Integrating Subjective Trust into Networked Infrastructures

Mark D. Heileman, Ph.D., Modus Operandi, Inc.

Gregory L. Heileman, Ph.D., AHS Engineering Services

Jong S. Hwang, Air Force Research Laboratory

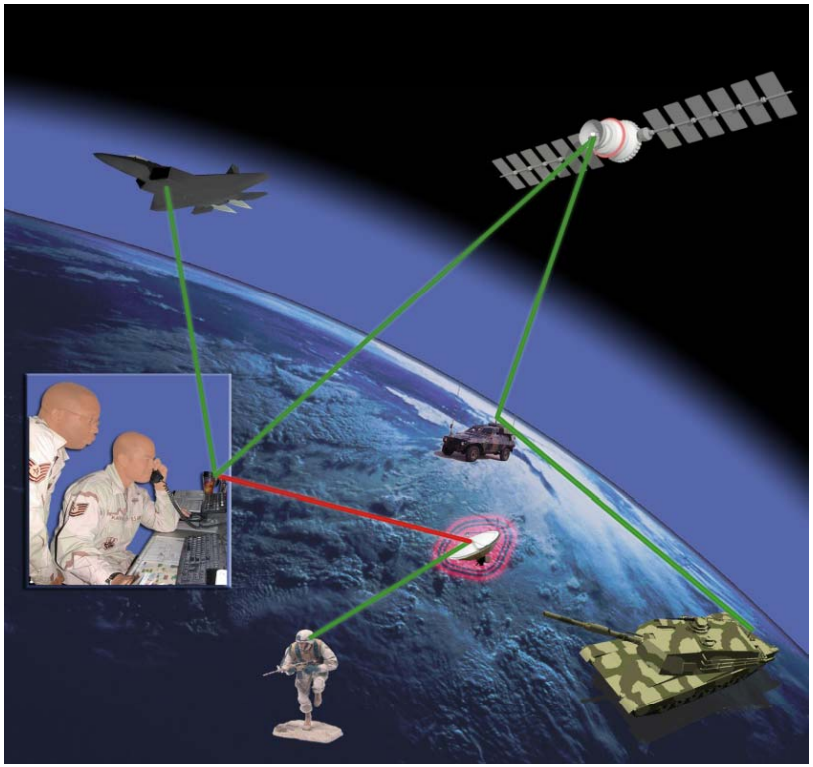
Prepared for SSTC 2009 – 22 April 2009

Acknowledgements

- The information presented is the result of work on a Small Business Innovation Research (SBIR) Phase I project.
- This work was funded in whole or in part by Department of the Air Force Contract FA8650-08-M-1441.
- The Air Force Research Laboratory's Distributed Collaborative Sensor System Technology Branch (AFRL/RYTC) personnel provided guidance and support for the project.

Presentation Agenda

- What we were trying to solve (Objectives).
- What we did (Progress).
- What the results were (Requirements).
- How we did it (Demonstrations).
- What we learned from the project (Recommendations).
- Questions and comments (Discussion).



OSD08-IA4: Assuring Trust between the Edges

Phase I Vision: Investigate and propose an architecture to determine/measure and convey the trust level of the various elements in a distributed or federated network. Provide architectural and design documents of a system concept that demonstrates the feasibility of the concept.

Phase I Research Objectives

1. Investigate and propose an architecture for determining and conveying trust to the various elements in a GIG-like architecture. (from call)
2. Provide architectural and design documents of a system that demonstrate feasibility. (from call)
3. Further validate our approach through a prototype that exhibits some initial functionality.

Progress Review

Task 1: Ontology Creation

Task 2: Decision Support

Task 3: HW/SW Technologies Support

Task 4: Trust Services

Task 5: Prototype Development

Task 6: Final Technical Report

Proposed Solution

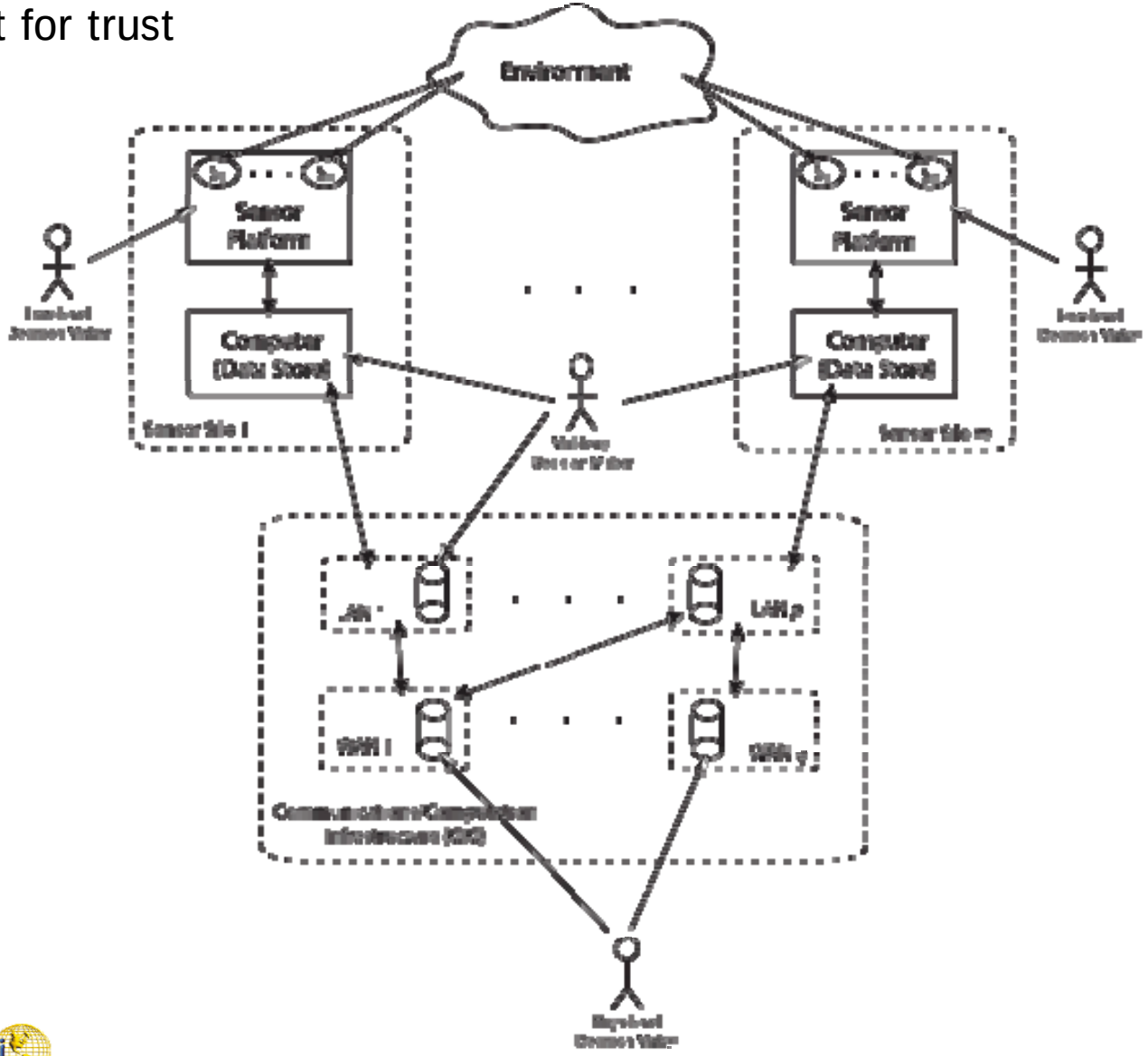
Green Wave – an open architectural framework for determining and conveying trust to the various elements in a GIG-like network.

