

A Web-Based Coordination System for MSDL/C-BML Coalitions

Priscilla McAndrews

Lisa Nicklas

Dr. Mark Pullen

GMU C4I Center

Presentation Outline

- Introduction
 - C-BML/MSDL Coalitions
- Status Monitor and Control:
 - What and Why
 - Design
 - Automated interface
- Experience: I/ITSEC 2011
- Conclusions

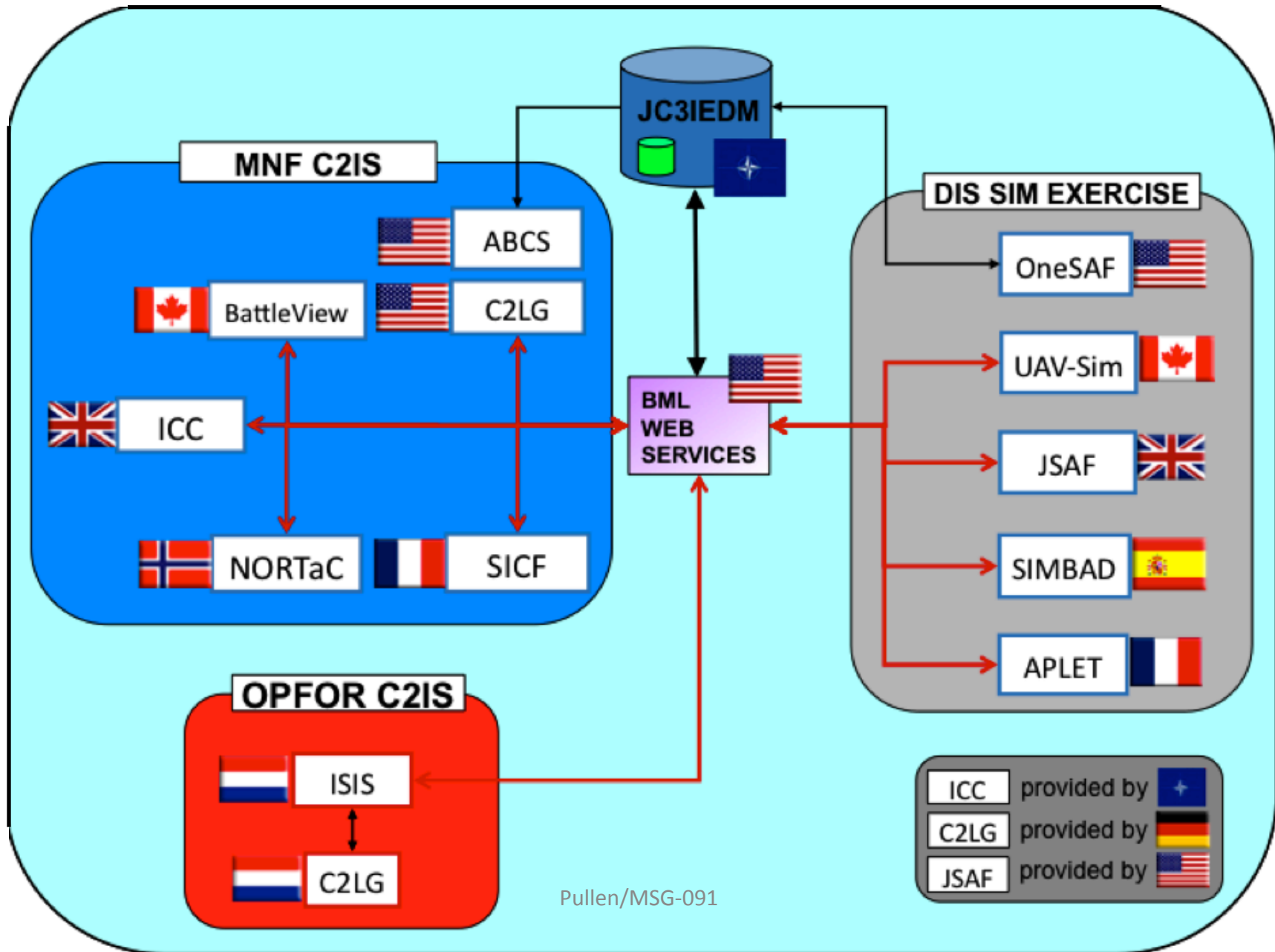
Background

- SISO is developing two standards to support C2-simulation interoperation for international coalitions
- Military Scenario Definition Language (MSDL)
 - Standardizes initialization of simulation systems
 - Recently shown also to be useful with C2 systems
- Coalition Battle Management Language (C-BML)
 - Orders, Requests, and Reports among C2 systems and simulations
- This paper addresses a problem arising from the complexity of operating such coalitions
 - Need to synchronize operations of the various systems
 - Need to keep the operators informed as to status

