Introduction to HLA 4

Björn Möller & Fredrik Antelius
Pitch Technologies

Agenda

- Introduction
- High-Level Architecture (HLA) for the Warfighter
- Technical capabilities of HLA
- Common federation object models
- HLA 4 Overview
- Common considerations
  - LVC gateways, performance, tools and security
- Summary and additional reading

Learning Objectives

After taking this tutorial you should be able to:

- Describe the requirements for interoperability in distributed simulation
- Explain the principles of the HLA standard
- Explain how the HLA standard is used for distributed training purposes
- Describe the new features of HLA 4
- Describe some common Federation Object Models (FOMs) for defense training

Purpose of HLA

- Provide one single interoperability standard
  - Meet a broad set of distributed simulation requirements
  - Support training, analysis, test & evaluation, concept development, engineering, etc.
  - Support a variety of domains and sub-domains with different and evolving object models
- Enable reuse of simulation components, thus reducing cost
  - Vendor-neutral interoperability through open standardization
  - Facilitate an open eco-system for COTS, GOTS and in-house simulations

IEEE

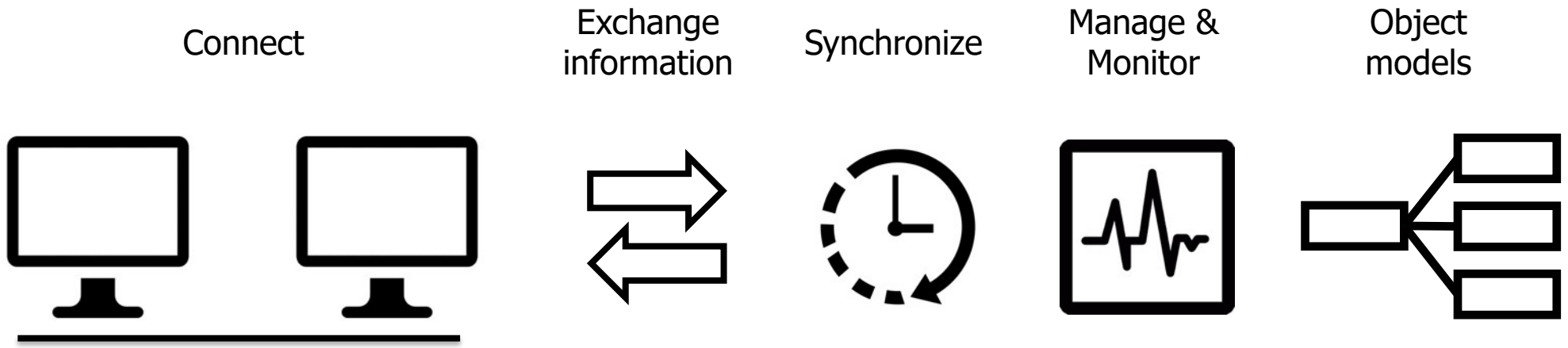
SISO

HLA 4

COTS = Commercial-off-the-shelf
GOTS = Government-off-the-shelf

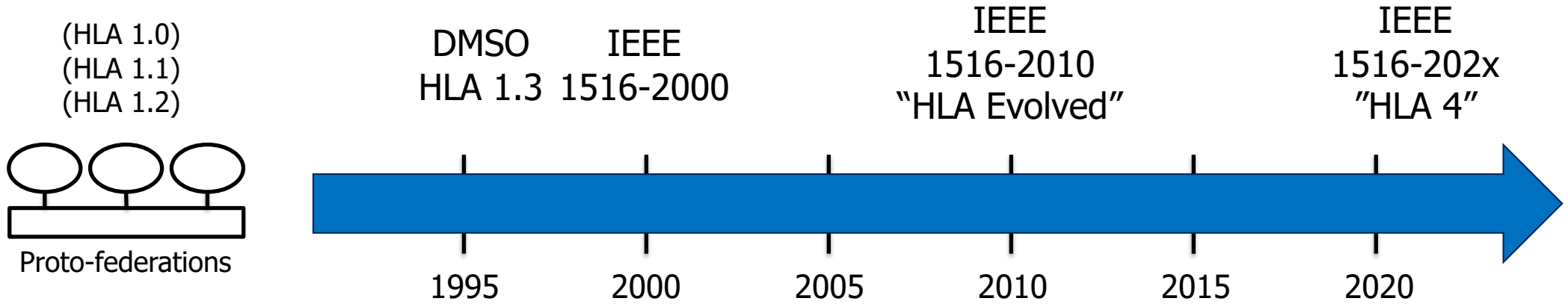
IEEE = Institute of Electrical and Electronics Engineers
SISO = Simulation Interoperability Standards Organization

Overview of HLA Capabilities



- Connect simulations running on different hosts
- Services for information exchange using publish-subscribe and filtering
- Services for synchronization of startup, execution and simulation time
- Services for managing and monitoring the simulations
- Support for object models for different domains

HLA Timeline



- HLA was initiated in the US DoD in the early 1990s
- After developing several proto-federations, HLA 1.3 was released in 1998
- It was brought to open standardization through IEEE, resulting in HLA IEEE 1516-2000
- The standard was updated in 2010 with several new features – "HLA Evolved"
- The standard is currently under revision – "HLA 4"
  - Passed balloting 2023

What's new in HLA 4?

HLA 4

aws

Google Cloud

kubernetes

docker

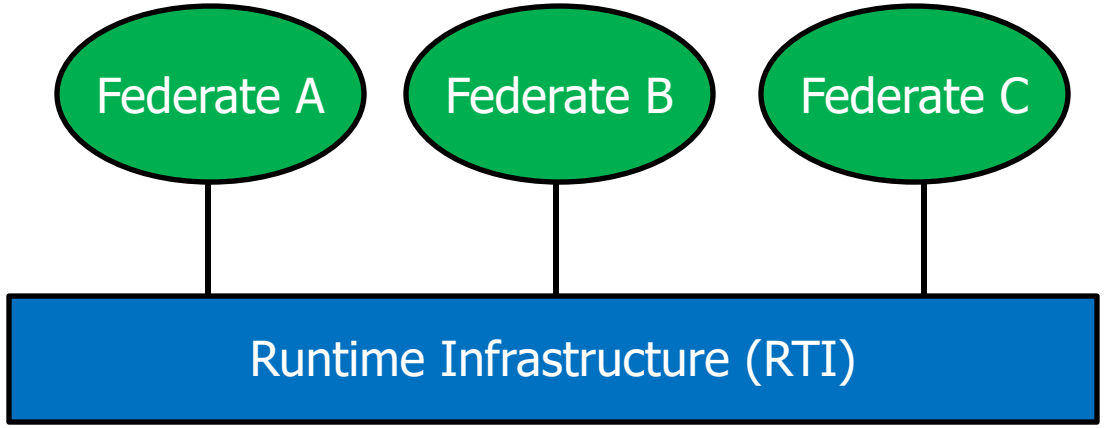
Powerful Modeling

Easier Deployment

Enhanced Security

Increased Scalability

Key HLA Concepts



Federation Object Model

```
<FOM>
  <Shared object classes>
  <Shared interaction classes>
  <More>
</FOM>
```

- Each participating member is called a Federate
- Information is exchanged using a Runtime Infrastructure (RTI)
- The information exchange follows a Federation Object Model (FOM)
- The federates together with the FOM and the RTI are called a Federation
- A session with a federation is called a Federation Execution

HLA for LVC Platform Training

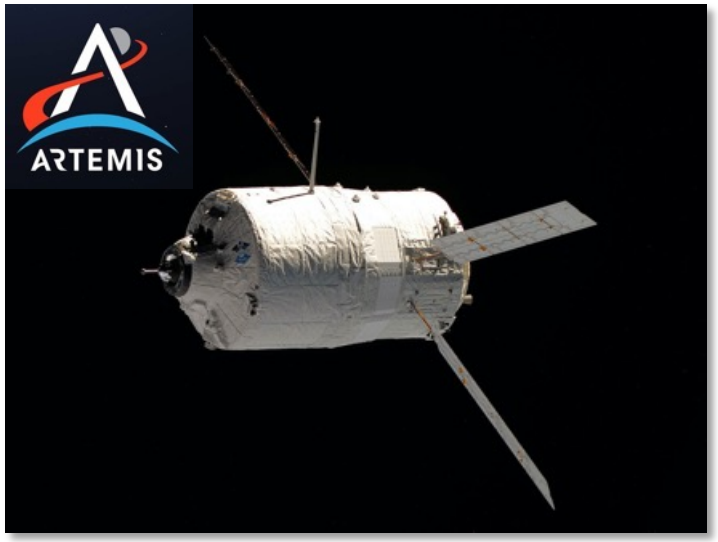
- Purpose:
  - Platform training, communication skills, joint and combined exercises, mission rehearsal, etc
- Federates:
  - Platform simulators, radio simulations, sensor simulations, visualizers/viewers, data loggers
- Sample FOMs:
  - Platforms, Warfare, Radio, Tactical Data Links (Link 16), Weather, Radar, Jamming, IFF, Enumerations (platform types), laser engagement
- Examples:
  - UK Royal Air Force “Gladiator”
  - US Navy Continuous Training Environment

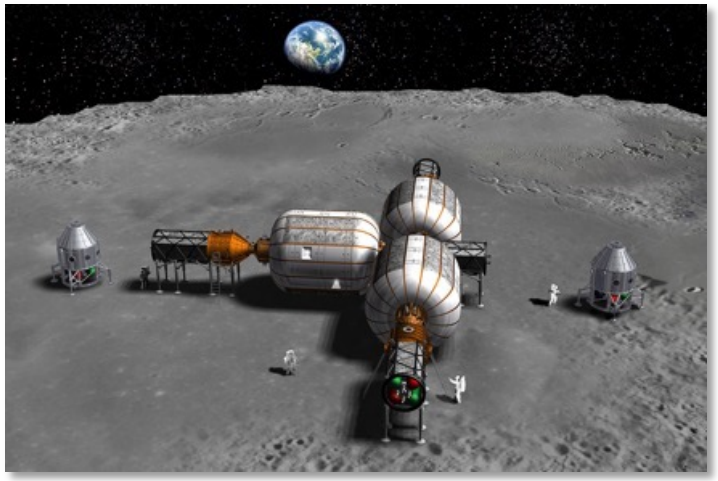
HLA for Command and Control Training

- Purpose:
  - Command post exercises, leadership and decision training, joint and multinational exercises, tactics development
- Federates:
  - Aggregate level simulations, interfaces to C2 systems, radio simulations, sensor/tracker simulations, visualizers/viewers, data loggers
- Sample FOMs and standards:
  - Aggregates, Order of battle, Platforms, Warfare, Radio, Tactical Data Links (Link 16), AIS, Weather, Cyber
  - Military scenario description language
- Examples:
  - Viking, US-Swedish lead training with 30 nations and NATO, United Nations and NGO participants

