Moving towards AI application for optimal sequential decision making

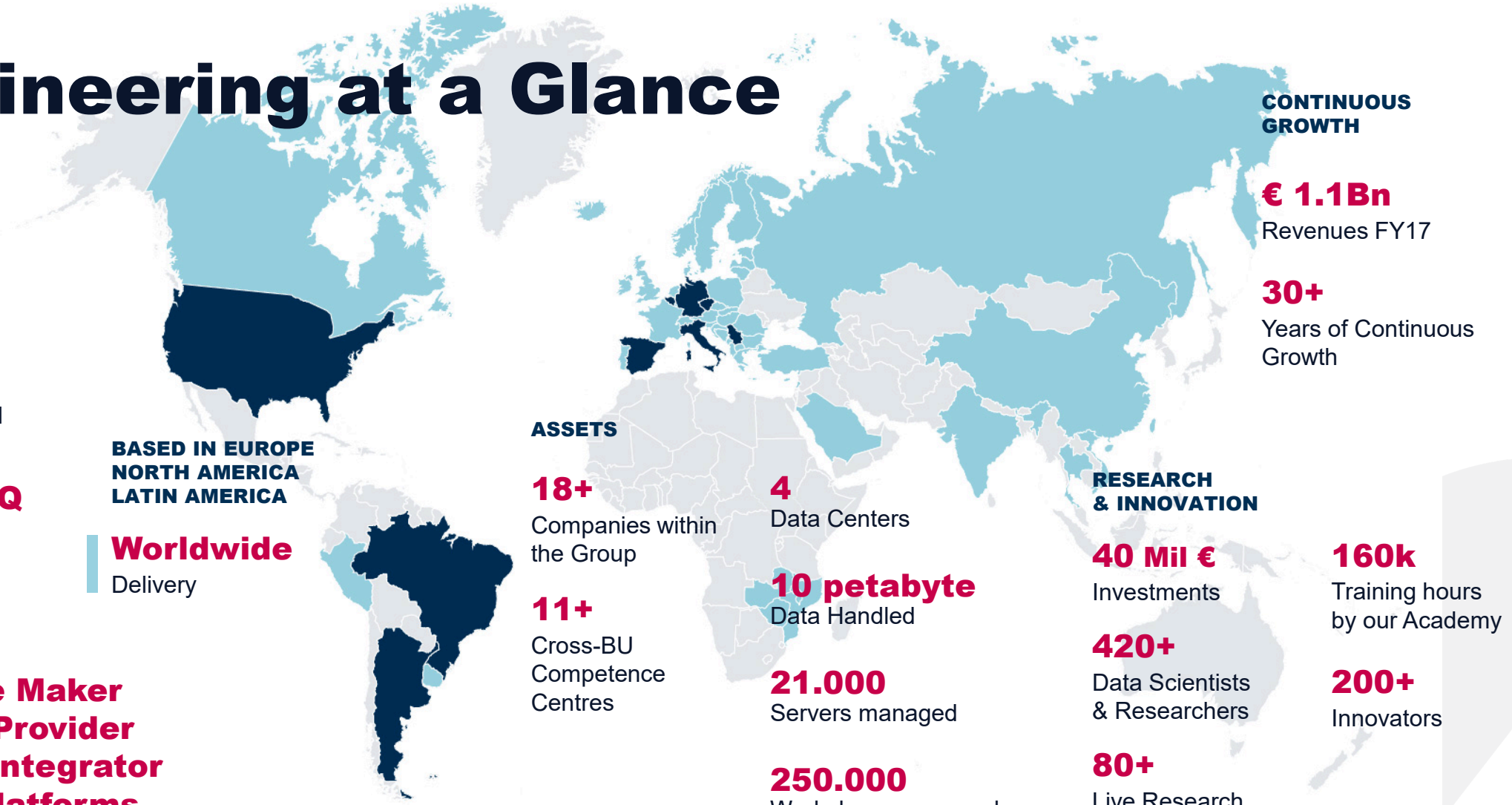
AnyLogic Conference

Austin, Texas

April 18th 2019

ENGINEERING

Engineering at a Glance



A GLOBAL COMPANY

11.000+
Associates

50+
Offices around the world

Global HQ
Rome, Italy

BASED IN EUROPE NORTH AMERICA LATIN AMERICA

Worldwide
Delivery

WHAT WE DO

Software Maker
Service Provider
System Integrator
Digital Platforms

ASSETS

18+
Companies within the Group

11+
Cross-BU Competence Centres

4
Data Centers

10 petabyte
Data Handled

21.000
Servers managed

250.000
Workplaces managed

CONTINUOUS GROWTH

€ 1.1Bn
Revenues FY17

30+
Years of Continuous Growth

RESEARCH & INNOVATION

