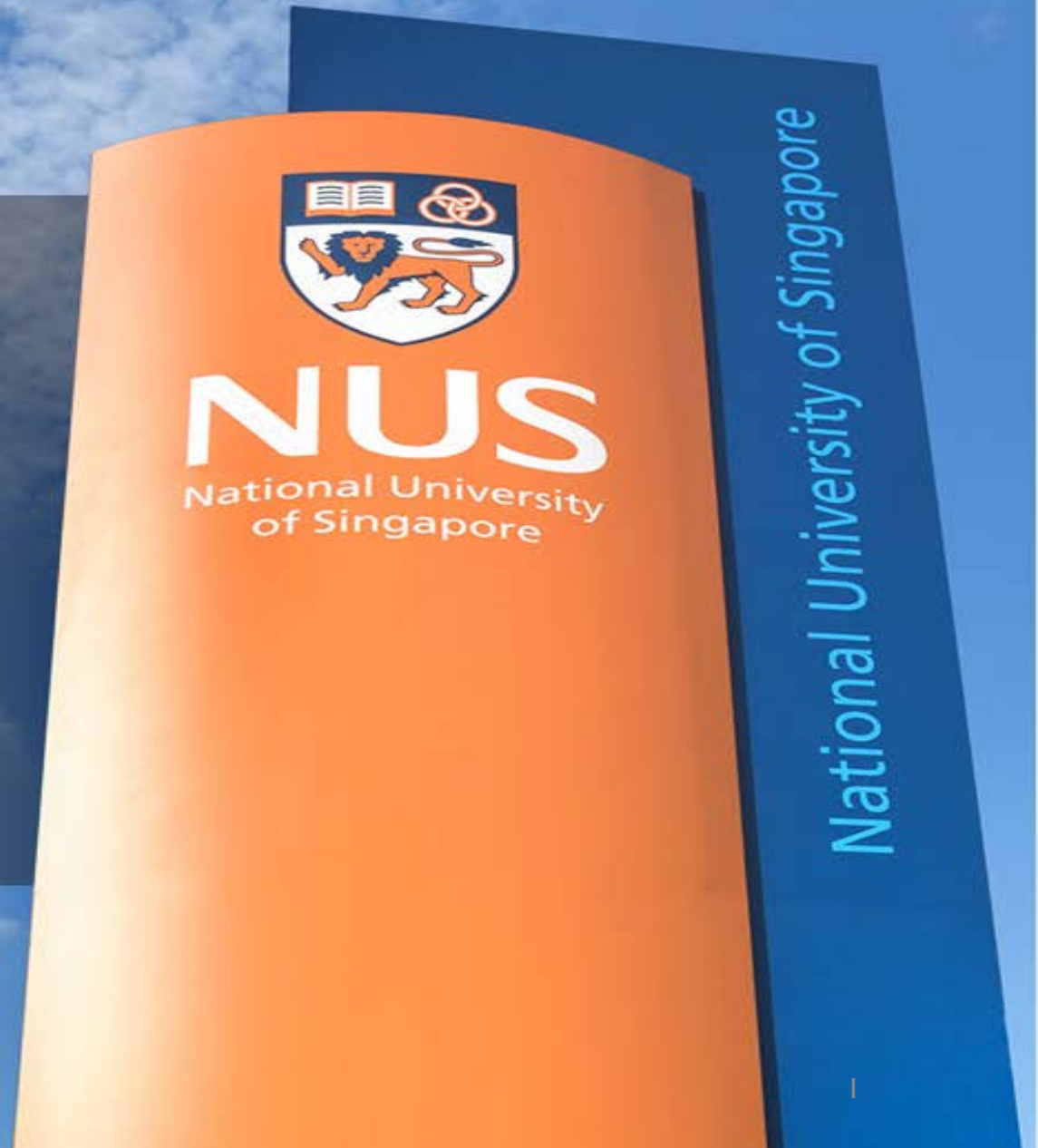
Development of Smart Digital Twin for Maritime Industry via O²DES Framework

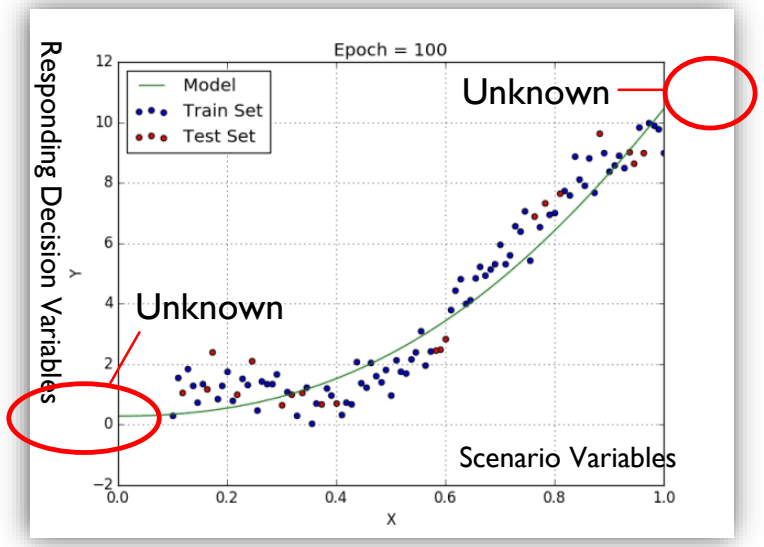
Dr. LI Haobin
National University of Singapore
Department of Industrial Systems Engineering and Management
Centre of Excellence in Modelling and Simulation for Next Generation Ports

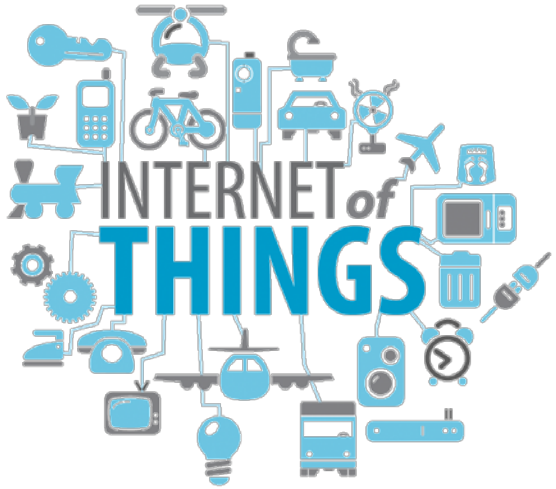


Outline

- Smart Digital Twins
  - Background – Industry 4.0
  - Complexity of Systems
  - Simulation as an Exploration Tool
  - The Four Dimensions
- Proposed Methodologies
  - Classification of Simulation
  - Formalisms for Discrete-Event Systems (DES)
  - Object-Oriented DES (O^2DES) Framework
- Future Trends
  - High-Level Architecture
  - Containerization with Microservice

