SS-CPS&IoT'2024

5th Summer School on Cyber-Physical Systems and Internet-of-Things

Budva, Montenegro, June 11-14, 2024

Proceedings of the 5th Summer School on Cyber-Physical Systems and Internet-of-Things

Vol. V

Editors:

Lech Jóźwiak
Eindhoven University of Technology, The Netherlands

Radovan Stojanović
University of Montenegro, MECOnet, Montenegro

Authors:

Alberto Marchisio, Andrej Škraba, Anish Bhobe, Claudio Rubattu, Dominique Blouin, Federico Manca, Francesco Ratto, Hans Vangheluwe, Jovan Đurković, Lech Jóźwiak, Luis Palacios, Morayo Adedjouma, Muhammad Shafique, Nabil Abdennadher, Nikhil Gaikwad, Radovan Stojanović, Rakshit Mittal, Ralf Lübben, Rizwan Parveen, Samir Ouchani, Sokol Kosta and Zakaria Chihani.

MECO Events and Publishing
MECOnet, MANT
Montenegro, June 2024

Disclaimer

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

The official website of the event:
https://mecoconference.me/ss-cpsiot2024/

Proceedings of the 5th Summer School on Cyber-Physical Systems and Internet-of-Things, Vol. V, June 2024

Edited by Lech Jóźwiak and Radovan Stojanović

Technical editors:

Prof. dr Budimir Lutovac, University of Montenegro
Jovan Djurkovic, MECOnet

Contributors:

Alberto Marchisio, Andrej Škraba, Anish Bhobe, Christoph Schmittner, Claudio Rubattu, Dominique Blouin, Federico Manca, Francesco Ratto, Hans Vangheluwe, Jovan Djurković, Lech Jóźwiak, Luis Palacios, Morayo Adedjouma, Muhammad Shafique, Nabil Abdennadher, Nikhil Gaikwad, Radovan Stojanović, Rainer Leupers, Rakshit Mittal, Ralf Lübben, Rizwan Parveen, Samir Ouchani, Sokol Kosta, Tarek El-Ghazawi, Zakaria Chihani.

Cite as:

Author/s, "Title of contribution-presentation", in Proceedings of the 5th Summer School on Cyber-Physical Systems and Internet-of Things, Editors: Lech Jozwiak and Radovan Stojanovic, Authors: Alberto Marchisio, Andrej Škraba, Anish Bhobe, Claudio Rubattu, Dominique Blouin, Federico Manca, Francesco Ratto, Hans Vangheluwe, Jovan Đurković, Lech Jóźwiak, Luis Palacios, Morayo Adedjouma, Muhammad Shafique, Nabil Abdennadher, Nikhil Gaikwad, Radovan Stojanović, Rakshit Mittal, Ralf Lübben, Rizwan Parveen, Samir Ouchani, Sokol Kosta and Zakaria Chihani, Vol. V, June 2024, pp. xx-yy, DOI: https://doi.org/10.5281/zenodo.12608092.

Published by Consortium of MECOnet d.o.o. and MANT Association.

For publisher: MECO Events and Publishing, https://pubs.embeddedcomputing.me/

Filipa Lainovica 19, Podgorica, Montenegro

Message from the Chairs,

The 5th Summer School on Cyber-Physical Systems and Internet of Things (SS-CPS&IoT'2024) is the fifth school in a series, organized in Budva, Montenegrin and Mediterranean pearl.

We were pleased to continue the tradition of hosting the SS-CPS&IoT'2024 in a hybrid format, accommodating both online and in-person participation.

SS-CPS&IoT Summer School traditionally set forth general 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.

European stakeholders", so it can be said that it was a Joint School of our community with this significant project.

SS-CPS&IoT'2024 is collocated with CPSIoT'2024, 12th International Conference on Cyber-Physical Systems and Internet-of-Things and MECO'2024, 13th Mediterranean Conference on Embedded Computing.

The Summer School participants were encouraged to submit their papers to CPSIoT'2024 and MECO'2024, and thus gain additional experience of presenting work in one of the TOP conferences in computing.

The CPS&IoT'2024 Summer School Program was composed of four days of lectures, demonstrations, practical hands-on sessions, and discussions, as well as free participation in MECO'2024 and CPSIoT'2024 sessions. The topics of the lectures, demonstrations, and practical hands-on sessions cover major CPS fields and applications. We had about 40 lecturers and students, coming from over 13 countries.

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.

What makes CPS&IoT'2024 Summer School exceptional, in addition to other results, are its Proceedings, which represent indispensable literature in this area. According to official statistics, they are highly quoted and cited.

In addition to their research and educational component, they serves as a supplement to the diplomas awarded to School's participants, after testing their activities and knowledge.

The Chairs of the SS-CPS&IoT'2024 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 5th generation of students who successfully finished School and showed an enviable level of knowledge and interest.

We are very grateful to Jovan Djurkovic, Publication officer of CPSIoT'2024 and MECO'2024 helping us to compose these Proceedings, which represents only part of the results carried out by SS-CPSIoT'2024.

We hope to see you again next year in good health and friendly atmosphere.

Yours,

Lech Jóźwiak
Eindhoven University of Technology, The Netherlands

Radovan Stojanović
University of Montenegro, Montenegro

In Budva, June 2024

Contents

CPS&IoT

SMART4ALL

CPS&IoT'2024 Summer School

Budva, Montenegro
June 11-14, 2024

Introduction

Lech Jóźwiak and Radovan Stojanović

1

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’2024 Summer School

1. Introduction to CPS and IoT
2. Green CPS and IoT
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&IoT2024 Summer School

3

CPS&IoT'2024 Summer School
Budva, Montenegro, June 11-14, 2024

Green Systems for Green World

Lech Jóźwiak

L.Jozwiak@tue.nl

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. Conclusion

2

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

3

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

4

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

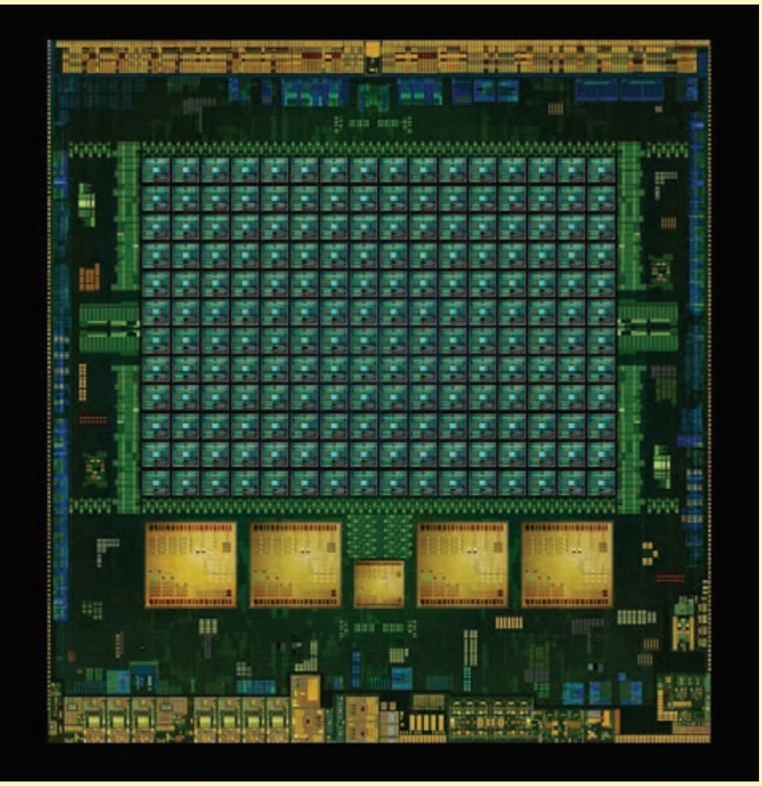
5

Introduction: What are cyber-physical systems?

- 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***
- **Cyber** comes from Greek adjective ***kyberneticos*** (***cybernetic***) that means skilled in steering or governing
- **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
- **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)

6

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

7

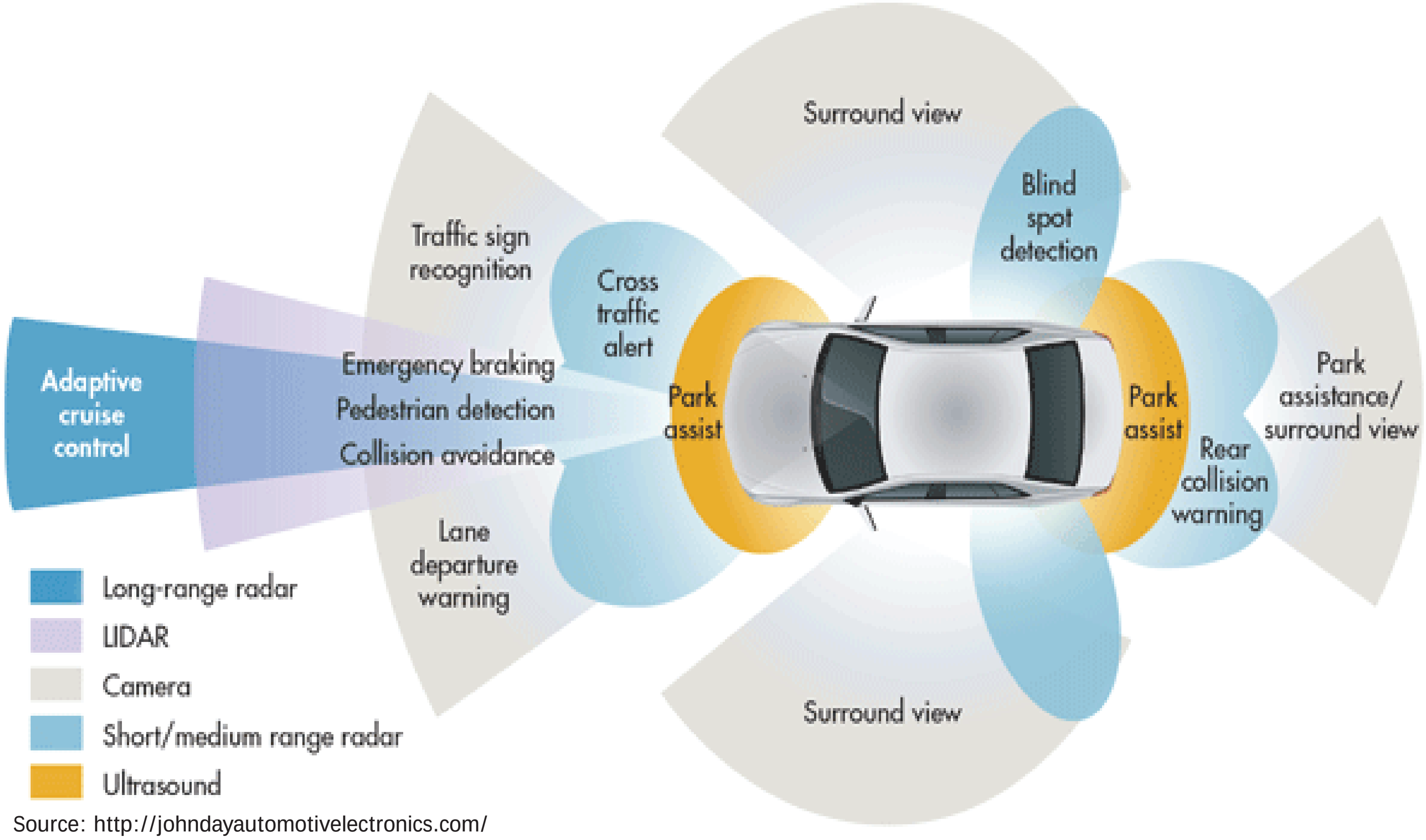
Introduction: cyber-physical technology revolution

- **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

  enabled sophisticated and affordable CPS for numerous new applications (e.g. smart robots, homes, cars, etc.) and 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)**

8

Examples of modern CPS: autonomously-driving cars



Source: http://johndayautomotivelectronics.com/

Examples of modern CPS: smart wearables

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

10

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 modern CPS is very high and rapidly increases**

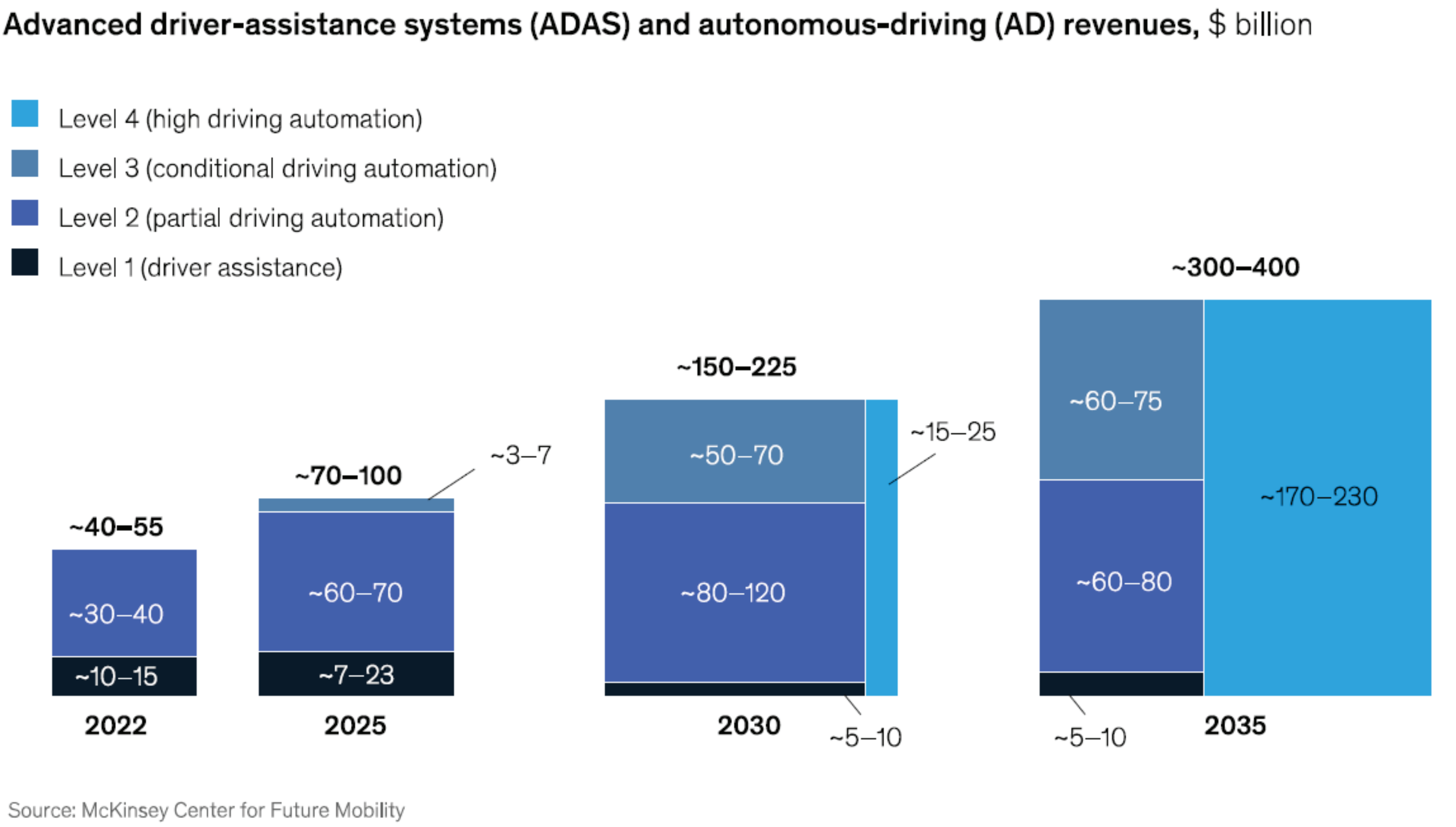
11

Rapid growth of CPS and IoT markets

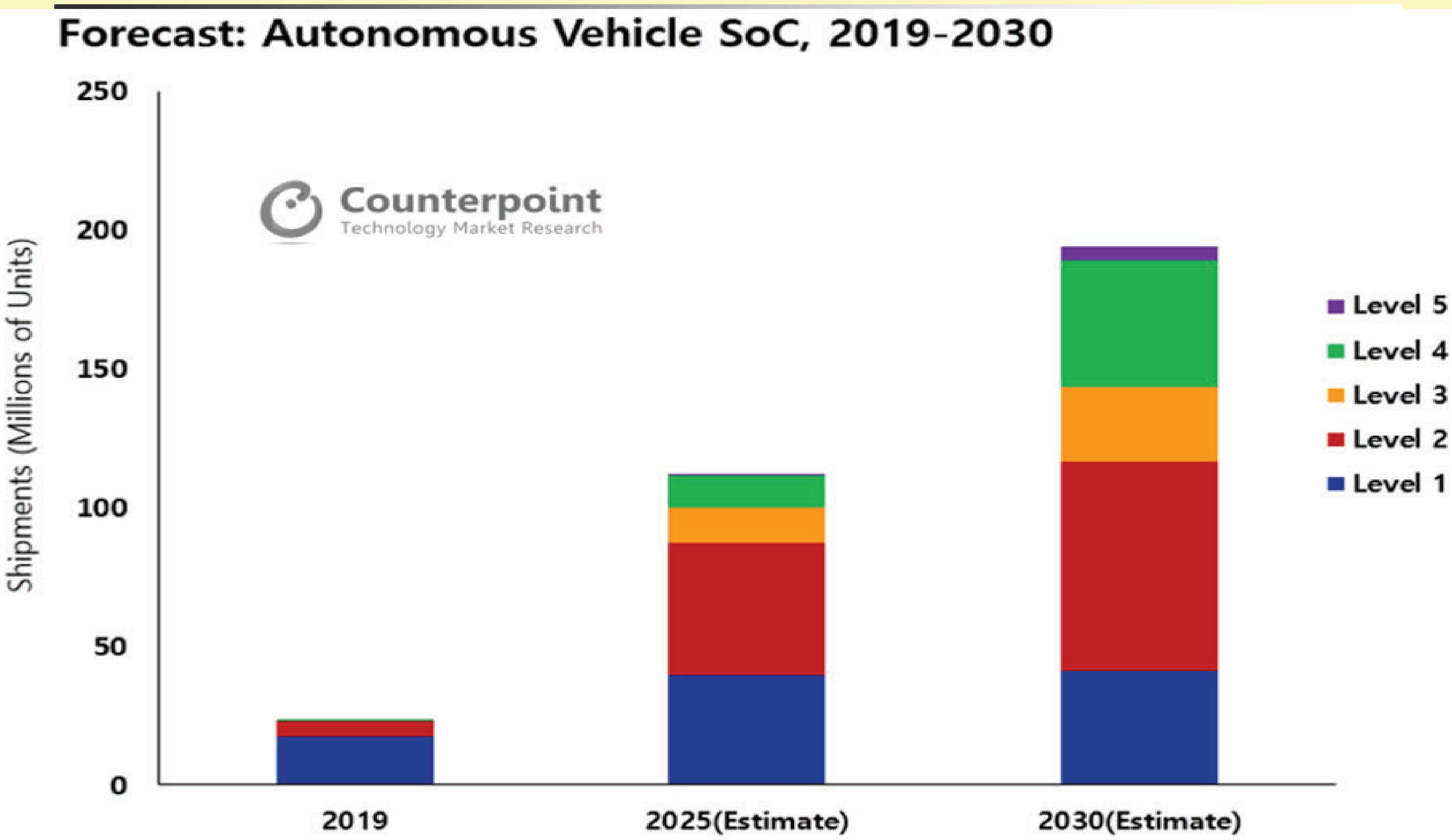
- The number of connected IoT devices was 12.2 billion in 2021 (IoT Analytics)
- IoT Analytics forecasted 14.4 billion connected IoT devices in 2022 and 27 billion connected IoT devices by 2025
- Allied Market Research finds that the global IoT market size was $740.5 billion in 2020, $878.49 billion in 2021 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
- The strongest contributors to the global IoT market are currently industrial manufacturing, healthcare, consumer electronics, automotive and wearables

12

Rapid growth of the modern CPS and IoT markets



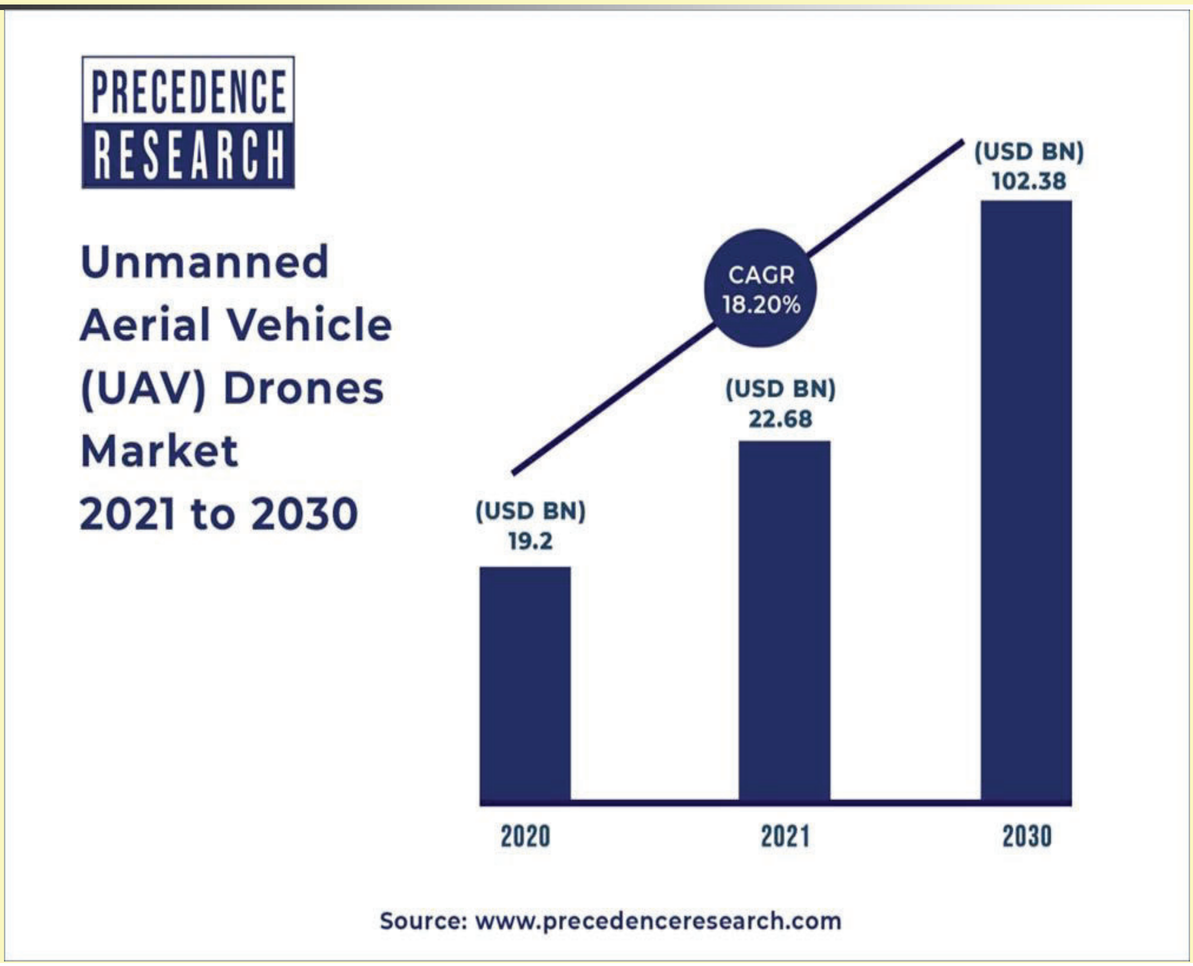
Rapid growth of the modern CPS and IoT markets



Source: Counterpoint, January 2020

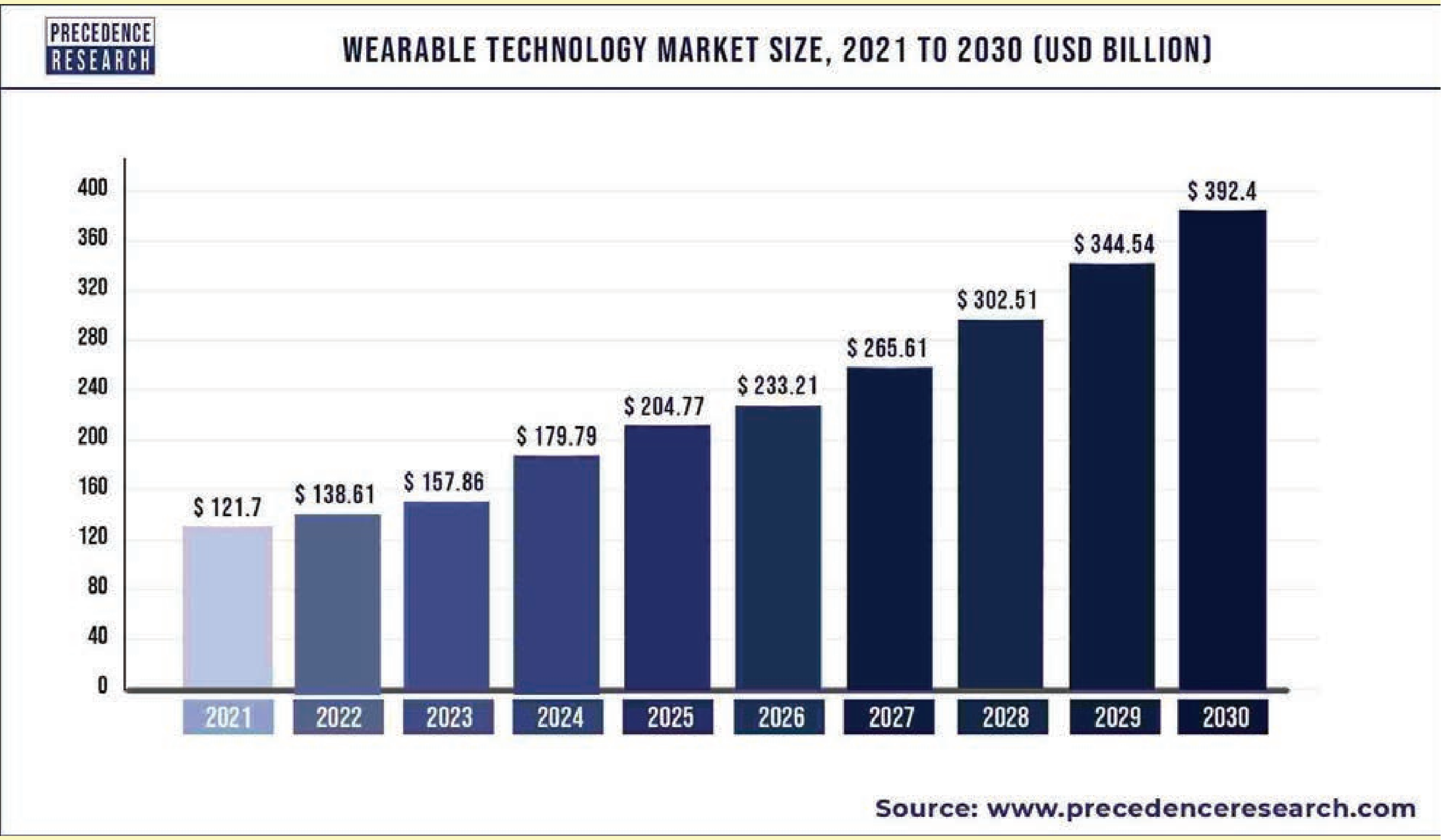
14

Rapid growth of the modern CPS and IoT markets



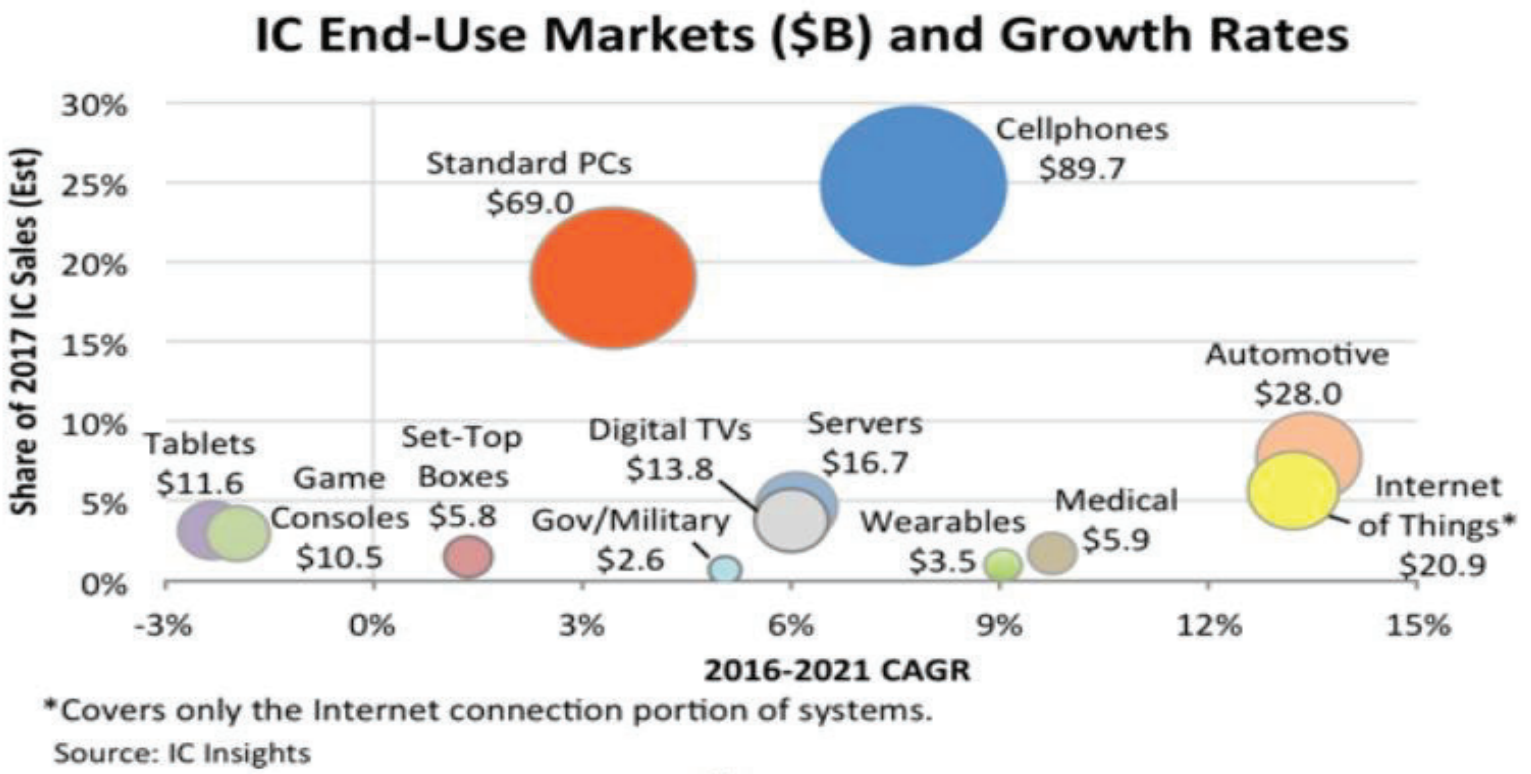
15

Rapid growth of the modern CPS and IoT markets



16

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



Source: IC Insights

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

17

Semiconductor market related to CPS/IoT in 2021/2022

- According to Semiconductor Industry Association (SIA) and World Semiconductor Trade Statistics (WSTS), 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 shipped in 2021
- In 2022, the global semiconductor industry sales achieved a new record value of $574 billion
- The growth was mainly driven by the automotive, industrial and consumer application sectors, while the sale of chips for PC/computer sector substantially decreased (by 5%)
- A further semiconductors sales growth in the CPS/IoT-related automotive, industrial and consumer sectors is expected to continue up to 2030
- For 2023, WSTS, Gartner, and IC Insights expect a small decline of the global semiconductor market in the range of 4% to 5%

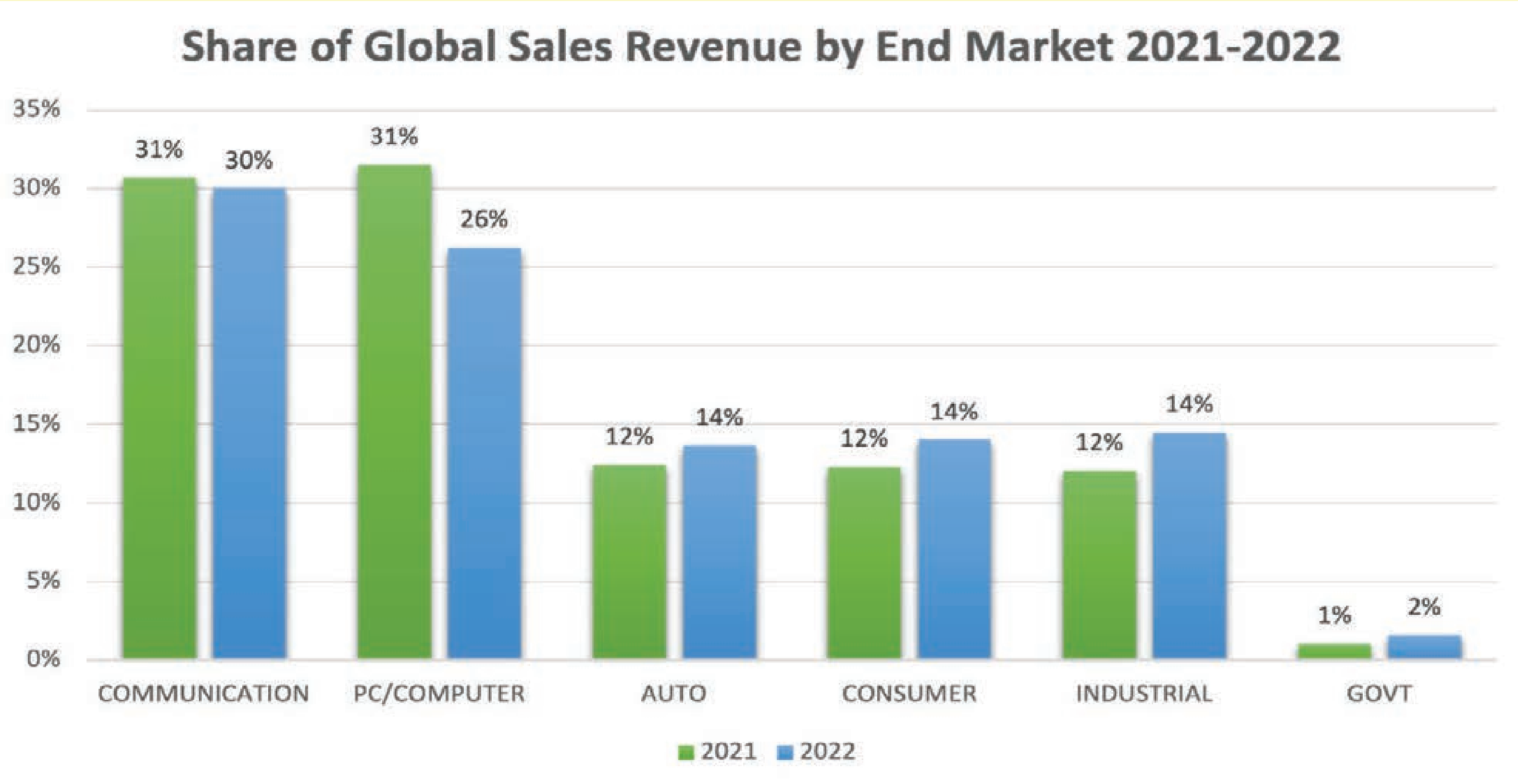
18

Semiconductor market related to CPS/IoT in 2021/2022

- According to Gartner, the market of chips for AI will increase at an annual rate of more than 20 percent to USD 53.4 billion in 2023, USD 67 billion in 2024, and USD 119.4 billion in 2027 (Gartner, August 2023)
- AI is extremely important for CPS and IoT: it provides the intelligent, automated and timely decision-making based on the big amounts of data generated by numerous CPS and IoT devices
- On June 14, 2023, the European Parliament adopted the Artificial Intelligence Act (AI Act) being the first set of rules to manage AI risks and to promote AI uses in line with the EU values
- Recently a special attention of the AI system and hardware developers has been focussed on the generative AI (GenAI or GAI)
- While the traditional AI recognizes existing patterns in data and acts upon them using a certain set of rules, the generative AI creates new the most likely occur patterns of data based on the data on which it was trained
- Generative Ai could be used for ver many purposes, but on the training side it requires to process a huge amount of data (large language models or LLMs)

19

Semiconductor market related to CPS and IoT in 2022

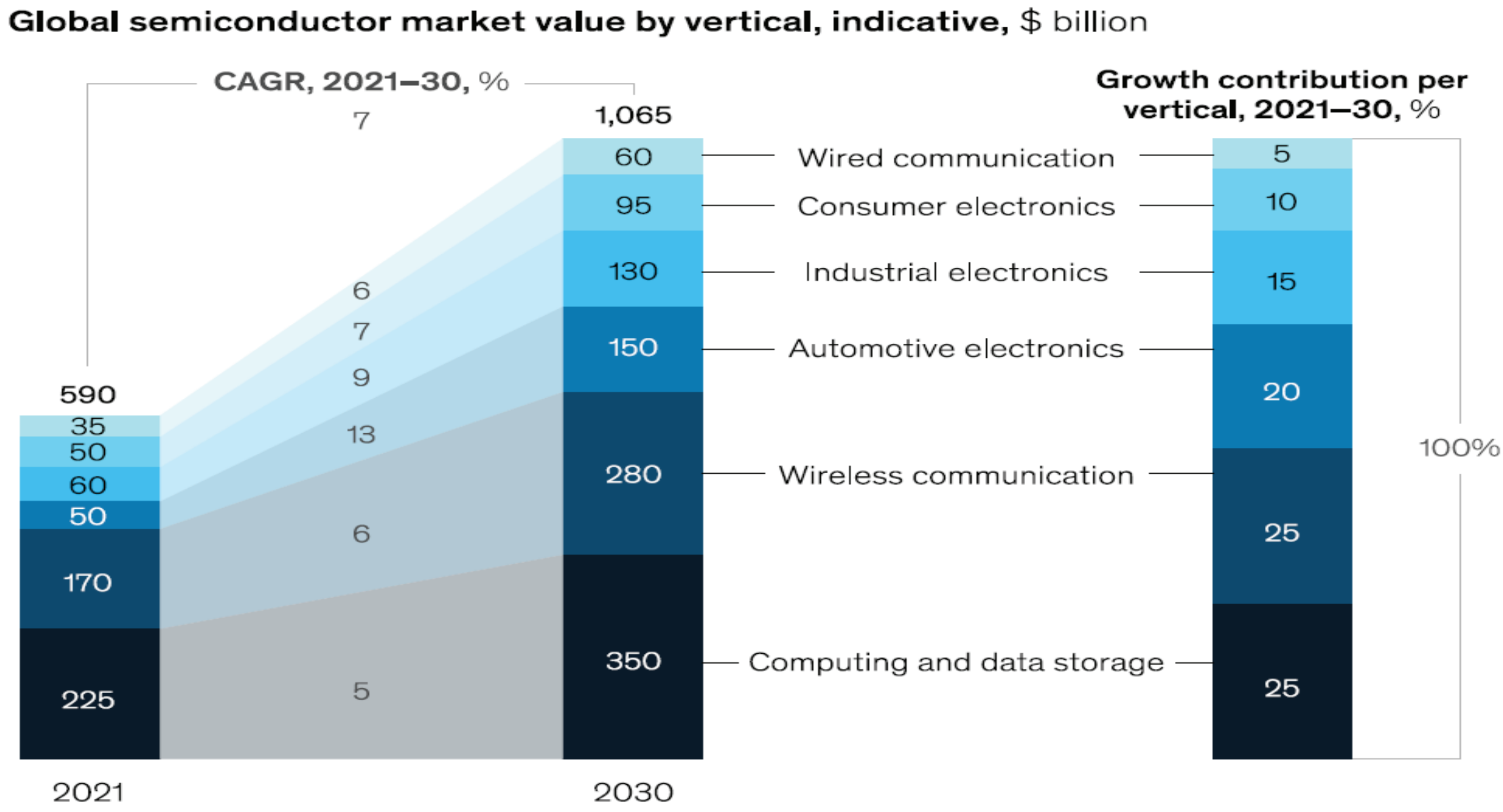


Source: SIA

- PC/COMPUTERs only account for 26%, while a large majority of the rest is related to CPS and IoT

20

Rapid growth of the **semiconductor market** related to CPS and IoT



Source: McKinsey

- A large part of wireless communication, computing and storage, as well as automotive, industrial and consumer are related to CPS and IoT

21

Challenges: unusual complexity and ultra-high demands

- The huge and rapidly developing markets of sophisticated CPS and IoT 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**

22

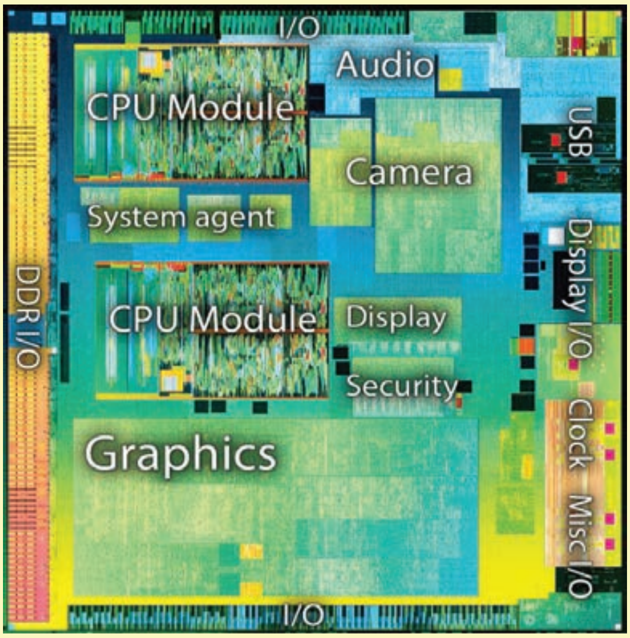
Challenges: application parallelism and heterogeneity

- 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

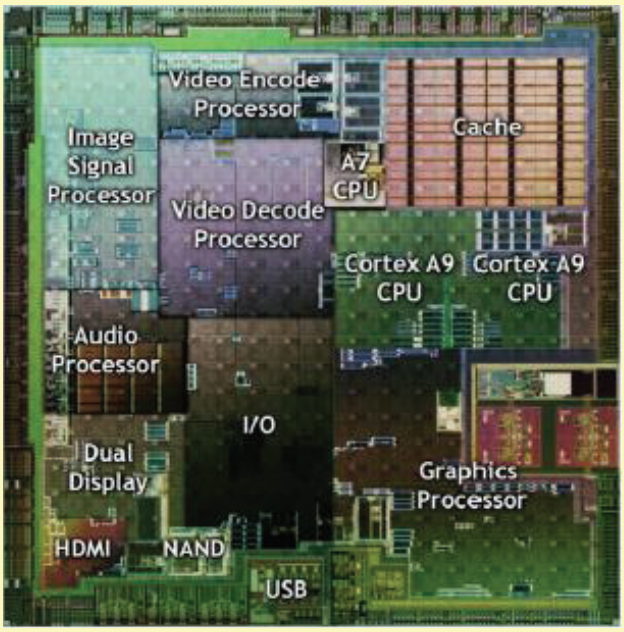
23

Challenges: application complexity, parallelism and heterogeneity

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



Intel Atom Z3770*



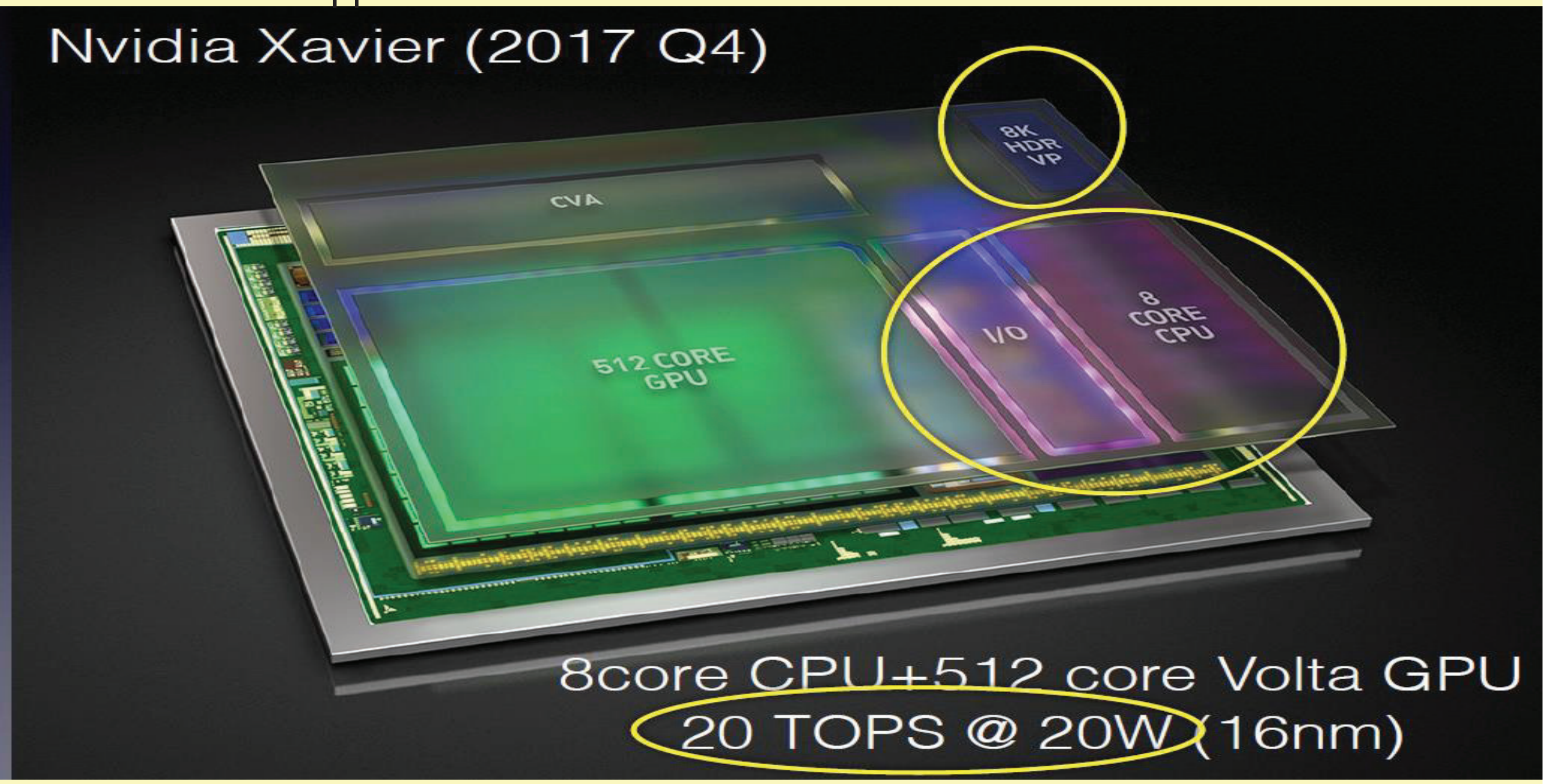
Nvidia Tegra 2⁺

*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

24

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

25

Quality-driven Model-based Design

- The rapidly growing system complexity and demands of (ultra-)high performance and/or (ultra-)low energy consumption, while requiring a high reliability, safety and security, created a new difficult situation that cannot be well addressed without an adequate design methodology and design automation
- When considering a **system and design methodology adaptation**, we have first to ask: *what general system approach and design approach seem to be adequate to solve the problems and overcome the challenges*?
- **Predicting the current situation,** more than 20 years ago I proposed such **system paradigm** and **design paradigm**:
  - 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)

26

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

- When using the quality-driven design methodology to develop high-quality collaborating cyber-physical systems, in which the sophisticated cyber systems 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***

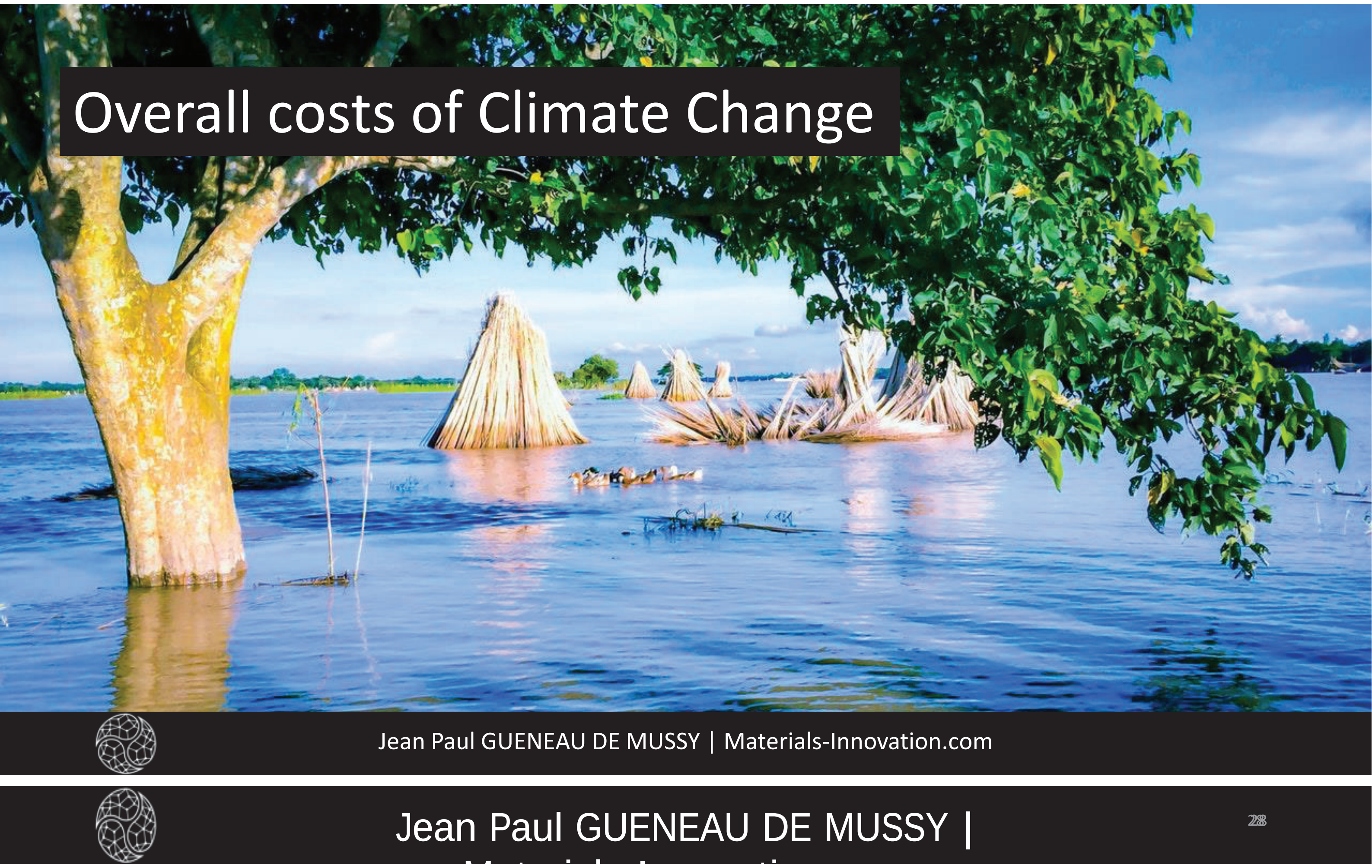
- **We also have no chance to not do this**

- ***Our social, physical and life systems have to be significantly and immediately improved***

- **Why?**

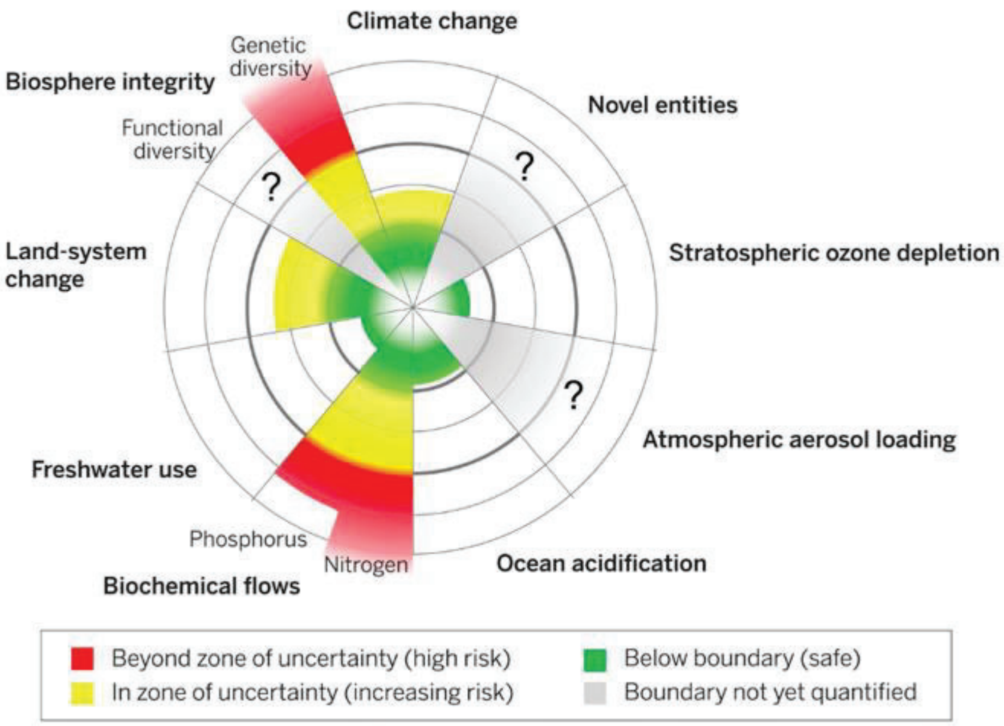
- Please watch the following few slides that I got from my friend Dr. Jean Paul Gueneau de Mussy, Sustainability and Innovation Expert, CEO of Materials and Systems Innovation Company, https://materials-innovation.com/

27


Overall costs of Climate Change
Jean Paul GUENEAU DE MUSSY | Materials-Innovation.com
Jean Paul GUENEAU DE MUSSY |
28


Biodiversity loss
Massive use of Resources
Jean Paul GUENEAU DE MUSSY | Materials-Innovation.com
Jean Paul GUENEAU DE MUSSY |
29

Planetary Boundaries



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

Jean Paul GUENEAU DE MUSSY |

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.

31

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

32

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.

33

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.

34

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 !!!**

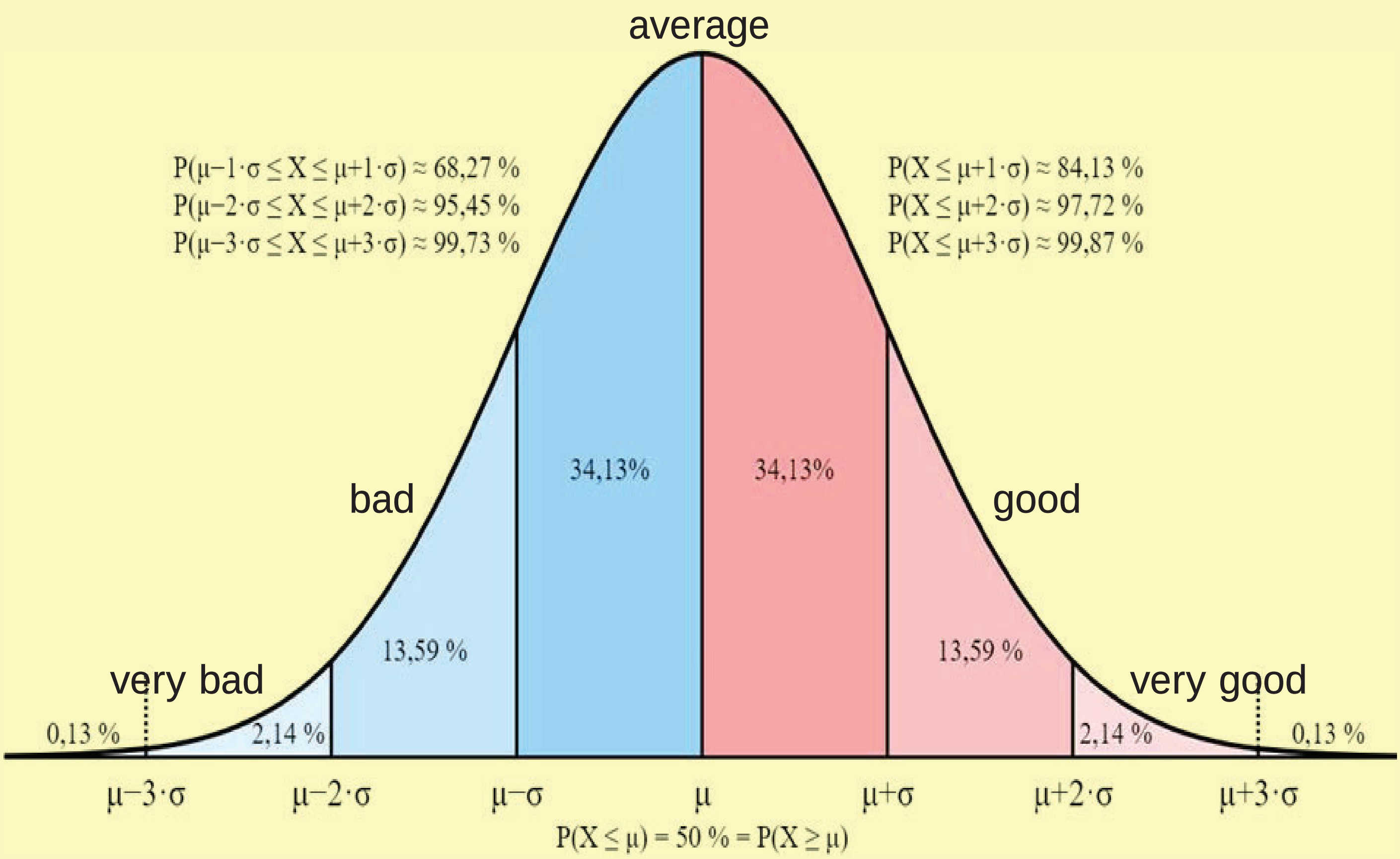
35

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

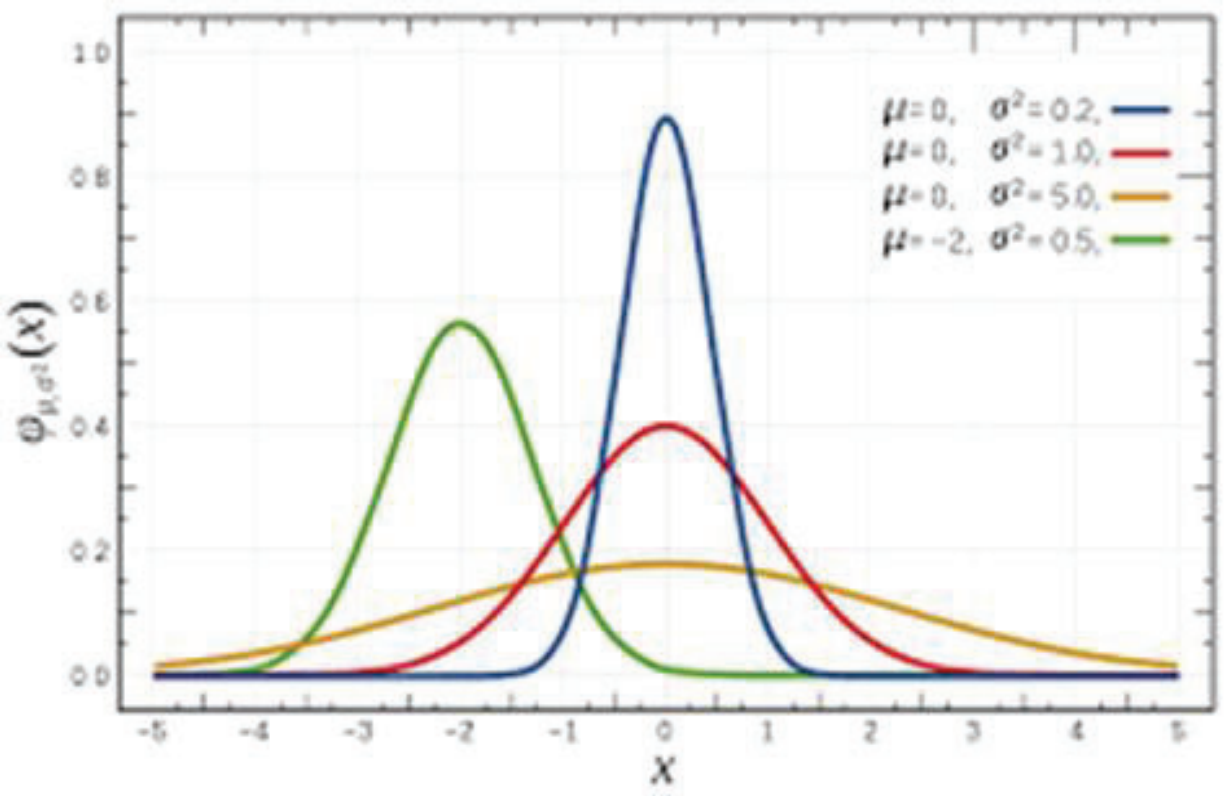
36

Broader context of the destruction



37

Broader context of the destruction



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

Source: Wikipedia

38

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

39

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 fight 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**?

40

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

41

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, energy and information flows in CPS and IoT***
- ***What is circular regenerative economy?***

42

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

43

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
- 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
- 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***

44

What can and should be the role of the modern CPS and IoT technologies in recovery from this disaster?

- The main role of the CPS&IoT technologies can and should be:
  - high increase of the effectiveness and efficiency of the energy and materials consumption in all kinds of systems, collaboration chains and related flows, and
  - big decrease of waste related to the systems, chains and flows
- With their smart sensing, networking, processing and actuation solutions, modern CPS&IoT technologies make it possible to effectively and efficiently collect, transmit, process and use information for (remote) system monitoring, collaboration and control
- Through enabling an effective and efficient energy and materials management in different distributed collaborating systems, and through exploitation of smart grid, smart mobility, smart city, smart home and other smart system concepts, CPS&IoT can very much contribute to achievement of the energy efficiency goals with renewable energy sources and energy harvesting, as well as to reduction of materials consumption and waste

45

We have to recover from this disaster ASAP

- 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
- 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

46

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

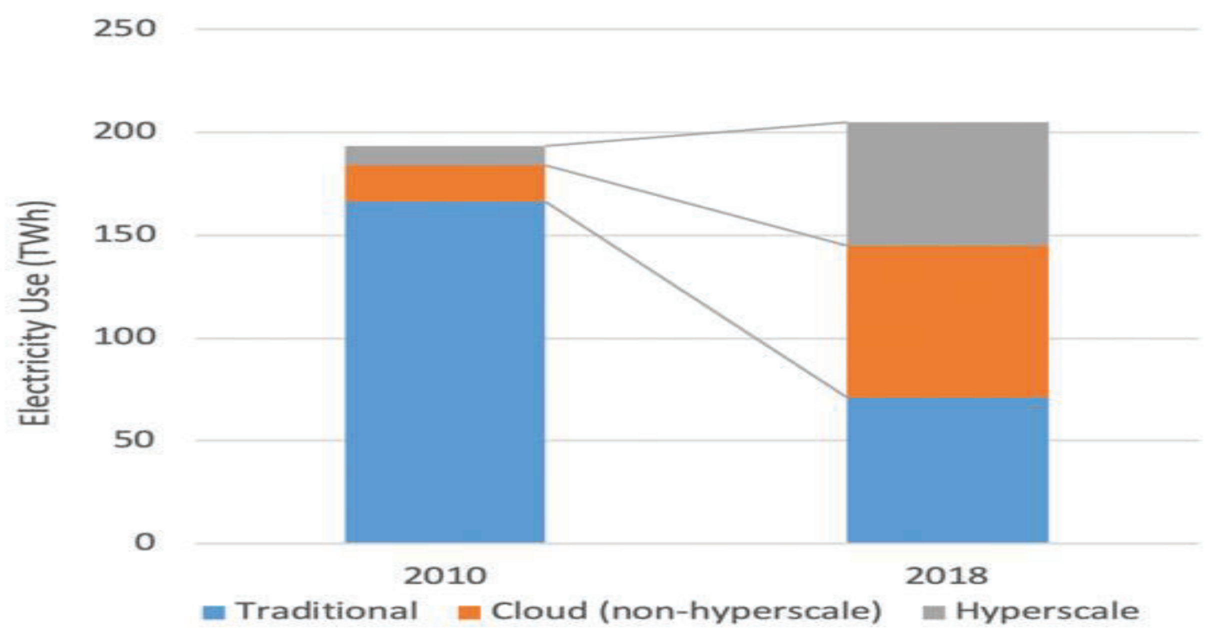
47

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

48

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

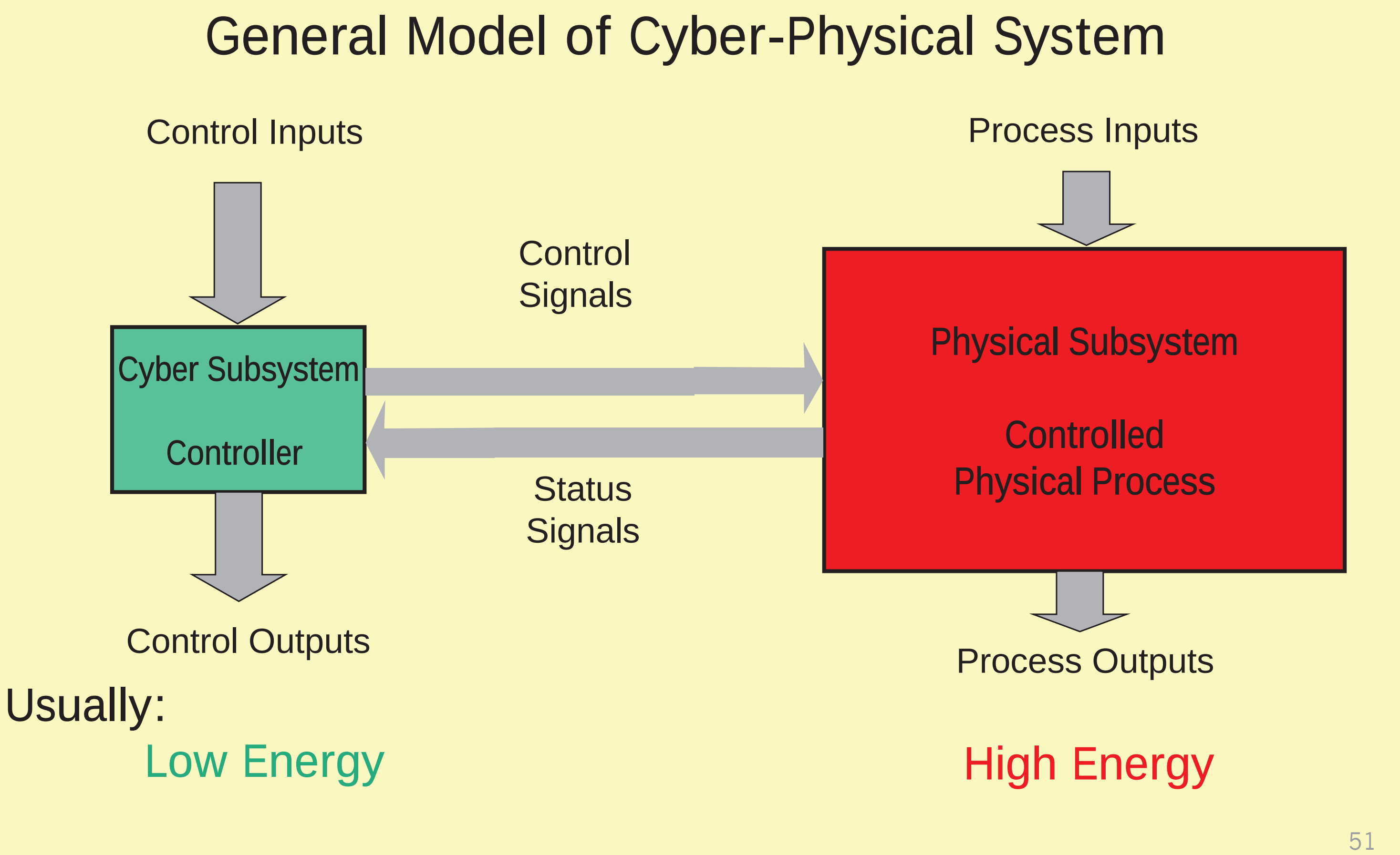
49

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

50

Environmental footprint of cyber-physical systems



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

52

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

53

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**

54

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

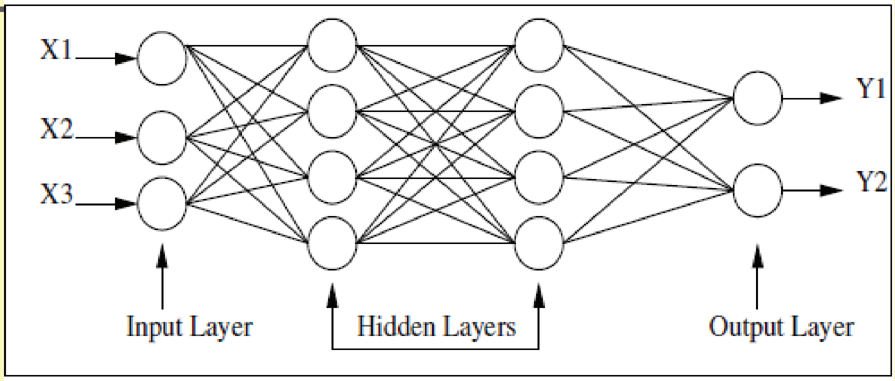
55

Edge AI and Edge ML

- A vast majority of ML methods/algorithms use various models, such as: artificial neural networks, support-vector machines, decision trees, belief networks, etc.
- Based on the training data machine learning methods/algorithms build/train such model which is then used to process additional data to make decisions or predictions
- 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
- Machine learning system is an organized system that implements one or more machine learning methods/algorithms
- 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 motion prediction, robot or vehicle path planning, etc.
- 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

56

Edge ML and and Deep Learning (DL)



- ANN is a ML model involving nodes called neurons which are connected with edges
- 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
- ANN with several hidden layers is called a deep ANN (DNN)
- The spectacular progress in the massively parallel computing platforms in the recent 10 years enabled the implementation of much more complex neural networks and the reincarnation of neural networks and related fields in the form of deep learning

57

New Edge Computing Platforms for ML and AI

- The interest in Machine 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
- 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
- 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.)

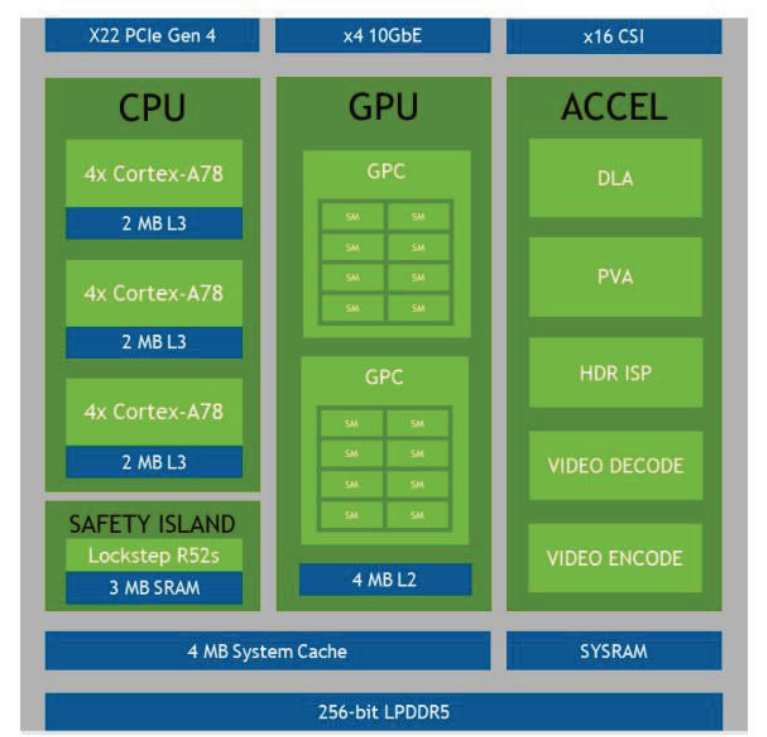
58

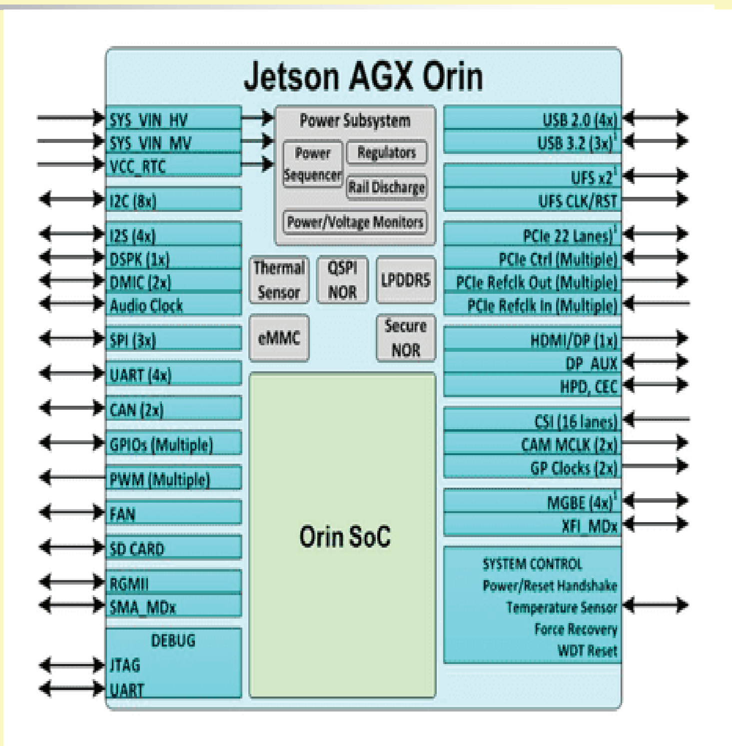
New Edge Computing Platforms for ML and AI

- 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

59

New Edge Computing Platforms for ML and AI





Orin SoC Block Diagram

Jetson AGX Orin System-on-Module

Source: NVIDIA

60

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

61

New Edge Computing Platforms for ML and AI

Source: Mobileye, an Intel Company

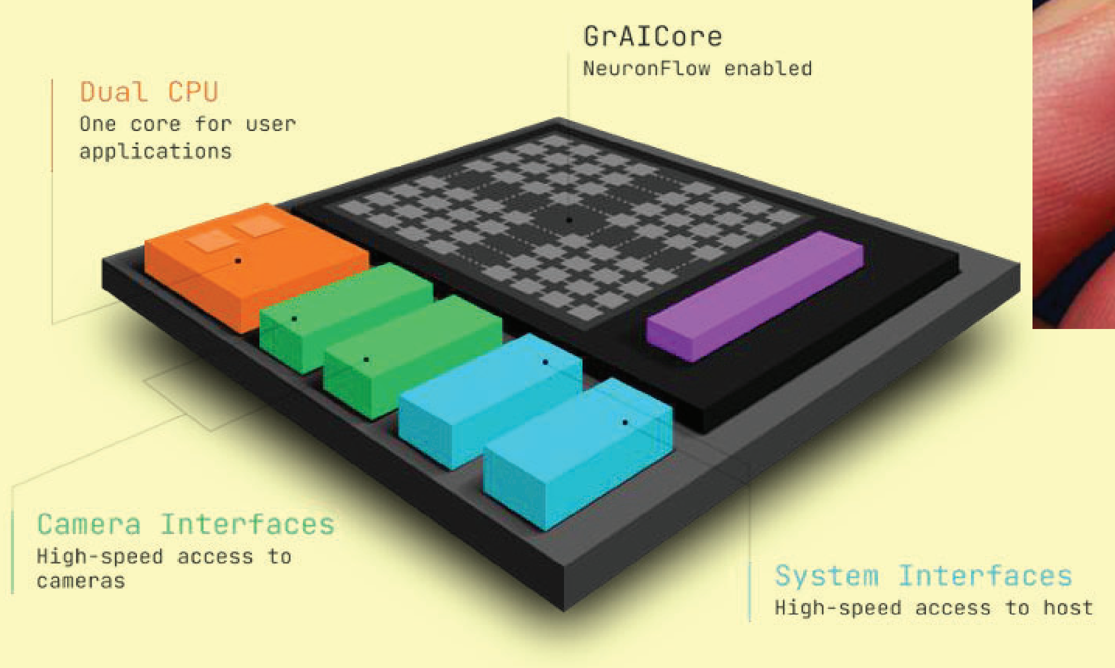
62

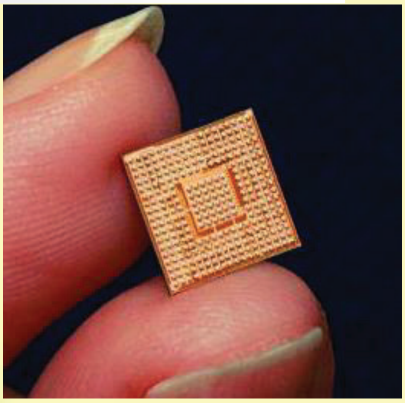
New Edge Computing Platforms for ML and AI

- 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 two 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 (few ms for ResNet-50) and very low-power (< 0,5 W for ResNet-50)

63

New Edge Computing Platforms for ML and AI





GrAI VIP SoC

Source: GrAI Matter Labs

64

New Edge Computing Platforms for ML and AI

AMD Versal AI Edge Series Gen 2 adaptive SoC

Source: AMD

65

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'', livestock tracking in agriculture, etc.

66

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

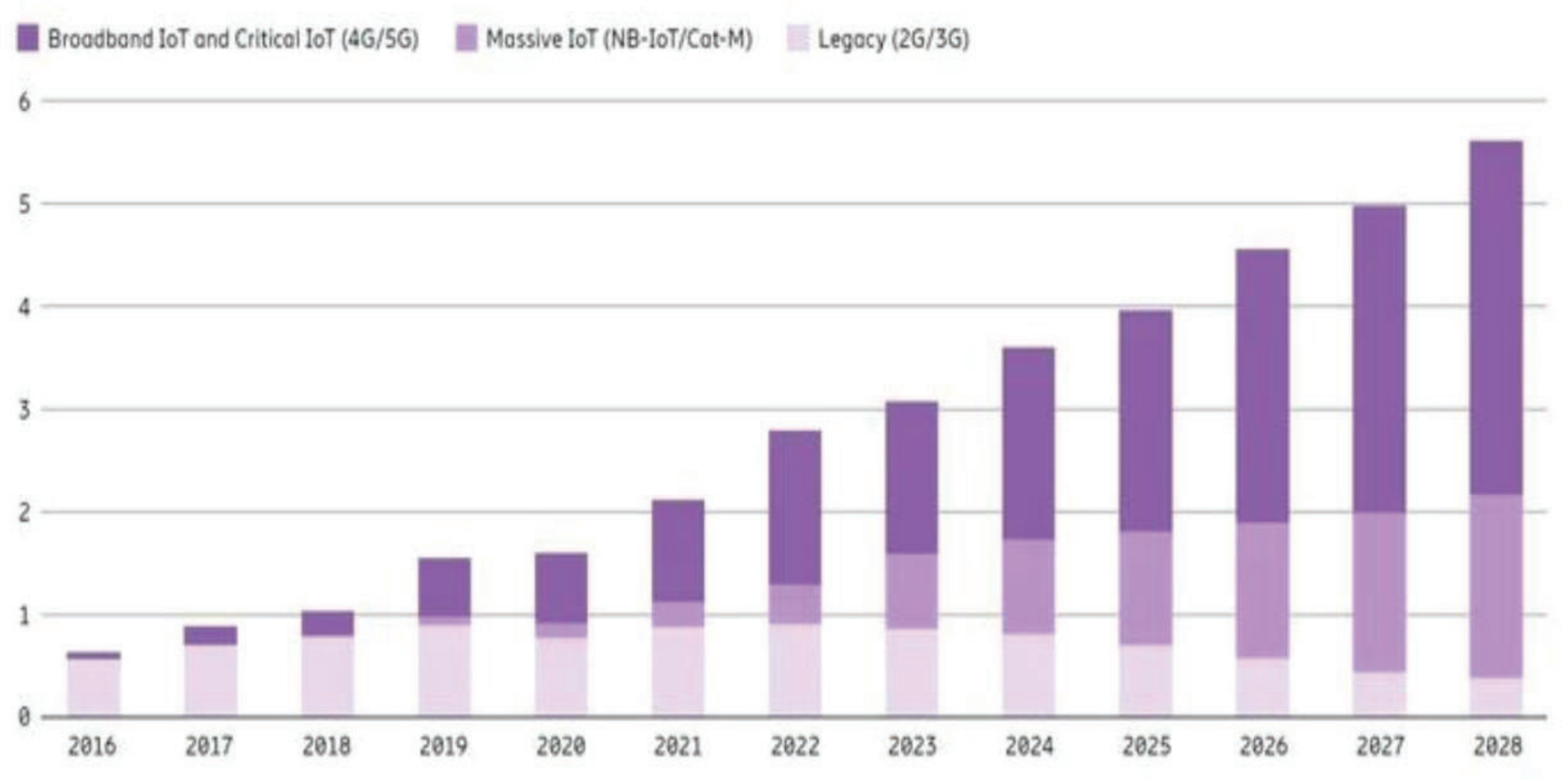
67

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 solutions, 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

68

Number of IoT connections (in billion)

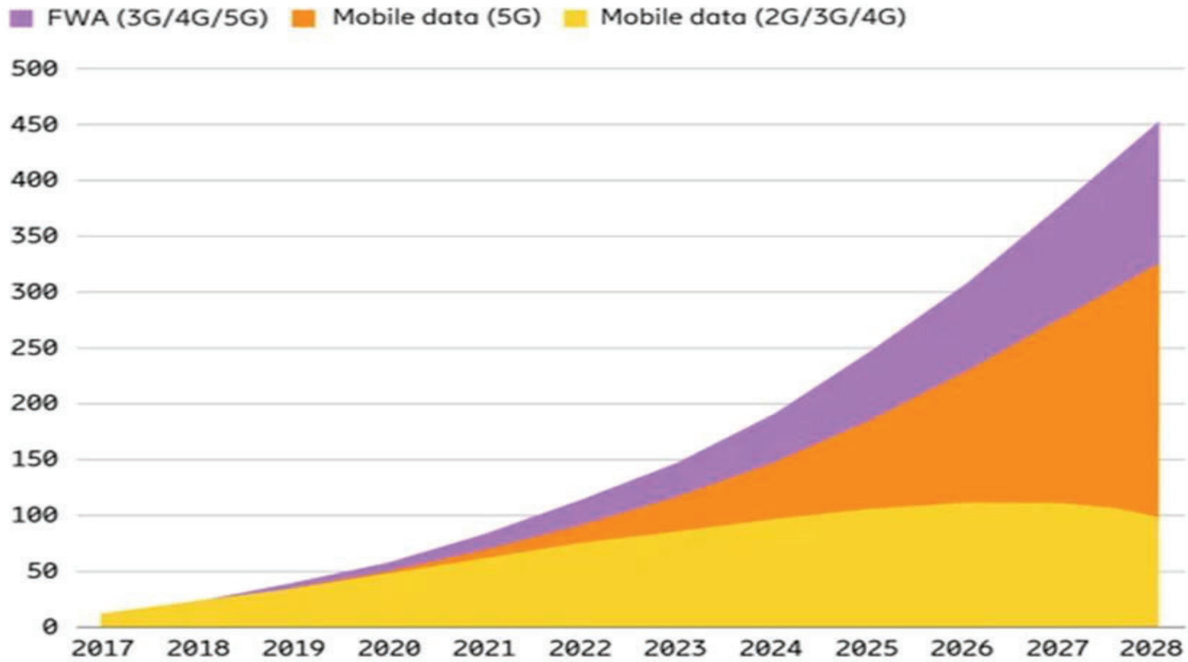


Source: Ericsson Mobility Report, November 2022

- Broadband and Massive IoT will co-exist
- Broadband IoT (4G/5G) connections (including critical) will dominate

69

Mobile network data traffic (in EB per month)



Source: Ericsson Mobility Report, November 2022

- 5G to drive the mobile data growth and 4G/3G/2G data traffic will decline by 2028
- Fixed Wireless Access (FWA) will increase
- Ericsson forecasts that between the end of 2023 and 2029 the global 5G subscriptions will grow by more than 330 percent: from 1.6 billion to 5.3 billion (Ericsson Mobility Report, November 2023)

70

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

71

CPS&IoT'2024 Summer School
Budva, Montenegro, June 11-14, 2024

Introduction to Quality-Driven Design of Cyber-Physical Systems

Lech Jóźwiak
Department of Electronic Systems
Faculty of Electrical Engineering
Eindhoven University of Technology
L.Jozwiak@tue.nl

1

Outline

1. Challenges and demands of modern CPS
2. Quality-driven Model-based Design Approach
3. What is quality?
4. Quality-driven Design: difficulties and design models
5. Main concepts of the quality-driven model-based design
6. Quality-driven design space exploration
7. Example: Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS
8. Conclusion

2

Challenges: unusual complexity and ultra-high demands

- The huge and rapidly developing markets of sophisticated CPS and IoT 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**

3

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 **design automation**

4

Quality-driven Model-based Design Approach

- 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
- This **research confirmed the adequacy of the quality-driven model-based design methodology**

5

Quality-driven Model-based Design Approach

- ***What is the quality-driven design?***

- **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?

6

What is quality?

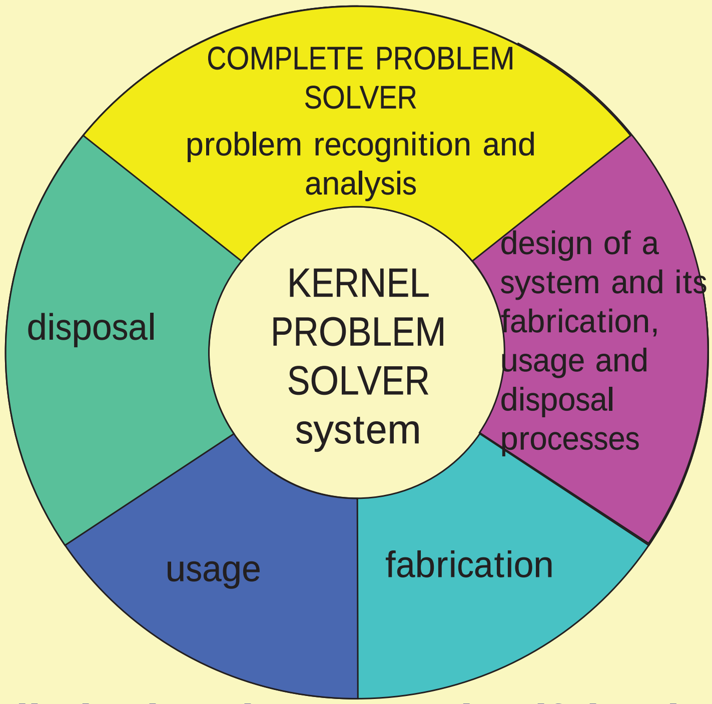
- When I started my work in quality-driven design, I analysed very many definitions of quality and concluded that for many reasons no one of the existing definitions could be used for quality-driven design

- **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)*

7

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

8

Problems with the existing definitions of quality

- None of these definitions was precise enough to enable the systematic consideration, measurement and comparison of quality
- Their assumption of perfectly known and inviolable customer's requirements was 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

9

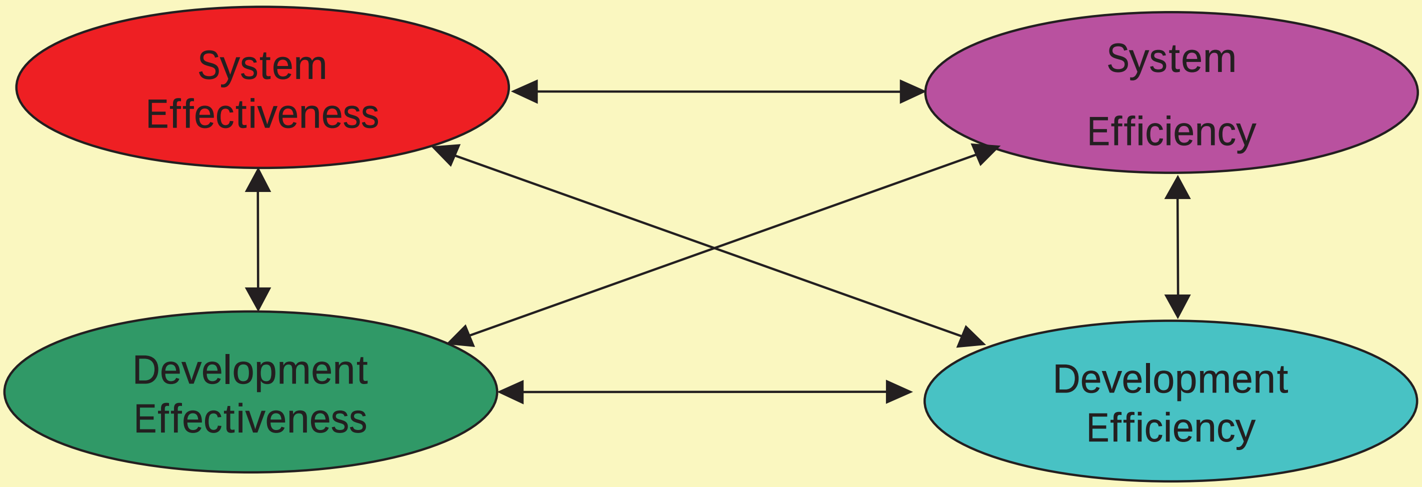
New quality definition proposed by me 20 years ago

***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

10

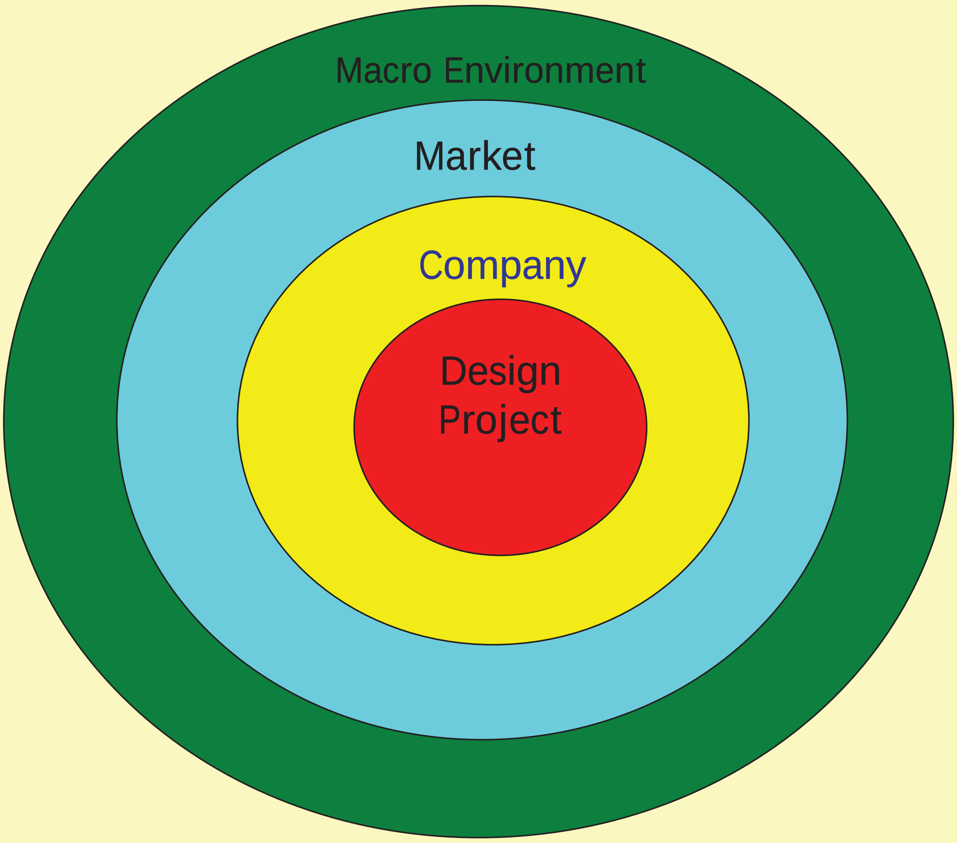
Quality-driven Design - Difficulties



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

11

Quality-driven Design - Difficulties



Interactions of a design project with its context

12

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

13

Quality-driven Design - Design models

Due to all the difficulties

⇓

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

- *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.

14

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

15

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

$$\Downarrow$$

- ***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

16

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.

17

Quality-driven Design: Limiting the design subjectivity

- **One of the main aims** of using the well-defined 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***

18

Quality-driven Design: Limiting the design subjectivity

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

19

Quality-driven Design: Government regulations and standards

- ***Adequate government and industry 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

20

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***

21

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

22

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

23

Quality-driven DSE

- During the design process the structural information is gradually added by the designers and synthesis tools to the created (partial) solutions.
- This evolutionary quality engineering processes applies the ***problem-solving framework of heuristic search*** and ***decomposes the total design problem into several issues***.
- In this framework, the process of design problem solving can be represented by a ***design search tree***:
  - the tree's **nodes** correspond to various ***design issues*** (sub-problems)
  - the tree's **branches** correspond to various ***design options*** (alternative solutions)
  - for each issue, many various alternative solutions are typically possible.
- A **design decision** is a choice of a particular option, or the option chosen
- Each option chosen may recursively raise new issues, expanding the design search tree downwards until a final design will be obtained

24

Quality-driven design space exploration

- For each issue, many various alternative solutions are typically possible.
- For each issue, we can construct some **issue's quality models**, composed of some selected and abstracted functional, structural and parametric requirements extracted in an appropriate manner from the total quality model of the considered system.
- In particular, the issue's **decision model** can be constructed that is a base for decision making in the scope of a certain issue
- A **decision model** is a ***partial*** (reduced to only certain concerns) and ***abstract*** (reduced to the necessary and/or possible precision level) **model** of the required quality, ***expressed in the decision-theoretical terms***.
- ***Decision models*** and ***design parameter estimators*** enable application of the ***multi-objective decision methods*** for construction, improvement and selection of the most promising solutions.

25

Quality-driven Design - Decision models

- The decision model of a given issue **must account for all system characteristics substantially relevant** to the issue
- It **must specify preferences of values for all the characteristics**, expressed by hard constraints, objectives, and trade-off information
- For each single characteristic, the preferences of its values can be characterized by specifying a utility (effectiveness or efficiency) function $u_i(x_i)$ for the characteristic x_i
- Each utility function $u_i(x_i)$ describes the level of satisfaction from a particular value of the characteristic x_i
- Due to the ***multi-aspect nature of systems*** and possible ***trade-offs***, the **relative importance of different characteristics** or the **reference points in the utility space have to be specified**

26

Quality-driven Design - Decision models

- This can be done in **different ways** dependent on the problem characteristics, for example by:
  - establishing an order for the objectives,
  - constructing a multi-objective utility function,
  - defining ranking information,
  - establishing local preferences for small changes in values of the objectives, or
  - defining some **reference (aspiration) points in the utility or parameter space**
- With such models the **total system quality Q** can be modelled as a **function of utility levels of all the important system characteristics** influencing the systems effectiveness or efficiency
- **Such design decision models make it possible to apply the multi-objective decision methods for invention and selection of solutions that are "totally optimal"**

27

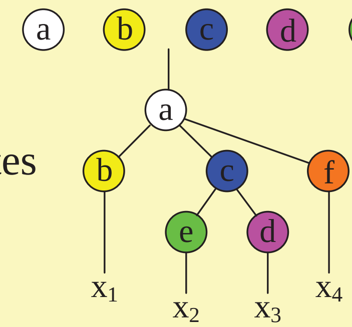
Modeling quality Q as a (vector) function of utility levels of the system characteristics

$$Q(y)=Q(x_1(y), x_2(y),..., x_n(y))=F(v_1(x_1), v_2(x_2),..., v_n(x_n))$$

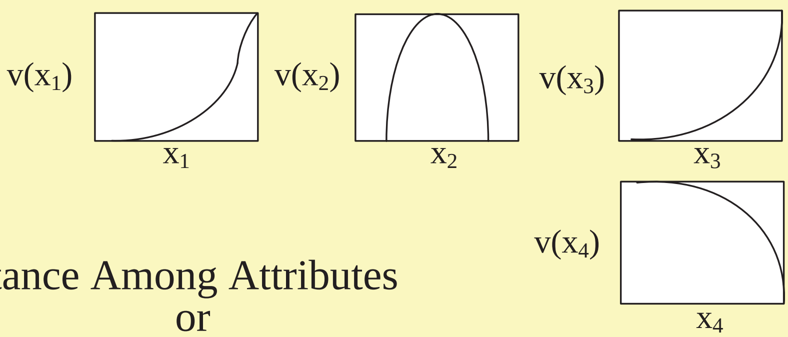
Attributes

Hierarchy of Atributes

Physical Measures



Utility Functions



Tradeoffs:

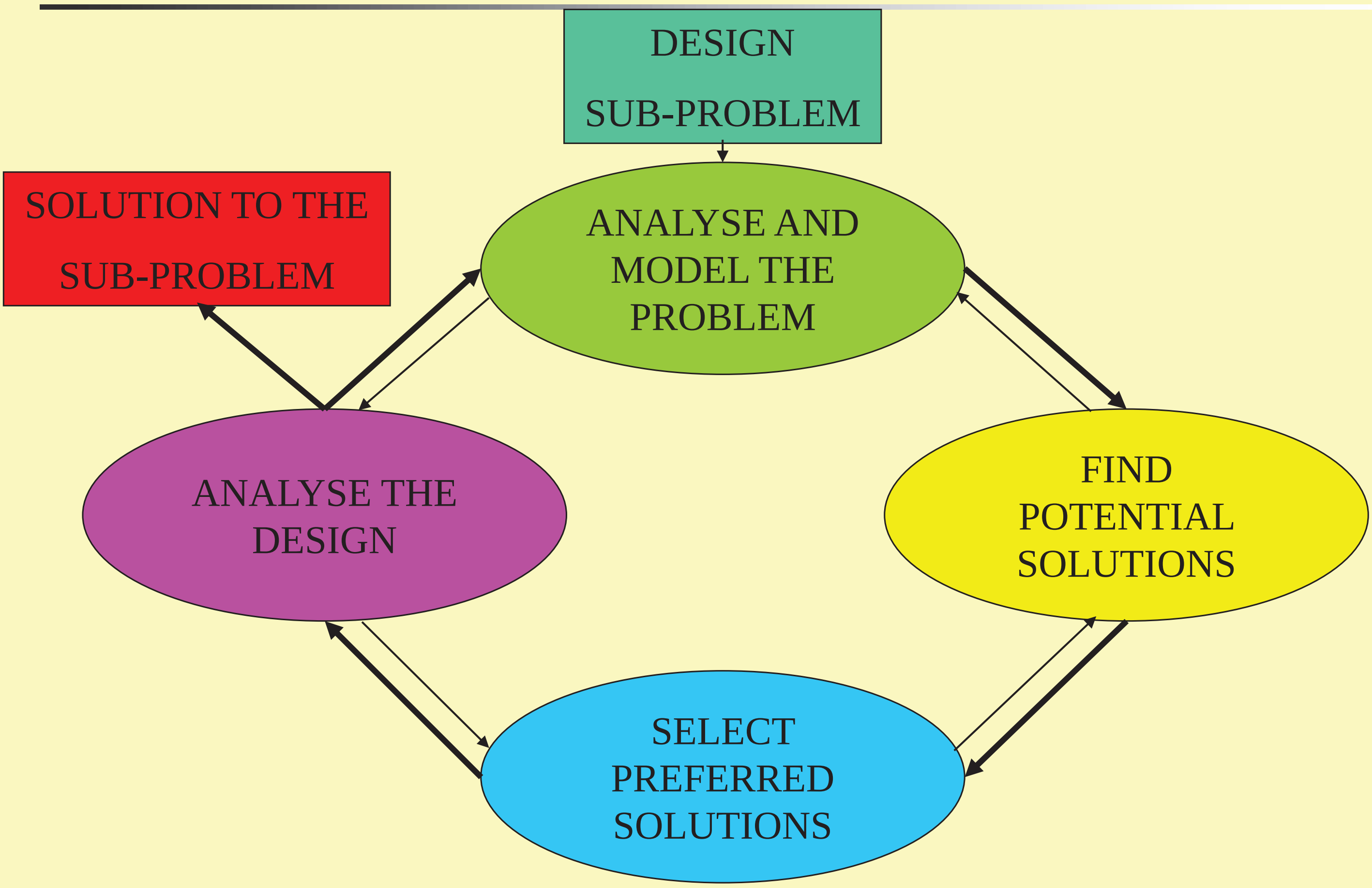
- Relative Importance Among Attributes

or

- Reference Points in Utility Space (or Parameter Space)

28

Generic model of the quality-driven design space exploration

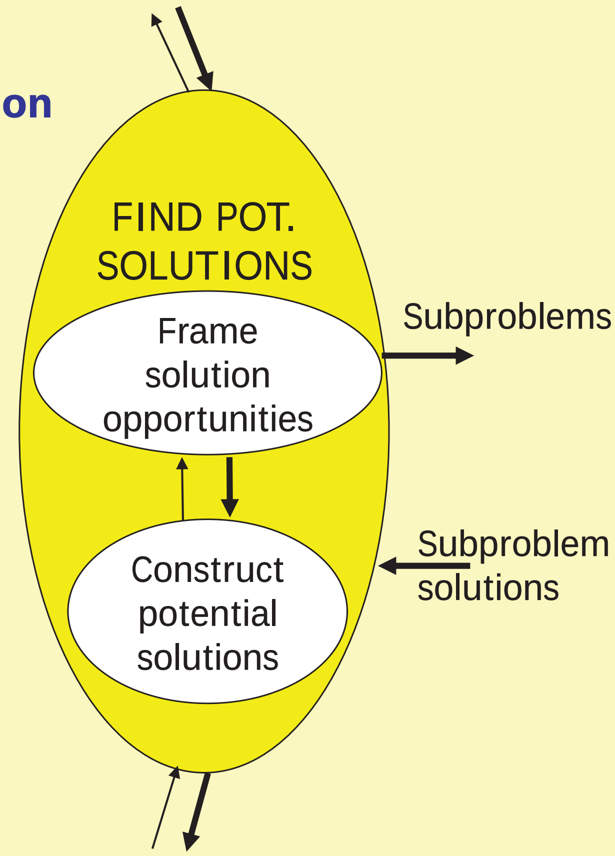


29

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



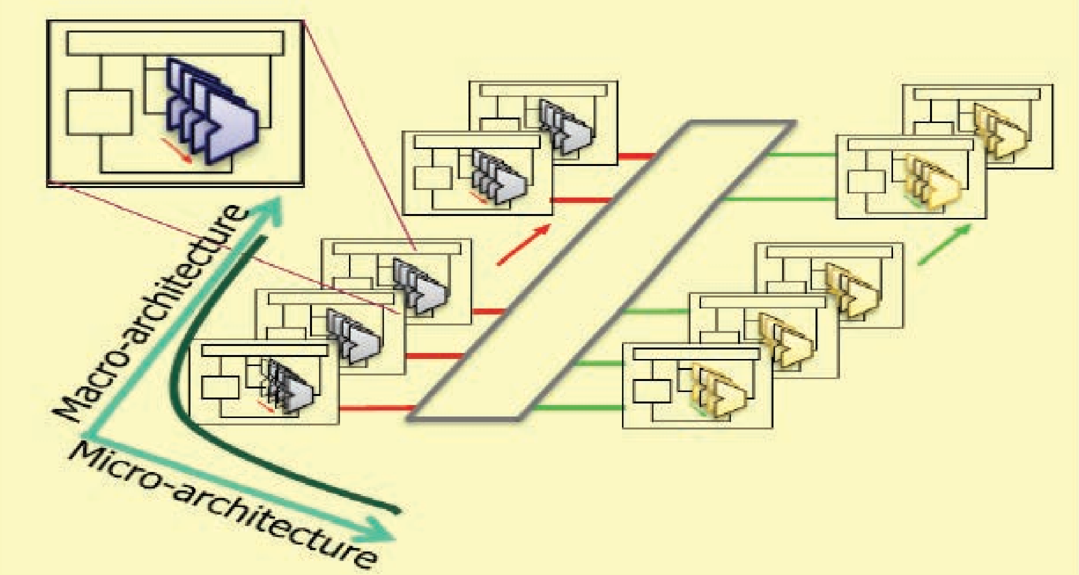
30

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***

31

Example: Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS: *palallel multi-processor technology*



- Complex massively-parallel multi-processors are necessary, with micro-architectures of elementary processors spanning the full spectrum from serial, through partially-parallel, to fully parallel.
- A very high number of possible macro-architecture/micro-architecture combinations and related computation mappings
- A huge design space of various possible multi-processor architectures with different characteristics.

32

Example: Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS: *issues and callanges*

- The application’s parallelism has to be exploited at two architecture levels: system macro-architecture and processor micro-architecture level.
- Similar performances can be achieved with:
  - less processors, each being more parallel and better targeted to a particular part of a complex application,
  - more processors, each being less parallel or less application-specific.
- Each of the alternatives can have different physical and economic characteristics, such as power consumption or circuit area.
- This results in the necessity to explore and decide the various possible tradeoffs between the micro-architecture and macro-architecture design.
- Each micro-/macro- architecture combination requires different compatible memory and communication architectures.
- Exploitation of data parallelism in a computing unit micro-architecture usually demands getting the data in parallel for processing.
- This requires simultaneous access to parallel memories and simultaneous data transmission.

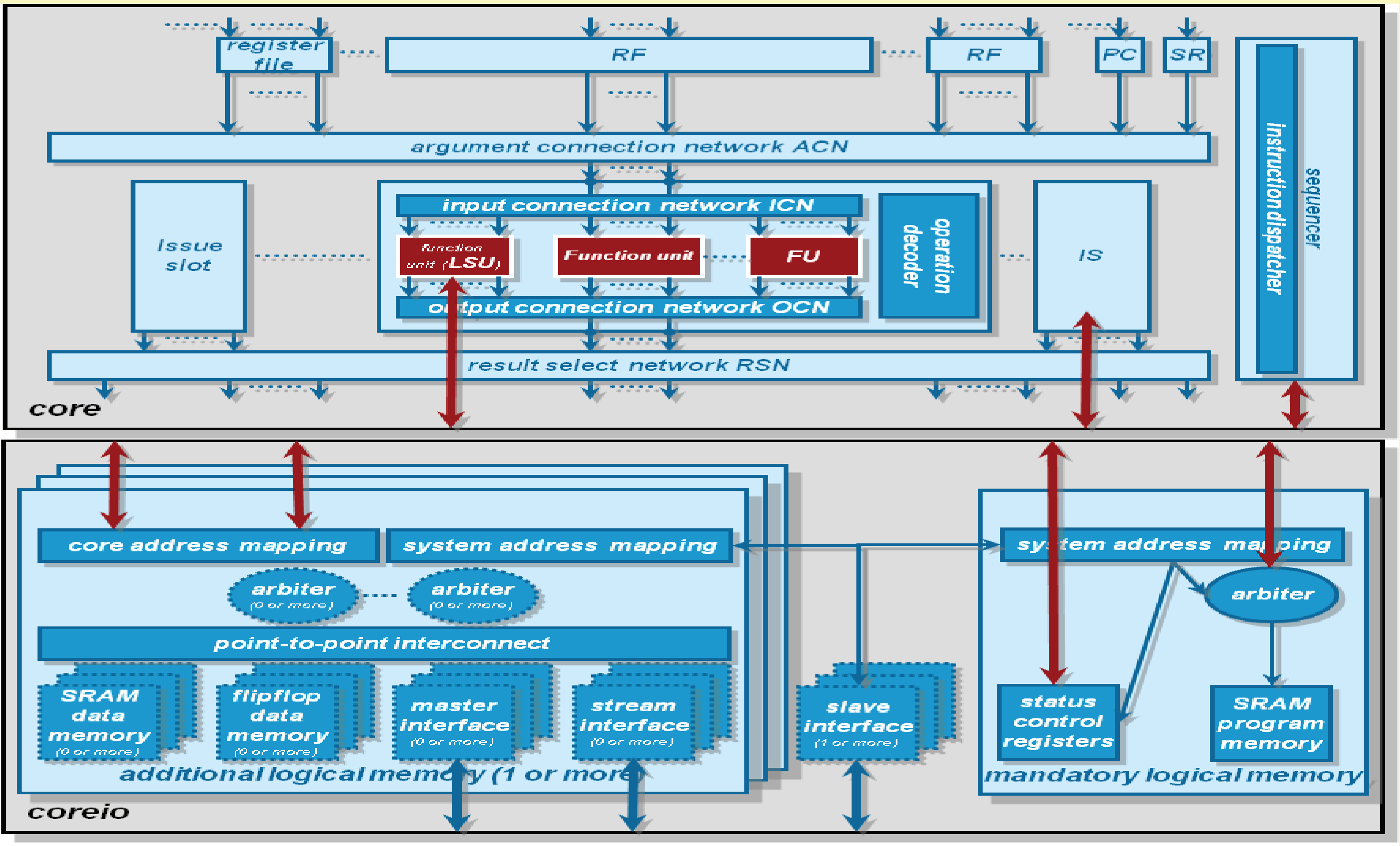
33

Example: *Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS:* multi-ASIP case (ASAM project)

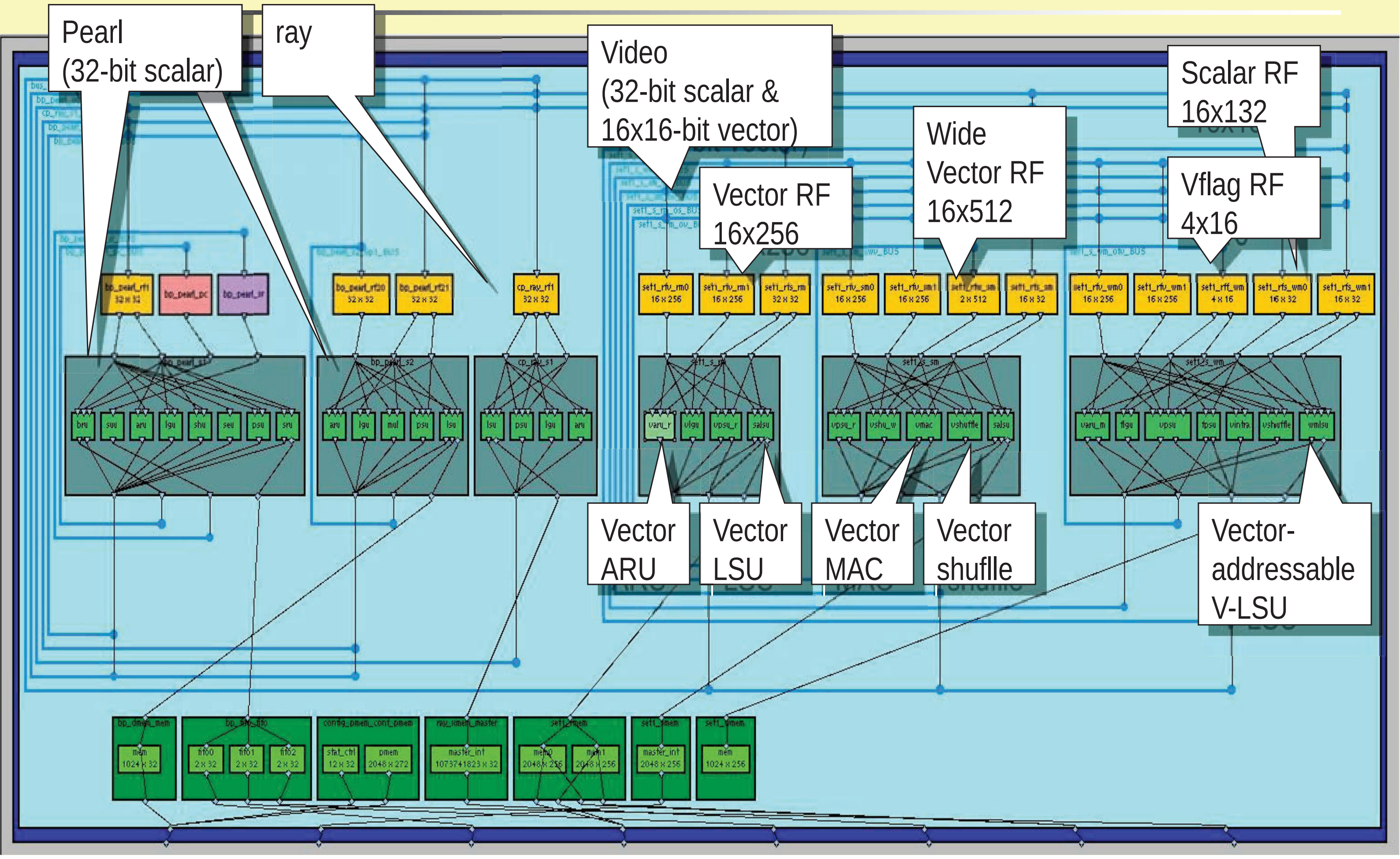
- To develop the complex multi-ASIP MPSoCs, a sophisticated design space exploration is necessary in which only the most promising ASIP and MPSoC architectures will be efficiently constructed, and the best of these architectures will be selected for further analysis, refinement and actual implementation
- The ASAM multi-ASIP MPSoC design-space exploration implements the *quality-driven model-based system design methodology*
- According to this methodology quality has to be modeled, measured, and compared
- The **quality** of the multi-ASIP MPSoC required **is modeled** in the form of the:
  - demanded system behavior (application C-code)
  - structural constraints: generic ASIP and MPSoC architecture templates and their pre-characterized generic parts included in the IP library, and
  - parametric constraints and objectives to be satisfied by the MPSoC design
- Based on the analysis of the so modeled required quality, the generic architecture templates are adequately instantiated and used in **design space exploration** that **constructs** one or several most promising MPSoC designs supporting the required behavior and satisfying the demanded constraints and objectives

34

Example of Generic WLIW ASIP Architecture Template
(Intel Benelux, used in ASAM project)



Example instances of the generic ASIP template: video processor



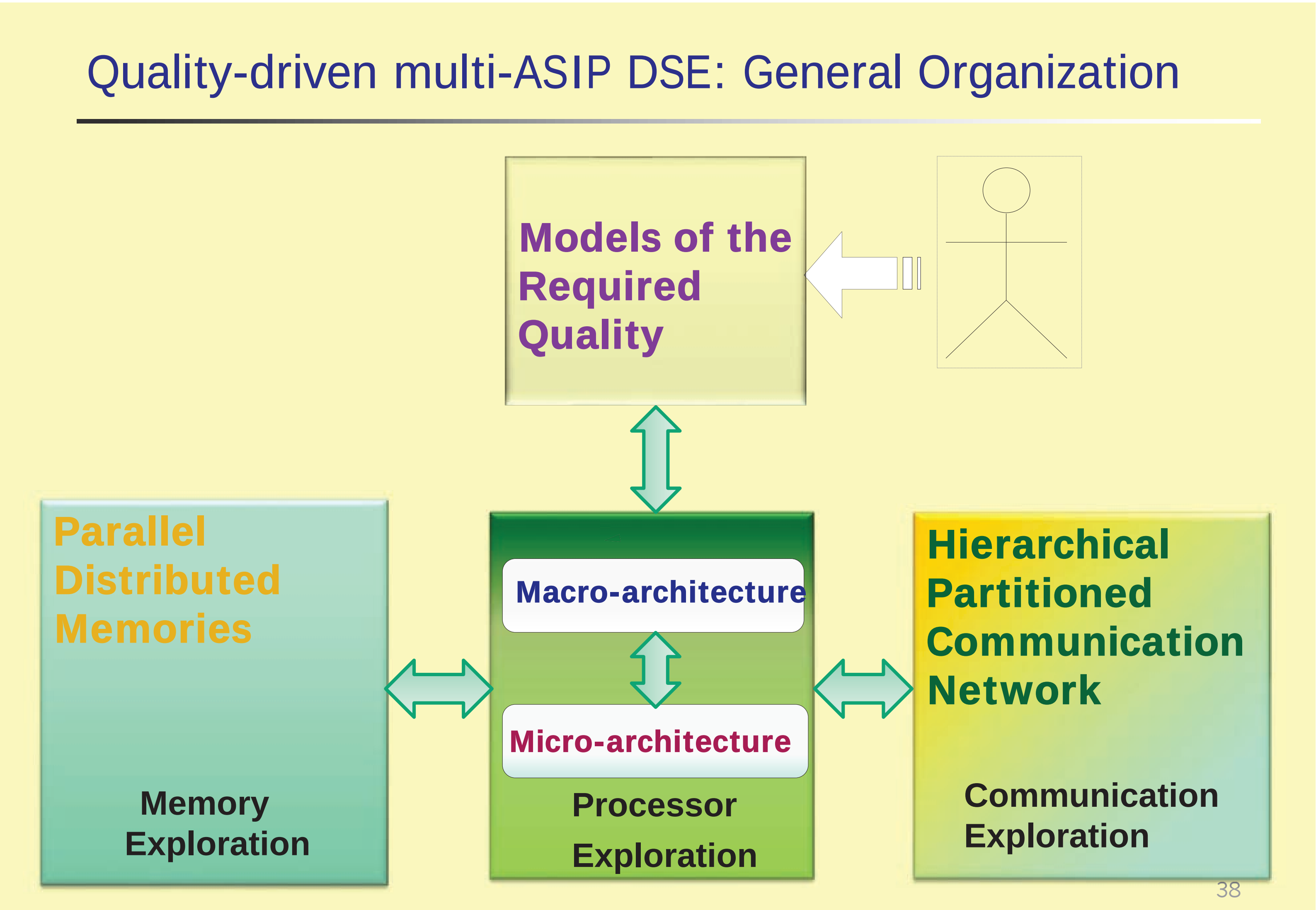
Quality-driven model-based automated design of multi-ASIP MPSoCs: Quality-driven DSE

- Based on the analysis of the so modeled required quality, the generic architecture template is adequately instantiated and used in **design space exploration** that aims at:
  - **analysis** of various architectural choices regarding:
    - processor micro-architectures and multi-processor macro-architecture
    - parallel memories architectures
    - parallel communication architectures
    - macro-/micro-architecture tradeoffs
    - processor, memory and communication tradeoffs,

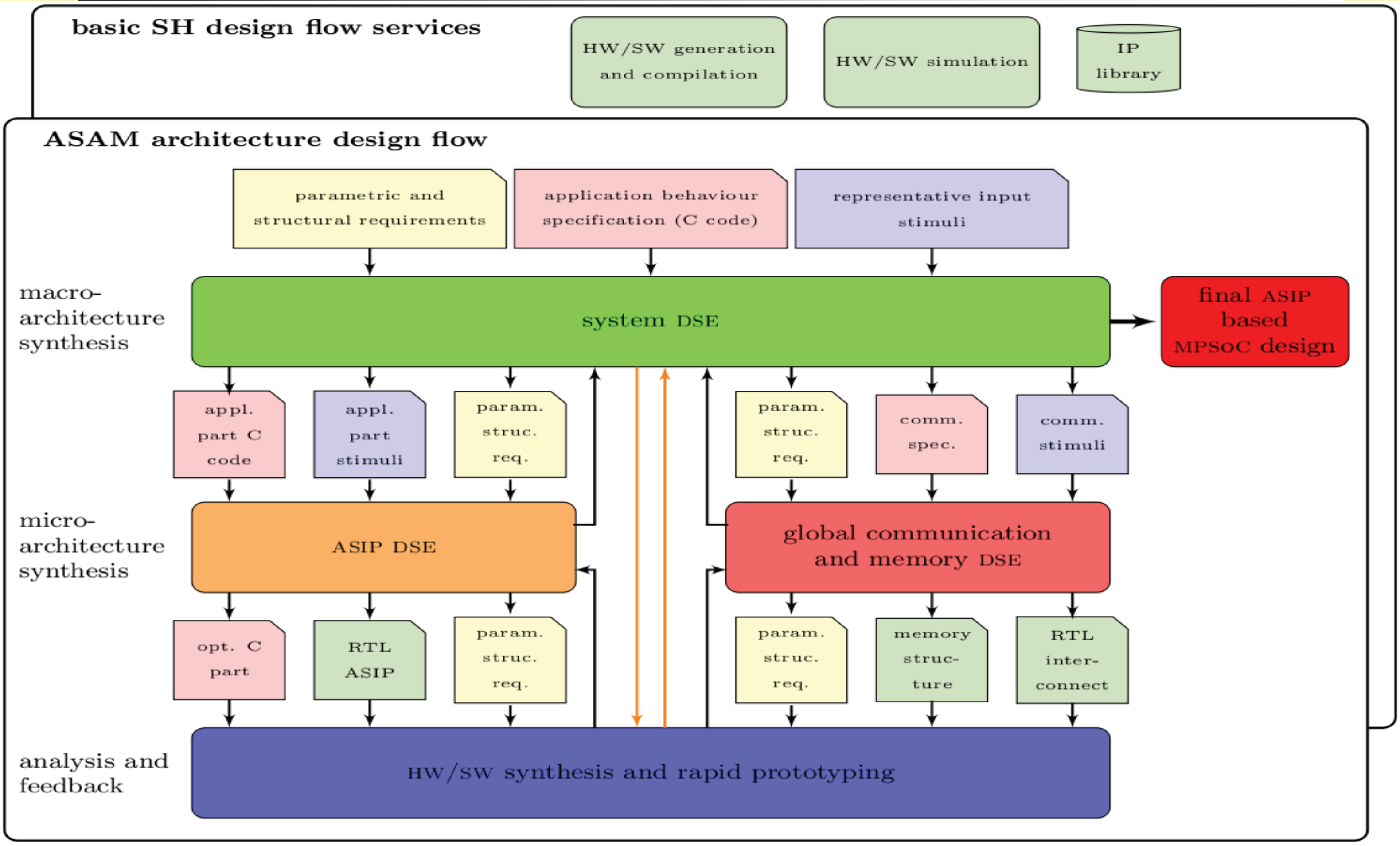
  and based on this analysis,

  - **construction** of one or several most promising (sub-)system architectures supporting the required behavior and satisfying the demanded constraints and objectives.

37


Quality-driven multi-ASIP DSE: General Organization
Models of the Required Quality
Parallel Distributed Memories
Memory Exploration
Macro-architecture
Micro-architecture
Processor Exploration
Hierarchical Partitioned Communication Network
Communication Exploration
38

ASAM main result: quality-driven design method, flow and tools for the automated synthesis of heterogeneous ASIP-based MPSoCs



Conclusion

- The huge and rapidly developing markets of sophisticated CPS and IoT represent great opportunities
- However, these opportunities come with a price of unusual system complexity and demands of an (ultra-)high performance and (ultra-)low energy consumption, while requiring a high reliability, safety and security
- This extreme complexity, demands and requirements cannot be adequately addressed without an adequate design methodology and design automation
- More than 20 years ago I proposed the **methodology of quality-driven model-based system design**, and 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
- This **research confirmed the adequacy of the quality-driven model-based 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)

40

GPU Virtualization and Remoting Service for AI Acceleration at the Edge

Nikhil Gaikwad[1], Ralf Lübben[2], Sokol Kosta[1]
[1]Aalborg University, Denmark
[2]Flensburg University of Applied Science, Germany

Project: 101097560 – CLEVER – HORIZON-KDT-JU-2021-2-RIA

Content

- CLEVER Use Cases
- Need for GPU Virtualization and Remoting for Edge
- GPU Virtualization vs GPU Remoting
- GPU Remoting for Edge
- State of the Art
- Introduction to GVirtuS
- GVirtuS' Split Driver Model
- GVirtuS Components
- Experimentation Setup and Demonstration
- Initial Results and Future Challenges

CLEVER Use Cases

Use Case 1: Digital Twin for in-factory Optimization

- Implement multi-step inspection on a rigid body using reinforcement learning
- Utilize digital model of e-drives for inspection
- Focus on movement stabilization and integration of additional sensors (camera, tactile, ultrasonic, etc.).
- Determine inspection points based on sensor data.

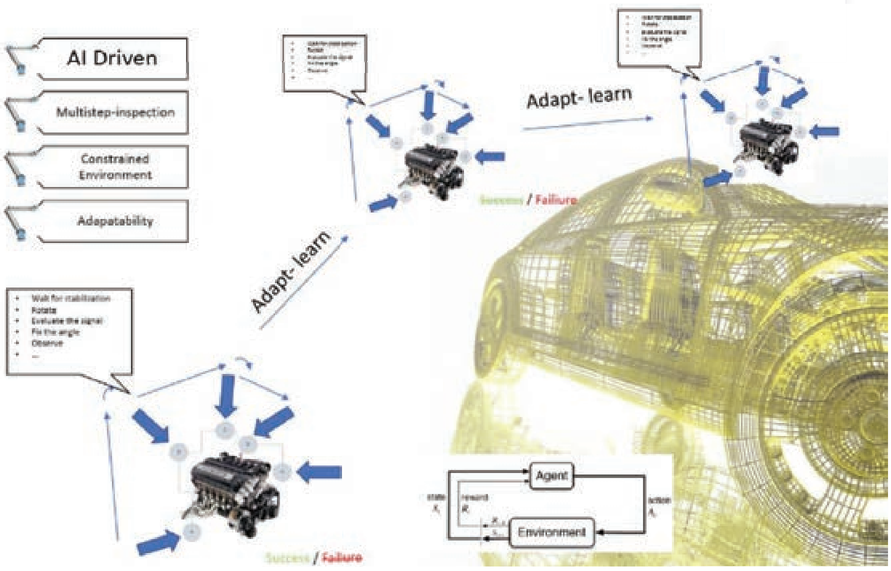


Fig 1: Digital twin for in-factory optimization architecture

CLEVER Use Cases

Use Case 2: Smart agriculture for high yield Eco-farms

- Real-time monitoring and analysis of orange production in eco-friendly farming practices
- Utilize edge device for offloading of image processing and AI algorithms to analyze fruit development
- Seamless integration with cloud-based platform for data storage and management
- Two Implementation Scenarios: Edge on Tractor and One Edge Four Cameras

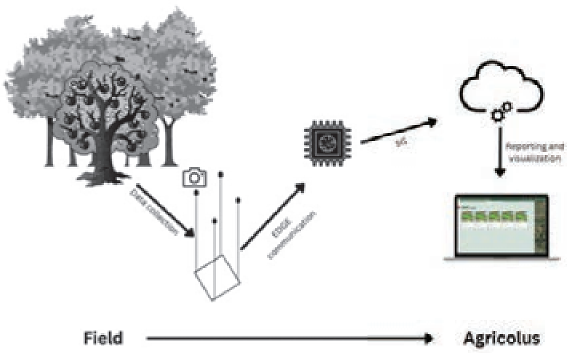


Fig 2: Smart agriculture high yield eco-farms architecture

CLEVER Use Cases

Use Cases 3: Augmented Reality (AR) for Shopping Sites

- AR relies heavily on wearable devices for real-world enhancement
- Limitations include data exchange constraints, computational requirements, and security concerns
- Overcomes limitations with edge-based processing and AI/ML execution
- Utilizes hardware acceleration for video processing and ensures data privacy

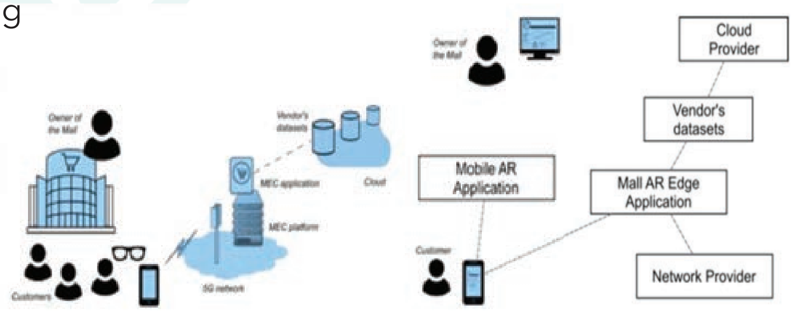


Fig 3: AR Augmented Reality for shopping sites architecture

Need for GPU Virtualization and Remoting for Edge

- Empowers edge devices with remote GPU access for AI, complex workloads
- Optimizes resource use by sharing a single physical GPU for multiple users
- Scales on-demand to meet changing processing needs at the edge
- Boosts security by processing data locally with virtual GPUs
- Unlocks advanced applications like real-time analytics and machine learning
- Reduces over all system cost

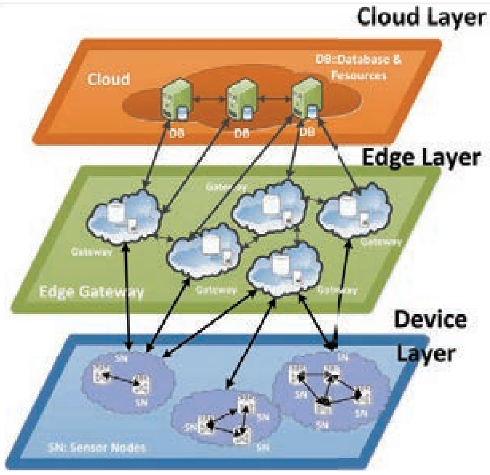


Fig 4: Edge computing for CLEVER Use Cases

26-Jun-24 KEY DIGITAL TECHNOLOGIES JOINT UNDERTAKING Project: 101097560 – CLEVER – HORIZON-KDT-JU-2021-2-RIA EUROPEAN PARTNERSHIP AALBORG UNIVERSITY 6

Need for GPU Virtualization and Remoting for Edge

On-premises

- Optimize computing infrastructure procurement
- Minimize the total cost of ownership
- Enable "improvement unGPGPUed" machines to GPGPU computing (minimize teaching costs, time to market)

On cloud

- Allocate computing resources in a better way
- "Rent" multiplexed GPGPUs - improve the business!
- Save money on the cloud bill

Fig 5: General Purpose GPU (GPGPU)

GPU Virtualization vs GPU Remoting

- **GPU Virtualization:** Allows multiple virtual machines (VMs) to share a single physical GPU, providing isolated & simultaneous access to GPU resources.
- **Resource Sharing:** Multiple VMs access the GPU simultaneously.
- **Isolation:** VMs operate independently.
- **Performance:** Slightly reduced due to shared resources.
- **Use Cases:** virtual desktop infrastructure (VDI), HPC, data centres, multi-user environments

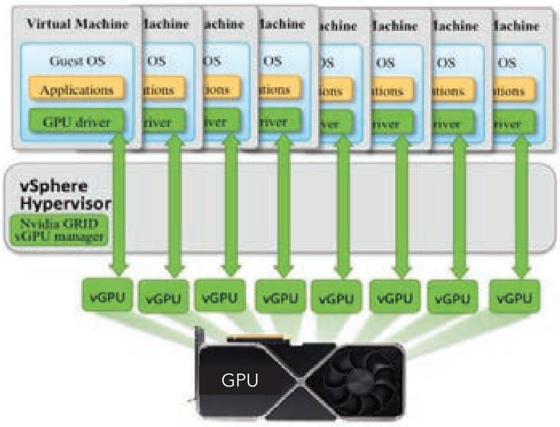


Fig. 6: GPU Virtualization with Multiple VMs [1]

GPU Virtualization vs GPU Remoting

- **Remoting:** Enables remote access to a GPU over a network, allowing devices to leverage GPU power from a different physical location.
- **Remote Access:** Use GPUs from anywhere with a network connection.
- **Flexibility:** Ideal for mobile devices or thin clients.
- **Performance:** affected by latency and bandwidth of Network.
- **Use Cases:** remote workstations, distributed computing, edge computing

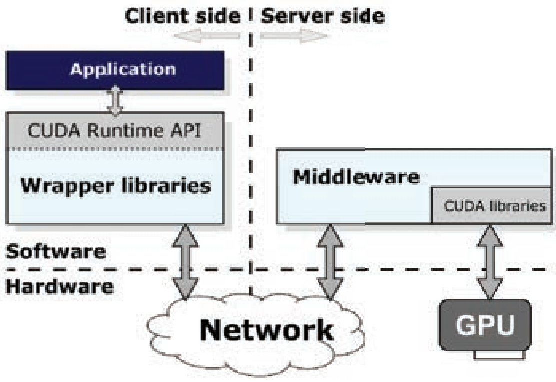


Fig. 7: GPU Remoting with Client Application [2]

GPU Remoting for Edge

- Use case 1: GPU Remoting for sharing GPU resources among servers connected with high-speed interfaces.
- Use case 2: GPU Remoting for sharing GPU resources among clients and devices connected via relatively slow and wireless interfaces.
- Limited research on accelerating AI workloads to share GPU resources has been done for the network edge, which belongs to use case 2.
- This work explores transparent GPU remoting using the GVirtuS framework for distributed AI workloads at the edge.
- This GPU remoting will be tuned according to the CLEVER use cases.

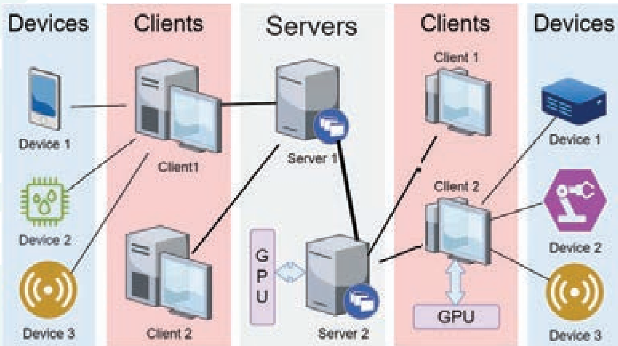


Fig. 8: GPU Remoting with a generalized use case.

State of the Art

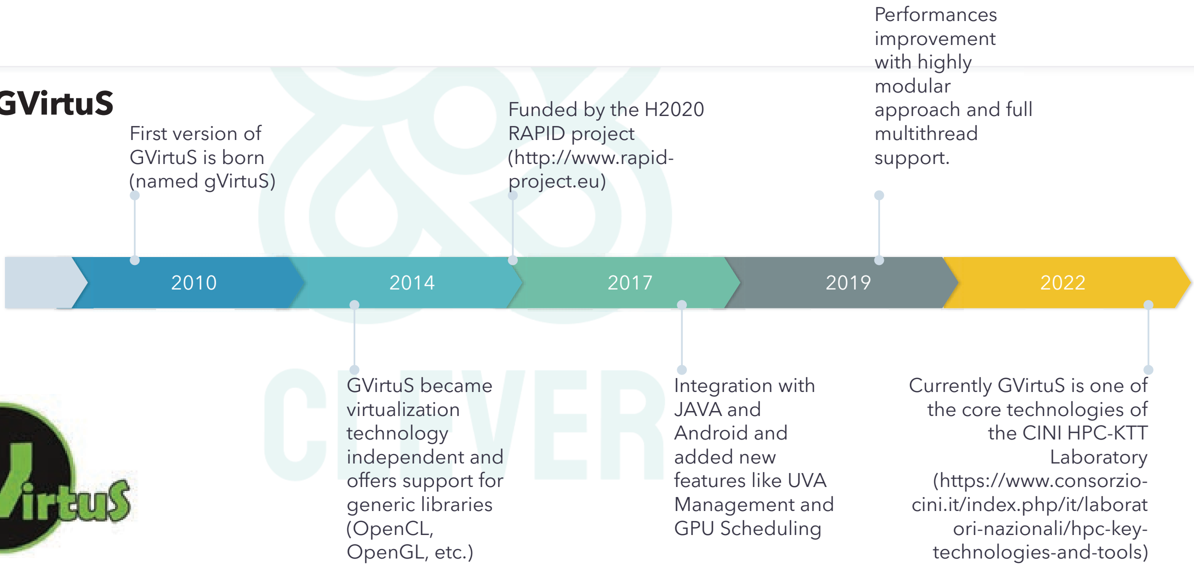
- Virtualization Framework

Virtualization Tools	Reference	On Suite		Remoting/ Applications			Communication Protocol
		Hardware	Socket	HPC	Data Center/ Cloud	Edge	
GViM	[1]	GPGPU system NVIDIA 8800 GTX PCIe	NA	Yes	NA	NA	InfiniBand
vCUDA	[2]	GPGPU, GTX470 of NVIDIA	NA	YES	NA	NA	TCP/IP protocol
rCUDA	[3]	NVIDIA Tesla C1060	Sockets API	Yes	Yes	NA	40 Gbps InfiniBand
	[4]	NVIDIA V100 GPU	NA	NA	NA	Yes	The sockets API (TCP)
GVirtuS	[5]	GPGPU Tesla 1060C plus	VMSocket	NA	Yes	NA	TCP/IP
	[6]	GPGPU NvidiaGeForceTitanXGPUs	NA	NA	Yes	NA	TCP/IP
Shadowfax	[7]	9800GTS GPGPU	NA	Yes	Yes	NA	1 Gbps Ethernet fabric
DS-CUDA	[8]	NVIDIA GeForce GTX 560 Ti	InfiniBand IBverb	NA	Yes	NA	InfiniBand and Gigabit
FairGV	[9]	NVIDIA TitanX	NA	YES	YES	NA	TCP/IP
HFGPU	[10]	NVIDIA V100	rsocket	YES	YES	NA	InfiniBand
NVIDIA virtual GPU (vGPU)	[11]	Most of the NVIDIA GPUs	CPU sockets	YES	YES	NA	NVLink

State of the Art

- **Introduction to GVirtuS**

Framework for facilitating the development of split-drivers for virtualization solutions

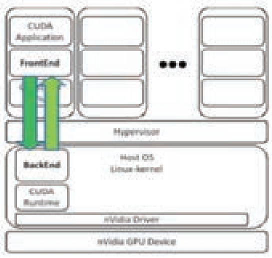


https://github.com/nicholaspiantadosi/GVirtuS

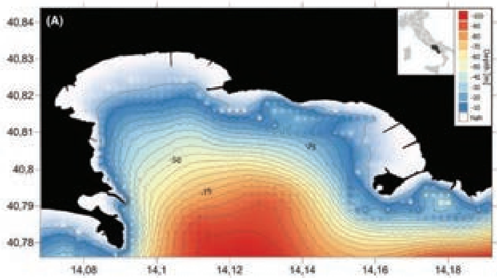
State of the Art

- **Applications of GVirtuS**

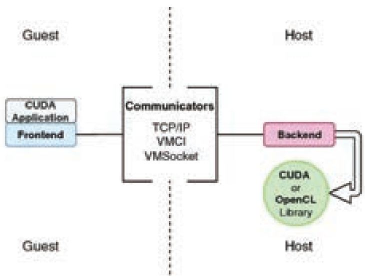
Image Denoising [12]



Interpolation Algorithms [6]



Matrix Multiplication [13]



State of the Art

- **Research Gaps**

The Limited research exists on accelerating computing workloads using GPU virtualization, especially with a focus on the Network Edge

- **Motivation**

Implementation of GPU Virtualization using the GVirtuS framework aims at sharing computing resources in Edge Scenarios

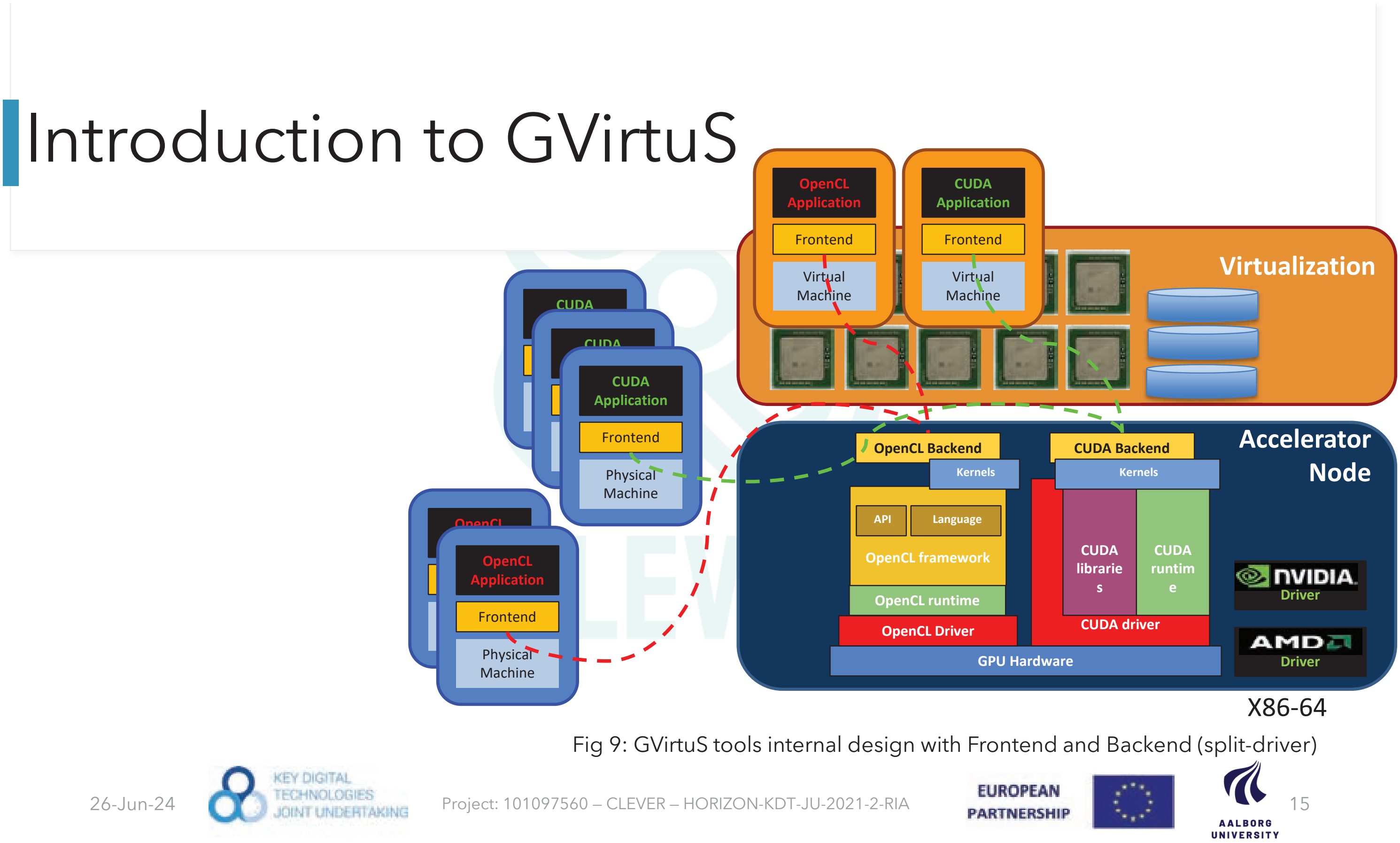


Fig 9: GVirtuS tools internal design with Frontend and Backend (split-driver)

Introduction to GVirtuS

- GVirtuS is a software component for GPGPU virtualization and "remoting".
- **Transparent:**
  - It is a "fake" CUDA runtime/driver library;
  - When a CUDA enabled binary invokes a CUDA function, it invokes a stub library function imitating the regular one.
- **Independent:**
  - Producers: machines hosting GPGPU devices
  - Consumers: machines running CUDA enabled binaries

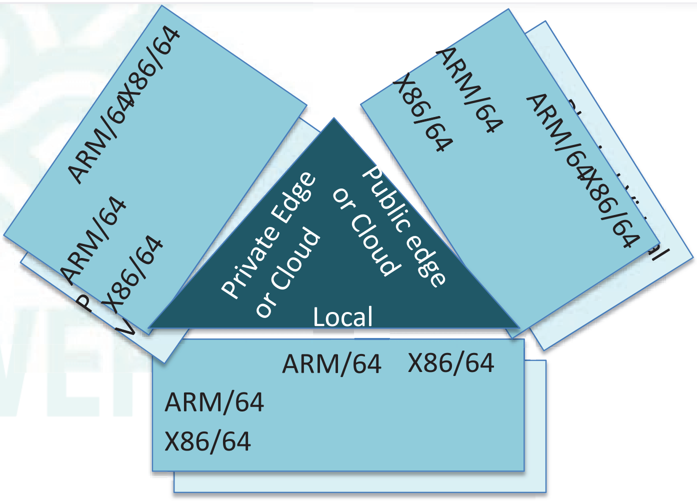


Fig 10: Execution Engine – Transparent/ Architecture

GVirtuS' Split Driver Model

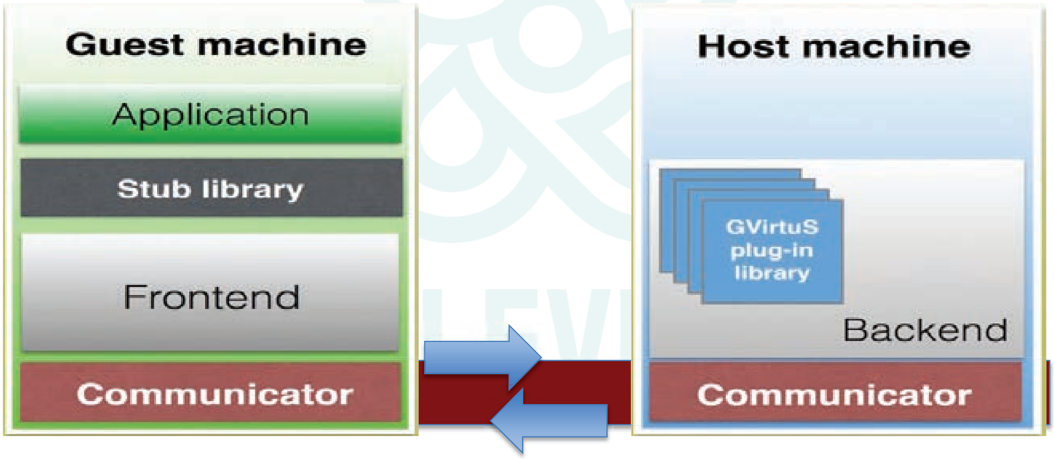


Fig 11:GVirtuS frontend and backend with Split Driver Model

GVirtuS’ Split Driver Model

- GVirtuS Frontend
  - Dynamic loadable library
  - Same application binary interface
  - Run on guest user space
- On cloud
- Server application
  - Run in host user space
  - Concurrent requests

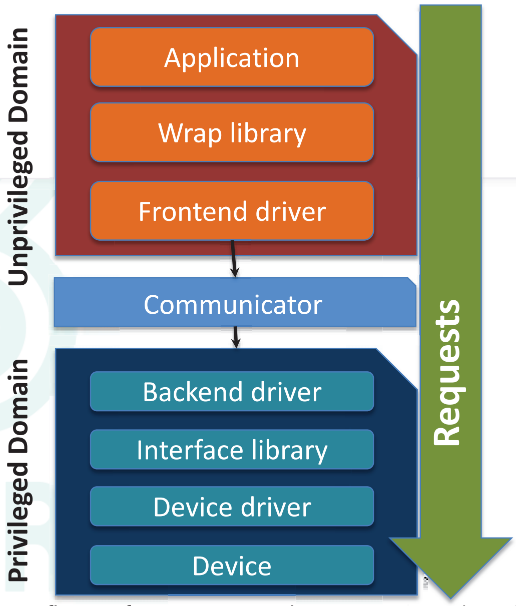
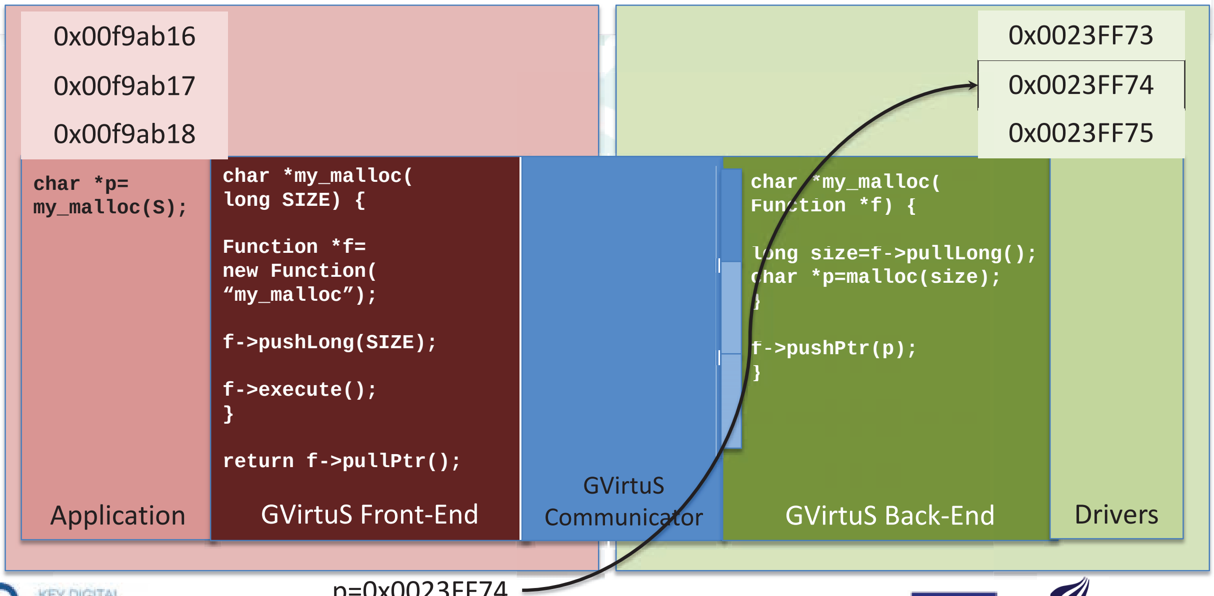


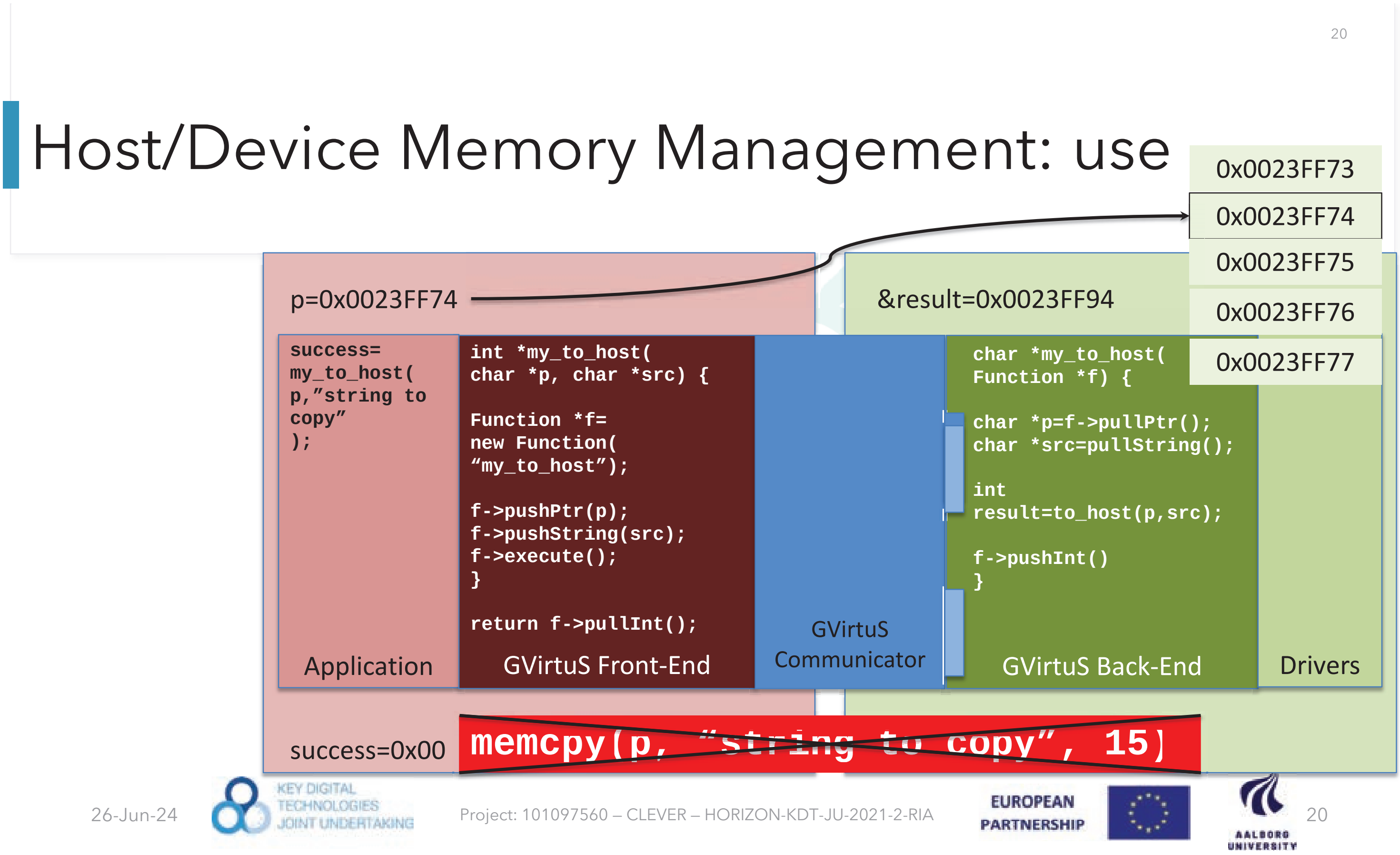
Fig 5: Execution flow of GVirtuS Tools: a. Frontend with dynamic loadable library and the same application binary interface runs in the guest user space. b. GVirtuS Backend with a server application that executes concurrent requests runs in the host user space.

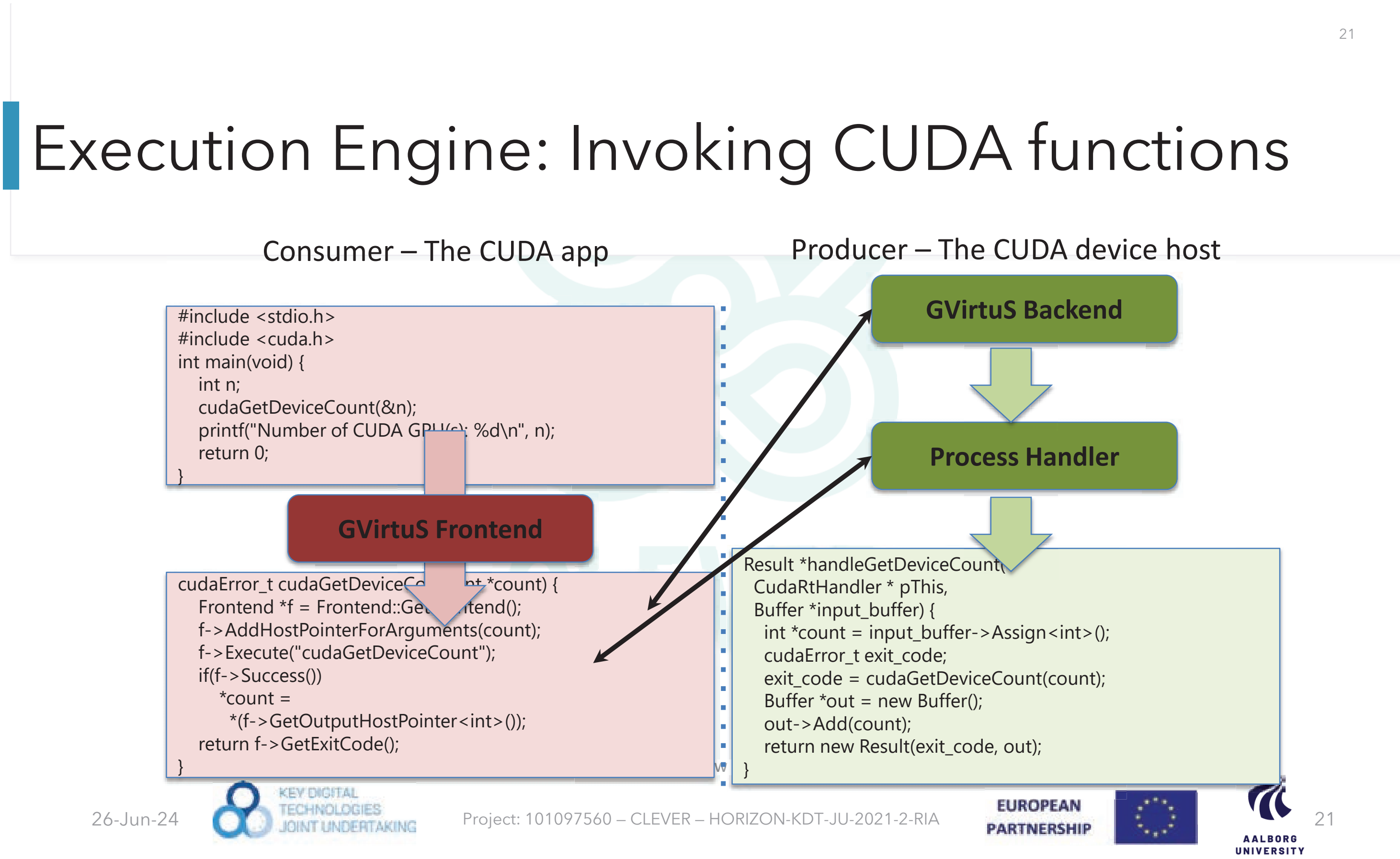
19

Host/Device memory management: allocation

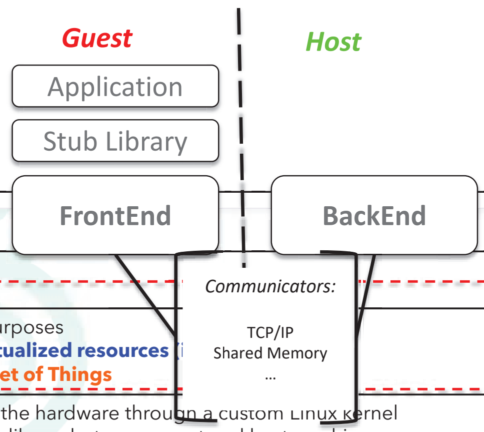


26-Jun-24 KEY DIGITAL TECHNOLOGIES JOINT UNDERTAKING Project: 101097560 – CLEVER – HORIZON-KDT-JU-2021-2-RIA EUROPEAN PARTNERSHIP AALBORG UNIVERSITY 19


20
Host/Device Memory Management: use
0x0023FF73
0x0023FF74
0x0023FF75
0x0023FF76
0x0023FF77
p=0x0023FF74
&result=0x0023FF94
success=
my_to_host(
p,"string to
copy"
);
Application
int *my_to_host(
char *p, char *src) {
Function *f=
new Function(
"my_to_host");
f->pushPtr(p);
f->pushString(src);
f->execute();
}
return f->pullInt();
GVirtuS Front-End
GVirtuS
Communicator
char *my_to_host(
Function *f) {
char *p=f->pullPtr();
char *src=pullString();
int
result=to_host(p,src);
f->pushInt()
}
GVirtuS Back-End
Drivers
success=0x00
~~memcpy(p, "string to copy", 15)~~
26-Jun-24
KEY DIGITAL TECHNOLOGIES JOINT UNDERTAKING
Project: 101097560 – CLEVER – HORIZON-KDT-JU-2021-2-RIA
EUROPEAN PARTNERSHIP
AALBORG UNIVERSITY
20


21
Execution Engine: Invoking CUDA functions
Consumer – The CUDA app
Producer – The CUDA device host
#include <stdio.h>
#include <cuda.h>
int main(void) {
int n;
cudaGetDeviceCount(&n);
printf("Number of CUDA GPU(s): %d\n", n);
return 0;
}
GVirtuS Frontend
cudaError_t cudaGetDeviceCo nt *count) {
Frontend *f = Frontend::Ge ntend();
f->AddHostPointerForArguments(count);
f->Execute("cudaGetDeviceCount");
if(f->Success())
*count =
*(f->GetOutputHostPointer<int>());
return f->GetExitCode();
}
GVirtuS Backend
Process Handler
Result *handleGetDeviceCount(
CudaRtHandler * pThis,
Buffer *input_buffer) {
int *count = input_buffer->Assign<int>();
cudaError_t exit_code;
exit_code = cudaGetDeviceCount(count);
Buffer *out = new Buffer();
out->Add(count);
return new Result(exit_code, out);
}
26-Jun-24
KEY DIGITAL TECHNOLOGIES JOINT UNDERTAKING
Project: 101097560 – CLEVER – HORIZON-KDT-JU-2021-2-RIA
EUROPEAN PARTNERSHIP
AALBORG UNIVERSITY
21

Communicator



Hypervisor	FE/BE communicator	Notes
No hypervisor	Unix Sockets	Used for testing purposes
Generic	TCP/IP	• Communication testing purposes • **Remote / Distributed virtualized resources** (i • **High Performance Internet of Things**
Xen	XenLoop	• runs directly on the top of the hardware through a custom Linux kernel • provides a communication library between guest and host machines • implements low latency and wide bandwidth TCP/IP and UDP connections • app transparent and offers an automatic discovery of the supported VMs
VMware	Virtual Machine Communication Interface (VMCI)	• commercial hypervisor running at the application level • provides a datagram API to exchange small messages • a shared memory API to share data • an access control API to control which resources a virtual machine can access • and a discovery service for publishing and retrieving resources
KVM/QEMU	VMchannel	• Linux loadable kernel module now embedded as a standard component • supplies a high performance guest/host communication • based on a shared memory approach

Experimentation Setup

- **Test Applications**

1. SAXPY $y_i = \alpha \cdot X_i + Y_i \quad for\ i = 1,2,..,n$

where, i ranges from 1 to n , representing the i^{th} element of the vectors X and Y. In the experimentation, SAXPY1 is n=1, SAXPY2 is n=10, SAXPY3 is n=100 and SAXPY4 is n=1000.

2. Image Classification using CNN

In the CNN forward-pass experimentation, CNN1 is single image, CNN2 is 10 numbers, CNN3 is 100 numbers, CNN4 is 1000 numbers for image classification executed.

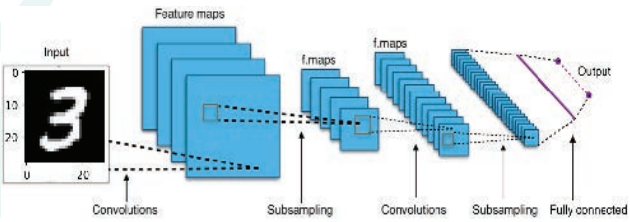


Fig. 6: CNN for number classification trained and tested on the MNIST dataset.

Experimentation Setup

- **Test Scenarios:**

Test 1: Normal GPU Execution

Test 2: Backend on Workstation and Frontend on VM (Virtual Machine)

Test 3: Backend on Workstation and Frontend on PC without GPU, connected by TCP/IP

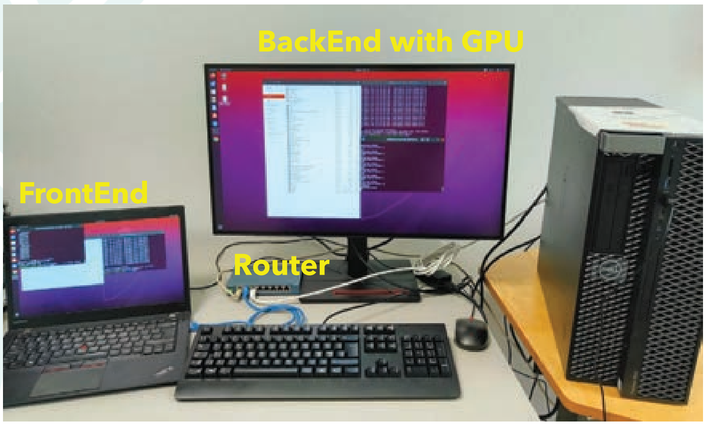


Fig 6: Experiment Setup

Demonstration

Frontend>

Backend>

Initial Results

- The AI offloading verified without losing any classification accuracy.
- However, execution is relatively slower compared to normal GPU execution.
- Latency performance can be improved by replacing TCP/IP with more advanced communication methods and optimizing the GVirtuS framework for optimum speed.

Results:

Table 1: Computational latency results (ms).

Tests	Test 1	Test 2	Test 3
SAXPY1	0.17	3.18	5.69
SAXPY2	0.19	3.73	6.34
SAXPY3	0.31	4.12	8.39
SAXPY4	1.75	10.43	11.91
CNN1	0.09	12.78	23.19
CNN2	0.76	139.98	273.45
CNN3	7.3	1125.9	3353.26
CNN4	27.3	13323.5	35636.6

Future Challenges

- Currently GvirtuS supports cudart, cublas, curand and cudnn.
- GVirtuS only offered full support for a limited range of CUDA functions and libraries.
- Extend GVirtuS CUDA functions and libraries to support AI applications of CLEVER use cases.
- The latest version of GVirtuS Tools only be set up using specific old versions of Ubuntu, CUDA, and cuDNN.
- GVirtuS Tools needs be the recent version compatible with CLEVER use case applications.
- GPU virtualization and remoting for embedded systems

References

[1] Vishakha Gupta, Ada Gavrilovska, Karsten Schwan, Harshvardhan Kharche, Niraj Tolia, Vanish Talwar, and Parthasarathy Ranganathan. 2009. GViM: GPU-accelerated virtual machines. In Proceedings of the 3rd ACM Workshop on System-level Virtualization for High Performance Computing. ACM, 17–24.

[2] Lin Shi, Hao Chen, and Jianhua Sun. 2009. vCUDA: GPU accelerated high performance computing in virtual machines. In Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, 2009 (IPDPS'09). IEEE, 1–11.

[3] Jose Duato, Antonio J. Pe ́ na, Federico Silla, Rafael Mayo, and Enrique S. Quintana-Ort ̃ ́ı. 2010b. rCUDA: Reducing the number of GPU-based accelerators in high performance clusters. In Proceedings of the 2010 International Conference on High Performance Computing and Simulation (HPCS'10). IEEE, 224–231.

[4] Peñaranda, Cristian, Carlos Reaño, and Federico Silla. "Exploring the use of data compression for accelerating machine learning in the edge with remote virtual graphics processing units." Concurrency and Computation: Practice and Experience 35.20 (2023): e7328.

[5] Giulio Giunta, Raffaele Montella, Giuseppe Agrillo, and Giuseppe Coviello. 2010. A GPGPU transparent virtualization component for high performance computing clouds. In Euro-Par 2010-Parallel Processing. Springer, 379–391.

[6] Montella, Raffaele, Livia Marcellino, Ardelio Galletti, Diana Di Luccio, Sokol Kosta, Giuliano Laccetti, and Giulio Giunta. "Marine bathymetry processing through GPGPU virtualization in high performance cloud computing." Concurrency and Computation: Practice and Experience 30, no. 24 (2018): e4895

[7] Alexander M. Merritt, Vishakha Gupta, Abhishek Verma, Ada Gavrilovska, and Karsten Schwan. 2011. Shadowfax: Scaling in heterogeneous cluster systems via GPGPU assemblies. In Proceedings of the 5th International Workshop on Virtualization Technologies in Distributed Computing. ACM, 3–10.

References

[8] Masahiro Oikawa, Atsushi Kawai, Keigo Nomura, Koichi Yasuoka, Kenichi Yoshikawa, and Tetsu Narumi. 2012. DS-CUDA: A middleware to use many GPUs in the cloud environment. In Proceedings of the 2012 SC Companion to High Performance Computing, Networking, Storage and Analysis (SCC). IEEE, 1207–1214.

[9] Hong, Cheol-Ho, Ivor Spence, and Dimitrios S. Nikolopoulos. "FairGV: fair and fast GPU virtualization." IEEE Transactions on Parallel and Distributed Systems 28.12 (2017): 3472-3485.

[10] Gonzalez, Nelson Mimura, and Tonia Elengikal. "Transparent I/O-aware GPU virtualization for efficient resource consolidation." 2021 IEEE international parallel and distributed processing symposium (IPDPS). IEEE, 2021.

[11] NVIDIA virtual GPU (vGPU) User Guide: https://docs.nvidia.com/grid/latest/pdf/grid-vgpu-user-guide.pdf Product page: https://www.nvidia.com/en-us/data-center/virtual-solutions/

[12] Galletti, Ardelio, Livia Marcellino, Raffaele Montella, Vincenzo Santopietro, and Sokol Kosta. "A virtualized software based on the NVIDIA cuFFT library for image denoising: performance analysis." Procedia computer science 113 (2017): 496-501.

[13] Mentone, Antonio, Diana Di Luccio, Luca Landolfi, Sokol Kosta, and Raffaele Montella. "CUDA virtualization and remoting for GPGPU based acceleration offloading at the edge." In International Conference on Internet and Distributed Computing Systems, pp. 414-423. Cham: Springer International Publishing, 2019.

