safety and security features for CPU
    - **WP3T31-02 - Interconnect to support Accelerators integration**: Interconnect: AXI pulp library Integration
    - **WP3T31-03 - Safety and security hardware support:** Extensions to the interconnect and other NOEL-V components for Security and Safety
    - **WP3T32-09 - Runtime Bandwidth Regulator**: Memory bandwidth regulator that can be integrated on FPGA-based accelerator clusters, to improve main memory QoS and interference mitigation. This is a joint innovation with UNIVAQ.
- Application PUs
  - CPU and OS Support (HW Platform)
    - **WP3T32-11 - Smart Interrupt distribution system:** The smart interrupt distribution system prevents low priority interrupts to block real time critical threads
    - **WP3T33-03 - CVA6 (former Ariane) RISC-V core:** CVA6 (former ARIANE) application core enhanced with 32b flavor

© 2020 FRACTAL. All rights reserved

Edge HW Platform

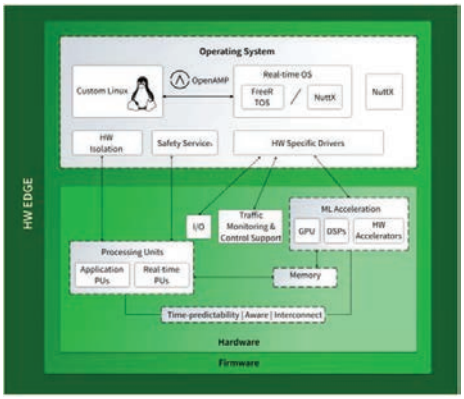


Components

- Real-time PUs
  - CPU and OS Support (HW Platform)
    - **WP3T32-08 - Real-time aware caches:** Real time aware cache prevents the user application to flush realtime code from the caches
    - **WP3T32-12 - Security services - TL2AXI adapter:** Bus protocol adapter allows to interface openTitan which hosts all secure services with pulpissimo
- Task Monitoring and Control Support:
  - Safety and security features for CPU
    - **WP3T31-01 - Edge-oriented monitoring unit:** AXI-compliant statistics unit to support safety measures and validation in the context of edge systems
- ML Acceleration
  - AI Accelerator. Hardware and Software Support
    - **WP3T32-07 - Age and Gender identifier at the Edge**
- HW Accelerators
  - AI Accelerator. Hardware and Software Support
    - **WP3T32-01 - HW accelerator (SIEFRACC):** HW accelerator for vision-based AI inference
    - **WP3T32-05 - ML acceleration on PULPissimo:** Example application that uses ML acceleration on PULPissimo
    - **WP3T35-02 - Accelerator Adaptation to AI library:** Implementing support for missing functionalities/layers and data formats
  - Safety and security features for CPU
    - **WP3T32-06 - Redundant Acceleration Scheme:** Integration of a redundant AI inference accelerator in the platform

© 2020 FRACTAL. All rights reserved

Edge HW Platform



Components

- Safety Services
  - Safety and security features for CPU
    - **WP3T34-01 - Driver for the edge-oriented monitoring unit:** Driver for the statistics unit supporting safety measures and validation in edge systems
    - **WP3T34-02 - Drivers for the SW diverse redundancy library:** Driver to read PMCs (Performance Monitoring Counters) from a remote core, and to issue SIG_STOP and SIG_CONT signals to remote cores
- Custom Linux
  - CPU and OS Support (HW Platform)
    - **WP3T36-01 - Linux for CVA6 (former Ariane):** Linux support for CVA6 (both 32b and 64b): UBoot (boot loader) + OpenSBI (firmware) + Yocto (embedded Linux generation)
- FreeRTOS
  - CPU and OS Support (HW Platform)
    - **WP3T32-04 - FreeRTOS port to PULP:** Port of the FreeRTOS to PULP
- Nuttx
  - CPU and OS Support (HW Platform)
    - **WP3T36-03 - Nuttx on PULP:** Porting Nuttx to PULP
- HW Specific Drivers
  - AI Accelerator. Hardware and Software Support
    - **WP3T34-03 - Versal Model deployment layer:** Model deployment on the Versal APU + DPU control from model repository images
    - **WP3T35-01 - SW driver for HW Accelerator:** SW driver for managing HW accelerator and data movements

© 2020 FRACTAL. All rights reserved

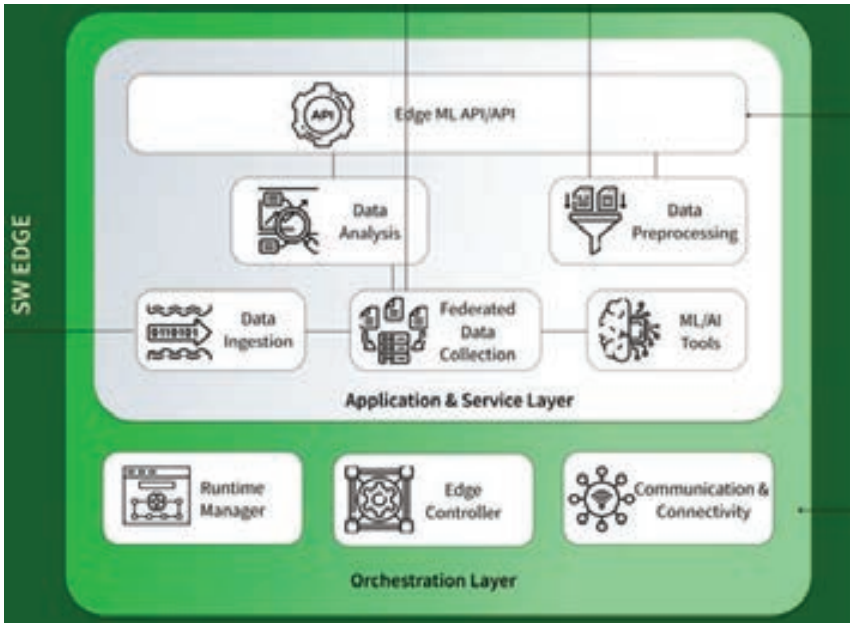
Edge SW Platform

In the Edge SW Platform, there are the components of the Edge Node that correspond to the Service Orchestration Layer & Application Layer

Application Layer

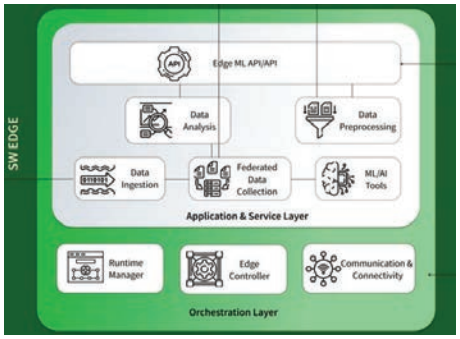
Service Orchestration Layer

Node Layer



© 2020 FRACTAL. All rights reserved

Edge SW Platform



Components

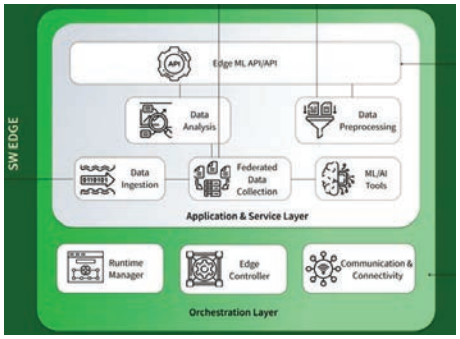
- Data Ingestion
  - **WP6T62-01 - Data Ingestion**: Data ingestion and data streaming processing tools for the High-End and Mid-End Fractal nodes
- Data Collection
  - **WP6T62-02 - Federated Data Collection:** Data storage capabilities for the High-End and Mid-End nodes
- ML/AI Tools
  - **WP6T61-05 - ML/AI Tools:** Tools to Enable ML/AI processing
  - **WP3T35-03 - LEDEL (Low Energy EDDL):** EDDL integration on NOEL-V
- Data Preprocessing
  - **WP6T61-04 - Data Preprocessing:** A component for data preparation. Transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks
- Data Analysis
  - **WP6T61-03 - Data Analysis:** Cleaning, transforming, and modeling data to discover useful information for decision-making
- Edge ML API
  - **WP6T61-01 - Edge ML API:** Services and interface that provide a set of functions and machine learning APIs for to use and integrate different AI and ML techniques to obtain relevant data and access edge software or other applications
  - **WP6T61-01 - Edge ML API:** Services and interface that makes it easy for one application to consume capabilities or data from another application
- Orchestration
  - **WP6T62-03 - Run time Manager:** Management of components interaction, tasks scheduling and node-to-node communication
  - **WP6T62-06- Orchestration:** Edge controller custom orchestrator tools & Code. The Edge controller is an autonomous orchestrator for containers to support K8S, Docker, or orchestrator-less Fractal nodes

© 2020 FRACTAL. All rights reserved

Edge SW Platform

Components

- Application & Service Layer
  - **WP3T35-04 - Deep learning based automatic iris diagnosis:** Automatic diagnosis of iris pathologies from high-definition ocular fundus images. Based on a deep learning architecture for classification potentially enhanced with advanced pre-processing techniques such as CLAHE for contrast augmentation.
  - **WP3T35-05 - Idiom Recognition:** Idiom/Language recognition system based on speech signal registration through Speech-to-Text ML solutions.
  - **WP3T36-02 - Load Balancing Module:** Software module designed to collect computational loads from nodes and, in case of overload, able to distribute computional load.



© 2020 FRACTAL. All rights reserved

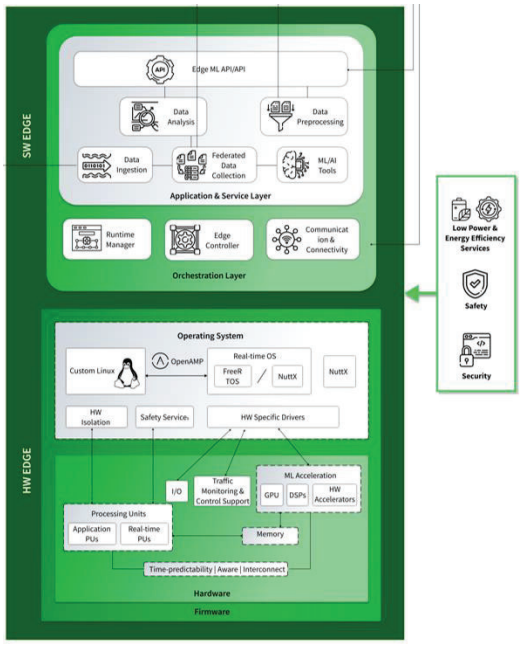
Edge Node

The Fractal Edge Node has certain features that affects to several parts in Edge HW Platform and Edge SW Platform: Low power and Energy efficiency Services, Safety and Security

Application Layer

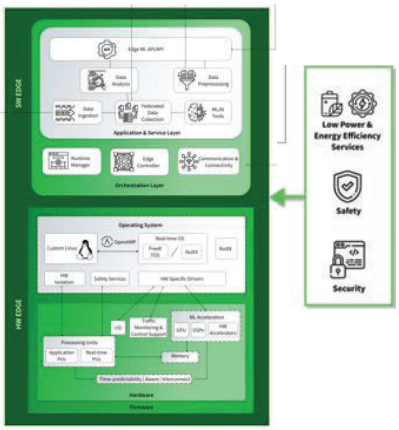
Service Orchestration Layer

Node Layer



© 2020 FRACTAL. All rights reserved

Edge Node

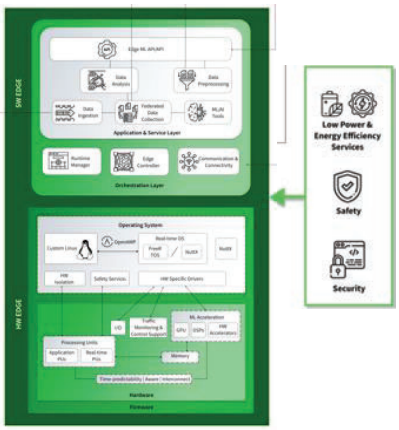


Components

- Low Power Services
  - **WP4T41-01 - Data Compression for Low-Power Services:** Data compression technique for low-power devices, to be applied at system level (affects to Federated Data Collection, in SW Edge)
  - **WP4T41-04 - Versal RPU access for Power Services:** Access Dynamic Power, Frequency Scaling features on Versal [VCK190] (affects to Custom Linux, FreeRTOS and Nuttx in HW Edge)
  - **WP4T41-05 - Agreement protocol for Low-Power Services:** Implementation for the agreement protocol on a wireless network on low-power devices (affects to Custom Linux, FreeRTOS and Nuttx in HW Edge)
  - **WP4T41-02 – HATMA:** Hierarchical Adaptive Time-triggered Multi-core Architecture to facilitate services at the different hierarchies (affects to Hardware layer in HW Edge)
  - **WP4T41-03 - Low Power services for PULP systems:** Enhancements to existing platforms to support low power FRACTAL services (affects to Hardware layer in HW Edge)
  - **WP4T41-06 - Versal Isolation Design - Functional Safety:** SW driver for managing HW accelerator and data movements (affects to Hardware layer in HW Edge)
- Energy Efficiency Services
  - **WP4T42-01- Capabilities for AI supported adaptability in PULP**: Enhancements to existing platforms to support resource allocation (affects to Hardware layer in HW Edge)
  - **WP4T42-02 - Versal RPU access to AI acceleration:** Enhance RPU libraries to (1) access APU based AI as a service, (2) enable local AI [acceleration] deployment from RPU (affects to Custom Linux, FreeRTOS and Nuttx in HW Edge)
  - **WP4T42-03 - Scenario Generator:** Scenario Generator that provides the inputs for the machine learning algorithm (affects to ML/AI Tools in SW Edge)
  - **WP4T42-04 - GA-Scheduler:** Scheduler (Genetic Algorithm) that provides the solutions of the scheduling problems given by the Scenario generator component (affects to ML/AI Tools in SW Edge)
  - **WP4T42-05 - AI-Scheduler Model:** ML model used to predict schedules (affects to ML/AI Tools in SW Edge)
  - **WP4T42-06 - Schedule Verifier:** Schedule verifier/reconstructor that takes the predictions of the machine learning model and convert them into a schedule (affects to ML/AI Tools in SW Edge)
  - **WP4T42-07 - Hierarchical Metascheduler:** Offline tool to compute time-triggered schedules by considering context events such as dynamic slack, failure scenarios (affects to ML/AI Tools in SW Edge)

© 2020 FRACTAL. All rights reserved

Edge Node

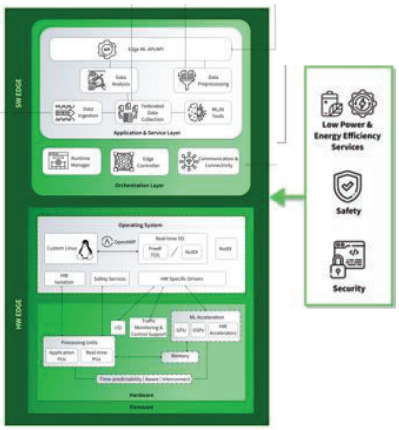


Components

- Safety
  - **WP4T43-01 - Performance monitoring services:** Services to configure the multicore-aware monitoring unit and retrieve information on the multicore interference observed (located in Safety Services, in HW Edge)
  - **WP4T43-02 - Safety services for PULP systems:** Enhancements to existing PULP systems to support safety services (located in Safety Services, in HW Edge)
  - **WP4T43-03 - SW diverse redundancy library:** Library allowing to run a task redundantly in two RISC-V cores enforcing some staggering among them to avoid common cause faults (located in Safety Services, in HW Edge)
  - **WP4T43-04 – ATTNoC:** Adaptive TTNoC provides time triggered communication for NoC and allow the systems to switch schedules in case of any failures occurs in the NoC (located in Time-predictability aware interconnect, in HW Edge)
  - **WP4T43-05 - Redundant Acceleration Scheme Safety Analysis:** Safety analysis after inspection and fault-injection of a redundant acceleration scheme (affects to HW accelerators in HW Edge)
  - **WP4T43-06 - FPGA Fault-injector:** A tool to inject faults in the NOEL-V multicore. It is actually suitable for any Ultrascale + FPGA (affects to HW accelerators in HW Edge)
  - **WP4T43-07 - Safety Case:** Safety Case ensuring that any hazard in the control of automotive components / system cannot lead to potential life treating danger (located in Safety Services, in HW Edge)
  - **WP4T43-08 - Seamless redundancy for ATTNoC**: Seamless redundancy provides fault tolerance on the NoC, by sending two set of seamless data with seamless path at the same time, so failures in one path can be mask
  - **WP4T43-09 - Safety functions consideration during ML:** Integration of safety functions during preprocessing, to avoid trigerring safety reaction in the field (located in Safety Services, in HW Edge)
  - **WP4T43-10 - Safety functions consideration during Reinforcement learning:** Integration of safety functions in learning procedure for reinforcement learning to avoid trigerring safety reaction in the field (located in Safety Services, in HW Edge)
  - **WP4T43-11 - Time-Triggered Extension Layer for VERSAL NoC:** Time Triggered extension layer is an extension layer developed for VERSAL NoC that allow the VERSAL NoC to transfer messages using Time triggered traffic (located in Time-predictability aware interconnect, in HW Edge)

© 2020 FRACTAL. All rights reserved

Edge Node



Components

- Security
  - **WP4T44-01 - Security services for PULP systems:** Enhancements to existing PULP systems to support security services / components/technologies for secure operation (affects to Operating System in HW Edge)
  - **WP4T44-02 - OS Security Layer:** Implementation of security countermeasures in a transversal security layer (affects to Operating System in HW Edge)
  - **WP4T44-03 - Security services at application and network layers:** Security services are being implemented at app and network level, making applications running on the platform secure and also the communications between nodes or microservices exchanging data. E.g. HTTPs, TLS, user auth (affects to orchestration layer, in SW Node, and to the communication with other nodes)
  - **WP4T44-05 - IoT Gateway:** IoT network Gateway for external communication monitoring (affects to orchestration layer, in SW Node, and to the communication with other nodes)
  - **WP4T44-07 - Node monitoring and system status:** Metrics collector for security related aspects and issue addressing (affects to orchestration layer, in SW Node, and to the communication with other nodes)
  - **WP4T44-08 - TLS Implementation on containers:** Inter-nodal secure communications (affects to orchestration layer, in SW Node, and to the communication with other nodes)
  - **WP4T44-09 - Runtime security:** Process isolation through containerization and user control (affects to orchestration layer, in SW Node, and to the communication with other nodes)

© 2020 FRACTAL. All rights reserved

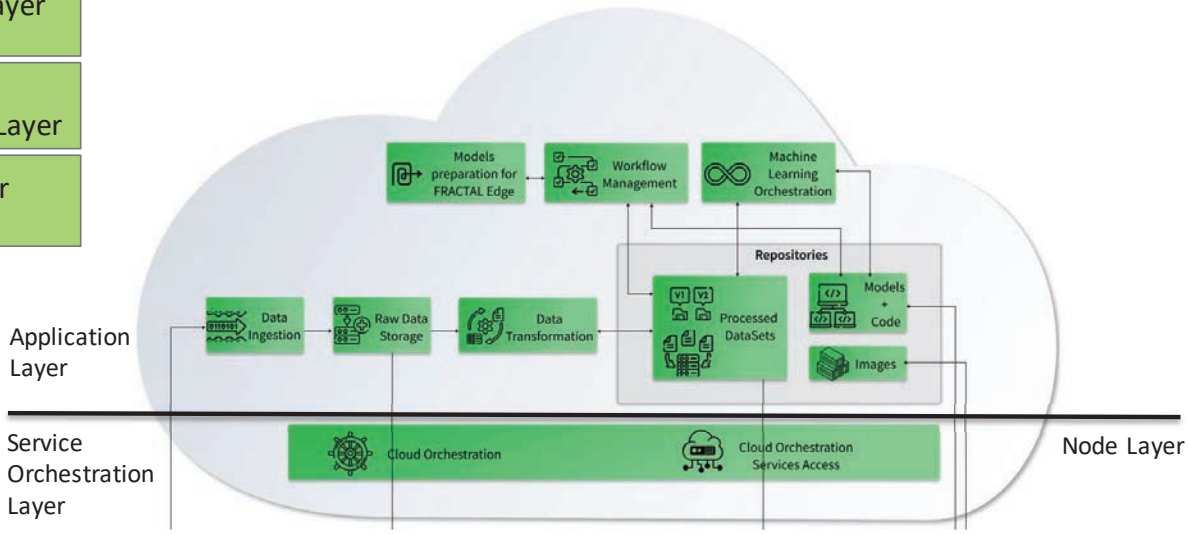
Cloud Node

In the Cloud Node, the node layer corresponds to the infrastructure provided by the Cloud provider, and the components developed for this node correspond to the Service Orchestration Layer and Application Layer.

Application Layer

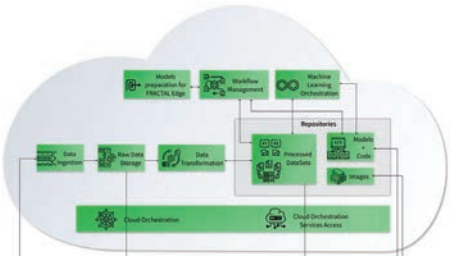
Service Orchestration Layer

Node Layer



© 2020 FRACTAL. All rights reserved

Cloud Node



Components

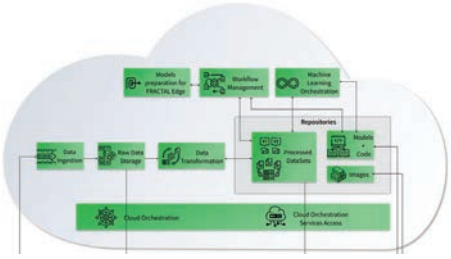
- Data Ingestion
  - **WP5T52-01-01 - Data Ingestion Service**: The platform allows to ingest data in two different approaches: on the first approach, streaming data ingestion, and bulk data ingestion. Streaming data ingestion will be addressed by the data ingestion service (i.e., Kafka), whereas bulk data ingestion will be supported by the raw data storage component.
- Raw Data Storage
  - **WP5T52-02-01 - Raw Data Object storage service:** Cloud service to store the raw data for training/retraining models
- Data Transformation
  - **WP5T52-03-01 Data Transformation:** This component retrieves the raw data stored in the data storage component and applies to them different data pre-processing techniques in order to transform the raw data into a dataset that is prepared and optimized to be exploited by using advanced data analytic techniques
- Repositories
  - Processed Datasets

    Storage and management of the processed data sets that will be used to train machine learning models
    - **WP5T52-04-03 - S3 compatible data storage**: Data Lake. High-performance, S3 compatible object (images, video) storage.
    - **WP5T52-04-05 - Datasets version control**
    - **WP5T52-04-06 - Feature storage:** Stores trained features. Serves the features for ML models inference or trainings
  - Models + Code

    Storage and management of Machine Learning models (code, parametrization, configurations,...)
    - **WP5T52-04-01 - Models version control**.
    - **WP5T52-04-08 - Model repository:** Metadata version control and Models version control framework
    - **WP5T52-04-10 - MLBuffet as a Cloud Model Storage:** Light-weight Model repository
  - Images
    - **WP5T52-04-07 - Images repository:** Container Registry for Docker Images

© 2020 FRACTAL. All rights reserved

Cloud Node



Components

- Machine Learning Orchestration
  Management and orchestration of the workflows related to the machine learning models (training, evaluation and optimization, serving and integration with models' storage components)
  - **WP5T52-04-09 - Machine learning pipeline**
  - **WP5T52-06-02 - ML pipeline connection to model repository**.
  - **WP5T52-05-01 - MLBuffet as a Cloud Model Orchestrator:** FRACTAL models orchestration (Manage and run models on cloud). It can be used as a validation for the models to be deployed on the Edge to be running properly.
- Workflow Management
  - **WP5T52-05-02 - Data pipelines and workflows orchestrator**
- Models' preparation for Fractal Edge
  - **WP5T52-06-01 - Model preparation for Fractal Edge** (Versal Xilinx Vitis AI): Workflows to compile models for Versal with Xilinx Vitis AI. Add containerized toolchain to the cloud
- Orchestration of cloud services
  - **WP5T52-07-01 - Kubernetes-based cloud platform container orchestrator** (Versal Xilinx Vitis AI): Managed Kubernetes Service for containerized services orchestration.
  - **WP5T52-07-02 - Cloud container orchestrator services access:** Cloud container orchestrator services access

© 2020 FRACTAL. All rights reserved



FRACTAL FEATURES



FRACTAL Features

How can we better define what FRACTAL is and which components are to be included in a use case?

© 2023 FRACTAL. All rights reserved



FRACTAL Features

There are many definitions of Feature, coming from the Software Product Line (SPL) community. Using the most notable definition from FODA*:

A "feature" is defined as a "prominent or distinctive user-visible aspect, quality, or characteristic of a system".

SPLs refer to engineering techniques for creating a portfolio of similar systems from a shared set of assets (components) using a **common means of production**. This portfolio of similar systems is called a **Family of Products**.

* Kang, K.C. and Cohen, S.G. and Hess, J.A. and Novak, W.E. and Peterson, A.S., "*Feature-oriented domain analysis (FODA) feasibility study*", Technical Report CMU/SEI-90-TR-021, SEI, Carnegie Mellon University, November 1990

© 2023 FRACTAL. All rights reserved



FRACTAL Features

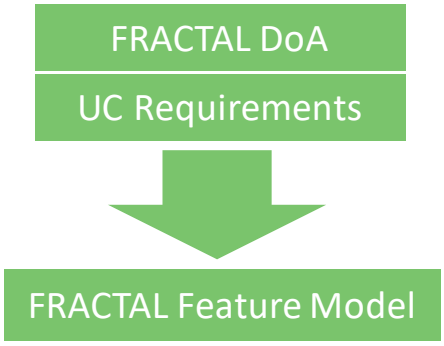
A **FRACTAL Feature** is a **distinguishing characteristic of FRACTAL**, visible to **users that will configure** FRACTAL for their use cases.

FRACTAL Features can describe functional characteristics (hardware, software), nonfunctional characteristics (performance or other criteria) or even other parameters (cost, max weight).

A **Feature Model** is a ***hierarchical diagram*** that visually depicts the features of a solution in ***groups of increasing levels of detail***. Feature Models are compact representations of all the products in a Family in terms of features and provide a great way to summarize the features that will be included in a solution and how they are related in a simple visual manner.

© 2023 FRACTAL. All rights reserved

FRACTAL Features



Use Case **requirements** are the functional and non-functional needs captured by FRACTAL use cases at the beginning of the project.

From FRACTAL Concepts on the Document of Agreement (DoA) and the specific UC requirements a set of **features** were established to give to the requirements a technical notion.

© 2022 FRACTAL. All rights reserved



FRACTAL Features

Some FRACTAL concepts from DoA and project title

- **Adaptability**
  - Ability to adapt system behavior to surrounding scenario.
- **Context-awareness**
  - Ability to detect and react to context and surroundings.
- **Openness**
  - Fractal Core should be Open to be used.
- **Safety**
  - Be resilient to internal failures.
- **Security**
  - Be resilient to external attacks and menaces.
- **Reliability**
  - The system should behave consistently well.
- **Low Power**
  - Ability to support low power scenarios.
- **Fractality**
  - Ability to support fractal configurations.

© 2022 FRACTAL. All rights reserved



FRACTAL Definition

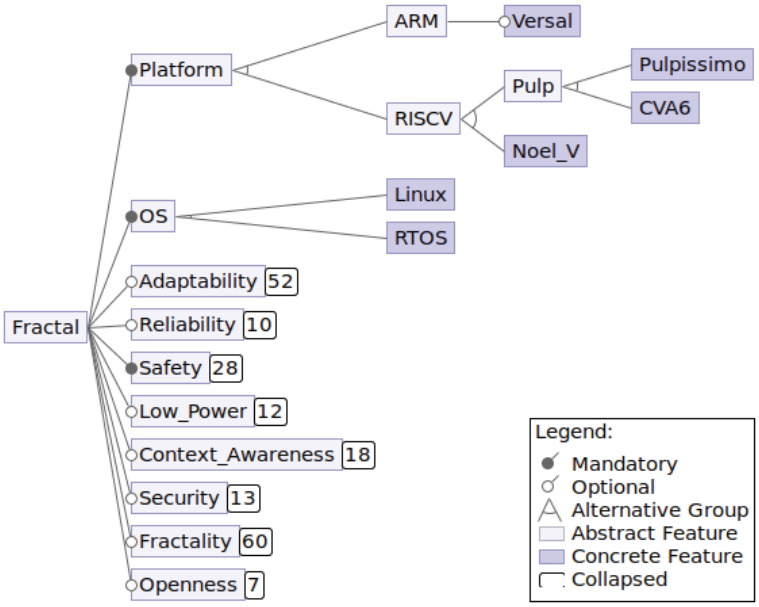
A **FRACTAL Edge Node** is a flexible EDGE node whose architecture must support following characteristics:

- Adaptability
- Context-awareness
- Openness
- Safety
- Security
- Reliability
- Low Power
- Fractality

•A node architecture that supports these characteristics will be the building block of scalable **decentralized** Internet of Things.

© 2022 FRACTAL. All rights reserved

FRACTAL Feature Model: High Level Features



The **FRACTAL Feature Model** describes the variability in **FRACTAL's family of products** in terms of **FRACTAL Features**.

Use Cases represent **variants of possible FRACTAL products**, this is, specific product realizations.

© 2022 FRACTAL. All rights reserved



FRACTAL Features from UC Requirements I

REQ_UC4_06 - *The edge node shall provide dedicated HW accelerator to process the CNN layers (Yolo) of AI inference*

The following features have been defined:

- Adaptability -> AI -> HW -> AI ML Accelerator
- Adaptability -> AI -> SW -> Inference -> Location -> Edge Node
- Adaptability -> AI -> SW -> Learning/Training -> Algorithm -> Yolo
- Adaptability -> AI -> SW -> Learning/Training -> Algorithm -> CNN

© 2020 FRACTAL. All rights reserved



FRACTAL Features from UC Requirements II

REQ_UC6_14 - *The edge node shall acquire images from at least one HD camera.*

The following features have been defined:

- Context Awareness -> Sensors -> HD Camera

© 2020 FRACTAL. All rights reserved



FRACTAL Features from UC Requirements III

REQ_UC2_02 - *The edge node shall be capable of communicating with cloud services for federated learning as well as diagnosis and adaptation.*

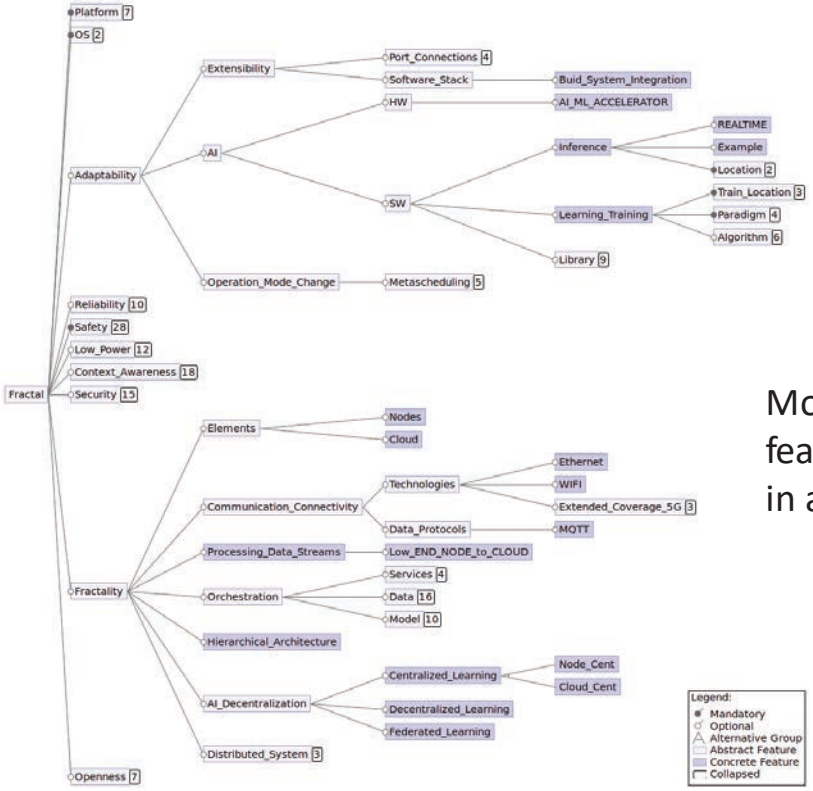
The following features have been defined:

- FRACTALITY -> AI DECENTRALIZATION -> FEDERATED LEARNING
- FRACTALITY -> COMMUNICATION
- FRACTALITY -> ORCHESTRATION -> DATA

© 2020 FRACTAL. All rights reserved

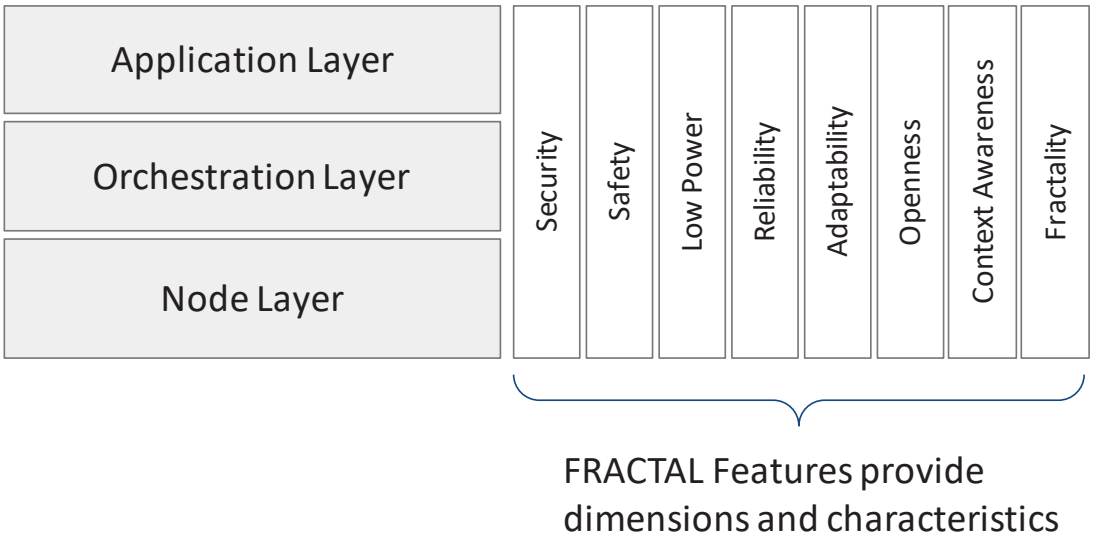
FRACTAL Complete Feature Model





More than 200 features organized in a tree

FRACTAL Features & FRACTAL High Level Architecture



© 2023 FRACTAL. All rights reserved



FRACTAL Features & FRACTAL High Level Architecture

Adaptability

ADAPTABILITY -> AI -> SW -> LEARNING/TRAINING

- AI training is provided by the corresponding business logic components in the application Layer.
- Service Orchestration defines where AI training components are executed (cloud / edge / both).
- AI training components make use of node characteristics
  - HW acceleration

FRACTAL Features cross-cut!

Fractality

FRACTALITY -> AI DECENTRALIZATION

- Centralized / Decentralized / Federated Learning have implications on Service Orchestration for AI Training components

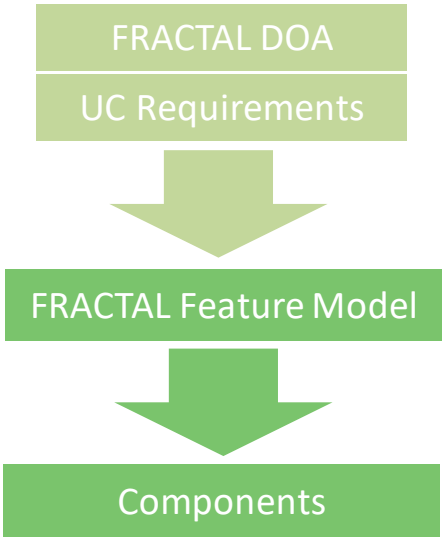
© 2022 FRACTAL. All rights reserved

FRACTAL Features & FRACTAL Components

The **components** are the building blocks that fulfill the FRACTAL features. They can be made of software or hardware and have been developed in the technical work packages.

Features provide **traceability** from UC needs to FRACTAL components

- **WP5T52-01-01: Data Ingestion Service (Cloud)**
  - FRACTALITY -> ORCHESTRATION -> DATA -> INGESTION
- **WP6T62-01: Data Ingestion (Node)**
  - FRACTALITY -> ORCHESTRATION -> DATA -> INGESTION
- **WP3T35-03: LEDEL (Low Energy EDDL)**
  - ADAPTABILITY -> AI -> SW -> LIBRARY -> LEDEL
  - LOW POWER -> AI LIBRARIES -> LEDEL



© 2022 FRACTAL. All rights reserved



BUILDING FRACTAL



Building a FRACTAL Use Case



Based on the needs of the Use Case, features will be selected conforming the specific set of features for the UC.

For example:

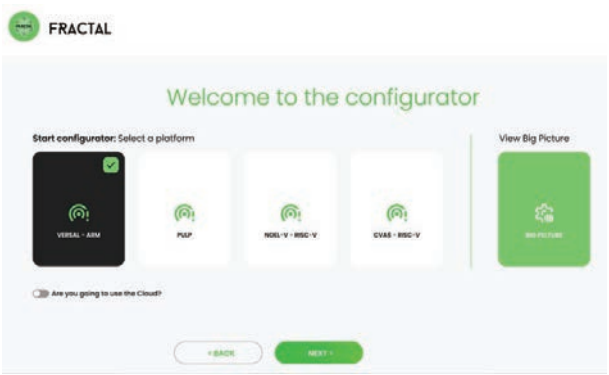
- **PLATFORM -> VERSAL** will be selected as the more powerful platform used in FRACTAL.
- **RELIABILITY -> AVAILABILITY -> LOAD BALANCING** will be selected as processes will be load balanced among available nodes.
- **SAFETY -> MONITORING -> PERFORMANCE** will be selected as nodes performance will be monitored to decide process balancing.

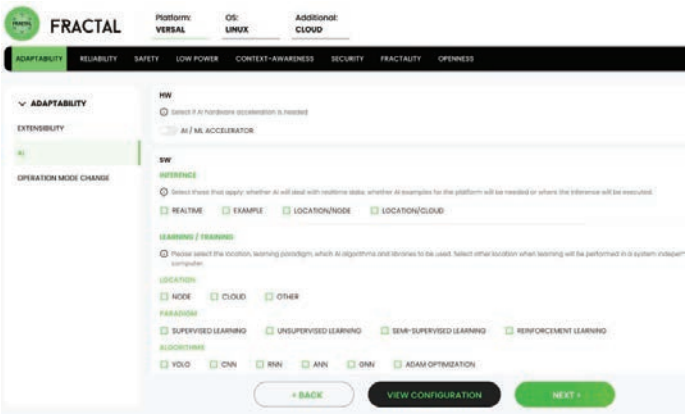
The specific selection of FRACTAL Features by a UC is called a **Bill of Features.**

© 2022 FRACTAL. All rights reserved

Feature Selection Demo

UC Requirements → Feature selection → Bill Of Features → Architecture & Selected Components → Building Process

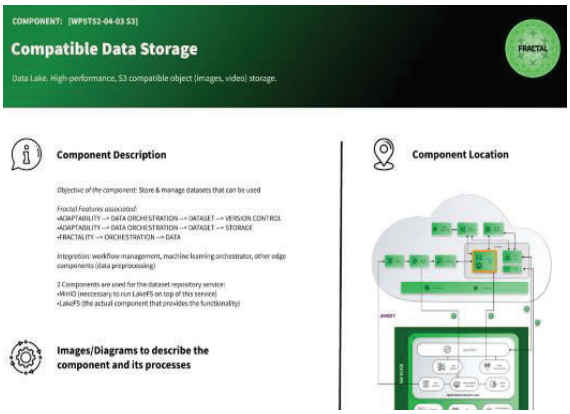




© 2022 FRACTAL. All rights reserved

Feature Selection Demo

UC Requirements → Feature selection → Bill Of Features → Architecture & Selected Components → Building Process



Building Process:

- FRACTAL Platform Construction
- Component Usage Descriptions

© 2022 FRACTAL. All rights reserved



Building Inteligent Totem UC

UC Remarkable features and corresponding selected components:

- **RELIABILITY -> AVAILABILITY -> LOAD BALANCING:**
  - **WP6T62-03 Runtime Manager:** Management of components interaction, tasks scheduling and node-to-node communication
  - **WP3T36-02 Load Balancing Module**: Software module designed to collect computational loads from nodes and, in case of overload, able to distribute computional load.

- **SAFETY -> MONITORING -> PERFORMANCE**
  - **WP3T32-09 Runtime Bandwidth Regulator:** Memory bandwidth regulator that can be integrated on FPGA-based accelerator clusters, to improve main memory QoS and interference mitigation. This is a joint innovation with UNIVAQ.

© 2022 FRACTAL. All rights reserved



USE CASE - INTELLIGENT TOTEM

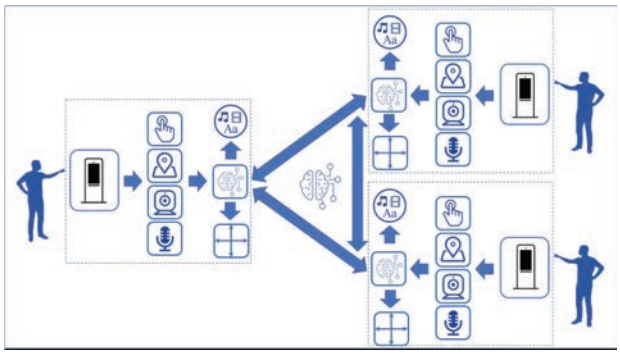
Use Case Intelligent Totem - Overview (1)

- The goal of the UC6 "Intelligent Totem" is to transform a shopping mall into a **sentient space**;
  - By embedding FRACTAL processing resources within the environment.
- These totems **could have a disruptive impact** on retail and shopping mall business
  - personalized advertisements;
  - product recommendations;
  - driving customers towards their selected destination/product.



© 2023 FRACTAL. All rights reserved

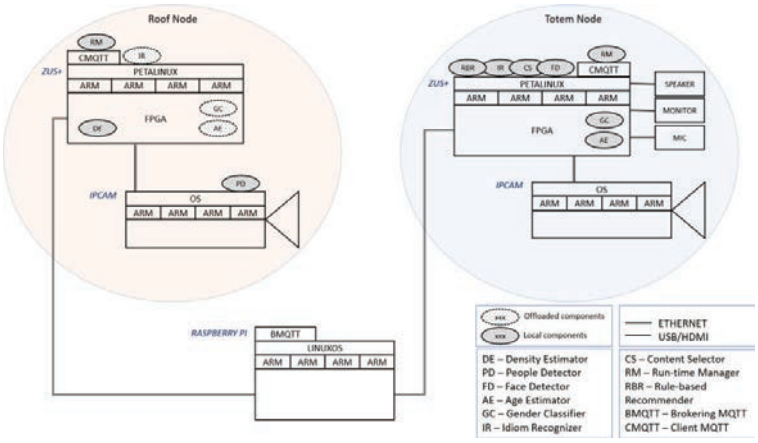
Use Case Intelligent Totem - Overview (2)



- Obtained information from totems can be shared with other elements in the same area;
  - To exchange locally detected information.
- Similar contents can be shown to the **same group of people, in different locations**, using different totems;
- The main detected information are:
  - Customer's gender and age range using images;
  - Customer's language using audio processing;
  - Etc.

© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem - System Architecture

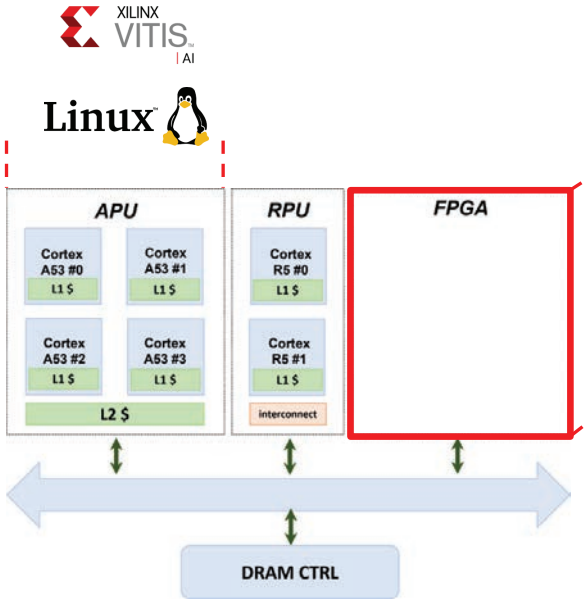


- Overall **system architecture** of the Intelligent Totem UC;
- The system architecture is composed of **three main FRACTAL nodes**:

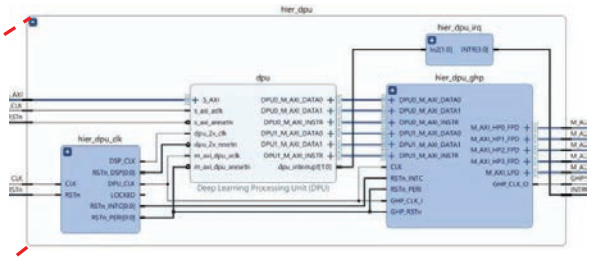
1. **Roof node**: CCTV camera and a FRACTAL node that executes video analysis and hosting some components;
2. **Totem node**: interactive totem with camera, microphone, speakers and a touch display hosted on a FRACTAL node;
3. **MQTT broker**: used for the MQTT communications between FRACTAL nodes.

© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem - FRACTAL nodes configuration



XILINX Zynq UltraScale+ (ZCU102)



XILINX DPU engine for AI acceleration

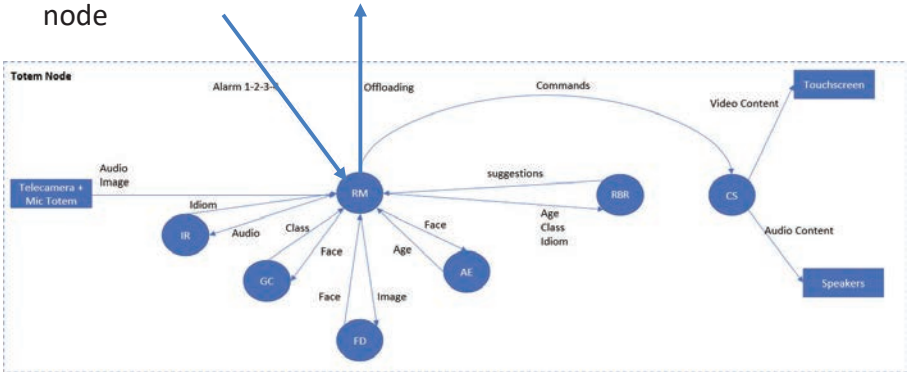
- **Software**:
  - Petalinux with Vitis AI runtime
- **Hardware**:
  - XILINX DPU engine deployed on FPGA.

© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem - TOTEM node

From the ROOF node

To the ROOF node



- **Inputs**:
  - **Video and audio** from Camera and Microphone;
  - 4 different **alarms** from the ROOF node.
- **Outputs**:
  - **Personalized advertisements** using AI.

- Overview of the main SW components **deployed on the TOTEM node;**
  - Runtime manager;
  - Face detector;
  - Idiom recognizer;
  - Age classifier;
  - Gender classifier;
  - Rule-based recommender;
  - Content selector.

© 2020 FRACTAL. All rights reserved



Use Case Intelligent Totem – RM

- The **Runtime Manager (RM)** coordinates and manages **task scheduling** and **load balancing operations** between modules in one or more FRACTAL nodes at runtime;
- It performs the scheduling of various operations which are entirely configurable.
- In addition, it provides **load balancing capabilities** using the interface with the Load Balancer component, sending the task execution to a different node.

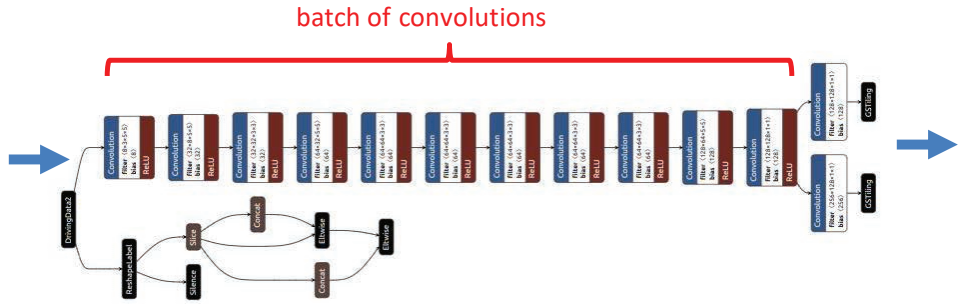
© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem – FD

- The **Face Detector** component is based on a **DenseBox** deep learning model.

(Input image from the TOTEM camera)



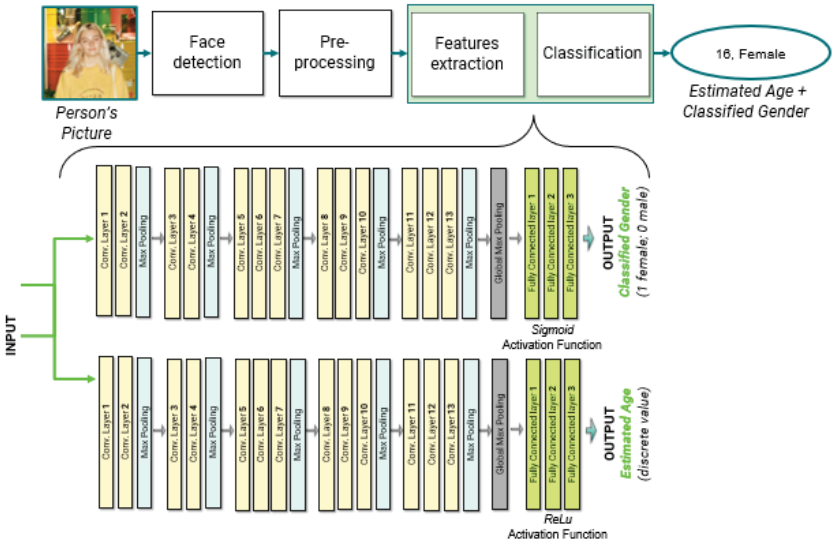


A sequence of small images containing faces

Lichao Huang and Yi Yang and Yafeng Deng and Yinan Yu, ***DenseBox: Unifying Landmark Localization with End to End Object Detection***, arXiv, 2015.

© 2020 FRACTAL. All rights reserved

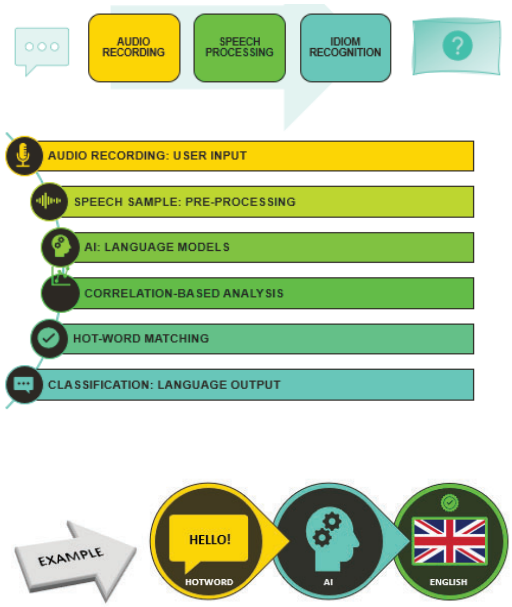
Use Case Intelligent Totem – AE/GC



- Given an input image the AE/GC components can estimate:
  - Value for the Age and Gender.

© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem – IR



- The IR module needs a speech recording to process;
- To recognize the language the module needs to detect a hotword;
- Currently supported hotwords and corresponding languages:
  - **Italian**: Ciao, Buongiorno;
  - **English**: Hi, Hello.

© 2020 FRACTAL. All rights reserved

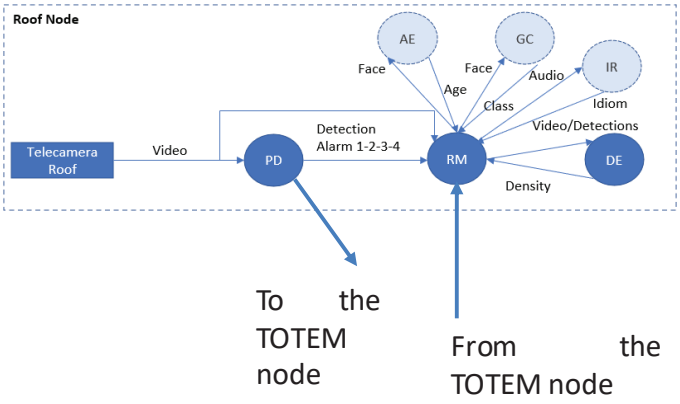
Use Case Intelligent Totem – RBR/CS

- The Rule Based Recommender (RBR) is the **final stage of the TOTEM node** components;
- Based on the information coming from the other modules, provides the **right content to the final user**.



© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem - ROOF node



- Overview of the main SW components **deployed on the ROOF node;**
  - Runtime manager;
  - People detector;
  - Density estimator.
- crucial features of the TOTEM (AE, GC and IR) can be **offloaded to the ROOF using FRACTALITY.**

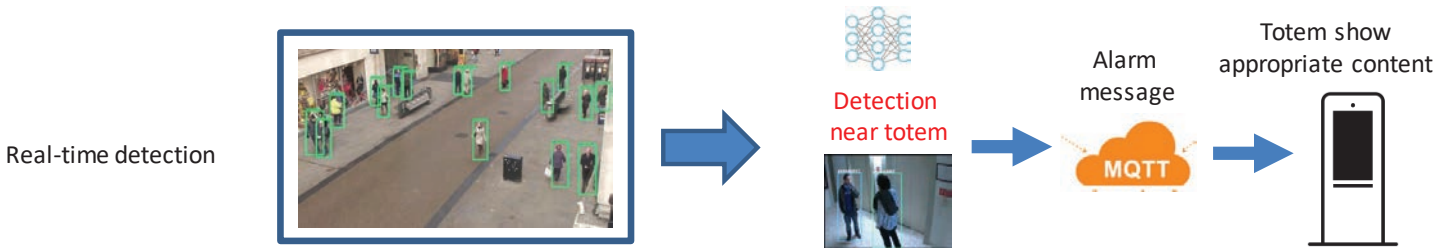
- **Inputs**:
  - **Video** from a camera.
- **Outputs**:
  - 4 different **alarms** to the TOTEM node.

© 2020 FRACTAL. All rights reserved



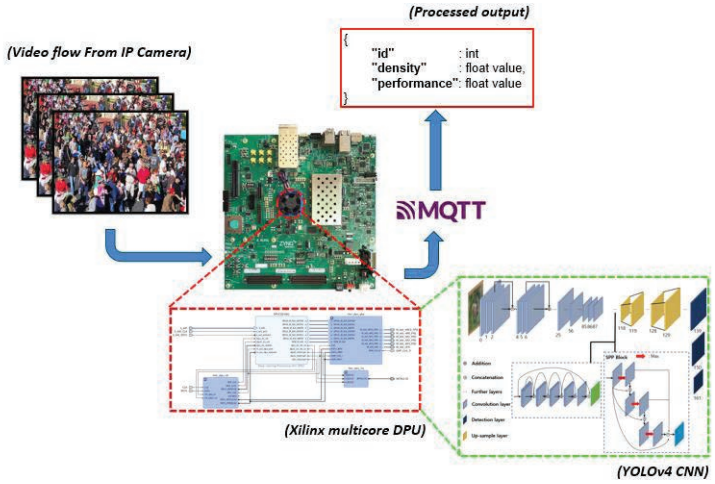
Use Case Intelligent Totem – PD

- The People Detection (PD) is done using the object detection technique known as **SSD (Single Shot Detector);**
- Different alarms are triggered when people are detected:
  - When at least one person is in the hall;
  - When there is a person which is using the totem.



© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem – DE



- The Density Estimator (DE) takes as input a sequence of image frames and provides an **estimation of the people density** as output;
- The DE is based on the **YOLOv4** neural network model.

Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, ***YOLOv4: Optimal Speed and Accuracy of Object Detection***, arXiv, 2020.

© 2020 FRACTAL. All rights reserved

Use Case Intelligent Totem – Overview of the system



- Whole area of interest of the totem;
- A user approaching the totem is detected by the People Detector (PD) component;
- The totem activation area.

© 2020 FRACTAL. All rights reserved



Use Case Intelligent Totem – Scenarios

- The objective of the **final demonstrator** to be presented in the last year of the FRACTAL project, concern the evaluation of the TOTEM in the following two configurations:

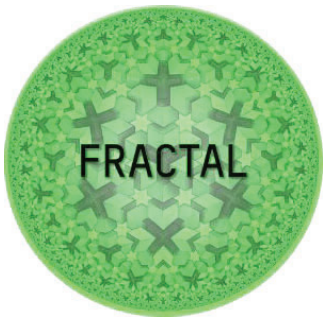
- **Scenario 1**: Crowded environment with one person approaching the TOTEM:
  - No more than a single person can approach the totem at the same time.

- **Scenario 2**: Crowded environment with multiple people approaching the TOTEM:
  - The personalized advertisement provided by the totem can take into account multiple users.

© 2020 FRACTAL. All rights reserved

THANK YOU

IKERLAN
P.º José María Arizmendiarrieta, 2 - 20500 Arrasate-Mondragón
T. +34 943712400 F. +34 943796944



Embedded reliable HPC Integration enabling smart Automotive & IoT Systems

Prof. Dr.-Ing. Dr. h. c. Jürgen Becker
Institut für Technik der Informationsverarbeitung - ITIV
Karlsruhe Institute of Technlogy - KIT, Germany
becker@kit.edu

Abstract*— The field of reliable embedded electronic HPC systems, incl. heterogenous multi-core (MC), accelerator and cyber-physical systems (CPS) integration, is emerging, essential and challenging for actual and future silicon technologies. Here, on one side increasing monolythic integration of cooperating computational and physical elements is necessary, e. g. in embedded and smart on-demand automatized environments as diverse as space, avionics, automotive, chemical processes, civil infrastructure, energy, healthcare, manufacturing (Industry 4.0), transportation, entertainment, and communication/consumer appliances. Moreover, future complex dynamically adaptive and multi-domain electronic system integration tends to be more and more dependent on distributed real-time and embedded High-Performance Computing (HPC) availability. This results in the strong demand of newly operating decentralized/centralized intelligent, interconnected, and silicon-based technology integrated solutions, subject to increased performance needs to facilitate computationally intensive algorithms, power consumption to be minimized for mobile devices, and sufficient degrees of reliability and verifiability to employ digital systems in safety-critical environments. Existing technology must evolve in order to meet such high requirements, whereas open source hardware and artificial intelligence (AI) could also play key roles in this process. Multipurpose adaptivity and reliability are crucial, especially in scaling down silicon technologies according to Moore for future processor technologies incl. dynamically reconfigurable accelerators and embedded HPC architectures. This requires in addition new approaches for parallel programming, reference technology platforms (RTP) and correct design and deployment methodologies incl. standardized tool flows. The contribution will discuss challenges of high-end heterogenous multi-core/accelerator HPC platforms incl. their reliable SoC hardware/software integration, e. g., smart scalable & embedded Automotive & IoT systems towards future automatizable environments like autonomous driving, drones and robotics scenarios etc.

Keywords- embedded computing, automotive systems, reliability, performance, HPC, AI

References *(National & European Project Initiatives):*

[1] *ARAMiS* (Automotive, Railway and Avionics Multicore Systems) https://www.aramis2.com
[2] *EPI* (European Processor Initiative) https://www.european-processor-initiative.eu
[3] *XANDAR* (Safety and Security for Embedded Software Systems) https://xandar-project.eu
[4] *CeCaS* (Central Car Server) https://www.infineon.com/cms/en/about-infineon/press/press-releases/2023/INFXX202302-070.html

About author



Jürgen Becker *is a Professor for embedded electronic systems and Head of the Institute for Information Processing Technologies (ITIV) at the Karlsruhe Institute of Technology (KIT) since 2001. From 2005-2009 he has been appointed as Vice Rector for Study and Teaching at Universitaet Karlsruhe (TH) and as Chief Higher Education Officer (CHEO) at KIT from 2009-2012. Since 2012 till 2014 he served as Secretary General of CLUSTER, an association of 12 leading Technical Universities in Europe. His research interests include Hardware/Software System-on-Chip (SoC) Integration, Cyber-Physical Systems (CPS), Heterogenous Multicore (MC) Architectures and High-Level Synthesis & Design Methods, Reconfigurable Computing and Accelerator & AI Integration, with application in Embedded Systems (Automotive, Industry 4.0, Avionics, HPC Scientific Applications incl. Physics Detector Experiments). He authored over 400 papers in peer-reviewed international journals and conferences, as well as 30 patents. Prof. Becker is active in numerous international conferences, as Chairman in TPC & Steering Committees, e. g. IEEE ISVLSI, IEEE SOCC, RAW, FPL, PATMOS, IFIP VLSI-SoC, DATE, SBCCI, ARC, FCCM, FPT, among others.*

ANDANTE

AI for **N**ew **D**evices **An**d **T**echnologies at the **E**dge



CPS&IoT'2023 Summer School on Cyber-Physical Systems and Internet-of-Things

Mario Diaz Nava

STMicroelectronics Grenoble, France

1



Outline

1 Data Analytics : Cloud or/and Edge

2 What is the ANDANTE Motivation?

3 What is the ANDANTE Project?

4 ANDANTE Work Plan and Activities

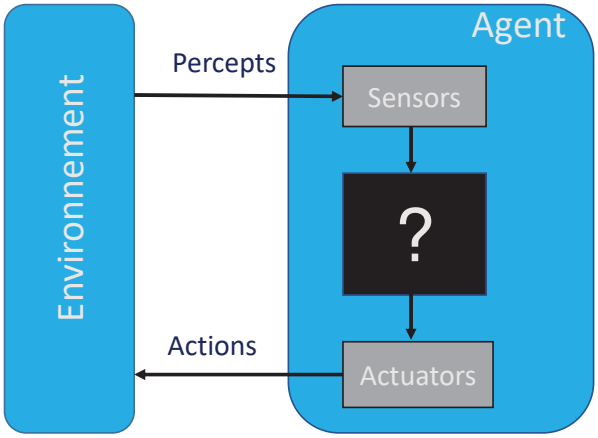
5 Take-Away

Data Analytics : Cloud or/and Edge



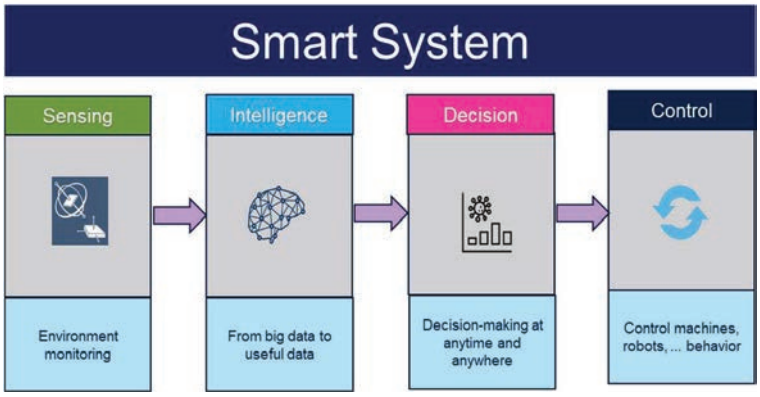
Agent, IOT & CPS

- **Agents** interacts with **environments** through **sensors** and **actuators**
- **IoT** (Internet of Things) enables to **monitor devices to take decisions and optimize actuators** to generate results
- **CPS** (Cyber-Physical Systems) consists of **computation and control components** tightly **combined with physical processes**



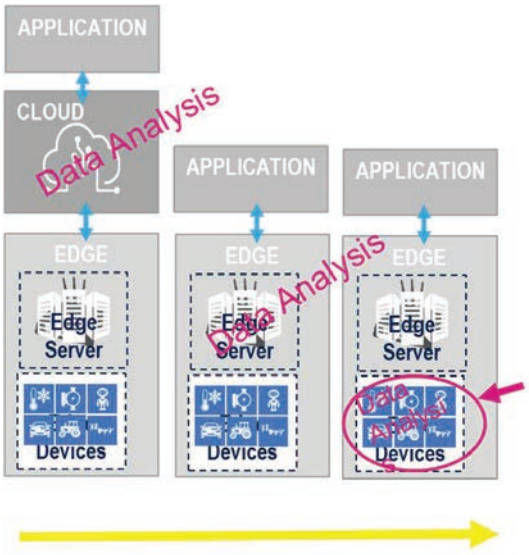
AI and IOT & CPS

- IoT and CPS are evolutions of existing monitoring / control systems
- IoT and CPS are a revolution for the applications in multiples areas
- IoT and CPS bring higher system flexibility, accuracy, precision, distributed computing power, … at cost effective



Data Analytics from Cloud to Edge

- Nowadays, **AI** is strongly penetrating large **market** segments
- **Data analysis** is moving **from** the **Cloud to** the **Edge** creating new opportunities for the European industry
- At the **Edge**, innovative, efficient, and cost-effective **IC solutions** are required



What is the ANDANTE Motivation ?



IoT & CPS System Requirements

Technologies

- Multi-sensors and actuators, robots, UVA, satellites
- Efficient Edge / Cloud computing capabilities
- Data analysis: AI, Machine Learning, Deep Learning
- Security and privacy
- Adapted connectivity
- Low-power consumption
- Reliable technology
- Computer vision
- Tools for: Modeling, data analysis, SW development, prediction, system debug and validation

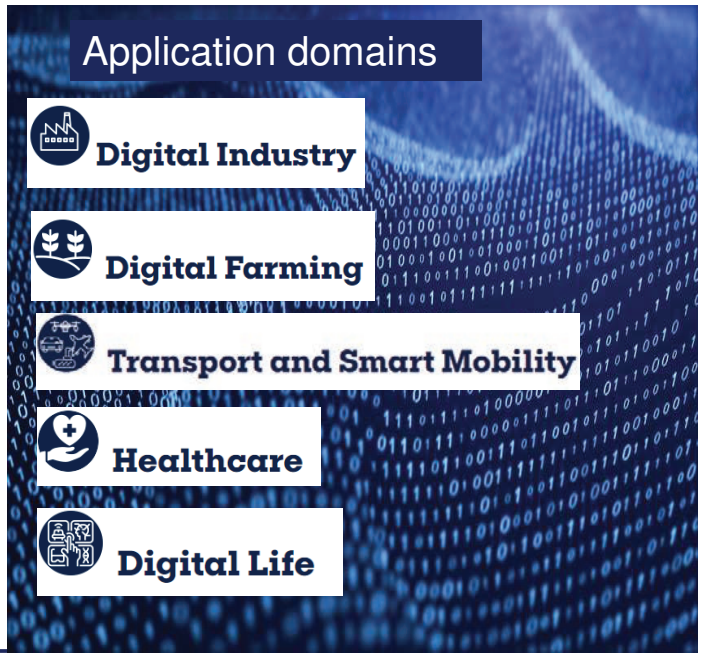
Applications

- Autonomy
- Large coverage
- Integration
- Cost
- Robustness
- Reliable
- Easy to use
- Availability
- Flexibility
- Support and Maintenance

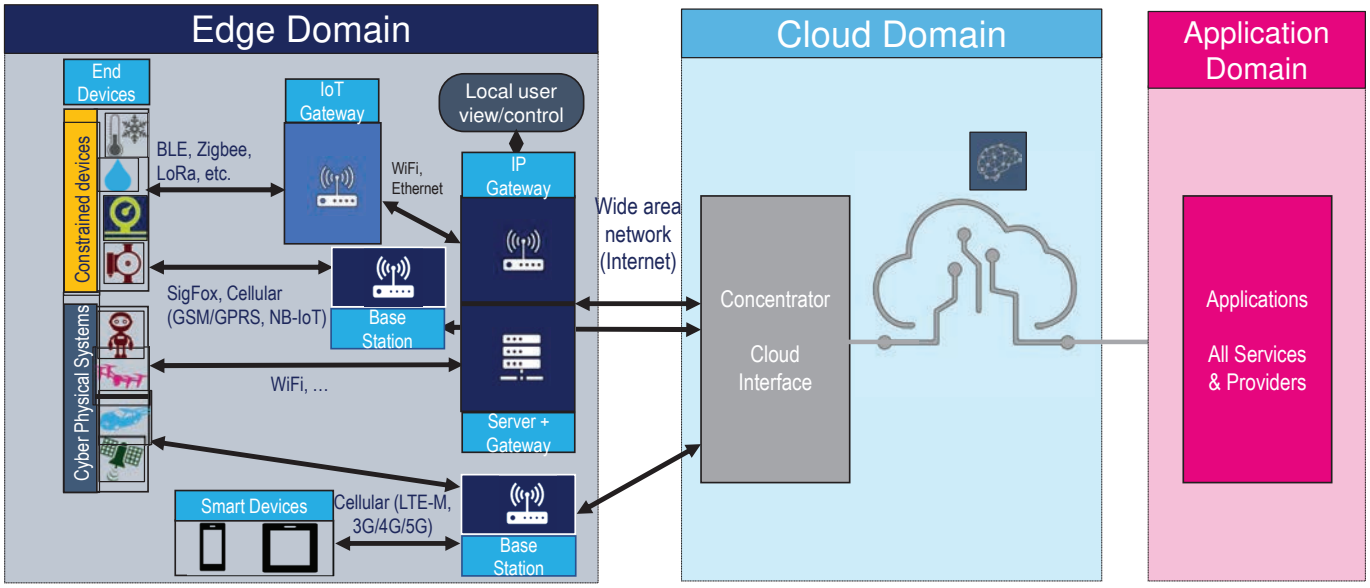
- Digital Twin for prediction: maintenance, production, ….

Provide Efficient AI Solutions

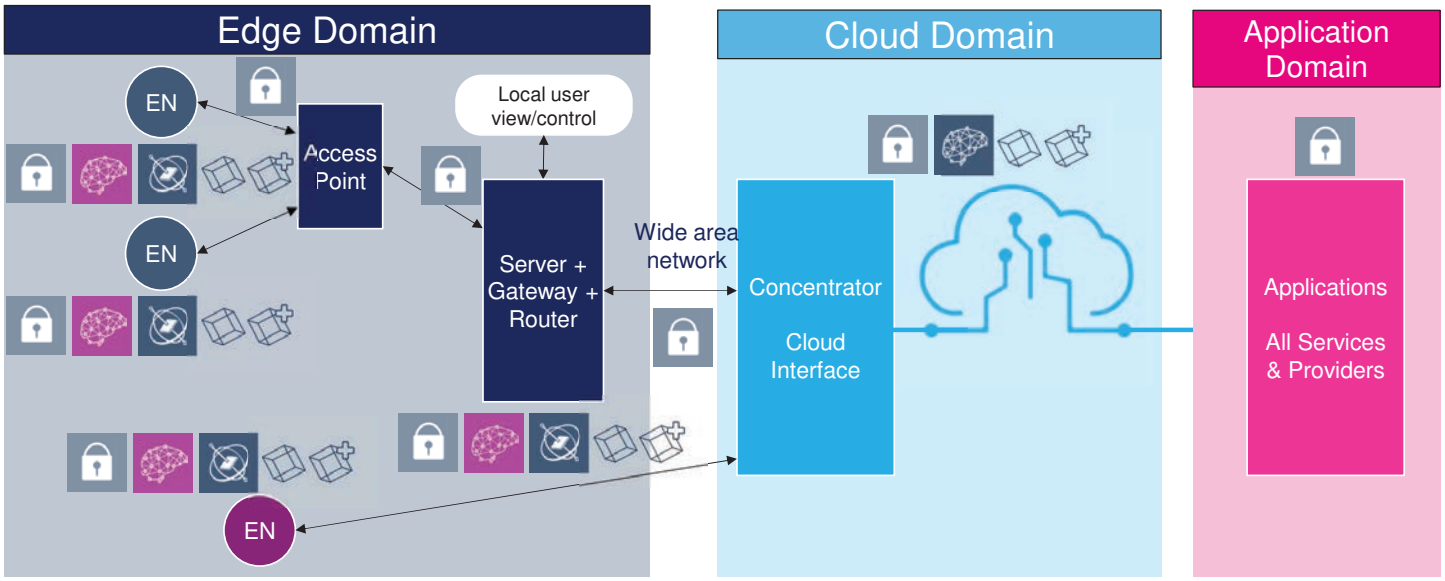
- Digitalization brings many advantages in many application domains
- An important **drawback** is the **electricity consumed** with the use of **cloud computing solutions**
- Bloomberg estimated that about **1% of the world's electricity** goes to cloud computing and for 2030 will increase **conservatory to 8%** to power the future cloud
- New solutions should be found at all levels to remediate this unsustainability



IoT Reference Architecture



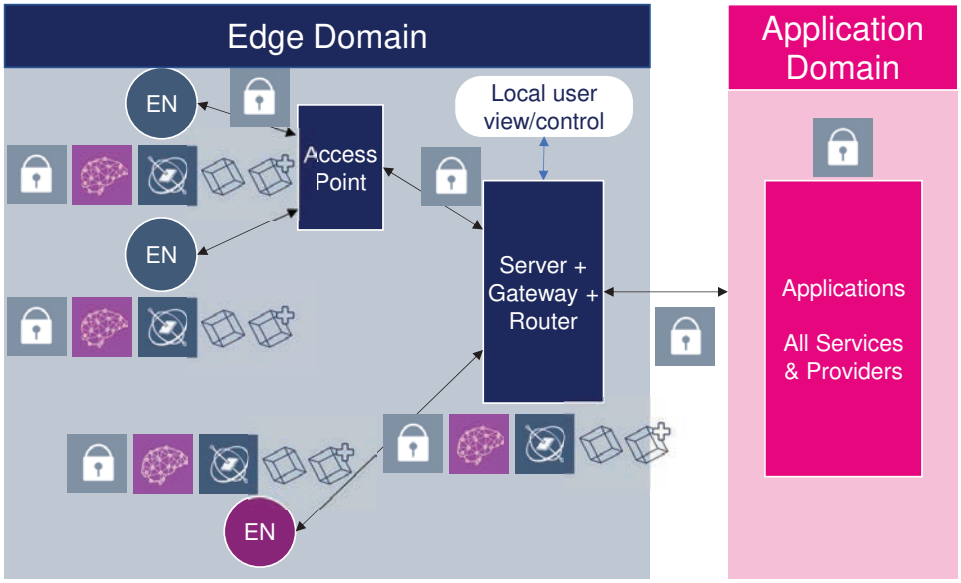
IoT & CPS System evolution



Move from Cloud to Edge

Benefits

- Reduced latency
  → Real-time applications
- Reduced bandwidth
- Higher cyber-security and privacy
- Lower power consumption
- Local Private Network

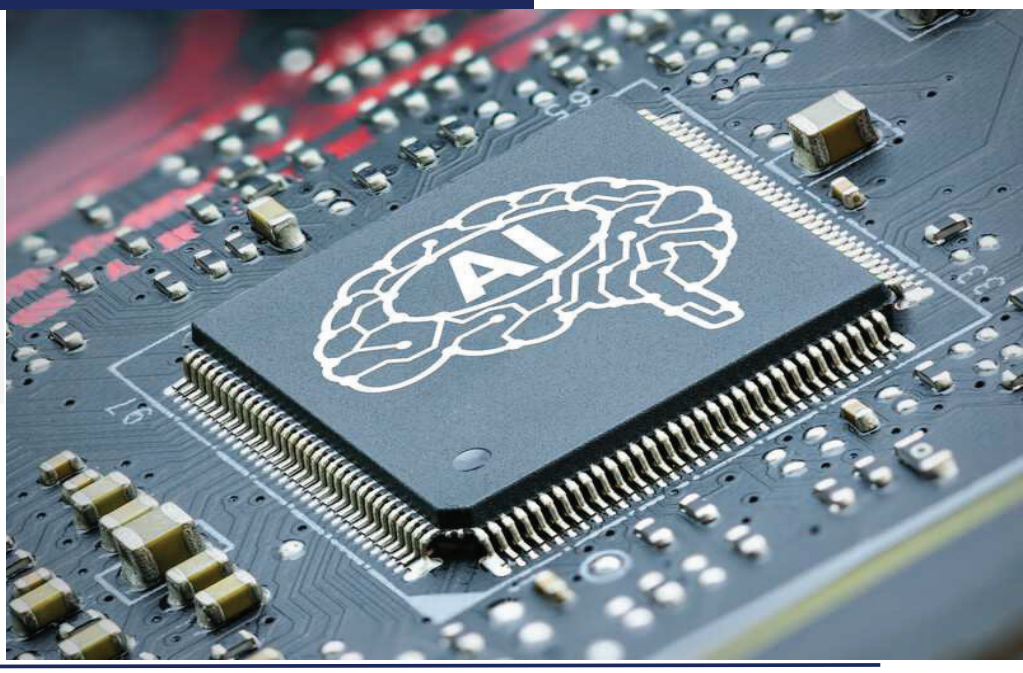


WHAT is the ANDANTE Project?



Main goal

To leverage innovative **IC designs** to build powerful **HW&SW platforms** for **artificial and spiking** neural networks



Main Activities

Emerging eNVM memories	Tools & Methodologies	Neuromorphic ASICs & FPGAs	AI Platforms & Applications
• OxRAM, • PCM • FeFET, • SOT-MRAM	• SW-HW co-design • Training, profiling and mapping a neural network on a HW target	• SNN and ANN architectures • Digital, mixed-design strategies	- Measure **KPI** - Validate, and evaluate **solutions pertinence**

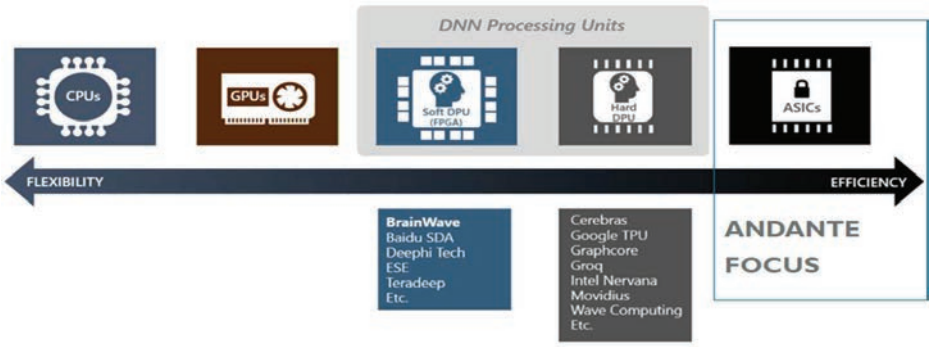
ANDANTE Edge Application Areas

Innovative solutions in the Edge with **strong Market Impact**



Main Focus

Highly efficient artificial and spiking neural network **ASICs and FPGAs** to perform inference, classification, ...



ANDANTE Consortium Members



ANDANTE facts and figures



ANDANTE Value Chain

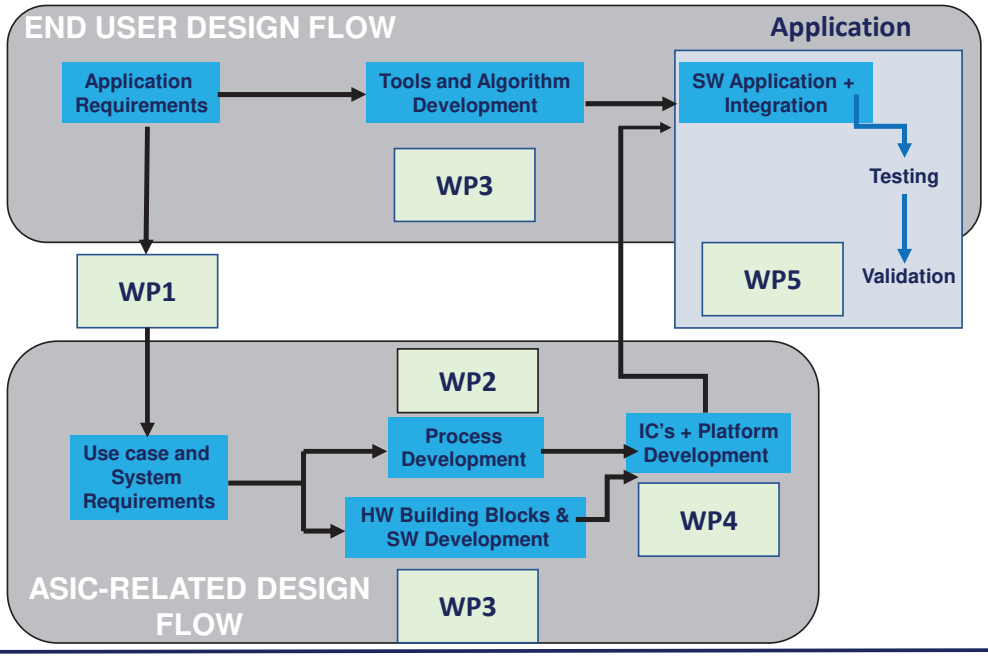
Complete value chain covered from technology formulation towards system realization



ANDANTE Work plan and activities



Project Structure



22 ANDANTE: CPS & IoT SUMMER SCHOOL BUDVA, MONTENEGRO, JUNE 6th -10th , 2023

Applications



Digital Industry



UC1.1: Indoor Positioning, Recognition and People Counting

Description: Indoor positioning recognition and people counting for smart laboratory/factory applications (e.g., robot co-working)

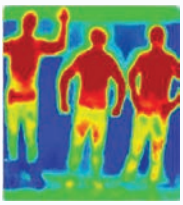
Challenges:

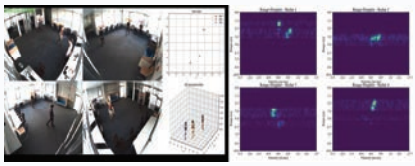
- Real time computation
- Scaling to handle input of multiple sensors

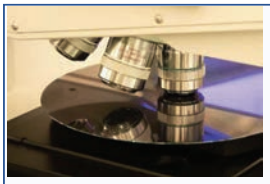
Positioning vs State-of-the-Art:

- Existing solutions are too slow to fulfill the safety requirement of a smart factory/laboratory

Partners: IFAG, EESY, TUD, FHG, HEI







UC 1.2: Color Classification at the Edge for Quality Control

Description: Integration of simple neuronal networks for color classification into color sensor nodes for real time control and monitoring devices at the edge

Challenges:

- Size of the AI solution
- Energy consumption and cost of the overall solution

Positioning vs State-of-the-Art:

- Existing solutions are too big, consume too much energy and are too costly to be integrated into color sensors

Partners: IFAG



Digital Farming



Partners: Bordeaux-INP, CEA and STGNB



UC2.1: Autonomous Weeding System

Description:

- Crops and weeds detection
- Intra-row weeding
- Mechanically: alternative to chemical weeding
- Autonomous: limiting human intervention

Challenges:

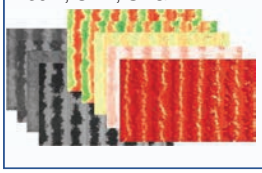
- Computation in real-time
- High precision needed to differentiate crops and weeds

Positioning vs State-of-the-art

- To date, for most crops, only inter-row mechanized/autonomous weeding solutions exists



Partners: CCTI, Italagro, TPRO-Tech., CEA, STGNB









UC2.2: Tomato pests and diseases forecast

Description: Pest and disease detection model for the tomato agriculture industry.

Challenges:

- Collecting enough and good quality data
- Time-series analysis for forecasting
- Model selection and development for edge devices

Positioning vs State-of-the-art

- State-of-the-art solutions do not resort to image analysis with ANNs, which produce many false positive events.

Transport and Smart Mobility



Partners: Thales, CEA, STGNB



UC3.1: Drones/USV

Description: detection, classification and segmentation of high-altitude images using either ANN, SNN or hybrid technology

Challenges:

- Real time computation
- High resolution inputs
- Power consumption

Positioning vs State-of-the-Art:

- Existing solutions are not compatible with drone constraints



Partners: ALSEAMAR, CEA, STGNB, Synsense



UC3.2: Underwater Acoustic Signal Classification

Description: The ocean soundscape is a continuously changing mosaic of sounds that originate from various sources. This is of primary interest to recognize in real time the components of the soundscape. Study on marine mammals classification.

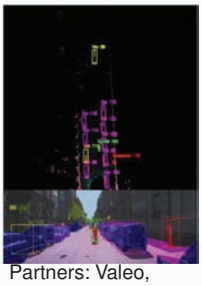
Challenges:

- Real time computation
- Low power

Positioning vs State-of-the-Art:

- Long-term monitoring
- Real time communication with shore

Transport and Smart Mobility



Partners: Valeo, CEA, STGNB, UZH



UC3.3: 3D Object Detection and Classification of Road Users based on LiDAR and camera

Description: Object detection on lidar point clouds that will be fused with camera semantic image data implemented on ANDANTE Platform

Challenge:

- Real time computation with received sensor data (bandwidth bottlenecks)
- Scaling to handle input of multiple sensors

Positioning vs State-of-the-Art:

- No fusion of both sensors/sensor data yet done on neuromorphic hardware



Partners: BR&T-E, Gradiant, TVES, CARTO



UC 3.4: Robust Autonomous Landing

Description: Four critical functionalities are considered: 1) image-based runway relative localization for navigation, 2) image registration for navigation, 3) foreign object detection on runway, 4) robust communications.

Challenges:

- Real time computation
- Adaptation of large networks to efficient hardware without sacrificing performance levels
- Learning on the edge

Positioning vs State-of-the-Art:

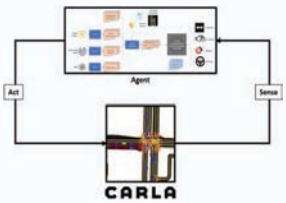
- Many smaller aircraft cannot permit energy cost of large number of conventional AI algorithms in standard hardware, which on the other hand, are necessary to enable autonomous operations.

Transport and Smart Mobility



Partners: GML, Valeo





UC3.5: Path Planning for Autonomous steering

Description: Continuously calculate trajectories, based on deltas in world map, avoid intersects, while optimally steering the vehicle, e.g., by solving many diff equations

Challenges :

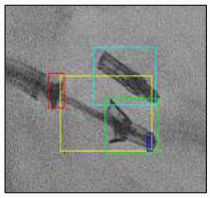
- 100Hz world map update rates and <10ms latency ➔ huge comp. loads
- Small form-factor (<200x200mm) and power consumption (<2W)

Positioning vs State-of-the-Art:

- Reduce power by 100x over SotA
- Reduce latency by 10x over SotA

Healthcare





UC 4.1: Multi-modal image processing and device tracking in medical X-ray & ultrasound images

Description: Navigating a medical device like a Mitral Clip to the right location in the heart is challenging and requires accurate and intuitive image guidance.

Challenges:

- Detection accuracy → placement of device
- System latency → eye-hand coordination

Partners: PMS, imec-NL

Positioning vs State-of-the-art

- Benchmark the SNN version of the detection algorithm against an implementation in state-of-the-art GPU HW





UC 4.2: Ultrasound acquisition or processing

Description: Lung ultrasound can detect healthy/unhealthy (Covid-19, Pneumonia) patients. We create a neural network to automatically detect healthy/unhealthy lungs

Challenges:

- Make NN fit on the two platforms

Positioning vs State-of-the-Art:

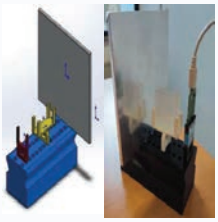
- Automatic detection
- Need low power consumption for mobile ultrasound

Partners: PRE, imec-NL and GML

Healthcare



Partners: EESY, IFAG



Use Case 4.3: Glucose Monitoring

Description: Apply SNN algorithms to high-frequency sensor data to classify different glucose level in water dilutions.

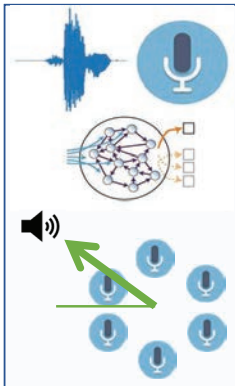
Challenges:

- To distinguish standard human body glucose concentrations

Positioning vs State-of-the-art

- Leverage the faster computation of the SNN to reach accurate results of the glucose level using less time and energy resources.

Digital Life



UC5.1: Consumer Auditory Processing

Overall Challenges:

- Continuous real time computation
- Low-latency, low-power requirements

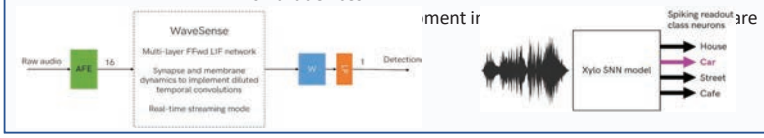
5.1a: Continuous audio scene classification. The audio environment is monitored continually, to assist in selecting a noise reduction scheme appropriate for the environment.

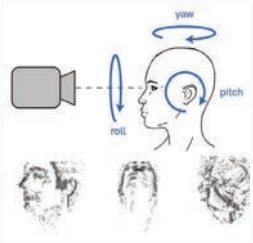
5.1b: Audio event detection: Continuous monitoring of audio signals for pre-defined trigger events, such as glass break (for security purposes) or distress call (for health monitoring purposes).

5.1c: Multi-microphone auditory processing: A low-power sensory processing task designed for multiple simultaneous input channels (i.e., a microphone array), to assist in noise reduction in smart home devices.

5.1d: Voice Activity Detection: Monitoring of audio scene for detection of voiced speech which acts as a wake up signal for smart devices.

Partners: Synsense, UZH, FHG, STGNB





UC5.2: Vision -based human computer interaction application

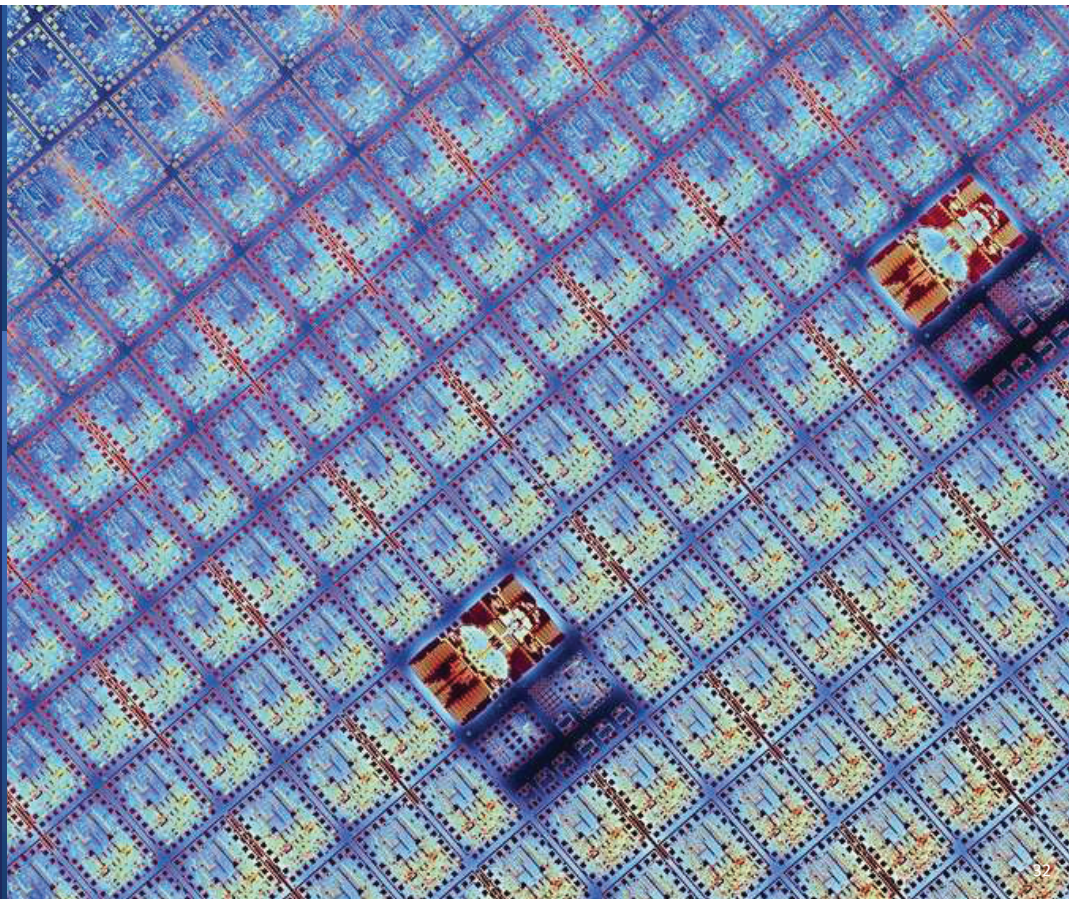
Description: Glance detection for mobile hand-held devices, to be used as a smart wake-up trigger

Challenges:

- Low-latency and low-energy requirements
- Low false-reject rate requirements

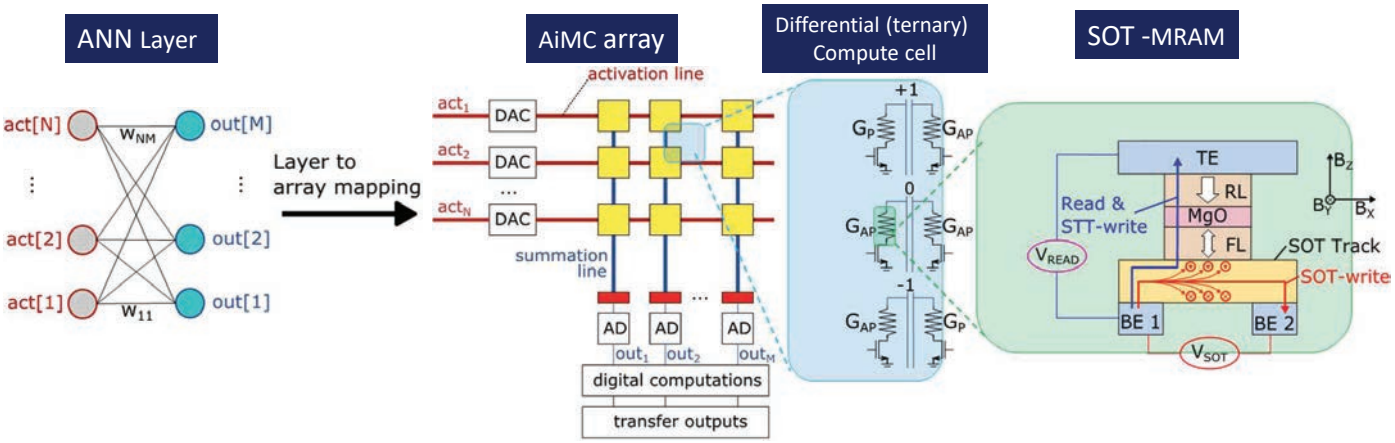
Partners: Synsense, UZH, CSEM, STGNB

New Memory Technologies for AI Applications



eNVM Features

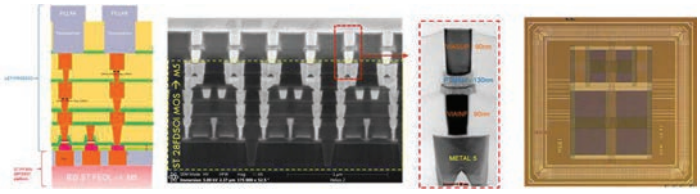
- Matrix vector multiplications (MVMs).
- Minimizing the data movement through IMC (In Memory Computing techniques) and high storage density



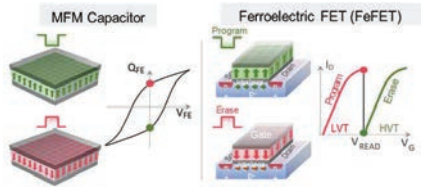
eNVM: OxRAM, PCM, FeFET, SOT-RAM,

- New low-power architectures for edge AI applications.
- New memory technology options for In Memory Computing (IMC) and high storage density.

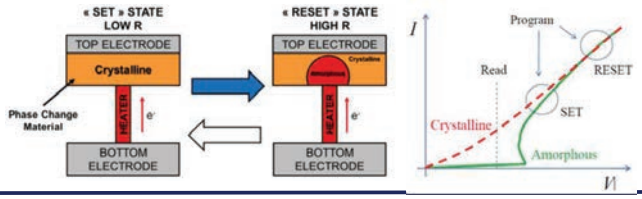
Large scale integration in OxRAM



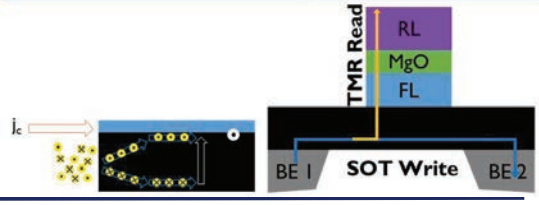
High endurance of Ferroelectric FET (FeFET)



Large scale integration Phase Change Memory



"Unlimited endurance" Spin-Orbit Torque MRAM

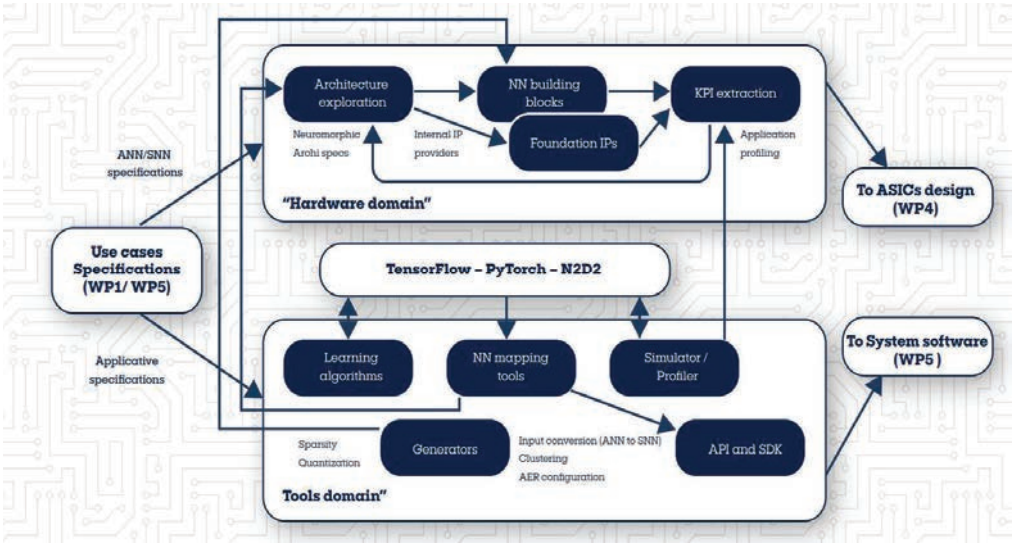


Tools and Methods for training, profiling and mapping NN on a HW target

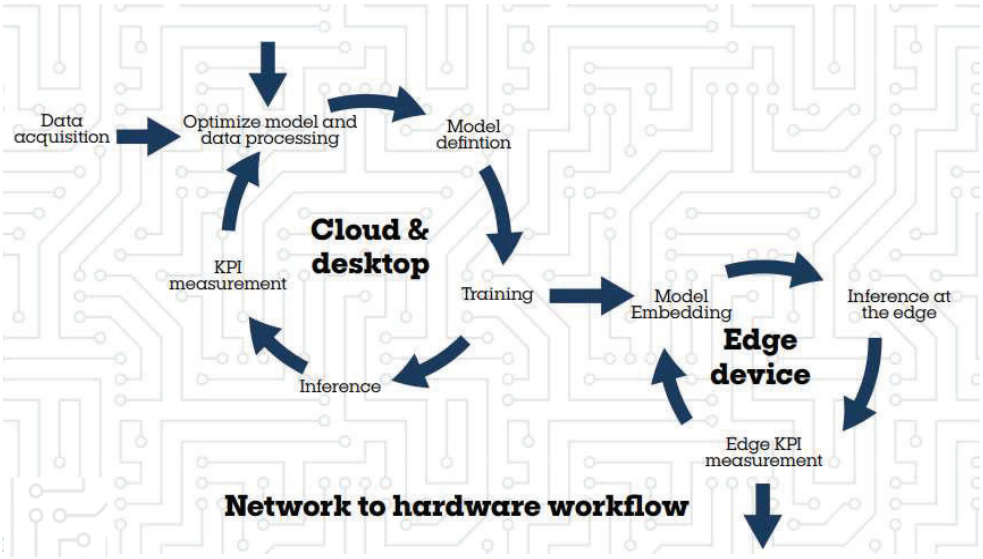


Design Challenges of Edge AI processors

SW-HW co-design methodology pursued in ANDANTE



Workflow for embedded neural networks



Application dependent high-level design choices

What coding style to use?

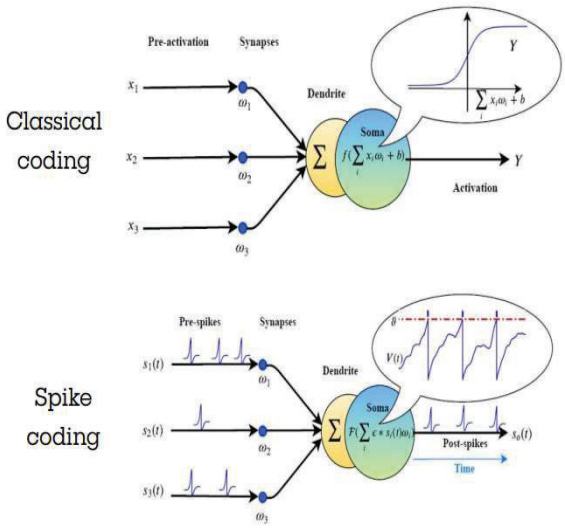
- Classical for still inputs (e.g., images)
- Event / Spike for always-on, low-activity inputs (e.g. surveillance)
  - o Rate code for "Frame-type" inputs
  - o Temporal code is best on Temporal series (e.g. audio)

What network topology?

- A deep, feed forward for high accuracy and fixed inputs applications
- A shallow, recurrent for temporal series (e.g., audio)

What implementation strategy?

- A fully digital for very deep, large, networks
- An analog mixed-signal for shallower networks



Tools

Training Tools

Spike-Coding

- ANN-to-SNN conversion
- Direct SNN training (offline and On-chip learning)

Classical coding

- Transfer learning
- Quantization Aware (QAT) Training: reduces memory footprint and enables the use of embedded NVM

HW aware training (HAT) for mixed-signal implementations of ANN ASICs

- Takes non-idealities into account during the training phase

Generation, Mapping and Simulation Tools

Hardware generator

- Writes RTL (for digital NN)
- Generates schematic and layout (for analog NN)

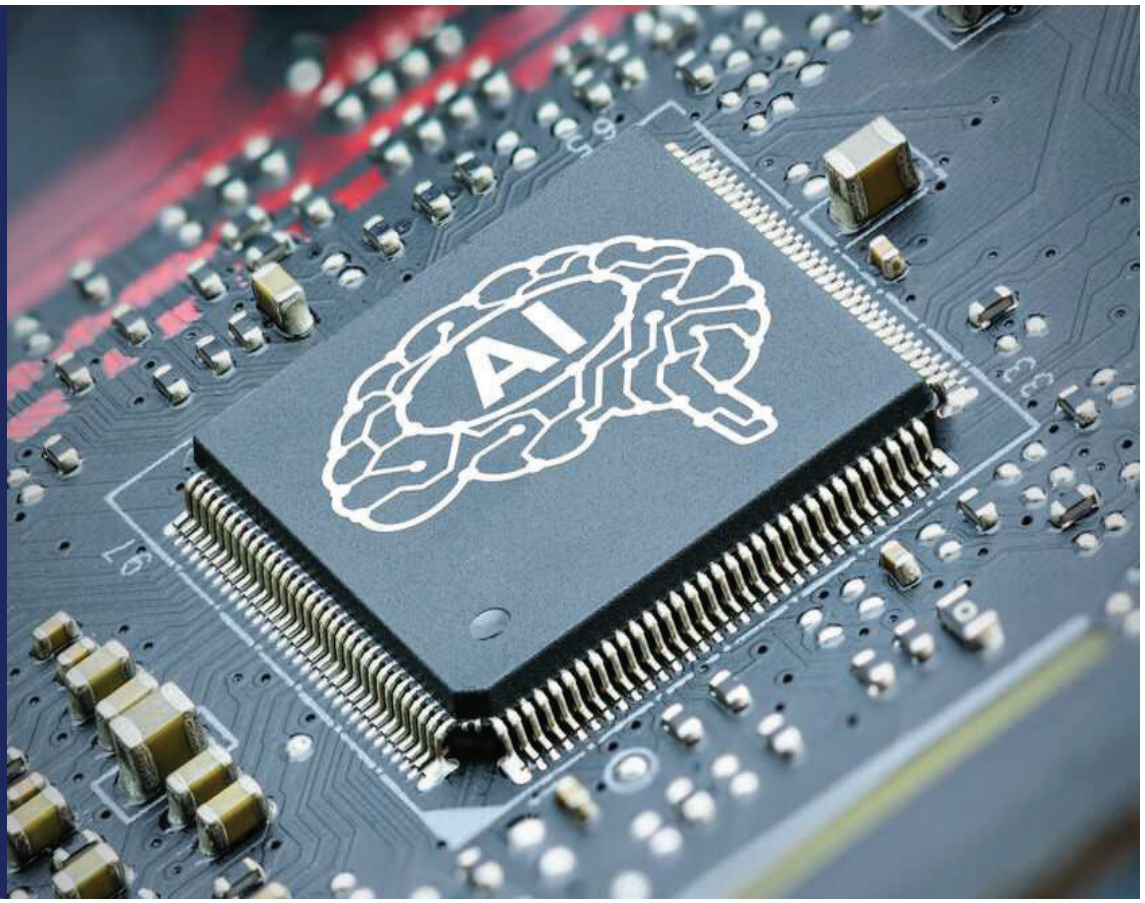
Neural Network Mapper on Hardware

- Maps onto Processing engines or NVM arrays

Simulation

- Estimates power dissipation

Design of Neuromorphic SoCs, ASICs and FPGAs, and Platforms



Objectives and Challenges



- Design **inference accelerators** based on **analog, digital and spiking** neural neurons
- **ASICs, SoCs, and platforms** for **5** application domains
- Performance evaluation through **14 use cases**



- Managing several **design types** in advanced technologies
- Managing efficiently HW/SW **co-design tools** of neuromorphic hardware
- Managing the support of **various applications in the same platform**

Design of Edge AI processors (1/2)

8 ASIC Designs

Designs

- 2 Mixed-signal SNN
- 1 Digital ANN (inference accelerator)
- 1 Digital CNN (smart vision)
- 3 Mixed-signal inference accelerators with in-memory computing
- 1 SoC (MCU + NPU (AI accelerator))

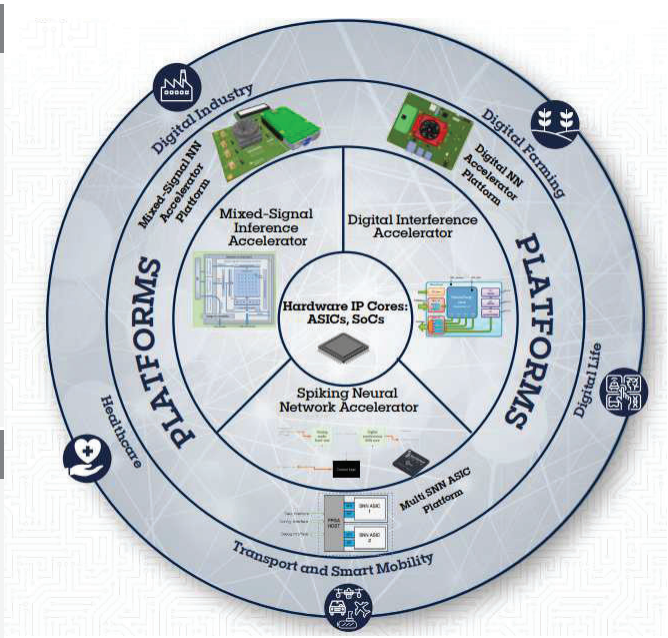
Benefits

- High energy efficiency
- Low latency

6 Platforms

Applications

- 5 Application areas
- Validation
- KPIs measurement



Design of Edge AI processors (2/2)

ANDANTE

6 FPGA Designs

Designs

- 16 SpiNNaker2 processing element
- Hybrid inference accelerator
- Neuromorphic accelerator, SENeCA
- Accelerated inference of DNN for runway detection, Yolo v4
- Accelerated inference of DNN for object detection and image registration, RetinaNet and VGG19
- Accelerated inference of DNN for process communication parameters MobilNet

Benefits

- High level of (HW/SW) programmability
- Design Flexibility
- Easy to update and Optimize

Xilinx Platforms

Applications

- 2 Application areas
- Validation
- KPIs measurement

VCU118



ZCU102



ZCU111



Take-away



ANDANTE contribute to reach Green Deal goals



- New **eNVMs** provide very efficient solutions
- Data analysis at the **Edge** bring the intelligence close to data sources
- **ASICs and FPGA HW** architectures (based on SNN, ANN) provide very efficient solutions
- Tools and Methodologies developed simplify the **HW/SW co-design**
- Cost effective and more **intelligent and autonomous CPS and IoT** devices



- Handling efficiently **data analytics at the Edge**
- **Reduce power consumption, latency, data storage**
- **Increase the intelligence and autonomy of CPS and IoT Systems**

ANDANTE

AI for New Devices And Technologies at the Edge

andante-ai.edu | linkedin.com/company/andante-ai | mario.diaznava@st.com (coordinator)

"This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement № 876925. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Belgium, Germany, Netherlands, Portugal, Spain and Switzerland"

ECSEL Joint Undertaking

Scan Me
to visit website

Thank You

46



X-by-Construction Design Framework for Engineering Autonomous and Distributed Real-time Embedded Software Systems

Tutorial: Simulation-based development of networked avionics systems using the XANDAR toolchain

Tobias Dörr[1], Florian Schade[1], Alexander Ahlbrecht[2]
[1]Karlsruhe Institute of Technology (KIT)
[2]German Aerospace Center (DLR)



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

Agenda

- **Part I:** The software development methodology of XANDAR
  - Motivation + XANDAR project overview
  - Software development approach
  - Simulation and target deployment
  - Safety pattern concept + two sample patterns
- **Part II:** Programming models for concurrent systems
  - Fundamentals + selected programming models
  - Applicability to the XANDAR toolchain
- **Part III:** Live demonstration of XbCgen
  - Overview and introductory examples
  - Application to an avionics use case

Part I

The software development methodology of XANDAR

Motivation

- Designing embedded systems is an error-prone task
- **Flight 501 of Ariane 5 (1996):**
  - Programming error in the control software [1]
  - Failure of the Inertial Reference System during test flight
  - Self-destruction after 37 seconds
- **Remote access to 1.4 million road vehicle (2015):**
  - Vulnerability in the vehicle's head unit [3]
  - Remote reprogramming of a gateway component
  - Full access to a CAN bus of the vehicle

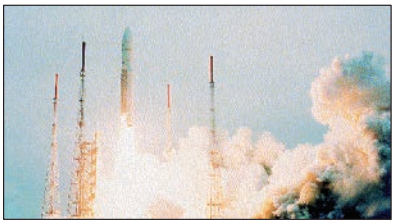


Image source: [2]



Image source: [3]

[1] B. Nuseibeh, **"Ariane 5: Who Dunnit?"**, IEEE Software, vol. 14, no. 3, May-June 1997.
[2] European Space Agency, Bulletin Nr. 89, online: https://www.esa.int/esapub/bulletin/bullet89/dalma89.htm, accessed 2 May 2023.
[3] C. Miller, "**Lessons Learned from Hacking a Car**", IEEE Design & Test, vol. 36, no. 6, December 2019.

