



Universal CPS Environment
for Federation

UCEF I.O.O BETA Release

Tom Roth, Marty Burns NIST

Thanks to those who contributed code to this release!

NIST

Abhinav Pandey
Art Griesser
Christopher Lemieux
Eugene Song
Geoffry Roberts
Hohyun Lee
James Arnold
Khalid Halba
Martin Burns
Thomas Kramer
Thomas Roth

Vanderbilt

Andras Nadas
Gabor Pap
Gergely Varga
Harsh Vardhan
Himanshu Neema
Stephen Rees
William A. Emfinger
Yogesh D. Barve

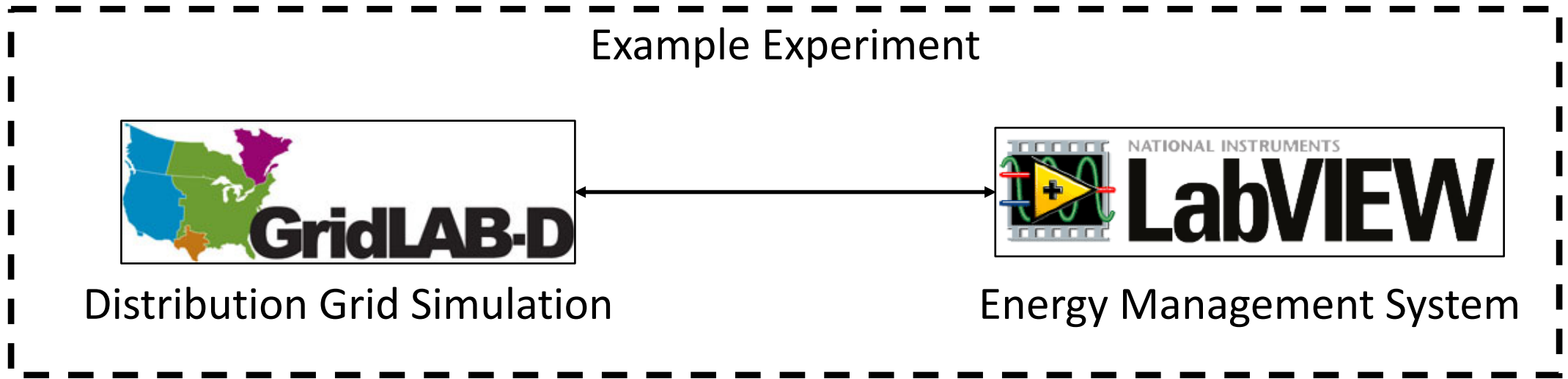
Agenda

- What is UCEF?
- Selected Use Case: Transactive Energy Challenge
- UCEF 1.0.0-BETA Features
- Collaboration Opportunities
- Video Demonstration

What is UCEF?

Universal CPS Environment for Federation (UCEF)

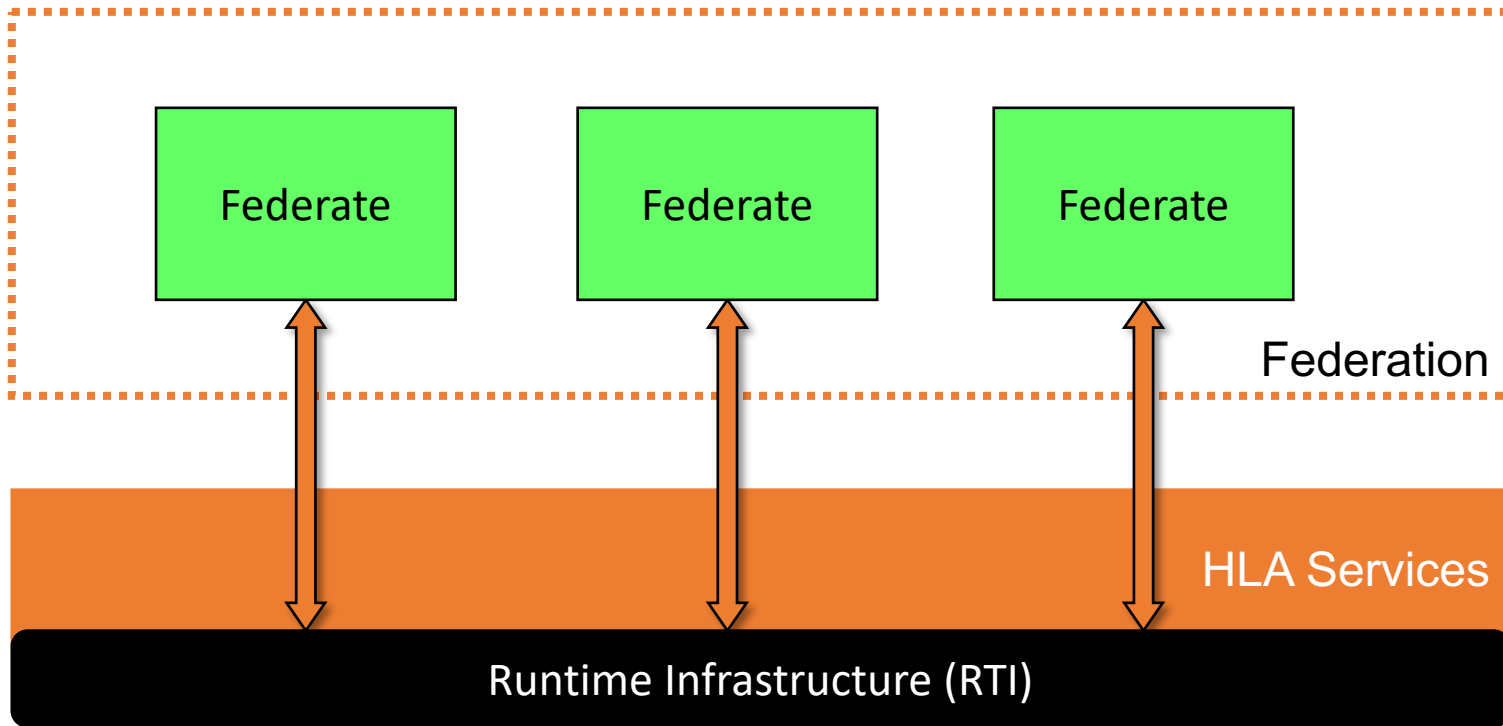
- A software tool that allows users to graphically compose its supported simulators into a runnable experiment:



- UCEF is intended to make it easier for researchers to design and run CPS experiments without experience in distributed simulation.

