



Technical Description of C2SIM

October 2023

Dr. J. Mark Pullen

George Mason University C4I & Cyber Center

Director Emeritus

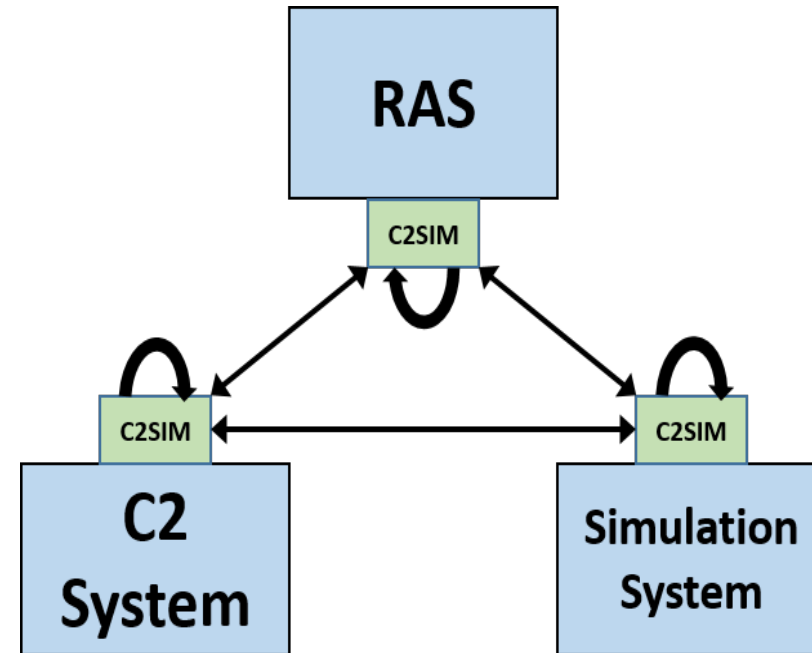
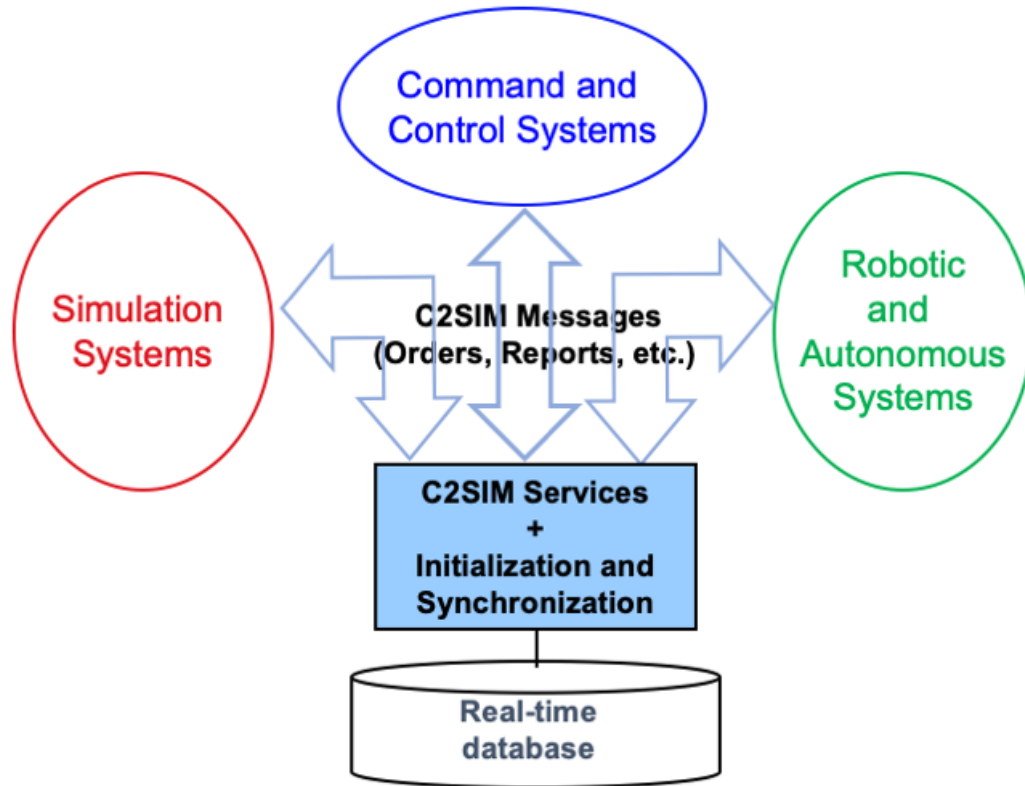
16 July 2023



Outline

- C2SIM and FMN Standards (SISO, STANAG, GitHub)
- C2SIM Ontology to Schema transformation
- C2SIM Communication
- C2SIM XML document flow
- C2SIM Infrastructure
 - Server/ClientLib, Editor, Control, and Sandbox