Features:

- Flexible distributed architectural framework for experimenting with trust.
- Use of semantic technologies incorporated into a hybrid-based trust management system.

Proposed Solution

Environment for trust



Architectural Requirements Review

- Driving requirement: Reasoning about trust computationally means we need a ***machine conveyable & actionable measure for trust***.
- This implies: A (formal) model for trust (what computers need).
- This requires: A method for expressing the trust model (a policy language).
- This allows: the calculation of a trust measure.
- This supports: the use of trust in decision making (beyond the scope of this proposal).

Architectural Requirements Review

- Requirement 1. **Trust Measure.** The trust measure must be quantifiable, allowing for different levels of trust.
- Requirement 2. **Method for Expressing and Calculating Trust.** Trust policies must be expressible in a form that can be used by a network entity to calculate a trust measure.
- Requirement 3. **Use of Trust in Decision-Making.** Ultimately, we want to have some policy for detailing how trust measures should be used in decision-making.
  - Beyond the scope of the current research.
  - Important to recognize this requirement, as it provides the motivation for the calculation of trust to begin with.

Architectural Design – Trust Model

- **Trust Sources** – all trust in a system emanates from identifiable trust sources
  - certification authorities
  - peer assessments
- **Dynamic** – trust evolves over time
  - as new information is obtained
  - as resources change their behavior
- **Evidence-based** – observable actions (or lack thereof) are the basis for direct trust measures
- **Composable** – ability to combine trust measures from different network elements allows for indirect trust measures

Prototype Demonstrations

1. Demonstration of how semantic technologies, trust policies, trust calculations, and decision making can be integrated in Trust Evaluation Architecture

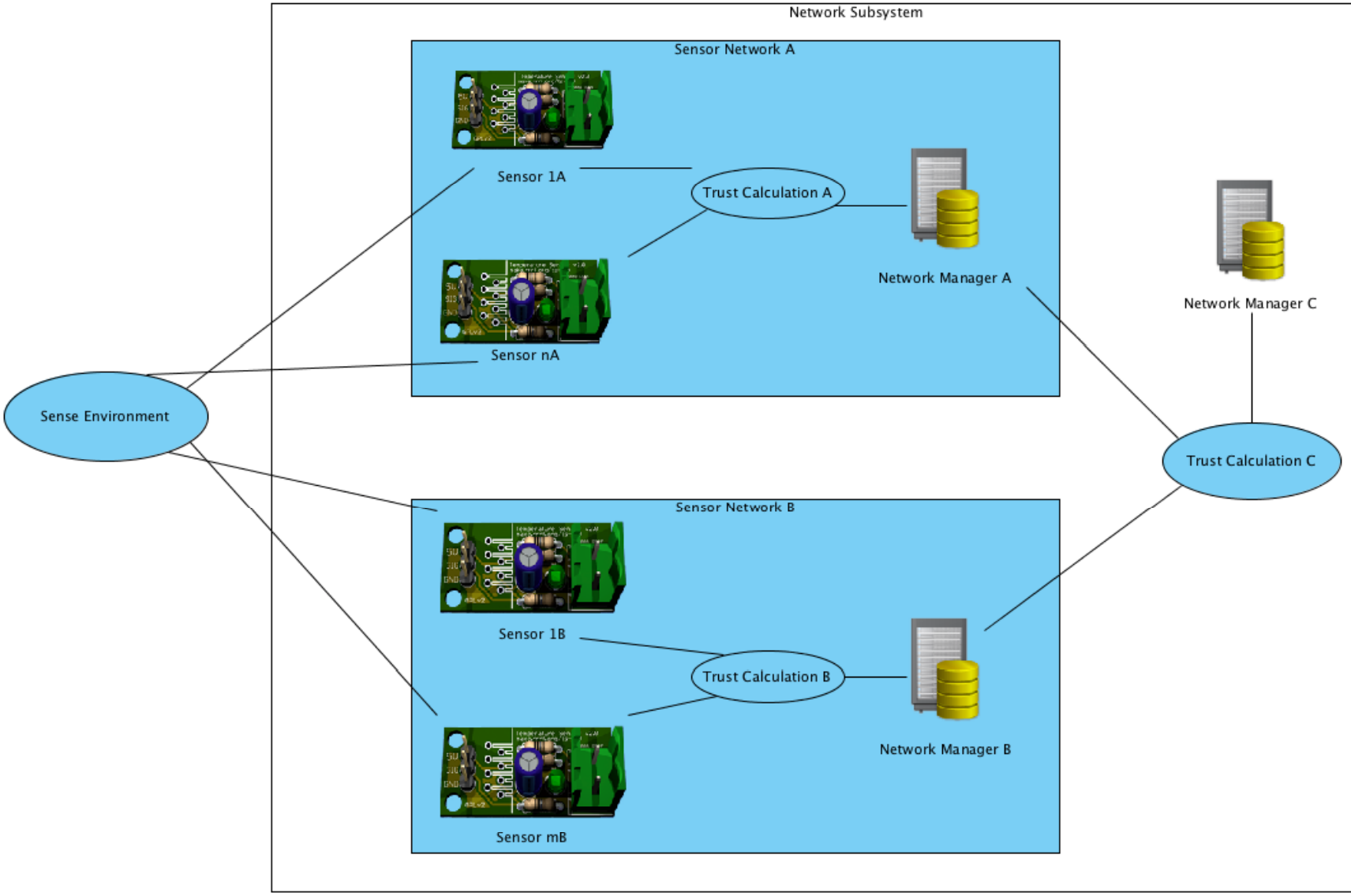
2. Demonstration of how these technologies can be implemented in a next-generation networking infrastructure.

Demonstration Scenario

Use Case 5 (from kickoff briefing): Proximate sensors in a sensor network are providing disparate information. Which sensors do you believe? A functioning sensor can provide spurious values. Are there really chemical weapons on the battlefield? More importantly, can we account for the trust (and provenance) of information within the decision-making framework? Can we show that a "downstream" decision relies on a few facts that have low trust?

