Technical Description of C2SIM

J. Mark Pullen
Director Emeritus, C4I & Cyber Center
George Mason University
UNITED STATES

mpullen@gmu.edu

ABSTRACT

NATO Modelling and Simulation Group 211 (MSG-211) developed a Research Technical Course titled "Modelling and Simulation Standards in NATO Federated Mission Networking." The course will be presented in 2023 and 2024 in hybrid format. This Educational Notes paper presents the course content for the topic "History of Standardized C2-Simulation Interoperability (C2SIM)," one of 16 topics presented during the overall course. The paper describes technical details of C2SIM: standards, architecture, communication, and infrastructure. The approaches described in this paper are intended to be illustrative, not exhaustive. As the engineering community works with M&S in FMN, expanded alternatives will be identified.

1.0 C2SIM STANDARDS

The C2SIM Standard, developed by SISO with NATO M&S Group support and adopted by NATO as STANAG 4856, specifies mechanisms to define data exchange among simulation and C2 systems and messaging approach to achieve that data exchange. The general C2SIM architecture is shown in Figure 1; an alternate view including Robotic and Autonomous Systems in shown in Figure 2.

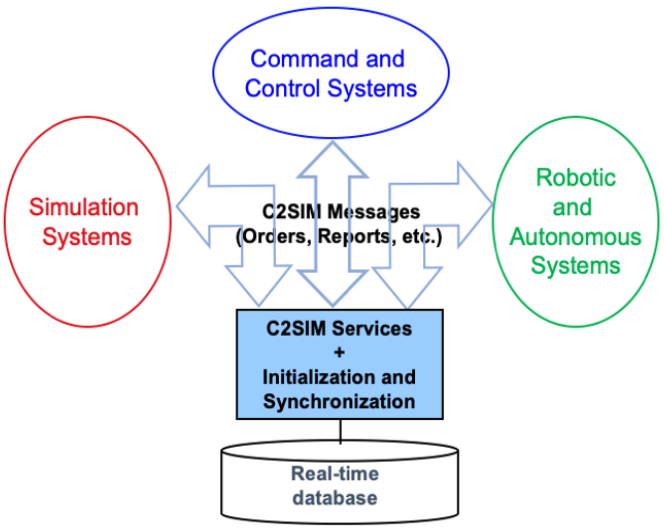

Figure 1: General C2SIM architecture.

The C2SIM Standard was developed by SISO, the Simulation Interoperability Standards Organization, an industry standards group that uses teams with industry, academic, and government people to develop and support open standards. It was developed in an open, balloted process (see https://sisostds.org). Most of

these are for distributed operation and military use. C2SIM has been adopted (like the SISO/IEEE HLA standard) as a NATO Standardization Agreement that NATO nations should use to work together for C2-simulation interoperation.

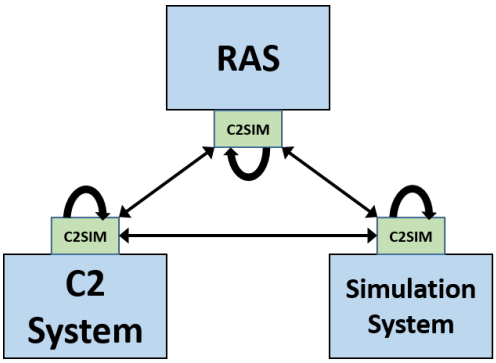

Figure 2: C2SIM architecture including robotic and autonomous systems.

A top-level description of the C2SIM standard's history is:

- 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

Figure 3 shows how C2SIM can be combined with other M&S standards presented in this course, to support FMN.

