Giovanni Caire – TILAB

2003 January

JADE developments for the Te.S.C.He.T. project

ENGINEERING INGEGNERIA INFORMATICA

ISUFI
ISTITUTO SUPERIORE UNIVERSITARIO DI FORMAZIONE INTERDISCIPLINARE
e-Business Management School

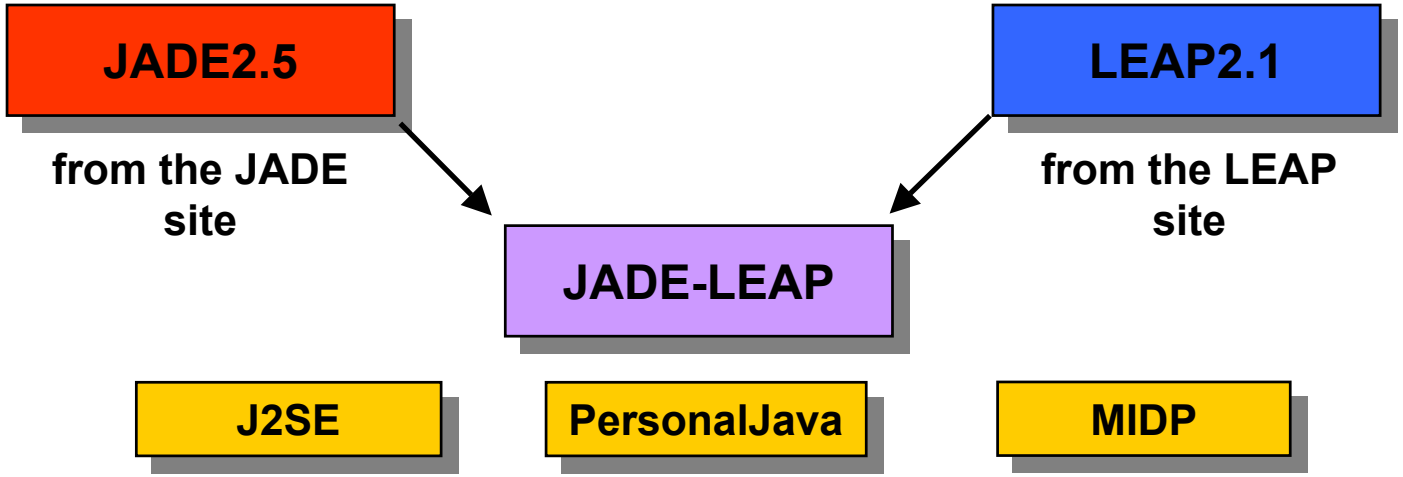
TELECOM LAB ITALIA

Outline

- Wireless
  - LEAP
  - Split container
- Expressiveness
  - Support for content languages and ontologies
  - Input to FIPA activity
- Security
  - JADE-S
  - Input to FIPA activity
- What's next

LEAP

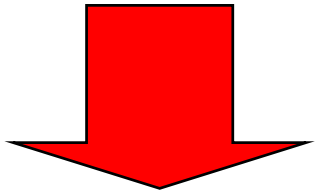
- JADE was initially conceived for the Internet
- LEAP IST project: porting to wireless handheld devices
- Since February 2002



- "Complex" process / Lots of settings needed → build errors
- Difficult to maintain → JADE2.6 no longer compatible with LEAP

JADE / LEAP integration

- With release 3.0 (expected for the beginning of March) LEAP will be integrated into JADE