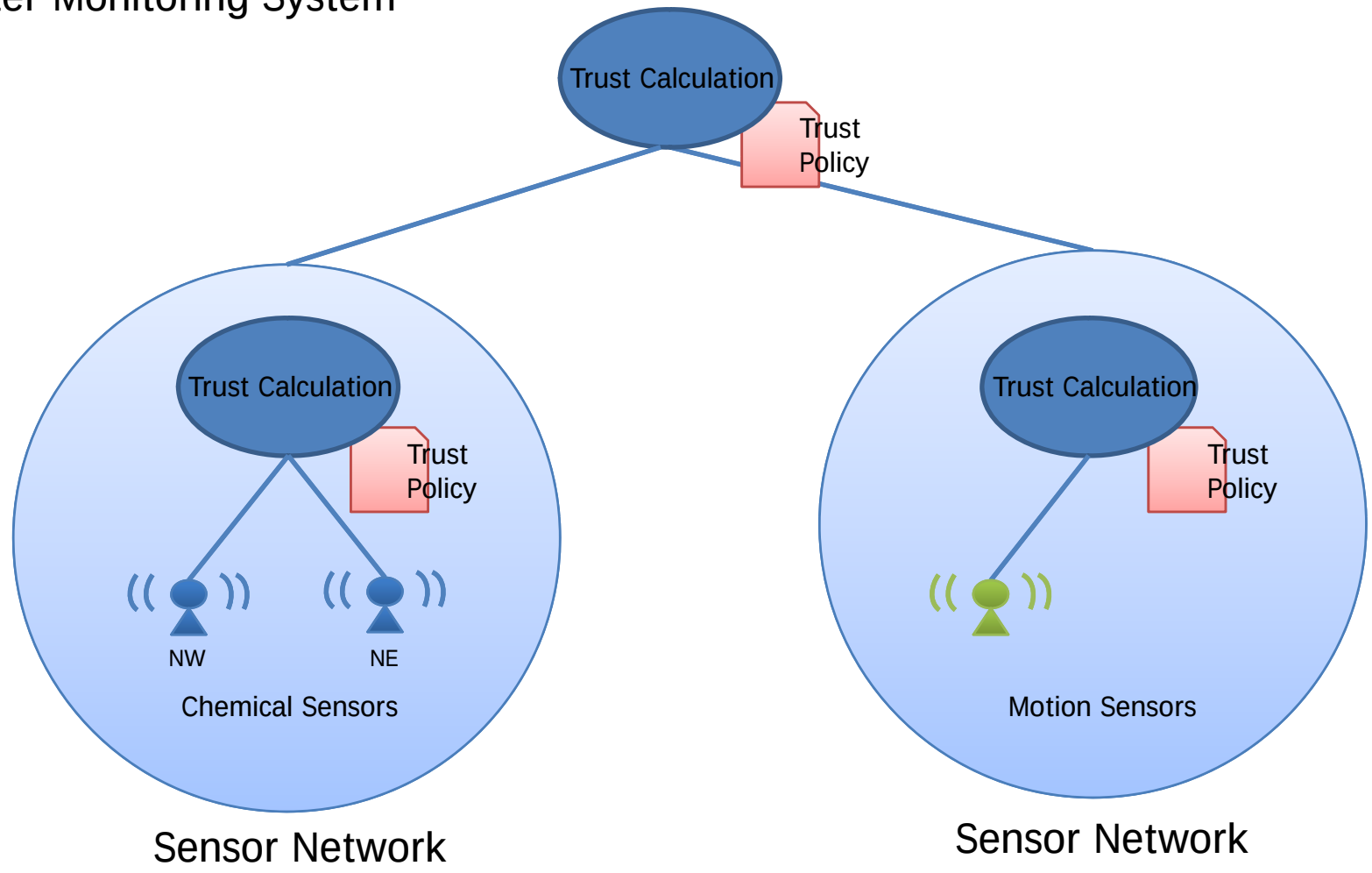
Elaboration of Use Case 5 – Perimeter monitoring system with two sensor networks using semantic technology in trust management.

