Services to Support Experimentation for Operational Use of Simulations in Coalition Command and Control

Dr. Mark Pullen, Douglas Corner, Lisa Nicklas and Priscilla McAndrews

Presenter: Douglas Corner

George Mason University C^{4}I Center

{mpullen,dcorner,lnicklas,pmcandre}@c4i.gmu.edu

Presentation Overview

- MSDL/C-BML background
- Experimentation in NATO MSG-048
- Architecture and functions of SBMLServer
- Converging MSDL with C-BML
- Expanding SBMLServer to support MSDL
- Role of Status Monitor and Control
- Early application of new capability
- Conclusions

NOTE: This is about a capability, not an experiment.

MSDL/C-BML Background

Vision

- We envision a day when the members of a coalition interconnect their networks, C2 systems, and simulations simply by turning everything on and authenticating, in a standards-based environment.
- A major step forward in C2 for coalition agility.

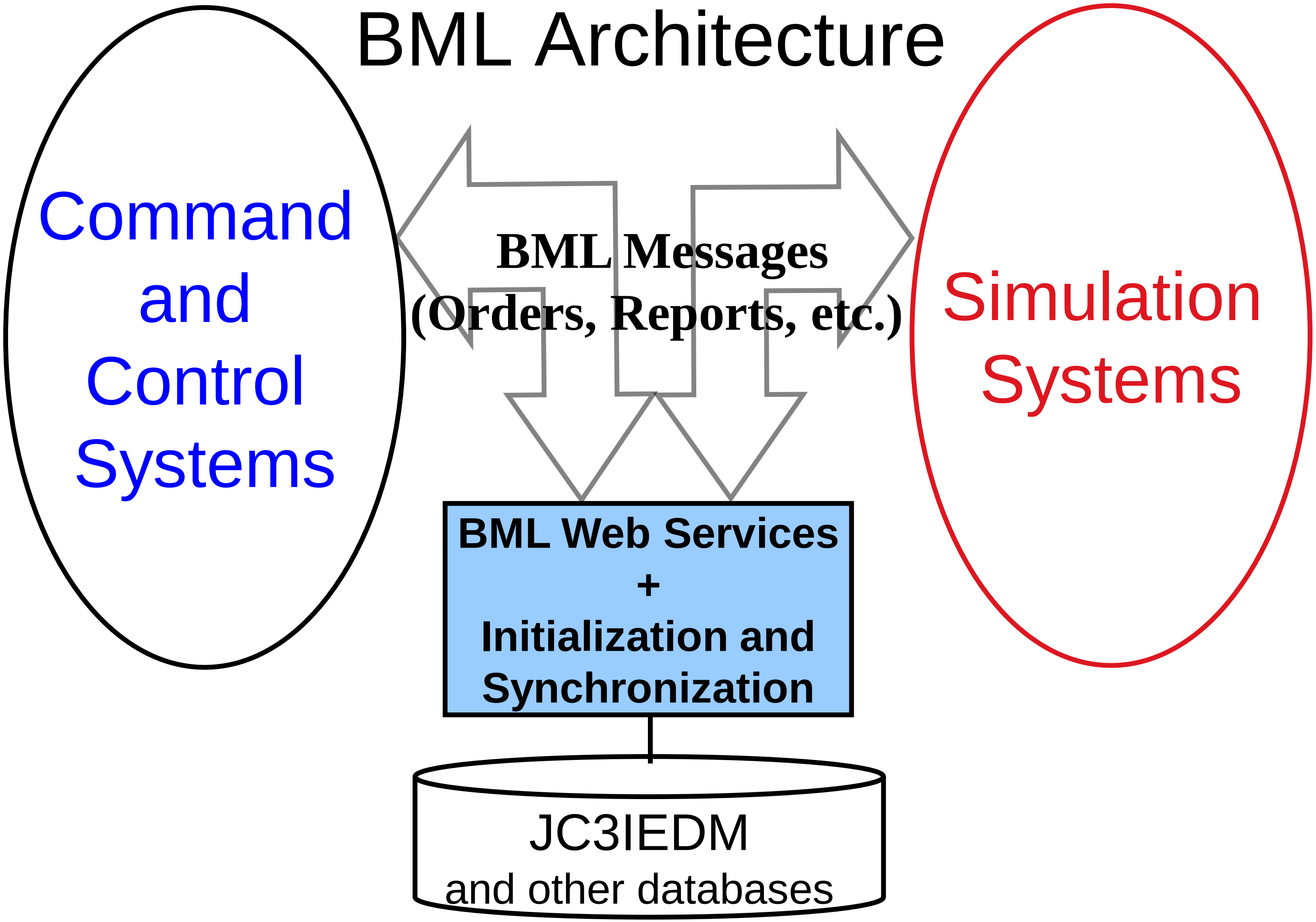
BML Purpose and Operation

- Facilitates C2-Simulation interoperation
  - Exchange of Orders and reports in standard format
- Current architecture uses a repository service to hold state submitted by client C2 and Simulation systems
  - Web service with XML input – Network Centric
  - Data stored in JC3IEDM and can be replicated
- Now using SISO draft Coalition BML (C-BML) standard

