

# **C2-Simulation Communications Infrastructure**

Dr. Mark Pullen  
C4I Center  
George Mason University

# Presentation Overview

- Introduction/overview
- Categories of C2-Sim Infrastructure
- Server systems
- GUI systems
- Conclusion

# Introduction/Overview

- This presentation relates to the communications infrastructure used by US and NATO MSG organizations in the process of conducting MSG-085 and its predecessors
- The audience is presumed to have learned about the primary SISO standards for C2-Sim from other presentations in this workshop:
  - Coalition Battle Management Language (C-BML)
  - Military Scenario Definition Language (MSDL)

# BML: Definition

## WHERE WE STARTED

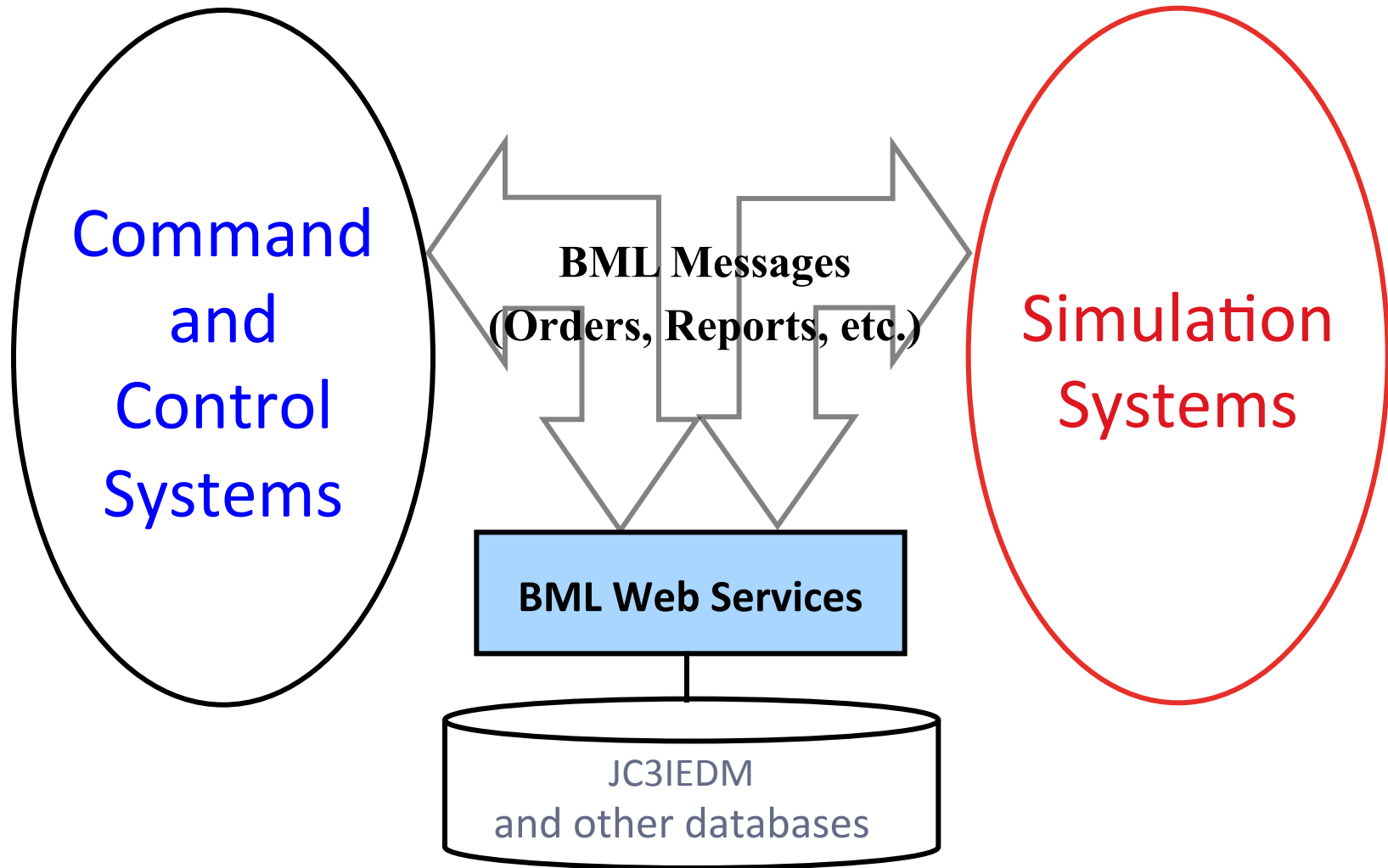
BML - an unambiguous language to:

Command and control live and simulated forces conducting military operations, and

Provide for situational awareness and a shared, common operational picture.

*Shared Semantics between C2 and M&S via a Common Tasking Description*

# Generic BML Architecture



# What is a Web Service?

- Generalization of servers that provide webpages in the Internet
- Functions on a single transaction basis (stateless)
  - Get/Pull and Post/Push
- Makes a TCP connection for each one
- Can be seen as a Remote Procedure Call
  - Using Simple Object Access Protocol (SOAP)
- Or as a way to share and fetch XML documents
  - Could use Representational State Transfer (RESTful)

# Why XML?

- Extensible Markup Language (XML) uses a generalization of the HyperText Markup Language (HTML) used for webpages
  - Information expressed as `<tag>value</tag>`
  - For example `<title>Battle Management Language</title>`
  - Permissible tags defined by a “schema”
- Defined set of metadata tells how to use document
- Organized as a “tree” starting from “root”

