Policy-based Access Control in Mobile Social Environments

Sara Hachem, Alessandra Toninelli, Animesh Pathak, Valérie Issarny

INRIA - Paris - Rocquencourt

France

07/06/2011

Outline

Background and Motivation

Contribution

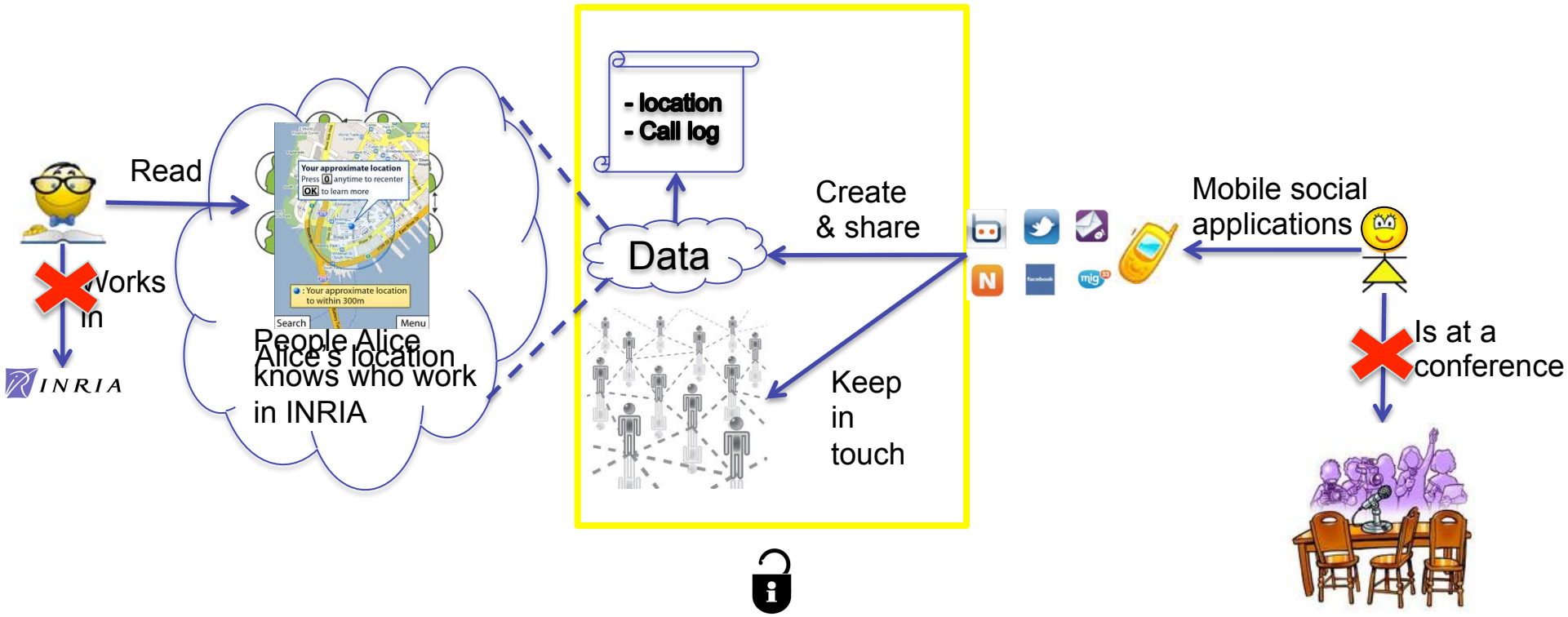
- Policy-based Access Control for Mobile Social Environments
- Integrating Access Control in a Mobile Social Middleware
- Policy Editor

Evaluation

Conclusions and Future Work

Access Control in Mobile Social Applications

Creating & sharing social information using mobile devices



Requirements for Access Control for Mobile Social Applications

Expressive and flexible policies

- "Anyone can see the list of people Alice knows" vs.
  "Bob and Tom can see Alice's friends who belong to INRIA"

Socially- and context- aware model

- Defined in terms of social concepts

Intuitive user friendly policy editor

- Any user can specify policies

Related Work (1)

Extend popular social applications with enhanced models [Kodeswaran et Al., 2010], [Shehab et Al., 2010], [Lymberopoulos et Al., 2003]

- Do not provide expressive policy models
- Do not support inference
- Do not support enforcement infrastructures

Access control based on semantic technologies

[Carminati et Al., 2009], [Elahi et Al., 2008]

- Centralized systems
- Mostly rule-based
- Expert management [Bachant at Al,1988]

Solutions relying on relation-based access control

[Giunchiglia et Al.,2008]

- Allow inference but do not provide extensible social models

Related Work (2)

Few solutions provide User-Interfaces [Wishart et Al., 2010]

- Not intuitive to users

No existing full policy infrastructures

- Proof-of-concept implementations
- Applic

Contributions

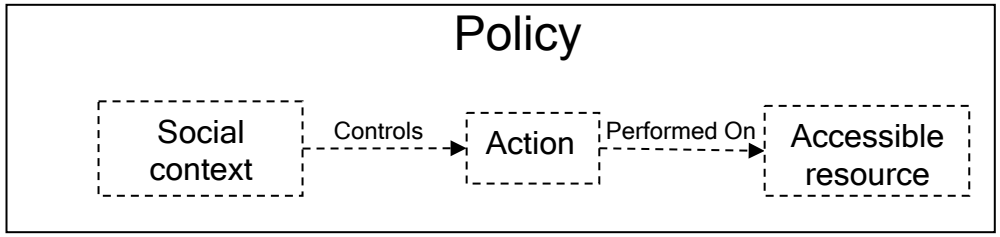
Access control model for social data in mobile environments

