Challenges

- Observability, testability, and predictability of CPS behaviour highly limited with real world consequences

- Contemporary DevOps practices and tools potentially right solution, but currently not developed for CPS domains

Vision

- Develop novel DevOps tools, methodologies, and techniques that enable effective, continuous development and evolution of CPS

- Increase the level of reliability, dependability, trustworthiness, and adaptability of CPS

- Delivers proven DevOps advantages and benefits to Europe’s CPS development community

Project Overview
Lessons Learned and Success Stories

- **DevOps Pipelines for CPS**
  - Catalogue of practices and challenges, toolkit for bad practice identification
  - Generative tools to support the definition of CI/CD pipelines

- **V&V and Security Assessment of DevOps pipelines**
  - Combining reinforcement learning + metaheuristic search leads to effective detection of safety requirement failures
  - Adoption of a runtime verification framework in industrial settings
  - Metamorphic security testing automates the detection of 100+ vulnerability types

- **DevOps Tools for CPS Software Evolution**
  - Classification and automated detection of performance antipatterns in CPS via static analysis and data mining techniques
  - Improving cost effectiveness of regression testing for CPS by combining evolutionary intelligence with principal component analysis
  - Real-world maps and Bézier curves to generate test scenarios for self-driving cars

- **Self-healing and Self-adaptability Tools for CPS**
  - 50% reduction testing costs for CPS via Digital Twins and AI-based regression testing approaches in DevOps pipeline
  - Automated field tests replication as well as change-based prediction/monitoring of critical CPS changes via DevOps pipeline
Recommendations for the Future

- Regulatory improvements
  - More dynamic (i.e. Agile and DevOps based) certification techniques and regulations for agile CPS
    - Support for CI/CD pipeline development
    - Testing and runtime verification
    - Maintenance and evolution of CPS

- Education improvements
  - Dedicated CPS curricula (e.g. courses and education material)
  - Prepare future generation of software engineers able to deal with complexity of CPS

- Increased collaboration
  - Academia with open source and private industrial organisations
  - CPS challenges can only be addressed with coordinated effort
  - Multidisciplinary approach required
    - CPS Developers and Researchers
    - Software Engineering Researchers
    - AI and Security Researchers
    - Physics, Human-machine Interaction, etc.

- Focused Research funding
  - Reliability, Security and Safety challenges of collaborative CPS
  - Self-adaptability of CPS to diverse environments, humans, longevity
  - Further challenges for DevOps for CPS
    - AI driven development and devices
Concertation and Consultation on Computing Continuum:
From Cloud to Edge to IoT

*Success Stories*

Thanos Stratikopoulos
The University of Manchester

Brussels, Belgium
The context

• Big Data applications process large amounts of data arriving as streams from IoT devices
• Edge processing holds the key for:
  ➢ increased responsiveness
  ➢ better energy efficiency
  ➢ data privacy

The ELEGANT Vision

• Unification of IoT and Big Data programming environments
  ➢ Automatic and easy deployment of existing code from Big Data platforms to IoT devices
    and backwards; in a self-adaptable way

The ELEGANT Objectives

• Unification of programming environments
• Dynamic Code Motion
• Intelligent resource selection and allocation
• Secure, Reliable, and Dependable code deployment
Lessons learnt and success stories

• Challenges
  ➢ Diversification of the programming models
  ➢ Lack of interoperability between Big Data and IoT deployments
  ➢ Lack of dynamic semantic code adaptation
  ➢ Inability to dynamically orchestrate code
  ➢ Weakened security features

• ELEGANT Solutions
  ➢ Unified API for Cloud, Edge, IoT
  ➢ Elastic Runtime
  ➢ Intelligent Orchestrator
  ➢ Acceleration Service
  ➢ Code Verification Service
  ➢ Networking Cybersecurity Layer
  ➢ DevOps tools
ELEGANT Use Cases

- Secure Smart Riding (KTM)
- Large-Scale Secure Smart Metering (UNIDATA/CNIT)
- Video Surveillance (UBITECH/UNISYSTEMS)
- Medical Wearables (SPARK WORKS)
Recommendations for the Future

• Performance & Energy efficiency
• Scalability
• Programmer-friendly Tools/Libraries
• Critical Use Case Requirements (e.g. Latency, Privacy)
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957286
FOCETA Vision and Scientific Pillars

Introduce a mixed approach for engineering trustworthy learning-enabled autonomous systems based on combining the advantages of data-based (performance) and model-based (guarantees) techniques.

