Digital Twins in Monitoring Applications Using the Web of Things

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

PRISM
https://iotprismlab.com

Prof. Marco Di Felice **Dr. Luca Sciullo**

Department of Computer Science and Engineering
IoT PRISM Laboratory
University of Bologna, Italy
{marco.difelice3, luca.sciullo}@unibo.it

IoT Prism Laboratory @ UNIBO

IOT-PRISM LABORATORY ↓ RESEARCH ↓ GRADUATE WITH US ↓ NEWS CONTACT FORM DARK MODE SEARCH

IoT-Prism Laboratory

Welcome to **IoTPrism Laboratory**, a cutting-edge research laboratory located within the Department of Computer Science and Engineering at the University of Bologna.

Founded in 2018 by **Prof. Marco Di Felice** and **Prof. Luciano Bononi**, our lab is comprised of esteemed faculty members and talented young researchers dedicated to pushing the boundaries of **scientific and industrial research and innovation on pervasive IoT systems**.

Our mission is to investigate, design, and develop **innovative IT solutions for next-generation pervasive and mobile systems**, characterized by ubiquitous and autonomous sensing and connectivity, decentralized data management, adaptive and extreme edge processing and intelligence, and seamless cloud integration in the continuum. With a focus on IoT interoperability, data trustworthiness, and edge computing, we have addressed cutting-edge research challenges that are crucial to realizing the full potential of the IoT paradigm.

Over the last five years, we have collaborated on several **national and European research projects** implementing the IoT paradigm for **smart agriculture, industrial automation, condition and structural monitoring applications**. We have also published

SEARCH

search

RECENT POSTS

Designing a Hybrid Push-Pull Architecture for Mobile Crowdsensing using the Web of Things

A Decentralized Oracle Architecture for a Blockchain-Based IoT Global Market

The 1st International Workshop on the Internet of Time-Critical Things (IoTime 2022)

Mobile crowdsensing simulation

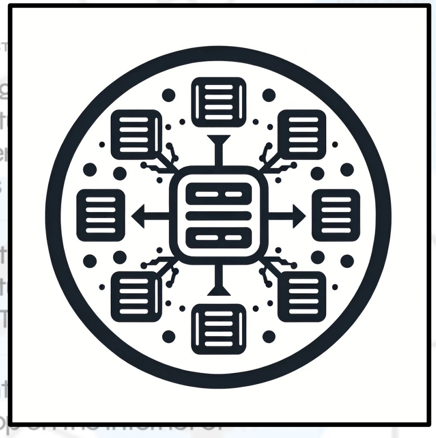
https://iotprismlab.com

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

IoT Prism Laboratory @ UNIBO

INTEROPERABLE SENSING

EDGE- CLOUD CONTINUUM



EDGE AND FEDERATED AI

https://iotprismlab.com

IoT Prism Laboratory @ UNIBO

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

WoT Research activities @ IOT PRISM Lab

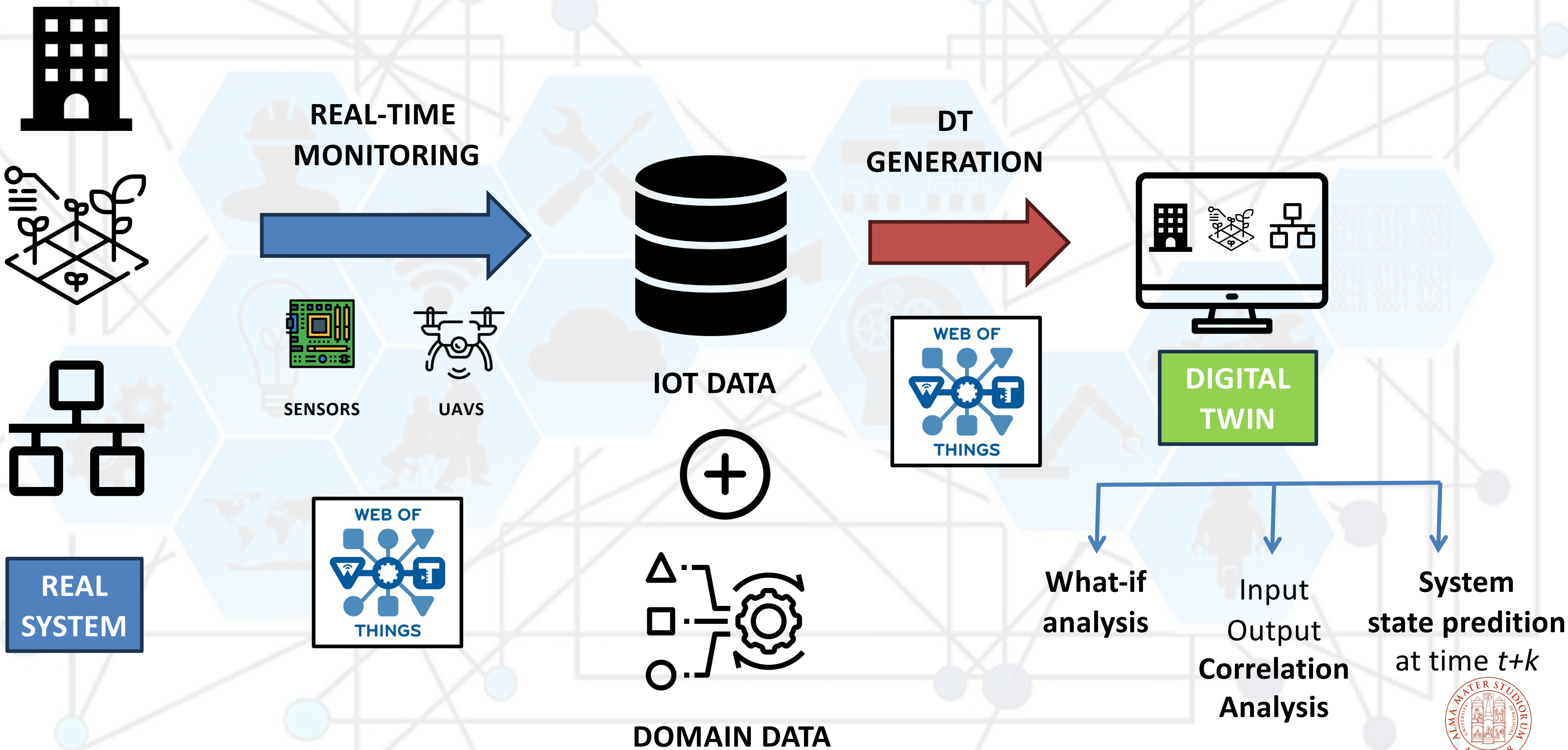
- Comparison/integration with other interoperability standards and platforms (Arrowhead, FIWARE)
- Bringing WoT to constrained devices (e.g., micro-controllers)
- Thing Discovery (**ZION** tool)
- **Tools for the WoT ecosystem**

PRISM

WEB OF THINGS

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Scenario



REAL SYSTEM
REAL-TIME MONITORING
SENSORS
UAVS
WEB OF THINGS
IOT DATA
DOMAIN DATA
DT GENERATION
WEB OF THINGS
DIGITAL TWIN
What-if analysis
Input Output Correlation Analysis
System state predition at time *t+k*

Objectives

1. Design and development of large-scale/pervasive IoT monitoring systems of physical assets

CONTRIBUTION: Leverage W3C WoT for interoperable data collection

WEB OF THINGS

SMART AGRICULTURE

HEALTHCARE

SMART BUILDINGS

2. Support the generation of Digital Twin (**DT**) of the physical asset

CONTRIBUTION: Leverage W3C WoT towards semi-automatic DT generation

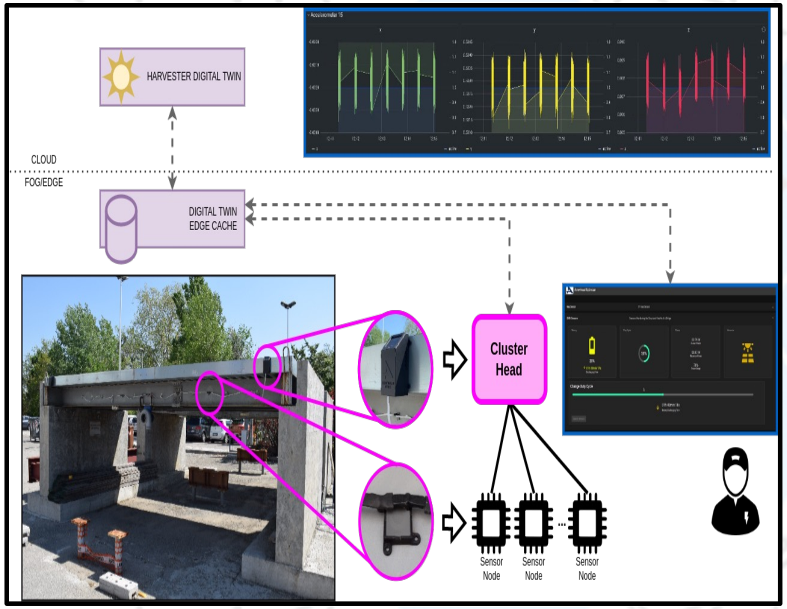
WEB OF THINGS

Monitoring Civil Infrastructures

Structural Health Monitoring (SHM): assess the *structural solidity* of civil buildings and infrastructures through sensing and data analytics techniques.