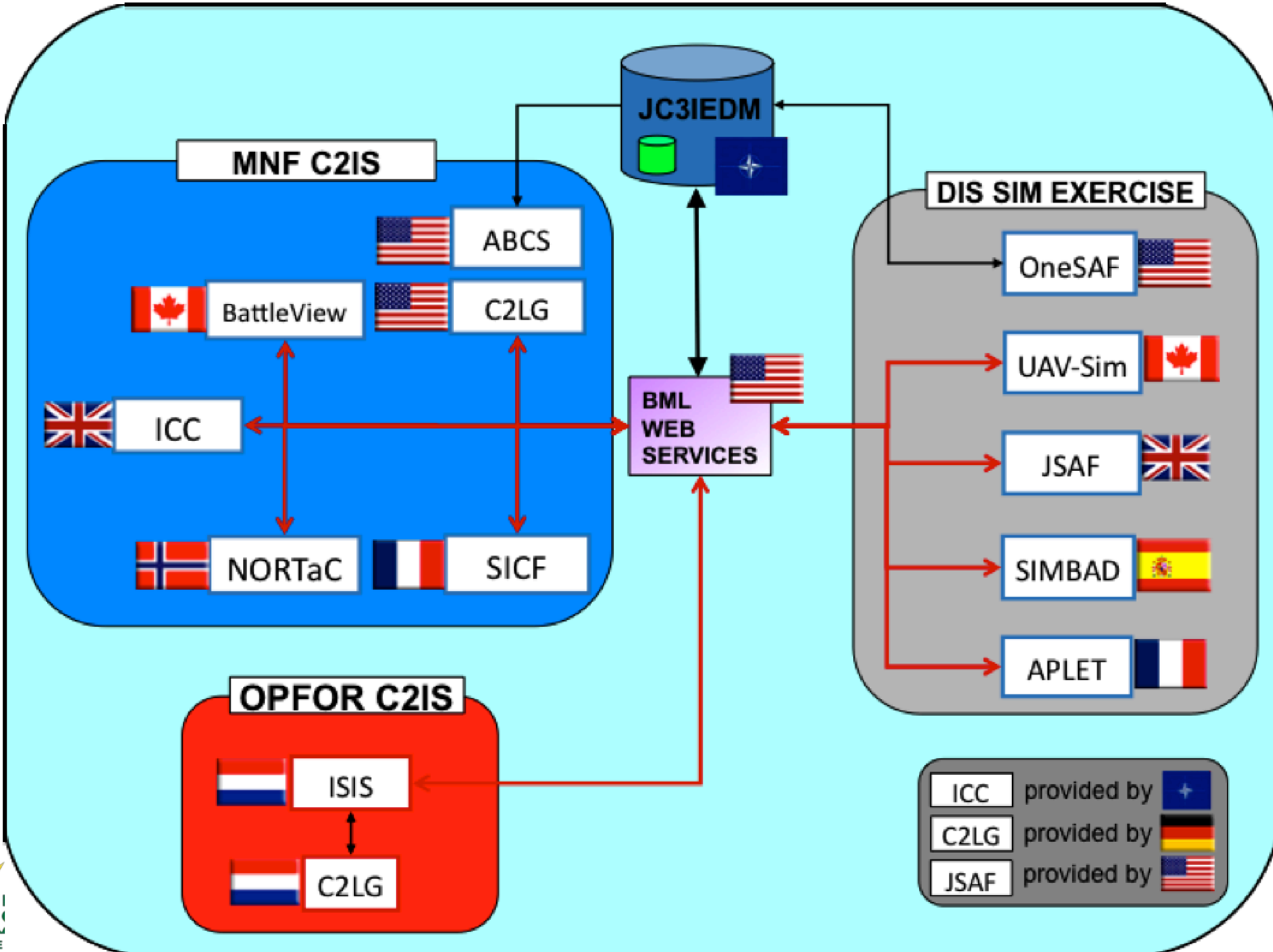
Practical Requirements for BML Operation

- Coordinated initialization
  - All participating systems must share common initialization data
  - This requires assembling and publishing data from the coalition C2 and simulation systems
- Mechanism to display and synchronize operations with the system-of-systems
  - All operators need to know status of all systems
  - Master Controller starts and stops the simulations
  - Preferred interface: automatic via Web service

Evolving
BML Architecture



MSG-048 2009 Architecture



SBMLServer Architecture and Functions

Scripted BML Web Services

- Middleware functions don’t change
  - Mapping BML to JC3IEDM and push/pull to database
  - Program these once and get them right
- Interpreted WS offers flexibility
  - Rapid implementation of new BML constructs
  - Easy to modify underlying data model
    - MIP standard also continues to change
  - Reduces time and cost for prototyping
  - Scripting language provides a concise definition of BML-to-data model mappings
  - Although bugs still happen, the number of possible mistakes is far smaller
- Scripted operation may, however, be slower
  - Multithreading helps this
  - But a hard-coded implementation is likely to perform better

The Old Way: IDEF1x Mapping Definition

JBML mapping to JC3IEDM

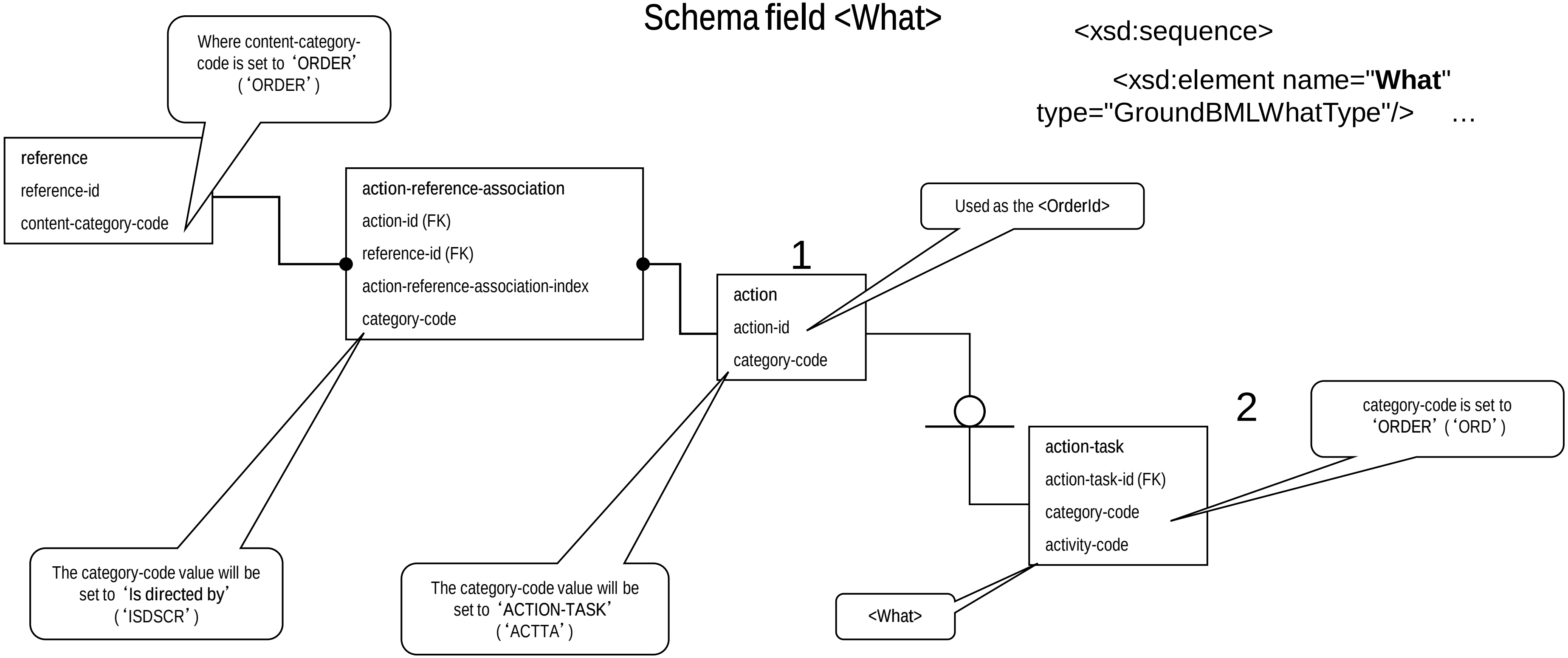
Schema field <What>

Schema Reference:

<xsd:complexType name="CommandType">

<xsd:sequence>

<xsd:element name="**What**" type="GroundBMLWhatType"/> …



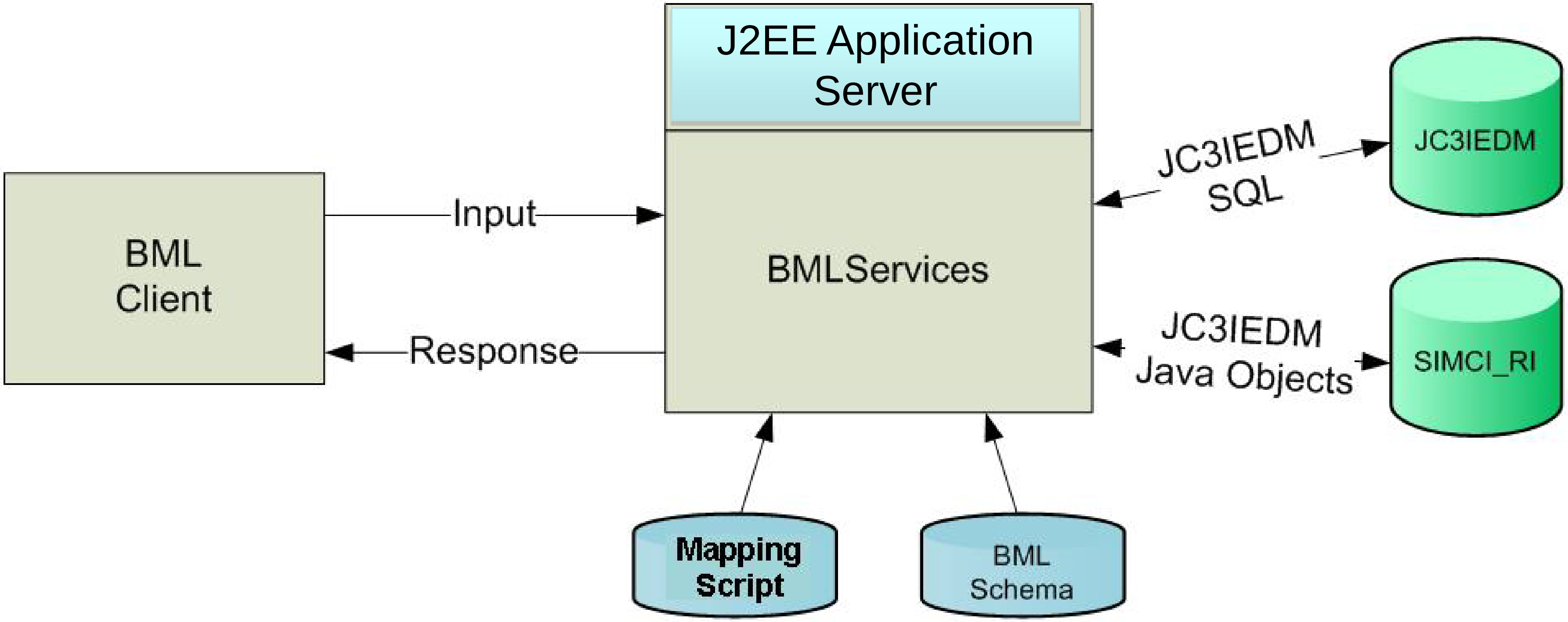
Not machine readable though highly structured

Script is a concise XML coding of this

Scripted BML WS Design

- Basic operations: *push* and *pull*
  - Currently, servers for SQL and RI databases
  - Scripts implement BML Orders and Reports
- Script defines implementation of Business Objects (constituents of the higher-level BML grammar) over the JC3IEDM data model
  - BO is an XML subtree rooted at a defined node in the XML file – can invoke other BO
- Interpreter uses two files plus WS input
  - Mapping file contains script
  - BML schema file provides necessary context
  - XML namespace capable