[14] MNIST Dataset: https://www.kaggle.com/datasets/hojjatk/mnist-dataset

Thank You !

MYRTUS

Adaptive CNN execution on edge FPGAs

Francesco Ratto, Federico Manca and Claudio Rubattu
{fratto, fmanca2, crubattu}@uniss.it

HORIZON EUROPE

European Commission

Outline

MYRTUS

Introduction

HORIZON EUROPE

European Commission

Who we are

The University of Sassari (**UNISS**) was founded in 1558:
- over 10.000 students
- 34 bachelor courses
- 28 master courses
- 7 international degree courses (joined with other foreign Universities)

From August 2024 the first Engineering Department of the University of Sassari will be up and running

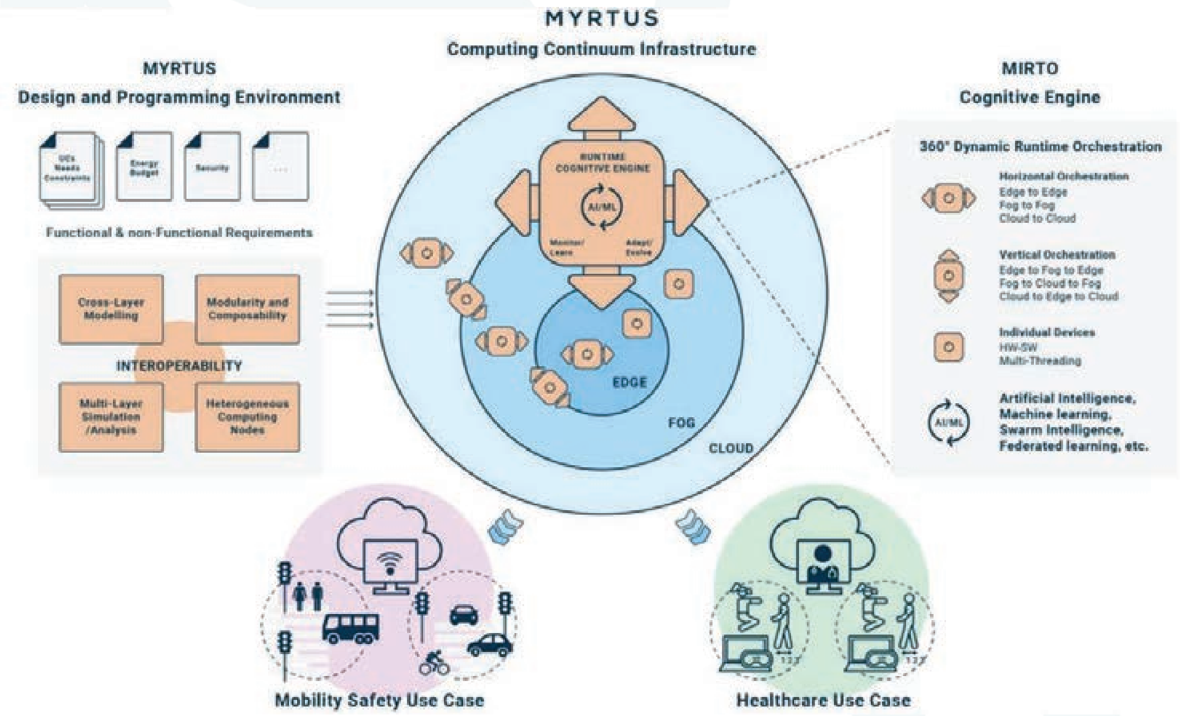
Who we are

Francesco Ratto is a Postdoc Researcher at UNISS.

Federico Manca is an Assistant Researcher at UNISS.

Claudio Rubattu is an Assistant Professor at UNISS.

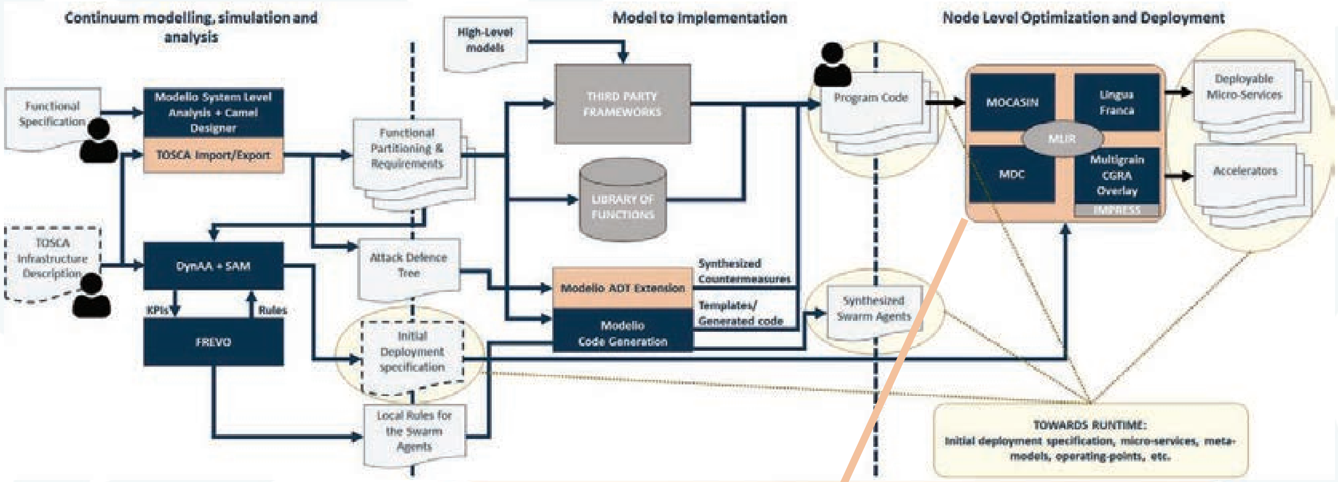
MYRTUS approach



The **MYRTUS consortium** comprises 8 countries and **14 partners**, and brings together different types of competencies from low-level architectural details definition to software management strategies.

MYRTUS has received funding from the European Commission for ~5,6M €.

Activities in MYRTUS



Multi-dataflow Composer tool extension:
support for approximate computing and
CNN deployment from ONNX

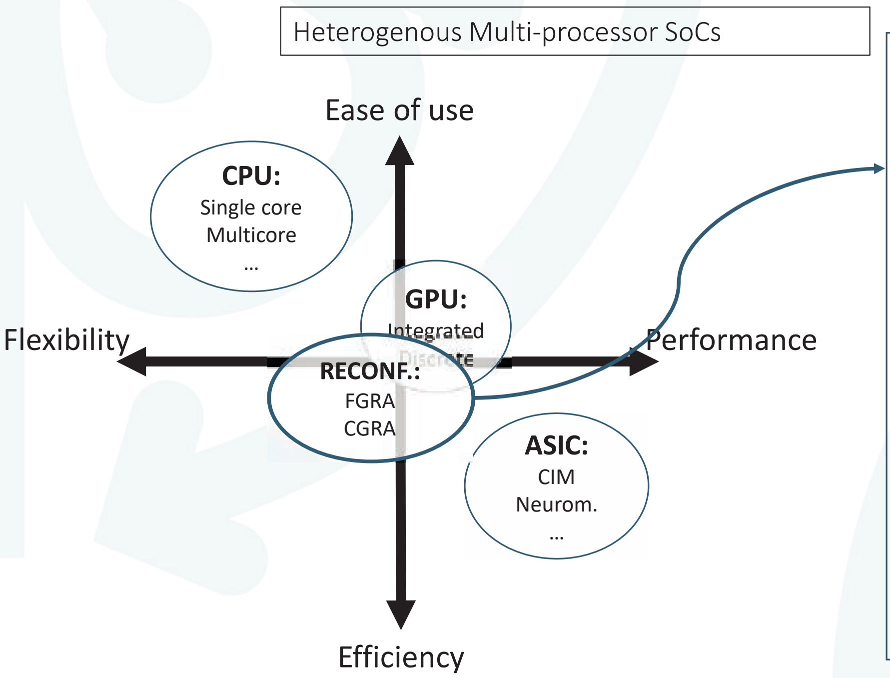
MYRTUS

Model inference on reconfigurable edge devices

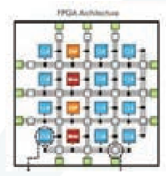
HORIZON EUROPE

European Commission

Reconfigurable architectures



- **Fine Grained (FG):**
  - FPGA only
  - Bitstream load

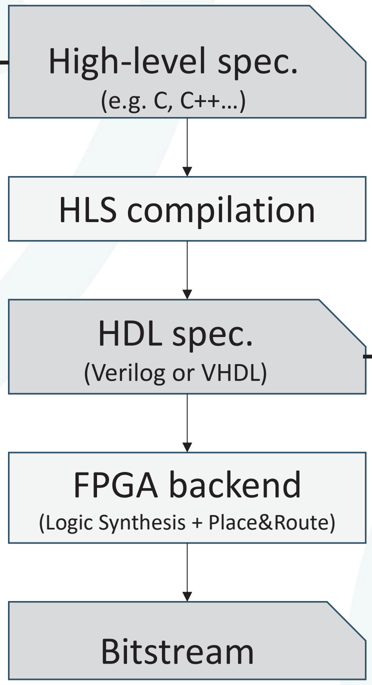


- **Coarse Grained (CG):**
  - both in ASIC and FPGA
  - Register configuration

	FG	CG
Granular.	bit-level	word-level
Flexibility	☺	😐
Speed	😐	☺
Memory	☹	☺

High-level Synthesis

```
int add(int a, int b){
        return a+b;
}
```



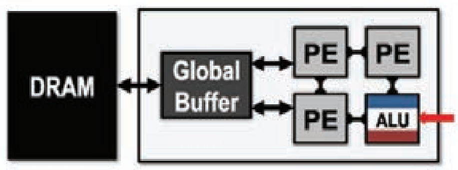

```
module add (
        input   ap_start;
        output  ap_done;
        output  ap_idle;
        output  ap_ready;
        input  [31:0] a;
        input  [31:0] b;
        output  [31:0]
ap_return;
);
assign ap_done = ap_start;
assign ap_idle = 1'b1;
assign ap_ready = ap_start;
assign ap_return = (b + a);
endmodule //add
```

Architectures for CNN inference on FPGAs

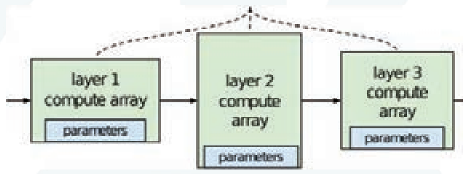
- a **vector processor** with instructions specific to accelerating the primitives' operations of convolution [Cococcioni, Garofalo].

RISC-V
RISC-V: The Free and Open RISC Instruction Set Architecture

- A **single processing engine**, usually in the form of a systolic array [Cnp, FPDNN, NEURAGHE, AMD-DPU].



- a **streaming/dataflow architecture**, consisting of one processing engine per network layer [FINN, HLS4ml, Ratto].



Programmability ⟷ Specialization

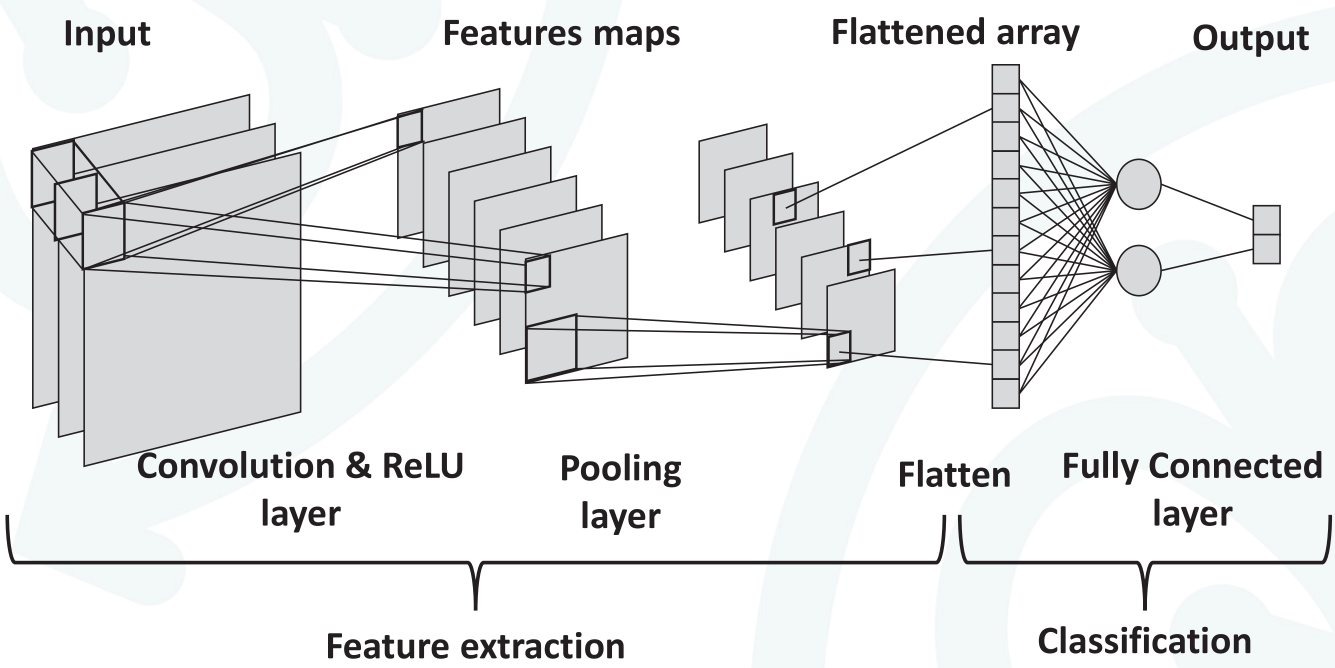
MYRTUS

Model training for reconfigurable edge devices

HORIZON EUROPE

European Commission

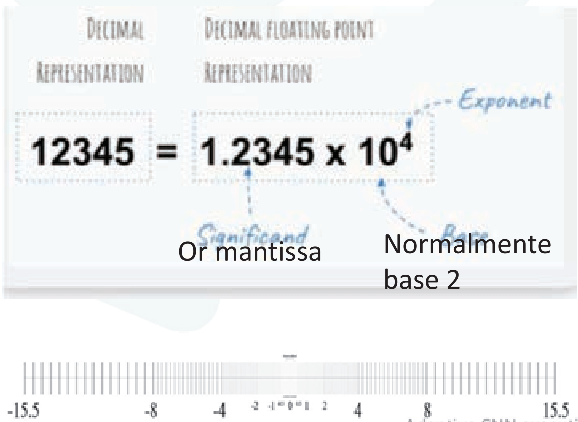
Convolutional Neural Networks (CNN)



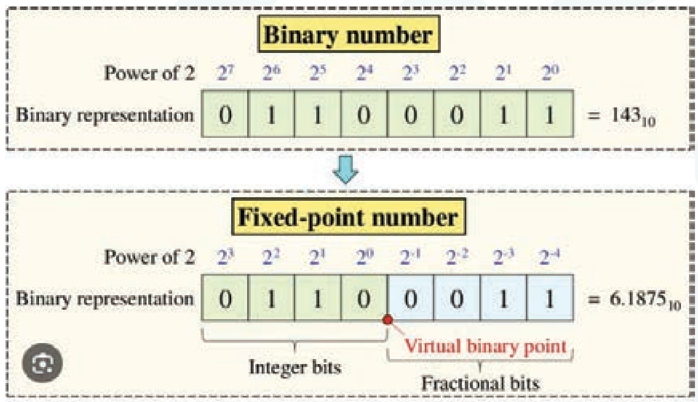
Reducing CNN Complexity: Approximation

Approximate computing trades off computation quality with effort expended [Mittal]

Floating point



Fixed point



Reducing CNN Complexity: Approximation

Approximate computing trades off computation quality with effort expended [Mittal]

Floating point

Decimal Representation — Mantissa — Exponent

$$12345 = 1.2345 \times 10^4$$

IEEE 754 float

S	Exponent	Mantissa
1 bit	8 bits	23 bits

Fixed point

-2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
0	1	1	0	1	0

$= 3.25_{10}$

Integer bits — Fractional bits

Reducing CNN Complexity: Quantization

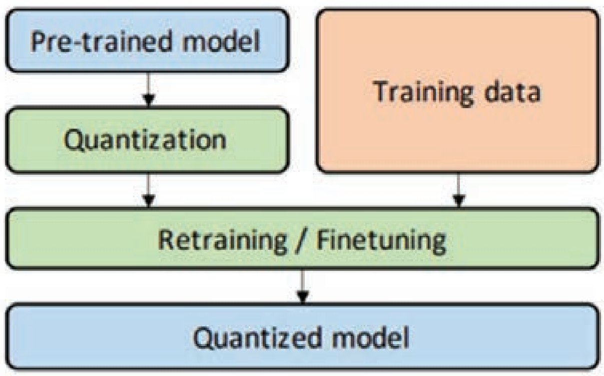
Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.

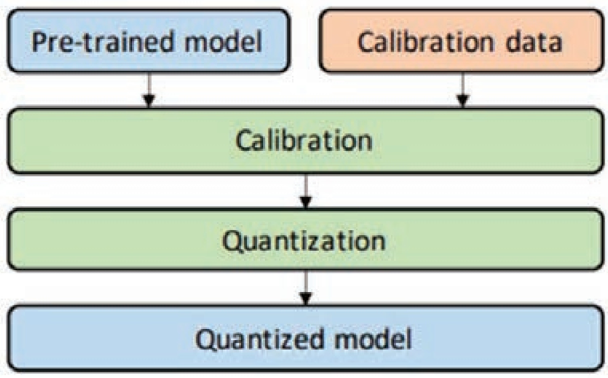
The former are represented by floating-point values, the latter by **fixed-point** values:

$$Q(x) = \Delta \times \left\lfloor \frac{x}{\Delta} \right\rceil$$

Quantization is particularly relevant in applications like NNs that have demonstrated remarkable **resilience to errors** [Hubara].

Reducing CNN Complexity: Training





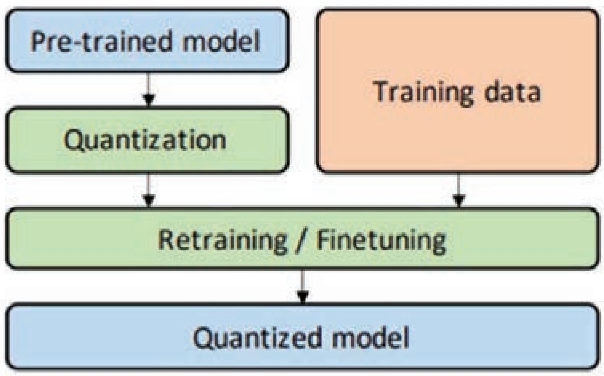
Quantization-Aware Training (QAT):

- achieves higher accuracy;
- It uses quantized data in the forward pass and float in the backward pass [Gholami];

Post-Training Quantization (PTQ):

- It is faster and simpler than QAT;

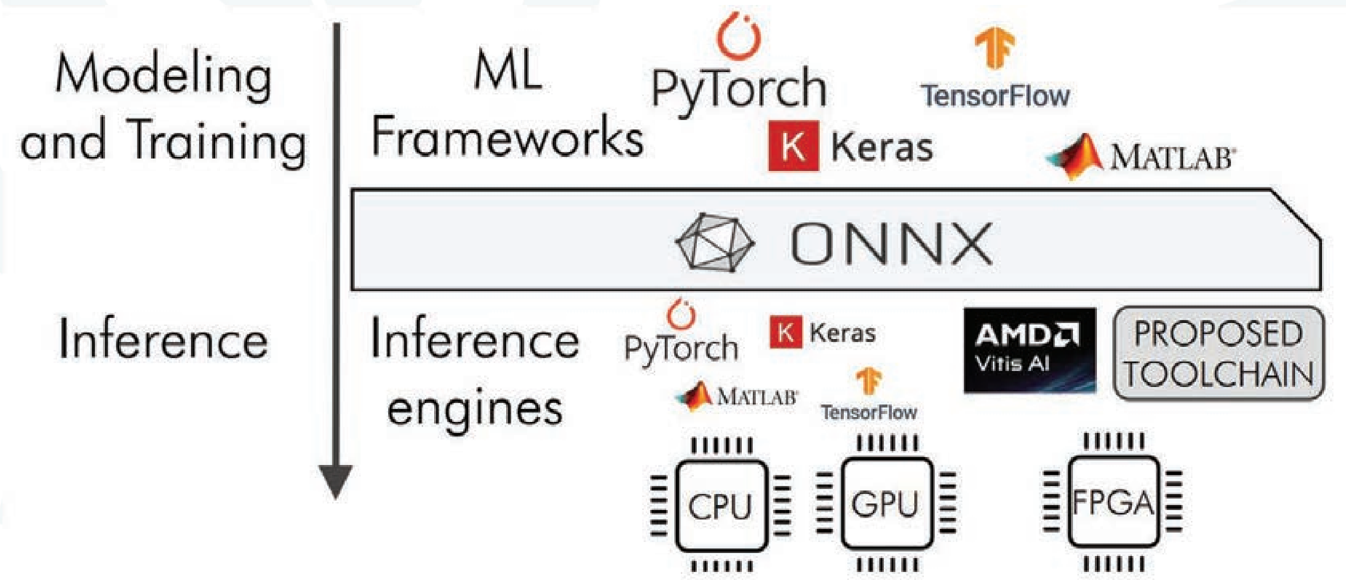
Reducing CNN Complexity: QAT libraries



Quantization-Aware Training (QAT):
- achieves higher accuracy
- It uses quantized data in the forward pass and float in the backward pass

Library	From	API
Brevitas	Xilinx	Pytorch
Larq	Larq	Keras - TF
QKeras	Google	Keras - TF

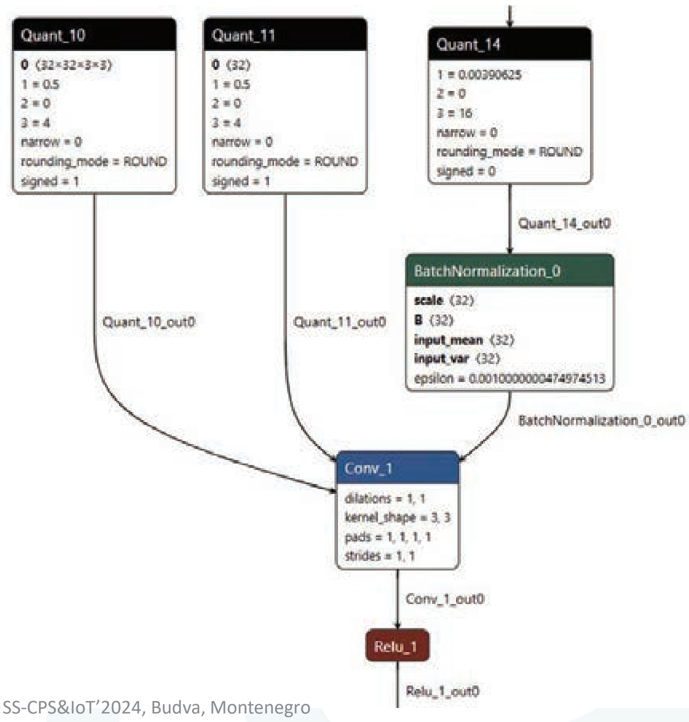
Exporting a CNN: the ONNX format



Exporting a CNN: the QONNX format

QONNX (Quantized ONNX) introduces new custom operators for quantization to represent arbitrary-precision uniform quantization in ONNX [Qonnx]:

- Quant
- BipolarQuant
- Trunc



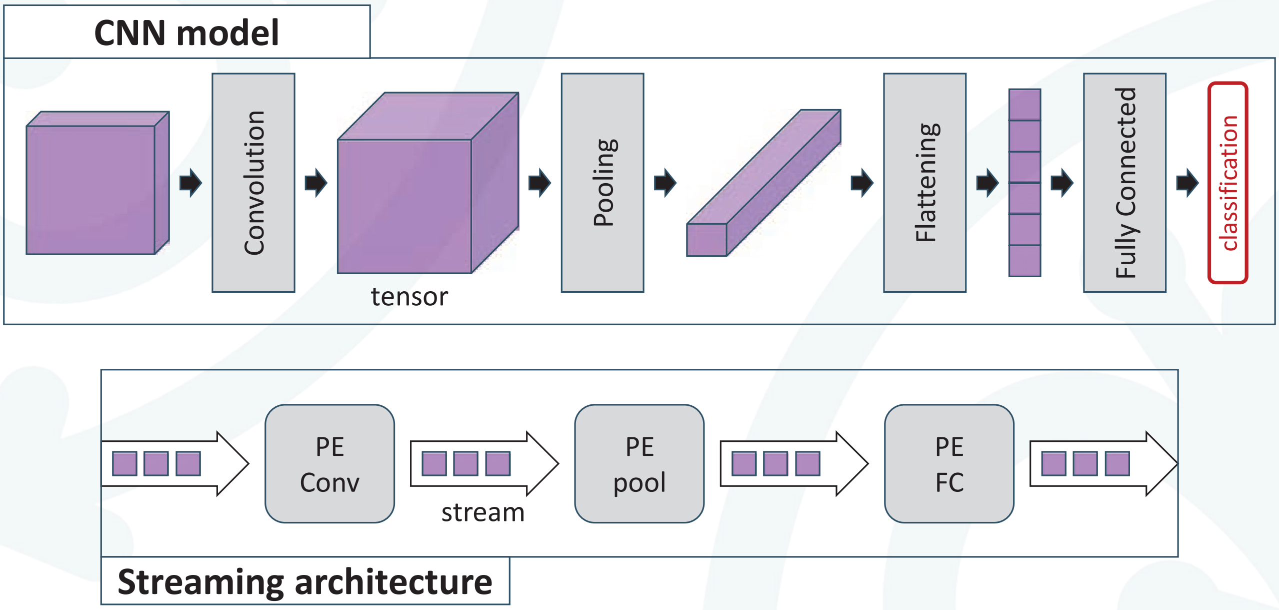
MYRTUS

MYRTUS Toolchain for CNN Inference on FPGAs

HORIZON EUROPE

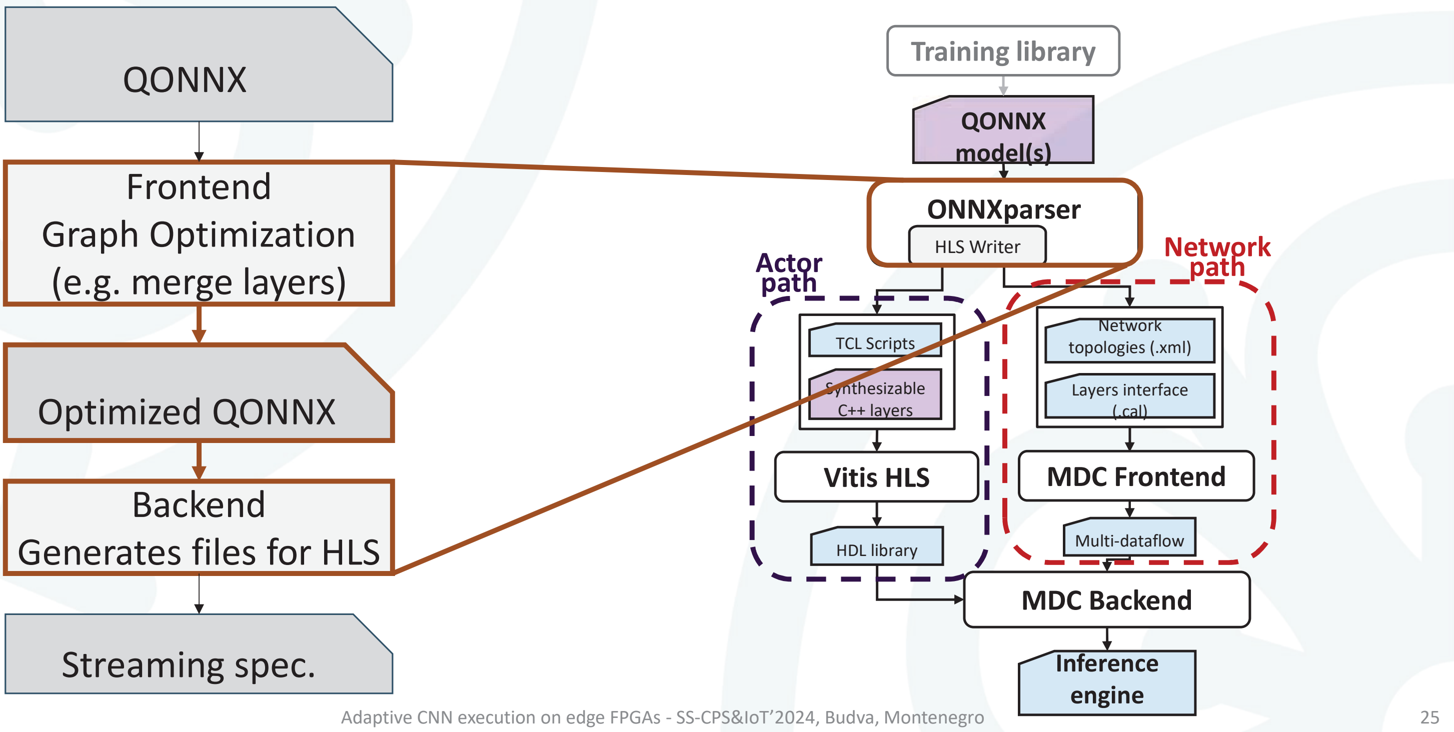
European Commission

From a CNN to a Streaming Architecture



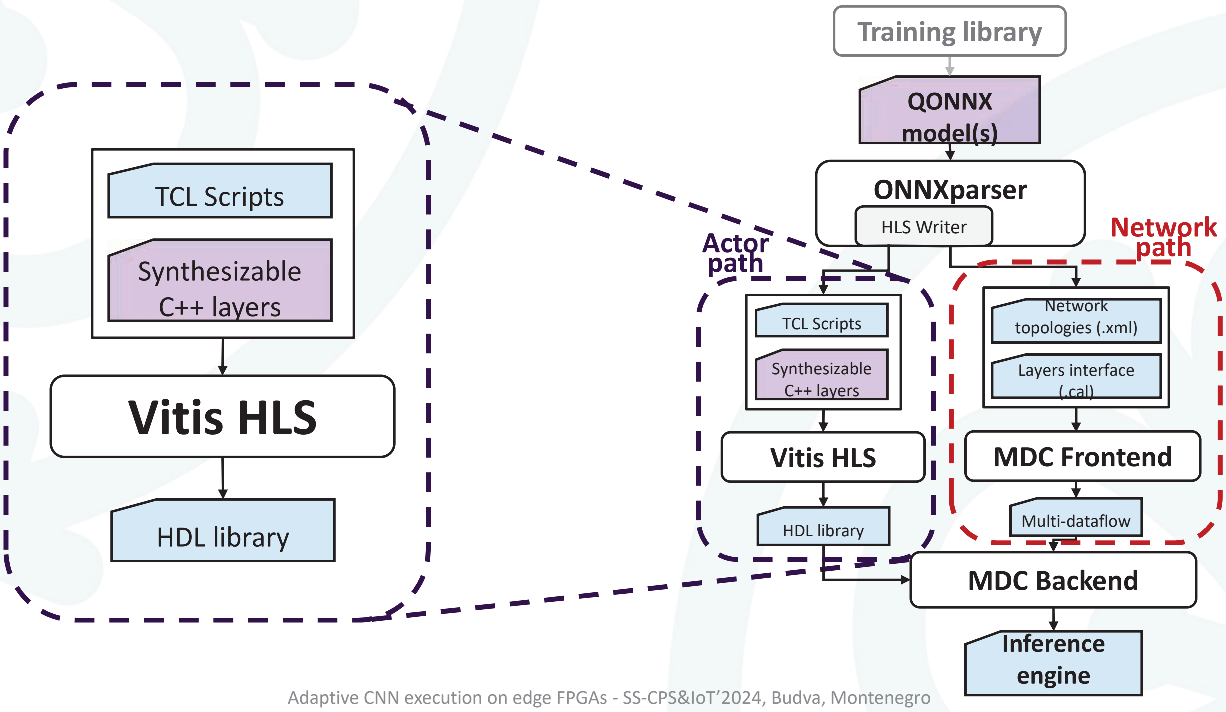
Adaptive CNN execution on edge FPGAs - SS-CPS&IoT'2024, Budva, Montenegro 24

From QONNX to a Streaming Specification

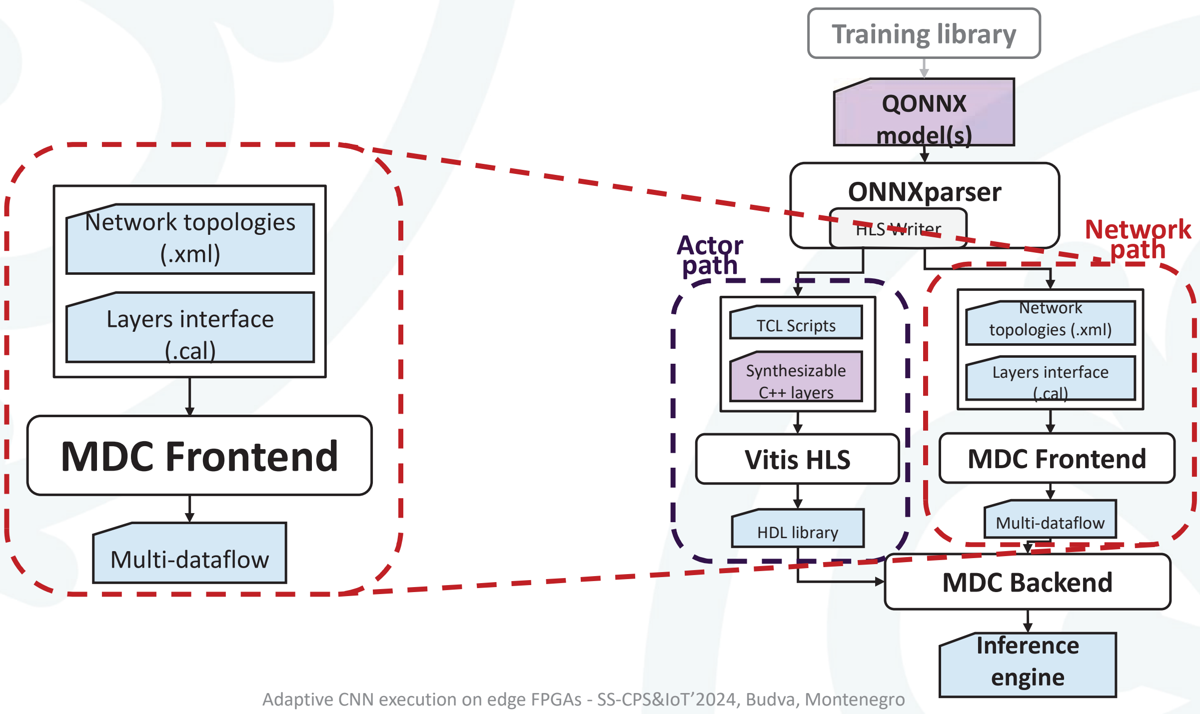


Adaptive CNN execution on edge FPGAs - SS-CPS&IoT'2024, Budva, Montenegro 25

Streaming Architecture Synthesis: Actors

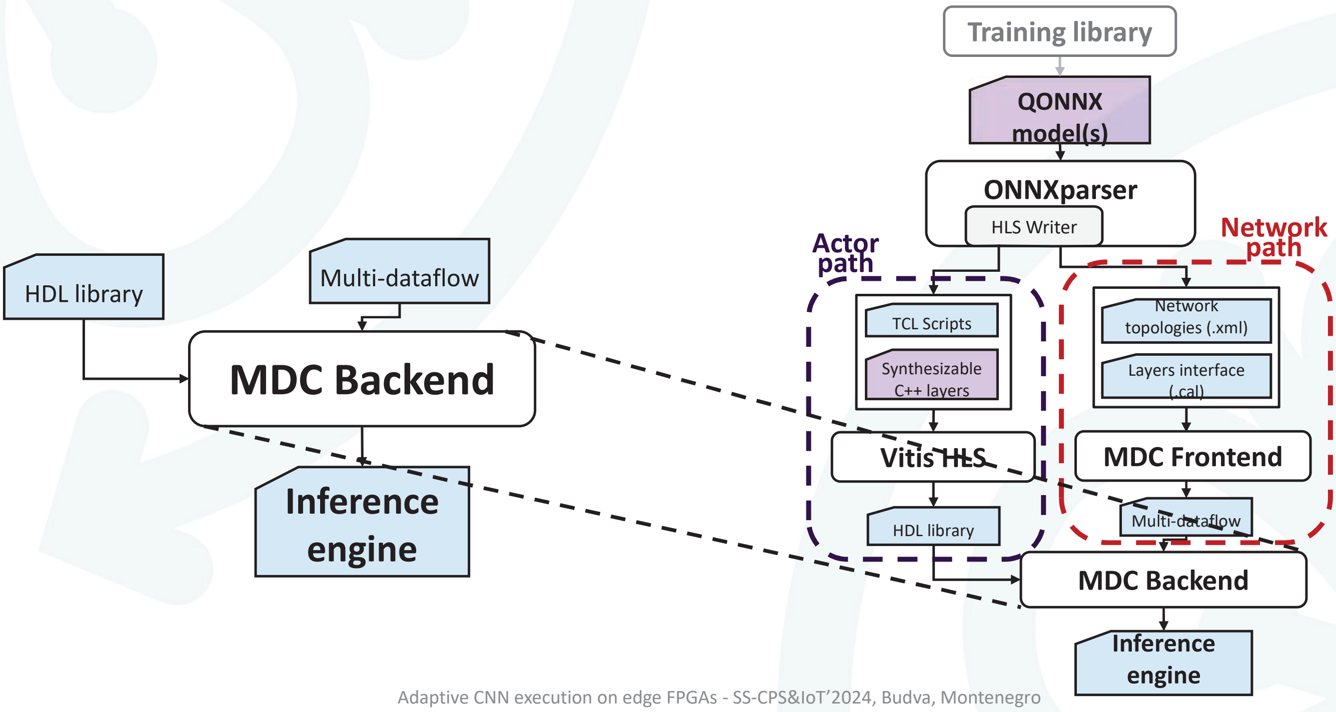


Streaming Architecture Synthesis: Network

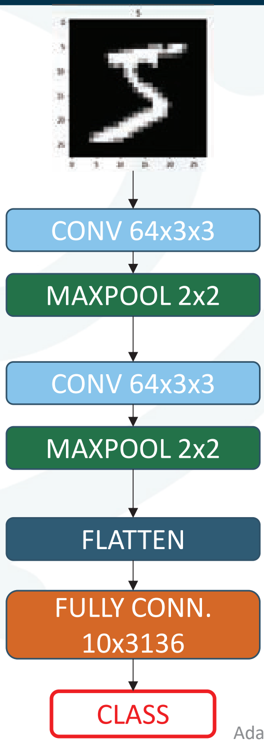


Adaptive CNN execution on edge FPGAs - SS-CPS&IoT'2024, Budva, Montenegro 27

Streaming Architecture Synthesis



Results: tiny CNN for MNIST classification

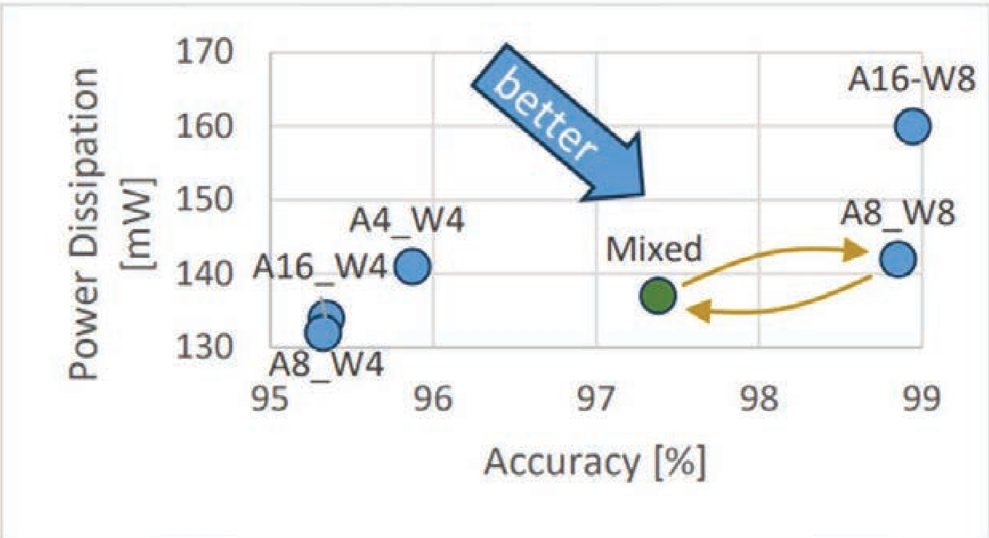


Application: tiny CNN for MNIST classification [Manca]

Board: AMD KRIA SoM

Adaptive CNN execution on edge FPGAs - SS-CPS&IoT’2024, Budva, Montenegro 29

Results: Execution trade-offs

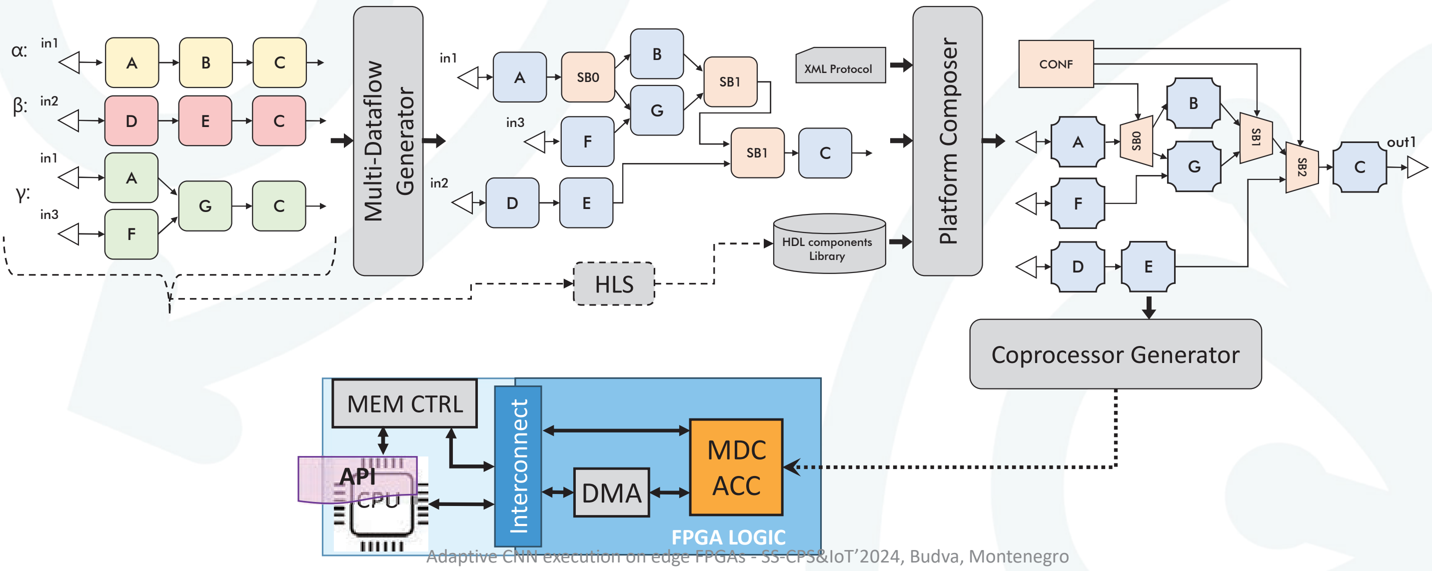


Ax_Wy:

- **x** is the number of bits used for representing **activations**;
- **y** is the number of bits used for representing **weights**;

Towards Adaptivity: Multi-Dataflow Composer

- MDC is an open-source tool for designing and deploying CG reconfigurable accelerators [Sau].

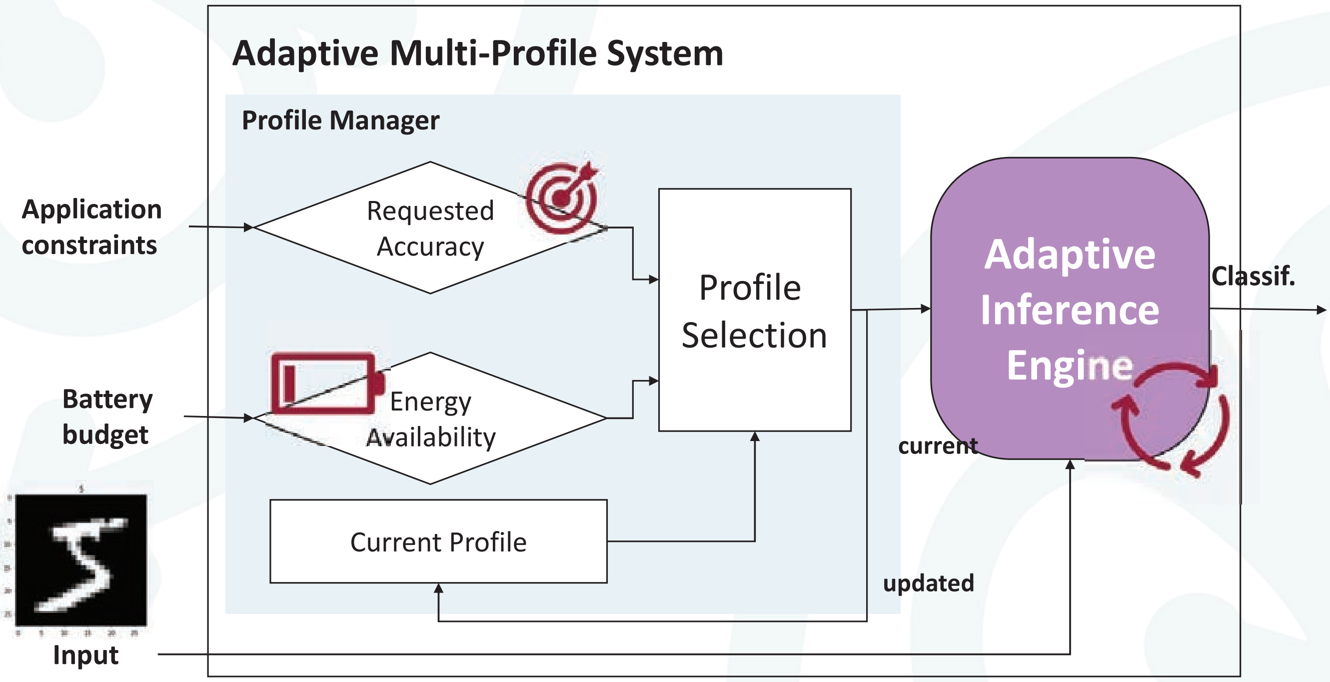


Results: Execution trade-offs

Ax_Wy:

- **x** is the number of bits used for representing **activations**;
- **y** is the number of bits used for representing **weights**;

Towards Adaptivity: MYRTUS Approach



Adaptive CNN execution on edge FPGAs - SS-CPS&IoT'2024, Budva, Montenegro 33

Bibliography

1. Vivienne **Sze**, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. "Efficient processing of deep neural networks: A tutorial and survey." In: Proceedings o
2. Marco **Cococcioni**, Federico Rossi, Emanuele Ruffaldi, and Sergio Saponara. "A lightweight posit processing unit for RISCV processors in deep neural network applications." In: IEEE Transactions on Emerging Topics in Computing 10.4 (2021), pp. 1898– 1908. f the IEEE 105.12 (2017), pp. 2295– 2329.
3. Angelo **Garofalo**, Manuele Rusci, Francesco Conti, Davide Rossi, and Luca Benini. "PULP-NN: accelerating quantized neural networks on parallel ultra-low-power RISC-V processors." In: Philosophical Transactions of the Royal Society A 378.2164 (2020), p. 20190155.
4. Clément Farabet, Cyril Poulet, Jefferson Y Han, and Yann LeCun. "**Cnp**: An fpga-based processor for convolutional networks." In: 2009 International Conference on Field Programmable Logic and Applications. IEEE. 2009, pp. 32–37.
5. Yijin Guan, Hao Liang, Ningyi Xu, Wenqiang Wang, Shaoshuai Shi, Xi Chen, Guangyu Sun, Wei Zhang, and Jason Cong. "**FPDNN**: An automated framework for mapping deep neural networks onto FPGAs with RTL-HLS hybrid templates."
6. Paolo Meloni, Daniela Loi, Gianfranco Deriu, Marco Carreras, Francesco Conti, Alessandro Capotondi, and Davide Rossi. "Exploring **NEURAGHE**: A Customizable Template for APSoCbased CNN Inference at the Edge." In: IEEE Embedded Systems Letters PP (Oct. 2019), pp. 1–1. doi: 10.1109/LES.2019. 2947312.
7. Thea Aarrestad, Vladimir Loncar, Nicolò Ghielmetti, Maurizio Pierini, Sioni Summers, Jennifer Ngadiuba, Christoffer Petersson, Hampus Linander, Yutaro Iiyama, Giuseppe Di Guglielmo, et al. "Fast convolutional neural networks on FPGAs with **hls4ml**." In: Machine Learning: Science and Technology 2.4 (2021), p. 045015.
8. Nicholas J Fraser, Yaman Umuroglu, Giulio Gambardella, Michaela Blott, Philip Leong, Magnus Jahre, and Kees Vissers. **FINN** "Scaling binarized neural networks on reconfigurable logic." In: Proceedings of the 8th Workshop and 6th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms. 2017, pp. 25–3
9. Deep learning Processor Unit (**DPU**) designed for the Zynq® UltraScale+™ MPSoC, DPUCZDX8G for Zynq UltraScale+ MPSoCs Product Guide (PG338). https://docs.amd.com/r/en-US/pg338-dpu?tocId=3xsG16y_QFTWvAJKHbisEw
10. **Mittal**, Sparsh. "A survey of techniques for approximate computing." ACM CSUR 48.4: 1-33 (2016).
11. **Hubara**, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks. *Advances in neural information processing systems*, *29*.
12. **Gholami**, Amir, et al. "A survey of quantization methods for efficient neural network inference." *arXiv preprint arXiv:2103.13630* (2021).
13. Pappalardo, A., Umuroglu, Y., Blott, M., Mitrevski, J., Hawks, B., Tran, N., ... & Duarte, J. (2022). **Qonnx**: Representing arbitrary-precision quantized neural networks. *arXiv preprint arXiv:2206.07527*.
14. **Sau**, Carlo, et al. "The Multi-Dataflow Composer tool: An open-source tool suite for optimized coarse-grain reconfigurable hardware accelerators and platform design." *Microprocessors and Microsystems* 80 (2021): 103326
15. Palumbo, Francesca, et al. "**MYRTUS**: Multi-layer 360° dYnamic orchestration and interopeRable design environmenT for compute-continuum", To appear, https://doi.org/10.1145/3637543.3654618
16. **Ratto**, F., Máinez, Á.P., Sau, C. et al. An Automated Design Flow for Adaptive Neural Network Hardware Accelerators. J Sign Process Syst 95, 1091–1113 (2023). https://doi.org/10.1007/s11265-023-01855-x
17. **Manca**, F., Ratto, F., Palumbo, F. ONNX-to-Harware Design Flow for Adaptive Neural-Network Inference on FPGAs, Proceedings of the XXIV SAMOS International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation, June 29 - July 4, 2024 (To appear)

MYRTUS

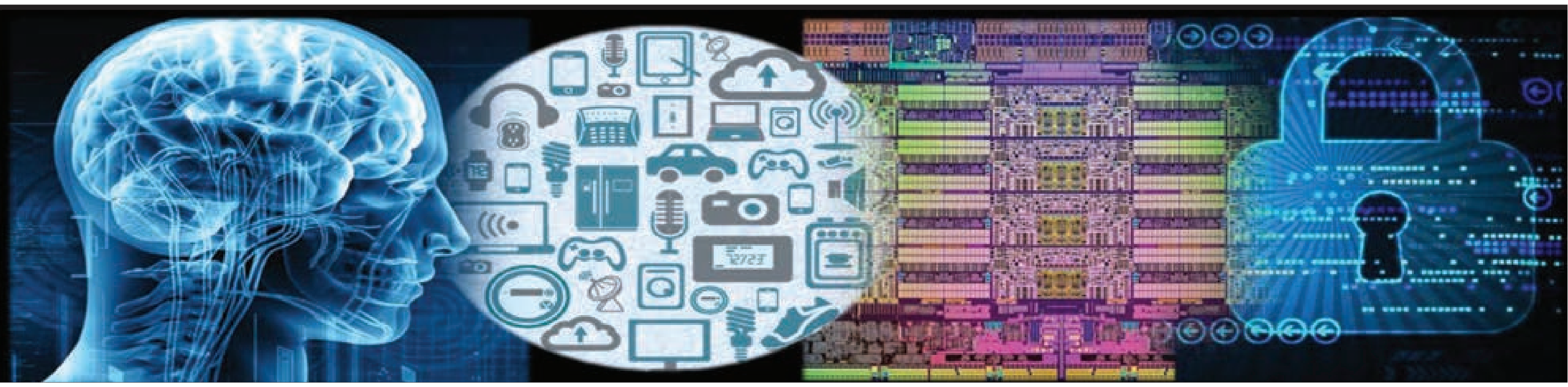
MYRTUS has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101135183.

HORIZON EUROPE

European Commission

Energy-Efficient and Robust Deep Learning for Autonomous Systems

Alberto Marchisio and Muhammad Shafique
eBrain Lab, Division of Engineering, New York University (NYU), Abu Dhabi, UAE



2

Who Ruled the World!

Age of Power

Man-Power (#), Skills, Strength, Courage, etc.

Age of Resources and Industry

Fuel, Industrial Tech., Economic Politics, etc.

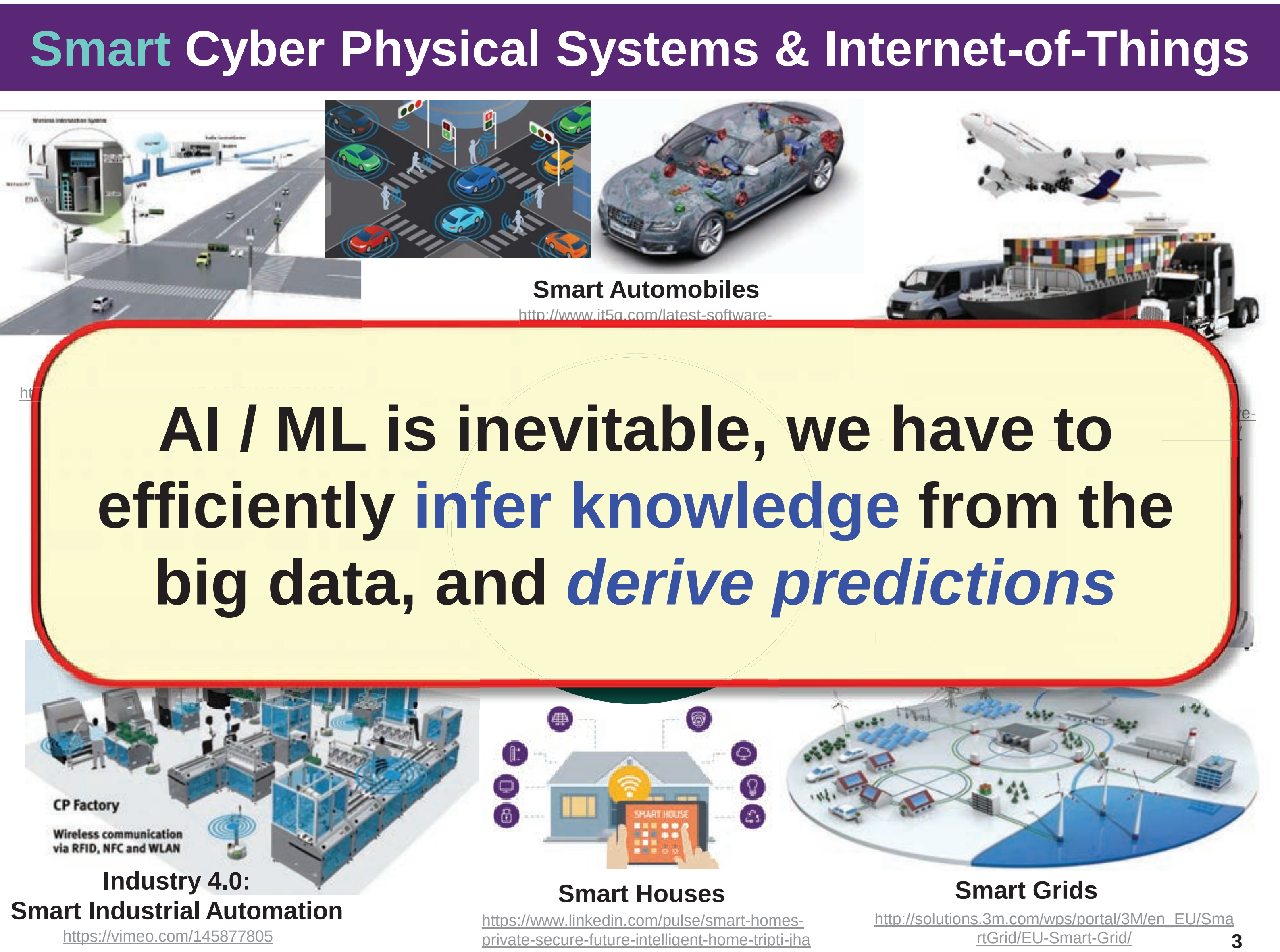
Age of Data and AI

Data is the New Fuel

Innovation in Technology is the New Politics

Nation-wide Race for Dominance in AI

2


Smart Cyber Physical Systems & Internet-of-Things
Smart Automobiles
AI / ML is inevitable, we have to efficiently infer knowledge from the big data, and derive predictions
CP Factory
Wireless communication via RFID, NFC and WLAN
Industry 4.0:
Smart Industrial Automation
https://vimeo.com/145877805
SMART HOUSE
Smart Houses
https://www.linkedin.com/pulse/smart-homes-private-secure-future-intelligent-home-tripti-jha
Smart Grids
http://solutions.3m.com/wps/portal/3M/en_EU/SmartGrid/EU-Smart-Grid/
3

Autonomous Mobile Agents with Neural Networks

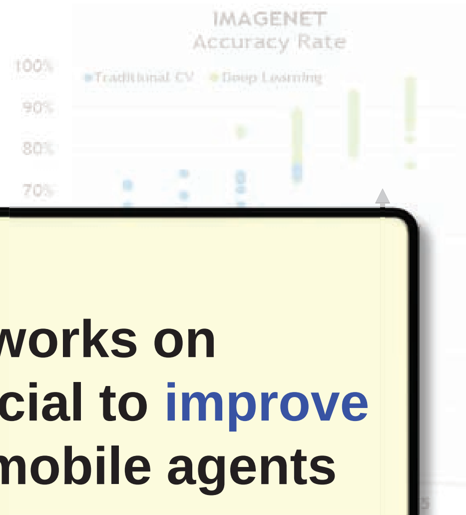
- Mobile Agents for Exploration

https:/

https://www.dlr.de/content/en/articles/news/2022/03/20220701_robotics-team-practises-lunar-exploration-on-mount-etna.html

https://www.mining-technology.com/features/autonomous-exploration-the-potential-for-drones-in-the-mining-industry/

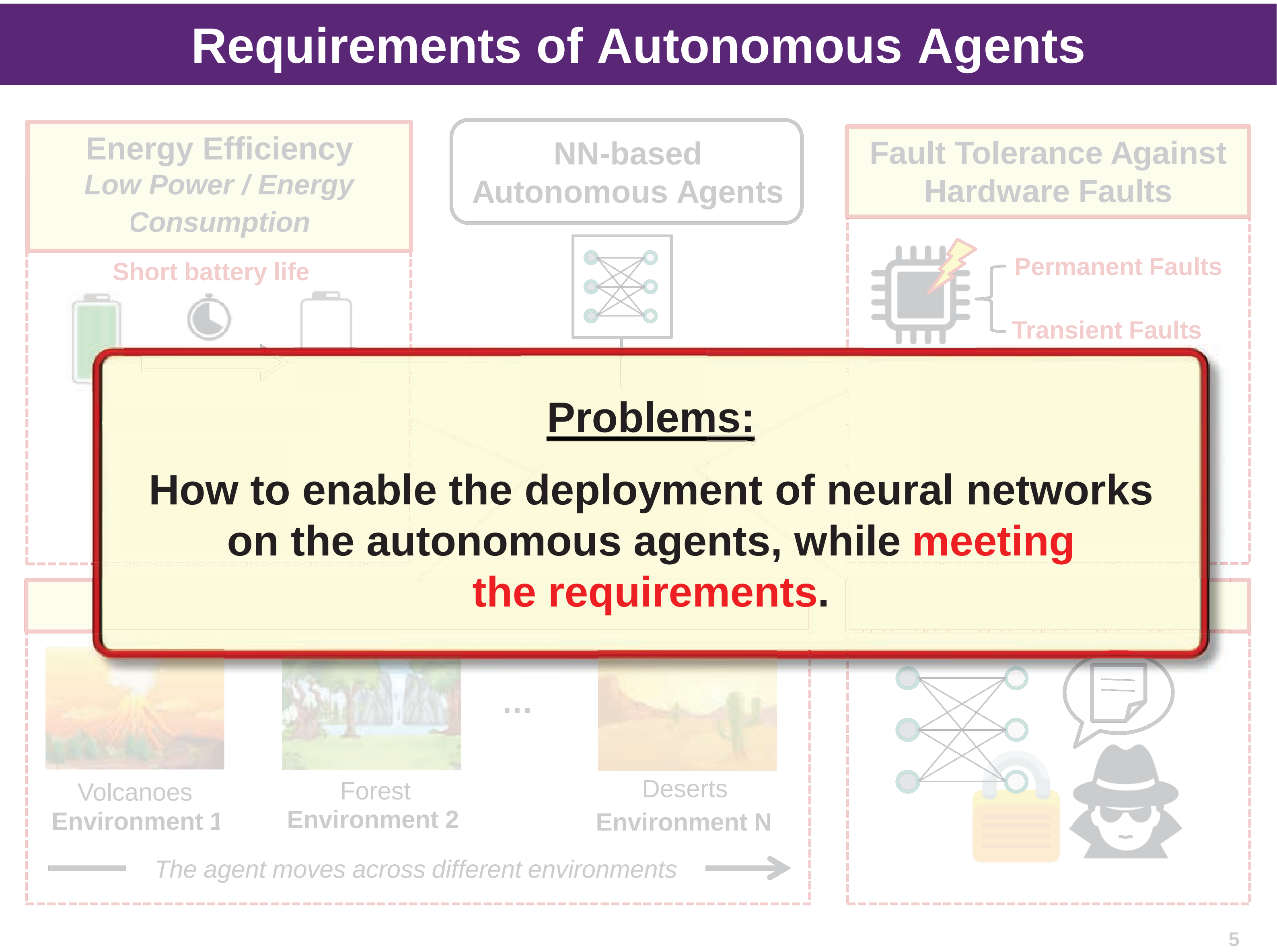
- Neural Networks



Neural Network (NN) algorithms achieved state-of-the-art accuracy for object recognition

The employment of neural networks on autonomous mobile agents is beneficial to improve the performance (accuracy) of the mobile agents

4

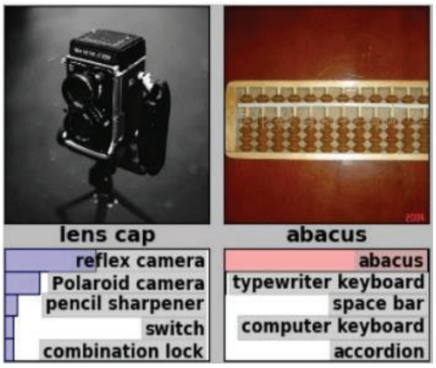

Requirements of Autonomous Agents
Energy Efficiency
Low Power / Energy Consumption
Short battery life
NN-based Autonomous Agents
Fault Tolerance Against Hardware Faults
Permanent Faults
Transient Faults
Problems:
How to enable the deployment of neural networks on the autonomous agents, while meeting the requirements.
Volcanoes
Environment 1
Forest
Environment 2
...
Deserts
Environment N
The agent moves across different environments
5

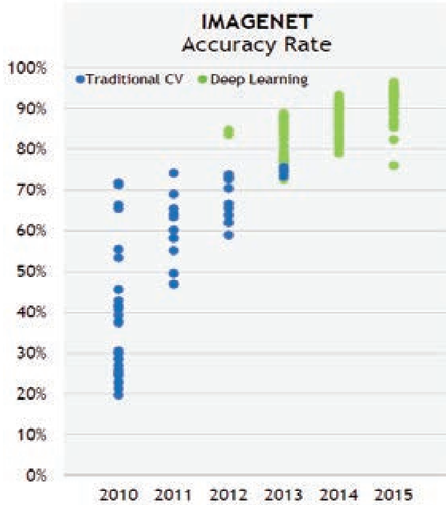
6

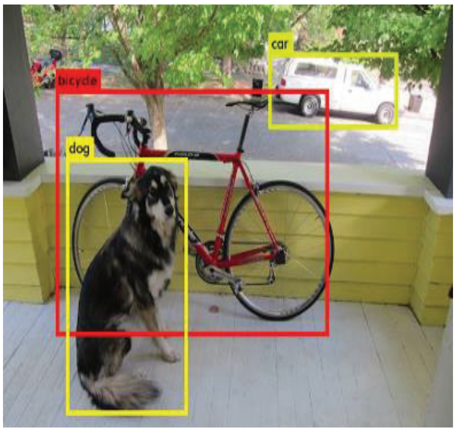
AI / ML Applications => require High Efficiency Gains

Autonomous Driving

Image Classification







Object Detection & Localization

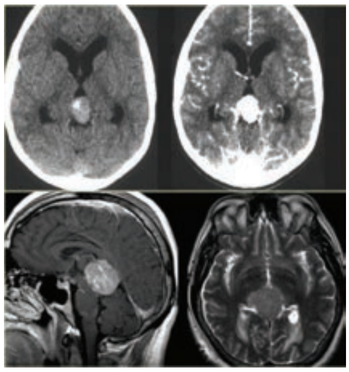
Machine Translation

Strategy Games

Natural Language Processing

Google ASSISTANT

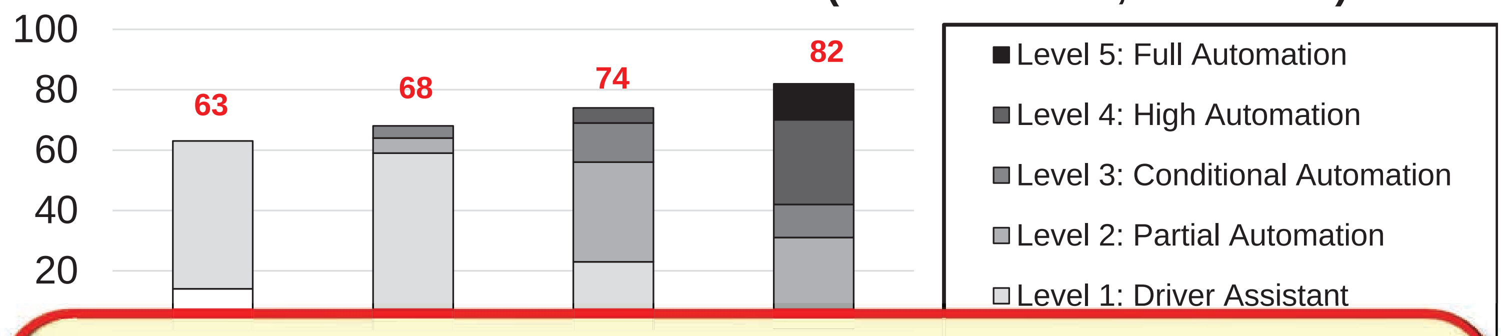
Forex/Stocks Trading



Cancer Detection

6

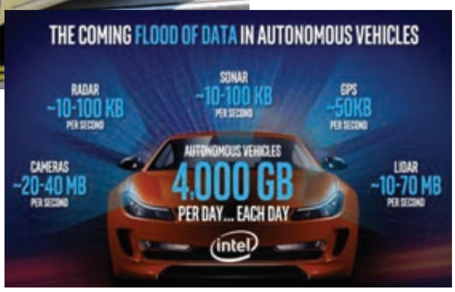
Autonomous Cars: The Big Data Processing Challenge!



Problem

AI on Big Data@Edge => Complexity^2

- Radar: ~10-100KB/sec
- Sonar: ~10-100KB/sec
- Camera: ~20-40MB/sec
- GPS: ~50KB/sec



4000 GB per day

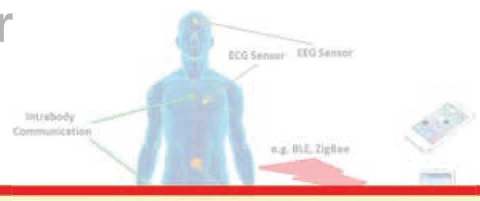
Sources: https://www.networkworld.com/article/3147892/one-autonomous-car-will-use-4000-gb-of-dataday.html

7

Smart CPS & IoT => The Robustness Challenge!

… should consider

- ❑ Robustness
  - ❑ Reliability



Challenging Question

How to process such huge amount of data in power/energy efficient way, while providing robustness?

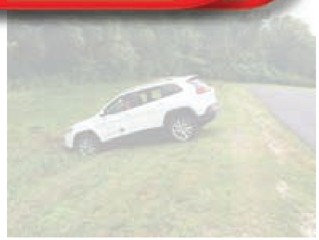
  - ❑ Privacy
  - ❑ Interoperability

Hacking Jeep Cherokee 4x4 (2015)

Sent the instructions through Entertainment systems

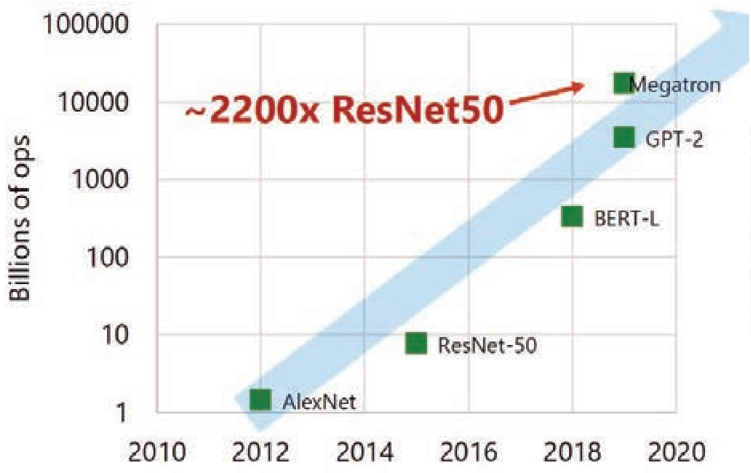
- Change the in-car temperature
- Control the steering
- Control the braking system

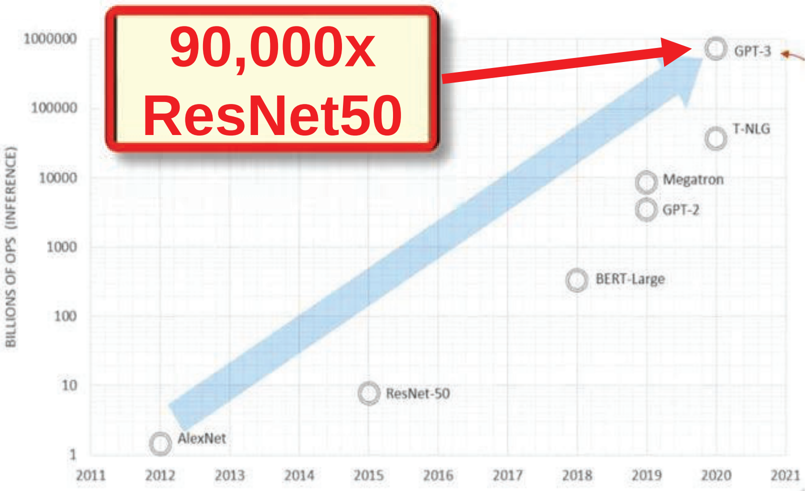
https://www.ophtek.com/4-real-life-examples-iot-hacked/
https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/



8

Complexity: Exponential Growth in Model Sizes!





Source: Eric Chung, "Accelerating Microsoft's AI Ambitions", Microsoft, Azure AI and Advanced Architectures Group, 2019.
Source: https://www.microsoft.com/en-us/research/blog/a-microsoft-custom-data-type-for-efficient-inference/.

Megatron is a **8.3 billion parameter transformer** language model with trained on **512 V100 GPUs**, making it the largest transformer model ever!

9

Google TPU-v3 vs. Nvidia's DGX Supercomputers

Google TPU-v3 supercomputer

288 kW of power

(https://www.nextplatform.com/2018/05/10/tearing-apart-googles-tpu-3-0-ai-coprocessor/)

Nvidia's Selene supercomputer (DGX-SuperPod)

1125 kW of power

(https://developer.nvidia.com/blog/dgx-superpod-world-record-supercomputing-enterprise/)

Figure sources:
https://www.eetimes.com/nvidia-google-both-claim-mlperf-training-crown/#

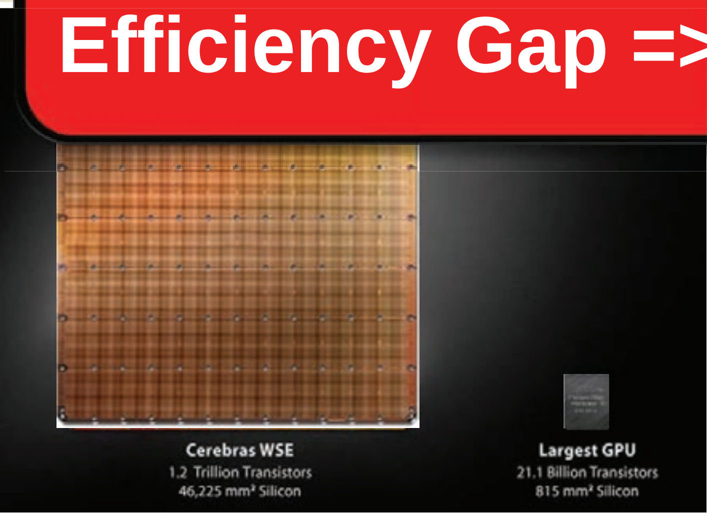
10

Today's ML Training Chip?

Cerebras 2nd Generation Wafer Scale Engine

Human Brain => 20W

Efficiency Gap => 1,000x → 100,000x!!!



push to the chip through 12x 4 kW power supplies

Figure sources:

1. https://www.anandtech.com/show/16000/342-transistors-for-every-person-in-the-world-cerebras-2nd-gen-wafer-scale-engine-teased
2. https://www.cerebras.net/

11

Robustness for Machine Learning: News Feed

Beware: Galaxy S10's Facial Recognition Easily Fooled with a Photo

Jesus Diaz - Freelance Writer
Updated Mar 11, 2019

Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian

Tempe police said car was in autonomous mode at the time of the crash and that the vehicle hit a woman who later died at a hospital

Hackers trick a Tesla into veering into the wrong lane

https://www.youtube.com/watch?v=a7L51u23YoM

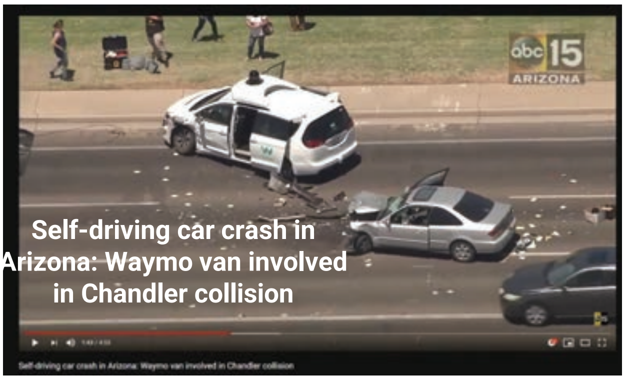
Tesla Model 3: Autopilot engaged during fatal crash

BBC

17 May 2019

Tesla driver dies in first fatal crash while using autopilot mode

The autopilot sensors on the Model S failed to distinguish a white tractor-trailer crossing the highway against a bright sky



https://www.technologyreview.com/f/613254/hackers-trick-teslas-autopilot-into-veering-towards-oncoming-traffic/

12

Adversarial Attacks on Tesla Autopilot by Tencent Keen Security Lab

Digital Adversarial Examples

❑ Insert the noise into the DNN input

Rainy Score: 0.0113

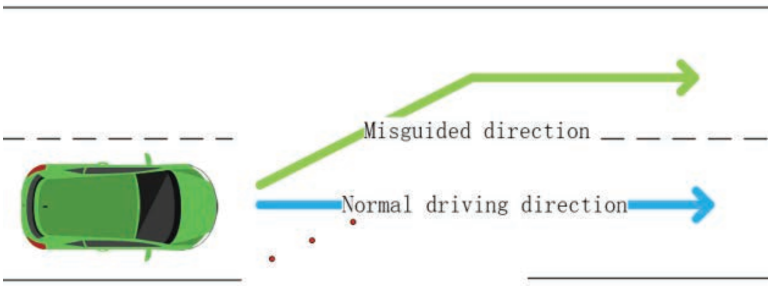
Adversarial Noise

Rainy score: 0.8204

Black-Box Attack

Physical World Adversarial Examples

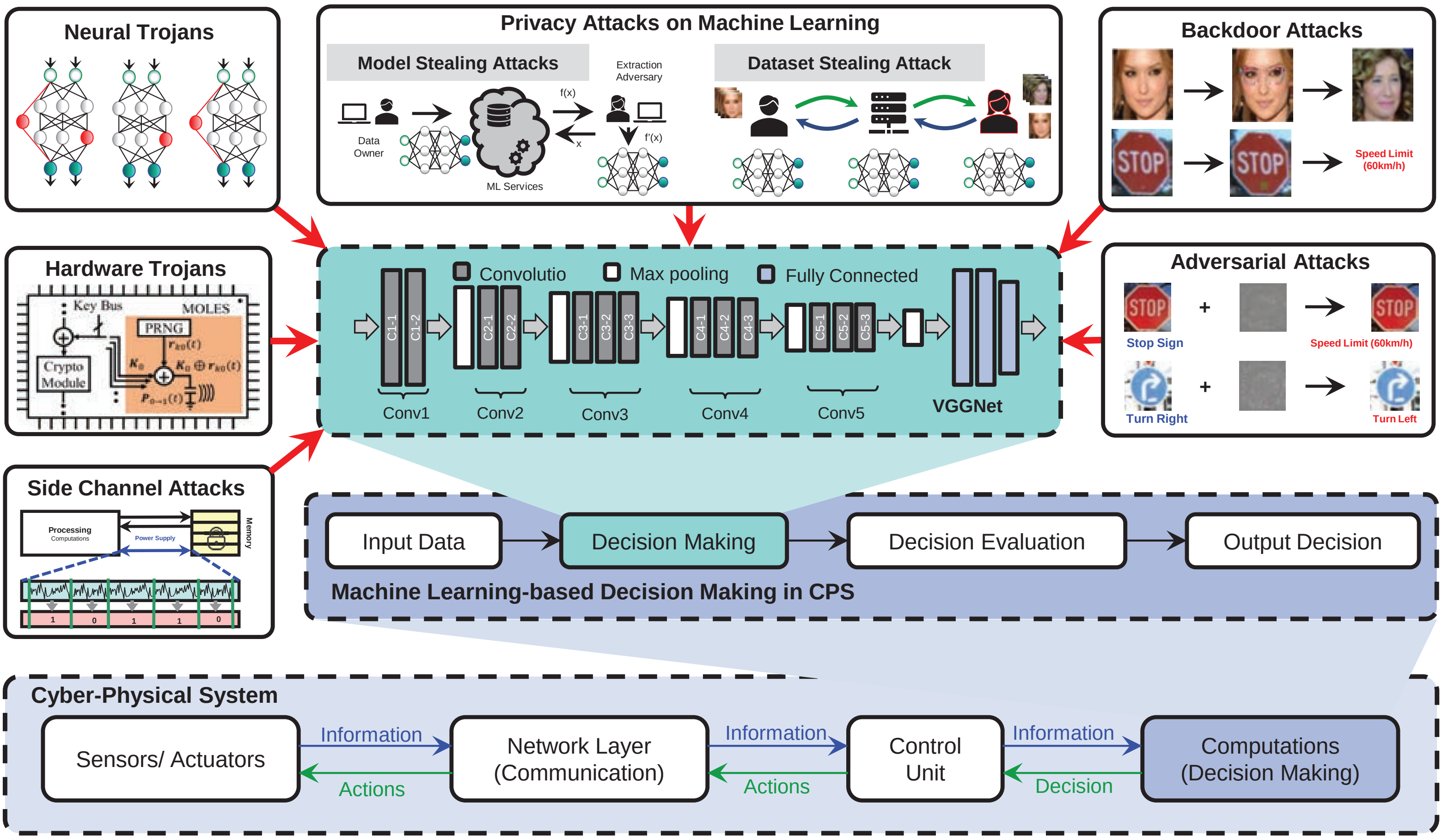
❑ Place the small stickers on the ground



Tencent Keen Security Lab, "Experimental Security Research of Tesla Autopilot" Technical Report 2019-03

13

Security Vulnerabilities in Machine Learning



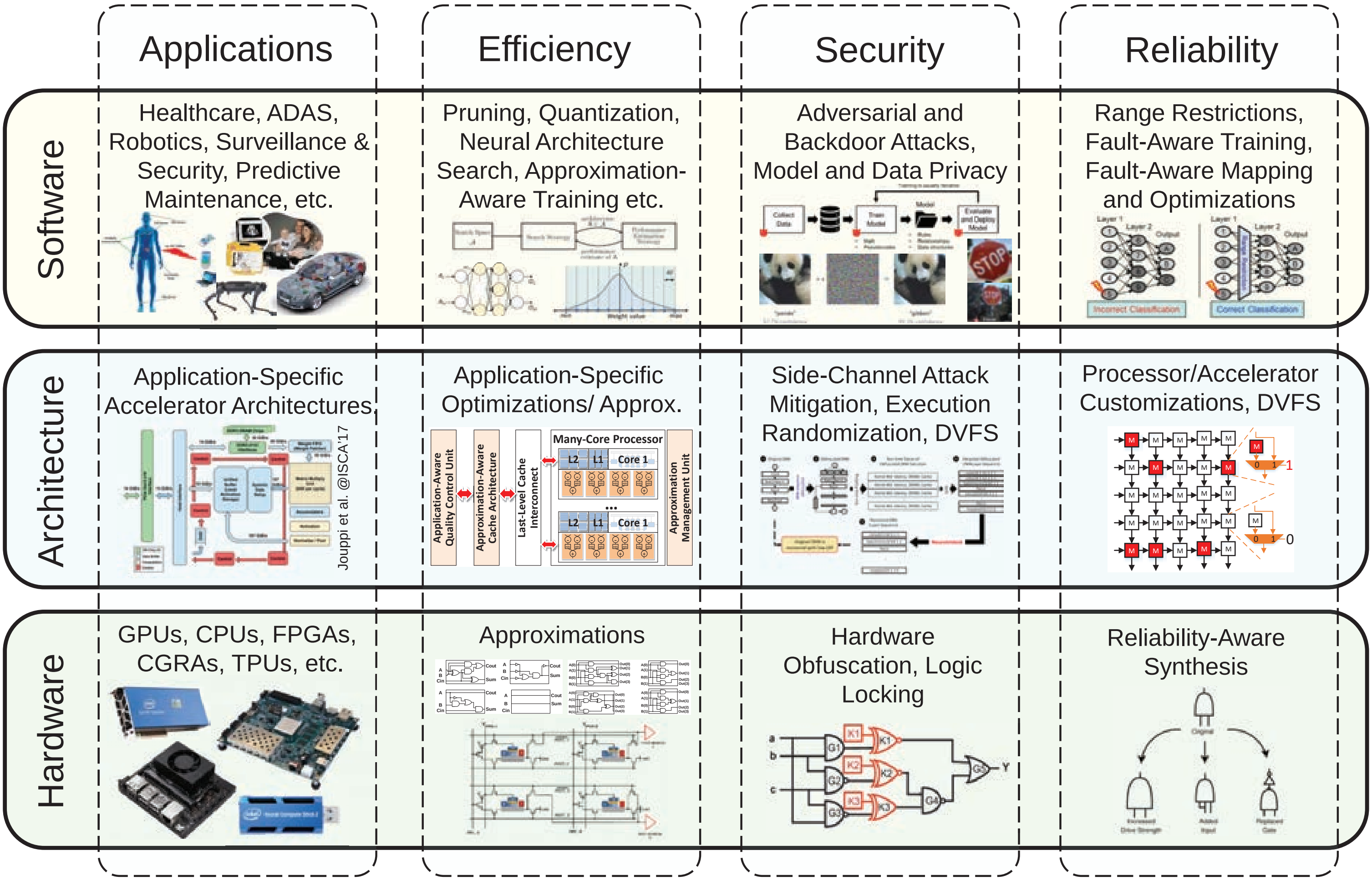
- M. A. Hanif, F. Khalid, R. V. W. Putra, S. Rehman, M. Shafique, "Robust Machine Learning Systems: Reliability and Security for Deep Neural Networks", in IOLTS-2018, Platja d'Aro, Spain, pp. 257 - 260.
- F. Kriebel, S. Rehman, M. A. Hanif, F. Khalid, M. Shafique, "Robustness for Smart Cyber-Physical Systems and Internet-of-Things: From Adaptive Robustness Methods to Reliability and Security for Machine Learning", ISVLSI-2018, Hong Kong, China, pp. 581-586.

14

Research Overview @ eBRAIN Lab

15

Overview



Applications
Efficiency
Security
Reliability
Software
Healthcare, ADAS, Robotics, Surveillance & Security, Predictive Maintenance, etc.
Pruning, Quantization, Neural Architecture Search, Approximation-Aware Training etc.
Adversarial and Backdoor Attacks, Model and Data Privacy
Range Restrictions, Fault-Aware Training, Fault-Aware Mapping and Optimizations
Incorrect Classification
Correct Classification
Architecture
Application-Specific Accelerator Architectures
Jouppi et al. @ISCA'17
Application-Specific Optimizations/ Approx.
Application-Aware Quality Control Unit
Approximation-Aware Cache Architecture
Last-Level Cache Interconnect
Many-Core Processor
L2
L1
Core 1
Approximation Management Unit
Side-Channel Attack Mitigation, Execution Randomization, DVFS
Processor/Accelerator Customizations, DVFS
Hardware
GPUs, CPUs, FPGAs, CGRAs, TPUs, etc.
Approximations
Hardware Obfuscation, Logic Locking
Reliability-Aware Synthesis

16

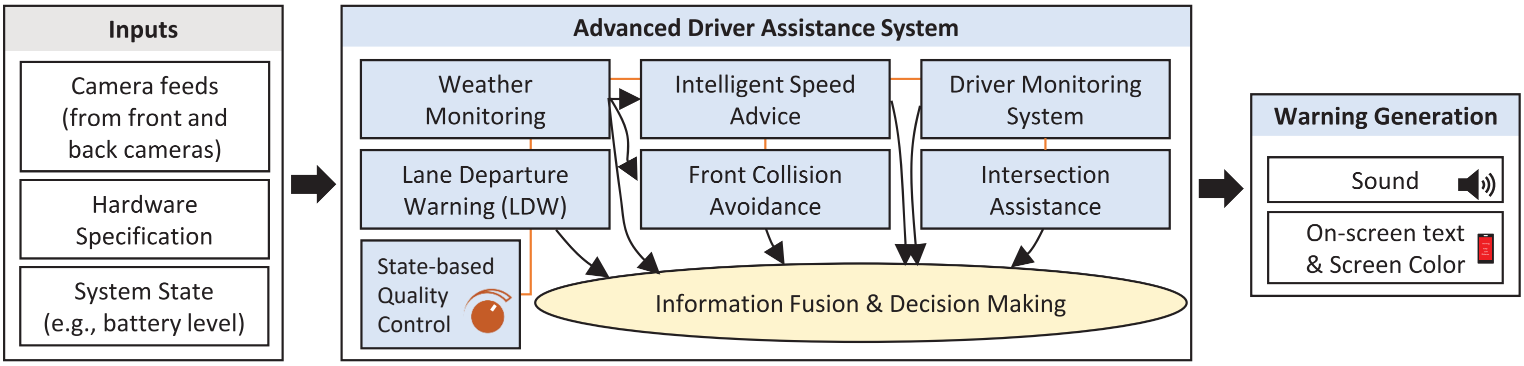
Advanced Driver Assistance System

17

ADAS Features and System Overview

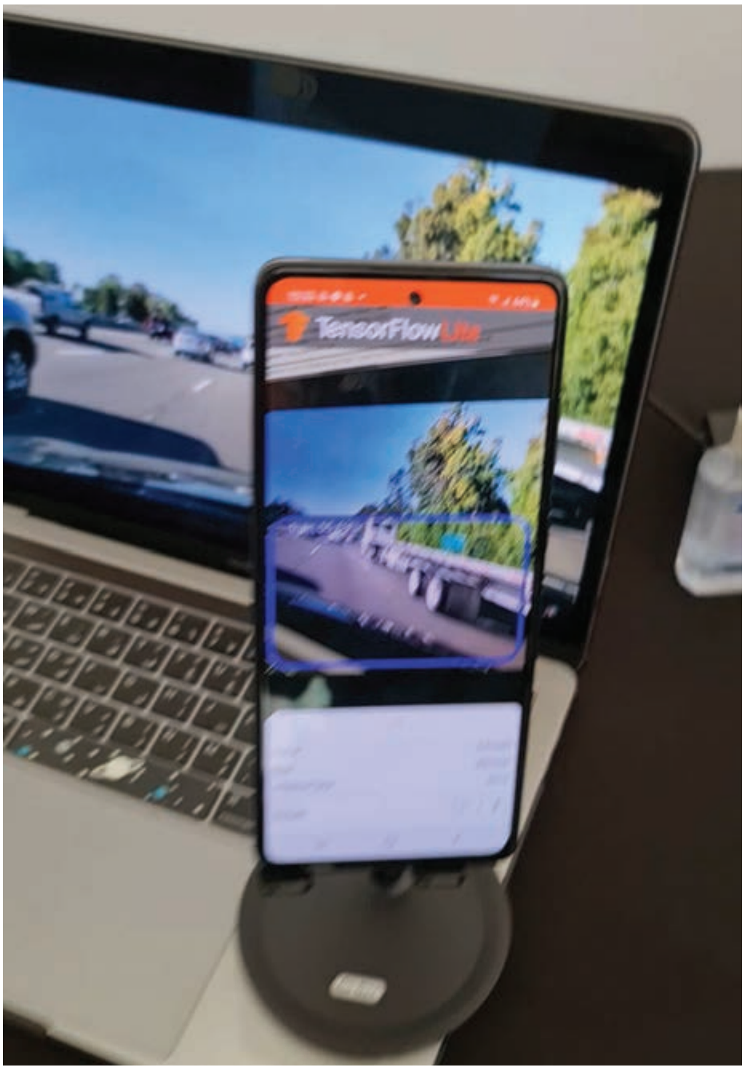
❑ Features

- ❑ Weather Monitoring
- ❑ Speed Advice
- ❑ Lane Departure Warning (LWD)
- ❑ Front Collision Avoidance
- ❑ Driver Monitoring System
- ❑ Intersection Assistance



18

Advanced Driver Assistance System for Mobile Devices



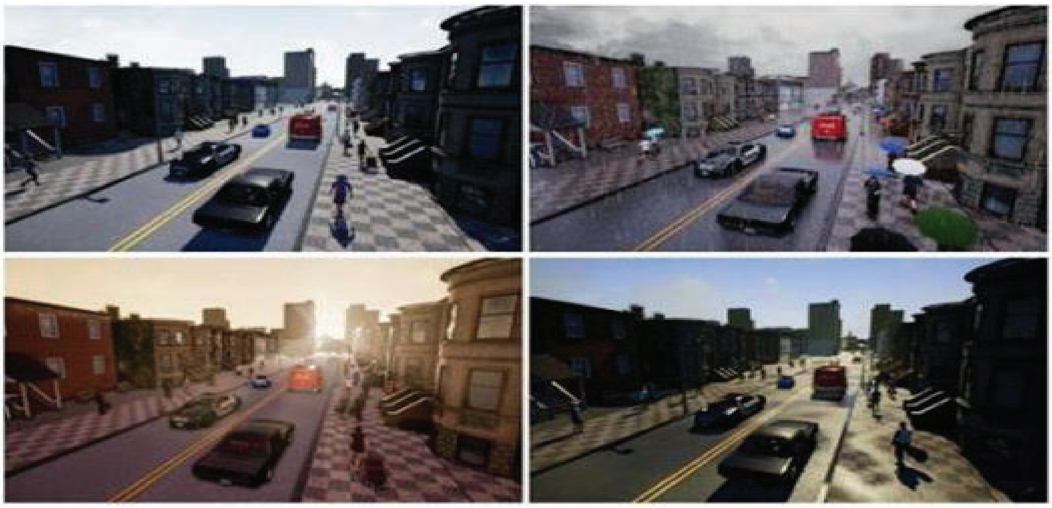
19 19

CARLA Simulator

20

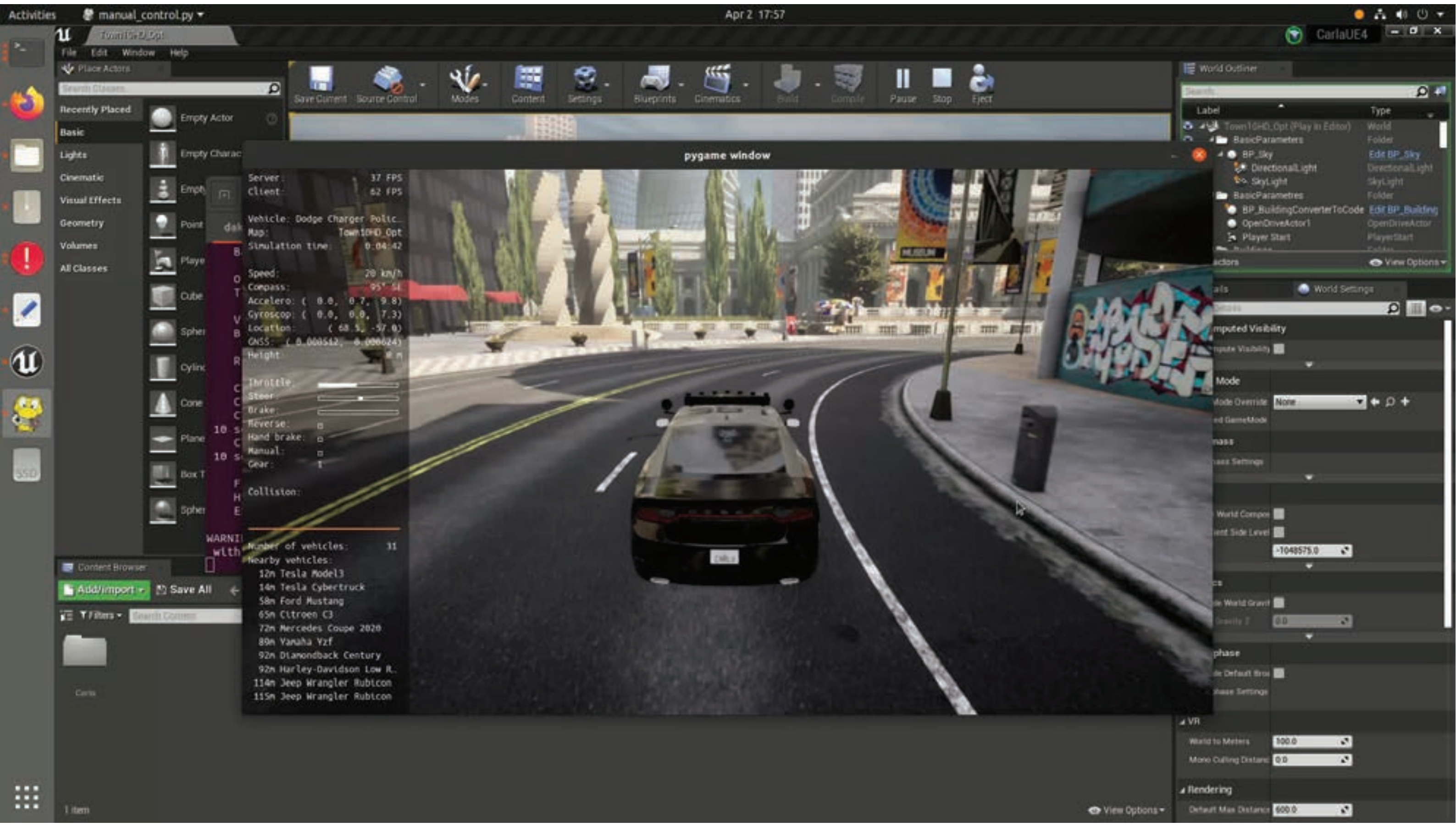
What is CARLA?

- Open-source simulator for autonomous driving research
- Consists of a scalable client-server architecture.
- Provides a realistic simulation environment for testing and validating autonomous driving algorithms
- Features realistic graphics, dynamic weather conditions, and configurable sensor suites
- 4 types of usable sensors:
  - Depth Camera
  - LIDAR Sensor
  - RGB Camera
  - RSS Sensor



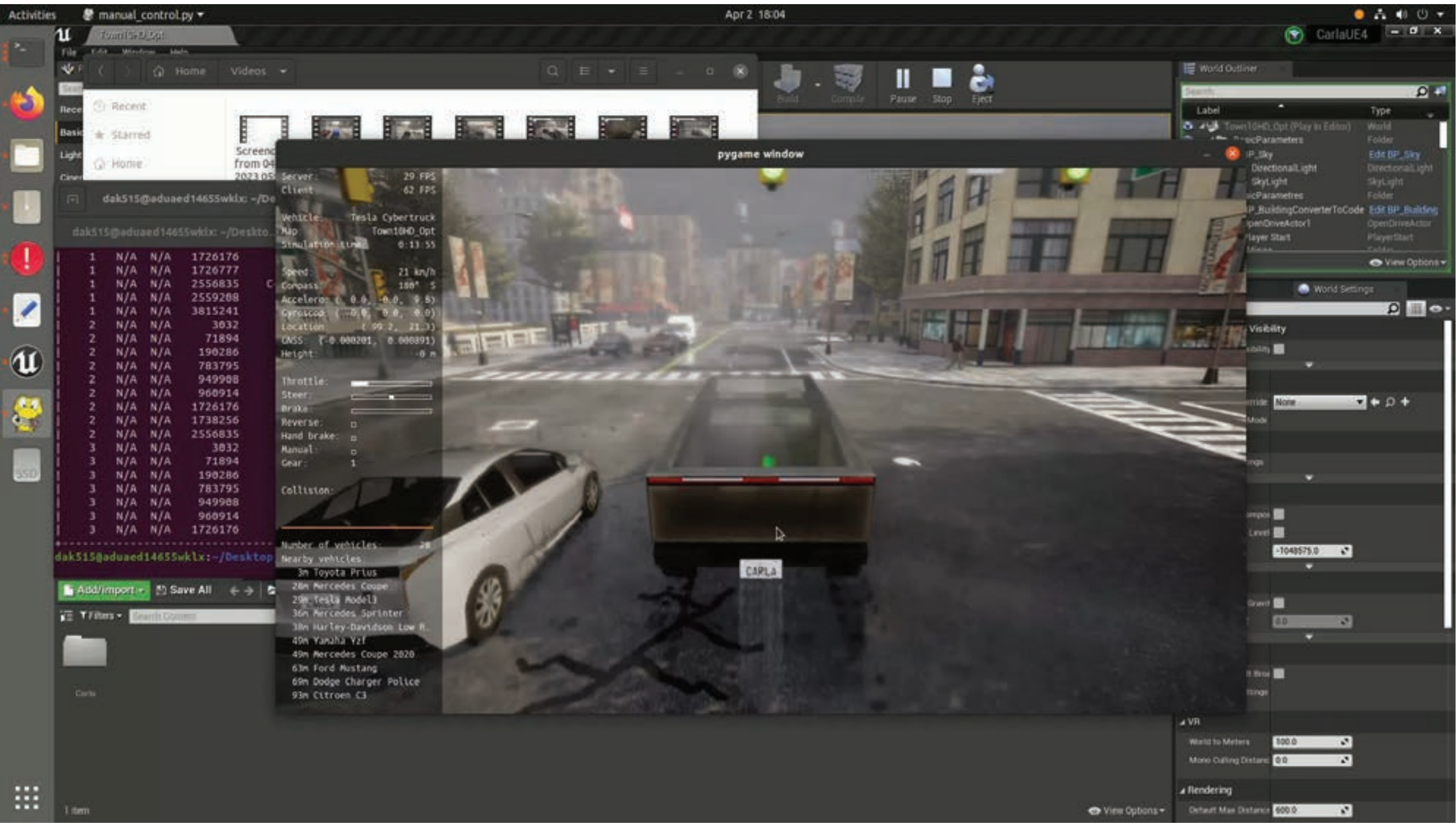
21

CARLA Simulation



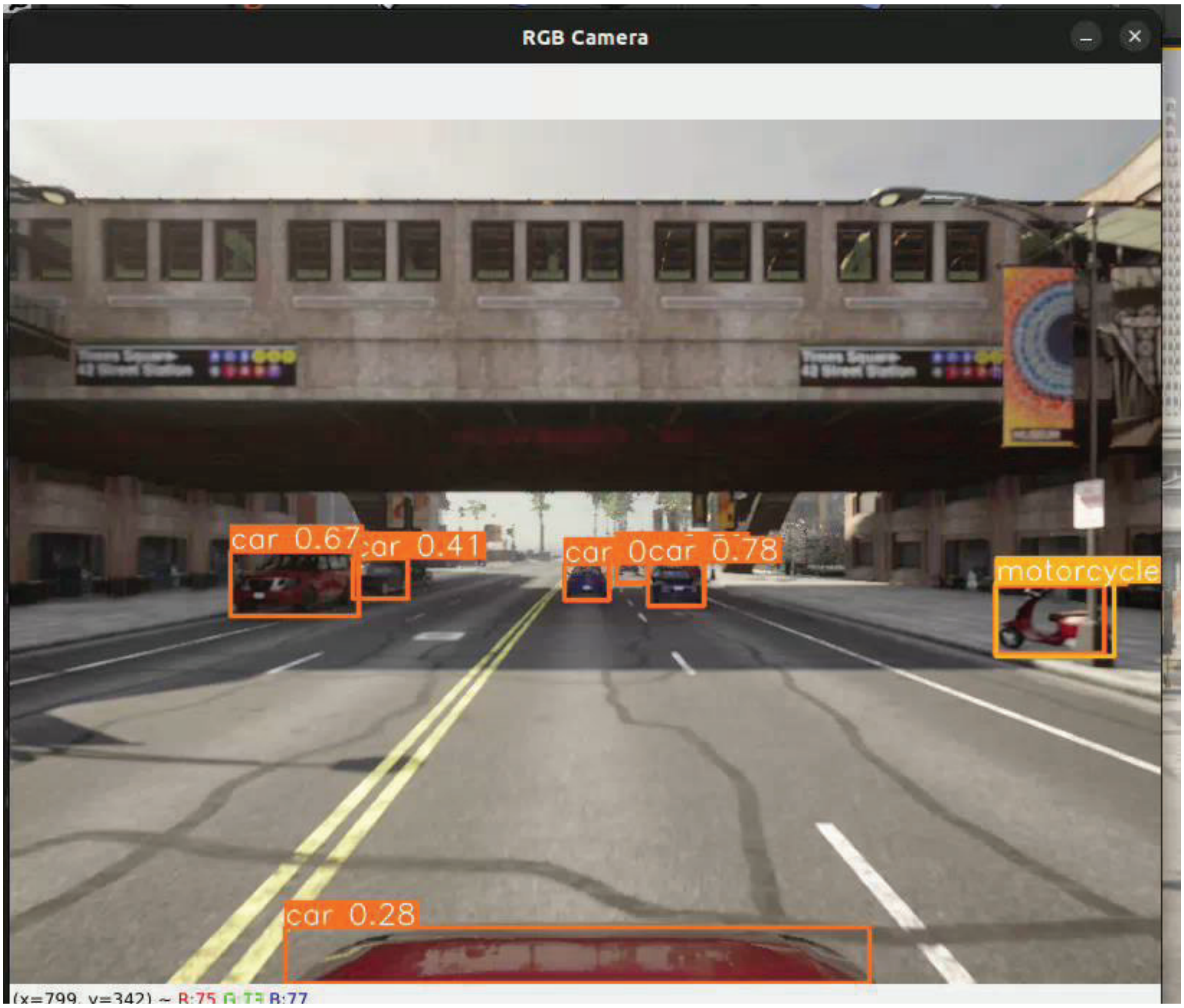
22

CARLA Simulation



23

Object Detection



24

Android Application & Cityscapes Demo

- Deployed YOLOv8 Android Application on Samsung S23 and A70
- Using PyTorch Android and Camera APIs
- Cemented the development pipeline to develop ML modules in eager mode, to deployment in the application
- Next steps include latency and energy efficiency evaluations on the edge

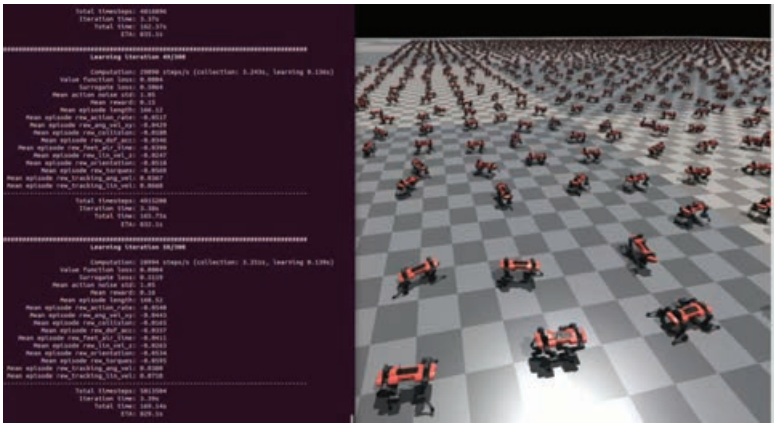
25

Robots and SLAM

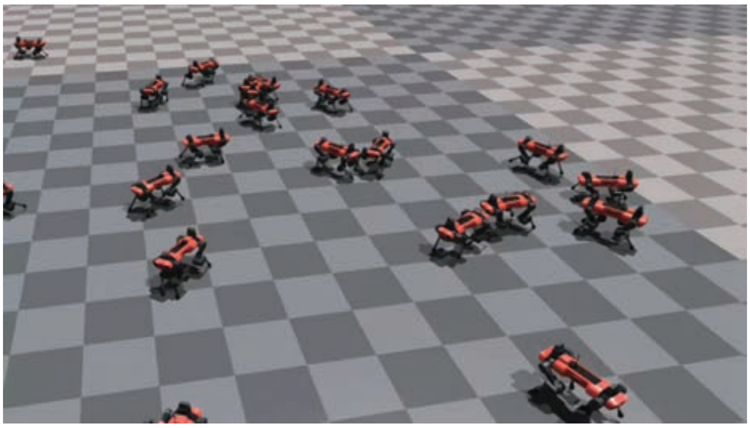
26

Training Legged Robots

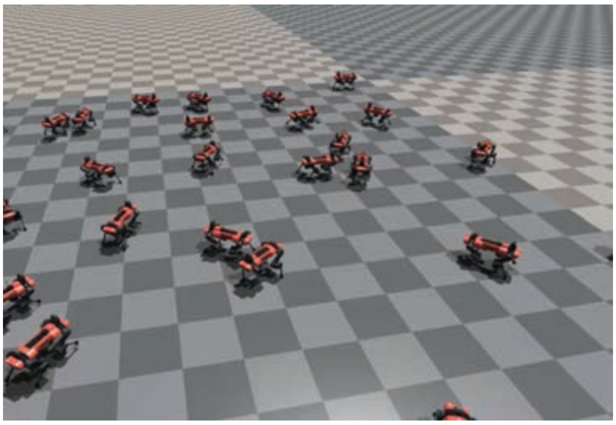
Training Video



Test Proximal Policy Optimization (PPO)

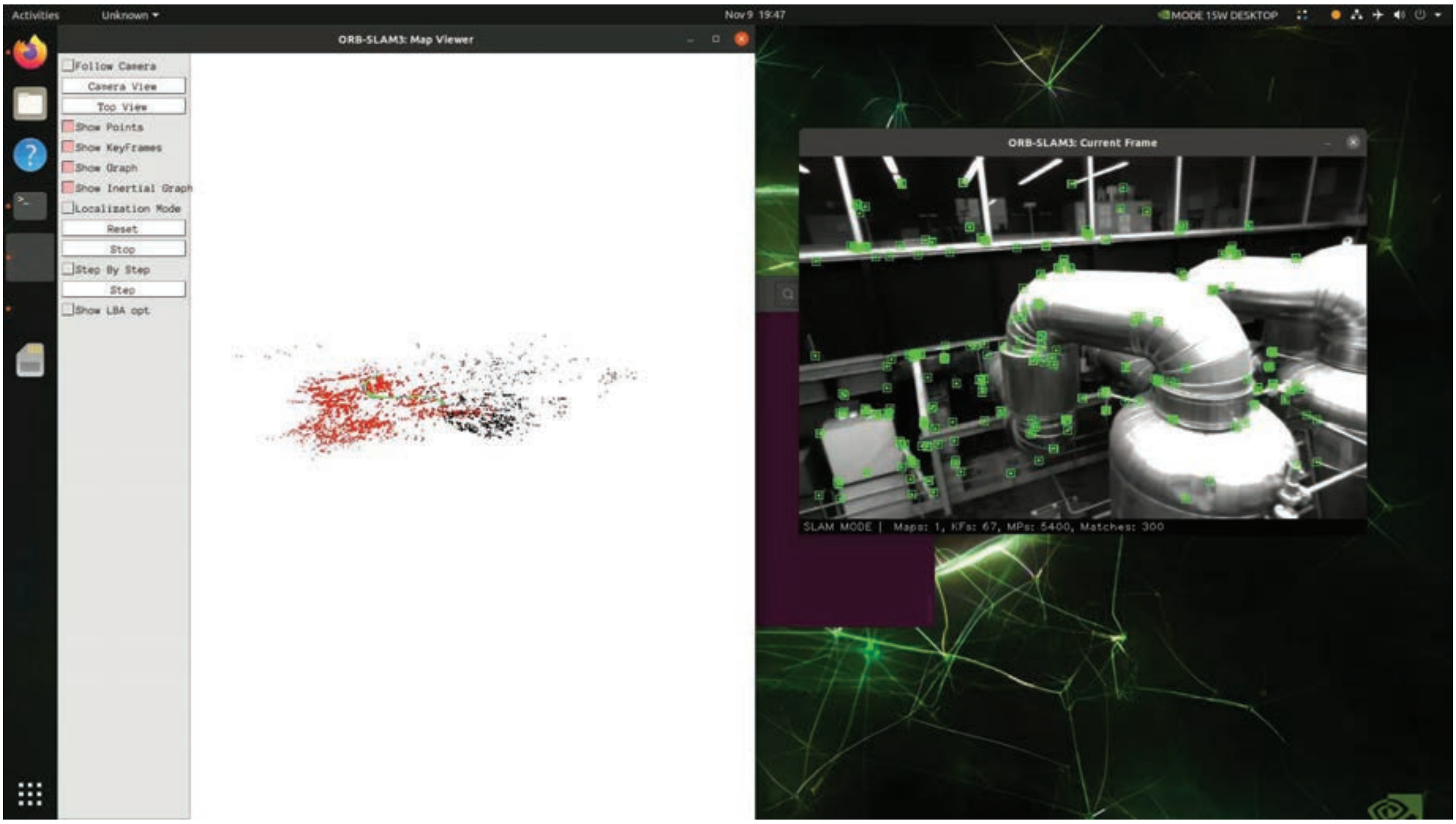


Test Curiosity



27

ORB-SLAM3 Xavier AGX Video Inference



28

Demo: Neural Slam Using Husky Navigation Pro

29

29

Embedded ML/AI for Spot (Boston Dynamics)

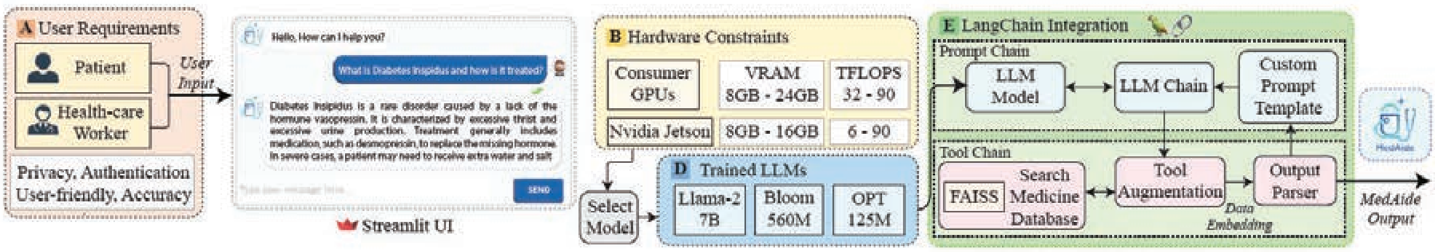
30

30

Medical

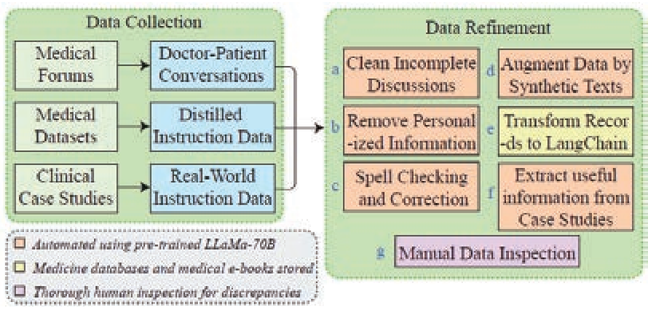
31

MedAide Research

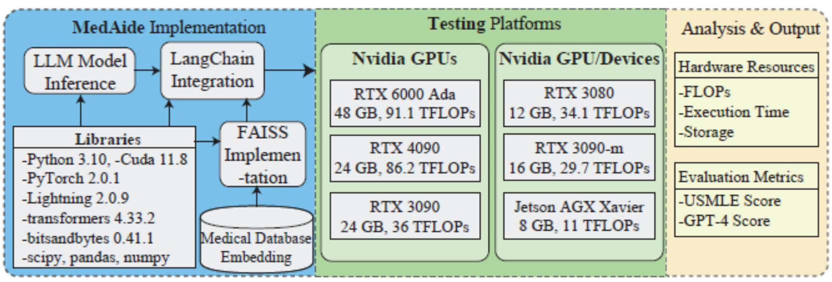


MedAide System Overview

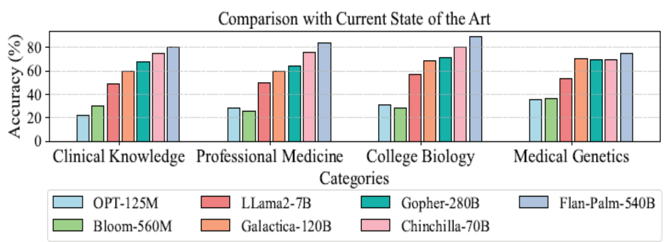
Demo



Dataset Generation



Experimental Setup



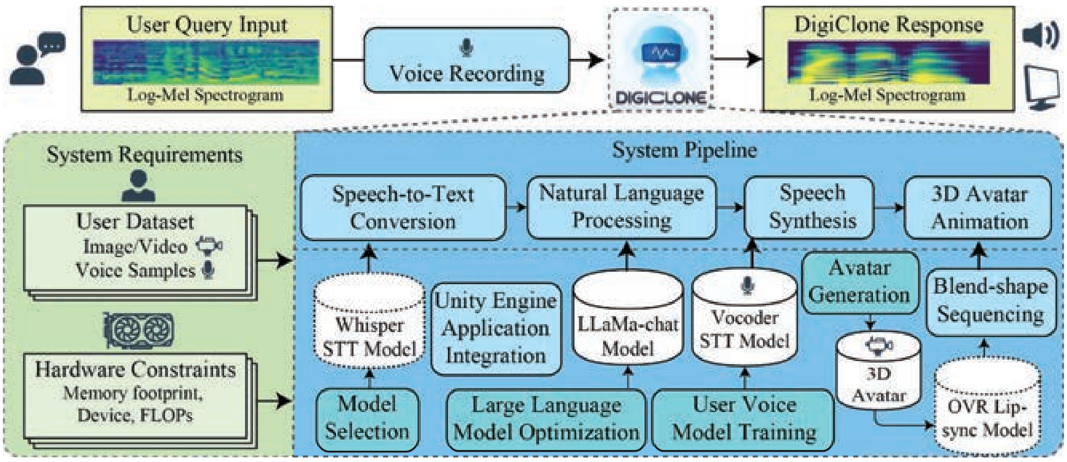
Accuracy Comparison

32

DigiClone Research

Demo

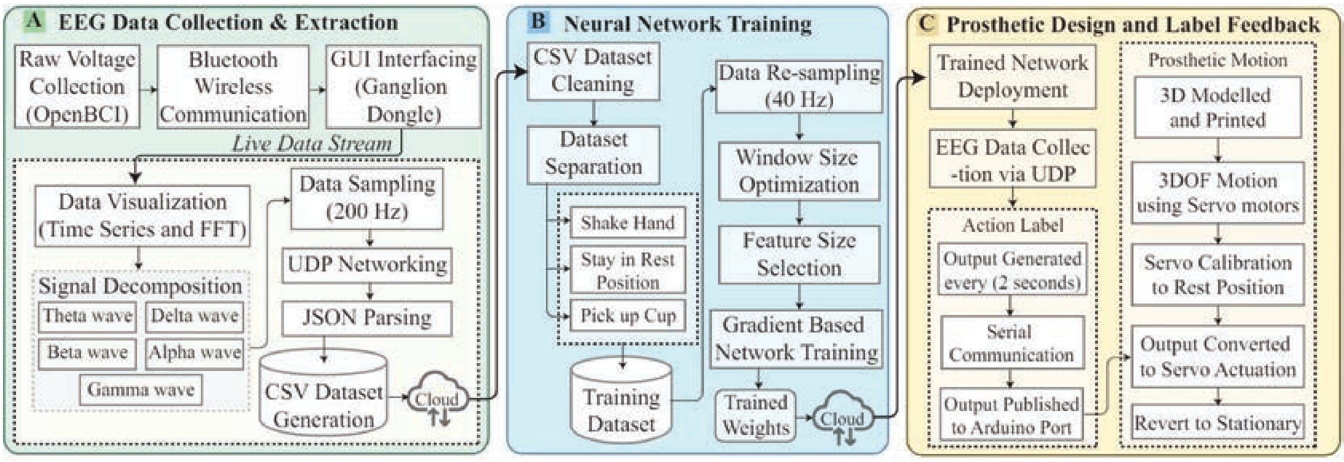
Motivation
Nvidia Omniverse



DigiClone System Overview

33

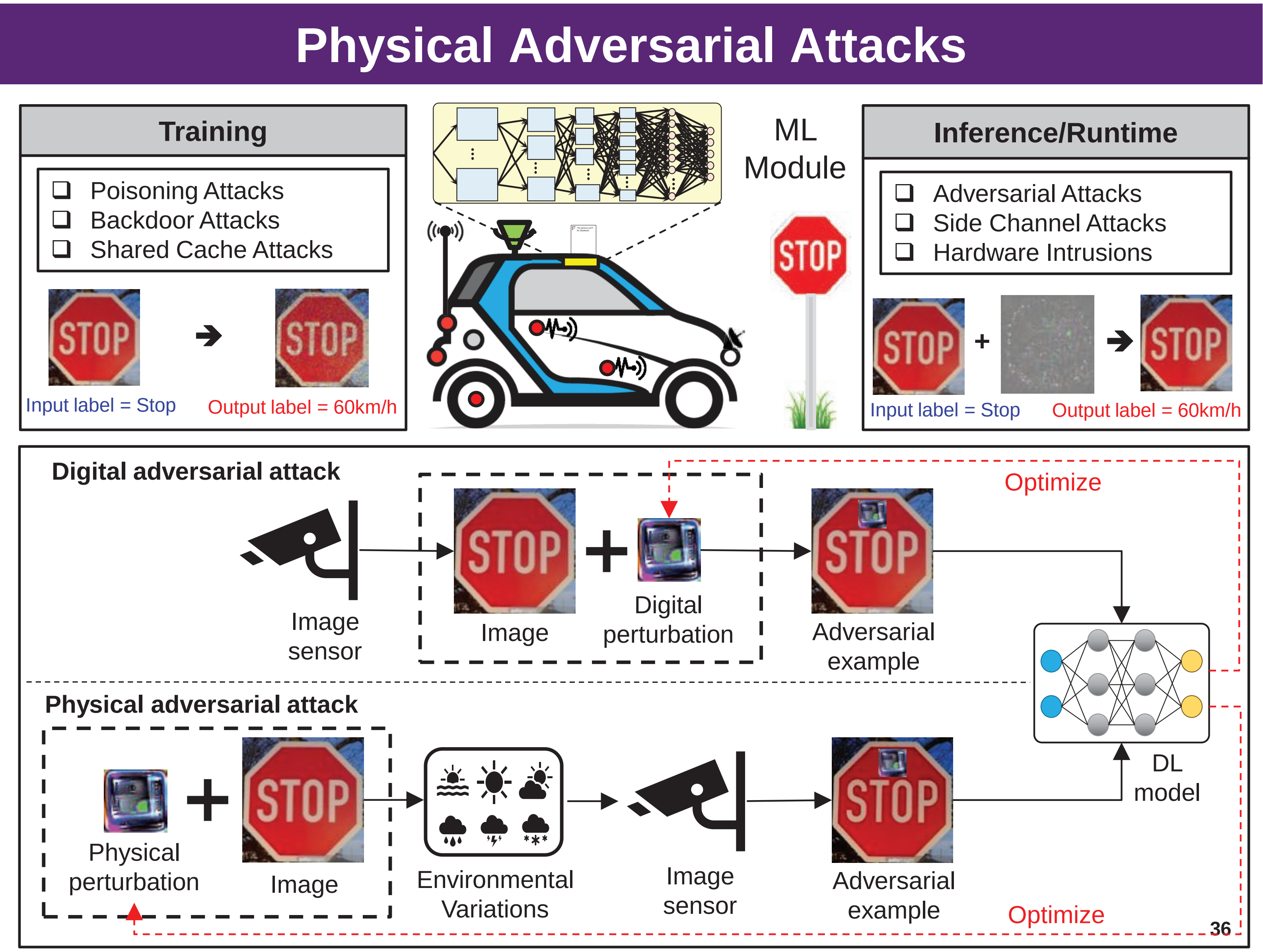
MindArm Research



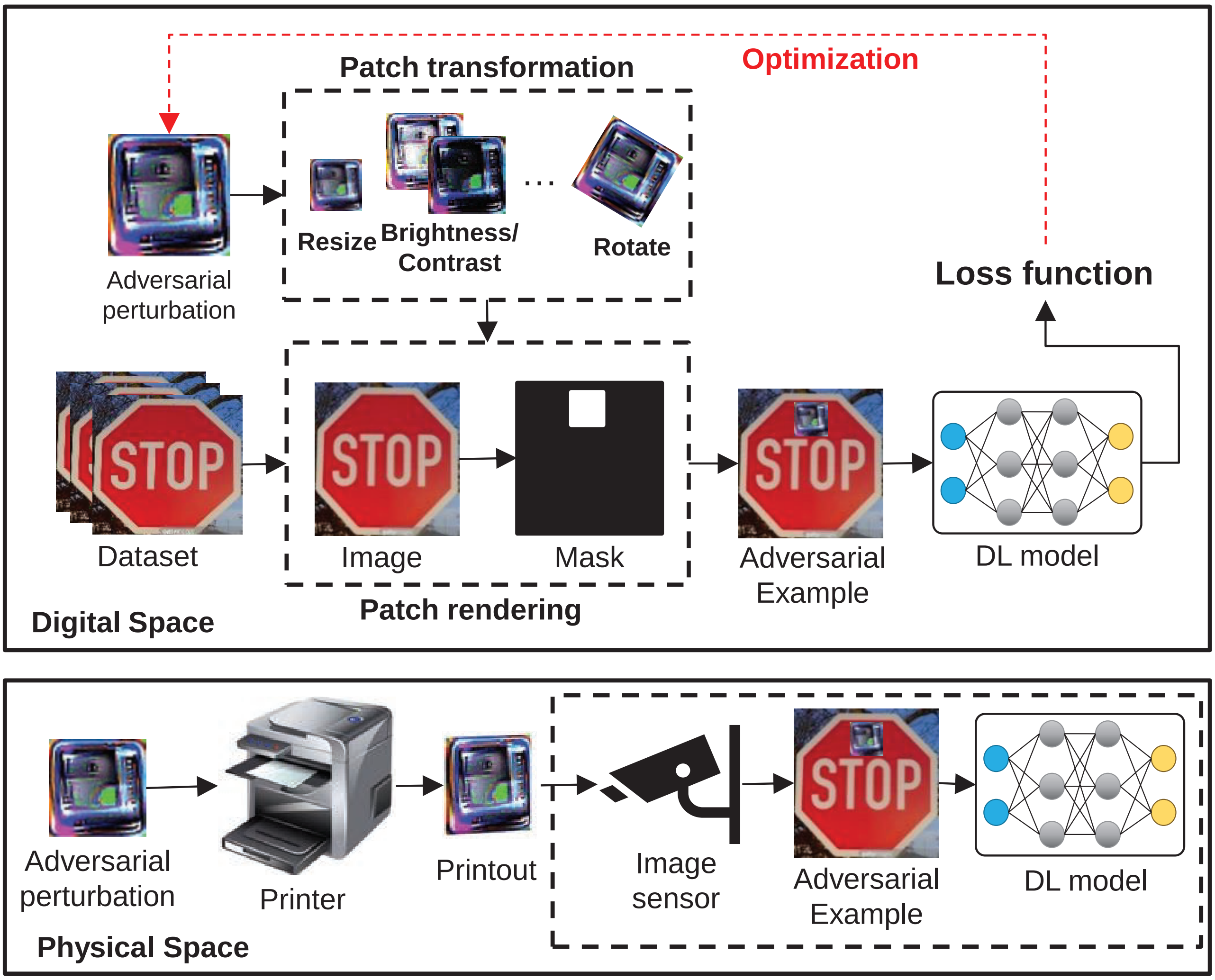
34

Physical Adversarial Attacks

35


Physical Adversarial Attacks
Training
Poisoning Attacks
Backdoor Attacks
Shared Cache Attacks
Input label = Stop
Output label = 60km/h
ML Module
Inference/Runtime
Adversarial Attacks
Side Channel Attacks
Hardware Intrusions
Input label = Stop
Output label = 60km/h
Digital adversarial attack
Image sensor
Image
Digital perturbation
Adversarial example
Optimize
DL model
Physical adversarial attack
Physical perturbation
Image
Environmental Variations
Image sensor
Adversarial example
Optimize
36

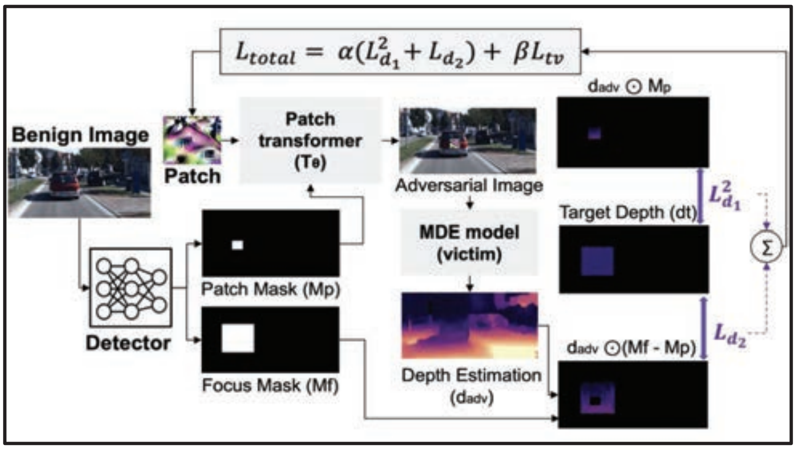
Physial Adversarial Attacks



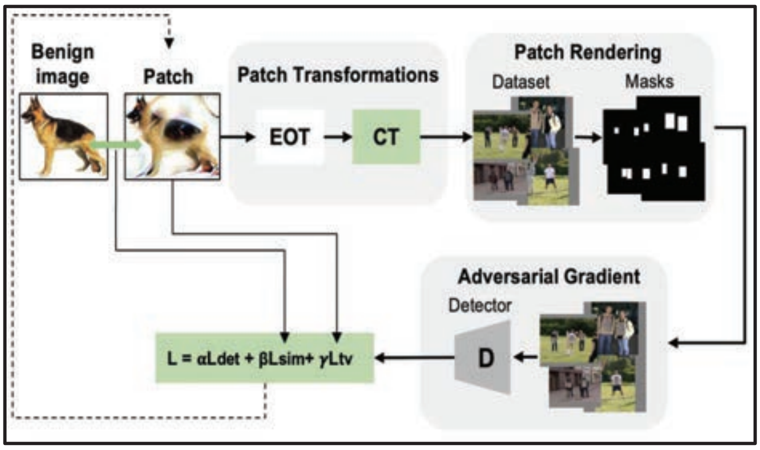
37

Adversarial Attacks and Defenses

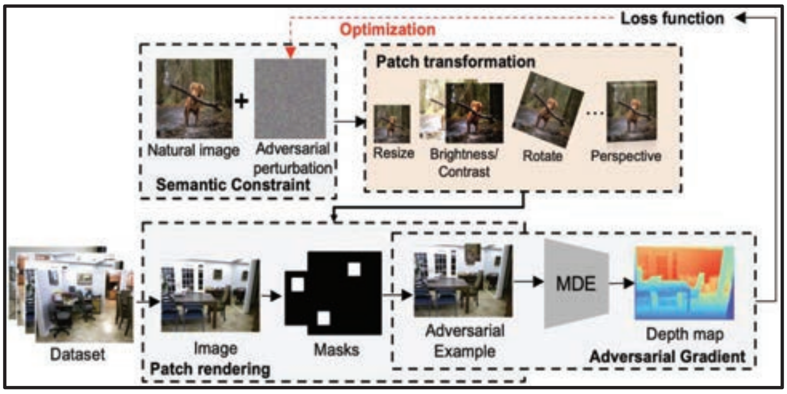
Adversarial Attacks on MDE



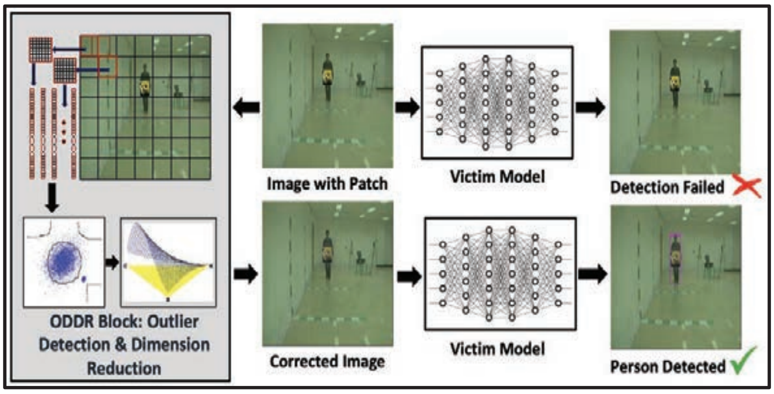
Adversarial Attacks on Person Detection



Adversarial Attacks on MDE

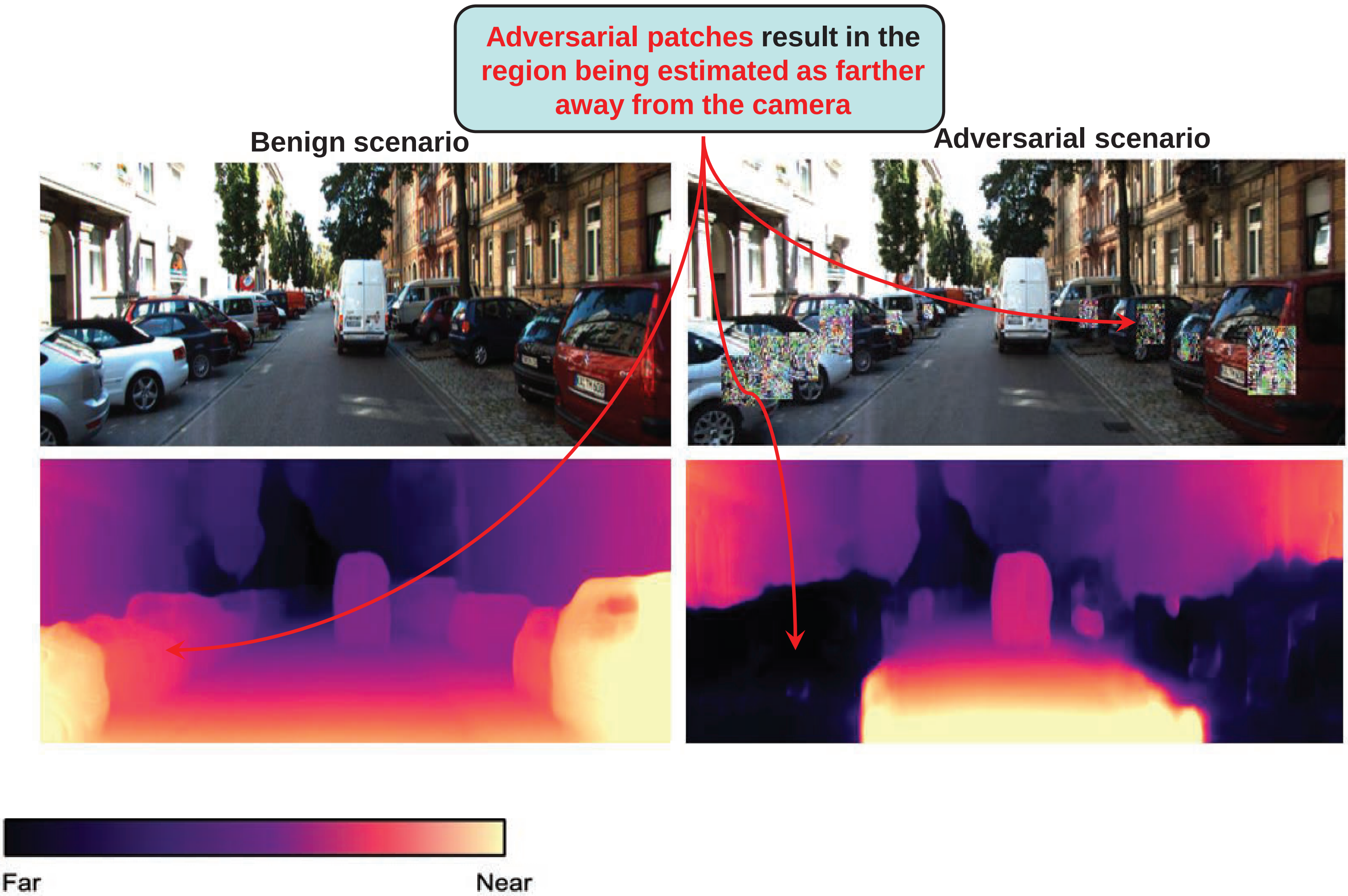


Defenses against Physical Adversarial Attacks



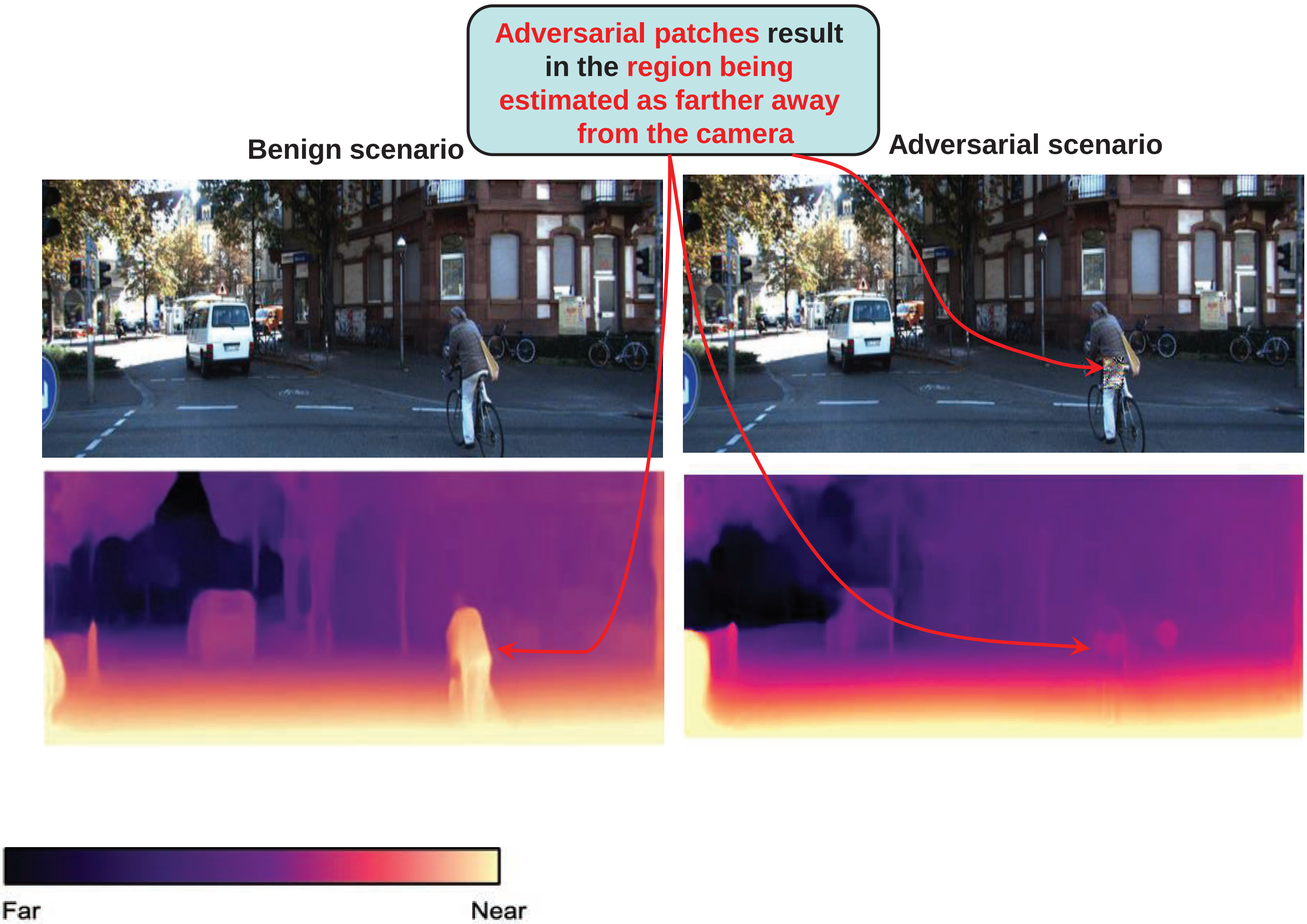
38

Adversarial Attacks on MDE



39

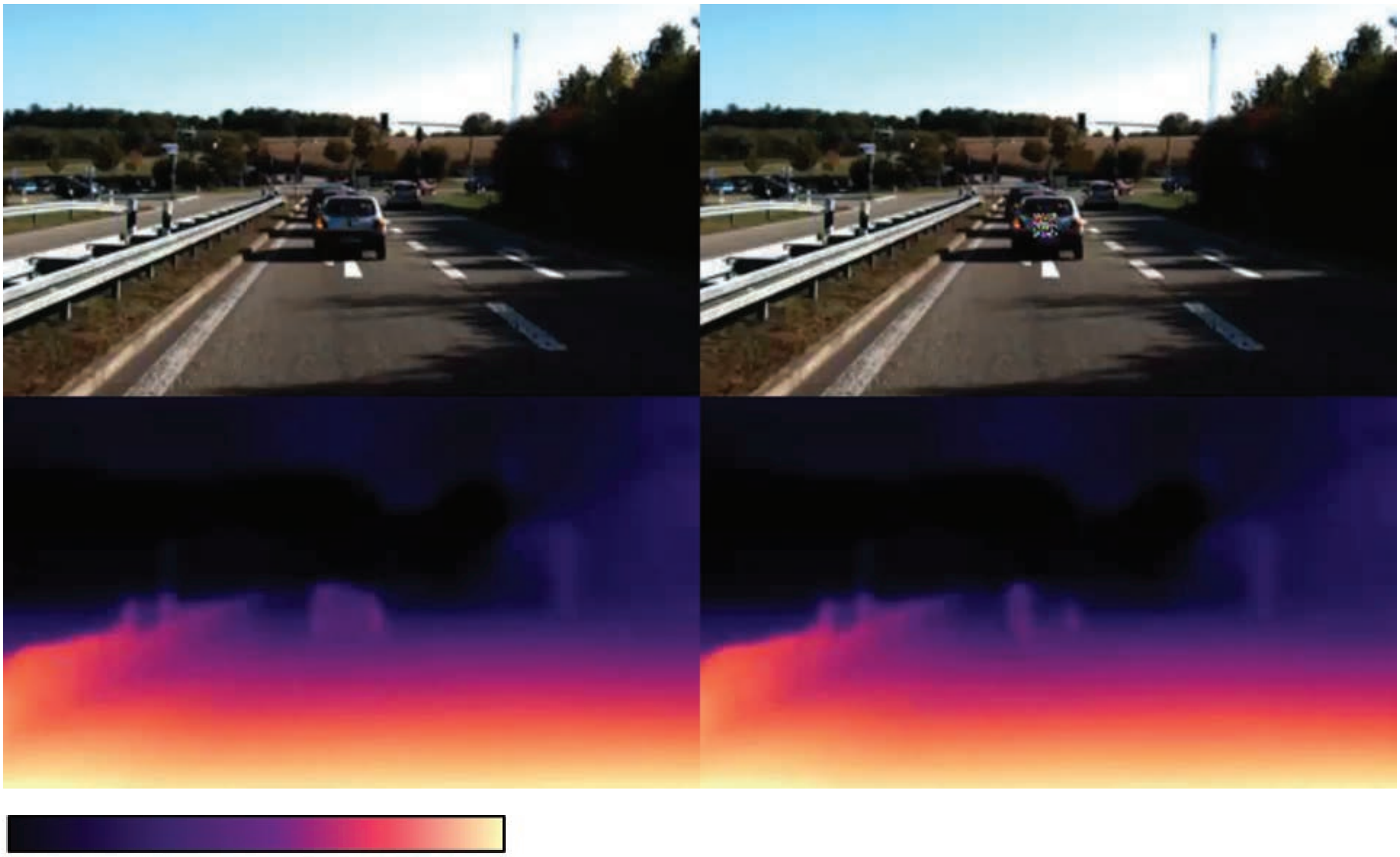
Adversarial Attacks on MDE



40

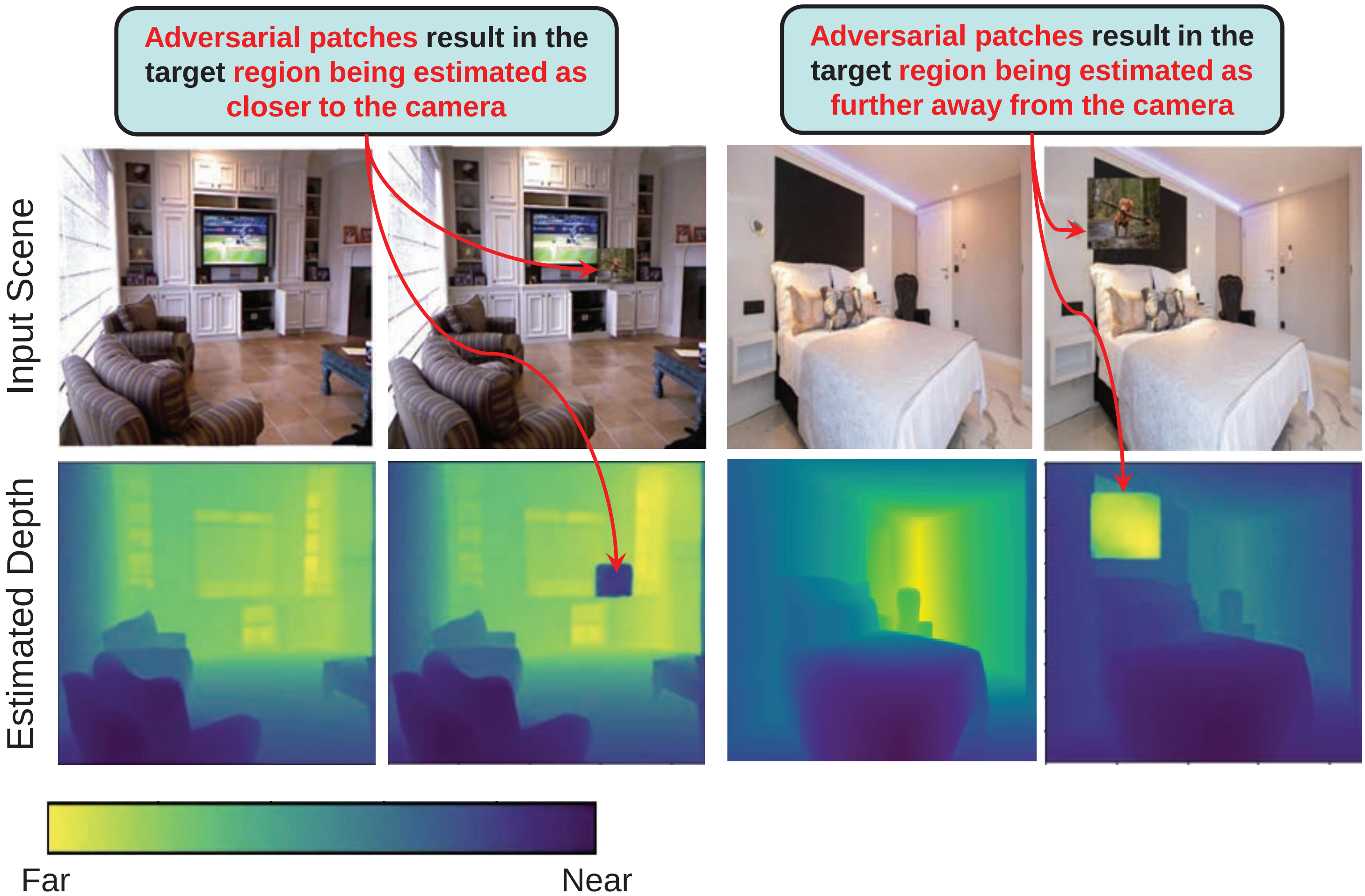
Adversarial Attacks on MDE

Benign scenrio **Adversarial scenario**



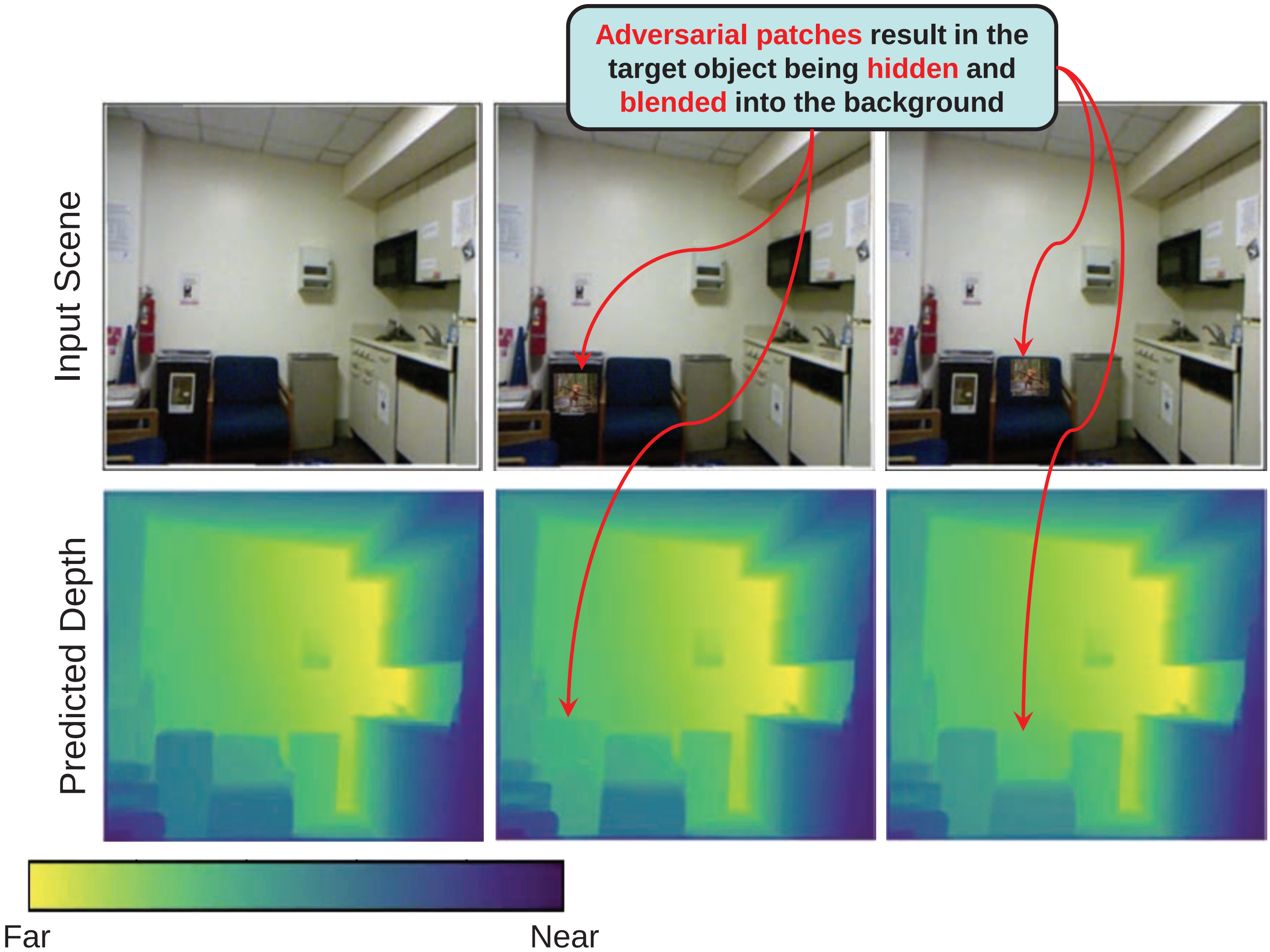
41

Adversarial Attack on MDE



42

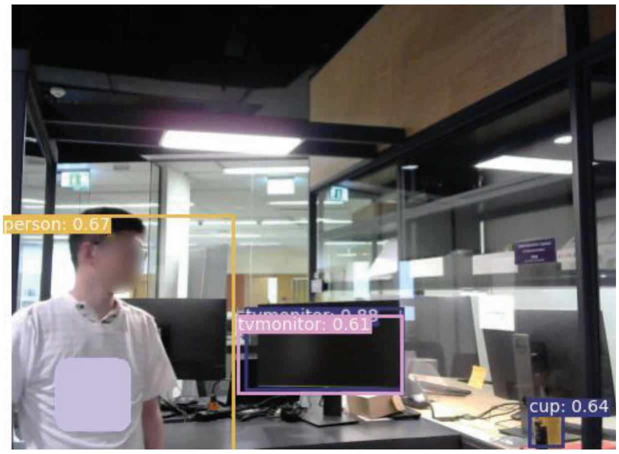
Adversarial Attacks on MDE



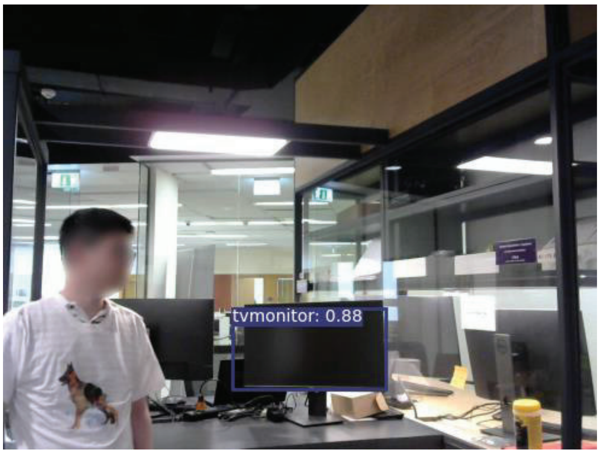
43

Adversarial Attacks on Person Detection

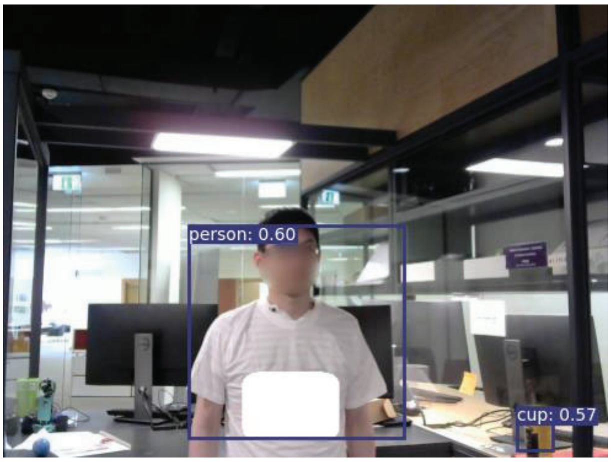
Without Patch: Person **detected**



With Patch: Person **not detected**



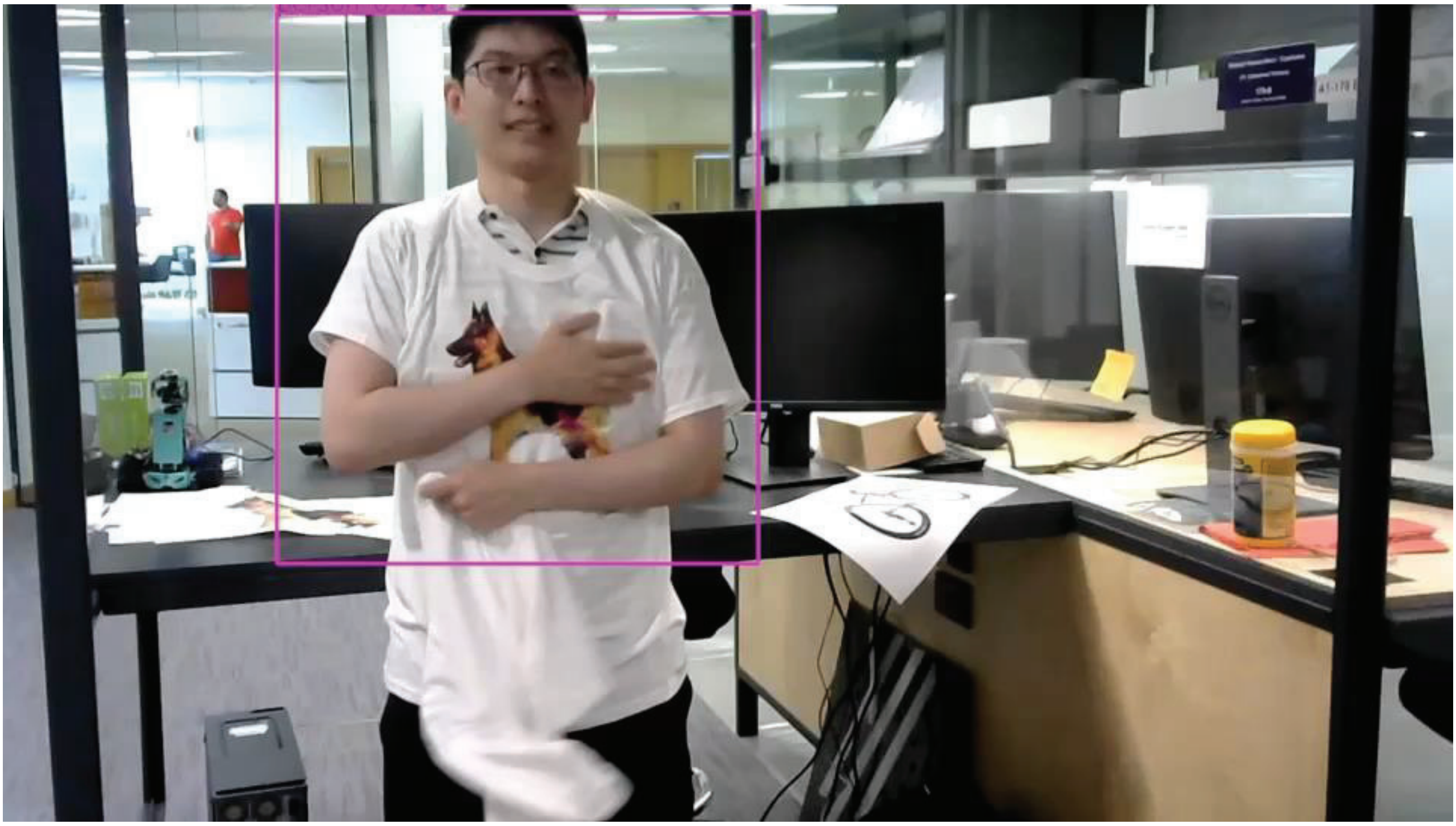
Without Patch: Person **detected**



With Patch: Person **not detected**

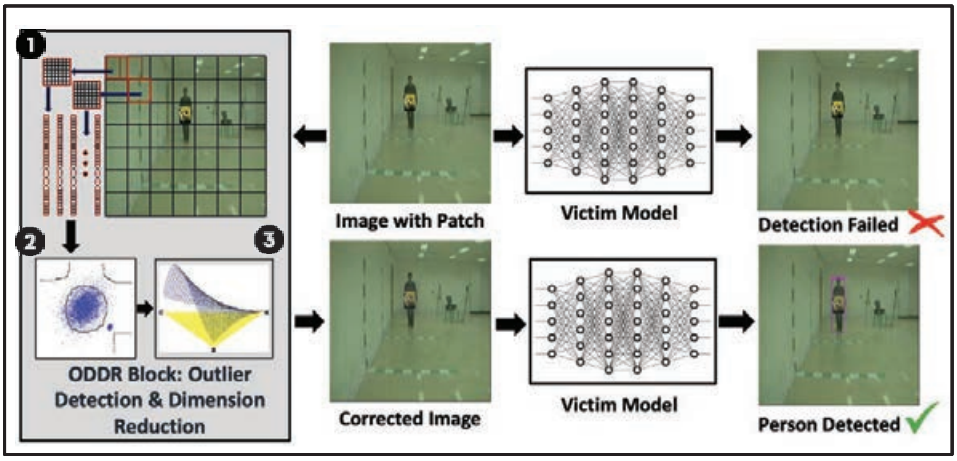
44

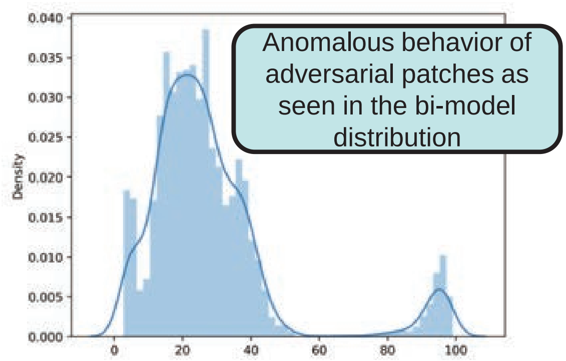
Adversarial Attacks on Person Detection



45

ODDR: Defense Against Patch-based Attacks





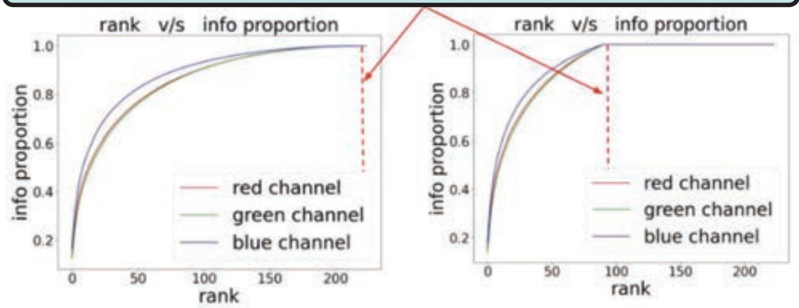
Plot of Mahalanobis distances of segments

1 Fragmentation: Segmenting the image into partially overlapping fragments using a moving square kernel

2 Segregation: Anomaly scores are ascertained to each fragment. Outliers are selected based on a pre-defined threshold. A window of a stipulated size is marked around the cluster center forming a mask

3 Dimension reduction: apply a singular value decomposition on the mask, with a specific hyper-parameter that selects the proportion of information to preserve

Useful information is squeezed within a small portion of the rank

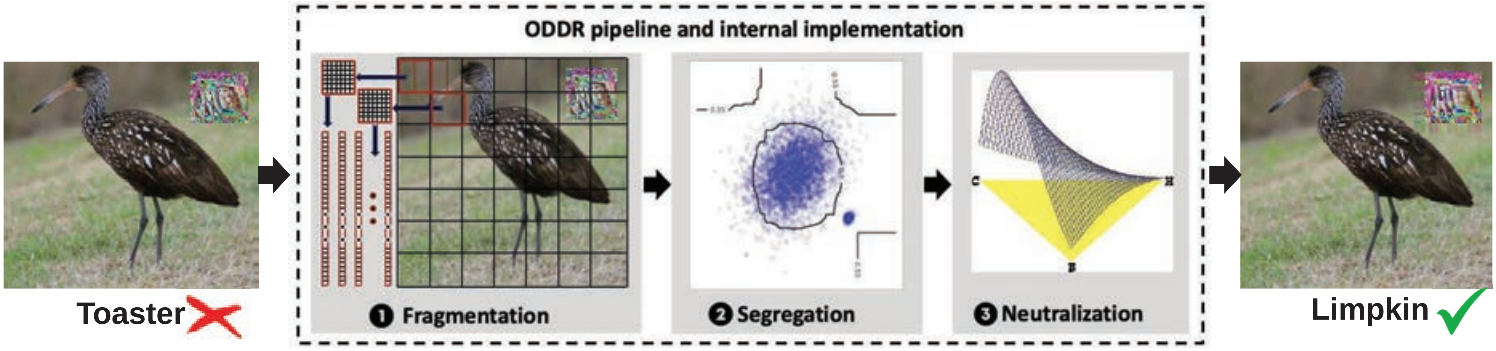


Sample with patch **After dimension reduction**

46

ODDR: Defense Against Patch-based Attacks

- **Classification Task**



❖ **GoogleAp + ImageNet dataset**

Model / Neural Network	Baseline Accuracy	Adversarial Accuracy	Robustness (w/ patch)
ResNet 152	81.2%	39.9%	79.1%
ResNet 50	78.4%	38.8%	75.6%
VGG 19	74.2%	39.1%	72.8%

❖ **GoogleAp + Caltech dataset**

Model / Neural Network	Baseline Accuracy	Adversarial Accuracy	Robustness (w/ patch)
ResNet 152	94.1%	48.6%	90.8%
ResNet 50	90.9%	49.2%	86.4%
VGG 19	88.6%	47.1%	85.6%

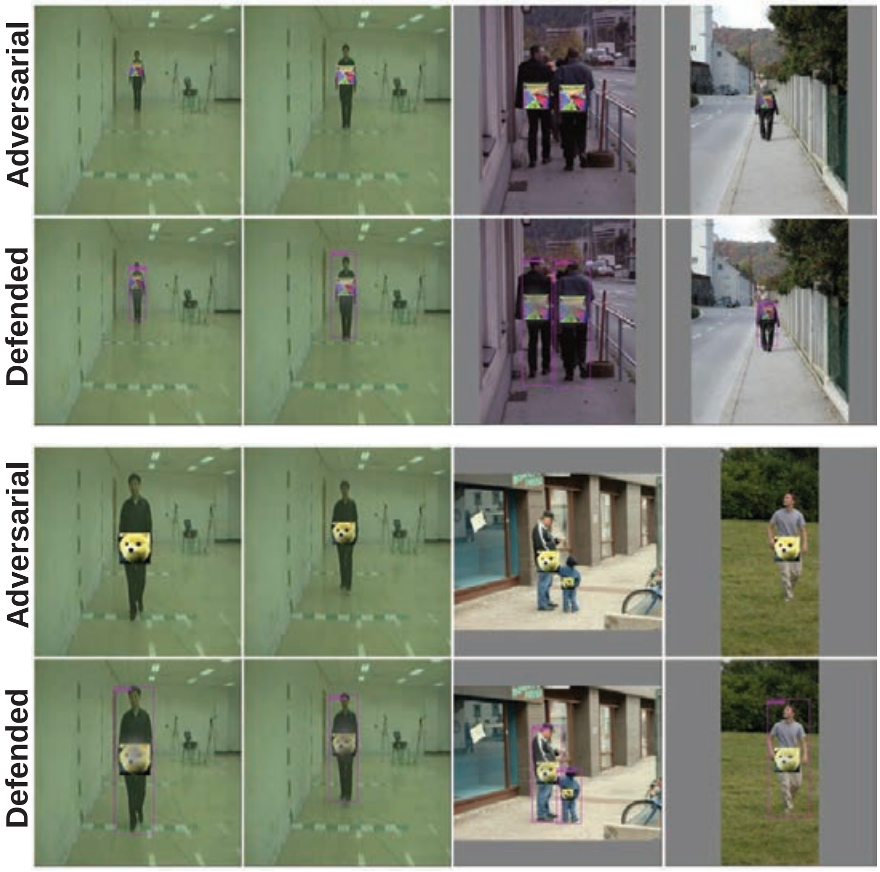
❖ **ODDR vs State-of-the-art Defenses**

Defense	Adversarial Accuracy	Robust Accuracy
LGS [21]	39.26%	53.86%
Jujutsu [5]	39.26%	60%
Jedi [24]	39.26%	64.34%
DS[17]	39.26%	35.02%
PG [26]	39.26%	30.96%
Ours	39.9%	**79.1%**

47

ODDR: Defense Against Patch-based Attacks

- **Object Detection Task**



❖ **AdvYOLO patch**

	INRIA		CASIA	
Defense	Robust Avg. Precision	Recovery Rate	Robust Avg. Precision	Robust Avg. Rate
LGS [21]	21%	27%	84%	48%
Jedi [24]	28.03%	42%	80%	50%
Ours	**82.14%**	**43.86%**	**93.54%**	**61.29%**

❖ **Naturalistic patch**

	INRIA		CASIA	
Defense	Robust Avg. Precision	Recovery Rate	Robust Avg. Precision	Robust Avg. Rate
LGS [21]	55%	29%	50%	48%
Jedi [24]	63%	**51%**	67%	76%
Ours	**65%**	47%	**87.5%**	**90.27%**

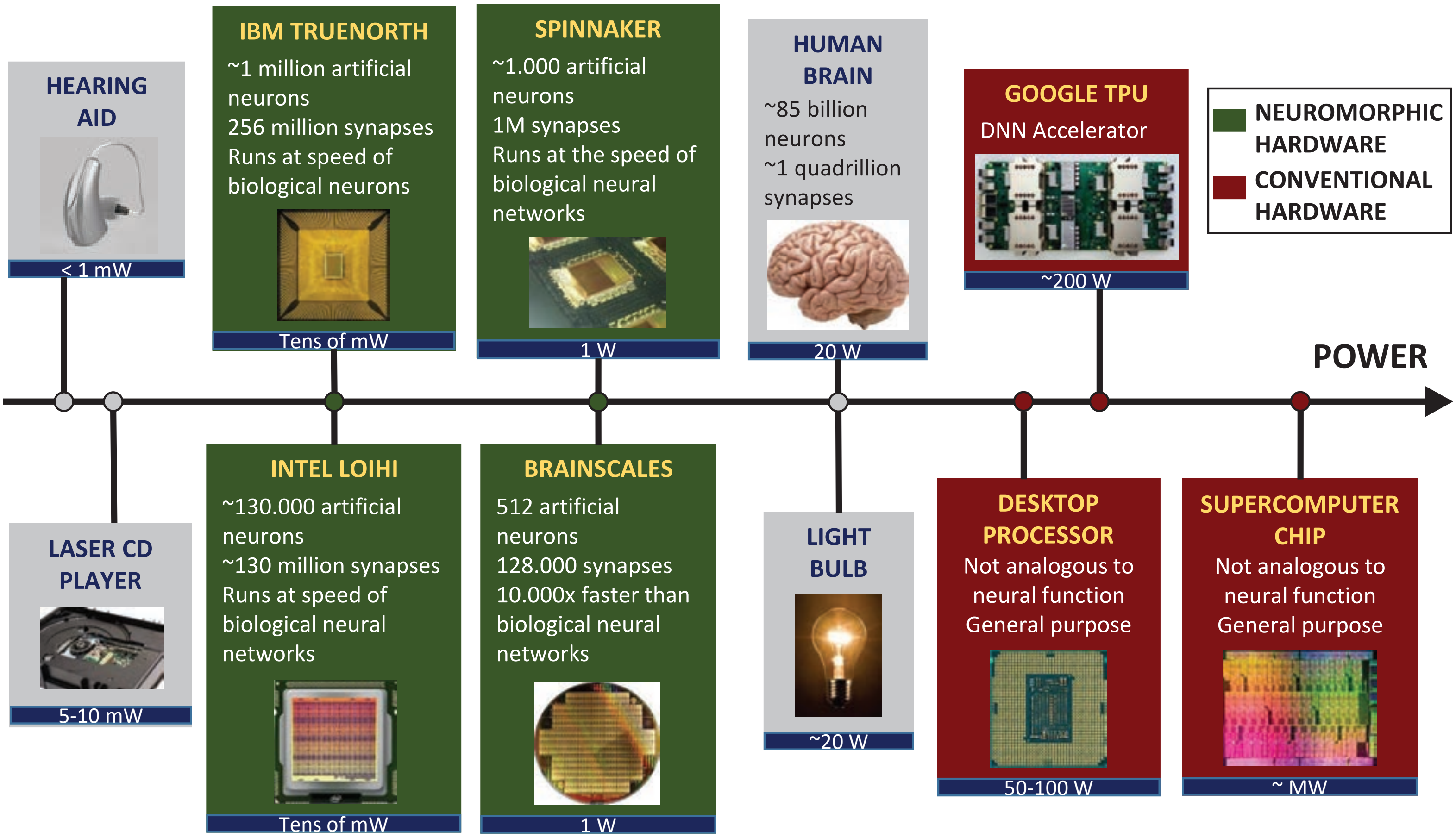
(a) Clean (b) Adversarial (c) ODDR

48

Neuromorphic Hardware & Spiking Neural Networks

49

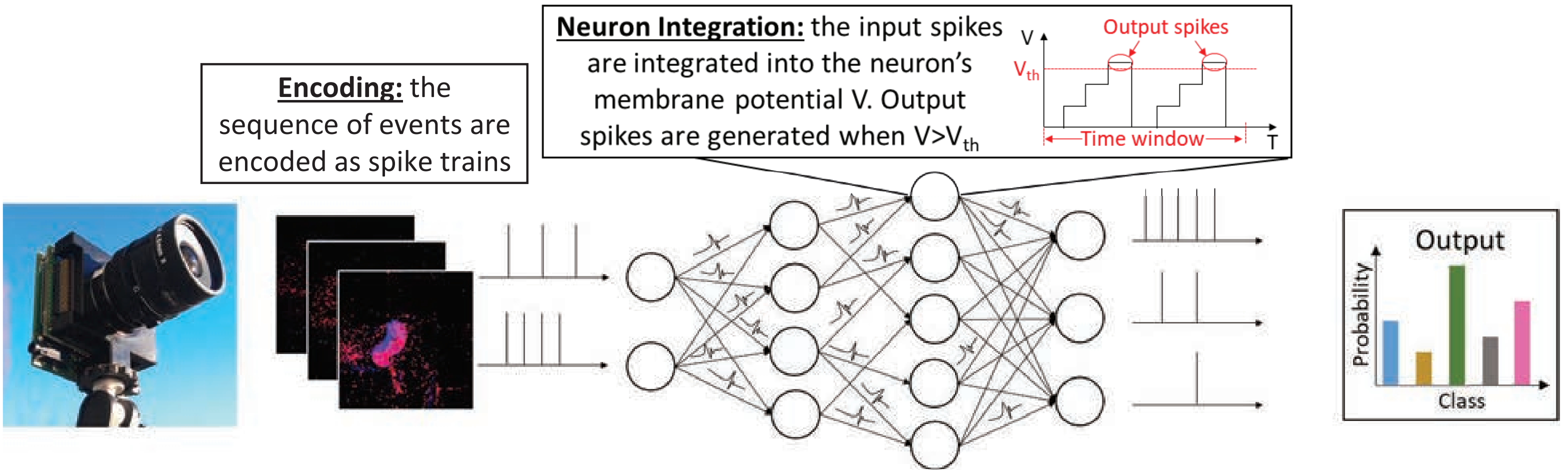
Neuromorphic vs. Conventional Hardware



A. Marchisio, M. Shafique et al. @IEEE Access’20

50

SNNs: Spiking Neural Networks

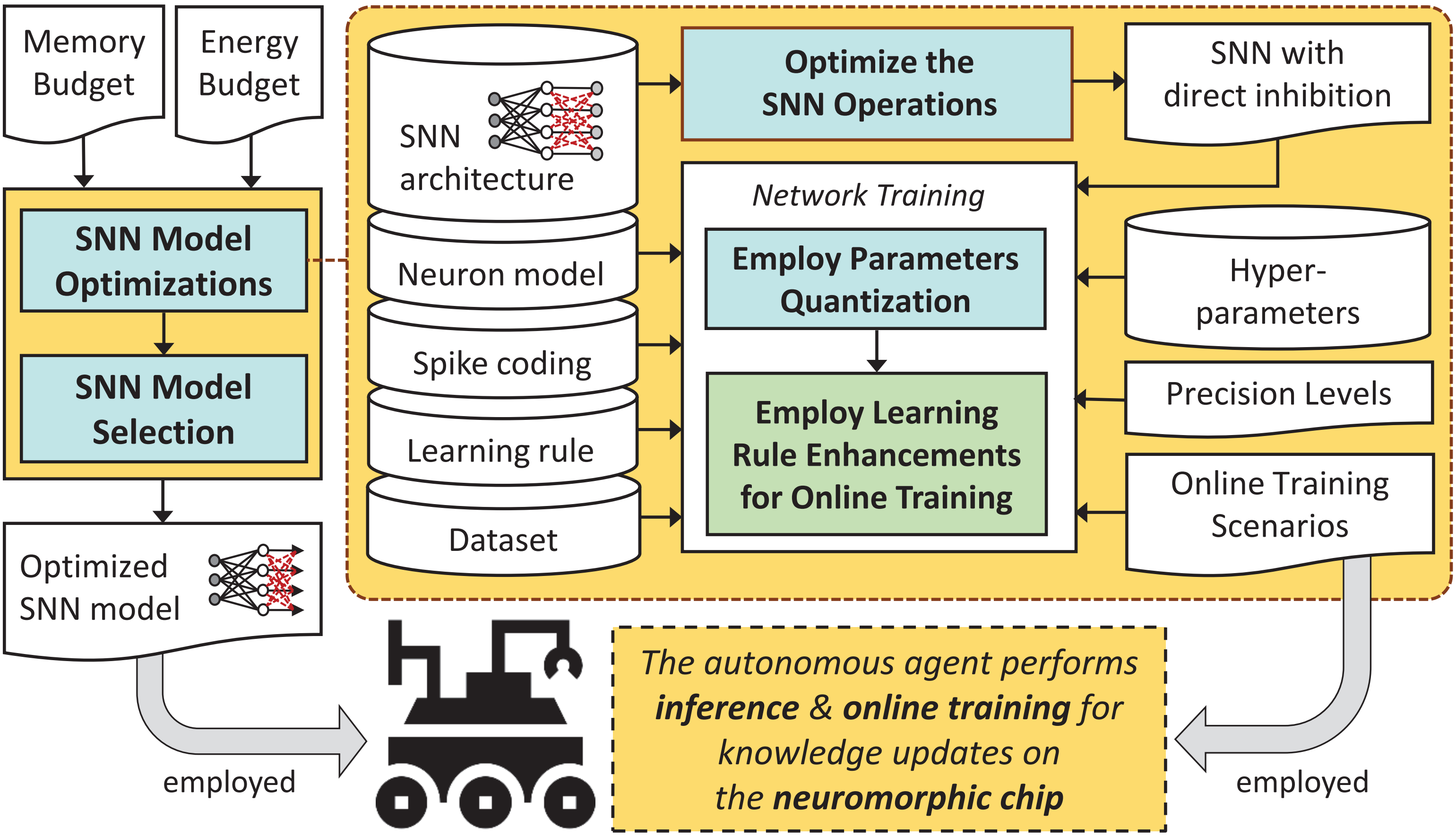


Key Advantages of SNNs:

1. **Biological plausibility**: similar to the human brain's functionality.
2. **Unsupervised learning capability,** due to bio-plausible learning.
3. **Low power/energy consumption,** due to sparse spike-based computations, i.e., event-driven computations activate only in the presence of input spikes.
4. Straightforward interface with **event-based sensors**.