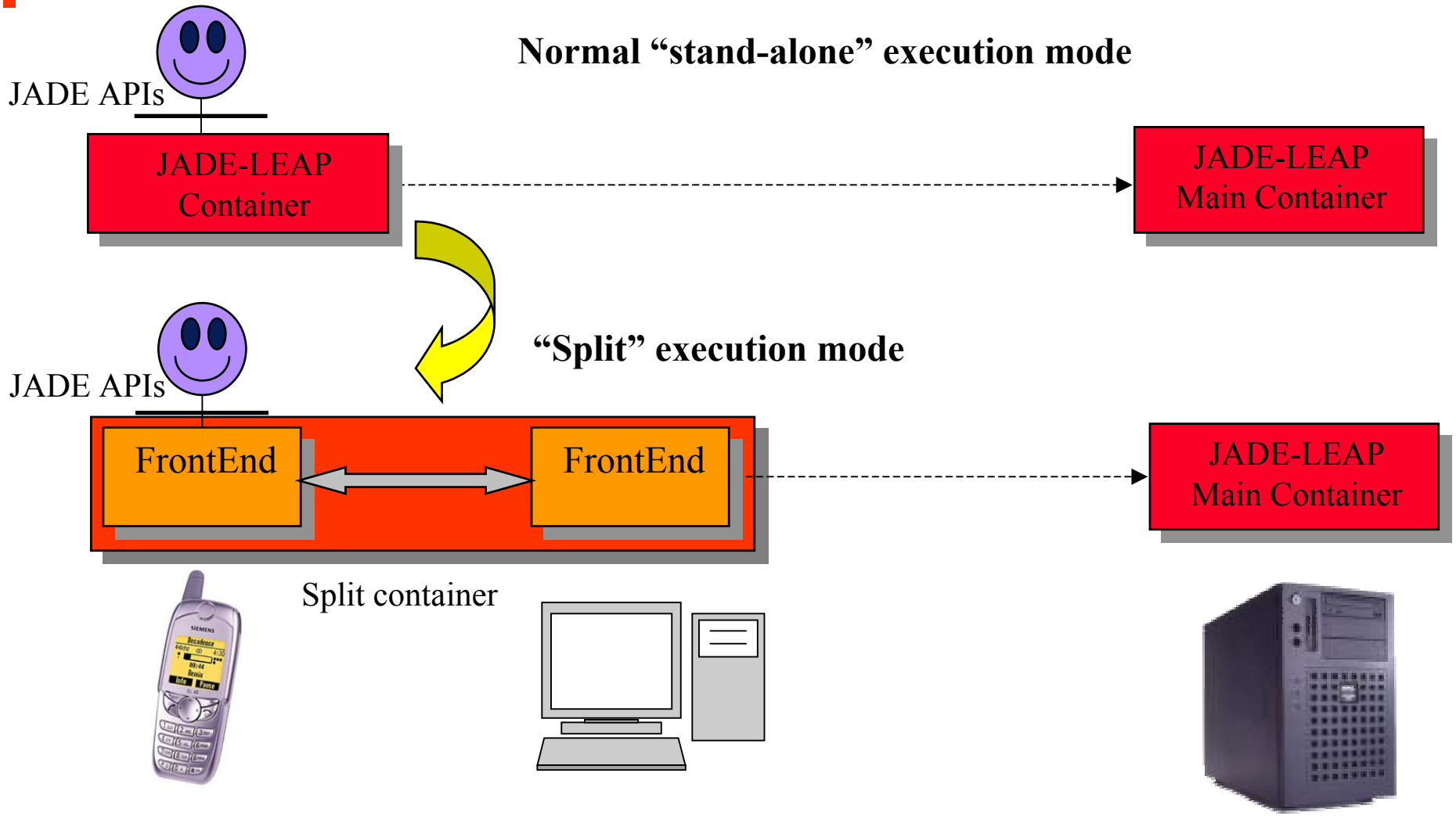


- JADE-LEAP directly downloadable in binary for from the JADE site
  - No longer need to build it
- LEAP software available as a JADE add-on
  - For users who need to modify JADE-LEAP the build process is simpler and requires less settings

Other improvements

- Simplified build environment
- Improved documentation
- Simplified configuration option
- Uniform way of launching JADE and JADE-LEAP
- New JADE features introduced since release 2.6 now available in JADE-LEAP too

