High Level Architecture
Module 1
Basic Concepts

The Society for Computer Simulation

McLeod Institute of Simulation Sciences
California State University, Chico

Roy Crosbie
John Zenor

California State University, Chico

High Level Architecture
Module 1
Management of Time

Part 6

Types of Simulation Models

- Types of Simulation Models
  - Continuous Models
  - Discrete Models
- Types of Simulation Implementations
  - Event Driven Implementations
  - Time-Stepped Implementations
  - Real-Time Implementations
  - Scaled Real-Time Implementations
  - Non Real-Time Implementations

Time-Stepped Example: HelloWorld

- Continuous Model of a Discrete Process
- Time-Stepped Implementation
- Non Real-Time
  - No correlation with wall clock time
- Uses Events to Communicate Existence of Countries
- Original Model Built Assuming That Each Federate Starts at Time 0

Time Management Schemes

- No Time Management
  - Each Federate Advances Time at Its Own Pace
- Conservative Synchronization
  - Federates Advance Time Only When Guaranteed That No Past Data Will Be Received
- Optimistic Synchronization
  - Free to Advance Logical Time, May Have Roll-back
- Activity Scan
  - Advance Time by Mutual Agreement With Other Federates

Logical Time Restrictions

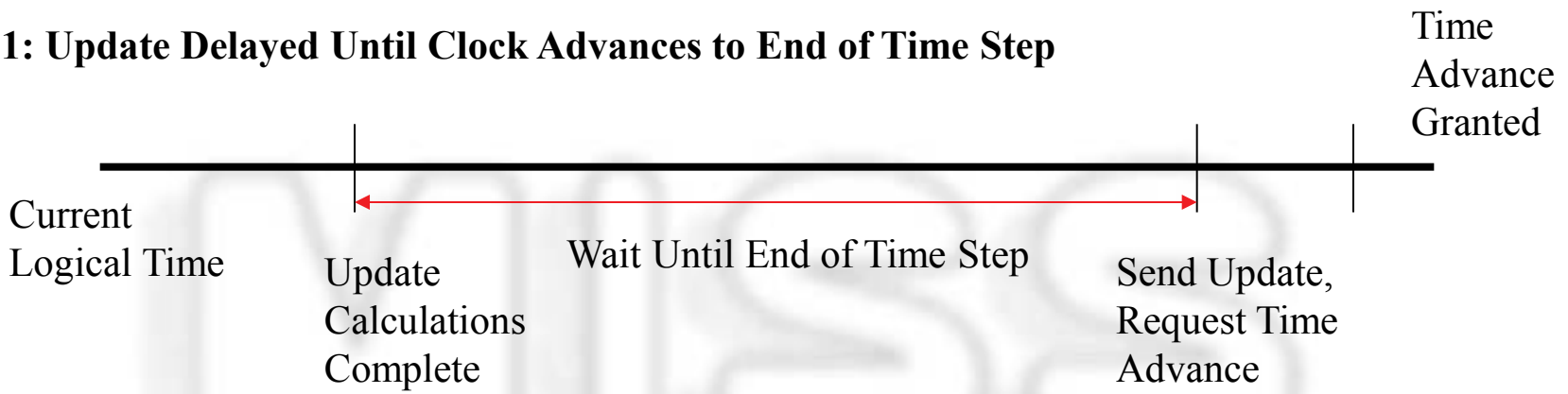
- Initial Value
- Value Not Tied to Any System of Units
- Well Ordered
- Always Greater Than (or equal to) Initial Time
- Time is Effectively Discrete
- Has Special Value of Positive Infinity that is Greater Than Any Other Value

