

Client Perspective: Implementing C2SIM in a Client

Dr. Robert Wittman

APPROVED FOR PUBLIC RELEASE

Topics

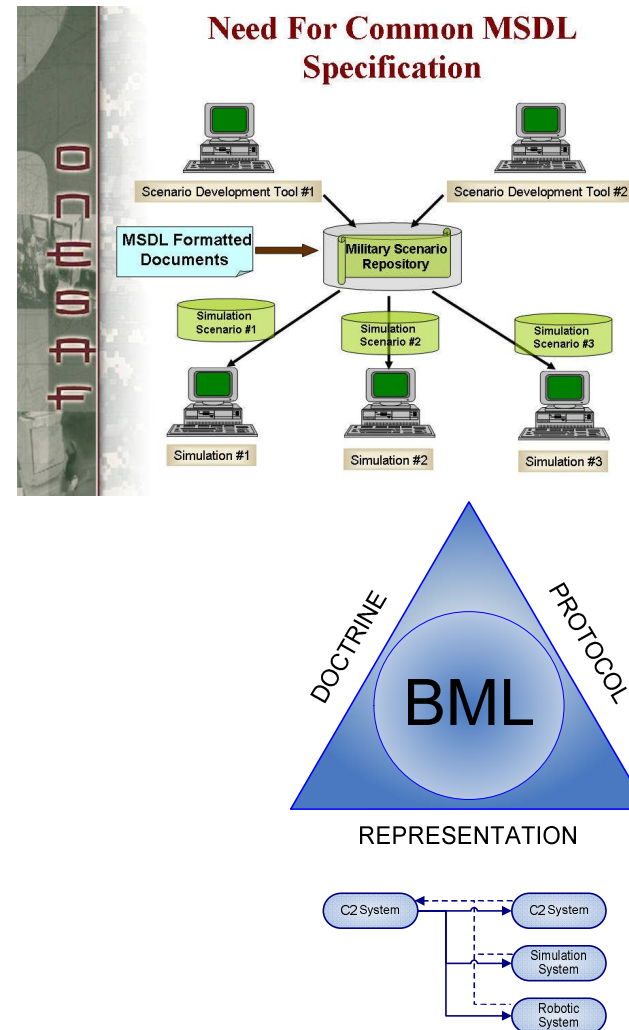
- **Why use C2SIM (MSDL & C-BML)**
- **The Crawl, Walk, Run Implementation Approach**
- **An Coalition Example**
- **Client Side Lessons Learned & Related Core Capabilities**
- **Future C2SIM Architecture**

Client-Side Experience

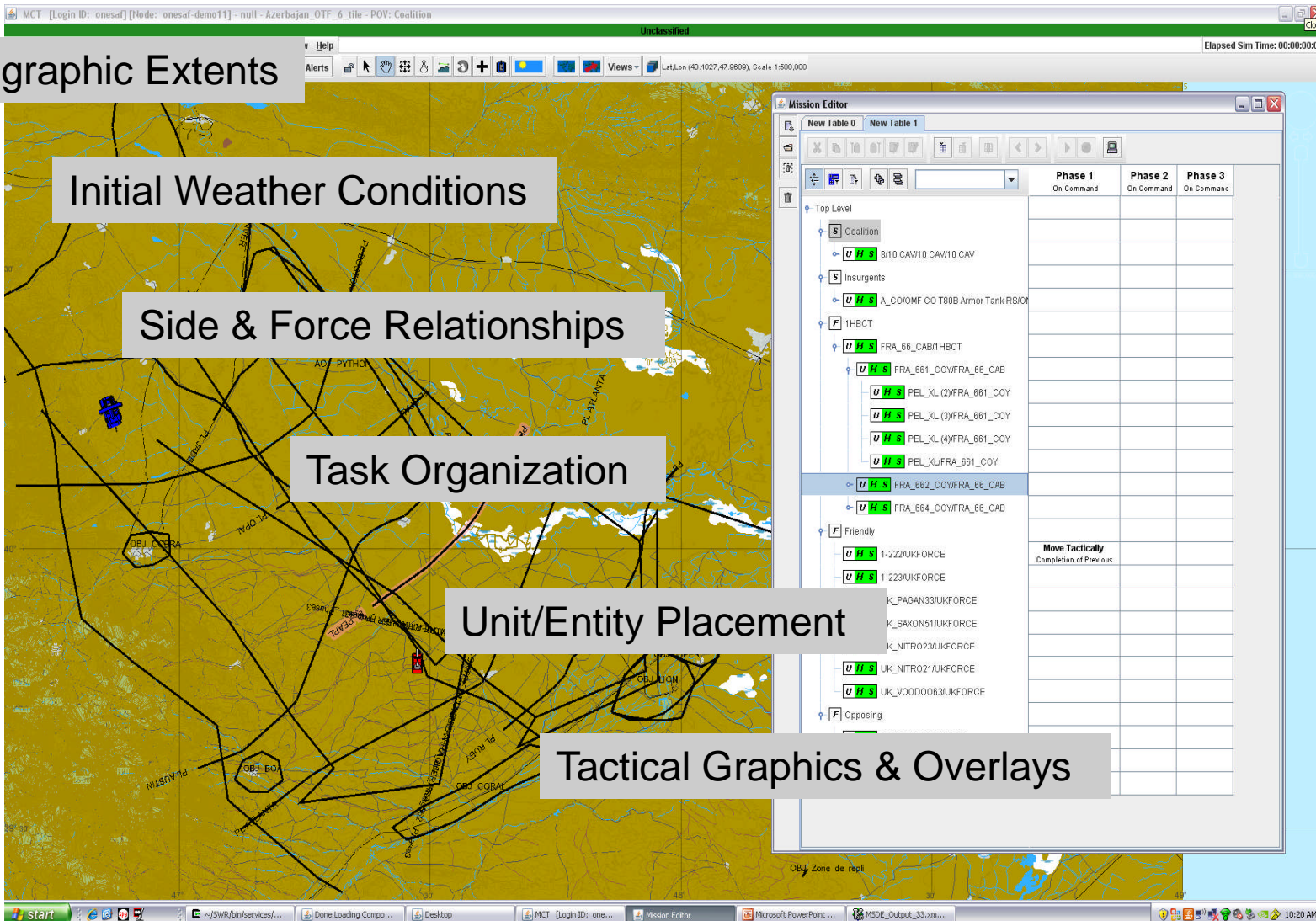
- **US Simulation System Development**
 - OneSAF MSDL & C-BML Capability Development
- **Standards Development**
 - MSDL & C-BML standards development leadership and participation
- **NATO MSG-085 Coalition Wide Scenario Collaboration & Demonstrations**
 - ITEC 2011, 2013
 - I/ITSEC 2011, 2012, 2013
 - Ft Leavenworth Final Demo Dec 2013

Why Use MSDL & CBML Data Format

- **Import format versus native format**
 - Allows standard formats (MSDL & C-BML) and specific simulation initialization & order, tasking, & reporting native formats to evolve at their own pace
 - Allows simulations to define specific information needs
 - Allows standard format for scenario start data and runtime order-based exchanges for all participating systems