“Split” execution mode



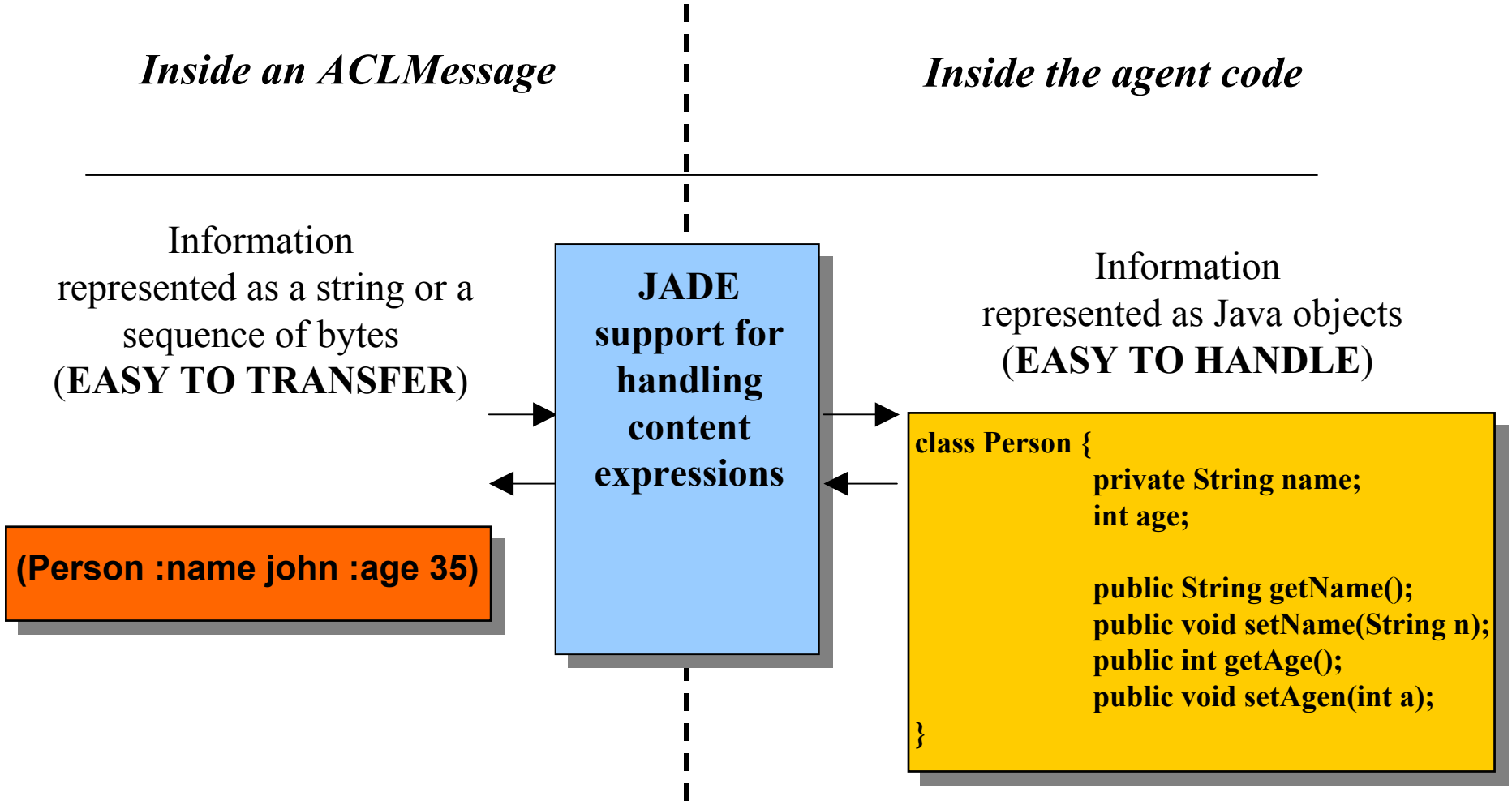
Advantages

- Less memory required
  - less classes
  - Far less objects allocated
  - With ROM-izing technique we expect to have a memory footprint < 10 KByte
- Less bytes transmitted over the wireless link
- Faster startup phase
  - 10 / 15 sec. instead of > 45 sec. over GPRS

