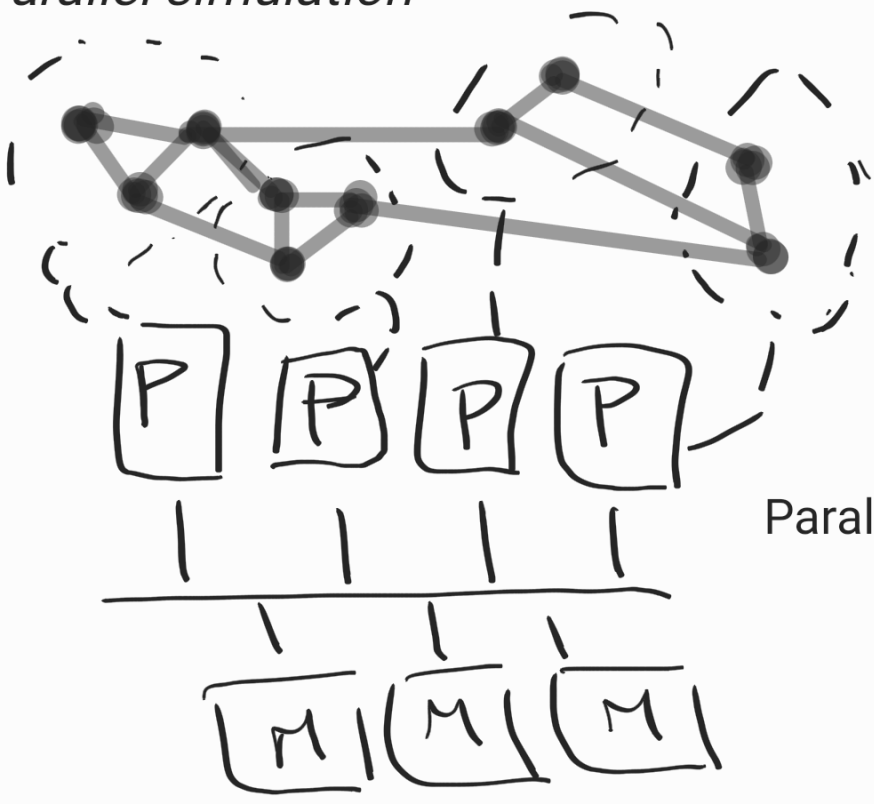
Parallel and distributed simulation

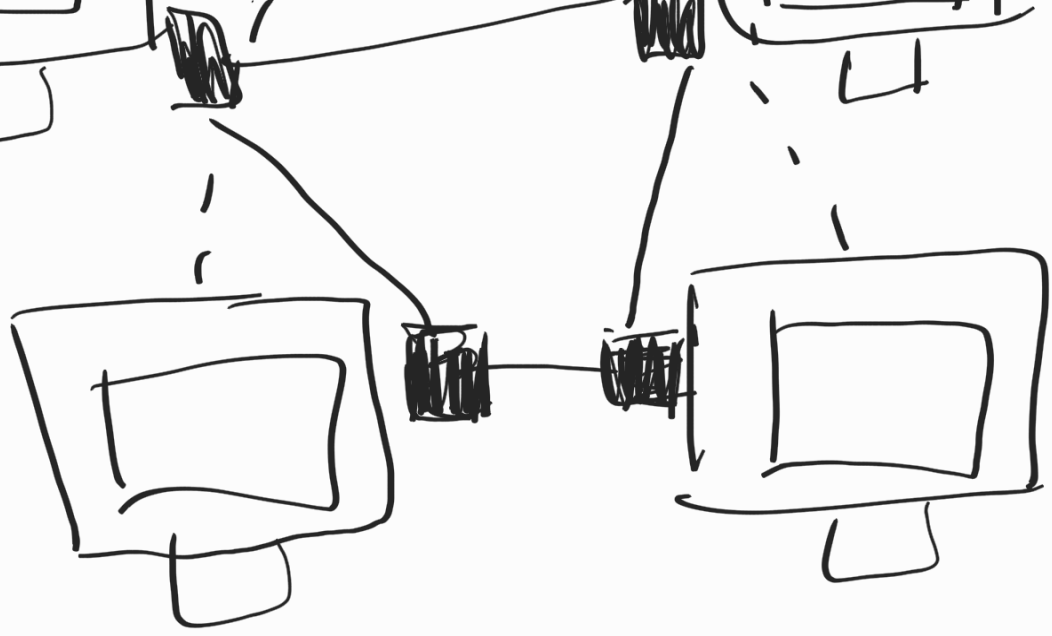
Parallel simulation



Parallel processor

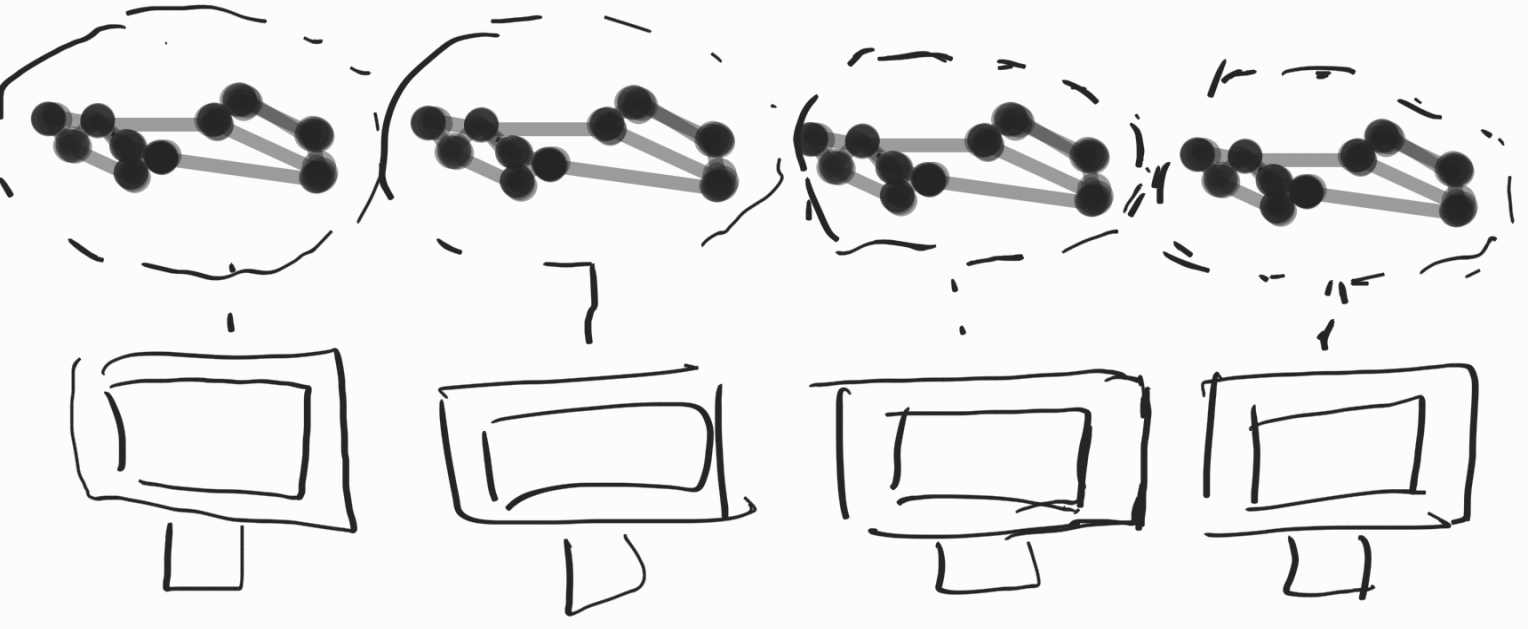
Execution of a single simulation program on a collection of tightly coupled processors

Distributed simulation



Execution of a single simulation program on a collection of loosely coupled processors

