Agents Meet the Semantic Web on the Wire and in the Aether

Tim Finin

University of Maryland, Baltimore County

AAAI 2004, San Jose, 26 July 2004

http://ebiquity.umbc.edu/v2.1/event/html/id/46/

Joint work with Anupam Joshi, Yun Peng, Scott Cost & many students.

This work was partially supported by DARPA contract F30602-97-1-0215, NSF grants CCR007080 and IIS9875433 and grants from IBM, Fujitsu and HP.

"XML is Lisp's bastard nephew, with uglier syntax and no semantics. Yet XML is poised to enable the creation of a Web of data that dwarfs anything since the Library at Alexandria."

-- Philip Wadler, *Et tu XML? The fall of the relational empire*, VLDB, Rome, September 2001.

"The web has made people smarter. We need to understand how to use it to make machines smarter, too."

-- Michael I. Jordan, paraphrased from a talk at AAAI, July 2002 by Michael Jordan (UC Berkeley)

"The Semantic Web will globalize KR, just as the WWW globalize hypertext"

-- Tim Berners-Lee

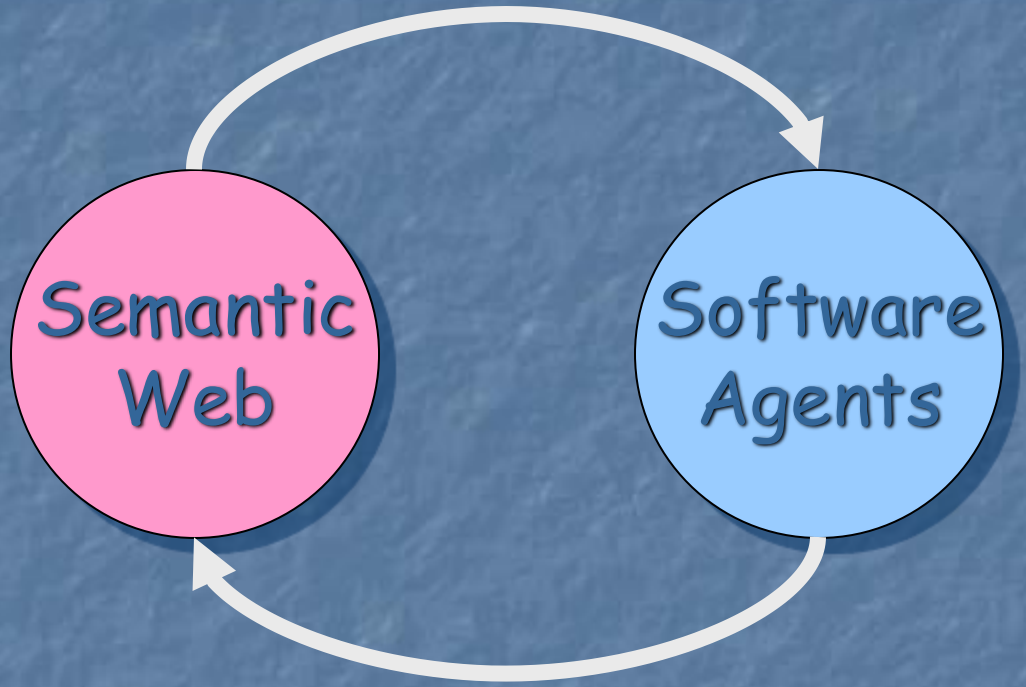
"The multi-agent systems paradigm and the web both emerged around 1990. One has succeeded beyond imagination and the other has not yet made it out of the lab."

-- Anonymous, 2001

This talk

- I'll focus on how we've found semantic web technology useful for agents and for pervasive computing
- I'll illustrate this using five recent projects:
  - (1) Taga: trading agents and the semantic web
  - (2) Rei: Policies for security, trust and privacy
  - (3) Cobra: context aware pervasive computing
  - (4) MoGatu: Trust in mobile data management
  - (5) ManetID: Intrusion detection in mobile ad-hoc networks
- Pointing out the lessons we've learned…

(1) The Celebrity Couple

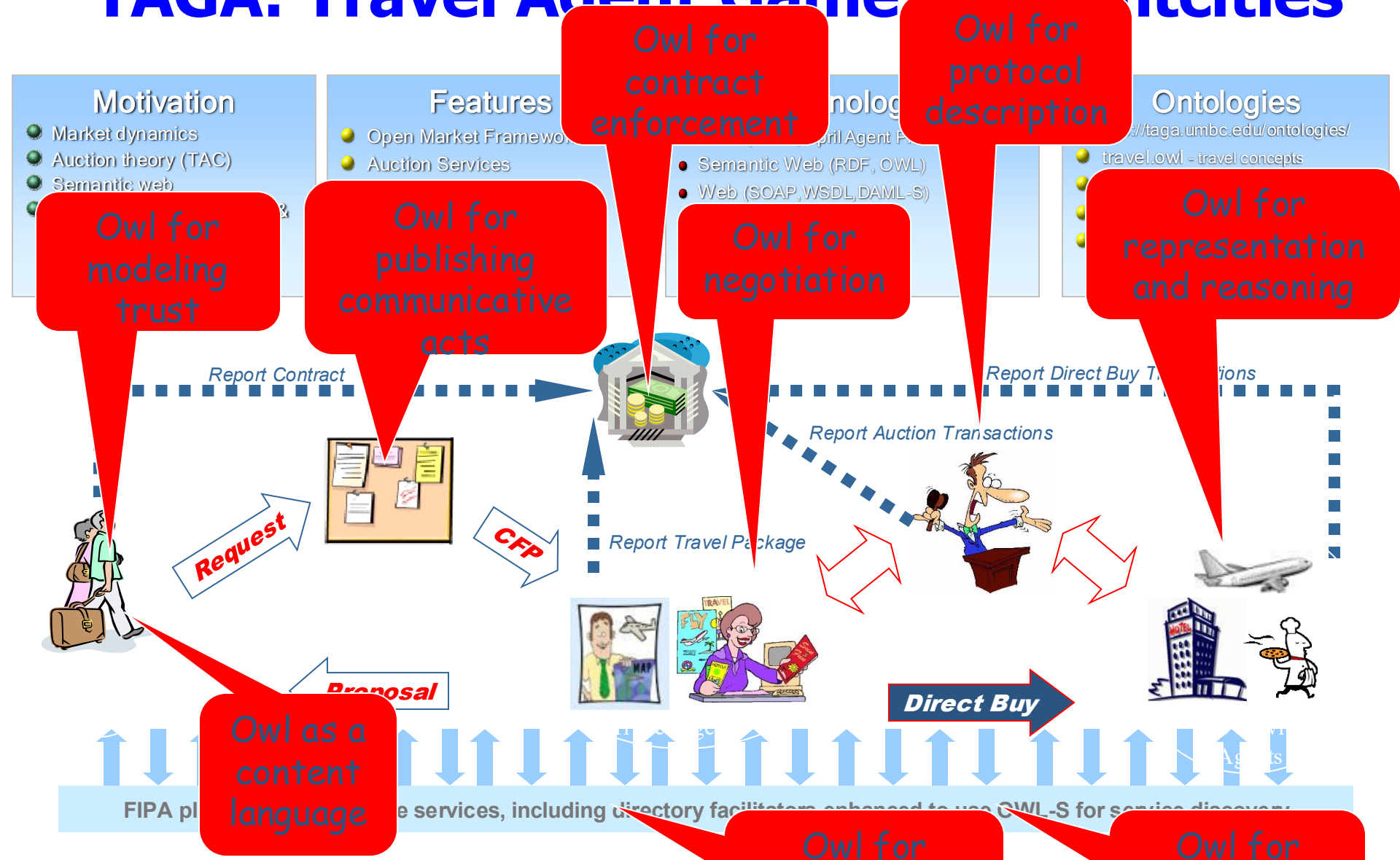


In 2002, *Geek Gossip* gushed "The semantic web will provide content for internet agents, and agents will make the semantic web "come alive". *Looks like a match made in Heaven!"*

(1) Trading Agents

- We've built an agent-based environment inspired by TAC, the Trading Agent Competition
  - TAC is a forum for dynamic trading agent research with games run in the last five years
  - TAC Classic involves a travel procurement, with agents buying and selling goods for clients and scored on the cost and clients' preferences for trips assembled.
  - TAC is organized around a central auction server
- Our goal was to open up the system, allowing peer-to-peer communication among agents as well various kinds of mediator, auction, discovery, service provider agents … and to see how well the semantic web works as the common knowledge infrastructure.

TAGA: Travel Agent Game in Agentcities



http://taga ed

What we learned

- OWL is a good KR language for a reasonably sophisticated MAS
  - Integrates well with FIPA standards
- OWL made it easy to mix content from different ontologies unambiguously
  - Supporting partial understanding & extensibility
- The use of OWL supported web integration
  - Using information published on web pages and integrating with web services via WSDL and SOAP
- OWL has limitations: no rules, no default reasoning, graph semantics, …
  - Some of which are being addressed

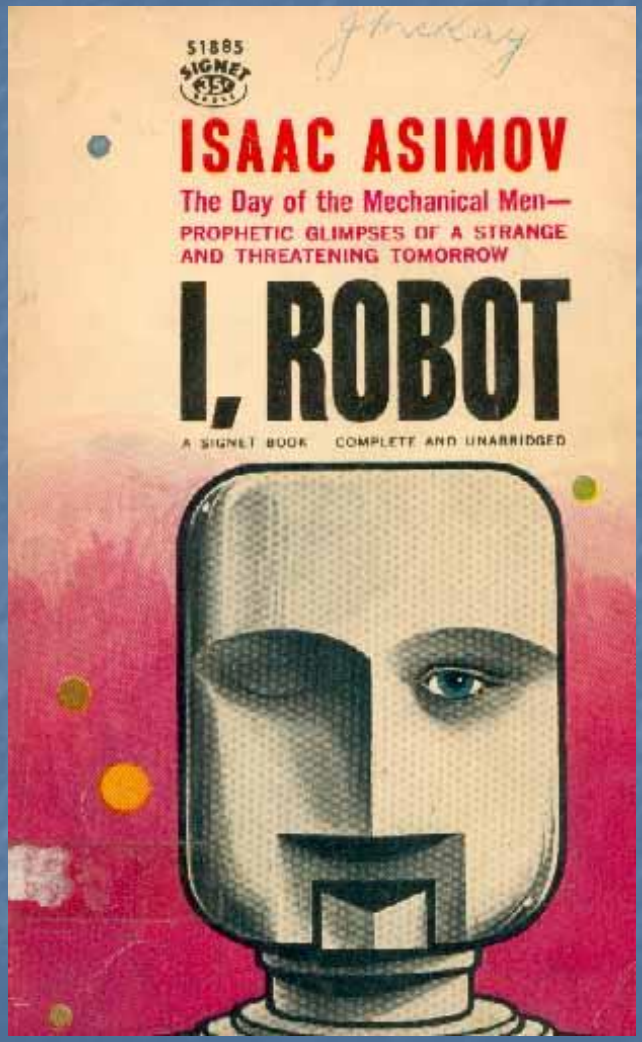
(2) It's policies all the way down

1 A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

2 A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

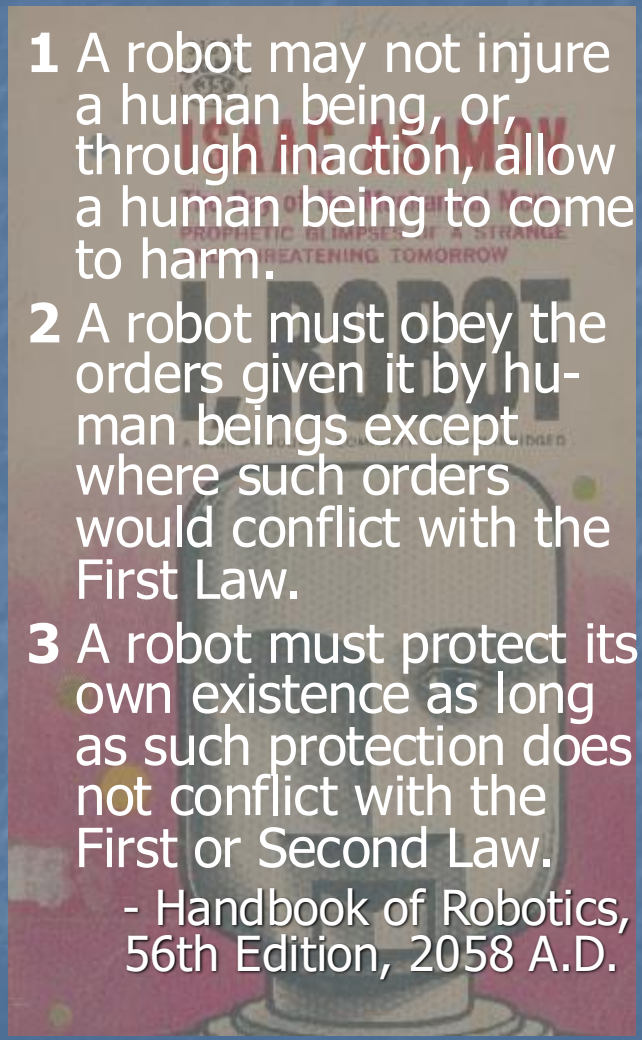
3 A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

- Handbook of Robotics, 56th Edition, 2058 A.D.