Background – Industry 4.0

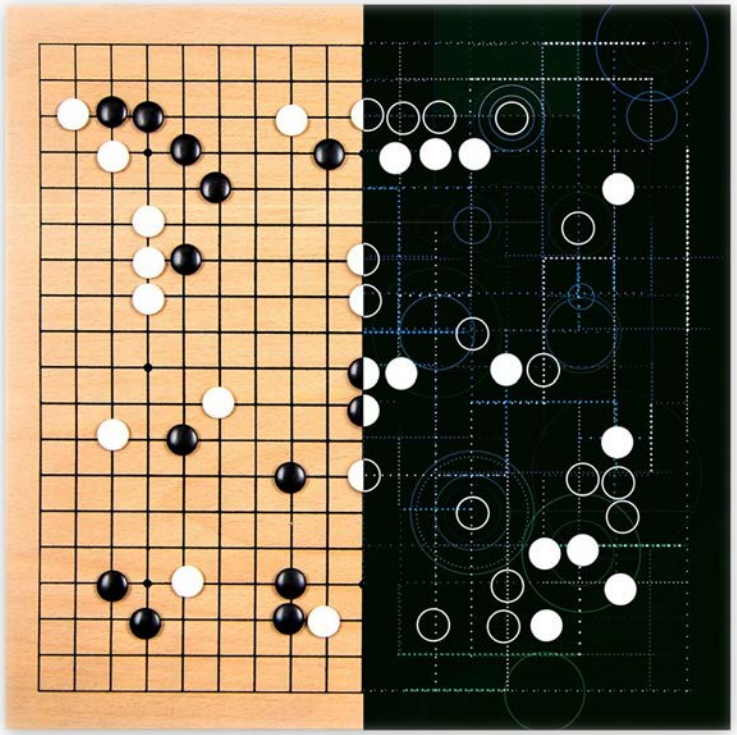




Reinforcement Learning + Deep Learning

Explore (Mining) → **Summarize** (Refining) → **Propagate** (Transporting)

Complexity of Systems



The unknown is not about individual components,

but how they are combined and interact.

Same principle applies for a complex industrial system, e.g., a mega container port

Simulation as an Exploration Tool

Commercial off-the-shelf simulation software

- For evaluating specific design / configuration
- Manually configured

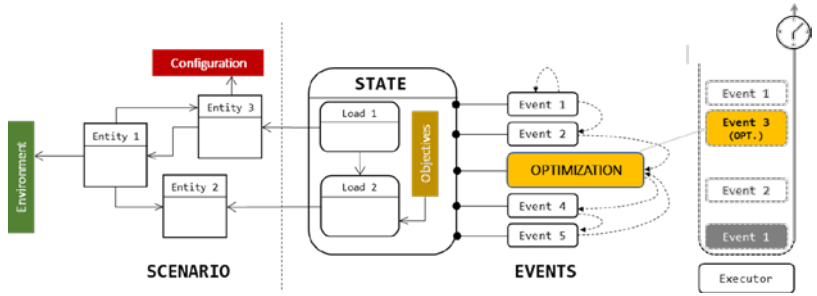
Interface with Human Users

Simulation

Interface with Machine / Algorithms

Bridging Physical & Digital World

Simulation as an Exploration / Experiment Tool

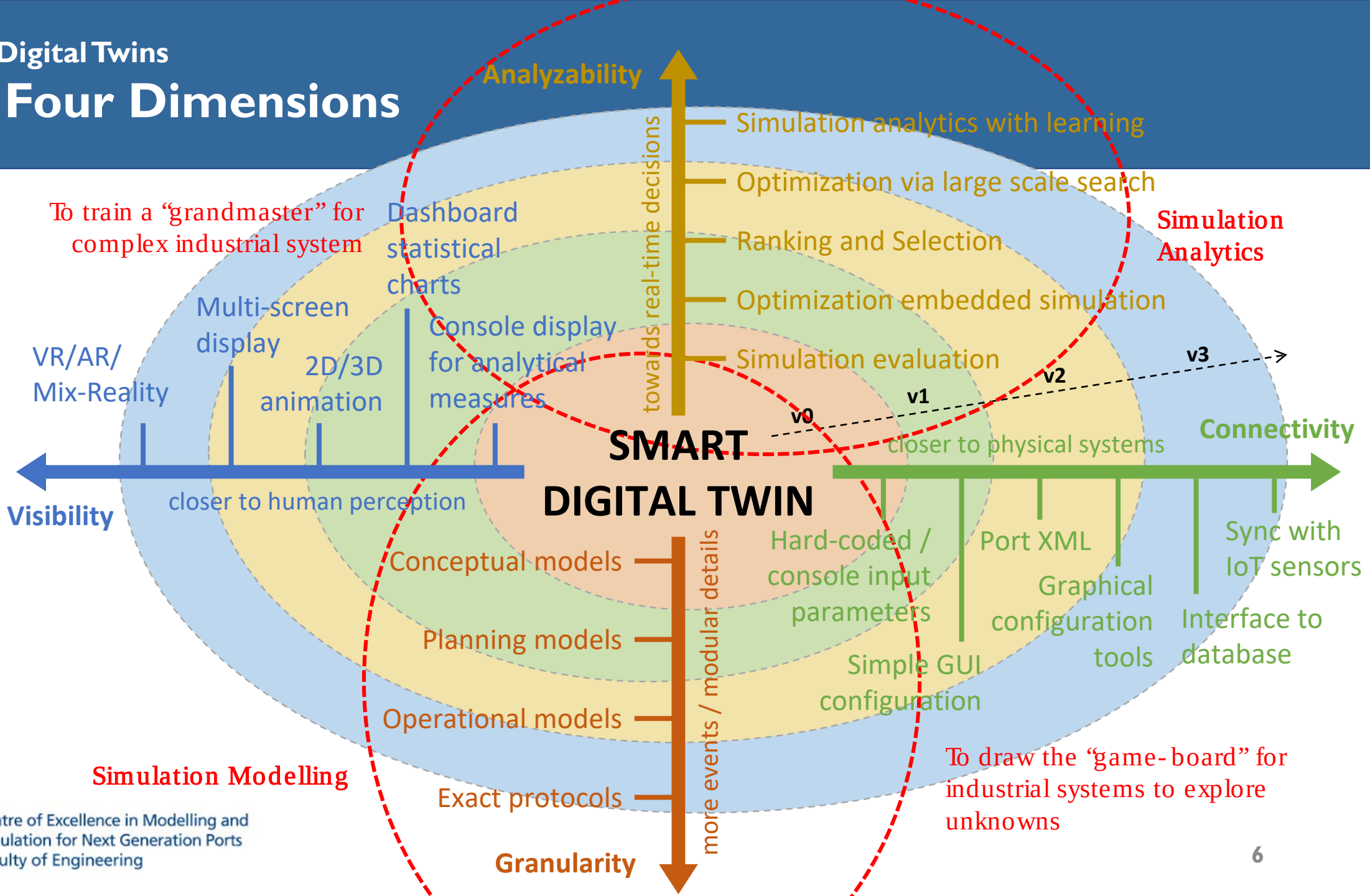


An object -oriented framework / paradigm to

- Facilitate DES model building
- Automate model configuration
- Integrate DES with optimization
- Bridge DES to other AI components, e.g., deep learning & data analytics