40 Mil €
Investments

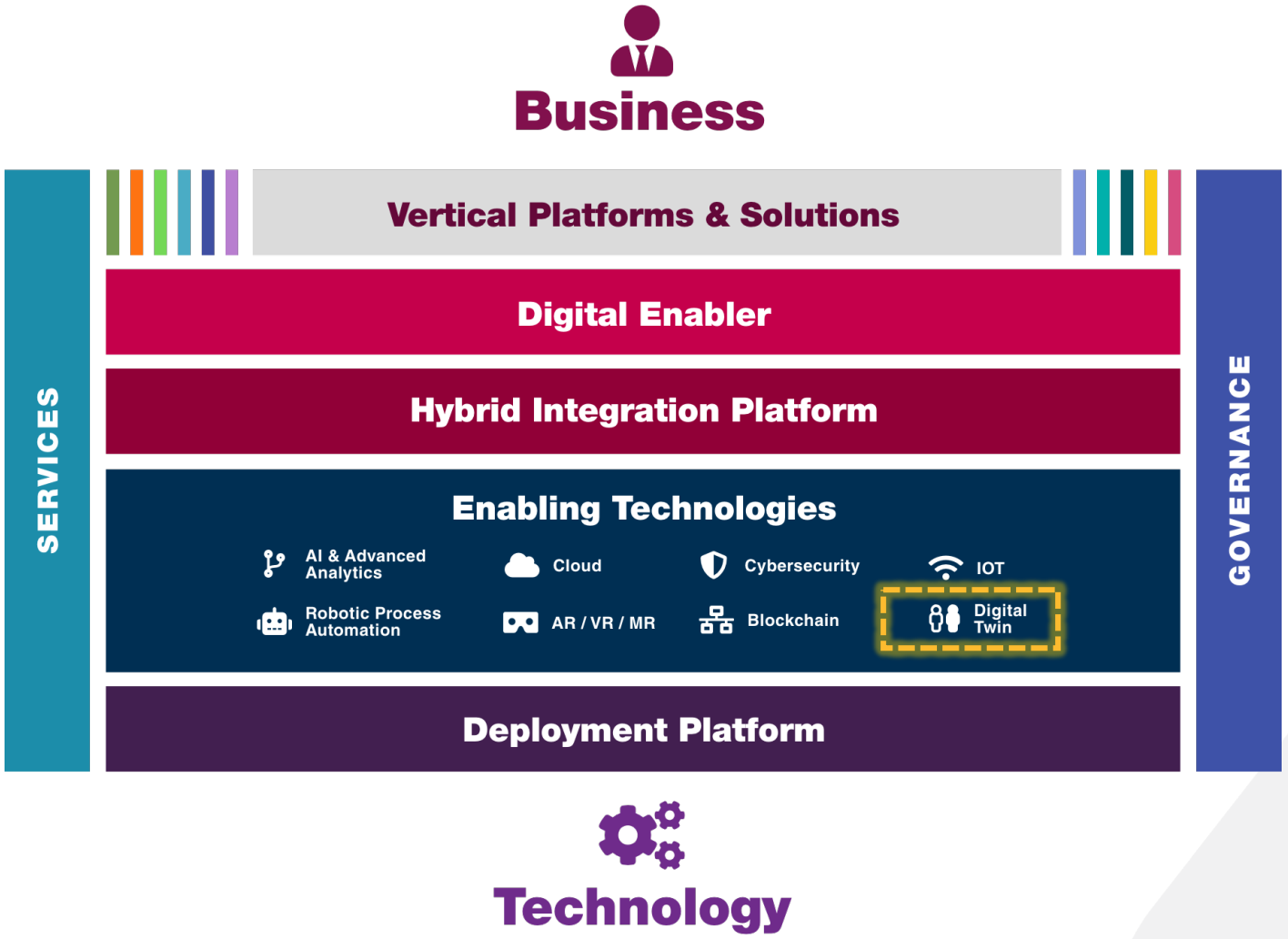
420+
Data Scientists & Researchers

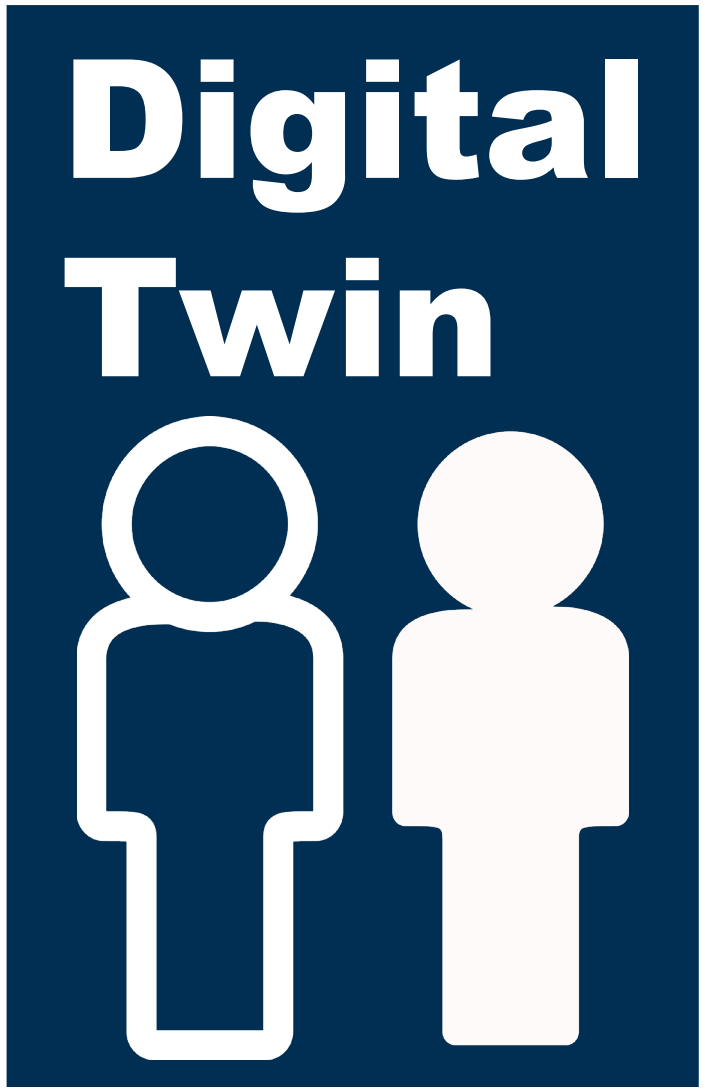
80+
Live Research Projects

160k
Training hours by our Academy

200+
Innovators

Our Technology & Platform Stack





Our team is focused on:

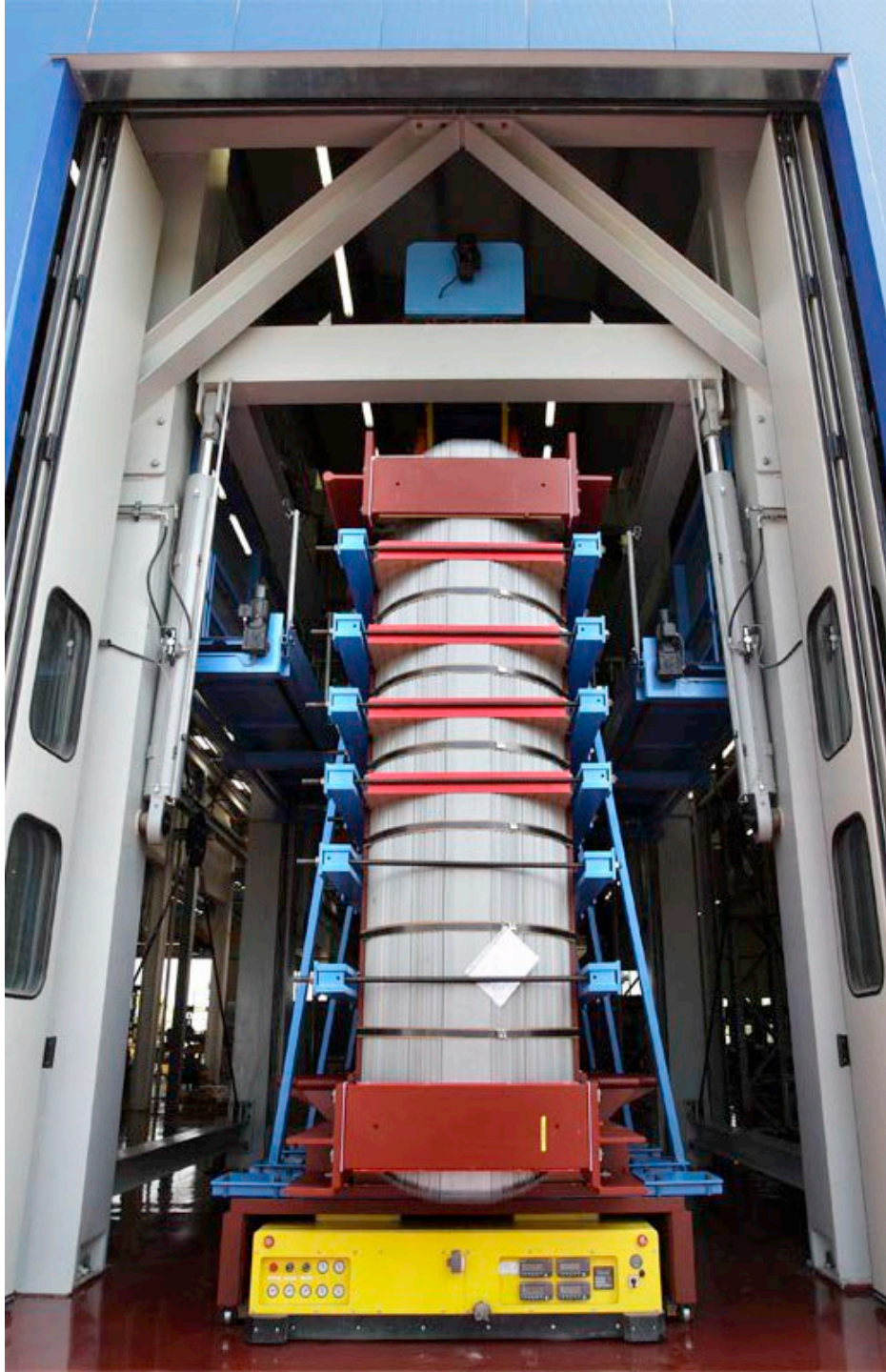
- **Digital-Twin Application** → helping customers to take informed decisions through a digital copy of their business - this masters uncertainty and maximizes chances at success;

- **Forecasting and Optimization** → supporting customers in forecasting and optimizing business performance through the use of Simulation Modelling, Optimization Techniques, Machine Learning and A.I.