(2) It's policies all the way down

- In Asimov's world, the robots didn't always strictly follow their policies
  - Unlike traditional "hard coded" rules like DB access control & OS file permissions
- Autonomous agents need policies as "norms of behavior" to be followed to be good citizens
- So, it's natural to worry about …
  - How agents governed by multiple policies can resolve conflicts among them
  - How to deal with failure to follow policies – sanctions, reputation, etc.
  - Whether policy engineering will be any easier than software engineering



Our Approach

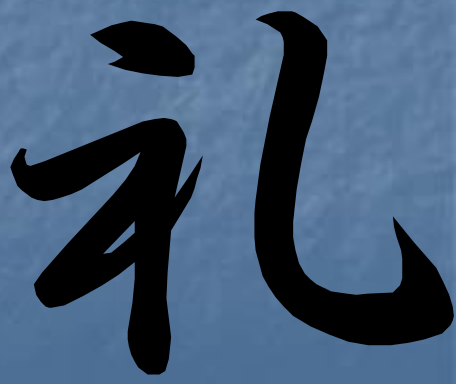
- Policies are useful at virtually all levels
  - OS, networking, data management, applications
- Declarative policies guide the behavior of entities in open, distributed environments
  - Positive & negative authorizations & obligations
  - Focused on domain actions
  - Policies are based on attributes of the action (and its actor and target) and the general context – not just on their *identity* of the actor

Rei Policy Language

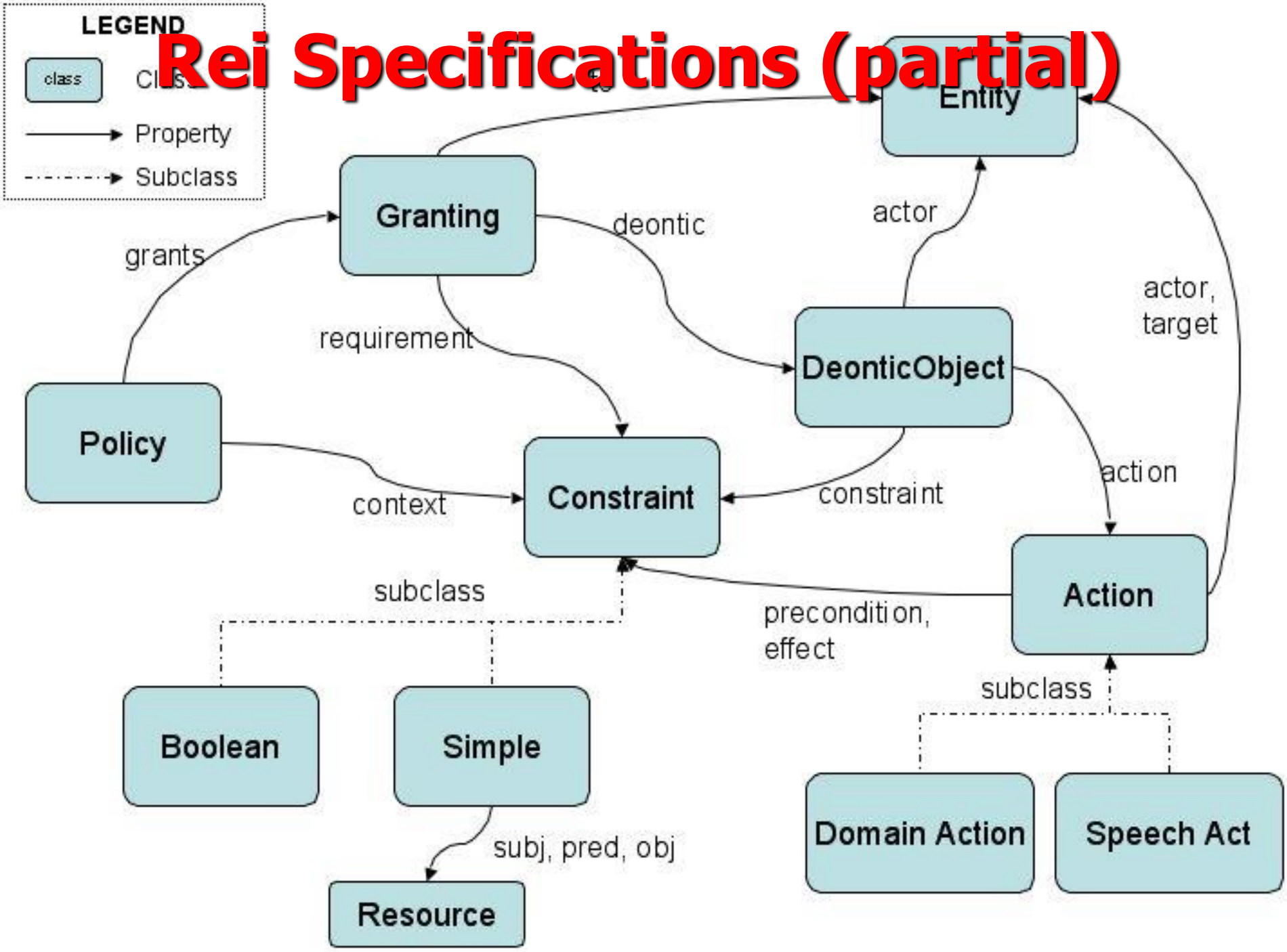
- Developed several versions of Rei, a policy specification language, encoded in (1) Prolog, (2) RDFS, (3) OWL
- Used to model different kinds of policies
  - Authorization for services
  - Privacy in pervasive computing and the web
  - Conversations between agents
  - Team formation, collaboration & maintenance
- The OWL grounding enables policies that reason over SW descriptions of actions, agents, targets and context

Rei Policy Language

- Rei is a declarative policy language for describing policies over actions
  - Reasons over domain dependent information
- Currently represented in OWL + logical variables
- Based on deontic concepts
  - Permission, Prohibition, Obligation, Dispensation
- Models speech acts
  - Delegation, Revocation, Request, Cancel
- Meta policies
  - Priority, modality preference
- Policy engineering tools
  - Reasoner, IDE for Rei policies in Eclipse



Rei Specifications (partial)



Applications – past, present & future

- Coordinating access in *supply chain management system*
- Authorization policies in a *pervasive computing environment*
- Policies for team formation, collaboration, information flow in *multi-agent systems*
- Security in *semantic web services*
- Privacy and trust on the *Internet*
- Privacy in *pervasive computing environments*

Example: Security and Trust for Semantic Web Services

- Semantic web services are web services described using OWL-S
- Policy-based security infrastructure
- Advantages of using policies:
  - Expressive -- can be over descriptions of requester, service & context
  - Authorization: Rules for access control
  - Privacy: Rules for protecting information
  - Confidentiality: Cryptographic characteristics of information exchanged

Example policies

- Authorization
  - Policy 1: Stock service not accessible after market closes
  - Policy 2: Only LAIT lab members who are Ph.D. students can use the LAIT lab laser printer
- Privacy/Confidentiality
  - Policy 3: Do not disclose my my SSN
  - Policy 4: Do not disclose my home address or facts from which it could be easily discovered
  - Policy 5: Do not use a service that doesn't encrypt all input/output
  - Policy 6: Use only those services that required an SSN if it is encrypted

Example

- Mary is looking for a reservation service
  - foaf description
  - Confidentiality policy
- BravoAir is a reservation service
  - OWL-S description
  - Authorization policy

    Only users belonging to the same project as John can access the service

Mary

```
<!-- Mary's FOAF description -->
<foaf:Person rdf:ID="mary">
<foaf:name>Mary Smith</foaf:name>
    <foaf:title>Ms</foaf:title>
    <foaf:firstName>Mary</foaf:firstName>
    <foaf:surname>Smith</foaf:surname>
    <foaf:homepage
    rdf:resource="http://www.somewebsite.com/marysmith.html"/>
    <foaf:currentProject rdf:resource="
    http://www.somewebsite.com/SWS-Project.rdf "/>

    <sws:policyEnforced rdf:resource="&mary;ConfidentalityPolicy"/>
</foaf:Person>

</rdf:RDF>
```

Bravo Policy

```
<entity:Variable rdf:about="&bravo-policy;var1"/>
<entity:Variable rdf:about="&bravo-policy;var2"/>

<constraint:SimpleConstraint
        rdf:about="&bravo-
    policy;GetJohnProject"
     constraint:subject="&john;John"
     constraint:predicate="&foaf;currentProject"
     constraint:object="&bravo-policy;var2"/>

<constraint:SimpleConstraint
        rdf:about="&bravo-
    policy;SameProjectAsJohn"
     constraint:subject="&bravo-policy;var1"
     constraint:predicate="&foaf;currentProject"
     constraint:object="&bravo-policy;var2"/>

<!-- constraints combined -->
<constraint:And rdf:about="&bravo-
    policy;AndCondition1"
     constraint:first="&bravo-
    policy;GetJohnProject"
     constraint:second="&bravo-
    policy;SameProjectAsJohn"/>
```

```
<deontic:Right rdf:about="&bravo-
    policy;AccessRight">
    <deontic:actor rdf:resource="&bravo-
    policy;var1"/>
    <deontic:action rdf:resource="&bravo-
    service;BravoAir_ReservationAgent"/>
    <deontic:constraint rdf:resource="&bravo-
    policy;AndCondition1"/>
</deontic:Right>

.....

<rdf:Description rdf:about="&bravo-
    service;BravoAir_ReservationAgent">
    <sws:policyEnforced rdf:resource="&bravo-
    policy;AuthPolicy"/>
</rdf:Description>
```

How it works

Mary

BravoAir
Web service

URL to foaf desc +
query request

Matchmaker
+ Reasoner

<sws:policyEnforced rdf:resource =
"&bravo-policy;AuthPolicy"/>

Bravo Service
OWL-S Desc

How it works

Mary

Mary's query = Bravo Service ? YES

Extract Bravo's policy

BravoAir Web service

```
<deontic:Right rdf:about="&bravo-policy;AccessRight">
        <dec
        <dec                                                    ent"/>
        <dec
</deontic:Ri

<policy:Gran
        <pol
        <policy:deontic rdf:resource="&bravo-policy;AccessRight"/>
</policy:Gra

<sws:AuthorizationPolicy rdf:about="&bravo-policy;AuthPolicy">
        <pol
</sws:Autho

<rdf:Descrip
        <sws
</rdf:Descri
```

```
<constraint:SimpleConstraint
  rdf:about = "&bravo-policy;GetJohnProject"
  constraint:subject="&john;John"
  constraint:predicate="&foaf;currentProject"
  constraint:object="&bravo-policy;var2"/>

var2 = http://www.somewebsite.com/SWS-Project.rdf
```

```
<foaf:currentProject rdf:resource =
"http://www.somewebsite.com/SWS-Project.rdf"/>
```

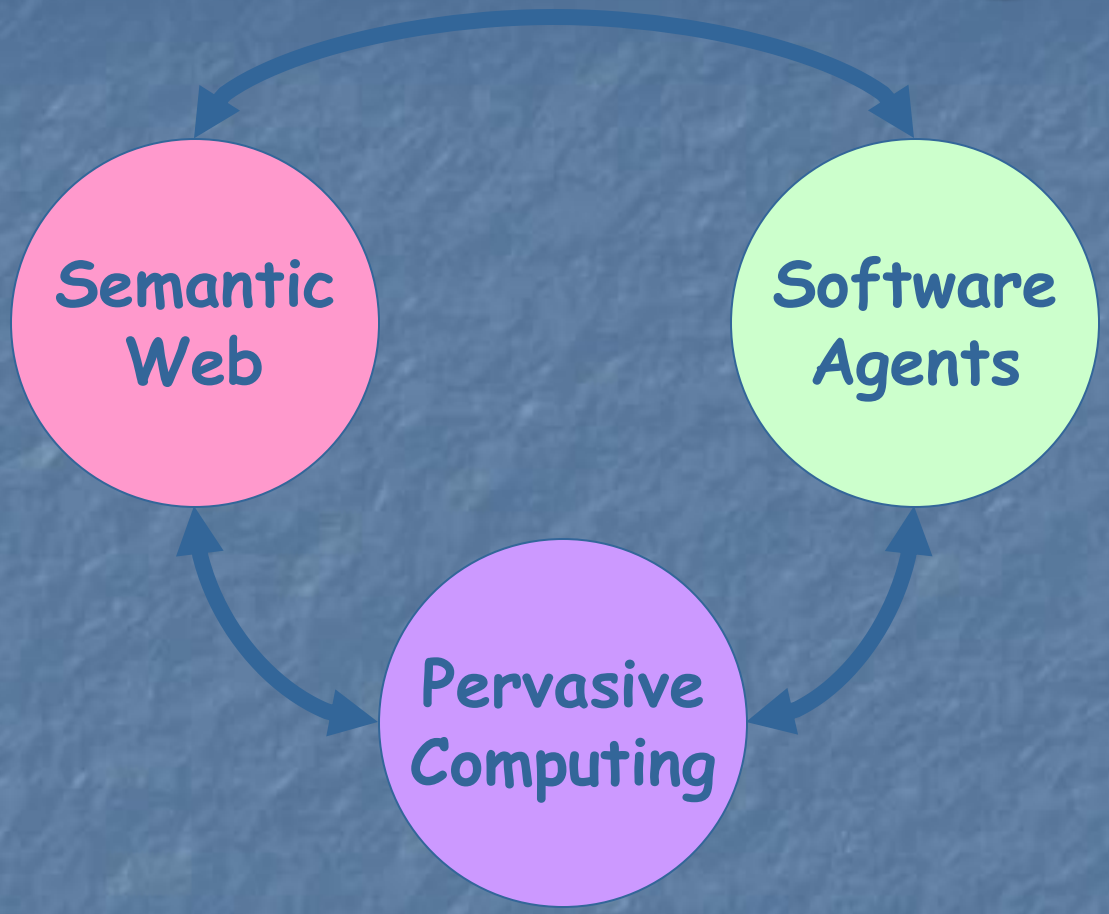
```
<constraint:SimpleConstraint
  rdf:about="&bravo-policy;SameProjectAsJohn"
  constraint:subject="&bravo-policy;var1"
  constraint:predicate="&foaf;currentProject"
  constraint:object="&bravo-policy;var2"/>

Is  the constraint true when
var2 = http://www.somewebsite.com/SWS-Project.rdf
var1 = http://www.cs.umbc.edu/~lkagal1/rei/examples/sws-sec/MaryProfile.rdf
```

What we learned

- Declarative policies can be used to model security, trust and privacy constraints
- Reasonably expressive policy languages can be encoded on OWL
- This enables policies to depend on attributes and context information available on the semantic web
- Policies are applicable at almost every level of the stack, from systems and networking to multiagent applications.

(3) A Love Triangle?



Even matches made in Heaven don't always work out as planned.

