A Semantic e-Wallet to Reconcile Privacy and Context Awareness

Fabien L. Gandon & Norman M. Sadeh

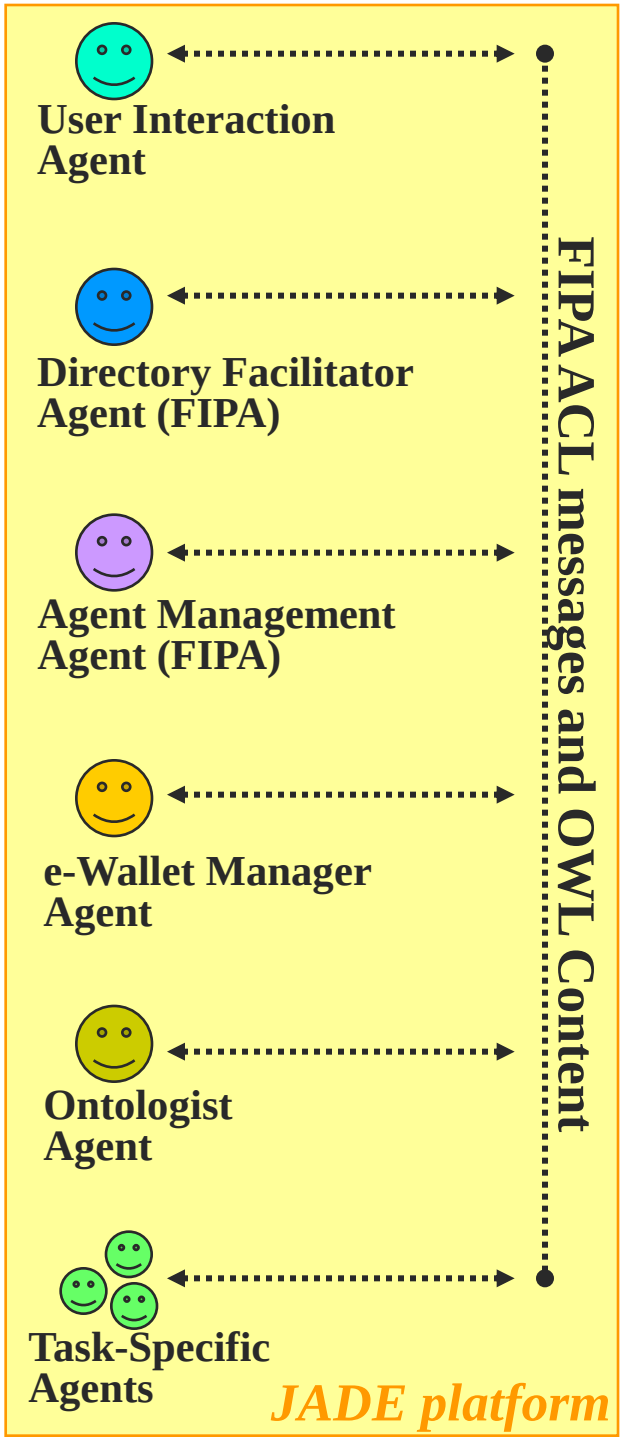
Mobile Commerce Lab. – Carnegie Mellon University