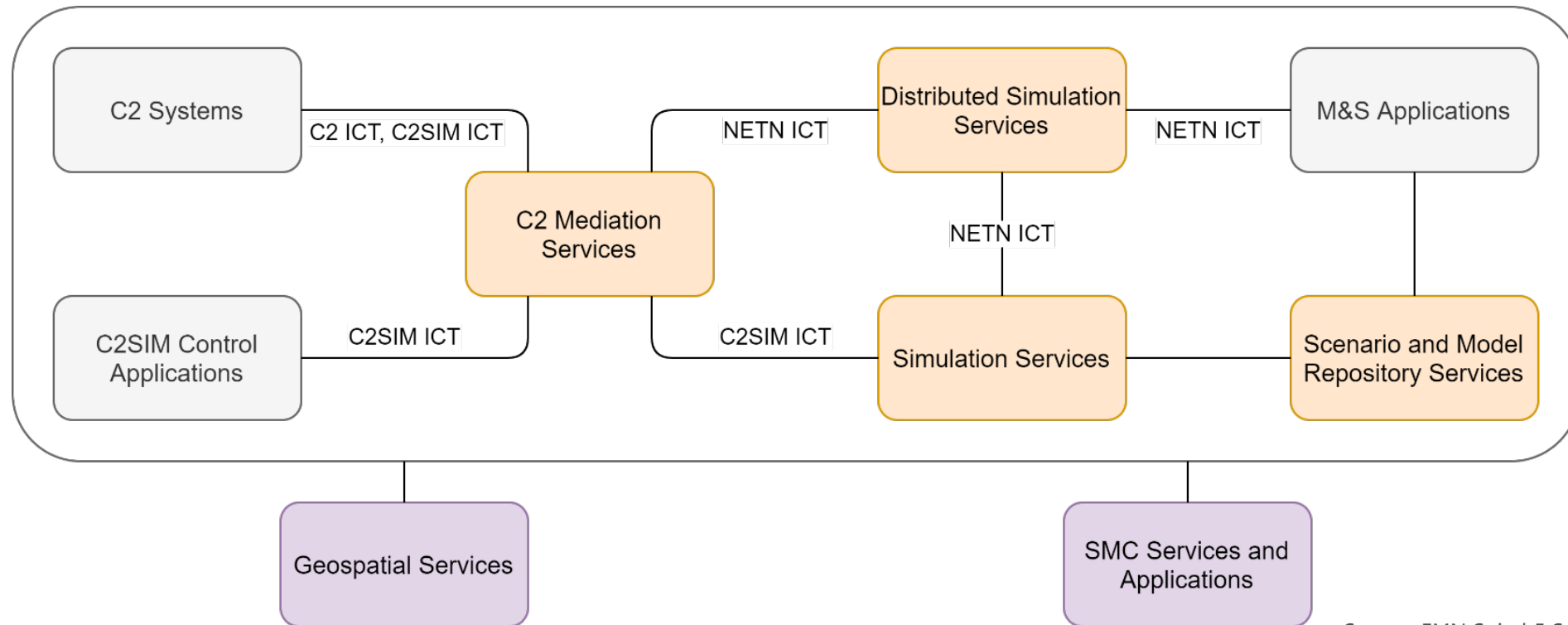
C2SIM Concept



Foundation for
specification of
information
interchange across
broad classes of
systems

Nine-way interactions
(inter-system and extra-system)

M&S Architecture in FMN



Source: FMN Spiral 5 Service Instructions for Modelling and Simulation, 2022.

C2: Command and Control; C2IS: C2 Information System; C2SIM: C2 Systems – Simulation Systems Interoperation; ICT: Initialization, Control, Tasking and Reporting interactions; M&S: Modelling and Simulation; NETN: NATO Education and Training Network

OpenC2SIM on GitHub

<https://OpenC2SIM.github.io>

C2SIM Standard Balloted March 2020

- SISO develops international, open standards
- Initial versions of C2-simulation interop standards
 - Military Scenario Definition Language (MSDL) supports initialization
 - Coalition BML (C-BML) provides for exchange of Tasking (orders and requests) and Reporting information based on MIP JC3IEDM
- Unified second-generation standard: C2SIM
 - C2SIM Core and Standard Military Extension (SMX) Ontologies
 - Messaging: Initialization/Synchronization + Tasking & Reporting
 - Extension Mechanism and Land Operations Extension
 - Guidance document

C2SIM STANAG

- STANAG is a NATO Standardization Agreement
- Mostly based on industry standards these days
- MSG-145 sponsored SISO C2SIM to be a STANAG
 - With NATO M&S Coordination Office support
 - STANAG 4856 Edition 1 approved March 2023
 - Now the “official” NATO way to achieve C2-simulation interoperation

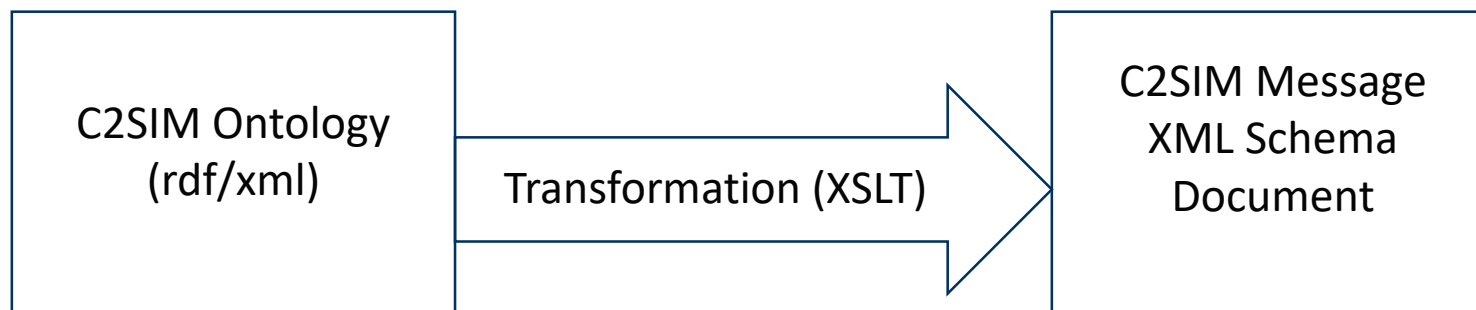
C2SIM Ontologies

- Ontology: Set of concepts and categories in a subject area or domain that shows their properties and the relations between them
- Core: data classes and properties that are needed by all C2 and simulation systems to interoperate: Who, what, when, where
- Standard military extension (SMX): classes and properties that are needed by all military C2 and simulation systems
 - Mostly more properties for core classes, e.g. Entity has a ForceSide
- Land Operations Extension (LOX): ground warfare classes and properties
 - Separate standard; example for other new extensions

