UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Digital Twin: how, where and why…

Digital Twin: la strada 4.0 per l'efficienza dei processi e la qualità dei prodotti
2019 07 May

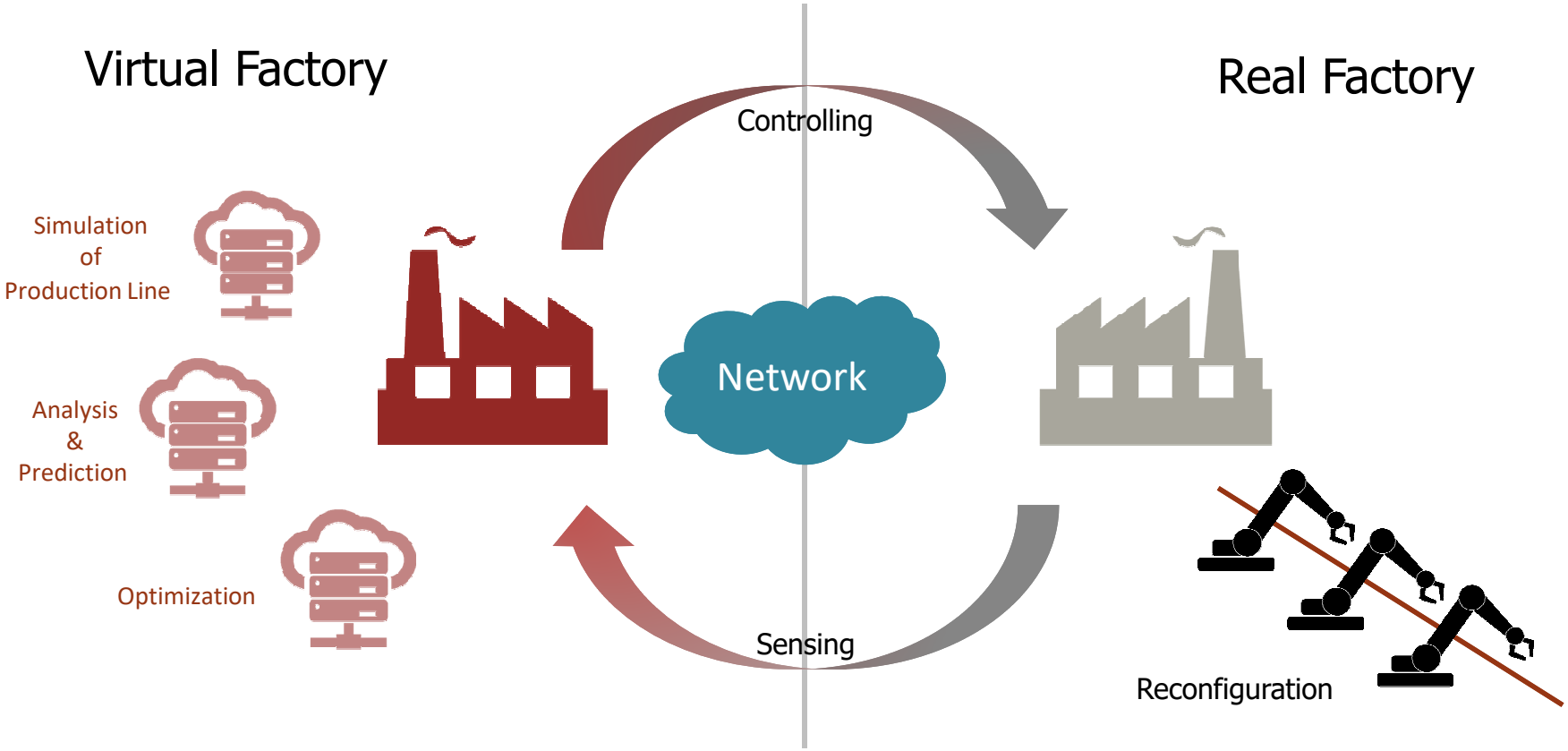
Franco Fummi

Computer Science Department
University of Verona - Italy

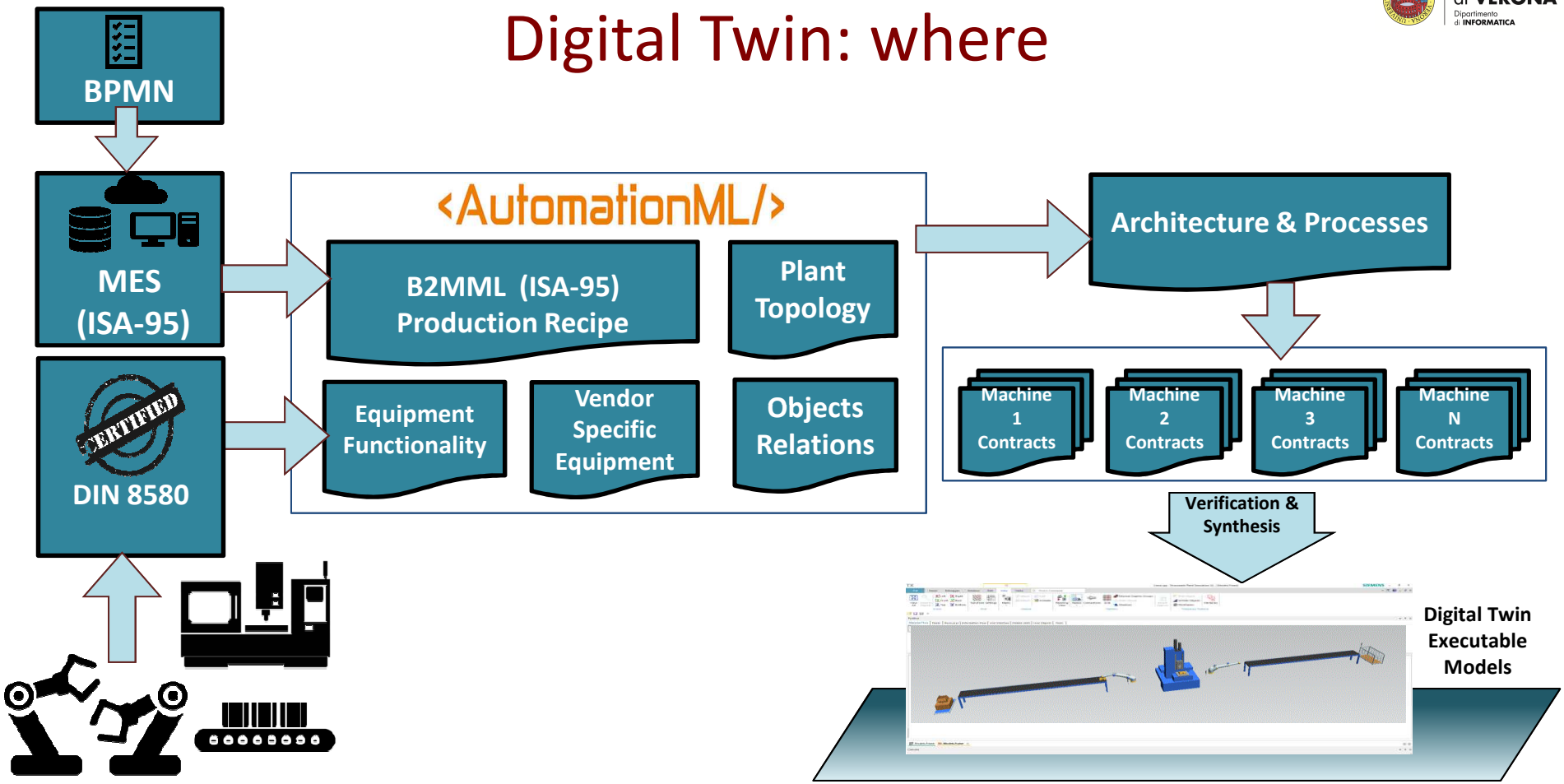
improvenet
ICT for Smart Manufacturing Processes
Veneto Network

Dipartimento di Eccellenza 2018/2022

Digital Twin: why



Digital Twin: where



UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Stefano Centomo – Franco Fummi

DEFINING AND INTEGRATING MODELS

Business Process Model

Activities

Task

A **Task** is a unit of work, the job to be performed.

Transaction

A **Transaction** is a set of activities that logically belong together.

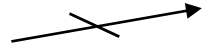
Event Sub-Process

An **Event Sub-Process** is placed into a Process or Sub-Process. It is activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (non-interrupting) depending on the start event.

Call Activity

A **Call Activity** is a wrapper for a globally defined Sub-Process or Task that is reused in the current process.

A **Sequence Flow** defines the execution order of activities.



A **Default Flow** is the default branch to be chosen if all other conditions evaluate to false.

A **Conditional Flow** has a condition assigned that defines whether or not the flow is used.

Gateways

Exclusive Gateway - without Marker

Exclusive Gateway - with Marker

Event-based Gateway

Parallel Gateway

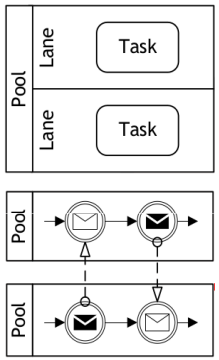
Inclusive Gateway

Complex Gateway

Exclusive Event-based Gateway

Parallel Event-based Gateway

Swimlanes



Pools and **Lanes** represent responsibilities for activities in a process.

The **order** of **message exchanges** can be specified by combining message flow and sequence flow.

