An Architecture for Web-Services Based Interest Management in Real Time Distributed Simulation

Mark Pullen and Priscilla McAndrews
George Mason University C3I Center
Katherine Morse and Ryan Brunton
SAIC San Diego
Andreas Tolk and James Muguira
Old Dominion University
Virginia Modeling, Analysis and Simulation Center

Presentation Overview

- Background: XMSF and Web Services
- Web Service Issues
- Interest Management Project
- WSIM Architecture
  - Area of Interest Management
  - Aggregation Interest Management
  - Role-Based Access Control
- Streaming Delivery Issues
- Conclusions

XMSF Motivation

- Transformational technologies are needed to scale up defense modeling/simulation to meet real-world needs
- Web technologies provide a common framework:
  - Dynamic capabilities, open standards, Web business model provide lift to support government and commercial success
  - Easy use and open extensibility for developers and users, fueling rapid growth of interoperable simulations
  - Bring defense modeling/simulation/tactical support into mainstream of enterprise-wide best-business practices

XMSF Precepts

- **Web-based technologies** can provide an extensible modeling and simulation architecture, to support a new generation of interoperable applications
- Simulation support is needed for operational warfighting capabilities
- XML-based architecture can provide a bridge between emerging rehearsal/reality/replay requirements and open/commercial Web standards
- Particularly promising for C4I-Simulation interoperation
- Web = best tech strategy + best business case

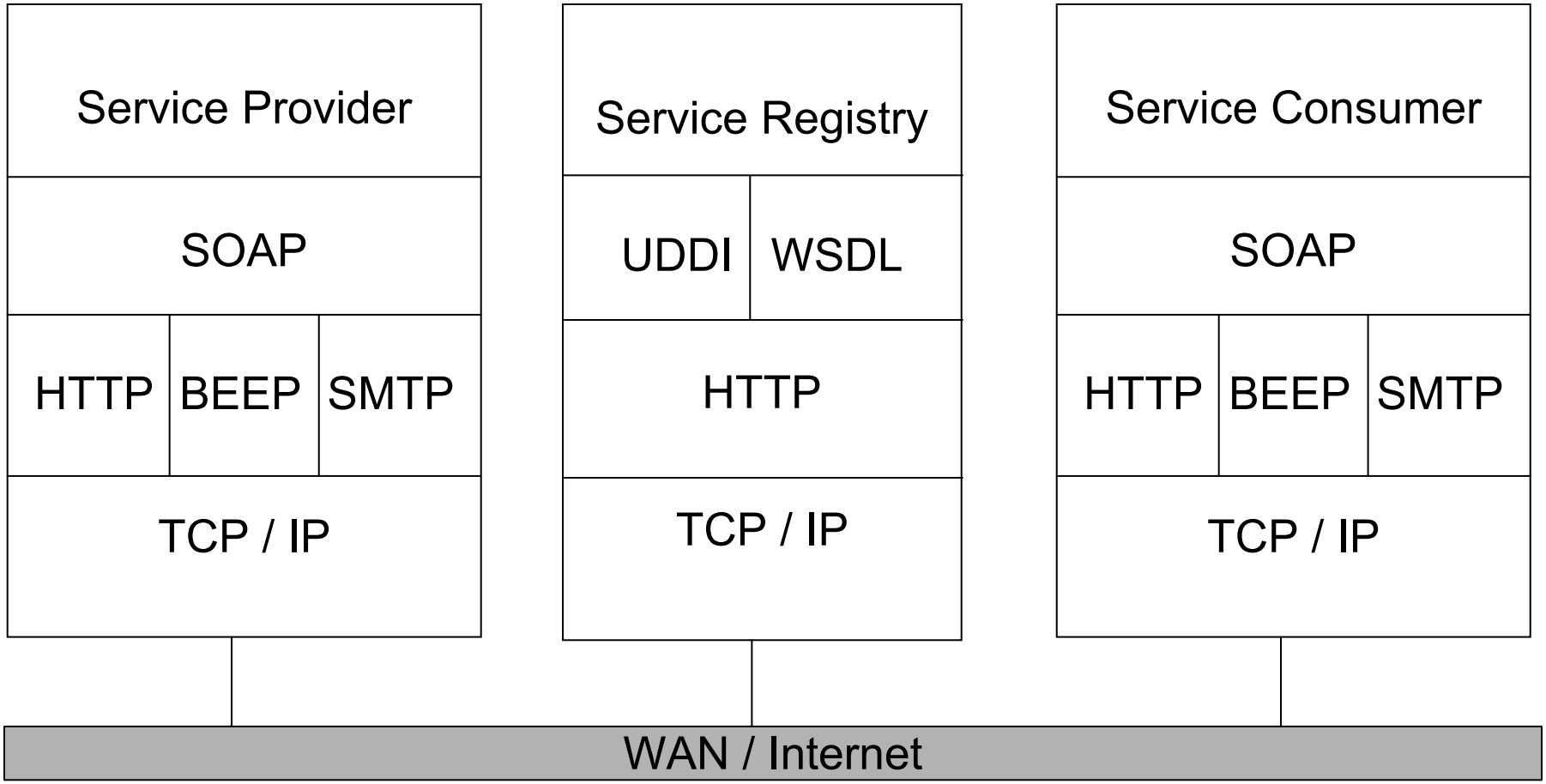
What Does XMSF "Look Like?"