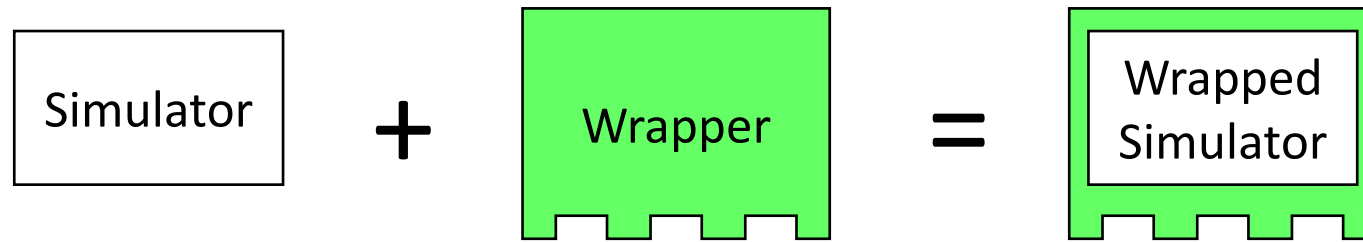
IEEE 1516-2010 High Level Architecture (HLA)

UCEF uses the High Level Architecture (HLA), an IEEE standard that defines the services a set of *federates* can use in a *federation*.

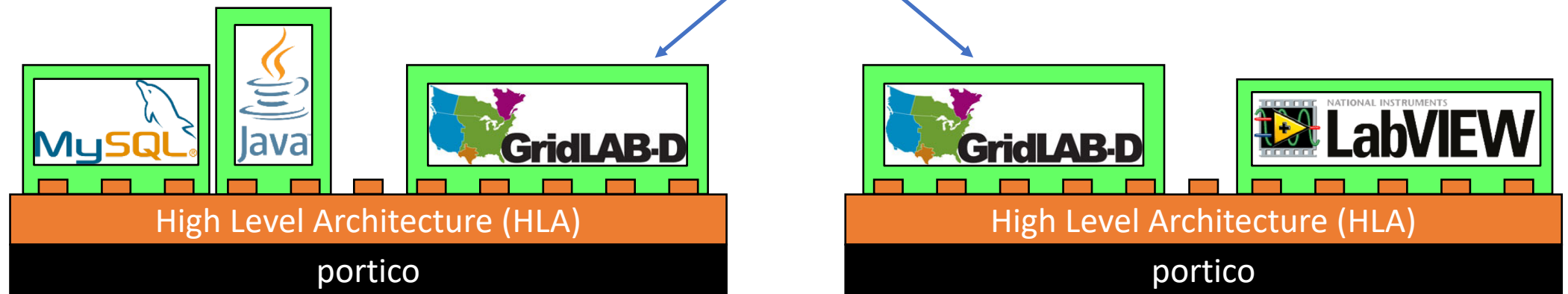


Software wrappers integrate new things into UCEF

- A **wrapper** is software that defines the method of *time synchronization* and *data exchange* used for a simulator in a federation.



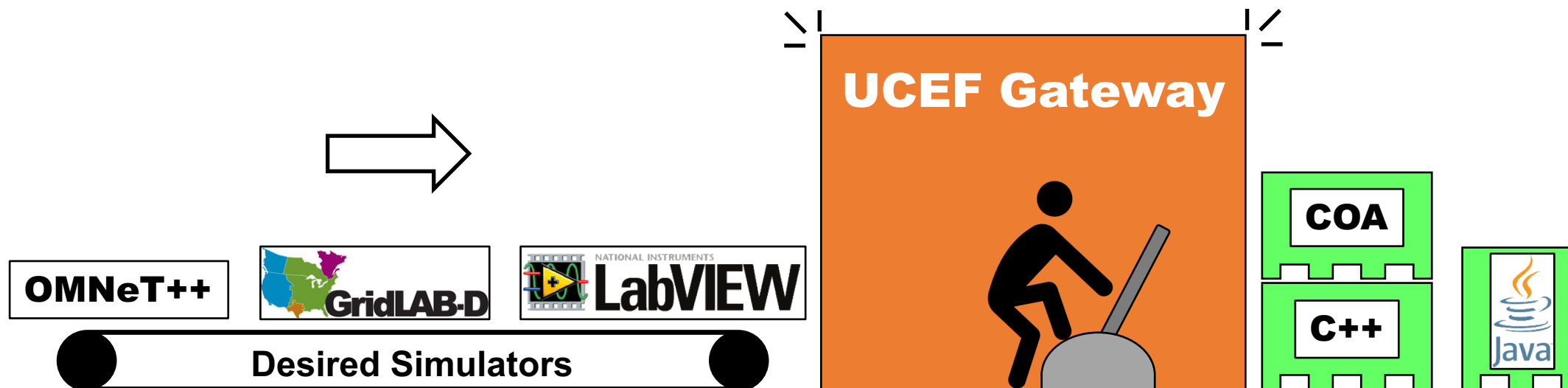
- A simulator with a wrapper can be *re-used* in multiple federations:



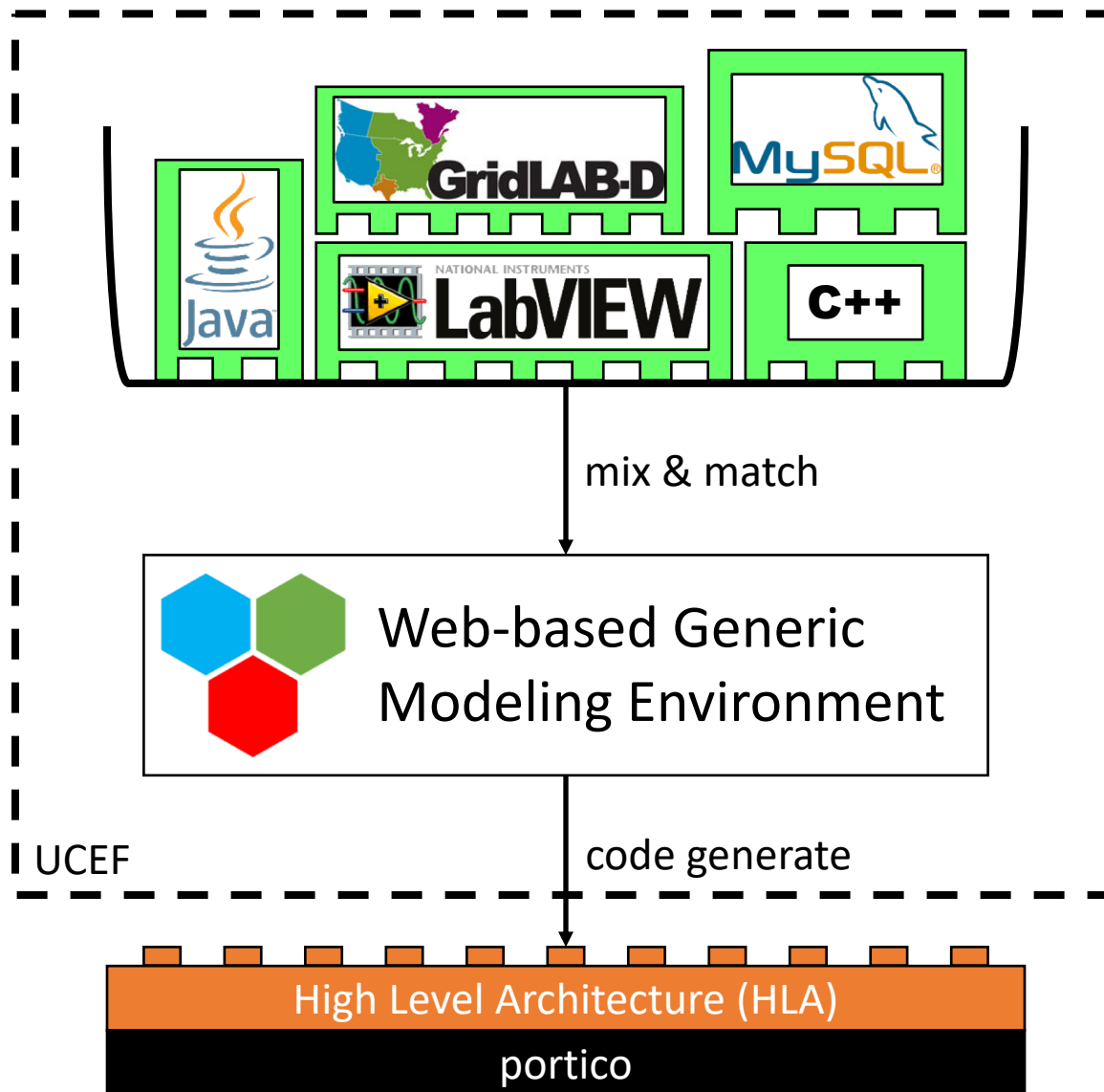
UCEF Gateway is used to create new wrappers

A *Java library* designed to be:

- Usable with limited HLA expertise
- ***Easily integrated with new things***
- Agnostic to the federation data model



UCEF is a portable HLA development environment



UCEF Virtual Machine



Web-based Graphical Modeling Environment (WebGME)

- The graphical language used to design federates:



- The **federates** designed in this language are transformed into executable code/simulation models using code generation.

Example WebGME Federation

GME > SigDur > master > IntegrationModel

Find...

VISUALIZER SELECTOR

- Composition
- Crosscut
- Attribute
- COA
- CPNFederate
- CppFederate
- Deployment
- Experiment

IntegrationModel

0.8x

```

    graph LR
      LightSensor[LightSensor] --> Brightness[Brightness]
      ProximitySensor[ProximitySensor] --> ProximityInfo[ProximityInfo]
      Brightness --> LightController[LightController]
      ProximityInfo --> LightController
      LightController --> SwitchOnOff[SwitchOnOff]
      SwitchOnOff --> Switch[Switch]
  
```

OBJECT BROWSER

- SigDur
 - BasePackage
 - ActionBase
 - C2WInteractionRoot
 - FederateJoinInteractio
 - FederateObject
 - FederateResignInterac
 - HighPrio
 - InteractionRoot
 - LowPrio
 - MediumPrio

PROPERTY EDITOR

Attributes Pointers Meta Preferences

GUID 0cf117fc-d536-7e6b-5...