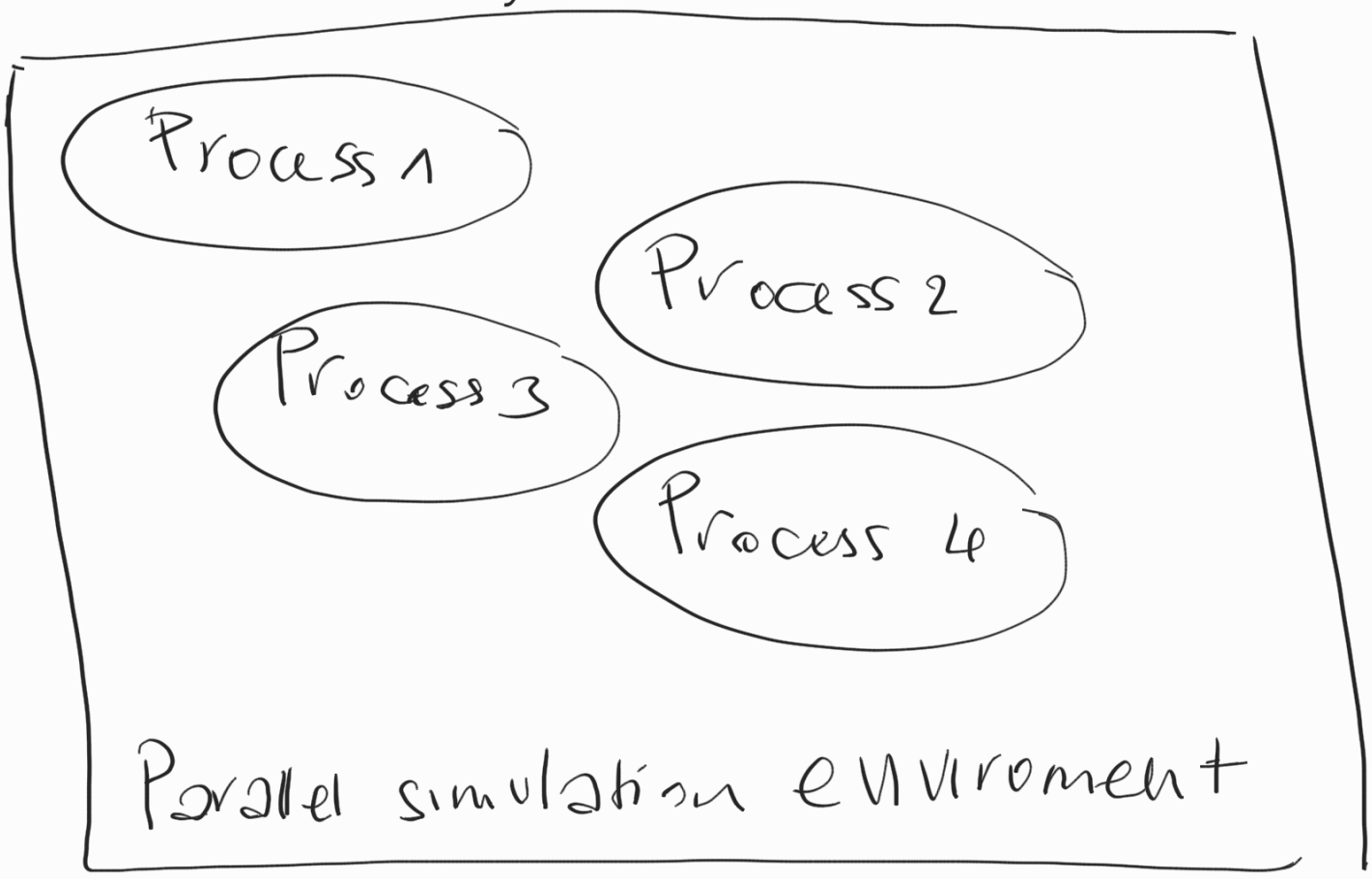
Replicated trials



Execution of several independent simulations concurrently on different processors