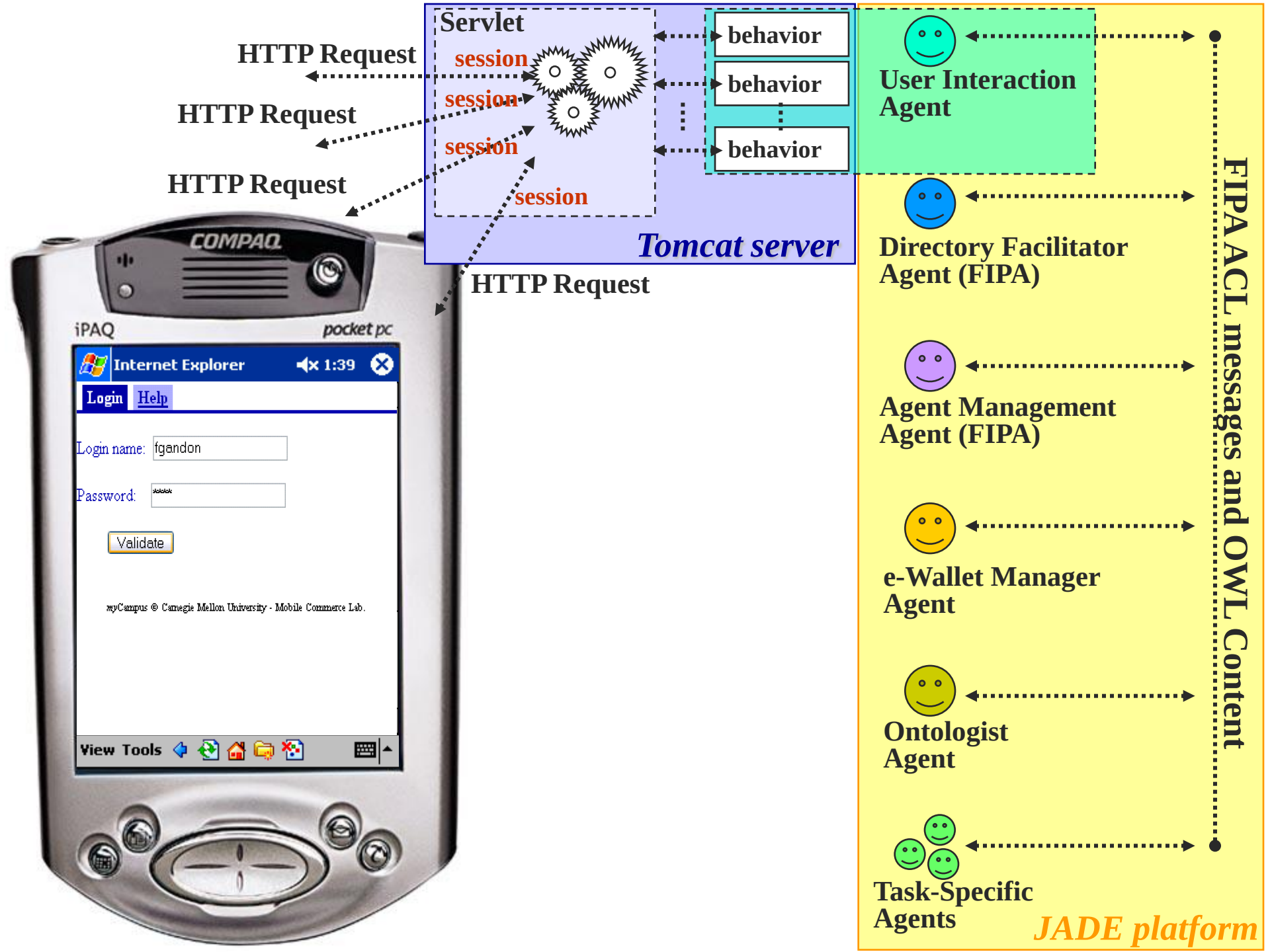
Motivations

- *Pervasive Computing*
  - Multiple sources of contextual information
    e.g. calendar, location tracking, organizational info, pref.
- *Virtual Enterprise Collaboration*
  - Selectively sharing of information with prospective and actual customers and suppliers
    e.g. collaborative design, supplier evaluation, available-to-promise/capable-to-promise information, order tracking
- Rather than exposing all these resources as individual semantic web services, organizations and individuals will want to have *unified gateways to their information*… "e-Wallets" to *allow resource identification & enforce confidentiality logic*

Vision - *my*Campus

- Campus as “everyday life microcosm”
- Enhance campus life through context-aware services accessible over a WLAN
- Approach:
  - Involve stakeholders in the design
  - Semantic Web and agent technologies
- A growing collection of context-aware agents that:
  - Users can pull into their own personal environment
  - Customize themselves through automated identification and access of relevant contextual resources


User Interaction Agent
Directory Facilitator Agent (FIPA)
Agent Management Agent (FIPA)
e-Wallet Manager Agent
Ontologist Agent
Task-Specific Agents
FIPA ACL messages and OWL Content
JADE platform


HTTP Request
HTTP Request
HTTP Request
HTTP Request
Servlet
session
session
session
session
behavior
behavior
behavior
Tomcat server
User Interaction Agent
Directory Facilitator Agent (FIPA)
Agent Management Agent (FIPA)
e-Wallet Manager Agent
Ontologist Agent
Task-Specific Agents
FIPA ACL messages and OWL Content
JADE platform
COMPAQ
iPAQ
pocket pc
Internet Explorer
1:39
Login
Help
Login name:
fgandon
Password:
Validate
myCampus © Carnegie Mellon University - Mobile Commerce Lab.
View
Tools

HTTP Request

User Interaction Agent

Yellow Pages

Type
Service
Owner
…

Directory Facilitator Agent (FIPA)

White Pages

Name
Address
…

Agent Management Agent (FIPA)