Outline

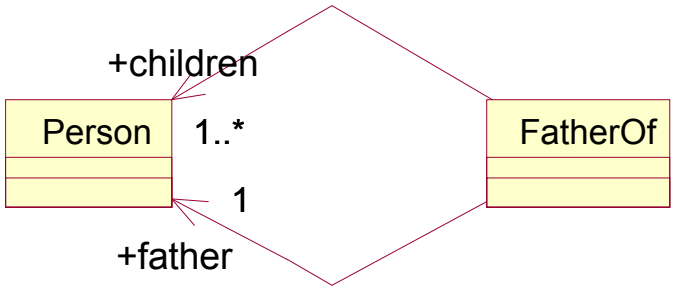
- Wireless
  - LEAP
  - Split container
- Expressiveness
  - Support for content languages and ontologies
  - Input to FIPA activity
- Security
  - JADE-S
  - Input to FIPA activity
- What's next

JADE support for handling content expressions

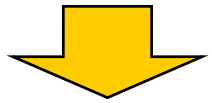


How it works

- Creating the Ontology (domain specific)
  - Defining the schemas for elements in the ontology (domain specific)
  - Defining the corresponding Java classes
  - Selecting a Content Language (domain independent)
- Creating content expressions as Java objects




```
Person john = new Person("John", 35);
Person bill = new Person("Bill", 67);
FatherOf f = new FatherOf();
f.setFather(bill);
f.addChildren(john);
```



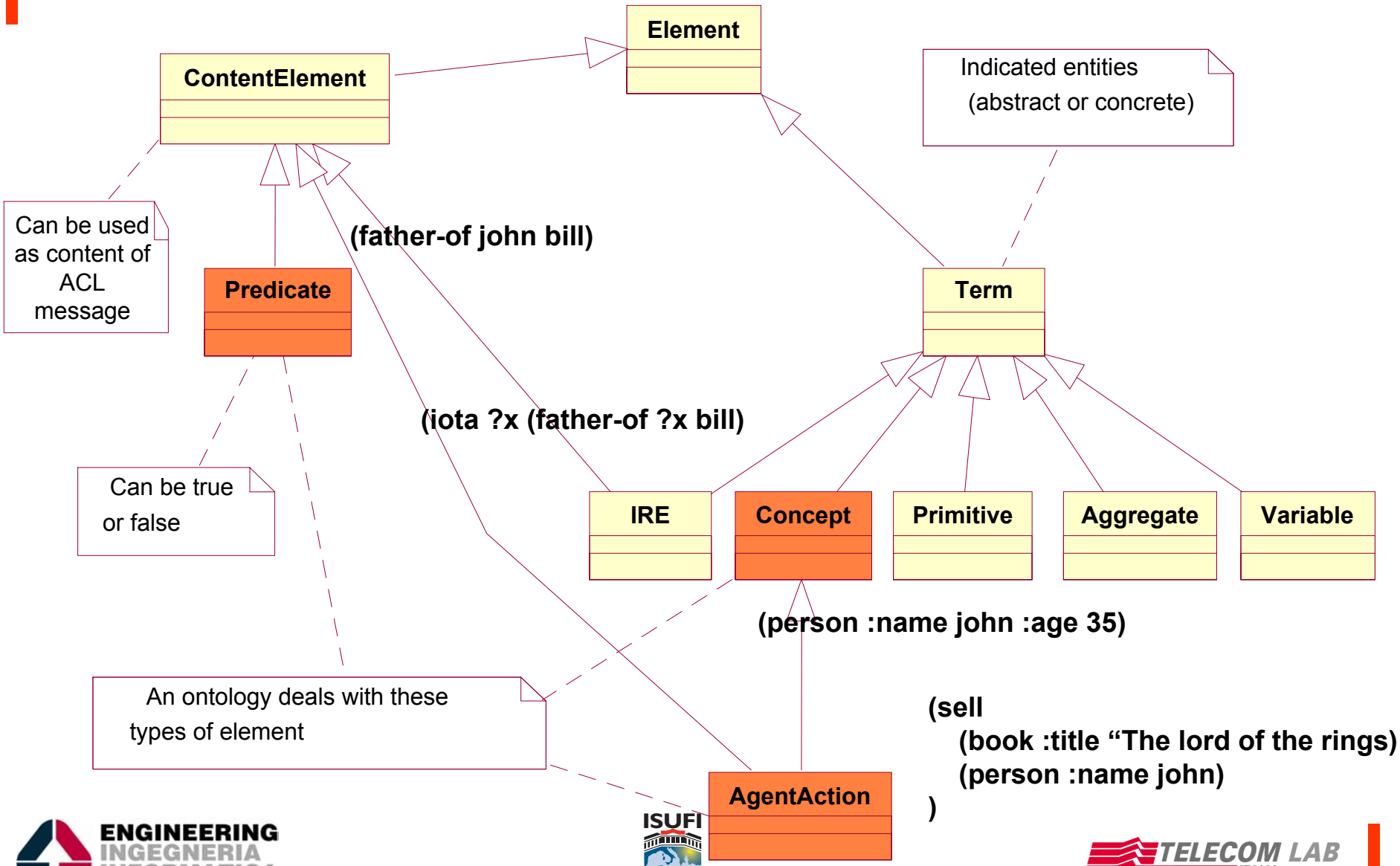
SL Language

```
(father-of (person :name Bill :age 67) (set (person :name John :age 35) ) )
```

