COSMOS

DevOps for Complex Cyber-physical Systems

The COSMOS Project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 957254.

COSMOS:
DevOps for Complex Cyber-physical Systems

Sebastiano Panichella

Zurich University of Applied Sciences (ZHAW)

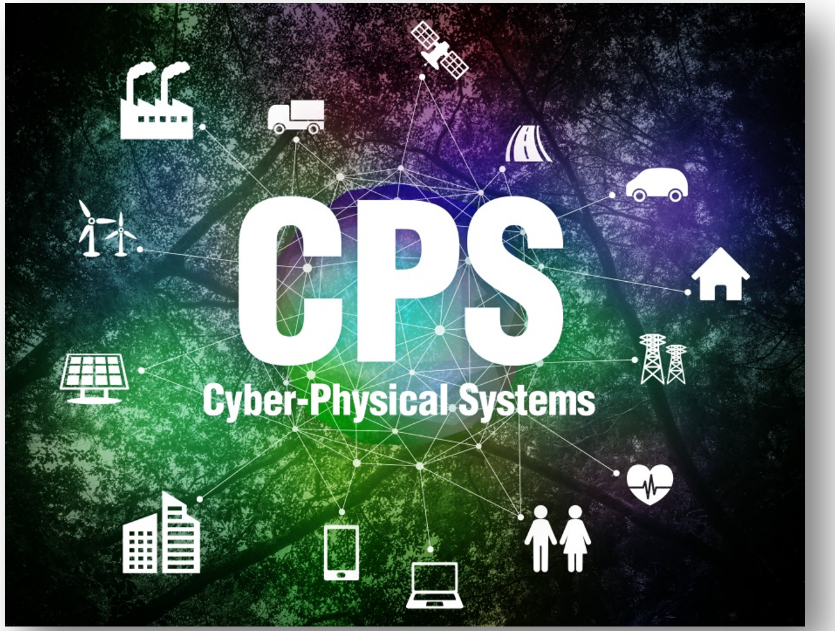
Cognitive Cloud Infrastructure-webinar 2024

www.COSMOS-DevOps.org **@COSMOS_DEVOPS**

Context

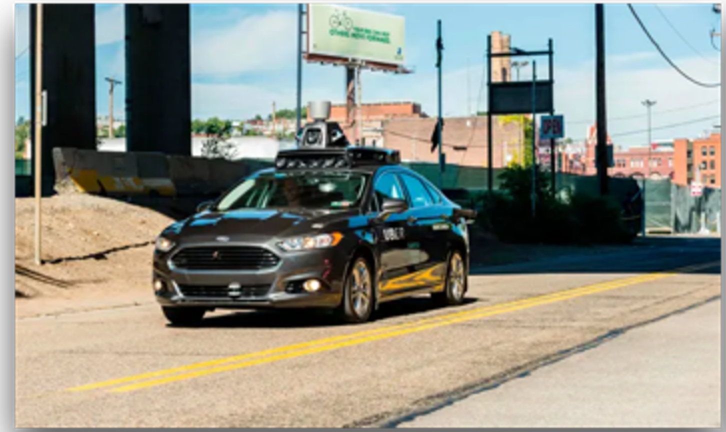
- Emerging Cyber-physical Systems **(CPS)** play a crucial role in the quality of life of European citizens and the future of the European "smart everywhere" economy

- **CPS relevant sectors**
  - Healthcare
  - Avionics
  - Automotive
  - Utilities
  - Railway
  - Manufacturing
  - Smart Cities
  - Many others…



Challenges

- **C1:** Observability, testability, and predictability of behaviour of CPS is highly limited and, unfortunately, their usage in the real world can lead to fatal crashes sometimes tragically involving also humans



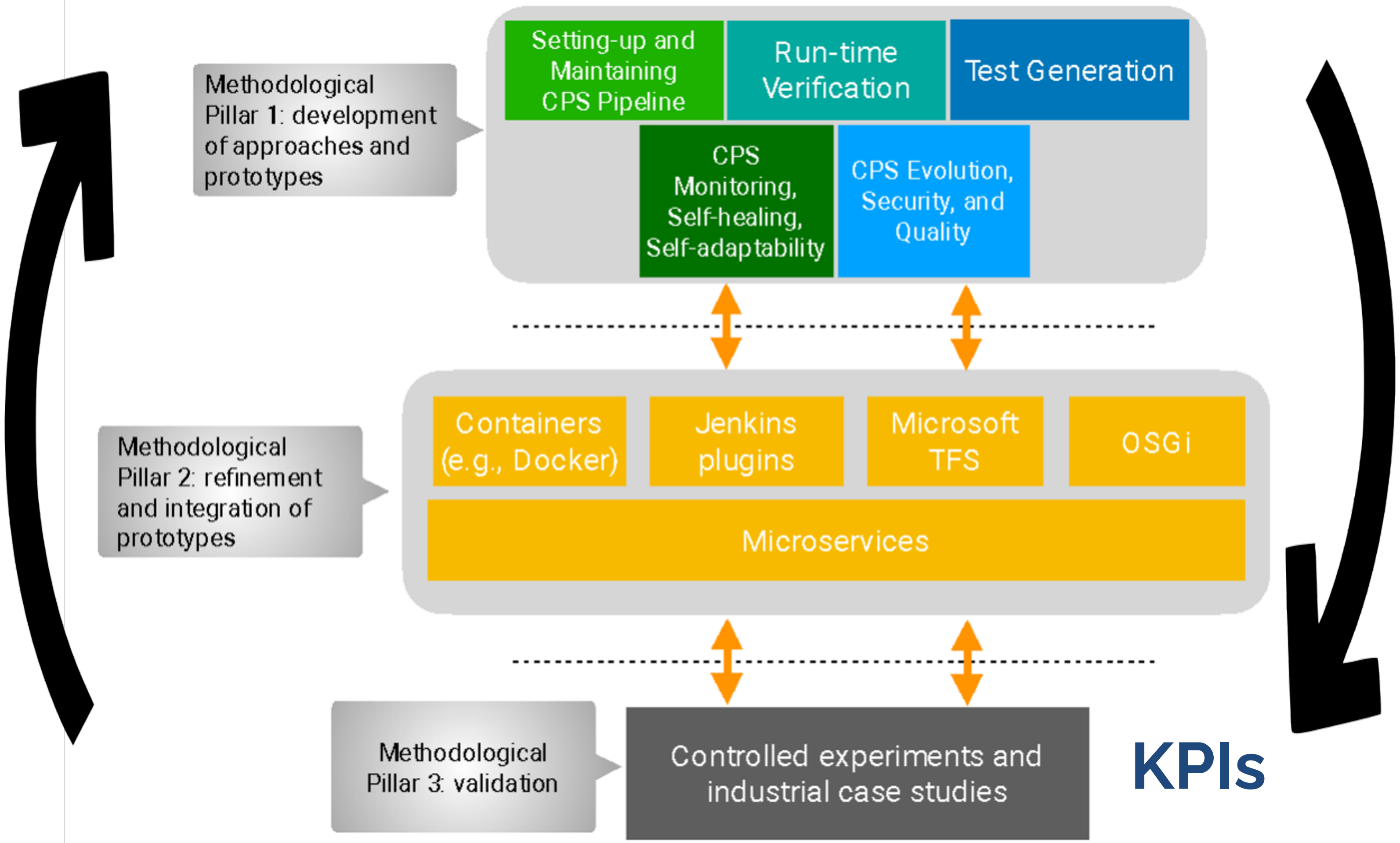
"Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian"