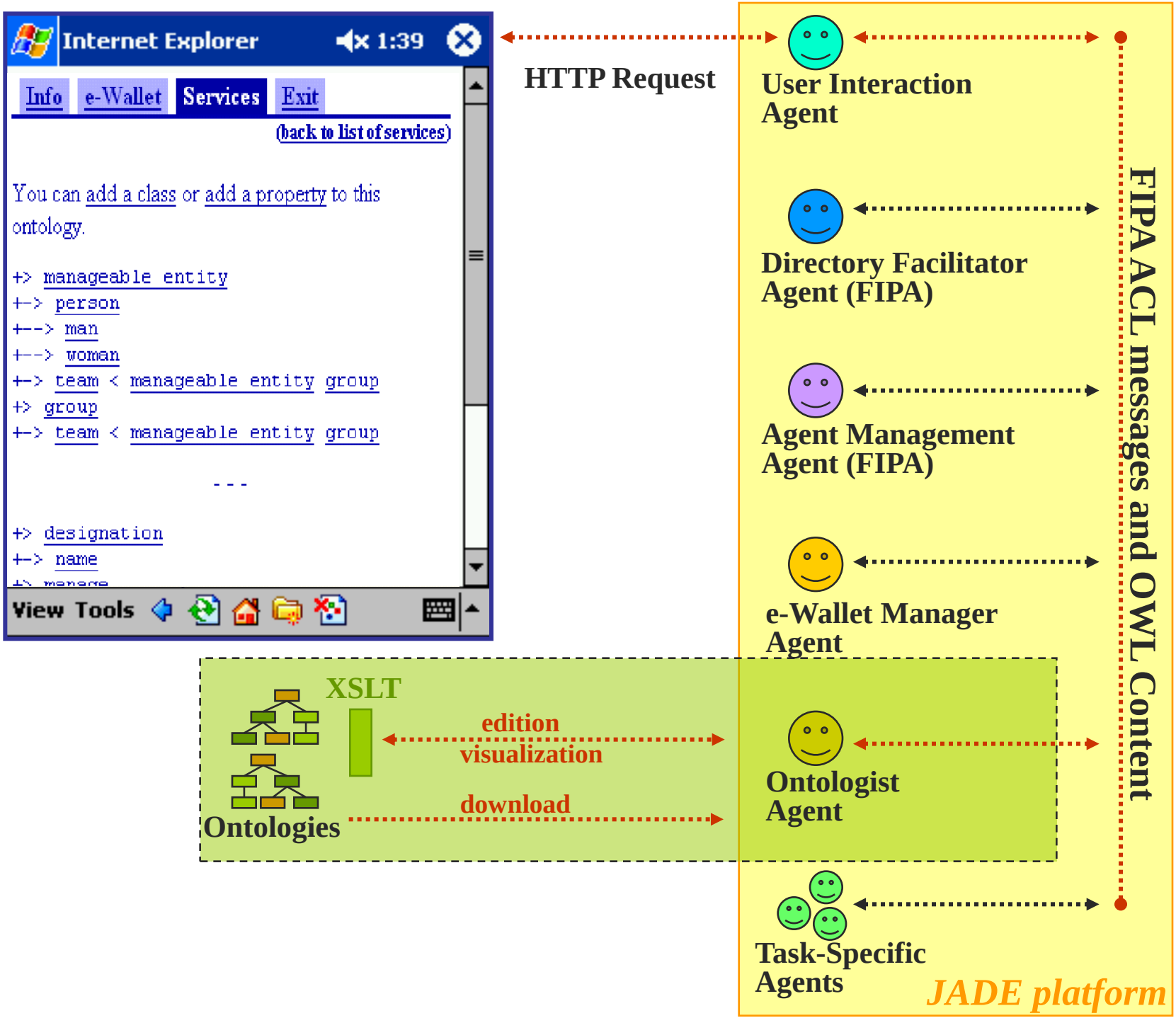
e-Wallet Manager Agent

Ontologist Agent

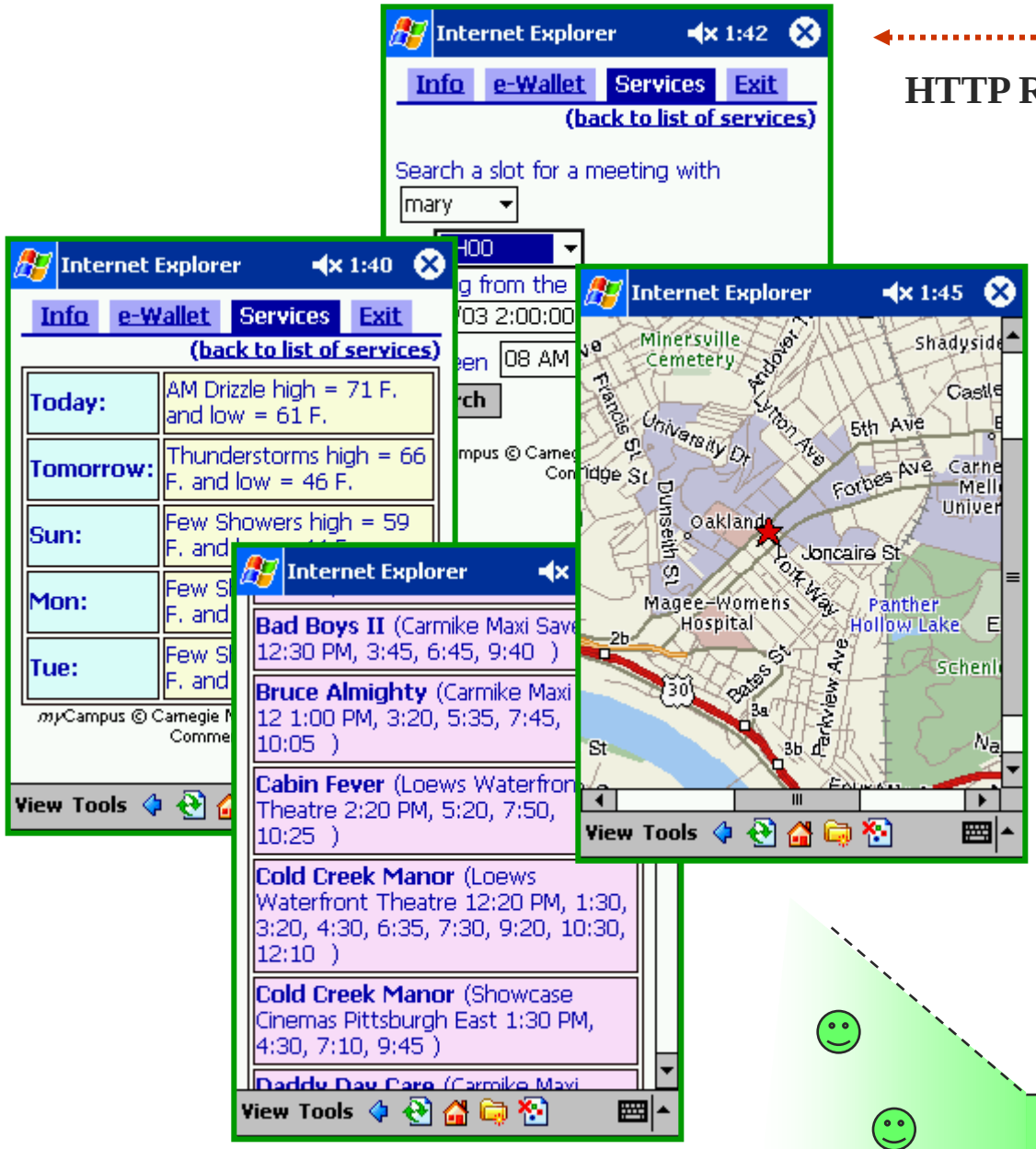
Task-Specific Agents

FIPA ACL messages and OWL Content

JADE platform


Internet Explorer
1:39
Info
e-Wallet
Services
Exit
(back to list of services)
You can add a class or add a property to this ontology.
+> manageable entity
+-> person
+--> man
+--> woman
+-> team < manageable entity group
+> group
+-> team < manageable entity group

+> designation
+-> name
View Tools
HTTP Request
User Interaction Agent
Directory Facilitator Agent (FIPA)
Agent Management Agent (FIPA)
e-Wallet Manager Agent
FIPA ACL messages and OWL Content
XSLT
edition
visualization
download
Ontologies
Ontologist Agent
Task-Specific Agents
JADE platform



HTTP Request

User Interaction Agent

Directory Facilitator Agent (FIPA)

Agent Management Agent (FIPA)