What the MSDL Data Model Provides



Geographic Extents

Initial Weather Conditions

Side & Force Relationships

Task Organization

Unit/Entity Placement

Tactical Graphics & Overlays

Mission Editor

	Phase 1	Phase 2	Phase 3
Top Level	On Command	On Command	On Command
Coalition			
810 CAV10 CAV10 CAV			
Insurgents			
A_COIOMF CO T80B Armor Tank RSIOT			
1HBCT			
FRA_66_CAB1HBCT			
FRA_661_COYFRA_66_CAB			
PEL_XL(2)FRA_661_COY			
PEL_XL(3)FRA_661_COY			
PEL_XL(4)FRA_661_COY			
PEL_XLFRA_661_COY			
FRA_662_COYFRA_66_CAB			
FRA_664_COYFRA_66_CAB			
Friendly			
1-222UKFORCE			
1-223UKFORCE			
K_PAGAN33UKFORCE			
K_SAXON51UKFORCE			
K_NITRO23UKFORCE			
UK_NITRO21UKFORCE			
UK_VOODOO63UKFORCE			
Opposing			

What the C-BML Data Model Provides

- **C-BML provides a standard data model (XML) for automated sharing of military and non-military orders, reports, and tasks**
 - Who: Identifies commanders & subordinate (unit or individual)
 - What: Identifies task to be performed (i.e. occupy area)
 - Why: Identifies rationale for performing the task (defend, defeat, suppress, etc.)
 - When: Identifies the time frame for performing the task
 - Where: Identifies the location of the task

Topics

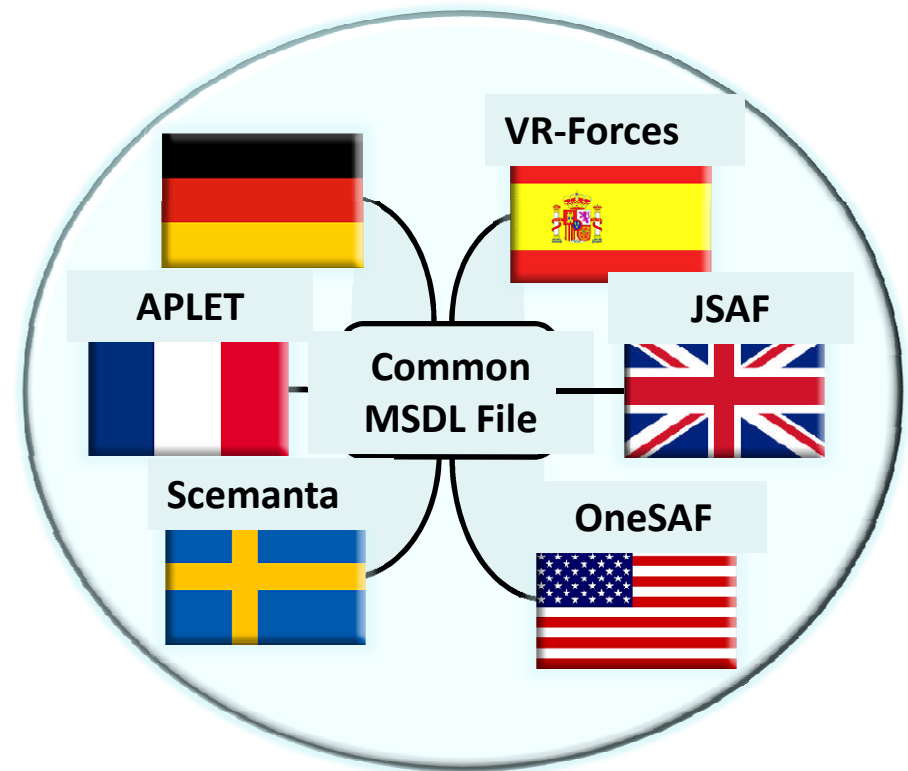
- Why use C2SIM (MSDL & C-BML)
- **The Crawl, Walk, Run Implementation Approach**
- A Coalition Example
- Client Side Lessons Learned & Related Core Capabilities
- Future C2SIM Architecture

2010/2011 - Crawl Stage 1/3

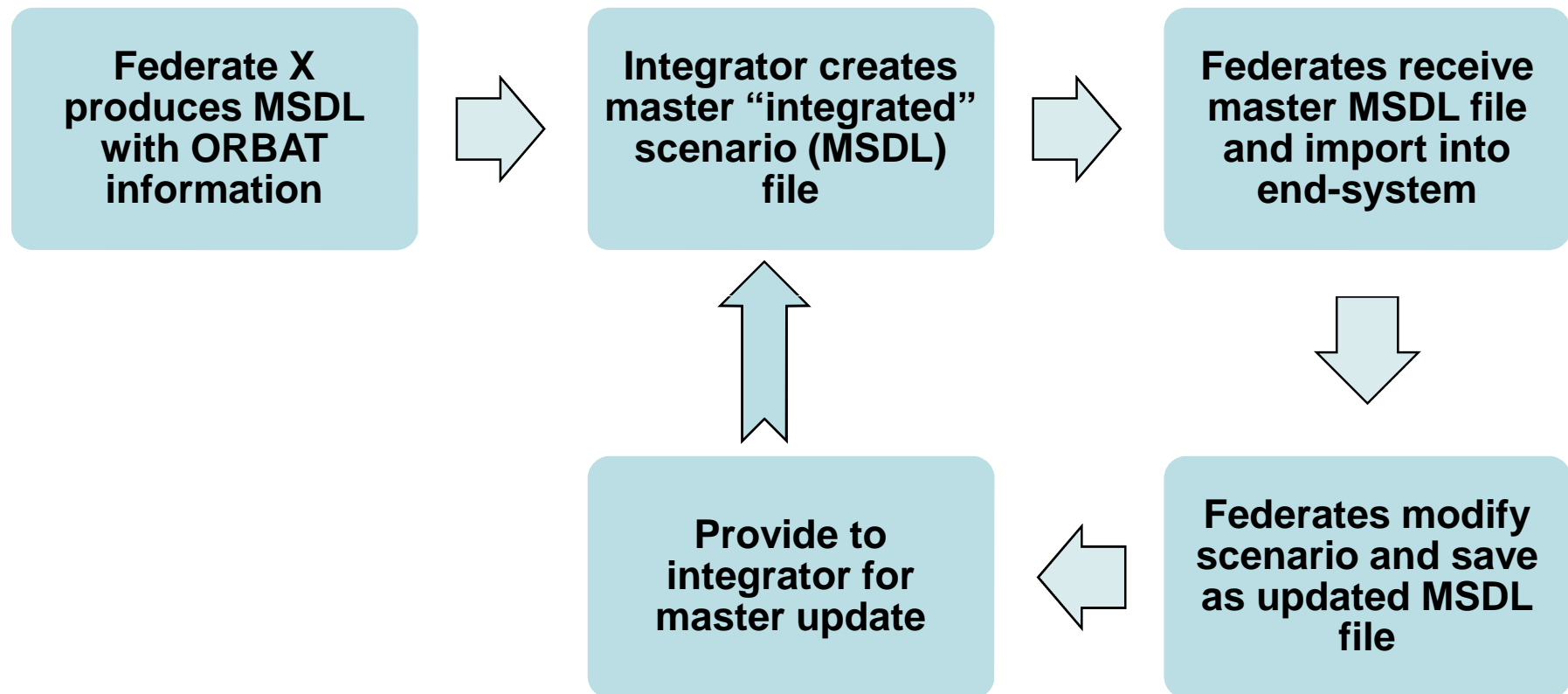
- **Development timeline 3 month time-box**