"A simple software update was the direct cause of the fatal crashes of the Boeing 737"

"Swiss Post drone crashes in Zurich - again"

- **C2:** Contemporary DevOps practices and tools are potentially the right solution to this problem, but are currently not developed to be applied in CPS domains

COSMOS Vision: Three Methodological Pillars



Scientific and Technological Foundations of COSMOS

DevOps Technological Foundations of COSMOS

Empirical Validation of COSMOS Innovations

Industrial CPS Evaluations

COSMOS

aicas realtime

Automotive

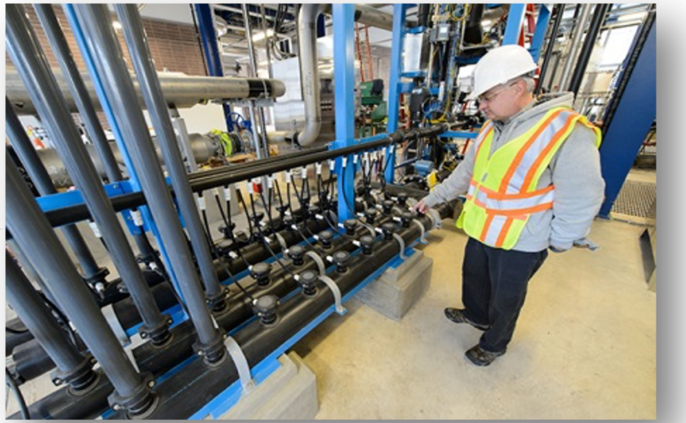
SIEMENS Healthineers

Medical

Avionics

gmv INNOVATING SOLUTIONS

UNPARALLEL



Utilities

Railways

Q-media

Evaluations conducted at both mid-project and during final project months

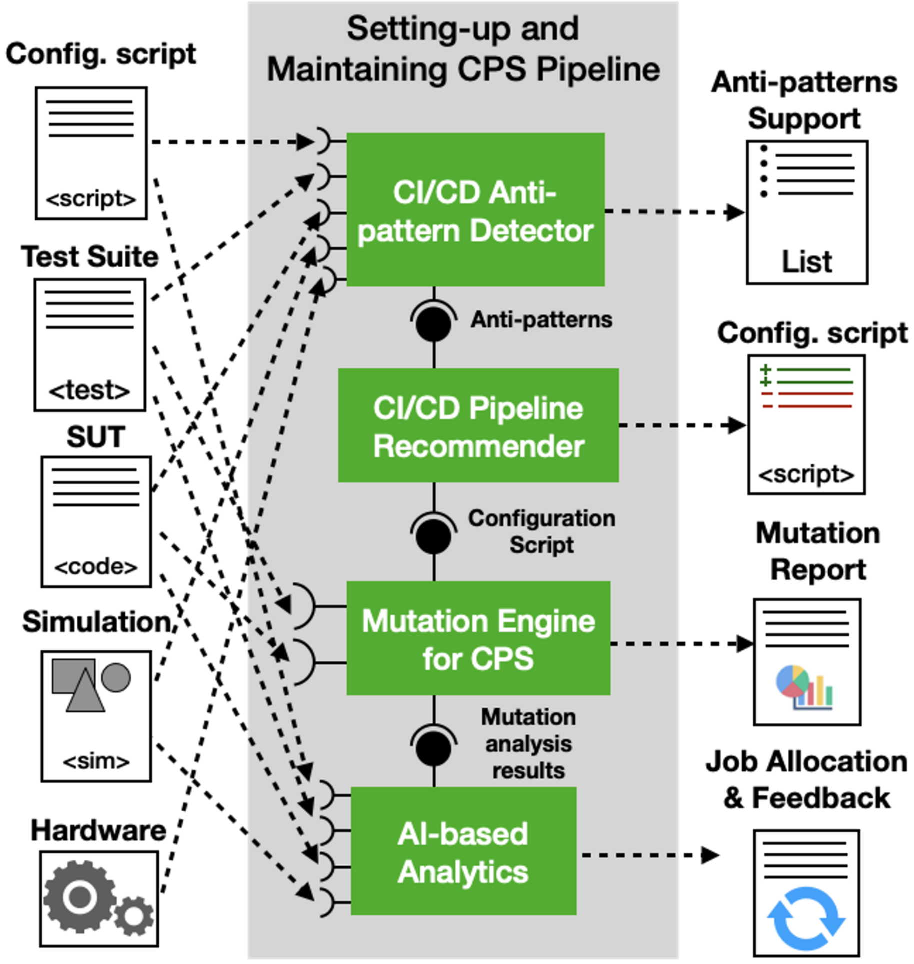
Innovation Area 1: DevOps Pipelines for CPS

COSMOS

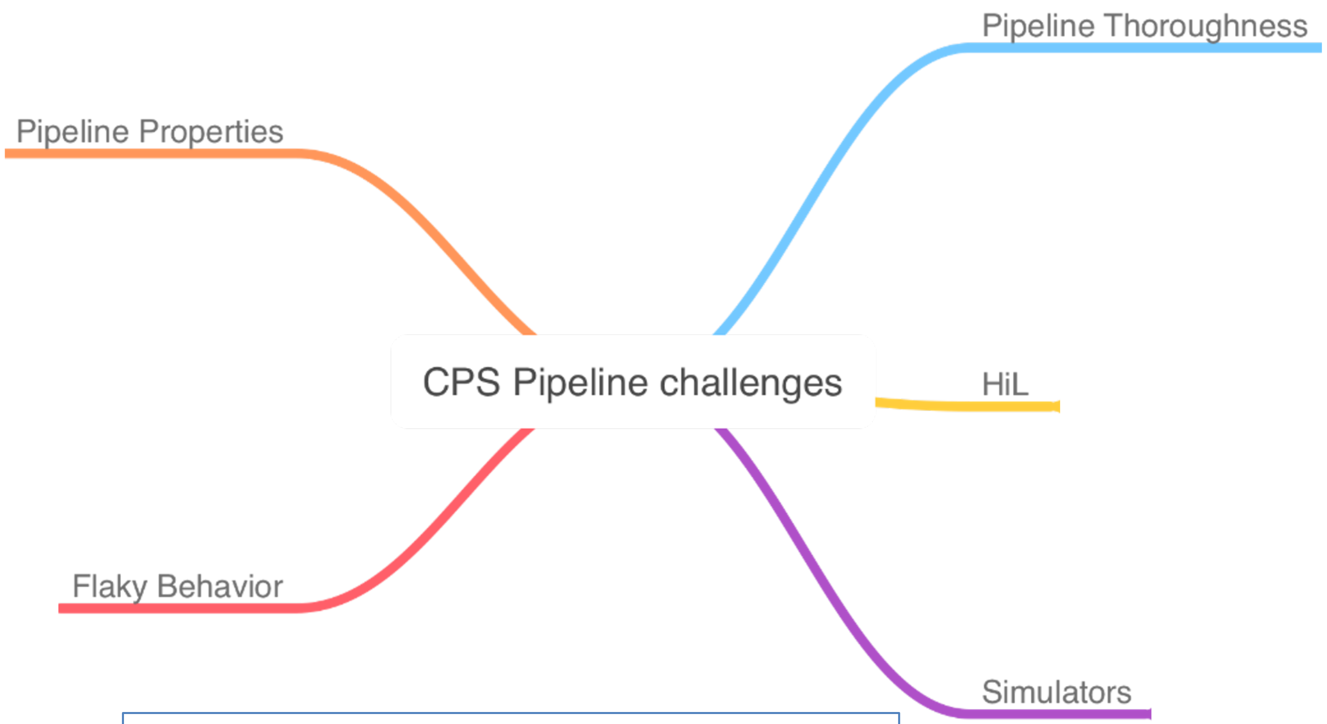
WP3: Methodology for Setting-Up and Maintaining COSMOS DevOps Pipelines

