

High Level Architecture

Module 1

Basic Concepts



The Society for
Computer
Simulation



Roy Crosbie
John Zenor

McLeod Institute of Simulation Sciences
California State University, Chico

California State University, Chico

High Level Architecture

Module 1

Basic Concepts

Part 5



Declaration Management

- Declaration Management serves the purpose of coordinating data exchange between federates.
- Declaration Management:
 - Specifies the data a federate will send and receive.
 - Controls where data is sent based on external interest.

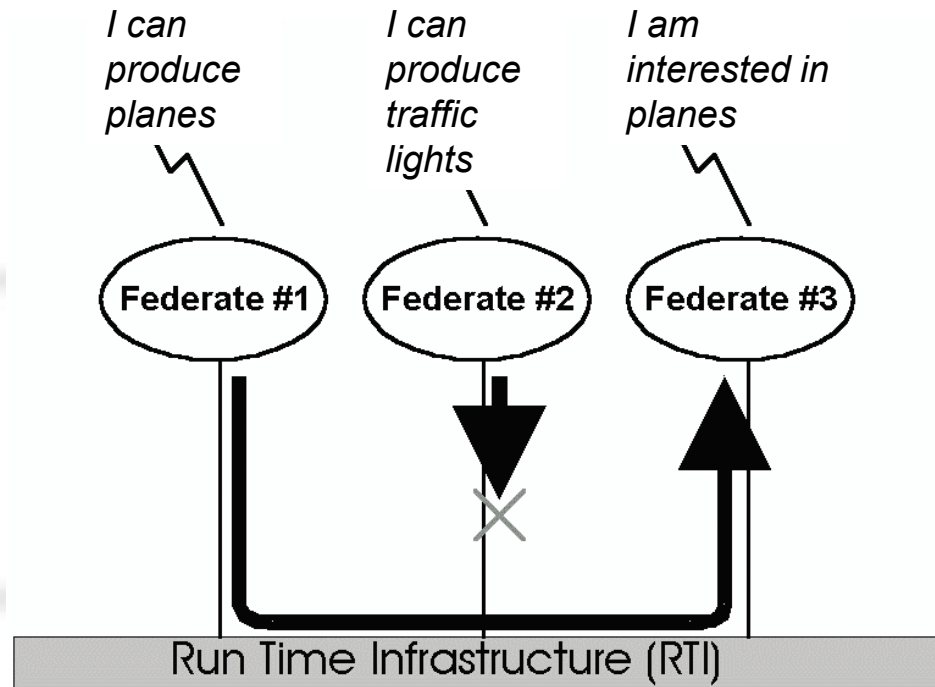


Declaration Management

- Declaration Management includes publication, subscription, and supporting control functions.
 - Federates that produce objects (or object parts) or that produce interactions must declare exactly what they are able to publish (i.e. generate and provide to the federation.)
 - To begin receiving updates of an attribute or an interaction, a federate must declare (subscribe) its interest in the attribute.



Declaration Management (2)



The RTI uses this information to match what is being published to information which has been subscribed to by other federates. This matching enables the RTI to tell producing federates whether to bother updating an attribute or producing interactions of a given class.