- A set of profiles rather than a single architecture
  - Formal technical specifications for interoperability of Web based technologies in support of modeling and simulation
  - A profile may define a new capability or define interoperability between two or more existing capabilities
- XMSF profiles will include
  - Applicable Web technologies, protocol standards, data and metadata standards
  - A tailoring of the set of selected standards
  - Recommendations and guidelines for implementation

Web Services

Repositories Where approved services reside	**Administrative** Exemplar: DoD XML Registry
Services Discovery Publish, search capabilities	**UDDI, LDAP** Universal Description, Discovery Integration Lightweight Directory Access Protocol
Services Description Detailed methods, parameters	**WSDL, BPEL4WS** Web Services Description Language Business Process Execution Language for Web Services
XML Messaging Simple XML encoding/decoding	**XML-RPC, SOAP, XMLP** Remote Procedure Calls, XML Protocol
Service Transport Move messages between apps	**HTTP, SMTP, FTP, BEEP** Transfer is independent of messages

Web Services Protocol Stack



XC2I Viewer

- US JFCOM experimentation environment
  - Complex and rich
  - Basically a very large LAN-based NVE
  - 100k objects in hybrid HLA/DIS system
- Desirable to extend access over WAN
  - View a subspace
  - Control object behavior

WSIM Motivation
XC2I Potential Information Flow Estimate

- Viewers
  - Each potentially has 10000 objects viewable
  - 100 different simultaneous views maximum
  - Viewers may or may not overlap
  - A viewer that zooms out uses aggregation service such that there are no more updates per second from the service than when zoomed in
- Federates
  - 250 processors
  - 5000 objects per processor
  - Average update period 2.5 seconds
- Worst-case aggregate flow:
  400 K updates/s (~100 bytes each)
  40 MBytes/s = 320 Mb/s => not feasible on WAN
  (sensitive to the viewable objects and max views)