Integrating the model in a middleware

A graphical policy editor

Access Control Model

We use **policies** to control access to social data



3 building blocks

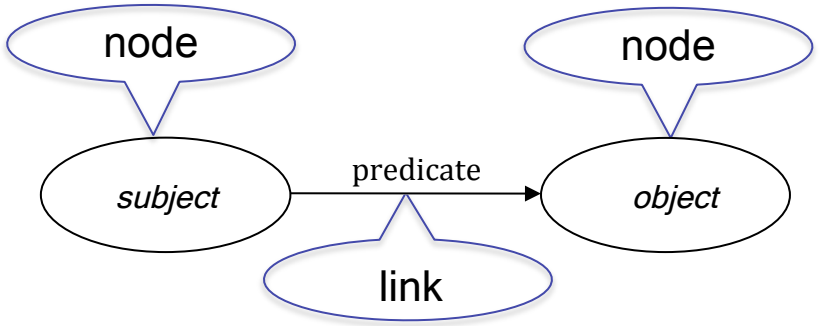
1. Accessible Resource: object to access
2. Action: the action to be performed on a resource
3. Social Context: conditions constraining access to resources, related the social situation of an entity

1. Accessible Resources

Accessible resources are social data of each user

Social data represented using Resource Description Framework (RDF) *[RDF]*:

- RDF statements:
  - <subject, predicate, object>
- RDF Triples linked together to form an RDF graph



[*RDF*] RDF. www.w3.org/RDF/

2. Actions

3 types of actions:

- Read: view triple(s)
- Add: add triple(s)
- Remove: remove triple(s)

3. Social Context

Social context: any socially related information that constrains access to a resource.

- “Bob”
- “People Alice knows”
- “People who belong to INRIA”
- “People Alice knows who belong to INRIA”

Formal Representation of Policies

Policies defined as queries over RDF data

- SPARQL query language *[SPARQL]*:

```
CONSTRUCT {%subject% %predicate% %object%.}
  WHERE{?req rdf:type ns:requestor.
        ?req ns:allowed ns:(read/add/remove).
        ns:(read/add/remove)ns:PerformedOn %subject%.
        %subject% %predicate% %object%.}
```

The query contains triples

- Subject, predicate, object known -> RDF statement
- Subject or predicate or object unknown -> SPARQL triple

[SPARQL] SPARQL. http://www.w3.org/TR/rdf-sparql-query

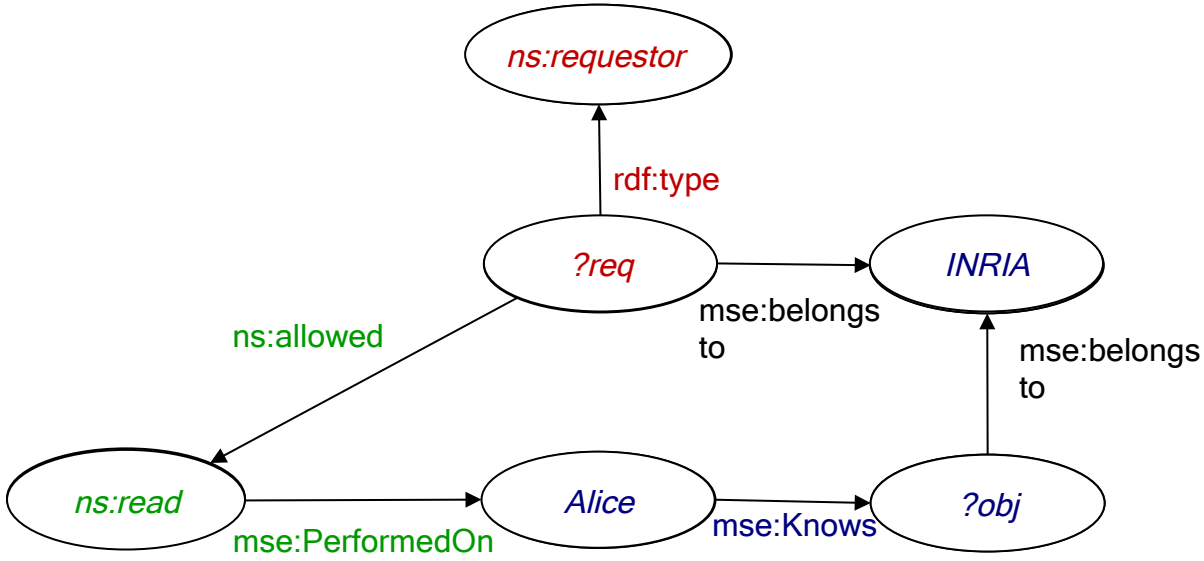
Read Example (1)

Request – Bob wishes to read from Alice's KB list of people Alice knows who are members of INRIA"