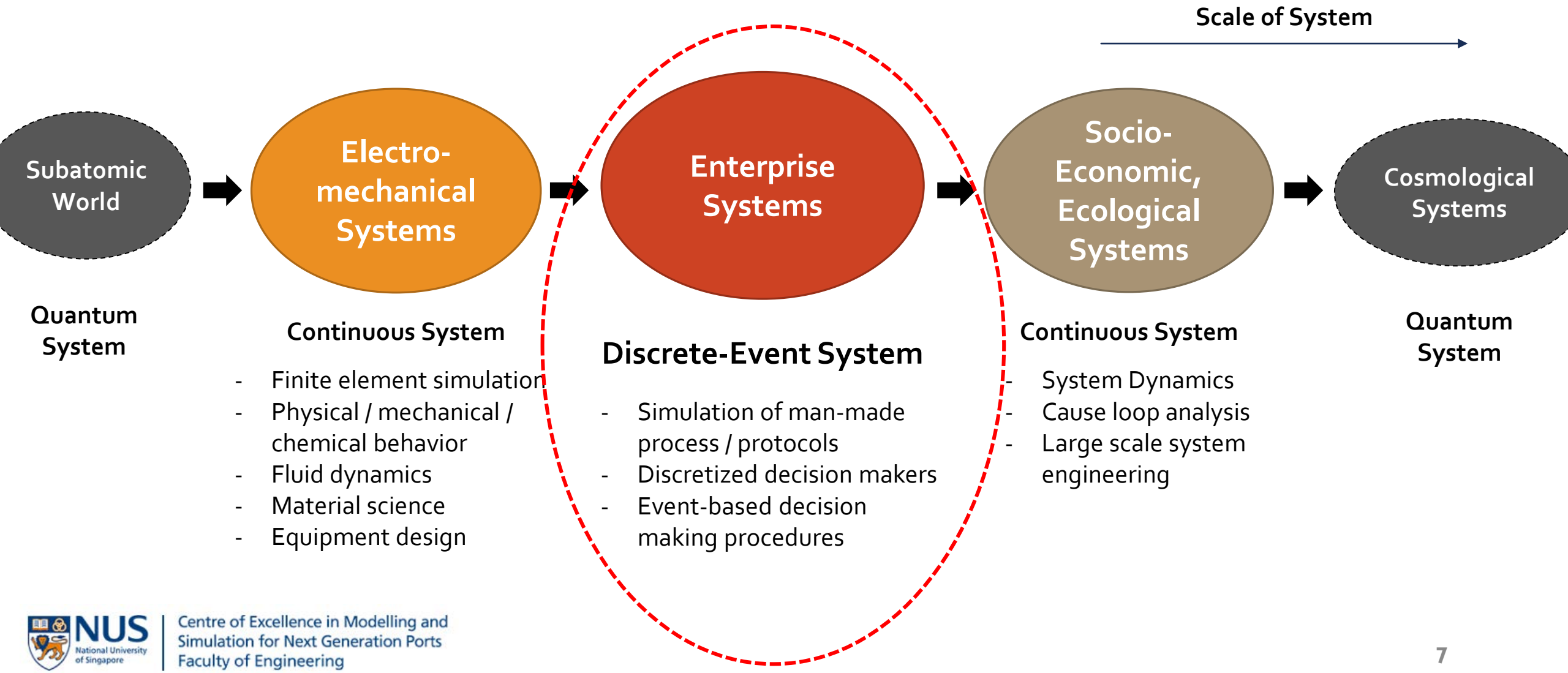
Smart Digital Twins

The Four Dimensions

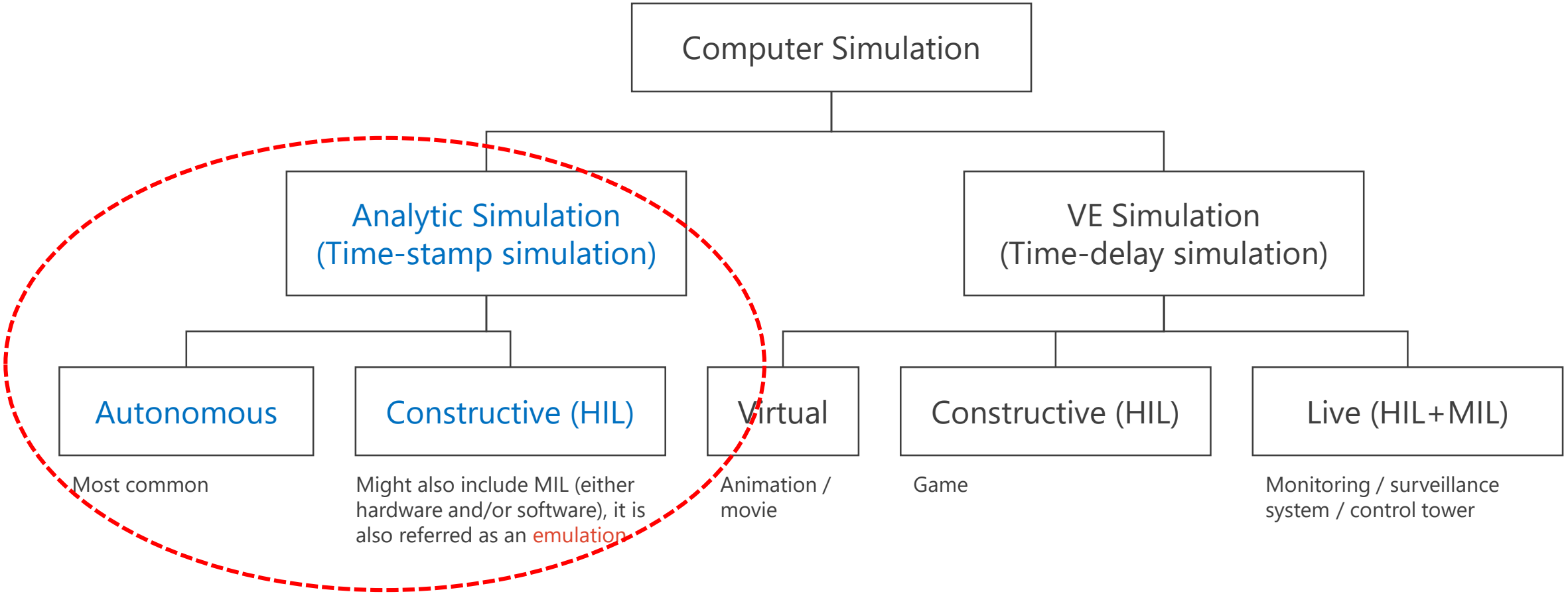


Proposed Methodologies

Classification of Simulation (by State Variables)