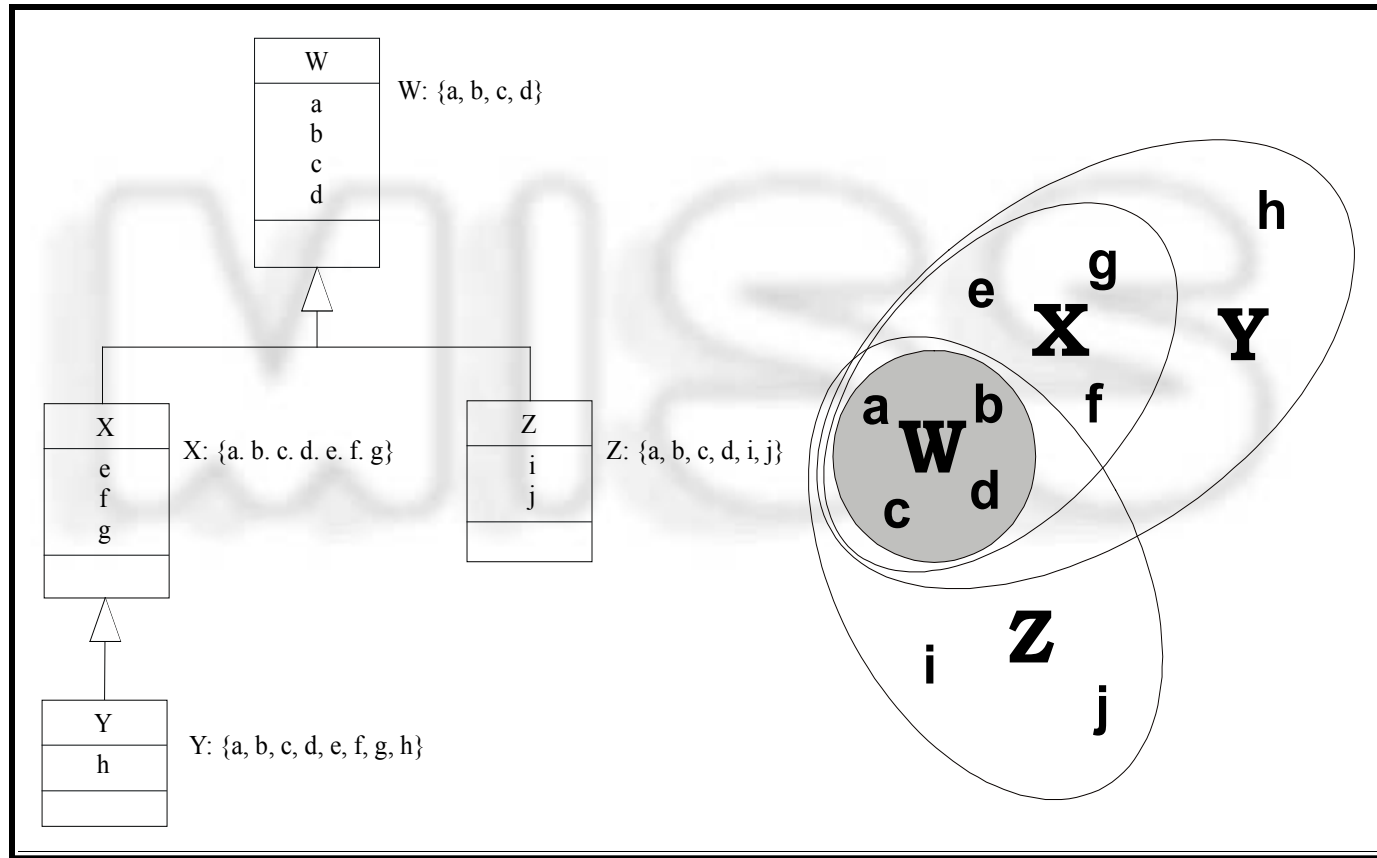


Object Vocabulary

- Object Classes
 - Comprised of Attributes
 - Describe Types of Things That Can **Persist**
- Interaction Classes
 - Comprised of Parameters
 - Describe Types of Events (Transitory)
- **Objects Persist, Interactions Do Not**

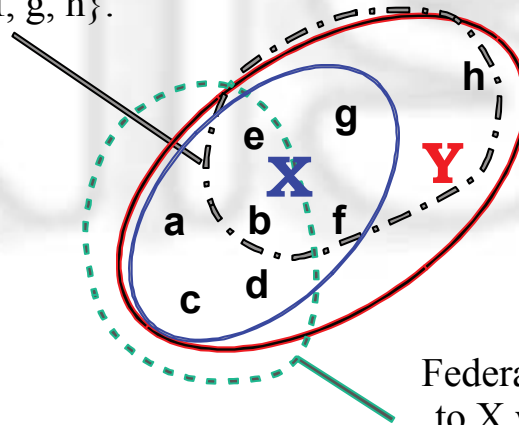


Class Hierarchy - Venn Diagram



Object Publication And Subscription

Federate #1 Publishes Y w/
the following attributes:
 $\{b, e, f, g, h\}$.



