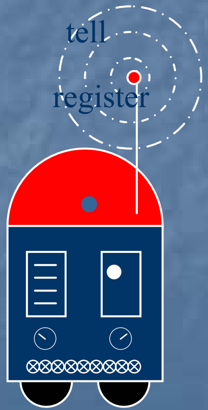
Security related research at UMBC

Tim Finin

University of Maryland Baltimore County

January 2002

Joint work with Anupam Joshi, Yun Peng, Scott Cost, Yelena Yesha and many students.



This work was partially supported by DARPA contract F30602-97-1-0215 and NSF grants CCR007080 and IIS9875433.

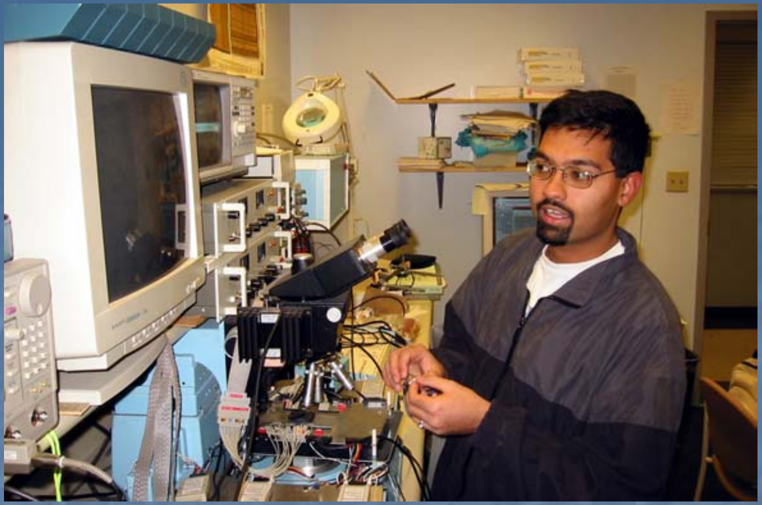
Overview

- What is UMBC?
- UMBC Center for Information Security and Assurance (CISA)
- Maryland Adaptive and Intelligent Systems (MAISE)
- Distributed trust research
- Comments and conclusions

What is UMBC

- The University of Maryland Baltimore County is
  - One of the three research campuses in the University of Maryland System
  - Ranked in top tier of nation's research universities-- *Doctoral/Research Universities-Extensive* -- by the Carnegie Foundation
- Has 500 full time and 335 part time faculty, 10K undergraduate and 2K graduate students
- Located in suburban Baltimore County, between Baltimore and Washington DC.
- Special focus on science, engineering, information technology and public policy with ~$75M in external funding for research and development

IT @ UMBC



- **Information Technology has UMBC's largest concentration of faculty and students**
  - Over 100 faculty and more than 3200 students
- Degree programs (graduate and undergraduate)
  - Computer Science, Computer Engineering, Information Systems, Electrical Engineering, and Digital Imaging.
- Certificate and training programs (degree and non-degree)
  - Electronic commerce, Information Security, Web Development, Systems Administration, Oracle, CISCO, …
- Many institutes and centers
  - Center for Women and Information Technology, Institute for Global Electronic Commerce, Center for Information Security and Assurance, Center for Photonics, …

CISA

- **UMBC Center for Information Security and Assurance**
- Information Assurance (IA) encompasses the scientific, technical, and management disciplines required to ensure computer and network security.
- CISA is an interdepartmental center promoting research and education in information security and assurance.
  - Involves the Departments of Computer Science and Electrical Engineering, Information Systems, Policy Science, and Mathematics
- In 2001, The US National Security Agency (NSA) designated UMBC as a *Center of Academic Excellence in Information Assurance Education*.

Security Research

- **UMBC faculty and students are doing research in many security areas.**
- Cryptography and cryptanalysis
- Key management protocols and systems
- Quantum computing and cryptography
- Intrusion detection, isolation and containment
- Intrusion tolerant database systems
- Survivable Internet Scale Information Systems
- Security for mobile and pervasive computing environments
- Security for agent based systems
- …

Security Education

- **UMBC offers a variety of courses on security related topics.**
- Professional certificate training programs on information assurance.
- Graduate and undergraduate courses on all aspects of security, information assurance and cryptography.
- We are currently building a new "hacking lab" which will give students hands on experience in security vulnerabilities and how to prevent them.

Ebiquity Group

- **The UMBC Ebiquity group is focused on building intelligent, adaptive systems for mobile and pervasive environments.**
- Communication technologies: Bluetooth, 802.11b, CDMA, IR.
- Software technologies: Java, Jini, web, semantic web, logic programming, FIPA, …
- Software architectures: multi-agent systems, web centric systems
- Applications: for mobile and pervasive computing

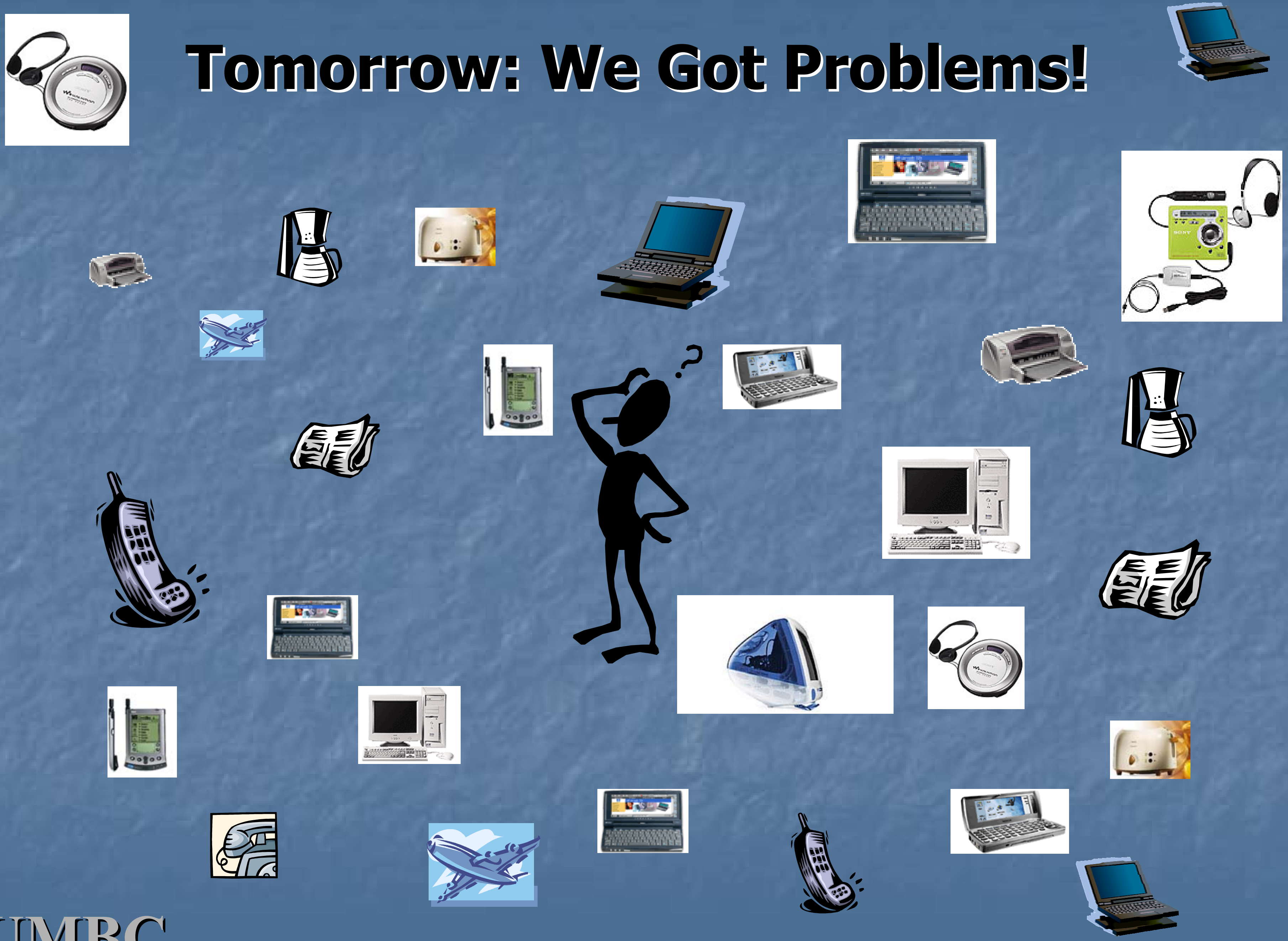
MAIS

- **The Maryland Adaptive and Intelligent Systems group has 11 faculty.**
- **Faculty:** Finin, Nierenberg, Nicholas, Peng, Joshi, Kalpakis, Kargupta, desJardins, Oates, Yoon, Cost
- **Interests:** machine learning, data mining, genetic algorithms, agent-based systems, ontologies, information retrieval, machine translation, semantic web.
- **Applications:** ecommerce, bioinformatics, document translation, expert systems, intrusion detection, web mining, etc.