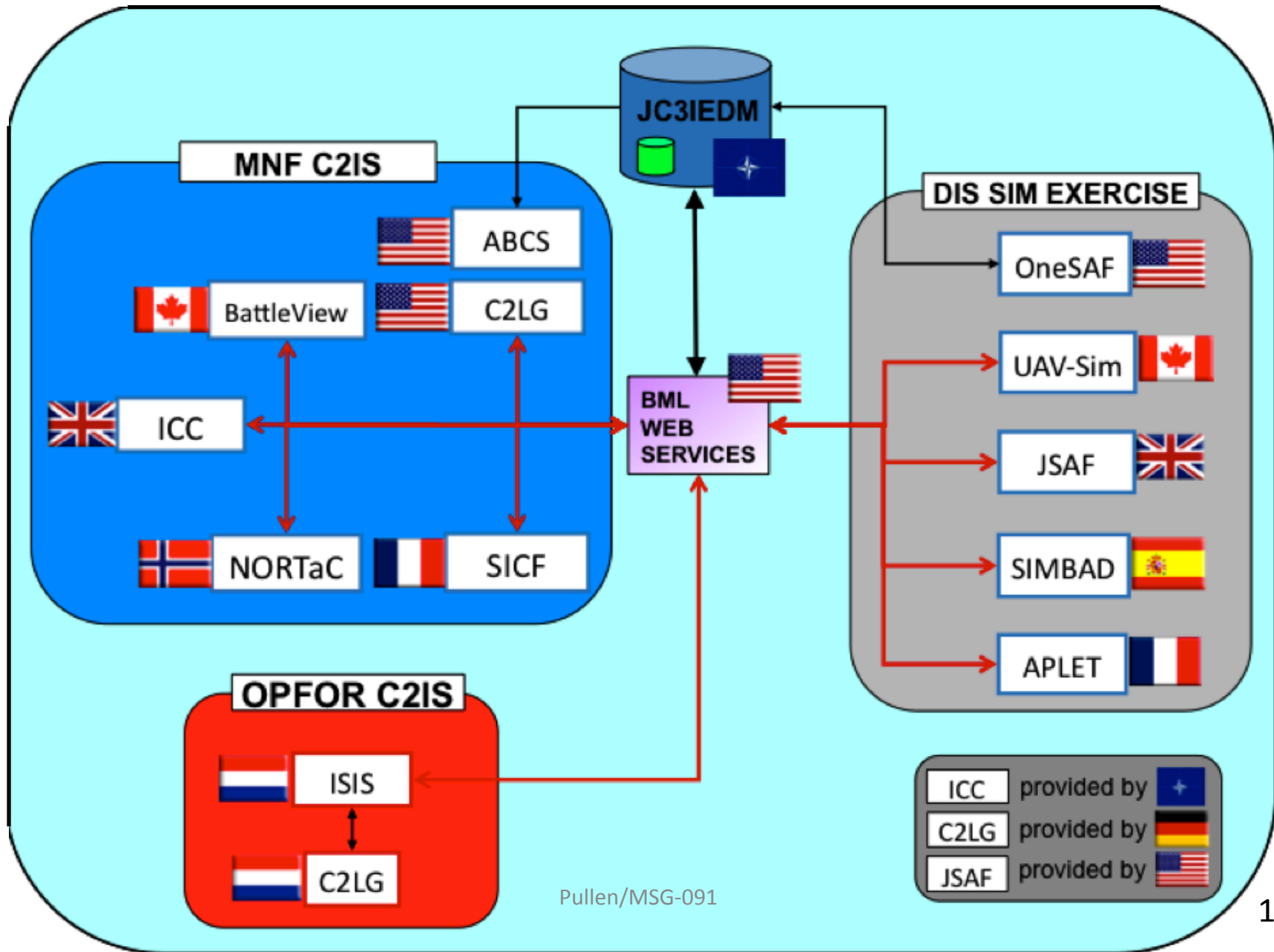
# XML Example: Report

```
<?xml version="1.0" encoding="UTF-8"?>
<BMLReport
  xmlns:jc3iedm="urn:int:nato:standard:mip:jc3iedm:3.1a:oo:2.0"
  xmlns:bml="http://netlab.gmu.edu/IBML"
  xmlns:msdl="http://netlab.gmu.edu/JBML/MSDL">
  <Report>
    <CategoryOfReport>StatusReport</CategoryOfReport>
    <TypeOfReport>GeneralStatusReport</TypeOfReport>
    <StatusReport>
      <GeneralStatusReport>
        <ReporterWho>
          <bml:UnitID>1-22</bml:UnitID>
        </ReporterWho>
        <Hostility>FR</Hostility>
        <Executer>
          <bml:Taskee>
            <bml:UnitID>1-22</bml:UnitID>
          </bml:Taskee>
        </Executer>
      </GeneralStatusReport>
    </StatusReport>
  </Report>
</BMLReport>
```

# XML Example: Report

```
<OpStatus>MOPS</OpStatus>
  <WhereLocation>
    <bml:GDC>
      <bml:Latitude>33.424079</bml:Latitude>
      <bml:Longitude>44.682716</bml:Longitude>
      <bml:ElevationAGL>2.054</bml:ElevationAGL>
    </bml:GDC>
  </WhereLocation>
  <When>20070101000000.000</When>
  <ReportID>506</ReportID>
  <Credibility>
    <bml:Source>HUMINT</bml:Source>
    <bml:Reliability>A</bml:Reliability>
    <bml:Certainty>RPTFCT</bml:Certainty>
  </Credibility>
</GeneralStatusReport>
</StatusReport>
</Report>
</BMLReport>
```