Time-Constrained Federate

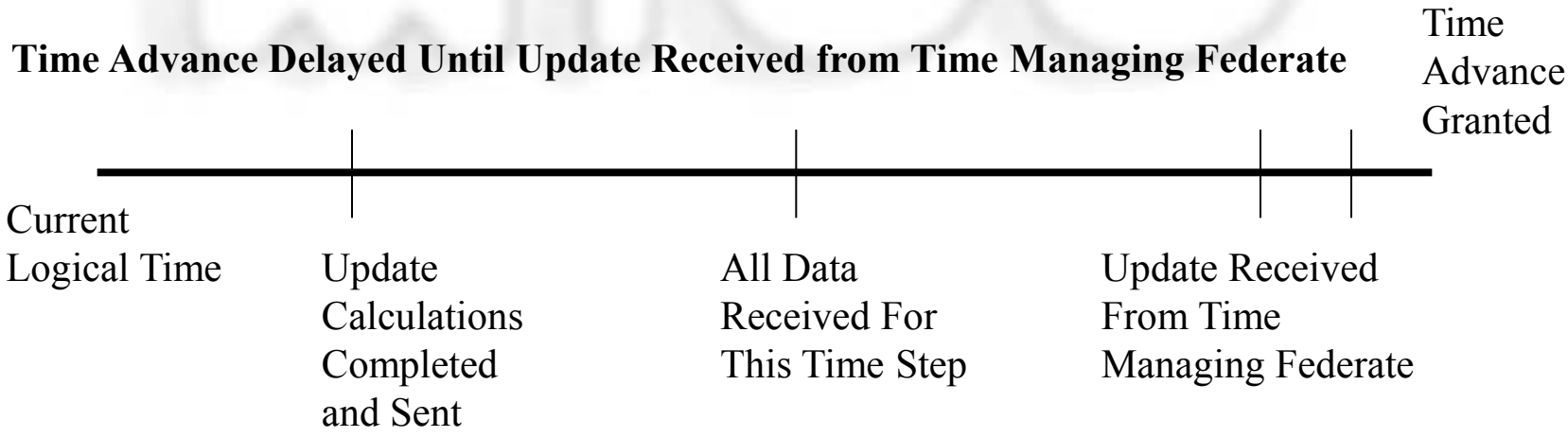
- TSO Events Delivered in Time-Stamp Order
  - Irrespective of Order in Which Originating Events are Sent
- No Event Delivered With Time-Stamp Less Than Current Logical Time for Federate

Synchronization of Logical Time to Real Time

Case 1: Update Delayed Until Clock Advances to End of Time Step



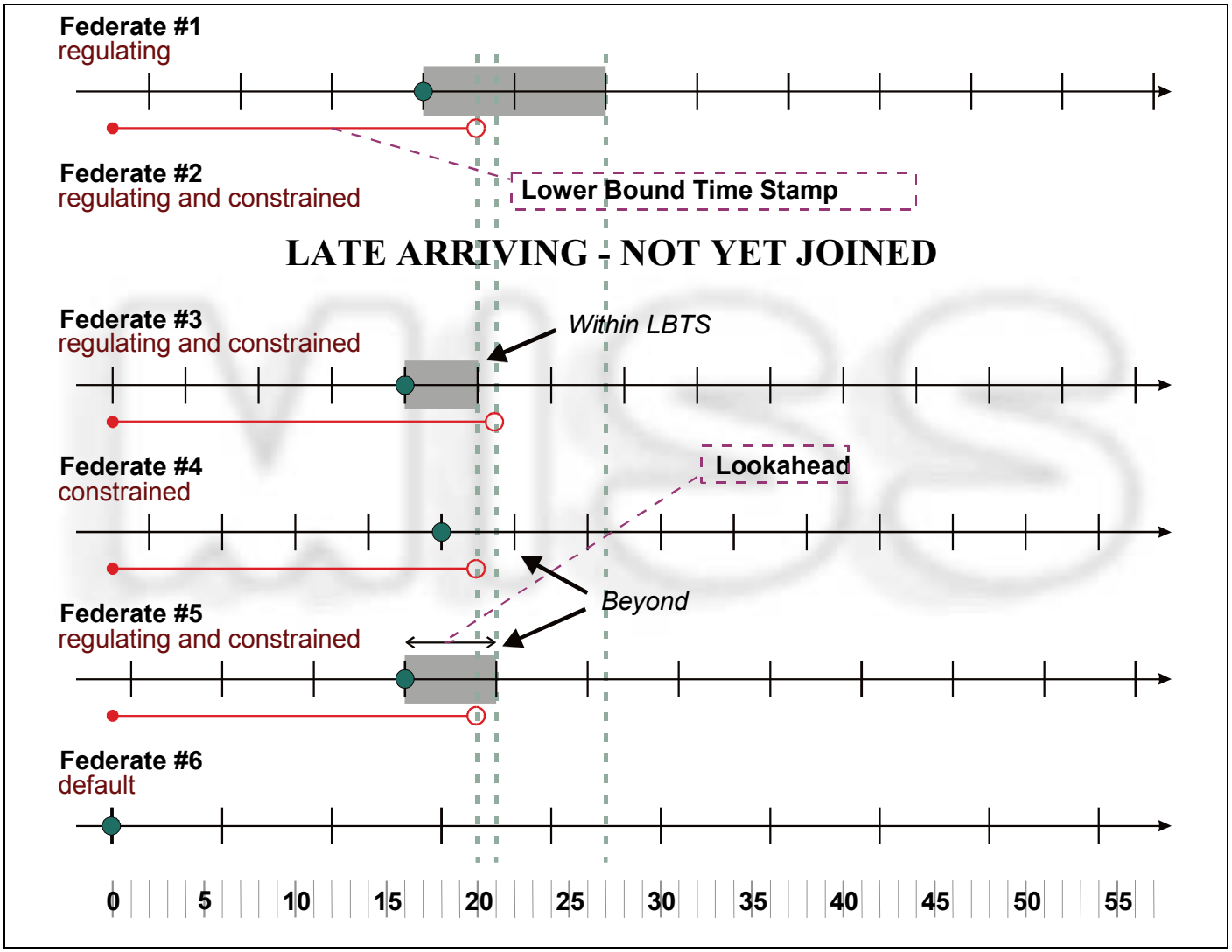
Case 2: Time Advance Delayed Until Update Received from Time Managing Federate