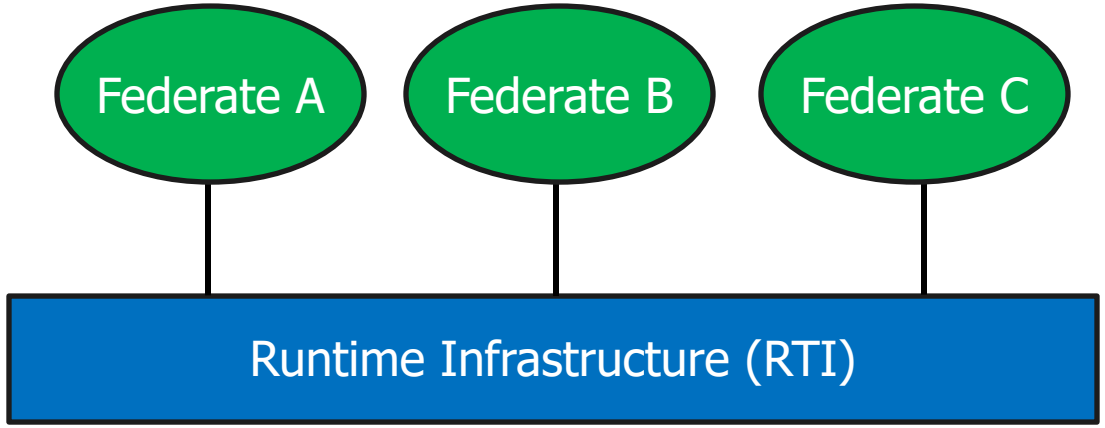
HLA for Space Simulation

- Purpose:
  - Simulation for concept, design, analysis, production, testing, training and ultimately flight.
  - Simulate in any reference frame/coordinate system, not only geocentric systems
- Federates:
  - Space vehicles, space environment, sensors, navigation
- Sample FOMs:
  - SpaceFOM, Radio, Environment, Navigation
- Examples:
  - NASA Artemis ”Return to the Moon” with simulations from major contractors
  - ESA Harwell Robotics Lab, VV&A
  - Simulation Exploration Experience (Academic)





HLA Services



Federation Object Model

<FOM>
<Shared object classes>
<Shared interaction classes>
<More>
</FOM>

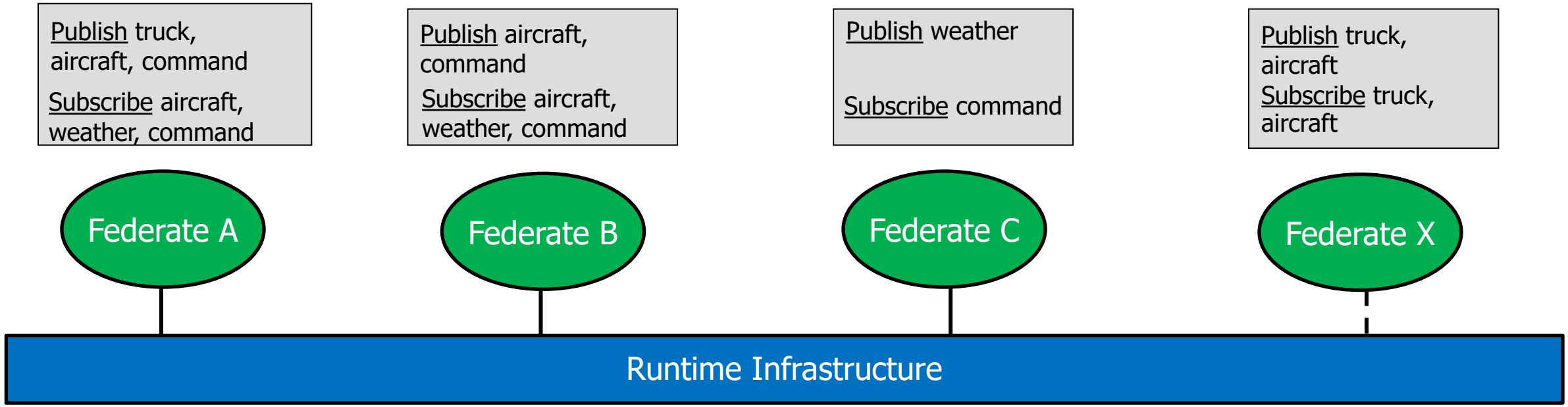
- The RTI provides simulation services to the federates:
  - Connect and join simulators together
  - Declare information interest and intent with publish & subscribe
  - Exchange information about entities and events
  - Transfer modeling responsibility between simulators
  - Run in faster or slower than real-time, or run as fast as possible
  - Handle large scale scenarios with filtering, e.g. geographical filtering

Objects vs. Interactions

The HLA shares data between federates via two mechanisms:

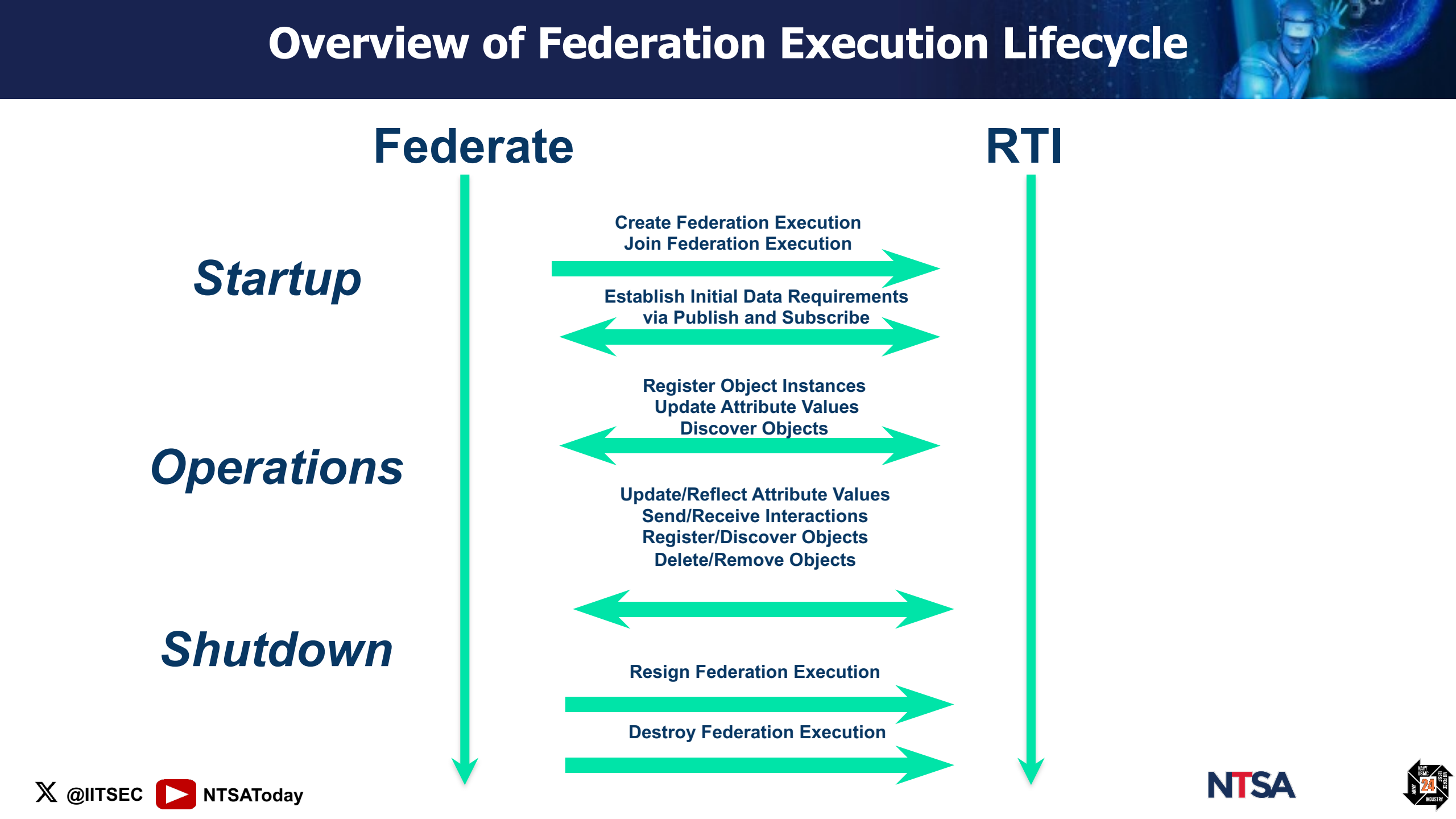
- Objects for entities
  - Persistent
  - Usually represent long-lived entities in the federation
  - State is represented by attributes
  - State can be partially updated by owning federate
  - Example: Aircraft, Human
- Interactions for events
  - Not persistent
  - Usually represent events in the federation
  - State is represented by parameters
  - Full state must be sent in the interaction
  - Example: Fire, Radio signal

HLA uses the Publish/Subscribe Pattern



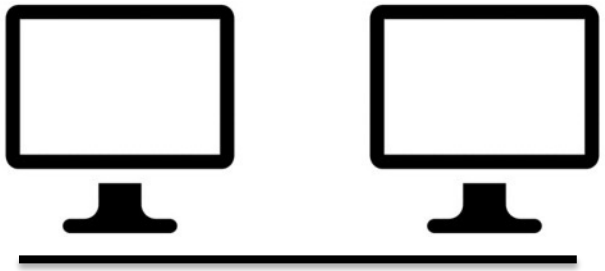
- A federate can publish the information that it produces to the federation
- Another federate can subscribe to information that it requires.
  - Publishing and subscribing is based on the Federation Object Model (FOM)
- The RTI routes any relevant information to a subscribing federate – no need for federates to connect directly to other federates.
  - Allows for multiple RTI implementations, including central server and peer-to-peer
- Enables interoperability and reuse

Overview of Federation Execution Lifecycle



Federation Management

- Purpose: Establish well-managed sessions with a controlled set of simulations.
- Functionality: Creating federation executions. Coordinated start-up. Synchronization points. Check pointing.
- Essential HLA Services
  - Connect to the RTI, create and join federation execution.
  - Resign. Destroy federation execution. Disconnect.
  - Synchronization points.
  - Save/restore.