Others

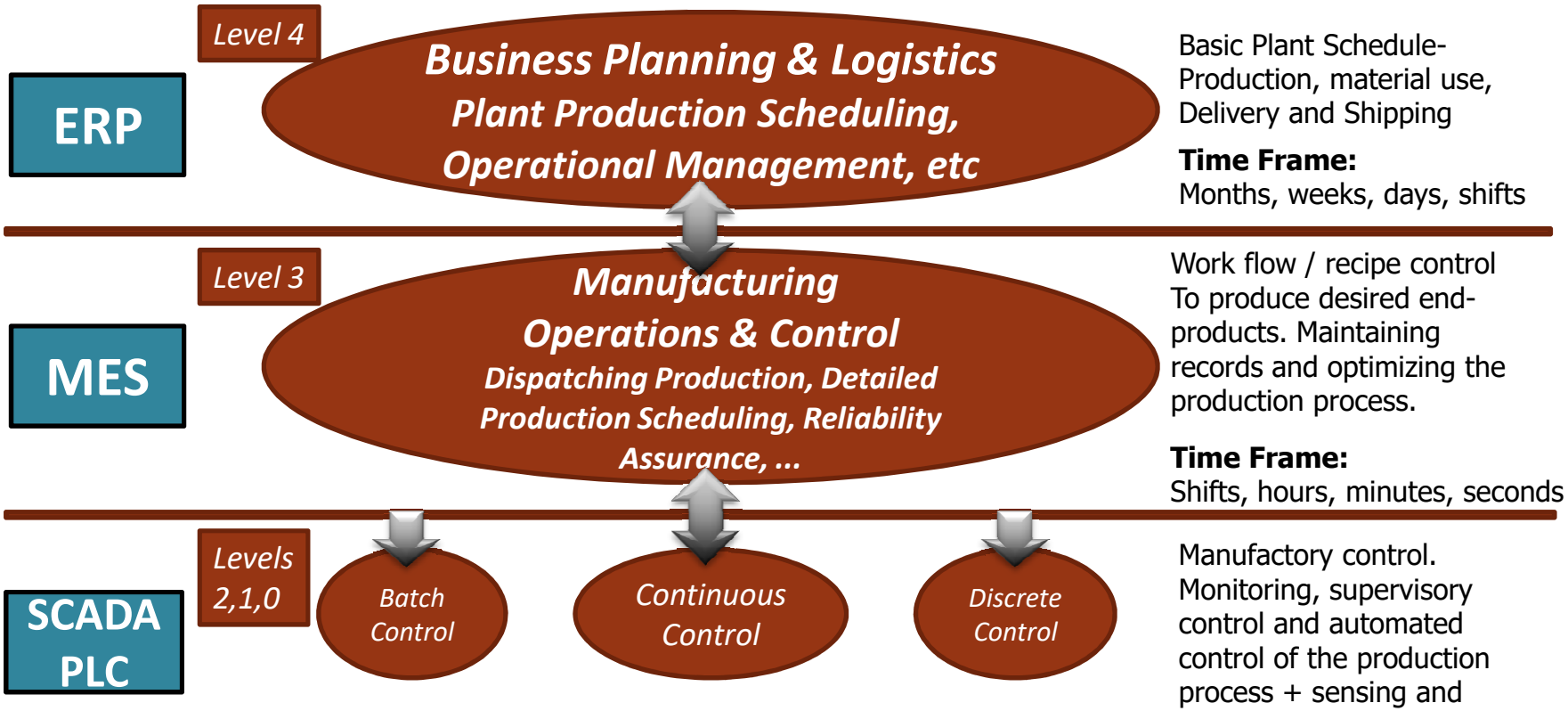
Receive Task

Send Task

Message: Receiving and sending messages.

Timer

ISA-95: Functional Hierarchy Levels description

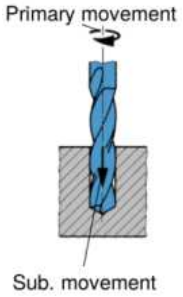


B2MML as a Solution

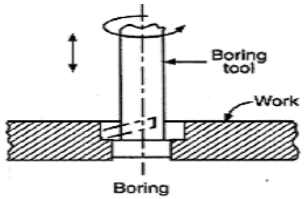
- **B2MML**: Business to Manufacturing Mark up language
  - Implementation of ISA-95 in XML
  - XML elements which comprise information
  - providing a generic / common / extendable platform
  - For data exchange between scheduling component and manufacturing environment

DIN 8580 Taxonomy: the milling machine

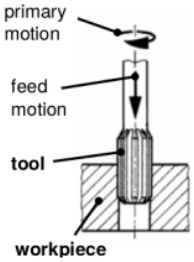
3.2.2.1. (DIN)
1.1.1.1.2.1. (Todd)
Drilling
Drilling for making cylindrical hole



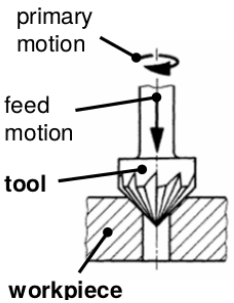
3.2.2.2. (DIN)
1.1.1.1.1.2. (Todd)
Boring
Boring for enlarge drilling hole



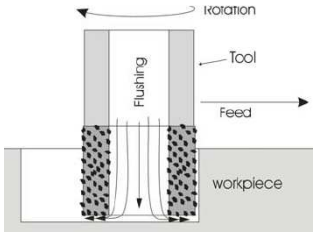
3.2.2.3. (DIN)
1.1.1.1.2.2. (Todd)
Reaming
Reaming for finishing hole or slightly remove material from hole



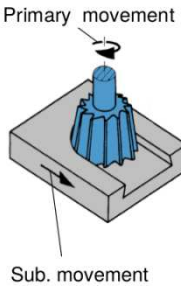
3.2.5. (DIN)
Countersink



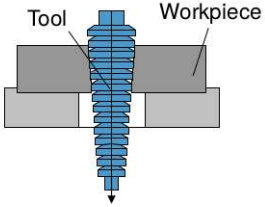
3.3.1. (DIN)
Grinding with rotating tool



3.2.3. (DIN)
1.1.1.1.2.3. (Todd)
Milling



3.2.9. (DIN)
1.1.1.1.2.4. (Todd)
Broaching

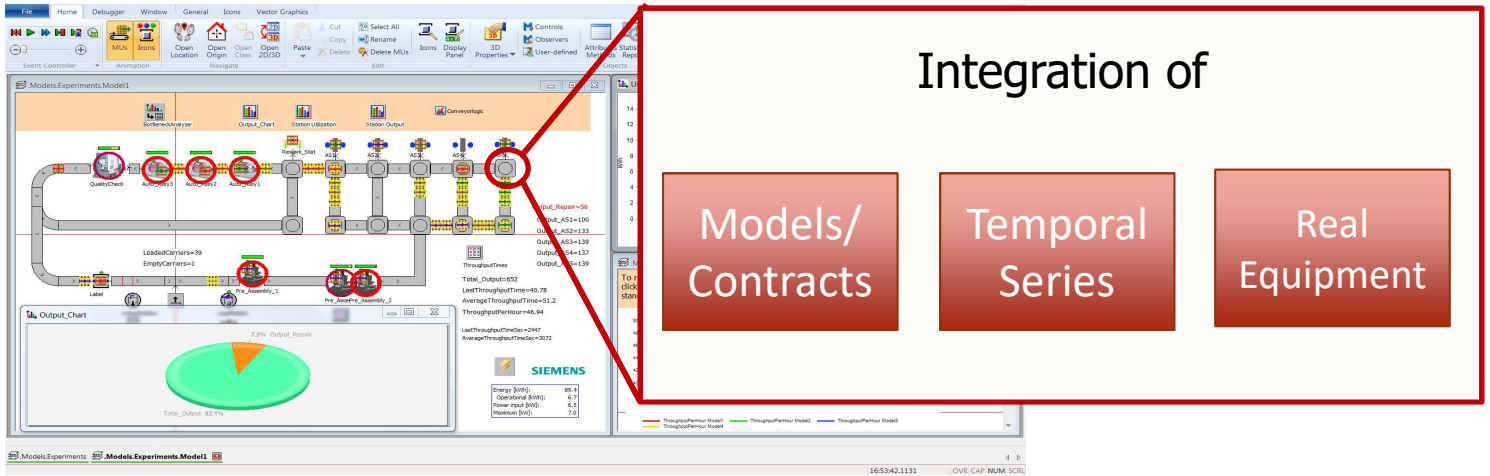


Simulation Infrastructure

- Several tools to model & simulate production lines (**Siemens**, FlexSim, Simio, Simul8 ecc..)
- Easy and really intutive to use (Drag & Drop components from a library)

High level of abstraction → Fast simulation but... **Loss of details!**

TecnoMatix Plant Simulation (Siemens)

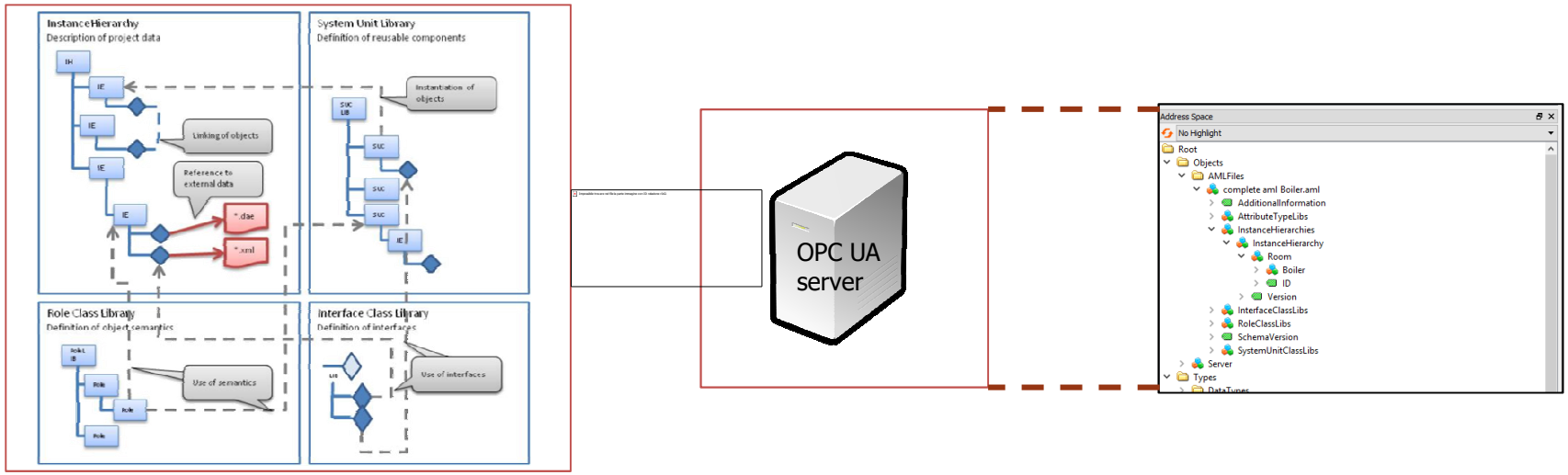


Standards for Models Integration

- Model encapsulation
  - FMI Standard
  - OPC-UA
- Functional Mockup Interface (FMI)
  - Standard interface to exchange models
- OPC-UA
  - Machine to machine communication protocol for industrial automation