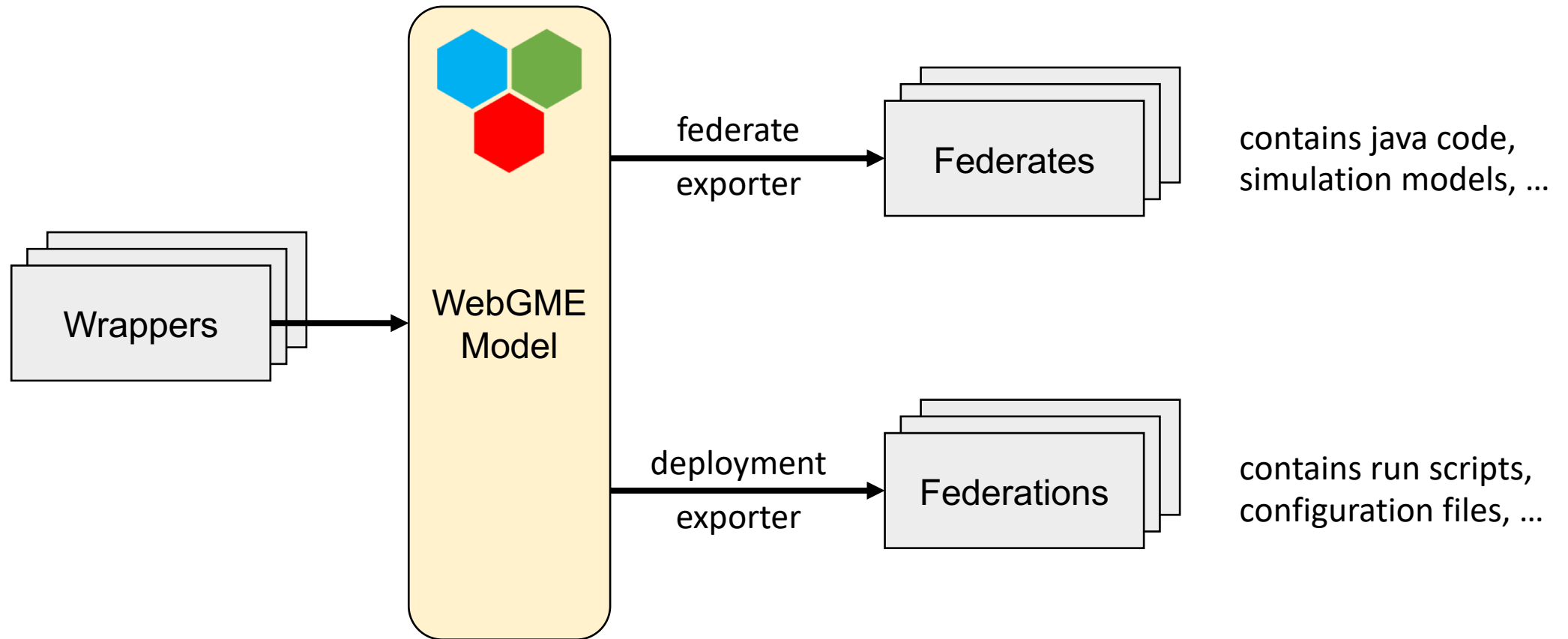
ID /2140716171/y

Meta type FOMSheet

Attributes

name IntegrationModel

WebGME Code Generation



Selected Use Case

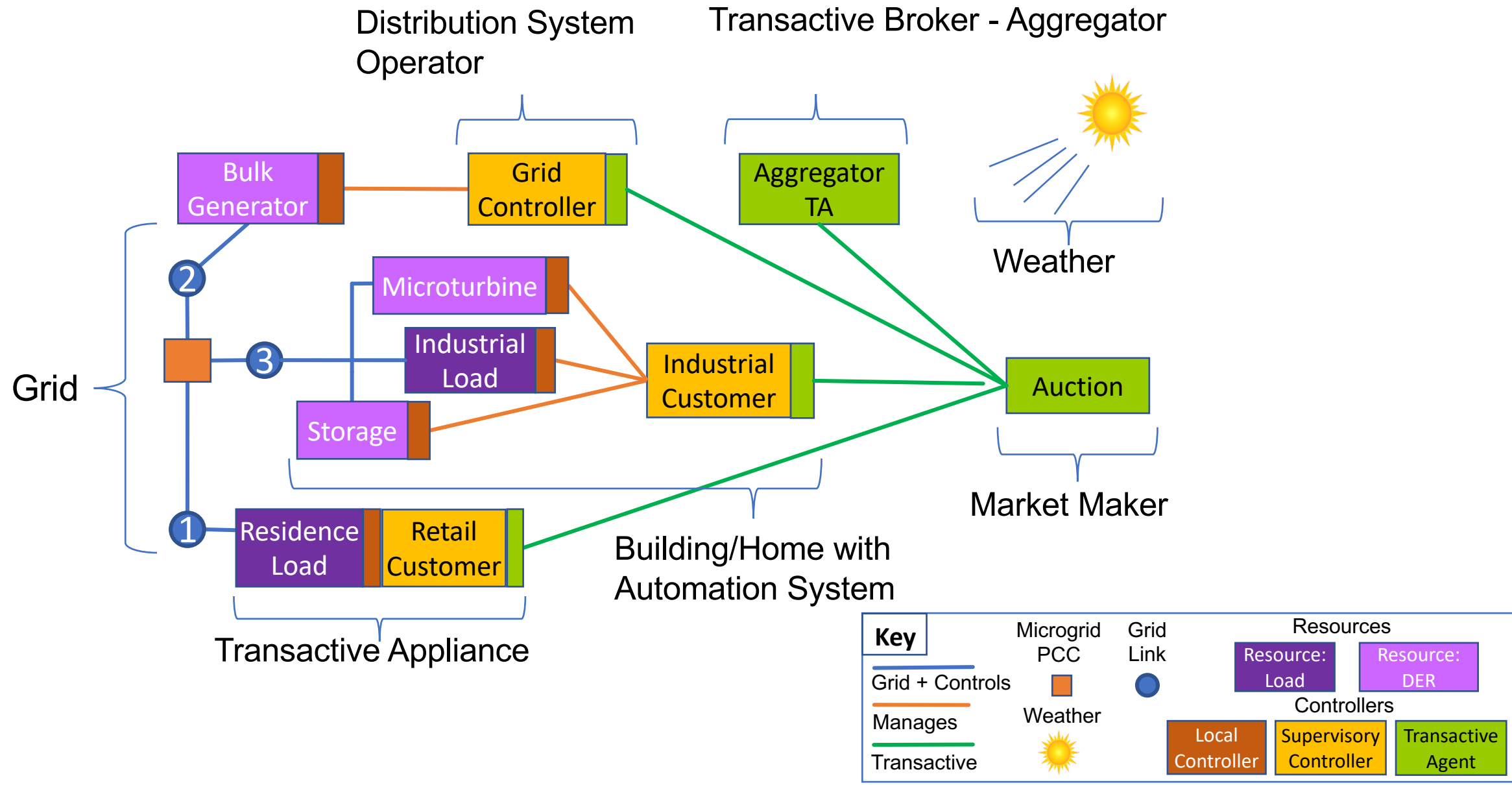
Transactive Energy Testbed

Purpose and Scope

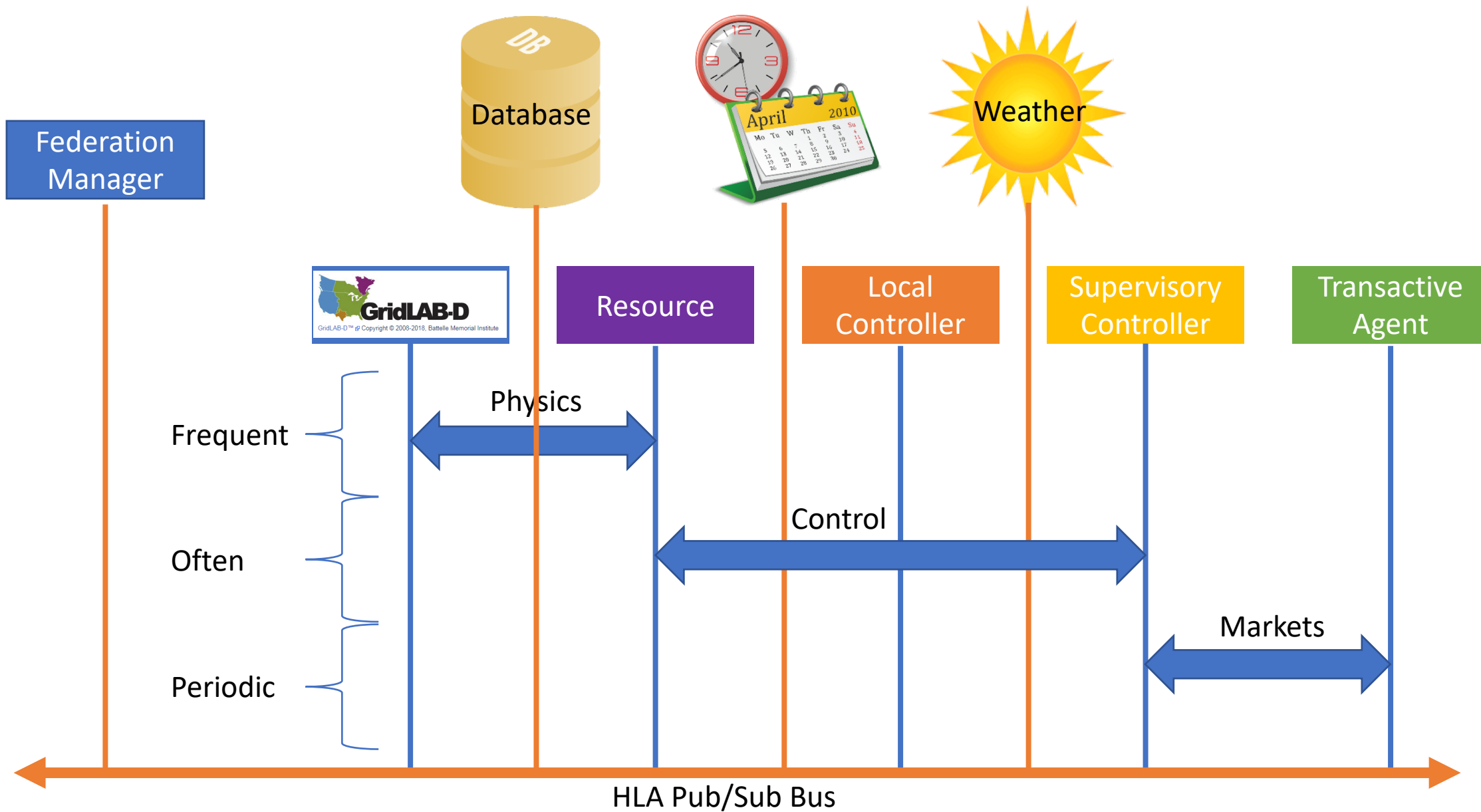
Transactive Energy (TE) is a metaphor for the changing nature of the business and technical dimensions of the distribution of electric power. Transactive energy is a new approach for managing the electric power distribution system based on principles of economic value, with engineering constraints.