BML in NATO Modeling and Simulation Group (MSG)

- MSG has been nexus for BML cooperation
- US Joint BML project with France Aplet
 - Early demonstration 2006
 - Led to Exploratory Team 16
- MSG-048 Technical Activity started 2007
 - Demonstrations at I/ITSEC 2007 and 2008
 - One-week experimentation 2009 (9 nations)
- MSG-085 Technical Activity started 2010
 - Goal: operational use of MSDL/C-BML (12 nations)

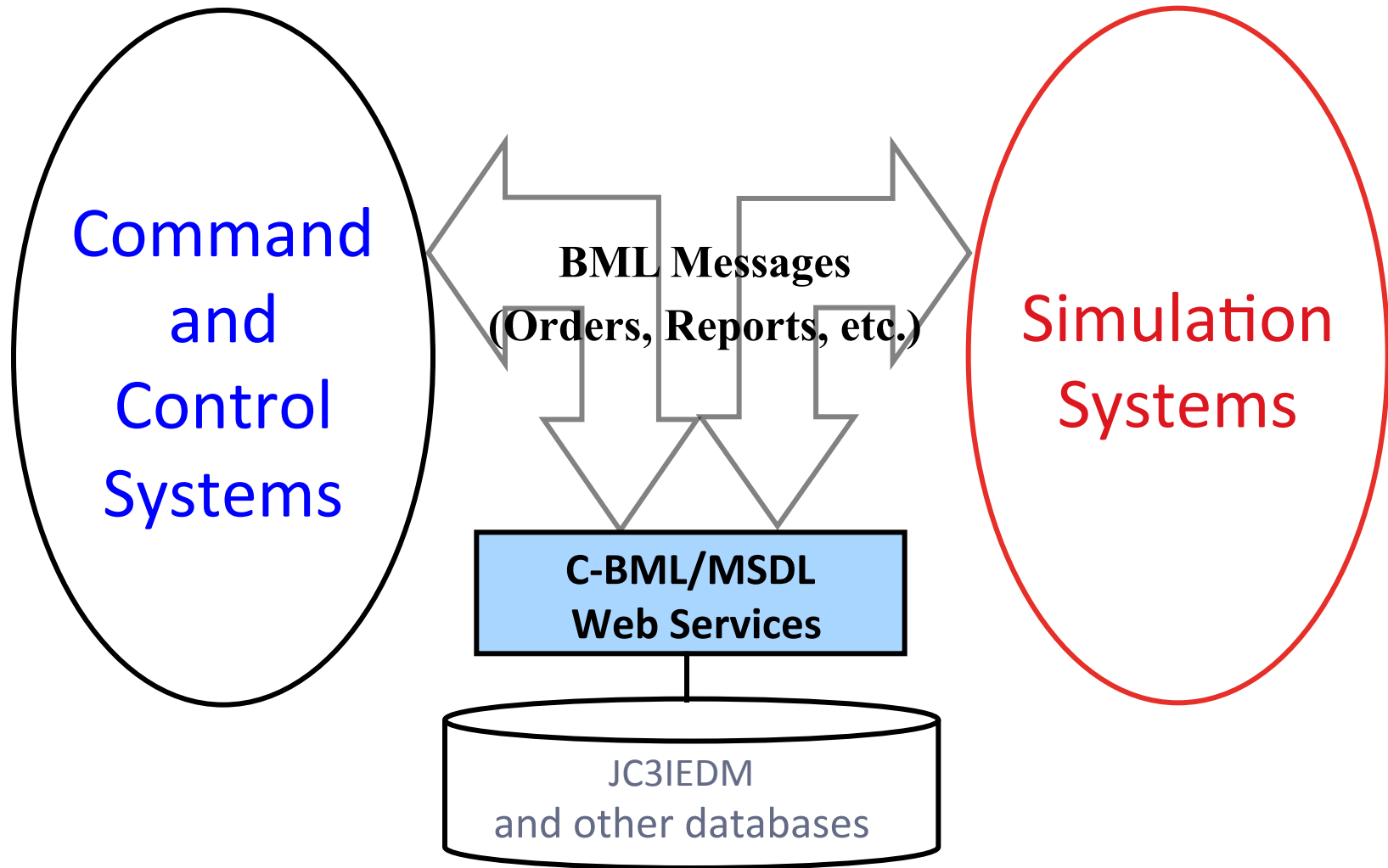
Example: MSG-048 2009 Architecture



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

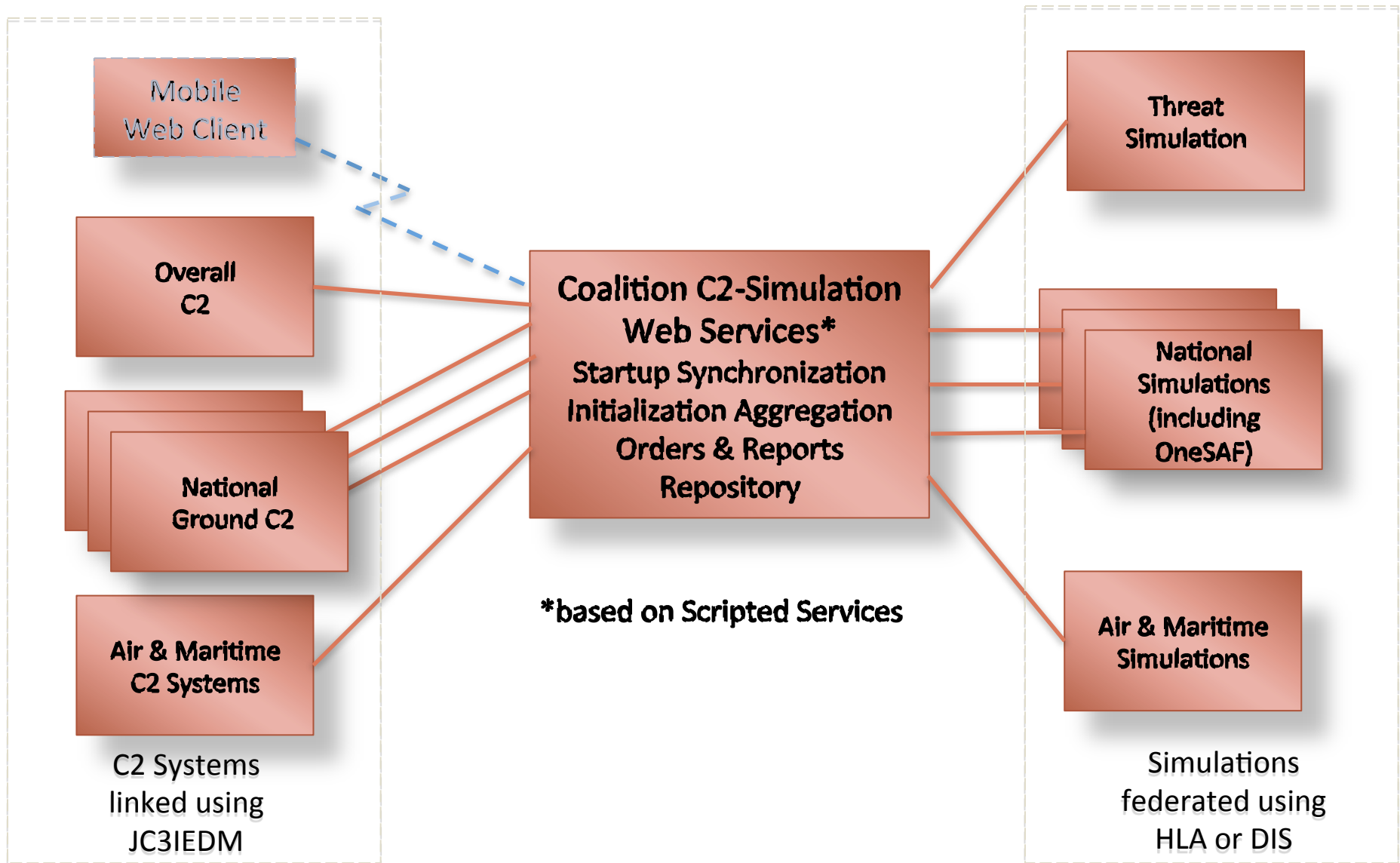
Generic BML Architecture



MSDL Data Components

- Geographic Region of Interest
- Force/Sides
- Units
- Equipment
- Installations
- Overlays
- Graphics