• *<Bob rdf:type requestor>*	Bob is the requestor
• *<Bob mse:allowed mse:read.>*	"Read" request
• *<Alice mse:knows ?obj.>*	People Alice knows
• *<?obj mse:belongsTo INRIA.>*	Members of INRIA

Policy: Any member of INRIA can read the list of people Alice knows who also belong to INRIA

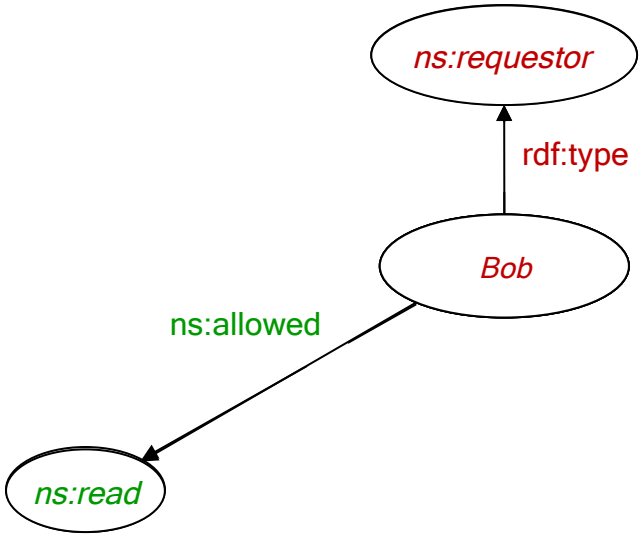
Read Example (2)



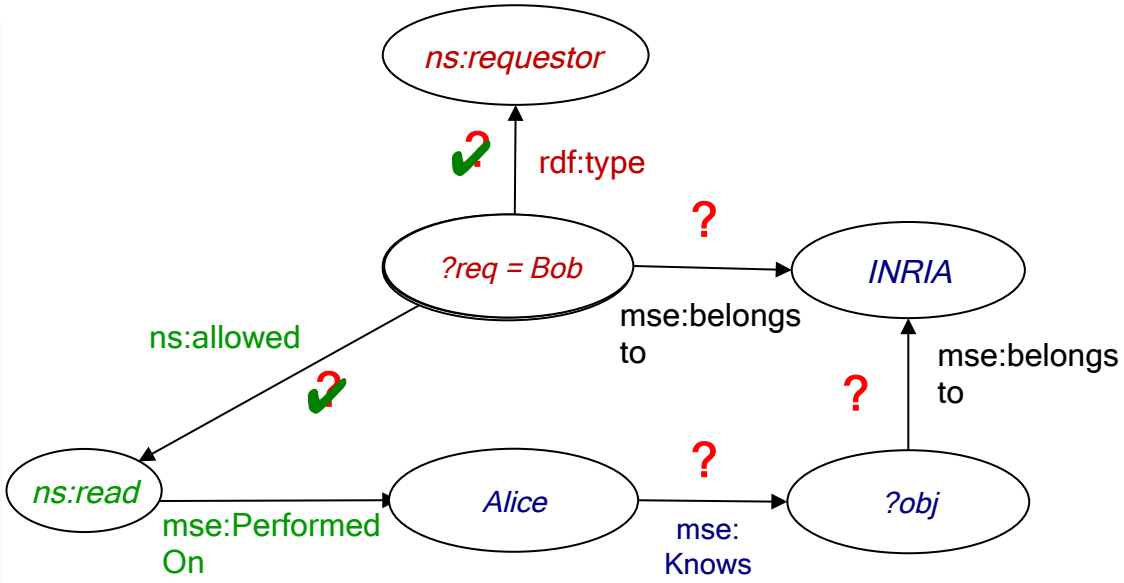
"Any member of INRIA can read the list of people Alice knows who are members of INRIA"

Read Example: Matching request against policy (1)

"Bob wishes to read list of people Alice knows who are members of INRIA"