- **CI/CD Antipatterns Identification** for CPS
- Definition of a **DevOps-based** Methodology to **Support the Development of Self-Adaptive** CPS
- COSMOS **Pipeline Optimization**

COMPONENTS

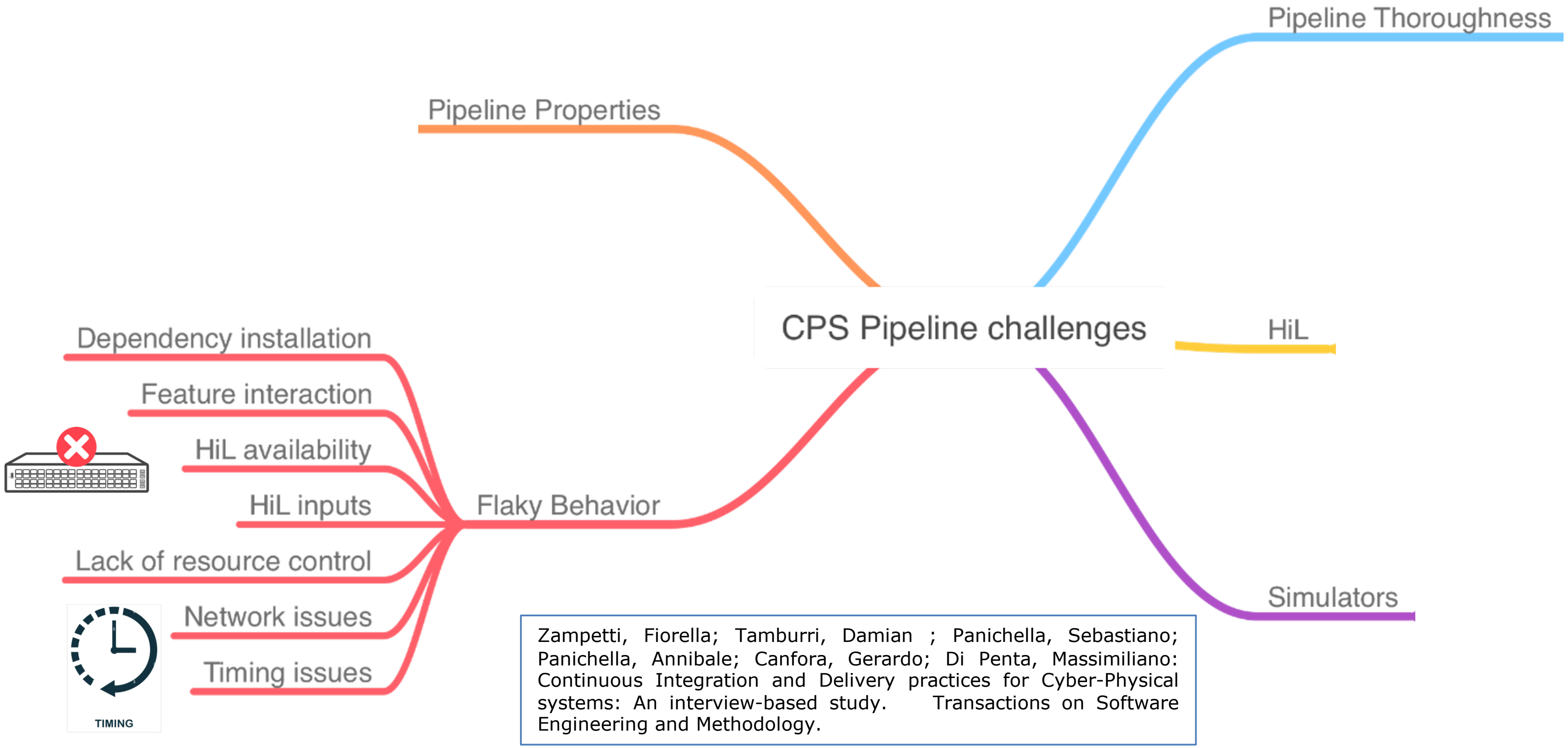


Finding overview - CPS DevOps challenges



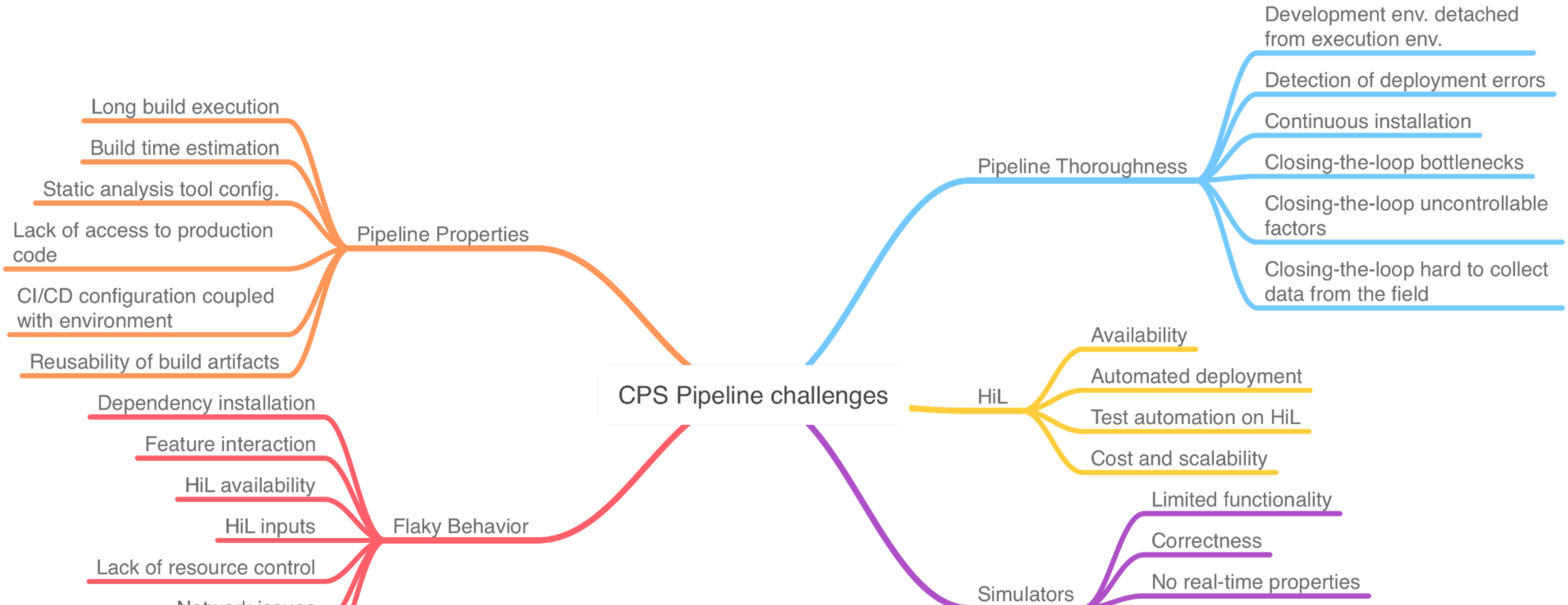
Zampetti, Fiorella; Tamburri, Damian ; Panichella, Sebastiano; Panichella, Annibale; Canfora, Gerardo; Di Penta, Massimiliano: Continuous Integration and Delivery practices for Cyber-Physical systems: An interview-based study. Transactions on Software Engineering and Methodology.