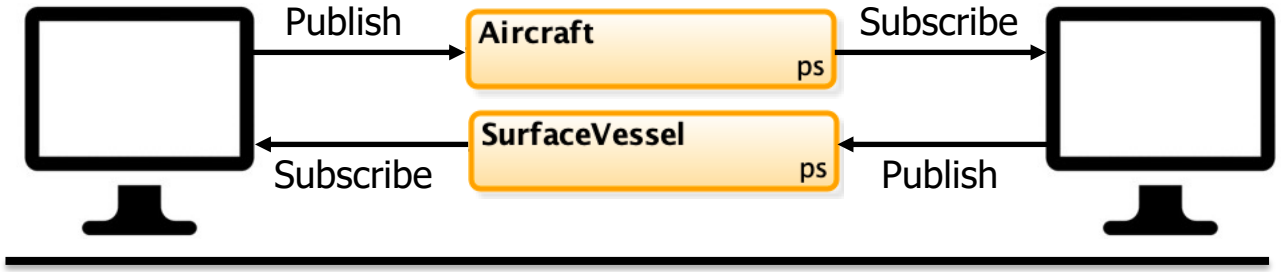


Declaration Management

- Purpose: Declare what data each federate will send or wishes to receive, based on the federation object model (FOM), in order to optimize the data exchange.

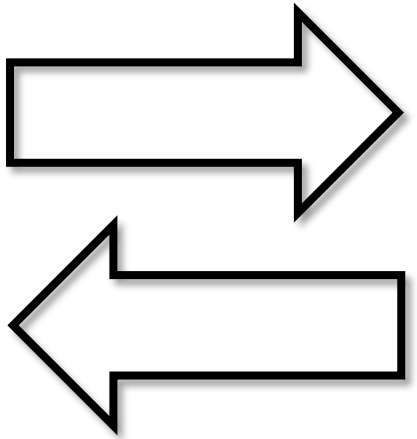
- Functionality: Publish and subscribe object classes with attributes, and interaction classes with parameters.

- Essential HLA Services
  - Publish.
  - Subscribe.



Object Management

- Purpose: Data exchange.
- Functionality: Create and delete object instances that are shared between federates. Update their attributes. Send and receive interactions.
- Essential HLA Services
  - Register, discover, delete and remove object instances.
  - Update and reflect attribute values.
  - Send and receive interactions.
  - Request updates.

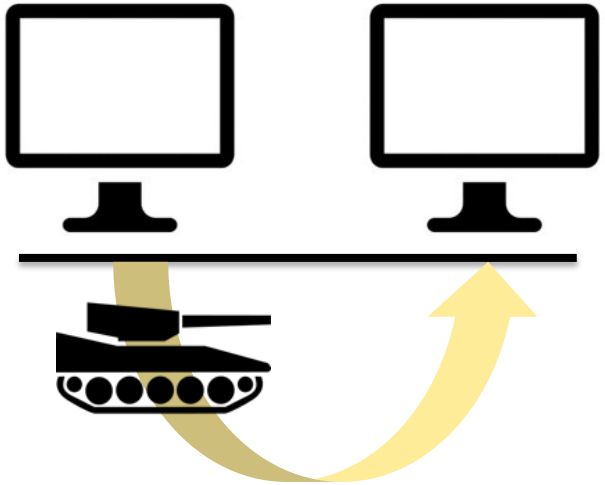


Ownership Management

- Sample use case: enable a constructive simulation to hand over the responsibility for simulating an aircraft to a virtual simulation.

- Functionality: Hand over the ownership (responsibility for updating) of attributes from one federate to another during execution. This can be either “push” or “pull,” and with or without negotiation.

- Essential HLA Services:
  - Request ownership acquisition for an attribute.
  - Divest ownership for an attribute.
  - Query attribute ownership.



Time Management

- Sample use case: run exercises in real-time, scaled real-time, varying speed, as-fast-as-possible. Run Monte-Carlo simulations as fast as possible.

- Functionality: Manage the time advancement for federates and time-stamped data exchange between federates for consistency and repeatability. This can be time-stepped or event-driven.

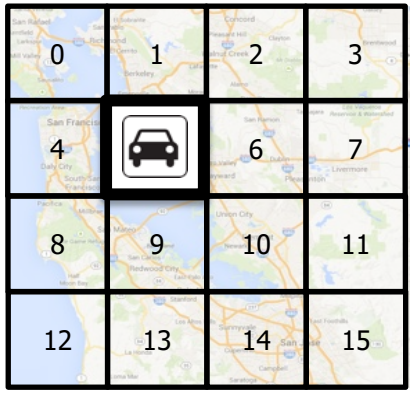
- Essential HLA Services:
  - Enable/disable sending and receiving of time-stamped data.
  - Request and grant advancement to a given time.
  - Request and grant advancement to the time of the next message.
  - Query the RTI for current time values.

Data Distribution Management

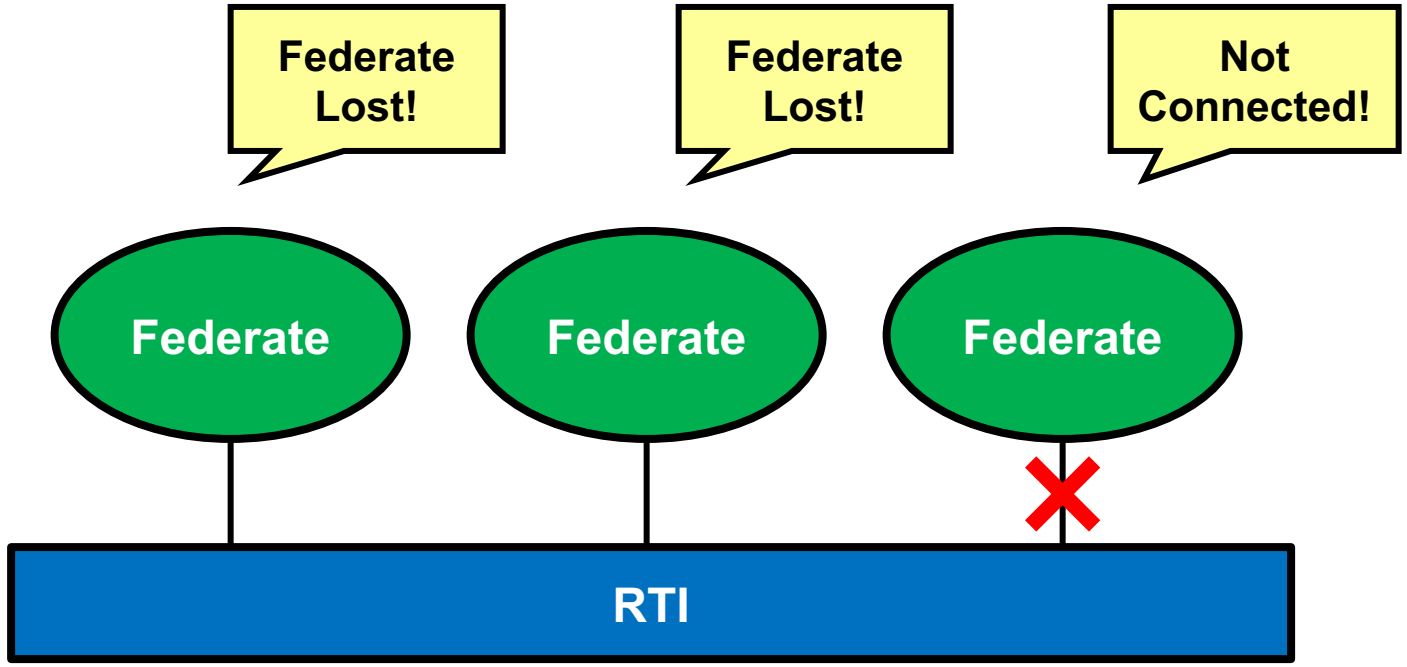
- Sample use case: build large federations by enabling federates to focus on a smaller part of the battlefield or on entities belonging to a particular force.

- Functionality: Enable federates to filter data not only on class and attribute, but also on their static or dynamic data (like friendly/opposing or latitude/longitude).

- Essential HLA Services:
  - Create and modify regions that specify data ranges for filtering.
  - Send attribute updates and interactions with associated regions.
  - Subscribe to attributes and interactions with associated regions.