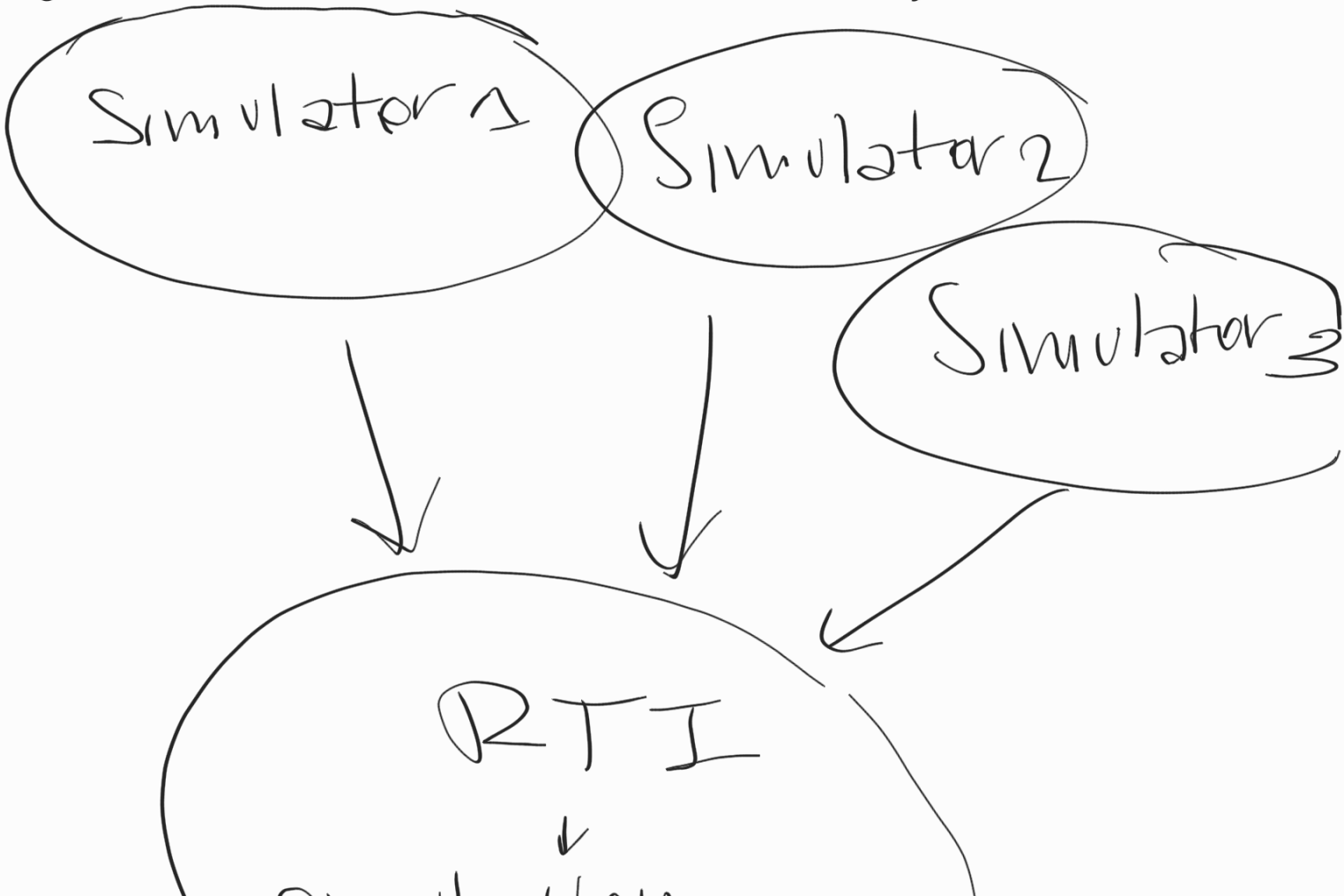
Un'altra differenziazione è STAND ALONE VS FEDERATED SIMULATION SYSTEMS

Stand-alone simulation system



Tipo nel caso dell'object oriented

High level Architecture--> federated simulation systems



Application domain :

- Parallel discret event simulation
- Distributed virtually enviroment

Time line HLA :
born in 1996,
HLA1516 (2000-2005),
HLA evolved in 2010

Concept of INTEROPEREABILITY:
Is a property referring to the ability of different systems and organisations to work together. Usually used in industry and military sectors.
Moreover, it's the capability by joint conformance with a given set of standards, that enables heterogeneous equipment, generally built by various vendors, to work together in a network environment IEEE 1278.2

Software interoperability
The ability of 2 or more systems or components to exchange information and to use the information that has been exchanged. IEEE Standard Computer Dictionary

Simulation composability
The capability to select and assemble simulation component in various combinations into valid simulation systems to satisfy problem changes from the composition of components to one of component interoperability when a substantial effort is required

DIS Interoperable : two or more simulations that , for a give exercise, are DIS compliant and DIS compatible and whose performance characteristics support the fidelity required for the exercise IEEE 1278.4

Simulation interoperability
The ability of a model or simulation to provide services to and accept services from other models and simulations, and to use the services so exchanged to enable them to operate effectively together.