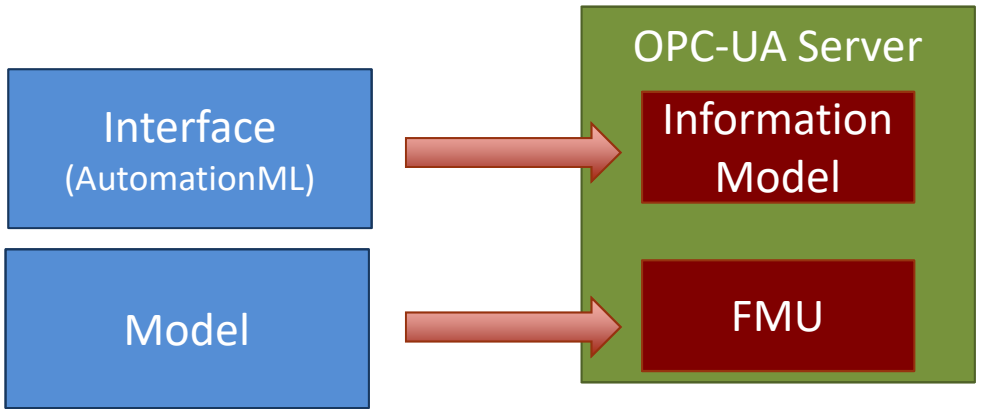
Automation Markup Language (AML)

- **AutomationML**
  - a neutral data format based on XML for the storage and exchange of plant engineering information
  - it is provided as open standard
  - to interconnect heterogeneous tools
- Mapping from the AML domain to the OPC UA domain



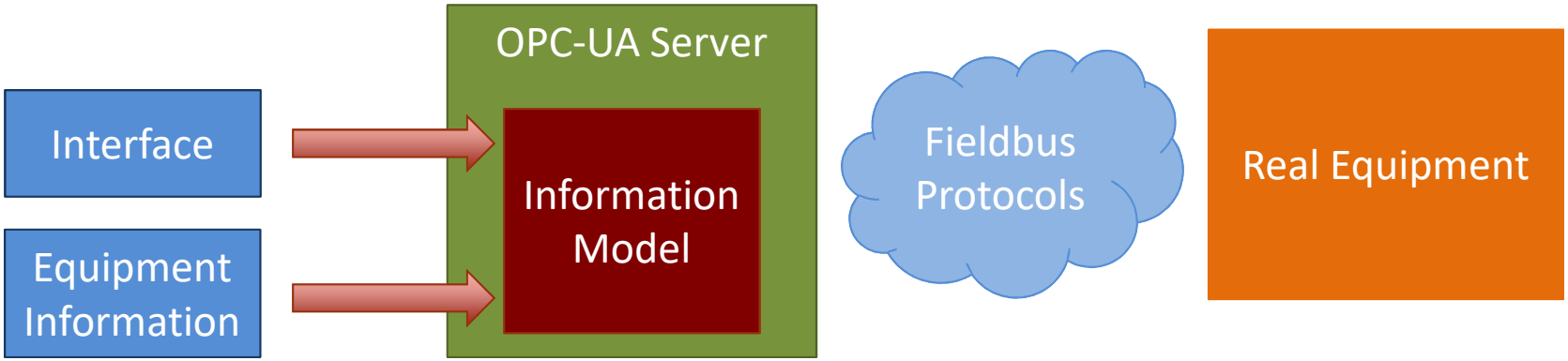
OPC-UA Server Design

1. Interface Definition (AutomationML)
2. AutomationML to OPC-UA Information Model
3. FMU Generation
4. FMU Integration in OPC-UA Server

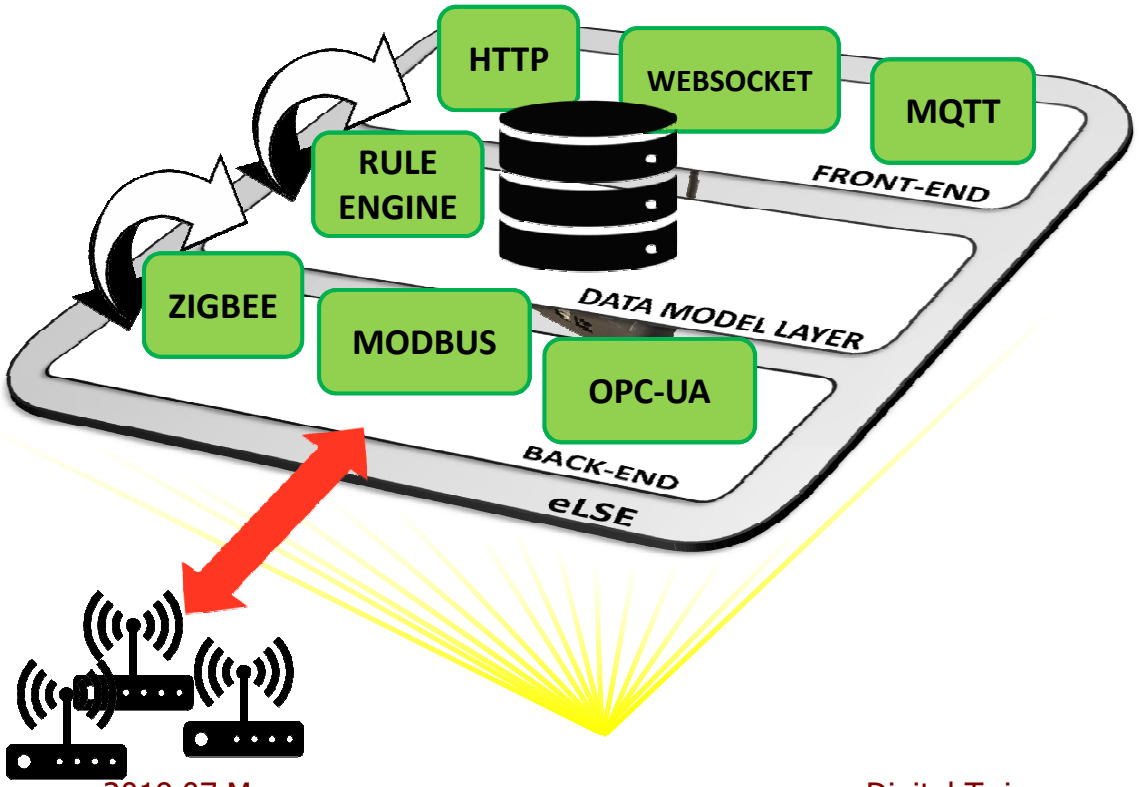


OPC-UA Server for a Real Equipment

1. Interface Definition (AutomationML)
2. Equipment Classification (ISA-88/95)
3. OPC-UA Information Model Generation (Step 1 + Step 2)



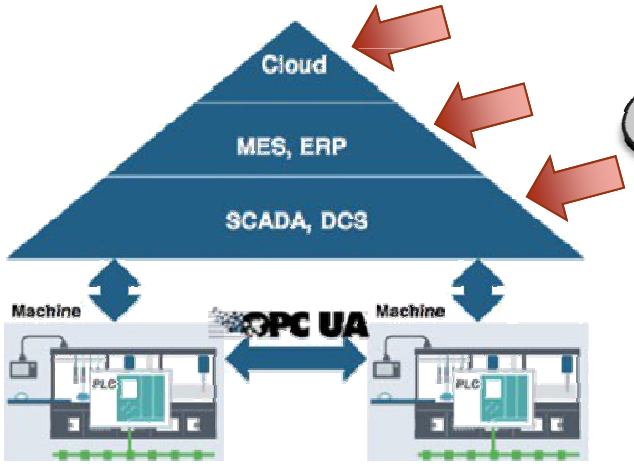
BOX-IO

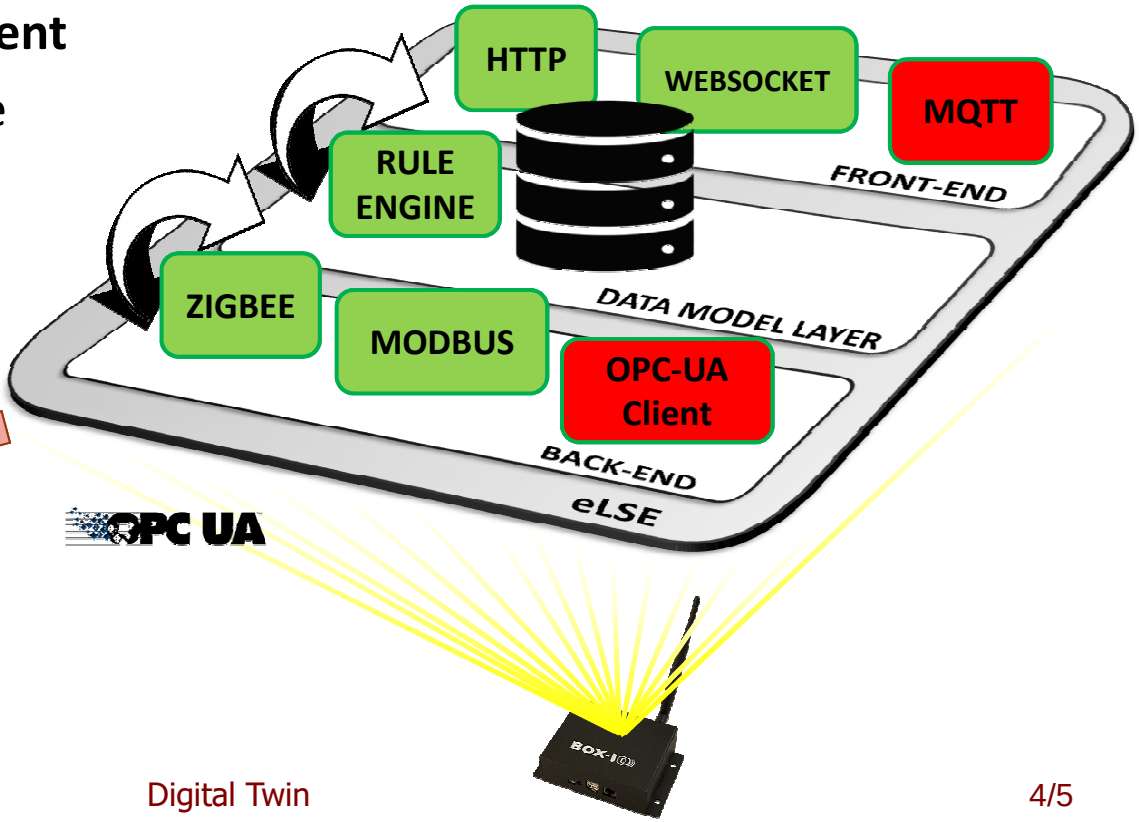


- BOX-IO is a data aggregator
  - it can be connected to nodes that can be **sensors** and **actuators**
- The «heart» of BOX-IO software is eLSE
- eLSE is composed of three parts:
  - **Frontend Layer** for interaction with the application and the site for example
  - **Data model Layer** for modeling data
  - **Backend Layer** for interacting directly with the devices
- The three levels can communicate with each other