- Multi-federate initialization from common MSDL file
 - Consistent unit/entity mapping
 - Initial locations
 - Tactical graphics
 - Geographic extents
- Initial battle book development
 - Combined spreadsheet
- Initial participants
 - DEU, ESP, GBR, FRA, SWE, USA

Initial MSDL Experience NATO MSG-085

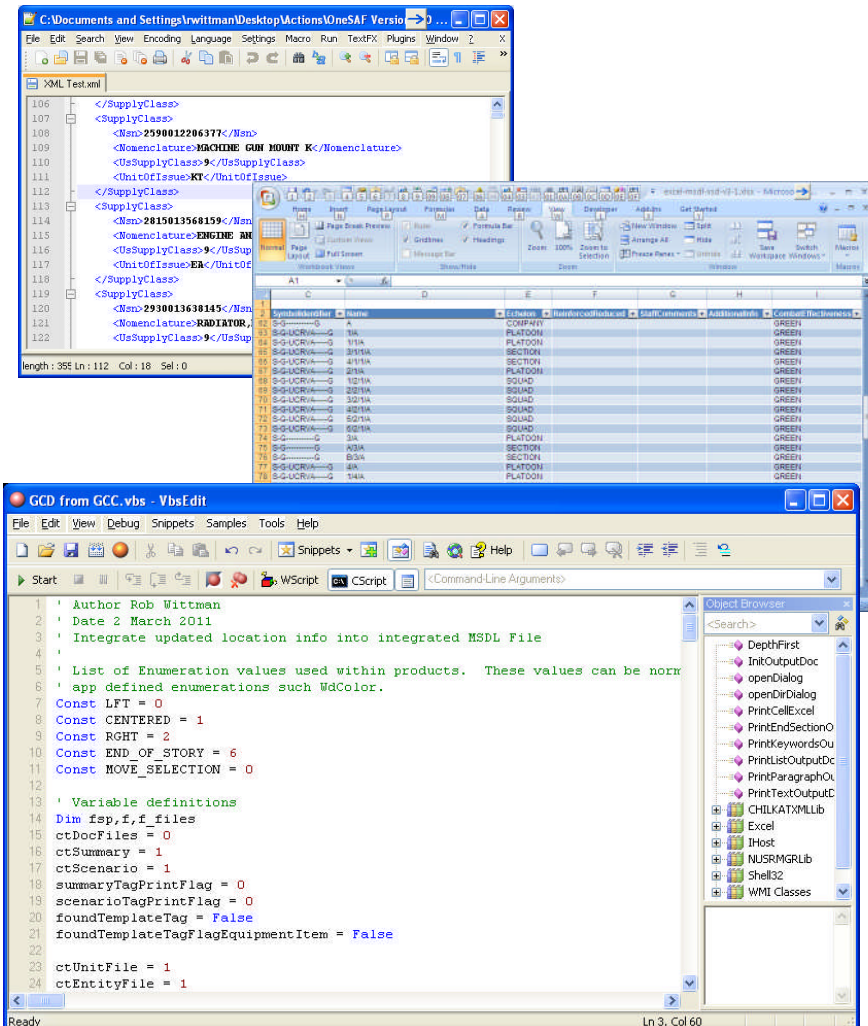


Crawling: Initialization Process



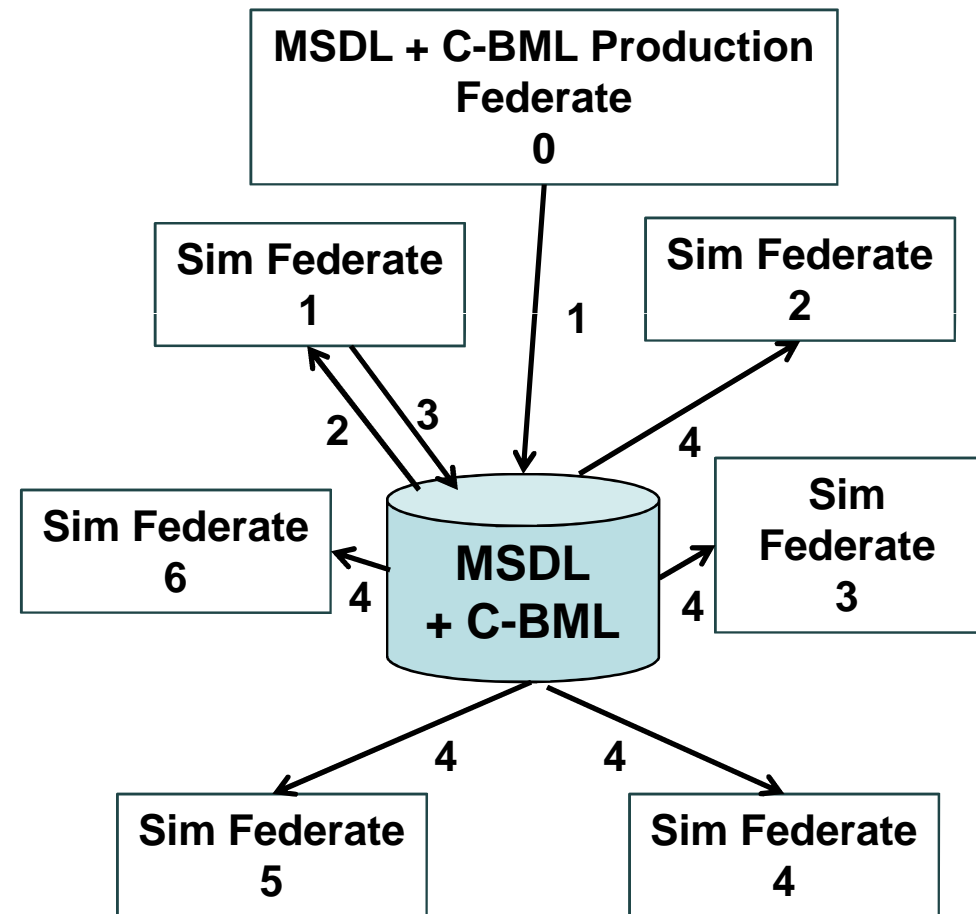
Crawling: Initialization Integrator 3/3

- **Integration tools – current & evolving**
 - Excel to list, identify, and manage all contributed data
 - Notepad++ for edit, cut-and-paste
 - Visual Basic scripting
- **Coordinating decisions**
 - XML namespace
 - ScenarioID information
 - Coordinate system
 - Environment extents
 - Sides and Forces
 - Mixed ORBAT



2011/2012 - Walking 1/3

- **Transition federation from initialization to running state**
 - Provide link to planned set of C-BML orders referencing MSDL units, entities, and tactical graphics
- **Formalize round trip initialization process**
 - Initialize mission planning system with original MSDL
 - Save MPS output in MSDL and C-BML formats
 - Use files to initialize remaining federates