HLA Fault Tolerance



- HLA includes a mechanism for fault tolerance
  - Connected federates can discover that a federate has been lost
  - The lost federate can discover that it has lost connection
- Enables several design patterns (fail-over federates, reconnecting federates, …)

Object Model Template – Minimal Version

- Identification
  - Name, version, purpose, etc.
- Object Classes with Attributes
- Interaction Classes with Parameters
- Data Types
  - Simple, records, enumerations, etc.
  - Used for Attributes and Parameters

Object Class

Aircraft
- Type
- Marking
- Position

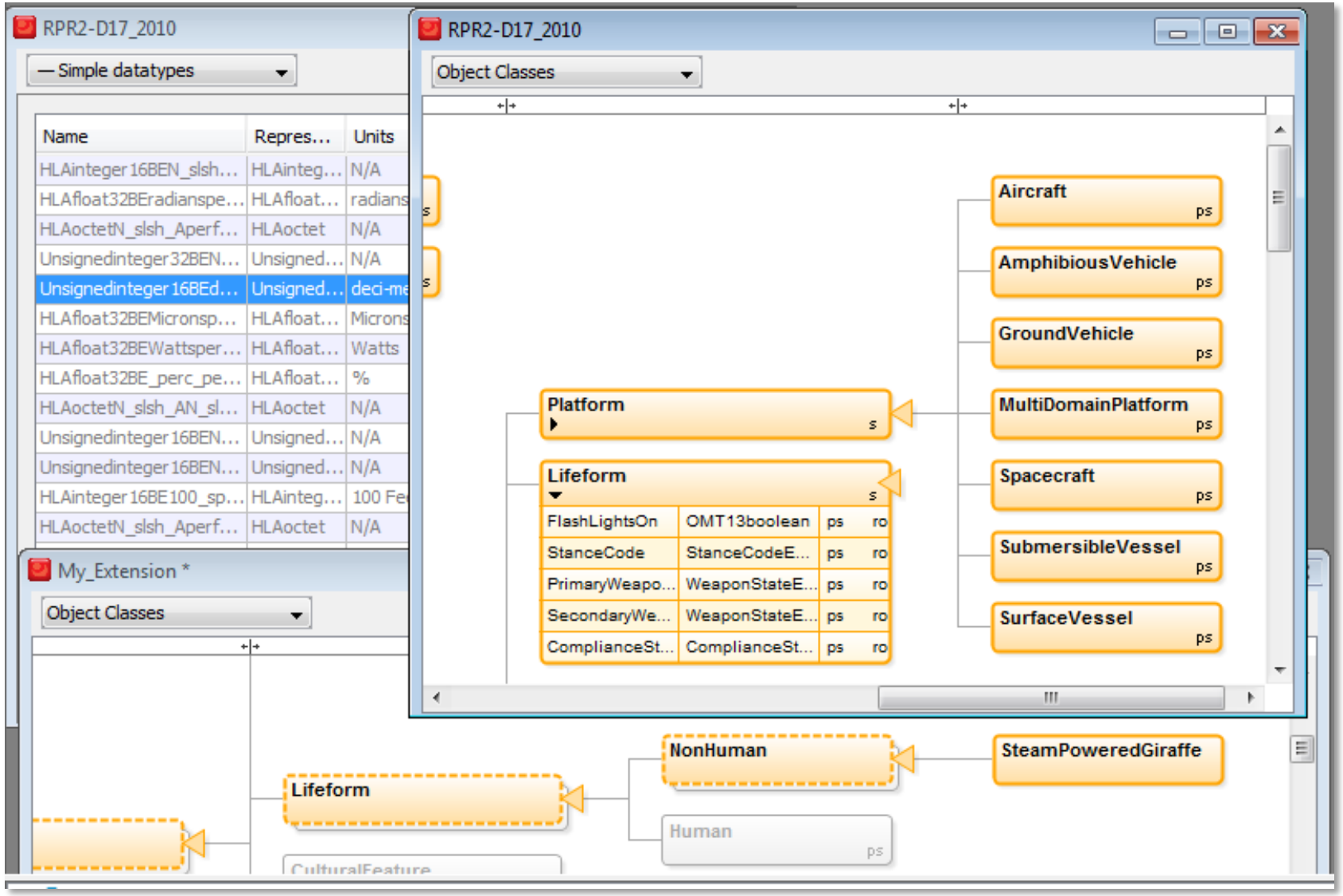
Interaction Class

MunitionFire
- FiringObject
- TargetObject
- MunitionType

Data type

PositionRecord
- Latitude
- Longitude
- Altitude

Object Class Structure Table Example

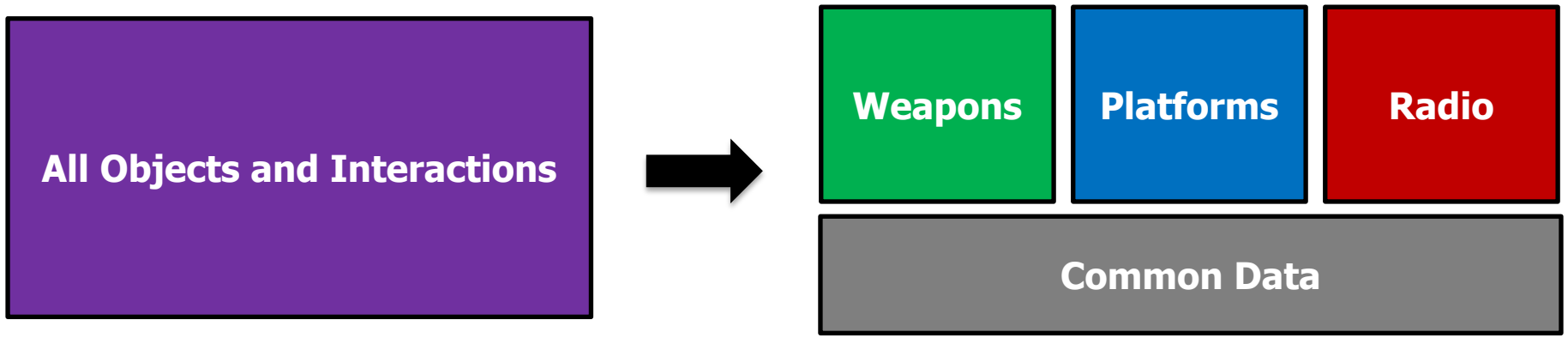


A hierarchical visual representation

Object Model Template – Full Version

- Object model identification table
- Object class structure table
- Interaction class structure table
- Attribute table
- Parameter table
- Dimension table
- Time representation table
- User-supplied tag table
- Synchronization table
- Transportation type table
- Update rate table
- Switches table
- Datatype tables
- Notes table

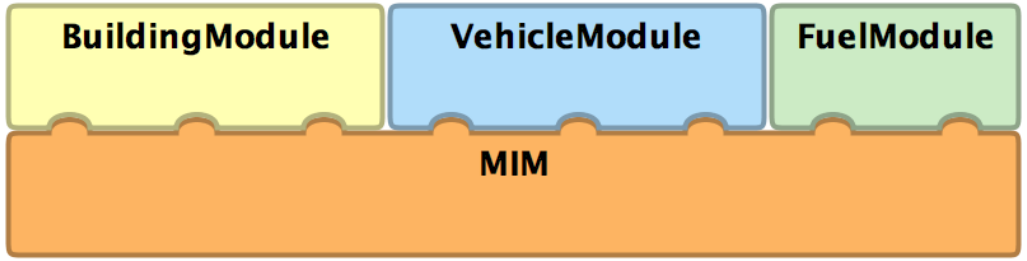
FOM Modules



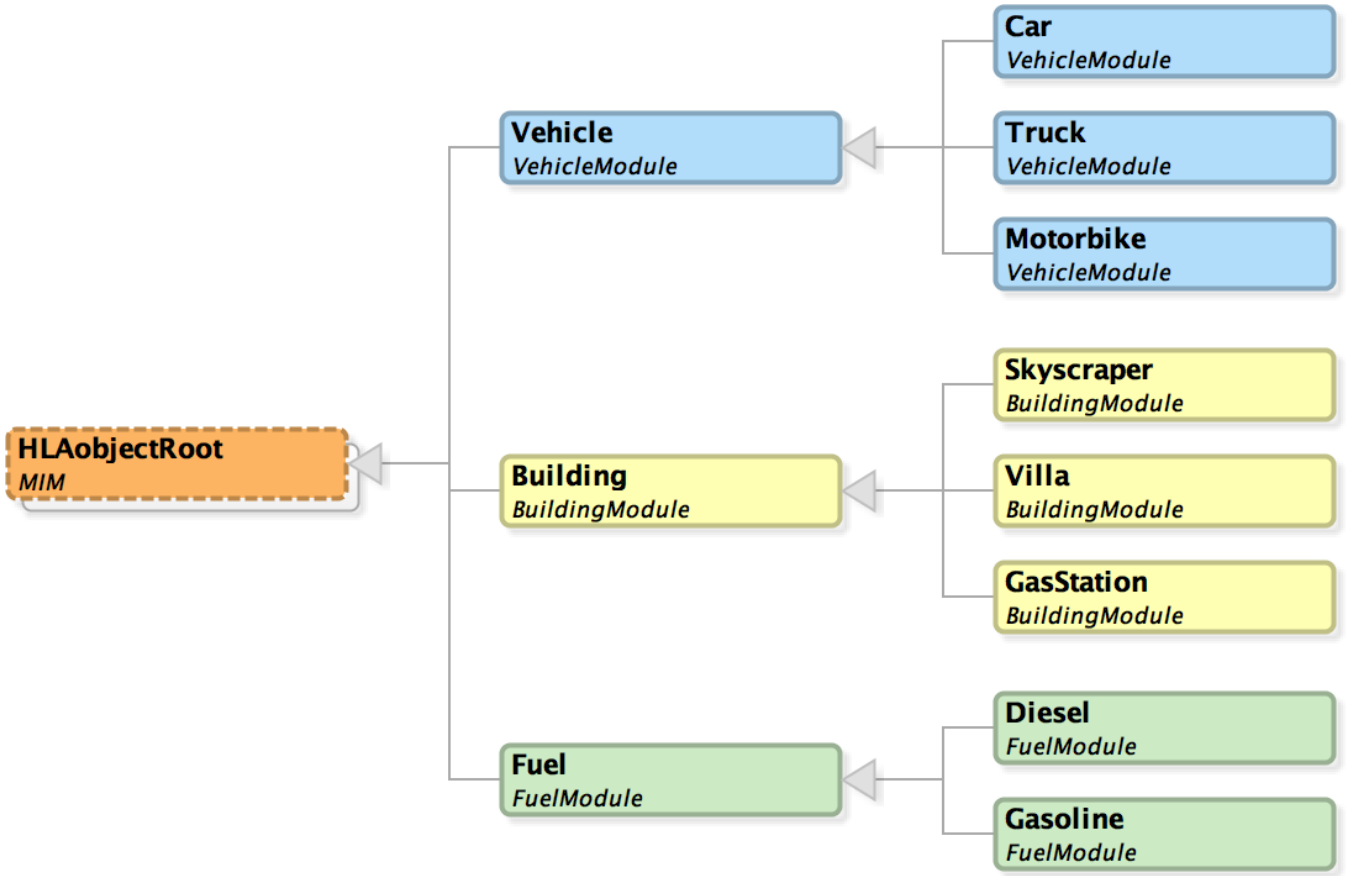
- The same FOM, but split into modules
  - Remember: It was a long time ago since we gave up developing computer programs as one large chunk of code.
- More efficient and flexible development, maintenance, reuse and standardization
- Projects/programs can reuse and extend standardized modules and add their own modules.