51

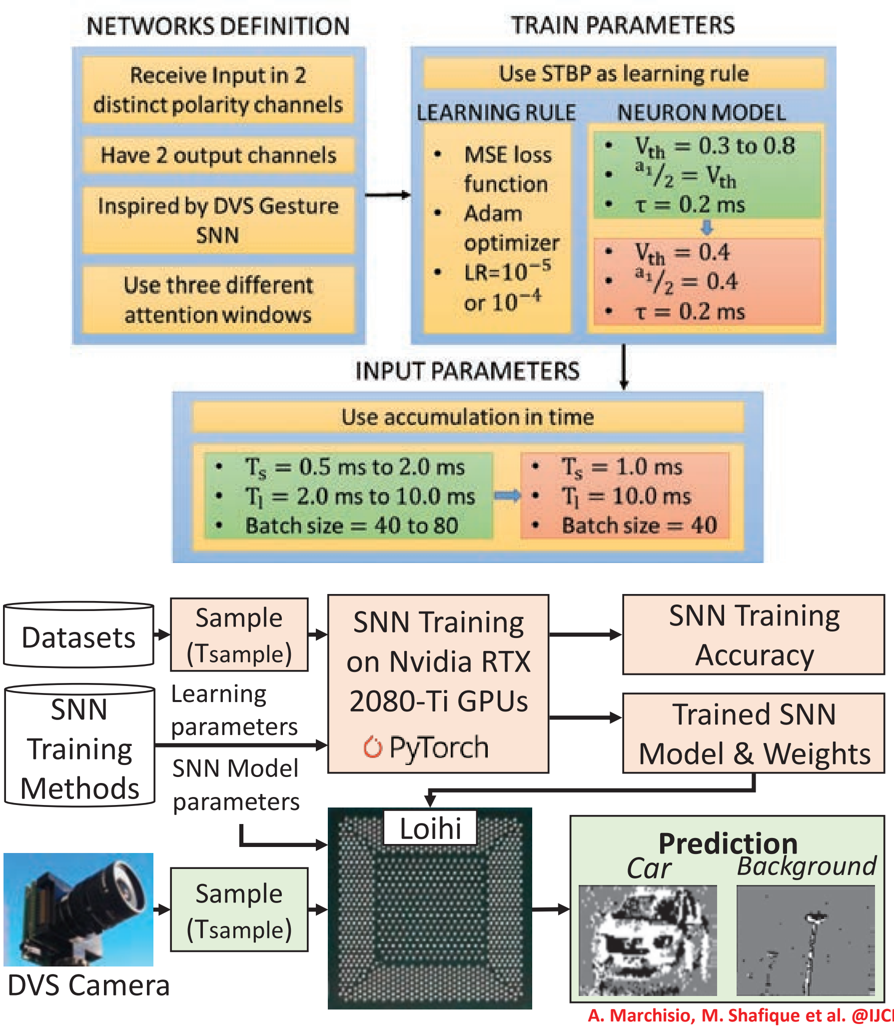
Our Integrated Methodology for Improving Energy Efficiency and Adaptability of SNNs for Autonomous Agents



[Putra and Shafique: Mantis @ ICARA'23]

52

CarSNN: An Event-Based SNN for Autonomous Cars on the Loihi Neuromorphic Processor



A. Marchisio, M. Shafique et al. @IJCNN'21

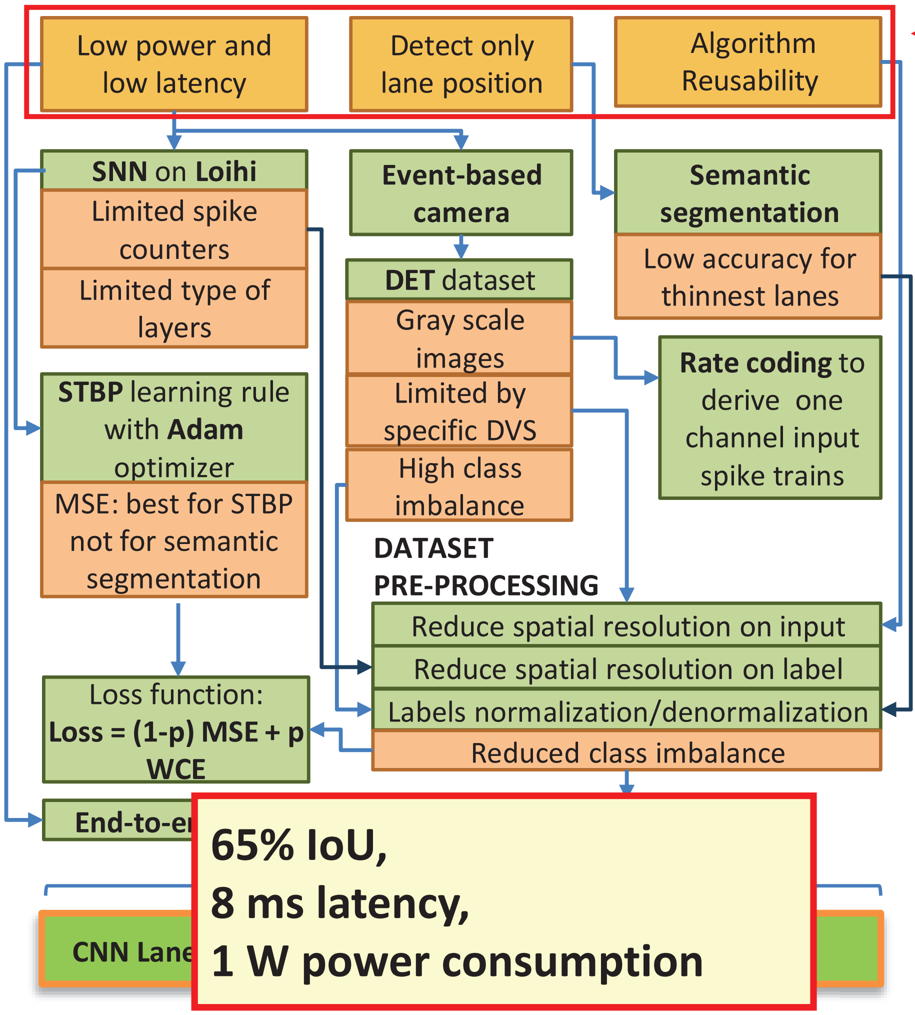
Key Design Steps

- ❑ Attention Window Strategy: the focus is concentrated in the subset of the image where most of the events occur.
- ❑ STDB-based SNN training.
- ❑ Accumulation in time.
- ❑ Exploration of parameters of the SNN:
  - ❑ Threshold Voltage
  - ❑ Batch Size
  - ❑ Sample Length

86% accuracy,
0.72 ms latency,
315 mW power consumption

53

LaneSNNs Design Methodology



Desired Properties

Key Challenges

- ❑ Practical & spatial limitations due to Loihi and DVS.
- ❑ MSE works well for STDP; WCE works well for semantic segmentation.
- ❑ Low accuracy due to high class imbalance (thin lanes).

Key Design Decisions

- ❑ Reduce spatial resolution with pre-processing.
- ❑ Novel Loss Function:

  Loss = (1-p) · MSE + p · WCE

- ❑ Denormalization and normalization of labels; anti-overfitting strategies.

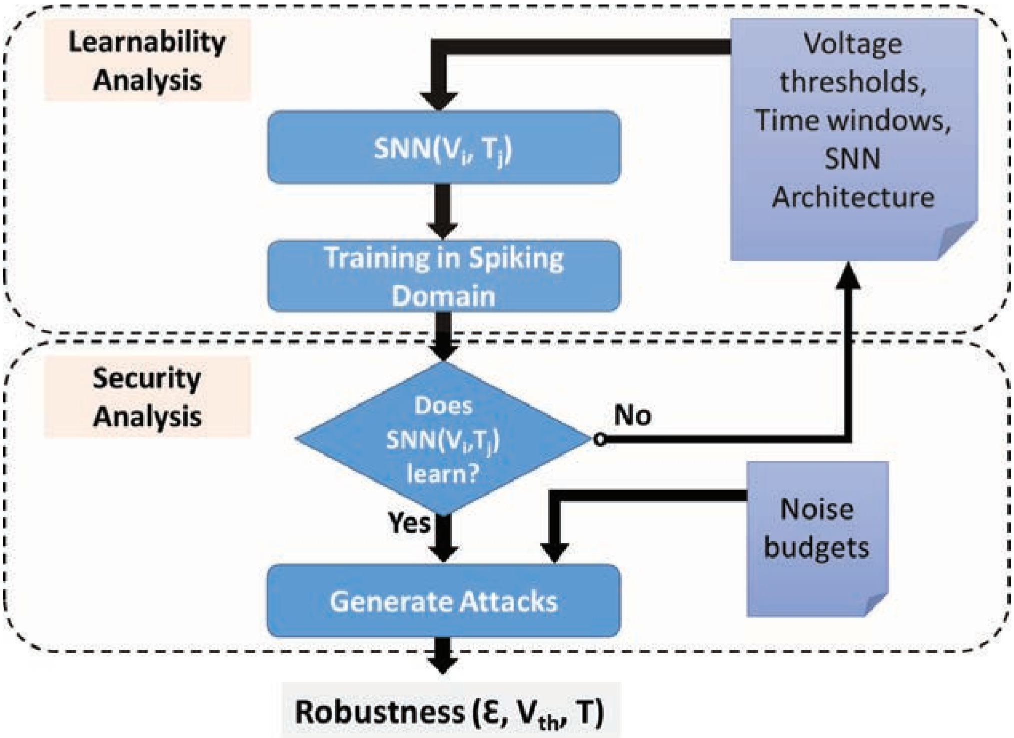
A. Marchisio, M. Shafique et al. @IROS'22

54

SNN Security for Static Data

55

Our Robustness Exploration Methodology

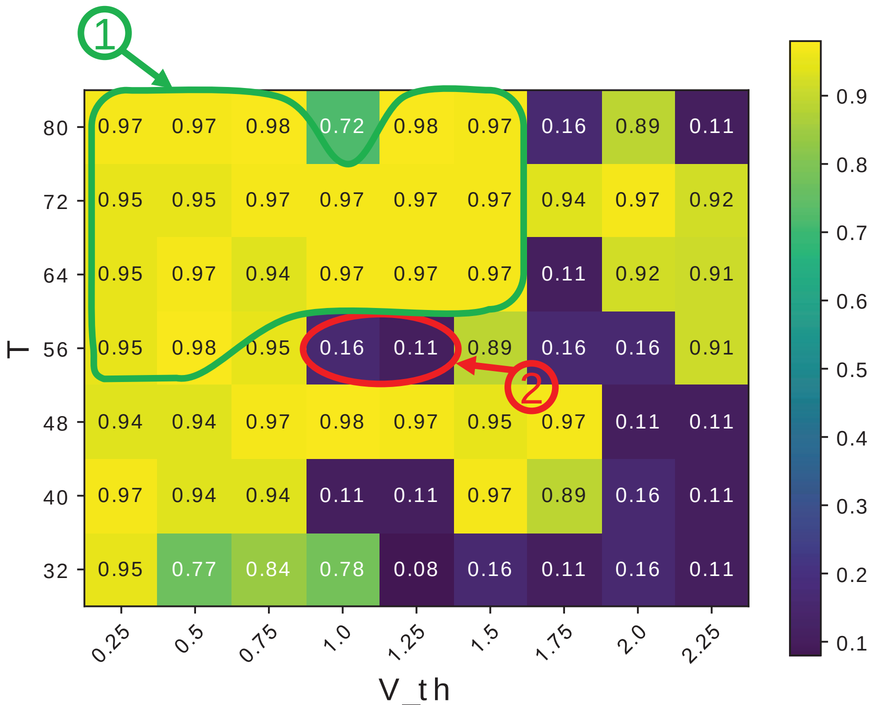


Parameters:

1. **Voltage Threshold (V_{th}):** threshold to be reached by the membrane potential to emit a spike.
2. **Time Window (T):** observation period in which the SNN receives the same input.
3. **Noise Budget ($\mathcal{E}$):** amount of adversarial perturbation allowed by the attack.

A. Marchisio, M. Shafique et al. @DATE'21

Results: Learnability Analysis



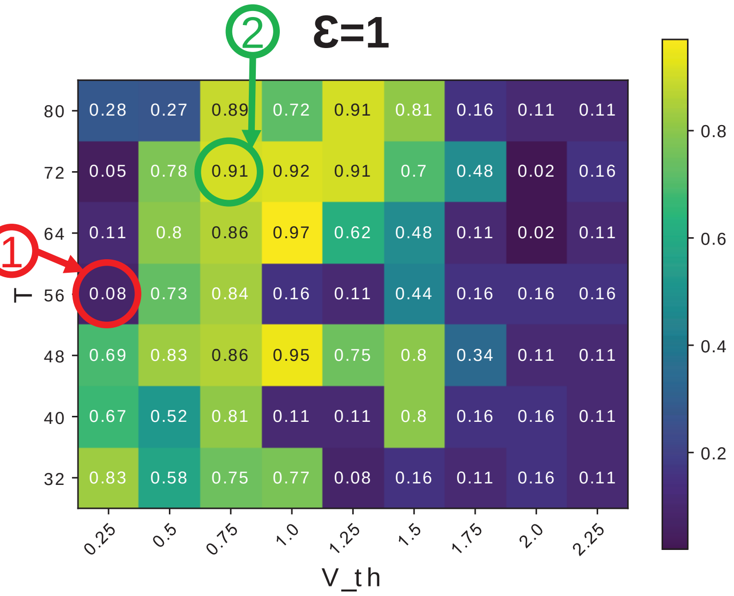
Key Observations:

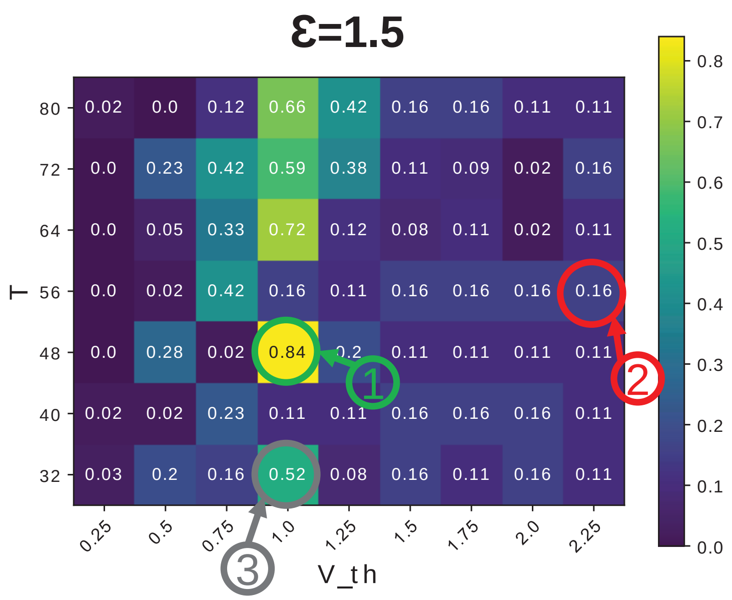
Accuracy heat map for different combinations of (V_{th}, T)

1. The **highest-accuracy** combination tends to be towards the **top-left corner**, i.e., low V_{th} and high T.
2. The heat map is clearly **not monotonic**. For example, there are combinations with an accuracy lower than 16% which are surrounded by combinations with accuracy higher than 89%.

A. Marchisio, M. Shafique et al. @DATE'21

Results: Security Analysis





Key Observations:

Two SNNs with a high baseline accuracy may have different behaviors under attack.

1. One **drops drastically to 8%**.
2. Another one **loses only 6%** of its initial accuracy.

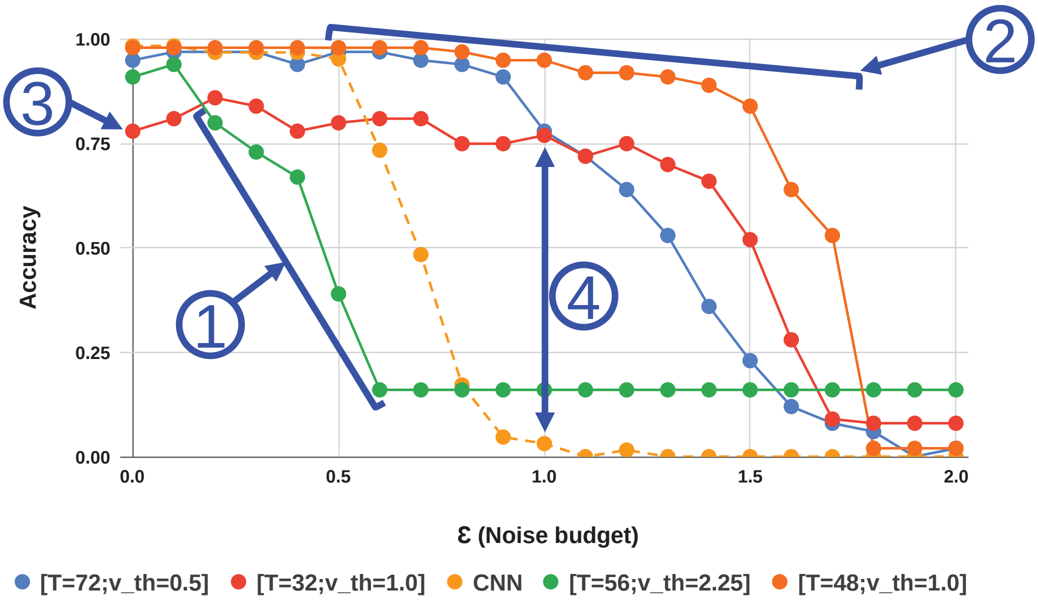
Different types of Attack Robustness:

1. **High robustness** , e.g., (V_{th}, T) = (1, 48)
2. **Low robustness** , e.g., (V_{th}, T) = (2.25, 56)
3. **Medium Robustness**, e.g., (V_{th}, T) = (1, 32)

A. Marchisio, M. Shafique et al. @DATE'21

58

Results: CNN vs. SNN Comparison



Key Observations:

1. (V_{th}, T) = (2.25, 56) has **lower robustness than the CNN**.
2. (V_{th}, T) = (1, 48) has very **high robustness**.
3. (V_{th}, T) = (1, 32) has **clean accuracy of only 78%**, but…
4. It has **75% higher accuracy** than the CNN when Ɛ>1.

A. Marchisio, M. Shafique et al. @DATE'21

59

SNN Security for Event-Based Data

60

Normally-Distributed DVS Perturbations

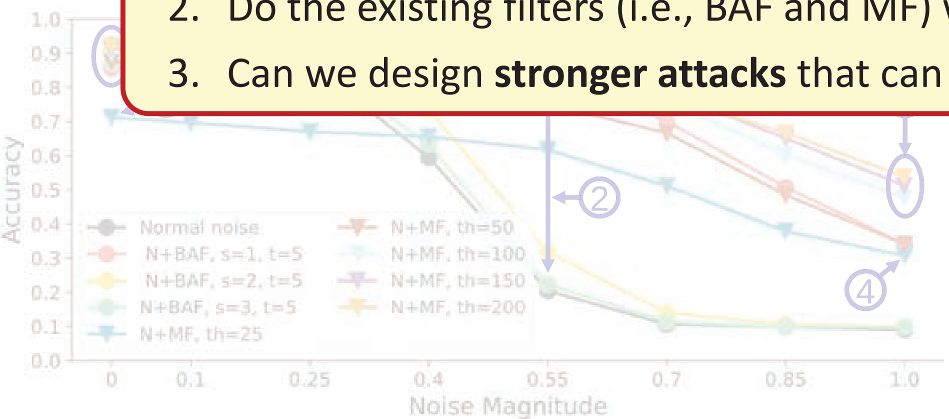
Background Activity Filter (BAF):

Mask Filter (MF):

A mask is set based on the event activities of

Research Questions:

1. How to generate **adversarial attacks for DVS-based signals**?
2. Do the existing filters (i.e., BAF and MF) work as a **defense**?
3. Can we design **stronger attacks** that can bypass the noise filter?



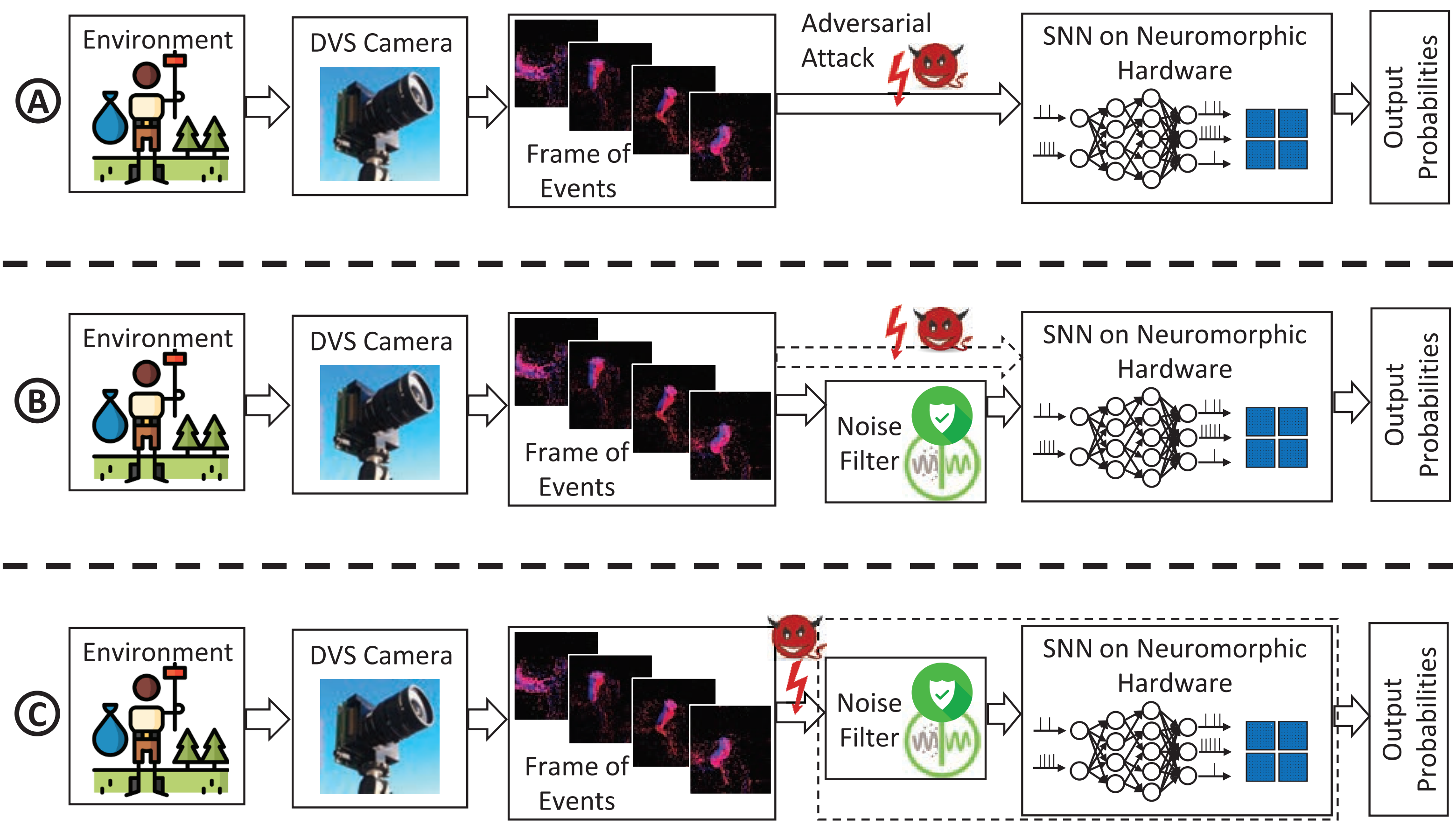
Key Observations:

1. Small accuracy reduction when **no**
3. The **MF with T>150** is robust against large perturbations.
4. The **BAFs are less robust than MFs** for large perturbations.

A. Marchisio, M. Shafique et al. @IJCNN'21

61

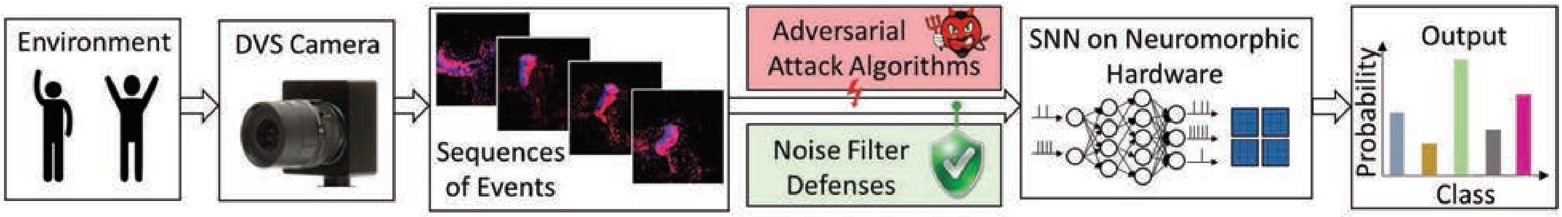
Our R-SNN Methodology – Adversary Threat Models



A. Marchisio, M. Shafique et al. @IROS'21

62

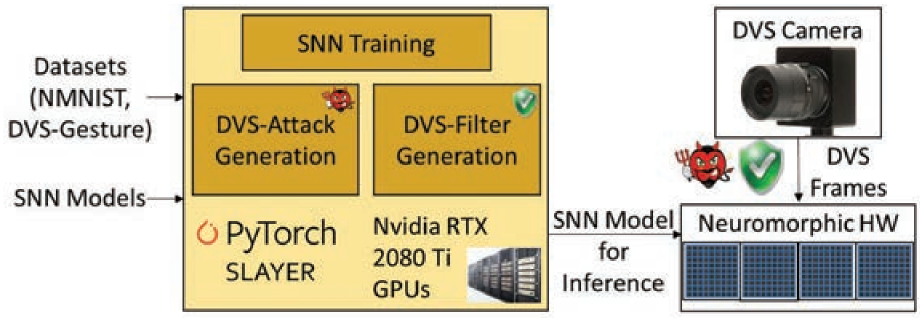
DVS-Attacks Methodology: Adversarial Attacks on Dynamic Vision Sensors for SNNs



Key Features:

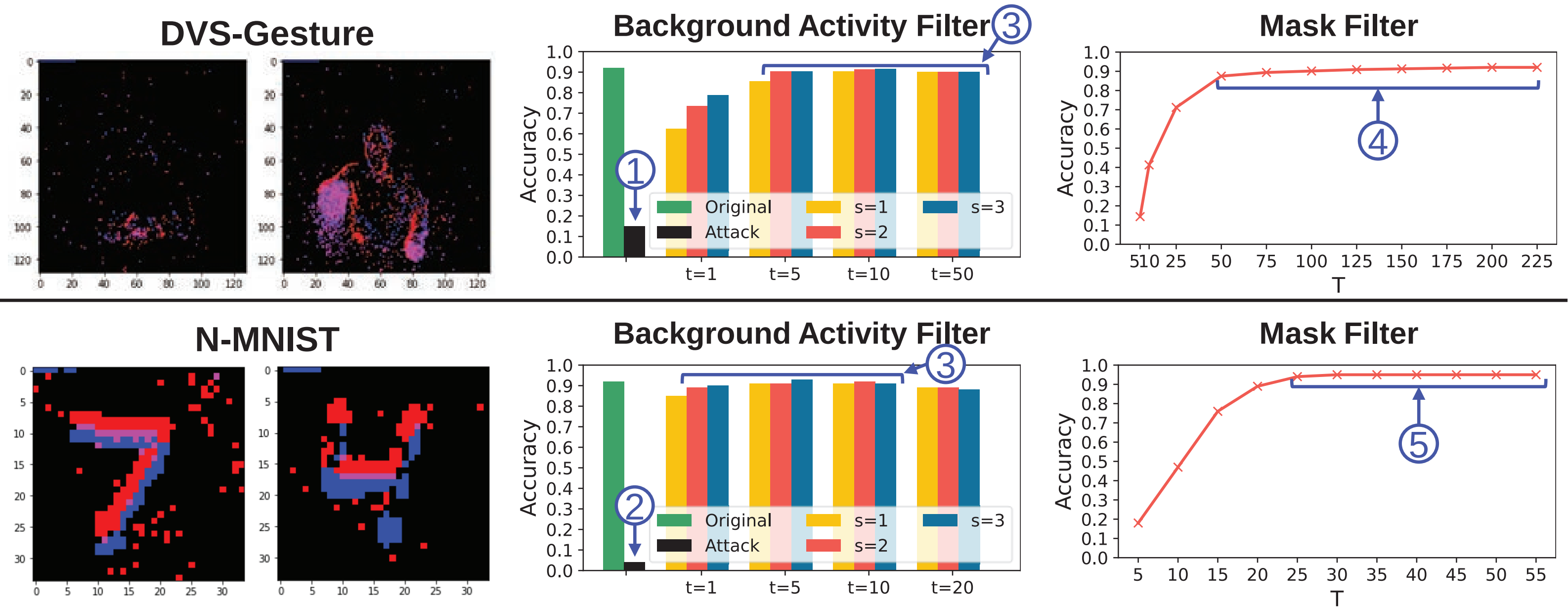
- 5 adversarial attack algorithms
- 2 DVS-noise filters (BAF, MF)
- Tuning of spatial and temporal filter parameters.
- Evaluation on 2 datasets (N-MNIST, DVS-Gesture)

1. Sparse Attack
2. Frame Attack
3. Corner Attack
4. Dash Attack
5. MF-Aware Dash Attack



A. Marchisio, M. Shafique et al. @IJCNN'21

Results: SNN Robustness against Sparse Attack



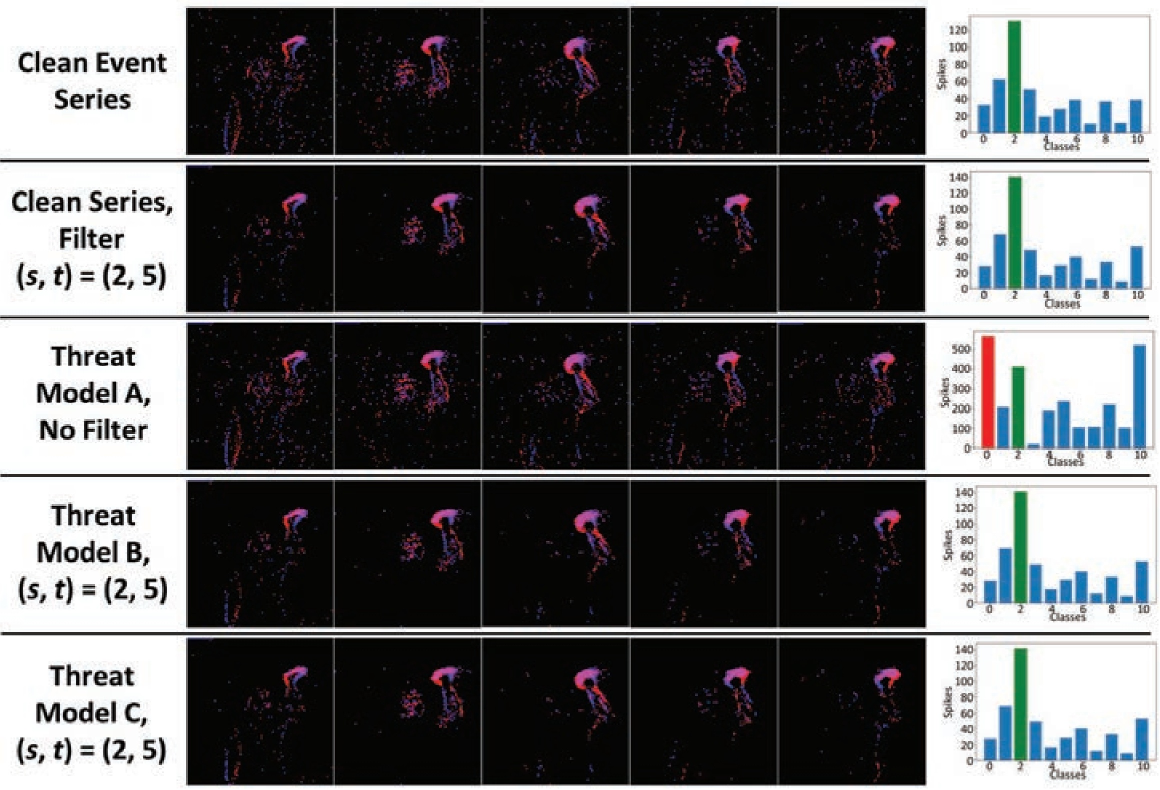
Key Observations:

1. The accuracy drops to 15% for DVS-Gesture.
2. The accuracy drops to 4% for N-MNIST.
3. The BAF restores the clean accuracy when the Sparse Attack is applied.
4. The **MF with T>50** is robust against Sparse Attack on DVS-Gesture.
5. The **MF with T>25** is robust against Sparse Attack on N-MNIST.

A. Marchisio, M. Shafique et al. @IJCNN'21

64

Results: SNN Robustness against Sparse Attack

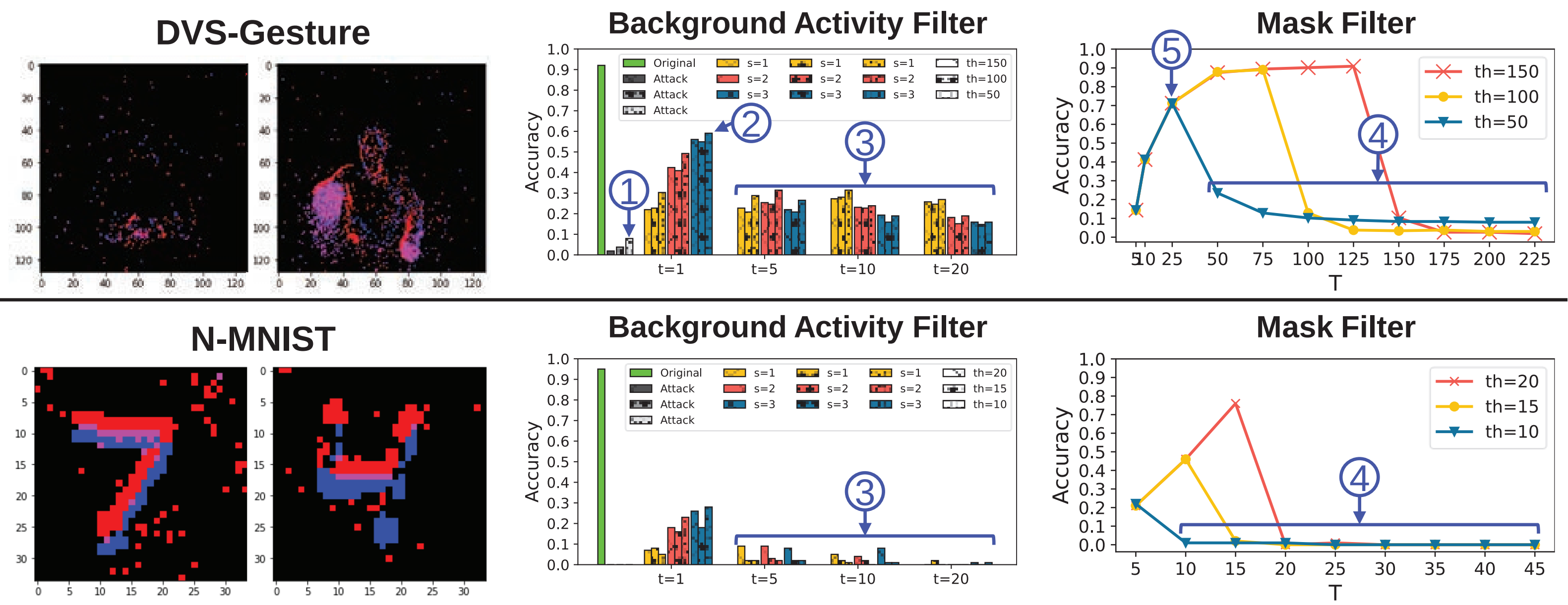


A. Marchisio, M. Shafique et al. @IROS'21

Key Observations:

1. Correct classification of clean inputs.
2. With filter, the frames of events are different, but the change in the output probability is small.
3. Threat model A: The attack generates a misclassification.
4. Threat model B: correct classification thanks to the filter.
5. Threat model C: similar to B.

Results: SNN Robustness against MF-Aware Dash Attack



Key Observations:

1. The accuracy drops to **< 8% with no filter**.
2. The accuracy is restored to 59% using the **BAF** with (s, t) = (3, 1) for DVS-Gesture.
3. Using the **BAF** with t >= 5, the accuracy is < 32% for DVS-Gesture and < 13% for N-MNIST.
4. Using the **MF** with T >= th, the accuracy is < 25% for DVS-Gesture and < 2% for N-MNIST.
5. The peak reached by the MF with T = 25 against the MF-Aware Dash Attack with th = 50 has 71% accuracy (**21% lower than the original SNN accuracy**).

A. Marchisio, M. Shafique et al. @IJCNN'21

References: ML Papers @ eBRAIN Lab

- Amira Guesmi, Ruitian Ding, Muhammad Abdullah Hanif, Ihsen Alouani, Muhammad Shafique: DAP: A Dynamic Adversarial Patch for Evading Person Detectors. CVPR, 2024.
- Nandish Chattopadhyay, Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, Muhammad Shafique: DefensiveDR: Defending against Adversarial Patches using Dimensionality Reduction. DAC, 2024.
- Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, Muhammad Shafique: SAAM: Stealthy Adversarial Attack on Monocular Depth Estimation. IEEE Access, 2024.
- Amira Guesmi, Muhammad Abdullah Hanif, Ihsen Alouani, Bassem Ouni, Muhammad Shafique: SSAP: A Shape-Sensitive Adversarial Patch for Comprehensive Disruption of Monocular Depth Estimation in Autonomous Navigation Applications. CoRR abs/2403.11515, 2024.
- Rachmad Vidya Wicaksana Putra, Muhammad Shafique: Mantis: Enabling Energy-Efficient Autonomous Mobile Agents with Spiking Neural Networks. ICARA, 2023.
- Nandish Chattopadhyay, Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, Muhammad Shafique: ODDR: Outlier Detection & Dimension Reduction Based Defense Against Adversarial Patches. CoRR abs/2311.12084, 2023.
- Alberto Viale, Alberto Marchisio, Maurizio Martina, Guido Masera, Muhammad Shafique: LaneSNNs: Spiking Neural Networks for Lane Detection on the Loihi Neuromorphic Processor. IROS, 2022.
- Alberto Marchisio, Giacomo Pira, Maurizio Martina, Guido Masera, Muhammad Shafique: R-SNN: An Analysis and Design Methodology for Robustifying Spiking Neural Networks against Adversarial Attacks through Noise Filters for Dynamic Vision Sensors. IROS, 2021.
- Rida El-Allami, Alberto Marchisio, Muhammad Shafique, Ihsen Alouani: Securing Deep Spiking Neural Networks against Adversarial Attacks through Inherent Structural Parameters. DATE, 2021.
- Alberto Marchisio, Giacomo Pira, Maurizio Martina, Guido Masera, Muhammad Shafique: DVS-Attacks: Adversarial Attacks on Dynamic Vision Sensors for Spiking Neural Networks. IJCNN, 2021.
- Alberto Viale, Alberto Marchisio, Maurizio Martina, Guido Masera, Muhammad Shafique: CarSNN: An Efficient Spiking Neural Network for Event-Based Autonomous Cars on the Loihi Neuromorphic Research Processor. IJCNN, 2021.

67

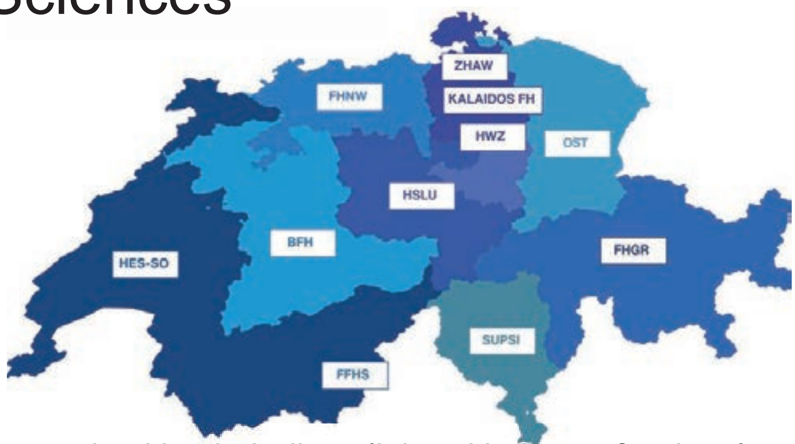

Thank You!
BRAIN
Dr. Alberto Marchisio
alberto.marchisio@nyu.edu
#myNYUAD

Distributed Cloud Continuum Platform for Federated Learning Based Self-Adaptive IoT Applications

Nabil Abdennadher

CPS&IoT’2024 Budva (Montenegro), 11 - 14 June 2024

Universities of applied Sciences (UAS) in Switzerland

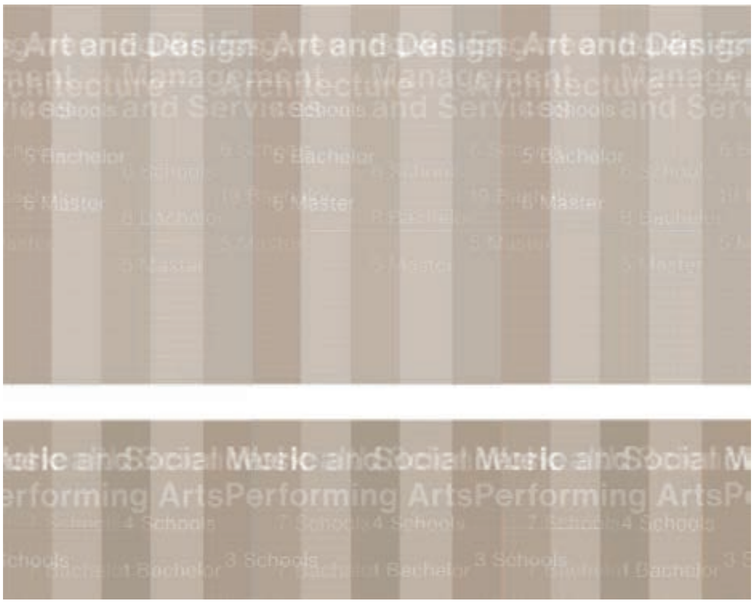


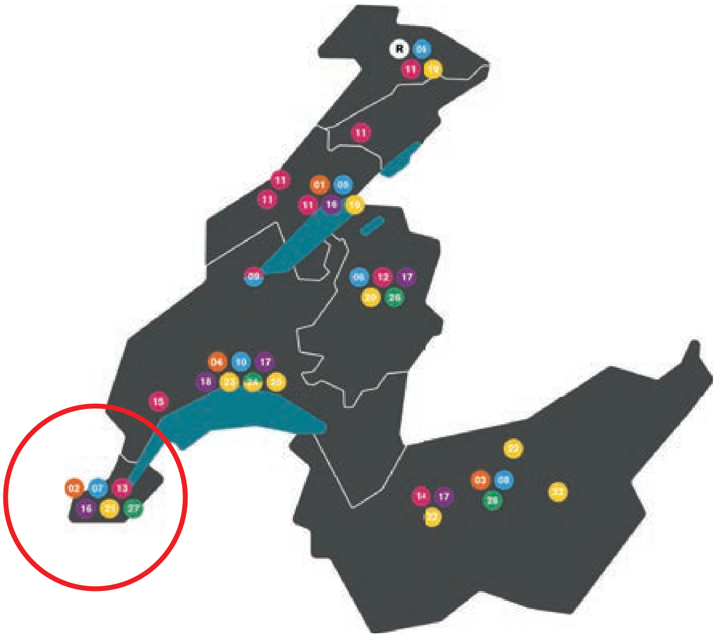
- UAS studies are practice-orientated, characterised by their direct links with the professional world.
- The UAS offer Bachelor's, Master's degree courses … and doctorates in collaboration with the universities.
- The UAS carry out practice-oriented research and innovation projects.
- One UAS in Western Switzerland: the **HES-SO** with 21'270 students.

2

Hes·so

Six fields of study





3

Hes·so

The HES-SO @ glance

- Founded in 1998
- More than 21,000 students
- 28 schools in 7 cantons
- 70 Bachelor's and Master's programmes
- 18,524 employees (4,323 FTE)
- 78 institutes and research units

Hes·so

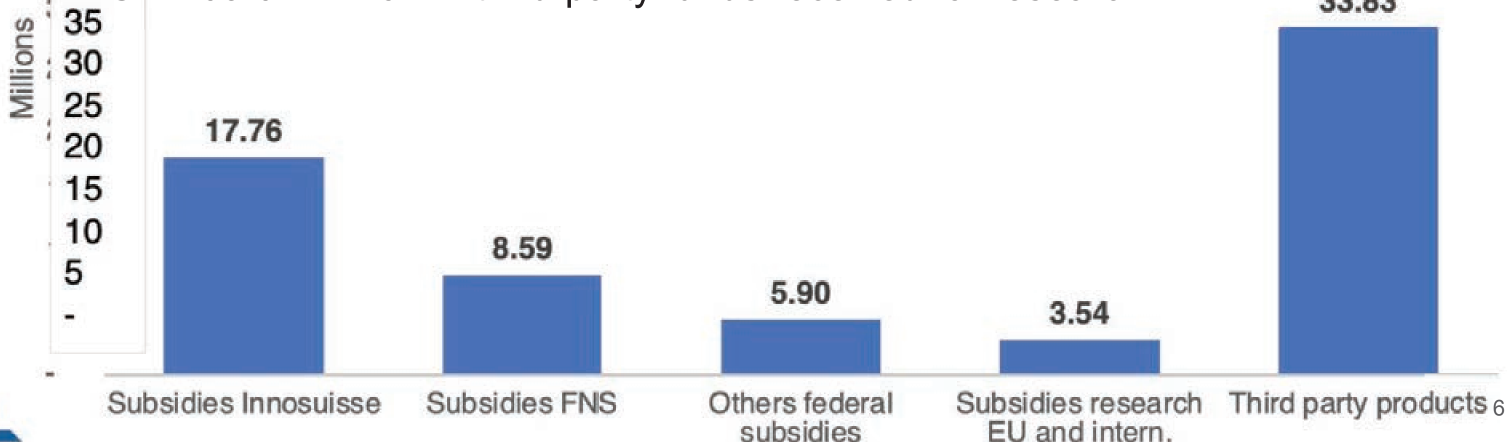
Applied Research @ HES-SO

- More than 750 competitive projects.
- Projects supported mainly by **Innosuisse**, Swiss National Fund and European fundings.
- Results are directly useful for companies, cultural and social/health institutions, authorities or civil society.
- PhD jointly with a university of applied sciences.

5

Hes·so Research funding in 2021

- CHF 44.38 million allocated to the Impulse Research Fund (IRF)
- CHF 21.15 million received from the Confederation for federal research funding
- CHF 23.23 million contribution from the cantons
- CHF 69.61 million in third-party funds received for research



6

Hes·so

Distributed Cloud Continuum Platform for Federated Learning Based Self-Adaptive IoT Applications

CPS&IoT'2024 Budva (Montenegro), 11 - 14 June 2024

Raoul Dupuis

Nabil Abdennadher

7

Hes·so Before starting …

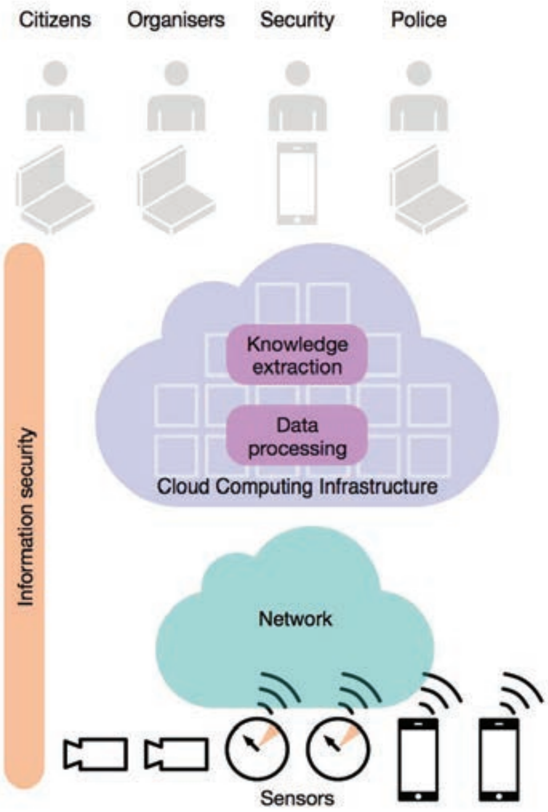
- **Are you familiar with:**
  - Docker / Docker Compose / Docker SWARM ?
  - Kubernetes ?
  - Cloud IaaS / PaaS / SaaS ?
  - IoT ?
  - AI / Machine Learning / Deep Learning ?
  - MQTT ?

Container Orchestration Engine (COE)

8

Hes·so The context

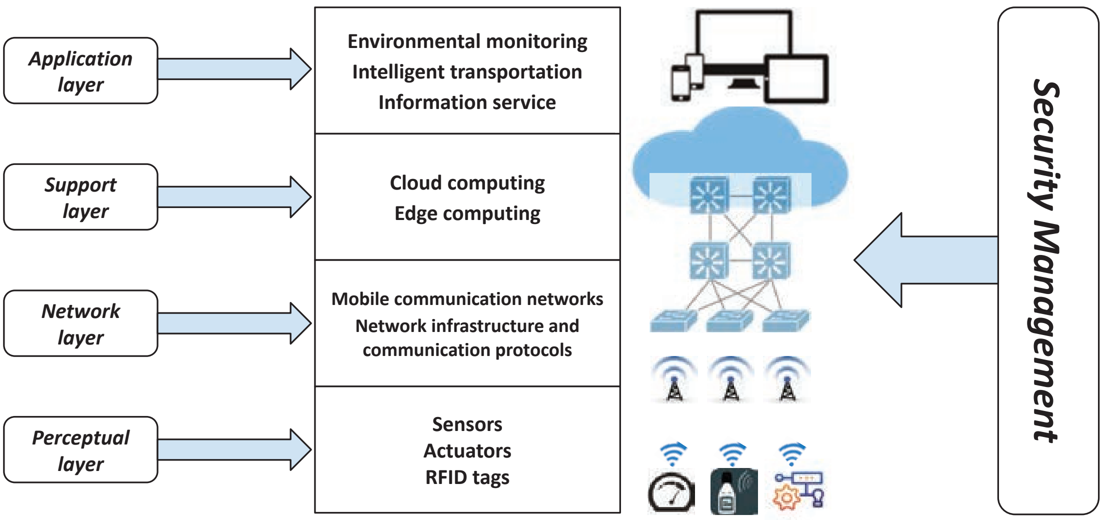
- Thousands of connected IoT sensors deployed at a large scale.
- Several applications are emerging.
  - Data sensitive
  - Compute intensive
  - Context-aware
- Centralised IoT platforms are ill equipped to cope with the huge quantity of collected data.
- … And here where the **Edge-to-Cloud** and **Cloud Continuum** come in



9

The context

The 4 layers of the IoT platforms



10

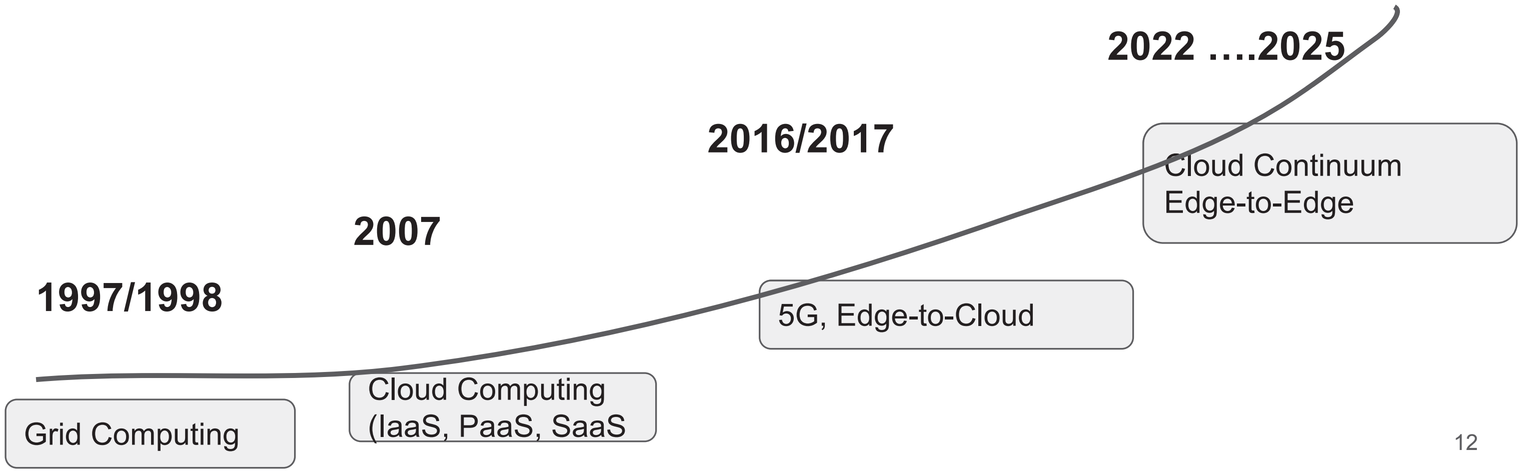
Hes·so Plan

- What is Cloud Continuum ?
- Why Cloud Continuum ?
- Use-cases
- Cloud Continuum (Edge-to-Cloud) solutions
- The Smart Grid Application

11

What is Cloud Continuum?

- Convergence between Cloud and the Internet of Things
- A spectrum of computing tools (HW/SW) which covers Cloud Data Centers, High Performance Computing, Edges nodes, AI, 5G/6G

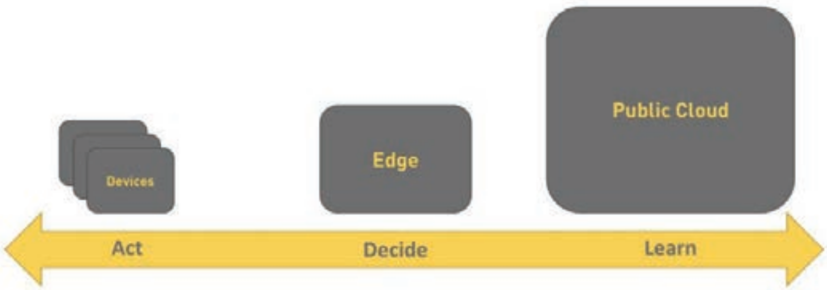


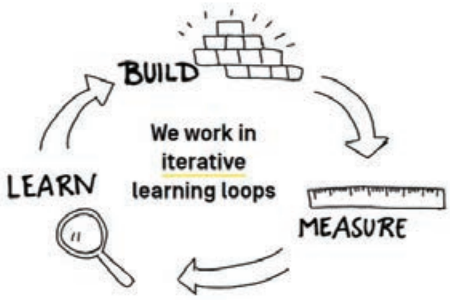
12

Hes·so

What is Cloud Continuum? Attempted definition

It is the current trend of developing, deploying and running **highly distributed, context-aware, computing intense and data-sensitive** applications on a set of IT resources ranging from **high density compute to lightweight embedded computers** running on batteries or solar power





13

Hes·so Cloud Continuum / Compute Continuum

One integrated tool to:

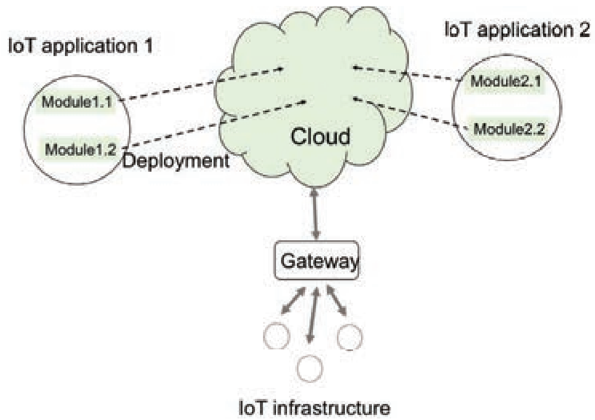
- Develop,
- Deploy (placement, match making),
- Monitor and control running

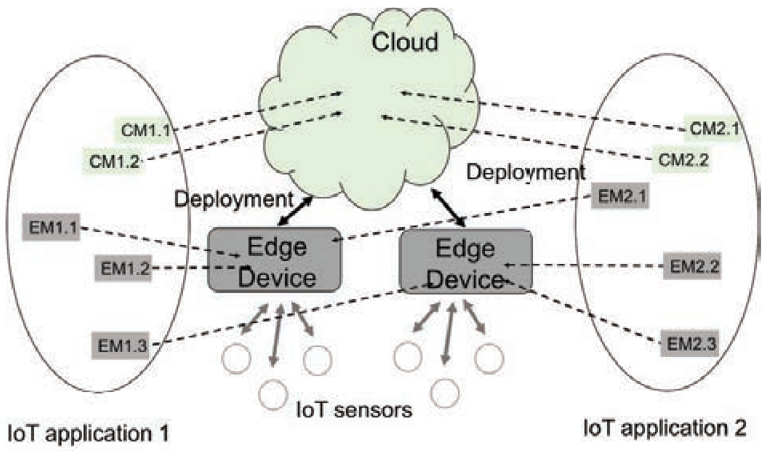
Of "Data and Compute sensitive" IoT applications
On large scale distributed platforms (HPC, Cloud, Edge devices, IoT sensors)

14

Cloud Continuum is also about software engineering

From a centralised to a decentralised paradigm





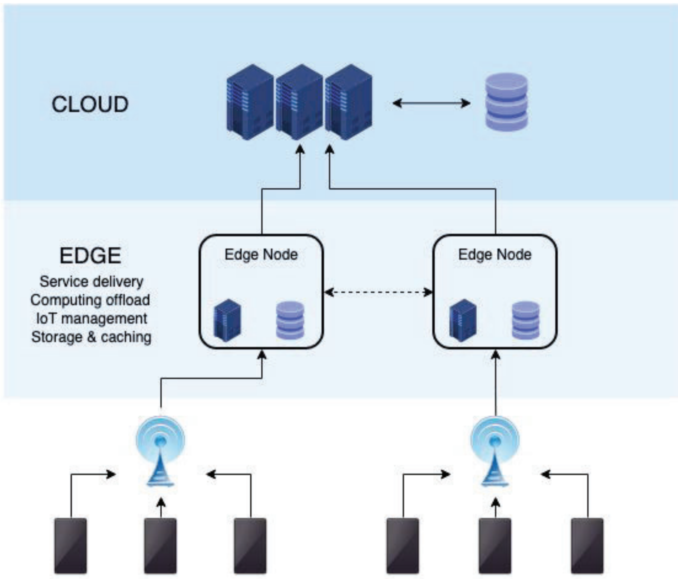
- AWS IoT Core
- Azure IoT hub
- ~~Cloud IoT Core (Google)~~

- Edge-to-Cloud Computing

15

What is Edge Computing ?

- A method of optimising applications by taking some "portions" away from central nodes to the other extreme (the "edge").
- A practice of processing data near the edge of the network, where the data is being generated



16

Hes·so

Why Edge Computing ?

Four objectives are behind Edge computing:

- Limit the traffic between IoT devices and the cloud
- Keep decision as close as possible to the IoT devices
- Enhance security
- Unload the cloud.

17

Hes·so Why Edge Computing ?

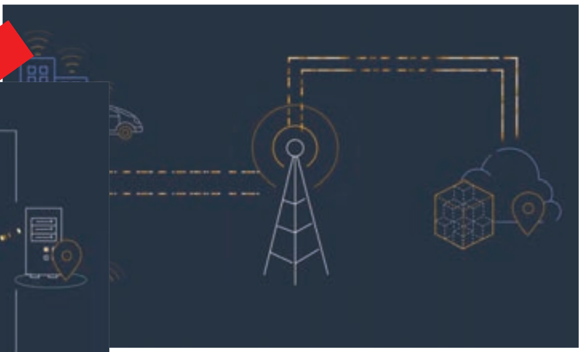
If the answer is **yes** to one or more of these questions, then you need an **intelligent edge**:

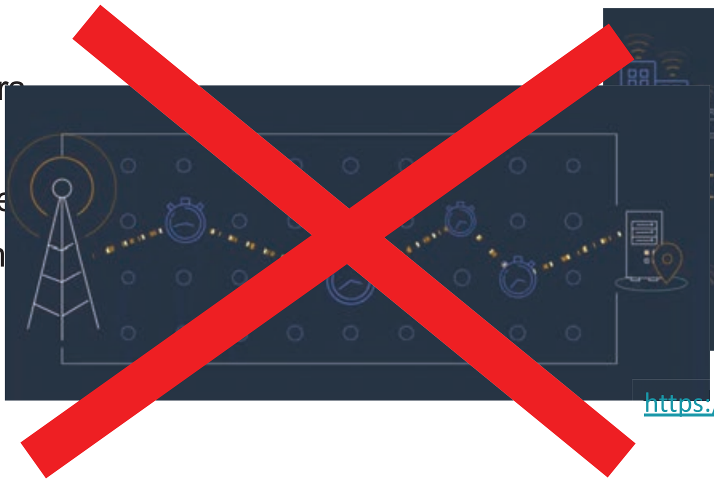
- Is there a need for **near-real-time action** on data collected by sensors?
- Is the **data** generated **too big to transfer** to the cloud?
- Is the **internet link** between the sensors/actuators and the cloud **unreliable** ?
- Is there a **privacy/security issue** with transferring or processing the data in the cloud (public or private)?

18

AWS wavelength zones

- AWS Wavelength is an AWS Infrastructure offering optimized for mobile edge computing applications.
- Application traffic from 5G devices can reach application servers running in Wavelength Zones without leaving the telecommunications network
- Take full advantage of the latency and bandwidth benefits offered by modern 5G networks.
- **Wavelength zones are associated with 5G providers**
- Use-cases
  - Connected cars
  - AR/VR
  - ML assisted he
  - Real time gam
  - …





https://aws.amazon.com/wavelength/

Hes·so Fog Computing

- Edge computing usually takes place directly on the edge device to which the sensors are attached.
- Fog computing moves the edge computing activities to a set of edge devices that belong to the same LAN
- In Fog Computing, computing activities are physically more distant from the sensors and actuators.

20

Hes·so "Side effects" of Edge Computing

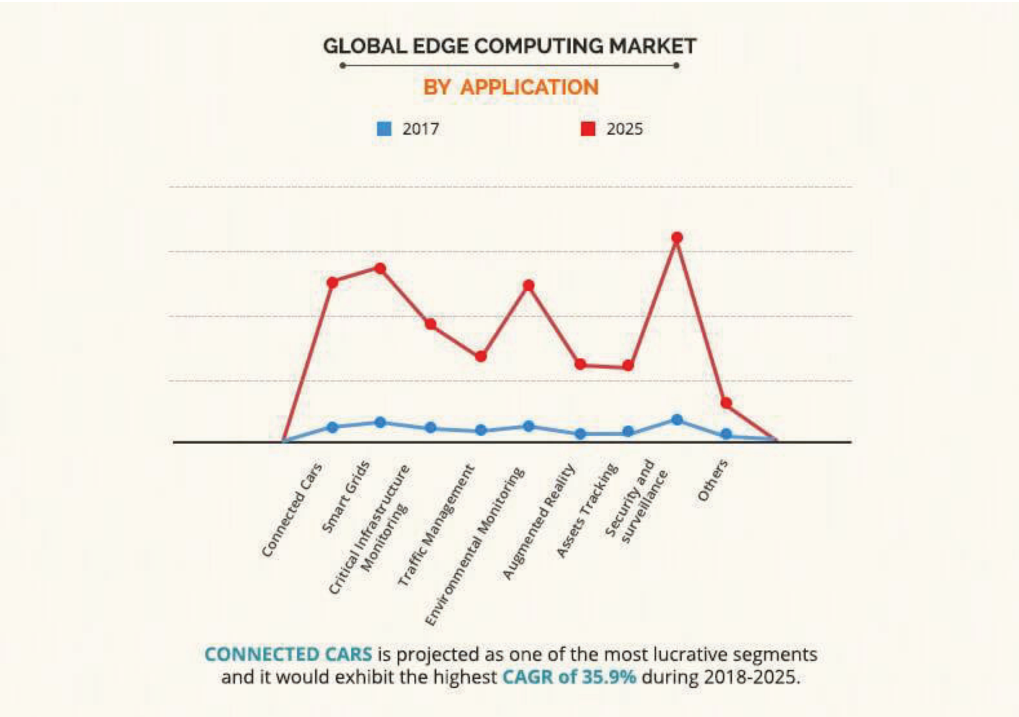
Strike the balance between:

- "keeping data at the edge" and "bringing it into a central cloud"
- "sophisticated algorithms in the cloud" and "lightweight analytical processes" in the edge

21

Hes·so Edge Computing market

https://www.alliedmarketresearch.com/edge-computing-market

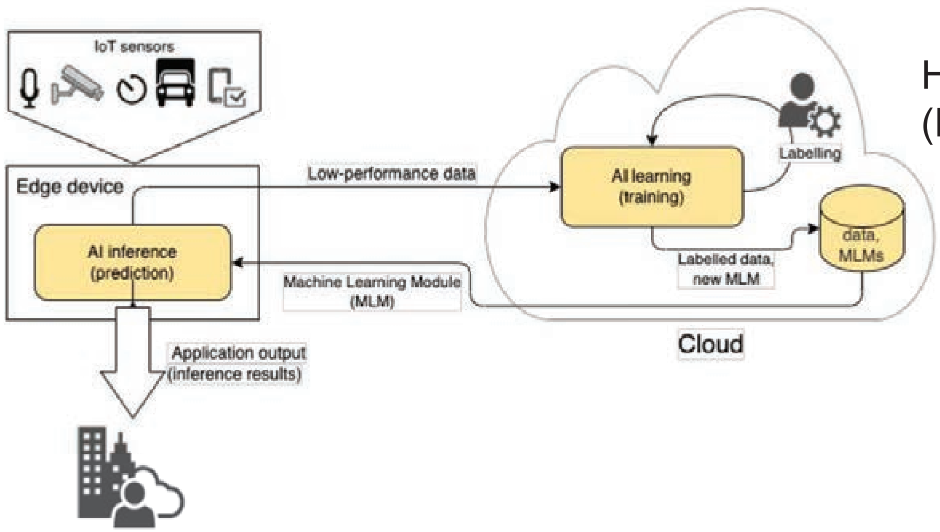


Hes·so Plan

- What is Cloud Continuum ?
- Why Continuum Computing ?
- **Use-cases**
- Cloud Continuum (Edge-to-Cloud solutions) solutions
- The Smart Grid Application

23

What is Self-adaptive ML based applications ?



Human in the Loop (HITL)

Cloud Continuum refers to the development, deployment and execution of self-adaptive machine learning-based IoT applications.

24

Hes·so

Use-case 1: Smart public lighting

MEDINA

- Reduce energy consumption
- Reduce light pollution
- Provide meaningful information to citizens, policy makers & operations teams

Innovation project supported by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Innosuisse – Swiss Innovation Agency

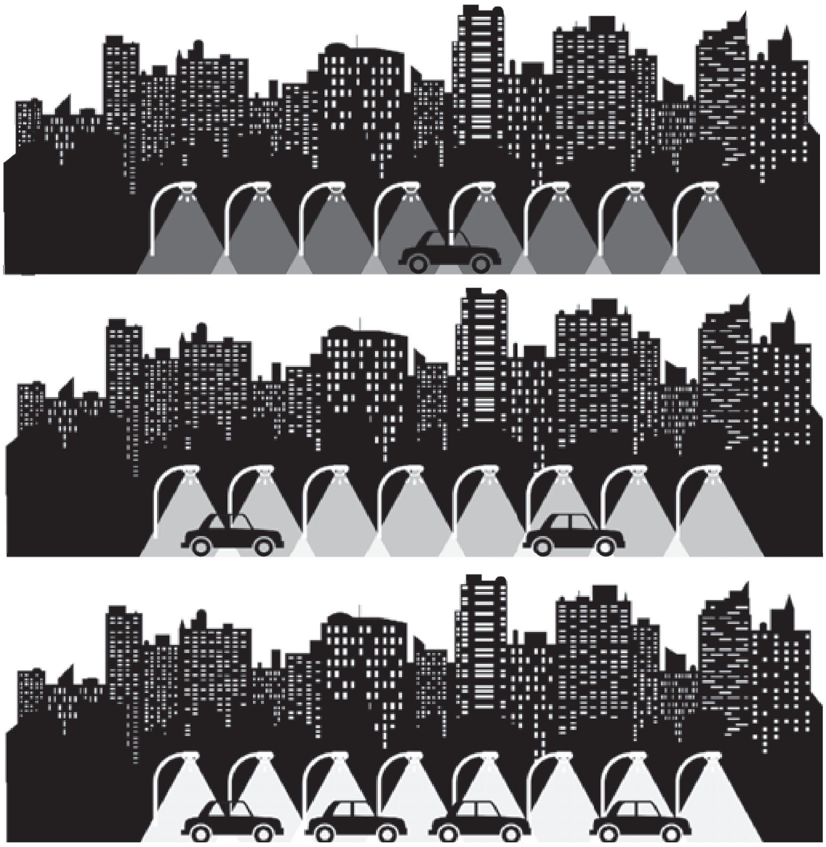
RÉPUBLIQUE ET CANTON DE GENÈVE

sixsq.

25

Hes·so

Use-case 1: Smart public lighting



Low light

Low traffic

Medium light

Medium traffic

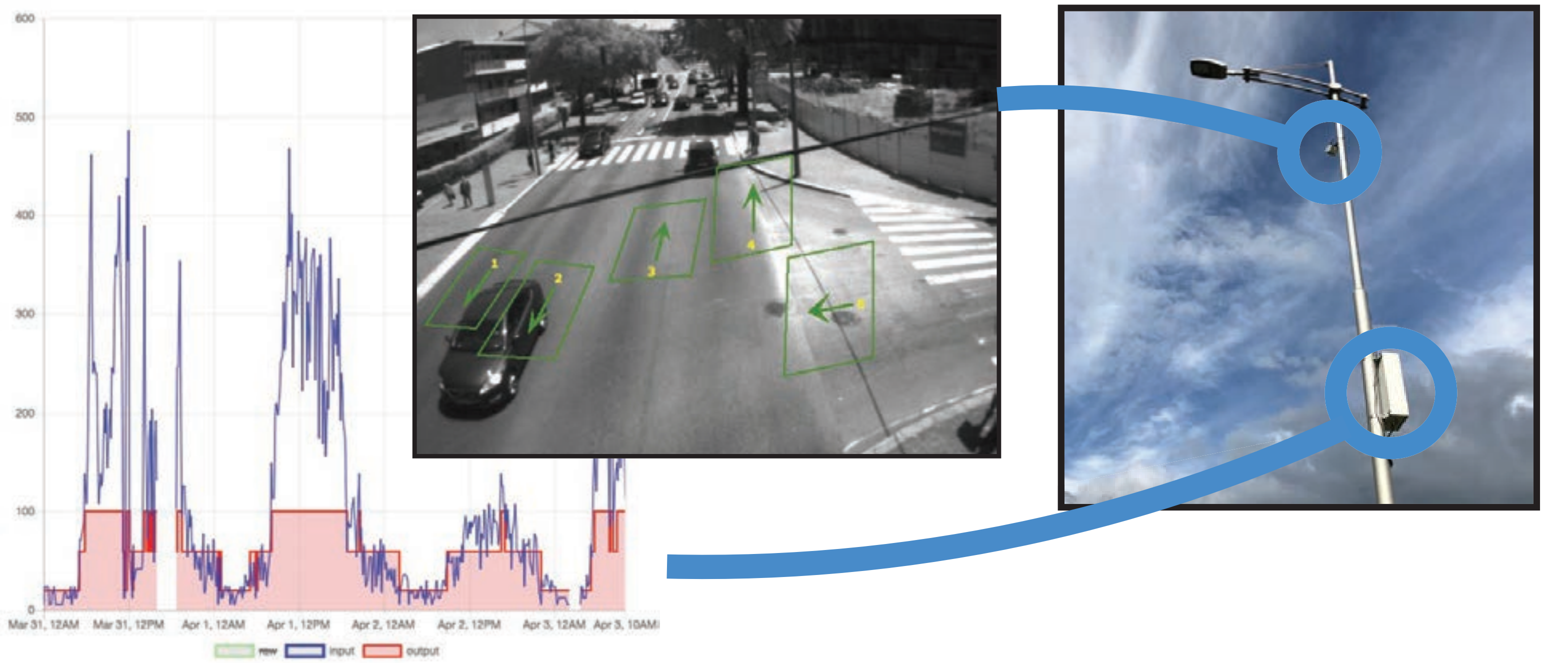
Full light

High traffic

26

Hes·so

Use-case 1: Smart public lighting



27

Hes·so

Use-case 1: Smart public lighting

28

Hes·so

Use-case 1: Smart public lighting

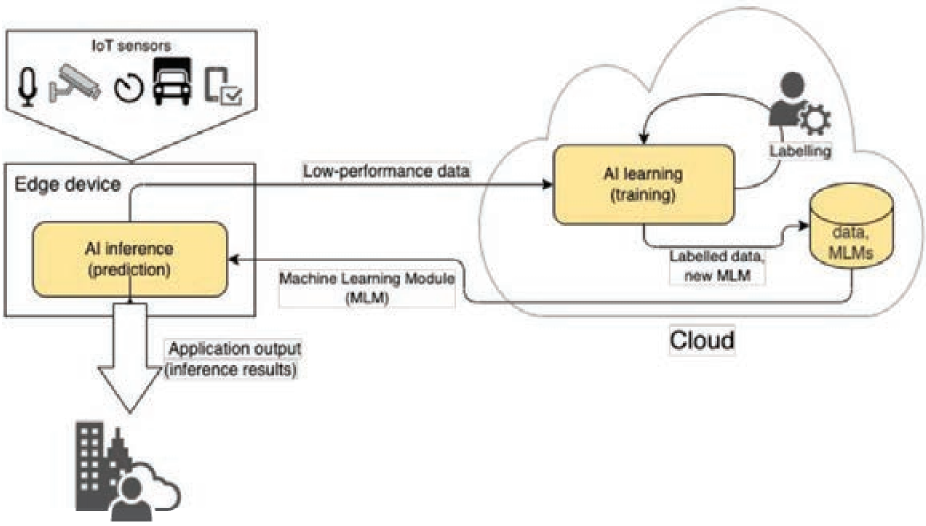
https://www.youtube.com/watch?v=ZygoAQ9ro-g&t=4s

29

Use-case 1: Smart public lighting

Two questions:

- How to recognise a bad "decision" ?
- What information the edge must forward to the cloud ?

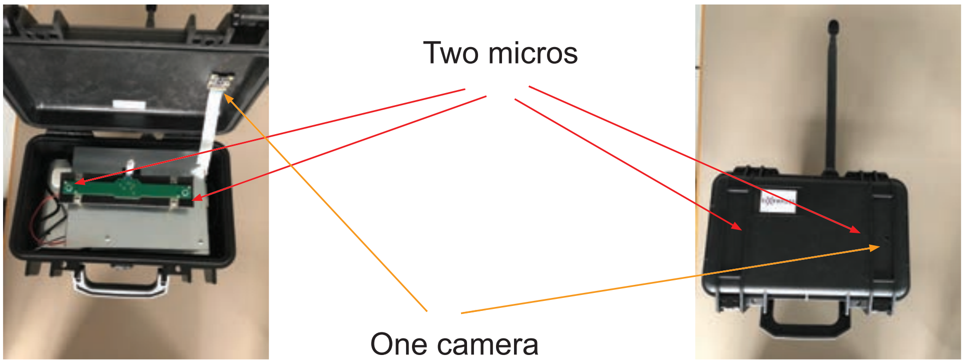


30

Hes·so

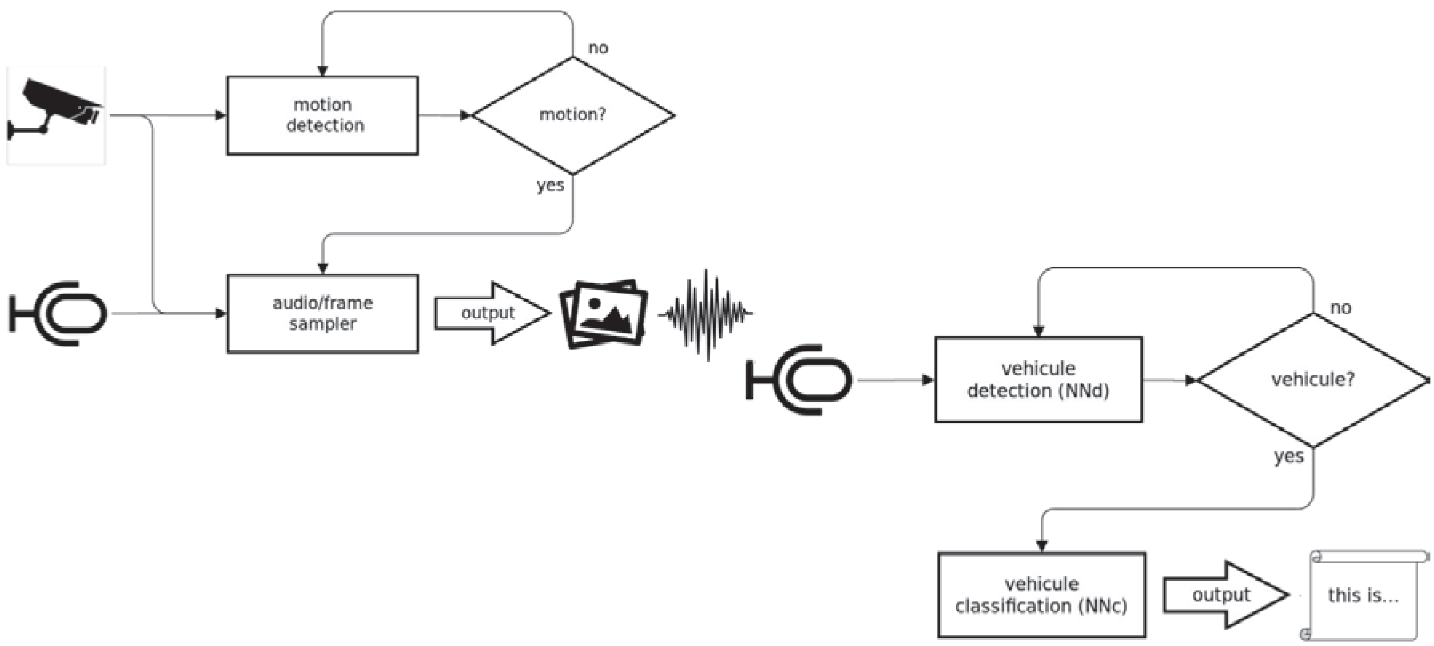
Use-case 2: Traffic noise monitoring

- A noise **"blind"** detector based on acoustic and AI technologies
    - No camera → No privacy problems
- Classifying vehicles (cars, truck, buses, motorcycles, electric cars)



31

Hes·so Use-case 2: Learning/Inference



32

Hes·so

Use-case 2: Learning/Inference

33

Hes·so

SECURAXIS

Use-case 2: Traffic noise monitoring

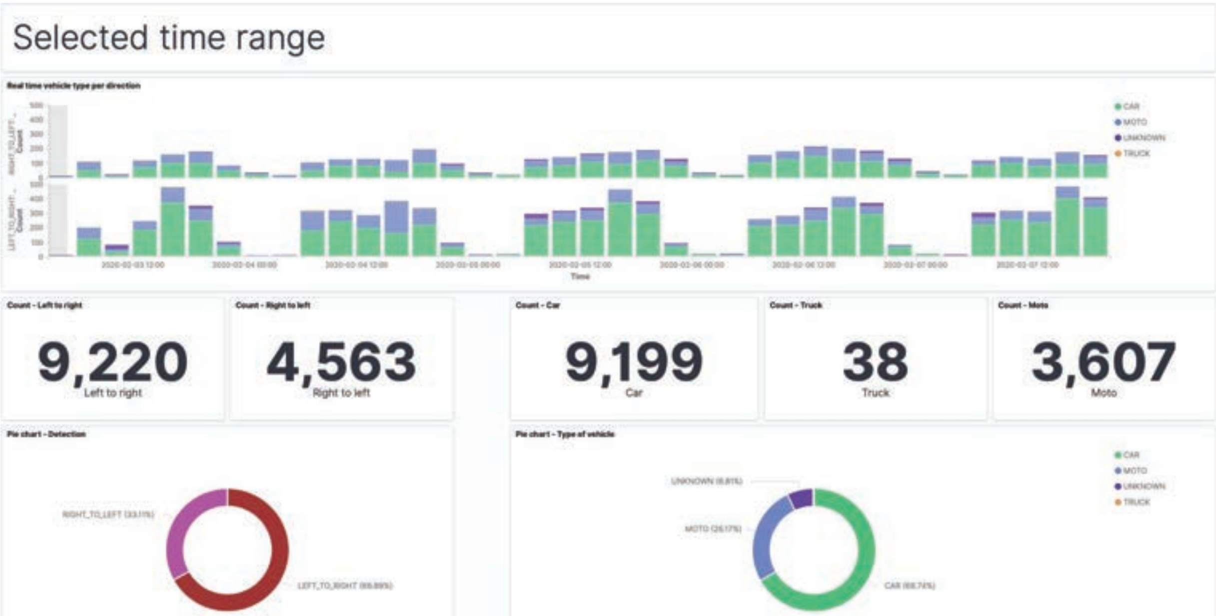
- Sept. 2019: A first prototype based on a raspberry deployed at HES-SO, HEPIA campus

34

Hes·so

Use-case 2: Traffic noise monitoring

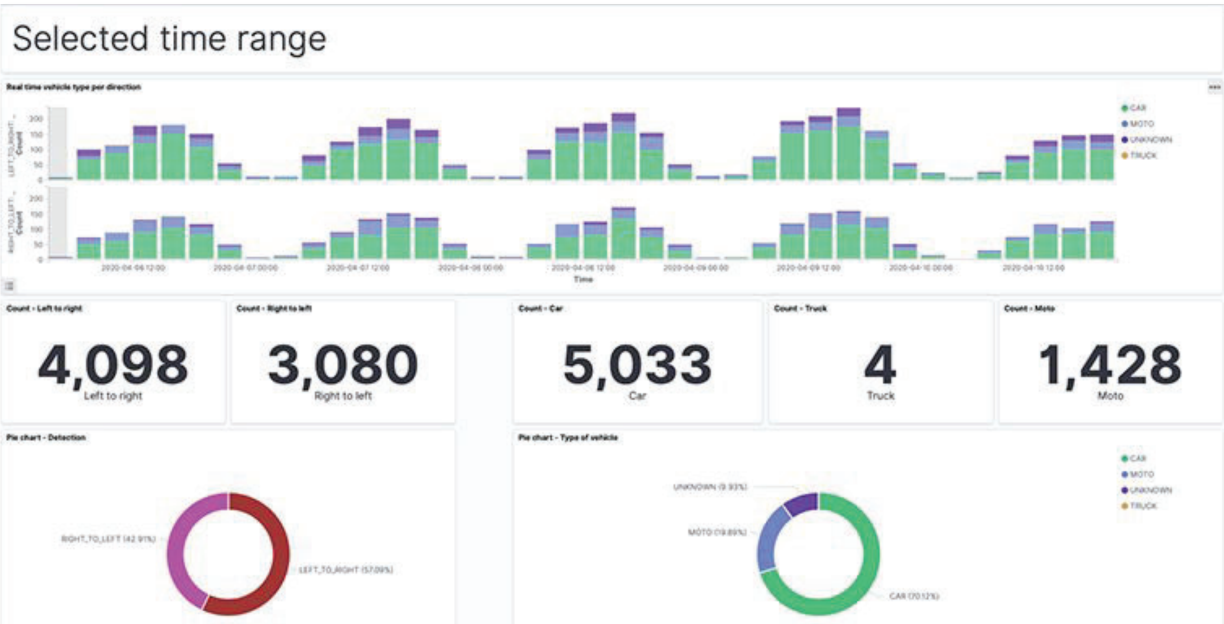
From Monday Feb. 3 to Friday Feb. 7 (2020): 13'783 vehicles



35

Use-case 2: Traffic noise monitoring

From Monday April. 6 to Friday 10 April (2020) : 7'178 vehicles



36

Hes·so

Use-case 2: Traffic noise monitoring

- March 2021: ***Noise Radar (NORA)***
  Innosuisse project: ,
  - FPGA based edge
  - The intelligence is context-aware (30 km zone, 1 lane, 2 lanes, 4 lanes, weather, snowing, rainy day, etc.)
  - "Remotely" control the intelligence in case of misbehaving sensors or switching from one context to another

Innovation project supported by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Innosuisse – Swiss Innovation Agency

EPFL

RÉPUBLIQUE ET CANTON DE GENÈVE

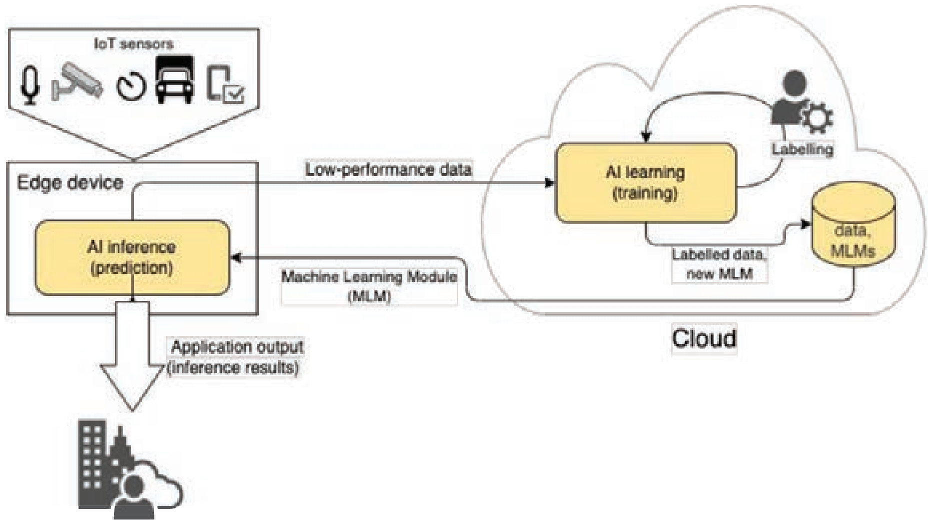
SECURAXIS

37

Use-case 2: Traffic noise monitoring

Two questions:

- How to recognise a bad "decision" ?
- What information the edge must forward to the cloud ?



38

Hes·so

Use-case 2: To summarise ….

https://www.youtube.com/watch?v=aZaObw-H0ac&t=39s

39

Hes·so

Use-case 3: Monitoring building heating systems

SG-ENERGIES

40

Hes·so

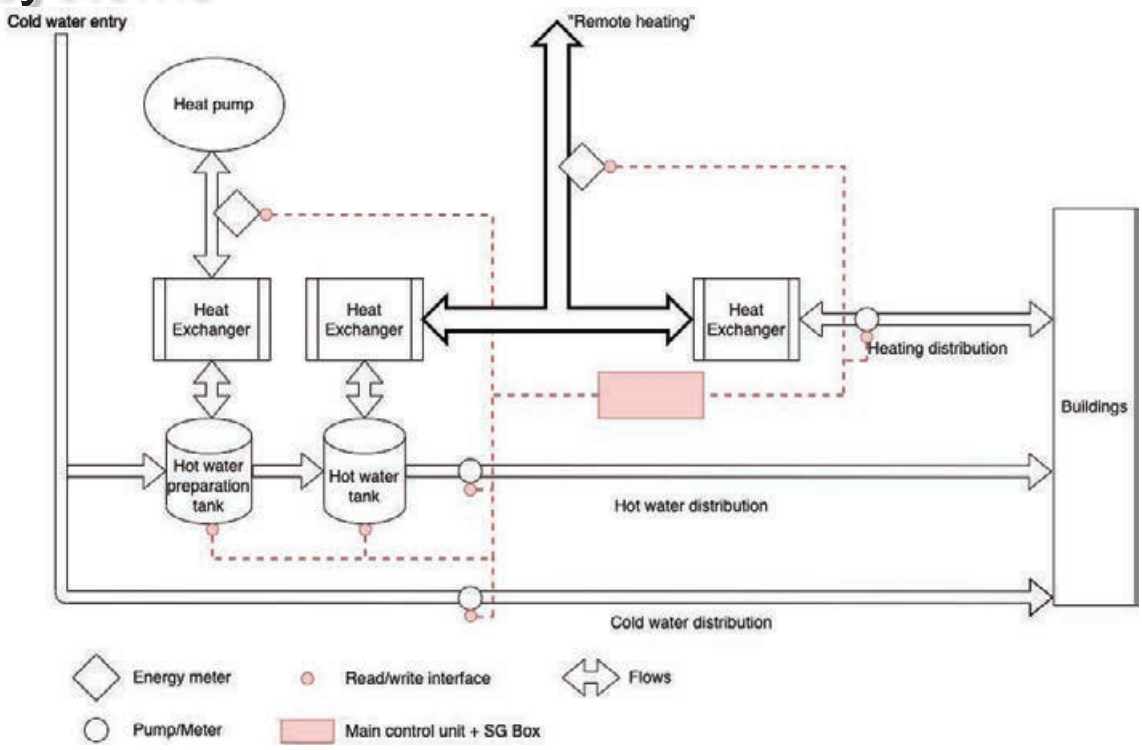
Use-case 3: Monitoring building heating systems

- SG-Box (BACnet, ModBus, LoRa, etc.)
- The goals are to:
  - collect data and take real time decisions
  - decrease the frequency and duration of breakdowns
  - decrease the on-site technical visits
  - In the near future, forecast (in order to control/optimise)

SG-ENERGIES

41

Use-case 3: Monitoring building heating systems



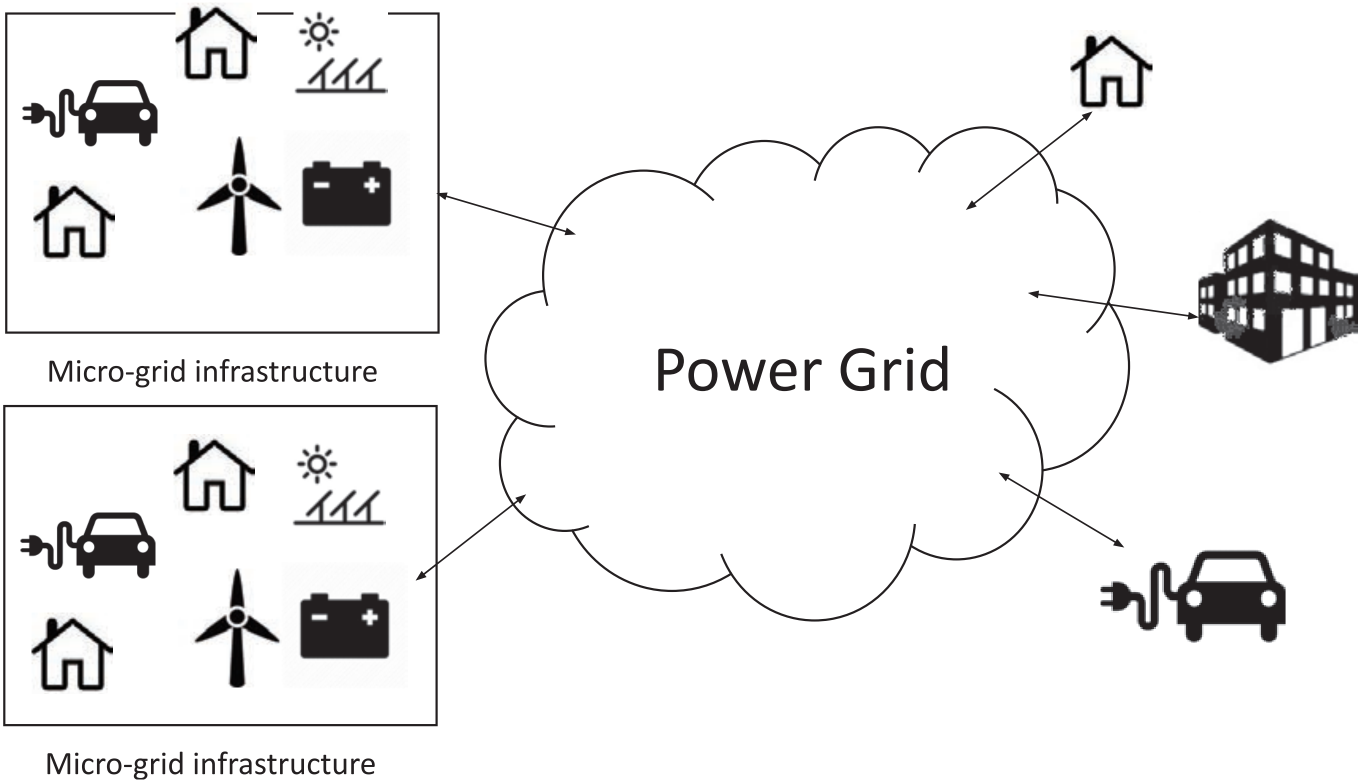
42

Use-case 3: Monitoring building heating systems

43

Hes·so

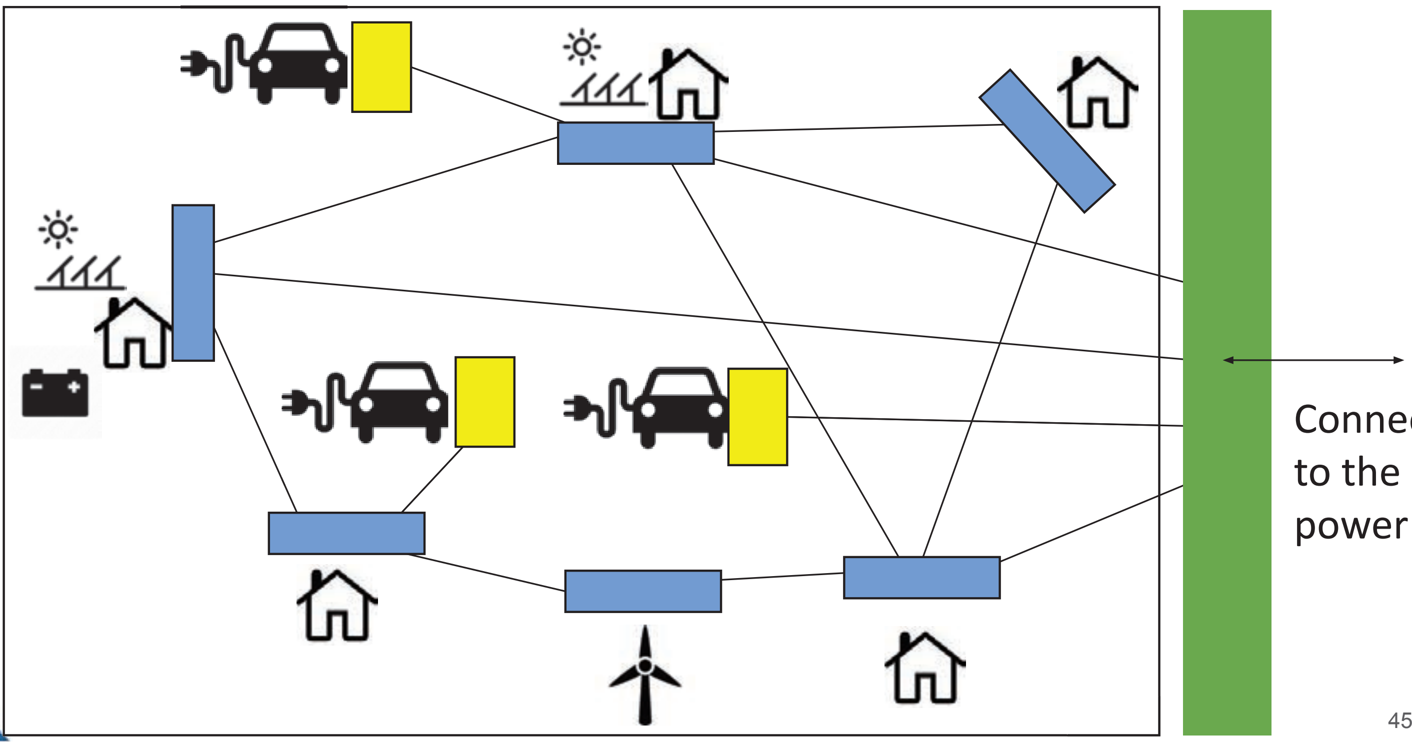
Use-case 4: Microgrid



44

Hes·so

Use-case 4: Microgrid



45

Hes·so

Use-case 4: To summarise ….

https://www.youtube.com/watch?v=mPMTgBPpZes&t=68s

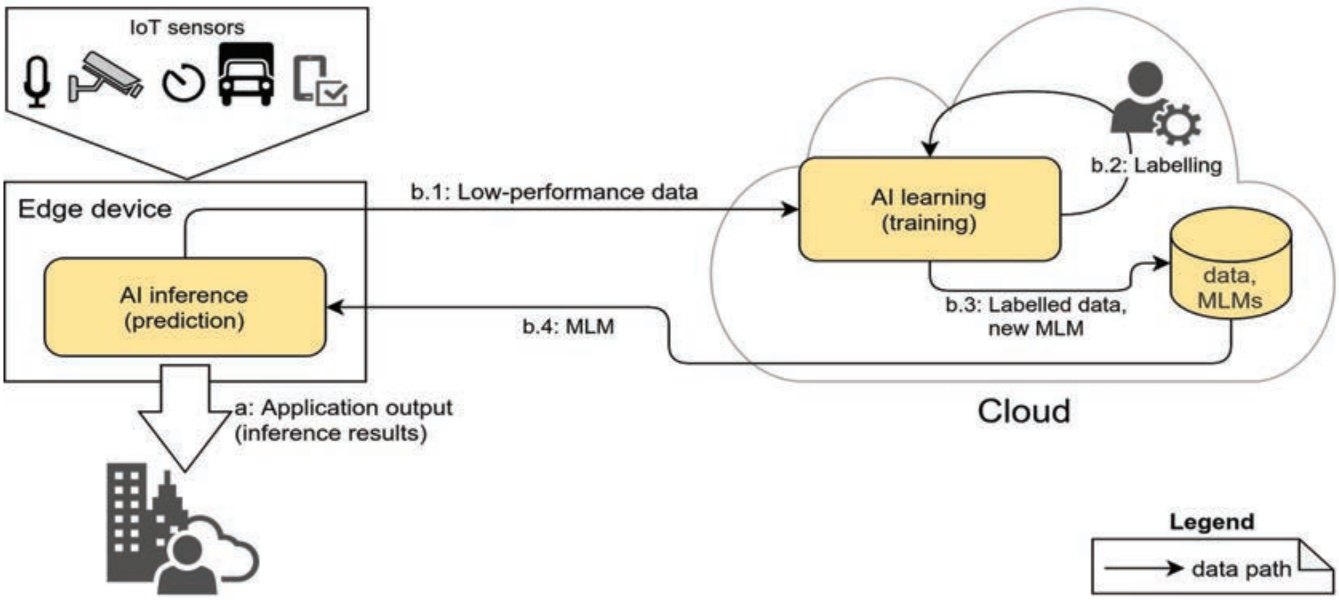
46

Hes·so Plan

- What is Cloud Continuum ?
- Why Continuum Computing ?
- Use-cases
- **Cloud Continuum (Edge-to-Cloud) solutions**
- The Smart Grid Application

47

Cloud Continuum: the closed loop (Self Adaptability)



48

Hes·so

Targeted applications

- IoT
- Highly distributed
- Context-aware
- Computing intense and data-sensitive

49

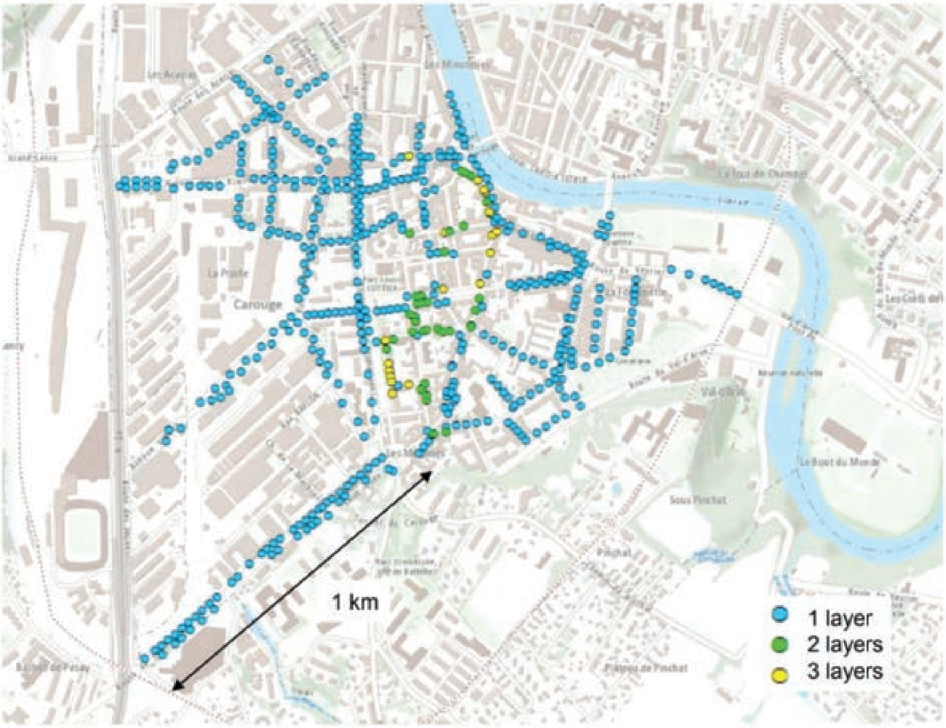
Hes·so What is a "context aware" applications ?

- Sensors deployed at a city scale are exposed to different and varying conditions that cannot be tackled with a unique inference (MLM) configuration.
- We assume here that intelligence (MLM) cannot be generic enough to efficiently handle all possible contexts.
- Groups (or clusters) of sensors are expected to belong to different spatio-temporal *contexts*, each needing its own MLM configuration

50

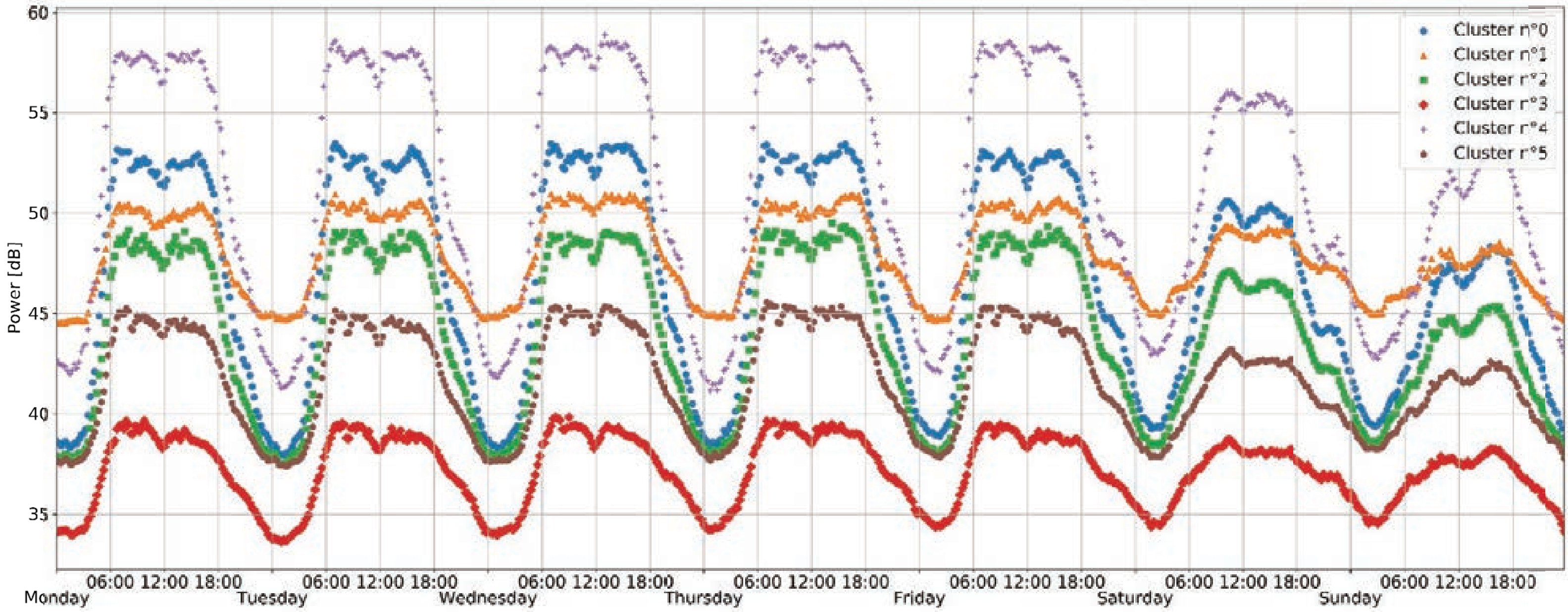
Context-awareness: Example 1

~550 LoRa noise sensors



51

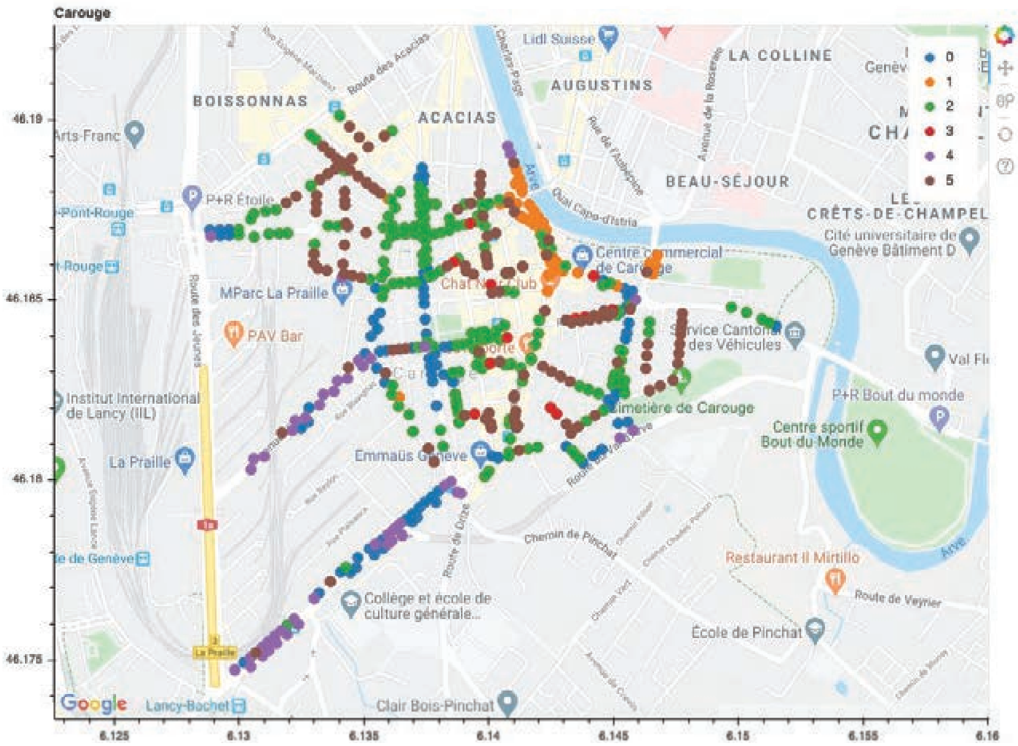
Hes·so Context-awareness: Example 1



52

Hes·so

Context-awareness: Example 1



53

Hes·so

Context-awareness: Example 2

Energy consumption/production

54

Hes·so

Three comparative criteria

- **Service level**
  - What services are supported by the edge-to-cloud solutions?
- **Application level**
  - Application level criteria: From "end-user" perspective
- **Operating Cost**
  - What is the deployment cost ?

55

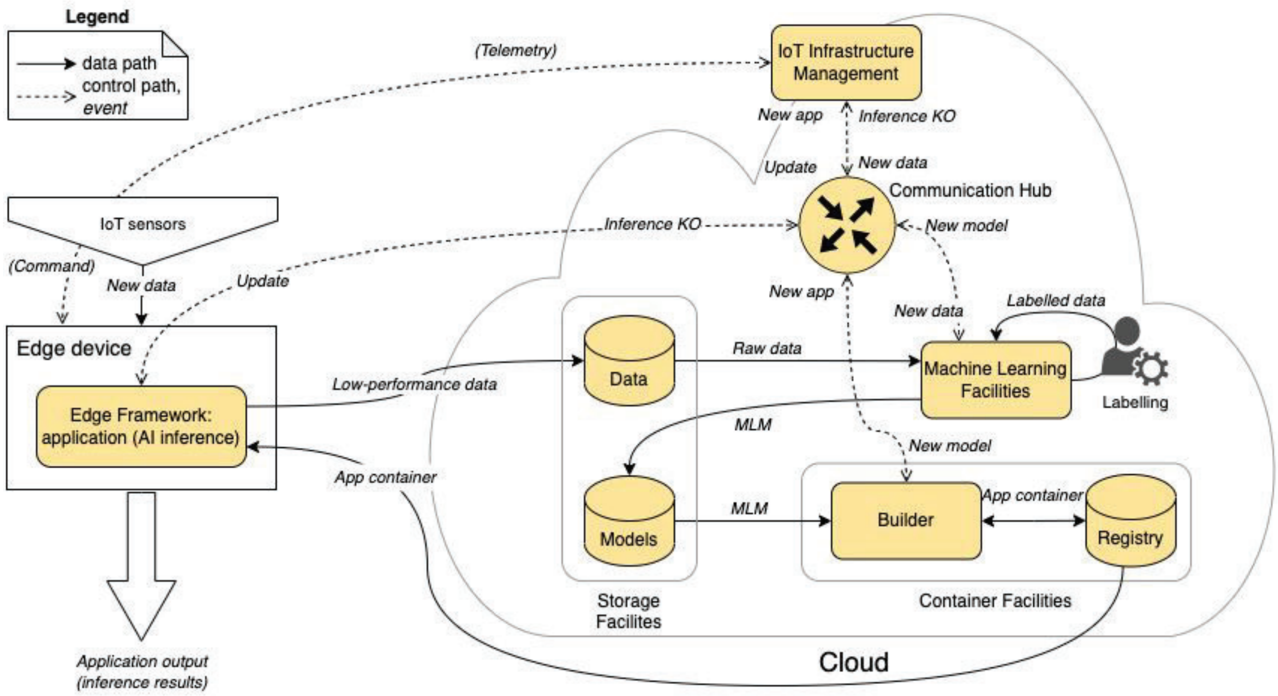
Hes·so

Anatomy of a Cloud Continuum solution (6 services)

1. **IoT Infrastructure Management**: An orchestration service that composes, provisions (configures and deploys) and monitors Edge/IoT networks.
2. **Edge Framework**: enable edge application modules programming and execution.
3. **Container Facilities**: build a *container* (such as Docker) with a trained ML Models (MLM)
4. **Communication Hub**: create a messaging infrastructure for networked components
5. **Storage Facilities**: store training data (labeled images in the MNIST case) and several MLM versions in (possibly different) Cloud data warehouses.
6. **Machine Learning Facilities**: build and train a MLM in the Cloud.

56

Hes·so Anatomy of a Cloud Continuum solution



57

Hes·so

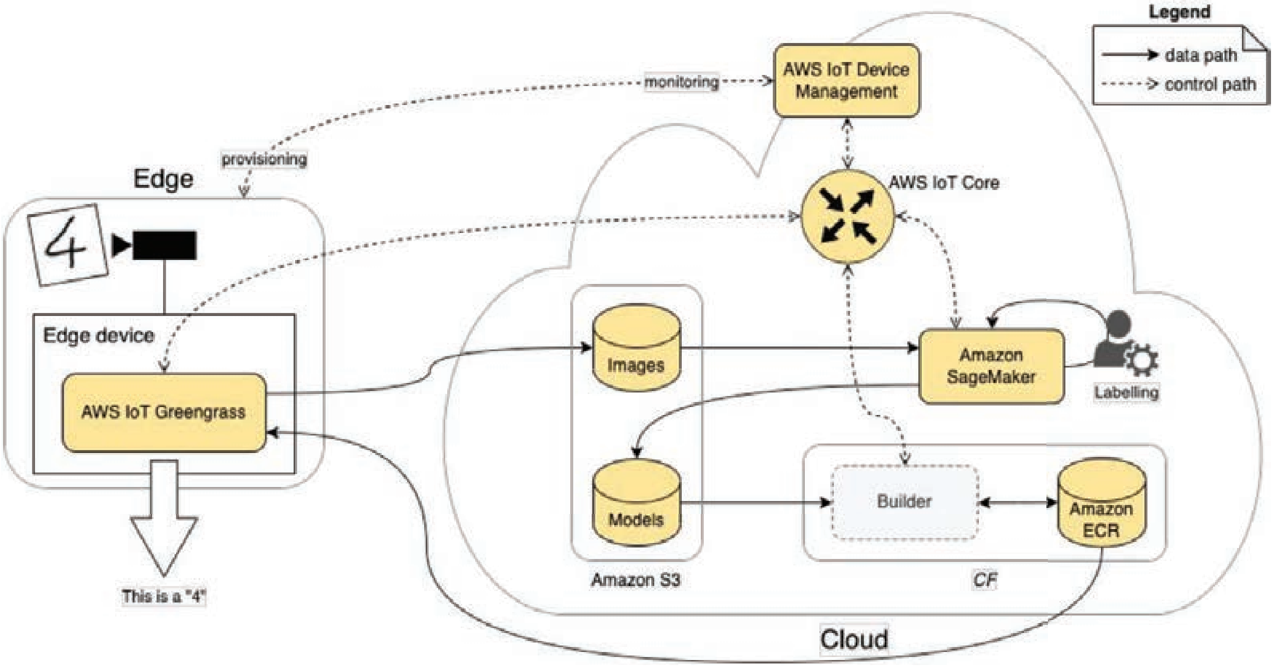
Anatomy of a Cloud Continuum solution

	IoT Infrastr. Manag.	Edge Framework	Container Facilities	Comm. Hub	Storage facilities	ML facilities
AWS Amazon	AWS IoT Device Manag.	GreenGrass	Elastic Container Registry (ECR)	AWS IoT Core	S3	Sagemaker
Azure	IoT Hub	IoT Edge Runtime	Azure Container Registry (ACR)	IoT Hub	Azure	Azure AI
Google	Cloud IoT Core	Coral Accelerator (HW)	Google Container Engine (GKE)	Pub/Sub	Cloud Storage (S3)	AutoML Vertex AI (Jan 24)
NuvlaEdge SixSq)	Nuvla	NuvlaEdge				
Balena	BalenaOS	balenaCloud				

58

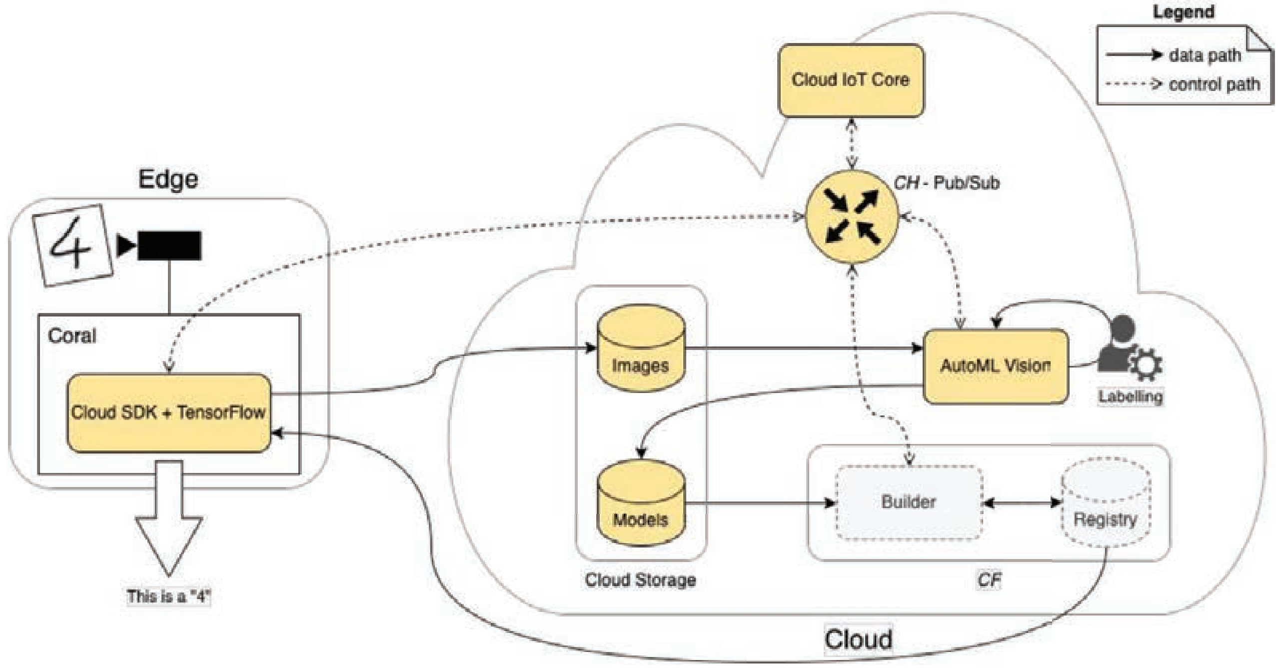
Hes·so

AWS Amazon solution



59

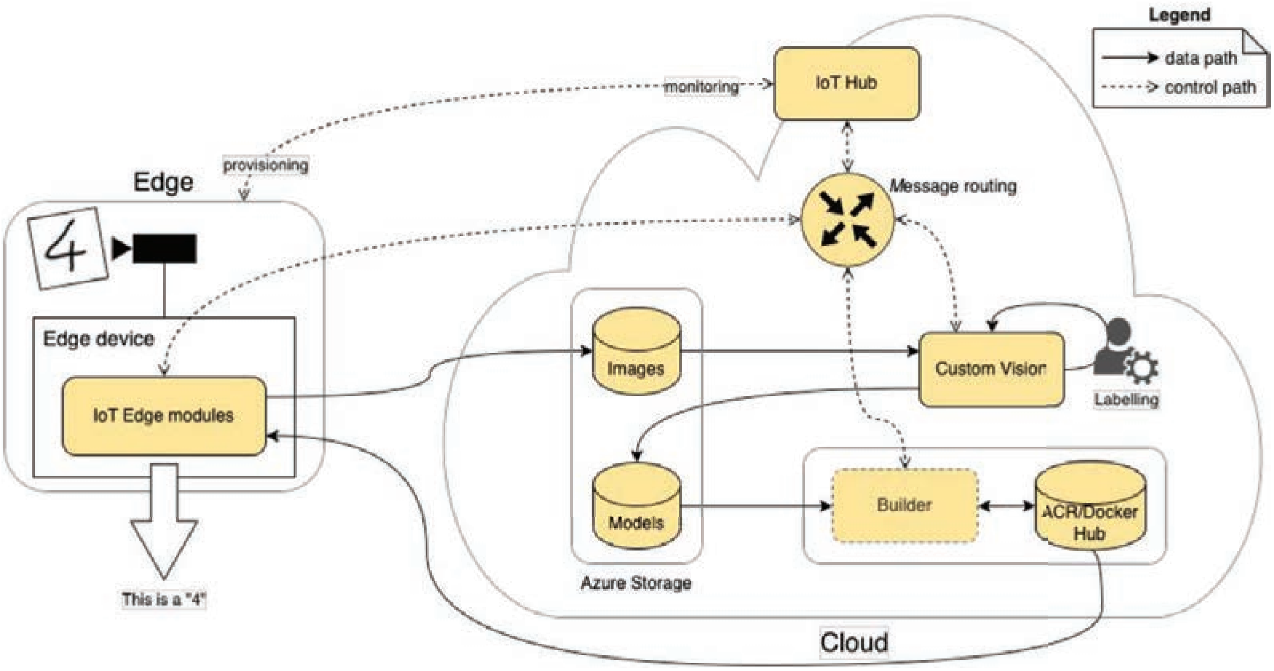
Google solution



60

Hes·so

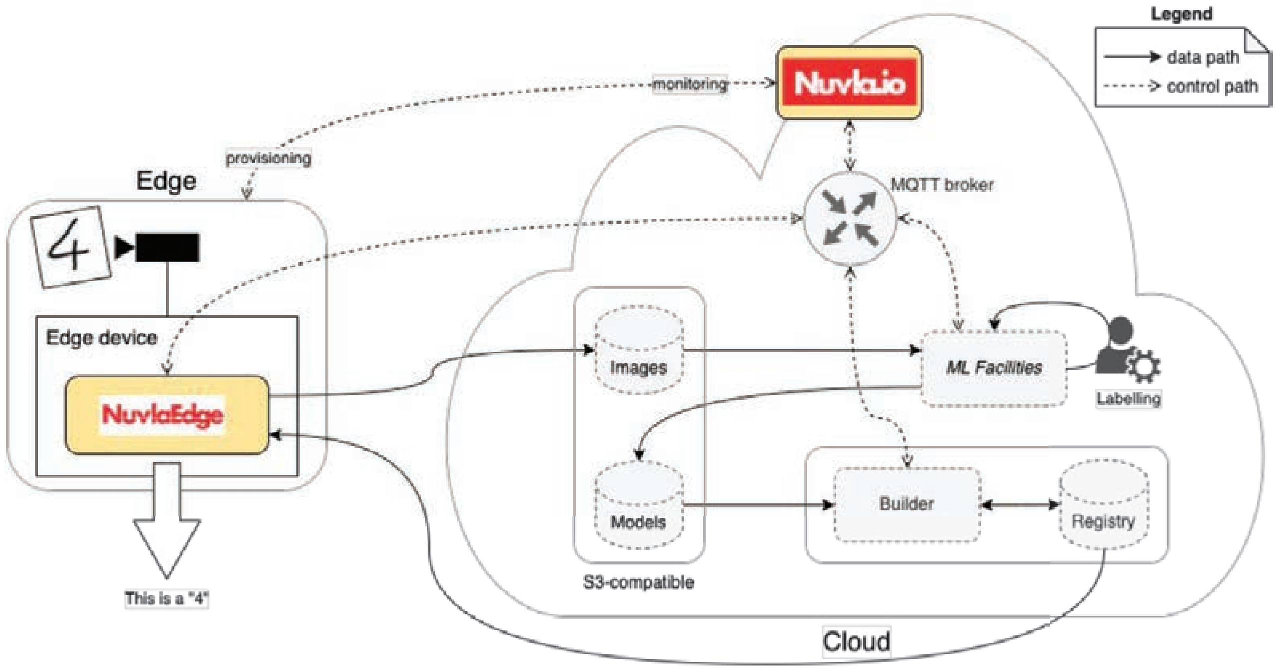
Azure solution



61

Hes·so

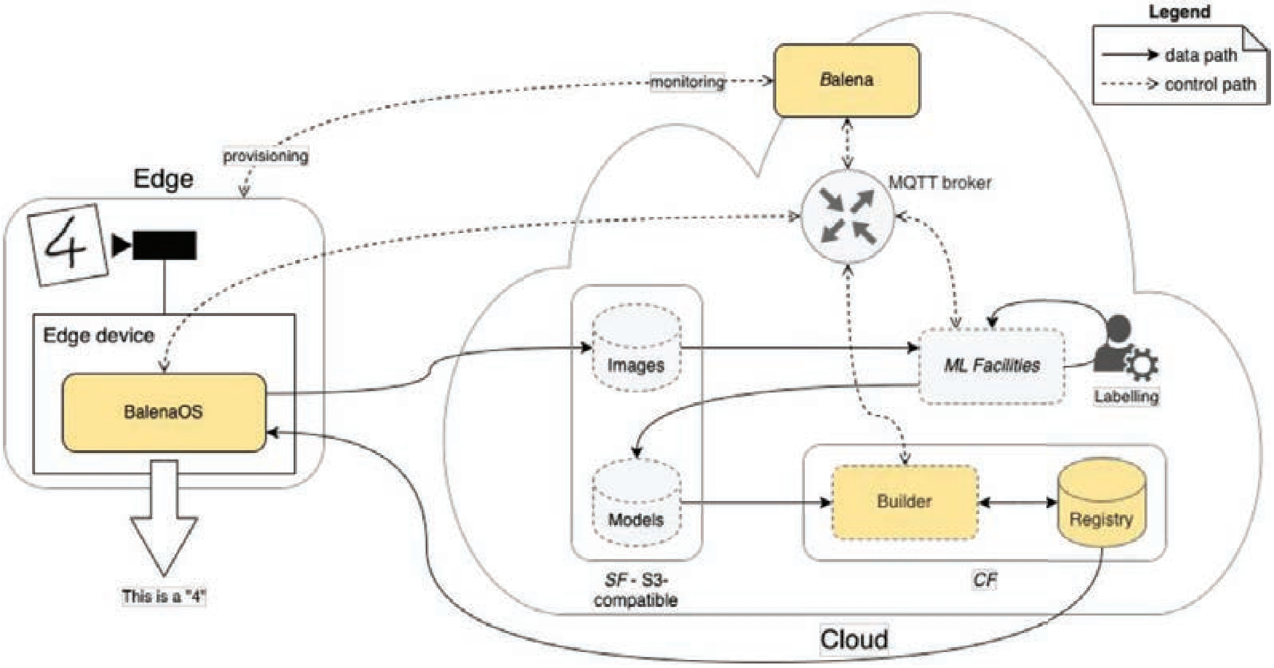
Nuvla/NuvlaEdge solution



62

Hes·so

Balena solution



63

Hes·so

To summarise …

	Amazon AWS	Google Cloud	MS Azure	SixSq Nuvla	Balena
IoT Infrastructure Management	provisioning monitoring	monitoring	provisioning monitoring	provisioning monitoring	provisioning ~monitoring: log watch only
Edge Framework[2]	serverless container (Docker-like)	~container (TensorFlow models only)	~serverless (via Docker) container (Docker-like)	container (Docker)	container (Docker-like)
Container Facilities	private/public registry	private registry	private/public Docker registry ~builder		private Docker registry ~ builder (balenaCloud only)
Communication Hub[3]	MQTT, HTTPS	MQTT, HTTPS	AMQP, MQTT	MQTT[4]	
Storage Facilities	S3	RESTful (JSON, XML), ~S3	RESTful (HTTPS), ~S3	(S3 indexing only)	
Machine Learning Facilities	labeling training	training (TensorFlow only)	~ labeling training		
Specialized Edge HW	proprietary, certified	proprietary	certified	certified	proprietary
Control Room Facilities	dashboard, CLI	dashboard, CLI	dashboard,CLI	dashboard, API	dashboard, CLI

64

Hes·so

To summarise …

- *Amazon AWS* and *MS Azure* are the most "complete" platforms, in terms of privately-integrated potential.
    - But this comes at the price of reduced flexibility
- *Nuvla* and *Balena* have their strength in the possibility of deploying *anything* on *any* cloud platform.
    - In a freedom-of-choice perspective, ***less* might be *more***.
- The *Google* solution is more a ML Models development rather than full edge application support

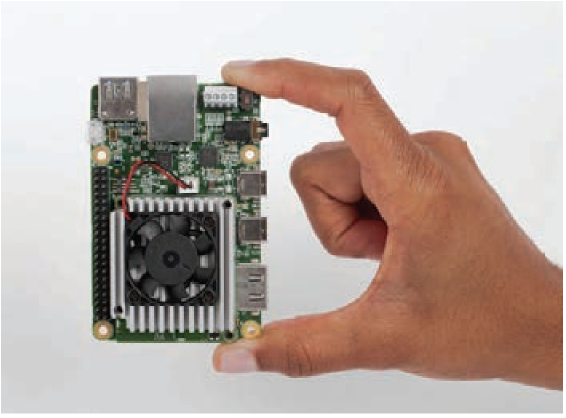
65

Hes·so Google Coral Dev Board

hepia
Haute école du paysage, d'ingénierie et d'architecture de Genève

- A single-board computer that contains an Edge TPU* coprocessor.
- Ideal for projects that demand fast on-device machine learning models.

- *: Tensor Processor Unit



Hes·so // GENÈVE

66

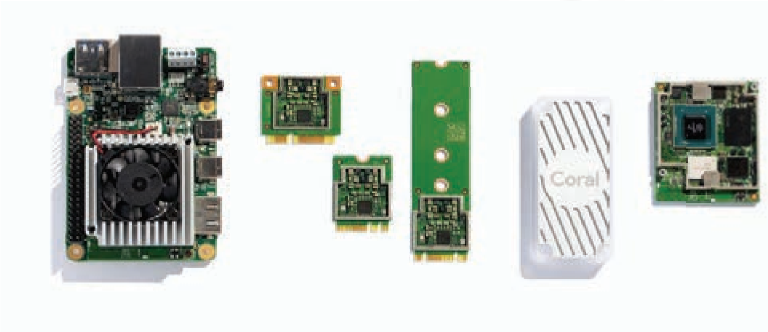
Hes·so

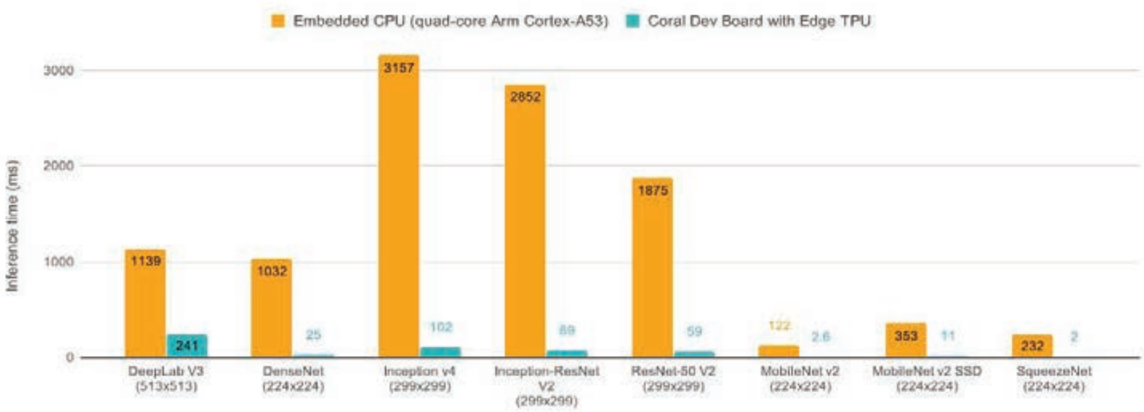
Coral device

hepia

Edge TPU coprocessor (ASIC)

TensorFlow Lite API





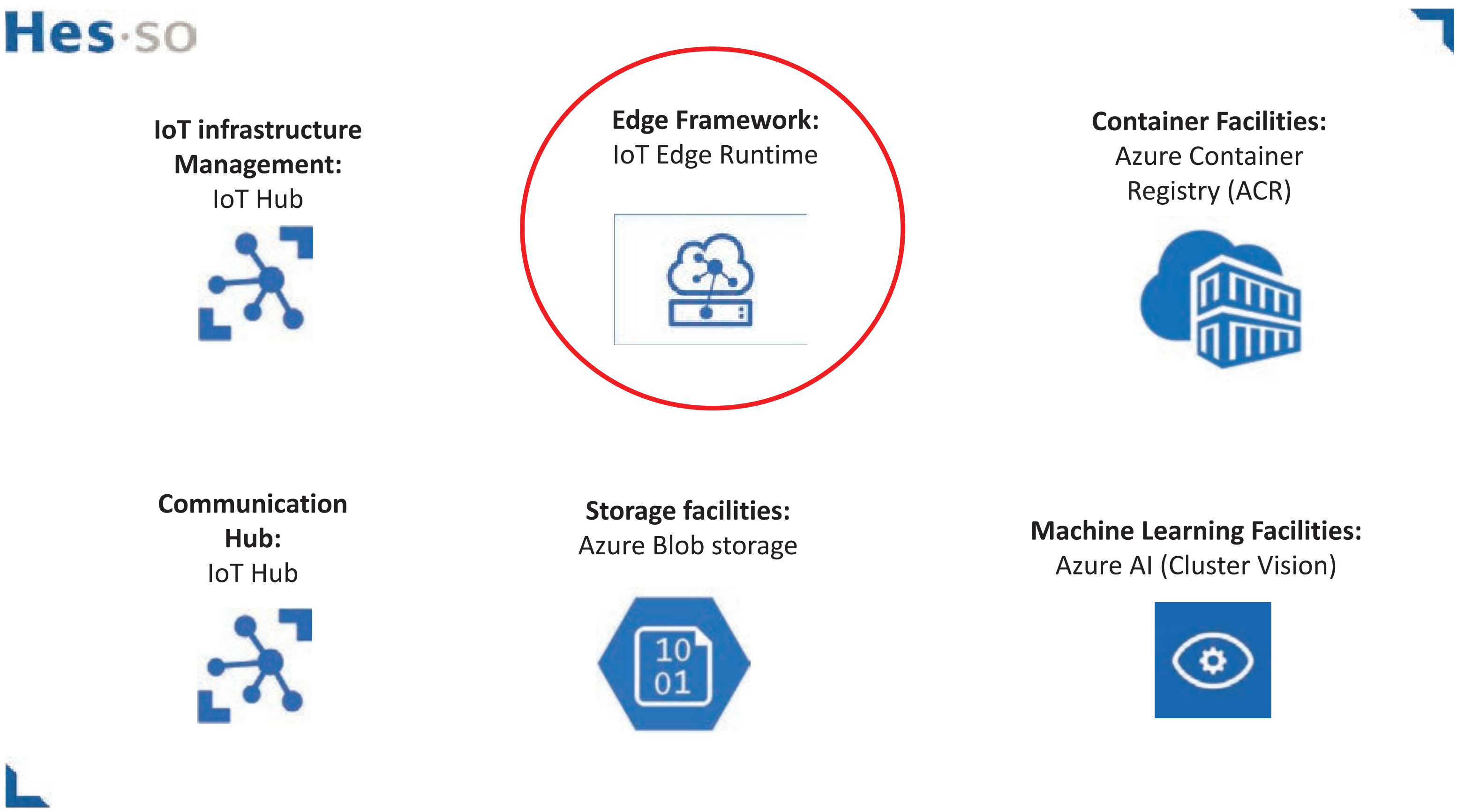
Hes·so GENÈVE

67

Hes·so

Azure IoT Edge Demo

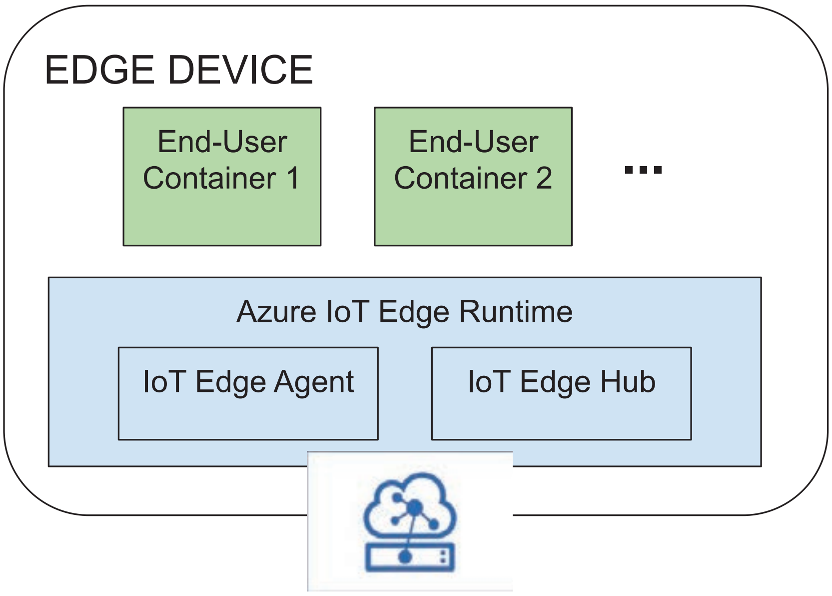
68


Hes·so
IoT infrastructure Management:
IoT Hub
Edge Framework:
IoT Edge Runtime
Container Facilities:
Azure Container Registry (ACR)
Communication Hub:
IoT Hub
Storage facilities:
Azure Blob storage
10
01
Machine Learning Facilities:
Azure AI (Cluster Vision)

Azure IoT Edge runtime

Two containers :

- IoT Edge Agent
- IoT Edge Hub



Hes·so

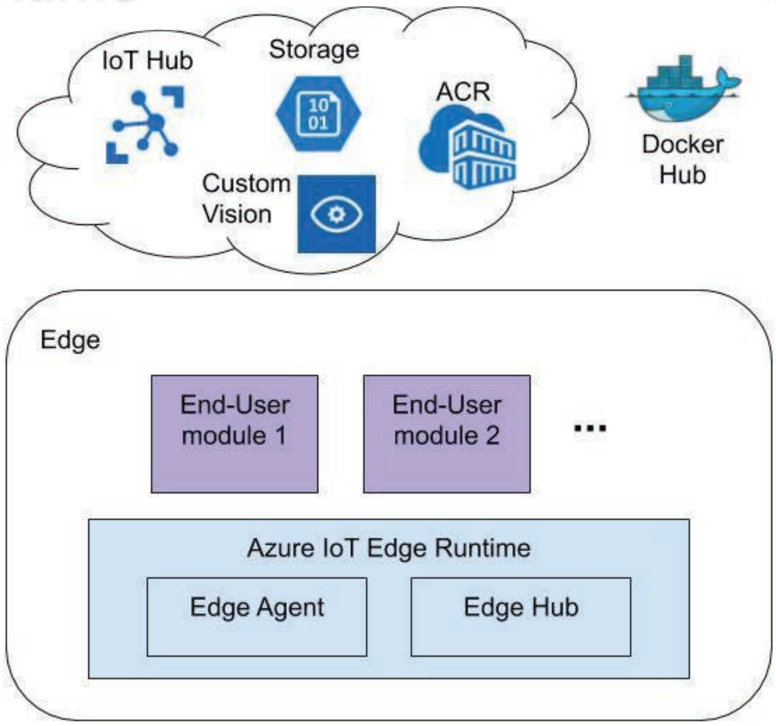
Azure IoT Edge runtime

Composed of two containers:

- IoT Edge Agent
- IoT Edge Hub

End-user containers:

- Module 1
- Module 2
- ...



71

Hes·so IoT Edge Hub

- Acts as a local proxy of IoT Hub
  - The clients can connect to the IoT Edge Hub just as they would to IoT Hub.
- Mimics Azure IoT Hub
  - It is for IoT devices what cloud is for Edge devices.
- Functionalities:
  - Authentication
  - Reducing bandwidth
  - Working offline
  - Module communication

72

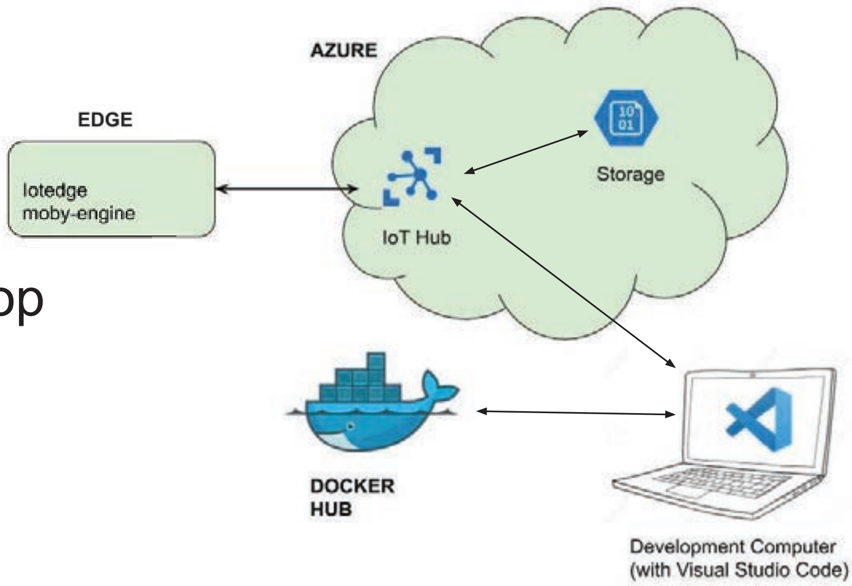
Hes·so IoT Edge Agent

- Instantiates modules
- Ensures that they continue to work
- Reports the status of the module back to the IoT Hub (Cloud)
- Retrieves the information from the deployment manifest

73

Configuration

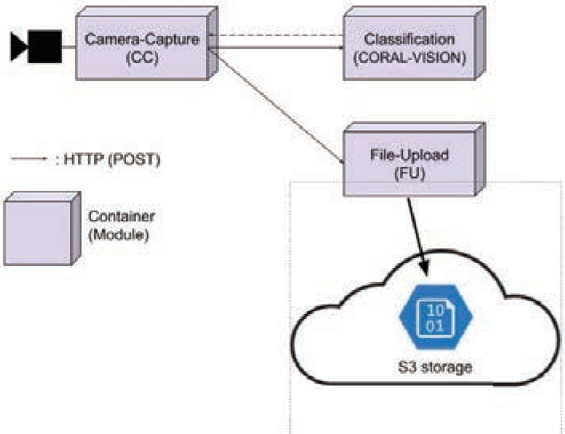
1. IoT Hub – laptop (Visual Studio Code)
2. IoT Hub - Storage
3. Registry container - laptop (Visual Studio Code)
4. Edge – IoT Hub
5. Custom Vision



74

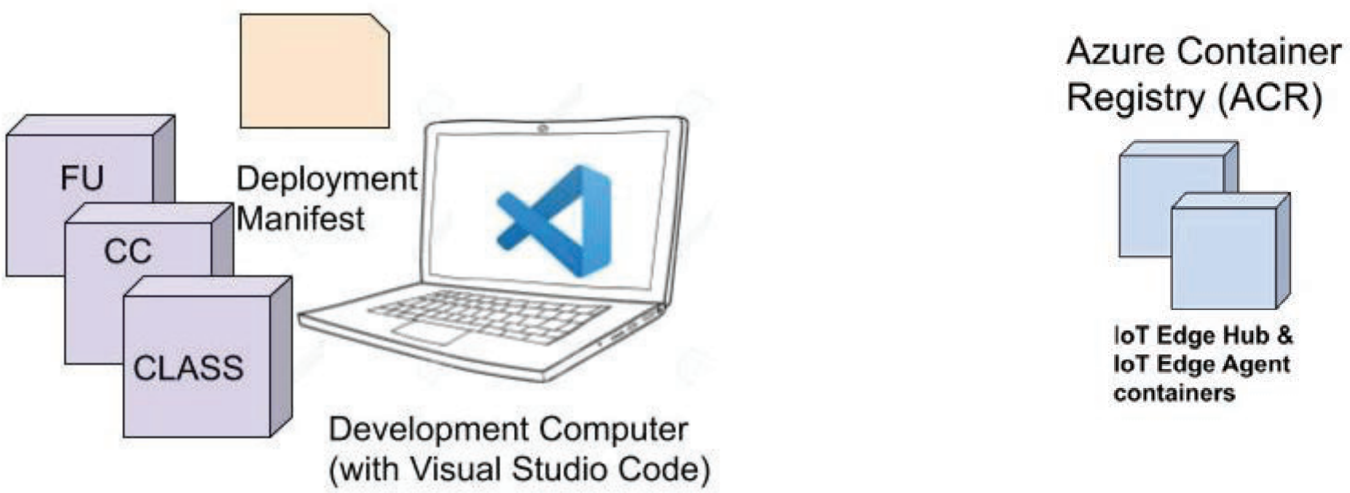
Application overview

- **CC** takes a frame and pre-process it
- **CC** sends to **CLASS** the frame, **CLASS** answers with an inference result
- In case of low performance data, **CC** sends the frame to **FU**
- **FU** sends the frame to the cloud Storage



75

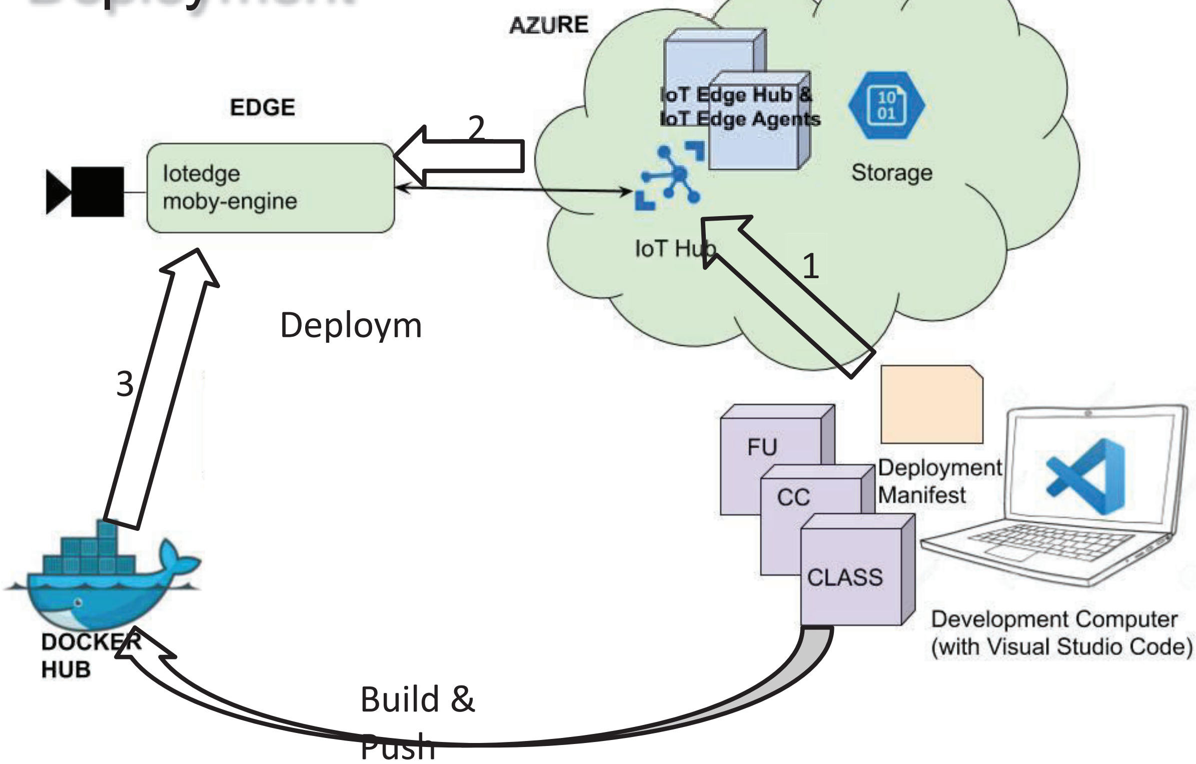
Hes·so Development & Deployment manifest



76

Hes·so

Deployment

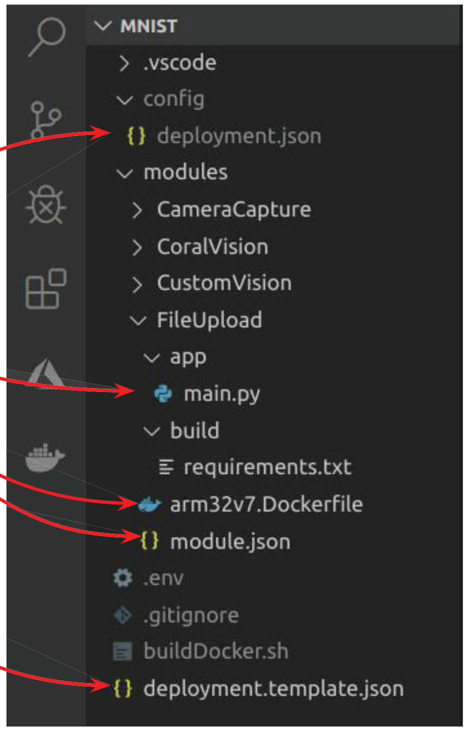


77

Hes·so Development (Visual Studio)

Files for IoT Edge Solution :

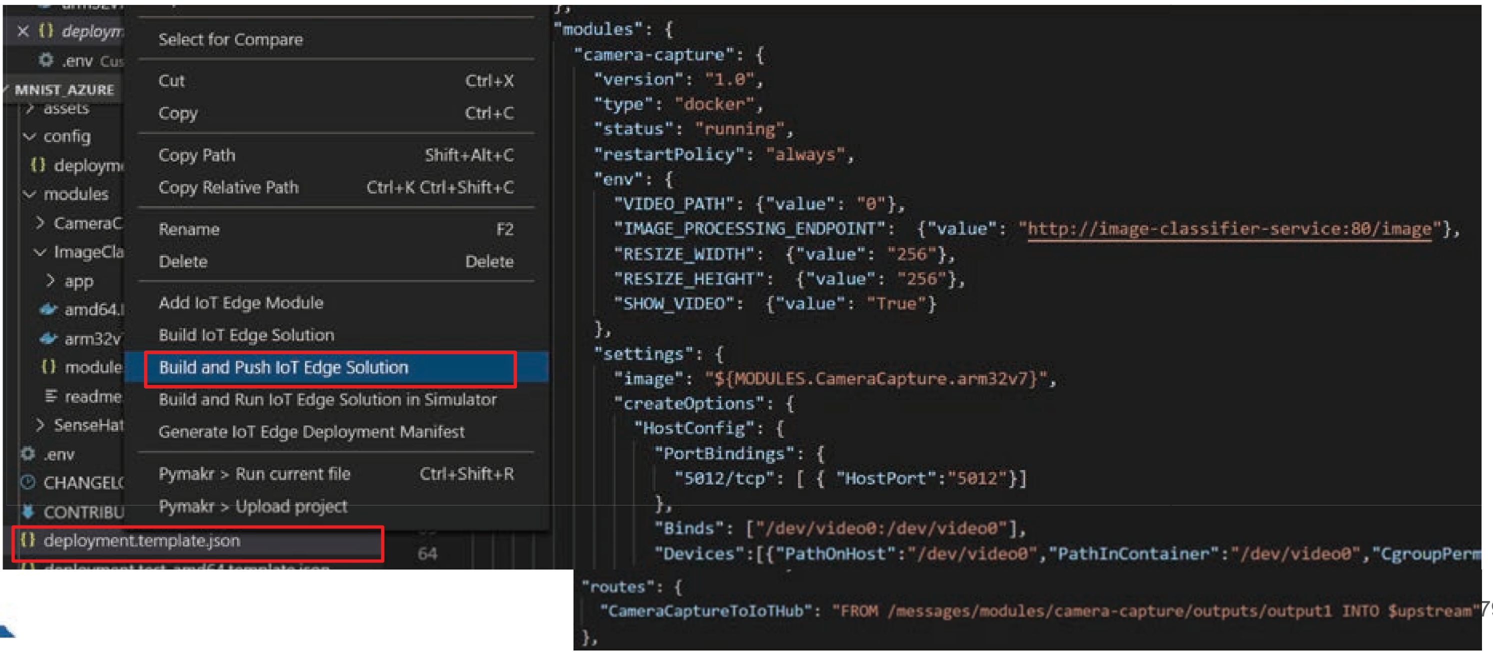
- Program code
- Dockerfile with cross-compile
- module.json
  repository URL, dockerfile, version ...
- deployment.template.json
  Build and Push IoT Edge Solution
- deployment.json
  generated by previous step
  Create Deployment



78

Deployment (Visual Studio)

Build and Push to Registry



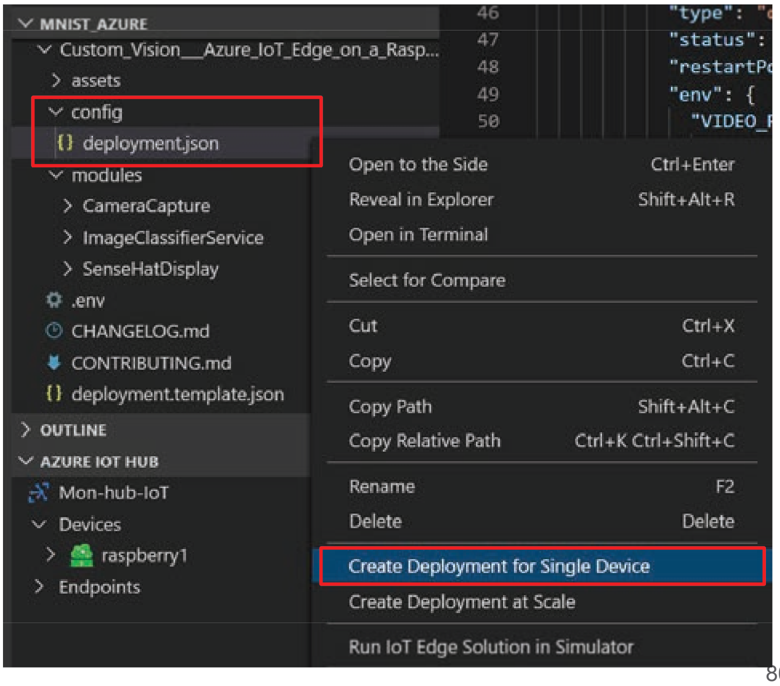
79

Hes·so Deployment (Visual Studio)

Deployment

File "config/deployment.json" generated by previous step

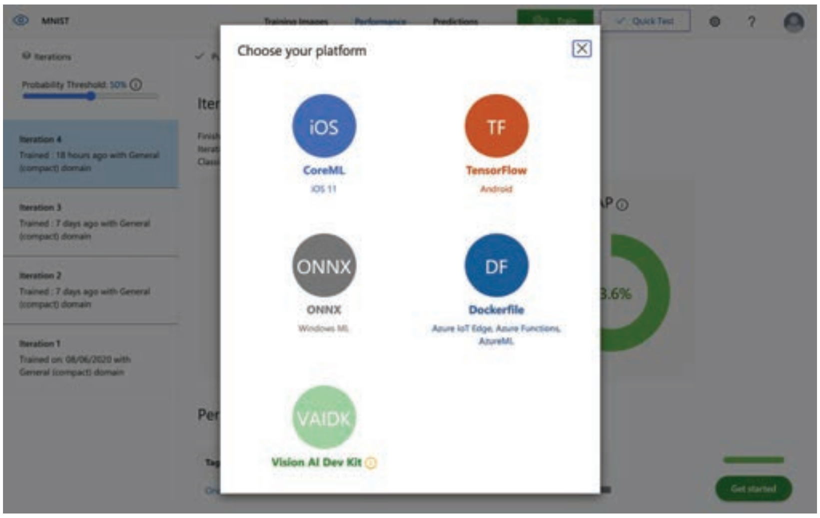
Send JSON file to edge device



80

Hes·so Custom Vision

- Easy way to create image classifier (CLASS)
- “Dockerfile” option: a flask REST server embedded in a docker container
- “TensorFlow” option:
  To “customize” your application



81

Hes·so Three comparative criteria

- Service level
- **Application level**
- Operating Cost

82

Hes·so

Application level

- **Licensing model.**
- **Hosting model.**
- **Hardware support.**
- **Documentation quality:**
- **Integration**
  - **Interoperability.**
  - **Components on premises.**
  - **Orchestration.**
  - **Monitoring, logging and telemetry.**

Scan me!

83

Hes·so

Application level

- **Security**
  - **End-to-end encryption.**
  - **Identity and access management (IAM).**
  - **Integrity enforcement.**
  - **Controlled commissioning.**
- **Risks**
  - **Vendor lock-in**
  - **Security breaches**
  - **Fault tolerance and resilience:**

Scan me!

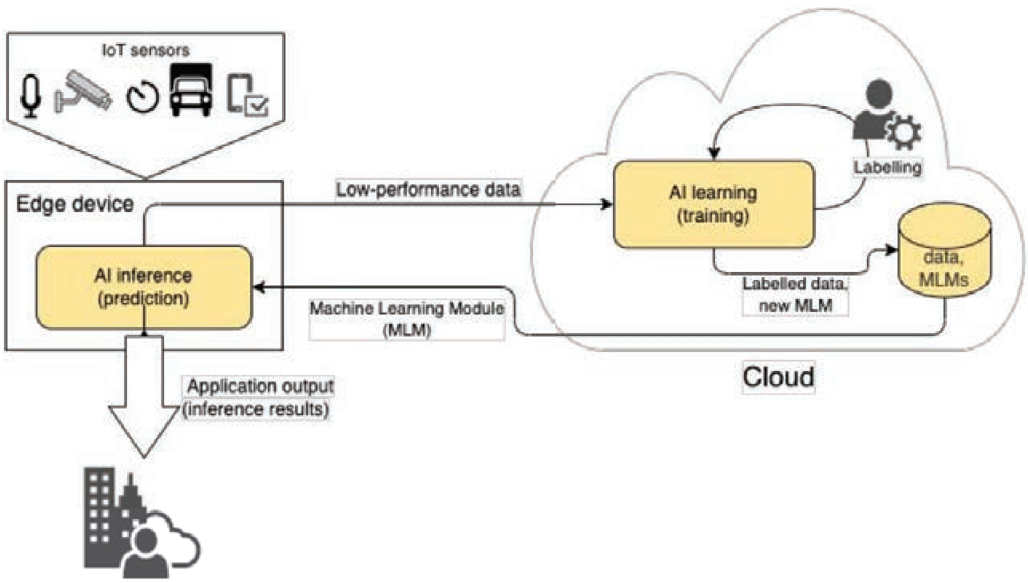
84

Three comparative criteria

- Service level
- Application level
- **Operating Cost**

85

A reminder: What is Self-adaptive ML based applications ?

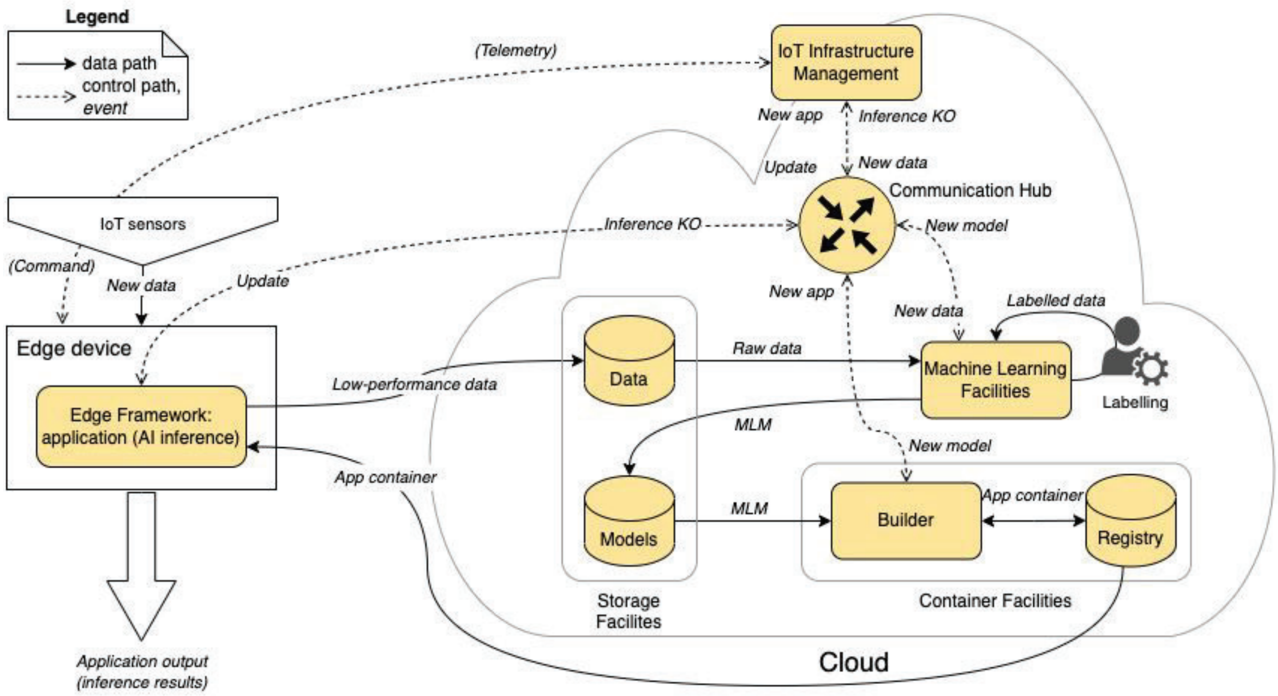


Human in the Loop
(HITL)

86

Hes·so

A reminder: What is Self-adaptive ML based applications ?