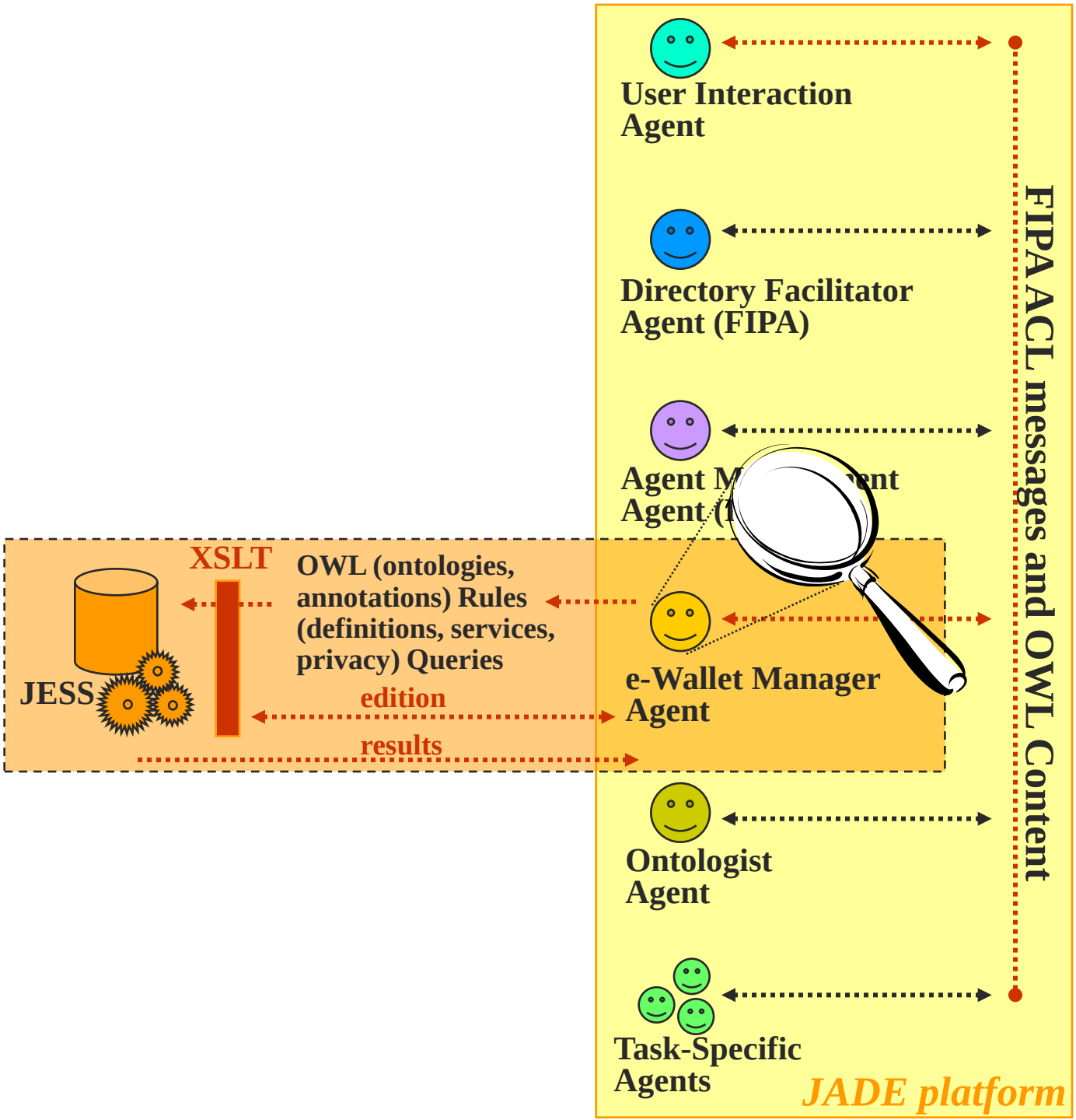
e-Wallet Manager Agent

Ontologist Agent

Task-Specific Agents

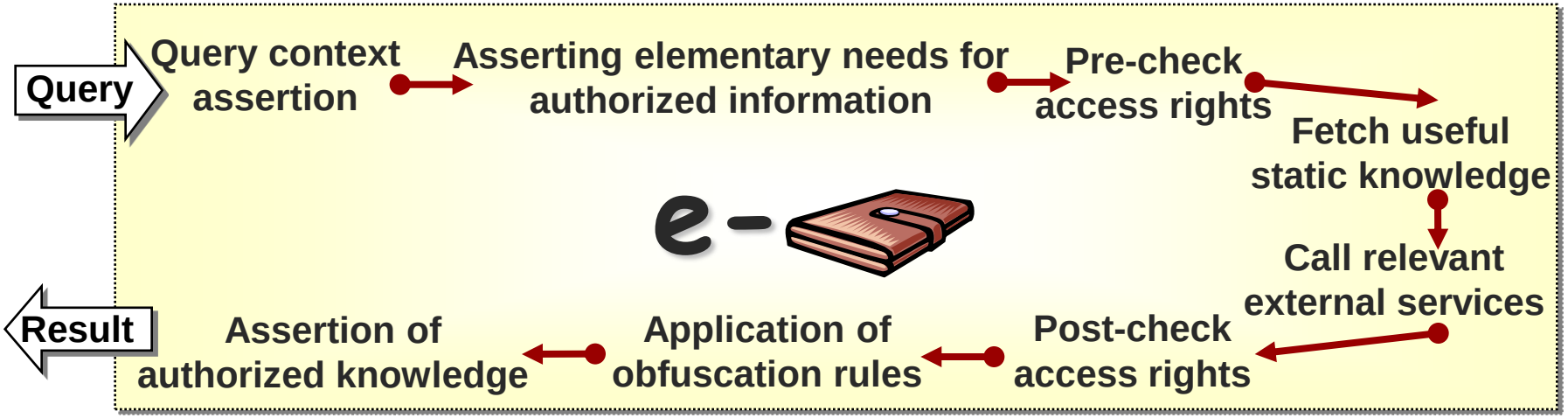
FIPA ACL messages and OWL Content

JADE platform


User Interaction Agent
Directory Facilitator Agent (FIPA)
Agent M ent Agent (
e-Wallet Manager Agent
Ontologist Agent
Task-Specific Agents
FIPA ACL messages and OWL Content
JADE platform
XSLT
OWL (ontologies, annotations) Rules (definitions, services, privacy) Queries
JESS
edition
results

e-Wallet

- Each user has a *semantic e-Wallet*
  - Automated identification and access of a user's personal resources subject to privacy preferences
  - Personal resources implemented as semantic Web services
- Needs for procedural knowledge *i.e.* rules
  - Resource identification rules
  - Privacy / confidentiality rules
    - Access Control rules
      e.g. "Only my colleagues can see my location"
    - Obfuscation rules
      e.g. "My colleagues can only see the building I am in but not the actual room"
- Note: The same concept applies to virtual organizations and B2B scenarios



Example: Query from John inquiring about Mary's location

Step-1 The sender of the query is John.

Step-2 The query requires finding Mary's location.

Step-3

(a) Is John allowed to see Mary's location given what we currently know?

(b) Checking Mary's privacy/confidentiality preferences, e.g.: Only her colleagues can see her location – and only when she is on campus.

(c) Is John a colleague of Mary? Yes.

Step-4 No action in this example.

Step-5 Finding Mary's location by accessing some location tracking functionality or looking in her calendar.

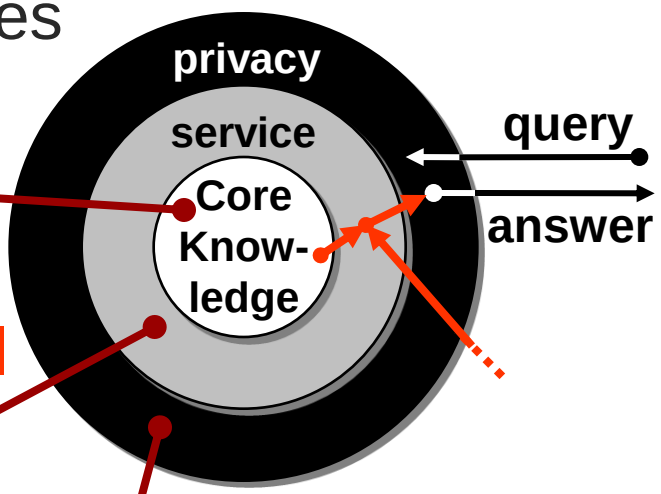
Step-6 Is Mary on campus? Yes.