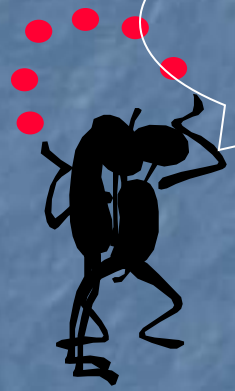
(3) Pervasive Computing

"*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*" – Mark Weiser

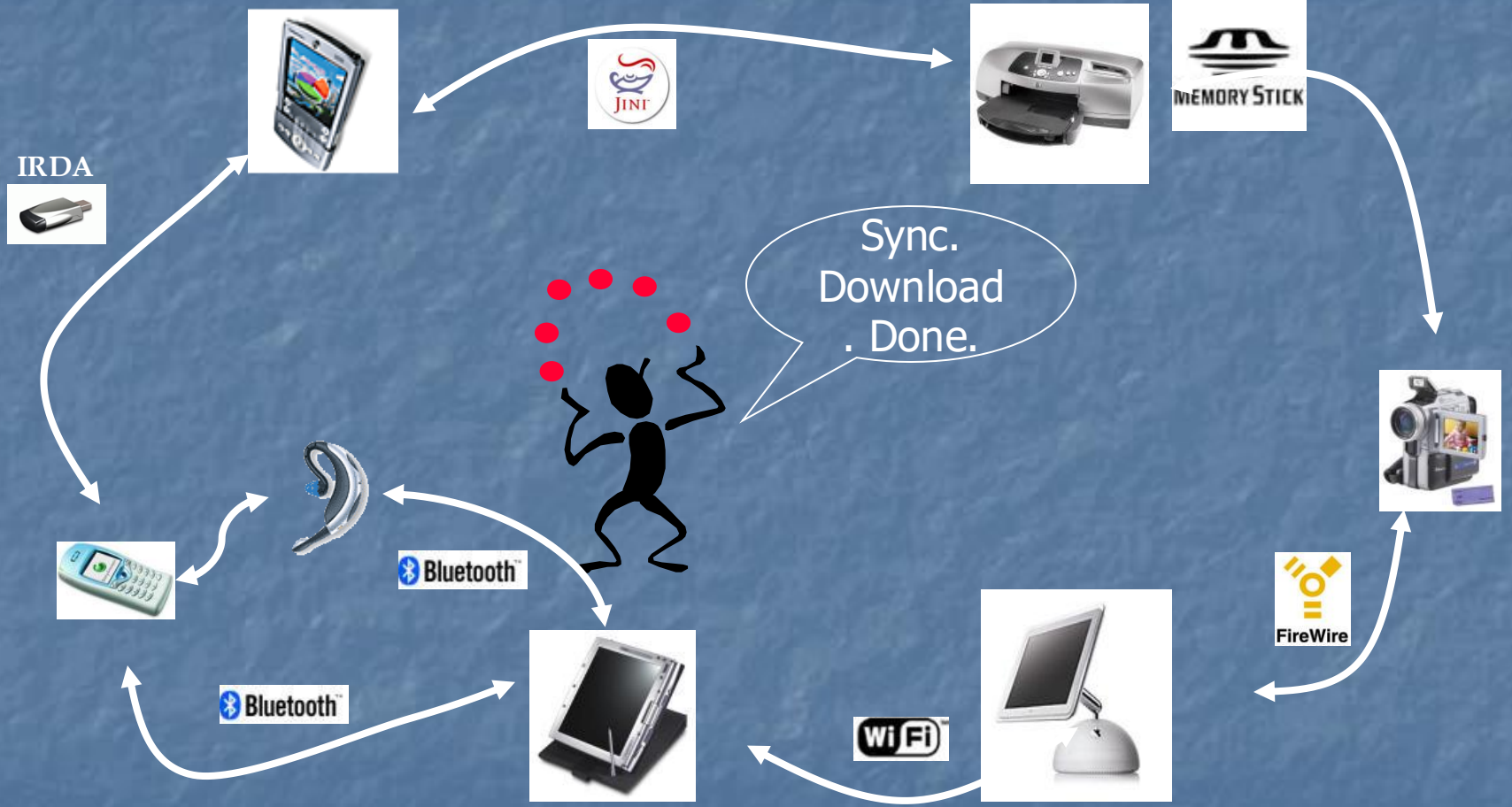
Think: writing, central heating, electric lighting, water services, …

Not: taking your laptop to the beach, or immersing yourself into a virtual reality

Communication is a key enabler

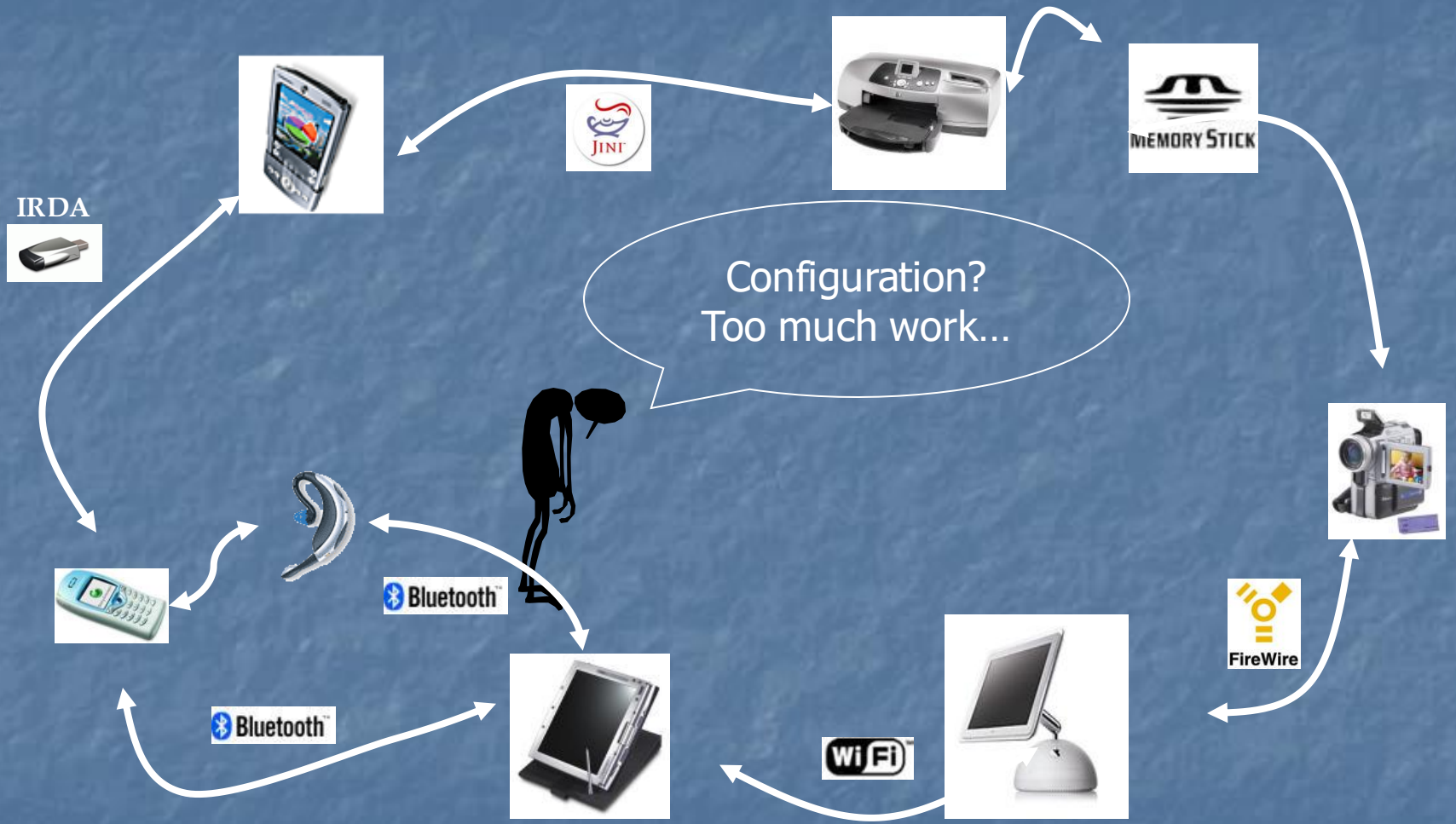


We have many standards



Wireless technologies like WI-FI, IRDA, Bluetooth, UltraWideBand, CDMA, GSM, GPS, etc. are opening up many possibilities.

We have many standards



But your have to be a dedicated geek to configure everything to interoperate and continually tweak things to keep them working.

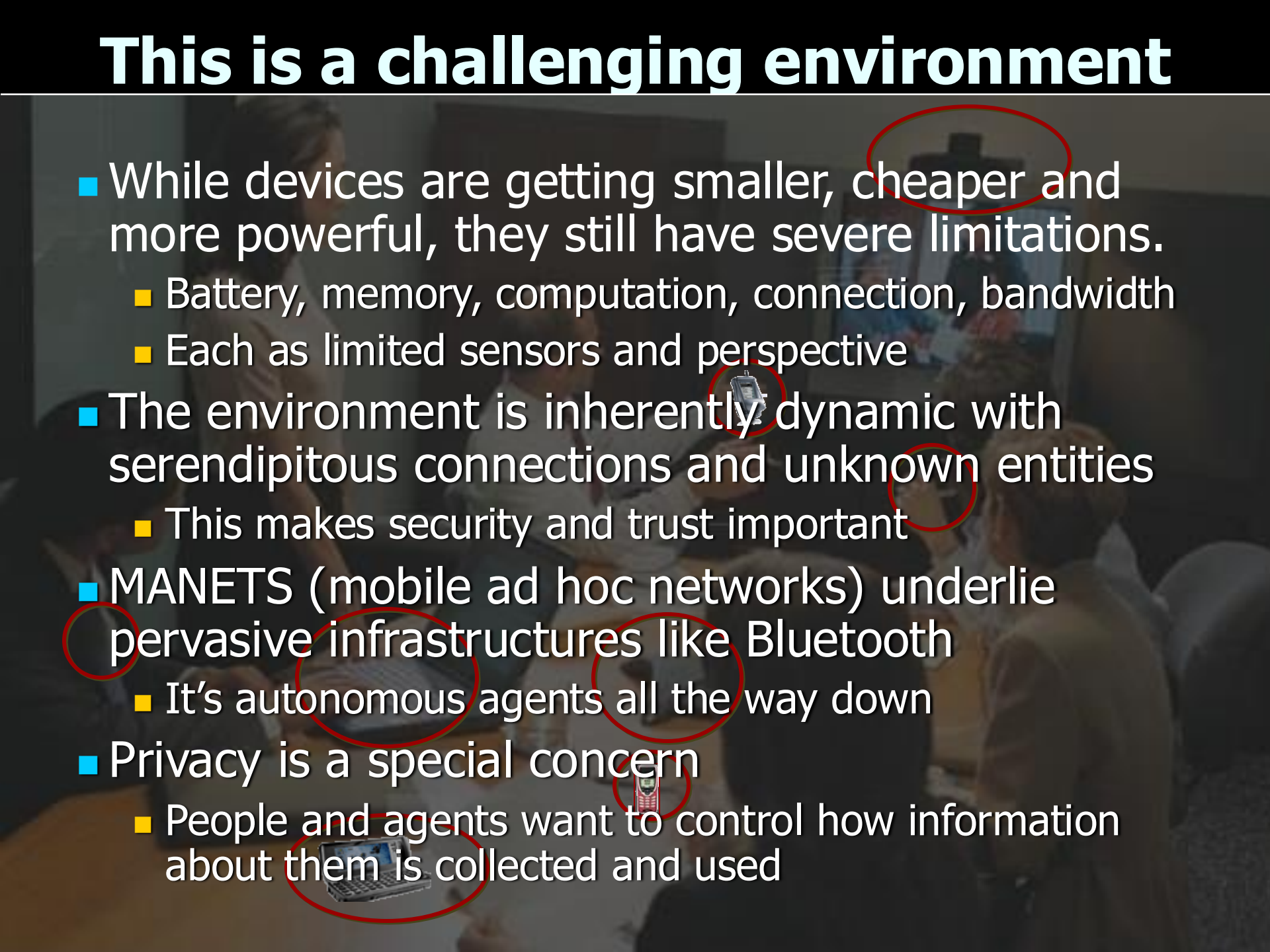
The devices must be more social

The devices must be more social

and intelligent

This is a challenging environment

- While devices are getting smaller, cheaper and more powerful, they still have severe limitations.
  - Battery, memory, computation, connection, bandwidth
  - Each as limited sensors and perspective
- The environment is inherently dynamic with serendipitous connections and unknown entities
  - This makes security and trust important
- MANETS (mobile ad hoc networks) underlie pervasive infrastructures like Bluetooth
  - It's autonomous agents all the way down
- Privacy is a special concern
  - People and agents want to control how information about them is collected and used

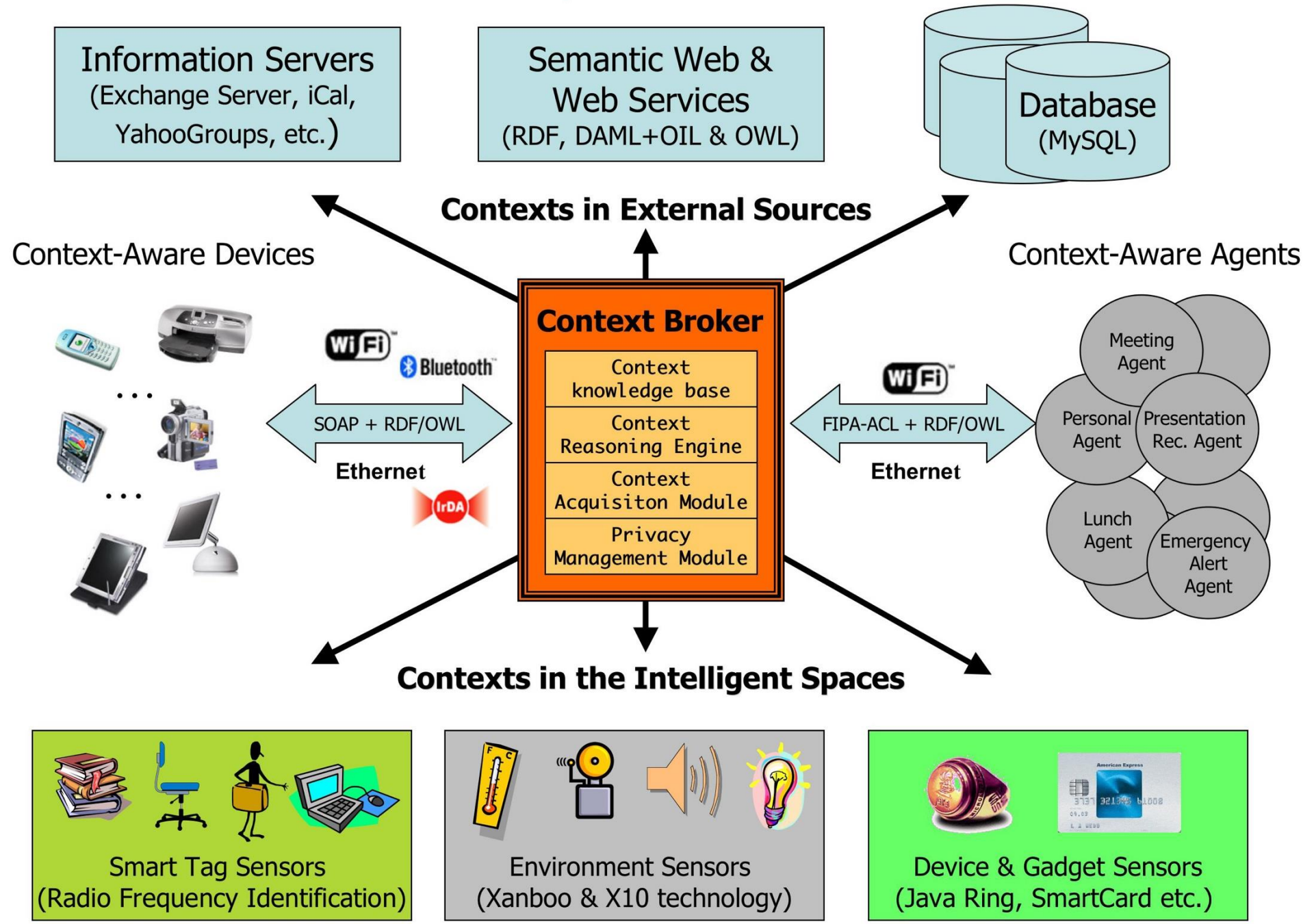


Representing and Reasoning about Context

CoBrA: a broker centric agent architecture for supporting pervasive context-aware systems