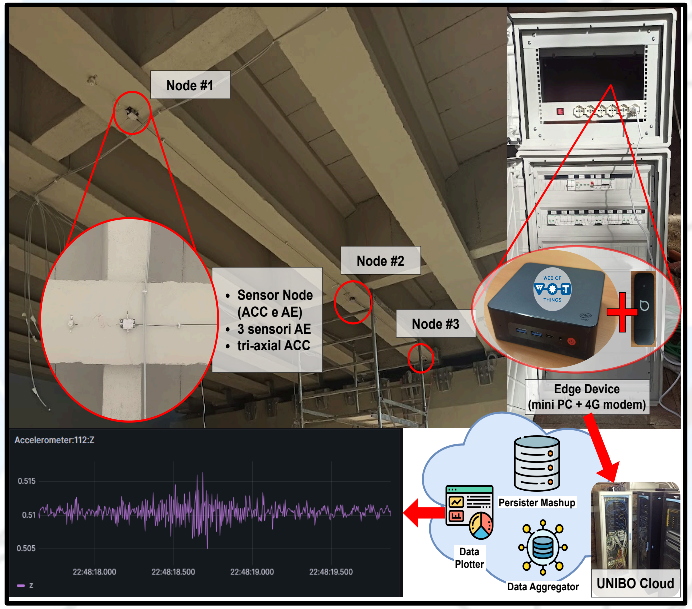
Monitoring Civil Infrastructures

Research projects @ IoT Prism UNIBO



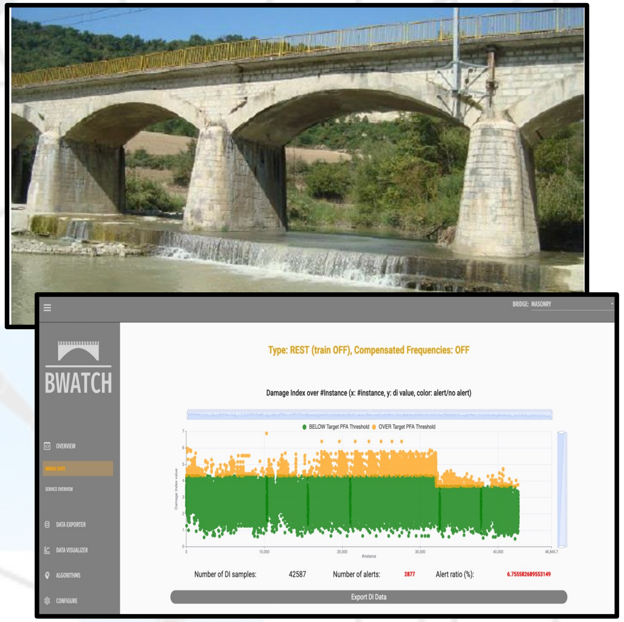
ARROWHEAD TOOLS

Founded by EU KDT program
Target: Civil infrastructures
Goal: interoperability



DS2 & MAC4PRO

Founded by INAIL BRIC program
Target: Civil & industrial infrastructures
Goal: sensing & edge processing



RFI Research

Founded by RFI company
Target: Bridges
Goal: real-time alerting

Monitoring Civil Infrastructures

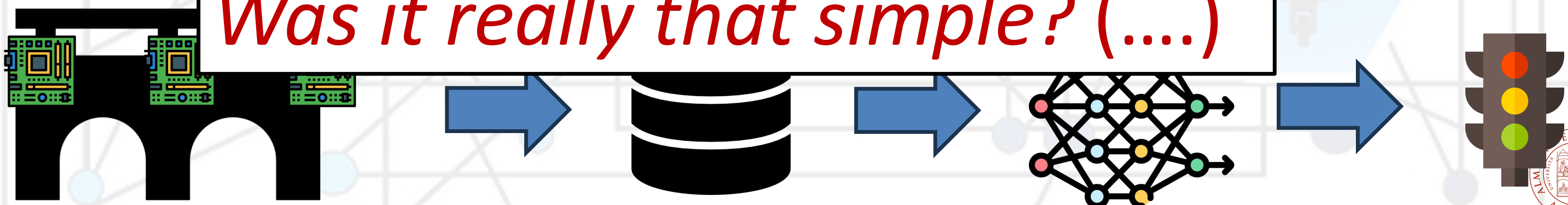
Building a **digital structure** …

STRUCTURE

DIGITAL STRUCTURE

- Real-time monitoring
- Real-time anomaly detection
- Real-time damage classification
- 3D & immersive visualization
- Damage evolution & prediction

Was it really that simple? (….)

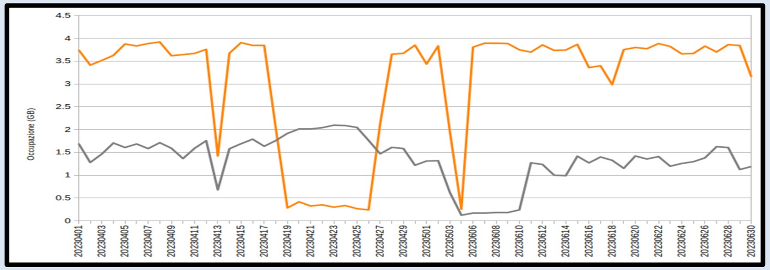


IOT DATA

AI ALGORITHM

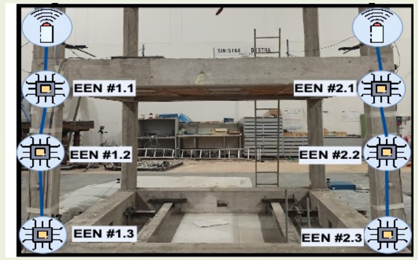
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

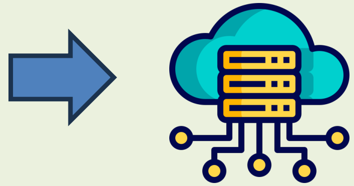
Monitoring Civil Infrastructures



Challenge 1

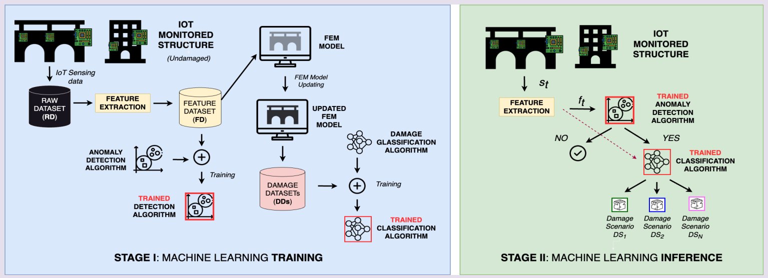
(Big) Data Management





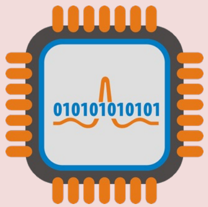
Challenge 2

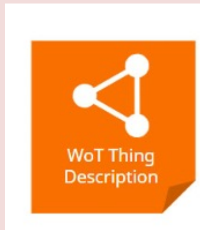
Edge-cloud integration



Challenge 3

AI Training



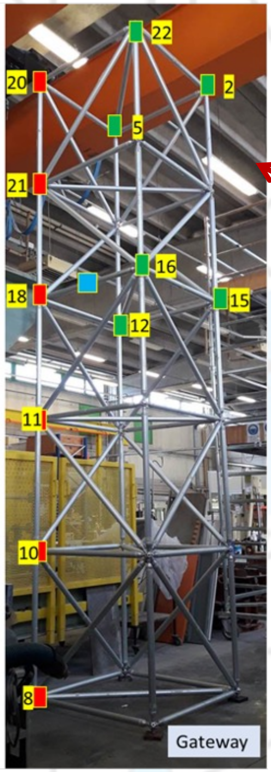


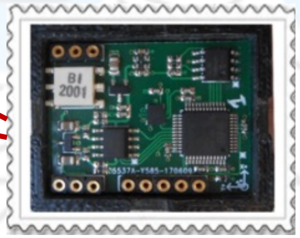
Challenge 4

Interoperability

Monitoring Civil Infrastructures

SHM sensors may be *heterogeneous* and use different data formats and protocols.