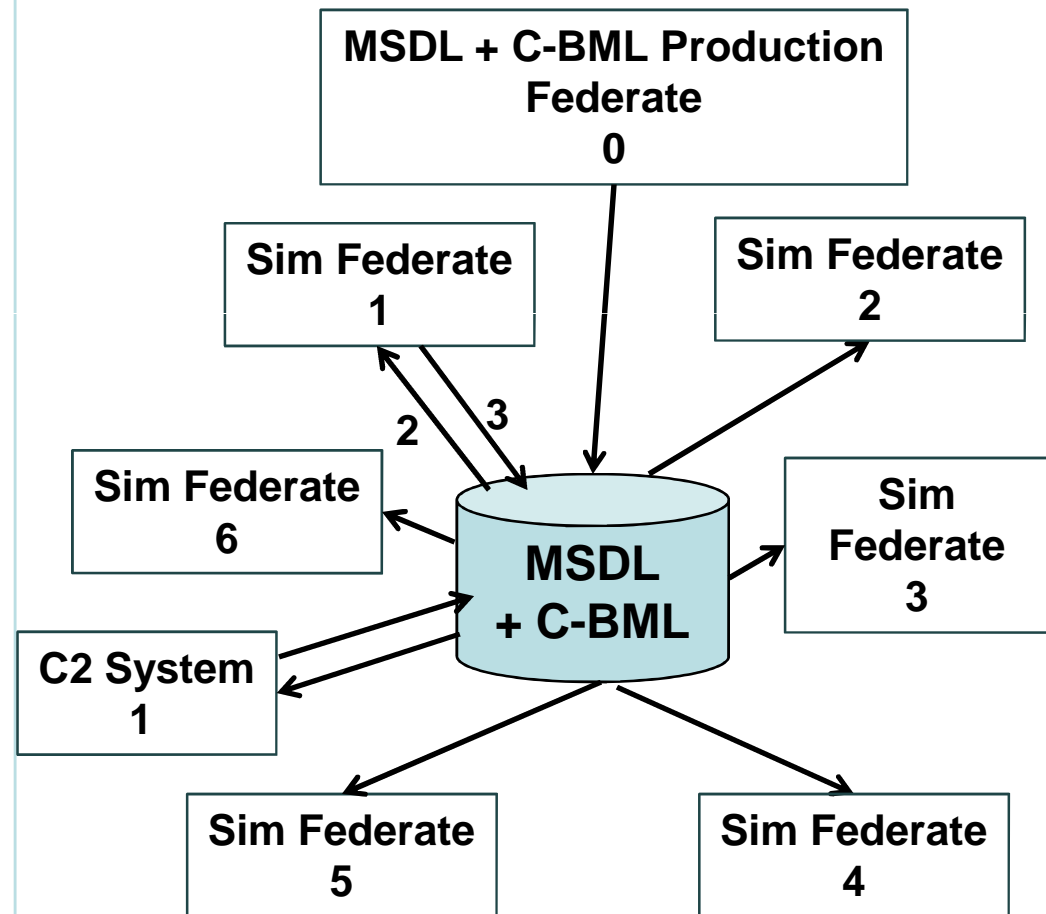


Walking 2/3

- **More complex initialization-based federation agreements**
 - Introduce unique identifier federate ranges for important items units, entities, messages, reports, etc.
 - Introduce standards-based entity & unit name conventions (e.g. DIS enumerations)
 - Create or locate MSDL element to house enumeration or code list
 - Identify FOM for future integrated HLA federation activities
 - Introduce multiple terrain extents
 - Formalize sequence for drawing tactical graphics
 - Continue integrator tool development to prove early, pre-initialization-time federate feedback

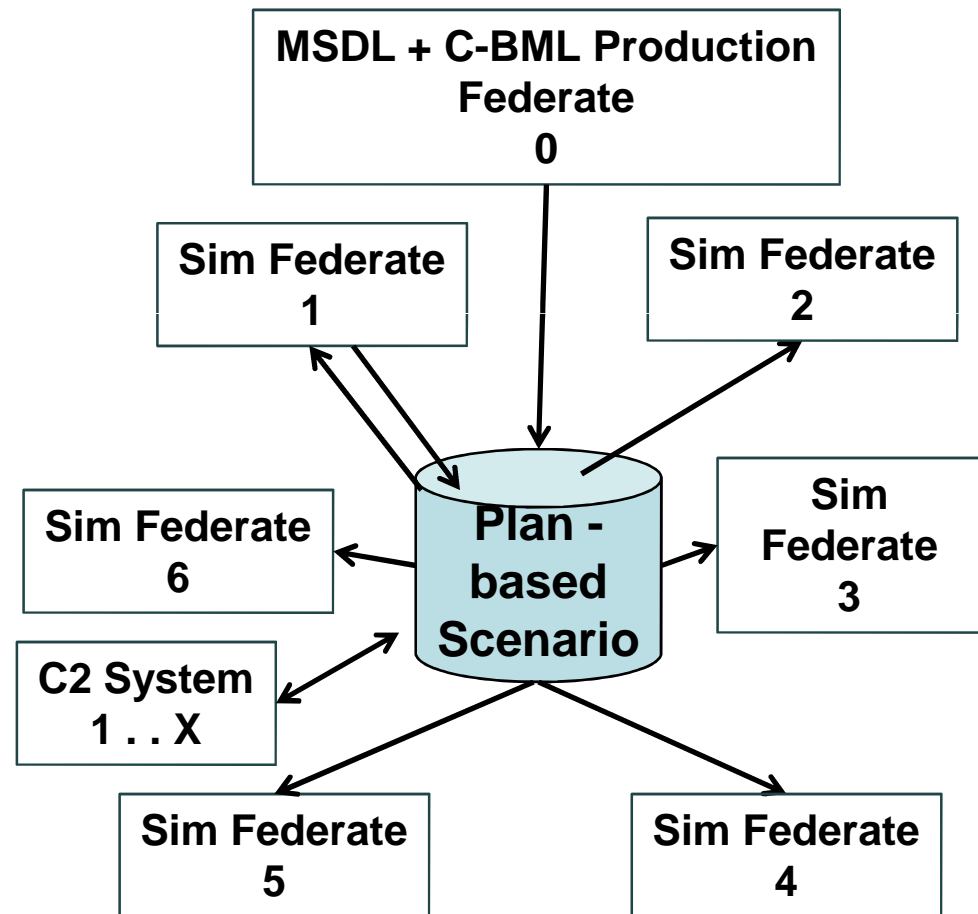
Walking 3/3

- **Introduce C2 systems**
 - Simple C2 Initialization of the Joint Advanced Deep Operational Coordination System (JADOCS) with friendly ORBAT
- **Initialize GMU web-services with MSDL**
- **Extend MSDL participation with enhanced battle book**
- **Provide feedback to C2SIM PDG**



2013/2014 - Running

- Multi-simulation and C2 coalition initialization process
- Fully integrated MSDL and C-BML
- Demonstrate in operationally useful event
- Extend participant list
- Feedback to SISO
- Increase coalition speed and agility