Coalition C2-Simulation Architecture



Web-Based Coordination System

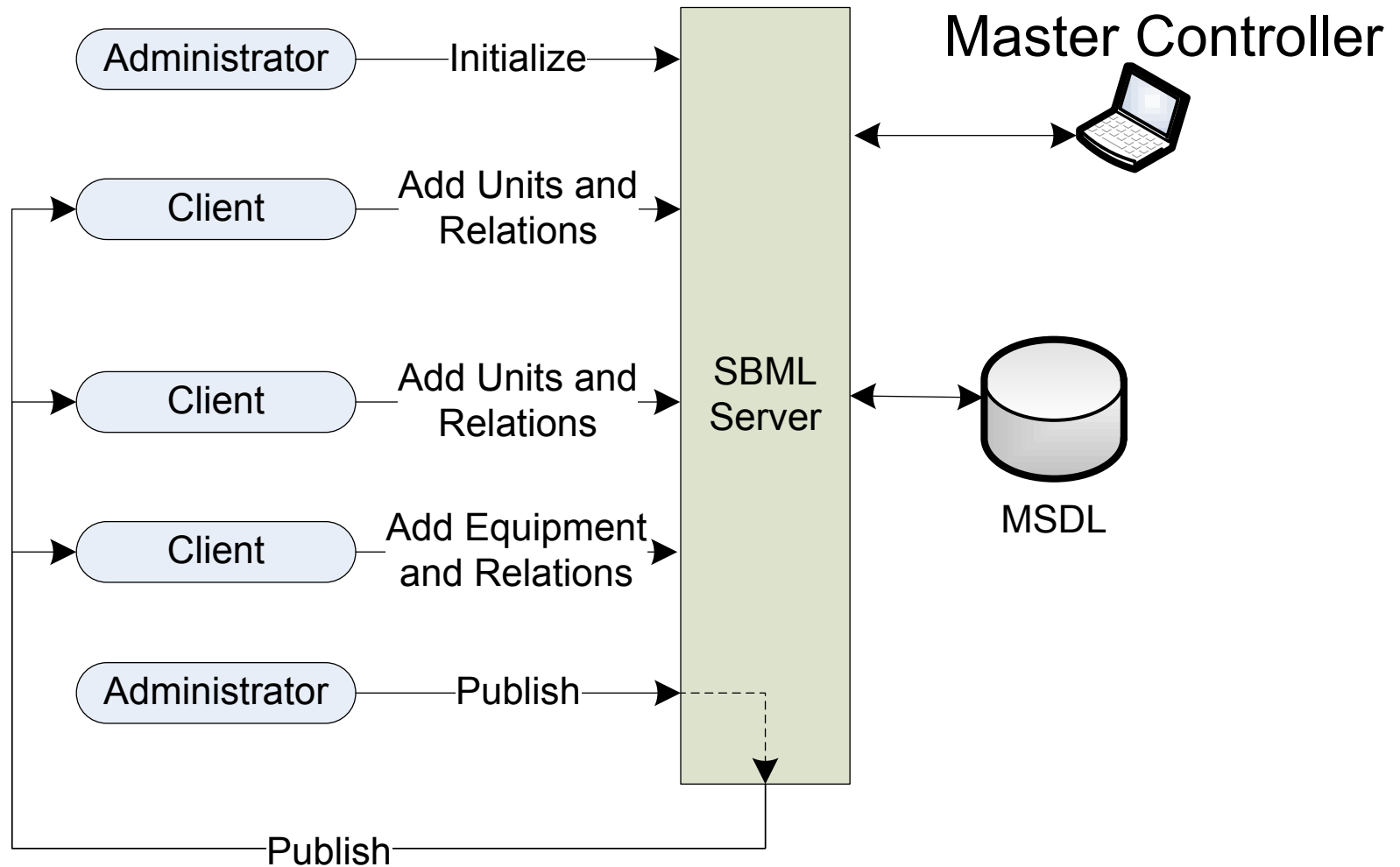
Status Monitor and Control Functions

- Start, pause, and stop all coalition systems in a synchronized way
- Participating systems
 - Coalition Server – accepts XML documents and publishes to participating systems
 - SMC Server – interacts with humans via webpage; may also interact with other systems via XML
 - C2 System Clients – coalition command and control
 - Simulation System Clients – simulate coalition forces
 - Master Controller – provides overall intended status
 - Client System Controllers – operate C2 & simulation
 - Observers – monitor status but control nothing