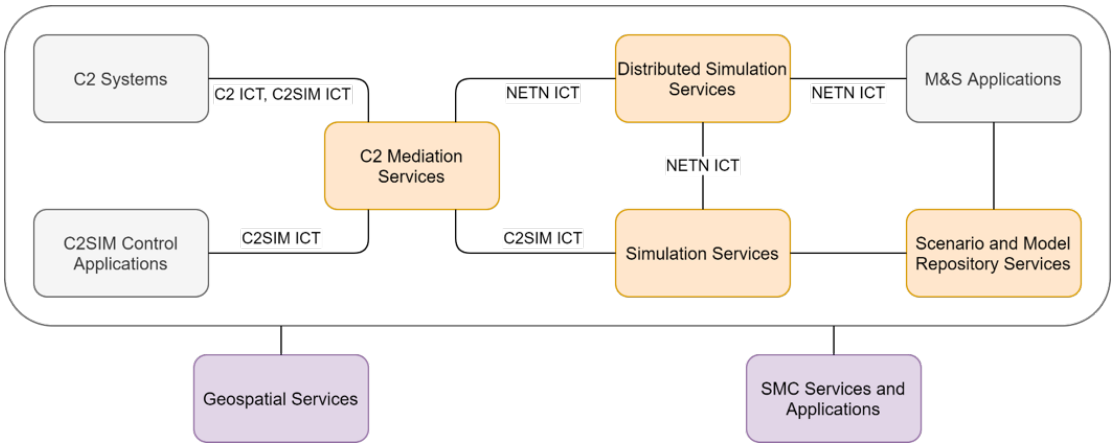

Figure 3: General architecture for M&S in FMN, including C2SIM.

2.0 C2SIM ONTOLOGY

A key aspect of the C2SIM standard is its incorporation of an ontology to define the data model used. An ontology is set of concepts and categories in a subject area or domain that shows their properties and the relations between them. The ontologies defined by the C2SIM standards [1], [2], [3] are:

- **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 (this is a separate standard, and serves as an example for other new extensions.

There is significant momentum in the academic, commercial, and government arenas for adoption of stronger semantic representations of data. Class 2.5 in this course focuses on some of the important aspects that formed a foundation for the technical approach to the C2SIM specification.

Automated generation of a workable XML schema is a practical enabler for early adopters of the C2SIM standard. Developers can continue to apply the technical approach with which they have many years of experience. Exploitation of capabilities enabled by the strong semantic representation will emerge as developers gain deeper understanding in the ontology representations (which also are available as XML but have other renderings as well). Figure 4 shows this process.

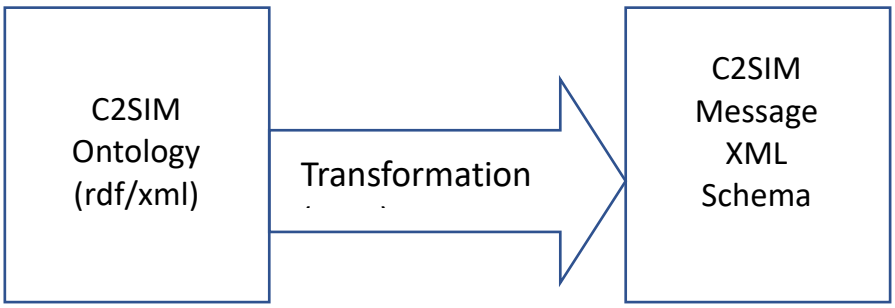

Figure 4: Ontology to XML schema transformation.

3.0 C2SIM COMMUNICATION

C2SIM requires group communication (often called “multicasting”): when a system sends a message, all others in its group receive that message. This can be refined by the use of Topics, where the sent message is assigned a Topic that is used to filter which messages are delivered to individual receivers. Currently, this concept is implemented in operating C2SIM systems via Web Service technology, as shown in Figure 5. The process works like this:

- Messages are 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 and simulation systems that subscribe to "Topic".