Ways to Reduce Network Impact of Viewer

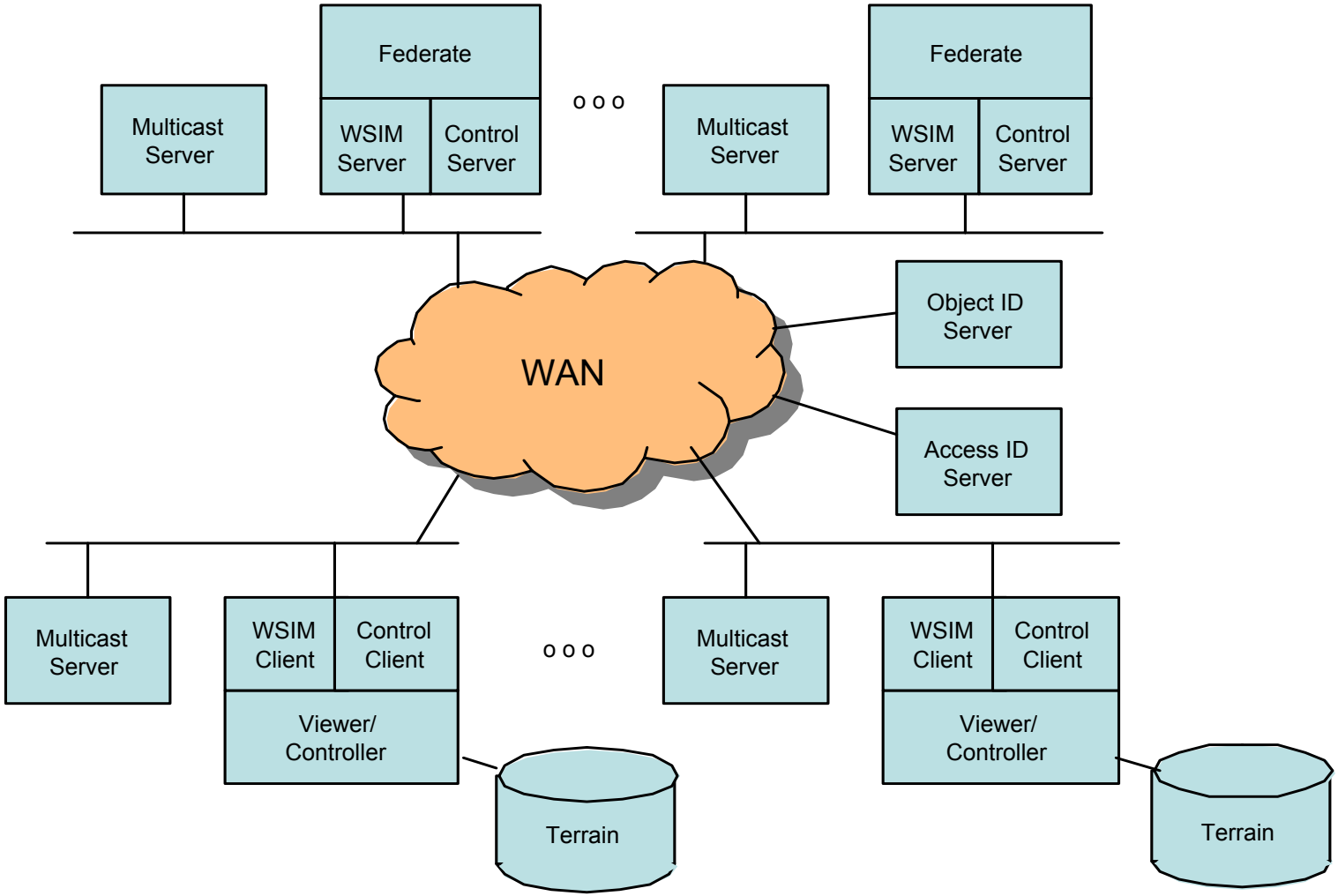
- Limit scope in geographic and other dimensions
- Aggregate objects at server
- Don’t transmit movements too fine to be seen
- Decrease the viewer refresh rate to preclude network overload
  - statically as startup parameter
  - or dynamically as necessary during execution
- Use streaming multicast for high-volume flows

WSIM Overview

The user subscribes to types of entities in a geographic region using a GUI

- Make the process as easy and visual as possible
  - Point & click
  - Drag & drop
- Insulate the user from the details of the Interest Management (IM) protocol and underlying, native IM mechanisms
  - Mapping is handled at layers beneath the viewer
- A user can only subscribe to entities in the current viewbox
  - If an entity of interest moves out of the viewbox (“out of scope”), its updates won’t be delivered again until it’s back in scope, but the subscription will remain in effect
  - This is enforced by the viewer, not by the IM protocol