FOM Module Example

Modules



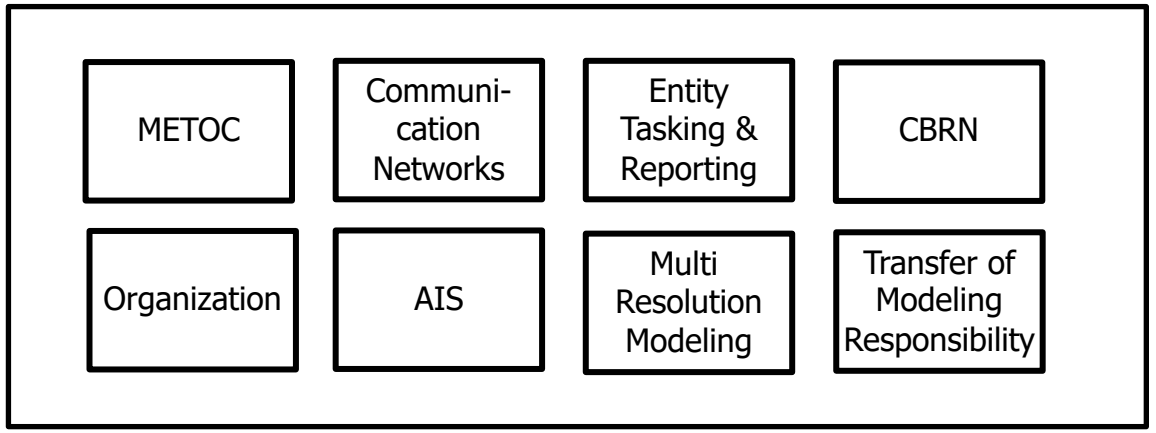
Object Classes



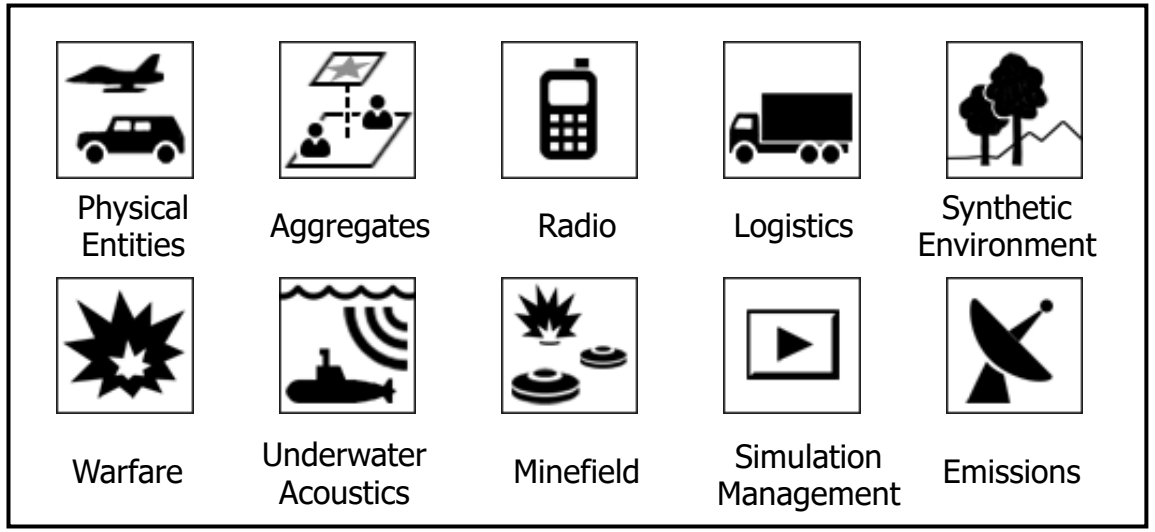
Common Aerospace and Defense FOMs

- The RPR FOM 2.0 is the most commonly used FOM for defense simulation
  - RPR FOM 3 now being finalized
  - IFF, InfOps, etc.
  - Download from www.sisostds.org
- NATO Education and Training Network FOM (NETN)
  - Builds on top of RPR FOM
  - Growing number of modules
  - Download from github.com/AMSP-04/
- SISO Space Reference FOM
- EUROCAE Air Traffic Management FOM
- Medical FOM (US Army)

NATO NETN FOM Modules

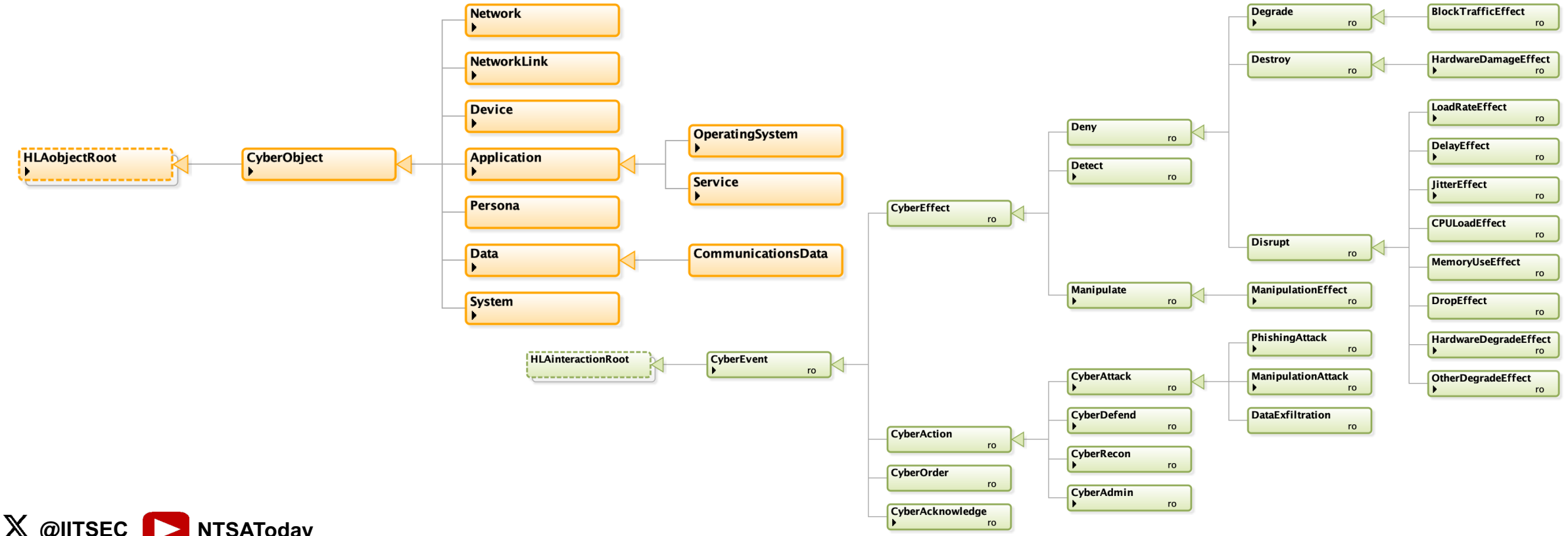


RPR FOM Modules



Cyber Data Exchange Model (DEM) and Federation Object Model (FOM)

- SISO is developing a Cyber FOM standard to represent cyber events and objects.
- The Cyber FOM will enable integration of cyber simulations, cyber ranges, and kinetic simulations.



Important New Features in HLA 4

HLA 4

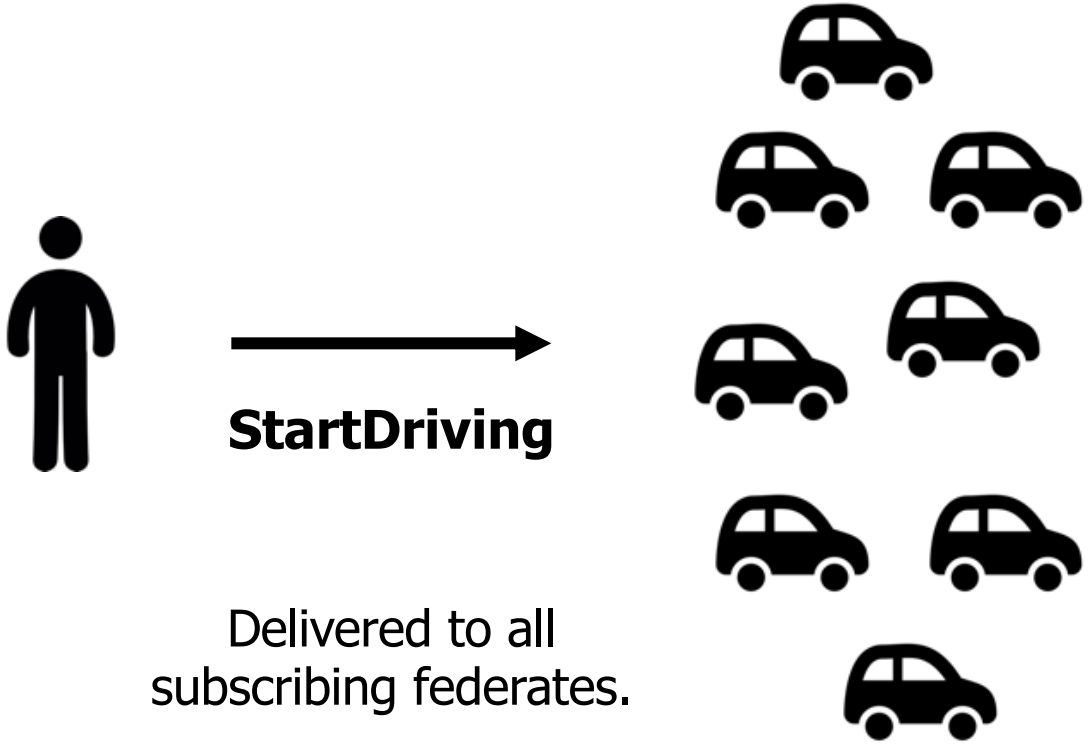
- More powerful modeling
  - Directed Interactions
  - Easier to extend reference FOMs
- Security
  - Secure authorization of federates
  - Secure communication
- Deployment
  - Support unreliable links, like 3G/4G/5G
  - Support more programming languages
  - Increased vendor-independence
- Scalability
  - Cloud-based simulation with local deployment
  - “Elastic cloud” scalability
  - Improved data distribution management
- Business models
  - Modeling and Simulation as a Service