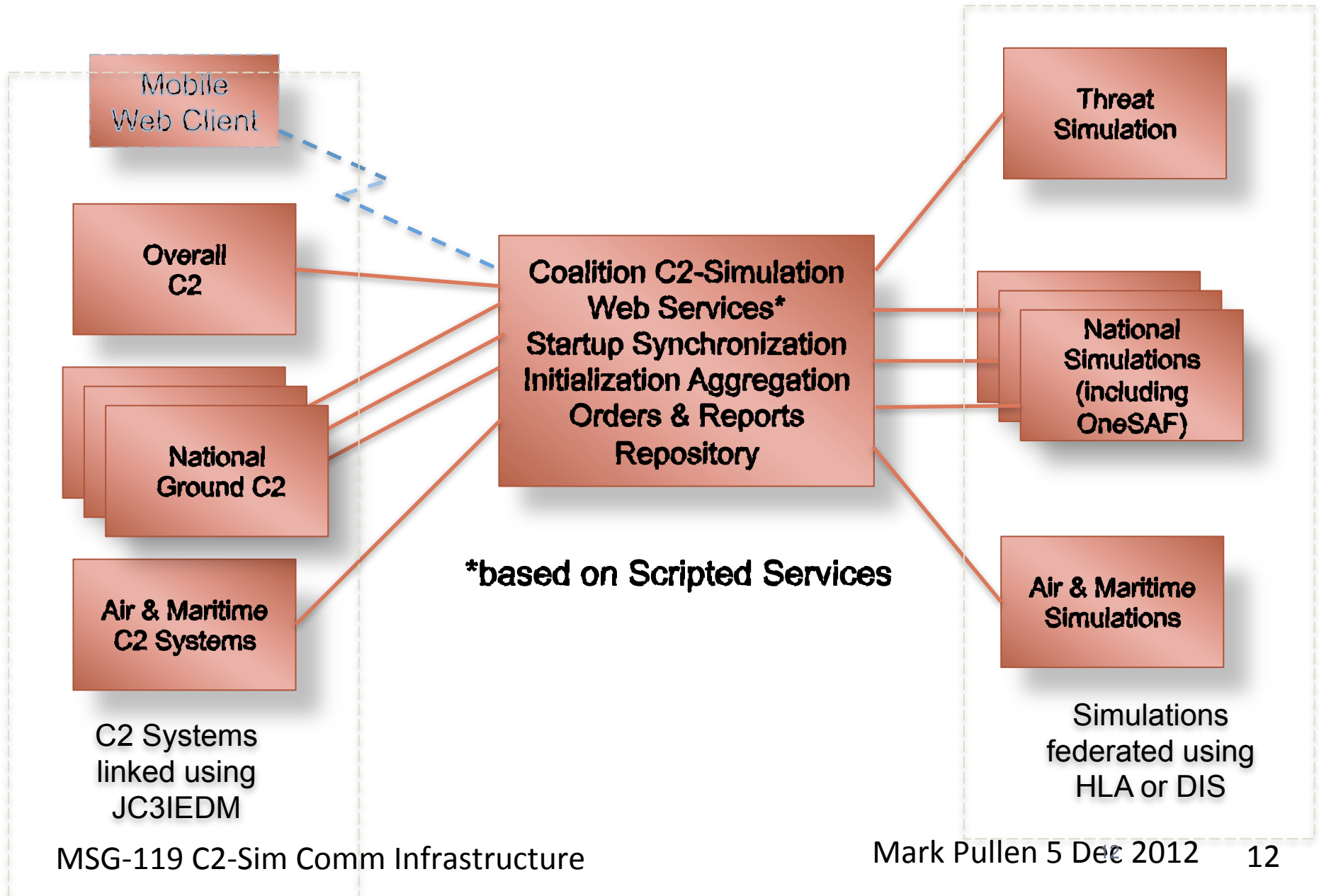
# Example: MSG-048 2009 Architecture



# Categories of C2-Sim Infrastructure

- Server
- Graphic User Interface
- Status monitoring and control

# Coalition C2-Simulation Architecture



# C-BML/MSDL Servers

# Primary Server Functions

- Accept Push/Post of XML documents and store
  - C-BML Order and Reports; MSDL scenario files
- Accept Subscriptions by Topic
  - e.g. all General Status Reports
- Publish documents to subscribers as they arrive
  - And respond to Get/Pull for them

# Additional Server Functions

- Namespaces
  - XML tagnames can be qualified by addition of a “namespace” code: <bml:Report>
  - This allows tagnames from different sources to work together safely
- Schema validation
  - Server confirms that each document received conforms to the schema
  - This identifies a likely source of incompatibilities
  - However, it slows the service
- Filtering data
  - Restrict delivery based on user-defined criteria

# Additional Server Functions

- Ontology support
  - to capture semantics (i.e. meaning) of information
  - Commonly encoded using languages such as OWL
- Common components of ontologies i(from Wikipedia):
  - Individuals: instances or objects ...
  - Classes: sets, collections, concepts ...
  - Attributes: aspects, properties, features ...
  - Relations among individuals and classes ...
  - Function terms: complex structures formed from certain
  - Restrictions: formally stated descriptions of what must be
  - Rules: statements in the form of an if-then ...
  - Axioms: assertions (including rules) ...
  - Events: the changing of attributes or relations

# Additional Server Functions

- Scripting
  - Server's function is described by scripts that are sequences made up of simple directions the server is to follow as it processes documents
  - This allows rapid reconfiguration of the server in the experimental environment
  - But it generally results in lower performance than a server where the processing is written in a programming language such as Java or C++
- Logging/replay
  - Server writes a file showing every transaction it receives, with time stamps
  - Server is capable of replaying this file to recreate the original sequence of Orders and Reports at original time intervals

# Additional Server Functions

- Bridged servers
  - Multiple servers can be tied together to increase load capacity and geographic scope of the C2-Sim coalition
- Translation among multiple schema
  - Server parses the XML document according to its appropriate schema; produces output conforming to a different designated schema
  - Possible only where the data contained support the same semantics