BOX-IO as a Link to Cloud

- It can be placed in the plant as an **advanced equipment**
- At **different levels** of the automation hierarchy





UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Roberta Chirico – Franco Fummi

FROM CONTRACT TO MODELS

Machine library of A/G contracts

Machine: 1

Actions from DIN Taxonomy: 2

DIN Taxonomy → Machine Capabilities

Elementary actions: 3

Action1 | Action2 | … | ActionN

A/G Contracts Library: 4

Assumptions ↕ Plant Environment

Guarantees ↕ Machine Behaviours (Actions)

Library example: Milling machine

Machine: 1

Actions from DIN Taxonomy: 2

DIN Taxonomy →
- 3.2.2.1. Drilling
- 3.2.2.2. Boring
- 3.2.2.3. Reaming
- 3.2.3. Milling
- 3.2.5. Countersink
- 3.2.9. Broaching

Elementary actions: 3

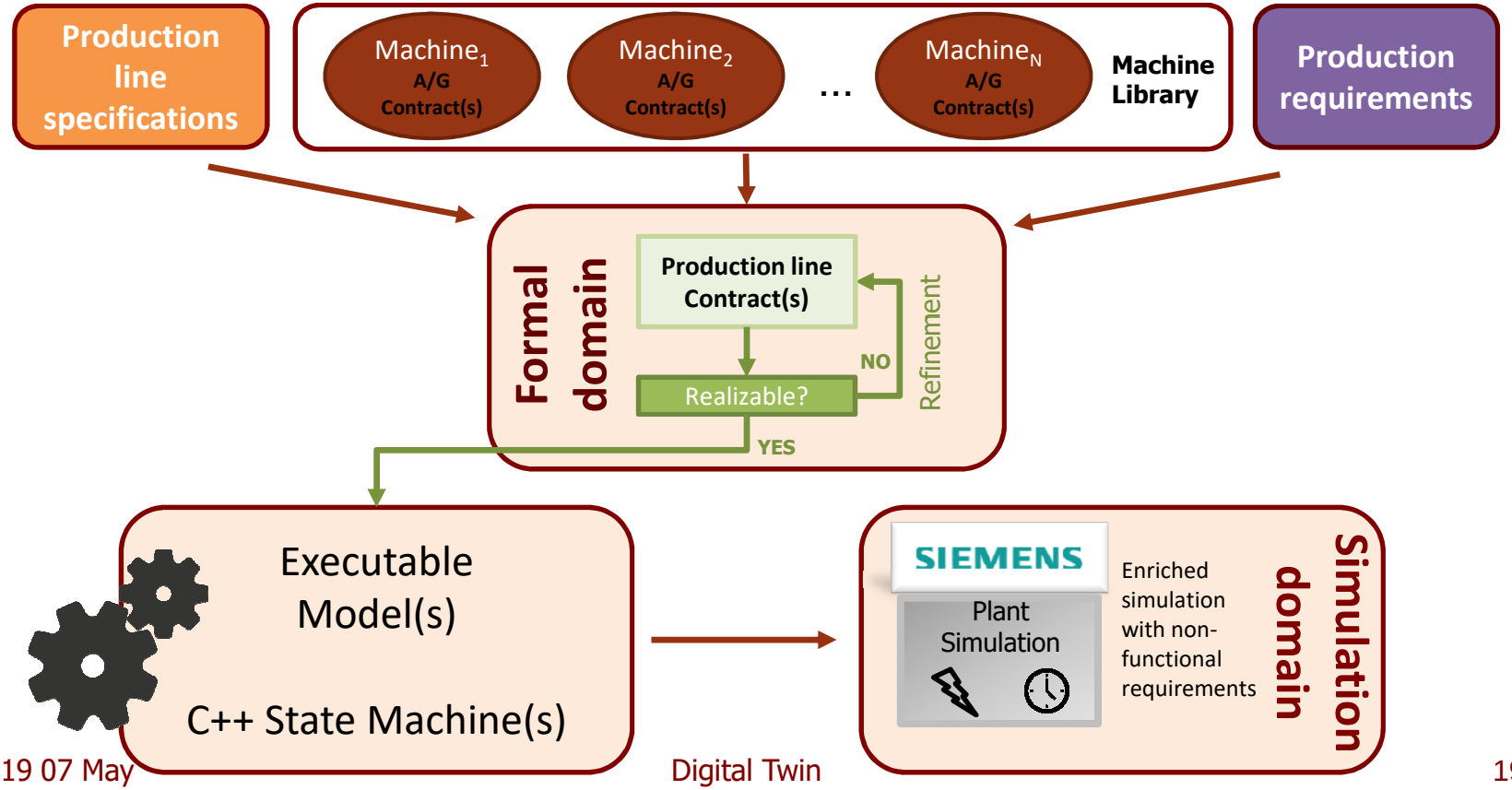
Drilling | Boring | Reaming | Milling | Countersink | Broaching

A/G Contracts Library: 4

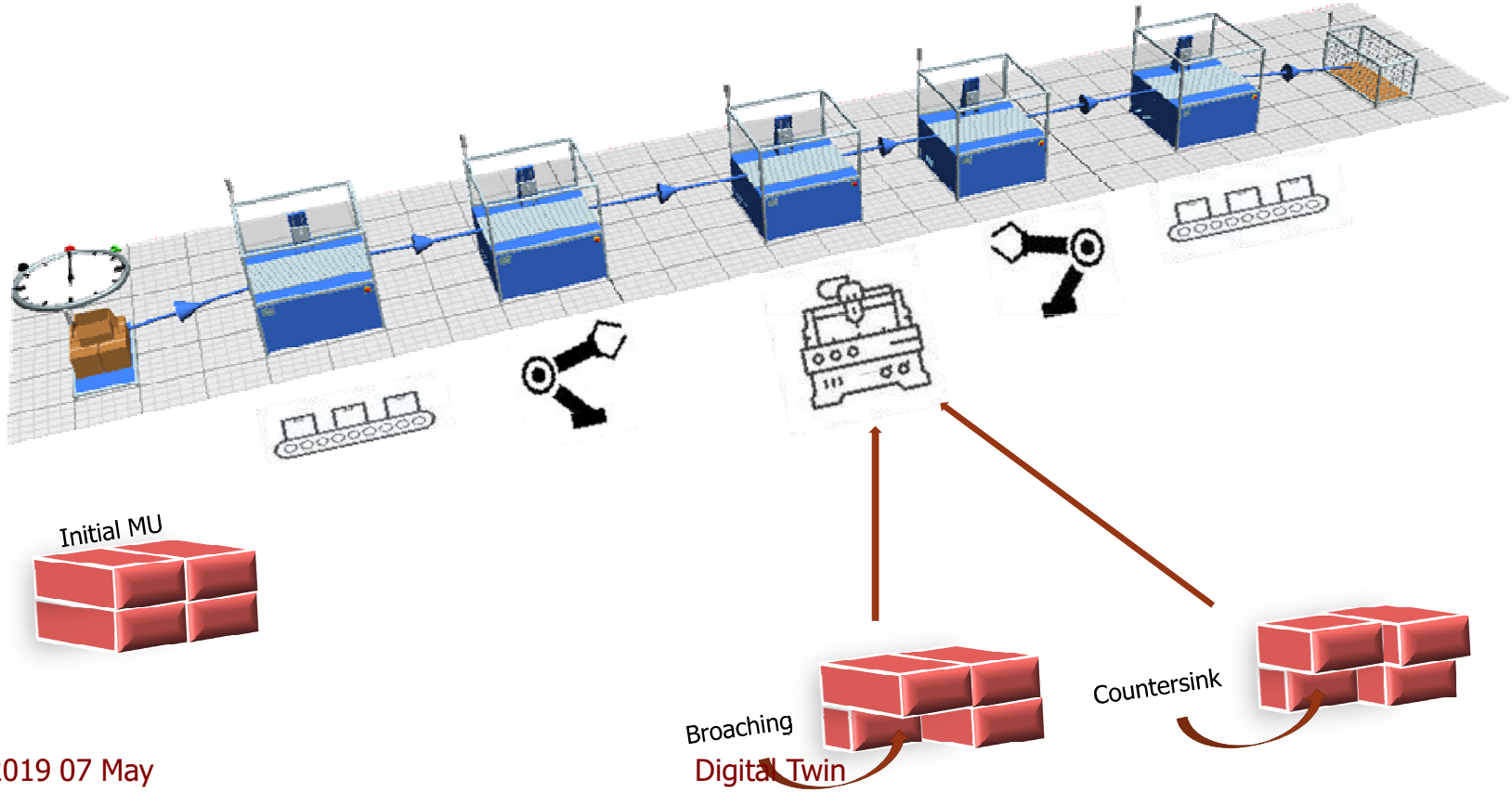
Machine library

Drilling A/G Contract	Boring A/G Contract	Reaming A/G Contract	Milling A/G Contract	Countersink A/G Contract	Broaching A/G Contract