What Motivates Us

Today: Life is Good.

Tomorrow: We Got Problems!



Our Conceptual Tools

Open Systems and Agents

- Agents offer relatively advanced approaches to many of the problems faced by modern “open systems” such as the web, mobile computing, and pervasive computing. All assumed an open, dynamic, ad hoc environment.
- Common issues:
  - Service description, discovery, composition.
  - Negotiation for services and information
  - Authentication, authorization, and trust
  - Delegation and degrees of autonomy
  - Coordination and teamwork models

Semantic Web

- *"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."* -- Berners-Lee, Hendler and Lassila, The Semantic Web, Scientific American, 2001
- The semantic web provides a good approach, language and tools to support agents as well as mobile and pervasive computing.
  - This isn't obvious, since the SW seems grounded in the "traditional" wired web.
  - But, the principles which drive it are the right ones for the "open systems" envisioned by agents as well as pervasive computing.

Recent Work

Some UMBC Work

Here are some ongoing projects involving mobile/pervasive computing and the semantic web.

(1) Centaurus communication infrastructure

(2) Enhancing Jini with DAML for service description and discovery

(3) Enhancing Bluetooth's SDP with DAML

(4) A model of distributed authorization and trust

(5) Agents2go -- a simple mobile application

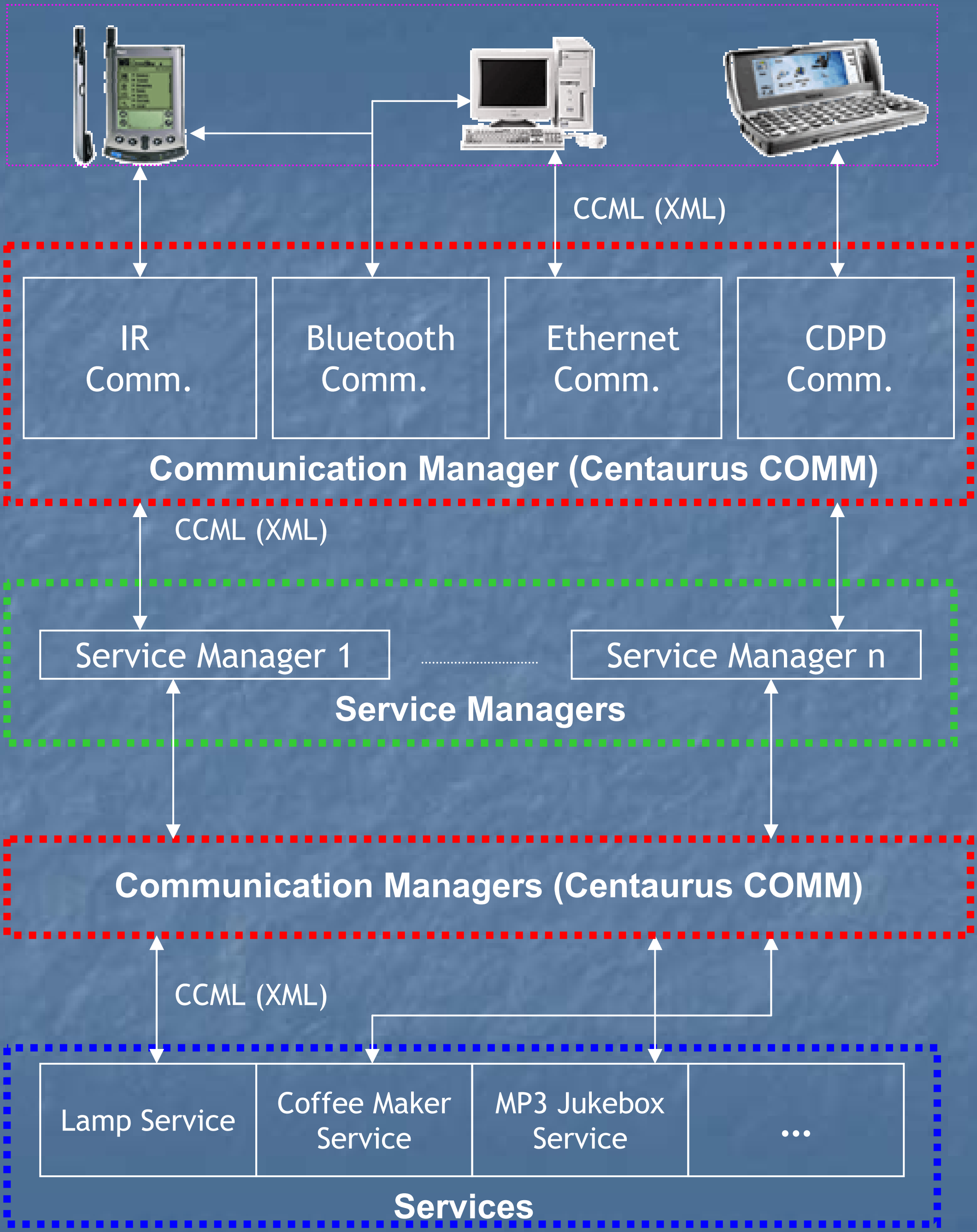
(6) ITtalks – a semantic web application

See http://research.ebiquity.org/

(1) Centaurus

Centaurus Communication (Centaurus COMM) provides a message passing network architecture that allows heterogeneous devices to communicate through varied communication mediums in a uniform fashion

Runs on PDAs and other small devices



UMBC
an Honors University in Maryland

(2) Enhancing Jini's registration server

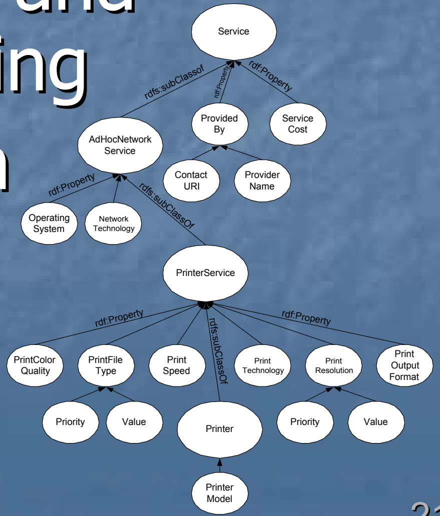
- Jini is Sun's technology for building "self describing and self organizing" distributed systems.
- Jini is a very attractive collection of ideas and components.
- One deficiency is the Jini registration server's inexpressive approach to describing services offered and sought.
- We've produced a modified Jini registration server which allows agents to use DAML+OIL to describe services offered or sought
- Supports reasoning during matching (e.g., constraint satisfaction)

(3) Enhancing Bluetooth's SDP

- Bluetooth is a short-range RF wireless technology that supports ad-hoc networks and uses P2P protocols.
- Bluetooth Service Discovery Protocol:
  - Simple service discovery mechanism
  - Services and attributes represented by UUIDs
  - UUID-based matching (128 bit number!)
  - No registration, aggregation, multicasting, event notification
- Not very expressive!