# Additional Server Functions

- Multithreading for performance
  - Server throughput can be improved by processing multiple messages in parallel
  - Requires a multithreaded design
- Dynamic Topics for publish/subscribe
  - The ability to configure publish/subscribe Topics at runtime enables adaptive data delivery
- Multiple web service interfaces
  - Traditional Web service protocol is SOAP
  - RESTful interface avoids RPC overhead where not needed
  - Server could support either or both
  - Client must match server (RESTful is better for C++)

# Additional Server Functions

- Aggregating MSDL inputs
  - In a coalition each C2 and simulation system can have different initialization requirements
  - A consolidated MSDL file is needed for consistency
  - Server can aggregate them automatically
  - A change on any system is reflected to all
- Openness of source code
  - Fully Open Source development allows products to be re-used by anyone, with no license charge
  - Government Open Source is allows source release only to parties designated by sponsoring agency
  - Open Technology Development aims at government-wide release

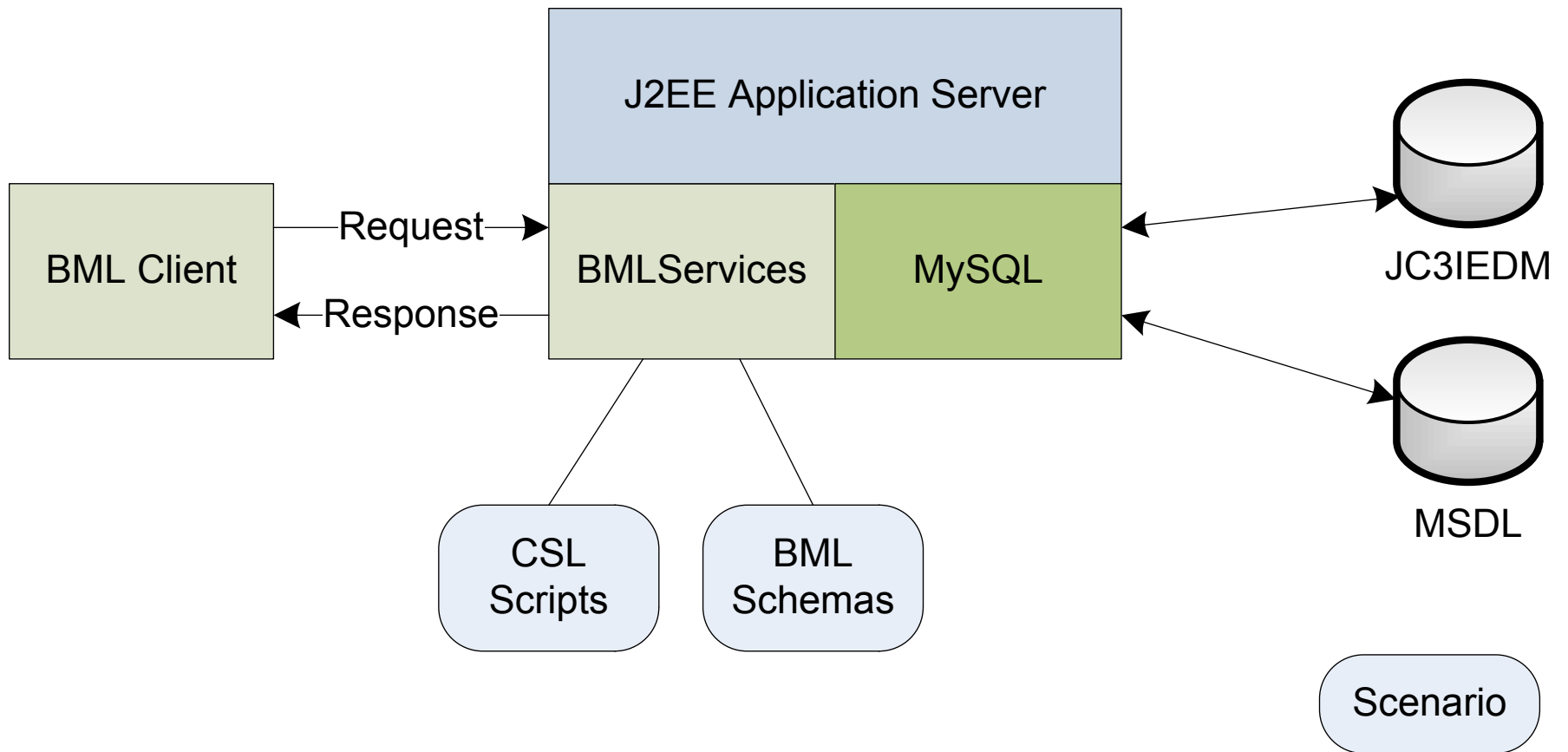
# Servers in Use Today

- Scripted BML Server  
(SBML)
- Coalition Battle Management Services  
(CBMS)
- FKIE server
- Saab WISE/SBML