The Client: Lagor

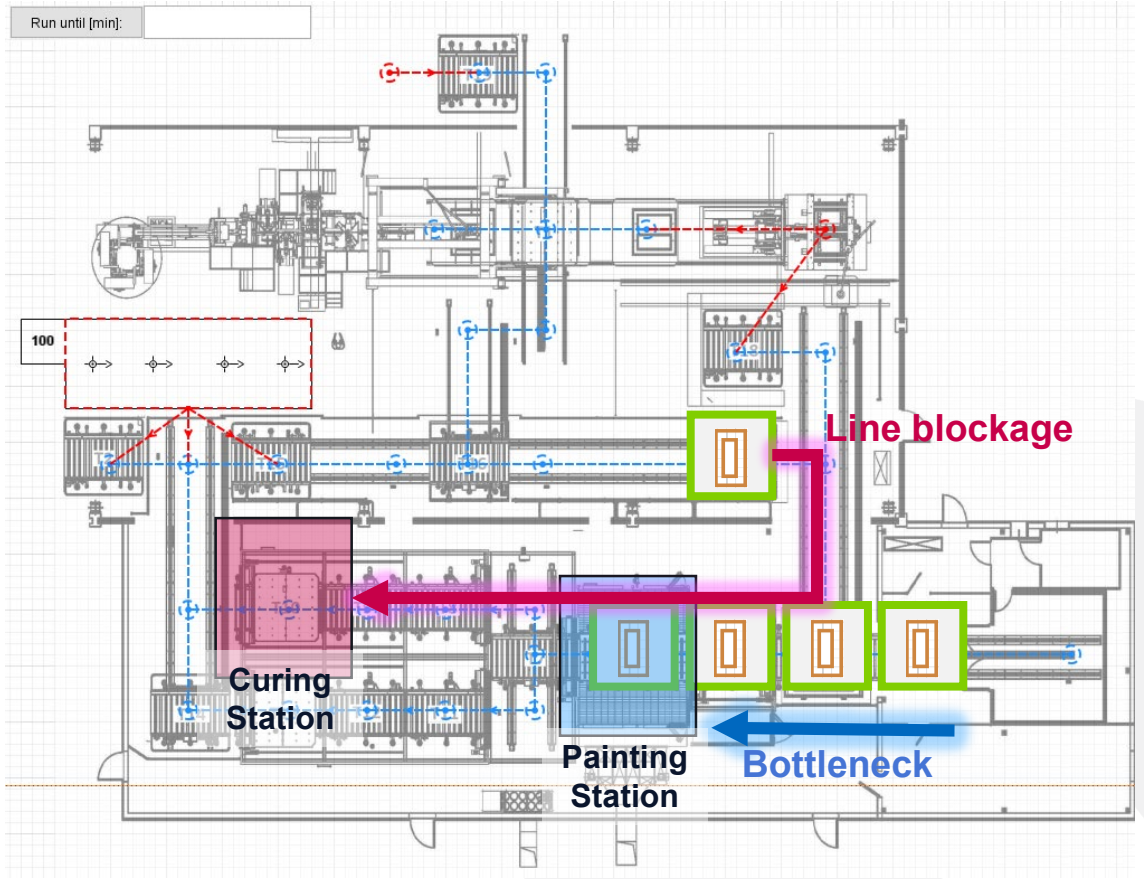
- Lagor is an Italian manufacturer of ferromagnetic cores for distribution and power transformers (PT Cores)
- PT Cores are highly customized to meet the specific requirements of each transformer

LAGOR
PT CORES
Guaranteed performance by Pass-Line



Problem description

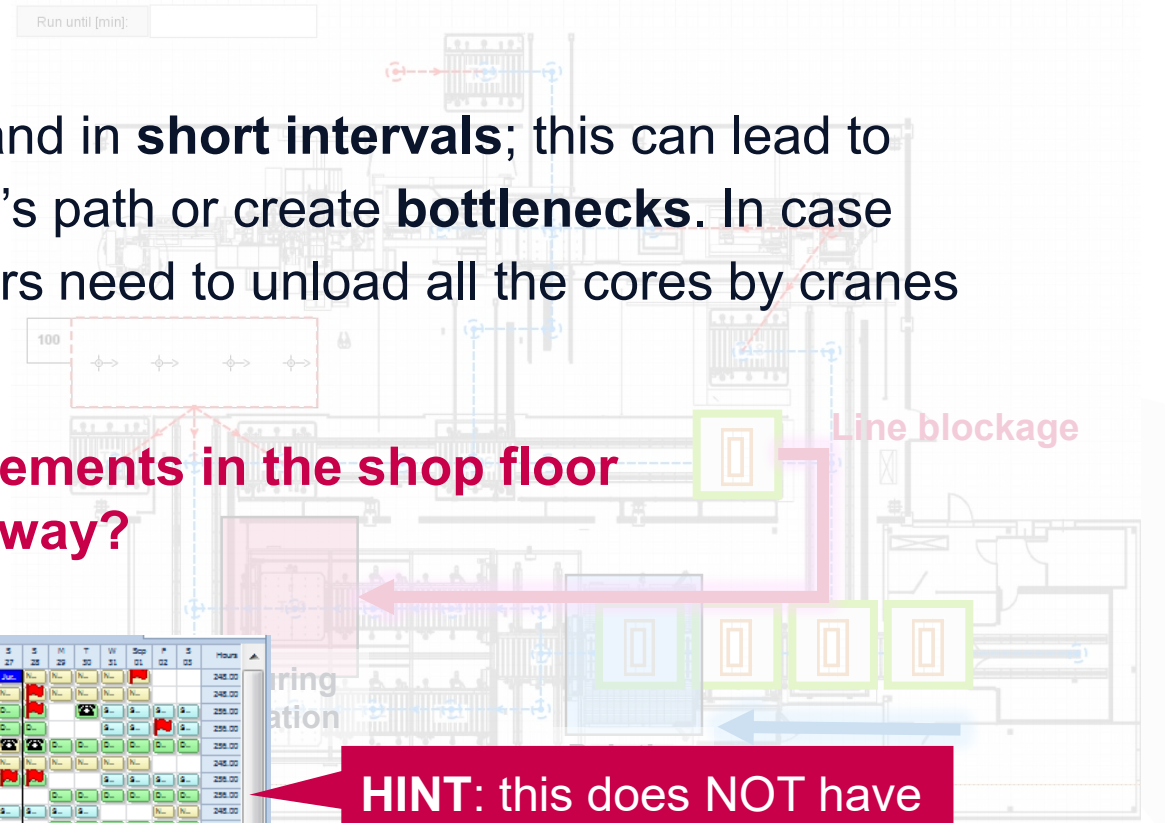
- PT Cores are essentially built from many layers of coils which can weigh up to **8 tons** and require **different production cycles** depending on their size and client-specific customizations.
- Since the very first operation, the cores are loaded on steel pallets which are moved across the different work stations using roller or shuttle conveyors.
- At any given time, **several different cores** can be found on the line.
- Due to the above features, sequencing the operations in the production line can be critical to avoid **bottlenecks** and **line blockage.**



Challenges

Currently, the line planning is done **manually** and in **short intervals**; this can lead to situations in which the objects **block** one other's path or create **bottlenecks**. In case the core flows become **too knotty,** the operators need to unload all the cores by cranes and reset the line.