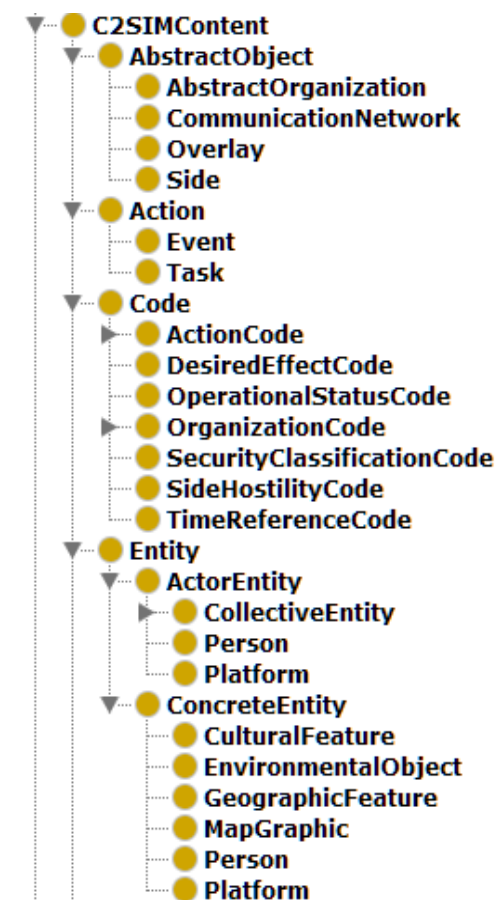
Ontology Provides Stronger Semantic Specification

- Semantic interoperability is achieved by assuring that multiple systems share a common vocabulary set of meanings (semantics)
- Literature points to the use of ontologies for stronger semantic formalization of data models
- The World Wide Web is evolving into a Semantic Web of information based on a set of accepted representations
- The Web Ontology Language (OWL) was used as the specification language for C2SIM, the new SISO standard for Command and Control Systems – Simulation Systems Interoperation
- The use of ontology creates opportunities for automated reasoning, query, and rule-based reasoning at the scale of the Web

Transformation to XML for Developers



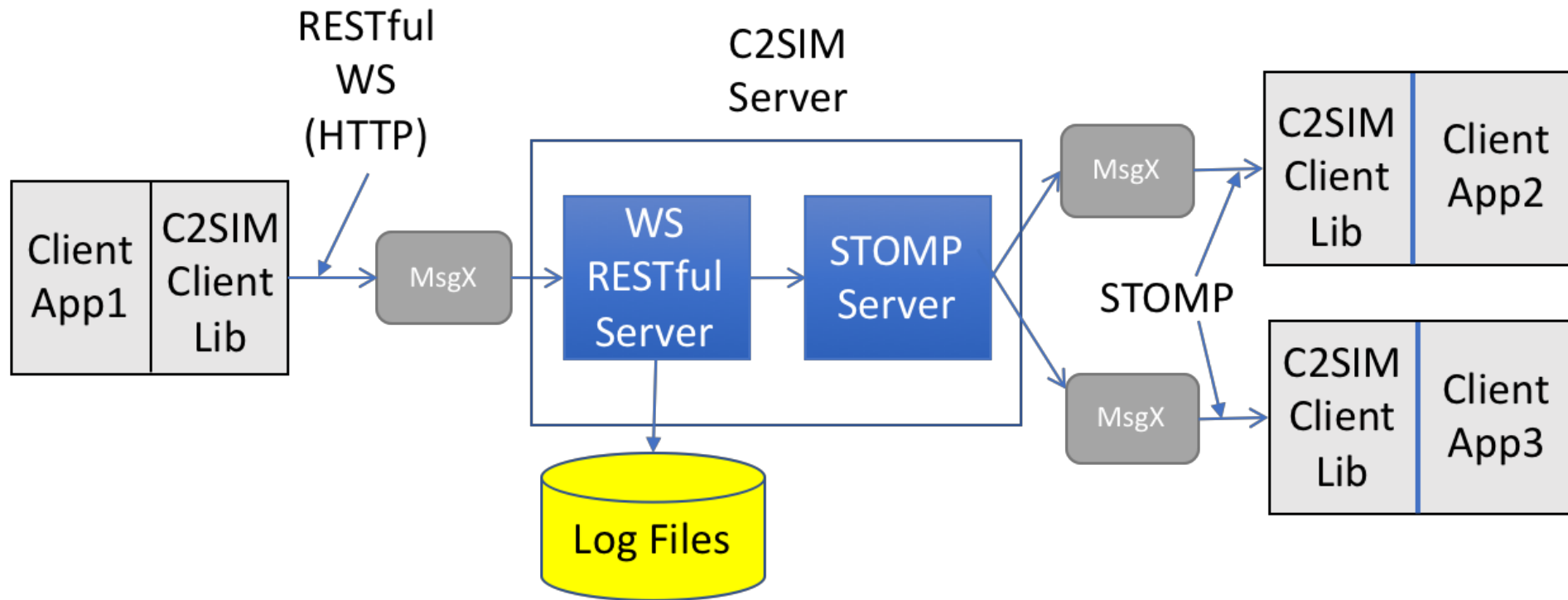
- XSLT (Extensible Stylesheet Language Transformations)
 - World Wide Web Consortium Recommendation (<https://www.w3.org/TR/2017/REC-xslt-30-20170608/>)
 - XML language for translating XML documents into text-based documents (including other X)
- Fraunhofer FKIE has developed a Java implementation
- Start with a C2SIM ontology file (core plus zero or more extensions) expressed in rdf/xml format produced by the Protégé ontology editing tool
 - Resource Description Framework (RDF)
 - Protégé: open-source tool available at <https://protege.stanford.edu>



C2SIM Communication

- Messages coded in eXtensible Markup Language (XML)
 - Data structured as “tree” expanding from root
 - Each data element has descriptive “tag”
- Communication via “Web Service” server
 - Technology grew out of World Wide Web (WWW)
- Input REpresentational State Transfer (REST)
 - Document submitted in temporary TCP/IP connection
- Output Streaming Text-Oriented Message Protocol (STOMP) to subscribing group of clients
 - Document forwarded via sustained TCP/IP connection to all C2 & simulation systems that subscribe to “Topic”