Use Case 5 – UML

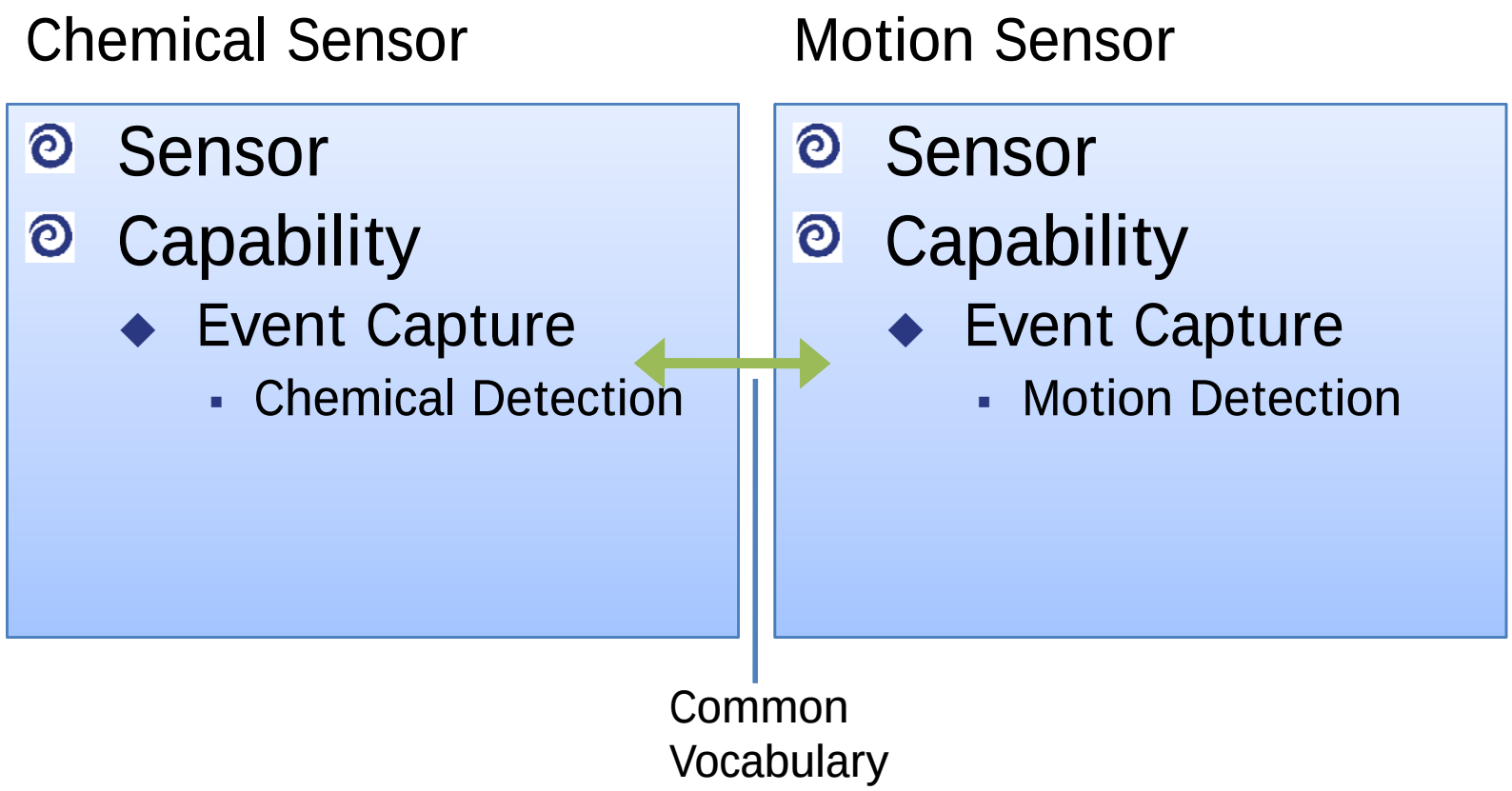


Use Case 5: Sensor Networks

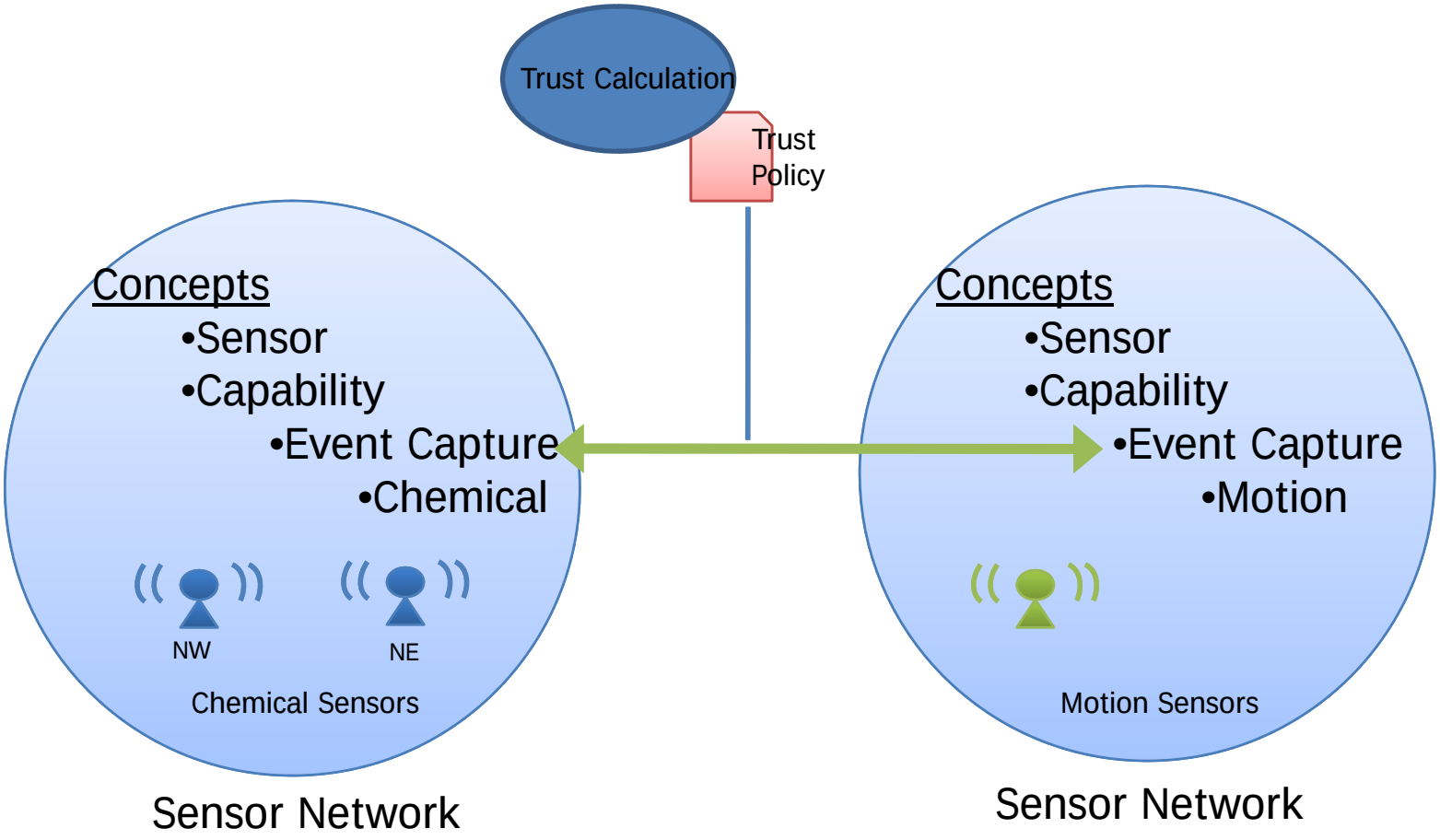
Perimeter Monitoring System



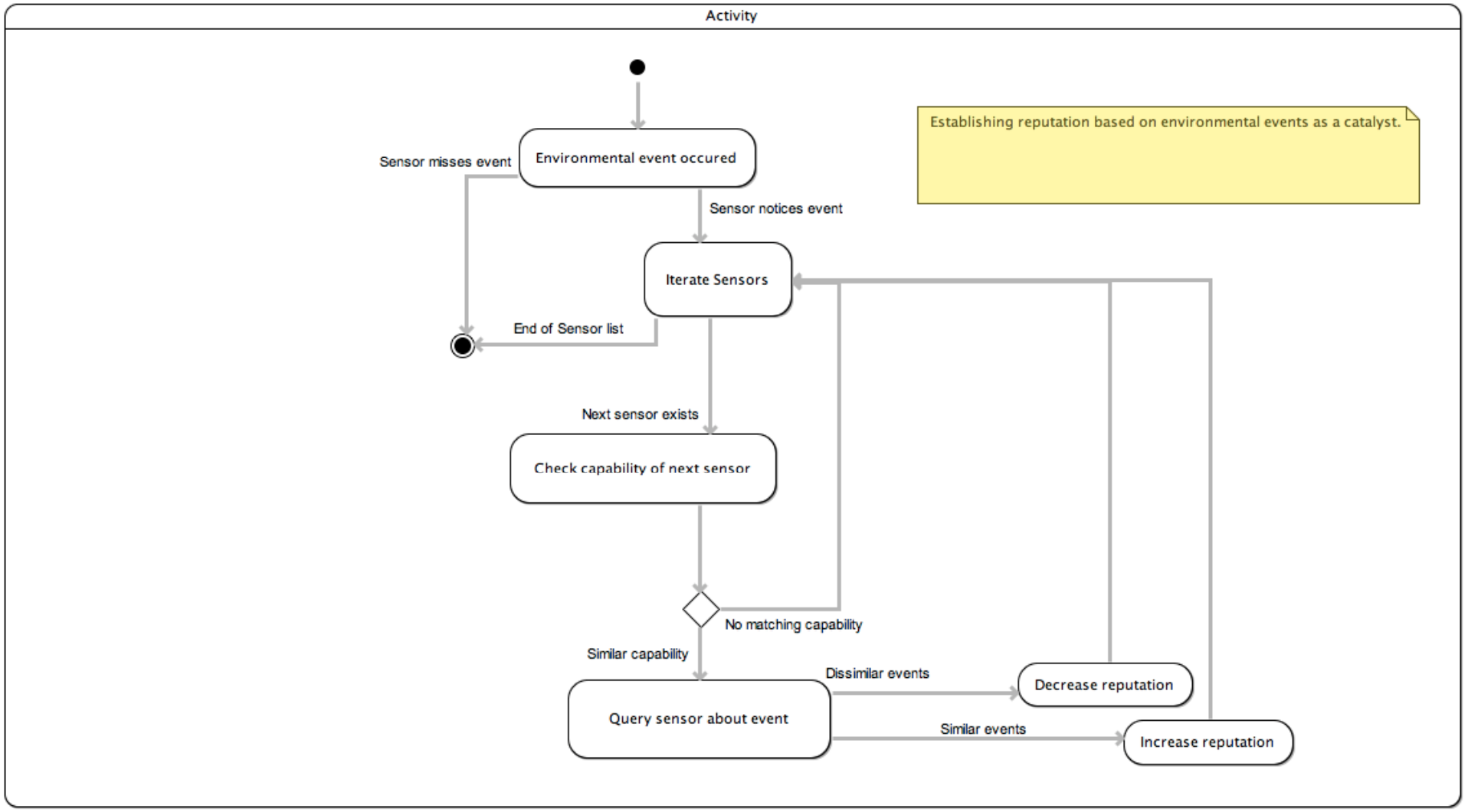
Harmonization



Semantic Assistance in Policies



Use Case 5: Activity Diagram



Demonstration 1

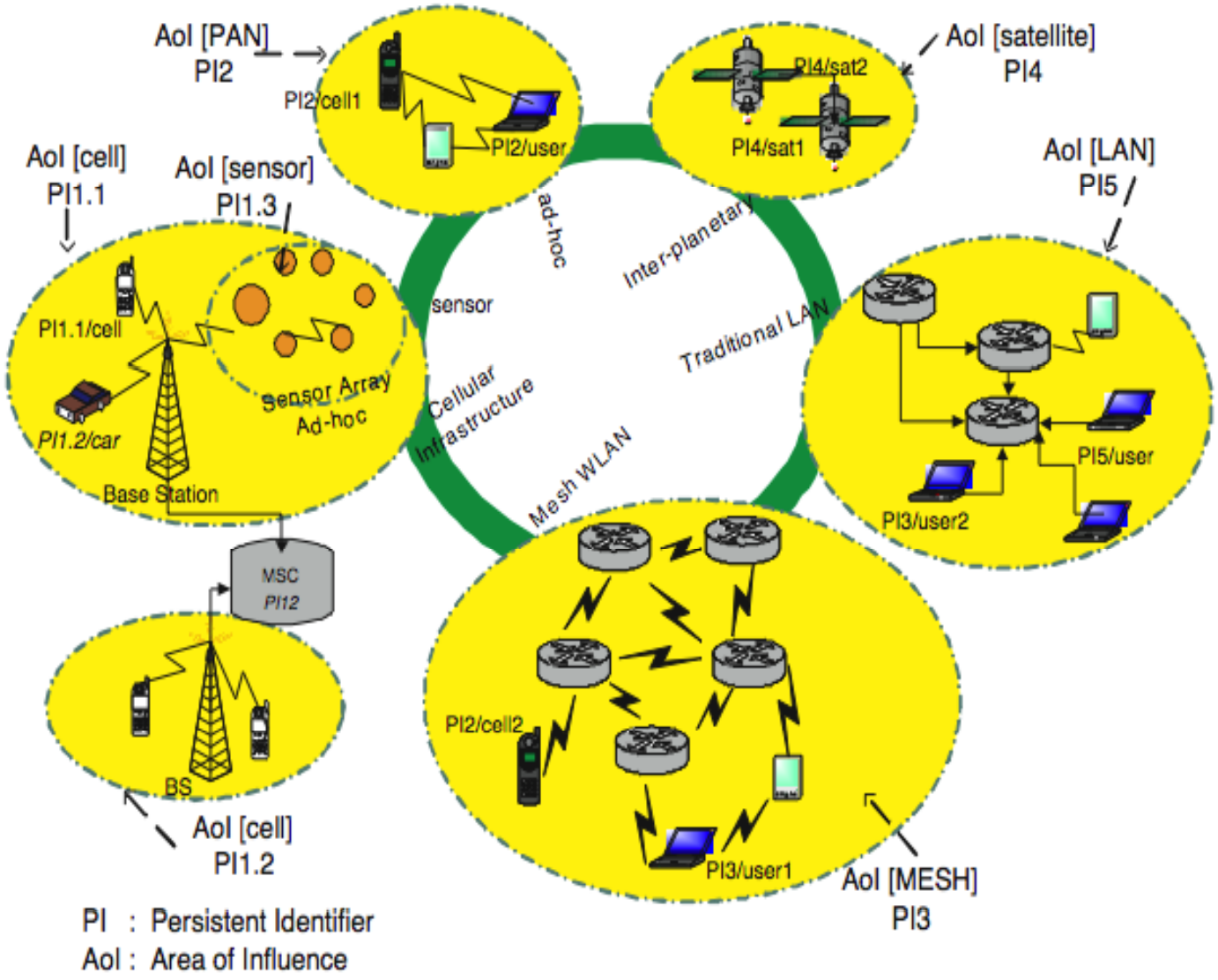
- Fuel Depot at Cape Canaveral AFS
  - Sensor Network A – chemical sensors
  - Sensor Network B – satellite data
- Front-end involves integration with geospatial data to incorporate proximity into trust policies.
- Makes use of Web Information Quality Assessment (WIQA) Framework for the policy language.
- Represents the trust ontology using the TriG syntax.