Motivation

- **Properties of modern cyber-physical systems** aggravate the design challenge:
  - **Consolidation of functions on multicore platforms**, e.g. in the automotive domain
  - **Integration of Artificial Intelligence (AI) algorithms**, e.g. for object detection
  - **Functional safety and cybersecurity requirements**, e.g. fault tolerance
  - **Continuous interaction** with remote entities, e.g. in a cloud-edge setting
- **Conflicting requirements** are particularly difficult to address, for example:

Centralization of automotive on-board architectures [4]

⇒ Lack of physical separation between components
⇒ Potential for timing interferences, fault propagation, …
⇒ Functional safety issues

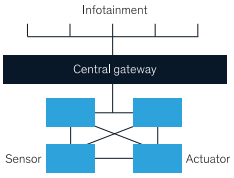


Image source: [4]

[4] O. Burkacky, J. Deichmann, J. P. Stein, "**Automotive software and electronics 2030: Mapping the sector's future landscape**", McKinsey & Company, 2019.

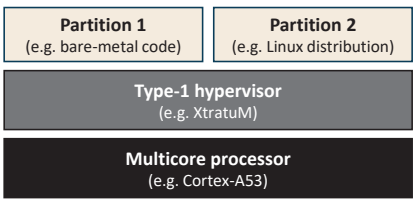
Motivation

- **Methodological gap** between requirements and implementations, e.g.:
  - ISO 26262 [5] specifies Freedom from Interference (FFI) requirements
  - A hypervisor such as XtratuM [6] is a building block that contributes to this goal
  - The correct application of such building blocks can be difficult (timing interferences, …)

ISO 26262 – Functional safety standard for road vehicles (initially published in 2011). Covers the following scope [5]: *"This document addresses possible hazards caused by malfunctioning behaviour of safety-related E/E systems, including interaction of these systems."*

Hypervisor – Software layer providing independent execution environments on a single processor. Hypervisors running directly on the target hardware (without a host OS) are called type-1 hypervisors.

Example of a hypervisor-based system

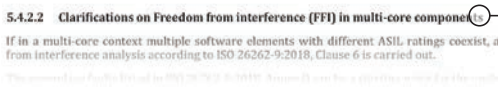


[5] ISO 26262-1:2018, **"Road vehicles — Functional safety — Part 1: Vocabulary"**, Geneva, 2018.
[6] M. Masmano, I. Ripoll, et al., **"XtratuM: a Hypervisor for Safety Critical Embedded Systems"**, 11th Real-Time Linux Workshop, 2009.

Motivation

- **Motivation:** How to apply such building blocks in a provably correct manner?

Excerpt of ISO 26262-11:2018


5.4.2.2 Clarifications on Freedom from interference (FFI) in multi-core componen(s)

If in a multi-core context multiple software elements with different ASIL ratings coexist, a freedom from interference analysis according to ISO 26262-9:2018, Clause 6 is carried out.

"Techniques such as hypervisors can help to achieve software partitioning […]"

X-by-Construction (XbC) – The *"step-wise refinement process from specification to code that automatically generates software (system) implementations that by construction satisfy specific non-functional properties concerning security, dependability, reliability, or resource/energy consumption, to name but a few"* [7].

- **Key idea:** Apply the X-by-Construction (XbC) paradigm to the design process
  - Auto-generate implementation artefacts that leverage suitable low-level techniques
  - Simplify the post-hoc verification process in a manner comparable to [8]

[7] M. H. ter Beek et al., **"X-by-Construction"**, Leveraging Applications of Formal Methods, Verification and Validation: Modeling, ISoLA '18, Springer, Cham, October 2018.
[8] B. W. Watson, D. G. Kourie, et al., **"Correctness-by-Construction and Post-hoc Verification: A Marriage of Convenience?"**, Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques, ISoLA '16, October 2016.

Overview of the XANDAR project

https://xandar-project.eu

Duration:
01/2021 – 12/2023

Budget:
€ 4.96 million

Project coordinator:
Prof. Jürgen Becker (KIT)

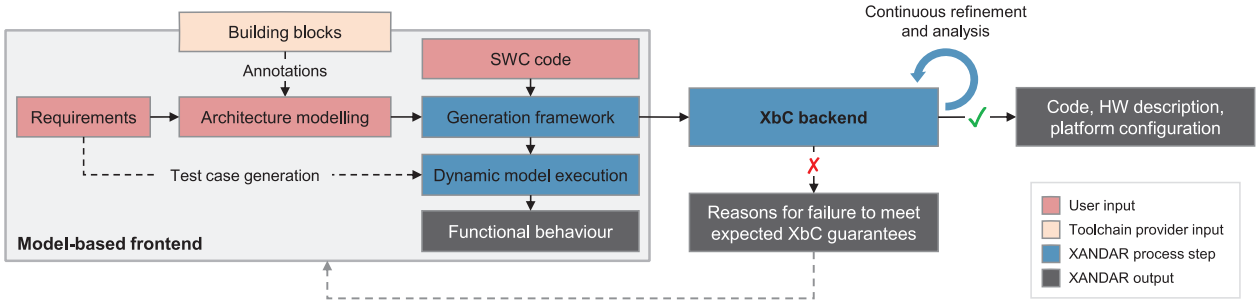
Scientific coordinator:
Prof. Nikolaos Voros (UoP)

Goal: Deliver a mature software toolchain that uses the X-by-Construction paradigm to generate system implementations with guaranteed properties

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

Overview of the XANDAR project

- The toolchain allows users to apply the XANDAR development process [9]
- This process defines a sequence of **iteratively traversed steps**:

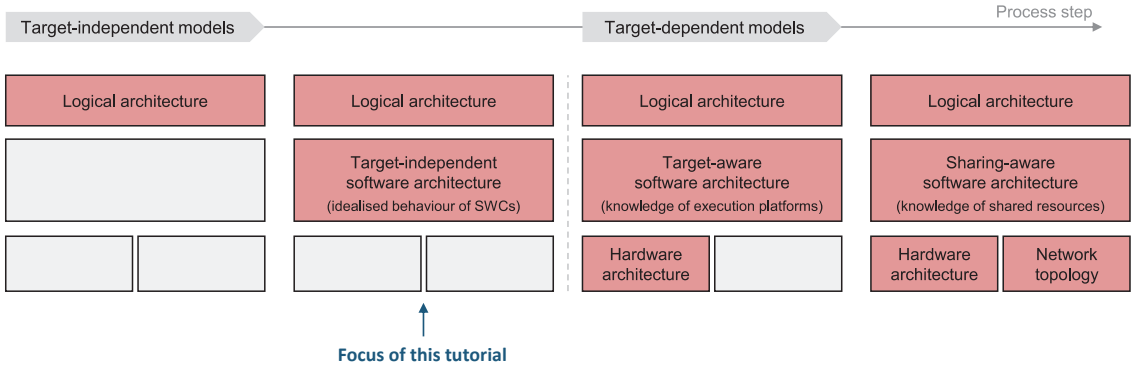


Acronyms: XbC = X-by-Construction, SWC = Software Component, HW = Hardware

[9] L. Masing, T. Dörr, et al. **"XANDAR: Exploiting the X-by-Construction Paradigm in Model-based Development of Safety-critical Systems"**, DATE '22, March 2022.

Overview of the XANDAR project

- With every iteration, the architecture model is refined or extended
- Relevant abstraction levels inspired by the PREEvision [10] layer model
- **Evolution of an architecture model** along the development process:



[10] J. Schäuffele, **"E/E architectural design and optimization using PREEvision"**, SAE 2016 World Congress and Exhibition, 2016.

Overview of the XANDAR project

- **Toolchain evaluated** in a laboratory setup with two use cases:
  - DLR ⇒ Resilient Avionic Architecture for Urban Air Mobility (UAM)
  - BMW ⇒ Autonomous Systems with Integrated Machine Learning Applications
- **Logical architecture** excerpt of DLR's pilot assistance system for UAM [11]:

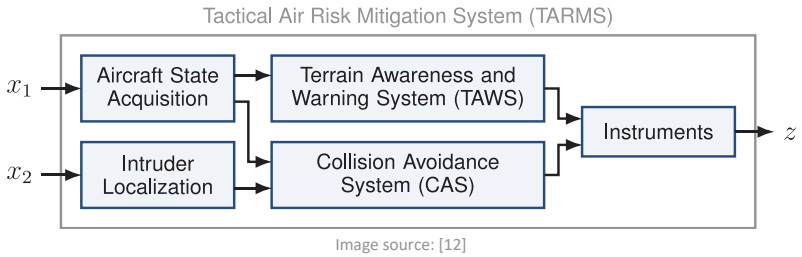


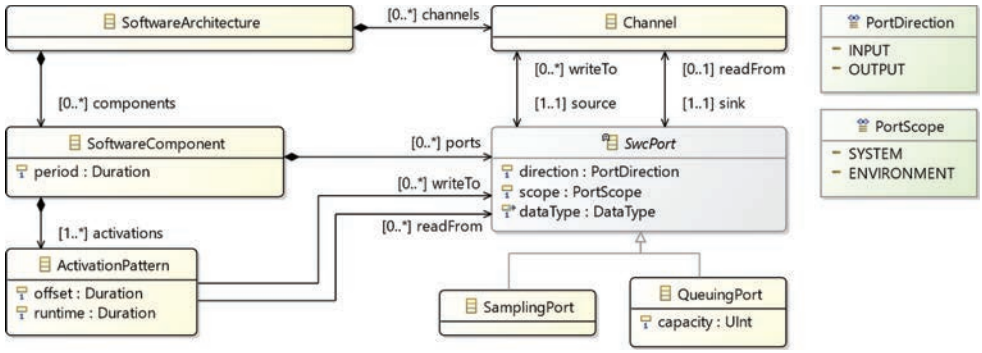
Image source: [12]



[11] J. Athavale, A. Baldovin, et al., **"Chip-Level Considerations to Enable Dependability for eVTOL and Urban Air Mobility Systems"**, DASC '20, October 2020.
[12] T. Dörr, F. Schade, et al., **"A Behavior Specification and Simulation Methodology for Embedded Real-Time Software"**, DS-RT '22, September 2022.

Software development approach

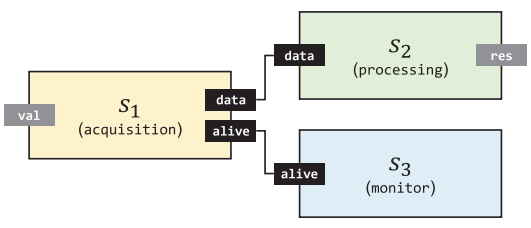
- **Description of the software architecture** by the toolchain user:
  - Software Component (SWC) entities communicating via message passing
  - Logical Execution Time (LET) parameters [13] capture the desired timing behaviour
- **Software architecture metamodel** as class diagram:



[13] T. A. Henzinger, B. Horowitz, C. M. Kirsch, **"Embedded Control Systems Development with Giotto"**, SIGPLAN Not., vol. 36, no. 8, August 2001.

Software development approach

- **Sample network of SWCs** and textual description in JSON5 [14] syntax:



JSON5 representation

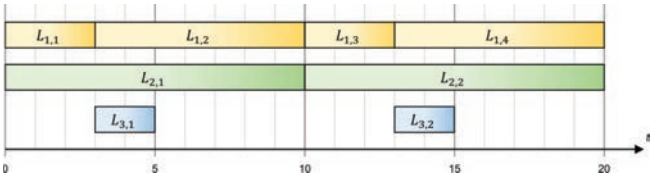
```
acquisition: {
    period: 10,
    activations: {
        check: {
            offset: 0,
            runtime: 3,
            write_to: [ 'alive' ],
        },

        gen: {
            offset: 3,
            runtime: 7,
            write_to: [ 'data' ],
        },
    },

    system_ports: {
        data: { /* … */ },
        alive: { /* … */ },
    },

    env_ports: {
        val: { /* … */ },
    }
}
```

- **LET frame sequence**:



[14] A. Kishore, J. Tucker, **"The JSON5 Data Interchange Format"**, v1.0.0, March 2018.

Software development approach

Logical Execution Time (LET) – "*LET determines the time it takes from reading program input to writing program output regardless of the time it takes to execute the program*" [15].

- **Generation** of SWC code skeletons:

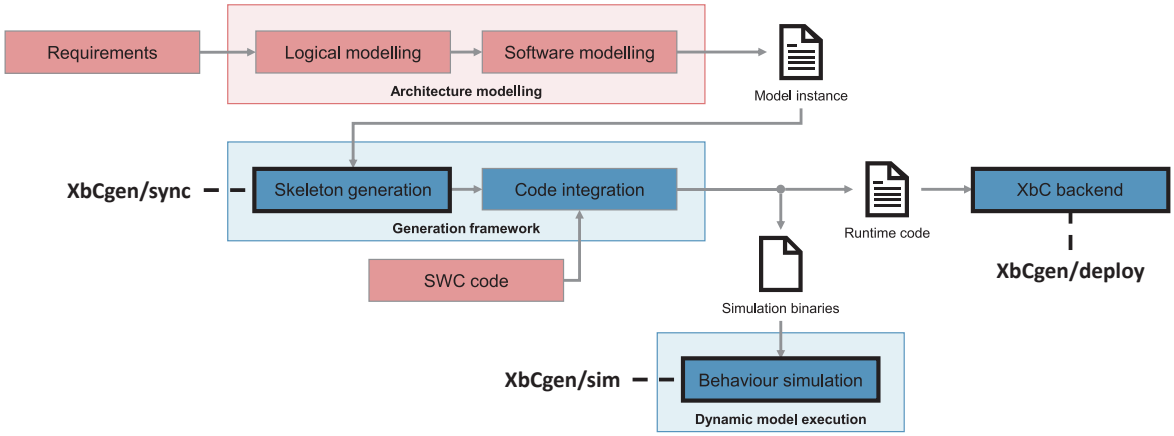
```
// Initialization hook:
void swc_init(void);
// Trigger hook:
void swc_trigger(enum activation_id activation, struct swc_port_map *port);
```

- **Access to input/output ports** via the framework-provided port map
  - Support for structured data structures (based on the Protocol Buffers [16] format)
  - Built-in support for access to input/output controllers of the hardware
- **On the target hardware:** invocation of the trigger hook for each SWC activation

[15] C. M. Kirsch, A. Sokolova, **"The Logical Execution Time Paradigm"**, Advances in Real-Time Systems, Springer, Berlin, Heidelberg, 2012.
[16] Google LLC, **"Google Developers: Protocol Buffers"**, online: https://developers.google.com/protocol-buffers, accessed 2 May 2023.

Simulation and target deployment

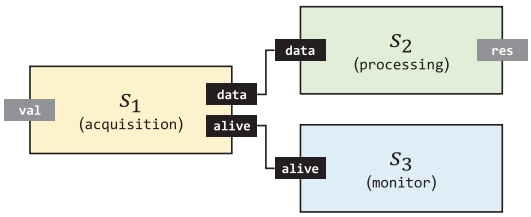
- **Starting point:** Fully deterministic specification of the software behaviour
- **Tool support** provided by the XbCgen framework of XANDAR:



Simulation and target deployment

- **XbCgen/sim:** Generation of execution traces for verification and validation
  - Discrete-event simulation with a user-provided plant/environment model (in Ptolemy II)
  - Automatic synthesis of an executable model (calling SWC simulation binaries)
  - Refer to previous work in [12] for a detailed description

- Sample excerpt of an execution trace:



⇒

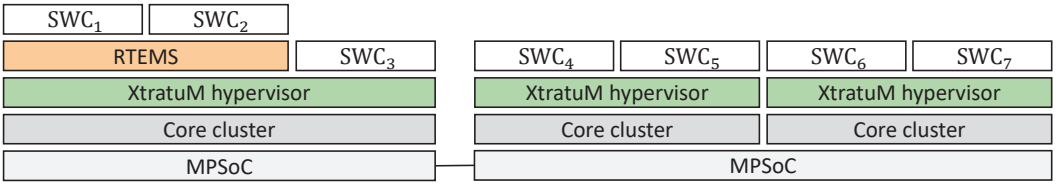
```
acquisition.gen(3..10ms)
 - data = [1566, 1564, 1563]
acquisition.check(10..13ms)
 - alive = true
monitor.main(13..15ms)
processing.main(10..20ms)
 - res = 1564.3333333333333
acquisition.gen(13..20ms)
 - data = [1558, 1557, 1555]
acquisition.check(20..23ms)
 - alive = true
monitor.main(23..25ms)
```

[12] T. Dörr, F. Schade, et al., **"A Behavior Specification and Simulation Methodology for Embedded Real-Time Software"**, DS-RT '22, September 2022.

Simulation and target deployment

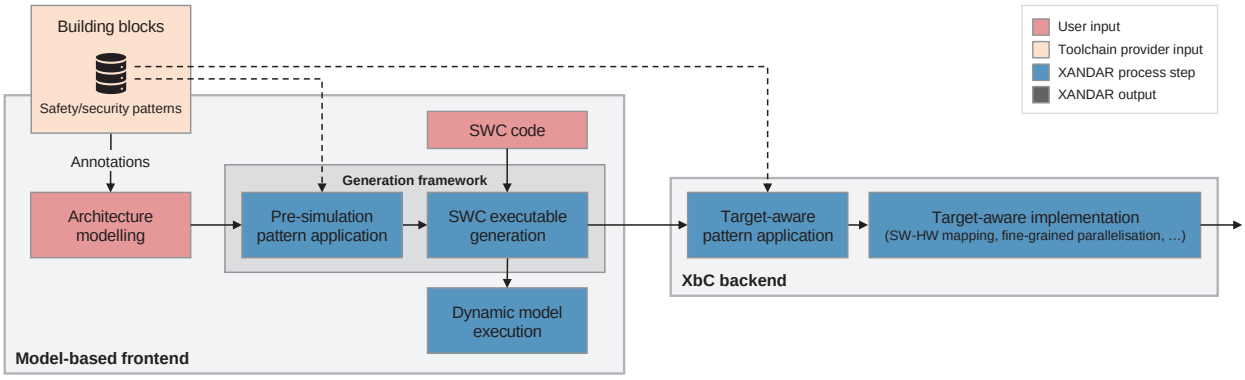
- **XbCgen/deploy:** Automatic generation of implementation artefacts for MPSoCs
- The target runtime environment consists of:
  - A hypervisor layer currently implemented using the XtratuM hypervisor
  - An optional operating system layer (RTEMS, FreeRTOS, Linux, …)

- SWCs are deployed to XtratuM partitions or OS tasks/processes, for example:



Safety pattern concept

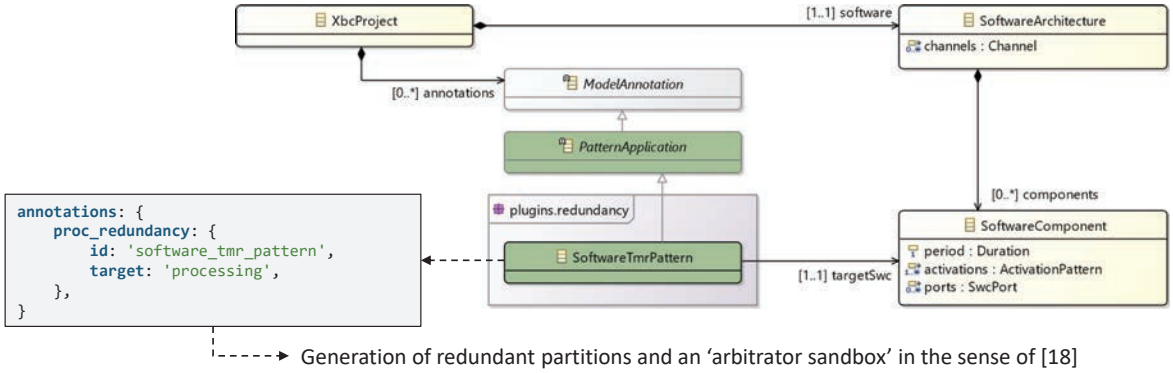
- **XbC pattern library:** Collection of verified design-time procedures for safety and security that can be annotated to system entities [17]
- Automatic implementation of the corresponding mechanisms:



[17] T. Dörr, F. Schade, et al., **"Safety by Construction: Pattern-Based Application of Safety Mechanisms in XANDAR"**, ISVLSI '22, July 2022.

Safety pattern concept

- **Pattern usage**: Explicit specification in the textual model instance (JSON5)
- **Example:** Triple Modular Redundancy (TMR) for fault tolerance



- **Next up:** Information Flow Control (IFC) + runtime logging for AI components

[18] E. Missimer, R. West, Y. Li, **"Distributed Real-Time Fault Tolerance on a Virtualized Multi-Core System"**, OSPERT '14, July 2014.

Information Flow Control (IFC) safety pattern

- **Motivation**: Unintended information flow via shared on-chip resources can lead to the violation of safety/security requirements, for example…

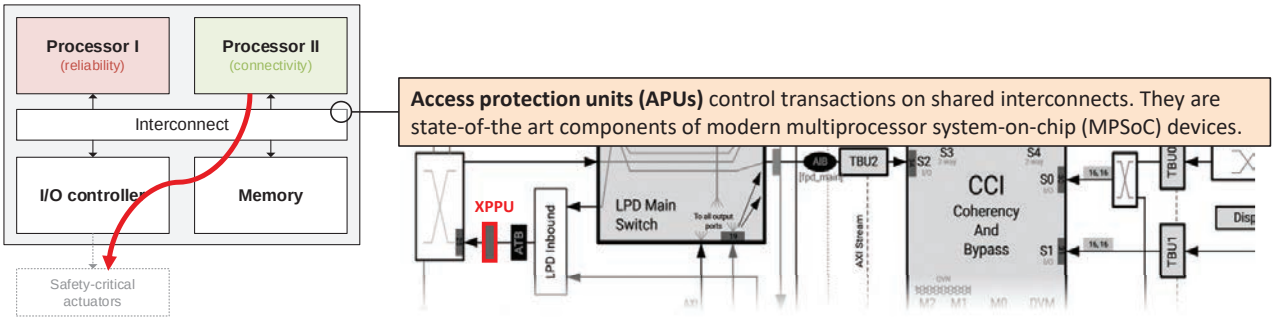


Image source: [19]

- **Concept:** Auto-generate APU configurations that are as prohibitive as possible and compare potentially feasible information flows to an accept list

[19] AMD/Xilinx, **"Zynq UltraScale+ Device: Technical Reference Manual"**, UG1085 (v2.3), September 2022.

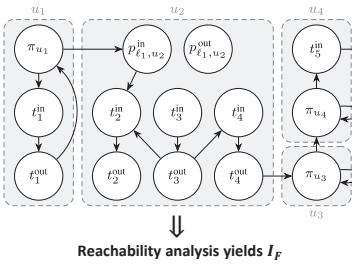
Information Flow Control (IFC) safety pattern

- **Underlying methodology** from the state of the art [20]:
  - Inputs are the system architecture and the desired information flow policy (I_{A})
  - Graph-based algorithm to determine all potentially feasible information flows (I_F)

Algorithm 1 G_β construction from a model instance

```
G_β ← (V, E), V ← ∅, E ← ∅
for all ℓ ∈ L : ω_L(ℓ) = 0 do
    if ∃ u ∈ φ_L(ℓ) : δ_U(u) = 0 then
        V ← V ∪ {π_ℓ}                          ▷ Add link sharing node
for all u ∈ U : δ_U(u) = 0 do
    V ← V ∪ {π_u}                              ▷ Add unit sharing node
    for all ℓ ∈ φ_L(u) : ω_L(ℓ) = 0 do
        E ← E ∪ {(π_u, π_ℓ), (π_ℓ, π_u)}
for all u ∈ U : δ_U(u) = 1 do
    for all ℓ ∈ φ_L(u) do
        V ← V ∪ {p^in_{ℓ,u}, p^out_{ℓ,u}}          ▷ Add port nodes
        if π_ℓ ∈ V then
            E ← E ∪ {(p^src_{ℓ,u}, u), (p^in_{ℓ,u}, p^dst_{ℓ,u})}
```

⇒



- **XbC guarantee:** $\mathrm{I_F} \subseteq I_A$ holds for interactions between SWCs, ports, …

[20] T. Dörr, T. Sandmann, J. Becker, **"Model-based configuration of access protection units in networks of heterogeneous multicore processors"**, Microprocessors and Microsystems (MICPRO), vol. 87, November 2021.

Runtime logging for AI components

- Upcoming **normative and legislative constraints on AI components**:
  - Ethics of Connected and Automated Vehicles [21] by the European Commission
  - EASA Concept Paper: First usable guidance for Level 1 machine learning applications [22]

- Selected excerpts focused on the **transparency** of algorithmic decisions:

"User-centred methods and interfaces for the explainability of AI-based forms of CAV decision-making should be developed."

Source: [21]

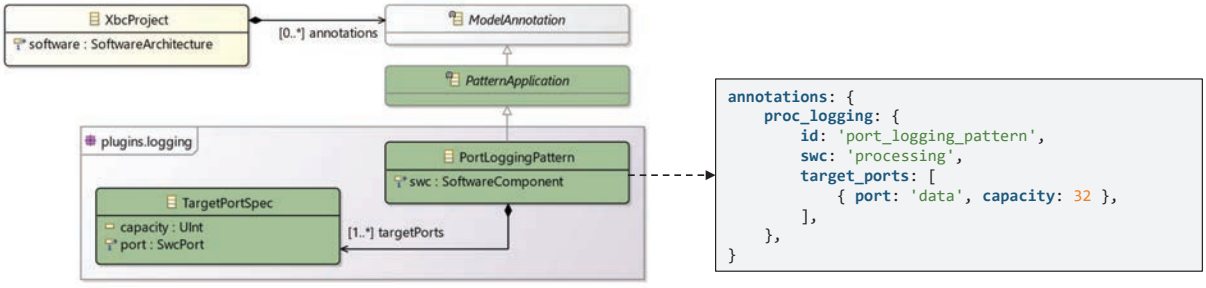
"Did you put adequate logging practices in place to record the decision(s) or recommendation(s) of the AI-based system?"

Source: [22]

[21] European Commission, **"Ethics of Connected and Automated Vehicles: recommendations on road safety, privacy, fairness, explainability and responsibility"**, 2020.
[22] European Union Aviation Safety Agency, **"EASA Concept Paper: First usable guidance for Level 1 machine learning applications"**, issue 01, 2021.

Runtime logging for AI components

- **Library-provided logging support** at the level of SWC ports:



- **XbC guarantee:** Correct synthesis and deployment of logging modules
  - Allocation of a dedicated memory region (incl. IFC pattern compatibility)
  - Non-interference with the LET-based communication model

Part II

Programming models for concurrent systems

Fundamentals

Concurrency – Two or more actions are in progress at the same time [23].
Parallelism – Two or more actions execute simultaneously [23].

Shared memory communication – Concurrent modules communicate via objects in memory [24].
Message passing – Concurrent modules communicate by transmitting messages over a channel [24].

- Cyber-physical systems (CPS) are inherently concurrent [25]
  ⇒ Approaches to achieve **deterministic concurrency** are essential
- Incorrect data sharing is a common source of concurrency issues [26]
  ⇒ Techniques based on **message passing particularly interesting** for CPS applications

[23] C. Breshears, **"The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications"**, O'Reilly Media, Inc., 2009.
[24] R. Miller, M. Goldman, 6.031 — Software Construction, Reading 21: Concurrency, online: https://web.mit.edu/6.031/www/sp22/classes/21-concurrency, 2022.
[25] P. Derler, E. A. Lee, A. Sangiovanni Vincentelli, **"Modeling Cyber-Physical Systems"**, Proceedings of the IEEE, vol. 100, no. 1, January 2012.
[26] P. Koopman, **"A Case Study of Toyota Unintended Acceleration and Software Safety"**, online: https://betterembsw.blogspot.com/2014/09/a-case-study-of-toyota-unintended.html, 2014.

Kahn Process Networks (KPN)

- **Idea:** simple language for parallel programming [27]
  - Processes (*computing stations*) = sequential programs with Algol-like syntax
  - Channels (*communication lines*) = unbounded first-in-first-out queues
  - Asynchronous message passing (nonblocking writes and blocking reads)

- KPN programs are deterministic [28]:

  "Kahn showed that concurrent execution was possible without nondeterminism"

- Expressivity deliberately limited

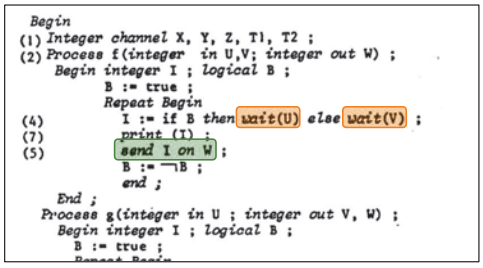


Image source: [27]

[27] G. Kahn, **"The semantics of a simple language for parallel programming"**, IFIP Congress, 1974.
[28] C. Ptolemaeus (editor), **"System Design, Modeling, and Simulation using Ptolemy II"**, Ptolemy.org, 2014.

Communicating Sequential Processes (CSP)

- **Key principle:** parallel composition of sequential processes [29]
  - Introduced to coordinate/synchronise multiprocessor machine communication
  - Synchronous message passing (blocking writes and blocking reads)

- Part of programming languages such as occam-π [30] and Go [31]:

```
func main() {
    exit := make(chan bool)
    channel := make(chan uint8)

    go consumer(channel, exit)
    go producer(channel)
    <-exit
}
```

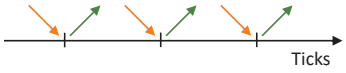
```
func consumer(in <-chan uint8, exit chan<- bool) {
    fmt.Println(<-in)
    exit <- true
}
```

```
func producer(out chan<- uint8) {
    time.Sleep(time.Second)
    out <- 42
}
```

[29] C. A. R. Hoare, **"Communicating Sequential Processes"**, Communications of the ACM, vol. 21, no. 8, 1978.
[30] P. H. Welch, F. R. M. Barnes, **"Communicating Mobile Processes: Introducing occam-pi"**, Communicating Sequential Processes: The First 25 Years, Springer, Berlin, Heidelberg, 2005.
[31] J. Whitney, C. Gifford, M. Pantoja, **"Distributed execution of communicating sequential process-style concurrency: Golang case study"**, Journal of Supercomputing, vol. 75, 2019.

Synchronous languages

- **Key principle:** combination of synchrony and concurrency [32]
  - Examples are Lustre [33] or more recent approaches such as Blech [34]
  - Immediate reactions (simultaneous reads and writes)
  - Temporal semantics based on a global clock

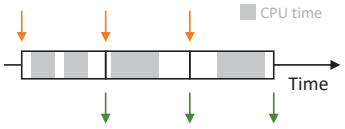


- Strong foundation for formal verification
- Widespread use for safety-critical system design:
  - Lustre is a fundamental part of the SCADE toolset by Esterel Technologies
  - Applications include nuclear power plants and flight control software [32]

[32] A. Benveniste, P. Caspi, et al., **"The synchronous languages 12 years later"**, Proceedings of the IEEE, vol. 91, no. 1, January 2003.
[33] N. Halbwachs, P. Caspi, et al., **"The synchronous data flow programming language LUSTRE"**, Proceedings of the IEEE, vol. 79, no. 9, September 1991.
[34] F. Gretz, F.-J. Grosch, **"Blech, Imperative Synchronous Programming!"**, FDL '18, September 2018.

Logical Execution Time (LET)

- **Origin:** Giotto methodology for embedded control system design [13]
  - Periodic invocation of sequential tasks ⇒ time-triggered framework
  - Reading of input ports at the invocation time
  - Writing of output ports at the end of the period



- Other manifestations of the LET abstraction:
  - xGiotto for event-driven programming [35]
  - Timing Definition Language (TDL), cf. [36]
  - System-level LET for distributed systems [37]
  - The synchronous LET paradigm [38]

[13] T. A. Henzinger, B. Horowitz, C. M. Kirsch, **"Embedded Control Systems Development with Giotto"**, SIGPLAN Not., vol. 36, no. 8, August 2001.
[35] A. Ghosal, T. A. Henzinger, et al., **"Event-Driven Programming with Logical Execution Times"**, Hybrid Systems: Computation and Control, HSCC '04, Springer, Berlin, Heidelberg, 2004.
[36] W. Pree, J. Templ, **"Modeling with the Timing Definition Language (TDL)"**, ASWSD '06, Springer, Berlin, Heidelberg, 2006.
[37] R. Ernst, L. Ahrendts, K.-B. Gemlau, **"System Level LET: Mastering Cause-Effect Chains in Distributed Systems"**, IECON '18, October 2018.
[38] F. Siron, D. Potop-Butucaru, et al., **"The synchronous Logical Execution Time paradigm"**, ERTS '22, June 2022, hal-03694950.

Reactors and Lingua Franca (LF)

- **Concept:** reactors as coordination model for concurrent systems [39]
  - A reactor is composed of **reactions** in a target language (C, Python, Rust, …)
  - Reactions are executed in response to **trigger events** (carrying a value and a tag)
- Lingua Franca (LF) for the development of reactor-based systems:

```
target C;
main reactor Timer {
    timer t(0, 1 sec);
    reaction(t) {=
        printf("Timer expired!\n");
    =}
}
```

"Lingua Franca is a polyglot coordination language for reactive, concurrent, and time-sensitive applications."
www.lf-lang.org

[39] M. Lohstroh, C. Menard, et al. **"Toward a Lingua Franca for Deterministic Concurrent Systems"**, ACM Transactions on Embedded Computing Systems, vol. 20, no. 4, May 2021.

Applicability to the XANDAR toolchain

- Compatibility with the full feature set of XANDAR is essential
- **LET has been shown to offer a promising trade-off**
  - Straightforward integration into the overall framework
  - Compatible with the requirements of XANDAR's use cases
- Relaxation of current constraints is future work
- Ideas for steps beyond the current model:
  - Integration of LF, which is a generalisation of LET [40]
  - Add support for the synchronous LET paradigm [38]

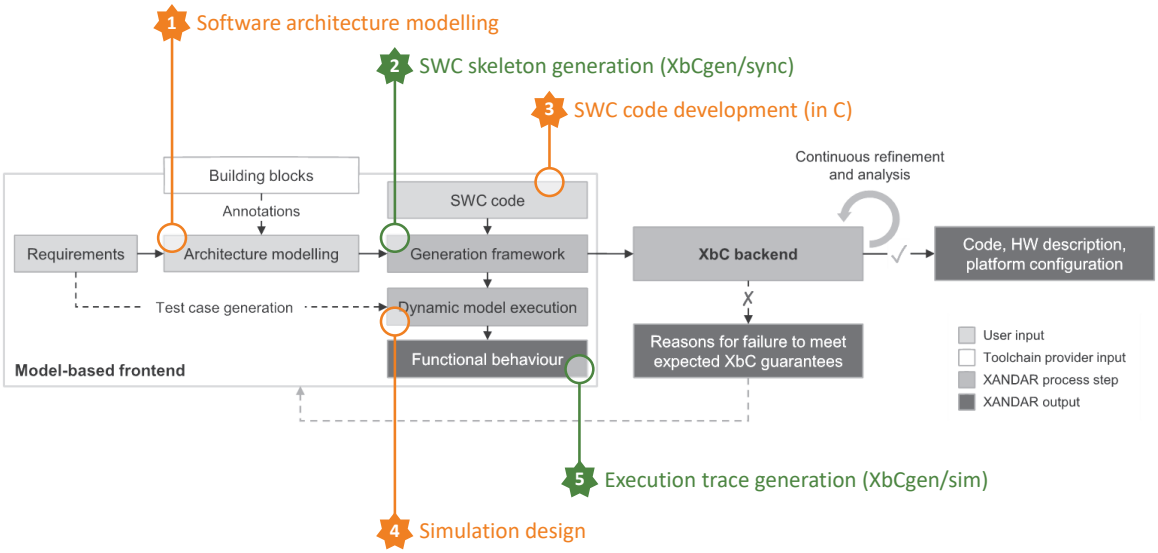


[38] F. Siron, D. Potop-Butucaru, et al., **"The synchronous Logical Execution Time paradigm"**, ERTS '22, June 2022, hal-03694950.
[40] E. A. Lee, M. Lohstroh, **"Generalizing Logical Execution Time"**, Principles of Systems Design - Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday, July 2023.

Part III

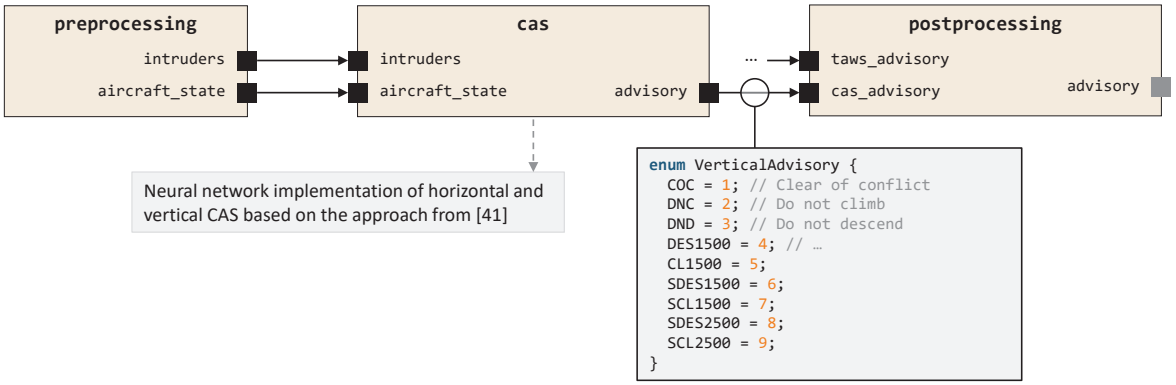
Live demonstration of XbCgen

Overview and introductory examples



Application to an avionics use case

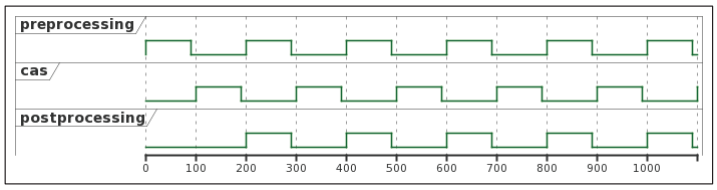
- SWC chain from the **Tactical Air Risk Mitigation System (TARMS)** by DLR:



[41] K. D. Julian, M. J. Kochenderfer, **"Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems"**, DASC '19, September 2019.

Application to an avionics use case

- Sample activation pattern specification:



- FlightGear [42] scenarios serve as simulation stimuli:



⇒

```
struct scenario_trace_item scenario_trace[] = {
    { 3,63.99467872,-22.65,500,77.74823761,-0,180,0,0,63.
    { 3.51667,63.99381768,-22.65,500,77.75377655,-0,180,0
    { 4.05,63.9936291,-22.65,500,77.75377655,-0,180,0,0,6
    { 4.55833,63.99344053,-22.65,500,77.75377655,-0,180,0
    { 5.025,63.99327291,-22.65,500,77.75377655,-0,180,0,0
    { 5.50833,63.99310828,-22.65,500,77.75377655,-0,180,0
    { 6.03333,63.99291073,-22.65,500,77.75377655,-0,180,0
    { 6.55,63.99272215,-22.65,500,77.75377655,-0,180,0,0,
    { 7.00833,63.99256052,-22.65,500,77.75377655,-0,180,0
    { 7.53333,63.99237195,-22.65,500,77.75377655,-0,180,0
    { 8.05,63.99218637,-22.65,500,77.75377655,-0,180,0,0,
    { 8.55833,63.99200977,-22.65,500,77.75377655,-0,180,0
```

[42] FlightGear Flight Simulator website, online: www.flightgear.org, accessed 2 May 2023.

Closing

Summary

Summary

- **Software development methodology of the XANDAR toolchain:**
  - Software modelled as a network of SWCs with LET parameters
  - Pre-verified safety/security patterns are annotated by the toolchain user
  - Behaviour simulation + automatic deployment to a type-1 hypervisor on MPSoCs
- **Programming models for concurrent systems:**
  - LET has been shown to be a suitable model for the XANDAR framework
  - Relaxation of current constraints (e.g. by moving to LF) is future work
- **Live demonstration of XbCgen:**
  - Flow from modelling to execution trace generation
  - Application to the DLR use case



Embedded Systems Modeling, Analysis and Automatic Code Generation with AADL and RAMSES (Hands-on Tutorial)

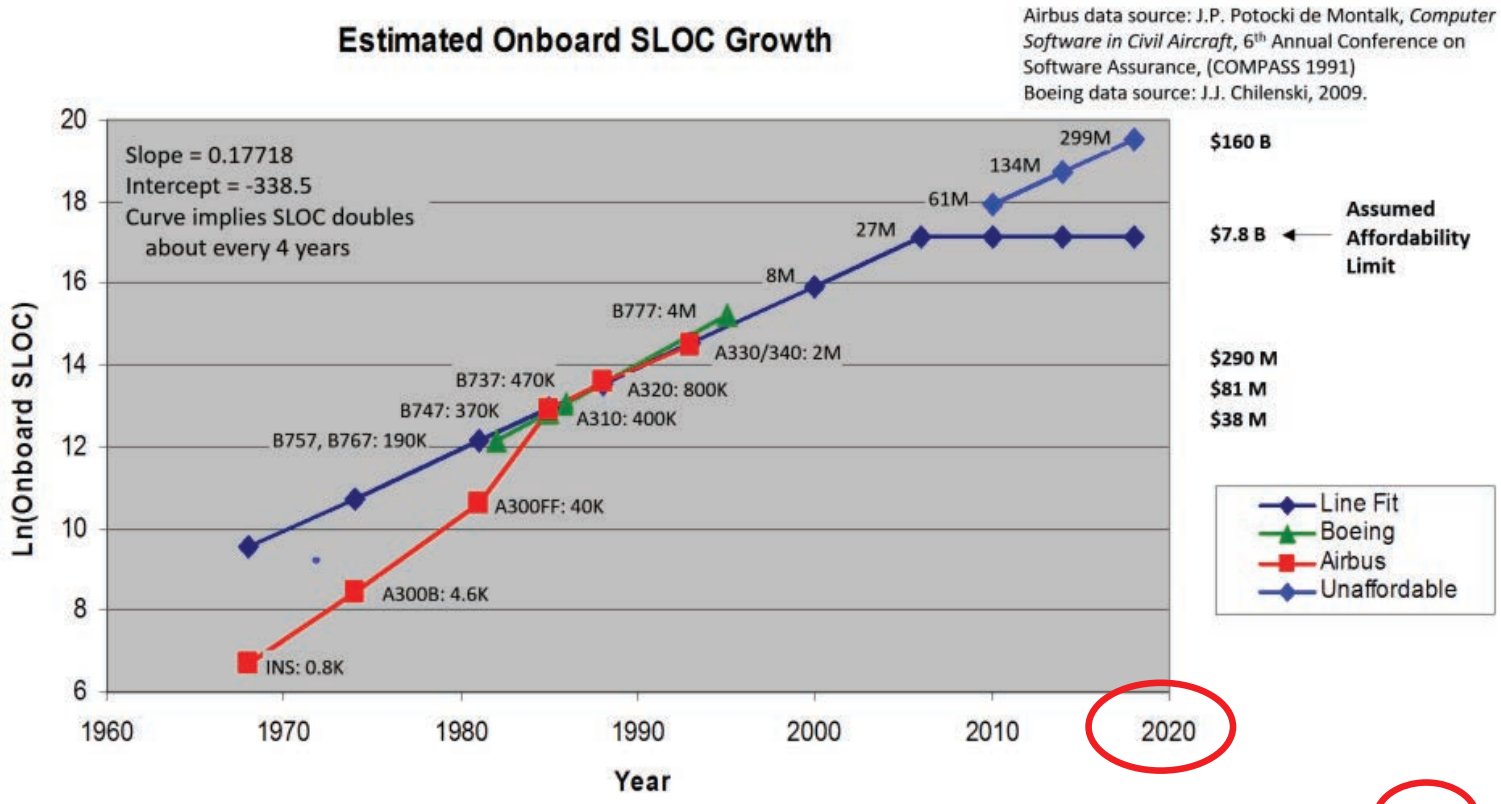
Dominique Blouin[1], Anish Bhobe[1], Rakshit Mittal [1,2] and Etienne Borde[1]
[1]LTCI Lab, Telecom Paris
Institut Polytechnique de Paris, France
dominique.blouin@telecom-paris.fr
anish.bhobe@ip-paris.fr
etienne.borde@telecom-paris.fr
[2] Department of Informatics
University of Antwerp
Rakshit.Mittal@uantwerpen.be



Outline

- Model-Based Engineering
- Introduction to AADL
- Timing Analysis with AADL
- Automatic Code Generation with RAMSES
- Introduction of the Hands-on Exercise

Increasing Systems Complexity and Unaffordable Development Costs



Source: Feiler, Hansson, de Niz and Wrage. "System Architecture Virtual Integration: An Industrial Case Study", 2009.

Non-Linear Development Effort Increase: F35 versus F16 Example





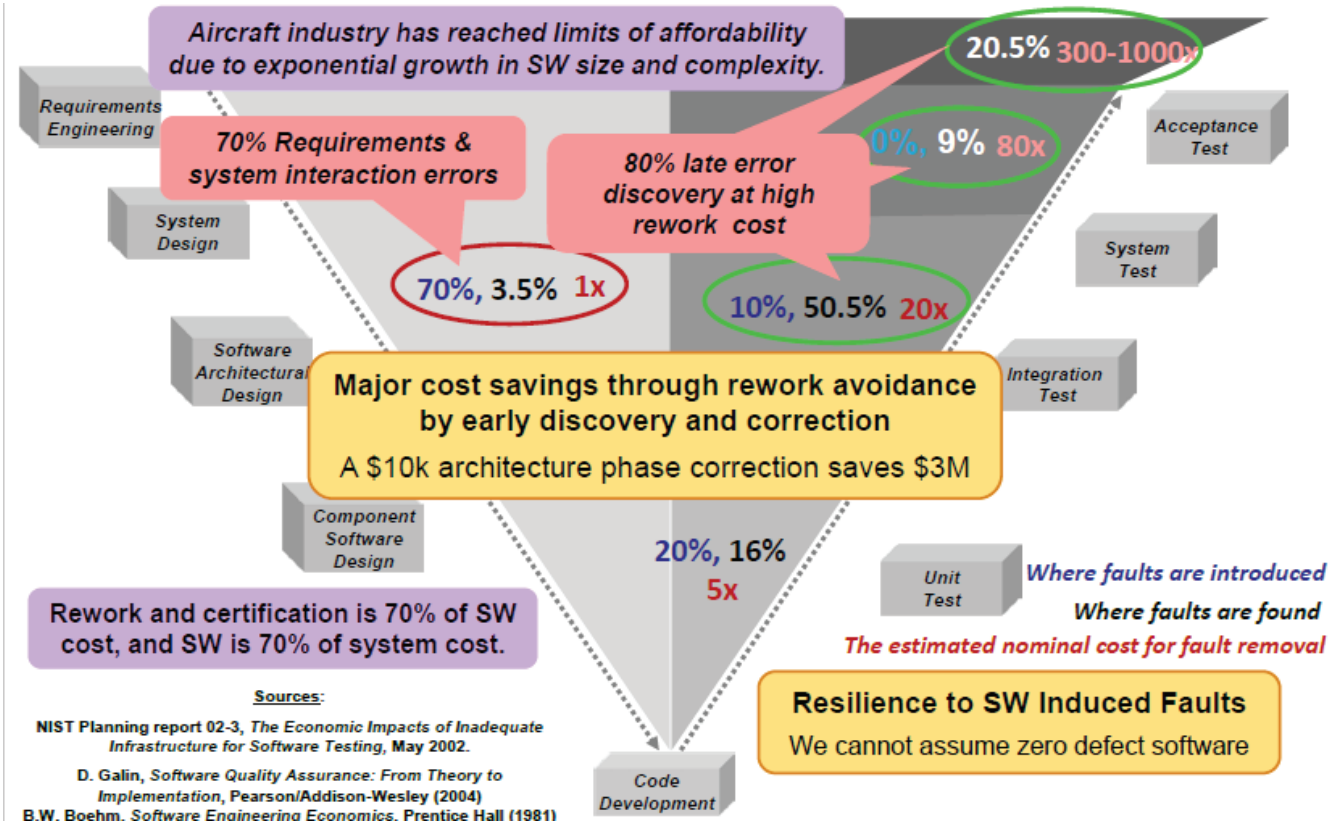


- **F35 SLOC / F16 SLOC ~ 175**
- **F35 Effort / F16 Effort ~ 300**
  - Source: SAVI Project (https://savi.avsi.aero/)

A400M:

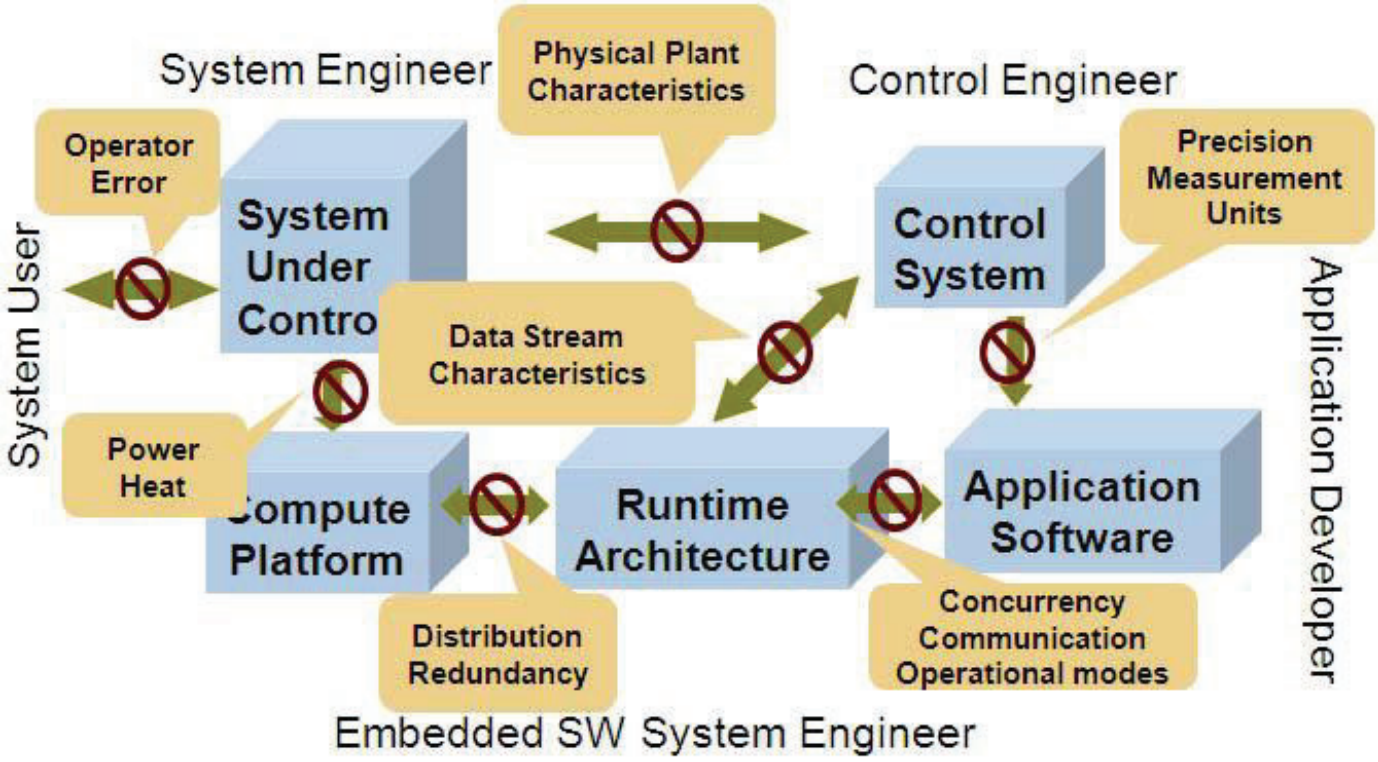
Over 10 years delayed (2013)
6.2 billion euros over budget (30% overrun)
Source: https://www.rt.com/business/airbus-a400m-france-delays-561/

Source of Costs : Errors Introduced Early



Source: P. Feiler and J. Delange, "Design and Analysis of Cyber-Physical Systems: AADL and Avionics Systems", 2013

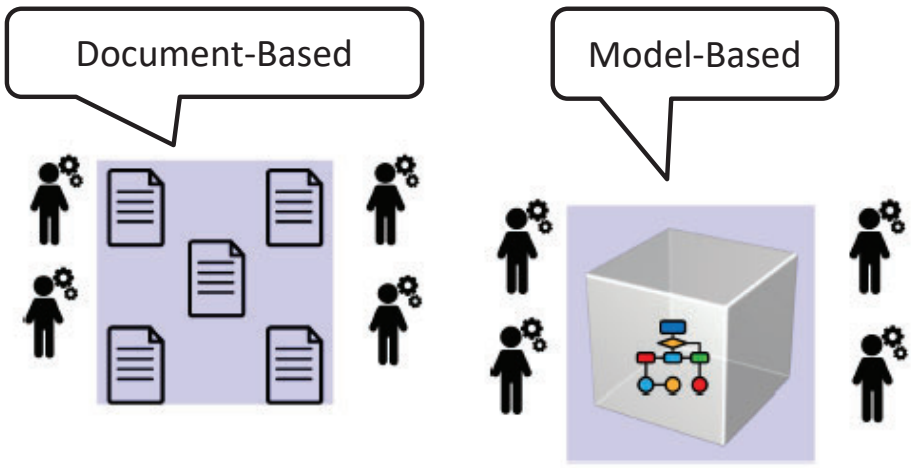
Mismatched Assumptions in Collaborative Engineering



Source: P. Feiler, "Model-based validation of safety-critical embedded systems", 2010

New Paradigm: Model-Based Engineering (MBE)

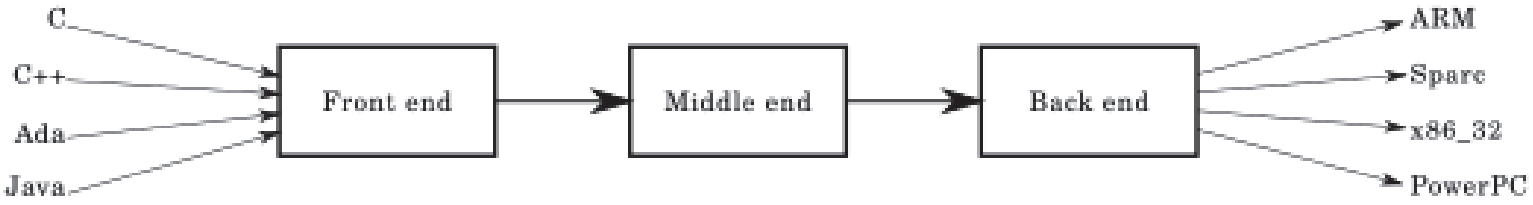
Paradigm shift: From natural language documents to models



Provide common vocabulary
Enforce more precision
Allow building tools to process specifications (models)
Allow detecting errors / inconsistencies *early*
Quite effective for avionics development (> 25 % costs reduction)

Analogy with Code Compilation

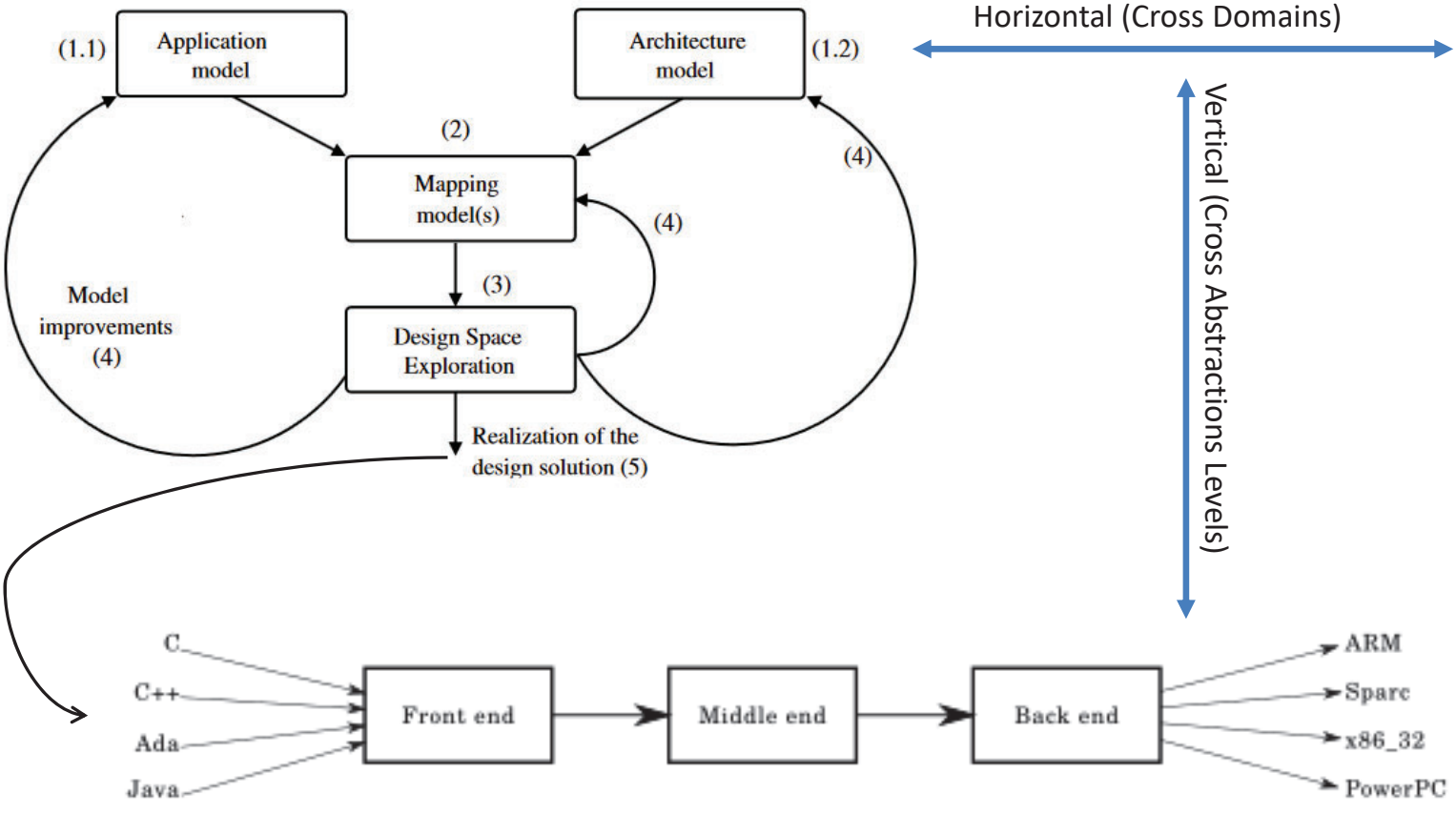
Compilation chain:



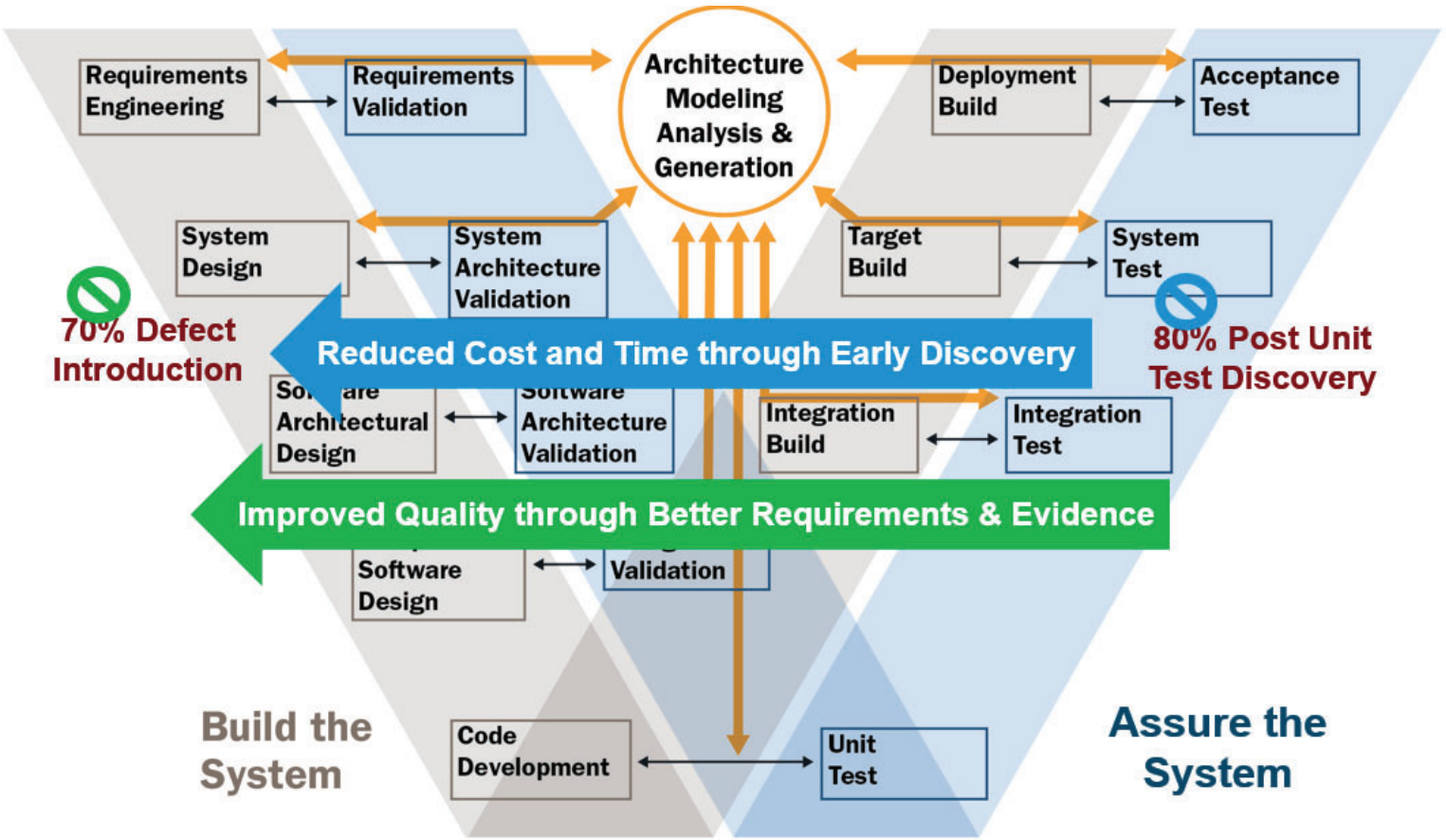
Key points:

- Increase level of abstraction
- Execution platform independence

Separation of Concerns



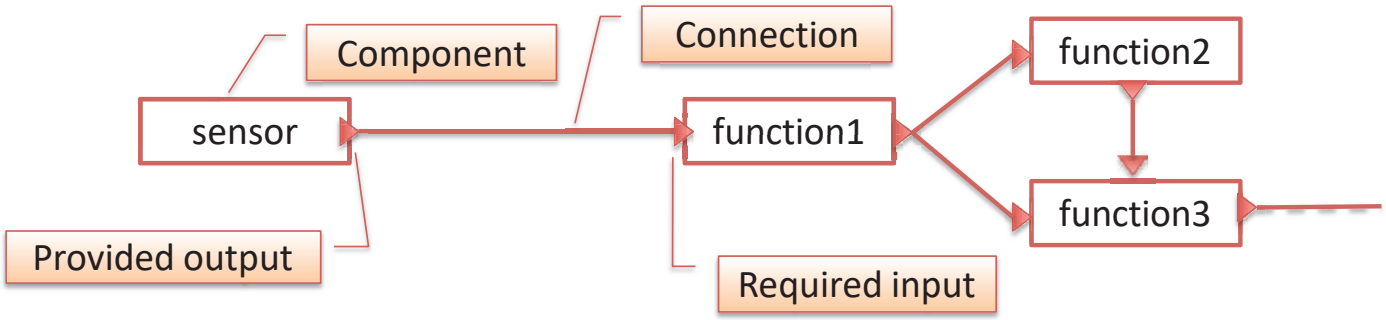
V-Cycle Model with Model-Based Virtual Integration Activities



Source: J. McGregor, P. Gluch and P. Feiler, "Analysis and Design of Safety-critical, Cyber-physical Systems", 2017.

Components-based Architecture Models

- Architecture models represent the **organization of a computer system** as a **set of components** and their **interactions**
- Main artefacts: boxes and arrows
  - ⌘ Components : main elements of the design
  - ⌘ Interfaces : what components offer and what they need
  - ⌘ Connections : satisfy components needs



- Then drawing becomes programming… or at least designing… Nothing new conceptually
- What about the semantic?

Architecture Description Languages (ADL)

- Formal, i.e., based on a mathematically sound definition of their semantic
  - Meant to formally verify/prove expected properties of a computer system
  - Wright ADL, data flow graphs, state machines, …
- Domain-specific
  - Meant to describes the design and implementation of computer systems constituents
  - UML, UML MARTE, AUTOSAR, ***AADL***
- Abstract
  - Meant to describe the organization of a computer system without providing a precise semantics
  - ArchJava, Fractal

Note: some ADLs are standardized (e.g., UML, UML MARTE, AUTOSAR, AADL), which provides a common understanding of the notation, but to the cost of slow evolutions through a committee.

Outline

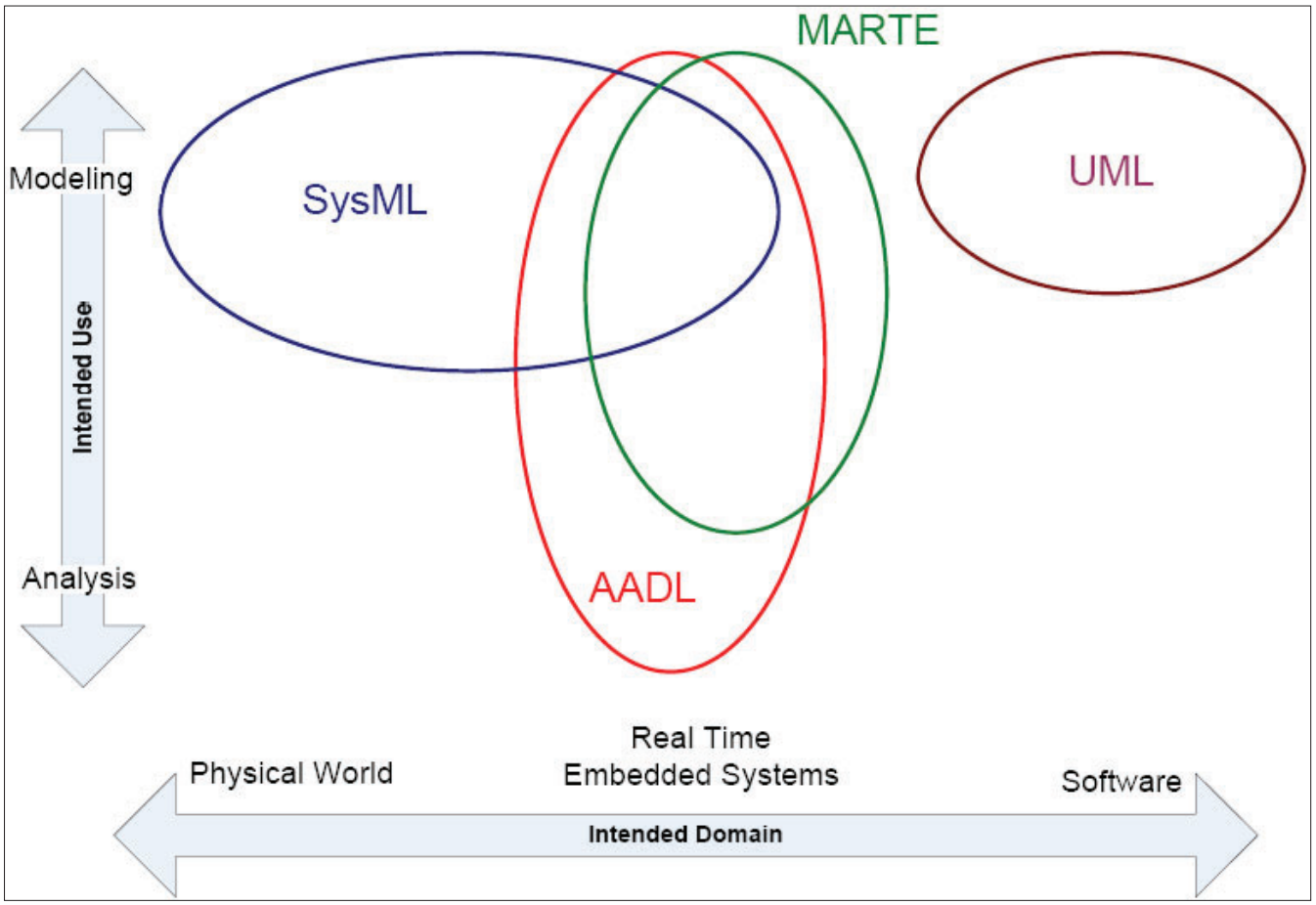
- ❑ Model-Based Engineering
- ❑ Introduction to AADL
- ❑ Timing Analysis with AADL
- ❑ Automatic Code Generation with RAMSES
- ❑ Introduction of the Hands-on Exercise

AADL (Architecture Analysis and Design Language)

- An ADL for real-time embedded systems
  - ⌘ Uses the principles of a concrete DSL (components, interfaces and connections)
  - ⌘ Define properties for real-time and embedded systems analysis
    - o Scheduling policy, compute execution time, latency…
    - o Software components to hardware components allocation

- Objective: assist the design of such systems
  - ⌘ Standardized semantics (formulated with natural language)
  - ⌘ Textual and graphical syntax
  - ⌘ Strongly typed (components category, composition rules, …)
  - ⌘ Extensible (property definition language and annexes)

Comparison with other Architecture Description Languages (ADL)



Source: Steven P. Miller, AADL standards winter meeting, 2011

General Characteristics

- Components are the main modeling entities
- The standard defines categories of components (keywords of the language); examples of categories a component can belong to:
  - System
  - Process, thread, data…
  - Processor, memory, bus…
- Components definition is divided into types, implementations, and subcomponents
  - Type: how the component is viewed from outside (e.g. interaction interfaces)
  - Implementation: internal structure of the component (e.g. subcomponents)
  - Subcomponents: instances of components, starting from a root system implementation
- Characteristics of components are structured into sections (e.g. **features, properties, subcomponents**) identified by keywords of the language
- Details:
  - Components can be declared in any order
  - The language is case insensitive

Language Constructs: Running Example

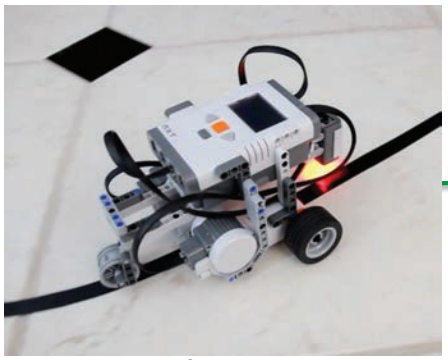
- Initial plan: a really complex system











- Actual case-study
  - Simpler system but still representative

System Level Viewpoint Categories

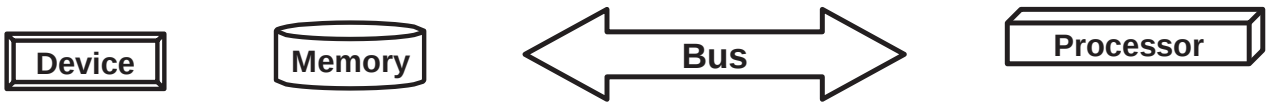
- Two categories: system and abstract



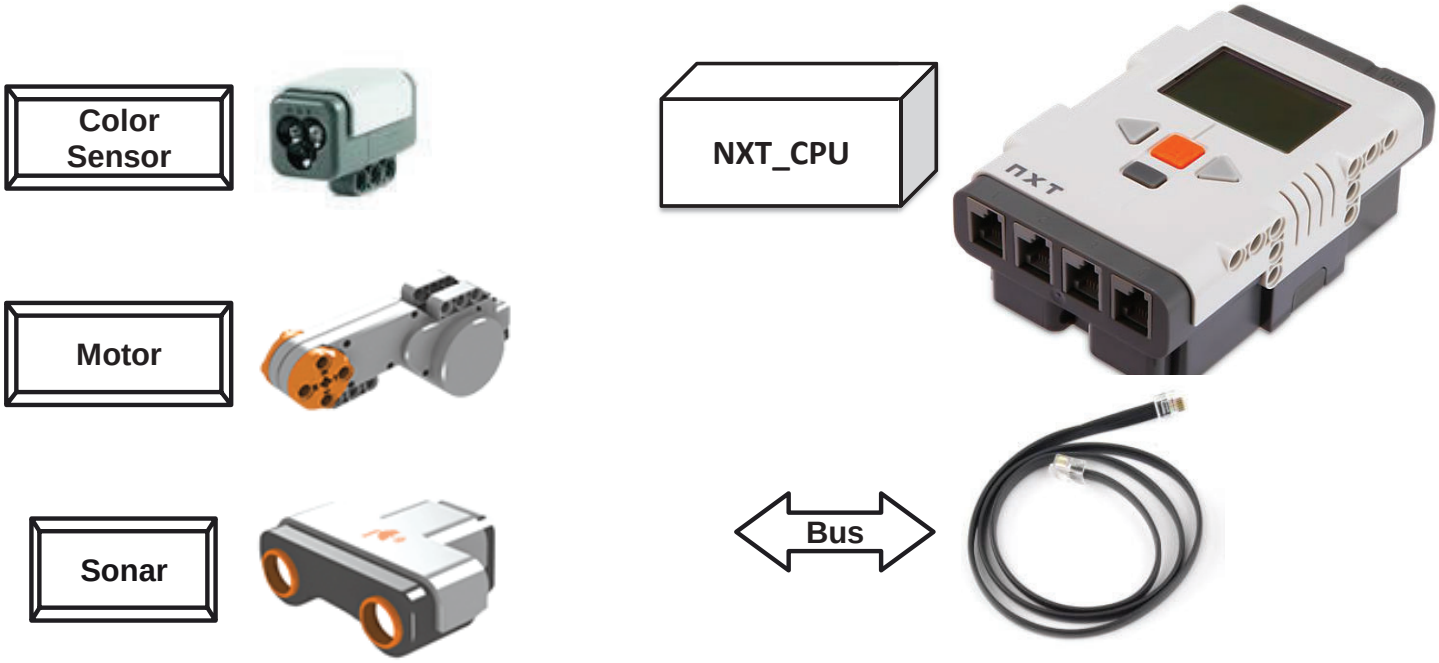
- Different possible objectives
  - ⌘ Represent, from a very abstract view point, the main constituent of the system, their interfaces and connections.
  - ⌘ System:
    - o Aggregate, by composition, subcomponents describing the execution platform and subcomponents describing the software architecture.
    - o Define the main operational modes of the system
  - ⌘ Abstract:
    - o Define structure and interaction without knowing yet the nature of the component

Execution Platform Viewpoint Categories

- Instances of components categories dedicated to the execution platform description
  - ⌘ processor : hardware computation unit + tasks scheduling capabilities
  - ⌘ memory : storage component (may be RAM, hard disk drive, cache, etc.)
  - ⌘ bus : physical communication link (network cable, etc.)
  - ⌘ device : interface with the physical environment of the system (sensors/ actuators)



Example Execution Platform Components



Software Architecture Viewpoint Categories

- AADL component categories for software
  - ⌘ Data: information that can be exchanged among software components
  - ⌘ Subprogram: sequentially executable software, like functions in C programming language
  - ⌘ Thread: task executing a sequence of functions
  - ⌘ Process: memory address space allocated for the execution of its thread subcomponents



- These categories focus on operating system and programming elements

Example of Software Components

data Light
end Light;



subprogram Compute_Angle_PID
end Compute_Angle_PID;



thread Trajectory_Control
end Trajectory_Control;



process Line_Follower
end Line_Follower



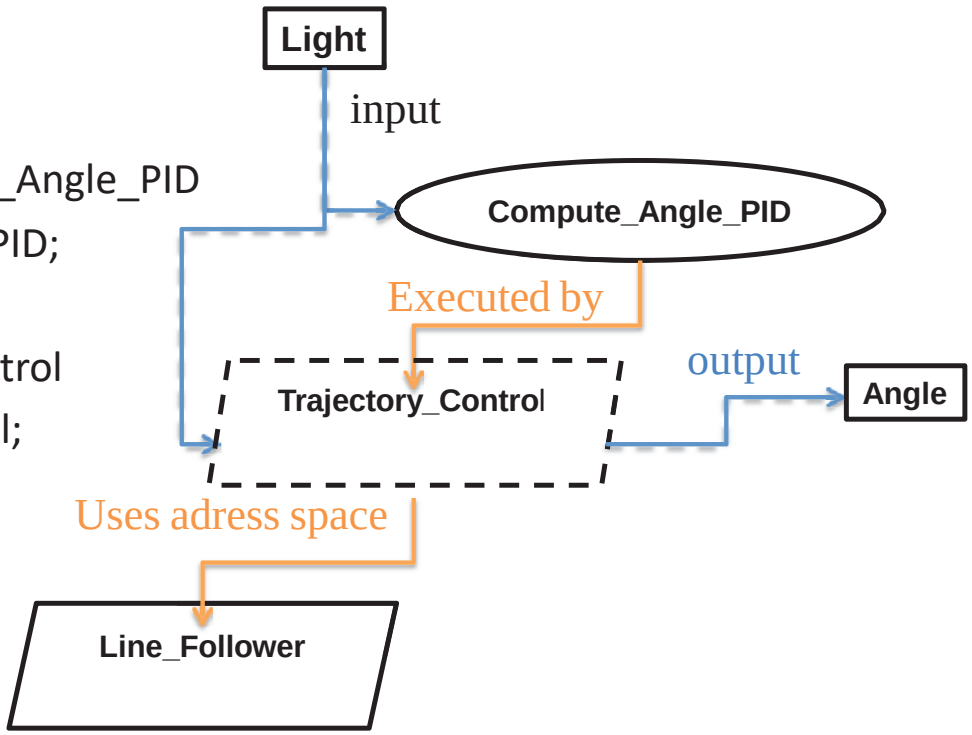
Example of Software Components

data Light
end Light;

subprogram Compute_Angle_PID
end Compute_Angle_PID;

thread Trajectory_Control
end Trajectory_Control;

process Line_Follower
end Line_Follower

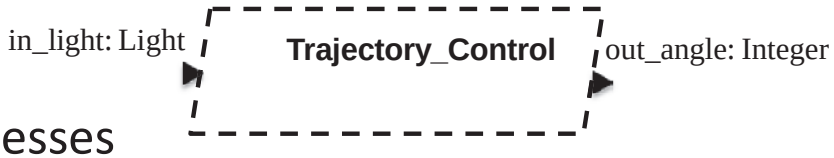


How to represent these interactions and allocations in AADL?

First, Define Component Interfaces (Component Types in AADL)

- Parameters
  - ⌘ **in**, **out**, or **inout**
  - ⌘ Usable for subprograms
- Requires or provides data access
  - ⌘ Usable for subprograms and threads
- Ports
  - ⌘ **in**, **out**, or **in out**
  - ⌘ **Data**, **Event** or **Event Data**
  - ⌘ Usable for threads and processes





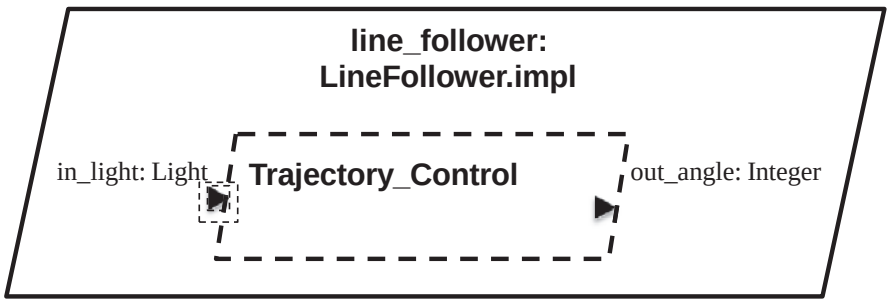
Semantical Differences among Features of Software Components

- Data Port versus Event Data Port
  - ⌘ Data Port : single value shared among components (no queueing).
  - ⌘ Event or Event Data Port : multiple values queued.

- Data Port versus Data Access
  - ⌘ Data Access allows access to the data at anytime during the execution of a task / subprogram
  - ⌘ Data Port defines the following semantics
    - o Data becomes available on an input port when the thread starts its execution. Data not used in the previous execution of the thread is lost. Data is not updated during the execution of the task.
    - o Data produced on an output port are sent to the recipient port at the end of the producer task.

Second, need to compose

- Thread subcomponents in process implementation:




```
process implementation Proc.simple
   subcomponents
      C_Th   : thread ContrThread.Impl;
end Proc.simple;
```

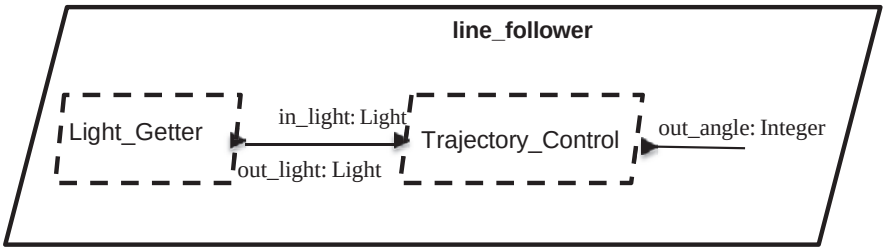
- Subprogram calls in call sequences

```
thread implementation ContrThread.Impl
calls
   call1:
   {
     cp   : subprogram  computePID;
     cs   : subprogram computeSpeed;
   };
end ContrThread.Impl;
```

Sequence of subprogram calls

Third, need to connect

- Components features are connected hierarchically
  - ⌘ Thread subcomponents in the process




```
process implementation Proc.simple
  subcomponents
    C_Th   : thread ContrThread.Impl;
  connections
    c2: port Bg_Th.out_light -> C_Th.in_light;
end Proc.simple;
```

  - ⌘ Subprogram calls in the thread but also the thread features with the subprogram calls

```
thread implementation ContrThread.Impl
calls
  call1:
  {
    cp   : subprogram computePID;
    cs   : subprogram computeSpeed;
  };
connections
  cc0 : parameter cp.currentLight -> in_light;
  cc1 : parameter cp.angle -> cs.angle;
end ContrThread.Impl;
```

Value received on the input port is passed as a parameter to the subprogram

Value produced by computePID is passed to compute Speed

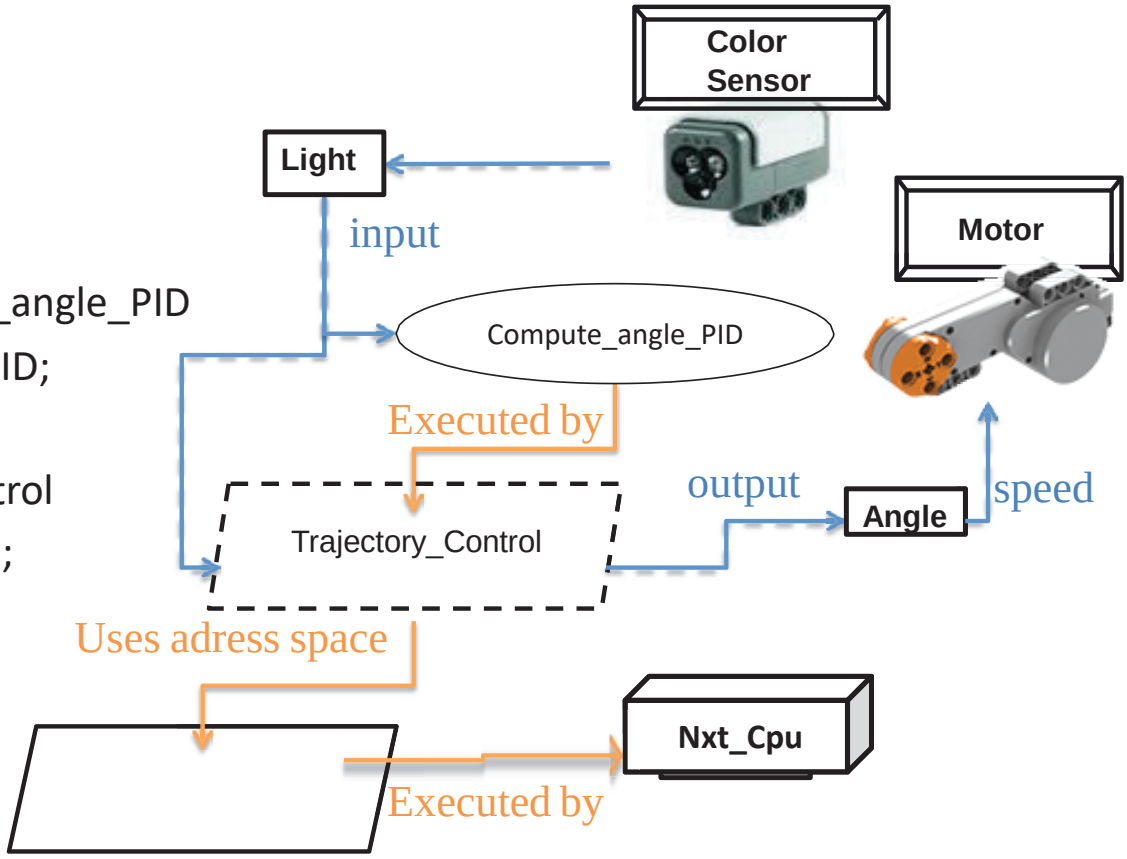
Combining Execution Platform and Software Architecture Viewpoints

data Light
end Light;

subprogram compute_angle_PID
end compute_angle_PID;

thread trajectory_control
end trajectory_control;

process line_follower
end line_follower



A few words about properties

- Properties in AADL is a way to decorate your model
  - ⌘ Properties can be associated to almost any element of your model.
  - ⌘ The standard defines a property language: AADL users can define their own **property sets** (means to extend the possibilities of the language).
  - ⌘ The AADL standard predefines a set of properties for most common used properties in ADLs and real-time embedded systems.
- Example of property definition

```
Actual_Processor_Binding: inherit list of reference (processor,
                                                     virtual processor,
                                                     system, device)
        applies to (thread, thread group, process,  system,
                    virtual processor, device);
```

- Example of property association

```
system implementation synchronous.others
   subcomponents
      my_platform : processor CPU;
      my_process : process my_process.impl;
   properties
      Actual_Processor_Binding => ( reference(my_platform) ) applies to my_process;
end synchronous.others;
```

Things we could not present

- AADL has much more than this
  - ⌘ Lots of standardized properties
  - ⌘ Modes, with mode-specific architectures (reconfiguration)
  - ⌘ Flows, to analyse the worst-case latency and jitter of data
  - ⌘ System/Hardware/Network configuration representation
  - ⌘ Behavior as state machines (BA)
  - ⌘ Errors and faults propagation (EMV2)

Open-Source AADL Tool Environment (OSATE)

- Developed at SEI (Carnegie Mellon University)
- Synchronized textual and graphical editors
- Eclipse-based (EMF, Ecore, Xtext, Graphiti, etc.)
- Actively maintained (more info at https://osate.org/)



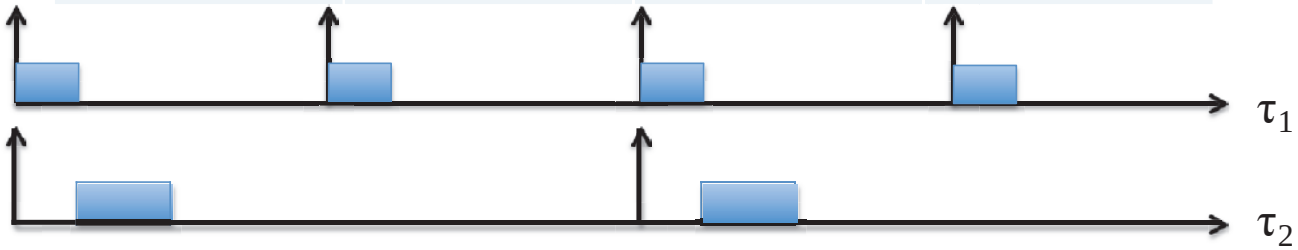
Outline

- ❑ Model-Based Engineering
- ❑ Introduction to AADL
- ❑ Timing Analysis with AADL
- ❑ Automatic Code Generation with RAMSES
- ❑ Introduction of the Hands-on Exercise

Principles of Response Time Analysis (RTA)

- Problem: the computation unit is shared among different tasks… Is that safe from a timing performance viewpoint?
  - Expressed as: tasks never miss their deadlines
- Assume a set of periodic tasks with a fixed priority scheduling configured using the Rate Monotonic Scheduling principles
  - RMS: the higher the frequency, the higher the priority
- RTA is meant to compute the worst--case response time of a task, based on:
  - The tasks period
  - The tasks worst case execution time (WCET) for the code of the task
  - The tasks deadline

Task	Period	WCET	Deadline
τ_1	10 ms	2 ms	6 ms
τ_2	20 ms	3 ms	10 ms



AADL Properties for RTA

- In terms of structure, we need
  - ⌘ Threads, representing tasks
  - ⌘ The processor on which threads are deployed
  - ⌘ The property value assigning threads to a processor

- Scheduling properties are
  - ⌘ *Dispatch_Protocol*, usually set with value ***periodic***
  - ⌘ *Compute_Execution_Time* represents the execution time interval, where the upper bound is the WCET
  - ⌘ The *Priority*, meaning is obvious.
  - ⌘ *Deadline*, meaning is obvious.
  - ⌘ *Scheduling_Protocol*, associated to the processor, defines the scheduling policy applied by the operating system running on the processor

Example (1/2)

```
PACKAGE synchronous_Pkg
PUBLIC
WITH Base_Types;

SYSTEM synchronous
END synchronous;

SYSTEM IMPLEMENTATION synchronous.others
  SUBCOMPONENTS
    my_platform : PROCESSOR CPU;
    my_process : PROCESS my_process.impl;
PROPERTIES
Actual_Processor_Binding =>
         ( reference(my_platform) )
           applies to my_process;
END synchronous.others;
```

```
PROCESSOR CPU
PROPERTIES
    Scheduling_Protocol => (RMS);
END CPU;


PROCESS my_process
END my_process;
```

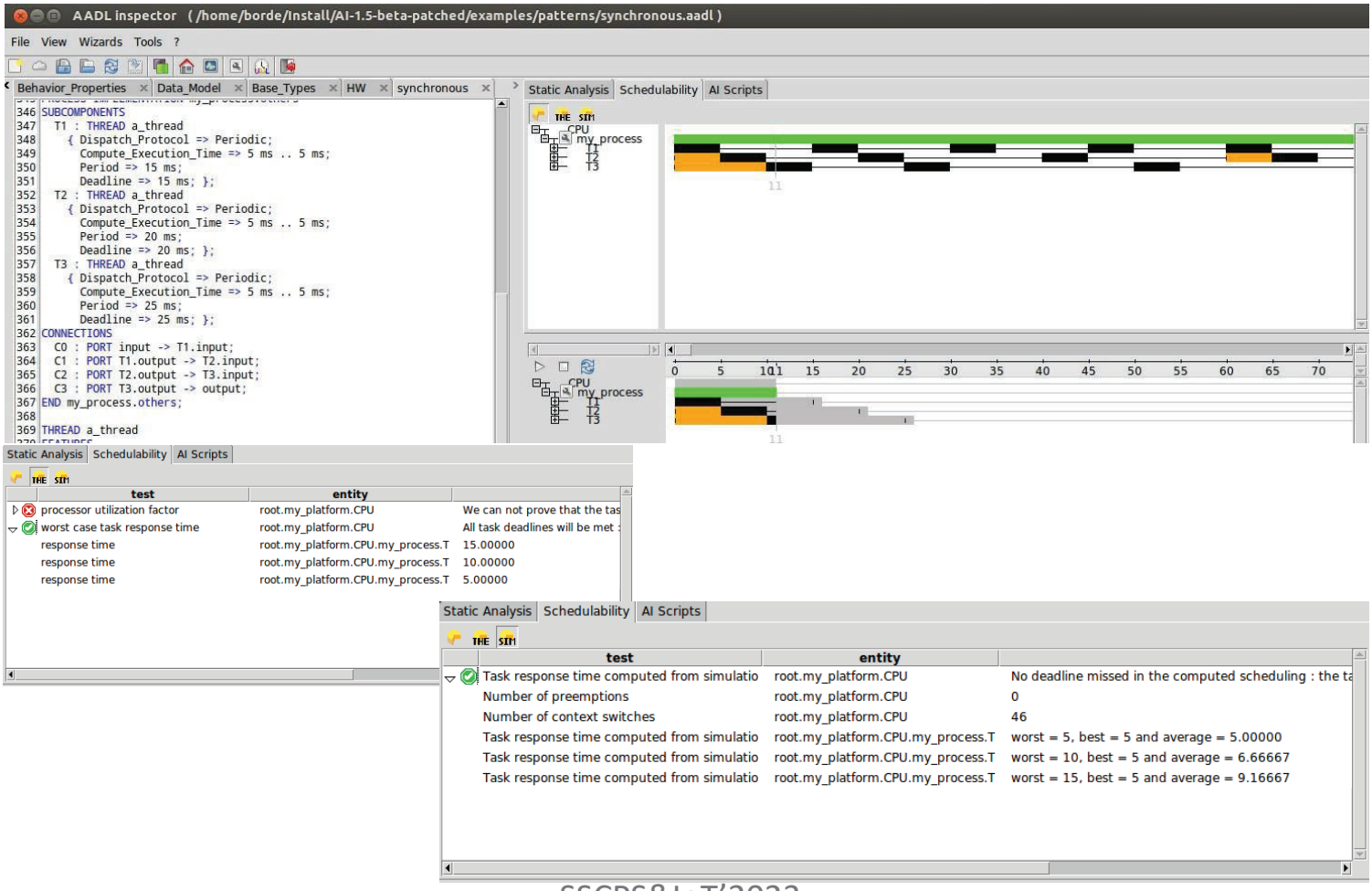
Example (2/2)

```
PROCESS IMPLEMENTATION my_process.impl
SUBCOMPONENTS
T1 : THREAD a_thread
   { Dispatch_Protocol => Periodic;
     Compute_Execution_Time=>5 ms..5 ms;
     Period => 15 ms;
     Deadline => 15 ms; };
T2 : THREAD a_thread
   { Dispatch_Protocol => Periodic;
     Compute_Execution_Time=>5 ms..5 ms;
     Period => 20 ms;
     Deadline => 20 ms; };
T3 : THREAD a_thread
   { Dispatch_Protocol => Periodic;
     Compute_Execution_Time=>5 ms..5 ms;
     Period => 25 ms;
     Deadline => 25 ms; };
END my_process.impl;
```

```
THREAD a_thread
END a_thread;

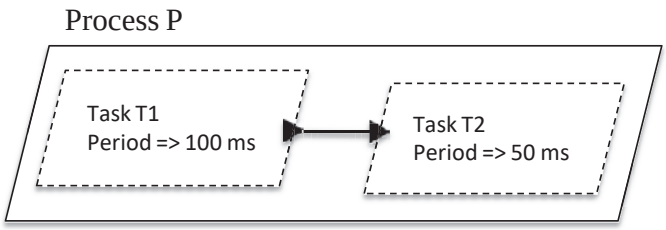

END synchronous_Pkg;
```

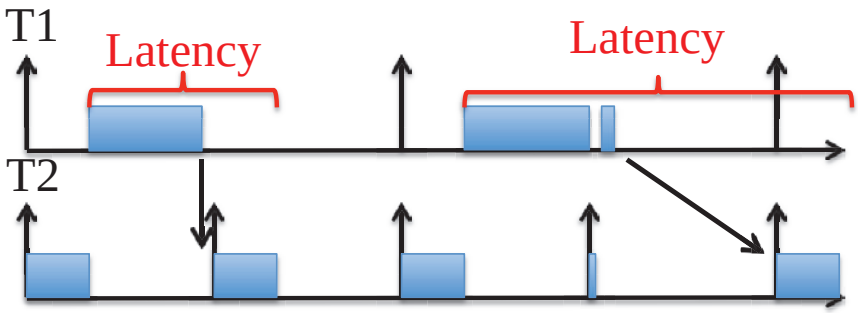
AADL Inspector



Communications through Data Ports

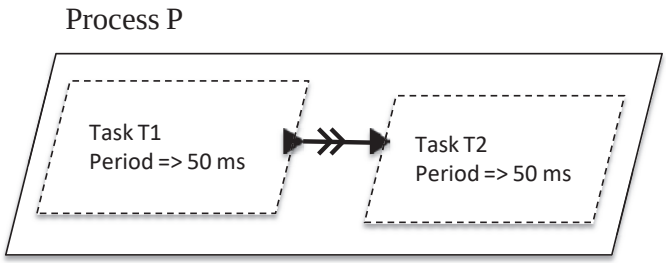
- Configured with the *Timing* property; possible values:
  - ⌘ Sampled (default): similar to a shared variable, except for the read/execute/write semantics
    - o Advantage: simplicity
    - o Disadvantage: undeterministic

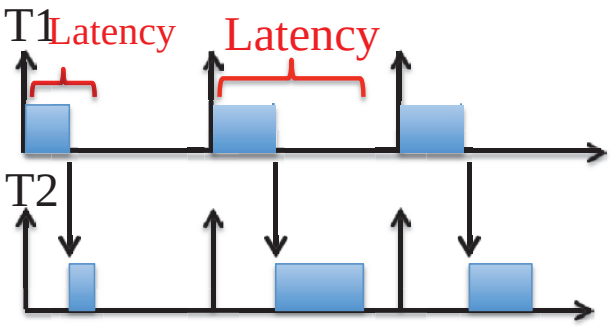




Communications through Data Ports

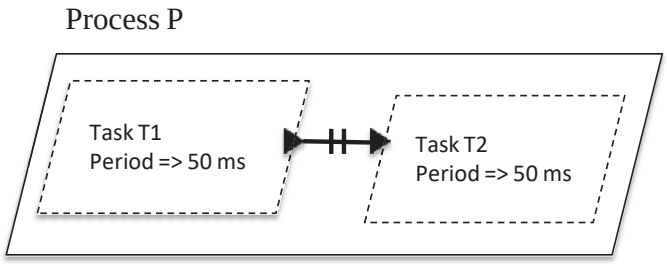
- Configured with the Timing property; possible values:
  - ⌘ Immediate: the recipient is not supposed to start until the output port of the connected thread has been updated
    - o Advantages: deterministic, reduces latency
    - o Disadvantages: put constraints on the scheduler and the model (no cycle, consistent periods)

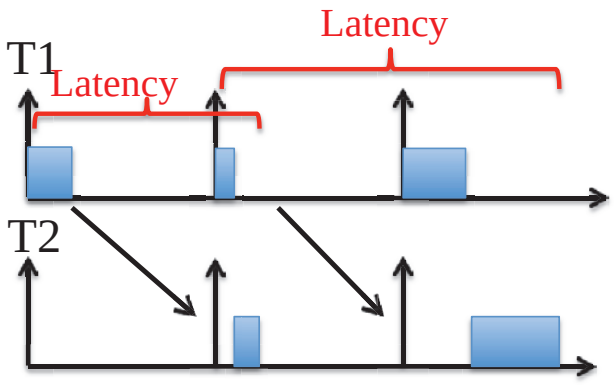




Communications through Data Ports

- Configured with the Timing property; possible values:
  - ⌘ Delayed: the output port is updated at the deadline of its thread
    - o Advantages: deterministic, reduces jitter
    - o Disadvantages: increases latency

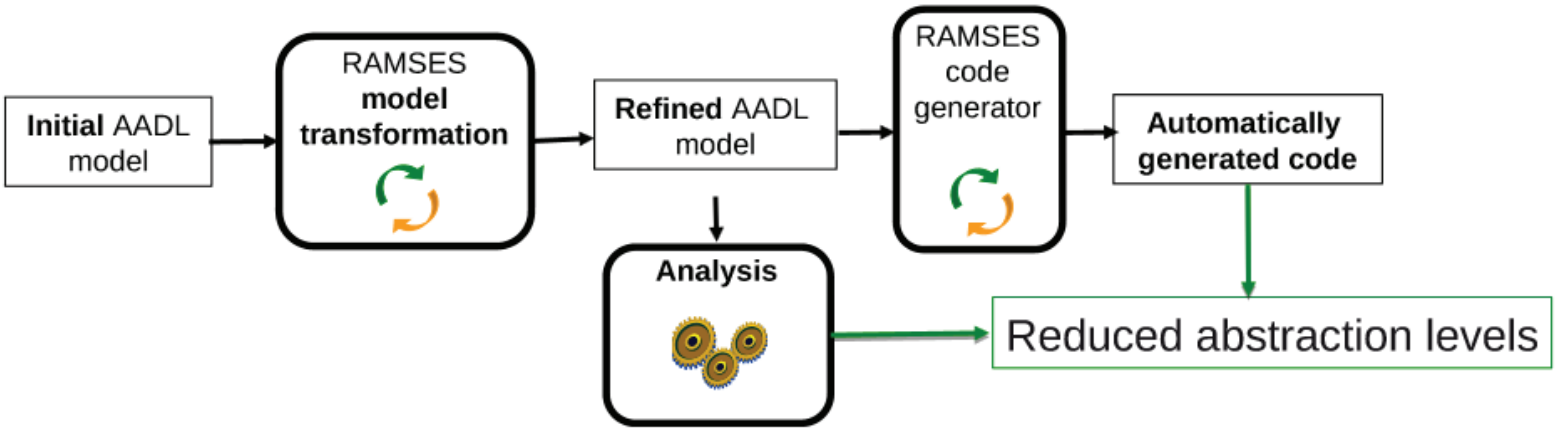




Outline

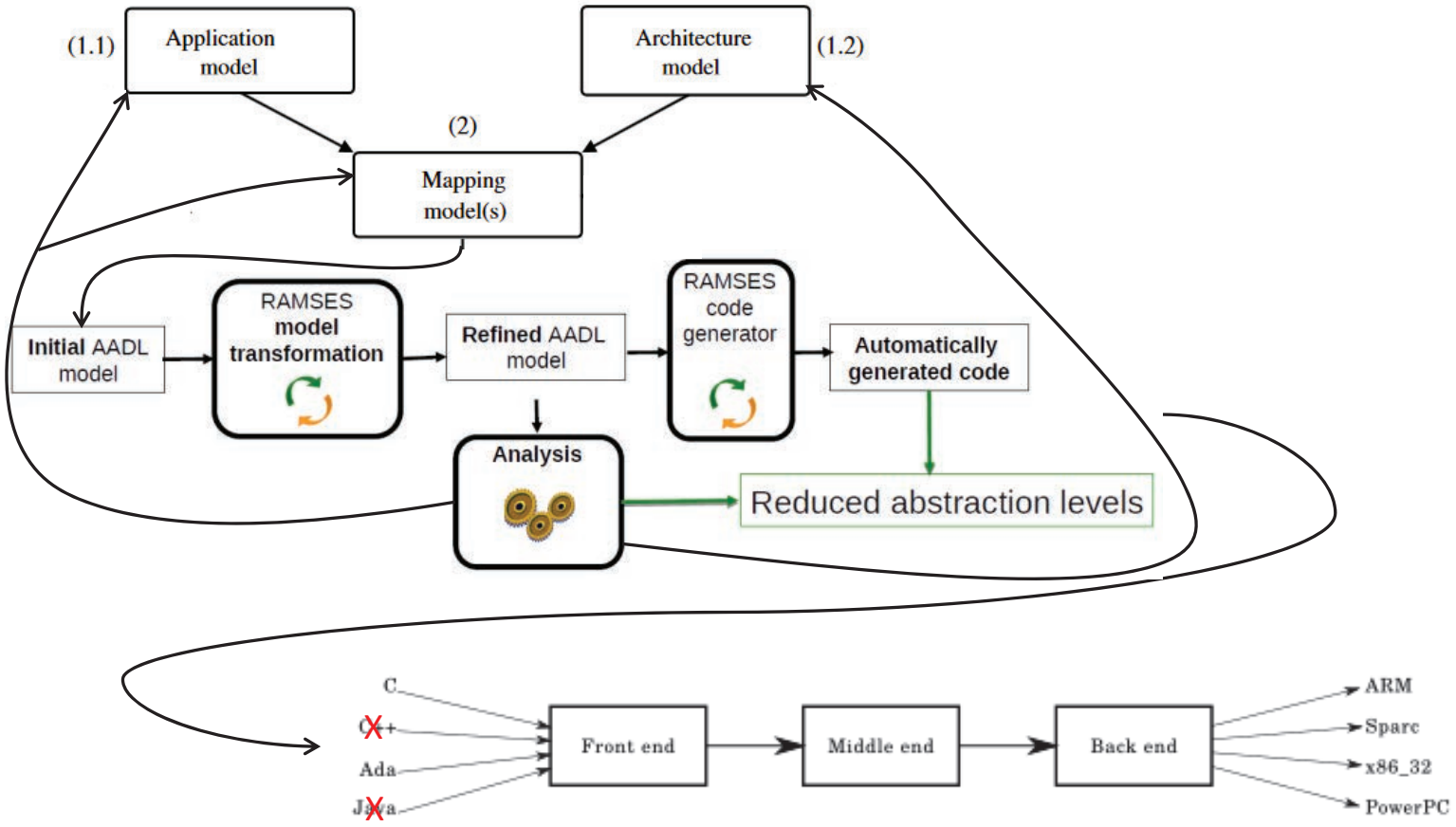
- ❑ Model-Based Engineering
- ❑ Introduction to AADL
- ❑ Timing Analysis with AADL
- ❑ Automatic Code Generation with RAMSES
- ❑ Introduction of the Hands-on Exercise

RAMSES (Refinement of AADL Models for Synthesis of Embedded Systems)

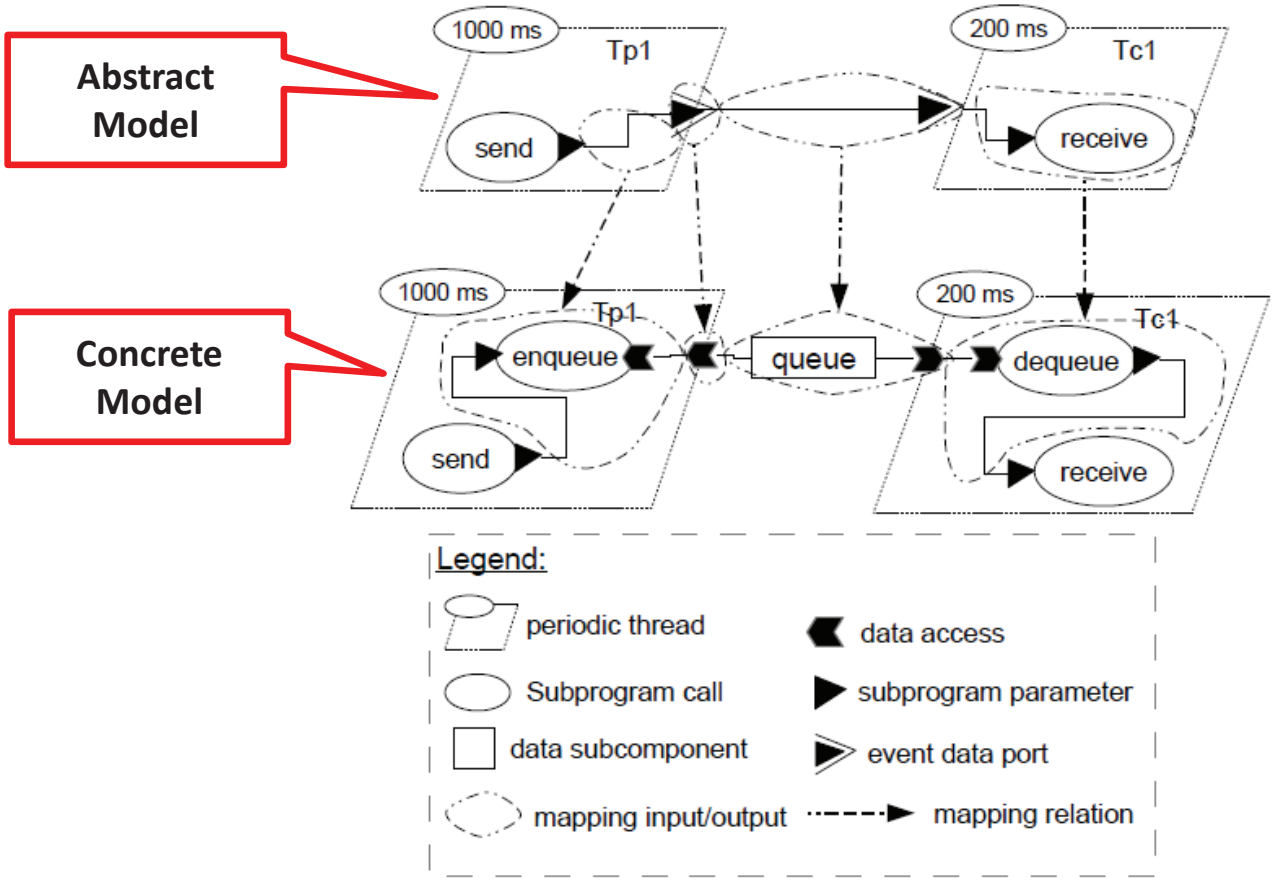


- More info at https://mem4csd.telecom-paristech.fr/blog/index.php/ramses/

Model Refinement and Code Generation



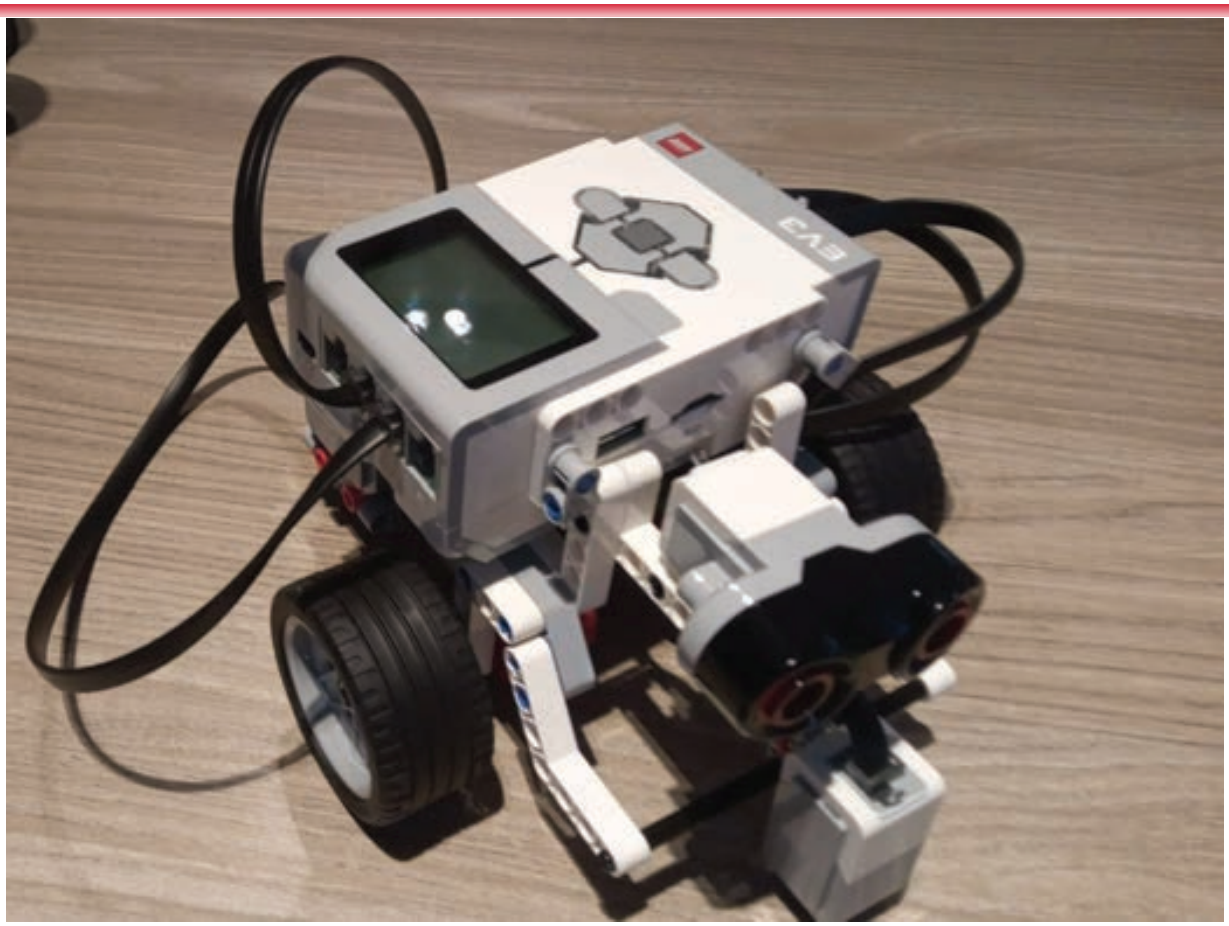
Example of RAMSES Refinement Rule



Outline

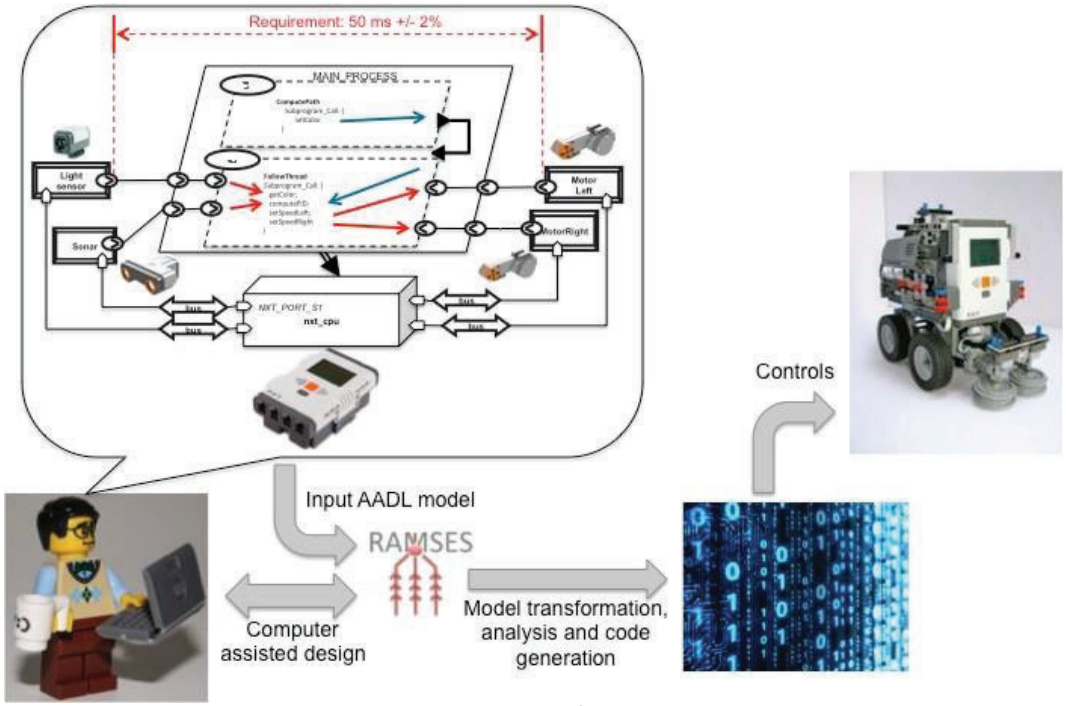
- ❑ Model-Based Engineering
- ❑ Introduction to AADL
- ❑ Timing Analysis with AADL
- ❑ Automatic Code Generation with RAMSES
- ❑ Introduction of the Hands-on Exercise

Case Study: EV3 Dev Minstorm Lego Robot



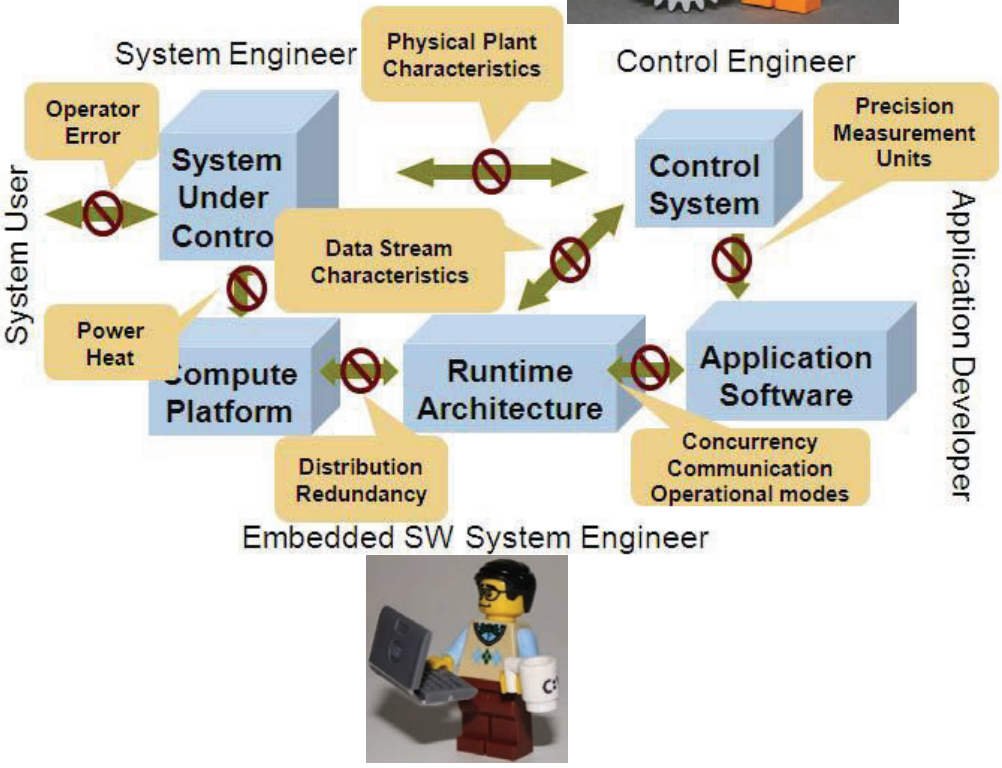
Objectives of the Hands-on Exercise

Understand how to model resource sharing in order to
1 – Ensure it does not jeopardize the program execution because of performance issues
2 – Define timing properties of communicating tasks
3 – Integrate new functions such as an obstacle detection task

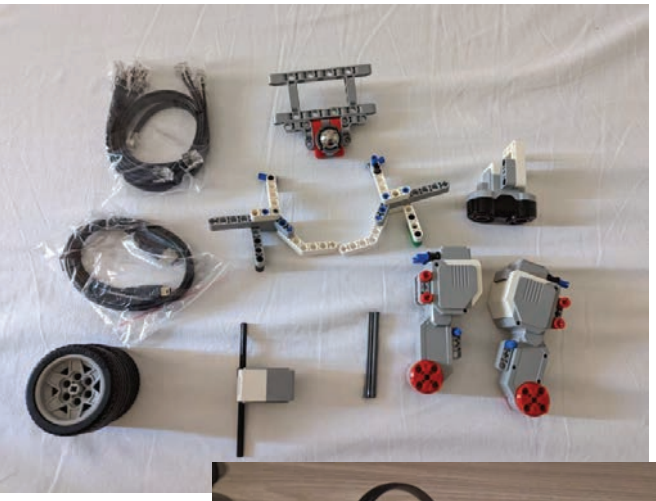


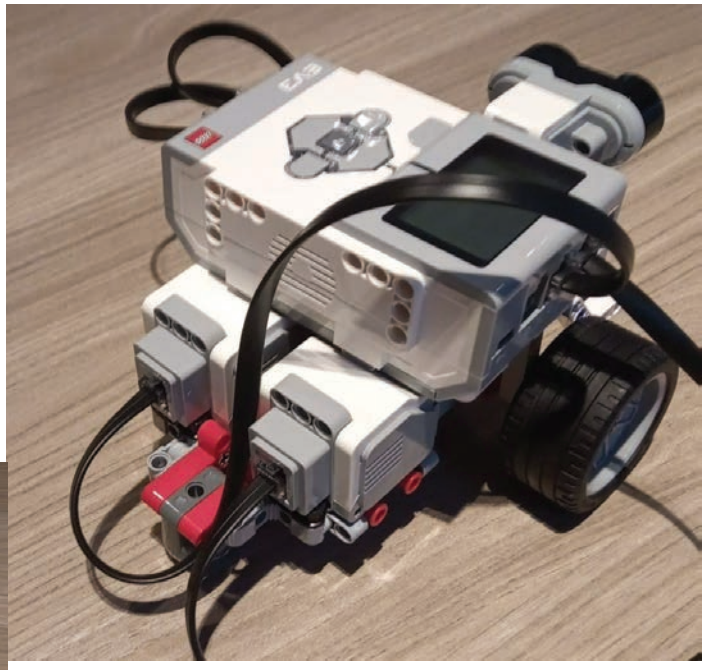
Engineering Teams: Two Domains

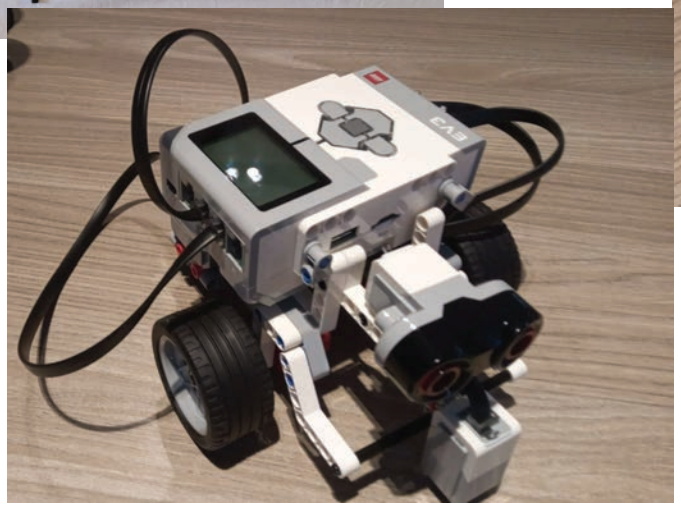




Mechanical Engineering Task: Assemble the Robot







Embedded Software Engineering Task: Develop the Embedded Software

- Follow instructions at:
  - https://mem4csd.telecom-paristech.fr/blog/index.php/training-schools/cps-iot-summer-school-2023/

Making Teams

- 12 robots ➔ How many students per team?
- Requires Linux or Windows
- Linux: Work as is provided that GCC is installed
- Windows 10: Use the Ubuntu Windows Subsystem for Linux (WSL) command line
- Windows 11: Use the Linux GUI WSL

Thank you for your attention

Questions ???