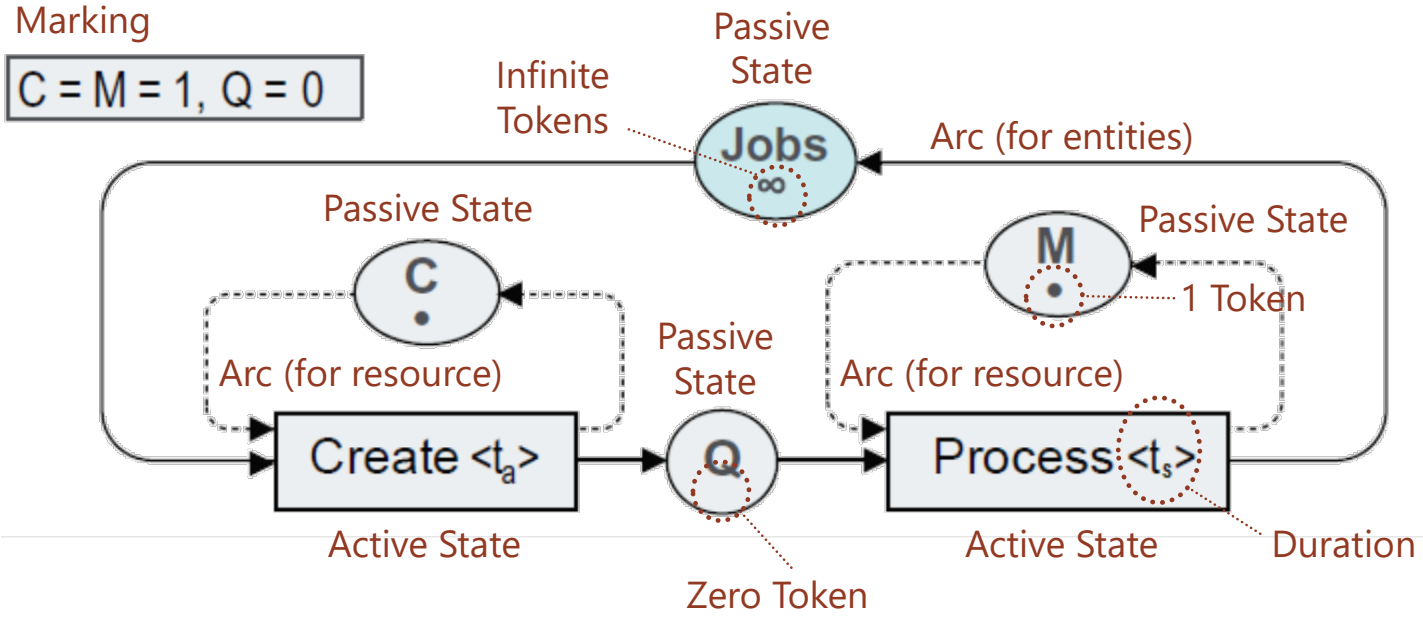


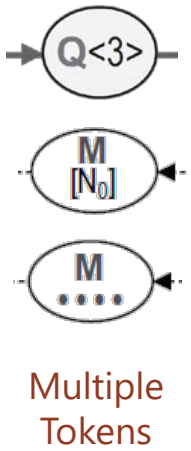
Classification of Simulation (by Time Representation)



Formalisms for Discrete-Event Systems (DES)

- In the classical activity-cycle diagram an activity typically represents the interaction between an entity* and active resources. An entity or an active resource is either in a passive state called a queue or in an active state called an activity. Queue nodes and activity nodes are connected by arcs.





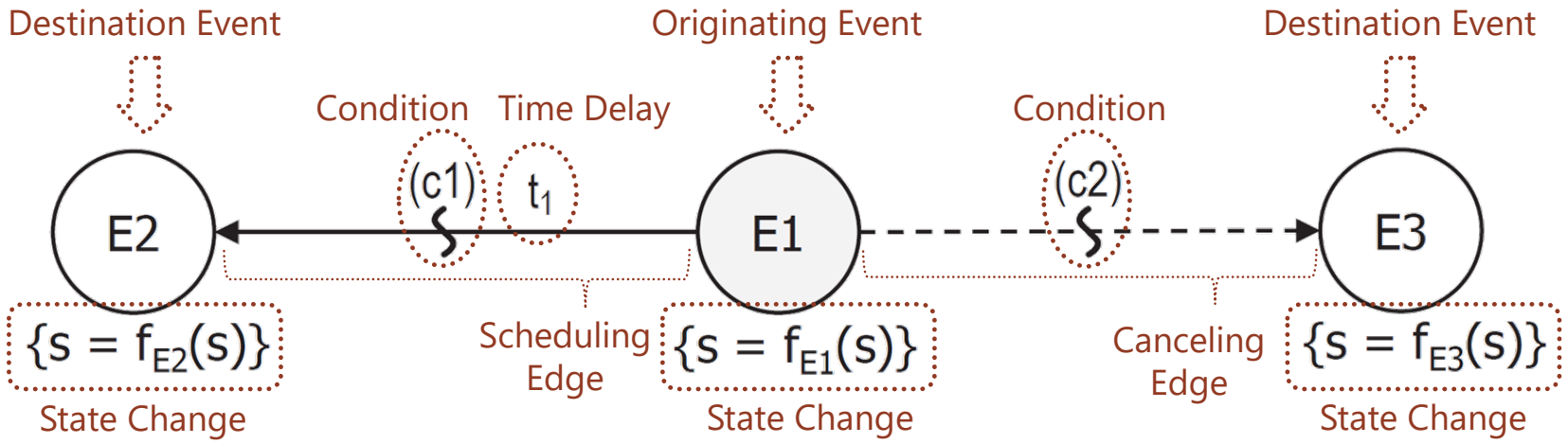
Intuitive,

but less flexible

Activity-based Formalism

Formalisms for Discrete-Event Systems (DES)

- An event graph is a graphical formal model consisting of a set of event nodes and a set of directed edges. It provides a complete description of a discrete event system (DES) in a concise and clear manner, and its execution rules have to be described unambiguously.
- A graphical model can be specified in algebraic form (to be analyzed by a human logically) as well as in computer-readable form (to be executed on a computer).

