Semantic Web Standards

Presented By: David Shelly

Topics

- SemID Ontology
- Distributed Service Deployment
- Web Services Performance

SemID (Semantic ID) Ontology

Mohammad M. R. Chowdhury

Josef Noll

Juan Miguel Gomez

UniK- University Graduate Center, Kjeller, Norway

Universidad Carlos III de Madrid, Madrid, Spain

http://www.semid.org/

Enabling Access Control and Privacy through Ontology

Mohammad M. R. Chowdhury[1], Josef Noll[1] and Juan Miguel Gomez[2]
[1]UniK-University Graduate Center, Kjeller, Norway.
[2]Universidad Carlos III de Madrid, Madrid, Spain.
[1]{mohammad, josef}@unik.no; [2]juanmiguel.gomez@uc3m.es

Abstract

The need for information security and privacy in today's connected systems is overwhelming. This paper focuses on the access control and privacy issues in a project based business environment to access project resources and to maintain privacy of members. In this regard, the SemID ontology is proposed which formalizes roles of the members, and controls access to project resources by means of formalized privacy policies and rules. The ontology is modeled from a corporate project scenario using the Protégé ontology editor platform.

1. Introduction

Project sensitive information is often contained in today's connected systems, there is an increased need for adequate security and privacy support. Access control in distributed and dynamic systems is crucial for secure service access. We believe that the capabilities of semantic technology can contribute to providing solutions to these problems. This paper proposes an ontology which addresses the access control and privacy issues of project oriented corporate networks, to secure access to project resources and to maintain the privacy of its members. To tackle these issues, we have formulated policies and rules to control access. This knowledge needs to be encoded in ways to facilitate understanding and manipulation by computers. This encoding is achieved through an ontology. Using semantic technology, we propose the SemID[1] (semantic identity) ontology to handle access control and privacy issues.

Currently, managing various forms of identities to represent people on the web is also crucial for secure service access and privacy. Chowdhury in his paper [1] proposes a concept of "Digital identity" that comprises personal (PID), corporate (CID) and social (SID) identities. SemID is expected to deal with the CIDs. It focuses on the mechanism of access control to resources and privacy assurance in a corporate environment.

The Semantic Web term was coined in [2] and is seen as the next generation information management system of the Web. Semantic Web technology is focusing on the meaning of information and on community awareness (Web 2.0). Ontologies [3] are its cornerstone technology, providing structured vocabularies that describe a formal specification of a shared conceptualization.

The impact of Semantic Web technology is wide ranging. The Project10X (a consulting firm) study found that more than 190 companies including Adobe, Google, HP, Oracle and Sony are involved in developing Semantic Web based tools [4]. But making it easier to comb through online data carries security implications. Among the challenges of security issues, policy-awareness and access control to Web resources play a major role, particularly given that these are two of the most significant requirements of information access and exchange. As a result, there has been a strong focus on access control policy languages for the Web [5]. In this paper, OWL, Web Ontology Language is used to formalize and define project member's roles, access control and privacy policies. OWL is chosen because it facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S). Among the sublanguages of OWL, we use OWL DL because of its computational completeness and decidability. This paper presents an approach to apply semantics for both policy-aware access control and privacy services in a corporate working environment.

2. Motivation

2.1 Objective

Nowadays people increasingly work in project oriented groups. Members of these groups come from different organizations. Managing access to resources through the web and ensuring privacy in projects is a big concern. Members of a group can be categorized based on the roles they play in the project. Each of the

[1] Semantic Identity (SemID), http://www.semid.org/

978-1-4244-1841-1/08/$25.00 ©2008 IEEE

Introduction

- Problems Addressed:
  - Access control in distributed and dynamic systems
  - Privacy issues in project oriented corporate networks