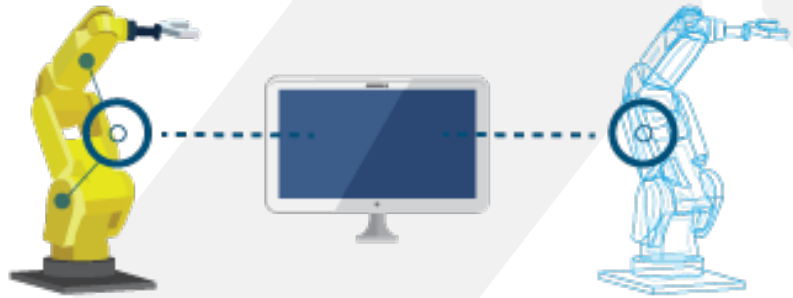
How to plan and manage the movements in the shop floor in an efficient way?



HINT: this does NOT have to be the only answer

Digital Twin: Simulation

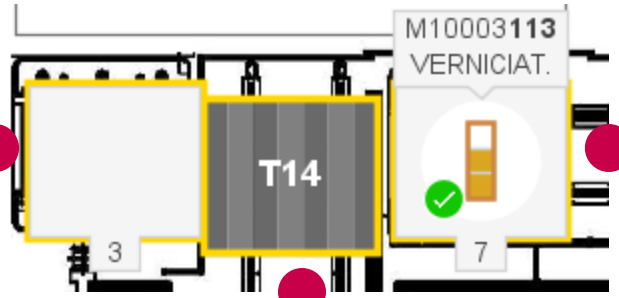
- Create a **digital replica** of the shop floor
- Use an **agent-based modeling** approach to consider all the details, such as different PT core types (and related production cycles) and variable production plan
- Implement the **logic** and **heuristics** that must govern the movement of different objects
- Connect to the SCADA system to get an updated line status
- Allow the line manager to **simulate** the whole production plan and verify that the selected plan is achievable



Digital Twin: Main Elements Reproduced

Steel pallets

Critical resources (**finite capacity**) which PT cores must be placed on to move along the production line. Pallets must never be unloaded from the line and they must be **right-size** to optimize the operations.



PT Core

Moved along the different work stations of the line according to the specific production cycle.

Conveyors

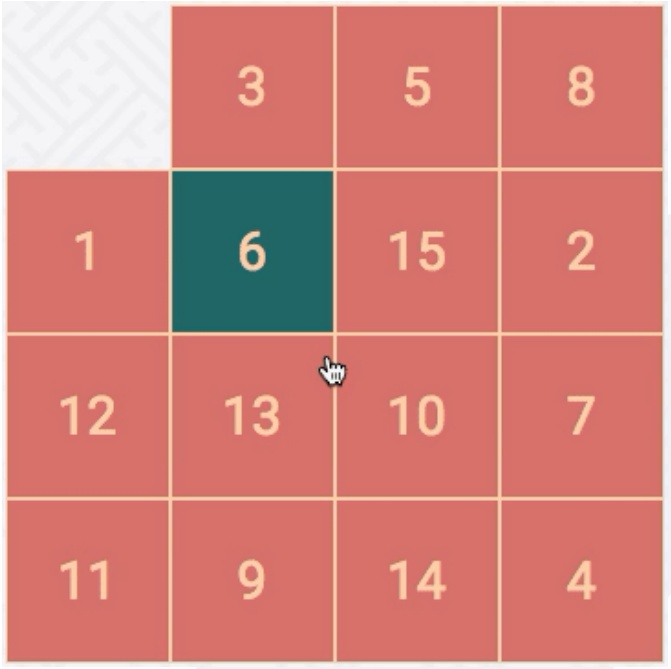
Pallet-carrying component; can be static (rolling) or moving (shuttle).

Digital Twin: Main Elements Reproduced

Line Manager

- A virtual agent in charge of designing, assessing and executing the handling of the line.
- Its embedded algorithm is conceived to reproduce the decision-making process (heuristics-based) of the human production manager:
  - ❑ Evaluate the “best” route for each situation in order to avoid unnecessary movements, anticipate possible criticalities, resolve conflicts (both for Cores and Steel Pallets), and to ultimately respect delivery dates.

LINE MANAGER AT WORK

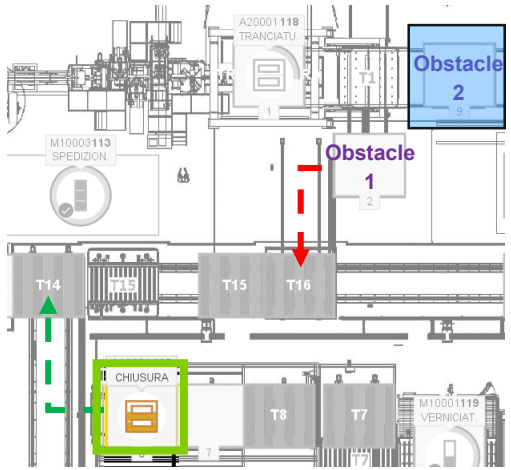


trivializing…

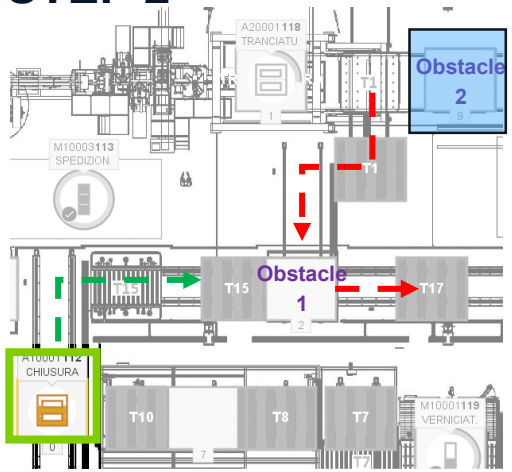
Simulation showcase

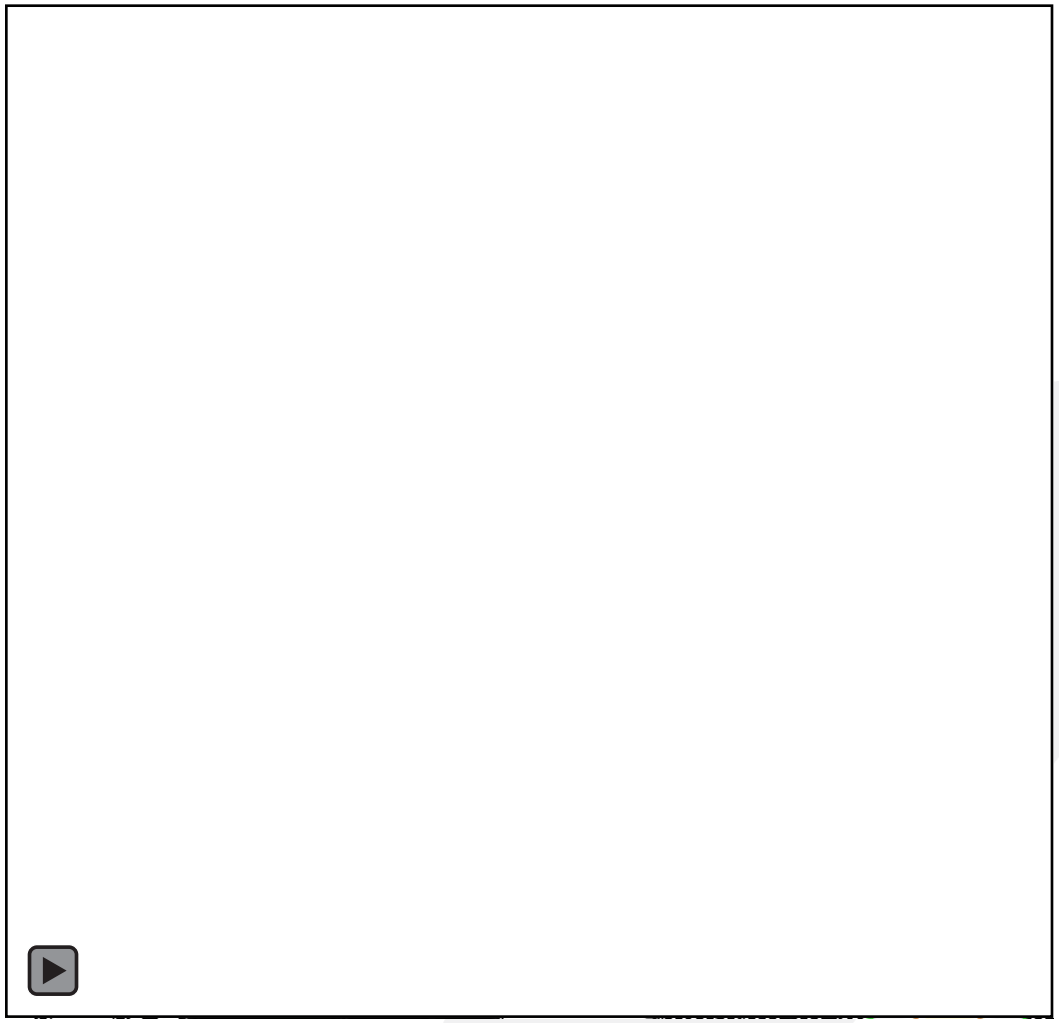
- Each Transformer Core  is moved towards its target position
- All **obstacles** must be removed