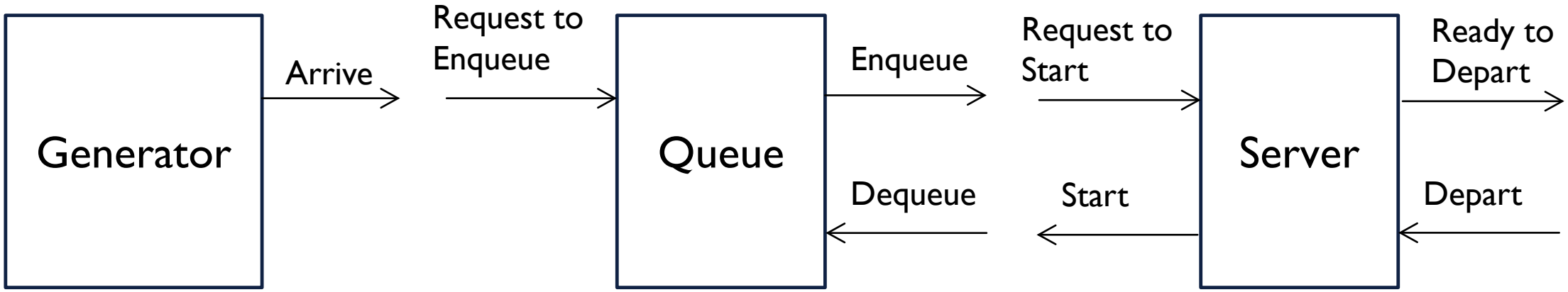


Precise,

but Sophisticated

Event-based Formalism

Formalisms for Discrete-Event Systems (DES)



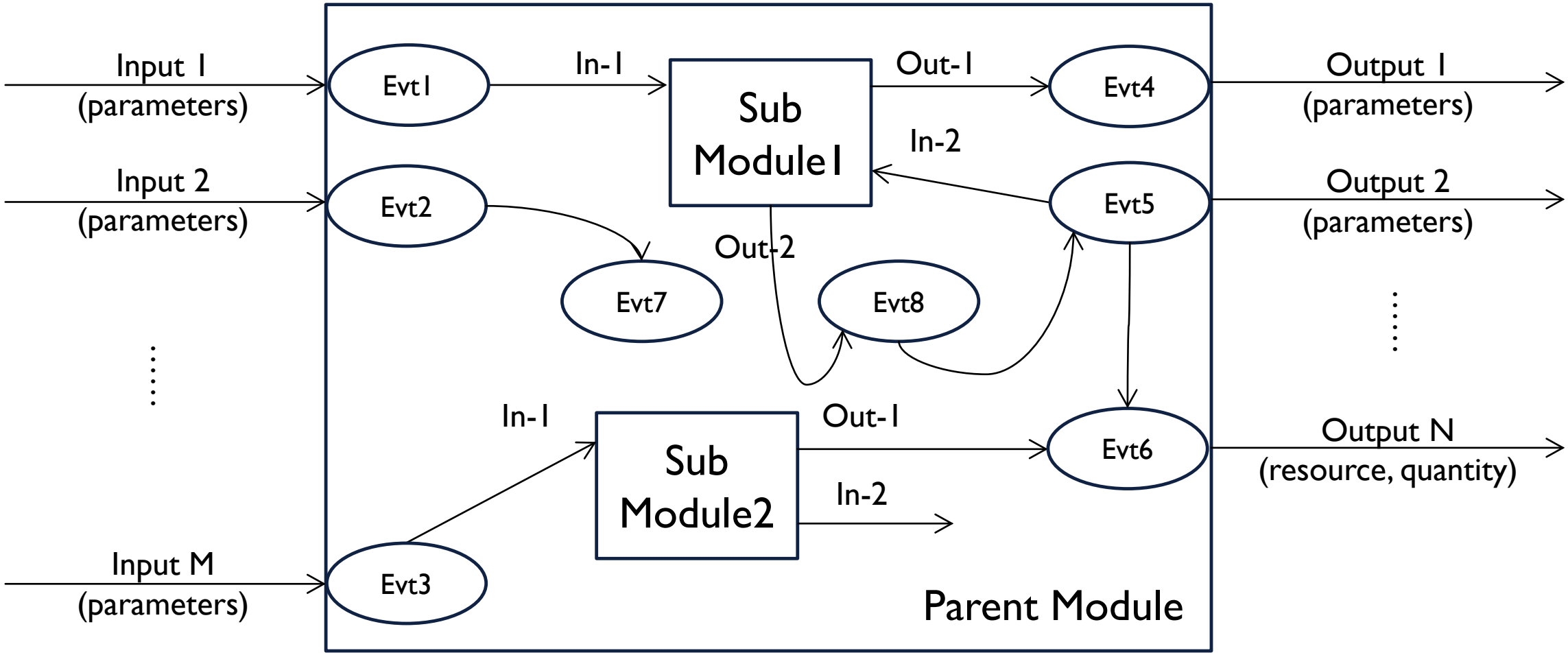
State:
How many arrivals are generated?

State:
How many are in the queue?
- When the queue has finite capacity, how many are requested but pending to enqueuer?

State:
How many being served?
- When the queue has finite capacity, how many are requested but pending to start?
- How many are "stuck" due to subsequent procedures?