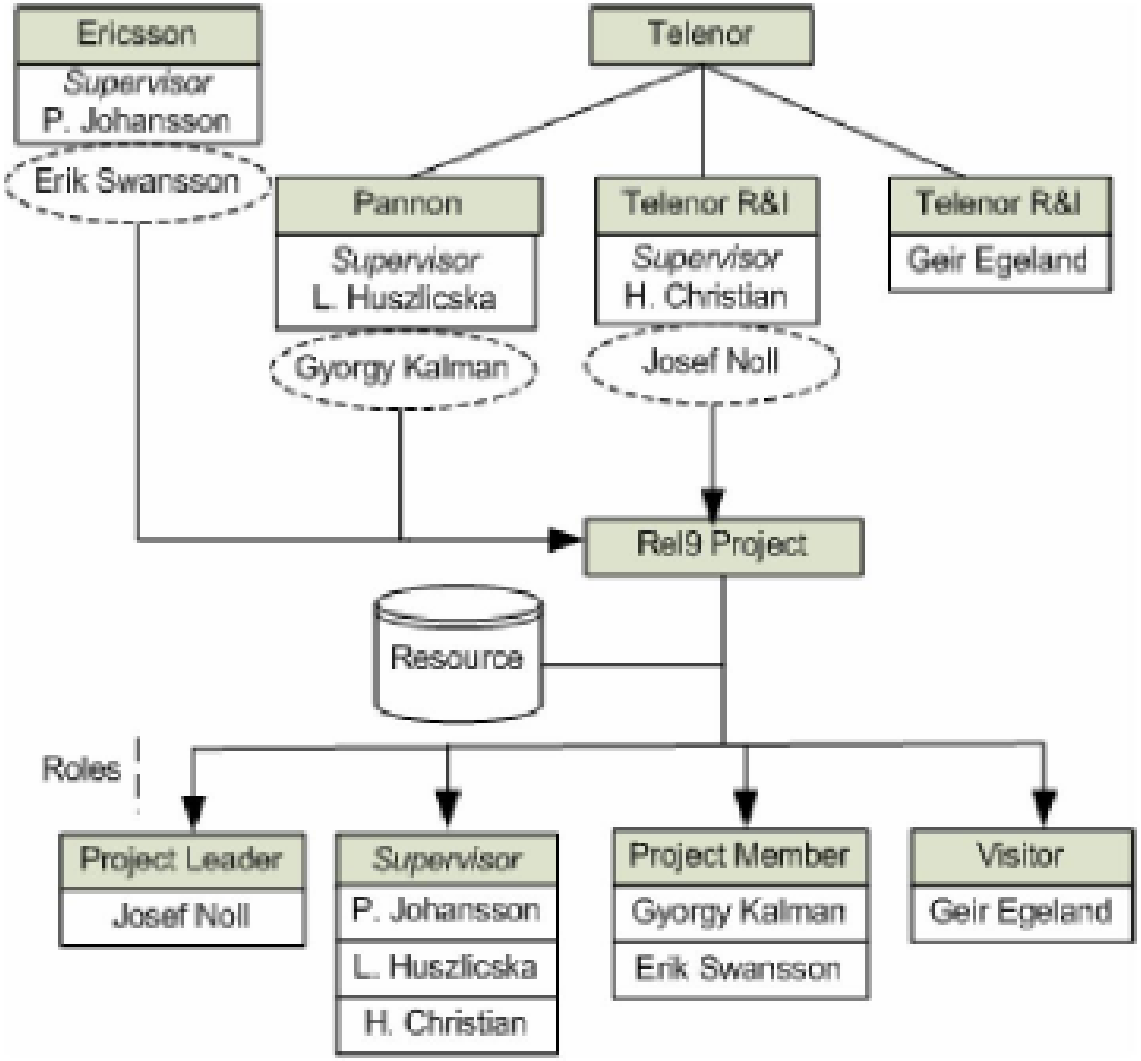
- Ontology Solution:
  - Secure access to project resources
  - Maintain privacy of members

"Ontologies are [the Semantic Web's] cornerstone technology, providing structured vocabularies that describe a formal specification of a shared conceptualization."

Roles

Project roles	Privileges	Project resources
Project leader	Administrator	Membership details
	Final Approval	Deliverables
	Read/Write	Documents
	Visibility	Member details
Supervisors	Read/Write	Deliverables
		Documents
	Visibility	Member details
Members	Read/Write	Documents
	Visibility	Member details
Visitors	Read only	Documents
	No visibility	Member details

Use case: Rel9 Project



Functional Architecture

- Formalize the semantics of roles, policies, and rules
  - Role – Has certain policy or policies assigned to it
  - Policy – Represents the privilege reserved for each role in a community and expressed through a set of Rules (R_1, R_2,… R_n)

$$P = \{R_1, R_2, \ldots R_n\}$$

  - Rules – Takes an access request as an input and results in an action (permit, deny, or not-application)