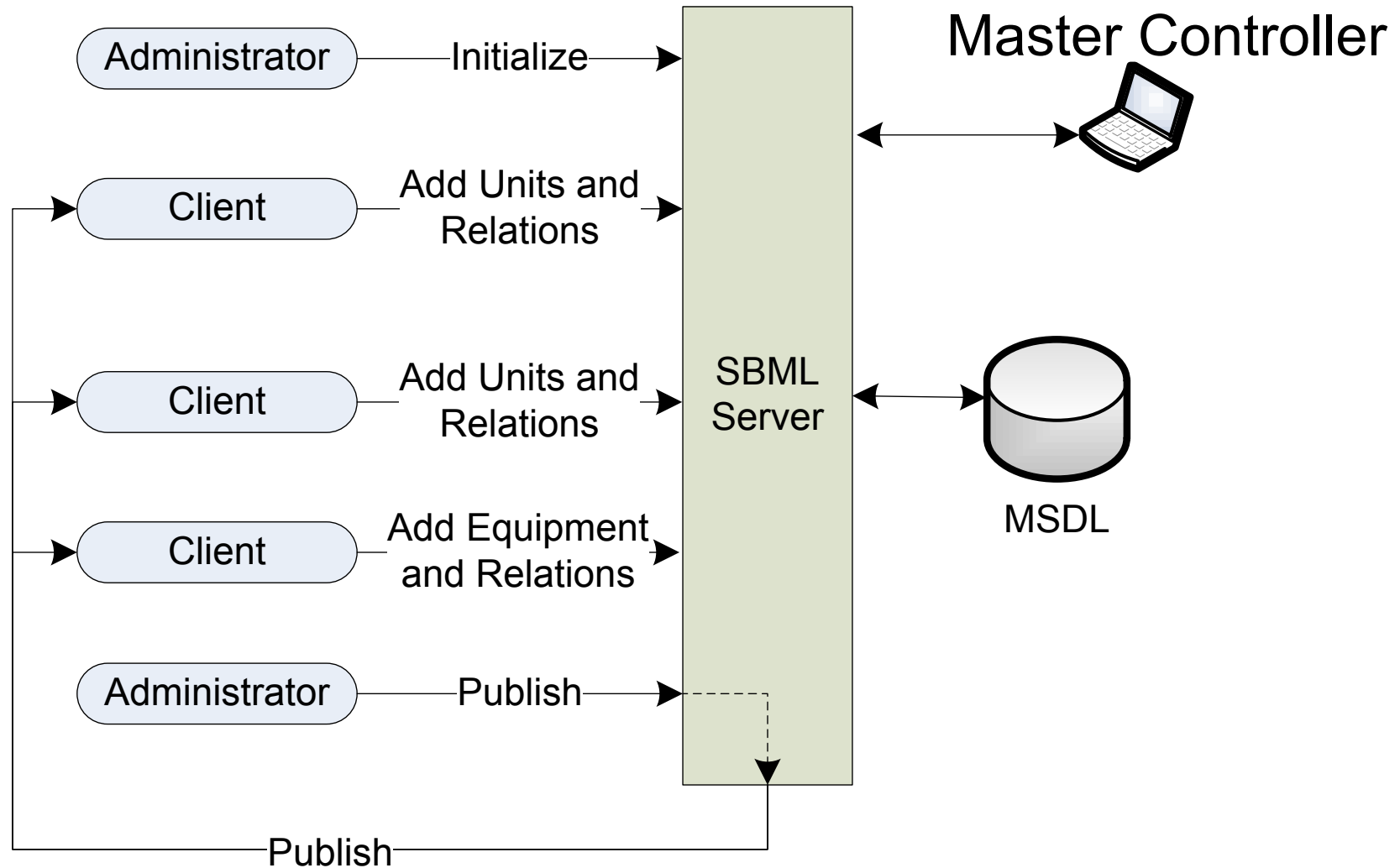
# Scripted BML (SBML) Server

- Developed by GMU C4I Center to support coalition activities such as MSG-048
  - Open source
  - Offered reference implementation for SISO C-BML
- Scripted server concept
  - New constructs can be implemented rapidly
  - Changes to BML and JC3IEDM applied quickly
  - Reduced cost for experimentation prototyping
  - Script provides a concise definition of BML to data model mappings
- Lower throughput (10 transactions/second)