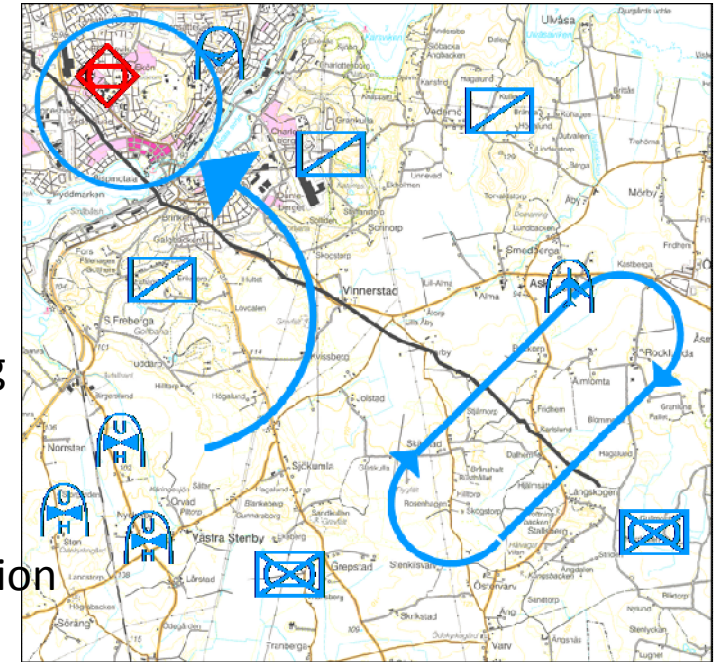


Topics

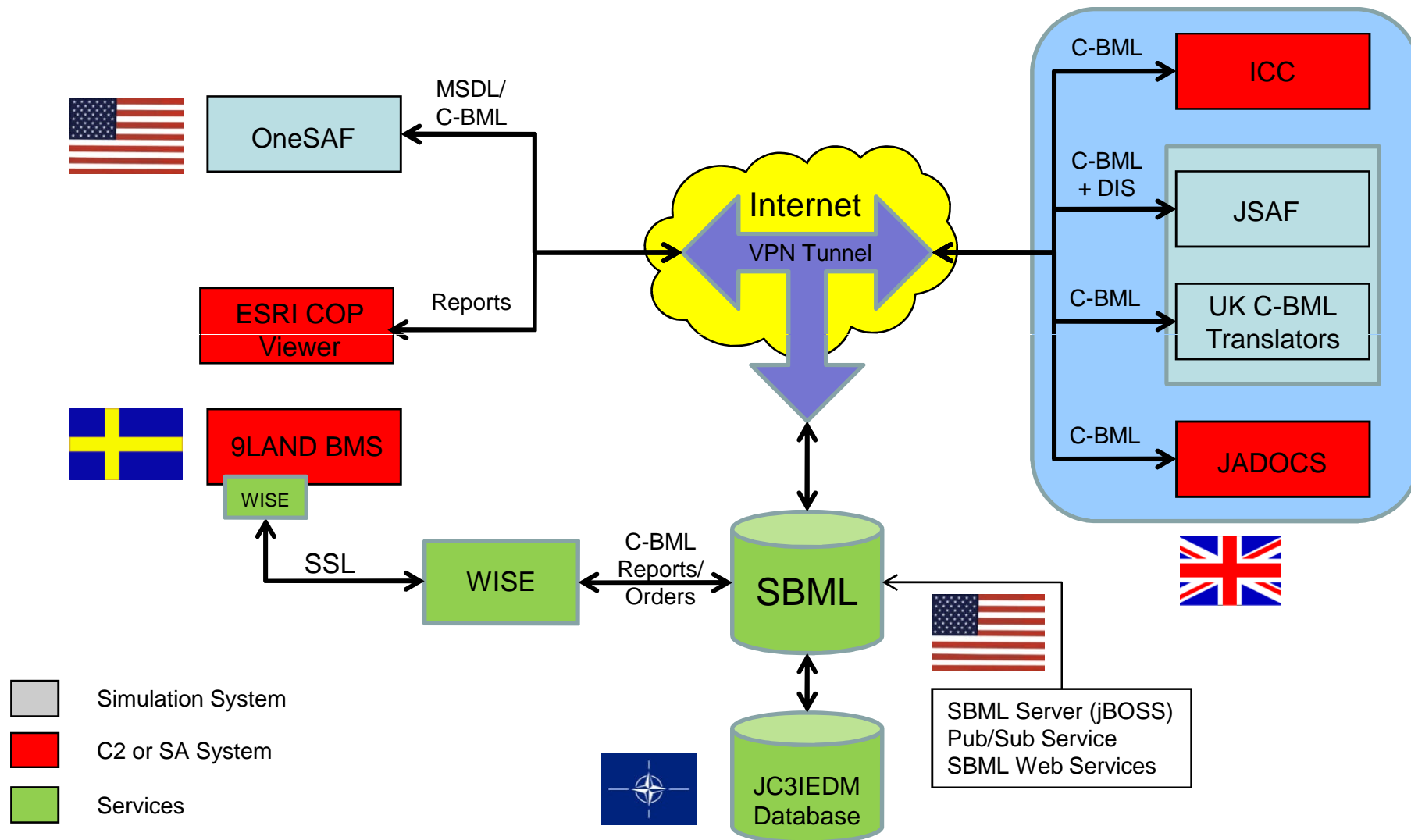
- Why use C2SIM (MSDL & C-BML)
- The Crawl, Walk, Run Implementation Approach
- **A Coalition Example**
- Client Side Lessons Learned & Related Core Capabilities
- Future C2SIM Architecture

ITEC 2013 AIR-LAND VIGNETTE: SWEDEN, UK & USA

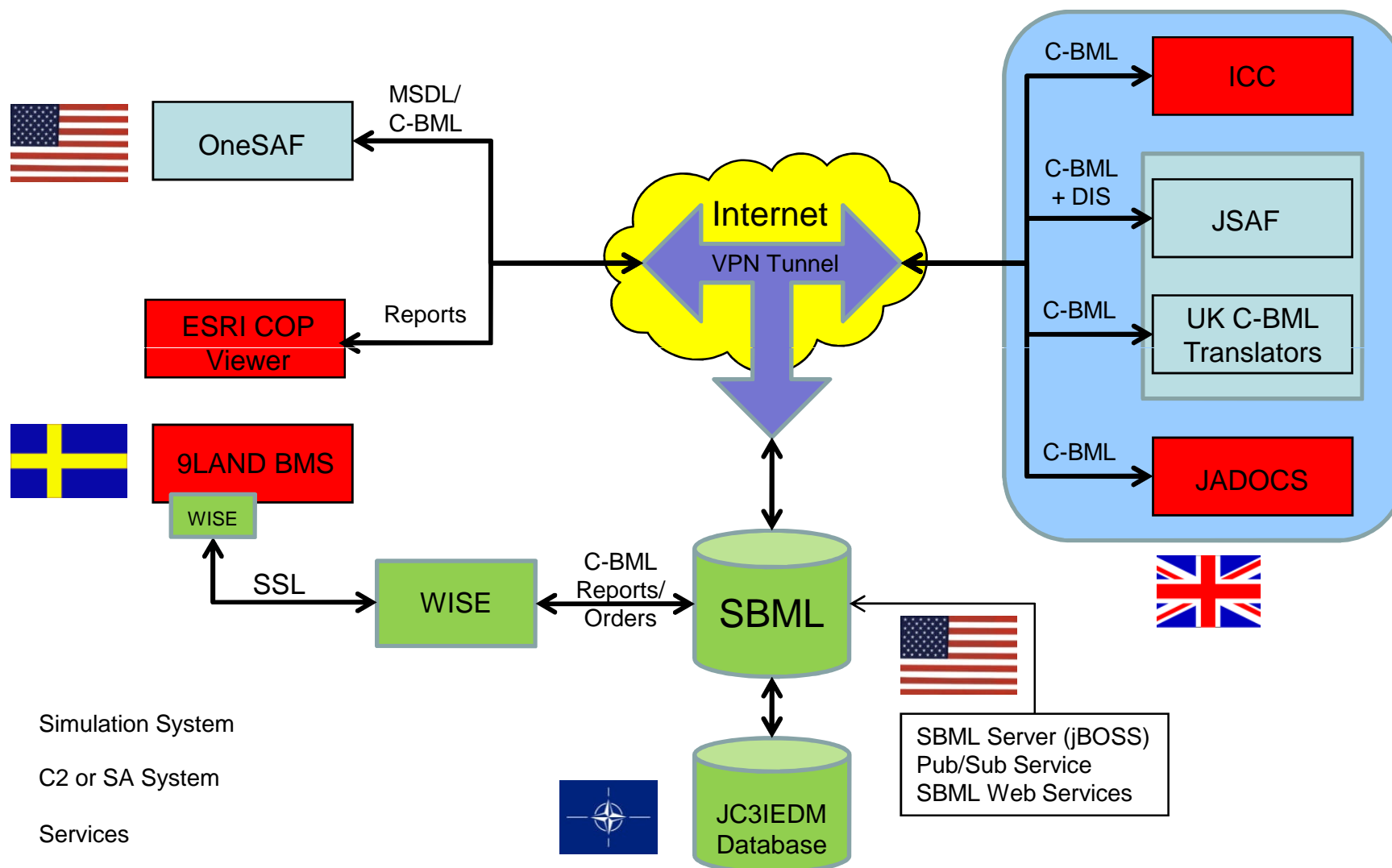
- **Operational Thread and Capability Demonstrated**
 - Recce/integrated Air-Land vignette w BOGALAND Scenario
 - Ground force Recce cues UAV & CAS mission
 - Insertion of troops by helicopter
 - Enables mission planning & rehearsal & C2 training
- **Technical Capability**
 - MSDL/C-BML infrastructure information
 - MSDL toolset/population mechanism for initialization
 - Scripted BML Server 2.5 bridges multiple schema version
 - WISE (SAAB)/SBML server and populating 9LandBMS
 - Use of Esri Tracking server for visualization
 - C2 systems: ICC, JADOCS using JSAF and OneSAF
 - WAN-based distribution