Design Process



Simulation example of a Production Line



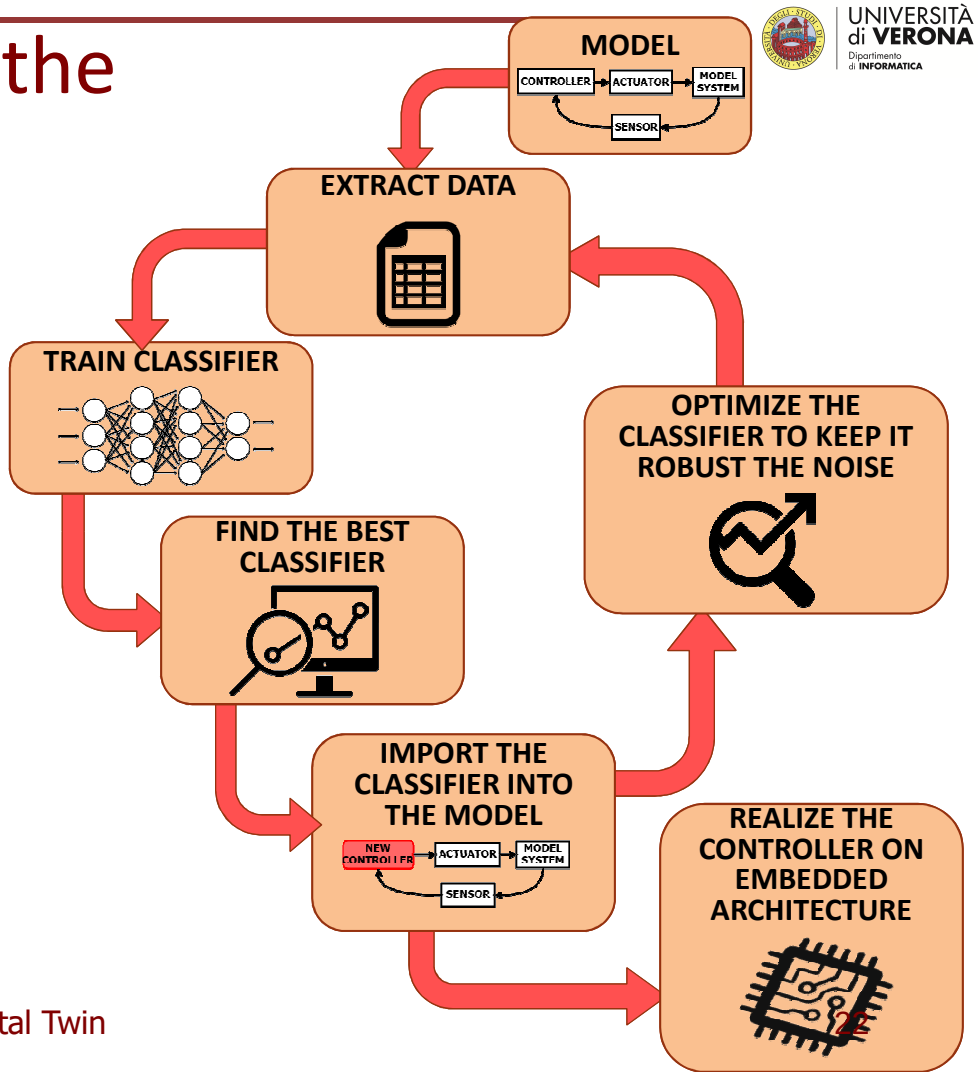
UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Alessia Bozzini – Franco Fummi

FROM TEMPORAL-SERIES TO MODELS

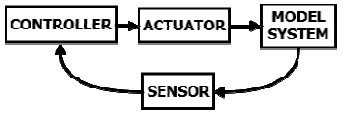
Black-box Discovery of the Control Algorithm

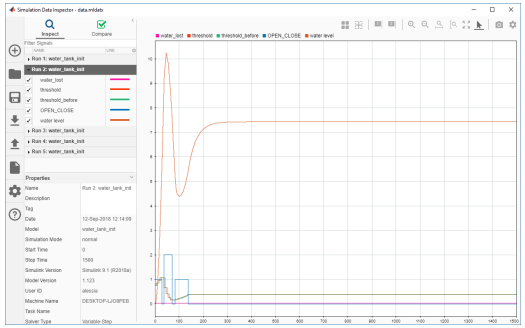
- Main steps are:
  - Export the dataset
  - Build of the trained classifier
  - Implement in system
  - Optimize the classifier
  - Generate the code



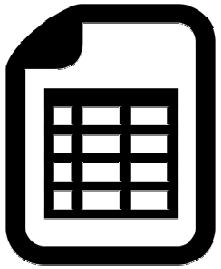
Export dataset

- Use Simulation Data Inspector
  - Observe the interested signals
  - Export and save them in the Matlab workspace
  - Export the data in a CSV-file





Simulation Data Inspector



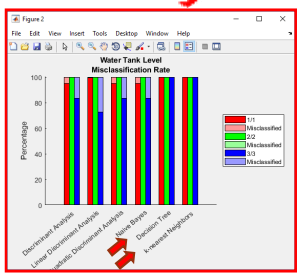
File .csv

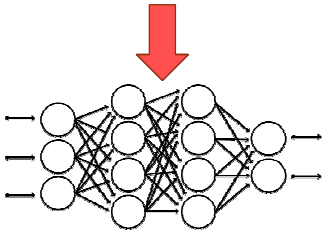
Build the trained classifier

Manual method

Statistics and Machine Learning Toolbox

- Import Dataset
- Build the various classifiers with their confusion matrix
- Compare the confusion matrices graphically
- Or calculate accuracy, precision and recall and choose the best classifier



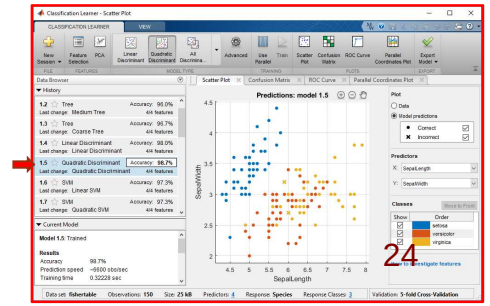


NEW CONTROLLER

Automatic method

Classification Learner App

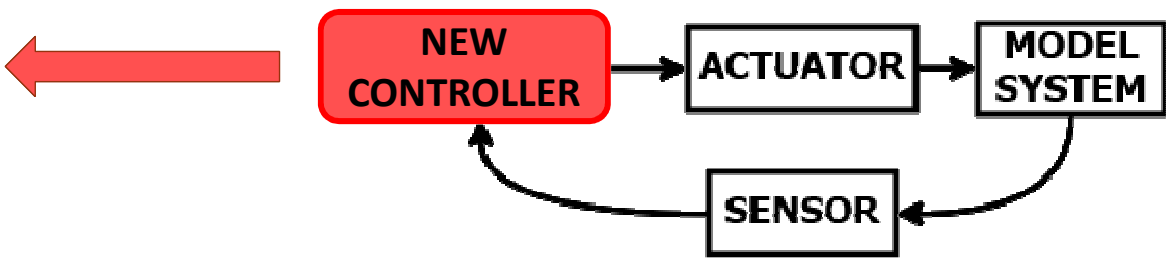
- Import dataset
- Try it on all types of classifiers
- Choose the best classifier



Implementation in the system

- Put the classifiers in
  - **M-function** blocks
  - **S-function** blocks
    - Simulating the **same result** is obtained with the M-functions
    - The S-function is **faster** than M-function

```
function OPEN_CLOSE = fcn(u1, u2, …)
persistent mdl;
if isempty(mdl)
    mdl = loadCompactModel('my_classifier.mat');
end
OPEN_CLOSE = int8(predict(mdl,[u1, u2, …]));
end
```