- Purpose:
 - Measuring performance of the grid to assess different market-based grid-control mechanisms.
 - Measure the performance differences between fixed price, time of use, real-time price, and dynamic bidding on UCEF Testbed.
- Scope:
 - Utilize the CPS Testbed to simulate transactive energy models varying the grid, control algorithms, and market designs.
 - Implement the Abstract Component Model for Transactive Energy in UCEF
 - Allows separate federate models that can be composed for various scenarios.
 - Separate models for loads, generators, grid, local controllers (e.g. thermostat), supervisory controllers (bas or ha system), weather, transactive agents
 - For each pricing scheme a separate supervisory controller/transactive agent pair is constructed to model the behavior

Composability of TE Simulations



Common Platform Canonical Simulation



Advantages of using UCEF for this experiment

- The TE Challenge Component Model was designed to mix and match components from different simulation environments, platforms, scale and location.
- The HLA Publish and Subscribe mechanism focuses on the data exchanged and the name of the published instance which is decoupled from the software components that emit and consume them.
- This allows:
 - One simulation to be run the same way regardless of how many different kinds of loads, controllers, and markets are present.
 - The uniform capture of all data exchange supports a single method of running the simulation and postprocessing on any experimental instance.
- UCEF has:
 - GridLAB-D Federate Model that allows simple to complex grid models to simulated
 - Ucef-database Federate Model provides for universal data acquisition for results analysis

Gridlab-D



GridLAB-D™ Copyright © 2008-2018, Battelle Memorial Institute

- Powerful discrete event simulator for Smart Grid from PNNL
 - C++ based solver can solve very large grid models with very few computing resources
 - Tested and validated models of most grid components
- Simple Federate Model
 - Configuration file contains description of interactions and objects published and subscribed.
 - Mapping to GridLAB-D object model components in XML
 - Drop in and use like any other federate in WebGME