Finding overview - CPS DevOps challenges



Zampetti, Fiorella; Tamburri, Damian ; Panichella, Sebastiano; Panichella, Annibale; Canfora, Gerardo; Di Penta, Massimiliano: Continuous Integration and Delivery practices for Cyber-Physical systems: An interview-based study. Transactions on Software Engineering and Methodology.

Finding overview - CPS DevOps challenges



Zampetti, Fiorella; Tamburri, Damian ; Panichella, Sebastiano; Panichella, Annibale; Canfora, Gerardo; Di Penta, Massimiliano: Continuous Integration and Delivery practices for Cyber-Physical systems: An interview-based study. Transactions on Software Engineering and Methodology.

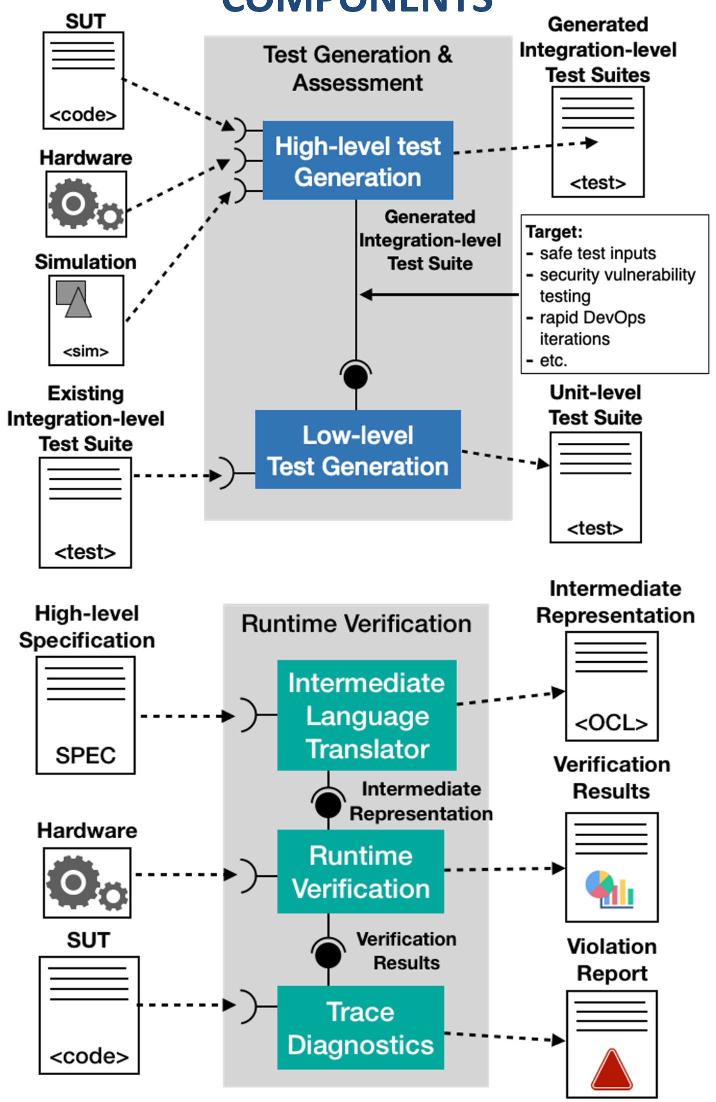
Innovation Area 2: V&V and Security Assessment of DevOps pipelines

COSMOS

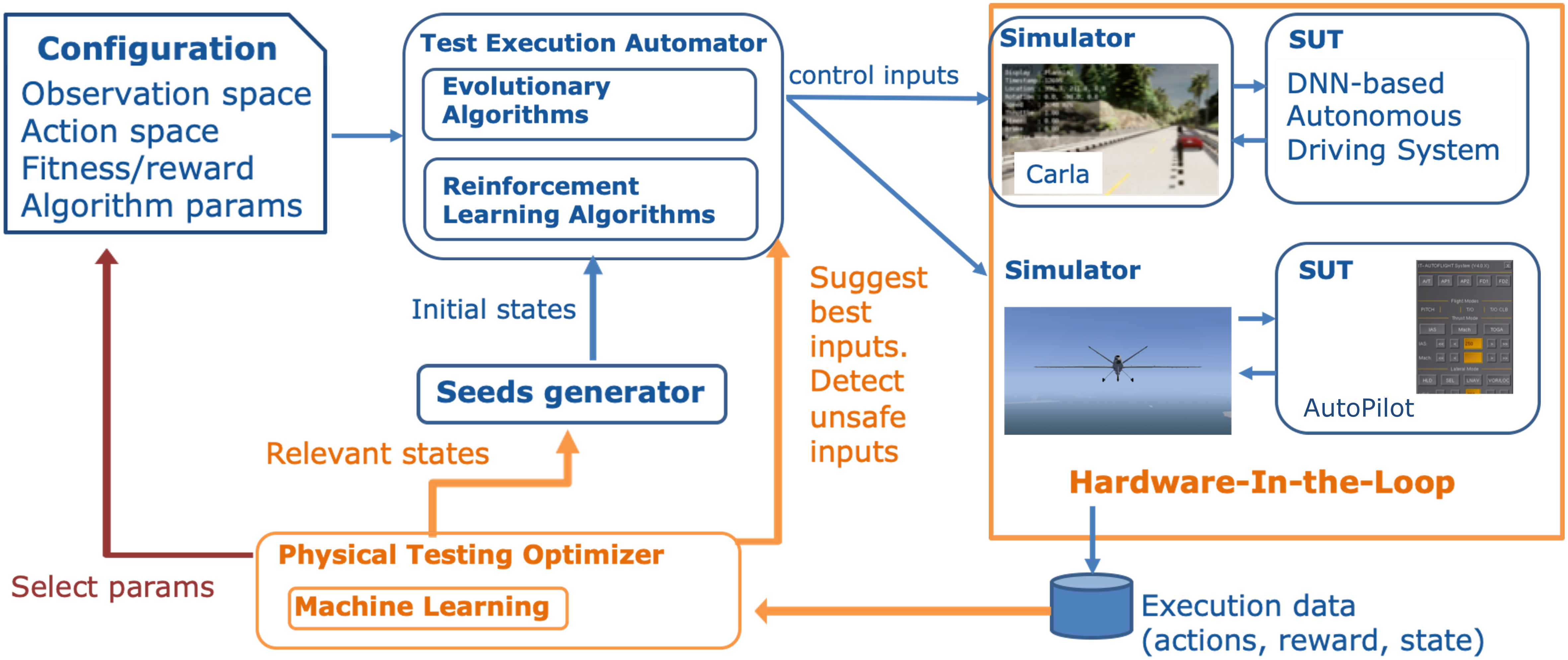
WP4: V&V and security assessment of COSMOS DevOps pipelines