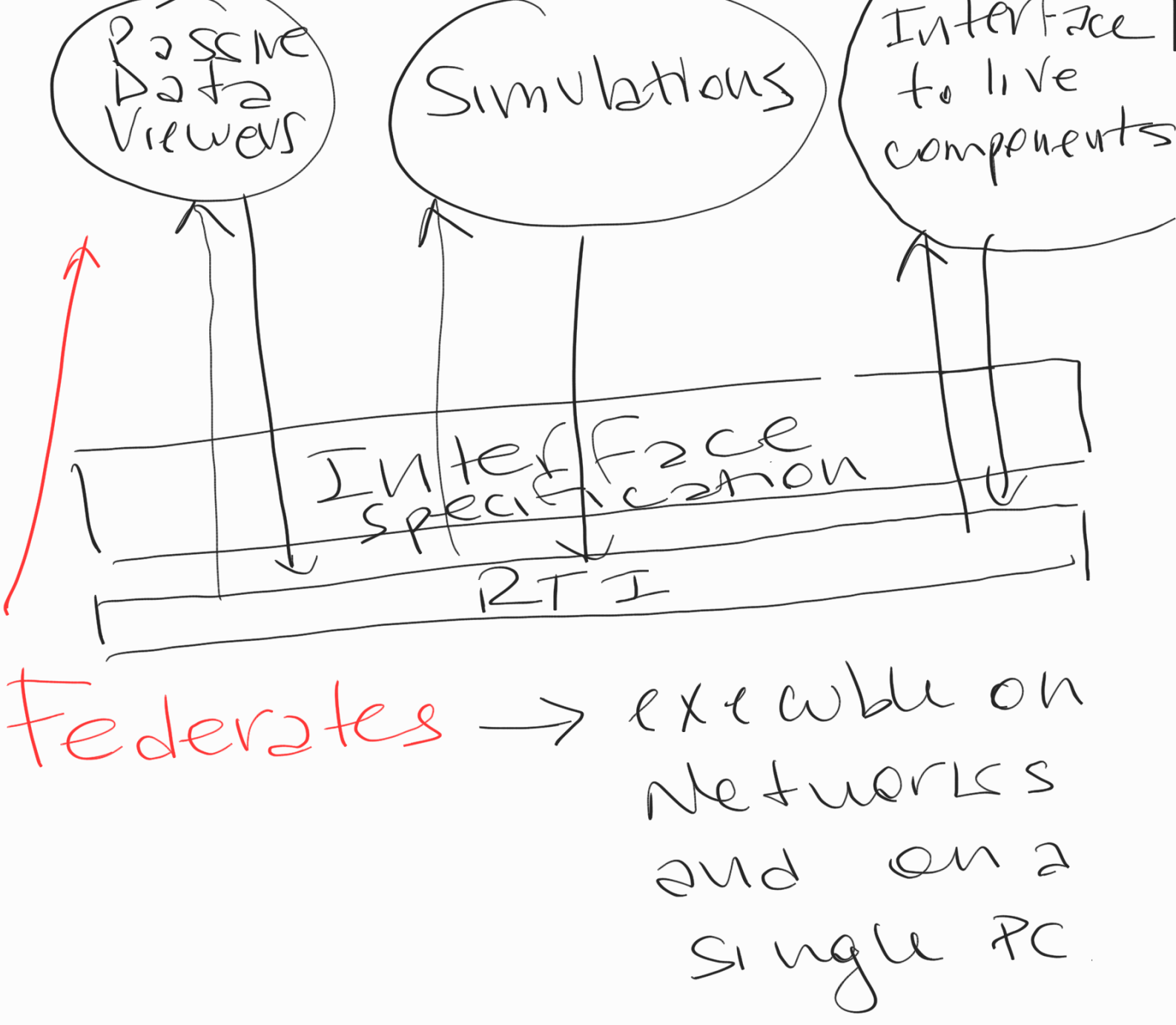
V&V for complex systems - validation and verification two activity to guarantee always during simulation

DA RIVEDERE

HIGH LEVEL ARCHITECTURE

Designed to facilitate interoperability among simulations and to promote reuse of simulation and their components. The HLA is composed of three major :
1.HLA rules : set of ten basic rules that together describe the general principles defining HLA.
2. HLA INTERFACES: description of interface between federates and RTI.
3. HLA object model template: a specification of the common format and structure for documenting HLA object models

Example of HLA Federation



Originally the behaviour of HLA was conceptualised to create repository library of objects and permit a link among them. If it was not possible to create a standard on the rules to exchange data, doesn't matter because IoT allow to define characteristics of objects and so exanchange that objects defined as I want to.