Demonstration 2

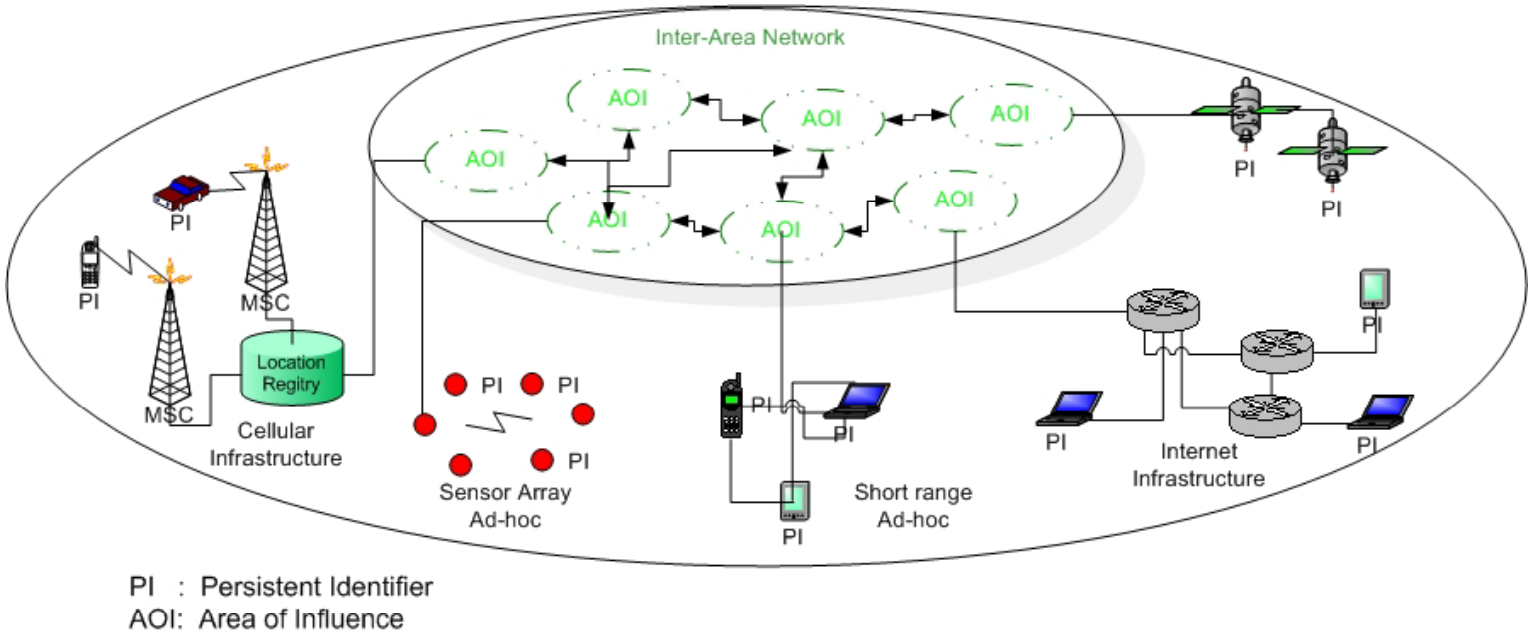
Based on the Transient Network Architecture:

- Persistent Identification
  - Global Uniqueness
  - Certification and Name Resolution
    - Instance (red), Local (yellow), Global (green)
    - Green implies yellow implies red
- Distributed Control-Plan Functionality
  - Supports unstructured networks
  - Persistent Identifier (PI)
    - Holds information associated with each network entity
  - Control-Plane Provisioning
    - Uses the ghost/shell model

Transient Network Architecture



Transient Network Architecture

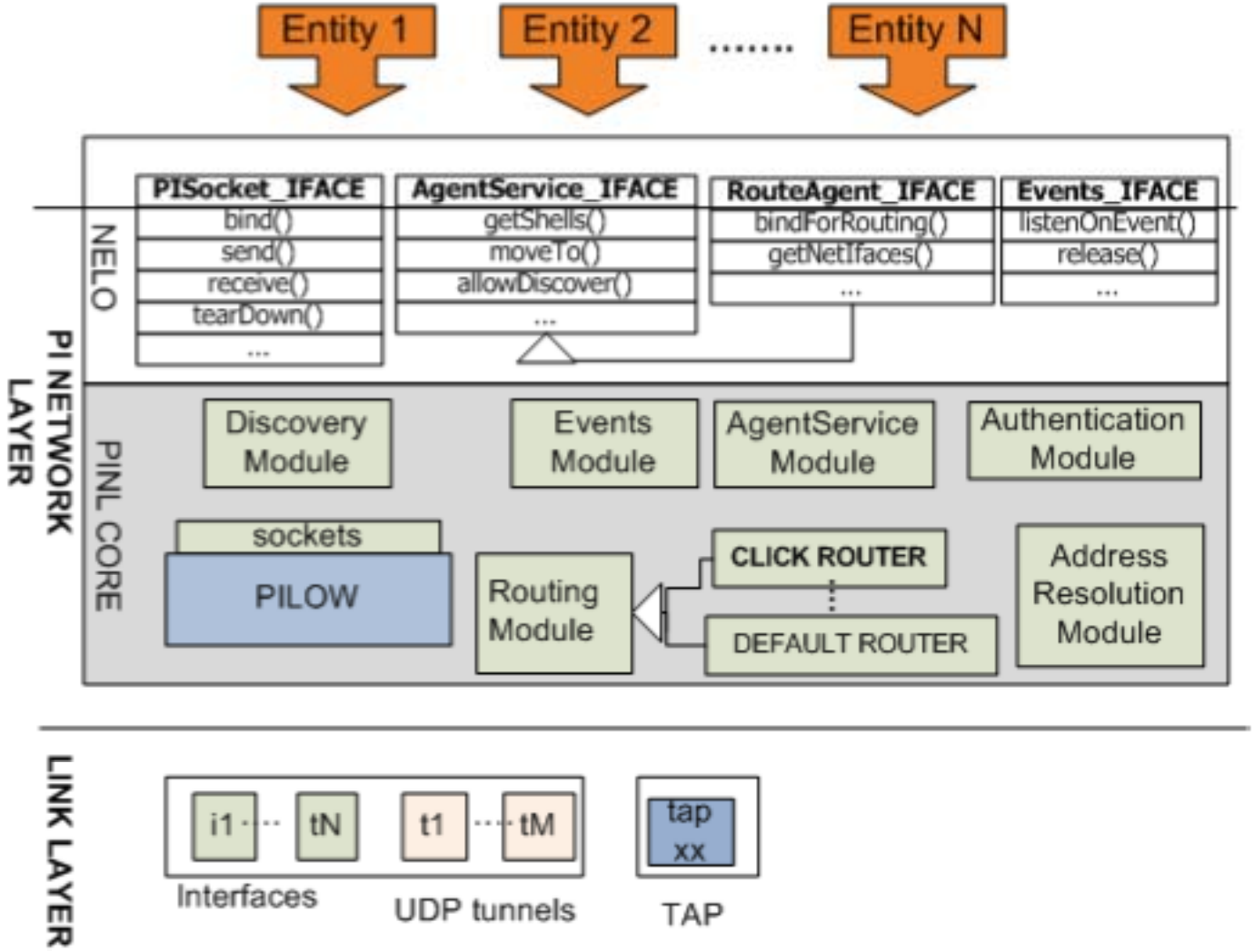


- Three levels of resolution and certification
  - Instance or Red: Ghost
  - Local or Yellow: AoI
  - Global or Green: Green Networks