- Development of Automated Techniques for Software **Testing** for CPS
- Development of **Run-time Verification** Techniques for **Checking** and Diagnosing **CPS** Executions
- Development of Solutions for Detecting **Security Vulnerabilities** in CPS

COMPONENTS



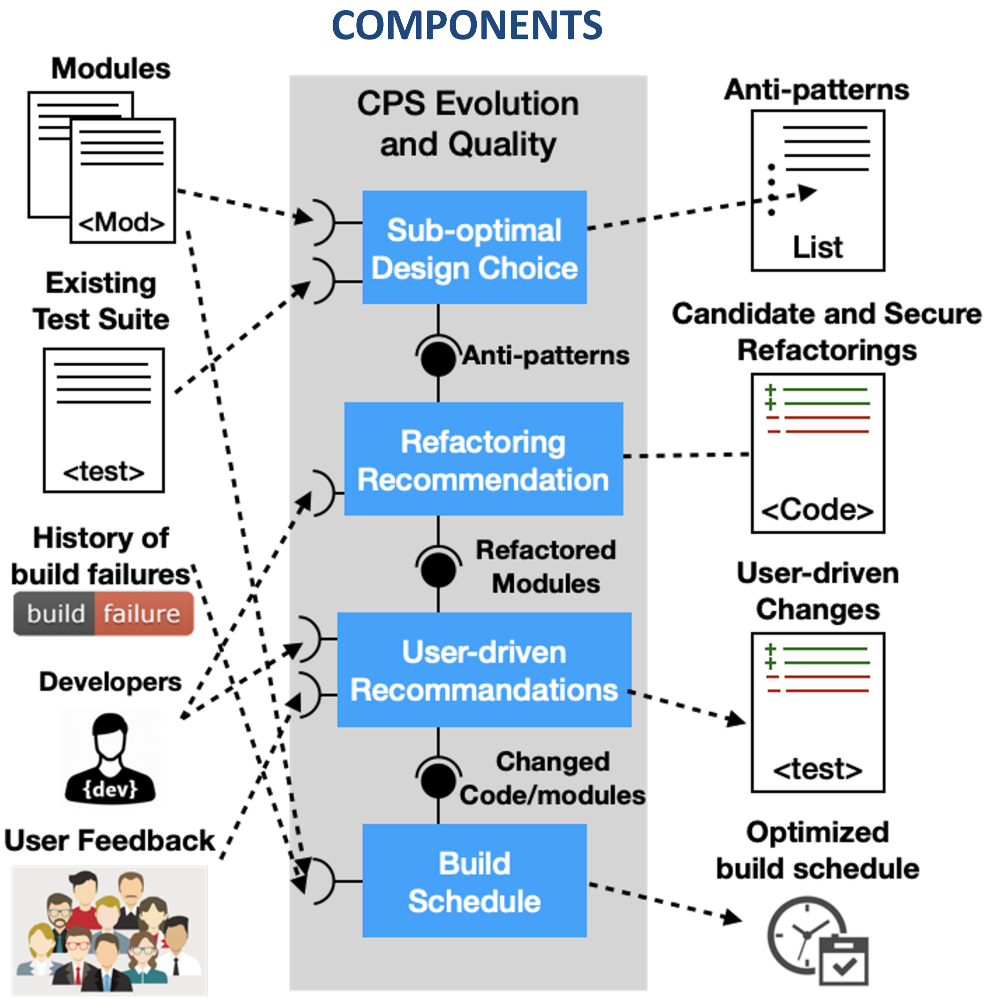
V&V of DevOps pipelines for CPS



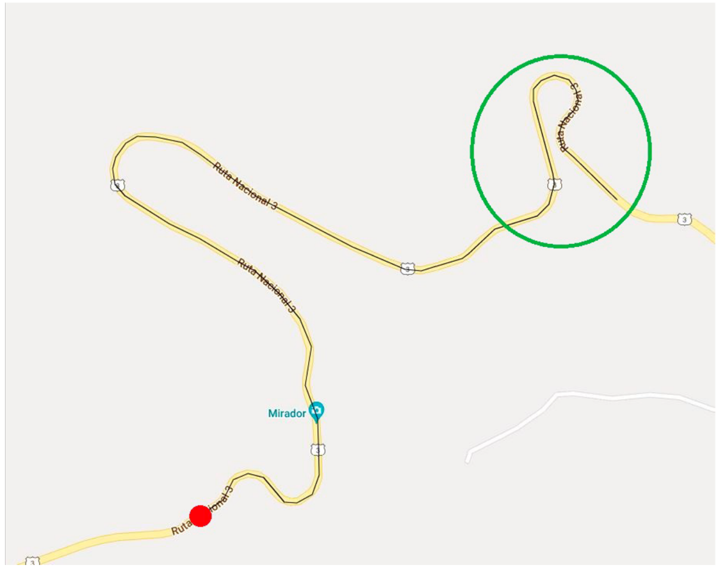
Innovation Area 3: Tools for High Quality CPS Software Evolution

WP5: Development of Tools to support High Quality CPS Software Evolution

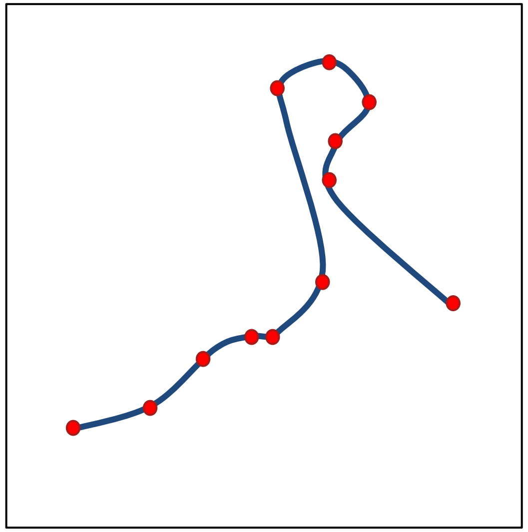
- Design and **Development of Refactoring Framework** for Secure and Reliable CPS
- Development of **Test Case Generation** Tools for **Rapid DevOps Iterations**
- Development of Tools to support **User-oriented Maintenance and Testing**



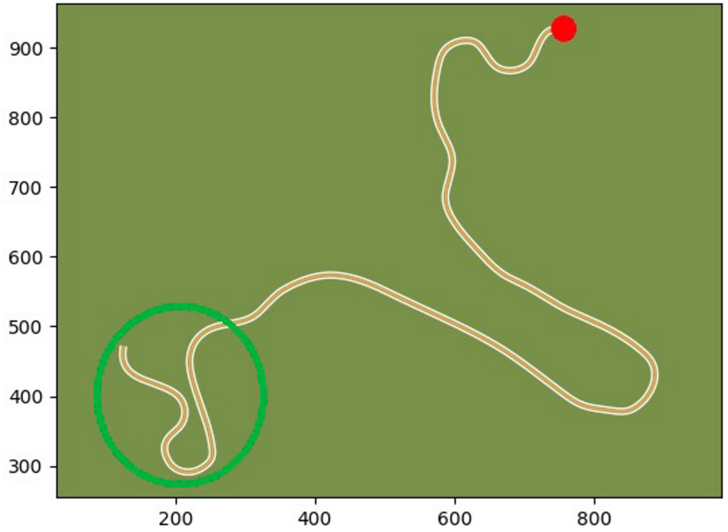
Seeding Strategies based on Real-world Streets