Step-7 Checking Mary's obfuscation rules e.g. Mary is only willing to disclose the building she is in but not the specific room.

Step-8 "Mary is in Smith Hall".

Design of an e-Wallet

- Three-layer architecture: *security through typing*
  - Core knowledge: static & dynamic knowledge of user
  - Service Layer: invoke external sources of knowledge - web services and
  - y rules
    s control
    & obf
  - Back
    y rules, service rules,
    static



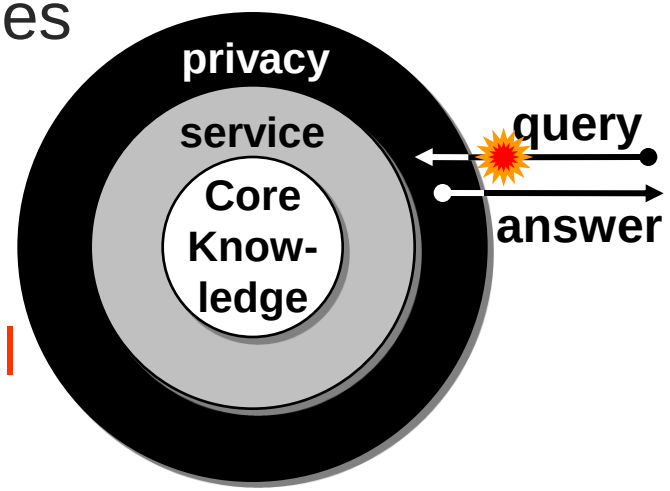

```
(deftemplate triple
   (slot predicate (default ""))
   (slot subject (default ""))
   (slot object (default ""))
)
```

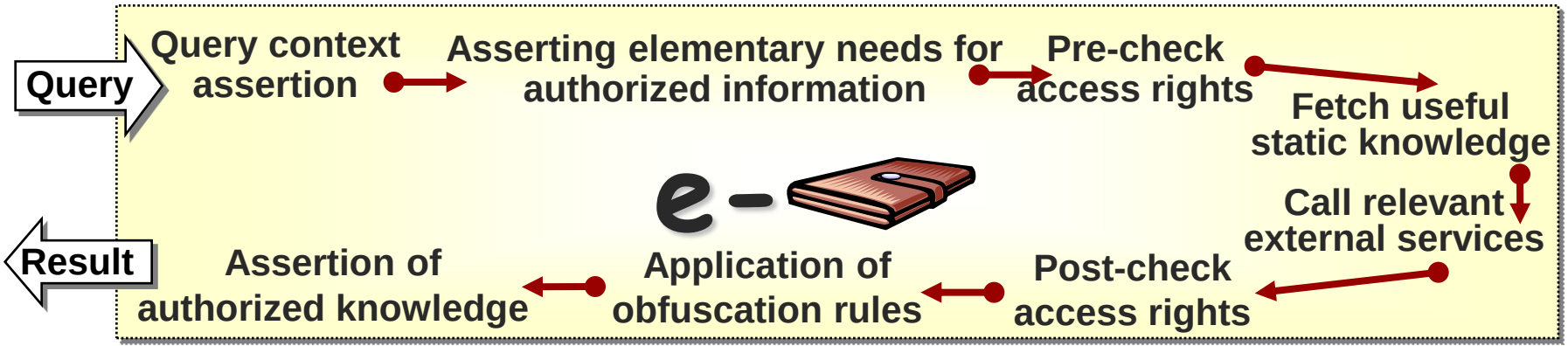
```
(deftemplate service_triple
   (slot predicate (default ""))
   (slot subject (default ""))
   (slot object (default ""))
)
```

```
(deftemplate authorized_triple
   (slot predicate (default ""))
   (slot subject (default ""))
   (slot object (default ""))
)
```

Design of an e-Wallet

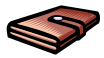
- Three-layer architecture: *security through typing*
  - Core knowledge: static & dynamic knowledge of user
  - Service Layer: invoke external sources of knowledge - web services and personal resources
  - Privacy layer: enforce privacy rules on external requests - access control & obfuscation
  - Backward chaining migration: privacy rules, service rules, static migration rules

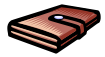
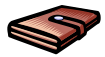
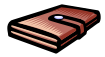