Scripted BML WS Configuration

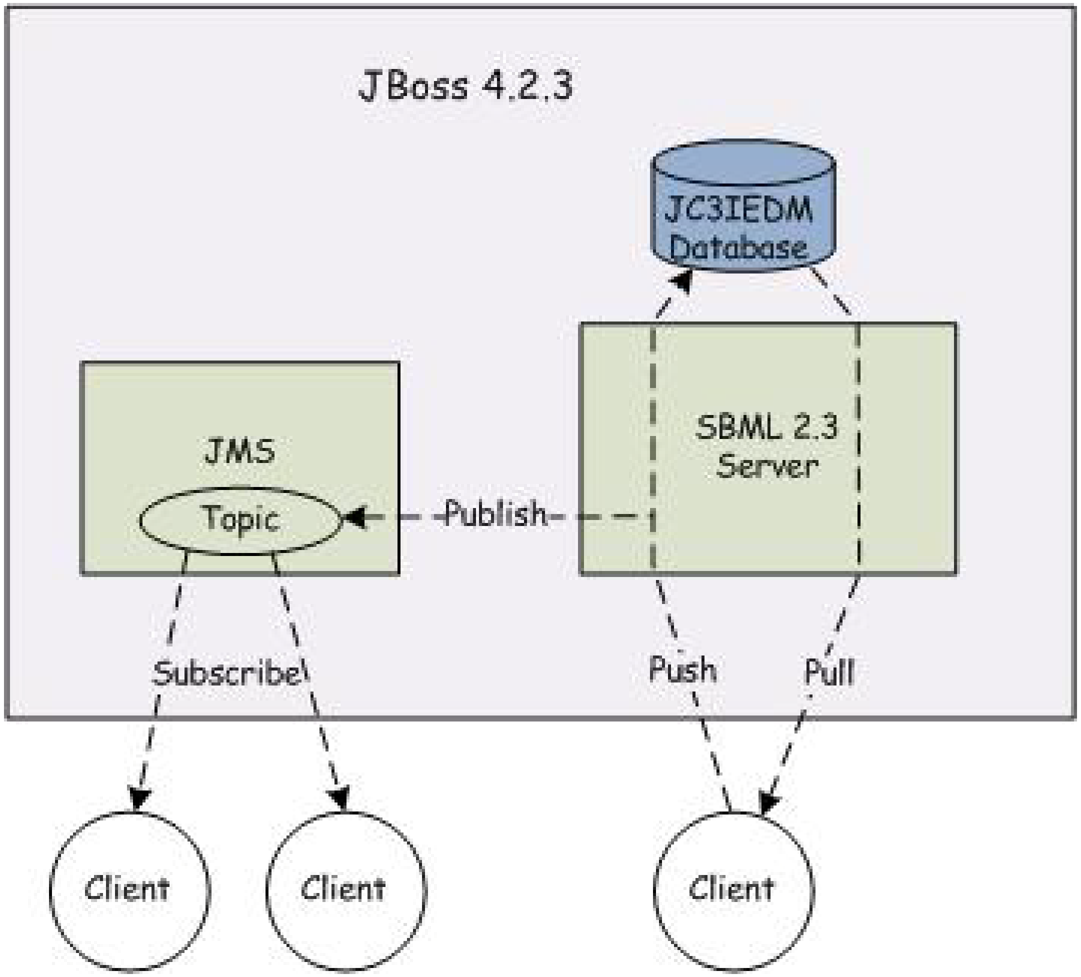


Two implementations: MySQL and SIMCI RI

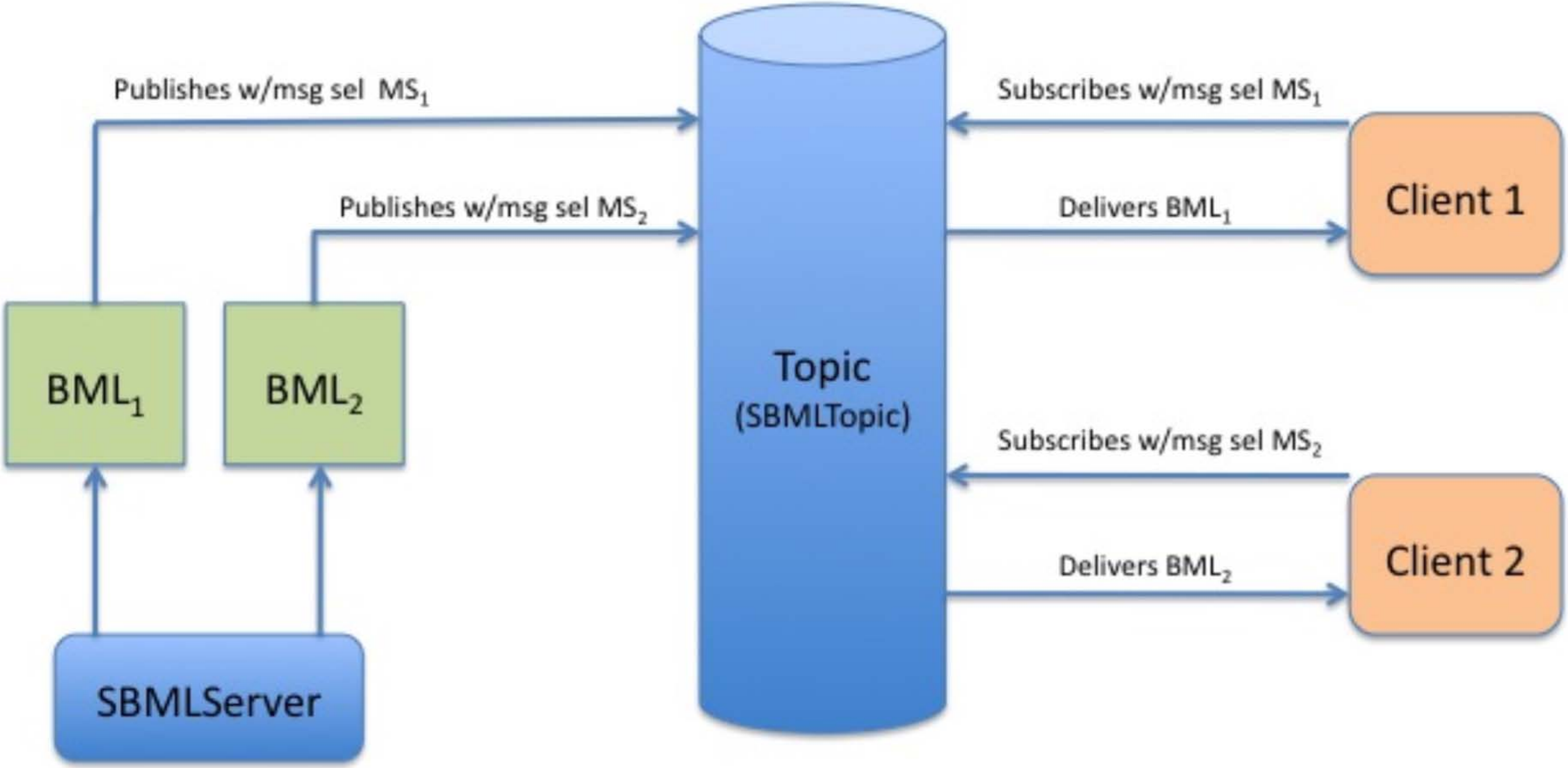
Publish/Subscribe in SBMLServer

- Using Publish/subscribe, clients identify the categories of information they need – they *subscribe* to *Topics*
- Server sends them a copy of every update associated with each subscribe Topic
- More timely updates and a dramatic reduction in overhead
- Avoids inefficiencies:
  - Server must re-read information written to database
  - Redundant polling
  - Separate server cycle needed for each client
- SBMLServer implements Dynamic Topics
  - Can be changed during operation