Prototyped Solution

- Assume Bluetooth ad-hoc networks with at least one resource rich device (e.g., each room has a facilitator).
- Enhanced SDP
  - Services and attributes described in DAML using a "standard" ontology
  - All available information from service and attribute descriptions used for matching
  - Tries to obtain *closest* possible match
  - Support service registration facility



UMBC
an Honors University in Maryland

(4) Delegation Based Model for Distributed Trust

- We are developing a delegation based model for distributed authorization and trust for use in both wired and wireless scenarios.
- Focus on trust from a "security perspective"
- Building on concepts like authentication, authorization, role-based access control, public key infrastructure, digital signatures, authoritative sources of information, etc.
- Agents make speech acts about and reason over these properties and relations.
- Grounded in an ontology represented in DAML

What is Distributed Trust

- Issues
  - No central authority
  - *logging in* is not possible
  - Access control for entities never encountered before
- We use *Distributed Trust* to solve these issues
- trust = policies + credentials + delegation actions + proofs of deontic properties

Some Scenarios

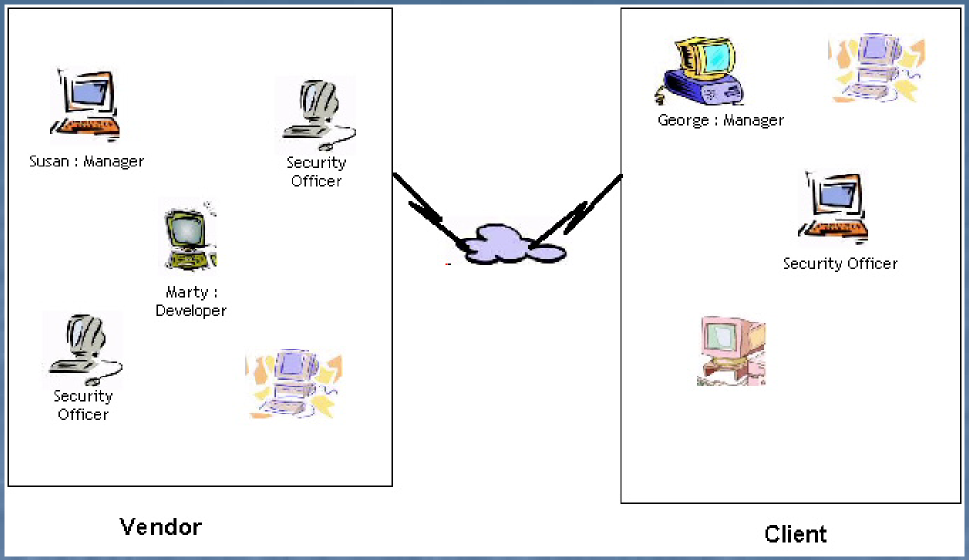
- Supply Chain Management System
  - ✓ Already implemented
- Dynamic Wireless Environment
  - ✓ Already Implemented
- Distributed trust for FIPA platforms
  - Ongoing work
  - Applications
    - Web Services, e.g. ITTALKS (http://www.ittalks.org/)
    - Pervasive Computing

UMBC
an Honors University in Maryland

Scenario 1: Supply Chain Management

- Inter-company information access
- Sharing/accessing information, and performing actions across (or within) organizations
- have to observe organizational policies for security and authorization.
- Implemented for the NIST ATP EECOMS project

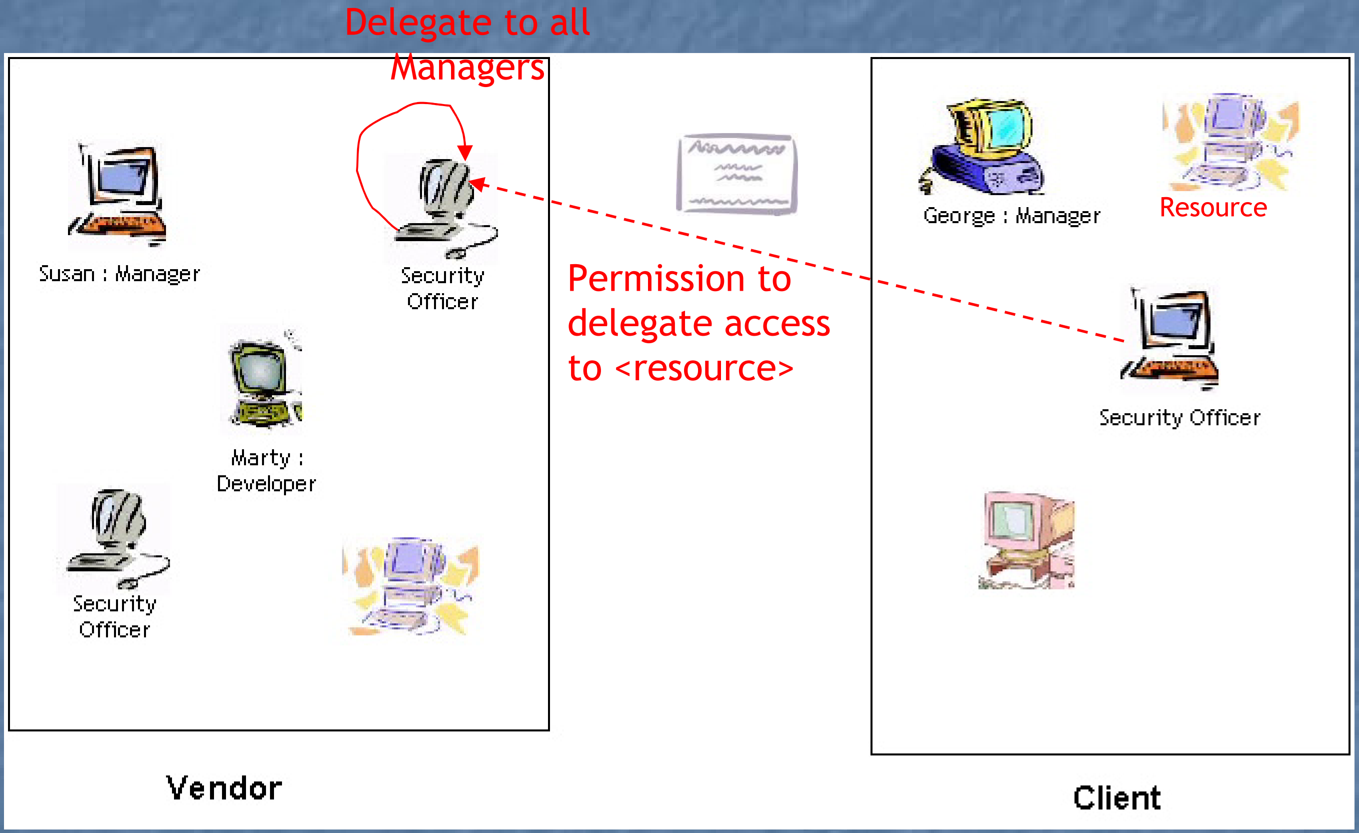
Design for SCM



- Companies have security policies
- Policy enforced by 'security agents'
- Each agent has an ID certificate (X.509)
- All communication via signed messages
- Trust and policy info specified as logic assertions encoded as horn clauses.
- Security agents reason to prove access rights and issue signed statements attesting to them.