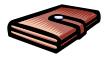


e-Wallet and representation needs

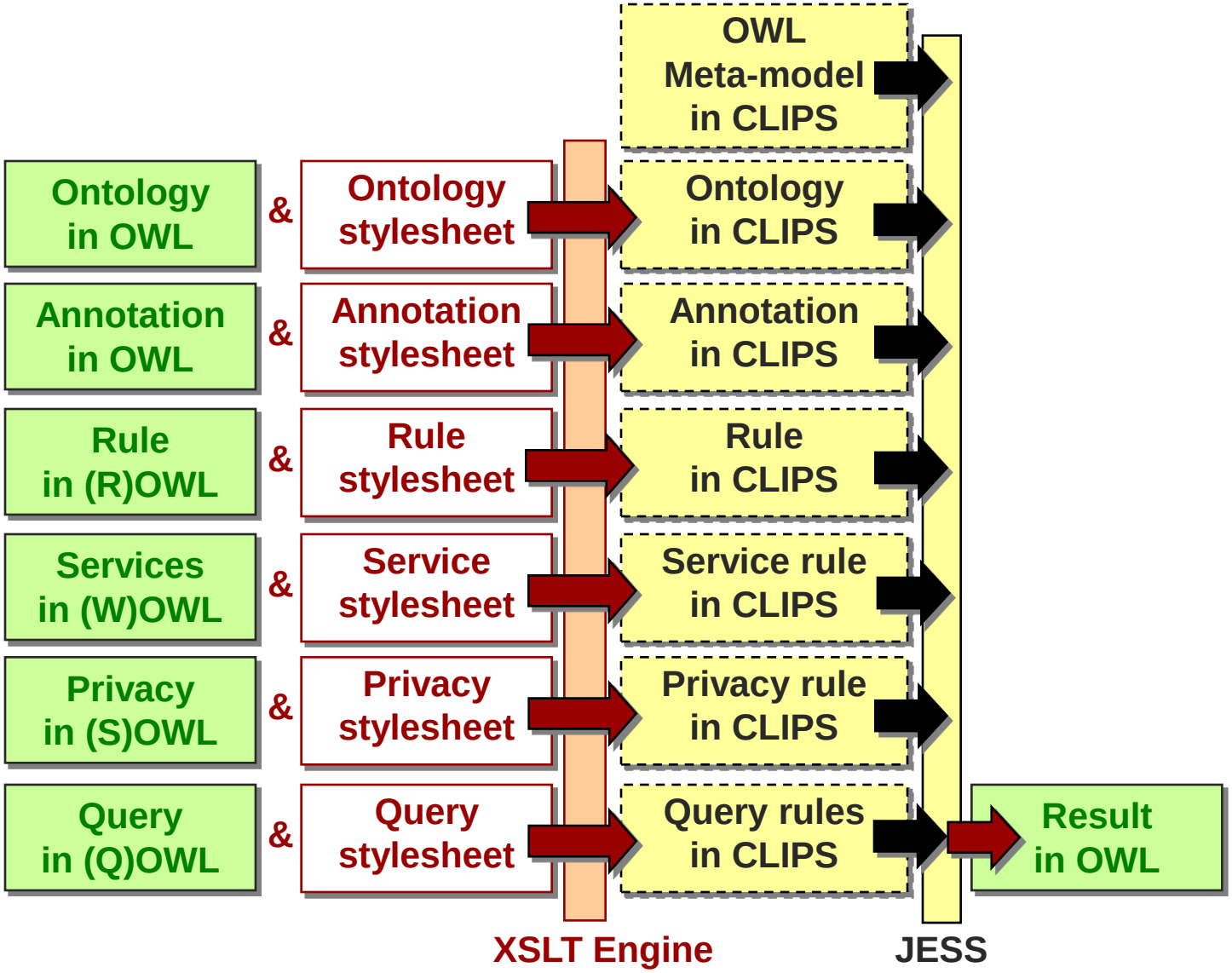
- Static assertional knowledge:
  - User's static profile: OWL annotations.
  - Static contextual knowledge: OWL annotations.
- Dynamic assertional knowledge:
  - User's dynamic profile:
    - Rules in (R)OWL to update profile
    - Result: entailed facts
  - Dynamic contextual knowledge:
    - Rules in (W)OWL to identify and invoke web services
    - Result: facts returned by the web service
  - Security and privacy
    - Rules in (S)OWL to grant access and obfuscate
    - Result: authorized facts for query solving
- Ontologies in OWL.





Summary on the e-Wallet implementation



e-Wallet semantic engine

- RDF Triple model

```
Triple: (predicate, subject, object)
```

```
(deftemplate triple "Template representing a RDF triple"
   (slot predicate (default ""))
   (slot subject (default ""))
   (slot object (default ""))
)
```

- RDFS & OWL meta-model (e.g., symmetry of properties)

```
<rdfs:Class rdf:ID="SymmetricProperty">
  <rdfs:label>SymmetricProperty</rdfs:label>
  <rdfs:subClassOf rdf:resource="#ObjectProperty"/>
</rdfs:Class>
```

```
(triple
 (predicate "http://www.w3.org/2000/01/rdf-schema#subClassOf")
 (subject   "http://www.w3.org/2002/07/owl#SymmetricProperty")
 (object    "http://www.w3.org/2002/07/owl#ObjectProperty")
)
(defrule symmetry (declare (salience 100))
 (triple
  (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
  (subject   ?p)
  (object    "http://www.w3.org/2002/07/owl#SymmetricProperty"))
 (triple (predicate ?p) (subject ?x) (object ?y))
=> (assert (triple (predicate ?p) (subject ?y) (object ?x))) )
```

e-Wallet semantic engine

- Ontologies: (e.g., declare person, location, *etc.*)

```
<owl:Class rdf:ID="Person">
 <rdfs:subClassOf rdf:resource="#Entity" />
</owl:Class>
<owl:ObjectProperty rdf:ID="location">
 <rdfs:domain rdf:resource="#Entity" />
 <rdfs:range  rdf:resource="#Place"/>
</owl:ObjectProperty>
```

```
(triple
 (predicate "http://www.w3.org/2000/01/rdf-schema#subClassOf")
 (subject   "http://sadehlab.cs.cmu.edu/mycampus#Person")
 (object    "http://sadehlab.cs.cmu.edu/mycampus#Entity")
) ...
```

- Annotations: (e.g., Mary is in Smith Hall, *etc.*)

```
<mc:Woman rdf:ID="http://cs.cmu.edu/People/~mary">
  <mc:location rdf:resource="http://cmu.edu/SmithHall"/>
</mc:Woman>
```

```
(triple
 (predicate "http://sadehlab.cs.cmu.edu/mycampus#location")
 (subject   "http://cs.cmu.edu/People/~mary")
 (object    "http://cmu.edu/SmithHall")
) ...
```

- Available online with XSLT translation stylesheets

e-Wallet semantic engine

- Rules: (e.g., when in I am in a meeting I am busy)

```
<rowl:Rule direction="forward">
 <rdfs:label>Meeting means busy</rdfs:label>
 <rowl:head>
  <mc:Person rdf:ID="&variable;#person">
   <mc:availability><mc:Busy rdf:ID="&mc;#Busy"/></mc:availability>
  </mc:Person>
 </rowl:head>
 <rowl:body>
  <mc:Person rdf:ID="&variable;#person">
   <mc:activity>
    <mc:Meeting rdf:ID="&variable;#activity"/>
   </mc:activity>
  </mc:Person>
 </rowl:body>
</rowl:Rule>
```

```
(defrule Meeting-means-busy
  ...
  (triple (predicate "http://sadehlab.cs.cmu.edu/mycampus#activity")
   (subject ?person) (object ?activity))
  (triple (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
   (subject ?activity) (object "http://sadehlab.cs.cmu.edu/mycampus#Meeting")
  )
  =>
  ...(assert (triple
  (predicate "http://sadehlab.cs.cmu.edu/mycampus#availability")
  (subject ?person)(object "http://sadehlab.cs.cmu.edu/mycampus#Busy")
```