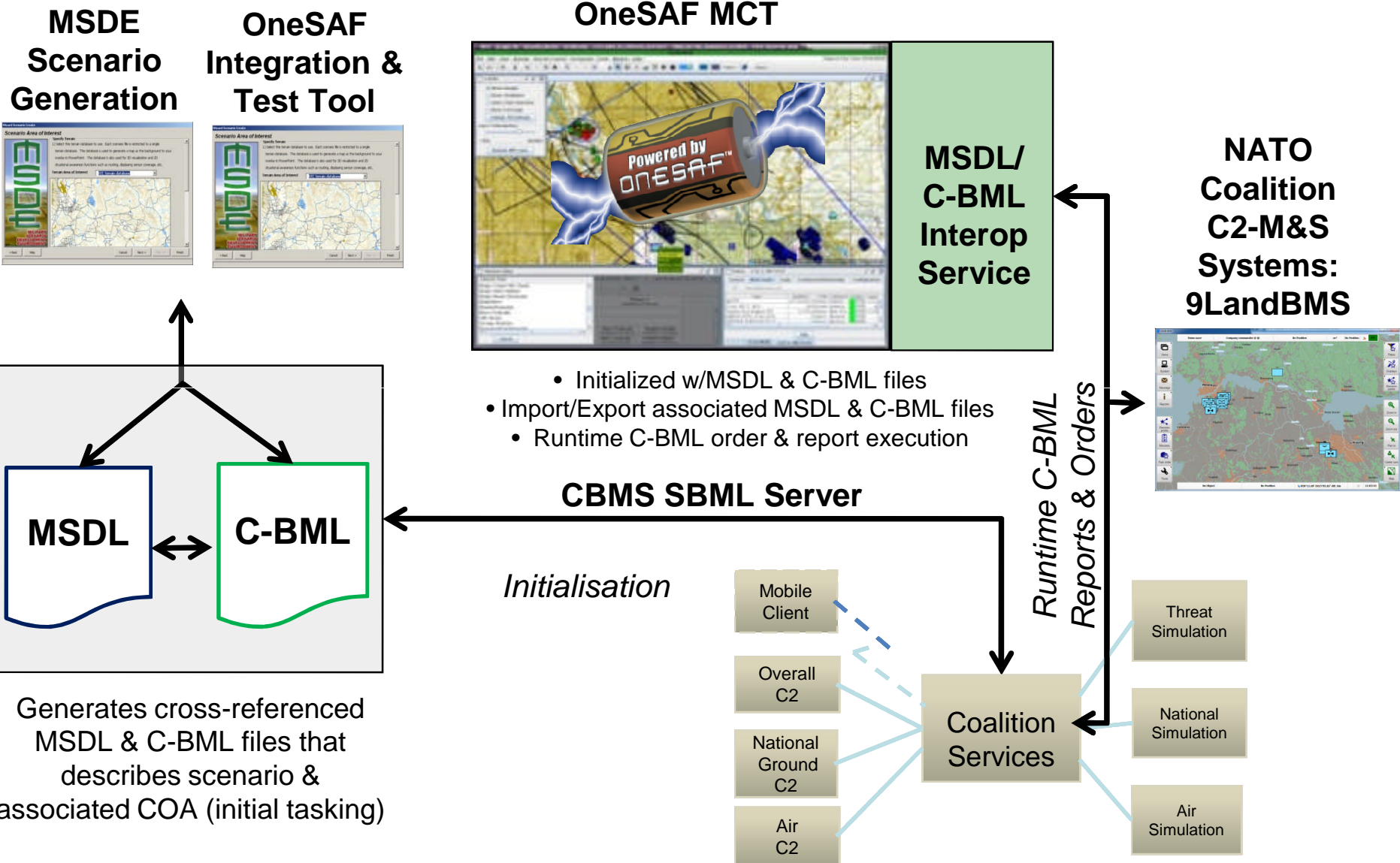


DEMO – TECHNICAL ARCHITECTURE



DEMO – TECHNICAL ARCHITECTURE



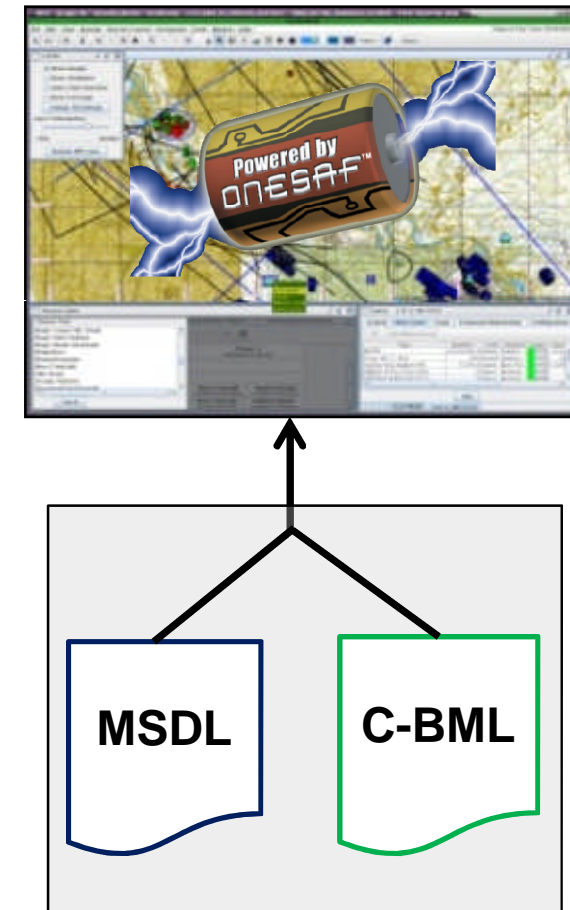


Topics

- Why use C2SIM (MSDL & C-BML)
- The Crawl, Walk, Run implementation approach
- A Coalition Example
- **Client Side Lessons Learned & Related Core Capabilities**
- Future C2SIM Architecture