Offloading and Reconfiguration of Functional Components in Embedded Hardware Resources: Application to AI Components

Héctor Posadas, Jose Luis Vázquez
University of Cantabria

DAIS – Distributed Artificial Intelligent systems



DAIS has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No x. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Sweden, Netherlands, Germany, Spain, Denmark, Norway, Portugal, Belgium, Slovenia, Czech Republic, Turkey.

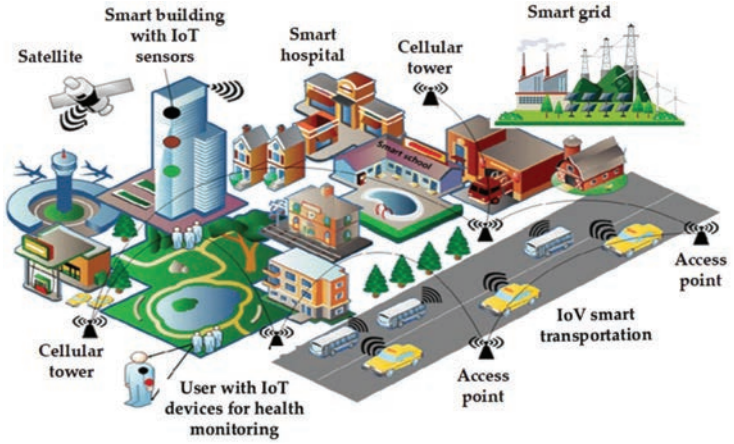


Agenda

- Introduction:
  - Cloud – Fog - Edge
- Model-Driven Design of CPSoS
- S3D Framework
- Execution and communication semantics
  - Fog – Edge application
- Dynamic system reconfiguration
- Application of AI
  - AI modelling in UML
  - Automatic Training
  - Mapping to microcontroller / FPGA

Introduction

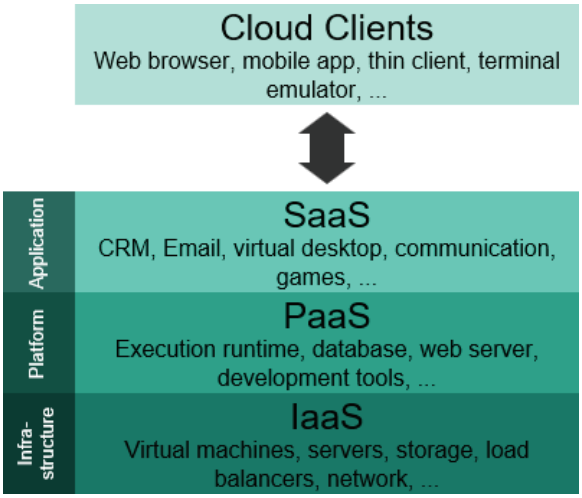
- Services provided on computing platforms of many kind
- Programming the Internet of Everything
  - In close interaction with the physical world => CPSoS-IoT
  - Large amount of data and computation



3

Cloud Computing

- Centralized computing
  - Computing center
  - Request services through Internet
    - **compute power**, **database storage**, **applications**, and other IT resources
    - Flexible, scalable
  - Not expensive
    - Pay per use: low initial investment
    - Typical of OPEX (operational expense) models
      - Vs CAPEX – Capital expense



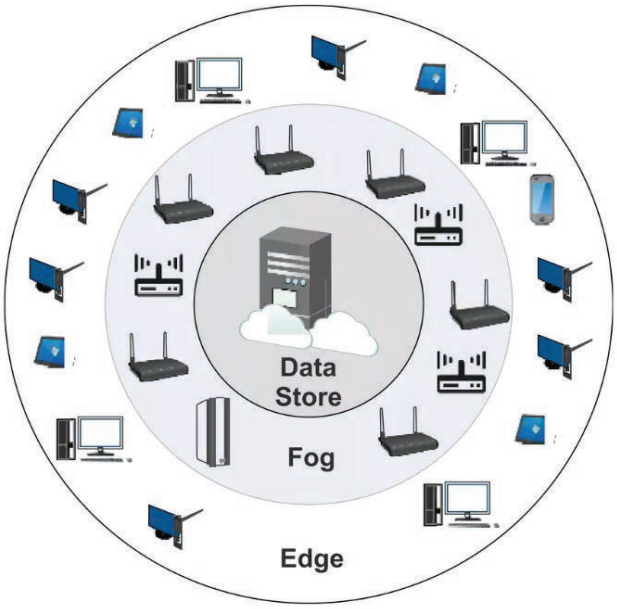
Cloud Computing

- **CyberPhisical layer**
  - Link with the phisical world
    - Sensors, actuators
  - Obtain data for computations
  - Receive results from computation



The Edge of the Cloud

- **Limitations of cloud computing solutions**
  - **Latency** – Real-time interaction
    - Video Streaming – Movies, Video Conferencing
    - Gaming – Role playing games
    - Augmented Reality – Pokémon Go
  - **Mobile Deployment** – Nodes will be moving
    - Cell phones
    - Vehicles
  - **Large number of Nodes**
    - Networks of sensor nodes
    - Large amounts of data
  - **Easier to attack than local computation**
    - Safety, security, privacy



Cloud extension

- **Move computation near the edge**
  - **Edge computing**
    - Computation next to the physical devices
      - One powerful microcontroller can be in charge of one or a few sensors
  - **Fog computing**
    - Extend the cloud to the local environment
      - Computational units in charge of local sensors
      - One PC in charge of all the sensors in the área
  - General characteristics
    - Predominance of **Wireless Access**
    - Strong presence of **Streaming** and **Real-time Applications**
    - **Heterogeneity**

Local computing: Considerations

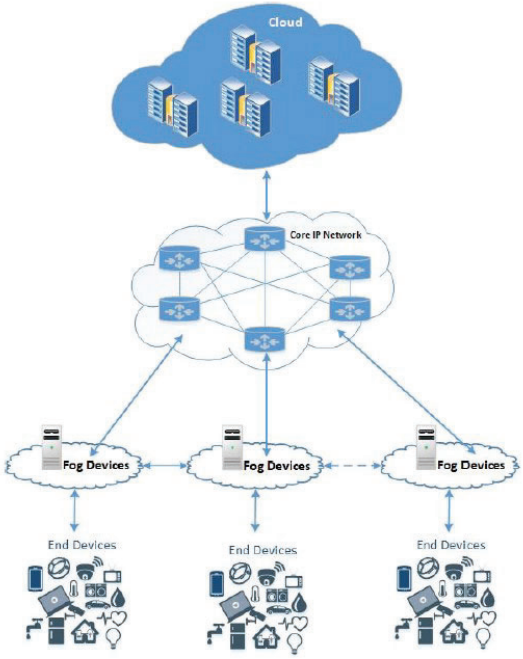
- ***Real-Time IoT Data Analytics***
  - Fog/edge computing enable real-time processing of device data.
  - Clouds are typically for
    - batch processing
    - near real-time processing.
    - historical, comprehensive, and posterior processing
- There are many use cases wherein faster responses are not needed.
  - Turning on the lights, closing the garage door, or checking the vending machine status, and so on come under this category.

Local computing: Considerations

- ***Resource-Constrained Sensors***
  - Many sensors do not have enough compute, memory, and storage power in order to have their own IPv6 address.
  - Hence, they can hide behind an edge device which has the power to have its own IPv6 address.
  - It is, therefore, pertinent to configure ground-level and resource-constrained sensors and actuators behind the edge device.
- ***IoT Data Security***
  - Any external attack on sensors can be stopped at the edge device, which acts as a shelter, strength, stronghold, and savior for feeble sensors.
  - The idea is to prolong the livelihood of sensors.
  - Also, the sensor data are transmitted to edge devices to be stocked and subjected to specific investigations over any local network only.

Fog Computing

- ***Edge devices request compute, storage and communication services from the Fog.***
- ***Fog: Low-latency virtualized services***
  - *is linked to the Cloud Computing infrastructure*
  - *forwards relevant data for computationally intensive processing*



Edge Computing	Fog Computing
Awareness of devices and few services; unaware of the entire domain	Device independent, intelligent, and aware of entire fog domain
Limited control in the edge domain	Controls all devices in the domain
Cloud unaware	Extends cloud to Fog level as a continuum
Limited network scope	Complete network scope
No IoT vertical awareness	Supports and enabler for multiple IoT verticals
Uses Edge Controllers that are focused on edge device command and control	Uses Fog Nodes that are very versatile and capable of performing a variety of functions like RT Control, Application Hosting, and management
Security scope is limitied to devices	End-to-End (E2E) security
Analytics scoped to a single device	Fog Analytics enables collection, processing, and analysis of data from multiple devices in the edge for analysis

Goals

- ***Combine benefits from edge/fog computing***
  - Static / dynamic needs
- ***Support heterogeneity***
  - *Beyond PC / GPU*
    - *Microcontrolers*, FPGA
    - Example: AI - neural network inference
- ***Integrated design flow***
  - *Specification – implementation – documentation*
  - *Easy exploration of design alternatives*

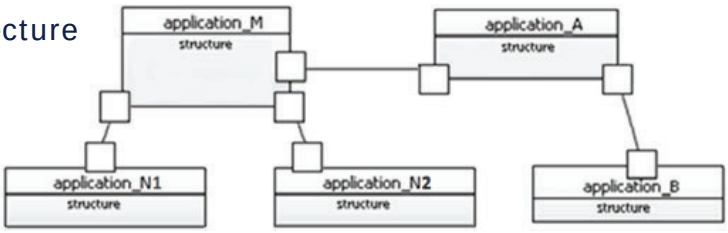
Design of Cyber-Physical SoS

- ***Model-Driven Design (MDD)***
  - High-abstraction level
  - Mature SW engineering methodology
- ***State-of-the-Art***
  - Matlab-Simulink
    - Proprietary, only one MoC, M language
    - Application to UAVs
      - Autopilot + Physics
    - ROS toolbox
  - CoFluent
    - Proprietary, a few MoCs, C/C++ language
  - Ptolemy II
    - Academic, any MoC, C/C++ inside a Java block
  - HEPSYCODE
    - Academic, several MoCs, SystemC
  - …

1

Model-Driven Design

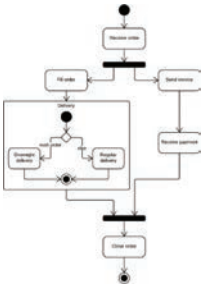
- ***Need for integration / estandarization***
- ***Unified Modeling Language (UML)***
  - OMG Standard, any (user-defined) MoC, any language
  - Natural way to capture system architecture
  - OpenSource tolos
    - Eclipse / Papyrus
  - Lack of semantic
    - Domain-specific profiles
    - Modeling and Analysis of Real-Time and Embedded Systems (MARTE)
      - OMG standard

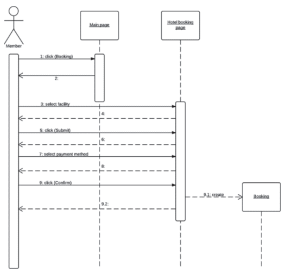


1

UML

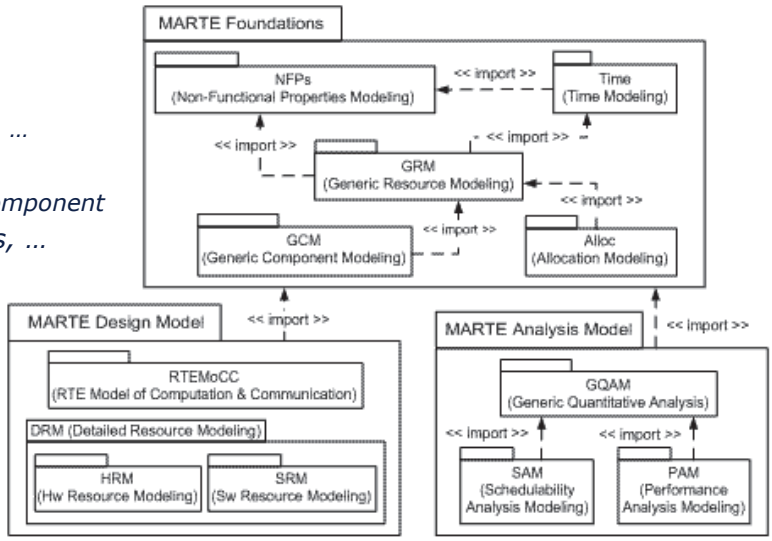
- ***Graphical descripción: diagrams***
  - *Components, activity, sequence, clases, etc.*
- ***XML file***
  - *Textual description. To be used in tools, shared, etc.*
  - *All information in XML, not necessarily in diagrams*
- ***Use***
  - *Specification, graphical view, documentation*
  - *Standard in the components to at disposal*
  - *Non standard in how to specify a system*
  - *+ Input to design tools*
    - *Reuse work, automatic integration of updates, maintenance*





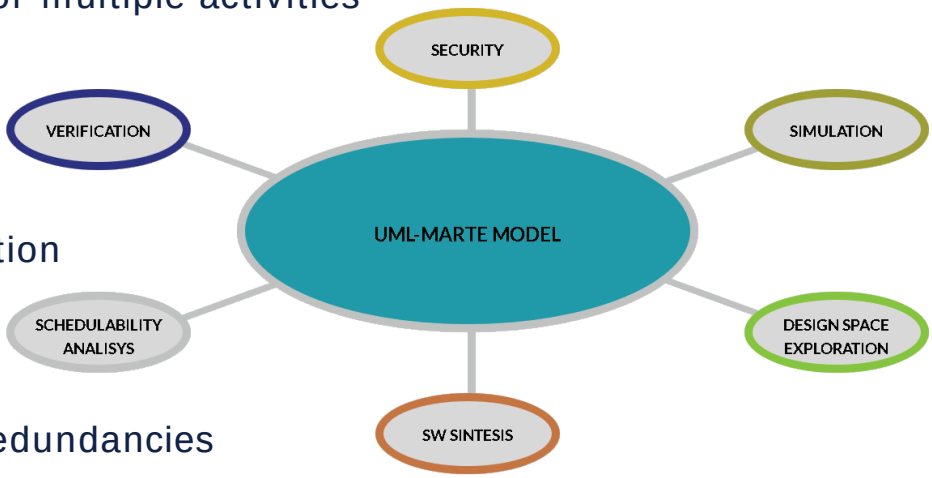
MARTE

- ***Extend UML elements with arguments***
  - *Example: a Component can be*
    - *A HW component*
      - *Processor, memory, bus*
      - *Indicate frequency, size, type, …*
    - *An application component*
      - *active component / passive component*
    - *An operative system, a process, …*



S3D Framework

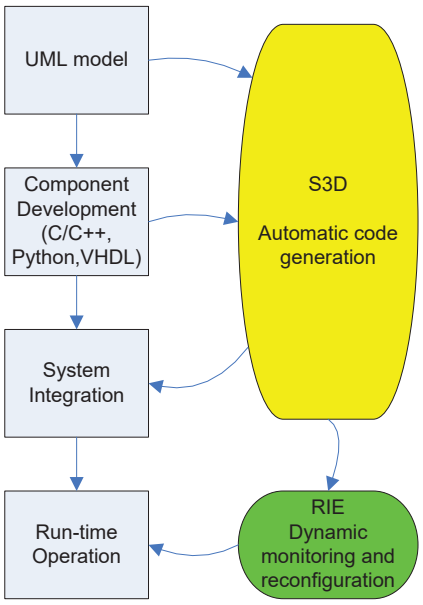
- Single-Source System Design
  - Use a single UML model for multiple activities
    - Modeling
    - Code generation
    - Early Simulation
      - Host-compiled simulation
    - Performance analysis
    - Schedulability analysis
  - Simple modeling: avoid redundancies
  - Focus on communication semantics



17

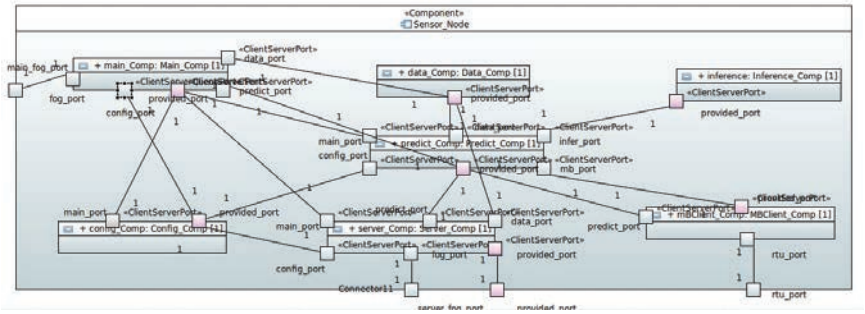
S3D current goals

- Modeling and analysis technologies to support AI components to be mapped on HW
  - Automatic generation of interconnection codes for communication with other functional components
    - Different communication and execution semantics
      - Concurrency
      - HW resources
    - Different resource mapping
      - PC, Microcontroller, FPGA
  - Dynamic component reconfiguration
  - Application to IA components



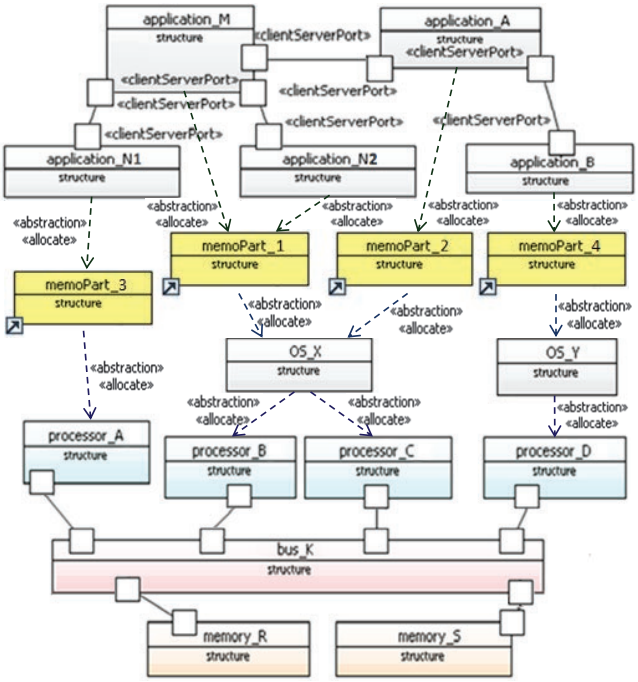
S3D-Functional description

- System defined as set of components
  - UML model
- Component functionality defined with
  - Source files (C++)
  - Diagrams (neural networks)
- Components connected by service calls
  - Ports
    - Interfaces
      - Provided/required
      - Function calls
  - Communication semantics: combines client and server attributes
    - blocking/non blocking, protected/read-write, timeout, buffer, …



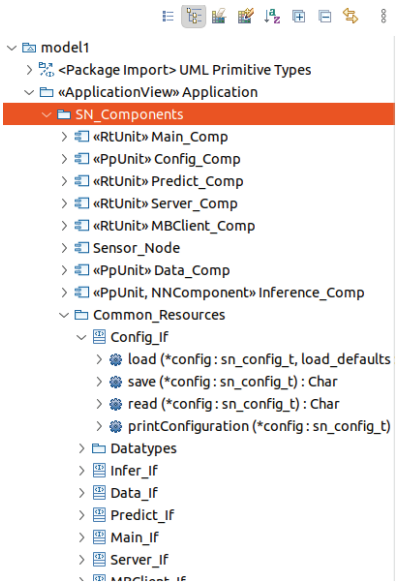
S3D Framework - Modeling

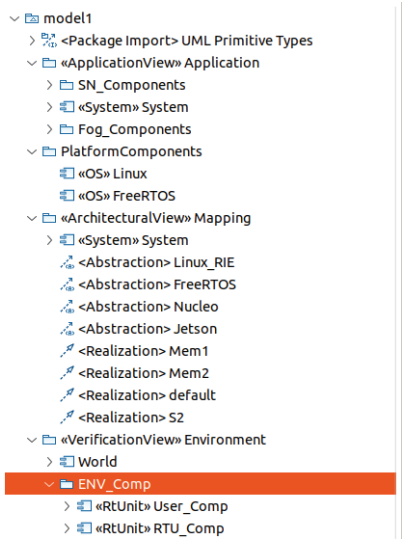
- Architectural (Functional) Design
  - *Code Reuse and/or Development*
    - *Platform Independent*
- Architectural Mapping
- HW/SW Execution Platform



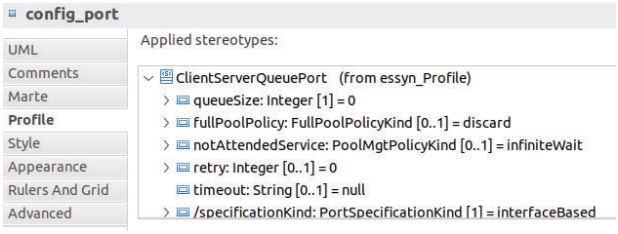
S3D Framework - Modeling

UML Model elements

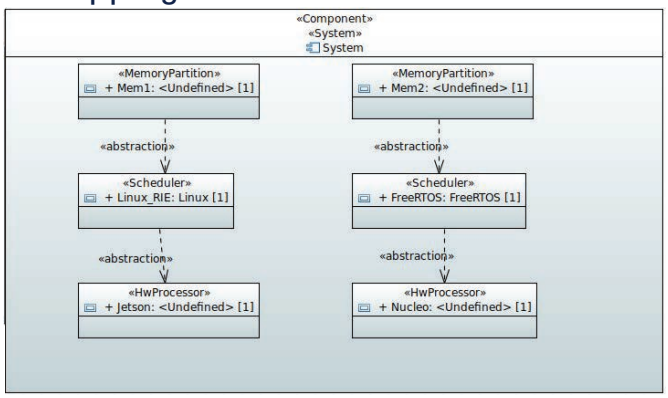




Communication semantics



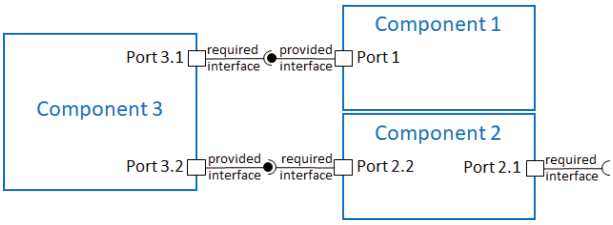
Mapping



2023-04-25

S3D : Code Generation – Commuication and Concurrency

- Automatic implementation of communication and deployment codes
  - Depending on communication semantics (MoCs)
  - Depending on Mapping



- Generation for
  - Functional execution (Host)
  - Execution on target platform
  - Virtual platform

22

Attributes to define communication properties

- Based on UML/MARTE profile
  - Some extensions added

Operation			
	RTService	concPolicy	reader, writer, parallel
		concurrency	sequential, guarded, concurrent
		exeKind	deferred, remoteImmeidate, localImmediate
		syncKind	synchronous, asynchronous, delatedSynchronous, rendezVous
Port			
	ClientServerQueuePort	Extends ClientServerPort	
		fullPoolPolicy	block, removeFirst, removeLast, flush, other
		notAttendedService	intiniteWait, timedWait, dynamic, exception, other
		queueSize	
		retry	
		timeout	
Components			
	RTUnit		
		isMain	
		isDynamic	
		srPoolSize	

23

Derived MoCCs– FOG: Concurrency

- Identify configurations representing models of computation
  - Function Call/RPC/RMI , CSP, Rendezvous, KPN, SDF, DF, ...

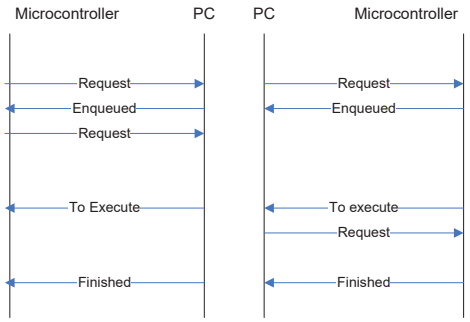
Required Port		RtService			Provided Port		
NotAttendedService	retry	concurrency	exekind	syncKind	queueSize	FullPoolPolicy	MoC
infiniteWait	none	G or C	rem.lm.	sync.	none	none	exactly once
infiniteWait	none	G or C	rem.lm.	async.	none	none	at most once
dynamic	none	G or C	rem.lm.	sync.	none	none	exactly once
dynamic	none	G or C	rem.lm.	async.	none	none	at most once
timedWait	0	G or C	rem.lm.	sync.	none	none	exactly once
timedWait	0	G or C	rem.lm.	async.	none	none	at most once
timedWait	> 0	G or C	rem.lm.	sync.	none	none	at least once
timedWait	> 0	G or C	rem.lm.	async.	none	none	maybe once

Derived MoCCs– Edge: Resource optimization

- Edge Computing: Microcontroller
  - Main Limitations
    - Frequency, Memory size
  - SW platform
    - FreeRTOS, lwIP, EmbedTLS, …
  - Memory consumption
    - Global Memory
      - Functionality / communications
    - Threads (stack)
      - Functionality / communications:

Derived MoCCs– Edge: Resource optimization

- Global Memory
  - Communications: save memory with communication protocol
    - If a requests is enqueued, where to store data: client / server ?
    - Alternatives in protocol



- Threads (stack)
  - Functionality / communications:
    - Threads waiting for incoming packages, threads executing requested services
    - Optimization: Co-routines, hooks
      - Requirements : Avoid blockings
      - Solution: use communication semantics to generate temporary threads for blocking acceses
    - Thread stack: constant to avoid memory fragmentation

Dynamic reconfiguration

- **Adapt to system conditions**
  - Processing elements
    - CPU utilization, memory, deadlines
  - Computational requirements
    - Activate / deactivate services
    - Number of incoming requests
    - Different code: output accuracy, input source conditions
  - Other platform elements
    - Required services not available
    - Communication bottlenecks

Dynamic reconfiguration

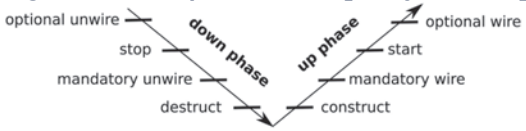
- **Static / dynamic support**
  - Different code: S3D
  - Run-time support: RIE
- **RIE: Reconfigurable Implementation of Embedded systems**
  - Implementation-oriented C++ library
  - Provide application-independent reconfiguration infrastructure
    - Low performance impact
    - Avoid dead time during reconfiguration
  - Provide runtime monitoring infrastructure
    - Development of runtime adaptation infrastructure

Remote Component Implementations

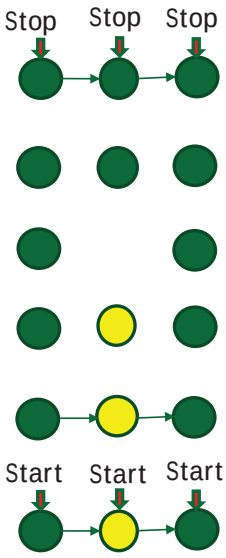
- **Component views**
  - Local component implementation
    - Access to remote component services
    - Manage remote component
    - Remote component monitoring
  - Remote component implementation
    - Component server implementation
    - Provide component functionality
- **SW Platforms supported**
  - Linux
  - FreeRTOS(CMSIS) – lwIP (tested in STM32)
- **Communication**
  - Edge-Fog: TCP/IP
  - Fog-cloud: gRPC

Dynamic reconfigurable systems

- Reconfiguration process [Boy2017]



  - Stop: it informs a module that it should stop processing its input messages and generating output messages until it is started again.
  - Unwire: remove node connections
  - Destruct: Remove old node
  - Construct: Create the new node
  - Wire: establish new connection
  - Start: launch node execution



Initial Configuration

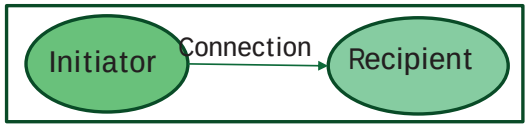
Final Configuration

[Boy2017] Boyer, F., Gruber, O., and Pous, D. (2017) A robust reconfiguration protocol for the dynamic update of component-based software systems. *Softw. Pract. Exper.*, 47: 1729– 1753

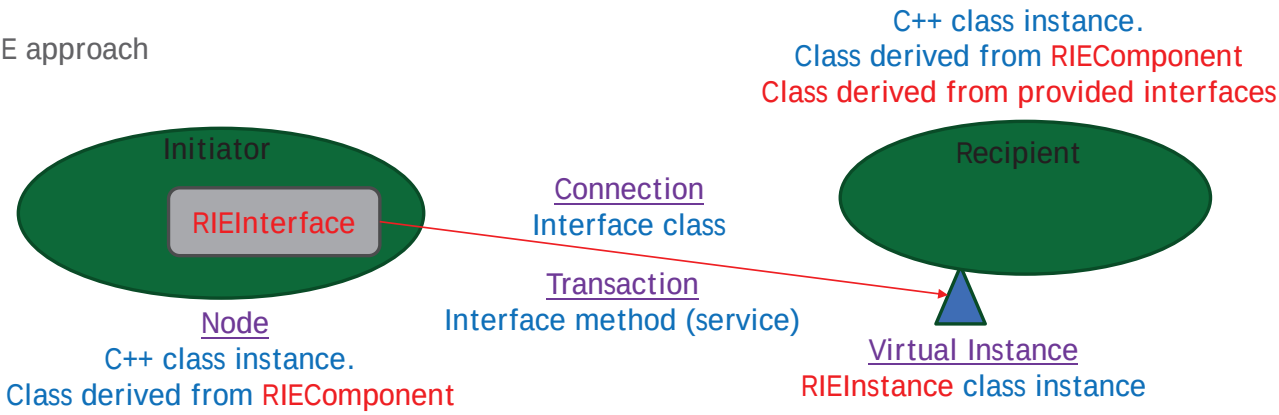
RIE approach

- Provide application-independent reconfiguration infrastructure
  - Low performance impact
  - Avoid dead time during reconfiguration
- Provide runtime monitoring infrastructure

- Classical



- RIE approach

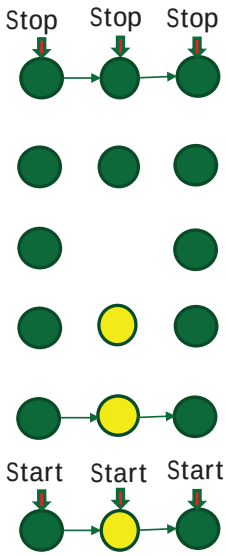


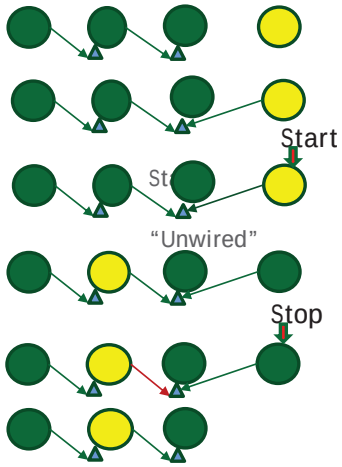
RIE-based reconfiguration

Classical

RIE approach

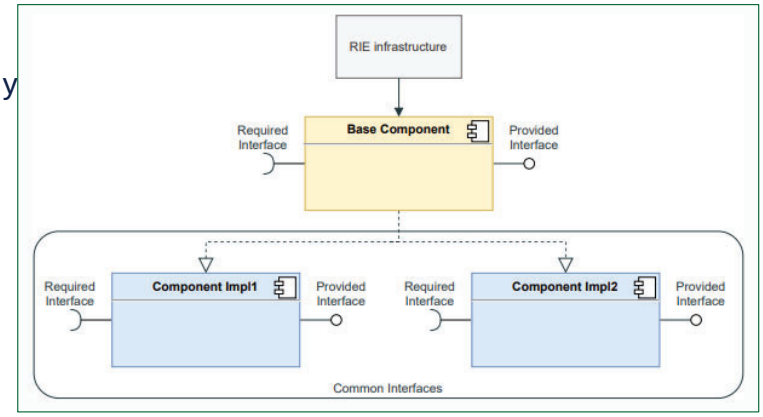
- Stop: Construct
- Unwire Wire
- Destruct
- Construct
- Wire Stop
- Start Destruct





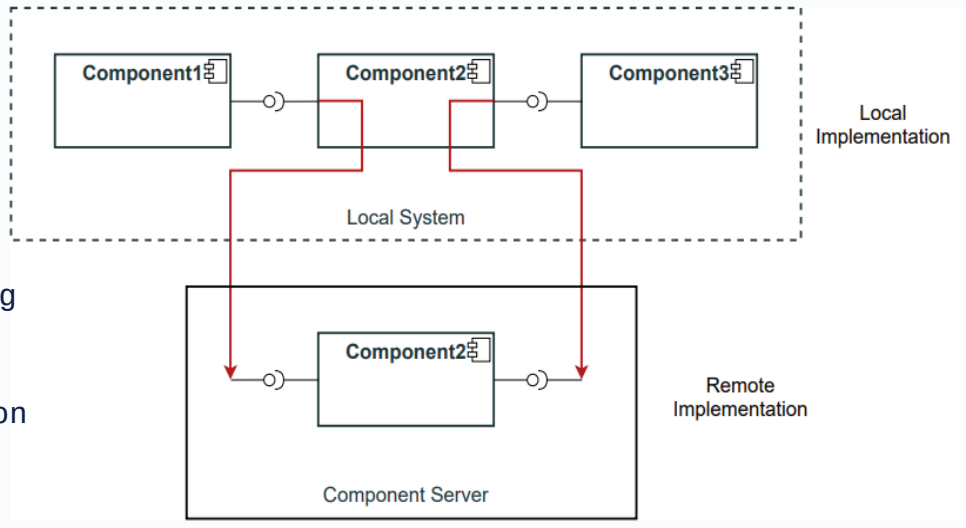
RIE Components

- Component = C++ class that derives from RIE infrastructure (RIEComponent class)
  - RIEComponent functionality
    - Component identification
      - Component instance, class and library
    - Component management lists
      - Provided and required interface list
    - Instance list (hierarchical components)
      - Complete component instance list
    - Component configuration
    - Component reconfiguration functions
    - Component runtime monitoring



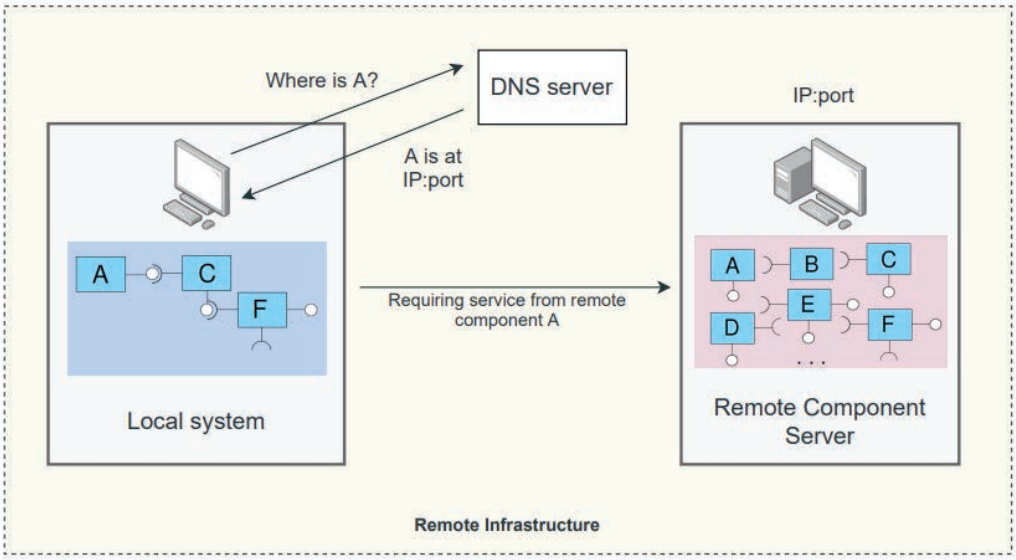
Remote Component Implementations

- Component views
  - Local component implementation
    - Access to remote component services
    - Manage remote component
    - Remote component monitoring
  - Remote component implementation
    - Component server implementation
    - Provide component functionality

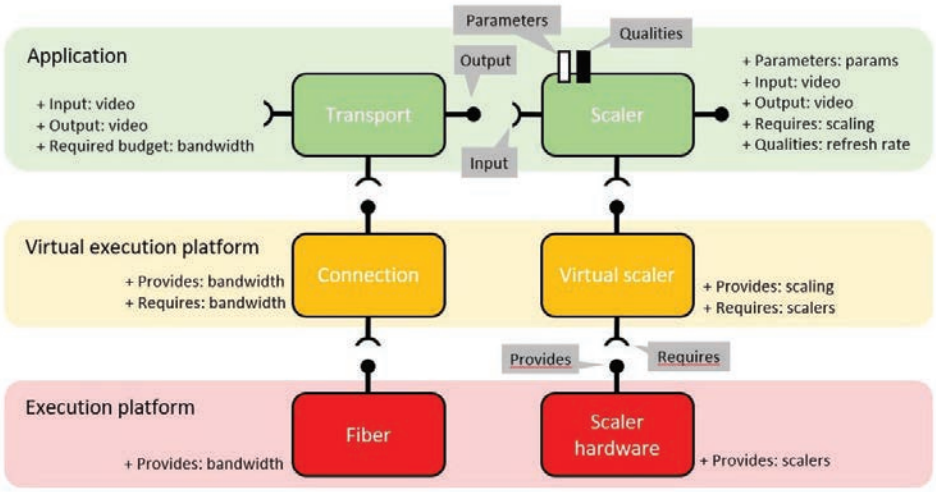


Service discovering

- Component DNS server



Reconfiguration: System Status



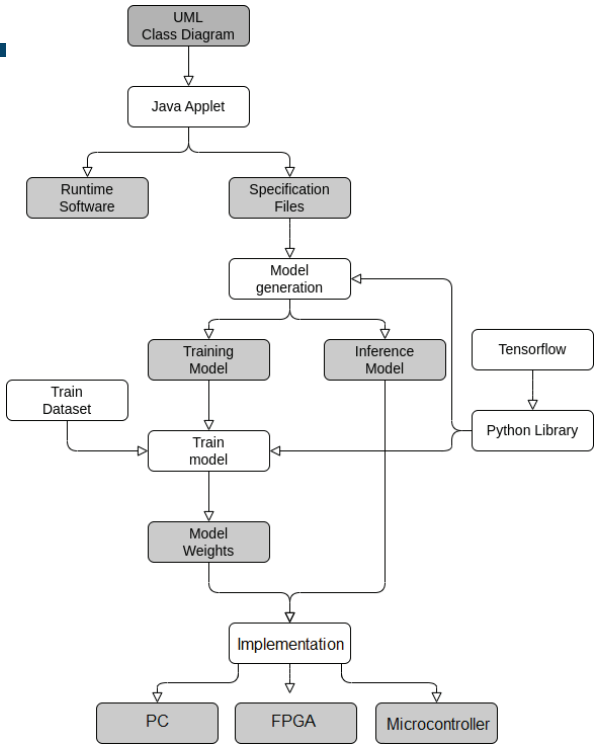
Application layer

Virtual platform layer:
abstract physical platform details

Physical platform

Application to Artificial Intelligence

- Integration of Artificial Neural Networks
  - Automatic procedure
  - Integrated with development environment
  - Easy to explore alternatives
- Steps
  - Create the network model
  - Train the model
  - Adapat to the platform
  - Integrate with the rest of the system

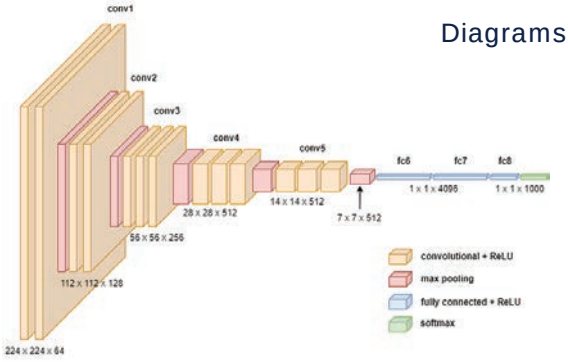


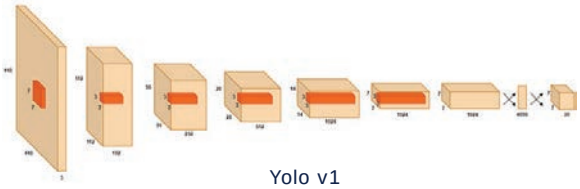
37

Application to Artificial Intelligence

- Integration of Artificial Neural Networks
  - Generate the initial network model
    - Design / visualization
      - Textual descripción
      - Graphical description
    - Proposal: UML graphical description
      - Hierarchical model
      - Specific stereotypes
        - Include layer information
        - Include training information
    - Generate:
      - Network model configuration file
      - Python scripts for training

Diagrams.net

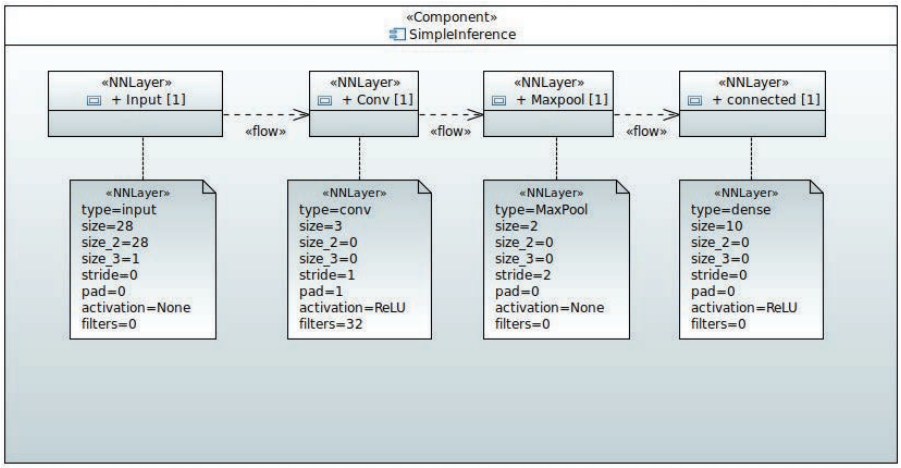




Yolo v1

How to Easily Draw Neural Network Architecture Diagrams | by Kenneth Leung | Towards Data Science

UML modeling of neural networks





```
[input]

dim0=28
dim1=28
dim2=1

[convolution]
filters=32
size=3
stride=1
pad=1
activation=relu

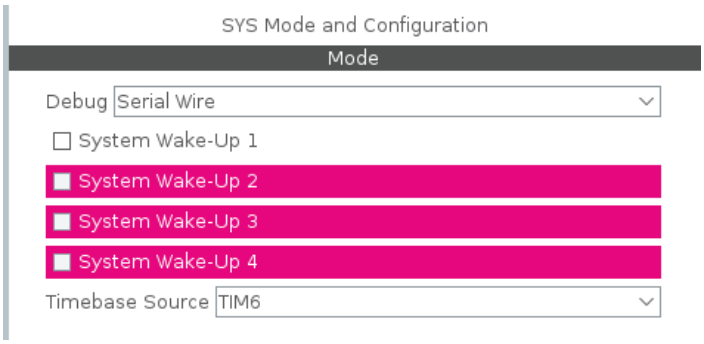
[maxpool]
size=2
stride=2

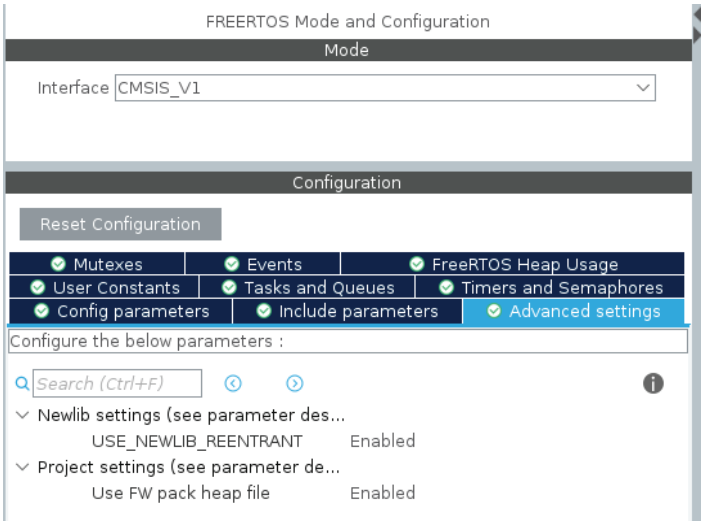
[connected]
outputs=10
activation=relu
```

39

Edge: Implementation - Microcontroller

- Example: STM32 microcontrollers
  - Device STM32F756
    - ARM® 32-bit Cortex®-M7 with FPU – 256MHz
    - 1MB of Flash memory / SRAM: 320KB
  - Tool: STM32 X-CUBE-AI +
    - FreeRTOS + LWIP socket client





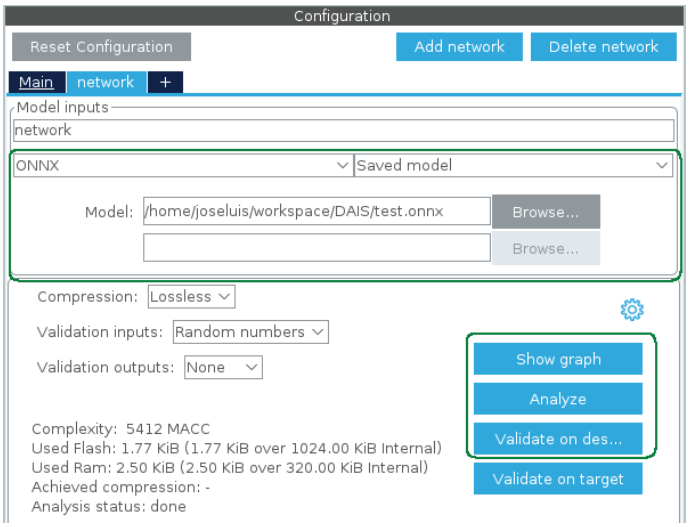
Initial network configuration

- Automatic C code generation of AI models in STM32Cuve tool
  - X-CUBE-AI requires a pre-trained model from KERAS, Tensorflow o ONNX
- Implement model
  - Create base Project on STM32Cube tool with IOC
    - Activate Middleware: FreeRTOS, LWIP
    - In Pinout and Configuration tab of IOC:
      - Software Packs > Select Components
    - In Project Manager tab
      - Increase minimal heap & stack



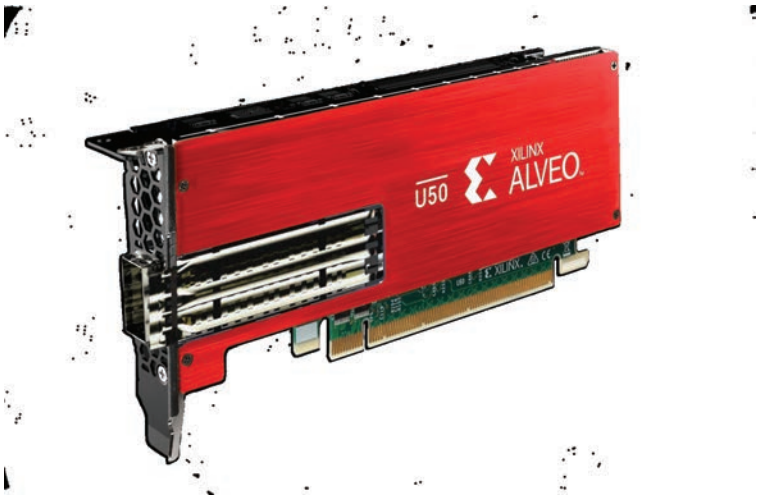
Configuration X-CUBE-AI

- Support several compression alternatives
  - Reports requires requirements to adapt selection to platform
- Recommended to validate the model
  - Compression could change accuracy

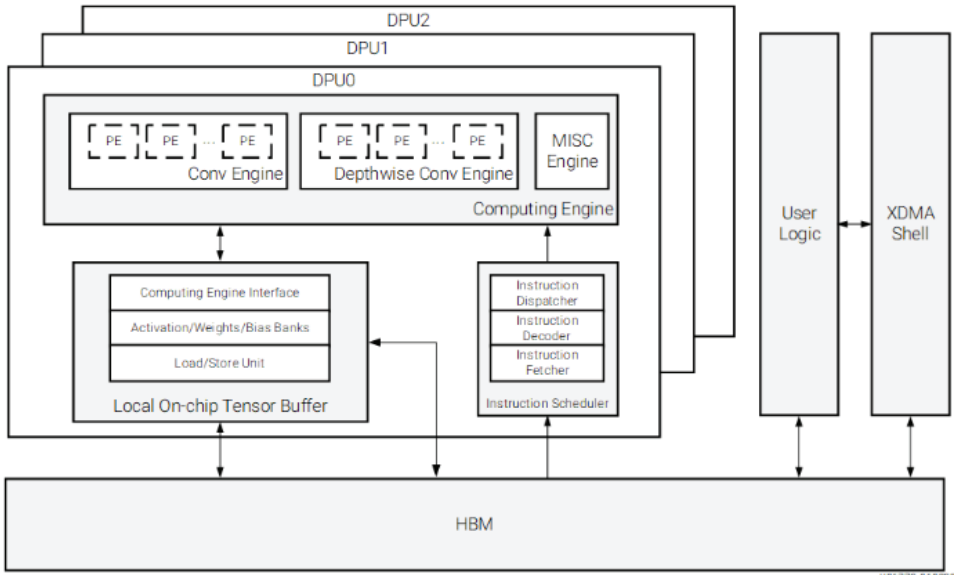


AI implementation in FPGA

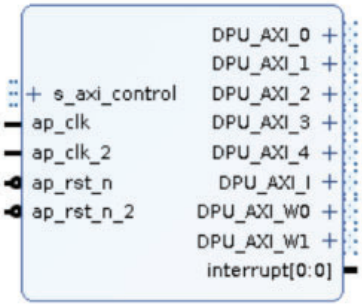
- S3D (WP2): UML -> Vitis AI Run-Time
- Hardware: Alveo (A50), Ultrascale (ZCU111)
- Workflow
  - Model definition
  - Train
  - Freeze
  - Quantize
  - Compile



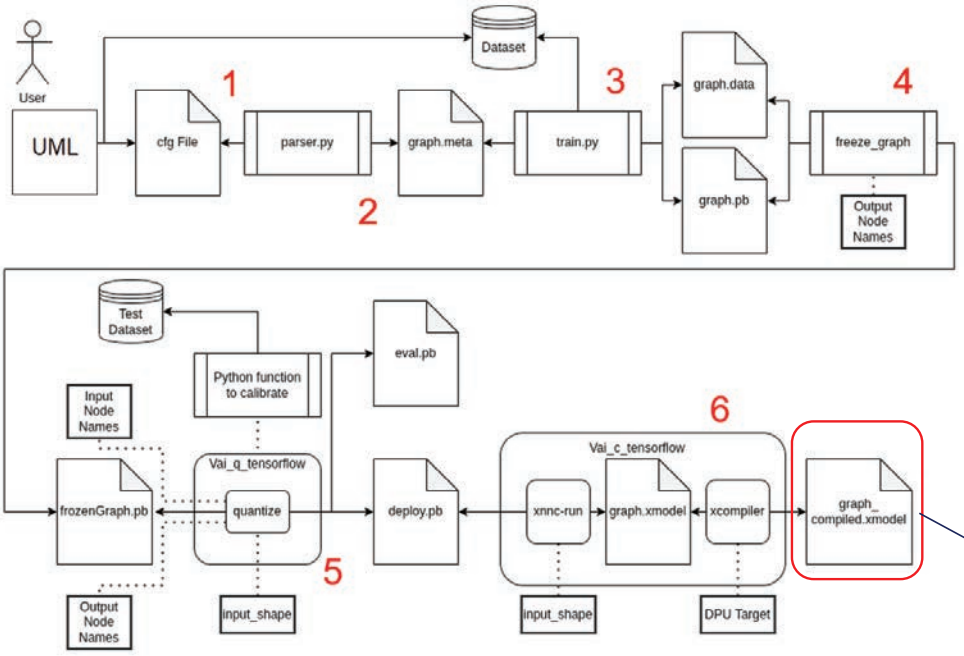
Hardware architecture



- ALVEO U50
  - DPUCAHX8H
  - 2 cores (3 PEs + 3 Pes)
  - 300 MHz
  - Peak Performance (GOPS): 7373



Complete Workflow



1.Model definition

2.Tensorflow graph

3.Train graph

4.Freeze graph

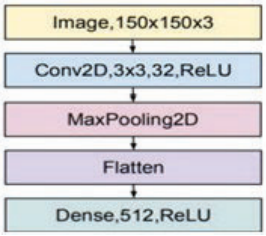
5.Quantize graph

6.Compile graph

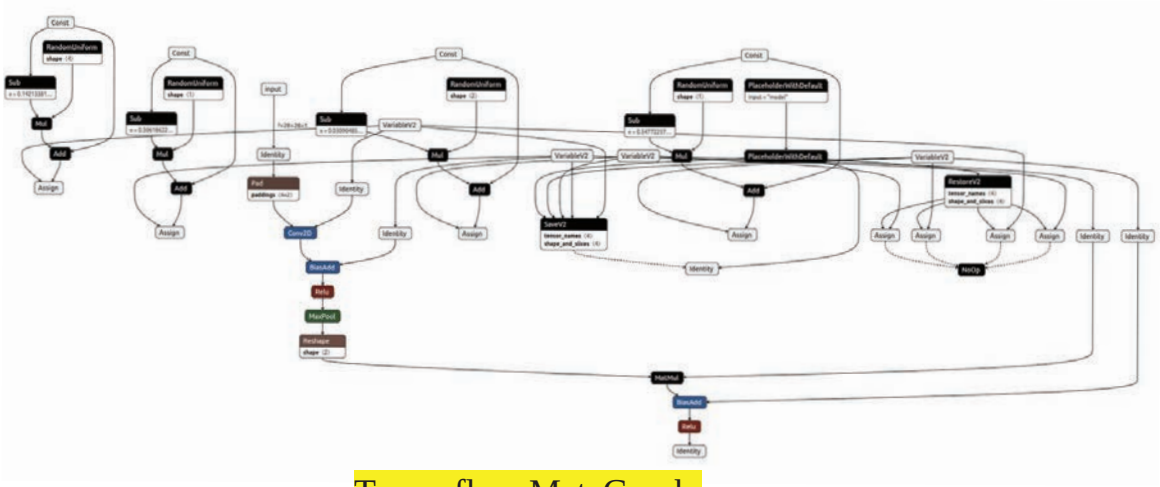
45

Model definition

UML

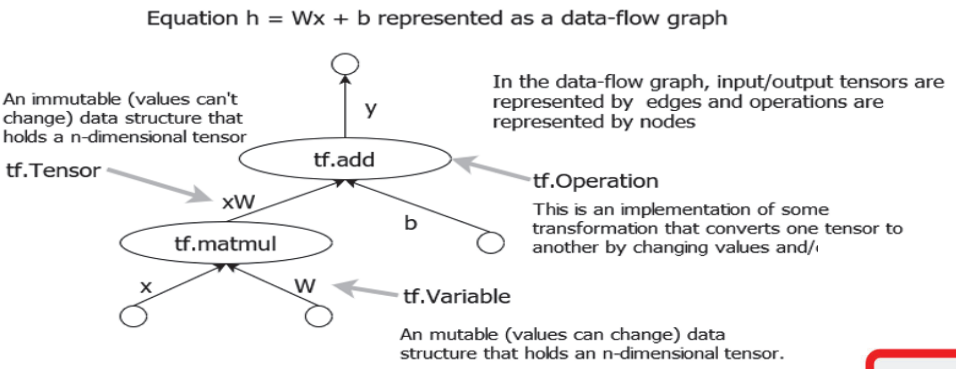


Deep Learning Framework (Tensorflow, Pytorch, Caffe, Darknet...)



Tensorflow MetaGraph

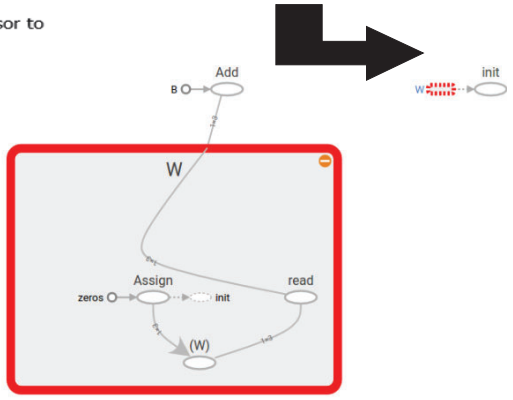
Freeze

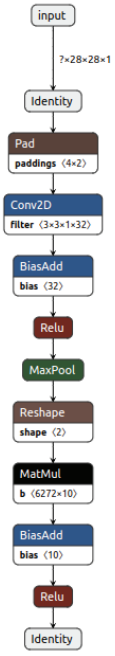


Frozen graph

✓Const Nodes

✗Variables Nodes





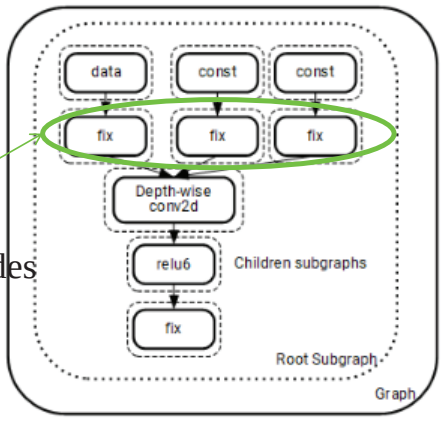
- W is a Variable Node
- B is a Constant Node

Compiling for FPGA

Frameworks

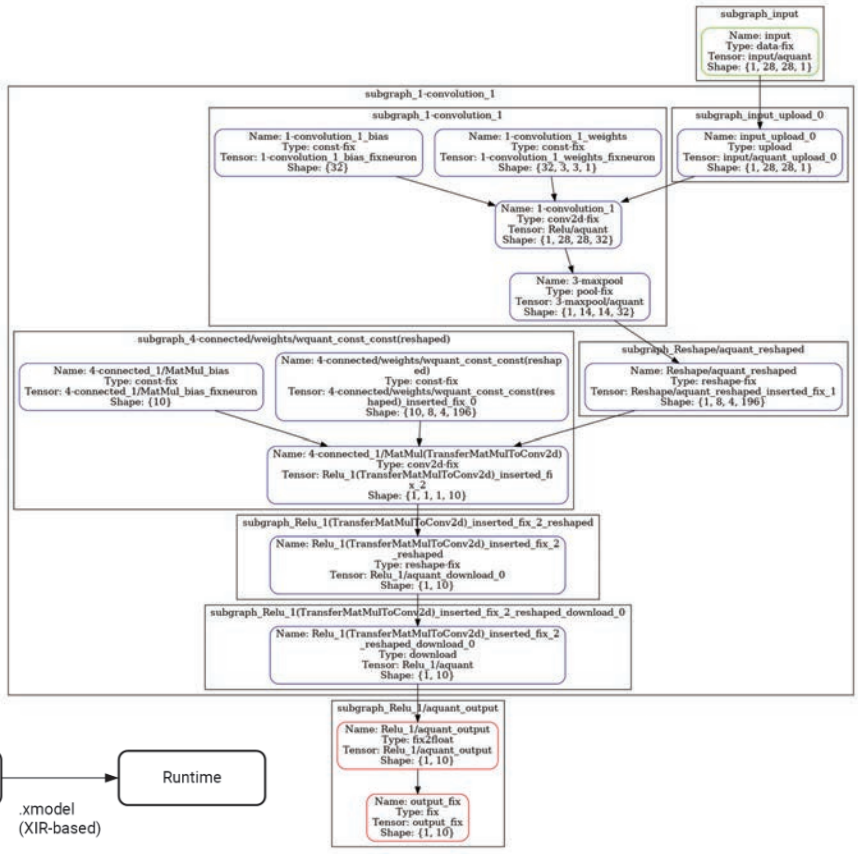
Tensorflow

XIR

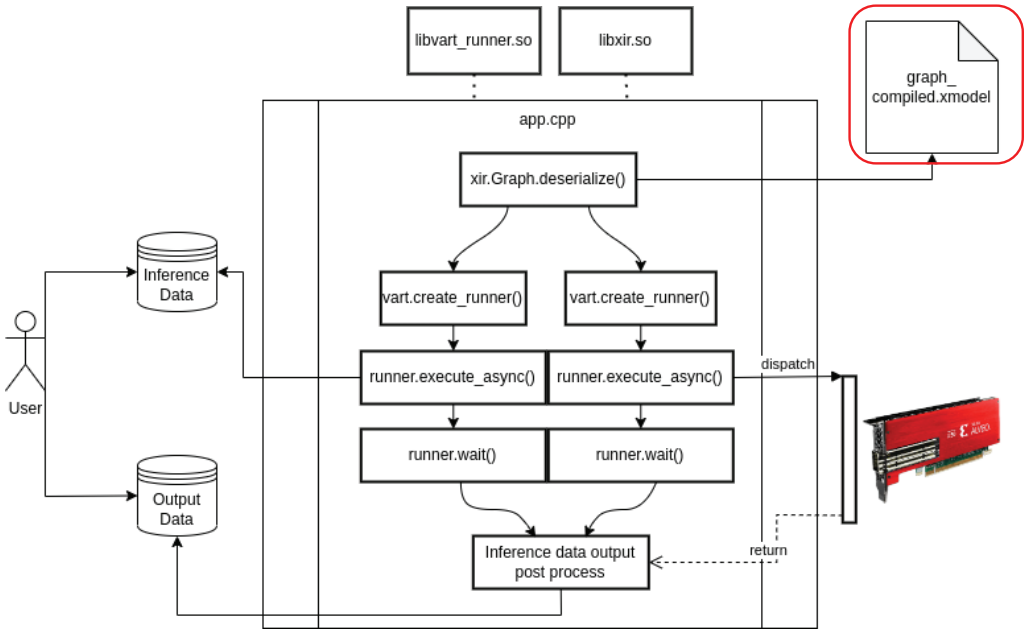


Quantized nodes

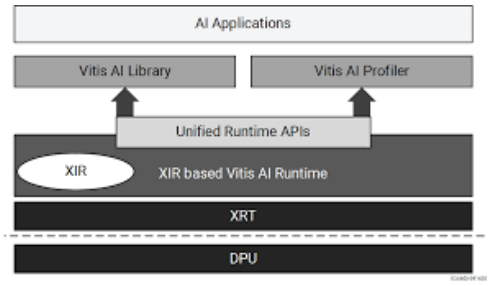




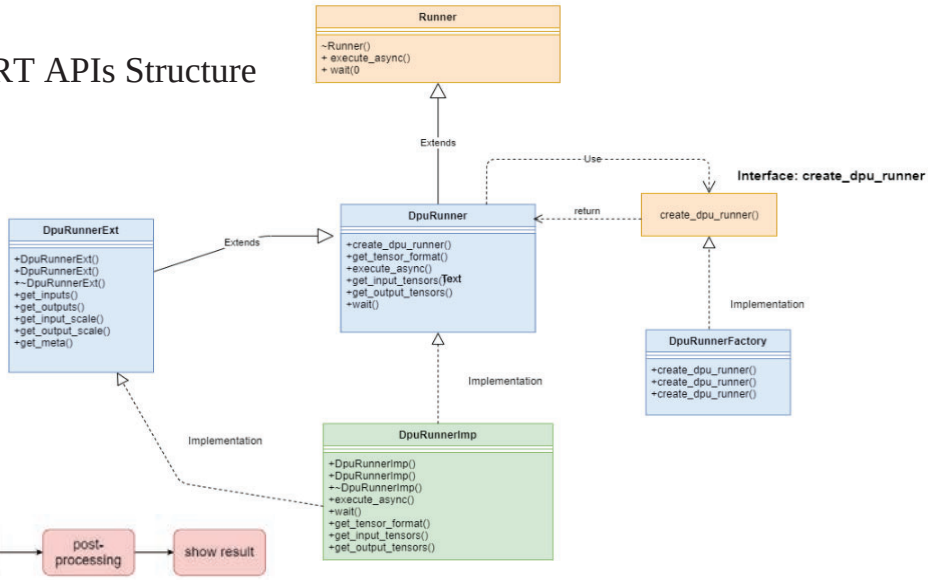
Execution Model



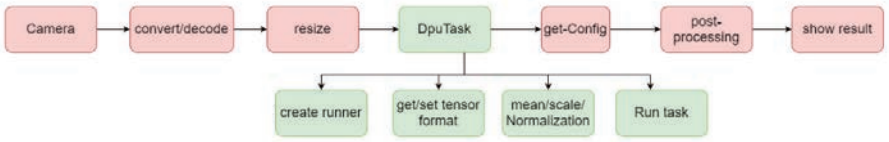
Vitis AI RunTime (VART)



VART APIs Structure



General processing flow



FPGA - Time Results

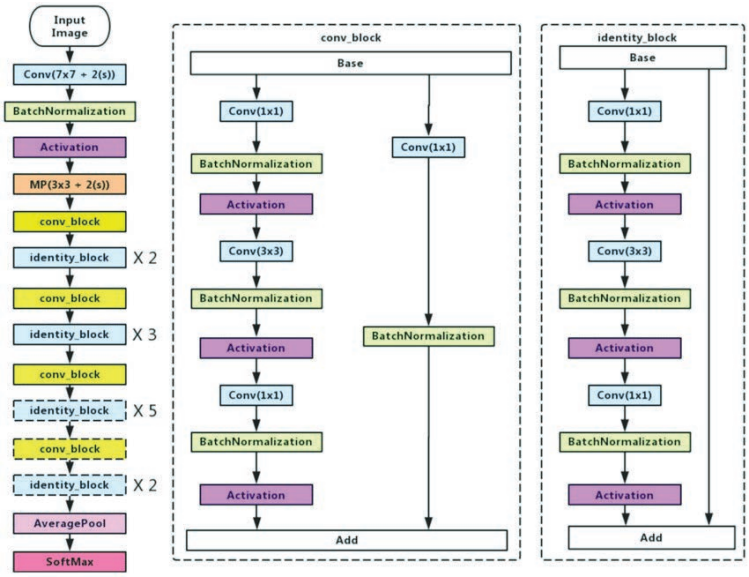


→10000 Images
→126.7 Sec
→12.7 ms/Inference



→10000 Images
→21.6 Sec
→2.16 ms/Inference

Resnet50

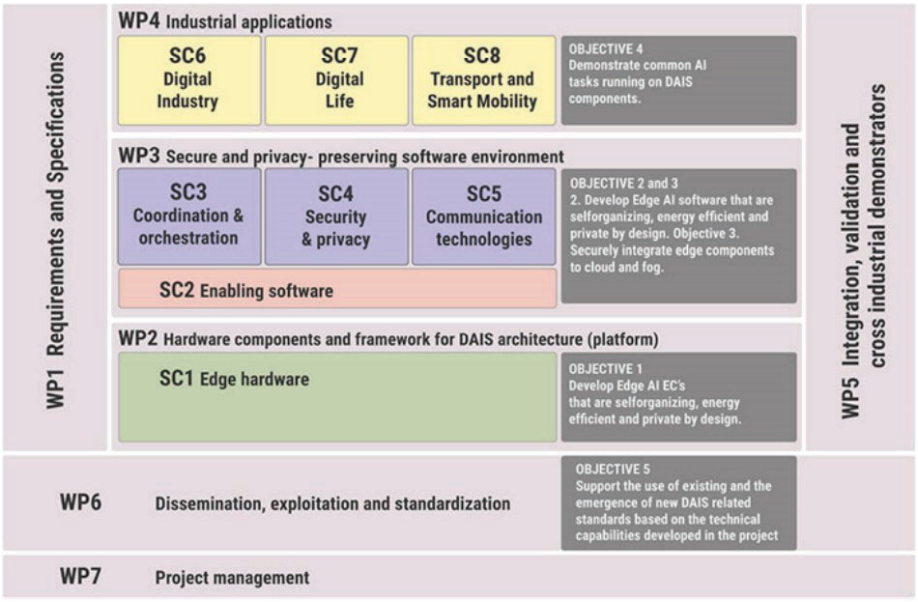


Application example and development

- Distributed Arficial Intelligence Systems (DAIS)
  - European Project
    - H2020



DAIS Work Packages



Use Case SC6.2 (WP4.2.2)

- Energy management
  - Edge/fog/cloud
- ITI: Cloud infrastructure
- 7Sols, UGR: Time Sensitive Network (TSN)
- Schneider: Remote Terminal Unit (RTU)
  - Monitoring transformer: voltaje, power, oil temp.
- Gradiant: IA in FPGA
- TST: Microcontroller board
- RISE: Neural network development
- UC: UML and middleware support

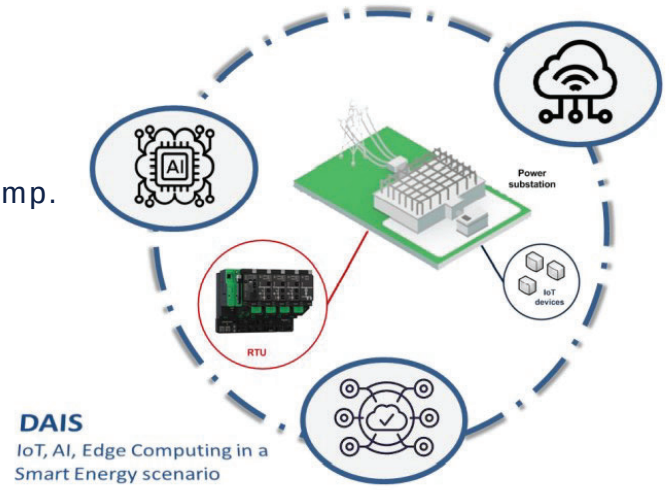


Figure 12: Energy management scenario based on fog computing approach



avn group
innovating for people



CI/CD automations for complex pipelines with hardware in the loop

The XANDAR Use Case

Dimitris KARADIMAS – dimitris@avntechgroup.com
Innovative Technology Solutions (AVN), Cyprus





Outline

Background Session

1. Introduction
2. Workflow
3. Implementing CI/CD
4. Tools
5. Best Practices
6. Challenges
7. Case studies
8. Jenkins
9. Future

Hands on Session

- Live demonstration with practical examples





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Introduction

- CI/CD is a development approach that emphasizes frequent integration and deployment of code changes into production.
- It aims to reduce the time it takes to deliver new features to customers by automating the process of building, testing, and deploying software.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD and DevOps

- DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to improve collaboration and automate the process of software delivery.
- DevOps aims to break down the silos between development and operations teams and create a culture of collaboration and communication.
- DevOps practices include continuous integration, continuous delivery, and continuous deployment, which enable teams to deliver high-quality software quickly and reliably.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD and DevOps (contd.)

- DevOps also emphasizes automation, infrastructure as code, and monitoring and feedback to increase efficiency and reduce errors.
- The goal of DevOps is to enable faster time-to-market, better product quality, and increased customer satisfaction.
- DevOps is not a specific tool or technology, but rather a philosophy and a way of working that requires a shift in culture and mindset.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





How CI/CD fits into DevOps

CI/CD fits into DevOps by:

- enabling teams to deliver high-quality software quickly and reliably
- fostering a culture of collaboration, automation, and continuous improvement

The combination of DevOps practices and CI/CD enables teams to:

- respond quickly to changing business requirements and customer needs
- continuously deliver value to the organization.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD in Embedded Systems

- CI/CD practices can be applied to embedded systems development to enable faster and more reliable delivery of software updates to devices.
- Embedded systems often have hardware dependencies and constraints that require special considerations when implementing CI/CD pipelines.
- CI/CD can help automate the process of building, testing, and deploying firmware updates to embedded devices, reducing the risk of errors and improving the reliability of the software.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD in Embedded Systems (contd.)

- CI/CD can also enable faster time-to-market for embedded systems, which is important in industries where rapid innovation and product differentiation are key factors.
- CI/CD can bring significant benefits to embedded systems development, enabling faster time-to-market, improved software quality, and better customer satisfaction.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Benefits of CI/CD

- Faster Time to Market
  - By automating the build, test, and deployment process, CI/CD reduces the time it takes to deliver new features to customers.
  - This enables companies to respond more quickly to changing market demands and gain a competitive advantage.
- Increased Quality
  - By automating testing and deployment, CI/CD reduces the risk of introducing bugs to production.
  - This leads to higher-quality software and a better user experience.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Benefits of CI/CD (contd.)

- Reduced Risk
  - By automating the build and deployment process, CI/CD reduces the risk of human error during manual deployment.
  - This results in more reliable software and fewer production incidents.
- Increased Collaboration
  - CI/CD encourages collaboration between developers and operations teams.
  - By sharing code and automating the build and deployment process, development and operations teams can work more closely together.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Benefits of CI/CD (contd.)

- Improved Feedback Loop
  - CI/CD provides faster feedback on code changes.
  - By automating the build and testing process, developers can receive feedback on code changes more quickly, enabling them to make corrections earlier in the development process.
- Lower Costs
  - By automating the build, test, and deployment process, CI/CD reduces the need for manual intervention.
  - This results in lower labor costs and fewer production incidents, leading to cost savings for the company.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Workflow

1. Continuous Integration
2. Continuous Deployment
3. Continuous Delivery





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Workflow / Continuous Integration

Definition:

Continuous Integration (CI) is a development practice where developers frequently merge their code changes into a shared repository, triggering an automated build and testing process.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Workflow / Continuous Integration

- Step 1: Code changes: Developers make changes to the codebase and push them to the shared repository.
- Step 2: Build: An automated build process pulls the latest code from the repository and compiles it.
- Step 3: Test: Automated tests are run to verify that the code changes have not introduced any regressions.
- Step 4: Report: The results of the build and tests are reported to the development team.
- Step 5: Merge: If the build and tests are successful, the code changes are merged into the main branch of the codebase.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Workflow / Continuous Deployment

Definition:

Continuous Deployment (CD) is a development practice where code changes are automatically deployed to production after passing a series of automated tests.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Workflow / Continuous Deployment

- Step 1: Code changes: Developers make changes to the codebase and push them to the shared repository.
- Step 2: Build: An automated build process pulls the latest code from the repository and compiles it.
- Step 3: Test: Automated tests are run to verify that the code changes have not introduced any regressions.
- Step 4: Deploy to staging: If the build and tests are successful, the code changes are automatically deployed to a staging environment for further testing.
- Step 5: Test in staging: The code changes are tested in the staging environment to ensure that they work correctly in a production-like environment.
- Step 6: Deploy to production: If the tests in the staging environment pass, the code changes are automatically deployed to production.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Workflow / Continuous Delivery

Definition:

Continuous Delivery (CD) is a development practice where code changes are automatically built, tested, and prepared for release to production.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Workflow / Continuous Delivery

- Step 1: Code changes: Developers make changes to the codebase and push them to the shared repository.
- Step 2: Build: An automated build process pulls the latest code from the repository and compiles it.
- Step 3: Test: Automated tests are run to verify that the code changes have not introduced any regressions.
- Step 4: Package: The code changes are packaged into a deployable artifact.
- Step 5: Deploy to staging: The deployable artifact is deployed to a staging environment for further testing.
- Step 6: Test in staging: The code changes are tested in the staging environment to ensure that they work correctly in a production-like environment.
- Step 7: Approval: The changes are approved for release to production.
- Step 8: Deploy to production: The changes are deployed to production.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Implementing CI/CD

- Choose the right tools
- Define your workflow
- Automate everything
- Monitor and improve





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Tools & Technologies

- Version Control Systems (VCS): Git, SVN, Mercurial, etc. are commonly used VCS for source code management.
- Build Automation Tools: Maven, Gradle, Ant, etc. are popular build automation tools that help automate the build process.
- Continuous Integration Tools: Jenkins, Travis CI, CircleCI, GitLab CI/CD, etc. are commonly used CI tools that help to integrate code changes frequently and automatically.
- Configuration Management Tools: Ansible, Chef, Puppet, etc. are used for infrastructure automation and management.
- Containerization Technologies: Docker, Kubernetes, etc. are used for containerization and orchestration of applications.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Tools & Technologies (contd.)

- Cloud Computing Platforms: AWS, Azure, Google Cloud Platform, etc. provide cloud-based infrastructure for hosting applications.
- Monitoring and Alerting Tools: Prometheus, Nagios, Splunk, etc. are commonly used for monitoring and alerting.
- Test Automation Tools: Selenium, JUnit, TestNG, etc. are used for automated testing.
- Continuous Delivery/Deployment Tools: Spinnaker, AWS CodePipeline, Jenkins X, etc. are used for automated deployment.
- Infrastructure as Code (IaC) Tools: Terraform, CloudFormation, etc. are used to manage infrastructure as code.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Continuous Integration Tools

- Jenkins
- CircleCI
- TravisCI
- GitLab CI/CD
- AWS CodePipeline
- Github Actions





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

CI/CD Tools / Jenkins

- Jenkins is an open-source CI/CD tool that has been around for many years.
- It offers a wide range of plugins and integrations and is highly customizable.
- Jenkins is well-suited for complex build pipelines and large-scale deployments.



Jenkins

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Tools / Circle CI

- CircleCI is a cloud-based CI/CD tool that offers a simple configuration and easy setup.
- It supports multiple deployment environments and offers many integrations and plugins.
- CircleCI is well-suited for teams looking for a fast and easy-to-use CI/CD tool.







This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Tools / Travis CI

- Travis CI is a cloud-based CI/CD tool that offers a simple configuration and easy setup.
- It supports multiple deployment environments and offers many integrations and plugins.
- Travis CI is well-suited for teams looking for a fast and easy-to-use CI/CD tool.







This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Tools / Gitlab

- GitLab CI/CD is an integrated tool that is part of the GitLab platform.
- It offers a simple YAML-based configuration and supports multiple deployment environments.
- GitLab CI/CD is well-suited for teams using GitLab for version control and project management.







This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

CI/CD Tools / AWS CodePipeline

- AWS CodePipeline is a cloud-based CI/CD tool that is part of the Amazon Web Services (AWS) platform.
- It offers a simple visual interface for building pipelines and supports multiple deployment environments.
- AWS CodePipeline is well-suited for teams using AWS for their infrastructure and services.



AWS CodePipeline

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

CI/CD Tools / Github Actions

- GitHub Actions is a native CI/CD tool that is integrated with GitHub.
- It offers a simple YAML-based configuration and supports multiple deployment environments.
- GitHub Actions is well-suited for teams using GitHub for version control and project management.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Tools / Summary

Tool	Open-source	Cloud-based	Integration	Ease of use	Customization
Jenkins	Yes	No	Many	Complex	Highly customizable
GitLab CI/CD	Yes	No	GitLab	Simple	Moderate customization
GitHub Actions	No	No	GitHub	Simple	Limited customization
CircleCI	No	Yes	Many	Simple	Limited customization
Travis CI	No	Yes	Many	Simple	Limited customization
AWS CodePipeline	No	Yes	AWS	Simple	Limited customization





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Best Practices

- Automate Everything
  - Automate as much as possible in your build and deployment processes to reduce the risk of human error.
  - Use automation tools to automate builds, testing, and deployments.
- Keep Builds Fast
  - Keep your build times as short as possible to reduce feedback cycles and increase developer productivity.
  - Use techniques such as caching and parallelization to speed up builds.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Best Practices (contd.)

- Use Version Control
  - Use version control to track changes to your code and configuration files.
  - Use branching and pull requests to manage changes and enforce code reviews.
- Build Immutable Artifacts
  - Build immutable artifacts that can be deployed repeatedly and consistently across environments.
  - Use containerization tools like Docker to build immutable artifacts.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Best Practices (contd.)

- Test Early and Often
  - Test your code early and often in the development process to catch bugs and issues as early as possible.
  - Use automated testing tools to run tests quickly and frequently.
- Use Environment Variables
  - Use environment variables to store configuration data and secrets securely.
  - Avoid hardcoding secrets or configuration data in your code or configuration files.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Best Practices (contd.)

- Implement Continuous Feedback
  - Implement continuous feedback mechanisms to help identify and resolve issues quickly.
  - Use monitoring tools to collect and analyze data on application performance and usage.
- Ensure Consistency Across Environments
  - Ensure consistency across development, testing, and production environments to reduce the risk of issues in production.
  - Use configuration management tools to manage environment configurations and ensure consistency.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Best Practices (contd.)

- Use Continuous Delivery
  - Implement continuous delivery to ensure that your code is always ready for deployment.
  - Use automated testing and deployment tools to deploy code changes quickly and safely.
- Implement Security Best Practices
  - Implement security best practices throughout your CI/CD pipeline to ensure the security of your applications and data.
  - Use security scanning tools to identify vulnerabilities and risks.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Best Practices (contd.)

- Continuously Improve
  - Continuously review and improve your CI/CD pipeline to ensure that it is meeting your needs.
  - Use metrics and feedback to identify areas for improvement and make changes as needed.
- Use virtualization and simulation tools to simulate hardware environments and automate the process of testing software updates on real hardware.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Challenges

- Complexity
  - CI/CD pipelines can be complex and difficult to manage, especially for large or distributed systems.
  - Managing multiple tools and integrations can add to the complexity
- Tooling
  - Choosing the right CI/CD tools and integrating them with existing systems can be a challenge.
  - Maintaining and upgrading tools can also be a challenge.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Challenges (contd.)

- Testing
  - Testing can be time-consuming and difficult to automate, especially for complex applications.
  - Balancing the need for comprehensive testing with the need for fast feedback can be a challenge.
- Legacy Systems
  - Integrating CI/CD with legacy systems and applications can be difficult.
  - Legacy systems may not support modern CI/CD practices and may require manual intervention.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Challenges (contd.)

- Culture
  - Implementing CI/CD requires a cultural shift towards a more collaborative, agile, and DevOps-focused approach.
  - Changing the culture can be a challenge, especially in organizations with traditional silos between development and operations teams.
- Security
  - Security is a critical concern in CI/CD pipelines, and organizations must ensure that security is built into every step of the pipeline.
  - Implementing security measures can be challenging, especially in fast-paced environments.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Challenges (contd.)

- Scalability
  - Scaling CI/CD pipelines to support large or rapidly growing organizations can be a challenge.
  - Ensuring that pipelines can handle increasing volumes of code and deployments can be a challenge.
- Compliance
  - Organizations in regulated industries may face compliance challenges when implementing CI/CD.
  - Ensuring that pipelines comply with industry regulations can be a challenge.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Challenges (contd.)

- Hardware
  - Need to test the software on actual hardware increases the complexity of integrating hardware and software components.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies

- Netflix
- Amazon
- Etsy
- Nordstrom





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Netflix

Netflix is a well-known pioneer in the world of DevOps, and their approach to continuous integration and delivery has been a major factor in their success. Here are some key aspects of Netflix's CI/CD pipeline





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Netflix key features

- Microservices Architecture:
  - Netflix has a microservices-based architecture, where applications are broken down into smaller, independent services that can be developed, tested, and deployed independently
  - This approach enables faster release cycles and makes it easier to scale and maintain the system.
- Automated Testing:
  - Netflix has a comprehensive automated testing strategy that includes unit testing, integration testing, and end-to-end testing.
  - They use a combination of open-source and custom testing tools to ensure that the system is thoroughly tested before each release.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Netflix key features (contd.)

- Continuous Deployment:
  - Netflix has a fully automated continuous deployment pipeline that enables them to deploy code changes to production multiple times per day.
  - They use a combination of canary releases, blue-green deployments, and feature flags to safely roll out changes without affecting the entire system.
- Chaos Engineering:
  - Netflix is known for its approach to chaos engineering, which involves deliberately introducing failures into the system to identify and fix weaknesses.
  - They use a custom tool called Chaos Monkey to randomly kill instances of their services in production, which helps them identify and fix issues before they become major problems.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

CI/CD Case Studies / Netflix key features (contd.)

- Cloud-based Infrastructure:
  - Netflix runs its services on Amazon Web Services (AWS), which provides the scalability and flexibility needed to support their high traffic and availability requirements.
  - They use a combination of AWS services and open-source tools to manage their infrastructure and deployments.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Amazon

Amazon, one of the largest e-commerce companies in the world, uses CI/CD to deliver its services to customers.

With a complex infrastructure and hundreds of services, Amazon uses its own CI/CD tool called AWS CodePipeline to automate the process of building, testing, and deploying code changes.

This enables them to quickly respond to customer needs and market changes, resulting in improved business agility and customer satisfaction.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Amazon key features

- AWS CodePipeline:
  - Amazon's CI/CD pipeline is built using AWS CodePipeline, a fully managed continuous delivery service that enables the automation of code releases.
  - CodePipeline provides a streamlined workflow for building, testing, and deploying code changes, with integration to other AWS services such as AWS CodeBuild and AWS CodeDeploy.
- AWS CodeBuild:
  - Amazon uses AWS CodeBuild to automate the building and testing of code changes. CodeBuild supports a variety of programming languages and provides pre-built build environments, which can be customized as needed.
  - CodeBuild integrates with CodePipeline to automate the build and testing process as part of the overall pipeline.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

CI/CD Case Studies / Amazon key features (contd.)

- AWS CodeDeploy:
  - Amazon uses AWS CodeDeploy to automate the deployment of code changes to production. CodeDeploy provides a flexible deployment model that supports rolling updates, blue-green deployments, and canary releases.
  - CodeDeploy integrates with CodePipeline to automate the deployment process as part of the overall pipeline.
- CloudFormation:
  - Amazon uses AWS CloudFormation to automate the creation and management of AWS resources. CloudFormation provides a templating language that enables developers to define infrastructure as code, which can be version controlled and automated.
  - CloudFormation integrates with CodePipeline to enable the automatic creation and deletion of AWS resources as part of the overall pipeline.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

CI/CD Case Studies / Amazon key features (contd.)

- Serverless Computing:
  - Amazon has pioneered serverless computing with AWS Lambda, a service that enables developers to run code without managing servers.
  - Lambda supports a variety of programming languages and can be used to build highly scalable and cost-effective applications.
  - Lambda integrates with CodePipeline to enable the automatic deployment of serverless applications as part of the overall pipeline.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Etsy

Etsy is an online marketplace that uses CI/CD to enable continuous delivery of its services.

With a focus on experimentation and rapid prototyping, Etsy uses a tool called Deployinator to manage its CI/CD pipelines.

This enables them to quickly test and deploy code changes to production, resulting in improved business agility and customer satisfaction.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Etsy key features

- Feature Flags:
  - Etsy uses feature flags to enable continuous delivery of new features and improvements. Feature flags allow developers to selectively turn features on or off, which enables them to release new code to production with minimal risk.
  - Etsy has built a custom feature flagging system that integrates with their CI/CD pipeline to automate the process of turning features on or off as part of the overall release process.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Etsy key features (contd.)

- Continuous Integration:
  - Etsy uses Jenkins as their continuous integration server. Jenkins is used to build, test, and package code changes automatically.
  - Etsy's approach to continuous integration is characterized by a strong focus on test automation, with a suite of automated tests that run on every code change.
- Canary Releases:
  - Etsy uses canary releases to gradually roll out new features and improvements to production. Canary releases involve releasing new code changes to a small subset of users, monitoring the performance and stability of the new code, and gradually increasing the percentage of users who see the new code over time.
  - This enables Etsy to catch and fix issues quickly, before they affect a large number of users.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Etsy key features (contd.)

- ChatOps:
  - Etsy uses ChatOps to facilitate collaboration between developers, operations, and other teams.
  - ChatOps involves integrating chat tools such as Slack into the CI/CD pipeline, enabling developers to communicate and collaborate in real-time as part of the release process.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Nordstrom

Nordstrom, a popular department store chain, uses CI/CD to enable continuous delivery of its services.

With a focus on delivering a seamless customer experience, Nordstrom uses Jenkins to manage its CI/CD pipelines.

This enables them to quickly deploy code changes to production, resulting in improved business agility and customer satisfaction.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Nordstrom key features

- DevOps Culture:
  - Nordstrom has developed a strong DevOps culture that emphasizes collaboration between development and operations teams.
  - They have adopted agile methodologies and a "fail fast" mentality, which enables them to rapidly iterate on software development and deploy changes to production quickly.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Nordstrom key features (contd.)

- Continuous Integration:
  - Nordstrom uses Jenkins as their primary continuous integration server. They have built a custom pipeline that automates the build, test, and deployment process for their applications.
  - This pipeline includes a suite of automated tests that run on every code change to ensure that new code is production-ready.
- Microservices Architecture:
  - Nordstrom has adopted a microservices architecture for their applications, which enables them to break down large, monolithic applications into smaller, more manageable components.
  - This architecture makes it easier to develop, test, and deploy changes to production quickly and with minimal risk.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD Case Studies / Nordstrom key features (contd.)

- Canary Releases:
  - Nordstrom uses canary releases to gradually roll out new features and improvements to production.
  - They release new code changes to a small subset of users, monitor the performance and stability of the new code, and gradually increase the percentage of users who see the new code over time.
  - This enables Nordstrom to catch and fix issues quickly, before they affect a large number of users.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins

- Jenkins is an open-source automation server that is widely used for Continuous Integration and Continuous Delivery (CI/CD) of software projects.
- Jenkins Pipeline is a suite of plugins that enables the creation of pipelines for building, testing, and deploying software projects.
- Jenkins Pipeline provides a powerful and flexible way to define continuous delivery workflows as code, allowing for easy automation and management of the entire software development lifecycle.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Agents

Jenkins agents, also known as Jenkins slaves:

- worker nodes that perform build and test tasks on behalf of a Jenkins master node
- can be physical or virtual machines
- used to distribute workloads across a Jenkins cluster
- master node controls the agents and coordinates the build process.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Agents / Benefits

- **Scalability**: Agents can be added or removed from the Jenkins cluster as needed, making it easy to scale the build environment.
- **Parallelism**: By using multiple agents, builds can be executed in parallel, reducing build times.
- **Flexibility**: Agents can be used to run builds on different operating systems, hardware, and environments, allowing for greater flexibility in the build process.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline

Jenkins Pipeline is a tool for creating, managing, and visualizing entire Continuous Delivery pipelines.

- It allows users to define their pipeline as a code using the Groovy scripting language, which provides a powerful and flexible way to define complex workflows.
- It can be used to automate a wide range of tasks, including building, testing, and deploying software applications.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Key Features

- **Pipeline as code**: Jenkins Pipeline allows users to define their pipeline as code using the Groovy scripting language. This makes it easy to version control and manage the pipeline and enables the pipeline to be easily shared among team members.
- **Multiple stages**: Jenkins Pipeline enables the creation of multiple stages for building, testing, and deploying software projects. This allows for a more fine-grained approach to the software development lifecycle.
- **Parallel execution**: Jenkins Pipeline allows for stages to be executed in parallel, enabling faster build times and more efficient use of resources.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Key Features (contd.)

- **Plugin integration**: Jenkins Pipeline integrates with a wide range of plugins, including Git, Maven, and Docker, allowing for easy integration of existing tools and workflows.
- **Pipeline visualization**: Jenkins Pipeline provides a graphical view of the pipeline and the status of each stage, making it easy to monitor the progress of the build and identify any issues that need to be addressed.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Key Features (contd.)

- **Error handling**: Jenkins Pipeline provides robust error handling capabilities, allowing for easy recovery from failed builds and enabling the pipeline to continue running in the event of an error.
- **Environment management**: Jenkins Pipeline enables the management of environments, allowing for easy provisioning and deployment of resources required by the pipeline, such as containers and virtual machines.
- **Customizability**: Jenkins Pipeline is highly customizable, enabling users to define their own stages, steps, and scripts, and to incorporate custom plugins and integrations.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Advantages

- **Automation**: Jenkins Pipeline enables the automation of the entire software development lifecycle, from building and testing to deployment and monitoring.
- **Flexibility**: Jenkins Pipeline provides a flexible and powerful way to define complex workflows, enabling users to customize their pipelines to meet their specific needs.
- **Collaboration**: Jenkins Pipeline enables teams to collaborate more effectively, by providing a common way to define and manage the pipeline as code.
- **Visibility**: Jenkins Pipeline provides a graphical view of the pipeline, making it easy to monitor the progress of the build and identify any issues that need to be addressed.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example

Prerequisites:

- Jenkins installation,
- Installation of plugins: Pipeline,
- Pipeline: Stage View,
- Pipeline: GitHub Groovy Libraries,
- Pipeline: Build Step





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example (contd.)

1. Create a new pipeline
   To create a new pipeline, navigate to the Jenkins homepage and click on the "New Item" button. Enter a name for your pipeline and select the "Pipeline" type. Click "OK" to create the new pipeline.
2. Define the pipeline
   In the pipeline configuration page, you will see a text editor where you can define the stages of your pipeline. A simple pipeline may have three stages: build, test, and deploy.
3. Add stages
   To add stages to your pipeline, use the syntax "stage('stage-name') { // stage content }". For example, to add a build stage, use the following syntax: "stage('Build') { // build content }". Similarly, you can add test and deploy stages.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example (contd.)