Road from Google Map



Map reconstruction via Spline interpolation



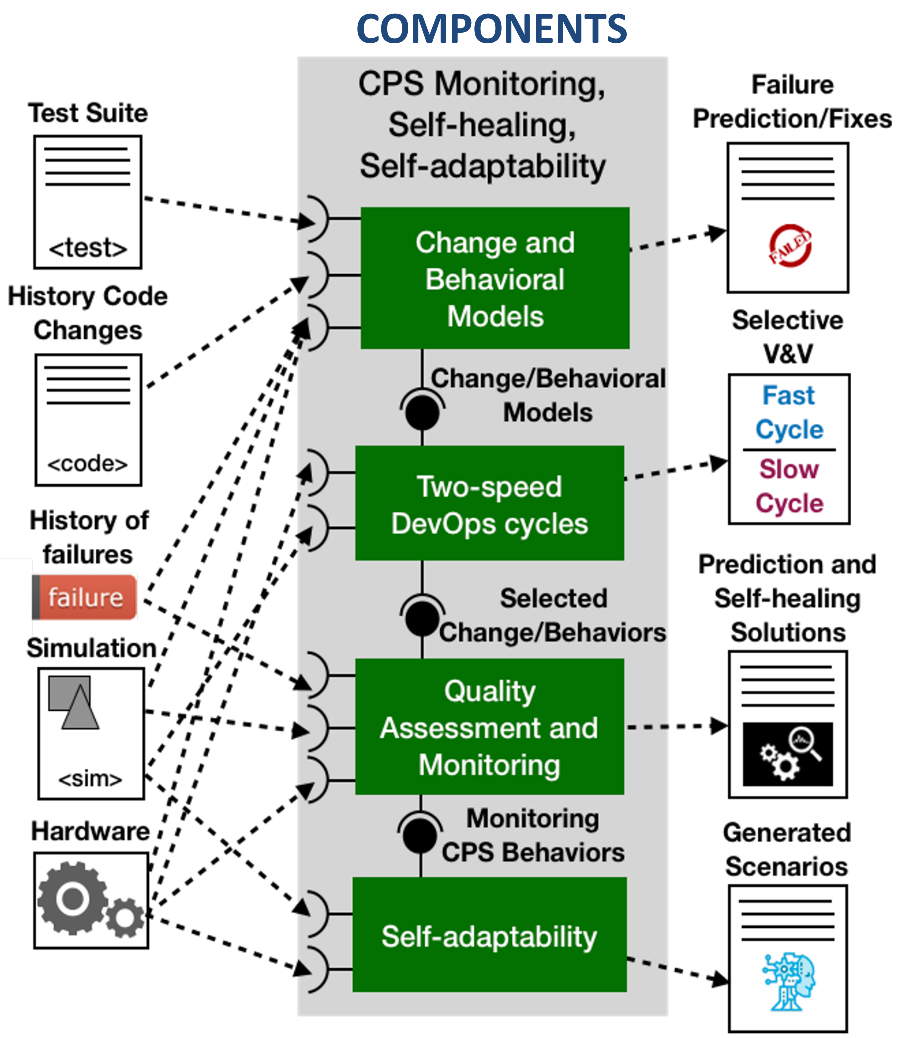
Simulation-based Driving Scenario

Innovation Area 4: Tools for Monitoring, Self-healing and Self-adaptability of CPS

COSMOS

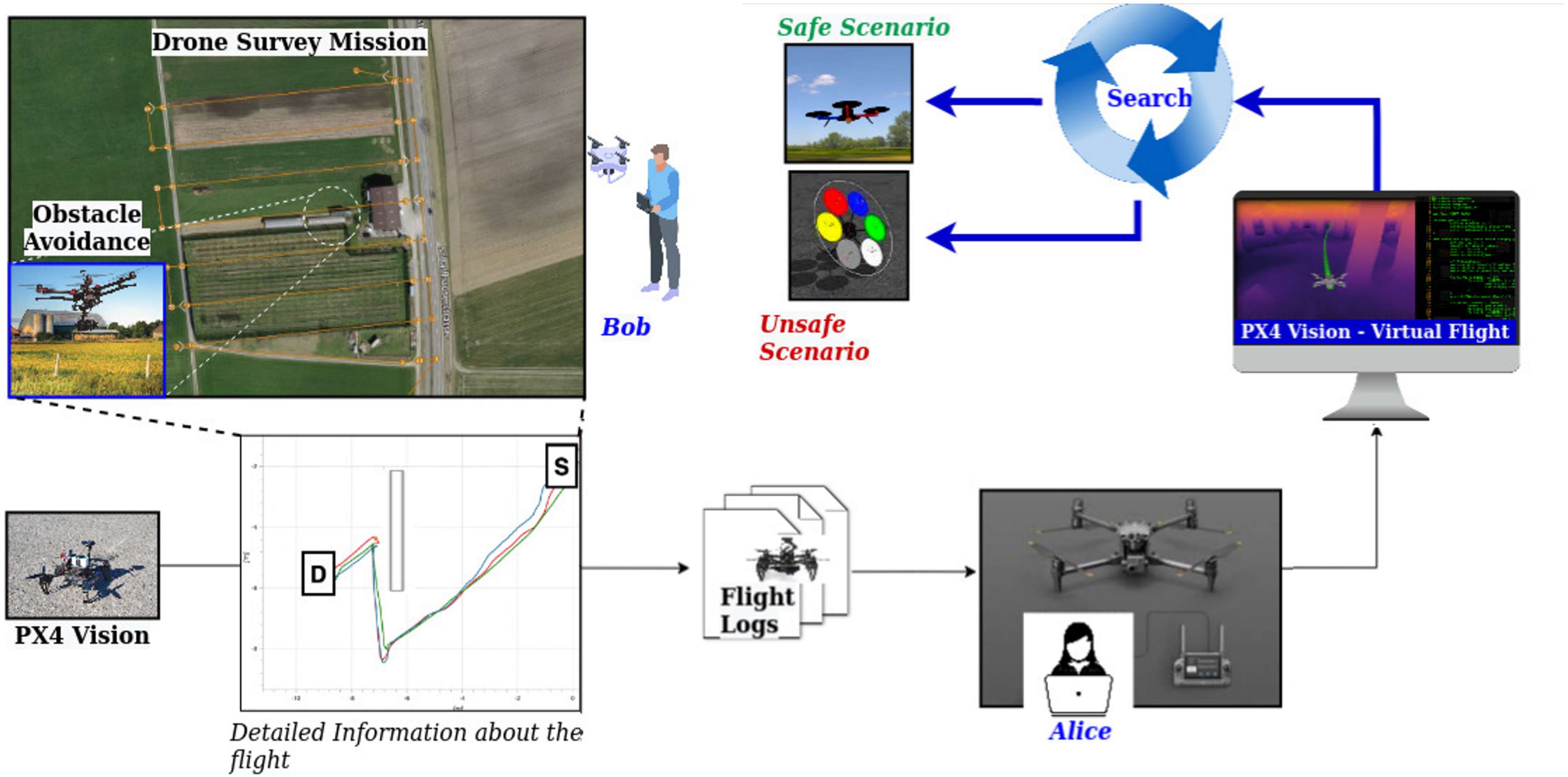
WP6: Development of Tools to support Monitoring, Self-healing, and Self-adaptability of CPS **in the Field**