Publish/Subscribe Architecture



Message Selectors for Dynamic Topics



MSDL and C-BML

MSDL Enables Effective Initialization

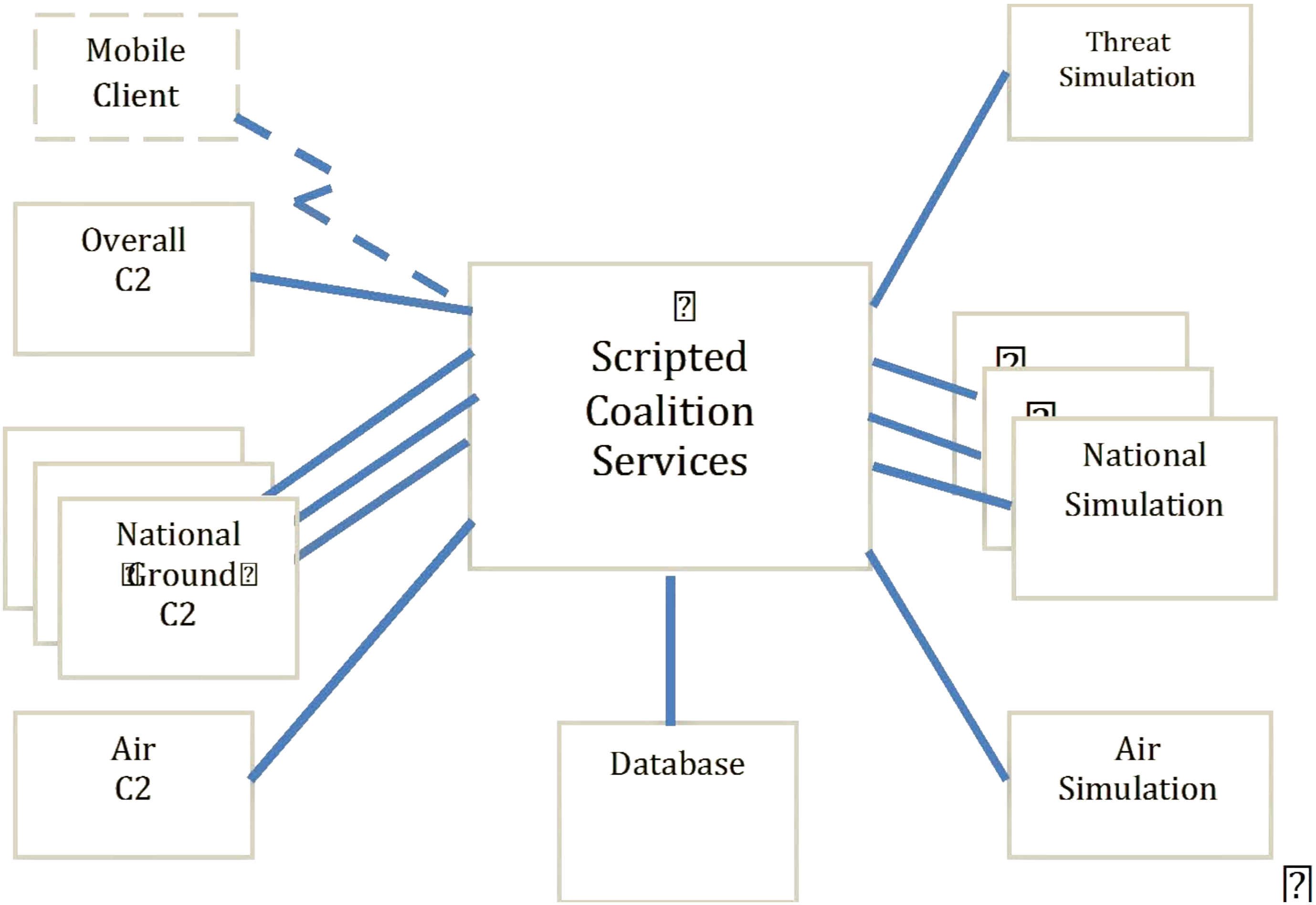
- MSDL is a SISO standard that grew out of US Army OneSAF program
- Intended to reduce development time and cost by re-use of scenarios
- Information to initialize:
  - Task Organization Definition
    - Name and type of unit, parent and child relations
  - Tasking Definition
    - Who/What/When/Where/Why
  - Tactical graphics
    - US MIL STD 2525C / NATO APP-6C

Converging MSDL and C-BML

- MSDL has well-developed Task Organization
  - C-BML has only a placeholder
  - Conclusion: use MSDL TaskOrg format
- C-BML has well-developed Tasking Definition
  - MSDL has only a placeholder
  - Conclusion: use C-BML Tasking
- Standards for tactical graphics well established
  - Conclusion: use MIL STD 2525C / NATO APP-6C
  - NOTE: May not be completely consistent with JC3IEDM