- Using SW ontologies for context modeling and reasoning about devices, space, time, people, preferences, meetings, etc.
- Using logical inference to interpret context and to detect and resolve inconsistent knowledge
- Allowing users to define policies controlling how information about them is used and shared

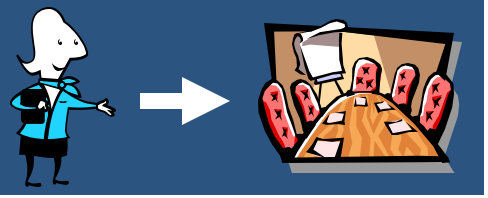
A Bird's Eye View of CoBrA



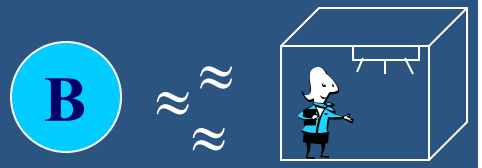
A Typical CoBrA Use Case

Alice in Wonderland*

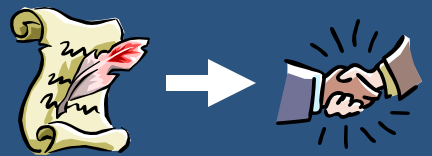
Alice enters a conference room



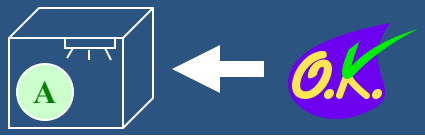
The broker detects Alice's presence



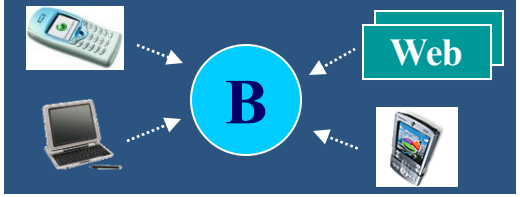
The broker negotiates privacy policy with Alice



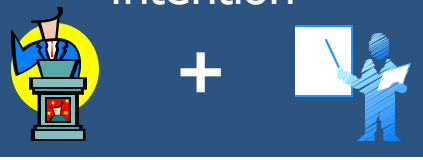
Policy says, "can share with any agents in the room"



The broker builds the context model



The broker knows Alice's role and intention

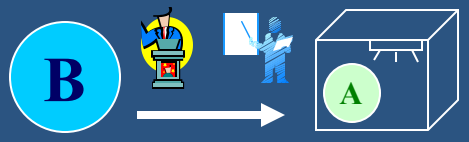


* Our intelligent meeting room

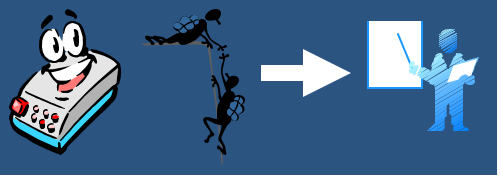
A Typical CoBrA Use Case

Alice in Wonderland

The broker informs the subscribed agents

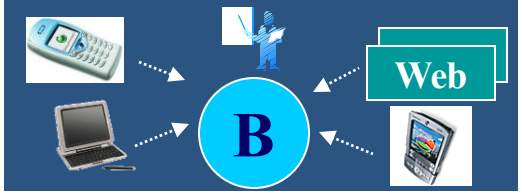


The projector agent wants to help Alice

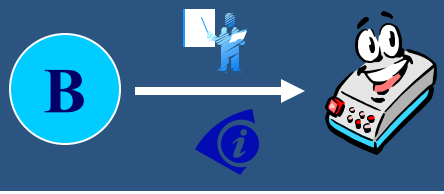


The projector agent asks slide show info.

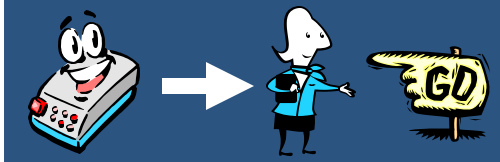
The broker acquires the slide show info.



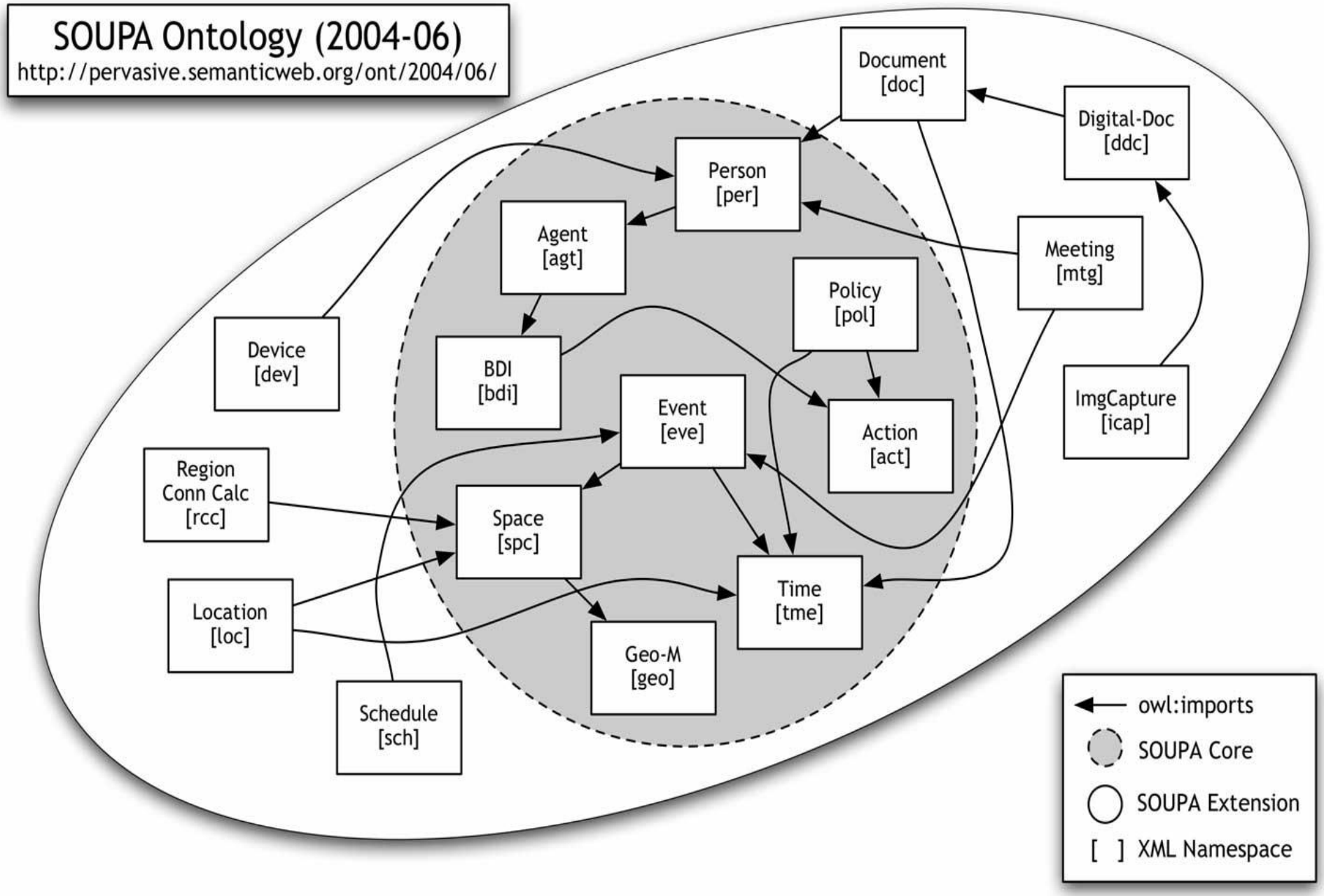
The broker informs the projector agent



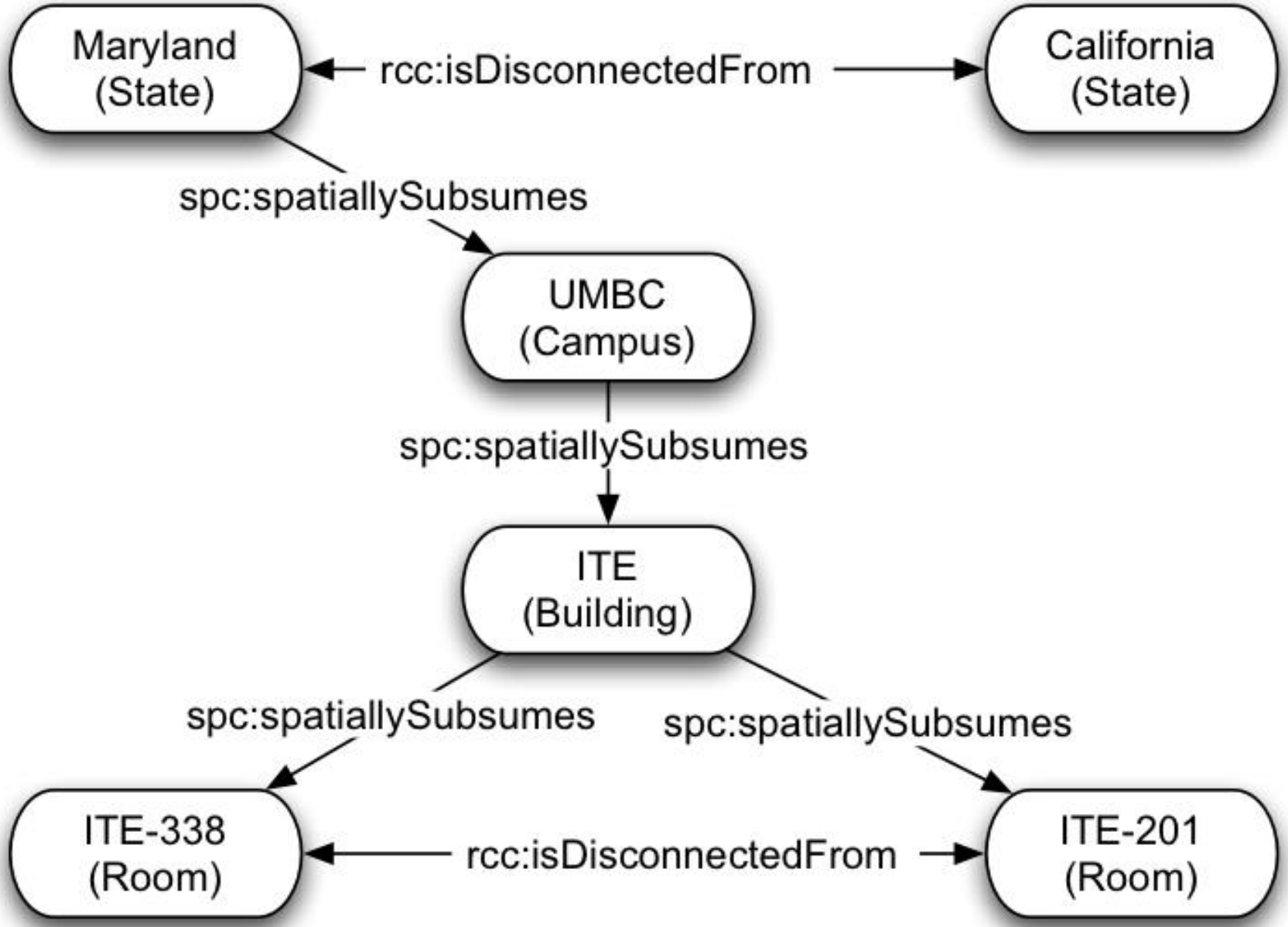
The projector agent sets up the slides



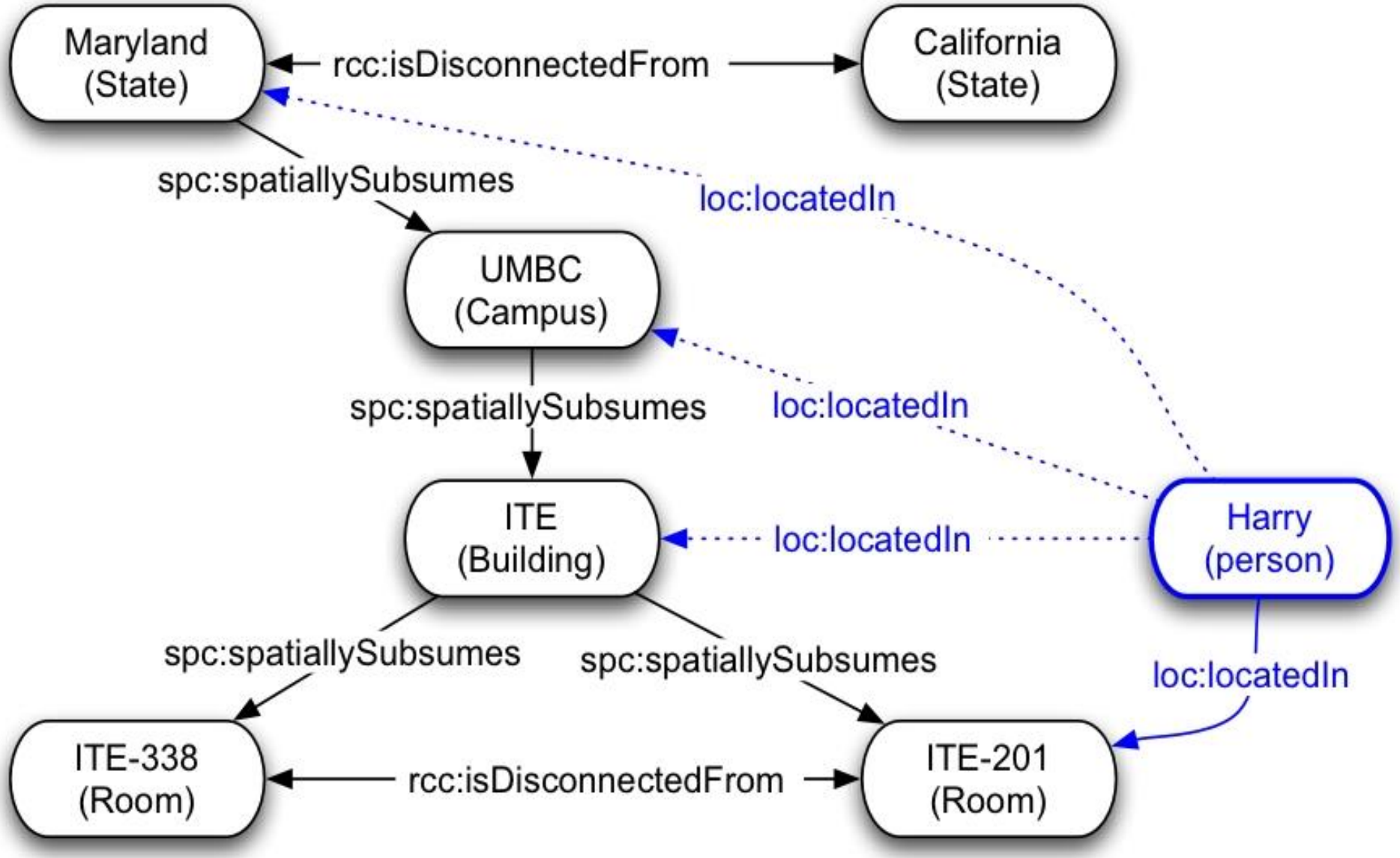
SOUPA Ontology provides common vocabulary