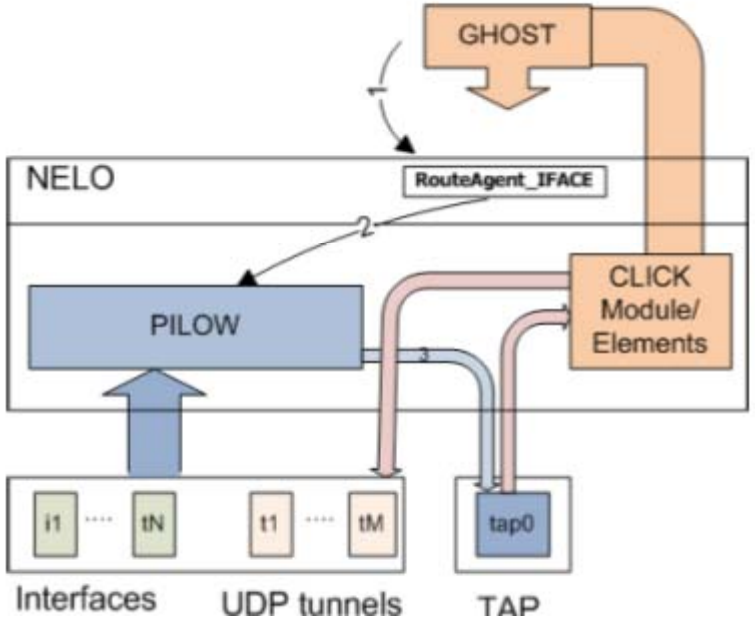
Transient Network Architecture

- Complete network layer, replacing IP.
- Routing based on PIs.
- Entities interact with Persistent Identifier Networking Layer (PINL) via a Neutral Environment Language for Operation (NELO).
- Provided Interfaces:
  - PILOW - routing component.
  - Agent Service - general ghost service.
  - Discovery Module – allows entities to discover one another.
  - Routing Module – accepts packets from PILOW, and routes based on PI.

PINL Layer

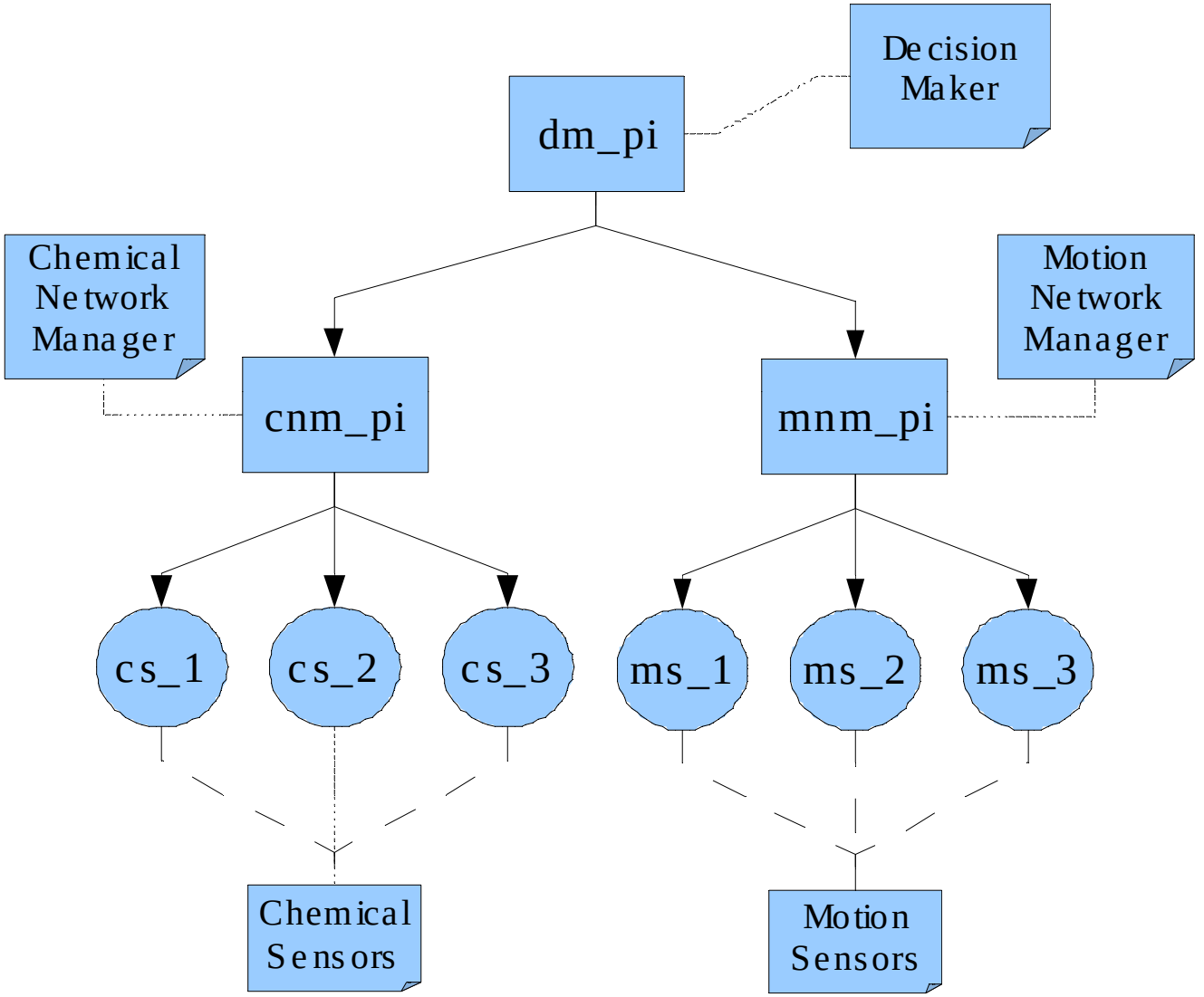

Entity 1
Entity 2
.....
Entity N
PI NETWORK LAYER
NELO
PISocket_IFACE
bind()
send()
receive()
tearDown()
...
AgentService_IFACE
getShells()
moveTo()
allowDiscover()
...
RouteAgent_IFACE
bindForRouting()
getNetIfaces()
...
Events_IFACE
listenOnEvent()
release()
...
PINL CORE
Discovery Module
Events Module
AgentService Module
Authentication Module
sockets
PILOW
Routing Module
CLICK ROUTER
DEFAULT ROUTER
Address Resolution Module
LINK LAYER
i1
tN
t1
tM
Interfaces
UDP tunnels
tap xx
TAP