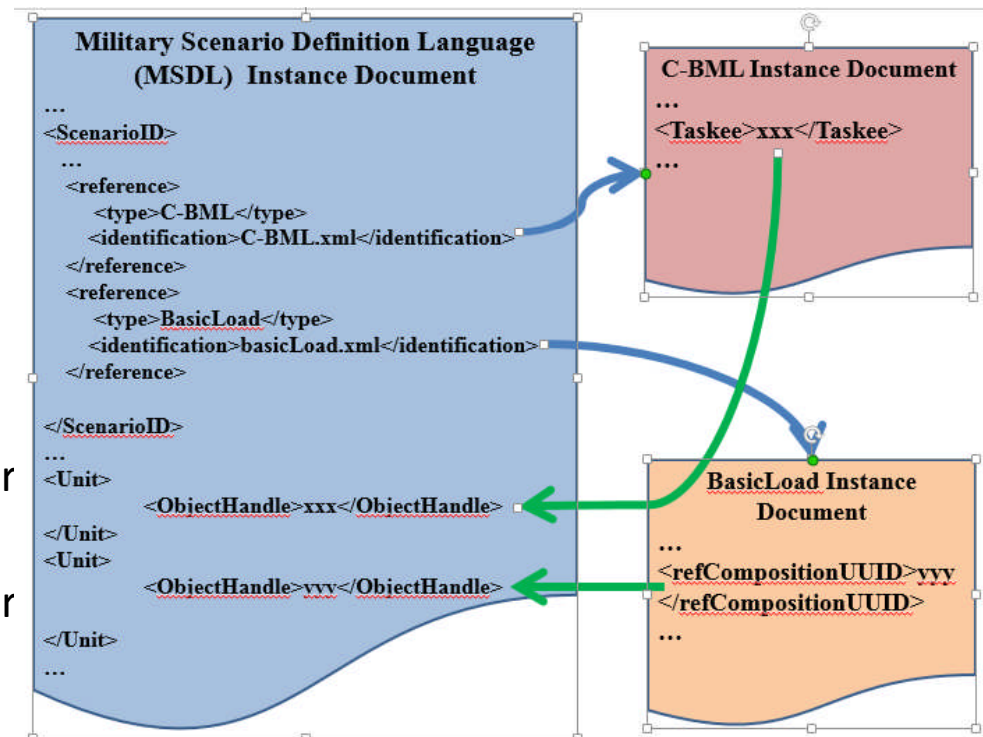
Lessons Learned Core Capabilities – Import (1/2)

- **Read/Write MSDL & C-BML scenario data**
 - Leverage standardized XML read/write libraries
 - Support local and web-enabled subscription-based scenario data stores/servers
- **Validate MSDL& C-BML scenario data**
 - Provide status/error detail on import
 - Provide tools to clean and store MSDL
- **Map to or dynamically create model representations of MSDL & C-BML content**
 - Instantiate units and entities
 - Assign orders, tasks to units
 - Build entity/unit perceptions from reports



Lessons Learned Core Capabilities – Import (2/2)

- Reference other files (orders, planning, and other data) external to MSDL file
 - Use MSDL Reference field to hold reference file name and other meta-data
 - Provide XML schema with externally referenced file
 - Associated with MSDL data using MSDL data element Unique Identifier to relate external data to MSDL Unit or Equipment item
 - Technique has been shown to work for C-BML files, unit and equipment supply data, and communications information