Exploiting the expressiveness of ACL

- Content expressions are exchanged by agents within ACL messages
- The communicative intention on an ACL message intrinsically ascribes a well-defined semantic meaning to the content of the message
  - INFORM → Something that can be true or false
  - REQUEST → Something that the sender agent wants me to do
- When the content expression is extracted from the message this meaning is lost.
- We need a Content Reference Model

An ACL-based Content Reference Model



ENGINEERING INGEGNERIA INFORMATICA

ISUFI
ISTITUTO SUPERIORE UNIVERSITARIO DI FORMAZIONE INTERDISCIPLINARE
e-Business Management School

TELECOM LAB ITALIA

Input to FIPA standardization activity

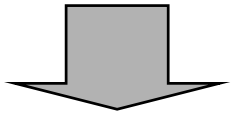
- FIPA should define a standardized Content Reference Model
- Enables easy translations between different Content Languages and between different Ontology Representation Languages → supports interoperability
- Avoids ambiguities
  - REQUEST (Person john)
- It must not necessarily be that used by JADE
- It MUST be derived from and consistent with the semantics of ACL

Outline

- Wireless
  - LEAP
  - Split container
- Expressiveness
  - Support for content languages and ontologies
  - Input to FIPA activity
- Security
  - JADE-S
  - Input to FIPA activity
- What's next