STEP 1



STEP 2





Simulation results

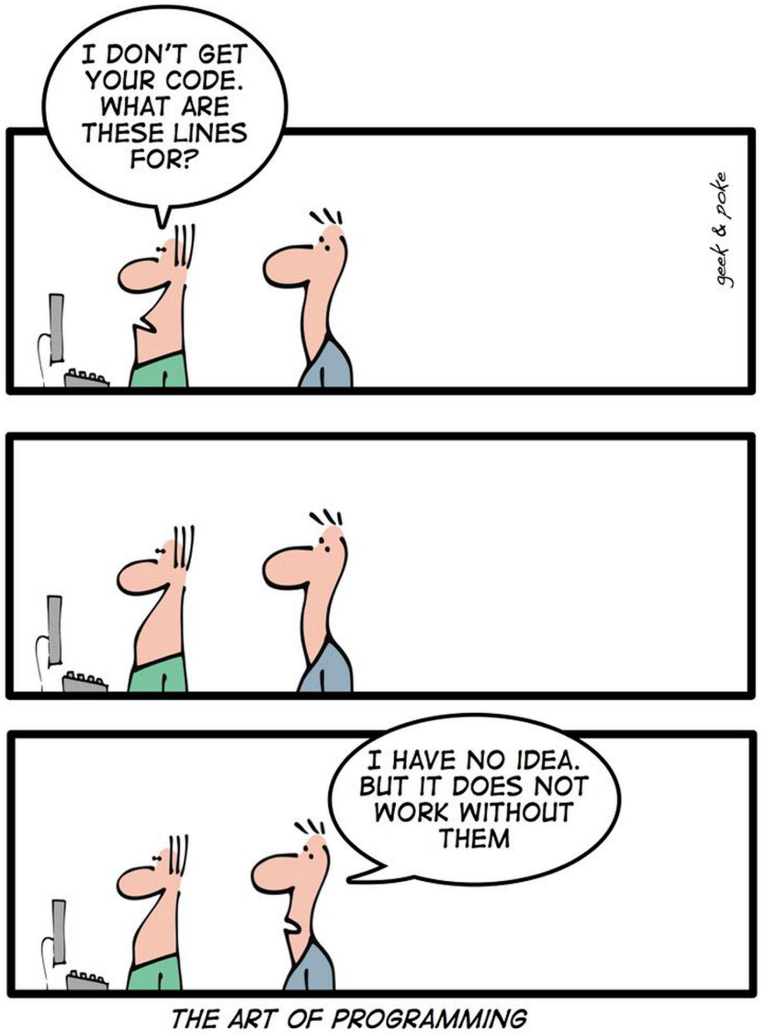
We created a simulation model of the shop floor:

- **Complex heuristics** for the system behavior are devised
- Line manager can clearly **investigate** the production plan
- Sequences can be rearranged through a **what-if** approach
- Lagor can **anticipate** possible issues, so as to work around or through them and therefore **reduce its operating costs**.

Simulation limits

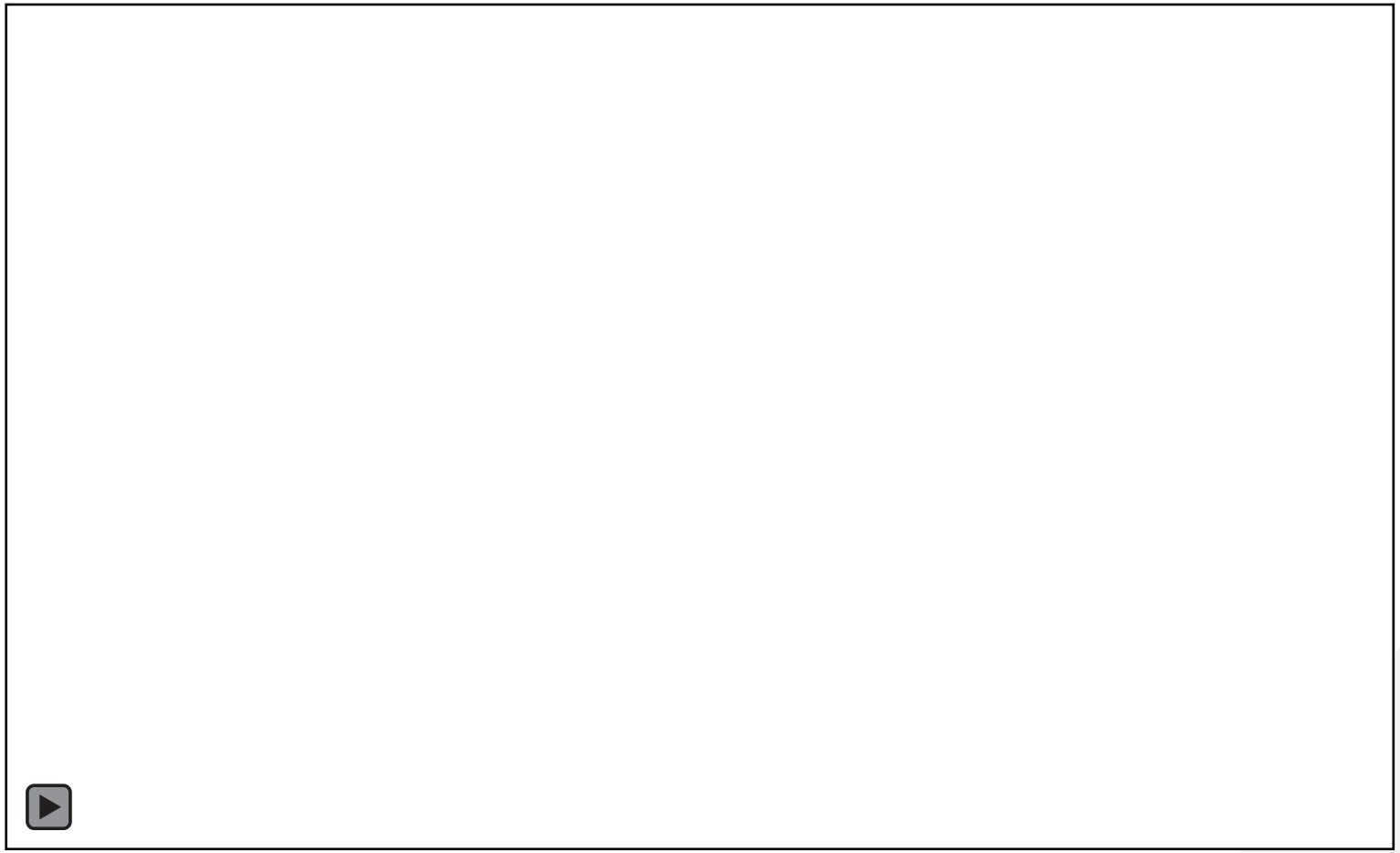
The sequence planning must be done by using many complicated **heuristics.**

No matter how profoundly engineered these heuristics are, there are still cases that end up with undesired and unexplainable results.



https://www.hackerearth.com/csec-heuristics-0

Simulation limits



Alternative?