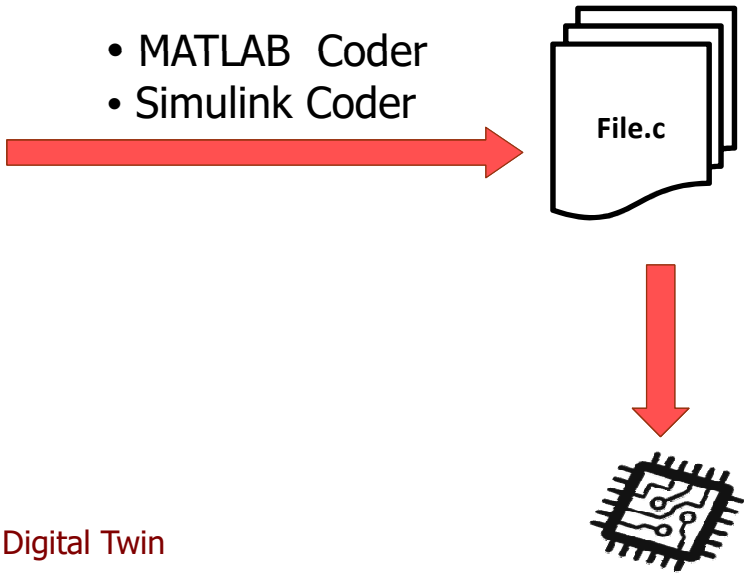


Code generation

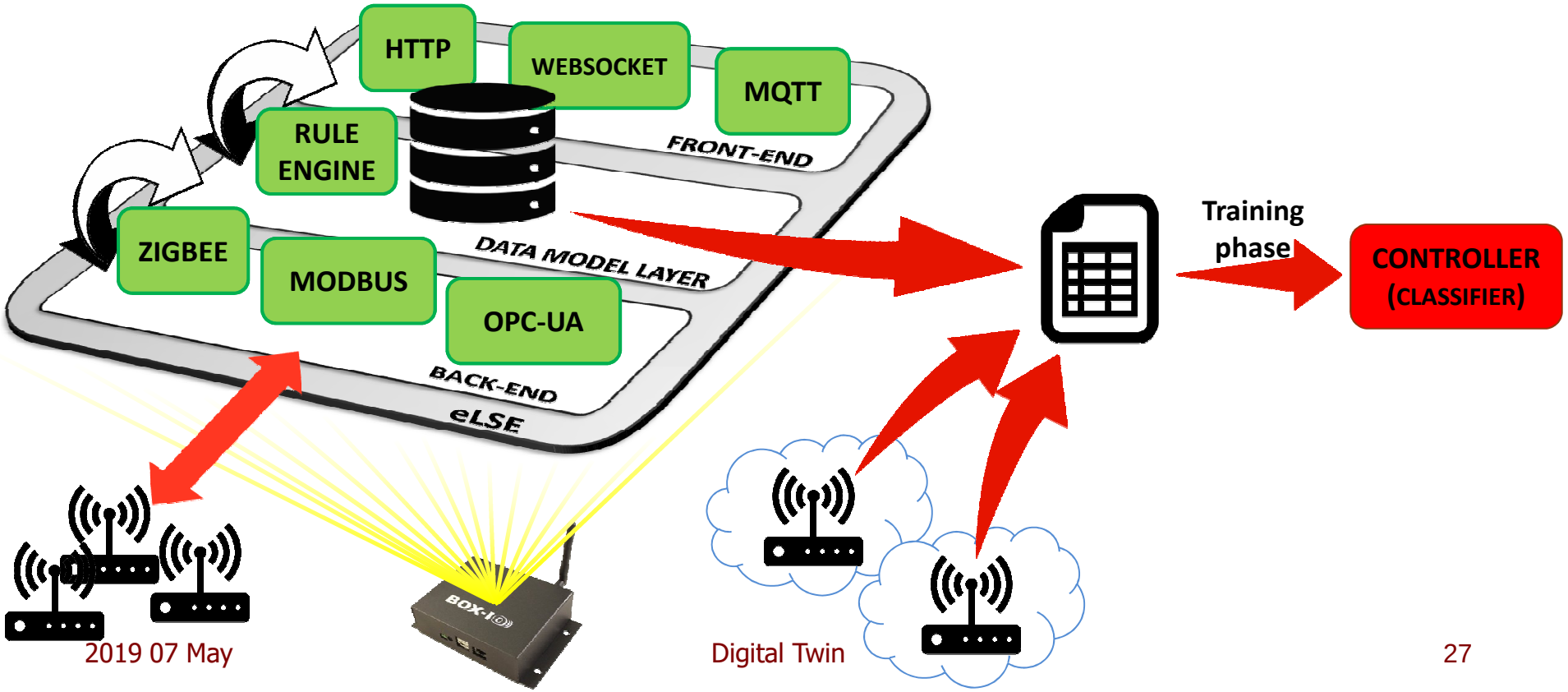
- From M-file to C-file
  - MATLAB Coder
  - Simulink Coder

```
function OPEN_CLOSE = fcn(u1, u2, …)
persistent mdl;
if isempty(mdl)
    mdl = loadCompactModel('my_classifier.mat');
end
OPEN_CLOSE = int8(predict(mdl,[u1, u2, …]));
end
```