# SBML Architecture



# SBML/MSDL Server Operation



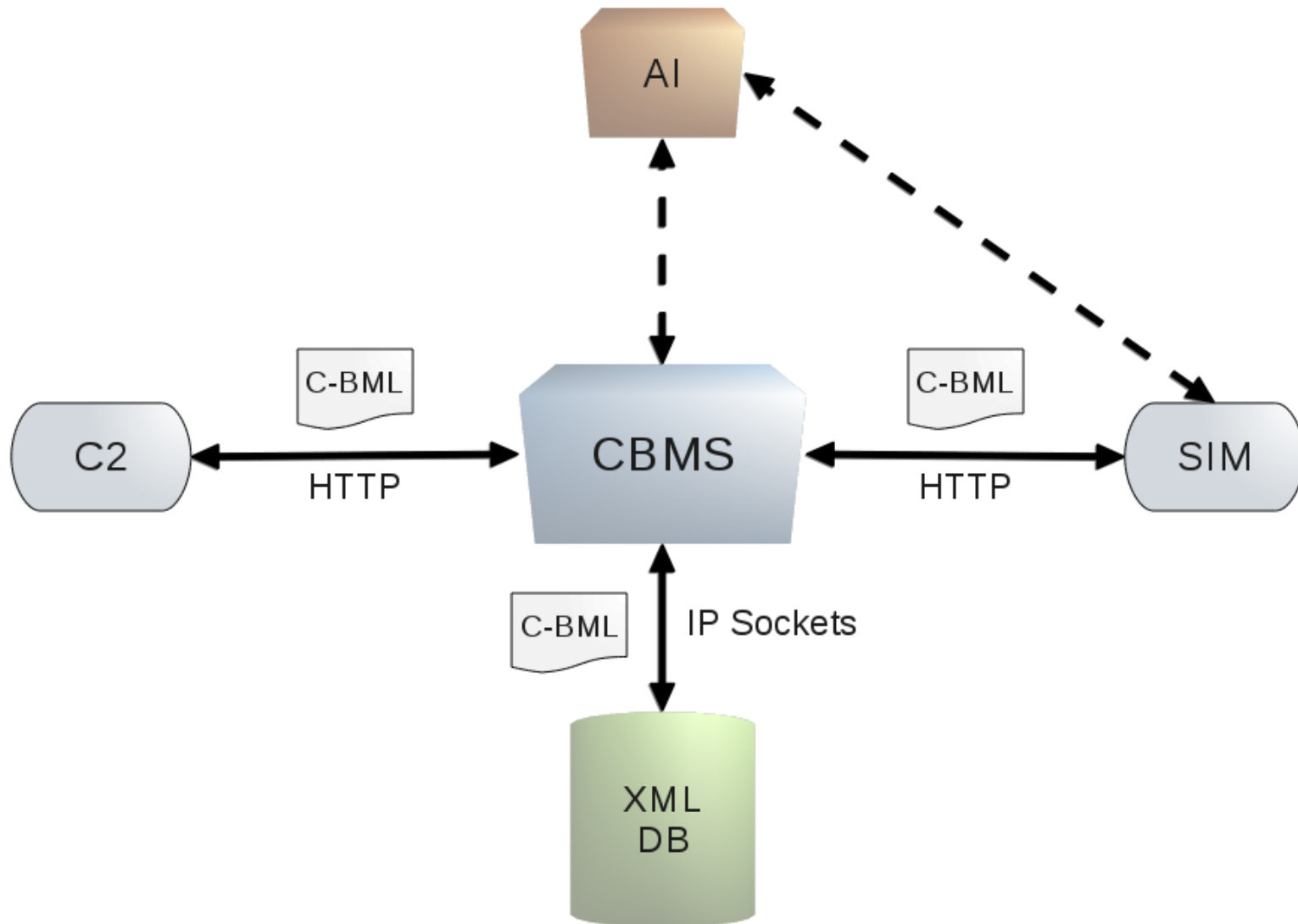
# SBML Additional Capabilities

- Namespaces
- Schema validation
- Filtering data for subscription topics
- Scripting
- Logging/replay
- Bridging
- Translation
- Multithreading
- Dynamic topics
- SOAP and REST
- Aggregating and serving MSDL
- Auxiliary document mode
- Fully open source

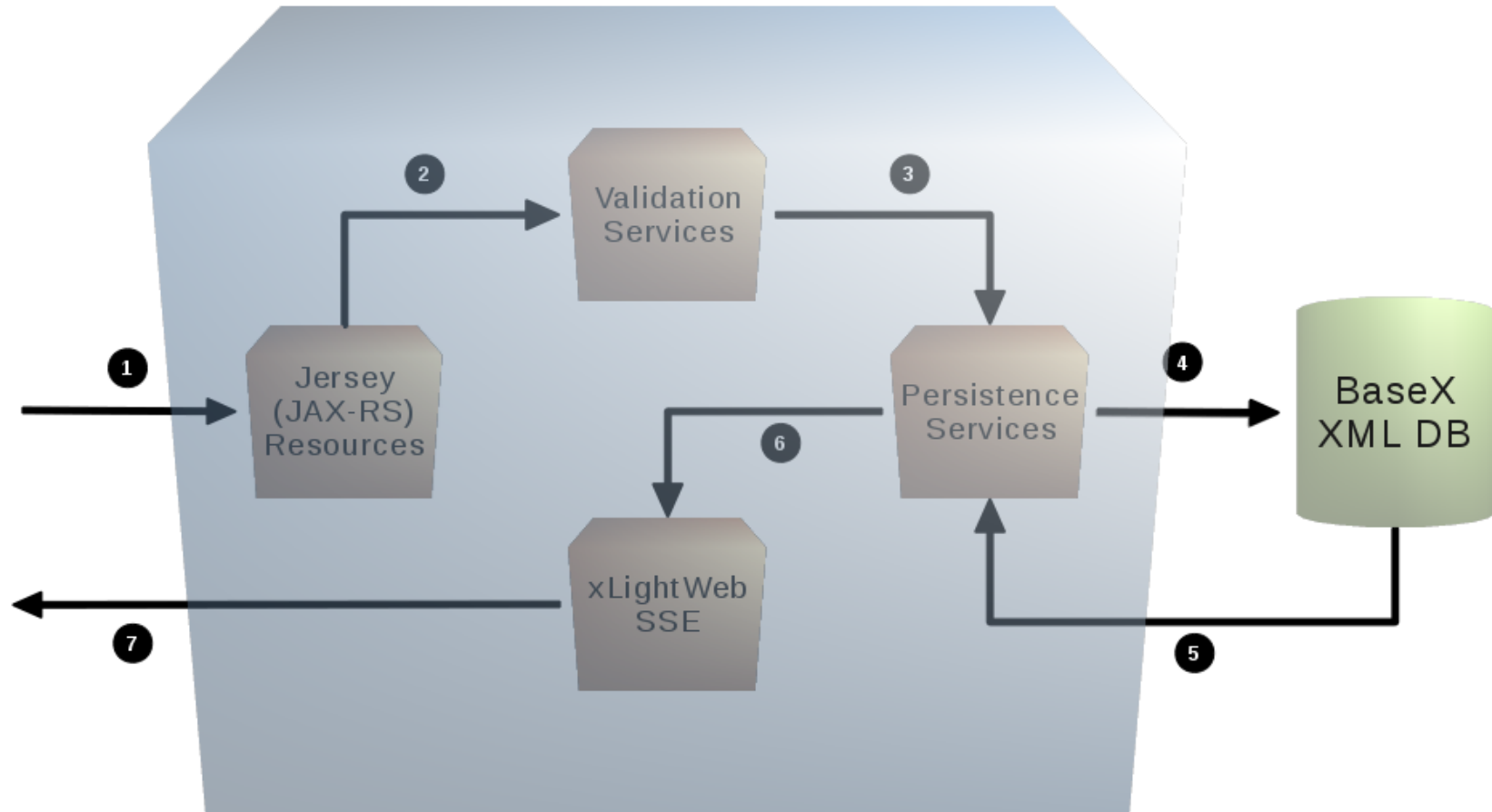
# Coalition BML Services (CBMS)