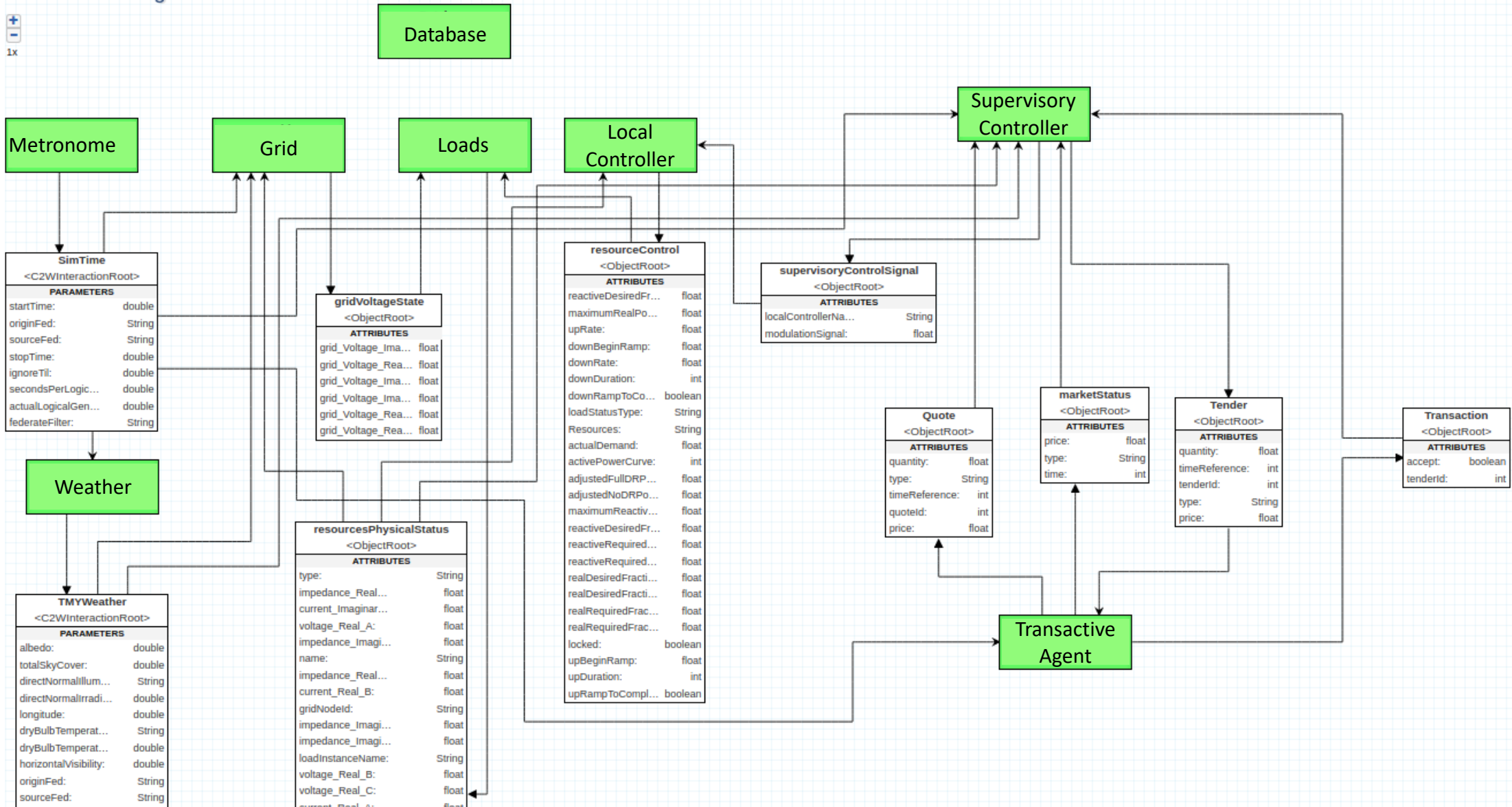
UCEF-Database



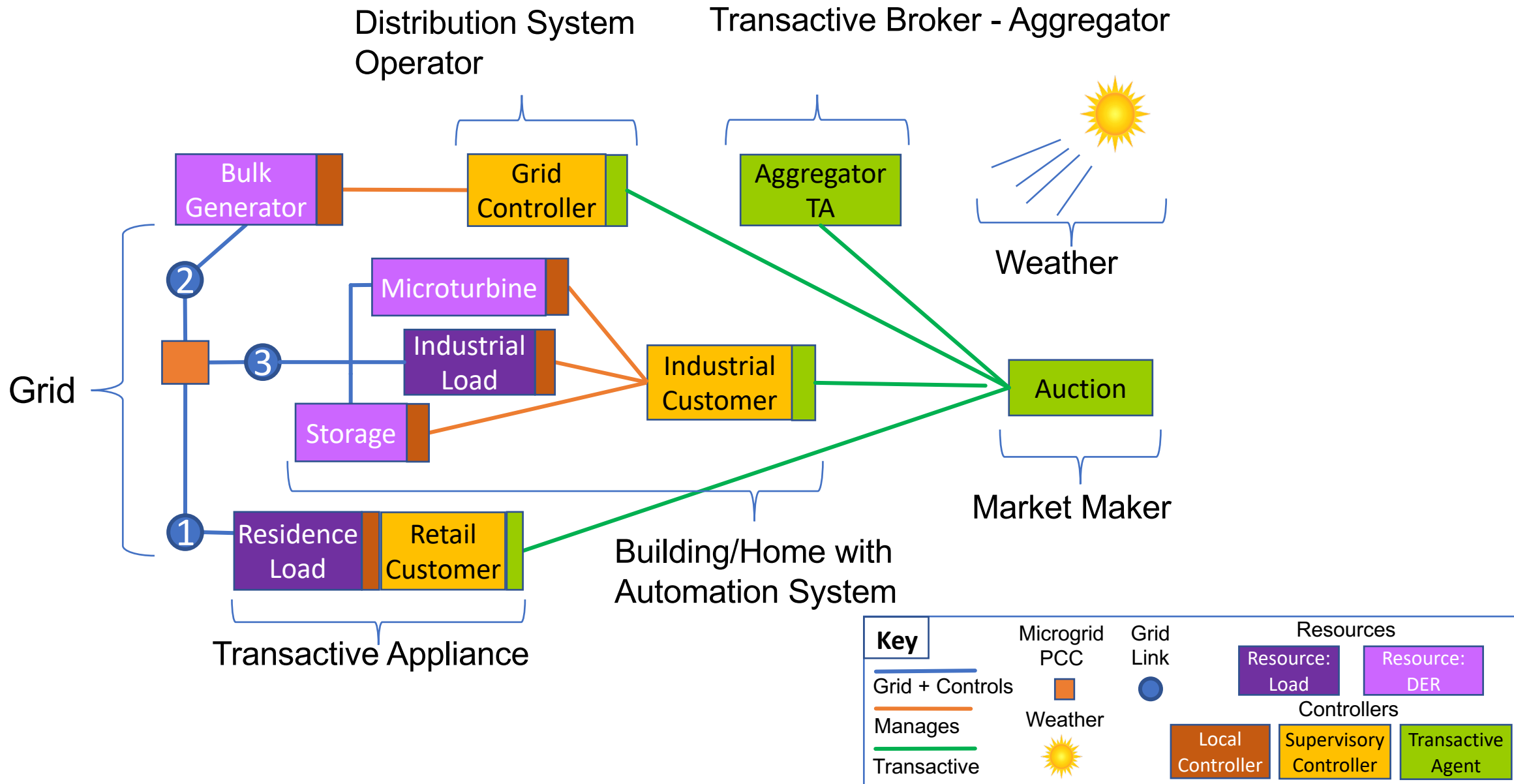
- Simple Federate Model
 - Gets a configuration file describing all interactions and objects
 - Listens on the HLA bus
 - Creates one table per information component published by any federate
 - Records time ordered results of message traffic
 - Each experiment creates own schema
- MySQL Database
 - Default database – can be modified to other Java SQL Connections
 - MySQL Workbench included in UCEF
 - Database Tools transform SQL into JSON objects for later processing

TE Challenge Federation on UCEF

FederationDesign



Thus UCEF Provides an Elegant Platform for This Experiment



UCEF 1.0.0 BETA

UCEF I.O.O BETA Feature List

• Supported Wrappers

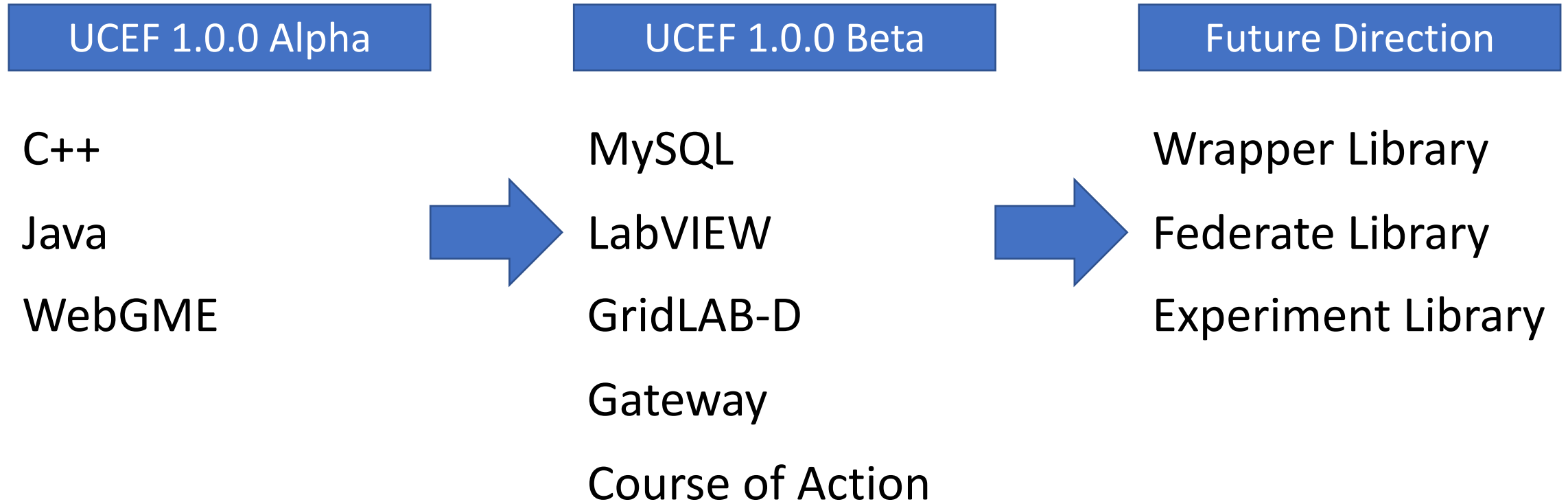
- Java
- C++
- MySQL
- LabVIEW
- GridLAB-D