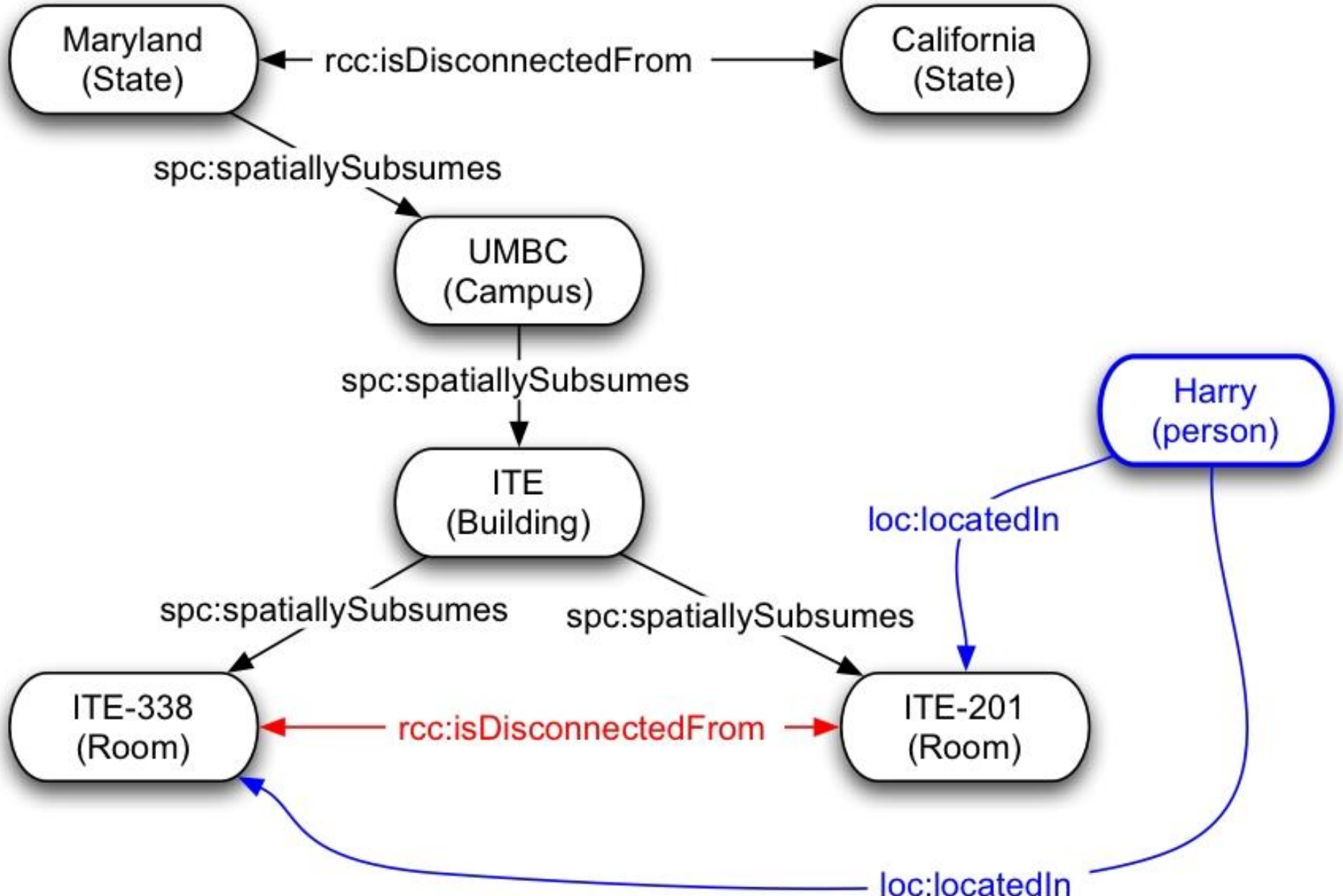
A Simple Spatial Model of UMBC



Where's Harry?



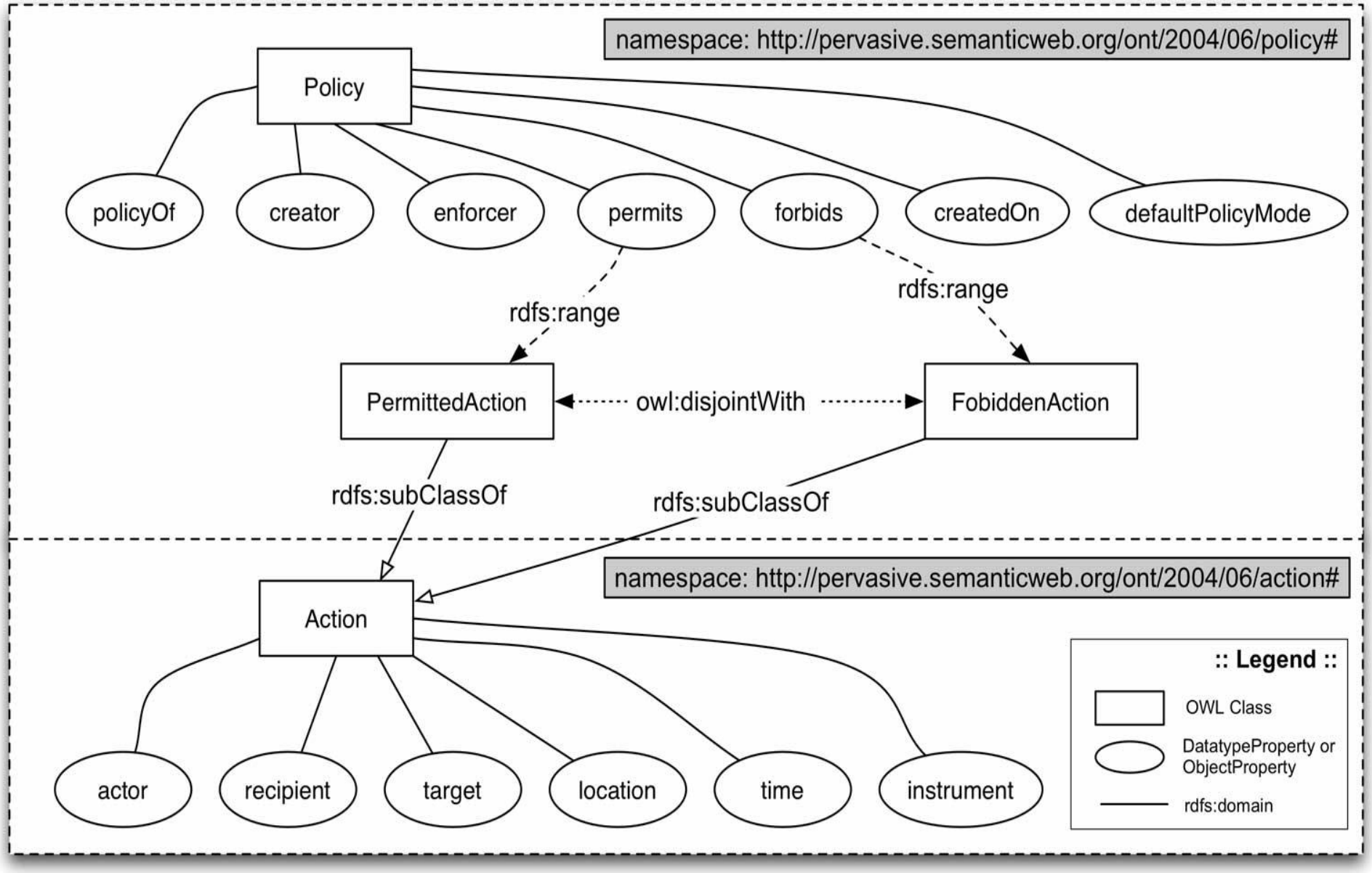
Detecting Inconsistencies



Privacy Protection in CoBrA

- Users define policies to permit or prohibit the sharing of their information
  - Policies are provided by personal agents or published on web pages
  - and use the SOUPA ontologies as well as other SW assertions (e.g., FOAF, schedules)
- The context broker follows user defined policies when sharing information, unless contravened by higher policies

The SOUPA Policy Ontology



Policy Reasoning Use Case

- The speaker doesn't want others to know the specific room that he's in, but is willing for others to know he's on campus
- He defines the following privacy policy
  - Share my location with a granularity >= "State"
- The broker
  - isLocated(US) => Yes!
  - isLocated(Maryland) => Yes!
  - isLocated(UMBC) => Uncertain..
  - isLocated(ITE-RM210) => Uncertain..

What we learned

- FIPA and OWL were good for integrating disparate components
- Even when some of these were running on cell phones!
- OWL made it easy to mix content from different ontologies unambiguously
- The use of OWL made it easy to take advantage of information published in XML on the web
  - e.g., foaf information, privacy policy