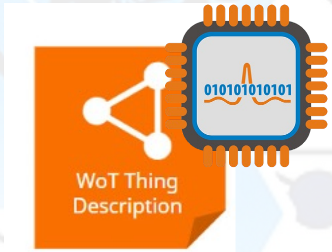




TRI-AXIAL
ACCELEROMETER

WEB OF THINGS

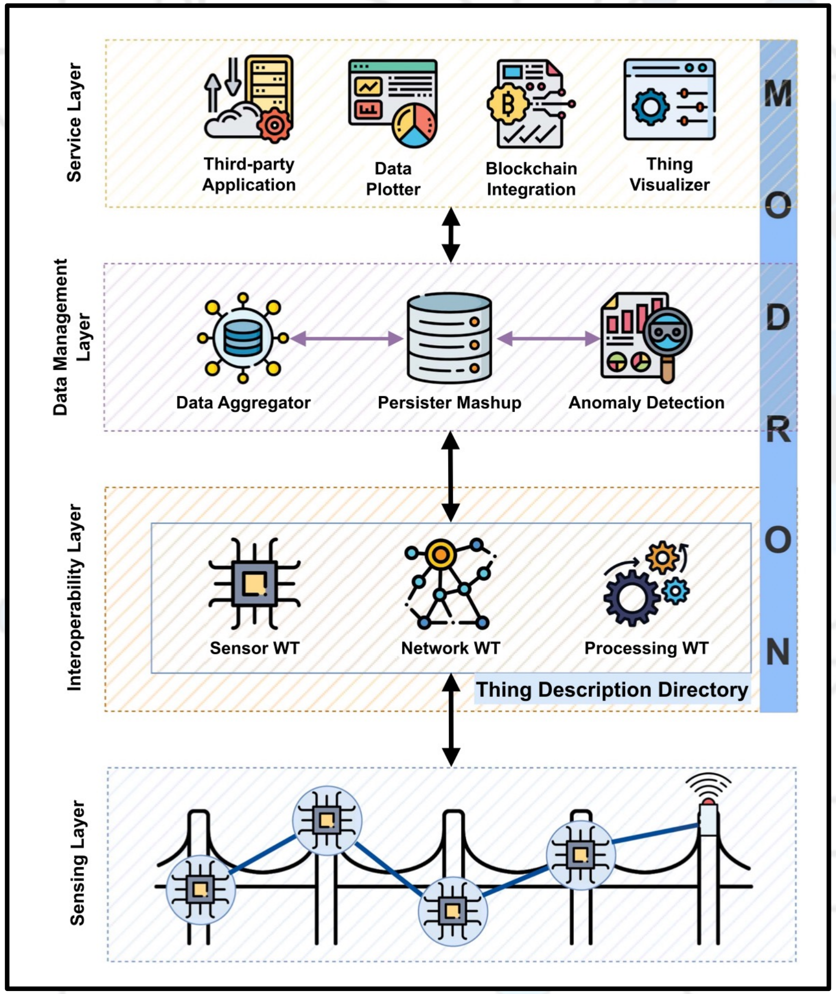
W3C WEB THING



DATA MODEL

- **PROPERTIES**: `ValX, ValY, ValZ`
- **ACTIONS**: `ChangeFrequency()`
- **EVENTS**: `DataOverThreshold`

Monitoring Civil Infrastructures



- **Sensor-related WTs**: We associate a WT to each sensing unit, exposing the data produced as readable properties, the configuration settings as writable properties, and supported commands as actions.
- **Network-related WTs**: We assign a WT to each Sensor Network (SN), modeled as a whole. In such case, the WT includes links to the sensor WTs composing that SN.
- **Processing-related WTs**:We associate WTs with software tasks in charge of processing the sensor data, extracting second-layer information from the monitored structures (e.g. features).

Monitoring Civil Infrastructures

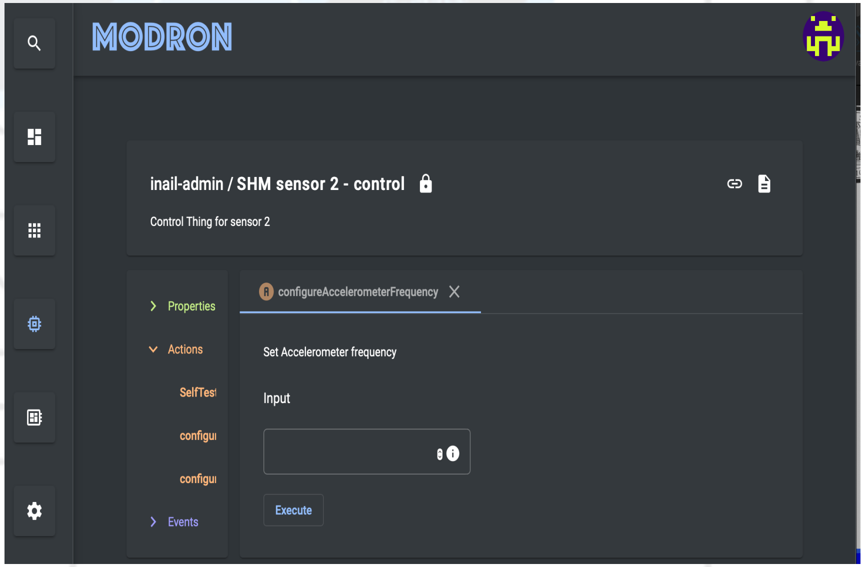
WoT Store: WT Management platform [1]

[1] WoT Store: Managing resources and applications on the web of things L Sciullo, L Gigli, A Trotta, M Di Felice - Internet of Things, 2020

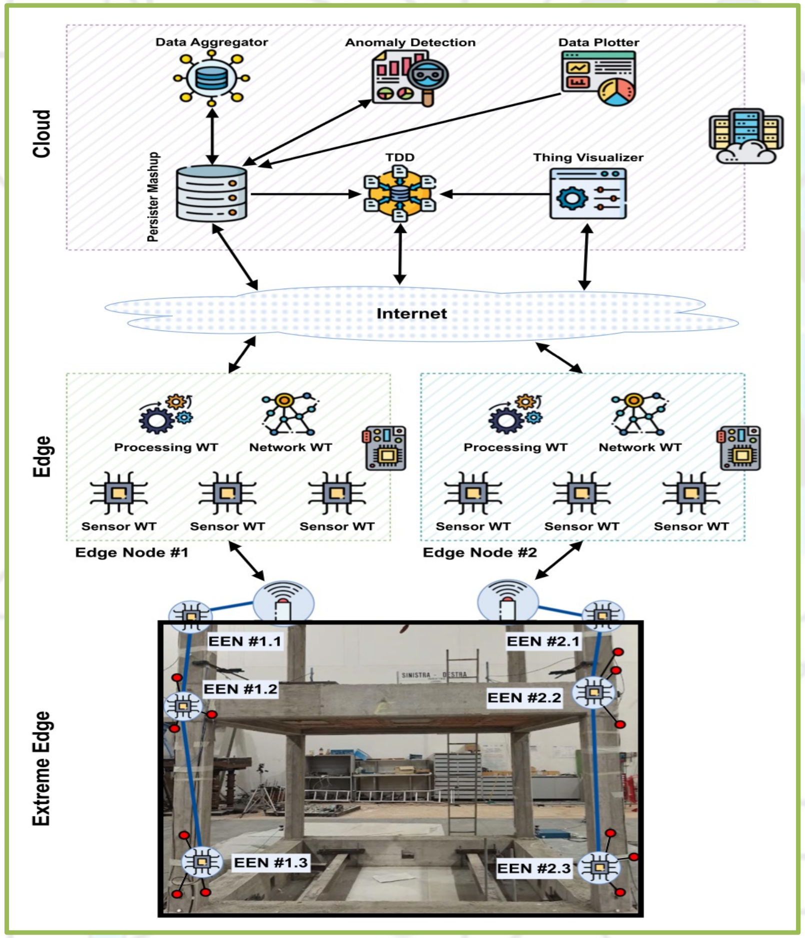
MODRON: WoT Store customization for SHM scenarios [2]

[2] Modron: A scalable and interoperable web of things platform for structural health monitoring, C Aguzzi, L Gigli, L Sciullo, A Trotta, F Zonzini, M. Di Felice et al - 2021 IEEE 18th Annual Consumer Communications, 2021