87

Hes·so

Cost model's parameters

Parameter	**Description**
`event_rate`	Rate at which the MLM **(inference)** is triggered
`raw_data_size`	Size of a raw data item fed to the MLM
`app_output_size`	Size of an output item produced by the MLM
`ml_error_rate`	Inference error rate: fraction of events which the MLM is unable to classify/predict over a given period of time
`ml_model_size`	Size of the MLM
`ml_point_size`	Size of a data point used to train the MLM: roughly equivalent to `raw_data_size` + (negligible) label metadata

88

Hes·so

Cost model's parameters

Parameter	Description
ml_train_size	Number of data points (of size ml_point_size) used to train the MLM
ml_train_time	Computing time needed for training the MLM on 1 CPU core with all the data set in RAM
ml_train_rate	Rate of MLM training rounds in the Cloud
ml_underperf	Fraction (i.e., the "underperforming") of all MLMs that must be retrained at each round
edge_img_size	Size of the MLM (including OS, containers, libraries) deployed to any edge device
daily_connect_time	Number of minutes per day during which an edge device is connected to the Cloud

89

Hes·so

Cost model's parameters

Parameter	Description
`deployment_size`	Number of edge devices deployed
`tmetry_metrics`	Number of different telemetry metrics collected at the edge devices
`tmetry_msg_rate`	Rate at which telemetry messages are sent form the edge application to the Cloud
`tmetry_msg_size`	Size of a telemetry message

90

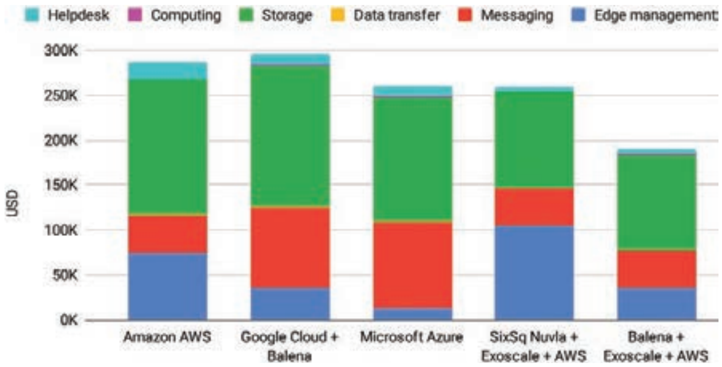
Hes·so

Service costs

Edge device management: yearly subscription, device registration fees, connectivity charges	
Messaging: telemetry & "end-user" applications	
Data transfer: cloud-to-edge, edge-to-cloud and intra-cloud	
Storage: space (standard "hot" S3 tier), read and write operations	
Computing: VM, GPU, etc.	
Helpdesk: business/professional tier for 1 user.	

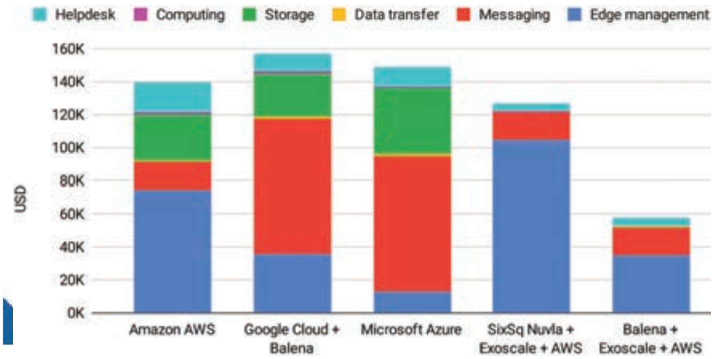
91

Operating cost comparison: results – 1K/1Y



Road traffic management

- Computing and data transfer ~negligible: why?
  - Network bandwidth is cheap
  - No zone replication (no site redundancy)
  - Optimistic computing model



Smart grid

- Computing and data transfer ~negligible (see above)
- Storage: small footprint + Exoscale (no fees for operations) ⇒ advantage for Balena and SixSq

92

Hes·so

Plan

- What is Cloud Continuum ?
- Why Continuum Computing ?
- Use-cases
- Cloud Continuum (Edge-to-Cloud) solutions
- **The Energy Smart Grid Application**

93

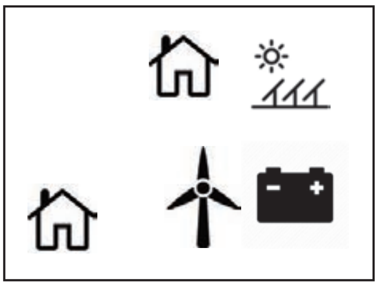
Smart Grid Energy application

SWARM (Eurostar project) - Market driven applied research project

LASAGNE (ERA-NET project) - Research oriented project

94

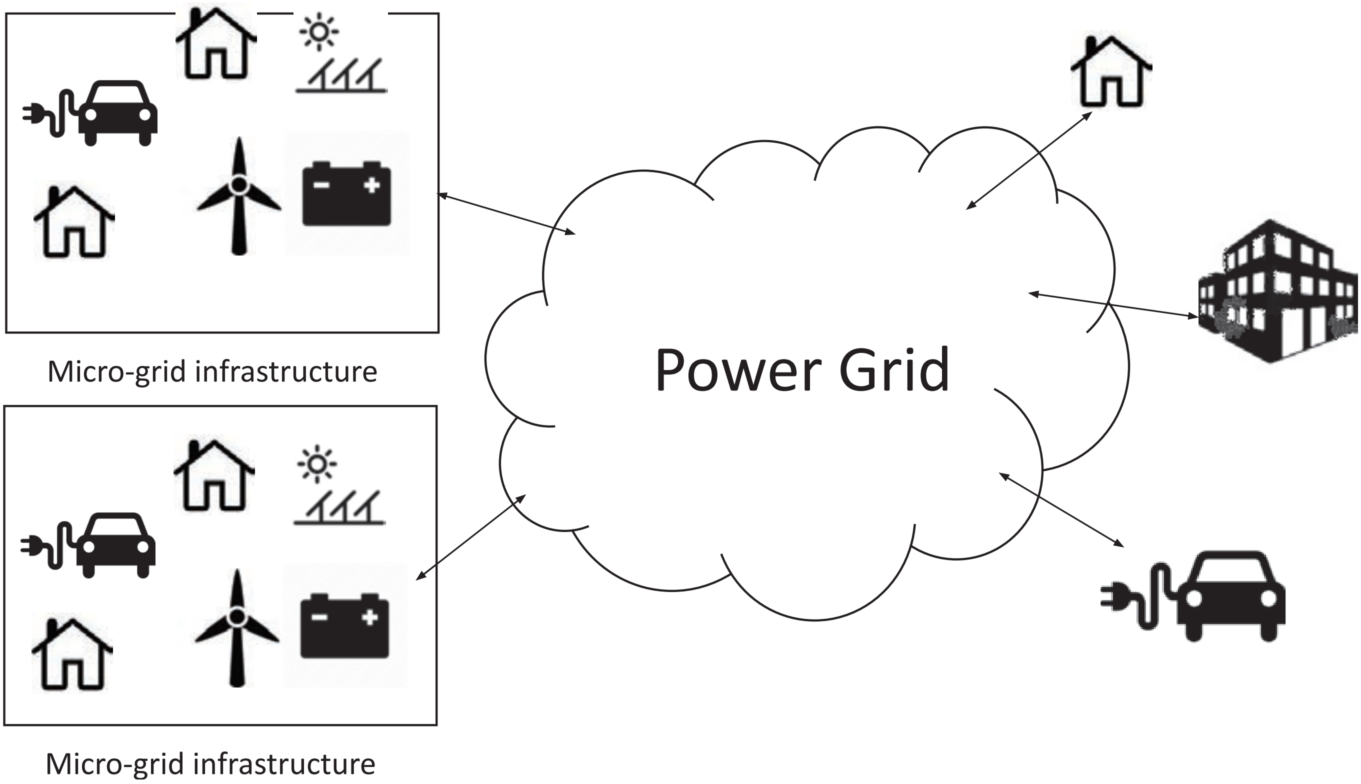
Hes·so The context



95

Hes·so

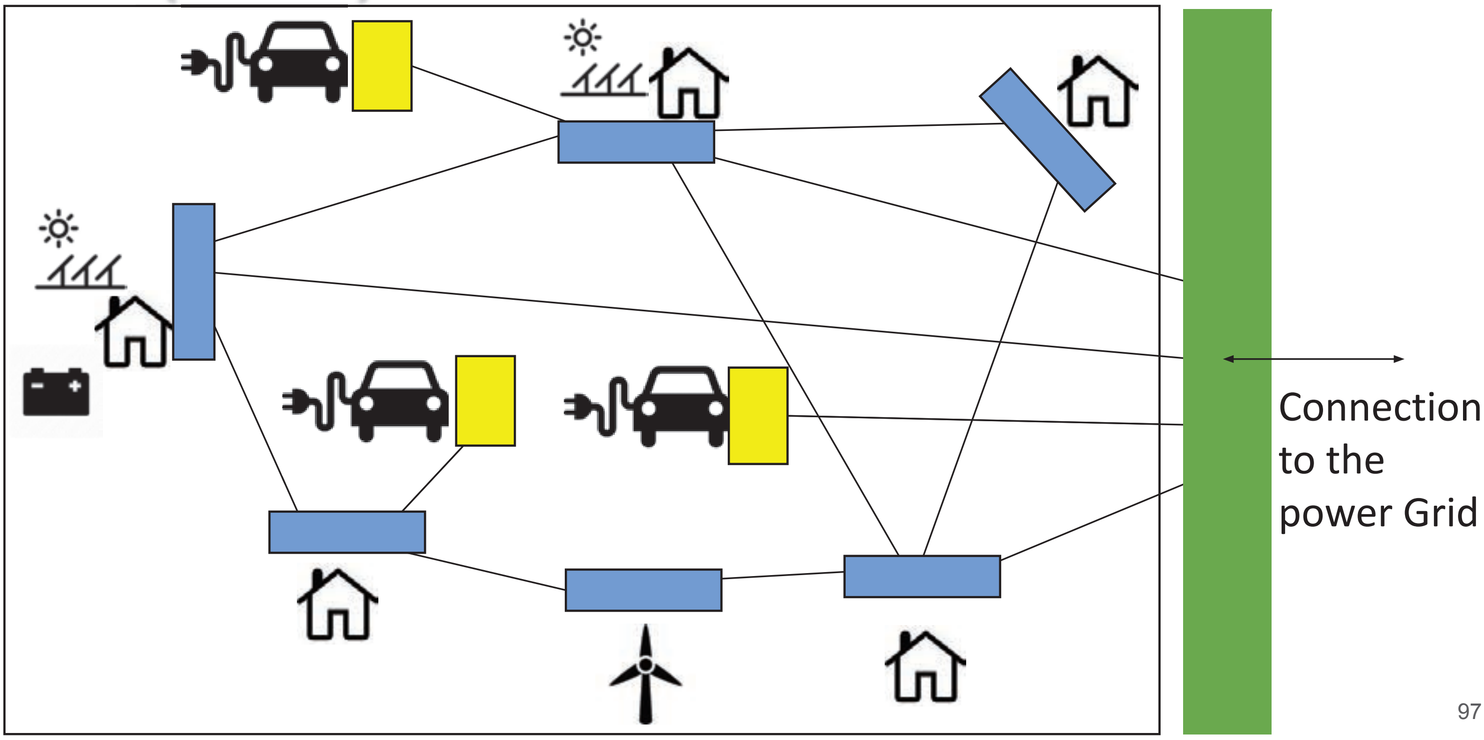
Energy SmartGrid Application



96

Hes·so

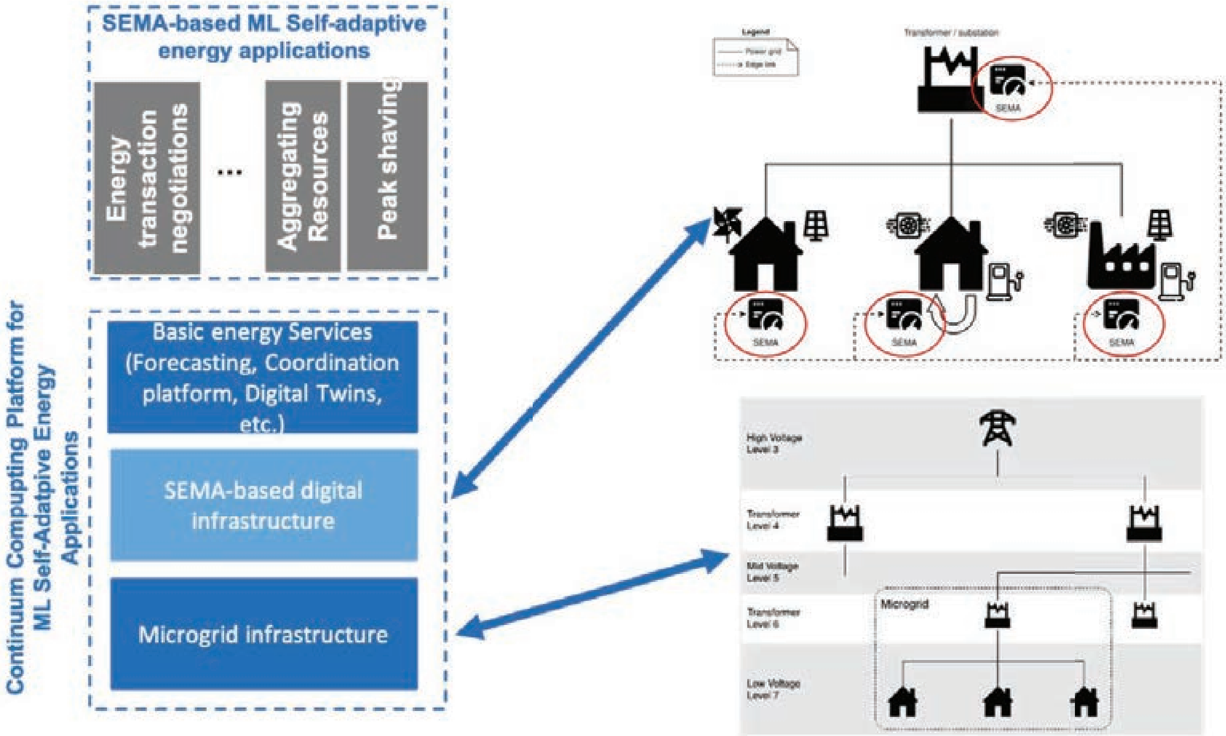
Smart Energy Management Appliance (SEMA)



97

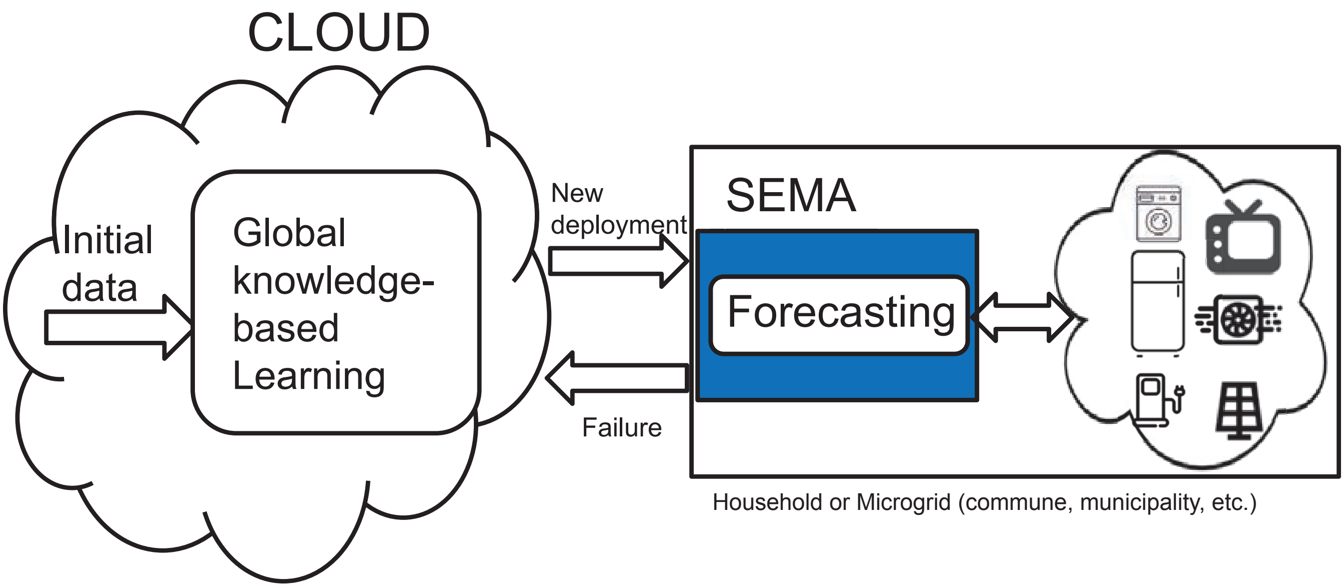
Hes·so

SmartGrid Application



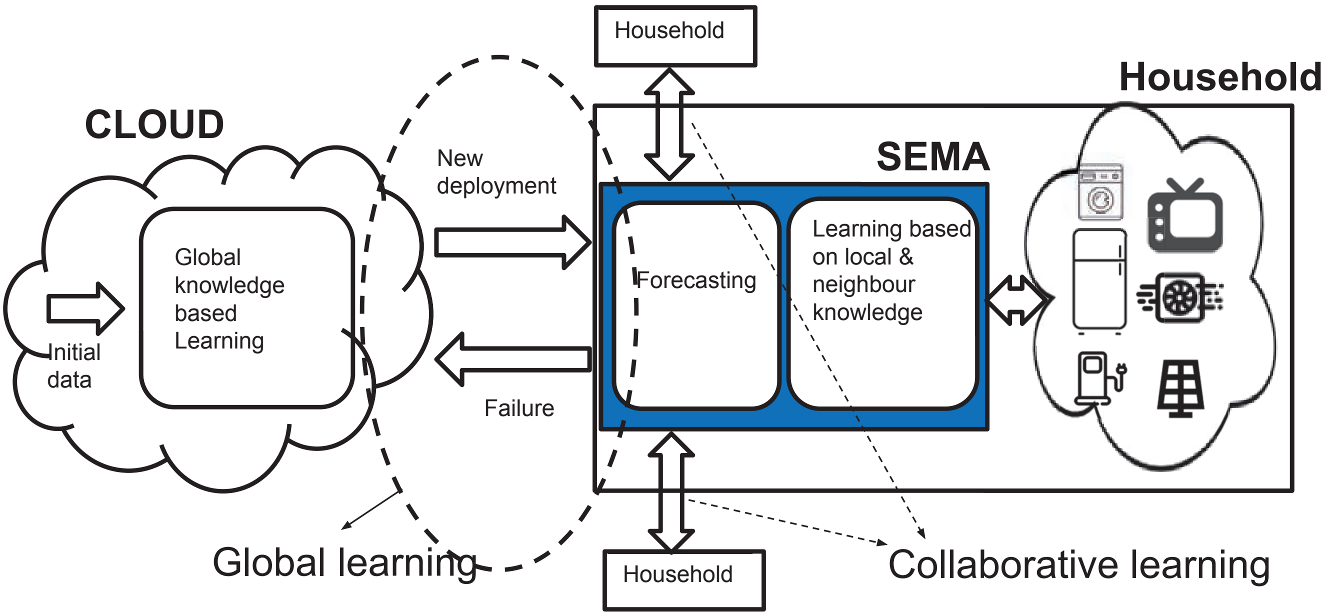
98

Self Adaptive and context-aware Intelligence



99

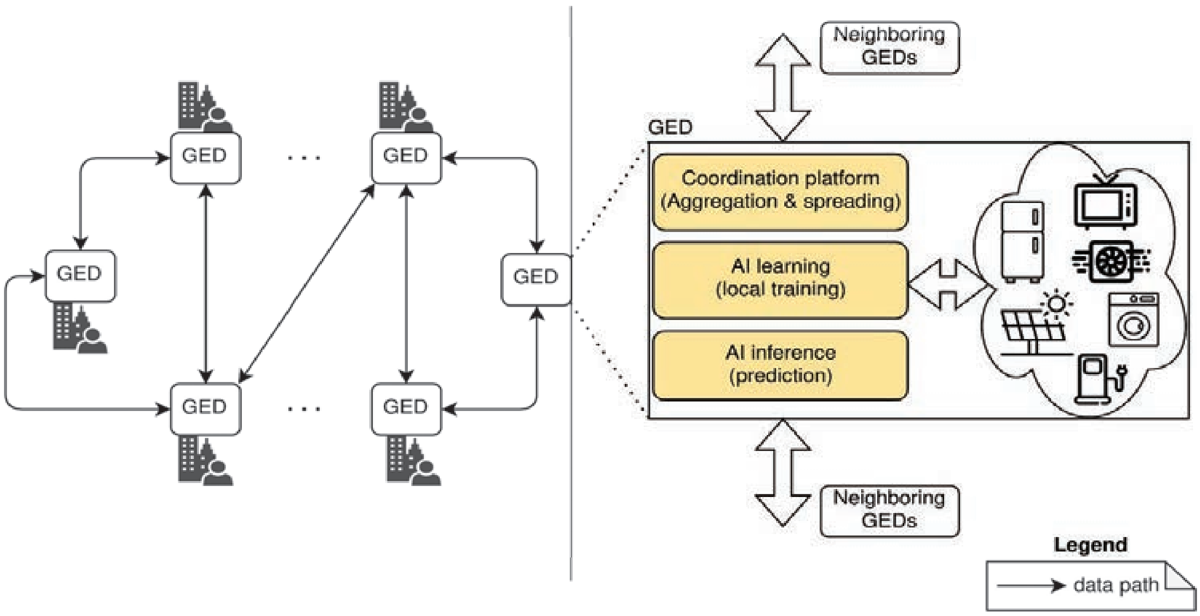
Self Adaptive and context-aware Intelligence



Hes·so

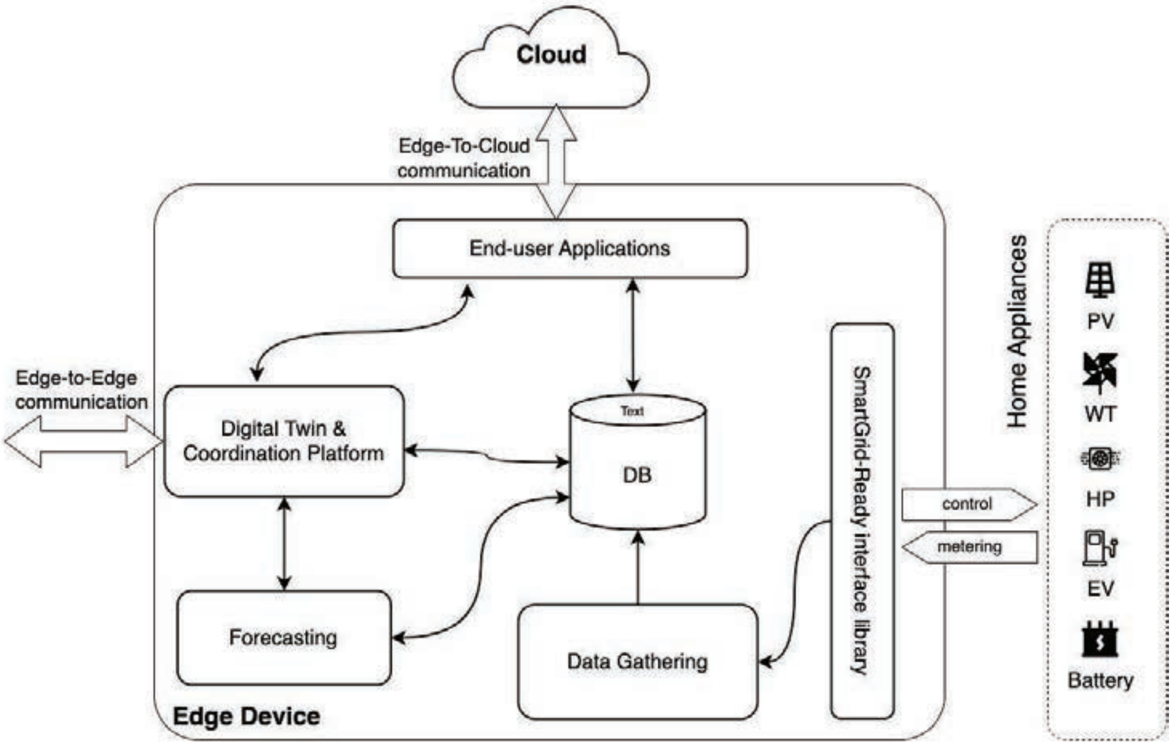
Coordination platform and intelligent digital twins

Decentralised learning through edge-to-edge (E2E) deployment



101

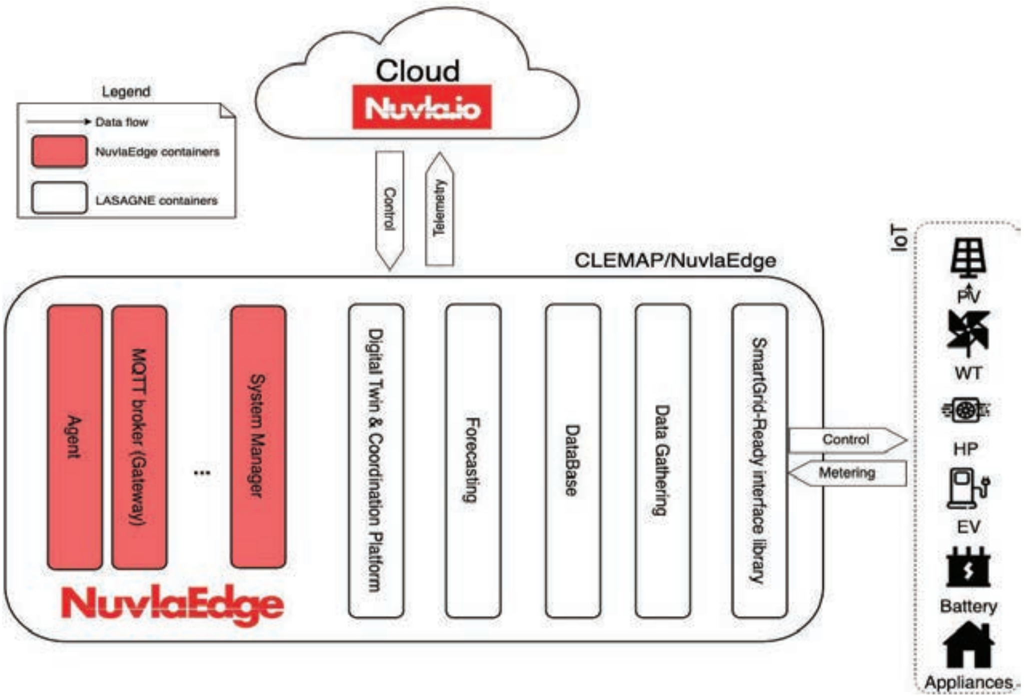
The edge device: What is on board?



102

Hes·so

The edge device (SEMA): What is on board?



103

Learning algorithms

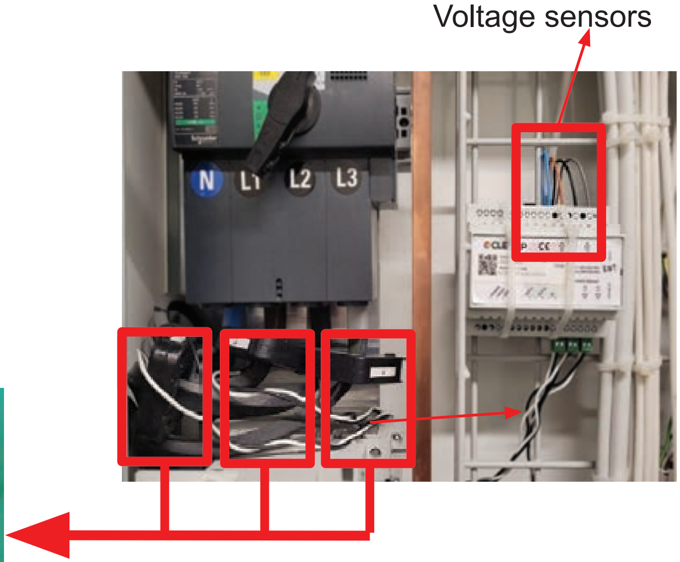
- Long short-term memory(LSTM)
- Hybrid model, CNN-LSTM
- Attention-based CNN-LSTM
- Transformer

104

Hes·so

CLEMAP edge device

Voltage sensors



Linux (Raspberry pi 3)

Sensor : Voltage, Current (3-phases)

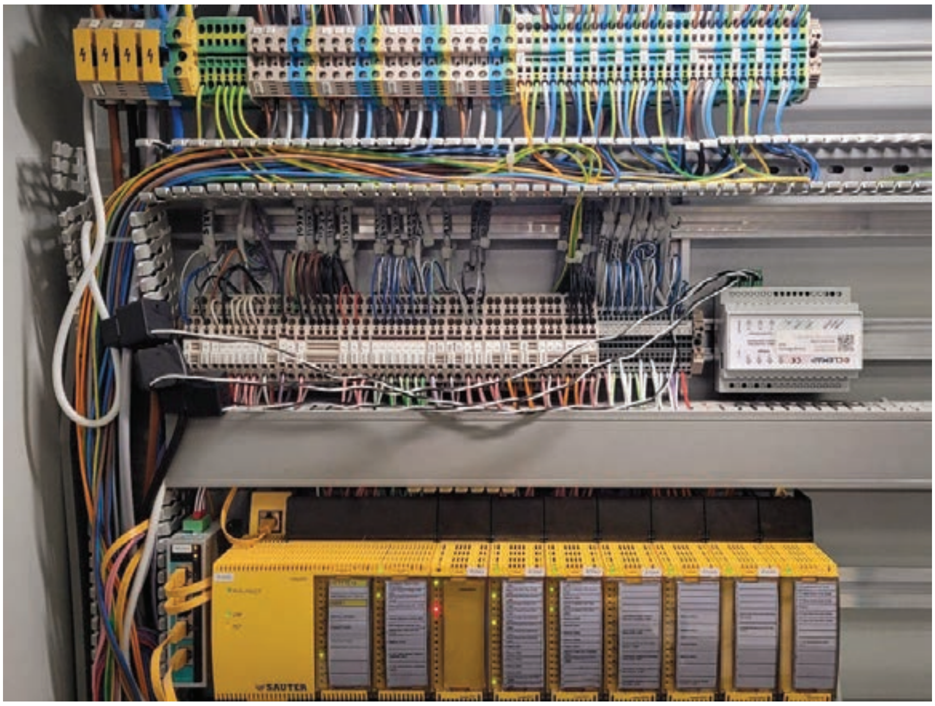
CLEMAP

Current sensors

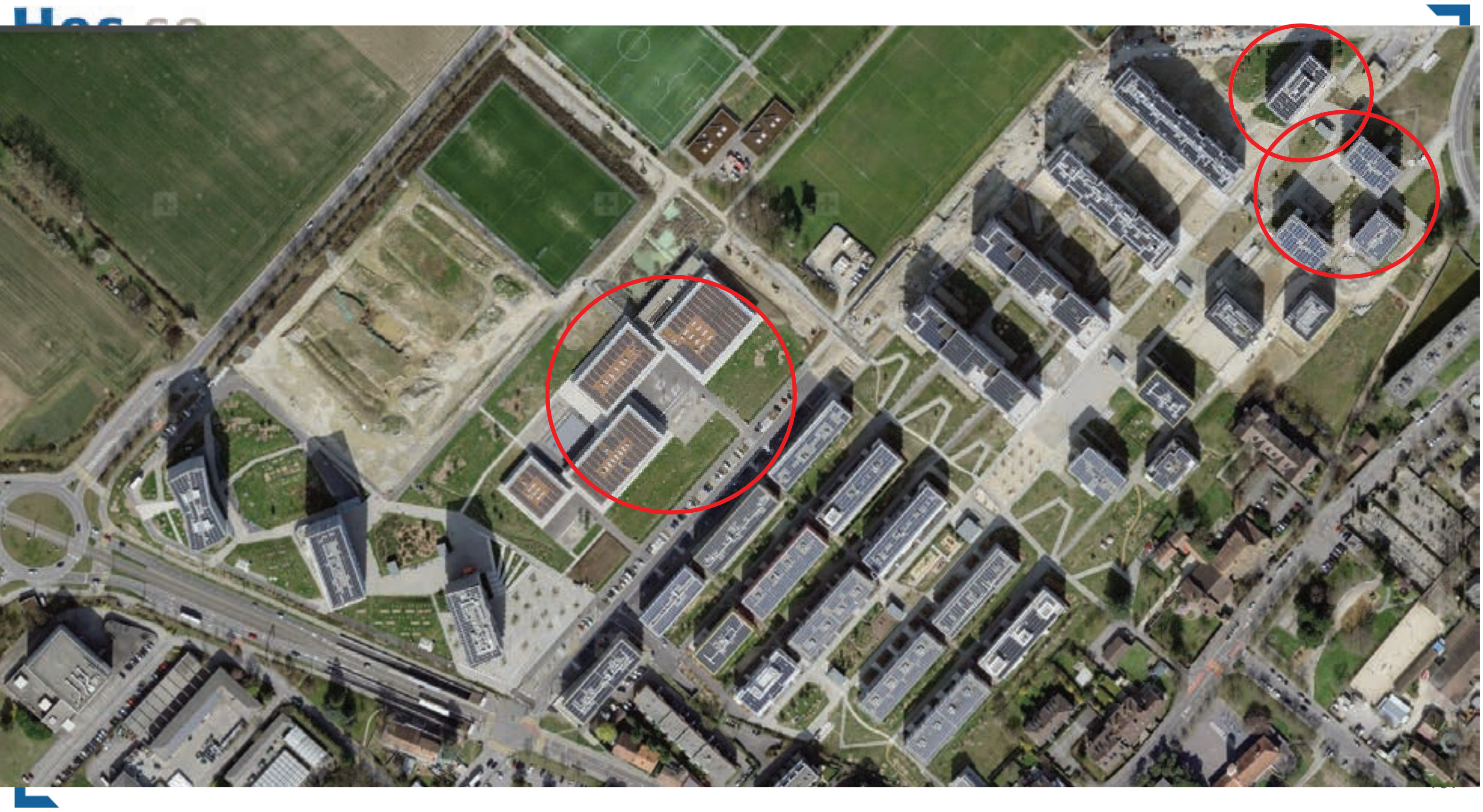
105

Hes·so

CLEMAP edge device

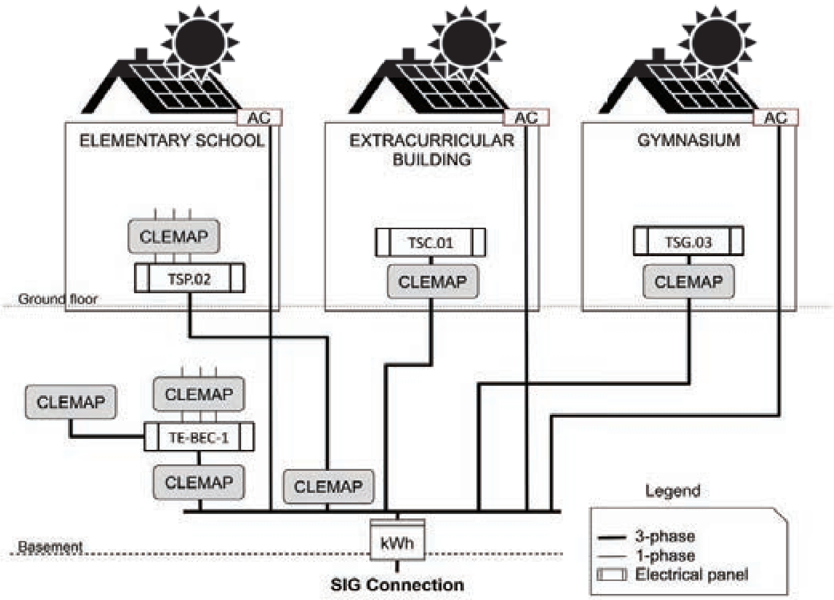


106



Hes·so

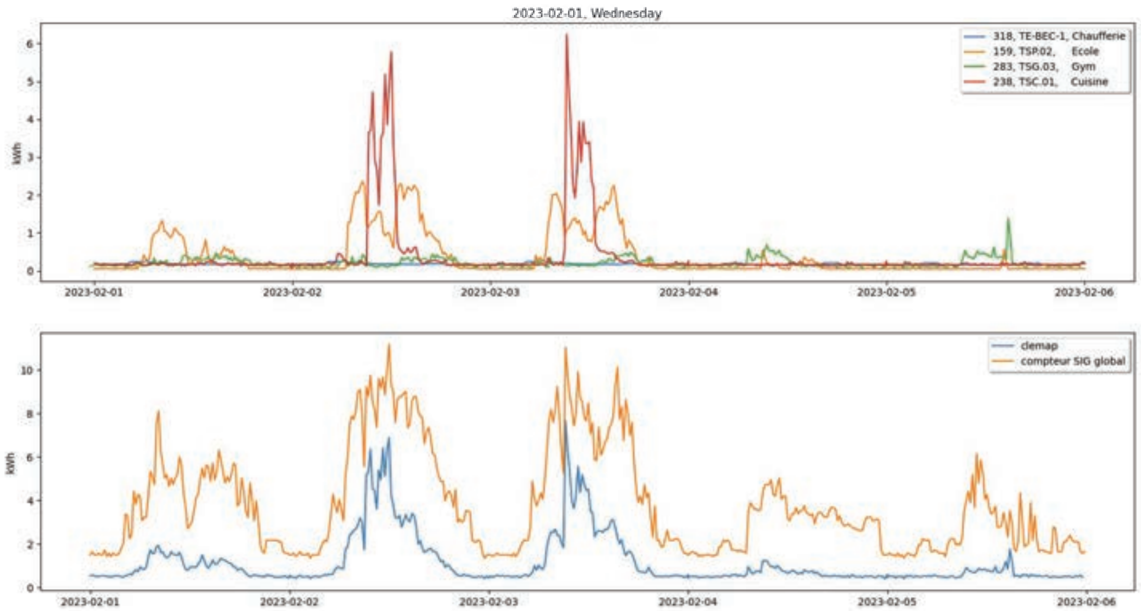
Vergers School



108

Hes·so

Vergers School



109

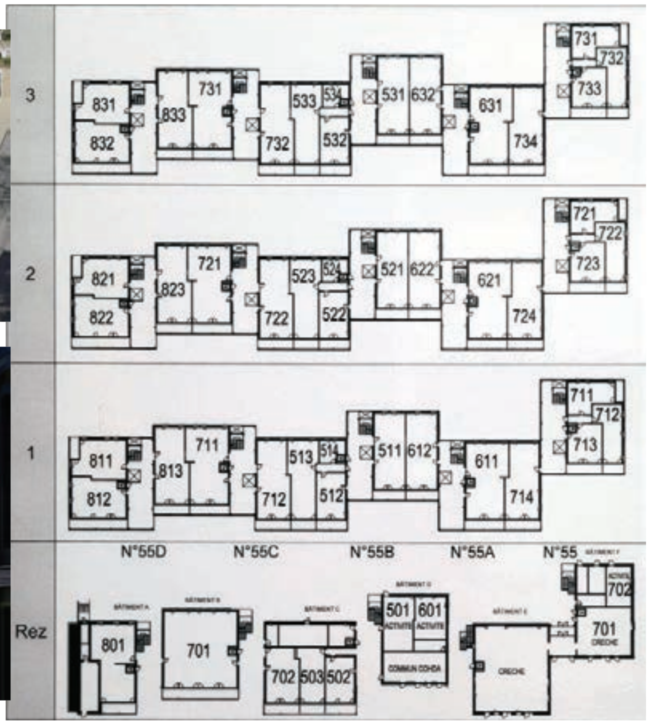
Hes·so

The Polygones "microgrid"

5-floor building with around 28 apartments
5 CLEMAP devices will be installed soon

110

Hes·so Chêne-Bougeries (Geneva), CODHA



Hes·so

Nulva/NuvlaEdge Demo

sixsq.

112

Hes·so

Create a simple, secure and future proof edge infrastructure

Any hardware

NuvlaEdge

NuvlaEdge software

- Turns any computer into a smart edge device
- Hardware agnostic
- Connects securely to Nuvla.io for control

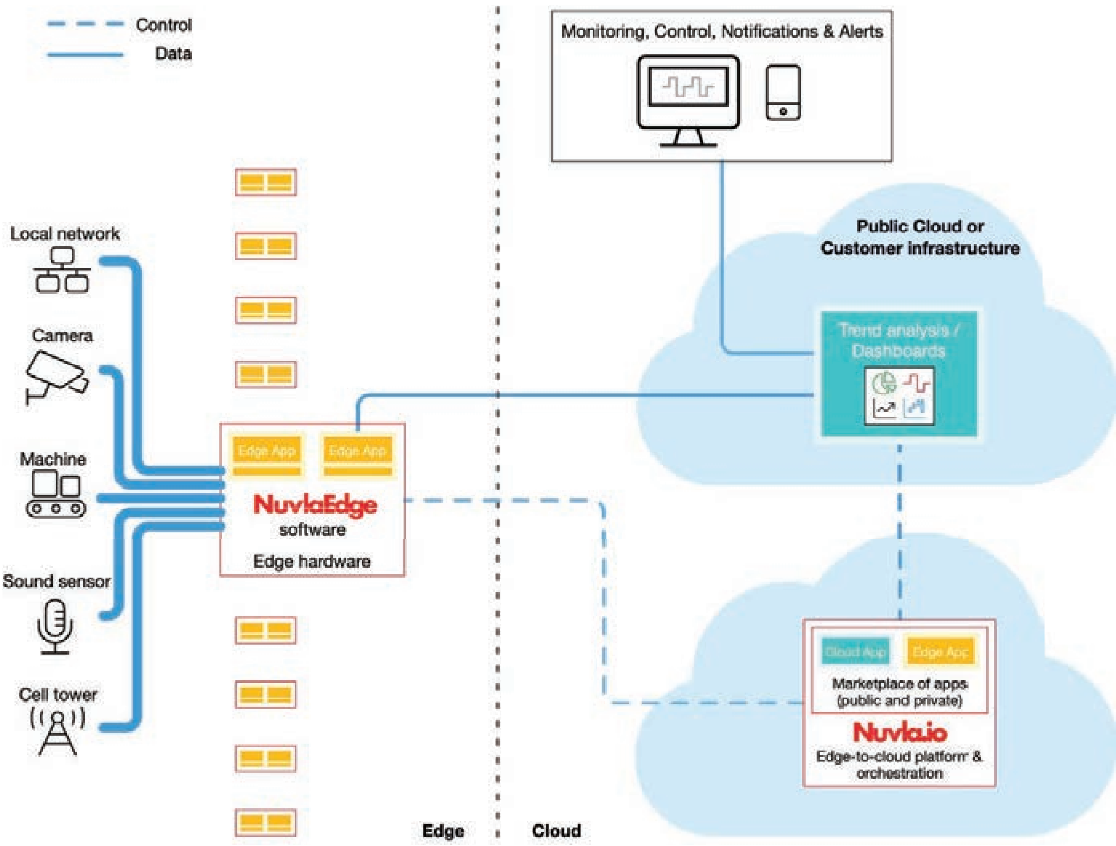
SaaS B2B platform

Nuvla.io

Nuvla.io platform

- Application centric
- Cloud neutral
- Container native
- Open, secure & agile

113


Control
Data
Monitoring, Control, Notifications & Alerts
Public Cloud or
Customer infrastructure
Trend analysis /
Dashboards
Local network
Camera
Machine
Sound sensor
Cell tower
Edge App
Edge App
NuvlaEdge
software
Edge hardware
Cloud App
Edge App
Marketplace of apps
(public and private)
Nuvla.io
Edge-to-cloud platform &
orchestration
Edge
Cloud

114

Hes·so

Nuvla

- A B2B edge management software available as either a stand-alone software stack for installation on premises or as a PaaS via Nuvla.io.
- Enables users to manage their edge devices and deploy applications that combine edge and cloud modules.
- All applications are packaged as containers images and stored in a registry.
- Edge devices and Cloud Computing instances which support Containers Orchestration Engines (COE) can be onboarded and used to provision applications.

Nuvla.io

115

NuvlaEdge

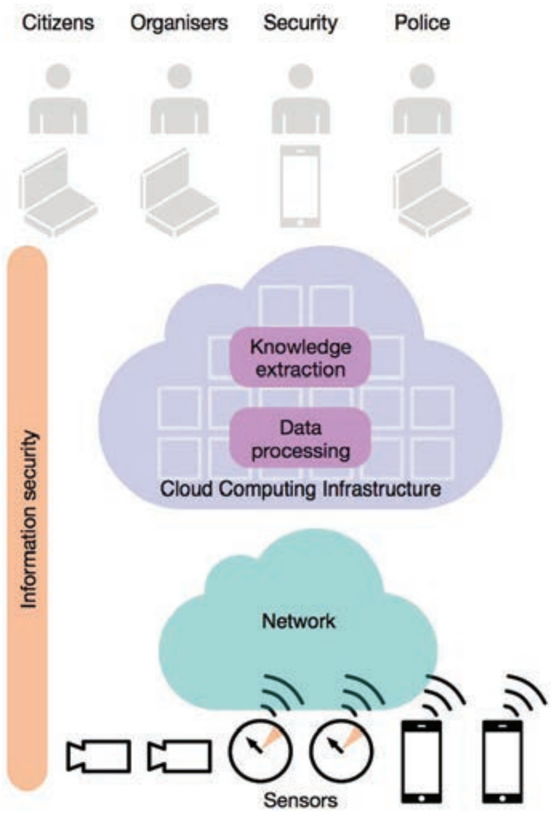
- An Edge framework runtime software composed of a set of microservices, used to transform any device into an "NuvlaEdge" Edge device.
- Allows the user (through Nuvla) to connect to and monitor each edge device individually.
- NuvlaEdge microservices are containers deployed under the control of the Nuvla cloud service.
- Microservices provide facilities for: VPN-based networking, MQTT-based internal messaging, application monitoring, security and discovery of attached HW components

116

Hes·so

To summarise

- Thousands of connected IoT sensors deployed at a large scale.
- Data/Compute sensitive, Context-aware Applications
- Centralised IoT platforms are ill equipped to cope with the huge quantity of collected data.
- → **Edge-to-Cloud → Cloud Continuum**

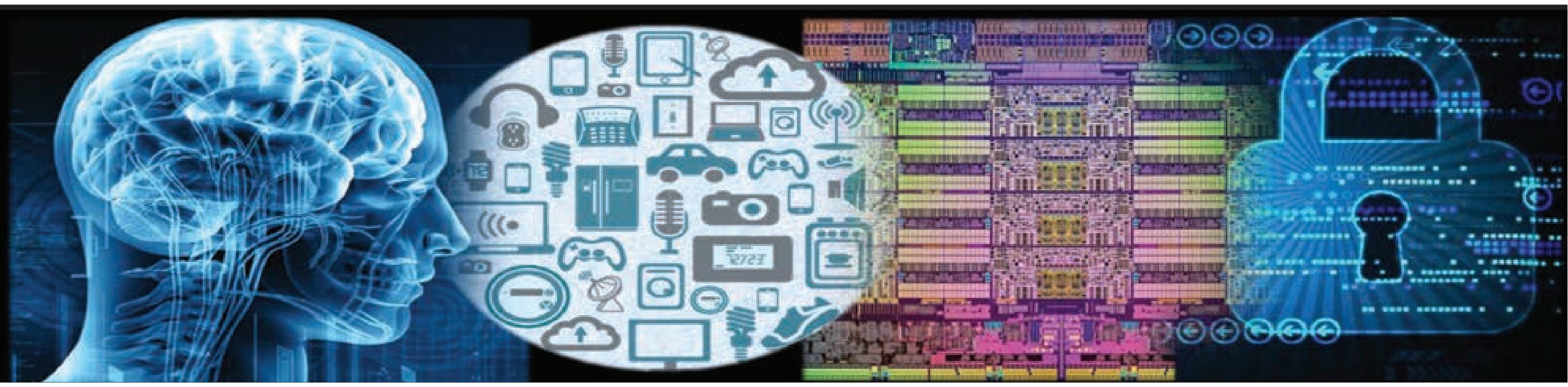


7

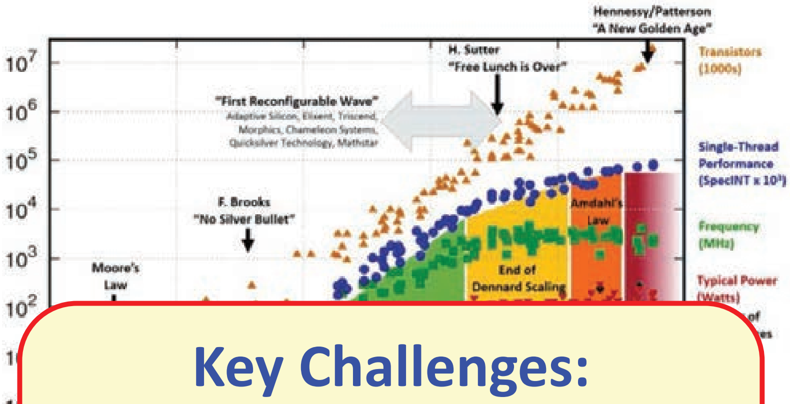

Thank you !

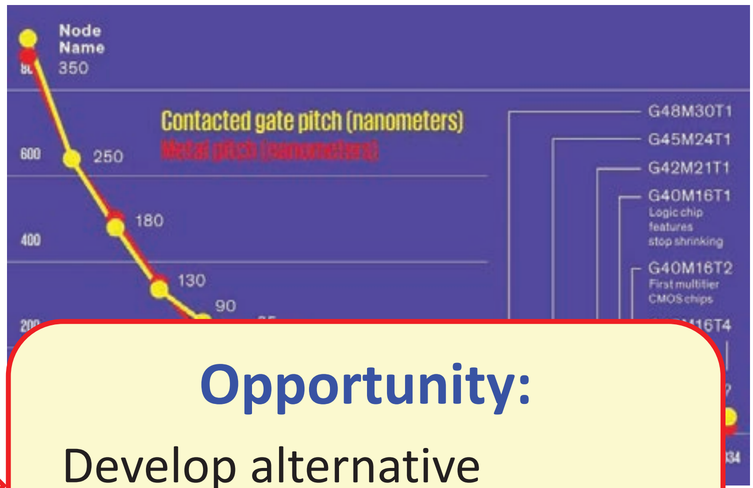
Design Space Exploration of Efficient Quantum Machine Learning Systems

Alberto Marchisio and **Muhammad Shafique**
eBrain Lab, Division of Engineering, New York University (NYU), Abu Dhabi, UAE



AI vs. Technology Scaling



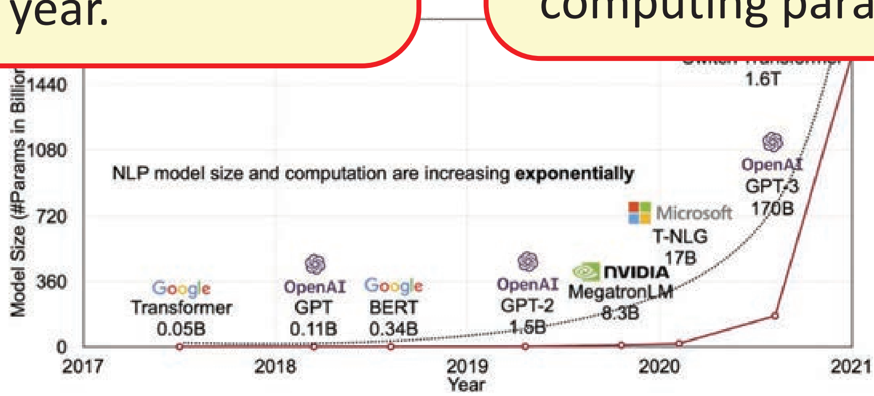


Key Challenges:

1. Saturation of Moore's Law.
2. AI Model sizes increasing 10x every year.

Opportunity:

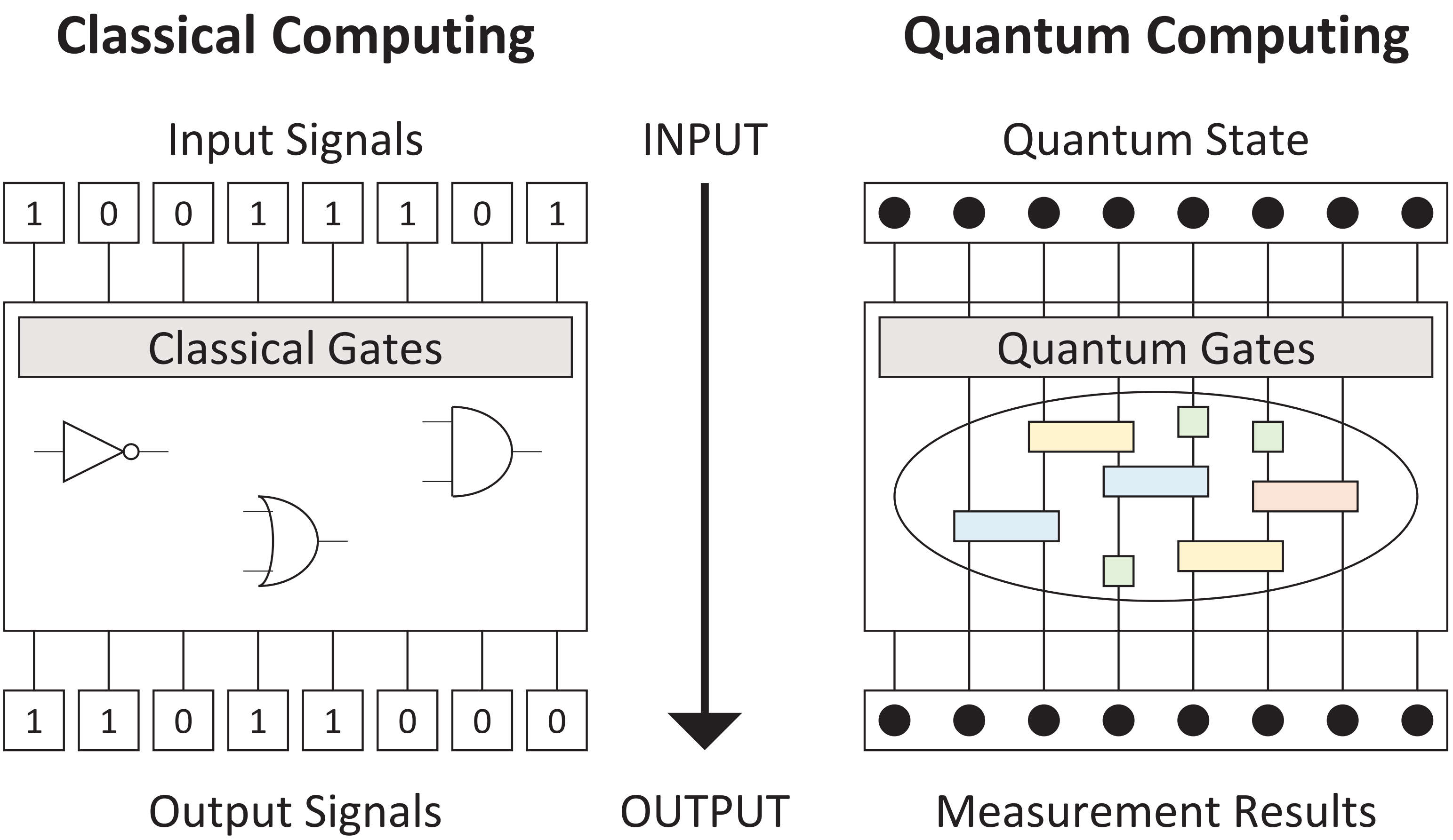
Develop alternative technologies for different computing paradigms



https://www.shaip.com/blog/a-guide-large-language-model-llm/

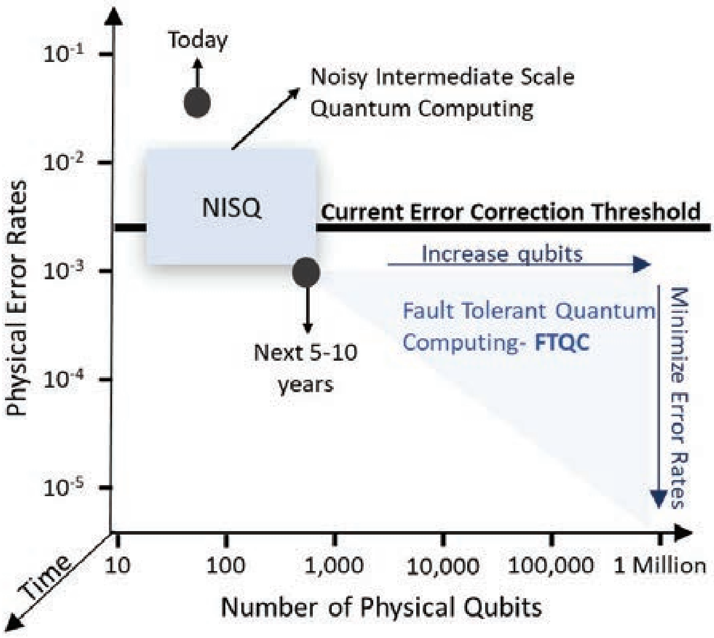
2

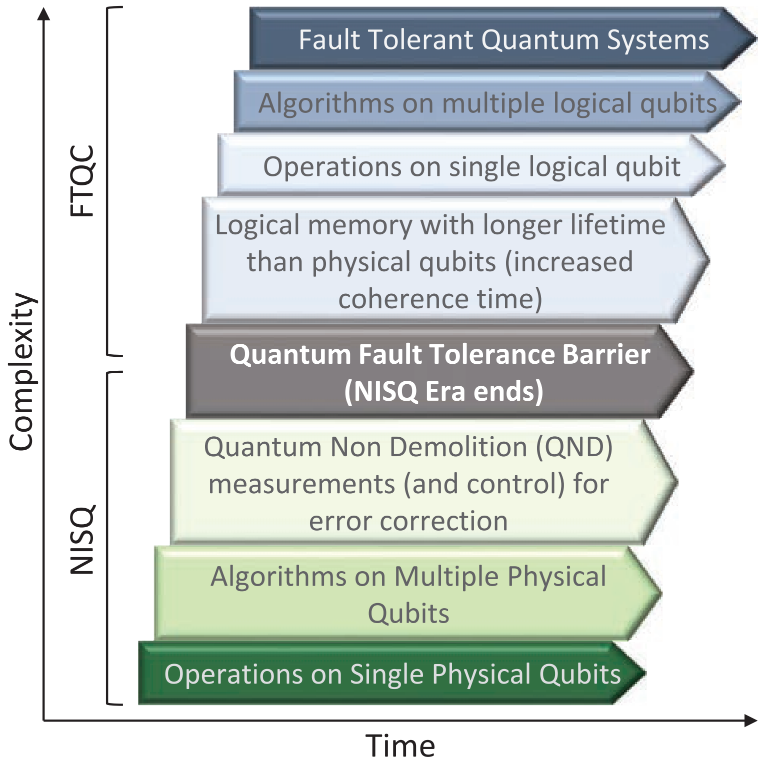
Classical vs. Quantum Computing



3

From NISQ to FTQC





4

Qubit

Classical Bit	
0	1

Quantum Bit (Qubit)

$_0$ α0 + β1 $_1$

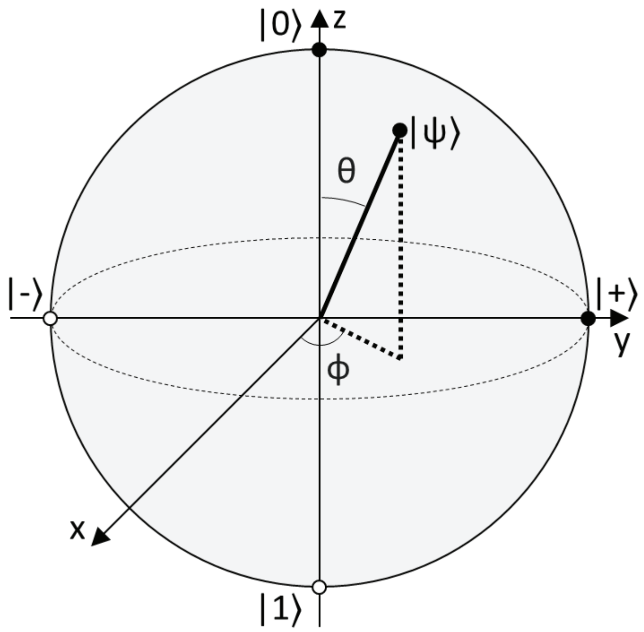
Classical vs. Quantum Bit

Classical bit: at one instance of time, it can hold a value of 0 or 1.

Qubit: at one instance of time, it can represent a mixture of the 0 and 1 state values simultaneously. The final state is determined by measurement.

Bloch Sphere

Visual representation of a two-level qubit system for a single qubit



5

Single Qubit Superposition

Max-Born rule: $|\alpha|^2 + |\beta|^2 = 1$

	0	1
(a)	$\alpha 0 + \beta 1$ $\lvert\alpha\rvert^2 = 1.0$ and $\lvert\beta\rvert^2 = 0.0$	
(b)	$\alpha 0 + \beta 1$ $\lvert\alpha\rvert^2 = 0.7$ and $\lvert\beta\rvert^2 = 0.3$	
(c)	$\alpha 0 + \beta 1$ $\lvert\alpha\rvert^2 = 0.5$ and $\lvert\beta\rvert^2 = 0.5$	
(d)	$\alpha 0 + \beta 1$ $\lvert\alpha\rvert^2 = 0.3$ and $\lvert\beta\rvert^2 = 0.7$	
(e)	$\alpha 0 + \beta 1$ $\lvert\alpha\rvert^2 = 0.0$ and $\lvert\beta\rvert^2 = 1.0$	

Examples

a) Fully measured to be in state 0.

b) 70% probability to be measured state 0 and 30% for state 1.

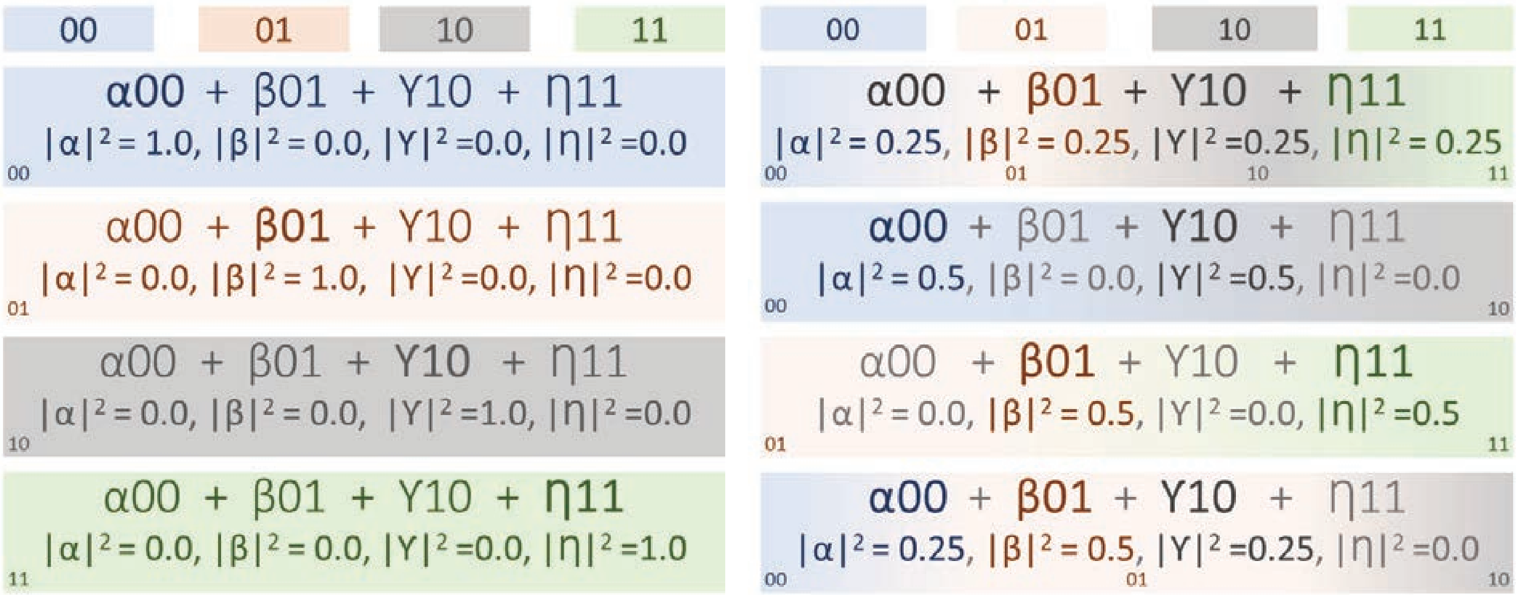
c) Equal probability (50%) to be measured in state 0 and state 1.

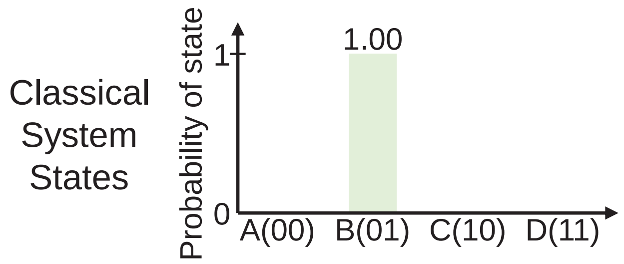
d) 30% probability to be measured in state 0 and 70% for state 1.

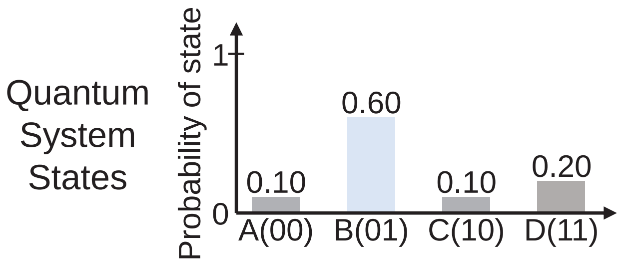
e) Fully measured to be in state 1.

6

Two-Qubit Superposition







7

Quantum Gates & Circuits

Quantum Gates

Mathematically described by unitary matrices U, where:

$$adj(U) = U^{-1}$$

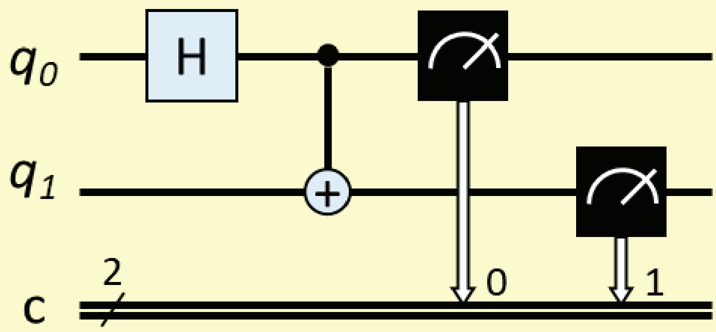
List of possible gates

- Identity
- Pauli Gates (X, Y, Z)
- Controlled Not (CNOT)
- Controlled Z (CZ)
- Hadamard
- Phase Shift
- Swap
- …

Quantum Circuits

Collections of quantum gates interconnected by quantum wires that perform a certain task.

Example of a Quantum Circuit



8

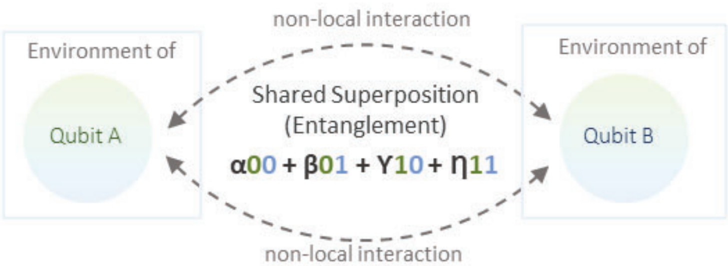
Correlation & Entanglement

Correlation

Given a system of two (or more) parts, the correlation quantifies how much we can predict about the second part when knowing the the first part, compared to how much we can predict about the second part without that knowledge.

Entanglement

Correlation between two particles even if they are separated by a great distance.



Decoherence

Loss of quantum coherence (i.e., definite phase relation between different quantum states).

9

Quantum Noise, Error Correction & Mitigation

Uncertainty Principle

It is not possible to accurately know the values of specific related pairs of physical quantities of a particle (e.g., velocity, position, momentum).

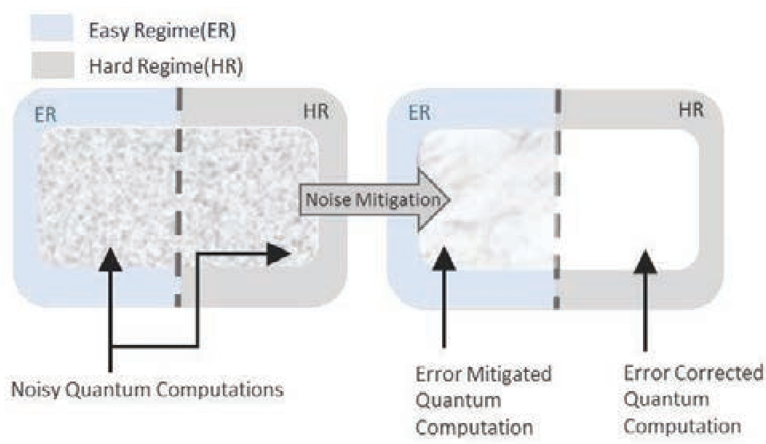
⇨

Quantum Noise

Since qubits physically function on quantum mechanical principles, they are susceptible to interference from the environment and imperfect fabrication.

Quantum Noise Countermeasures

- ❑ **Error Correction**: the goal is to restore the correct value after the error occurs. Redundancy is introduced by spreading the information of a single qubit onto an entangled state of multiple qubits.
- ❑ **Error Mitigation**: the goal is to reduce or suppress errors that occur during computation.



10

Quantum Stack – From Hardware to Applications

Quantum Algorithms

Quantum Programming Languages

Quantum Compiler

Quantum Instruction Set Architecture

Quantum Microarchitecture
(Error Correction and Mitigation)

Control and Readout Electronics

Quantum Hardware – Qubit level

Perfect Qubits (simulated)	Realistic Qubits (superconducting qubit, ion trapped qubit, quantum dot, neutral atom qubit)

11

Benefits of Quantum Computing

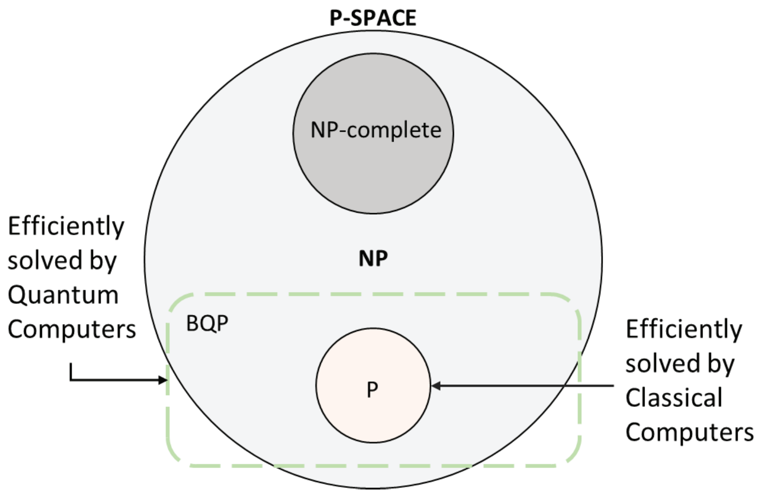
Quantum Dimensionality Reduction

Typically used in the data preprocessing stage of ML tasks, for example using Principal Component Analysis (PCA).

Preserving Maximum Information

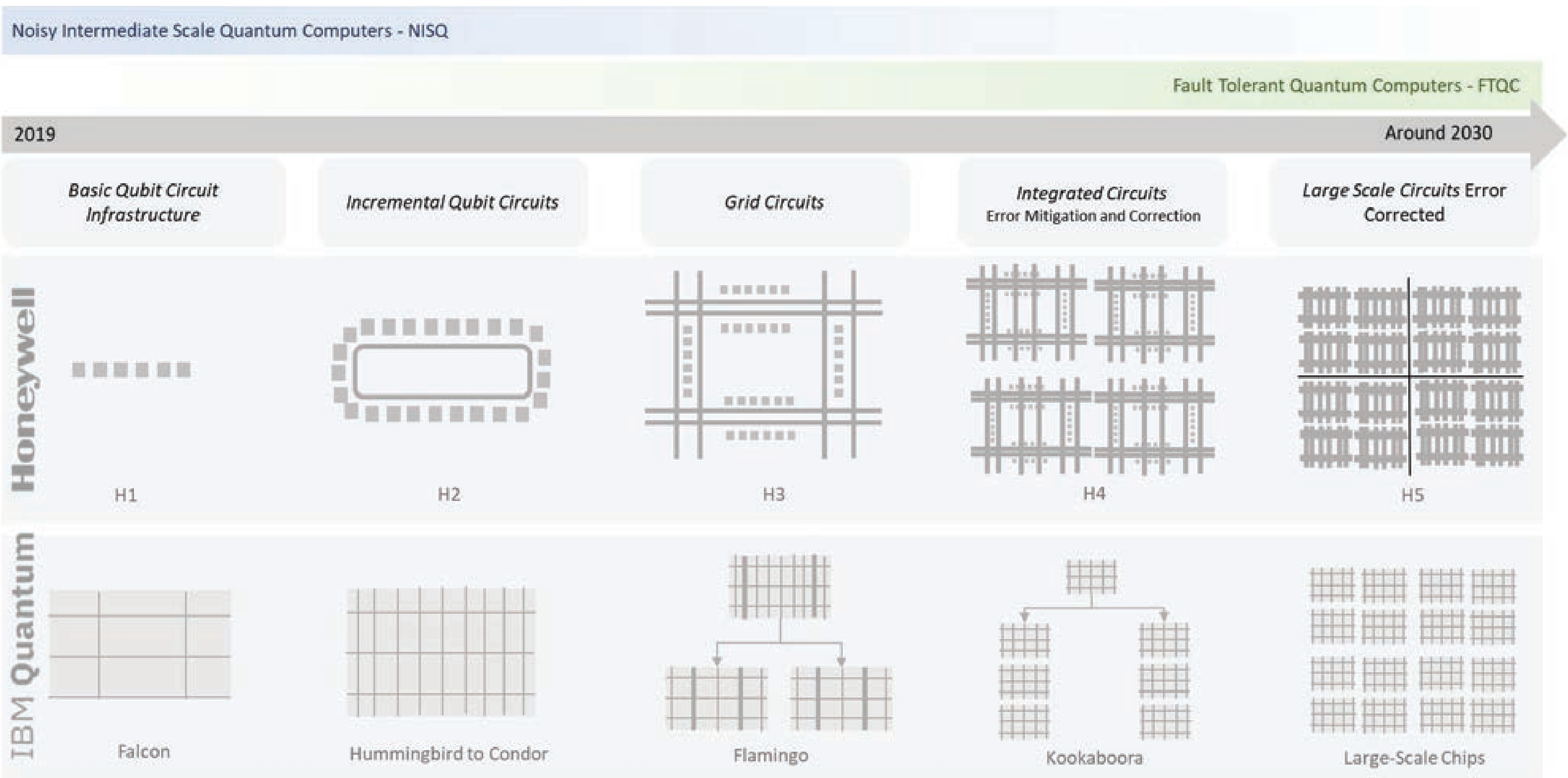
While classical bits can take either 0 or 1 values, qubits can represent a dual state of 0 and 1 simultaneously due to the superposition. Therefore, to express n-bit combinations, we need 2^n combinations in classical systems, while quantum systems can find the solution faster.

Quantum Supremacy for Solving Complex Problems



12

Roadmap to Design Large-Scale Quantum Chips



13

HW/SW Players in QC

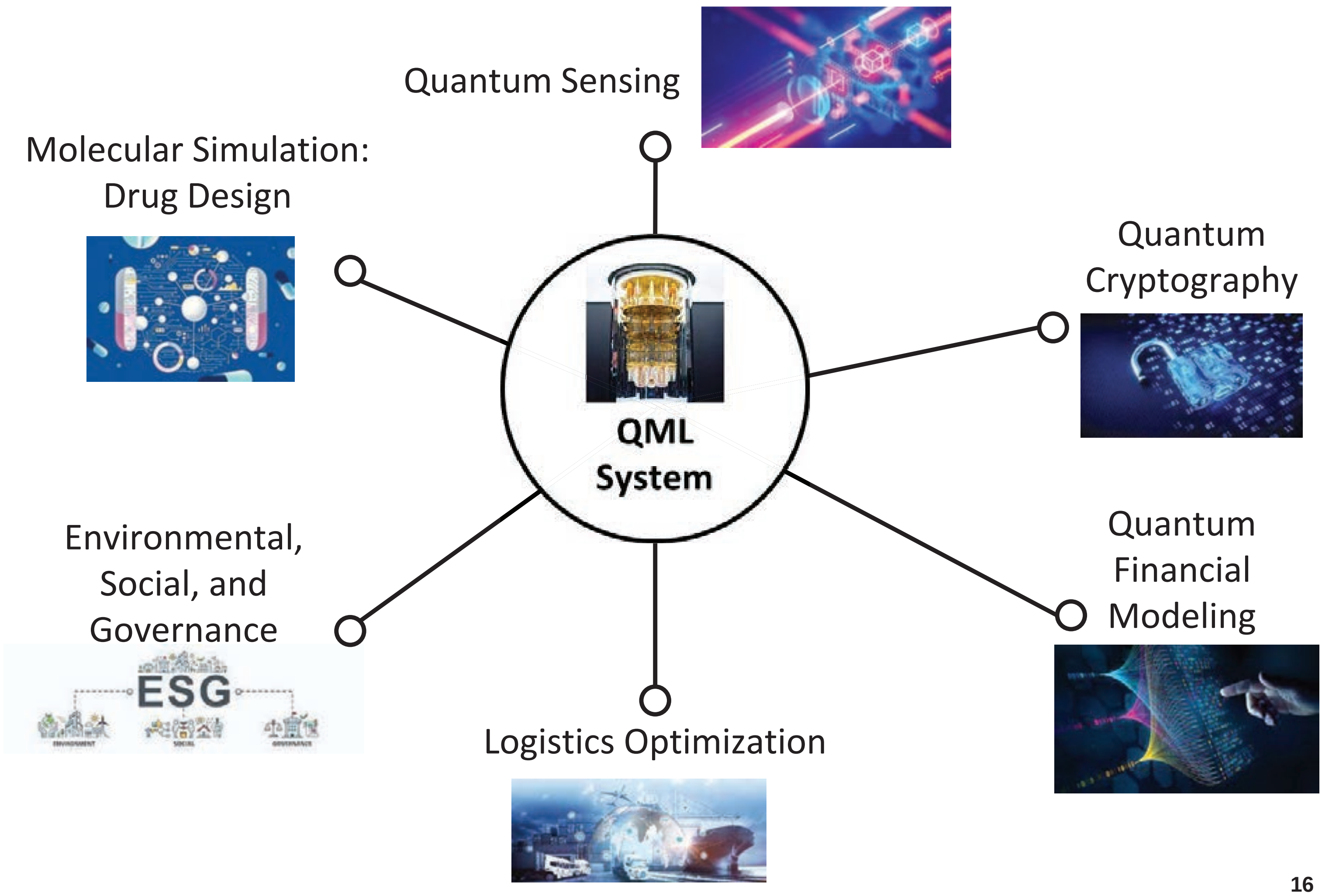
Disclaimer: The list may not be comprehensive. It's just an excerpt.

14

Quantum Machine Learning

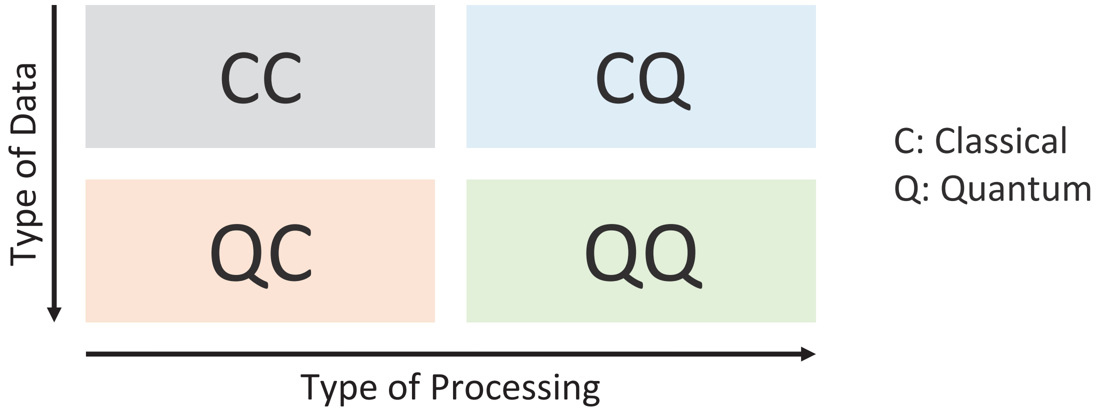
15

Applications for QML



16

Categorization of QML Approaches

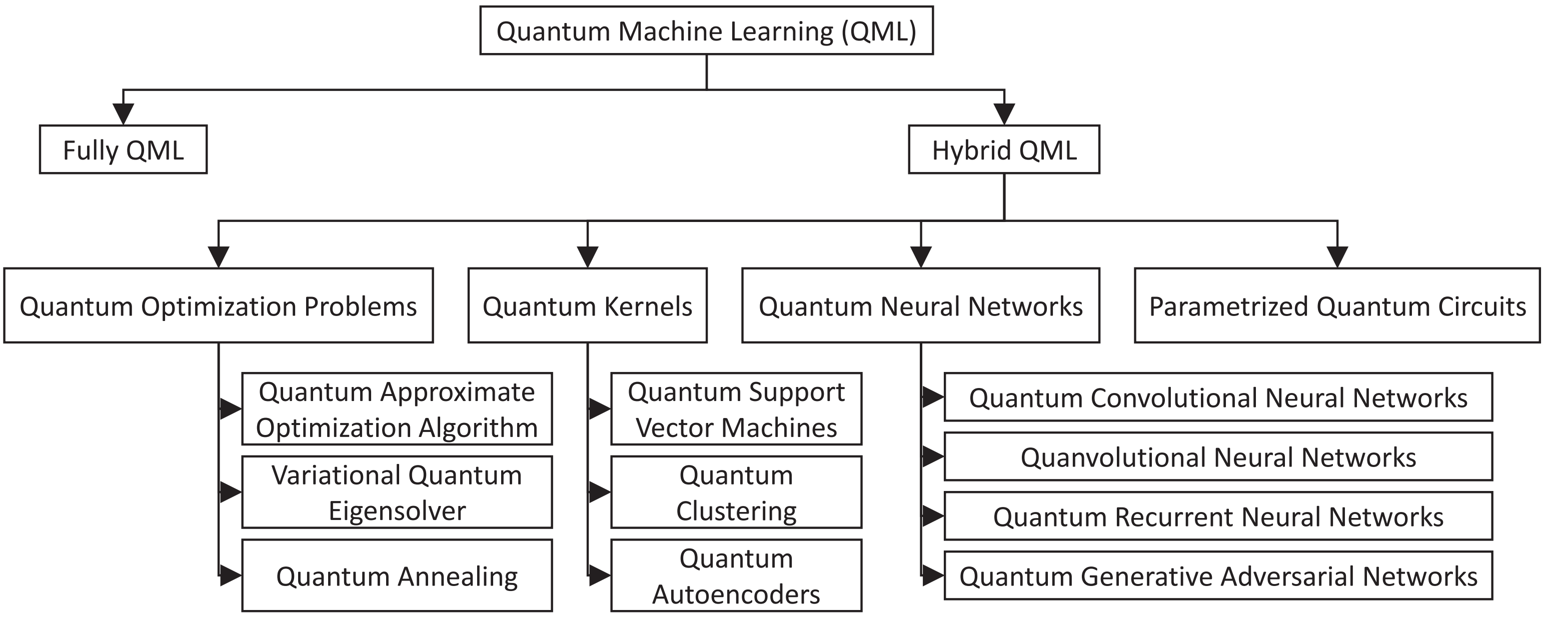


QML Categorization

1. **CC – Classical data using Classical computing**, but using algorithms inspired by quantum computing, such as the recommendation system algorithm.
2. **CQ – Classical data using Quantum ML algorithms.** This is the main focus of this work.
3. **QC – Quantum data using Classical computing.** This is an active area of investigation, with classical ML algorithms used in many quantum computing areas, such as qubit characterization, control, and readout.
4. **QQ – Quantum data using Quantum ML algorithms.** It is a future investigation area that can be developed during a more mature stage of quantum computing.

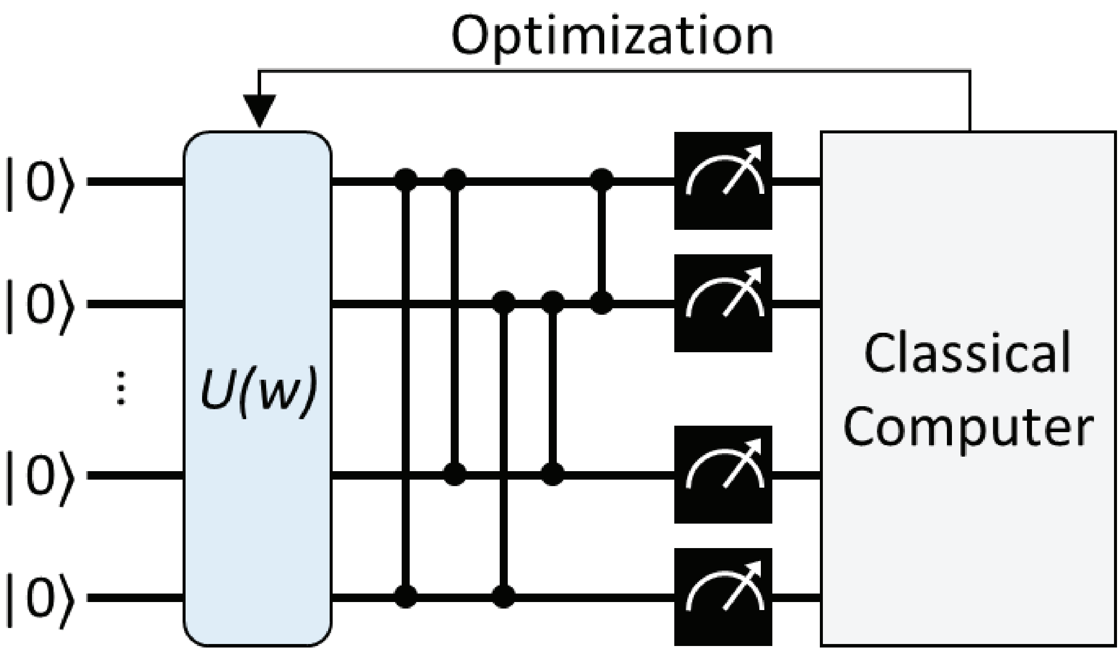
17

Quantum Machine Learning Algorithms



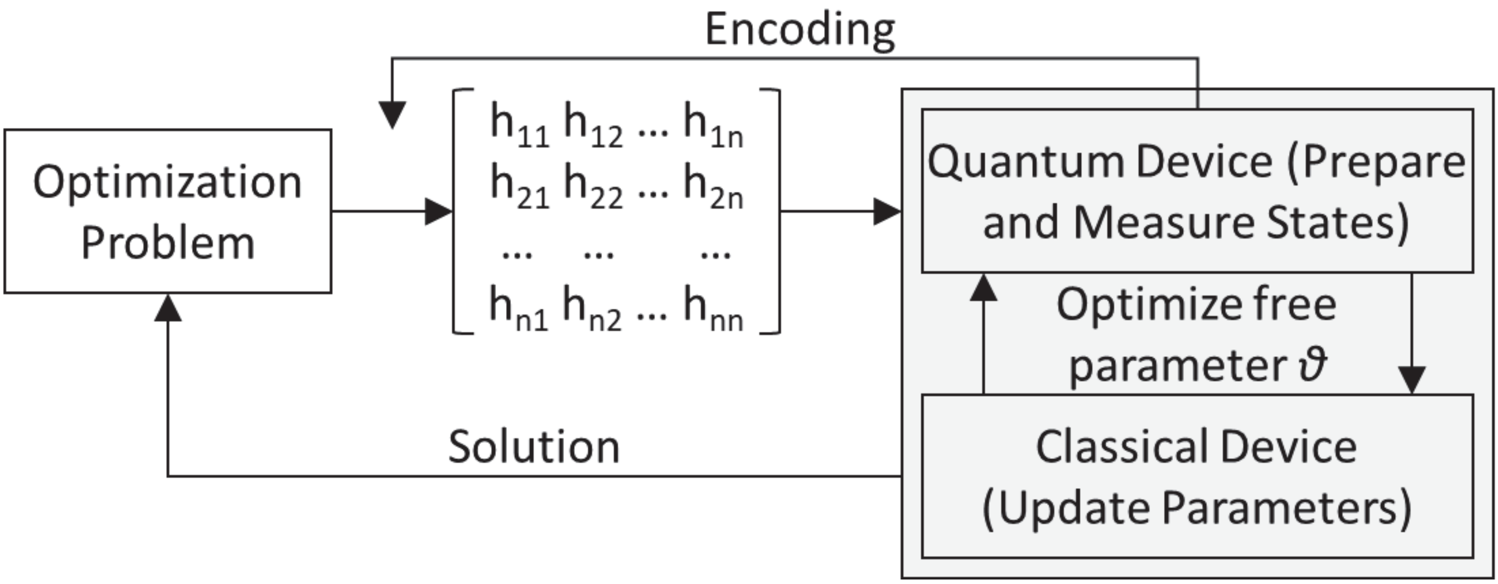
18

Parametrized Quantum Circuits



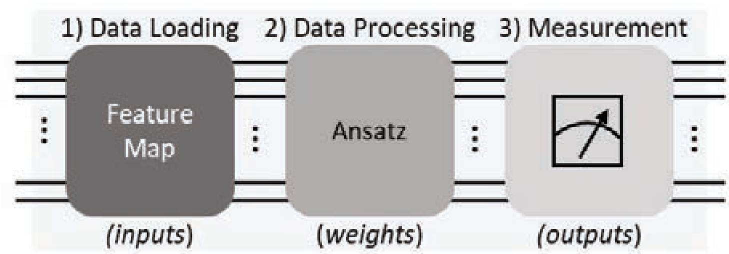
19

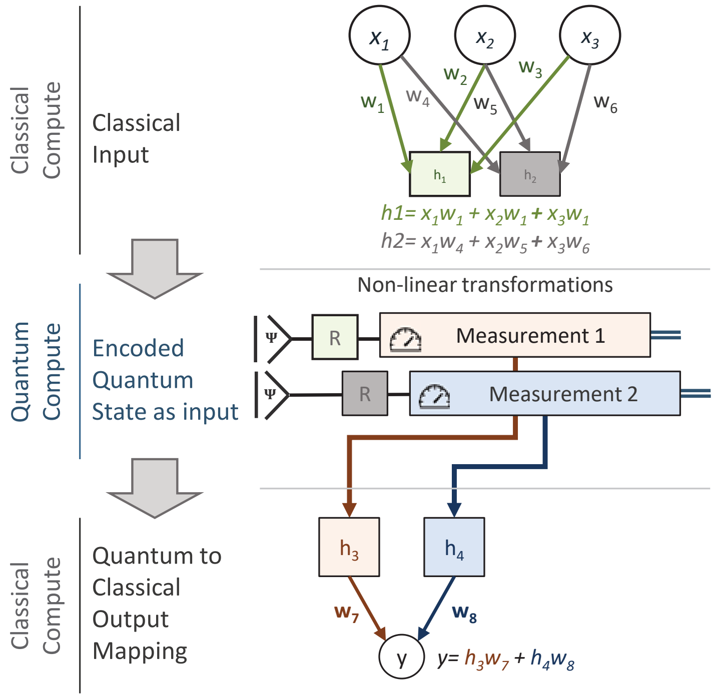
Quantum Optimization Problems



20

Quantum Neural Networks

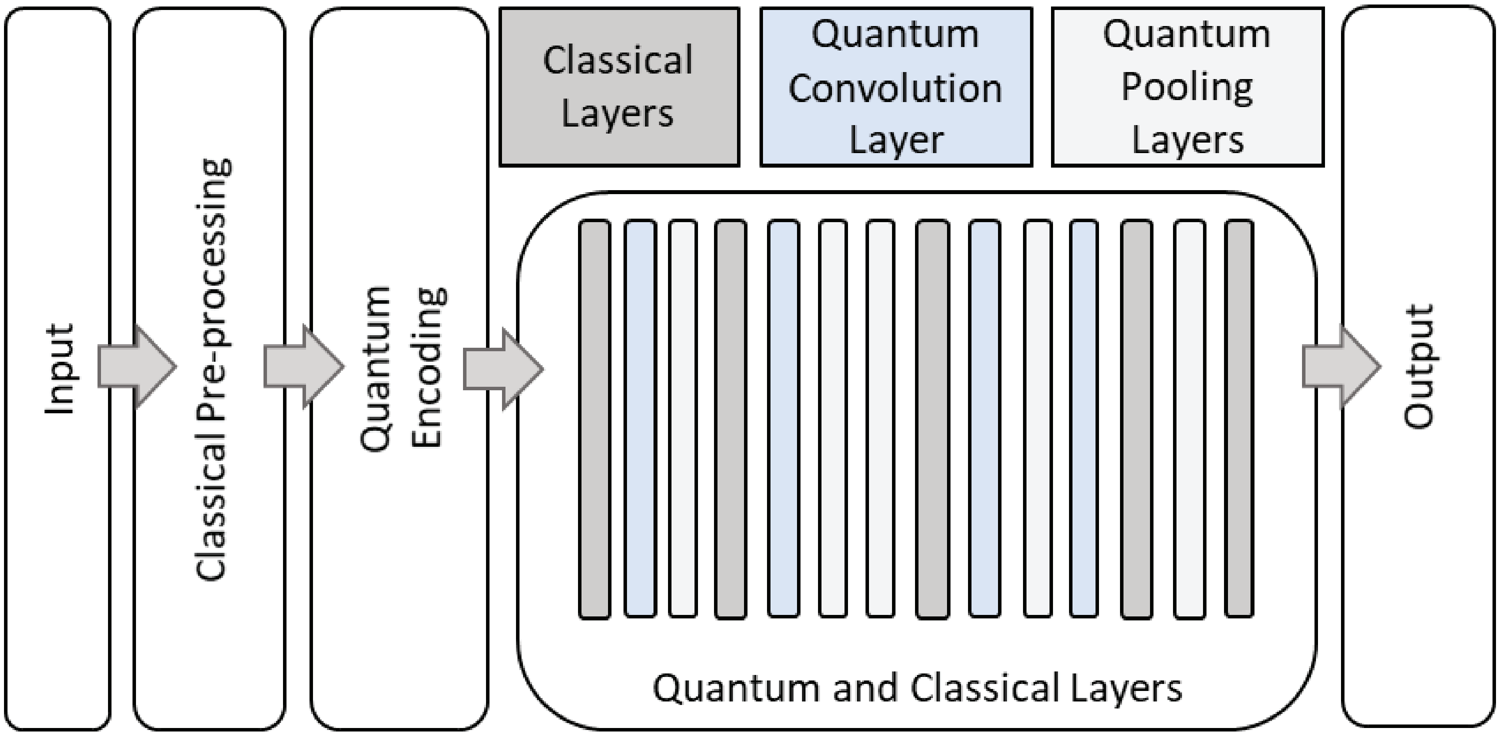




ANNs vs QNNs

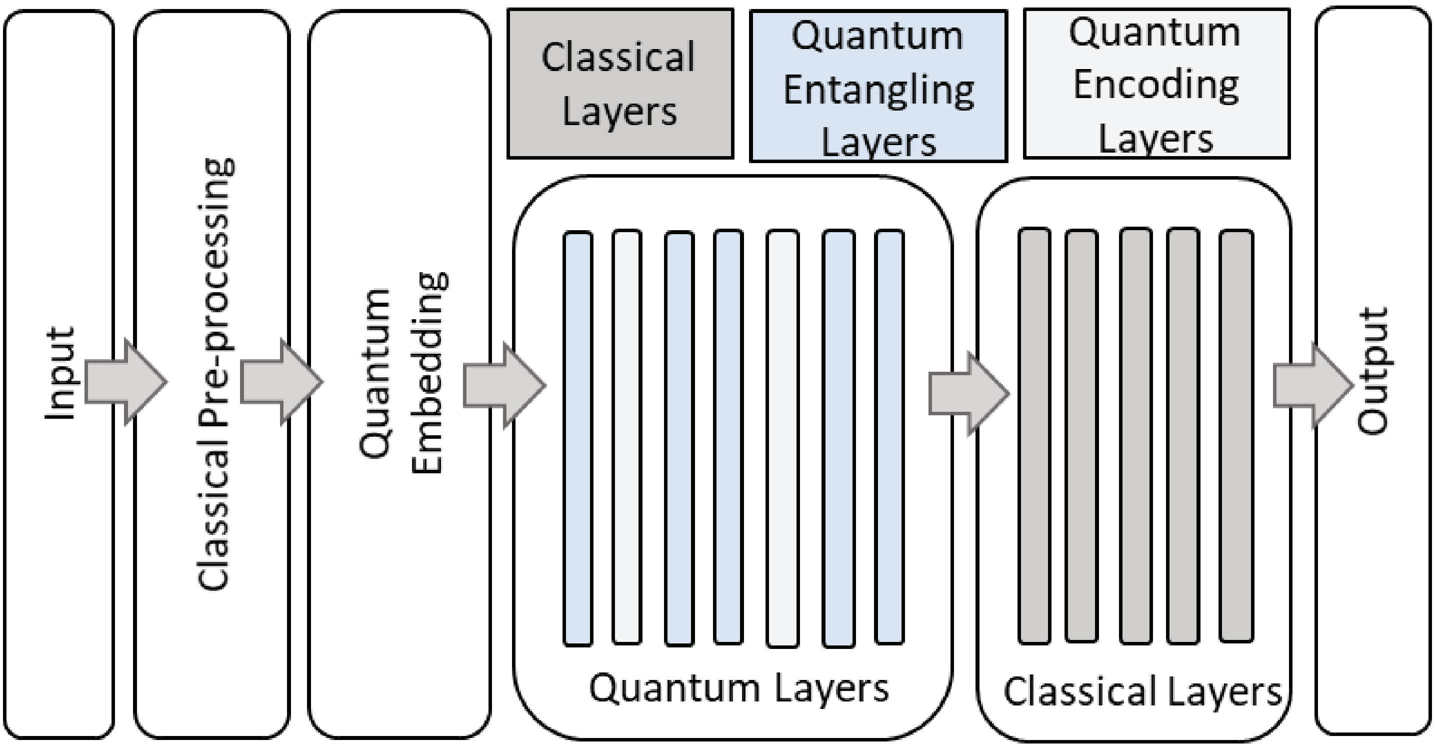
Compared to traditional ANNs quantum circuits are differentiable. This allows us to compute the changes in the control parameters to make the QNN better at a given task.

Quantum Convolutional Neural Networks



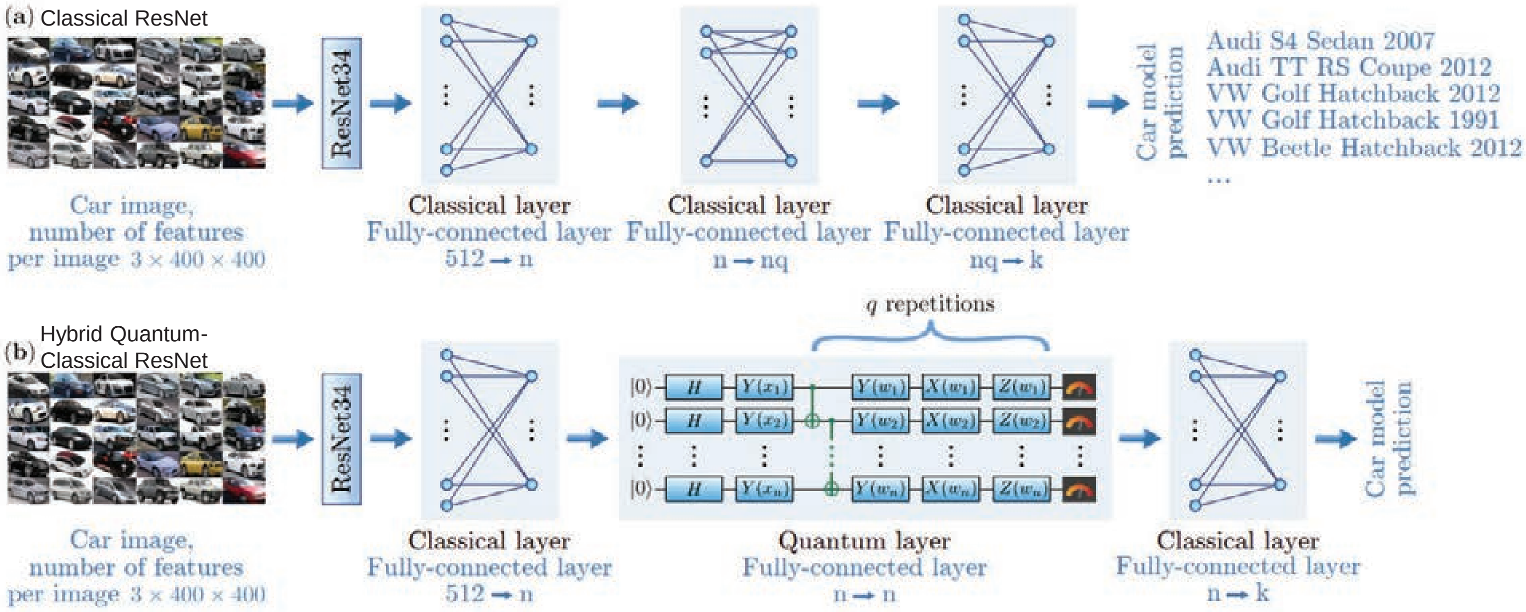
22

Quanvolutional Neural Networks



23

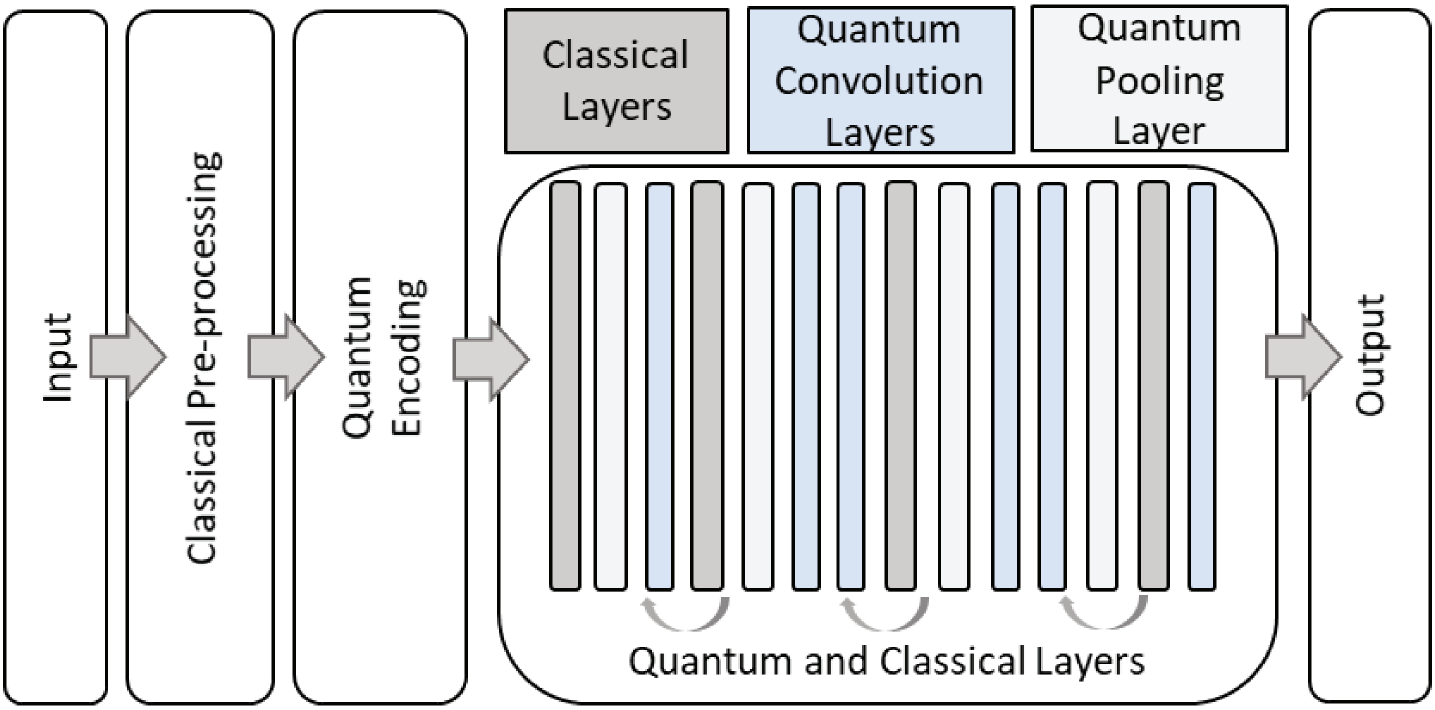
Quantum ResNet



Sagingalieva, A. et al. Hybrid quantum ResNet for car classification and its hyperparameter optimization. Quantum Mach. Intell. 5, 38 (2023)

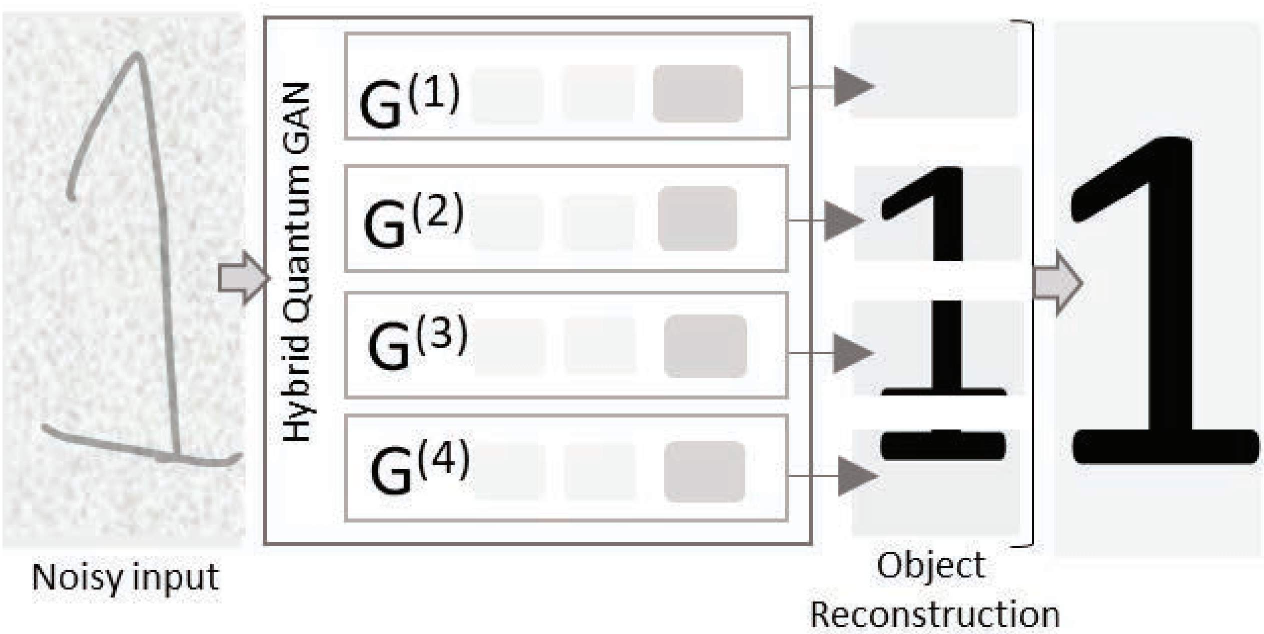
24

Quantum Recurrent Neural Networks



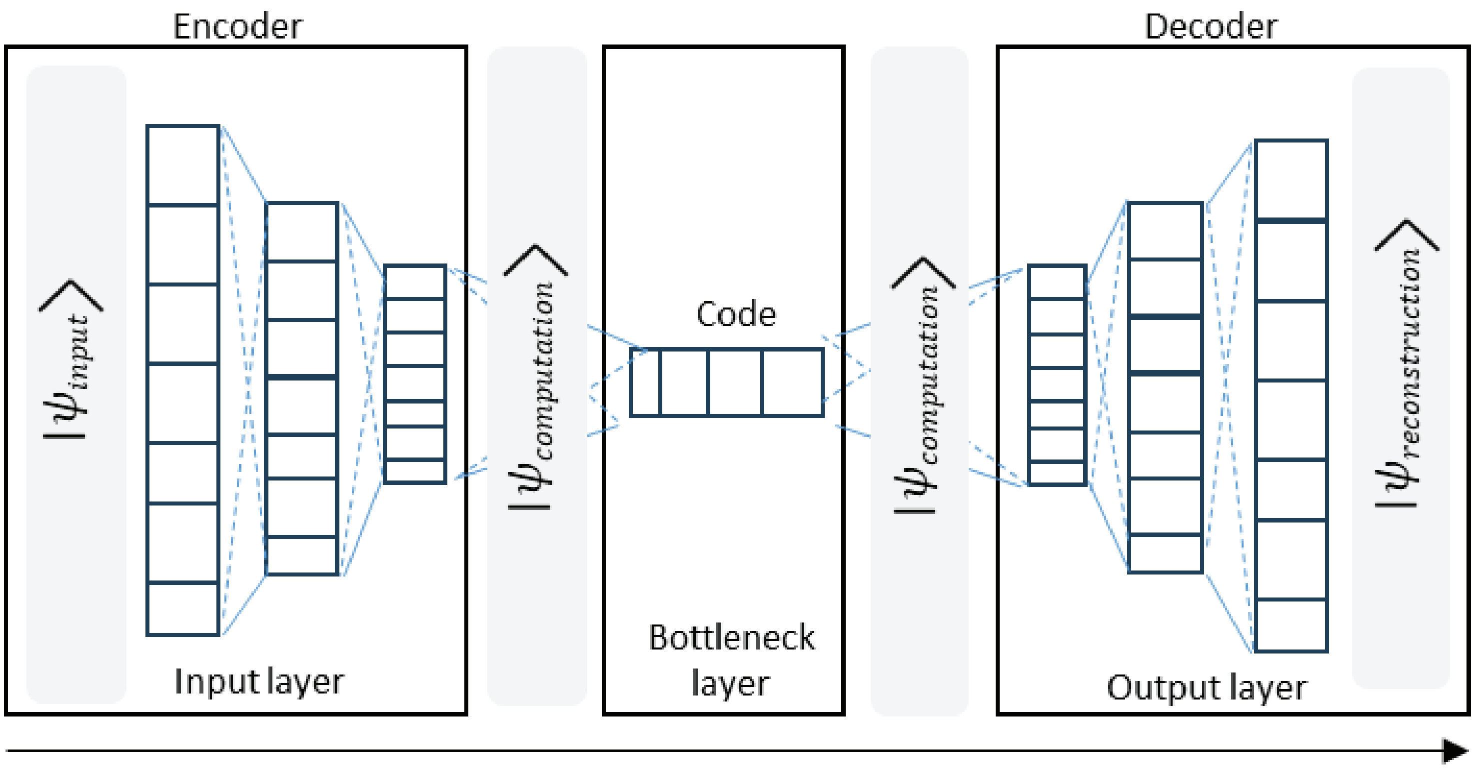
25

Quantum Generative Adversarial Networks



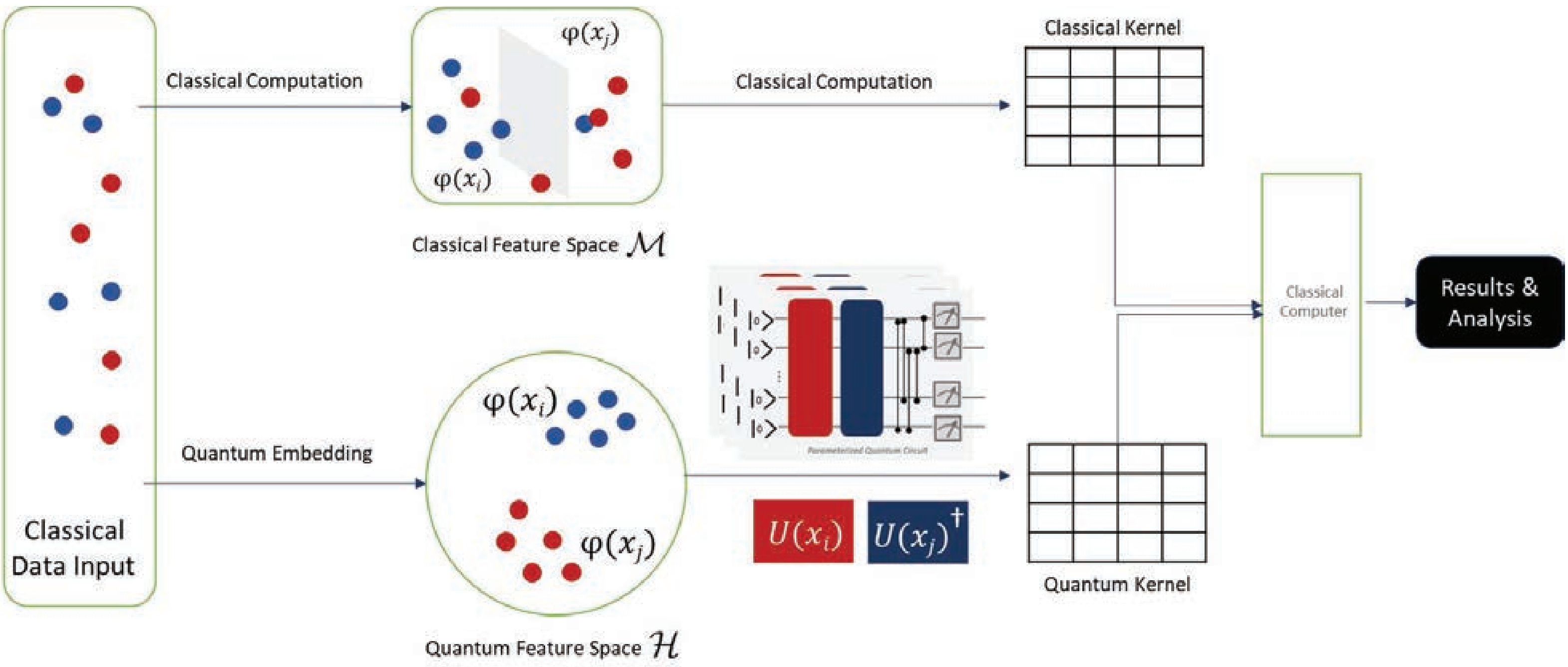
26

Quantum Autoencoders



27

Quantum Kernels



28

Quantum Data Preparation

29

Generic vs. Quantum Pre-Processing Pipeline

Generic Pre-Processing Pipeline

1 Data Cleaning
- Missing Data
- Noisy Values
- Removing Outliers

2 Data Integration
- Unifying data sources
- Consistency

3 Data Transformation
- Generalization
- Normalization
- Attribute Selection
- Aggregation

4 Data Reduction
- Dimensionality reduction
- Data cube aggregation
- Data compression
- Discretization
- Numerosity reduction
- Attribute subset selection

5 Data Quality Assessment
- Completeness
- Information accuracy and reliability
- Feature consistency
- Data Validity

6 Data Quality Assurance
- Data profiling
- Data cleaning
- Data monitoring

Quantum Pre-Processing Pipeline

1 Data Collection & Identification
- Classical data
- Data Cleaning
- Identify nature of data (discrete, continuous, complex)

2 Exploratory Data Analysis
- Clustering
- Normalization
- Correlation Plots

3 Dimensionality Reduction
- Feature Selection
- Feature Extraction
- Principal Component Analysis
- Linear Discriminant Analysis

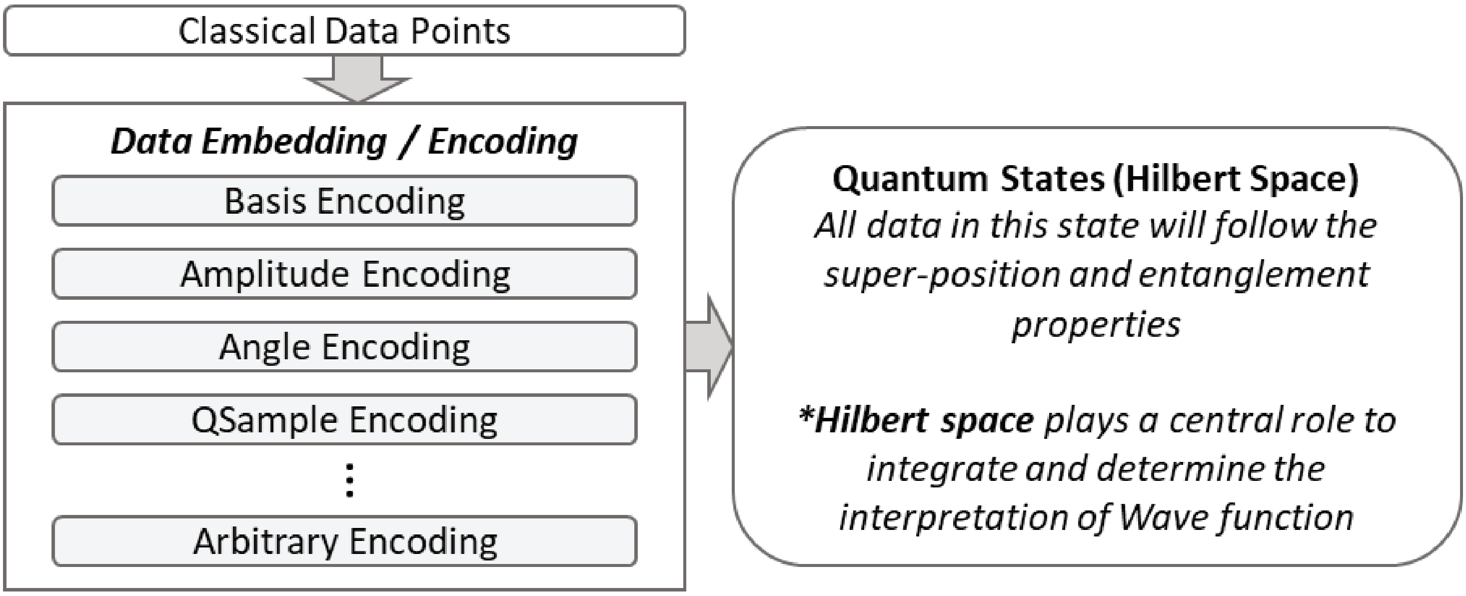
4 Quantum Encoding
- Transform classical information into quantum states
- Data as state representations

5 NISQ Algorithm
Apply the NISQ algorithm with respect to problem at hand

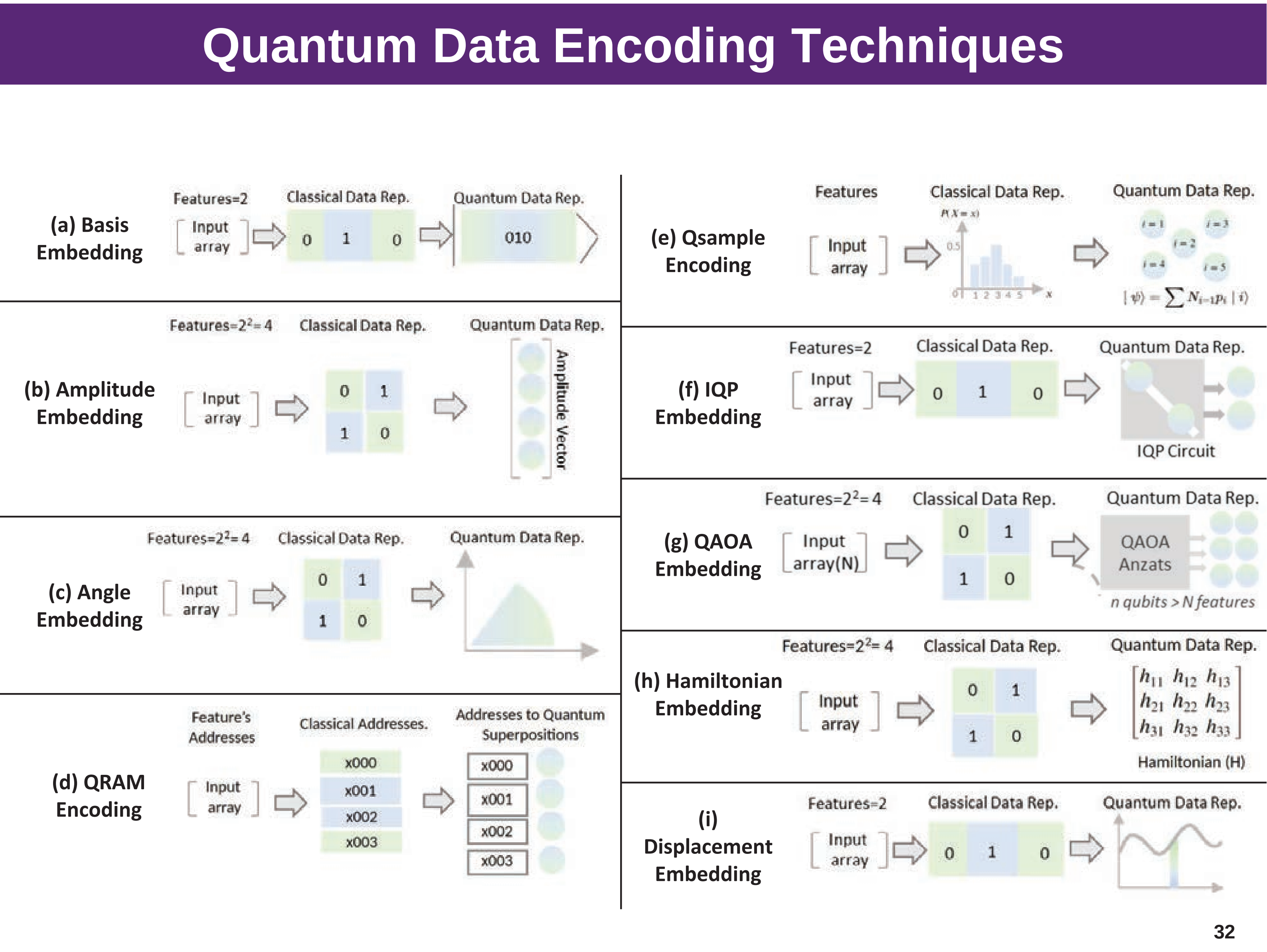
6 Extracting Perspective
- Quantum to Classical Mapping
- Result interpretation

30

Quantum State Preparation Model



31



Quantum Data Encoding Techniques
(a) Basis Embedding
Features=2
Classical Data Rep.
Quantum Data Rep.
Input array
0 1 0
010
(b) Amplitude Embedding
Features=2²=4
Classical Data Rep.
Quantum Data Rep.
Input array
0 1
1 0
Amplitude Vector
(c) Angle Embedding
Features=2²=4
Classical Data Rep.
Quantum Data Rep.
Input array
0 1
1 0
(d) QRAM Encoding
Feature's Addresses
Classical Addresses.
Addresses to Quantum Superpositions
Input array
x000
x001
x002
x003
(e) Qsample Encoding
Features
Classical Data Rep.
Quantum Data Rep.
Input array
(f) IQP Embedding
Features=2
Classical Data Rep.
Quantum Data Rep.
Input array
0 1 0
IQP Circuit
(g) QAOA Embedding
Features=2²=4
Classical Data Rep.
Quantum Data Rep.
Input array(N)
0 1
1 0
QAOA Anzats
n qubits > N features
(h) Hamiltonian Embedding
Features=2²=4
Classical Data Rep.
Quantum Data Rep.
Input array
0 1
1 0
Hamiltonian (H)
(i) Displacement Embedding
Features=2
Classical Data Rep.
Quantum Data Rep.
Input array
0 1 0
32

Quantum HW-SW Support

33

Quantum Hardware Technologies

Superconducting Qubit → IBM Quantum, rigetti, Google, D:Wave

Trapped Ion Qubit → IONQ, Honeywell, QUANTINUUM, oxford ionics, eleQtron

Neutral Atom → atom computing, PASQAL, |QuEra⟩ COMPUTING INC., ColdQuanta

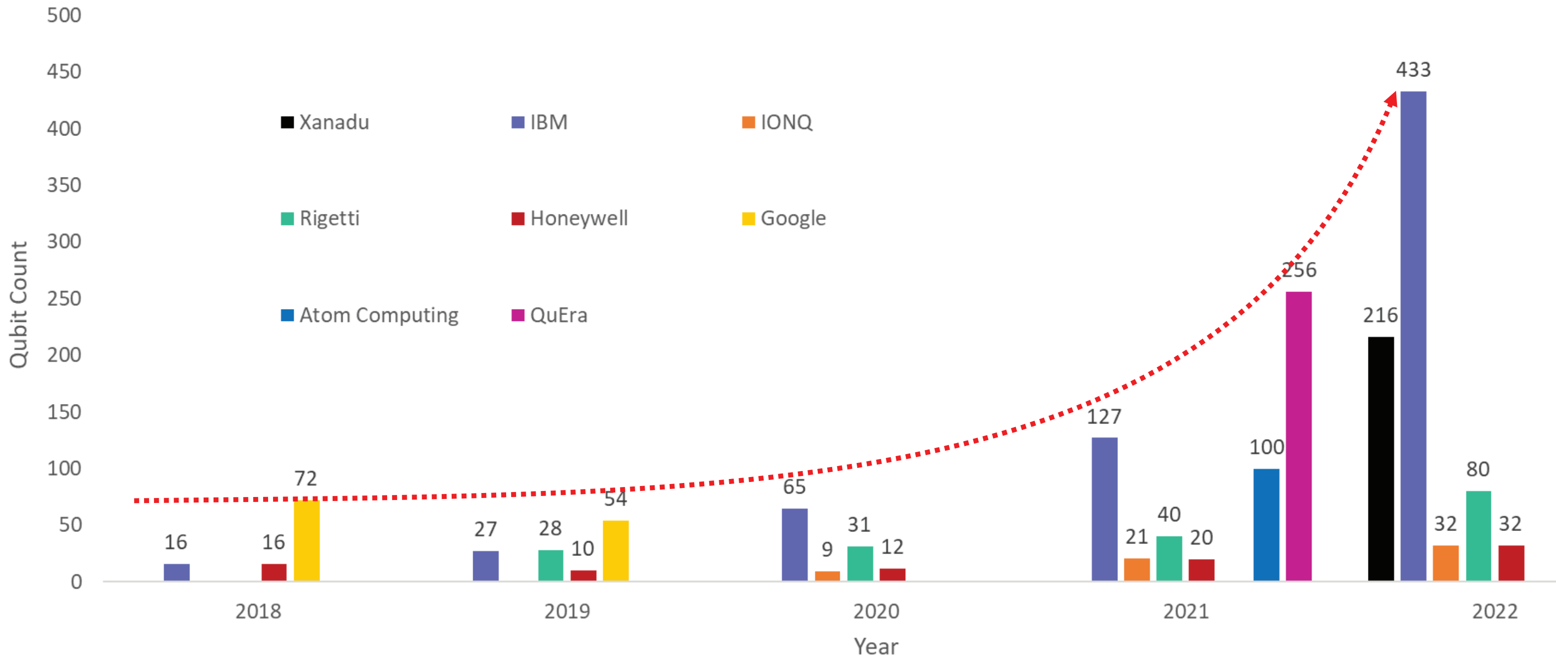
Photonic Qubit → XANADU, ORCA COMPUTING, PsiQuantum

Topological Qubit → Microsoft Azure

Spin Qubits → intel

34

Quantum Hardware Chips



35

Quantum Simulators

Qiskit

By IBM Quantum

Features	*Application Modules*		*Community Engagement*
Open Source	Qiskit Machine Learning	Qiskit Runtime	Access on IBMQ Cloud Experience
		Qiskit Finance	
Quantum Community	Qiskit Optimization	Qiskit Nature	Accessible Learning and Educational resources
Circuit Visualization	Qiskit Experiments	Qiskit Composer	Target Audience - Beginner to Advanced

Pennylane

By Xanadu

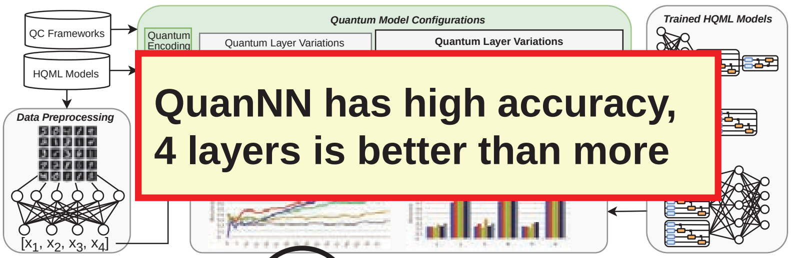
Features	*Application Modules*		*Community Engagement*
Documented	Quantum Datasets	Quantum circuits auto differentiation	Open-Access through Xanadu Cloud
Strong Community			
Research Focused	Quantum Chemistry	Interfaces with other quantum languages	Target Audience - Beginner to Advanced
Open Source	Quantum ML	Useful Templates	Tutorials and Learning

36

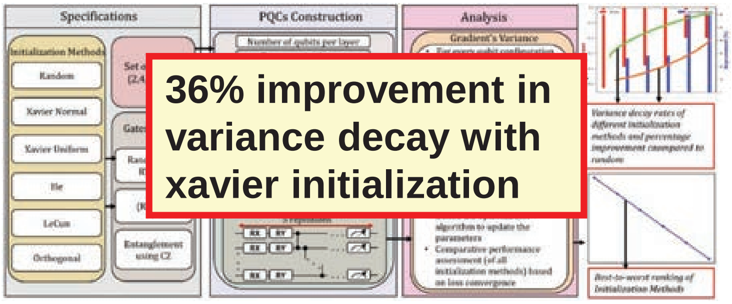
QML Research @ eBrain Lab

37

Quantum Neural Networks Research



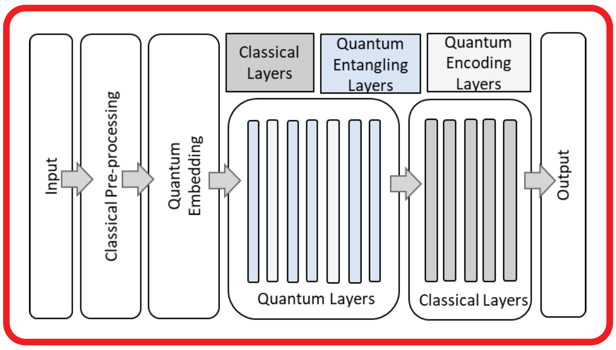
QuanNN has high accuracy, 4 layers is better than more



36% improvement in variance decay with xavier initialization

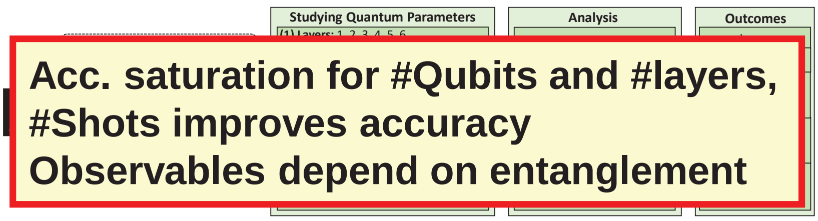
1

Design Space Exploration of QNN Architectures (arxiv.org/abs/2402.10540)



3

Alleviating Barren Plateaus in QNNs (DATE'24, arxiv.org/2311.13218)



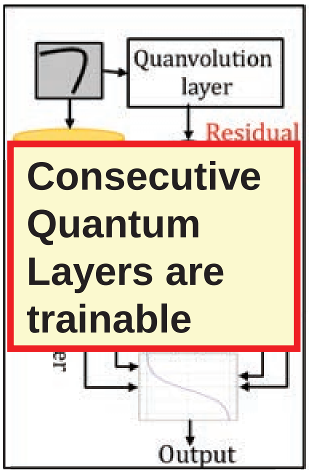
Acc. saturation for #Qubits and #layers,
#Shots improves accuracy
Observables depend on entanglement

Impact of Quantum-Specific Hyperparameters (arxiv.org/abs/2402.10605)

2

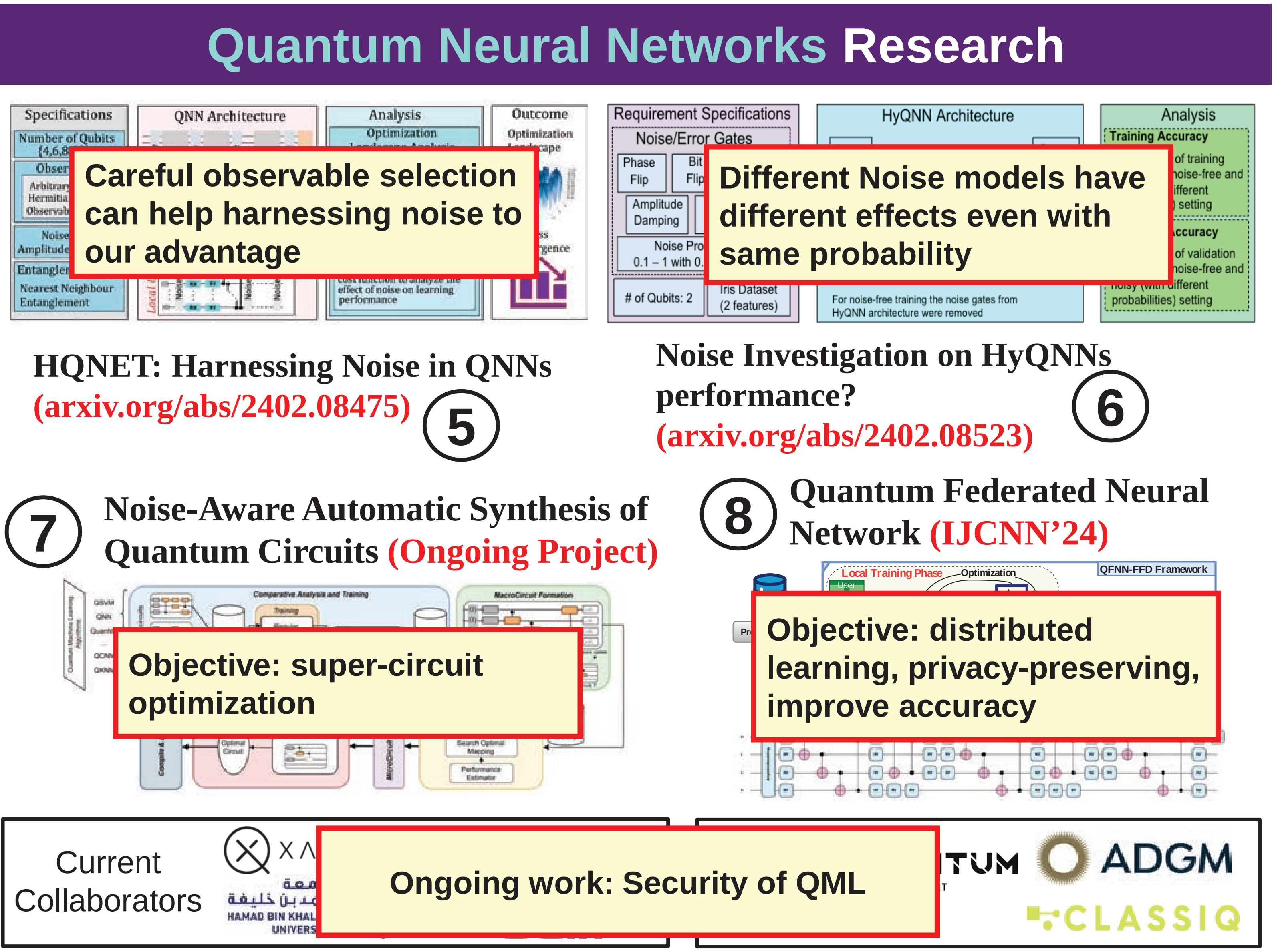
4

ResQuNNs: Residual Quanvolutional Neural Networks (arxiv.org/abs/2402.09146)

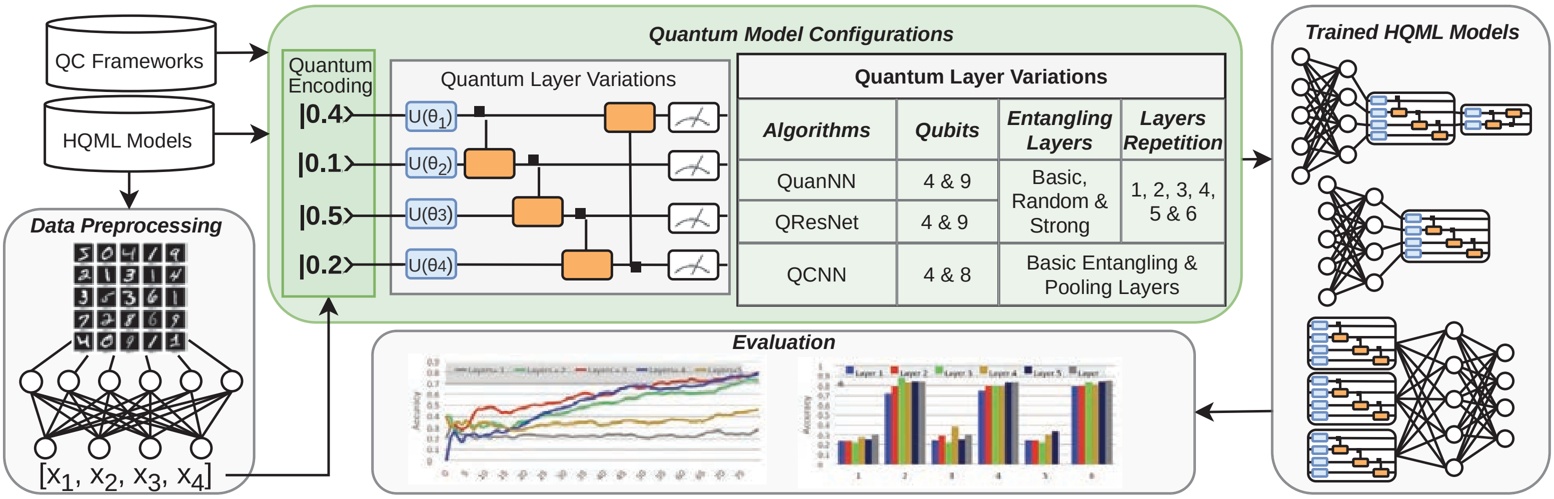


Consecutive Quantum Layers are trainable

38

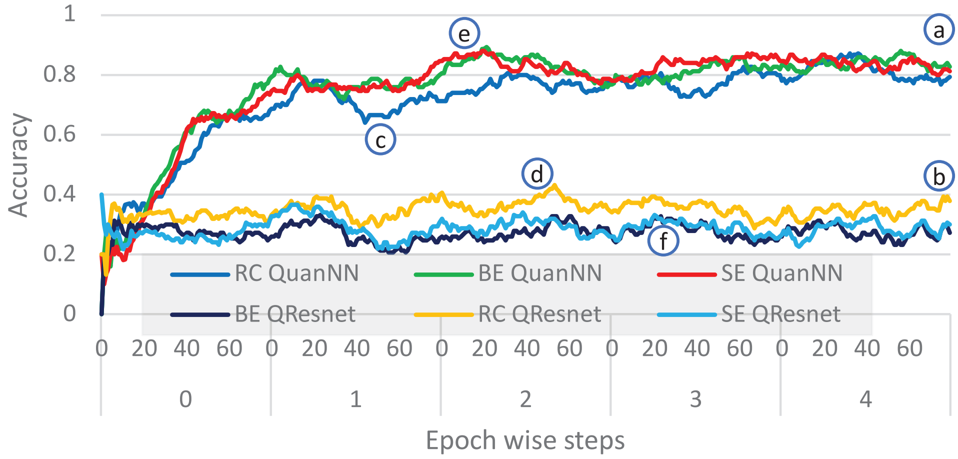

Quantum Neural Networks Research
Specifications
QNN Architecture
Analysis
Outcome
Careful observable selection can help harnessing noise to our advantage
HQNET: Harnessing Noise in QNNs
(arxiv.org/abs/2402.08475)
5
Requirement Specifications
Noise/Error Gates
HyQNN Architecture
Analysis
Training Accuracy
Different Noise models have different effects even with same probability
# of Qubits: 2
Iris Dataset (2 features)
For noise-free training the noise gates from HyQNN architecture were removed
Noise Investigation on HyQNNs performance?
(arxiv.org/abs/2402.08523)
6
7
Noise-Aware Automatic Synthesis of Quantum Circuits (Ongoing Project)
Objective: super-circuit optimization
8
Quantum Federated Neural Network (IJCNN'24)
Local Training Phase
Optimization
QFNN-FFD Framework
Objective: distributed learning, privacy-preserving, improve accuracy
Current Collaborators
Ongoing work: Security of QML
HAMAD BIN KHAL
ADGM
CLASSIQ

Design Space Exploration of QNN Architectures



40

Results: Entangling Circuit Variations



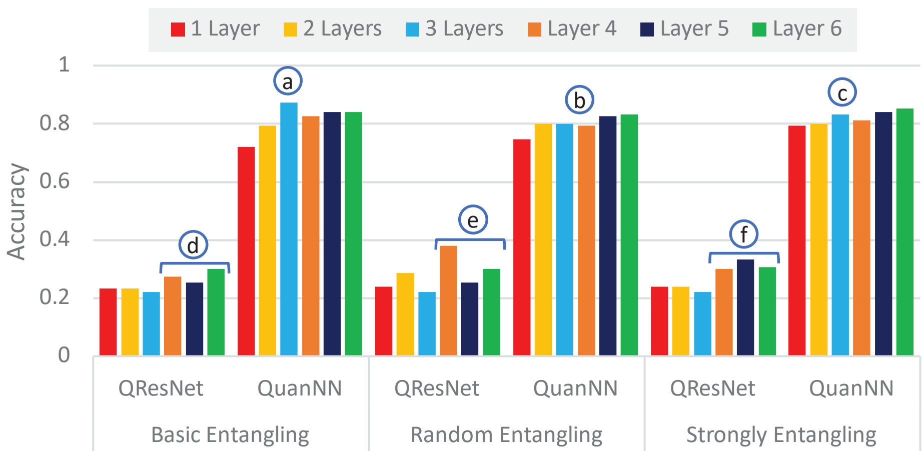
RC: Random Circuit, **SE:** Strongly Entangled, **BE**: Basic Entangling

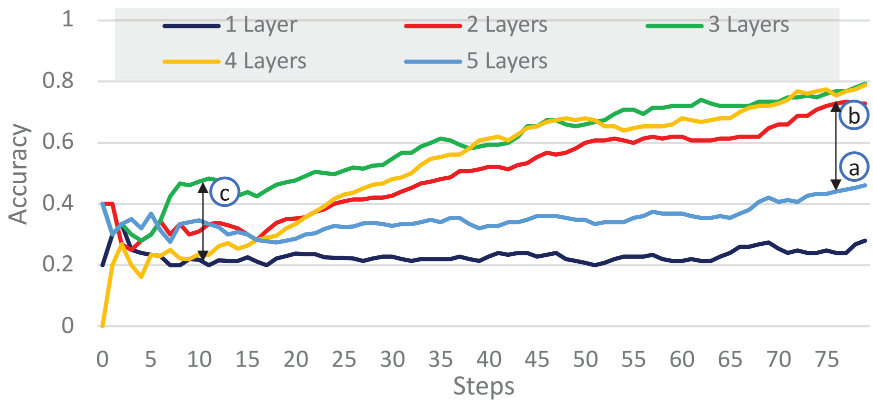
Key Observations

- **(a) and (b):** huge accuracy gap between QuanNN and QResNet.
- **(c):** the Random Circuit has the worst accuracy on the QuanNN.
- **(d):** the Random Circuit has the best accuracy on the QResNet.
- **(e) and (f):** Basic and Strongly Entangling Circuit have similar learning curves for both algorithms

41

Results: Layer Count Variations



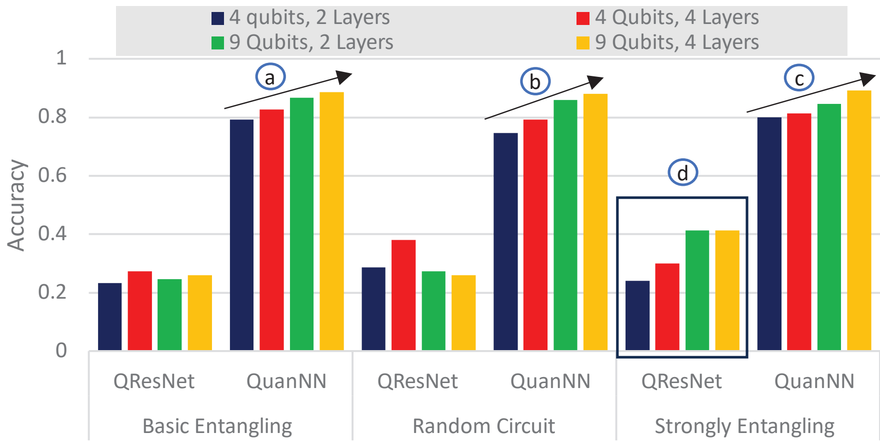


Key Observations

- **(b) and (c):** We can observe an overall trend of improved accuracy with increasing the number of layers, but there are some deviations.
- **(a), (d), (e), and (f):** Adding more than 4 layers does not always contribute to increasing the accuracy.
- **(c):** For 1 layer, the learning curve is mostly constant with little improvement.
- **(a) and (b):** The peak is reached by 4 layers, while the QuanNN with 5 layers has low accuracy.

42

Results: Qubit Count Variations

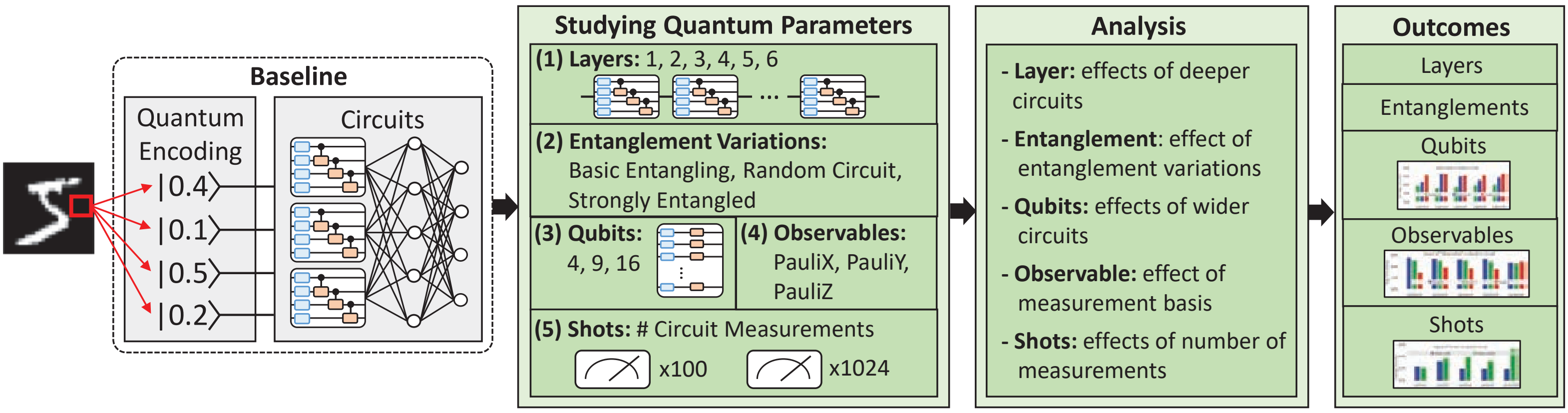


Key Observations

- **(a), (b) and (c):** positive correlation between number of qubits and accuracy of the QuanNN.
- the QResNet with Basic Entangling and Random Circuit does not have any significant trend.
- **(d):** the Strongly Entangling QResNet shows an increase in accuracy with increasing its qubit count and layers.

43

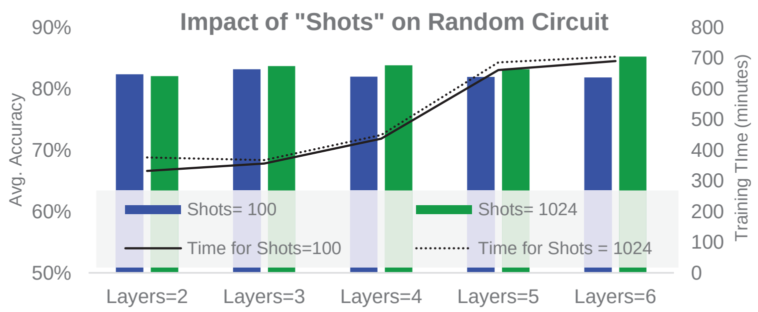
Impact of Quantum-Specific Hyperparameters

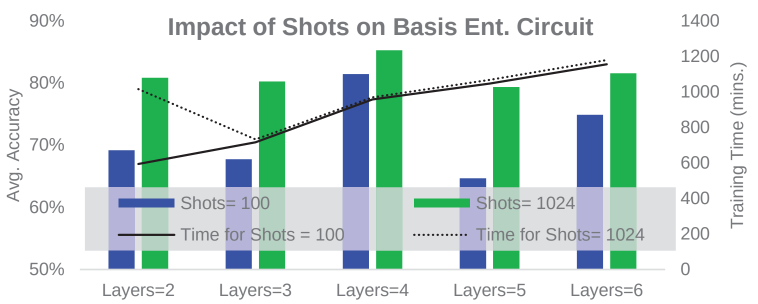


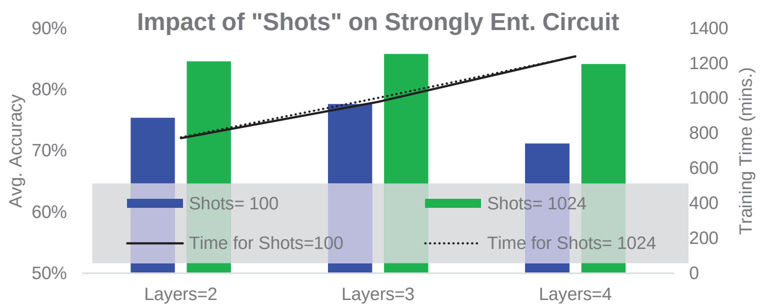
- **Number of layers**: how many repetitions of the quantum circuit.
- **Type of Layer**: entanglement variation
- **Number of Qubits**
- **Type of observable**: By changing the measurement basis of a circuit, we can observe the impact of different observables on the measurement outcome of a model.
- **Shots:** number of times a circuit is executed and measured.

44

Results: Impact of Shots





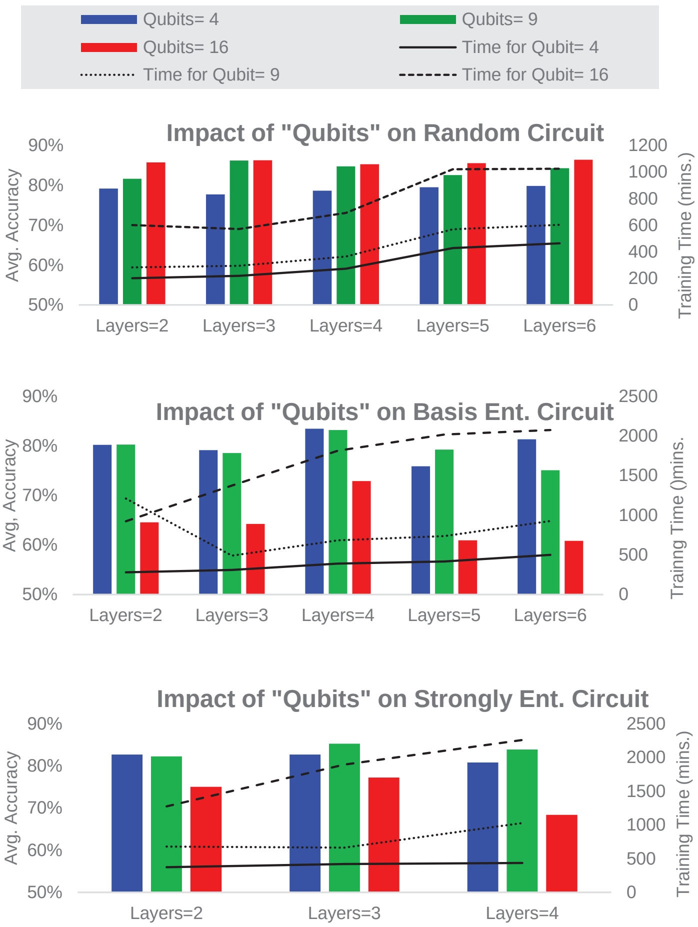


Key Observations

- ❑ Higher number of shots always improving performance
- ❑ Random Circuit: Insignificant performance gap
- ❑ Basic Entangling Circuit: Significant Performance gap
- ❑ Strongly Entangling Circuit: Significant Performance gap

45

Results: Impact of Qubit Count

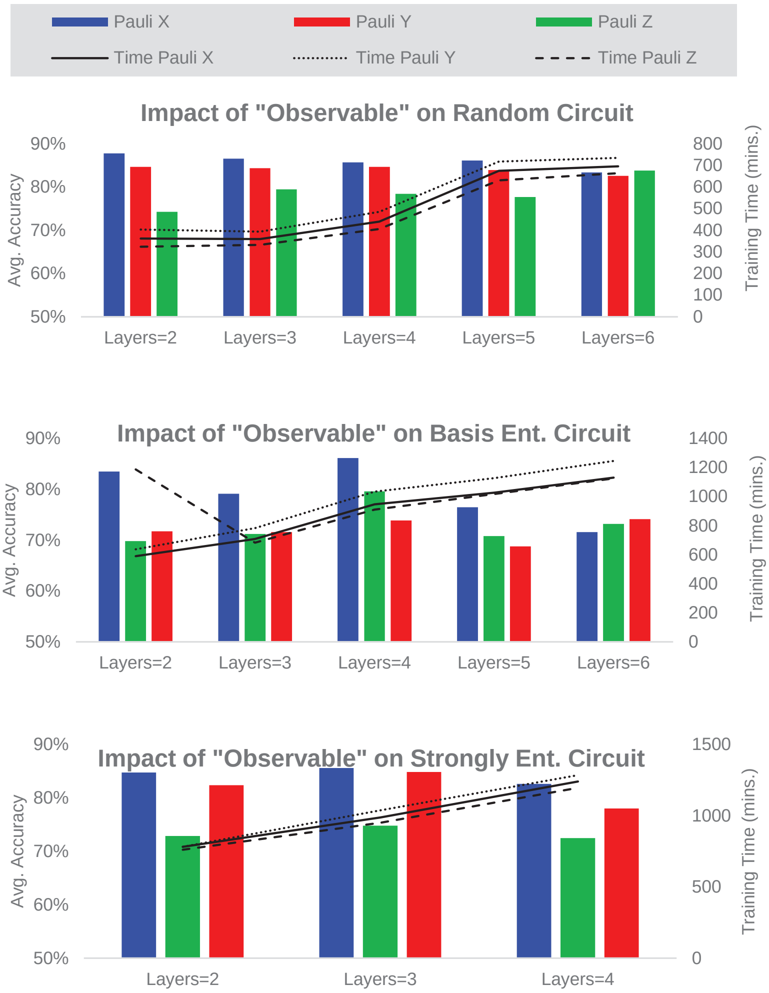


Key Observations

- Time variation with number of qubits is very high
- Random Circuit: Consistent positive correlation in accuracy and time when increasing the qubits, without drastic time differences
- Basic Entangling Circuit: Significant performance variation on accuracy and time, best results for 4 layers
- Strongly Entangling Circuit: Good performance for 3 layers and 9 qubits

46

Results: Impact of Measurement Observables

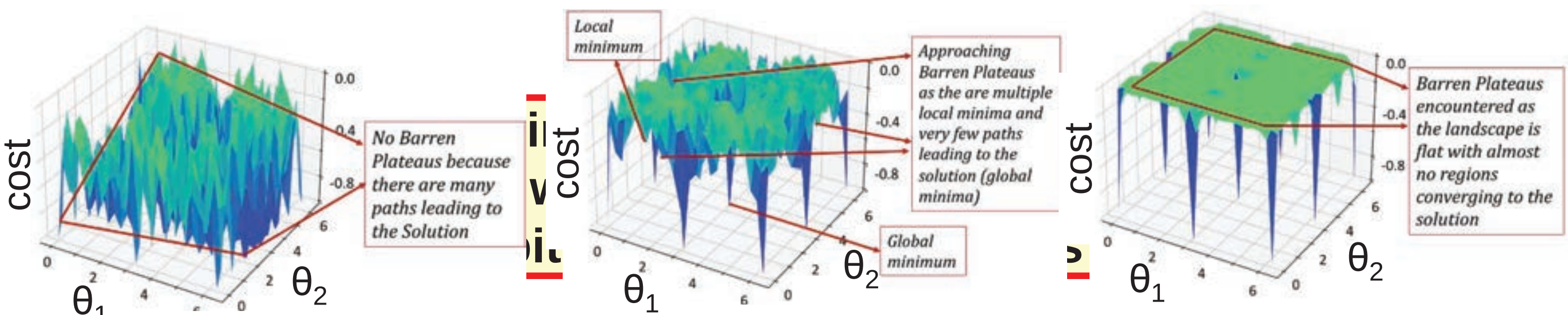


Key Observations

- The Pauli X has the best impact on the accuracy.
- The Pauli Z has the lowest training time cost, followed by Pauli X.
- Random Circuit: Pauli Y has second-highest accuracy.
- Basic Entangling Circuit: both Pauli Y and Pauli Z have accuracy drop compared to Pauli X.
- Strongly Entangling Circuit: Pauli Y has the worst accuracy, while Pauli X and Pauli Z have similar accuracy.