Top-Level Architecture With WSIM



WSIM Functions
Area of Interest Management

The IM protocol is focused on C2 viewers

- Not as general as HLA DDM because it explicitly includes geographic location and entity type
- But broader than JFCOM JUO
  - Tailoring is handled in one of the mapping layers
- The same protocol can be used with other federations by changing only the bottom mapping layer

WS Interest Management Layers

Generic	JFCOM J9 JUO
User's IM language/interface	GUI: user selects lat/lon/alt, entity (or aggregation) type
Mapping between user's IM & IM protocol	Mapping between GUI inputs & IM protocol
IM protocol	XML Interest Expressions (some tags derived from C2IEDM)
Mapping between IM protocol and native IM	Mapping between IM protocol, and JSAF POB & RTI regions
Native IM	JSAF POB & RTI regions

Not necessarily a one-to-one mapping to physical architecture components

WSIM Functions
Aggregation Interest Management

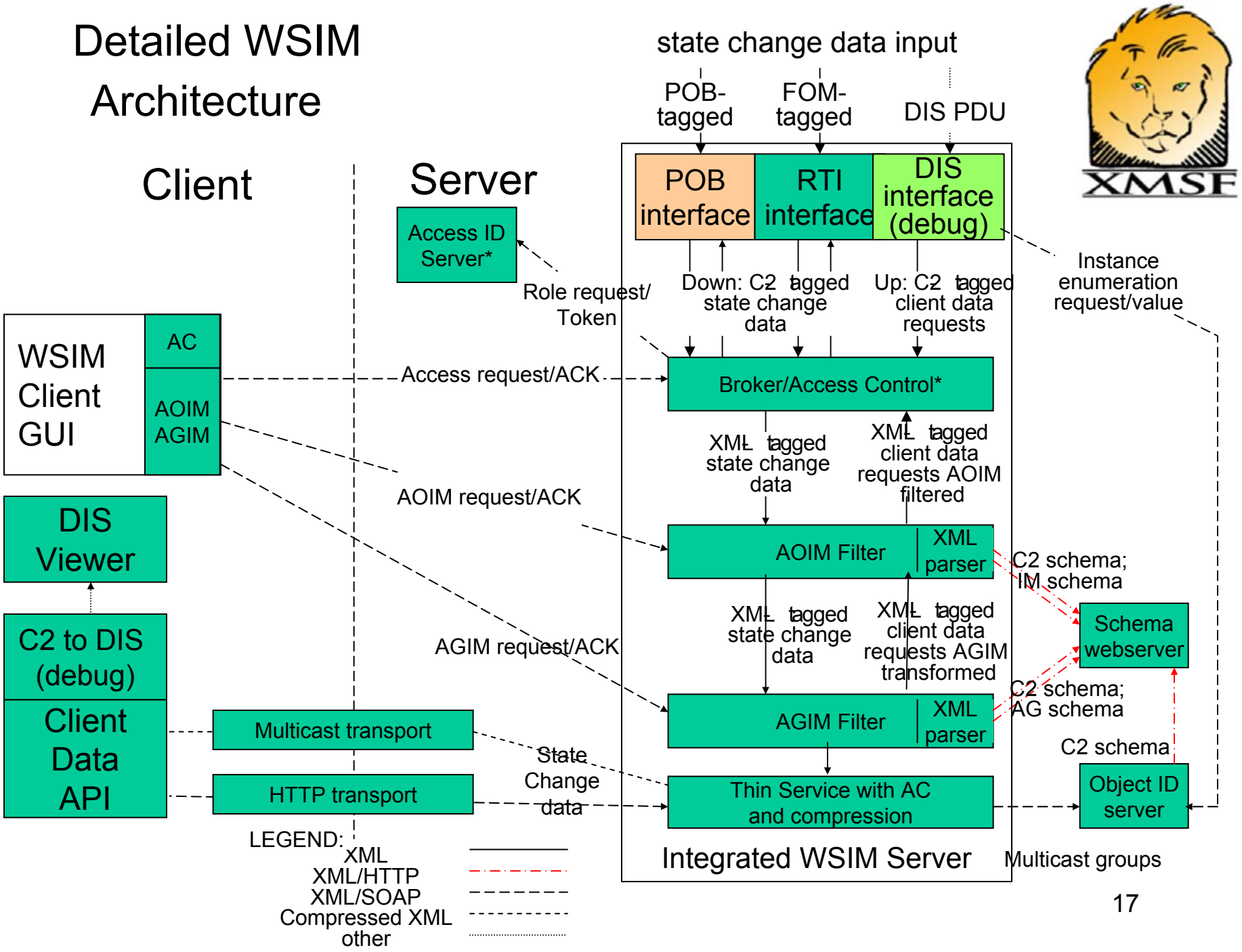
Aggregation:

- "the ability to group entities while preserving the effects of entity behavior and interaction"
- type driven: based on the organization types of the simulated platform and entities established in the military order of battle
- instance driven: based on specific user requirements, *e.g.* "show these three objects as one icon"
- Instance driven overrides type driven
- Applicable to generic military command & control

WSIM Functions
Role-Based Access Control

- Participants have defined access rights
- Don’t send viewer data that will not be displayed
- Combined with latest Web security
  - Globally unique signed certificate
  - Distributed identity management, *e.g.* LDAP
  - GUI for role selection (within user’s prescribed rights)

Detailed WSIM Architecture



Web Service Overhead (~3000%)

Pure Web service

Connect	- 136 Bytes
HTTP Request Seg 1	- 1500 Bytes
Client Ack 1	- 40 Bytes
HTTP Request Seg 2	- 120 Bytes
Client Ack 2	- 40 Bytes
HTTP Response Seg 1	- 833 Bytes
HTTP Response Seg 2	- 40 Bytes
Client Ack for seg 1	- 40 Bytes
Client Ack for seg 2	- 40 Bytes
Response 1	- 40 Bytes
Ack 1	- 48 Bytes
Response 2	- 48 Bytes
Ack 2	- 40 Bytes

Total Per Computation : 2829 Bytes

Grand Total = 136 +350 * 2829

= 990286 Bytes

Web service plus multicast

Connect	- 136 Bytes
HTTP Request Seg 1	- 1500 Bytes
Client Ack 1	- 40 Bytes
HTTP Request Seg 2	- 175 Bytes
Client Ack 2	- 40 Bytes
HTTP Response Seg 1	- 835 Bytes
HTTP Response Seg 2	- 40 Bytes
Client Ack for seg 1	- 40 Bytes
Client Ack for seg 2	- 40 Bytes
Response	- 40 Bytes