• Other Tools

- WebGME
- Course of Action
- Gateway Library



UCEF Release Timeline



Collaboration and Future Webinars

Collaborate with us!

- For General Information on UCEF:
 - NIST UCEF Information: <https://www.nist.gov/el/cyber-physical-systems/ucef-universal-cps-environment-federation>
- For Collaboration and Technical Developer Details:
 - NIST UCEF Collaboration Site: <https://pages.nist.gov/ucef/>
- For Source Code and Building / Downloading the UCEF Virtual Machine:
 - UCEF Development Site: <https://github.com/usnistgov/ucef>
 - UCEF Virtual Machine Issues List: <https://github.com/usnistgov/ucef/issues>
- For Email List Discussion (New)
 - UCEF Mail List: ucef+subscribe@list.nist.gov

NIST Site for UCEF

<https://www.nist.gov/el/cyber-physical-systems/ucef-universal-cps-environment-federation>

Engineering Laboratory

CYBER-PHYSICAL SYSTEMS

UCEF: Universal CPS Environment for Federation

Why Do We Need UCEF?

How Does UCEF Work?

What are UCEF's Key Features?

CPS Public Working Group (PWG)

Big Data (PWG)

Smart Grid

Smart America/Global Cities

Archived CPS Events

UCEF: Universal CPS Environment for Federation



Cyber-physical systems (CPS) are highly interconnected systems that will provide new functionalities to improve quality of life and enable technological advances in critical areas. Among these critical areas are personalized health care, emergency response, traffic flow management, smart manufacturing, defense and homeland security, and energy supply and use. CPS and related systems (including the Internet of Things



The UCEF logo shows the four letters—"U" "C" "E" and "F". Each of the four letters

UCEF Collaboration Site

<https://pages.nist.gov/ucef/>

< NIST



UCEF 1.0.0-BETA Released!

Download UCEF 1.0.0 BETA with this link

Upcoming Webinar: Universal CPS Environment for Federation (UCEF) BETA Release (September 14, 2018, 11:00 a.m. EDT)

NIST will host a one and one half hour webinar—from 11:00 a.m. to 12:30 p.m., EDT, on September 14, 2018—to present the next release of the **Universal CPS Environment for Federation (UCEF)**. This new release (our BETA version) incorporates lessons learned through early usage and adds several substantial components to the UCEF repertoire—LabVIEW, GridlabD, Omnet++, and COA. These are in addition to native Java and C++ applications. The BETA version also features a universal gateway component that was developed to facilitate integration of domain-specific tools used in cyber-physical systems (CPS).

The webinar will be open to the first 250 participants to log in or call in. To join the webinar by computer, please go to <https://global.gotomeeting.com/join/832557933> . To join by telephone, please call 1(408)650-3131 and use the access code

Developer Resources

<https://pages.nist.gov/ucef/development/>

< NIST



[About Us](#)

[Library](#)

[Development](#)

[Community](#)

[Join List](#)

Table of Contents for this Page

- [Development Resources](#)
 - [General Development Topics](#)
 - [Development of Federates, Federations and Experiments](#)
- [UCEF Git Source Repositories](#)
 - [UCEF Core Components](#)
 - [UCEF Wrappers](#)
 - [UCEF Library](#)

Development Resources

Source Code and Building

<https://github.com/usnistgov/ucef>

Search or jump to... Pull requests Issues Marketplace Explore

usnistgov / ucef Unwatch 12 Star 6 Fork 2

Code Issues 2 Pull requests 0 Projects 0 Wiki Insights Settings

Universal CPS Environment for Federation <https://pages.nist.gov/ucef> Edit

Manage topics

186 commits 4 branches 2 releases 7 contributors

Branch: develop New pull request Create new file Upload files Find file Clone or download

MartyBurns minor updates and test of build with new features in meta Latest commit c6c5821 2 days ago
build minor updates and test of build with new features in meta 2 days ago
test some refinements and documentation to scripts 6 days ago
.gitignore migrated to gridlab-d on github; testing not completed yet 10 months ago
ACKNOWLEDGEMENT removed keys and secure web sites from project build a year ago
LICENSE removed keys and secure web sites from project build a year ago
README.md Update README file for current structure 4 months ago

Technical Library

<https://pages.nist.gov/ucef/library/>

UCEF Technical Library

This is the technical library of resource for the UCEF platform. You will find videos, presentations, and software to view and download.

Software

UCEF1.0.0-BETA Virtual Machine (software)

UCEF1.0.0-BETA-20180914

UCEF1.0.0-ALPHA Virtual Machine (software)

UCEF1.0.0-ALPHA-20170725

HowTo Videos

Create PingPong Federation and Generate Code (videos)

This video illustrates how to use WebGME to create a federation and generate code.

Create PingPong Federation and Generate Code (videos)

This video illustrates how to use the Eclipse IDE to add behavior to the PingPong federation and run the resulting code.

Upcoming UCEF Webinars (one per month)

Third Thursday of Each Month, 12:00 P Eastern Time, Beginning October 18.