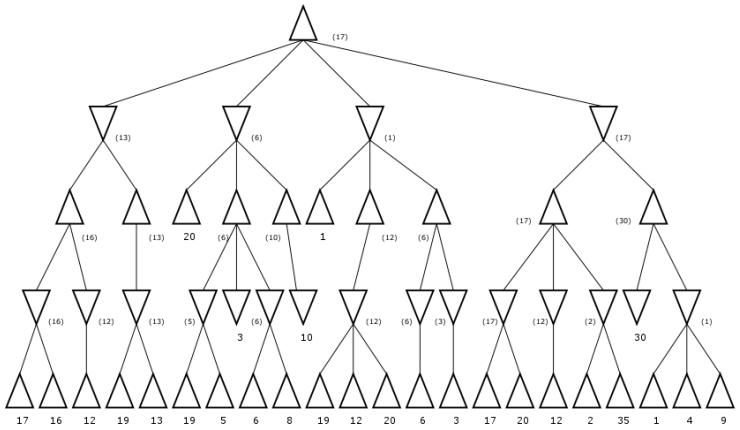
Can we get rid of complicated heuristics?

What is the “best” movement sequence?

Indeed, by using heuristics, we are trying to avoid a set of sequences in which we end up losing (i.e. line blockage), **is that possible?**

One way out is to explore all possible cases that can emerge in all future decision steps (i.e. brute force search tree)

Geri's game, by PIXAR, 1997



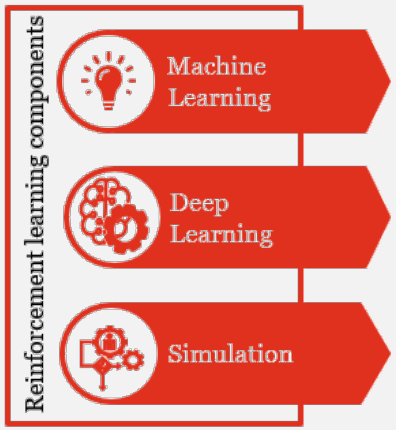
IBM Deep Blue for chess,
It explores only between 6 and 8 moves ahead

Another Alternative

Can we get rid of complicated heuristics?

What is the “best” movement sequence?

Can we use **AI** to solve such decision-making problem?



Using algorithms to learn from data and solve business problems without being explicitly programmed

Leveraging cutting-edge machine learning algorithms inspired by artificial neural networks especially for unstructured data

Developing models of real-world processes and testing their performance/success under various scenarios

PwC Artificial Intelligence Accelerator, AL conf. 2018

Why should we enter this game?

The agent learns via interactions with the environment. Therefore, a reliable **environment** is indispensable for training the agent.

How to make an environment

Real physical world

Dangerous and costly

Simulation modeling

Extensive records

Limited and costly

Simulation and modeling are the best candidate to facilitate the application of the reinforcement learning and our expertise lies in this field

What are the challenges?

Formulate decision sequence, either Discrete or continuous

Appropriate RL algorithm

High-quality environment (simulator) representing correct casual dynamics in the system. How to create one?

Reward function, are we spoon feeding or giving no clue?

Observation representativeness: formulation and transparency

Seamless Integration of the two worlds

The learning process is partially opaque, and the final learnt policy is a black box