4. Add steps
   Within each stage, you can add steps to define the tasks that need to be performed. For example, to build your application, you can add a "sh" step that runs a shell command to compile your code.
5. Add triggers
   You can add triggers to your pipeline to specify when the pipeline should run. For example, you can add a trigger to run the pipeline whenever a new code change is pushed to your repository.
6. Save and run
   Once you have defined your pipeline, save the configuration and run the pipeline. You can view the progress of your pipeline in the Jenkins dashboard.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example / **Structure**

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                // Build steps go here
            }
        }
        stage('Test') {
            steps {
                // Test steps go here
            }
        }
        stage('Deploy') {
            steps {
                // Deploy steps go here
            }
        }
    }
}
```





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example / **Init**

```
pipeline {
    agent any
    environment {
        BUILD_NUMBER = "${env.BUILD_NUMBER}"
        VERSION_NUMBER = "1.0.${env.BUILD_NUMBER}"
    }
    stages {
        stage('Checkout') {
            steps {
                checkout([$class: 'GitSCM', branches: [[name: '*/main']],
userRemoteConfigs: [[url: 'https://github.com/myorg/myrepo.git']]])
            }
        }
```





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example / **Build**

```
stage('Build') {
      steps {
            sh './gradlew clean build'
      }
      post {
            success {
                  archiveArtifacts artifacts:'build/libs/*.jar'
            }
      }
}
```





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example / **Test**

```
stage('Test') {
     steps {
          sh './gradlew test'
     }
     post {
          always {
               junit 'build/test-results/**/*.xml'
          }
     }
}
```





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example / **Deploy**

```
stage('Deploy') {
    when {
        branch 'release/*'
    }
    steps {
        sh './gradlew publish'
    }
}
```





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins / Pipeline / Example / **Post result**

```
post {
    always {
        emailext body: "Build URL:
${env.BUILD_URL}", subject: "Build Status:
${currentBuild.result}", to: 'team@example.com'
    }
}
```





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins & Embedded Devices / Challenges

- Limited resources (memory, processing power, storage)
- Variety of hardware platforms and architectures
- Lack of standard interfaces for testing
- Real-time constraints





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins & Embedded Devices / Integration

- Using Jenkins as a build server
- Installing Jenkins plugins for embedded device testing
- Configuring Jenkins to run tests on embedded devices
- Example: Jenkins integration with Raspberry Pi





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins & Embedded Devices / Plugins

- Device plugins (e.g., Android Emulator Plugin, Arduino CI Plugin)
- Testing plugins (e.g., CppUnit Plugin, Google Test Plugin)
- Build and deployment plugins (e.g., Gradle Plugin, Docker Pipeline Plugin)





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins & Embedded Devices / Tests Execution

- Cross-compilation and remote execution
- Using emulators and simulators
- Physical testing on embedded devices
- Example: Jenkins pipeline for testing on Raspberry Pi





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins & Embedded Devices / Best Practices

- Use lightweight test frameworks and tools
- Minimize the use of hardware resources
- Automate the deployment and configuration of devices
- Use a version control system to track changes to hardware and software configurations
- Regularly review and optimize the test suite





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Jenkins & Embedded Devices / Pipeline

- Clone repository stage
  - Checkout code from version control system
  - Use specific branch or tag for the embedded device
- Build stage
  - Cross-compile the code for the embedded device
  - Generate binary file for the device
- Deploy stage
  - Copy binary file to the device
  - Configure the device with required libraries and dependencies
- Test stage
  - Run automated tests on the device
  - Capture test results and generate reports
- Report stage
  - Publish test results and reports
  - Send notifications to team members





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





Future of CI/CD

1. **Increased use of cloud-native architectures:** As more organizations adopt cloud-native architectures, there will be a greater need for CI/CD tools that integrate with these architectures. This will lead to increased use of cloud-based CI/CD solutions that can scale to meet the needs of large, complex applications.
2. **More automation:** There will be a greater focus on automating as much of the CI/CD process as possible. This will include automated testing, automated deployment, and automated rollback in the event of a failed deployment.
3. **Greater use of machine learning and artificial intelligence:** Machine learning and artificial intelligence will be used to optimize the CI/CD process, identify bottlenecks, and improve overall efficiency.





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

Future of CI/CD (contd.)

4. **Increased focus on security:** With the rise of cyber threats and data breaches, there will be a greater focus on security in the CI/CD process. This will include the use of tools and techniques to identify and mitigate security vulnerabilities.
5. **More emphasis on collaboration:** As CI/CD becomes more integrated with other parts of the software development process, there will be a greater emphasis on collaboration between developers, testers, and operations teams.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD in XANDAR: Objectives

To ensure working prototypes of the XANDAR Project tools and software systems in all phases of the project

- Setup of the CI/CD infrastructure that serves as the central platform for the continuous integration and testing automation of XANDAR software artifacts
- Use of CI/CD pipeline during the project to integrate all the involved components
- Run automated unit and integration tests for the various application profiles
- Integrate components from all partners and contribute component-specific test-cases to support the continuous development process





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.





CI/CD in XANDAR: Requirements

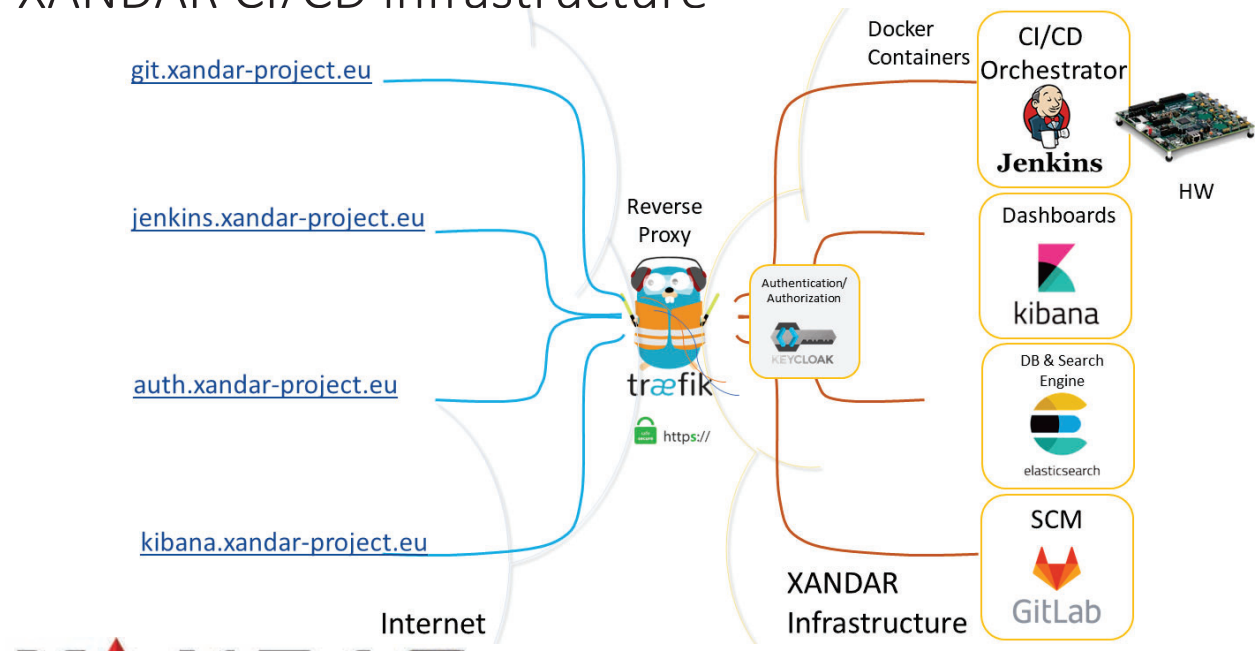
- Support a Distributed revision control
- Central Source Code Management
- CI/CD service should be able to execute automations
- Authenticated/Authorized/Accounting (AAA) access to CI/CD infrastructure
- Solution must be hosted within the consortium premises and not by 3rd party online services
- Deliver a container-based solution





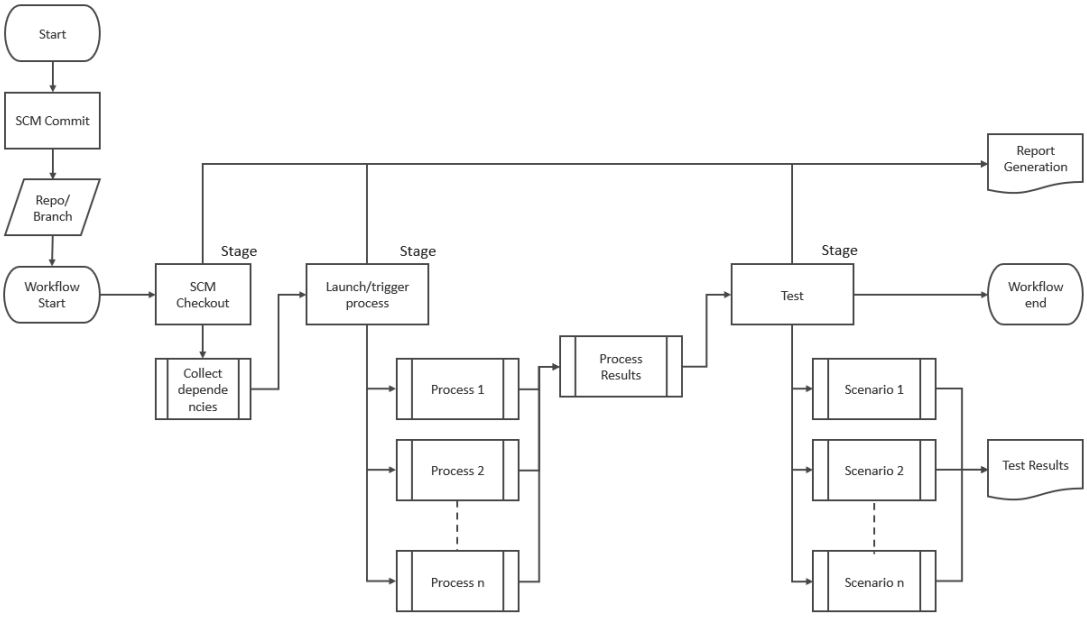
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

XANDAR CI/CD infrastructure



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

XANDAR CI/CD process



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

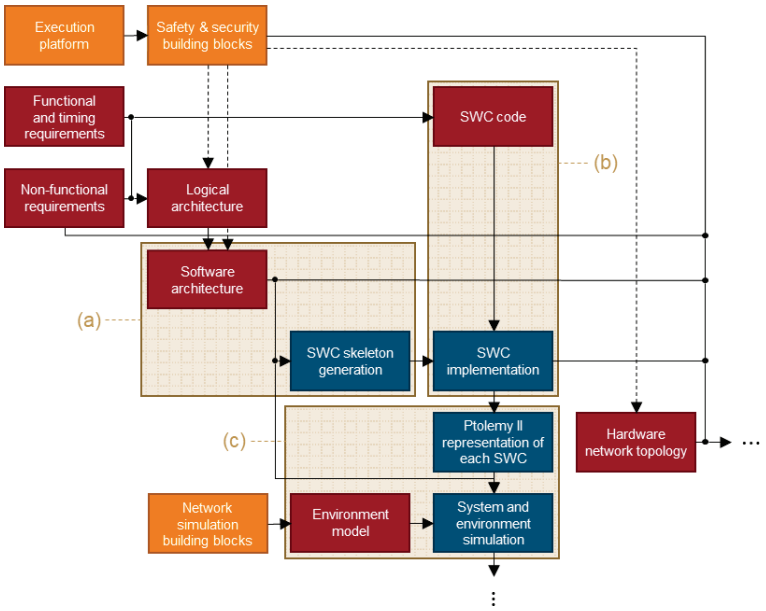
Software architecture processing & functional simulation

3-discrete pipeline stages:

a) Generation of SWC skeletons given the SW architecture model

b) Build of target independent SWCs implementations

c) Functional simulation of the network of SWCs

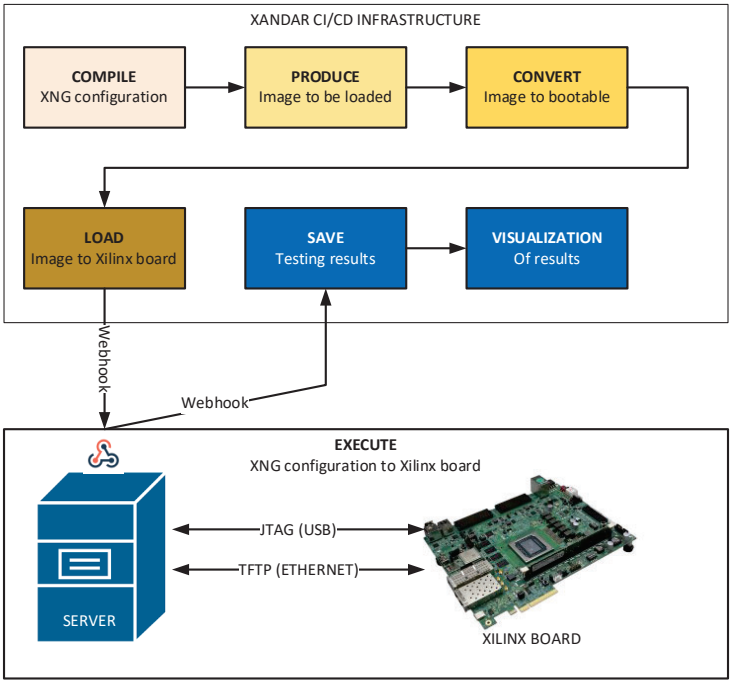


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

XNG based SWCs

CI/CD pipeline stages:

- Compilation of XNG configuration to produce final image
- Convert image to Xilinx bootable format
- Load the bootable image onto a Xilinx Zynq dev board (JTAG/TFTP)
- Execute XNG configuration on a Xilinx dev board



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

Network Components Configuration for short/long range wireless communications

Model-based network configuration based on application requirements

1. Inject Jenkins-based CI/CD strategy within NS3:

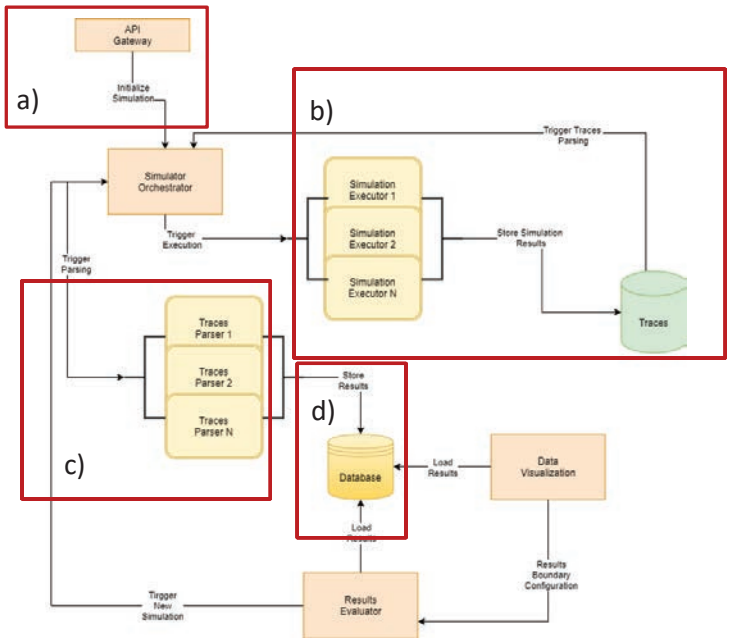
a) Webhook triggered by the orchestrator initializes simulations (dependencies collection & network/simulation configuration loading)

b) Webhook triggered by the orchestrator starts the simulation

c) Webhook triggered by the orchestrator parses simulation results

d) Simulation results are stored in database

2. Proceed with Continuous Deployment to network nodes



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.

Hands on Session

1. Live demonstration with practical examples

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957210.



Dimitris KARADIMAS – dimitris@avntechgroup.com
Innovative Technology Solutions (AVN), Cyprus





4th Summer School on Cyber Physical Systems and Internet of Things

Morayo Adedjouma
Luis Palacios
Réda Nouacer
DRT.LIST.DILS.LSEA
CEA – NanoInnov , Palaiseau - France

FROM KNOWLEDGE-BASED TO MODEL-BASED ENGINEERING OF EMBEDDED INTELLIGENCE AND TRUST IN CPS

Thursday 8 June -2023

Overview

- **Complex Systems Engineering** require specific design and engineering techniques to ensure the resulting system *complies* with its *requirements*.
- Due to the multi-disciplinary nature of this process, heterogeneous stakeholders need to interact, each one with specific concerns & viewpoints (safety, functional, electrical, mechanical, logical, legal, etc.), as well as specific levels of abstraction of the system being modelled.
- The design decisions propagate to other stakeholders and affect their results.
- To harmonize this interaction, we propose to integrate **Ontologies** (Knowledge Base) into system design tools. (i.e. Papyrus)
- Enable to evaluate **external constraints** (generated outside of the tooling environment) against the current design.
- This enables early detection of errors and misalignments, that would propagate to the implementation and deploy phases, saving time, energy and money.
- This integration enables the **reuse of (expert) knowledge**.

Knowledge Representation & Reasoning

Knowledge Representation & Reasoning

- Knowledge Representation & Reasoning (**KRR**) as a field of Artificial Intelligence (**AI**), is concerned with how humans acquire, store, process, learn and use knowledge, to provide machines with these abilities.
- A particular case are Description Logics (DL) ontologies, provided with formal semantics, a solid mathematical background and tractable computational properties.
- The **Web Ontology Language (OWL)**[1] is the current W3C recommendation syntax for ontologies.

1 https://www.w3.org/OWL/

Model-Based Systems Engineering - Papyrus

Model-Based System Engineering **(MBSE)** provides good practices and formalized syntax that make the engineering process systematic.

It notably helps in sharing the same interpretation of the models among experts.

Among the existing modeling languages UML[1] and SysML[2].

Papyrus[1] is a Modelling Environment tool, supporting these languages.

From the resources provided by UML we target:

- ✓ Class Diagrams : Description of the entities that exist in our design, as well as their relations.
- ✓ Composite Structure diagrams: description of composite structures, composed of parts, and the connections between them.
- ✓ Instance specifications: concrete implementations of classes, relations, connections, etc.

https://www.eclipse.org/papyrus/index.php

KRR integration into MBSE

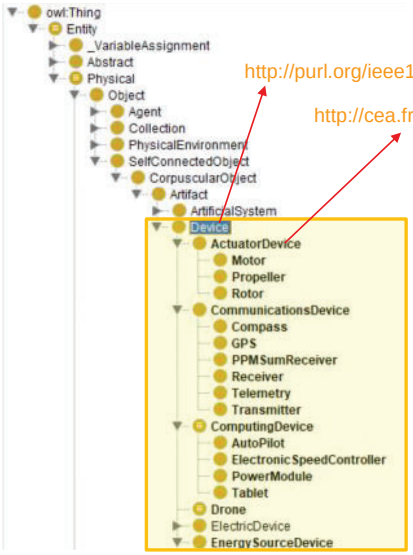
To bridge the gap between **MBSE** tooling and **KRR**, we have identified the following challenges:

- ✓ The **terminology** in the ontology should be automatically made available to the designer, saving time and avoiding mistakes.
- ✓ The system's model designed in the MBSE tooling environment should be described in terms of this existing knowledge (ontology). In our context this means the annotation of the UML model with the ontology concepts.
- ✓ The system's design should be exported from the MBSE environment as an OWL compliant representation, thus providing a tool-agnostic formal representation of the system.
- ✓ The enhanced models obtained this way need to be suitable for automatic reasoning tasks, like consistency and instance checking.

ODrone

Bases : CORA, CORAX, RPARTS, Dronetology, C4D

- ✓ We have developed an UAVs ontology (**ODrone**), and integrated it into IEEE1872-CORA[1], to provide a standardized formal domain specific vocabulary for UAVs.
- ✓ This ontology targets the physical components viewpoint.

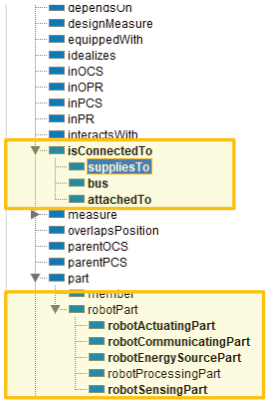


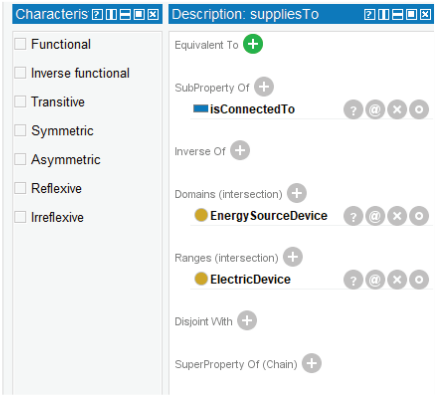
http://purl.org/ieee1872-owl/sumo-cora#Device

http://cea.fr/vocabulary/drone-physical#ActuatorDevice

CORA establishes the taxonomy and main concepts for autonomous systems, but it does not specify the devices that compose a system.

We extend this taxonomy targeting the UAVs domain.





RPARTS (included in CORA) provides the roles of: *part, robotPart,* and their specializations for certain types of devices, e.g.: *robotActuatingPart, robotSensingPart*, etc.
We also extend it with relations to connect these parts.

1 https://github.com/srfiorini/IEEE1872-owl

Operational Design Domain And Hazard Analysis

"SAFETY", "FUNCTIONAL SAFETY" AND "SAFETY OF THE INTENDED FUNCTIONALITY"– KEY FACTORS IN AUTOMATED SYSTEM INNOVATION

- **Key requirements for technical & societal acceptance**
  - **Safety of robotics applications must be guaranteed**
  - **Legal directives and standards compliance must be fulfilled!**
  - **Avoid emergency stops and ensure system stability**



Safety is the condition of being protected from harm or other non-desirable outcomes. It can also refer to risk management.

Functional safety is the part of the overall safety of a system or piece of equipment that depends on automatic protection operating correctly in response to its inputs or failure in a predictable manner.

Safety of the Intended Functionality (SOTIF) concerns with guaranteeing the safety of a functionality that can have safety risks in the absence of a fault.

<< Guidance on measures to ensure the absence of unreasonable risk due to a hazard caused by failures & insufficiencies of functionalities where proper situational awareness is essential to safety >>

"SAFETY", "FUNCTIONAL SAFETY" AND "SAFETY OF THE INTENDED FUNCTIONALITY"– KEY FACTORS IN AUTOMATED SYSTEM INNOVATION

Challenges

1. Complex/changing operational contexts; ambiguous scenarios;

Context: In practice, the number of possible scenarios which have to be managed by an automated tends to be infinite. Because the NNs learned from data, it is impossible to ensure that these data capture the infinite number of scenarios in which automated systems must operate, which makes their safety evaluation challenging.

Goal: We need a mean to define the scenario-space in which the automated system must operate safely without having to enumerate the different scenarios individually. The scenario-space is specified through the **operational design domain.**

2. Data noise; degraded sensor quality and sensor failures, processing algorithms error.

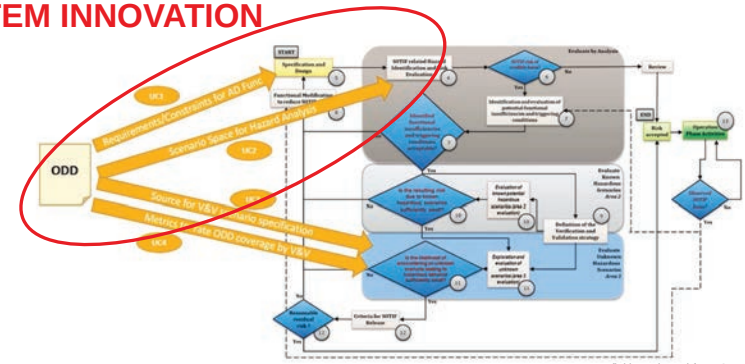
Context: The integration of AI component in automated systems brought new types of risk in such systems, e.g. limited performance due to sensors capabilities, hazardous behavior due to functional insufficiencies, uncertainty in the verdict of the AI due to the non-interpretability of their decision making process, etc.

Goal: We must provide a risk analysis that make the system be resilient to unsafe events coming from its environment and from its internal faults by setting up threshold for properties of interest that ensure safety without compromising system performance. The risk analysis must be defined based on the **operational design domain.**

OPERATIONAL DESIGN DOMAIN & SAFETY ANALYSIS – KEY FACTORS IN AUTOMATED SYSTEM INNOVATION

ODD will play a key role in future automated systems (AS)

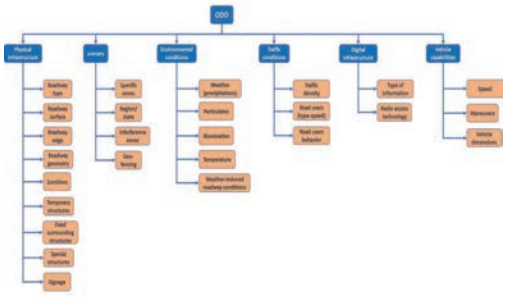
Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, **environmental**, **geographical**, and **time-of-day** restrictions, and/or the requisite presence or absence of certain **traffic** or **roadway characteristic**.



*Definition adapted from SAE J3016

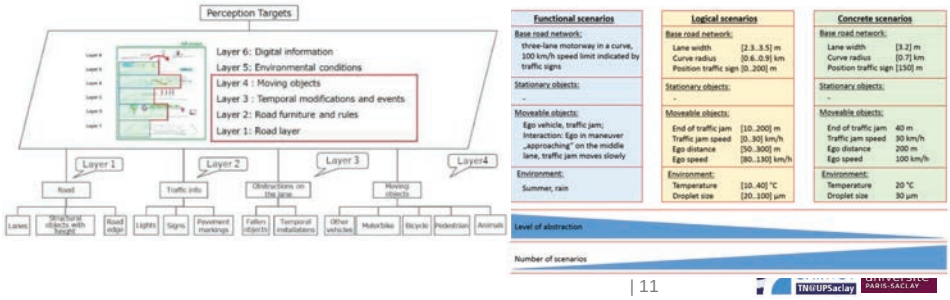
ODD Taxonomy can be an effective way to define and implement the ODD

It will be **required** to implement **safety requirements** in the **system's design**

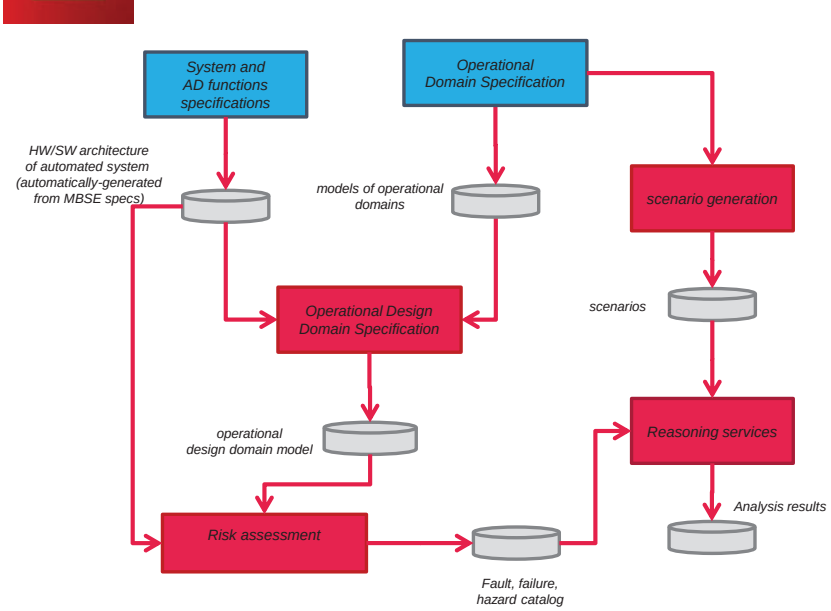


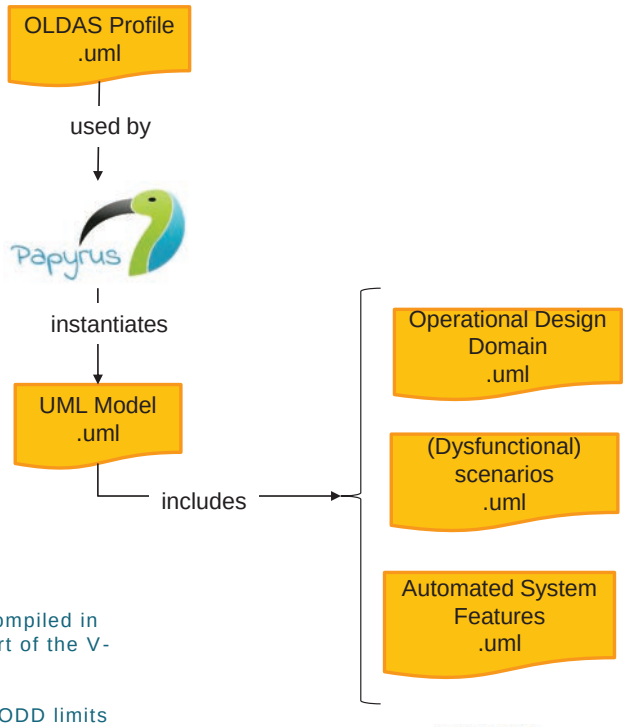
Structuring of scenarios can be achieved from ODD attributes

Description from **functional** to **concrete** **scenario**



APPROACH

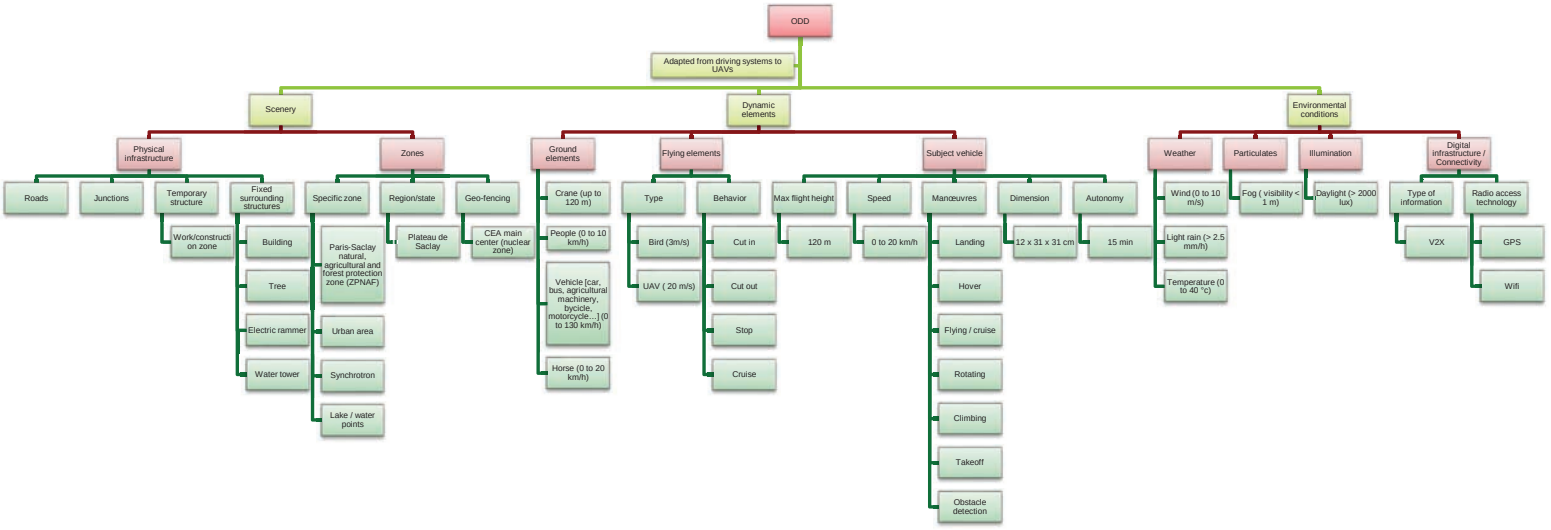




Design Time: Ensure that Automated System can drive in the operational conditions compiled in ODD. It is defined during the design process and verified and tested during the right part of the V-cycle.

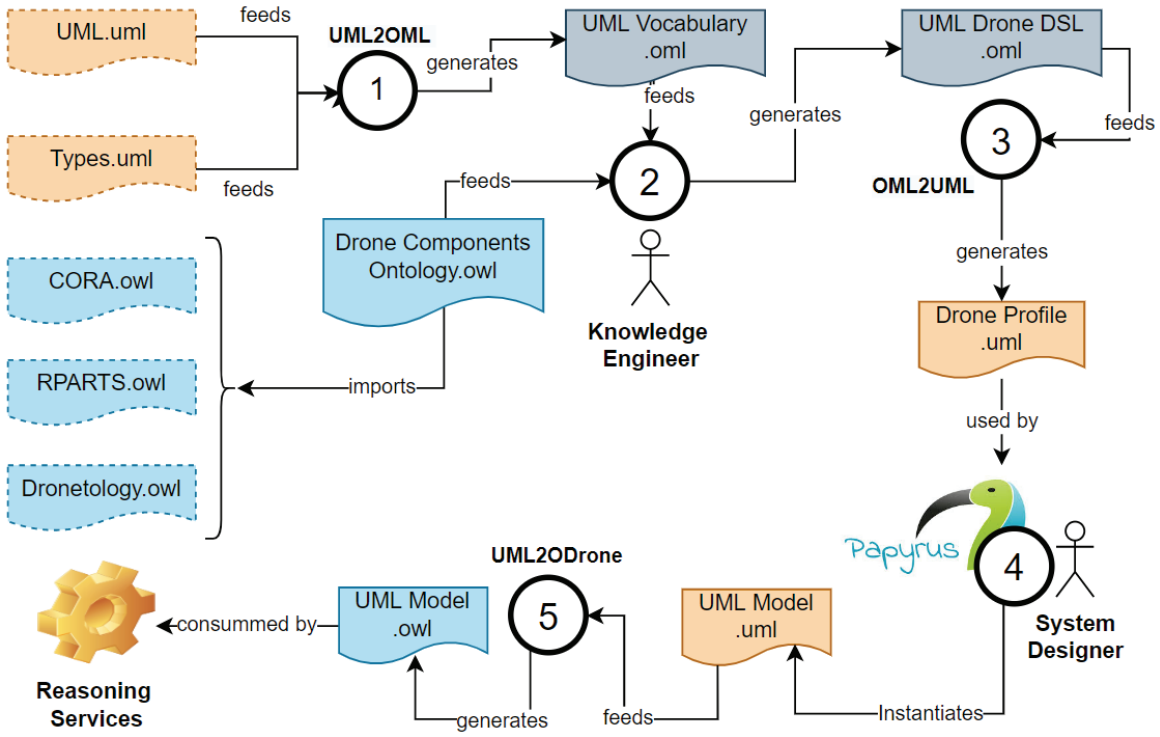
Operation Time: Ensure that the system remains in an environment where the defined ODD limits are valid. It concern the monitoring of functional boundaries during the operational phase.

OLDAS based ontology

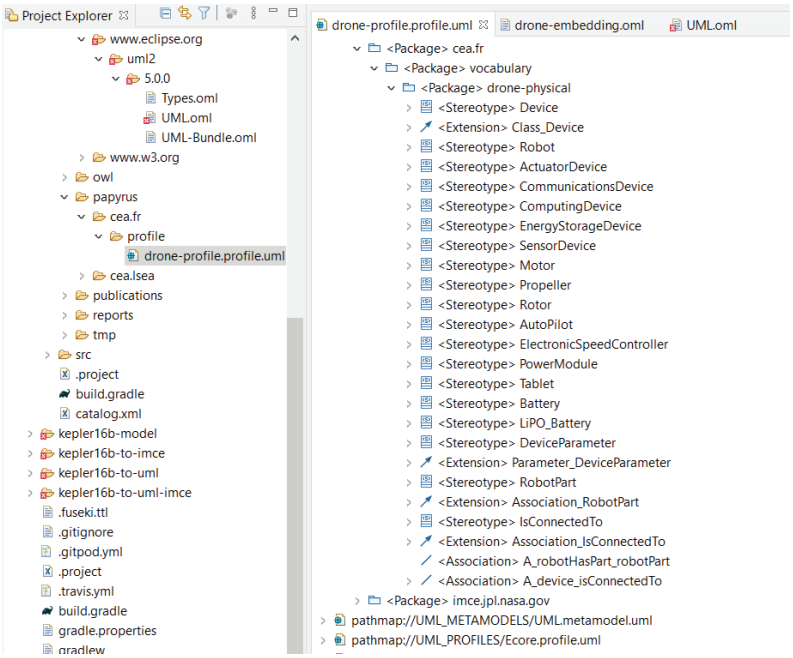


Knowledge Integration into MBSE Approach

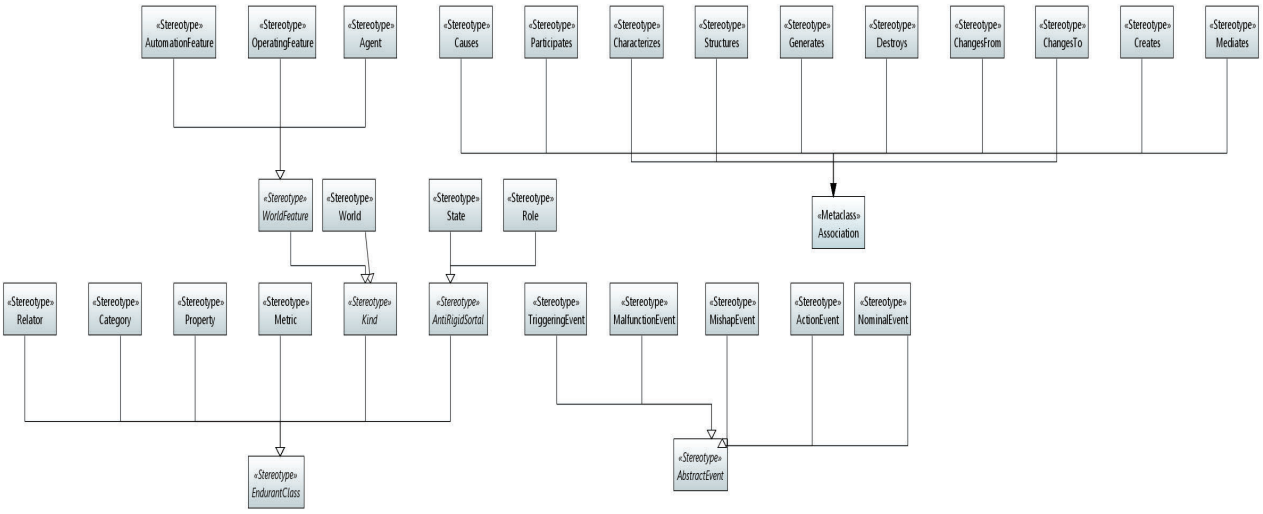
Overview - Workflow



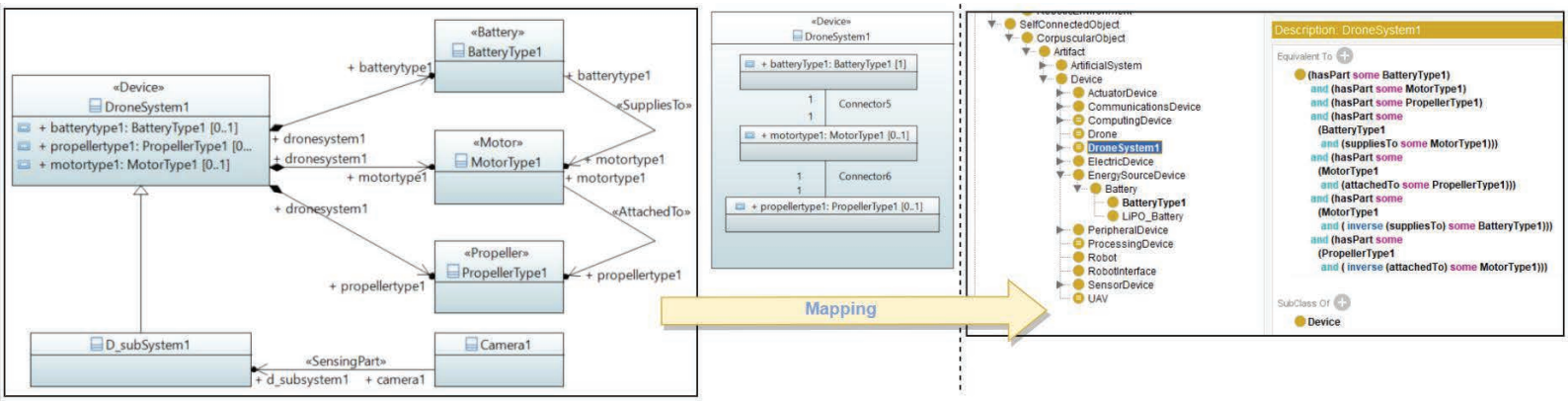
Step 3: Drone UML Profile



OLDAS UML Profile



UML Model to OWL Ontology (ODrone)

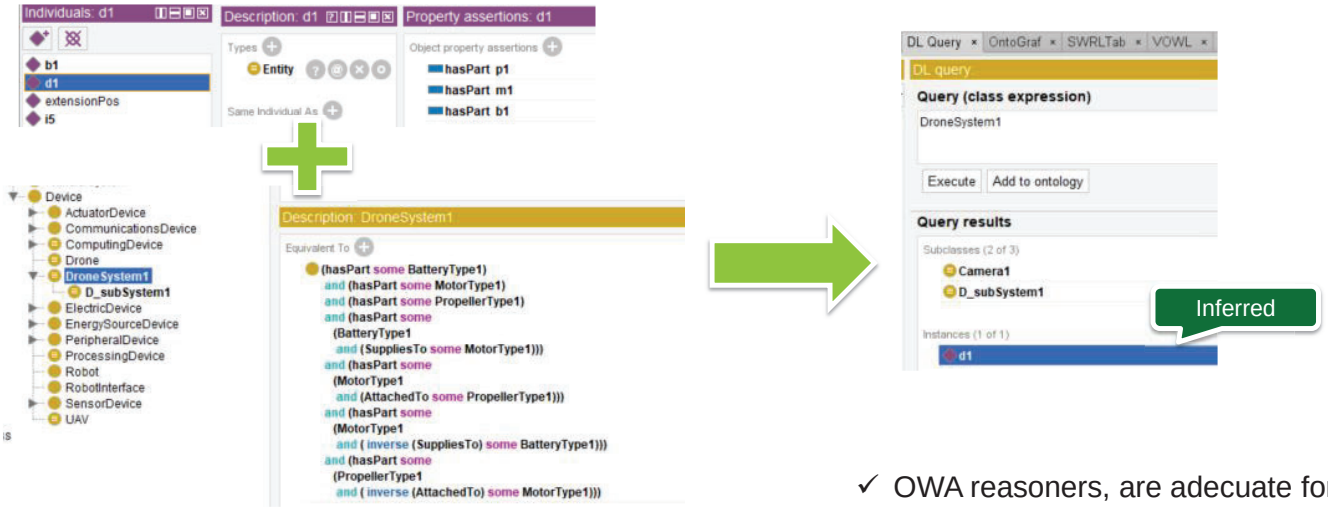


Reasoning

Example1 : Instance Checking

In this example we define an instance of an anonymous device : **d1**.

Because **d1** it has (at least) all necessary and sufficient *properties* imposed by the class definition (DroneSystem1), we can infer **d1** is indeed a **DroneSystem1** type of system…



✓ OWA reasoners, are adecuate for this task

Reasoning

Example 2 : Conformity Check

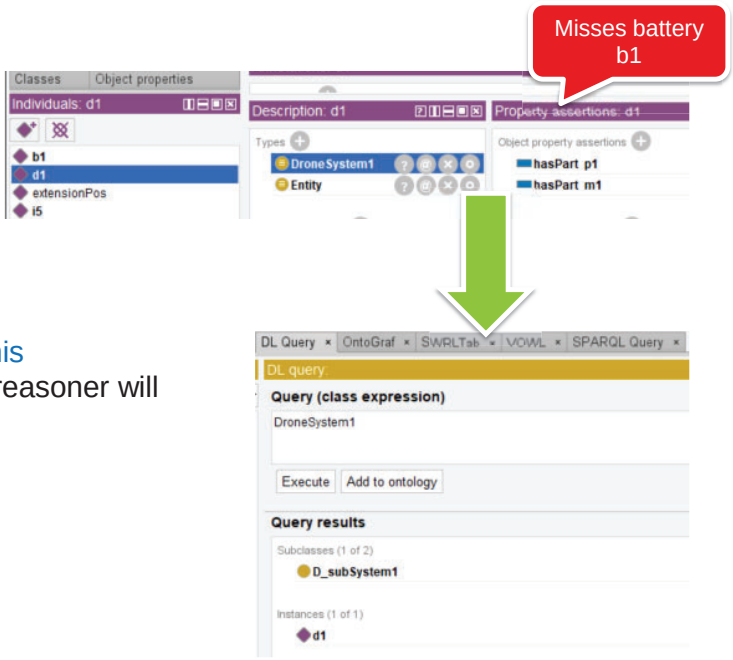
A designer assumes her design/implementation is correct.

We would like to verify if it conforms to a specification.

Assume **s1** of example 1, if we explicitly add the Type **DroneSystem1** (we want to verify it is1 conforms to this definition), and remove the link to the battery b1. The reasoner will infer **s1** is of type **DroneSystem1.**

OWA reasoning is not suitable for this task

We resource to **SPARQL**



Reasoning

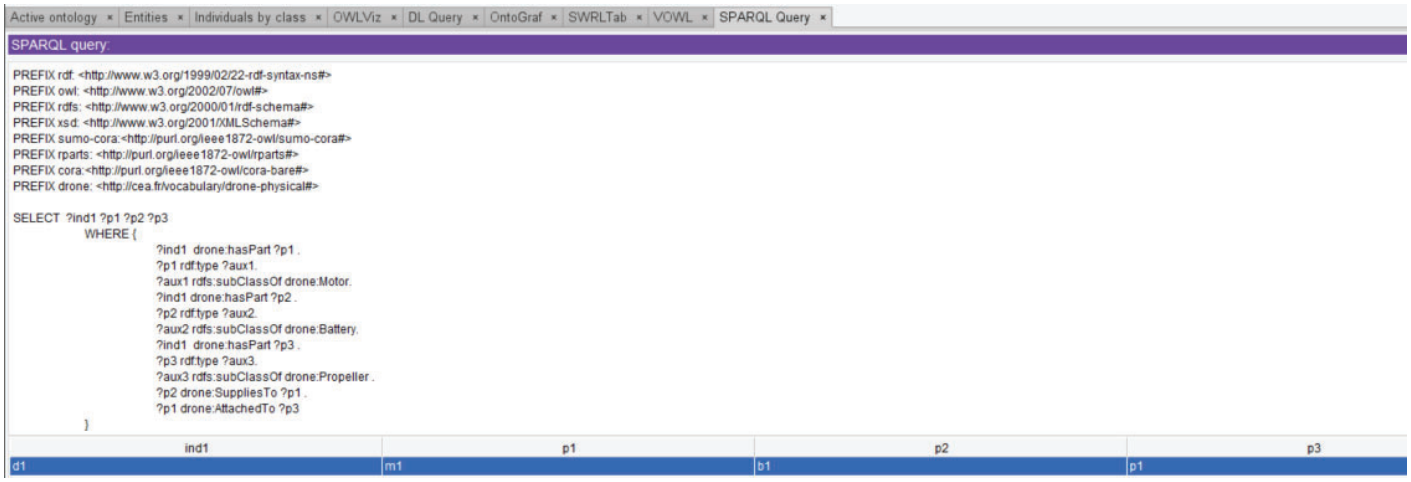
Example 2 : Conformity Check (continued…)

The following **SPARQL**, corresponds to **DroneSystem1** definition.

In contrast with OWA reasoning, in this case, every necessary property of a DroneSystem1 must be **explicitly** stated.

The instance d1 in Example 2 (previous slide) , wont be considered a DroneSystem1, unless hasPart(**b1**) property is added.

Using grounded graphs (+ inferences) and SPARQL (+SWRL) enable **conformity check**, and knowledge reuse under CWA.



Examples



Cranes:
80m
Flight Regulation :
120 m max.



http://www.paris-saclay.com/fonctionnalites/photosvideos-128/le-campus-urbain-vu-par-drone-1853.html?cHash=490d6658504c4e2c57940e735c2ad4e0



Idea 1: drones at max speeed.

Route 1: unfesaible, because too much time/

Route 2: feasible/

Increase Drone Max Speed + Battery =>

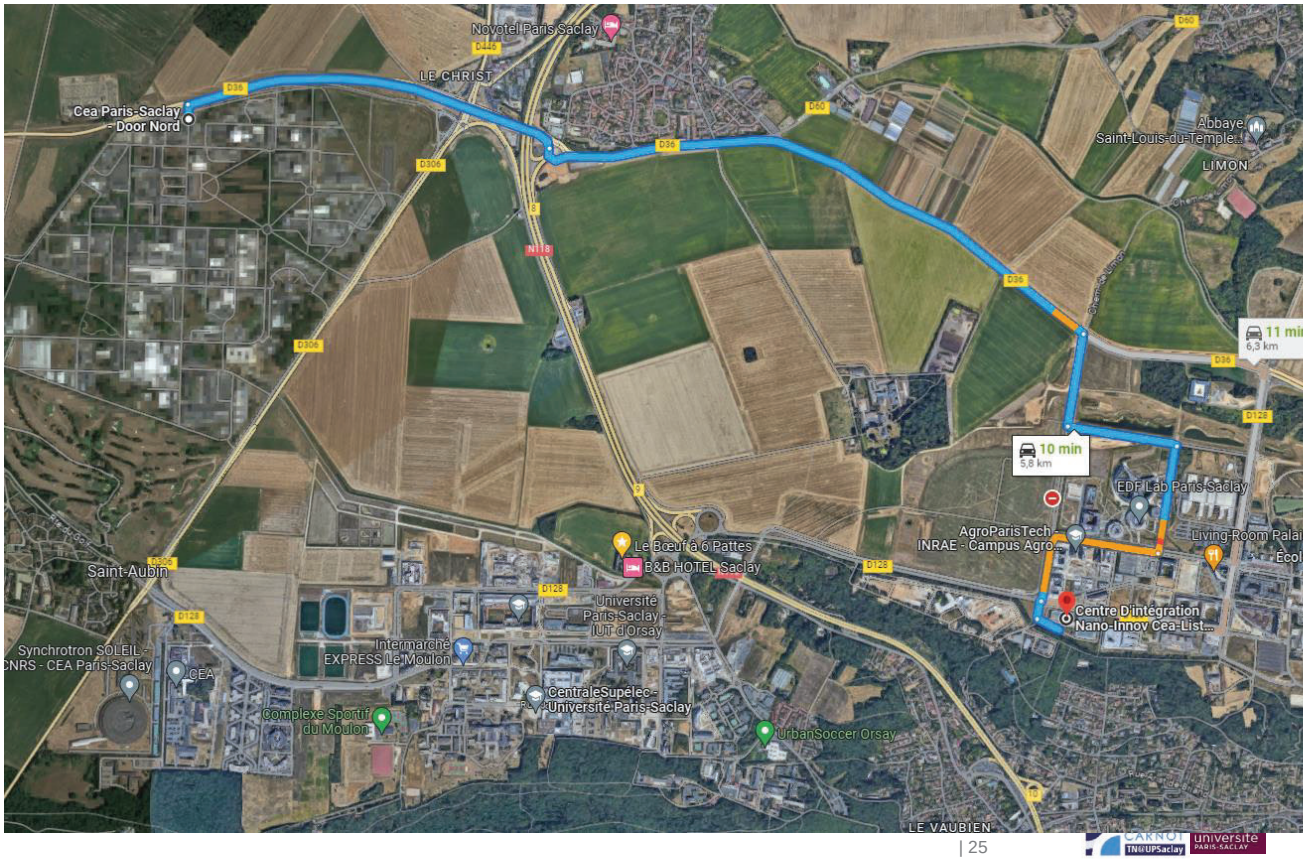
Reroute 1 feasible





Idea 2: drones and environement

a) Adda forbidden zone > route X is not valid anymore
b) Weahter= conditions above a treshold (à Saclay there is FOG in winter), therefore Obstacle Detection via camera not safe enough. 2: solutions: either OD via lidar/sonar , or change route.



Tutorial

Current / Further Work

- **Extend viewpoints** : Functional + C4D Reference Architecture.
- Integration of **Hazard Analysis** for safety critical systems.
- **Reasoning** tasks:
  - Consistency and Instance Checking,
  - SPARQL : patterns / graphs, anti-patterns,
  - Rules : SWRL, SPIN, SHACL.
- **Knowledge Acquisition & Reuse**.
- **Models Generation** : Provide candidate models that satisfy a specification.
- **Analysis**: Semantic Clustering, Knowledge Discovery, Link prediction (Knowledge Graphs).
- **Publish** ontology and demo.
- Direct **integration** in Papyrus.

Thanks

luis.palacios@cea.fr

morayo.adedjouma@cea.fr

reda.nouacer@cea.fr

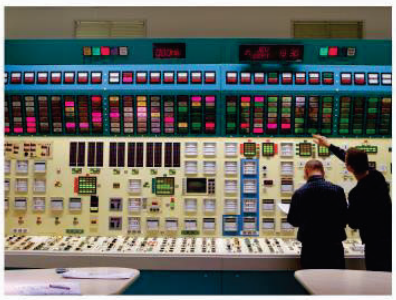
Towards Characterizing AI Trustworthiness

A three-players game

Zakaria.Chihani@cea.fr

Life cycle of a **safety-critical** software

A critical system is a system whose failure may cause physical harm, economical losses or damage the environment







Life cycle of a **safety-critical** software



Specify Input? Output? (*Speed? Memory?)* Privacy? **Safety? Security?**

Semi-informal text, scope of statement, properties to verify. Auto extraction etc.

Design Algorithm? Components? Pieces? Options?

Model-based tools available (e.g. Papyrus), autogen of code possible

Implement Language? Methods? Types? Data structures? Modules? Interfaces?

Decompose, annotate, comment, autogen from formal spec

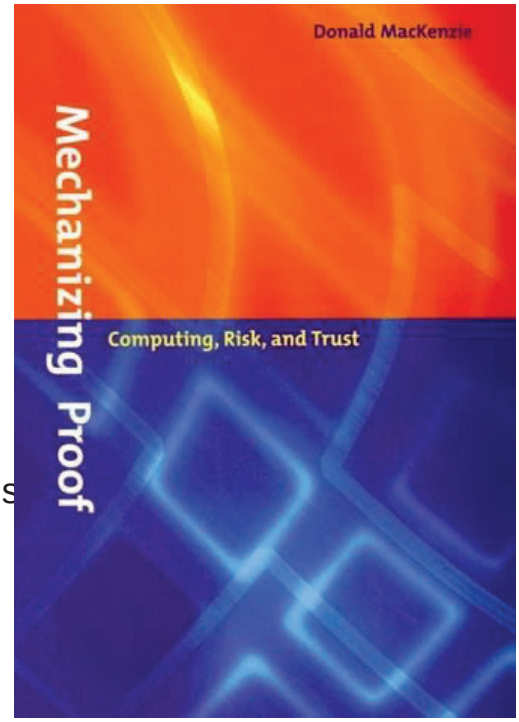
Validate Tests? **Formal Methods?** Tools? Regulations? Standards?

Test coverage, source→binary validation,...

Maintain **Graceful degradation? Monitoring?** Debugging & patching?

Maximal acceptable failure, overhead of monitoring, patching, significant error msg

History





Tests? **Formal Methods?** Tools? Regulations

Test coverage, source→binary validation,...

History

In 1979:

"[P]rogram verification is bound to fail. We can't see how it's going to be able to affect anyone's confidence about programs"

"Social processes and proofs of theorems and programs", Communications of ACM.
By Richard De Millo, Richard Lipton, and Alan Perlis.

History

In 1979:

"[P]rogram verification is bound to fail. We can't see how it's going to be able to affect anyone's confidence about programs"

"Social processes and proofs of theorems and programs", Communications of ACM.
By Richard De Millo, Richard Lipton, and Alan Perlis.

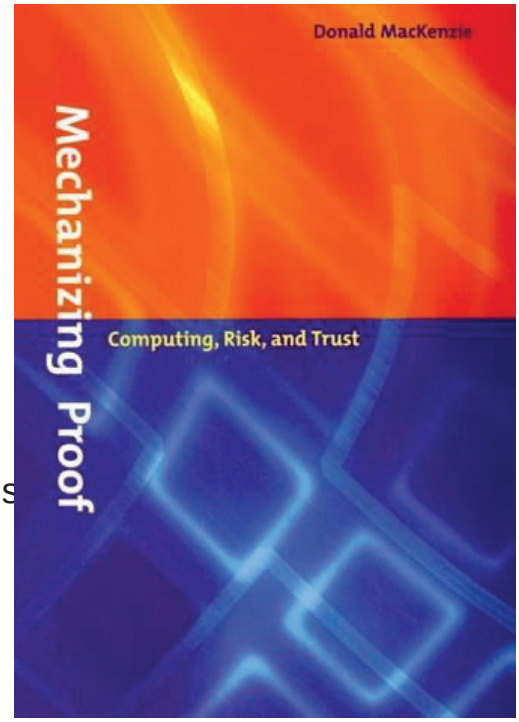
- Distinguished Professor of Computing at the Georgia Tech
- VP and CTO of Hewlett-Packard

- Yale, Berkeley, Princeton, Georgia Tech
- Knuth Prize winner

- ACM, Carnegie Mellon, Yale, Purdue
- The first recipient of the Turing Award

History

We have come a VERY long way...





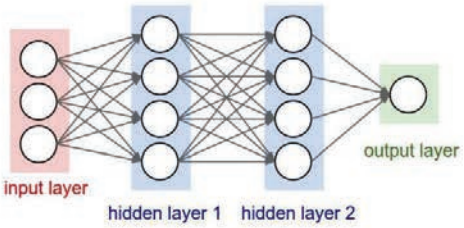
Tests? **Formal Methods?** Tools? Regulations

Test coverage, source→binary validation,...

What happens if you add AI ?



Human written code



Machine learned code









What happens if you add AI ?



Specify

How to specify unstructured data?

Input? Output? (*Speed? Memory?)* Privacy? Safety? Security?

New problems (e.g., bias, GIGO)

Semi-informal text, scope of statement, properties to verify. Auto extraction etc.

Design

Algorithm? Components? Pieces? Options?

Dataset selection and analysis, rule conceptions for rule-based systems, etc.

Model-based tools available (e.g. Papyrus), autogen of code possible

Implement

Language? Methods? Types? Data structures? Modules? Interfaces?

Decompose, annotate, comment, autogen from formal spec

Kernel functions for SVM, activation functions for NN, learning algorithm, hyperparamters, etc.

Validate

Tests? Formal Methods? Tools? Regulations? Standards?

Test coverage, source→binary validation,...

Maintain

Graceful degradation? Monitoring? Debugging & patching?

Maximal acceptable failure, overhead of monitoring, patching, significant error msg

How to patch a NN without retraining everything?

What happens if you add AI ?



Specify Input? Output? (*Speed? Memory?)* Privacy? Safety? Security?

Semi-informal text, scope of statement, properties to verify. Auto extraction etc.

Design Algorithm? Components? Pieces? Options?

Model-based tools available (e.g. Papyrus), autogen of code possible

Implement Language? Methods? Types? Data structures? Modules? Interfaces?

Decompose, annotate, comment, autogen from formal spec

Validate Tests? Formal Methods? Tools? Regulations? Standards?

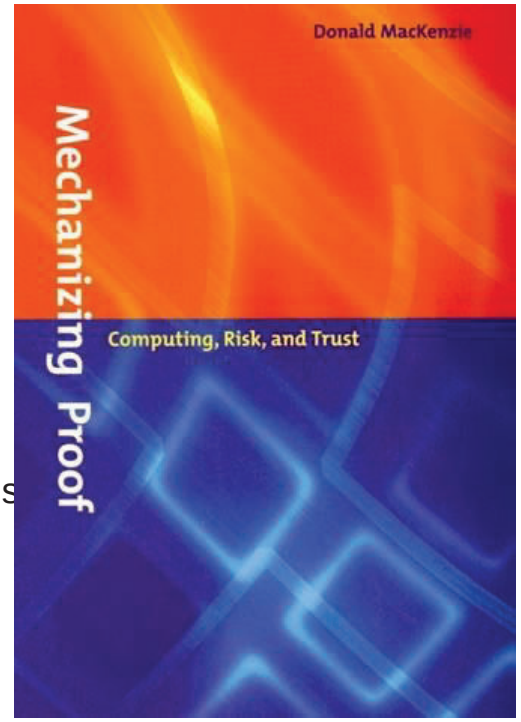
Test coverage, source→binary validation,...

Maintain Graceful degradation? Monitoring? Debugging & patching?

Maximal acceptable failure, overhead of monitoring, patching, significant error msg

What happens if you add AI ?

We have come a VERY long way...



Validate

Tests? **Formal Methods?** Tools? Regulations

Test coverage, source→binary validation,...

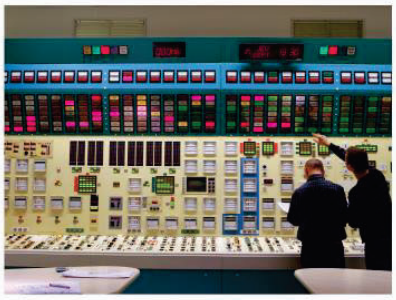
Do we really have to go all the way back…?

Formal Methods

- What are formal methods ?
  - Math-based techniques with rigorously established theoretical foundations
  - Used for the specification, development, test and verification of software and hardware
- Why use formal methods ?
  - Non validated software can have dire consequences and mathematical analysis can contribute to the reliability and robustness
  - Some certification standards call for (e.g., DO-178C for avionics) or even **mandate** (e.g., ISO/IEC 15408) the use of formal methods

Formal Methods

A critical system is a system whose failure may cause physical harm, economical losses or damage the environment

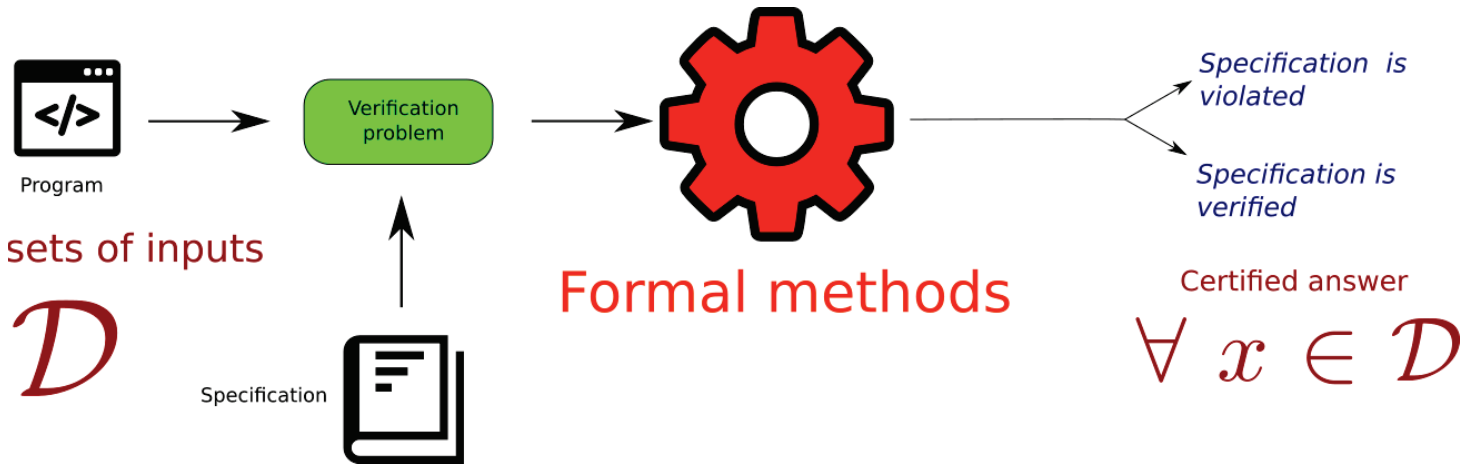






Goal: guarantee that the system respects a *safety specification* ϕ

Formal Methods



Formal Methods

The Ariane 5 reused the inertial reference platform from the Ariane 4, but the Ariane 5's flight path differed considerably from the previous models.

The greater horizontal acceleration caused a data conversion from a 64-bit floating point number to a 16-bit signed integer value to **overflow** and cause a hardware exception.



Formal Methods

The Ariane 5 reused the inertial reference platform from the Ariane 4, but the Ariane 5's flight path differed considerably from the previous models.

The greater horizontal acceleration caused a data conversion from a 64-bit floating point number to a 16-bit signed integer value to **overflow** and cause a hardware exception.

The subsequent automated analysis of the Ariane code (written in Ada) was the first example of large-scale static code analysis by abstract interpretation.



Formal Methods

The Ariane 5 reused the inertial reference platform from the Ariane 4, but the Ariane 5's flight path differed considerably from the previous models.



The greater horizontal acceleration caused a data conversion from a 64-bit floating point number to a 16-bit signed integer value to **overflow** and cause a hardware exception.

The subsequent automated analysis of the Ariane code (written in Ada) was the first example of large-scale static code analysis by abstract interpretation.

If you don't use formal methods, your failure will be in every presentation of every FM conference.

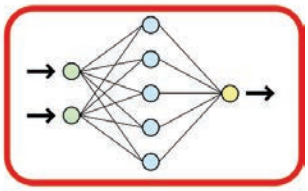
Formal Methods in industry



Tools of formal methods



What happens if you add AI ?

Classical programs	Deep learning
	
Explicit control flow	Generated control flow
Explicit specifications	Implicit specifications
Abstractions and well known concepts	Very few abstractions and reusability
Needs to be robust	Needs to be robust



What happens if you add AI ?

- The first solvers and analyzers were **not** efficient or scalable.
- For example, today's SAT solvers can automatically solve problem instances involving **tens of thousands of variables and millions of constraints**.
- But it wasn't always the case! We needed to invent DPLL, CDCL, Symmetry breaking, two-watched literals, WalkSAT, adaptive branching, random restarts, portfolio, divide-and-conquer, parallel local search…

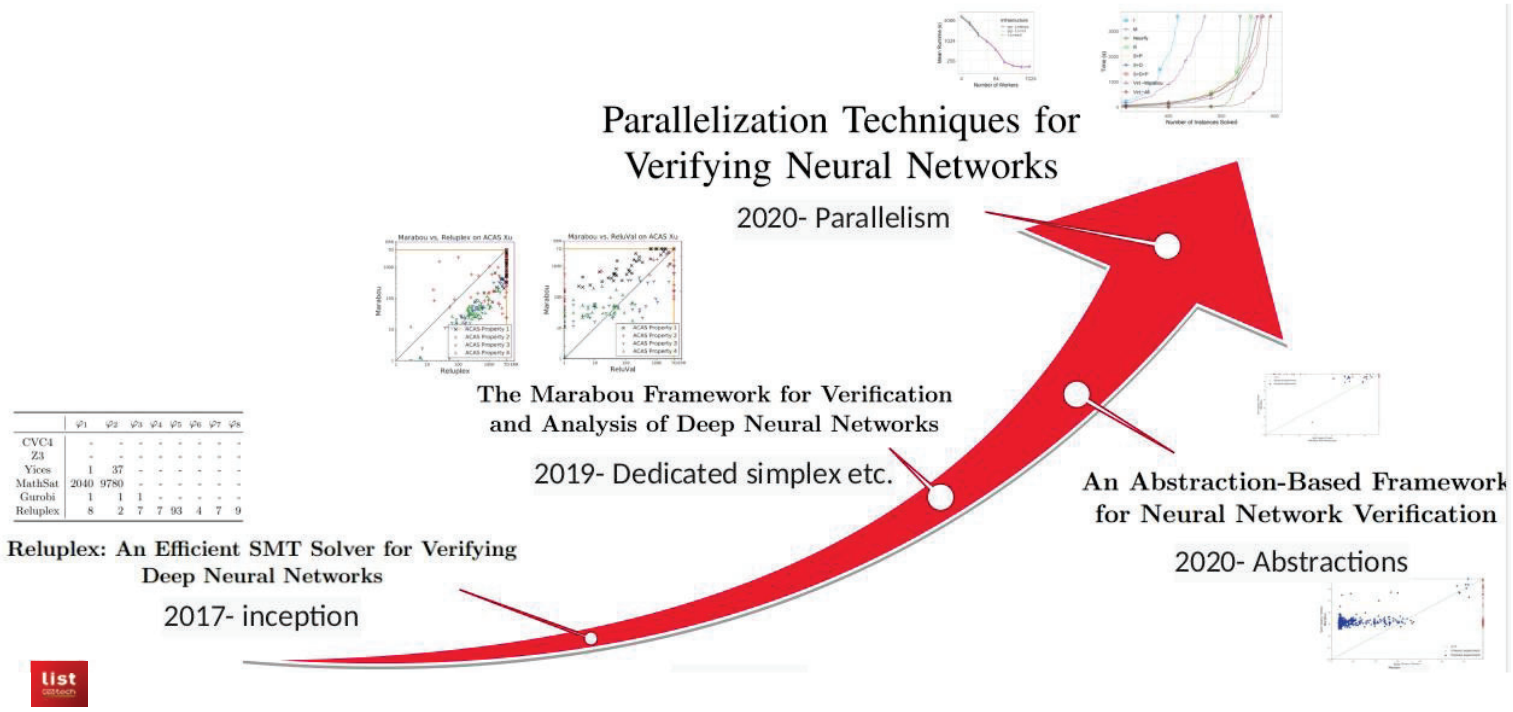
What happens if you add AI ?

Cambrian explosion: Just in the past few years, more than 20 tools.
Competition for resources: Each paper published increases the scalability.
Cross-fertilization: Good ideas from one tool are implemented in others.
Niche creation: Some solvers are more specialized into particular models and type of properties.
Adaptative Pressure: New models, new architectures, and in general new AI-technologies are born every year and the tools to validate them must keep up.
Domestication: ML practitioners should be made aware of the choices in implementation that can make their models more amenable to FM, so that they can factor this aspect in their decision process.

What happens if you add AI ?

- Artificial Intelligence Safety Engineering (WAISE, at SafeComp)
- AISafety (at IJCAI)
- Safe AI (at AAAI)
- Verification of Neural Networks (VNN, at AAAI or CAV)
- Formal Methods for ML-Enabled Autonomous Systems (FoMLAS, at CAV)
- Machine Learning with Guarantees (ML with Guarantees, at NeurIPS)
- Safe Machine Learning (SafeML, at ICLR)
- Privacy in Machine Learning (PriML, at NeurIPS)
- Security and Safety in Machine Learning Systems (AISecure, at ICLR)
- Dependable and Secure Machine Learning (DSML, at DSN)

What happens if you add AI ?



The general picture

Verification and Validation of AI

A three-players game

A three-players game

Developer's side

- What is the architecture of the software, how can it be modified to be more amenable to verification, will these modifications cost too much ?
  (Activation functions of NN, kernel function of SVM, etc.)

Object to certify

- What to verify, how to formally specify it, how is it decomposed in smaller bits?
  (Robustness, metamorphism, behavior specification, etc.)

Properties to verify

Validator's side

- How to verify, what methods fit my problem, can the tools be helped with heuristics?
  (Abstract interpretation, SMT solving, symbolic execution, Constraint programming, etc.)

Methods and tools

Set up of the game

Questions to ask, things to consider, "be prepared"

Set up of the game

What is more important ?
- Fewer false negatives ? Fewer false positives ? ...

Have we covered all necessary inputs ?
- Have we overlooked important factors ?
- Are there unimportant factors ? (e.g. “gender” for hiring algorithms)
- Are the ranges of inputs too narrow (cases overlooked) or too wide (costly verification)?

What performance do I need ?
- Will I need a split-second decision ? (time constraints)
- Will I embed my system in a limited resources environment (memory consumption)

What are the attack surfaces my workflow? (more cyber)
- Am I including publicly available resources in my training data ? (data poisoning)
- Will my system answer queries over the internet? (Model theft)
- Will it use data from unprotected sources? (adversarial attacks)

The Object

What is the architecture of the software,
how can it be modified to be more amenable to verification,
will these modifications cost too much

Influence of the architecture

Several architecture choices influence what we can do:

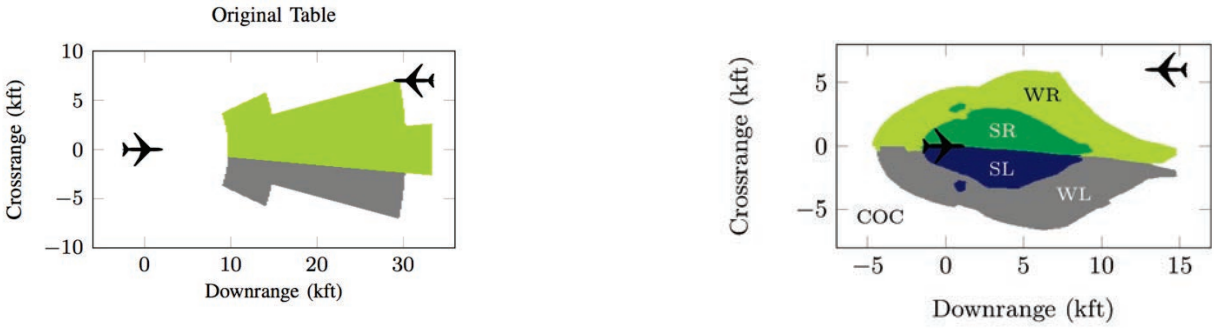
- Activation function ? Can we use ReLU ?
- Connectivity ? Fewer connections lead to more scalable verification.
- Number of parameters ? Can we prune the network ?
- The objective function ? Can we modify it so that it does not only learn the goal of the NN but also "some" information to guide the provers' heuristics ?

The Property

What to verify,
how to *formally* specify it,
how is it decomposed in smaller bits

Properties on structured data

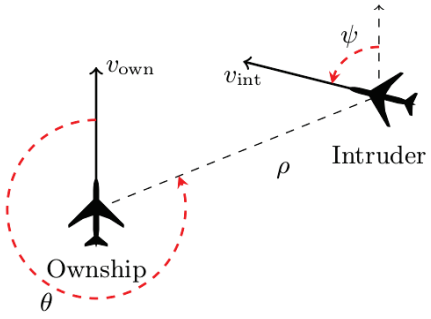
There are two possibilities: The inputs are specifiable (a distance, a speed, a location, etc.). e.g. Reluplex paper applied to an Air-Collision Avoidance System.



Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks (2017, Katz, Barrett, Dill, Julian, Kochenderfer)

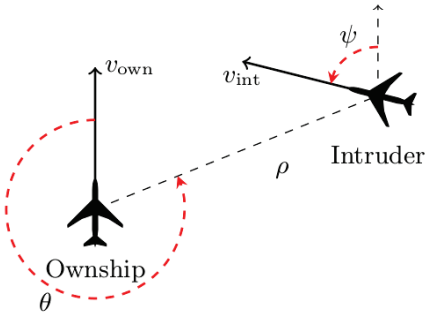
Properties on structured data

There are two possibilities: The inputs are specifiable (a distance, a speed, a location, etc.). e.g. Reluplex paper applied to an Air-Collision Avoidance System.



Properties on structured data

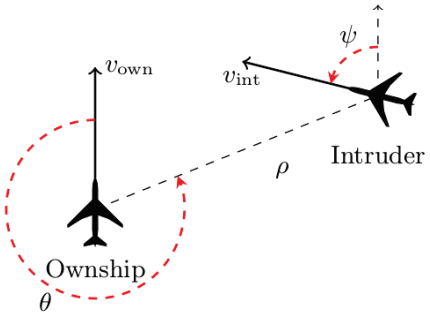
There are two possibilities: The inputs are specifiable (a distance, a speed, a location, etc.). e.g. Reluplex paper applied to an Air-Collision Avoidance System.



Description: If the intruder is near and approaching from the left, the network advises "strong right".

Properties on structured data

There are two possibilities: The inputs are specifiable (a distance, a speed, a location, etc.). e.g. Reluplex paper applied to an Air-Collision Avoidance System.



Description: If the intruder is near and approaching from the left, the network advises "strong right".

Input constraints: $250 \leq \rho \leq 400$, $0.2 \leq \theta \leq 0.4$, $-3.141592 \leq \psi \leq -3.141592 + 0.005$, $100 \leq v_{\text{own}} \leq 400$, $0 \leq v_{\text{int}} \leq 400$.

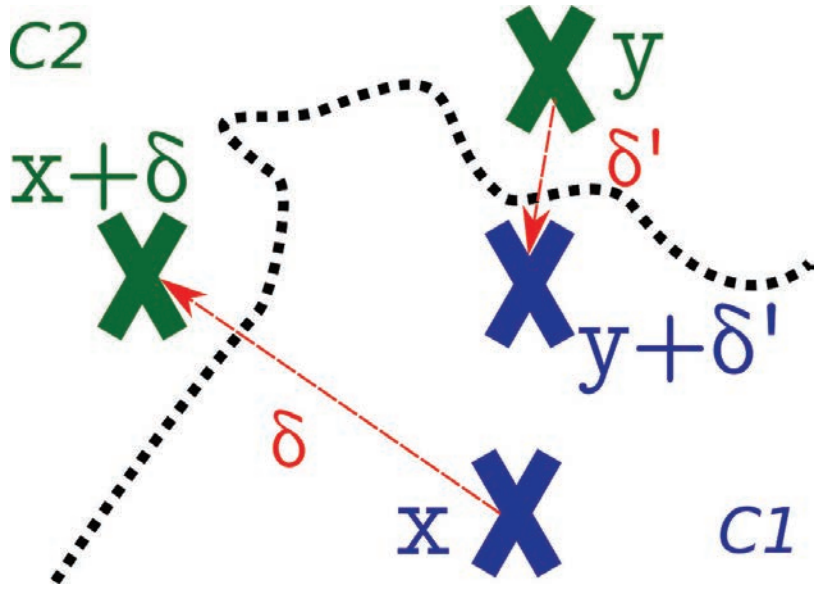
Properties on unstructured data

The inputs are **not specifiable** (images). For example, what is a pedestrian?

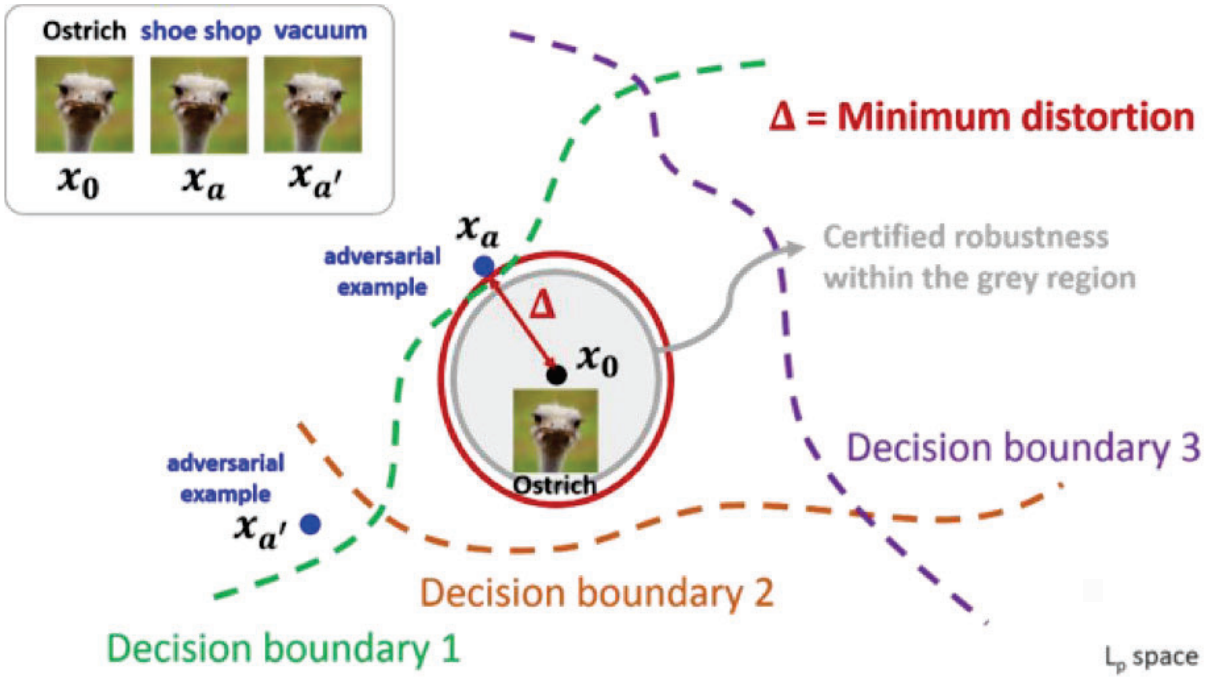
How can you formally specify the desirable property "if pedestrian on the road, then brake"?

These properties are different: considering notions such as "input space" and "points that are close to each other", a property could be "if two inputs are closer than some measure d, they should be classified the same".

Properties on unstructured data



Properties on unstructured data



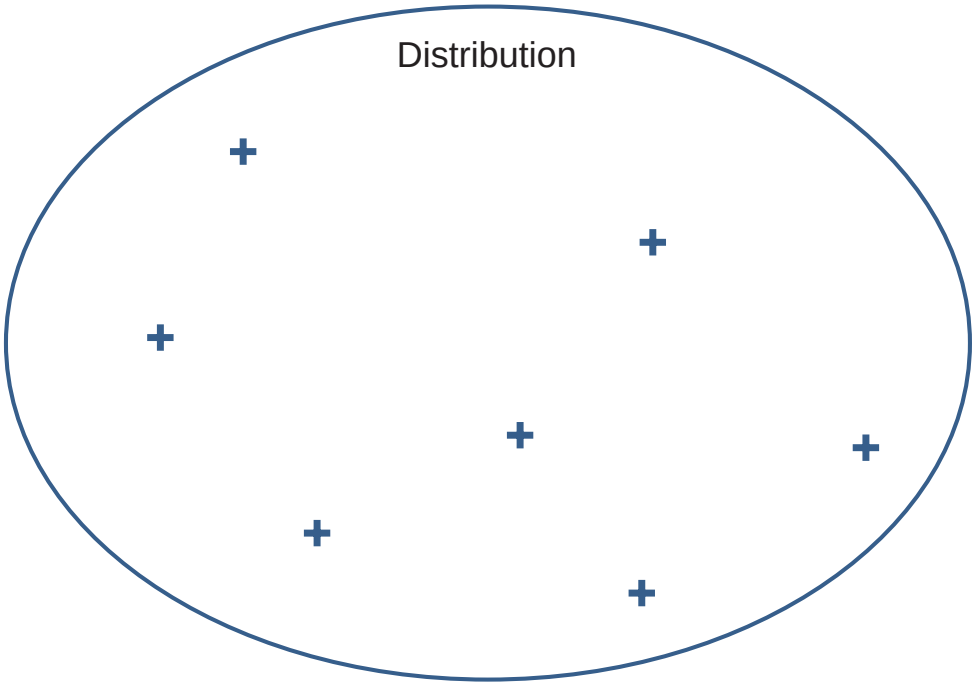
Properties on unstructured data

Can be used to account for sensors error, for example

$$\forall x.\forall y.(|x - x^*| \leq \delta \wedge y = \nu(x)) \implies |y - y^*| \leq \epsilon$$

Properties on unstructured data

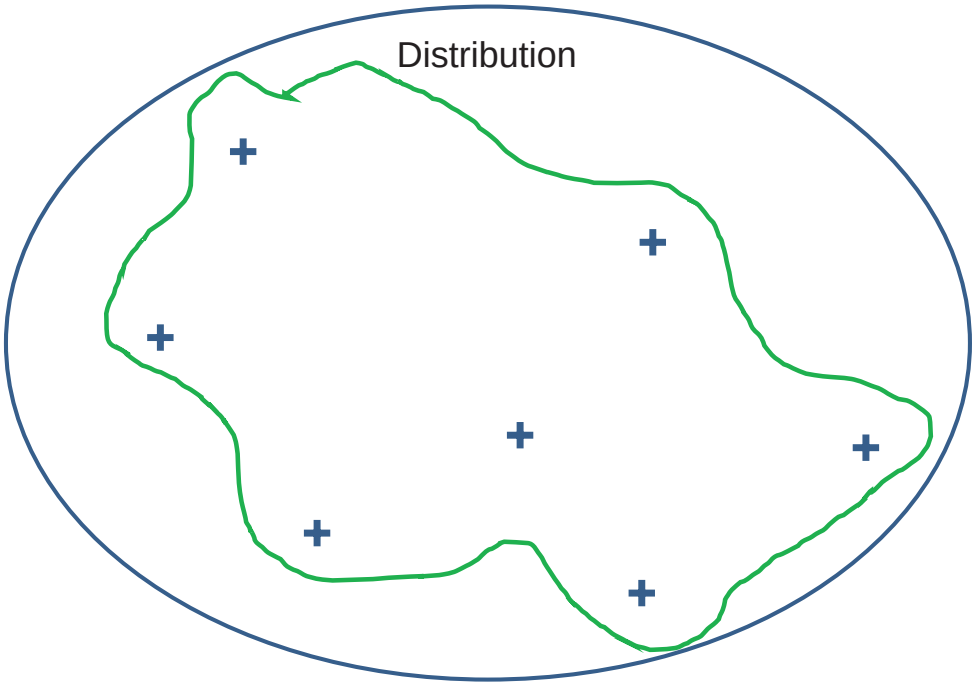
Ideally the selected data to build and validate the model is representative of the intended distribution.



40

Properties on unstructured data

Ideally the selected data to build and validate the model is representative of the intended distribution.

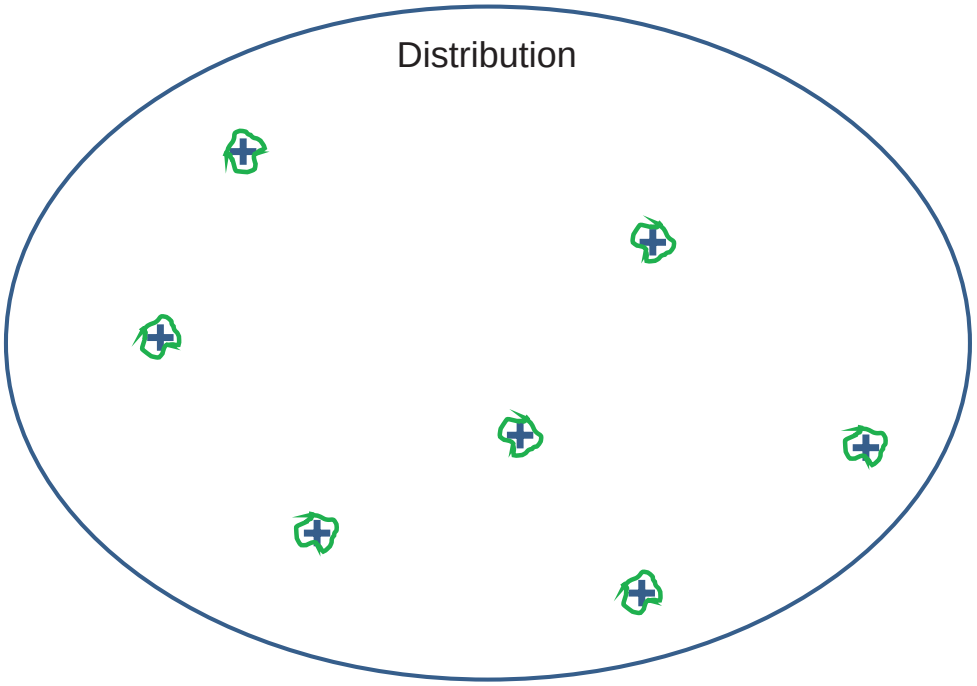


41

Properties on unstructured data

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown



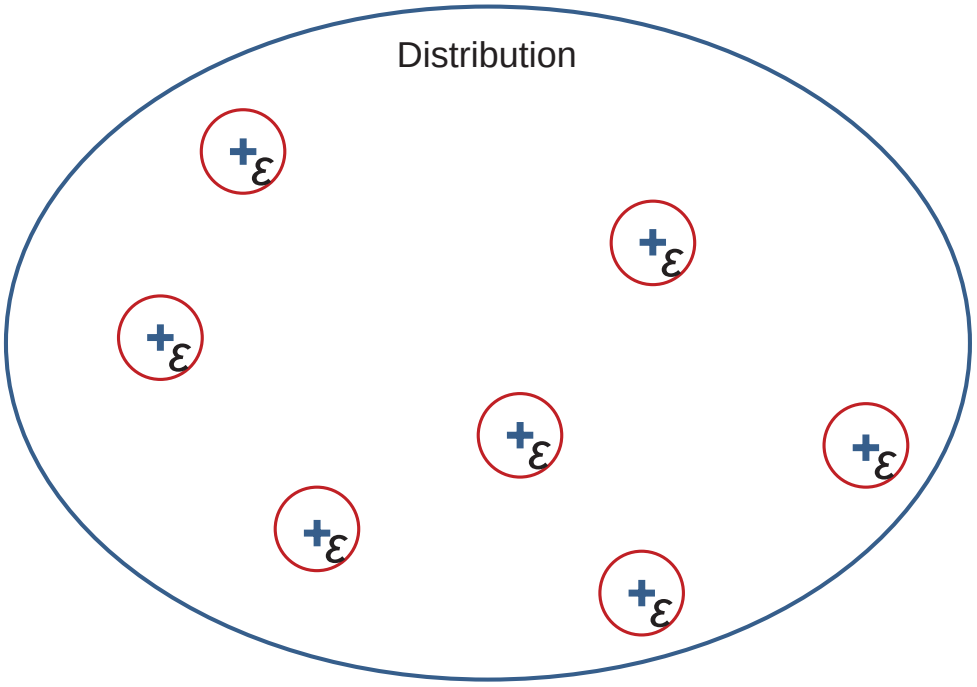
42

Properties on unstructured data

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?



43

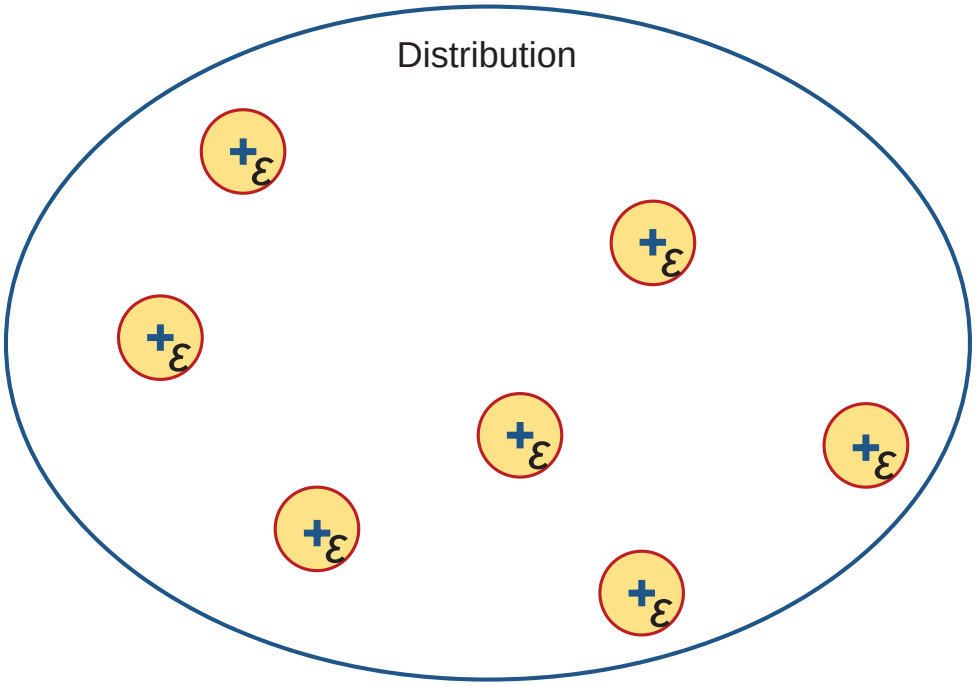
Properties on unstructured data

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

- Formally **verify** : guarantee, but can be costly



44

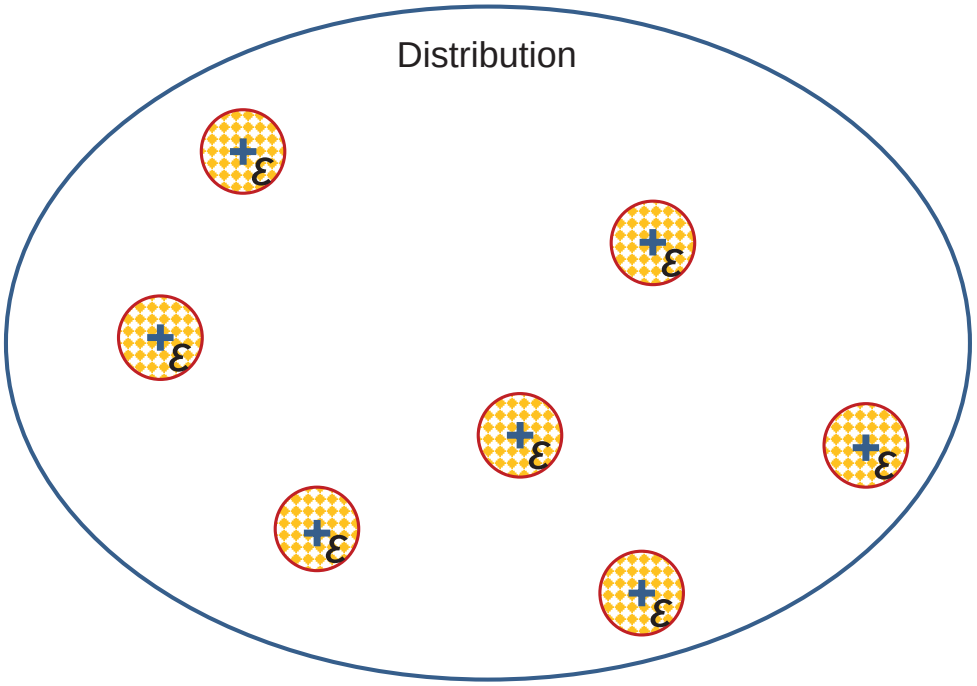
Properties on unstructured data

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

- Formally **verify** : guarantee, but can be costly
- **Test a lot**: no guarantee, but faster



45

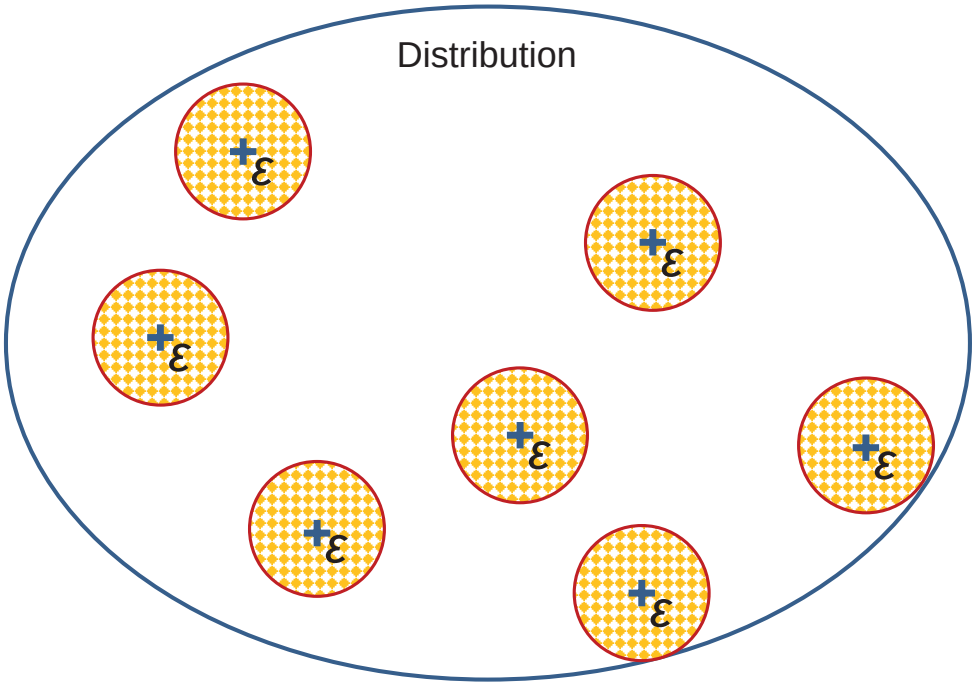
Properties on unstructured data

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

- Formally **verify** : guarantee, but can be costly
- **Test a lot**: no guarantee, but faster



46

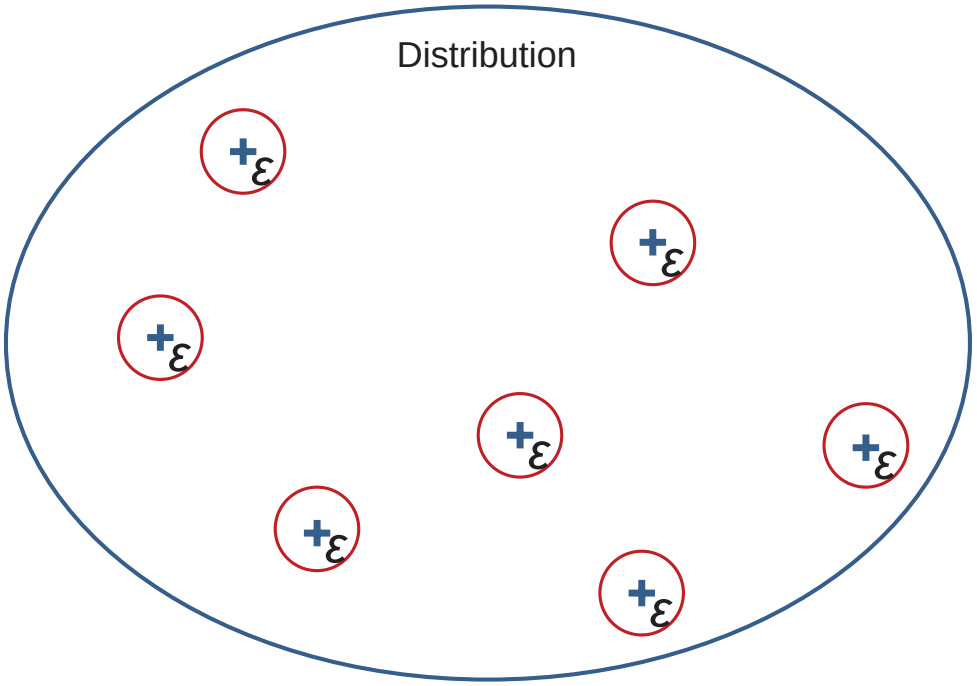
Properties on unstructured data

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

What is it good for?



47

Properties on unstructured data

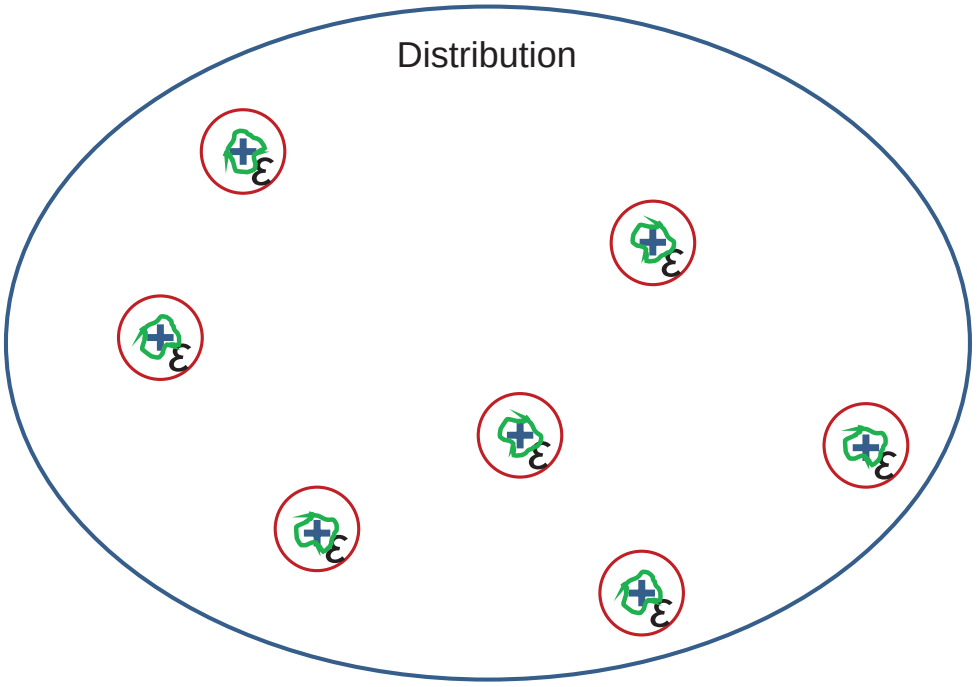
Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

What is it good for?

Can detect this…



48

Properties on unstructured data

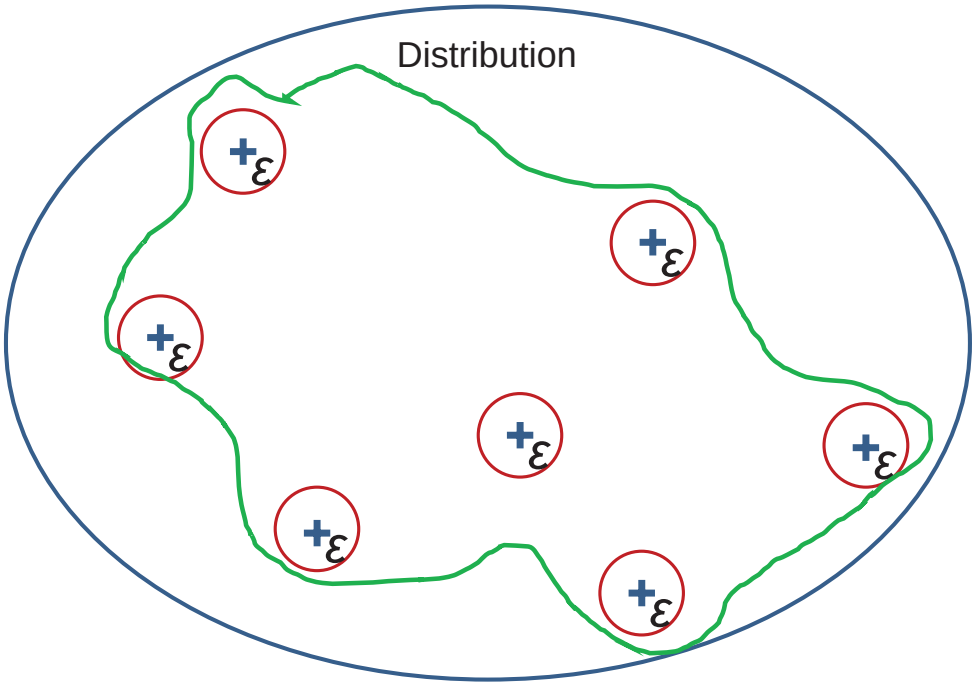
Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

What is it good for?

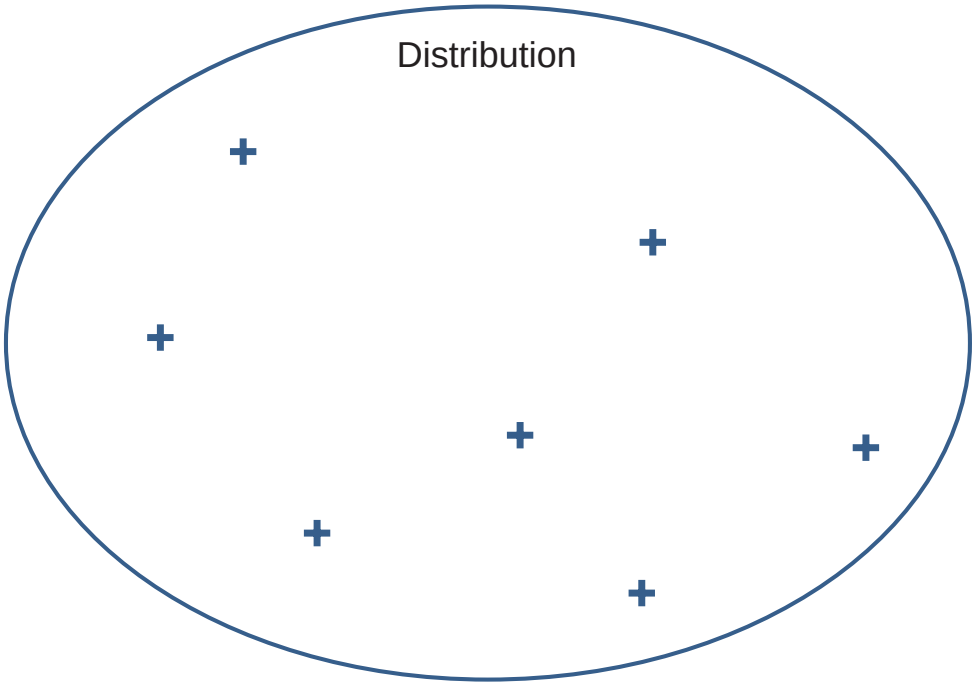
… But doesn't imply this



49

Properties on unstructured data

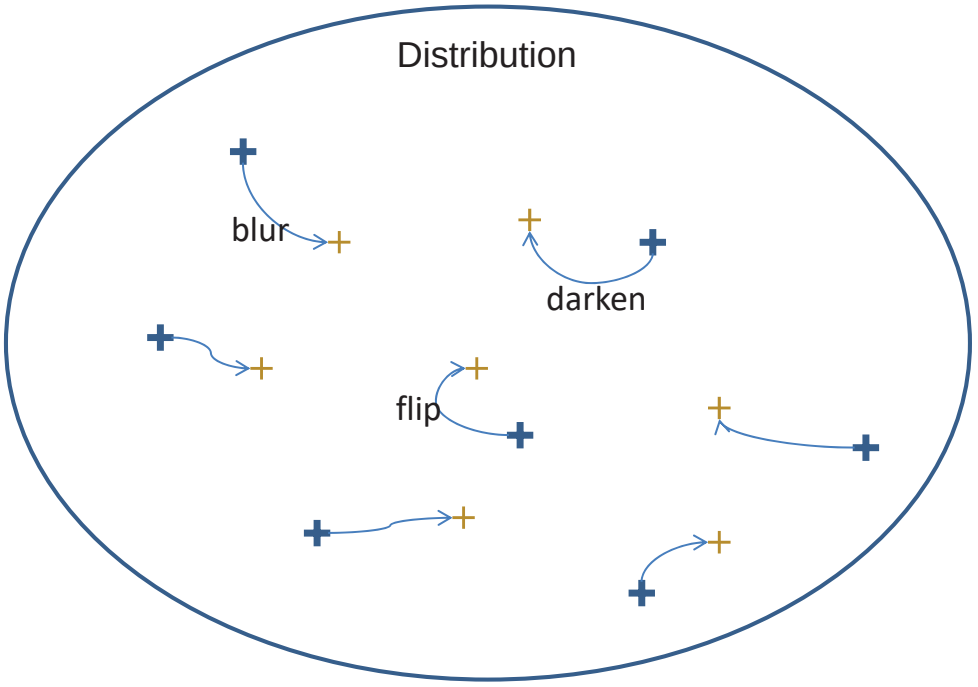
Can also generate other data from acceptable transformations



50

Properties on unstructured data

Can also generate other data from acceptable transformations

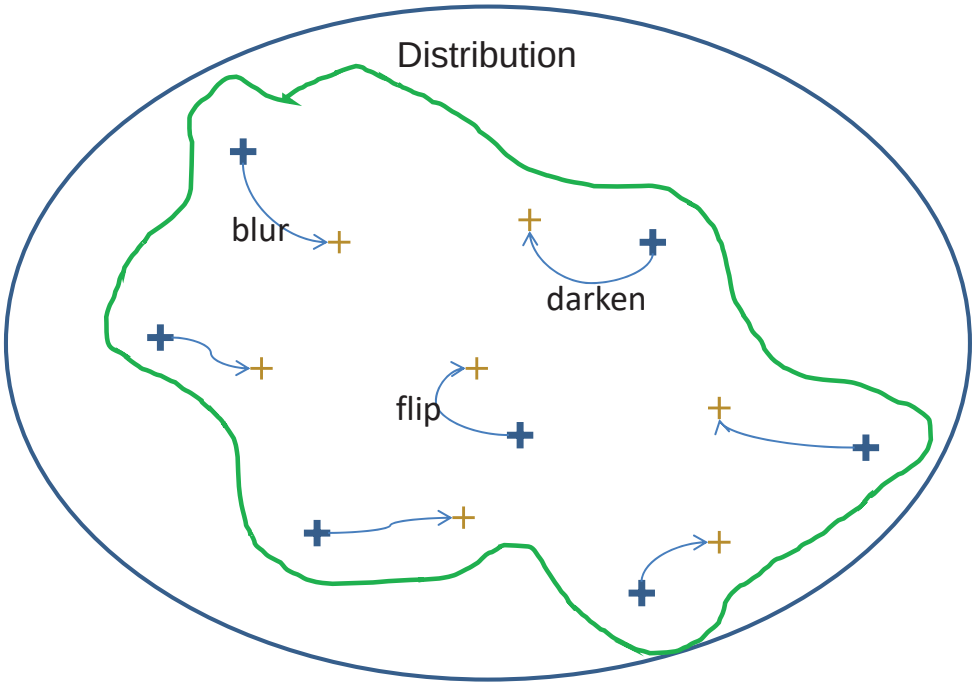


51

Properties on unstructured data

Can also generate other data from acceptable transformations

To see if the network can handle a wider distribution.



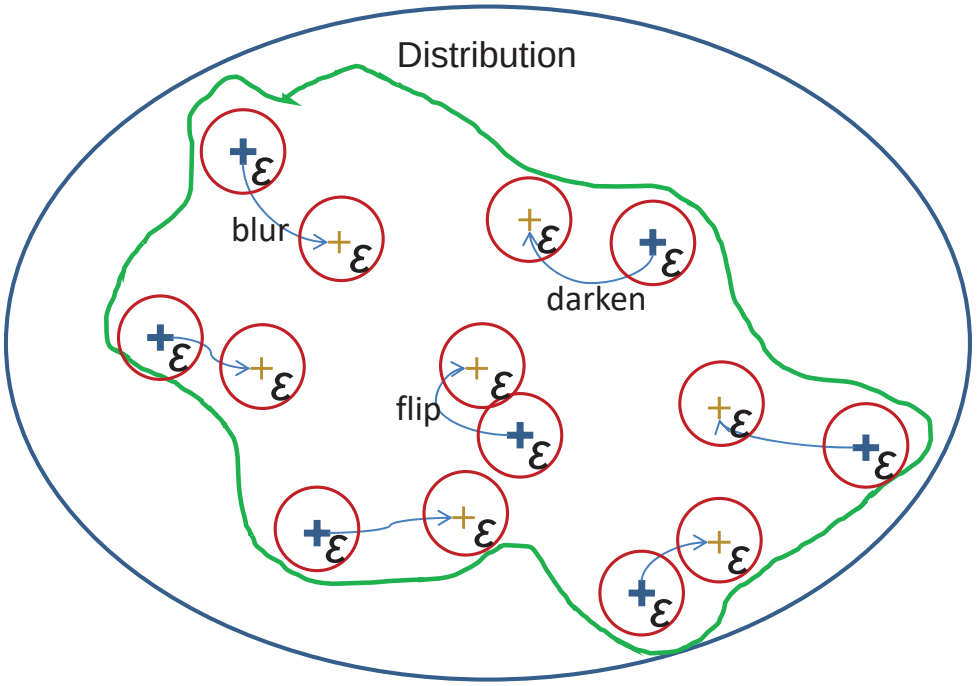
52

Properties on unstructured data

Can also generate other data from acceptable transformations

To see if the network can handle a wider distribution.

And we can combine methods



53

Properties

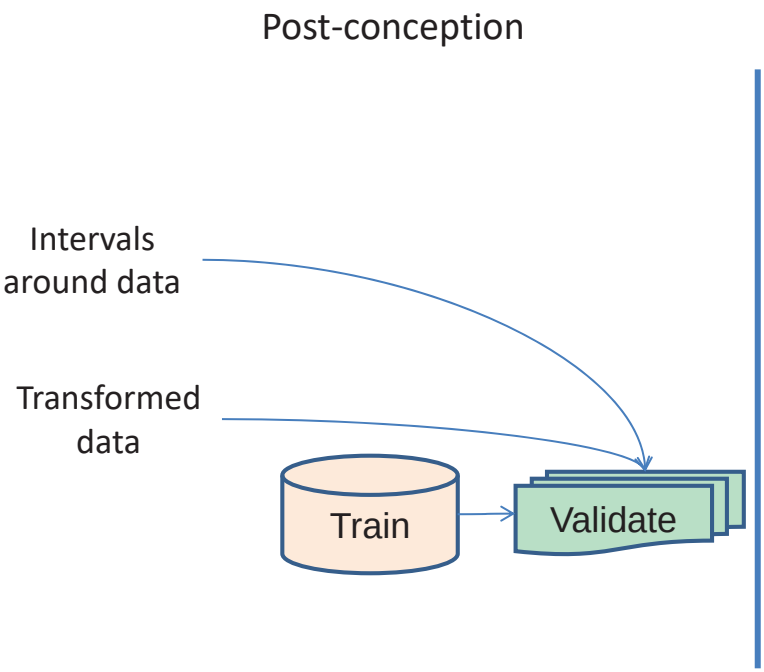
Formally specifiable (structured data)

- **Semantics** (directly related to the domain) can be described through math and logic
  - The input is a distance, a speed, a physical property (e.g., an observable), a tabular data, (e.g., an age), etc.
- Can be proved on all possible inputs

Not directly specifiable (unstructured data)

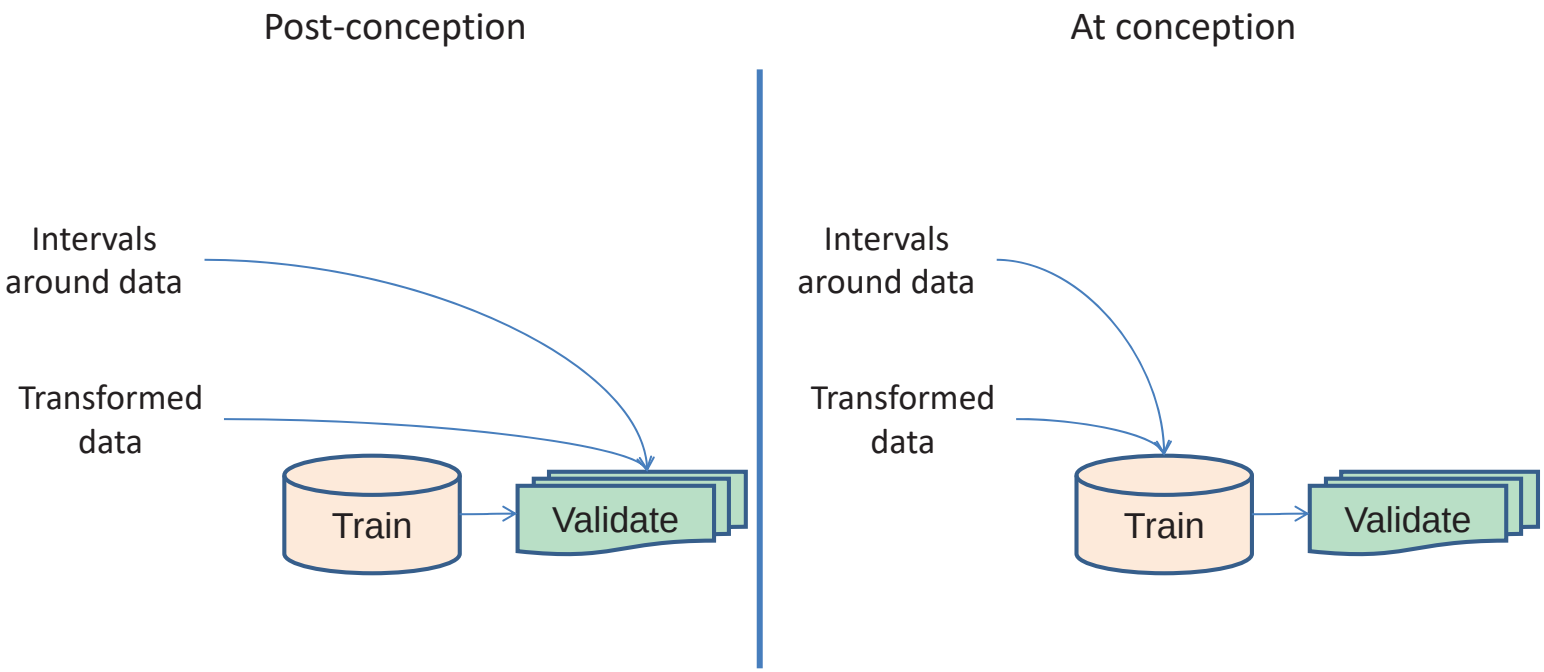
- Semantics of the input cannot be described through math and logic (e.g. an image containing a jacket)
  - The input is a collection of semanticless values (a pixel) that, together, can form a meaning
- Only abstracted properties can be described through math and logic (e.g. "distance" between two images)
- Cannot realistically be proved on all possible inputs

Side note: what phase ?



55

Side note: what phase ?



The Methods

How to verify,
what methods fit my problem,
can the tools be helped with heuristics

Plan

AI



Safety

Constraint programming

Abstract Interpretation

Monitoring

B-Method

Satisfiability Modulo Theory

Hoare logic

SAT Solving

Linear programming

Interactive provers

Automatic provers

Runtime verification

Metamorphism

Tests

state machines

Adversarial training

automata

Symbolic Execution

process algebra

formal semantics

Plan

AI

Neural networks
Capsule networks
Support Vector Machines
Random Forests
Expert Systems
.
.
.

Safety

Constraint programming
Monitoring
Abstract Interpretation
B-Method
Satisfiability Modulo Theory
Hoare logic
SAT Solving
Linear programming
Interactive provers
Automatic provers
Runtime verification
Metamorphism
Tests
state machines
Adversarial training
automata
Symbolic Execution
process algebra
formal semantics

Plan

AI

Neural networks

Safety

Abstract Interpretation

Satisfiability Modulo Theory

Metamorphism

Symbolic Execution

Metamorphic testing

Sometimes we don't have an oracle.

Rely on **properties**.

Find relations between inputs: *x R y*
That imply relations on the outputs: *f(x) R' f(y)*.

Metamorphic testing

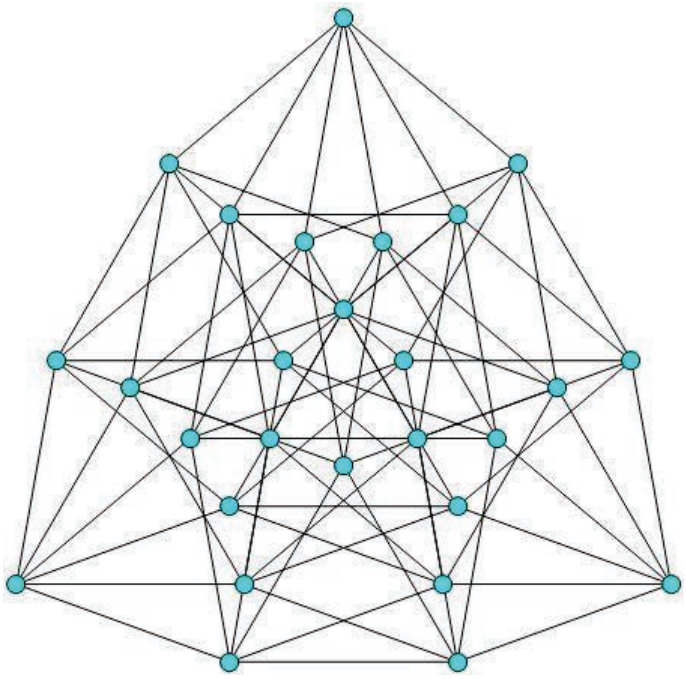
Sometimes we don't have an oracle.

Rely on **properties**.

Find relations between inputs: *x R y*
That imply relations on the outputs: *f(x) R' f(y)*.

e.g. Instead of testing that the algorithm finds the shortest path, we check that it preserves the following property:

If there is a shortest path, then :
$A \rightarrow B = B \rightarrow A$
For any two points A, B in a graph.



Metamorphic testing







Metamorphic testing






Metamorphic testing

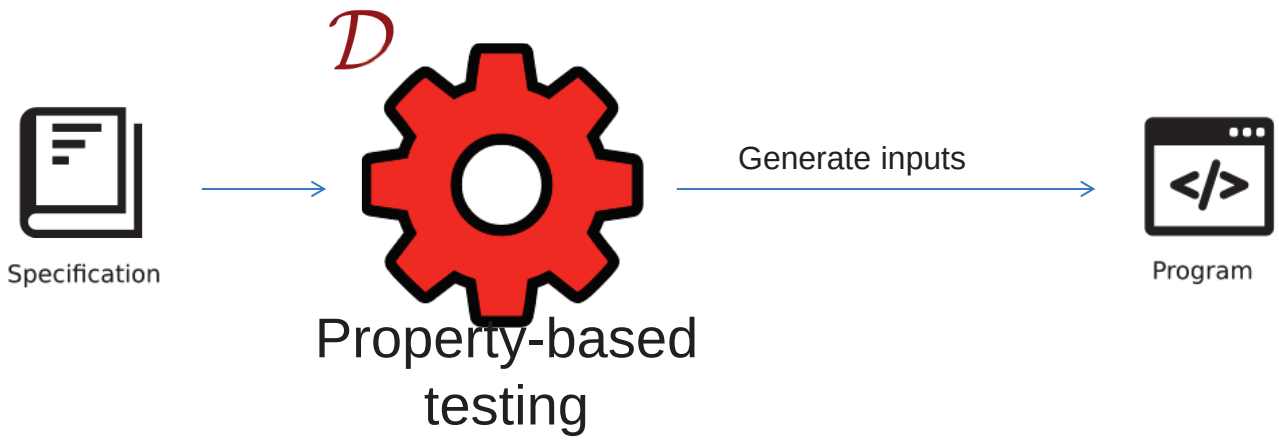


Metamorphic testing





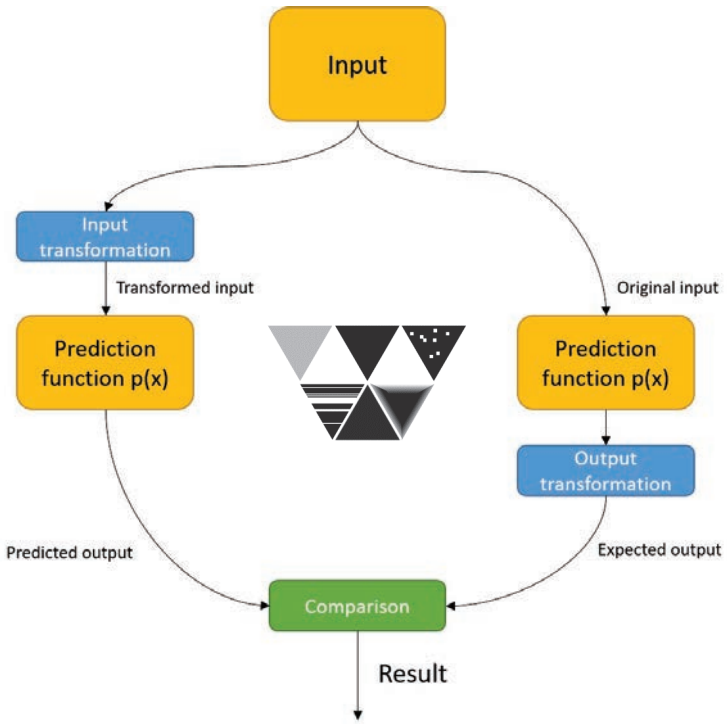
Metamorphic testing



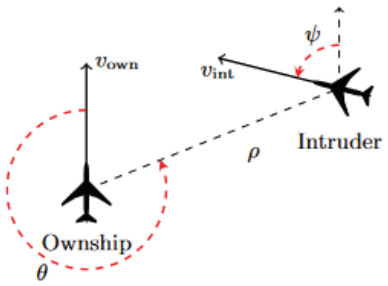
Real world example: through AIMOS

AI Metamorphism Observing Software

- Testing AI component is crucial as unexpected inputs can bring unforeseen consequences to the whole system
- As part of the validation process of AI component we propose AIMOS to test metamorphic properties on them
- Designed to be as agnostic as possible of the model and work in most of the cases
- Allows to compare multiple models



Real world example: ACAS through AIMOS

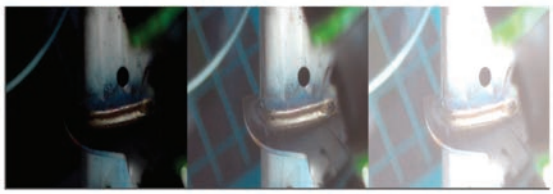


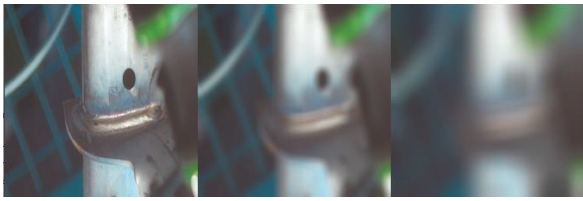
Right Left

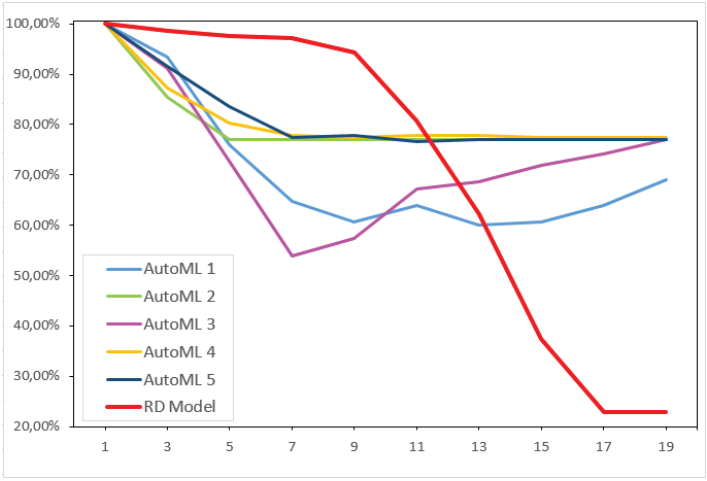
Previous advisory	Stability	Stability on restricted space
Clear-of-conflict	97.4%	89.7%
Left	97.6%	95.9%
Right	97.5%	99.6%
Strong left	97.5%	95.3%
Strong right	97.8%	99.8%

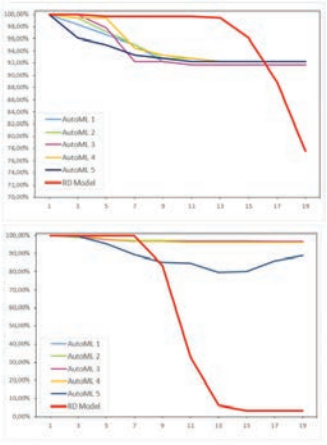
Real world example: Renault through AIMOS











Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

Satisfiability Modulo Theory

Any logical formula can be converted
into **Conjunctive Normal Form**

Satisfiability Modulo Theory

Equivalence rules can be used to translate any formula to CNF.

eliminate $\Rightarrow$	$A \Rightarrow B \equiv \neg A \vee B$
reduce the scope of $\neg$	$\neg(A \vee B) \equiv \neg A \wedge \neg B$, $\neg(A \wedge B) \equiv \neg A \vee \neg B$
apply distributivity	$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$, $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

Satisfiability Modulo Theory

However, there is a *linear time* translation to CNF that produces an *equisatisfiable* formula. Replace the distributivity rules by the following rules:

$$\frac{F[l_i \ op \ l_j]}{F[x], x \Leftrightarrow l_i \ op \ l_j}\,(*)$$

$$\frac{x \Leftrightarrow l_i \vee l_j}{\neg x \vee l_i \vee l_j, \neg l_i \vee x, \neg l_j \vee x}$$

$$\frac{x \Leftrightarrow l_i \wedge l_j}{\neg x \vee l_i, \neg x \vee l_j, \neg l_i \vee \neg l_j \vee x}$$

(*) x must be a fresh variable.

Satisfiability Modulo Theory

Translation of $(p \wedge (q \vee r)) \vee t$:

$$(p \wedge (q \vee r)) \vee t$$

$$(p \wedge x_1) \vee t, x_1 \Leftrightarrow q \vee r$$

$$x_2 \vee t, x_2 \Leftrightarrow p \wedge x_1, x_1 \Leftrightarrow q \vee r$$

$$x_2 \vee t, \neg x_2 \vee p, \neg x_2 \vee x_1, \neg p \vee \neg x_1 \vee x_2, x_1 \Leftrightarrow q \vee r$$

$$x_2 \vee t, \neg x_2 \vee p, \neg x_2 \vee x_1, \neg p \vee \neg x_1 \vee x_2, \neg x_1 \vee q \vee r, \neg q \vee x_1, \neg r \vee x_1$$

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a'+b
b'+c
c+e
c'+a'

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a'+b
b'+c
c+e
c'+a'

Decide a

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

b
b'+c
c+e
c'

Implies...
- b is true
- c is false

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

b

c

c+e

c'

Implies...

- b is true
  - c is true
- c is false

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a'+b
b'+c
c+e
c'+a'

Decide a
=> Contradiction.

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a'+b
b'+c
c+e
c'+a'

Backtrack

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a'+b
b'+c
c+e
c'+a'

Now we know :
a **must be** *false*

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

b'+c
c+e

Now we know :
a **must be** *false*

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

b'+c
c+e

Decide b

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

c
c+e

Implies
- c is true

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

c

c+e

Implies
- c is true

Then e can have any value

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

c

c+e

Implies

- c is true

Then e can have any value

Let's decide e is true

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

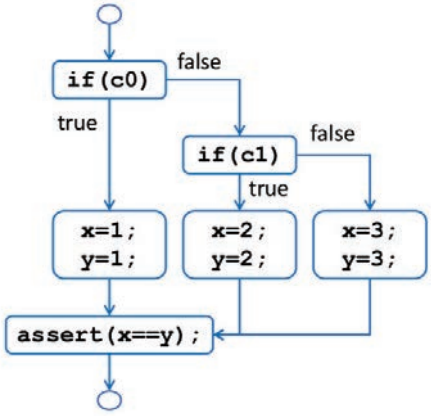
a'+b
b'+c
c+e
c'+a'

So now the assignment is:
False: a
True: b, c, e

Satisfiability Modulo Theory