These components are part of the UCEF 1.0.0 BETA Platform:

- UCEF Gateway – rapid development adaptor to UCEF Platform
- GridLAB-D™ – Powerful re-usable Federate based on ucef-gateway to integrate PNNL Power System simulator
- Database – UCEF database federate and tools to automate software-less data capture from federated experiments
- LabVIEW™ – Integrate hardware and simulations using National Instruments LabVIEW platform, a re-usable federate based on the ucef-gateway
- Courses of Action (COA) – graphical experiment orchestration language in WebGME

Demonstration Videos

UCEF Work Flow

Model the federate

- Choose the federate type
- Create/import objects and interactions
- Define publications and subscriptions

Generate the code

- Run the Federates Exporter
- Run the Deployment Exporter

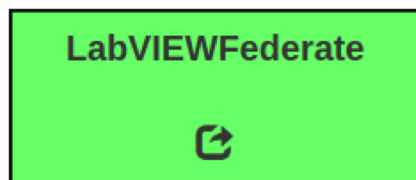
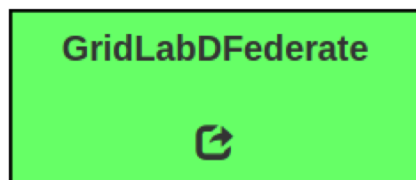
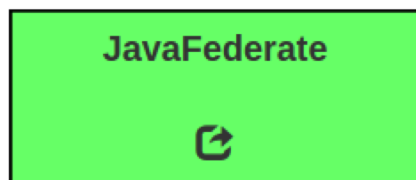
Implement the code

- Import the generated code into Eclipse
- Implement the **TODO** comment blocks

Run the federation

WebGME Model Components

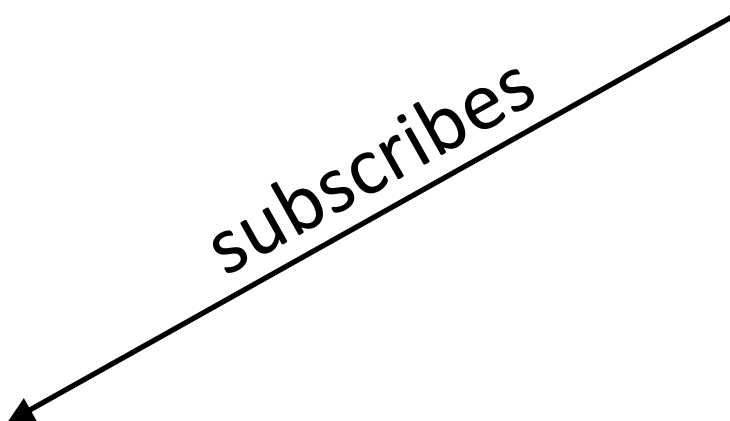
Wrapped Simulators



publishes



subscribes



Message Types

Interaction <C2WInteractionRoot>	
ATTRIBUTES	
originFed:	String
sourceFed:	String
actualLogicalGen...	double
federateFilter:	String

Object <ObjectRoot>	
ATTRIBUTES	
attribute1:	String
attribute2:	double

WebGME Modeling Flow

The screenshot displays the WebGME modeling environment. The breadcrumb path at the top is GME > TutorialProject > master > IntegrationModel. The main workspace shows a diagram of the IntegrationModel with a central green box labeled 'ExampleFederate'. A red arrow labeled 'drag & drop' points from the 'JavaFederate' element in the left 'VISUALIZER SELECTOR' to the 'ExampleFederate' box. Another red arrow labeled 'select & rename' points from the 'ExampleFederate' box to the 'name' field in the 'PROPERTY EDITOR' on the right, which contains the text 'ExampleFederate'.

© 2017 Vanderbilt University version: 2.15.1

IN SYNC NOTIFICATIONS [0] CONNECTED ON

This video illustrates how to use WebGME to create a federation and generate code.

<https://s3.amazonaws.com/nist-sgcps/UCEF/videos/CreateAPingPongFederationAndCode.mp4>

UCEF Work Flow

Model the federate

- Choose the federate type
- Create/import objects and interactions
- Define publications and subscriptions

Generate the code

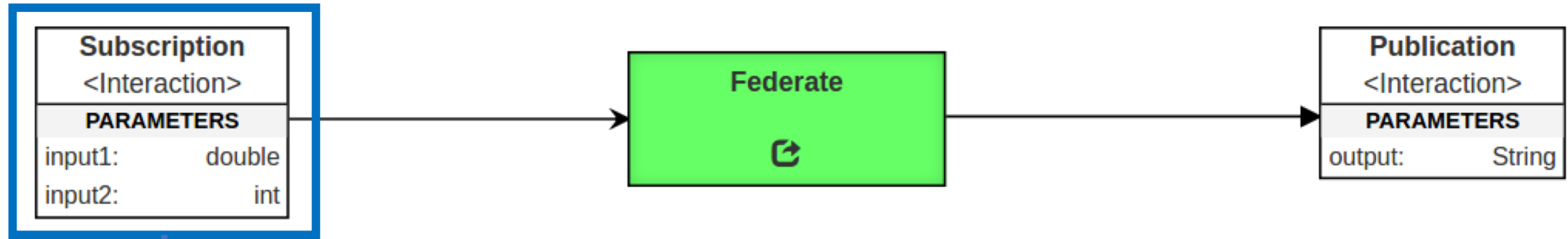
- Run the Federates Exporter
- Run the Deployment Exporter

Implement the code

- Import the generated code into Eclipse
- Implement the **TODO** comment blocks

Run the federation

Code Generation: Subscriptions



```
private void handleInteractionClass(Subscription interaction) {
    //////////////////////////////////////
    // TODO implement how to handle reception of the interaction
    //
    // double input1 = interaction.get_input1();
    // int input2 = interaction.get_input2();
    //////////////////////////////////////
}
```


Code Generation: Publications



```

////////////////////////////////////
// TODO send interactions that must be sent every logical time step below.
// Set the interaction's parameters.
//
// Publication vPublication = create_Publication();
// vPublication.set_output( < YOUR VALUE HERE > );
// vPublication.sendInteraction(getLRC(), currentTime + getLookAhead());
////////////////////////////////////

```

This video illustrates how to use the Eclipse IDE to add behavior to the PingPong federation and run the resulting code.

<https://s3.amazonaws.com/nist-sgcps/UCEF/videos/CompilingEditingRunningPingPong.mp4>

Collaborate with us!

- For General Information on UCEF:
 - NIST UCEF Information: <https://www.nist.gov/el/cyber-physical-systems/ucef-universal-cps-environment-federation>
- For Collaboration and Technical Developer Details:
 - NIST UCEF Collaboration Site: <https://pages.nist.gov/ucef/>
- For Source Code and Building / Downloading the UCEF Virtual Machine:
 - UCEF Development Site: <https://github.com/usnistgov/ucef>
 - UCEF Virtual Machine Issues List: <https://github.com/usnistgov/ucef/issues>
- For Email List Discussion (New)
 - UCEF Mail List: ucef+subscribe@list.nist.gov