How to integrate RL and Anylogic

Reinforcement algorithm

DL4J

PyTorch

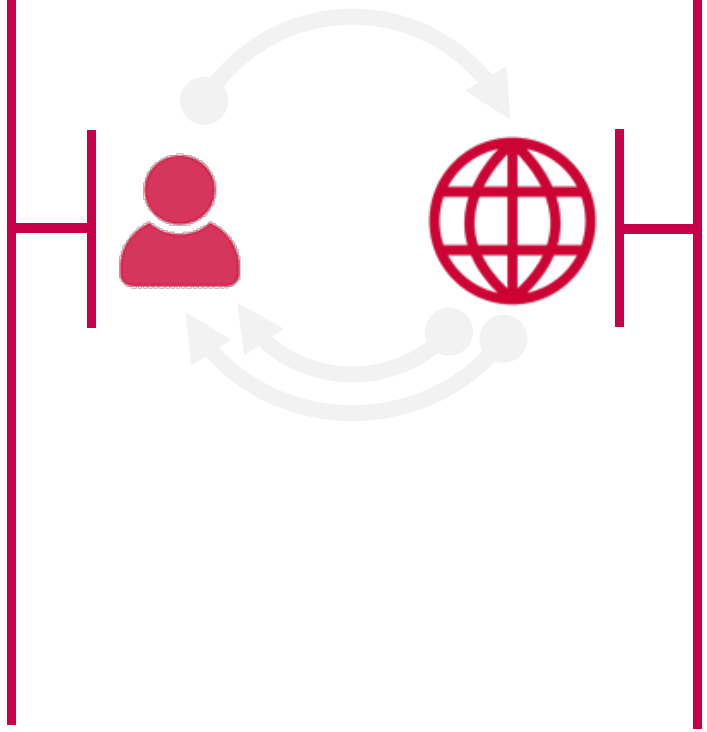
Apache Spark

Apache kafka
A distributed streaming platform

Apache hadoop

Ramin Mirzazadeh

Cristina Morariu



Environment

anylogic®

Leading simulation software for business applications

anyLogistix

Supply chain design, analytics and optimization software

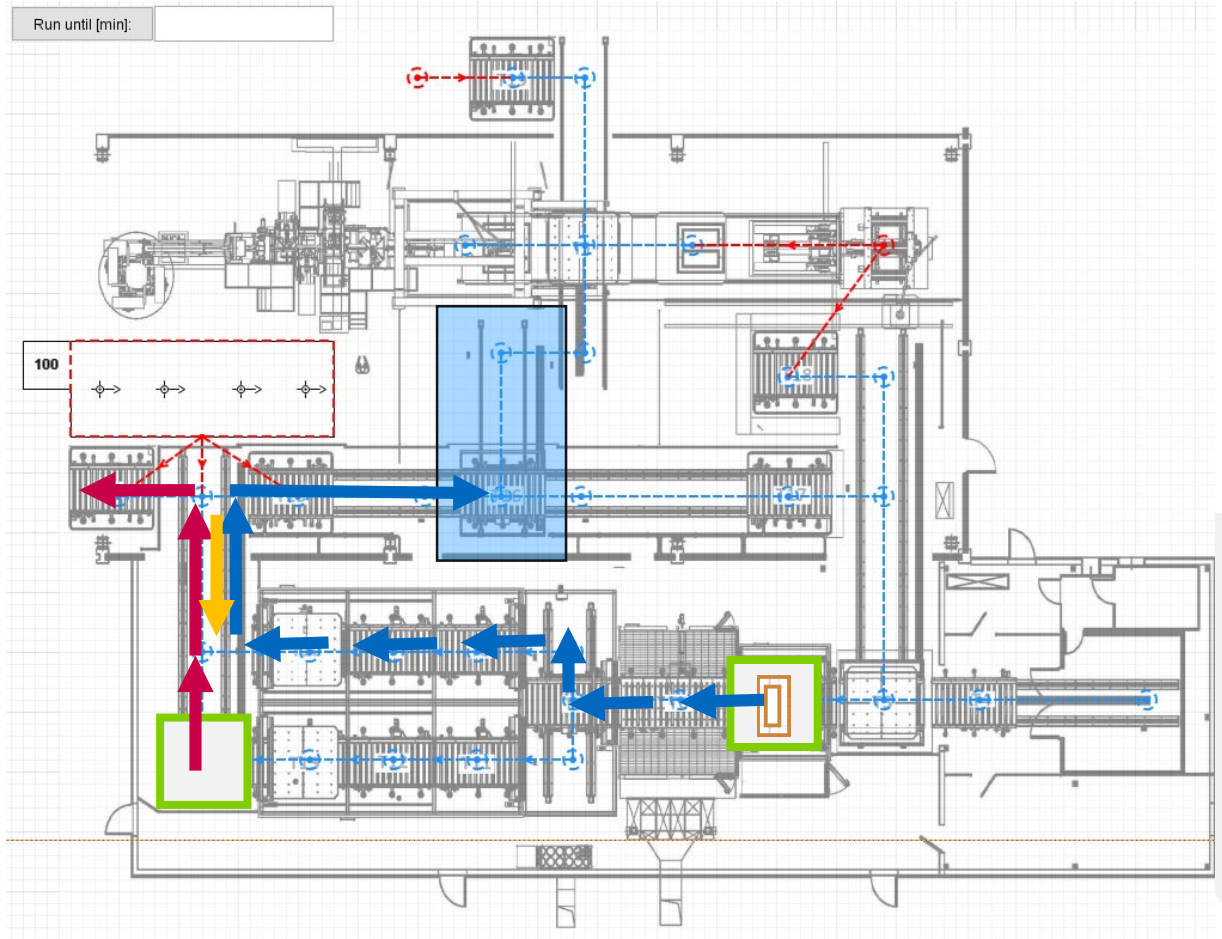
Roberto Grugni

Stuart Rossiter

Simplify!

In order to **evaluate the capability of RL** let's simplify the problem:

Take a set of movements in sequence that move the Transformer Core to the target position starting from a random point and considering one extra supporting steel pallets that can block the way



