



Network Based Hard/Soft Information Fusion

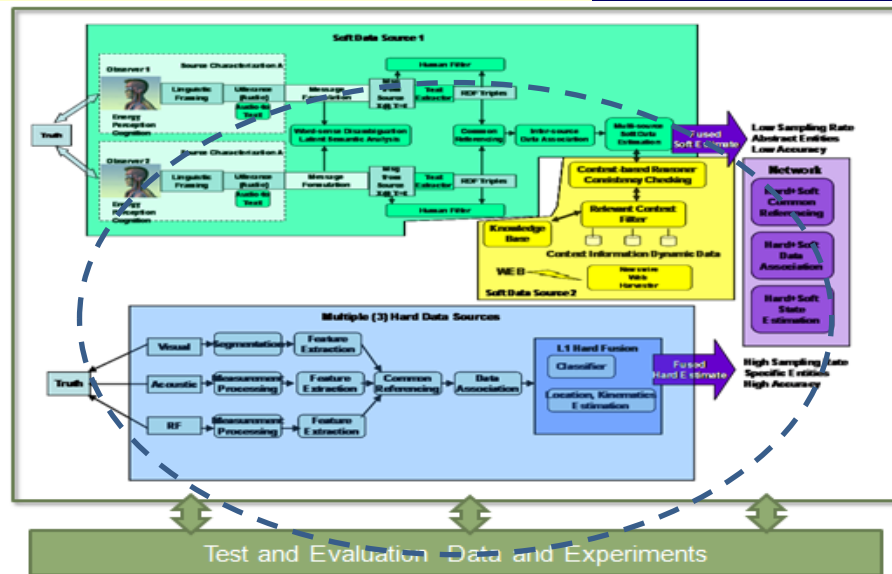
Network Architecture/SOA

J. Rimland



Objectives:

- Develop, demonstrate and evaluate an information architecture, cyber-infrastructure, data standards and tools to allow integration and transition of distributed hard and soft data fusion



Scientific/Technical Approach

- Investigate, leverage, and extend service oriented architecture, message oriented middleware, multi-agent systems, complex event processing, and community standard data representations
- Develop an understanding and approach to optimize human versus machine functions for observations, sense-making, situation awareness and collaboration

Accomplishments

- Designed and implemented an integration and transition environment
- Extended SOA and stream-based computing to a distributed hard/soft processing environment
- Developed framework for iteratively adjudicating and optimizing human vs. machine roles in observation, sense-making and situation awareness

Challenges

- Rapid evolution of SOA and related environments
- Heterogeneous observing and computing environment



Main Scientific/Technical Accomplishments

Network Architecture: SOA



Motivation

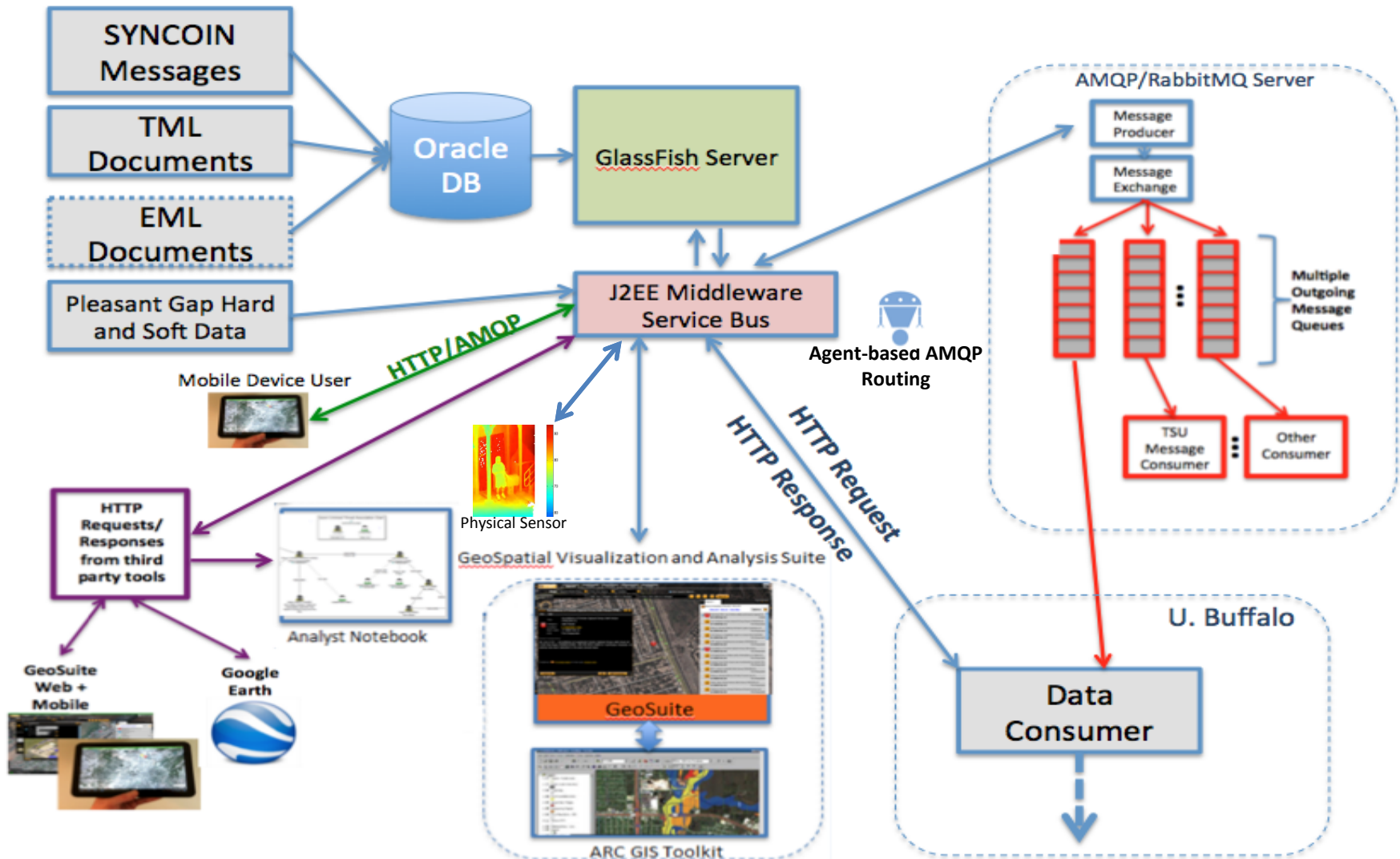
- Human-centric distributed hard and soft fusion fundamentally requires linking distributed processes (including physical sensor observations, human reports, hard and soft fusion and collaborative analysis) over a distributed network
- The heterogeneous observing , communication and computing environment involves combinations of COTS, GOTS software, data and legacy tools