```
if(c0)
  x=1;y=1;
else
  if(c1)
    x=2;y=2;
  else
    x=3;y=3;
assert(x==y);
```

```
if(c0)
  x1=1;y1=1;
else
  if(c1)
    x2=2;y2=2;
  else
    x3=3;y3=3;
x4=ite(c0,x1,ite(c1,x2,x3));
y4=ite(c0,y1,ite(c1,y2,y3));
assert(x4==y4);
```



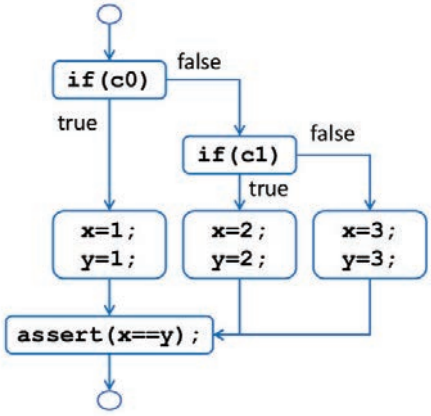
$$c_0 \rightarrow x_4 = 1 \qquad c_0 \rightarrow y_4 = 1$$
$$\neg c_0 \wedge c_1 \rightarrow x_4 = 2 \qquad \neg c_0 \wedge c_1 \rightarrow y_4 = 2$$
$$\neg c_0 \wedge \neg c_1 \rightarrow x_4 = 3 \qquad \neg c_0 \wedge \neg c_1 \rightarrow y_4 = 3$$

Control flow-guided smt solving for program verification (2018, Chen, Jianhui, and Fei He)

Satisfiability Modulo Theory

```
if(c0)
  x=1;y=1;
else
  if(c1)
    x=2;y=2;
  else
    x=3;y=3;
assert(x==y);
```

```
if(c0)
  x1=1;y1=1;
else
  if(c1)
    x2=2;y2=2;
  else
    x3=3;y3=3;
x4=ite(c0,x1,ite(c1,x2,x3));
y4=ite(c0,y1,ite(c1,y2,y3));
assert(x4==y4);
```



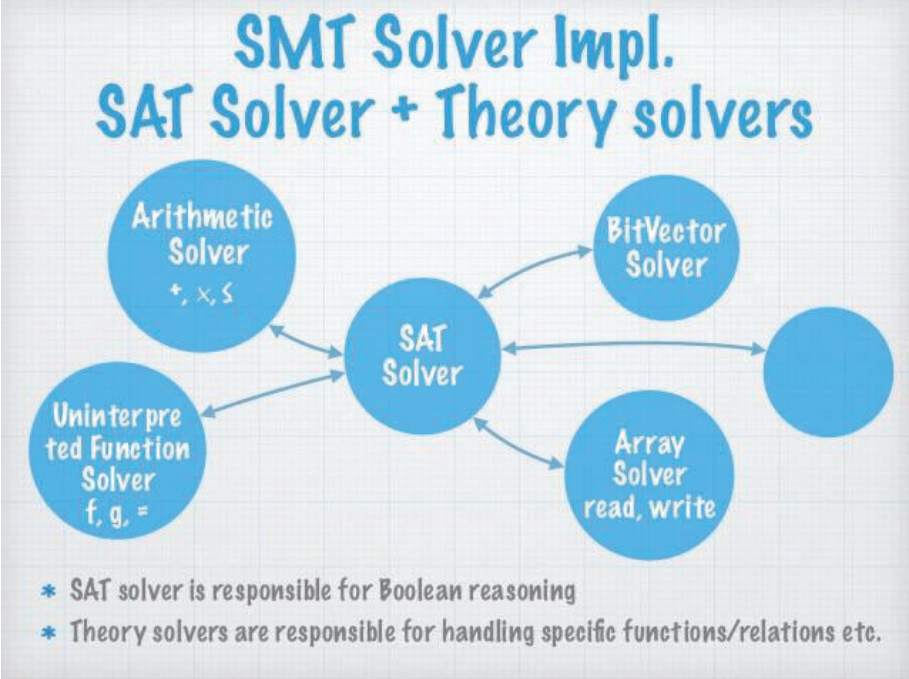
????

$$c_0 \rightarrow x_4 = 1 \qquad c_0 \rightarrow y_4 = 1$$

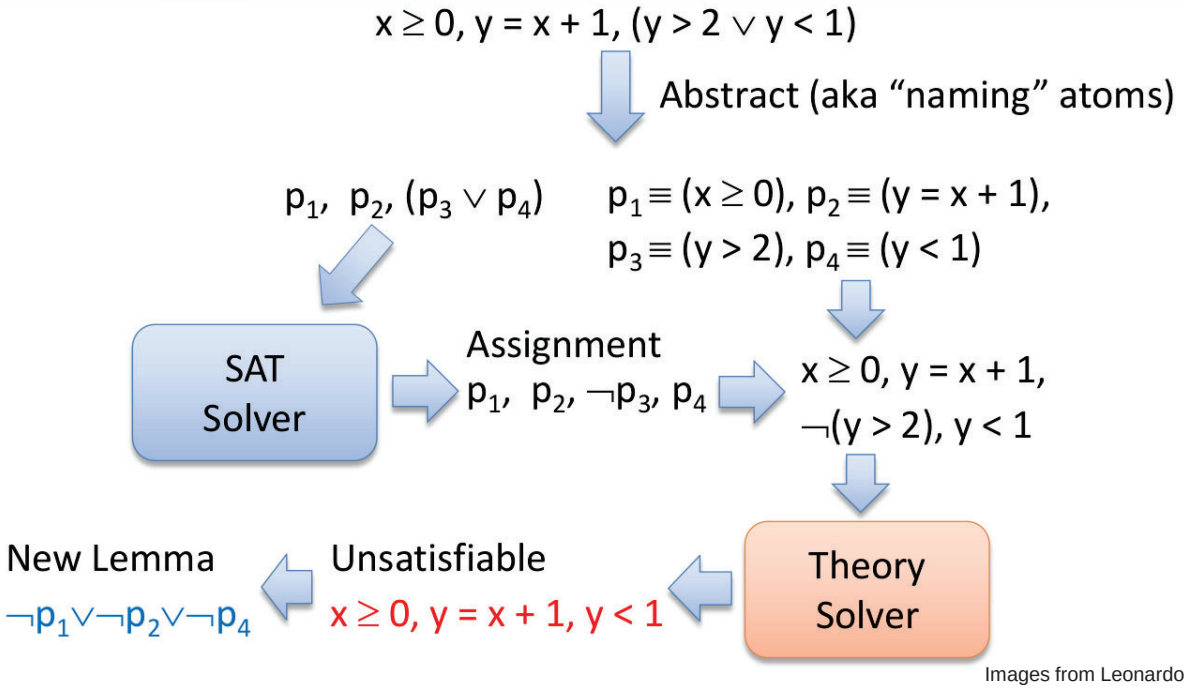
$$\neg c_0 \wedge c_1 \rightarrow x_4 = 2 \qquad \neg c_0 \wedge c_1 \rightarrow y_4 = 2$$

$$\neg c_0 \wedge \neg c_1 \rightarrow x_4 = 3 \qquad \neg c_0 \wedge \neg c_1 \rightarrow y_4 = 3$$

Satisfiability Modulo Theory



Satisfiability Modulo Theory



Images from Leonardo de Moura

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a : x > 3

a'+b
b'+c
c+e
c'+a'

So now the assignment is:
False: a
True: b, c, e

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a : x > 3 means a' : x <= 3

a'+b
b'+c
c+e
c'+a'

So now the assignment is:
False: a
True: b, c, e

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a : x > 3 means a' : x <= 3

Suppse e stands for x > 5

a'+b
b'+c
c+e
c'+a'

So now the assignment is:
False: a
True: b, c, e

Satisfiability Modulo Theory

A literal is an atom or its negation (a or a'). Clauses are disjunctions of literals (e.g. a+b'+c). A SAT problem is a conjunction of clauses. This is called **Conjunctive Normal Form.**

A great deal of research is dedicated to speed up SAT solving (a notoriously NP-complete problem), such as Conflict Driven Clause Learning, or symmetry breaking. But we will not see that here.

a : x > 3 means a' : x <= 3

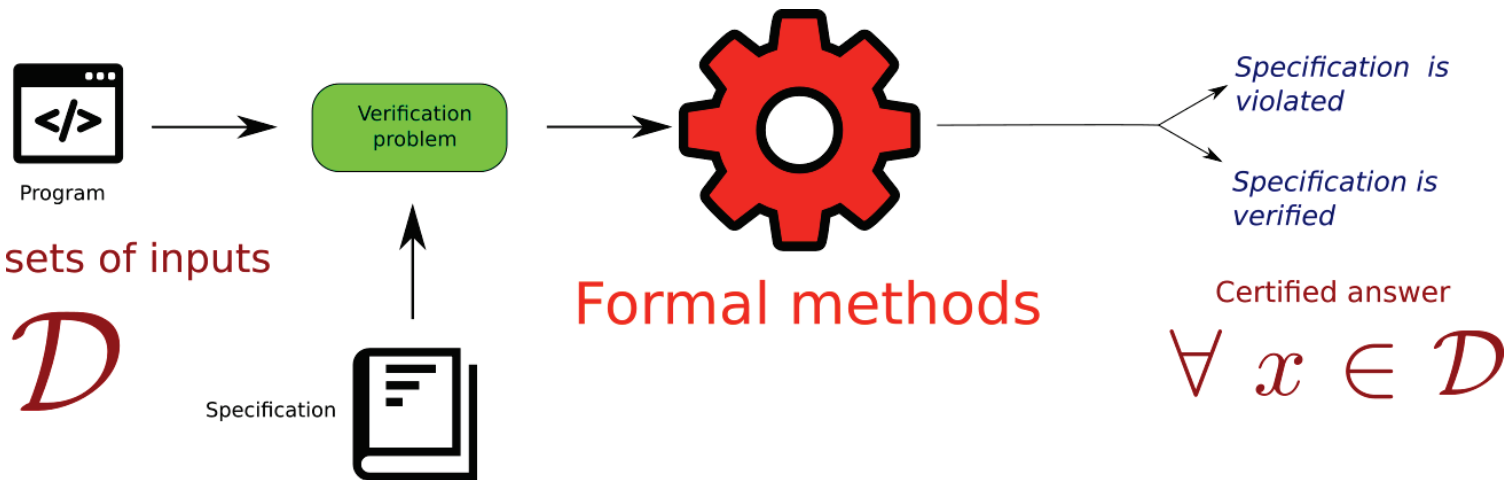
Suppse e stands for x > 5

Now suddenly, our assignment is **no longer valid**. Not at the **Boolean level** , but at the **theory level.**

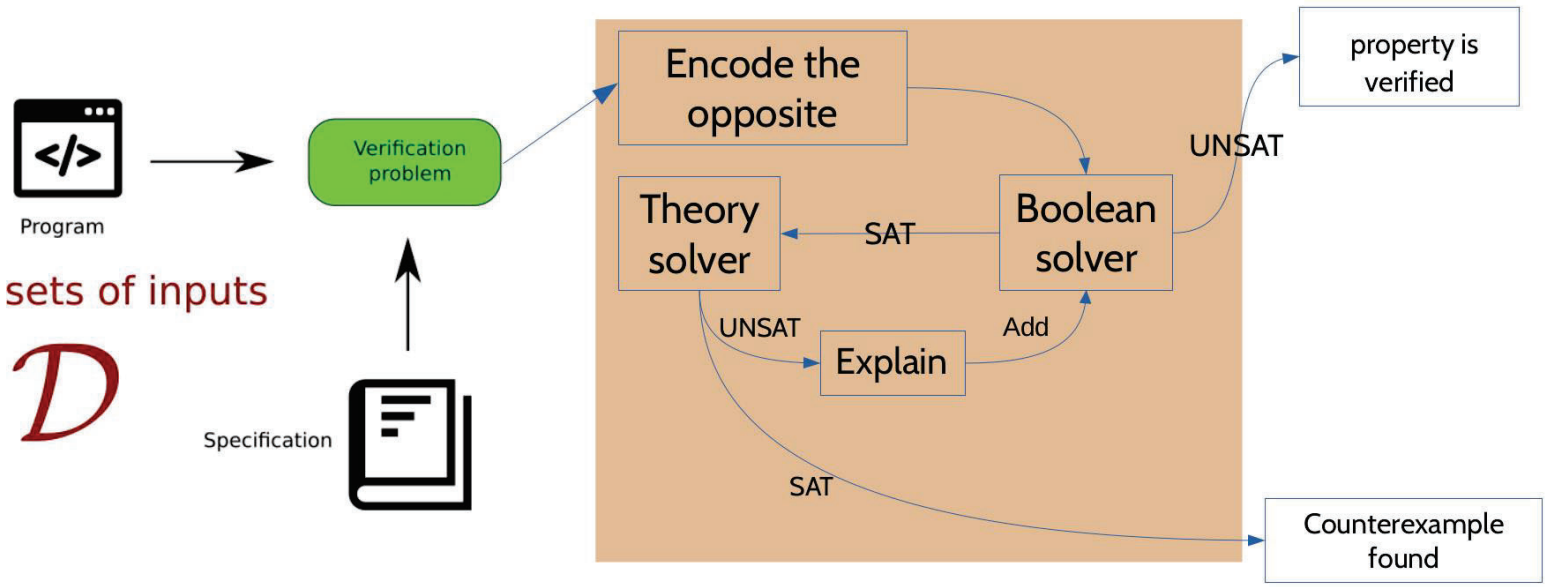
a'+b
b'+c
c+e
c'+a'

So now the assignment is:
False: a
True: b, c, e

Formal Methods



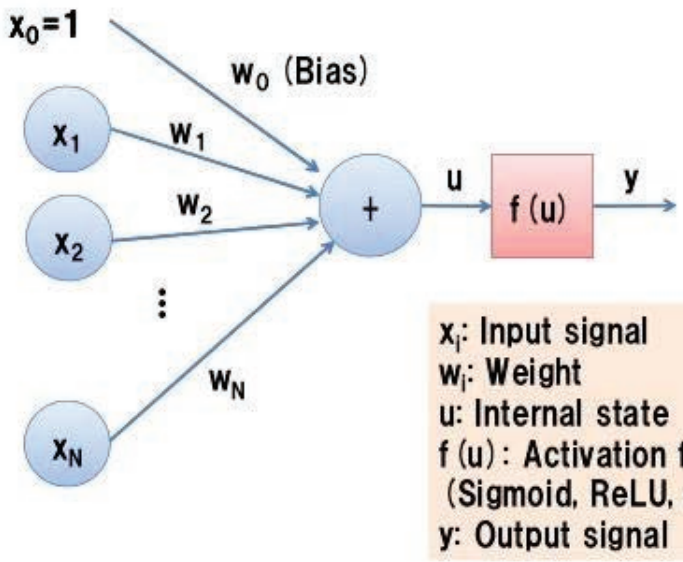
Satisfiability Modulo Theory



Back to Neural networks!

Recall

Artificial Neuron

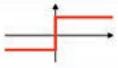
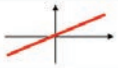
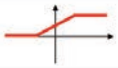
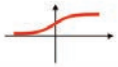


$$y = f(u)$$

$$u = \sum_{i=0}^{N} w_i x_i$$

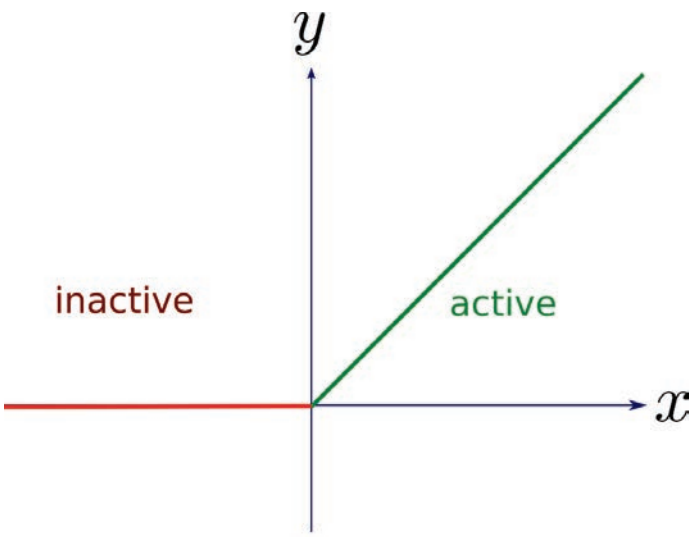
x_i: Input signal
w_i: Weight
u: Internal state
f (u): Activation function
(Sigmoid, ReLU, etc.)
y: Output signal

Recall

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	

Copyright © Sebastian Raschka 2016 (http://sebastianraschka.com)

Recall



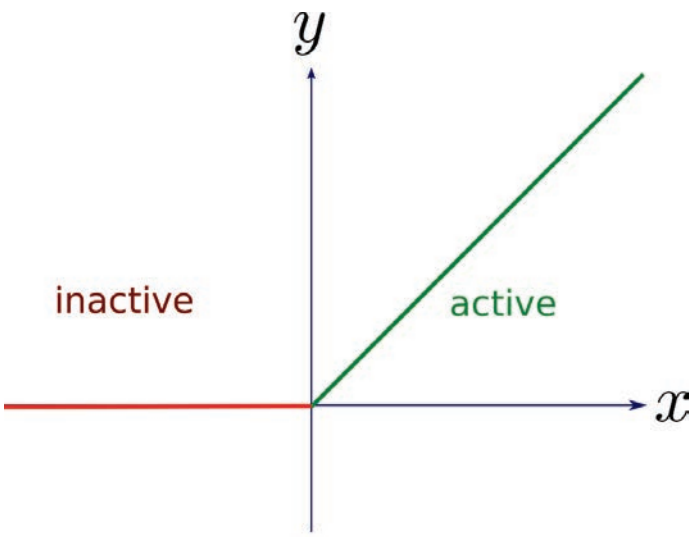
$$ReLU : x \rightarrow max(x, 0)$$

SMT

We know how to encode an ite already:
y = If (x < 0) then 0 else x
Becomes
(x < 0) => (y = 0)
(x >= 0) => (y = x)
Which becomes
Not (x<0) or y = 0
Not (x>=0) or y = x

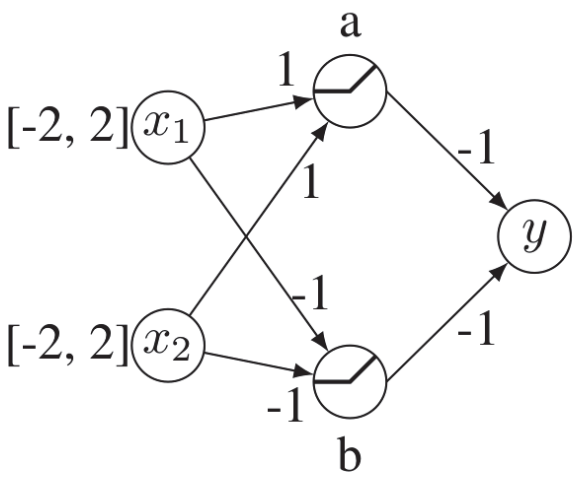
We now that the relu is just a max, which is just an ite.
So let's speak simpler

Recall



$$ReLU : x \rightarrow max(x, 0)$$

Example

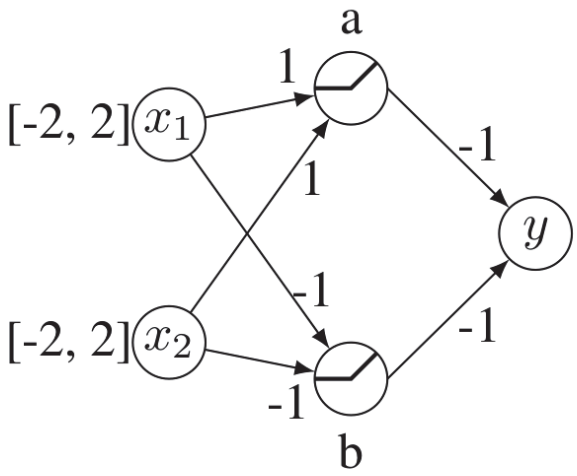


Prove that $y > -5$

Example

$$-2 \leq x_1 \leq 2$$
$$a_{\text{in}} = x_1 + x_2$$
$$a_{\text{out}} = \max(a_{\text{in}}, 0)$$

$$-2 \leq x_2 \leq 2 \qquad y \leq -5$$
$$b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$
$$b_{\text{out}} = \max(b_{\text{in}}, 0)$$



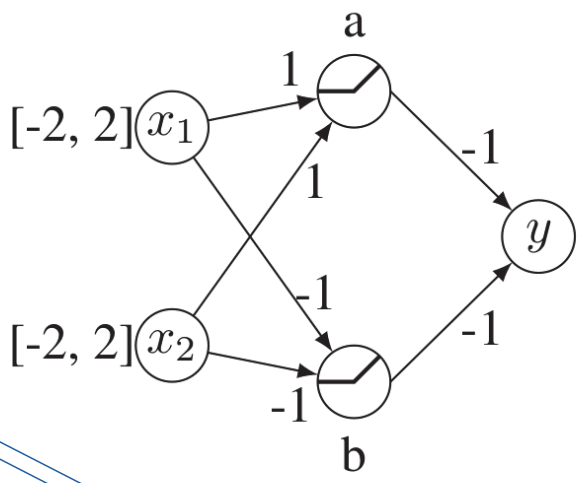
Prove that $y > -5$

Piecewise Linear Neural Network verification: A comparative study (2017, Bunel, Turkaslan, Torr, Kohli)

Example

$$-2 \leq x_1 \leq 2$$
$$a_{\text{in}} = x_1 + x_2$$
$$a_{\text{out}} = \max(a_{\text{in}}, 0)$$

$$-2 \leq x_2 \leq 2 \qquad y \leq -5$$
$$b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$
$$b_{\text{out}} = \max(b_{\text{in}}, 0)$$



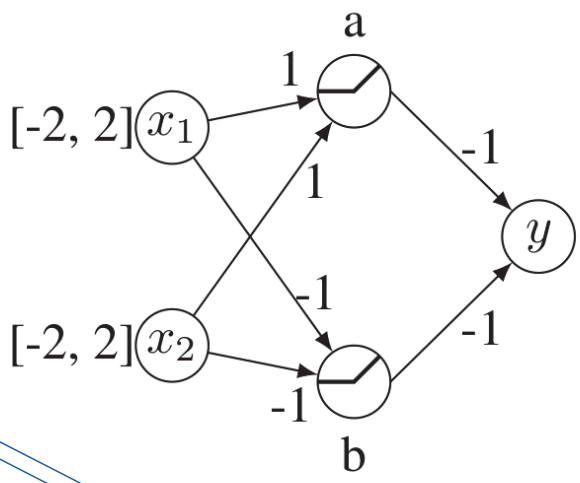
Prove that $y > -5$

Piecewise Linear Neural Network verification: A comparative study (2017, Bunel, Turkaslan, Torr, Kohli)

SURPRISE EXAM: Does this property hold?

$$-2 \leq x_1 \leq 2$$
$$a_{\text{in}} = x_1 + x_2$$
$$a_{\text{out}} = \max(a_{\text{in}}, 0)$$

$$-2 \leq x_2 \leq 2 \qquad y \leq -5$$
$$b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$
$$b_{\text{out}} = \max(b_{\text{in}}, 0)$$



Prove that $y > -5$

Piecewise Linear Neural Network verification: A comparative study (2017, Bunel, Turkaslan, Torr, Kohli)

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
```

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))
```

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))
```

```
(assert (<= (- 2) x1))
(assert (<= (- 2) x2))
(assert (<= x1 2))
(assert (<= x2 2))
```

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)          (assert (<= (- 2) x1))
(declare-fun x2 () Int)          (assert (<= (- 2) x2))
(declare-fun ain () Int)         (assert (<= x1 2))
(declare-fun aout () Int)        (assert (<= x2 2))
(declare-fun bin () Int)
(declare-fun bout () Int)        (assert (= ain (+ x1 x2)))
(declare-fun y () Int)           (assert (= bin (- x1 (- x2))))
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))
```

SMTLib standard

$-2 \leq x_1 \leq 2$ $\quad -2 \leq x_2 \leq 2$ $\quad y \leq -5$

$a_{\text{in}} = x_1 + x_2$ $\quad b_{\text{in}} = -x_1 - x_2$ $\quad y = -a_{\text{out}} - b_{\text{out}}$

$a_{\text{out}} = \max(a_{\text{in}}, 0)$ $\quad b_{\text{out}} = \max(b_{\text{in}}, 0)$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))
```

```
(assert (<= (- 2) x1))
(assert (<= (- 2) x2))
(assert (<= x1 2))
(assert (<= x2 2))

(assert (= ain (+ x1 x2)))
(assert (= bin (- x1 (- x2))))
```

```
(assert (= aout (reluR ain)))
(assert (= bout (reluR bin)))
```

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))

(assert (<= (- 2) x1))
(assert (<= (- 2) x2))
(assert (<= x1 2))
(assert (<= x2 2))

(assert (= ain (+ x1 x2)))
(assert (= bin (- x1 (- x2))))

(assert (= aout (reluR ain)))
(assert (= bout (reluR bin)))

(assert (= y (- aout (- bout))))
```

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))

(assert (<= (- 2) x1))
(assert (<= (- 2) x2))
(assert (<= x1 2))
(assert (<= x2 2))

(assert (= ain (+ x1 x2)))
(assert (= bin (- x1 (- x2))))

(assert (= aout (reluR ain)))
(assert (= bout (reluR bin)))

(assert (= y (- aout (- bout))))

(assert (<= y -5))
```

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))

(assert (<= (- 2) x1))
(assert (<= (- 2) x2))
(assert (<= x1 2))
(assert (<= x2 2))

(assert (= ain (+ x1 x2)))
(assert (= bin (- x1 (- x2))))

(assert (= aout (reluR ain)))
(assert (= bout (reluR bin)))

(assert (= y (- aout (- bout))))

(assert (<= y -5))
(check-sat)
```

SMTLib standard

Tedious isn't it...

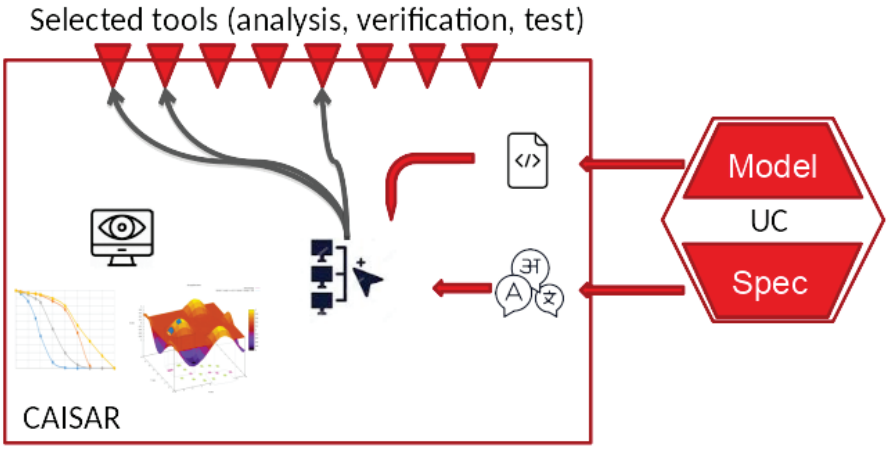


CAISAR

Characterizing Artificial Intelligence Safety and Reliability

A federative platform for analysis of artificial intelligence system components

CAISAR platform



INPUT CHARACTERIZATION

PERTURBATIONS ABSTRACTION
RANGES ANALYSIS
PERFORMANCE-AWARE

Available paper: https://hal.archives-ouvertes.fr/hal-03687211/
Code: https://git.frama-c.com/pub/caisar

CAISAR platform

1. Property specification in WhyML
2. **Automatic** translation into the PyRAT and Marabou respective specification languages
3. **Automatic** launch of verification with PyRAT and Marabou
4. **Automatic** interpretation, and post-processing if needed, of (sub-)results

```
theory T
  use NN.AsTuple
  use ieee_float.Float64

  goal G: forall x0 x1.
    -0.328423 <= x0 <= 0.679857 /\
    -0.5 <= x1 <= 0.375 ->
    let (y0,y1,y2) = net_apply x1 x2 in
    y0 <= y2 \/ y1 <= y2
end
```

SMTLib standard

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2 \qquad y \leq -5$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

```
(declare-fun x1 () Int)
(declare-fun x2 () Int)
(declare-fun ain () Int)
(declare-fun aout () Int)
(declare-fun bin () Int)
(declare-fun bout () Int)
(declare-fun y () Int)
(define-fun relu ((res Int )) Int (ite (> res 0 ) res 0 ))

(assert (<= (- 2) x1))
(assert (<= (- 2) x2))
(assert (<= x1 2))
(assert (<= x2 2))

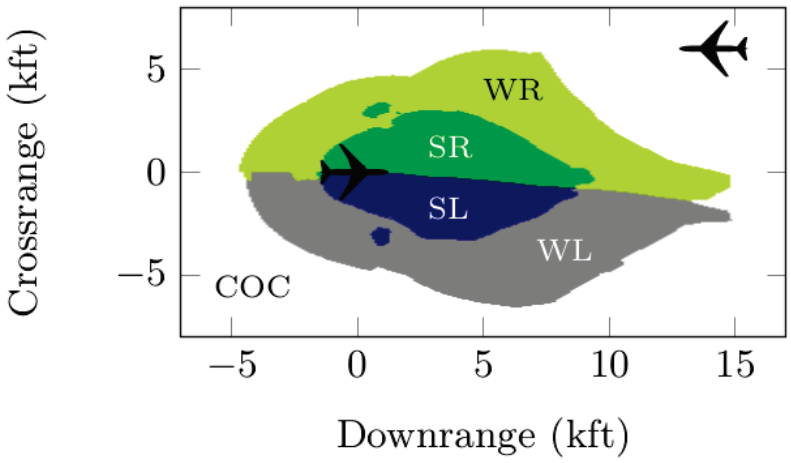
(assert (= ain (+ x1 x2)))
(assert (= bin (- x1 (- x2))))

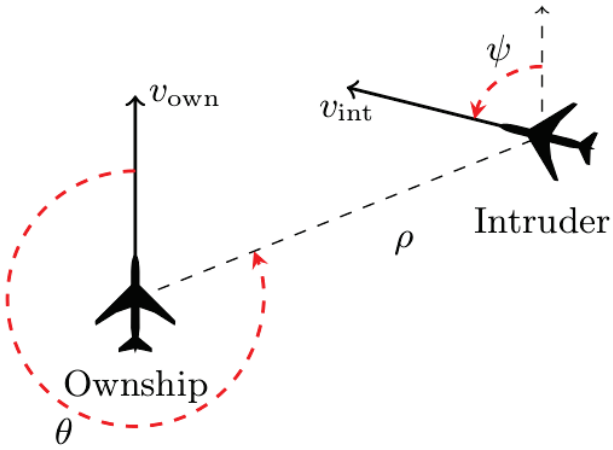
(assert (= aout (reluR ain)))
(assert (= bout (reluR bin)))

(assert (= y (- aout (- bout))))

(assert (<= y -5))
(check-sat)
```

Reluplex


Crossrange (kft)
Downrange (kft)
WR
SR
SL
WL
COC
5
0
−5
−5
0
5
10
15


v_{own}
v_{int}
ψ
ρ
θ
Ownship
Intruder

Reluplex

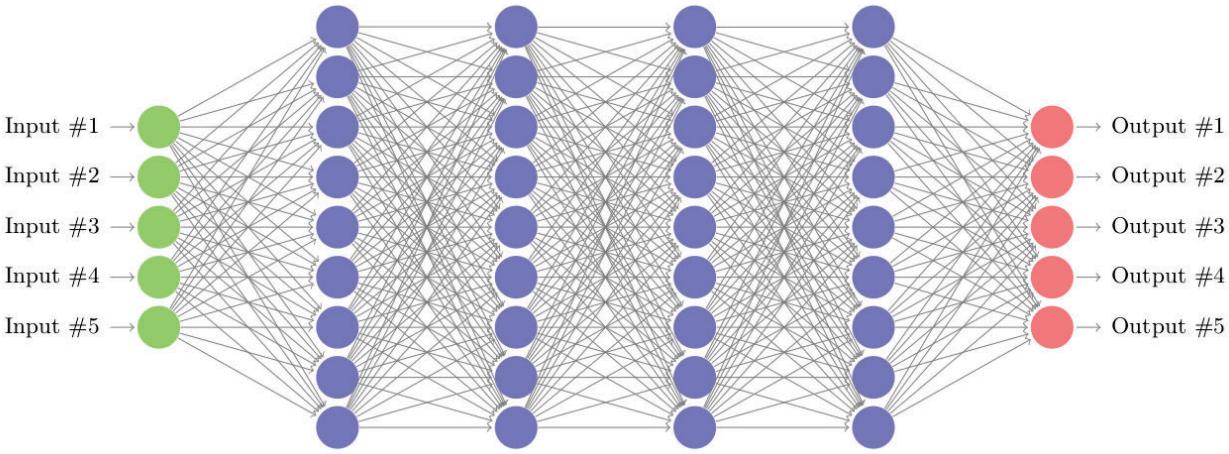
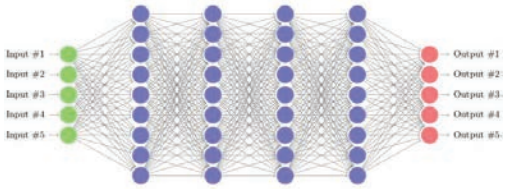


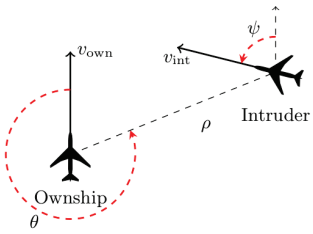
Fig. 1: A fully connected DNN with 5 input nodes (in green), 5 output nodes (in red), and 4 hidden layers containing a total of 36 hidden nodes (in blue).

Reluplex

; Input nodes
; x1 ρ
; x2 θ
; x3 ψ
; x4 vown
; x5 v int
; Output nodes
; y1 coc
; y2 weak right
; y3 strong right
; y4 weak left
; y5 strong left

Network Functionality. The ACAS Xu system maps input variables to action advisories. Each advisory is assigned a score, with the lowest score corresponding to the best action. The input state is composed of seven dimensions (shown in Fig. 6) which represent information determined from sensor measurements 19: (i) ρ: Distance from ownship to intruder; (ii) θ: Angle to intruder relative to ownship heading direction; (iii) ψ: Heading angle of intruder relative to ownship heading direction; (iv) v_{own}: Speed of ownship; (v) v_{int}: Speed of intruder; (vi) τ: Time until loss of vertical separation; and (vii) a_{prev}: Previous advisory. There are five outputs which represent the different horizontal advisories that can be given to the ownship: Clear-of-Conflict (COC), weak right, strong right, weak left, or strong left. Weak and strong mean heading rates of 1.5 °/s and 3.0 °/s, respectively.





Reluplex

Property ϕ_1.

- Description: If the intruder is distant and is significantly slower than the ownship, the score of a COC advisory will always be below a certain fixed threshold.
- Tested on: all 45 networks.
- Input constraints: $\rho \geq 55947.691$, $v_{\text{own}} \geq 1145$, $v_{\text{int}} \leq 60$.
- Desired output property: the score for COC is at most 1500.

Reluplex

Property ϕ_1.

- Description: If the intruder is distant and is significantly slower than the ownship, the score of a COC advisory will always be below a certain fixed threshold.
- Tested on: all 45 networks.
- Input constraints: $\rho \geq 55947.691$, $v_{\text{own}} \geq 1145$, $v_{\text{int}} \leq 60$.
- Desired output property: the score for COC is at most 1500.

```
(assert (>= |X_0_0| 55947.691))
(assert (>= |X_3_0| 1145))
(assert (<= |X_4_0| 60))
```

```
(assert (>= |y_out_0_0| 1500))
(check-sat)
```

Reluplex

But you also need to encode the NN

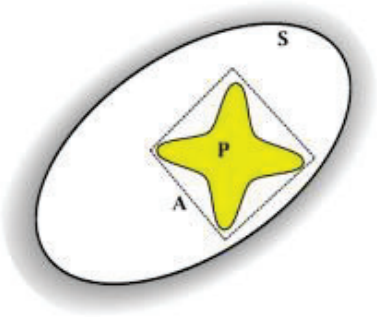
And that can get very big...

Abstract Interpretation

Conservative overapproximation to encapsulate the real behaviour

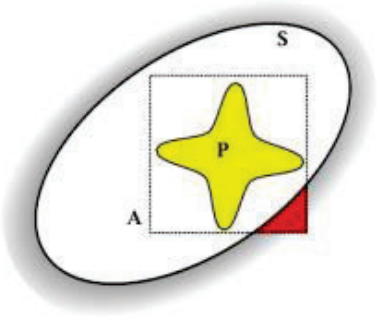


Abstract Interpretation



precise analysis

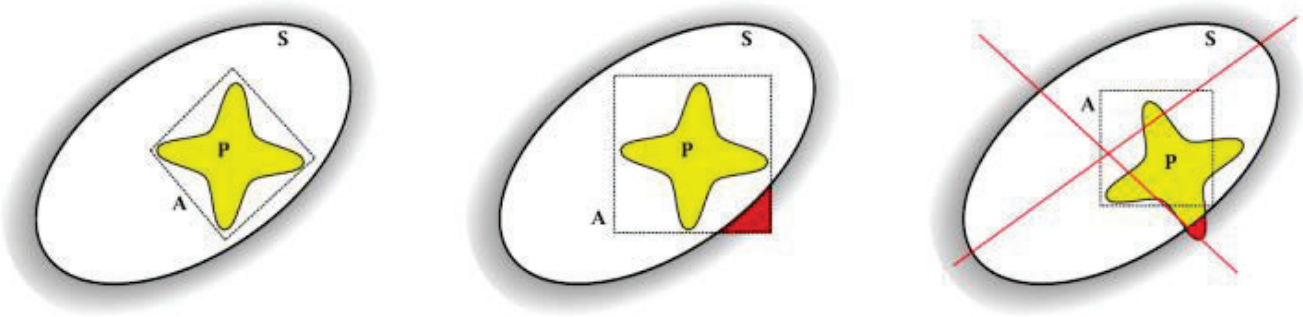
$A \subseteq S \implies P \subseteq S$



false alarm

$A \not\subseteq S$ but $P \subseteq S$

Abstract Interpretation



precise analysis
$A \subseteq S \implies P \subseteq S$

false alarm
$A \not\subseteq S$ but $P \subseteq S$

unsound analysis
$A \subseteq S$ but $P \not\subseteq S$

Abstract Interpretation

We would like to verify a property on the all possible values of inputs $x \in [a,b]$ and $y \in [c,d]$ in some program.

e.g.:

$$x + y \in [a+c, b+d]$$

Do the same for all operations in the program.

Use other types of domain for more precision (not just intervals).

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```

Property: "f(y)=100 → Critical vibration frequency "

Concret

```
f (int y){
 int x;                    ..... ..-2,-1,0,1,2,..
 x = 3 * (y²+1);           ..... 3,6,9,..
 if x > 100 then           ..... 102,105,108,..
    x = x + 10;            ..... 112,115,118,..
 else                      ..... 3,6,9..93,96,99
    x = x - 2;             ..... 1,4,7,..91,94,97
 return x;                 ..... 1,4,..94,97,112,115..
}
```

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;                  .....
 x = 3 * (y²+1);         .....
 if x > 100 then         .....
    x = x + 10;          .....
 else                    .....
    x = x - 2;           .....
 return x;               .....
}
```

	Concret	Intervals
.....	..-2,-1,0,1,2,..	-∞,+∞
.....	3,6,9,..	3,+∞
.....	102,105,108,..	102,+∞
.....	**112,115,118**,..	**112,+∞**
.....	3,6,9..93,96,99	3,99
.....	**1,4,7,..91,94,97**	**1,97**
.....	1,4,..94,97,112,115..	1,+∞

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;                  .....
 x = 3 * (y²+1);         .....
 if x > 100 then         .....
    x = x + 10;          .....
 else                    .....
    x = x - 2;           .....
 return x;               .....
}
```

	Concret	Intervals
.....	..-2,-1,0,1,2,..	-∞,+∞
.....	3,6,9,..	3,+∞
.....	102,105,108,..	102,+∞
.....	**112,115,118**,..	**112,+∞**
.....	3,6,9..93,96,99	3,99
.....	**1,4,7,..91,94,97**	**1,97**
.....	1,4,..94,97,112,115..	1,+∞

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;                  .....
 x = 3 * (y²+1);         .....
 if x > 100 then         .....
    x = x + 10;          .....
 else                    .....
    x = x - 2;           .....
 return x;               .....
}
```

	Concret	Intervals	modulo
.....	..-2,-1,0,1,2,..	-∞,+∞	0%1
.....	3,6,9,..	3,+∞	0%3
.....	102,105,108,..	102,+∞	0%3
.....	**112,115,118**,..	**112,+∞**	1%3
.....	3,6,9..93,96,99	3,99	0%3
.....	**1,4,7,..91,94,97**	**1,97**	1%3
.....	1,4,..94,97,112,115..	1,+∞	1%3

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;                  .....
 x = 3 * (y²+1);         .....
 if x > 100 then         .....
    x = x + 10;          .....
 else                    .....
    x = x - 2;           .....
 return x;               .....
}
```

Concret	Intervals	modulo
..-2,-1,0,1,2,..	-∞,+∞	0%1
3,6,9,..	3,+∞	0%3
102,105,108,..	102,+∞	0%3
112,115,118,..	**112,+∞**	1%3
3,6,9..93,96,99	3,99	0%3
1,4,7,..91,94,97	**1,97**	1%3
1,4,..94,97,112,115..	1,+∞	1%3

Property: "f(y)=100 → Critical vibration frequency "

		Concret	Intervals	modulo
f (int y){				
int x;	.....	..-2,-1,0,1,2,..	-∞,+∞	0%1
x = 3 * (y^2+1);	.....	3,6,9,..	3,+∞	0%3
if x > 100 then	.....	102,105,108,..	102,+∞	0%3
x = x + 10;	.....	**112,115,118**,..	**112,+∞**	1%3
else	.....	3,6,9..93,96,99	3,99	0%3
x = x - 2;	.....	**1,4,7,..91,94,97**	**1,97**	1%3
return x;	.....	1,4,..94,97,112,115..	1,+∞	1%3
}				

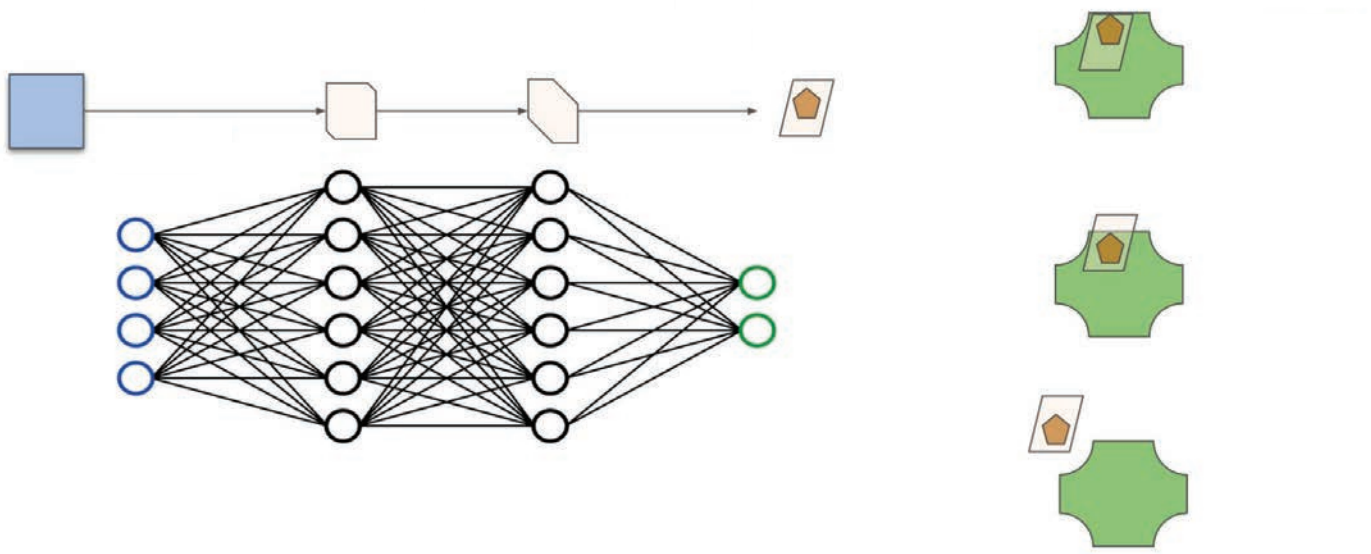
Conservative over-approximation: the concretisation of the abstract domains contains reality
The inverse is not necessarily true. 1,4..94,97,**100,103,106,109**,112,115..

Property: "f(y)=100 → Critical vibration frequency "

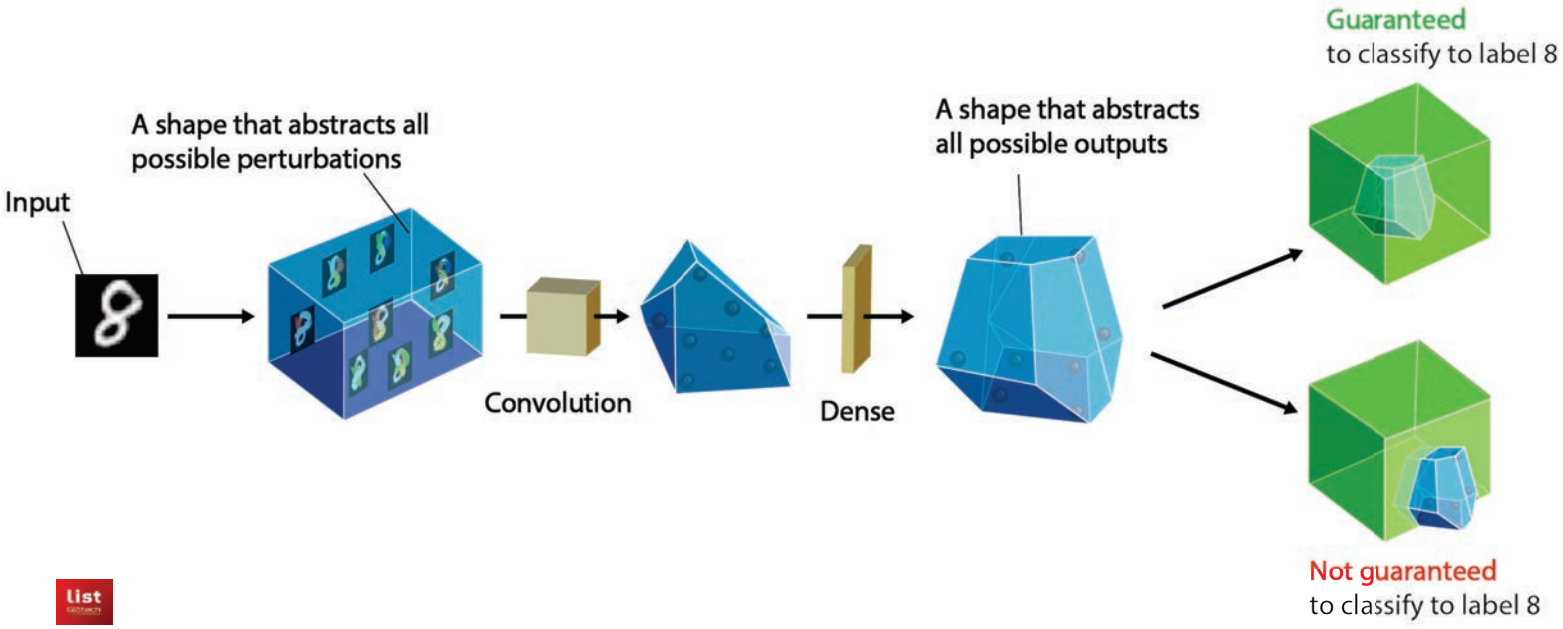
		Concret	Intervals	modulo	Union of intervals
f (int y){					
int x;	.....	..-2,-1,0,1,2,..	-∞,+∞	0%1	-∞,+∞
x = 3 * (y^2+1);	.....	3,6,9,..	3,+∞	0%3	3,+∞
if x > 100 then	.....	102,105,108,..	102,+∞	0%3	102,+∞
x = x + 10;	.....	**112,115,118**,..	**112,+∞**	1%3	**112,+∞**
else	.....	3,6,9..93,96,99	3,99	0%3	3,99
x = x - 2;	.....	**1,4,7,..91,94,97**	**1,97**	1%3	**1,97**
return x;	.....	1,4,..94,97,112,115..	1,+∞	1%3	*[1,97]∪ [112,+∞[*
}					

Conservative over-approximation: the concretisation of the abstract domains contains reality
The inverse is not necessarily true. 1,4..94,97,**100,103,106,109**,112,115..

Application to NN



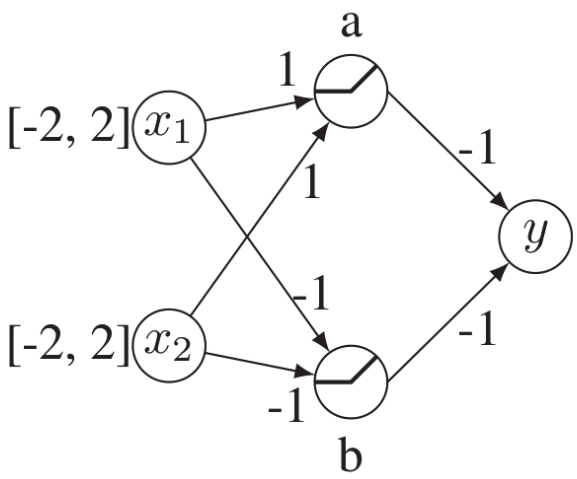
Application to NN



Abstract Interpretation

$$-2 \leq x_1 \leq 2$$
$$a_{\text{in}} = x_1 + x_2$$
$$a_{\text{out}} = \max(a_{\text{in}}, 0)$$

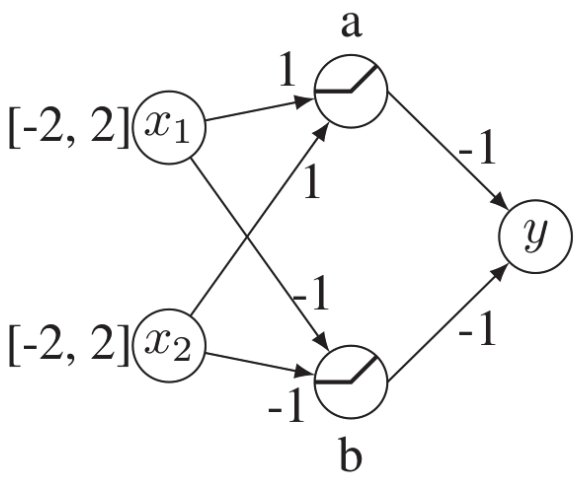
$$-2 \leq x_2 \leq 2 \qquad y \leq -5$$
$$b_{\text{in}} = -x_1 - x_2 \qquad y = -a_{\text{out}} - b_{\text{out}}$$
$$b_{\text{out}} = \max(b_{\text{in}}, 0)$$



Prove that $y > -5$

Abstract Interpretation

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$
$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$
$$y = -a_{\text{out}} - b_{\text{out}}$$



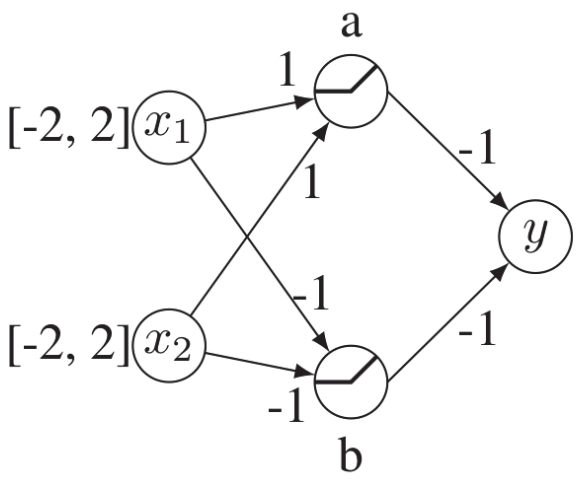
Prove that $y > -5$

Abstract Interpretation

$a_{\text{in}} = x_1 + x_2$ $\quad$ $b_{\text{in}} = -x_1 - x_2$

$a_{\text{in}} =$

$b_{\text{in}} =$



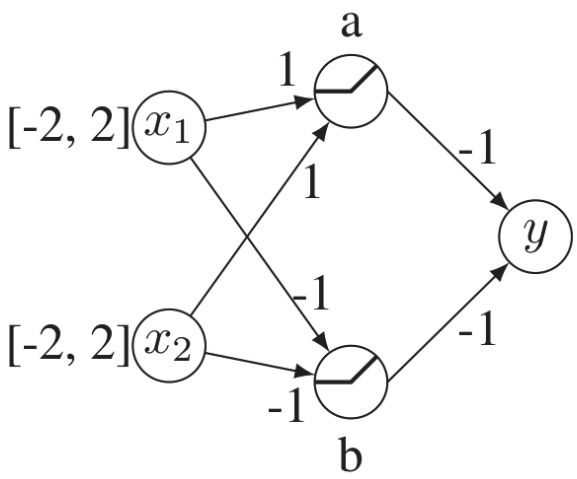
Prove that $y > -5$

Abstract Interpretation

$a_{\text{in}} = x_1 + x_2$ $\quad$ $b_{\text{in}} = -x_1 - x_2$

$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$



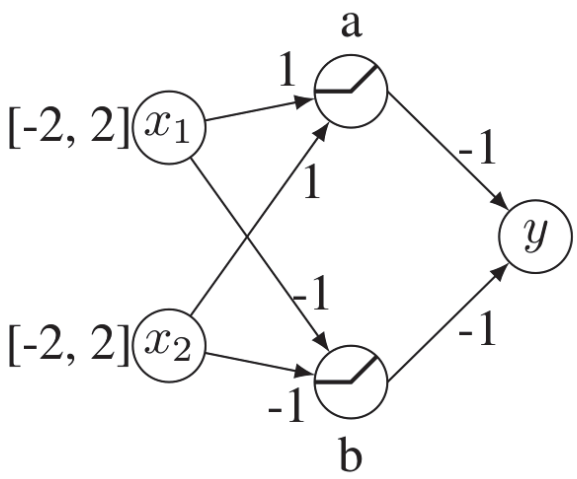
Prove that $y > -5$

Abstract Interpretation

$a_{\text{in}} = x_1 + x_2$ $\quad$ $b_{\text{in}} = -x_1 - x_2$

$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]



Prove that $y > -5$

Abstract Interpretation

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

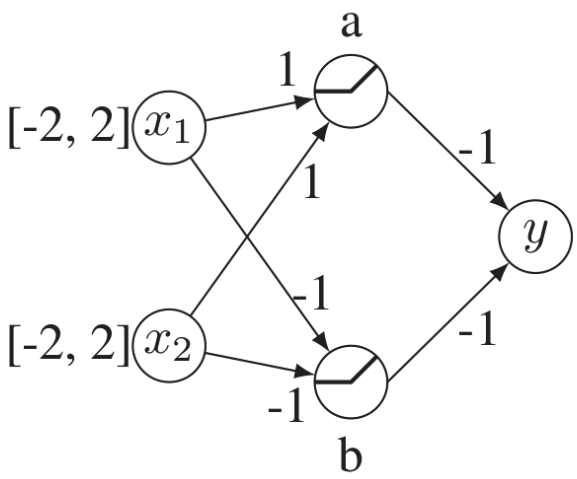
$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$

$b_{\text{out}} =$

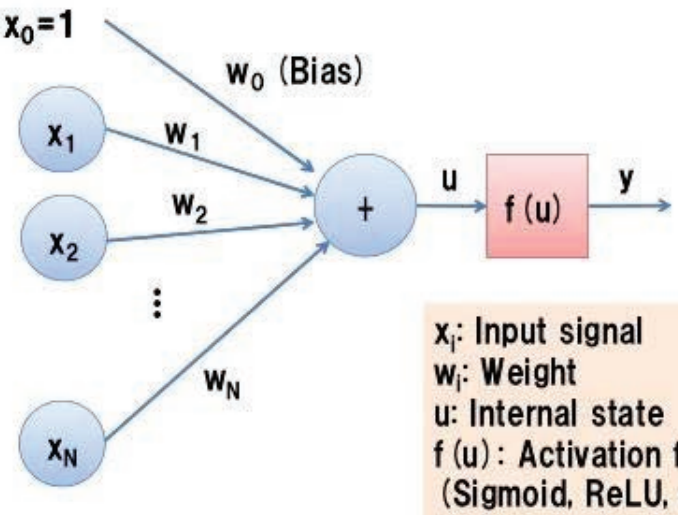
$y =$



Prove that $y > -5$

Recall

Artificial Neuron



$$y = f(u)$$

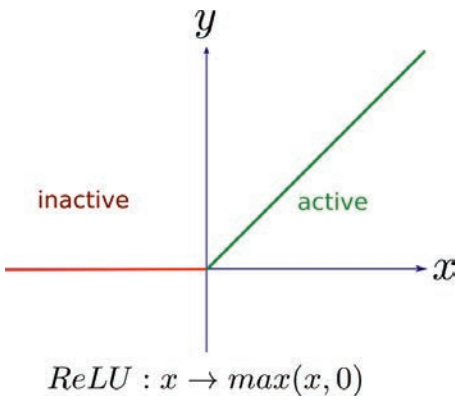
$$u = \sum_{i=0}^{N} w_i x_i$$

x_i: Input signal
w_i: Weight
u: Internal state
f (u): Activation function
(Sigmoid, ReLU, etc.)
y: Output signal

Recall

$$a_{\text{out}} = \max(a_{\text{in}}, 0)$$

	a_{in}	a_{out}
$a_{\text{in}} = x_1 + x_2$	[-4,4]	
if $a_{\text{in}} > 0$ then	]0,4]	
$a_{\text{out}} = a_{\text{in}}$		]0,4]
else	[-4,0]	
$a_{\text{out}} = 0$		[0,0]
		[0,4]



Abstract Interpretation

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

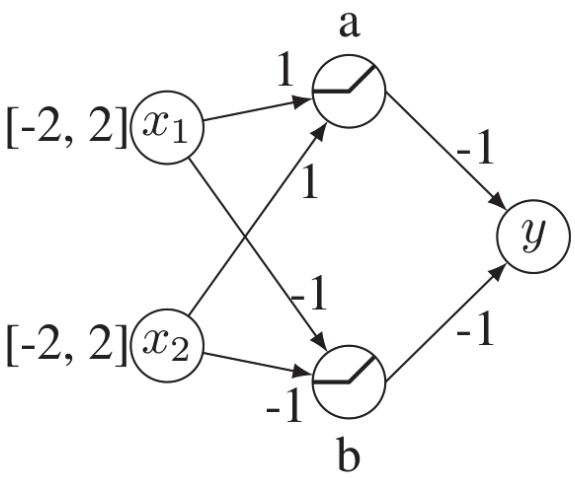
$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$

$b_{\text{out}} =$

$y =$





Prove that $y > -5$

Abstract Interpretation

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

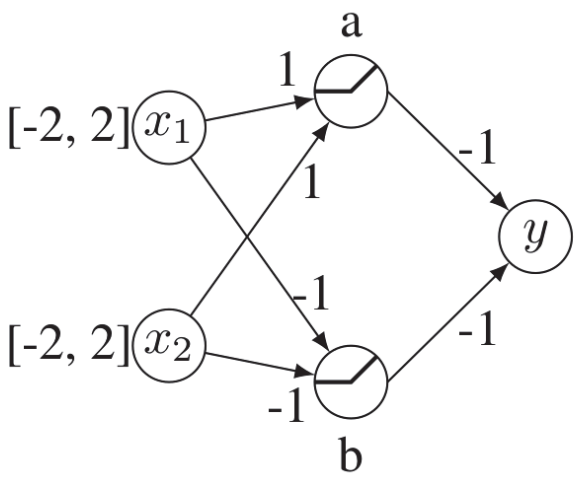
$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$ [0,4]

$b_{\text{out}} =$ [0,4]

$y =$



Prove that $y > -5$

Abstract Interpretation

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

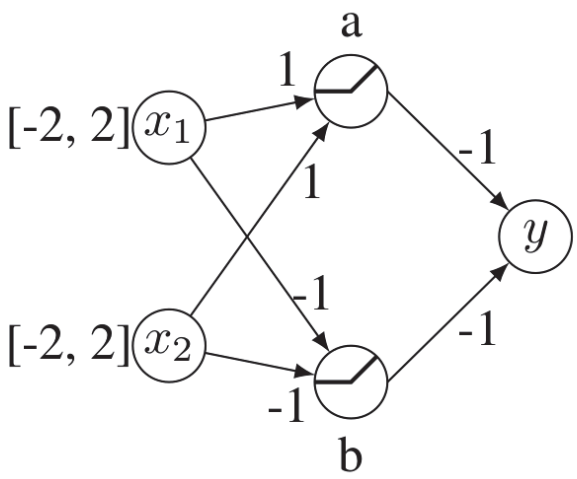
$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$ [0,4]

$b_{\text{out}} =$ [0,4]

$y =$ [-8,0]



Prove that $y > -5$

Abstract Interpretation

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

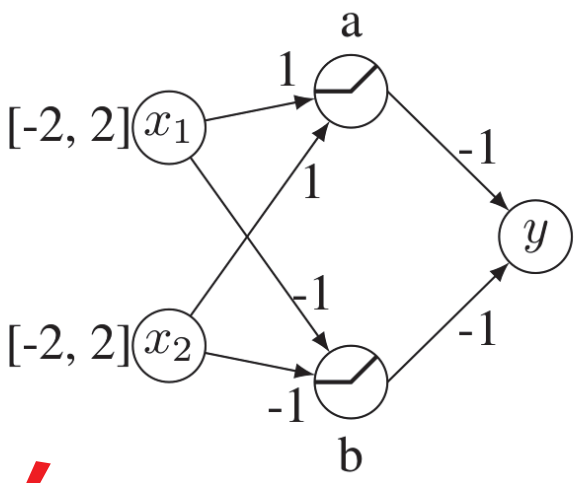
a_{in} = [-2,2] + [-2,2] = [-4,4]

b_{in} = - [-2,2] - [-2,2] =[-4,4]

a_{out} = [0,4]

b_{out} = [0,4]

y = [-8,0]

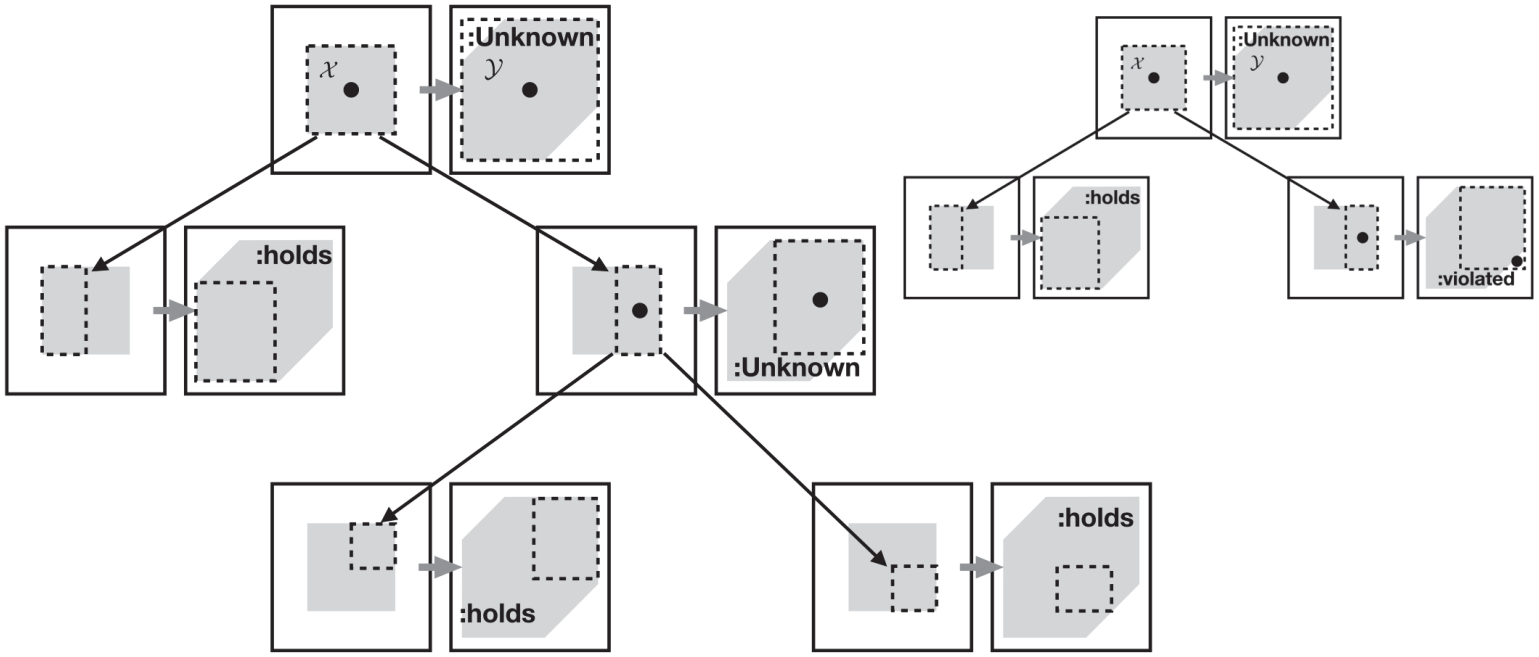


Prove that $y > -5$

Abstract Interpretation

A lot more clever stuff (eg, to handle loops)
Other domains (e.g., zonotopes and deltas, which are *relationals,* ie keep relations between variables…)
Splitting heuristics…
...

Abstract Interpretation



Abstract Interpretation

2018 IEEE Symposium on Security and Privacy

AI^2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri*, Martin Vechev

Differentiable Abstract Interpretation for Provably Robust Neural Networks

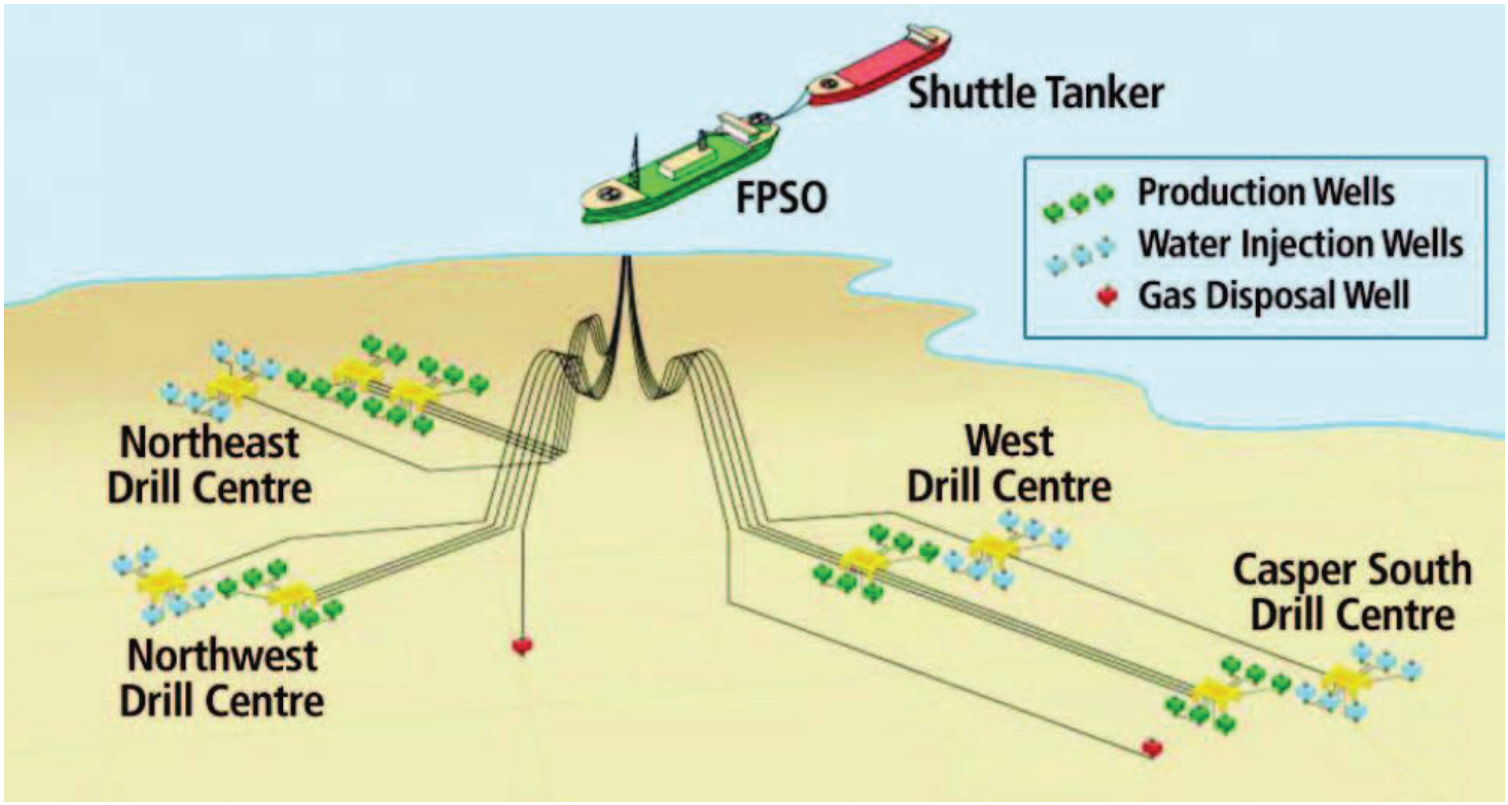
Matthew Mirman [1] **Timon Gehr** [1] **Martin Vechev** [1]

Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation

Antoine Miné
Sorbonne Universités, UPMC Univ. Paris 06, CNRS, LIP6

Real world example

Mooring line failures

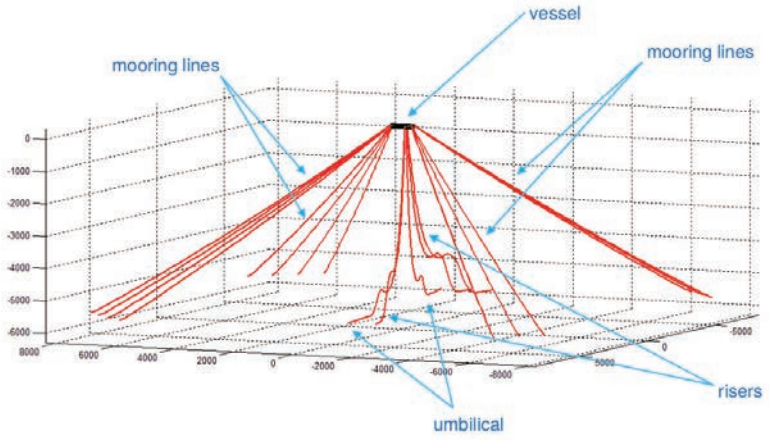


CONFIDENTIAL Technip Energies/ CEA

The problem

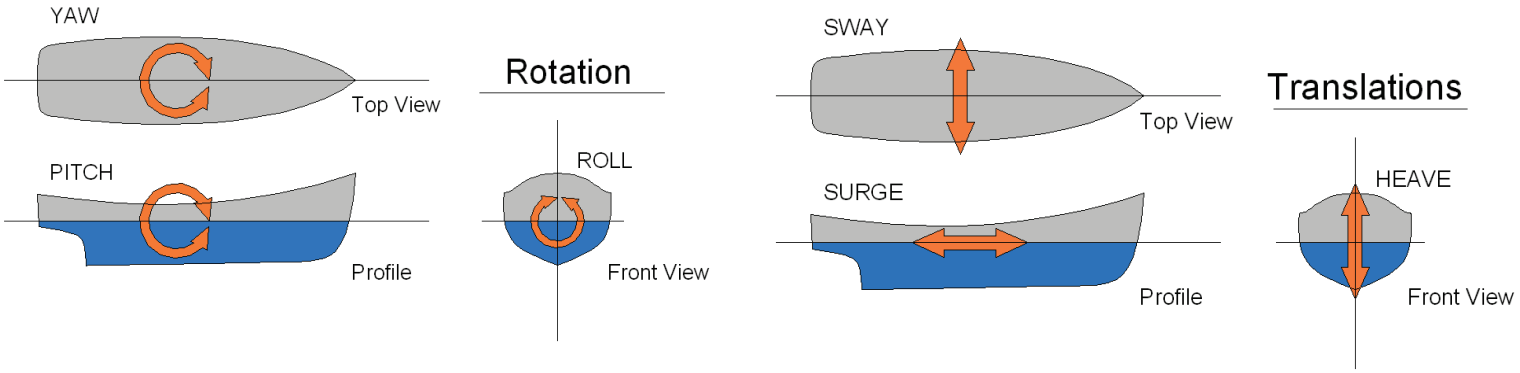
Mooring incidents (DeepStar® data from 1997-2012):

- 107 incidents from 73 facilities across the industry
- Potentially dire consequences
- Many FPSO have no means of monitoring lines (78% of North Sea units)
- Those who do face technical problems (robustness of equipment)



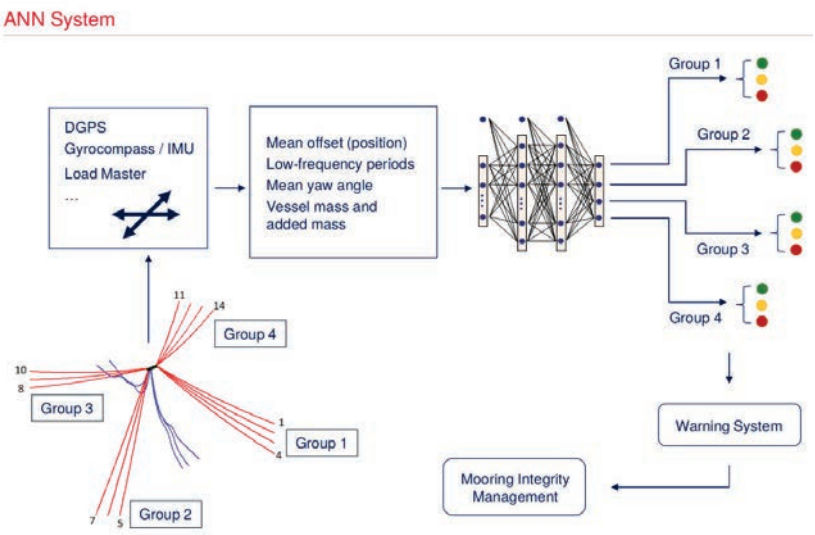
Their idea

Patented dry monitoring detection systems, based on vessel positions and low-frequency periods (which can be obtained from Dual GPS)



The challenges

- Highly non-linear problem (machine learning to recognize and classify patterns)
- Ability to deal with some degrees of variations from various system components (such as mooring line stiffness) and with error or noise from monitoring system
- Cover a complete range of vessel drafts, expected vessel responses from environment conditions and directions and mooring line conditions



The model

- Input: Vessel movement, mass, offset, …
- Output: group-line failures

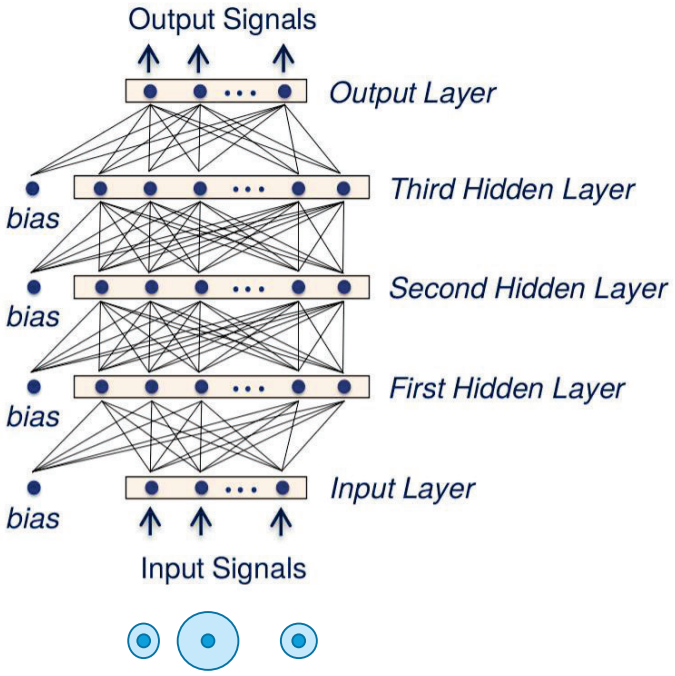
The work

Ensuring robustness properties

- Stability of classification in presence of perturbation
- Domain agnostic
- Perturbation per input
- Different perturbations for different inputs

Results

- Frama-C/Eva: Based on Abstract Interpretation, heavy machinery for C, slow
- FLDLib: Better results, better execution times
- PyRAT: Tailor-made for NN, yet better results and execution time, no encoding





Python Reachability Analysis Tool

A formal verification tool to analyse the robustness and safety of neural network

Symbolic Execution



Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.

Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.
Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```



Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```

y = **a**

Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```

y = **a**

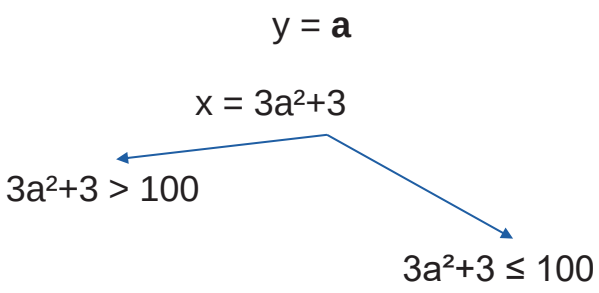
$x = 3a^2+3$

Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```

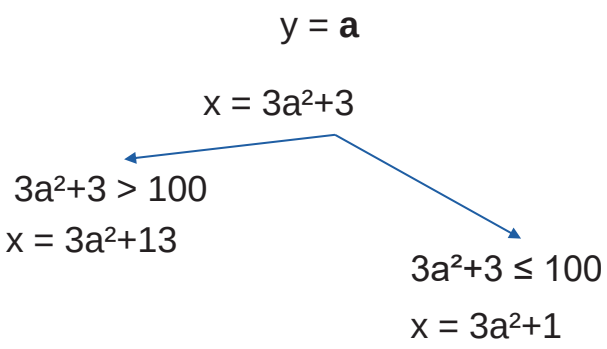


Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```

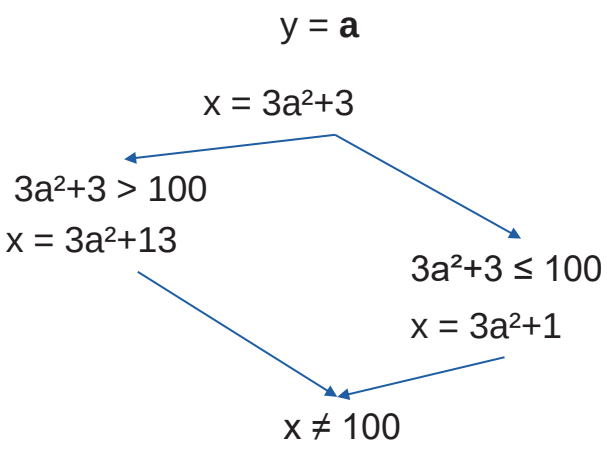


Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```

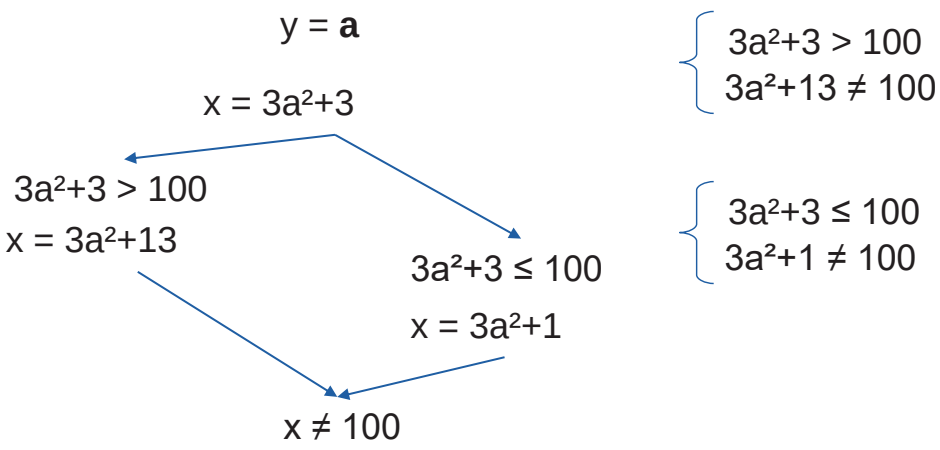


Symbolic Execution

Replace program entries by symbolic values, express all the program in terms of these values, accumulate path conditions along the symbolic execution and verify properties on these paths.

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;
 x = 3 * (y²+1);
 if x > 100 then
    x = x + 10;
 else
    x = x - 2;
 return x;
}
```



Symbolic Execution

Symbolic Execution for Deep Neural Networks

Divya Gopinath[1], Kaiyuan Wang[2], Mengshi Zhang[2], Corina S. Păsăreanu[1], Sarfraz Khurshid[2]

Symbolic Execution for Importance Analysis and Adversarial Generation in Neural Networks

Divya Gopinath*, Mengshi Zhang†, Kaiyuan Wang†, İsmet Burak Kadron‡, Corina S. Păsăreanu*, Sarfraz Khurshid

Symbolic Execution for Attribution and Attack Synthesis in Neural Networks

Corina S Pasareanu, Divya Gopinath, Sarfraz Khurshid, Kaiyuan Wang, Mengshi Zhang



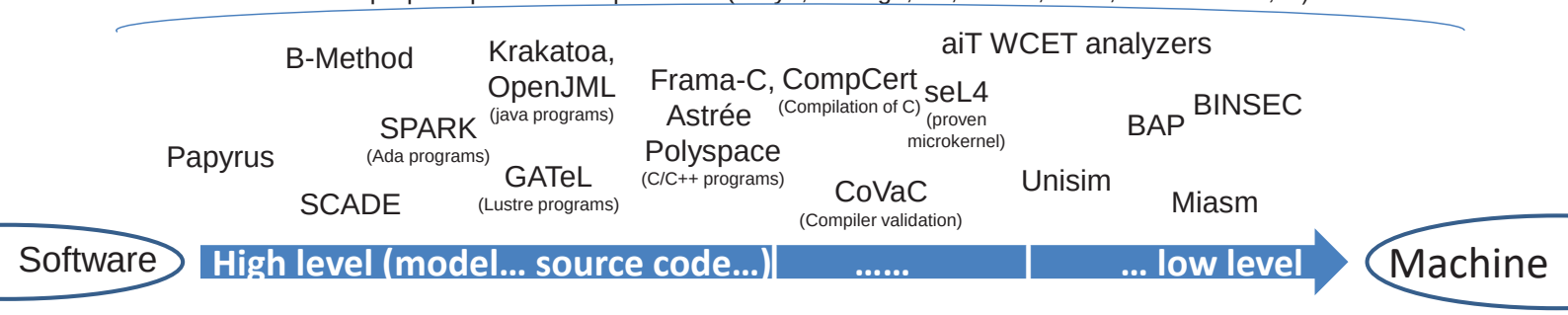
Algorithms for Verifying Deep Neural Networks

Changliu Liu
Carnegie Mellon University, Pittsburgh, PA 15213

Tomer Arnon
Christopher Lazarus
Clark Barrett
Mykel J. Kochenderfer
Stanford University, Stanford, CA 94305

Not the complete picture…

"Traditional" human-written software



Machine generated models

178

Not the complete picture…

"Traditional" human-written software

General purpose provers and platforms (Why3, Alt-Ergo, Z3, Colibri, TLA$^+$, K-framework,…)

B-Method

Krakatoa, OpenJML (java programs)

Frama-C, CompCert (Compilation of C)

aiT WCET analyzers

seL4 (proven microkernel)

BINSEC

BAP

SPARK (Ada programs)

Astrée

Polyspace (C/C++ programs)

Papyrus

GATeL (Lustre programs)

CoVaC (Compiler validation)

Unisim

Miasm

SCADE

Software

High level (model… source code…)

……

… low level

Machine

CNN-Cert

Planet

ReluVal

Neurify

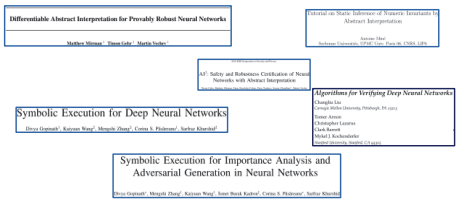
Eran

PyRAT

SAVer

Sherlock

MIPVerify



Machine generated models

179

Not the complete picture…

"Traditional" human-written software

General purpose provers and platforms (Why3, Alt-Ergo, Z3, Colibri, TLA$^+$, K-framework,…)

B-Method

Krakatoa, OpenJML (java programs)

Frama-C, CompCert (Compilation of C)

aiT WCET analyzers

seL4 (proven microkernel)

BINSEC

BAP

SPARK (Ada programs)

Astrée

Polyspace (C/C++ programs)

Papyrus

GATeL (Lustre programs)

CoVaC (Compiler validation)

Unisim

Miasm

SCADE

Software → High level (model… source code…) …… … low level → Machine

CNN-Cert

Planet

ReluVal

Neurify

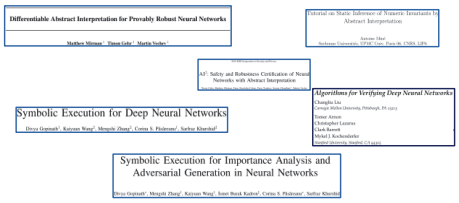
Eran

PyRAT

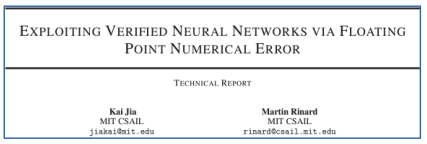
SAVer

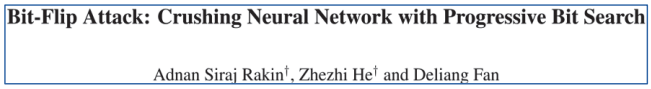
Sherlock

MIPVerify











Machine generated models

180

Not the complete picture…

"Traditional" human-written software

General purpose provers and platforms (Why3, Alt-Ergo, Z3, Colibri, TLA$^+$, K-framework,…)

B-Method

Krakatoa, OpenJML (java programs)

Frama-C, CompCert (Compilation of C)

aiT WCET analyzers

seL4 (proven microkernel)

BINSEC

BAP

SPARK (Ada programs)

Astrée

Polyspace (C/C++ programs)

Papyrus

GATeL (Lustre programs)

SCADE

CoVaC (Compiler validation)

Unisim

Miasm

Software → High level (model… source code…) …… … low level → Machine

Not NN

SAVer

Neurify

CNN-Cert

Planet

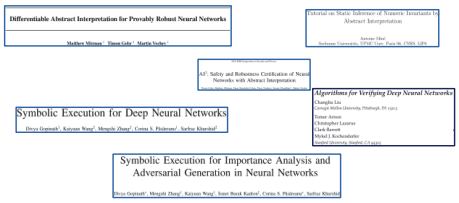
ReluVal

Eran

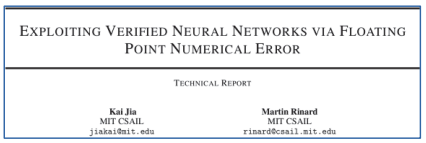
PyRAT

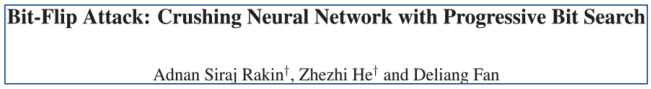
Sherlock

MIPVerify











Machine generated models

181

Challenges

- Scalability: clever heuristics? Parallelisation?
- Generality: Focus on relu networks… other activation functions? Recurrent networks? Transformers? Other models: boosting? Etc.
  - Need for more expressive languages to describe more specifications and tools able to handle them
- Coverage of all compilation steps until the embedded software.

- Community building: compare to the SMT-Comp, runtime verification comp, CADE ATP System Competition…
  - Common language, common benchmark, common infrastructure (VNN comp)
- Encourage formation: university masters courses, tutorial in conferences, summer schools…

182

Other FM possibilities: Colibri

Principle: Safe-by-design Symbolic AI through a constraint solving library

- Separately prove, in Why3, the necessary bricks for constraint solving: Floating-point numbers, integers, bit-vectors, strings, etc.
- Allow for selection of these bricks to tailor the construction of a solver to the needs of the user
- Automatically extract a C implementation of the solver

Target: XGBoost models, embedded software

Application: Energy sector (e.g., IRSN), space (e.g., NASA). Can also be used as a verification tool (winner of SMT-Competition since 2017), which makes it an essential brick of other tools such as Frama-C and GATeL.

Background: Constraint solving is used in several critical software domains



Colibri & co
Constraint Library
Colibri2
ColibriCS: Certified Solver

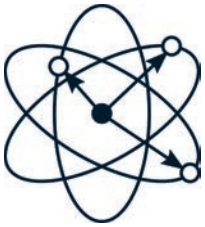
183

Formal Methods are not a panacea

Principle: Detect recurring parts in a dataset through unsupervised learning
- Pluggable: added to an existing backbone, fraction of the size
- Frugal: no need to fine-tune the backbone
- Non-invasive: minimal access to the backbone and a fraction of the data
- Fast: convergence in a few epochs
- Measured: gives confidence measures of the detections

(patent pending)
Target: Neural networks

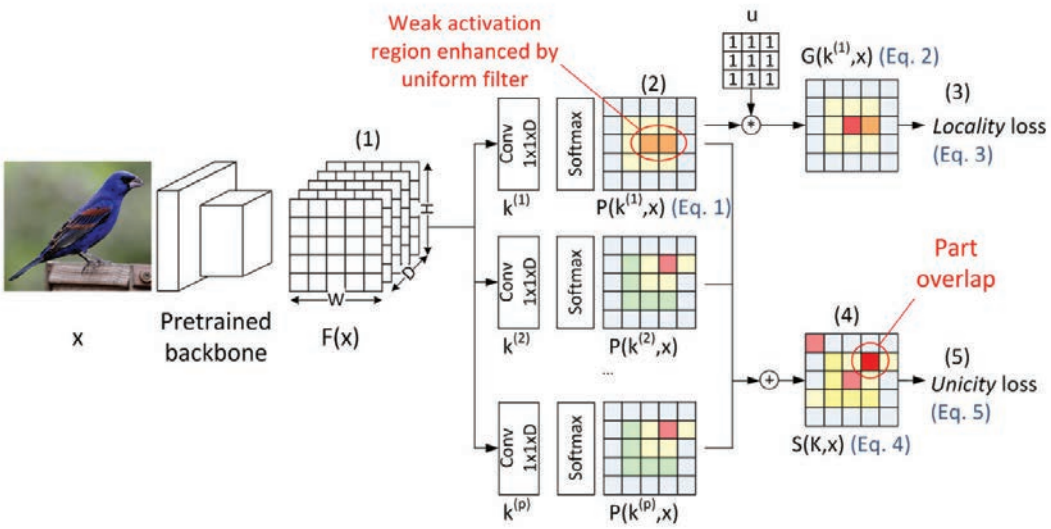
Applications:
- Interpretable out-of-disribution detection
- Boosting classification
- Aided annotation
- Explainability, as a brick of case-based reasoning



PARTICUL
PART Identification with Confidence through Unsupervised Learning

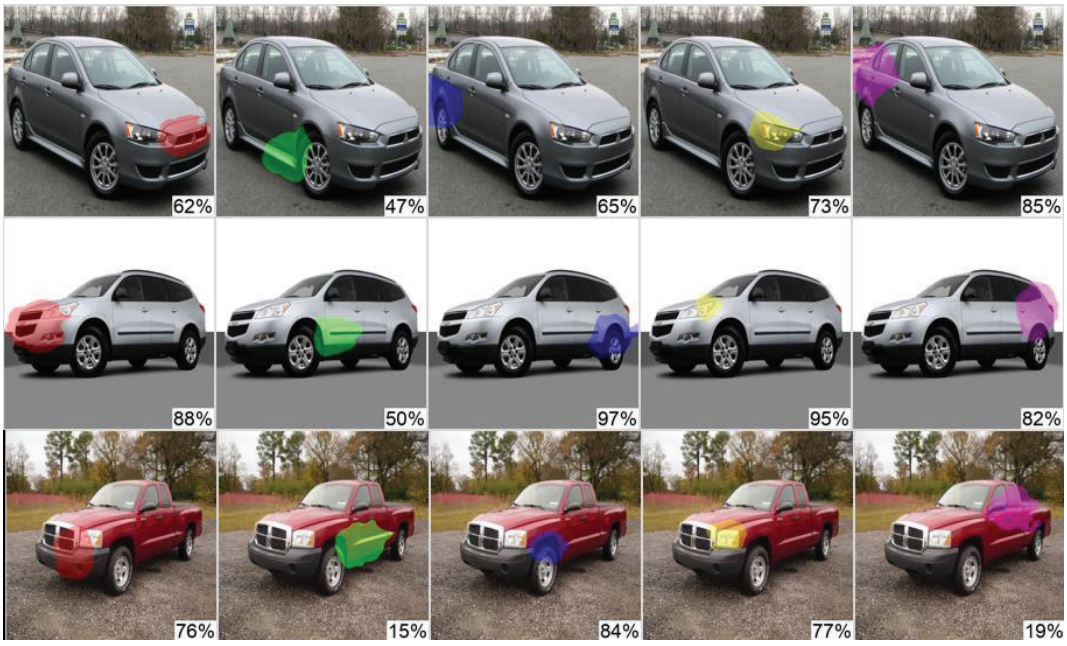
184

Formal Methods are not a panacea



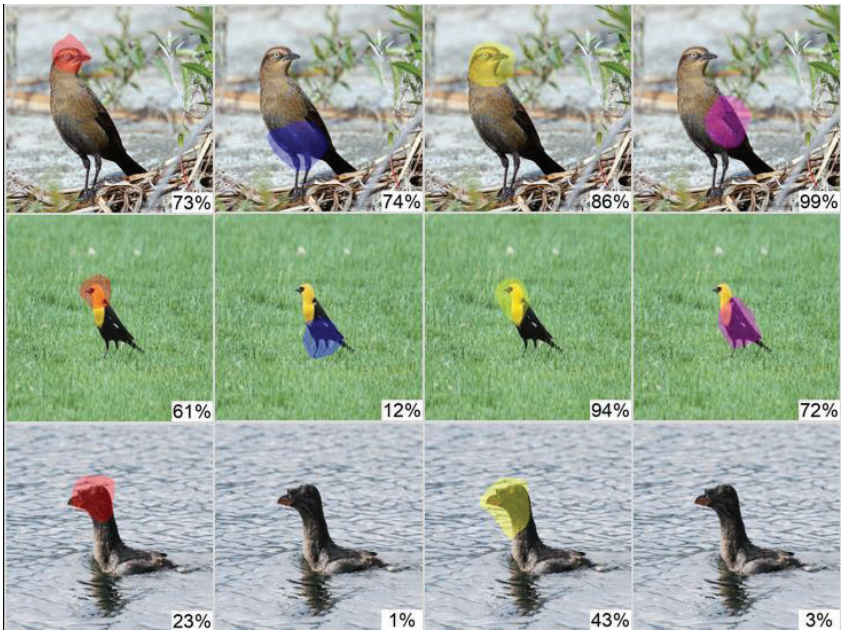
185

Formal Methods are not a panacea



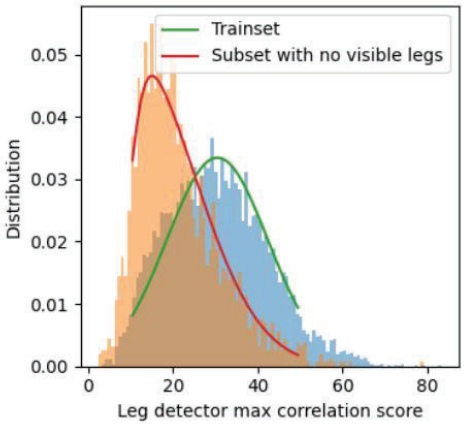
186

Formal Methods are not a panacea

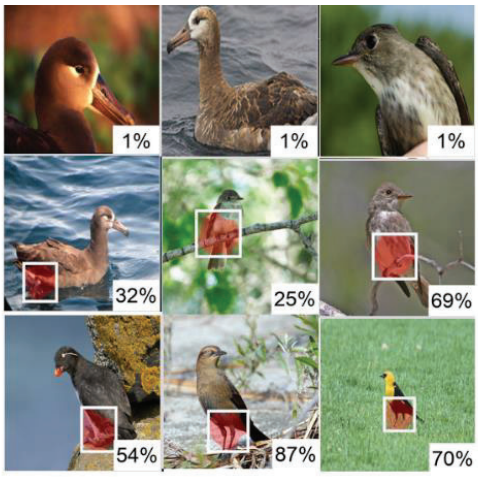


187

Formal Methods are not a panacea



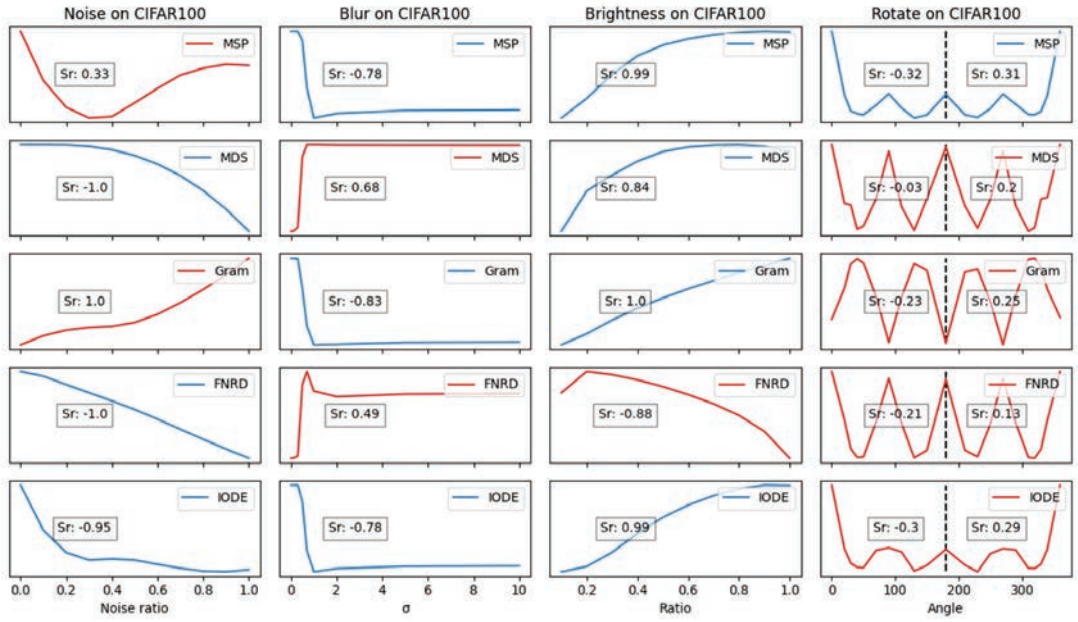
(a) Distribution of maximum correlation scores on the CUB-200 training set (in blue) and on a subset containing only images with non-visible legs (red).