Current Web Service Implementation of C2SIM Group Communication



C2SIM Messaging

- Standard header for all C2SIM messages
 - Implements IEEE FIPA formal communication rules
 - Handled by standard library to ease implementation
- Standard message bodies for various required content
 - Object and System Initialization information
 - Domain Messages: Order, Report and Plan information
 - System Commands: Synchronize initialization and execution
 - Acknowledgement: Confirm information receipt where needed

Roles of C2 and Simulation Client Systems

C2 systems

- Produce Orders/Requests and consume Reports
- Send Orders to server by REST
- Subscribe to Reports from server by STOMP
- Ideally, able to start/pause/stop simulation

Simulation systems

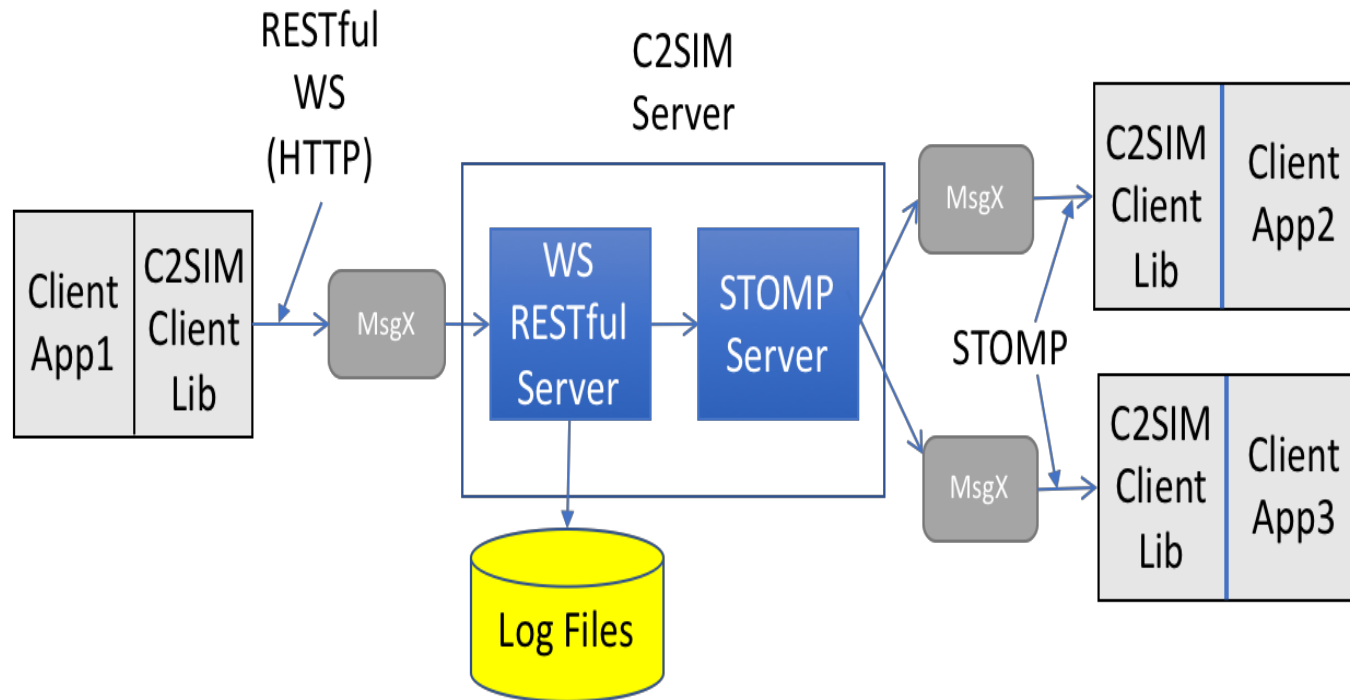
- Consume orders/requests and produce reports
- Subscribe to Orders/Requests from server by STOMP
- Send Reports to server by REST
- Controllable to start/pause/stop simulation
- Produce log of activities for replay/restart

C2SIM XML Document Flow

- Client Application creates message
- Client Application passes message to ClientLib
- Message submitted to Web Services server via REST
- REST Server processes message
- Server sends message to be published to STOMP server
- STOMP server sends message to all subscribed clients
- Client Library passes published message to client