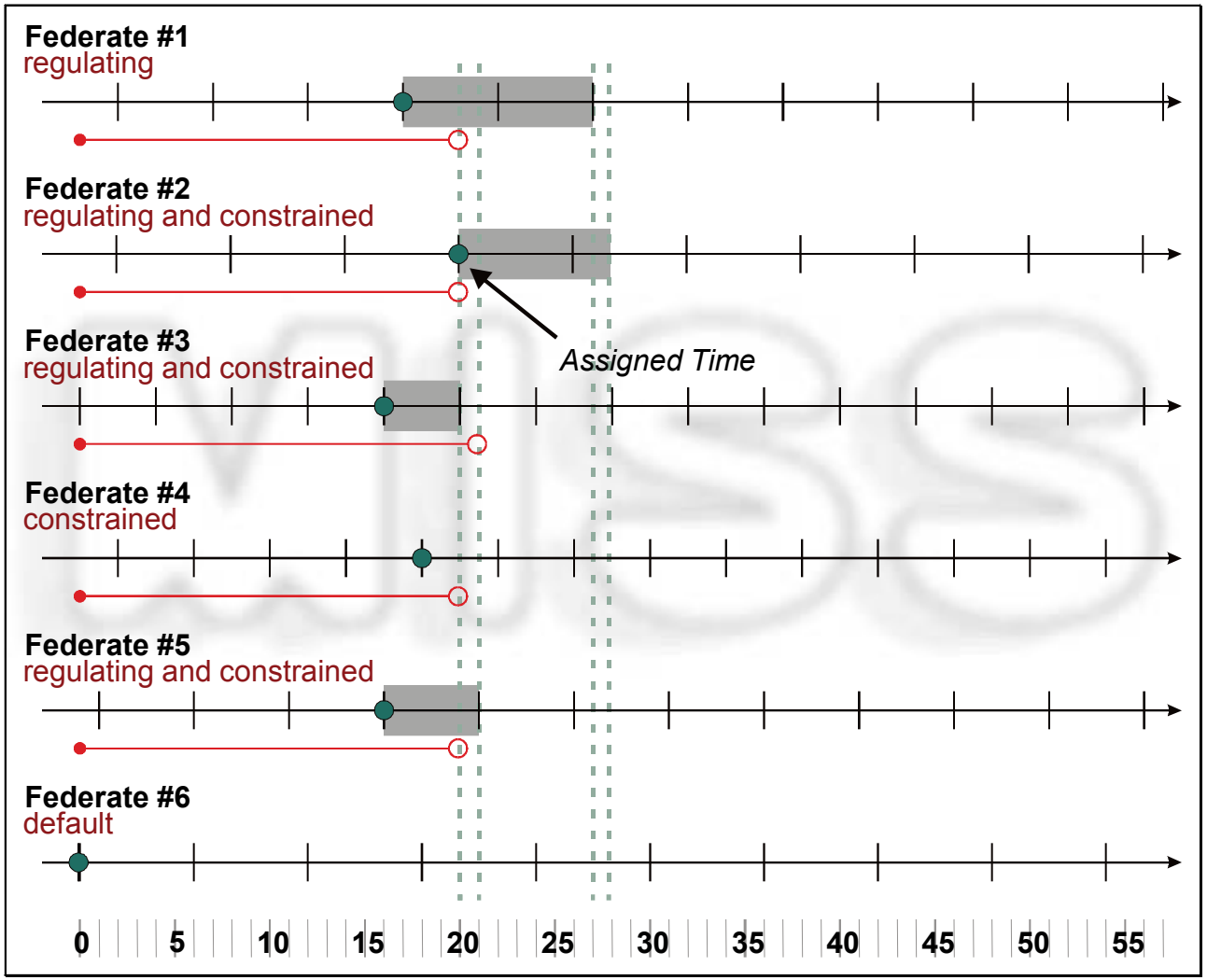
Sending and Receiving Events

- Sending Events
  - *Update Attribute Values*
  - *Send Interaction*
  - *Delete Object Instance*
- Receiving Events
  - *Reflect Attribute Values*†
  - *Reflect Interaction*†