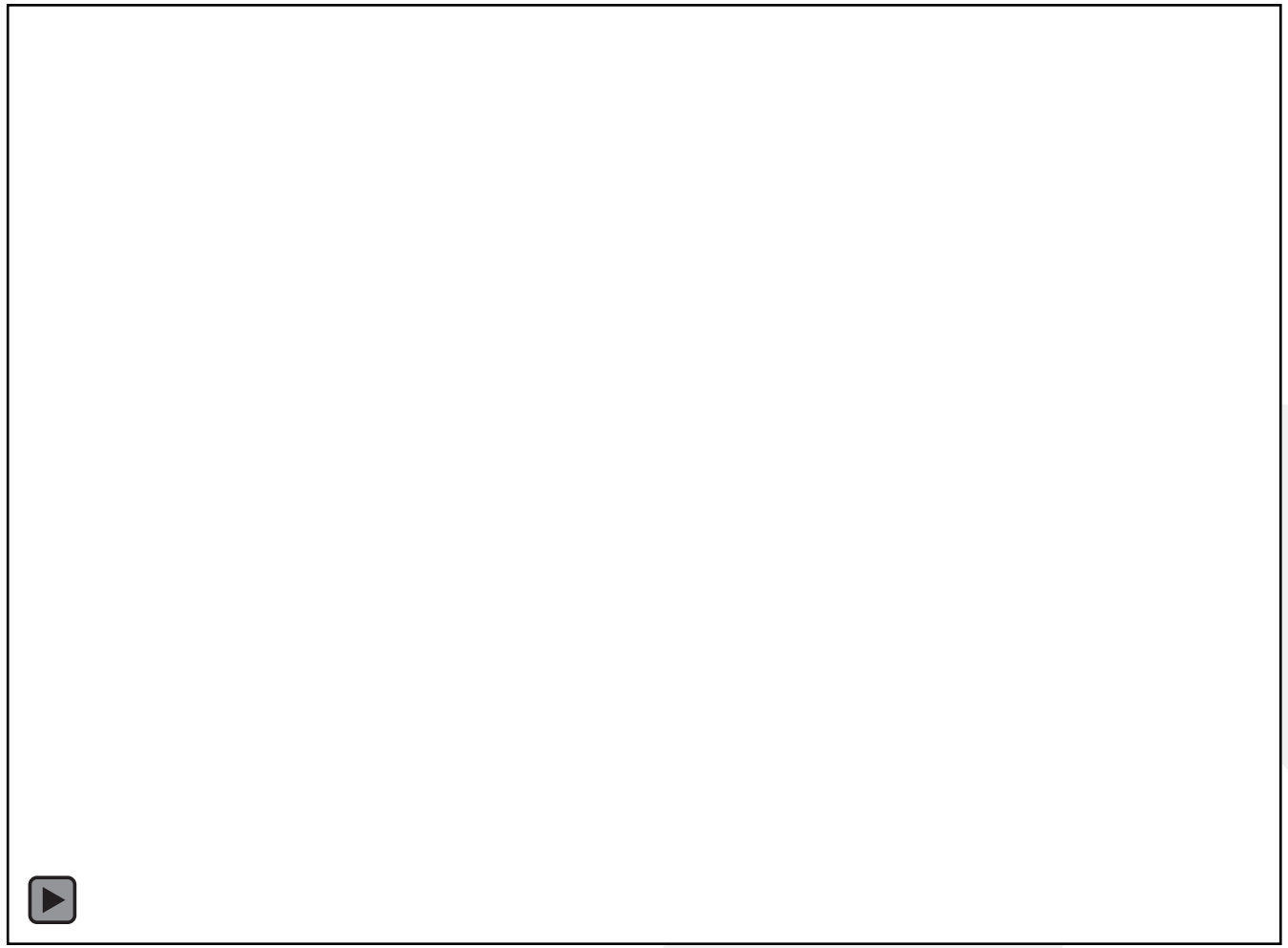
Demo: Exploration

Possible actions: 64

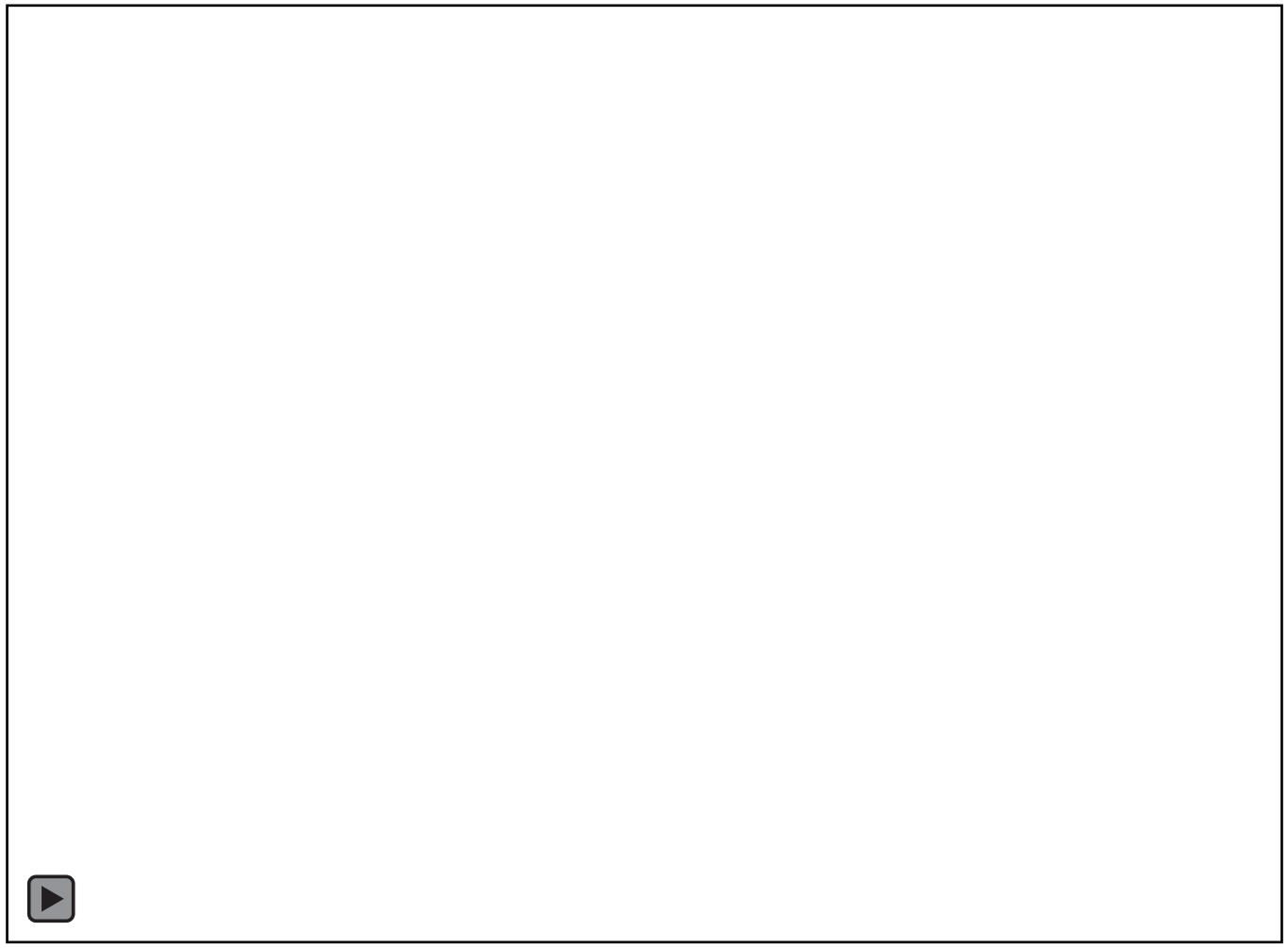
One-hot encoded state vector size: 92

State-Action pairs: 36,578,304

Depending on the actual layout some actions are **valid** and change state, the others are **invalid**



Demo: Learnt policy



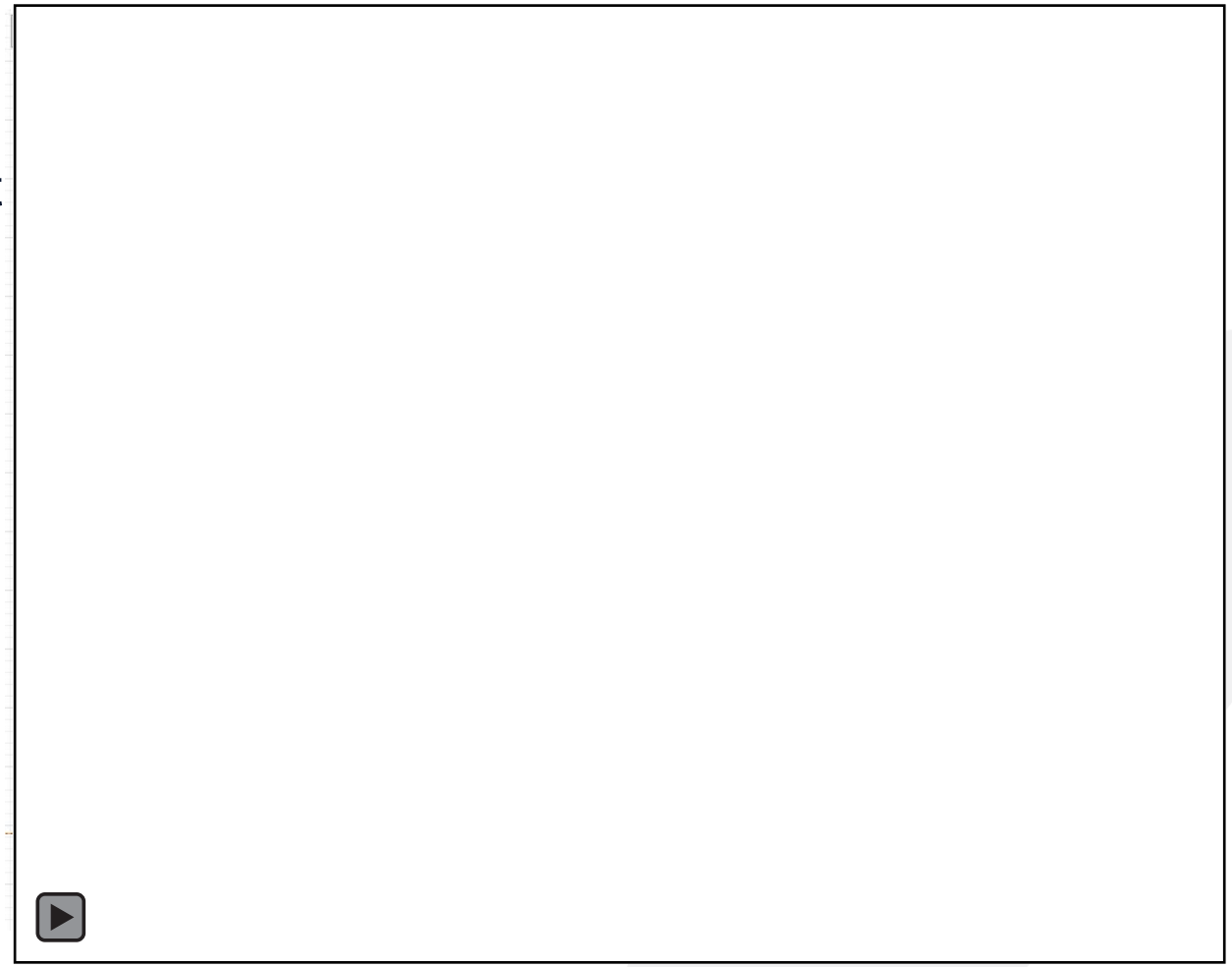
The model (Learnt Policy) can take the core to the target position starting from any given initial layout!

Doing a step further: put together what we learnt

We now know how to make models taking the cores to a target starting at any given position but we can’t produce any core yet!

We can break the **big problem** (production cycle) into a **series of targets**.

We can train and assign **one model** for **each step**.



Remarks and Questions

- Simplify your problem, otherwise the computational effort grows exponentially!
- Design the learning scenario intelligently by breaking it down into several pieces
- Designing smart “reward function” is sometimes the key to a successful learning
- Opaqueness of RL is a challenge for both the learning phase and the selling phase

- How to reformulate problems to handle high dimensional observation and action spaces?
- There are many RL paradigms: tabular vs approximator scenario, single agent vs multi agent, off-policy vs on-policy. How to choose?
- Most of the effort in developing RL algorithms are in Python. Java based packages can be restrictive.
- How negative is overfitting? What about forgetting?

We positively examined the possibility of using Reinforcement Learning

For us, this is just the beginning of a journey for a happy reunion of AI and Simulation!

A final remark

RL is not still a complete **off-the-shelf product**, we need to build up expertise in our simulation community and be a part of this evolution!

Luigi Manca
Delivery Manager
luigi.manca@eng.it