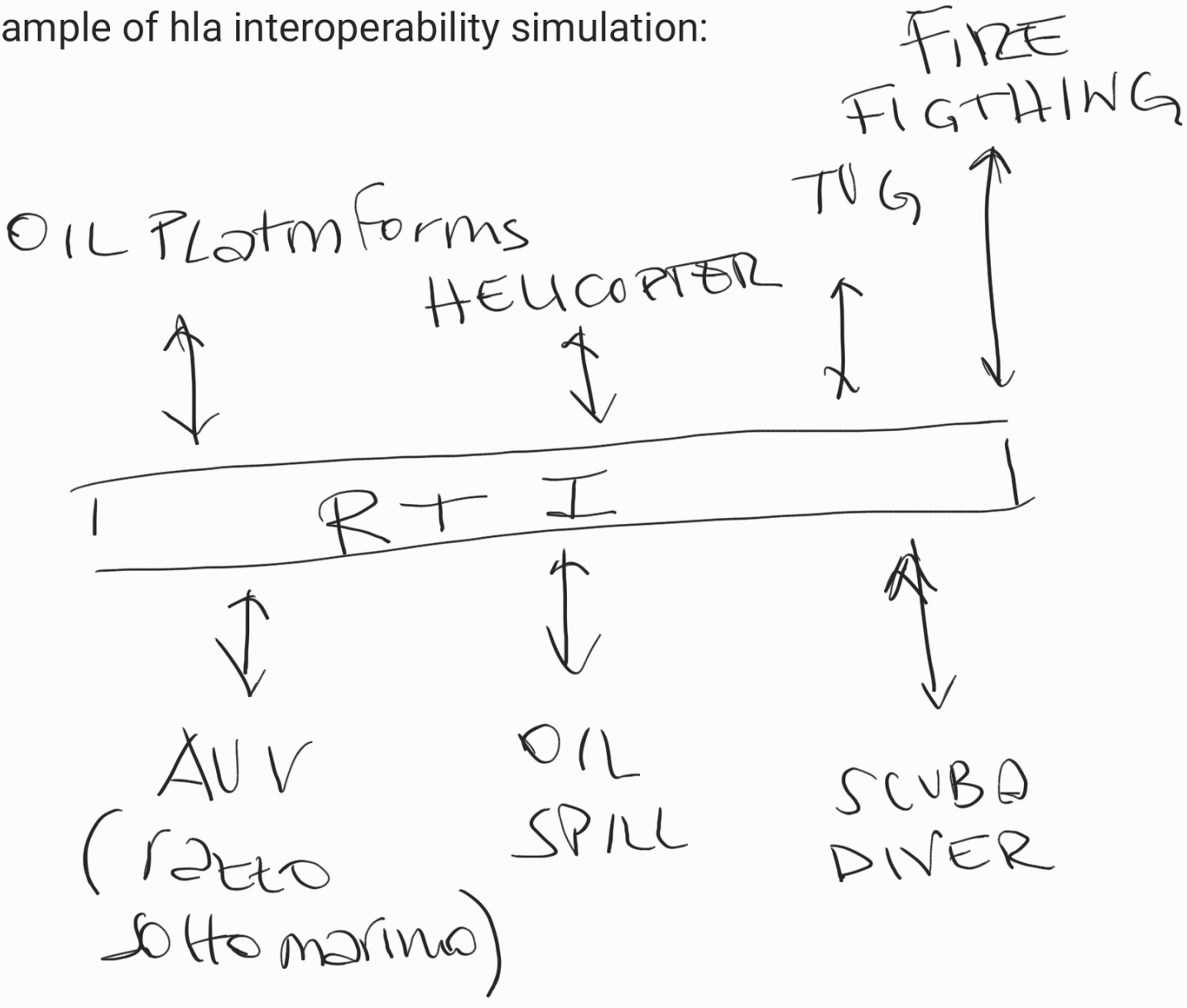
INTERFACE SPECIFICATION :

- Federation management: to create/join/syncronize/delete Federation execution
- Declaration management: what kind of attribute to publish and what to subscribe and the interactions

- Object management: create and delete object instances. Control attribute and interaction publication. Create and delete object reflection
- Time management: issue about logical time to guarantee is syncrinized and advanced in the proper way
- Data Distribution management: to distribute info's of data

Set of RTI services. Interface specifications identifies the callback functions each federate must provide.

Example of hla interoperability simulation:



FEDERATION RULES of HLA

10 rules : 5 for federete and 5 for Federation that tdefine how to deal to carry out interoperability. Describes the simulation and federate responsibility.

OBJECT MODEL TEMPLATE

Provide a common method for recording information.
3 components.

5 RULES OF FEDERATION:
1. Each Federation shall have a federation object model, to define how it organised, documented in accordance with OMT.
2. All rappresentation of objects in the FOM shall be I the federates, not in RTI. THAT MEANS RTI DOESN'T INCLUDE ANY ELEMENT OF FEDERATION.
3. During a FedExec , all exchange data among federates shall occur via the RTI.
4. During a Federation execution, federates shall interact with the RTI I accordance with the HLA interface specification
5. During FedExec an attribute of an instance of an object shall be owned by only one federate at any given time

5 RULES OF FEDERATE:
6. Every federates shall have a SOM (SIMULATION OBJECT MODEL), documented in accordance with OMT
7. Should be able to update or reflect any attributes of objects in their SOM
8. Federates should be able to transfer or accept ownership of attributes dynamically during a FedExec
9. Federates should be able to vary the conditions under which they provide updates of attributes of objects
10. federates should be able to manage local time in a way which will allow them to coordinate data exchange with other members of a Federation.