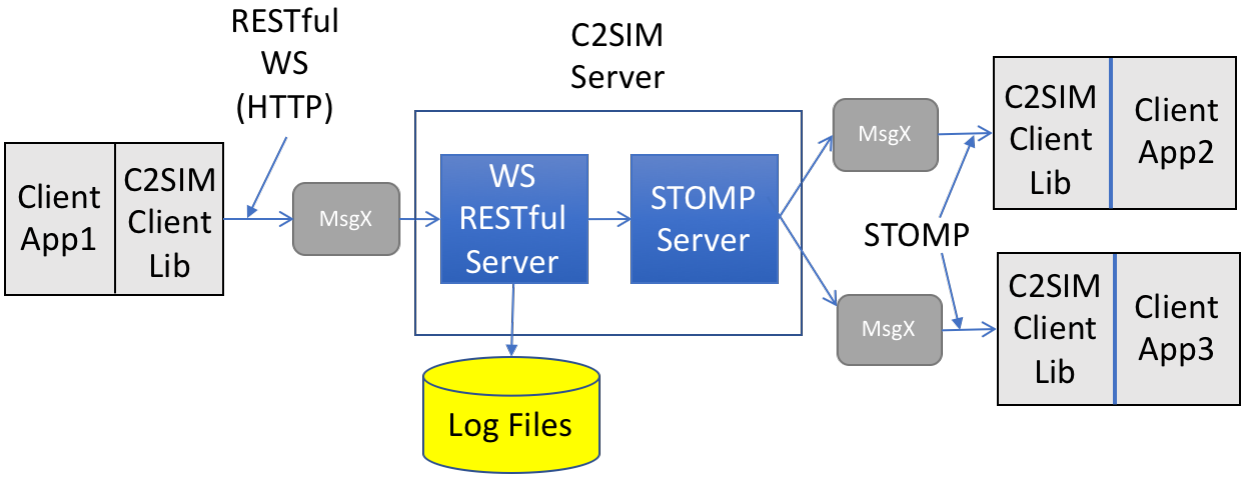

Figure 5: Web service implementation of C2SIM communications.

The C2SIM standard defines formats of messages to be sent using a standard header for all C2SIM messages, which implements IEEE FIPA formal communication rules for reliability. Typically, header format is handled by a shared library to ease implementation. There are standard message bodies that support several functions: Domain Messages(Orders, Reports, and Plans), Initialization Messages. System Messages that synchronize the initialization and operation of C2SIM Coalitions, and Acknowledgement Messages used where reliable transmission is required.

C2SIM messages are used as follows:

- *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

Resulting XML document flow is:

- Client Application creates message
- Client Application passes message to supporting 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

4.0 C2SIM INFRASTRUCTURE

Current infrastructure was inherited from C2SIM development and has been extended and made more robust by multiple years of CWIX testing.

The infrastructure falls into two areas: server including client library) and editors. We provide details here on the open-source software available on the GitHub at https://Open C2SIM.github.io. Other servers have been developed, however none is generally available as open source.

4.1 C2SIM Reference Implementation Server

This server, developed by GMU C4I & Cyber Center, has the following properties [4]:

- 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 support distributed servers using B2B Client

This server operates in the following freeware 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 coalition states coordinated by the server can be controlled by the C2SIM User Control, which has been derived from the C2SIM Editor described in section 4.2. Its graphic interface is shown in Figure 6.

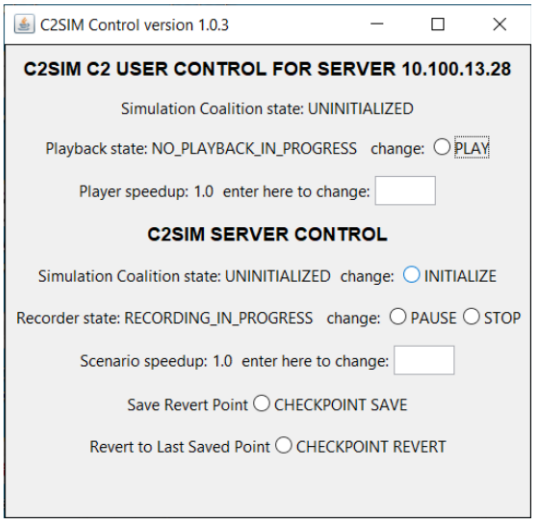

Figure 6: C2 User Control graphic interface.

The Reference Implementation C2SIM Server and associated C2 User Control implements the coalition states shown in Figure 7, which are defined in the SISO C2SIM standard.