C2 and Simulation Systems are Different

- C2 systems:
 - Provide orders and display situation
 - May not require MSDL initialization
 - May not have a concept of “stopped” or “paused” since military operations are continuous
- Simulation systems:
 - Generate data from model - representing expected actions of coalition forces and their results
 - May perform checkpoint/restart

MSDL Server Operation



Master Controller Inputs

- Stop
- Initialize (start MSDL generation & loading)
- Run
- Pause

MSG-085 Status Monitor

MASTER CONTROLLER STATUS

Scenario: scenario1

Current Order: **stop**

Scenario MSDL status: not started

Number of observers: 2

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		setting up				
C2IS2	rpt2		setting up				new timeout
C2IS3	rpt3	stopped					
C2IS4	rpt4	offline					
C2IS5	rpt5	offline					
C2IS6	rpt6	offline					
C2IS7	rpt7	offline					
clientName1	clientId1	stopped					new comment

Client Status Phases

- Stopped
- Initializing
- Ready
- Running
- Paused

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:

[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					

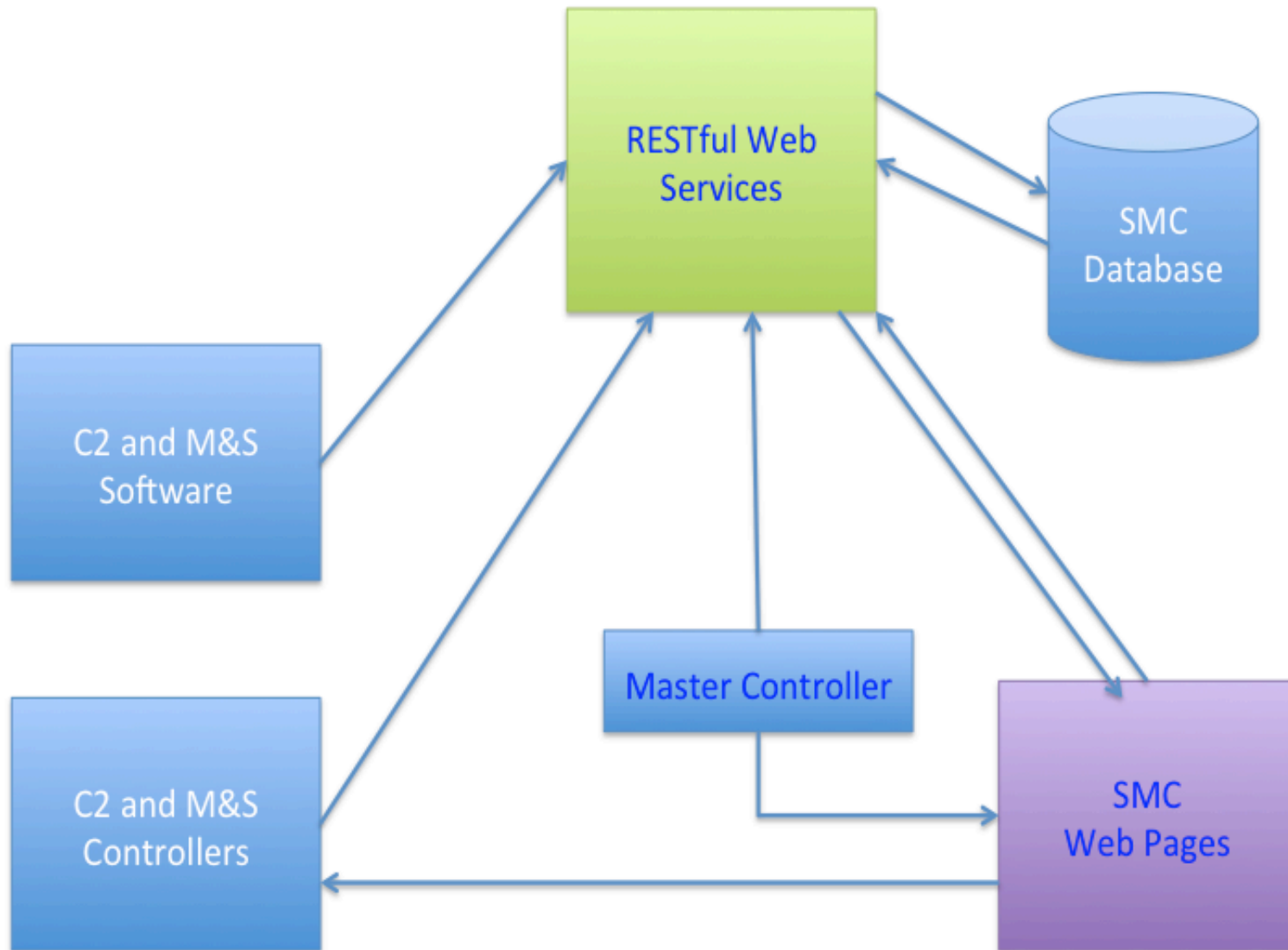
SMC Layout

- General layout:
 - Heading (identifies system/role)
 - Current directive from Master Controller
 - MSDL Scenario name and status
 - Number of observers logged in
- Input areas:
 - Overall directive (Master Controller only)
 - Dropdown to provide state (Client Controller only)
 - Text box for comments
- Output area:
 - Last provided status of each client (initially *stopped*)

Automated Interface

- Human Client Controllers are not good status sources: they experience many distractions
- We have created a RESTful Web Service that lets the C2 and simulation systems interface with SMC
- Input methods:
 - c2SimAdd
 - c2SimModSlientStatus
 - csSimModClient
 - c2SimModSessionKey
 - csSimModAutomated
- Output: C2 and simulations can subscribe to status

Interaction Control Flow



SMC Experience: I/ITSEC 2011

MSG-085 I/ITSEC 2011 DEMONSTRATION

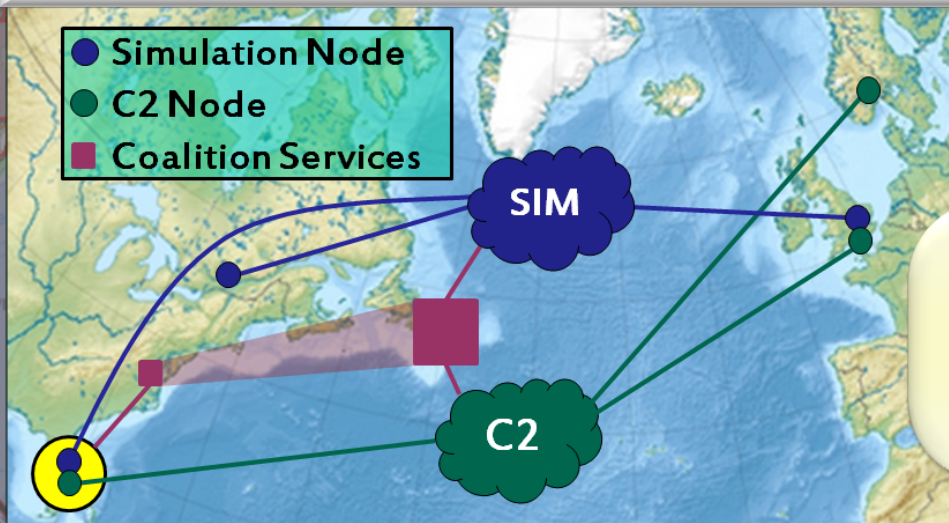
MSDL for initialization, C-BML for execution
Demo Harness 1: NOR, UK, US

COALITION TRAINING CAPABILITIES

Combined C2-Simulation Initialization
Automated Order Execution
Automated Reporting