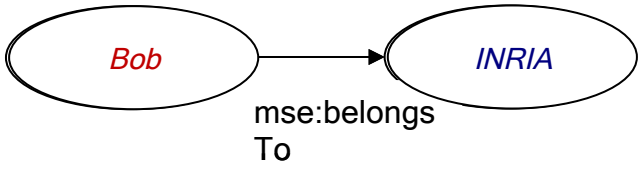
Triples added to KB



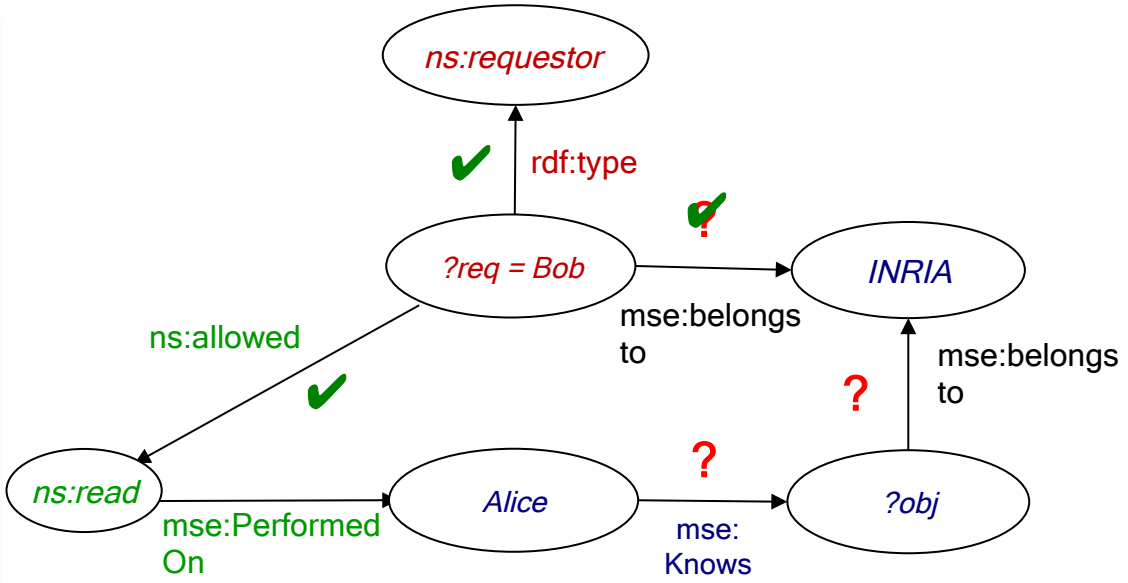
Policy after substitution

Read Example: Matching request against policy (2)

"Bob wishes to read list of people Alice knows who are members of INRIA"
Condition: the requestor should belong to INRIA