Big Picture

NOTE: most Apps both send REST and receive STOMP



C2SIM Infrastructure

Current C2SIM Reference Implementation

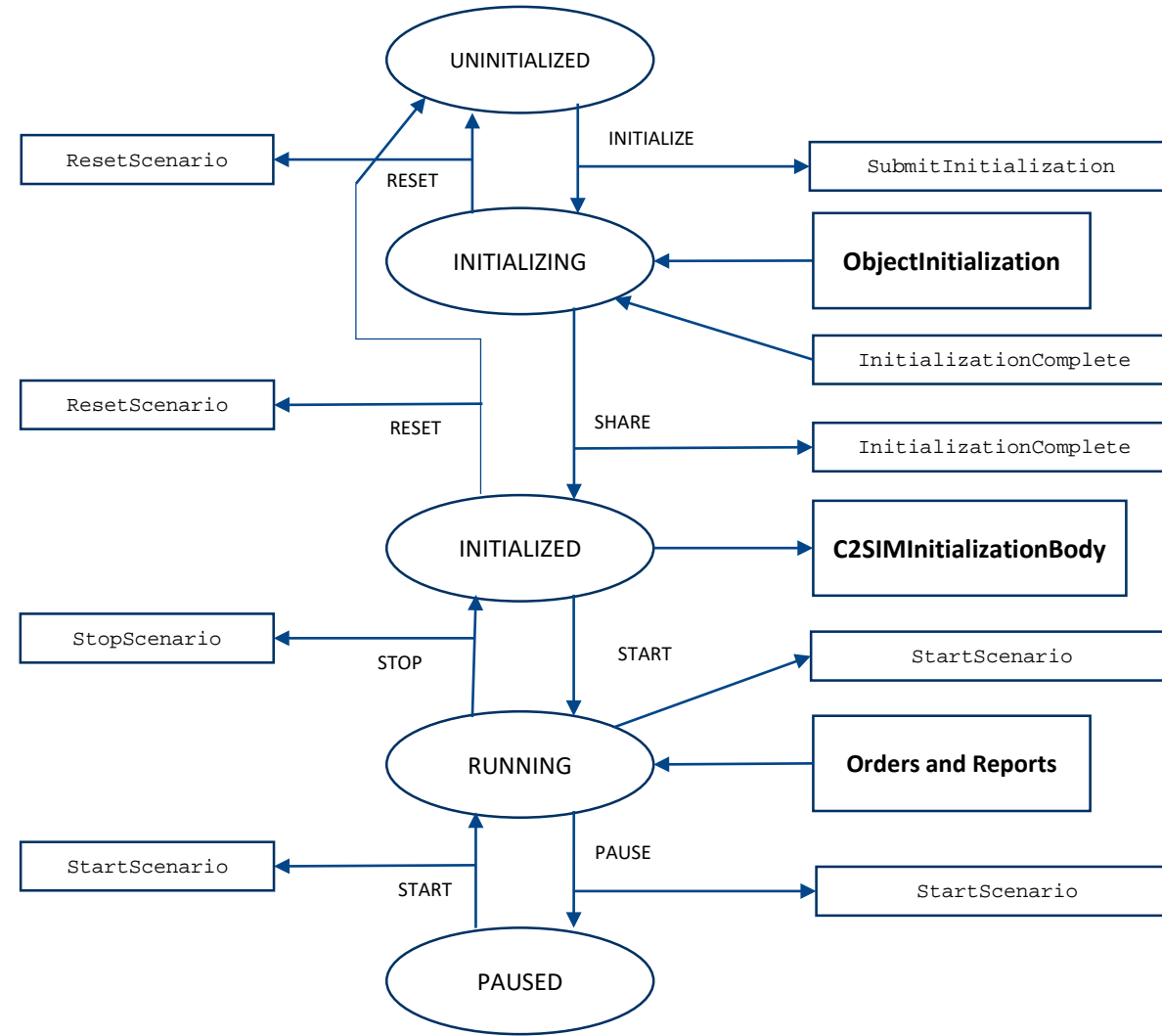
Open Source Java Server Functions

- Supports a *Coalition* of C2 & simulation systems
- REST input, STOMP publish-subscribe output
- Client Library assists integration (Java and C++)
- Translating feature allows MSDL/C-BML compatibility
- Reports and Orders among C2SIM standard, CBML Light and IBML09
- Initialization between MSDL and C2SIM initialization
- Supports C2SIM coalition and record/playback synchronization messages
 - Initialize, Ready, Start, Stop, Pause
- Logs all transactions and supports playback
- Can supports distributed servers using B2B Client

Open Source COTS Server Environment

- Centos 7 Linux Server
- running in a VMWare Virtual System
- Tomcat 8.0.30 Web Service Application Server
- Apache Apollo 1.7.1 STOMP Messaging

C2SIM Standard Exercise Control



C2SIM Coalition/Server User Control intended for C2 users

C2SIM Control version 1.0.3

C2SIM C2 USER CONTROL FOR SERVER 10.100.13.28

Simulation Coalition state: UNINITIALIZED

Playback state: NO_PLAYBACK_IN_PROGRESS change: PLAY

Player speedup: 1.0 enter here to change:

C2SIM SERVER CONTROL

Simulation Coalition state: UNINITIALIZED change: INITIALIZE

Recorder state: RECORDING_IN_PROGRESS change: PAUSE STOP

Scenario speedup: 1.0 enter here to change:

Save Revert Point CHECKPOINT SAVE

Revert to Last Saved Point CHECKPOINT REVERT

Distributed C2SIM Servers

■ How:

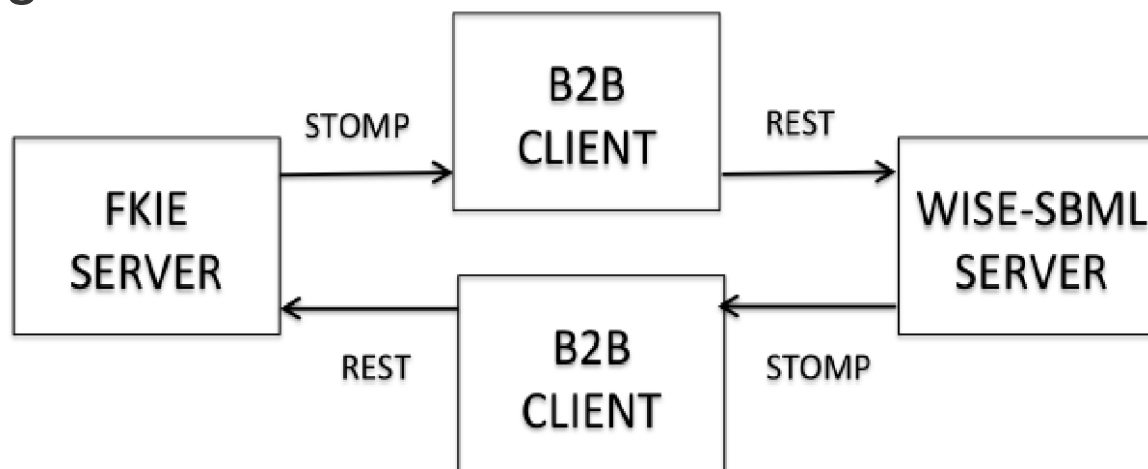
- Build an interface that subscribes to another server and forwards its published documents
- Simple implementation: back-to-back (B2B) client
- Ensure that forwarded messages cannot loop

■ Why:

- Reduced network load by clustering
- Sharing publish workload

Back-to-Back Clients

- Building block for distributed servers:

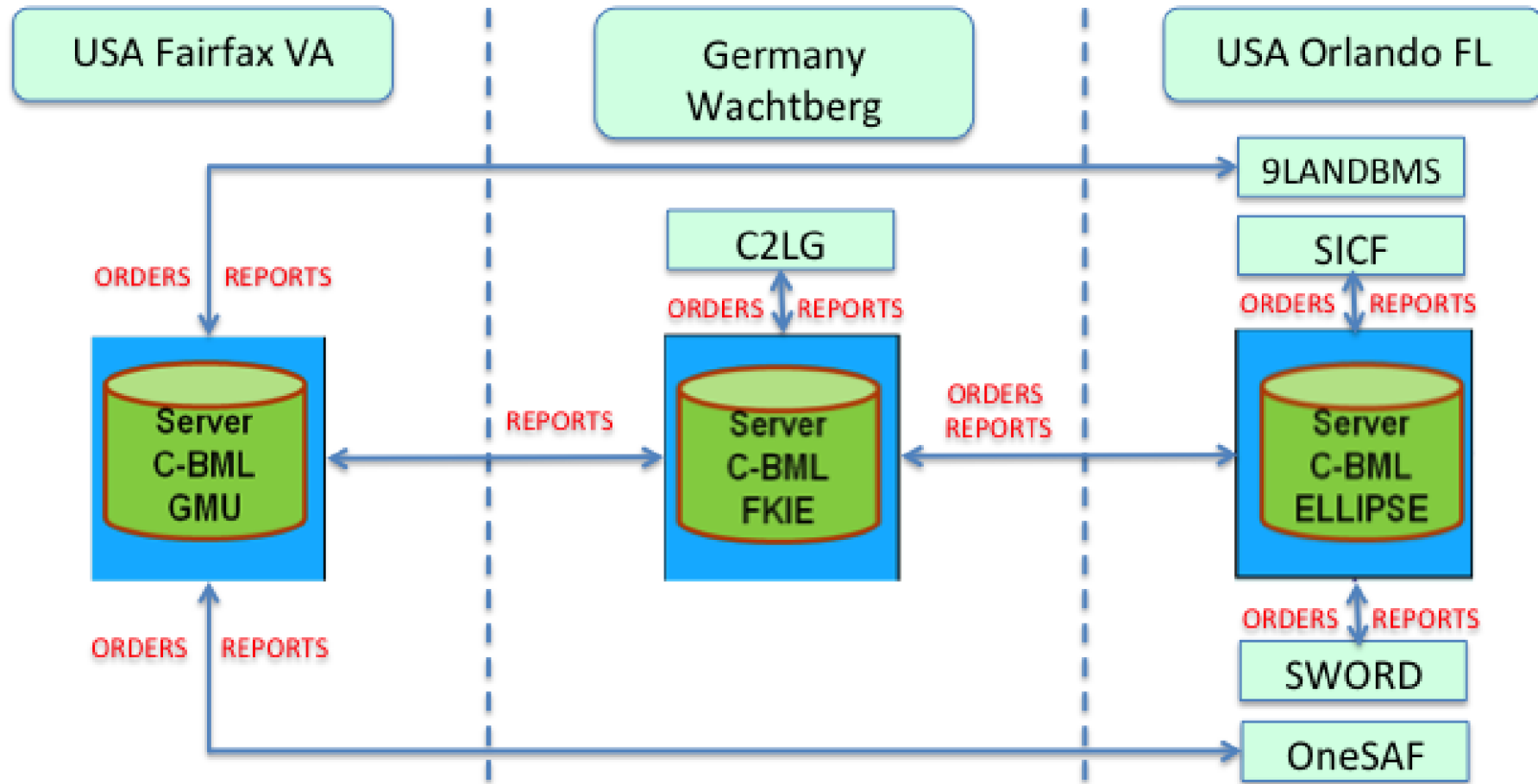


- Used in:
 - MSG-085 Final Demonstration (USA-Europe)
 - I/ITSEC-2014 Demonstration (France-Germany-USA/Sweden)
 - CWIX 2019 C2SIM testing and MiniEx (USA-MSCOE)

Precluding Distributed Server Loops

- Label each published message with ID of forwarding server(s)
 - Install filter in server and/or B2B client
 - Drop any message receiving server has already forwarded
 - Supported by C2SIM Reference Implementation
- Configure servers in a star (tree)
 - Guarantees no “back-door” loops make multiple delivery via different path than first delivery