DISTRIBUTED TRAINING

- Simulation Node
- C2 Node
- Coalition Services

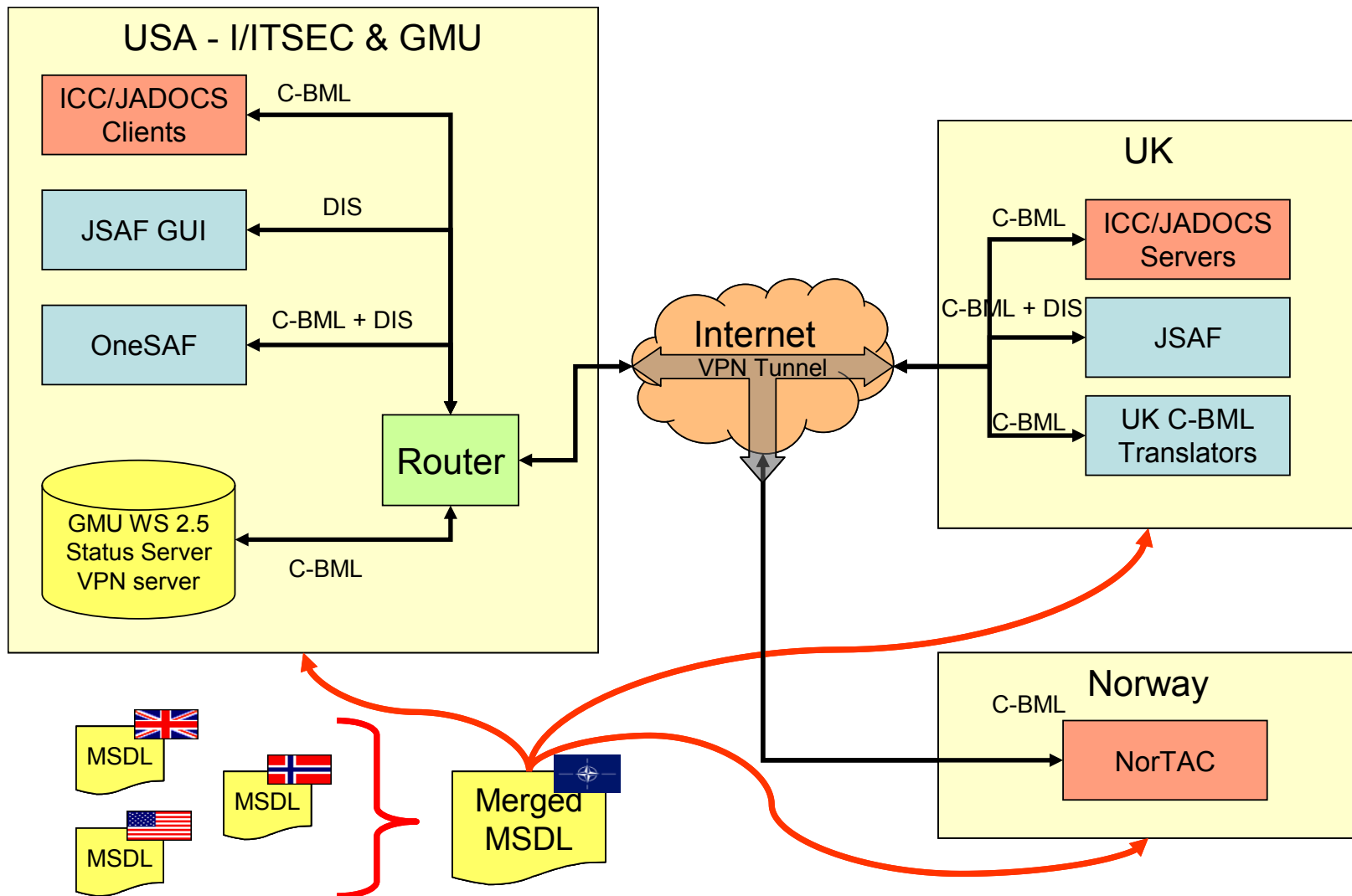


TRAINING VIGNETTES

Air Reconnaissance
Combined Ops with Logistics
Ground Maneuver

Web-Based Coordination System

System Architecture for I/ITSEC 2011 DH1



Web-Based Coordination System

Conclusions

- Experience indicates SMC is useful and necessary
 - Both I/ITSEC and continued coalition testing
- Webpage-based play is simple and effective
- Human status input is a weak link
- For best results, software-to-software interface via Web Service is best
- SMC is available as open source from <http://c4i.gmu.edu/OpenBML>