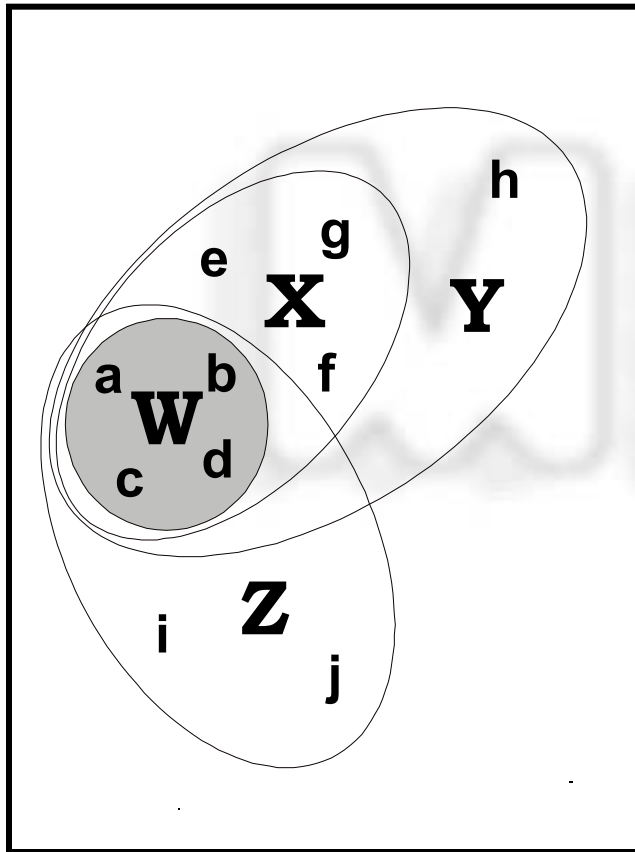
Federate #2 Subscribes
to X w/ the following
attributes: $\{a, b, c, d, e\}$.

Interaction Publication

- As with object classes, each federate must state explicitly which interaction classes it intends to produce.
- Interactions are produced as “all or nothing”.
 - It isn’t possible to specify which parameters in an interaction will be published.



Object Subscription and Type Promotion



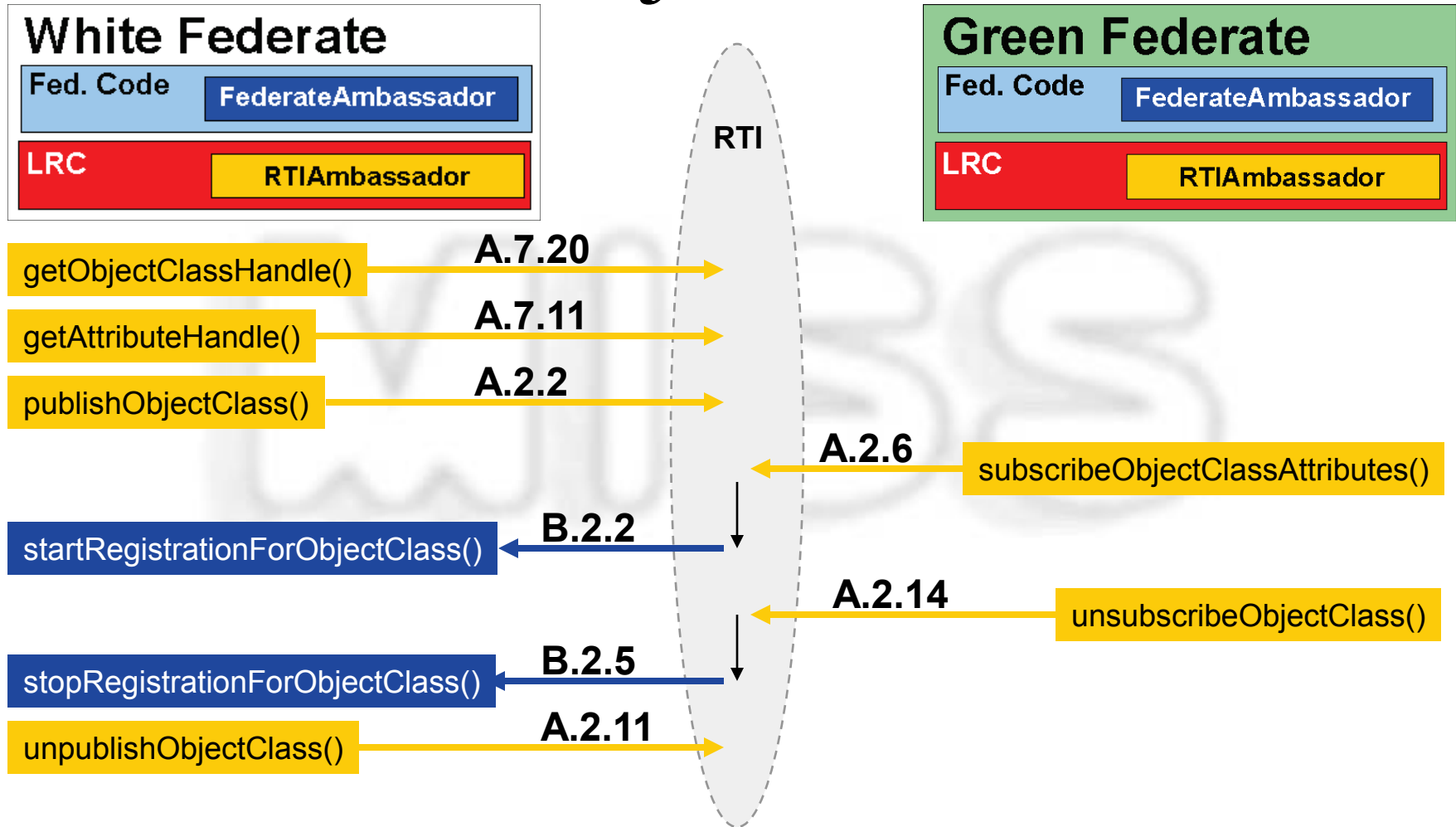
- If subscribed to **W** and **X**:
 - Instances of Class **W** and **X** seen without promotion
 - Instances of Class **Y** seen as instances of Class **X**
 - Instances of Class **Z** seen as instances of Class **W**