Security issues in a Multi Agent System

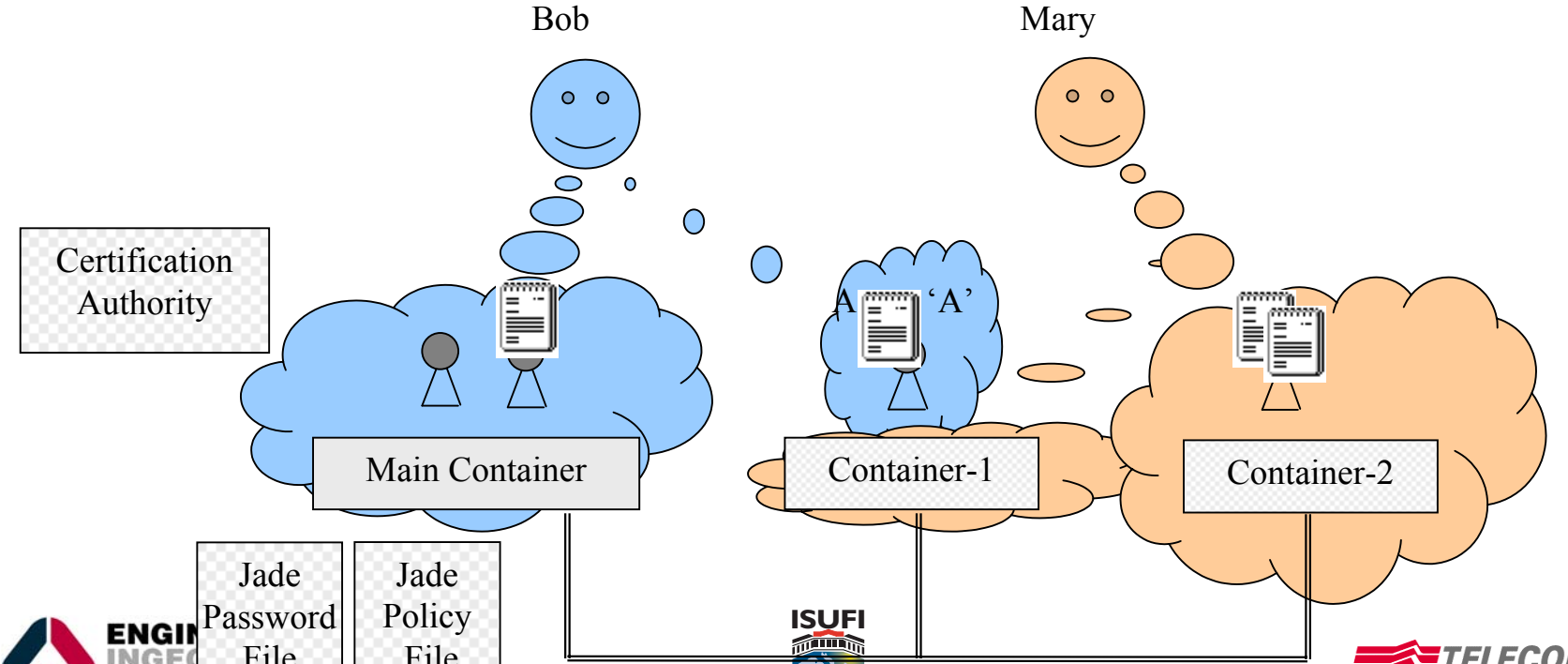
- When we have agents representing different stakeholders with conflicting interests security issues arise
- A malicious agent can
  - kill all its “competitors”
  - moves on on a competitor’s container and format the disk
  - deregister/modify registrations of “competitors” with the DF
  - send false messages
- Messages can be eavesdropped and tampered



Proper security mechanisms are needed

JADE-S: Support for security and multi-user

- Each agent is owned by an authenticated user
- Each agent has a signed certificate
- Each user/agent has certain permissions (java permissions e.g. writing a file + JADE-specific permissions e.g. CREATE_AGENT, KILL_AGENT....)
- Whenever an agent takes an action the platform checks his certificate and allows the action only if the agent has the permissions to do it.



ENGI INGEC INFORMATICA

ISUFI
ISTITUTO SUPERIORE UNIVERSITARIO DI FORMAZIONE INTERDISCIPLINARE
e-Business Management School

TELECOM LAB ITALIA

Other features

- Delegation
  - An agent can delegate at runtime some permissions he has to another agent
  - Delegated permissions can be limited in time
- ACL messages can be sent over an SSL channel

Current limitations and further developments

- JADE-S is based on the security mechanisms of JDK1.4
  - While JADE only requires JDK1.2 to run, you need JDK1.4 to run JADE with the JADE-S add-on
  - JADE-S does not work with JADE-LEAP
  - We are working to extend the security mechanisms of JADE-S to the wireless environment
- There are no permissions for DF actions
- Only JADE-specific permissions can be delegated at present
- Security mechanisms only work within a single platform

Input to FIPA standardization activity

- Inter-platform security model
  - who owns certificates
  - certification authorities
  - who authorizes actions
- What actions require permissions
- Permissions format
- What security information need to be exchanged how and when

Future activities

- Fault tolerance
  - The Main container is currently a fault tolerance bottleneck
  - Completely remove the Main container and distribute the information it holds across containers
  - Replicate the Main container so that in case of fault a new Main container arises
- Persistence
  - Agent level, container level, platform level
- Scalability
  - from thousands of agents to millions of agents

Demos

- Cinema organizer
  - Agents on mobile devices support a group of friends in deciding which film to see this evening
  - Information retrieval, automatic decision making, personalization
  - Personal Java
- Simple chat
  - Proof of concept: agents on a MIDP phone

Annexes

Ubiquitous deployment