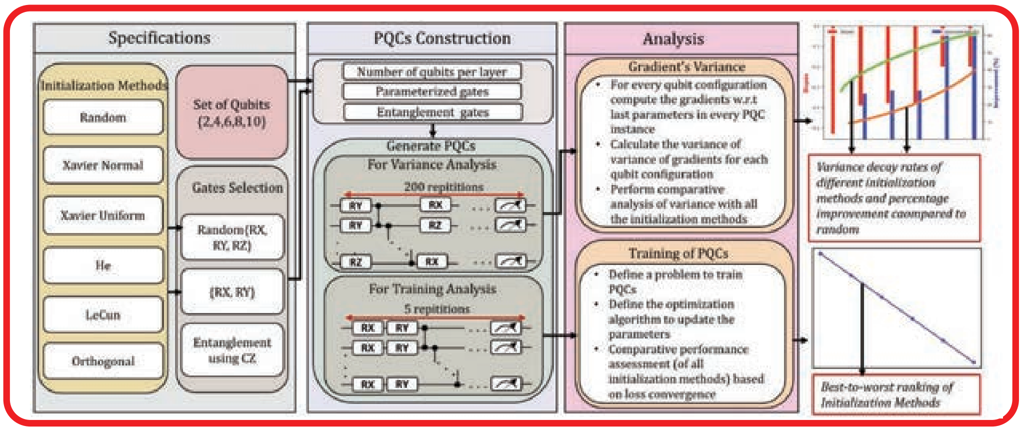
47

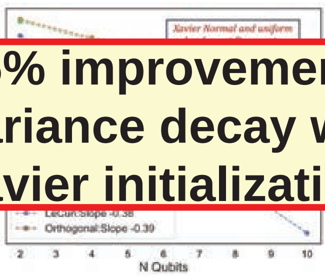
Alleviating Barren Plateaus in QNNs Research



2 Qubits **5 Qubits** **10 Qubits**

Alleviating barren plateaus in QNNs (DATE’24)



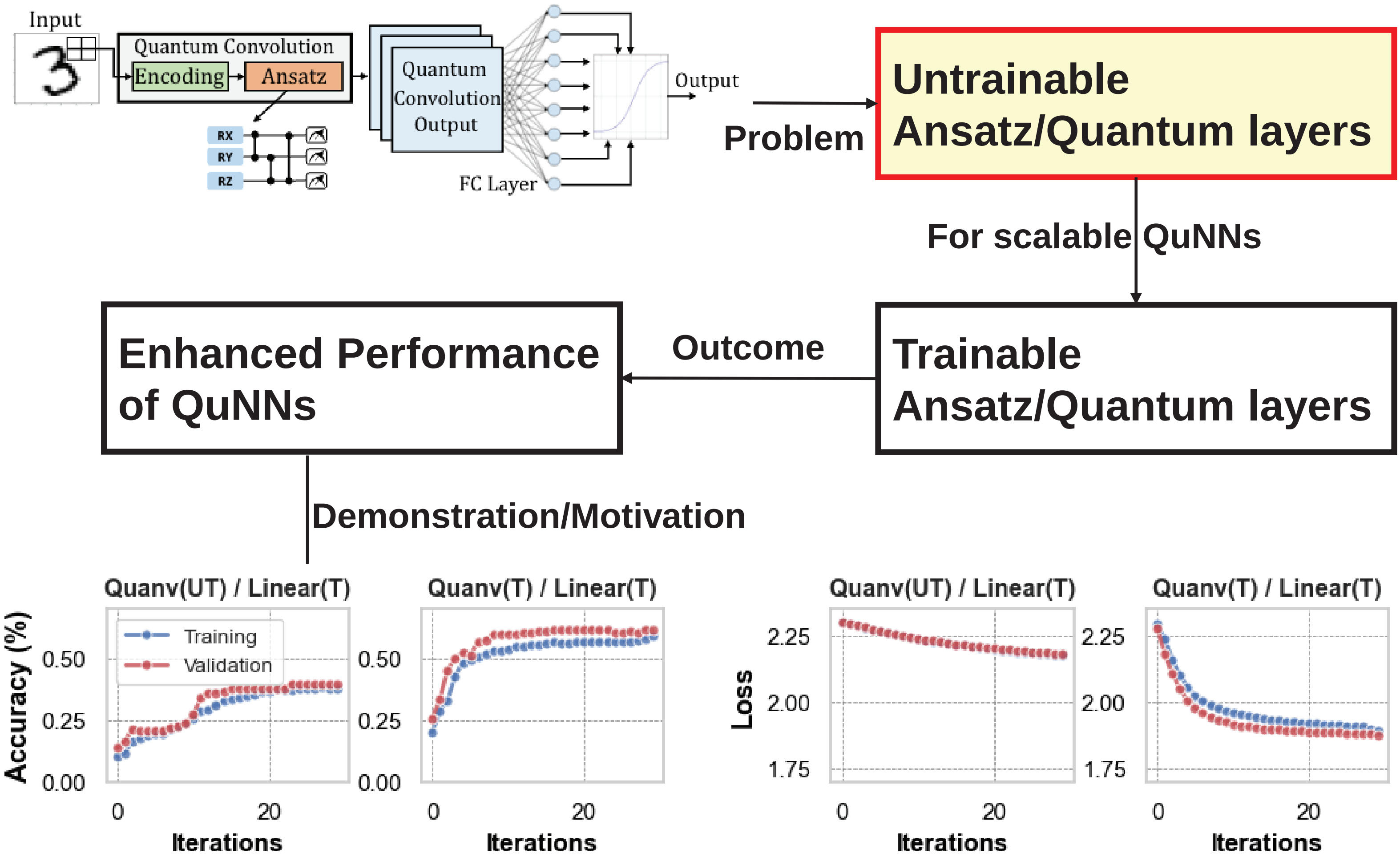


36% improvement in variance decay with xavier initialization

All initialization methods results in improved training compared to random initialization with xavier being the best

48

ResQuNNs Research



49

ResQuNNs Research

Scaling to Multilayered trainable QuNNs

Problem

Gradients Only available for last quanvolutional layer

No gradients for first quanvolutional layer

```
Epoch: 0

Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : None
qconv2.circuit.weights.grad : tensor([[-4.1302e-04,  9.2646e-04, -8.7912e-04, -2.6830e-04],
        [-1.6980e-03, -3.3707e-03,  6.7682e-04, -1.3442e-03],
        [ 9.0237e-04, -9.0989e-04,  1.6334e-03,  8.7910e-04],
        [ 3.0149e-03, -1.4837e-03,  1.4948e-03,  2.9540e-03],
        [-7.1405e-20, -1.1964e-03,  1.2280e-03,  1.6086e-03]])
------------------------------------
```

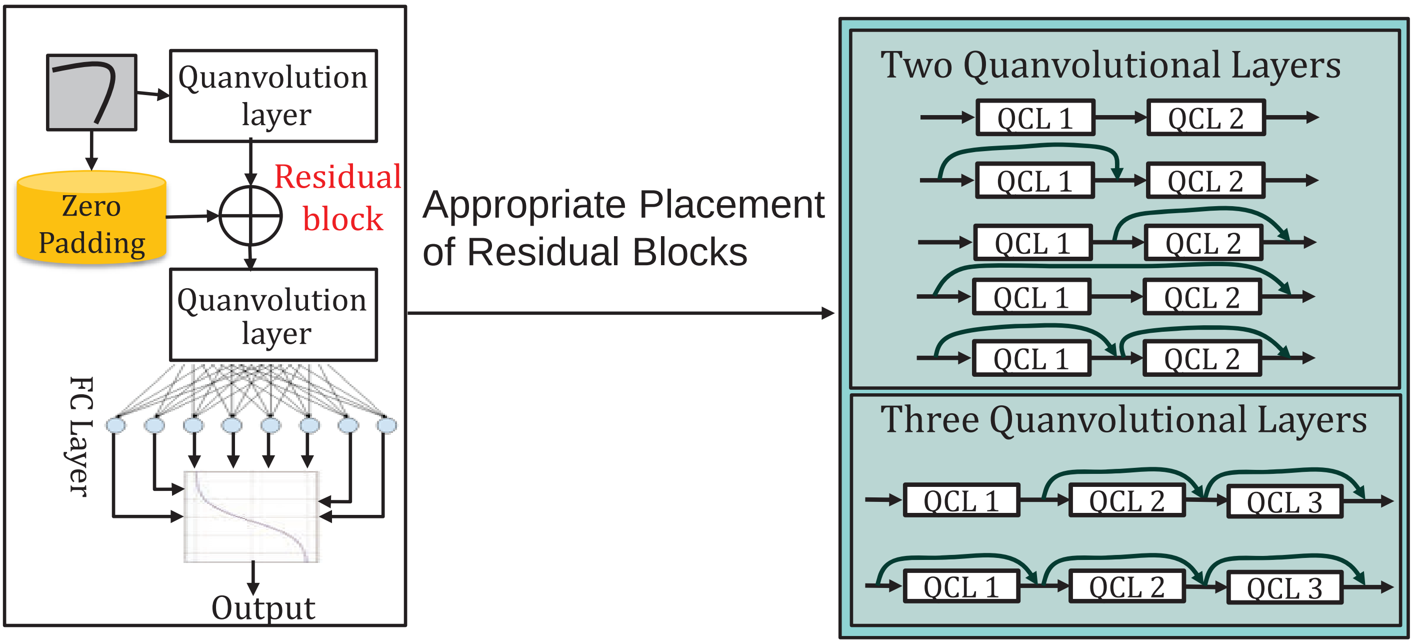
No gradients for first quanvolutional layer

```
Epoch: 0

Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : None
qconv2.circuit.weights.grad : None
qconv3.circuit.weights.grad : tensor([[-3.3361e-04, -5.7636e-04, -2.2633e-04,  1.3984e-04],
        [-1.9798e-04,  3.9540e-04,  1.0374e-04,  3.0715e-04],
        [ 1.6112e-05, -7.0889e-04,  9.2504e-05,  6.1045e-04],
        [-4.3064e-04, -3.6777e-04,  2.5115e-04,  5.5998e-05],
        [ 6.0384e-20, -3.6117e-05,  2.5950e-04, -7.7258e-04]])
------------------------------------
```

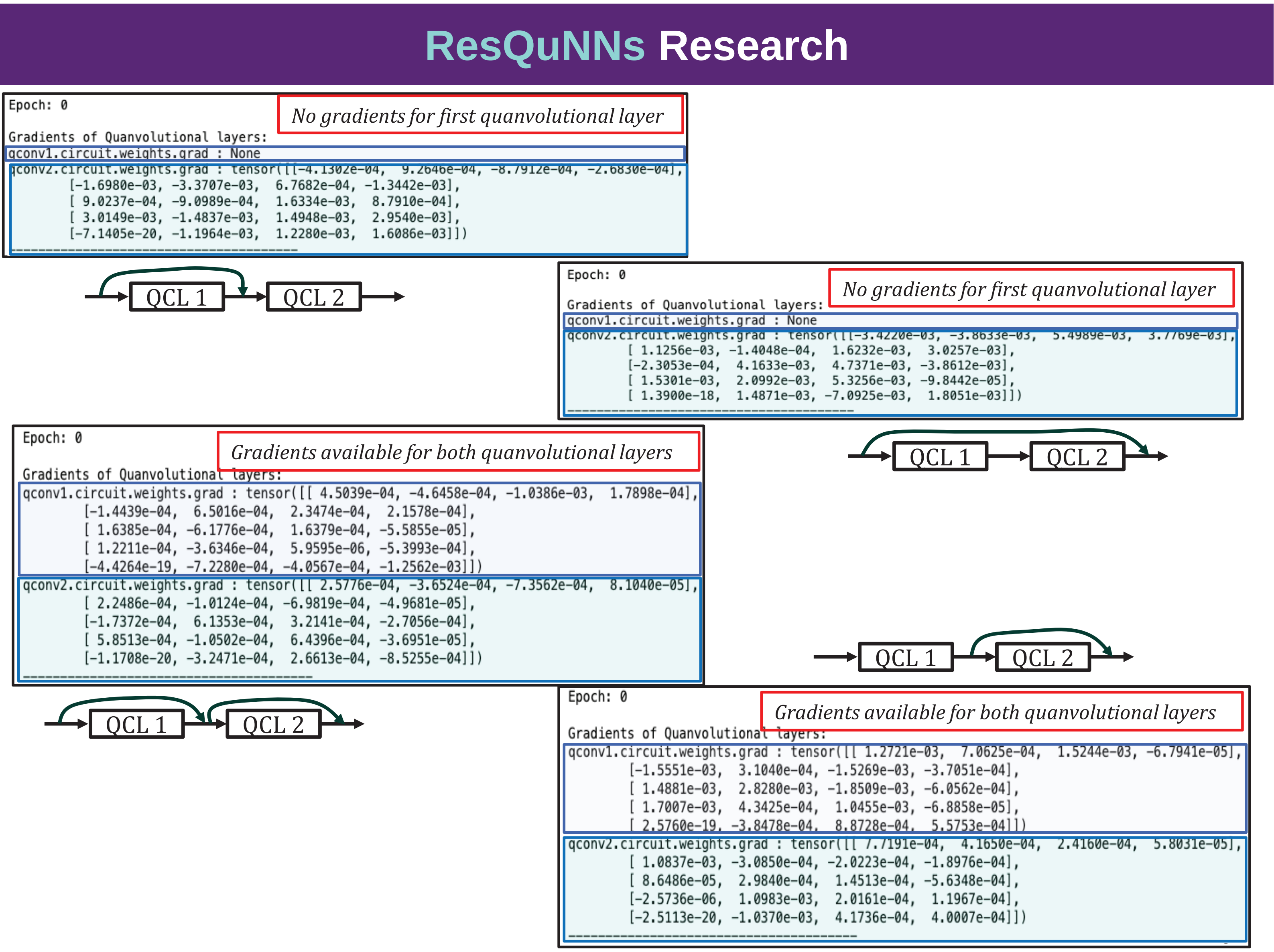
50

ResQuNNs Research

Proposed ResQuNns to overcome gradient accessibility issue



51


ResQuNNs Research
Epoch: 0
No gradients for first quanvolutional layer
Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : None
qconv2.circuit.weights.grad : tensor([[-4.1302e-04, 9.2646e-04, -8.7912e-04, -2.6830e-04],
[-1.6980e-03, -3.3707e-03, 6.7682e-04, -1.3442e-03],
[9.0237e-04, -9.0989e-04, 1.6334e-03, 8.7910e-04],
[3.0149e-03, -1.4837e-03, 1.4948e-03, 2.9540e-03],
[-7.1405e-20, -1.1964e-03, 1.2280e-03, 1.6086e-03]])
QCL 1
QCL 2
Epoch: 0
No gradients for first quanvolutional layer
Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : None
[1.1256e-03, -1.4048e-04, 1.6232e-03, 3.0257e-03],
[-2.3053e-04, 4.1633e-03, 4.7371e-03, -3.8612e-03],
[1.5301e-03, 2.0992e-03, 5.3256e-03, -9.8442e-05],
[1.3900e-18, 1.4871e-03, -7.0925e-03, 1.8051e-03]])
Epoch: 0
Gradients available for both quanvolutional layers
Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : tensor([[4.5039e-04, -4.6458e-04, -1.0386e-03, 1.7898e-04],
[-1.4439e-04, 6.5016e-04, 2.3474e-04, 2.1578e-04],
[1.6385e-04, -6.1776e-04, 1.6379e-04, -5.5855e-05],
[1.2211e-04, -3.6346e-04, 5.9595e-06, -5.3993e-04],
[-4.4264e-19, -7.2280e-04, -4.0567e-04, -1.2562e-03]])
qconv2.circuit.weights.grad : tensor([[2.5776e-04, -3.6524e-04, -7.3562e-04, 8.1040e-05],
[2.2486e-04, -1.0124e-04, -6.9819e-04, -4.9681e-05],
[-1.7372e-04, 6.1353e-04, 3.2141e-04, -2.7056e-04],
[5.8513e-04, -1.0502e-04, 6.4396e-04, -3.6951e-05],
[-1.1708e-20, -3.2471e-04, 2.6613e-04, -8.5255e-04]])
QCL 1
QCL 2
QCL 1
QCL 2
QCL 1
QCL 2
Epoch: 0
Gradients available for both quanvolutional layers
Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : tensor([[1.2721e-03, 7.0625e-04, 1.5244e-03, -6.7941e-05],
[-1.5551e-03, 3.1040e-04, -1.5269e-03, -3.7051e-04],
[1.4881e-03, 2.8280e-03, -1.8509e-03, -6.0562e-04],
[1.7007e-03, 4.3425e-04, 1.0455e-03, -6.8858e-05],
[2.5760e-19, -3.8478e-04, 8.8728e-04, 5.5753e-04]])
qconv2.circuit.weights.grad : tensor([[7.7191e-04, 4.1650e-04, 2.4160e-04, 5.8031e-05],
[1.0837e-03, -3.0850e-04, -2.0223e-04, -1.8976e-04],
[8.6486e-05, 2.9840e-04, 1.4513e-04, -5.6348e-04],
[-2.5736e-06, 1.0983e-03, 2.0161e-04, 1.1967e-04],
[-2.5113e-20, -1.0370e-03, 4.1736e-04, 4.0007e-04]])

ResQuNNs Research


```
Epoch: 0

Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : tensor([[ 1.3983e-04, -3.6823e-04, -5.3045e-04,  5.4463e-04],
        [-2.0843e-04,  6.7892e-05, -2.5184e-04,  1.0055e-03],
        [-4.8478e-05, -1.1647e-03,  1.4009e-04,  9.5095e-04],
        [-2.4307e-04, -8.2385e-04, -2.2135e-04,  1.0867e-04],
        [-3.0169e-19, -3.4162e-04, -2.9458e-04,  8.7503e-04]])
qconv2.circuit.weights.grad : tensor([[ 1.3837e-04,  5.7586e-04, -9.2042e-04,  2.1189e-04],
        [ 1.3915e-04,  3.0827e-04, -1.0298e-05, -3.3603e-04],
        [-1.6321e-04,  1.3787e-04,  3.9168e-04,  8.4424e-04],
        [-5.1840e-04,  2.9377e-04, -3.4994e-04, -3.3091e-04],
        [ 5.5787e-20, -4.9666e-04, -1.8947e-04, -1.1867e-04]])
qconv3.circuit.weights.grad : tensor([[-6.5886e-05, -5.2884e-04,  6.6197e-04, -6.9840e-05],
        [-1.1867e-04,  1.7709e-04, -5.4581e-05,  7.7833e-05],
        [-3.7169e-04, -3.1033e-04,  2.5479e-04,  6.3815e-04],
        [-1.6536e-04,  3.4741e-06,  1.6655e-04, -2.6275e-04],
        [ 6.8355e-21,  5.2233e-04,  2.5734e-04,  1.8372e-04]])
```

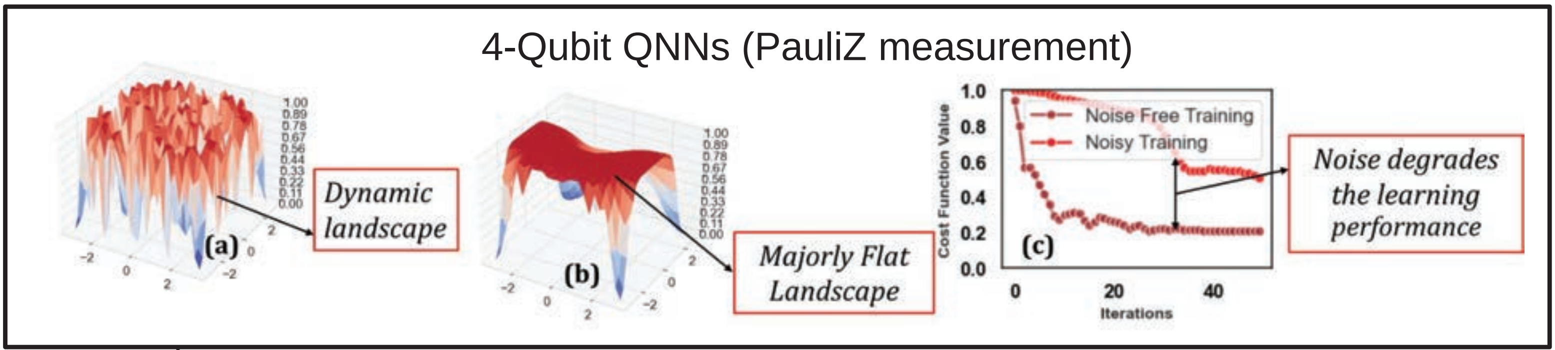
Gradients available for all three quanvolutional layers

```
Epoch: 0

Gradients of Quanvolutional layers:
qconv1.circuit.weights.grad : tensor([[ 4.1779e-04, -3.5562e-04, -4.9648e-04, -1.1720e-04],
        [ 1.0519e-03, -1.5035e-04,  2.3253e-04, -1.6365e-04],
        [ 4.1258e-04,  1.8928e-04, -1.5234e-04,  3.1661e-04],
        [ 1.1174e-04, -6.4044e-04, -5.3017e-04, -1.1546e-04],
        [-3.6694e-19, -3.5474e-04,  7.4533e-04, -1.7833e-04]])
qconv2.circuit.weights.grad : tensor([[ 2.4955e-04, -1.0011e-05,  7.6074e-05,  1.1483e-04],
        [ 3.5497e-05, -5.5178e-04,  1.2842e-05,  1.4225e-05],
        [ 5.7061e-04, -4.1772e-05,  1.7036e-04,  7.0727e-05],
        [ 2.0882e-04,  1.2254e-06, -2.6590e-04, -3.5221e-05],
        [ 1.9641e-19,  5.1496e-04, -1.0349e-04,  1.3623e-04]])
qconv3.circuit.weights.grad : tensor([[ 2.1265e-04, -3.5032e-04,  1.0178e-04,  2.3617e-04],
        [ 6.8907e-05,  7.0250e-05,  2.8331e-04, -3.5580e-04],
        [-2.8655e-05,  5.4647e-04,  3.4060e-04,  1.9941e-04],
        [-8.0421e-05,  2.9545e-05, -1.4645e-04, -1.8066e-05],
        [-1.7398e-19,  9.2292e-05,  2.1506e-04,  5.0952e-04]])
```

53

HQNET Research

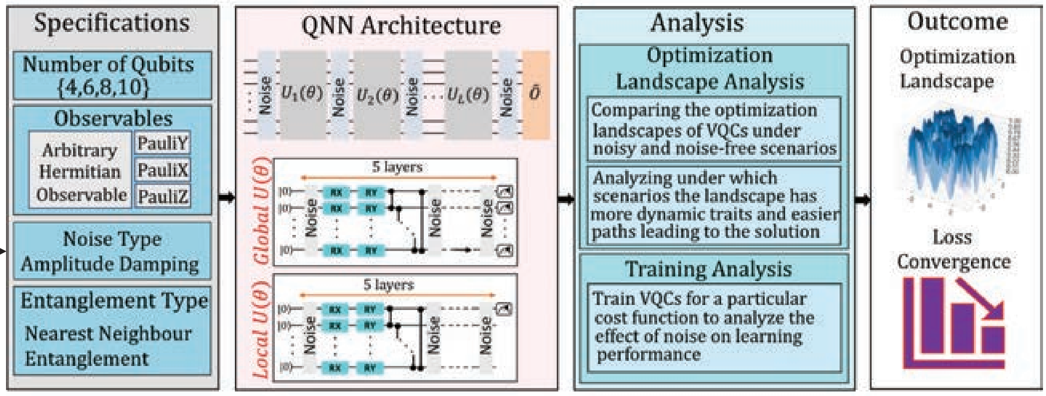
Harnessing Quantum Noise to enhance the training of QNNs



Proposed Solution

Careful observable selection can help harnessing noise to our advantage

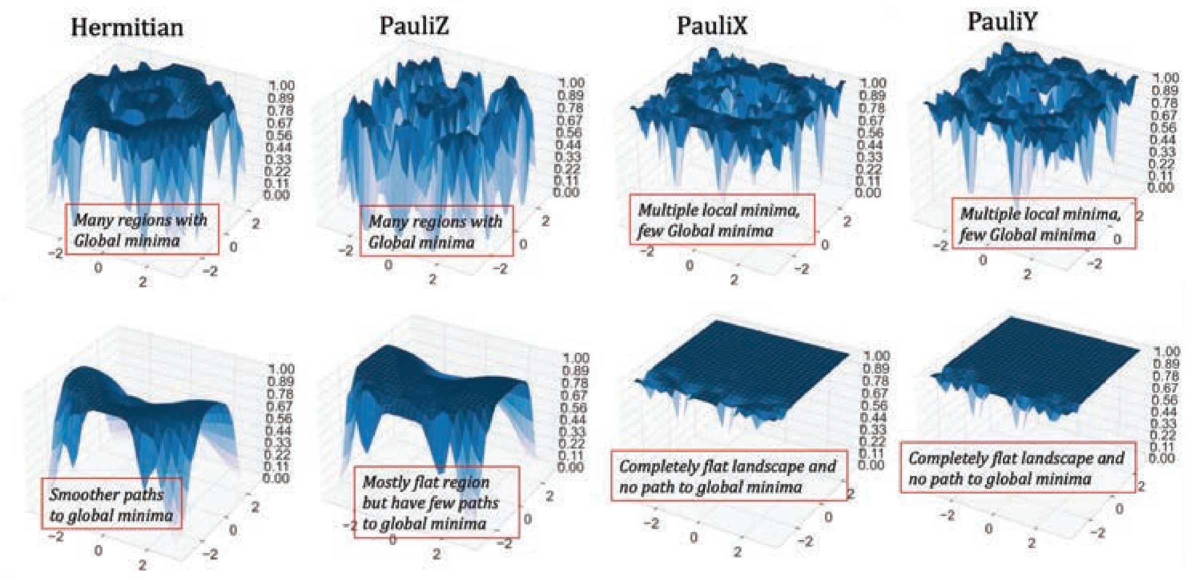
Methodology



https://arxiv.org/abs/2402.08475

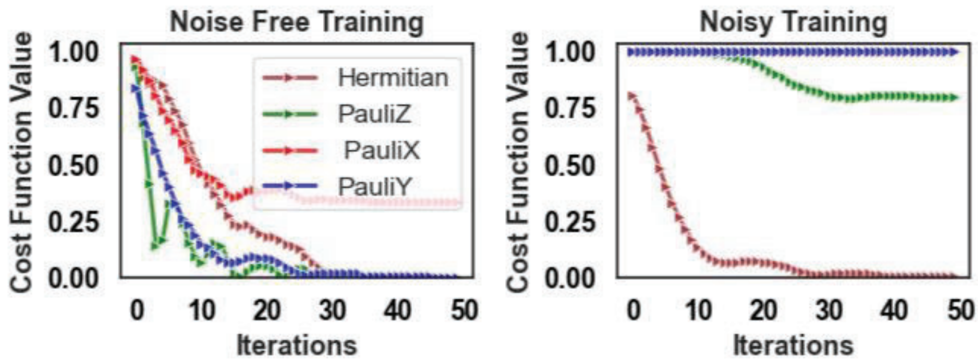
54

HQNET Research



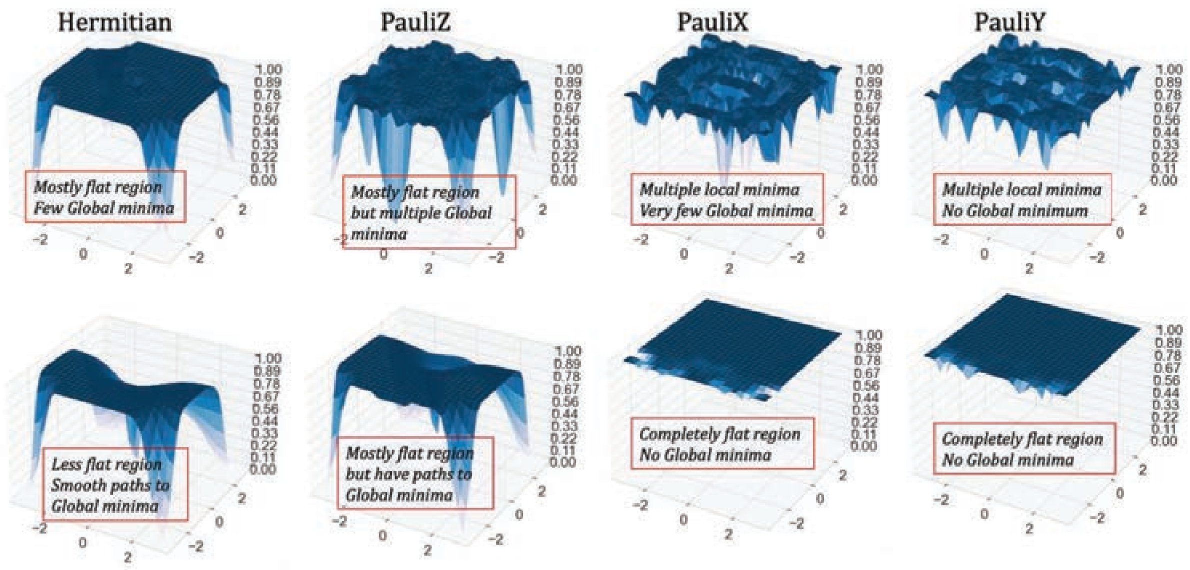
4 Qubits Global QNN Noise Free Training

4 Qubits Global QNN Noisy Training



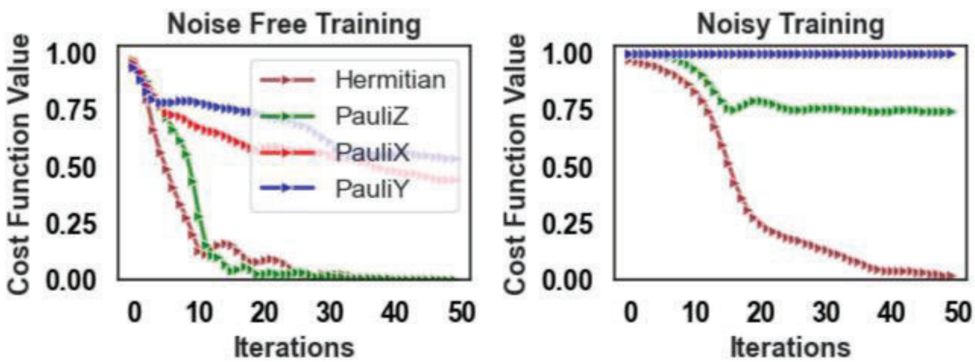
55

HQNET Research



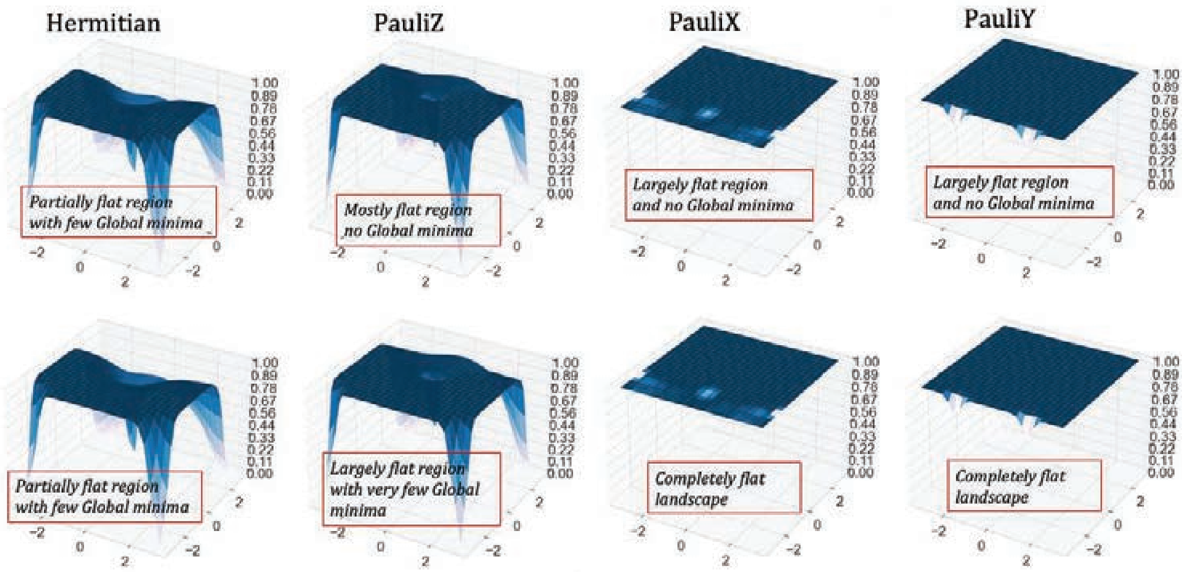
6 Qubits Global QNN
Noise Free Training

6 Qubits Global QNN
Noisy Training



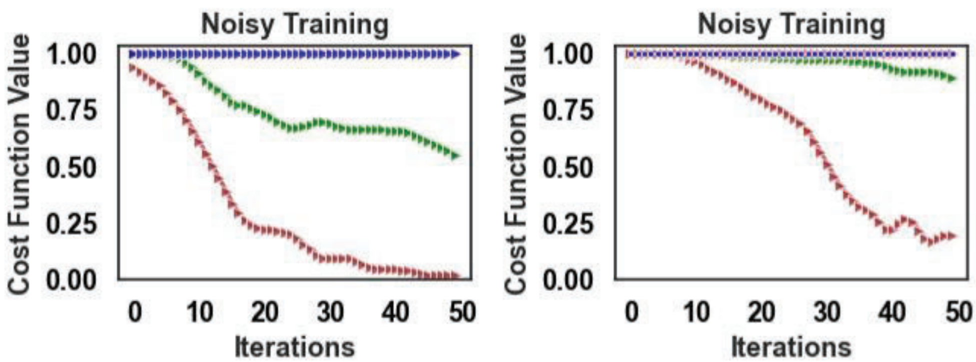
56

HQNET Research



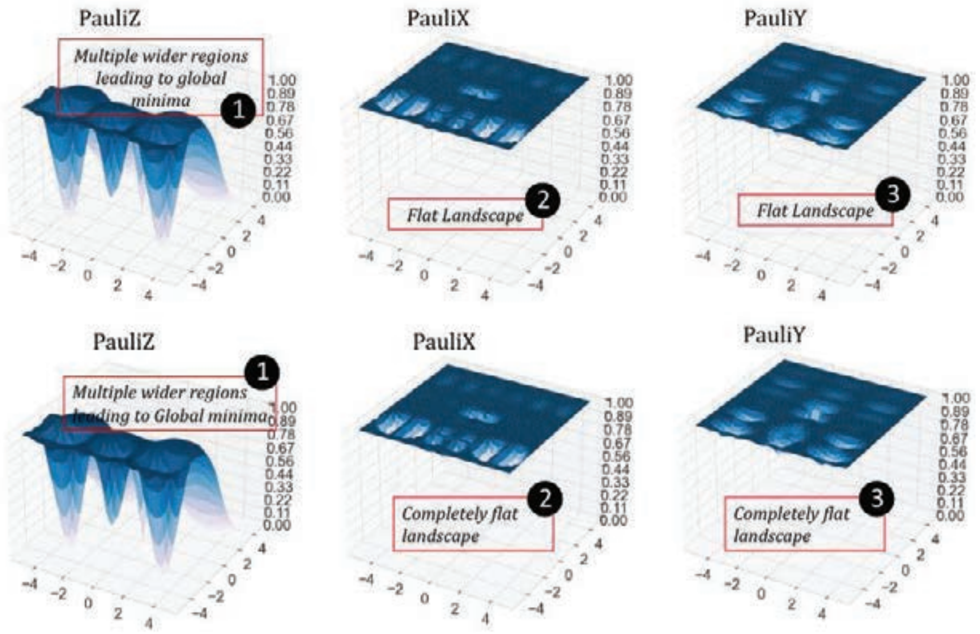
8 Qubits Global QNN Noisy Training

10 Qubits Global QNN Noisy Training



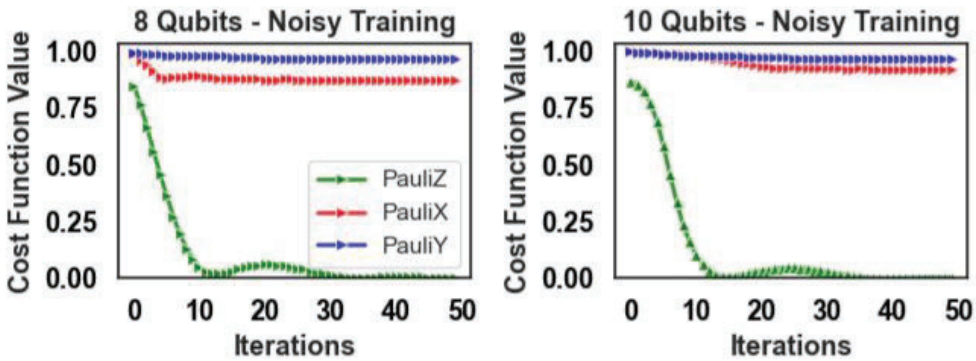
57

HQNET Research



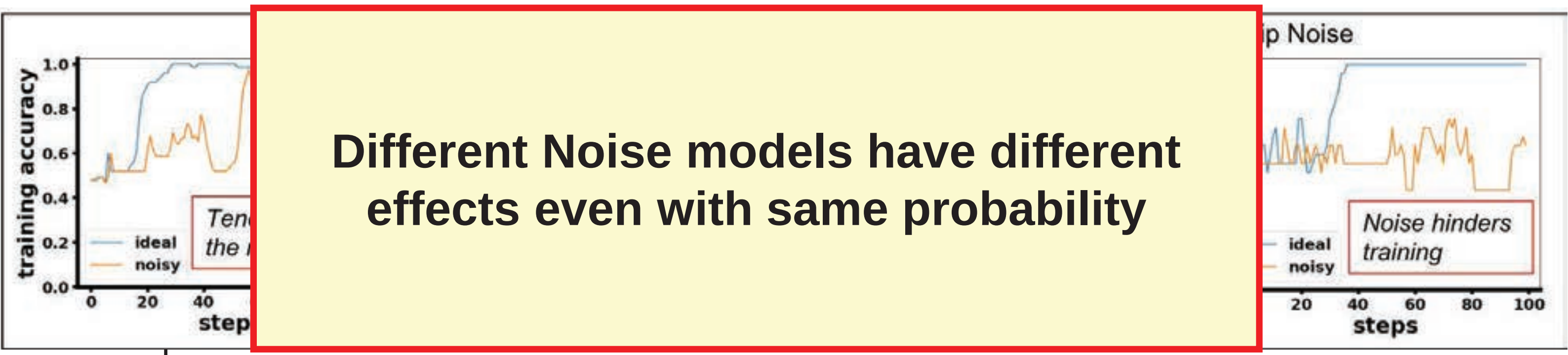
8 Qubits Local QNN Noisy Training

10 Qubits Local QNN Noisy Training



58

Noise Analysis in Hybrid QNNs Research

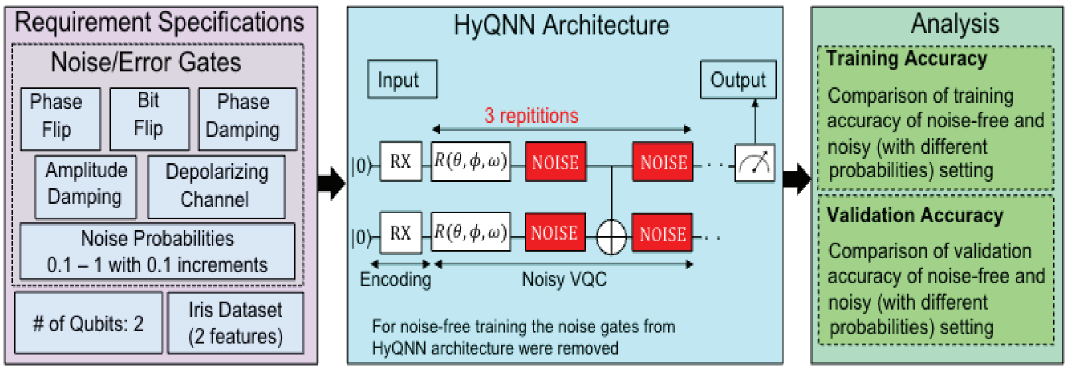


Different Noise models have different effects even with same probability

Solution

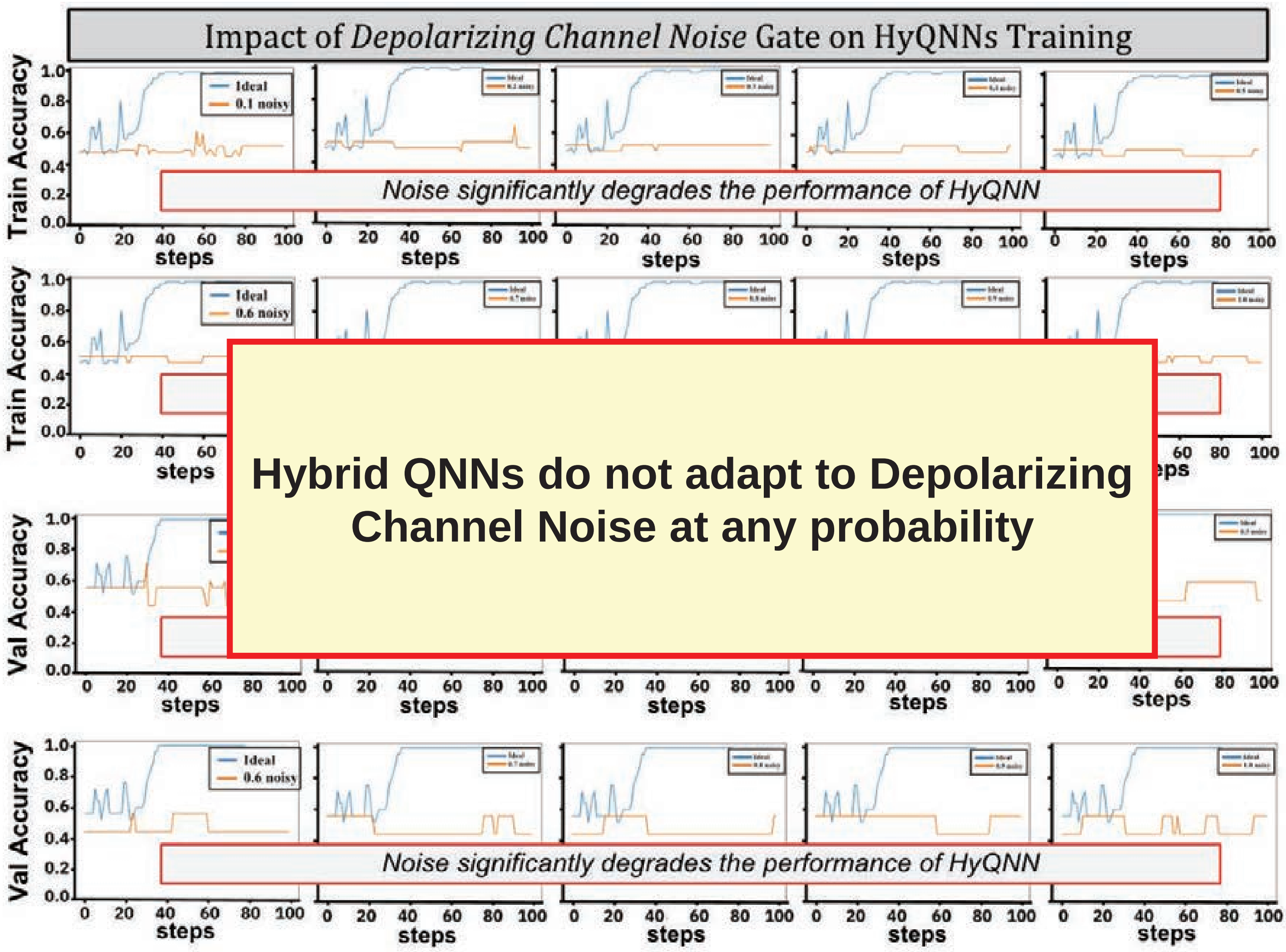
Comprehensive analysis of different noise model on Training of Hybrid QNNs

Methodology



https://arxiv.org/abs/2402.08523 59

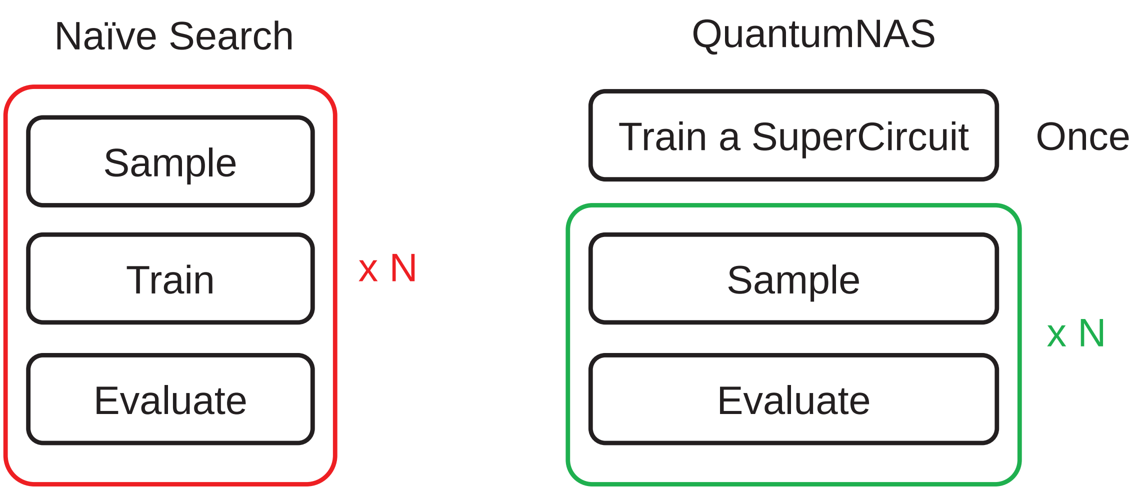
Noise Analysis in Hybrid QNNs Research



60

QuantumNAS: Noise-Adaptive Search

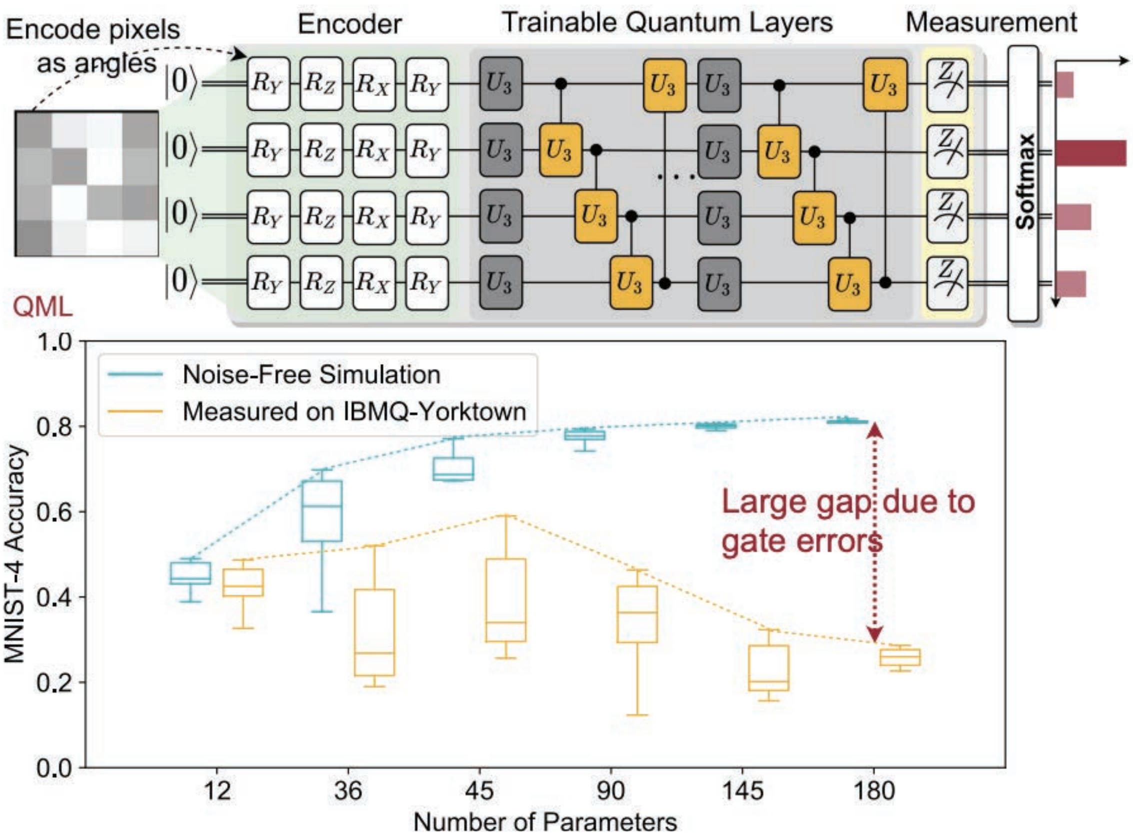
- ❑ Quantum circuits are noisy
  - ❑ More gates: higher capacity, but also higher noise
- ❑ Need to search for noise-robust circuit architecture
  - ❑ Naïve search: train each possible circuit individually
  - ❑ QuantumNAS: train all circuits at once, amortize training cost



Wang et al. "QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits," HPCA 2022.

61

Motivation for Quantum Neural Architecture Search

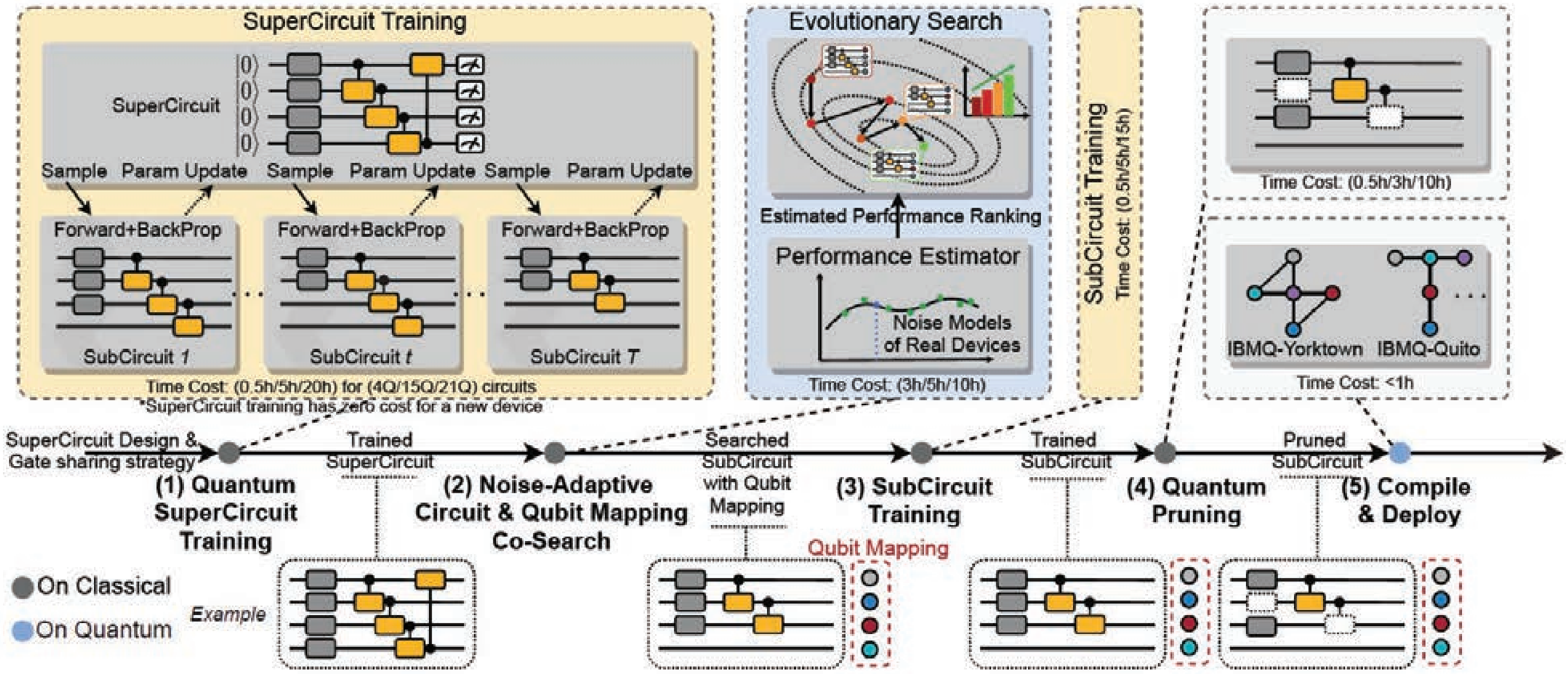


Wang et al. “QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits,” HPCA 2022.

62

QuantumNAS

- ❑ Decoupling the training and search

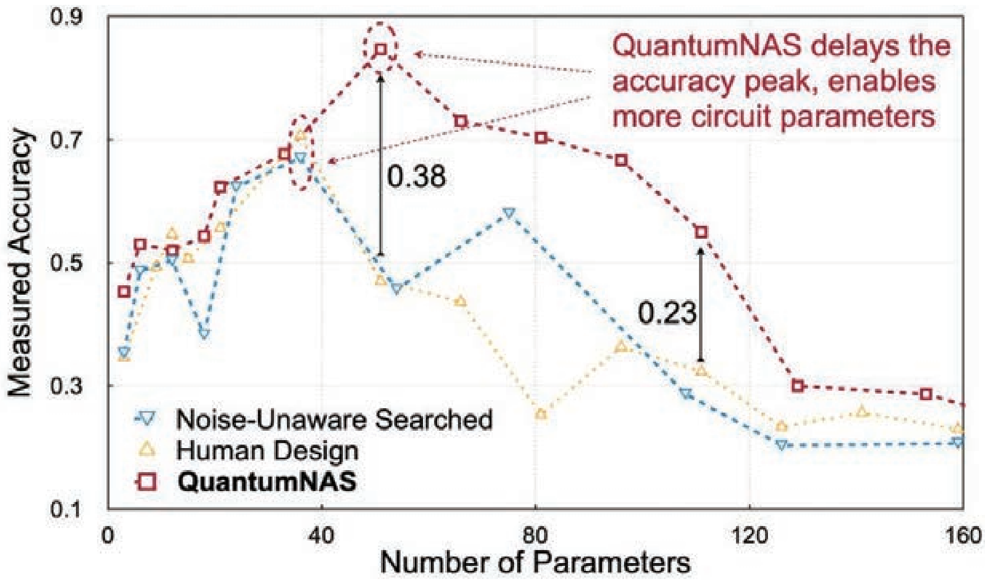


Wang et al. "QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits," HPCA 2022.

63

QML Results

- 4-classification: MNIST-4 U3+CU3 on IBMQ-Yorktown



Wang et al. "QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits," HPCA 2022.

64

Consistent Improvements on Diverse Devices

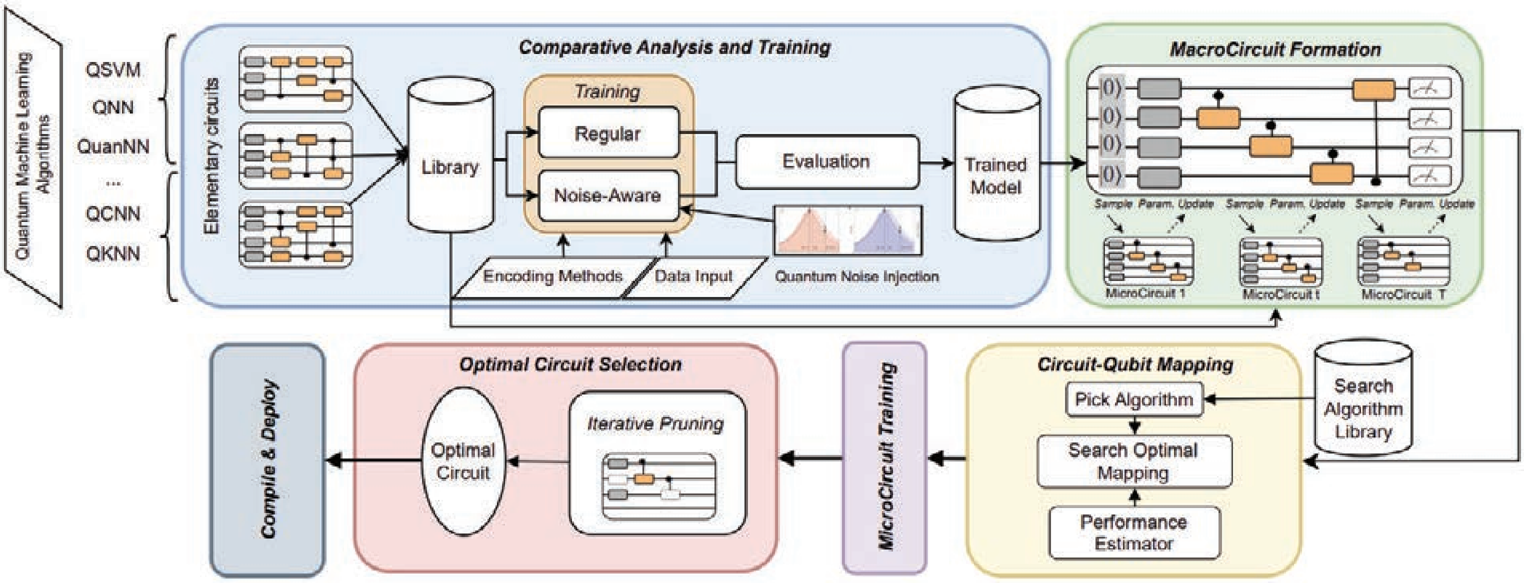
- QuantumNAS is effective for different real quantum devices
- On different 5-Qubit devices
- MNIST-4, Fashion-4, Vowel-4, MNIST-2, Fashion-2 averaged accuracy

Diverse devices

Method	Noise-Unaware Searched	Random	Human	QuantumNAS
Belem (5Q, 16QV)	47%	50%	67%	**76%**
Quito (5Q, 16QV)	73%	68%	74%	**79%**
Athens (5Q, 32QV)	50%	63%	68%	**77%**
Santiago (5Q, 32QV)	74%	73%	75%	**80%**

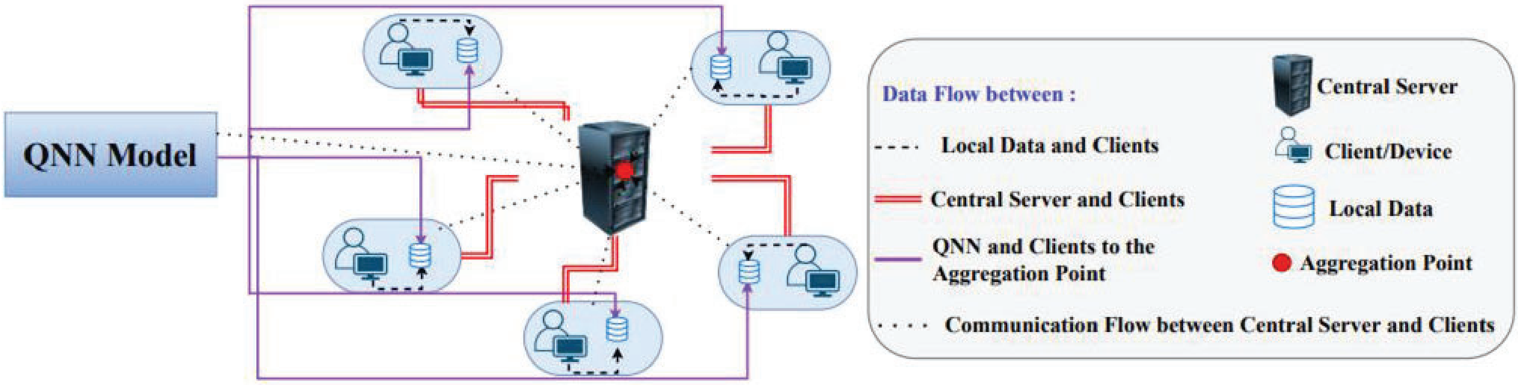
Wang et al. "QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits," HPCA 2022.

Developing Efficient QNNs in the NISQ Era



66

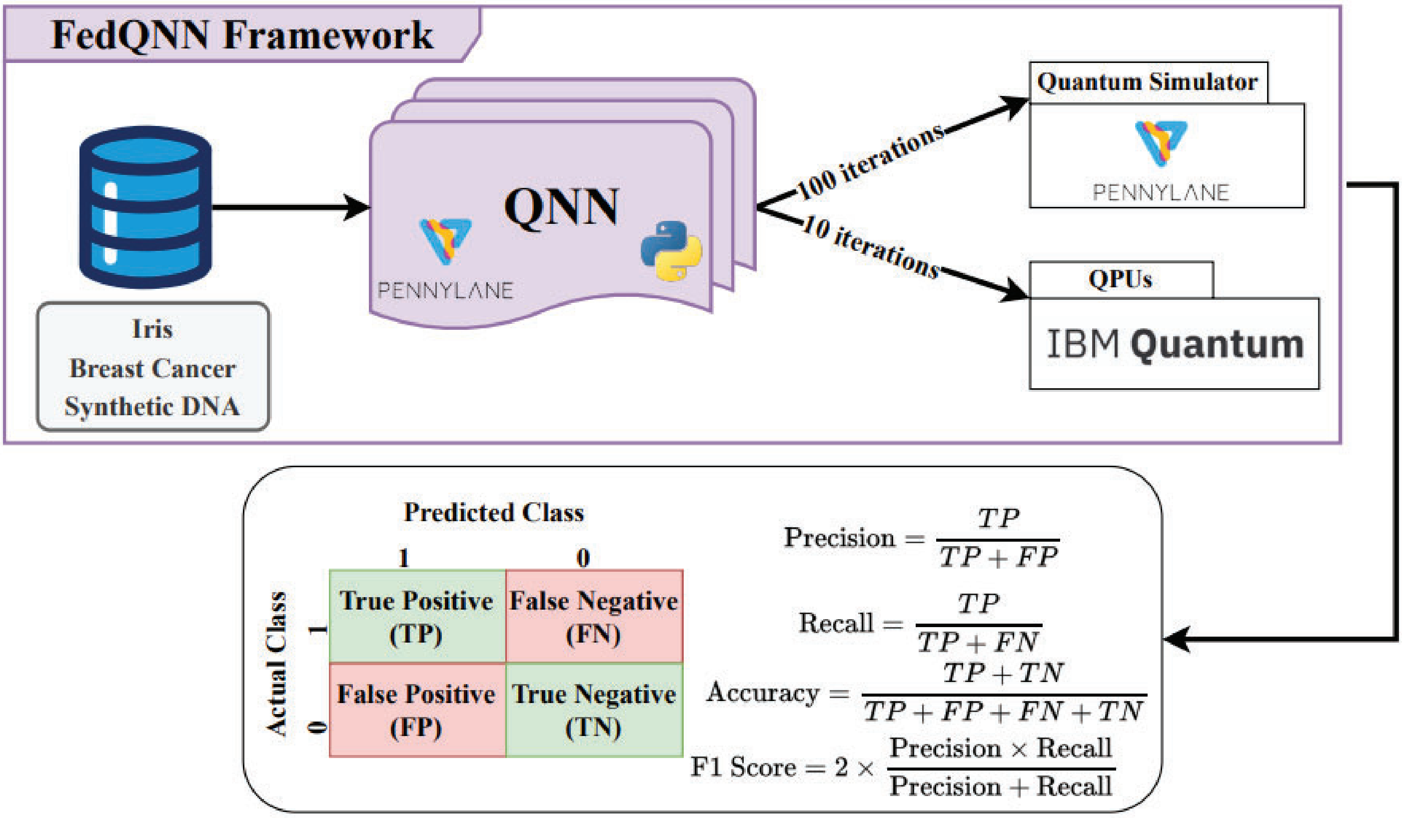
FedQNN Framework



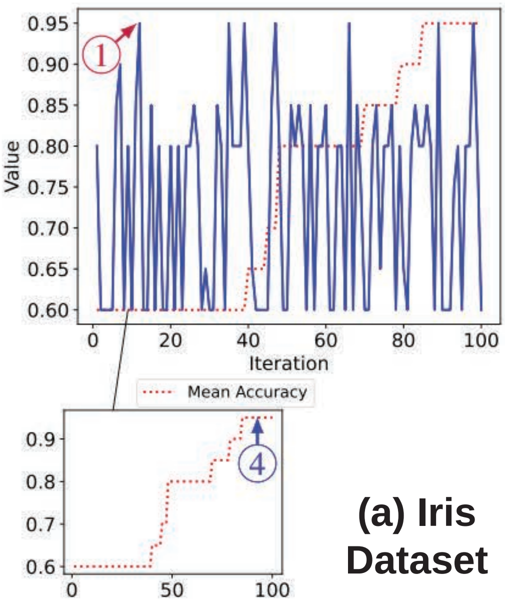
- The FedQNN allows each client to keep local data private, sharing only quantum model updates with a central server for aggregation.
- This framework mathematically combines client updates to form a global model, enhancing performance and maintaining data security.

67

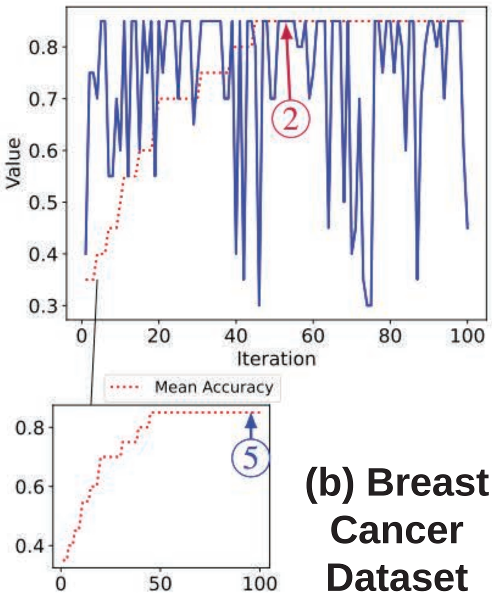
FedQNN Experimental Settings



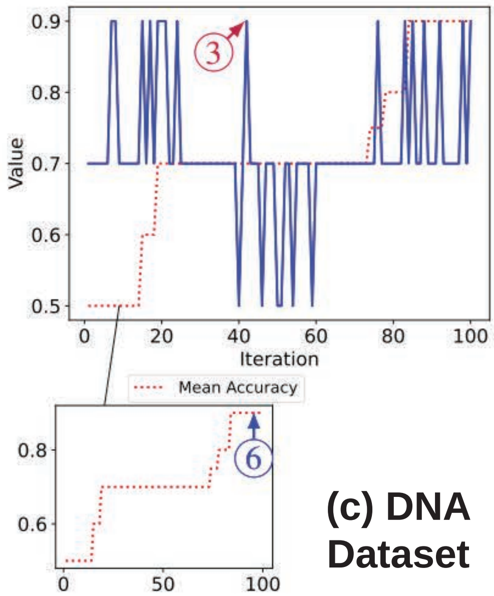
FedQNN Results: Accuracy Trends



(a) Iris Dataset



(b) Breast Cancer Dataset



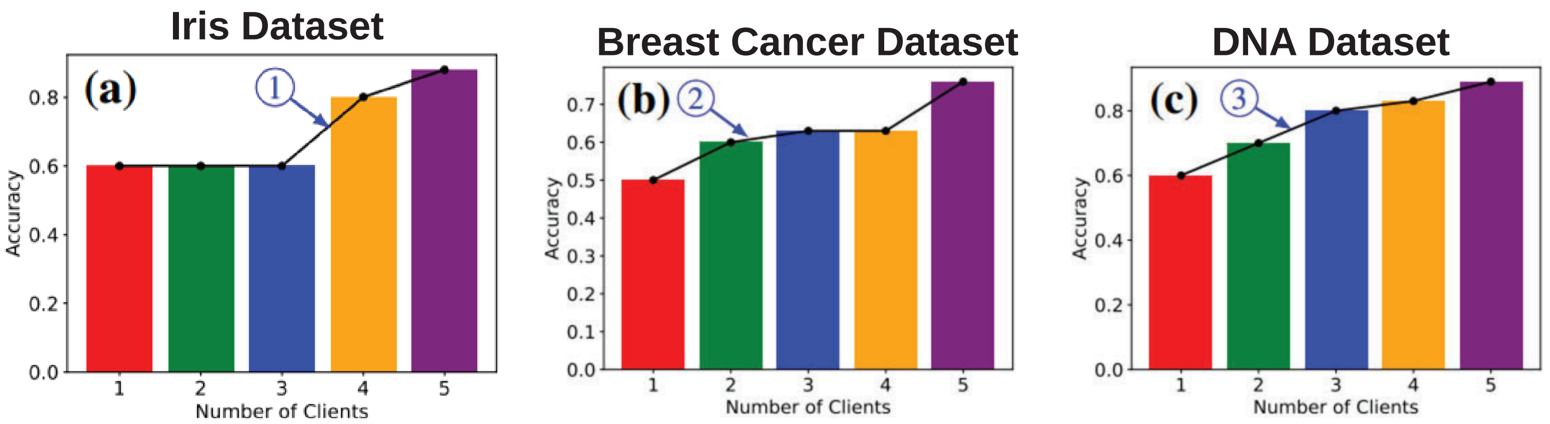
(c) DNA Dataset

Key Observations

- ❑ **(1)**, **(2)**, **(3)**: High accuracy values are periodically reached.
- ❑ Consistent peaking pattern across three datasets.
- ❑ **(4)**, **(5)**, **(6)**: Mean accuracy stabilization at the end of the iterations.

69

FedQNN Results: Client Number Impact on Accuracy

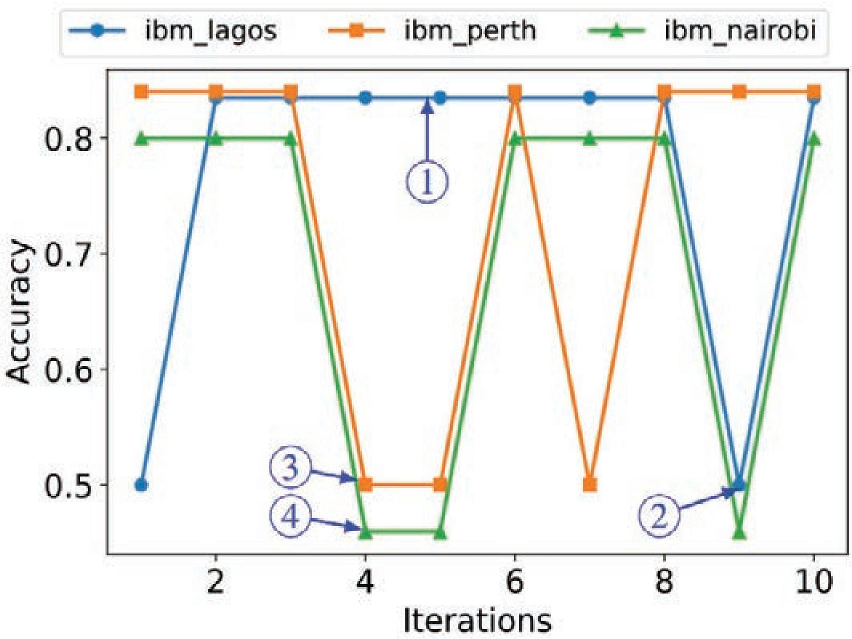


Key Observations

- **(1)** and **(3)**: Clear positive correlation between the number of clients and accuracy, with notable leaps at five clients.
- **(2)**: Consistent improvement as client numbers rise, though with a subtle plateau effect after the third client.
- **(1)**, **(2)**, **(3)**: More clients generally contribute to higher accuracy, demonstrating the advantages of collaborative learning in FedQNN.

70

FedQNN Results: IBM Quantum Processor

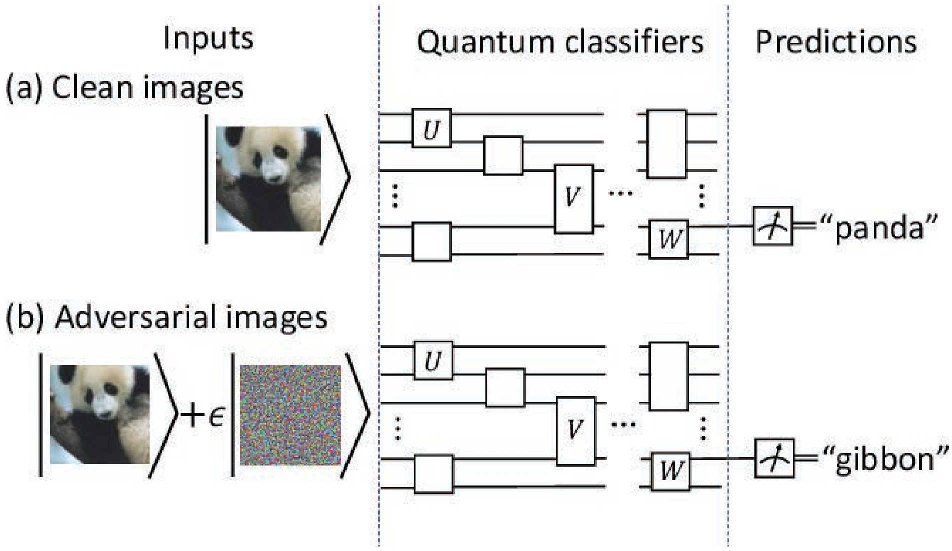


Key Observations

- IBM Lagos: Peaks early **(1)** with a subsequent dip **(2)**.
- IBM Perth: High accuracy, then a drop **(3)**, ending with recovery.
- IBM Nairobi: High to low accuracy **(4)**, ends with a rebound.
- All QPUs surpass the 80% accuracy threshold, despite fluctuations likely due to quantum noise and other operational factors.

71

Adversarial Attacks for Quantum Neural Networks

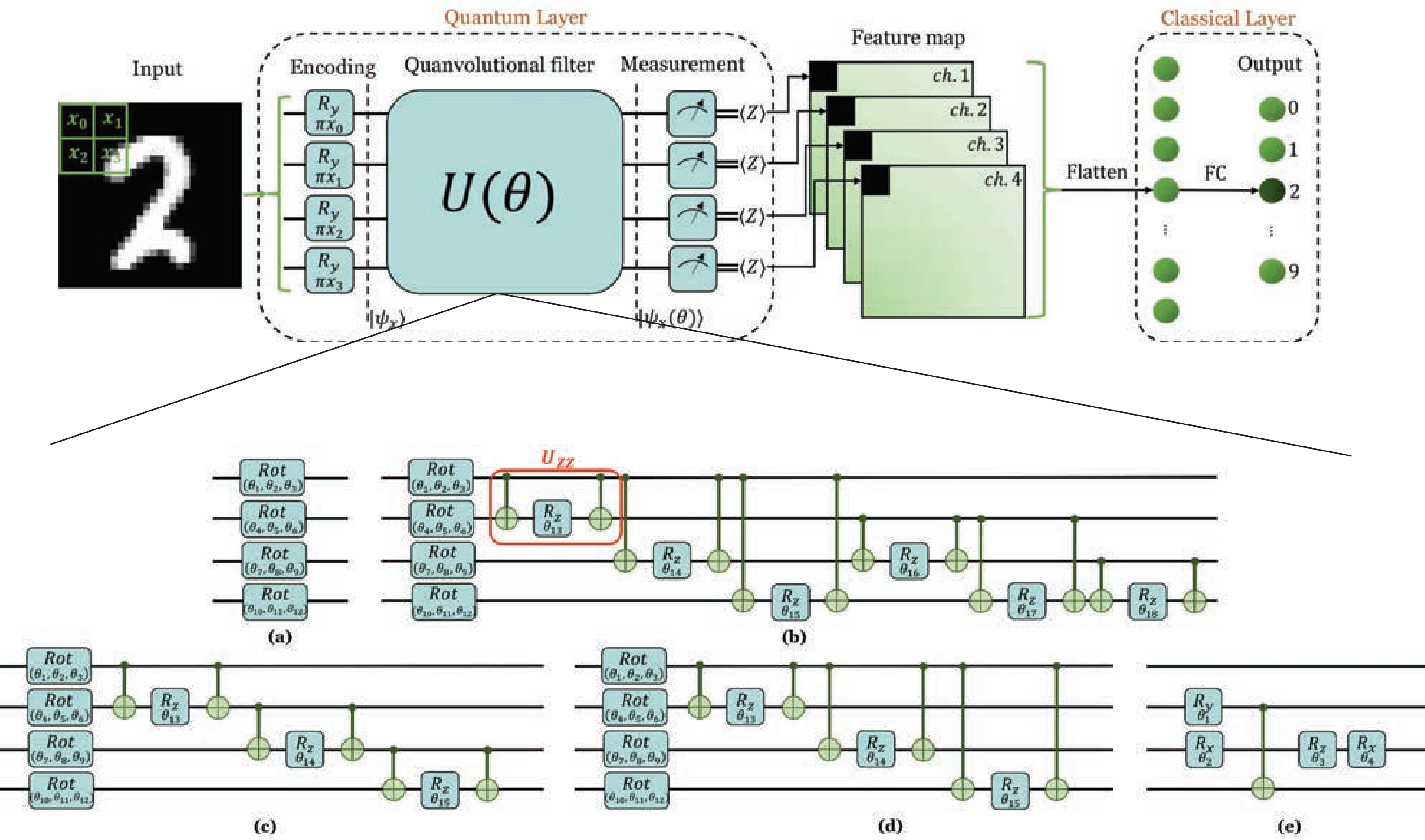


- Like classical DNNs, QNNs can also misinterpret data when subjected to adversarial attacks.
- In our study we reveal that certain QNN models exhibit vulnerabilities similar to their classical counterparts, particularly in classification tasks, **but not to the same extent**.
- QNNs leverage quantum properties like superposition and entanglement, enabling robust pattern recognition and data processing beyond classical ML capabilities.

Lu, Sirui, Lu-Ming Duan, and Dong-Ling Deng. "Quantum adversarial machine learning." *Physical Review Research* 2.3 (2020): 033212.

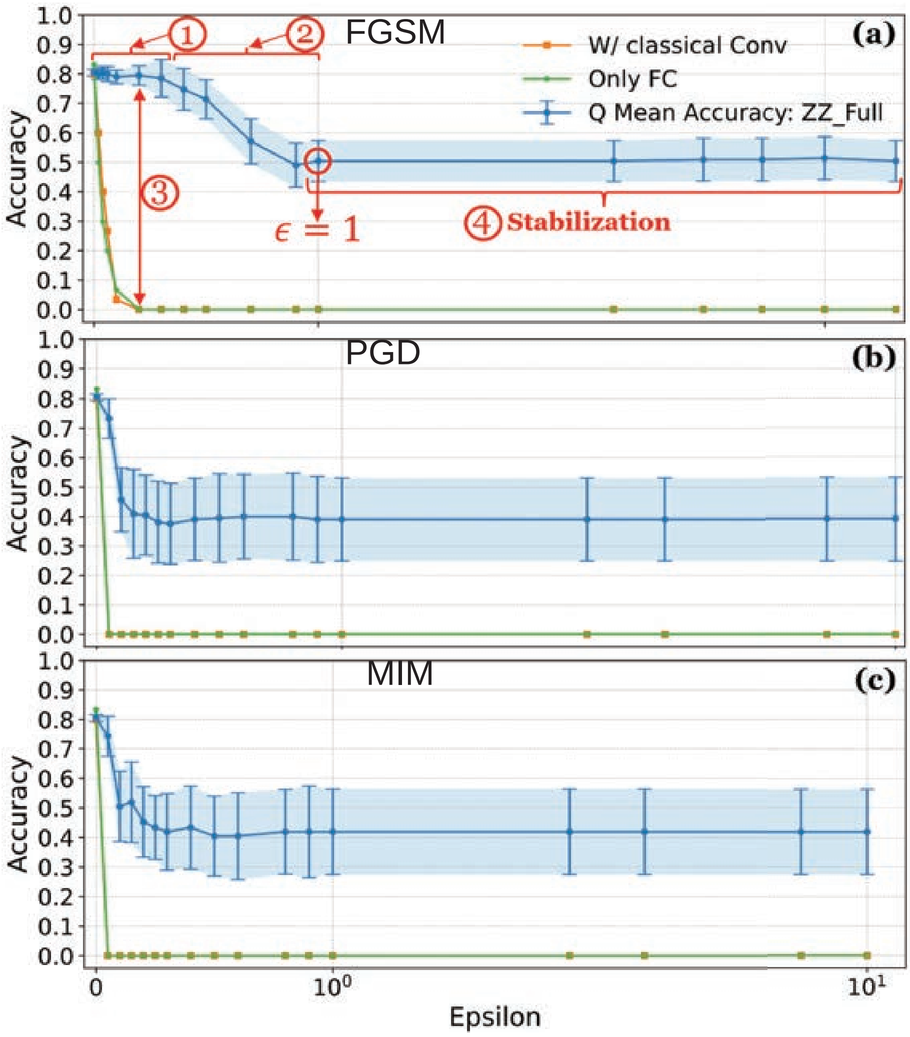
72

AdvQuNN: Ansatz Architectures Variations



73

AdvQuNN Results: QuNN vs Classical CNN

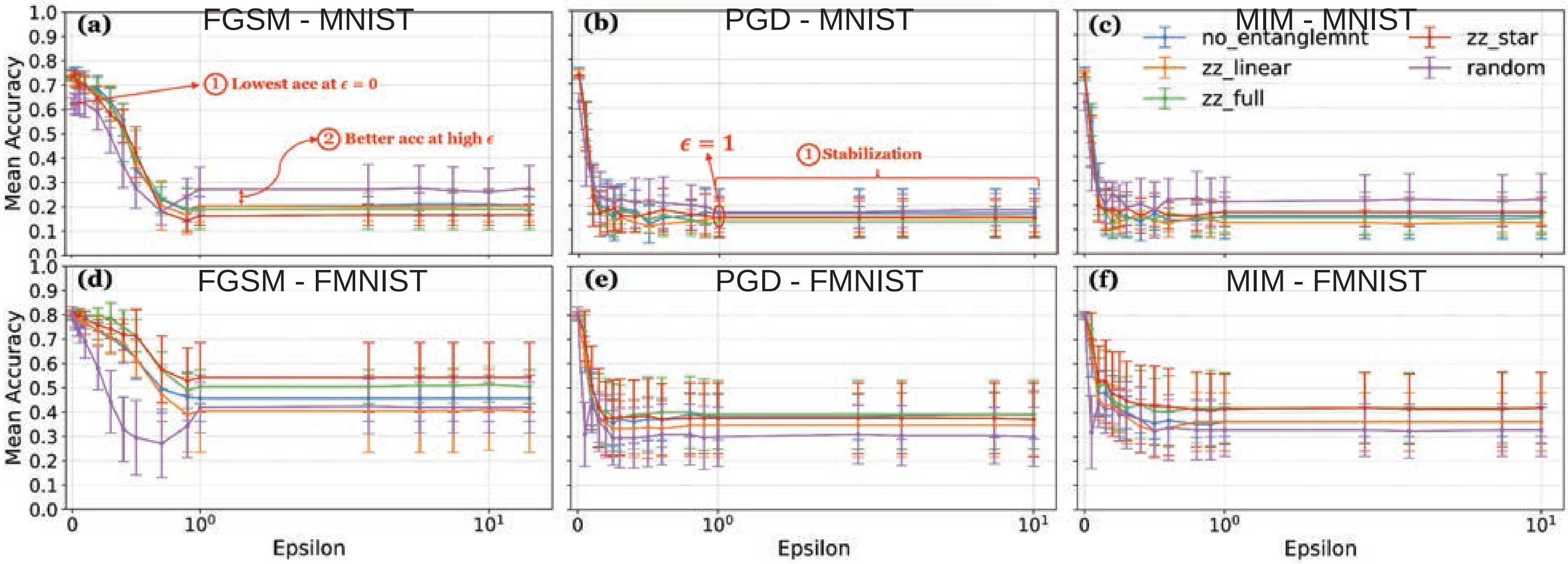


Key Observations

- (1), (2) QuNN requires significant perturbations to begin declining.
- (3) Compared to CNNs and FC, QuNN has up to 60% higher robustness for MNIST (3).
- (4) QuNN accuracy plateaus beyond a perturbation strength of 1 due to image clipping, but it maintains high accuracy despite severe perturbations.

74

AdvQuNN Results: Ansatz Circuit Variations



Key Observations

- ❑ **For the MNIST dataset**: ZZ full and star entanglement architectures are most robust against all three adversarial attacks, showing strong resilience.
- ❑ **For the FMNIST dataset**: The Random architecture outperforms others, despite not achieving the highest accuracy at zero epsilon.
- ❑ The robustness ranking of the Ansatz architecture for FMNIST differs from MNIST, indicating that the Ansatz architecture design is greatly affected by the dataset it is applied to.

75

References: QML Papers @ eBRAIN Lab

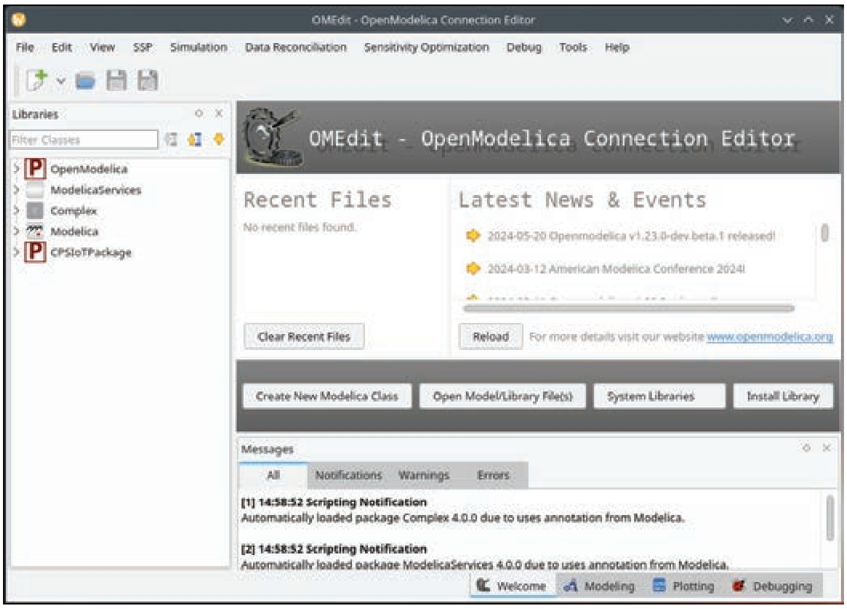
- Nouhaila Innan, Muhammad Al-Zafar Khan, Alberto Marchisio, Muhammad Shafique, Mohamed Bennai: FedQNN: Federated Learning using Quantum Neural Networks. IJCNN, 2024.
- Muhammad Kashif, Emman Sychiuco, Muhammad Shafique: Investigating the Effect of Noise on the Training Performance of Hybrid Quantum Neural Networks, IJCNN, 2024.
- Walid El Maouaki, Alberto Marchisio, Taoufik Said, Mohamed Bennai, Muhammad Shafique: AdvQuNN: A Methodology for Analyzing the Adversarial Robustness of Quanvolutional Neural Networks. QSW, 2024.
- Muhammad Kashif, Muhammad Rashid, Saif Al-Kuwari, Muhammad Shafique: Alleviating Barren Plateaus in Parameterized Quantum Machine Learning Circuits: Investigating Advanced Parameter Initialization Strategies, DATE, 2024.
- Muhammad Kashif, Muhammad Shafique: ResQuNNs: Towards Enabling Deep Learning in Quantum Convolution Neural Networks. CoRR abs/2402.09146, 2024.
- Nouhaila Innan, Alberto Marchisio, Muhammad Shafique, Mohamed Bennai: QFNN-FFD: Quantum Federated Neural Network for Financial Fraud Detection: CoRR abs/2404.02595, 2024.
- Kamila Zaman, Tasnim Ahmed, Muhammad Kashif, Muhammad Abdullah Hanif, Alberto Marchisio, Muhammad Shafique: Studying the Impact of Quantum-Specific Hyperparameters on Hybrid Quantum-Classical Neural Networks, CoRR abs/2402.10605, 2024.
- Kamila Zaman, Tasnim Ahmed, Muhammad Abdullah Hanif, Alberto Marchisio, Muhammad Shafique: A Comparative Analysis of Hybrid-Quantum Classical Neural Networks, CoRR abs/2402.10540, 2024.
- Muhammad Kashif, Muhammad Shafique: HQNET: Harnessing Quantum Noise for Effective Training of Quantum Neural Networks in NISQ Era, CoRR abs/2402.08475, 2024.
- Kamila Zaman, Alberto Marchisio, Muhammad Abdullah Hanif, Muhammad Shafique: A Survey on Quantum Machine Learning: Current Trends, Challenges, Opportunities, and the Road Ahead, CoRR abs/2310.10315, 2023.

76

Thank You!

Installing the tool: OpenModelica Connection Editor (OMEdit)

Download from:
https://openmodelica.org/#



Download the tutorial package from the handout website!!!!

mem4csd.telecom-paristech.fr

go to Training Schools > Summer School 2024 > OpenModelica

Modeling a Cruise Control System using OpenModelica and Verifying Safety Requirements using UPPAAL

Rakshit Mittal[1], Hans Vangheluwe[1], Rizwan Parveen[2]

[1]University of Antwerp – Flanders Make, Belgium

[2]Telecom Paris, France

2 hands-on tutorials with foundations in Multi-Paradigm Modeling

Case Study: Adaptive Cruise Control System (ACCS)

1a: Modeling the ACCS using OpenModelica

Rakshit Mittal[1], Hans Vangheluwe[1]

1b: Verifying ACCS Safety Requirements using UPPAAL

Rizwan Parveen[2]

2a: Modeling and Analyzing the Architecture of the ACCS controller using AADL

Dominique Blouin[2], Anish Bhobe[3]

2b: Synthesizing Code for the ACCS controller using RAMSES

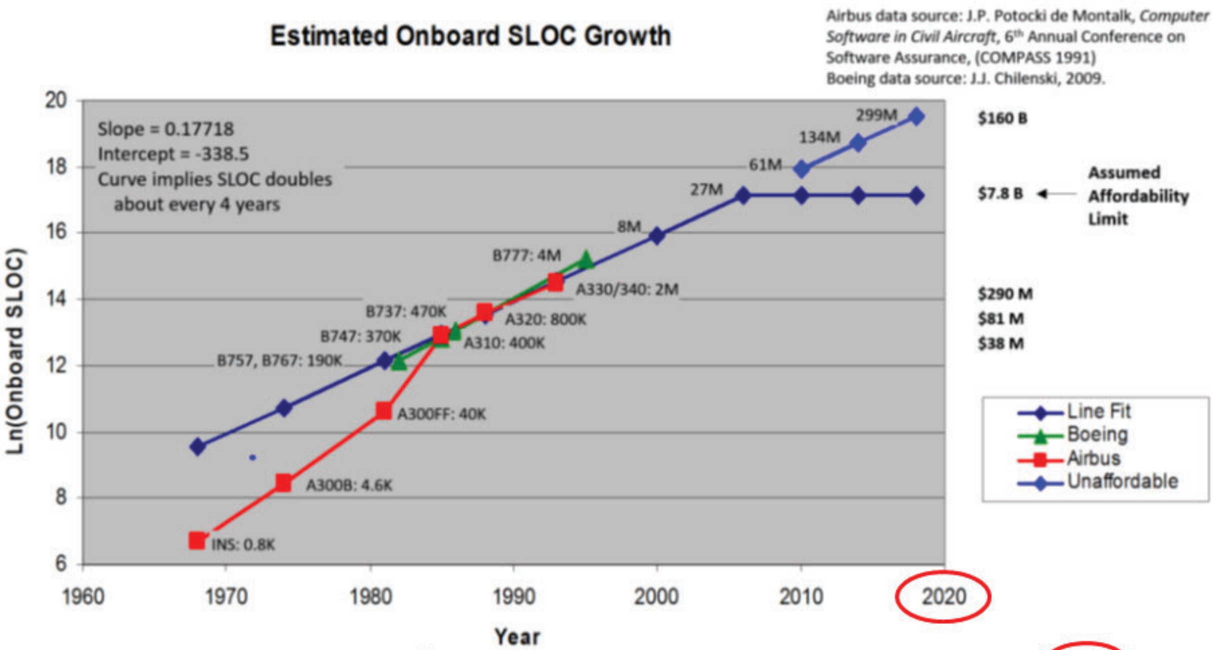
Dominique Blouin[2], Anish Bhobe[3]

[1]University of Antwerp – Flanders Make, Belgium

[2]Telecom Paris, France

[3]Institut Polytechnique de Paris, France

Increasing Systems Complexity



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

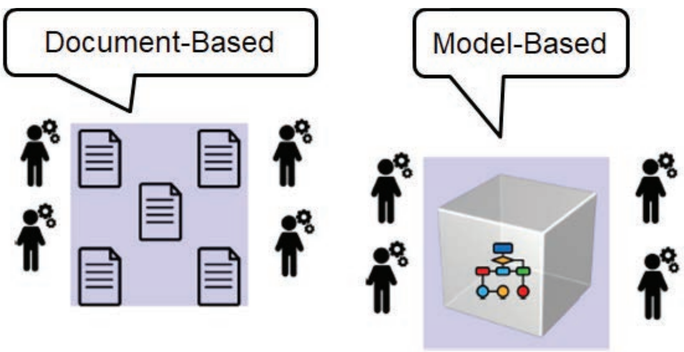
Non-Linear Development Effort Increase

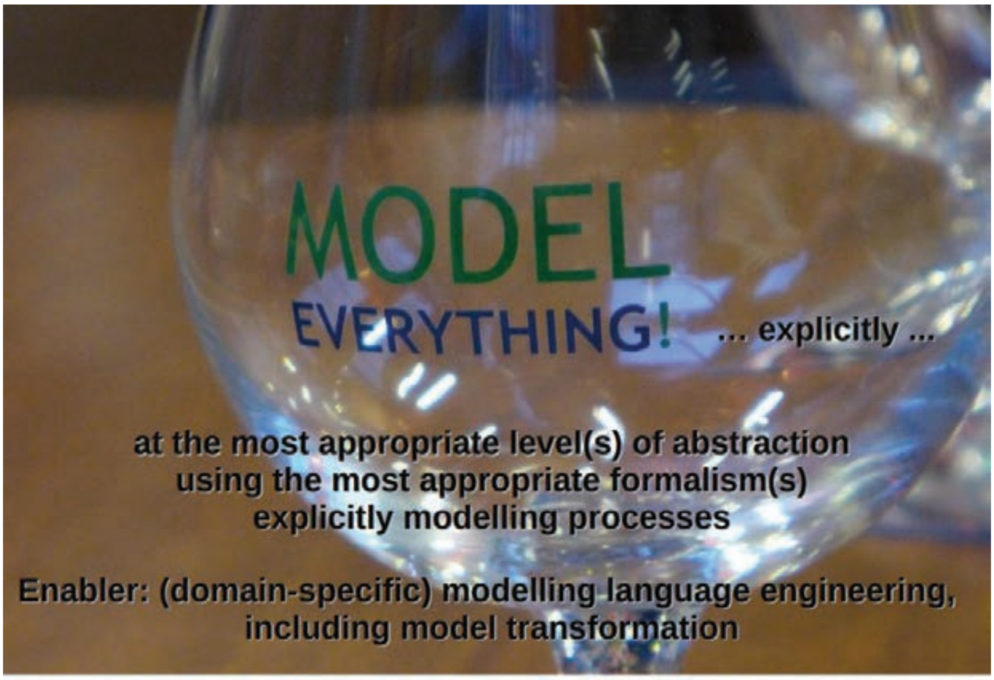
- **F35** SLOC / **F16** SLOC ~ 175
- **F35** Effort / **F16** Effort ~ **300**
  - Source: SAVI Project (https://savi.avsi.aero/)

- A400M:
  - Over 10 years delayed.
  - 6.2 billion euros over budget (30% overrun).
  - Source: https://www.rt.com/business/airbus-a400m-france-delays-561/

Paradigm Shift: Model-Based Systems Engineering (MBSE)

- From natural language documents to **models**.
- Provide common **vocabulary**.
- Enforce more **precision**.
- Allow building **tools** to process specifications (models).
- Allow detecting errors / inconsistencies **early** with these tools.
- Quite **effective** for avionics development (> 25 % costs reduction).





Pieter J. Mosterman and Hans Vangheluwe. Computer Automated Multi-Paradigm Modeling: An Introduction. Simulation: Transactions of the Society for Modeling and Simulation International , 80(9):433- 450, September 2004. Special Issue: Grand Challenges for Modeling and Simulation.

Multi-Paradigm Modeling for Cyber-Physical Systems

MPM4CPS

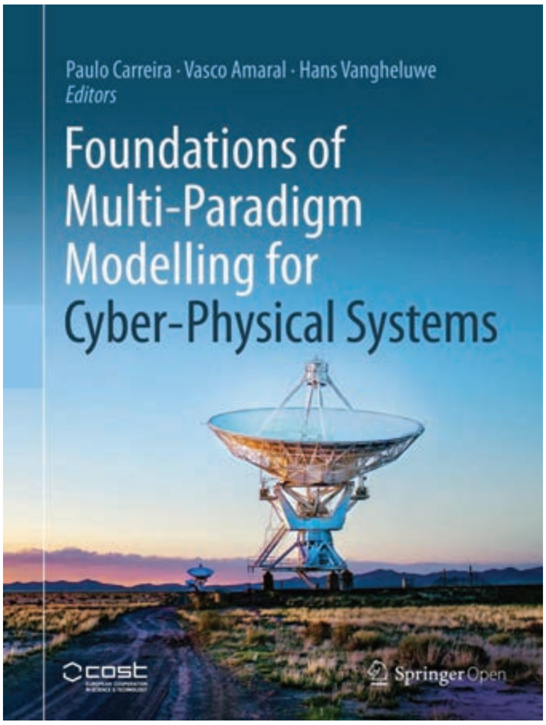
cost
EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

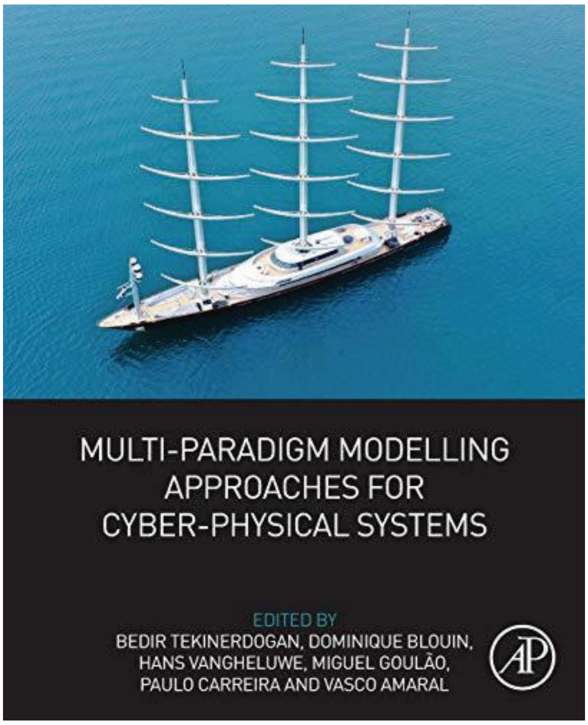
mpm4cps.eu

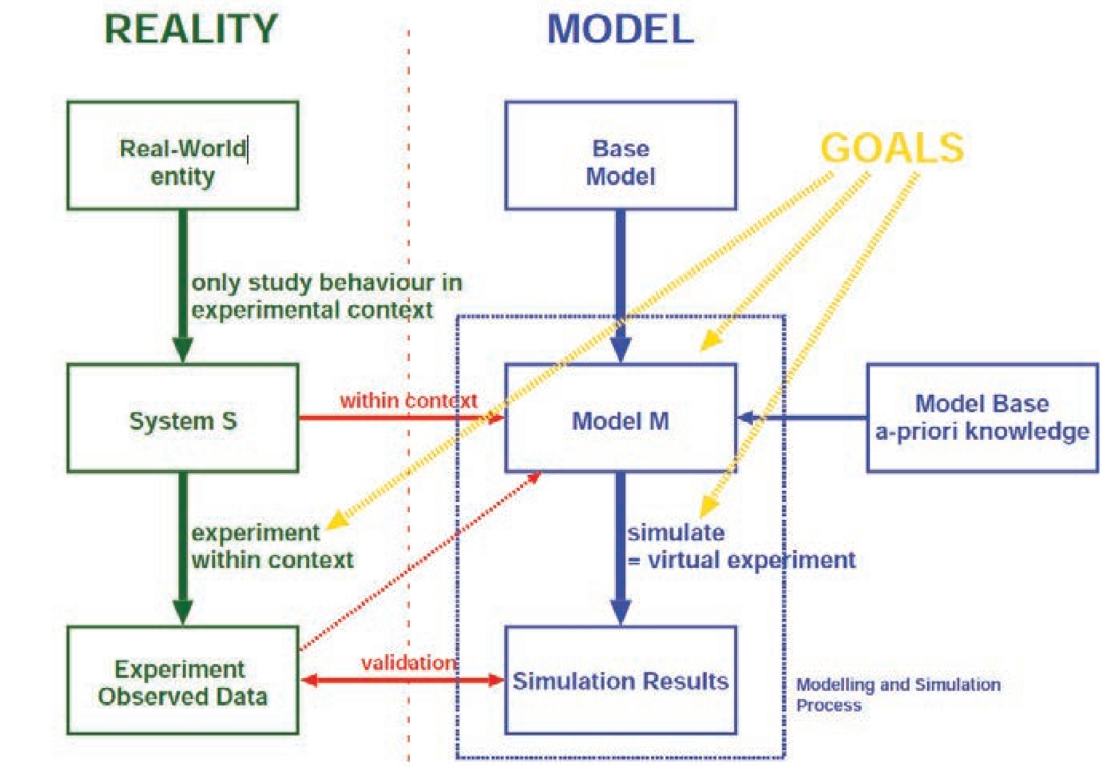
Hans is the pope

and Dominique is the bishop!

TELECOM Paris
IP PARIS


Paulo Carreira · Vasco Amaral · Hans Vangheluwe
Editors
Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems
cost
Springer Open

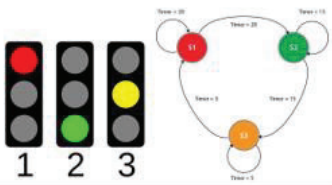
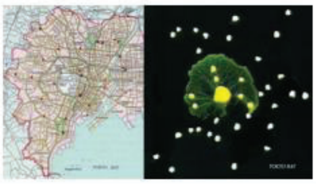
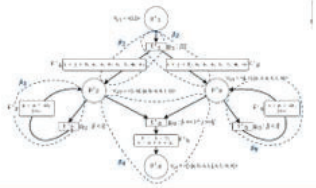

MULTI-PARADIGM MODELLING APPROACHES FOR CYBER-PHYSICAL SYSTEMS
EDITED BY
BEDIR TEKINERDOGAN, DOMINIQUE BLOUIN, HANS VANGHELUWE, MIGUEL GOULÃO, PAULO CARREIRA AND VASCO AMARAL



Bernard P. Zeigler. *Multi-faceted Modelling and Discrete-Event Simulation*. Academic Press, 1984.

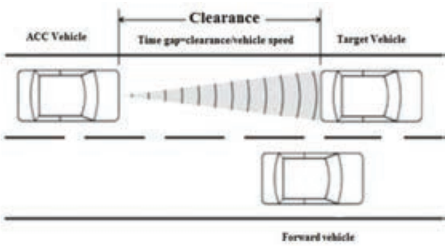
disclaimer

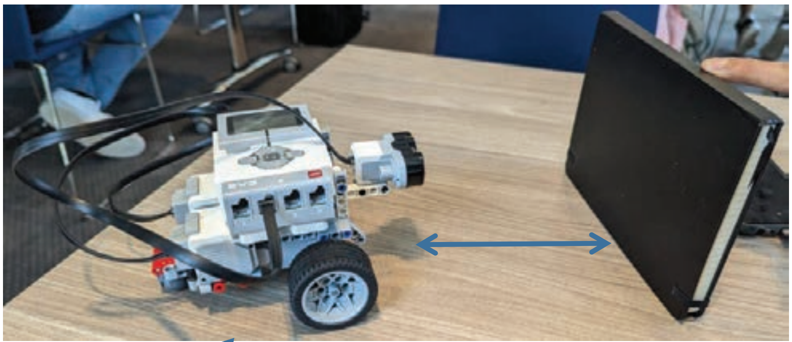
- The model need not always be 'conceptual', and the modelled system need not always be 'real'

	Real Model	Conceptual Model
Real System		
Conceptual System		

Case-Study

Adaptive Cruise Control System





The actual robot that you are going to use.

2 hands-on tutorials with foundations in Multi-Paradigm Modeling

Case Study: Adaptive Cruise Control System (ACCS)

1a: Modeling the ACCS using OpenModelica

Rakshit Mittal[1], Hans Vangheluwe[1]

1b: Verifying ACCS Safety Requirements using UPPAAL

Rizwan Parveen[2]

2a: Modeling and Analyzing the Architecture of the ACCS controller using AADL

Dominique Blouin[2], Anish Bhobe[3]

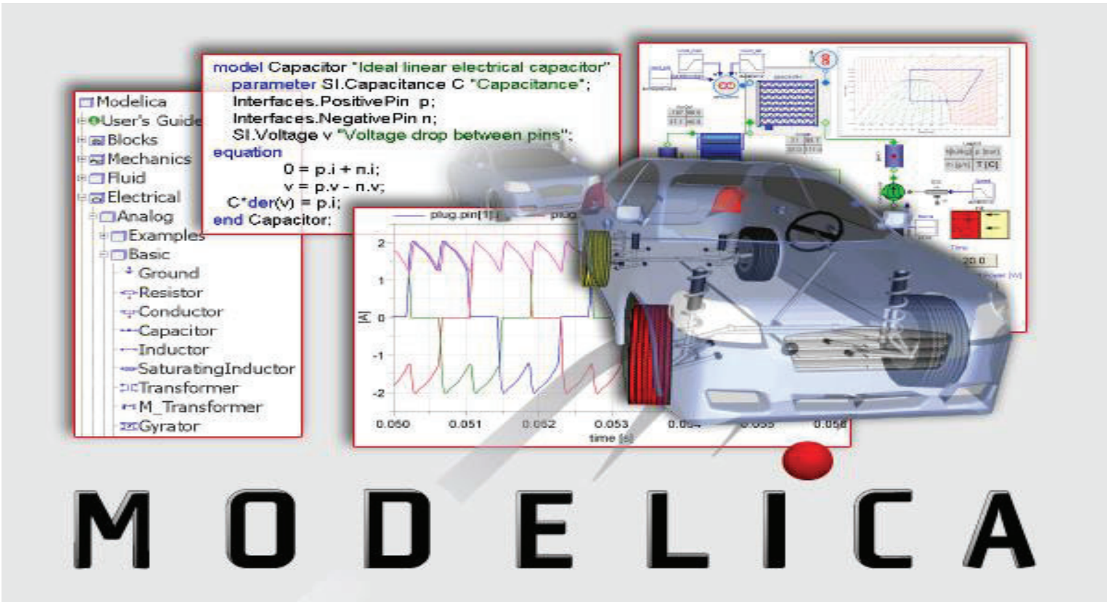
2b: Synthesizing Code for the ACCS controller using RAMSES

Dominique Blouin[2], Anish Bhobe[3]

[1]University of Antwerp – Flanders Make, Belgium

[2]Telecom Paris, France

[3]Institut Polytechnique de Paris, France


Modelica
User's Guide
Blocks
Mechanics
Fluid
Electrical
Analog
Examples
Basic
Ground
Resistor
Conductor
Capacitor
Inductor
SaturatingInductor
Transformer
M_Transformer
Gyrator
model Capacitor "Ideal linear electrical capacitor"
parameter SI.Capacitance C "Capacitance";
Interfaces.PositivePin p;
Interfaces.NegativePin n;
SI.Voltage v "Voltage drop between pins";
equation
0 = p.i + n.i;
v = p.v - n.v;
C*der(v) = p.i;
end Capacitor;
plug.pin[1].i
2
1
0
-1
-2
[A]
0.050
0.051
0.052
0.053
time [s]
MODELICA

Lund Institute of Technology

REPORT LUTFD2/(TFRT-1015)/1-226/(1978)

Karl Johan Åström

May 1978

Hilding Elmqvist

A Structured Model Language for Large Continuous Systems

A model language, called DYMOLA, for continuous dynamical systems is proposed. Large models are conveniently described hierarchically using a submodel concept. The ordinary differential equations and algebraic equations need not be converted to assignment statements. There is a concept, cut, which corresponds to connection mechanisms of complex types, and there are facilities to describe the connection structure of a system. A model can be manipulated for different purposes such as simulation and static calculations. The model equations are sorted and they are converted to assignment statements using formula manipulation. A translator for the model language is also included.

Author

nonlinear systems, compiler, permutations, graph theory

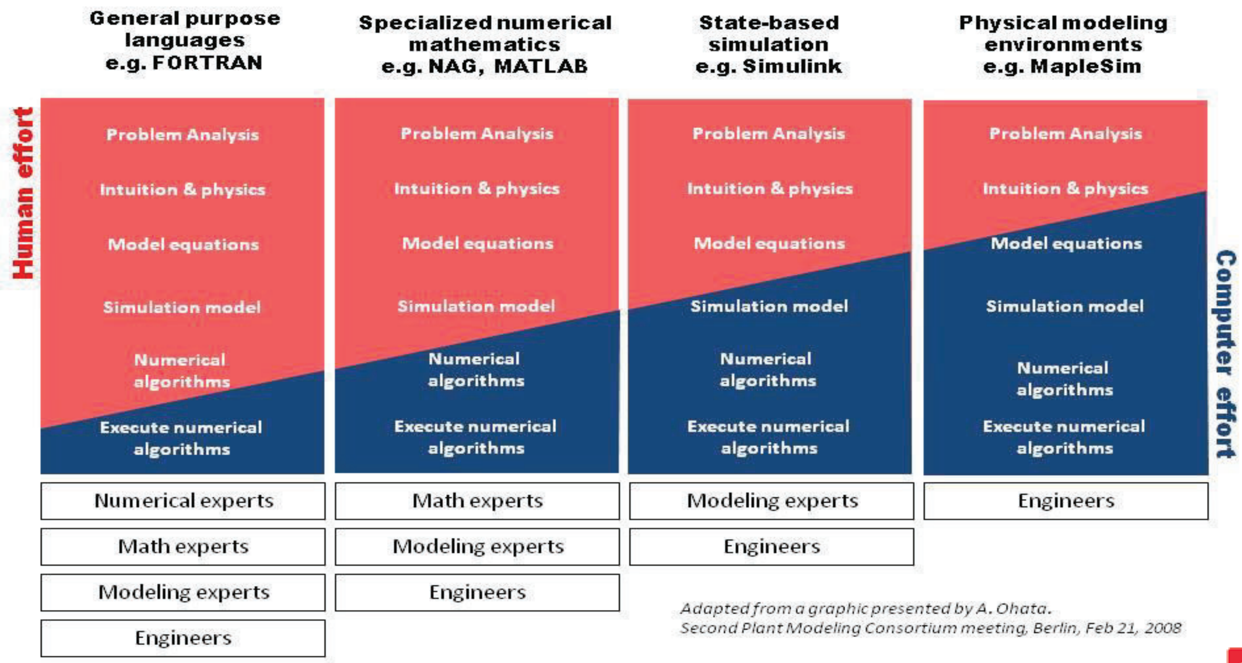
Mathematical models, Simulation languages, Computerized simulation, Nonlinear systems, Ordinary differential equations, Compilers.
(Thesaurus of Engineering and Scientific Terms, Eng. Joint Council, USA)

226 pages

English

Department of Automatic Control
Lund Institute of Technology
P O Box 725, S-220 07 Lund 7, Sweden

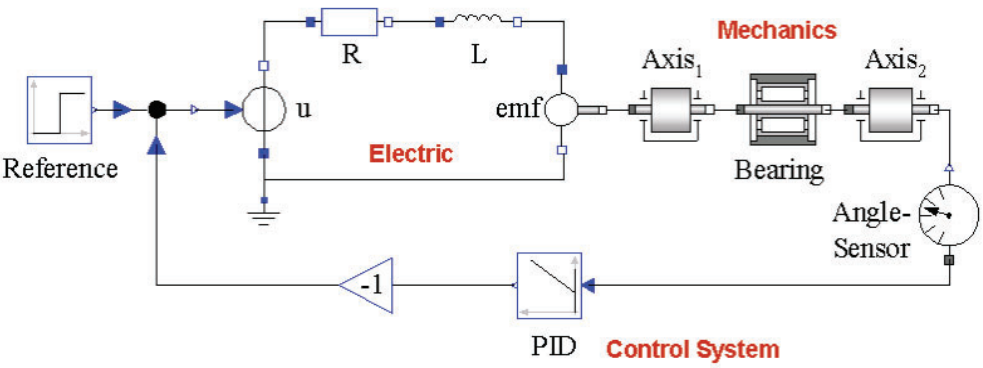
Blankett LU 11:25 1976-07



Adapted from a graphic presented by A. Ohata.
Second Plant Modeling Consortium meeting, Berlin, Feb 21, 2008

Multi-Domain Modeling

http://www.modelica.org



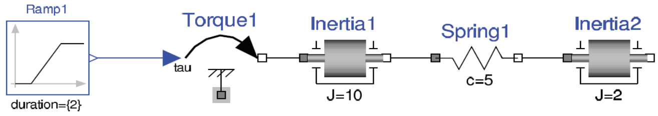
this slide from Peter Fritzson's Modelica tutorial

Multi-Domain Modeling

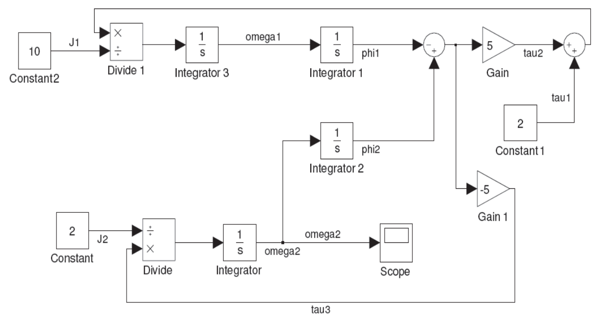
Visual Acausal Hierarchical Component Modeling

Keeps the physical structure

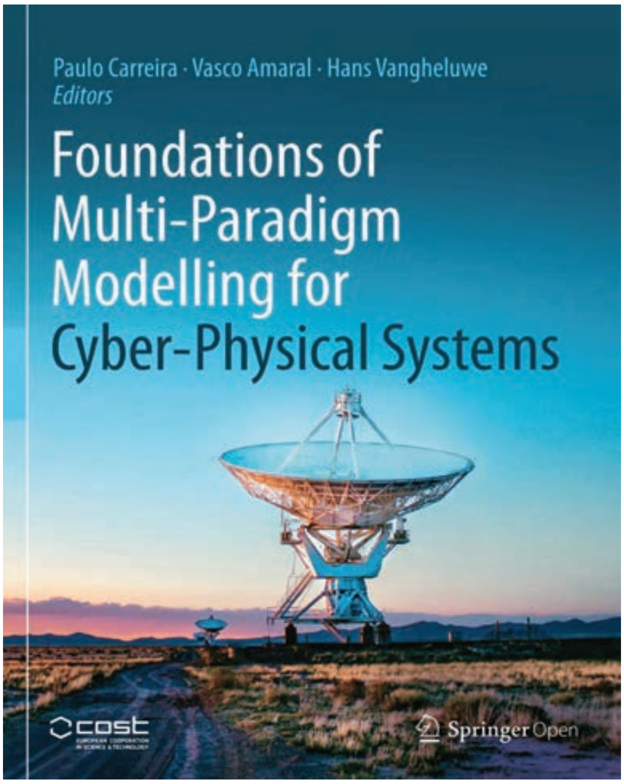
Acausal model (Modelica)



Causal block-based model (Simulink)



this slide from Peter Fritzson's Modelica tutorial



Modelica

https://modelica.org/documents/ModelicaTutorial14.pdf

OpenModelica

https://openmodelica.org/

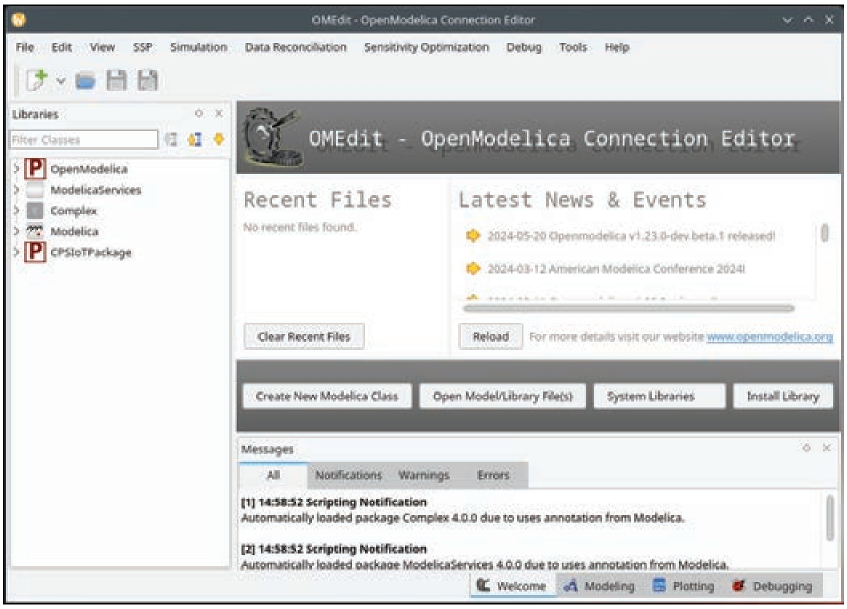
Modelica by Example

by Dr. Michael M. Tiller

https://mbe.modelica.university/

Fritzson P. (2020) Modelica: Equation-Based, Object-Oriented Modelling of Physical Systems.
In: Carreira P., Amaral V., Vangheluwe H. (eds) Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems. Springer, Cham.
https://doi.org/10.1007/978-3-030-43946-0_3

The tool:
OpenModelica Connection Editor (OMEdit)



Download the tool from:
https://openmodelica.org/#

The resources:
download from
https://nextcloud.rakshitmittal.net/s/iY4qRkgkW9yx8WB
or request a pen-drive!

Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

5 mins

The motor should not move too fast!
So the input to the motor controller is limited to [-300, 300].
Simulate the function using the test-bed. Modify the parameters and observe simulation output.

- CPSIoT24ModelicaTutorial
  - Part1_Procedural
    - LimitFunction
    - LimitModel

```
function LimitFunction
    input Real u "input";
    input Integer K_high "high limit";
    input Integer K_low "low limit";
    output Integer result;
  algorithm
    result := if u > K_high then K_high elseif u < K_low then K_low else integer(u);
  end LimitFunction;
```

```
model LimitModel
  parameter Integer k_high  "high limit";
  parameter Integer k_low = -k_high "low limit";
  Real u "input";
  Real y "output";
  equation
    y = LimitFunction(u, k_high, k_low);
end LimitModel;
```

Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

10 mins

The position of the lead car can be described by differential equations.
Three different kinds are already provided.
Simulate them, and then also create your own custommodel!

- CPSIoT24ModelicaTutorial
  - Part1_Procedural
  - Part2_Equation
    - LeadCarContextLinear
    - LeadCarContextExp
    - LeadCarContextHarmonic

```
model LeadCarContextLinear
  Real x(start = 10);
  equation
    der(x) = 5;
end LeadCarContextLinear;
```

```
model LeadCarContextExp
  Real x(start = 10);
  equation
    der(x) = x;
end LeadCarContextExp;
```

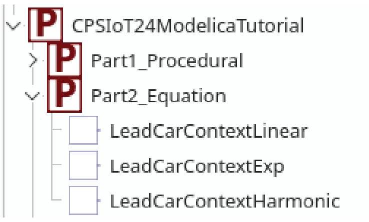
```
model LeadCarContextHarmonic
  Real x(start = 10);
  Real v(start = 0);
  equation
    der(x) = v;
    der(v) = -x;
// x(t) = A*sin(t) + B*cos(t)
// v(t) = A*cost(t) - B*sin(t)
end LeadCarContextHarmonic;
```

Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

10 mins

Simulate the harmonic equation with different settings:



<u>Simulation 1</u>
solver : dassl
stop-time: 20 s
step-size : 0.02 s

<u>Simulation 2</u>
solver : euler
stop-time: 20 s
step-size : 0.5 s

```
model LeadCarContextHarmonic
  Real x(start = 10);
  Real v(start = 0);
  equation
    der(x) = v;
    der(v) = -x;
// x(t) = A*sin(t) + B*cos(t)
// v(t) = A*cost(t) - B*sin(t)
end LeadCarContextHarmonic;
```

Which simulation is correct?

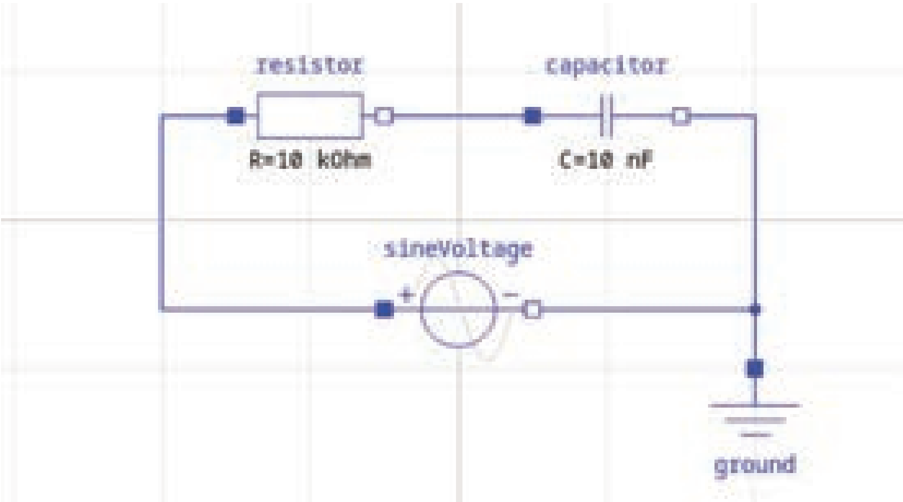
Notice the numerical in/stability.
Stability => The parametric plot should be bounded.

So, it not just about having the correct model, but also using the correct solver settings!

Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

Object-Orientation: concepts like classes/types, instances, encapsulation, specialization



An exemplar low-pass RC circuit

Electrical Types

```
type Time = Real (final quantity="Time", final unit="s");
type ElectricPotential = Real (final quantity="ElectricPotential",
                               final unit="V");
type Voltage = ElectricPotential;
type ElectricCurrent = Real (final quantity="ElectricCurrent",
                             final unit="A");
type Current = ElectricCurrent;
```

Electrical Pin Interface

```
connector PositivePin "Positive pin of an electric component"
       Voltage v "Potential at the pin";
  flow Current i "Current flowing into the pin";
end PositivePin;
```

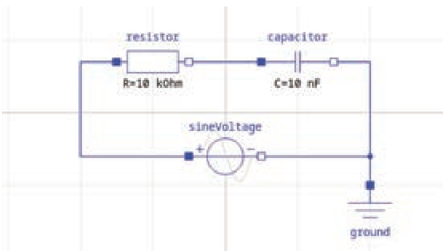
Electrical Port

```
partial model OnePort
  "Component with two electrical pins p and n
   and current i from p to n"
  Voltage v "Voltage drop between the two pins (= p.v - n.v)";
  Current i "Current flowing from pin p to pin n";
  PositivePin p;
  NegativePin n;
equation
  v = p.v - n.v;
  0 = p.i + n.i;
  i = p.i;
end OnePort;
```

Electrical Resistor

```
model Resistor "Ideal linear electrical resistor"
  extends OnePort;
  parameter Resistance R=1 "Resistance";
  equation
    R*i = v;
end Resistor;
```

What is the meaning behind the connections between these re-usable blocks?
How is this meaning extracted?

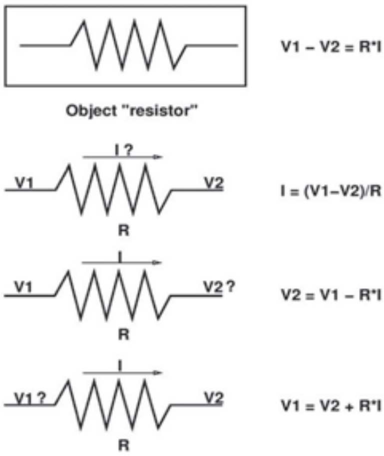



```
model IntroRCLPF
Modelica.Electrical.Analog.Basic.Resistor resistor(R(displayUnit = "kOhm") = 1e4)
Modelica.Electrical.Analog.Basic.Ground ground annotation( ...);
Modelica.Electrical.Analog.Basic.Capacitor capacitor(C(displayUnit = "nF") = 1e-8)
Modelica.Electrical.Analog.Sources.SineVoltage sineVoltage(V = 2, f = 100000)  ann
equation
connect(sineVoltage.n, ground.p) annotation( ...);
connect(ground.p, capacitor.n) annotation( ...);
connect(capacitor.p, resistor.n) annotation( ...);
connect(resistor.p, sineVoltage.p) annotation( ...);
end IntroRCLPF;
```

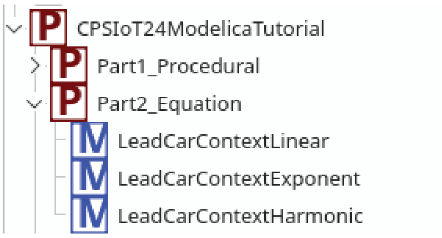
The meaning is always: a set of Differential Algebraic Equations (DAEs) !!

They are obtained by:
1.a. expanding inheritance
1.b. instantiation
2. flattening hierarchy, construct unique names
3. expanding connect() into equations (across vs. flow)

Object-oriented re-use and causality

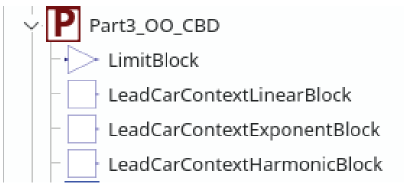


15 mins



Recall that we created at least 4 different models.

Can we now extend those models so that they can be re-used like blocks in the Modelica graphical syntax?



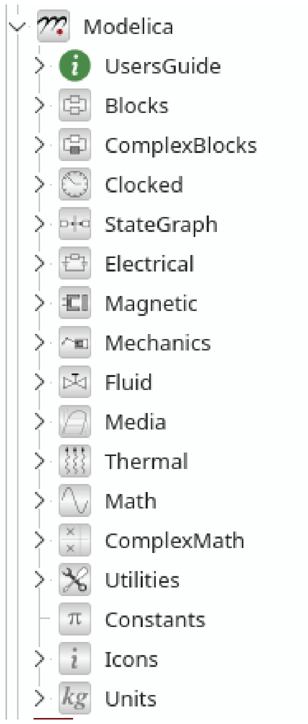
As an example, you will find (in part 3) the corresponding blocks for the four models from the previous parts of the tutorial.

You should look at the textual syntax of the models, and then use similar techniques to make the block for your custom model, that you created in part 2.

Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

MSL - Modelica Standard Library

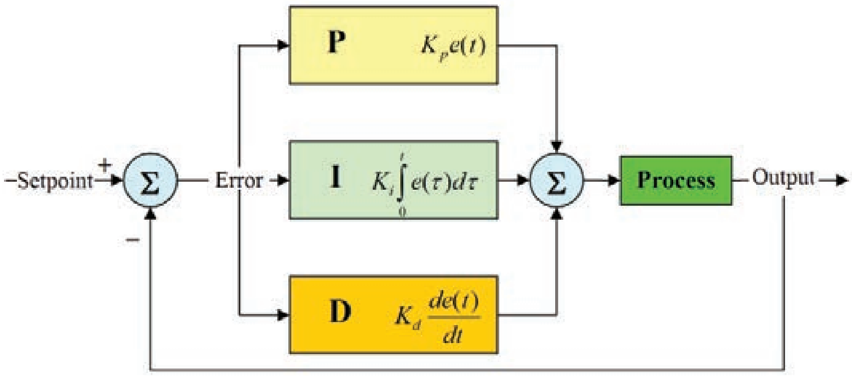


Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

PID Controller

Closed-loop system: better stability



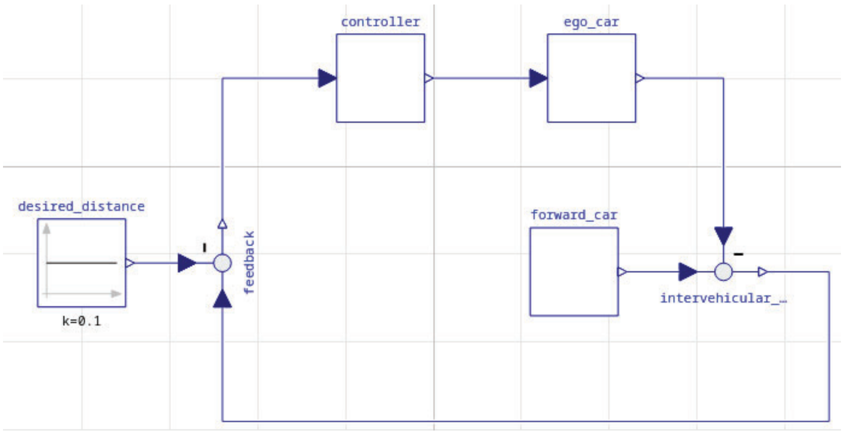
P control by itself is unable to get rid of the steady-state error, which results in a permanent offset.

The steady-state error is eliminated by the integral component, which gradually accumulates the error and modifies the controller's output. However, it may result in instability and oscillations from excessive integral activity.

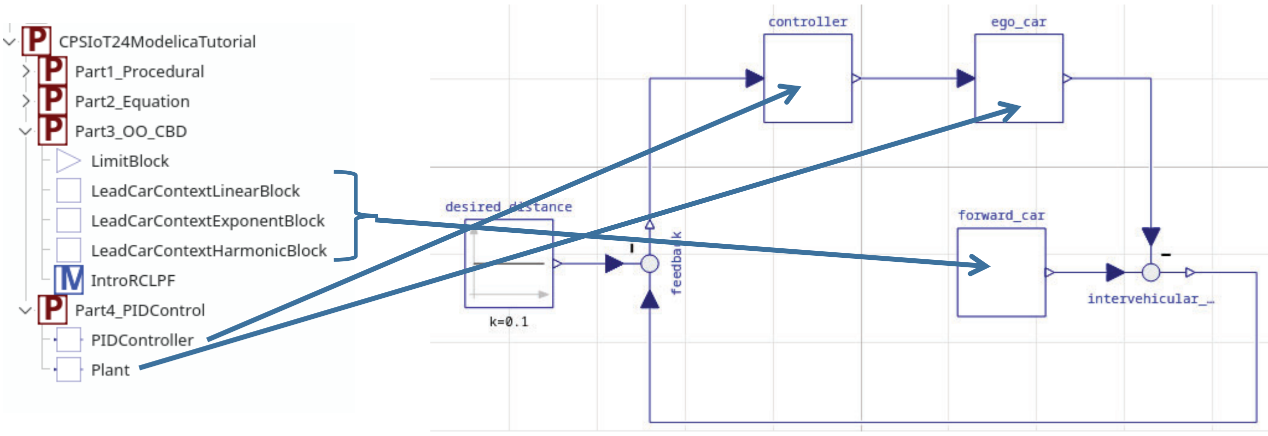
The derivative component forecasts the inaccuracy in the future. By increasing the derivative gain (Kd) by the error's derivative over time, it produces a damping effect. By doing this, the response is smoothed down and oscillations and overshoot are lessened.

https://www.wattco.com/2024/05/pid-controller-explained/

Given what you have learnt today, and considering that all blocks are provided.
Can you now make the following PID control loop model of the robot to simulate its behavior?



rem.
time



What are the best values for Kp, Ki, Kd ??

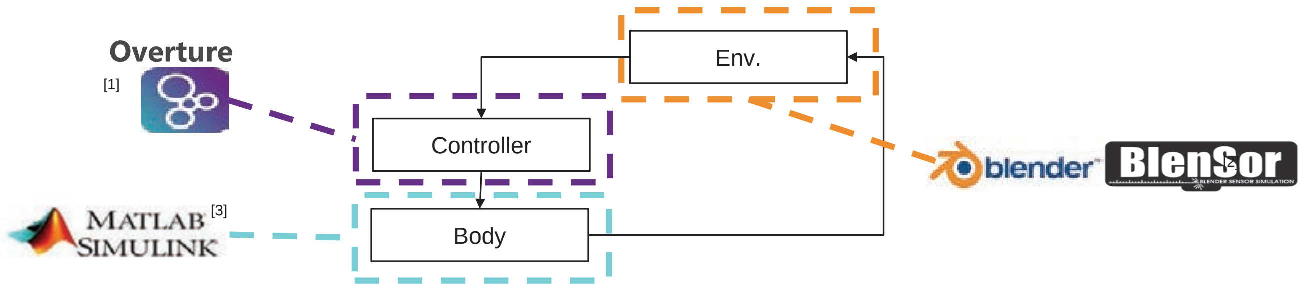
Remember these values, you will use them in the 2nd tutorial !

Equation-Based Object-Oriented Modelling of the Physics, with Modelica

- Programming: procedural code (function/algorithm)
- Equation-based (a-causal) modelling
- Behind the scenes: numerical approximations
- Object-Oriented modelling
- Libraries and the MSL
- Controller Modelling
- Extra time: Hiding IP: Composition of Functional Mockup Units (FMI)

problem: **full-system analysis**

(also when IP protected)

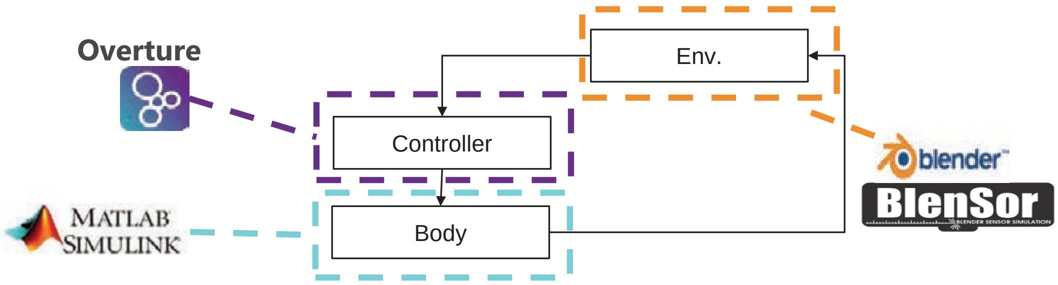


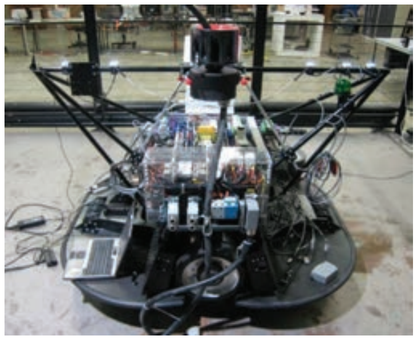
solution: combine sub-system **simulators**

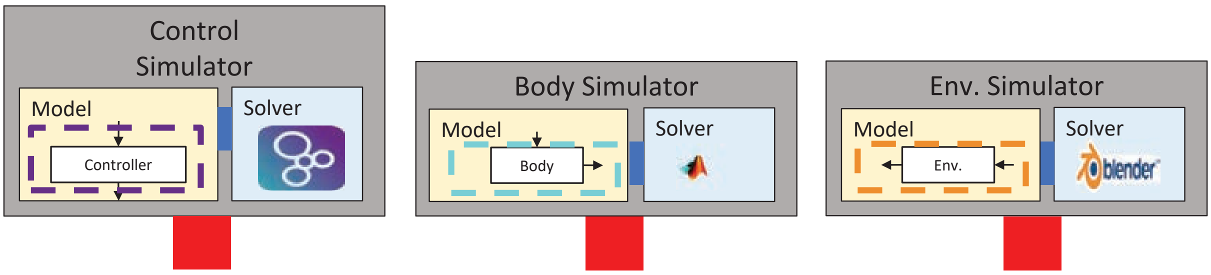
aka **co-simulation**

Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, Hans Vangheluwe
Co-Simulation: A Survey. ACM Comput. Surv.51(3): 49:1-49:33 (2018)

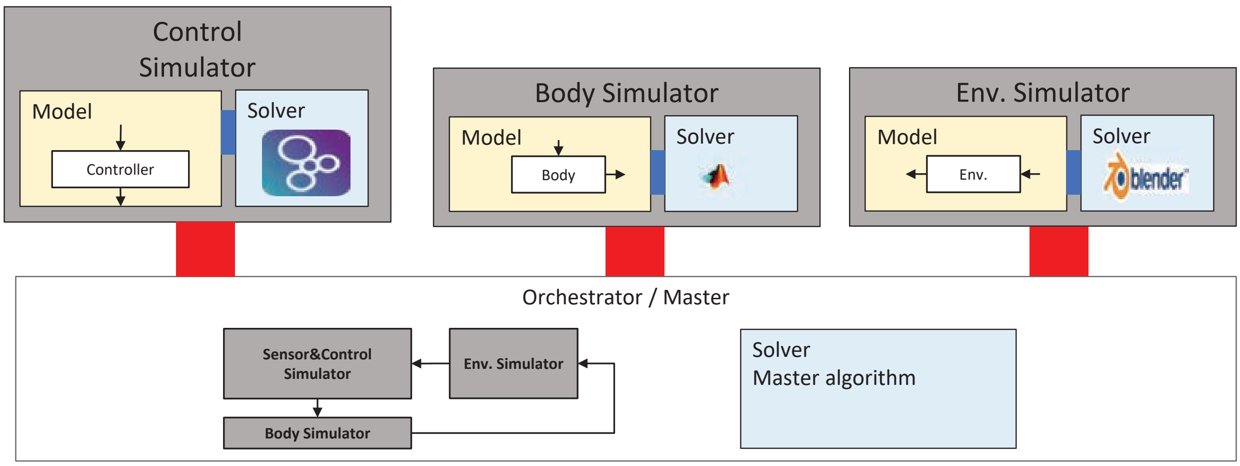
co-simulation: how? (when IP protected)

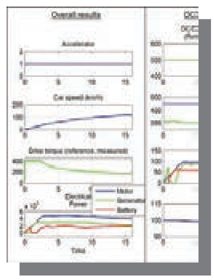






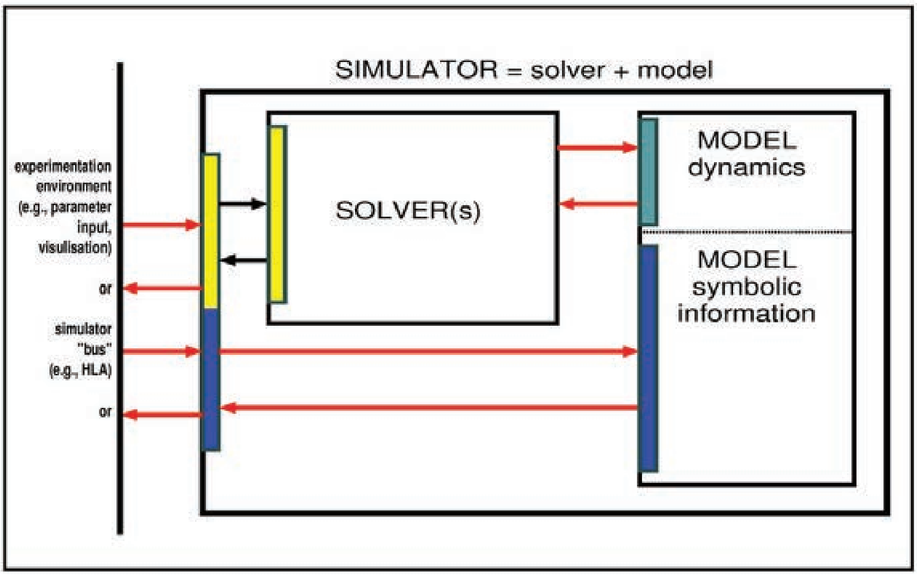
co-simulation: how? (when IP protected)





Minimally, Constrained Stable Switched Systems and Application to Co-Simulation
C Gomes, RM Jungers, B Legat, H Vangheluwe 2018
IEEE Conference on Decision and Control (CDC), 5676-5681

Model-Solver Interface
Simulator-Environment Interface

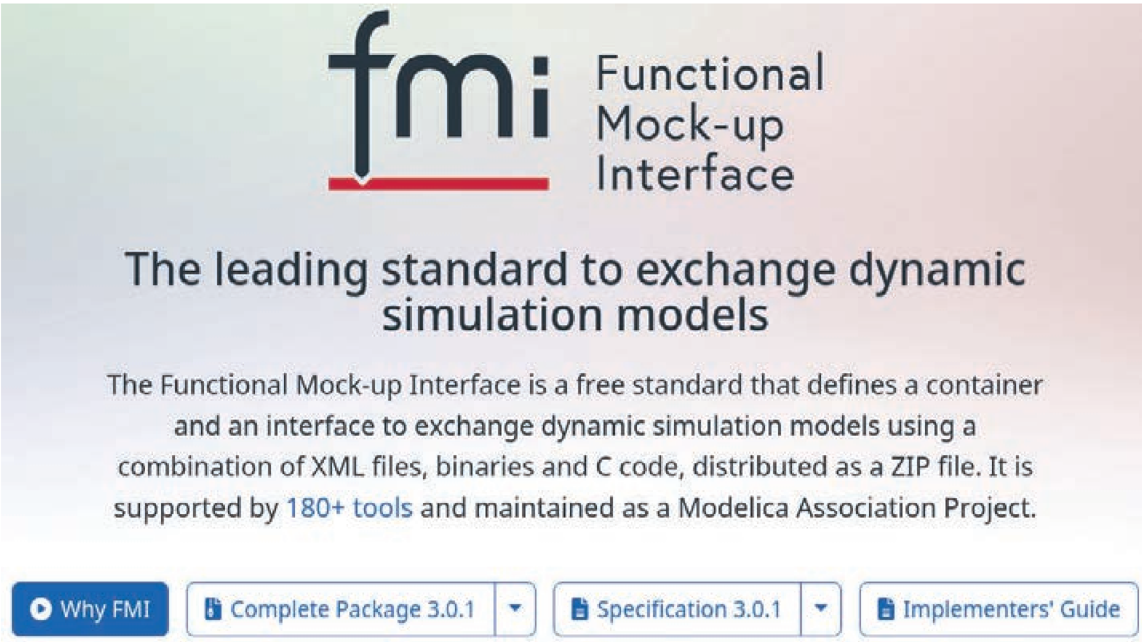


DSblock

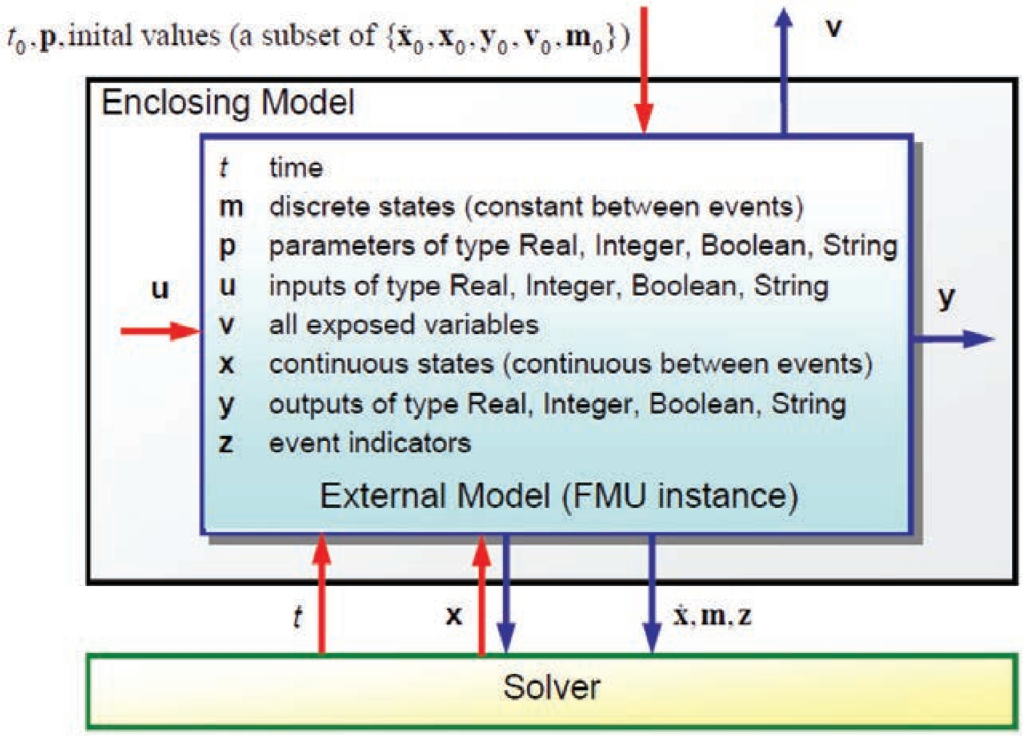
MSL-EXEC

Martin Otter and Hilding Elmquist.
The DSblock interface for exchanging model components. Eurosim '95 Simulation Congress. pp. 505- 510. 1995.

Henk Vanhooren, Jurgen Meirlaen, Youri Amerlinck, Filip Claeys, Hans Vangheluwe, and Peter A. Vanrolleghem.
WEST: Modelling biological wastewater treatment. Journal of Hydroinformatics , 5(1):27--50, 2003.



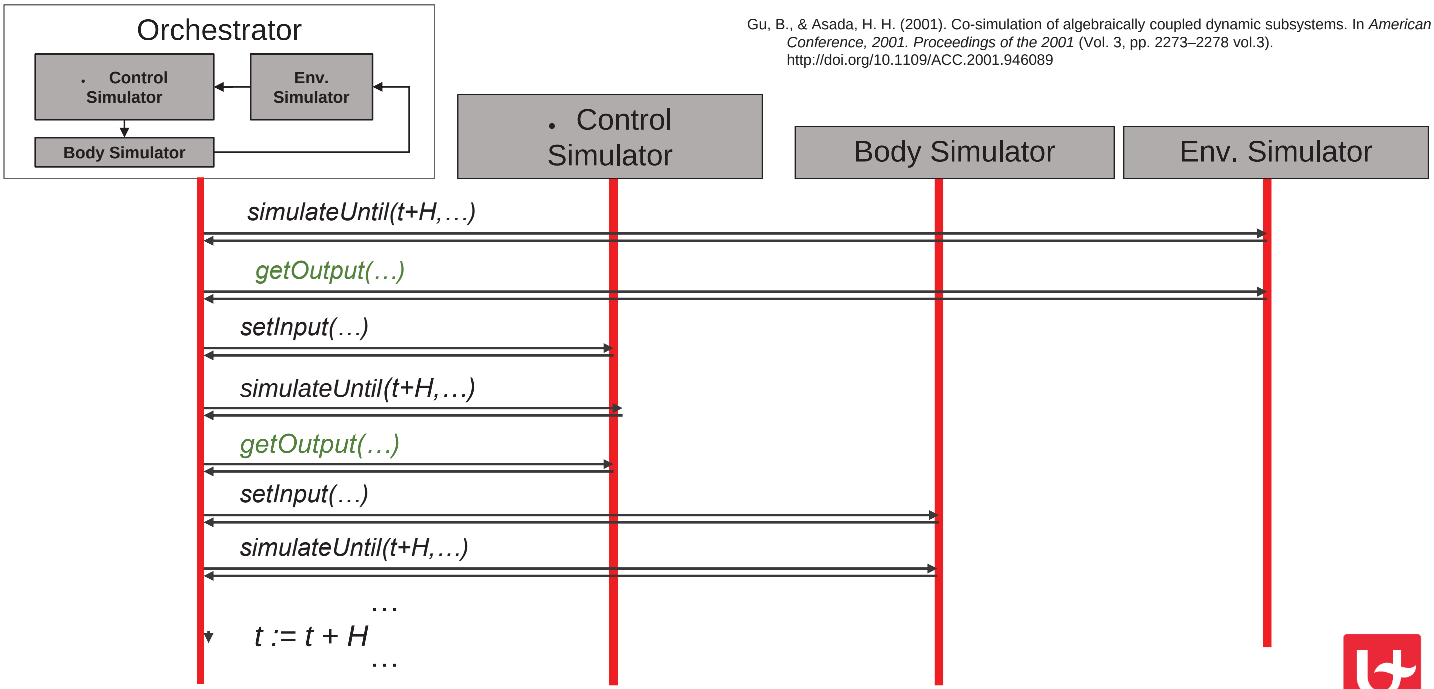
https://fmi-standard.org/



t_0, p, inital values (a subset of $\{\dot{\mathbf{x}}_0, \mathbf{x}_0, \mathbf{y}_0, \mathbf{v}_0, \mathbf{m}_0\}$)
v
Enclosing Model
t time
m discrete states (constant between events)
p parameters of type Real, Integer, Boolean, String
u inputs of type Real, Integer, Boolean, String
v all exposed variables
x continuous states (continuous between events)
y outputs of type Real, Integer, Boolean, String
z event indicators
External Model (FMU instance)
u
y
t
x
$\dot{\mathbf{x}}$, m, z
Solver

Co-simulation: how?

Gu, B., & Asada, H. H. (2001). Co-simulation of algebraically coupled dynamic subsystems. In *American Control Conference, 2001. Proceedings of the 2001* (Vol. 3, pp. 2273–2278 vol.3). http://doi.org/10.1109/ACC.2001.946089



2 hands-on tutorials with foundations in Multi-Paradigm Modeling

Case Study: Adaptive Cruise Control System (ACCS)

1a: Modeling the ACCS using OpenModelica

Rakshit Mittal[1], Hans Vangheluwe[1]

1b: Verifying ACCS Safety Requirements using UPPAAL

Rizwan Parveen[2]

2a: Modeling and Analyzing the Architecture of the ACCS controller using AADL

Dominique Blouin[2], Anish Bhobe[3]

2b: Synthesizing Code for the ACCS controller using RAMSES

Dominique Blouin[2], Anish Bhobe[3]

[1]University of Antwerp – Flanders Make, Belgium

[2]Telecom Paris, France

[3]Institut Polytechnique de Paris, France

Understanding Model-driven Design with UPPAAL Model Checker

Thursday, June 13, 2024

1

AGENDA

- Introduction to the basic concepts of modelling and model checking.
- Get to know basic features of the UPPAAL model checker.
- Illustration of UPPAAL tool through a few examples in the context of the formal verification

Thursday, June 13, 2024

2

OUTLINE

1. The role of Model Checking in design validation

2. The UPPAAL Tool

   1. Introduction
   2. Building model and formalizing properties
   3. Verification: writing queries
   4. An example
   5. Installation instructions

3. References

Thursday, June 13, 2024 3

OUTLINE

1. The role of Model Checking in design validation
2. The UPPAAL Tool
    1. Introduction
    2. Building model and formalizing properties
    3. Verification: writing queries
    4. An example
    5. Installation instructions
3. References

Thursday, June 13, 2024

4

1. WHY DESIGN VALIDATION?

- Design Validation is important step to **ensure design correctness** at very early phase of SDLC

- **Traditional Techniques**:
  - **Simulation** (on an abstraction or a model of the system)
  - **Testing** (often conducted on the actual product once built)

- **Formal Methods** (aimed at exhaustive validation)
  - different formal approaches are used for different kind of requirements.
  - The complexity of these methods made them only accessible to specialists (mathematicians).
  - **Model Checking (MC)**
  - MC is the first technique that is truly accessible for "normal" engineers
  - Applicable to (finite-state concurrent systems → automatic) sequential circuits, communication protocols, software… a wider spectrum of applications

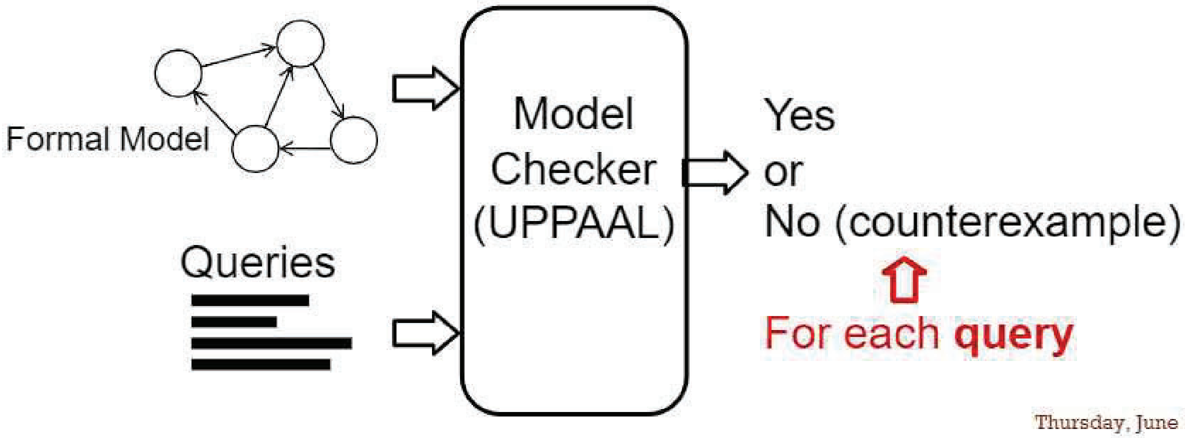
Thursday, June 13, 2024 5

PERFORM 3 STEPS FOR VERIFICATION

First, build a **model** for the system (abstract), in the form of a set of automata (called as Network of automata in UPPAAL)

Second, write the important **properties** to be verified using expressions, e.g. temporal logic (in case of UPPAAL, it is TCTL)

Third, use the model checker (a **tool like UPPAAL**) to generate the space of all possible states and to exhaustively check whether a property hold in each and everyone of the possible BEHAVIOURS of the model.



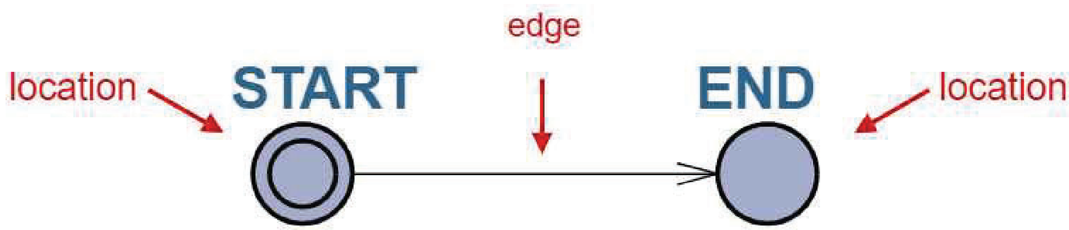
Thursday, June 13, 2024 6

OUTLINE

1. The role of Model Checking in design validation
2. The UPPAAL Tool
   1. Introduction
   2. Building model and formalizing properties
   3. Verification: writing queries
   4. An example
   5. Installation instructions
3. References

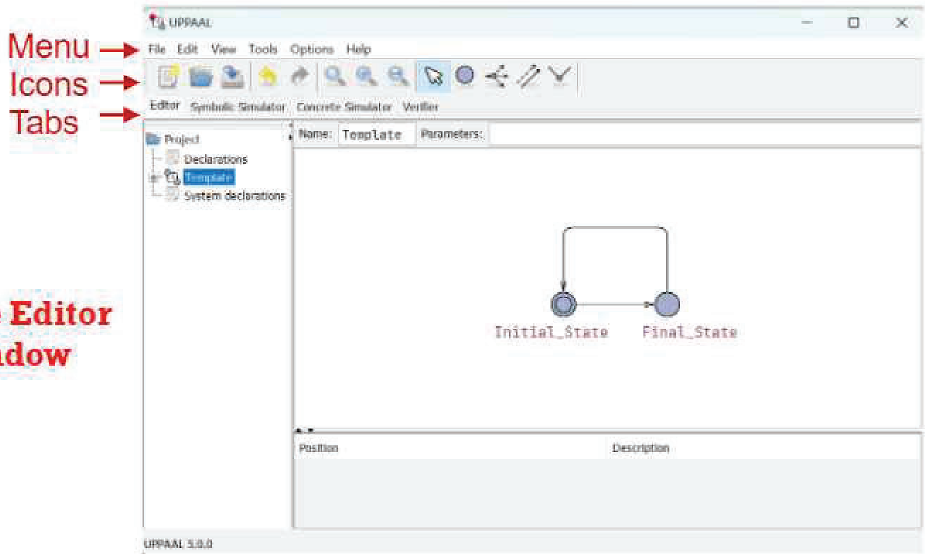
Thursday, June 13, 2024 7

2. UPPAAL



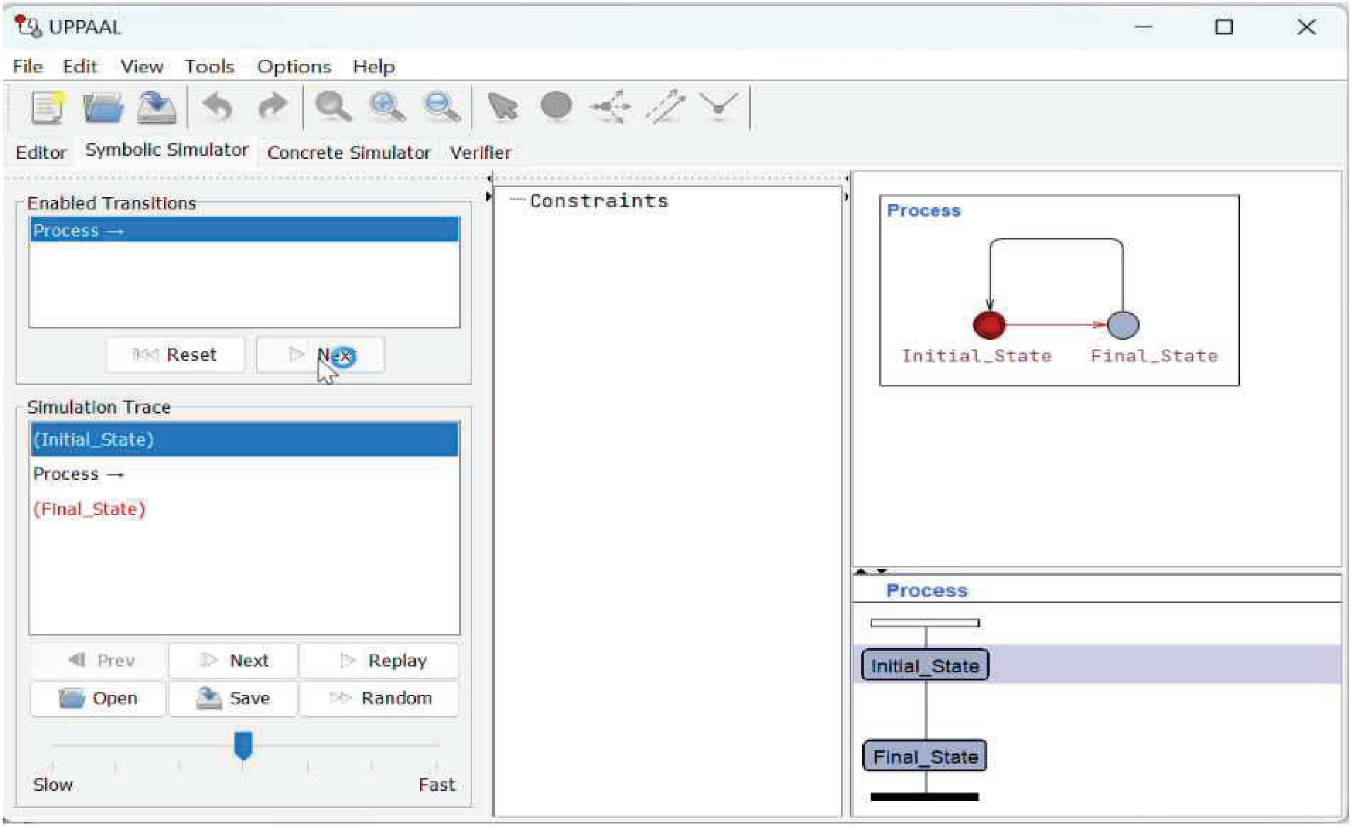
- Enable verification via automatic model- checking.
- It consists of three main parts:
  - a Graphical editor (run on the user's computer) and
  - a simulator
  - a verifier

All constitutes to a model-checker engine (by default executed on the same computer as the user interface, but can also run on a more powerful server)



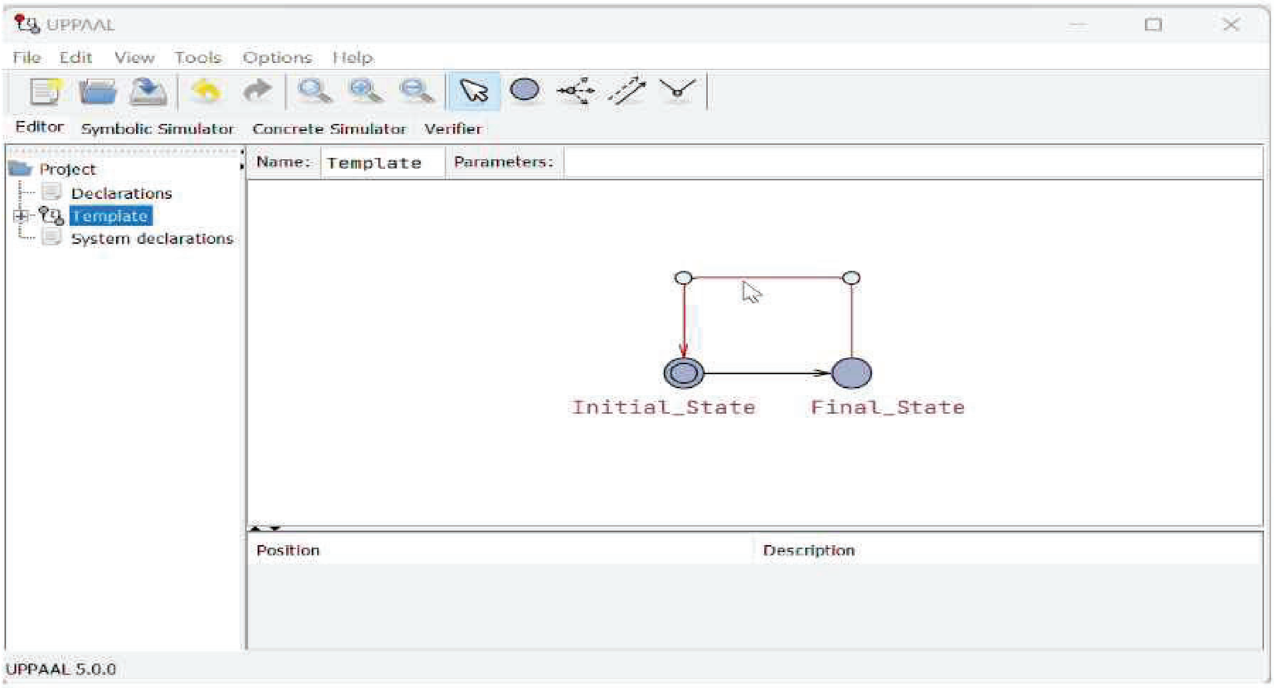
The Editor Window

THE SIMULATOR WINDOW



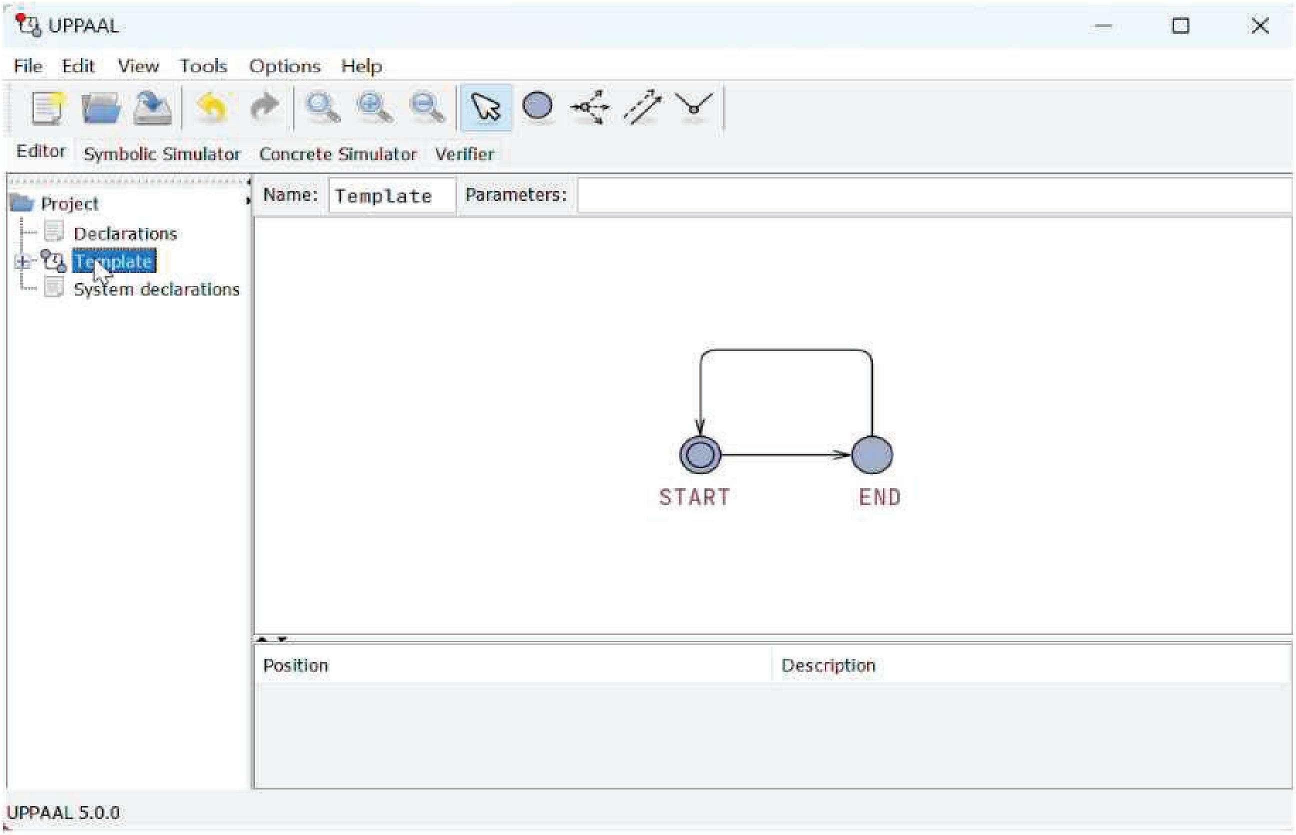
EDIT THE MODEL AND VERIFY

- An UPPAAL model is built as **a set of concurrent *processes***.
- Each process is graphically designed as a ***timed- automaton***.



Thursday, June 13, 2024 10

THE VERIFIER WINDOW: INSERT QUERY



Thursday, June 13, 2024

11

OUTLINE

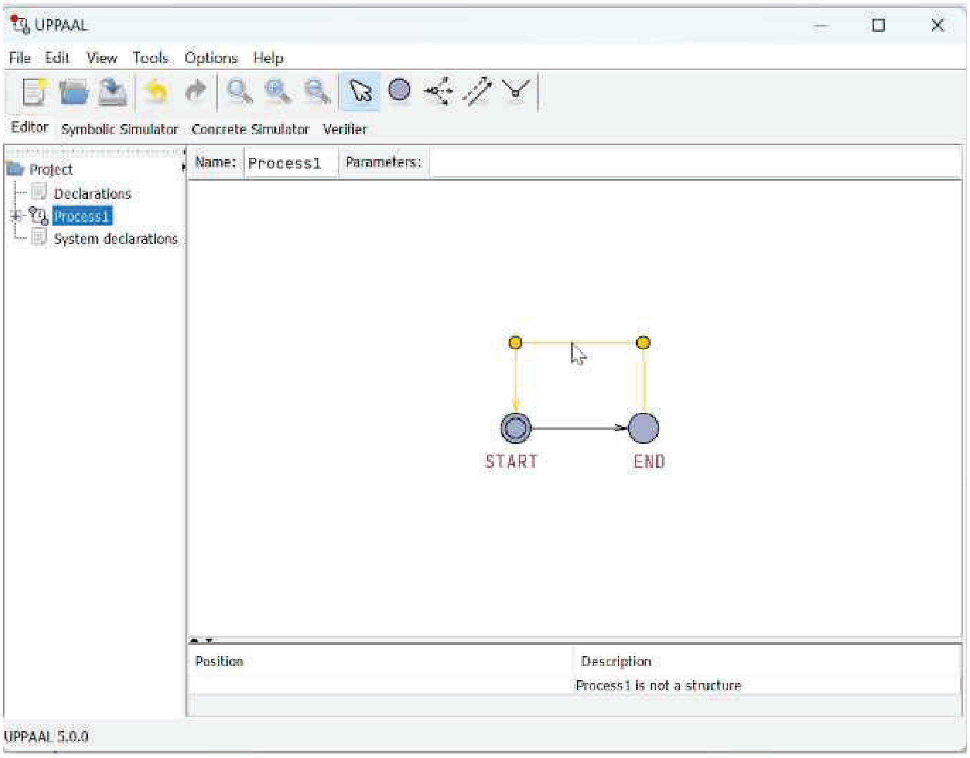
1. The role of Model Checking in design validation
2. The UPPAAL Tool
   1. Introduction
   2. Building model and formalizing properties
   3. Verification: writing queries
   4. An example
   5. Installation instructions
3. References

Thursday, June 13, 2024 12

2. MODELLING WITH UPPAAL
Synchronisations: Guard and channels

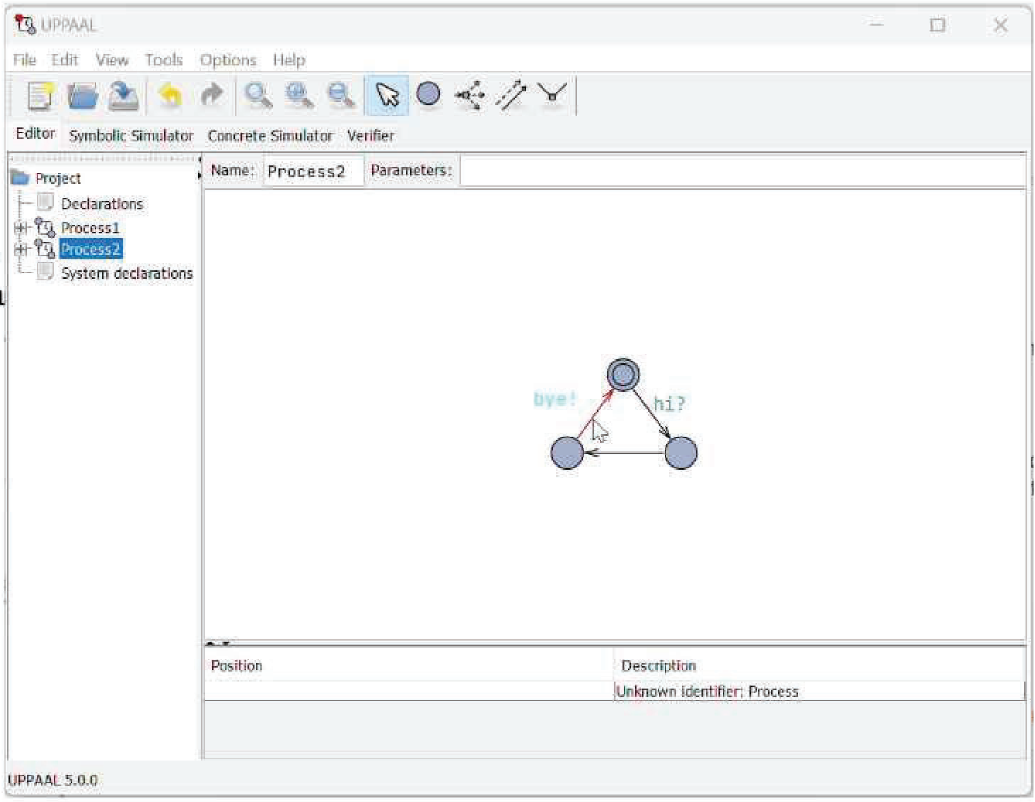
- Edges are annotated with ***selections, guards, synchronisations*** and ***updates***

- Using **channels** two (or more) processes to take a transition at the same time.
- Declare the **channel** (*c*) under declaration using keyword ***chan***.
- One process will have an edged annotated with *c**!*** *(send)* and the other(s) process(es) another edge annotated with *c**?*** *(receive)*



SYNCHRONISATIONS : GUARD AND CHANNELS

– If at a specific instant there are several possible ways to have a pair *c!* and *c?*, one of them is non-deterministically chosen during model checking.



Thursday, June 13, 2024 14

COUNTEREXAMPLE AND DIAGNOSTIC TRACE

This example will show:
A. how an error in model can be traced.

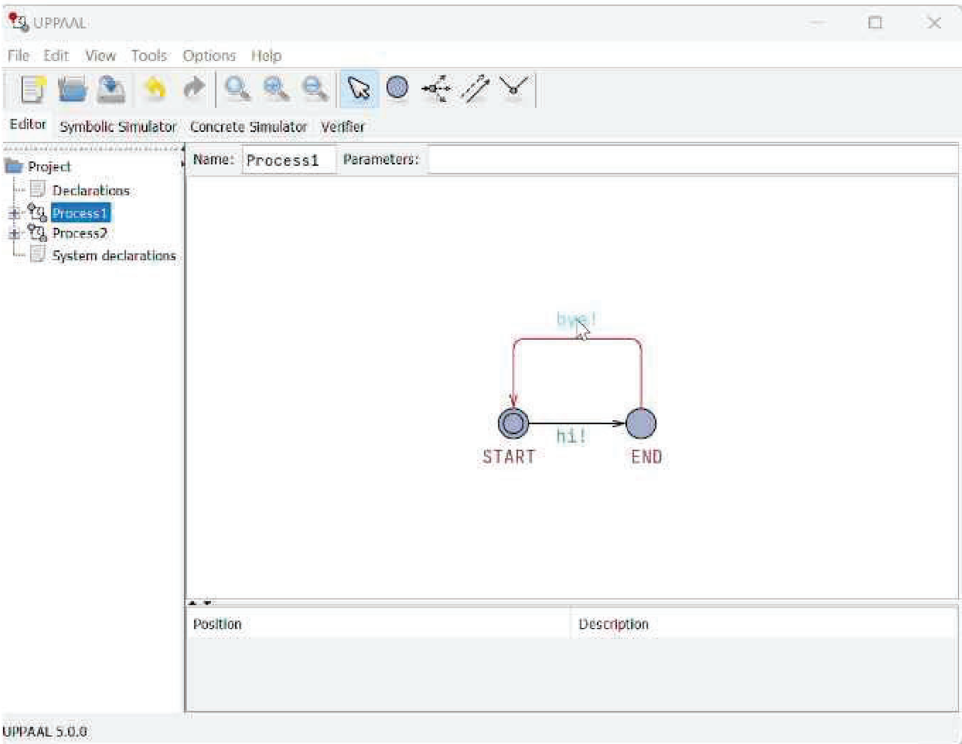
B. How to formalize query in TCTL.

- Verifying Properties:

1. to ensure that the model behaves as the system we wanted to model.
2. to detect some errors in the original design)

Formalize properties:

– Ex. In a network protocol, if a message is sent, it will be eventually received.

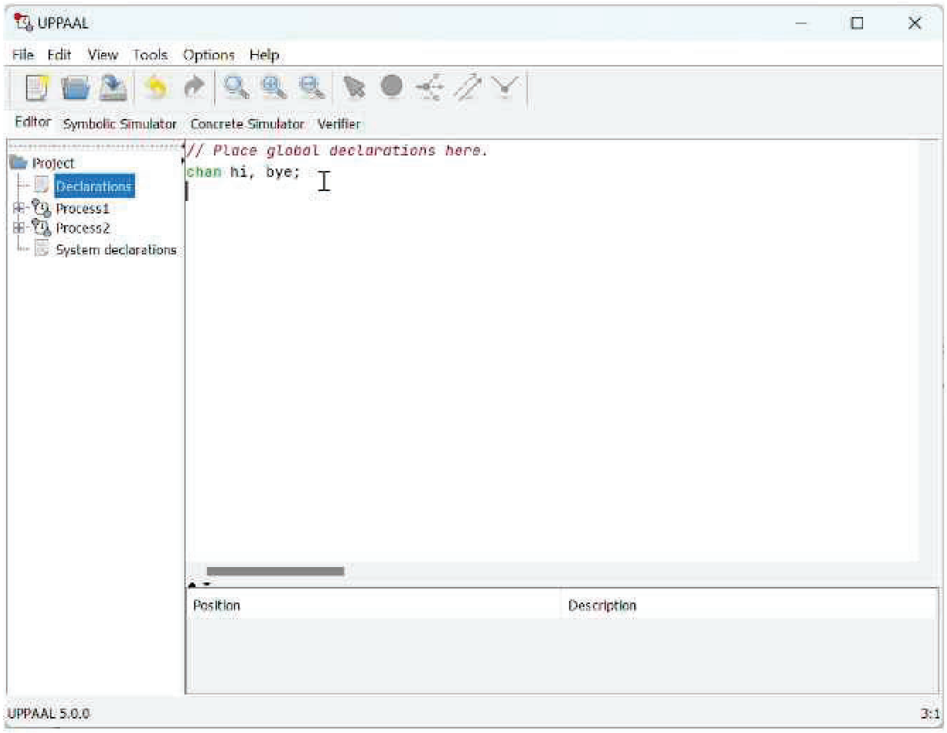


UPPAAL understands Timed Computational Tree Logic (TCTL). That means it is required to formalize those properties in TCTL (similar to LTL/CTL)

UPDATE AND GUARD

- **A guard is** an expression (a condition/action on the transition).
- It uses the variables and clocks of the model in order to indicate when the transition is enabled or not.
  - Note that several edges may be enabled at an specific time but only one of them will be fired → leading to different potential **interleavings**

An update is an expression that is evaluated as soon as the corresponding edge is fired. This evaluation changes the state of the system.



Thursday, June 13, 2024 16

EDGES

- **Three different kinds of synchronizations**:
  - **Regular channel** (leading to Binary Synchronization)
  - **Urgent channel:** time cannot lapse
  - **Broadcast channel:** all these transitions are enabled at receiving ends.