Interaction Subscription

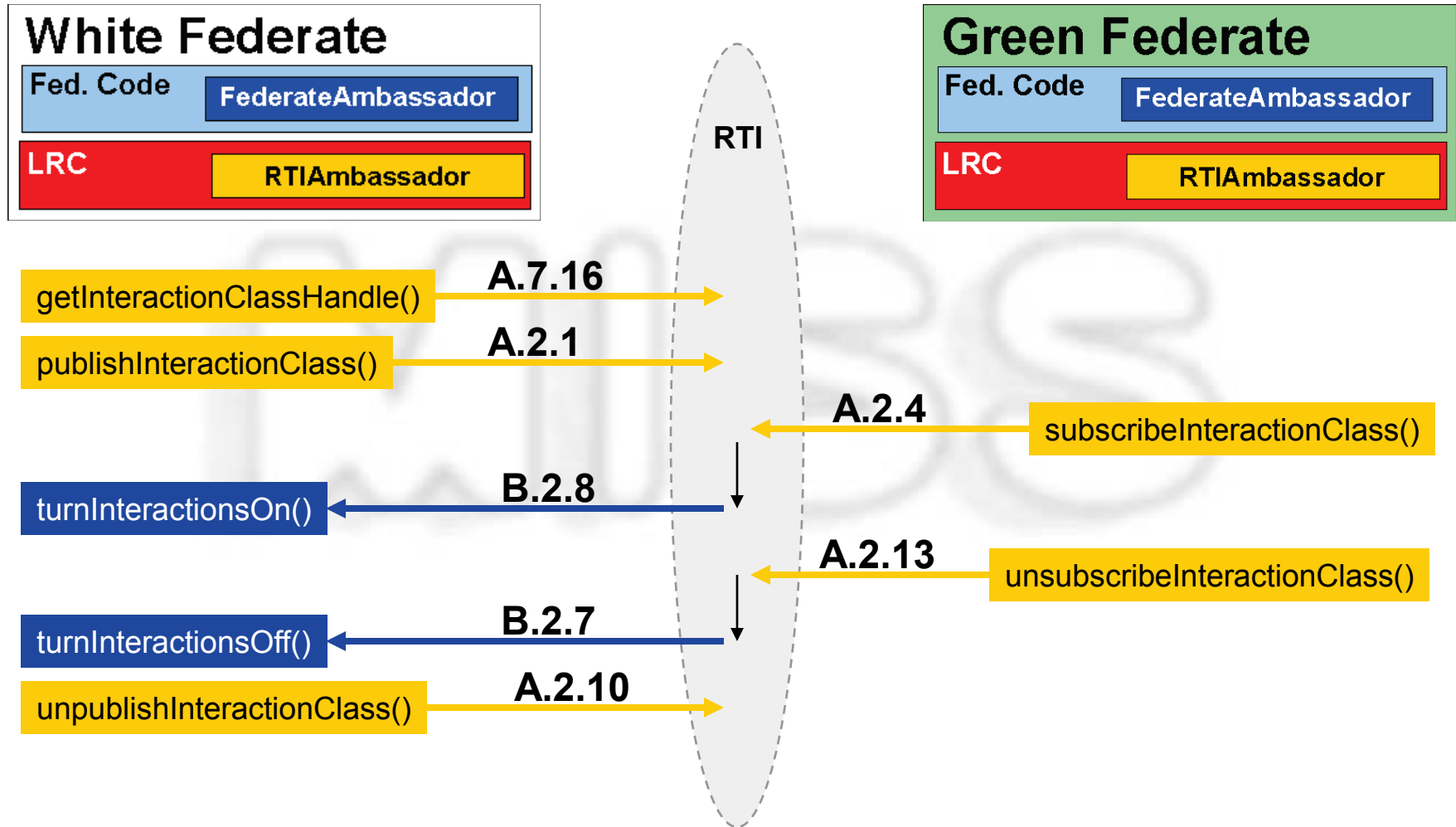
- Each federate subscribes to the interaction classes it wishes to receive.
- It is not possible to subscribe to individual parameters of an interaction class.
- A federate is informed about a new interaction instance if
 - The federate has subscribed to the interaction class of the instance
 - The instance can be promoted (i.e., up the hierarchy) to a subscribed interaction class.