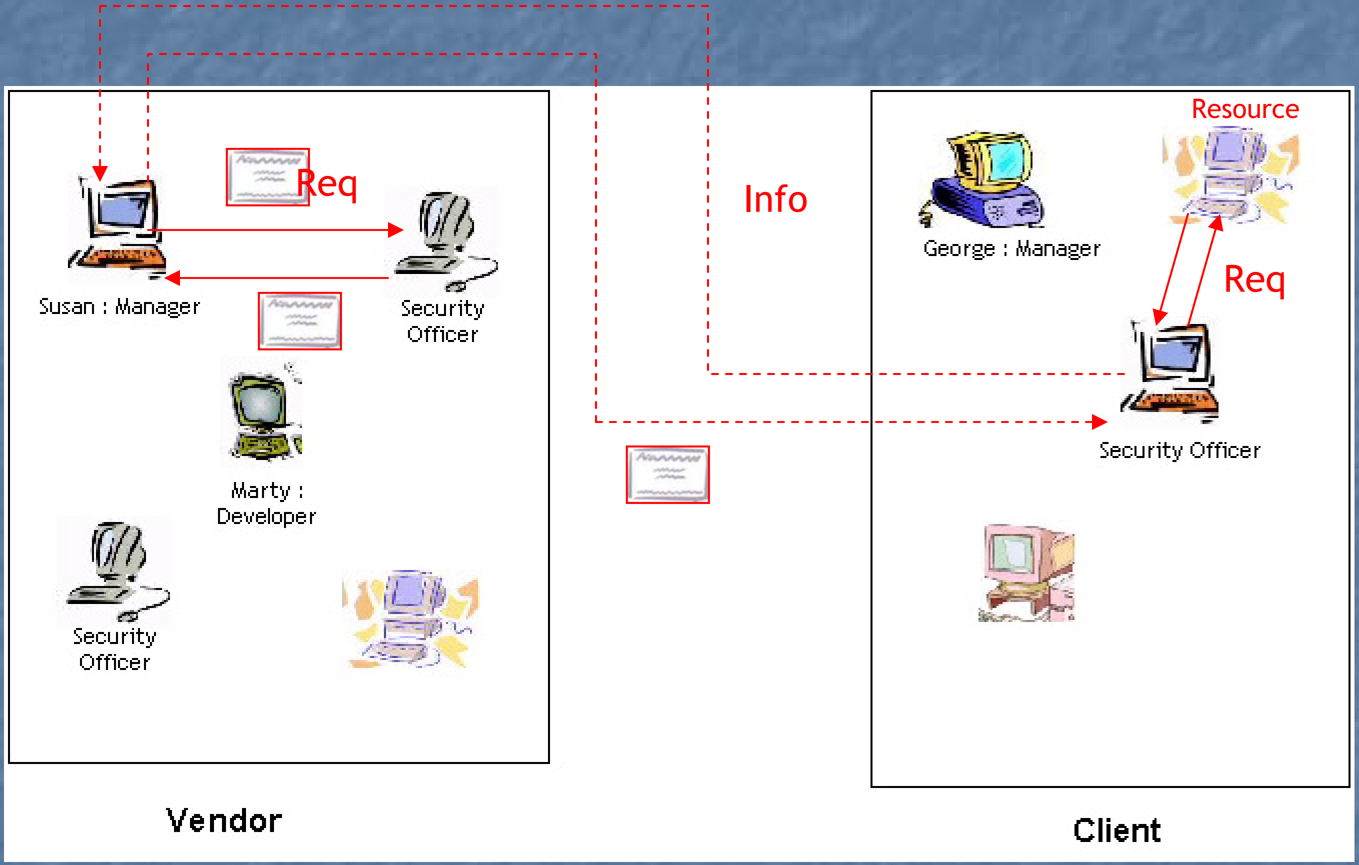
UMBC
an Honors University in Maryland

How it works : Initialization

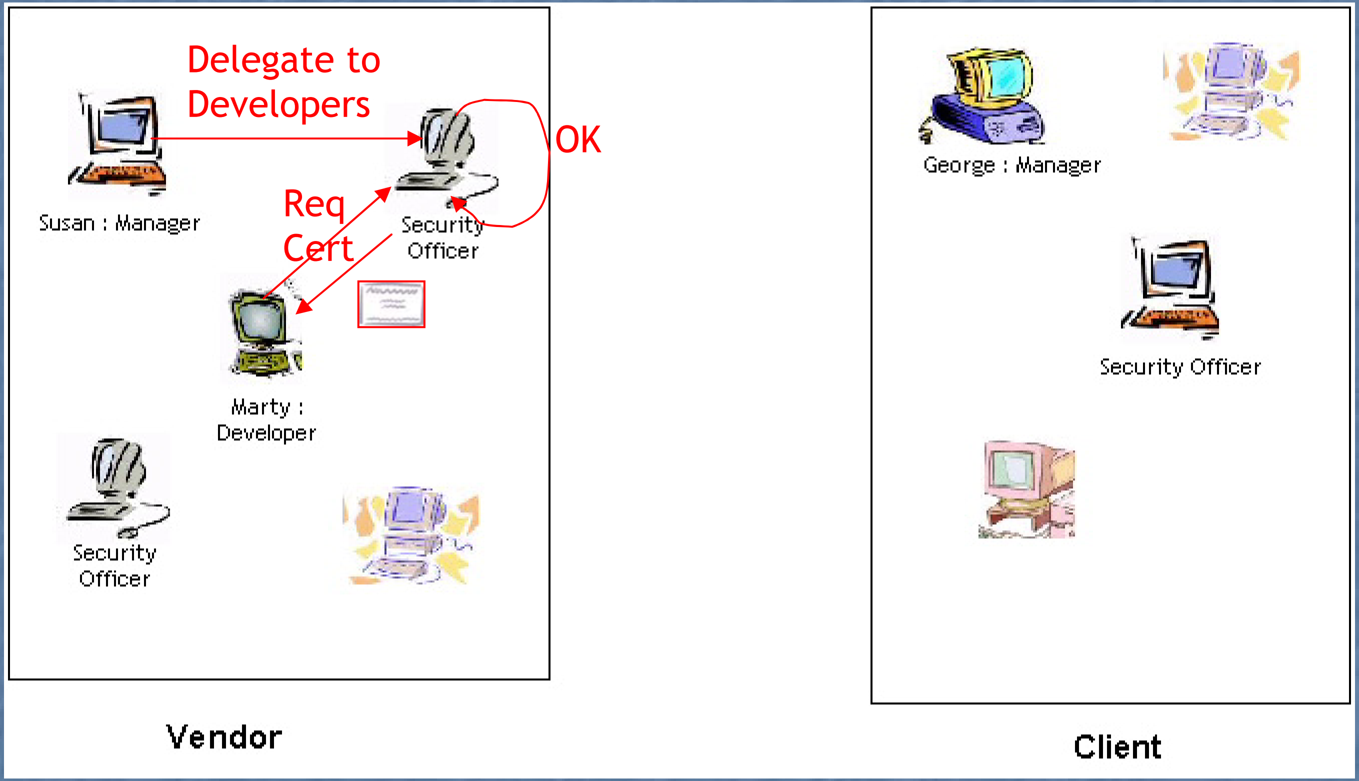


UMBC
an Honors University in Maryland

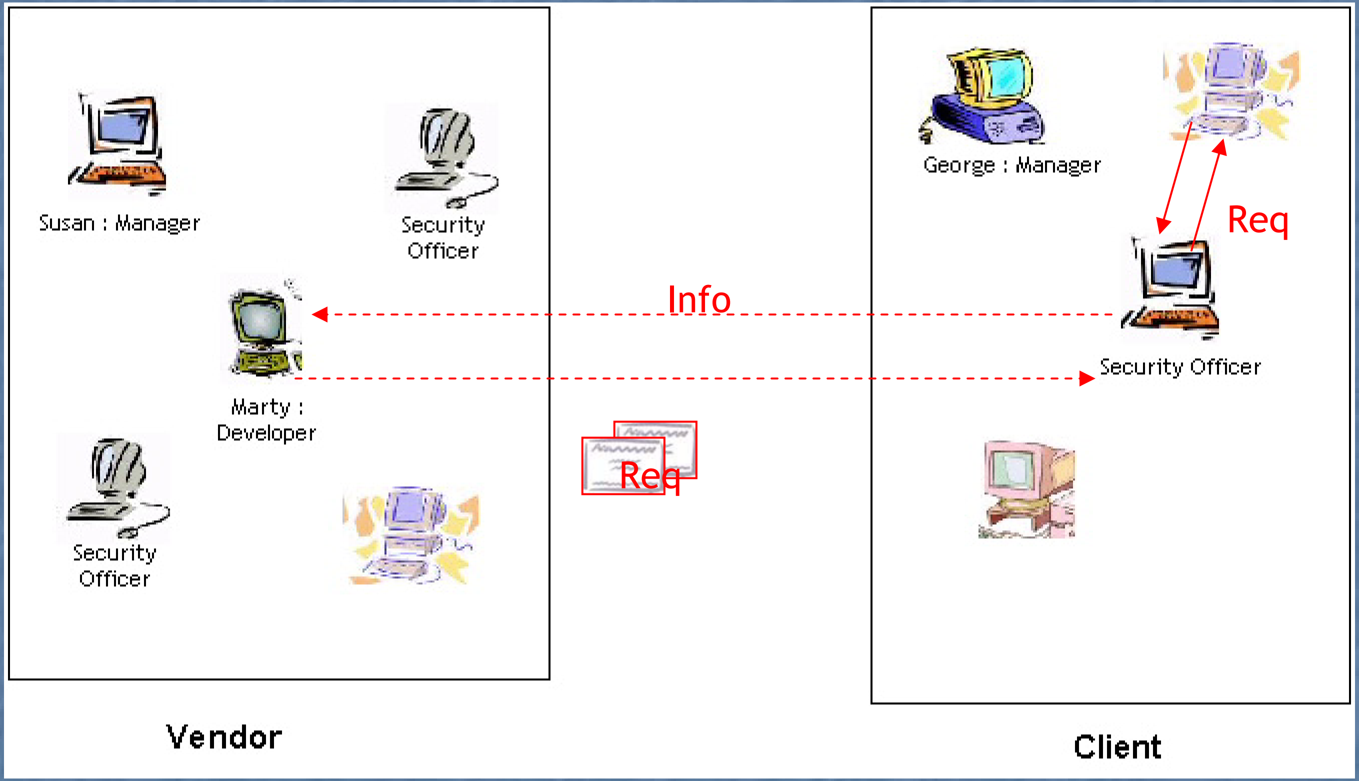
How it works : Request



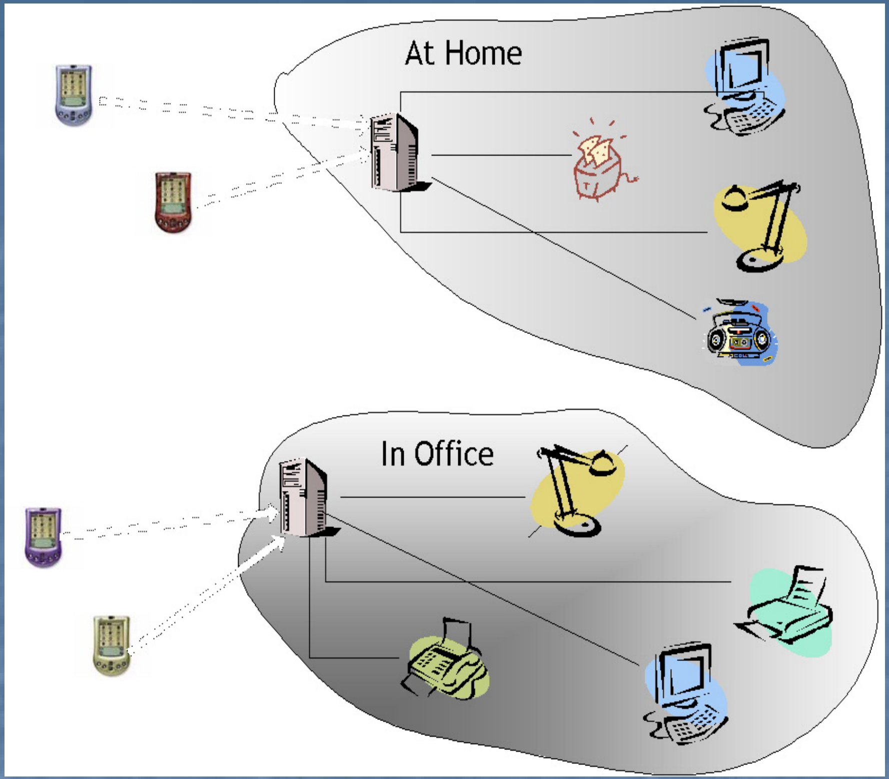
How it works : Delegation



How it works : Request



Scenario 2 : Dynamic Wireless Environment



Working with dynamic, ad hoc wireless environments like Bluetooth

- Unknown entities are involved
- Wireless devices are resource poor
- Authenticate other wireless devices
- Need to communicate and sometimes use other devices

Vigil : Security based on Distributed Trust in Dynamic Wireless Environments

- Extends Centaurus by addition of simplified SPKI