  - The update expression on an edge synchronizing on *c!* is executed **before** the update expression on an edge synchronizing on *c?*

31

Thursday, June 13, 2024

17

STATES (AKA LOCATIONS)

- **States can be of three different types** (that can be assigned by double-clicking on the location):
  - **Initial**
  - **Urgent** (time is not allowed to pass when a process is in an urgent location)
  - **Committed** (When a model has one or more active committed locations, no transitions other than those leaving said locations can be enabled)
  - **Normal** (all the rest)

A RECOMMENDATION ON MODELING

- **The state space grows very quickly** with the model complexity (state space explosion). It is necessary to:
  - It is better to model at suitable level of abstraction of a system.
  - Identify important properties to model and properties that are essential to be verified.

- More specifically:
  - The use of committed locations can reduce significantly the state space, but it can possibly take away relevant states.
  - The number of clocks and variables

This is rather an "art" (model checking may not be so "perfect" but it helps a designer to think)

51

Thursday, June 13, 2024

19

OUTLINE

1. The role of Model Checking in design validation
2. The UPPAAL Tool
   1. Introduction
   2. Building model and formalizing properties
   3. Verification: writing queries
   4. An example
   5. Installation instructions
3. References

Thursday, June 13, 2024 20

VERIFICATION AND TYPES OF QUERIES IN UPPAAL

The UPPAAL query language (TCTL) can be classified as:

[1] Reachability properties*. A specific condition holds in some state*. Expressed as : `E<> p` "Exists eventually p"

[2]. Safety properties*. A specific condition holds in all the states of an execution path.*

`E[] p` *"Exists globally* `p`*" (p holds for all the states of the path)*

`A[] p` "Always globally `p`" (For each (all) execution path p holds for all the states of the path)

[3]. Liveness properties*. A specific condition is guaranteed to hold* ***eventually*** *(= at some moment)*

`A<> p` "Always eventually p" (p holds for at least one state of the path)

q-->p "q always leads to p"

[4]. Deadlock properties*. If a deadlock is possible or not in the model*

`A[] not deadlock`

OUTLINE

1. The role of Model Checking in design validation
2. The UPPAAL Tool
   1. Introduction
   2. Building model and formalizing properties
   3. Verification: writing queries
   4. An example
   5. Installation instructions
3. References

Thursday, June 13, 2024 26

MOVEMENT OF A CAR

1. Avoiding Obstacle
2. Maintaining safe distance from the vehicle in front

To avoid obstacle, there are two actions:

1. Slow down the speed of the car

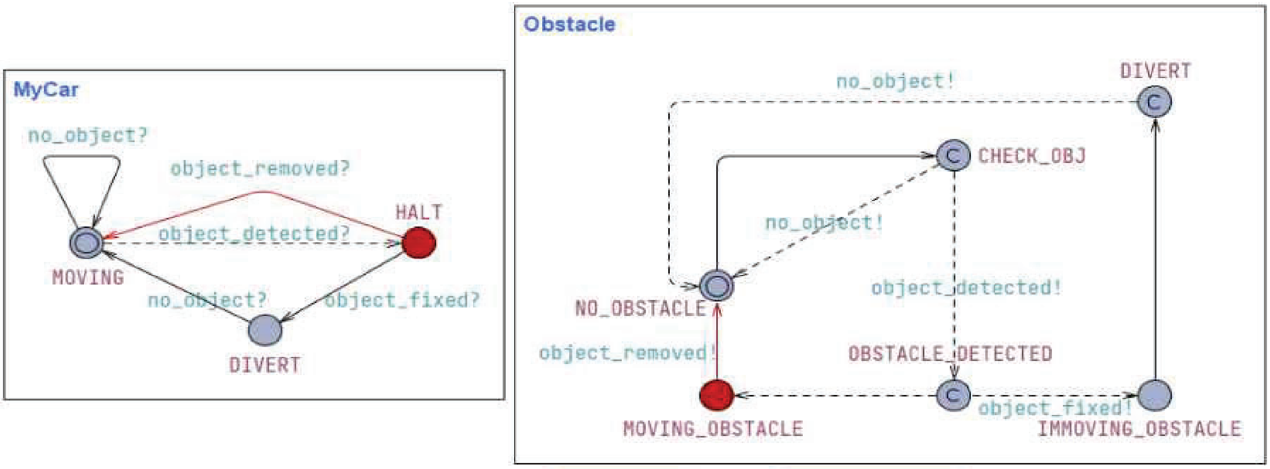
2.1 If it is movable obstacle, wait till the obstacle is removed from the path and resume moving.

2.2. If it is non-movable obstacle, wait and divert the path.

In a advanced model, we can add path planning/shortest part, etc. algorithm from the state of "divert".

Thursday, June 13, 2024 27

1. AVOIDING OBSTACLE



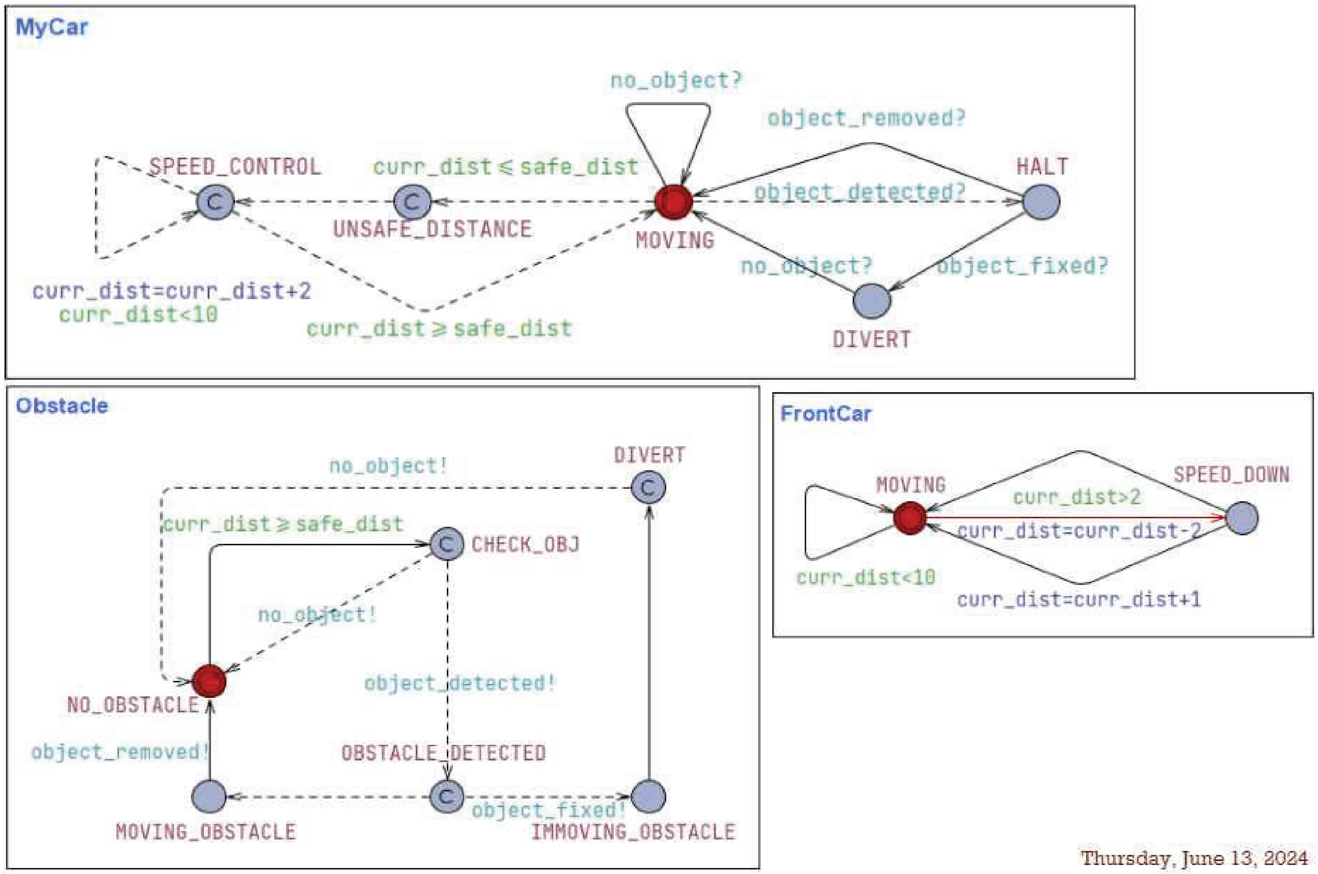
- The model shows two automata: MyCar and Obstacle
- Assume my car is in moving state. It keeps moving until it detects an obstacle.
- In the event of a obstacle detected, my car has two options:
  - A. To wait for obstacle to move away from the path and then continue moving on the path
  - OR B. My car chooses a different path and resume moving.

Thursday, June 13, 2024 28

2. MAINTAINING SAFE DISTANCE FROM THE VEHICLE IN FRONT

- We add one automaton in the existing model to represent the operation of a front car.
- Let's assume if this front car slows down its speed, maybe during a heavy traffic, that means the distance between my car and front car will be reduced and not in a safe range.
- There is a minimum safe-distance which my car has to maintain from the front car. Therefore, whenever the front car reduces the speed, my car checks if it is moving on a safe distance or not.
- If not, my car control its speed (reduce) and go to safe moving only when safe distance is recovered (that represented by FrontCar's normal moving state).

2. MAINTAINING SAFE DISTANCE FROM THE VEHICLE IN FRONT



Thursday, June 13, 2024 30

VERIFICATION

- Check for deadlock
- Check that MyCar should not be in MOVING state when obstacle detected.
- Check MyCar always maintain safe distance from the FrontCar

Thursday, June 13, 2024 31

OUTLINE

1. The role of Model Checking in design validation

2. The UPPAAL Tool

    1. Introduction
    2. Building model and formalizing properties
    3. Verification: writing queries
    4. An example
    5. Installation instructions

3. References

Thursday, June 13, 2024

32

LEARNING OBJECTIVE

- How to build model with UPPAAL?
- Identifying important properties and formalizing them.
- Verify important properties of the model.

Task to be performed:

- Follow the UPPAAL installation instruction given on next slide.
- Download the pre-build model of the car.
- Improve this model by implementing task #2: maintaining safe distance
- Write Safety properties and verify them

Thursday, June 13, 2024 33

INSTALLATION INSTRUCTIONS

- Make sure you have the Java version installed as per latest UPPAAL requirement.
  - E.g.: www.java.com/es/download/manual.jsp
- Go to the UPPAAL page: www.uppaal.org
- Click on the download tag and then on the link Uppaal 5.0 (current official release)

LINK: https://uppaal.org/downloads/#uppaal5.0

- Fill the (academic) license agreement form. Click on "Register & Download". You may need to provide your university email id to get this license.
- Unzip files
- To run UPPAAL double-click the file uppaal.jar

Thursday, June 13, 2024 34

REFERENCES

Some of the following references are used for creating this presentation and some useful for further reading

- Slide Credit: Julián Proenza, Systems, Robotics and Vision Group. UIB. SPAIN

- UPPAAL (available at *www.uppaal.org*)
  - *A Tutorial on Uppaal,* 17 Nov 2004 by G. Behrmann, A. David, and K. G. Larsen.
  - UPPAAL Online Help

- Model Checking:

- Behrmann, G., David, A., Larsen, K.G. (2004). A Tutorial on UPPAAL. In: Bernardo, M., Corradini, F. (eds) Formal Methods for the Design of Real-Time Systems. $SFM-RT$ 2004. Lecture Notes in Computer Science, vol 3185. Springer, Berlin, Heidelberg.

- Bouyer, Patricia (2009). "Model-checking Timed Temporal Logics". In: Electronic Notes in Theoretical Computer Science 231. Proceedings of the 5th Workshop on Methods for Modalities(M4M5 2007), pp. 323–341. ISSN: 1571-0661.

Thursday, June 13, 2024 35

Summer School on CPS & IoT 2024

Modeling, Analyzing and Synthesizing Embedded Systems with AADL using RAMSES

Dominique Blouin, Associate Professor
Anish Bhobe, PhD Student
Télécom Paris, Institut Polytechnique de Paris
dominique.blouin@telecom-paris.fr
anish.bhobe@telecom-paris.fr

Content

- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

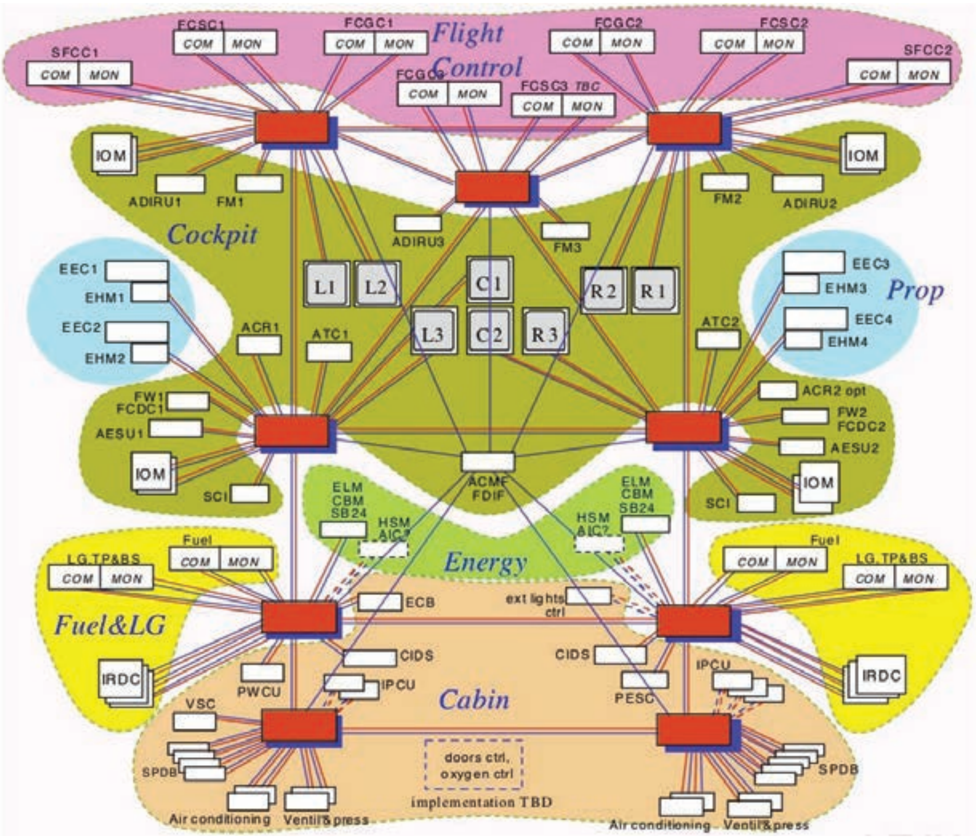
Architectures

- If you fail to plan, you are planning to fail!

Painting by Duplessis.
Source: Wikipedia

- Architectures are not only useful for buildings, but for complex software systems too!

Example of a Complex Avionics Architecture



Architecture Description Languages (ADL)

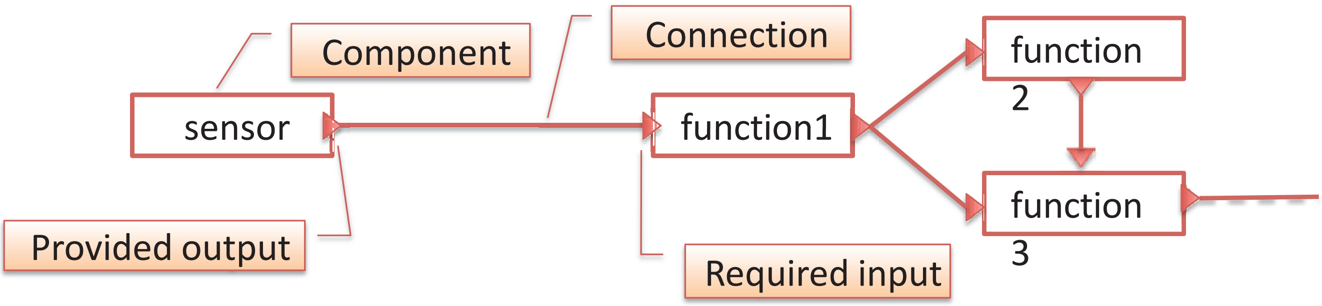
- Formal, i.e., based on a mathematically sound definition of their semantics.
  - Meant to formally verify/prove expected properties of a computer system
  - E.g.: Wright ADL, data flow graphs, state machines, …
- Domain-specific.
  - Meant to describe the design and implementation of computer systems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

- 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.

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…

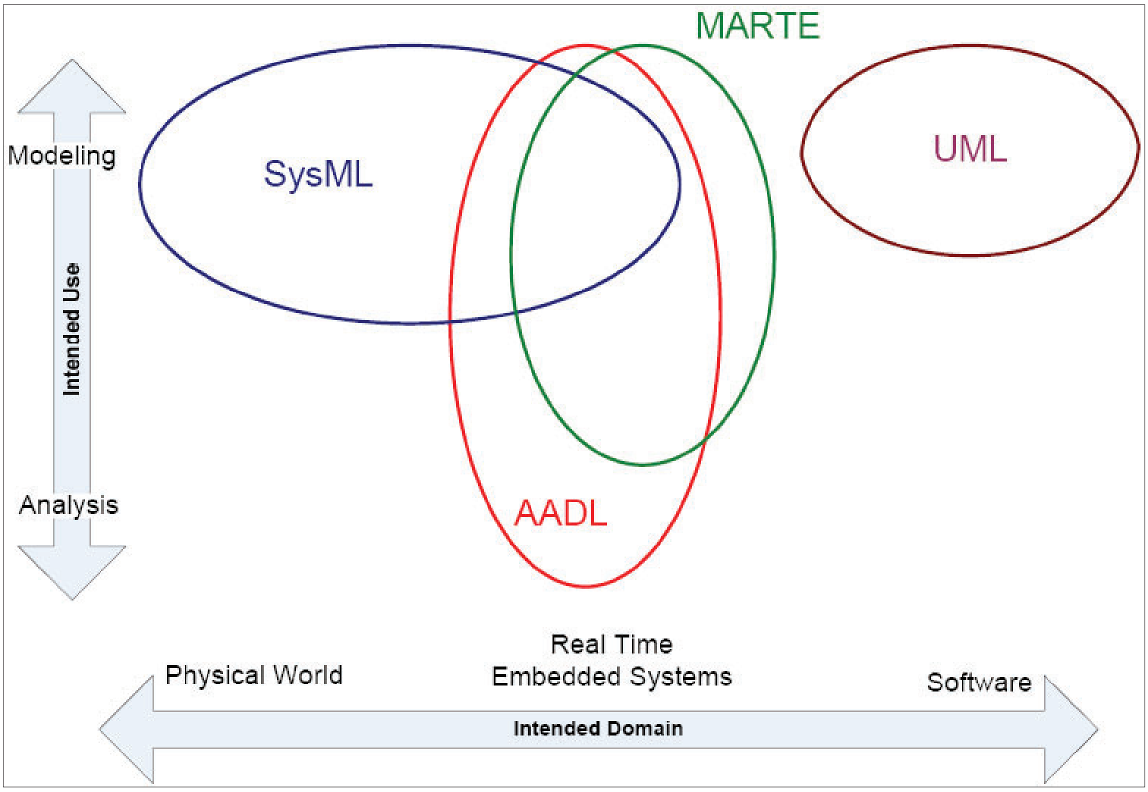
- What about the **semantics**?

AADL: Architecture Analysis & Design Language

- An **ADL** for real-time embedded systems.
  - Component-based (components, interfaces and connections).
  - Defines properties for real-time and embedded systems analysis.
    - Scheduling policy, compute execution time, latency…
    - Software components to hardware components allocation.
  - SAE Standard AS5506
    - https://www.sae.org/standards/content/as5506d/

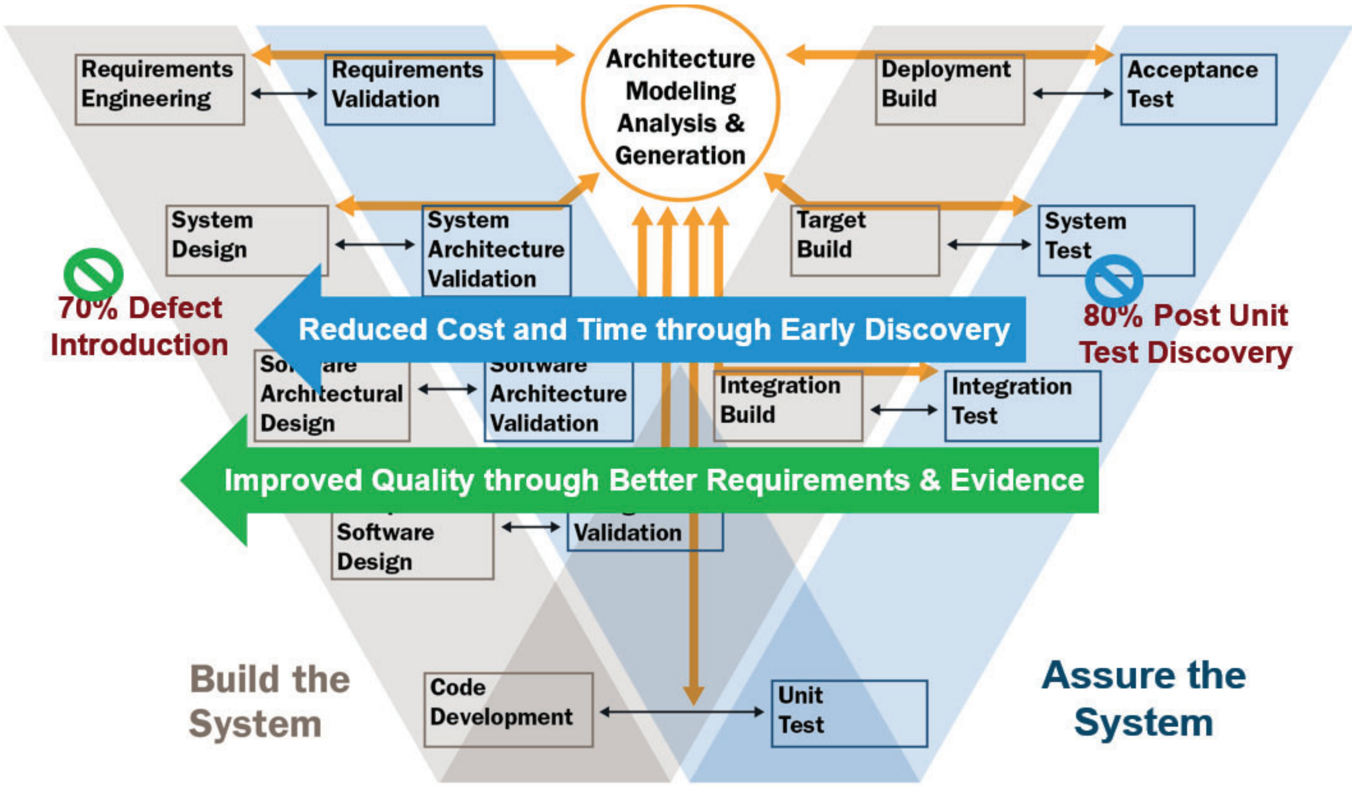
- **Objective**: Support the design of such systems.
  - Standardized semantics (formulated with natural language).
  - Textual and graphical syntax (blended syntax).
  - Strongly typed (components category, composition rules, …).
  - Extensible (property definition language and annexes).

Comparison with other Architecture Description Languages (ADL)



Source: Steven P. Miller, AADL Standards Winter Meeting, 2011

Architecture-Centric Virtual Integration Process (ACVIP)



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

General Characteristics of AADL

- **Components** are the main modeling entities.

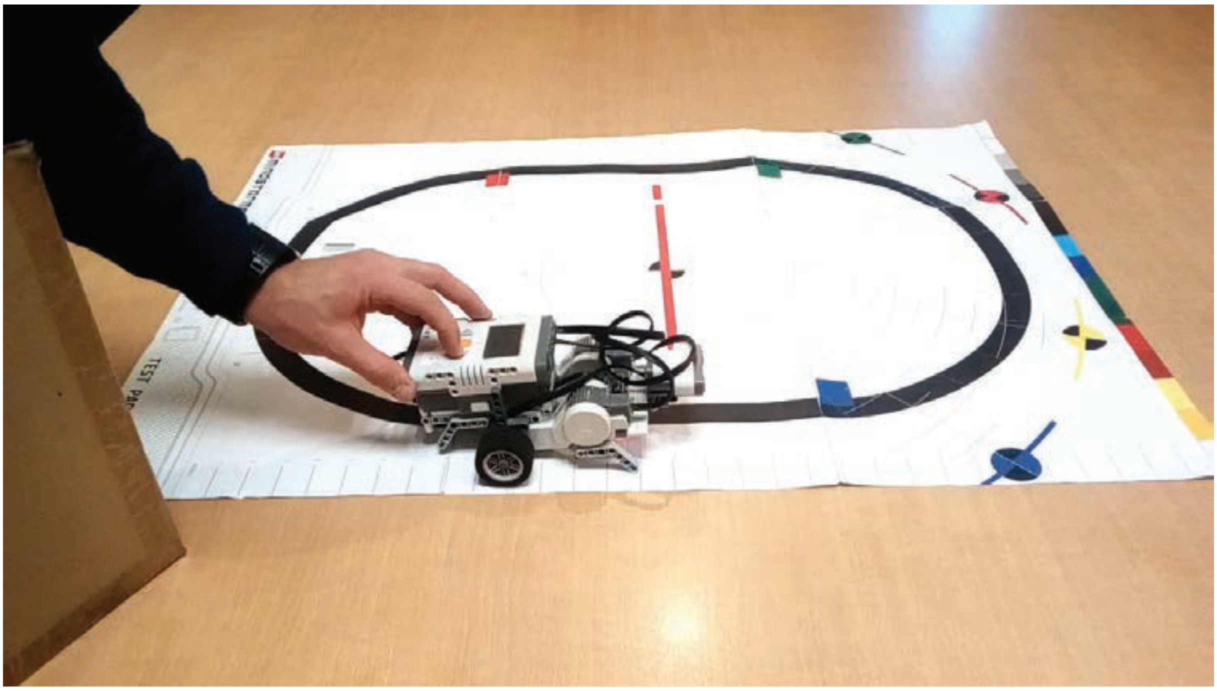
- The standard defines **categories** of components (keywords of the language), e.g.:
  - Composite: System, Abstract
  - Software: Process, Thread, Data, Subprogram…
  - Hardware: Processor, Memory, Bus, Device…

- Components definition is divided into **types**, **implementations**, and **subcomponents:**
  - Type: Defines how the component is viewed from outside (e.g., interaction interfaces)
  - Implementation: Defines the internal structure of the component (e.g., subcomponents)
  - Subcomponents: Instances of components, starting from a root system implementation.

- Components are structured into **sections** identified by keywords of the language (e.g., **features**, **properties**, **subcomponents**).
  - Components can be declared in any order.
  - The language is case insensitive.

Running Example: Line-Follower Robot with Obstacle Detection

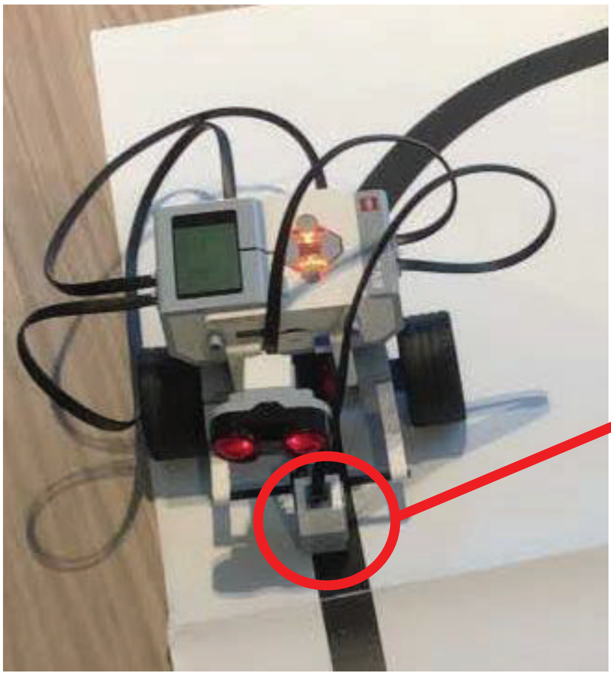
- Purpose: **Follow a line** to carry an object from point A to point B.
  - **Stop** when there is an obstacle (e.g., another robot), and **restart** when the obstacle is no longer there.

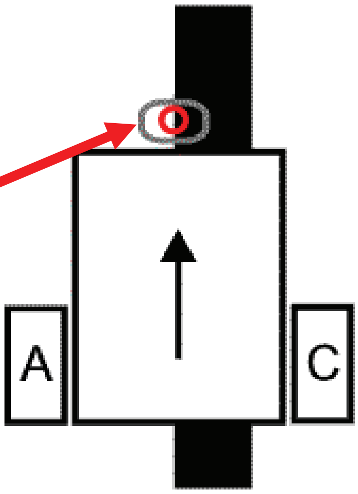


How it works…

- **Sensors**: light and sonar sensors.
- **Actuators**: Two wheels motors.
- **Brick** (includes the execution platform)

- **Control software**
  - PID controller…





AADL System Level Viewpoint Component Categories

- Two categories: **Abstract** and **System**.

Abstract **System**

- Different purposes:
  - Represent from a very abstract viewpoint the main constituent of a system, its interfaces and connections.
  - System:
    - Aggregates by composition subcomponents describing the execution platform and subcomponents describing the software architecture.
    - Define the main operational modes of the system
  - Abstract:
    - Define structure and interaction without knowing yet the nature of the component. E.g., system functions…

- What category should we use for the overall robot system?

Content

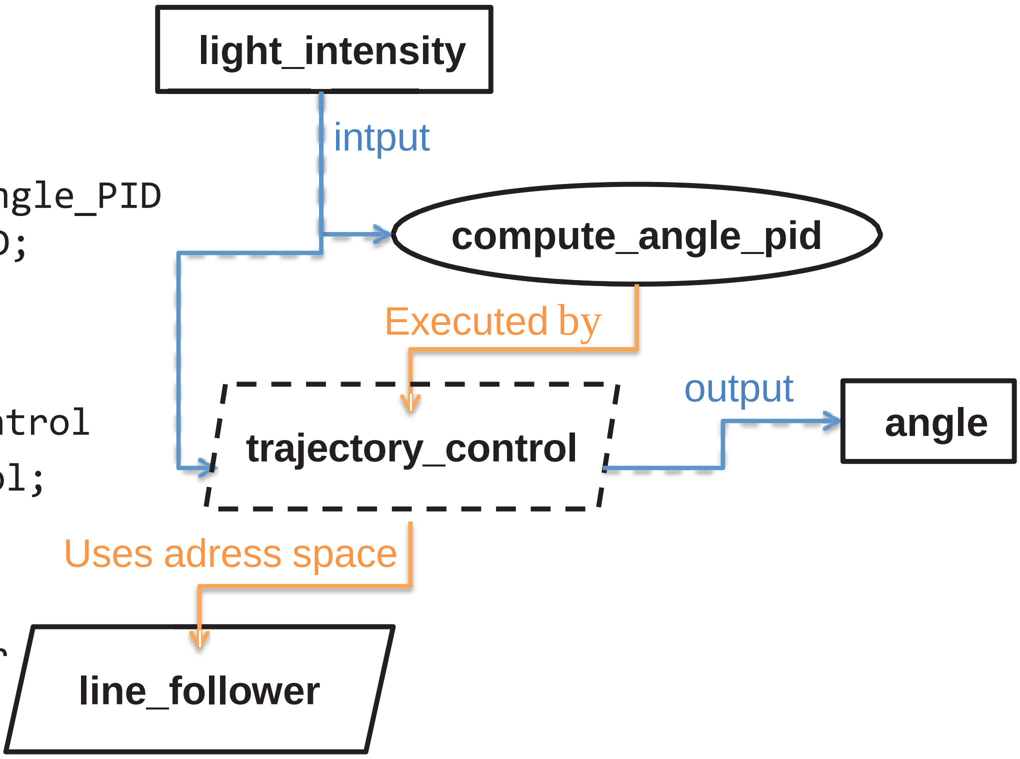
Software Architecture Viewpoint Categories

- 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 (schedulable unit) executing a sequence of functions.
  - Process: memory address space allocated for the execution of its thread subcomponents.
  - Etc.

- These categories focus on **operating system** and **programming** components.

Example of Software Components Types

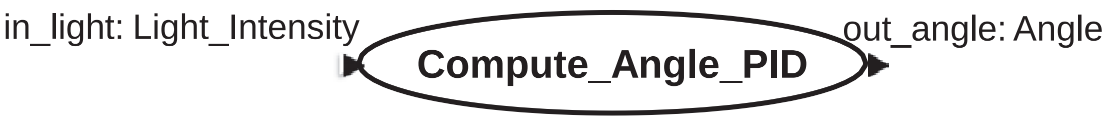
```
data Light_Intensity
end Light_Intensity;

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 Features

- Parameters:
  - `in`, `out`, or `inout`
  - Usable for subprograms only



- Requires or provides data access:
  - Usable for subprograms and threads

- Ports:
  - `in`, `out`, or `inout`
  - `data`, `event` or `event data`
  - Usable for threads, processes and systems.

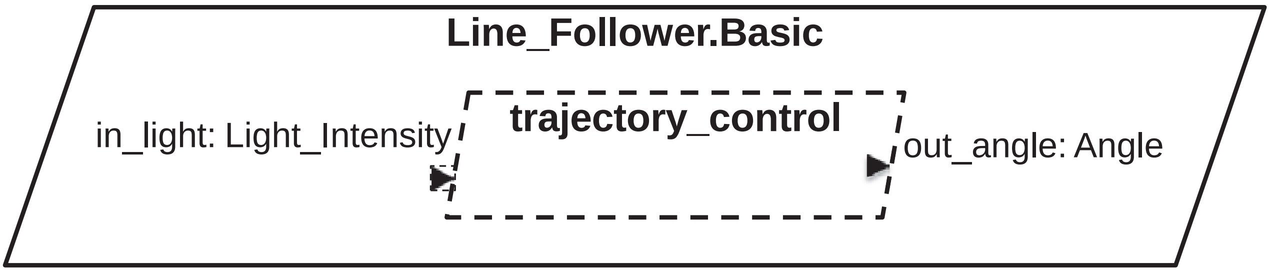
Semantics of Software Components Features

- **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:
    - Data becomes **available** on an input port when the thread **starts its execution**. Data not used during the previous execution of the thread is lost. Data is **not updated during the execution** of the task.
    - Data produced on an output port is sent to the recipient port at the **end** of the producer task execution.

Next, compose components

- Create thread subcomponent(s) in a process implementation:
  - Graphical syntax:



  - Textual syntax:

```
process implementation Line_Follower.Basic
    subcomponents
        trajectory_control: thread Trajectory_Control.Basic;
end Line_Follower.Basic;
```

Next, define subprogram calls in thread implementations

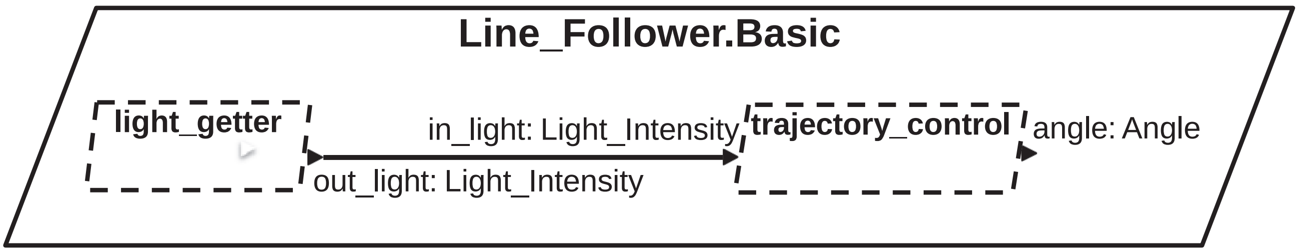
- Subprogram calls in threads:

```
thread implementation Trajectory_Control.Basic
    calls
        call_sequences {
             compute_angle: subprogram Compute_Angle;
             compute_power: subprogram Compute_Power;
        };
end Trajectory_Control.Basic;
```

Sequence of subprogram calls

Next, connect components

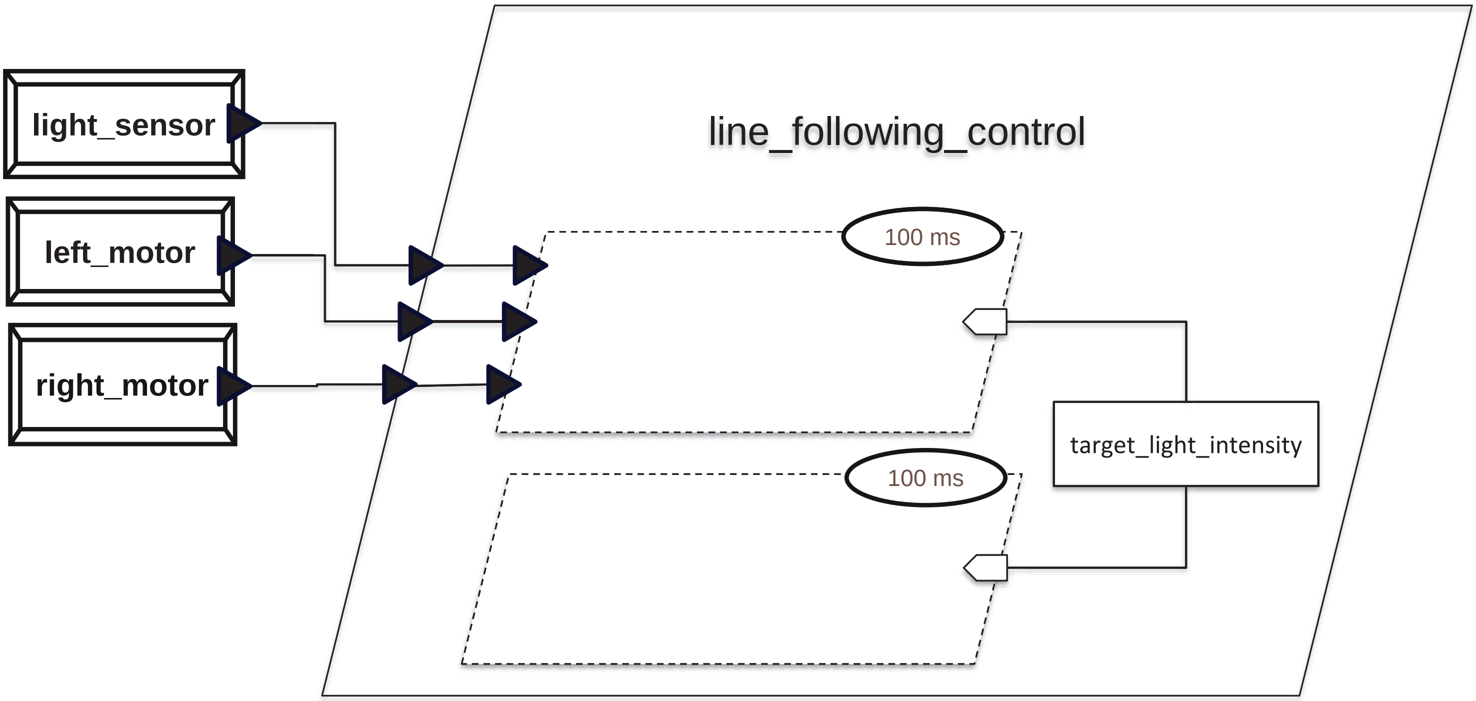
- Components features are connected **hierarchically**:
  - Thread subcomponents located inside a process.
- Graphical syntax:



- Textual syntax:

```
process implementation Line_Follower.Basic
    subcomponents
        trajectory_control: thread Trajectory_Control.Basic;
        light_getter: thread Light_getter.Basic
    connections
        light_intensity_con: port light_getter.out_light -> trajectory_control.in_light;
end Line_Follower.Basic;
```

Sensors and Actuators as Software Device Drivers

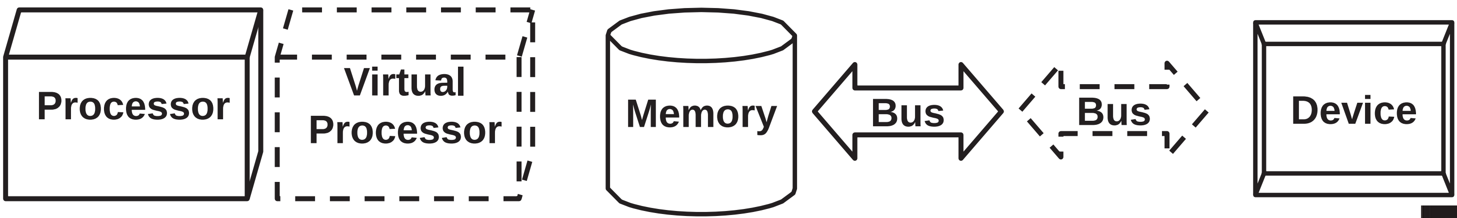


Content

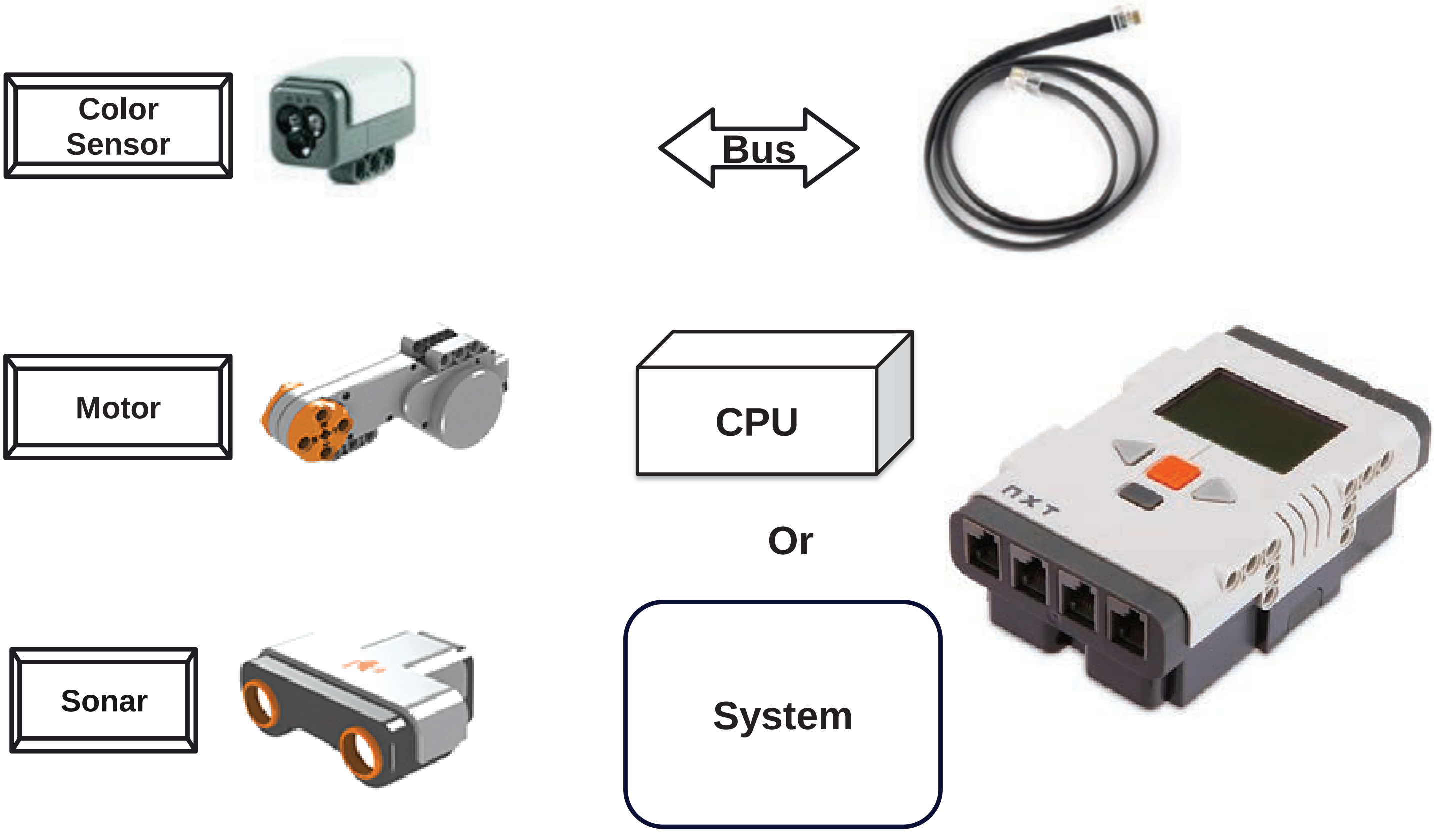
- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

Execution Platform Viewpoint Categories

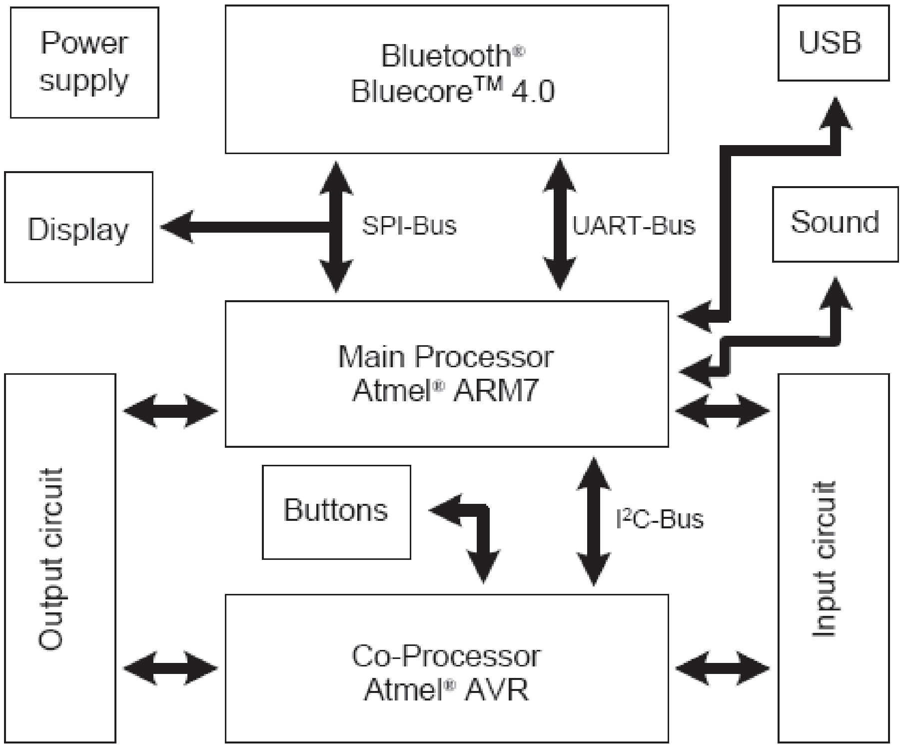
- Components categories dedicated to the **execution platform** specification.
  - **Processor**: Hardware computation unit + tasks scheduling capabilities
    - **Virtual Processor**: Processor logical partition.
  - **Memory**: Storage component (may be RAM, hard disk drive, cache, etc.).
  - **Bus**: Physical communication link (network cable, etc.).
    - **Virtual Bus**: Network
  - **Device**: Interface with the physical environment of the system (sensors/ actuators).
  - Etc.



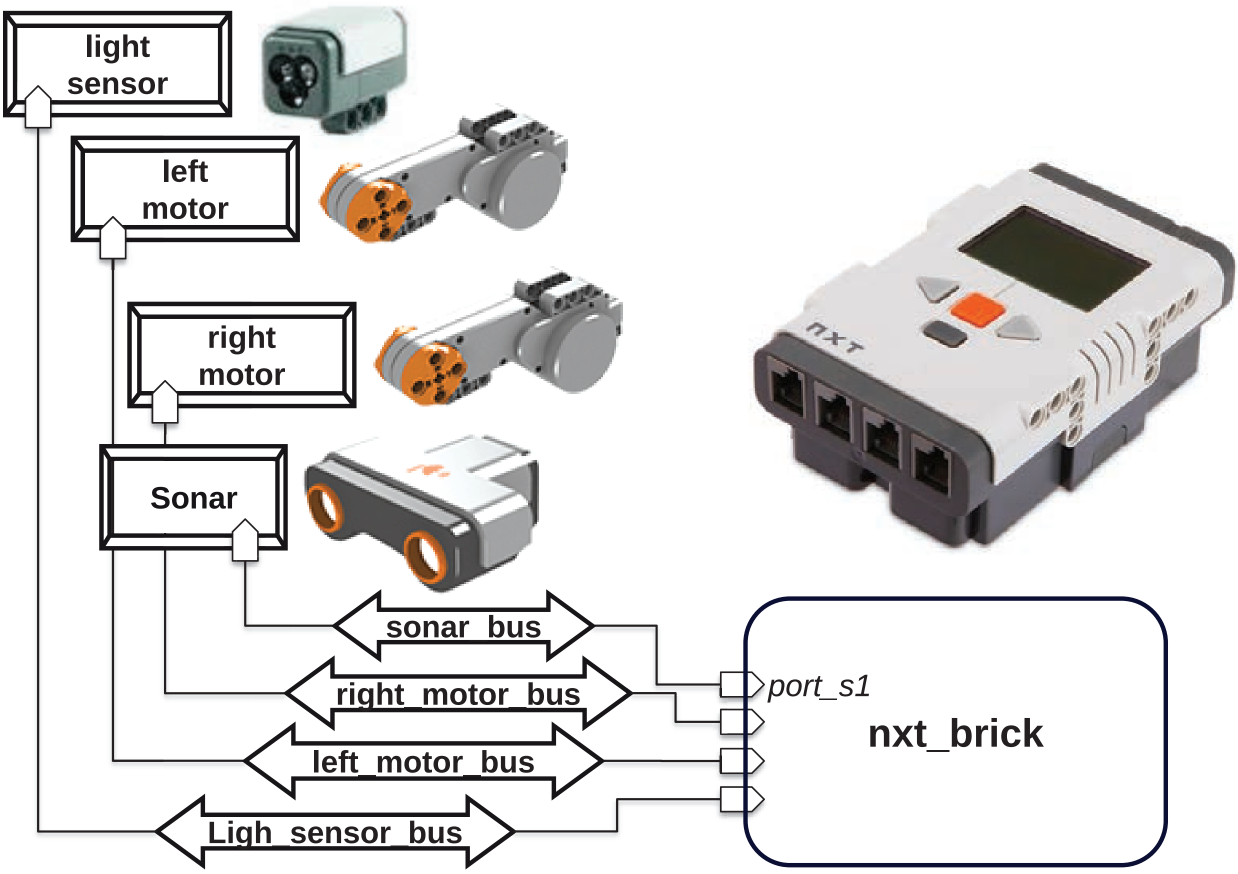
Example Execution Platform Components



Lego Robot Brick System



Lego Robot Hardware Platform



Lego Robot Hardware Platform in Textual Notation

- Define a **bus** for sensors and actuators wires.
- Add a **requires bus access** feature to the device.

```
bus Device_Bus
end Device_Bus;

device Light_Sensor
    features bus_access: requires bus access Device_Bus;
end Light_Sensor;
```

- **Connect** components to bus via bus access connection.

```
system implementation Robot_Hardware.Basic
    subcomponents
        light_sensor: device Light_Sensor;
        light_sensor_bus: bus Device_Bus;
        ...
         exec_platform: system Robot_Platform.Basic;
    connections
        light_sensor_con: bus access light_sensor_bus -> light_sensor.bus_access;
        platform_light_con: bus access light_sensor_bus -> exec_platform.light_sensor;
        ...
```

Content

- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

Components Composition Rules

- A component implementation may declare **subcomponents**.
- An AADL model is therefore a **tree of components** starting by a root component, usually of **system** category.
- **Legality rules** define what categories of components a component can contain as subcomponents.
  - Same case for which features a component type can own.
- Look at **OSATE help** for more info.
  - OSATE documentation includes nearly all the standard.

Components Composition Rules

Category	Allowed Subcomponent Categories
abstract	data, subprogram, subprogram group, thread, thread group, process, processor, virtual processor, memory, bus, virtual bus, device, system, abstract
data	data, subprogram, abstract
subprogram	data, subprogram, data
subprogram group	subprogram, subprogram group, data, abstract
thread	data, subprogram, subprogram group, abstract
thread group	data, subprogram, subprogram group, thread, thread group, abstract
process	data, subprogram, subprogram group, thread, thread group, abstract
processor	memory, bus, virtual processor, virtual bus, abstract
virtual processor	virtual processor, virtual bus, abstract
memory	Memory, bus, abstract
bus	virtual bus, abstract
virtual bus	virtual bus, abstract
device	bus, virtual bus, data, abstract
system	data, subprogram, subprogram group, process, processor, virtual processor, memory, bus, virtual bus, device, system, abstract

Packages

- Like in programming languages such as Java, AADL includes a **package** notion to contain component declarations.

- Packages contain a **public** section and optionally a **private** section.
  - Component declarations are contained in these sections.

- Example:

```
package robot_deployment
public
    with robot_platform, robot_software;

    system Robot_Deployed
    end Robot_Deployed;

    system implementation Robot_Deployed.Basic
        subcomponents
            light_sensor_driver: device robot_software::Light_Sensor;
...
end robot_deployment;
```

Content

OSATE: Open-Source AADL Tool Environment

- Developed at the Software Engineering Institute (SEI) of the Carnegie Mellon University (CMU).
- **Synchronized** textual and graphical editors.
- Eclipse-based: Eclipse Modeling Framework (EMF);
  - Ecore meta-meta-model.
  - Xtext to define textual languages
  - Etc.
- Actively maintained

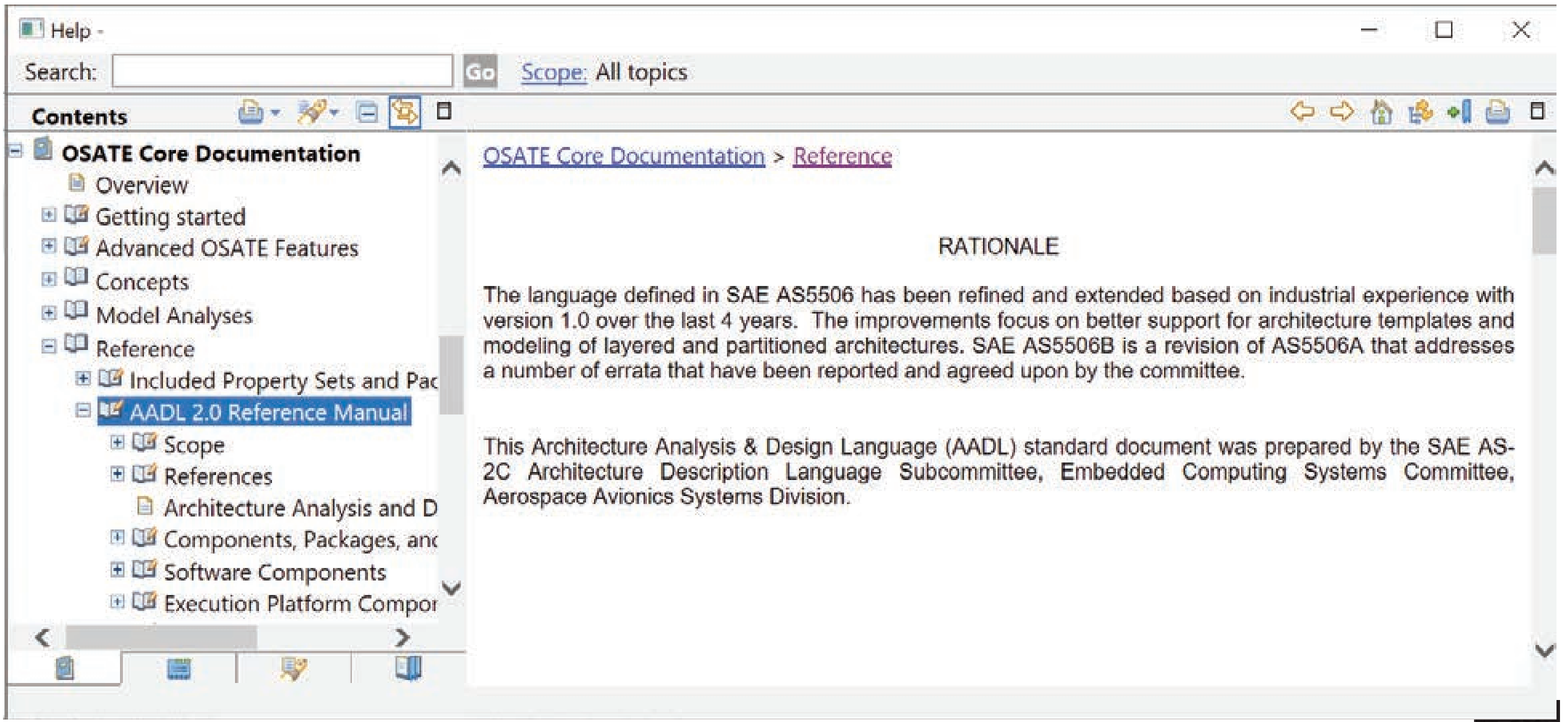
This is the OSATE Open Source AADL Tool Environment.

Version: 2.11.0.vfinal -- Build id: 2022-06-16

This offering is based on technology from the Eclipse Project.
Visit http://osate.org and http://www.eclipse.org

OSATE Documentation

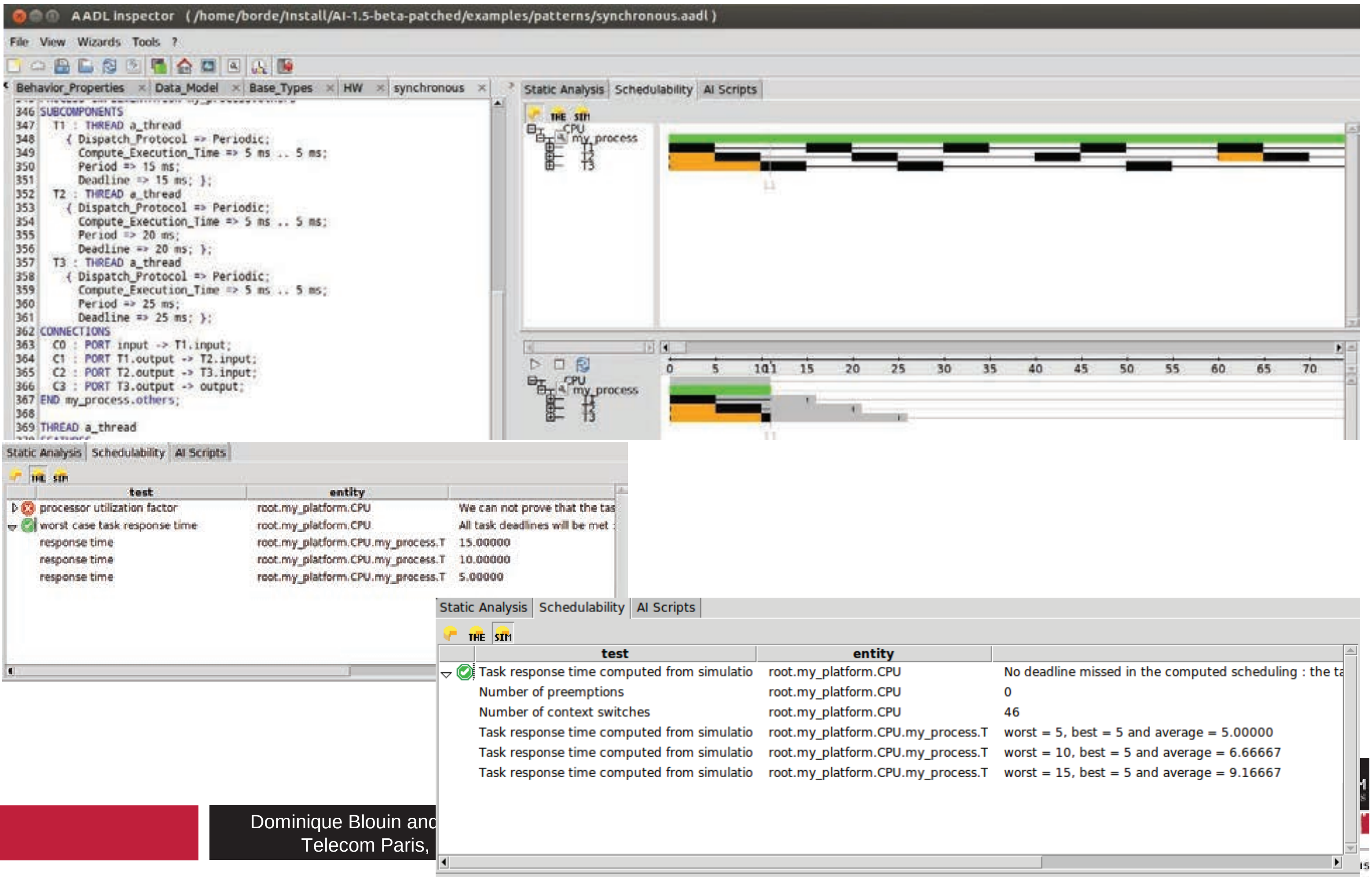
- Click menu *Help>>Help Contents.*
- Navigate to the *OSATE Core Documentation* branch.



Another Analysis Tool

- AADL Inspector: Standalone (not in Eclipse IDE) analysis tool for AADL.
- Developed by Ellidiss Technologies in France
  - Active member of AADL standards committee from the beginning.
  - Provided a **free evaluation license** for this course.
- Integrates other tools from academia research such as Cheddar.
  - Scheduling analyzer and simulator.
- Take as input same AADL textual files (.aadl) as OSATE.

AADL Inspector User Interface Views

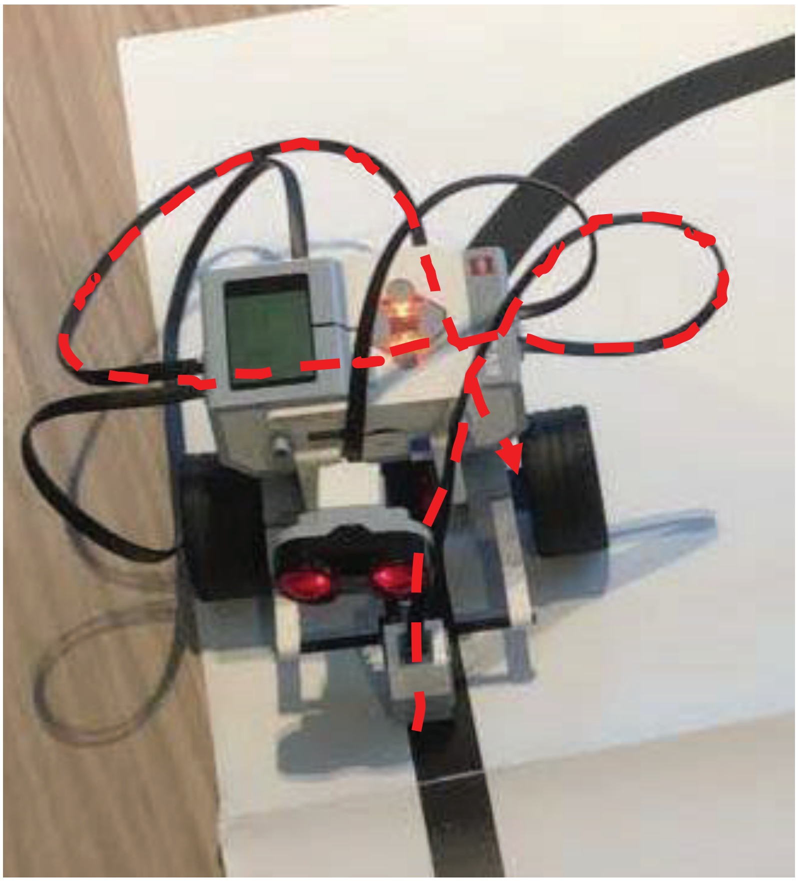


Dominique Blouin and
Telecom Paris,

Content

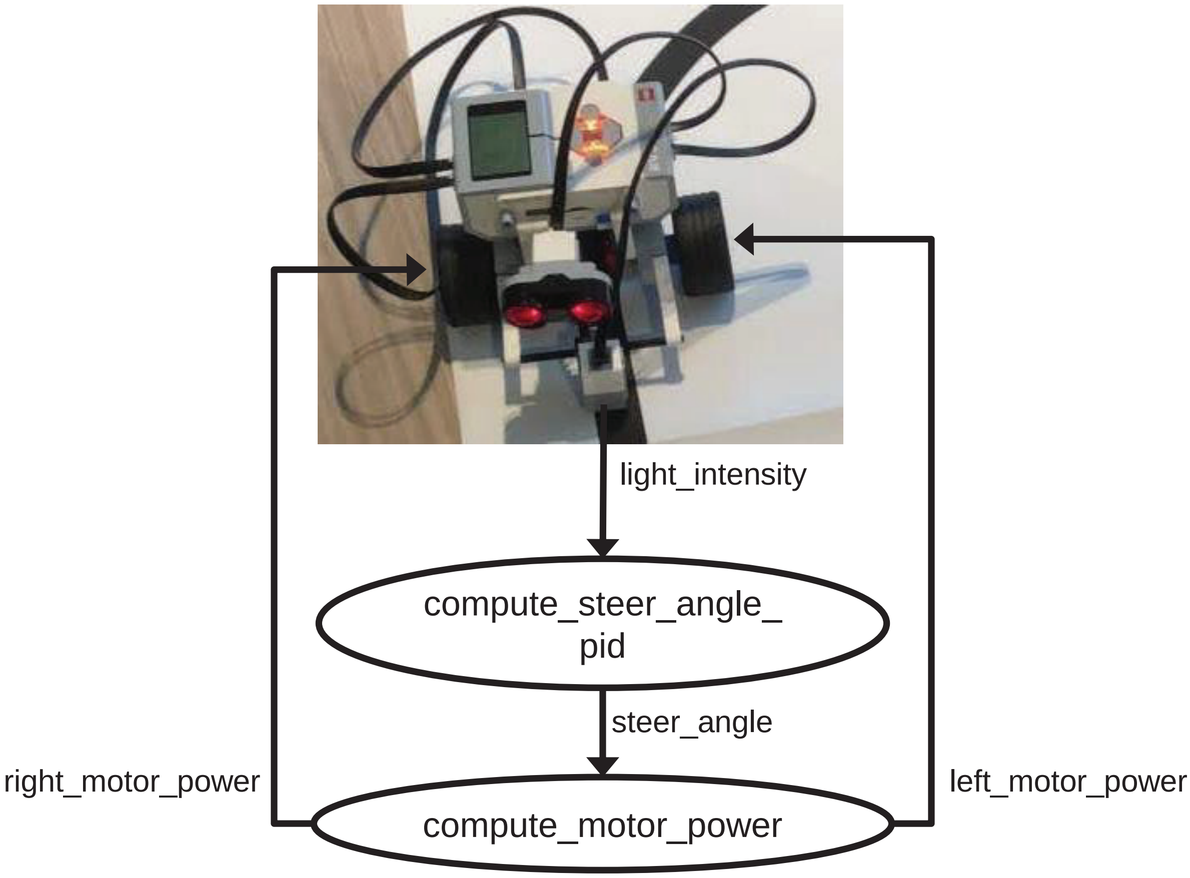
- **Introduction to AADL**
- **Modeling Software Applications**
- **Modeling Execution Platforms**
- **Organization of Declarations**
- **Introduction to OSATE and AADL Inspector**
- **Timing Analysis with AADL Inspector**
- **Model Refinement and Code Synthesis with RAMSES**

Importance of End-to-End Latency



- The **maximum allowed latency** on steering the robot upon a change of light intensity may depend on several parameters:
  - How fast the robot need to go:
    - The faster the robot goes, the lower the latency will need to be.
  - Minimum curvature radius of the path to follow:
    - The smaller the radius is, the lover the latency will need to be.
- Latency is a **primary concern** in designing real-time systems.

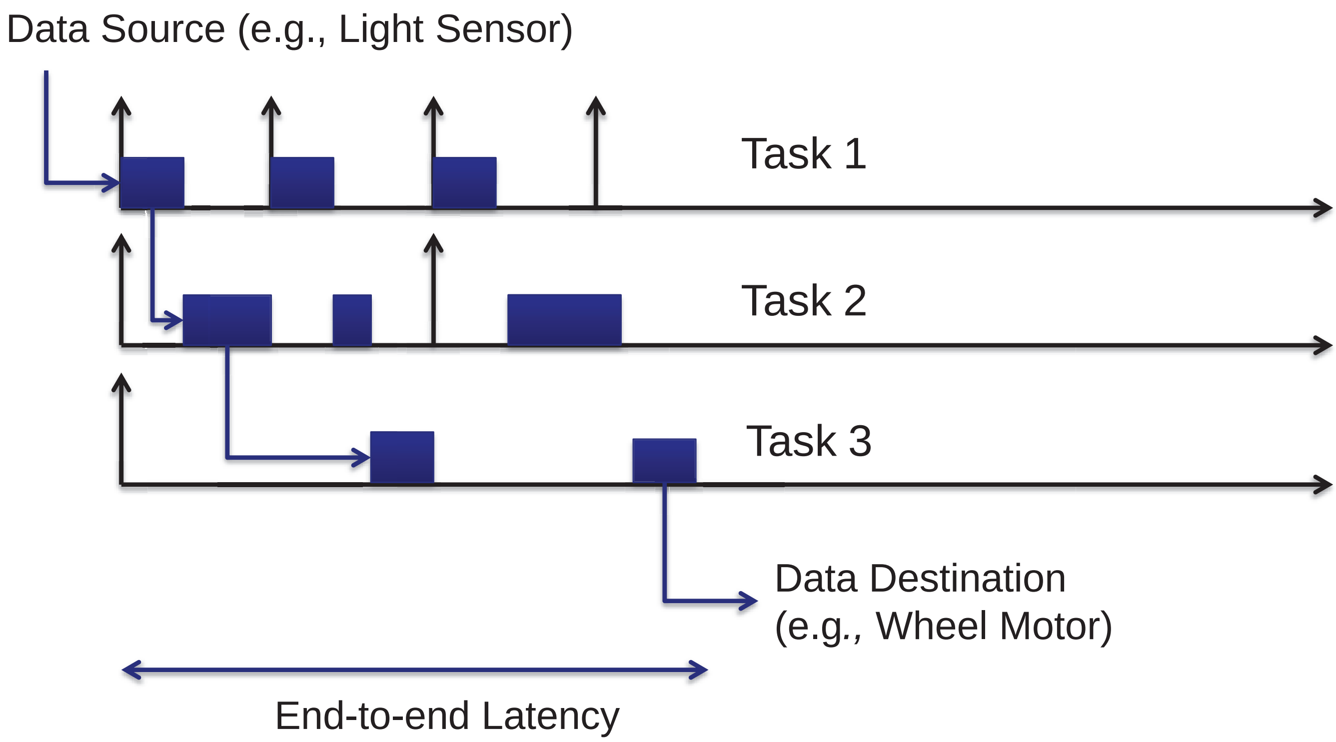
Importance of Response Time



Important Parameters

- **Sampling frequency**: How often do we need to execute the control loop function?
  - E.g., 50Hz (every 20 ms).
- What happens if the **computation time** is greater than the **period**?
  - E.g. 40 ms.
- What will be the impact on the system dynamics?
  - Any performance issues?
- What will be the impacts for users?
  - Any safety issues?
- Besides correct system functions,
  - Data must be available **in time**.
- **Real-time ≠ fast computing**.

Latency Example



Latency Contributors

- The tasks **response time** contributes to latency and jitter.
- What are other contributors?
- **Communications** between tasks:
  - Running on the **same** processor.
  - Or running on **different** processors.
    - Remote communication between tasks.
- Therefore, we need to add to our model information on **communications between components**…

AADL Properties

- A property allows associating a value of a given type to any model element in AADL (any component type, implementation, feature, etc.).
- The AADL standard defines a set of properties for common analyses of embedded systems.
- It is also possible to define user-specific properties declared in **property sets**.
- Properties is a sub-language of the core AADL
- Properties can define constraints on which component they can be applied to (applicability).

Properties Syntax

- A property has a **type**. It can be:
  - Boolean : `aadlboolean`
  - Integer: `aadlinteger`
  - Real: `aadlreal`
  - String: `aadlstring`
  - Enumeration: `enumeration`
  - Component classifier: `classifier` (component, connection, etc.)
  - Reference to a model element: `reference` (component…)
  - A range of values: : `range` …
  - A list of values: : `list of` …
  - A quantity unit : `unit`

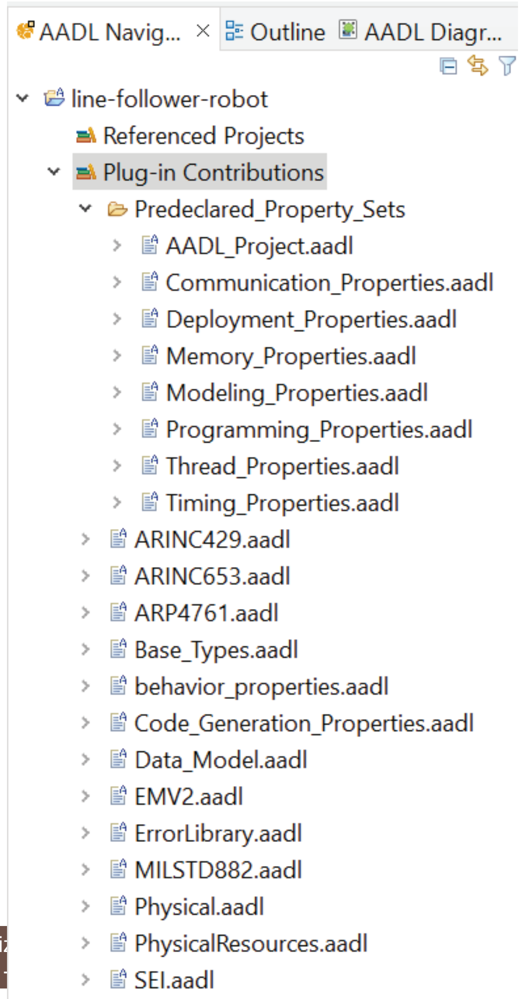
- It is possible to:
  - Define property types **reusing** existing types.
  - Associate a **default value** to a property.
  - Define the **applicability** of a property to component **categories** but also to specific component **types** or **implementations** (classifiers).

Standard Predefined Properties

- The AADL standard defines properties to be used for common analyses related to communication, memory, threading and timing.
  - Those are described in the standard documentation (available under OSATE help).

- Standard properties can be viewed within OSATE for any AADL project under the **Plug-in Contributions** library folder as shown in the screenshot.

- Other visible property sets are also part of the standard but are provided by annexes of AADL such as the Error Model Annex (EMV2).



Dominique Blouin and Anish Bhobe
Telecom Paris, IP Paris

Modeling, Analyzing and Synthesiz
with AADL using RAMSES -

AADL Properties that Influence Latency

- Connection **Timing**:

```
Timing: enumeration (sampled, immediate, delayed) => sampled applies to (port connection);
```

- **Deadline** or **Compute_Execution_Time**
  - Which one is used depends on the connection **timing**.

- **Transmission_Time** to be set on buses

```
Transmission_Time: record (Fixed: Time_Range; PerByte: Time_Range;) applies to (bus,
system, device, processor, memory, virtual bus, virtual processor);
```

- **Latency** used to specify previously known latencies to various model elements or requirement on an end-to-end flow (see later).

```
Latency: Time_Range applies to (flow, connection, virtual bus, bus, processor, virtual
processor, device, system, feature, memory);
```

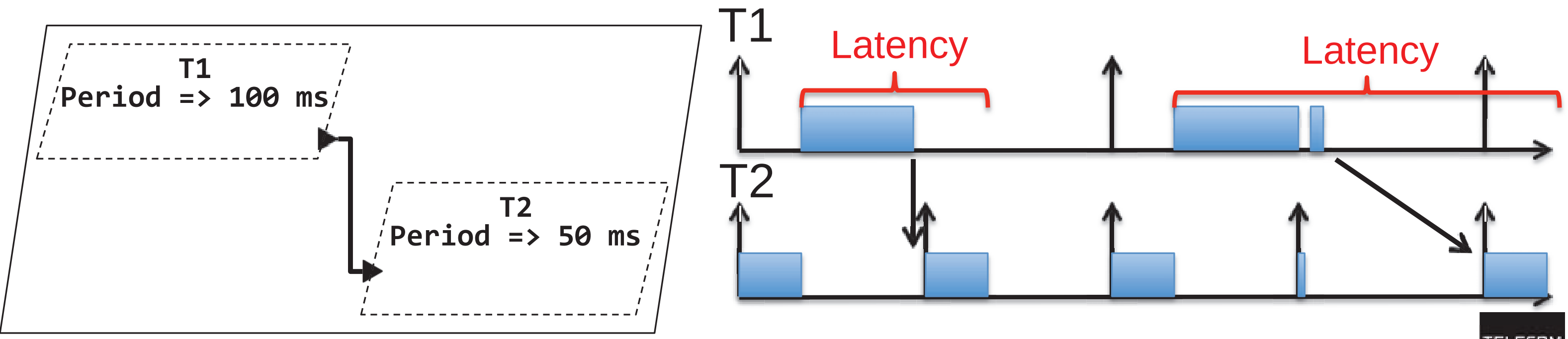
Examples

```
system implementation Exec_Platform.Basic
    subcomponents
        cpu_1: processor Arm.V7 {Scheduling_Protocol => (ARINC653);};
    properties
        Scheduling_Protocol => (RMS) applies to cpu_1;
end Exec_Platform.Basic;

processor Arm
    properties
        Scheduling_Protocol => (Round_Robin_Protocol);
end Arm;

processor implementation Arm.V7
    properties
      Scheduling_Protocol => (POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL);
end Arm.V7;
```

What will be the scheduling protocol?

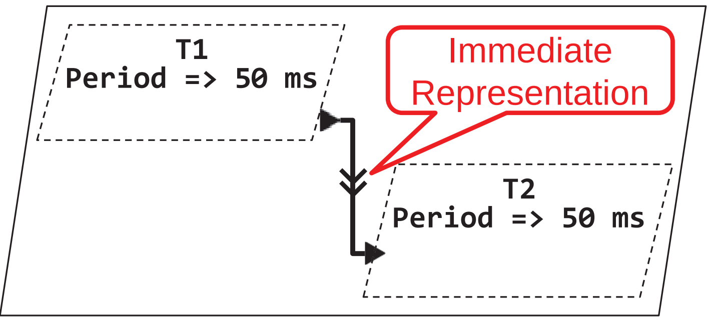
Data Port Connection Timing Property

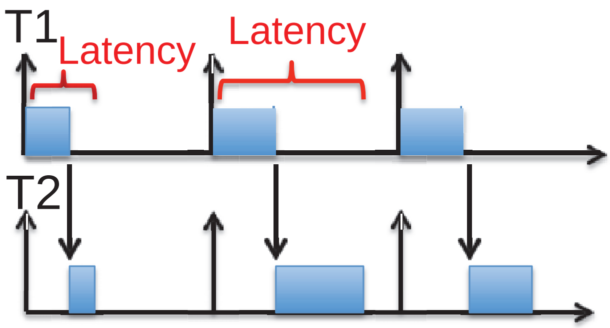
- **Sampled**:
  - Like a shared variable.
  - Thread T1 writes its data at the **end** of its execution and thread T2 reads the data at the **beginning** of its execution.
  - Advantage: simplicity.
  - Disadvantage: **non-deterministic**.



Data Port Connection Timing Property

- **Immediate**: The recipient only starts when the output port of the connected thread has been updated.
  - Advantages:
    - Deterministic.
    - Reduces latency.
  - Disadvantages: Imposes constraints on the scheduler and the thread model:
    - No task dependency cycle.
    - The execution of T1 must precede T2 so the period of T2 should be >= the period of T1.

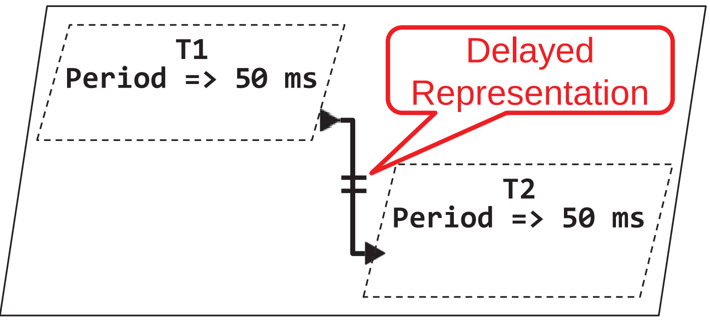


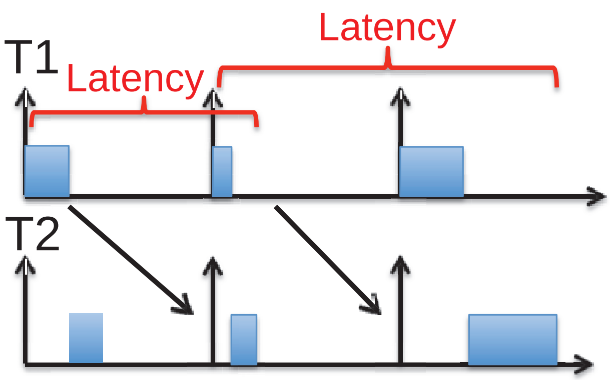


Data Port Connection Timing Property

- **Delayed**: The output port is updated at the **deadline** of its thread.
- The data is processed:
  - At the earliest after a period of T1 and a short execution of T2.
  - At the latest after a period ofT1 and a long execution of T2.

- Advantages:
  - Deterministic and reduces jitter.
- Disadvantages:
  - Increases latency.





Deployment Properties

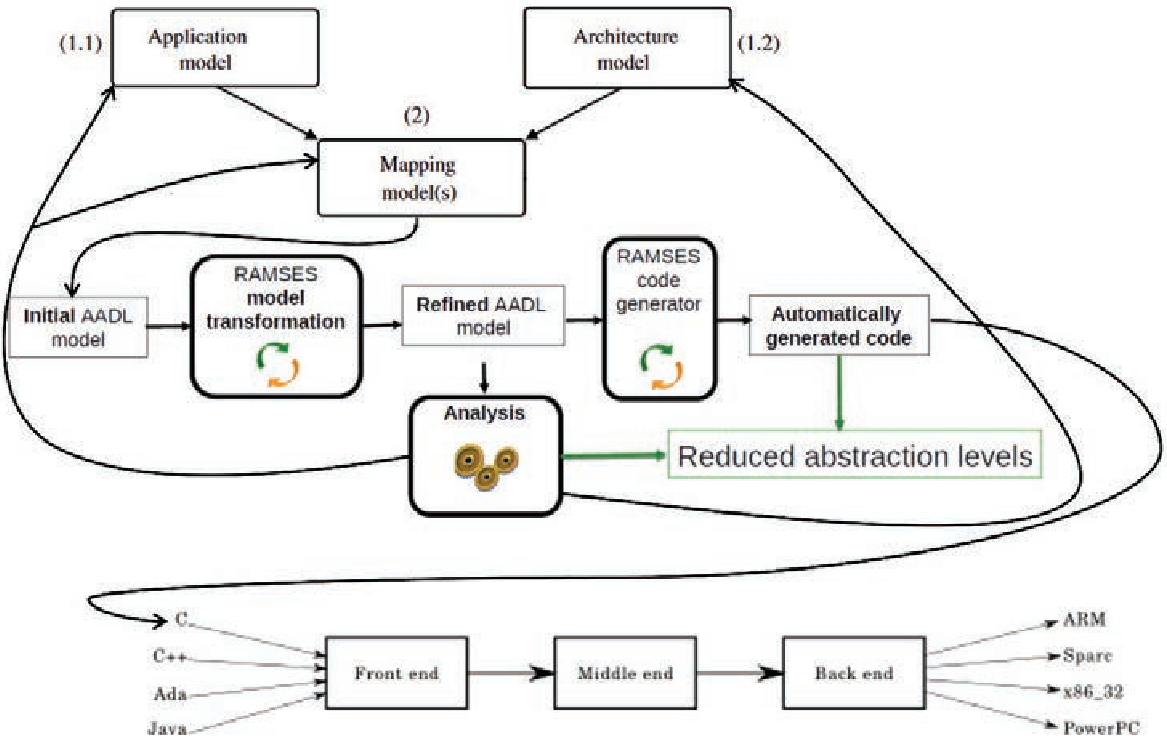
```
system implementation Synchronous.Others
    subcomponents
        my_platform : processor CPU;
        my_process : process my_process.impl;
    properties
         Actual_Processor_Binding => (reference(my_platform)) applies to my_process;

end Synchronous.Others;

-- Binding to nested subcomponents
system implementation Line_Follower_Robot.Basic
    subcomponents
        soft_app: system robot_software::Line_Follower_Application.Basic;
        hard_platform: system robot_platform::Robot_Hardware.Basic;
    properties

        Actual_Processor_Binding =>
(reference(hard_platform.exec_platform.basic_processor)) applies to
soft_app.line_following_controler;

        Actual_Memory_Binding => (reference(hard_platform.exec_platform.ram)) applies
to soft_app.line_following_controler;

end Line_Follower_Robot.Basic;
```

Content

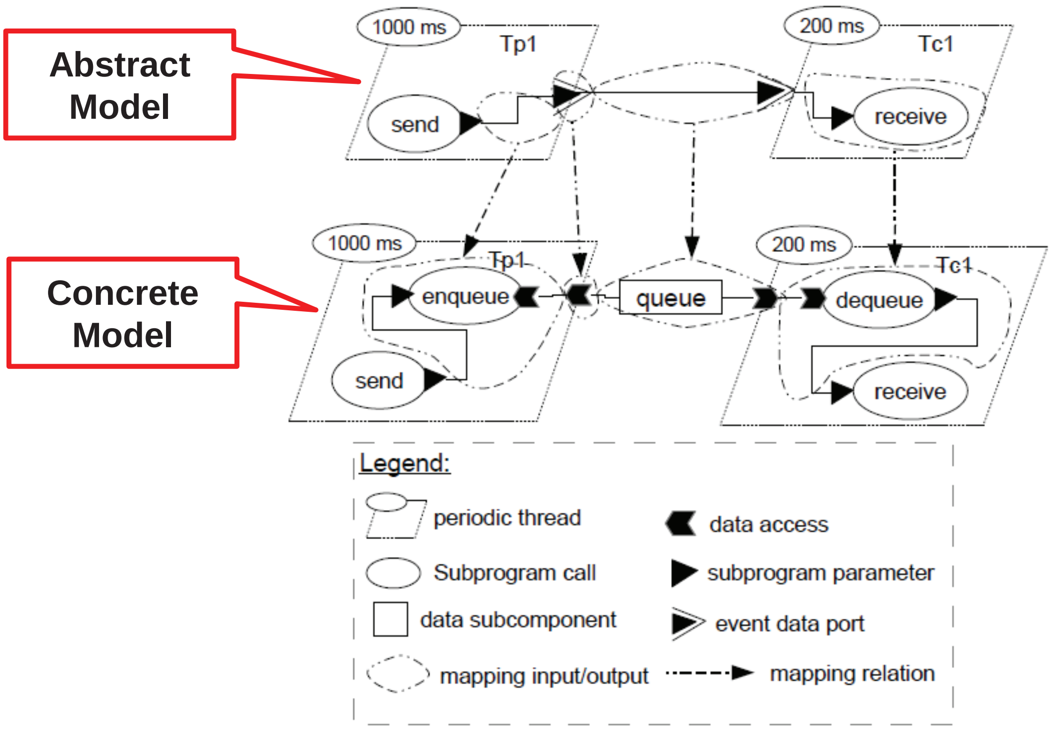
- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

Model Refinement and Synthesis (Code Generation)

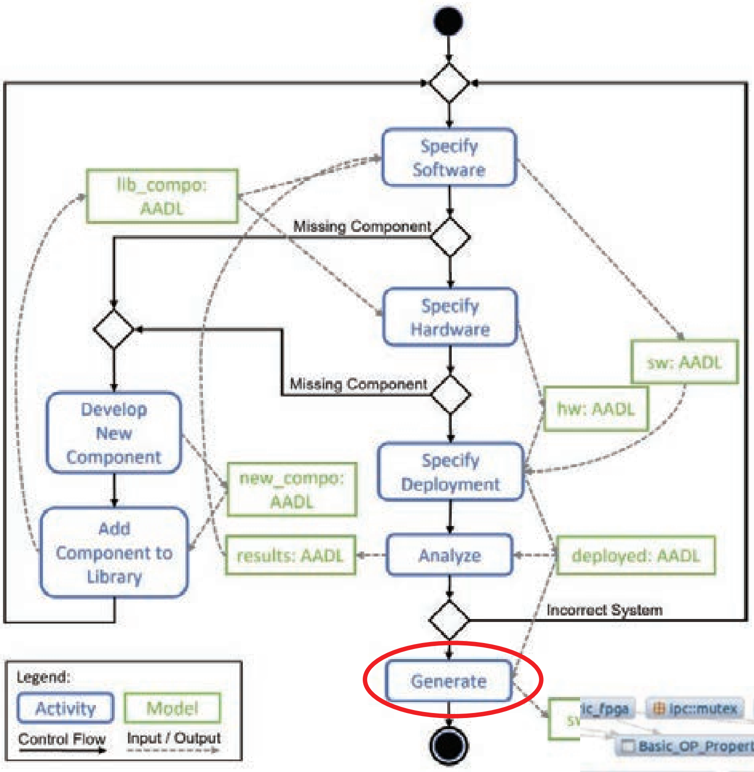
- **RAMSES**: Refinement of AADL Models for the Synthesis of Embedded Systems
  - https://mem4csd.telecom-paristech.fr/blog/index.php/ramses/

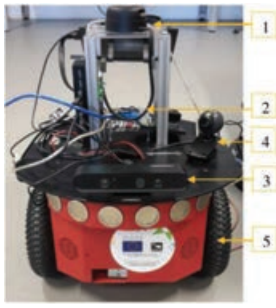


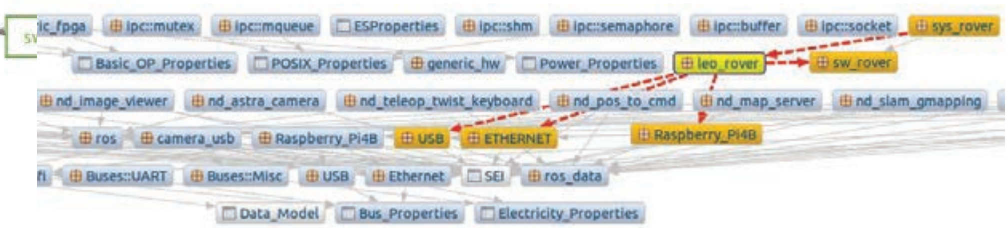
Example of a RAMSES Refinement Rule



Ongoing Project: RAMSES-ROS Extension for Complex Robotics Systems





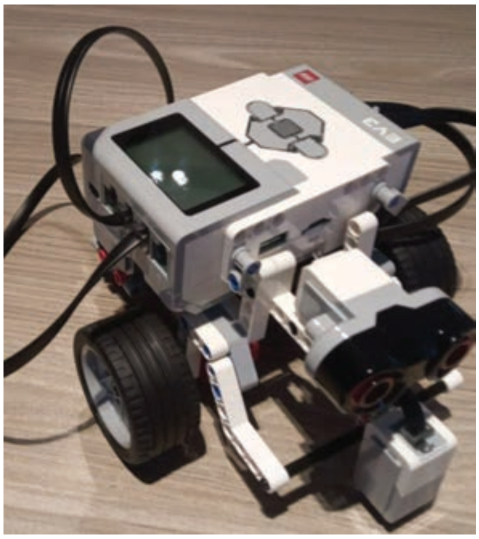


- AADL component library

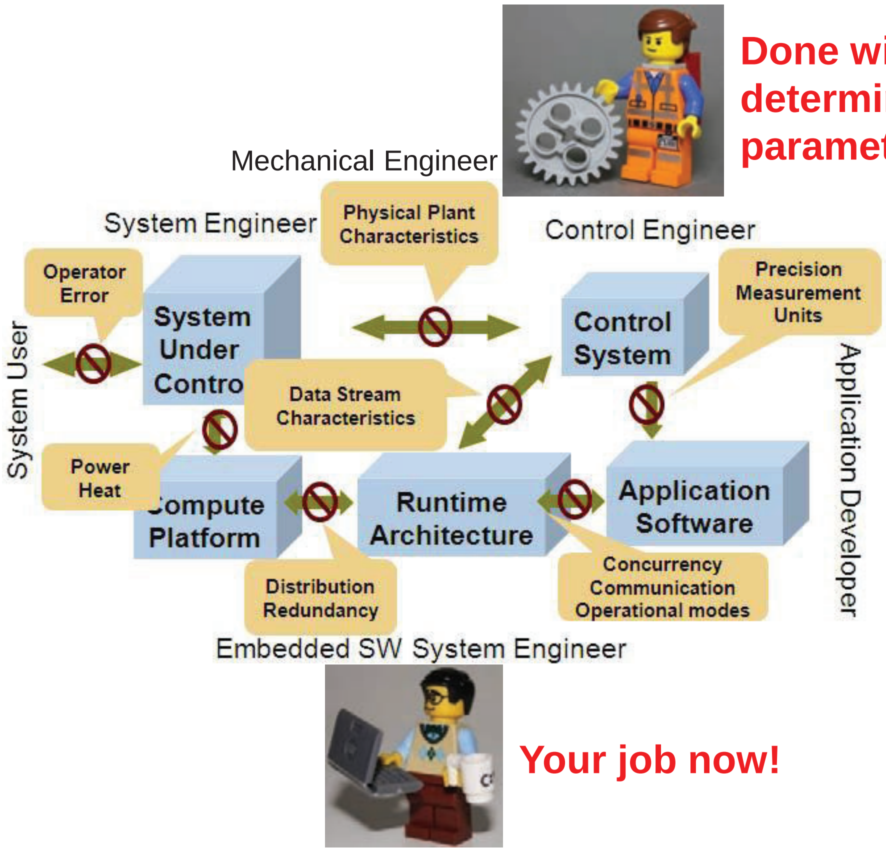
Exercise: Model and Synthesize a Cruise-Control System (Hands-On)

- Objectives:
  - Learn how to model a simple embedded system in AADL
  - Learn how to analyze the model and modify the design to ensure the program execution is not jeopardized due to performance limitations.
  - Synthesize the C code of the system from this model.

- Follow the instructions on our website:
  - https://mem4csd.telecom-paristech.fr/blog/index.php/training-schools/cps-iot-summer-school-2024/



Two Engineering Domains



Done with Modelica to determine PID parameters.

Your job now!

Making Teams

- 10 robots ➔ How many students per team?
- **Linux** required for the exercise.
  - GCC must be installed.
- Define the teams according to the available robots and Linux computers.

Smart CPS : Ensuring Trustworthiness in Autonomous Decisions through Formal Methods

Dr. Samir Ouchani
Research Director, CESI Lineact, Aix-en-Provence

CESI LINEACT

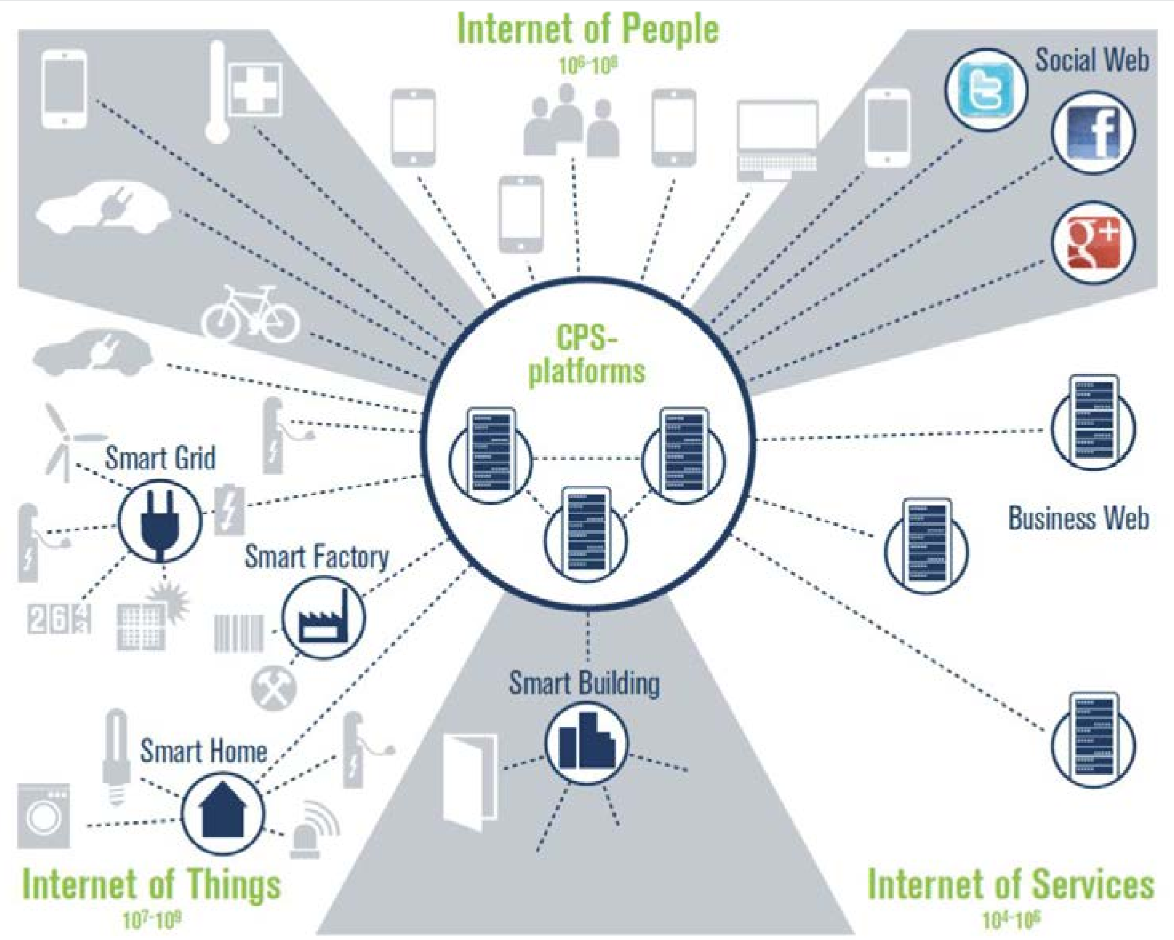
The Summer School on Cyber Physical Systems and Internet of Things (SS-CPS&IoT 2024)

Budva, Montenegro, June 11-14, 2024

Content

Research Statement

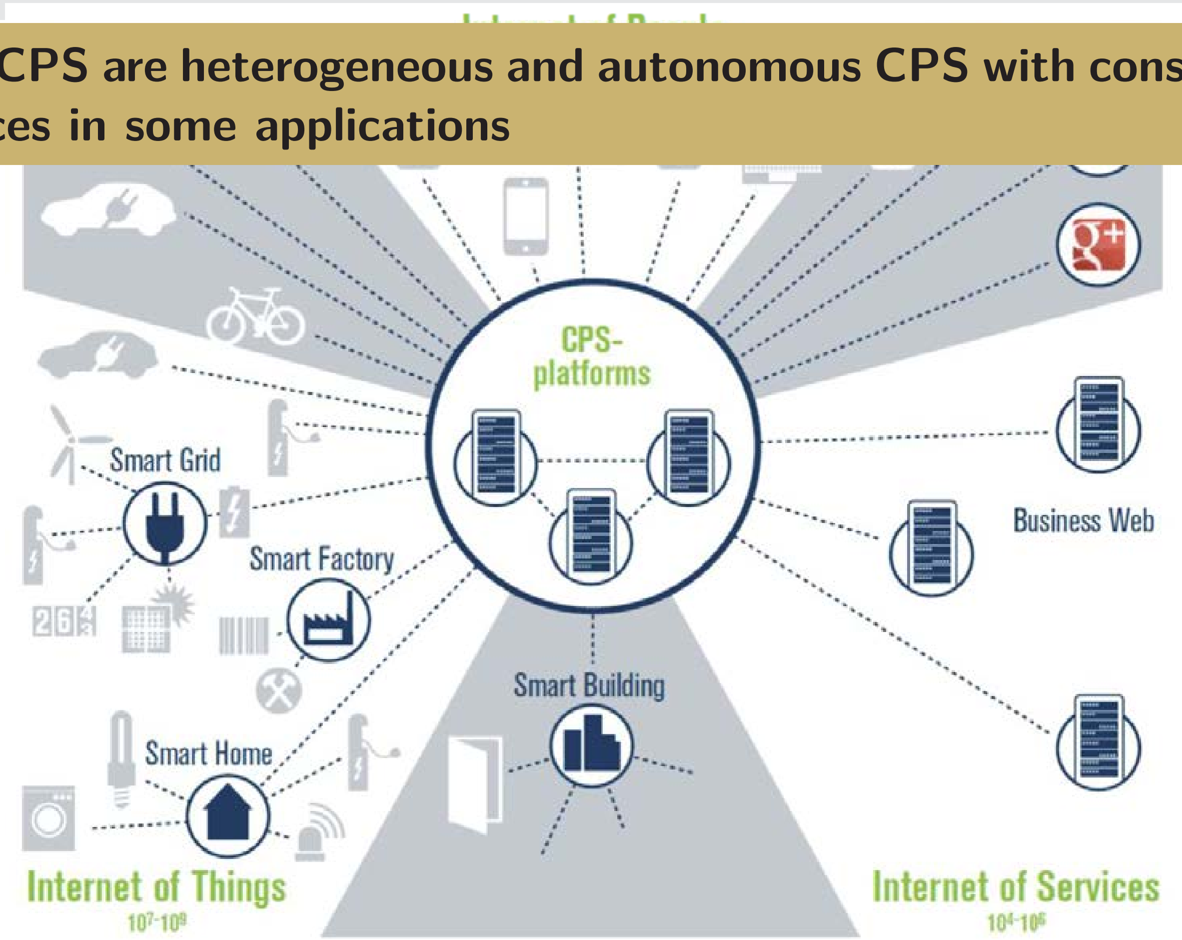
Context



Research Statement

Context

Smart CPS are heterogeneous and autonomous CPS with constrained resources in some applications

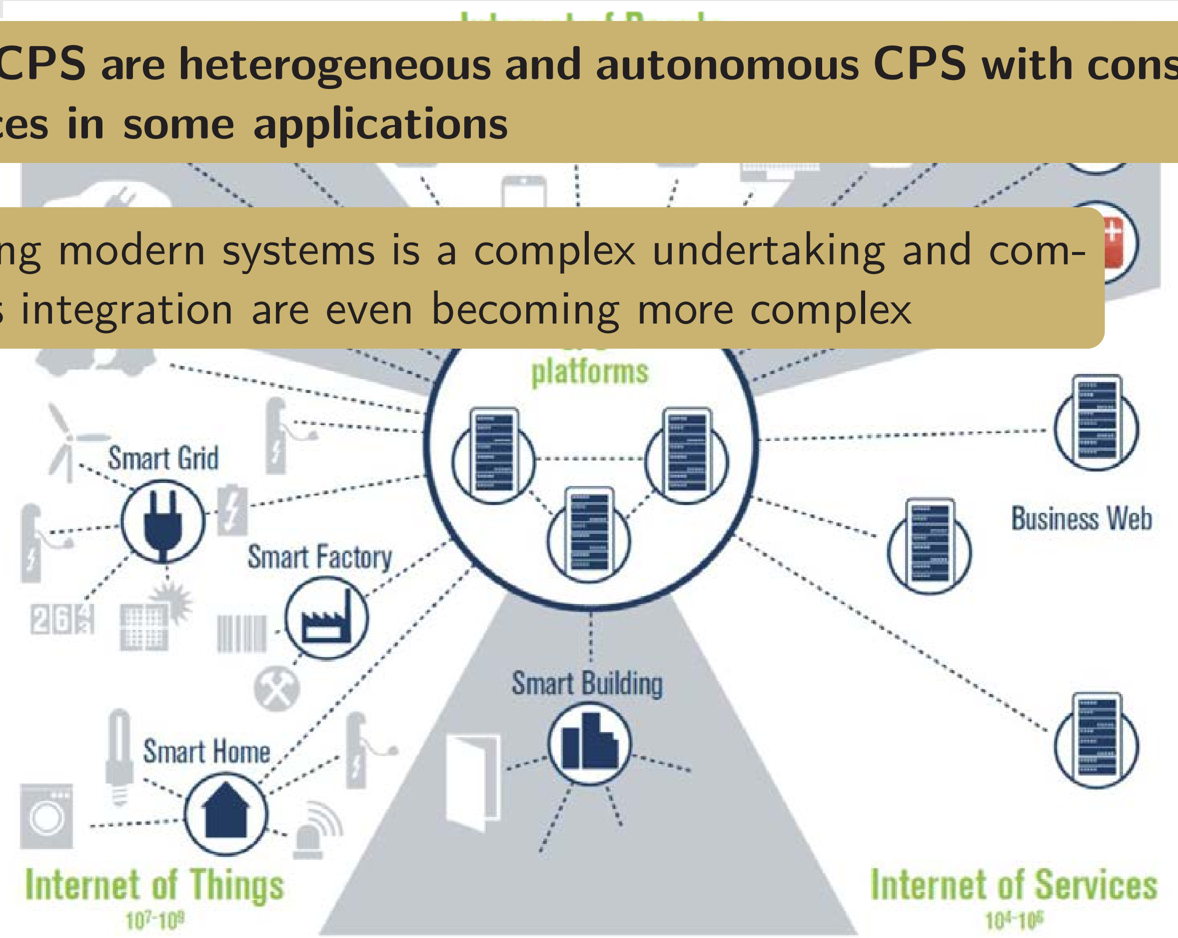


Research Statement

Context

Smart CPS are heterogeneous and autonomous CPS with constrained resources in some applications

Designing modern systems is a complex undertaking and components integration are even becoming more complex



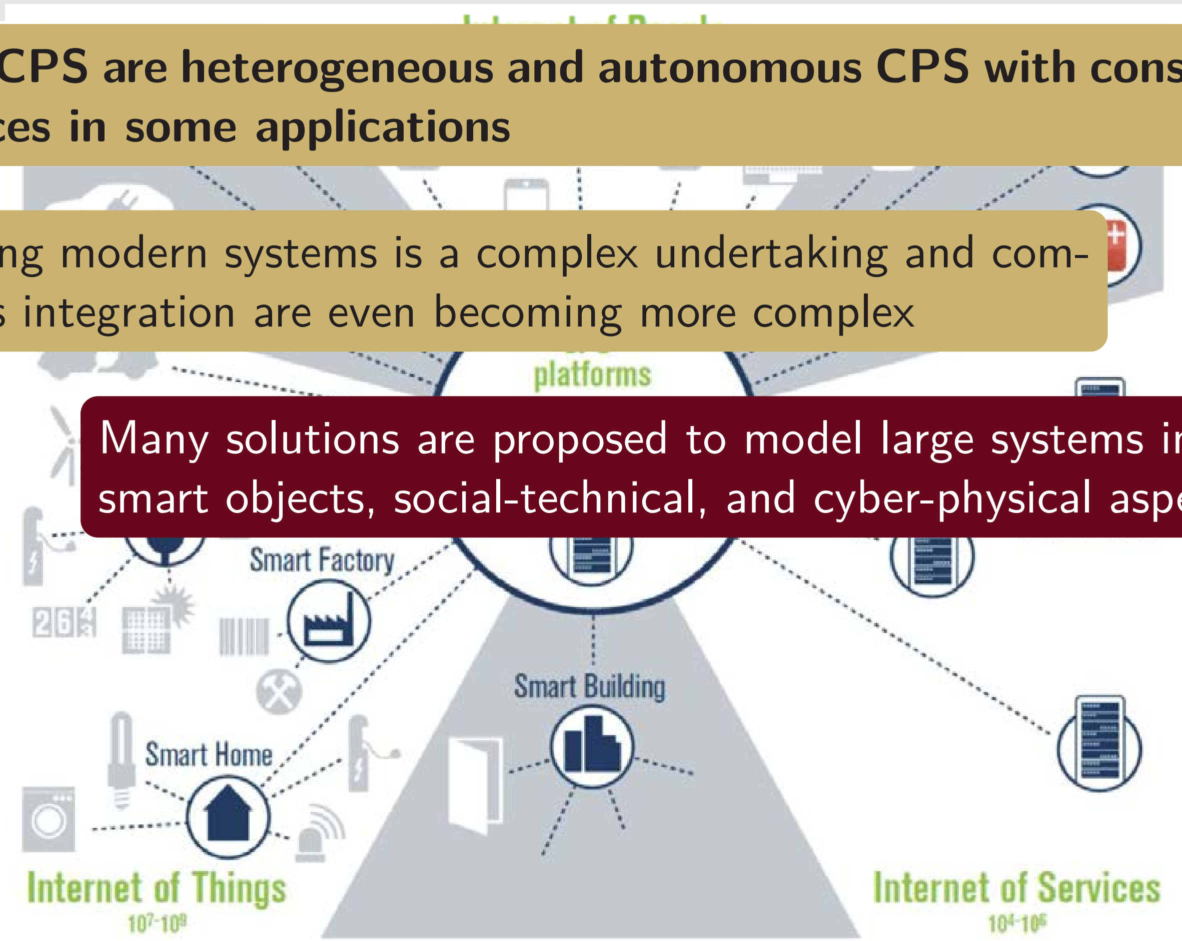
Research Statement

Context

Smart CPS are heterogeneous and autonomous CPS with constrained resources in some applications

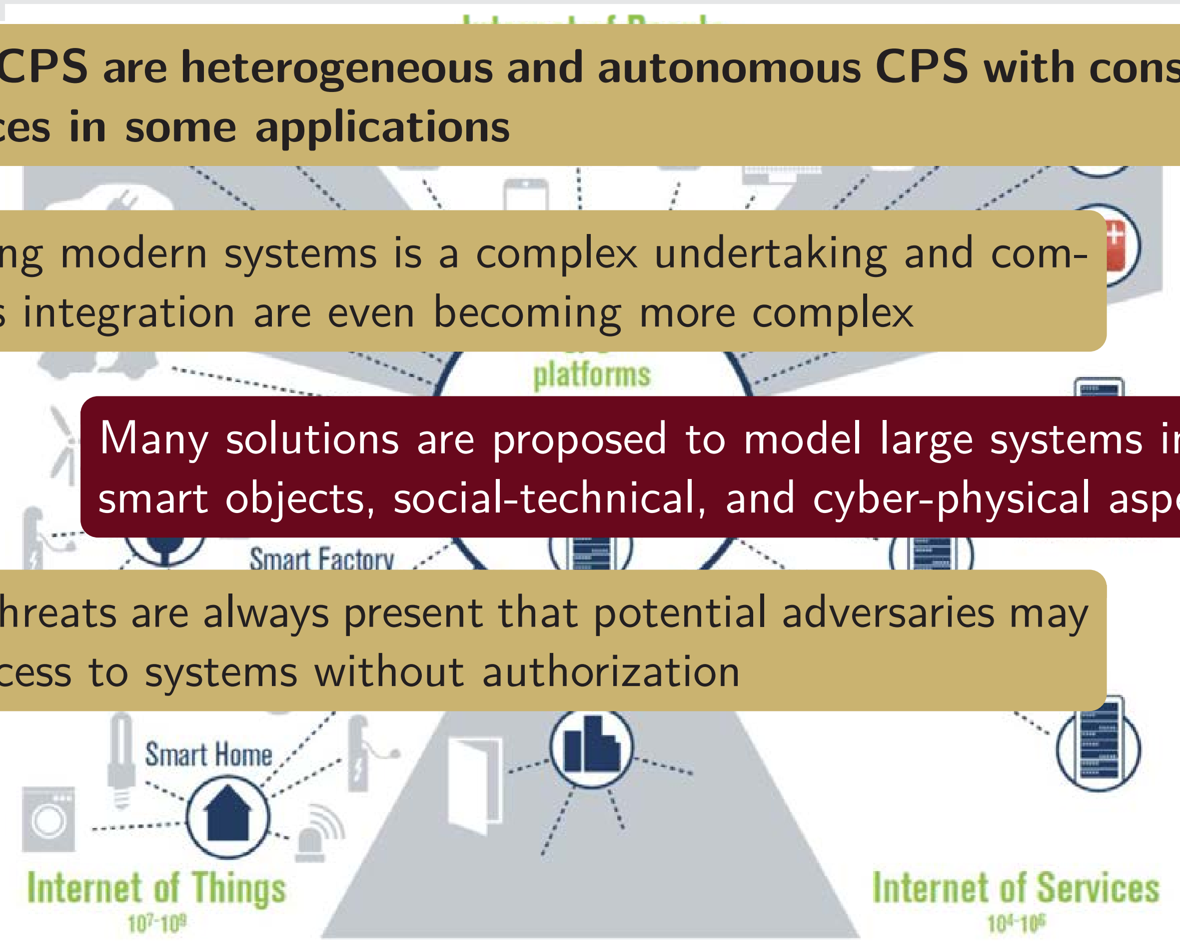
Designing modern systems is a complex undertaking and components integration are even becoming more complex

Many solutions are proposed to model large systems including smart objects, social-technical, and cyber-physical aspects



Research Statement

Context



Smart CPS are heterogeneous and autonomous CPS with constrained resources in some applications

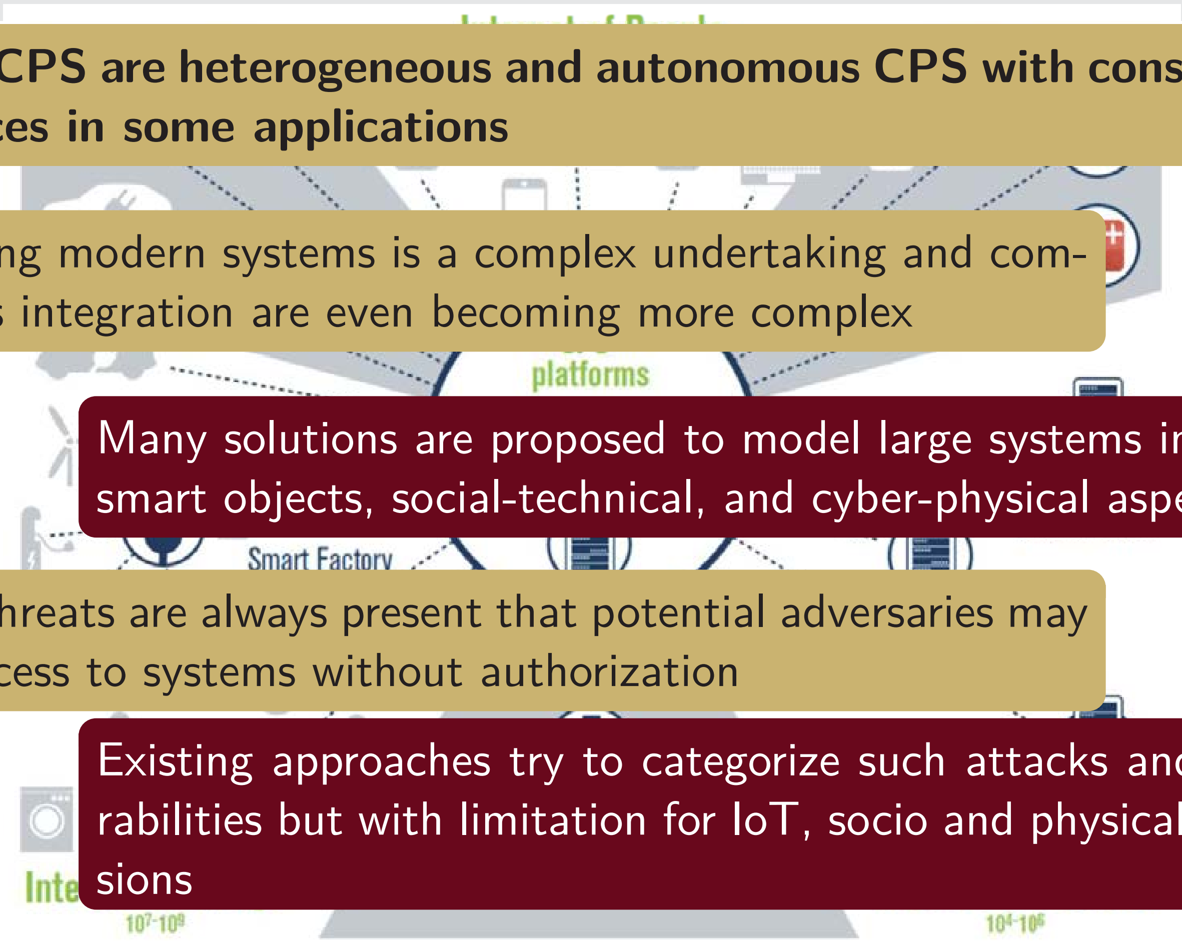
Designing modern systems is a complex undertaking and components integration are even becoming more complex

Many solutions are proposed to model large systems including smart objects, social-technical, and cyber-physical aspects

Major threats are always present that potential adversaries may gain access to systems without authorization

Research Statement

Context



Smart CPS are heterogeneous and autonomous CPS with constrained resources in some applications

Designing modern systems is a complex undertaking and components integration are even becoming more complex

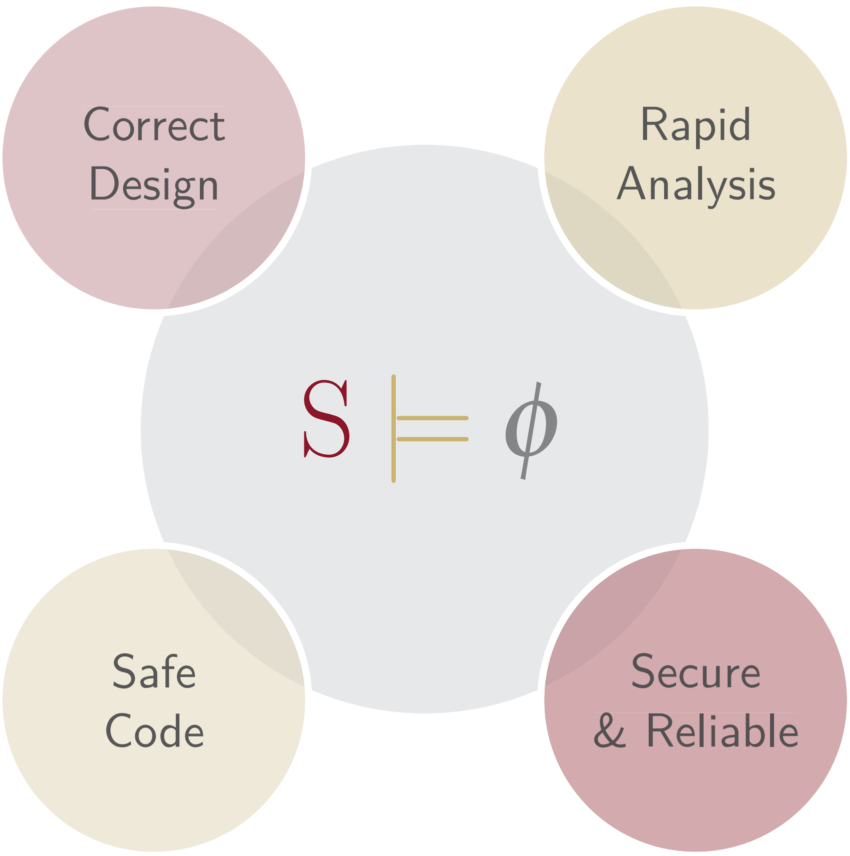
Many solutions are proposed to model large systems including smart objects, social-technical, and cyber-physical aspects

Major threats are always present that potential adversaries may gain access to systems without authorization

Existing approaches try to categorize such attacks and vulnerabilities but with limitation for IoT, socio and physical dimensions

Research Statement

Challenges



SCPS Semantics

System Modeling Language

Model-based Security Project

1. Abdelhakim, Baouya, Otmane Ait Mohamed, Djamal Bennouar, and **Samir, Ouchani**. A formal approach for maintainability and availability assessment using probabilistic model checking.
In *Modelling and Implementation of Complex Systems*, pages 295–309. Springer International Publishing, Cham, 2016.

2. Baouya, Abdelhakim, Djamal Bennouar, Otmane Ait Mohamed, and **Ouchani, Samir**. On the probabilistic verification of time constrained sysml state machines.
In *International Conference on Intelligent Software Methodologies, Tools, and Techniques*, pages 425–441. Springer International Publishing, 2015.

3. **Ouchani, Samir** and Mourad Debbabi. Specification, verification, and quantification of security in model-based systems.
Computing, 97(7) :691–711, 2015.

SCPS Semantics

System Modeling Language - Formal Definition

SysML activity diagrams are a graph-based representation where vertices are nodes that control flows in edges

Definition

An SysML activity diagram is a tuple $\mathrm{A} = \langle \mathscr{N}, \mathscr{E}, \mathscr{G}, \mathrm{Grd}, \mathrm{Prob} \rangle$, where :

- $\mathscr{N}$ is a finite set of activity nodes such as $\mathrm{a_i}$ and $\mathrm{a_f}$ denote the initial and the final nodes, respectively ;
- $\mathscr{E}$ is a finite set of activity edges,
- $\mathscr{G}$ is the set of guards,
- $\mathrm{Grd} : \mathscr{E} \mapsto \mathscr{G}$ is a partial function that returns a guard for an edge, and
- $\mathrm{Prob} : \mathscr{N} \mapsto \mathit{Dist}(\mathscr{N})$ is a partial probabilistic function that assigns for each node a convex discrete probability distribution $\mu \in \mathit{Dist}(\mathscr{N})$ over its output transitions.

SCPS Semantics

System Modeling Language - Systax

- Activity Calculus helps to formalize SysML activity diagrams
- $\mathscr{A}[\mathscr{N}]$ specifies $\mathscr{N}$ as a sub term of $\mathscr{A}$
- $|\mathscr{A}|$ denote a term $\mathscr{A}$ without tokens
- We denote $\mathscr{A}[\mathrm{a} \uparrow \mathscr{A}']$ by $\mathscr{A} \uparrow_{\mathrm{a}} \mathscr{A}'$

$$
\begin{array}{lcl}
\mathscr{A} & ::= & \varepsilon \mid \mathrm{l}\imath\mathrm{n}\mathscr{N} \\
\mathscr{N} & ::= & \mathscr{N}\mathrm{n} \mid \mathrm{lM(x,y)}\mathscr{N} \mid \mathrm{lJ(x,y)}\mathscr{N} \mid \\
 & & \mathrm{lF}(\mathscr{N},\mathscr{N}) \mid \mathrm{la}\uparrow\mathscr{A}\mathrm{n}\mathscr{N} \mid \mathrm{lD}((\mathrm{p},\mathrm{g},\mathscr{N}),(1-\mathrm{p},\neg\mathrm{g},\mathscr{N})) \mid \\
 & & \mathrm{l}\otimes \mid \mathrm{l}\odot \mid \mathrm{l}
\end{array}
$$

Smart CPS

SCPS Semantics

System Modeling Language - Semantics

- The execution of SysML activity diagrams is based on token's flow.
- We use structural operational semantics to formally describe how the computation steps of AC atomic terms take place

INIT-1	$l\imath\mathscr{N} \xrightarrow{l} l\imath\mathscr{N}$
BH-1	$\dfrac{\mathscr{A} = l'\imath\mathscr{N}' \qquad \forall n > 0}{la \uparrow \mathscr{A} n \mathscr{N} \xrightarrow{l} la \uparrow l'\imath\mathscr{N}' n-1 \mathscr{N}}$
BH-2	$\dfrac{\mathscr{A}[l'\odot] \xrightarrow{l'} \lvert\mathscr{A}\rvert \qquad \forall n > 0}{la \uparrow \mathscr{A} n \mathscr{N} \xrightarrow{l'} la \uparrow \mathscr{A} n \mathscr{N}}$
PDEC-1	$lD((p,g,\mathscr{N}_1),(1-p,\neg g,\mathscr{N}_2))m \xrightarrow{l}_p lD((p,g,\mathscr{N}_1),(1-p,\neg g,\mathscr{N}_2))m-1 \ \forall m > 0$
ACTIVITY	$\dfrac{\mathscr{N} \xrightarrow{\alpha}_p \mathscr{N}'}{\mathscr{A}[\mathscr{N}] \xrightarrow{\alpha}_p \mathscr{A}[\mathscr{N}']}$

SCPS Semantics

System Modeling Language - Semantics

- We define Σ as the set of non-empty actions labeling the transitions
- $\alpha \in \Sigma$ is the label of the executing active node
- A transition can be $\mathscr{A} \xrightarrow{\alpha}_{\mathrm{p}} \mathscr{A}'$ be $\mathscr{A} \xrightarrow{\alpha} \mathscr{A}'$

Definition (NuAC-PA)

A probabilistic automata of a NuAC term $\mathscr{A}$ is the tuple $\mathrm{M}_{\mathscr{A}} = (\bar{\mathrm{s}},\ \mathrm{L},\ \mathrm{S},\ \Sigma^{\mathrm{o}},\ \delta)$, where :

- $\bar{\mathrm{s}}$ is an initial state, such that $\mathrm{L}(\bar{\mathrm{s}}) = \{\mathrm{l}\imath\mathscr{N}\}$
- $\mathrm{L} : \mathrm{S} \to 2^{\mathscr{L}}$ is a labeling function where : $\mathscr{L} : \mathscr{L} \to \{\top, \bot\}$
- S is a finite set of states reachable from $\bar{\mathrm{s}}$
- Σ^{o} is a finite set of actions corresponding to labels in $\mathscr{A}$
- $\delta : \mathrm{S} \times \Sigma^{\mathrm{o}} \to \ \mathrm{Dist}(\mathrm{S})$ is a partial probabilistic transition function

SCPS Semantics

Algebraic specification

IoT-based Security Project

1. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. Towards a reliable smart city through formal verification and network analysis.
   Computer Communications, 180 :171–187, 2021.

2. Abdelhakim Baouya, Otmane Ait Mohamed, Djamal Bennouar, and **Samir Ouchani**. Safety analysis of train control system based on model-driven design methodology.
   Computers in Industry, 105 :1–16, 2019.

3. **Ouchani, Samir**. Towards a security reinforcement mechanism for social cyber-physical systems.
   In *The Third International Conference on Smart Applications and Data Analysis for Smart Cyber-Physical Systems (Invited Paper)*, page 14. LNCS, Springer, 2020.

4. **Ouchani, Samir**. A security policy hardening framework for socio-cyber-physical systems.
   Journal of Systems Architecture, 119 :102259, 2021.

SCPS Semantics

Algebraic specification

A system S is the tuple $\mathrm{Obj}, \mathrm{Srv}, \mathrm{Act}, \mathrm{Env}, \mathrm{Prot}$:

- The connected objects (Obj), and an object can be physical as digital with specific abilities : container, lockable, movable or/and destroyable.
- The client-server applications and services (Srv). A service Srv ensures a client-server based architecture : client applications, computation servers and web services.
- The social actors (Act),where an actor can be human being or smart robot agents
- The environment (Env) encloses all entities
- The communication protocols (Prot) that ensure the interaction and the communication between the different types of entities

SCPS Semantics

Algebraic specification - Environment

Env can be any human body or other natural species, or even a physical space that hosts objects.

Environment

Env is a tuple $\mathrm{E}, \mathrm{L}, \mathrm{O_E}, \mathrm{Actuator_E}$, where :

- E is a finite set of environments denoted by e, e', etc.
- L is a finite set of locations (l, l', etc.).
- $\mathrm{O_E}$ is a finite set of physical objects of type container.
- $\mathrm{Actuator_E} : \mathrm{O_E} \times \mathrm{O_E} \rightarrow 2^{\mathrm{O}}$ returns the set of objects linking containers by physical objects (e.g. doors connecting two rooms).

SCPS Semantics

Algebraic specification - Interaction Protocol

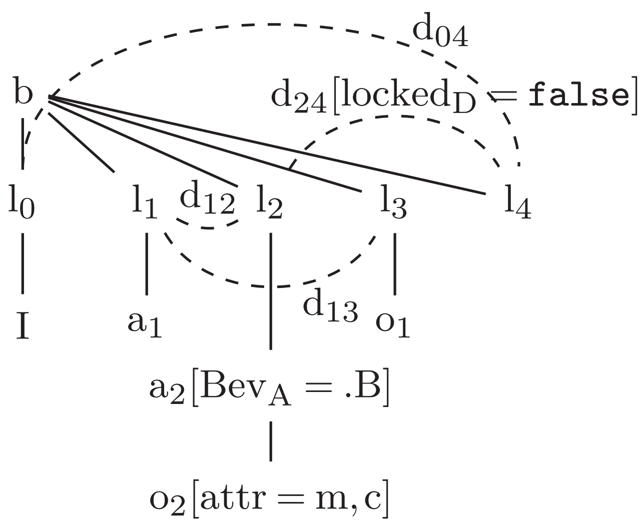
Prot orchestrates the communication between entities.

- Prot is a tuple $\mathrm{Prot}_{h,o}, \mathrm{Prot}_{o,o}, \mathrm{Prot}_{o,s}$ where $\mathrm{Prot}_{h,o}$ ensures the communications between social actors and the objects, $\mathrm{Prot}_{o,o}$ between objects, $\mathrm{Prot}_{o,s}$ between objects and services on servers
- A state $\mathrm{S} = \mathrm{S}_\mathrm{O}, \mathrm{S}_\mathrm{V}, \mathrm{S}_\mathrm{A}, \mathrm{S}_\mathrm{E}$ is an instance of $\mathrm{Obj}, \mathrm{Srv}, \mathrm{Act}, \mathrm{Env}$ composed from states of objects, services, actors, and the environment
- The transitions between states are denoted by $\mathrm{S} \, {}^{\ell,\mathrm{c},\mathrm{p}} \, \mathrm{S}'$, ℓ names the action to be executed with a cost c and a probability p

SCPS Semantics

Algebraic specification - Interaction Protocol

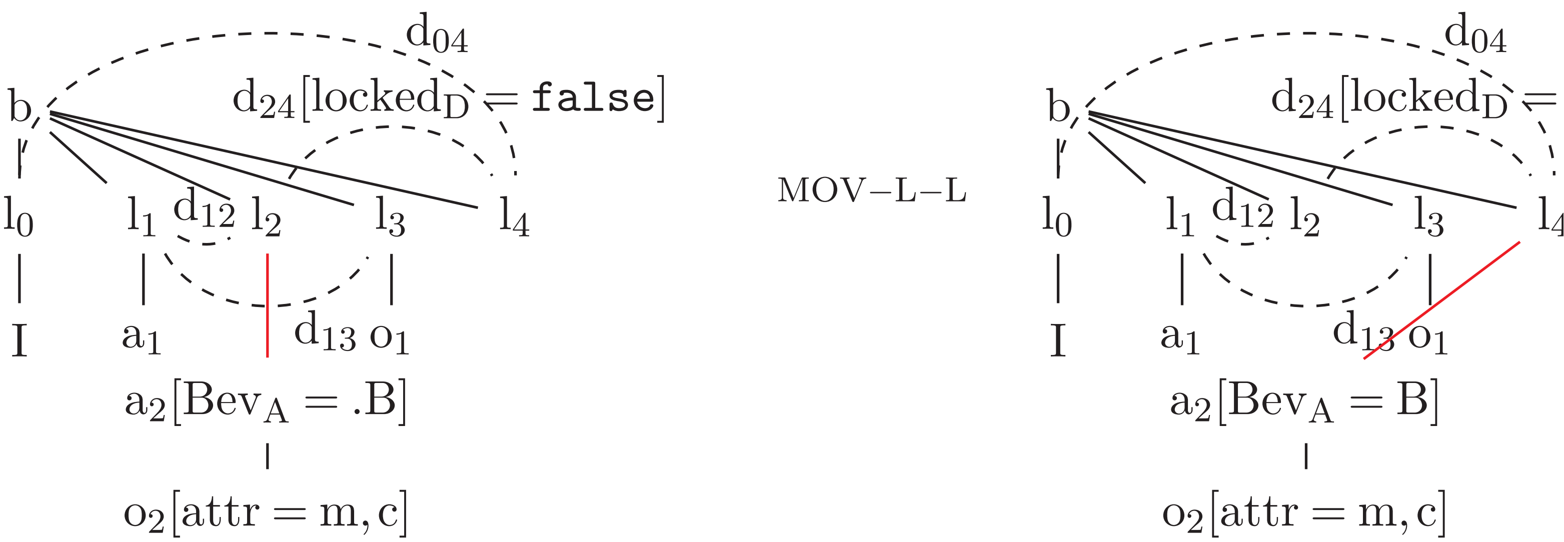
Example of a state



- A state is represented as a labelled multi-graph
- One initial vertex represents the name of the system
- Nodes are location, actors and objects
- Edges show the relation between the entities

Interaction Protocol

- This execution rule shows moving a_2 from l_2 to l_4 (MOV-L-L) : $\text{moving}_{a_2}(d_{24}, l_2, l_4), c, p$



Smart CPS

Subjects

1. Security and reliability
2. Data based approaches for formal methods techniques

Main contributions

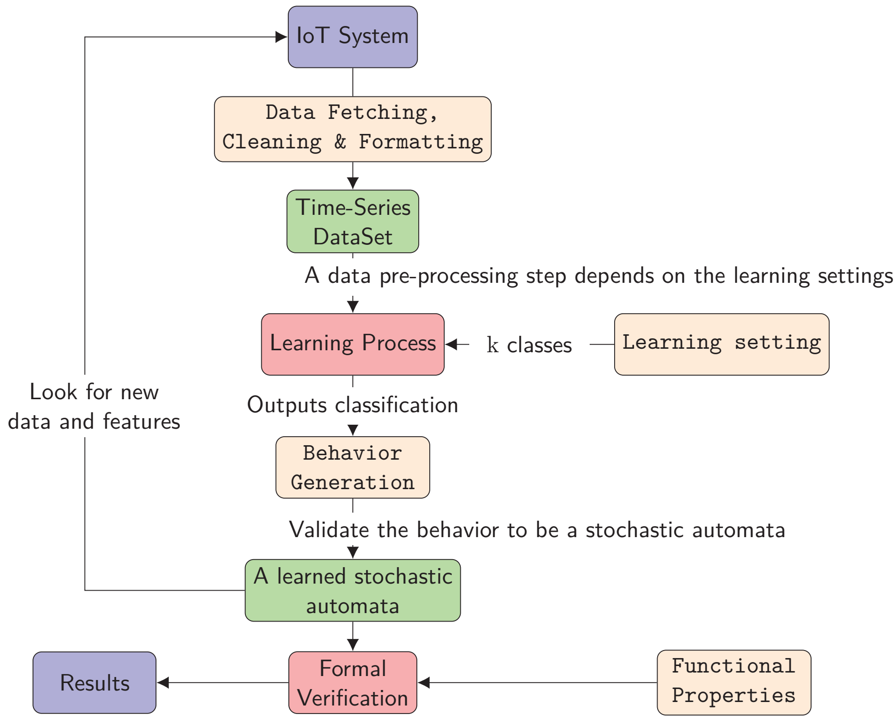
- Specifying and verifying reliability for deployment
- Assessing security in Smart CPS

Applications

- Smart environments : Industry 4.0 and automotive Systems

Specifying and verifying reliability for deployment

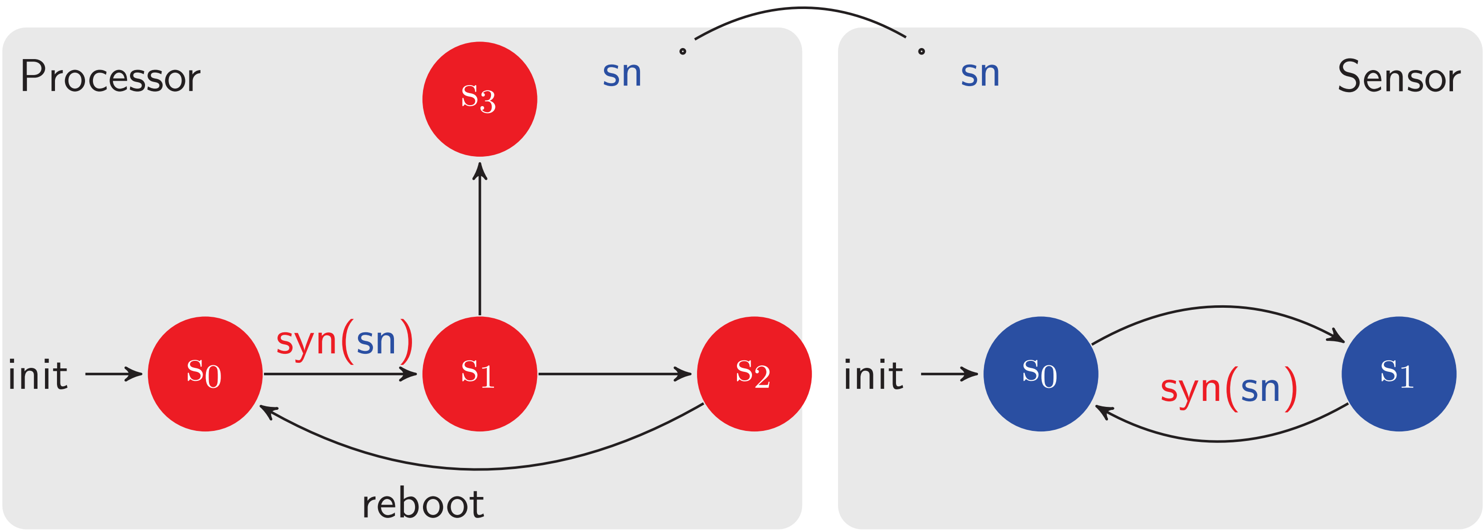
Generating Formal Models



Abdelhakim Baouya, Otmane Ait Mohamed, Djamal Bennouar, and **Samir Ouchani**. Safety analysis of train control system based on model-driven design methodology.
Computers in Industry, 105 :1–16, 2019 [Q1, IF :11.245]

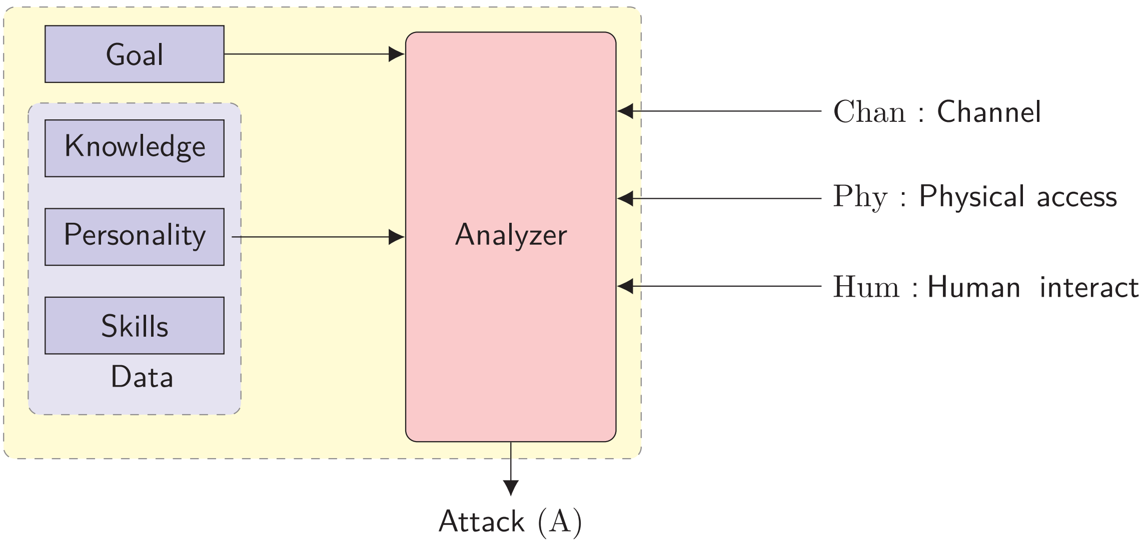
Specifying and verifying reliability for deployment

Formal Semantics



Assessing the Severity of Smart Attacks

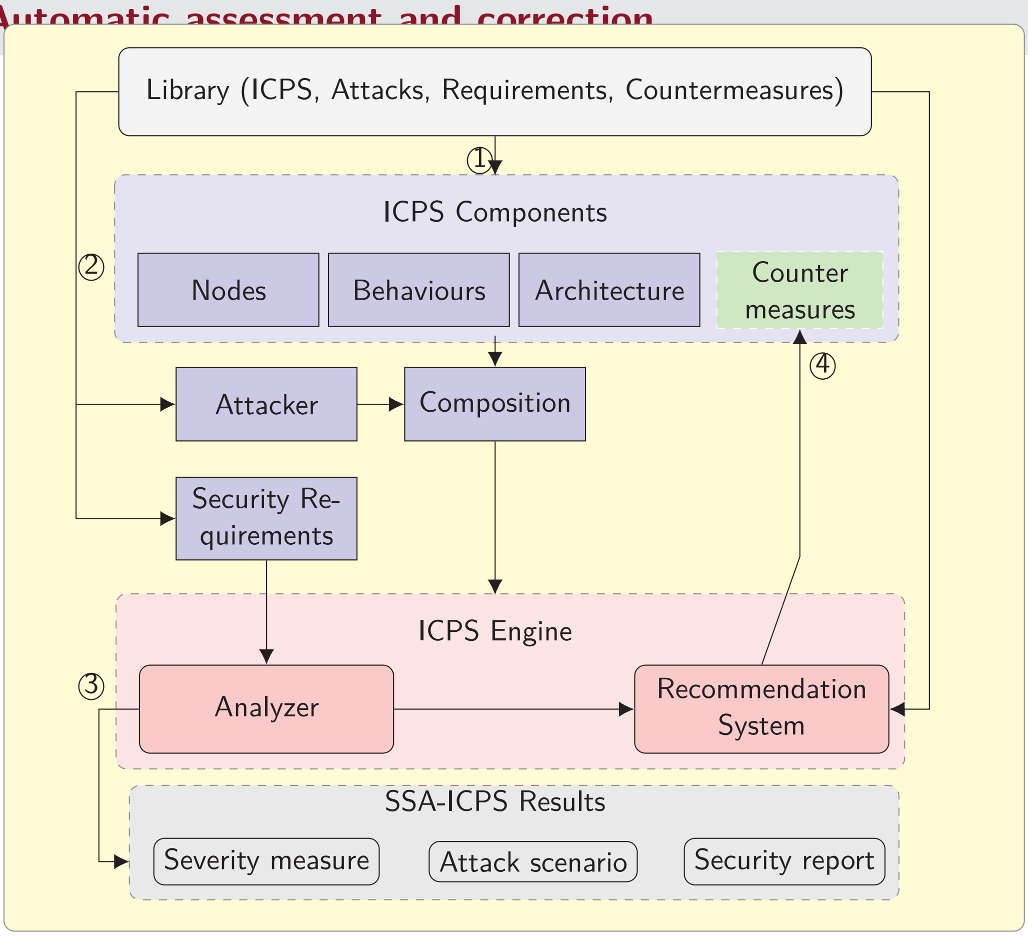
Attacker Model



Abd El-Aziz Khaled, **Samir Ouchani**, Zahir Tari, and Khalil Drira. Assessing the Severity of Smart Attacks in Industrial Cyber Physical Systems.
Transactions on Cyber Physical Systems, 2020 [Q1, IF :3.05]

Assessing the Severity of Smart Attacks

Automatic assessment and correction



① ICPS instantiation

② Security instantiation

③ Results generation

④ Security reinforcement

Trustworthy SCPS

Smart CPS

Outputs

1. Abdelhakim Baouya, Otmane Ait Mohamed, Djamal Bennouar, and **Samir Ouchani**. Safety analysis of train control system based on model-driven design methodology.
   Computers in Industry, 105 :1–16, 2019 [Q1, IF :11.245]

2. Abdelhakim Baouya, Salim Chehida, **Samir Ouchani**, Saddek Bensalem, and Marius Bozga. Generation and verification of learned stochastic automata using k-nn and statistical model checking.
   Applied Intelligence, Nov 2021 [Q2, IF :5.019]

3. Abdelhakim Baouya, Otmane Ait Mohamed, **Samir Ouchani**, and Djamal Bennouar. Reliability-driven Automotive Software Deployment based on a Parametrizable Probabilistic Model Checking.
   Expert Systems with Applications, page 114572, 2021 [Q1, IF :8.665]

1. Abd El-Aziz Khaled, **Samir Ouchani**, Zahir Tari, and Khalil Drira. Assessing the Severity of Smart Attacks in Industrial Cyber Physical Systems.
   Transactions on Cyber Physical Systems, 2020 [Q1, IF :3.05]

2. **Samir Ouchani** and Khaled Abdelaziz. A meta language for cyber-physical systems and threats : Application on autonomous vehicle.
   In *16th International Conference on Computer Systems and Applications AICCSA*, page 8. ACS/IEEE, 2019 [Core B]

Trusted Smart CPS

Subjects

Security, and data privacy, decentralization

Main contributions

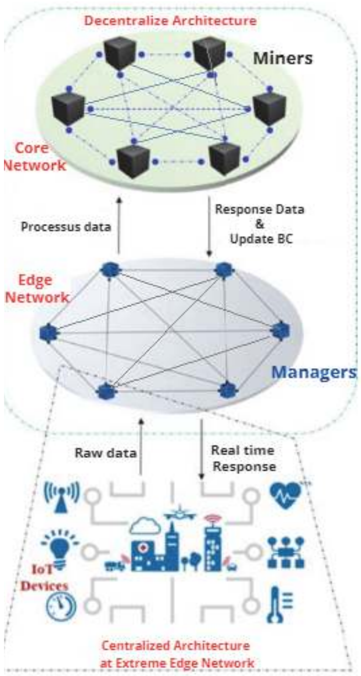
- Reinforcing security in Smart CPS
- Blockchain for privacy preservation in Smart Cities

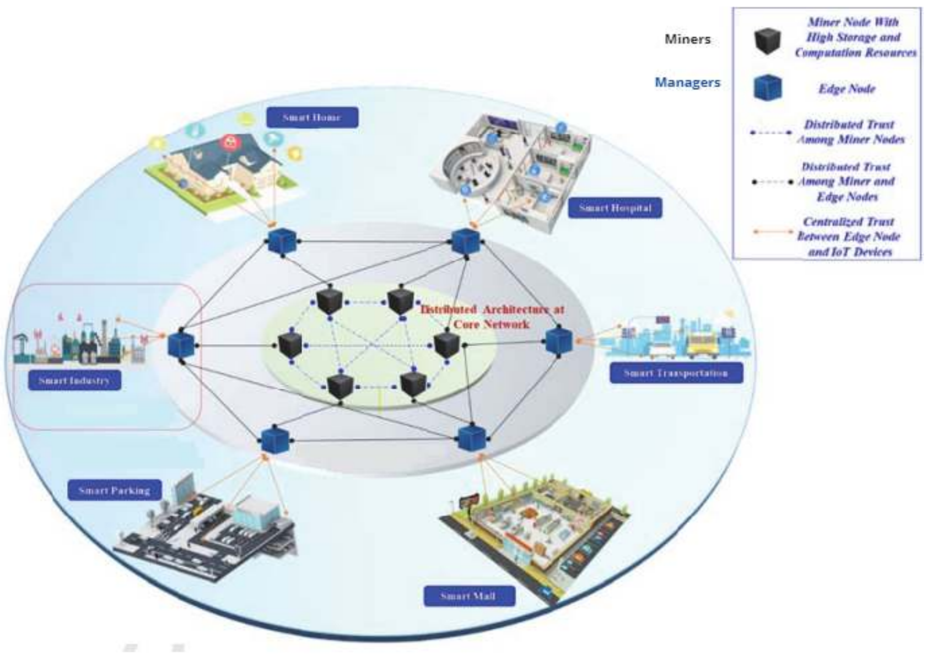
Applications

- Smart Living and Smart Cities

Secure Smart Systems

Decentralized Smart City Modeling

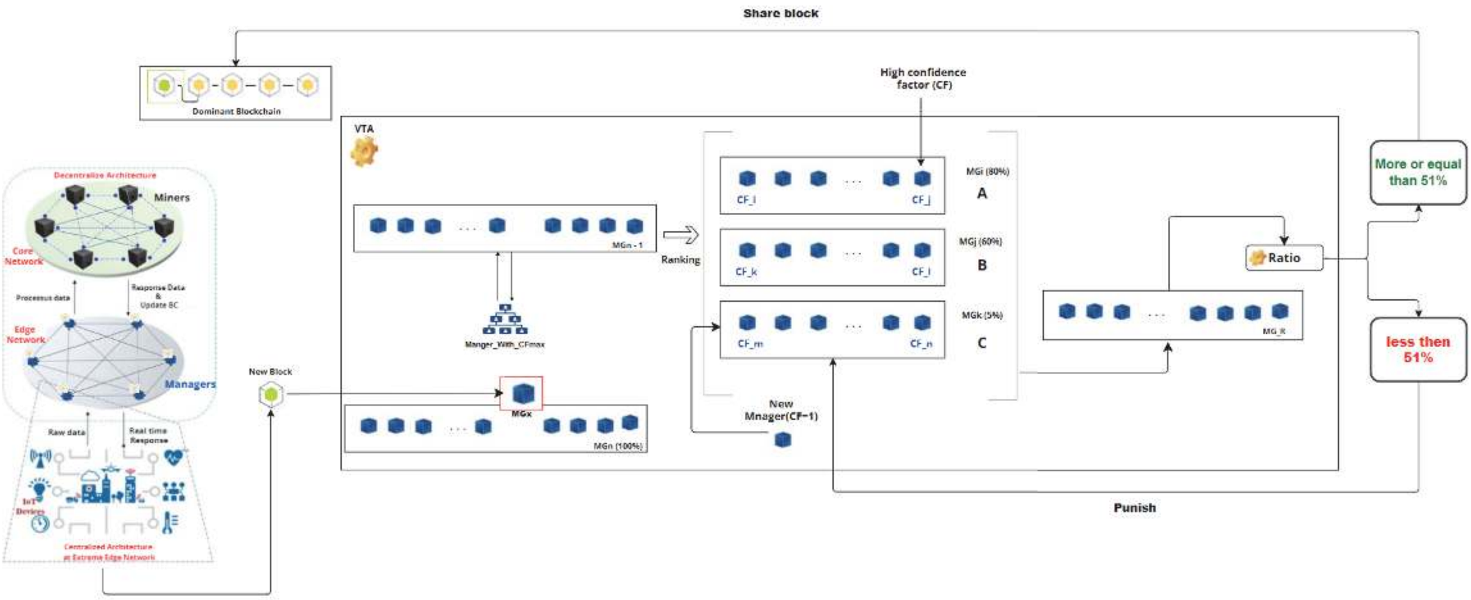




Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. Towards a reliable smart city through formal verification and network analysis.
Computer Communications, 180 :171–187, 2021 [Q1, IF :5.047]

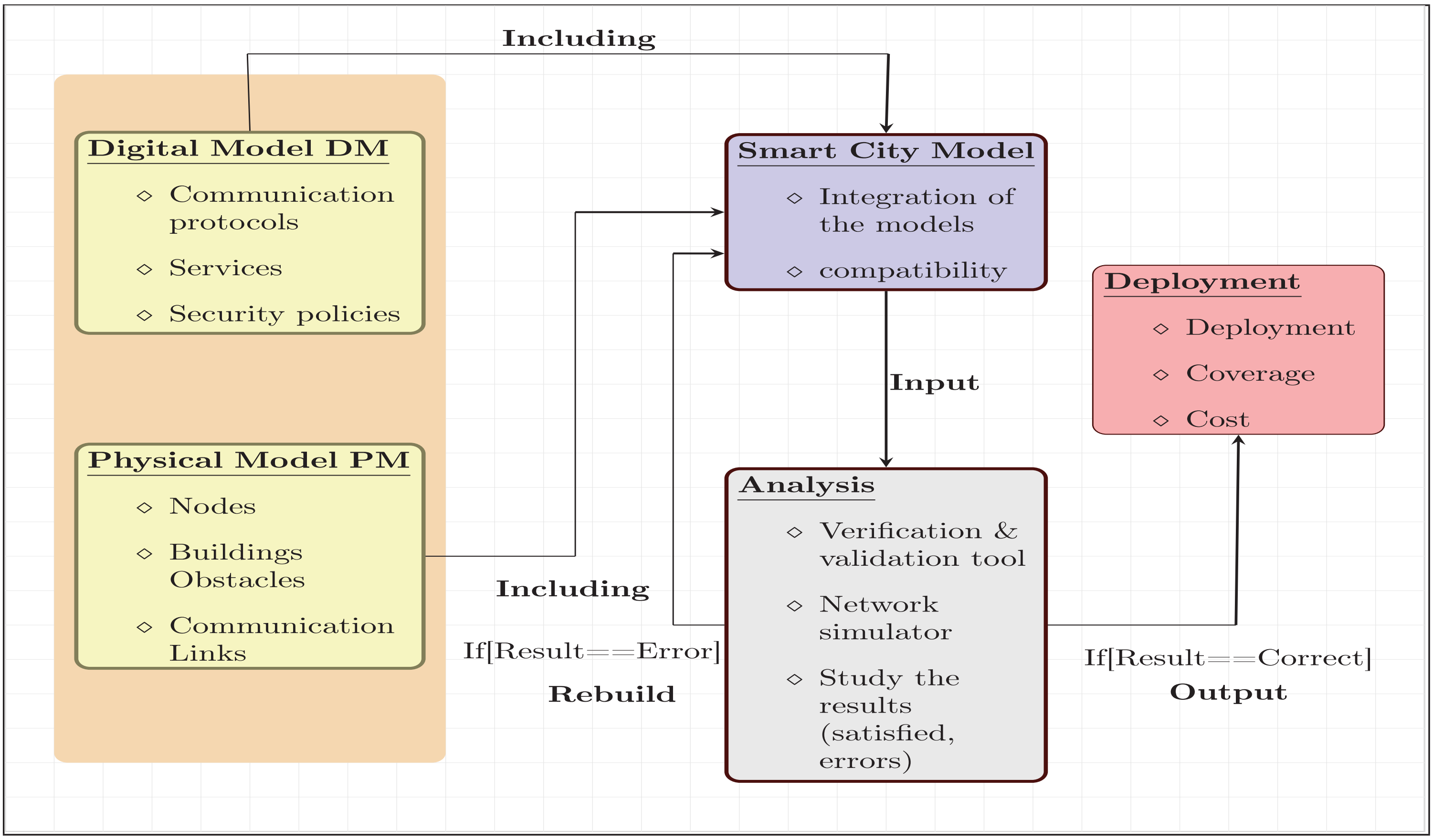
Secure Smart Systems

Blockchain Network



Secure Smart Systems

Simulation and Verification Approach



Secure Smart Systems

Outputs

Ph.D. Thesis - Walid Miloud Dahmane

1. **Title :** Security by Construction Through Formal Methods and Blockchain : Application On IoT Networks in Smart Cites
2. **Co-supervision :** Pr. Hafida Bouarfa
3. **Institution :** Blida University
4. **Duration :** 2018 -2022
5. **Publications :** 2 published journal papers, 4 published conference papers, and 1 journal paper under review

Selected papers

1. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. Guaranteeing information integrity and access control in smart cities through blockchain.
   Journal of Ambient Intelligence and Humanized Computing, pages 1–10, 2022 [Q1, IF :7.104]

2. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. Towards a reliable smart city through formal verification and network analysis.
   Computer Communications, 180 :171–187, 2021 [Q1, IF :5.047]

3. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. A smart living framework : Towards analyzing security in smart rooms.
   In *International Conference on Model and Data Engineering*, pages 206–215. LNCS Springer, 2019 [CORE C]

Lightweight and Secure Authentication

Subjects

Silicon PUF, IoT, identification, and lightweight authentication

Main contributions

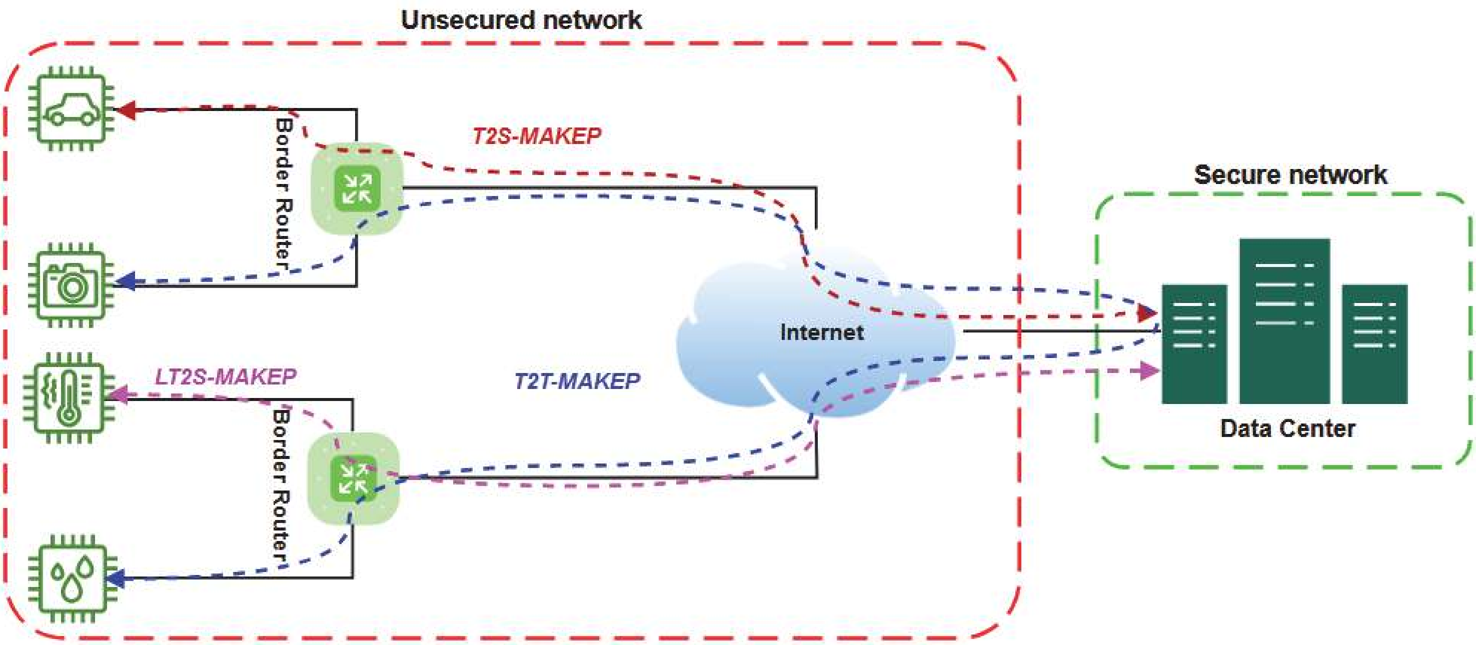
- Thing-to-thing and store-less lightweight authentication
- IoT-based cryptographic keys generation, reproduction, and correction
- Silicon PUF for IoT Networks

Applications

- Logistics and smart transportation

IoT-based Authentication

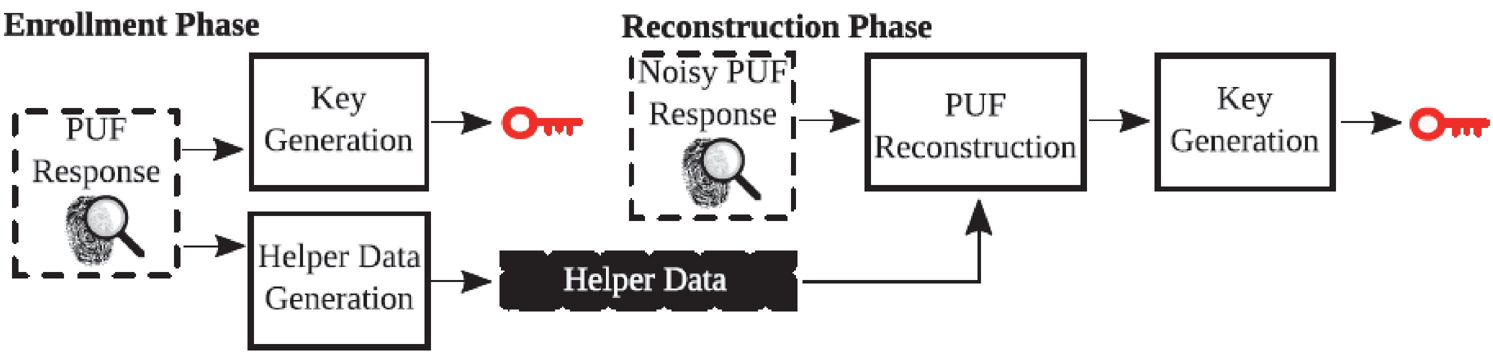
Lightweight and storeless schemes



Fahem Zerrouki, **Samir Ouchani**, and Hafida Bouarfa. Puf-based mutual authentication and session key establishment protocol for IoT devices.
Journal of Ambient Intelligence and Humanized Computing, pages 1–19, 2022 [Q1, IF :7.104]

IoT-based Authentication

Keys Generation, Correction, and Reproduction



Zerrouki Fahem, **Samir Ouchani**, and Bouarfa Hafida. A low-cost authentication protocol using arbiter-puf. In *International Conference on Model and Data Engineering*, pages 101–116. Springer, 2021

Analyzing SCPS

1. Fahem Zerrouki, **Samir Ouchani**, and Hafida Bouarfa. Puf-based mutual authentication and session key establishment protocol for IoT devices.
   Journal of Ambient Intelligence and Humanized Computing, pages 1–19, 2022 [Q1, IF :7.104]

2. Zerrouki Fahem, **Samir Ouchani**, and Bouarfa Hafida. A survey on silicon pufs.
   Journal of Systems Architecture, page 102514, 2022 [Q1, IF :5.836]

3. Zerrouki Fahem, **Samir Ouchani**, and Bouarfa Hafida. A low-cost authentication protocol using arbiter-puf.
   In *International Conference on Model and Data Engineering*, pages 101–116. Springer, 2021 [CORE C]

Analyzing SCPS

Model-based Security and Reliability

1. Abdelhakim, Baouya, Otmane Ait Mohamed, **Samir, Ouchani**, and Djamal Bennouar. Reliability-driven automotive software deployment based on a parametrizable probabilistic model checking.
   Expert Systems with Applications, page 114572, 2021.

2. Baouya, Abdelhakim, Otmane Ait Mohamed, and **Samir Ouchani**. Toward a context-driven deployment optimization for embedded systems : a product line approach.
   The Journal of Supercomputing.

3. Abdelhakim, baouya, **Samir Ouchani**, and Saddek Bensalem. Formal modeling and security analysis of inter-operable systems.
   In *The 35th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*. Springer, 2022.

4. **Ouchani, Samir**. Towards a fractionation-based verification : application on sysml activity diagrams.
   In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 2032–2039, 2019.

5. Abdelhakim, baouya, **Samir Ouchani**, and Saddek Bensalem. Formal modeling and security analysis of inter-operable systems.
   In *The 35th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*. Springer, 2022.

Fractionation-based Verification

Abstraction

- Ψ considers a PCTL expression ϕ to be verified on A
- Σ_{ϕ} is the set of the atomic propositions of ϕ s.t. $\Sigma_{\phi} \subseteq \mathscr{N}$

Definition

For a given System $\mathrm{A} \uparrow_{\mathtt{a}} \mathrm{A}'$ and a PCTL expression ϕ such that $\Sigma_{\phi} \subseteq \mathscr{N}$, we have

- $\forall \mathtt{a_x} \notin {}_{\phi} \wedge \mathtt{a_x} \in \mathscr{N} \cup \mathscr{N}' : (\mathtt{a_x N}) = \mathtt{N}$.
- $\Sigma_{\phi} \cap \mathscr{N}_{\mathrm{A}'} = \emptyset : \Psi(\mathrm{A} \uparrow_{\mathtt{a}} \mathrm{A}') = \mathrm{A}$.

Fractionation-based Verification

Reduction

- After abstraction, the size of A will be reduced
- Υ develops a set of reduction rules to compact more the resulted A

Definition

For a system A, we define a set of reduction rules that are applicable on the artifacts $\|$, $|$, $\blacklozenge$, and $\Diamond$ as follows.

- $\Upsilon(\|(a_1, \|(a_2, a_3))) = \|(a_1, a_2, a_3)$,
- $\Upsilon(|(a_1, |(a_2, a_3))) = |(a_1, a_2, a_3)$,
- $\Upsilon(\blacklozenge(a_1, \blacklozenge(a_2, a_3))) = \blacklozenge(a_1, a_2, a_3)$,
- $\Upsilon(\Diamond_p(a_1, \Diamond_{p'}(a_2, a_3))) = \Diamond_{p.p', p.(1-p'), (1-p).(1-p')}(a_1, a_2, a_3)$,
- $\Upsilon(\Diamond_g(a_1, \Diamond_{g'}(a_2, a_3))) = \Diamond_{g \wedge g', \neg g \wedge g', \neg g \wedge \neg g'}(a_1, a_2, a_3)$.

Composition-based Verification

Overview

- The decomposition operator "$\natural$" decomposes the PCTL property ϕ into local ones $\phi_{i:0\leq i\leq n}$ over A_i with respect to the call behavior actions $a_{i:0\leq i\leq n}$ (interfaces)
- The operator "$\natural$" is based on substituting the propositions of A_i to the propositions related to its interface a_{i-1}
- We denote by $\phi[y/z]$ substituting the atomic proposition "z" in the PCTL property ϕ by the atomic proposition "y"

Composition-based Verification

Property decomposition

Definition (PCTL Property Decomposition)

Let ϕ be a PCTL property to be verified on $A_1 \uparrow_a A_2$. The decomposition of ϕ into ϕ_1 and ϕ_2 is denoted by $\phi \equiv \phi_1 \natural_a \phi_2$ where AP_{A_i} are the atomic propositions of A_i, then :

1. $\phi_1 = \phi([l_a/AP_{A_2}])$, where l_a is the atomic proposition related to the action a in A_1.
2. $\phi_2 = \phi([\top/AP_{A_1}])$.

Composition-based Verification

Generalization

- We generalize the satisfiability of ϕ on A with n call behaviors

Proposition (CV-Generalization)

Let ϕ *be a PCTL property to be verified on* A, *such that :*
$\mathrm{A} = \mathrm{A}_0 \uparrow_{\mathrm{a}_0} \cdots \uparrow_{\mathrm{a}_{n-1}} \mathrm{A}_\mathrm{n}$ *and* $\phi = \phi_0 \natural_{\mathrm{a}_0} \cdots \natural_{\mathrm{a}_{n-1}} \phi_\mathrm{n}$, *then :*

$$\frac{\mathrm{A}_0 \models \phi_0 \cdots \mathrm{A}_\mathrm{n} \models \phi_\mathrm{n} \qquad \phi = \phi_0 \natural_{\mathrm{a}_0} \cdots \natural_{\mathrm{a}_{n-1}} \phi_\mathrm{n}}{\mathrm{A}_0 \uparrow_{\mathrm{a}_0} \cdots \uparrow_{\mathrm{a}_{n-1}} \mathrm{A}_\mathrm{n} \models \phi}$$

Analyzing SCPS

Hardening

Model-based Security

1. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. Towards a reliable smart city through formal verification and network analysis.
   Computer Communications, 180 :171–187, 2021.

2. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. A smart living framework : Towards analyzing security in smart rooms.
   In *International Conference on Model and Data Engineering*, pages 206–215. LNCS Springer, 2019.

3. Abdelhakim Baouya, Otmane Ait Mohamed, Djamal Bennouar, and **Samir Ouchani**. Safety analysis of train control system based on model-driven design methodology.
   Computers in Industry, 105 :1–16, 2019.

IoT-based Security

1. **Ouchani, Samir**. Towards a security reinforcement mechanism for social cyber-physical systems.
   In *The Third International Conference on Smart Applications and Data Analysis for Smart Cyber-Physical Systems (Invited Paper)*, page 14. LNCS, Springer, 2020.

2. **Ouchani, Samir**. A security policy hardening framework for socio-cyber-physical systems.
   Journal of Systems Architecture, 119 :102259, 2021.

Hardening

Policy Constrained Semantics

- By definition an intruder is freed from playing by the rules

Definition (Honest Trace)

An *honest trace* is a trace whose underlying sequence of states, $S_0 \cdot \ldots \cdot S_i \cdot S_{i+1} \cdot \ldots$ is such that $(S_i, S_{i+1}) \in$, for all $i \geq 0$ and where the label of is not the intruder's ID.

- For the set of traces $\texttt{Traces}_H(\mathscr{S})$ in $\mathscr{S}$ of honest agents (H), we consider the set of traces satisfying a given security statement

Definition (Trace satisfying φ)

A trace satisfying φ is a trace in $\texttt{traces}(\mathscr{S}, \varphi) = \texttt{Traces}(\mathscr{S}) \cap \texttt{Words}(\varphi)$. An honest trace satisfying φ is a trace in $\texttt{traces}_H(\mathscr{S}, \varphi) = \texttt{Traces}_H(\mathscr{S}) \cap \texttt{Words}(\varphi)$.

Hardening

Policy Constrained Semantics

- In an honest trace, the affectedness distinguishes the requirements whose validity can be changed if the policy is enforced from those whose validity is unchanged by it.

Definition (Requirements/Policies Affectedness)

Let φ be a requirement, π a policy, and $\mathscr{S}$ models the executions. We say that φ is affected by π in $\mathscr{S}$, and we write it $\varphi \leftharpoonup \varphi'$, when $\mathtt{traces}_{\mathrm{H}}(\mathscr{S}, \varphi) \subseteq \mathtt{traces}_{\mathrm{H}}(\mathscr{S}, \neg\pi) \neq \emptyset$.

- In $\mathscr{S}_{|\pi}$ where $\mathscr{S}$ is enforced by π, no requirement must change its validity

Definition

The system $\mathscr{S}$ constrained by π is a new $\mathscr{S}' = \mathrm{S}', \mathrm{S}_0,'$ satisfying.

- If $\mathscr{S} \not\models_{\mathrm{H}} \pi$ then $\mathscr{S}' \models_{\mathrm{H}} \pi$;
- For all p such that $\mathrm{p} \not\leftharpoonup \pi$, if $\mathscr{S} \models_{\mathrm{H}} \mathrm{p}$ then $\mathscr{S}' \models_{\mathrm{H}} \mathrm{p}$.

Analyzing SCPS

IoT-based Security Project

1. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. Towards a reliable smart city through formal verification and network analysis.
   Computer Communications, 180 :171–187, 2021.

2. Walid Miloud Dahmane, **Samir Ouchani**, and Bouarfa Hafida. A smart living framework : Towards analyzing security in smart rooms.
   In *International Conference on Model and Data Engineering*, pages 206–215. LNCS Springer, 2019.

3. Abdelhakim Baouya, Otmane Ait Mohamed, Djamal Bennouar, and **Samir Ouchani**. Safety analysis of train control system based on model-driven design methodology.
   Computers in Industry, 105 :1–16, 2019.

Analyzing SCPS

Probabilistic Verification

- PRISM checks probabilistic specifications (PCTL expression) over probabilistic models (probabilistic timed automata)
- A PRISM program is a set of *modules*, communicates à la CSP process algebra, each having a countable set of boolean or integer, local, variables
- A module is a set of probabilistic and/or Dirac commands : $[\alpha]\ \mathrm{g} \rightarrow \mathrm{p}_1 : \mathrm{u}_1 + ... + \mathrm{p}_\mathrm{m} : \mathrm{u}_\mathrm{m},\ \mathrm{u}_\mathrm{i}(\mathrm{v}'_\mathrm{j} = \mathrm{val}_\mathrm{j}) \& \cdots \& (\mathrm{v}'_\mathrm{k} = \mathrm{val}_\mathrm{k})$
- Dirac command : $[\alpha]\ \mathrm{g} \rightarrow \mathrm{u}$
- A state reward is expressed by $\mathrm{g} : \mathrm{r}$. A transition reward $[\mathrm{a}]\ \mathrm{g} : \mathrm{r}$
- The guard g is a propositional logic formula over local and global variables

Analyzing SCPS

Probabilistic Verification

- $\mathscr{T}_{\mathrm{P}}$ assigns for each entity an equivalent PRISM code fragment
- $\mathrm{o}_{\mathrm{o}_2}$ the object o possess o_2, l_{a} and l_{o} present the locations of a and o, and $\mathrm{p}_{\mathrm{o}_3}$ precises the physicality attribute of o_3.

$$\mathscr{T}_{\mathrm{P}}(\alpha) = \begin{cases} [\mathrm{Syn}_{\mathrm{o}_2}]\mathrm{o}_{\mathrm{o}_2} \wedge \mathrm{o1}_{\mathrm{o}_3} \wedge \neg \mathrm{p}_{\mathrm{o}_2} \wedge \neg \mathrm{p}_{\mathrm{o}_3} \to (\mathrm{o}_2' = \mathrm{o}_2); \\ [\mathrm{Syn}_{\mathrm{o}_2}]\mathrm{o}_{\mathrm{o}_2} \wedge \mathrm{o1}_{\mathrm{o}_3} \wedge \neg \mathrm{p}_{\mathrm{o}_2} \wedge \neg \mathrm{p}_{\mathrm{o}_3} \to (\mathrm{o}_3' = \mathrm{o}_2); \\ \texttt{iff:Send}_{\mathrm{O}}(\mathrm{o}_1, \mathrm{o}_2) \in \Sigma_{\mathrm{O}}^{\mathrm{o}1}, \texttt{Receive}_{\mathrm{O}}(\mathrm{o}_3, \mathrm{o}_2) \in \Sigma_{\mathrm{O}}^{\mathrm{o}2}. \\ [\mathrm{Tak}_{\mathrm{o}_1}]\mathrm{l}_{\mathrm{a}} = \mathrm{l}_{\mathrm{o}} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{lock}_{\mathrm{o}} \wedge \mathrm{p}_{\mathrm{o}_2} \to (\mathrm{a}'_{\mathrm{o}_2} = \top); \\ [\mathrm{Tak}_{\mathrm{o}_1}]\mathrm{l}_{\mathrm{a}} = \mathrm{l}_{\mathrm{o}} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{lock}_{\mathrm{o}} \wedge \mathrm{p}_{\mathrm{o}_2} \to (\mathrm{o}'_{\mathrm{o}_2} = \bot); \\ \texttt{iff:Receive}_{\mathrm{A}}(\mathrm{o}, \mathrm{o}_2) \in \Sigma_{\mathrm{A}}^{\mathrm{a}}. \\ [\mathrm{loc}_{\mathrm{o}_1}]\mathrm{o}_{\mathrm{o}_1} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{k}_{\mathrm{o}_1} \wedge \mathrm{p}_{\mathrm{o}_1} = \mathrm{p}_{\mathrm{o}_2} \to (\mathrm{k}'_{\mathrm{o}_1} = \top); \\ [\mathrm{loc}_{\mathrm{o}_1}]\mathrm{o}_{\mathrm{o}_1} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{k}_{\mathrm{o}_1} \wedge \mathrm{p}_{\mathrm{o}_1} = \mathrm{p}_{\mathrm{o}_2} \to (\mathrm{o}'_{\mathrm{o}_1} = \top); \\ \texttt{iff:Lock}_{\mathrm{O}}(\mathrm{o}_1, \mathrm{o}_2) \in \Sigma_{\mathrm{O}}^{\mathrm{o}}. \end{cases}$$

Analyzing SCPS

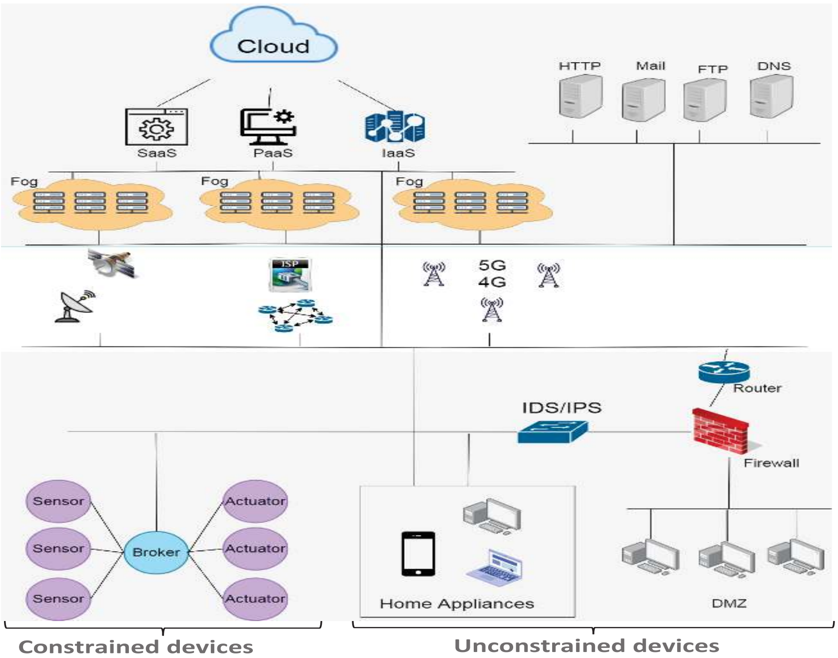
Probabilistic Verification

- $\mathscr{T}_{\mathrm{P}}$ assigns for each entity an equivalent PRISM code fragment
- $\mathrm{o}_{\mathrm{o}_2}$ the object o possess o_2, l_a and l_o present the locations of a and o, and $\mathrm{p}_{\mathrm{o}_3}$ precises the physicality attribute of o_3.

$$\mathscr{T}_{\mathrm{P}}(\alpha) = \begin{cases} [\mathrm{Syn}_{\mathrm{o}_2}]\mathrm{o}_{\mathrm{o}_2} \wedge \mathrm{o1}_{\mathrm{o}_3} \wedge \neg \mathrm{p}_{\mathrm{o}_2} \wedge \neg \mathrm{p}_{\mathrm{o}_3} \rightarrow (\mathrm{o}_2' = \mathrm{o}_2); \\ [\mathrm{Syn}_{\mathrm{o}_2}]\mathrm{o}_{\mathrm{o}_2} \wedge \mathrm{o1}_{\mathrm{o}_3} \wedge \neg \mathrm{p}_{\mathrm{o}_2} \wedge \neg \mathrm{p}_{\mathrm{o}_3} \rightarrow (\mathrm{o}_3' = \mathrm{o}_2); \\ \texttt{iff:Send}_{\mathrm{O}}(\mathrm{o}_1, \mathrm{o}_2) \in \Sigma_{\mathrm{O}}^{\mathrm{o}_1}, \texttt{Receive}_{\mathrm{O}}(\mathrm{o}_3, \mathrm{o}_2) \in \Sigma_{\mathrm{O}}^{\mathrm{o}_2}. \\ [\mathrm{Tak}_{\mathrm{o}_1}]\mathrm{l}_\mathrm{a} = \mathrm{l}_\mathrm{o} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{lock}_\mathrm{o} \wedge \mathrm{p}_{\mathrm{o}_2} \rightarrow (\mathrm{a}_{\mathrm{o}_2}' = \top); \\ [\mathrm{Tak}_{\mathrm{o}_1}]\mathrm{l}_\mathrm{a} = \mathrm{l}_\mathrm{o} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{lock}_\mathrm{o} \wedge \mathrm{p}_{\mathrm{o}_2} \rightarrow (\mathrm{o}_{\mathrm{o}_2}' = \bot); \\ \texttt{iff:Receive}_{\mathrm{A}}(\mathrm{o}, \mathrm{o}_2) \in \Sigma_{\mathrm{A}}^{\mathrm{a}}. \\ [\mathrm{loc}_{\mathrm{o}_1}]\mathrm{o}_{\mathrm{o}_1} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{k}_{\mathrm{o}_1} \wedge \mathrm{p}_{\mathrm{o}_1} = \mathrm{p}_{\mathrm{o}_2} \rightarrow (\mathrm{k}_{\mathrm{o}_1}' = \top); \\ [\mathrm{loc}_{\mathrm{o}_1}]\mathrm{o}_{\mathrm{o}_1} \wedge \mathrm{o}_{\mathrm{o}_2} \wedge \neg \mathrm{k}_{\mathrm{o}_1} \wedge \mathrm{p}_{\mathrm{o}_1} = \mathrm{p}_{\mathrm{o}_2} \rightarrow (\mathrm{o}_{\mathrm{o}_1}' = \top); \\ \texttt{iff:Lock}_{\mathrm{O}}(\mathrm{o}_1, \mathrm{o}_2) \in \Sigma_{\mathrm{O}}^{\mathrm{o}}. \end{cases}$$

Probabilistic and Network Simulation

Smart City Modeling



Level 3: Processing & storage
- Computing.
- Recording.
- Word Wide Web.

Level 2: Communication
- Providing internet.
- Connectivity.

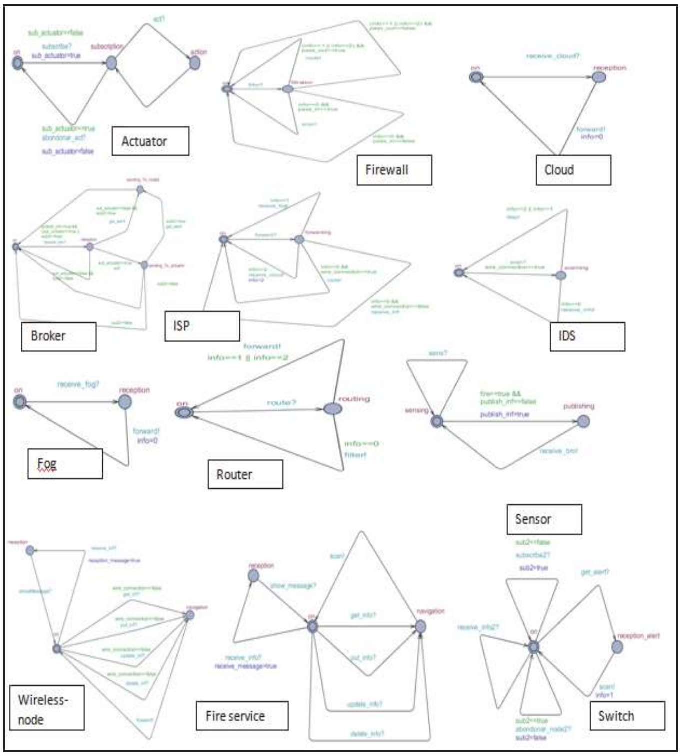
Level 1: Sensing & Action
- Monitoring
- Execution.
- Navigation.

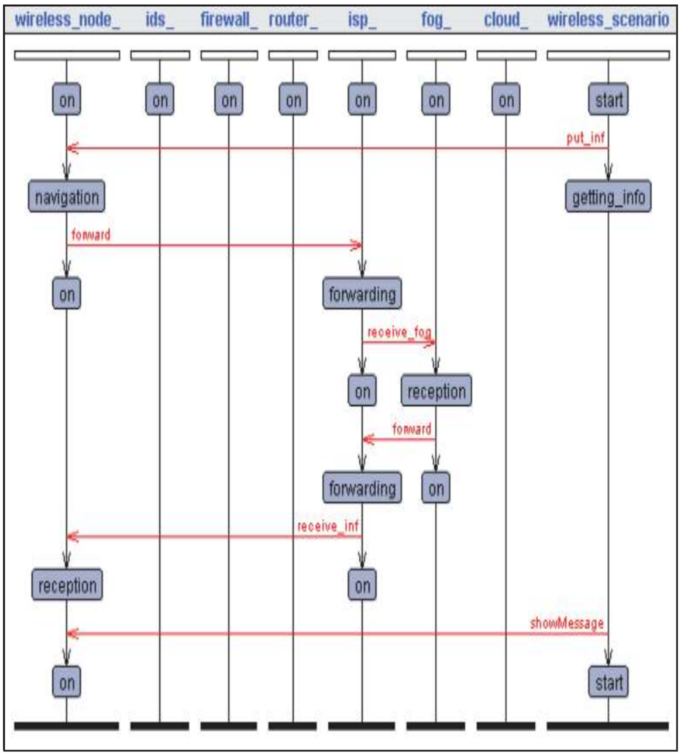
Smart City Application

Probabilistic and Network Simulation

Smart City Modeling

How do we test the compatibility of requirements ?

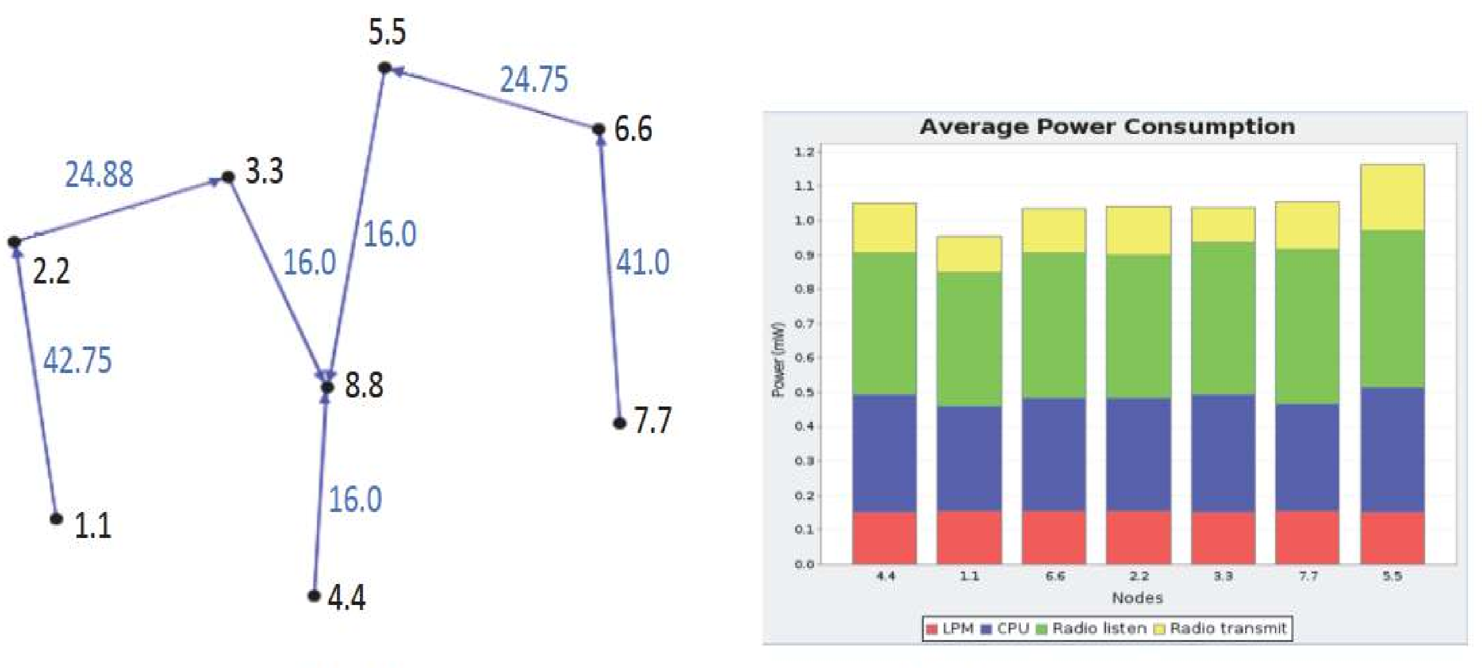




Probabilistic and Network Simulation

Smart City Modeling

How does deployment affect the spread of data and energy consumed ??



(a) DODAG. (b) Consummation of energy.

Conclusion

Research Project

- **Data Collection**
  - Collect **real-time data** on autonomous vehicle performance under various simulated threat scenarios
  - Gather **data on known cyber-attacks** or security breaches
  - Collect **user perception and awareness** data related to the security of autonomous vehicles
- **Data Analysis**
  - Analyze the **heterogeneous** security protocols in use
  - Analysis of **real-time performance** data under various threat scenarios
  - Analyze security breaches for **common patterns**, causes, or vulnerabilities that were exploited
  - Analyze user perception data to identify any gaps between user understanding and **security measures**

Thank you for your attention

Questions ?

CPSIot Tutorial
SCS and AI

Morayo Adjedouma

Luis Palacios

12/06/24

Introduction

Outline / Tutorial content & acitivities

- Lab presentation and R&D axes direction
- Based on last year work – Reacp (Luis + Morayo)
- Novelties
    - Tool to formalize ODD
        - Contains Graphical Description
        - Textual Description*
    - Check safety constraints from D.5.5 > CPS4EU (Luis)
    - Link/Check Drone as the use case/ »scenario » (Luis)
- Emphasis in the ODD tool

Tooling

- Papyrus > Profile > ODDs => Stand Alone tool (Morayo)
- Model (which model) interacts with the tool (TBD)

cea

2


LSEA lab
Lorem ipsum dolor sit amet
3

LSEA: Laboratoire conception de Systèmes Embarqués et Autonomes

Created
January 1st 2018
10 perm. + **1 Phd**

Group from former
LISE
Model-Driven Engineering Laboratory for Embedded Systems

Real-time embedded systems Design

Skills

- **Model Driven Engineering**
- **Real-time analysis**
- **Optimized deployment**
- **Tools and software platforms Development (Papyrus)**

Today

+35 members
15 permanents
06 CDDs / Post-docs
04 PhD
03 apprentices
Interns

Engineering of trusted autonomous cyber-physical systems

- Smart mobility
- Intelligent robotics
- Manufacturing
- Energy
- Digital health
- …

Additional skills

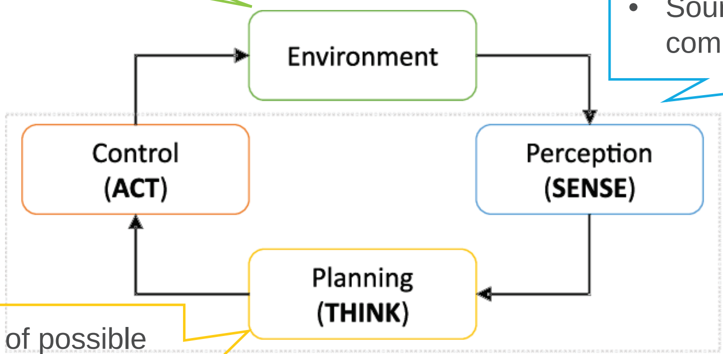
- **Artificial Intelligence**
- **Software Architectures for Robotics**
- **Augmented and collaborative engineering tools development**

4

Our challenges for autonomous CPS

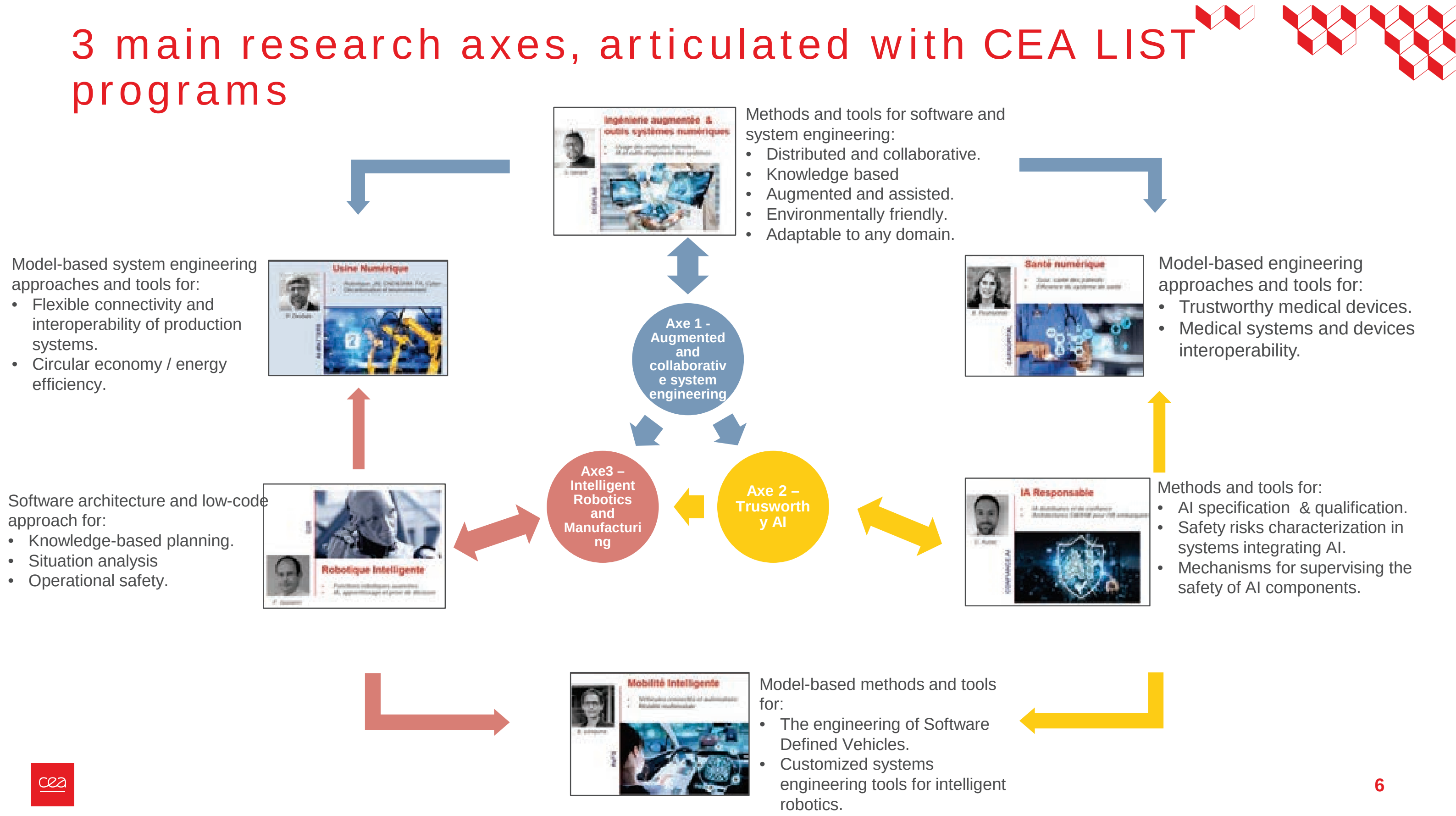
The world is complex (infinity of objects, combinations of states) and a source of hazards.

- Perception of complex situations.
- Use of not formally specified AIs and whose operation is unpredictable.
- Source of uncertainties which are complex to assess.



- A very large number of possible plans.
- The environment may evolve during planning (performance constraints).

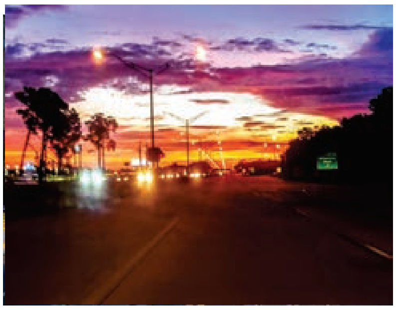
The complexity further increases in the context of collaborative autonomous systems (system of autonomous systems, etc.)

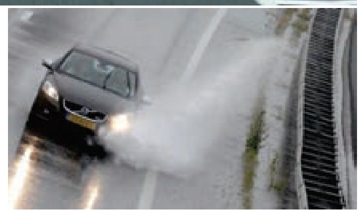
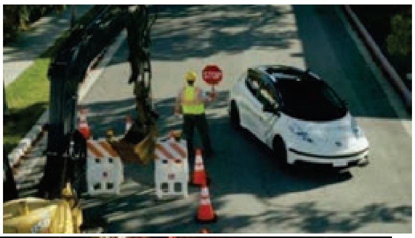

3 main research axes, articulated with CEA LIST programs
Ingénierie augmentée & outils systèmes numériques
Methods and tools for software and system engineering:
• Distributed and collaborative.
• Knowledge based
• Augmented and assisted.
• Environmentally friendly.
• Adaptable to any domain.
Usine Numérique
Model-based system engineering approaches and tools for:
• Flexible connectivity and interoperability of production systems.
• Circular economy / energy efficiency.
Santé numérique
Model-based engineering approaches and tools for:
• Trustworthy medical devices.
• Medical systems and devices interoperability.
Axe 1 - Augmented and collaborative system engineering
Axe3 – Intelligent Robotics and Manufacturing
Axe 2 – Trusworthy AI
Robotique Intelligente
Software architecture and low-code approach for:
• Knowledge-based planning.
• Situation analysis
• Operational safety.
IA Responsable
Methods and tools for:
• AI specification & qualification.
• Safety risks characterization in systems integrating AI.
• Mechanisms for supervising the safety of AI components.
Mobilité Intelligente
Model-based methods and tools for:
• The engineering of Software Defined Vehicles.
• Customized systems engineering tools for intelligent robotics.
cea
6

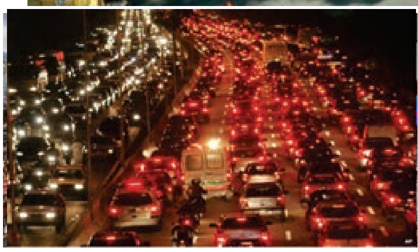

1
Operational Design Domain And Hazard Analysis
7

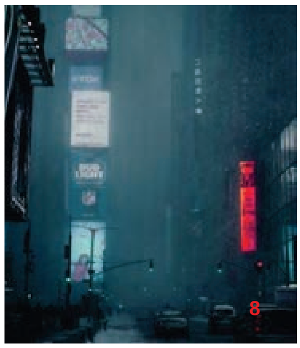
The motivation

In practice, the **number of possible scenarios** that have to be managed by an AI-enabled automated system tends to be **infinite**, which makes their safety evaluation challenging.





8

The challenges

1. **Complex/changing operational contexts; ambiguous scenarios;**

➔ 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.**

➔ 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.

1. … So, what is an ODD?

10

Operational Design Domain (ODD)

“**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**. “

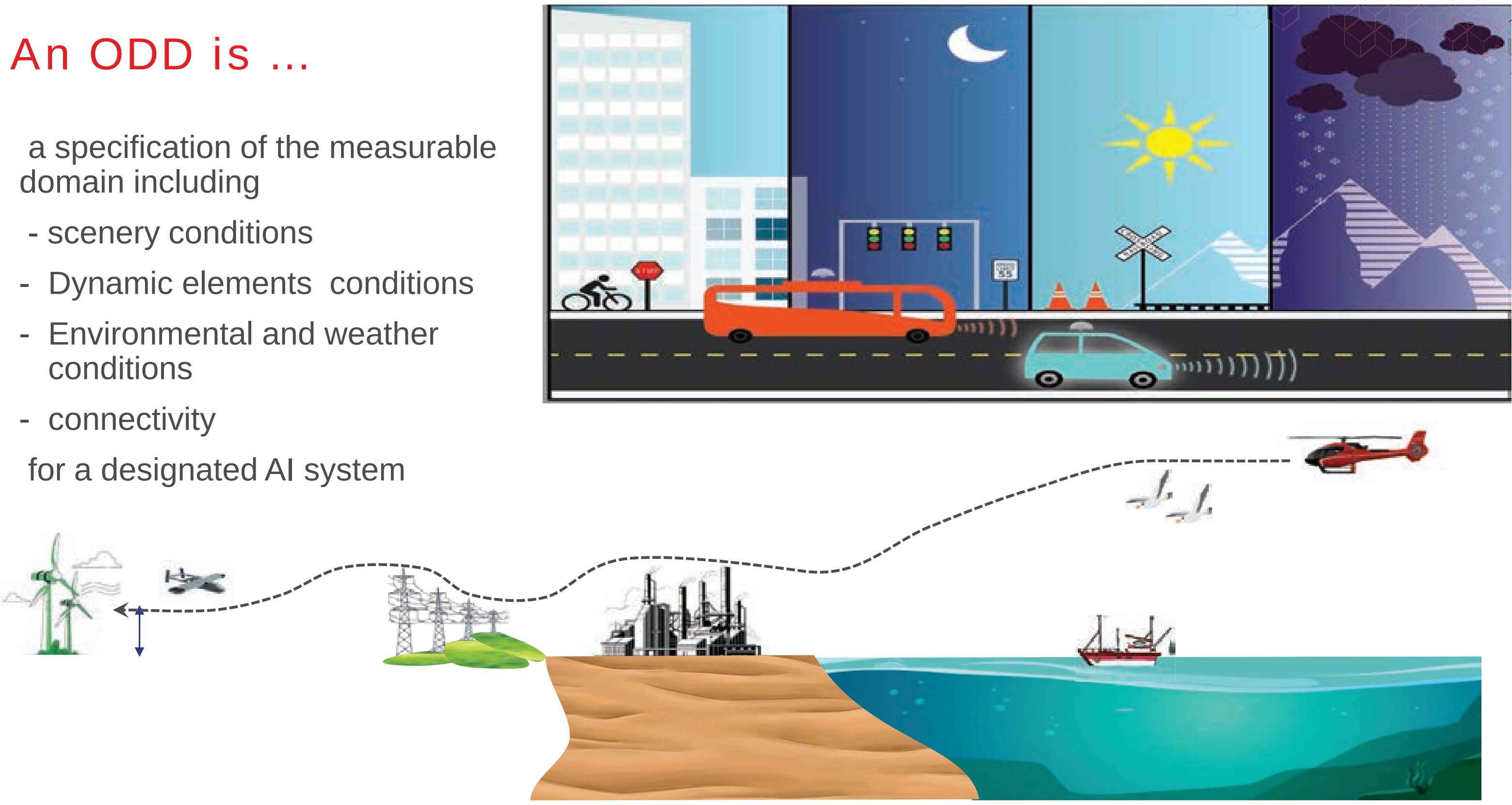
According to SAE J3016 (2021)

An ODD is …

a specification of the measurable domain including

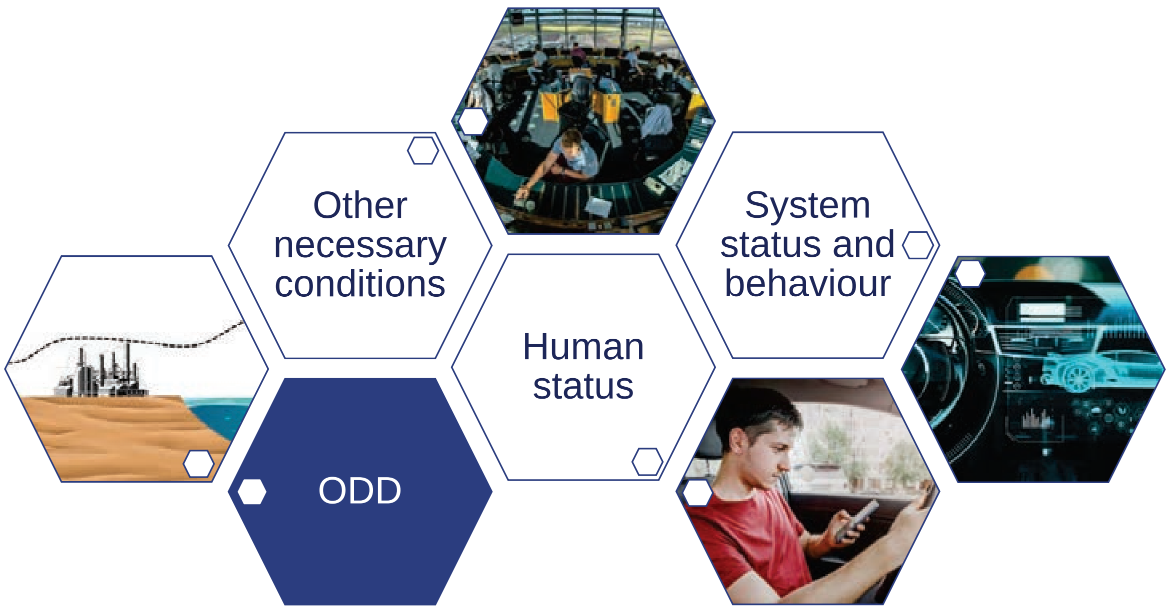
- scenery conditions
- Dynamic elements conditions
- Environmental and weather conditions
- connectivity

for a designated AI system



12

ODD is not the Operational Design Condition



13

“ ODD, a key concept in supporting EU AI regulation

International Organisation for Standardization (ISO)

14

« ..*Defining an ODD is crucial* for developers and regulators to establish clear expectations and communicate the intended operating conditions of automated systems.»

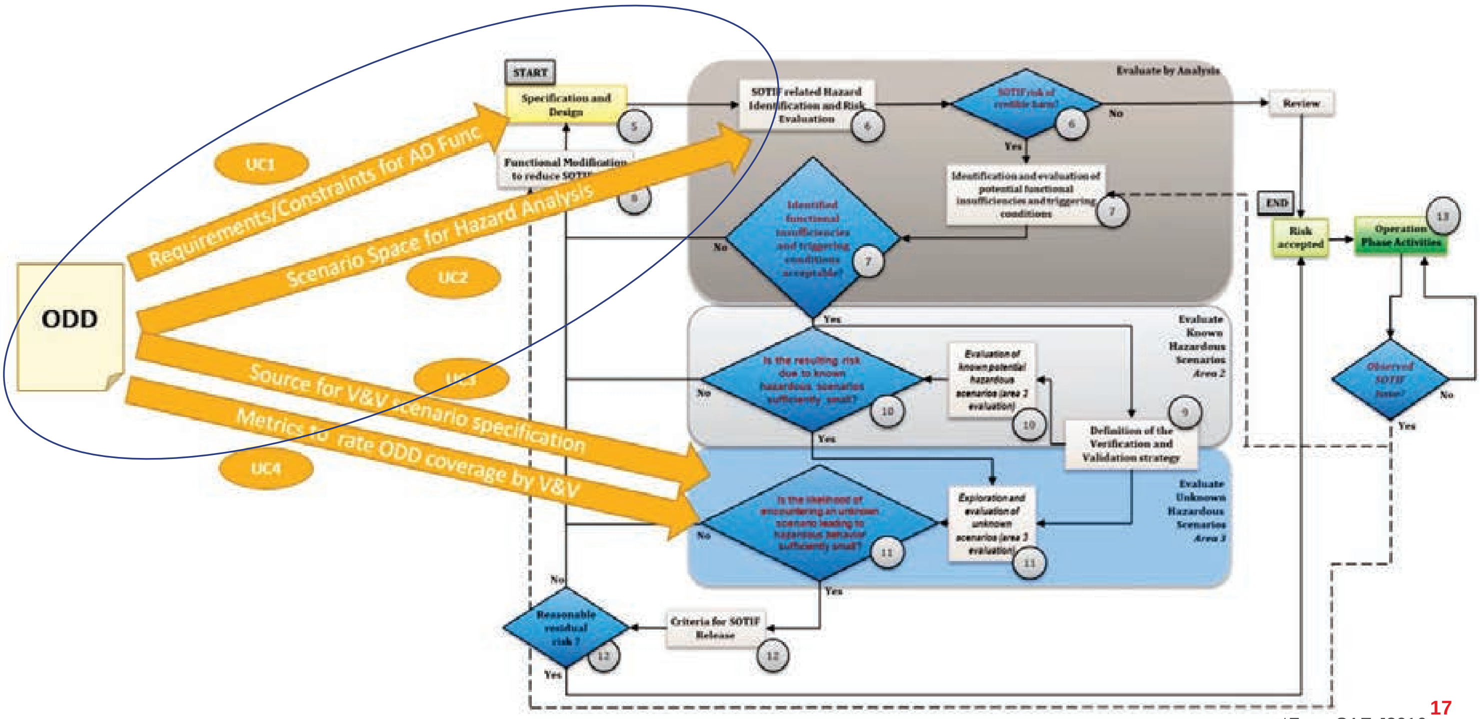
It is foreseen that ODD will *not be limited to safety issues* but can be used to ***address a continuum of assurance needs, from general purpose AI systems to automated systems***.

15

“ ODD: what for?

16

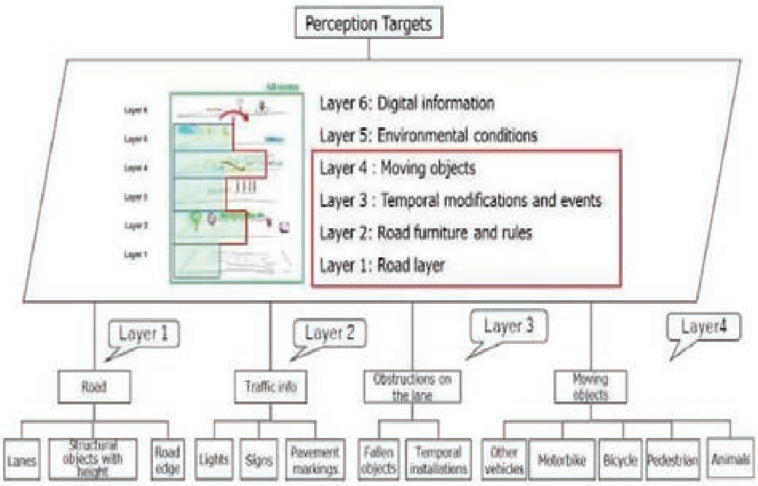
ODD: What usage?

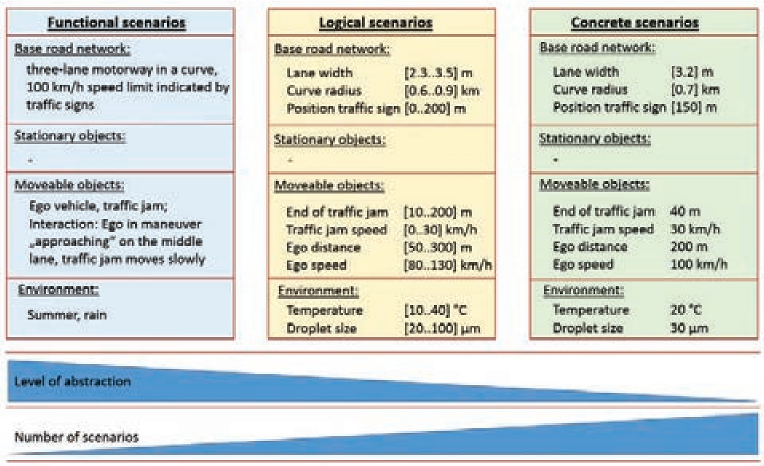


17

*From SAE J3016

Structuring of scenarios can be achieved from ODD

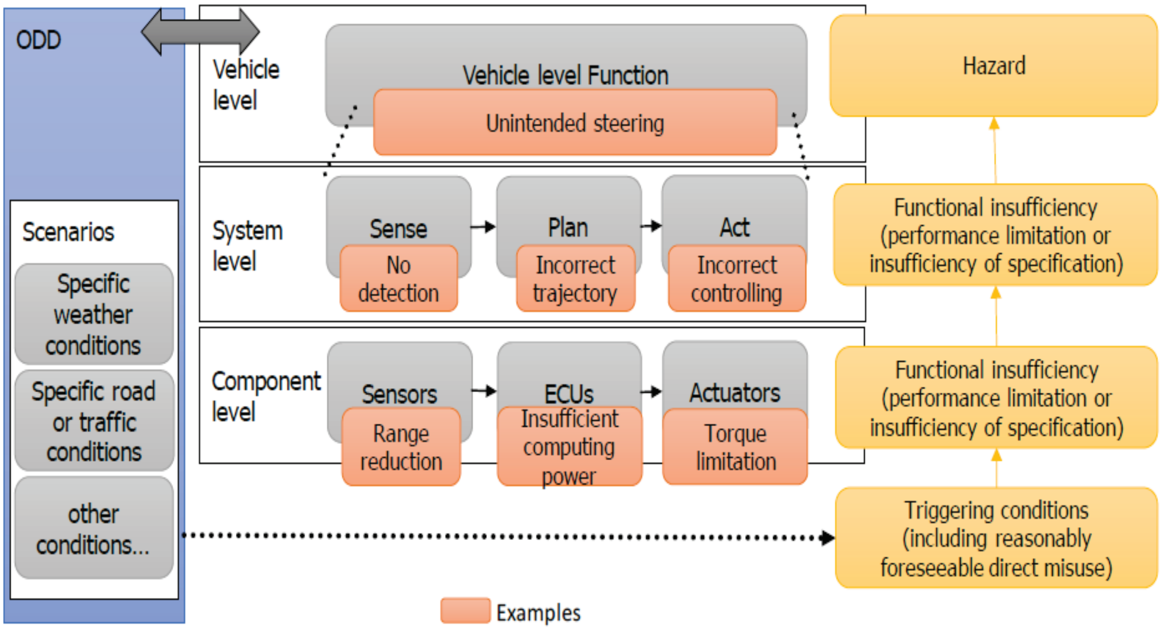




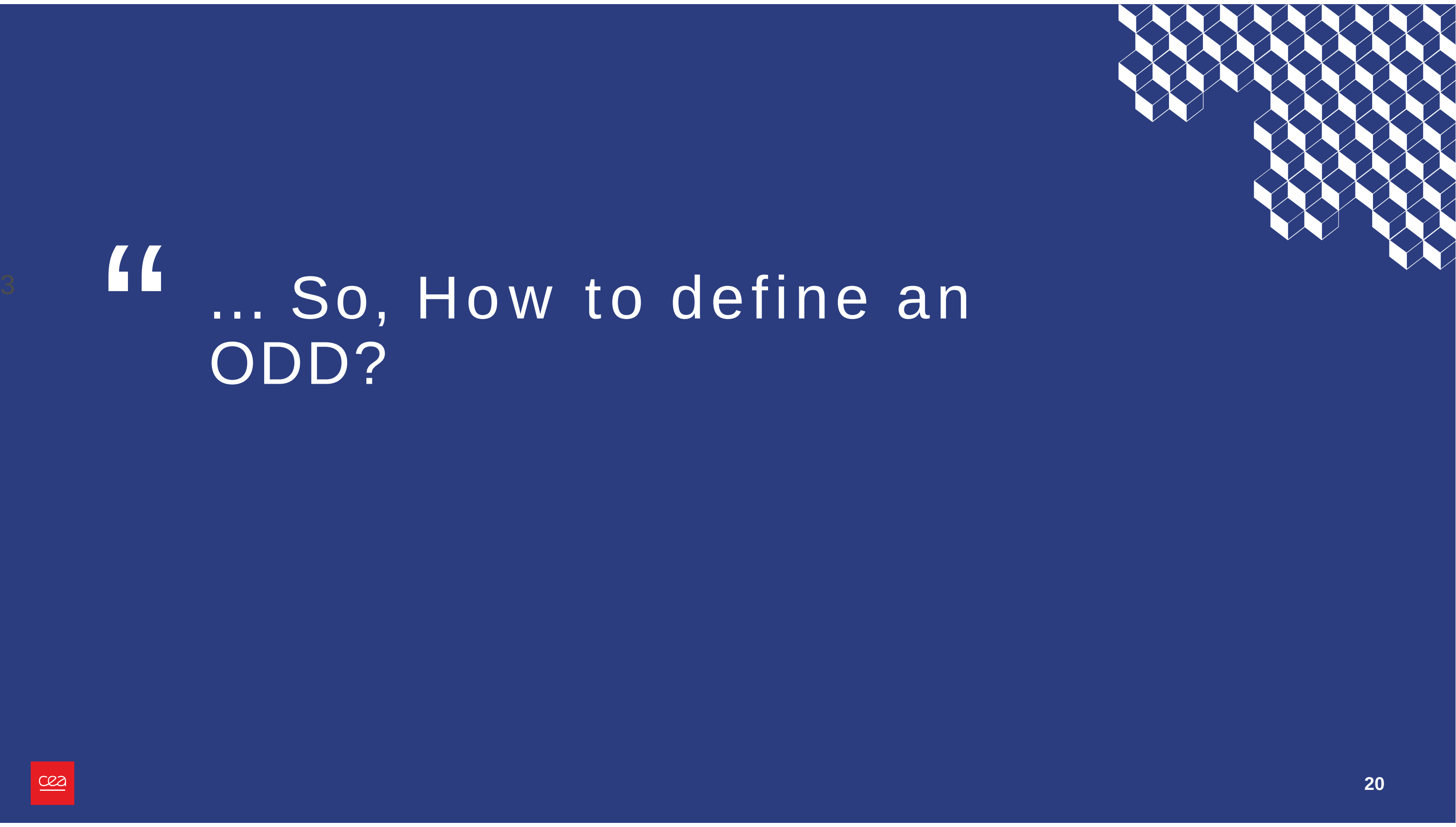
Description from functional to logical to concrete scenario
(and testing scenario as well)

18

ODD is key factor for Hazard analysis

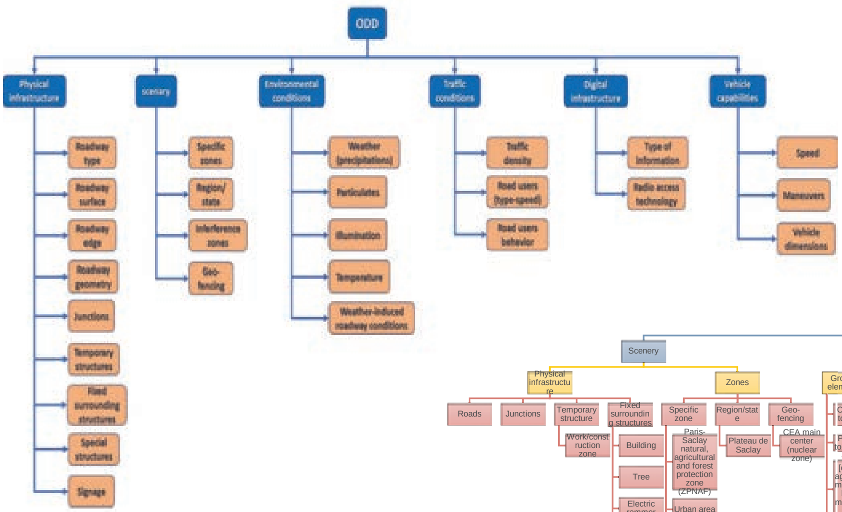


19

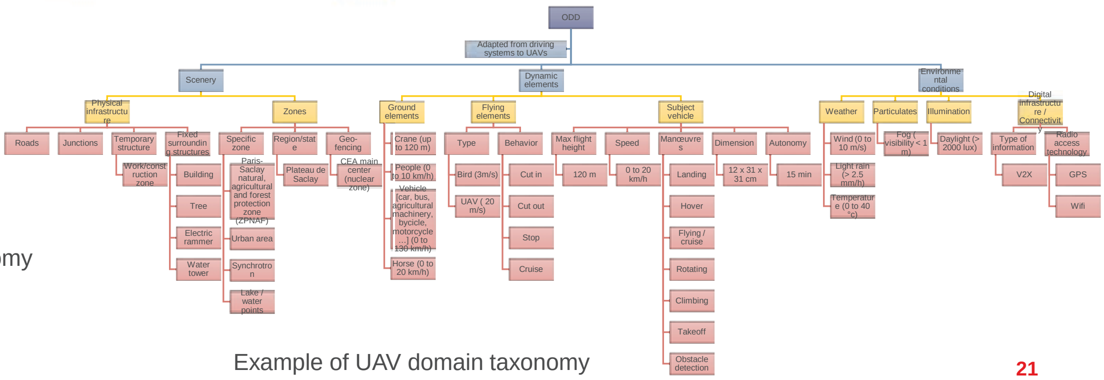

3
"
... So, How to define an ODD?
cea
20

Taxonomy for ODD definition

Taxonomy can be an effective way to define and implement the ODD



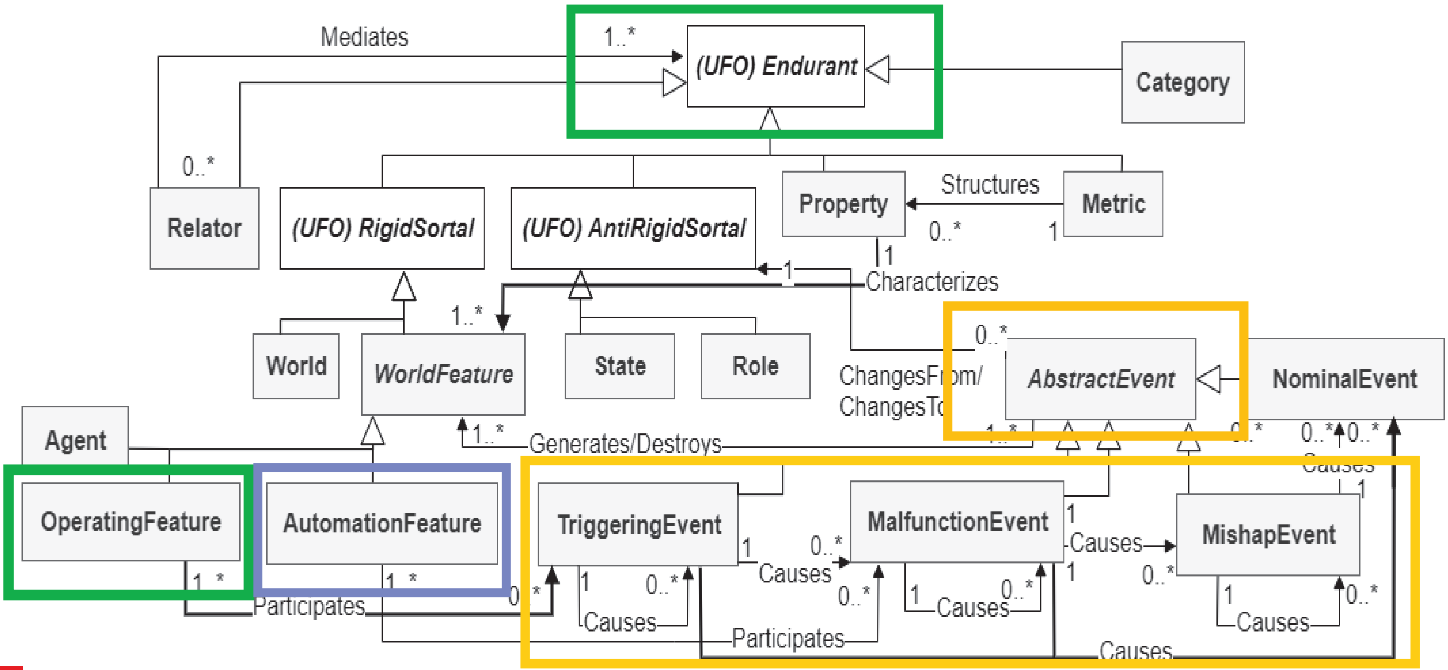
Example of automotive domain taxonomy



Example of UAV domain taxonomy

21

Ontology Language for the Dependability of Automated Systems (OLDAS)



22

Papyrus4ODD

- Tool to define Operational Design Domain (ODD)
  - Customization of Papyrus Eclipse (2023), an Eclipse based tool for Model Based Engineering, Aligned with **ISO 34503** ODD Taxonomy

Textual editor

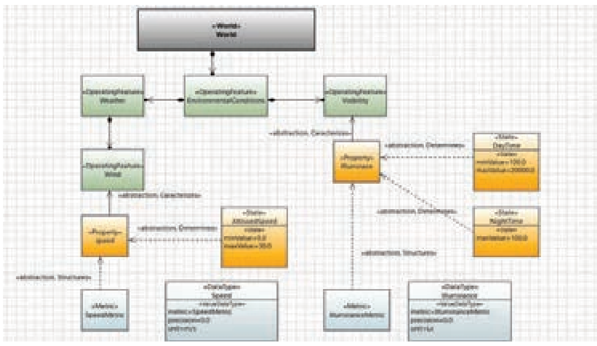
Implements OpenODD Domain-Specific Language

ODD_Example.odd