- Developed by VMASC for JCW
- Serves complete XML documents
  - Does not parse them unless schema validation is requested
  - Based on Apache open source components
  - This allows very high throughput:
    - Over 100 transactions/second
- Government Open Source
  - Currently available under Distribution Agreement to USA, AUS, CAN, NZ, UK

# Coalition Battle Management Services (CBMS) External Component Integration



# Coalition Battle Management Services (CBMS) Subscription Service Internal Flow



# CBMS Additional Capabilities

- Namespaces
- Semantic validation using ontology
- Schema validation
- Filtering data for general queries and subscription Topics
- Logging
- SOAP and REST
- Serving MSDDL
- Government open source (Open Technology Development)

# FKIE BML Server

- Developed by Fraunhofer FKIE, based on SBML Open Source
  - To support Germany-France BML experiments
- Implements a schema that grew out of MSG-048 work
  - Not SISO standard
- Executable only available to Germany, France, Denmark, Netherlands, Spain

# FKIE BML Server Additional Capabilities

- Namespace
- Filtering for distribution by Topic
- Logging
- SOAP and RESTful interfaces

## Saab WISE/SBML

- Widely Integrated Systems Environment (WISE) is Saab off-the shelf proprietary software for integration heterogeneous systems
  - BML server provided initially as transition of GMU open source SBML
  - Saab is funding GMU to inform further development
- Saab plans to increase level of C-BML/MSDL integration in WISE
  - Product extension and support by Saab
  - Comprehensive, high-performance service suite
- Lite version available free for development
  - including all MSG-085 activity

# WISE/SBML Capabilities

- Namespaces
- Schema validation
- Filtering data for subscription topics
- Scripting
- Logging/replay
- Bridging
- Translation
- Multithreading
- Dynamic topics
- SOAP and REST
- Aggregating and serving MSDL
- Auxiliary document mode

# Other Supporting Software For C-BML/MSDL Communications

# Graphical User Interfaces (GUIs)

- During development it is very useful to have a generic way to generate and inspect BML documents
  - Create an Order, Report, Request etc. and introduce it to server
  - Accept and display Order, Report, etc. from client system or server
  - Edit either of the above
- The GUI can serve as a limited/surrogate C2 system for experimentation

# BML GUI Functions

- Editing a C-BML or MSDL document
- Merging MSDL documents
- Serialization of document
- Grammar validation of document
- Schema validation of document
- Auto-configuration to schema
- Pulling a document
- Pushing a document
- Subscription to server Topics
- Retrieving latest reports
- C2 capability
- Displaying maps with standard overlays
- Geolocation entry to documents from maps

# BML GUIs in Use Today

- C2LG GUI
- BML C2 GUI

# C2LG GUI

- Command & Control Lexical Grammar (C2LG) Graphical User Interface (GUI) constructed by the German research center FGAN
  - now Fraunhofer FKIE
- Originally created to generate “pure” BML statements that were valid grammar statements.
- In many BML activities, the C2LG GUI has been used as an “integration hub” to take the input from C2 systems and construct a “valid” JBML Order that could be sent and ingested by different nation’s simulations.
- Also used as surrogate C2 system for experimentation
- Available as Government Open Source to MSG-085 participants

**Groundtask**

Create Order, Request or Commission

Order  
 Request  
 Commission

Regarding

Action

Choose an action verb

Units

Choose taskee

Tasker: FRA\_DEU\_LU

Taskee:

Info

Task Control Features

**Ground Order**

Header

Common information

Sender: FRA\_DEU\_LU

Addressee:

OrderID: C2LG\_1289897274825

Time: 20101116094754.825

Security:

Ground Tasks

#	BML
1.	m...
2.	re...
3.	fo...
4.	at...

Optionen

Send    Laden    Speichern

IBML File Import    IBML File Export

**MapView**

1:301.537

Di, 16 Nov 2010 09:50:22 MEZ

MapView

Console

Std Console

```

pf-1
Line: S->0.6455193:1.1681396 E->0.6440625:1.1683503
seize-1
Line: S->0.64511675:1.1662419 E->0.6457876:1.1659927
cover-1
Line: S->0.6471293:1.1675837 E->0.64582586:1.1681012