Three-Server Demo I/ITSEC 2014



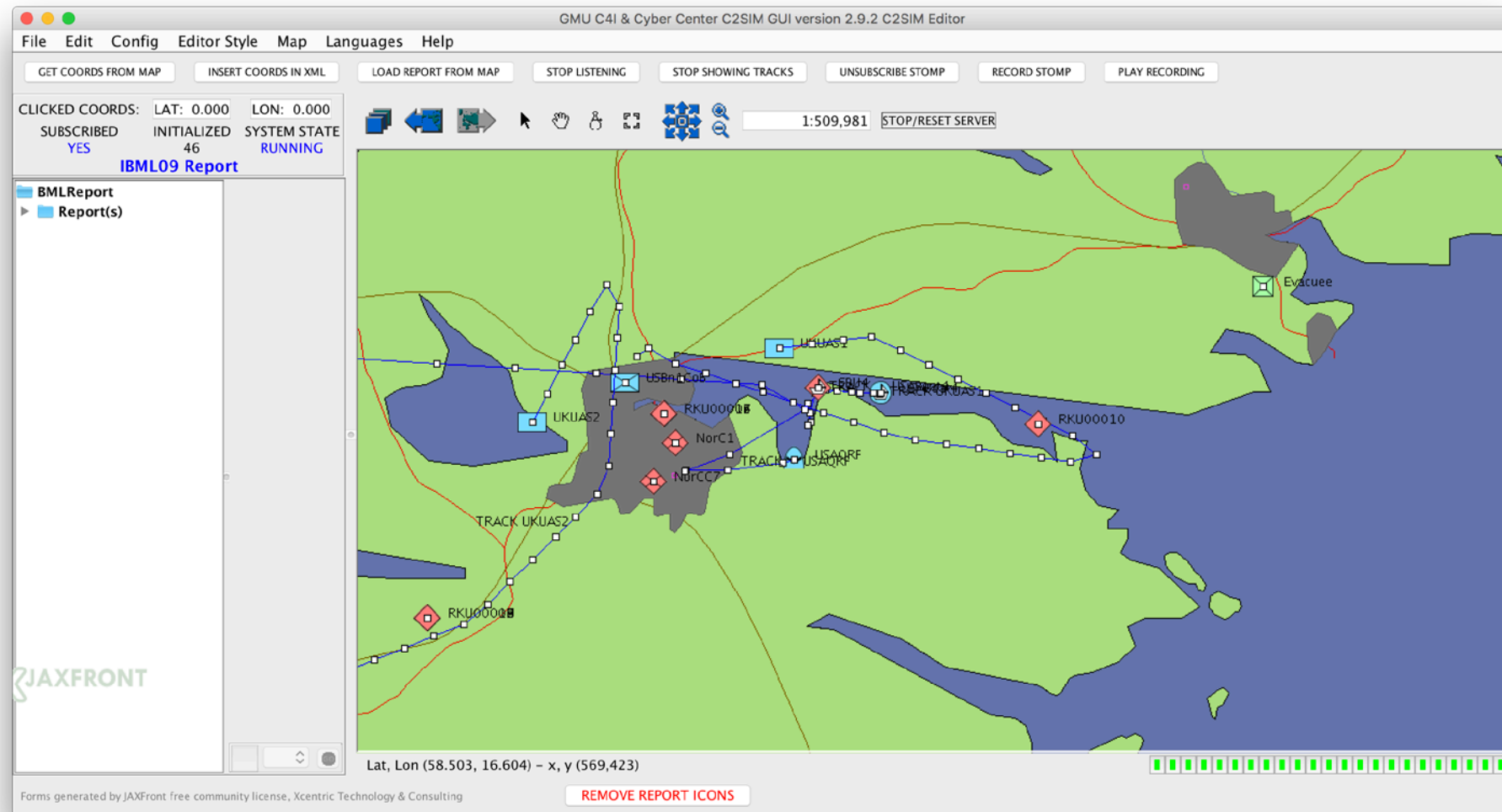
C2SIM Reference Implementation Client Libraries

- Implement common messaging functions
 - Support IEEE FIPA
 - Support REST and STOMP
- C2 and Simulation client implementation simple
 - Instantiate REST and STOMP modules/connections
 - Send and receive REST (similar to webpage I/O)
 - Listen for STOMP (blocking or non-blocking)
- Available for Java and C++ on GitHub [OpenC2SIM](#)

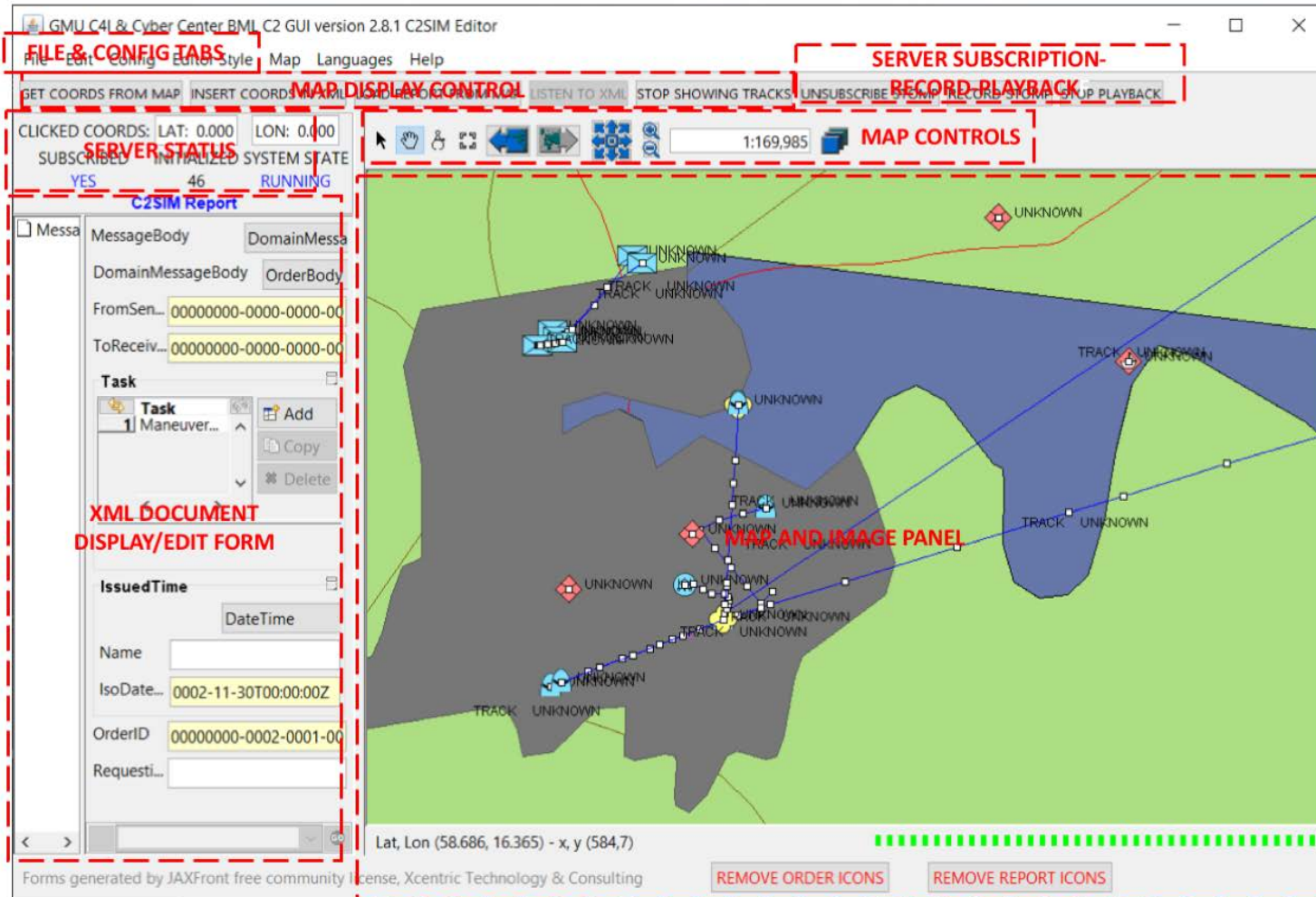
C2SIMGUI Editor

- interacts with a C2SIM server by
 - creating and/or editing XML Order and Report files
 - pushing such files into the server
 - subscribing to receive such files
 - displaying their tactical graphics (unit icons and related graphical control measures) on a map or image
 - sending and receiving C2SIM synchronization messages
 - C2SIMcontrol is better for C2 users
- open source software available on GitHub OpenC2SIM
 - patterned after similar C2LG GUI
 - developed by Fraunhofer-FKIE but not available open source
 - The two have diverged due to different research interests
- ***NOT* a real C2IS but has been used experimentally as surrogate for one**