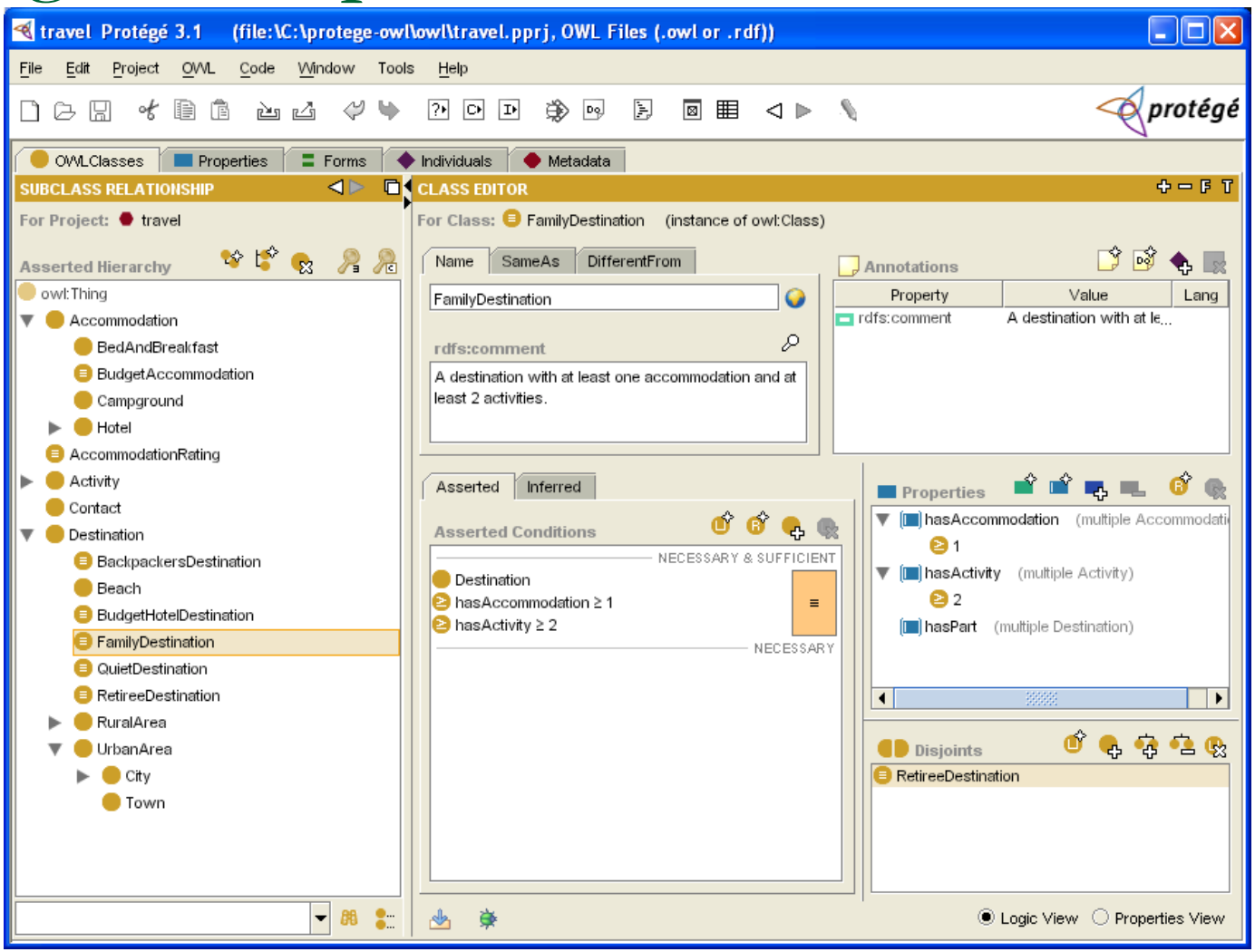
$$R = \{S, R, A\}$$

Example Rules

R = {*JosefNoll, Deliverables, Permit* }

R = {*GeirEgeland, Deliverables, Deny*}

Protégé Example



http://protege.stanford.edu/

Web Ontology Language (OWL)

- SemID Ontology has 10 properties
  - domain – classes to which a property is attached
  - range – allowed classes for properties

```
<owl:ObjectProperty rdf:ID="hasAction">
<rdfs:domain rdf:resource="#Rule">
<rdfs:range rdf:resource="#Action">
</owl:ObjectProperty>
```

Implementation

- Four different policies
  - Administrator
  - FinalApproval
  - Read
  - Read/Write

```
<Policy rdf:ID="Administrator">
<Policy rdf:ID="FinalApproval">
<Policy rdf:ID="Read">
<Policy rdf:ID="ReadWrite">
```

Implementation

- Four instances of role
  - Project Leader
  - Supervisor
  - Project member
  - Visitor

```
<Role rdf:ID="Project Leader">
<hasVisibilityOfGroup rdf:resource="#Rel9 Project">
<hasPolicy rdf:resource="#Administrator"/>
<hasPolicy rdf:resource="#FinalApproval"/>
<hasPolicy rdf:resource="#ReadWrite">
</Role>
```

Implementation

- Four properties in Identity Instance
  - hasGroup
  - hasVisibility
  - hasRole
  - hasSupervisor

```
<Corporate Identity rdf:ID="Erik Swansson">
<hasGroup rdf:resource="#Ericsson">
<hasGroup rdf:resource="#Rel9 Project">
<hasVisibility rdf:resource="#Ericsson">
<hasVisibility rdf:resource="#Rel9 Project">
<hasRole rdf:resource="#Project Member">
<hasSupervisor rdf:resource="#Peter_Johansson"/>
</Corporate_Identity>
```

Discussion

- Advantages of SemID over permissions schemes used in Windows/Linux?
- Is the SemID scheme usable? Will companies continuously update projects, roles, and permissions?