```
ADD Weather TO EnvironmentalConditions AS OperatingFeature
ADD Visibility TO EnvironmentalConditions AS OperatingFeature
ADD Illuminance TO Visibility AS Property
MESURE  IlluminanceMetric Illuminance Type Integer UNITS IlluminanceUnitKind.Lx
DETERMINE State DayTime WHEN [ 100 < Illuminance < 20000]
DETERMINE State NightTime WHEN [ Illuminance < 100]
ADD Wind TO Weather AS OperatingFeature
ADD Speed TO Wind AS Property
MESURE SpeedMetric Speed Type Real UNITS SpeedUnitKind.ms
DETERMINE State AllowedSpeed WHEN [ 0 < Speed < 30]
ACCEPT AllowedSpeed WHEN [DayTime]
```

Graphical editor

Implements an UML-based ontological language (OLDAS)



Papyrus4ODD

2. ...From ODD to Hazard analysis

25

"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.

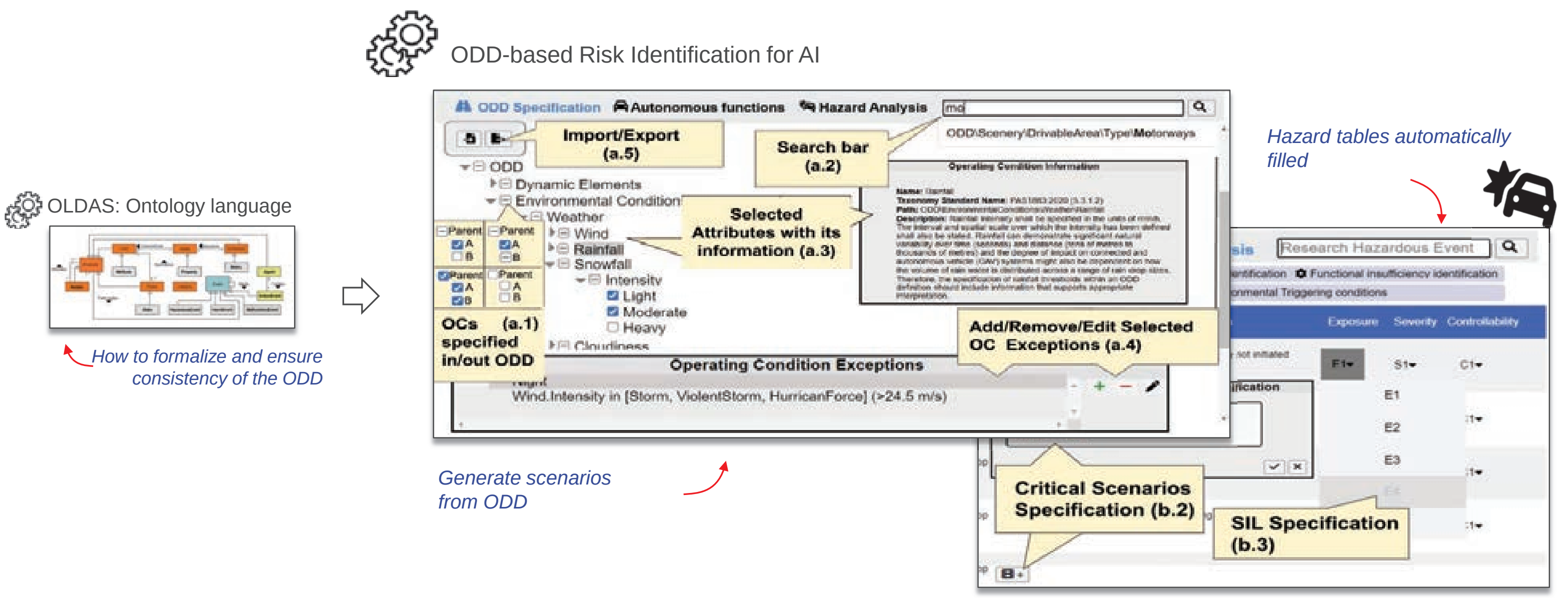
Safety of the Intended Functionality (SOTIF) concerns with guaranteeing the safety of a functionality that can have safety risks in the absence of a fault.

<< 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 >>

cea

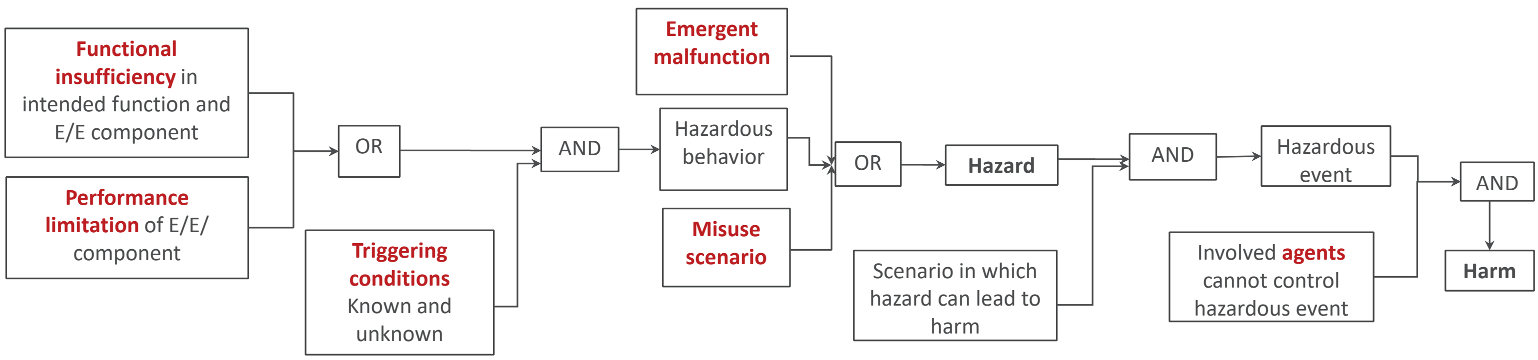
26

Approach: from ODD to Hazard Analysis



27

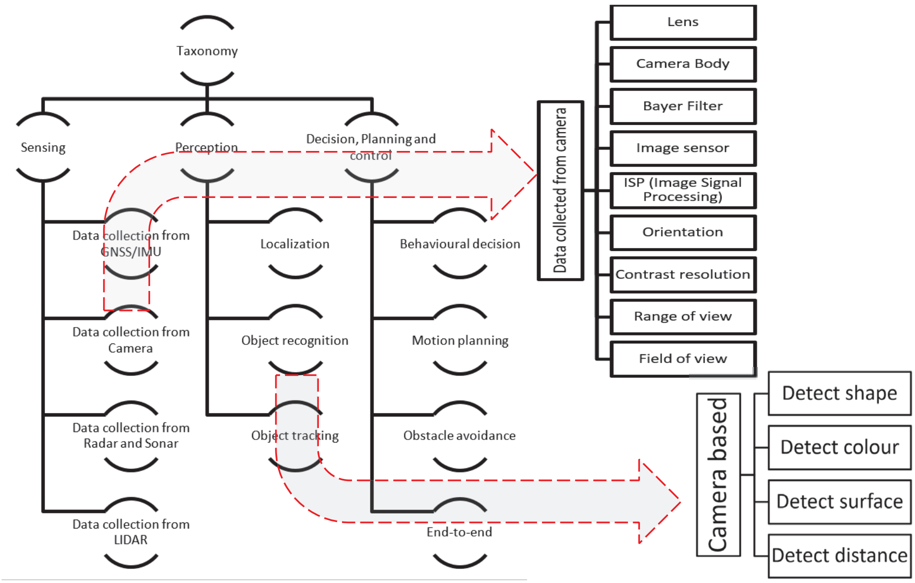
Discovering AI related Hazards



*Inspired by SOTIF

Discovering AI related Hazards

Information about possible hazards are identified based on System capabilities



⚠ Camera Weaknesses

- Detect shape
  - ➞ blur shape detection
- Detect colour
  - ➞ sudden light
  - ➞ weak light detection
- Detect surface
  - ➞ blur surface detection
- Detect distance
  - ➞ weak lighting conditions
- Lens
  - ➞ Occlusion
  - ➞ Reflection
- Range of view
  - ➞ reduced range of view
- Field of view
  - ➞ restricted field of view

29

Hazard analysis table example

About functional insufficiencies

Operating Conditions	In ODD?	Function	Acquisition	Interpretation/ processing	Decision	Keyword	Deviated function	Hazardous scenario	Consequence	Risk Classification
Weather: Clear **Time of day:** Day **Location:** Straight road, Minor road **Road condition:** Dry road, with pedestrian crossover **Vehicle Operation:** drive at low speed **Other road participants:** pedestrian is walking accross (from left)	yes	Perception	Not applicable	Object detection	Not applicable	Faulty	Unable to detect object due to faulty algorithm.	The perception of the environment is faulty while the vehicle is operating on a minor road at low speed, and a pedestrian crosses the lane in front of it from the left side of the road at a close distance.	Collision with pedestrian	Minor

scenario can (also) come from dataset analysis, expertise, accident database, …

30

A tool support for ODD-based Hazard Analysis

31

3. Drone Case Integrating System Design & ODDs

cea

32

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 (**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.
- The main components of an ontology are:
  - **Concepts**
  - **Individuals** that belong to these concepts
  - **Relations** between these

34

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 and SysML.

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

35

KRR integration into MBSE

To bridge the gap between **MBSE** tooling and **KRR**, we have identified the following challenges:

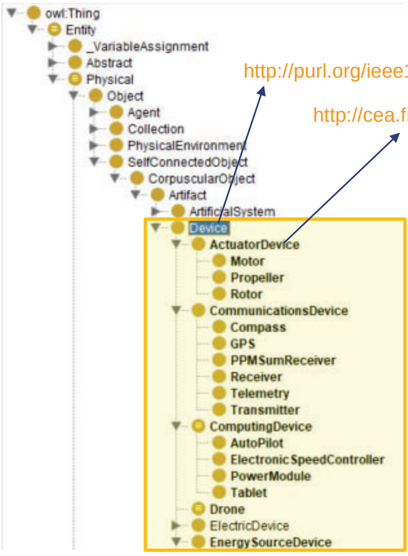
- ✓ The **terminology** in the ontology should be automatically made available to the designer, saving time and avoiding mistakes.
- ✓ The system's model designed in the MBSE tooling environment should be described in terms of this existing knowledge (ontology). In our context this means the annotation 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

Sources : CORA, CORAX, RPARTS, Dronetology, C4D

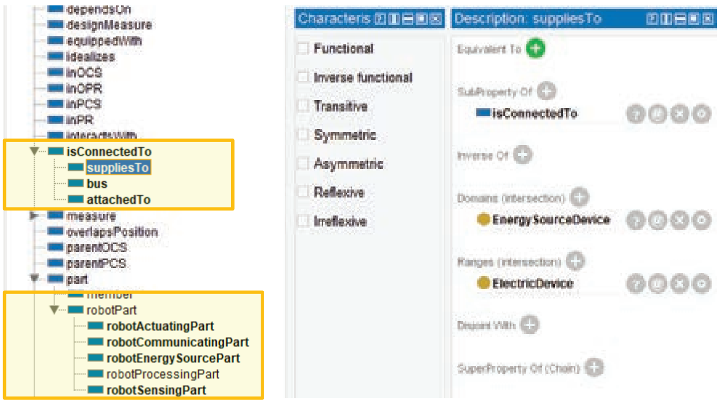
✓ We have developed an UAVs ontology (**ODrone**), and integrated it into IEEE1872-CORA[1], to provide a standardized formal domain specific vocabulary for UAVs.

✓ This ontology targets the physical components viewpoint.



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.

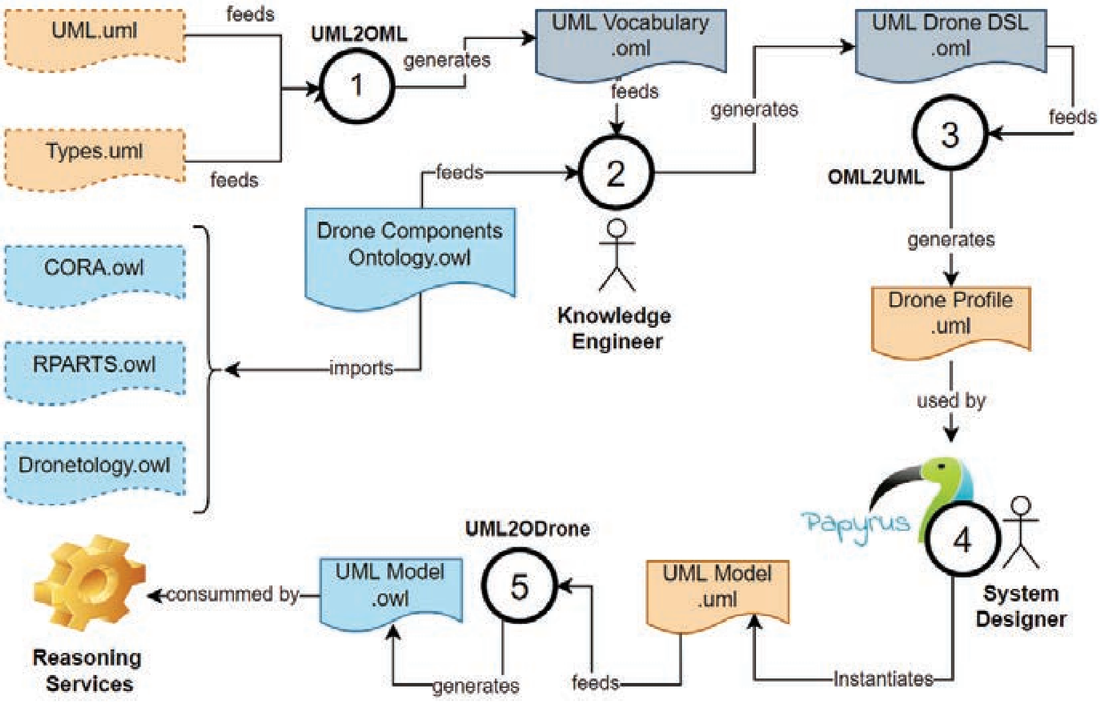


RPARTS (included in CORA) provides the roles of: *part, robotPart,* and their specializations for certain types of devices, e.g.: *robotActuatingPart, robotSensingPart*, etc.
We also extend it with relations to connect these parts.

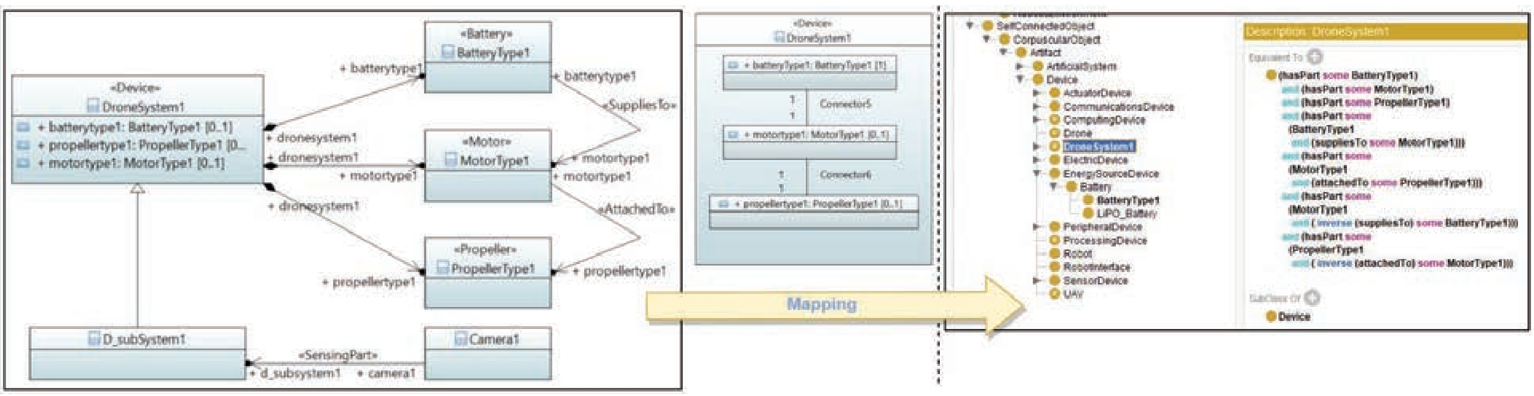
[1] https://github.com/srfiorini/IEEE1872-owl

37

KRR integration into MBSE – Approach Overview



UML Model to OWL Ontology (ODrone)

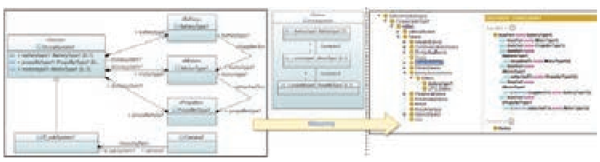


Reasoning over the model

Some examples of what reasoning can bring:

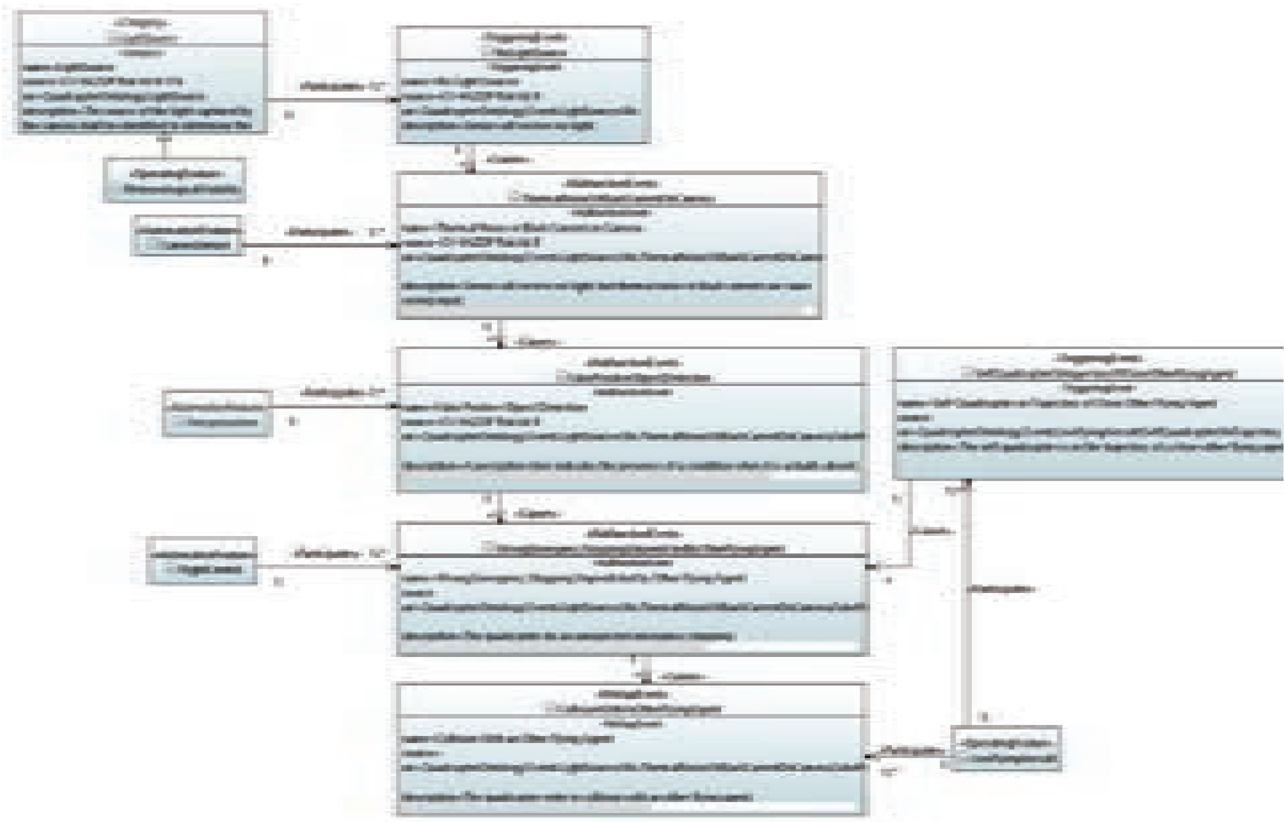
- ✓ What is a **Device** ?
  - ✓ Note that individuals c1,b1, m1, p1 and their classes Battery_Type1, Battery, Motor_Type1, etc.. are not Devices
  - ✓ We need to integrate ODrone (and CORA) to provide a context and definitions of what a Device means.
    - ✓ Note that once integrated, the individuals and classes above are inferred as devices.
    - ✓ Thanks to a Modular approach
    - ✓ We can "simply" import the ontologies, since ODrone *complies* with CORA
    - ✓ The UML model is annotated with ODrone

- ✓ Furthermore note that Camera_1 is a (CORA) Device.
  - ✓ This is a different inference. Nowhere in the model there is a hierarchical relation between Camera_1 and the Device class (via profiles)
  - ✓ This inference is possible, thanks to the **Sensing_Part** relation



40

Integrating System Design and ODDs

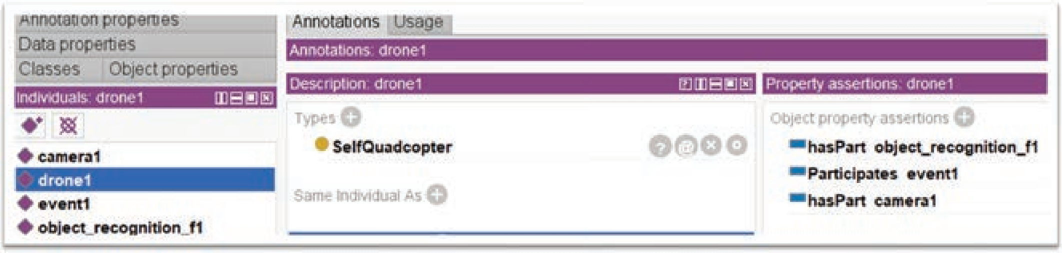


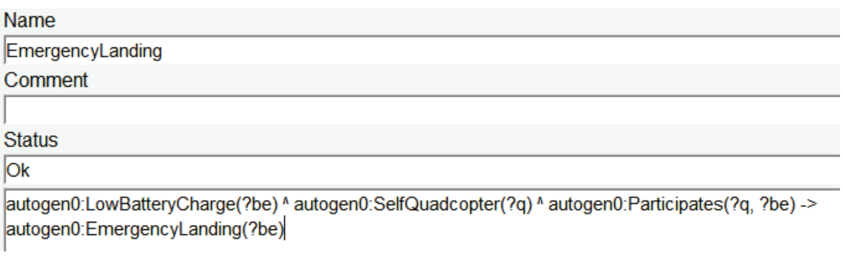
41

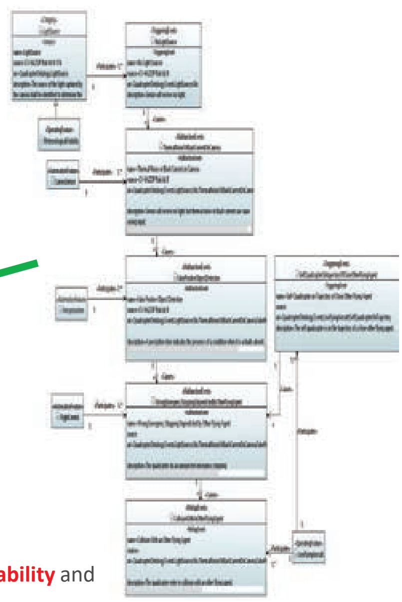
Integrating System Design and ODDs

Discovering AI related Hazards

Implemented in this case via SWRL rules (OLDAS+ODrone_1.owl):



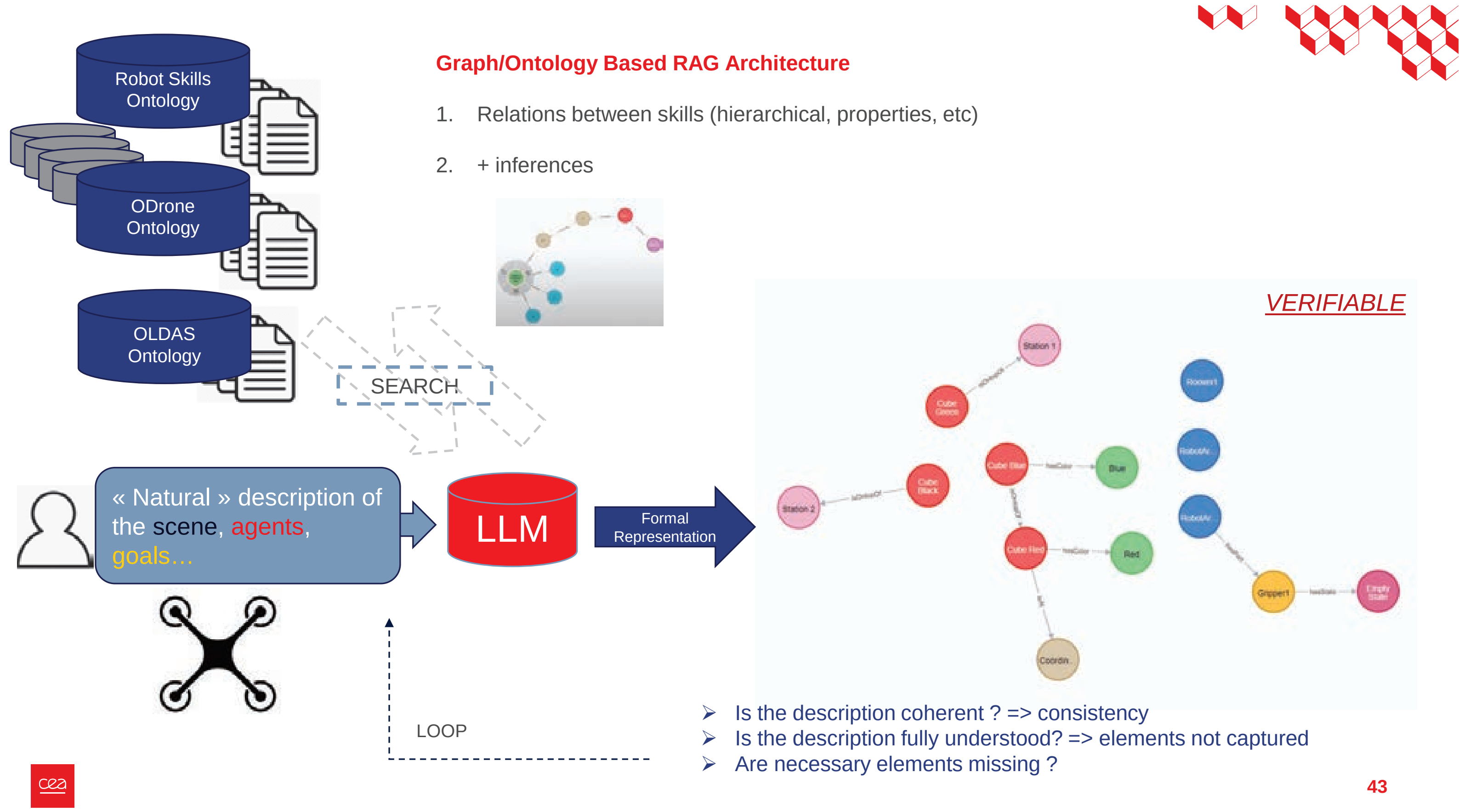




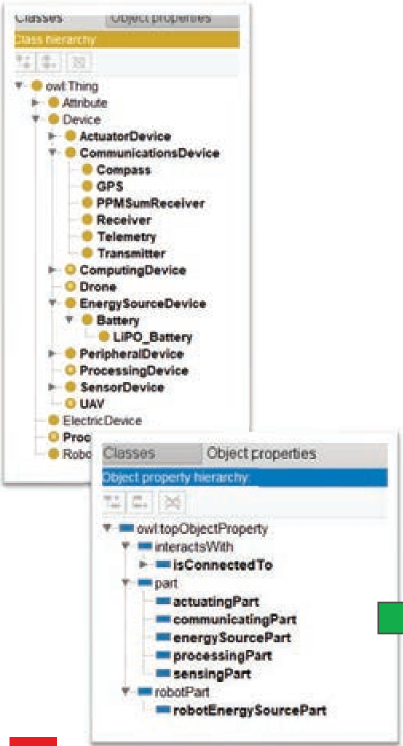
One of the **major challenges** is the selection of the right vocabulary, and the annotation process.

The coherence of the terminology among models coming from different sources, enables **semantic interoperability** and reasoning.

42


Graph/Ontology Based RAG Architecture
1. Relations between skills (hierarchical, properties, etc)
2. + inferences
Robot Skills Ontology
ODrone Ontology
OLDAS Ontology
SEARCH
« Natural » description of the scene, agents, goals…
LLM
Formal Representation
VERIFIABLE
Station 1
Cube Green
Cube Blue
Blue
Cube Black
Station 2
Cube Red
Red
Coordin…
Gripper1
Empty State
LOOP
➢ Is the description coherent ? => consistency
➢ Is the description fully understood? => elements not captured
➢ Are necessary elements missing ?
43

Ontology Selection and Automatic Annotation (assistance) - current work




```
[16]: # an example prompt
prompt = "Whta are the elements of a Devices and Drones"

# generate an embedding for the prompt and retrieve the most relevant doc
response = ollama.embeddings(
  prompt=prompt,
  model="mxbai-embed-large"
)
results = collection.query(
  query_embeddings=[response["embedding"]],
  n_results=1
)
data = results['documents'][0][0]
print(data)
```

```
Accelerometer, ActuatorDevice, Ampacity, AutoPilot, Battery, Camera, CellCount, CommunicationsDevic
e, Compass, ComputingDevice, Diameter, Dimensions, Drone, ElectronicSpeedController, EnergySourceDev
ice, GPS, Gimbal, Gyroscope, KV, LIDAR, LiPO_Battery, Magnetometer, Manufacturer, Mass, MaxContinuou
sCurrent, MaxVoltsPerPack, MinVoltsPerPack, Motor, PPMSumReceiver, PartNumber, PeripheralDevice, Pit
ch, PowerModule, Price, Propeller, RADAR, Receiver, RecommendedTrustRPM, Rotor, SONAR, SensorDevice,
SerialNumber, ShaftDiameter, Tablet, Telemetry, Transmitter, UAV, Voltage, actuatingPart, attachedT
o, bus, communicatingPart, energySourcePart, isConnectedTo, processingPart, robotEnergySourcePart, s
ensingPart, suppliesTo, hasAmperCapacity, hasDiameter, hasKV, hasMass, hasPitch, hasVoltage,
```

```
"The drone is equipped with an accelerometer to measure its acceleration and an actuator device for controlling its movements. Its battery provides the n
ecessary energy for flight, and a camera allows it to capture images and videos. The drone also has a GPS system for navigation and a compass to maintain
its direction. Additionally, it is equipped with a computing device for processing data and a gyroscope to maintain its stability. The drone's dimensions
are 10 cm in diameter and 20 cm in length, and it weighs approximately 1 kg. Its energy source is a LiPO_Battery, which has a voltage of 12V and an amper
acity of 5A. The drone also has a motor that provides the necessary power for flight, as well as a propeller for lift and thrust. Other components includ
e a transmitter, receiver, and telemetry system for communication, an electronic speed controller for regulating its speed, and a magnetometer for measur
ing its magnetic field. The manufacturer of the drone is XYZ Company, and it has a recommended trust RPM of 2000. The drone also has a PartNumber, a seri
al number, and a shaft diameter of 1 cm. It is connected to a bus and communicates with other devices through a peripheral device. The drone's energy sou
rce part is a LiPO_Battery, which has an KV of 10 and a mass of 200g. The robot energy source part is a LiPO_Battery, which has a voltage of 12V and an a
mperacity of 5A. The sensing part of the drone includes a camera, a GPS system, a compass, and a magnetometer. The drone also has a tablet for data proce
ssing and storage, as well as a SONAR device for measuring its distance from objects. The drone's price is $1000, and it is recommended to use it at a vo
ltage of 9V to 12V. Attached to the drone are an actuating part, which is the motor, and an attachedTo part, which is the propeller. The bus that the dro
ne communicates with is a PPMSumReceiver, and it supplies power to the drone through a communicatingPart, which is the energySourcePart. Additionally, th
e drone has a supplier of 5V DC power, and its hasAmperCapacity is 10A."

I hope this helps! Let me know if you have any further questions or need anything else.
```

44

cea list

Trustworthy AI: Industry-Guided Tooling of the Methods

Zakaria Chihani

zakaria.chihani@cea.fr

SS-CPSIoT

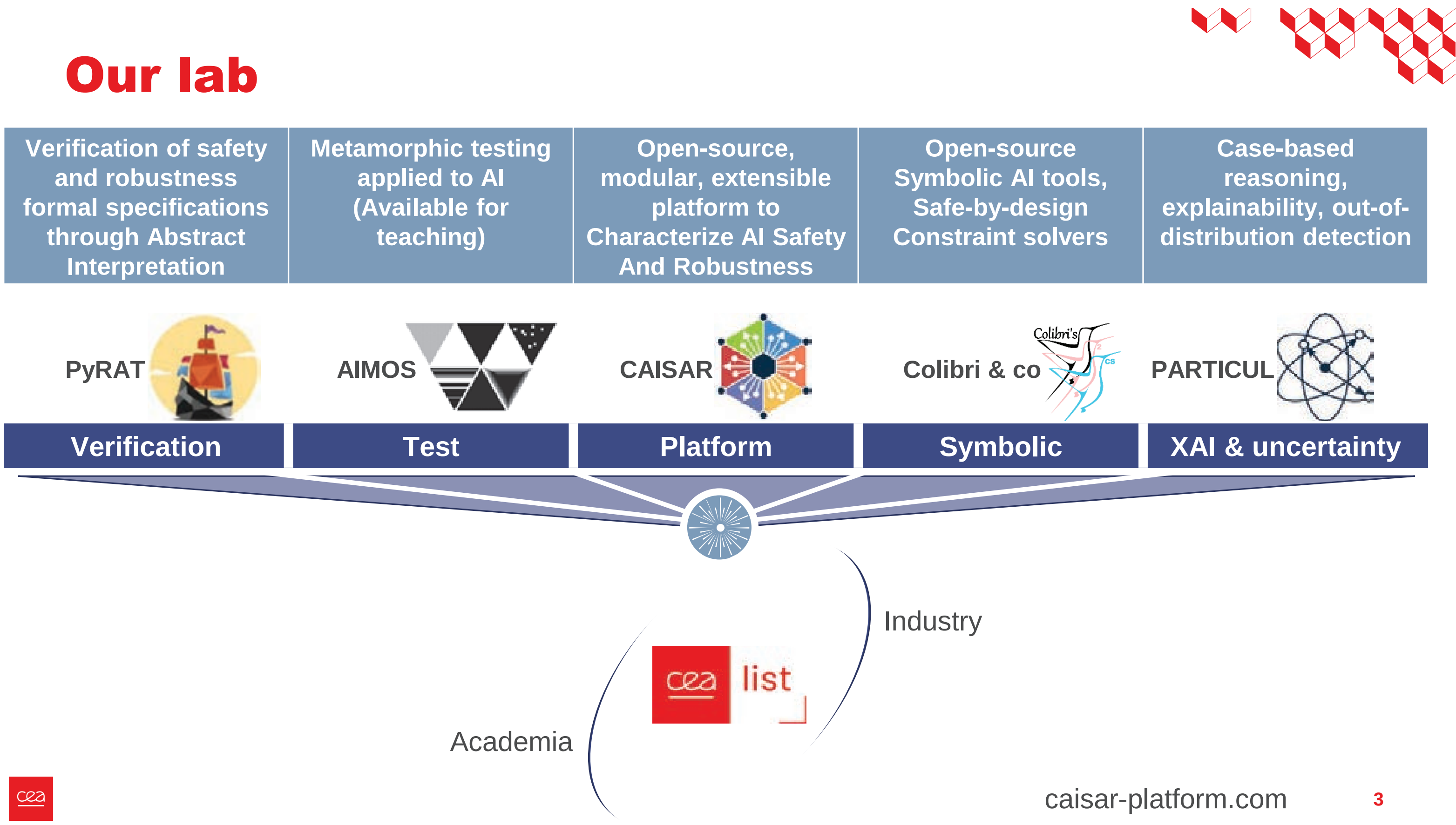
FRANCE 2030

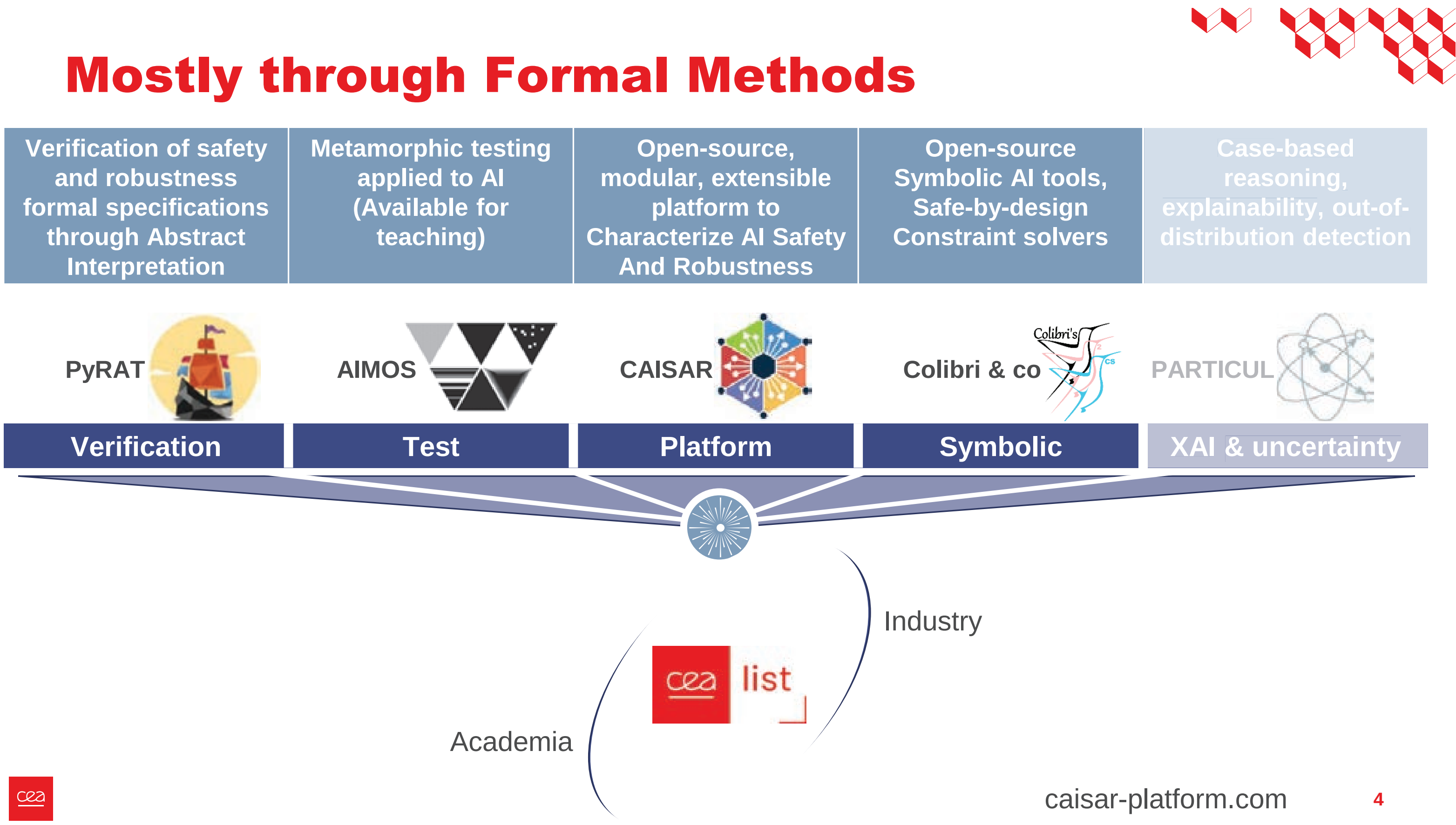
Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre de France 2030 portant la référence « ANR-23-PEIA-0006 »

Our lab

Verification of safety and robustness formal specifications through Abstract Interpretation	Metamorphic testing applied to AI (Available for teaching)	Open-source, modular, extensible platform to Characterize AI Safety And Robustness	Open-source Symbolic AI tools, Safe-by-design Constraint solvers	Case-based reasoning, explainability, out-of-distribution detection
PyRAT	AIMOS	CAISAR	Colibri & co	PARTICUL
Verification	**Test**	**Platform**	**Symbolic**	**XAI & uncertainty**

caisar-platform.com 2


Our lab
Verification of safety and robustness formal specifications through Abstract Interpretation
Metamorphic testing applied to AI (Available for teaching)
Open-source, modular, extensible platform to Characterize AI Safety And Robustness
Open-source Symbolic AI tools, Safe-by-design Constraint solvers
Case-based reasoning, explainability, out-of-distribution detection
PyRAT
AIMOS
CAISAR
Colibri & co
PARTICUL
Verification
Test
Platform
Symbolic
XAI & uncertainty
Industry
Academia
cea list
caisar-platform.com
3


Mostly through Formal Methods
Verification of safety and robustness formal specifications through Abstract Interpretation
Metamorphic testing applied to AI (Available for teaching)
Open-source, modular, extensible platform to Characterize AI Safety And Robustness
Open-source Symbolic AI tools, Safe-by-design Constraint solvers
Case-based reasoning, explainability, out-of-distribution detection
PyRAT
AIMOS
CAISAR
Colibri & co
PARTICUL
Verification
Test
Platform
Symbolic
XAI & uncertainty
Industry
Academia
cea list
caisar-platform.com
4

Rapid intro to FM

Examples for this talk:

- **Property-based testing (Renault)**
- **Verification of functional properties (Airbus)**
- **Robustness evaluation (Technip)**
- **Out-of-Distribution detection (Thales)**

cea 5

So what are Formal Methods

- Non snobbish definition
  - Math- and logic-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

cea

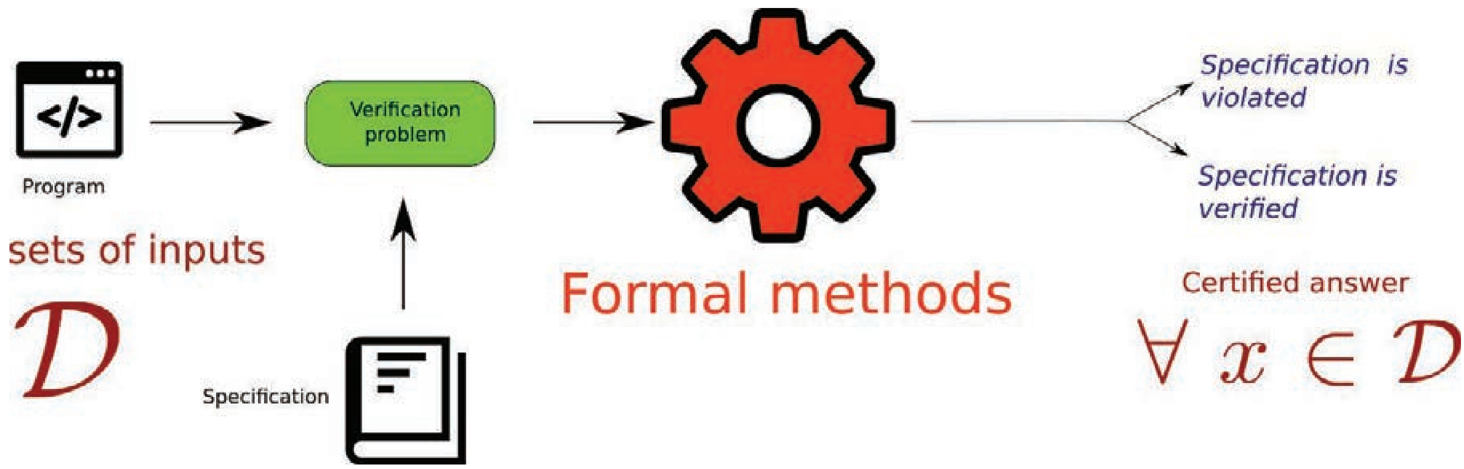
6

So why 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

So HOW Formal Methods (usually)



So WHO Formal Methods

Astrée Software

PolySpace Technologies

frama C Software Analyzers

ESTEREL Technologies

CompCert

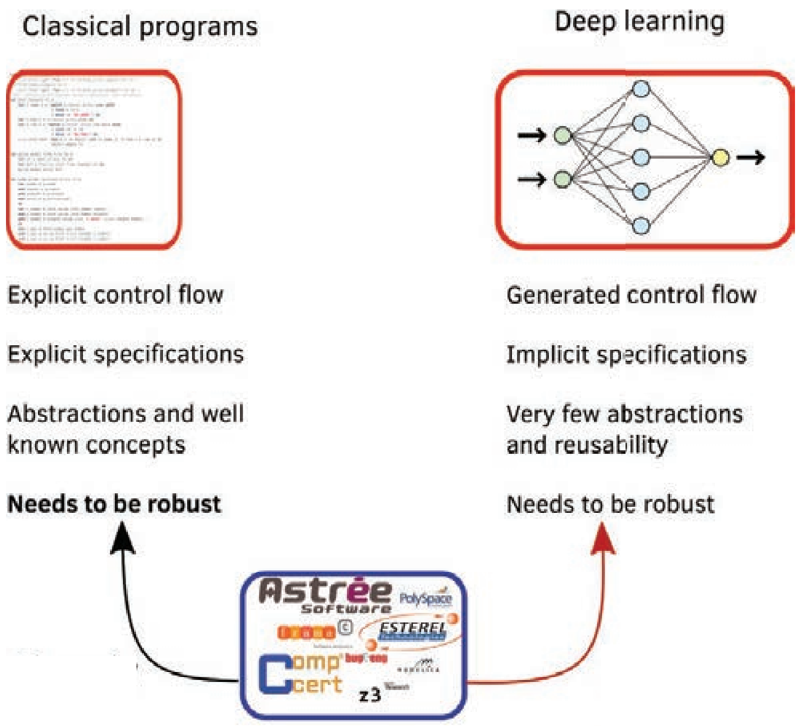
bugseng

MODELICA

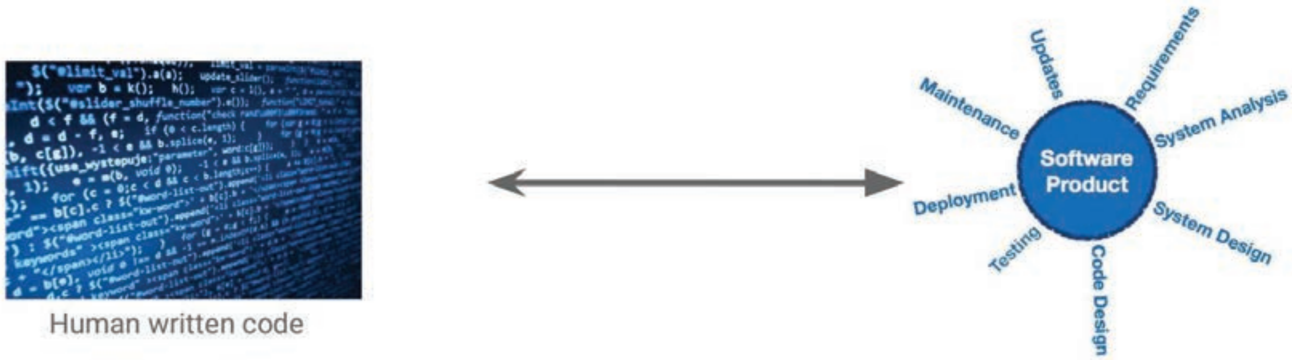
z3 Microsoft Research

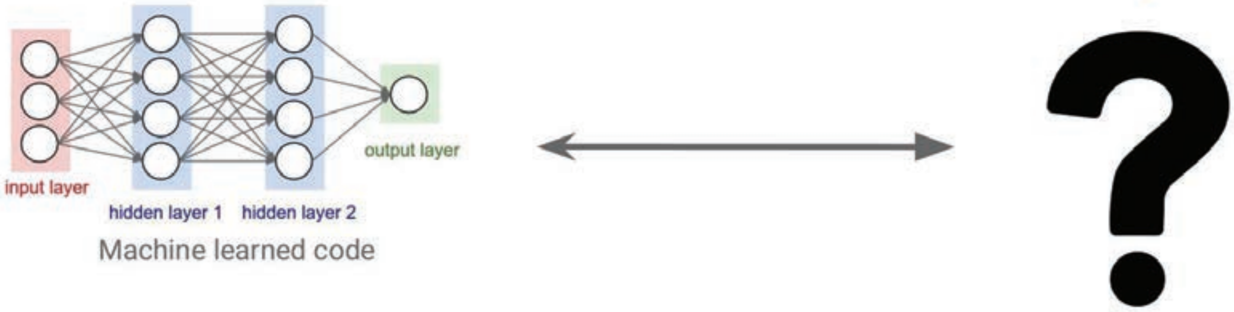
So for WHOM Formal Methods

So WHICH Formal Methods



So WHICH Formal Method for AI





A brief history of wrong predictions

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.

A brief history of wrong predictions

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"

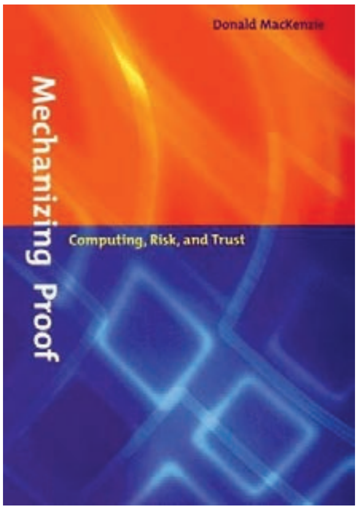
"Social processes and proofs of theorems and programs", Communications of ACM.
By Richard De Millo, Richard Lipton, and Alan Perlis.

- 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

Valid scepticism



- 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…

Restarting Formal Methods for 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.

cea

Restarting Formal Methods for 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)

General picture

Characterization of (AI) trustworthiness

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

Rapid intro to FM

Examples for this talk:

- **Property-based testing (Renault)**
- **Verification of functional properties (Airbus)**
- **Robustness evaluation (Technip)**
- **Out-of-Distribution detection (Thales)**

20

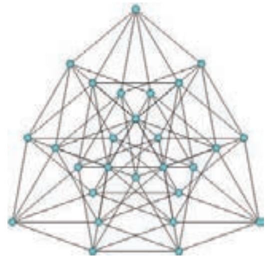
Metamorphic testing

Ideal setting for testing:

- Collection of inputs
- Corresponding collection of outputs

Metamorphic testing is used when:

- You don't know the actual **answer** (no oracle)
- But you know what **properties** should be satisfied by the inputs/outputs



Let
L (V,V) → int
be the length of the shortest path between two vertices, then :

L(a,b) = L(b,a)
For any two points a, b in a graph.

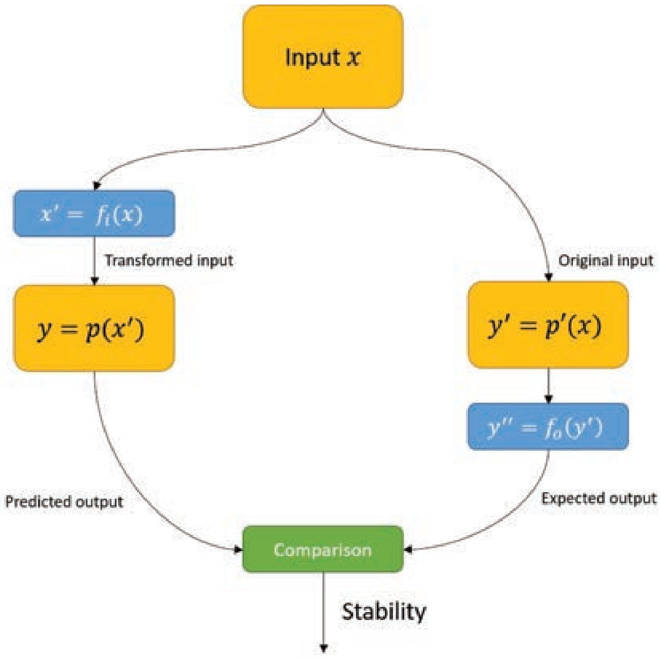
Input **symmetry**
↕
output **equivalence**

Metamorphic testing applied to AI : AIMOS

AIMOS (Artificial Intelligence Metamorphic Observing Software) is a tool to assess the stability of AI systems using metamorphic testing.

- No need to label data for testing.
- Automates the entire process of applying metamorphic properties on the inputs and outputs of models, comparing them and compiling the results into a stability score.
- Model agnostic (Neural Networks, Support Vector Machines, *etc.*).

Metamorphic testing applied to AI : AIMOS



caisar-platform.com

23

Metamorphic testing applied to AI : AIMOS
Easy to use

- Written in Python
- Model agnostic: only the inference functions are needed.
- Built-in support for various frameworks, input formats and model types.
- Built-in classical transformations (rotation, noise, symmetry, *etc.*).

Metamorphic testing applied to AI : AIMOS
Easy to use

- With a configuration file

```yaml
options:
  plot: True
  inputs_path: "inputs"
  transformations:
    - name: "gaussian_blur"
      fn_range: range(1, 10, 2)

models:
  - defaults:
      models_path: "models/model.onnx"
```

caisar-platform.com

25

Metamorphic testing applied to AI : AIMOS
Easy to use

- With a configuration file
- As a Python library

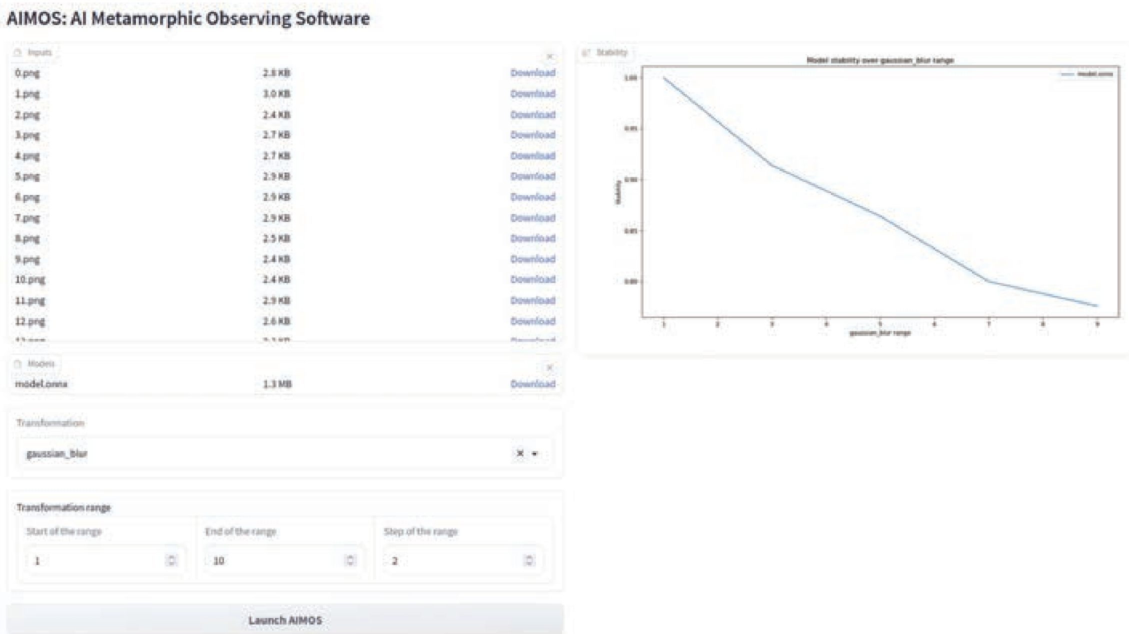
```
from aimos import core

core.main(
    "./inputs",
    "./models/model.onnx",
    "average_blur",
    fn_range=range(1, 10, 2),
    plot=True,
)
```

caisar-platform.com

26

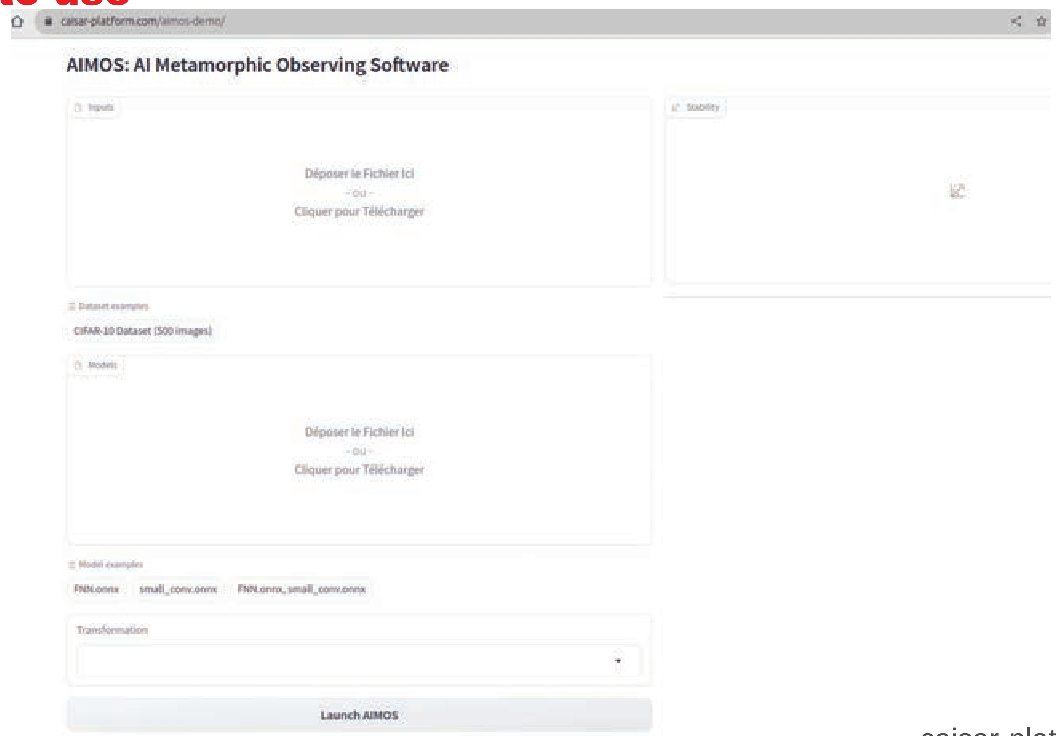
Metamorphic testing applied to AI : AIMOS
Easy to use

- With a configuration file
- As a Python library
- With a Graphical User Interface



caisar-platform.com

Metamorphic testing applied to AI : AIMOS
Easy to use



caisar-platform.com

28

Metamorphic testing applied to AI : AIMOS
Modular and extensible

Any operation can be replaced with a custom made Python function (loading the model, the inputs, new metrics, *etc.*).

```
def dead_columns(input, columns=np.uint8([50, 100, 150])):
    """ Adds dead pixel columns to an image. """
    input[:, columns, :] = 0
    return input
```

Metamorphic testing applied to AI : AIMOS

AIMOS is a tool that can be integrated in the verification and validation process of AI-based components.

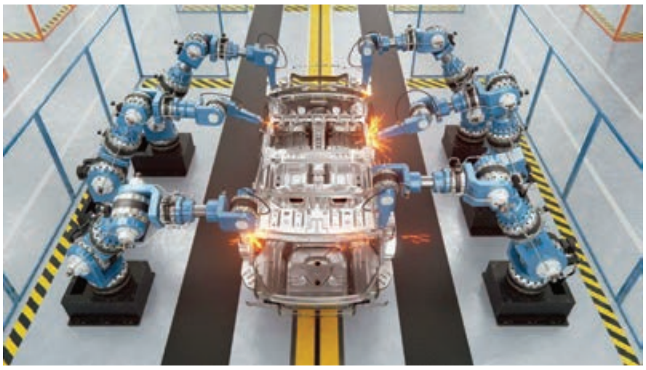
- Freely available for teaching and research purposes.
- Integrated in CAISAR, an open-source platform for characterizing safety in AI systems.

caisar-platform.com

Metamorphic testing applied to AI : AIMOS

The use-case

- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert

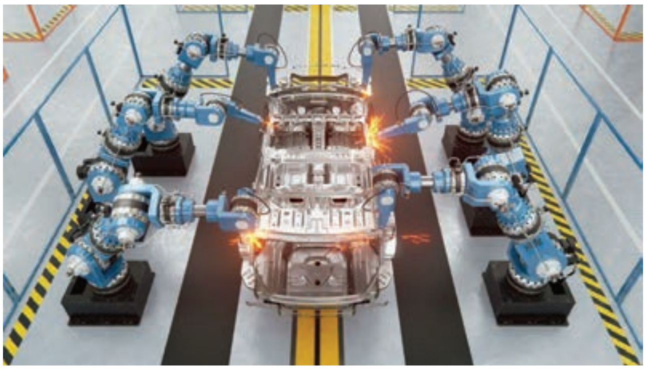


31

Metamorphic testing applied to AI : AIMOS

The use-case

- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert



32

Metamorphic testing applied to AI : AIMOS

The use-case

- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert

- 3 different production lines called C10, C20 and C34 and their corresponding weld.
- 5 AutoML models and 1 internal R&D composit model (NN+SVM) per production line.

Lemesle, A., Varasse, A., Chihani, Z., Tachet, D. (2023). **AIMOS: Metamorphic Testing of AI - An Industrial Application**. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_27

33

Metamorphic testing applied to AI : AIMOS

The use-case

- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert

- 3 different production lines called C10, C20 and C34 and their corresponding weld.
- 5 AutoML models and 1 internal R&D composit model (NN+SVM) per production line.

The environment => ODD => properties

- Day light changes + human workers pass by light sources => Robustness to **varying brigthness**
- Vibrating environment => Robustness to **blurring**

Lemesle, A., Varasse, A., Chihani, Z., Tachet, D. (2023). **AIMOS: Metamorphic Testing of AI - An Industrial Application**. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_27

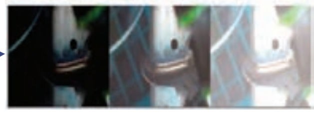
cea

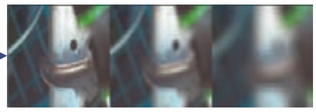
34

Metamorphic testing applied to AI : AIMOS

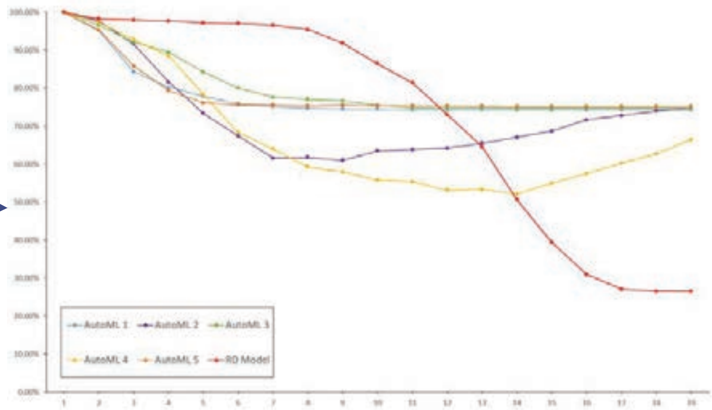
Metamorphic properties

AI Models





Representative dataset



35

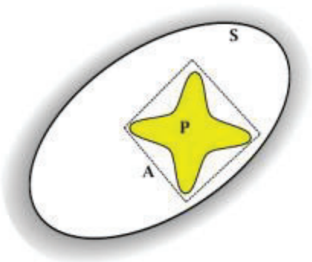
Rapid intro to FM

Examples for this talk:

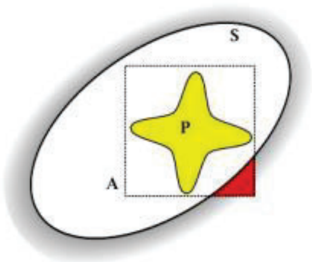
- Property-based testing (Renault)
- Verification of functional properties (Airbus)
- Robustness evaluation (Technip)
- Out-of-Distribution detection (Thales)

36

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

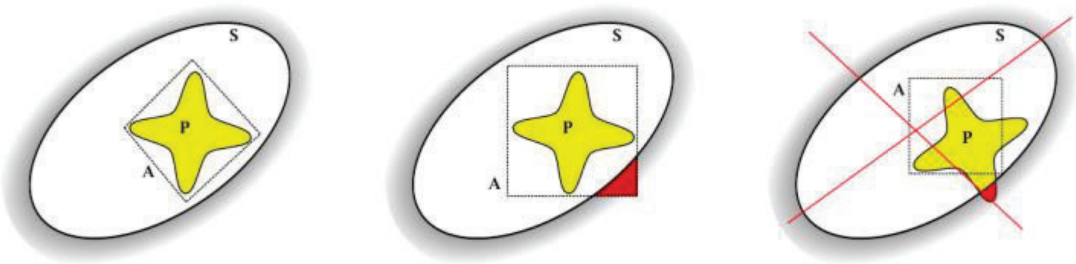


precise analysis
$A \subseteq S \implies P \subseteq S$



false alarm
$A \not\subseteq S$ but $P \subseteq S$

PyRAT: Python Reachability Assessment Tool Based on 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 $P \not\subseteq S$

PyRAT: Python Reachability Assessment Tool Based on 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).

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

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;               ..............
}
```

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

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..
}		

41

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

Property: “f(y)=100 → Critical vibration frequency ”

		Concrete	Intervals
f (int y){			
int x;	....	..-2,-1,0,1,2,..	-∞,+∞
x = 3 * (y²+1);	....	3,6,9,..	3,+∞
if x > 100 then	....	102,105,108,..	102,+∞
x = x + 10;	....	**112,115,118**,..	**112,+∞**
else	....	3,6,9..93,96,99	3,99
x = x - 2;	....	**1,4,7,..91,94,97**	**1,97**
return x;	....	1,4,..94,97,112,115..	1,+∞
}			

42

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

Property: "f(y)=100 → Critical vibration frequency "

	Concrete	Intervals	modulo
f (int y){			
int x;	..-2,-1,0,1,2,..	-∞,+∞	0%1
x = 3 * (y²+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
}			

cea

43

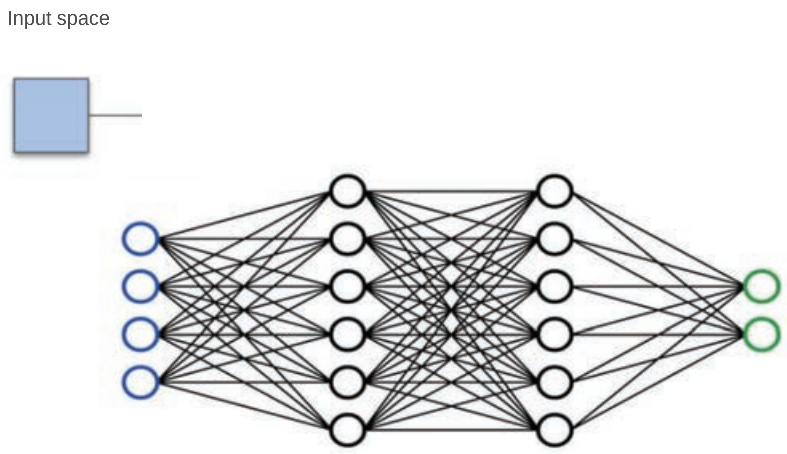
PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

Property: "f(y)=100 → Critical vibration frequency "

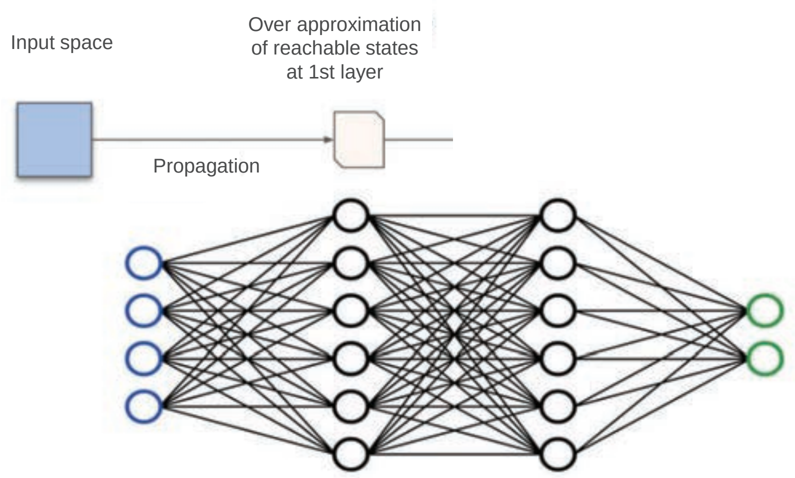
	Concrete	Intervals	modulo	Union of intervals
f (int y){				
int x;	..-2,-1,0,1,2,..	-∞,+∞	0%1	-∞,+∞
x = 3 * (y²+1);	3,6,9,..	3,+∞	0%3	3,+∞
if x > 100 then	102,105,108,..	102,+∞	0%3	102,+∞
x = x + 10;	**112,115,118**,..	**112,+∞**	1%3	**112,+∞**
else	3,6,9..93,96,99	3,99	0%3	3,99
x = x - 2;	**1,4,7,..91,94,97**	**1,97**	1%3	**1,97**
return x;	1,4,..94,97,112,115..	1,+∞	1%3	*[1,97]∪ [112,+∞[*
}				

Conservative over-approximation: the concretization of the abstract domains contains reality
The inverse is not necessarily true. 1,4..94,97,**100,103,106,109**,112,115..

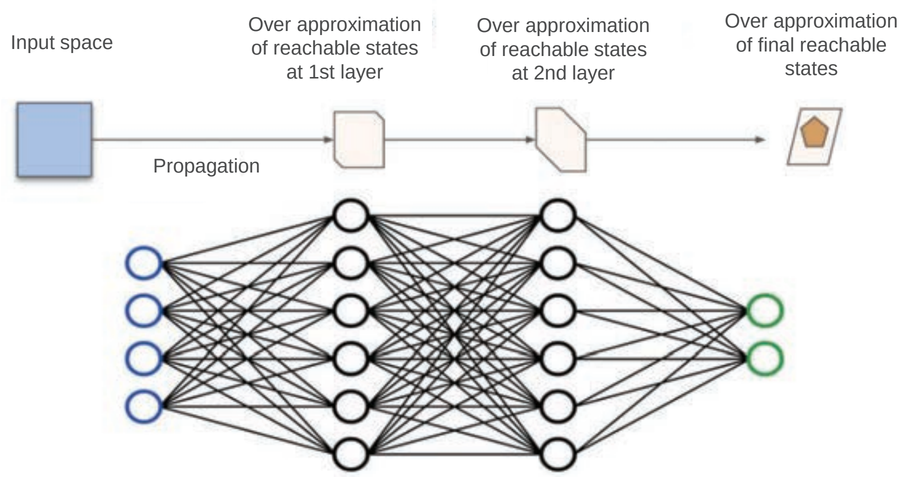
PyRAT: Python Reachability Assessment Tool
Application to NN



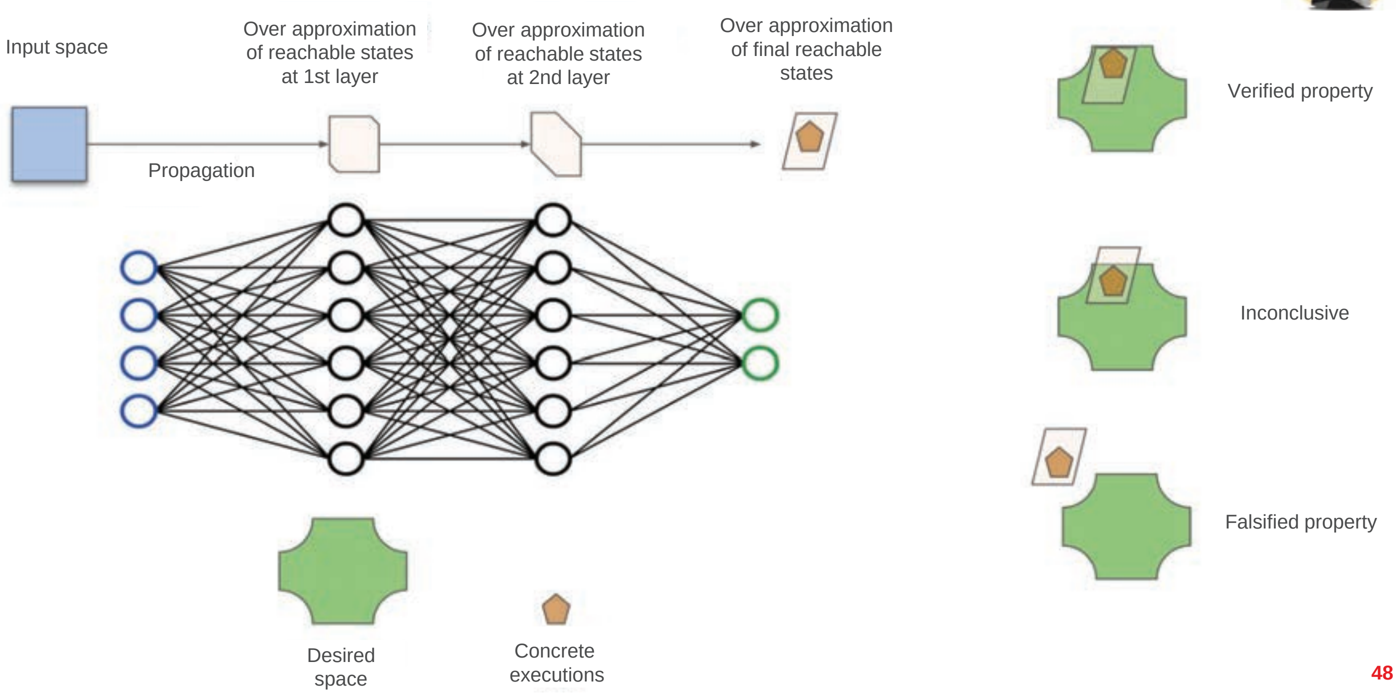
PyRAT: Python Reachability Assessment Tool
Application to NN



PyRAT: Python Reachability Assessment Tool
Application to NN

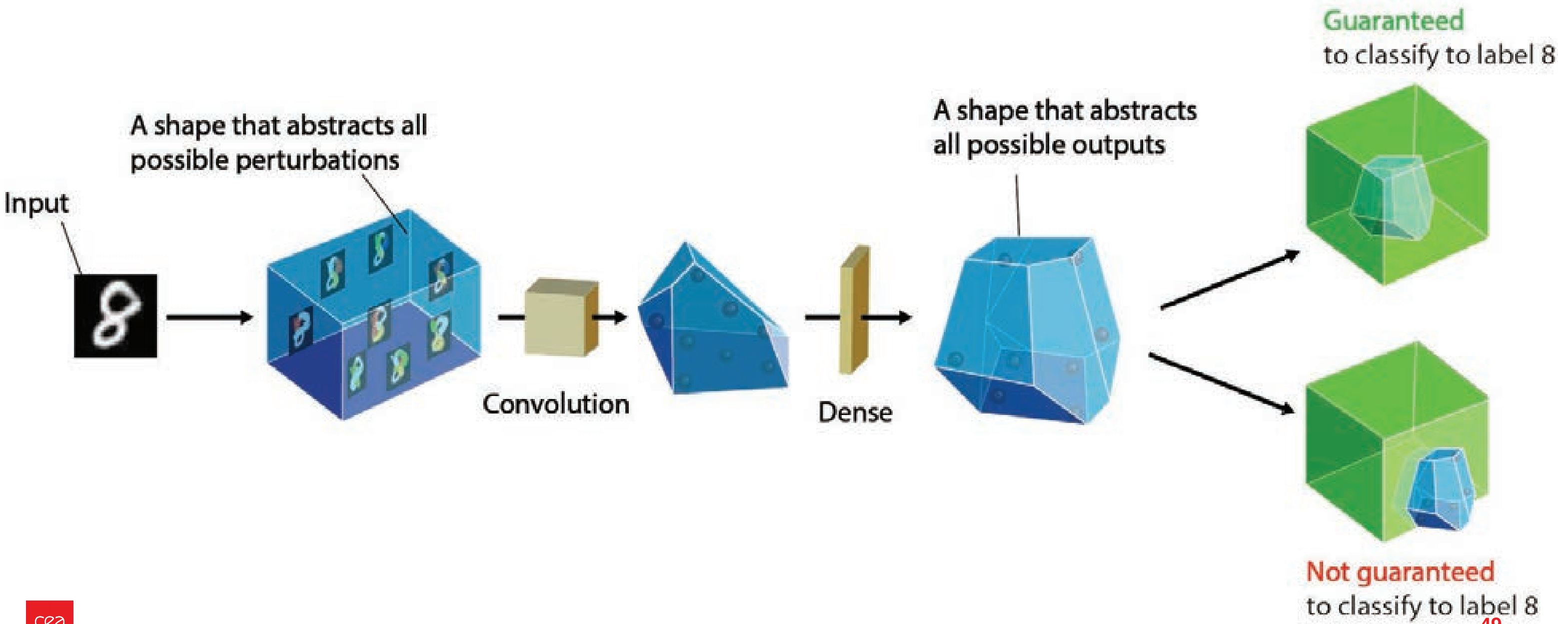


PyRAT: Python Reachability Assessment Tool Application to NN



48

PyRAT: Python Reachability Assessment Tool Application to NN



49

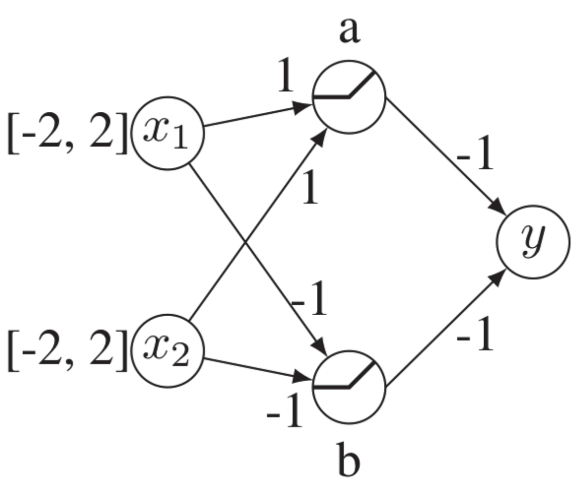
PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

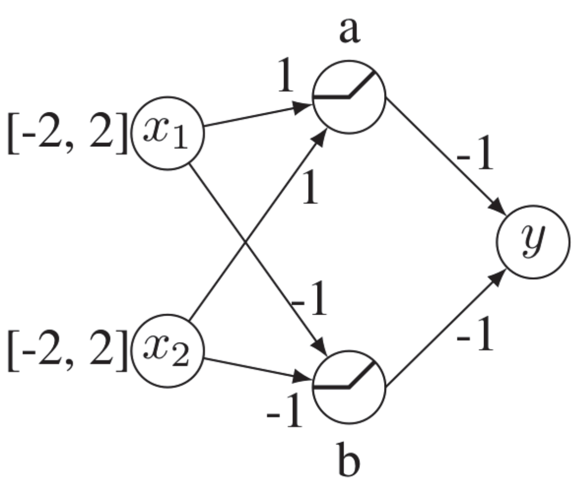


$\boxed{\textbf{Prove that } y > -5}$

PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$
$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$
$$y = -a_{\text{out}} - b_{\text{out}}$$



$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

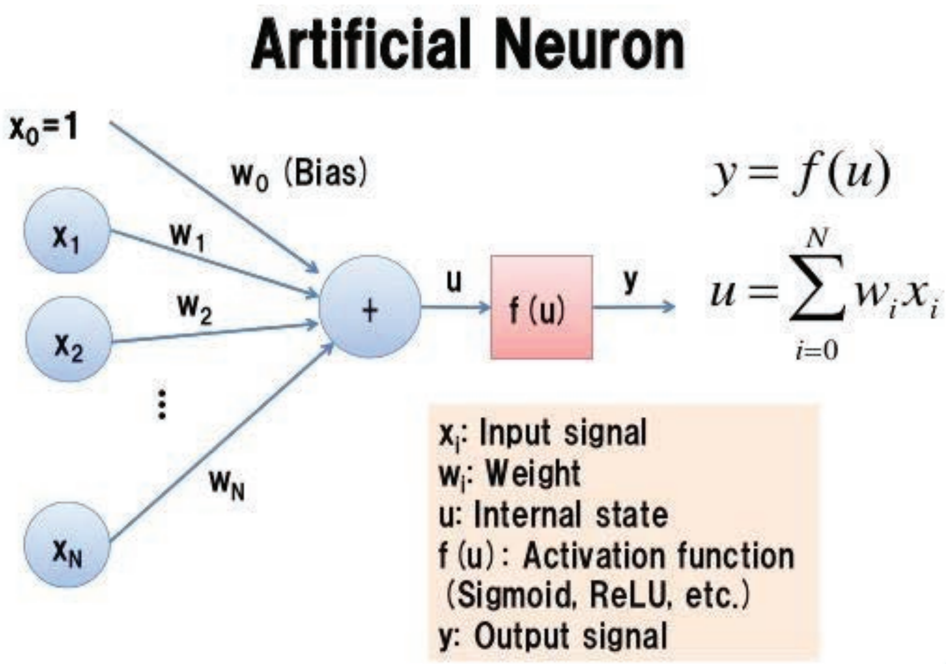
$a_{\text{out}} =$

$b_{\text{out}} =$

$y =$

$\boxed{\textbf{Prove that } y > -5}$

PyRAT: Python Reachability Assessment Tool
Application to NN

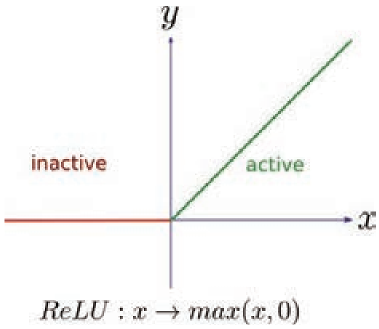


52

PyRAT: Python Reachability Assessment Tool
Application to NN

$$a_{\text{out}} = \max(a_{\text{in}}, 0)$$

	a_{in}	a_{out}
$a_{\text{in}} = x_1 + x_2$	[-4,4]	
if a_{in} > 0 then	]0,4]	
$a_{\text{out}} = a_{\text{in}}$		]0,4]
else	[-4,0]	
$a_{\text{out}} = 0$		[0,0]
		[0,4]



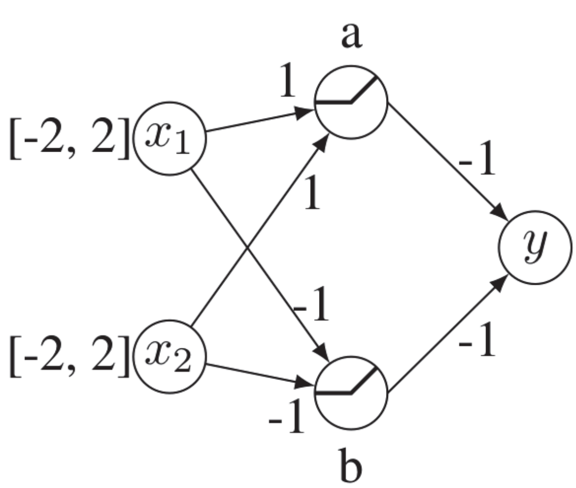
PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$



$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$

$b_{\text{out}} =$

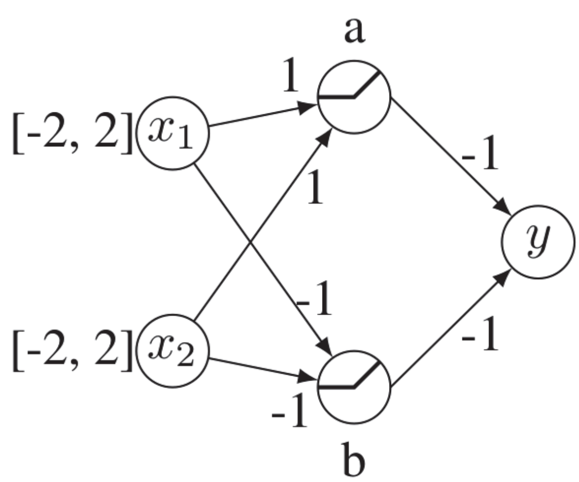
$y =$

$\boxed{\textbf{Prove that } y > -5}$

PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$
$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$
$$y = -a_{\text{out}} - b_{\text{out}}$$



$a_{\text{in}} =$ [-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]

$y =$

$\boxed{\textbf{Prove that } y > -5}$

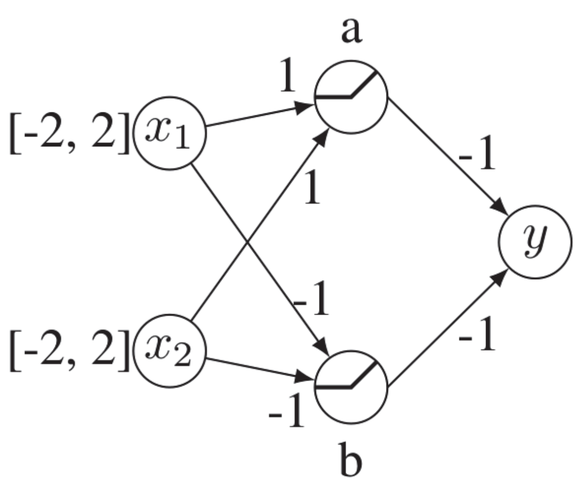
PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$



$a_{\text{in}} =$ [-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]

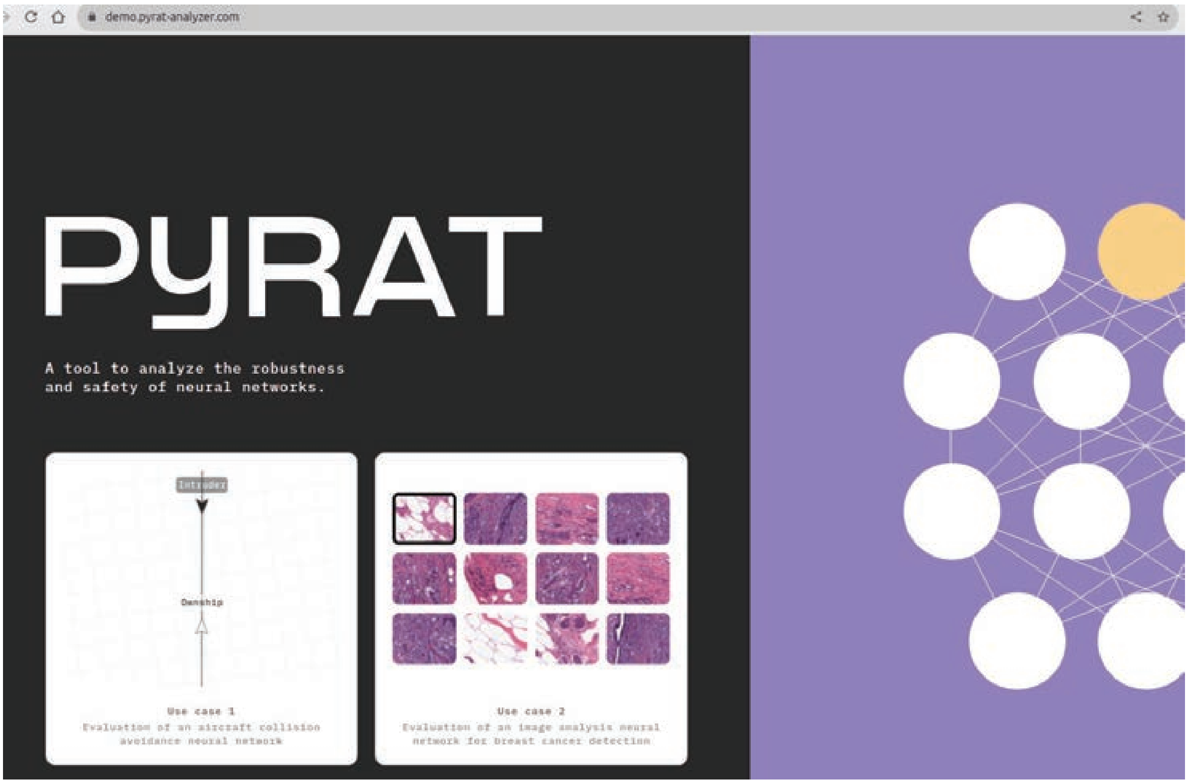
$\boxed{\textbf{Prove that } y > -5}$

PyRAT: Python Reachability Assessment Tool

- 3rd at VNNComp 2023
- Written in Python with PyTorch and Numpy backend
- Supports common layers and architecture in ONNX, Keras/Tensorflow and PyTorch
- Different abstract domains implemented: Box, Zonotopes, Constrained Zonotopes, ...
- Integrated in CAISAR, an open-source platform for characterizing safety in AI systems.

caisar-platform.com

PyRAT: Python Reachability Assessment Tool



58

Rapid intro to FM

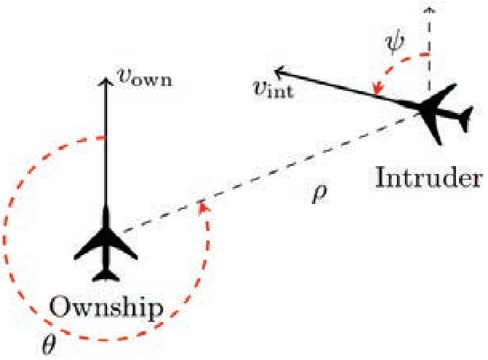
Examples for this talk:

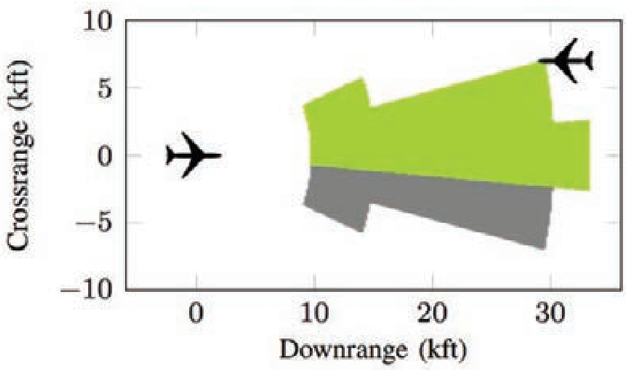
- Property-based testing (Renault)
- Verification of functional properties (Airbus)
- Robustness evaluation (Technip)
- Out-of-Distribution detection (Thales)

cea

59

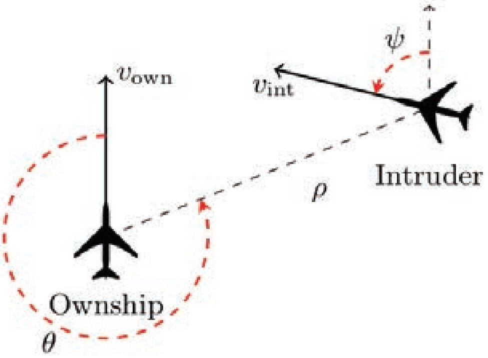
PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

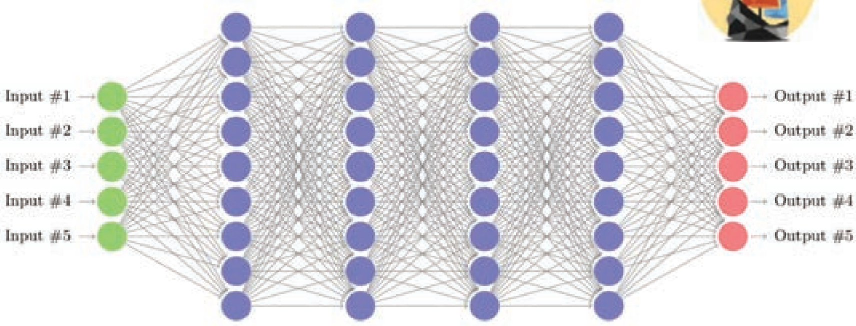


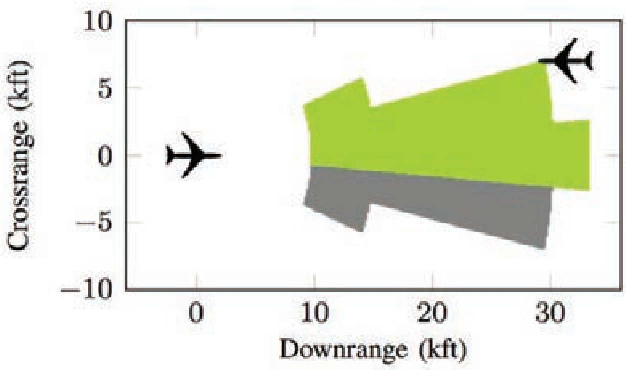


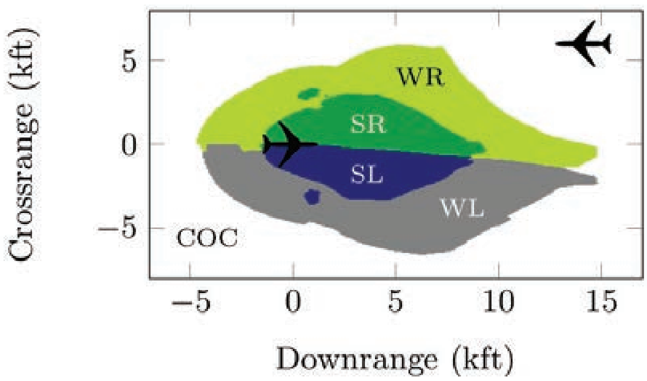
60

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System









61

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

Input nodes
x1 ρ
x2 θ
x3 ψ
x4 v own
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.

62

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

Description: If the intruder is near and approaching from the left, the network advises "strong right".

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

Description: If the intruder is near and approaching from the left, the network advises "strong right".

Input constraints: $250 \leq \rho \leq 400$, $0.2 \leq \theta \leq 0.4$, $-3.141592 \leq \psi \leq -3.141592 + 0.005$, $100 \leq v_{\text{own}} \leq 400$, $0 \leq v_{\text{int}} \leq 400$.

cea

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

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.

Rapid intro to FM

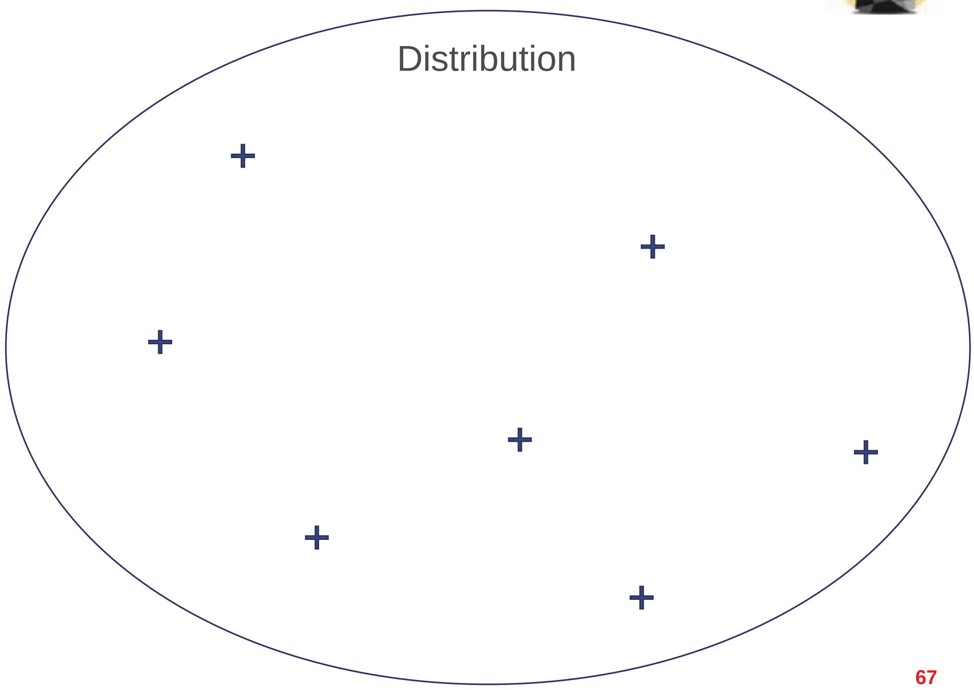
Examples for this talk:

- **Property-based testing (Renault)**
- **Verification of functional properties (Airbus)**
- **Robustness evaluation (Technip)**
- **Out-of-Distribution detection (Thales)**

66

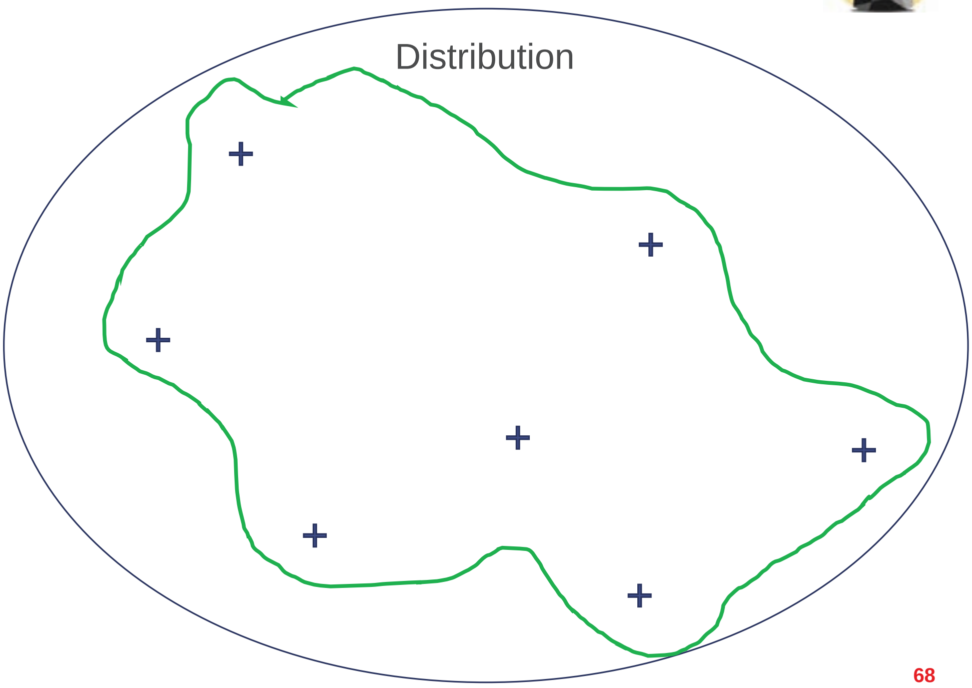
PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

Ideally the selected data to build and validate the model is representative of the intended distribution.



PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

Ideally the selected data to build and validate the model is representative of the intended distribution.

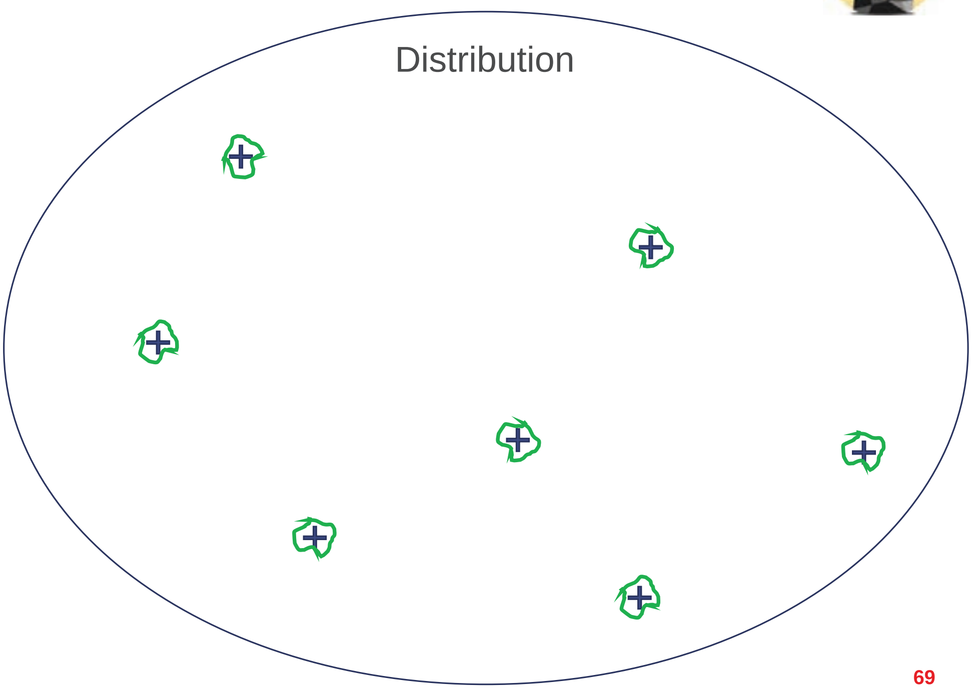


68

PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

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

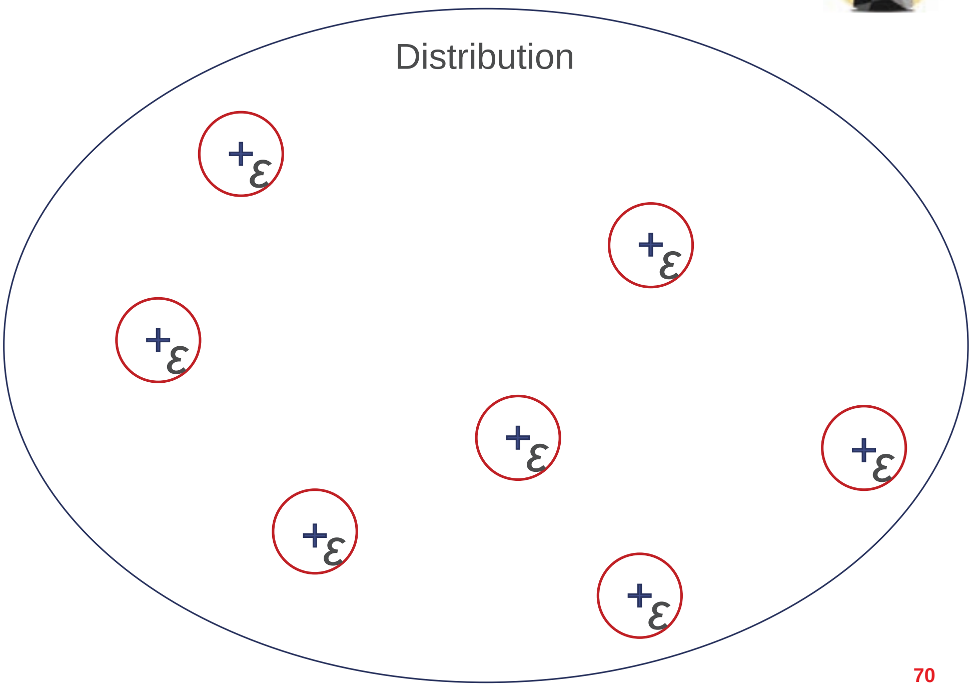


PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

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?



PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

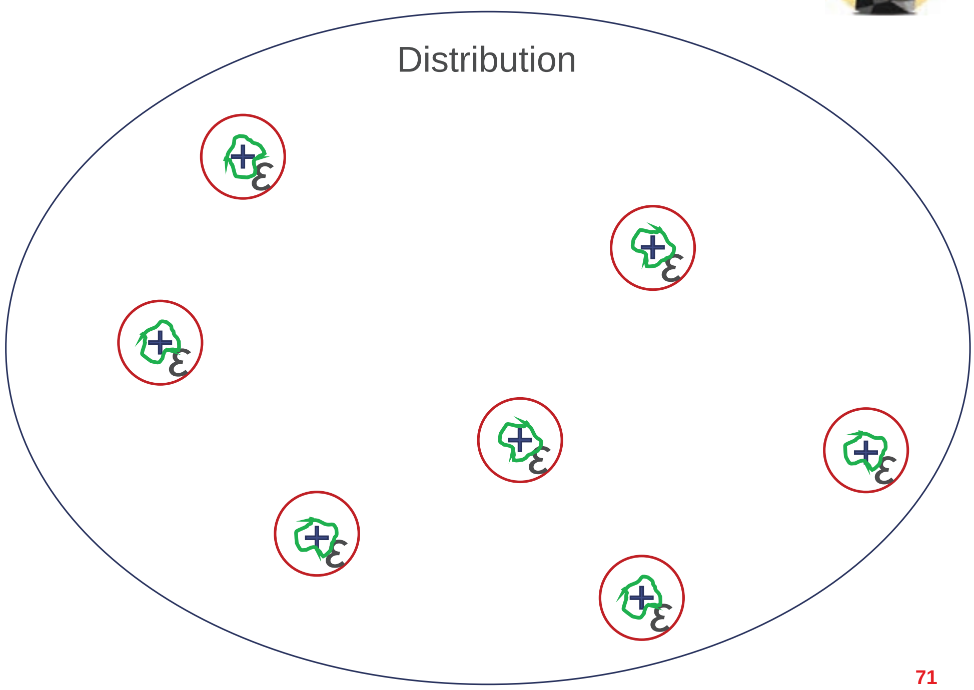
Ideally the selected data to build and validate the model is representative of the intended distribution.

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…



PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

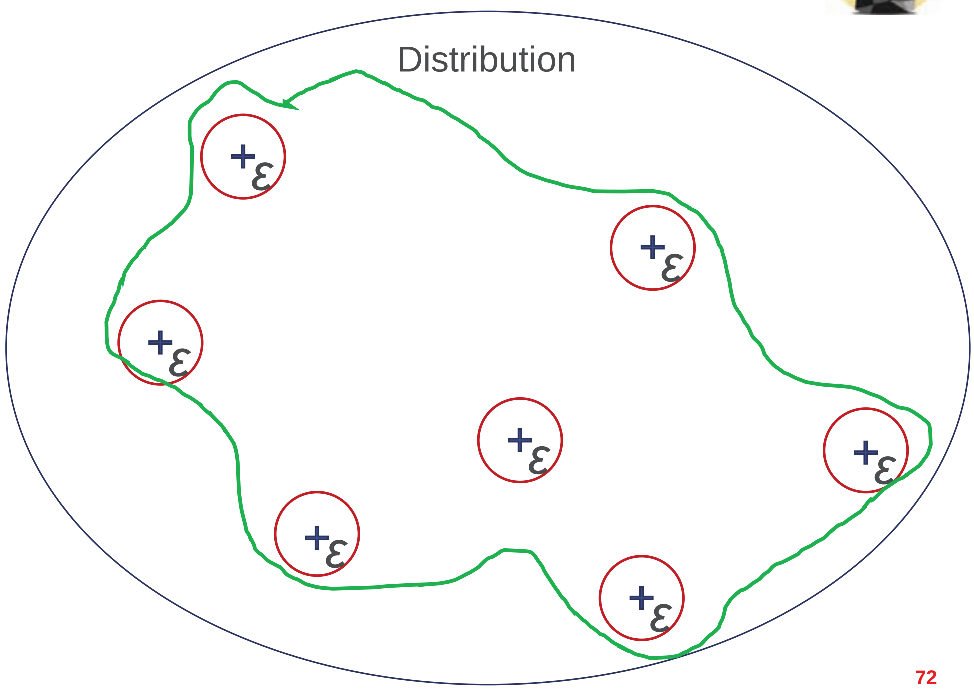
Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

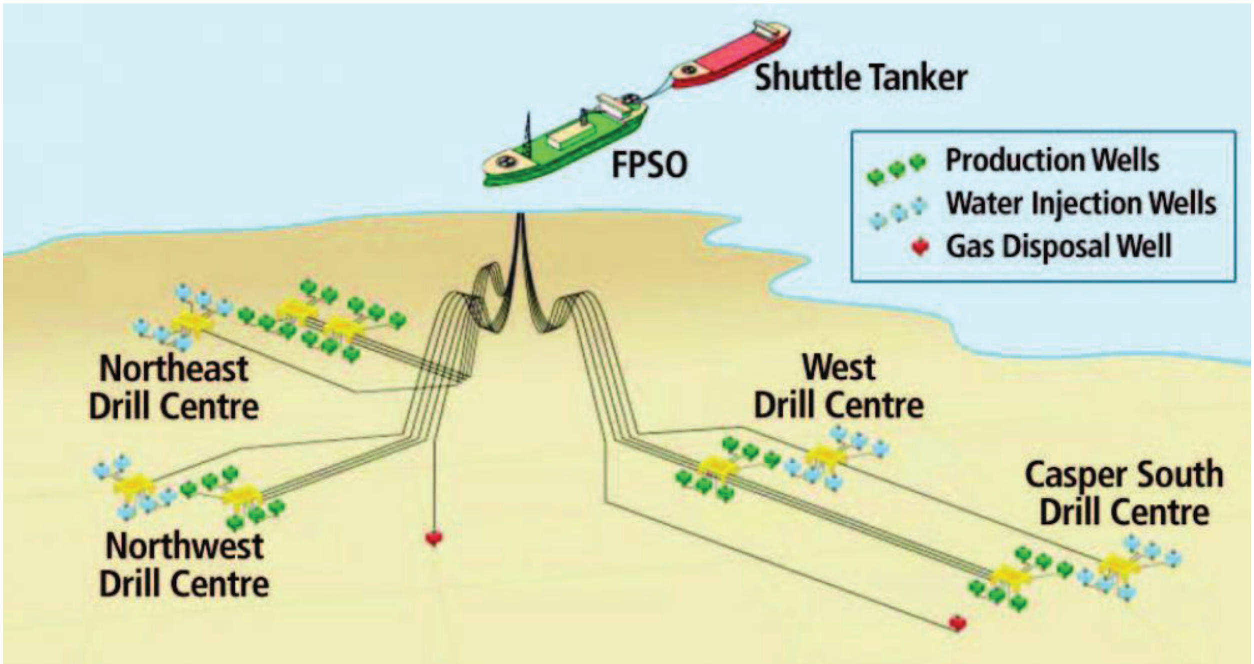
How does it behave with the neighborhood of selected data?

What is it good for?