Total for setup: 2886 Bytes

Multicast Packet Size average - 88 Bytes

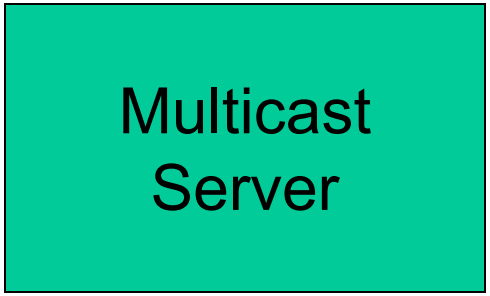
Grand Total = 2886 + 350 * 88

= 33686 Bytes

Web Services for XC2I

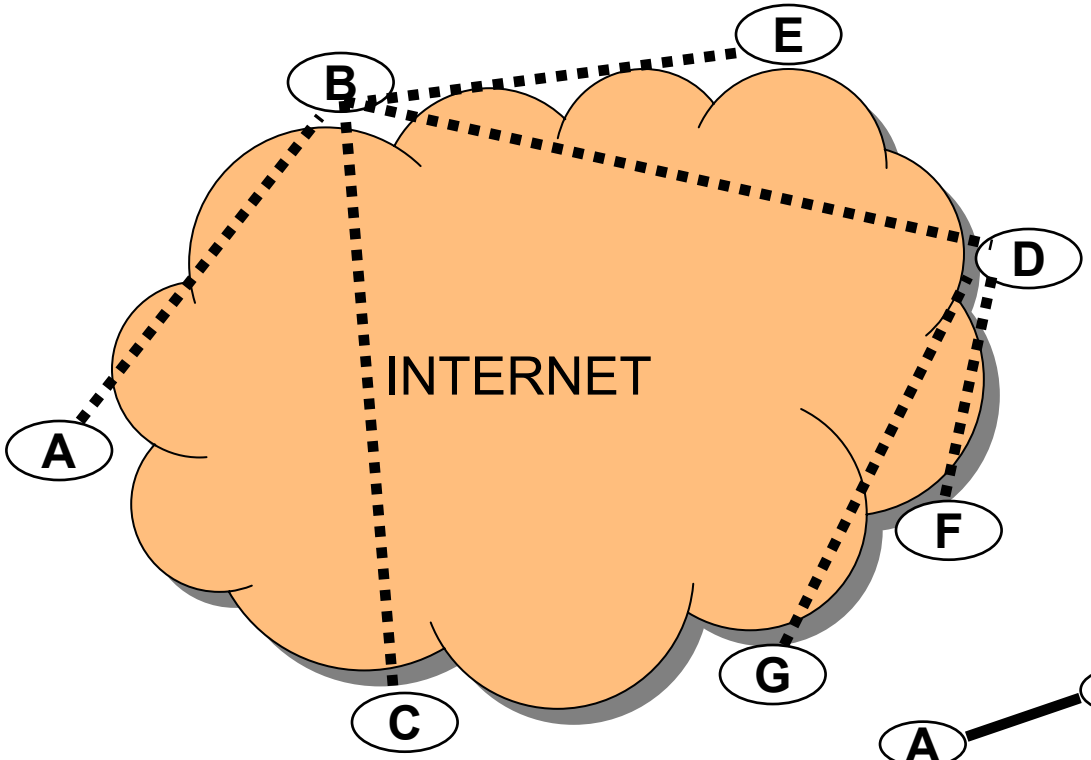
- Pro:
  - Easy to create
  - Easy to interface
  - Easy to compose
  - Use everywhere data volume is low
- Con:
  - Significant overhead
  - Don’t use for massive data flows

Multicast Server (XOM)

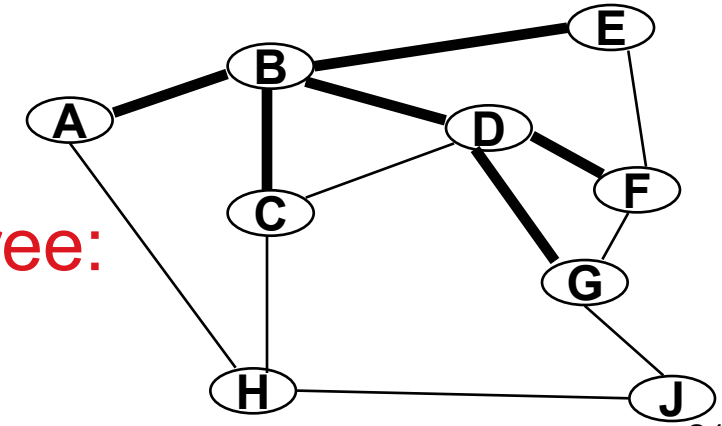


- Provides multicasting service over WAN
  - Couples to IP multicast on LAN
  - Minimizes traffic using optimal transfer tree
- See companion paper