(b) Confidence scores and part visualizations on images with non visible legs (top-row) and with visible legs (bottom rows).

188

Formal Methods are not a panacea



THALES

Conclusion

- FM need adaptation but the challenge is not new
- FM are advancing faster than ever before
- Very dynamic and fast moving environment
- FM are only one tool in what should be a diversified panoply
- FM can intervene at different stages of development

190

SELFSUSTAINED CROSS-BORDER CUSTOMIZED CYBERPHYSICAL SYSTEM EXPERIMENTS FOR CAPACITY BUILDING AMONG EUROPEAN STAKEHOLDERS

Design considerations and methodology in designing general purpose low-cost embedded systems

Prof. dr Radovan Stojanović
University of Montenegro and MECOnet

SS-CPSIoT2023, Budva, Montenegro

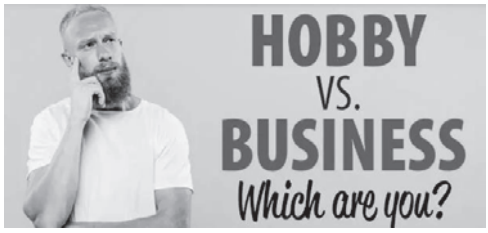
Co-funded by the Horizon 2020 programme of the European Union
DT-ICT-01-2019
Smart Anything Everywhere Area 2
www.smart4all-project.eu
Grant Agreement: 872614

Content

- Introduction
- Design methodology with examples
- Conclusions

Introduction

- **Resolve the dilemma. Hobby or business?**
- **Business. How can I solve the problem in the simplest and most economical way?**
- **Do not shoot sparrow with cannons?**
- **Technological business is different than standard one, needs knowledge and skills.**
- **"Sometimes, when you turn a hobby into a job, it becomes work". *Jeff Bennett***
- **"Whatever you like to do, make it a hobby and whatever the world likes to do, make it a business", *Warren Buffett***
- ***Story about function and arguments.***

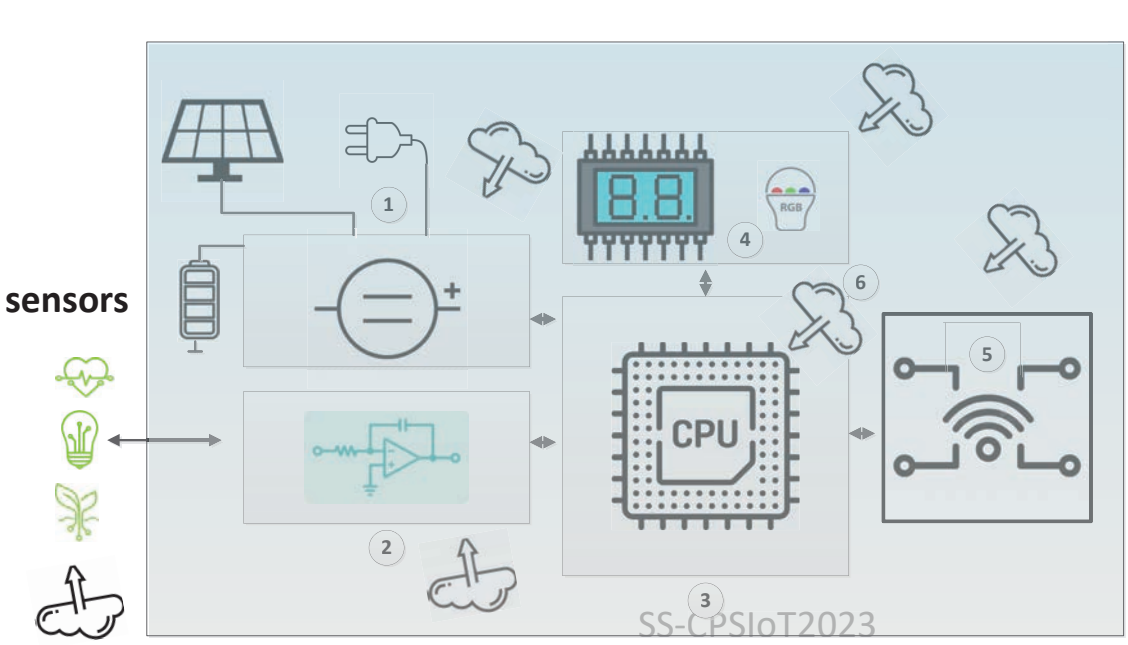






Design methodology

- **Each component of is a story in itself.**



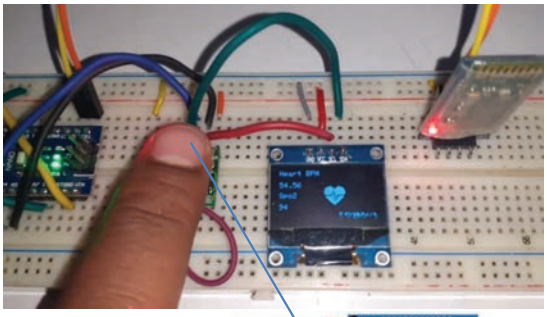


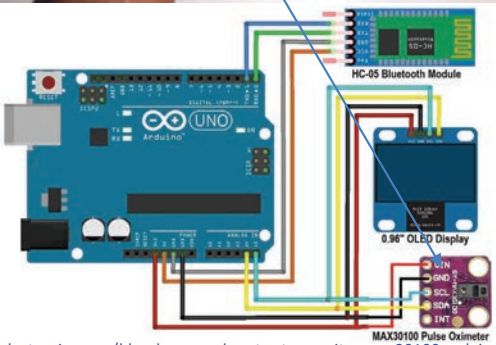
7

1: Power
2: Front End
3: CPU
4: VI (Visualization interface)
5: Communication
6: Software
7: Enclosure (box)

Design methodology

- Amateur, student or educator approach.





https://how2electronics.com/blood-oxygen-heart-rate-monitor-max30100-arduino/

- Commercial versions



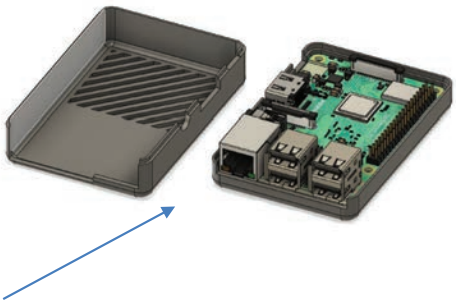
https://www.scientificanimations.com/pulse-oximetry-mechanism-history-use-sources/uncategorized/

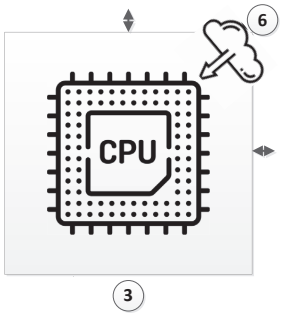


https://www.youtube.com/watch?v=SdK1y-52or4

Design methodology

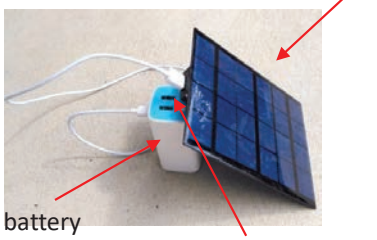
- Where to start?
- 1,2…7?
- 7,6,5….
- or other order?
- Experienced designer starts from 7.
- Educator from 0, that is not here.
- Amateur from 6 or 3.





Design methodology

- **Powering issues**:
  - Types: standard supply, battery, rechargeable battery, solar, wind, earth, USB..
  - Regulators, Linear Low Power, Low Drop Out (LDO), DC-DC convertors
  - Voltage capacity, Current capacity, **quiescent current**, **on-off**
- Sometimes we can take energy from nothing.
- **Simplest example .** No battery, 3 in 1. Analog metering. Use a soil as battery and analog circuits as an actuator.
- Similar with sun.



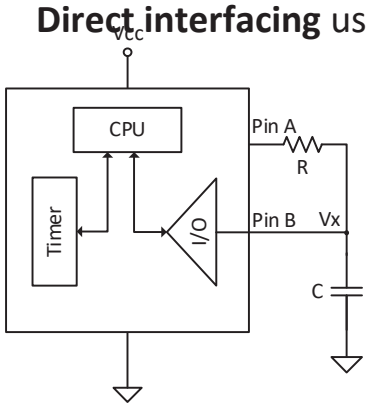




Design methodology

- **Front End (FE)**
- **Analog (AFE)** , direct interfacing, using passive components, using active components…
- **Digital (DFE),** using analog to time or frequency convertors or serial communications, i.e. SPI.
- **Conditions.** analog knowledge and experience. Use as much as possible built-in components in MCU, Comparator, A/D, D/A, PGAs. RC principle, charging discharging.
  - **Direct interfacing** using analog components. No depended on power supply.

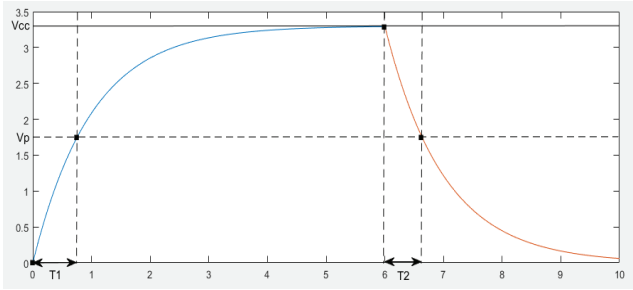




$$Cx = -\frac{1}{R}\frac{B + \sqrt{B^2 - 4AC}}{2A}$$

$$A = -[2 + \ln(2)]\ln(2)$$

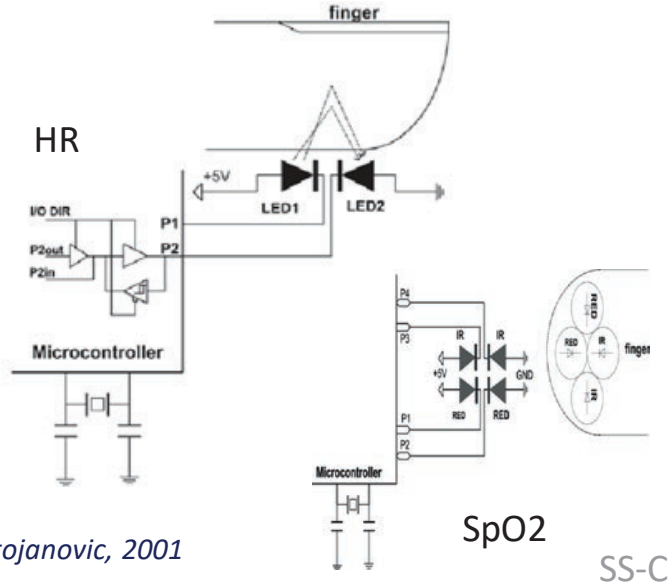
$$C = -\frac{{T_1}^2 + {T_2}^2}{2}$$



Design methodology

Direct interfacing. The simplest PPG front-end ever. HR configuration. Couple of LEDs are directly connected to MC's pins. One LED is using as light emitter and second as receiver.

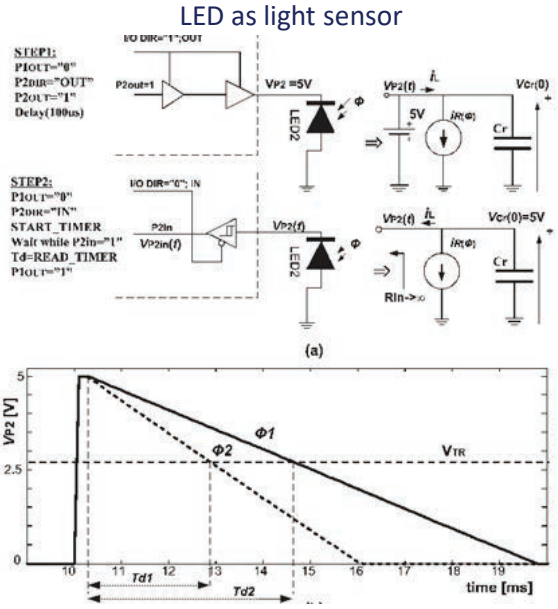
HR



Stojanovic, 2001

SpO2

LED as light sensor



SS-CPSIoT2023

Design methodology

- Convert analog value to frequency.
- Convert soil moisture in C, or R and then in voltage, current, time, frequency etc.. to be readable with MC.





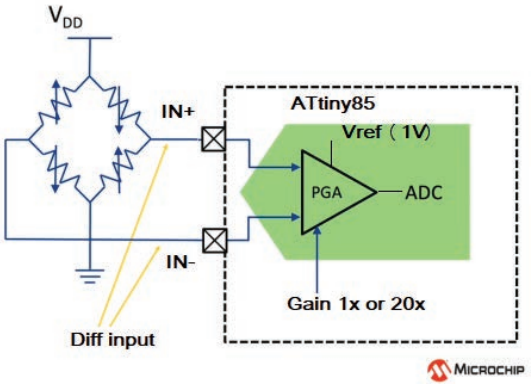
C



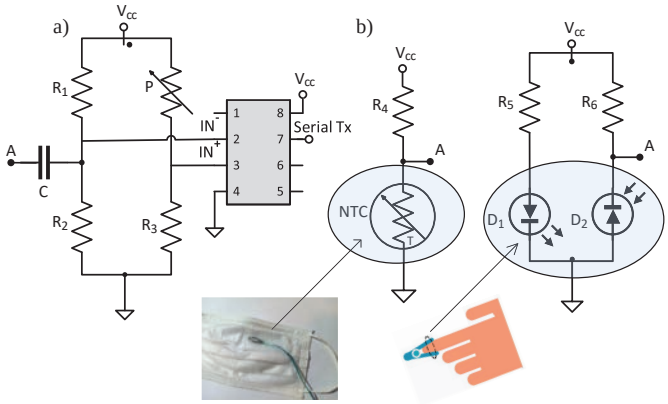


Design methodology

- **Examples of achieving good resolution with low end MC.** Heart Rate (RR) and Heart rate (HR) and Respiration Rate (RR) measurement. Vref=1.024V, PGA=20, achieving resolution at Diff mode 1mV/20=50uV. Even, temperature with K thermocouple could be measured with simple ATTiny.

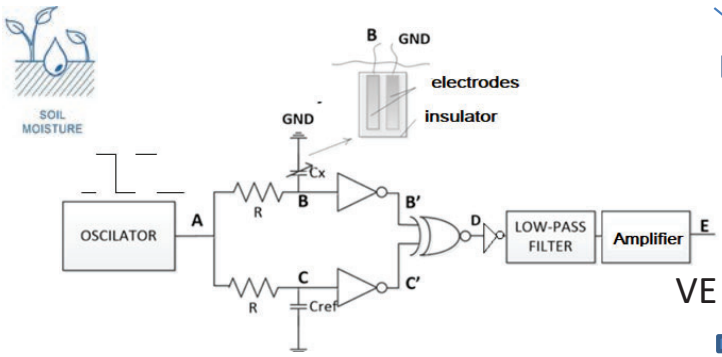


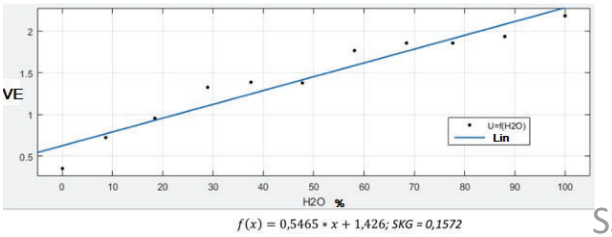




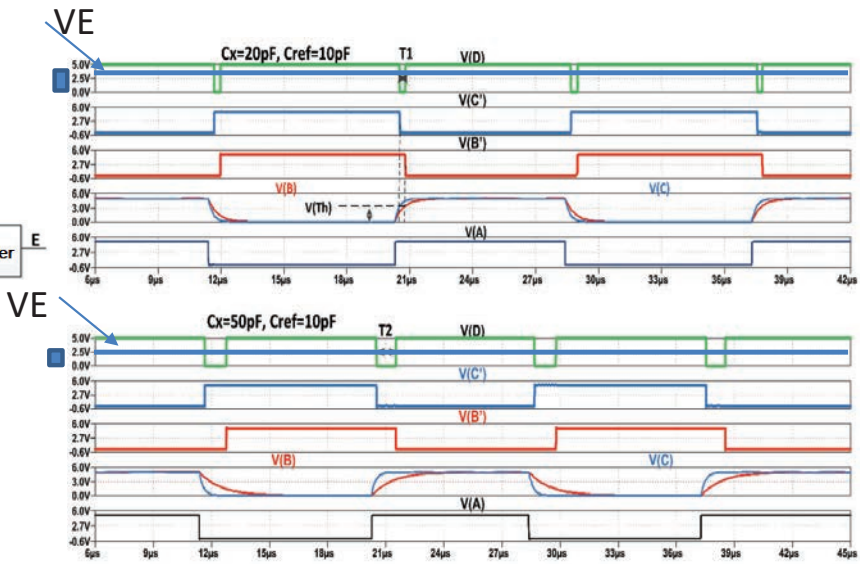
Design methodology

- **Delay to PWM and voltage.**



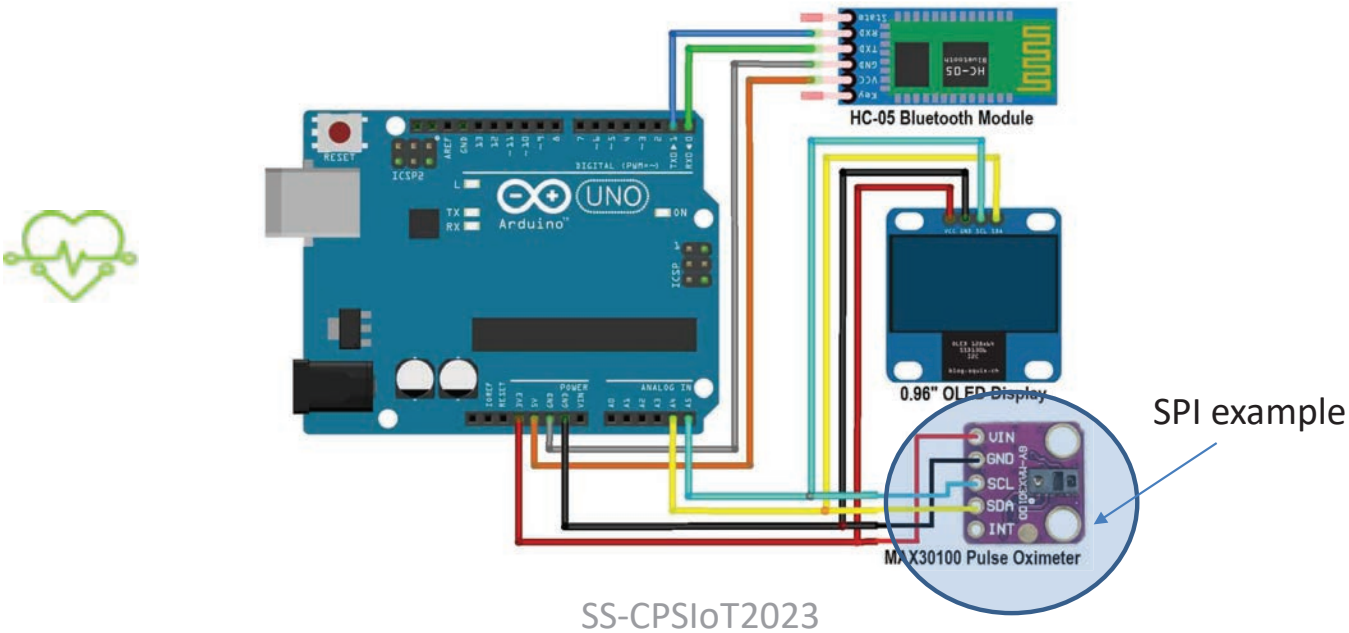


$f(x) = 0{,}5465 * x + 1{,}426;\ SKG = 0{,}1572$



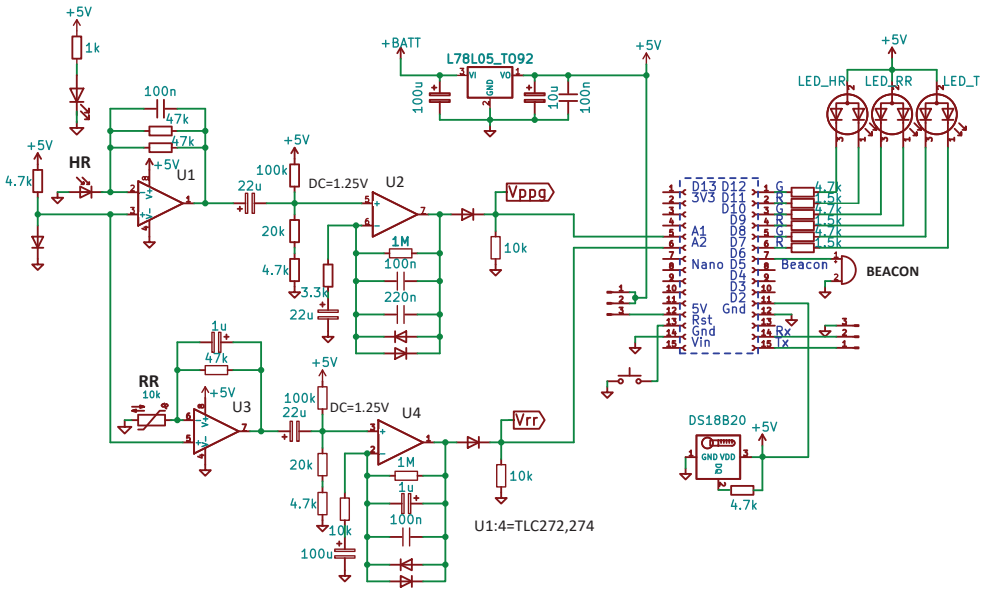
Design methodology

- Connection using Serial interfaces. SPI example



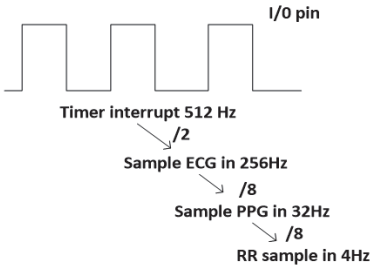
Design methodology

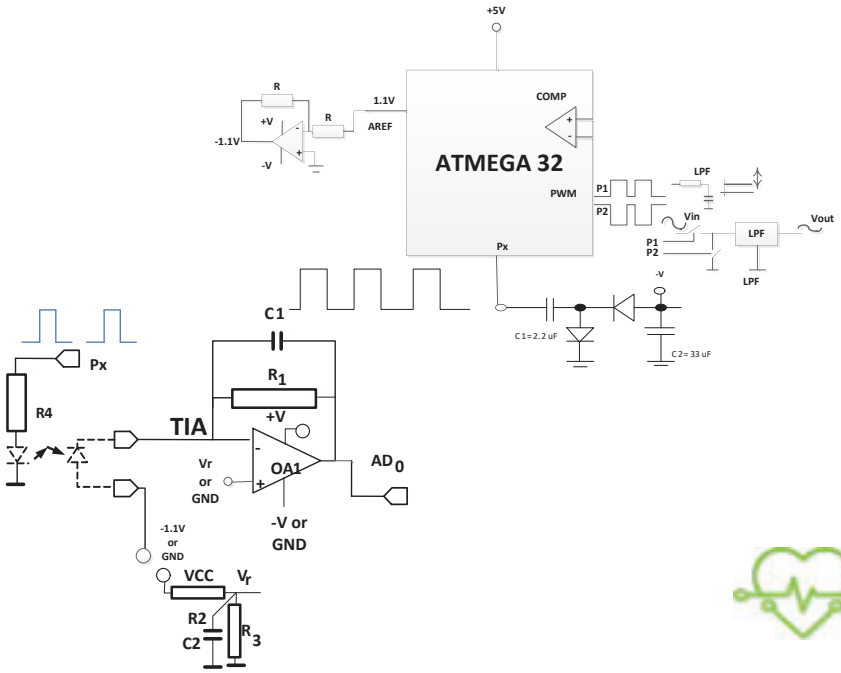
- **Analog design with active components**. HR and RR detection



Design methodology

Mixed AFE. Using all possible built in components in MCU and add the rest . As example with 4 OA (in one DIL14) we can implement very strong analog front end, Double supply+TIA+AGC+AMP.





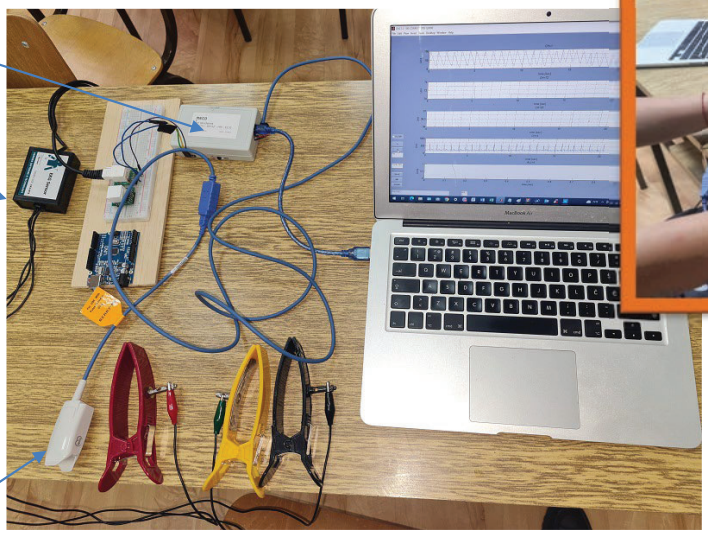
Design methodology

- **Combination of commercial and education**

MC Unit

ECG amplifiers

DS100 compatible
SpO2 probe





Students working
Their exercice

Design methodology

- Designing AFE
- Good knowledge of integrated electronics.
- Especially from filters and operational amplifiers.
- Single supply, double supply technique.
- Rail-to-Rail amplifiers.
- Voltage followers.
- Trans Impedance Amplifiers (TIA).
- Temperature dependence.
- Use LT Spice, very precision software.
- Simulate all scenario.
- Put original signals in LT spice.

Design methodology

3) MCU. Hardware, architecture**.**

- **8bits, 16 bits, 32 bits MC.**
- **From Atmel, Microchip ST Microelectronics, NXP, Espressif, Nordic**
- **Speed, internal or external crystal, power voltage limits, program memory, RAM, EEPROM, built in peripheries, timers, way of communication, way of programming, low power mode, package etc.**
- **Design request studious analyze in selecting proper MCU.**
- **You can use PIC and ATTiny Series, nowadays are popular ESP32, ESP8266 series. In many applications will finish job.**

Design methodology

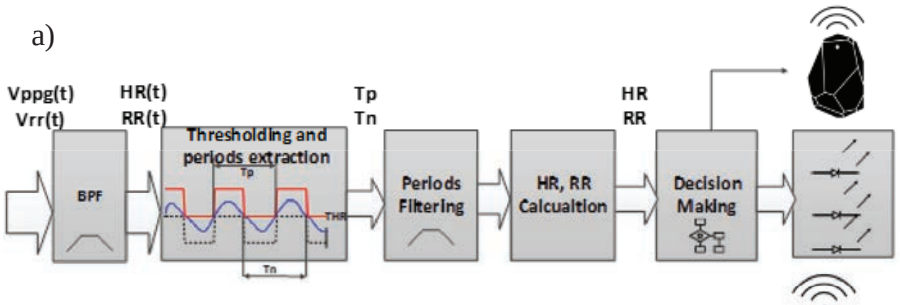
3) MCU. Software

- **general purpose,**
- **interrupt driven,**
- **DSP and AI**
- **Communication,**
- **Real time,**
- **Low power**
- **Code optimization techniques (memory and others)**

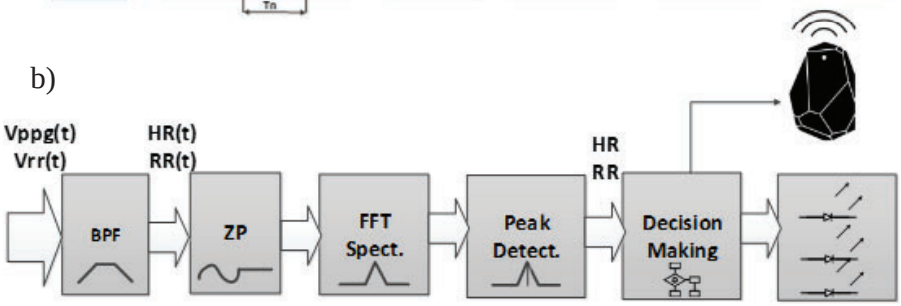
Design methodology

- **Processing steps in time domain a) and frequency domain b). On-chip implementation. HR and RR examples. Implementation in 8bits RISK**

a)



b)



Design methodology

- On-MC signal processing.
- All parameters should be on-place calculated, filter border frequencies, time responses, Spectrums, adaptive thresholds.
- Strongly integer (fixed point) based, highly optimized signal processing.
- Take care about sampling frequency and interrupt based sampling generation.

Design methodology

- **RC (LP) i CR (HP) 1st order filter implementation, using only fs and fc. Basic configuration.**

```
const int fs=200; //sampling frequency

//filter variables
const int fc_l=5; //corner frequency HP
float alfa=0; //coefficient LP
float y_old_lp=0; //previous value y LP

const int fc_h=15; //corner frequency LP
float beta=0; //coefficient HP
float y_old_hp=0; //previous value x HP
float x_old_hp=0; //previous value y HP

void setup()
{
.....
  alfa=calculate_alfa((float)(fc_h), fs); //calculate_alfa
  beta=calculate_beta((float)(fc_l), fs); //calculate beta
....
}
//coefficient alfa for LP filter
float calculate_alfa(float fc, float fs)
{
float alfa;
alfa=(2*PI*fc/fs)/((2*PI*fc/fs)+1);
return alfa;
}

//coefficient beta in HP filter
float calculate_beta(float fc, float fs)
{
float beta;
beta=1/((2*PI*fc/fs)+1);
return beta;
}
```

Calculation of coefficients

```
//LP filter of 1st order
float low_pass1(float alfa, float x)
{
float y=0;
y=alfa*x+(1.0-alfa)*y_old_lp;
y_old_lp=y;
return y;
}
//HP filter of 1st order
float high_pass1(float beta, float x)
{
float y=0;
y=beta*y_old_hp+beta*(x-x_old_hp);
y_old_hp=y;
x_old_hp=x;
return y;
}

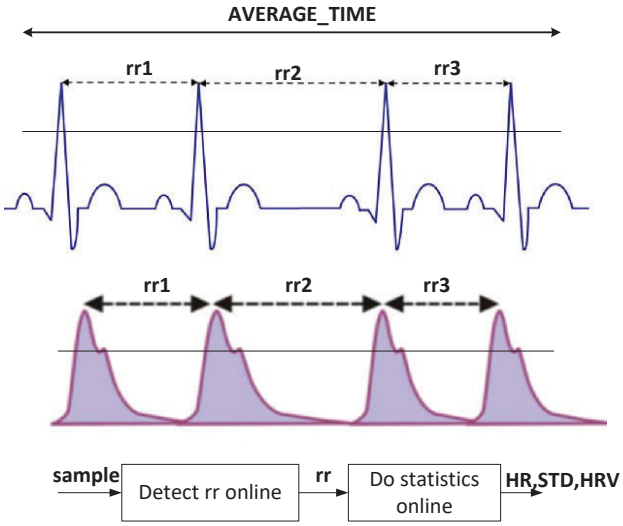
void loop()
{
.....
sample= analogRead(AD0);
y1=high_pass1(beta, float(sample));//HPF 5Hz
y2=low_pass1(alfa,y1); // LPF 15Hz
......
}
```

Design methodology

Calculation of statistical parameters like standard deviation on-line without occupying memory

```
//DO STATISTICS FOR HR AND STDEV
short int stat_count=0;
long int par_sum_rr=0; //partial sum for mean value
long int std_sum=0;  //partial sum for std value
short int HR_AVE=0; //HR in AVERAGE_TIME
int STD=0; // STD in AVERAGE_TIME
int arithmia=0; //arithmia counter
void do_statistics(int rr)
{
 float B;
 if(rr>1500 || rr<500) // Arrhythmias detected
{
  HR_AVE=0;
  arithmia++;
  par_sum_rr=0;
  std_sum=0;
  stat_count=0;
  STD=0;
}
else
{
 stat_count++;  //statistics counter
 if(stat_count>1)
 {
 par_sum_rr=par_sum_rr+long(rr); //partial sum for mean value
 std_sum=std_sum+long(rr)*long(rr); //partial sum for std value
                              if(stat_count>=AVERAGE_TIME)
                                {
                                 B=float(par_sum_rr)/float(AVERAGE_TIME-1); //mean value
                                HR_AVE=short(60000/B); // HR from mean value
                                B=(B*B); // mean*mean
        std_sum=std_sum/(AVERAGE_TIME-1);
        STD=round(sqrt(float(float(std_sum)-B))); //formula for standard deviation
                                stat_count=0;
                                par_sum_rr=0;
                                std_sum=0;
                                } } }
}
```

$s^2 = \frac{\sum x^2}{n} - \bar{x}^2$ instead $s^2 = \frac{\sum (x-\bar{x})^2}{n}$



Design methodology

```
//IIR FILTER FLOAT IMPLEMENTATION
//a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
  //       - a(2)*y(n-1) - ... - a(na+1)*y(n-na)
  // calling   yout=iir_filtar(xin, a_c, b_c, n);
#define N 4
double y[N+1]={0,0,0,0,0};
double x[N+1]={0,0,0,0,0};

// a(1)*y(n)=b(1)*x(n)+b(2)*x(n-1)-a(2)*y(n-1)
double a_c[]={1.0000, -0.9975}; //floating coefficients
double b_c[]={ 0.0013,  0.0013};

double iir_filtar(double p, double *a_coef, double *b_coef, int N_order)
{
  int i;
  x[0]=p;
  y[0]=*b_coef*x[0];
  for(i=1; i<=N_order; i++)
  y[0]=y[0]+(*(b_coef+i)*x[i]);
  for(i=1; i<=N_order; i++)
  y[0]=y[0]-(*(a_coef+i)*y[i]);
  for(i=N; i>0; i--) //Circular
  {
  y[i]=y[i-1];
  x[i]=x[i-1];
  }
  return(y[0]); }
```

```
//IIR FILTER INTEGER IMPLEMENTATION
long a_co[]={1, -199};  //integer coefficients
long b_co[]={29,  29};
long yi[N+1]={0,0,0,0,0};
long xi[N+1]={0,0,0,0,0};

long iir_filtar_int(long p, long *a_coef, long *b_coef, int N_order)
{
  short i;
  xi[0]=p;
  yi[0]=(*b_coef*xi[0])>>8;

  for(i=1; i<=N_order; i++){
  yi[0]=yi[0]-((*(a_coef+i)*yi[i])>>8);
  }


  for(i=N; i>0; i--) //Circular
  {
  yi[i]=yi[i-1];
  xi[i]=xi[i-1];
  }
  return(yi[0]);
}
```

```
p=(long)(x<<8);  //Calling integer IIR filter
yk=iir_filtar_int(p,a_co,b_co, 1);
```

Design methodology

```
//DC REMOVAL
float al=0.995;
float yn_1=0;
float xn_1=0;

float DC_removal(float x)
{
                    float y;
                    y=al*yn_1+x-
xn_1;
                    yn_1=y;
                    xn_1=x;
                    return(y);

}
```

```
//DC TRACKING
int32_t ydc_old=0;
int DC_Tracking(int x)
{
int32_t ydc;
  ydc= ydc_old+((((int32_t) x << 16) - ydc_old) >> 9);
  ydc_old=ydc;
  return (ydc>>16);
}
```

$$H(z)=\frac{1-2\cos\omega_0 z^{-1}+z^{-2}}{1-2r\cos\omega_0 z^{-1}+r^2 z^{-2}}$$

```
//IIR NOTCH FILTER WITH //COEFFICIENTS
CALCULATION
fs=1000;
f0=50;  // REMOVE 50Hz flicker
b0=1;
b1=-2*cos(2*pi*f0/fs);
b2=1; r=0.999;
a0=1;
a1=-2*r*cos(2*pi*f0/fs)
a2=r*r;
```

Design methodology

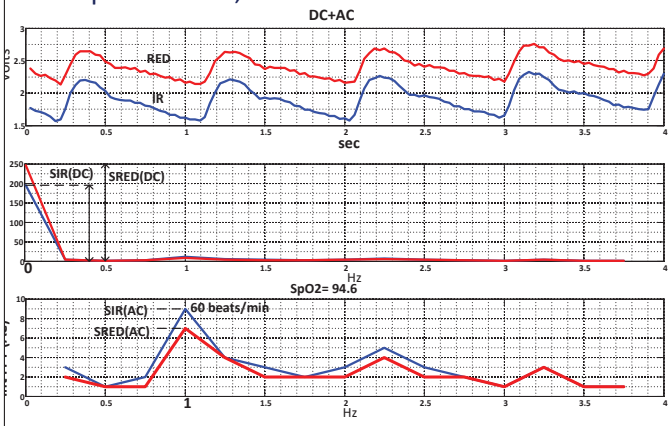
```
//SMOOTHING, CIRCULAR BUFFERING
float average_sum(float x)
{
short i;
float filterout=0.0;
// Direct-Form FIR
del[0] = x; // input for filter
filterout = del[0]; // Set up filter sum
for (i = LENGTH-1; i > 0; i--){ // Get sum of products
filterout += del[i];
del[i] = del[i-1]; // Renew input array
}
return (filterout);
}
```

```
//POSITIVE SLOPE calculation
int16_t x_old_slope_fix=0;
int16_t slope_fix(int16_t x)
{
int16_t slope=0;
slope=x-x_old_slope_fix;
if(slope<=0) slope=0;
x_old_slope_fix=x;
return slope;
}
```

Design methodology

```
//Finding SpO2 from FFT Spectrum for FFT_N points
…..
#include "fix_fft.h"  //Include fix_t libraray
…
//define variables for fix fft
char im[FFT_N], data_dc_red[FFT_N], data_dc_ir[FFT_N],
data_ac_red[FFT_N], data_ac_ir[FFT_N];
if(k> FFT_N) //when the number of samples exceed FFT_N(256){
  ……
//SPO2 calculation
      for(i=0; i<FFT_N; i++) im[i]=0;
      //RED FFT DC
      fix_fft(data_dc_red, im, 8, 0);  //Call Fix FFT
      MAX_DC_RED = sqrt(data_dc_red[0] * data_dc_red[0]
      + im[0] * im[0]); //Spectrum(0)
      //IR FFT DC
      for(i=0; i<FFT_N; i++) im[i]=0;
      fix_fft(data_dc_ir, im, 8, 0); //Call Fix FFT
      MAX_DC_IR = sqrt(data_dc_ir[0] * data_dc_ir[0] +
      im[0] * im[0]);
      //HR from IR and finding maximum in FFT AC RED
      // Spectrum
      dat=0;
      HR_RED=0;
      MAX_AC_RED=0;
      for(i=0; i<FFT_N; i++) im[i]=0;
      fix_fft(data_ac_red, im, 8, 0);
      for (i = 1; i < FFT_N/2; i++)
            {
            dat = sqrt(data_ac_red[i] * data_ac_red[i] +
            im[i] * im[i]);
            if (dat> MAX_AC_RED) {HR_RED=i;
             MAX_AC_RED=dat;}
            }
       //Finding maximum in FFT AC IR Spectrum
      dat=0;
      HR_IR=0;
      MAX_AC_IR=0;
      for(i=0; i<FFT_N; i++) im[i]=0;
      fix_fft(data_ac_ir, im, 8, 0);
      for (i= 1; i < FFT_N/2; i++)
            {
            dat = sqrt(data_ac_ir[i] * data_ac_ir[i] +
            im[i] * im[i]);
             if (dat> MAX_AC_IR) {HR_IR=i;
               MAX_AC_IR=dat;}
            }
      //Calculate RR
      float A= float(MAX_AC_RED)/float(MAX_DC_RED);
      float B= float(MAX_AC_IR)/float(MAX_DC_IR);
      RR=A/B;
      ........
      //Calculate SpO2
      SpO2=110-25*RR;
```

- Plying with peaks in FFT spectrum, case of SpO2 and HR, noise immune method



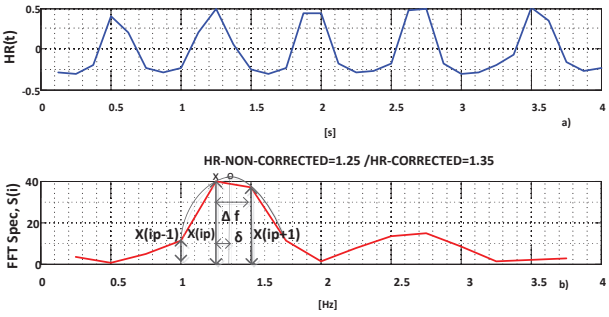
RR=SRED(AC))/SRED(DC)/SIR(AC)/SIR(DC)
SpO2=110-25*RR //approximatively equation

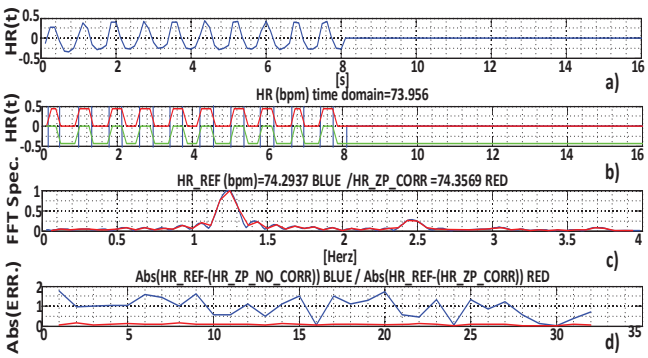
HR(Hz)=peak_position_in_Hz_off_SIR(AC)

Design methodology

- How I can implement strong embedded system i low end MC

$$\Delta f_{HR} = \Delta f/2 = fs/2 * NFFT \qquad f_{HRCORR} = f_{HR} + \delta = f_{HR} + \frac{X(ip+1) - X(ip-1)}{2 * (2 * X(ip) - X(ip+1) - X(ip-1))}$$





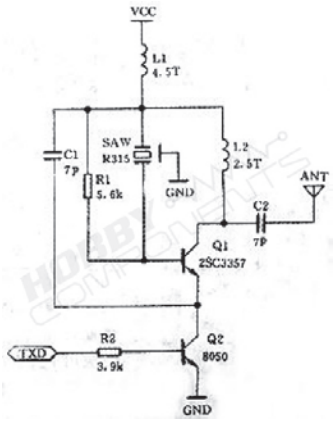
Design methodology

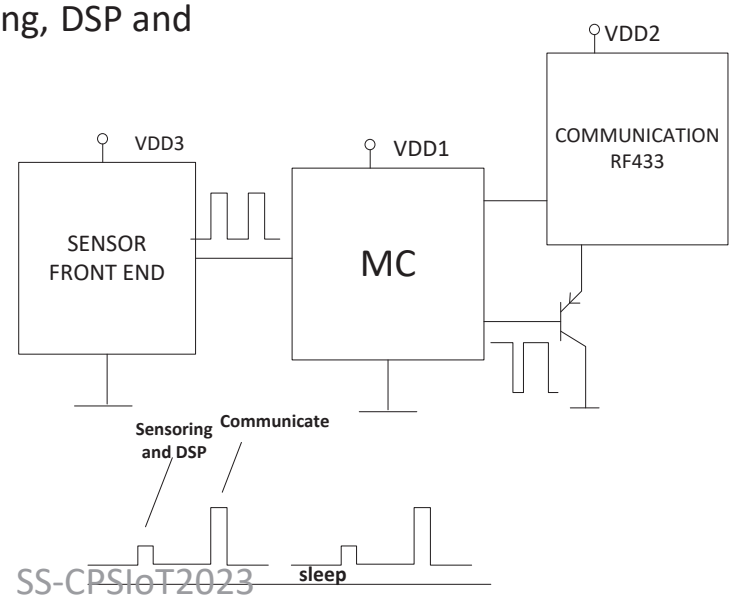
RAM Memory occupation for implementation of optimized HR and RR algorithms

RAM Memory occupation in function NFFT, 278 bytes are used for program and library variables		Minimum performances MCU
Time domain	311 bytes	ATTiny85
32 points	347 bytes	ATTiny85
64 points	470 bytes	ATTiny85
128 points	662 bytes	ATMega32P
256 points	1046 bytes	ATMega32P
512 points	1814 bytes	ATMega32P

Design considerations

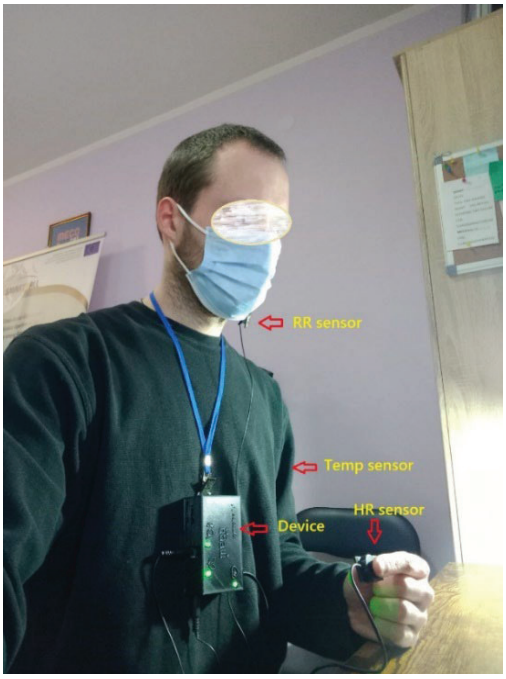
- **Power consumption**
- Use power consumptions saving modes on MC
- Take care about timing of sensoring, DSP and Communication.





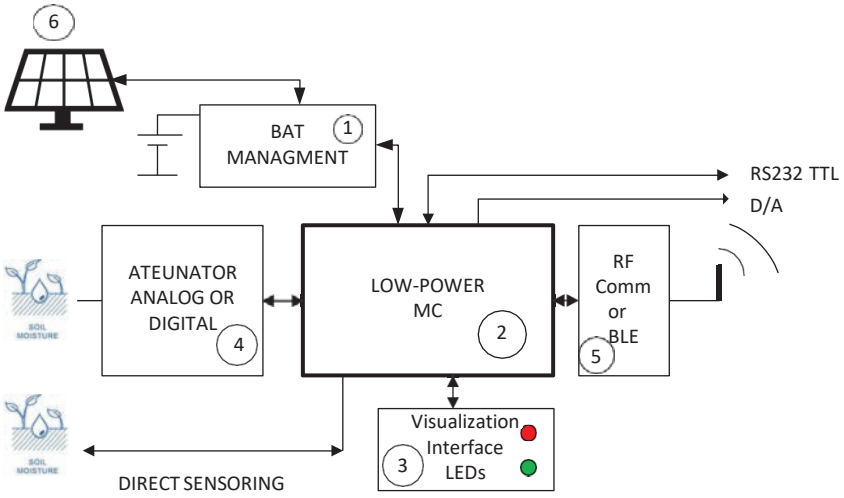
Integration

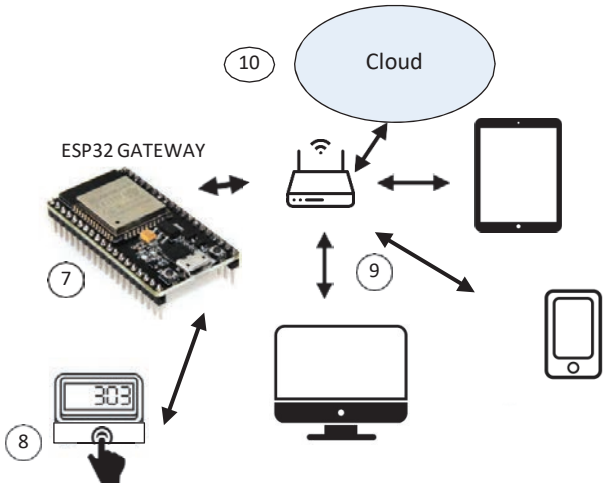
- Video Google



Integration

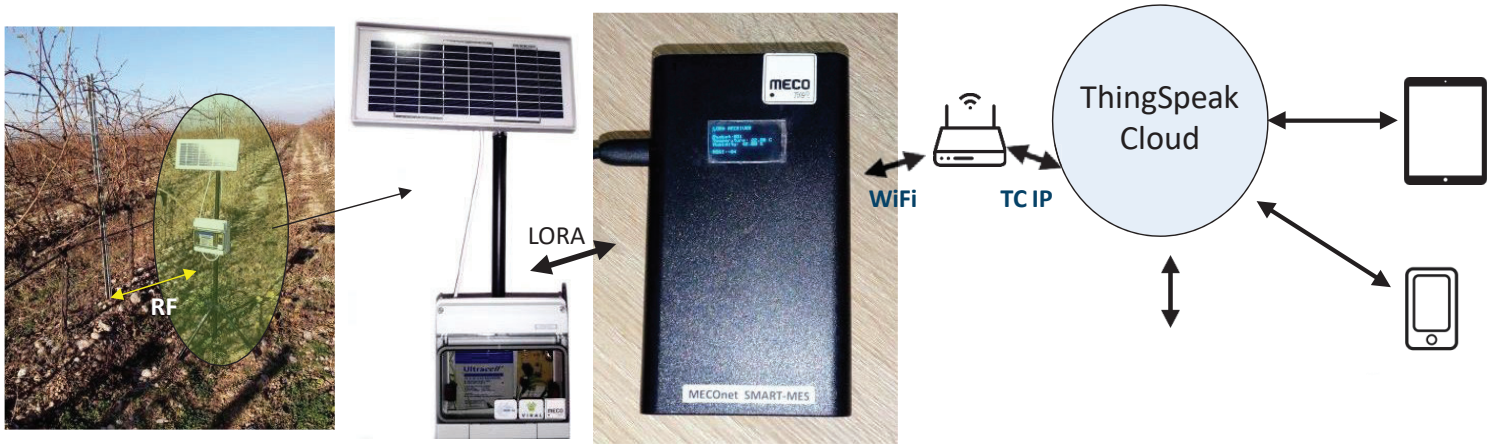
- **home(office) scenario**





Integration

- **On-field-cloud scenerio**

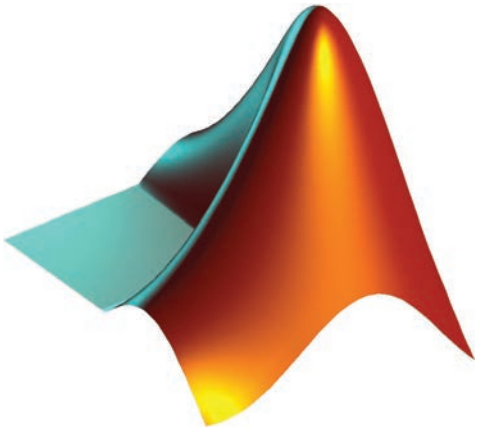


Design methodology

- How to develop algorithms for DSP on MCs
- LTSpice (Analog-Digital) –Microcontroller (Hardware Firware) – Matlab (DSP-AI)







Conclusions

- We discussed some of effective design and tips and tics.
- The approaches can be useful from basic till advanced levels of designing.
- Some examples of design considerations in both hardware and software are given.
- To design effective nodes we need wide knowledge, not only from computers.

Your feedback

Thanks. Questions, comments?



Intelligent Secure Trustable Things

Architecture and technology evolution for the 5G-IoT

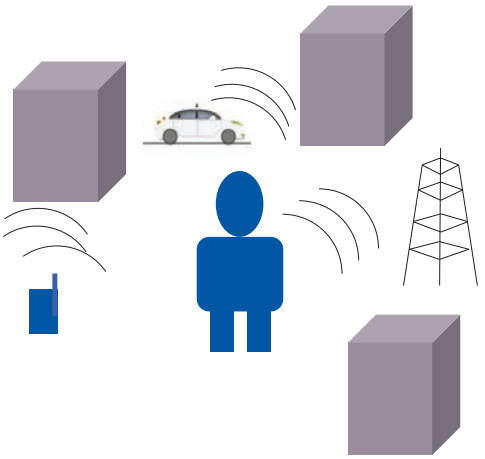
InSecTT has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey

Structure

- **Introduction to IoT**
- **Mobile communications**
  - 1G-5G
  - Technological transitions
  - IoT
- **Reference Architecture for IoT**
  - Views of the architecture
  - Layered models
  - InSecTT bubble and reference architecture
  - International standards for reference architecture
- **Technology analysis per view**

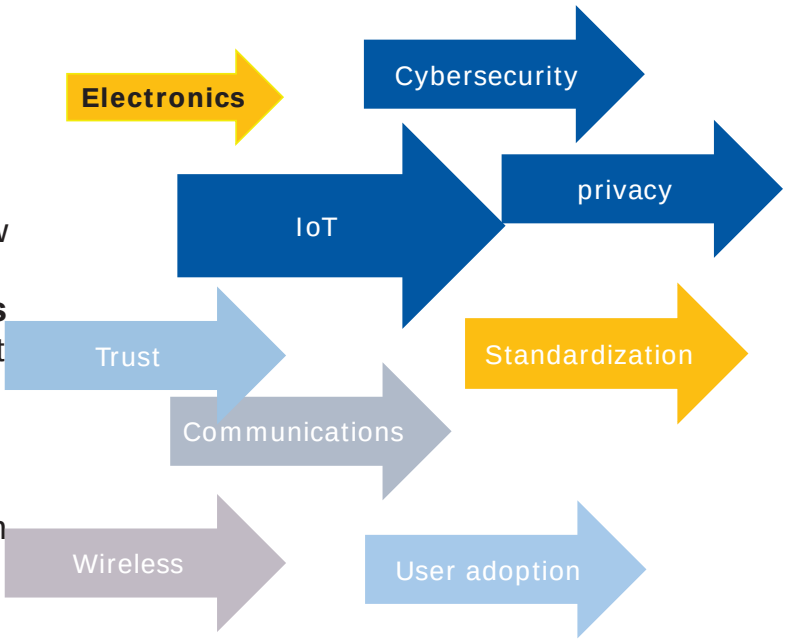
The Internet of Things

- A new paradigm for using **small embedded processors**, sensors, actuators in **every-day objects, machines or wearable devices** communicating with each other or with Edge/cloud infrastructure
- Objective: **Bridge the gap between the physical world of objects and the virtual world of computers**
- Huge implications on **automation, control**, **sensing of environmental parameters**, machine to machine communication, **automated industrial processes**, etc.
- Regarded as the **new industrial revolution**.
- **Internet of everything (IoE)**



How did we get here?

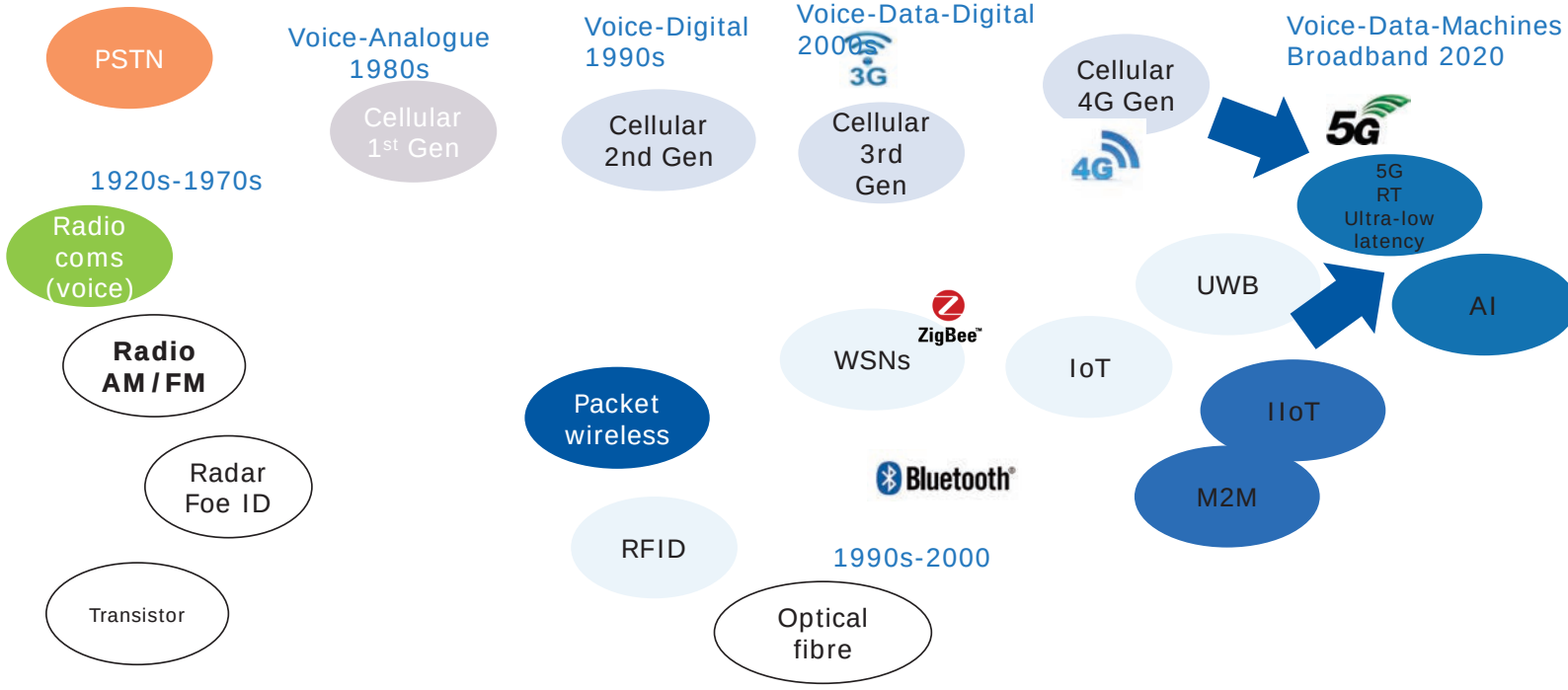
- **IoT** is an **evolution of multiple technologies**, **market acceptance**, and **socio-economical and political landscape**.
- Electronic components become more efficient and reduced in size. They are now cheap enough to be **embedded in small devices hosting sensors and actuators** that can detect/sense/modify environment variables or control machines.
- **Communication technologies** are also evolving to deal with constrained resource devices, low encryption capabilities, and in future scenarios real-time machines and applications.



How did we get here?

- **User adoption** has been evolving. The adoption of mobile communications provides a predecessor that can indicate openness of the market.
- At the political and social levels great efforts are being made to push for **standardization and certification of the potential new devices**. This obviously has strategic **global implications** and great **national security risks** that different countries are taking seriously.
- Some **EU relevant recent events**:
  - Privacy by design
  - **GDPR** adoption
  - Europrise privacy seal
  - IoT **trustworthiness label**
  - Cybersecurity budget in Europe increase substantially (**Cybersecurity act**)
    - Creation and strengthening of the ENISA

Technology evolution





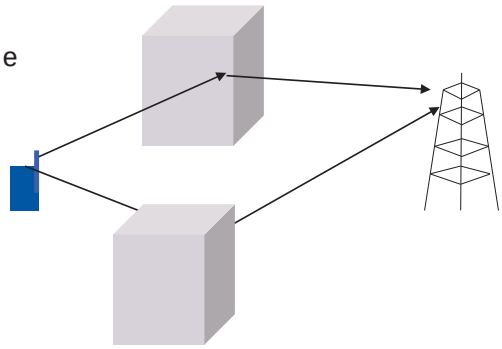
Intelligent Secure Trustable Things

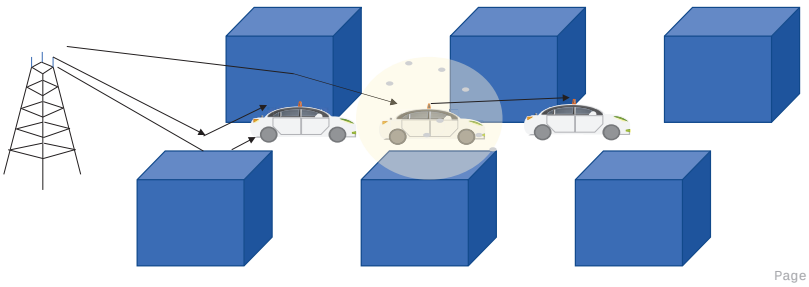
Mobile communications

InSecTT has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey

Mobile communications

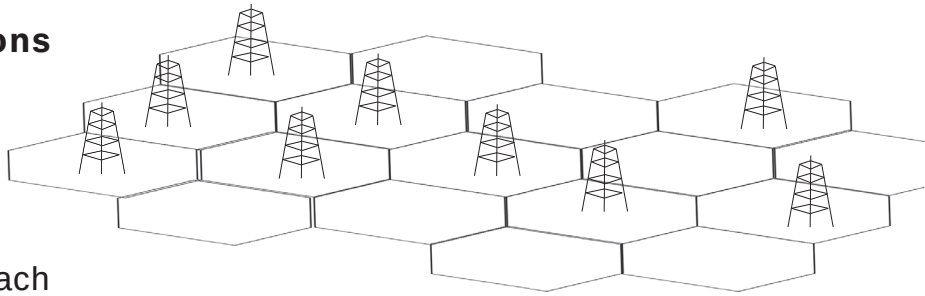
- Definition: **one of the ends changes position in time**
  - Challenges
    - More than one base station to provide coverage
    - Mobile terminal has lower complexity
    - **Multipath transmission**
    - Challenges of multipath channels (MHz-GHz)
      - Inter-symbol interference
      - Path loss
      - Shadowing
      - Frequency selectivity
      - Multiuser interference
  - Frequency reuse
  - Intra-cell and inter-cell interference
  - Doppler effects

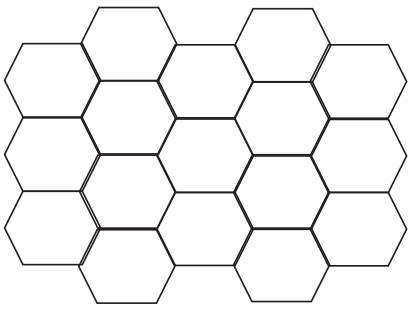




Cellular communications

- Urban service employs **base stations** positioned at **regular positions** creating an ideal **hexagonal grid**.
- Each base station creates a service area also called **cell**.
- In conventional cellular systems, each cell uses a **different frequency** to reduce intercell interference
- **Frequency bands are reused** according to available bandwidth
- **Handover between cells** is enabled by cellular BSs and a central station





Some parameters of Cellular networks

- **Spectral licensing** to operate entire urban services over cities or countries
  - Compared to **free licensing of 802.11** types of services
- **Complex Base station** to provide radio coverage in the order of 100-1000s meters
  - Cooling units
  - Radio resource units to provide capacity (voice and data)
  - Radio up and down conversion
  - Handover capabilities
- **Backhauling capacity** to forward traffic to the central phone unit or Access points
  - Microwave or optical fibre
- **Numbering and sigalling with conventional voice land networks**
- **Terminals certified to operate with network hardware**
  - Managed almost entirely by the operator



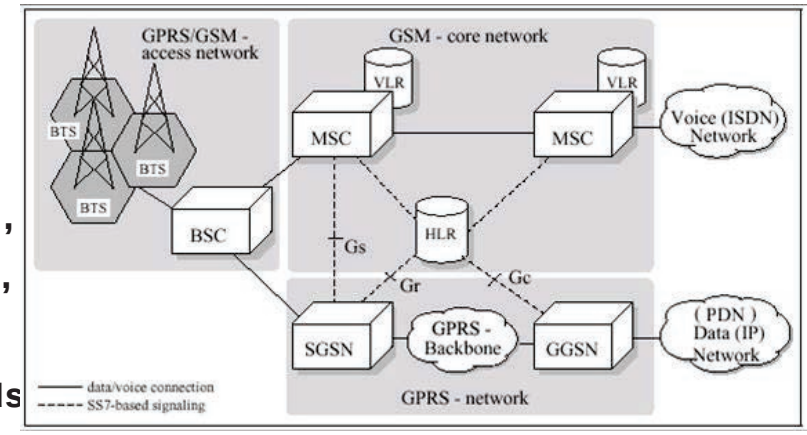
1G First Generation

- **Analogue voice transmission**
- Extension of PSTN (Mobile Switch): Public Swicthed Telephone Network
- Mainly vehicular devices
- **FDMA (Frequency division multiple access)**
- **AM/FM modulation**
- No data transmission
- Verticals
  - **Voice applications**
- North American standards: **AMPS (Advanced Mobile System): IS-45**
  - Carriers: **Vodafone, NTT, Verizon, e**tc



2G First Generation

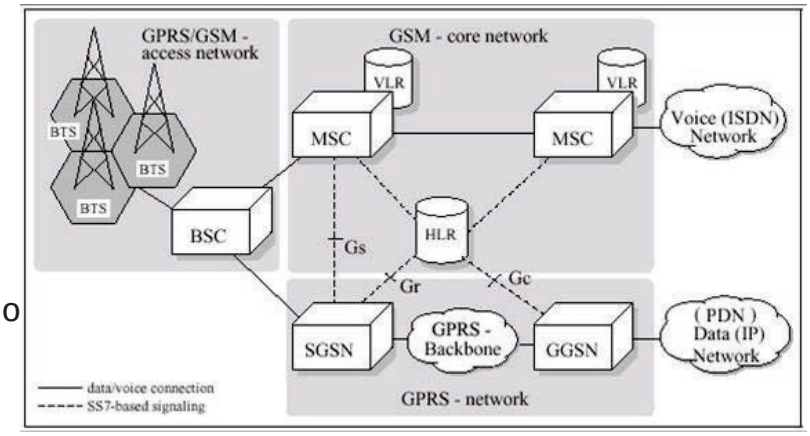
- **Digital Voice decoders**
- Circuit switches
- Limited **data transmission**
- **CDMA (Code Division Multiple Access),**
- **TDMA (Time Division Multiple Access),**
- **GSM (Global System for Mobile)**
- **GSM, IS-95 standards, Asian standards**
- More verticals:
  - Text messages
  - Location
  - Digital voice services
  - 2.5G Data connection 9.6kbps



BTS: Base transceiver station
BSC: Base Station Controller
MSC: Mobile Switch Centre
VLR: Visitor Locator Registry
HLR: Home Location register
GPRS: General Packet Radio Service
ISDN: Integrated Service Digital Network
PDN: Public Data Network

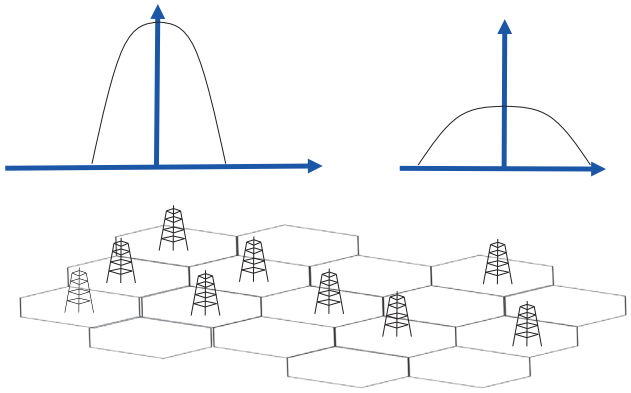
2.5G

- Support for **internet connectivity over a 2G network**
- **IWF** (InterWorking Function) added to the central unit
- **9.8 Kbps data rate supported**
- Not ideal for data connection
- Poor terminal support
- **Almost identical to landline modems**
- GSM: HDSPA , GPRS (General Packet radio Service)



CDMA (Code Division Multiple Access)

- **Spread spectrum**
  - Information is spread by **higher rate codes** that increase this spectrum of the signal
  - **Codes are orthogonal for different users and BSs.**
  - Signal is buried in the background noise
  - **Resistance to jamming and eavesdro**pping



- **Rake receivers**
  - Multipath components can be decoded by using rake receivers
  - Different multipaths create differences in the codes received at the destination

- Issues cellular CDMA
  - **Frequency reuse equal to one**
  - BSs are also identified by codes
  - **Noise created by code or pilot contamination**
  - **Breathing cell phenomena**

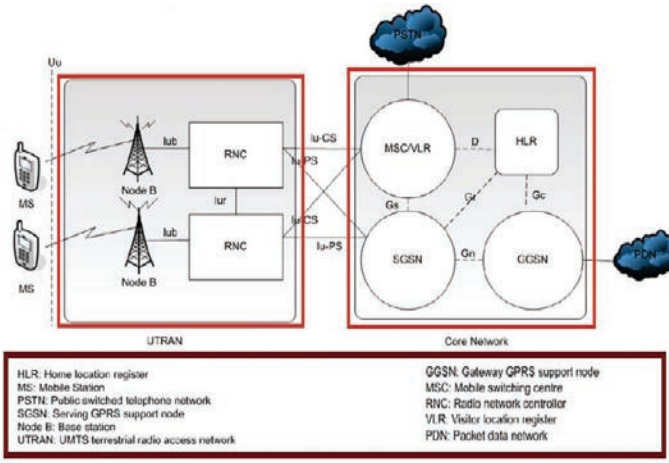
CDMA vs GSM (TDMA)

- **CDMA** for America and Asia (IS-95)
- **GSM** (Global System for mobile) based on TDMA for Europe (ETSI European Telecommunication Standard Institute)
  - CDMA seemed to be technologically superior but also expensive
    - Resistant to interference
    - Frequency reuse close to 1
    - Licensing costs?
  - TDMA is **simpler, but at the expense of capacity constraints in frequency reuse**.
    - Strong standardization in Europe
    - Wider variety of terminals
    - Short message service



3G Third Generation

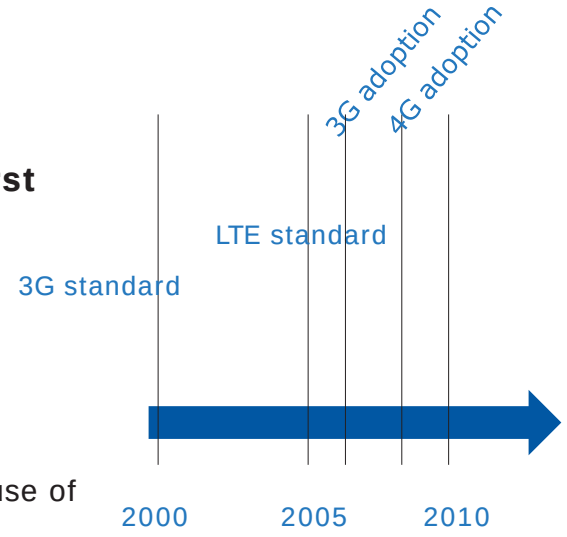
- Convergence **voice and data** (2Mbps)
- GPS chips in mobile phones
- **WCDMA (UMTS), CDMA200**
- Wideband CDMA
- Universal Mobile Telecommunications Systems
- 3GPP standards group
- **Standards ready by the year 2000**
- **Commercial success in Asia. World Cup 2002**



- Internet bubble bursts in early 2000s
- **3G resuscitates by 2007 with the introduction of Smart phones in Western countries.**
- In this case the communication technology standard did not create the market need.
- **The advances in mobile terminals paved the way for new digital services**

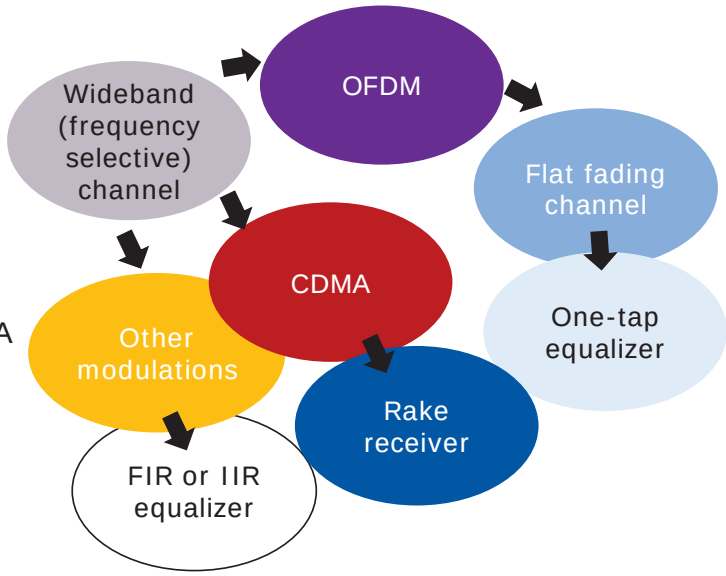
3G technology delay

- Standards were **ready by the year 2000**
- **No adoption until 2006-08.**
- 9/11 attacks created the first **technology bubble burst** crisis
- **Terminal technology not sufficiently evolved or attractive to the end user**.
- **Smart terminals changed the picture**
  - **Trust of the end user incresed**
  - In the meantime, **big 3G investor went bankrupt** because of this delay: British telecom
- WiFi gained momentum in this period.
- **GPS example: Trust in technology**



OFDMA (Orthogonal frequency Division Multiple Access)

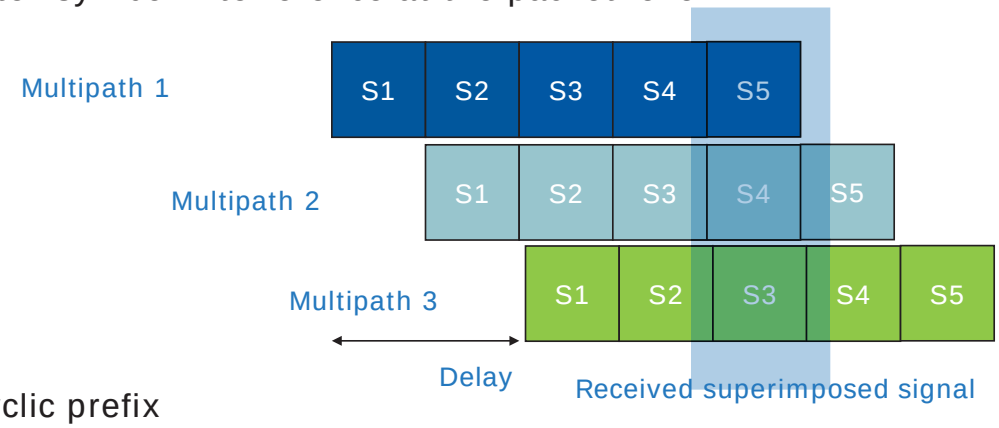
- OFDMA is a mixture of TDMA, FDMA and OFDM modulation
- OFDM modulation
  - Cyclic prefix and DFT compensate **multipath fading**
  - **Frequency selective channels become frequency flat**. So, one tap equalizers are used compared to long and complex equalizers of TDMA (depending on the number of multi-paths). **Reduction of equalization complexity**
  - **FFT implementations became low complex**
- OFDMA is suited for data transmission (Granularity in time and frequency)



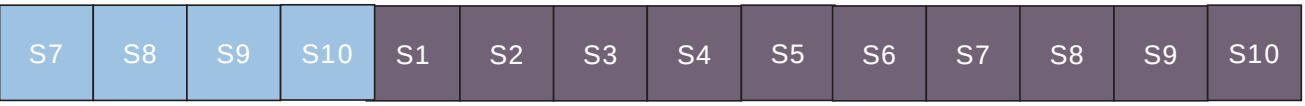
DFT: Discrete Fourier Transform
FFT: fast Fourier Transform
FIR: Finite Impulse Response
IIR: Infinite Impulse Response

OFDM (cyclic prefix)

- Inter-symbol interference at the packet level



- Cyclic prefix



OFDM (circulant matrix)

- Circulant matrix

$$\begin{bmatrix} h_1 & 0 & 0 & h_5 & h_4 \\ h_2 & h_1 & 0 & 0 & h_5 \\ h_3 & h_2 & h_1 & 0 & 0 \\ h_4 & h_3 & h_2 & h_1 & 0 \\ h_5 & h_4 & h_3 & h_2 & h_1 \\ 0 & h_5 & h_4 & h_3 & h_2 \\ 0 & 0 & h_5 & h_4 & h_3 \end{bmatrix}$$

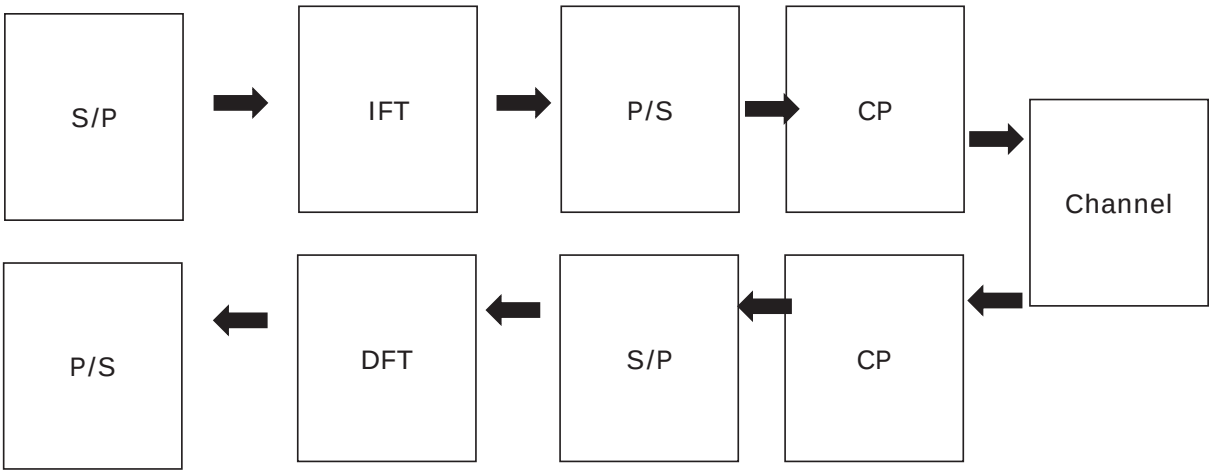
- DFT and IFT operations
  - Circulant matrices are diagonalized by the DFT matrix



OFDM (Information parallelization)

- Parallelization

P/S: Parallel to serial
S/P: Serial to Parallel
CP: Cyclic Prefix



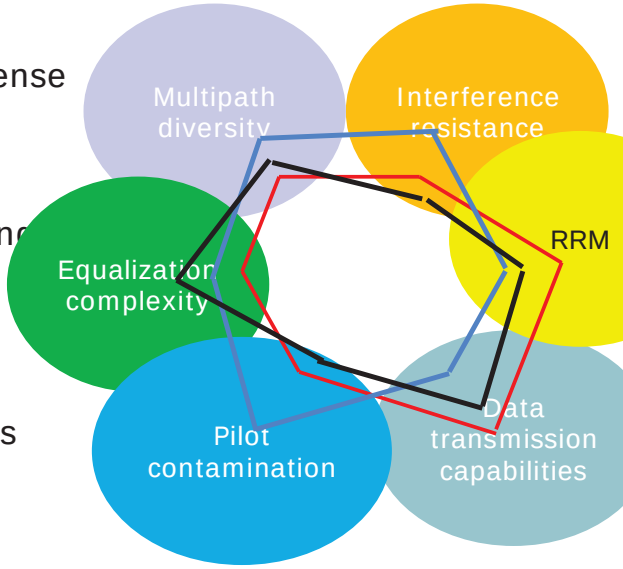
- **One tap equalizer**
  - Reduced complexity. **Frequency selective channel transformed into flat fading channel**

OFDM vs other solutions

- **OFDM vs conventional equalizer**
  - ***Reduction of complexity*** from FIR or IIR to one tap equalizer.
  - Training pilots are inserted as tones
- **OFDM vs CDMA**
  - CDMA uses rake receiver to exploit multipath diversity, while OFDM transforms it into flat fading
  - Rake receivers only consider the strongest paths
  - Tradeoff is strong diversity combining with high complexity costs of CDMA versus no diversity gains but a considerable complexity reduction of the equalizer design in OFDM
- **Commercial aspects**
  - OFDM reduces costs of terminals. Basis of 4G, 5G
  - CDMA offers gains for circuit switched technology. Basis of 2.5 and 3G.
  - CDMA associated with licensing to Qualcomm

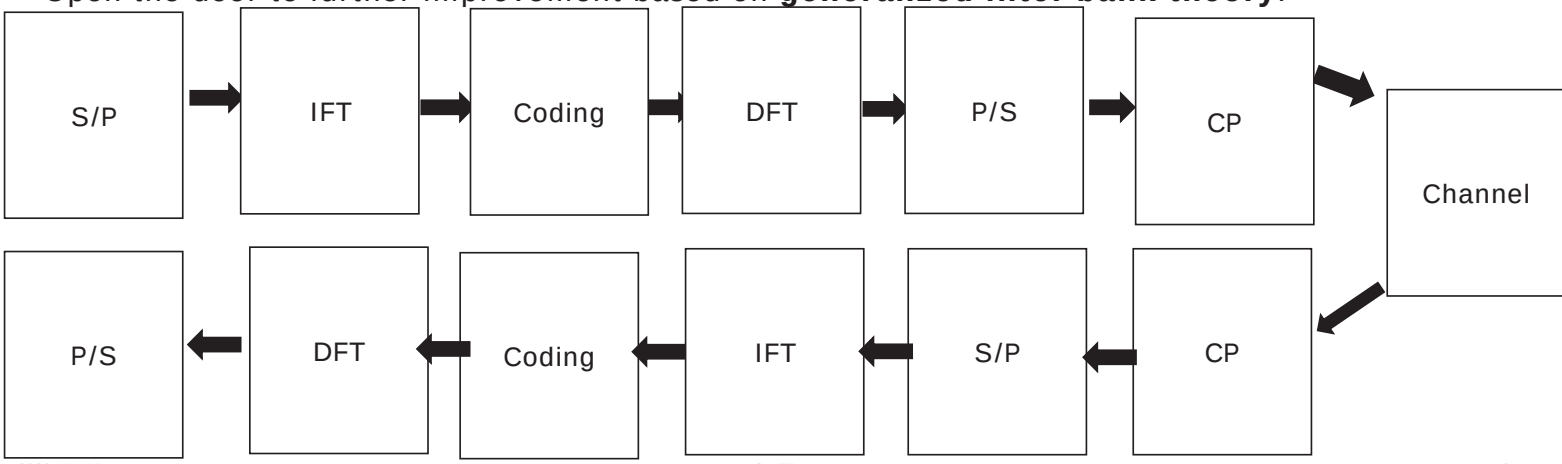
OFDMA vs CDMA vs TDMA

- OFDMA is **lower complex** than CDMA and TDMA
- OFDMA resolves easily **multipath fading** at the expense of **multipath diversity**.
- CDMA receiver exploits multipath diversity
- CDMA is **resistant to interference** due to high coding gains
- OFDMA is susceptible to interference from adjacent systems
- OFDMA has **high peaks of power** that distort signals
- TDMA suffers from **inter cell interference**
- CDMA suffers from pilot contamination
- OFDMA requires CQI reporting compared to conventional round robin TDMA



SCFDMA (Single Carrier frequency Domain Multiple Access)

- Similar to OFDMA but by using a **dual FFT (Fast Fourier Transform)** it avoids high power excursions.
- It is well-suited for **power constrained devices** (uplink)
- Carrier frequency offset and inter carrier interference could reduce performance more than in the OFDM itself
- Open the door to further improvement based on **generalized filter bank theory**.

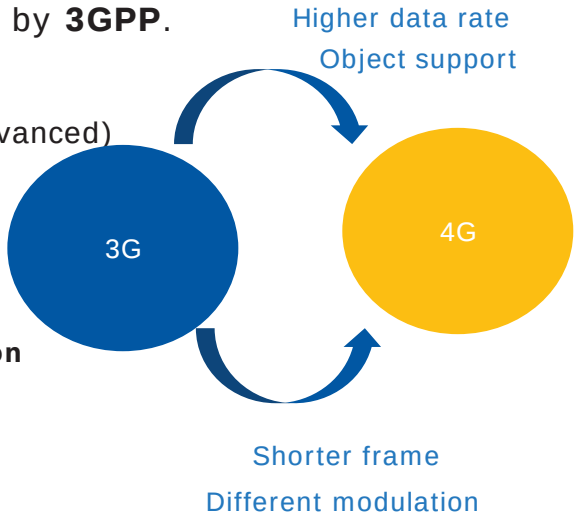


4G Fourth generation

- Data and voice improvement. 10-100 Mbps
- OFDMA and SCFDMA
- Homogeneous consensus for LTE standards under 3GPP
- **Human** Internet **connectivity**
- **Initial object and cloud connectivity (NB-IoT)**

3G vs 4G

- Complete change from **WCDMA** (3G) to **OFDMA/SCFDMA** (4G).
- America and Asia remained with some **CDMA2000** for 4G services
- **Long Term evolution (LTE)** is the term for 4G given by **3GPP**.
  - Change of frame to 10 ms.
  - Data rate improvement to 10-100 Mbps (1Gbps in LTE advanced)
  - **Influence of 802.11 technologies adopting OFDM**
  - OFDM **resilience to multipath fading**
  - **Vision to support data transmission**
    - **Business models with predominance of data transmission**
    - **Preliminary object support**



Manufacturers

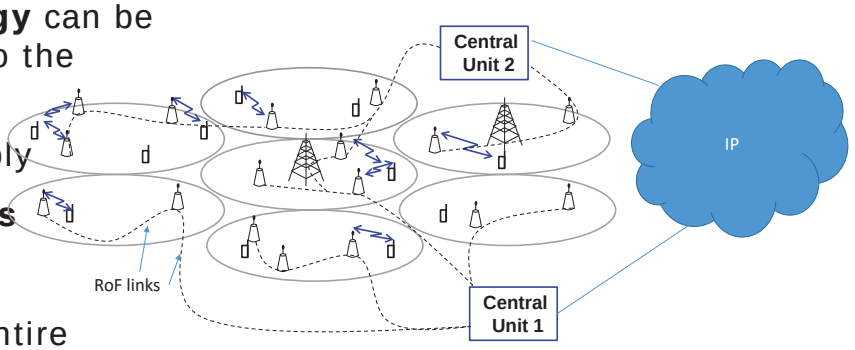
- Equipment manufacturer in 2G/3G: **Ericsson, Motorola**, Nortel, **Nokia, emergence of Huawei**
- 4G/5G: Dominance of Huawei, decline of others
  - Nokia shifted to terminals, but dominance eventually went to Samsung, LG, etc
- Manufacturer shift has caused controversy worldwide due to the change of the geopolitical landscape
  - Big European and American manufacturers have seen difficulties to tribe in the **base station and core network business**.

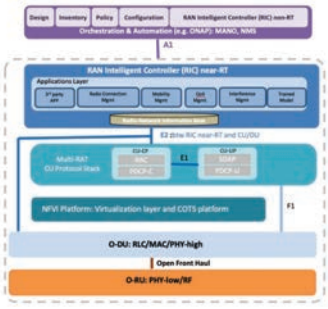
Operators

- **Small operators** tend to disappear or remain with a fixed customer base
- **Big operators** are becoming transnationals: Verizon, Vodafone, AT&T, MoviStar
- Big changes in **business model** with the introduction of **data transmission**
- Gap in the area of intelligent services, being addressed in the last few years
- Business models for **machine traffic** start to be needed
  - IoT, IIoY, M2M, industry 4.0.
- Transition from **terminals to objects and IoT devices**

Cloud RAN and Open RAN

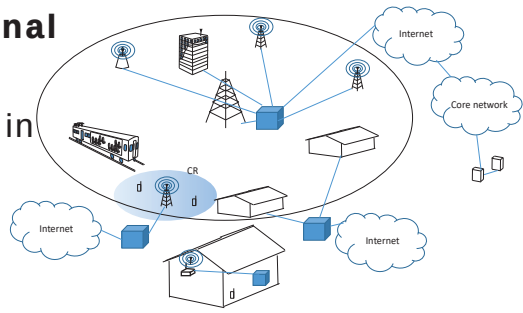
- Advanced **radio-over-fibre technology** can be used to relocate radio units from BSs to the central unit
- **BS complexity is reduced** considerably
- RoF allows **new cellular architectures**
- **Delay is introduced by RoF links**
- **Interference can be mitigated** for entire urban areas as compared to inside one cell
- **Load can be shared between sites**. Capacity can be adaptively allocated according to system needs or traffic load
- **OpenRAN**: Open BS specifications for fair competition and new manufacturer market





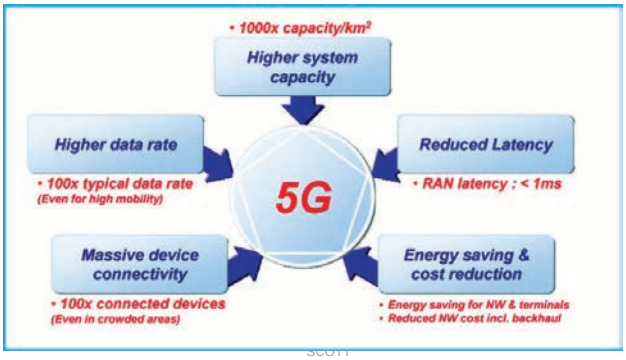
What is 5G?

- 5G is the **next generation of wireless cellular infrastructure**.
  - Convergence of cellular, wireless sensor, IoT, M2M and CPS systems
  - New standards for mobile communications are named as generations
- A generation is a set of complex **industrial technology standards** that lead to **business**, **marketing**, and multiple sets of **verticals or applications**
- Generations may differ from region to region
- They are endorsed by the technological leaders of the telecom industry, and they have **strategic regional and international importance**
- Generations are needed because there is a natural advance in technology that allows new features
  - **Features can be enabled by market need** or
  - **Features can enable market needs**



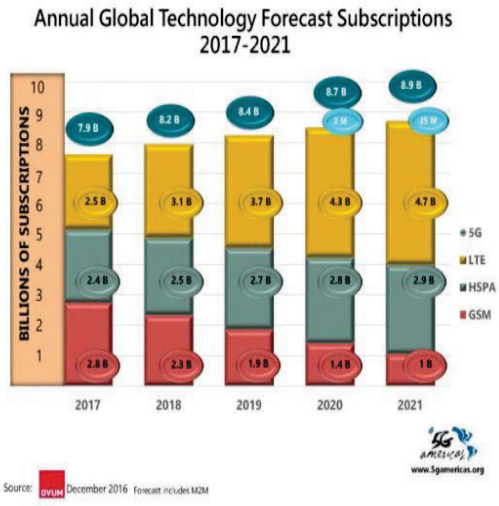
Objectives

- 5G has five main **objectives beyond 3G/4G systems**:
  - **higher capacity,**
  - **Higher data rate**
  - **ultra-low latency for machine-type traffic support, and**
  - **dense object connectivity demand (IoT or Internet of things)**
  - **Energy savings**



Motivations

- 50 billion devices will be connected to the cloud in 5-10 years time
- Industrial automation (M2M) and cyber-physical systems are expected to proliferate thanks to wireless pervasive connectivity (e.g., automated driving and structure health monitoring)
- Current cellular technologies cannot cope with the **scalability** of large numbers of "things" connected wirelessly with **ultra-low latency**
- 3G/4G solutions were designed for **human users**, not for machines.
- **WSNs cannot achieve cellular coverage** service for industrial IoT

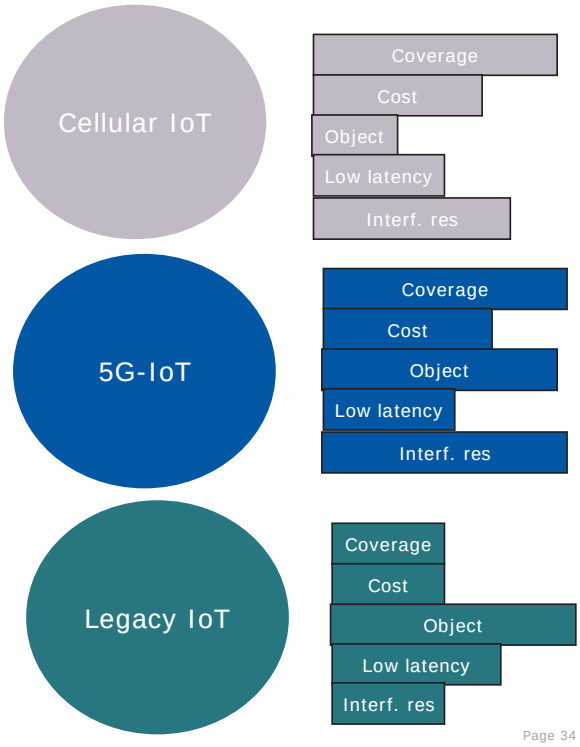


Is 5G bad for health?

- 5G will use a mmwave spectrum
  - Shorter wavelengths can have **higher impact on smaller objects and human organs**. However, the **mmwave signals decay really fast in space**.
  - **Cell sizes** are much smaller than mm waves. Cell mutation is not likely to occur.
  - 5G also uses **MIMO technology** that can potentially reduce Tx power, can also avoid body absorption.
  - 5G technology is also more **efficient in terms of power**
  - Mm wave spectrum is expected to be used in highly dense urban scenarios, not for home or indoor environments.
  - KEEP and EYE on the product: **exposure x power x frequency**
- **CONCLUSION**:
  - **More studies need to be conducted, but the issues raised by some groups seem to be not likely or with low probability**

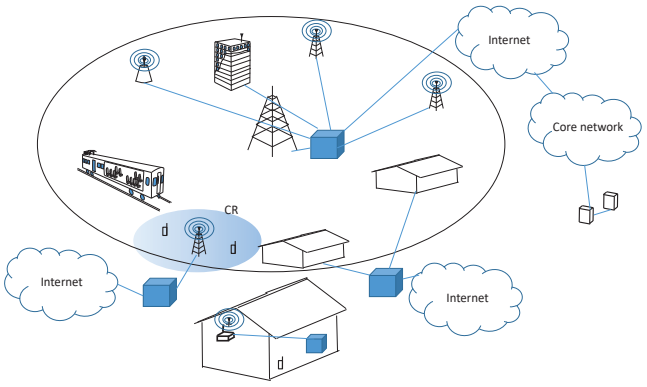
5G and IoT

- The advent of 5G means
  - objects can have a **direct access to the cloud / Edge**
  - **Lower cost per device**
  - **ultra low latency**,
  - **wider coverage than with other technologies**
  - Higher capacity,
  - higher scalability
  - Experience with mobility, handover, roaming, etc.
  - **Dedicated spectrum**



The security and privacy issue of 5G for IoT

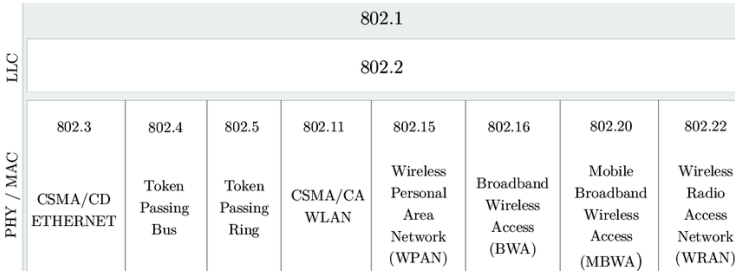
- One of the most attractive features of 5G is that objects can get a **direct connection to the cloud** via 5G.
- 5G links means that objects can have **multiple interfaces**, sometimes not completely controlled.
- **Information leakage** can be experienced when objects connect to multiple types of networks over different interfaces.
- This is particularly critical for 5G links that mimic an almost direct cloud connection.
- Most of the **IoT architectures do not have countermeasures to this new multiple link connectivity** problem.



- Critical infrastructure managed by 5G is at risk of network attacks

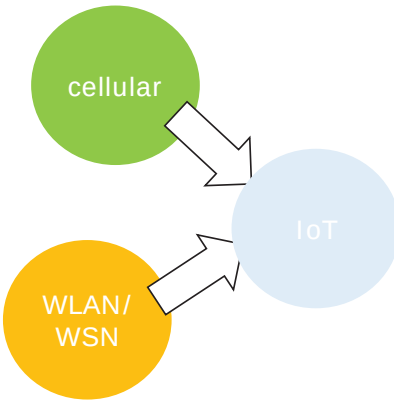
WSN/WLAN technology

- Multiple technologies used for **wireless sensor networks** and **local area networks**
- The main difference with normal wireless networks is the **resource and power constraints of Sensors**.
- They are also mainly **infrastructure-less solutions**, which reduces the performance at the expense of flexibility of applications.
- **IEEE802.x family** in many different flavors have been characteristic of this technology.
  - Original **802.11 based on CSMA/CD and in general ALOHA**, with low efficiency but great flexibility and complexity reduction
  - **Different flavors** eventually developed **fixed resource allocation or TDMA**.
  - **OFDM modulation eventually became the norm to deal with multipath**.
  - **UWB solutions** emerged to deal with complex scenarios and increased capacity



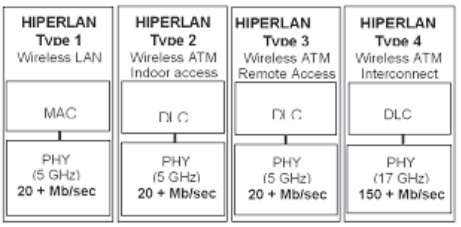
Cellular (5G) vs WSN/WLAN (802.xx)

- Cellular started as **circuit switched voice service** slowly **evolving to data using resource reservation**
- In recent approaches, **collision-based regimes are proposed to deal with ultra-low latency**
- Cellular has the advantage of experience with **handover, macro mobilty, spectrum licensing, etc**
- By comparison **WSN / WLAN** started mainly as **packet connection-les technology**, based on **simple ALOHA or collision-based** technology.
- Eventually it evolved to **ultra-low latency, targeting reserved spectrum as well, and covering some mobility scenarios for transportation or vehicular networks**.
- Competition between them, but some areas are complementary rather tan competitive**. 5G still has the advantage of dedicated spectrum, cellular mobility, and city plan design for sites**.
- However, 802.xx technology have good user acceptance, they are trying to address the issues and they are adapting faster to the needs of technology to Support ultra low latency applications



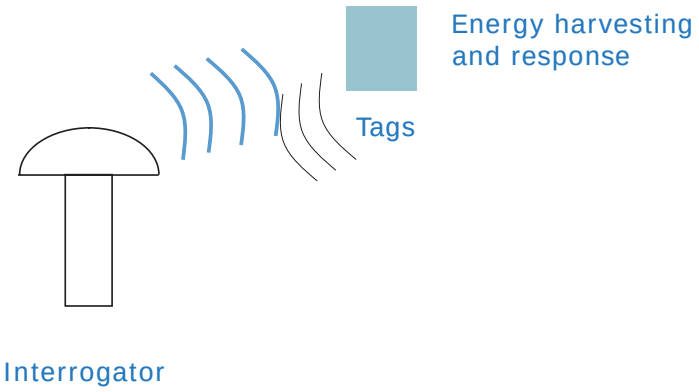
802.11 vs HiperLAN

- Standardization efforts from America and Europe collided with the proposal of **WLAN technology**
- **802.11 presented a solution based on ALOHA-type collision** resolution, which is flexible, simple and low cost. This is at the expense of channel efficiency.
- **HiperLAN** was a complex standard with highly efficient resource allocation and high channel efficiency.
- The state of the market dictated a **win for 802.11**
- This decision was correct at the time of a market not used to efficiency issues. We were looking for simple connectivity.



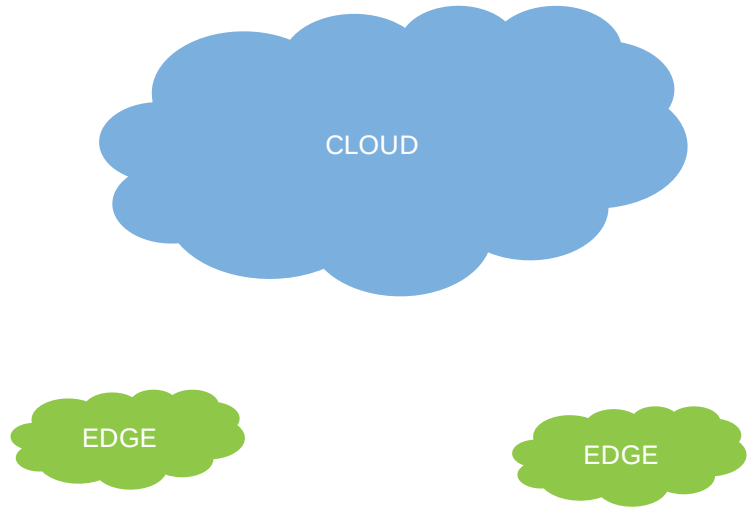
RFID (Radio Frequency Identification)

- Based on old radar **friend and foe detection technology**
- Evolved towards **energy harvesting** networks
- Basis of object identification for supermarket **thief detection**
- Evolution to **complex identification strings**, and some **processing capabilities.**
- Today, RFID comes in many types of systems passive, active, semi active, at different frequencies, and for different functionalities from access control, pet control, supply chain management, remote sensing, etc.



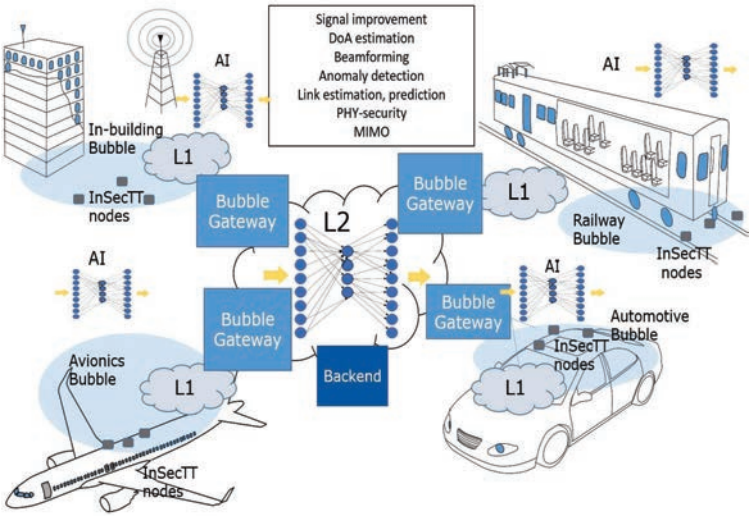
Edge vs Cloud computing

- Central servers can host **immense processing capabilities**. This means objects and terminals can off load many of their functionalities to the cloud. This can reduce costs and improve scalability of IoT
- However, **industrial critical applications can suffer high latency**
- In addition, **security issues can arise in centralized locations**, for example denial of service attacks
- **Edge processing** aims to bring some of the centralized cloud processing capabilities back to the edge of the network, in a **tradeoff between latency, cost reduction and security**



Artificial Intelligence and IoT

- **Artificial intelligence** is experiencing a boom in the last few years. Ability to learn from existing data and predict afterwards.
- It matches perfectly the **IoT as a source of data to be processed in the cloud or Edge**
- **Machine learning** can be used to refine, detect, **predict events** based on the collected information from objects and embedded processors.
- The fusion of AI and IoT, also called **AIoT, is one of the main technologies that will boom in the coming years**. It promises to bring IoT to a new level. For example: critical industrial applications





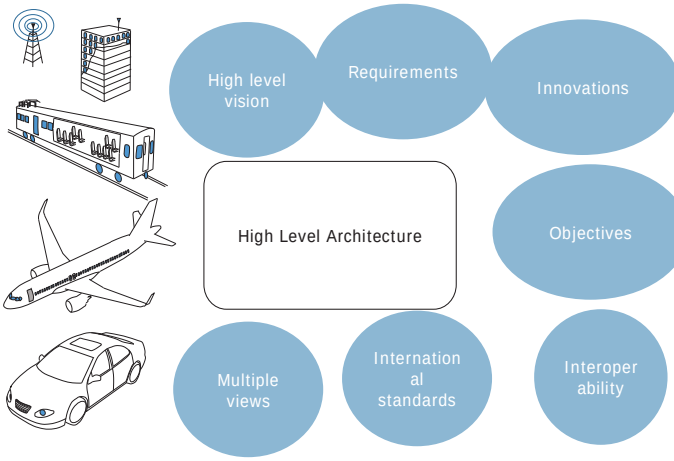
Intelligent Secure Trustable Things

Reference architecture

InSecTT has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey

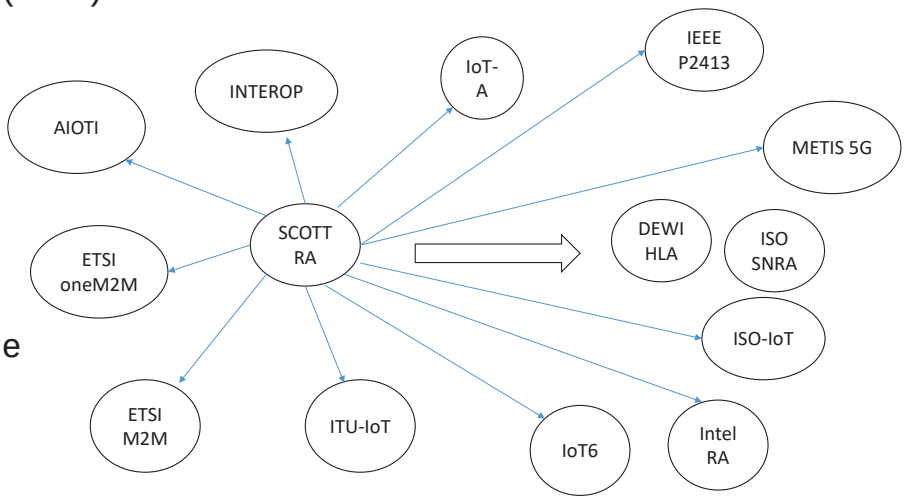
What is a reference architecture?

- **A generic architecture useful for design particular instances and use cases aligned with international standards and guidelines**
- A **high-level overview** for infrastructure design and application overview
- Not a simple scheme of boxes anymore
- Modern reference architectures are a **complex collection of views or perspectives**, each one providing particular insights that **highlight different features for different stakeholders**
- Reference architecture can be the basis for **alignment, design, validation, verification standardization and certification** of modern IoT systems.