  - *Remove Object Instance*†

Regulating and Constrained Federates



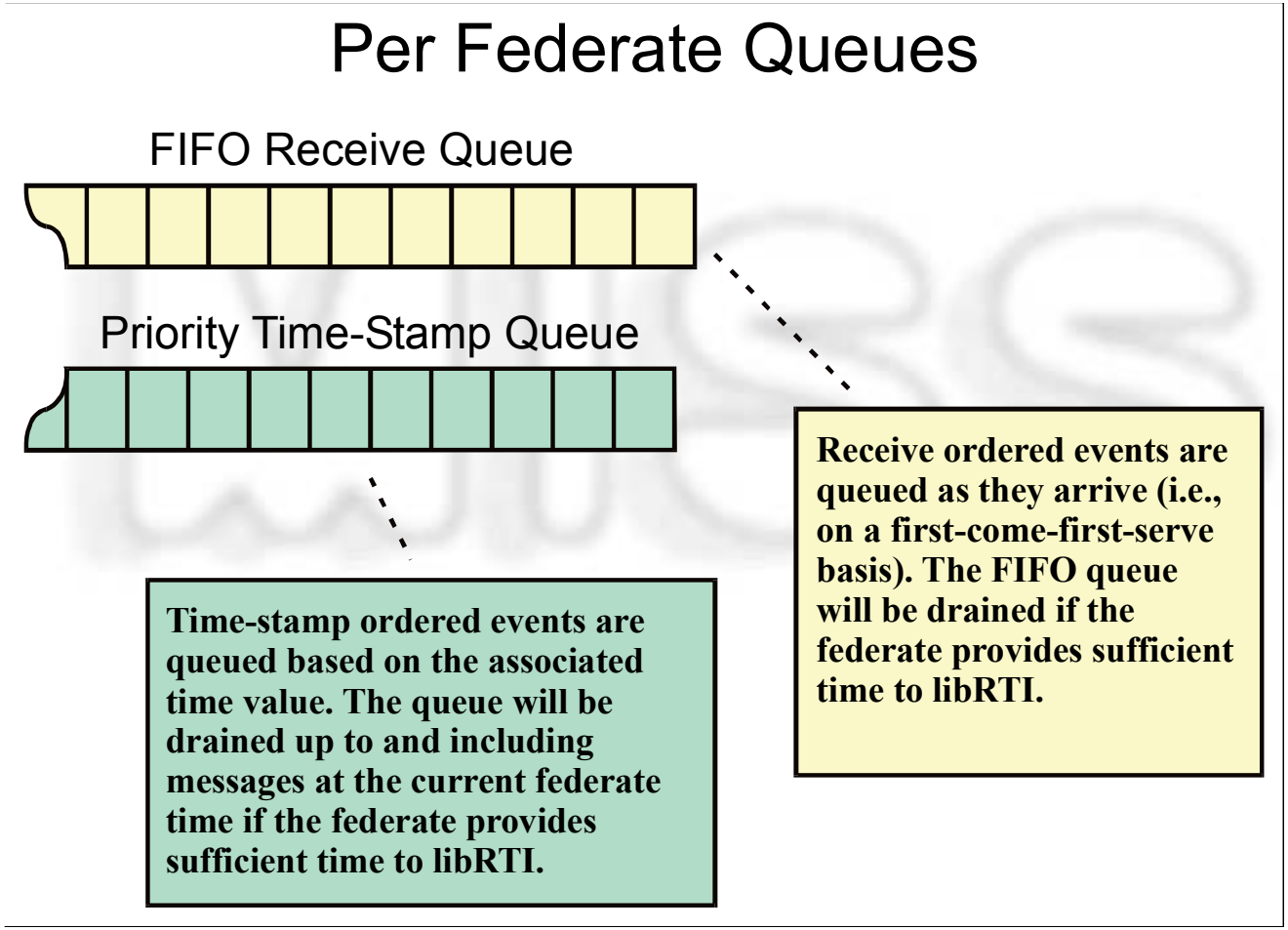
Late Arriving Federate



Requirements for TSO Delivery

- Sender Must be Time Regulating
- Receiver Must be Time Constrained
- Message Must be Designated TSO