Integrate Learning Enabled Components (LECs) and classical components on the level of models.

Generate a new paradigm for implementing safety-aware LECs by fusing learning from examples and synthesis from the specification.

Transfer verification technology for model-driven design to verification of LECs, and conversely, utilize ML to improve testing of models.

Three scientific pillars:

i) integration of learning-enabled components and model-based components via a contract-based methodology which allows incremental modification of systems, including threat models for cyber-security,

ii) adaptation of verification techniques applied during model-driven design to learning components to enable unbiased decision-making, and finally,

iii) Incremental synthesis techniques unified the enforcement of safety and security-critical properties and performance optimization.
**FOCETA RESULTS**

**Methodology for Continuous Updating**

- Design and Simulation
- Application and Operation
- Operation Analysis and System Evolution for Dependability Improvement
- Monitoring Architecture for Seamless Development and Operation

**Co-simulation Framework**

- Adversarial attacks and defenses
- System under Test Verification
- Training
- Offline V&V

**Design Flow for Trustable Learning Components**

- Machine Learning Life Cycle
- Upgrade
- Certification of Machine Learning

10-11 May, The Claridge, Brussels, Belgium
### Lessons Learned

The role played by **data** in AI-enabled Systems is central. The collection, cleaning, management, and continuous data update add new tasks.

**Uncertainty** is the dominant characteristic in AI-enabled Systems. Increased the urgency of making progress on how to model, analyze, and safeguard against the inherent uncertainty of our systems.

**AI specifications** are specifications of problems, **not the behavior** of systems.

**Verification** challenges are **inevitably exacerbated** in AI-enabled systems, given their inherent uncertainty.

The **continuous update** is a big challenge in AI-enabled systems.

We know the challenges of designing embedded systems that rely on integrating many disparate SW/HW components. **AI components** developed independently are another set of subcomponents whose behavior must be reliably predicted.

### Recommendations

Need for a **multidisciplinary Network of Excellence** with AI, software engineering, critical embedded systems, statistics, formal methods with European industrial actors, CPS and control theory, and SHS (law, ethics).

Support EU research that uses more rigorous, mathematical methods in dealing with AI.

The recommendation of FOCETA naturally is that for LEAS, the verification at design time only is not sufficient and that the combination of design-time verification and runtime assurance is a must.
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956123.
Programming trustworthy Infrastructure As Code in a sEcuRE framework [PIACERE]

Matija Cankar (XLAB)
Juncal Alonso (Tecnalia)

Brussels, 10th of May
Open PIACERE Framework

▪ **Vision:** DevSecOps framework for the **development**, **deployment** and **operation** of trustworthiness infrastructure-as-code.

▪ **Goal:** Framework with **tools** integrated in the IDE.

▪ **Status:** PoC version already available!
Lessons learned and Success stories

▪ Refocus a task according to a technology or market change can be very rewarding (Component security checker).

▪ Design of a language (standard) and implementation have different speeds, and this needs to be considered.

▪ Great interest from the communities:
  • OS: Eclipse, Linux Europe, Gaia-X, TOSCA
  • At least one product from consortium partner exploits PIACERE idea.

▪ PIACERE addresses a problem currently being faced by SW development companies due to the paradigm shift (from owned resources to outsourcing of the infrastructure management).
Recommendation for the Future

▪ Edge and IoT scope to be incorporated in the IaC paradigm (strategic and opportunistic topic for Europe !!)