  - Digital certificates are used to establish identity
- Uses ***Dynamic*** *Role Based Access Control*
  - Roles determined by credentials of entity/agent/user
  - Access rights can be changed without affecting the role
- Delegations and revocations are possible by authorized entities/agents/users
- Implemented for Bluetooth enabled laptops

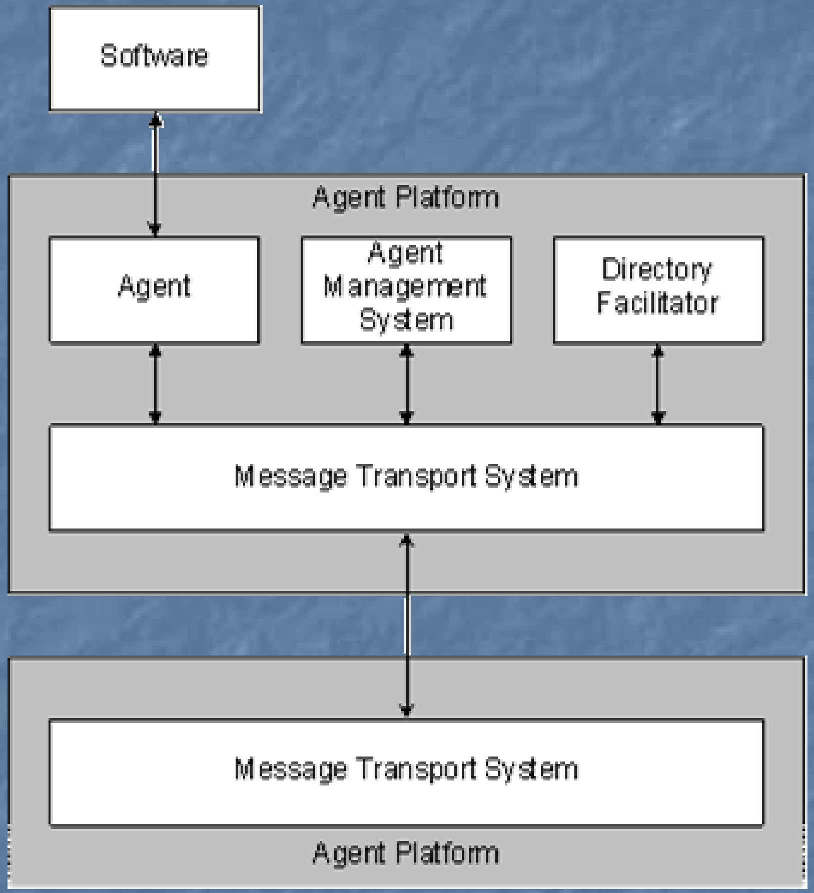
Scenario 3 : Distributed Trust For FIPA Agent Platforms

- Software Agents and Pervasive Computing are a good match
- Biggest problem with security is complicated usage and additional administration
- Adding security to the infrastructure makes it more acceptable
  - We are incorporating the distributed trust model into a widely used agent platform, JADE, and its lightweight counterpart, LEAP
- Part of our ongoing work