Expanding SBMLServer to Support MSDL

Scripted Coalition Services



expanding SBMLServer to support initialization and synchronization

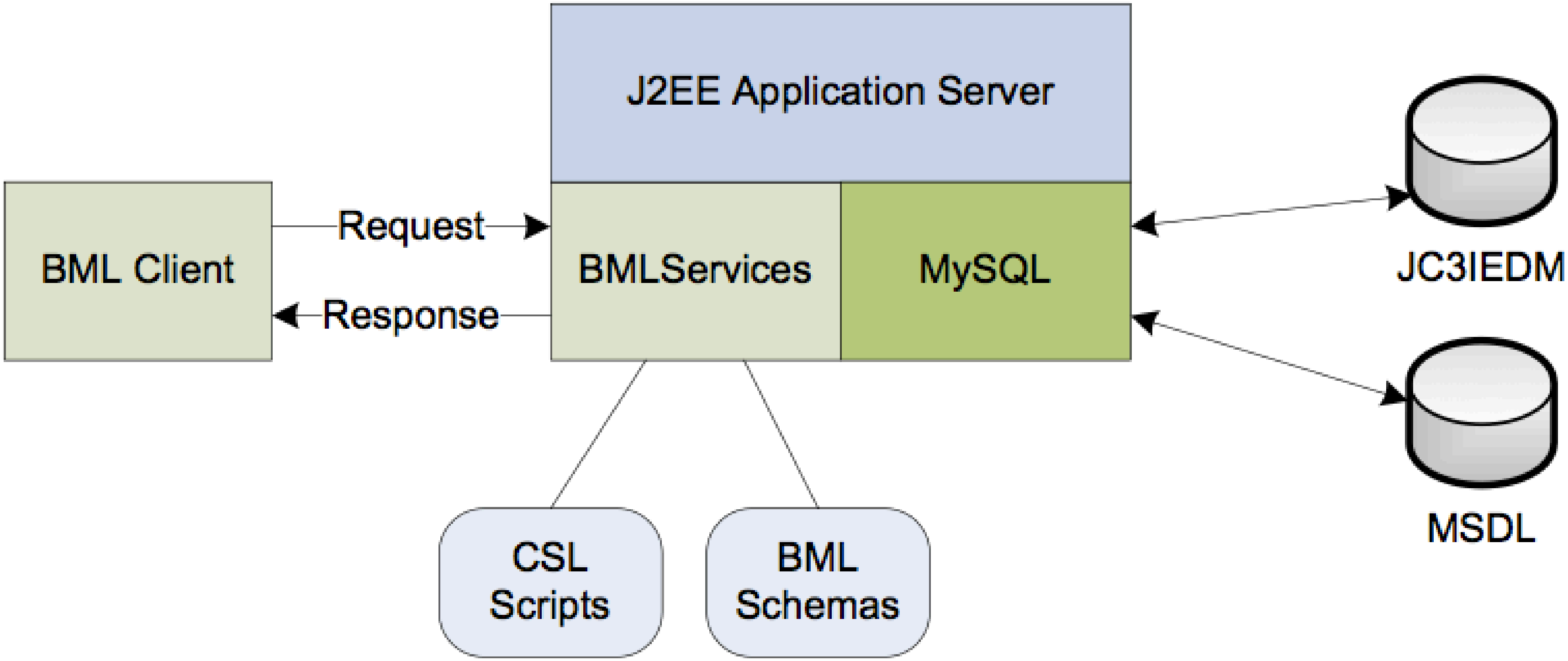
MSDL Scenario File Contents

- Force/Sides
- Units
- Equipment
- Installations
- Overlays
- Graphics

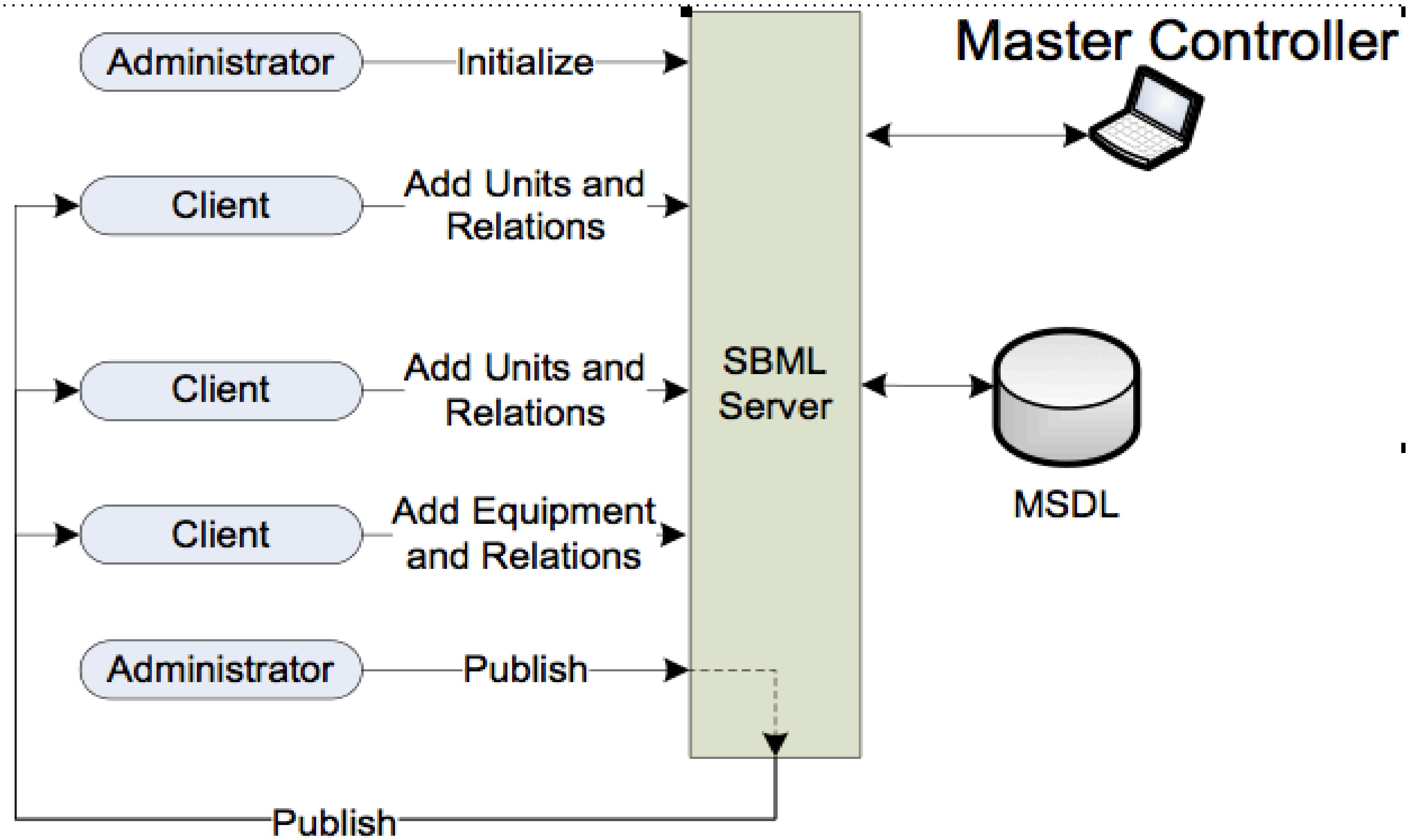
SBMLServer Support for MSDL

- Coalition C2 and simulation systems derive from different nations
  - And reflect different doctrines
- Ideally, each nation will employ its own C2 and simulation with its own forces
  - All interoperation through BML
- Each system contributes to the overall initialization MSDL file
  - SBML accepts individual files and aggregates them
- On command of Master Controller, the aggregated file is published to all systems

SBMLServer Architecture with MSDL



Aggregating MSDL Scenario Files



Status Monitor and Control

Why Status Monitor and Control?

- Operators of C2 and Simulation systems maybe distributed
  - Unable to coordinate by voice
- Even when co-located, in NATO MSG-048 six national C2 systems and five national simulations were difficult or impossible to coordinate
- Our solution to this is a webpage to display status
  - With inputs from each participating system and from Master Controller

How Status Monitor and Control Works

- Each system, and the Master Controller, has a separate input webpage
  - Input by dropdown containing all possible expected states of that system
  - Plus comment field for unexpected situations
- Master Controller inputs go-to-states
- Systems input state to which they have gone
- Current status displays to all pages
  - Color-coded for easy detection
- Automated interface possible through Web service

MSG-085 Status Monitor

MASTER CONTROLLER STATUS

Scenario: scenario1 **Current Order:** off-line

Scenario MSDL status: not started **Number of observers:** 2

CLIENT STATUS

C2IS1-rpt1 Current Status: stopped

Change status:

Add/change comment: power down

Client log out

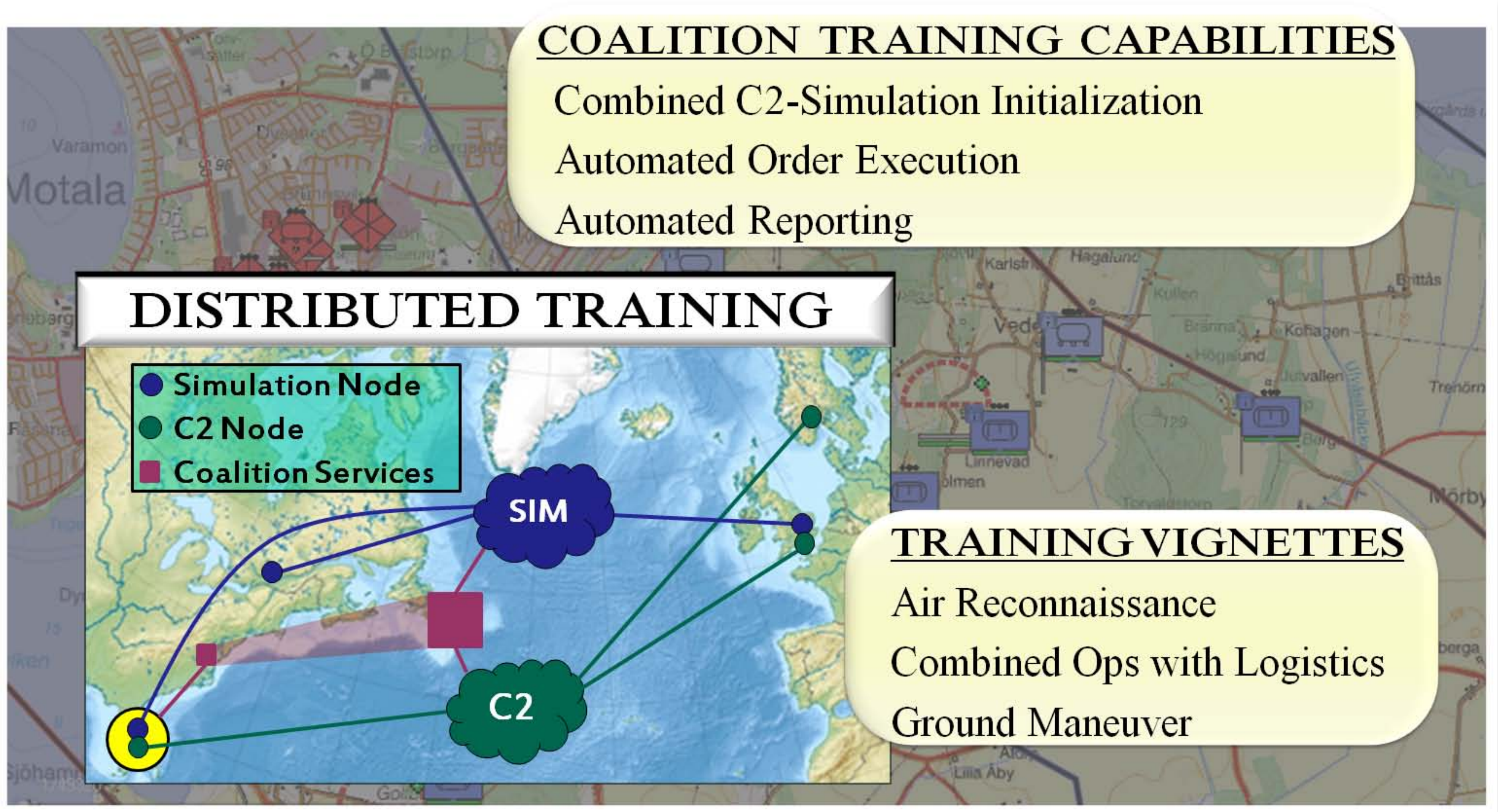
		STOPPED	INITIALIZING	READY	RUNNING	PAUSED	COMMENT
C2IS1	rpt1	stopped					power down
C2IS2	rpt2		setting-up				delay
C2IS3	rpt3		setting-up				
C2IS4	rpt4		MSDL Pushed				all OK
C2IS5	rpt5		setting-up				
C2IS6	rpt6		MSDL Pushed				waiting
C2IS7	rpt7	off-line					

Early Experience

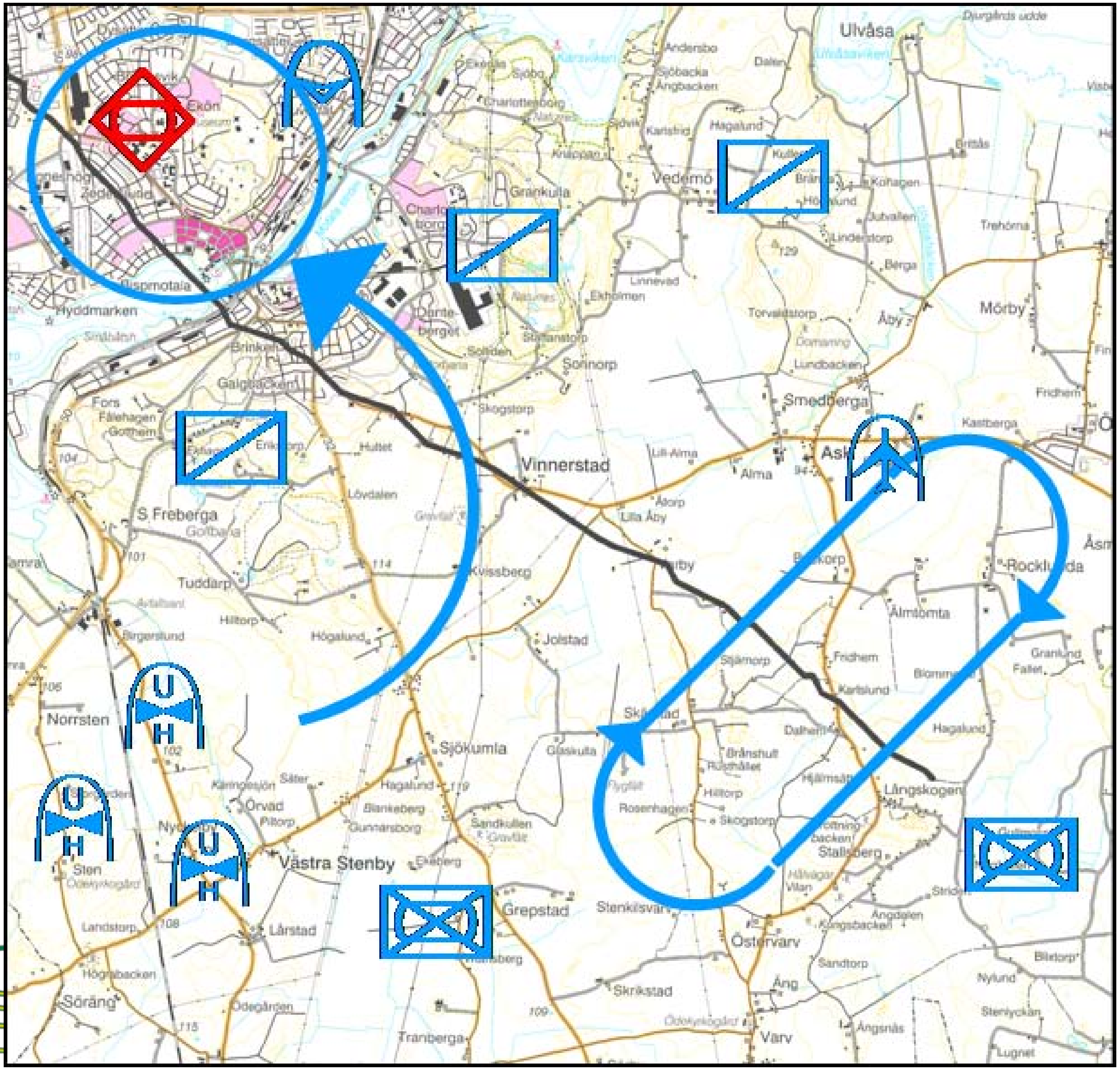
- Services described here were used in prototype form for InterService/Industry Training Simulation and Education Conference (I/ITSEC) 2011
  - Demo in NATO booth
- Part of overall effort to deploy effective operational experimentation by MSG-085
- Results were good:
  - Interoperated initialization and orders for UK, Norway, USA systems operating over Internet