Receiving Events



Time Regulation and Constraint Functions

enableTimeRegulation()

timeRegulationEnabled†()

disableTimeRegulation()

enableTimeConstrained()

timeConstrainedEnabled†()

disableTimeConstrained()

queryFederateTime()

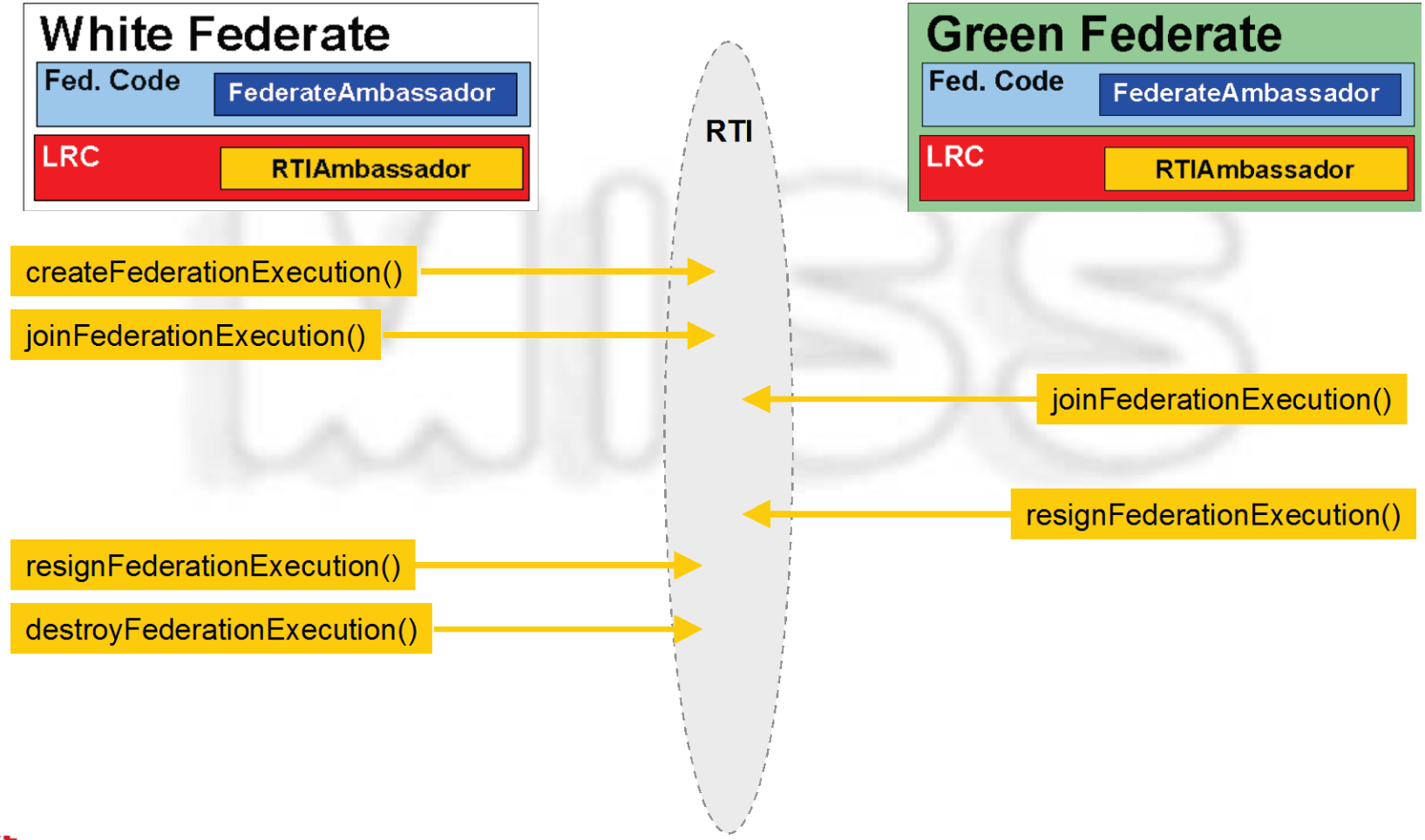
queryLBTS()

queryMinNextTimeEvent()