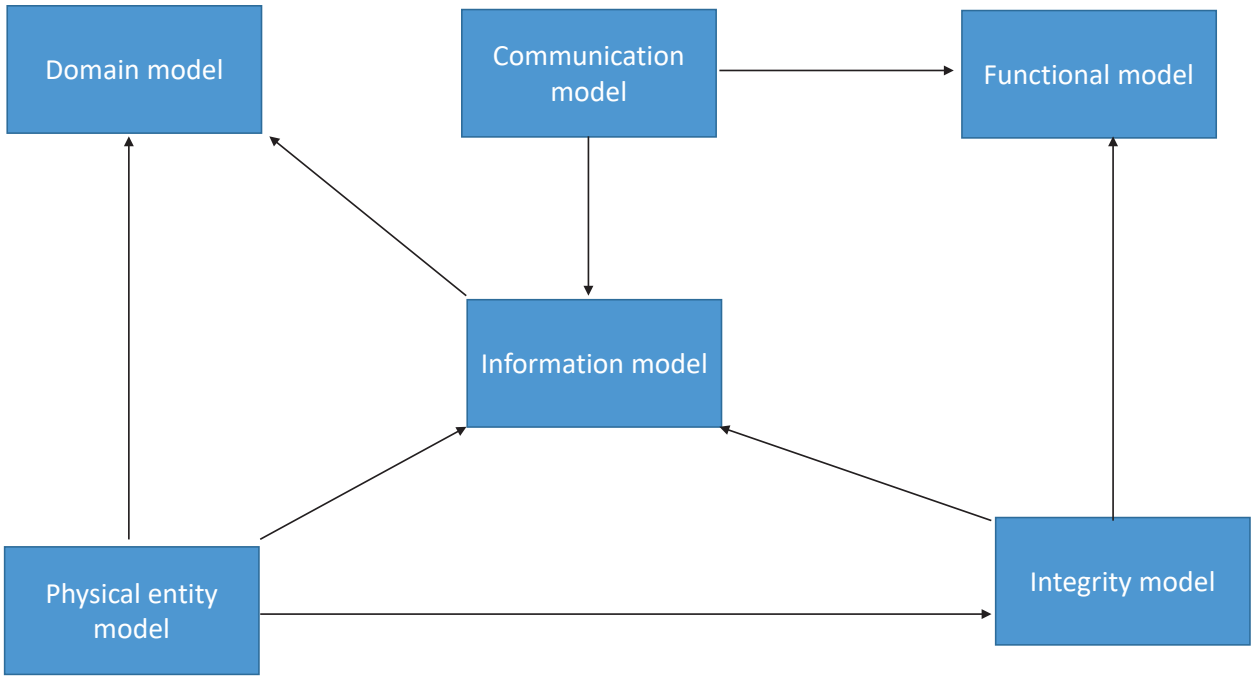
Modern IoT architecture standards

- IoT Architecture reference model (ARM)
- IEEE IoT architecture
- ISO reference architecture
- ITU reference architecture
- ETSI M2M architecture
- AIOTI architecture
- DEWI/SCOTT/InSecTT architecture



ITU: International Telecommunications Union
ISO: International Standard Organization
ETSI: European Telecommunication Standards Institute
IEEE: Institute of Electronic and Electrical Engineers
AIOTI: Alliance for IoT Innovation
M2M: Machine to Machine
SNRA: Sensor Network reference Architecture

ISO views of architecture

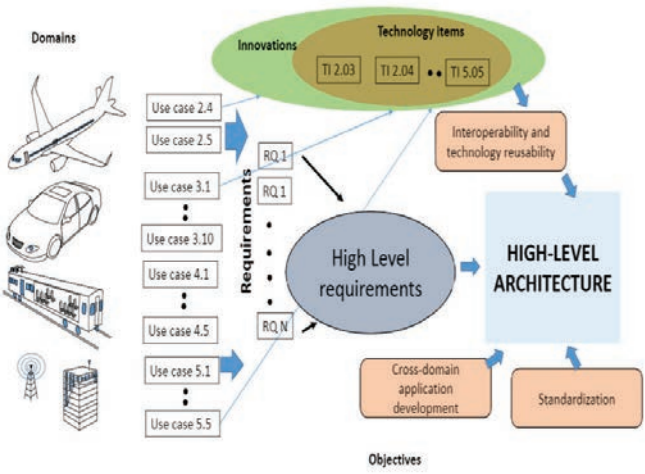


DEWI/SCOTT/InSecTT projects

- **DEWI (Dependable Embedded Wireless Infrastructure)**
  - Dependability of sensor networks
  - Interoperability
  - 21 use cases, 57 partners
- **SCOTT (Secure Connected Trustable Things)**
  - Security, privacy, trust
  - Internet of Things
  - 15 use cases, 57 partners
- **InSecTT (Intelligent Secure Connected Trustable Things)**
  - Artificial Intelligence , Edge Computing
  - Internet of Things
  - 15 use cases, 57 partners

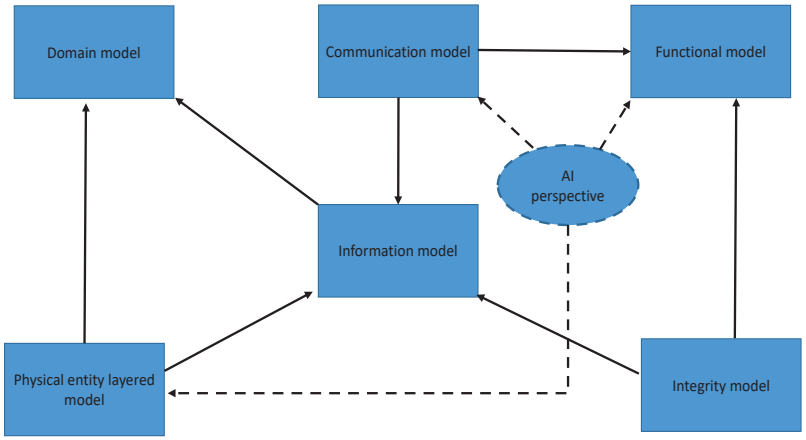
DEWI/SCOTT/InSecTT Reference architecture

- **Definition**: "A set of guidelines for infrastructure organization of IoT use cases supporting the objectives of the projects"
- The framework for a **high-level analysis** of all building blocks of use cases in different industrial domains
- **Interface** and **vulnerability** analysis per layer and entity.
- Framework for **reusability** and **cross-domain** interpretation
- High level perspective of use case requirements, road-map, and forecast analysis
- Compilation of **expertise** accumulated across different **use cases** in different **industrial domains**.
- Framework for **standardization** needs in detail (forensic analysis)



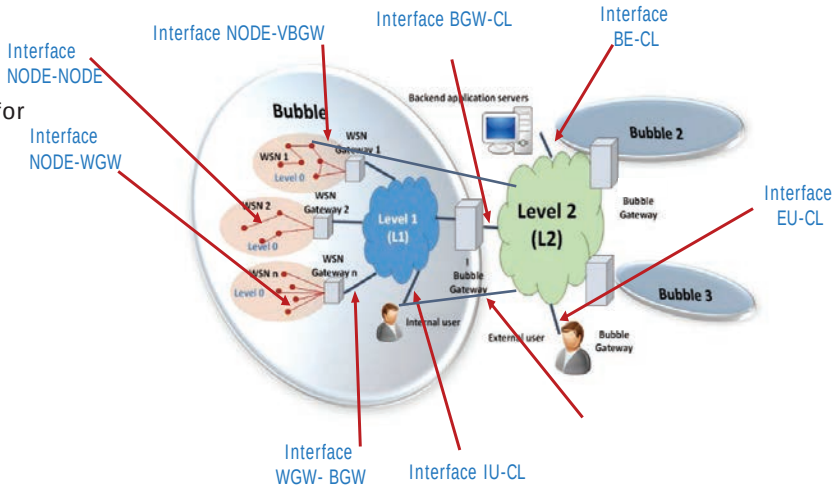
INSECTT Reference architecture

- The **INSECTT Reference architecture** consists of multiple views or perspectives of a generic IoT system
- The multiple views approach is useful for modern IoT use cases with multiple stakeholders
- The INSECTT RA consists of
  - Entity model
  - Functionality Model
  - Information Model
  - Domain Model
  - Communication model
  - Ontology model
  - AI perspective?



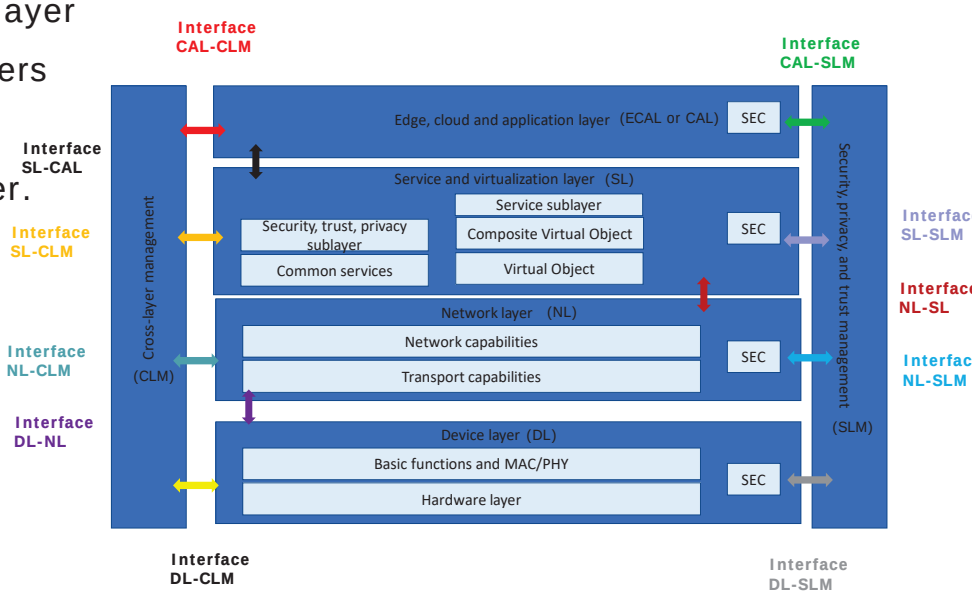
Entity model

- **Bubble** is a high-level WSN with a unique gateway (Bubble Gateway), controlling wireless and wireline infrastructure.
- Attributes:
  - Interoperability (single protocol or semantics model for interoperability)
  - Integration of new and legacy critical industrial sensors to a modern IoT infrastructure
- Three-level organization
  - **L0 Wireless**
    - Nodes and WSN Gateway
  - **L1 wireline- existing critical infrastructure**
    - For example: aeronautical internal bus, CAN bus
  - **L2 interoperability**
    - Cloud, Edge servers. The Bubble Gateway can also act as fog or Edge server.



Functional model

- Hybrid model combining **ISO, ITU, AIOTI, SNRA, and AIOTI**
- Security functionalities on each layer
- Software interfaces between layers
- HW interfaces between entities
- Trustworthiness metrics per layer.



The Bubble and high-level architecture evolution

Based on ISO/SNRA
Interoperability ETSI M2M, IoT-ARM
L0/L1/L2 layering for Wireless/wireline
Model

Dependability inside the bubble
Integration Wireless/wireline industrial WSN and IoT
Cross-domain reusability
Interoperability
Integrated sensors into IoT

Sensors

Full IoT architecture (around the bubble)
Hybrid ISO SNRA ITU, ISO, AIOTI, IEEE IoT architectures
L0/L1/L2 layering for Wireless/wireline
Security sublayers and processes

Dependability inside the bubble
Integration Wireless/wireline industrial WSN and IoT
Cross-domain reusability
Interoperability
Integrated sensors into IoT
Trustworthiness and security metrics
Bubble gateway as Edge processor
Inter-bubble communications based on trust indicator
Blockchain compatibility

Security IoT

Full IoT architecture (around the bubble)
Hybrid ISO SNRA ITU, ISO, AIOTI, IEEE IoT architectures
L0/L1/L2 layering for Wireless/wireline
Security sublayers and processes
Specific AI models and impact analysis

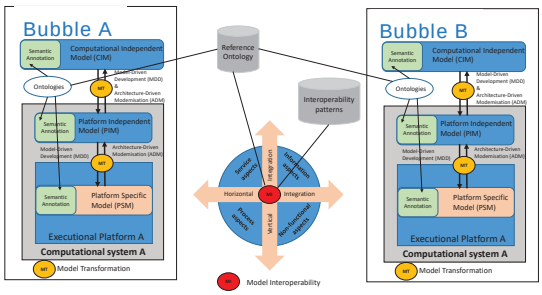
Virtualized Bubble
Multiple connections inside the Bubble
Long and short-range communications
Direct cloud connections inside the bubble and for internal users

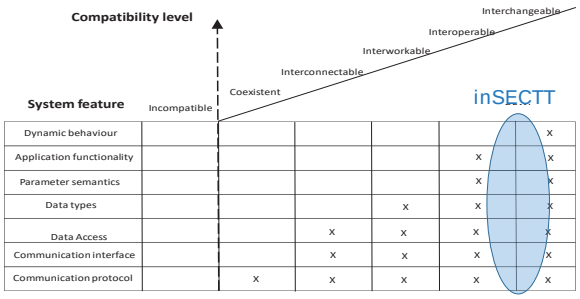
AIoT

InSECTT Bubble

- InSECTT envisions an **Internet of Bubbles**, each bubble encapsulating legacy or state of the art industrial technologies (wireless and wireline) for sensor and actuators.
  - The **encapsulation enforces dependability and security inside each domain network.**
- **Interoperability in SCOTT**: **Re-use of building-blocks, Co-operability in wireless context, and Co-existence with other wireless standards. (Mapping to the IEC concept of interoperability)**
- **Model driven design** helps us introduce interoperability together with non-functional requirements such as security, privacy and trustiness in the Bubble
- **Ontologies at the three different modeling levels** of the INTEROP architecture match the Bubble infrastructure and the SCOTT framework for security, privacy and trustiness

INTEROP perspective of Bubbles





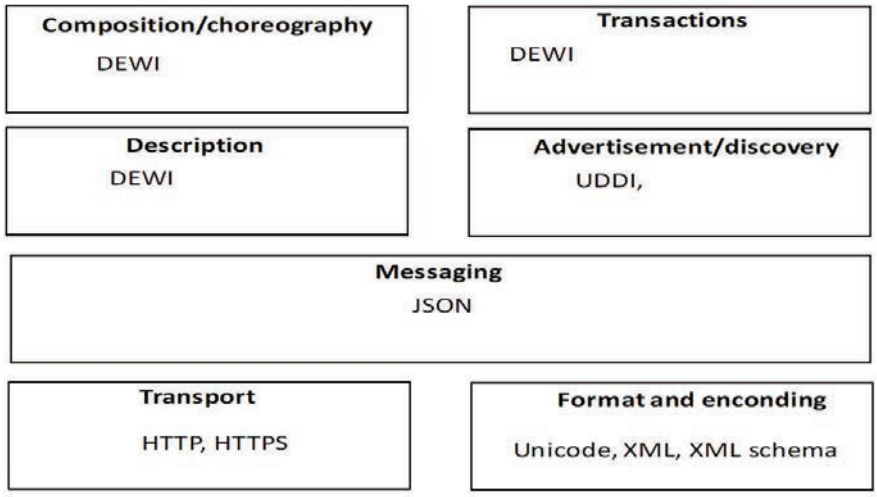
IEC interoperability mapping
IEC: International Electrotechnical Comitte

What do you get by following the Bubble specs?

- **Guidelines** to achieve **dependability, security, safety, privacy and trustworthiness** inside the bubble
- Specific measures for interaction between **Wireless and wireline infrastructure with real time constraints**
- **Cross-domain interoperability**
- **International IoT standards compatibility**
- **Trustworthiness multi-metrics evaluation** (extension of ARMOUR and SCOTT metrics)
- **Privacy and trustworthiness by design** approach
- Collected experience of real **industrial use cases**
- Per layer, per interface and per entity **trustworthiness metrics analysis**
  - **Extension or ARMOUR and ETSI metrics (CWSS, CVSS)**
- Integrated **trust methodology** to include end user and stakeholder perspective

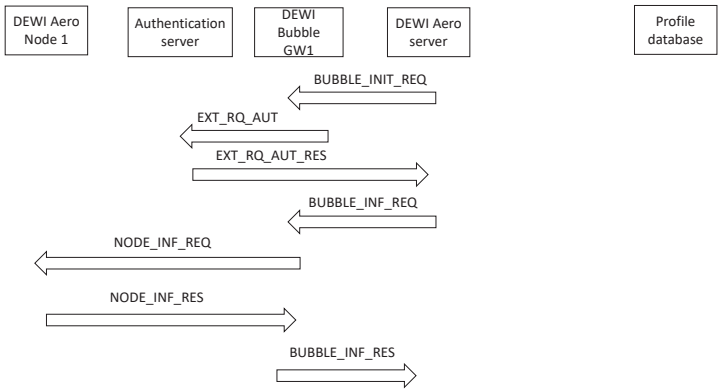
Information model

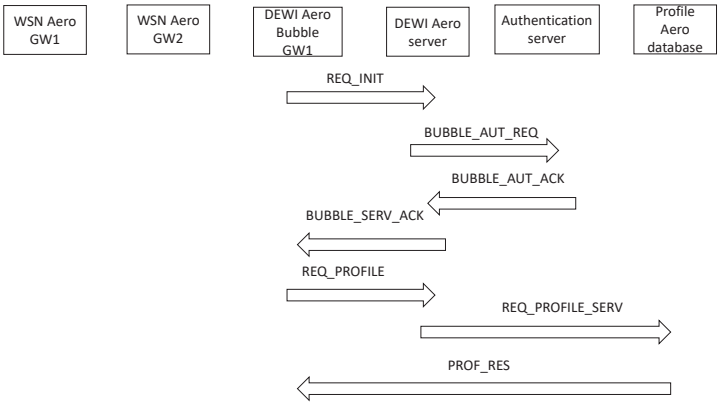
- **The information model describes how the information between entities or layers is exchanged, described, organized, transported, etc.**
- **Format/ encoding**
- **Transport.** How to send the information to the destinations
- **Messaging.** How information will be understood by the receiver
- **Description.** How resources are described
- **Advertisement.** How resources are advertised
- **Composition. Business layer**
- **Transactions**



Communication model

- The description of communication protocols between the entities and functionalities of the architecture

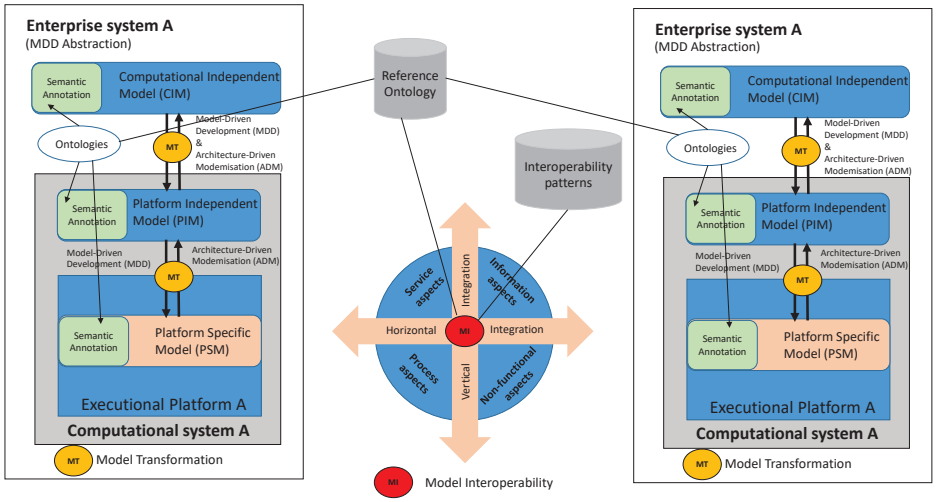




Reference model for conceptual integration (model driven)

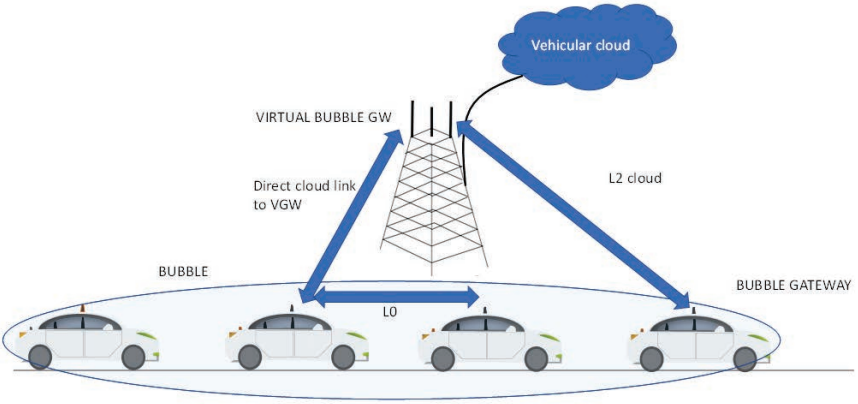
- **Model driven architectures for interoperability** allow us to introduce security solutions in the interaction between entities or layers inside the same entity
- The **INTEROP project** proposed the model driven interoperability view with a three – layer ontology approach to catch the different stages of model development.

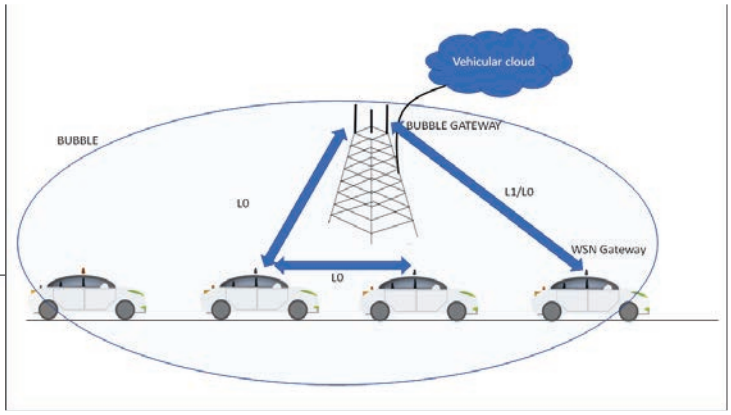
- **CIM (Computational Independent Model)**
- **PIM (Platform Independent Model)**
- **PSM (Platform Specific Model)**



Example Vehicle platoon

- In a vehicle platoon, different elements can be assigned to different responsibilities according to the Bubble model:
  - Option 1: BS as Edge BGW
  - Option 2: Leader as EBGW



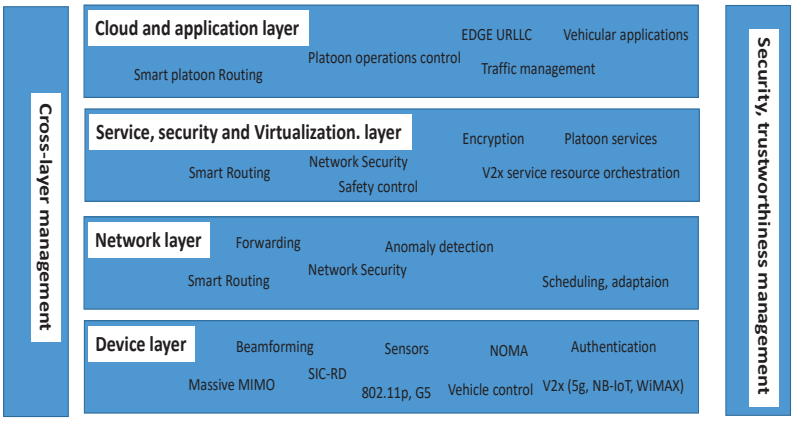


- Two types of links are recognized: V2V and V2I

Example vehicle platoon

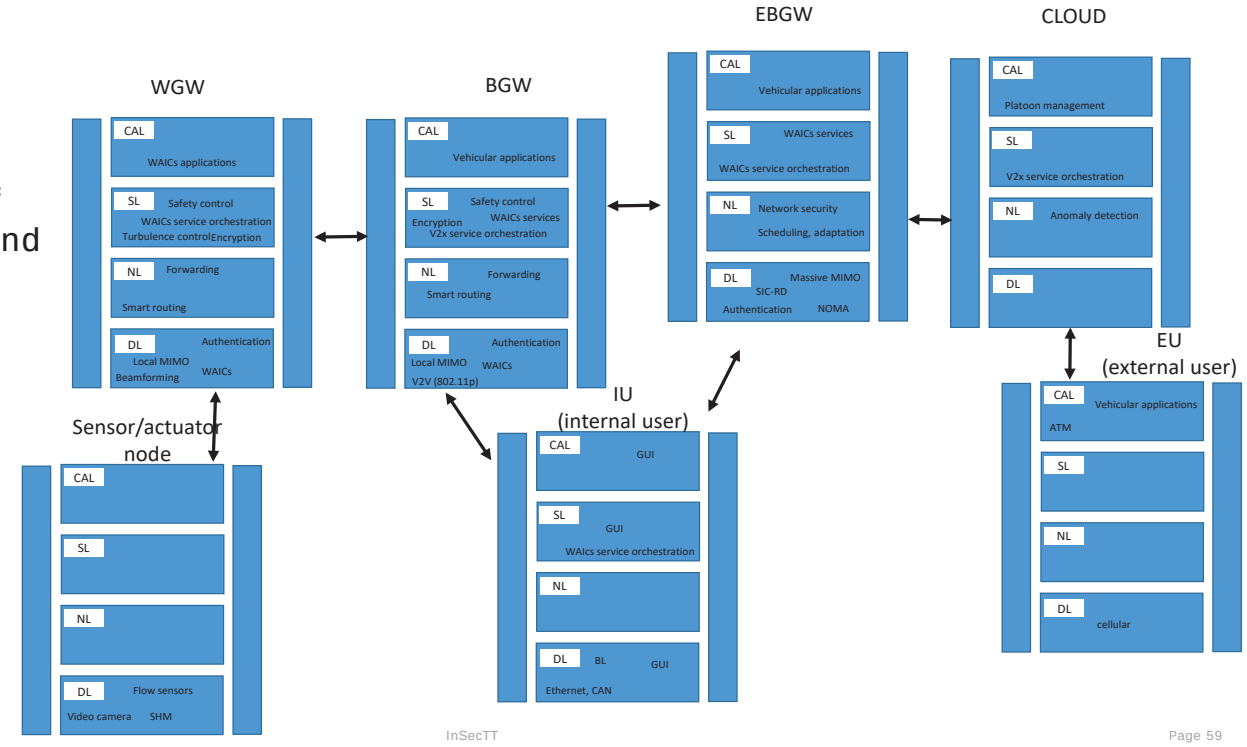
- The different functionalities can be arranged according to the functionality layered model.
- Interfaces between different functionalities or groups of functionalities can be analyzed in a preliminary fashion

- Lower layers are focused on V2V and V2I transmission using multiple antennas. 5G and 802.11p are allowed
- Upper layers focus on resource management and platoon management
- Security features are incorporated to ensure the integrity of the platoon



Example vehicle platoon

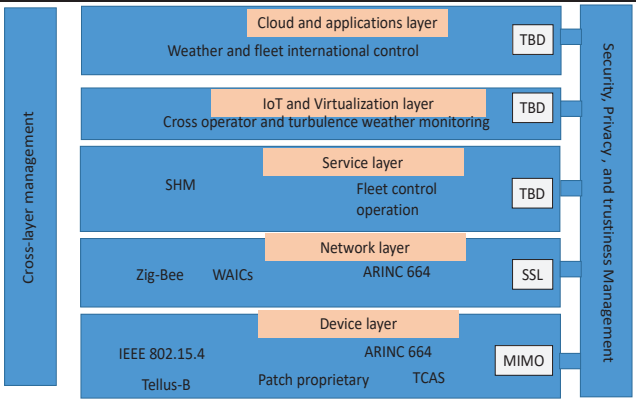
- Hybrid view
- Entity vs functionality.
- Detailed map of functionalities and interfaces

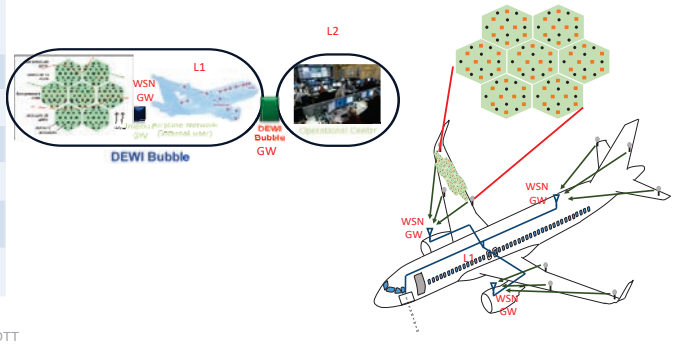


Alignment of use cases

- Example of alignment of an industrial use case (aeronautics)
- Dense sensor flow control
- **Entity, functionality, entity vs. functionality**

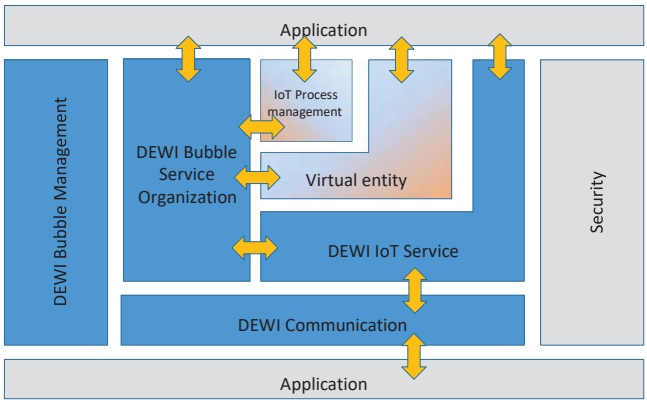
		L0 device to/from L0 GW		L0 Device to/from L0 Device		L0 GW to/from L1 GW		Bubble L1 to/from Internal User		Bubble GW to/from Service provider/Cloud	
Cloud and applications layer		-	N/A	-	-	N/A	-	Operator fleet control	-	Operator fleet control	Encryption
IoT Virtualization layer		-	N/A	-	-	WAIC server	-	Avionics layer	-	Avionics layer	PHY-access
Service layer	Security, trustability and privacy	SSL	SSL	-	-	SSL	-	SSL	-	SSL	-
	Common services	Flow control	HTTPS	-	-	WAICs	-	WAICs	-	HTTPS	-
Network layer	Transport	UDP	SSL	-	-	VL	-	VL	-	HTTP	-
	Network	ZigBee	Encryption	-	-	ARINC 664	-	ARINC 664	-	IP	-
Device layer	Basic functions and MAC/PHY layers	IEEE 804.15.4	MIMO-based	-	-	ARINC 664	-	ARINC 664	-	Ethernet	TBD
	Hardware layer	Pressure sensors, micropumps, TTP	Compression	-	-	ARINC 664	-	ARINC 664	-	Ethernet	-

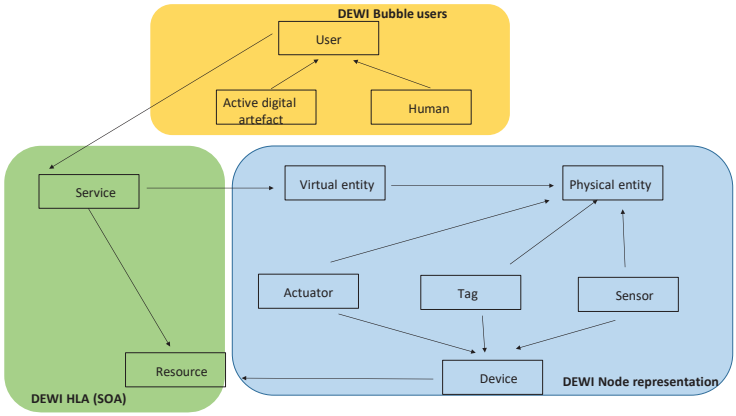




IoT-A reference architecture

- One of the first IoT reference architectures
- **Perhaps the most complete domain model**
- **Virtualization of entities**

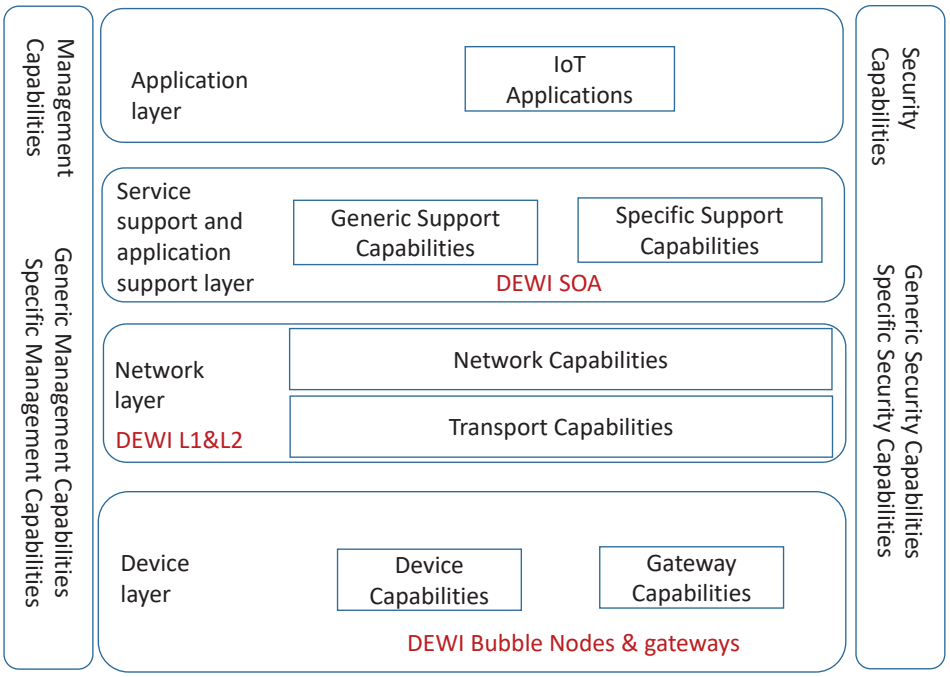




SCOTT

ITU Reference architecture

- **Layered functionality perspective**
- **Management support**
- **Security support**



AIOTI (Alliance for the Internet of Things Innovation)

- **Simplified layered functionality architecture**
- **Compatibility with the rest of architectures**

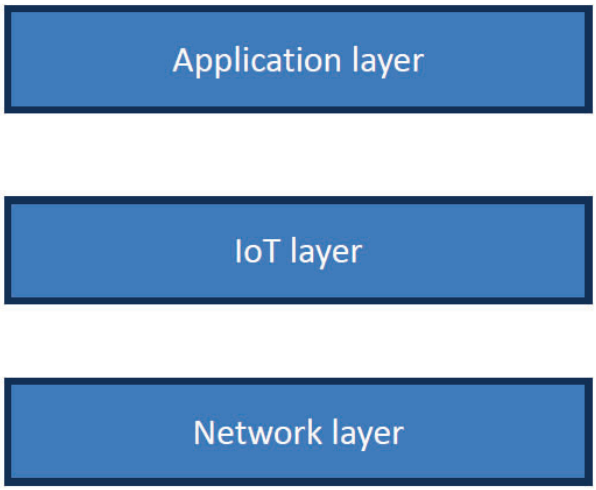


Figure 3: AIOTI three layer functional model.



Intelligent Secure Trustable Things

Technology analysis based on the architecture

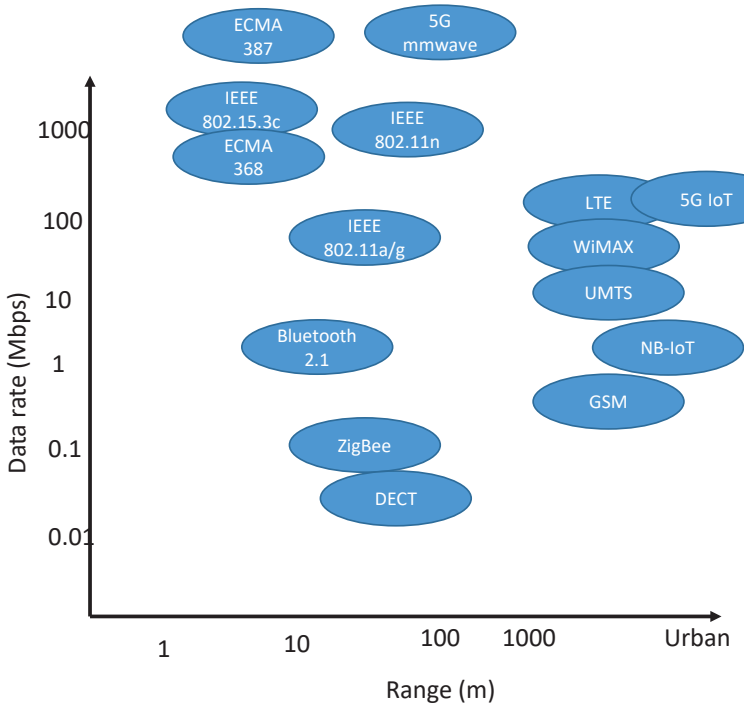


InSecTT has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey



Level 0 interface technologies

- **Multiple L0 technologies** exist in the market with different properties.
- **Data rate versus coverage distance** is one of the main evaluation criteria
- Other criteria include **scalability, latency, power consumption**, etc.
- Complex decision making might be necessary to **select the best interface**
- These L0 technologies are usually associated to other functionalities and standards of the architecture
- Tendency to cover **higher speeds** over wider ranges



Level 0 technologies

- WIMAX. Worldwide Interoperability for Microwave Access is a family of wireless broadband communication standards based on the IEEE 802.16 set of standards, which provide physical layer and media access control options.
- LTE. Long-Term Evolution is a standard for wireless broadband communication for mobile devices and data terminals, based on the GSM/EDGE and UMTS/HSPA standards.
- The Global System for Mobile Communications (GSM) is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe the protocols for second-generation (2G) digital cellular networks used by mobile devices such as mobile phones and tablets
- Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios

Level 0 technologies

- ECMA 387. This standard specifies PHY, MAC and PALs for flexible and heterogeneous multi-Gigabit Wireless Personal.
- IEEE 802.11p is an approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments, a vehicular communication system..

PHY-layer requirements per industrial domain

- Each domain has PHY-layer properties that must be matched to the L0 technology to be used.
- The environment determines several of the parameters to select the most appropriate transmission technology.

- Industrial domains present different physical features that determine propagation issues such as multipath, path-loss, time variations

Feature/domain	Aeronautical		Automotive		Rail		Building	
	Inside	Outside	Inside	Outside	Inside	Outside	Inside	Outside
Multipath (frequency selective)	Dense	Not dense	Dense	Dense/regular	Dense	Not dense	Dense	Dense/ regular
Fast fading	No	Maybe	No	Maybe	No	Maybe	No	No
Path loss Exponent	Low	High	Low	High	Low	High	High	Low
Interference sensitivity	High	High	High	High	High	High	Medium	High
Interference to other systems	High	Low	High	Medium	High	Low	High	High
MIMO feasibility	Yes	Yes	Maybe	Yes	Yes	Maybe	Yes	Yes

PHY-layer features

- Each technology can be selected to match the particular needs of each use case.
- ~

Feature/technology	802.15.4			802.11		ECMA standards		802.15.3	Bluetooth	
	ZigBee	ISA	15.4c	11.a,b,g	11n,	368	387	15.3	15.1	BLE
Multipath immunity	Regular	Regular	High	Regular	Regular	High	High	High	Regular	Regular
Fast fading immunity	Low	Regular	Very High	Low	Low	Very high	High	Very high		
Modulation /frequency(GHz)	DSCDMA (2.4)	DSCDMA (2.4)	UWB (3-6)	OFDM (2.4)	OFDM (5)	UWB	OFDM (60)	UWB (60)	FHSS (2.4)	FHSS (2.4)
Range	10-100m	10-100	10-100	10-100m	10-100m	10 m	5-10 m	1-10m	10-50m	10-20m
Jamming Interference sensitivity	Low	Low	Very low	High	High	Low	Low	Low	High	High
Interference to other systems	Low	Low	Very low	High	High	Low	Low	High	High	High
Availability	Very good	Good	Likely	Very good	Very good	Good	?	Likely	Very good	Very good
Other Signal characteristics	NLOS	NLOS	NLOS	NLOS	Indoor Low resources	Indoor high resource	LOS	LOS	NLOS	LOS
Cost	Medium	High	Very high	Low	High	High	High	High	Low	Low
Traffic density (Mbps/S/m2)	?	?	?	?	1.4	2.4	?	16	?	?

MAC – layer requirements

- Each domain has MAC-layer properties that must be matched to the L0 technology to be used.
- The environment determines several of the parameters to select the most appropriate transmission technology.

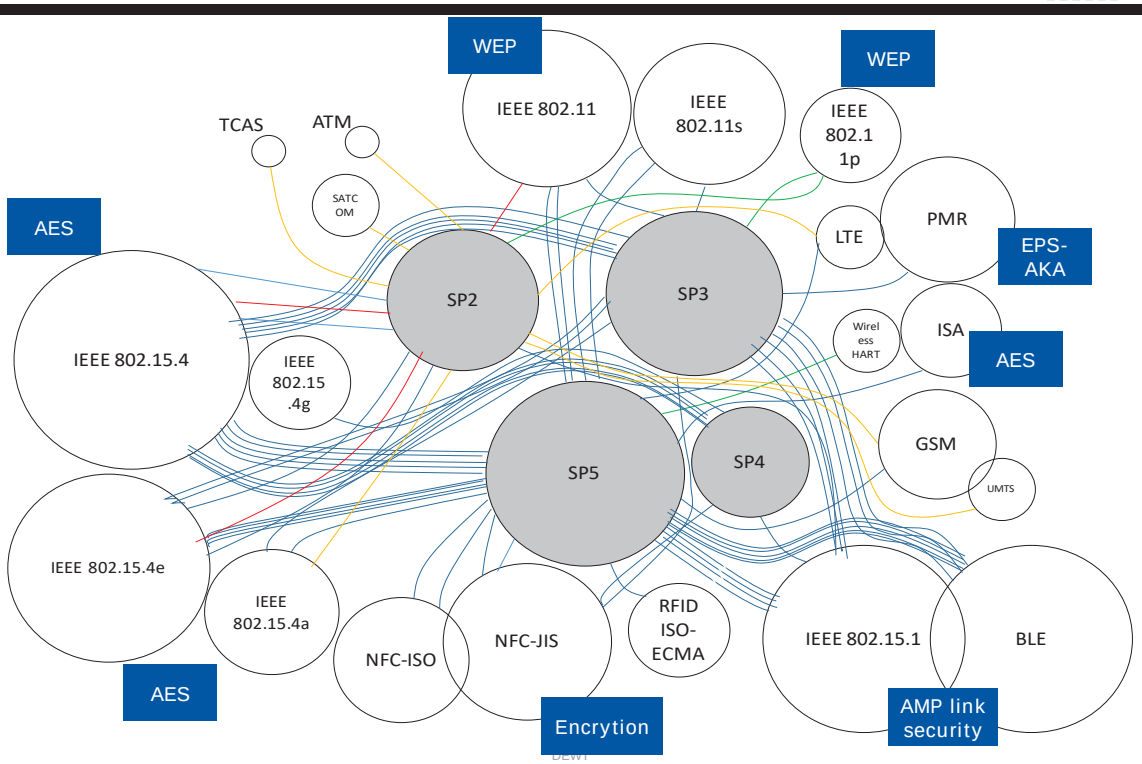
Feature/domain	Aeronautical		Automotive		Rail		Building	
	Inside	Outside	Inside	Outside	Inside	Outside	Inside	Outside
Expected Data rate (depends on application) medium term	Expected medium-high	Expected Low-medium	Expected medium-high	Expected medium-high	Medium	Medium	High	Medium
Contention based/centralized	Both	Contention	Centralized	Contention	Centralized	Contention adhoc	Both	Both
Density of nodes	Medium to high	Low to medium (high for SHM)	High	Medium	Medium	Medium	High	Medium
Resources needed	High	High	Medium	High	High	Medium	High	Low

MAC-layer features (COTS technologies)

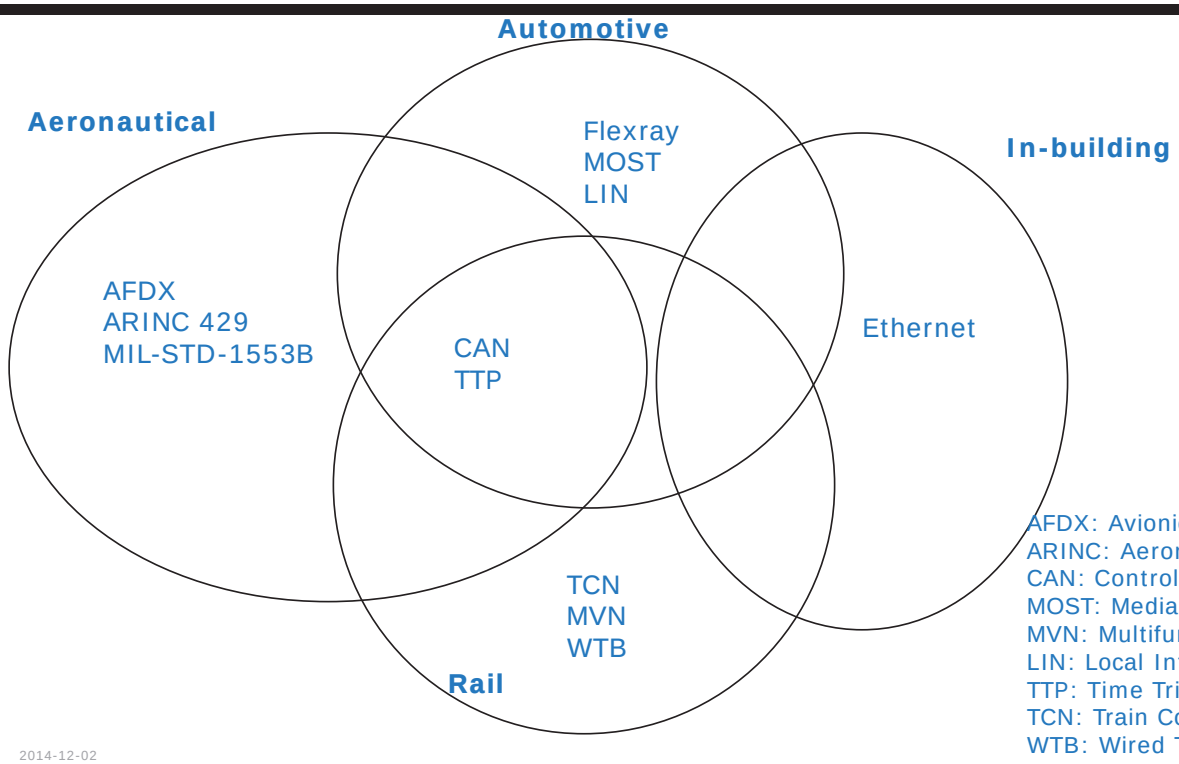
- Each technology can be selected to match the particular needs of each use case

Feature/ technology	802.15.4			802.11		ECMA standards		802.15.3	Bluetooth	
	ZigBee	ISA	15.4c	11.a,b,g	11n	368	387	15.3	15.1	BLE
Access	CSMA	CSMA	CSMA	CSMA	CSMA	FD/TDMA	FD/TDMA	CSMA	Master /slave	Master /slave
Discovery	Beacon	Beacon	Beacon	Beacon	Beacon	?	?	Beacon	On demand	On demand
Scalability	10-10000	10-10000	10-10000	10-100	10-100	?	?	?	10-10000	10-10000
Frame length	15ms	15ms	15ms	variable	10-100m	10 m	?	?	??	?
Multi-hop	Yes	Yes	Yes	Weak	Weak	?	?	Yes	No	No
Self configurability	Yes	Yes	No	Weak	Weak	?	?	?	Yes	Yes

Wireless standards mapping

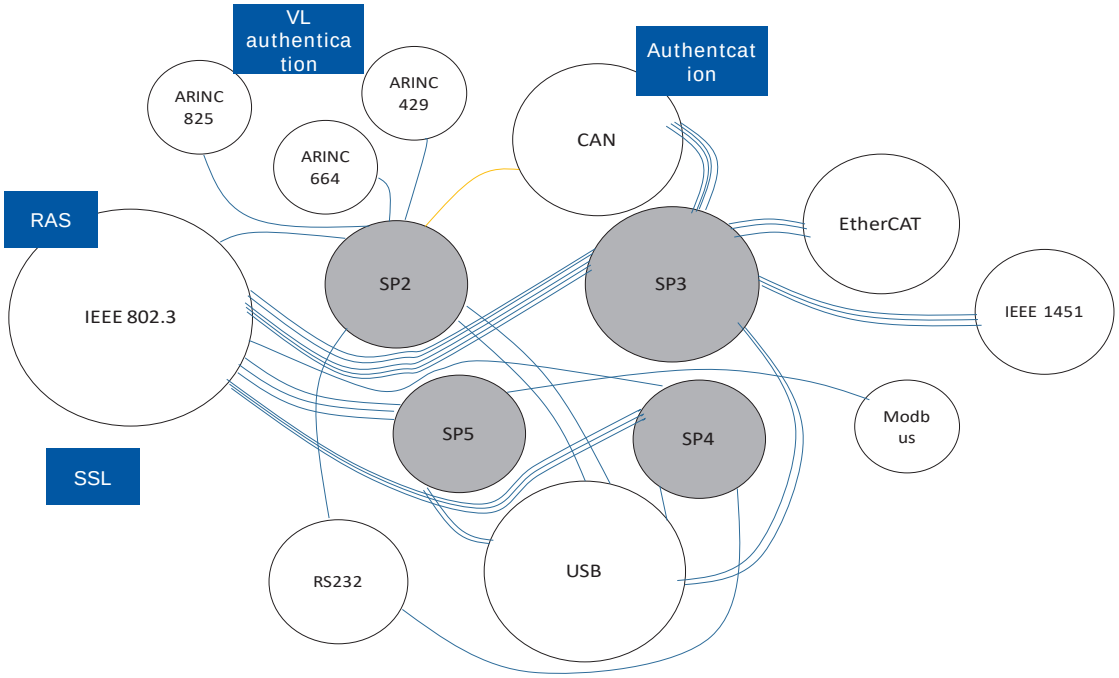


Existing infrastructure (buses)

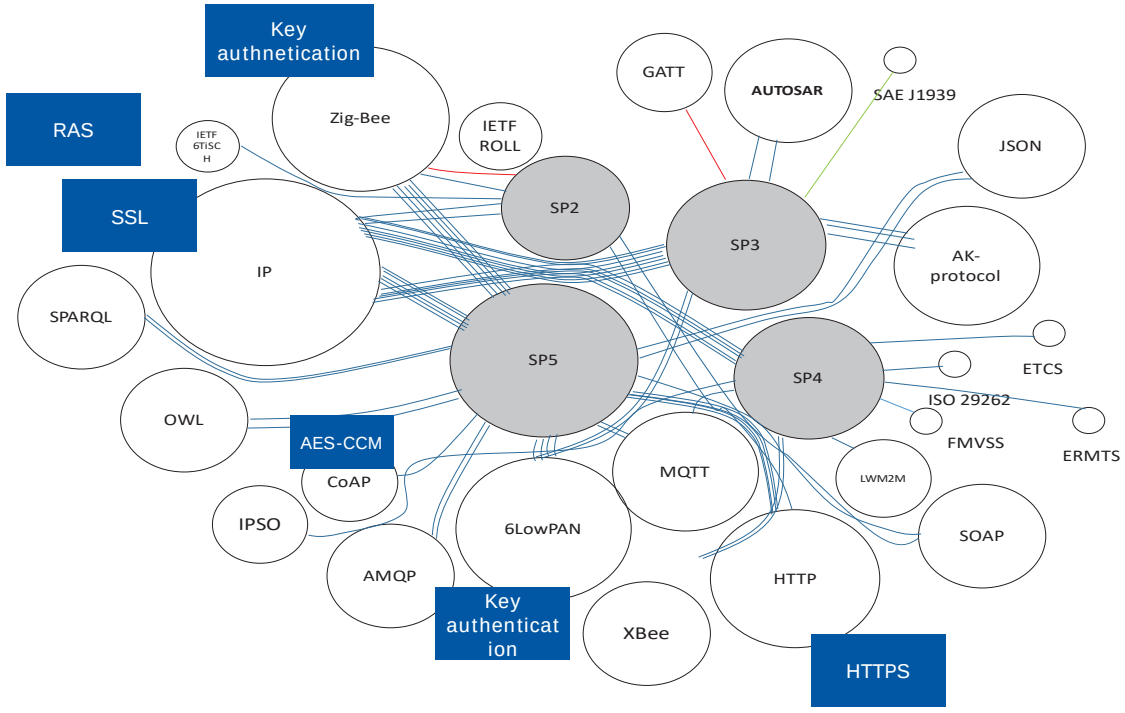


AFDX: Avionics Full-Duplex Switched Ethernet
ARINC: Aeronautical Radio, Incorporated
CAN: Controller Area Network
MOST: Media Oriented Systems Transport
MVN: Multifunction Vehicle Network
LIN: Local Interconnect Network
TTP: Time Triggered Protocol
TCN: Train Communication Network
WTB: Wired Train Bus

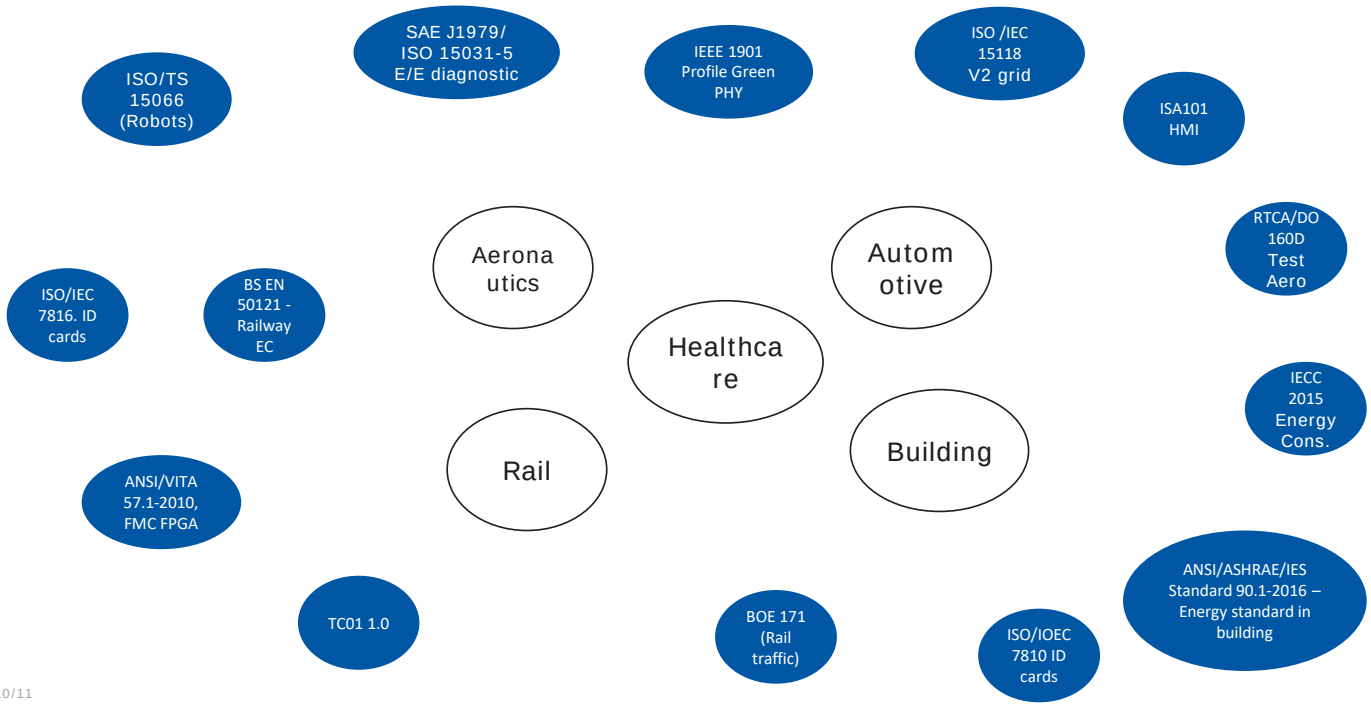
Wireline standards



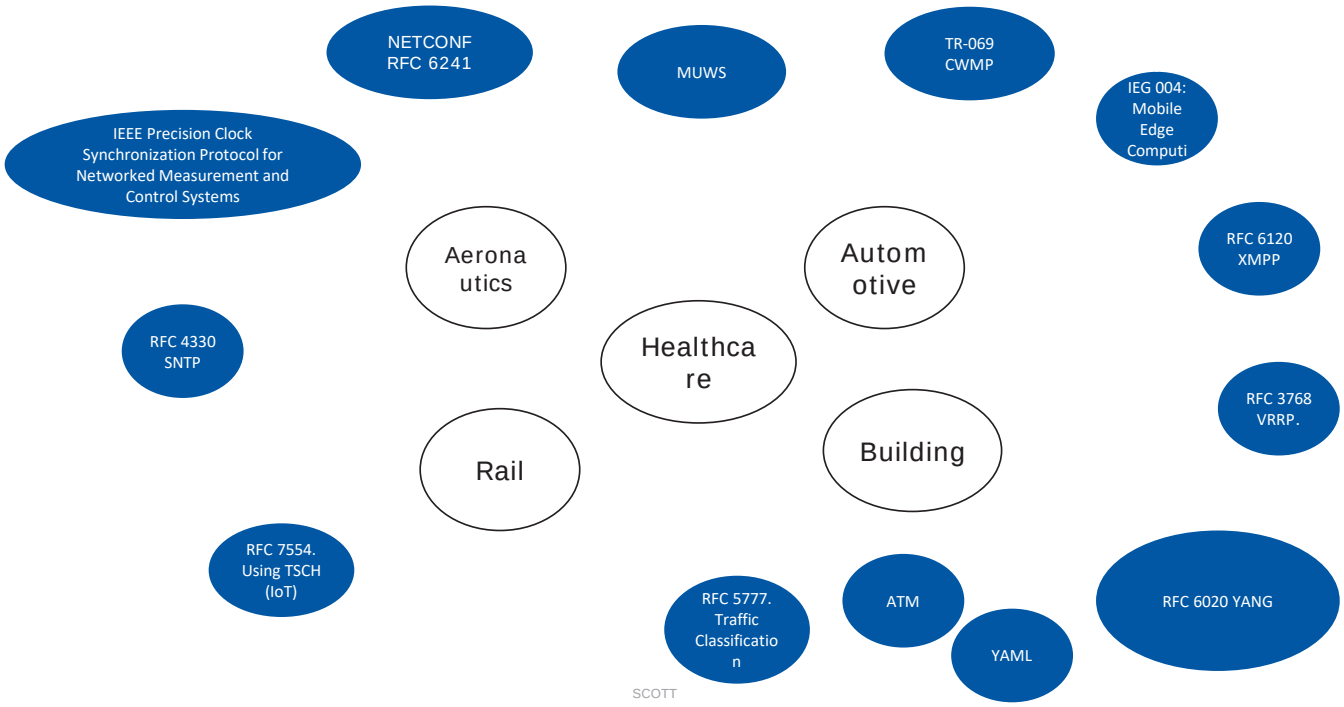
Higher layer standards



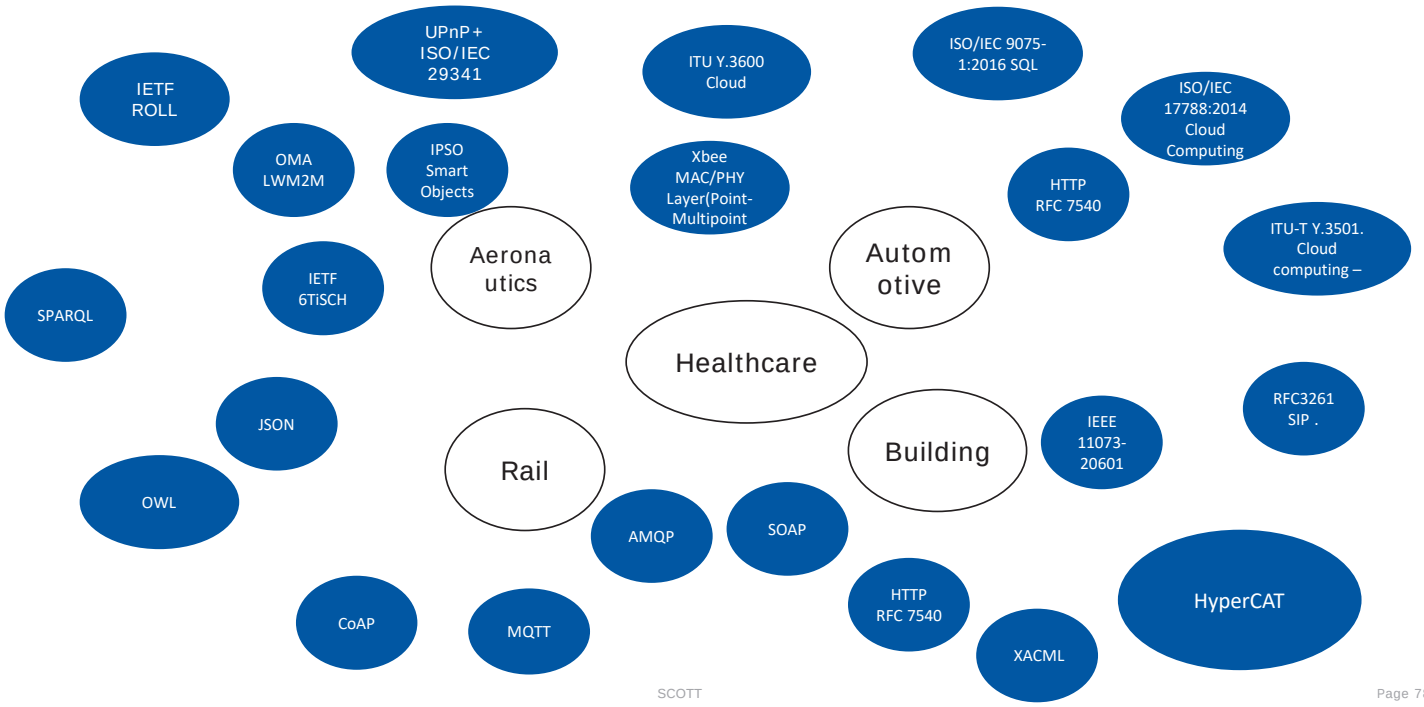
Device layer standards



Network and service layer standards

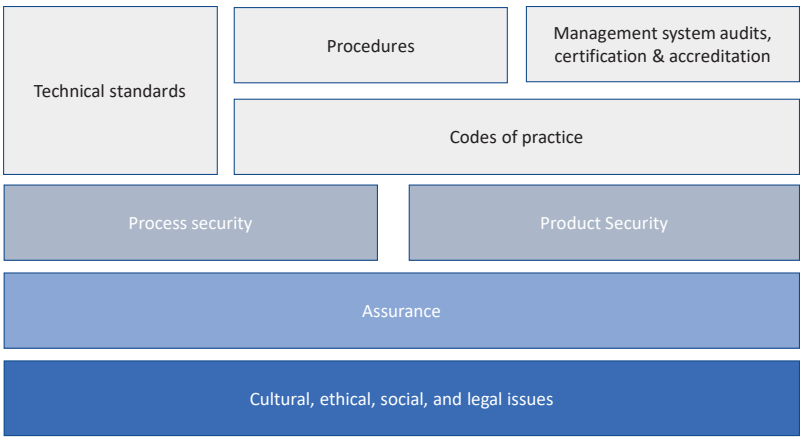


IoT and cloud standards

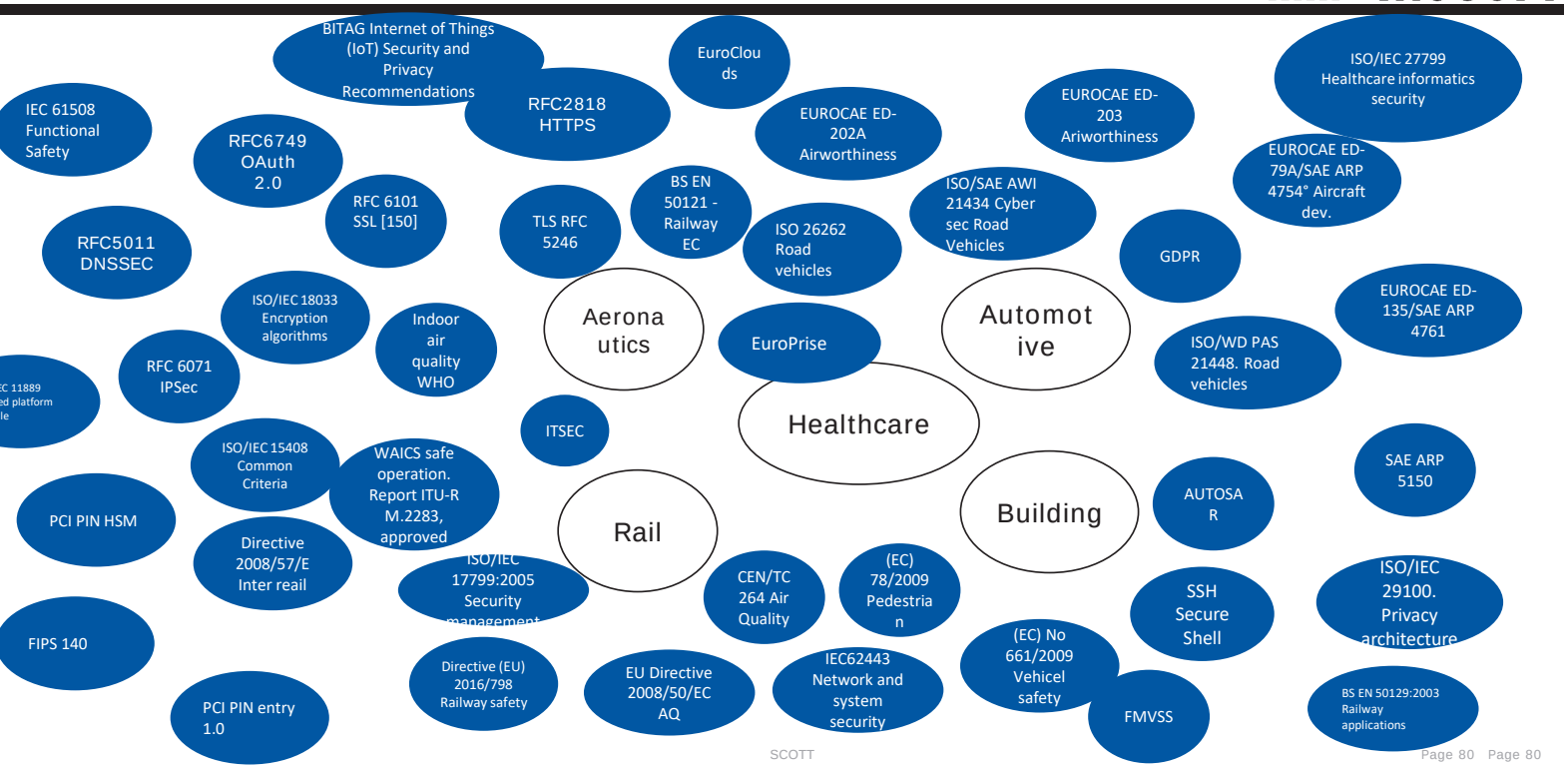


Security standards

- Security standards can be grouped as shown in the figure
- Each standard can be abstracted into different types of metrics to create security classes



Security and safety standards



Level 1 middleware technologies

- Multiple solutions to retrieve data inside the bubble
- Publish subscribe fit critical services
- Request/response fit non-delay sensitive

	Pattern	Resources	Complexity	Security	Application	QoS
MQTT	Pub/sub	Med. (TCP)	Low	Auth (SSL)	Telemetry NRT	At most once
MQTT-SN	Pub/sub	Low (UDP)	Med.	Auth	Telemetry	
AMQP	Pub/sub	High (TCP)	Med.	TLS SASL	Mess. exchange	Not standard
CoAP	Req/Res	Low (UDP)	Low	DTLS	Telemetry Constrained devices	Best effort
HTTP	Req/Res	High (TCP)	Low	HTTPS	Constrained devices	Best effort
DDS	Pub/sub	High (TCP) Low (UDP)	High	Auth.	real time	20 QoS policies

Definitions

- MQTT is a lightweight, publish-subscribe, machine to machine network protocol. It is designed for connections with remote locations that have devices with resource constraints or limited network bandwidth
- AMQP. The Advanced Message Queuing Protocol is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing, reliability and security.
- CoAP. Constrained Application Protocol is a specialized Internet application protocol for constrained devices, as defined in RFC 7252. It enables those constrained devices called "nodes" to communicate with the wider Internet using similar protocols
- Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes.
- The Data Distribution Service (DDS) for real-time systems is an Object Management Group (OMG) machine-to-machine (sometimes called middleware or connectivity framework) standard that aims to enable dependable, high-performance, interoperable, real-time, scalable data exchanges using a publish–subscribe pattern.

Messaging and transporting technologies

- Transport technologies are needed to provide end to end packet order connectivity

SOAP: Simple Object Access protocol
HTTP: Hyper-text Transfer Protocol
TCP: Transfer Control Protocol
DTLS:Datagram Transport Layer Security
NRT: Non-real-Time
UDP: user Diagram Protocol

No.	Resources	Complexity	Security	Application
SOAP	High (TCP)	High	HTTPS	Complex message exchange
CoAP	Low (UDP)	Low	DTLS	Telemetry resource constrained NRT
HTTP	High (TCP)	Low	HTTPS	Resource constrained NRT
Web services	High (TCP) Low (UDP)	High	HTTPS	

Resource Discovery and catalogue technology

- There is a need to provide resource addressing to help in discovery of services inside and outside the bubble
- Each standard can be abstracted into different types of metrics to create security classes
- UDDI and WSDL are widely used

UDDI: Universal Description Discovery and Integration
WSDL: Web service Description Language
WADL: Web Application Description Language
DTLS: Datagram Transport Layer Security
NRT: Non-real-Time
UDP: user Diagram Protocol
SSDP: Simple Service Discovery Protocol
LWM2M: Lightweight M2M
DNS-DS: Domain Name System Discovery Service

	Resources	Application Dependency
WSDL	High	Describes SOAP
WADL	High	Describes SOAP and RESTful
WS-discovery	High	Based on TCP/IP , UDP or SOAP
SSDP	High	Relies on TCP/IP
DNS-SD	High	
LWM2M	Low	Resource constrained
HyperCAT	Low	IoT
SA-REST	Low	
Swager	Low	JSON
UDDI	Medium	

Description

- UDDI (Universal Description, Discovery, and Integration) is an XML-based registry for businesses worldwide to list themselves on the Internet.
- HyperCat is an open, lightweight JSON-based hypermedia catalogue format for exposing collections of URIs
- WSDL is an XML notation for describing a web service. A WSDL definition tells a client how to compose a web service request and describes the interface that is provided by the web service provider.
- The Web Application Description Language (WADL) is a machine-readable XML description of HTTP-based web services
- SSDP (Simple Service Discovery Protocol) is a network protocol used in small networks, including home networks, to advertise and discover network services primarily supported by the Universal Plug-and-Play (UPnP) architecture
- Datagram Transport Layer Security (DTLS) is a communications protocol providing security to datagram-based applications by allowing them to communicate in a way designed to prevent eavesdropping, tampering, or message forgery.

Data serialization

- These technologies describe the way we order the data to be sent in a message
- XML (Extensible Markup Language)
- YAML (Yet Another Markup Language)
- CBOR (Concise Binary Object Representation)
- TLV (Type-length Value)
- JSON Java Script Object Notation.

	Resources	Complexity
XML	High	High
YAML	High	Low
CBOR	High	Low
TLV	High	High
JSON	High	

Definitions

- Extensible Markup Language is a markup language and file format for storing, transmitting, and reconstructing arbitrary data
- YAML is a human-readable data-serialization language. It is commonly used for configuration files and in applications where data is being stored or transmitted. YAML targets many of the same communications applications as Extensible Markup Language but has a minimal syntax which intentionally differs from SGML
- Concise Binary Object Representation (CBOR) is a binary data serialization format loosely based on JSON authored by C. Bormann. Like JSON it allows the transmission of data objects that contain name–value pairs, but in a more concise manner.
- TLV is an encoding scheme used for optional informational elements in a certain protocol.
- JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays

Semantic technologies

- Semantics technologies provide meaning to the data transmitted between nodes

	Type	Advantage/disadvantage
IPSO	Specific	Use with CoAP/RESTful Based on OMA LWM2M
OData	Specific	RESTful oriented
RDF	Ontology	Ontology vocabulary
OWL	Ontology	More concepts than RDF
SensorML	Ontology	XML for IoT
SSN	Ontology	
IoT-A	Ontology	Based on RDF and Web Ontology Language
X-GSN	Ontology	XML-based

IPSO: Universal Description Discovery and Integration
REST: Representational state transfer i
RDF: Resource Description Framework
OWL: Web Ontology Language
SSN: Semantic Sensor Network

Definitions

- The Internet Protocol for Smart Objects (IPSO) Alliance was an international technical standards organization promoting the Internet Protocol (IP) for what it calls "smart object" communications
- Open Data Protocol is an open protocol that allows the creation and consumption of queryable and interoperable REST APIs in a simple and standard way.
- The Resource Description Framework is a World Wide Web Consortium standard originally designed as a data model for metadata. It has come to be used as a general method for description and exchange of graph data.
- the Web Ontology Language is a family of knowledge representation languages for authoring ontologies
- The Semantic Sensor Network (SSN) ontology is an ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties, as well as actuators.

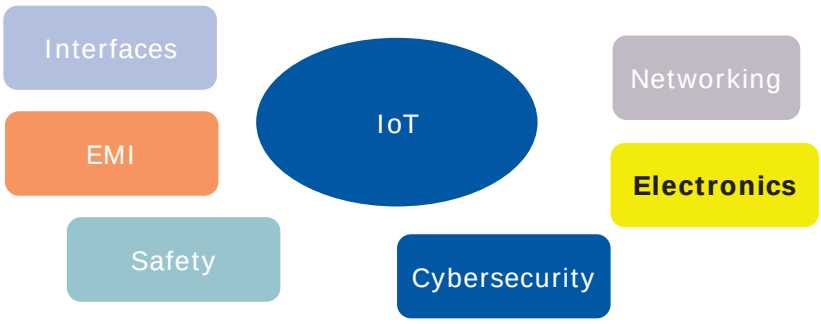
Certification and consumer trust

- The end consumers trust more in a product if it is certified by an industrial association or a regulatory body.
- The certification process can be complex, and it varies according to the type of product, country or region of world, and the consumer idiosyncrasy.
- Globalization has created the need to harmonize certification, standardization and regulation of multiple products, services or processes across countries.
- Labels, seals, marks, are the visible result of a product certification

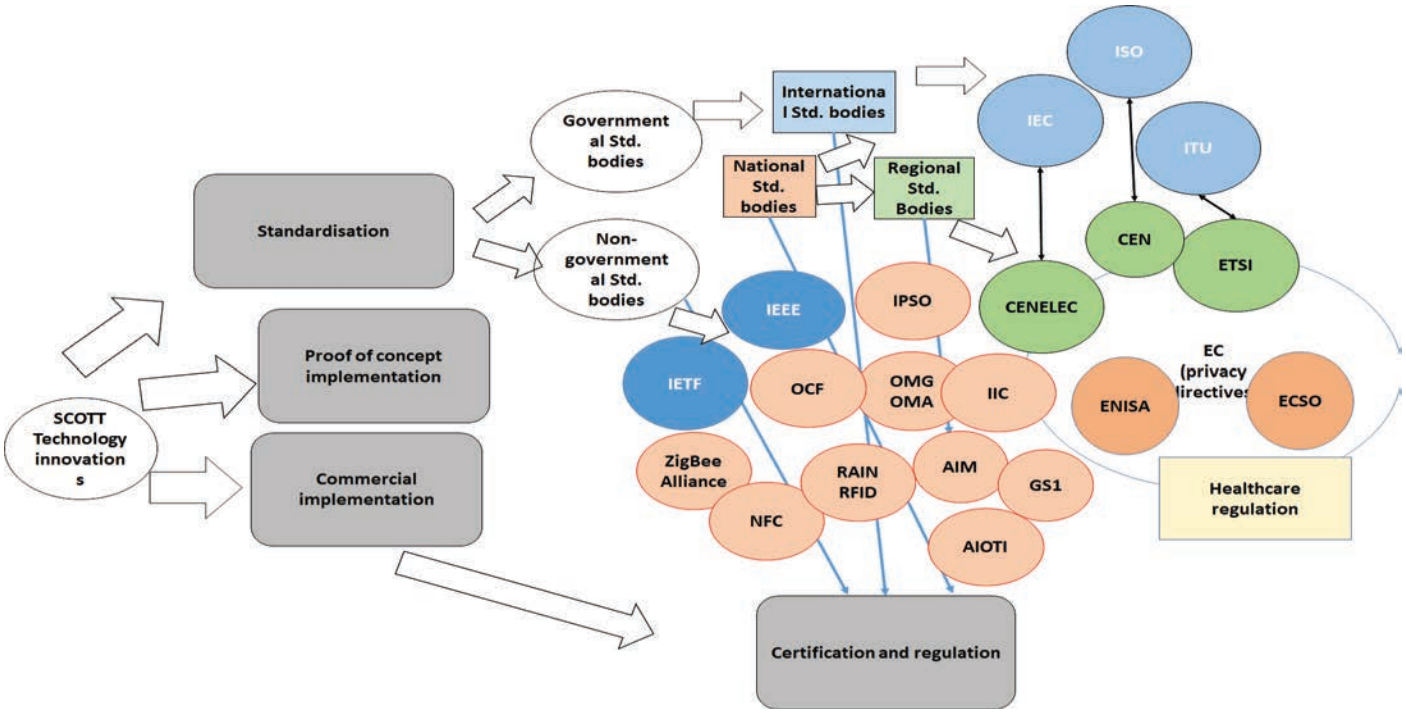


The need for certification on the Internet of Things

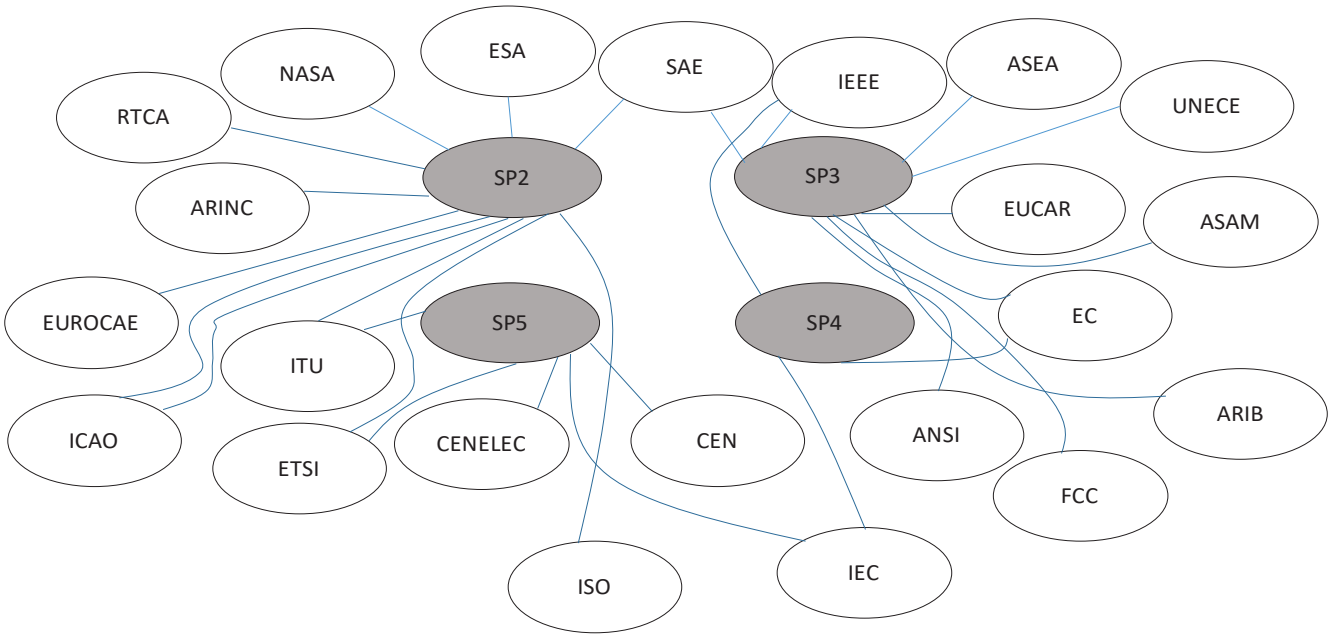
- IoT promises a new generation of devices with embedded processors with networking capabilities.
- New services, products and issue will be created by the upcoming technological advances.
- Great challenge for certification entities due to the wave of millions of products, services and implications to human lives that the services are expected to bring.
- More complex certification issues.



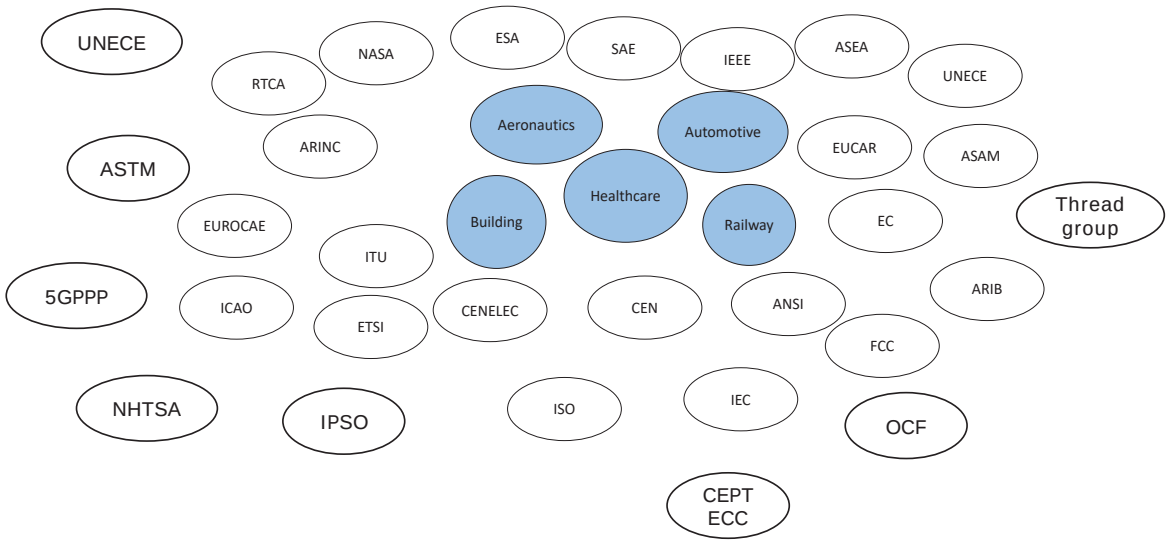
Standardisation and regulation bodies



Per industrial domain



Regulation framework



Questions?

Ramiro Samano Robles
CISTER/ISEP, Portugal
rasro@isep.ipp.pt



Faculty of Organizational Sciences

Overview of Several CPS&IoT Prototypes: Wheelchair, Group Heart Rate Monitoring, PID DC Motor Control

Prof. Dr. Andrej Škraba

Cybernetics & Decision Support Systems Laboratory

Wheelchair

- Cloud speech recognition applied in control of the wheelchairs for disabled persons
- Tight integration with Internet and its users, which continuously feed the database / possible to provide corrections
- Price, accuracy
- Drawbacks, such as latency
- Word error rate (WER – CER)
- Technical difficulty to use cloud Application Programming Interface (API)
- More new cloud speech recognition services available
- Develop efficient algorithms, which will combine speech recognition results

Specification

- Prototyping
- Control the movement of the wheelchair with speech
- The principle of cloud harvesting should be applied
- In addition to speech, control should be possible via web-based GUI
- Remote monitoring and control.
- Real-time video streaming from the wheelchair platform
- Biomedical signals
- Provide uniform GUI with interactive graphics of main parameters.

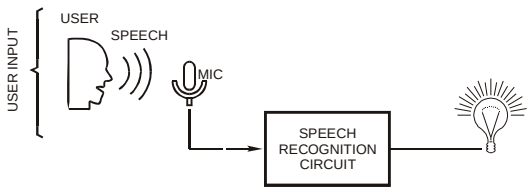
Cyber-physical Systems & Internet of Things

- A cyber-physical system (CPS) is a mechanism controlled or monitored by computer-based algorithms, **tightly integrated with internet and its users**
- Internet of Things (IoT) is a subset of CPS where „physical" is omitted i.e. monitoring or providing an information (still in formulation)
- Today, informatics should not only measure but also listen, watch, interact with users, move, handle, grab etc.

Overview of Several CPS&IoT Prototypes

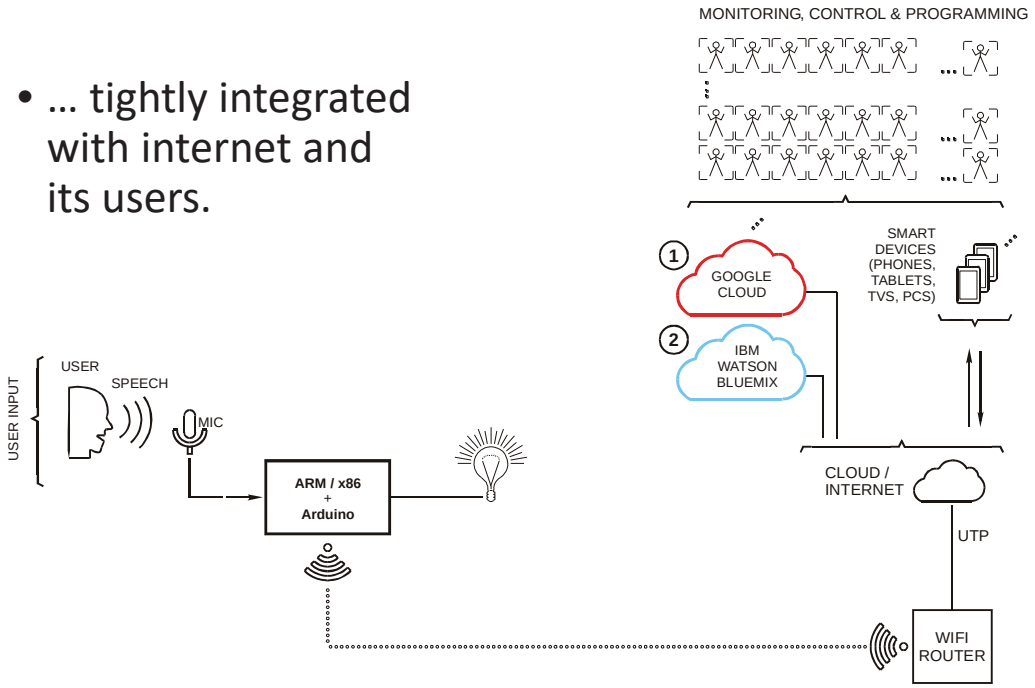
Example of speech controlled device

- Application of speech recognition circuit
- Embeded logic
- Hard to adapt
- Self sufficient device



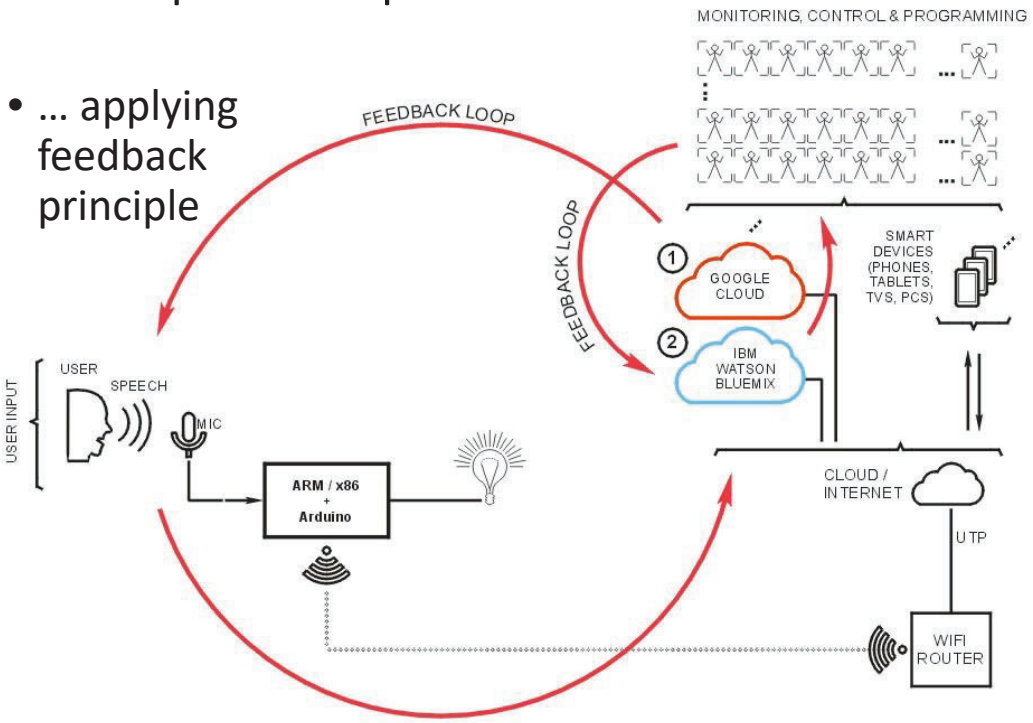
Example of speech controlled device (ver. 2)

- … tightly integrated with internet and its users.

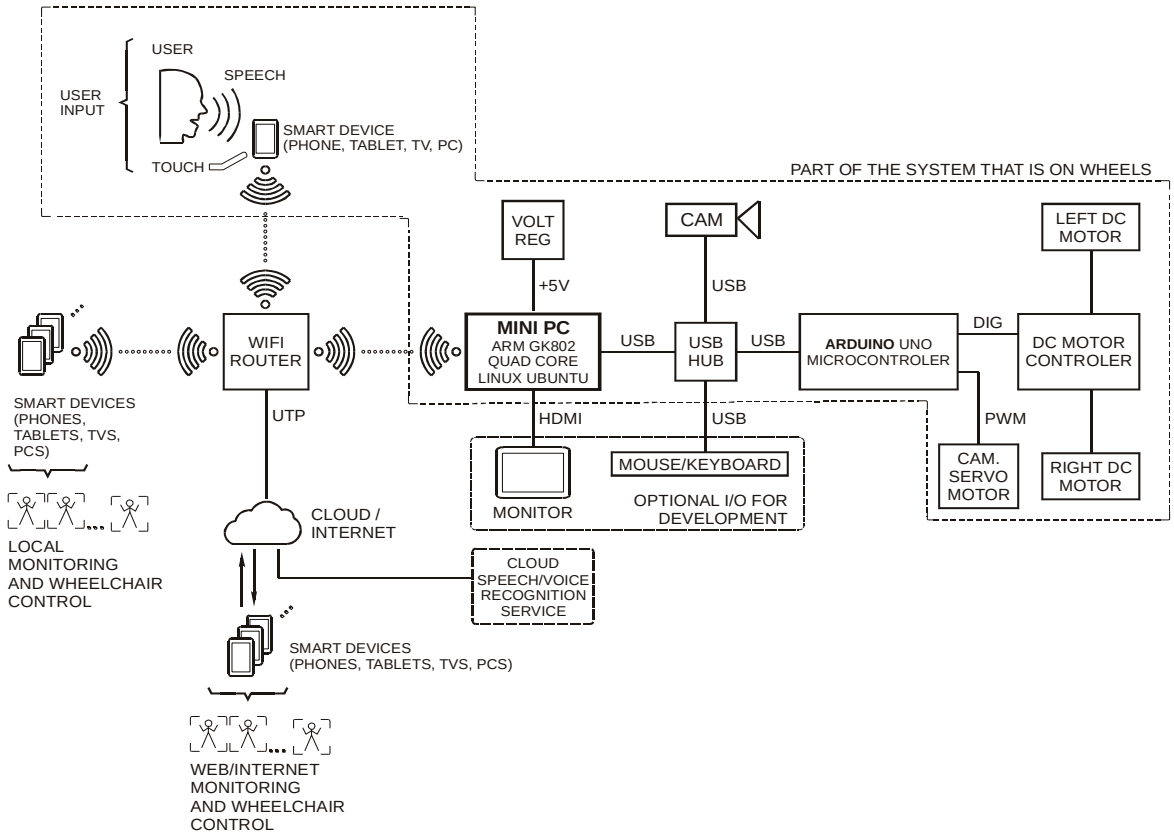


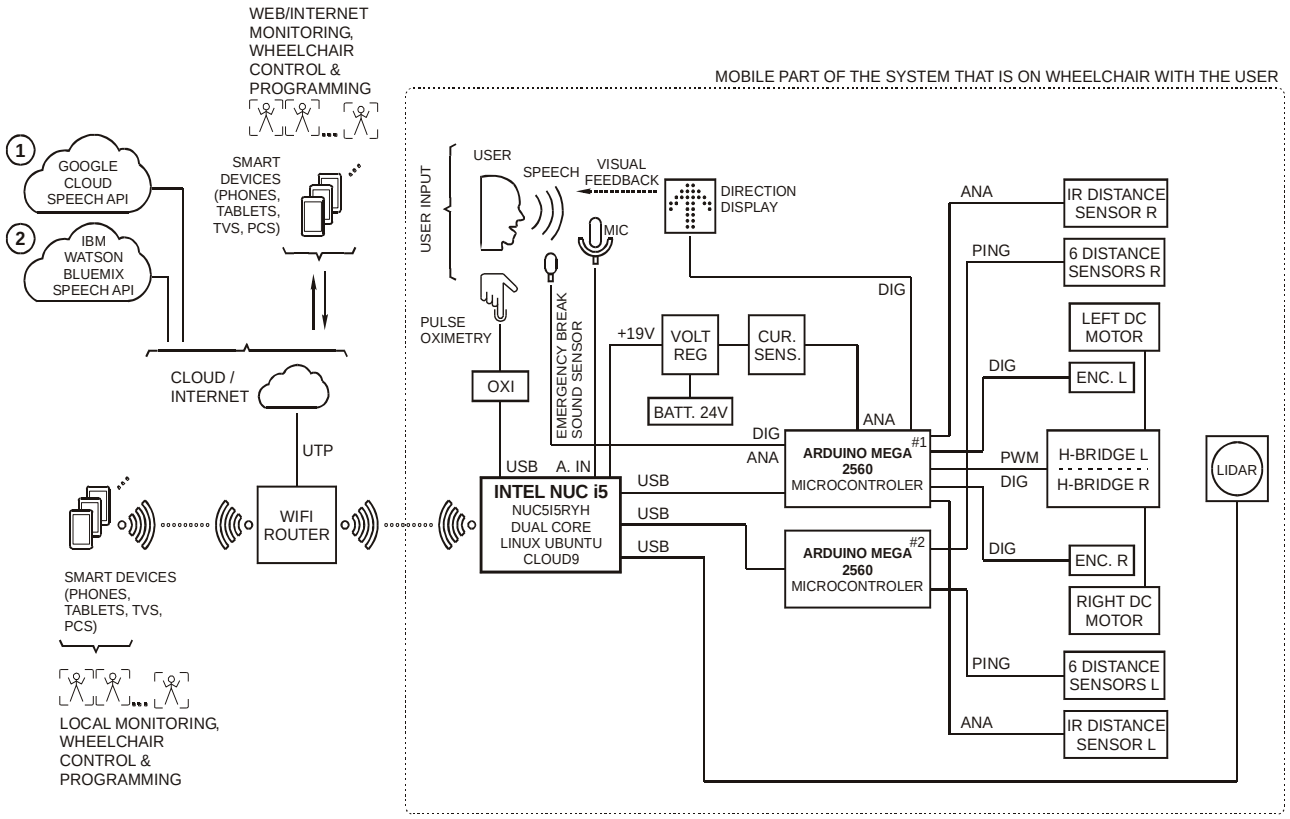
Example of speech controlled device (ver. 3)

- … applying feedback principle



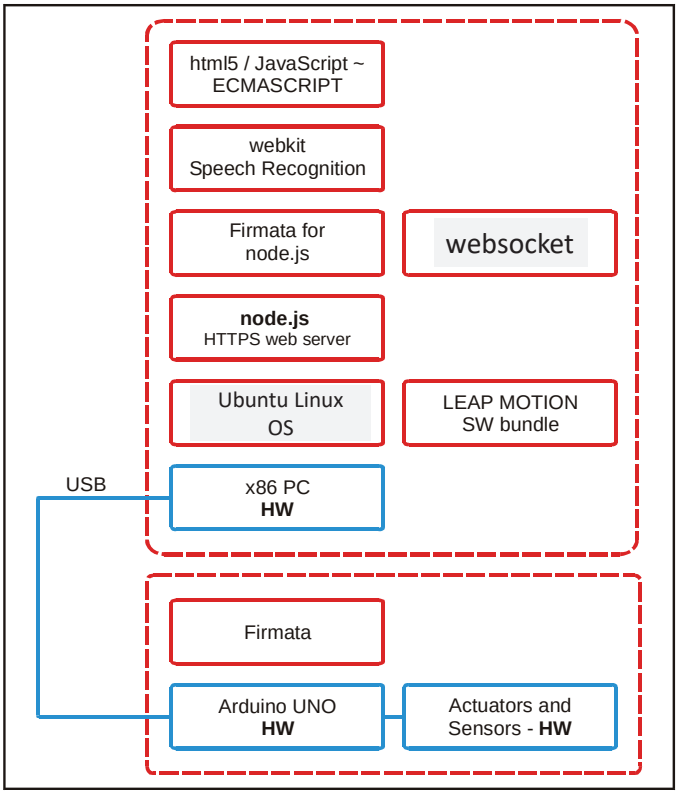
Wheelchair System Architecture 1st




WEB/INTERNET MONITORING, WHEELCHAIR CONTROL & PROGRAMMING
MOBILE PART OF THE SYSTEM THAT IS ON WHEELCHAIR WITH THE USER
1
GOOGLE CLOUD SPEECH API
2
IBM WATSON BLUEMIX SPEECH API
SMART DEVICES (PHONES, TABLETS, TVS, PCS)
CLOUD / INTERNET
UTP
WIFI ROUTER
SMART DEVICES (PHONES, TABLETS, TVS, PCS)
LOCAL MONITORING, WHEELCHAIR CONTROL & PROGRAMMING
USER INPUT
USER
SPEECH
VISUAL FEEDBACK
DIRECTION DISPLAY
MIC
PULSE OXIMETRY
OXI
EMERGENCY BREAK SOUND SENSOR
+19V
VOLT REG
CUR. SENS.
BATT. 24V
USB
A. IN
INTEL NUC i5
NUC5I5RYH
DUAL CORE
LINUX UBUNTU
CLOUD9
ARDUINO MEGA 2560 MICROCONTROLER #1
ARDUINO MEGA 2560 MICROCONTROLER #2
DIG
ANA
PWM
PING
IR DISTANCE SENSOR R
6 DISTANCE SENSORS R
LEFT DC MOTOR
ENC. L
H-BRIDGE L
H-BRIDGE R
ENC. R
RIGHT DC MOTOR
6 DISTANCE SENSORS L
IR DISTANCE SENSOR L
LIDAR

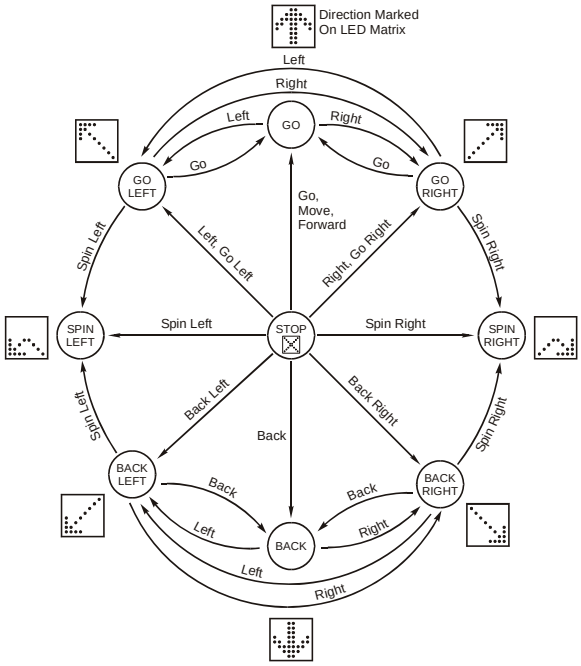
Software Stack

- node.js
- JavaScript / ECMAScript
- Firmata / Serial
- Ubuntu Linux
- Cloud Speech API
- Google & IBM Watson
- LEAP Motion SW Bundle



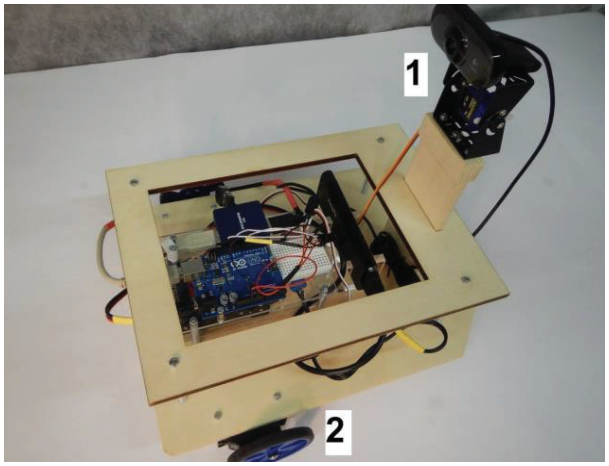
Transition Between States - Speech

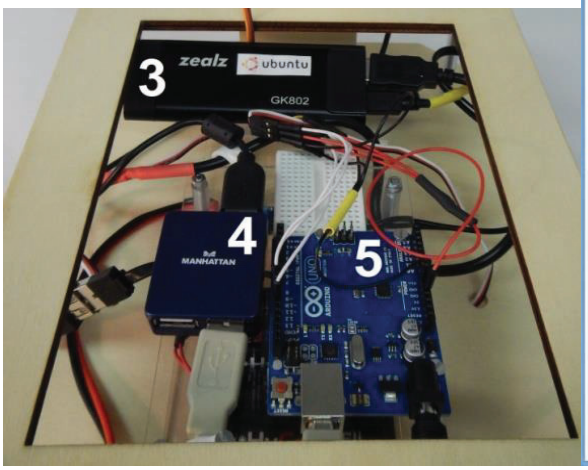
- Different interpretations depending on the sequence of issued commands

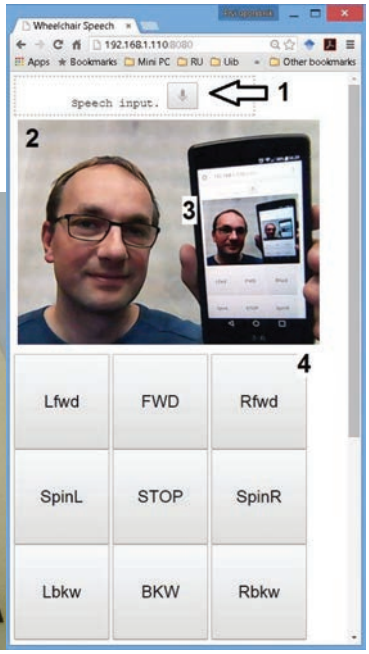


Prototype realization 0

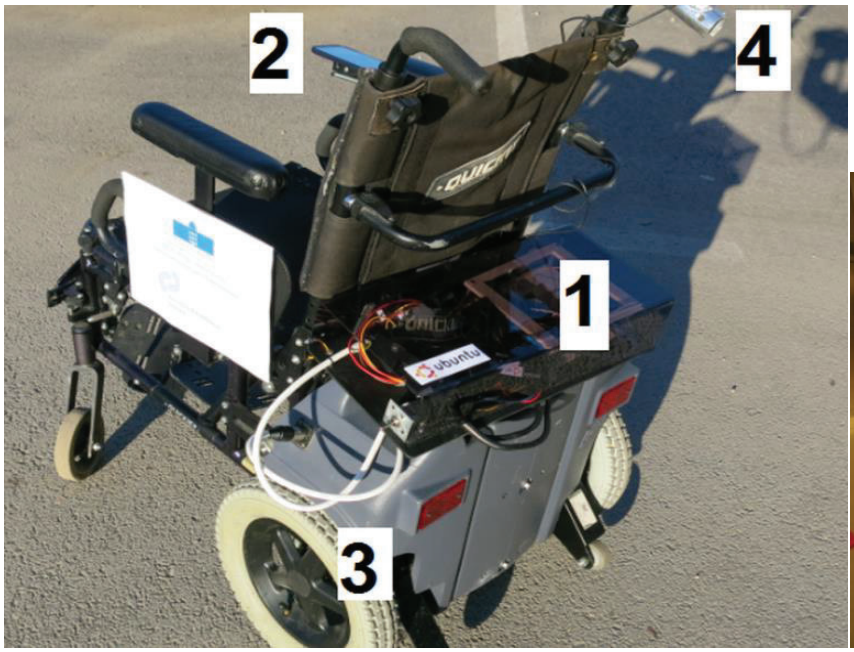
- ARM based solution GK802 quad core
- Speech controlled prototype: https://youtu.be/Y4El7lBTxQA







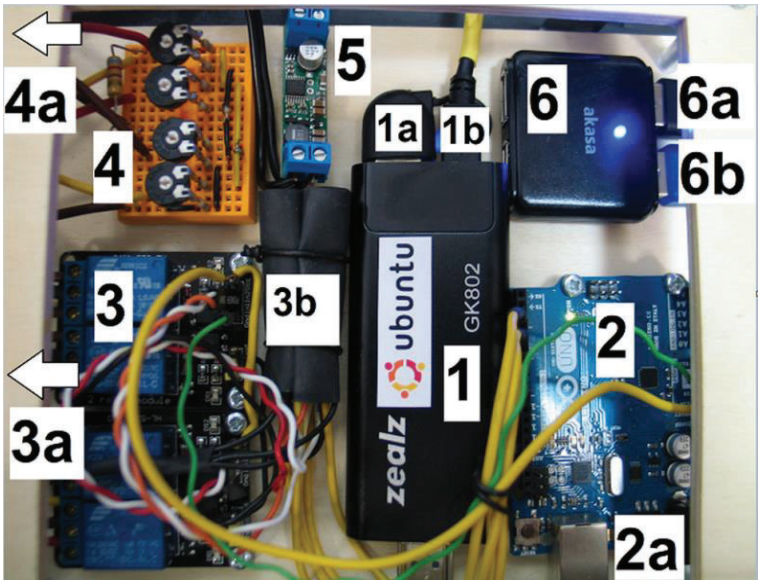
Full Scalled Prototype and Clinical Testing



https://youtu.be/FMjffyMWcKM?t=728

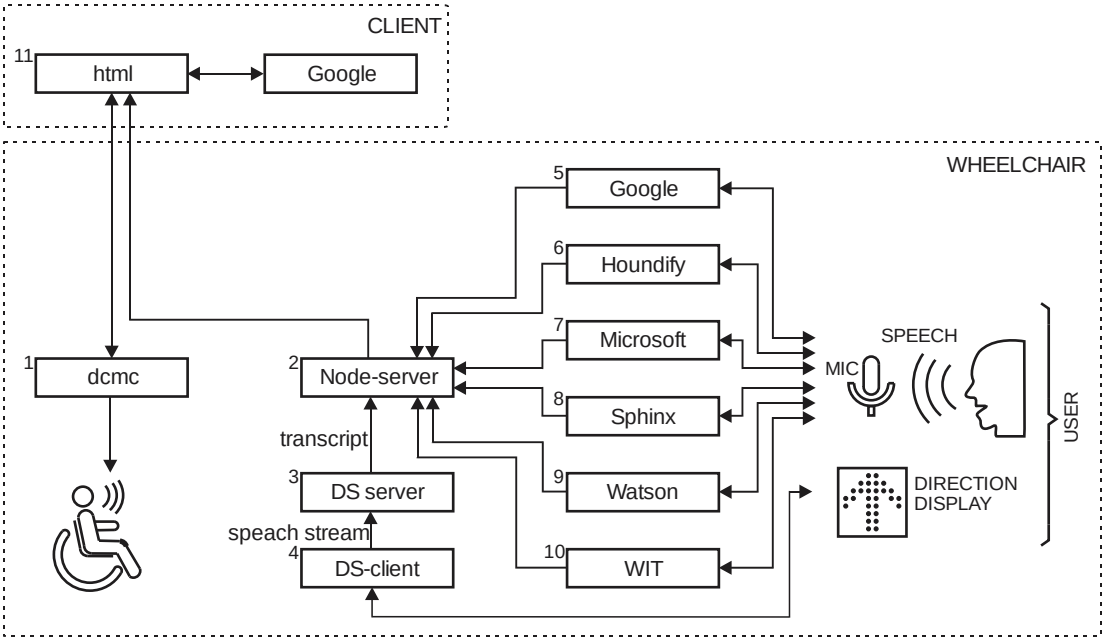


Comparison


4a
5
1a
1b
6
6a
6b
4
3
3b
2
1
3a
2a
ubuntu
GK802
zealz
akasa


+ Intel NUC i7

Client and server side of a hybrid cloud/edge speech recognition ensemble system



Conditions

- In order for the procedure to be sucesfull it should hold:

$$f_2 = (a \wedge \neg b) \vee (\neg a \wedge b)$$

- For three paralel systems:

$$f_3 = (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c)$$

- With CER when multiple clouds are harvested (+ latency etc.):

$$CER_m = \prod_{i=1}^{n} CER_i$$

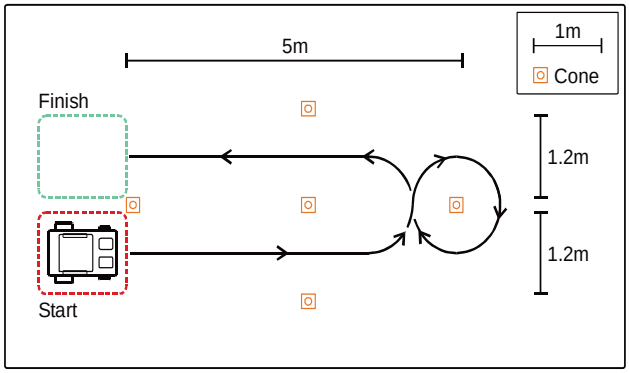
Speech-to-command cloud harvesting algorithm

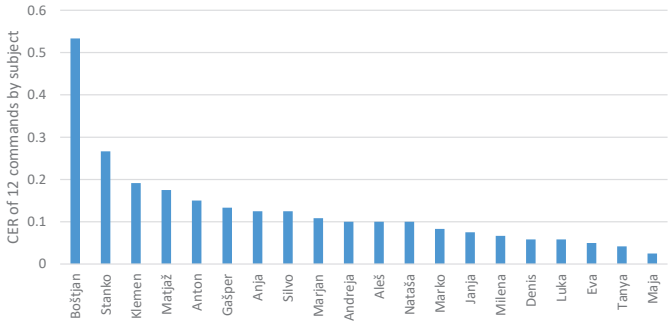
A1:	get user speech input
A2:	seed speech to the speech-recognition cloud field APIs
A3:	harvest set of interim transcripts and timestamps from cloud field
A4:	if $C_w > C_t$ add interim transcript to the $Cloud_i$ command subset
A5:	create unique union set of words for particular command from $Cloud_i$ command subset
A6:	check for pairwise disjoint condition for all unique union sets: $A_i \bigcap A_j \equiv \emptyset \ ; \ i \neq j$
A7:	if condition not met erase word pair
A8:	order checked unique union set by interim transcript timestamp
A9:	execute command with lowest timestamp