- Development and **Assessment of CPS Change** & Behavioral Models
- Developing **AI-based** Solutions to Support **Two-speed DevOps Cycles** for **CPS**
- Automated **Quality Assessment and Monitoring** of CPS in the Field
- Development of AI-based Solutions to **Increase CPS Self-adaptability to Diverse Contexts**

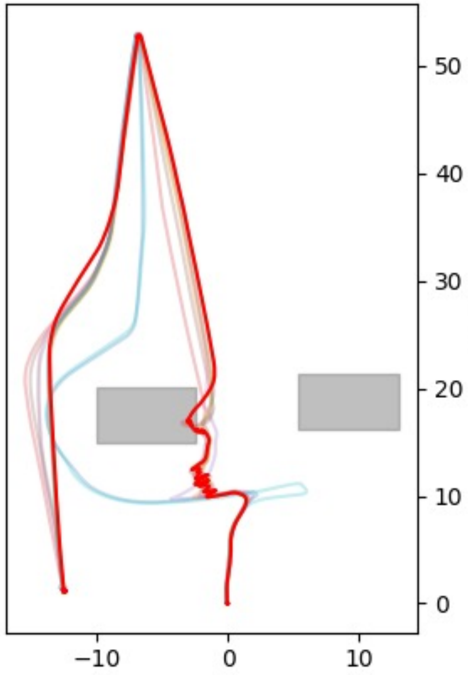


UAV Test Case Generation in the Neighborhood of Real Flights

COSMOS



▪Unsafe / Misbehavior



Sajad Khatiri, Sebastiano Panichella, Paolo Tonella: Simulation-based Test Case Generation for Unmanned Aerial Vehicles in the Neighborhood of Real Flights. International Conference on Software Testing, Verification and Validation. (ICST 2023)

Targeted Impacts

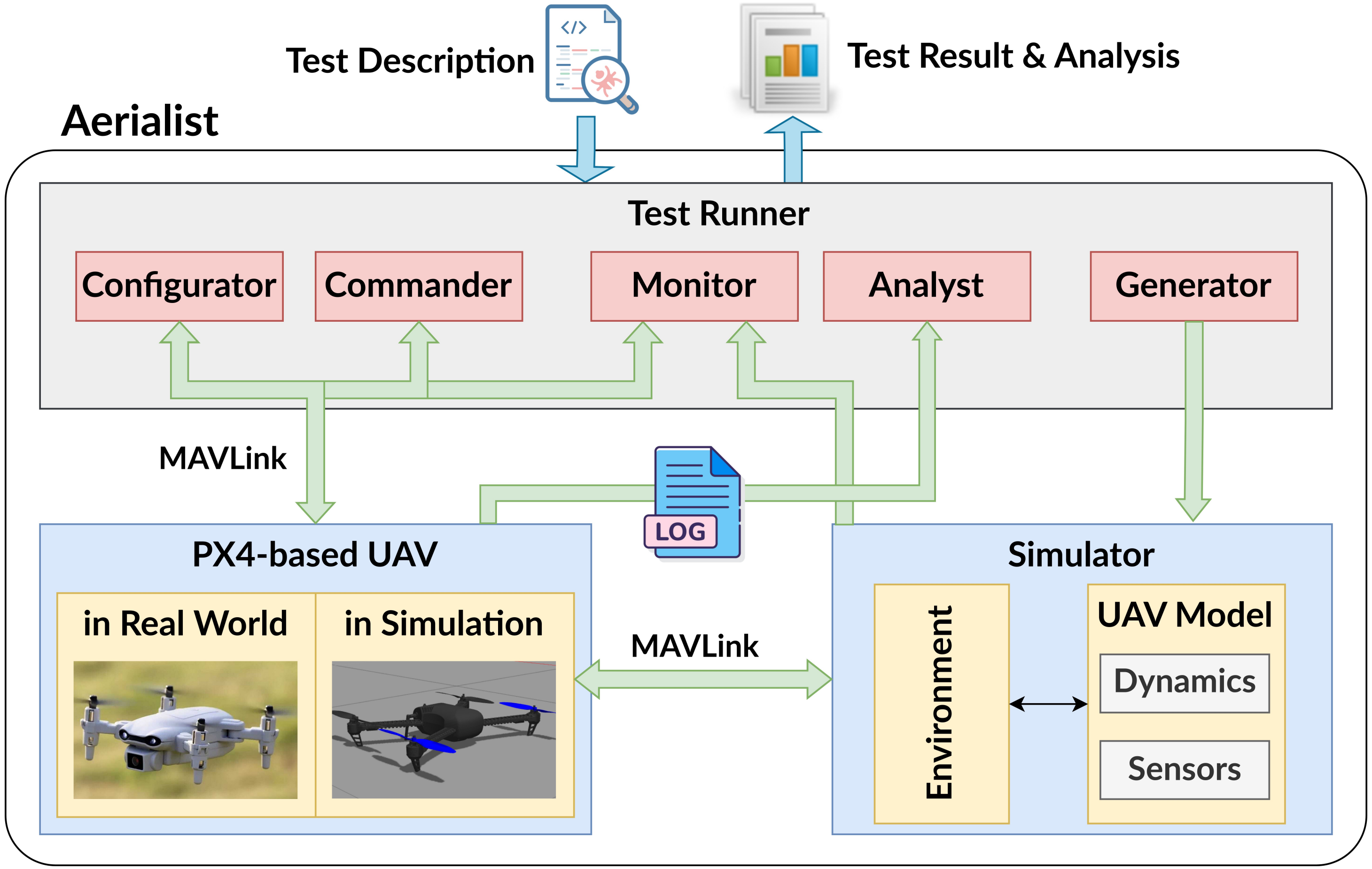
- **Industrial Impacts**
  - **Decreasing** percentage of **changes** that result in **CPS failure**
  - **Reducing CPS test execution** time and computational resource consumption
  - **Replacing manually generated tests** with automated CPS test coverage
  - **Improving test effectiveness** through tests able to discover more bugs
  - **Reducing** number of **security vulnerabilities** in CPS
  - **Reducing** component **integration** and **deployment time**
  - **Reducing** time to implement a change and make updated CPS operational
  - **Reducing downtime** when deploying new CPS hardware or software

Targeted Impacts

- **CPS DevOps Ecosystem**
  - Project technologies available in open source with actions to build a European community and ecosystem exploiting DevOps for CPS
- **Standardisation**
  - Usage of existing industry standards and proposed new standards and extensions to ensure “plug-n-play” of DevOps tools for CPS development
- **Academic impact**
  - Partners produced publications and are contributing to educational content

AERIALIST: Open Source Standard for Testing and Monitoring Unmanned Aerial Vehicles