Distributed Trust for an Agent Platform

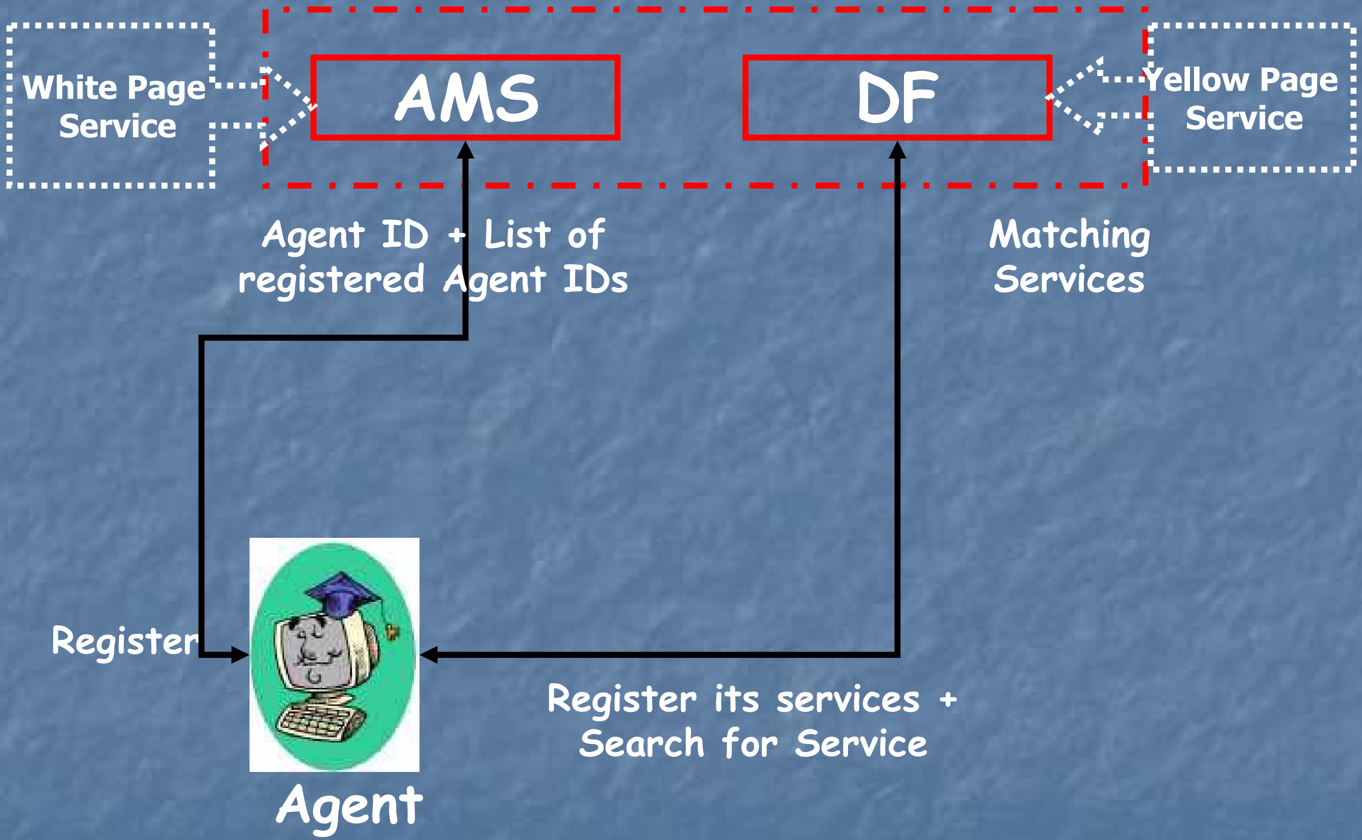
- Is being designed to be *general,* i.e. easily adaptable to several distributed computing scenarios
- This model will be extended to provide security for
  - Web Services
  - Pervasive Computing

FIPA Agent Platform



From FIPA Specifications
http://fipa.org/

FIPA Agent Platform

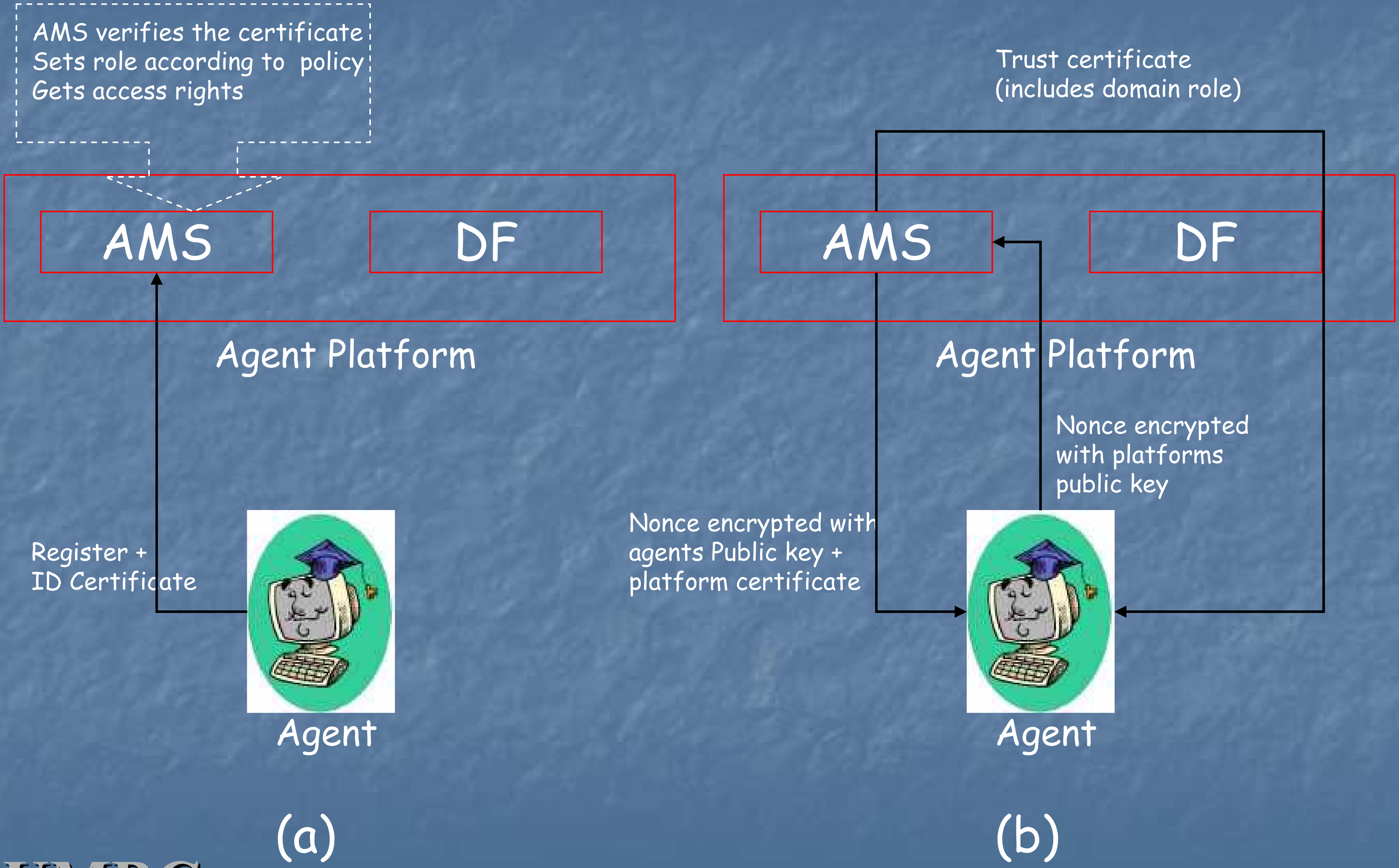

White Page Service
AMS
DF
Yellow Page Service
Agent ID + List of registered Agent IDs
Matching Services
Register
Register its services + Search for Service
Agent