RTI

Software that provides common services to simulation systems.

Implementation of the interface spec. An architectural foundation encouraging portability and interoperability. Separate simulation & communication. Improves on older standards. Facilities construction and destruction of federations. Supports object declaration and management between federates. Assists with federation time management.

OMT

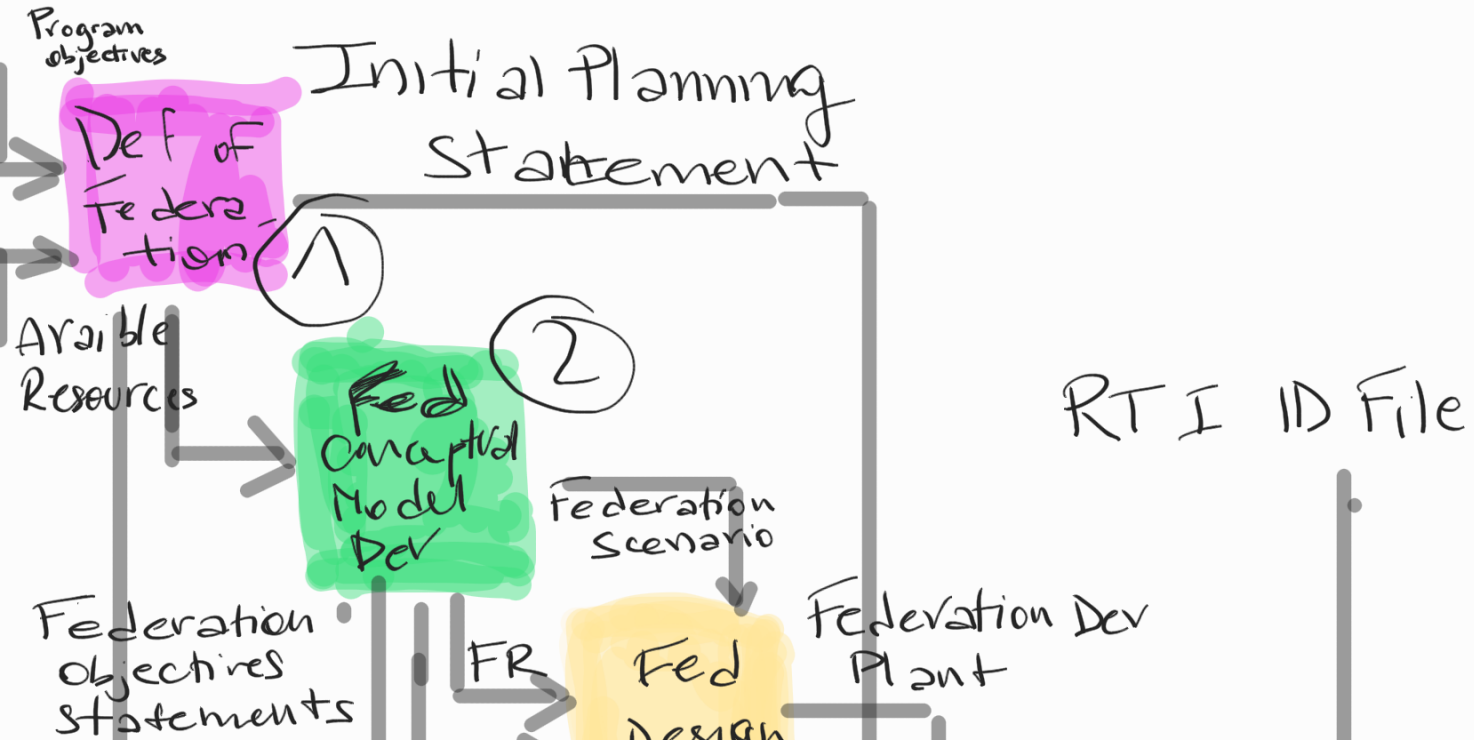
Provides a common framework for HLA. One of the goal is reusing different components.
We have informations such as Objrct Class structure table, Object Interation table, Attribute /parameter table, FOM/SOM lexicon.