Summary of Accomplishments

- Conducted literature review of related areas including Service Oriented Architecture, Message Oriented Middleware, Multi-Agent Systems, and Complex Event Processing
- Conducted assessment of open standards for sensor and distributed information encoding
- Designed and implemented an integration and transition environment
- Extended SOA and stream-based computing to a distributed hard/soft processing environment
- Developed framework for adjudicating and optimizing human vs. machine roles in observation, sense-making and situation awareness
- Implemented message connectivity (both “pull” and “push”) between University at Buffalo, the Pennsylvania State University and the Tennessee State University



ARL PSU Architecture Accomplishments





Infrastructure Contributions

Toolbox of techniques



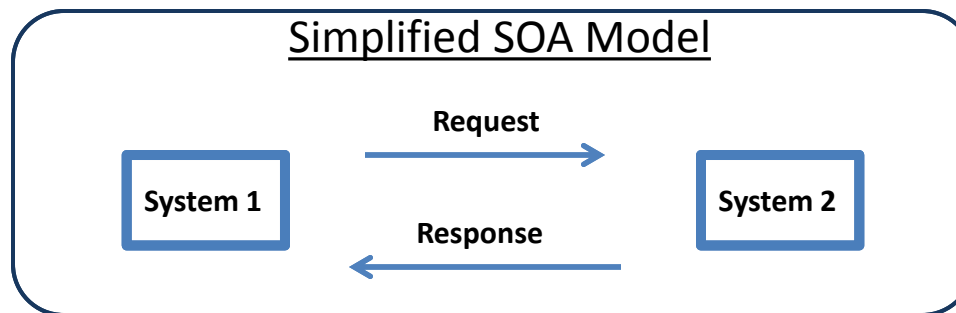
- 1) Portfolio of communication methods and protocols for various conditions and network configurations**
 - HTTP Request/Response, JMS, AMQP
 - Allows integration with 3rd party tools (e.g. GeoSuite)
- 2) Human-centric extensions to the service-oriented architecture (SOA) and Message Oriented Middleware (MOM) paradigms.**
 - Extending “loose coupling” to include human completion of certain sensing/cognition tasks.
- 3) Multi-agent Systems (MAS) approach to optimization of information flow and tasking.**
 - Further enables dynamic system operation/reconfiguration by encapsulating tasks into autonomous software services.
- 4) Application of Complex Event Processing (CEP)**
 - Transforms multiple streams of distributed heterogeneous data into logical hierarchies based on the application of local rules at multiple levels of abstraction.
- 5) Community Standard Data representation and exchange protocols**
 - EML, TML, SOS, SES, etc.



Key Accomplishment: Extending the SOA Paradigm

- Introduced in 1996
- Main tenets of SOA are:

- 1) Key functionalities are delivered as a service
- 2) Services are loosely coupled



* Actually coined in 1994 by Alexander Pasik in a class discussion. Later published by Garner analysts Roy Schulte and Yefin Natis in a 1996 paper [1].



Loosely Coupled Services

Loosely Coupled:

1) Resource Addressing

-For Scalability, Flexibility, Reliability

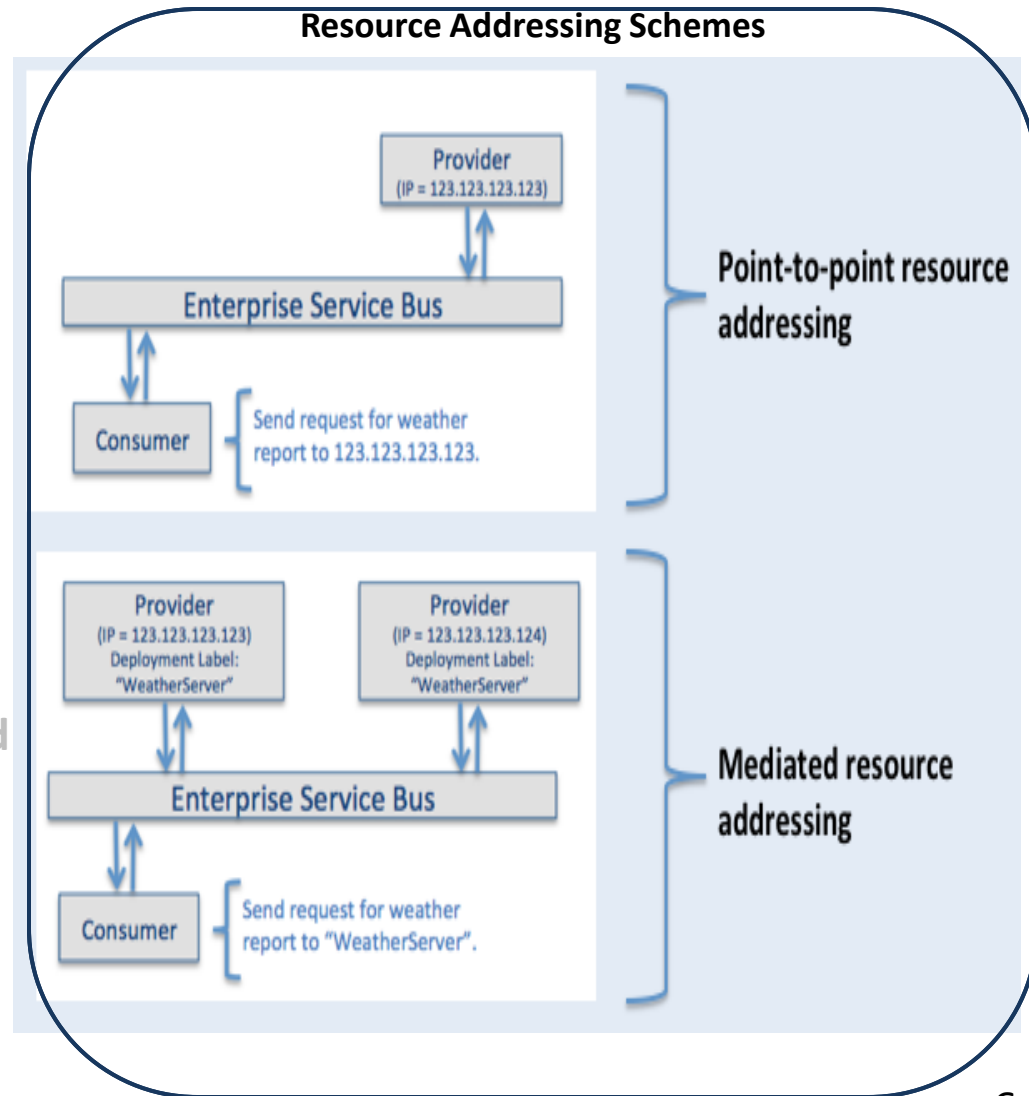
2) Communication Style

- Synchronous vs. Asynchronous

3) Composed Services

-Perform complex tasks not anticipated at design-time.

4) Human Contribution





Loosely Coupled Services

Loosely Coupled:

1) Resource Addressing

-For Scalability, Flexibility, Reliability

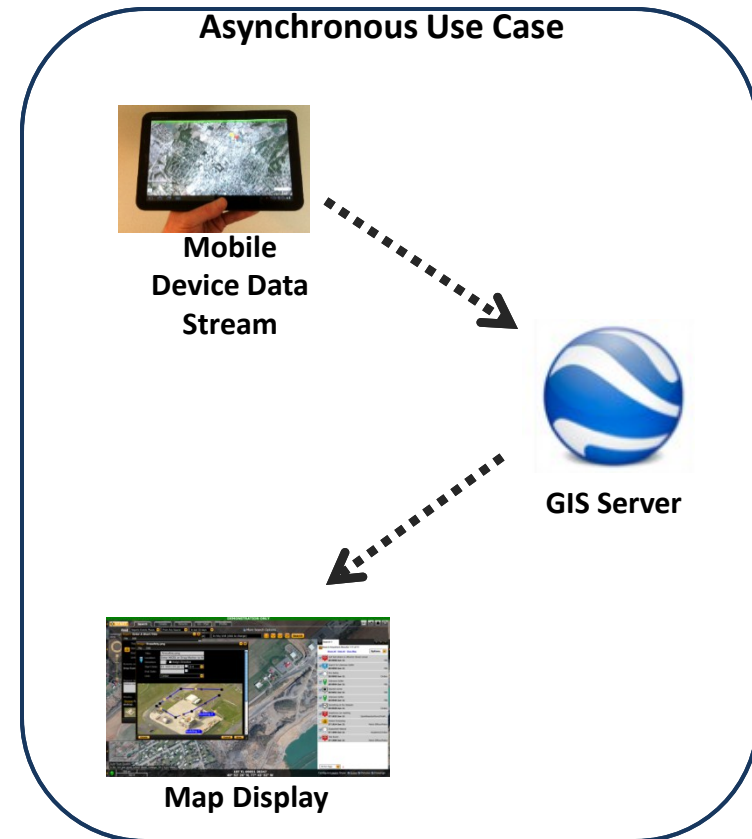
2) Communication Style

- Synchronous vs. Asynchronous

3) Composed Services

-Perform complex tasks not anticipated at design-time.

4) Human Contribution





Loosely Coupled Services



Loosely Coupled:

1) Resource Addressing

-For Scalability, Flexibility, Reliability

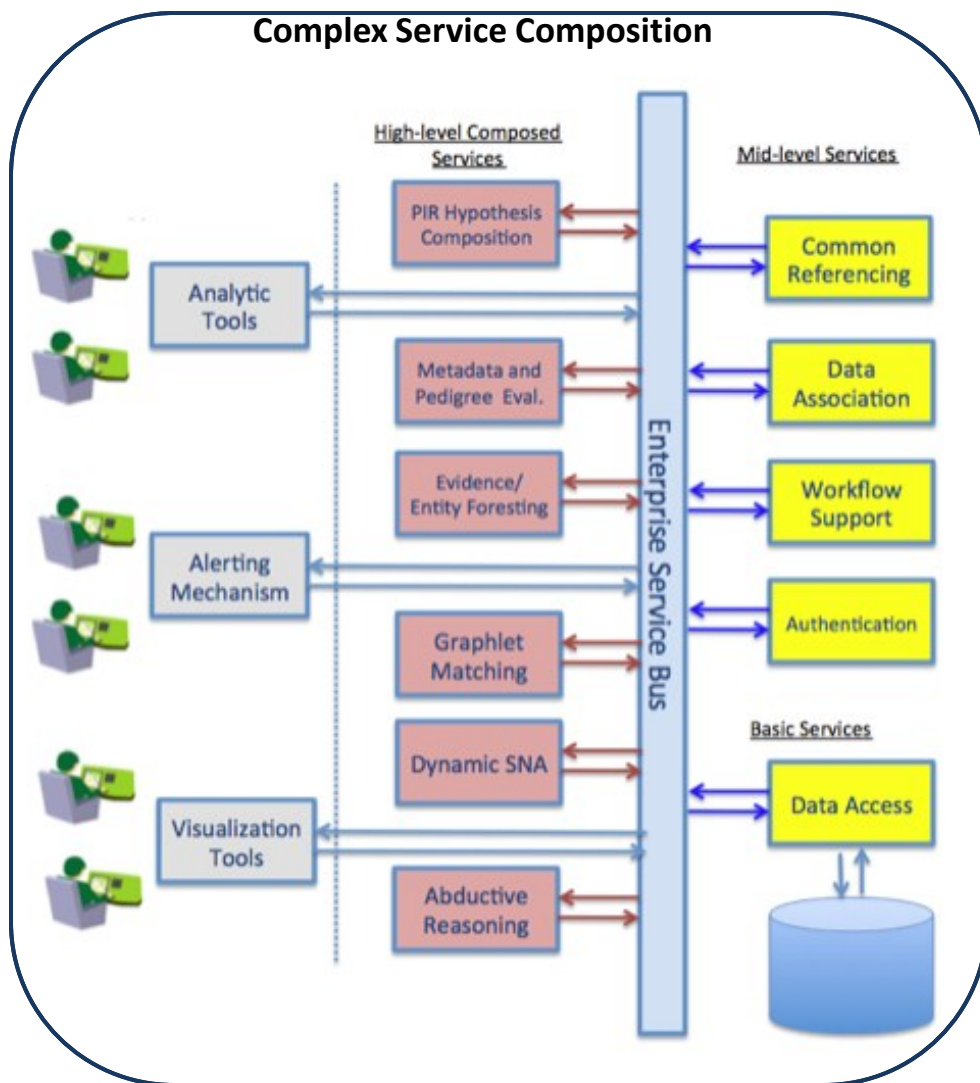
2) Communication Style

- Synchronous vs. Asynchronous

3) Composed Services

-Perform complex tasks not anticipated at design-time.

4) Human Contribution

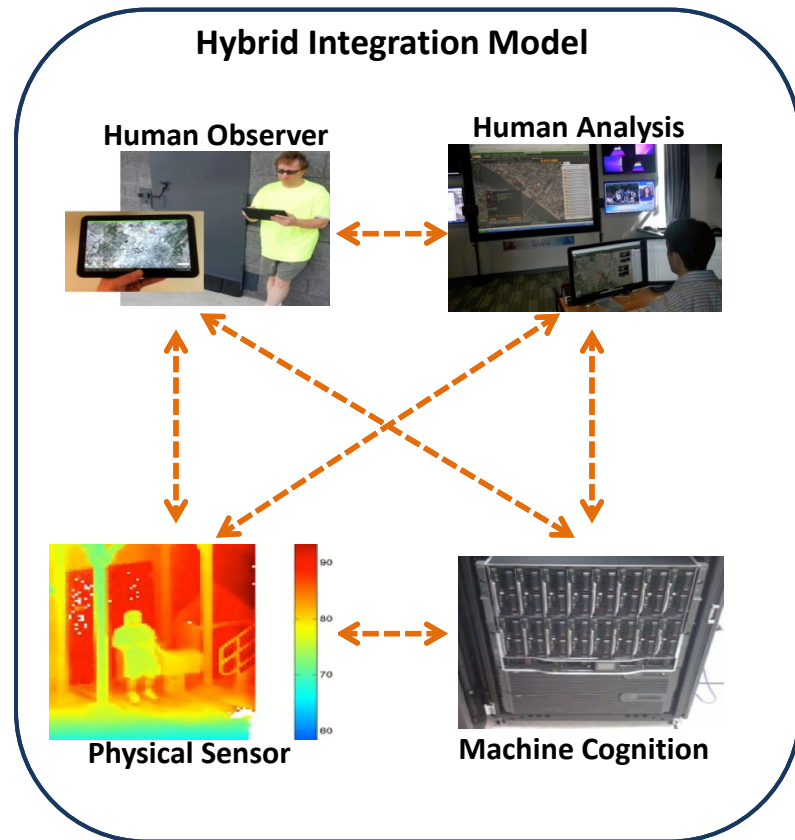




Key Accomplishment: Extending SOA for Human Contribution

Loosely Coupled:

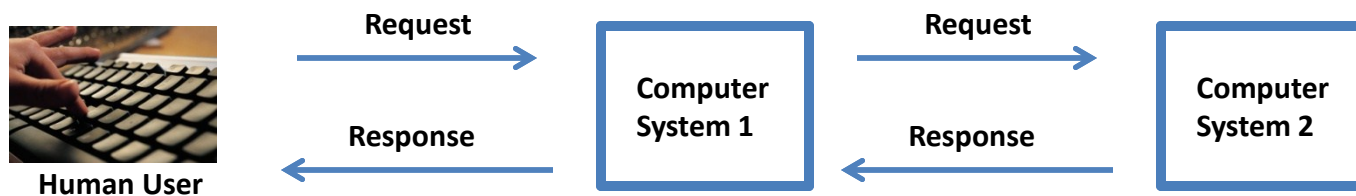
- 1) Resource Addressing**
-For Scalability, Flexibility, Reliability
- 2) Communication Style**
- Synchronous vs. Asynchronous
- 3) Composed Services**
-Perform complex tasks not anticipated at design-time.
- 4) Human Contribution**
-Enabled via novel connectivity approach