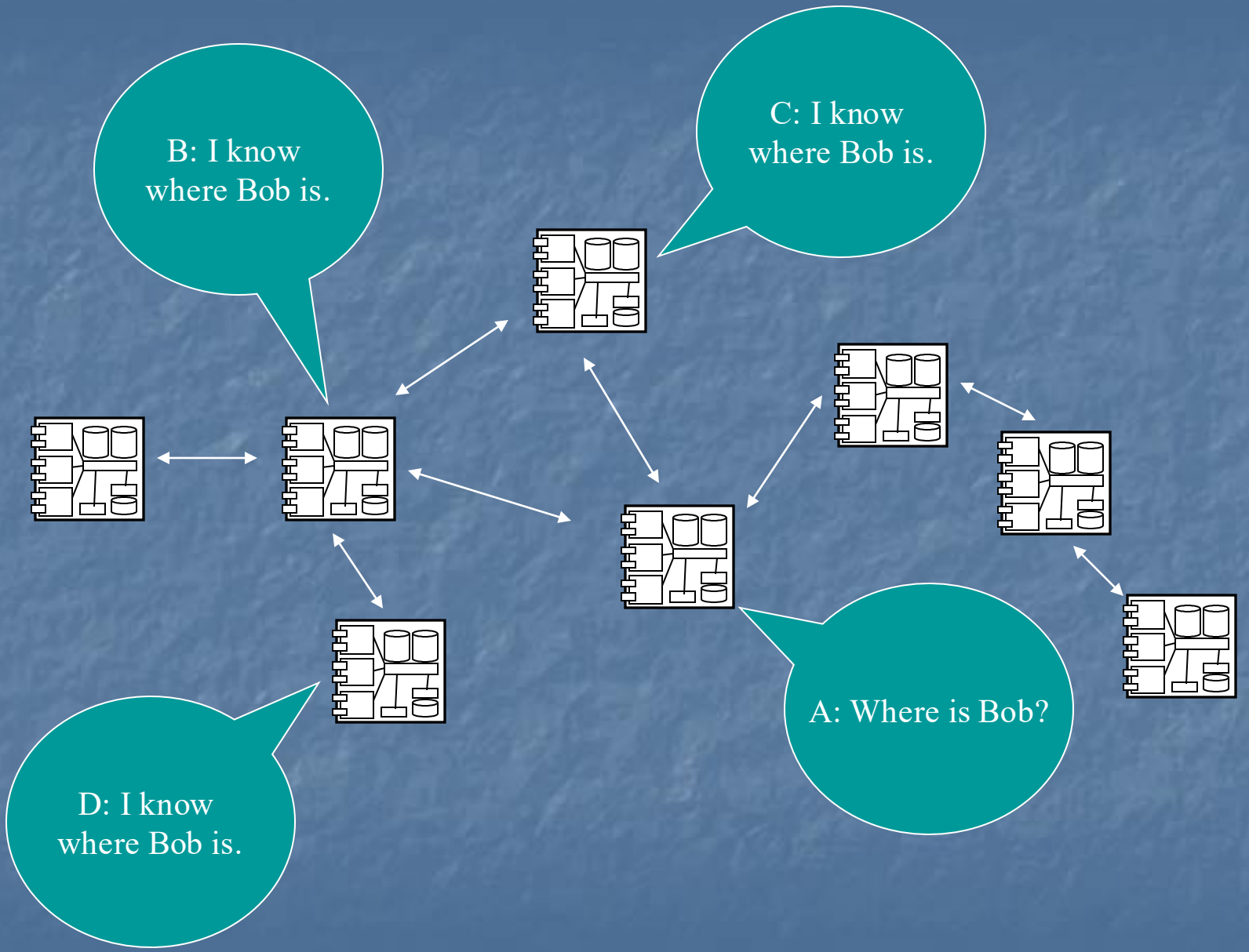
(4) TIVO for Mobile Computing

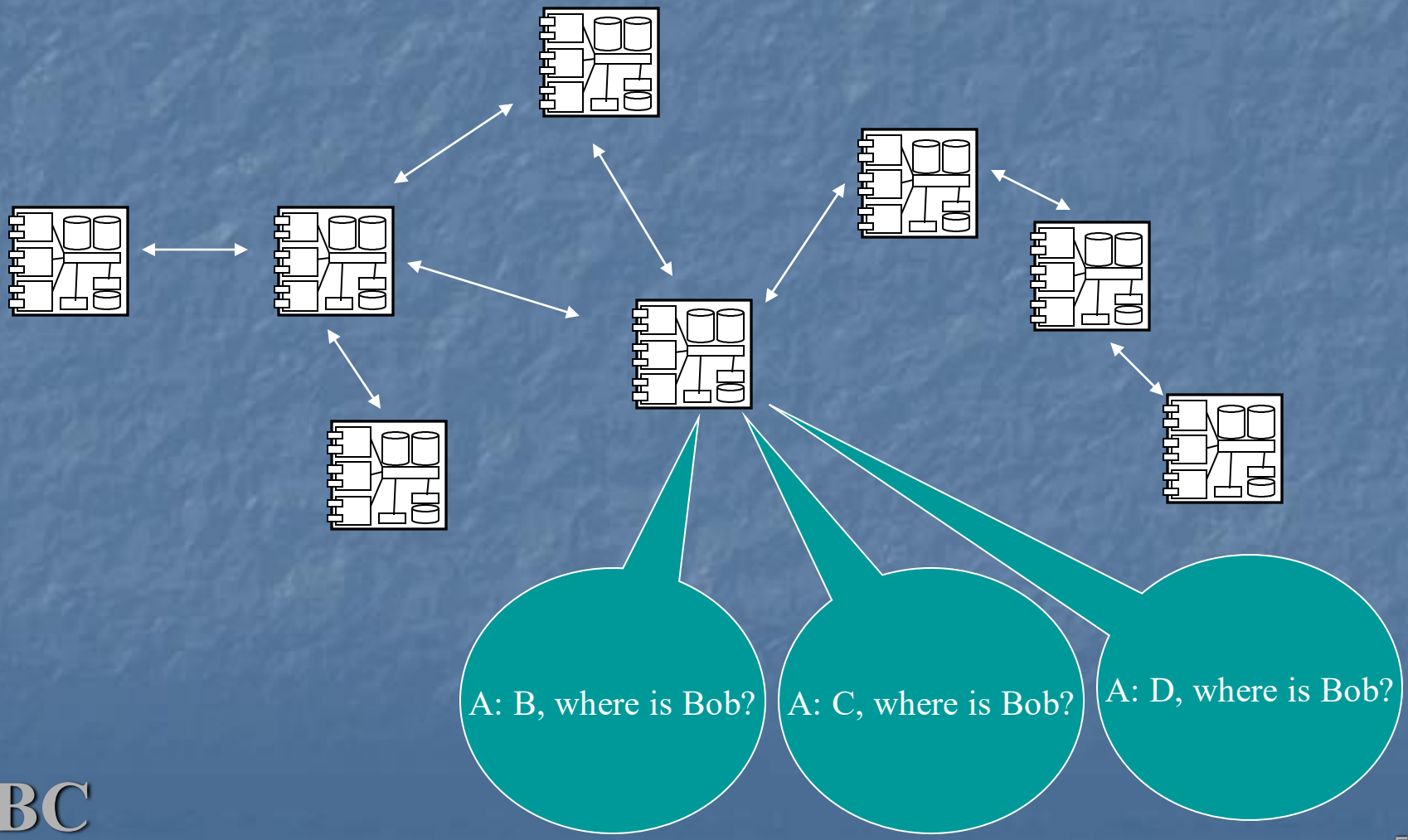
A mobile computing vision and a problem

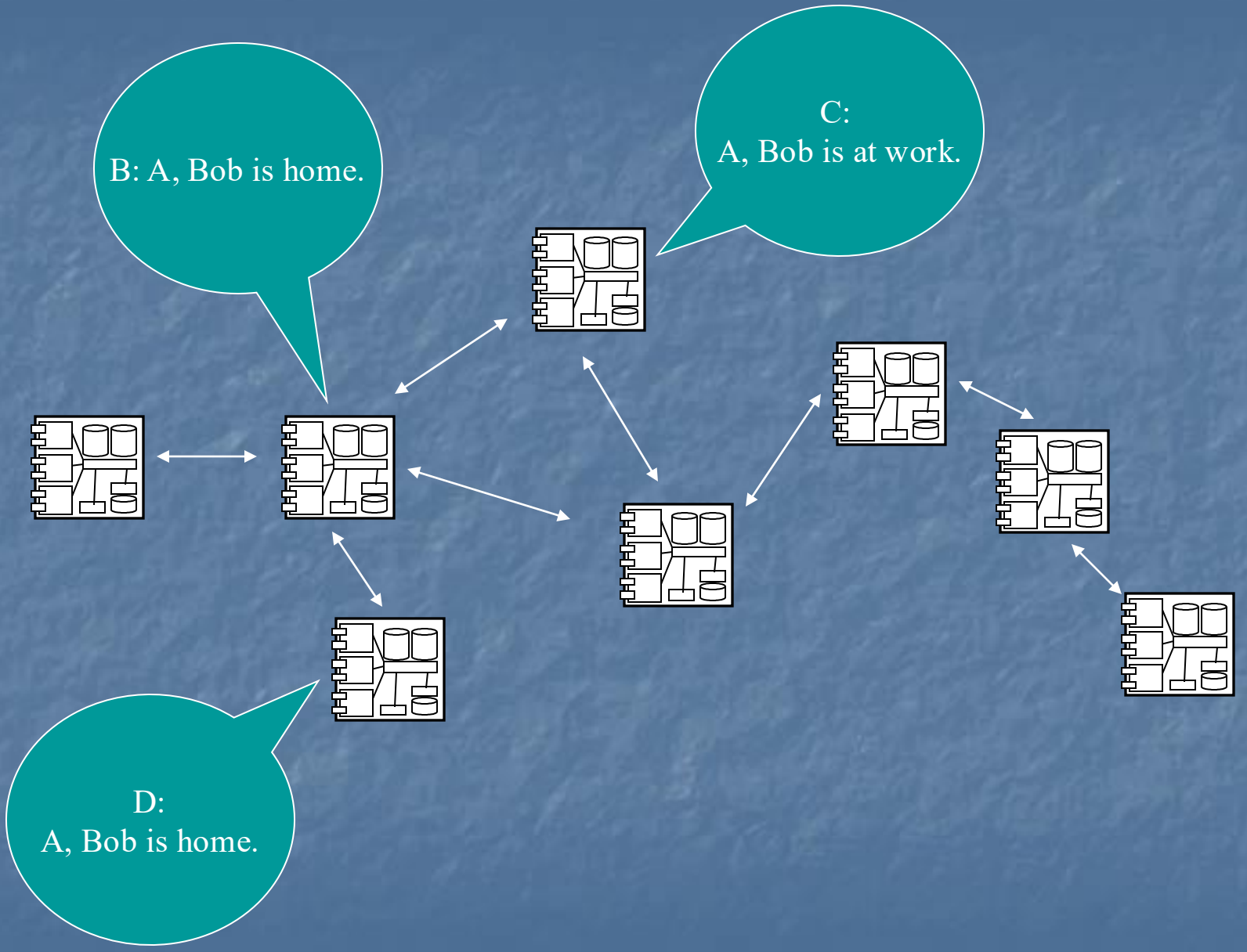
- Devices "broadcast" information and service descriptions via short-range RF (802.11, Bluetooth, UWB, etc.)
- As people and their devices move, they can access this data, but only while it's in range
  - The data may be out of range when it's needed
- Devices must anticipate their information need so they can cache data when it's available
  - Based on user model, preferences, schedule, context, trust, …
  - Compute a dynamic utility function to create a "semantic" cache replacement algorithm

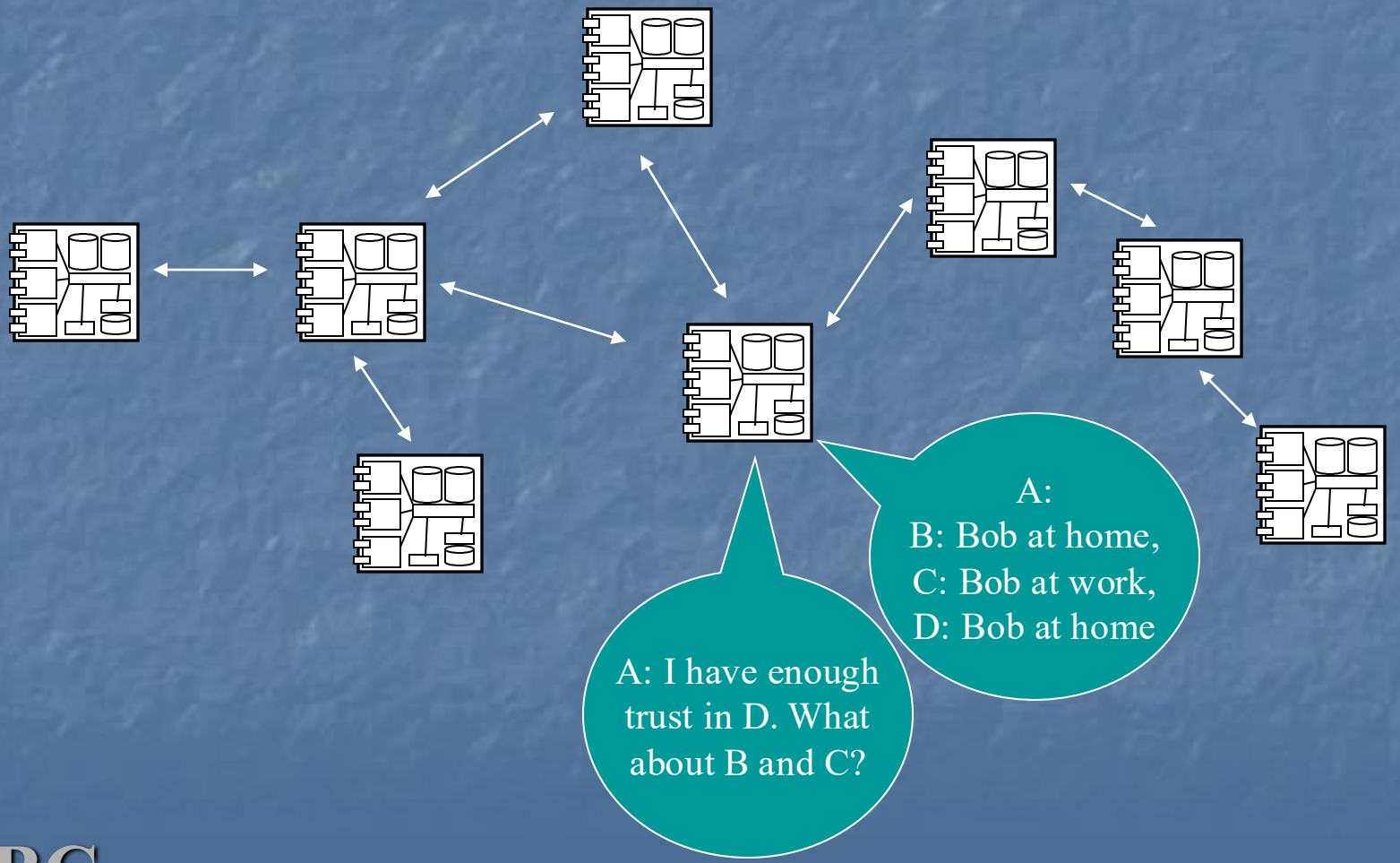
MoGATU's distributed belief model

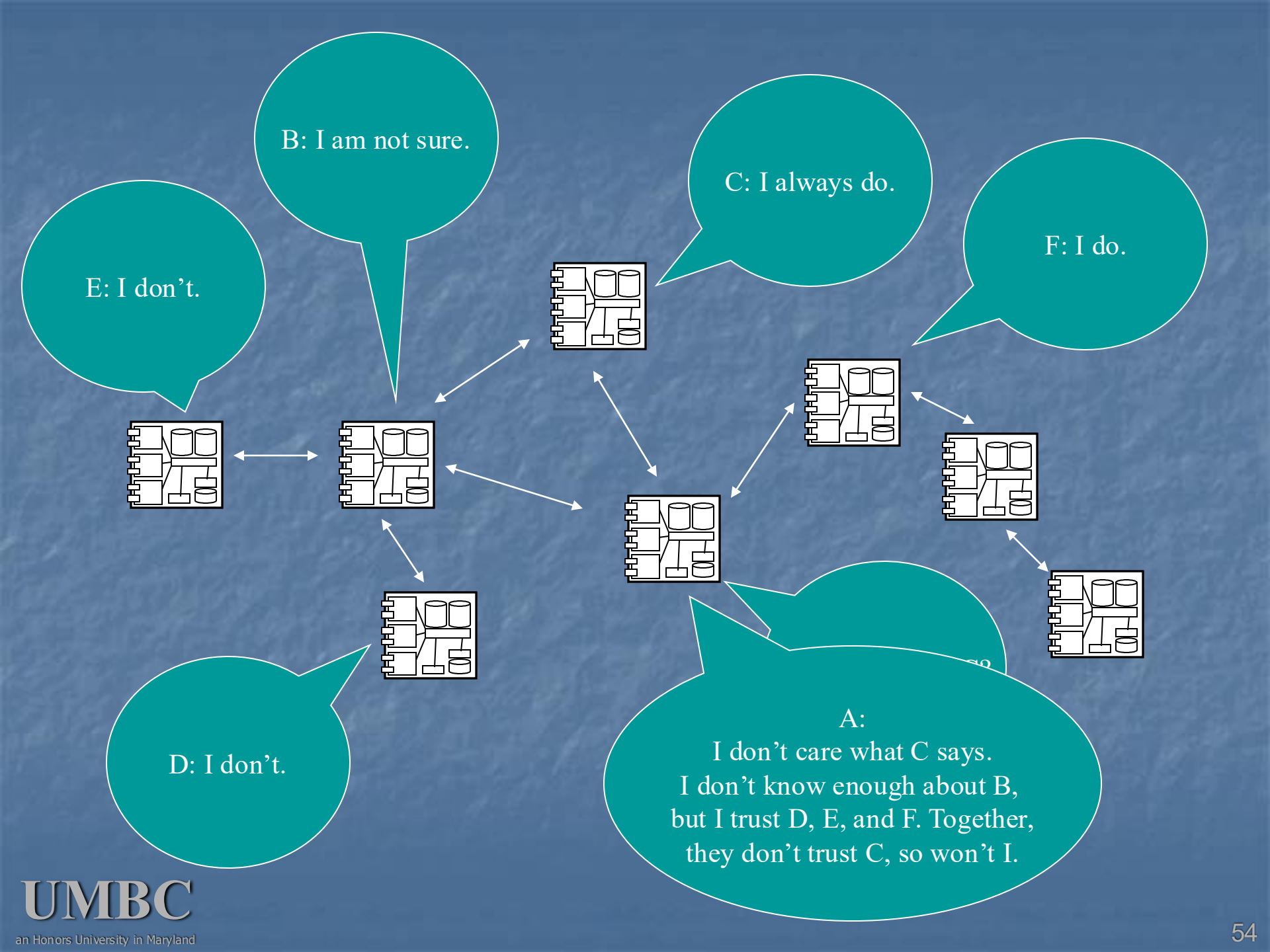
- MoGATU is a data management module for MANETs
- Devices send queries to peers
  - Ask its vicinity for reputation of untrusted peers that responded -- trust a device if trusted before or if enough trusted peers trust it
- Use answers from (recommended to be) trusted peers to determine answer
- Update reputation/trust level for all responding devices
  - Trust level increases for devices giving what becomes final answer
  - Trust level decreases for devices giving "wrong" answer
- Each devices builds *a ring of trust*…