▪ Trustworthiness is newer-ending task and requires constant attention.
Thank you!

www.piacere-project.org
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 957212
Automated Protection and Prevention to Meet Security Requirements in DevOps Environments

Main challenges

- Security vulnerability are omnipresent
- Number of security scenarios explodes
- Vulnerabilities cause losses for end-users
- Security mechanisms has to be built in and reinforced
- Security is difficult to retrofit in design
- Security has to support CI/CD
- Monitoring and traceability is a key property
Lesson learned

- Close collaboration with case studies
  - Planning all the phases and activities in advance - Frequent replanning
- Still room to enhance automation
- Domain specific / Generalization is not easy
- Prototype -> Product way is difficult and effort consuming
  - Low TRL for many results

Success story

- NLP datasets and models for Requirements classification and security guidelines mapping.
- ML-based anomaly detection and root cause analysis.
- Metamorphic testing generation as intelligent test generation with automated feedback.
- Vulnerability detection at early stages with scanners.

(More than 30 publications and 15 Key exploitable results)
Recommendations for the future

- Human in the loop in cyber-security
- DevOps at Edge
- Architecture-Driven Modernization (Industrial Automation)
- Maintainability of Quality Automation (Testing of tests)
- Automated resilience - Dev and Ops phase (automating bug fixes)
- Trustable recommendations

- Industrial automation, eHealth, maritime, telecom, energy
- Industry domains -
- Sovereign tech
- Societal resilience
- Societal challenges -
- Awareness and education, cyber range

- Technological areas
- Industry domains
- Societal challenges
Thank You

Contact:
X-By-Construction Design Framework for Engineering Autonomous and Distributed Real-time Embedded Software Systems

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957210.
Challenges in the design of autonomous embedded systems

- **Performance requirements** met only by centralized multicore processors
- **Continuous interaction** between internal subsystems and with remote entities
- **Non-functional requirements** towards functional safety and cybersecurity
- **Artificial Intelligence (AI)** applied to solve complex tasks in an efficient manner

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

**Project Overview**

**Project Duration:**
01/2021 – 12/2023

**Budget:**
€ 4.96 million

**Project Coordinator:**
Prof. Jürgen Becker (KIT)

**Scientific Coordinator:**
Prof. Nikolaos Voros (UoP)
Success Stories

- **X-by-Construction (XbC) Paradigm** realization shown by features such as:
  - *Pattern library of safety/security mechanisms*
  - Safe integration of *AI components* (ONNX, ...)
  - Timing-aware behaviour specification approach
  - *Hypervisor-based* security monitoring

- First **experimental Evaluation** phase successfully completed:
  - Application to SW development of *safety-critical use cases* (*DLR* + *BMW*)
  - **Automatic deployment** to bare-metal *hypervisors* on *multicore platforms*

- **Lessons learned** from Development Efforts:
  - Proving the fulfilment of *non-functional guarantees* for dynamic/adaptive systems is difficult
  - Many applications have *static invariants* that make them compatible with the *XbC paradigm*
  - Asking designers to specify these invariants is often successful for today’s systems
Recommendations

- **XANDAR has shown:** XbC is applicable to systems with *Static Invariants*

- **Open question:** Invariants to be defined in future *Cloud/Edge Networks*?
  - Dynamic workloads
    - multi-tenant use of *distributed AI accelerators*, ...
  - Continuously refined system configurations
    - *over-the-air* updates, ...
  - Evolving edge nodes
    - *self-learning AI* algorithms, ...

As **Enabler** for reliable *Cloud/Edge/IoT Networks*, for:
  - **Dynamics:** links to projects such as *Lingua Franca* *(Berkeley, USA)*
  - **Adaptivity:** by integrating XbC runners into the *entire DevOps cycle*

- **Research Challenges:** How to facilitate *dynamic XbC and open HW/SW Architectures*?
  - Can *time predictability* be achieved for *multi-tenant accelerators*?
  - Is it possible to guarantee *data confidentiality* in *adaptive cloud/edge nodes*?