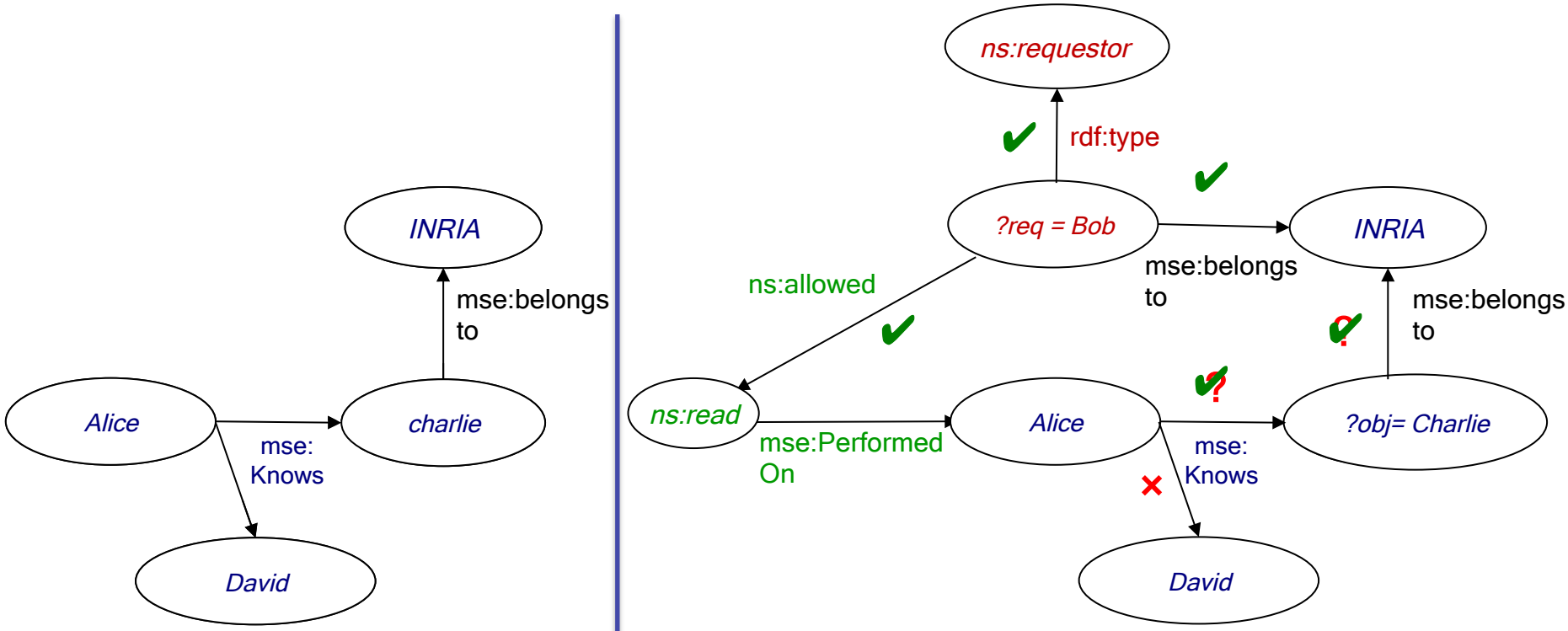
Triples found in KB



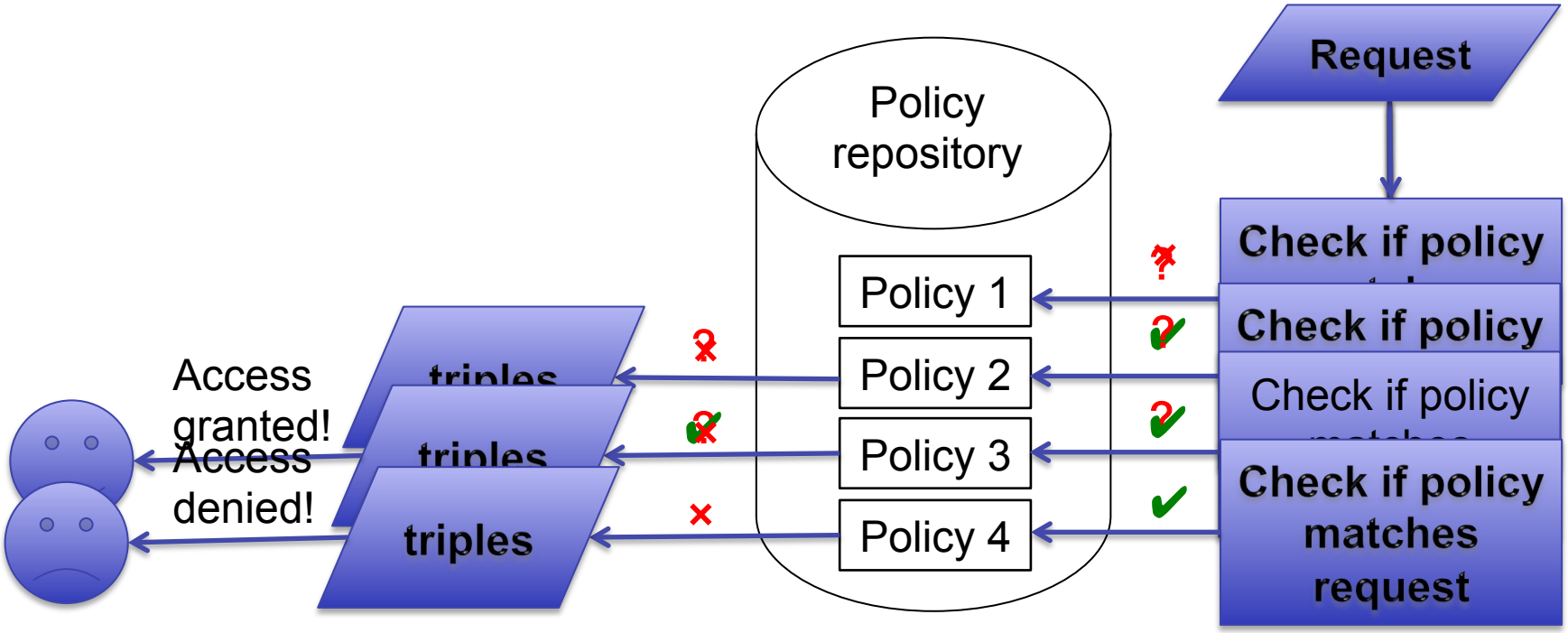
Policy after substitution

Read Example: Execute the query

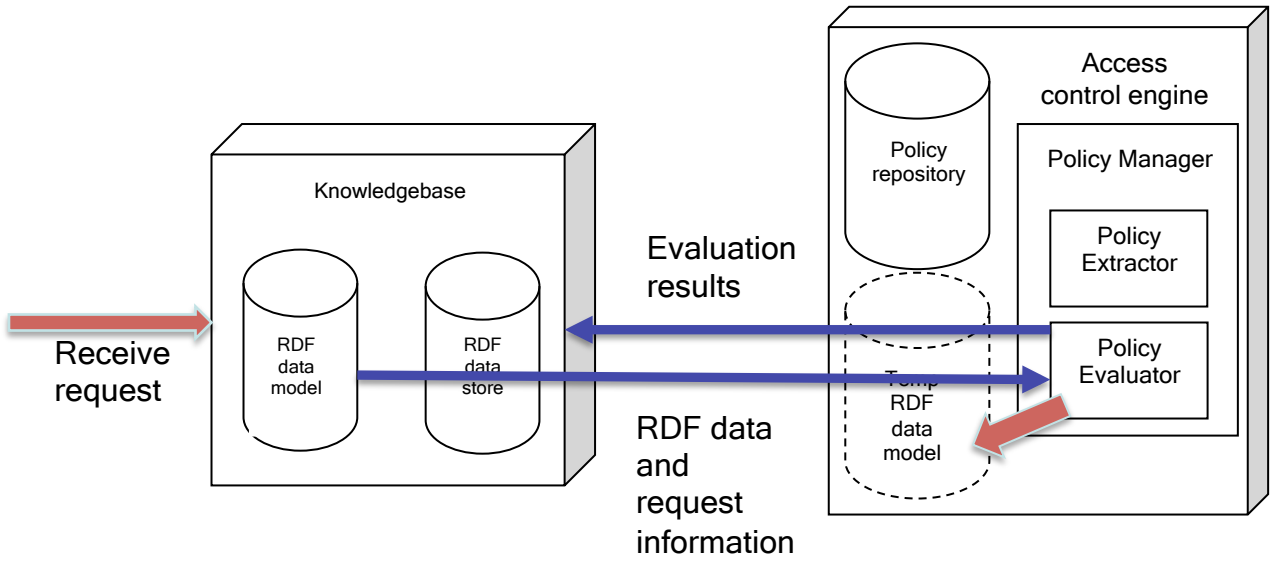
"Bob wishes to read list of people Alice knows who are members of INRIA"