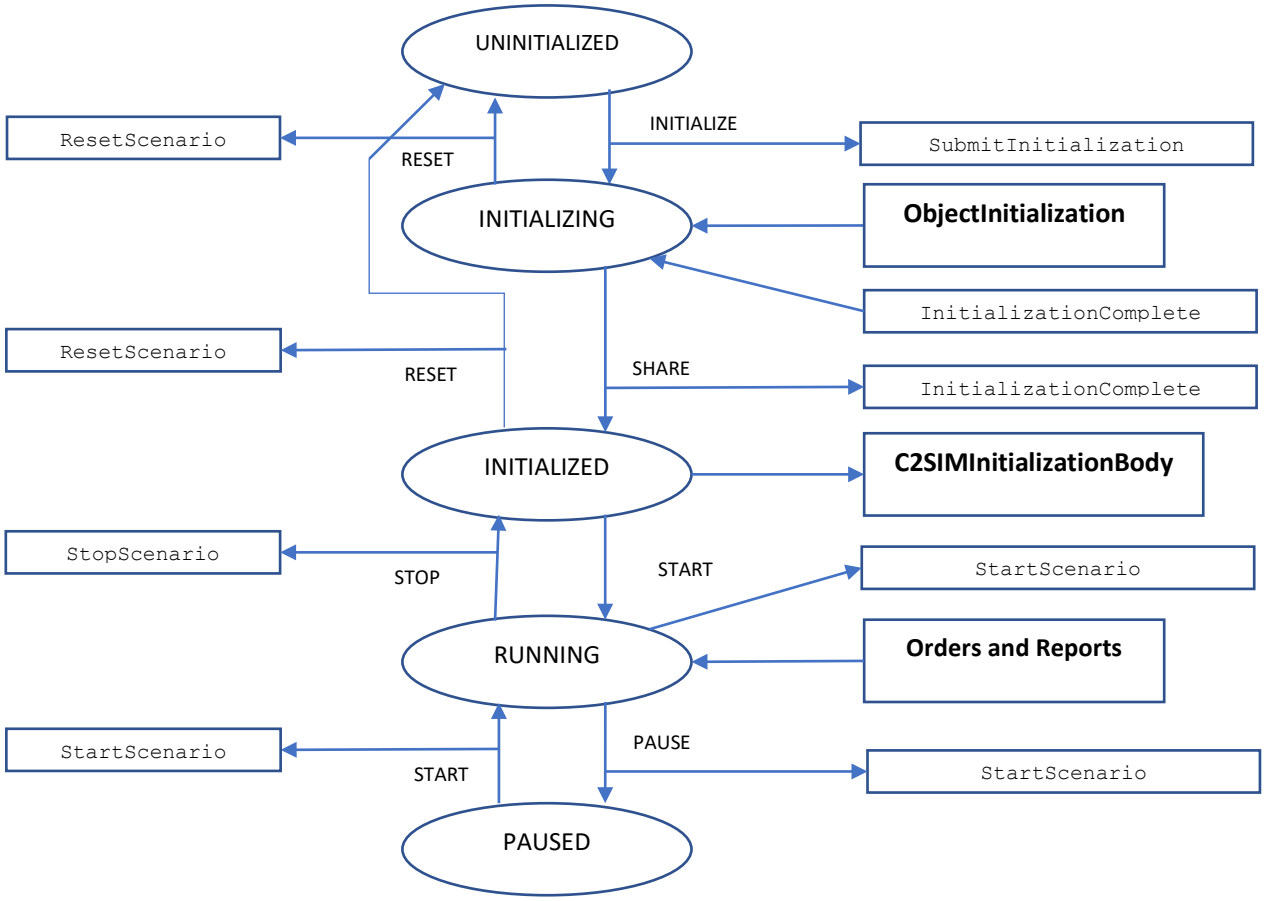

Figure 7: Coalition states supported by reference implementation server.

It is possible to use multiple C2SIM servers for increased performance or increase network efficiency by localizing traffic. Figure 8 shows use of a "back-to-back" (B2B) client to couple servers. This arrangement has been used in:

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

To preclude message loops in distributed servers:

- 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) to guarantee no "back-door" loops make multiple delivery via different path than first delivery.