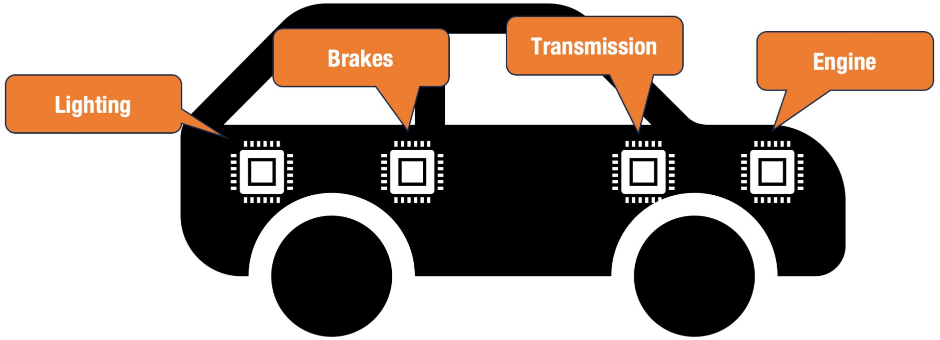
- Test Bench and Test Generation Platform for UAVs
- Using Popular Open-source UAV Autopilot

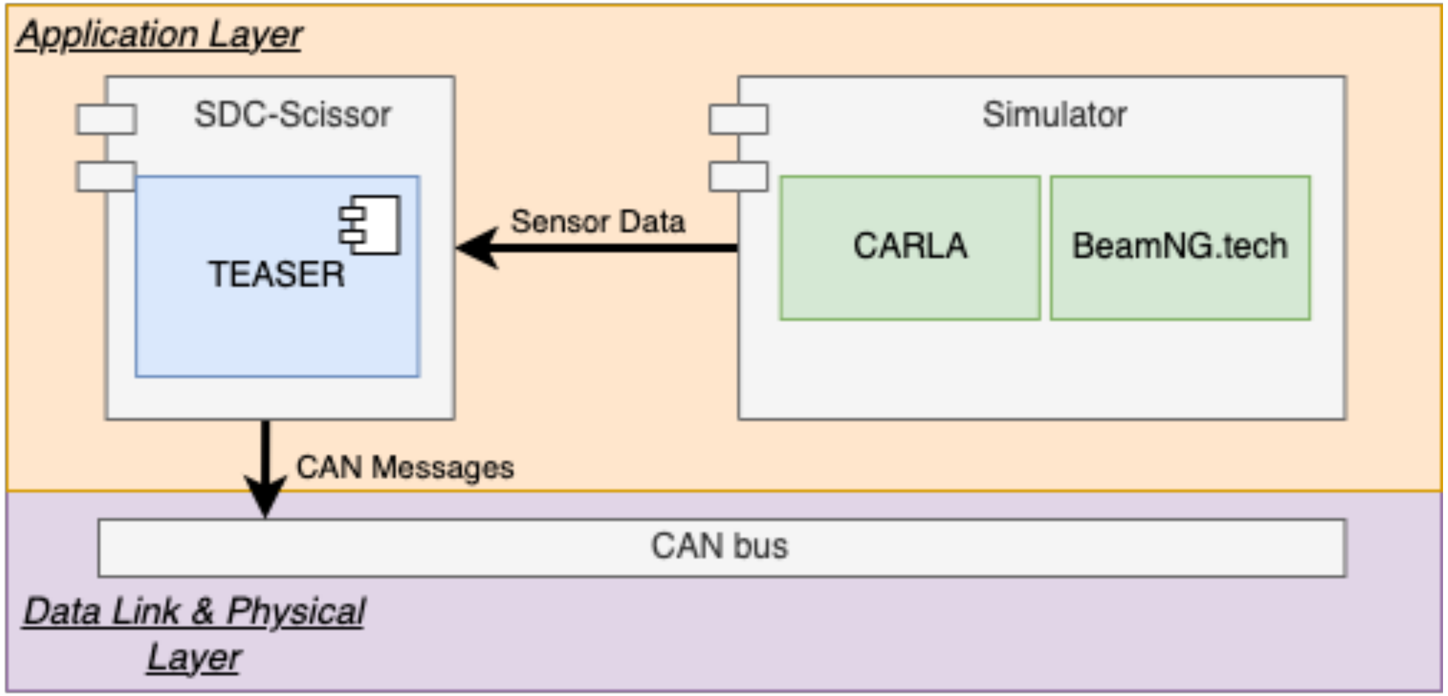


https://github.com/skhatiri/Aerialist https://github.com/skhatiri/UAV-Testing-Competition

TEASER: Open Source Standard for Testing Self-Driving Cars

CAN Bus Use Case





https://github.com/christianbirchler-org/sdc-scissor

TEASER: Open Source Standard for Testing Self-Driving Cars

Search docs

USER DOCUMENTATION

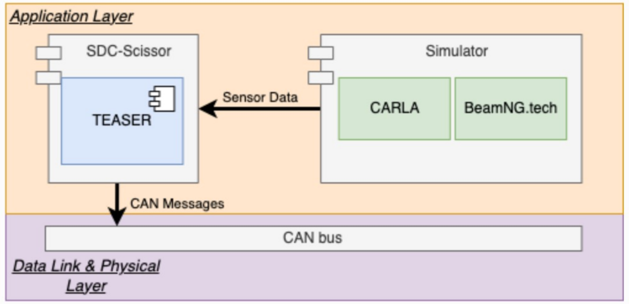
- Quick Start
- Test Generation
- Test Execution
- Feature Extraction
- Test Outcome Prediction
- Test Prioritization
- API

SOFTWARE GUIDEBOOK

- Introduction
- Context
- Functional Overview
- Quality Attributes
- Constraints
- Principles
- Software Architecture
- Infrastructure Architecture
- Deployment
- Operation and Support
- Decision Log
- Contributing
- Contributor Covenant Code of Conduct

Read the Docs v: latest

TEASER CAN bus component



TEASER's main objective is to extend the test runner of SDC-Scissor to enable CAN bus testing. The tool uses two open source python libraries; the python-can 2 and cantools 3 packages. The python-can library allows communication with the CAN bus over specific interfaces (e.g., sockets). Complementary to the first package, the cantools library provides functionality to compose the can messages to send on the CAN bus. Specifically, cantools allows the user to specify a CAN database file, which defines how signals are encoded into CAN messages. The following listing illustrates how the wheel speed, throttle, brake, and steering angle are encoded in a CAN message by specifying it in a CAN database file.

```
...
BO_ 177 sampleFrame2: 4 Vector__XXX
 SG_ wheelspeed : 16|16@1+ (0.2,0) [0|13107] "rpm" Vector__XXX

BO_ 161 sampleFrame1: 7 Vector__XXX
 SG_ throttle : 16|16@1+ (0.0001,0) [0|1] "%" Vector__XXX
 SG_ brake : 0|16@1+ (0.0001,0) [0|1] "%" Vector__XXX
 SG_ steering : 32|17@1- (0.01,0) [-655.36|655.35] "degree" Vector__XXX
 ...
```

TEASER Setup

https://github.com/christianbirchler-org/sdc-scissor

COSMOS
DevOps for Complex Cyber-physical Systems

SWForum.eu

The COSMOS Project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 957254.

www.COSMOS-DevOps.org @COSMOS_DEVOPS