Overview of Policy Checking



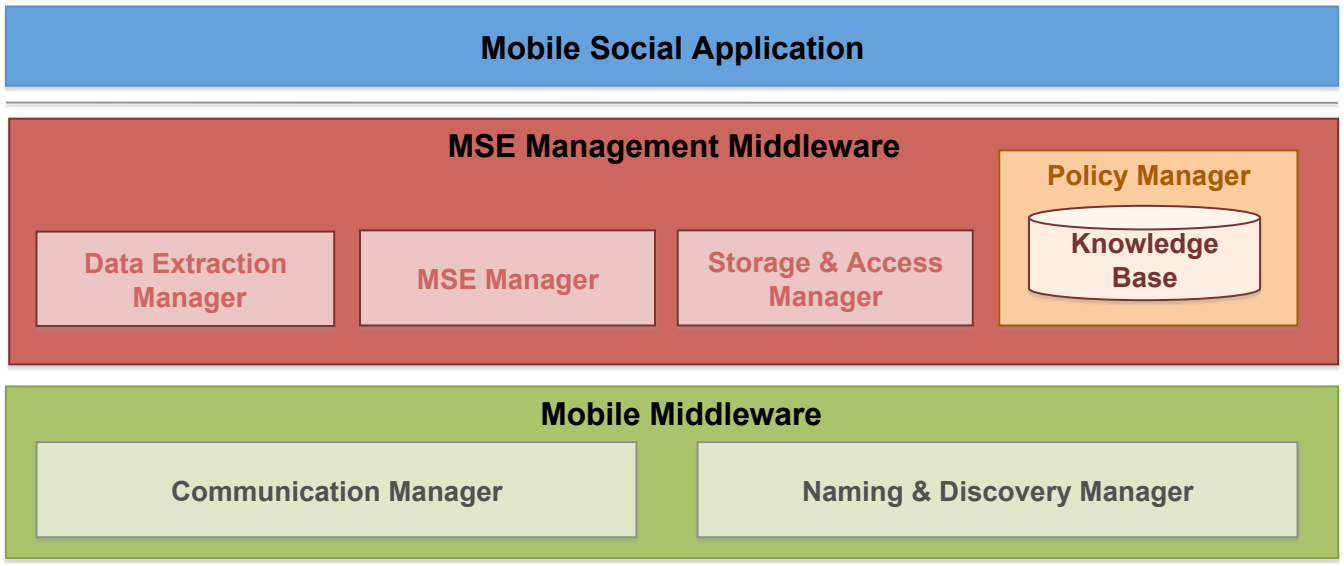
Architecture of the Access Control Component



Yarta Middleware

Our access control model is a component of Yarta developed by the ARLES team [*Yarta*]

Yarta is a middleware designed to support the development of mobile social applications



[Yarta] Yarta middleware for mobile social ecosystems. https://gforge.inria.fr/projects/yarta/

Prototype Implementation

Java2 SE

Deployed on:

- Windows XP and Mac OS X
- Android platform (v2.2)

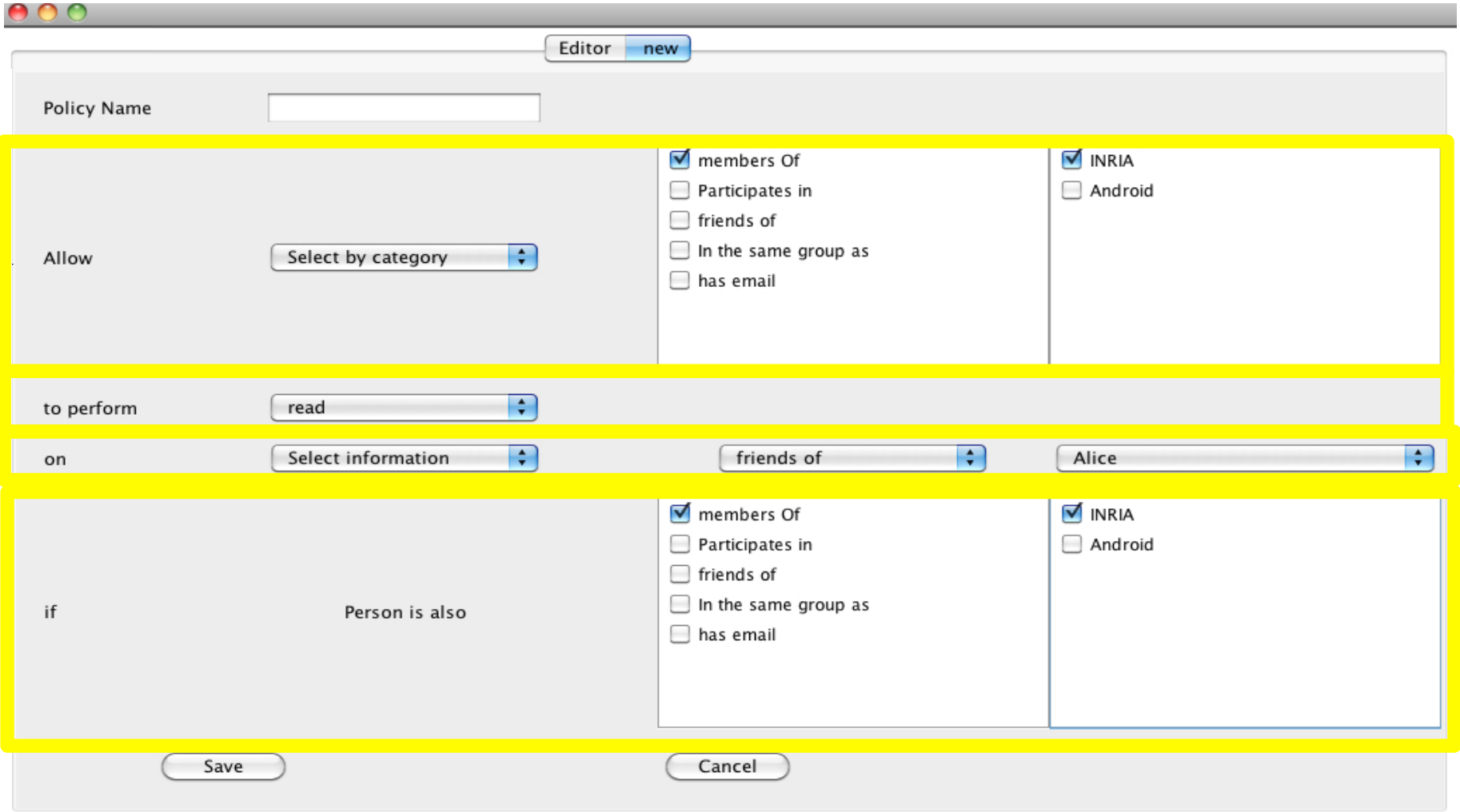
Policies saved as strings in a text file