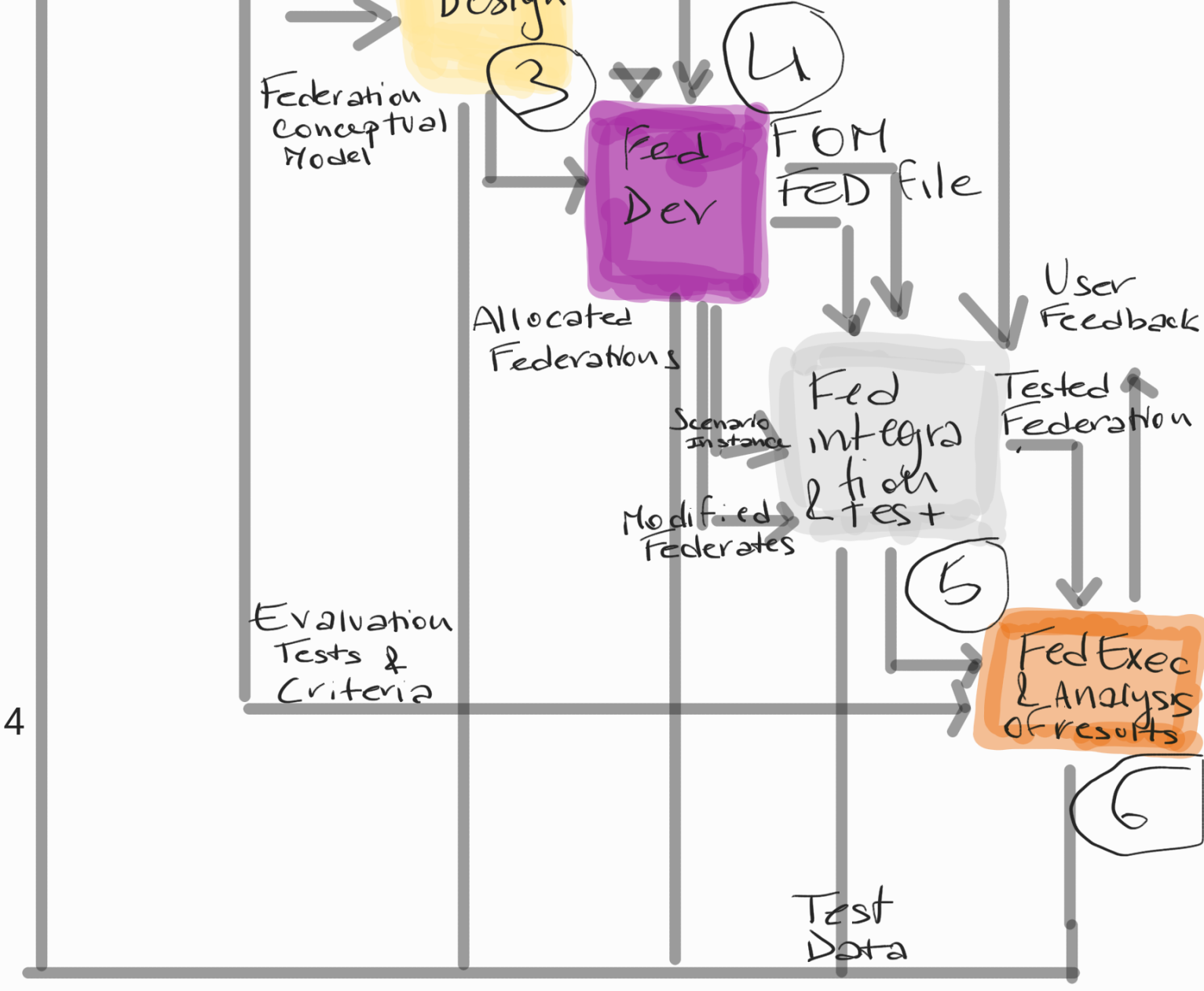
FOM --> one per Federation. it define all info's that are shared and refer to all the issues of interfederate relationship.

SOM --> one per federate. Describe characteristics of federate,describes inside operations.

MOM --> unibpversal definition. Identifies objects and interactions used to manage a Federation.

FEDEP





1. Define objective of federation:
Identify user and sponsor nees, develop objectives.
Resource defined could be: software language, compiler, tools etc

2. Develop federation/ conceptual model
Federation requirement so the input of design, the conceptual model, the SCENARIO. This last is very important because allow you to specify what you have to do.

3. Design Federation
Select federates
Developed what kind of Federation design.

Prepare a plan.

4. Federation development
Output the FED file, the scenario instance and modified federates.
Develop FED --> Federation Execution Data
Develop FOM --> Federation Object Model
Establish Federation agreements
Implement Federation designs
Implement Federation infrastructure

5. Integrate and Test Federation
First test is to see if federates and federations join.
Input: RTI RID file --> RTI INTERCHANGE DATA file
Output_1: tested Federation
Output_2: tested data

6. Execute Federation and prepare Results
Take in input the outputs of 5 and the test evaluation criteria from 2. Outputs : user feedback and reusable