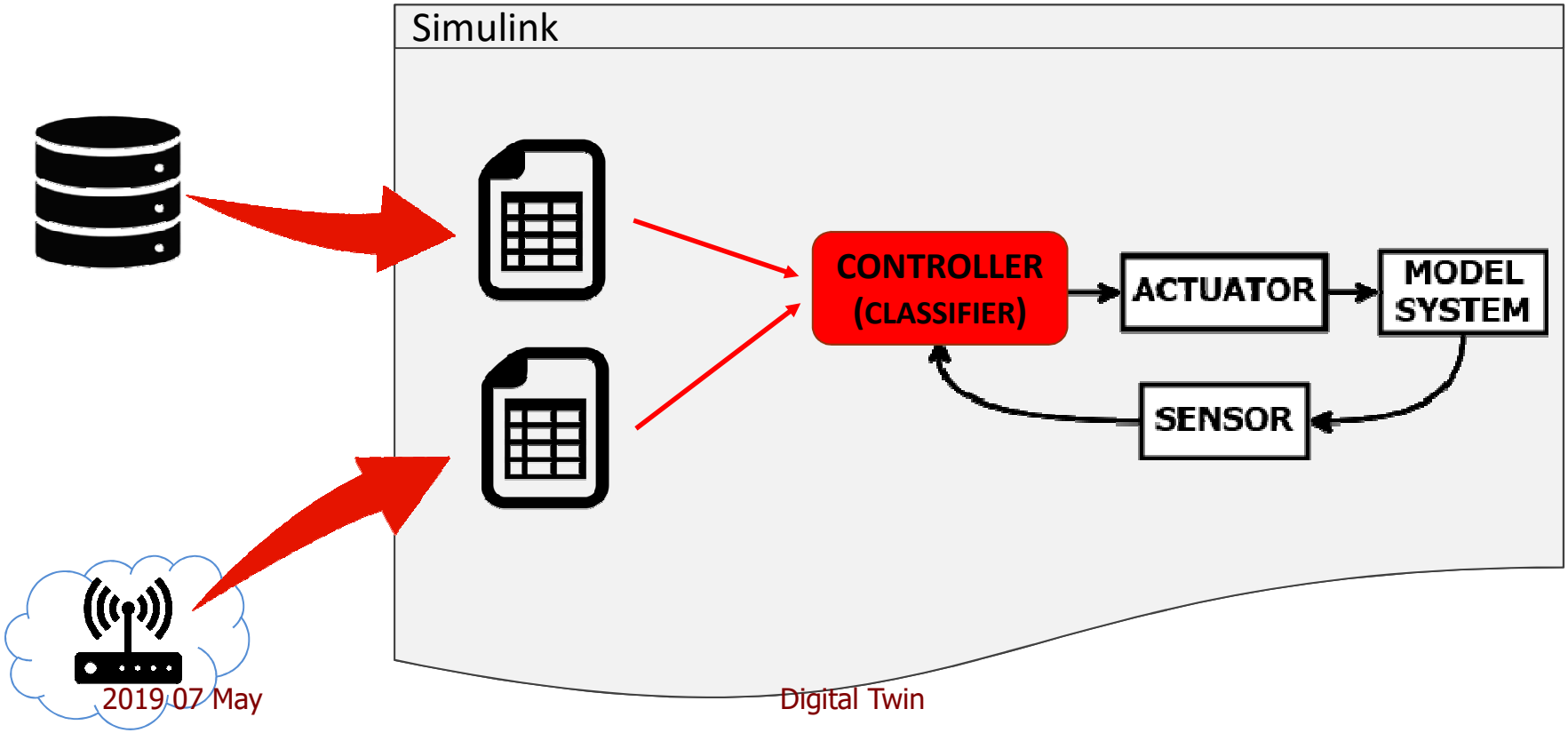
classifyX.m



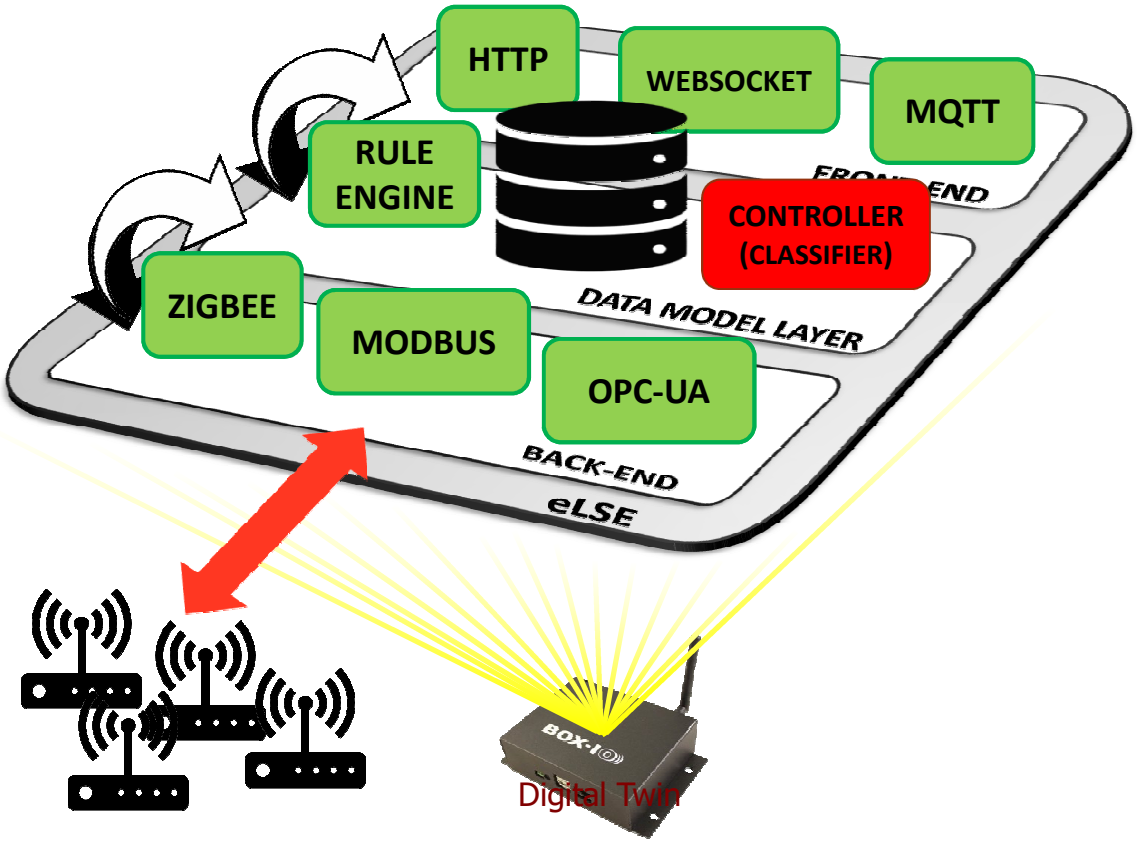
BOX-IO-based Implementation



BOX-IO


Simulink
CONTROLLER
(CLASSIFIER)
ACTUATOR
MODEL
SYSTEM
SENSOR

BOX-IO

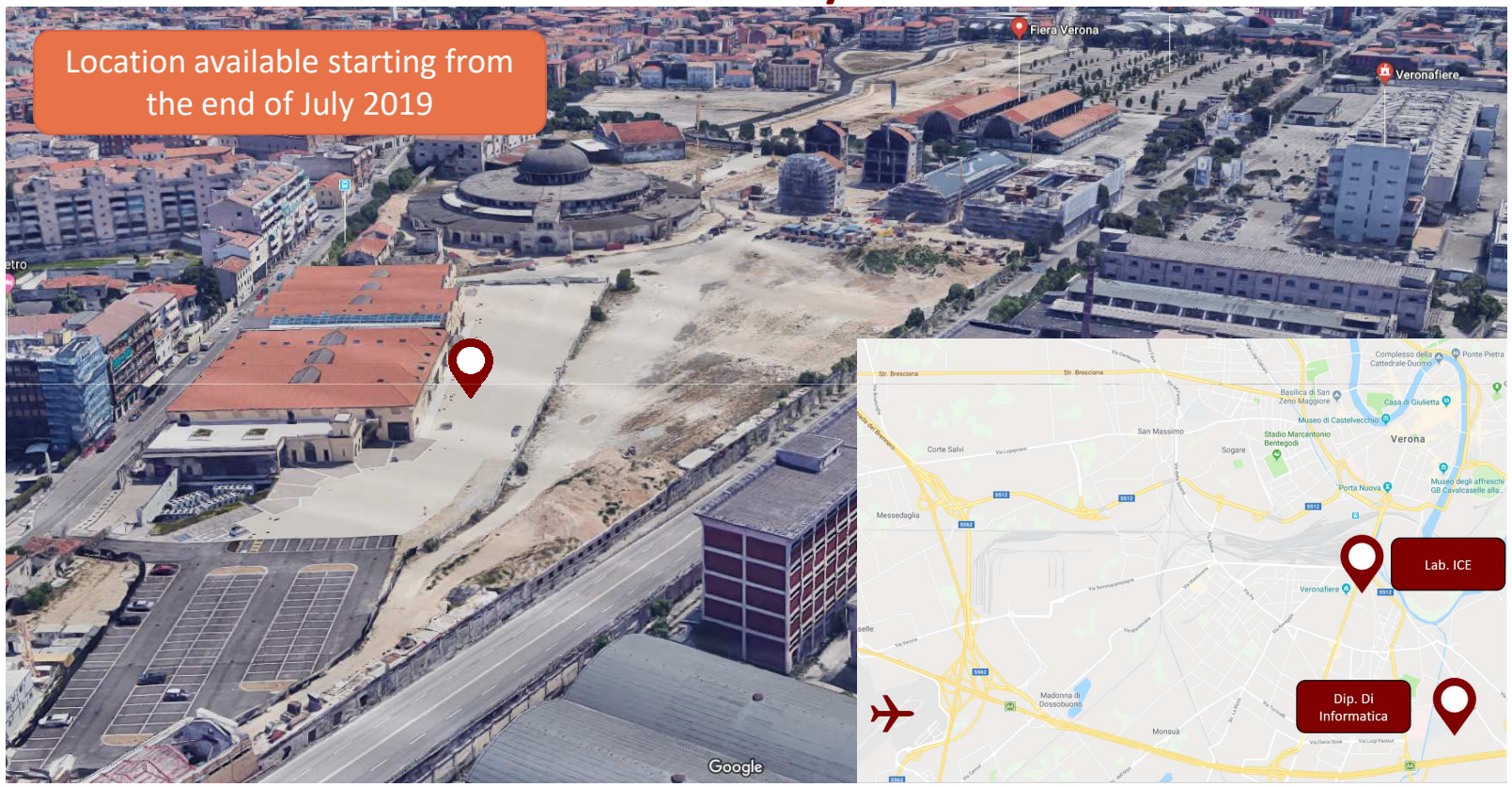

HTTP
WEBSOCKET
MQTT
RULE ENGINE
CONTROLLER (CLASSIFIER)
ZIGBEE
MODBUS
OPC-UA
DATA MODEL LAYER
BACK-END
eLSE
BOX-I

UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

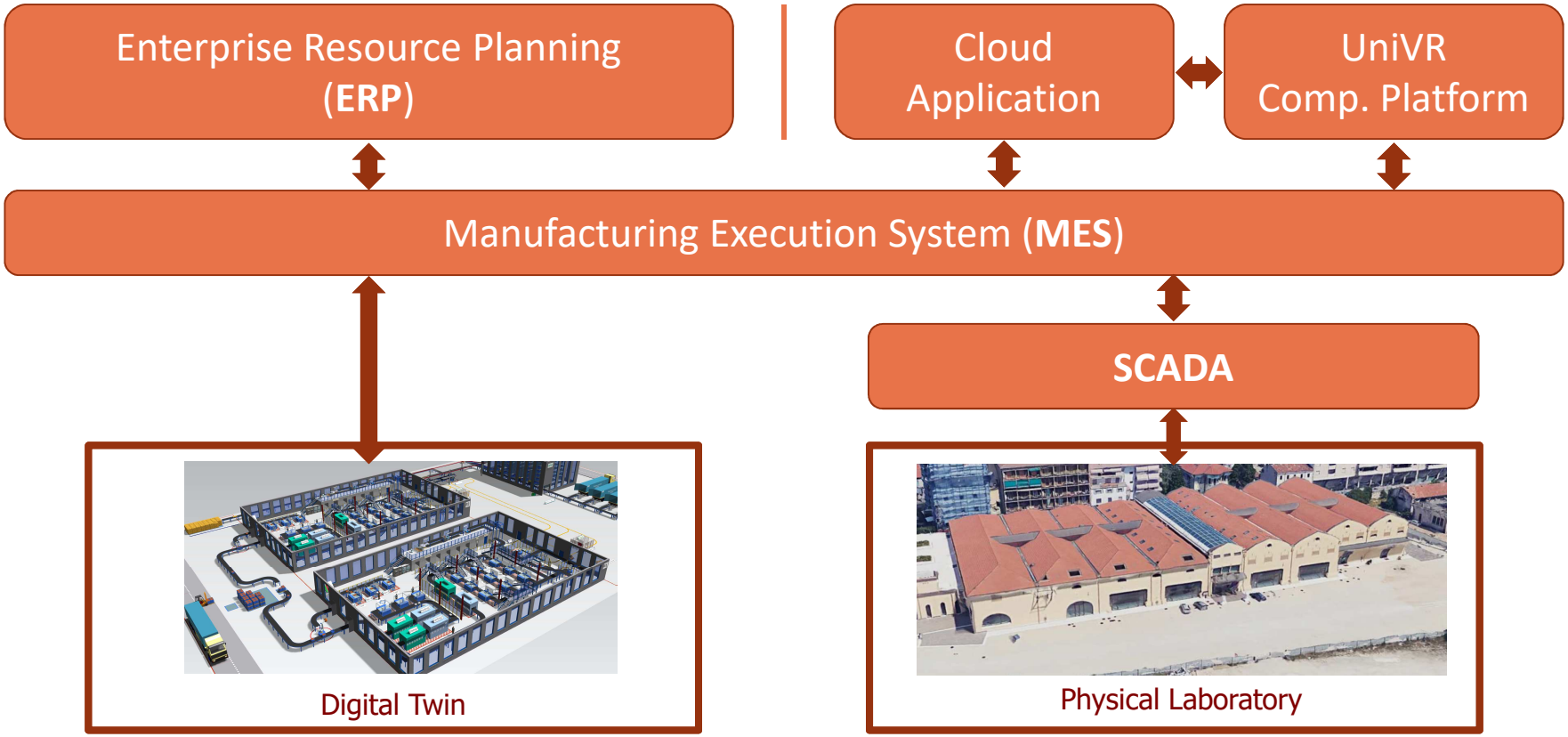
Marco Panato – Carlo Tadiello – Franco Fummi