MSG-085 I/ITSEC 2011 DEMONSTRATION

MSDL for initialization, C-BML for execution
Demo Harness 1: NOR, UK, US, FR   Demo Harness 2:

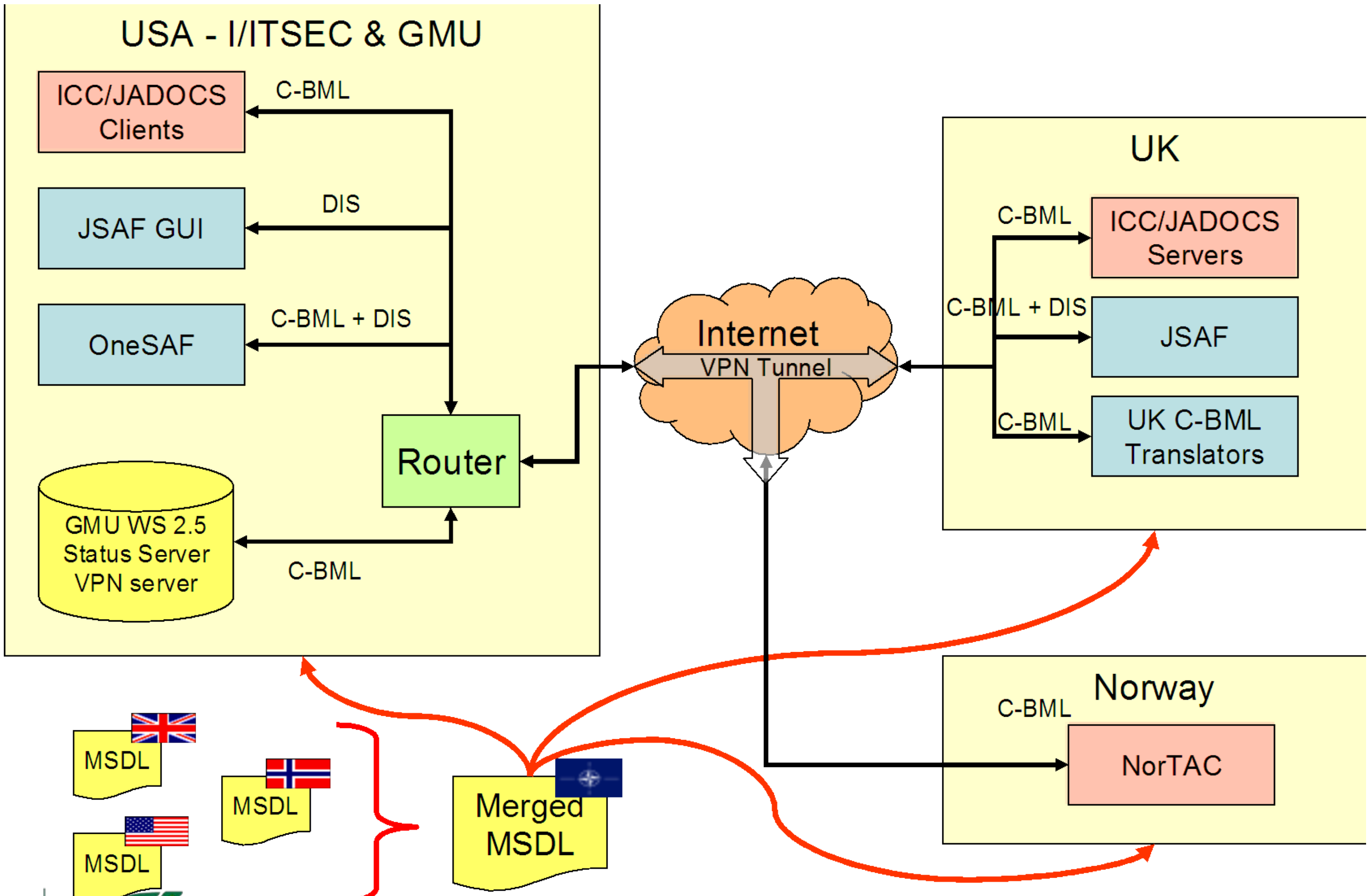


MSG-085 I/ITSEC'11
Recce Vignette: Operational View



GEORGE MASON UNIVERSITY

System Architecture for I/ITSEC 2011



Coalition Services Process I/ITSEC'11

- The overall process was coordinated using SMCS.
- Master Controller started MSDL service from GMU (VA)
- C2 system in Norway entered MSDL for basic scenario.
- Air systems in England added their MSDL inputs.
- SBMLServer published consolidated MSDL initialization.
- C2 systems in Norway (NorTAC) and England (ICC/JADOCS) submitted BML orders.
- JSAF simulation carried out the orders.
- Tactical Reports were generated by JSAF and returned to the C2 systems via the SBMLServer.

Conclusions

- Early experience successful
- More experience and convergence needed
- No evident technical barriers
- We hold to our vision:
  - When a future military coalition is formed, its components simply interconnect their networks, C2 systems, and simulation; authenticate, and start operations.
- Our open source software is intended to support progress toward that future.

http://c4i.gmu.edu/OpenBML