Service rules


```
<wowl:ServiceRule wowl:salience="50">
  <rdfs:label>provide location for IP Address</rdfs:label>
  <wowl:output>
    <mc:Entity rdf:ID="&variable;#entity">
      <mc:location rdf:resource="&variable;#location" />
    </mc:Entity>
  </wowl:output>
  <wowl:precondition>
    <mc:Entity rdf:ID="&variable;#entity"><mc:ip>&variable;#ip</mc:ip>
    </mc:Entity>
  </wowl:precondition>
  <wowl:call>
   <wowl:Service wowl:name="call-web-service">
     <wowl:qname>http://mycampus/WiFiService#</wowl:qname>
     <wowl:endpoint>http://128.2.68.34:7788</wowl:endpoint>
     <wowl:method>GetLocation</wowl:method>
     <wowl:ip>&variable;#ip</wowl:ip>
   </wowl:Service>
  </wowl:call>
</wowl:ServiceRule>
```

```
(defrule provide-location-for-IP-Address (declare (salience 50))
  …
    (need-dynamic_triple
     (predicate "http://sadehlab.cs.cmu.edu/mycampus#location")
     (subject   ?entity)
     (object    ?location)
    )
  …
  =>
    (call-web-service "qname" "http://mycampus/WiFiService#" (…) "ip" ?ip)
)
```

Privacy rules


```
<sowl:ReadAccessRule>
  <rdfs:label>people can only know I am on or off campus</rdfs:label>
  <sowl:target>
    <mc:Person rdf:ID="&variable;#owner">
      <mc:location rdf:resource="&variable;#location"/>
    </mc:Person>
  </sowl:target>
  <sowl:check>
   <rowl:And>
     <rowl:condition>
       <mc:E-Wallet rdf:ID="&variable;#e-Wallet">
         <mc:owner> <mc:Person rdf:ID="&variable;#owner"/> </mc:owner>
       </mc:E-Wallet>
     </rowl:condition>
     <rowl:not-condition>
       <qowl:Query rdf:ID="&variable;#query">
         <qowl:sender rdf:resource="&variable;#owner" />
       </qowl:Query>
     </rowl:not-condition>
     <rowl:condition>
       <mc:Place rdf:ID="http://www.cmu.edu">
          <mc:include rdf:resource="&variable;#location" />
       </mc:Place>
     </rowl:condition>
   </rowl:And>
  </sowl:check>
  <sowl:revision>
    <mc:Person rdf:ID="&variable;#owner">
      <mc:location rdf:resource="[illegible]"/>
    </mc:Person>
  </sowl:revision>
</sowl:ReadAccessRule>
```

http://gandon.at.home.fr/

Privacy rule: grant access to location when on campus but obfuscate ~~precision~~ truth

Query

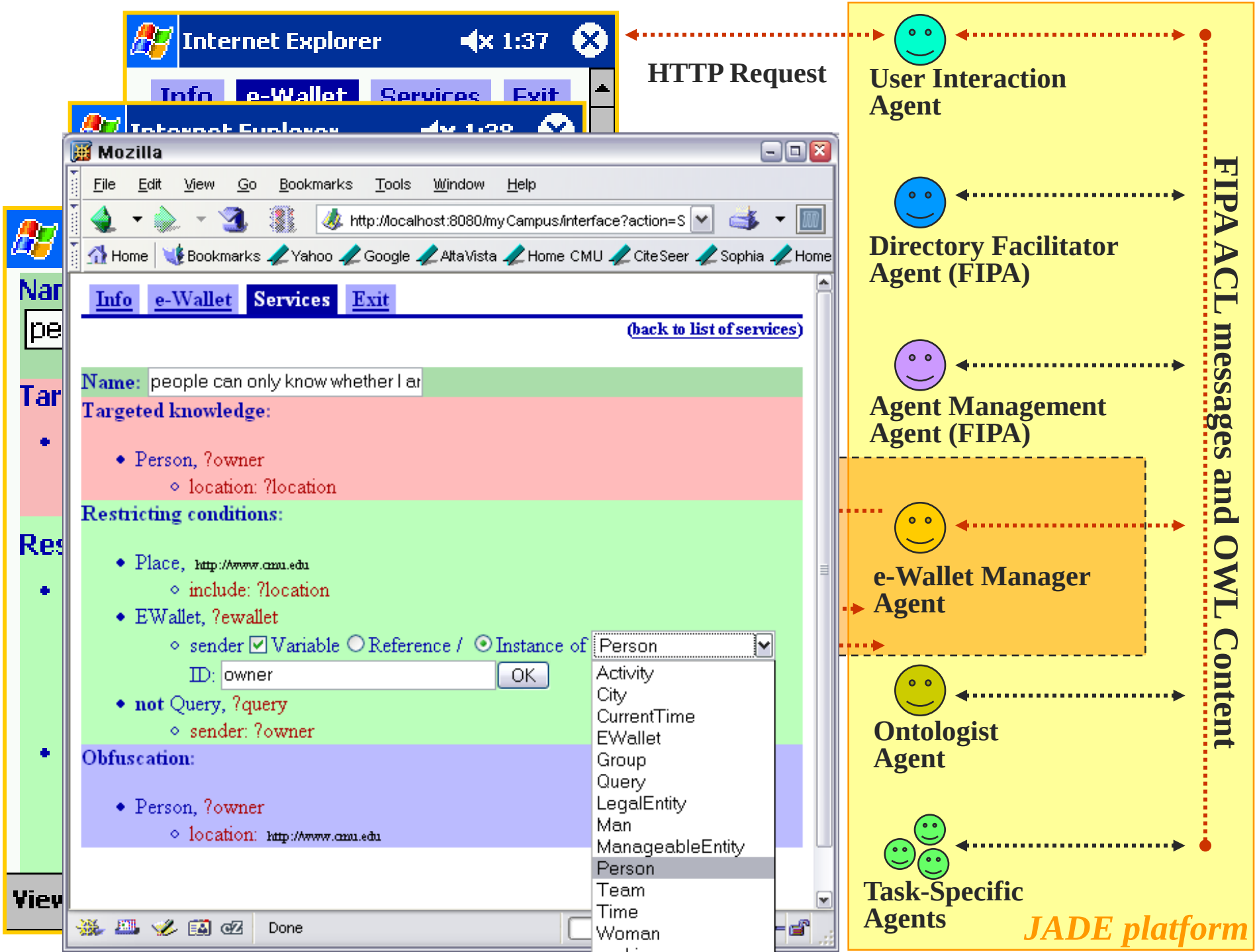

```
<qowl:Query rdf:ID="">
  <qowl:sender rdf:resource="http://cs.cmu.edu/~john"/>
</qowl:Query>
<mc:Person rdf:ID="http://cs.cmu.edu/~mary">
  <mc:location rdf:resource="&variable;#location" />
</mc:Person>
```