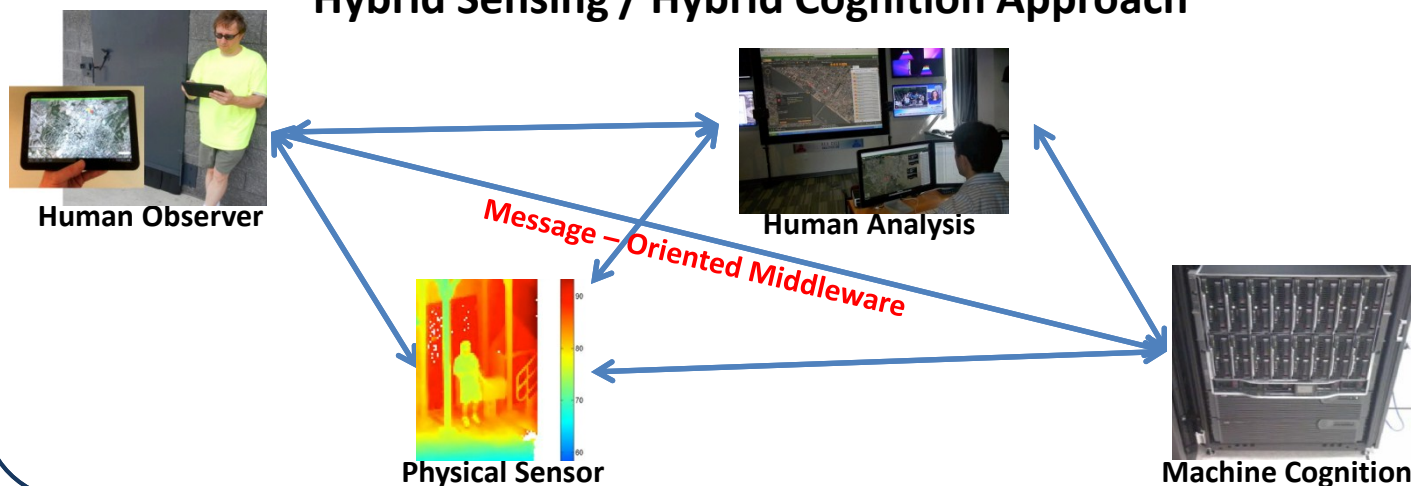


Extending SOA

Conventional Approach



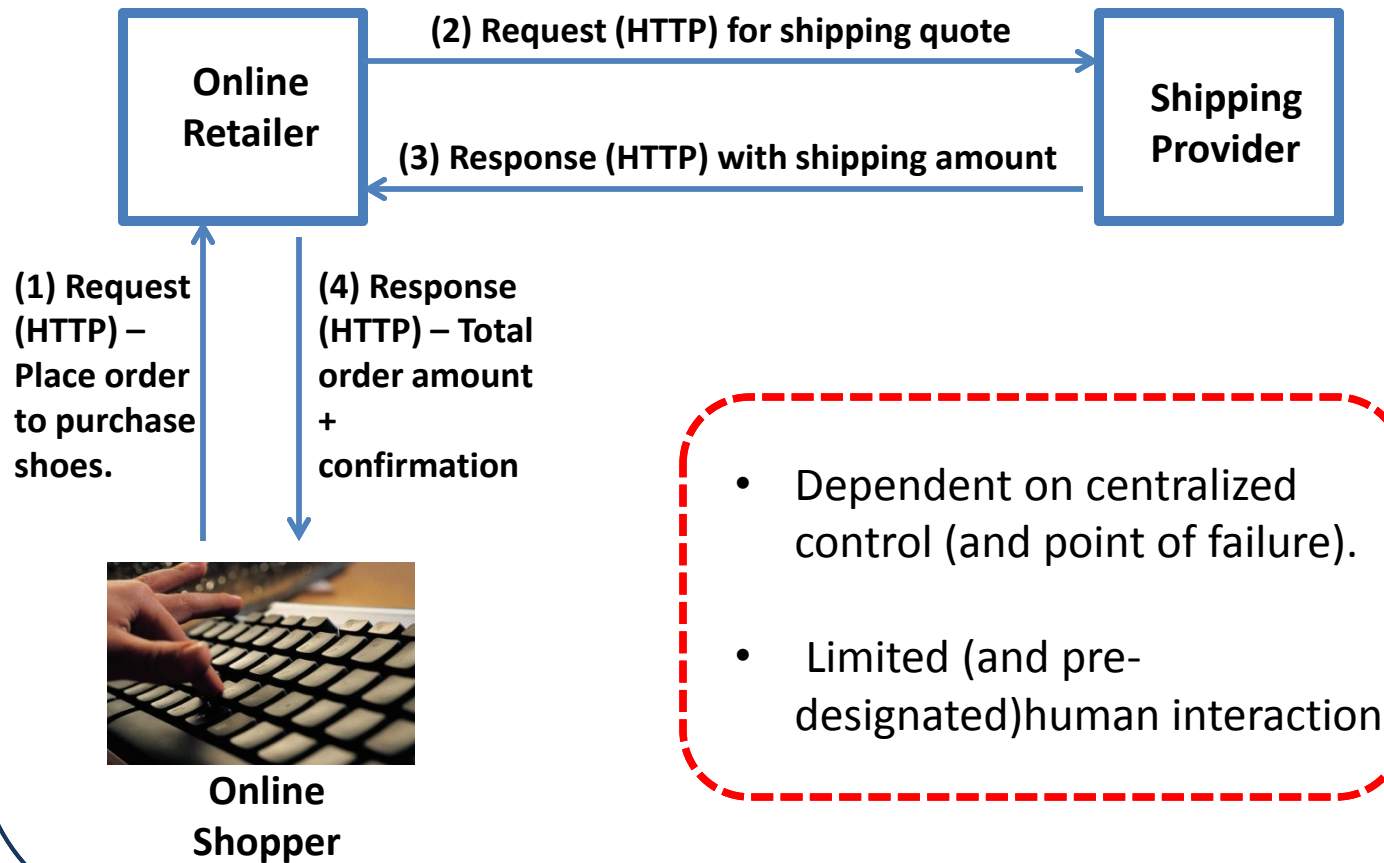
Hybrid Sensing / Hybrid Cognition Approach