HLA 4 passed IEEE Balloting in 2023

Object Modeling: Directed Interactions

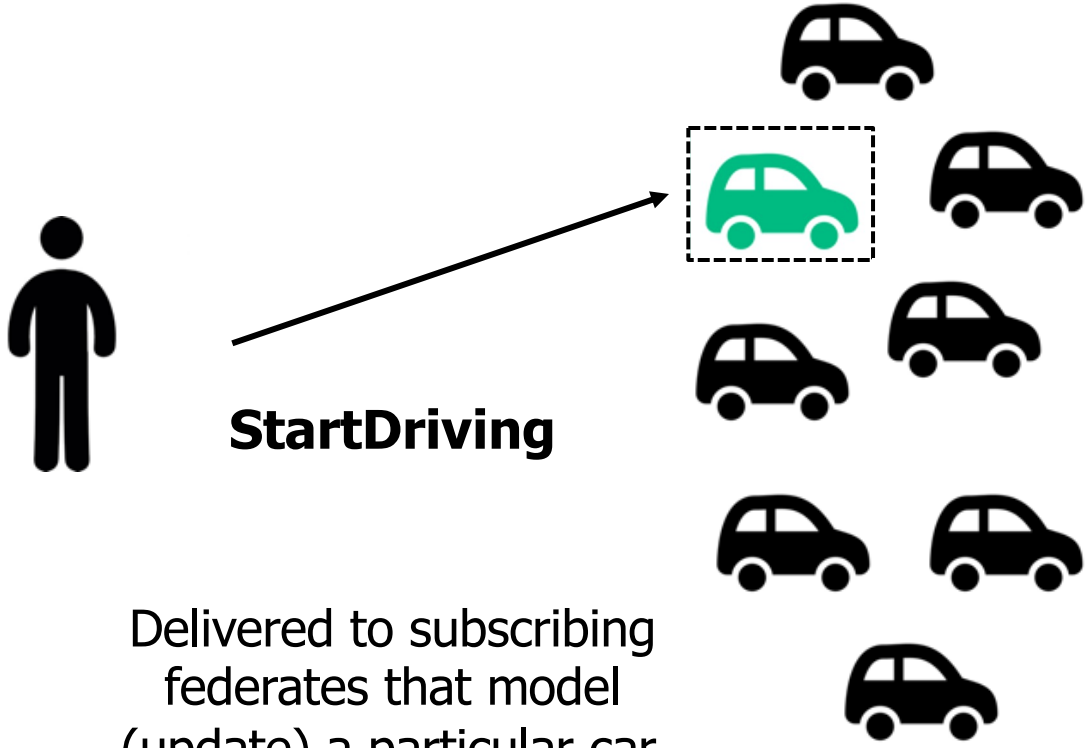
HLA 4

Classic "Global" Interactions



Delivered to all subscribing federates.

Directed Interactions



Delivered to subscribing federates that model (update) a particular car.

Object Modeling: Easier to Extend Models

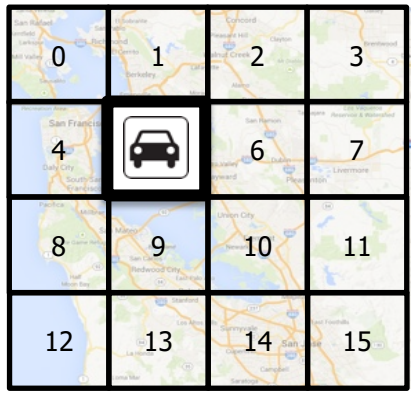
HLA 4

- Standardized “Reference Object Models” are one of the key benefits with HLA. They provide a baseline for interoperability. They are key for reuse.
- HLA 4 adds several features that makes it easier to extend existing information models with application-specific data.
- Example: add attributes to already defined object classes

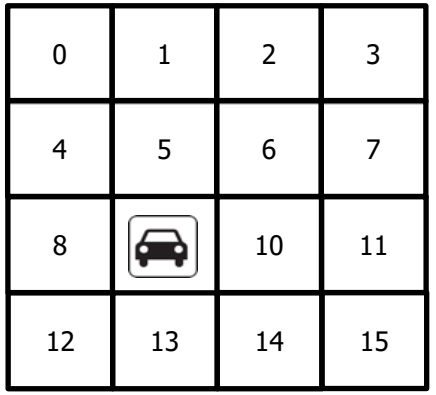
Standard Aircraft
Entity Type
Entity ID
Side
Spatial (position etc.)
Damage State

Extensions
Maintenance Date
Maintenance Type

Scalability: Improved DDM



Filter on Lat/Long

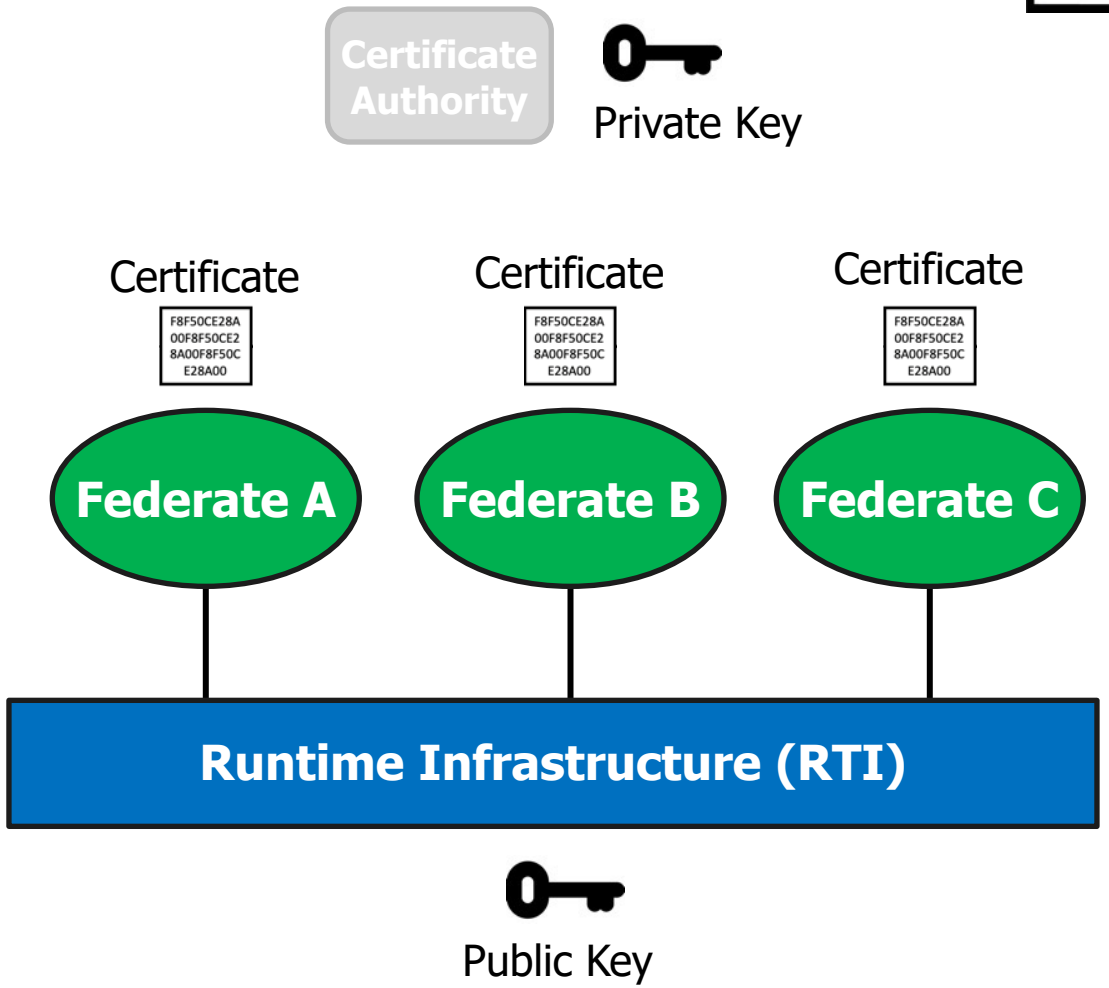


Filter on air, land, sea, space
Filter on friendly, opposing, neutral

- Modern RTIs can deliver a million updates/second to a federate, but few, if any federates, can process that incoming data rate
- Reducing the data flow to each federate is the key to scalability
- HLA Data Distribution Management (DDM) provides powerful ways to filter data
- HLA 4 makes it possible to extend reference FOMs with tailored DDM

Federate Authentication

- Federates may be required to provide credentials to connect and join a federation
- Support for passwords, PKI certificates (X.509), etc.
- Flexible mechanism through plug-in
- May connect to existing authorization services
- May implement different policies
- Also useful for Zero Trust Architectures

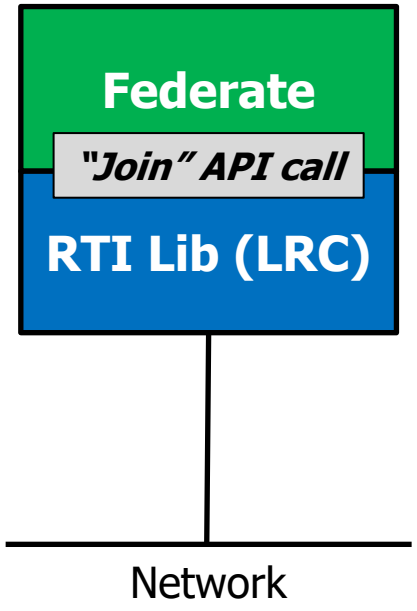


Federate Protocol

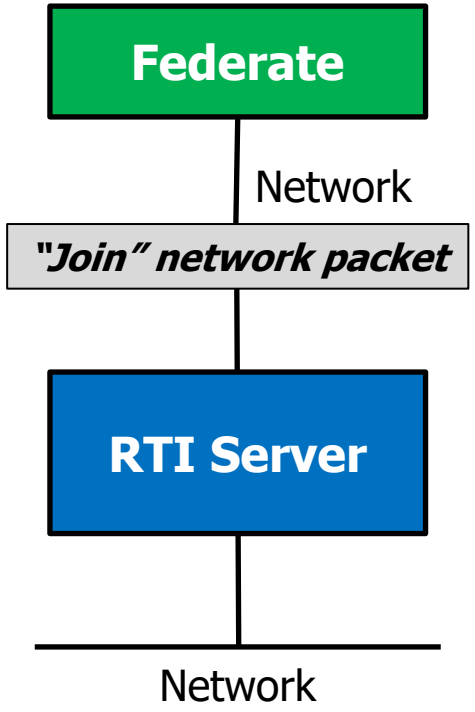
- A federate can call the RTI using a protocol instead of calling a local library
- Separates the federate from the RTI
- Better support for Live with 4G/5G links
- More cloud-friendly (containers, etc.)
- Simplifies accreditation
- Additional languages/OS