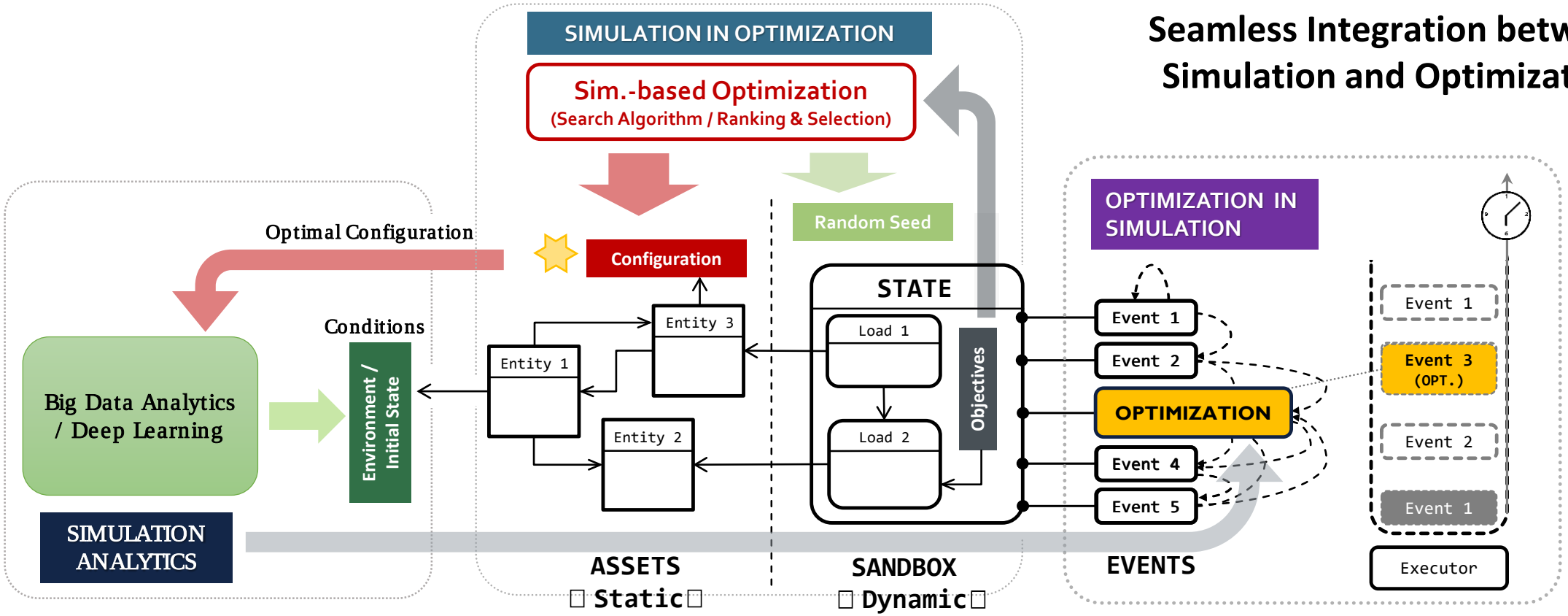
State-based Formalism

Object-Oriented DES (O^2DES) Framework



Object-Oriented Formalism

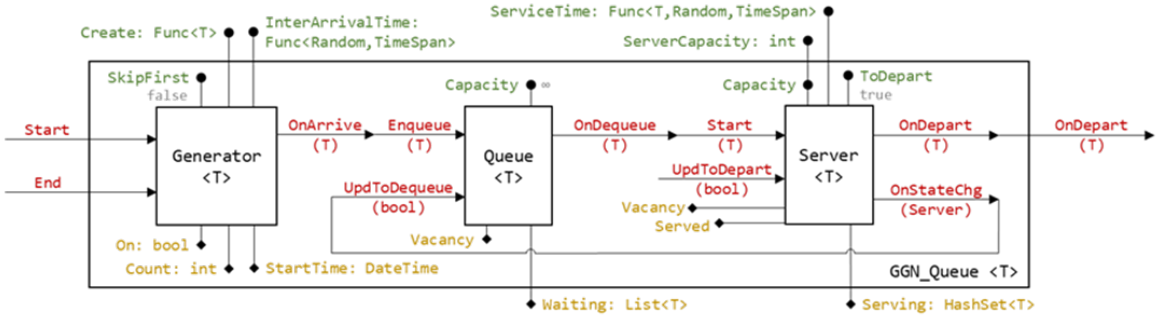
Object-Oriented DES (O²DES) Framework

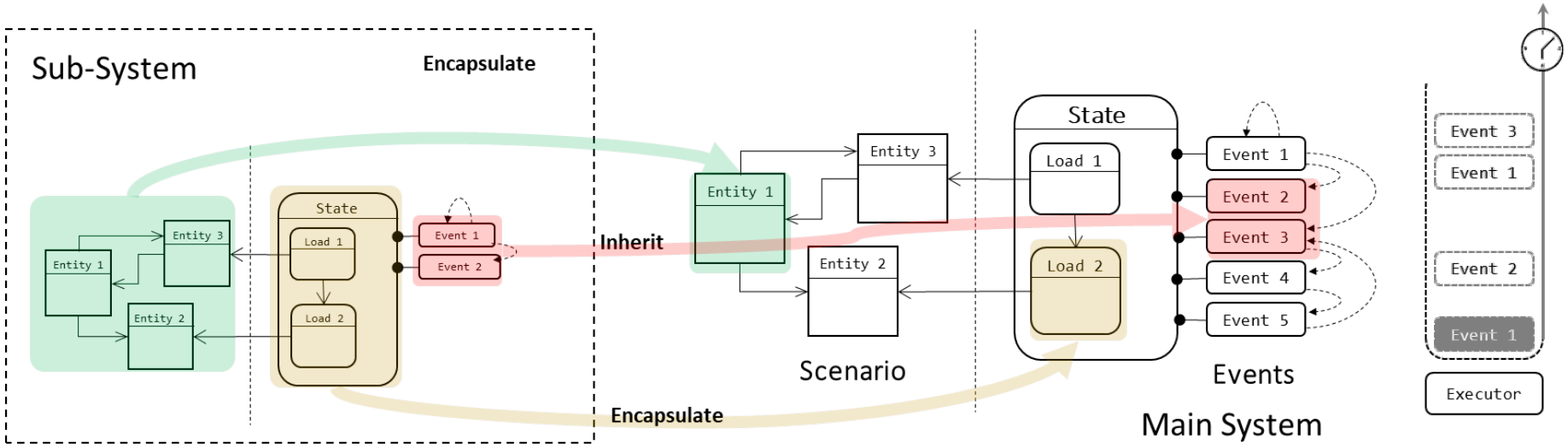


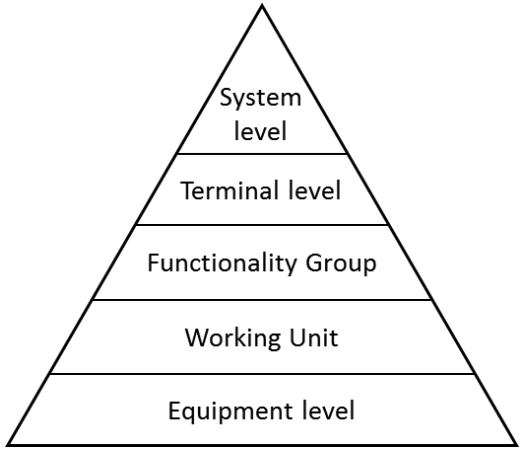
Seamless Integration between Simulation and Optimization