EXEMPLIFICATION ON THE ICE LABORATORY

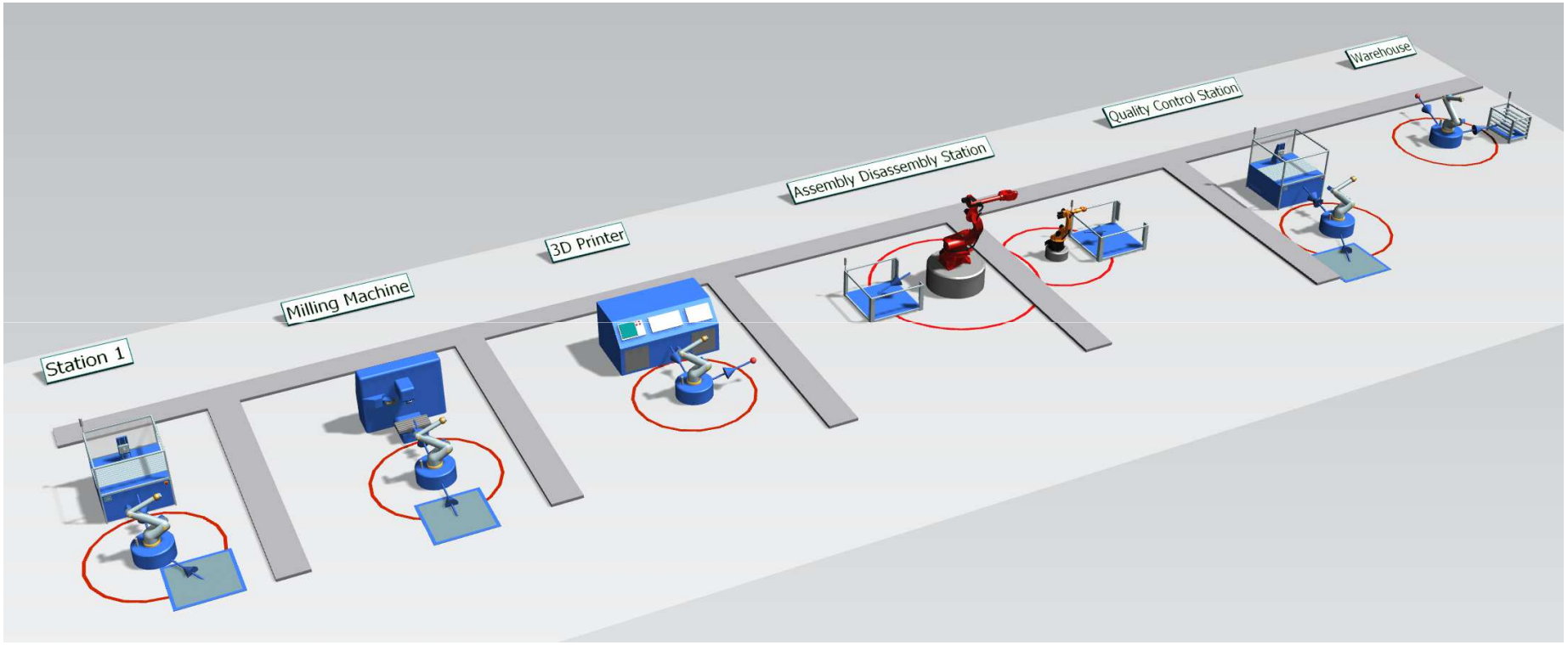
ICE Laboratory - Where



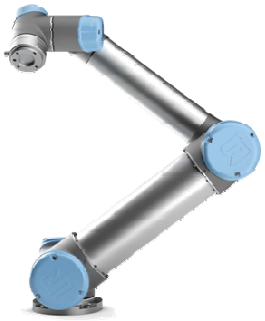
ICE Laboratory - System Architecture



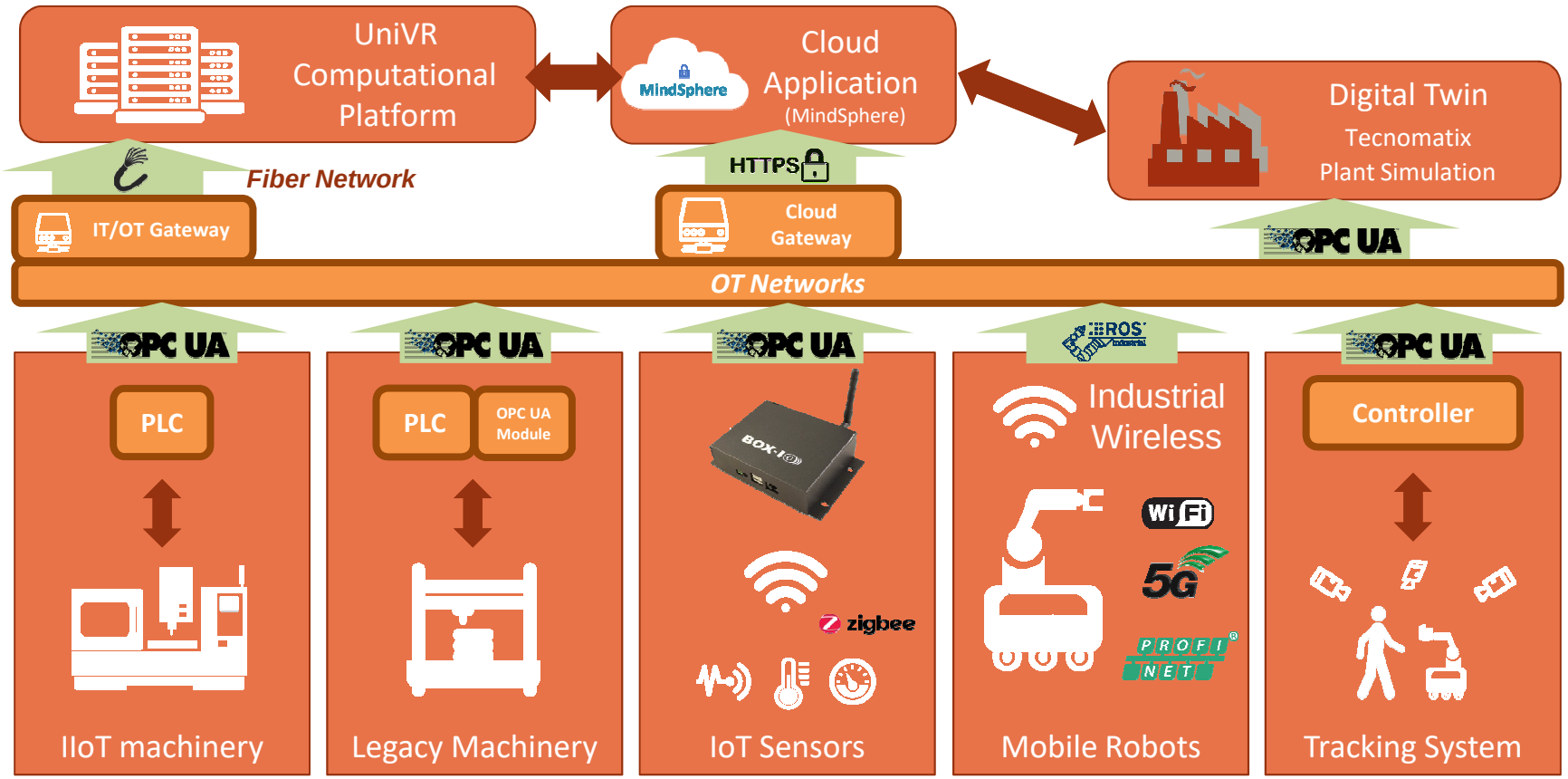
ICE Laboratory – Digital Twin Focus



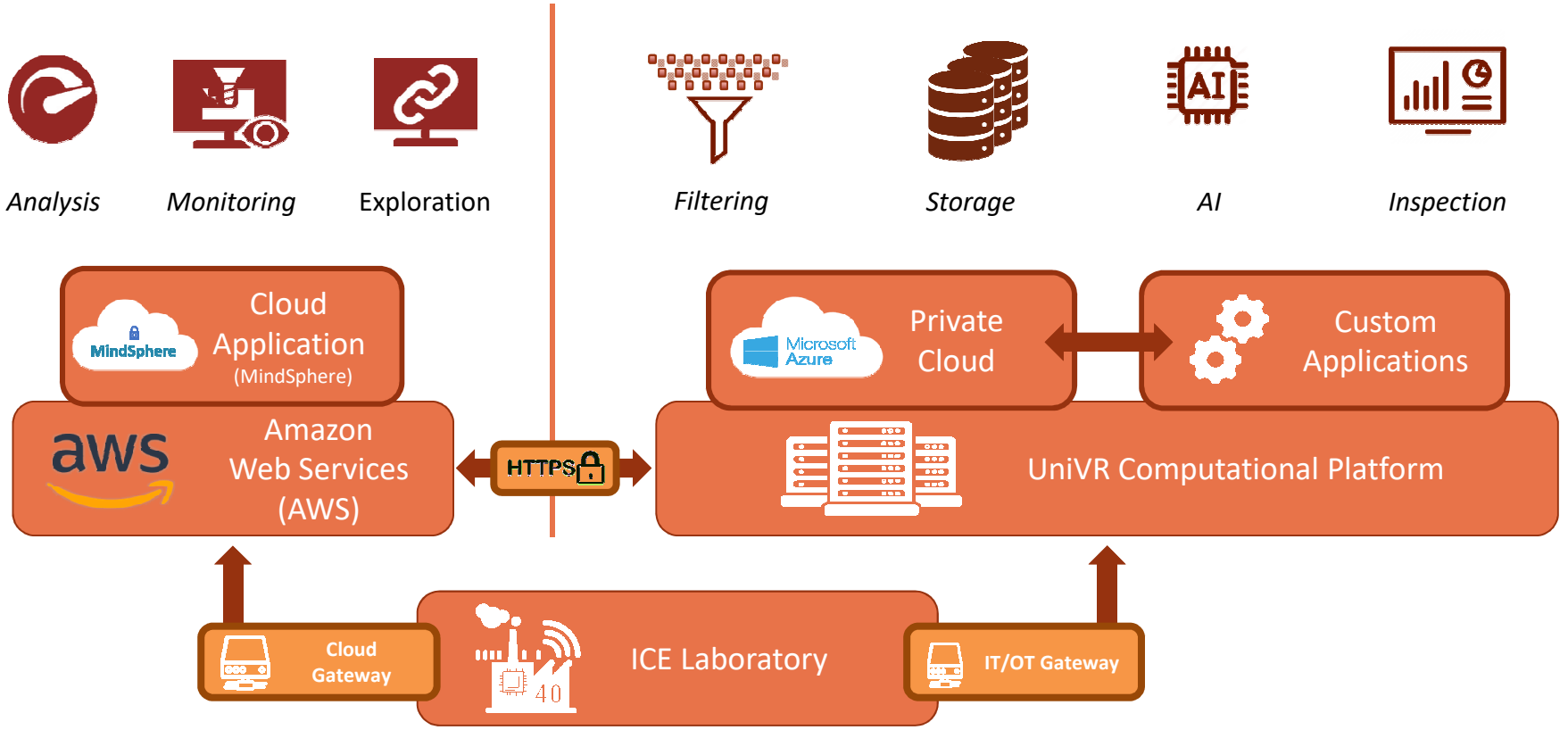
ICE Laboratory – Robots and Devices



ICE Laboratory – Sensors/Cloud Architecture



ICE Laboratory - Cloud Architecture



UniVR Computational Platform

Composed of two kind of nodes managed by ***OpenStack***

- *HPC node*
  - *CentOS 7* available to end-user
  - *Slurm* workload manager
    - *launch/schedule* tasks with specific HW resources
  - *Not supported* by all applications
- *Cloud node*
  - dedicated to host *Virtual Machines*
  - a running VM meets *loose real time* requirements
  - less *processing power* if compared with *HPC*

Industrial Advisory Board (IAB) - Opportunities

- A group composed of more than *35* companies
- Established for *Computer Engineering for Industry 4.0* project

- Companies are *actively participating* in our project by
  - suggesting *ICE lab components*
  - *giving opinions* over the new Master's Degree
  - developing new *teaching modules* to train students, employees, customers
  - getting the *annual research reports* over Industry 4.0 technologies
  - *testing new technologies* on the ICE lab after its construction

Industrial Advisory Board (IAB) - Composition

- IAB contains companies of different categories:
  - Industrial Automation
  - Software House
  - Buildings
  - Media and Communications
  - Engineering
  - System Integrator
  - Automation
  - IT consulting
  - Manufacturing
  - Food and Beverage
  - IT Hardware