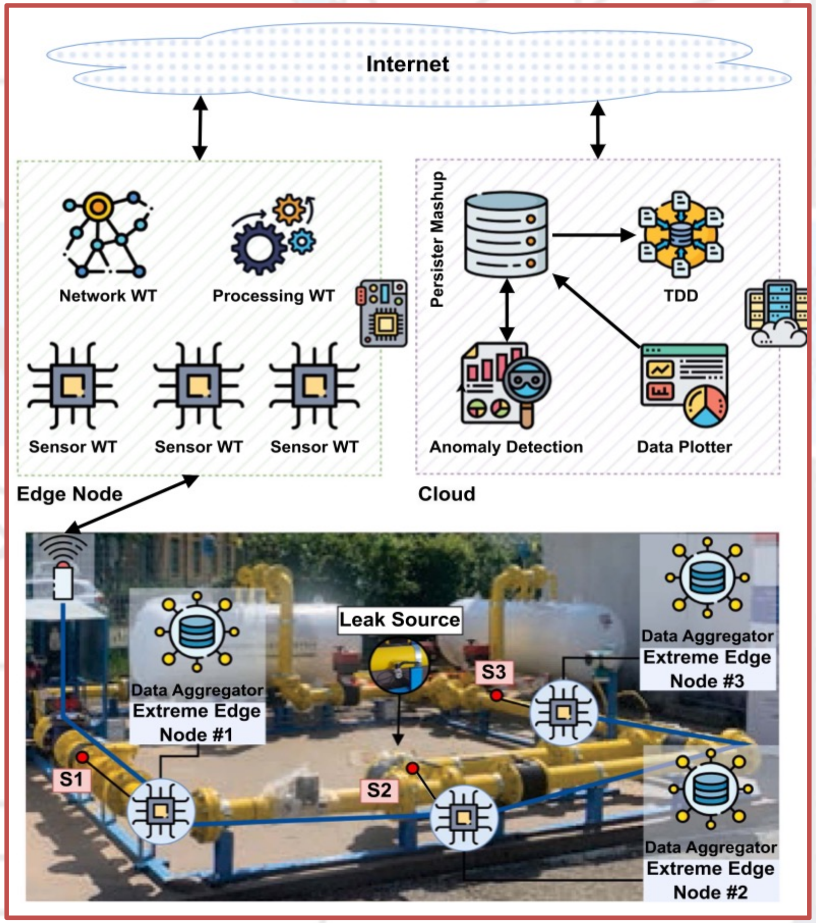
- WT Registration
- Rendering of TD into a Web dashboard
- Monitoring of readable properties
- Editing of writable properties
- Triggering of an action
- …



Monitoring Civil Infrastructures



Deployment plan on the **RC frame**



Deployment plan on the **hydraulic circuit**

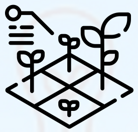
Monitoring Civil Infrastructures

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Objectives

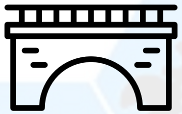
1. Design and development of large-scale/pervasive IoT monitoring systems of physical assets

CONTRIBUTION: Leverage W3C WoT for interoperable data collection



SMART AGRICULTURE

HEALTHCARE



SMART BUILDINGS

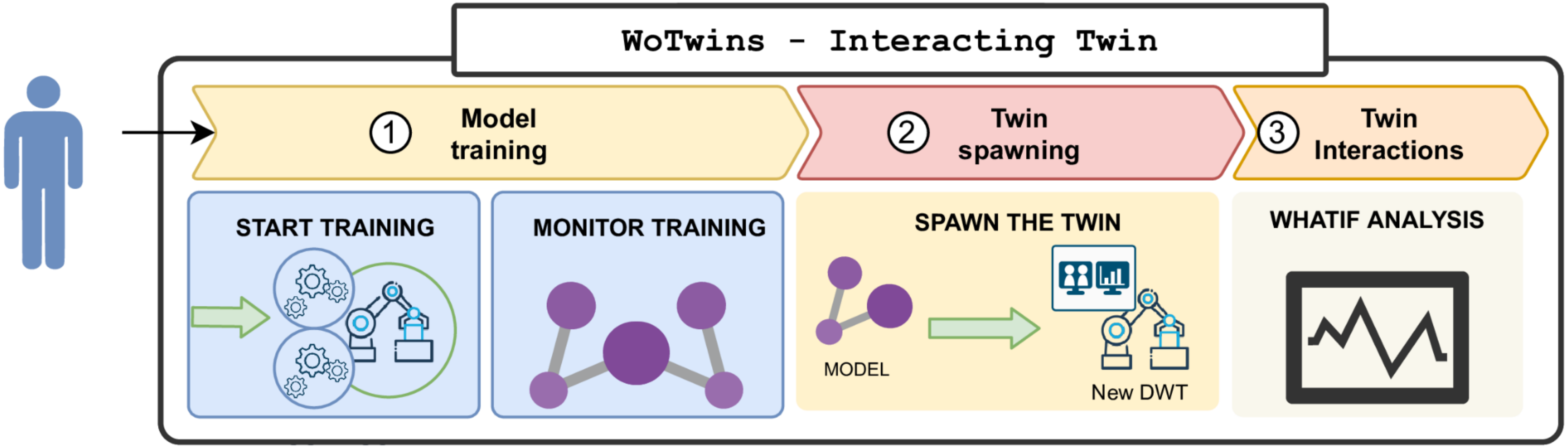
2. Support the generation of Digital Twin (**DT**) of the physical asset

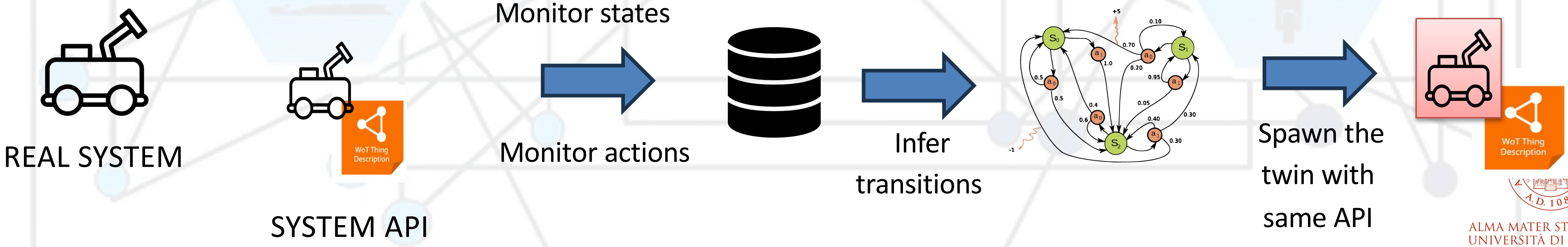
CONTRIBUTION: Leverage W3C WoT towards semi-automatic DT generation

L. Sciullo, A. Trotta, F. Montori, L. Bononi, M. Di Felice, WoTwins: Automatic Digital Twin Generator for the Web of Things, in Proc. of the 1st International Workshop on Massive Digital Twins for the Computer-Networks Evolution (IEEE TwinNets 2022), 14 June 2022 Belfast, UK

Goal 2: Automize DT generation

Automize data collection and Twin spawning



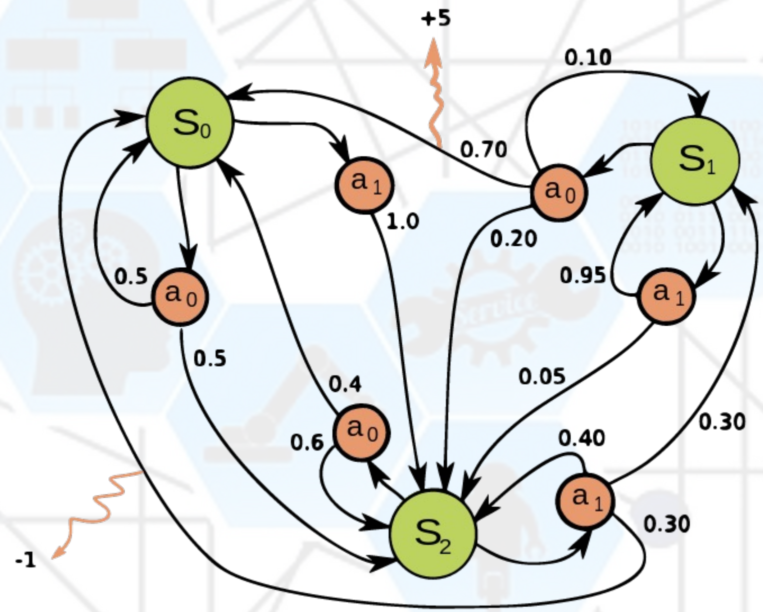


Goal 2: Automize DT generation

Generation of a Digital Twin of a running Web Thing

DT modeled as a *Markov Decision Process* (**MDP**)

- States
- Actions
- Transition probabilities
- Reward function
  (only if we want to achieve a goal)