4.2 C2SIM Editor

The C2SIM GUI is a custom editor for C2SIM. Its properties are Figure 5:

- 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

Another editor that has been used in CWIX is the Fraunhofer FKIE C2LG system. This high quality software preceded C2SIMGUI but unfortunately is not available as open source. It has graphic features superior to the C2SIMGUI as so is more useful as a surrogate C2 system. The C2SIMGUI was developed as open source, to allow a broader range of users.

Figure 8 is from the C2SIMGUI User Guide. It shows the various sections of the GUI frame: File and Configuration tabs; Server subscription and Local record/playback; Map display control; Server status and control; XML document editing; Map control, and an OpenMap Panel.

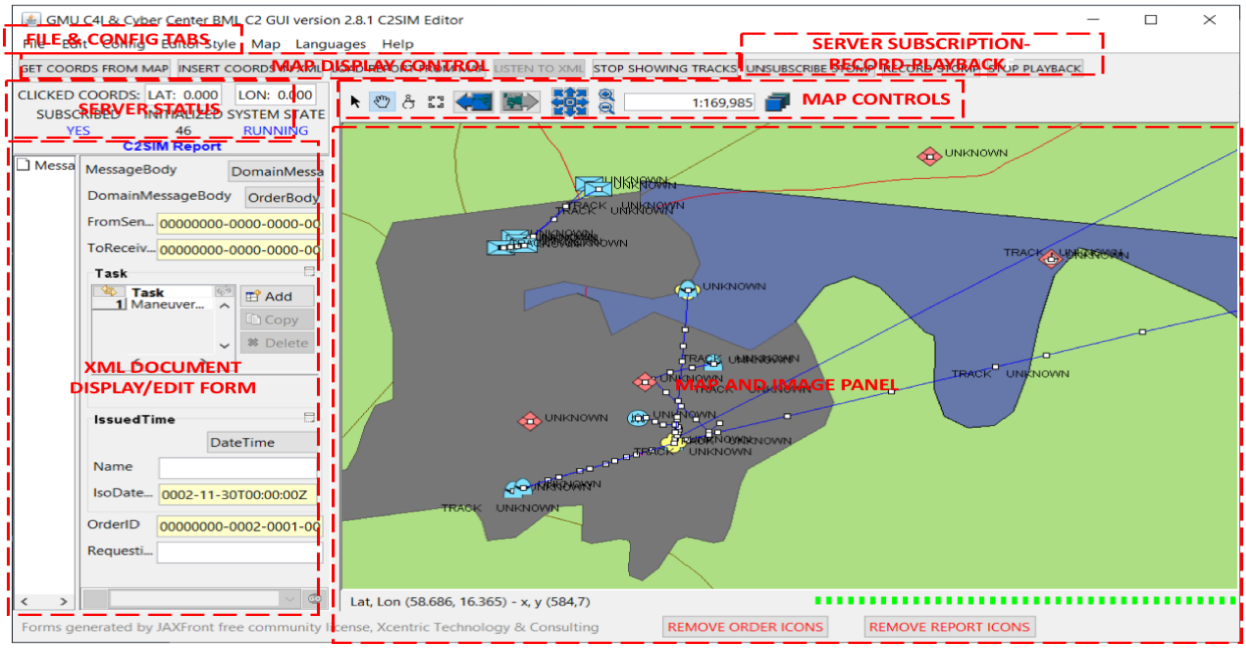

Figure 8: C2SIM GUI graphic interface with function areas identified.

5.0 C2SIM SANDBOX

The C2SIM Sandbox is an assembly of C2SIM software developed by GMU C4I & Cyber Center with some support from NATO STO CSO. It provides support for testing and demonstration of C2SIM systems. It incorporates the Reference Implementation C2SIM Server, the C2SIMGUI, and a C2SIM-enabled version of the commercial simulation system VRForces, all open source except VRForces which requires a license. It is packaged so as to be remotely accessible via Web browser (Google Chrome recommended). The complete Sandbox can be used for demonstrations. Also, it can be used to test some other C2SIM-compliant application (generally, a simulation) during development and has been used in that mode when interfacing new simulations to C2SIM. Figure 9 shows the structure of the C2SIM Sandbox.

