Distributed Trust

Lalana Kagal
Tim Finin
Scott Cost
Yun Peng

University of Maryland Baltimore County

UMBC

1. Overview and context
2. Scenarios
3. What is Distributed Trust
4. Design
5. How it works : An Example
6. Ongoing Work
7. Future Research Direction
8. Summary

http://umbc.edu/~finin/papers/ijcai01/

Context

- ✓ 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.

Three Scenarios

1. Supply Chain Management System
   - ✓Already implemented
2. Dynamic Wireless Environment
   - ✓Ongoing work
3. Distributed Trust for Web Services
   - ✓Future work
   - ✓To be applied to ITTALKS (http://www.ittalks.org/)

Scenario 1: Supply Chain Mgmt

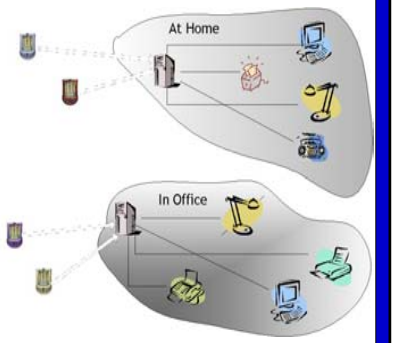
- ✓ Inter company information access
- ✓ Sharing/accessing information, and performing actions across (or within) organizations
- ✓ have to observe organizational policies for security and authorization.

Implemented for the NIST ATP EECOMS project

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



Scenario 3: ITTALKS

- **ITTALKS** is a database driven web site of IT related talks at UMBC and other institutions. The database contains information on
  - Seminar events
  - People (speakers, hosts, users,…)
  - Places (rooms, institutions,…)

http://ittalks.org/

- This database is used to dynamically generate web pages and DAML descriptions for the talks and related information.
- Notifications are sent to registered users and/or their agents via email, SMS, WAP, and/or KQML for talks matching their interests, location and schedule.

What is Distributed Trust

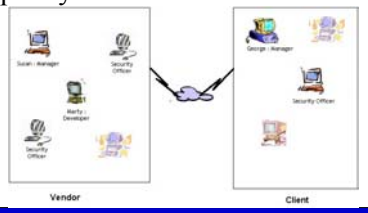
✓ Issues
- No central authority
- logging in is not possible
- Access control for entities never encountered before

We use *Distributed Trust* to solve these issues

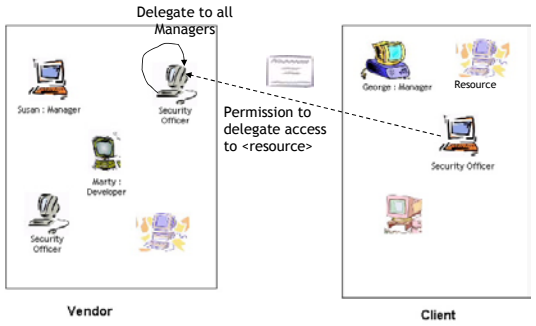
✓ trust = policies + credentials + delegation actions + proofs of deontic properties

Design for SCM

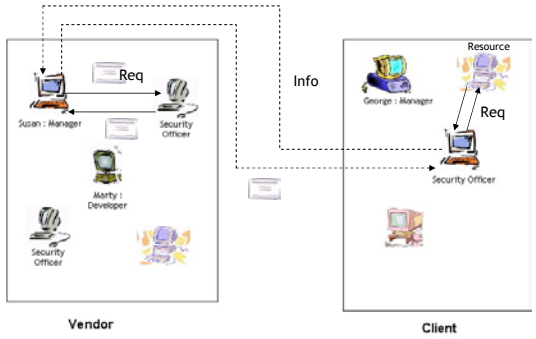
✓ Companies have security policies
✓ Policy enforced by a number of 'security officers'
✓ Each agent in the system has an ID certificate, X.509
✓ All communication via signed messages
✓ Trust and policy info encoded as horn clauses



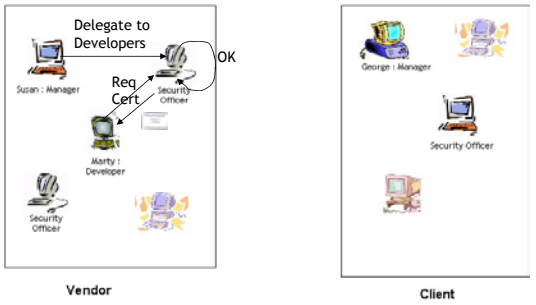
How it works : Initialization



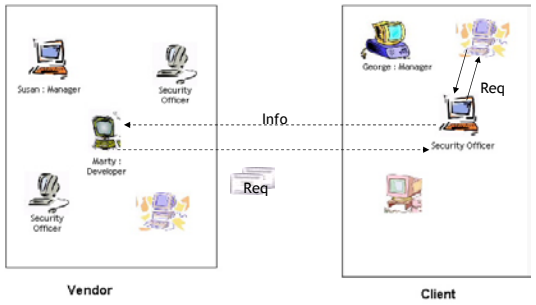
How it works : Request



How it works : Delegation



How it works : Request



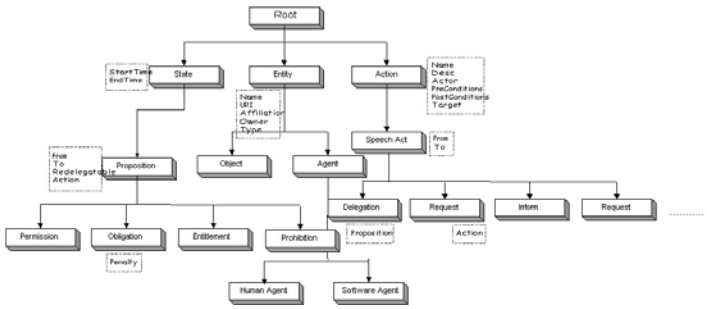
Ongoing Work

- ✓ 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 xyzzy 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 how do we determine who they are?
- ✓ 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 my belief about *phdAdvisee* property to all CSEE faculty"

Dtrust Ontology



What is DAML ?

DAML = **D**arpa **A**gent **M**arkup **L**anguage *

- ✓Goal is to define a language for the semantic web
- ✓Developing language spec, tools, applications
- ✓DAML is a language for the Semantic Web

* for more information see http://www.daml.org/

How does DAML help ?

DAML will enable the next major generation of Web/Internet technology...

- The 1st generation, the Internet, enabled disparate machines to exchange data.
- The 2nd generation, the World Wide Web, enabled new applications on top of the growing Internet, making enormous amounts of information available
- The next generation of the net is an "agent-enabled" resource (the "Semantic Web") which makes a huge amount of information available in machine-readable form creating a revolution in new applications, environments, and b2b e-commerce.

…by enabling "agent" communication at a Web-wide scale.


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

- ✓ Use XML Signatures to sign DAML statements
- ✓ Incorporate a reputation mechanism
- ✓ Handle conflicting policies
- ✓ Develop a *dtrust* language for web services

UMBC ebiquity

Summary

- ✓ We have developed an infrastructure for *distributed trust*
- ✓ Designed a representation for trust info, credentials and policies
- ✓ Shown its feasibility through implementation
- ✓ Discussed some of our current work with the representation of security and trust info in a semantic language like DAML
- ✓ Future research directions

UMBC ebiquity