HLA 4

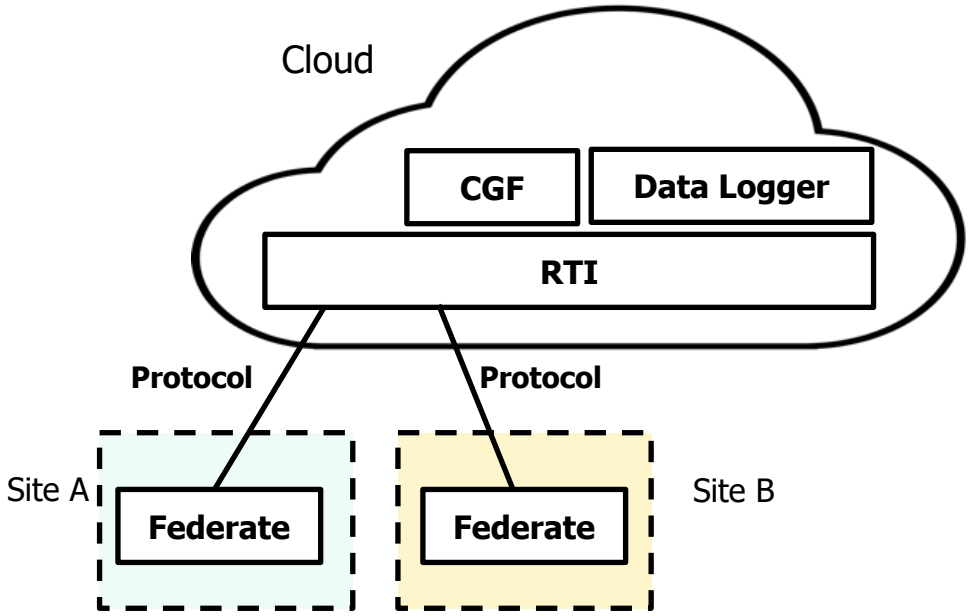
Standardized API

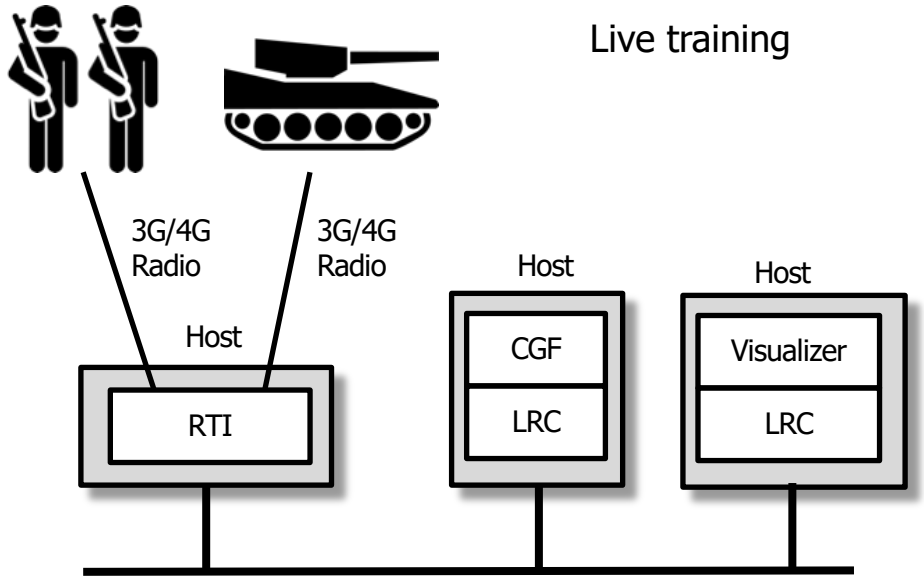


Standardized protocol



Deployment Examples: Cloud and Live Training

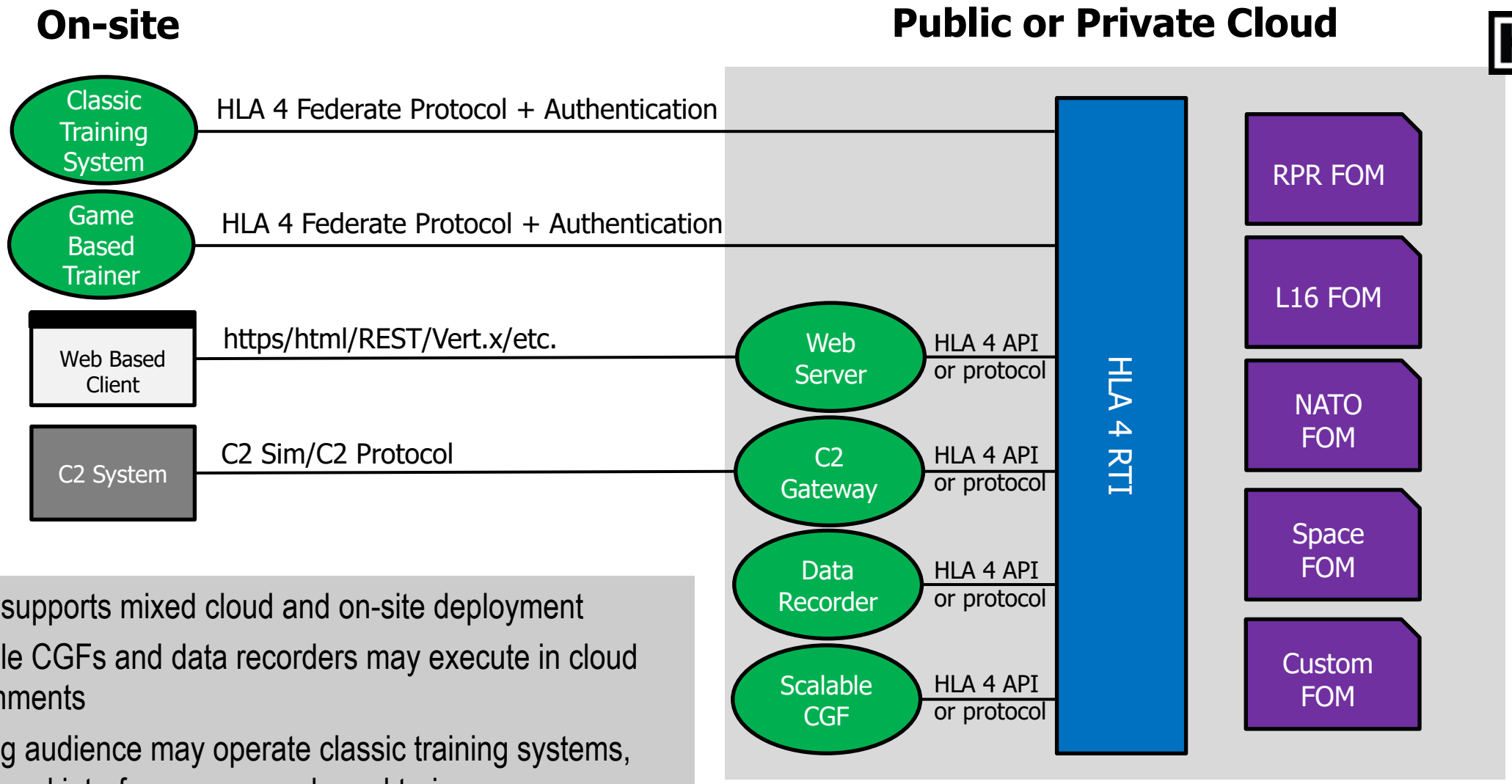




- Deploy large CPU consumers in the Cloud
- Deploy large GPU consumers locally
- Collect Big Data in the Cloud

- Connect live systems over 4G/5G
- Point-to-point communication
- Fault tolerance and fast reconnect

HLA 4 Based Architecture for Cloud, Scalability and MSaaS



- HLA 4 supports mixed cloud and on-site deployment
- Scalable CGFs and data recorders may execute in cloud environments
- Training audience may operate classic training systems, web-based interfaces or game-based trainers

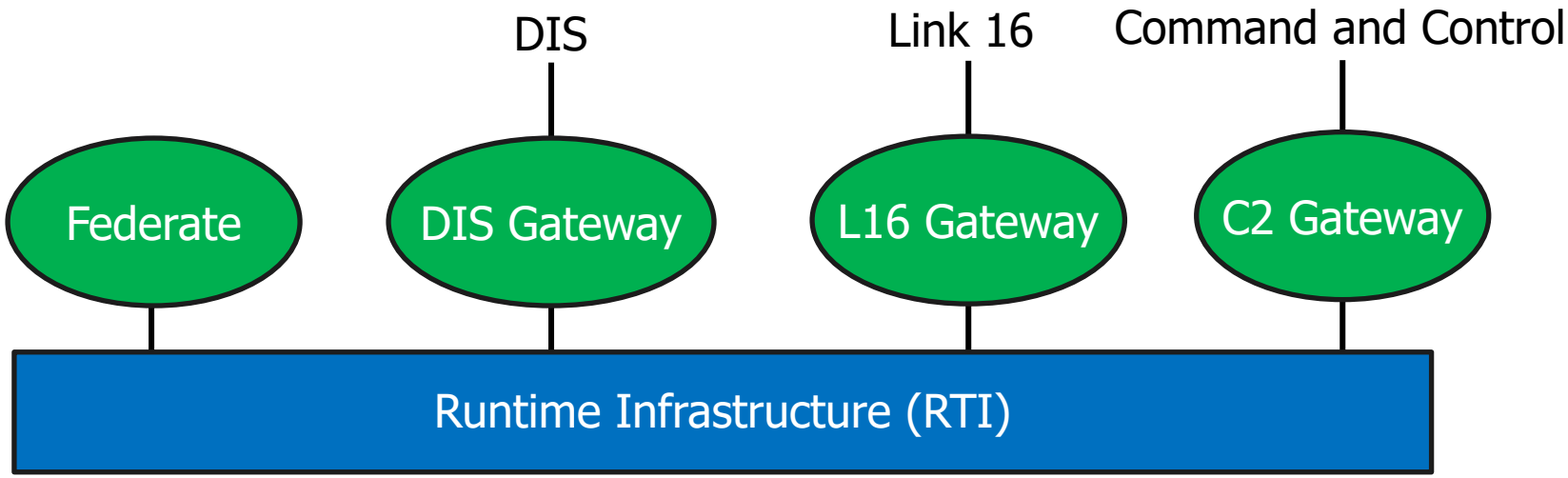
More New Features in HLA 4

HLA 4

- Better support for scalability through enhanced DDM
  - Easier to specify dimensions for entire object class, inheritance to subclasses
  - Add DDM to existing FOMs using separate module
  - More complete and flexible Dimension specification
  - Relaxed DDM