Security for an Agent Platform

Security must be enforced at several levels

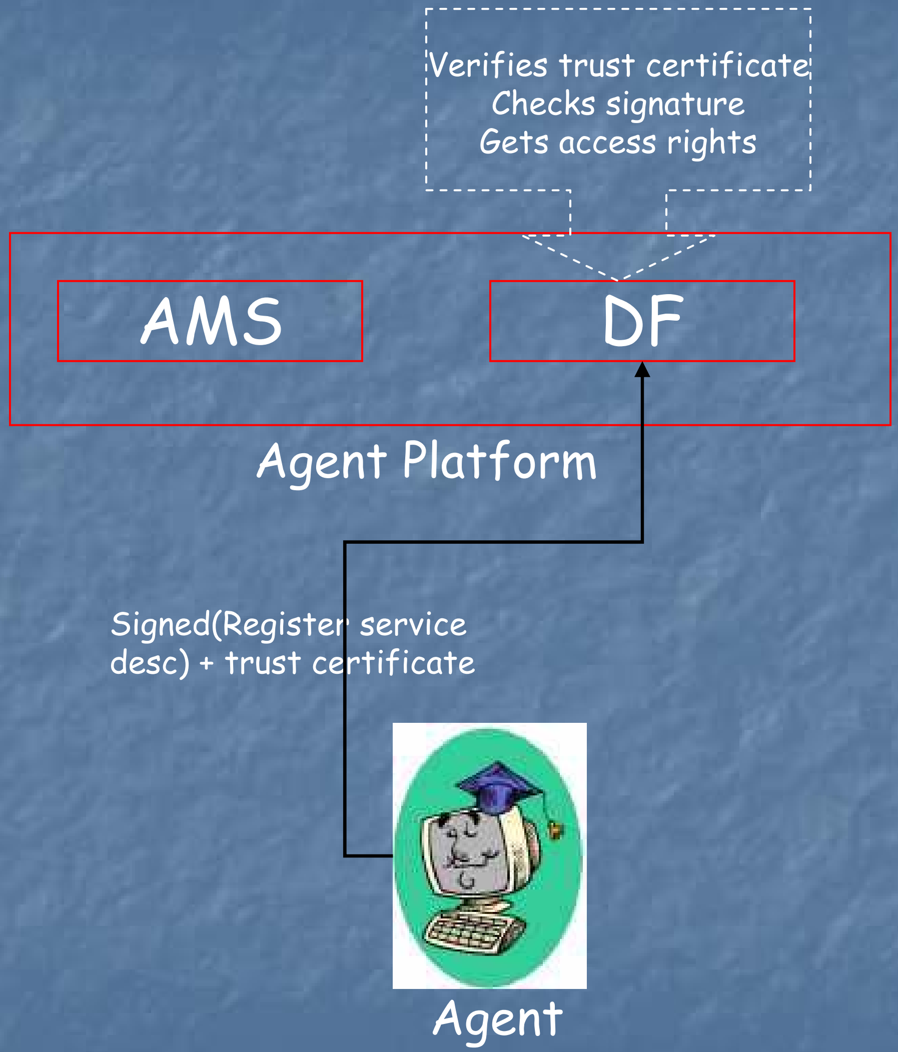
- Communication
  - Implemented in most widely used agent platform, JADE
- AMS (Agent Management Service)
  - Spoofing
  - Denial of service
- DF (Directory Facilitator)
  - Spoofing
  - Denial of service
  - False or misleading "advertising"
- Agent
  - Flexible control over access to its services
  - Accountability
  - Protection from a corrupted AMS and/or DF

Security for AMS

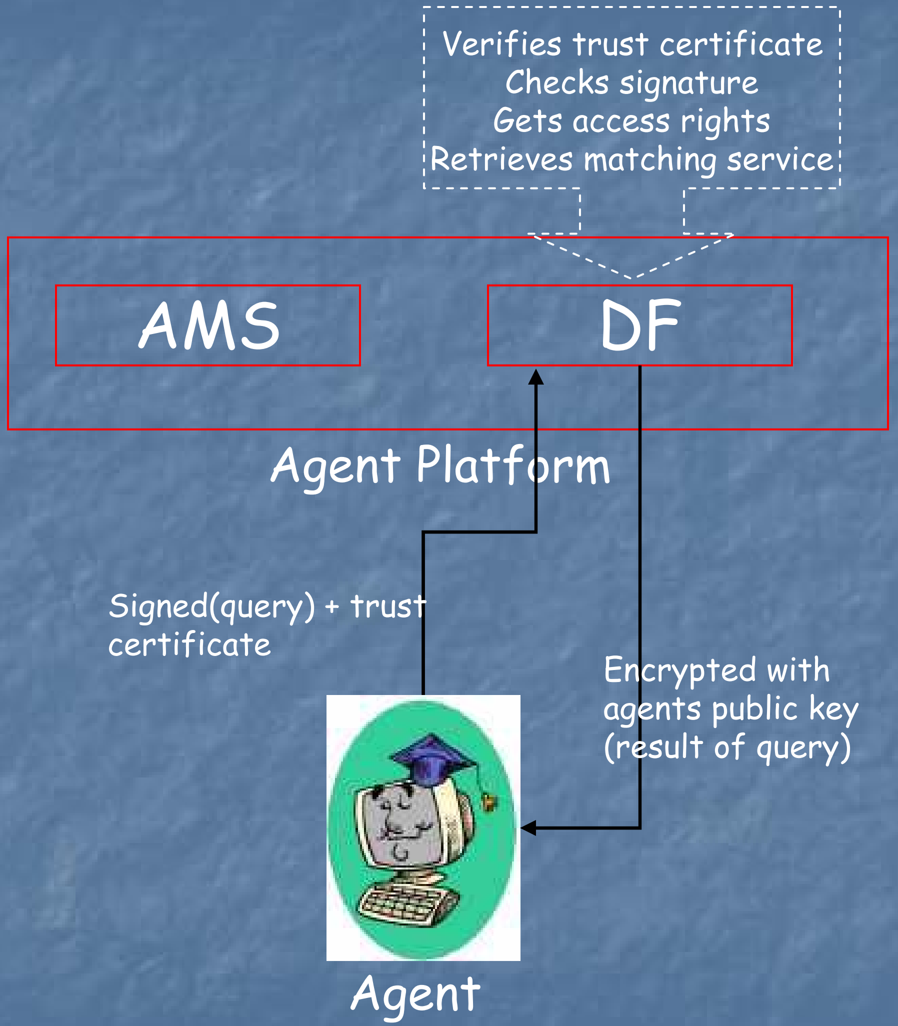


UMBC
an Honors University in Maryland

Security for DF


Verifies trust certificate
Checks signature
Gets access rights
AMS
DF
Agent Platform
Signed(Register service desc) + trust certificate
Agent

(a)


Verifies trust certificate
Checks signature
Gets access rights
Retrieves matching service
AMS
DF
Agent Platform
Signed(query) + trust certificate
Encrypted with agents public key (result of query)
Agent

(b)

Security for an Agent

- An agent can register itself with the AMS as Private, Secure or Open.
  - Private: The AMS will never display the Agent's ID to any other agent
  - Secure: The AMS will *only* display the Agent's ID to an agent that fulfills certain conditions specified by the former
  - Open: The AMS will display the Agent's ID to all agent
- A "service providing agent" can register its services with the DF as Private, Secure or Open.
  - Private: The DF will not allow any agent to find that service
  - Secure: The DF will *only* allow an agent, who satisfies the service providing agent's conditions, to find that service
  - Open: The DF will allow any agent to find that service

Distributed Trust for an Agent Platform

- Preliminary design based on our distributed trust model
- Policy driven system
  - Includes rules for role assignment, and access control
- Includes delegations, prohibitions & revocations by agents
  - E.g. An agent can delegate its ability to access a certain service to another agent by sending a 'delegation certificate' to the agent or sending the information to the DF with which the service is registered
  - Working on the effect of entitlements and obligations

Distributed Trust for an Agent Platform

- Handles the following problems
  - Corrupted AMS and DF (key components of an agent platform)
  - Insecure delegations
  - Lack of accountability
  - Access control for foreign agents/users
  - Lack of central control
  - Security across different agent platforms
- Several of these problems are common to Web Services and pervasive computing