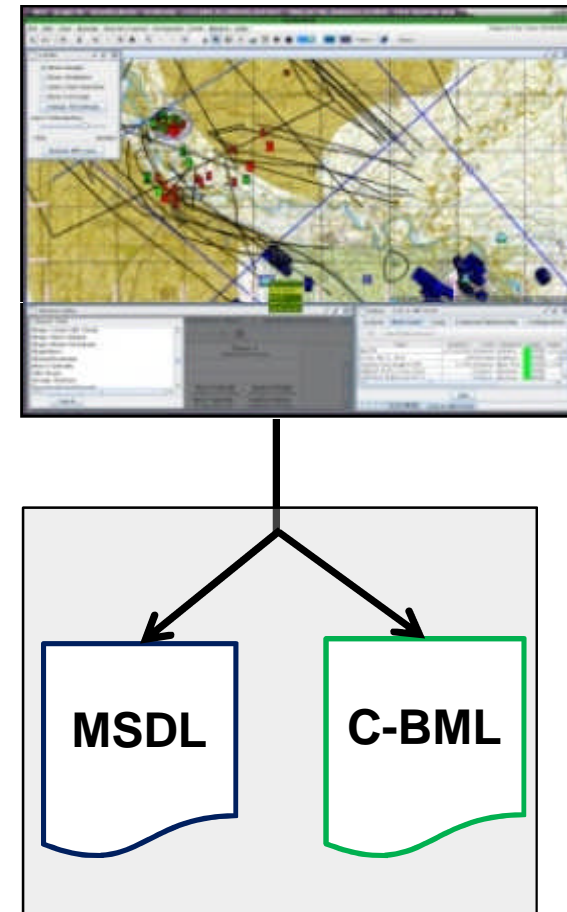


Scenario ID Element

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<msdl:MSDL_Initialize xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:sisostds:scenario:military:data:draft:msdl:1
  ../../SBML_HOME/schema/MSDL/MSDL_Initialization.xsd"
  xmlns:modelID="http://www.sisostds.org/schemas/modelID"
  xmlns:msdl="urn:sisostds:scenario:military:data:draft:msdl:1"
  xmlns:ns3="urn:int:nato:standard:mip:jc3iedm:3.1:oo:2.0">
  <ScenarioID>
    <modelID:name>msdlDemo</modelID:name>
    <modelID:type>OneSAF Export</modelID:type>
    <modelID:version>OneSAF 5.x</modelID:version>
    <modelID:modificationDate>2012-05-03-04:00</modelID:modificationDate>
    <modelID:securityClassification>Unclassified</modelID:securityClassification>
    <modelID:description>OneSAF Military Scenario Definition Language</modelID:description>
    <modelID:poc>
      <modelID:pocType>e-mail</modelID:pocType>
      <modelID:pocEmail>helpdesk@ideorlando.org</modelID:pocEmail>
    </modelID:poc>
    <modelID:reference>
      <modelID:type>C-BML</modelID:type>
      <modelID:identification>C-BML-Order-Init.xml</modelID:identification>
    </modelID:reference>
    <modelID:reference>
      <modelID:type>OneSAFComplocs</modelID:type>
      <modelID:identification>May-3-Stnd-Pass-1_complocs.xml</modelID:identification>
    </modelID:reference>
  </ScenarioID>
```

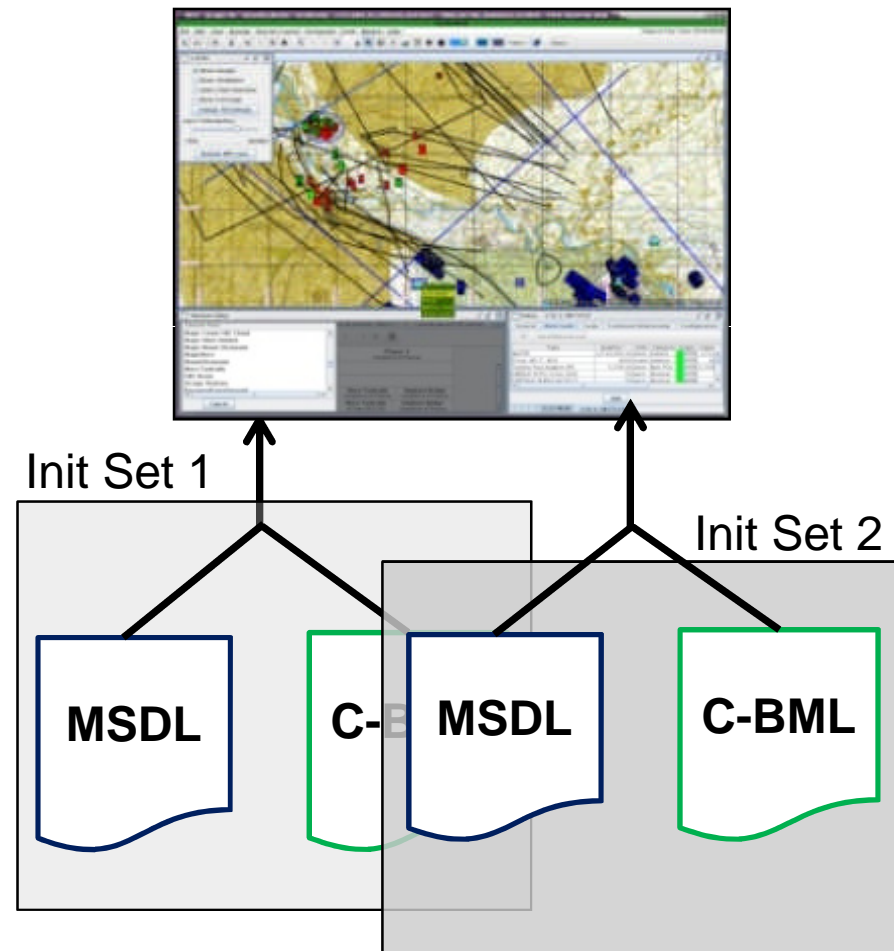
Lessons Learned Core Capabilities - Export

- **Write MSDL & C-BML formatted scenario data**
- **Validate MSDL & C-BML scenario data**
 - Provide detail on export
 - Provide tools to export valid MSDL for pre-run, run, or post run
 - Provide detail scenario report of MSDL contents
- **Write referenced file information for order and other referenced data**
 - Technique has been shown to work for C-BML files, unit and equipment supply data, and communications information



Lessons Learned Core Capabilities – Merge

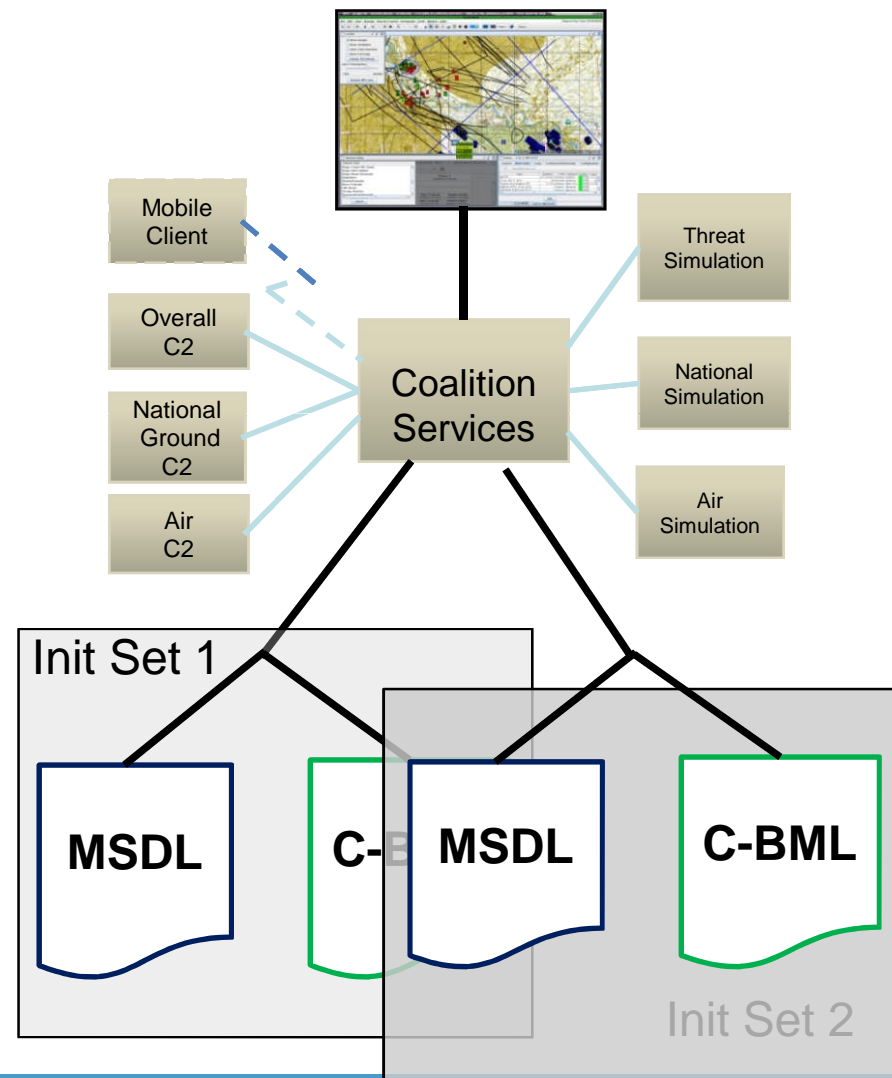
- **Merge scenario data sets**
 - Establish rules for automated merge and add as server-side capability
 - Allow choices during merge process
 - Allow configured defaults for fully automated merge
 - Report outcome of merge process with traceability to source files and Unique Identifiers
 - Support reference file updates
 - Produce valid MSDL and C-BML files and other local Reference data sets



Lessons Learned – Server Side Automation

- **Potential Server-Side Capabilities**

- Ingest and serve MSDL & C-BML independent or linked data
- Merge and serve MSDL & C-BML data
- Allow tool access to create, modify, delete, and manage MSDL & C-BML data
- Manage Coalition access to initialization and runtime MSDL and C-BML data
- Report MSDL and C-BML accessible data
- ...



Topics

- Why use C2SIM (MSDL & C-BML)
- The Crawl, Walk, Run implementation approach
- A Coalition Example
- Client Side Lessons Learned & Related Core Capabilities
- **Future C2SIM Architecture**

Future C2SIM Standards Development

- **MSDL and C-BML Integration**
 - Develop Consistent comprehensive “core” logical data model
 - Supporting process and tools to automatically generate platform specific runtime data models
 - Consider support for multiple runtime formats JSON, XML, FOM modules, etc.
 - Ballot the standard and guidelines within 3 years of Fall 2014
 - Provide compatibility with existing MSDL & C-BML standards
 - Demonstrate value in support of C2SIM STANAG
 - Continued NATO engagement and feedback to SISO C2SIM PDG
 - Formalized C2SIM SISO Standard as starting point for C2SIM STANAG