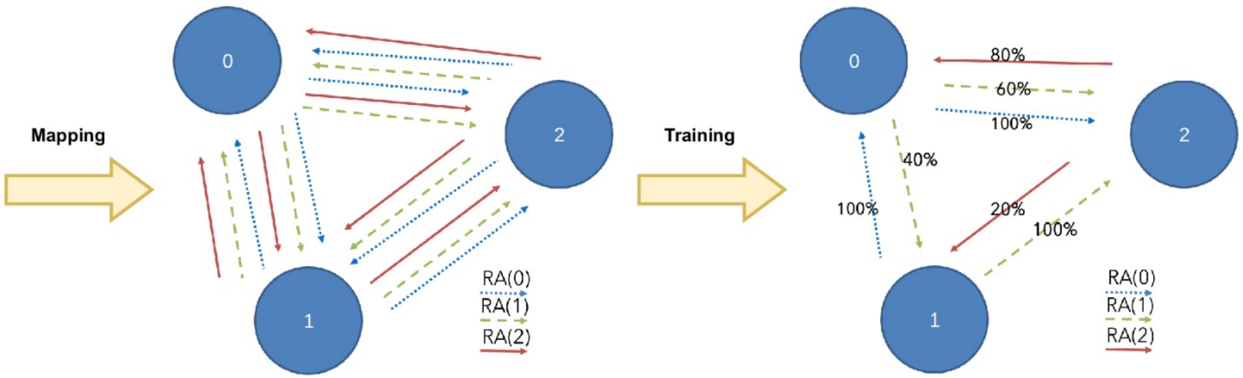
Goal 2: Automize DT generation

Our tool (**WoTwins**) translates the Thing Description into a MDP

- Set of MDP **states** ⟹ Set of *configurations* of the property values in the TD.
  - Property values are discrete and finite
- Set of MDP **actions** ⟹ Set of *actions* in the TD

Thing Description

Name	Affordance Type	Type	min,max (input)
counter	Property	Integer	[0,2]
randomAction(a)	Action	Integer	[0,2]



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Goal 2: Automize DT generation

The transition matrix is built by *observing the behavior of the Web Thing.*

- **Frequency-based**
  The probability T (s, a, s') is estimated from the relative occurrences of the transitions.

- **Machine Learning**
  The probability T (s, a, s') is estimated through an Artificial Neural Network (ANN) trained on the observed transitions.

- **Hybrid**
  It applies the frequency-based approach only if action a in state has been executed a minimum number of times

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

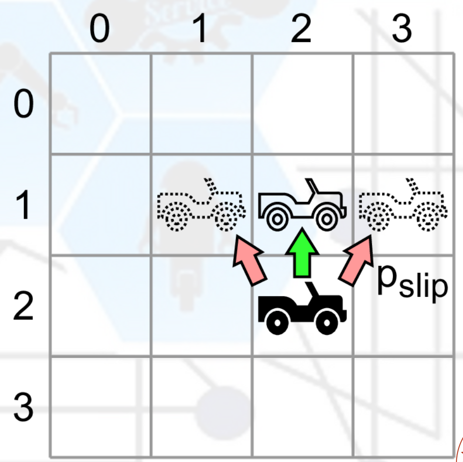
Goal 2: Automize DT generation

DT of a Ground Rover(GR) moving within a grid scenario with **unknown** obstacles

- States are rover's coordinates
- Actions are its movements
- Some cells can have slippery material or an obstacle:

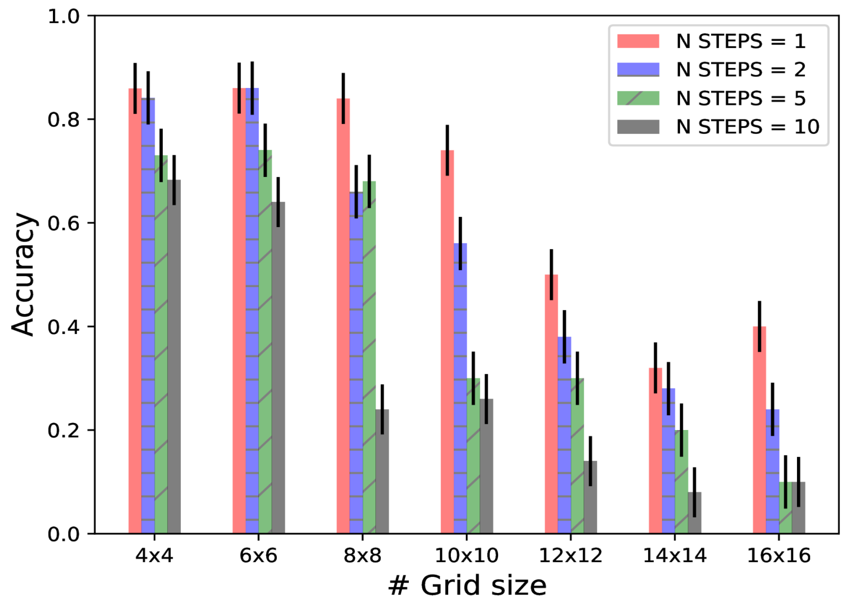

Source: Wikimedia Commons

AFFORDANCE	VALUE	DESCRIPTION
Property	x ∈ [0..N − 1]	The x-coordinate of the GR's position
Property	y ∈ [0..N − 1]	The y-coordinate of the GR's position
Action	move(d)	Move the GR. Here d is the possible direction with admissible values: d ∈ {up, down, left, right, stop}

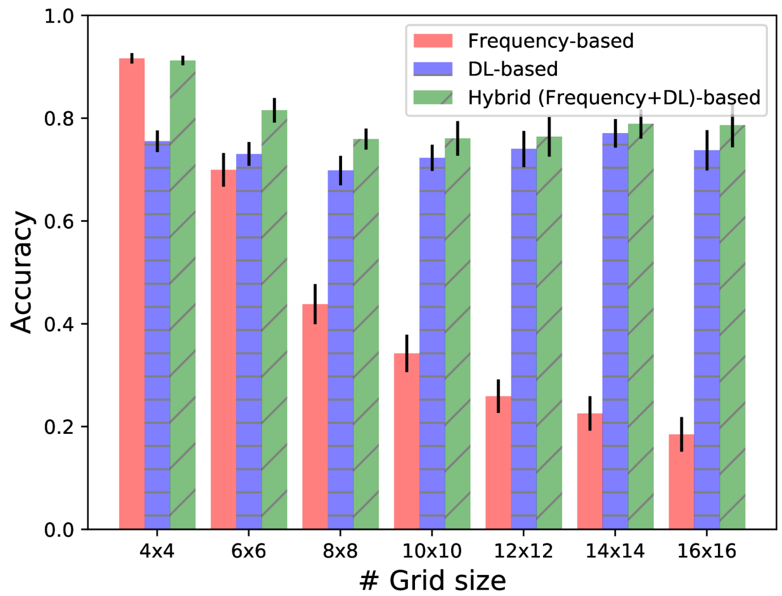


ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Goal 2: Automize DT generation



Impact of the **grid size** for Frequency-based approach



Impact of the **type of estimator**

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Goal 2: Automize DT generation

Future Generation Computer Systems 153 (2024) 521–536

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Relativistic Digital Twin: Bringing the IoT to the future

Luca Sciullo [a,b,*], Alberto De Marchi [c], Angelo Trotta [a], Federico Montori [a,b], Luciano Bononi [a,b], Marco Di Felice [a,b]

[a] Department of Computer Science and Engineering, University of Bologna, Italy
[b] Advanced Research Center on Electronic Systems, University of Bologna, Italy
[c] Institute of Applied Mathematics and Scientific Computing, Department of Aerospace Engineering, University of the Bundeswehr Munich, Germany

ARTICLE INFO

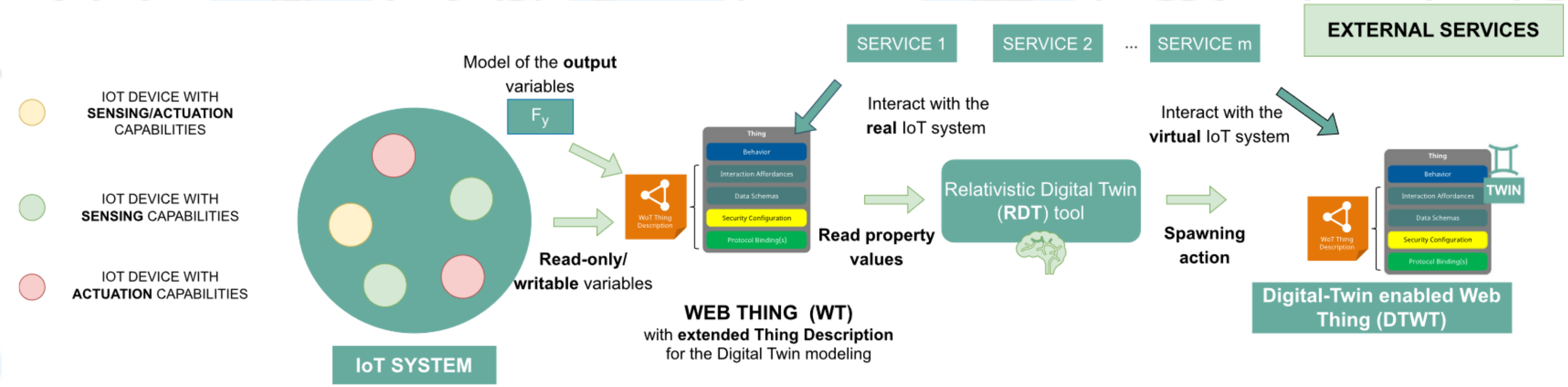
Keywords:
Internet of Things (IoT)
Digital Twin (DT)
Web of Things (WoT)
Machine learning