Object-Oriented DES (O²DES) Framework

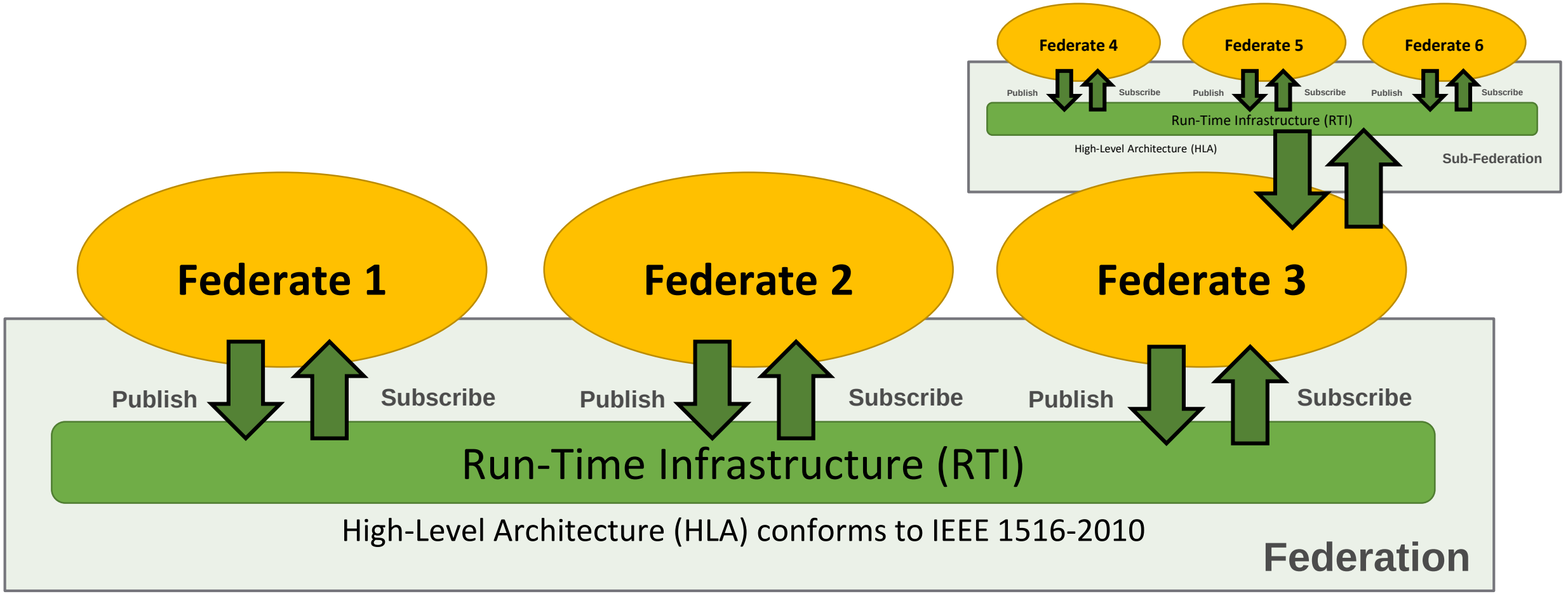
Hierarchical Modelling with Modularization



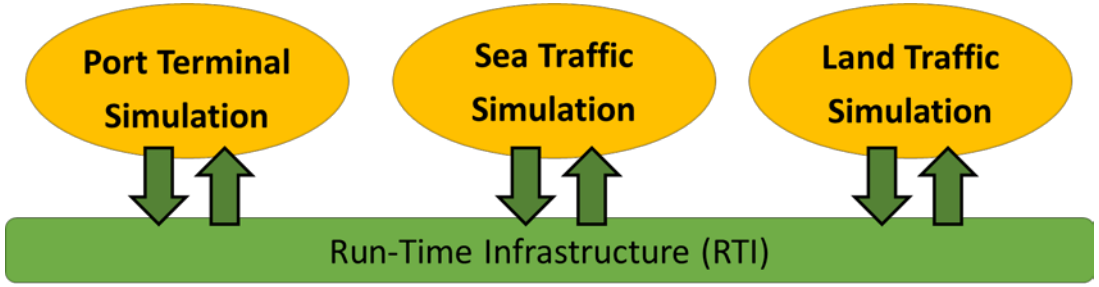




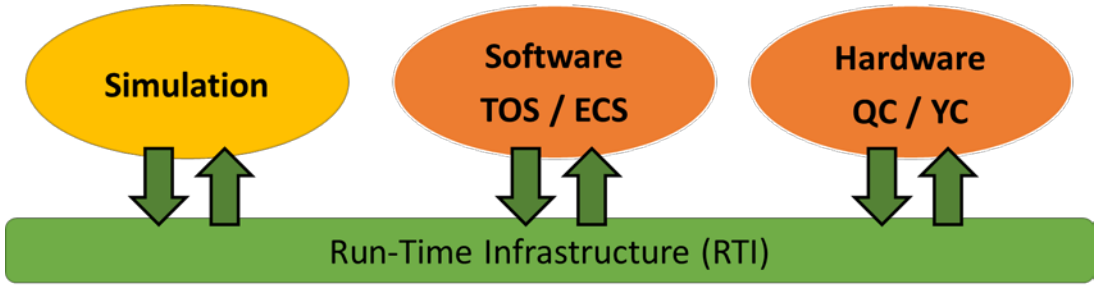
High-Level Architecture (HLA)



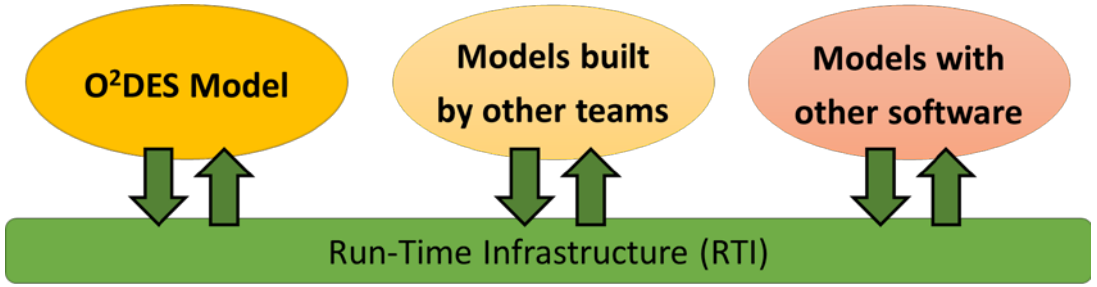
High-Level Architecture (HLA)



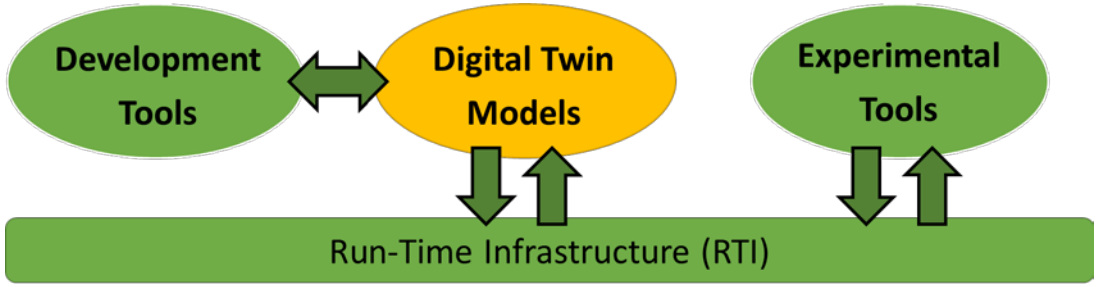
Multi-Module Distributed Run



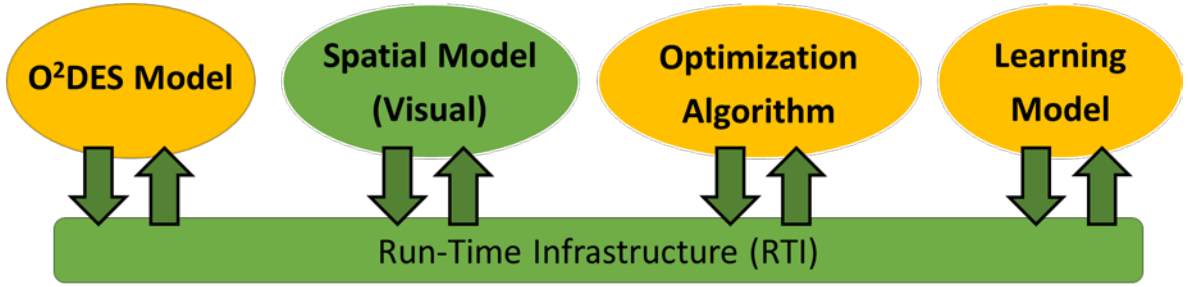
Simulation – Emulation Seamless Transition