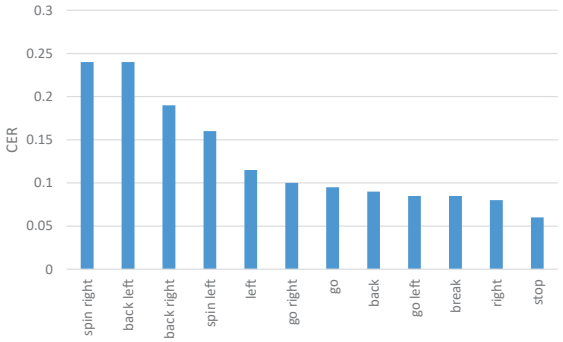
CER Measurement

- Testing 20 subjects
- Using poligon setup

$$CER = \frac{C_f}{\text{No}}$$

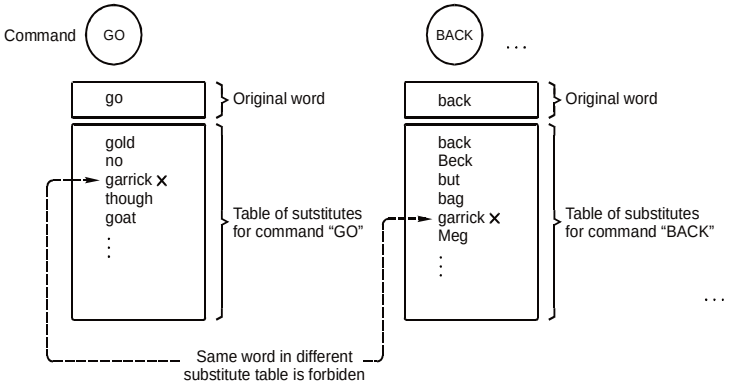






Algorithms

- Improve CER with table of substitutions
- Harvesting the clouds with corrections
- Application and automatic generation of substitution tables

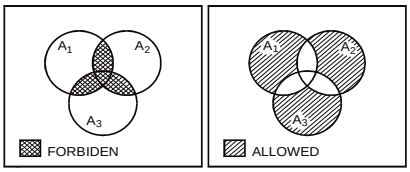


N substitution tables:

$A_1, A_2, A_3, \ldots, A_n,$

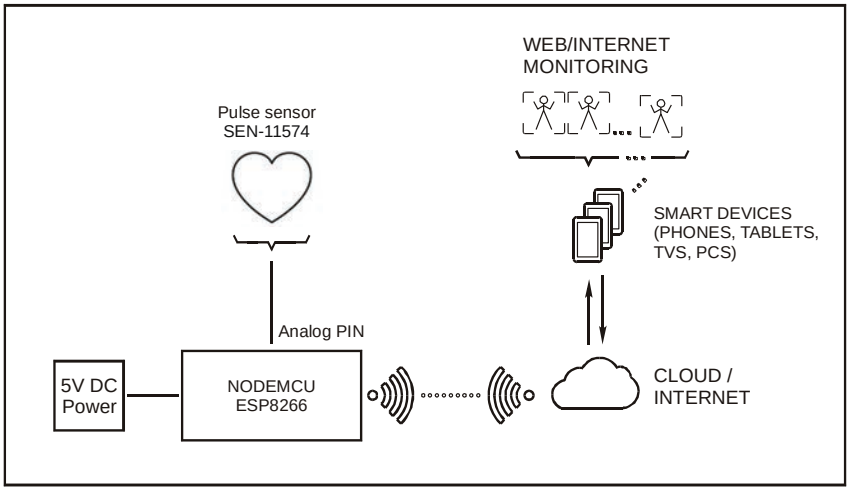
Pairwise disjoint

$A_i \bigcap A_j \equiv \emptyset \ ; \ i \neq j$



Streaming pulse data from Wheelchair

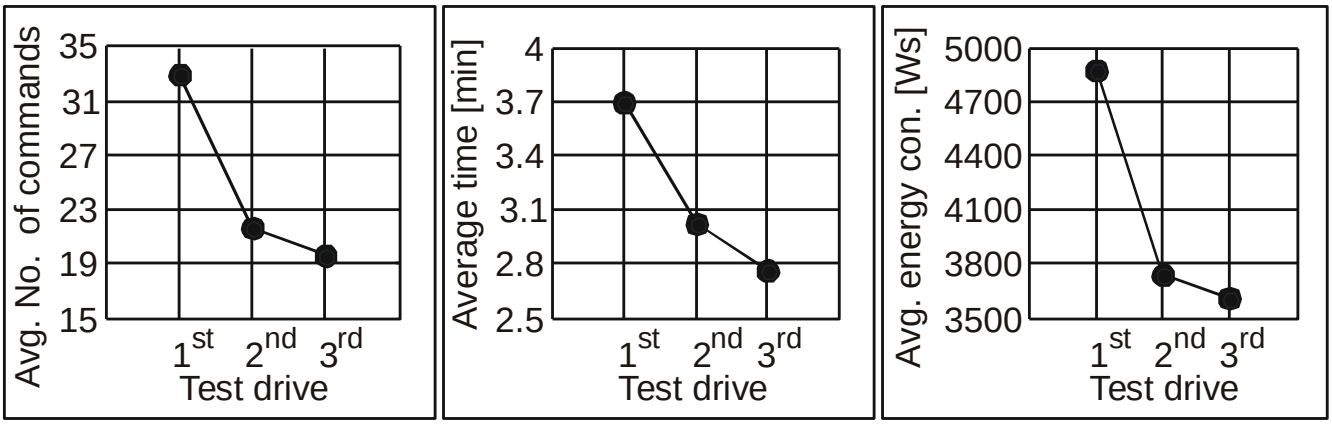
- NODEMCU ESP8266 module
- Less components and
- Directly connected to the Wi-Fi
- Sensor data processing is performed on the ESP8266
- Transmitted over Wi-Fi WebSocket to the cloud
- Additional data processing on the client side with JavaScript / ECMA Script



Average Energy Consumption

Run	N	AVG [#c.]	SD	N	AVG [min]	SD	N	AVG [W·s]	SD
1st	14	33.0	14.12	13	3.70	0.93	14	4875.79	1577.27
2nd	14	21.7	9.16	13	3.03	0.79	14	3742.50	1999.96
3rd	14	19.6	8.40	13	2.77	0.74	14	3606.64	1627.24

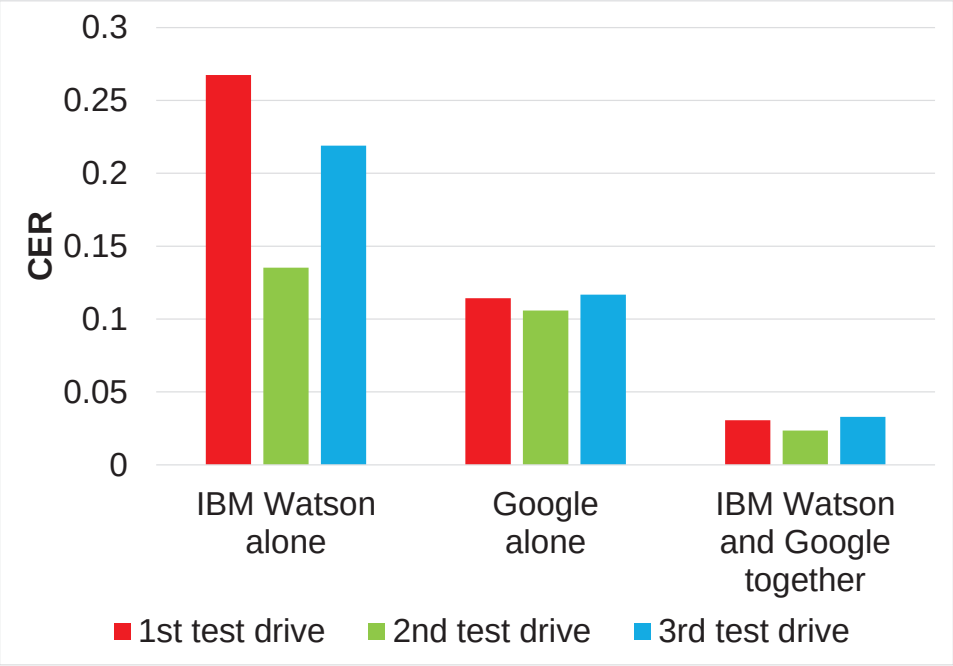
Learning effect



Harvesting Google & IBM Watson

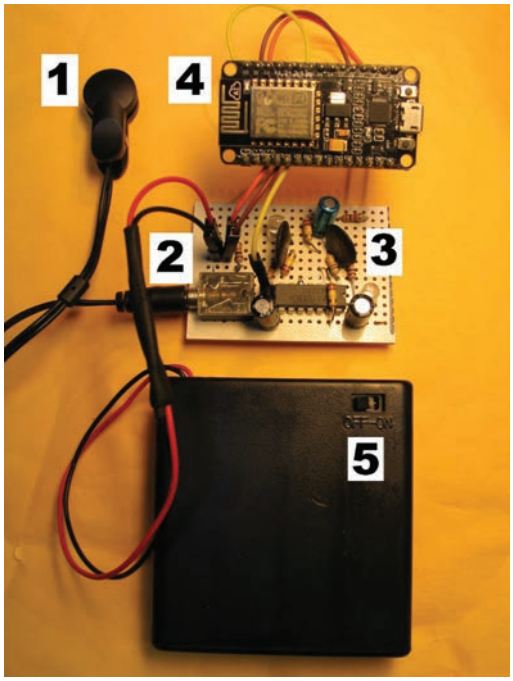
Run	No. of issued comm.	N	CER of Google API alone (CERg)	CER of IBM Watson API alone (CERw)	CER of Google & IBM Watson combined (CERgw)	Google CER improvement	Watson CER improvement
1st	490	20	0.11	0.27	0.03	8%	24%
2nd	340	20	0.11	0.14	0.02	8%	11%
3rd	274	20	0.12	0.22	0.03	8%	19%

Improvement



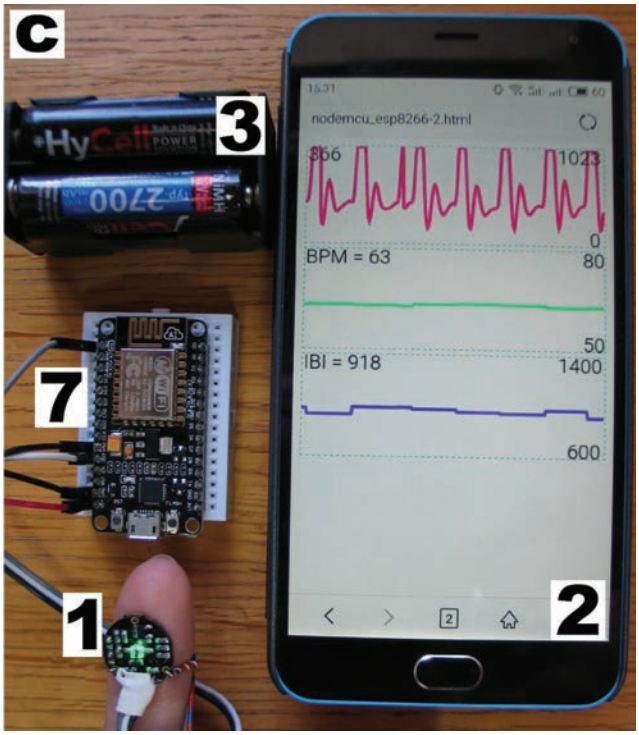
Group Heart Rate Monitoring

- NODEMCU ESP8266 based configuration
- By developed software stack and hardware realization the results could be monitored in the web browser
- ESP8266 websocket support
- Chrome – websocket support



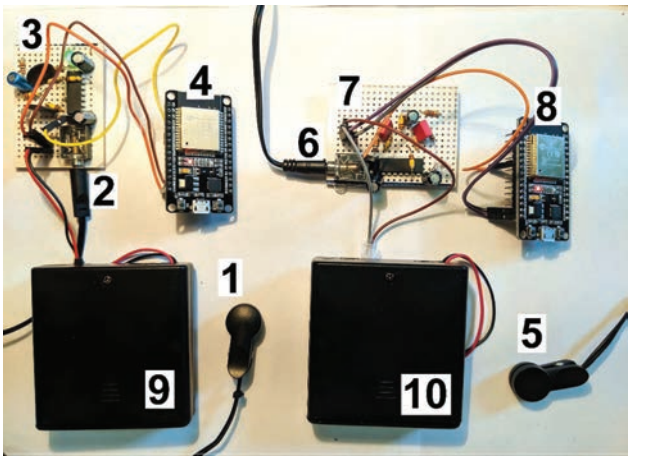
NODEMCU ESP8266

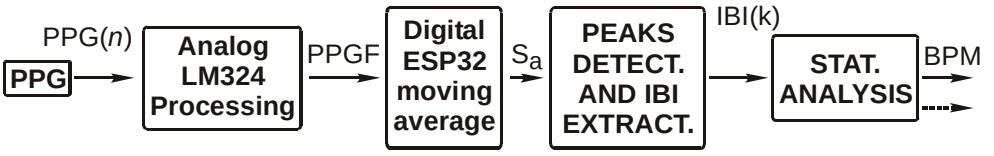
- Minimalistic regarding the hardware components
- The algorithm for processing the raw data, calculating the average number of Beats Per Minute (BPM), as well as the InterBeat Interval (IBI) in case of using NODEMCU was implemented on client's side in JavaScript
- Power consumption is a major issue



Example ~ ESP32

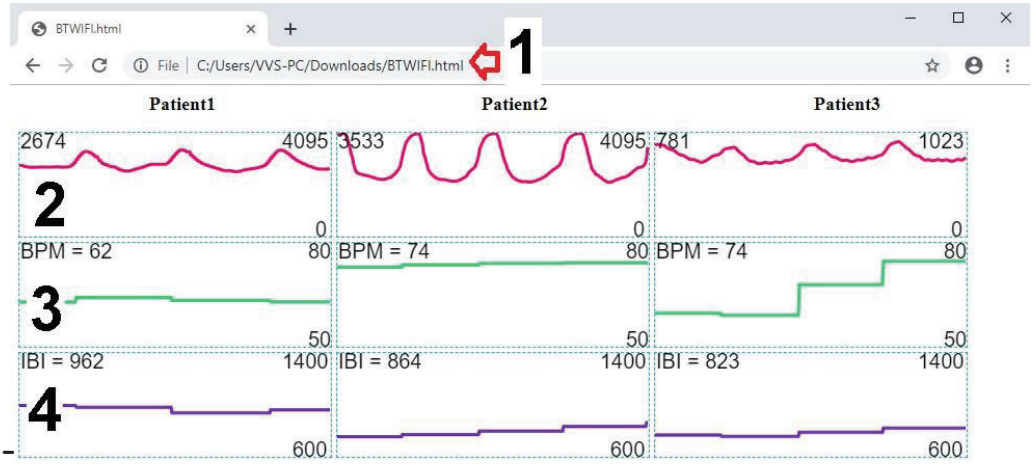
- ESP32 based configuration
- By developed software stack and hardware
  realization the results could be monitored in the web browser
- ESP32 websocket support
- Chrome – websocket support





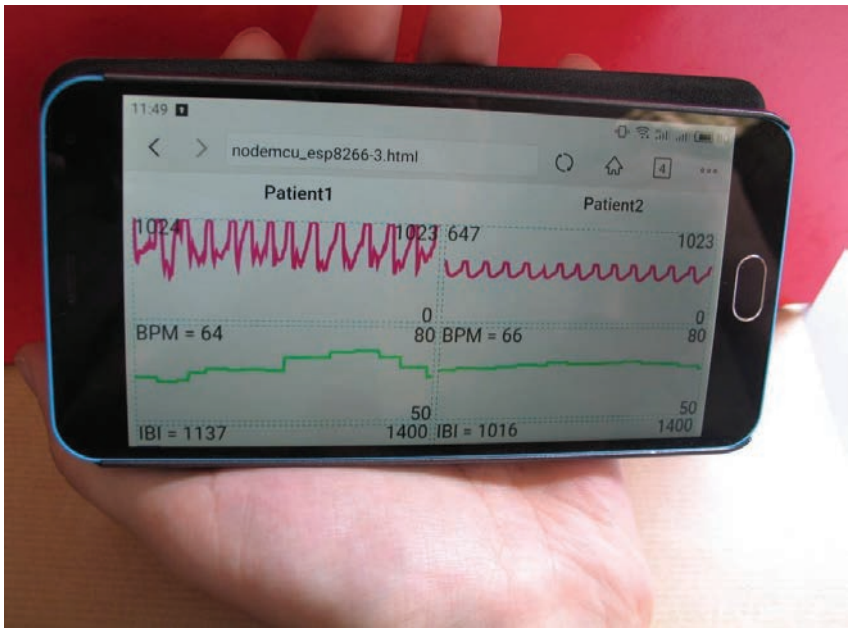
Monitoring of Multiple Patients

- Easy Development of GUI in javascript
- Responsivenes
- Possible to develop interfaces for larger group of patients (intensive care)
- Extending the functionality of existing equipment



Representation on the Phone

- Suitable for monitoring of group of patients
- SW for alarming
- Logging



Power consumption MK802V5LE Vs rPi0w

Cond.	Rpi0w [A] [a]	P [W]	MK802V5 LE [A] [b]	P [W]	[%]
Bare	0.125	0.664	0.245	1.169	76
+USB Hub	0.225	1.196	0.335	1.598	34
+USB Hub & Arduino	0.275	1.462	0.375	1.789	22
Complete w cloud9	0.335	1.781	0.425	2.027	14

a. @ 5.315V b. @ 4.770V

Power consumption (cont.)

Configuration	Current [mA]*
BT HC-06 + Arduino UNO SMD + Pulse sensor SEN 11547	**18**
BT HC-06 Bluetooth module only	7.5
BT LE Adafruit nRF8001 + Arduino UNO SMD + Pulse sensor SEN 11547	**11**
BT LE Adafruit nRF8001 Bluetooth module only / when transmitting	0.4
BT LE Adafruit nRF8001 Bluetooth module only / when not transmitting (search)	0.55
NODEMCU Amica ESP8266 MOD 80Mhz 4Mb RAM + Pulse sensor SEN 11547	**30.5**

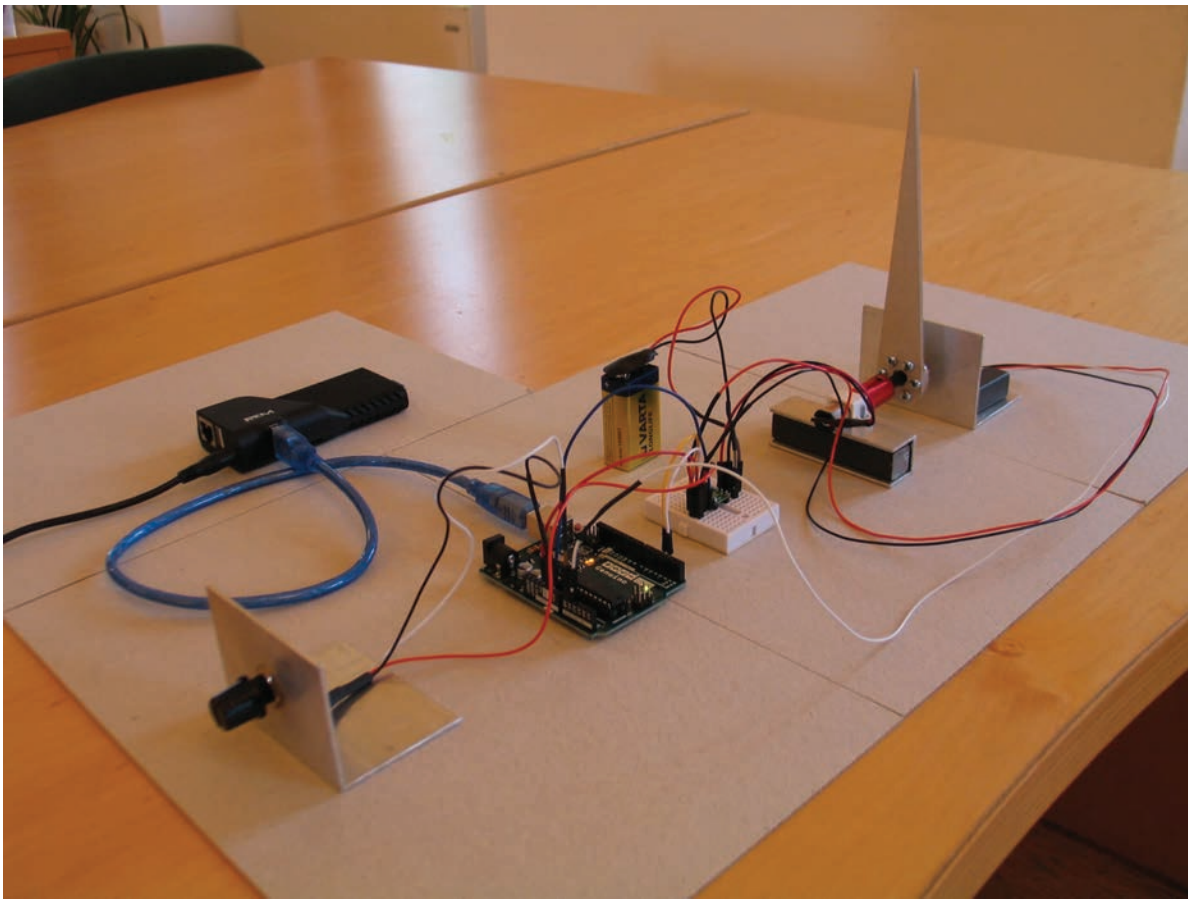
Power consumption (cont.)

Device	Comm.	noTr [A]	Trans. [A]	[V]	noTr P[W]	Trans. P[W]
ESP8266	WiFi	0.095	0.095	5.03	0.478	0.478
ESP32	WiFi	0.05	0.13	5.01	0.251	0.651
ESP32	BT	0.05	0.125	5.2	0.260	0.650

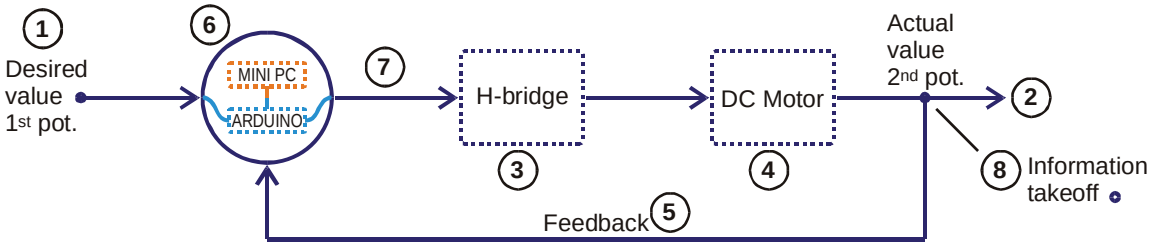
- Main difference in transmission (noTr/Tr) mode

Example M

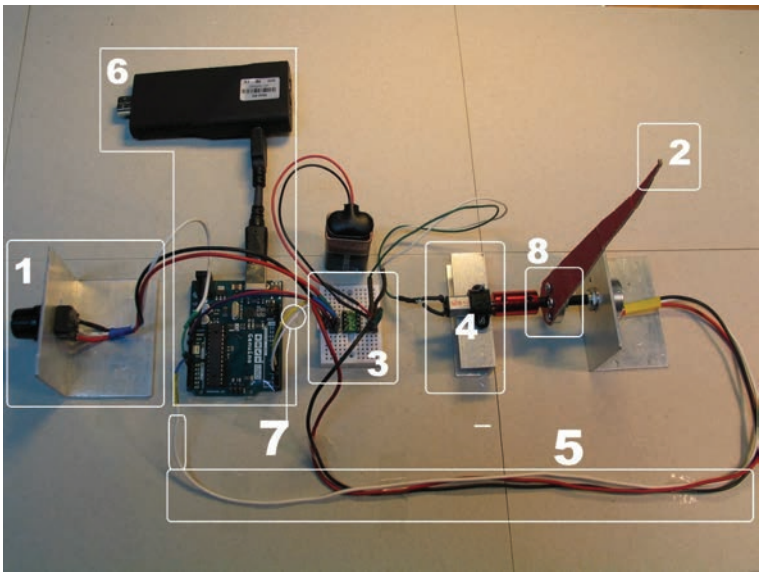
- Arduino
- Mini PC MK802V
- H-bridge
- DC motor
- Programming in browser
- Cloud9 IDE
- JavaScript/ ECMAScript



Example M (cont.)

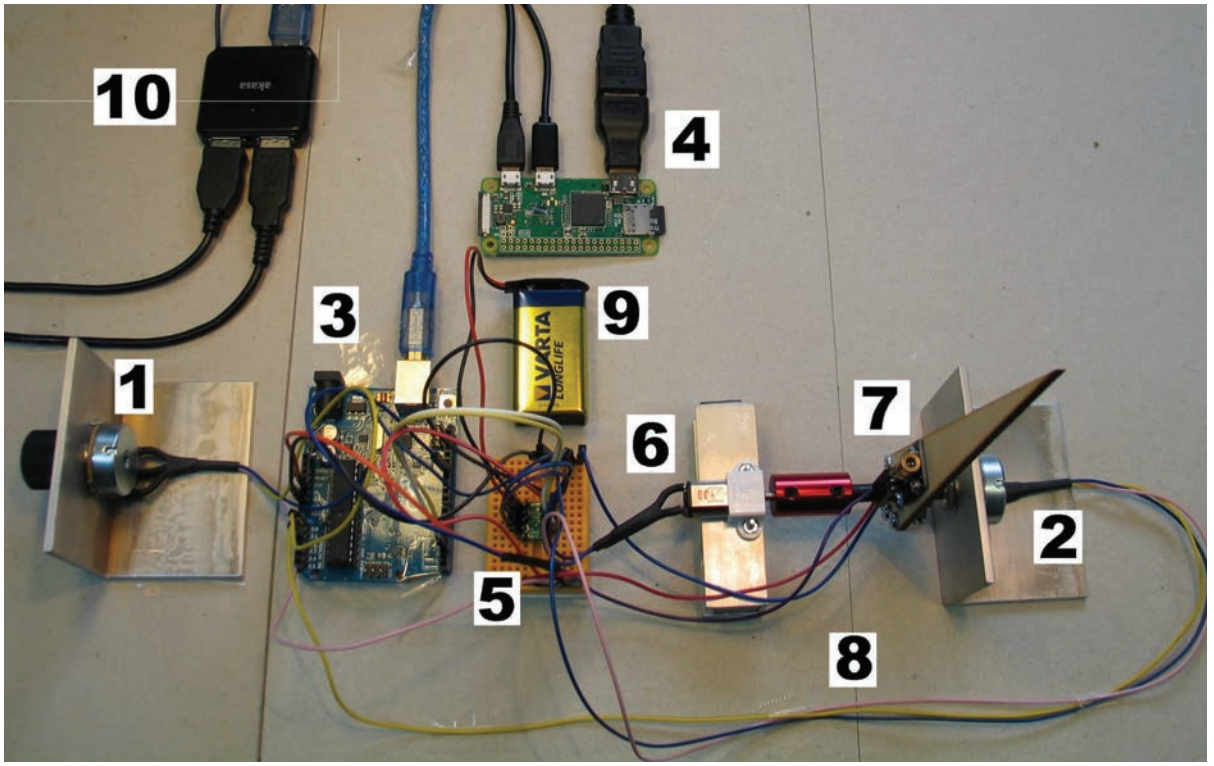


- Paralel with control system schematics



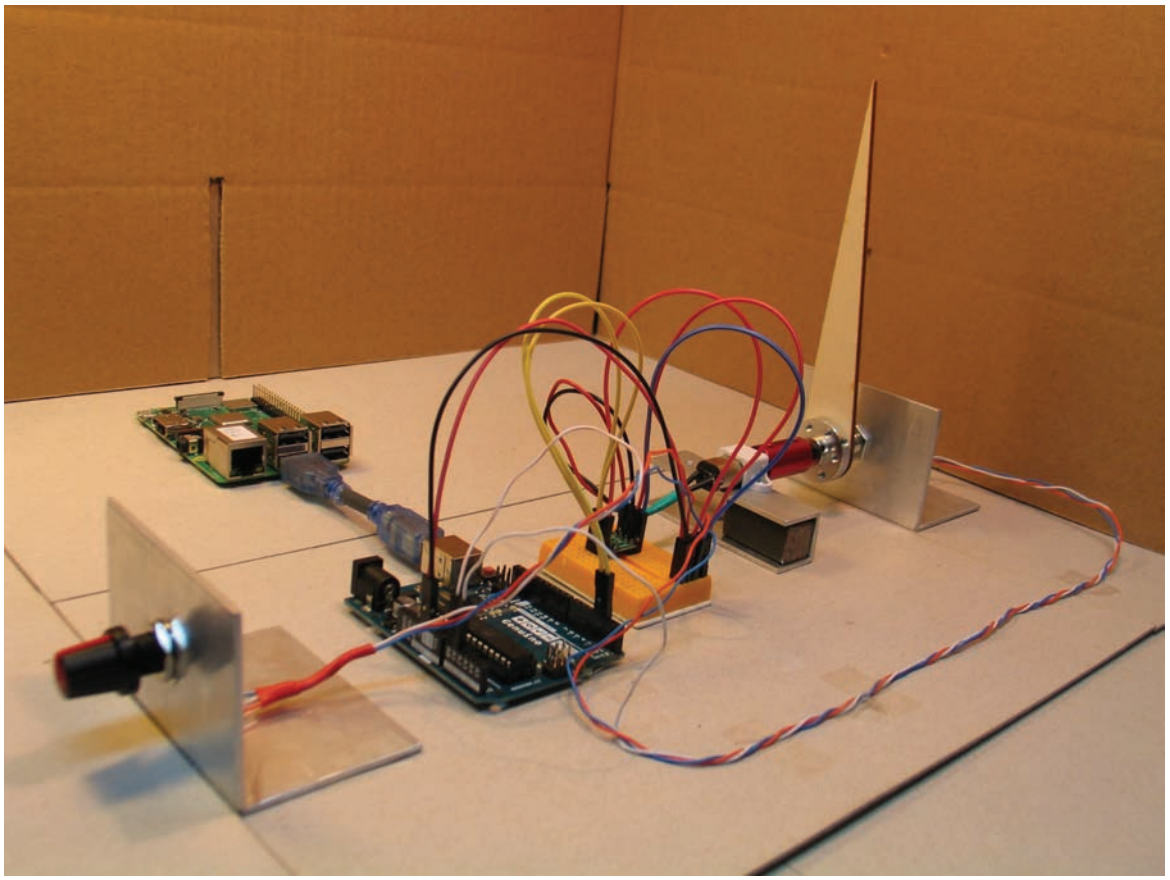
Example M

- Arduino
- rPi Zero w
- H-bridge
- DC motor
- Programming in browser
- Cloud9 IDE
- JavaScript/ ECMAScript

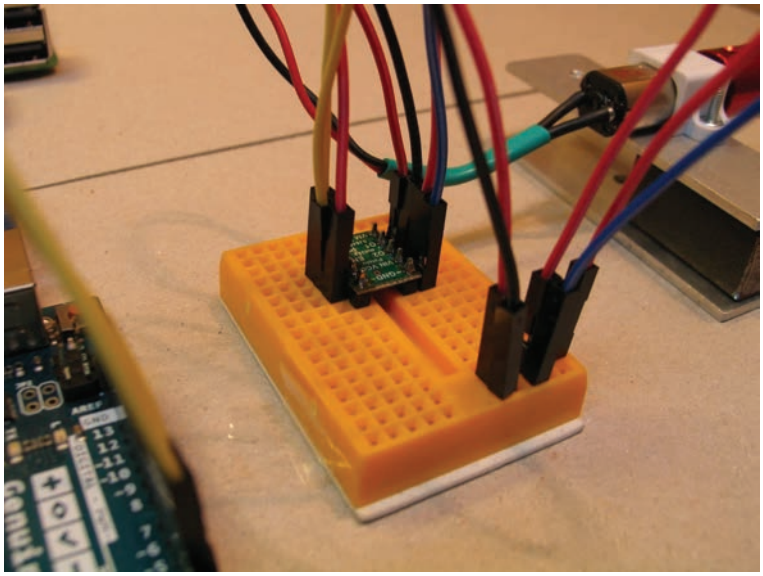


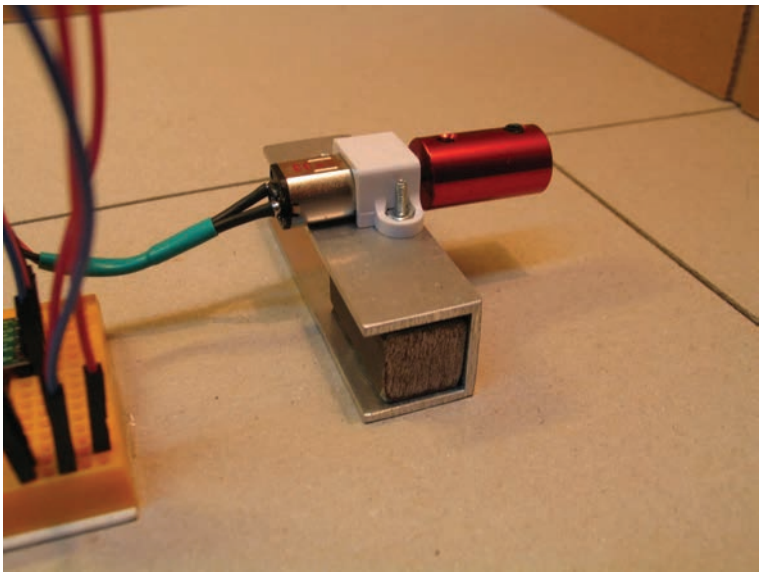
Example M

- Arduino
- rPi 3 Model B
- H-bridge
- DC motor
- Programming in browser
- Cloud9 IDE
- JavaScript/ ECMAScript

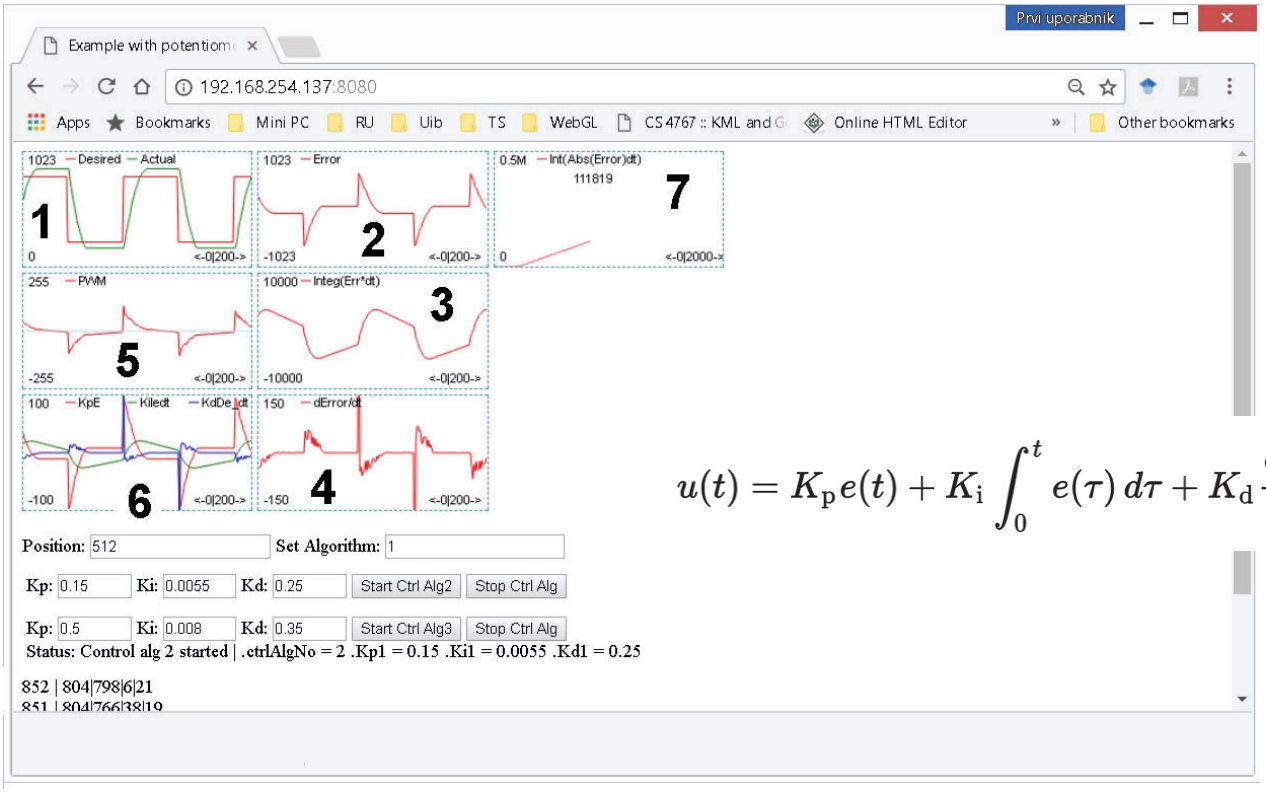


H-bridge and DC Motor Details





GUI for Monitoring Parameters

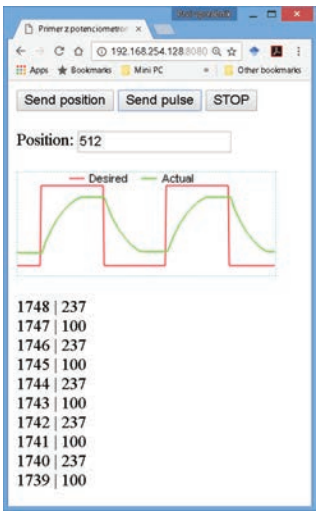


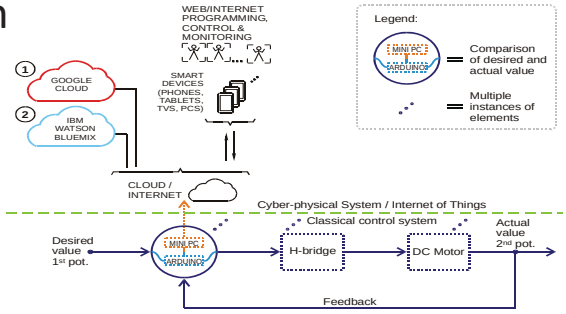
$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(\tau)\,d\tau + K_\mathrm{d} \frac{de(t)}{dt},$$

Application of the Control System in Class

- Relization of several platforms
- Realization of control algorithms with JavaScript
- Study of system response
- PID control algorithm realization









Transporting of Kits

- Kits can be used in standard class
- Power sockets (220V) should be provided
- WiFi network - router
- No other special equipment is needed



Discussion I

- New technologies enable us to develop complex cyber-physical systems based on cloud information systems and edge computing - prototyping
- An important characteristic of cyber-physical systems is that they are tightly integrated with the internet and its users.
- The new paradigm of complex cyber-physical systems development.
- Backed up by edge computing
- From a technical point of view, it is beneficial that several independent cloud service providers exist
- Successfully tested by different users in a clinical environment
- Significant improvements in the CER that correspond to the proposed theoretical model
- JavaScript/ECMA Script & node.js

Conclusion (II.)

- Educational aspects
- Personal realization of control systems by students improves understanding of control theory topics
- Currently technology „in demand", large interest
- Affordability of the technology, also regarding previous knowledge
- Practical application possible in several areas
- Possibility of innovative solutions
- Incorporation in the standard curriculum when addressing control systems, models, state space

Conclusion (III.)

- node.js, JavaScript, C++
- firmata, serial
- Exploration of the Cloud(s), development of algorithms
- Linux
- ESP8266
- ESP32
- Possibility to develop from prototype to full application
- Succesfull development of several prototypes
- Changing learning and technical system design paradigm

Acknowledgement

- This work was supported in part by the Slovenian Research Agency (ARRS) (Research program "Decision support systems in electronic commerce", program No.: UNI-MB-0586-P5-0018), ARRS SI-RF bilateral project "Efficient Control of Cyber-physical Systems & Internet of Things by the Application of Evolutionary and Biologically Inspired Algorithms" Proj. No.: BI-RU/16-18-040, ARRS SI-MNE bilateral project "Development of Speech Controlled Wheelchair for Disabled Persons as Cyber-Physical System" Proj. No.: BI-ME/16-17-022, Proj. NRP No: 3330-22-3515, NOO No: C3330-22-953012 and Erasmus+ Project: 2021-1-MK01-KA220-HED-000027646.



SMART4ALL

An extensive network of Digital Innovation Hubs for boosting technology and business development in South, Eastern and Central Europe

Christos Antonopoulos
University of Peloponnese
SMART4ALL Technical Manager
ch.antonop@uop.gr

Co-funded by the Horizon 2020 programme of the European Union

DT-ICT-01-2019
Smart Anything Everywhere Area 2

www.smart4all-project.eu
Grant Agreement: 872614

SMART4ALL ID Card

Funded by the Horizon 2020 Framework Programme of the European Union

DT-ICT-01-2019: *Smart Anything low energy computing Everywhere* – *Area 2: Customized powering CPS and the IoT*

Project information

SMART4ALL

Grant agreement ID: 872614

Status
Ongoing project

Start date
1 January 2020

End date
31 December 2023

Funded under:
H2020-EU.2.1.1.

Overall budget:
€ 8 660 872,50

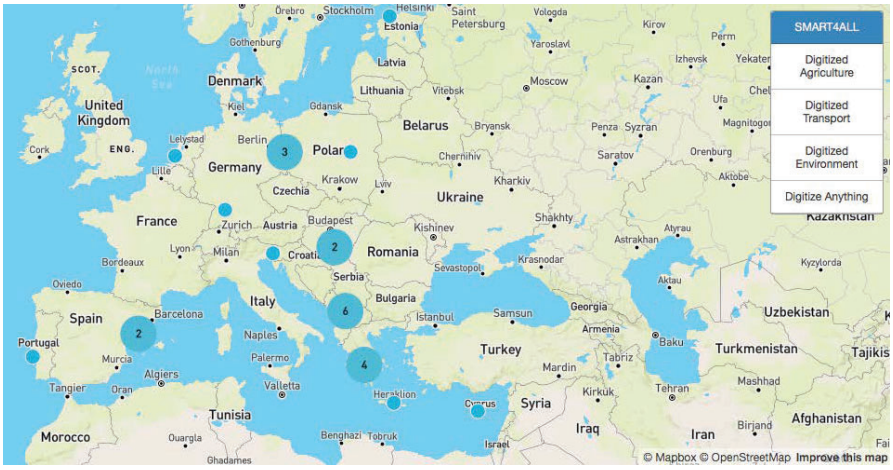
EU contribution
€ 7 997 647,50



Coordinated by:
UNIVERSITY OF PELOPONNESE
Greece

SMART4ALL Consortium

- 25 partners originated from **South**, **Eastern** & **Central Europe**
- The consortium is composed of universities, research institutes, investors, networking organizations, SMEs, Innovation Hubs and NGOs



Diversity in Partners Expertise	Geographic Diversity (15 countries)	
Universities: 10	Albania: 1	Montenegro: 1
Research institutes (R&D): 3	North Macedonia: 2	Poland: 2
Research institutes (BizDev): 1	Cyprus: 1	Portugal: 1
SMEs (R&D): 6	Estonia: 1	Serbia: 1
SMEs (BizDev): 2	Germany: 3	Slovenia: 1
Innovation Hubs: 1	Greece: 5	Spain: 1
NGOs: 1	Hungary: 1	The Netherlands: 1
	Kosovo: 2	

SMART4ALL in a Nutshell

- **Vision:** To build capacity amongst European stakeholders by joining different cultures, different policies, different geographical areas and different application domains
- **How:** Through the development of selfsustained, cross-border experiments that transfer knowledge and technology between *academia* and *industry*
- **Technological Areas:** Customized low energy computing (CLEC) & IoT
- **Application Areas:** Digitized Environment, Digitized Agriculture, Digitized Anything and Digitized Transport

SMART4ALL Objectives

- Unlock South-Eastern parties' potentials through **links to investors** across value chains and regions in order to accelerate CLEC CPS/IoT solutions development and industrialization
- **Develop/offer novel tools** to boost the use of CLEC CPS and the IoT technologies Europewide
- To **ensure post-project sustainability and growth** of the SMART4ALL experiments & DIHs network
- To bring innovative CLEC CPS and the IoT technologies in the lives of **sensitive social groups**
- To support **Digital Skills & Jobs policy of European Commission**

SMART4ALL PAE Types

PAE
Pathfinder Application Experiments

KTEs
Knowledge Transfer Experiments

Funding: 8K Euros
Type: Internship

FTTEs
Focused Technology Transfer Experiments

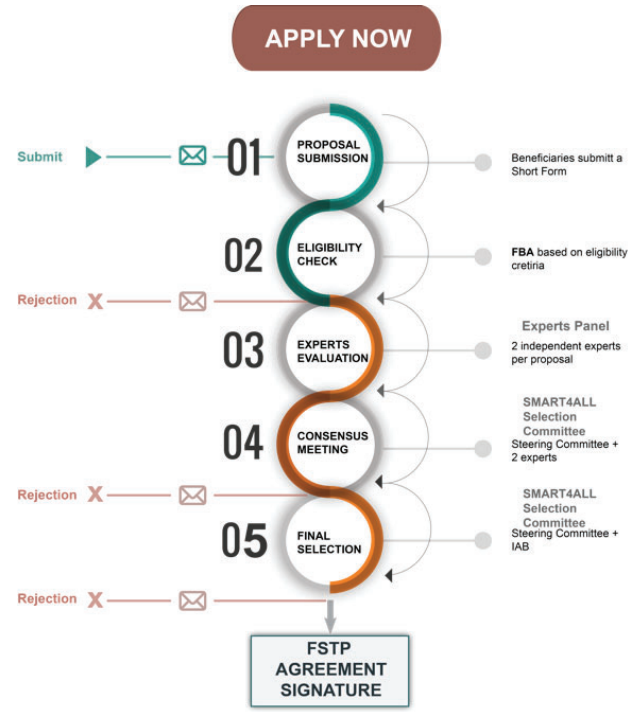
Funding: 80K Euros
Type: Project

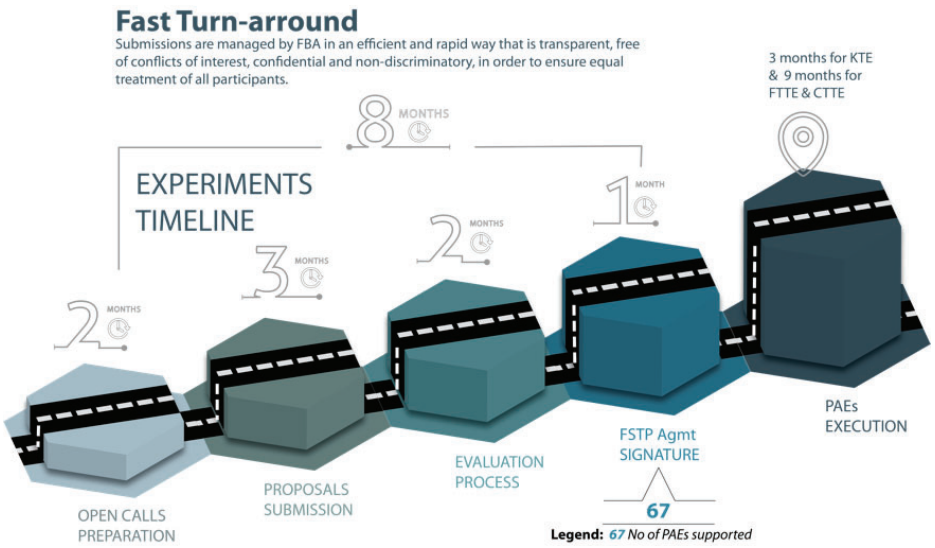
CTTEs
Cross-domain Technology Transfer Experiments

Funding: 80K Euros
Type: Project

- **Internal**: 21 FTTE (vertical) will be developed by SMART4ALL
- **External:** 67 KTEs/FTEs/CCTEs will be developed via open call funding

SMART4ALL PAE Application Stages





SMART4ALL PAE Cut Off Dates

PAE TYPE		Call 1	Call 2	Call 3
Knowledge Transfer Experiments (KTE)	Call Announcement:	Apr 15th, 2020	Mar 2021	Mar 15th, 2022
	Submission Deadline:	Sep 15th, 2020	Jun 15th, 2021	June 15th, 2022
Focused Technology Transfer Experiments (FTTE)	Call Announcement:	Jul 1st, 2020	Jun 15th, 2021	Jul 15th, 2022
	Submission Deadline:	Sep 30th, 2020	Sep 15th, 2021	Oct 17th, 2022
Cross Domain Technology Transfer Experiments (CTTE)	Call Announcement:	Dec 1st, 2020	Oct 15th, 2021	Oct 21th, 2022
	Submission Deadline:	Mar 15th, 2021	Jan 17th, 2022	Jan 17th, 2023

SMART4ALL An Extensive Network of DIHs

An extensive network of Digital Innovation Hubs for boosting technology and business development in South, Eastern and Central Europe

Ecosystem Services:

- Access to SMART4ALL network/ecosystem
- Identify funding opportunities
  - ✓ SMART4ALL Open Calls
  - ✓ Local and European funding frameworks
- Application oriented **ethics coaching**
- Interconnection with other networks/ecosystems



Business Services:

- Application oriented **coaching for business development**
- Partner search and business matchmaking
- Online training courses

Technological Services:

- Application oriented **technological coaching**
- Access to state-of-the-art infrastructures & services
- Technological matchmaking
- Online training courses

SMART4ALL High Performance Computing Center

SMART4ALL offers to the members of its network/ecosystem, via a state-of-the-art **High Performance Computing Center (HPCC)**, the following bouquet of services :

- Software-as-a-Service (SaaS)
- Hardware-as-a-Service (HaaS)
- Scalable architecture to meet high workloads and provide 24/7 availability
- 24/7 support by technology experts
- High speed network interconnection via GRNET backbone
- Open source software employed from virtualization to application layer

SMART4ALL **High Performance Computing Center (HPCC)** is hosted and maintained by

ECE Patras

Electrical & Computer Engineering Department

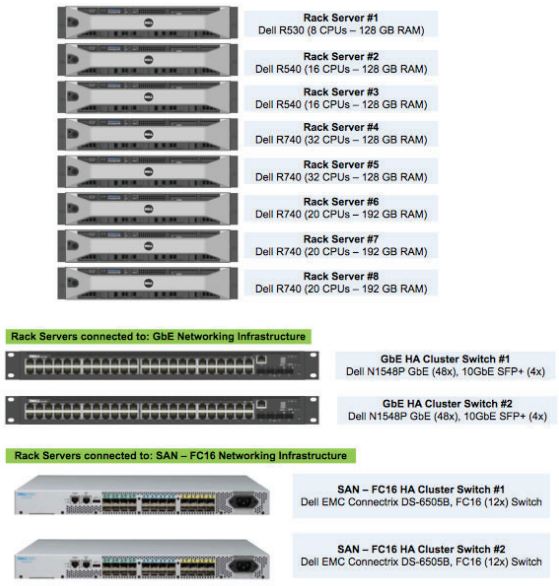
UNIVERSITY OF PELOPONNESE

ESDA LAB

Embedded System Design & Applications Laboratory

SMART4ALL High Performance Computing Center

High Performance Servers with Virtualization Technology



8 Rack Mounted High Performance Servers

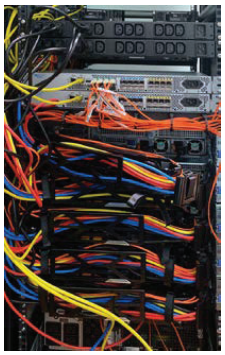
Total CPUs: **164**

Total Memory: **1,216 TBytes**

Type 1 Hypervisors: VMWARE ESXi / Proxmox

Virtualization Technologies

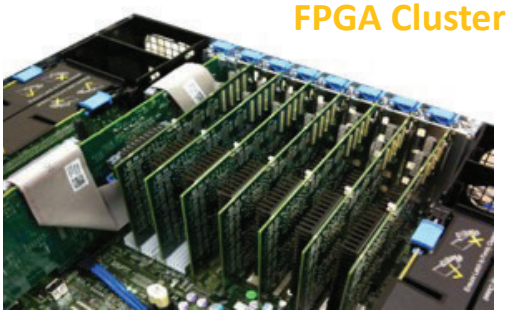




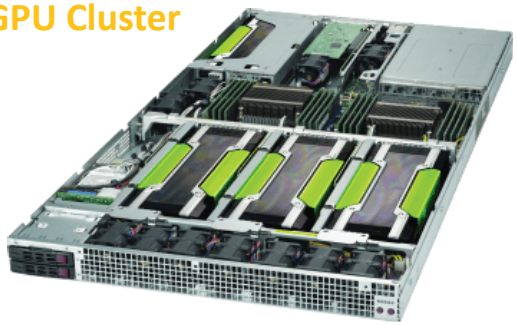
SMART4ALL High Performance Computing Center

High Performance Embedded FPGAs & High-End GPUs

FPGA Cluster



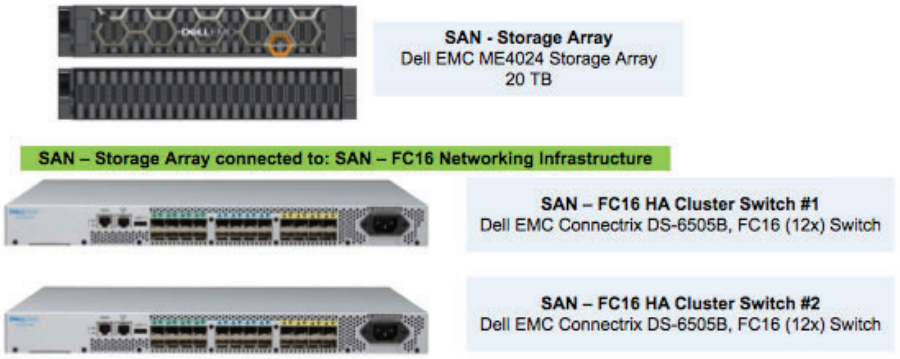
GPU Cluster



A cluster of high-performance embedded FPGAs and high-end GPUs will be available and offered through the SaaS frameworks

SMART4ALL High Performance Computing Center

High Performance Storage Area Networks



Storage Area Network (SAN) - Fibre Channel (FC @ 16 Gbps) Infrastructure

Capacity: **20 TBytes**

SMART4ALL High Performance Computing Center

High Performance Network Attached Storage



NAS - Storage #1
Synology DS 916+
10 TB



NAS - Storage #2
QNAP TS431
20 TB

Network Attached Storage (NAS @ 1 Gbps) Infrastructure
Capacity: **30 Tbytes**

NAS Storage devices connected to: GbE Networking Infrastructure



GbE HA Cluster Switch #1
Dell N1548P GbE (48x), 10GbE SFP+ (4x)



GbE HA Cluster Switch #2
Dell N1548P GbE (48x), 10GbE SFP+ (4x)

SMART4ALL High Performance Computing Center

High Security via a Cluster of Next Generation Firewalls



HA Firewall Cluster – H/W Firewall #1
Fortinet Fortigate-200E NGFW

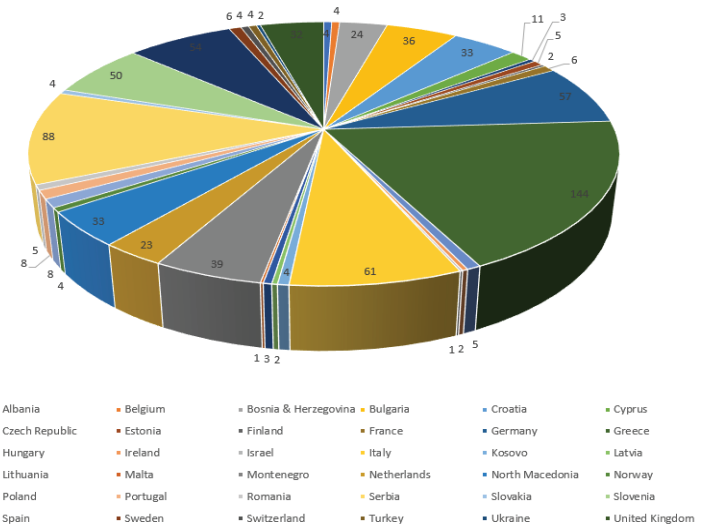
HA Firewall Cluster – H/W Firewall #2
Fortinet Fortigate-200E NGFW

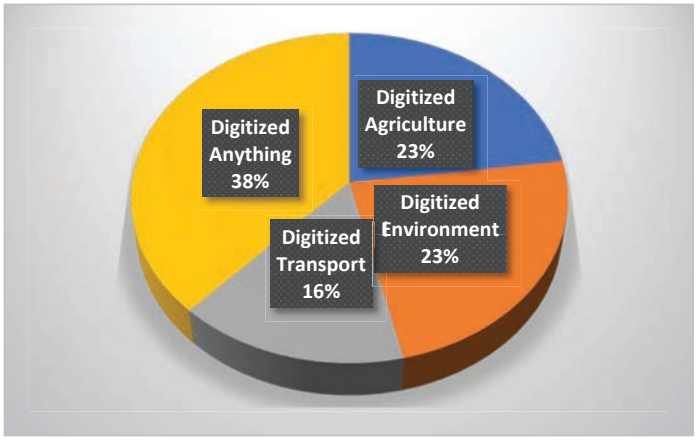
Next Generation Firewalls (NGFW) - Hardware / High Availability Cluster

Open Call Statistics

Applications per Country and Vertical

COUNTRIES (36 in total)



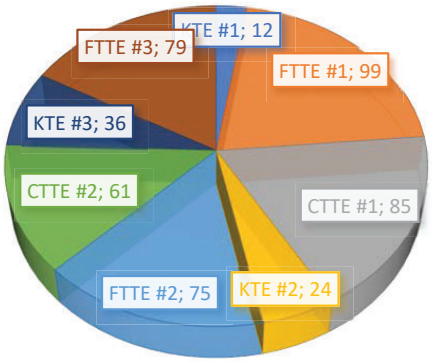


VERTICALS

Open Call Statistics

Allocation per call

APPLICATIONS PER OPEN CALL



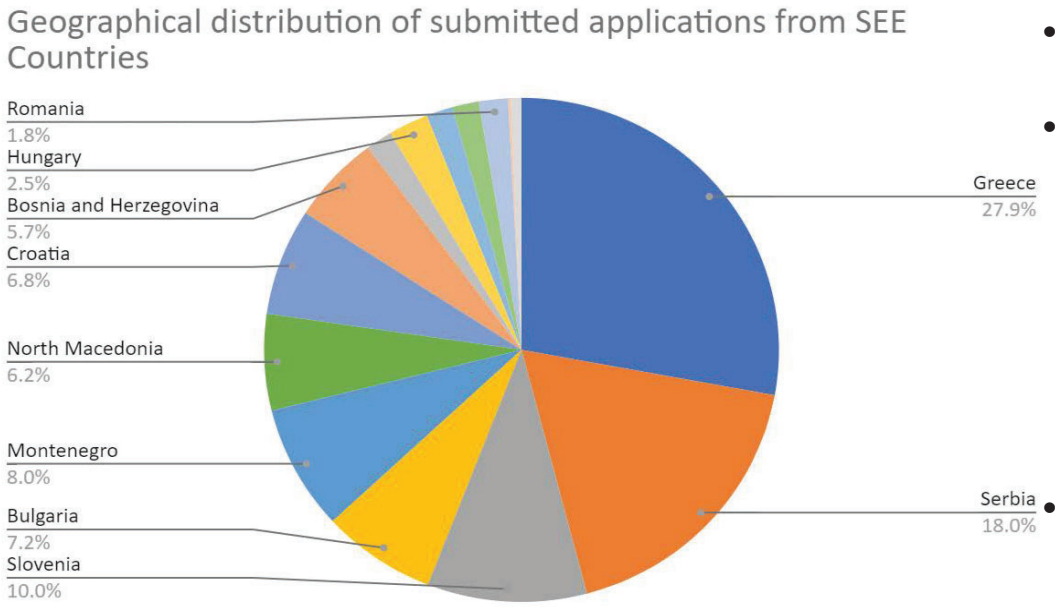
> 500 proposals

PAEs funded

- 67 PAEs have been selected for funding with an overall amount of 2.2M €

- ❑ 43 Knowledge Transfer Experiments – 8K €. 344K in total
- ❑ 12 Focused Technology Transfer Experiments – 80K €. 960K in total
- ❑ 12 Cross-domain Technology Transfer Experiments – 80K €. 960K in total

Applications per country



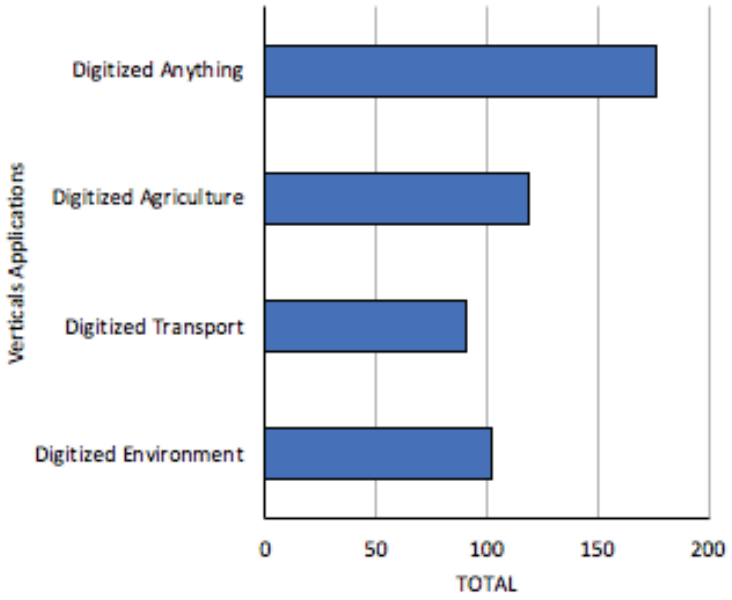
- Applications with a SEE country: 85%
- Top-5 countries regarding application submission:

Greece
Serbia
Italy
Germany
Spain

- Top-5 SEE countries:

Greece
Serbia
Slovenia
Bulgaria
Montenegro

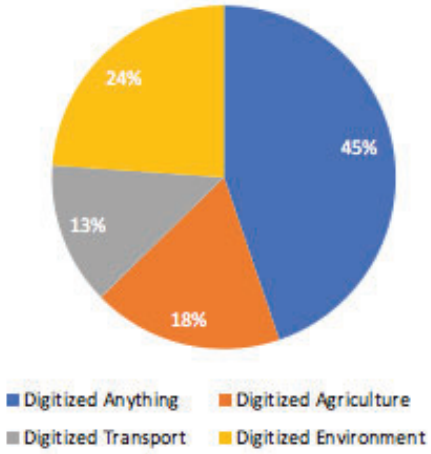
Applications per vertical



Balanced representation of all verticals – Digitized Anything has a priority, due to a wide range of application fields

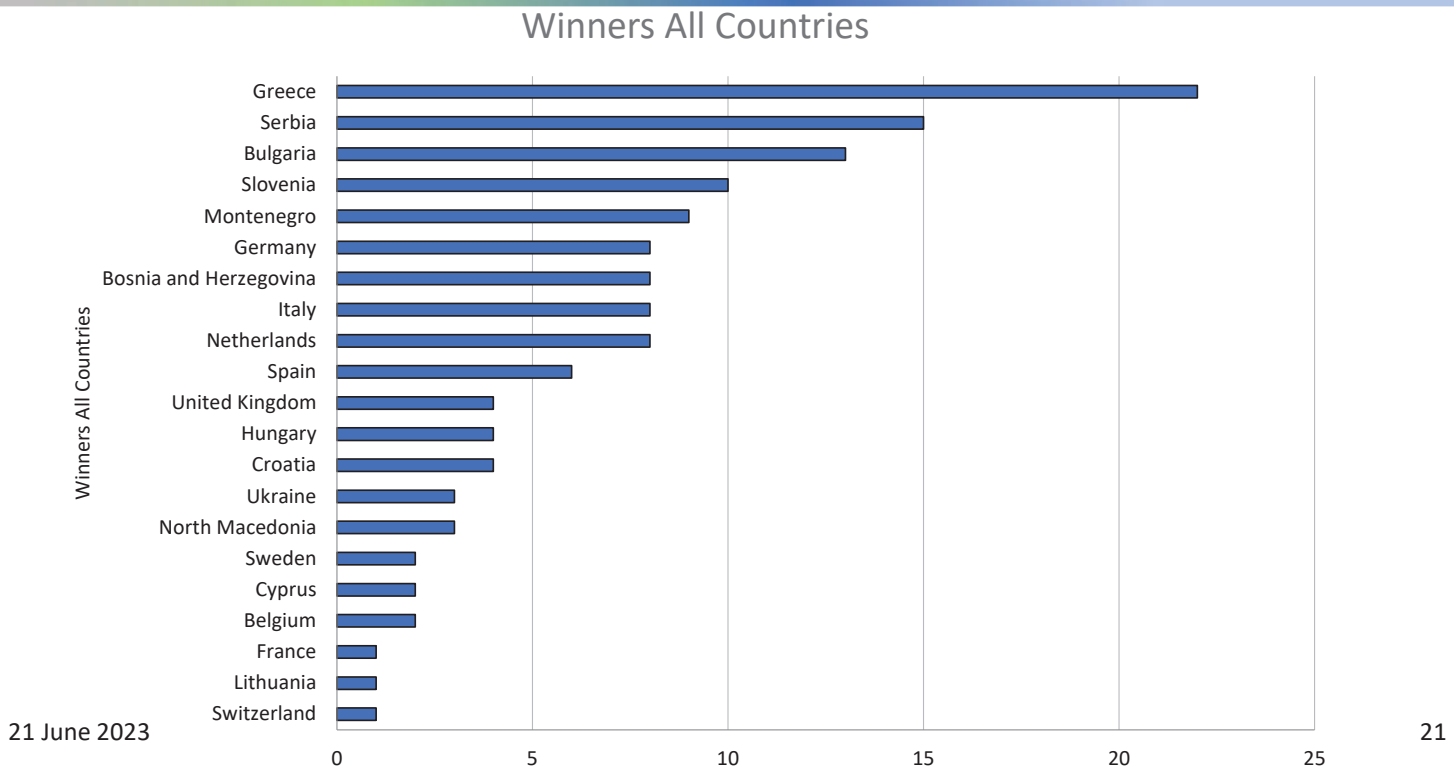
SMART4ALL winners/beneficiaries

- There were 148 partners, getting funded in 67 PAEs and partners from 22 countries

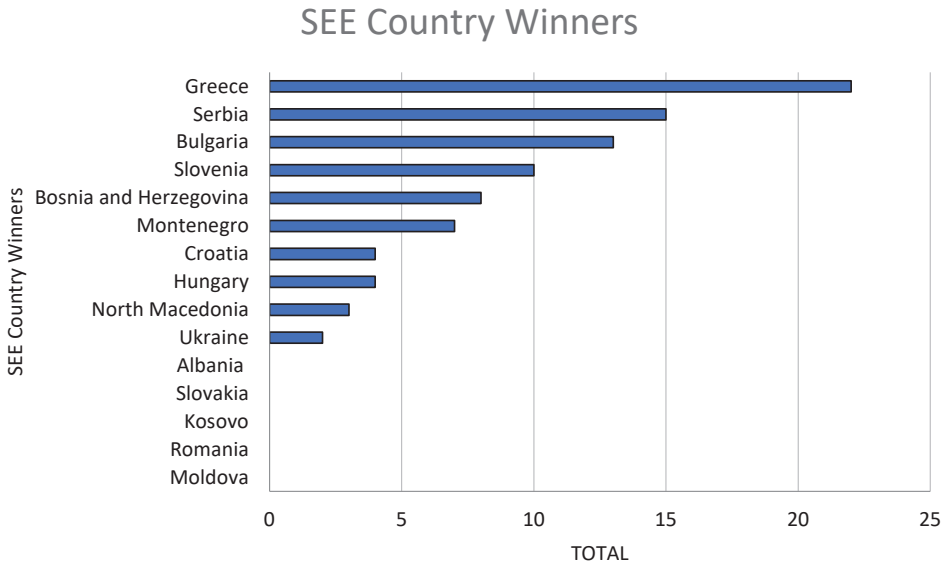


- 16 entities (SMEs and academic institutes) have been selected for more than 1 project in different calls thanks to multiple submissions allowed with a limit on the budget
- Two consortia have received funding in different calls (e.g. TUNNLL). Opportunity to improve products and processes

SMART4ALL beneficiaries/winners (countries)

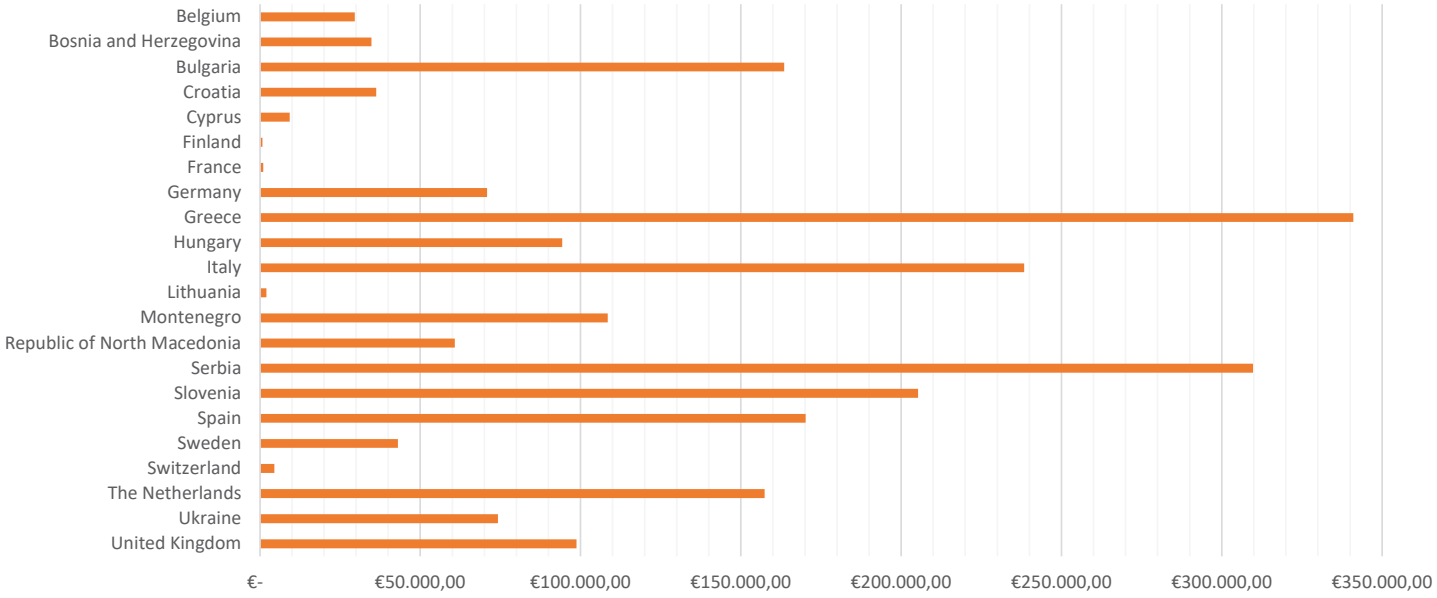


Winners from SEE countries

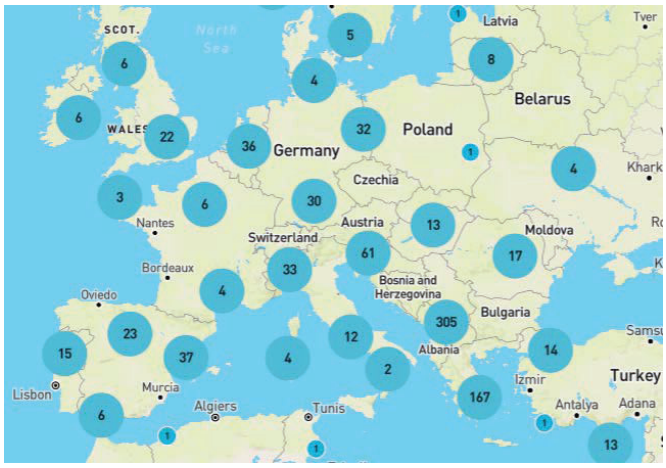


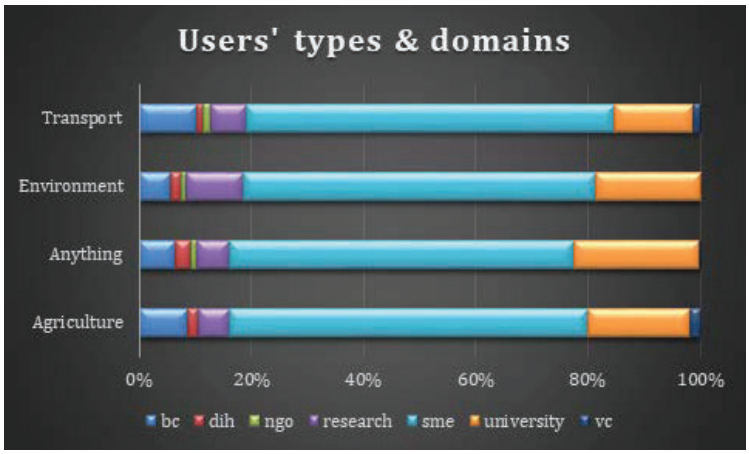
- Top-5 countries of all winners are SEE countries. In specific: Greece, Serbia, Bulgaria, Slovenia, Bosnia & Herzegovina
- Success rates comparing submitted applications and winning projects: 8% - 30%. Bulgaria has the highest score

Funding paid by country



SMART4ALL services (Marketplace)





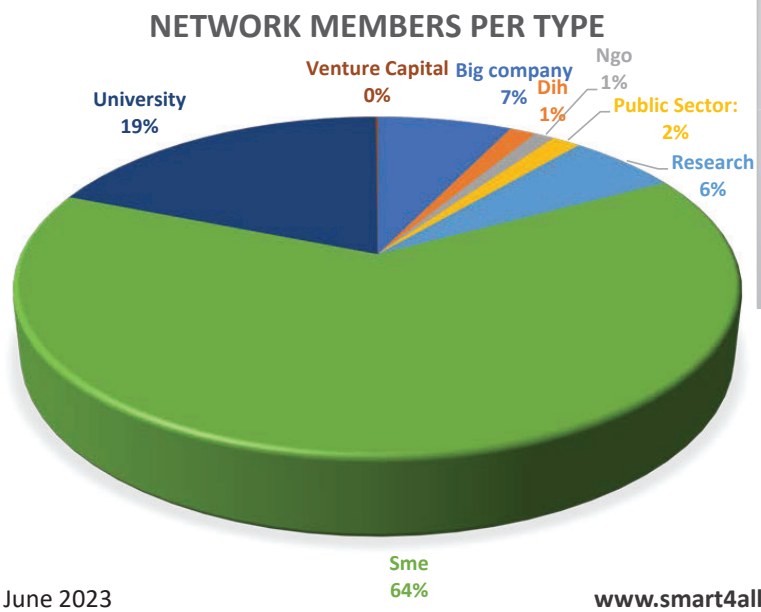
985 Network Members
and keep joining...

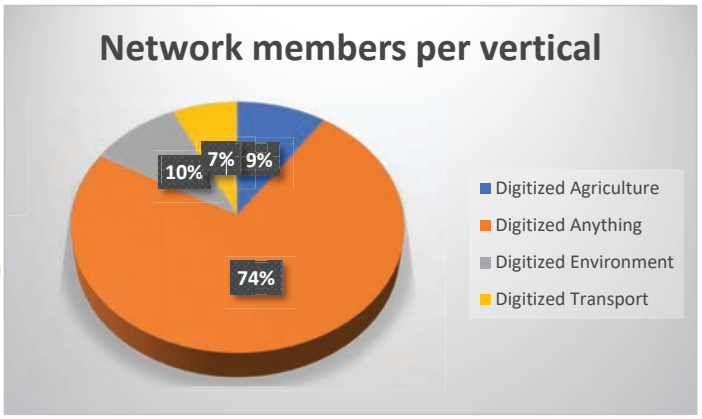
154 Artefacts
and keep increasing...

https://marketplace.smart4all-project.eu/

SMART4ALL Network

985 members





Top-10 countries:
Greece, Spain, Serbia, Germany, Albania, Italy, Montenegro, North Macedonia, UK, Slovenia



Developing Service & Tools towards Self-sustainability: The **SMART4ALL** Marketplace

Co-funded by the Horizon 2020 programme of the European Union

DT-ICT-01-2019
Smart Anything Everywhere Area 2

www.smart4all-project.eu
Grant Agreement: 872614

Project Objectives

To ensure post-project sustainability and growth of the SMART4ALL experiments and DIHs network

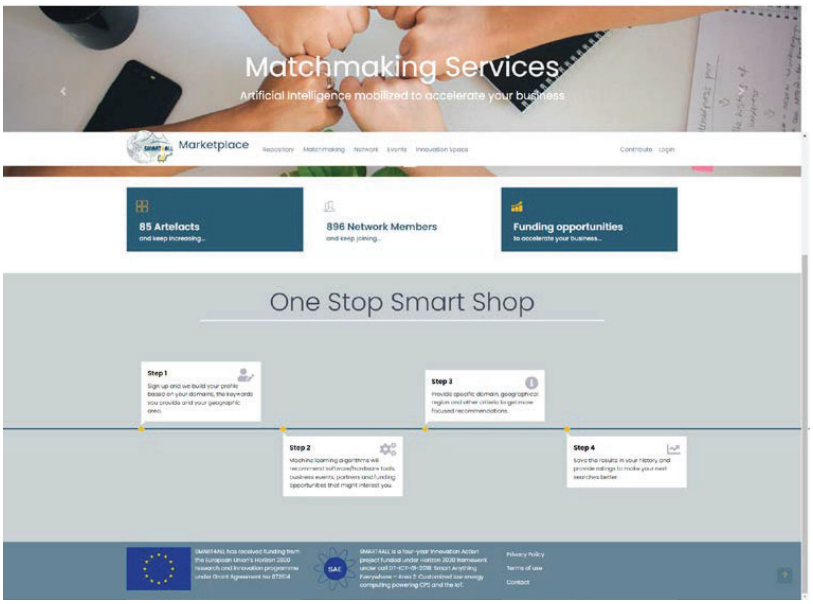
- SMART4ALL MaaS facilitates brokerage of Pathfinder Application Experiments (PAEs)
- Mature service attracting customers in the post-project period.
- Assistance to beneficiaries to access financing for their later-stage development
- Promote community-building
- Matchmaking Services

SMART4ALL Marketplace

Marketplace as a Service Concept is one the cornerstones/flagships of Smart4ALL

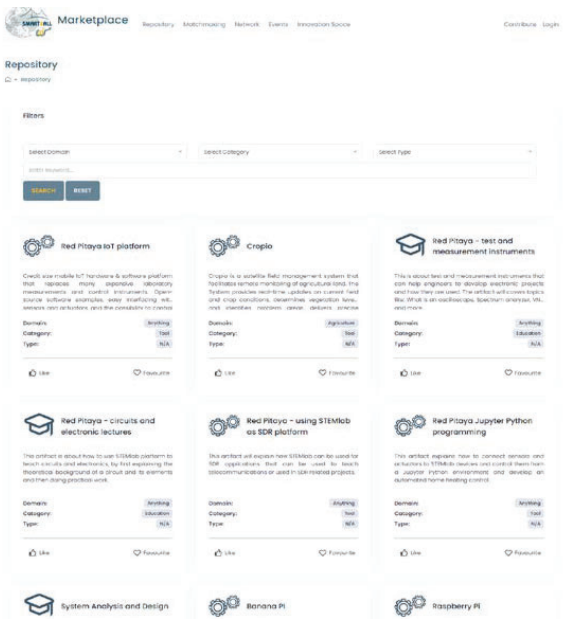
- **What is it ?** A novel one-stop-smart-shop for experts and non-expert third parties seeking (open-source mainly) ICT technologies
- **What is its main target ?** reduce the development time of an startup /SME/mid-cap that is doing business in one of the four SMART4ALL thematic areas
- **Unique selling point:** AI based match-making and collaboration activities will be hosted in the MaaS

SMART4ALL Marketplace



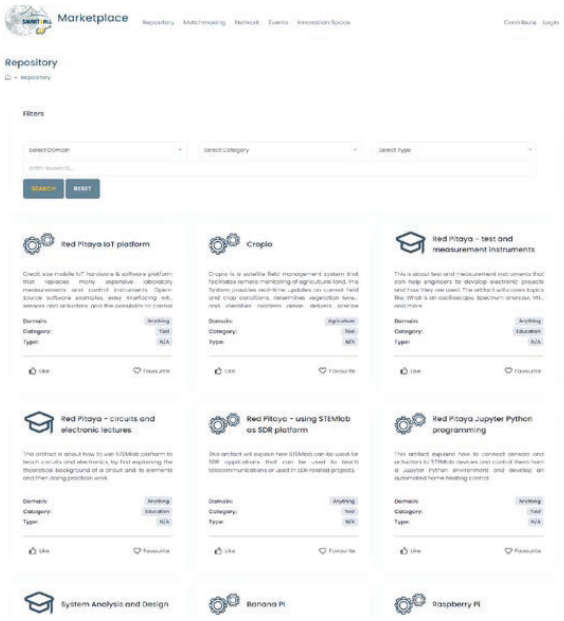
- One Stop Smart Shop
  - SMART4ALL Network
  - Matchmaking Service
  - Repository
  - Innovation Space
  - Events

Marketplace - Repository



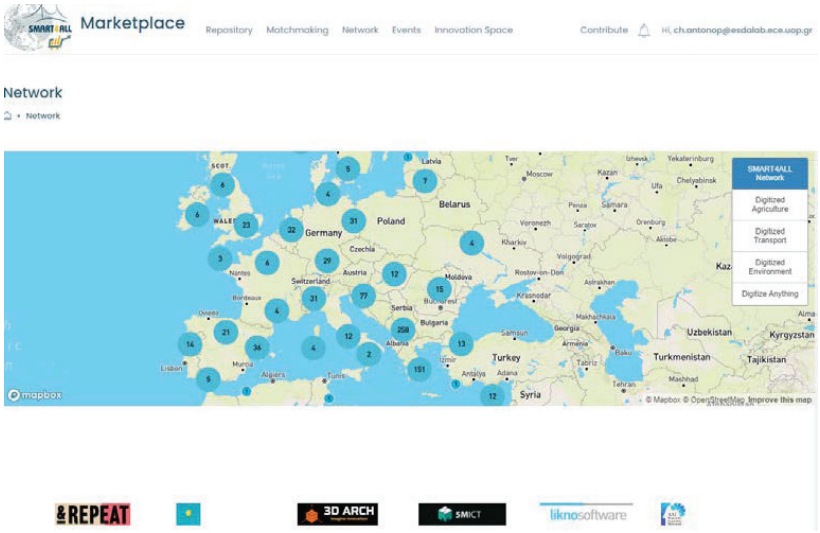
- 176 Artifacts from various domains
  - Open Courses
  - Tools and Web-Services
  - Code
  - Application Notes and Getting-Started Guides
  - Results of SMART4ALL PAEs
- Artefacts by domain:
  - Agriculture 19
  - Anything 127
  - Environment 10
  - Transportation 19

Marketplace - Repository



- Stores multitype artefacts
- Provides customized filtering
- Provides feedback capabilities
- Implemented with modern technologies to provide:
  - Fast retrieval of artefacts by the repository.
  - Flexibility on updates and extensions on the artefact's structure.
  - Logs aggregation

Marketplace – SMART4ALL Network



- **Growing Network of SMART4ALL partners**
  - **985 members**
- **Pillars**
  - **Digitized Agriculture**
  - **Digitized Transport**
  - **Digitized Environment**
  - **Digitized Anything**

Marketplace – SMART4ALL Network

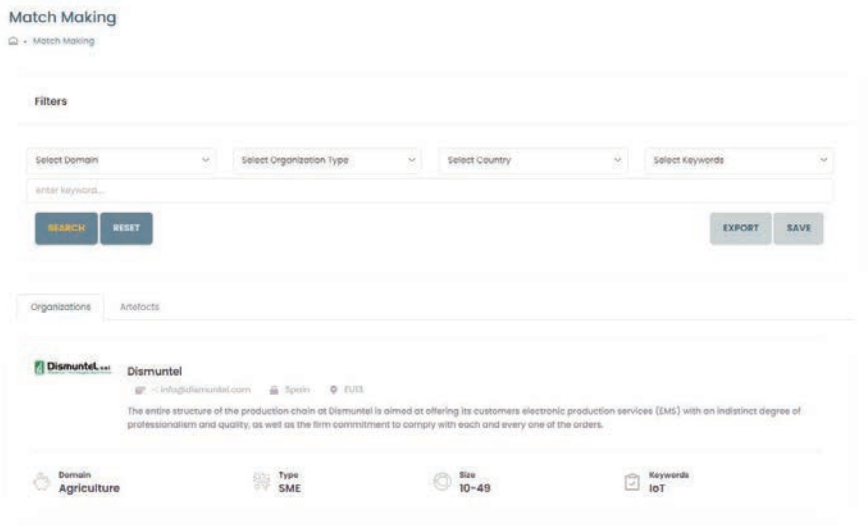




- Points of Interest clustering and filtering
- More info on MouseOver
- Direct navigation on the map using the "Show on the map" button

Marketplace – Matchmaking Service

https://marketplace.smart4all-project.eu/matchmaking

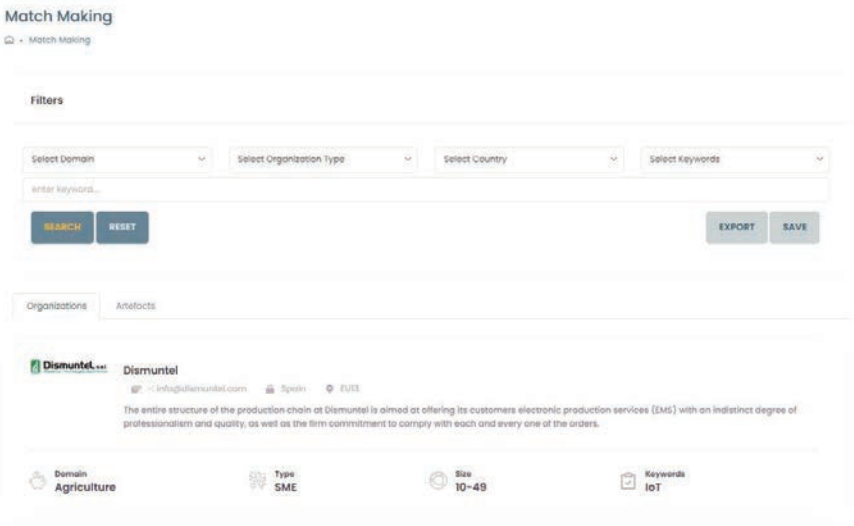


"Manual" Matchmaking

- Filtering based on domain, Type, Area, Keywords
- Multifaceted matchmaking considering suitable entities, artefacts, events (on-going) and relative success stories (on-going)
- Results can be exported and saves for future reference

Marketplace – Matchmaking Service

https://marketplace.smart4all-project.eu/matchmaking



AI Matchmaking

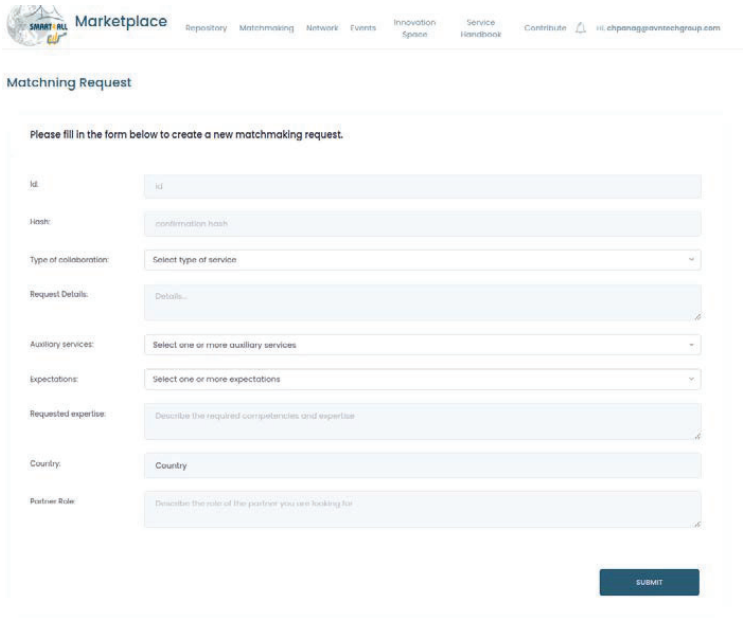
Step 1: Create a user profile based on his description, keywords and sector.
Step 2: Use of machine learning algorithms to suggest software tools, business events, partners and funding opportunities.

- NLP techniques (stemming and lemmatization)
- TF-IDF for description matching score
- Ontologies based on keywords

Step 3: More specialized searches will also be available, for example in another field.
Step 4: The results that look interesting are stored in user's history and can also provide a score that will be used in the future to optimize the recommendations

Marketplace – Matchmaking Service

https://marketplace.smart4all-project.eu/matchmaking



Matchmaking Request List

- All registered users can post a request for collaboration of any framework/type related to SMART4ALL objectives
- Matchmaking will take this list into consideration
- Instant messaging will be supported to facilitate connection

Marketplace - HPC Center Service (1)

High Performance Servers with Virtualization Technology

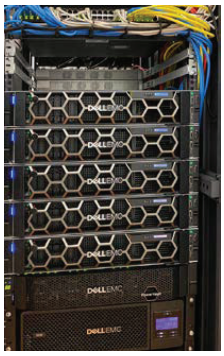
8 Rack Mounted High Performance Servers

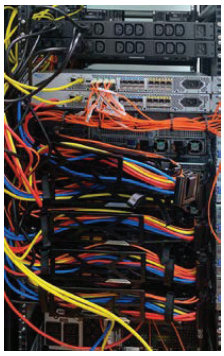
Total CPUs: **164**

Total Memory: **1,216 TBytes**

Type 1 Hypervisors: VMWARE ESXi / Proxmox

Virtualization Technologies

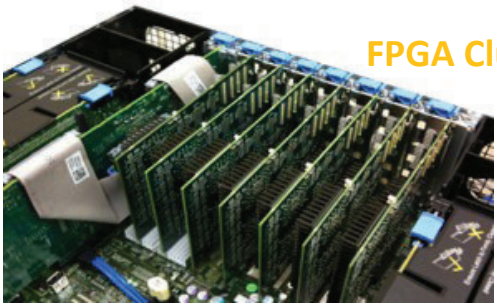




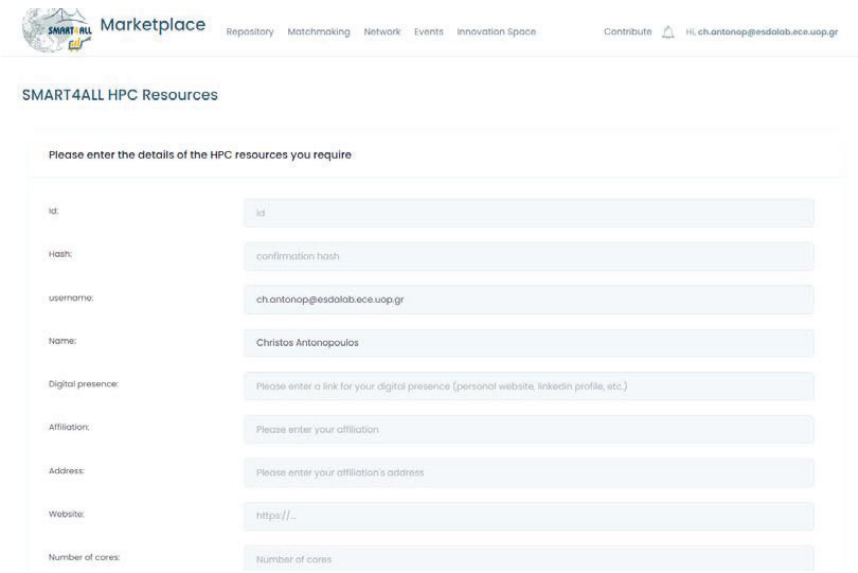
GPU Cluster



FPGA Cluster



Marketplace - HPC Center Service (2)



Resource Sharing

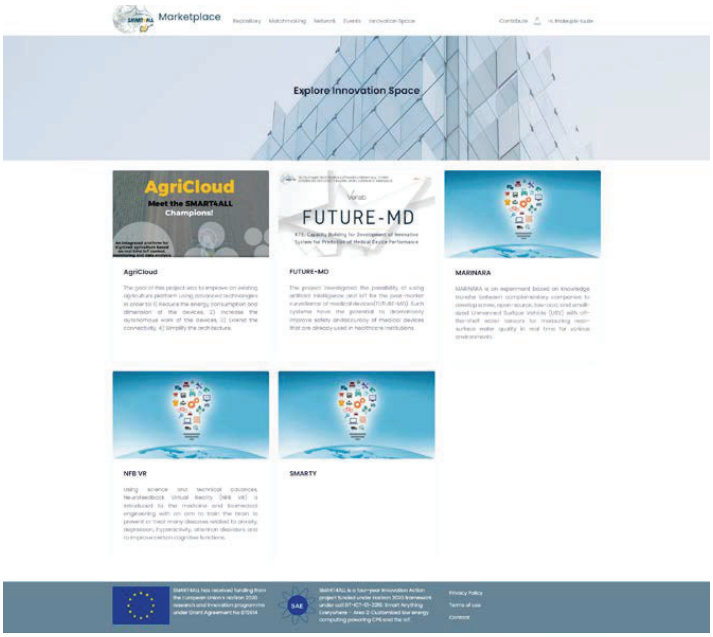
Step 1: A form is completed by the candidate indicating the resource and purpose

Step 2: Request is evaluated by SMART4ALL committee

Step 3: Resources are prepared in the adequate format e.g. VM, Docker etc.

Step 4: Resources are commissioned and monitored

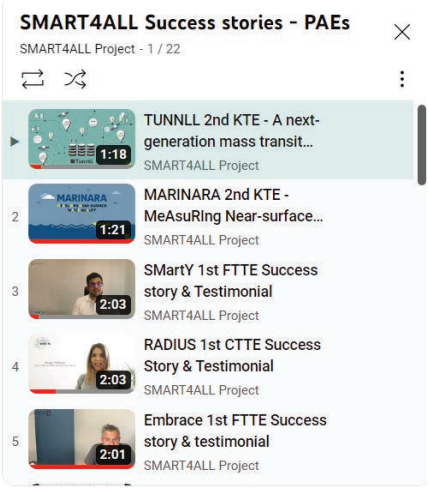
Marketplace – Innovation Space



- **Showroom for the PAEs**
  - **Opportunity to present the projects and the partners**
  - **SMART4ALL experts help with the creation of the presentation**
- **Matchmaking will include relative results**
- **Instant messaging will be provided to facilitate rapid interconnection**

Success stories showcasing

- Innovation Space: https://marketplace.smart4all-project.eu/innovation
- Success stories videos – playlist on YouTube
- Section on SMART4ALL website
- Articles publicized on HiPEAC magazine







Testimonials

"Being part of the SMART4ALL gave us knowledge of how to do a final optimization to make a product ready for the market"

"The SMART4ALL Repository offered great insights about possible technologies and services that can be used"

"Our primary benefit is the capacity building… 2-3 newly-created specialist jobs are foreseen in support of the increased service"

"SMART4ALL can act as a catalyst for complementary companies"





Marketplace – Digitization Maturity Assessment Service

Under Development

- **What is this service** → A tool indicating a roadmap for future digitalization within the services offered by SMART4ALL
- **Why it can be useful** → Identify key digitization aspects and especially bottlenecks and deficiencies
- **How will it work** → A set of questions to define the SMEs level of digitalization and profile
  - Get more information on what they want to do: e.g Digitalization in agriculture in a small company in Macedonia for instance.
  - The data collected will give us more info that our system will propose something like the following:
    - Current state of the SME in terms of digitalization: Level of digitalization, HR for digitalization, Know-how/expertise
    - Current state of SME in terms of business/finance: development strategies, investments in IT, access to funds/ availability of funds
    - Profile of the need: Abstract of the need, Selection of one or more verticals from SMART4ALL, keyword search from application notes with similarity of the need.
- **Output** → Digitization roadmap
  - Here's are few examples of implementation (refer to PAEs or internal PAEs artefacts) of similar projects
  - Here are few items on the marketplace that you can make use of in your idea: List of marketplace items and WIKIS from Technology Handbook
  - Here are some partners that may be interested in your project (expertise): List of partners from our ecosystem
  - Here are some potential funds available: Access to funds list from WP4

Join now SMART4ALL Network !

Join now SMART4ALL Network in South, Eastern and Central Europe by:

- Subscribing to SMART4ALL newsletter: https://smart4all-project.eu/joinus
- Using SMART4ALL matchmaking services: https://marketplace.smart4all-project.eu/matchmaking
- Applying to SMART4ALL Open Calls: https://smart4all-project.eu/opencalls-apply-now
- Employing SMART4ALL marketplace: https://marketplace.smart4all-project.eu/



SMART4ALL Social Media

Follow SMART4ALL on:

https://www.linkedin.com/company/smart4all-h2020/

https://twitter.com/Smart_4All

https://www.facebook.com/SMART4ALL.Project

https://www.youtube.com/channel/UCwmSl9LCl2vNBO-3k75dvJA

… and stay up to date for the latest SMART4ALL news, activities & funding opportunities !

SMART4ALL Contact persons

Project Coordinator



Prof. Nikolaos Voros

University of Peloponnese, Greece

voros@uop.gr

Central Europe Contact



Prof. Michael Huebner

Brandenburg University of Technology, Germany

huebner@b-tu.de

East Europe Contact



Prof. Tamás Kovácsházy

Budapest University of Technology and Economics, Hungary

khazy@mit.bme.hu

Balkan Contacts



Prof. Radovan Stojanovic

Mediterranean Excellence in Computing and Ontology, Montenegro

stox@ac.me

South Europe Contacts



Prof. Fransisco Blanes

Polytechnic University of Valencia, Spain

pblanes@ai2.upv.es

Thank you for your attention – Questions?



Christos Antonopoulos
University of Peloponnese
SMART4ALL Technical Manager
ch.antonop@uop.gr

CPS&IoT'2023 Summer School on Cyber-Physical Systems and Internet-of-Things

Budva, Montenegro, June 6-10, 2023

Schedule

Day 1, Tuesday 6 June:
08:00-09:00 Registration
09:00-09:15 Event Chairs and Special Guests
Title: Opening Ceremony of the CPS&IoT'2023 Summer School, and MECO'2023 and CPS&IoT'2023 Conferences
09:15-10:15 Andreas Burger, Robert Bosch GmbH, **DE**
Keynote: **Future Cyber-Physical Systems – between Digital Twins and Metaverse** **10.15-10.30 Break**
10.30-11.00 Lech Jóźwiak, TU/e, **NL**
Title: Introduction to the CPS&IoT'2023 Summer School
11.00-12.30 Lech Jóźwiak, TU/e, **NL**
Title: Green CPS and IoT for Green World
12.30-14.00 Lunch Break
14.00-15.30 Sakir Sezer, NVIDIA, **UK**
Title: DPU Technology: Redefining next generation Datacenter Security
15.30-17.00 Kevin Mika, Uni-Bielefeld, **DE** and Piotr Zierhoffer, Antmicro, **PL**
Title: VEDLIoT – Accelerated AIoT
17.00-17.30 Break
17.30-19.00 Aizea Lojo and Juan Besga, Ikerlan, **ES;** Iñaki Paz, LKS, **ES;** Gianluca Brilli and Paolo Burgio, from Unimore**, IT;** Martin Matschnig, Siemens, **AT**
Title: FRACTAL: A Cognitive Computing Platform Node for the Edge
Day 2, Wednesday 7 June:
09.00-10.00 Jürgen Becker, KIT, **DE**
Title: **Keynote**: **Embedded reliable HPC enabling Automotive & IoT Intelligence**
10:15-10:45 BREAK
10.45-12.15 Mario DIAZNAVA, ST Grenoble, **FR**
Title: ANDANTE: AI for New devices and Technologies at the Edge
12.15-13.45 Tobias Dörr and Florian Schade, KIT, and Alexander Ahlbrecht, DLR, **DE**
Title: Simulation-based development of networked avionics systems using the XANDAR toolchain
13.45-14.15 Lunch Break
14.15-17.15 Dominique Blouin, Telecom Paris, and Anish Bhobe, Institut Polytechnique de Paris, **FR**
Rakshit Mittal, University Antwerpen, **BE**
Title: Embedded Systems Modeling, Analysis and Automatic Code Generation with AADL and RAMSES (hands-on)
17.15-17.45 Break
17.45-19.15 Hector Posadas, Jose Luis Vazquez and Daniel Suarez, TEISA/UNICAN, **ES**
Title: Offloading and Reconfiguration of Functional Components in Embedded Hardware Resources: Application to AI Components
21.00 Gala Dinner
Day 3, Thursday 8 June:
09.00-10.15 Naim Dahnoun, University of Bristol, **UK**
Title: **Keynote**: Enhancing Remote and Face-to-Face Learning: Real-Time Feedback System for Teaching Platforms
10.15-10.45 Break
10.45-13.45 Dimitris Karadimas, AVN Innovative Technology Solutions, **CY**
Title: XANDAR: CI/CD automations for complex pipelines with hardware in the loop
13.45-14.45 Lunch Break
14.45-17.150 Morayo Adedjouma and Réda Nouacer, CEA, **FR**
Title: From Knowledge-based to Model-based Engineering of embedded intelligence and trust in CPS
17.15-17.45 Break
17.45-19.15 Zakaria Chihani and Réda Nouacer, CEA, **FR**
Title: Towards Characterizing AI Trustworthiness
Day 4, Friday 9 June:
09.00-10.30 Christoph Schmittner, AIT, **AT**
Title: Security Engineering for Aquaponic – Challenges between Industrial Control and Farming
10:30-11:00 Break
10.00-12.15 Radovan Stojanovic, University of Montenegro and MECOnet, **ME**
Title: Design of performance and energy efficient nodes for smart systems
12.15-13.30 Lunch Break
13.30-15.00 Ramiro Samano Robles, ISEP, **PT**
Title: Evolution and evaluation of wireless technologies and standards towards the convergence of 5G/6G and IoT
15.00-16.30 Andrej Škraba, University of Maribor, **SI**
Title: Overview of Several CPS&IoT Prototypes: Wheelchair, Group Heart Rate Monitoring, PID DC Motor Control
16.30-17.00 Break
17.00-18.30 Presenter Christos Antonopoulos, UoP, University of Peloponnese, **GR**
Title: SMART4ALL project: Customized Low-Energy Computing for CPS
18.30-19.00 Closing of the CPS&IoT'2023 Summer School
+ Free participation in sessions of the CPS&IoT'2023 Conference and MECO'2023 Conference Summer School participants are expected to come with their own laptops. Internet access will be guaranteed.
Participants of the tutorial "VEDLIoT – Accelerated AIOT" are requested to bring their own laptop in order to take part in the tutorial. It is recommended to use a Linux – based operating system, or Linux compatible systems like WSL (Windows Subsystem for Linux).
Day 5, Saturday 10 June: Excursion possible (excursion fee is not included in the summer school fee)

4th Summer School on Cyber Physical Systems and Internet of Things - SS-CPSIoT'2023

4th Generation (Students and Teachers)

Budva, Montenegro, 06-10.06.2023

Summer School on CPS&IoT 2023 – ALL			
#	**Students**	**Country**	**Affiliation**
1	Saleha Javed	Sweden	Luleå University of Technology
2	Rajeev Kanth	Finland	University of Turku
3	Aldin Beriša	Sweden	Mälardalens University
4	Yining Huang	France	Université Paris-Saclay, CEA, List
5	Bo Peng	New Zealand	University of Auckland
6	Inhwi Hwang	South Korea	Seoul National University
7	Mikel Garcia	Spain	LKS Next
8	Boris Radovanović	Serbia	University of Novi Sad
9	Mikael Ebrahimi Salari	Sweden	Mälardalen University, VeriDevOps
10	Hamza Ouhaichi	Sweden	Malmö University
11	Irman Faqrizal	France	Université Grenoble Alpes
12	Diana Deac	Belgium	Vrije Universiteit Brussel; Technical University of Cluj-Napoca
13	Roald Van Glabbeek	Belgium	Vrije Universiteit Brussel
14	Mikhail Fomichev	Germany	Technical University of Darmstadt
15	Tommaso Puccetii	Italy	University of Florence
16	Przemysław Hołda	Poland	Warsaw University of Technology
17	Kajetan Rachwał	Poland	Warsaw University of Technology
18	Björn Leander	Sweden	Mälardalen University
19	Pavlo Tokariev	France	Université Cote d'Azur, CNRS, Inria I3S
20	Mohamed Ahmed M. Hail	Germany	University of Lübeck, Institute of Telematics, Lübeck Germany
21	Vahid Saranirad	United Kingdom	Ulster University, Smart Manufacturing Data Hub (SMDH)
22	Thodoris Skandamis	Greece	University of Peloponnese
23	Helga Sallaku	Montenegro/Albania	University of Montenegro and University of Shkodra
24	Jovan Đurković	Montenegro	MECOnet
#	**Presenters**	**Country**	**Affiliation**
1	Andreas Burger	Germany	ROBERT BOSCH GMBH
2	Lech Jóźwiak	Netherlands	TU/e
3	Sakir Sezer	United Kingdom	NVIDIA
4	Kevin Mika	Germany	Uni-Bielefeld
5	Piotr Zierhoffer	Poland	Antmicro
6	Aizea Lojo	Spain	Ikerlan
7	Juan Besg	Spain	Ikerlan
8	Iñaki Paz	Spain	LKS
9	Gianluca Brilli	Italy	Unimore
10	Paolo Burgio	Italy	Unimore
11	Martin Matschnig	Austria	Siemens
12	Jürgen Becker	Germany	KIT
13	Mario Diaz Nava	France	ST Grenoble
14	Tobias Dörr	Germany	KIT
15	Florian Schade	Germany	KIT
16	Alexander Ahlbrecht	Germany	DLR
17	Dominique Blouin	France	LTCI Lab, Telecom Paris - Institut Polytechnique de Paris
18	Anish Bhobe	France	LTCI Lab, Telecom Paris - Institut Polytechnique de Paris
19	Rakshit Mittal	Belgium	University Antwerpen
20	Etienne Borde	France	LTCI Lab, Telecom Paris - Institut Polytechnique de Paris
21	Hector Posadas	Spain	TEISA/UNICAN
22	Jose Luis Vazquez	Spain	TEISA/UNICAN
23	Daniel Suarez	Spain	TEISA/UNICAN
24	Naim Dahnoun	United Kingdom	University of Bristol
25	Dimitris Karadimas	Cyprus	AVN Innovative Technology Solutions
26	Morayo Adedjouma	France	CEA
27	Luis Palacios	France	CEA
28	Réda Nouacer	France	CEA
29	Zakaria Chihani	France	CEA
30	Christoph Schmittner	Austria	AIT
31	Radovan Stojanovic	Montenegro	University of Montenegro and MECOnet
32	Ramiro Samano Robles	Portugal	ISEP
33	Andrej Škraba	Slovenia	University of Maribor
34	Christos Antonopoulos	Greece	University of Peloponnese

Certificate of Attendance

THIS ACKNOWLEDGES THAT

Marko Markovic

Montenegrin Association for New Technologies - MANT, Montenegro

has successfully attended and completed

The 4th Summer School on

Cyber Physical Systems and Internet of Things (SS-CPSIoT'2023)

(3 ECTS)

in Budva, Montenegro, June 06-10 2023

On behalf of the organizers:

Prof. dr. Lech Jozwiak — *Prof. dr. Betim Cico*

Prof. dr. Radovan Stojanović — *Prof. dr. Nikolaos Voros*

Author Index

Adedjouma Morayo, 513
Ahlbrecht Alexander, 279
Antonopoulos Christos, 906

Becker Jürgen, 232
Besga Juan, 155
Borde Etienne, 316
Bhobe Anish, 316
Blouin Dominique, 316
Brilli Gianluca, 155
Burgio Paolo, 155

Chihani Zakaria, 541

Diaz Nava Mario, 233
Dörr Tobias, 279

Jóźwiak Lech, 4

Karadimas Dimitris, 422

Lojo Aizea, 155

Matschnig Martin, 155
Mika Kevin, 134
Mittal Rakshit, 316

Nouacer Réda, 513

Palacios Luis, 513
Paz Iñaki, 155
Posadas Hector, 368

Samano Robles Ramiro, 767
Schade Florian, 279
Sezer Sakir, 106
Škraba Andrej, 862
Stojanović Radovan, 731
Suarez Daniel, 368

Vazquez Jose Luis, 368

Zierhoffer Piotr, 134

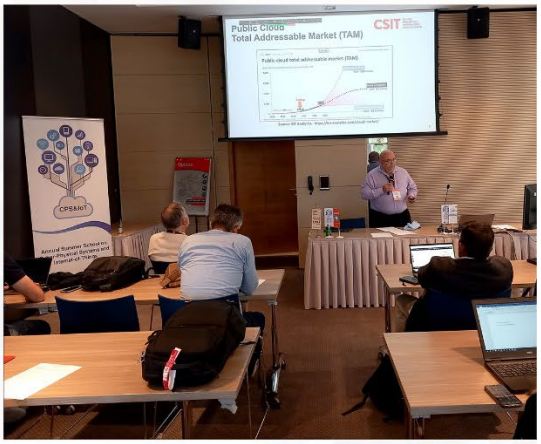
SS-CPS&IoT 2023 GALLERY















CPS&IoT'2023 4th Summer School on Cyber-Physical Systems and Internet-of-Things
Budva, Montenegro, June 6-10, 2023
Citation:
Author/s, "Title of contribution-pressentation", in Proceedings of the 4th Summer School on Cyber-Physical Systems and Internet-of-Things, Editors: Lech Jozwiak, Radovan Stojanovic and Christos Antonopoulos, Vol. IV, June 2023, pp. xx-yy, DOI: Assigned in Zenodo and ResearchGate

Technical editors:

Prof. dr Budimir Lutovac, University of Montenegro

Jovan Djurkovic, MECOnet

Publishers:

MECOnet Institute (Mediterranean Excellence in Computing and Ontology), www.meconet.me

Montenegrin Association for New Technologies - MANT, www.mant.me



Year/Place: Montenegro, Podgorica, 2023