- Many minor improvements
  - Ownership management – user supplied tags
  - Specify for what attributes values are required vs optional
  - Fault tolerance – handling unplanned leaving federate
  - New datatypes to better support RPR FOM
  - Glyph that follows html conventions
  - Switches – defaults
  - Exception handling
  - Updated sample FOM modules

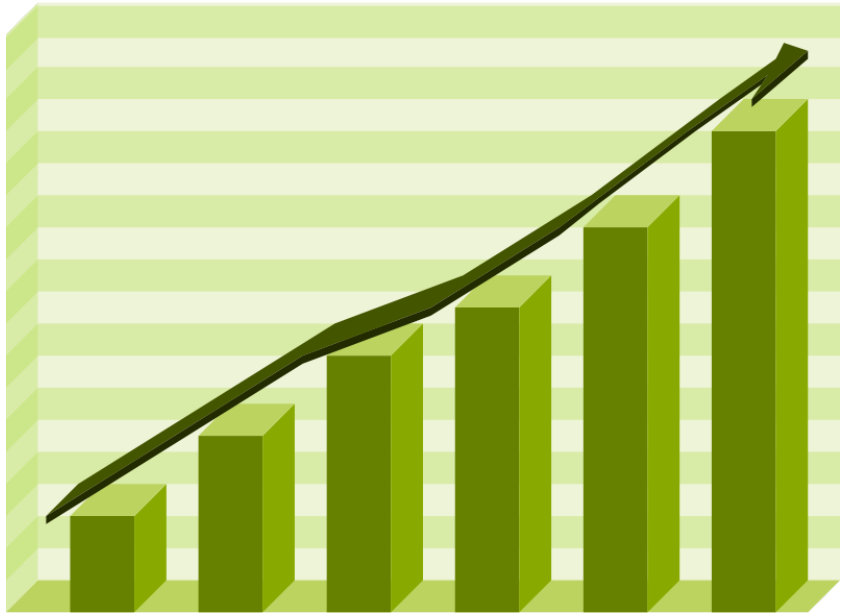
HLA and LVC – Gateways



- HLA can handle both Live, Virtual and Constructive simulation
- In LVC settings, information may still need to be bridged between different architectures
- Standard FOMs exist for bridging
  - The RPR FOM matches the DIS information model
  - Tactical Data Link (TDL) such as Link 16
  - C2SIM – connect to C2 systems
- Custom gateways may be needed

HLA and Performance

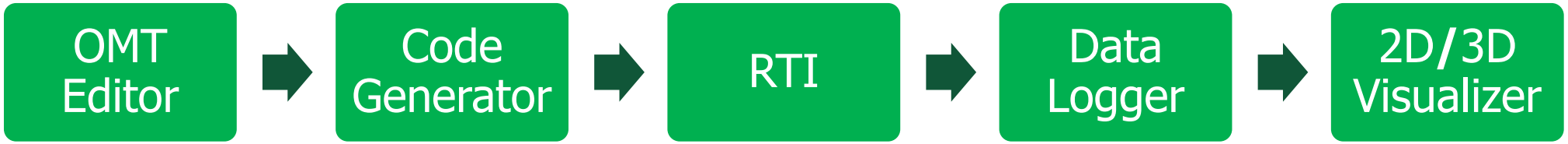
- HLA is a standard
  - It has no particular performance
- Actual RTI implementations exhibit performance
  - May be optimized for different use cases
- Sample RTI performance on regular Windows workstations in 2020
  - Hundreds of systems
  - 100 000's of updates per second, typically up to 90% of the network's raw bandwidth
    - 900 Mbps on Gigabit LAN
  - 100 000 object instances
  - Latency of 0.13 ms on 100 Mbps LAN



- Common federation bottlenecks:
  - LAN bandwidth
  - WAN latency and bandwidth
  - Federate processing time for incoming data
- Plan for performance!
  - Analyze performance requirements
  - Analyze federate and network capabilities

HLA and Tools

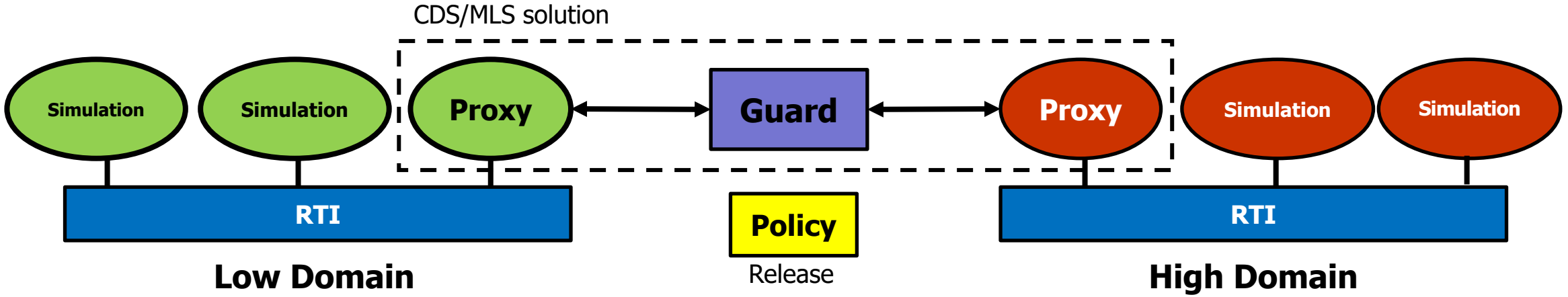
Sample tool chain



- A wide range of HLA compliant tools and federates are available
- Development tools (OMT editor, middleware, code generators, data loggers)
- Visualizers (2D, 3D, game based, …)
- Federates (Computer Generated Forces, flight models, dynamics, …)
- Bridges (HLA-DIS, HLA-HLA, HLA-TENA, HLA-games, HLA-DDS, …)

TENA = Test and Training Enabling Architecture
DDS = Data Distribution Services

HLA and Cross Domain Security/MLS



- It is possible to simulate with controlled data exchange between classification levels
- The federation is partitioned into different domains
  - May be different classifications within one nation or simulation between different nations
- There are security solutions that can connect to multiple domains, provide separation and release data based on a release policy
  - Different topologies: one host, several hosts, hardware/software guards, etc.
  - Different solutions meet different levels of assurance

HLA Benefits

Secure

Secure authentication

Secure transportation

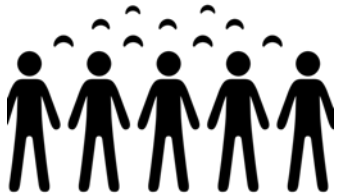
Cross-domain security

Proven

Viking, the largest CiMiC exercise

UK Royal Air Force

Down-selected by NASA



Scalable

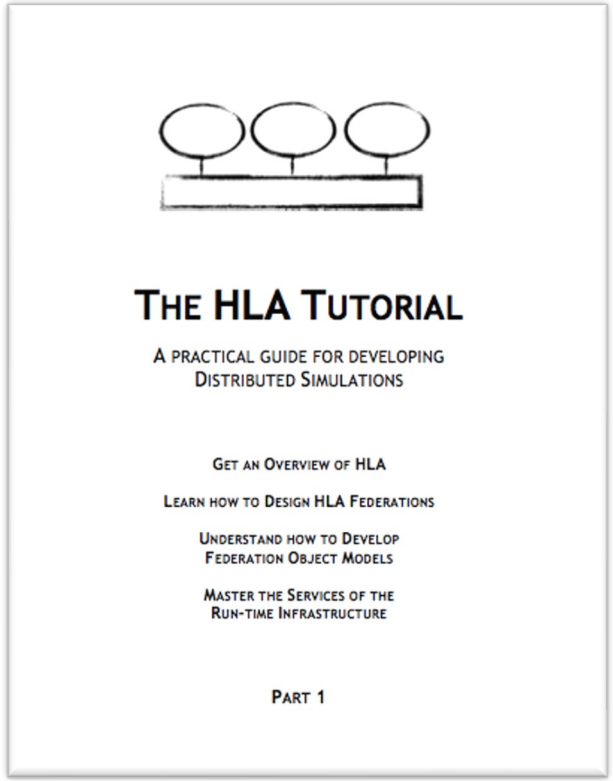
Millions of entities

Hundreds of systems

<0.1 ms latency

Additional Reading

- The standard
  - HLA IEEE 1516-2010 www.ieee.org
- HLA podcast – an introduction
  - https://www.warfighterpodcast.com/
- The HLA Tutorial – for developers
  - Free, redistributable PDF
  - www.pitchtechnologies.com/hlatutorial
- Plenty of SISO papers
  - www.sisostds.org
  - ResearchGate
- The HLA Product Development Group
  discussion archives
  - www.sisostds.org



Learning Objectives Revisited

After taking this tutorial you should be able to:

- Describe the requirements for interoperability in distributed simulation
- Explain the principles of the HLA standard
- Explain how the HLA standard is used for distributed training purposes
- Describe the new features of HLA 4
- Describe some common Federation Object Models (FOMs) for defense training

Questions?

Björn Möller Bjorn.Moller@pitch.se

Fredrik Antelius Fredrik.Antelius@pitch.se