```

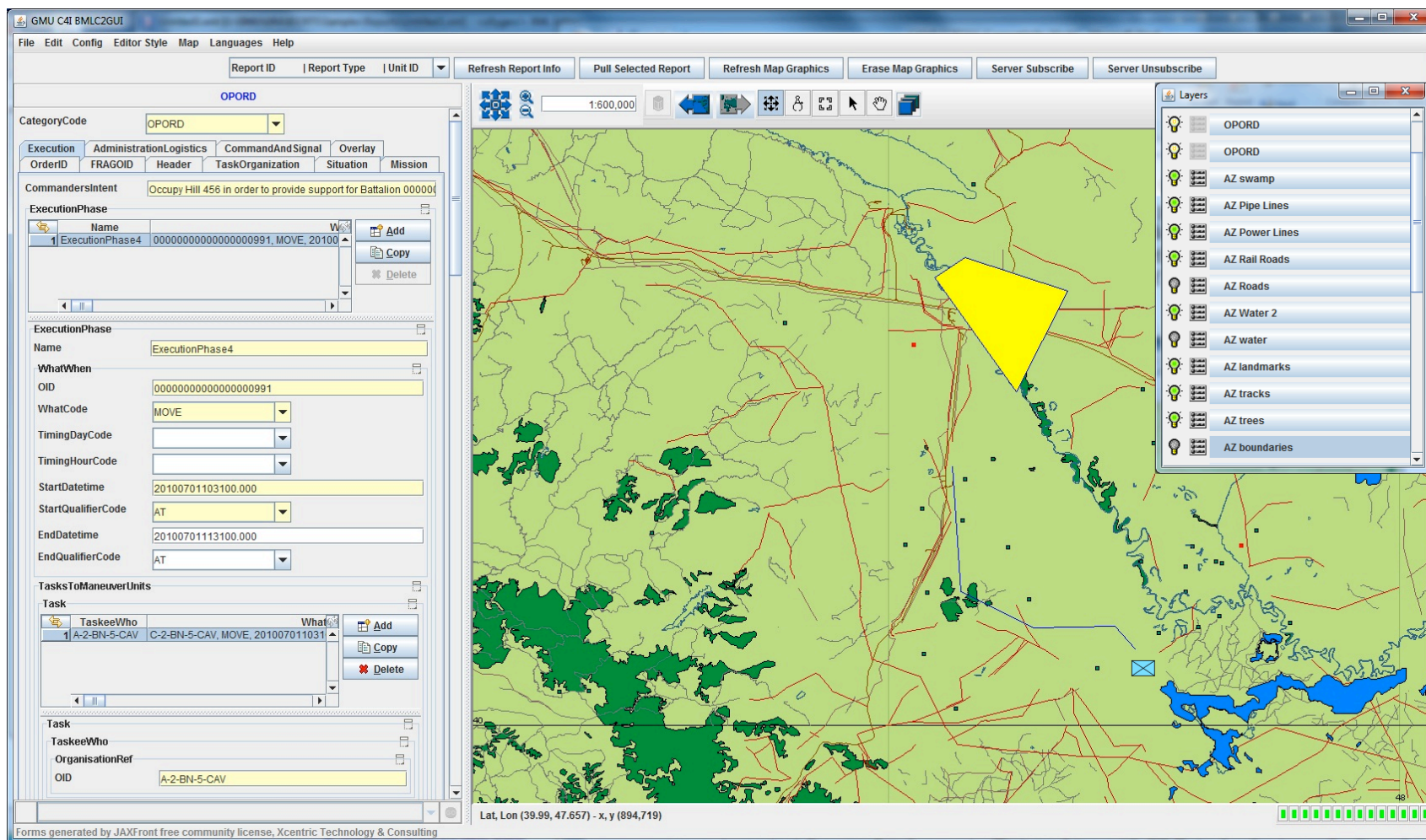
# C2LG GUI Functions

- Editing a C-BML or MSDL document
- Merging MSDL documents
- Serialization of document
- Grammar validation of document
- Schema validation of document
- Pulling a document
- Pushing a document
- Subscription to server Topics
- Retrieving latest reports
- C2 capability
- Displaying maps with standard overlays
- Geolocation entry to documents from maps

# BML C2 GUI

- Patterned after C2LG GUI
  - With advice from Fraunhofer FKIE
- Usable as editor or monitor
- Auto-configures to BML schema
- Available open source to anyone

# BML C2 GUI



# BML C2 GUI Functions

- Usable as editor or monitor
- Reads/writes Orders and Reports
  - Subscribes to Topics to receive them
- Auto-configures to any BML schema
- View and modify a BML-XML file
  - C-BML, MSDL, or other
- Map/image display shows 2525B icons from XML
- Graphically enter geolocation data in BML-XML file
- Open source at <http://c4i.gmu.edu/BML>

# Status Monitoring and Control

- Lesson learned in MSG-048:
  - It is impractical to coordinate multiple interoperating C2 and simulation systems with only spoken communication
- Solution: a coordinating webpage
  - Shows possible states of each coalition system
  - Master Controller provides coordinating guidance
  - Inputs can come from webpage interface
  - Or Web service client

# MSG-085 Status Monitor

## MASTER CONTROLLER STATUS

Scenario: scenario1

Current Order: initialize

Scenario MSDL status: not started

Number of observers: 4

Comment: new comment

## CLIENT STATUS

Client: C2IS1 - rpt1

Current Status: stopped

Change status:

Add/change comment:

Client log out

	STOPPED	INITIALIZING	READY	RUNNING	PAUSED	COMMENT
C2IS1 rpt1	stopped					
C2IS2 rpt2	stopped					
C2IS3 rpt3	stopped					stuck
C2IS4 rpt4		setting up				how are you
C2IS5 rpt5	offline					
C2IS6 rpt6	offline					
C2IS7 rpt7	stopped					let's go

# MSG-085 Status Monitor

## MASTER CONTROLLER STATUS

Scenario: scenario1

Current Order: **stop**

Scenario MSDL status: not started

Number of observers: 4

Change current order:  stop  initialize  run  pause

Add/change comment:

Log out client/observers:

Publish MSDL

Reset All

Master log out

## CURRENT CLIENT STATUS

	STOPPED	INITIALIZING	READY	RUNNING	PAUSED	COMMENT
C2IS1 rpt1	offline					
C2IS2 rpt2	<b>stopped</b>					
C2IS3 rpt3	<b>stopped</b>					stuck
C2IS4 rpt4		setting up				how are you
C2IS5 rpt5	offline					
C2IS6 rpt6	offline					
C2IS7 rpt7	<b>stopped</b>					let's go

# Conclusions

- Infrastructure is essential for C2-Sim progress
- Much work has resulted in useful supporting software for experimentation
- Development can be expected to continue for several years
  - Facilitating new facets of experimentation
- Capabilities are beginning to emerge in off-the-shelf / operational software