Example SOA: Conventional

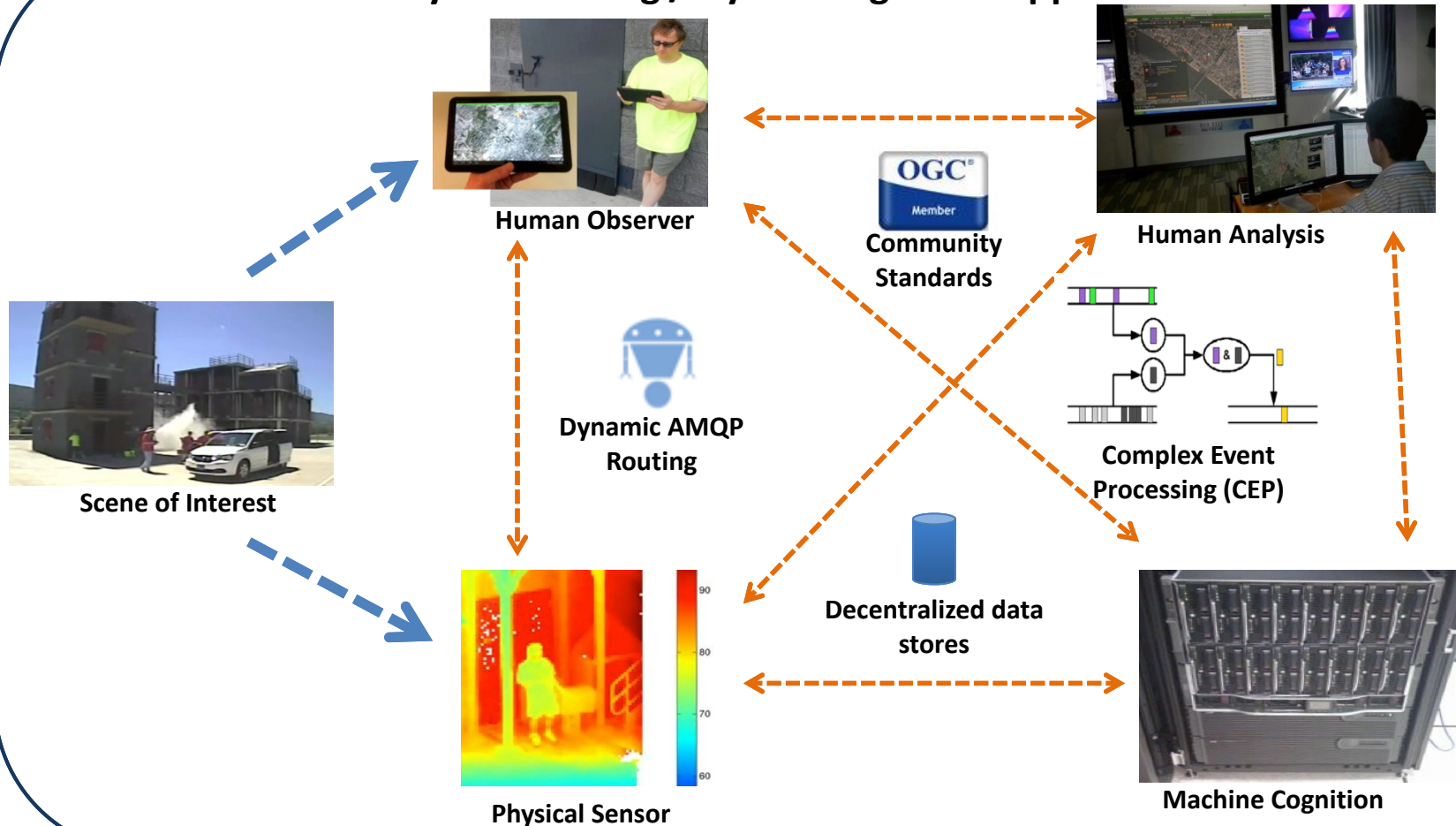
Conventional Approach





Key Accomplishment: Hybrid Sensing/Hybrid Cognition SOA

Hybrid Sensing / Hybrid Cognition Approach





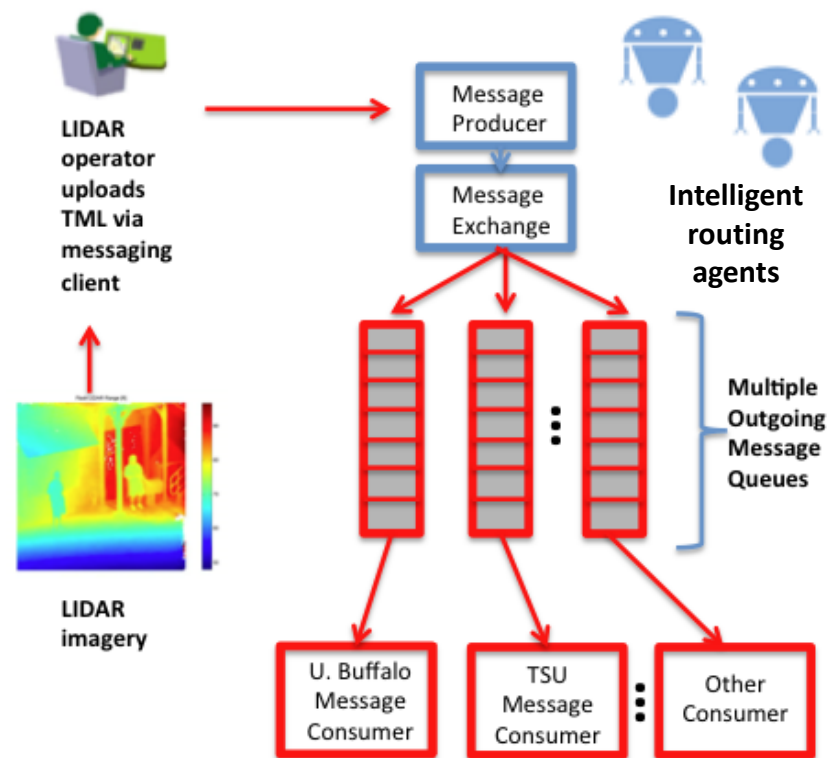
Key Accomplishment:

Dynamic AMQP Routing of Data



Misconception: All data should exist at each processing node in a distributed architecture

- Data should be routed based on probability of useful contribution, transmission cost, and security aspects.
- Agent-based, ontology-driven approach can optimize dynamic routing of data – so data is only transmitted where necessary.
- Advanced Message Queuing Protocol (AMQP) is a flexible, high-performance messaging middleware that can enable dynamic routing, tasking, and adjudication.