Multi-Formalism Synchronized Run

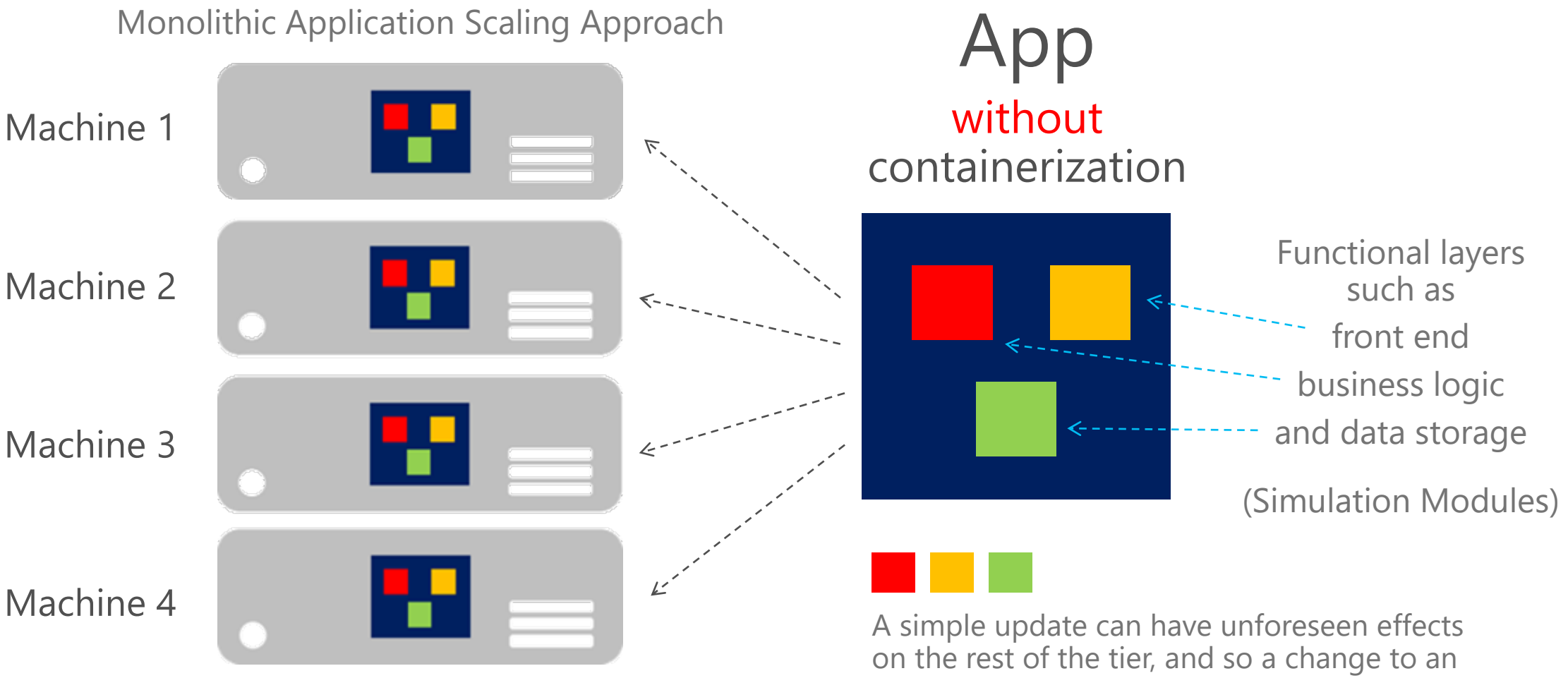


Digital Twins Development and Experiment

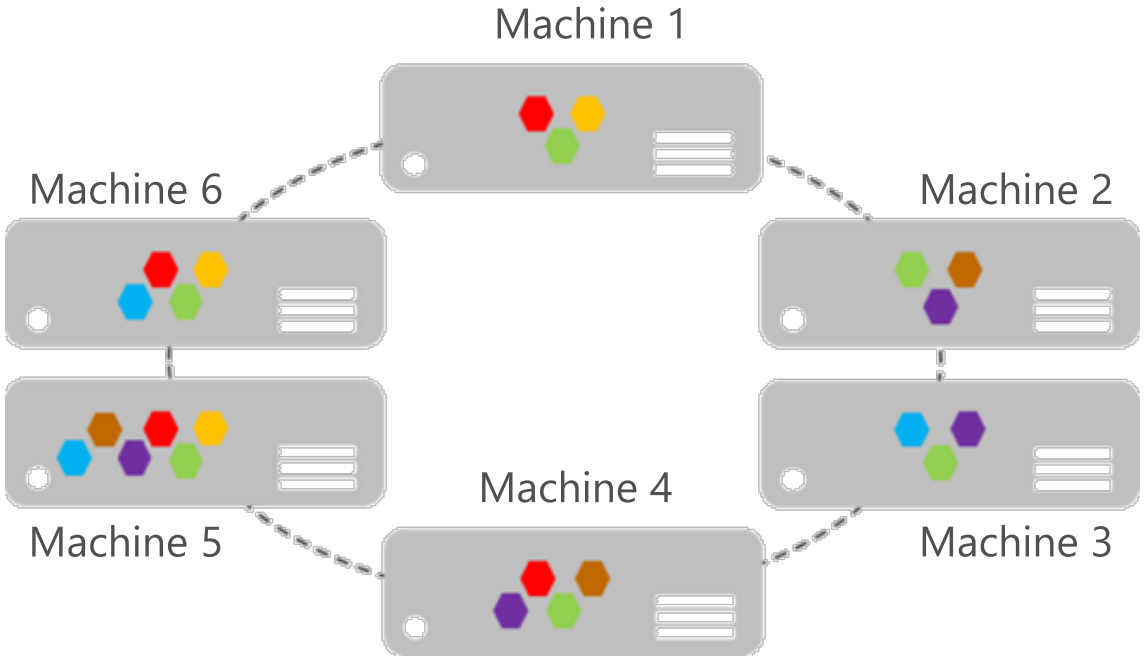


Integrated Simulation + Visualization + Optimization + Learning

Containerization with Microservice



Containerization with Microservice



Microservices Application Scaling Approach

scale + agility + reliability

App1 App2

with microservices (containerization)

(Simulation Modules)

Microservices are independent components that work together to deliver the application's overall functionality.

The term microservice emphasizes that applications should be composed of services small enough to reflect independent concerns.

Thank you!

Q&A