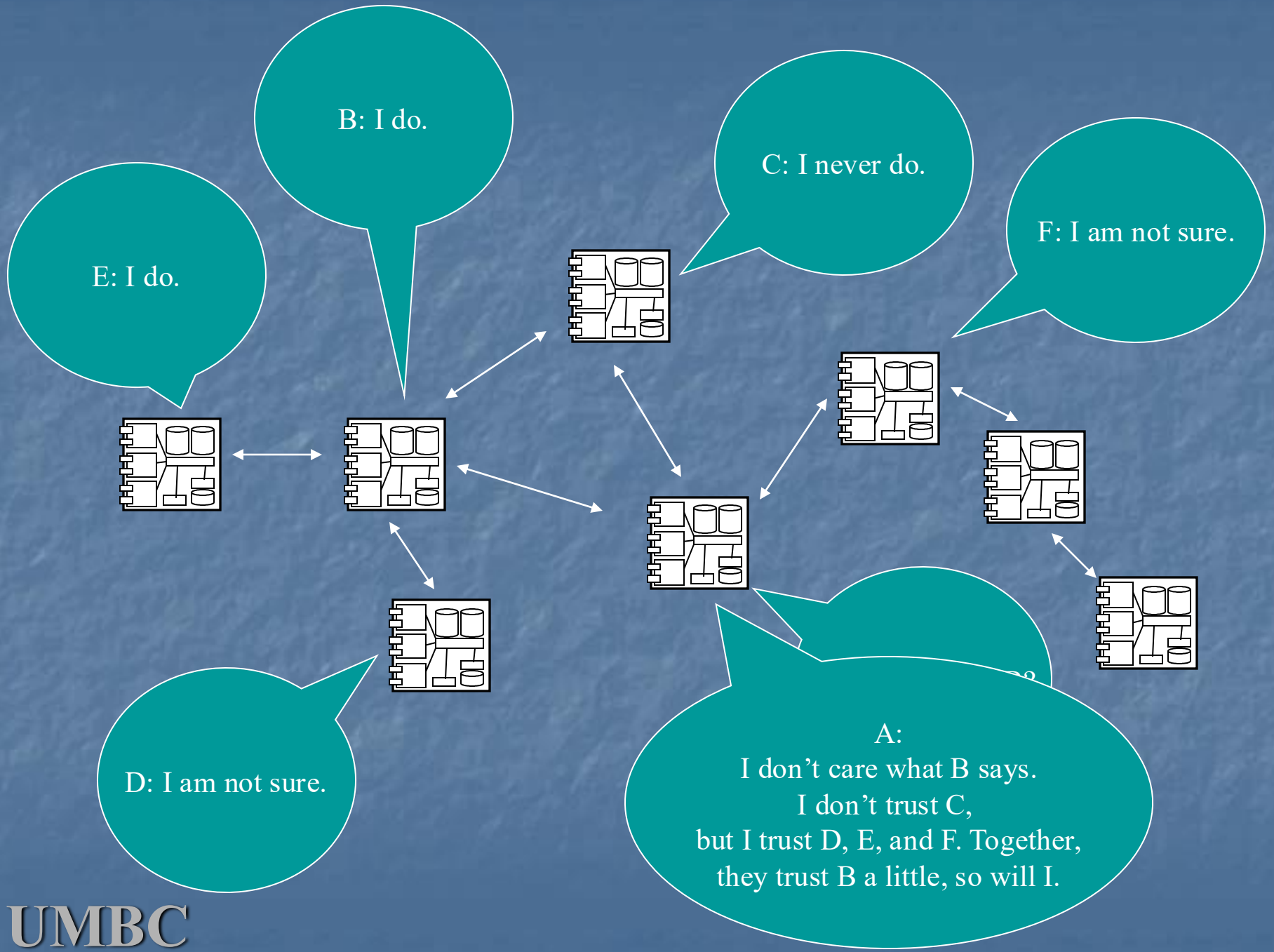

B: I know where Bob is.
C: I know where Bob is.
A: Where is Bob?
D: I know where Bob is.

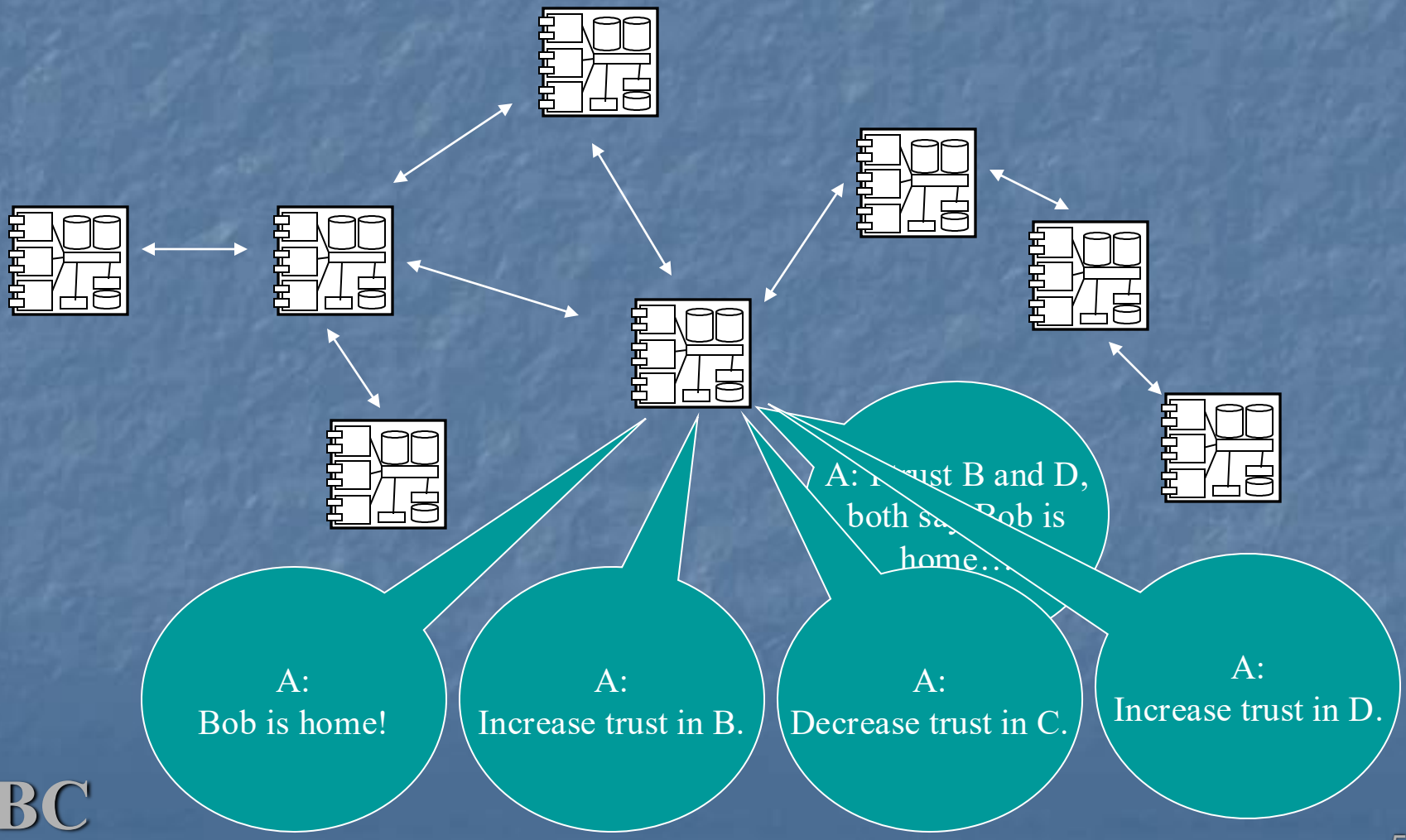

A: B, where is Bob?
A: C, where is Bob?
A: D, where is Bob?


B: A, Bob is home.
C:
A, Bob is at work.
D:
A, Bob is home.


A:
B: Bob at home,
C: Bob at work,
D: Bob at home
A: I have enough trust in D. What about B and C?


B: I am not sure.
C: I always do.
F: I do.
E: I don't.
D: I don't.
A:
I don't care what C says.
I don't know enough about B,
but I trust D, E, and F. Together,
they don't trust C, so won't I.


B: I do.
C: I never do.
E: I do.
F: I am not sure.
D: I am not sure.
A:
I don't care what B says.
I don't trust C,
but I trust D, E, and F. Together,
they trust B a little, so will I.

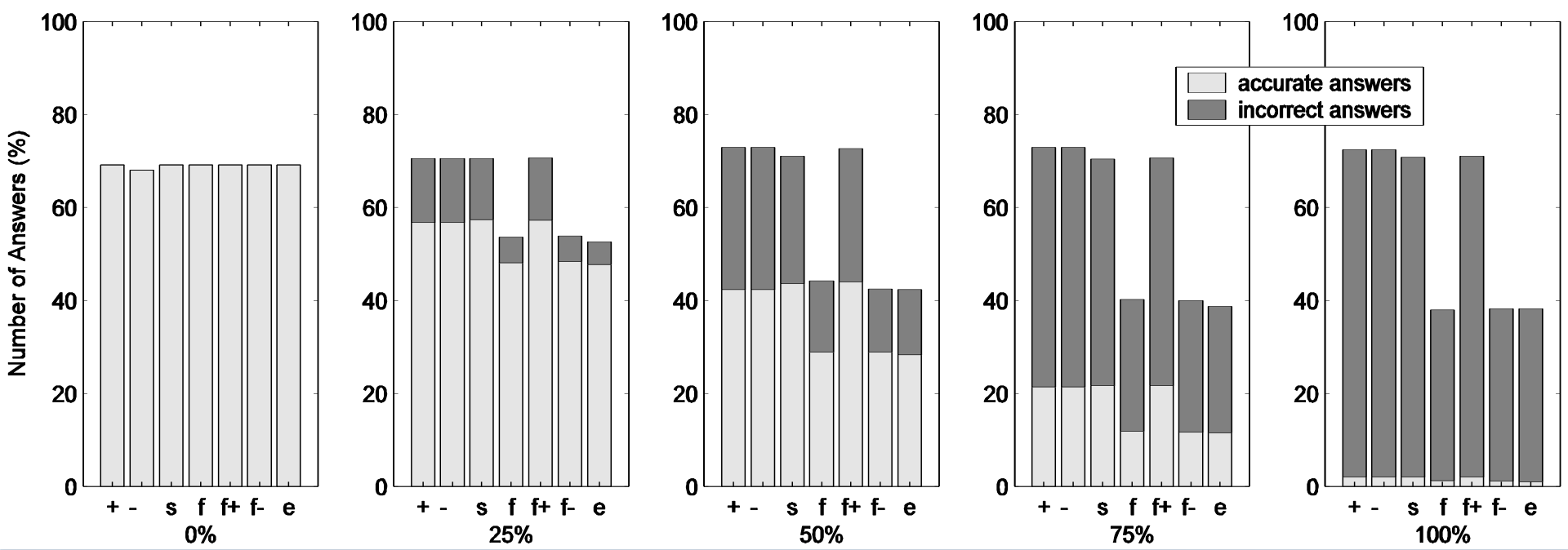

A:
Bob is home!
A:
Increase trust in B.
A:
Decrease trust in C.
A:
Increase trust in D.

Experimental results

- We are investigating the design of algorithms for these data management problems in MANETs via simulations for varying parameters
- For example
  - Answer accuracy vs. trust learning functions
  - Answer accuracy vs. accuracy merging Functions
  - Distrust convergence vs. dishonesty level

Answer Accuracy vs. Trust Learning Functions

- The effects of trust learning functions with initial optimistic trust in environments varying in dishonesty.
- The results are shown for Δ_{++}, Δ_{--}, Δ_{s}, Δ_{f}, Δ_{f+}, Δ_{f-}, and Δ_{exp} learning functions.
- For more results, see http://ebiquity.org/



What we learned

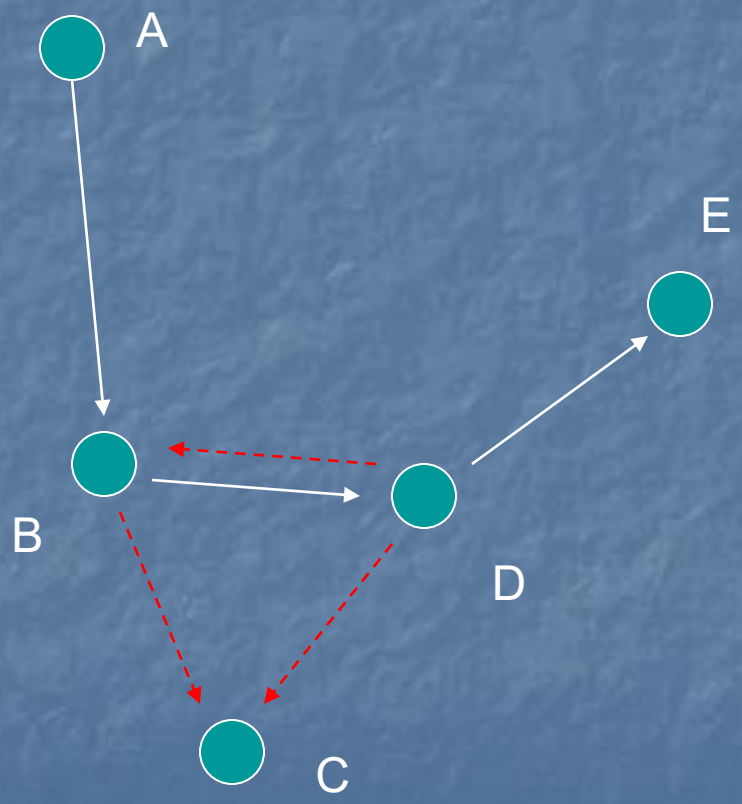
- OWL was a good language for capturing user profiles and the simple BDI models we needed
- Any of several simple trust models increase the accuracy of information
  - Designing a good trust model depends on the MANET assumptions
  - As well as the level of cooperation and honesty
- Trading reputation information boosts the performance of the algorithms

(5) Security in P2P Systems

- Peer-to-peer systems are manifest at multiple levels, such as ad hoc networking, file-sharing applications, and multiagent systems,
- Recognizing "bad actors" in P2P systems is hard
  - Bad actors might be having trouble, incompetent, uncooperative, or malicious
- Ad Hoc networks can be subverted by the introduction of malicious nodes
  - E.g.: blackhole routers that do not forward packets
- MANETS offer additional challenges

Neighborhood Watch in ad hoc networks

- Node A sends packet destined for E, through B & D.
- When B → D, B and C make snoop entry (A,E,Ck,B,D,*E*).
- B and C check if D forwarded the packet or dropped, altered, or misrouted it.