Ongoing Work

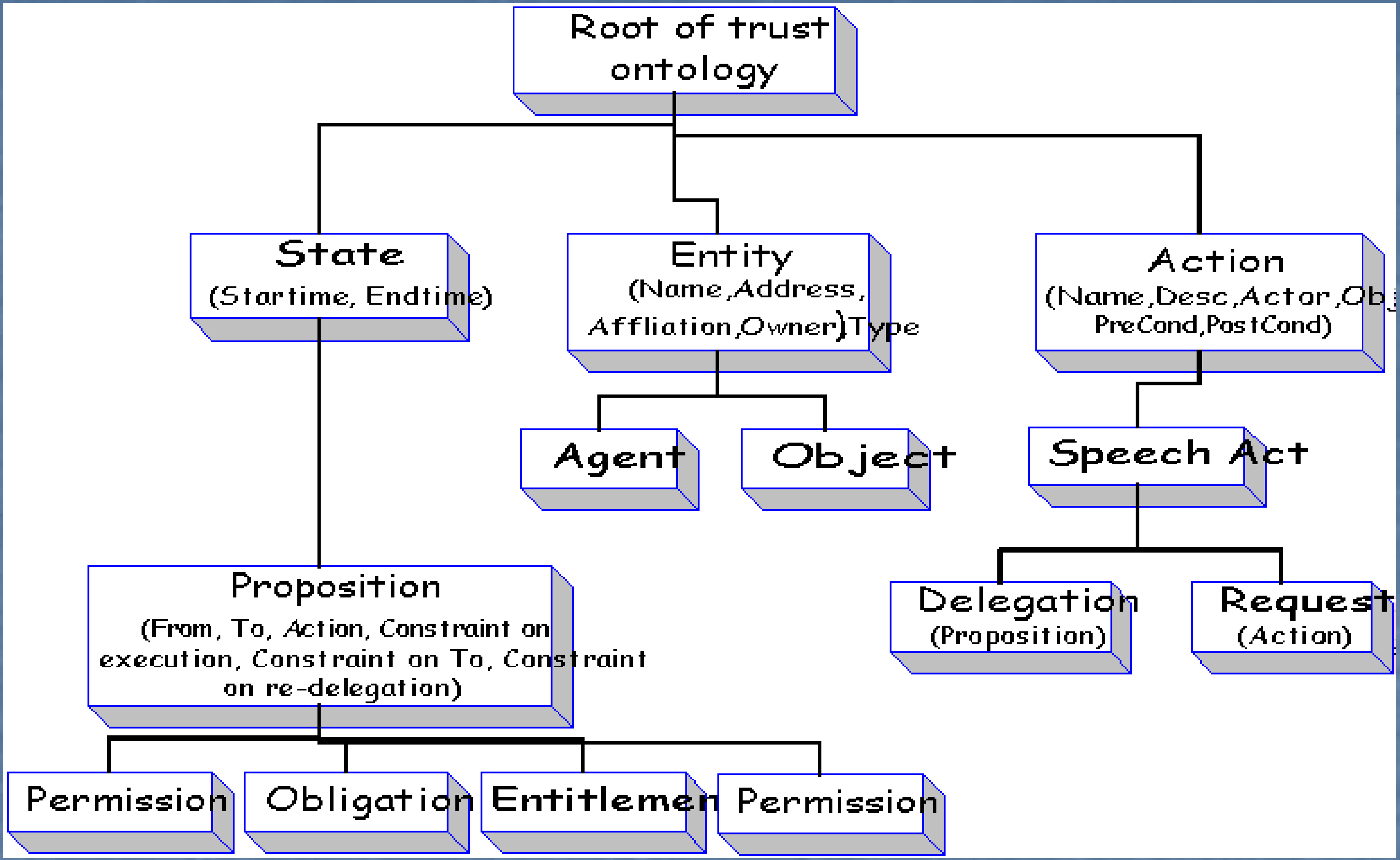
- Specifying ontology for permissions, obligations, entitlements, prohibitions in DAML/RDF
- Also model distributed belief
- Encoded in DAML and/or RDF
- Delegating of permissions, obligations, entitlements, prohibitions and belief
- To avoid the permission revocation problem we use "short lived propositions", e.g.

  "My proof that agent xyz has permission to do action X is good until time t."

Distributed Belief

- A policy specified that "UMBC CSEE faculty are allowed to do X", but who are faculty?
- Our dtrust language allows us to say
  - "*We accept http://www.csee.umbc.edu/faculty.html as a trusted source of information about membership in the class http://umbc.edu/ontologies/people#faculty*"
- *faculty.html* has a human-readable faculty list (in HTML) and (possibly signed) statements (in DAML) asserting who the faculty are.
- Beliefs can be delegated as well
  - "*I delegate belief of phdAdvisee(X,Y) to X if X is a CSEE faculty member*"

Dtrust Ontology



A DAML ontology for describing authorization and trust actions, states and policies.

UMBC
an Honors University in Maryland

DAML in One Slide

DAML is built on top of XML and RDF

It allows the definition, sharing, composition and use of ontologies

DAML is ~= a frame based knowledge representation language

It can be used to add metadata about anything which has a URI.

URIs are a W3C standard generalizing URLs

everything has URI

```
<rdf:RDF xmlns:rdf ="http://w3.org/22-rdf-syntax-ns#"
    xmlns:rdfs="http://w3.org/rdf-schema#"
    xmlns:daml="http://daml.org/daml+oil#“>
<daml:Ontology rdf:about="">
     <daml:imports rdf:resource="http://daml.org/daml+oil"/>
</daml:Ontology>
<rdfs:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hasFather"/>
    </daml:Restriction>  </rdfs:subClassOf> </rdfs:Class>
<Person rdf:about=“http://umbc.edu/~finin/">
<rdfs:comment>Finin is a person.</rdfs:comment>
</Person>
```

UMBC

an Honors University in Maryland

A DAML dtrust Example

- Susan *delegates* to Marty the ability to access all her files
- Between 10.00 am on 8/1/2001 to 12.00 am on 8/5/2001
- He is also allowed to *re-delegate* this ability
- But he can only *re-delegate* to agents affiliated to UMBC and on one of Susan’s file called file123.txt

A DAML dtrust Example

```
<!-- Susan’s agent -->
<agent>
        <name>susan-agent</name>
        <affiliation>UMBC</affiliation>
        <owner>susan</owner>
</agent>

<!-- Marty’s agent -->
<agent>
        <name>marty-agent</name>
        <affiliation>UMBC</affiliation>
        <owner>marty</owner>
</agent>

<!-- all agents affiliated to UMBC -->
<agent rdf:ID=”umbc-agent”>
        <affiliation>UMBC</affiliation>
</agent>

<!-- Susan’s file, file123.txt -->
<object>
        <name>file123.txt</name>
        <owner>susan-agent</owner>
        <type>FILE</type>
</object>
`
```

```
<!-- all files belonging to Susan -->
<object rdf:ID=”susan-files”>
     <owner>susan-agent</owner>
     <type>FILE</type>
</object>

%% informing the system the meaning of
     readfileaccesss
%% add more properties
<!-- ReadFileAction -->
<rdfs:Class rdf:ID="ReadFileAccess">
     <rdfs:subClassOf
     rdf:resource="#Action"/>
     <rdfs:label>ReadFileAccess</rdfs:label>
</rdfs:Class>
```

A DAML dtrust Example (cont)

```
<delegation rdf:ID="Delegation1">
	<from>susan-agent</from>
	<to>marty-agent</to>
	<permission>
		<from>susan-agent</from>
		<to>marty-agent</to>
		<starttime>2001:8:1:10:00</starttime>
		<endtime>2001:8:5:24:00</endtime>
		<readfileaccess>
			<name>ReadFileAccess</name>
			<actor>umbc-agent</actor>
			<targets>susan-files</targets>
			<redelegatable>
				<permission>
					<readfileaccess>
						<actor>umbc-agent</actor>
						<targets>file123.txt</targets>
					</readfileaccess>
				</permission>
			</redelegatable>
			<precondition>
				<request>
					<readfileaccess>
						<targets>susan-files</targets>
					</readfileaccess>
				</request>
			</precondition>
		</readfileaccess>
	</permission>
</delegation>
```

Future Work

- Explore the use of XML Signatures to sign DAML statements
- Incorporate a *reputation* mechanism to provide sanctions for failing to follow obligations
- Develop specifications for security policies
- Detect conflicting policies
- Develop a *dtrust* language for web services