ABSTRACT

Complex IoT ecosystems often require the usage of Digital Twins (DTs) of their physical assets in order to perform predictive analytics and simulate what-if scenarios. DTs are able to replicate IoT devices and adapt over time to their behavioral changes. However, DTs in IoT are typically tailored to a specific use case, without the possibility to seamlessly adapt to different scenarios. Further, the fragmentation of IoT poses additional challenges on how to deploy DTs in heterogeneous scenarios characterized by the usage of multiple data formats and IoT network protocols. In this paper, we propose the Relativistic Digital Twin (RDT) framework, through which we automatically generate general-purpose DTs of IoT entities and tune their behavioral models

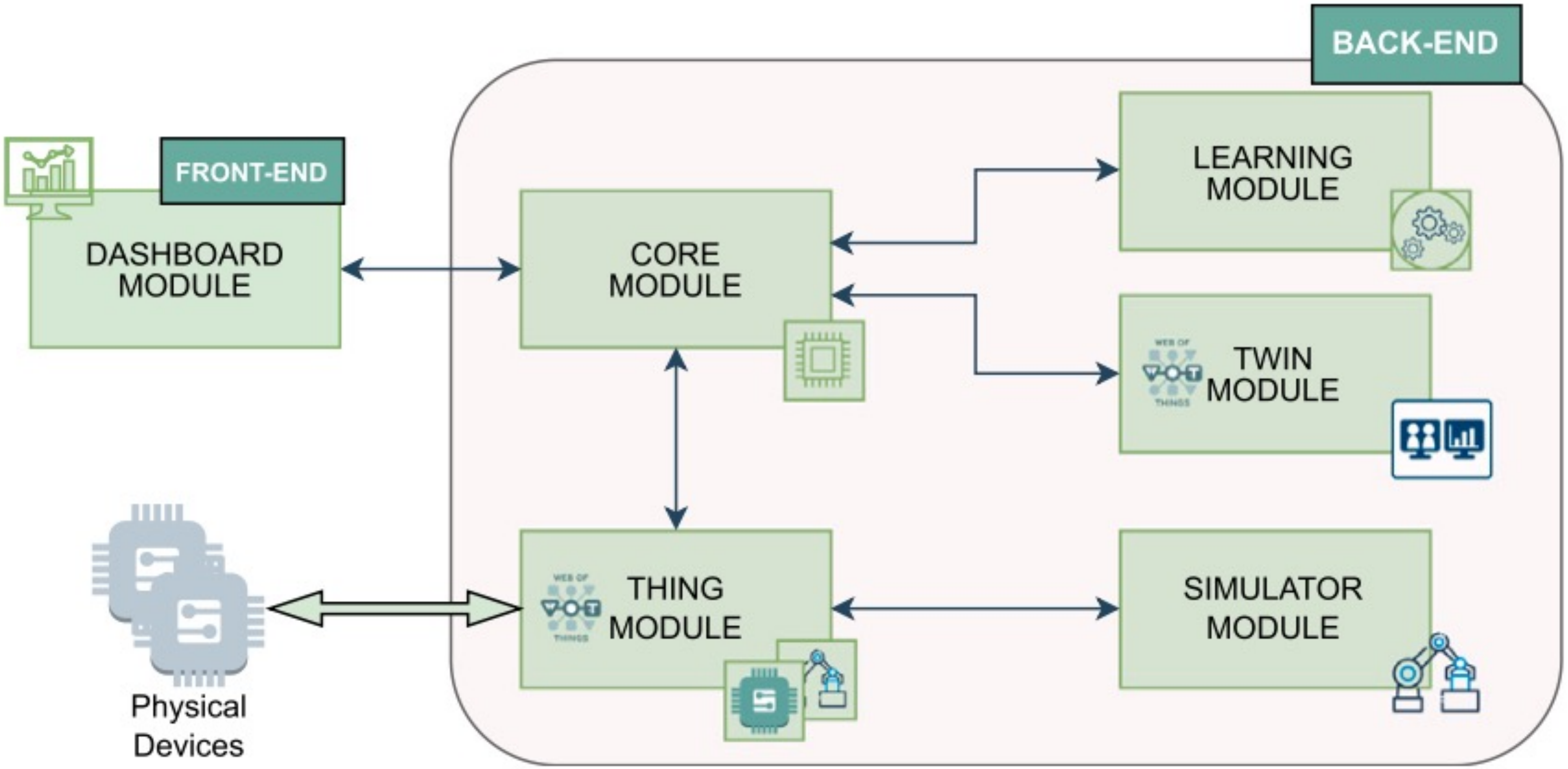
Goal 2: Automize DT generation

Same interaction flow than the previous `WoTwins` framework.



- Embed **domain knowledge** into the DT generator
  - **Behavioral** model: IoT system described by a set of equations over time
  - **Application-agnostic** approach
  - The DT can interact with the other elements in the system without having to stick to the current time.

Goal 2: Automize DT generation



Goal 2: Automize DT generation

DT Model formalization

- **Read-only** variables
- **Writable** variables

Each **readable** property is associated with a set of functions that describes its behavior:

- **Algebraic** equations: allow to model the instantaneous value of specific quantities

$$a_j(t) = g_j(t, B(t), W(t), P_j)$$

- **Differential** equations: describe the evolution of physical quantities over time

$$\dot{b}_j(t) = f_j(t, B(t), A(t), W(t), P_j)$$

Parameters must be estimated from data!

We assume the model knowledge (shape of g and f)

Thing Description Extension

We extended the **W3C TD** syntax to allow the description of a behavioral model

Term (dtwt:)	Description	Class	Type
model	Mathematical model of the property	PropertyAffordance	String
modelInput	Container for inputs of the model	PropertyAffordance	Class
ValueFrom	Indication from where to take the value of the property, i.e., from the physical sensor or from the computed mode	PropertyAffordance	String

Thing Description Extension: Model

The field model is an arithmetical expression, encoded in a Python-like syntax, that describes the behavior of the property. This is basically represented via a function that is provided by the manufacturer of the appliance

{ behavior } = { function } | {constraints } | { guess }

It is a set of initial value assignment regarding the params of the model

It is a set of preconditions regarding the params of the model

It is an arithmetical expression, encoded in python-like syntax. We have added a few keywords that point to definite values in our TD

It can be an algebraic expression, that is a fixed value that has no dynamics over time and depends on top of some static properties or a differential equation, therefore it depends on time and outputs its integration over time

Thing Description Extension: modelInput

The field modelInputs is a list of other properties of the WT that are used within the model. Each of these has five fields:

- **title:** a friendly name
- **propertyName:** the exact name of the property of the WT that this input is referring to
- **type:** the JSON-LD type
- **model:** a model (using the same syntax as described previously) that defines how this input affects the model of the property
- **modelType**: an optional field denoting a class assigned to this input that is only valid within the scope of the property and is used to aggregate all inputs belonging to the same modelType (if there is more than one)

The property model can use the modelInputs by using the construct **input({title})** or **inputType({modelType})**

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Thing Description Extension: ValueFrom

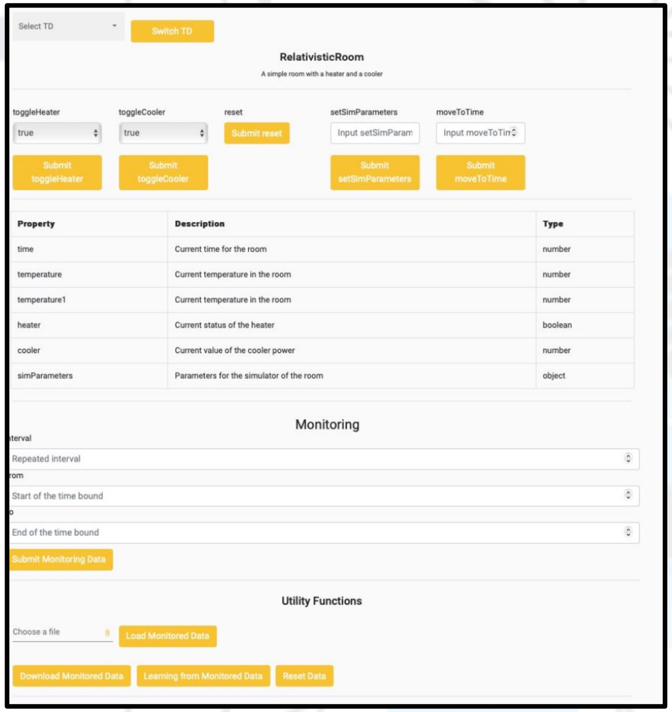
The field **valueFrom** describes if the real property:

- can be read (e.g., a real sensor), in such case the value of this field will be set to ‘‘readProperty’’.
- can be set to “model’’, which instead triggers the simulation of the property value (depending on the behavior, via algebraic evaluation or integration over time)

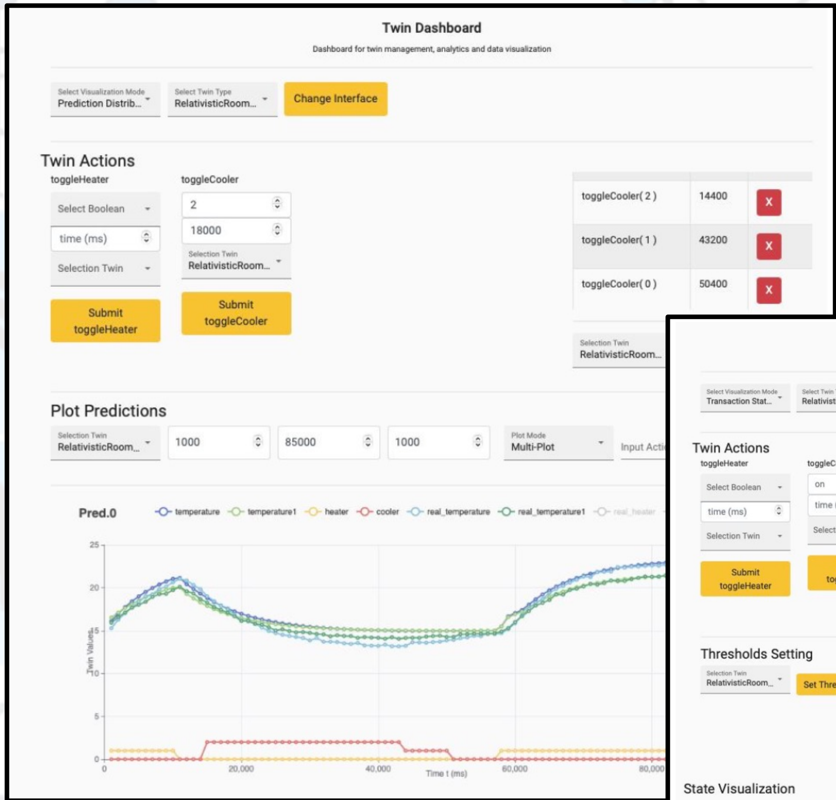
TD Extension: Example

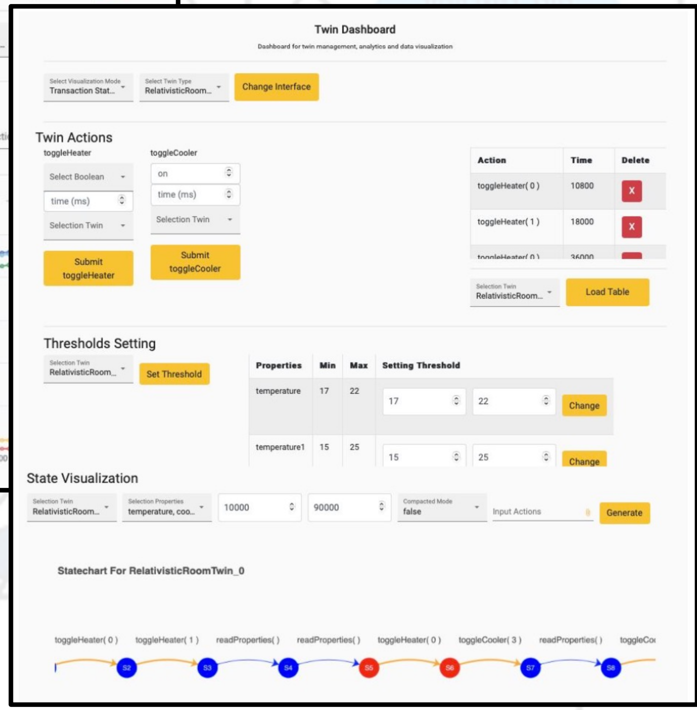
```
{
…
"valueFrom": "model",
"modelInput": [{
          "title": "coolerSetpoint",
          "type": "number",
          "model": "max(0, min(round(value()), 9))"}
          ],
"model": "dot(self) = params[0] * (params[1] *
            input(coolerSetpoint) - self) | params[0] >= 0.0,
            params[1] >= 0.0 | params[1] = 0.1, params[0] = 0.1"
..
}
```

Goal 2: Automize DT generation



DASHBOARD: THING MONITORING (training)





DASHBOARD: DT MONITORING & CONTROL

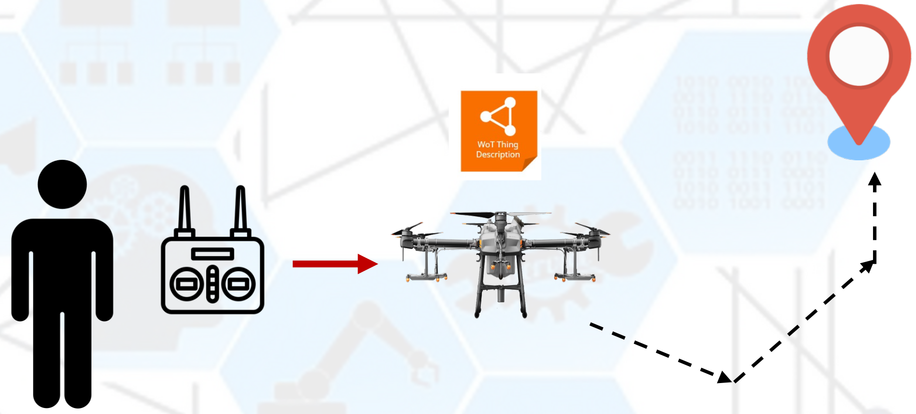
Goal 2: Automize DT generation

Use case:

DT of a quadcopter (*DJI* drone)

The system is controllable through these commands (**write-only** properties):

- **Throttle**(Th): commands climb/decent movement of the copter. Positive values of Th are for climbing request, descent otherwise.
- **Rudder**(Ru): t commands the yaw movements. Positive values of Ru are for the clockwise rotation of the copter.
- **Elevator**(El), that commands the pitch, i.e., the forward and backward movements. Here $El > 0$ means forward, backward otherwise.
- **Aileron**(Al), that commands the roll, i.e., the left and right movements, where $Ai > 0$ means left movements, right direction otherwise.



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

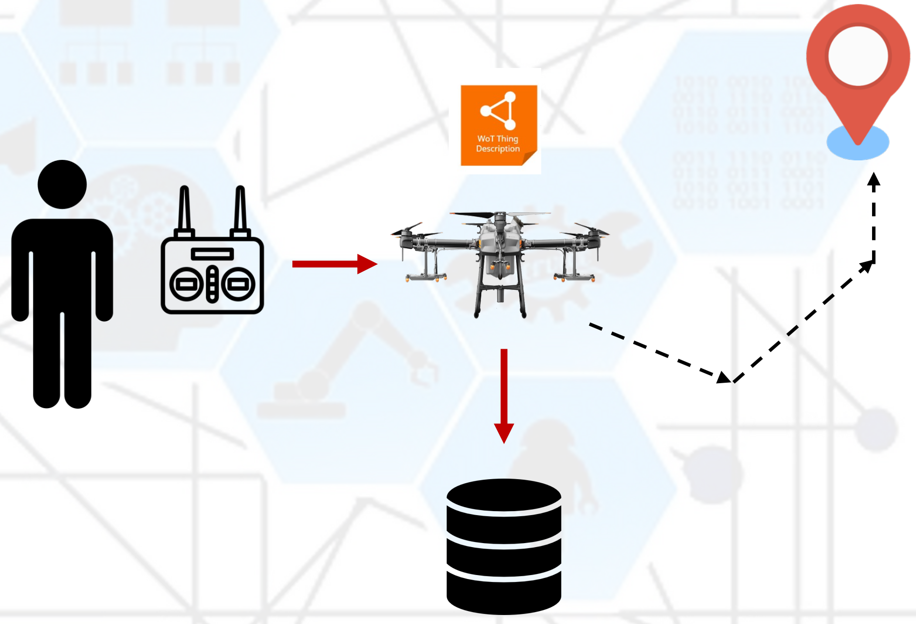
Goal 2: Automize DT generation

Use case:

DT of a quadcopter (*DJI* drone)

We measured the following quantities (**read-only** properties):

- *latitude, longitude -> mapped to x,y coordinates*
- *altitude -> mapped to zcoordinates*
- *compass (yaw)*
- *elevator, aileron, throttle, rudder ->* joystick user's inputs.



ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Goal 2: Automize DT generation

Use case:

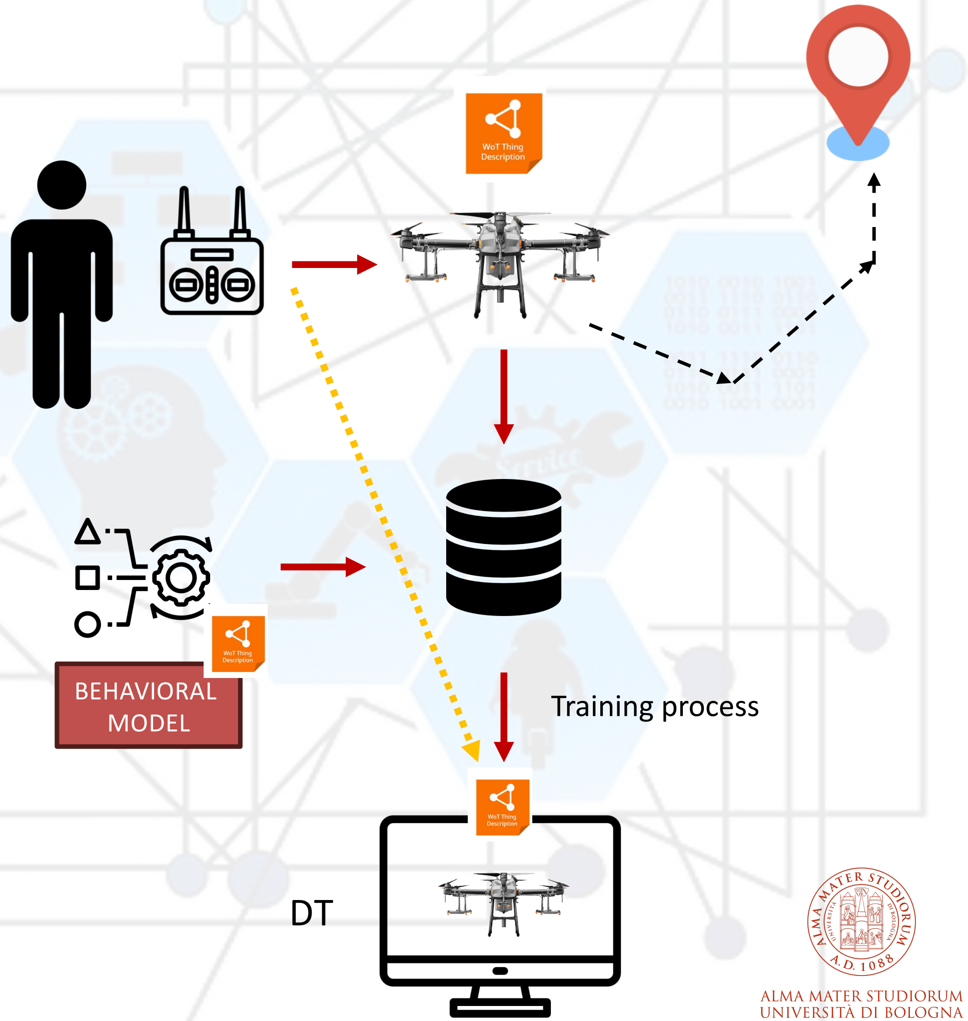
DT of a quadcopter (*DJI* drone)

Behavioral model: *Second-order linear dynamics model in the body reference frame*

LEARNT FROM DATA

$$\dot{q}_x = q_{vx}$$
$$\dot{q}_y = q_{vy}$$
$$\dot{q}_z = q_{vz}$$
$$\dot{q}_\psi = q_{v\psi}$$
$$\dot{q}^B_{vx} = \alpha^D_1(\alpha^D_2 El - q^B_{vx})$$
$$\dot{q}^B_{vy} = \alpha^D_3(\alpha^D_4 Ai - q^B_{vy})$$
$$\dot{q}_{vz} = \alpha^D_5(\alpha^D_6 Th - q_{vz})$$
$$\dot{q}_{v\psi} = \alpha^D_7(\alpha^D_8 Ru - q_{v\psi})$$

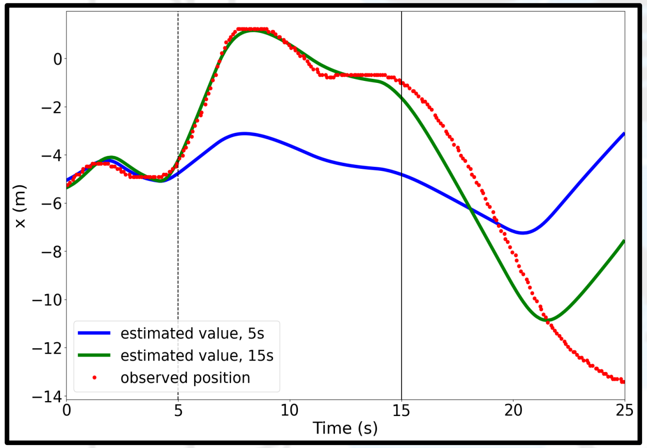
$$\begin{bmatrix} q^B_{vx} \\ q^B_{vy} \end{bmatrix} = \mathrm{RT}(q_\psi)^\top \begin{bmatrix} q_{vx} \\ q_{vy} \end{bmatrix}$$
$$\begin{bmatrix} \dot{q}_{vx} \\ \dot{q}_{vy} \end{bmatrix} = \mathrm{RT}(q_\psi) \begin{bmatrix} \dot{q}^B_{vx} \\ \dot{q}^B_{vy} \end{bmatrix} + \mathrm{RT}'(q_\psi) q_{v\psi} \begin{bmatrix} q^B_{vx} \\ q^B_{vy} \end{bmatrix}.$$



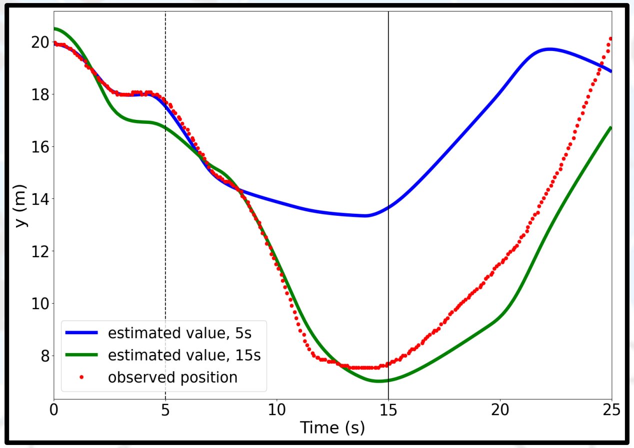
ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Goal 2: Automize DT generation

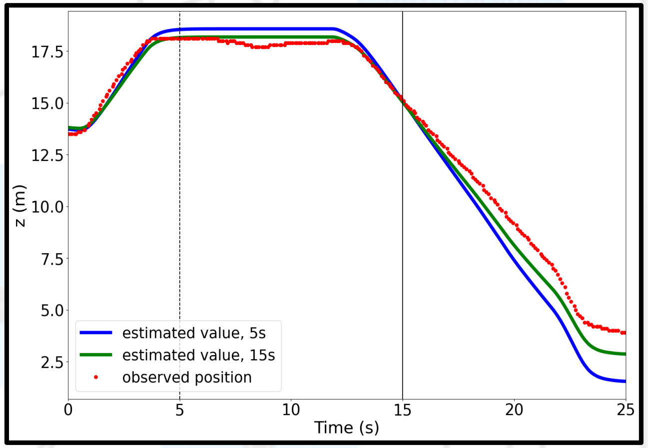
Use case: **DT of a quadcopter** (*DJI* drone)



X Coordinate



Y Coordinate



Z Coordinate

3D Coordinates of the drone. ***Observed*** position vs ***Estimated***

Ongoing activities

- W3C WoT in other IoT domains (Healthcare)
- Integrating **beahavioural model learning** and **MDP parameters learning** via DL techniques
- Extending the Relativistic WoT framework for **Federated Digital Twins** environments

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

THANK YOU

Prof. Marco Di Felice and Dr. Luca Sciullo

Department of Computer Science and Engineering
University of Bologna,Italy

marco.difelice3@unibo.it

Icons are courtesy from FlatIcon (https://www.flaticon.com/)

www.unibo.it