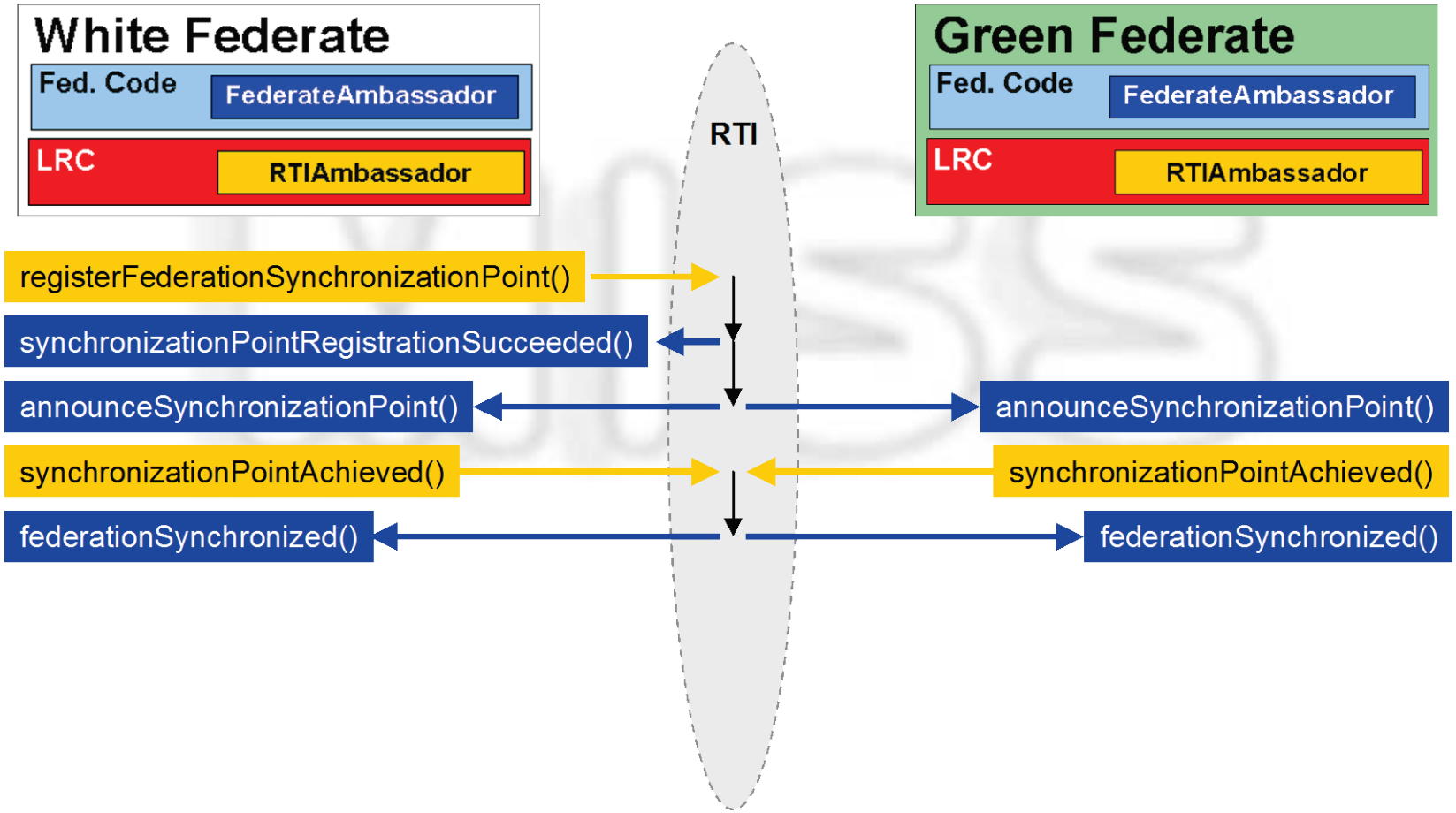
queryLookahead()

modifyLookahead()