Declaration Management Objects



Declaration Management Interactions

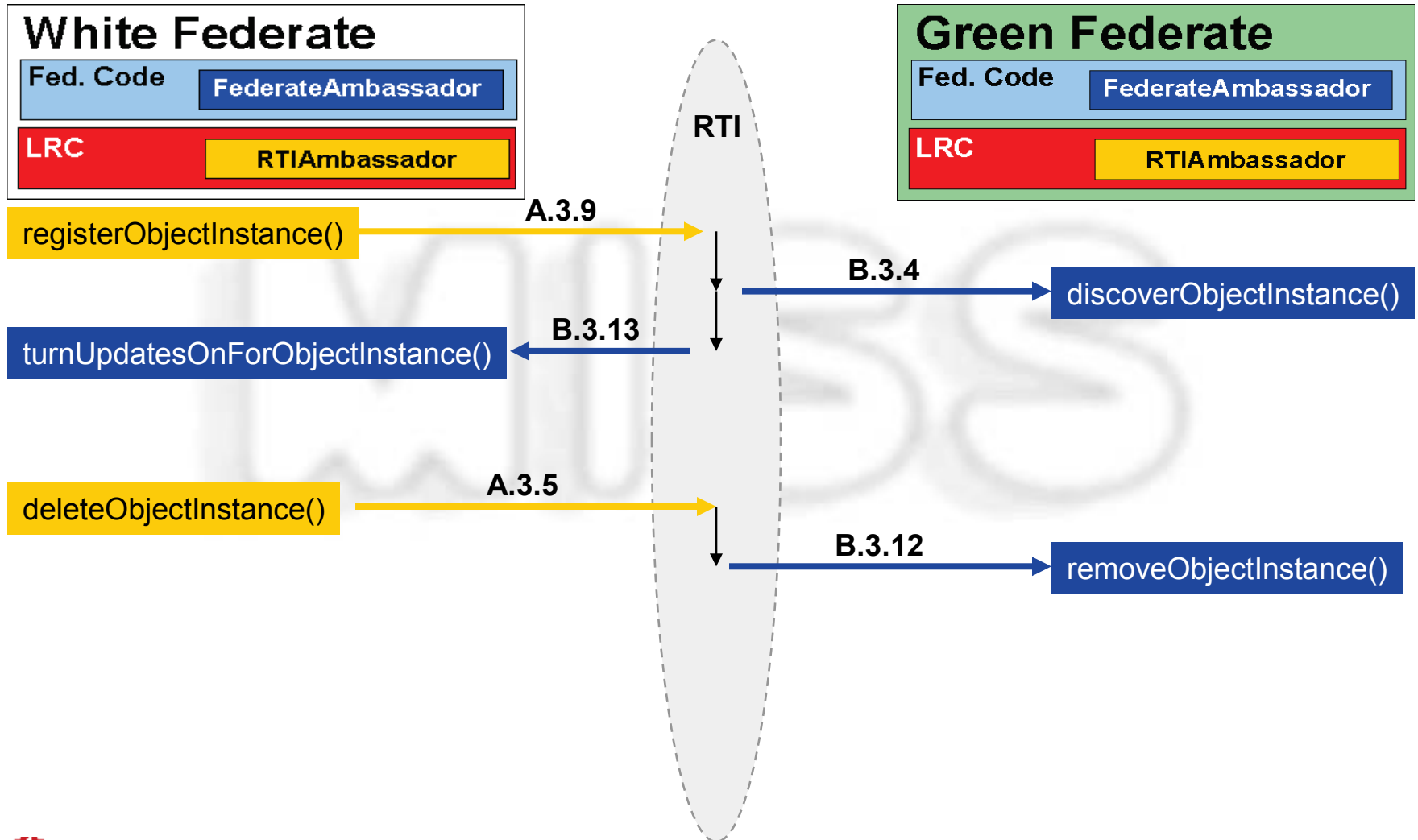


Objects

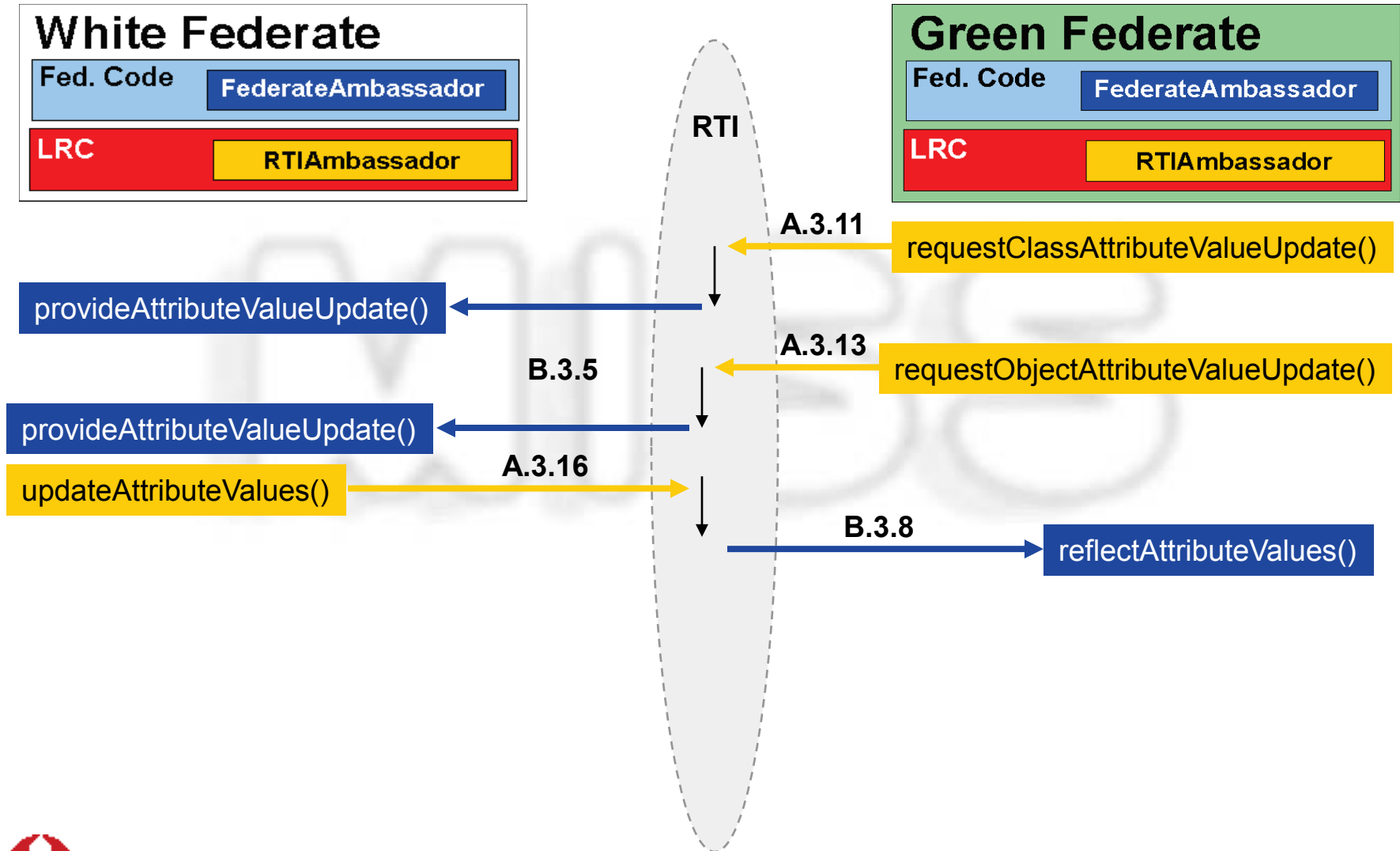
- To create an object, the federate must have published that object class (declaration mgt.)
- The federate registers the new instance of the object
- To discover this object, other federates must have subscribed to that object class (declaration mgt.)
- Those federates then discover the object instance