JENA semantic web framework (v.2.6.2) to reason over the RDF data and SPARQL policies [*Jena*]

AndroJena (v.0.1) is used with Android [*AndroJena*]

[Jena] JENA http://jena.sourceforge.net/
[*Androjena*] AndroJena. http://code.google.com/p/androjena/

Policy Editor



Testbed Setup

Google Nexus One

- 512 MB of RAM
- 1 GHz Qualcomm Snapdragon QSD processor

Facebook anonymized data set *[Wilson et Al., 2009]*

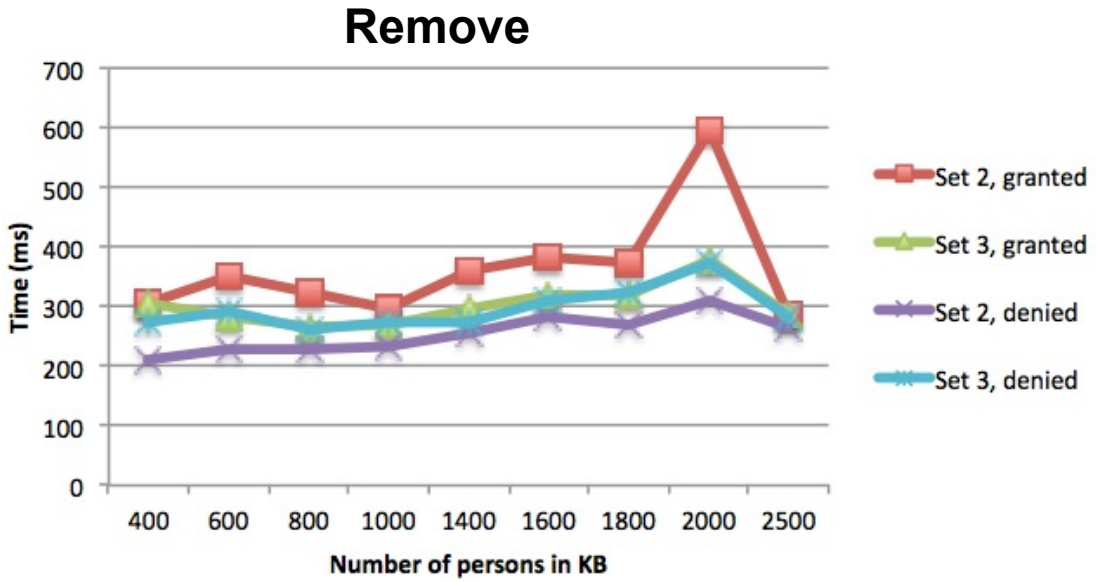
We measured time needed to execute requests

- 10 data sets and 10 runs for each set
- Variables:
  - Knowledge Base size (increasing from 400 to 2500 people)
  - Type of policy (requestor- , resource- related policies & policies with cross related elements)
  - Type of response (grant, deny)
  - Type of action (read, add, remove)

[*Wilson et Al., 2009*] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P. N. Puttaswamy, and Ben Y. Zhao, “User Interactions in Social Networks and their Implications”, 2009, Eurosys.]

Impact of the Social Data Set and Type of Policies

Linear dependence on the size of the KB for add, remove requests for any type of policies (max 600 ms).