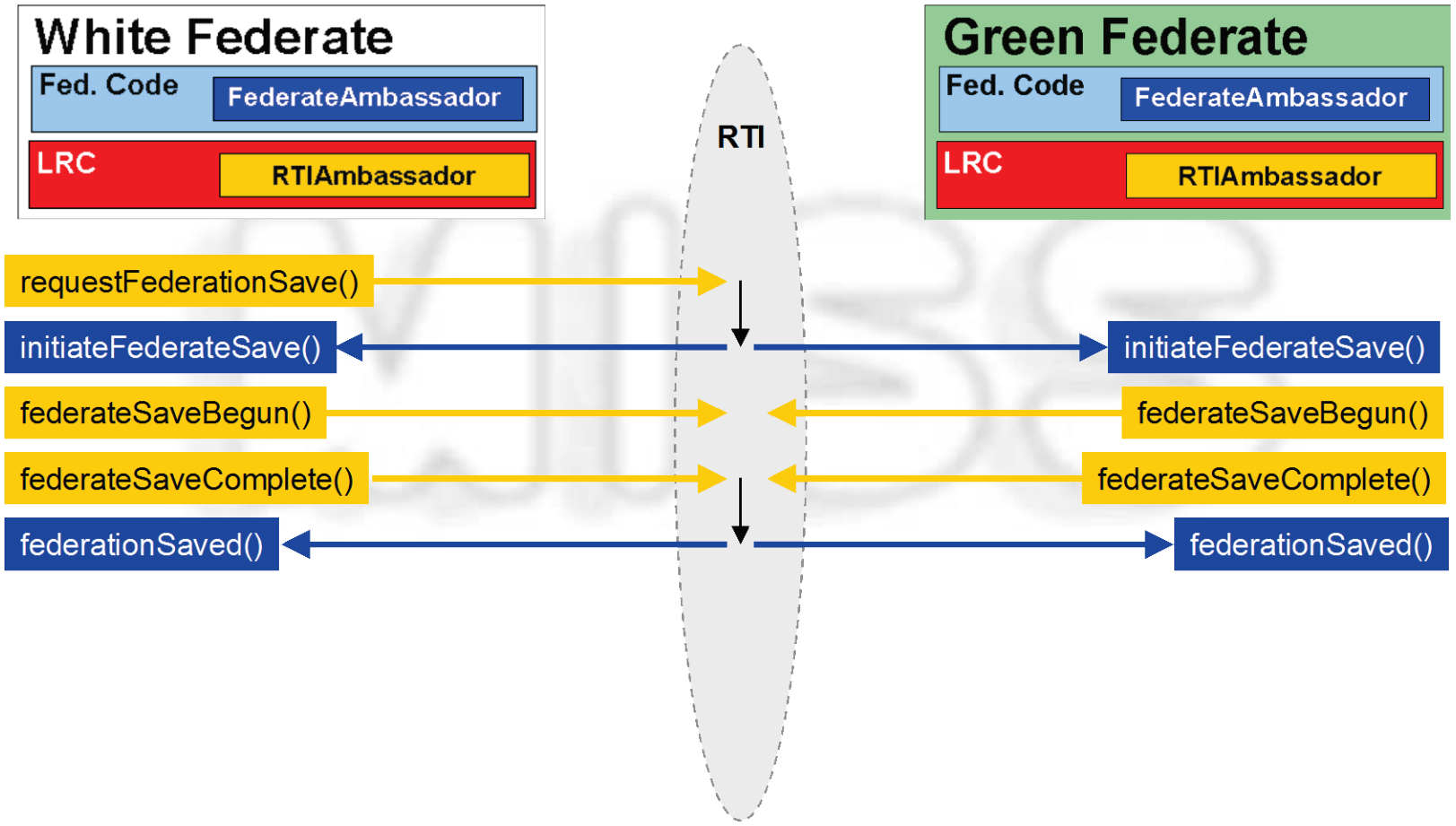
Federation Management Life Cycle



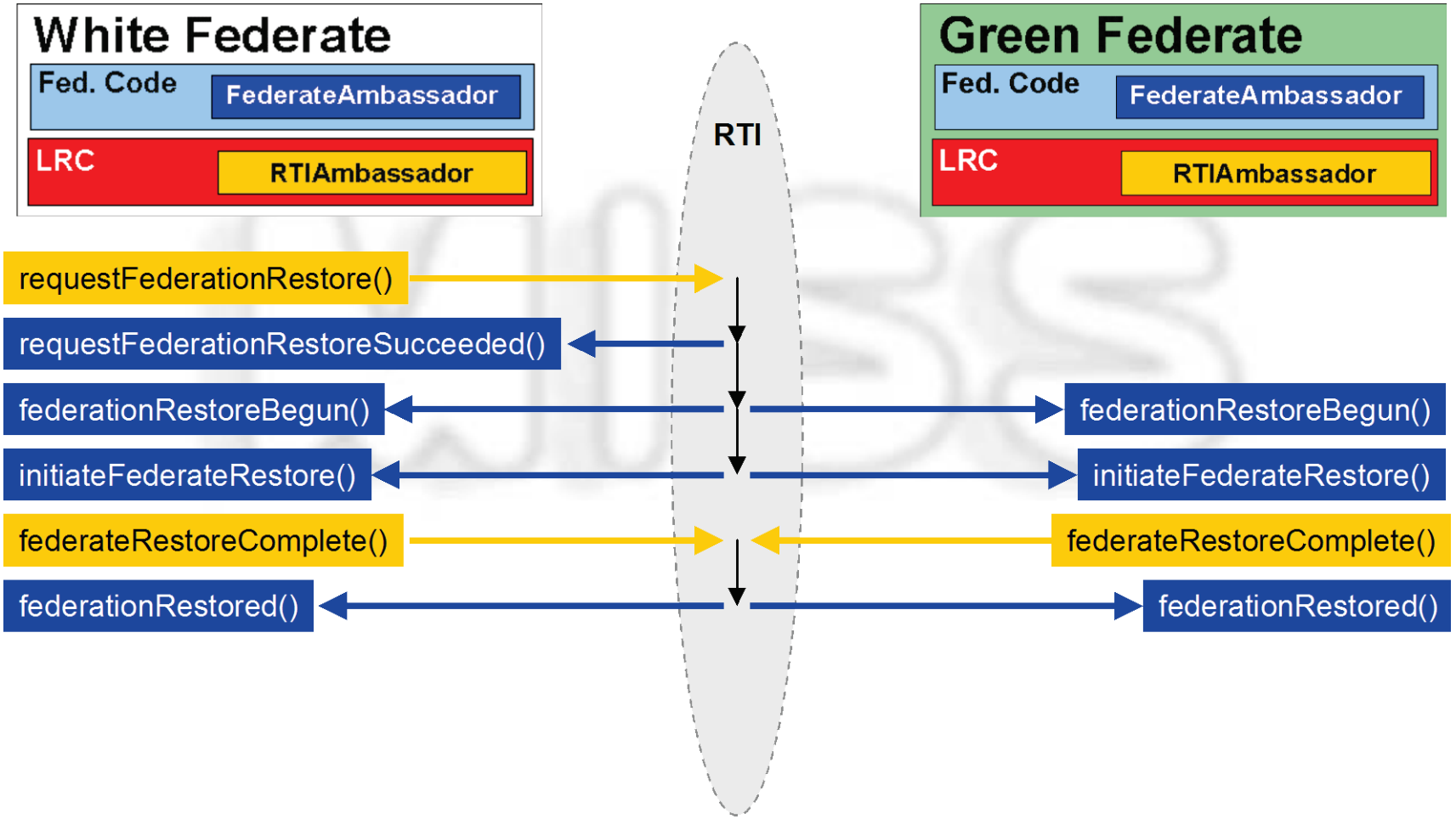
Federation Management Synchronization



Federation Management
Save



Federation Management
Restore



Federation Rules

1. Federations shall have a FOM, documented in accordance with the OMT.
2. All representation of objects in the FOM shall be in the federates, not in the RTI.
3. During a federation execution, all exchange of FOM data among federates shall occur via the RTI.
4. During a federation execution, federates shall interact with the RTI in accordance with the HLA interface specification.
5. During a federation execution, an attribute of an instance of an object shall be owned by only one federate at any given time.

Federate Rules

6. Federates shall have a SOM, documented in accordance with the OMT.
7. Federates shall be able to update and/or reflect any attributes of objects in their SOM, and send and/or receive SOM interactions externally, as specified in their SOM.
8. Federates shall be able to transfer and/or accept ownership of attributes dynamically during a federation execution, as specified in their SOM.
9. Federates shall be able to vary the conditions under which they provide updates of attributes of objects, as specified in their SOM.
10. Federates shall be able to manage local time in a way which will allow them to coordinate data exchange with other members of a federation.