PINL Layer



Bits	0 – 7	8 – 15	16 – 23	24 – 31
0	Dst. PI Address Length	Src. PI Address Length	Payload Length	
32	Src. Type		Dst. Type	
64	Header Checksum			
⋮	Src. PI Address			
	Dst. PI Address			
	Payload			

Demonstration 2

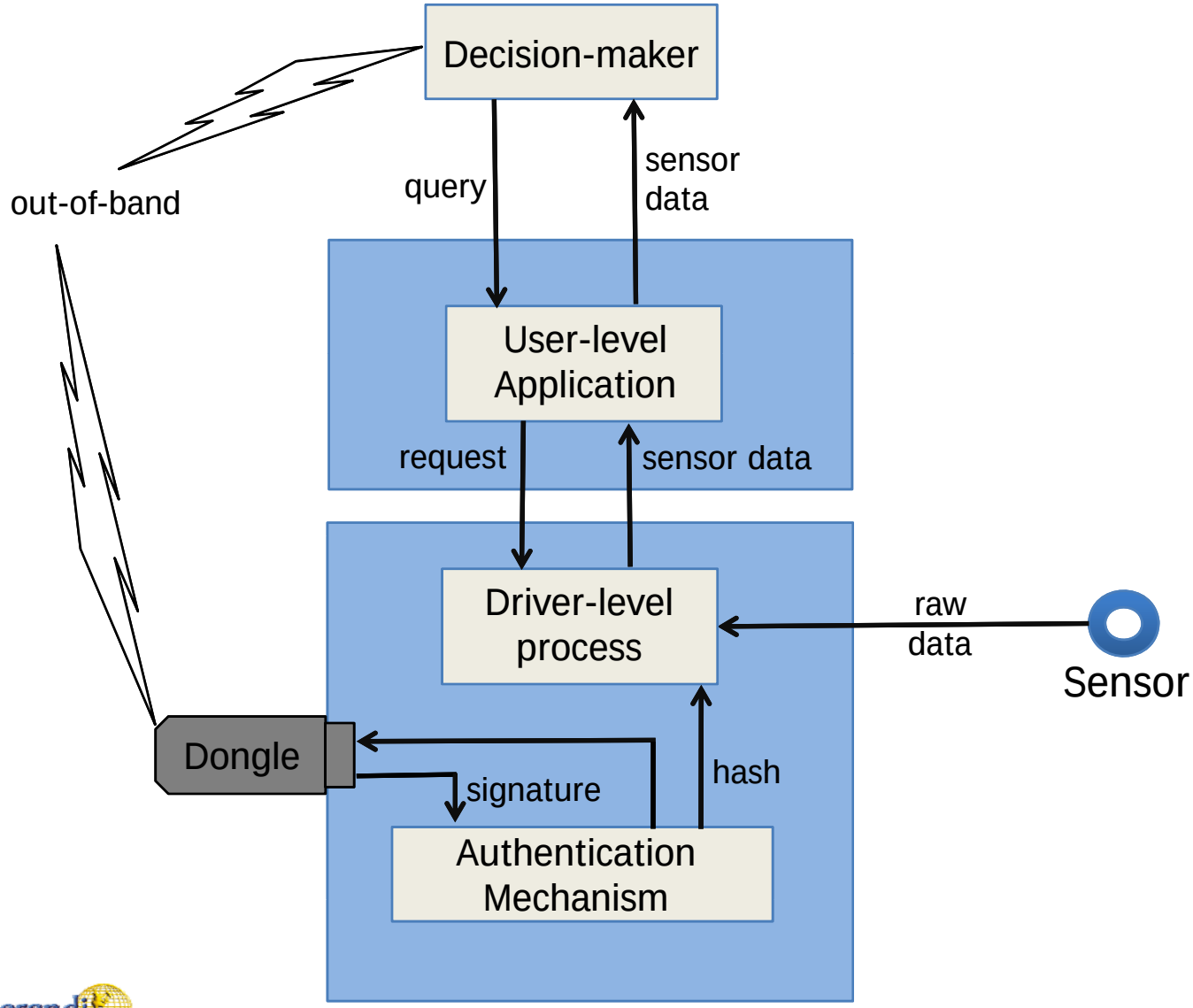


Use Cases

Use Case 4 (from kickoff briefing): An enemy is able to compromise network elements in order to inject spurious information. We do not want to allow the enemy to "escalate" an attack, but we are not sure if it really is an attack? Typically, if a network element is "suspect", it is excluded from the network, even though it may still possess some informational value. E.g., an enemy has corrupted a machine's database, but the GIS unit on the machine is still sending information that can be validated via message digests.

Elaboration of Use Case 4 – HW dongle used in out-of-band communication, and incorporated into trust calculations.

Use Case 4: Enemy Compromise



Recommendations

- We view the Phase I work as one cycle through an iterative and incremental development process (spiral model) aimed at creating a Trust Evaluation Architecture.
- Iteration 1:
  - Elicit initial requirements.
  - Design a preliminary architecture.
  - Prototype pieces of the architecture to validate ideas.
- Iteration 2..n:
  - Continue to iteratively develop the Trust Evaluation Architecture, building upon the previous iterations.
  - Allows for extensive feedback/input from project sponsors.

Specific Recommendations

1. Fully develop formal model for trust. A fully specified formal model for trust allows for:
   - Determination of limits of what the Trust Evaluation Architecture.
   - Proofs for the correctness of its operation under various threat models.
2. Create a complete trust policy specification language on top of the formal model, along with query capabilities with respect to these trust policies:
   - Integrate previous trust research.
   - Allow a decision-maker to specify arbitrarily complex policies, and to reason over these policies.
   - The use of semantic technologies within this reasoning framework should also be more fully explored.
3. Develop a communications framework that allows trust calculations to be integrated at various levels within a communications protocol stack, and within sensor networking environments (sandbox).
   - Initial work demonstrated how trust calculations can be integrated into an experimental communications substrate.
   - The Trust Evaluation Architecture must leverage state-of-the-art advances in security and networking.

Discussion

Acronyms and Terms

AoI	Area of Interest
Ghost	Generic host
GIG	Global Information Grid
NELO	Neutral Environment Language for Operation
PI	Persistent Identifier
PILOW	Persistent Identifier Tables
PINL	Persistent Identifier Networking Layer
SBIR	Small Business Innovation Research
TNA	Transient Network Architecture
WIQA	Web Information Quality Assessment Framework