- Query context assertion: query sent by John

```
(triple
 (predicate "http://mycampus.cs.cmu.edu/QOWL#location")
 (subject   "")
 (object    "http://cs.cmu.edu/~john")
) ...
```

- Query rule definition
  - Body: request for authorized triples
  - Head: storage & pretty printing function

```
(defrule query (declare (salience 0))
   …
 (authorized_triple
  (predicate "http://sadehlab.cs.cmu.edu/mycampus#location")
  (subject "http://cs.cmu.edu/~mary") (object ?location))
   =>
   (store-result location ?location)
)
```

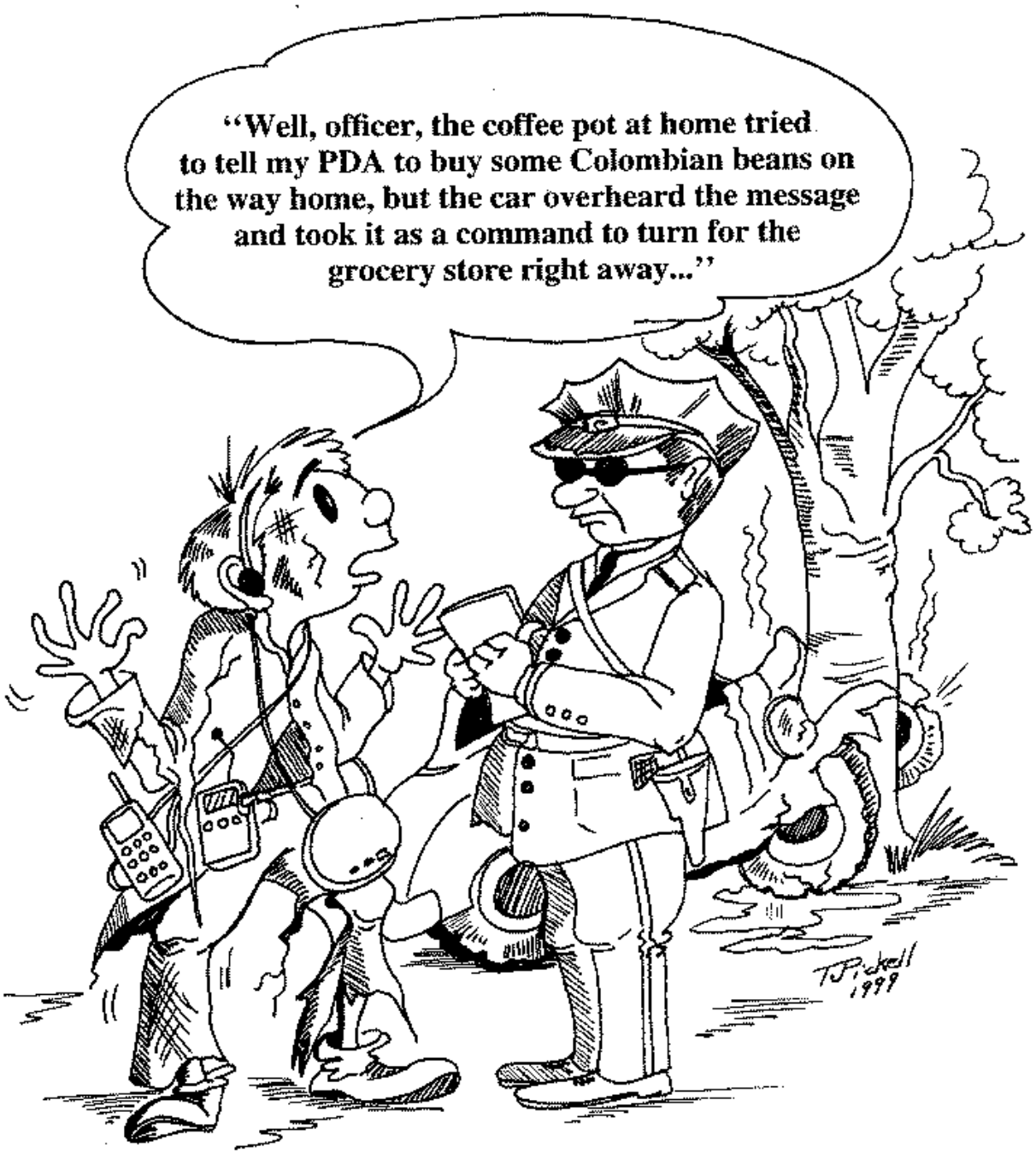

Internet Explorer
1:37
Info
e-Wallet
Services
Exit
Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/myCampus/interface?action=S
Home Bookmarks Yahoo Google AltaVista Home CMU CiteSeer Sophia Home
Info
e-Wallet
Services
Exit
(back to list of services)
Name: people can only know whether I ar
Targeted knowledge:
Person, ?owner
location: ?location
Restricting conditions:
Place, http://www.cmu.edu
include: ?location
EWallet, ?ewallet
sender Variable Reference / Instance of Person
ID: owner
OK
not Query, ?query
sender: ?owner
Obfuscation:
Person, ?owner
location: http://www.cmu.edu
Activity
City
CurrentTime
EWallet
Group
Query
LegalEntity
Man
ManageableEntity
Person
Team
Time
Woman
Done
HTTP Request
User Interaction Agent
Directory Facilitator Agent (FIPA)
Agent Management Agent (FIPA)
e-Wallet Manager Agent
Ontologist Agent
Task-Specific Agents
FIPA ACL messages and OWL Content
JADE platform

Concluding remarks

- Rather than directly publishing Web services, individuals and organizations will often want to expose a unified front-end ("**e-Wallet**") that:
  - Enforces automated resource identification logic
  - Enforces privacy/confidentiality (access & obfuscation)
- Implementation:
  - Multi-layer reasoning engine that distinguishes between different types of knowledge (core k., service invocation k., k. sanitized following application of confidentiality rules)
  - OWL & rule extension (essentially Horn clauses & variables) and rule editor
- Validation with students accessing context-aware agents on CMU's campus

Q&A

Fabien L. Gandon
Norman M. Sadeh



Source:http://www.firstmonday.org/issues/issue4_9/odlyzko/index.html