C2SIMGUI As Used in CWIX

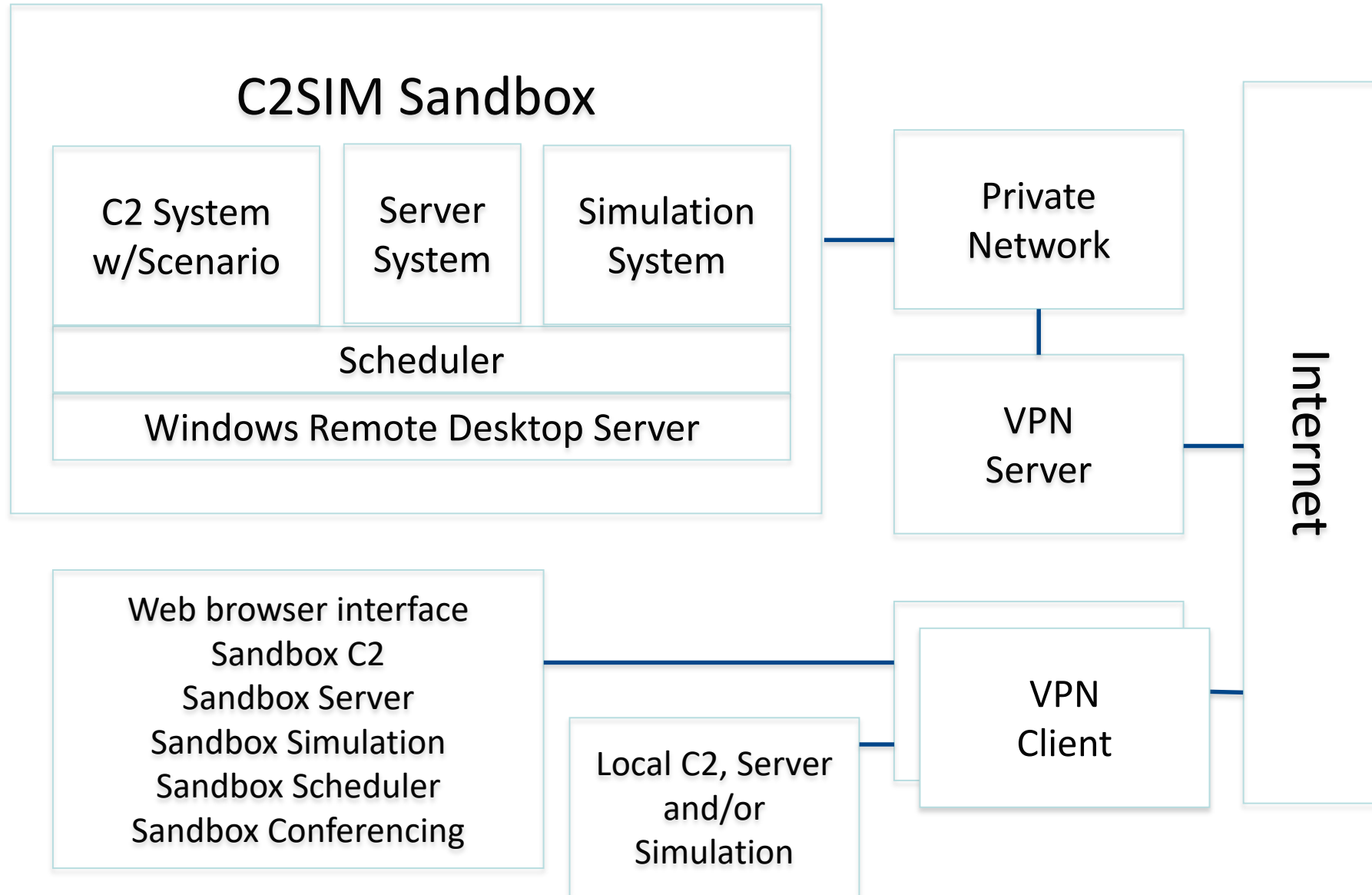


C2SIM GUI Layout from User Guide



C2SIM Sandbox

- Assembled to support distributed development by MSG-145
- Contains a complete C2SIM system
 - C2SIMGUI as surrogate for C2IS
 - Reference Implementation C2SIM Server
 - MÄK VR-Forces Simulation (requires license)
- Available to run over OpenVPN system with GMU license
- Includes Jitsi but more often used with Zoom
 - Jitsi server can be inside VPN; Zoom is commercial
- Accessible as remote desktop inside VPN
 - User needs only Google Chrome to access Sandbox
- VPN can include C2 or Simulation under development



Ways to Use C2SIM Sandbox

- C2SIM demonstrations
 - Initially IBML09
 - C2SIM standard as soon as we can prepare it
 - With generic scenario (others if contributed)
- C2SIM testing
 - Test C2 with Sandbox Server and Simulation
 - Test Server with Sandbox C2 and Simulation
 - Test Simulation with Sandbox C2 and Server
 - Test C2-Simulation Coalitions with the Server
 - Distributed configurations of all sorts
- C2SIM validation with SISO
- Limited-scope C2SIM-based exercises
- In the future: C2SIM as a Service