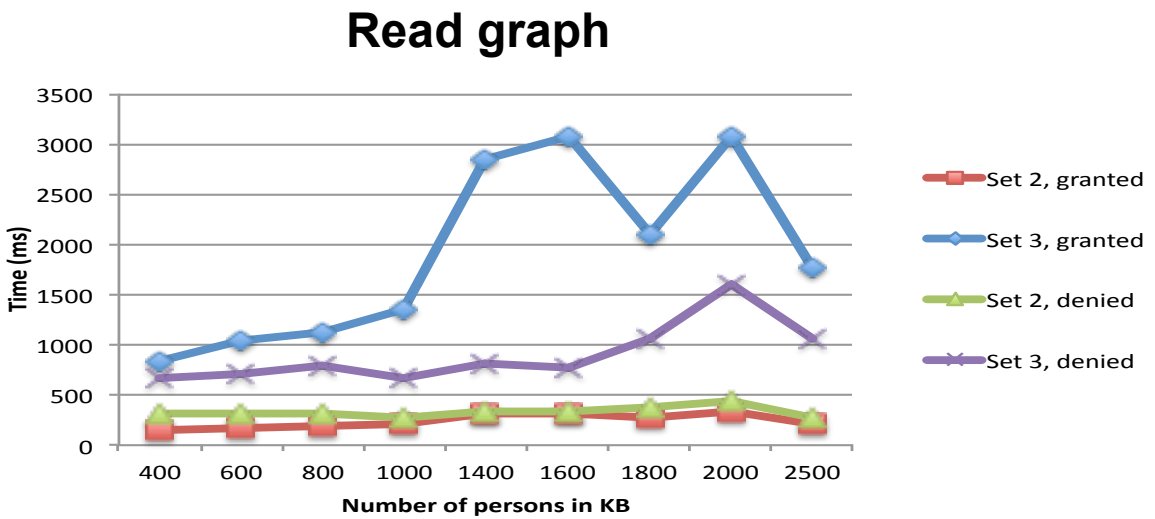
Impact of the Social Data Set and Type of Policies

For read requests

- Strong dependence on on KB & result size (max 3000 ms)

Independent of type of response

- Except read with policies with cross-related elements



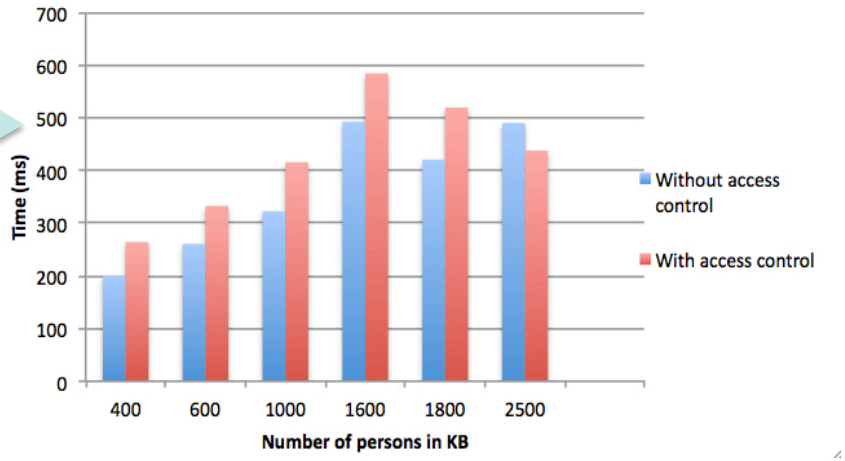
Overhead & Execution Times

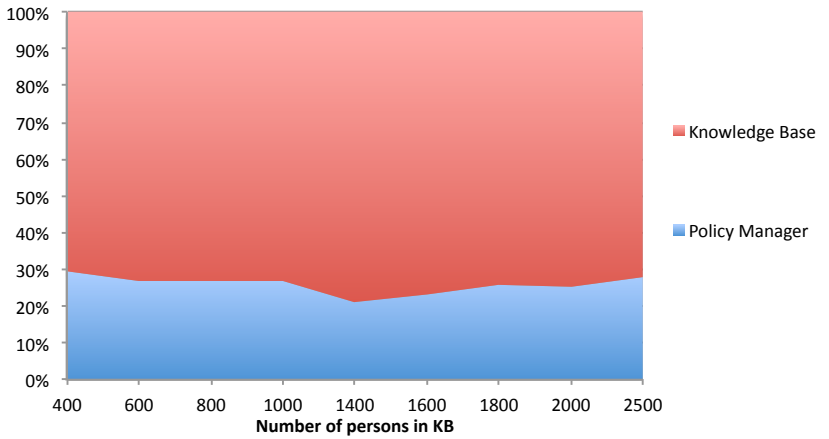
Access control results in

- Minor overhead for add
- Larger overhead for read & remove

Policy manager's share of total evaluation time:

- 30% for add
- 90% for read/remove





Conclusions

Socially aware access control model to protect social data in mobile environments

- Based on semantic technologies
- Access control can be fine-tuned to be more or less restrictive.

Java prototype integrated in Yarta middleware

- Deployed on both a laptop and on a smart phone.

Policy editor

Performance and scalability evaluation

- Acceptable overall performance & low overhead

Ongoing and Future Work

Validate framework in emerging application scenarios

Optimizing the policy evaluation process

Field testing with non-technical users

Thank you!

Sara.Hachem@inria.fr