Object Management Methodology



Object Management Updates

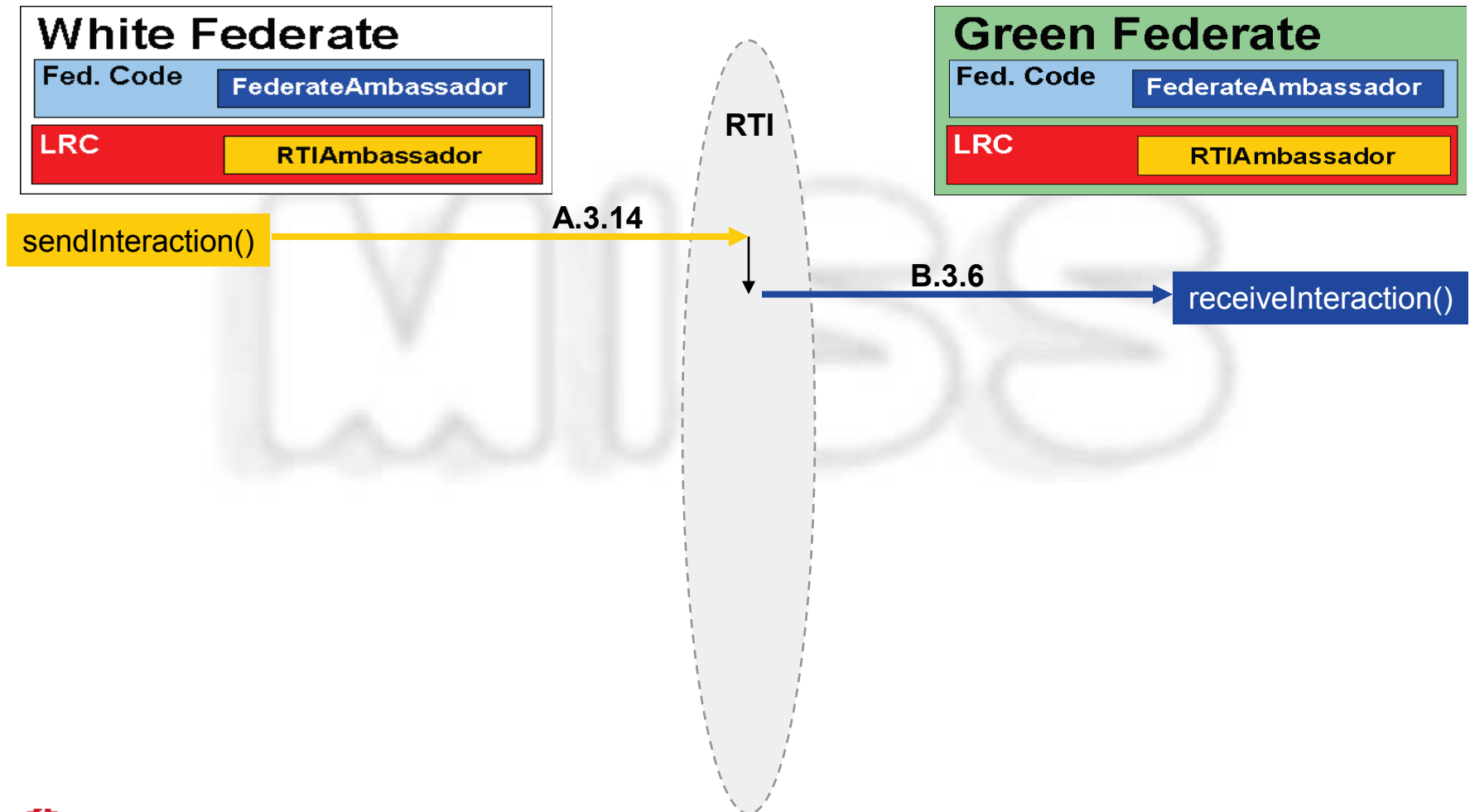


Interactions

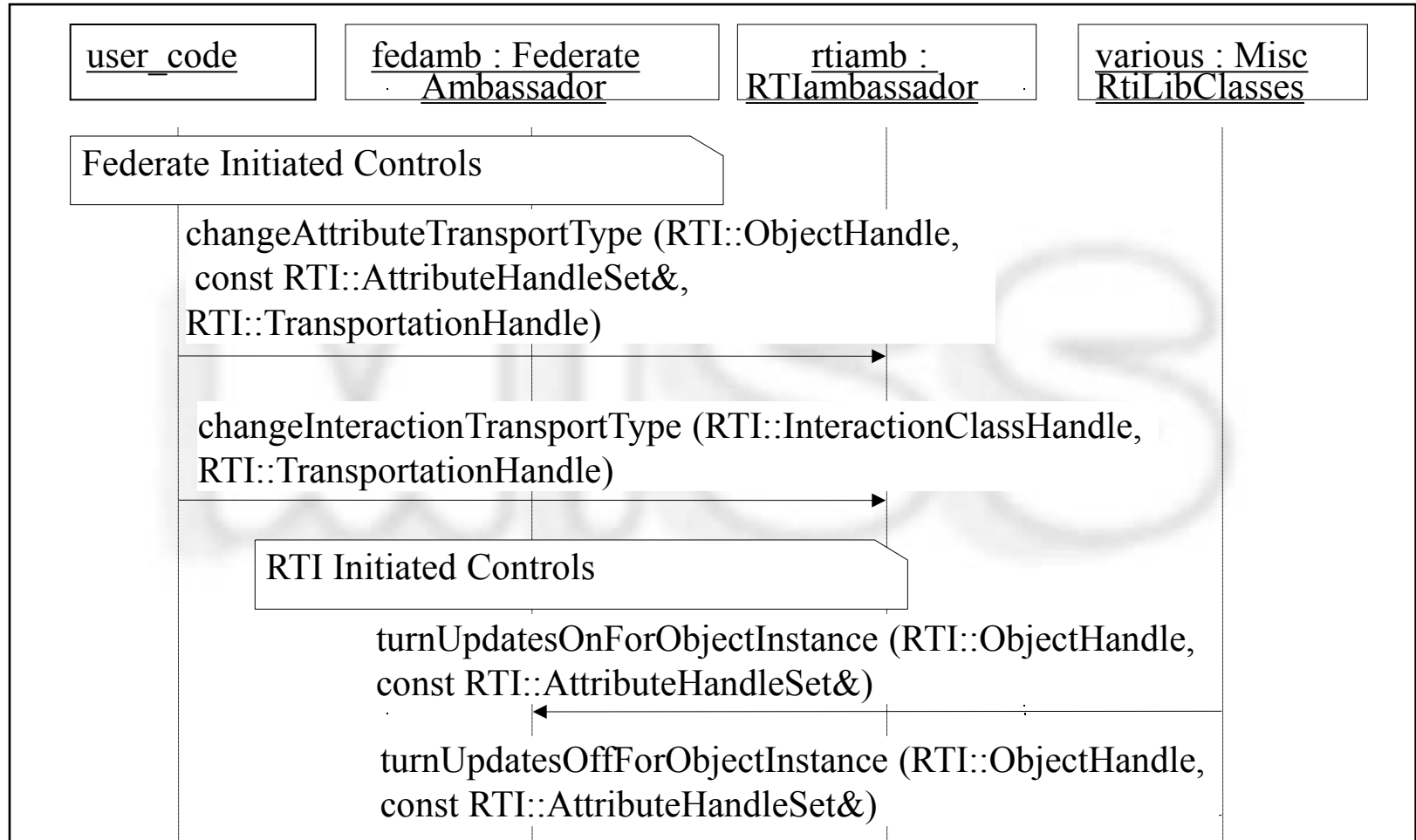
- To send an interaction, the federate must have published that interaction class (per declaration management)
- To receive this interaction, other federates must have subscribed to that interaction class (per declaration management)



Object Management Interactions



Additional Object Control



Scope Interactions