… But doesn't imply this



PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

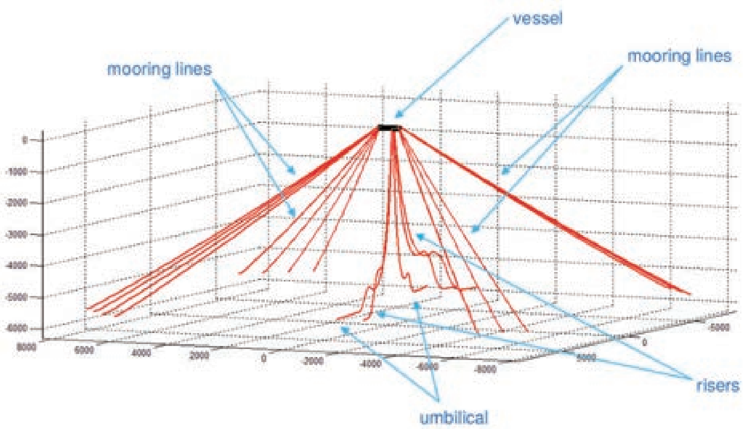


73

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

Mooring incidents (DeepStar® data from 1997-2012):

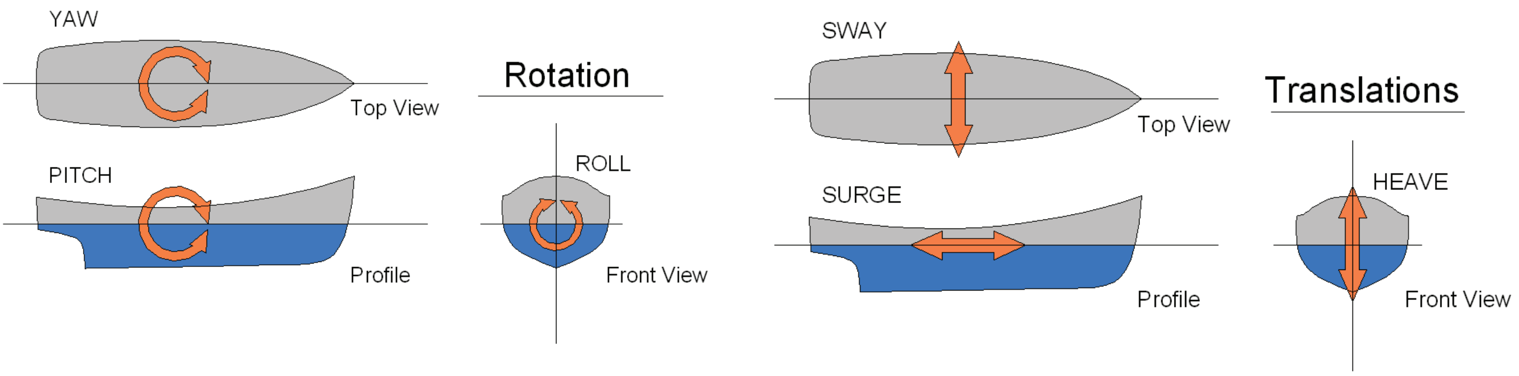
- 107 incidents from 73 facilities across the industry
- Potentially dire consequences
- Many FPSO have no means of monitoring lines
- Those who do face technical problems (robustness of equipment)



74

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

Patented dry monitoring detection systems, based on vessel positions and low-frequency periods (which can be obtained from Dual GPS)

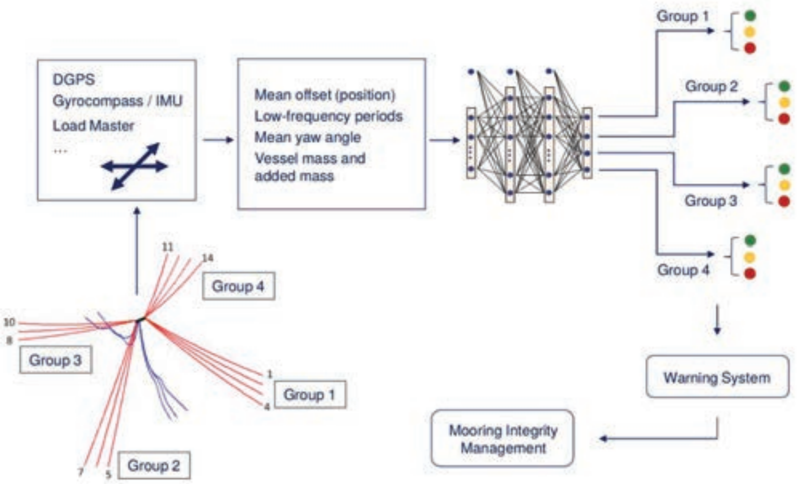


PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

- 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



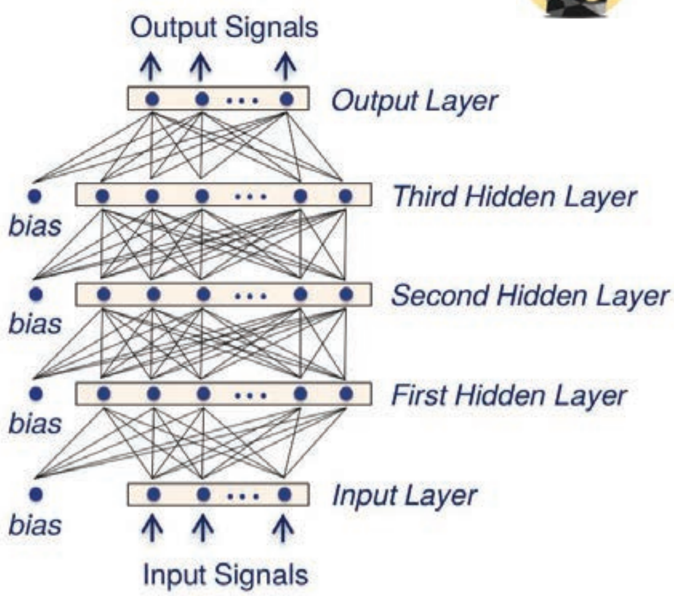
76

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

Ensuring robustness properties

- Stability of classification in presence of perturbation
- Perturbation per input (sensor sensitivity)
- Different perturbations for different inputs

(Also verified functional properties but NDA)



77

Rapid intro to FM

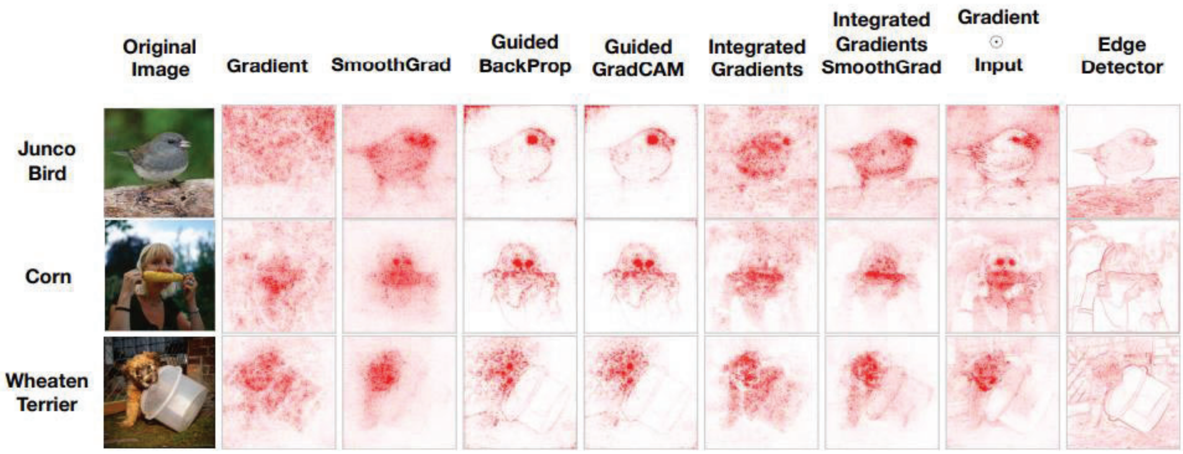
Examples for this talk:

- **Property-based testing (Renault)**
- **Verification of functional properties (Airbus)**
- **Robustness evaluation (Technip)**
- **Out-of-Distribution detection (Thales)**

cea

78

Peeking in the black box: saliency maps, post-hoc XAI

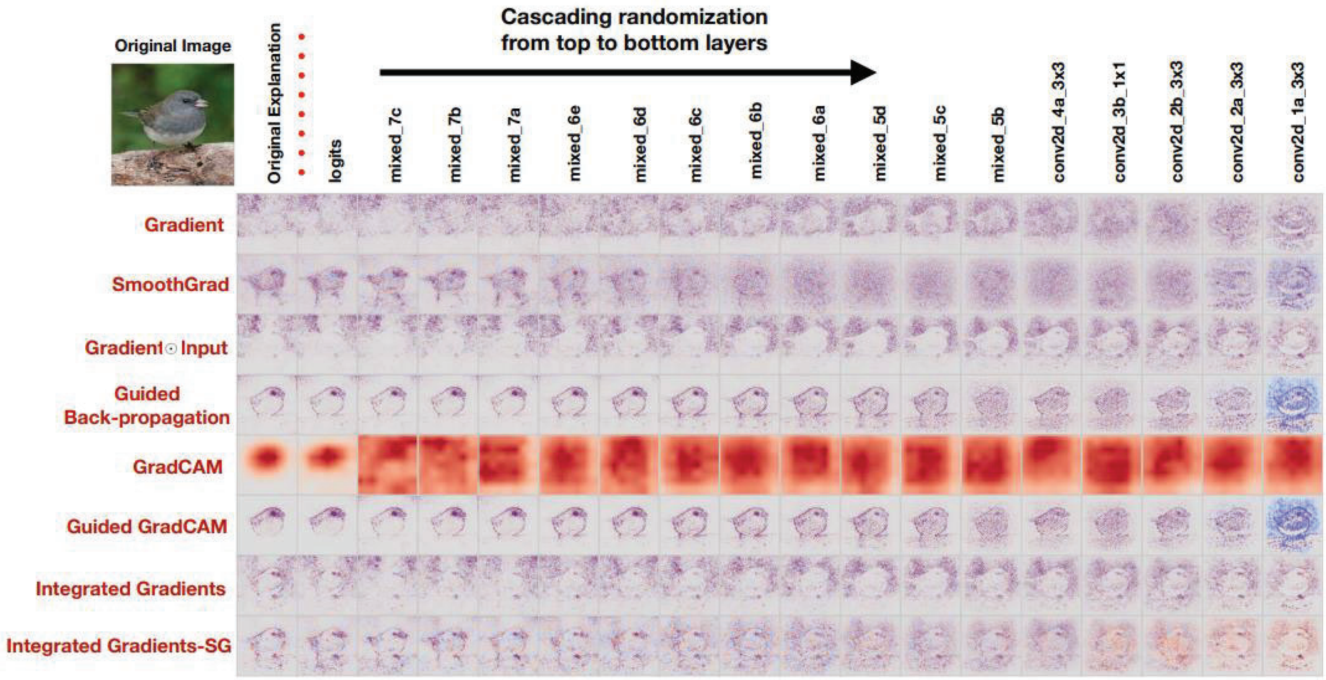


Sanity Checks for Saliency Maps

Julius Adebayo*, Justin Gilmer♯, Michael Muelly♯, Ian Goodfellow♯, Moritz Hardt♯†, Been Kim♯

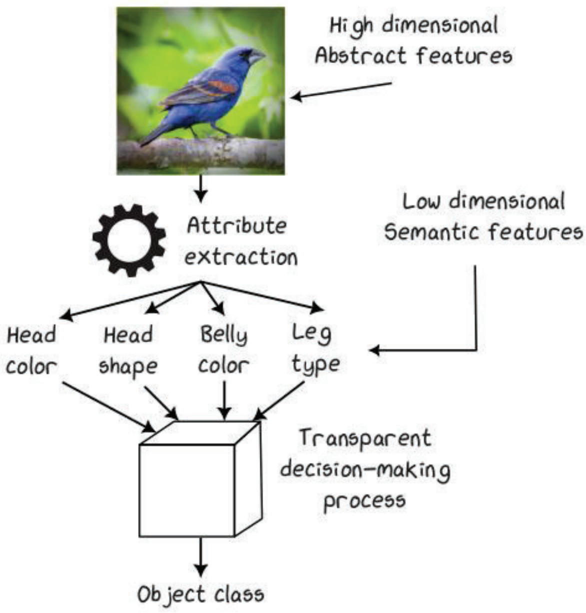
79

Peeking in the black box: saliency maps, post-hoc XAI



80

Peeking in the black box: XAI by design

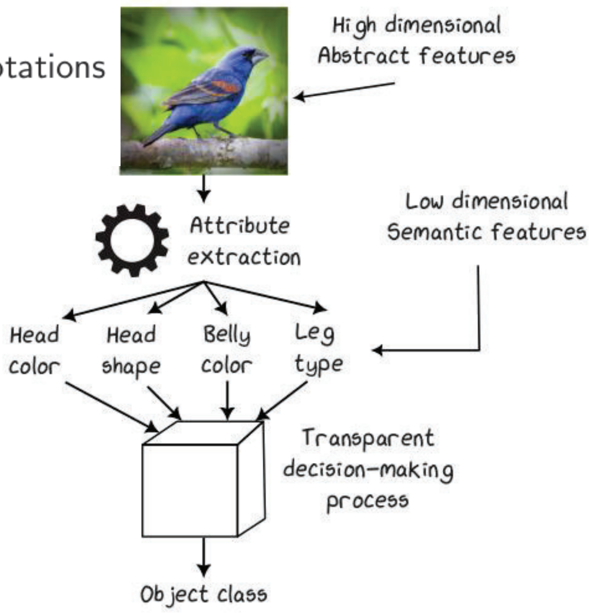


Peeking in the black box: XAI by design

Extract **semantic** information

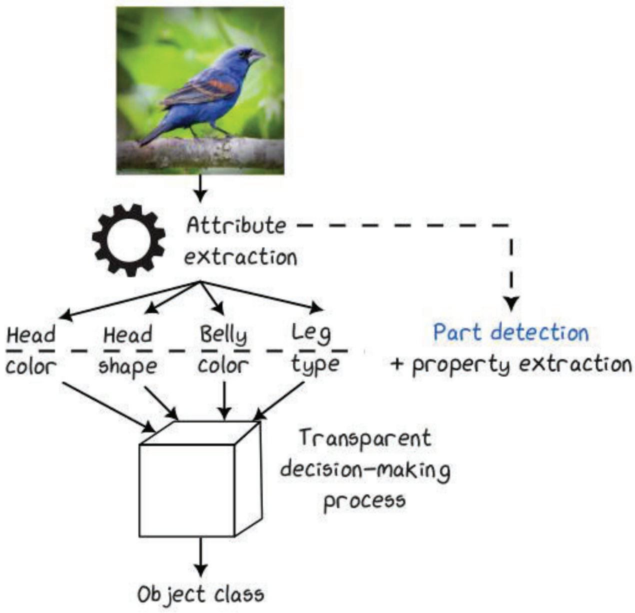
Attribute learning requires annotations

- Annotations are expensive
- Annotations can be incorrect



82

Peeking in the black box: XAI by design



PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Original

Goal

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

84

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

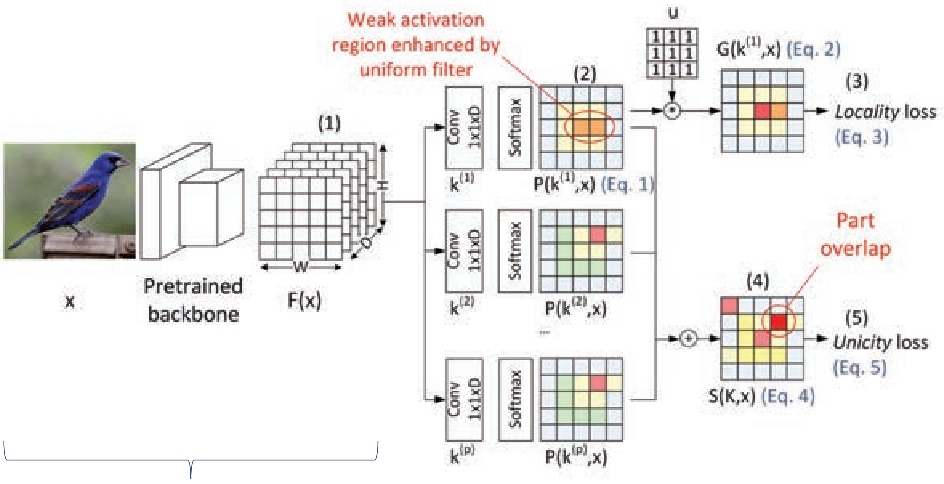
PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

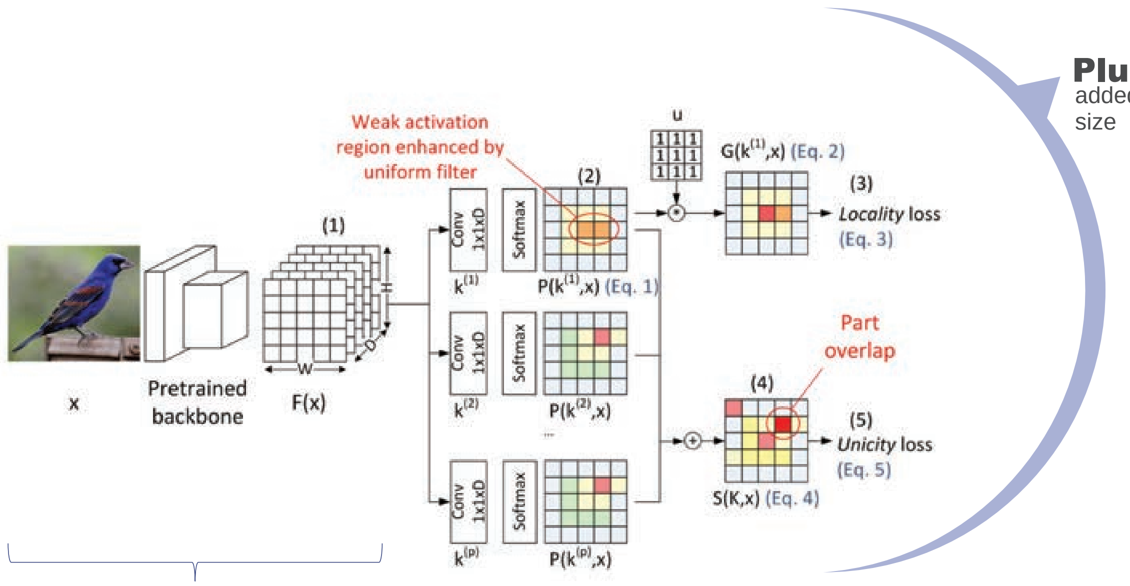


Taken as is

88

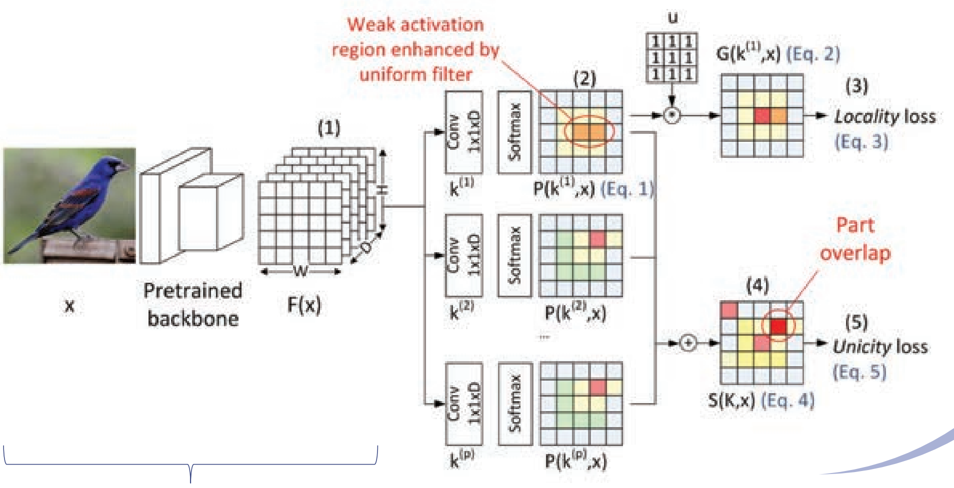
PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Pluggable
added to existing backbone, fraction of the size



Taken as is

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

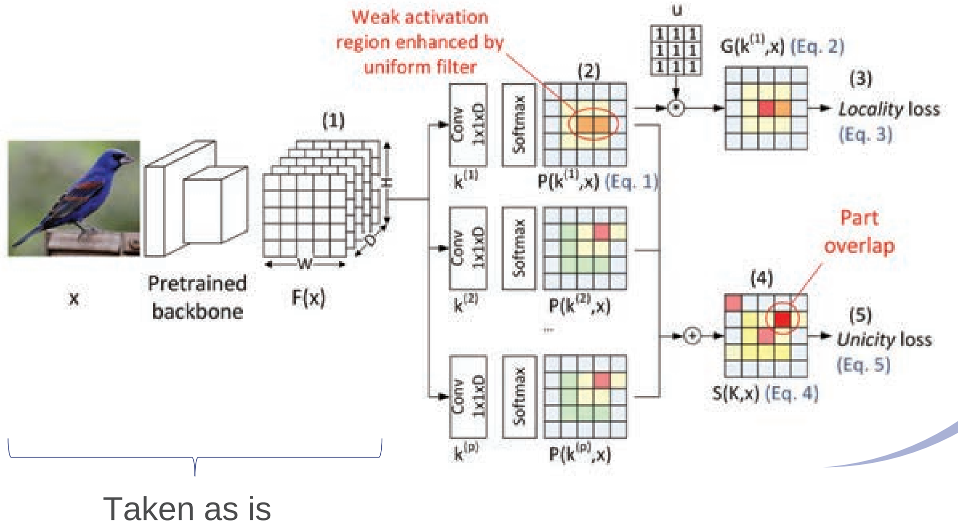


Pluggable
added to existing backbone, fraction of the size

Frugal
no need to fine-tune the backbone

Taken as is

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

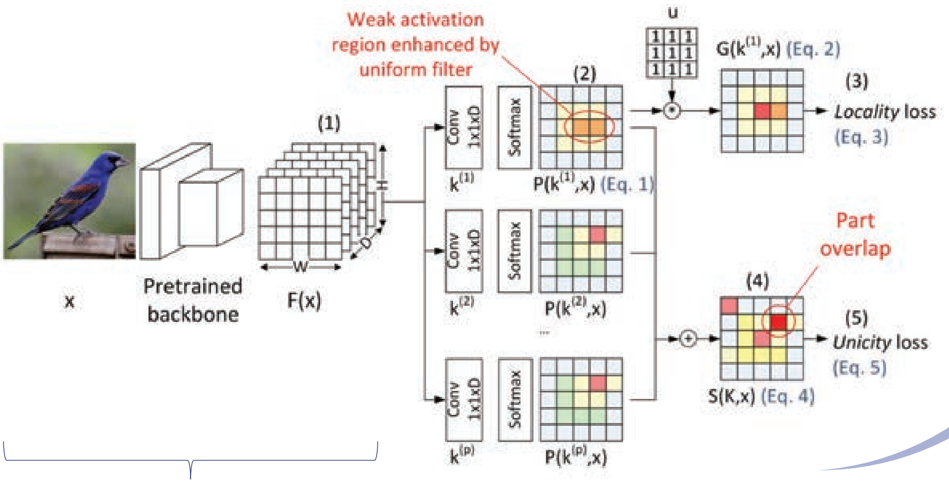


Pluggable
added to existing backbone, fraction of the size

Frugal
no need to fine-tune the backbone

Non-invasive
minimal access to backbone, fraction of the data

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



Taken as is

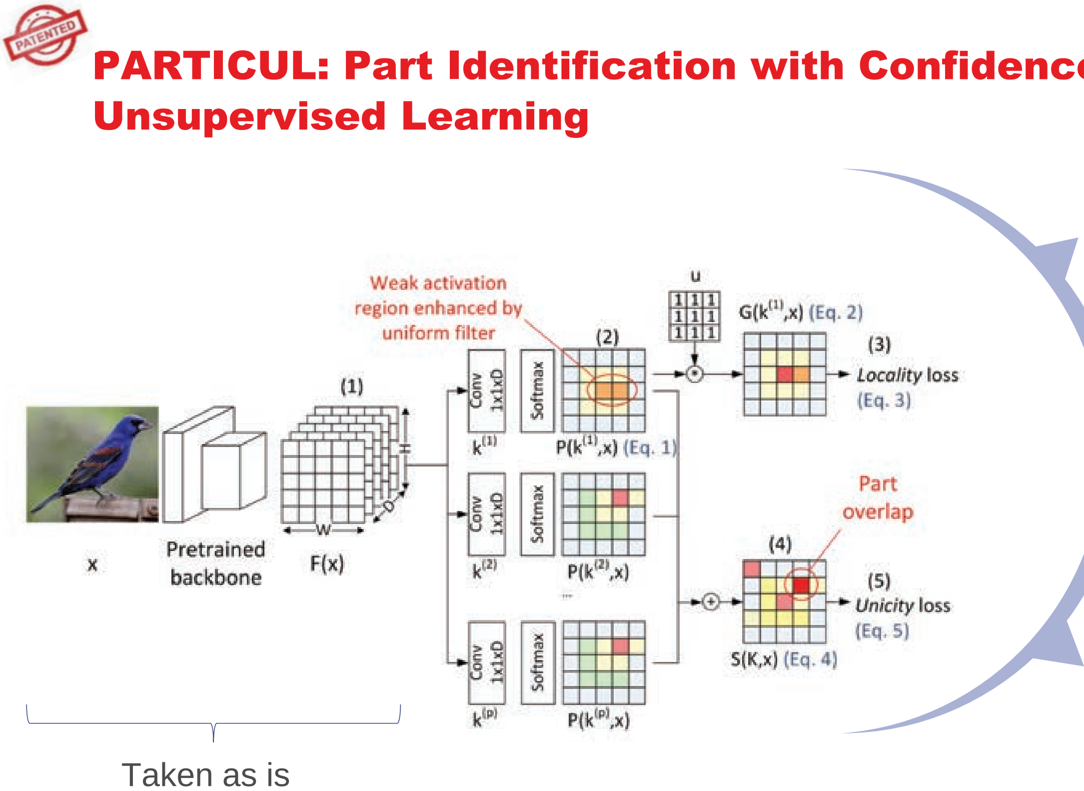
Pluggable
added to existing backbone, fraction of the size

Frugal
no need to fine-tune the backbone

Fast
convergence in a few epochs

Non-invasive
minimal access to backbone, fraction of the data

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



Pluggable
added to existing backbone, fraction of the size

Frugal
no need to fine-tune the backbone

Fast
convergence in a few epochs

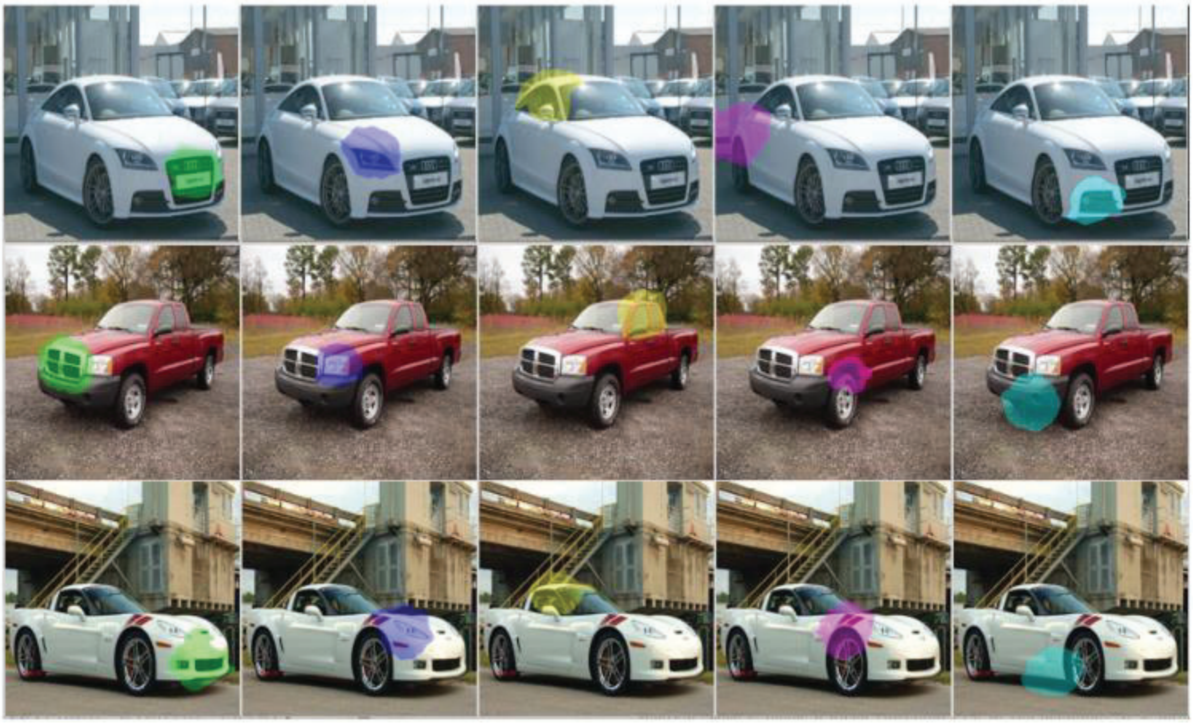
Measured
gives confidence measures of the detections

Non-invasive
minimal access to backbone, fraction of the data

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

94

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



95

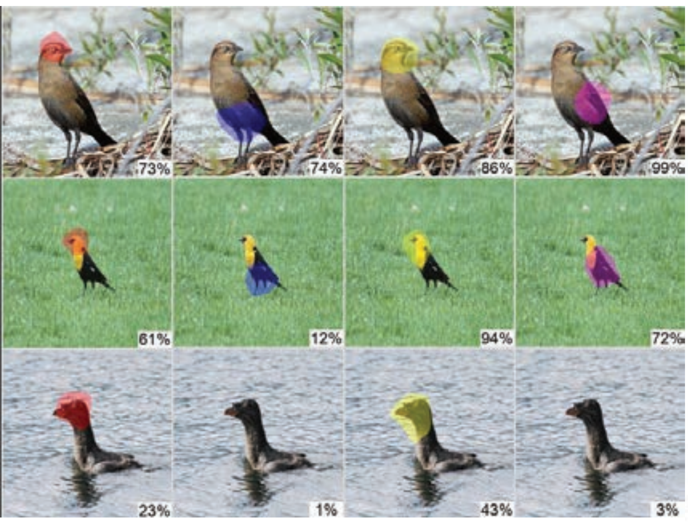
PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

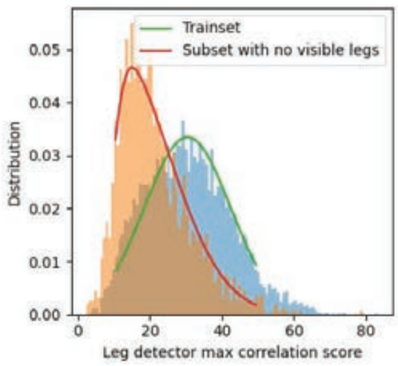
Detectors have no "knowledge" of which part they are detecting

Need for manual (human) definition (semantic value)

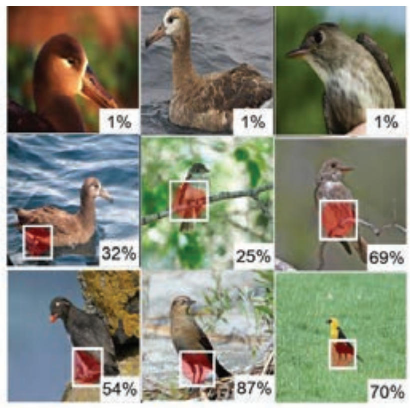
96

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning





(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).



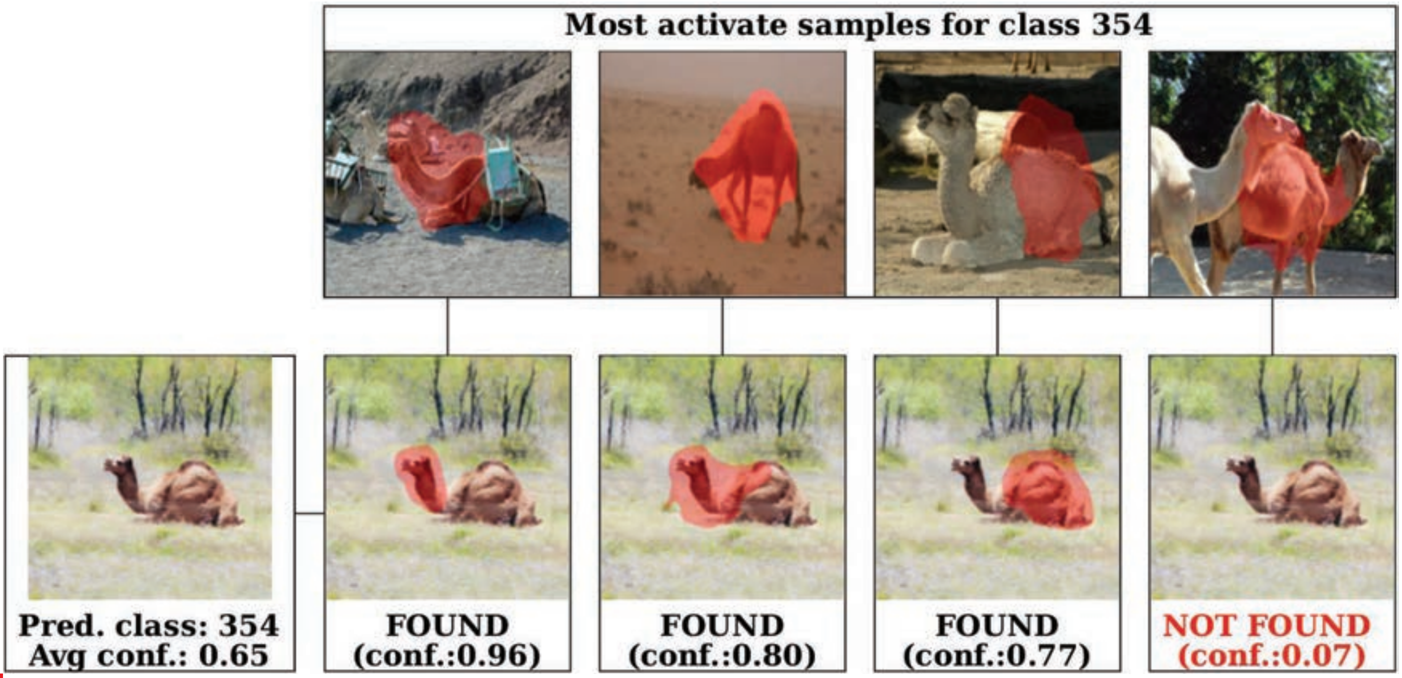
(b) Confidence scores and part visualizations on images with non visible legs (top-row) and with visible legs (bottom rows).

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

What can be done for one macro class can be done for many micro classes

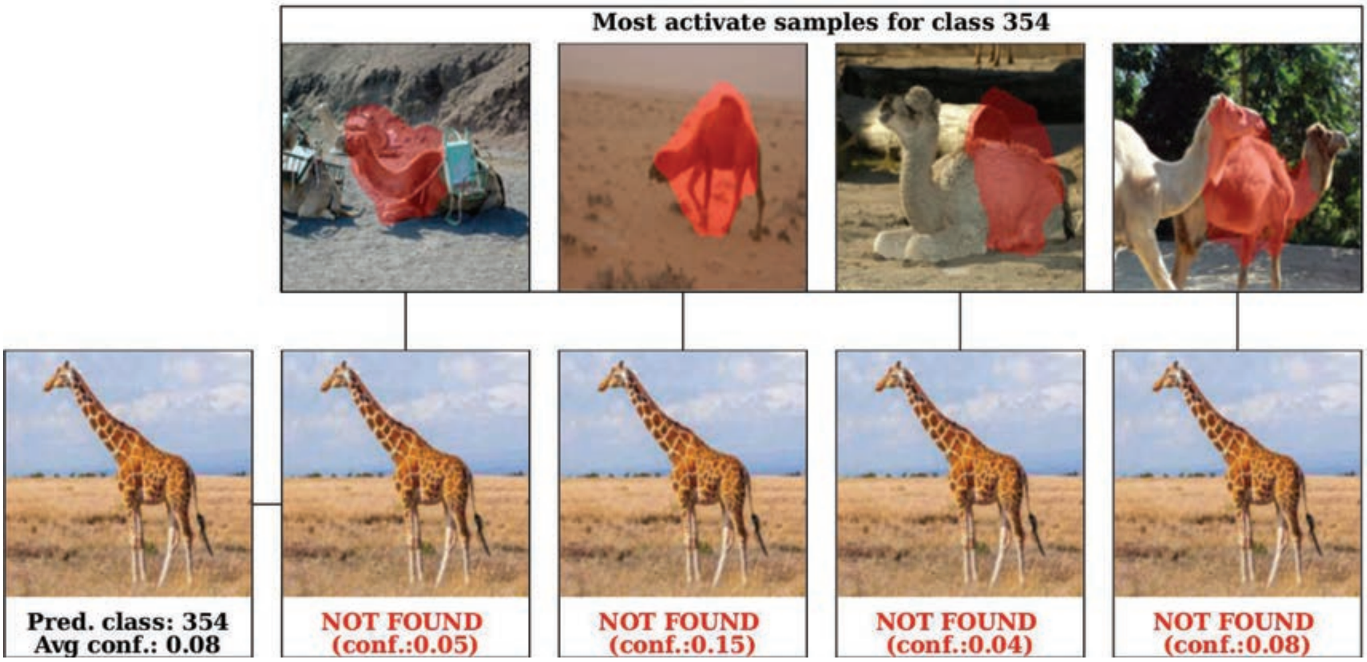
=> Set of detectors for each class

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



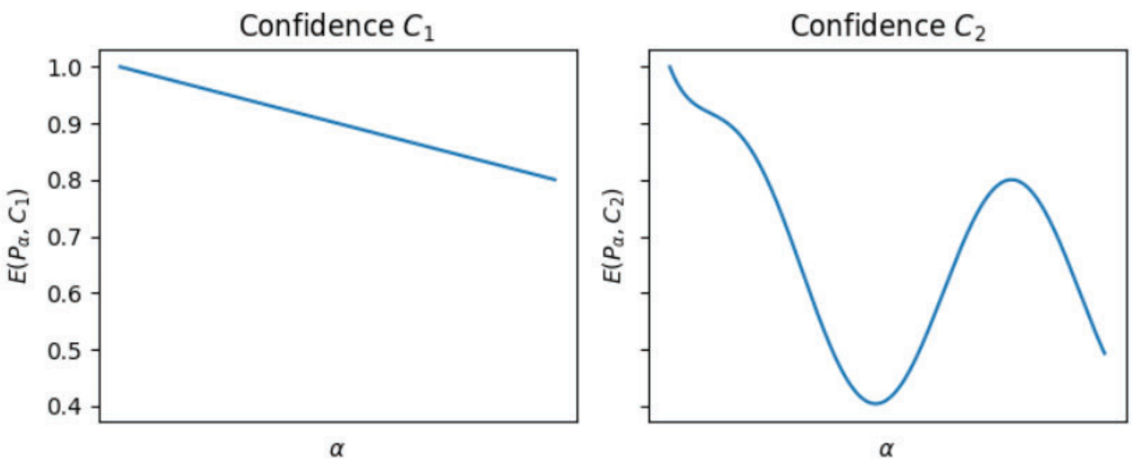
99

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



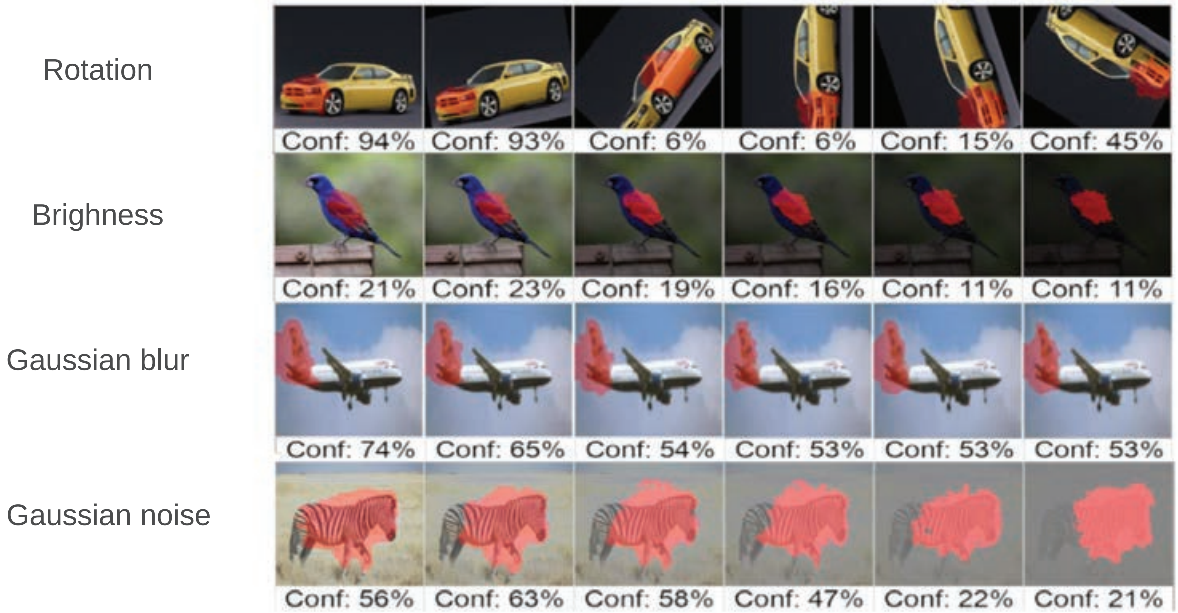
100

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



Xu-Darme, R., Girard-Satabin, J., Hond, D., Incorvaia, G., Chihani, Z. (2023). Contextualised Out-of-Distribution Detection Using Pattern Identification. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_36

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



Xu-Darme, R., Girard-Satabin, J., Hond, D., Incorvaia, G., Chihani, Z. (2023). Contextualised Out-of-Distribution Detection Using Pattern Identification. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_36

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification

Dissimilarity measures based on neuron activation bounds sometimes exhibit higher confidence on perturbed input

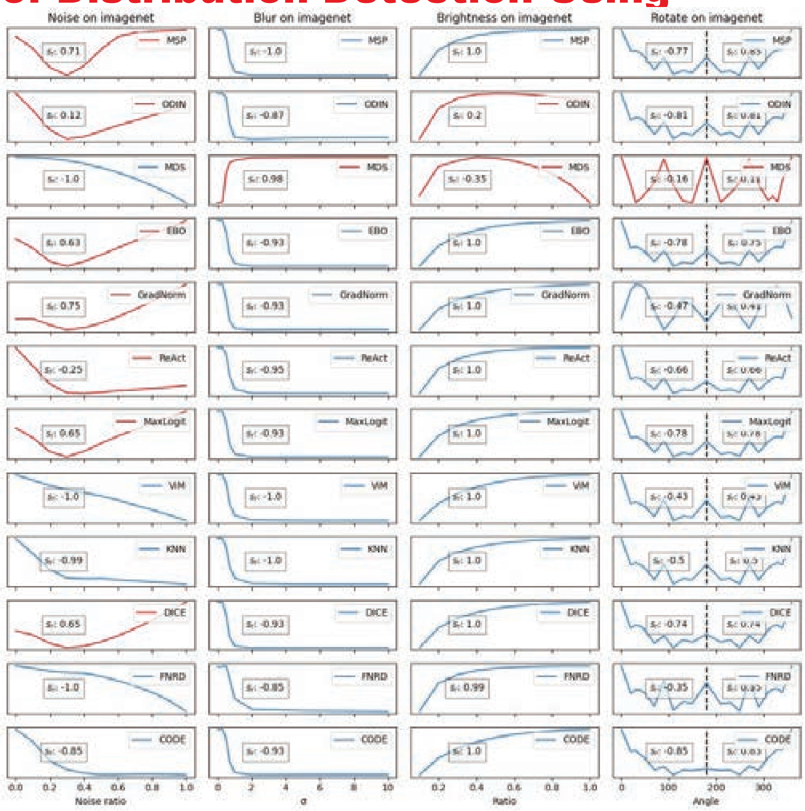
- => Maybe because some perturbations lower the amplitude of activation values, thus decreasing the propbabilty of activation outside of the bounds

Rotation is… wavy

- => Black filling at the angles ?

CODE seems consistent across perturbations and datasets (CIFAR10, CIFAR100, ImageNet).

Xu-Darme, R., Girard-Satabin, J., Hond, D., Incorvaia, G., Chihani, Z. (2023). Contextualised Out-of-Distribution Detection Using Pattern Identification. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_36



03

Rapid intro to FM

Examples for this talk:

- Property-based testing (Renault)
- Verification of functional properties (Airbus)
- Robustness evaluation (Technip)
- Out-of-Distribution detection (Thales)
- Bonus track

104

What CAISAR is

Principle: Maximize coverage of AI models and properties

- Common expressive specification language
- Easy extensibility through clear interfaces
- Heuristic-aided V&V analysis
- Common aggregation of analysis outputs

Target: SVM, Neural Networks, XGBoost models, ensemble models,…

Application: depending on the used plug-ins. Currently includes

- SAVer for SVM
- Colibri for XGboost
- PyRAT, AB-Crown, Nnenum, Marabou for NN

Background: The federative platform strategy for V&V has been successful for critical SW (see, for example, Frama-C and Why3)

What CAISAR is
Characterizing AI Safety And Robustness

Aimed at all AI systems

While the current frenzy of AI trustworthiness is mostly focused neural networks, our industrial partners can also use other types of AI (e.g. SVM, XGBoost) in their products. This is why **CAISAR targets a wider range of AI systems.**

Standard oriented

By relying on AI standards (ONNX, NNnet) and formal methods standards (SMT, CP), CAISAR maximizes the potential for **inclusiveness**. Any tool that supports these standards is a potential addition to the CAISAR platform.

Modular and extensible

Written in the functional language OCaml, adding a verification, analysis or testing software to CAISAR's toolsuit is made easier through a unified interface, and an instantiation guided by data-types.

Interoperability

An internal representation for property language and AI representation helps reinforce the **synergy between the different tools** in CAISAR, where one analyser can rely on, and complement, the partial output of another.

Maintainable

Functional programming provides easy mathematical reading, lowering the entry barrier for understanding the inner workings of CAISAR. Strong typing also minimizes errors with informative messages.

Open-source

Our vision of CAISAR is collaborative in essence. To encourage cooperation and build a community of trust-minded entities, an easy access to the platform, supplied with documentations and tutorials, is our priority.

cea

106

What CAISAR is
Characterizing AI Safety And Robustness

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.

```
let function normalize_t (i: t) (mean: t) (range: t) : t =
  (i .- mean) ./ range
let function denormalize_t (i: t) (mean: t) (range: t) : t =
  (i .* range) .+ mean
let function normalize_input (i: input) : input =
  Vector.mapi i normalize_by_index
let function denormalize_output_t (o: t) : t =
  denormalize_t o
    (7.5188840201005975316661533724982291460037231445312 5:t)
      (373.94992000000000200816430151462554931640625:t)
let runP1 (i: input) : t
  requires { has_length i 5 }
  (* constraints the inputs to respect the specification *)
  requires { valid_input i }
  requires { intruder_distant_and_slow i }
  ensures { result .≤ (1500.0:t) }  =
    let j = normalize_input i in
    let o = (nn @@ j)[clear_of_conflict] in
    (denormalize_output_t o)
```

108

What CAISAR is
Characterizing AI Safety And Robustness

```
goal pruned:
  CSV.forall_ dataset (fun _ e →
    forall perturbed_e.
      has_length perturbed_e (length e) →
      FeatureVector.valid feature_bounds perturbed_e →
      let perturbation = perturbed_e - e in
      ClassRobustVector.bounded_by_epsilon perturbation eps →
      let out_1 = nn_1@@perturbed_e in
      let out_2 = nn_2@@perturbed_e in
      .- delta .≤ out_1[0] .- out_2[0] .≤ delta
  )
```

Fig. 13: A WhyML specification with several NNs at once

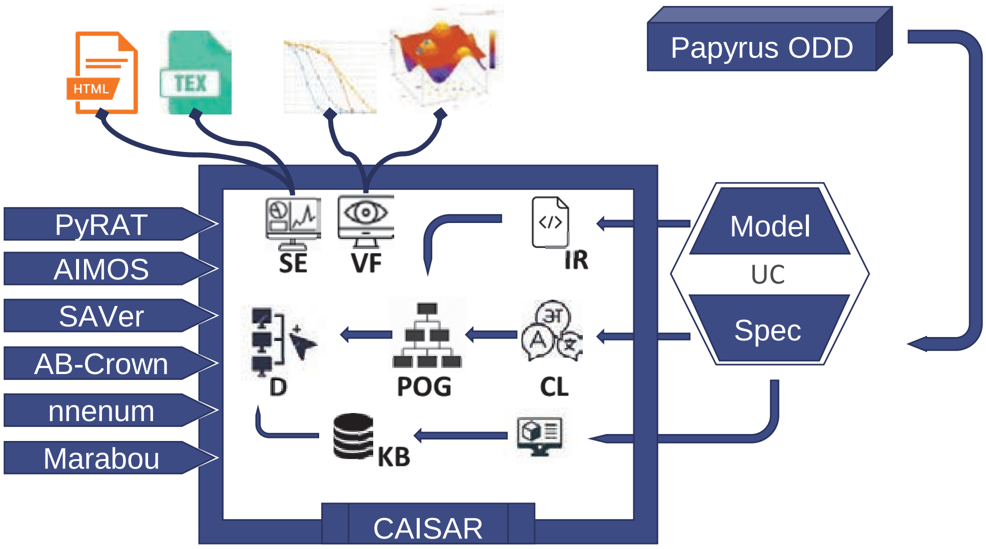
What CAISAR is
Characterizing AI Safety And Robustness

```
goal splitted:
  CSV.forall_ dataset (fun l e →
    forall perturbed_e.
      has_length perturbed_e (length e) →
      FeatureVector.valid feature_bounds perturbed_e →
      let perturbation = perturbed_e - e in
      ClassRobustVector.bounded_by_epsilon perturbation eps →
      let out1 = pre_nn@@perturbed_e in
      let out2 = post_nn@@out1 in
      forall j. Label.valid label_bounds j → j ≠ l →
      out2[l] .≥ out2[j]
)
```

Fig. 14: A WhyML specification for the composition of NNs

110

What CAISAR is (going to be)
Characterizing AI Safety And Robustness



caisar-platform.com

111

Our lab

Verification of safety and robustness formal specifications through Abstract Interpretation	Metamorphic testing applied to AI (Available for teaching)	Open-source, modular, extensible platform to Characterize AI Safety And Robustness	Open-source Symbolic AI tools, Safe-by-design Constraint solvers	Case-based reasoning, explainability, out-of-distribution detection
PyRAT	AIMOS	CAISAR	Colibri & co	PARTICUL
Verification	**Test**	**Platform**	**Symbolic**	**XAI & uncertainty**

Symbolic AI: Colibri's

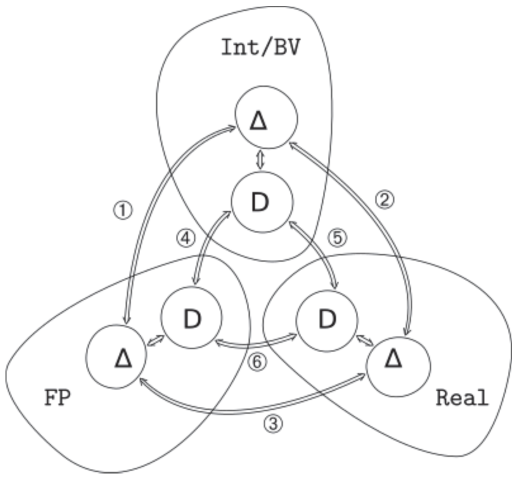
Principle: Safe-by-design Symbolic AI through a constraint solving library

- 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

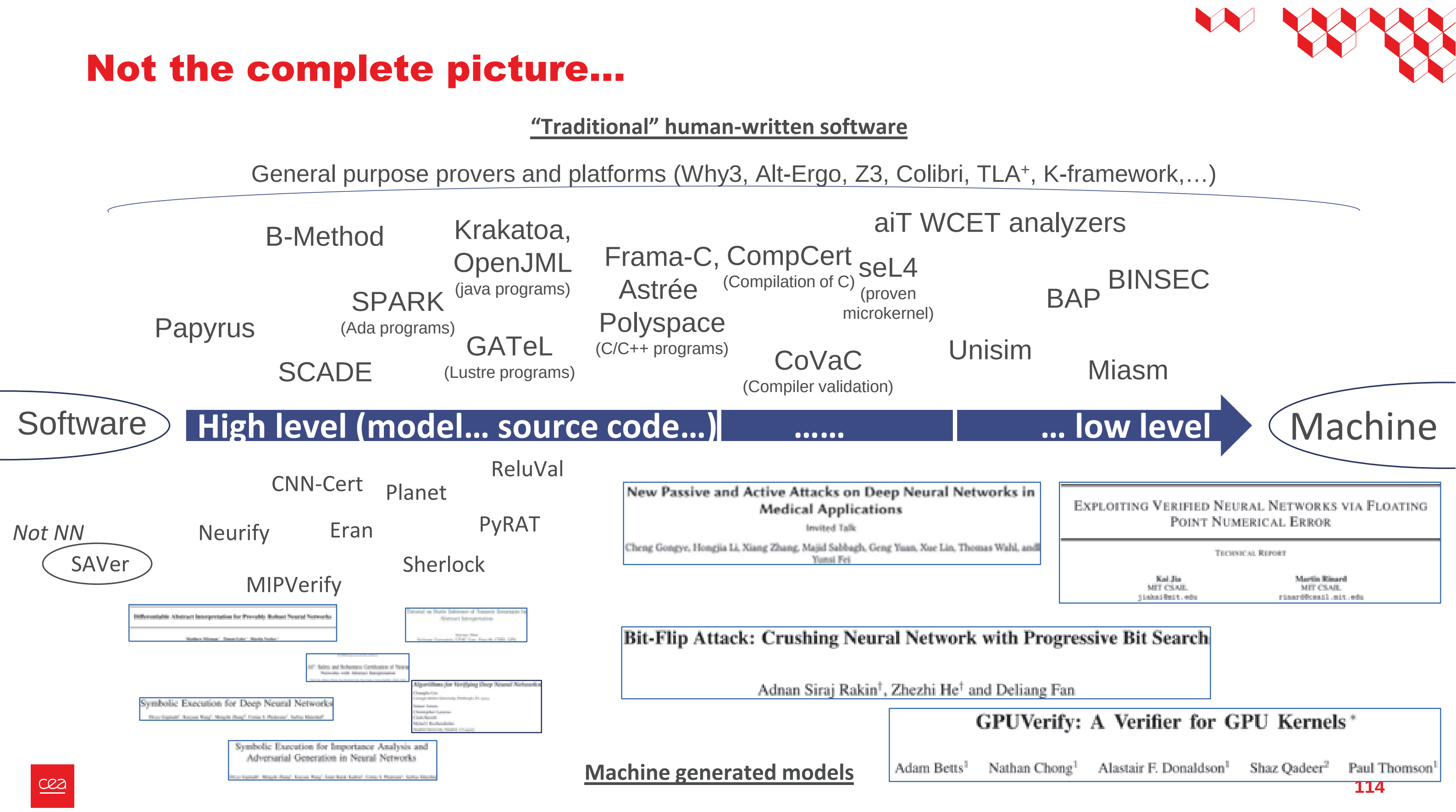
Target: XGBoost models, embedded software

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.frama-c.com


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, Astrée Polyspace (C/C++ programs)
CompCert (Compilation of C)
aiT WCET analyzers
seL4 (proven microkernel)
BINSEC
BAP
SPARK (Ada programs)
Papyrus
GATeL (Lustre programs)
SCADE
CoVaC (Compiler validation)
Unisim
Miasm
Software
High level (model... source code...)
......
... low level
Machine
CNN-Cert
ReluVal
Planet
Not NN
Neurify
Eran
PyRAT
SAVer
Sherlock
MIPVerify
New Passive and Active Attacks on Deep Neural Networks in Medical Applications
Exploiting Verified Neural Networks via Floating Point Numerical Error
Bit-Flip Attack: Crushing Neural Network with Progressive Bit Search
Adnan Siraj Rakin, Zhezhi He and Deliang Fan
Symbolic Execution for Deep Neural Networks
Symbolic Execution for Importance Analysis and Adversarial Generation in Neural Networks
GPUVerify: A Verifier for GPU Kernels
Adam Betts Nathan Chong Alastair F. Donaldson Shaz Qadeer Paul Thomson
Machine generated models
114

Conclusion

« Meta » overfitting on public datasets ?

Academia hasn't solved all trustworthiness problems, but industry can help us get there!

Trustworthiness challenges from industry are needed

VNN-Comp is calling for benchmarks (see aiverification.org, collocated with CAV)

Zakaria Chihani
zakaria.chihani@cea.fr

University of Maribor
Faculty of Organizational Sciences

Overview of 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

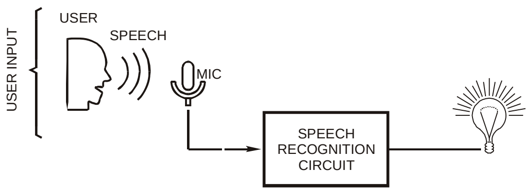
- Prototyping
- Control the movement of the wheelchair with speech
- The principle of cloud harvesting should be applied
- In addition to speech, control should be possible via web-based GUI
- Remote monitoring and control.
- Real-time video streaming from the wheelchair platform
- Biomedical signals
- 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.

Example of speech controlled device

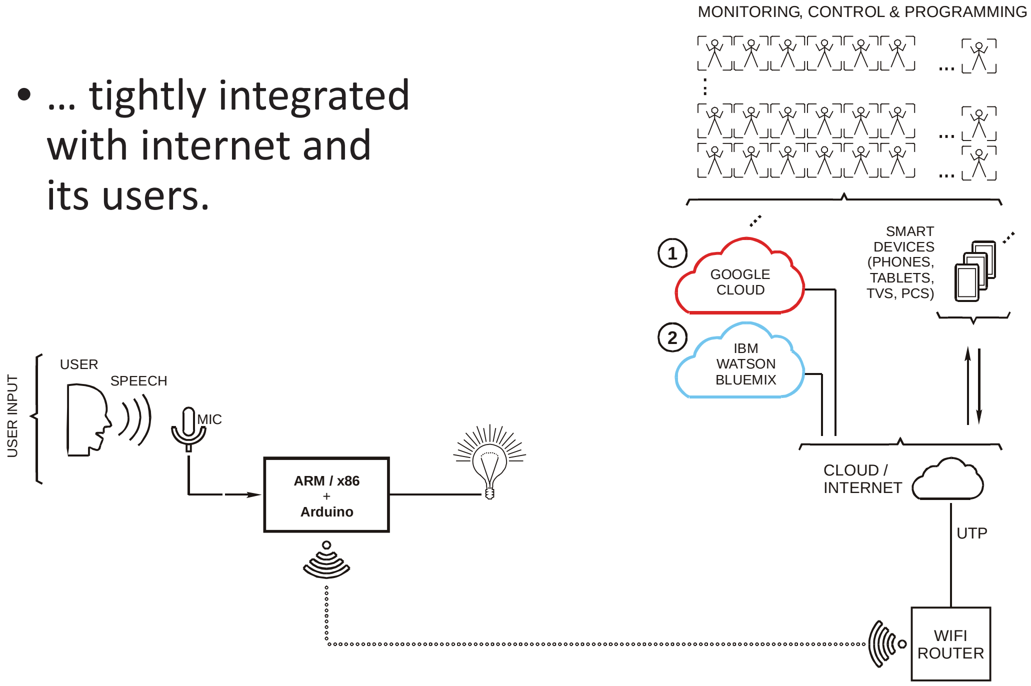
- Application of speech recognition circuit
- Embeded logic
- Hard to adapt
- Self sufficient device



Overview of Several CPS&IoT Prototypes

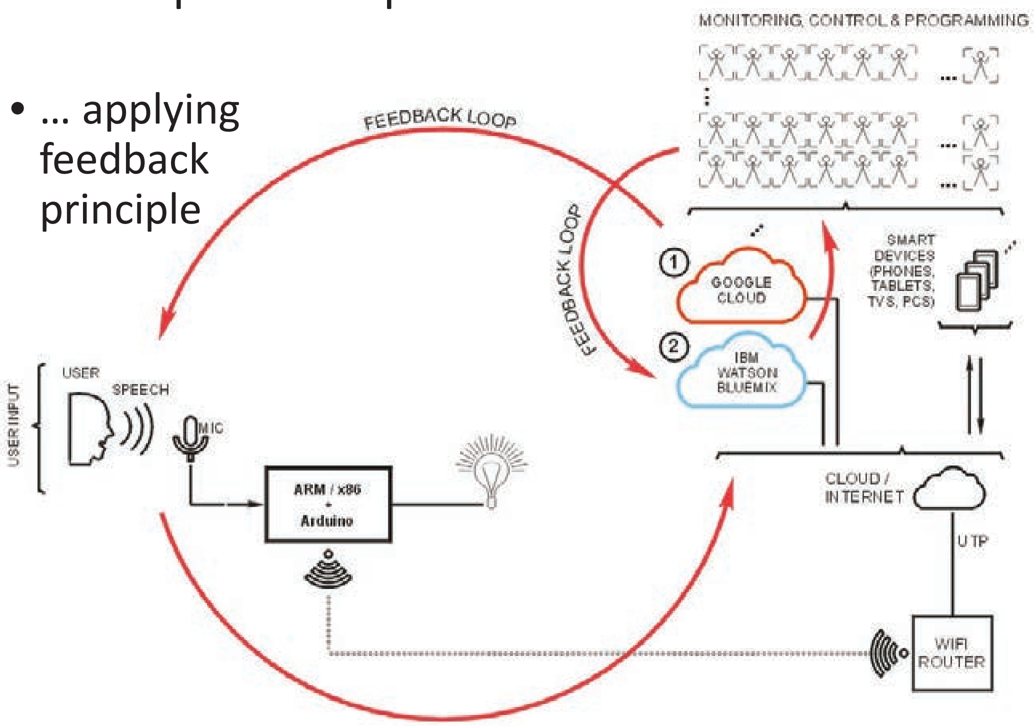
Example of speech controlled device (ver. 2)

- … tightly integrated with internet and its users.

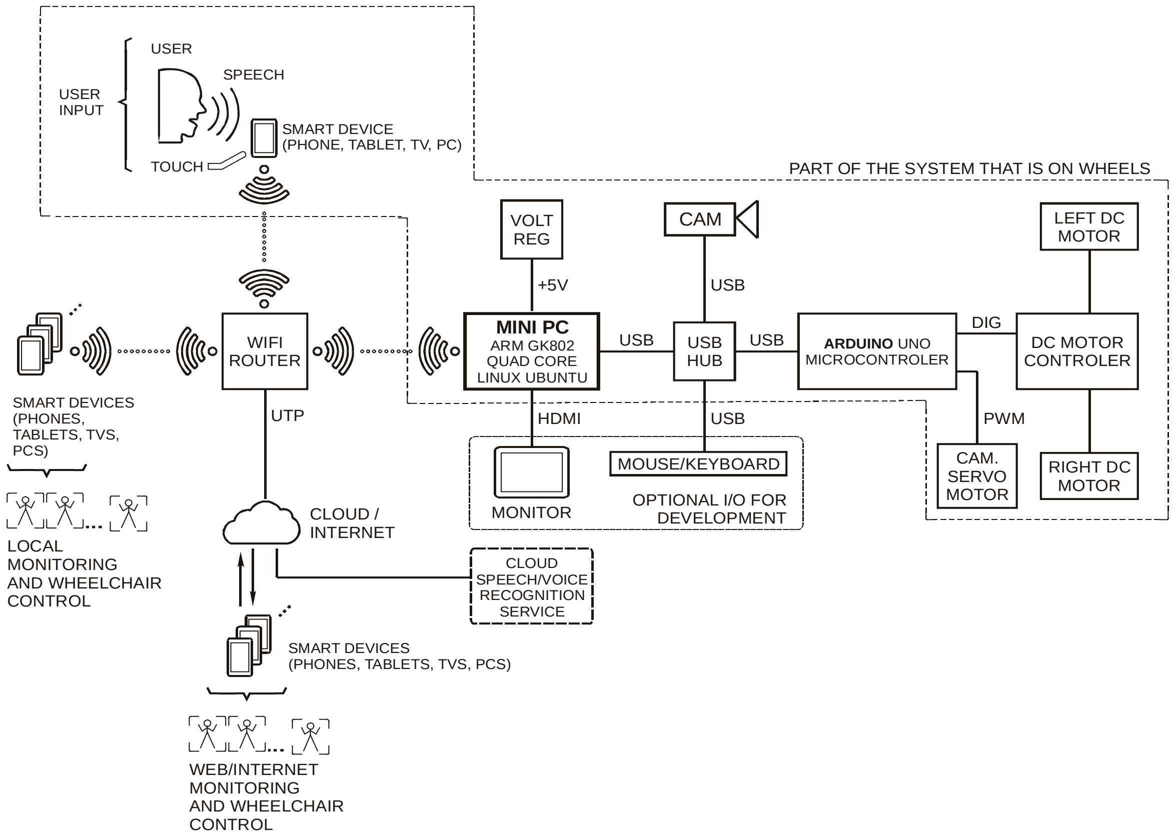


Example of speech controlled device (ver. 3)

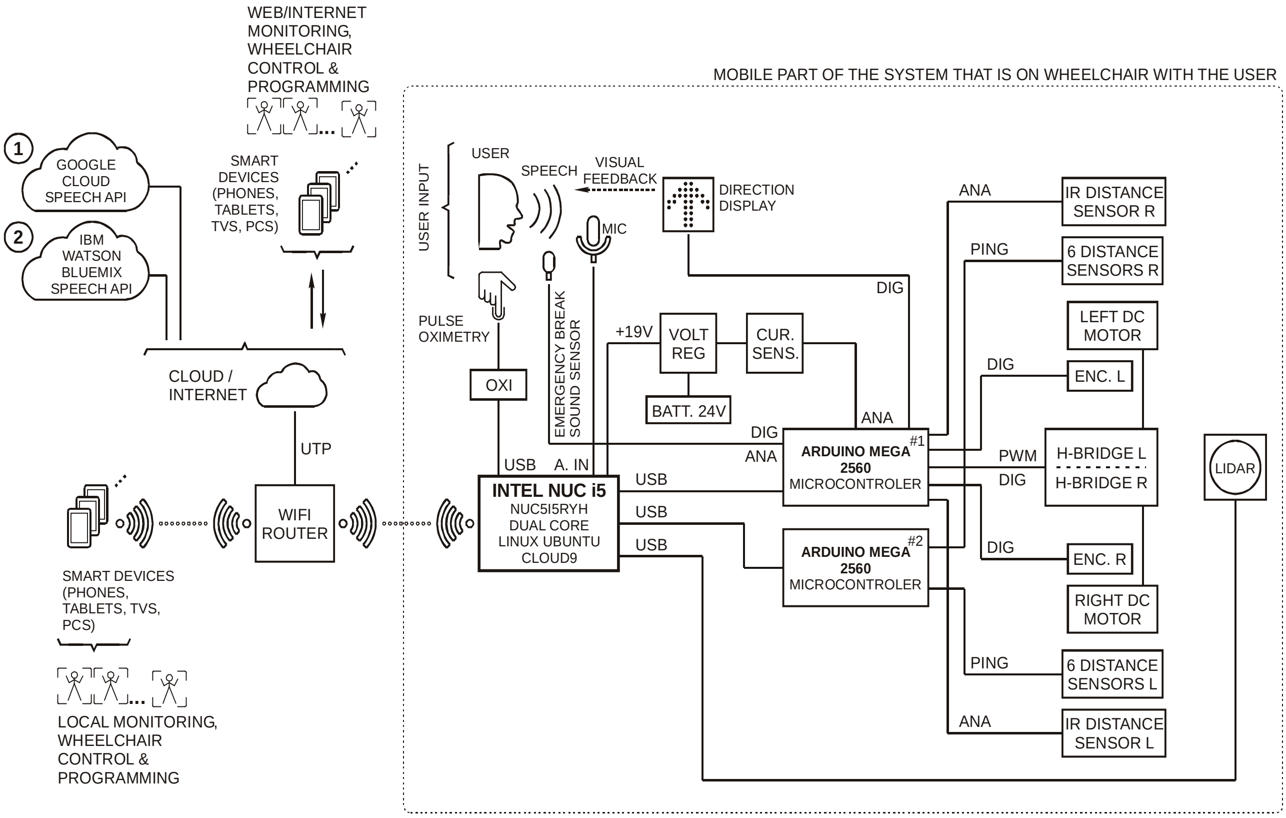
- … applying feedback principle



Wheelchair System Architecture 1st

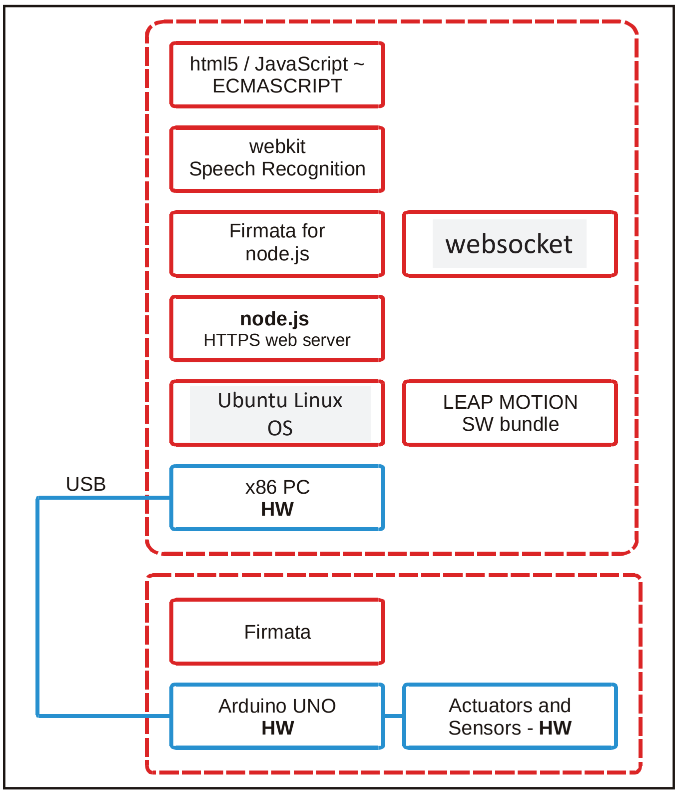


Wheelchair System Architecture 2nd



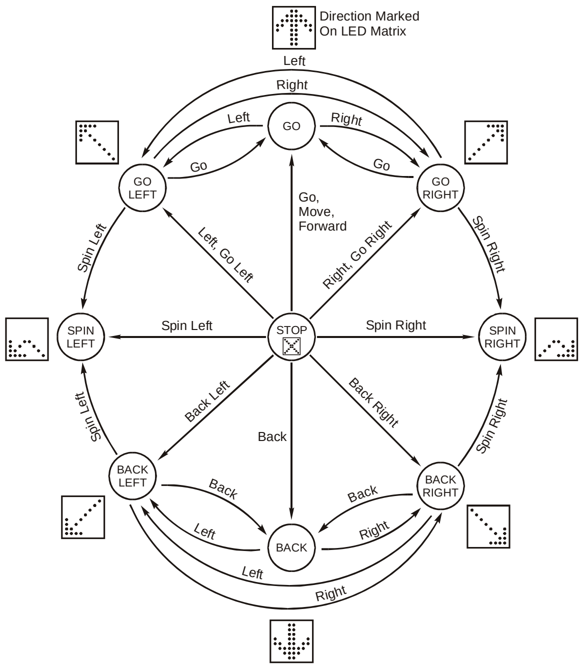
Software Stack

- node.js
- JavaScript / ECMAScript
- Firmata / Serial
- Ubuntu Linux
- Cloud Speech API
- Google & IBM Watson
- LEAP Motion SW Bundle



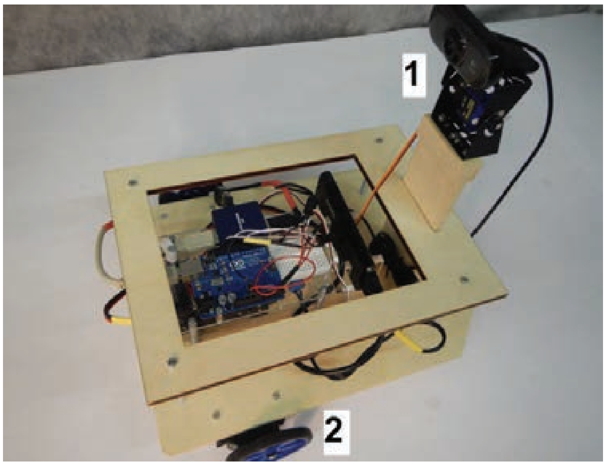
Transition Between States - Speech

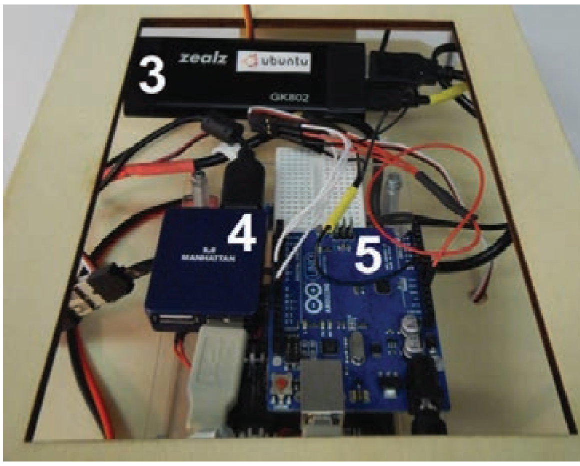
- Different interpretations depending on the sequence of issued commands

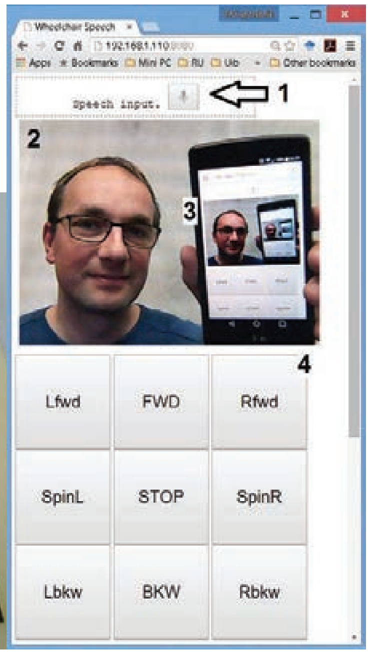


Prototype realization 0

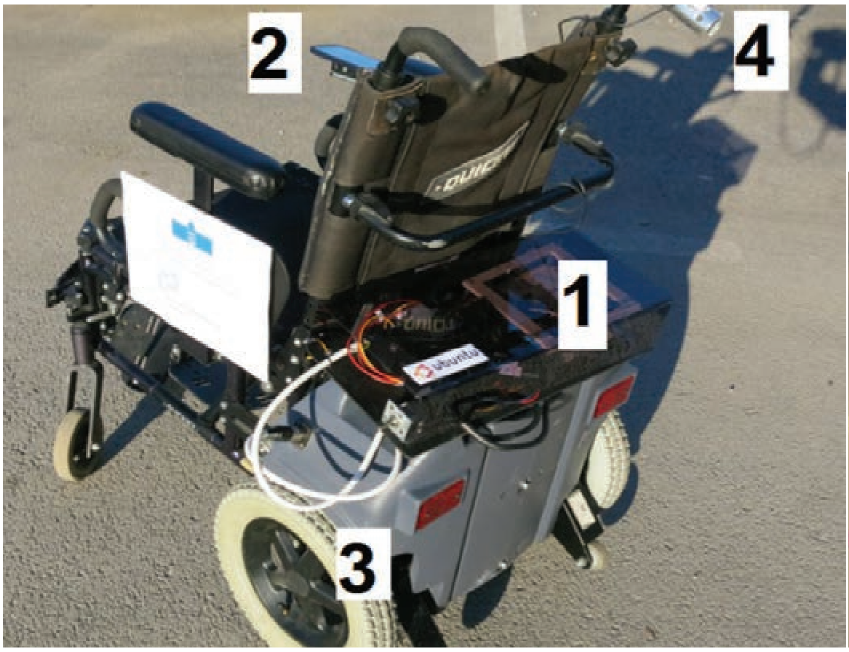
- ARM based solution GK802 quad core
- Speech controlled prototype: https://youtu.be/Y4EI7lBTxQA





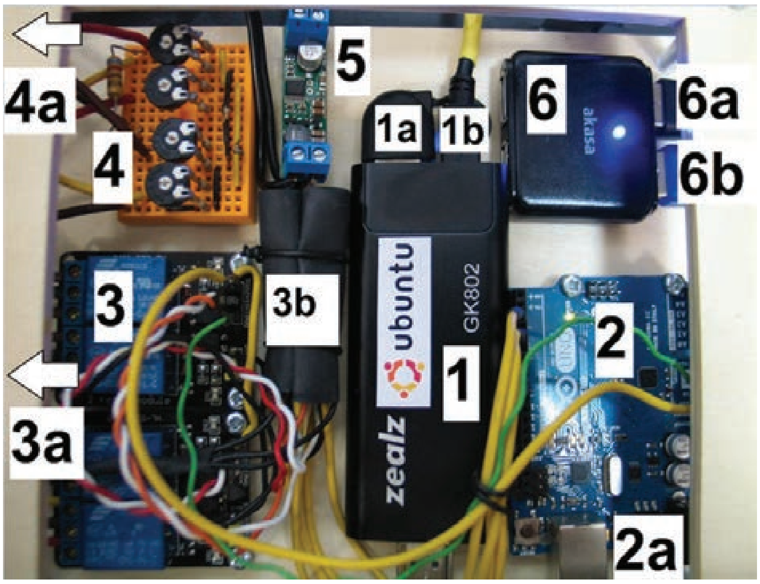


Full Scalled Prototype and Clinical Testing

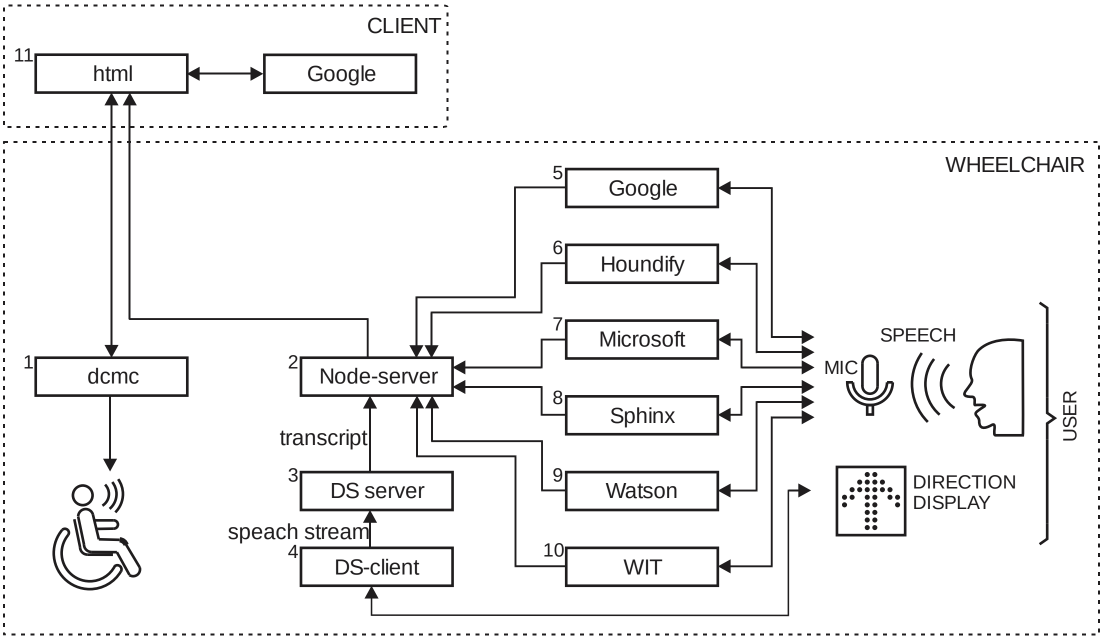


https://youtu.be/FMjffyMWcKM?t=728

Comparison



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:

$$f_2 = (a \land \neg b) \lor (\neg a \land b)$$

- For three paralel systems:

$$f_3 = (a \land \neg b \land \neg c) \lor (\neg a \land b \land \neg c) \lor (\neg a \land \neg b \land c)$$

- With CER when multiple clouds are harvested (+ latency etc.):

$$CER_m = \prod_{i=1}^{n} CER_i$$

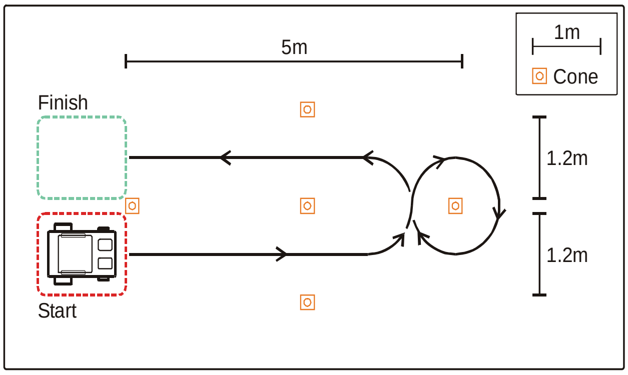
Speech-to-command cloud harvesting algorithm

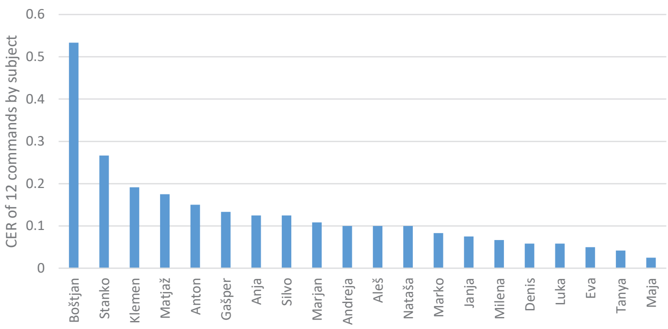
A1:	get user speech input
A2:	seed speech to the speech-recognition cloud field APIs
A3:	harvest set of interim transcripts and timestamps from cloud field
A4:	if $C_w > C_t$ add interim transcript to the $Cloud_i$ command subset
A5:	create unique union set of words for particular command from $Cloud_i$ command subset
A6:	check for pairwise disjoint condition for all unique union sets: $A_i \cap A_j \equiv \emptyset \ ; \ i \neq j$
A7:	if condition not met erase word pair
A8:	order checked unique union set by interim transcript timestamp
A9:	execute command with lowest timestamp

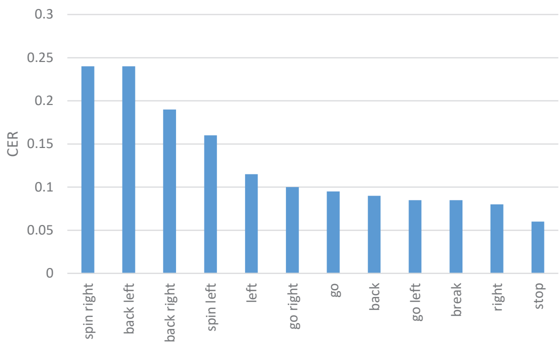
CER Measurement

- Testing 20 subjects
- Using poligon setup

$$CER = \frac{C_f}{\text{No}}$$

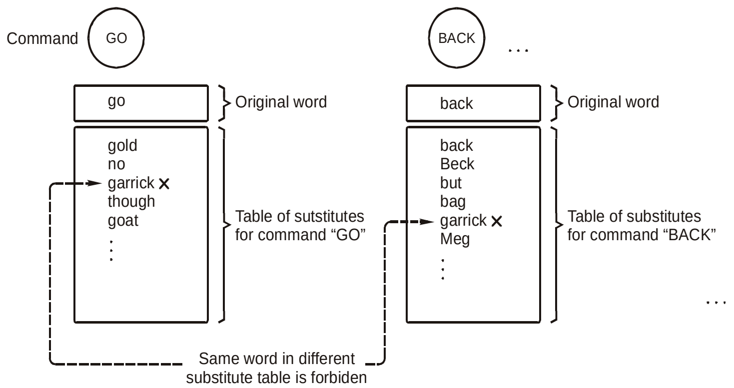






Algorithms

- Improve CER with table of substitutions
- Harvesting the clouds with corrections
- Application and automatic generation of substitution tables

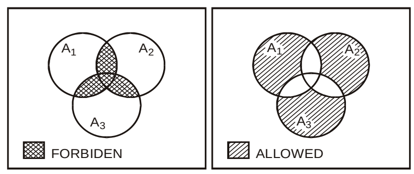


N substitution tables:

$A_1, A_2, A_3, \dots, A_n,$

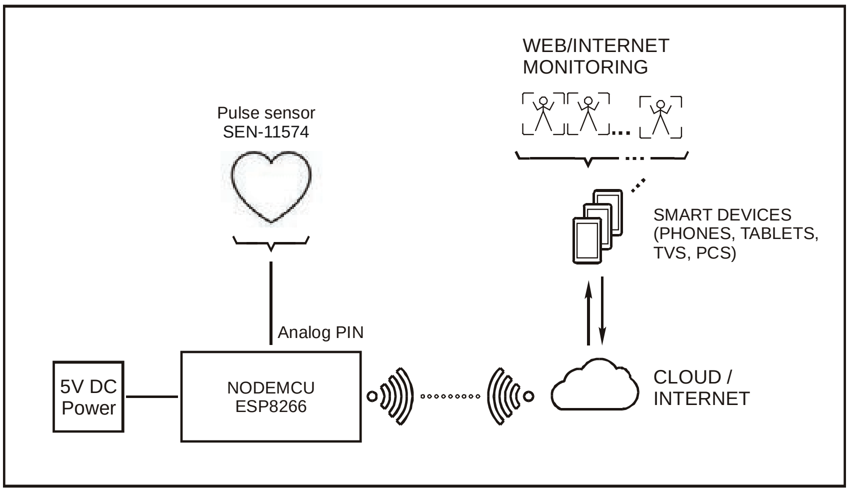
Pairwise disjoint

$A_i \bigcap A_j \equiv \emptyset \ ; \ i \neq j$



Streaming pulse data from Wheelchair

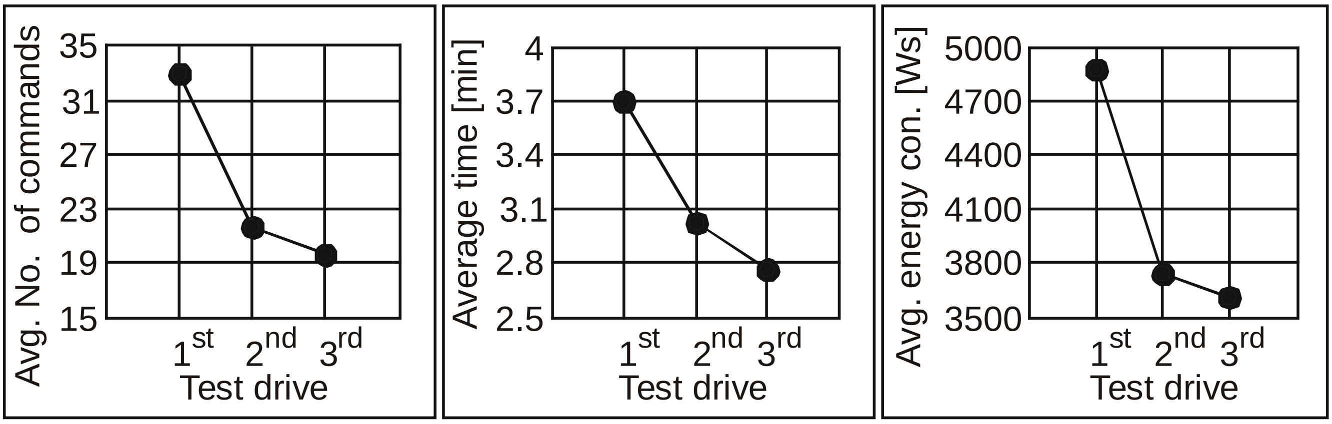
- NODEMCU ESP8266 module
- Less components and
- Directly connected to the Wi-Fi
- Sensor data processing is performed on the ESP8266
- Transmitted over Wi-Fi WebSocket to the cloud
- Additional data processing on the client side with JavaScript / ECMA Script



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

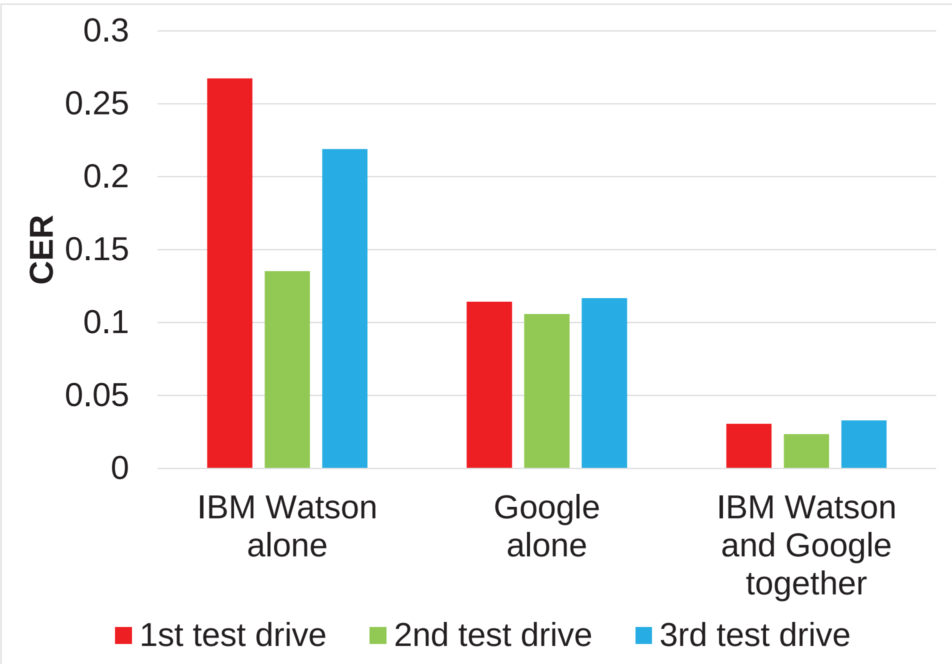
Learning effect



Harvesting Google & IBM Watson

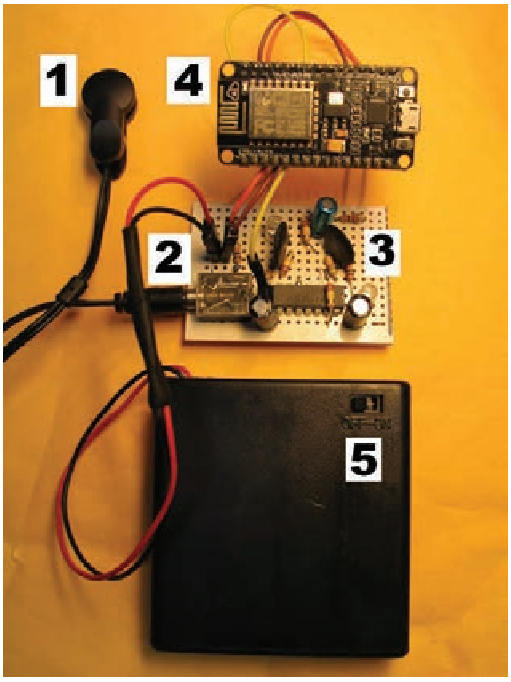
Run	No. of issued comm.	N	CER of Google API alone (CERg)	CER of IBM Watson API alone (CERw)	CER of Google & IBM Watson combined (CERgw)	Google CER improvem ent	Watson CER improvem ent
1st	490	20	0.11	0.27	0.03	8%	24%
2nd	340	20	0.11	0.14	0.02	8%	11%
3rd	274	20	0.12	0.22	0.03	8%	19%

Improvement



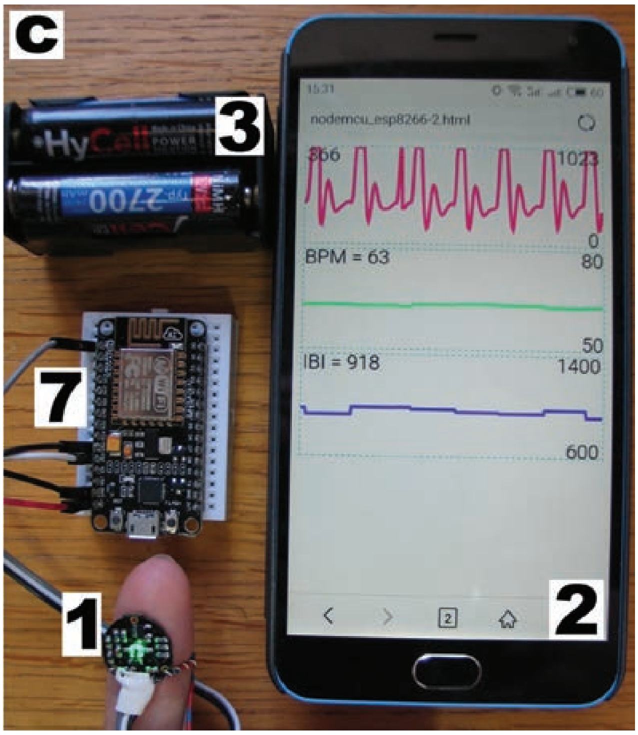
Group Heart Rate Monitoring

- 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



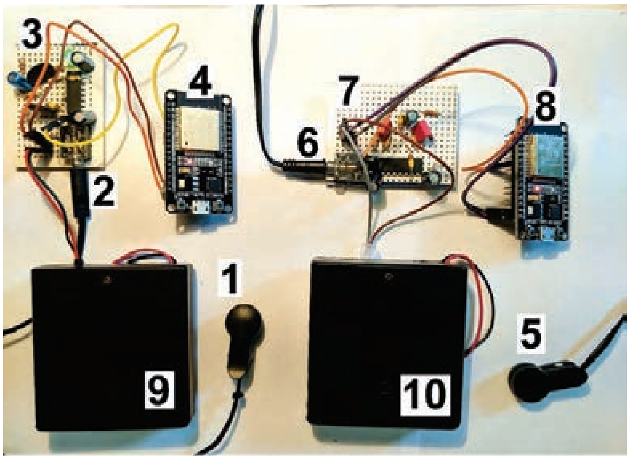
NODEMCU ESP8266

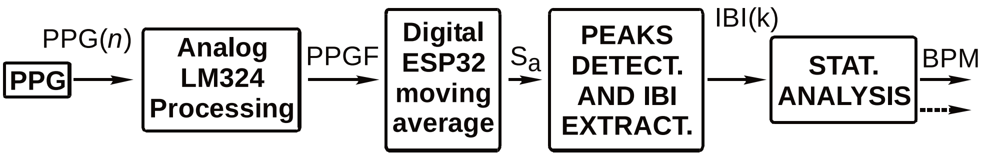
- Minimalistic regarding the hardware components
- The algorithm for processing the raw data, calculating the average number of Beats Per Minute (BPM), as well as the InterBeat Interval (IBI) in case of using NODEMCU was implemented on client's side in JavaScript
- Power consumption is a major issue



Example ~ ESP32

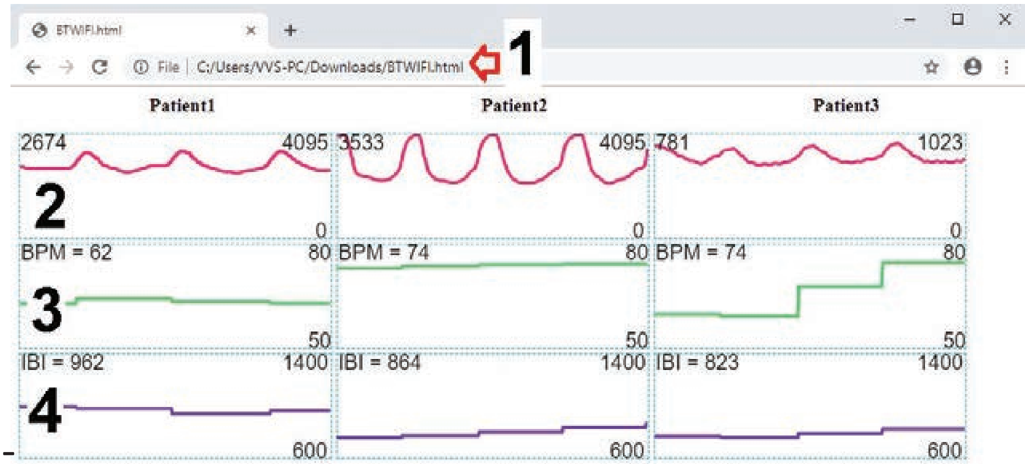
- ESP32 based configuration
- By developed software stack and hardware
  realization the results could be monitored in the web browser
- ESP32 websocket support
- Chrome – websocket support





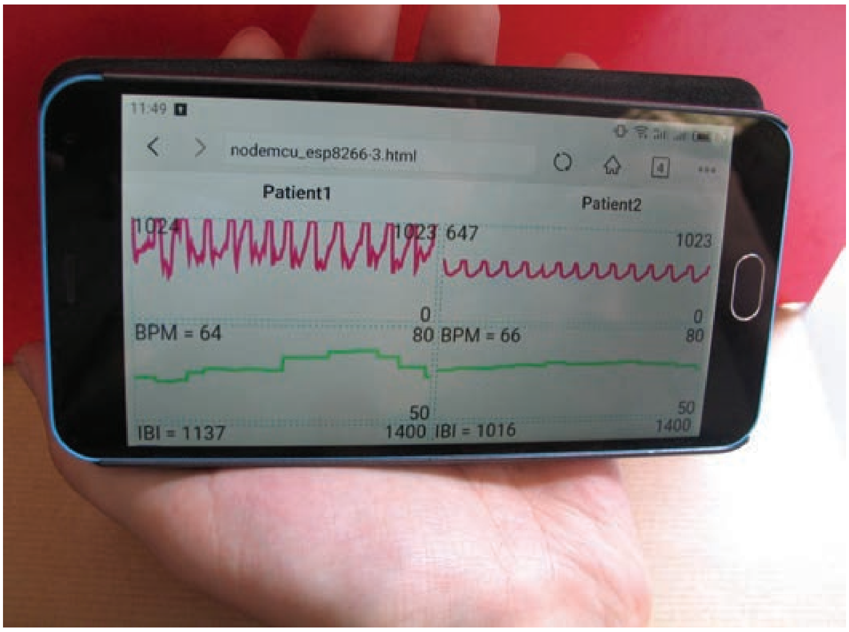
Monitoring of Multiple Patients

- 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**

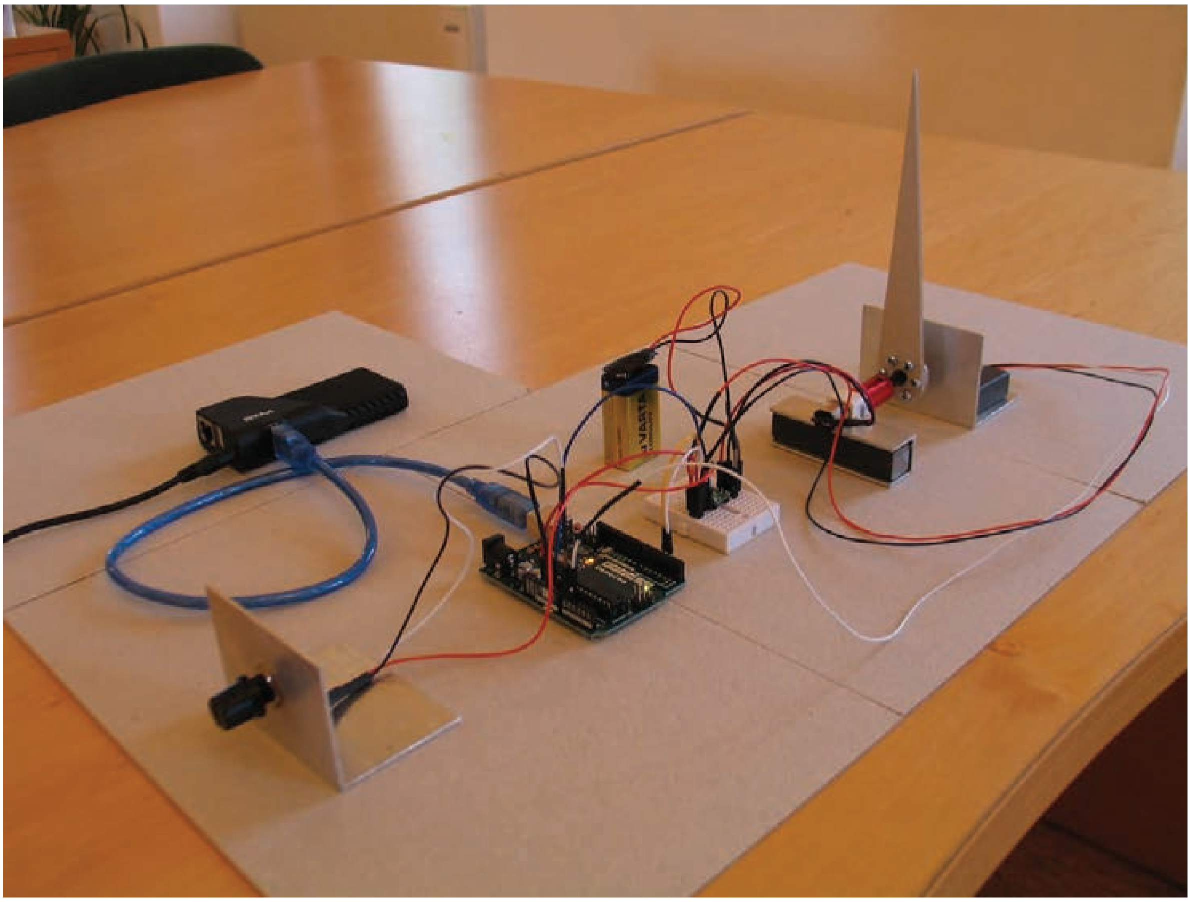
Power consumption (cont.)

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

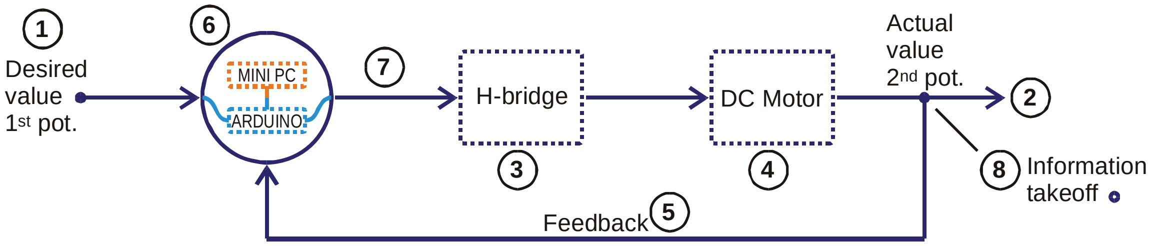
- Main difference in transmission (noTr/Tr) mode

Example M

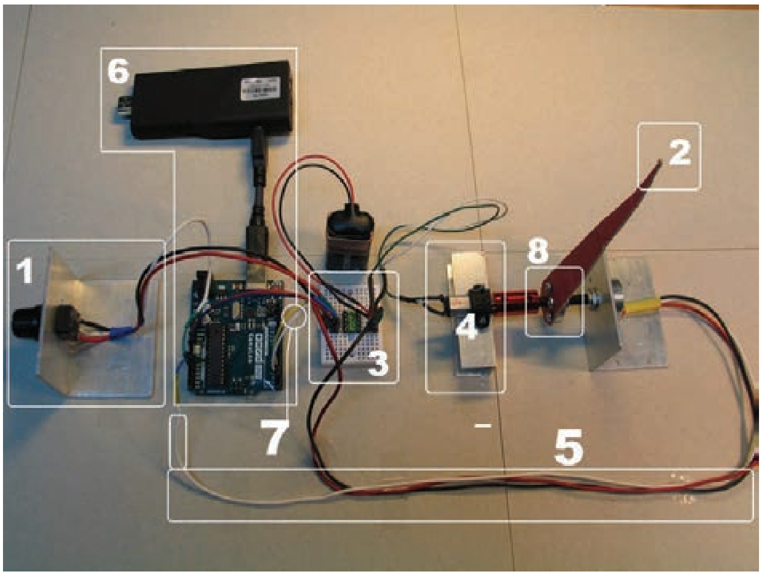
- Arduino
- Mini PC MK802V
- H-bridge
- DC motor
- Programming in browser
- Cloud9 IDE
- JavaScript/ ECMAScript



Example M (cont.)

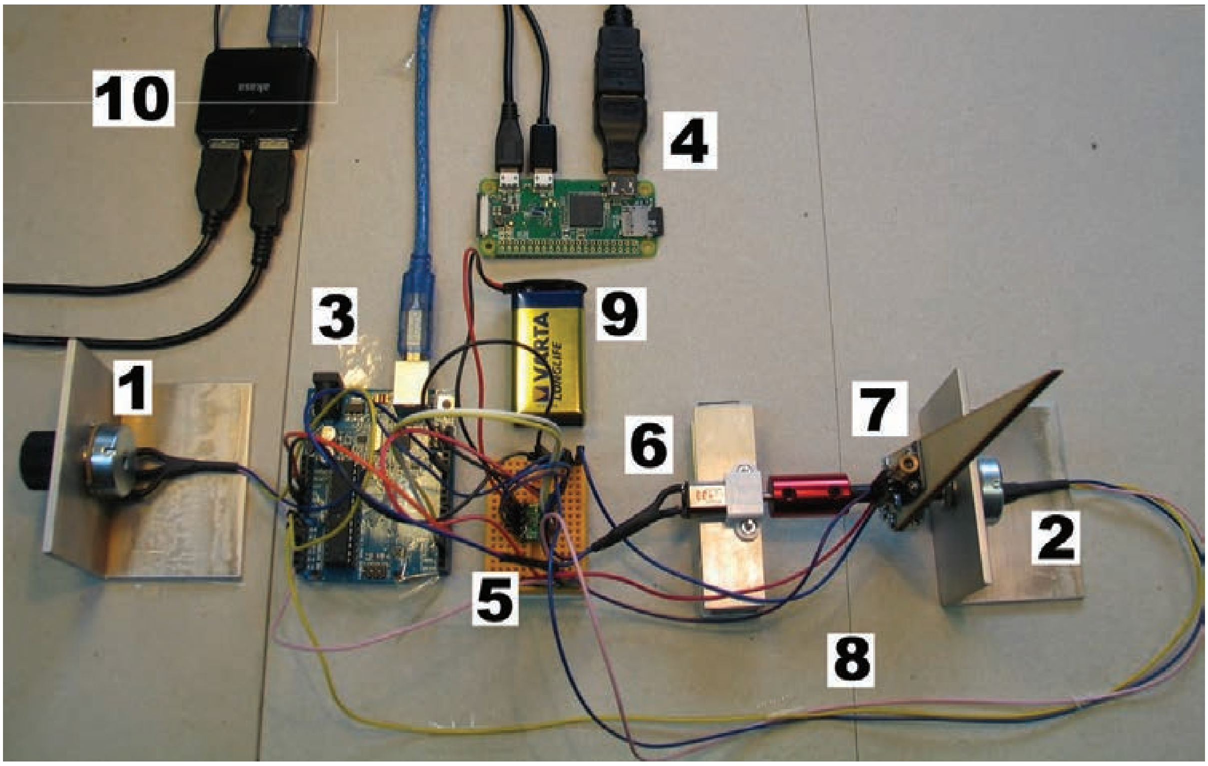


- Paralel with control system schematics



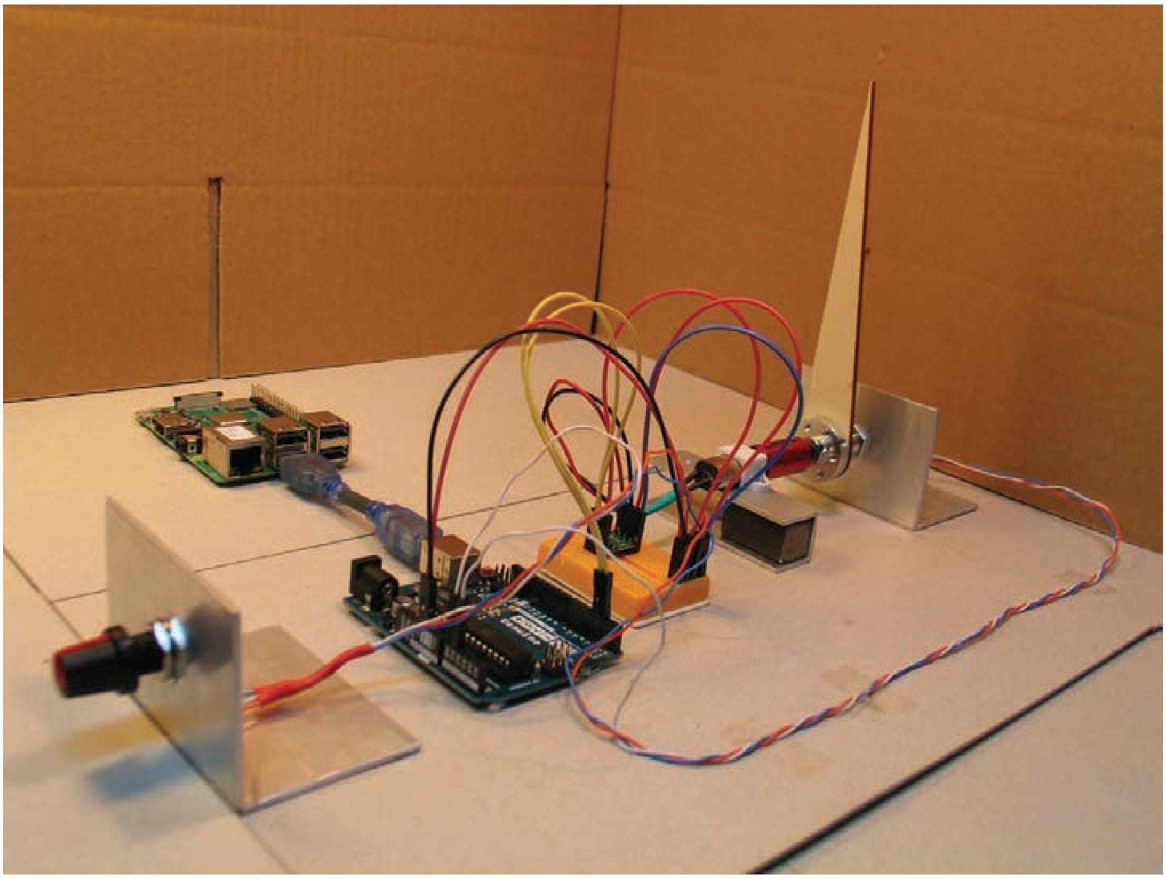
Example M

- Arduino
- rPi Zero w
- H-bridge
- DC motor
- Programming in browser
- Cloud9 IDE
- JavaScript/ ECMAScript

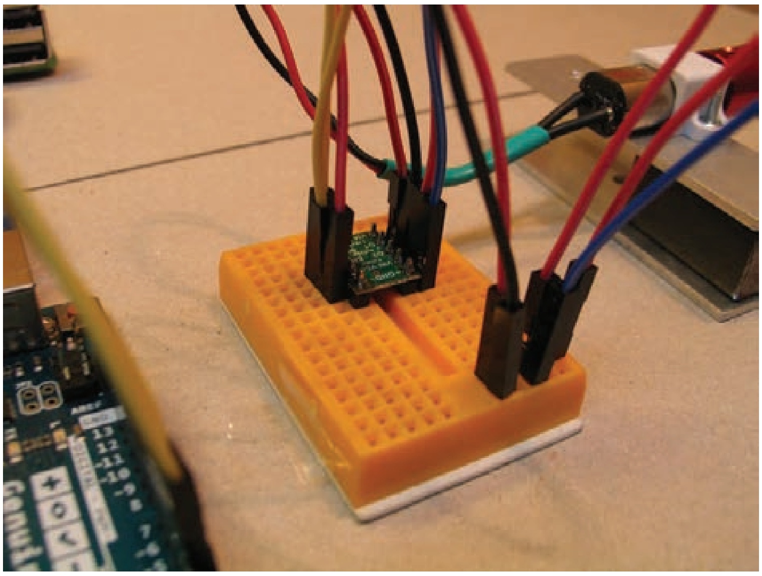


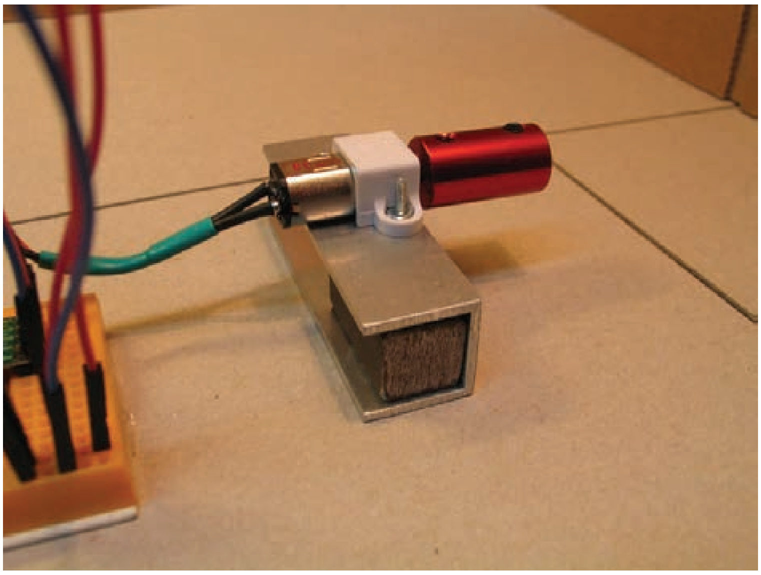
Example M

- Arduino
- rPi 3 Model B
- H-bridge
- DC motor
- Programming in browser
- Cloud9 IDE
- JavaScript/ ECMAScript

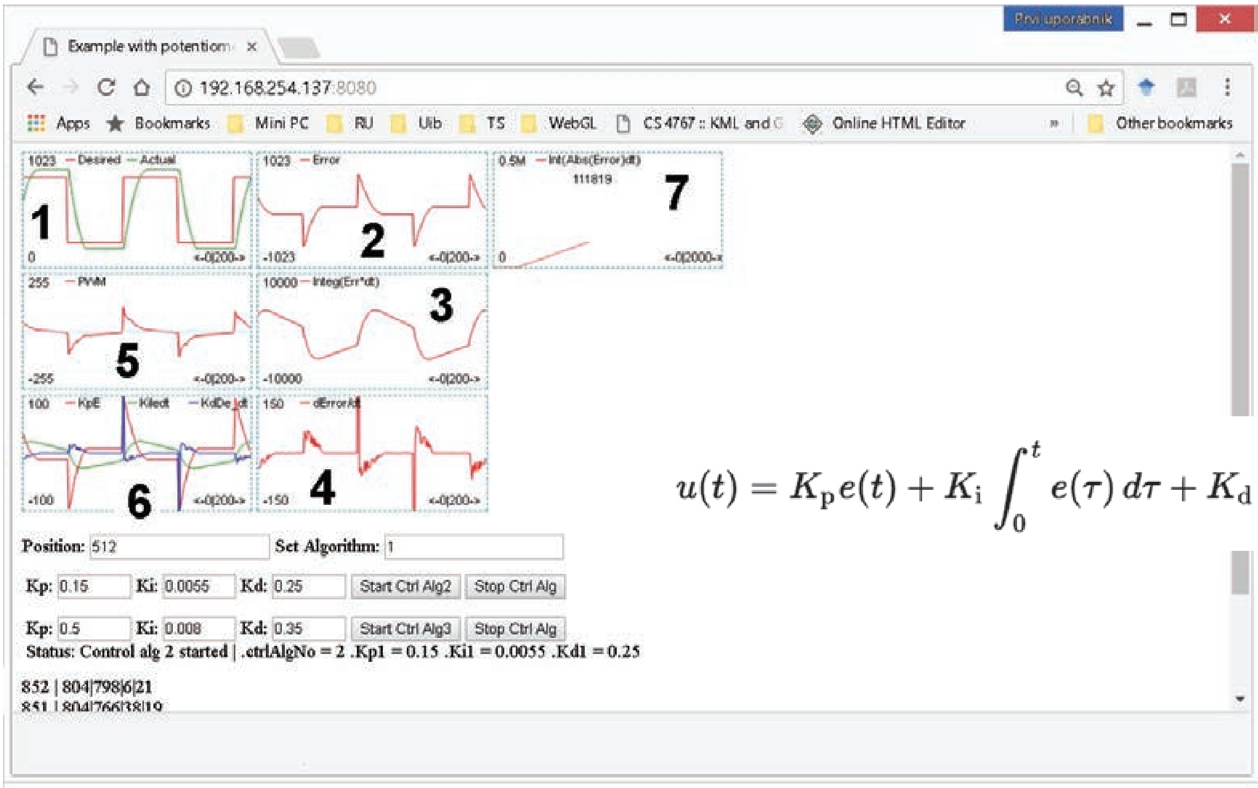


H-bridge and DC Motor Details





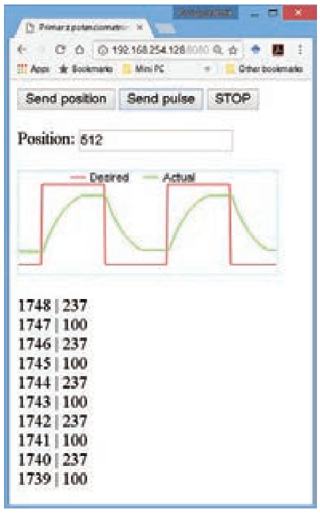
GUI for Monitoring Parameters

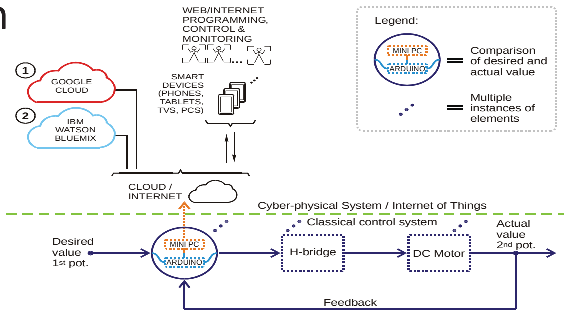


$$u(t) = K_{\text{p}} e(t) + K_{\text{i}} \int_0^t e(\tau)\, d\tau + K_{\text{d}} \frac{de(t)}{dt},$$

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



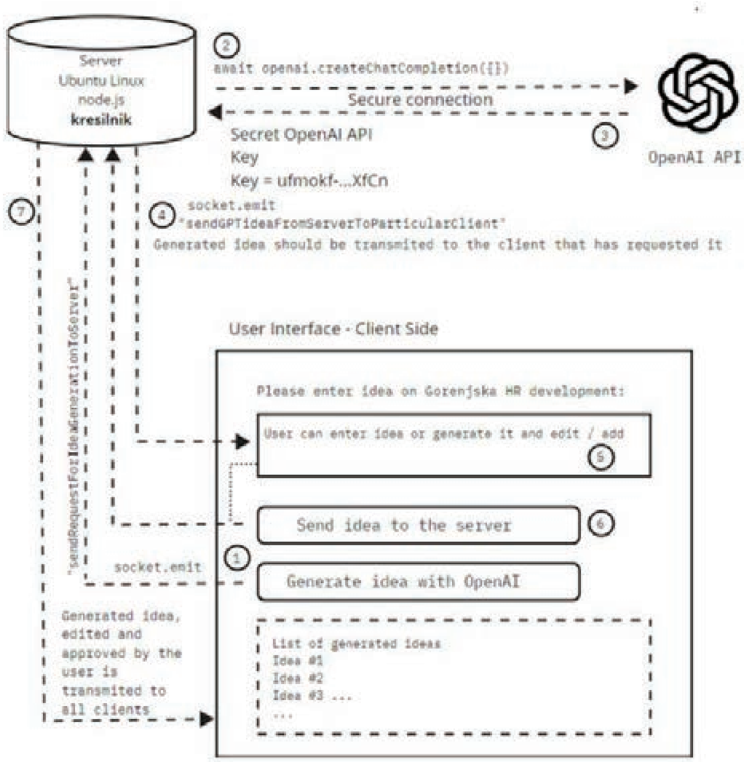


Transporting of Kits

- Kits can be used in standard class
- Power sockets (220V) should be provided
- WiFi network - router
- No other special equipment is needed

AI & CPS&IoT

- Application of artificial brain – new possibilities that should be explored
- Example of the interaction with OpenAI LLM API at the process of generating innovative ideas
- json as the main data structure
- Fast development
  doi: https://doi.org/10.3390/make5040065



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.

An appendix to the design of feasible health care wearables

Prof. dr Radovan Stojanović
University of Montenegro and MECOnet

Jovan Djurković, Dipl. el. ing.
MECOnet, www.meconet.me

5th Summer School on Cyber Physical Systems and
Internet of Things, Budva, Montenegro, June 2024

Overview

- Introduction
- Design issues and challenges
- Design examples (DE)
- Student Exercises (SE)
- Conclusion
- References

Introduction

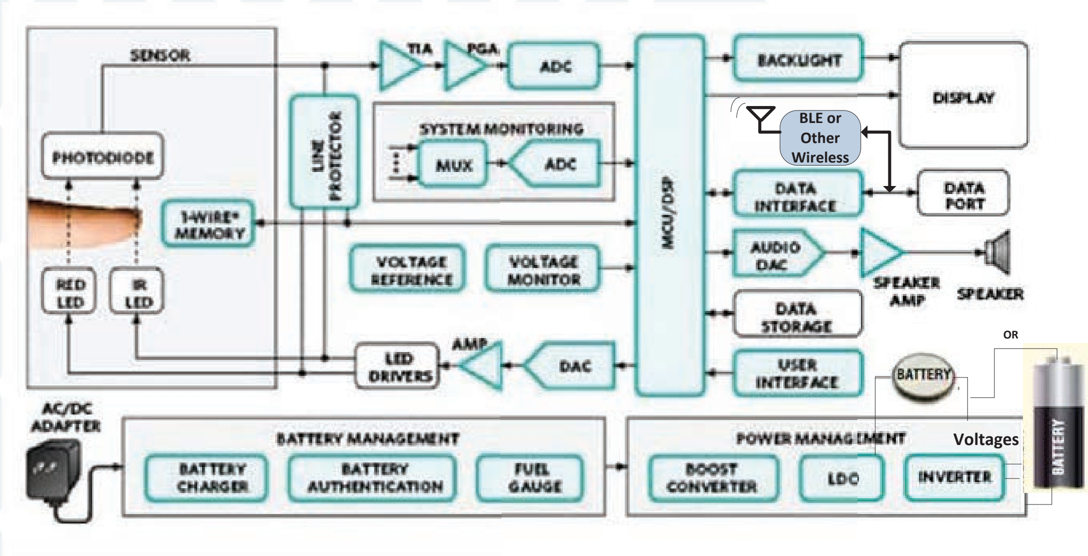
- Healthcare wearables (HeCaWe) are typical examples of embedded systems with elements of IoT and AI.
- All known strategies and knowledge in analog, digital, mixed and software world should be implemented in designing those devices.
- The medical standards are stronger than commercial and industrial, near to military, and those HeCaWe should be of required performances.
- However, today HeCaWe are unjustifiably expensive and with optimized design strategy their price should be reduced and thus they can be more spread-out to the population.
- Here we show some of the design strategy of HeCaWe based on off-the-shelf components we use in every day research and education processes.

Introduction

- Typical **HeCaWe** (client-host architecture)

Design issues and challenges

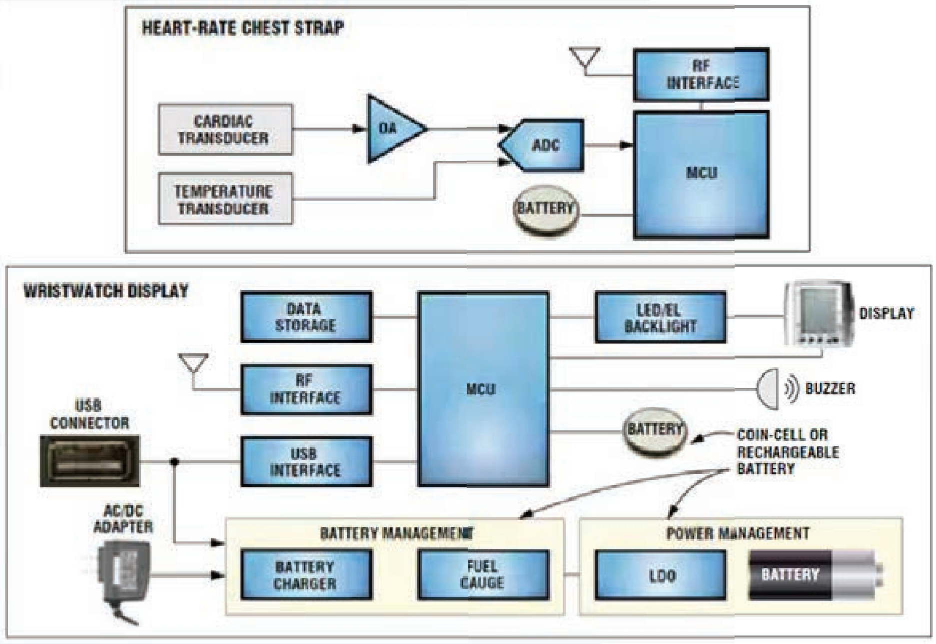
- Complexity of the client.
- All in one device architecture example
- The designee need take care about many different aspects, analog, MC, communication, software. Practically measurement+computer system in one device



HeCaWe Architecture. From Maxim electronics with author's modifications.
https://www.digikey.com/en/articles/reducing-wearable-health-fitness-device-design-time

Design issues and challenges

- The idea is to simplify the client by delegation as much as tasks to the host.
- Today, hosts are strong multifunctional devices in standalone or gadget forms, in good price and easy to use.
- The task delegation has its advantages and disadvantages.
- The advantages are numerous and most disadvantage is that the system become "non-user friendly" in term need several pieces.

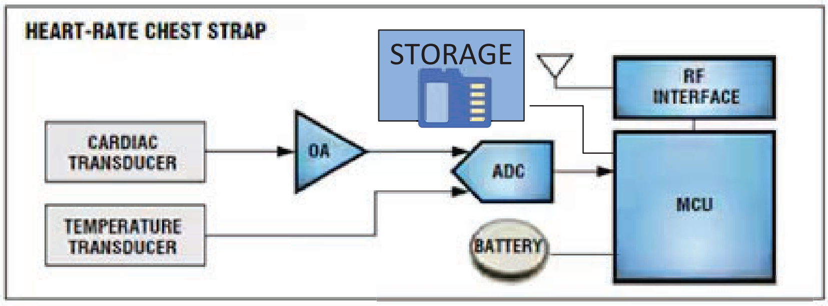


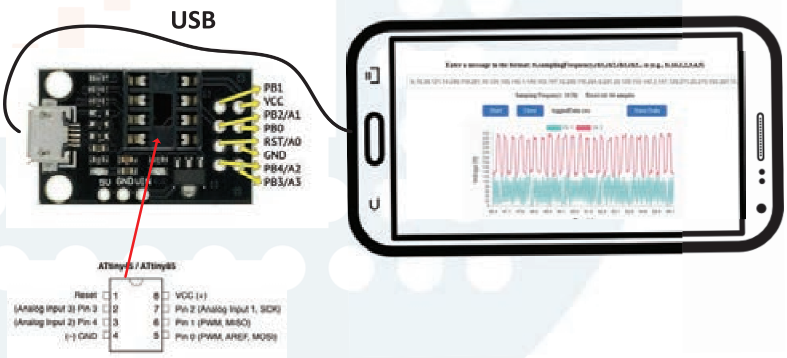
HeCaWe Architecture in different configurations in term of complexity. Different complexity options. From Maxim electronics with author's modification
https://www.digikey.com/en/articles/reducing-wearable-health-fitness-device-design-time

5th Summer School on Cyber Physical Systems and Internet of Things, Budva, Montenegro, June 2024

Design issues and challenges

- The examples of the client simplifications.
- The client can be simplified to one-chip device in form of general purpose MCU or ASIC as it is case of using 8 pins-8 bits tiny MC, Attiny 85, integrated in USB module, case of Digistump.
- The designer's skill is reflected in how skillfully and quickly he can assemble the available modules, and if one cannot fit, the designer should quickly make an adapter for its fitting.



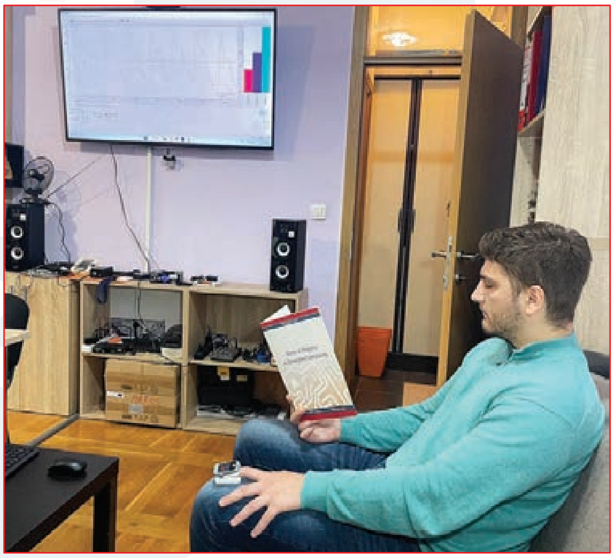


8 pin chip connected to the Gadget (smart phone) can do vital signs monitoring.
The client is in the form of Digistump (https://github.com/digistump/DigistumpArduino) development board featuring ATtiny85 for sensing and preprocessing. The communication with host is done by PS2 keyboard emulator. The signals are sent to the application as keyboard ASCII signs. By parsing the strings signals are visualized and features extracted.

Design examples (DE1)

- Commercial pulse Oximeter connected to the gadget
- Commercial oximeter with BLE (https://shberrymed.com/products/fingertip-pulse-oximeter-bm1000b-60). The BLE-USB Gateway is based on ESP32 node.
- HOST can be any PC or Gadget with custom design or general purpose GUI. The sending protocol given by manufacturer is parsed.

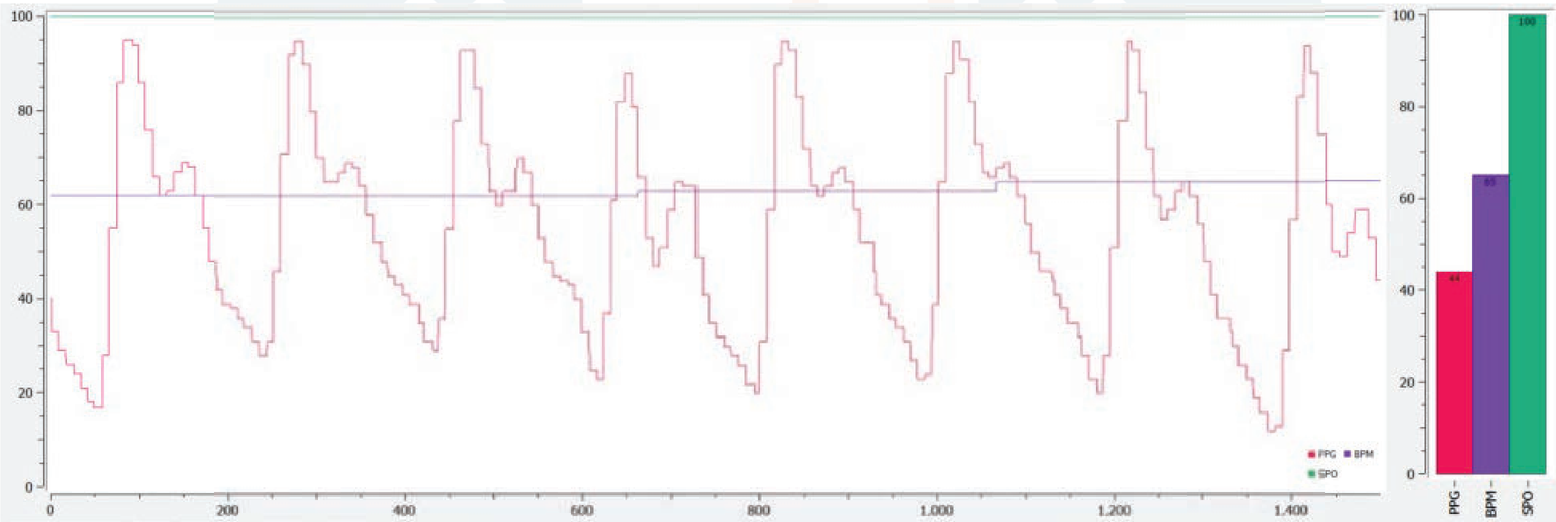
HOST



5th Summer School on Cyber Physical Systems and Internet of Things, Budva, Montenegro, June 2024

Design example (DE1)

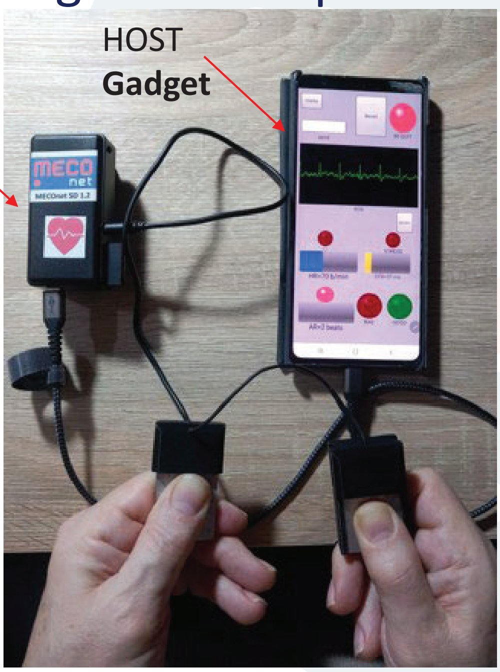
- On the host side we can handle and analyze the signals by bundle of commercial software, standard graphical plotters/emulators, MATLAB, LabVIEW etc… https://hackaday.io/project/5334-serialplot-realtime-plotting-software/log/192838-serialplot-v012-release



Design examples (DE2)

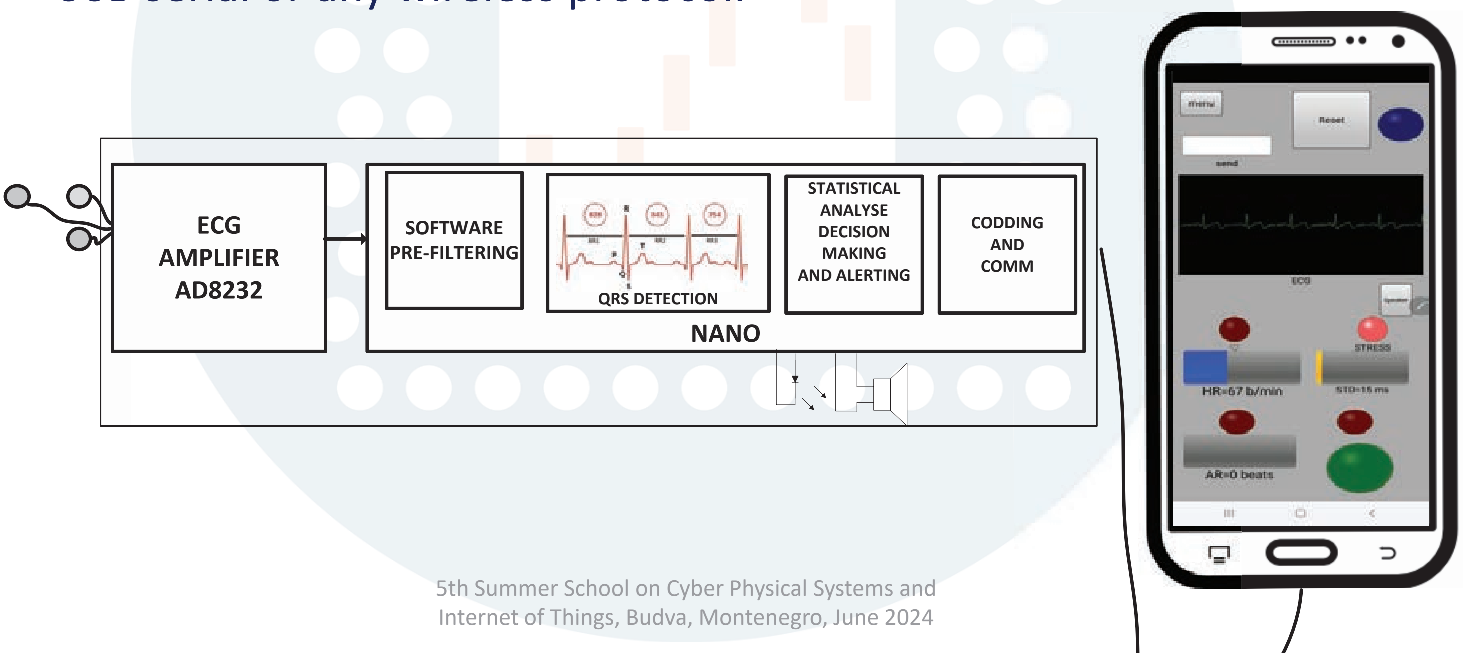
- Stress detector, autonomous device and connected to gadget. GUI implemented in https://roboremo.app/

Client



Design examples (DE2)

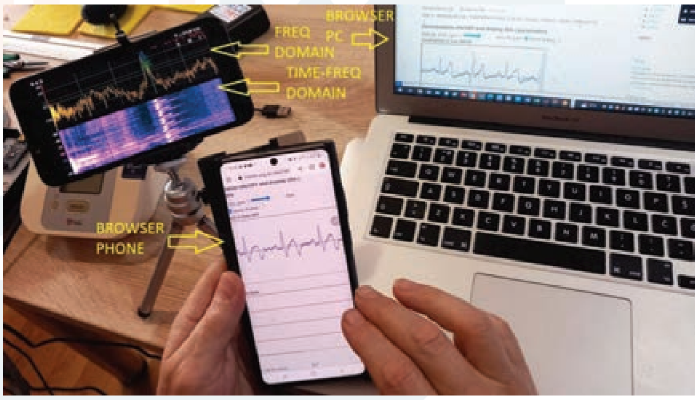
- Client (stress detector) can work separately, when LED and speaker indicate the state of the parameters, More detailed analyze has been done by gadget software, low cost. Client communicate with host by USB serial or any wireless protocol.



5th Summer School on Cyber Physical Systems and Internet of Things, Budva, Montenegro, June 2024

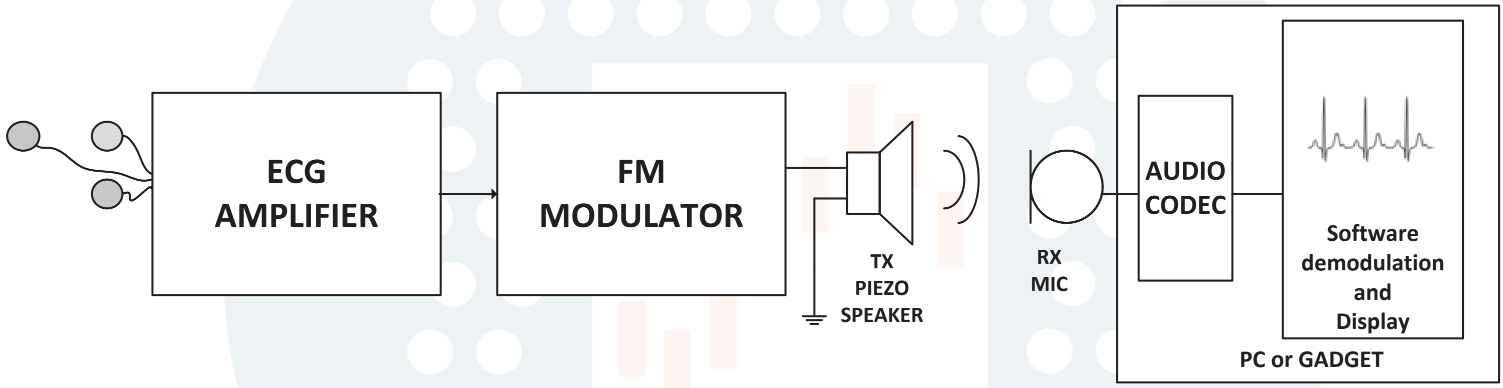
Design example (DE3)

- "Whispering heart". The simplest client, delegating more functions to the host. The communication is done in near ultrasound range.



Design example (DE3)

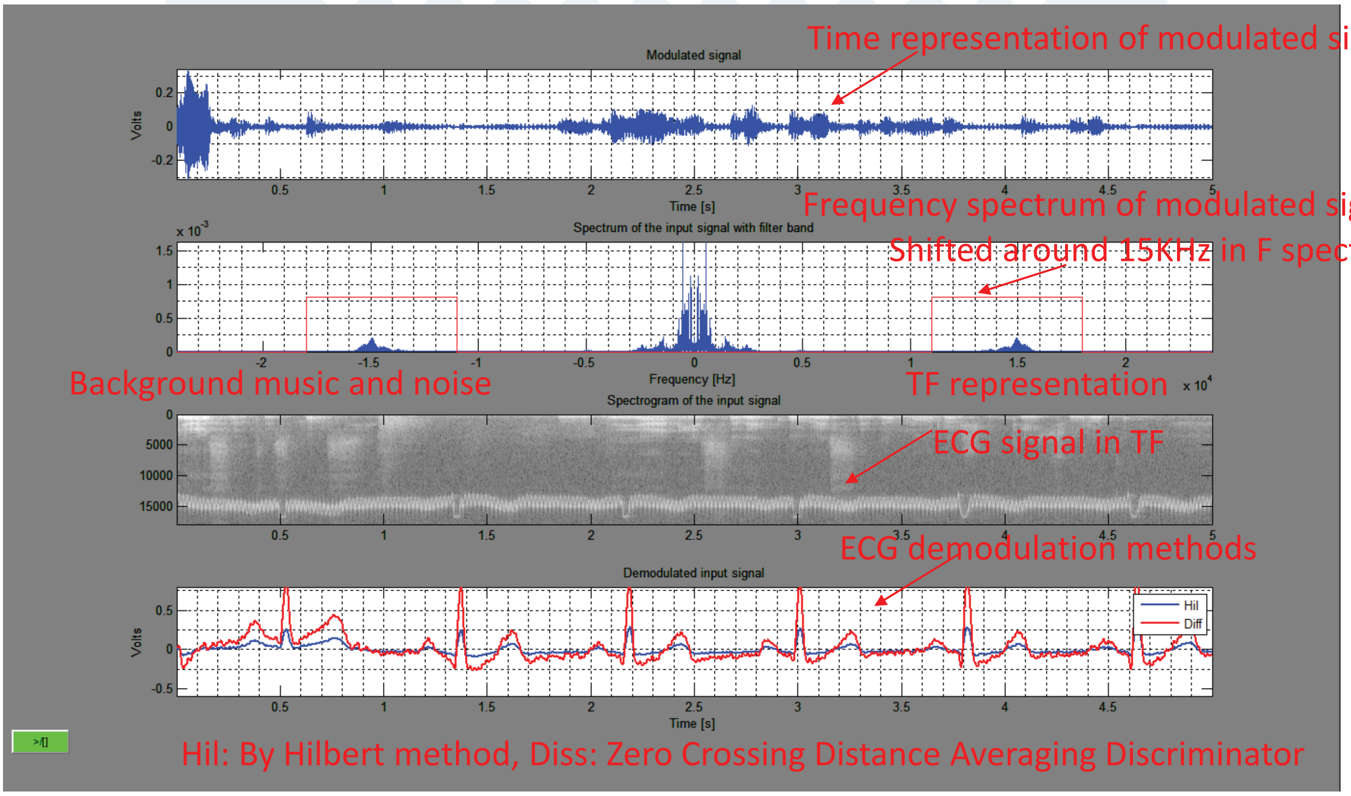
- Operational principle



- By FM modulation signal translated in near ultrasound band. Then signal is observed in time, frequency and time-frequency domains by host software implemented in MATLAB or JavaScript.
- Different ways of demodulation by Hilbert transform and frequency discriminator

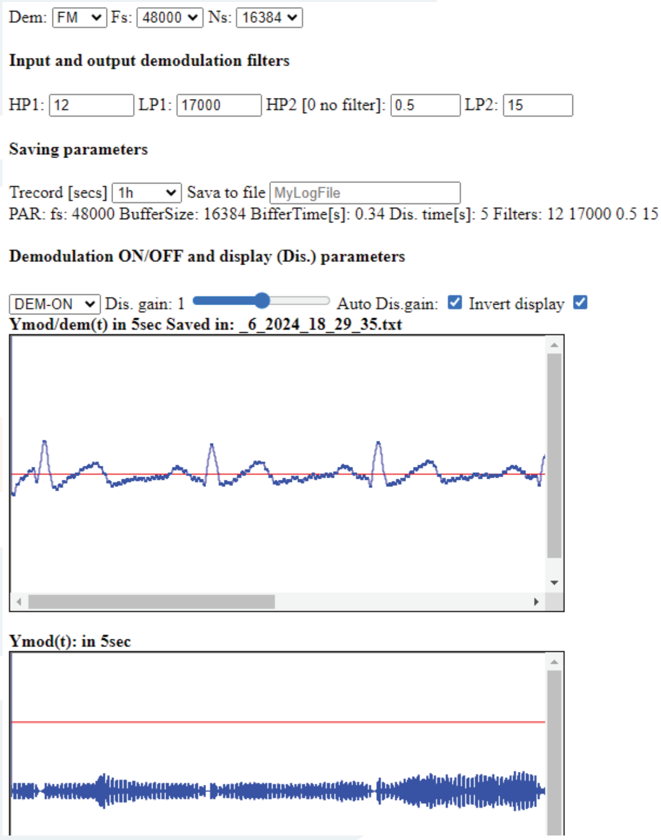
Design example (DE3)

- The analysis of the modulated ECG signal in the different domains: (T (time), (F, frequency) (TF – time/frequency). The aim is to shift source signal in higher frequency band, transmit it and demodulate.



Design example (DE3)

- Signal processing and signal demodulation in JavaScript.
- Platform independent web based processing using WEB audio API.



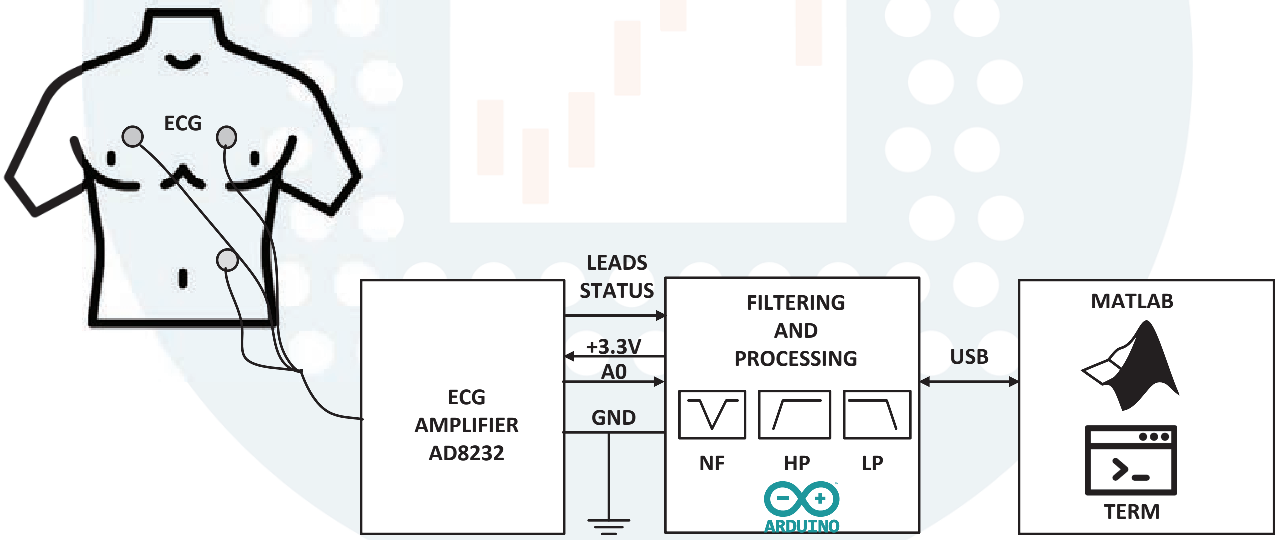
5th Summer School on Cyber Physical Systems and Internet of Things, Budva, Montenegro, June 2024

Design examples (more…)

- MedWearables from MECOnet,
- https://meconet.me/smarthealth/
- Syntrofos – A headset like wearable device to track COVID-19 symptoms
- Wireless and hands free vital signs monitors
- Portable monitors, stress and arrhythmia analyzers
- "Whispering heart", proximity contactless, browser-based ECG analyzer
- Wireless pulse oximeter monitor
- Fall detection using beacon and node MCU
- True measurement of blood pressure
- HRV analysis and arrhythmia detection

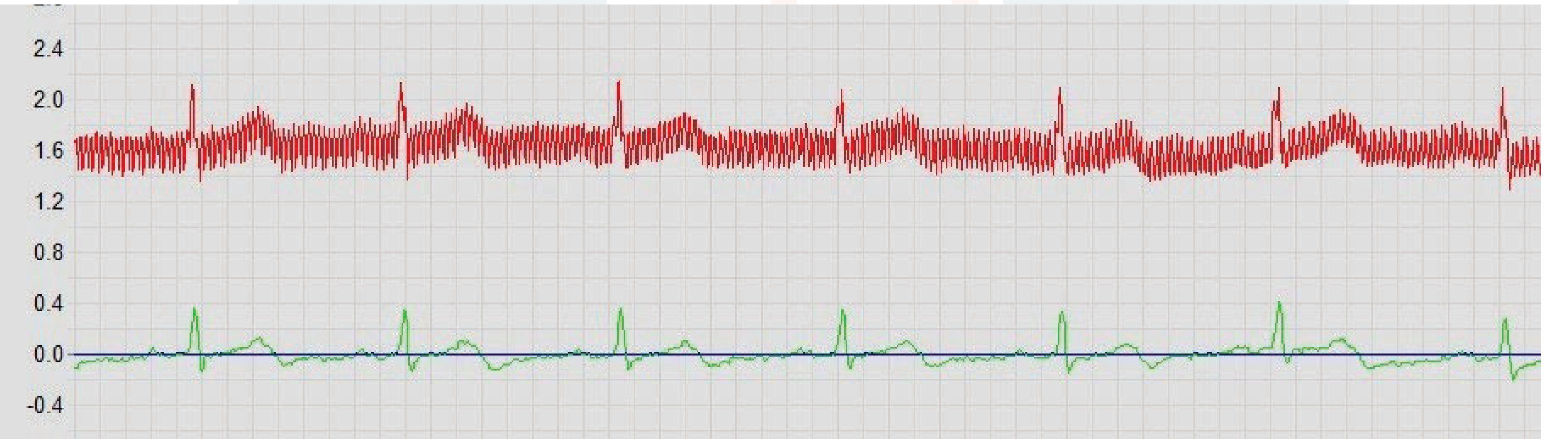
Exercise #1

- How to design the acquisition, software filtering, serial sending and visualization of ECG signal by using AD8232 ECG module, ARDUINO NANO and serial plotter or MATLAB GUI. Notch, HP and LP filtering are implemented in ARDUINO NANO. Observe the signals in Terminal-Plotter? Implement the same filtering in the MATLAB? Use the [Sources for Matlab and NANO files.] (ex1.m, MATLAB GUI, soft_filter_nano_ecg.ino, ARDUINO code)



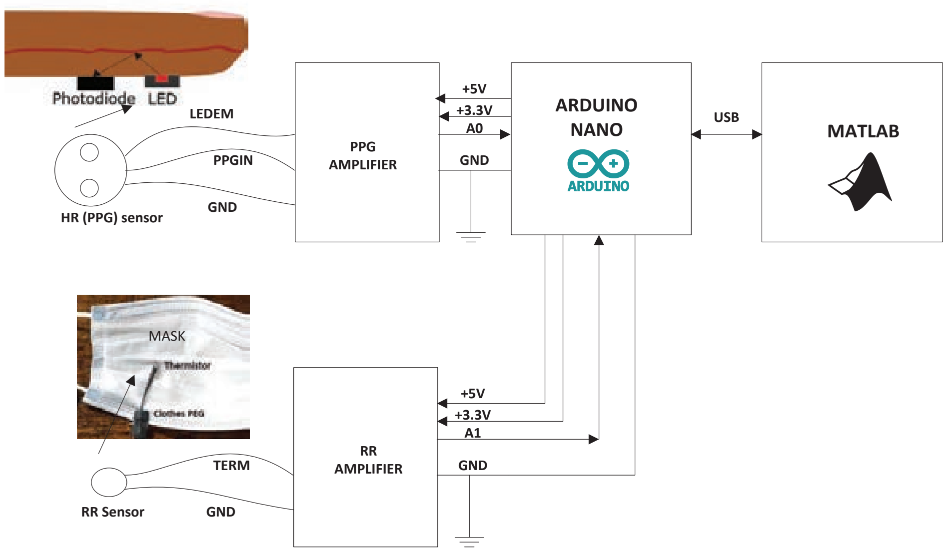
Exercise #1

- The effect of the software filtering by cascade Notch (50Hz) -> LP(25Hz) -> HP(0.5Hz), observed in serial oscilloscope, observed in free version https://x-io.co.uk/serial-oscilloscope/



Exercise #2

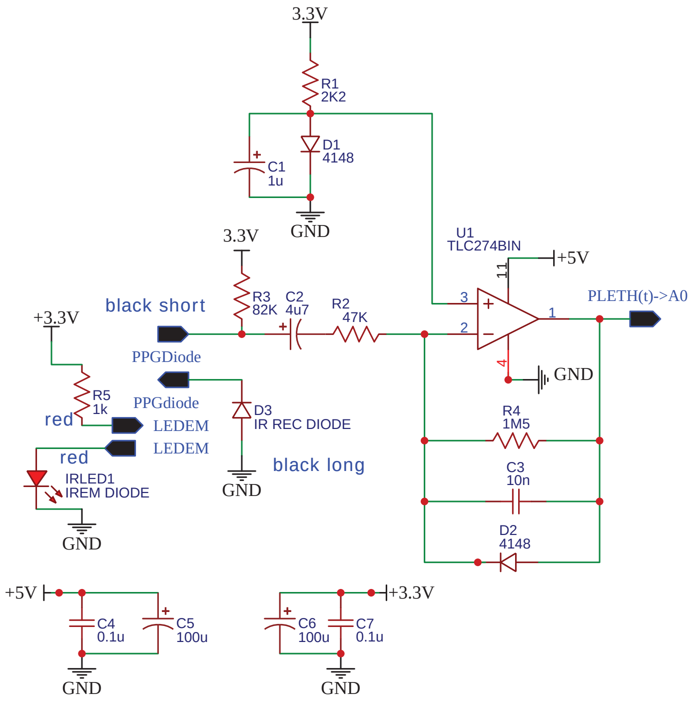
- Design of feasible HR – Heart Rate and RR - Respiration Rate monitor, by using ARDUINO NANO, low-cost sensors, one chip (2 OAs) analog front and their processing in MATLAB based GUI.



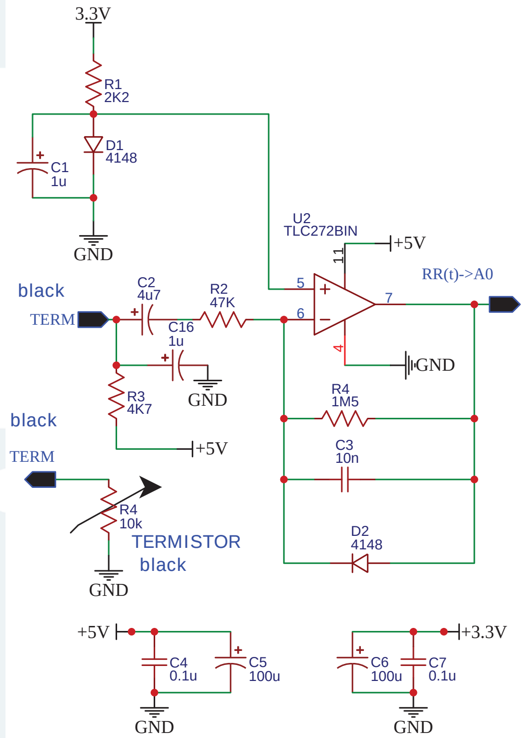
5th Summer School on Cyber Physical Systems and Internet of Things, Budva, Montenegro, June 2024

Exercise #2

- The schematics of HR and RR amps



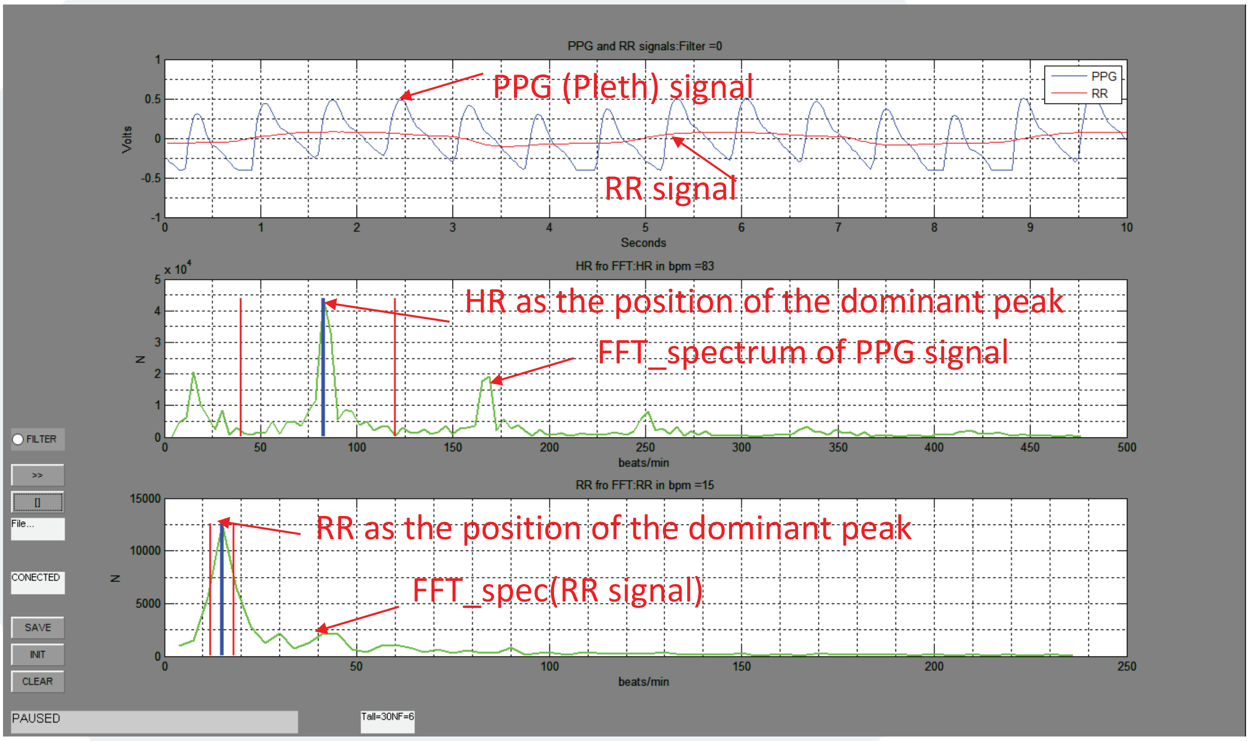
PPG AMPLIFIER



RR THERMISTOR AMPLIFIER

Exercise #2

- The PPG and RR signals in time and frequency domains. Extraction HR and RR from FFT spectrums, as the positions of dominant peaks.
- Software for Ex2 (ex1.m MATLAB GUI, ecg_pleth_rr_t_m.ino, ARDUINO NANO program)



CONCLUSION

- Performance effective medical wearables can be designed in open hardware/software by system integration of the existing low-cost modules or chip-sets in combination with available software tools.
- Sometimes, where it is necessary, we should to design and integrate hardware/software adaptors (gateways).
- The advantages of such solutions, inter alia, are low-cost, open-hardware/software, infinity possibilities in signal processing and storage, easy integration in local and telemedicine tools.
- The main disadvantage is necessity to use additional device gadget (smart phone…) or PC.

References

1. Radovan Stojanović, Jovan Djurković, Andrej Škraba, ECG and PPG Signals Monitoring by Using Web Audio API, In 2024 13th Mediterranean Conference on Embedded Computing (MECO) (pp. 461-465). IEEE.

2. Radovan Stojanović, Jovan Djurkovic, Blagoje Babić, Veselin N. Ivanović, Budimir Lutovac and Milan Stork, A Toolset for Blood Pressure Visualization and Measurement in Time, Frequency and TimeFrequency Domains, In 2024 13th Mediterranean Conference on Embedded Computing (MECO) (pp. 420-425). IEEE.

3. Djurkovic, J., Stojanović, R., & Cico, B. (2023). An Experimental Platform for Fall Detection Using Beacon, Node MCU and MATLAB. *WiPiEC Journal-Works in Progress in Embedded Computing Journal*, *9*(2).

4. Stojanović, R., Djurković, J., Mijušković, S., Lutovac, B., & Škraba, A. (2023, June). SYNTROFOS: A Wearable Device for Vital Sign Monitoring, Hardware and Signal Processing Aspects. In 2023 12th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-6). IEEE.

5. Stojanović, R., Škraba, A., Djurković, J., & Lutovac, B. (2022, June). Off-the-Shelf Solution for Measurement and Calculation of Respiration and Heart Rates for COVID-19 Diagnosis and Monitoring. In 2022 11th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-5). IEEE.

6. Stojanovic, R., & Skraba, A. (2021, June). Simplified open HW/SW pulse oximetry interface for purpose of COVID-19 symptoms detection and monitoring. In 2021 10th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-5). IEEE.

7. Stojanović, R., Škraba, A., & Lutovac, B. (2020, June). A headset like wearable device to track COVID-19 symptoms. In 2020 9th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-4). IEEE.

8. Stojanović, R., Hagara, M., Ondracek, O., & Caplanova, A. (2015, June). Addressing the need for practical exercises in biomedical engineering education for growing economies. In *2015 4th Mediterranean Conference on Embedded Computing (MECO)* (pp. 416-421). IEEE.

9. Stojanović, R., & Karadaglić, D. (2011, June). An economical and feasible teaching tool for biomedical education. In 2011 24th International Symposium on Computer-Based Medical Systems (CBMS) (pp. 1-5). IEEE.

10. Stojanovic Radovan, Challenging issues in cost effective wearable and IoT medical devices with example to Covid19, in Proceedings of the 2nd Summer School on Cyber-Physical Systems and Internet-of-Things, 2021, pp. 67-89, doi: 10.5281/zenodo.5086365

11. Radovan Stojanović, Design of performance and energy efficient nodes for smart systems, in Lech Jóźwiak, Radovan Stojanovic, & Christos Antonopoulos. (2023). Proceedings of the 4th Summer School on Cyber-Physical Systems and Internet-of-Things, Vol. IV, 2023 (1.0) [Computer software]. 4th Summer School on Cyber-Physical Systems and Internet-of-Things (SS-CPSIoT2023), Budva, Montenegro. Zenodo. https://doi.org/10.5281/zenodo.8113313

5th Summer School on Cyber Physical Systems and
Internet of Things, Budva, Montenegro, June 2024


Your digital way since 2012
Questions, comments?
5th Summer School on Cyber Physical Systems and
Internet of Things, Budva, Montenegro, June 2024

CPS&IoT'2024 Summer School on
Cyber-Physical Systems and Internet-of-Things
Budva, Montenegro, June 11-14, 2024

Schedule

Day 1, Tuesday 11 June:
09:00-09:15 Event Chairs and Special Guests
Title: **Opening Ceremony of the CPS&IoT'2024 Summer School, and MECO'2024 and CPS&IoT'2024 Conferences**
09:30-10:30 Tarek El-Ghazawi, George Washington University, US
Keynote: **The Future of Computing: From ExaFLOPS to Exotic Processor Technologies**
10.30-11.00 Break
Title: Introduction to the CPS&IoT'2024 Summer School
11.00-13.00 Lech Jóźwiak, TU/e, **NL**
Title: Green CPS and IoT for Green World
13.00-15.00 Lunch Break
Title: Quality-driven Design of Cyber-Physical Systems
15.00-17.00 Nikhil Gaikwad and Sokol Kosta, Aalborg University, **DK** and Ralf Lübben, Flensburg Univ. of Appl. Sciences, **DE**
Title: GPU virtualization service for AI at the Edge
17.00-17.30 Break
17.30-19.30 Francesco Ratto, Federico Manca and Claudio Rubattu, UNISS, **IT**
Title: Adaptive CNN execution on Edge FPGAs
21.00 Dinner

Day 2, Wednesday 12 June:
09.00-10.30 Nabil Abdennadher, Univ. of Applied Sciences, West. Switzerland, **CH**
Title: **Keynote**: **Towards a Distributed Continuum Computing Platform for Federated Learning Based Self-Adaptive IoT Applications**
10.30-11.00 Break
11.00-13.00 Alberto Marchisio and Muhammad Shafique, New York University Abu Dhabi, **UAE**
Title: Energy-Efficient and Robust Deep Learning for Autonomous Systems
13.00-14.00 Lunch Break
14.00-17.00 Nabil Abdennadher
Title: Distributed Cloud Continuum Platform for Federated Learning Based Self-Adaptive IoT Applications (hands-on and demo tutorial)
17.00-17.30 Break
17.30-18.30 Alberto Marchisio and Muhammad Shafique, New York University Abu Dhabi, **UAE**
Title: Design Space Exploration of Efficient Quantum Machine Learning Systems

Day 3, Thursday 13 June:
09.00-10.30 Rainer Leupers, RWTH Aachen, DE
Title: **Keynote**: **Multicore Design Technologies and HW Security – From Academia to Industry**
10.30-11.00 Break
11.00-13.30 Rakshit Mittal and Hans Vangheluwe, University of Antwerp, **BE** and Rizwan Parveen, Telecom Paris, **FR**
Title: Modeling a Cruise-Control System Using Open Modelica and Verifying Safety Requirements using UPPAAL (hands-on tutorial)
13.30-14.30 Lunch Break
14.30-17.00 Dominique Blouin, Telecom Paris, and Anish Bhobe, Institut Polytechnique de Paris, **FR**
Title: Modeling and Synthesizing a Cruise-Control System with AADL using RAMSES (hands-on tutorial)
17.00-17.30 Break
17.30-19.00 Samir Ouchani, CESI, **FR**
Title: Smart CPS: Ensuring Trustworthiness in Autonomous Decisions through Formal Methods

Day 4, Friday 14 June: (The Break will be within presentations)
09.00-10.30 Christoph Schmittner, AIT, **AT**
Title: Cyber-Physical System Security: Automated Risk Management with ThreatGet
10.30-12.00 Morayo Adedjouma and Luis Palacios, CEA, **FR**
Title: Trustworthy Design and V&V of AI-based systems: Case of a Drone Application (includes demo and hands-on)
12.00-13.00 Lunch Break
13.00-14.30 Zakaria Chihani, CEA, **FR**
Title: Trustworthy AI: methods and tools
14.30-16.30 Andrej Škraba, University of Maribor, **SI**
Title: Overview of Several CPS&IoT Prototypes
16.30-18.30 Radovan Stojanovic, University of Montenegro and MECOnet, **ME**
Title: An appendix to the design of usable and low-cost nodes for biomedical applications

18.30-19.00 Closing of the CPS&IoT'2024 Summer School

+ Free participation in sessions of the CPS&IoT'2024 Conference and MECO'2024 Conference
Summer School participants are expected to come with their own laptops. Internet access will be guaranteed.

Day 5, Saturday 15 June: Excursion possible (excursion fee is not included in the summer school fee)

ZOOM LINK FOR SUMMER SCHOOL:
https://us06web.zoom.us/j/8342653096?pwd=ZUFOaHppdXVkc1lWRDNTSnNIYmF4UT09

Summer School on CPS&IoT 2024 – Attendees

#	Students	Country	Affiliation
1	Roald Van Glabbeek	Belgium	Vrije Universiteit Brussel
2	Diana Deac	Belgium	Vrije Universiteit Brussel
3	Alberto Galassi	Italy	University of Studies of L'Aquila
4	Tiago Fonseca	Portugal	ISEP – School of Engineering, Polytechnic of Porto
5	Giuseppe Spadavecchia	Italy	Polytechnic University of Bari
6	Paulo Carvalho	Germany	AOX GmbH
7	Diego Liberati	Italy	Politecnico di Milano
8	Suryansh Sharma	Netherlands	Delft University of Technology (TU Delft)
9	Hossein Khalilnasl	Italy	University of Brescia
10	Matija Šuković	Montenegro	University of Montenegro
11	Velibor Došljak	Montenegro	University of Montenegro
12	Marija Džaković	Montenegro	University of Montenegro
13	Anđela Pantović	Montenegro	University of Montenegro
14	Anđela Iković	Montenegro	University of Montenegro
15	Milica Rajčić	Montenegro	University of Montenegro
16	Sara Milinković	Montenegro	University of Montenegro
17	Dragana Zorić	Montenegro	University of Montenegro
#	**Lecturers**	**Country**	**Affiliation**
1	Tarek El-Ghazawi	United States	George Washington University
1	Lech Jóźwiak	Netherlands	TU/e
2	Radovan Stojanović	Montenegro	University of Montenegro and MECOnet
3	Nikhil Gaikwad	Denmark	Aalborg University
4	Ralf Lübben	Germany	Flensburg Univ. of Appl. Sciences
5	Sokol Kosta	Denmark	Aalborg University
6	Francesco Ratto	Italy	UNISS
7	Federico Manca	Italy	UNISS
8	Claudio Rubattu	Italy	UNISS
9	Alberto Marchisio	UAE	New York University Abu Dhabi
10	Muhammad Shafique	UAE	New York University Abu Dhabi
11	Nabil Abdennadher	Switzerland	Univ. of Applied Sciences, West Switzerland
12	Rainer Leupers	Germany	RWTH Aachen
13	Rakshit Mittal	Belgium	University of Antwerp
14	Hans Vangheluwe	Belgium	University of Antwerp
15	Rizwan Parveen	France	Telecom Paris
16	Dominique Blouin	France	Telecom Paris
17	Anish Bhobe	France	Institut Polytechnique de Paris
18	Samir Ouchani	France	CESI
19	Morayo Adedjouma	France	CEA
20	Luis Palacios	France	CEA
21	Zakaria Chihani	France	CEA
22	Andrej Škraba	Slovenia	University of Maribor
23	Jovan Đurković	Montenegro	MECOnet

SS-CPSIoT

Certificate

OF ATTENDANCE

THIS CERTIFICATE ACKNOWLEDGES THAT

Marko Markovic

MONTENEGRO

MONTENEGRIN ASSOCIATION FOR NEW TECHNOLOGIES - MANT

Has successfully attended and completed:

The 5th Summer School on

Cyber Physical Systems and Internet of Things (SS-CPSIoT'2024)

(3 ECTS)

On behalf of the organizers:

Prof. dr. Lech Jozwiak Lech Jóźwiak

Prof. dr. Radovan Stojanović

In Budva, Montenegro, June 11-14, 2024

Author Index

SS-CPS&IoT 2024 Gallery

CPS&IoT'2024 5th Summer School on Cyber-Physical Systems and Internet-of-Things

Budva, Montenegro, June 11-14, 2024

Citation:

Author/s, "Title of contribution-presentation", in Proceedings of the 5th Summer School on Cyber-Physical Systems and Internet-of Things, Editors: Lech Jozwiak and Radovan Stojanovic, Authors: Alberto Marchisio, Andrej Škraba, Anish Bhobe, Claudio Rubattu, Dominique Blouin, Federico Manca, Francesco Ratto, Hans Vangheluwe, Jovan Đurković, Lech Jóźwiak, Luis Palacios, Morayo Adedjouma, Muhammad Shafique, Nabil Abdennadher, Nikhil Gaikwad, Radovan Stojanović, Rakshit Mittal, Ralf Lübben, Rizwan Parveen, Samir Ouchani, Sokol Kosta and Zakaria Chihani, Vol. V, June 2024, pp. xx-yy, DOI: https://doi.org/10.5281/zenodo.12608092.

Technical editors:

Prof. dr Budimir Lutovac, University of Montenegro
Jovan Djurkovic, MECOnet

Publishers:

MECO Events and Publishing, https://pubs.embeddedcomputing.me/

MECOnet Institute (Mediterranean Excellence in Computing and Ontology), https://meconet.me

Montenegrin Association for New Technologies – MANT, https://mant.me

Place/Year: Podgorica, Montenegro, 2024