There are a lot of ways to use the C2SIM Sandbox. We have used it several times for I/ITSEC demos; when the CWIX participants were installing heir interfaces they used it for testing; in CWIX we used it for validation testing; and it can support a small exercise as it did in the CWIX Mission Rehearsal exercise. Here are some interesting possibilities:

- C2SIM demonstrations
  - 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 or CWIX
- Limited-scope C2SIM-based exercises
- In the future: C2SIM as a Service?

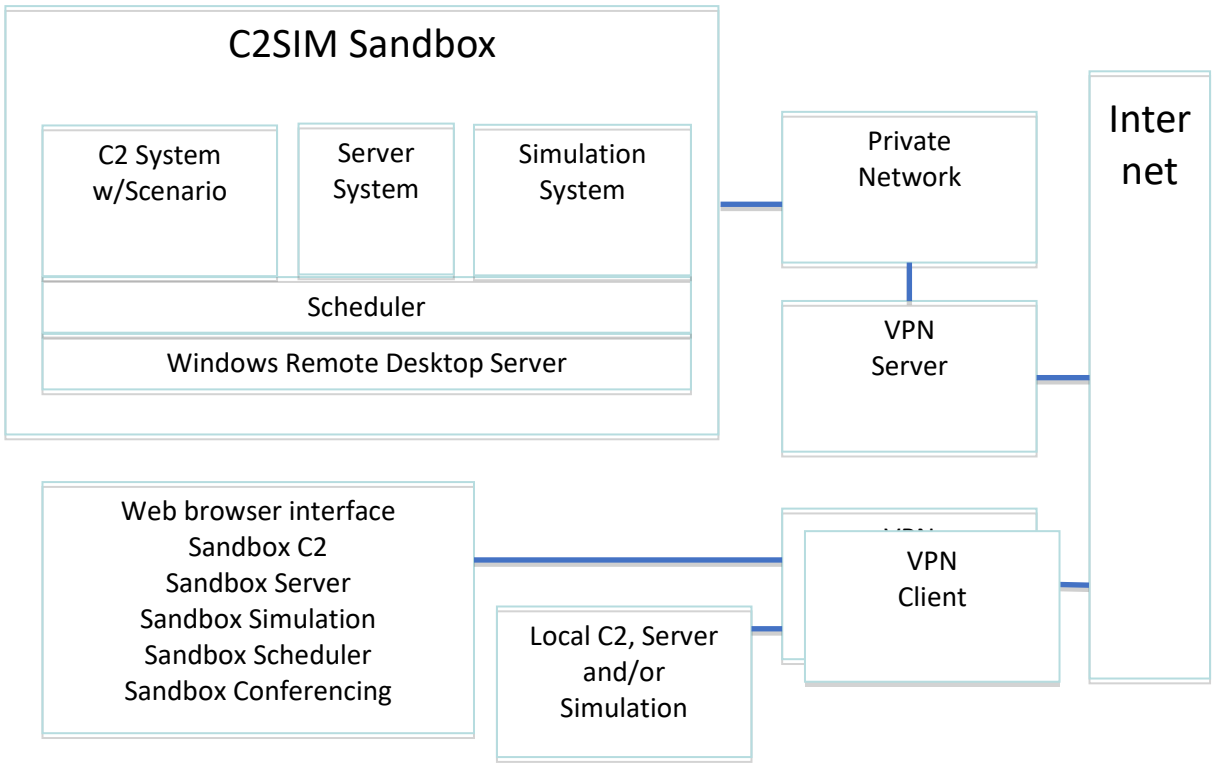

Figure 9: C2SIM Sandbox architecture.

6.0 REFERENCES

[1] SISO-GUIDE-010-2020 Guide for Command and Control Systems – Simulation Systems Interoperation.

[2] SISO-STD-019-2020 Standard for Command and Control Systems – Simulation Systems Interoperation.

[3] SISO-STD-020-2020 Standard for Land Operations Extension to Command and Control Systems – Simulation Systems Interoperation.

[4] https://github.com/OpenC2SIM/OpenC2SIM.github.io/blob/master/C2SIMGUI_User_Guide_v2.12.1.pdf

[5] https://openc2sim.github.io/C2SIMServerReferenceImplementationDocumentation4.8.2.3.pdf