Key Accomplishment: Integration with Community Standards



- TML has provided the ideal protocol for transmitting feature-level sensor data with metadata.
- Community Standards vs. “Harmonization” (e.g. EmpLastName == EmployeeLastName?)
- Transition-compatible via GeoSuite, FEF, etc.
- Ideal for transfer over SOA/ESB
- Demonstrated utility for hybrid simulated/actual sensor data
- Future Transition: Event Pattern Markup Language (EML)

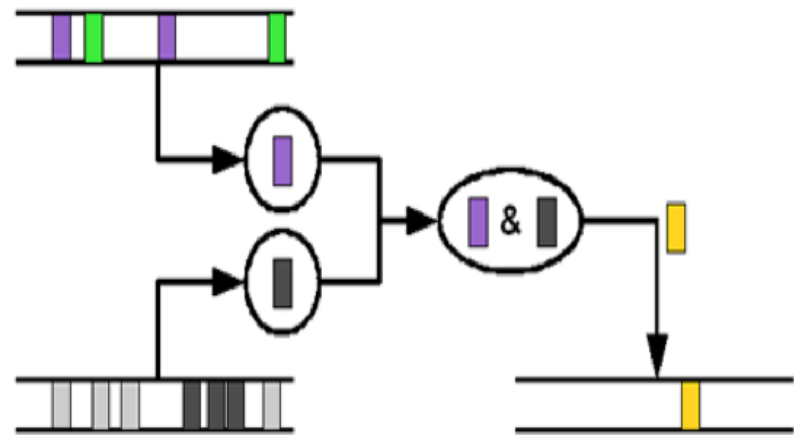
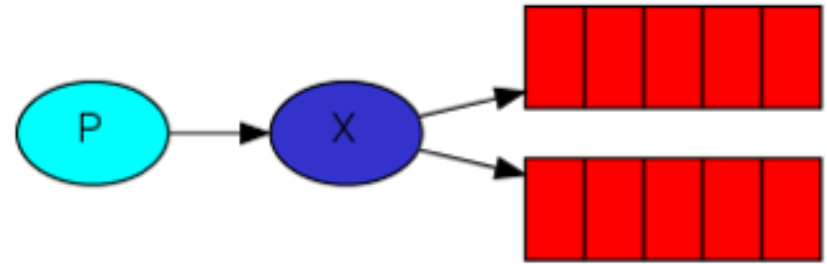


```
<tml:handleAsType>  
  double  
</tml:handleAsType>  
</tml:dataUnitEncoding>  
<tml:dataUnitEncoding name="MWIR_IdentityClass">  
<tml:handleAsType>
```



Key Accomplishment: Integration with SteamBase CEP

- Loose coupling via Message-Oriented Middleware (MOM) has proven critical for Complex Event Processing (CEP) over multiple streams of data.
- Event Pattern Markup Language (EML) facilitates “semantic snapshots” of complex event hierarchies that are flexible, portable, and query-able.
- Causality and other relationships can be determined via local rules applied at multiple levels to multiple streams.





Option Year Plans



Integration, transition and network-based processing



PENNSYLVANIA STATE UNIVERSITY



- **Capability Goal:**

- Provide capability for human in the loop experiments at all levels including; data collection (knowledge elicitation), human annotation of hard sensor data, human-on-the-loop analysis, and analyst/observer collaboration
- Provide end-to-end integration of SOA, messaging, multi-agent software tools, and complex event processing for each sensor and processing node across the universities.

- **Research Goals:**

- Conduct experiments with human-centric (human in and on the loop) information fusion
 - Demonstrate utility of “focus of attention” techniques
- Demonstrate effectiveness of smart data routing in distributed H/S fusion



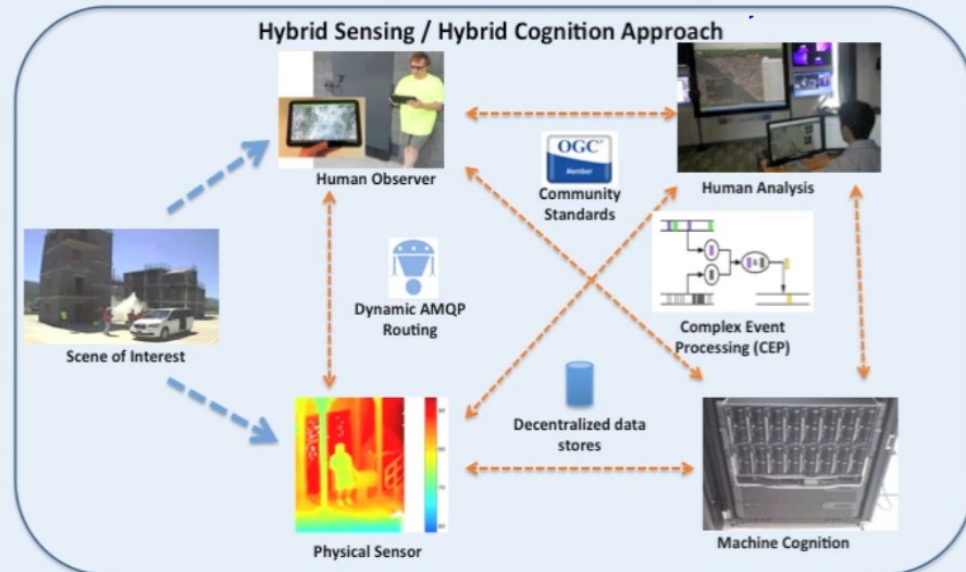
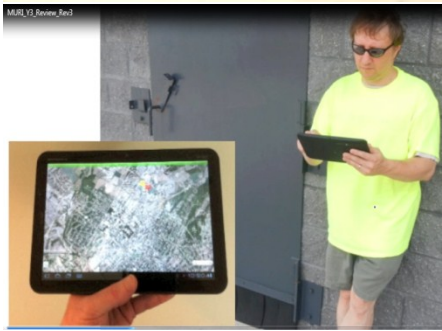
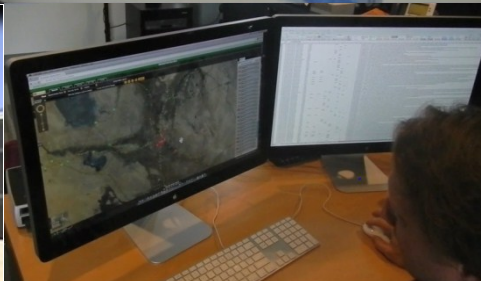
Demonstrations