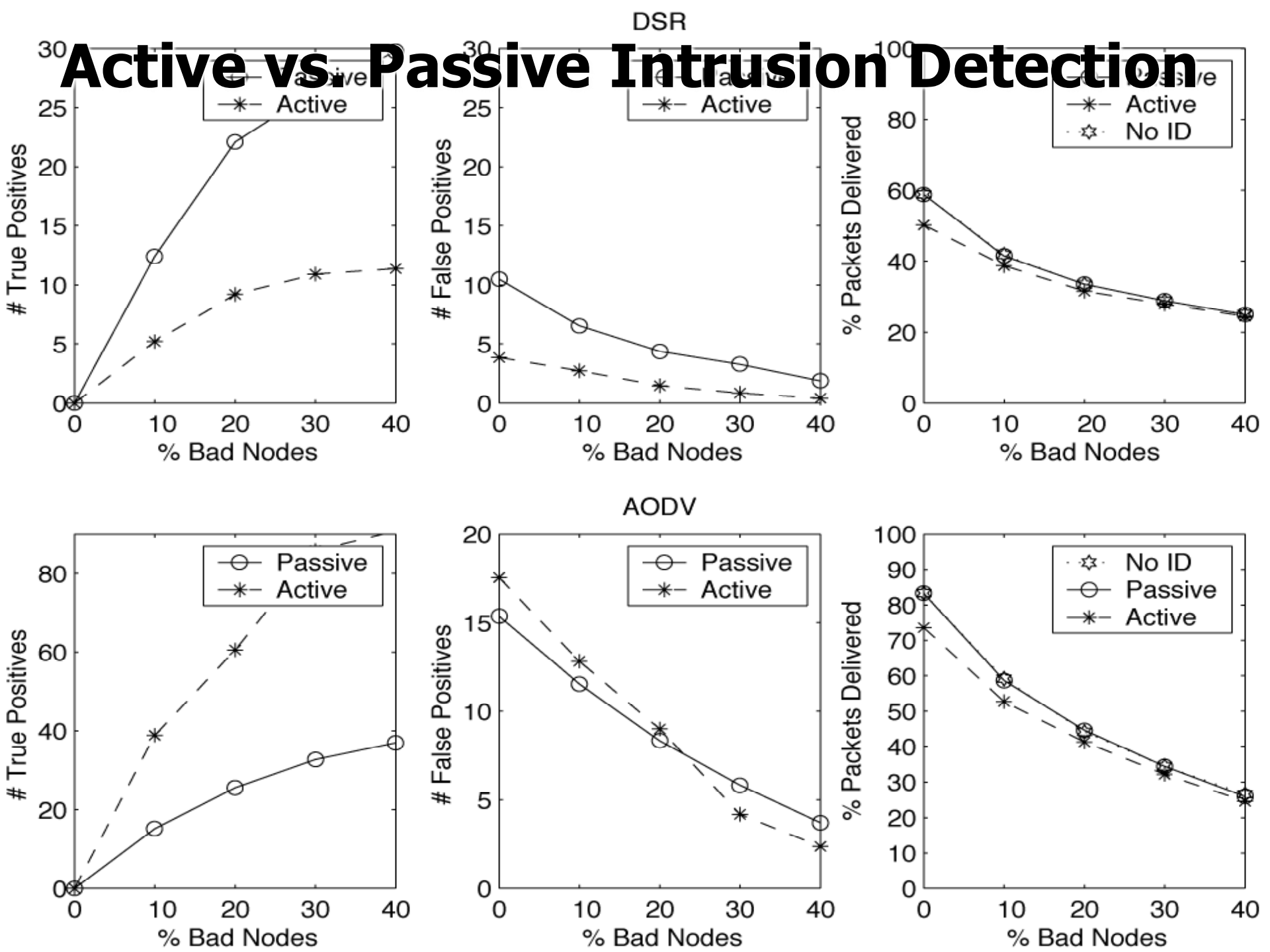
Simulation in GlomoSim

- We compared two MANET intrusion detection schemes
  - **Passive Intrusion Detection:** each node builds and maintains a trust model of each of its neighbors
  - **Active Intrusion Detection:** nodes form clusters, build a trust models and share them
- For different routing protocols (DSR and AODV)
- Using various Glomosim parameters
  - 16 nodes communication
    - 4 nodes sources for 2 CBR streams
    - 2 nodes pair CBR streams
  - Mobility 0 – 20 meters/sec
  - Pause time 0 – 15s
  - No bad nodes

Results

- Given the noisy data inherent in MANETS, false positives are a serious problem
  - Packets are lost, nodes move out of range, etc.
- Active intrusion detection reduced the rate of false positives at the cost of additional throughput reduction
  - **Passive ID**
    - false alarms > 50%
    - throughput rate decrease ~ 3%
  - **Active ID**
    - false alarms < 30%
    - throughput rate decrease ~ 25%

Active vs. Passive Intrusion Detection



What we learned

- A neighborhood watch algorithm can be used to detect bad actors in mobile ad hoc networks
- Using an active algorithm decreases false positive at a modest decrease in throughput
  - Social reputation collected via voting is effective
- Other techniques can be added, such as random anonymous audits
  - Which techniques are applicable depends on the communication assumptions (e.g., can messages be "overheard"?)

Conclusions & final thoughts

(1) How do SW languages fit into current agent technology?

(2) SW and might be a chance for us to get some AI out of the lab

(3) Requirements for pervasive computing applications motivate agents and the SW

(3) How do we get there

How does SW fit into agent frameworks?

- Most multiagent systems work assumes some kind of infrastructure to support discovery, communication, cooperation, etc.
  - Cougaar, KQML+KIF, FIPA, CoABS Grid, FIPA, etc.
- We found it easy to fit RDF & OWL into FIPA's framework and should fit well into others*
- Big wins: (1) open, distributed, published ontologies, (2) easy mixing of vocabulary, (3) adoption path from RDF to OWL-lite to OWL to …, (4) web standards compatible

** Your mileage may vary*

Rethinking the agent communication paradigm

- Much multi-agent systems work is grounded in Agent Communication Languages (e.g., KQML, FIPA) and associated software infrastructure.
  - This paradigm was articulated ~1990, about the same time as the WWW was developed.
  - Our MAS approach has not yet left the laboratory yet the Web has changed the world.
- Maybe we should try something different?
  - The MAS communication paradigm was inherited from client-server systems -- message oriented communication mediated by middleware

Rethinking the agent communication paradigm

As with people, messaging shouldn't be the only way

- Agents "publish" beliefs, requests, and other "speech acts" on web pages.
- Brokers "search" for and "index" published content
- Agents "discover" what peers have published on the web and browse for more details
- Agents "speak for" content on web pages by
  - Answering queries about them
  - Accepting comments and assertions about them

NLP text:people :: SW text:agents

Swoogle is a crawler based search an retrieval system for semantic web documents currently under development

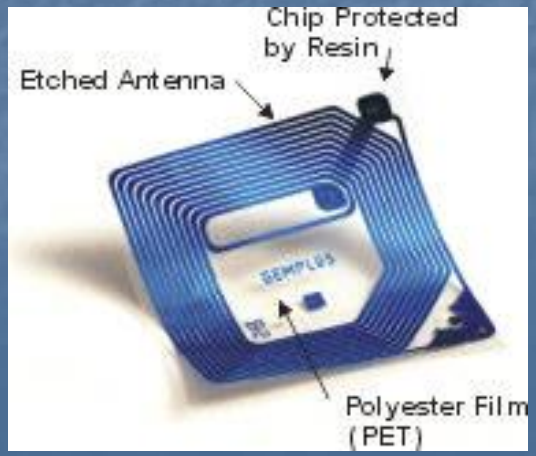
The symbol grounding problem


MIT's Cog

- An argument against human-like AI is that it's impossible unless machines share our perception of the world.
- A solution to this "symbol grounding problem" is to give robots with human inspired senses.
- But the world we experience is determined by our senses, and human and machine bodies may lead to different conceptions of the world (e.g. Nagel's *What Is It Like To Be a Bat?*)
- Maybe the Semantic Web is a way out of this problem?

Solving the symbol grounding problem

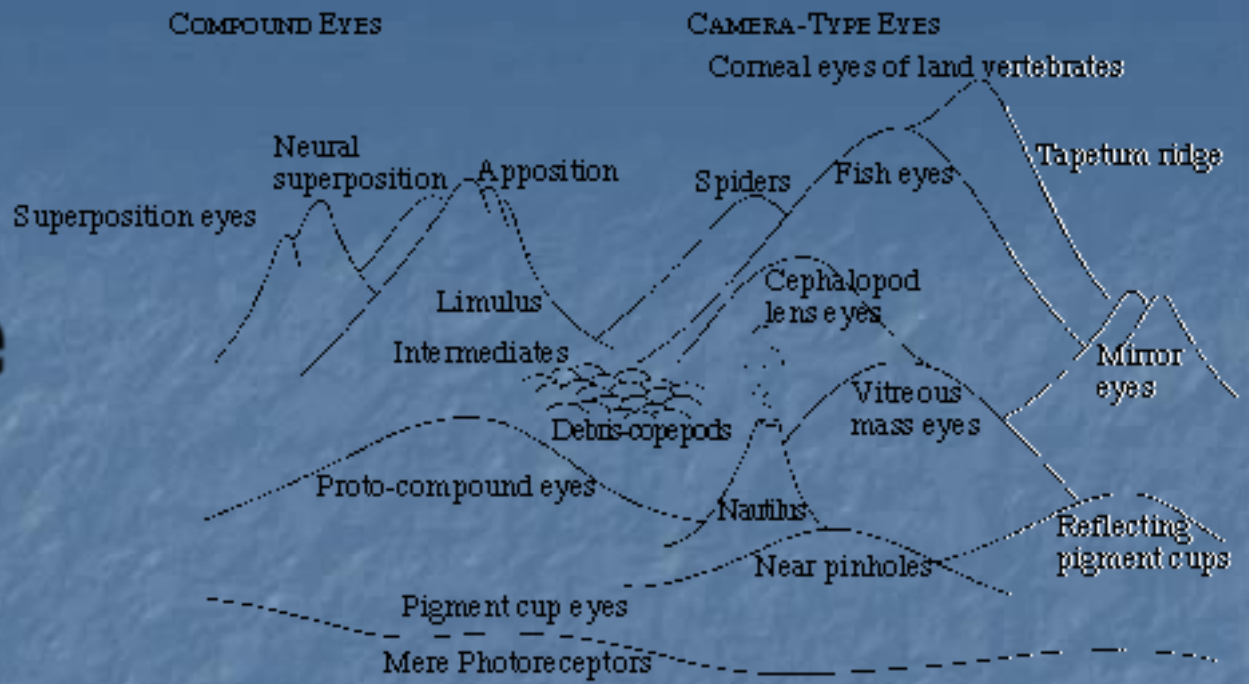
- The web may become a common world that both humans and machines can understand.
- Confession: the web is more familiar and real to me than much of the real world.
- Physical objects can be tagged with low cost (e.g., $0.05) transponders or RFIDs encoding their URIs
  - See HP's Cooltown project http://cooltown.com/



How do we get there from here?

- This semantic web emphasizes ontologies – their development, use, mediation, evolution, etc.
- It will take some time to really deliver on the agent paradigm, either on the Internet or in a pervasive computing environment.
- The development of complex systems is basically an evolutionary process.
- Random search carried out by tens of thousands of researchers, developers and graduate students.

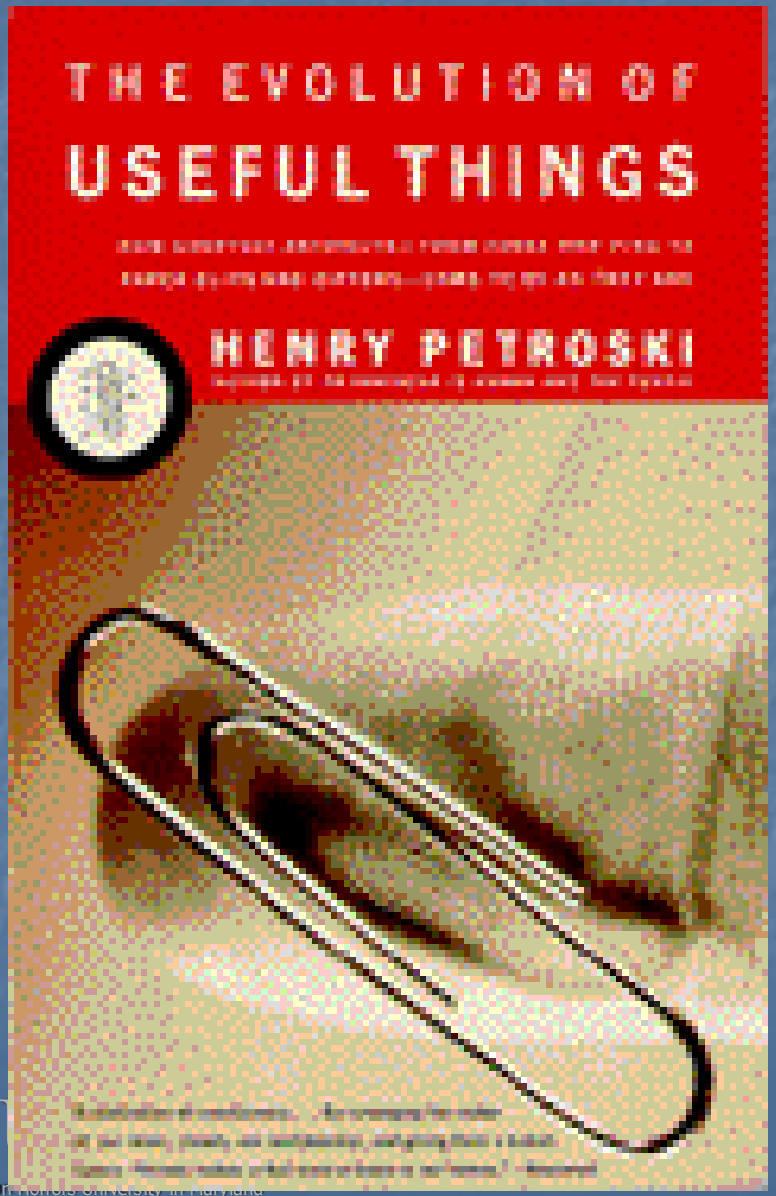
Climbing Mount Improbable



"The sheer height of the peak doesn't matter, so long as you don't try to scale it in a single bound. Locate the mildly sloping path and, if you have unlimited time, the ascent is only as formidable as the next step."

-- Richard Dawkins, *Climbing Mount Improbable*, Penguin Books, 1996.

T.T.T: things take time



- Prior to the 1890's, papers were held together with straight pens.
- The development of "spring steel" allowed the invention of the paper clip in 1899.
- It took about 25 years (!) for the evolution of the modern "gem paperclip", considered to be optimal for general use.

So, we should …

- **Start with the simple and move toward the complex**
  - E.g., from vocabularies to FOL theories
- **Allow many ontologies to bloom**
  - Let natural evolutionary processes select the most useful as common consensus ontologies.
- **Support diversity in ontologies**
  - Monocultures are unstable
  - There should be no THE ONTOLOGY FOR X.
- **The evolution of powerful, machine readable ontologies will take many years, maybe generations**
  - Incremental benefits will more than pay for effort

Biquity Research Group - Mozilla

ebiquity group

Tuesday, July 20, 2004, 10:22:24 EDT

ABOUT US RESEARCH PEOPLE PUBLICATIONS NEWS PHOTOS EVENTS CONFERENCES INTERNAL

For more information

http://ebiquity.umbc.edu/

About Us
Research
Publications
News
Photos
Blogger

eBiquity Research Group | About Us | Contact Us | Site Map | Legal | Privacy

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Page generated in 0.000 seconds.

Annotated in OWL