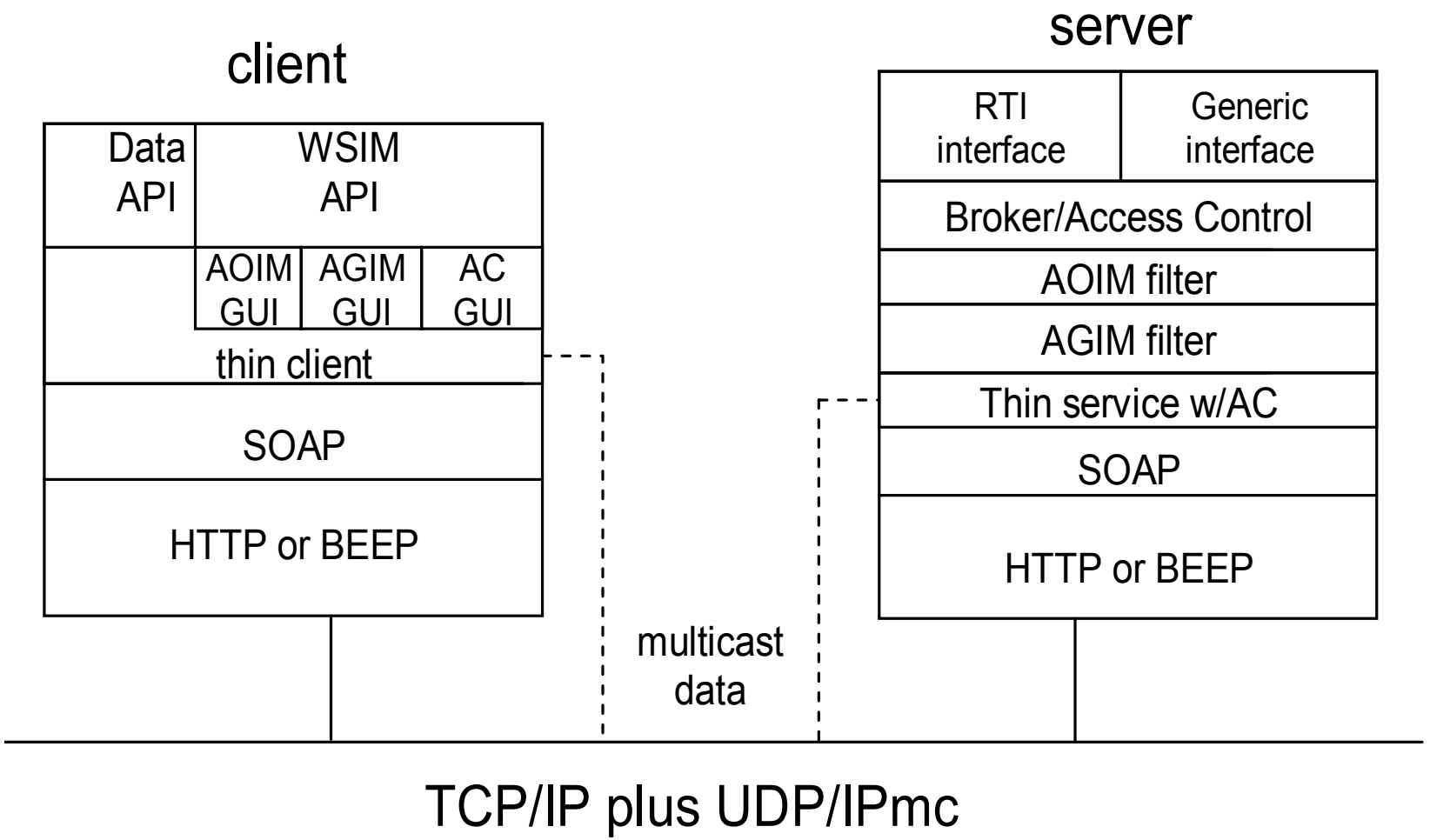
Overlay Multicast Tree



IP Multicast tree:



WSIM with Streaming Multicast



Conclusions

- The XMSF approach is showing great promise as a basis for distributed software interoperation
  - In particular, distributed simulation
- We have developed an architecture for generic Web-service interest management
  - Prototype now running
- However, we have found some limitations in the approach
  - Web services need to be extended for high performance systems
  - We used streaming multicast for the extension