Penn State University
Year 3 Research Highlights

Video Demonstration

- Link real-time data collection & processing to remote analysis site
- Allow integration with heterogeneous distributed data
- Support human-in-the loop experimentation/integration
- Extend/integrate existing tools and technologies

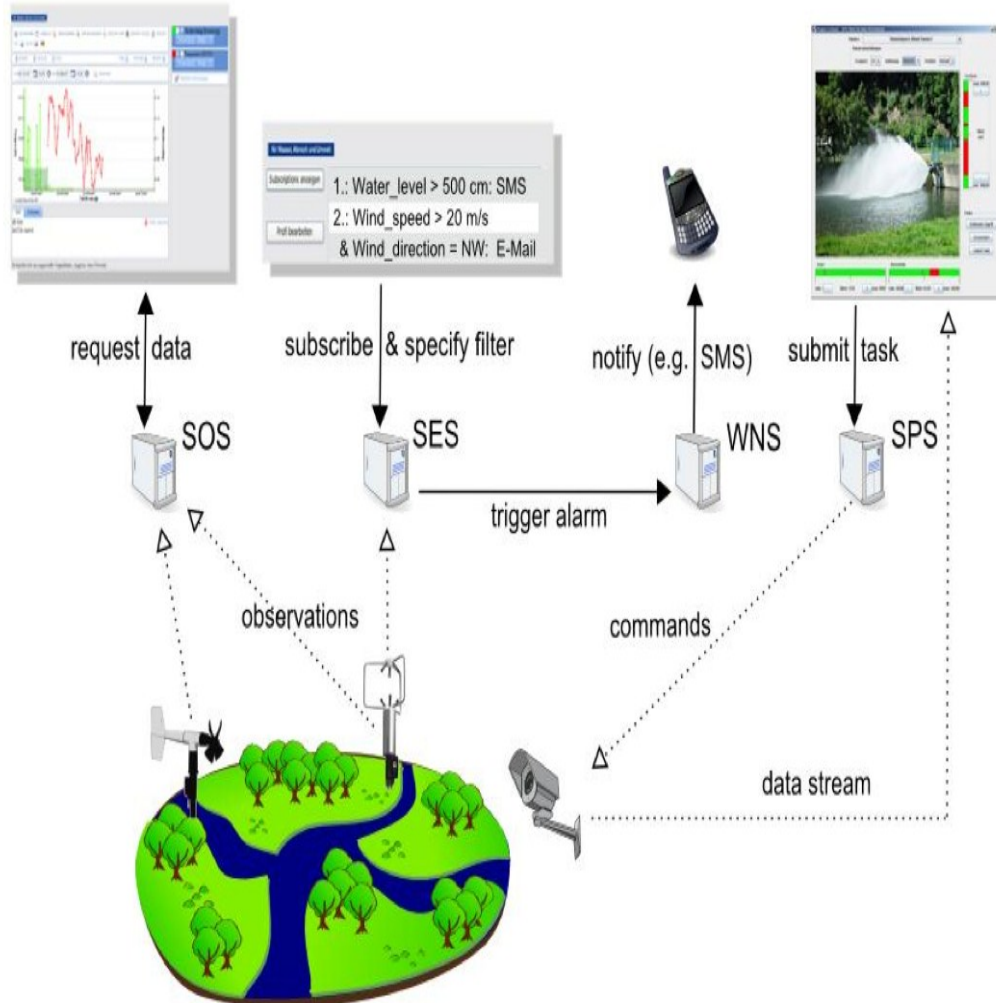




End of Presentation – Reference Slides Follow



Community Standards for Sensor Web Enablement



SOS (Sensor Observation Service):

Introduces observations into the system (typically encoded as TML, O&M, EML, etc.)

SES (Sensor Event Service):

Allows client to subscribe to filtered data stream of sensor events

WNS (Web Notification Service):

Enables asynchronous messaging between subscribers/publishers

SPS (Sensor Planning Service):

High-level coordination of task coordination (e.g. task feasibility, system status, etc.)



Architecture Lessons Learned

Interoperability is Key

Three communication methods explored were:

- 1) HTTP Request/Response
- 2) Java Message Service (JMS)
- 3) Advanced Message Queuing Protocol (AMQP)

- **HTTP Request/Response**

Good: No vendor-lock in, Network/Firewall friendly, Many supporting tools (WS-*)

Bad: More difficult to implement PUSH and other Pub/Sub options

- **Java Message Service (JMS)**

Good: Implements variety of Pub/Sub options (queues, topics, etc.)

Bad: Requires Java on both ends, can require network config, 3rd Party tool compatibility.

- **Advanced Message Queuing Protocol (AMQP)**

Good: Same Pub/Sub options as JMS, server config. In code, no Java req., “just works”

Bad: Not compatible with 3rd party tools using HTTP Request / Response



Agent-based Human/Computer Tasking / Adjudication

Hybrid Agent Roles:

- Sensing (Human observer , Hard Sensor / Annotated Soft)
- Cognition / Analysis (Preprocessing / Processing, Data Association and Fusion)
- Translation / Data Representation
- Tasking
- Adjudication
- Dynamic Interface