Distributed System Deployment

Artin Avanes
Johann-Christoph Freytag
Christof Bornhovd

Humboldt-Universitat zu Berlin
Berlin, Germany
SAP Labs, LLC
Palo Alto, California

Distributed Service Deployment in Mobile Ad-Hoc Networks

Artin Avanes
Humboldt-Universität zu Berlin
10099 Berlin, Germany
avanes@informatik.hu-berlin.de

Johann-Christoph Freytag
Humboldt-Universität zu Berlin
10099 Berlin, Germany
freytag@informatik.hu-berlin.de

Christof Bornhövd
SAP Labs, LLC
Palo Alto, 94304 California
christof.bornhoevd@sap.com

ABSTRACT

Today's applications are increasingly composed out of services. Standardized protocols, such as SOAP, WSDL, and UDDI are used to discover and invoke remotely located services. Nowadays, improved resource capabilities of mobile devices, e.g. PDAs, smart phones, and sensor devices allow the execution of services even on these smaller computing devices. In comparison to traditional centralized process management, a decentralized, cooperative execution of services on embedded real-time systems leads to higher system scalability, better system response time and higher data accuracy.

In this paper we describe an efficient way to deploy services onto highly distributed, mobile, and unreliable devices. To achieve an efficient resource tracking we utilize different group-based data retrieval strategies. Furthermore, we present a prototype system that implements our distributed service deployment algorithm and that evaluates our approach in terms of scalability for different network topologies.

I. INTRODUCTION

In recent years two major trends have significantly influenced and changed existing information system architectures and middleware platforms.

First, Service-Oriented Architectures (SOA) have become popular and have had an strong impact on how applications are built. Today, applications are not designed "from scratch", but are usually assembled from a set of loosely coupled, reusable application services. A (web) service represents a software system with a well-defined interface that describes the functionality of the service. The composition of individual services into more complex applications is facilitated by platform independent and machine-processable descriptions and protocols like SOAP, WSDL, and UDDI [20].

Second, the integration of small, mobile devices with sufficient computational and communication abilities has changed the way information processing takes place. In many application scenarios, a distributed execution of process services on small, distributed devices is beneficial compared to a process execution on a central backend server. A central approach does not scale well with the number of concurrent processes due to the increasing workload. Additionally, the exploitation of available device capabilities allows data aggregation and transformation of raw data within the network. As a consequence of these trends, the infrastructure of modern information systems faces new challenges:

- Increased Dynamics: Small devices, such as PDAs or sensor devices, are not fixed to one location. Instead, they can enter or leave a network at any time. Hence, possible topology changes need to be tracked by higher system layers.
- Limited Resource Capabilities: Even if these small devices possess advanced abilities to execute services, their resource capabilities (e.g. battery power or memory capacity) still remain limited. Sophisticated data dissemination strategies are required to reduce resource-consuming operations like sending and receiving of message packets.
- Limited Reliability: Devices may run out of resources or the communication infrastructure may temporarily be broken. The quality and quantity of collected data can vary at different points in time.
- Higher Demand for Scalability: The infrastructure must scale with the number of devices in use and services to be deployed. With a growing number of devices, more messages are routed through the network to reach devices at the network edge. In-network filtering and processing techniques are necessary to reduce redundancy, the number and the size of data packets and thus saving energy.

This paper focuses on efficiently deploying services onto participating devices of a mobile, ad-hoc network. The distributed service deployment is based on a tiered system infrastructure that connects stronger, reliable components of a traditional information system with smaller, unreliable devices of a dynamic network. We implemented a prototype system and integrated it into SAP's Smart Items Infrastructure (SII) [15]. SII was developed by SAP Research and supports the composition and deployment of distributed ubiquitous computing applications.

In summary the contributions of this paper are:

- an algorithm for deploying services to mobile, resource-constraint and unreliable devices in a distributed environment;
- different strategies for an efficient, adaptable retrieval of device profiles in dynamic networks;
- an implementation and evaluation of our approach regarding scalability for different network configurations.

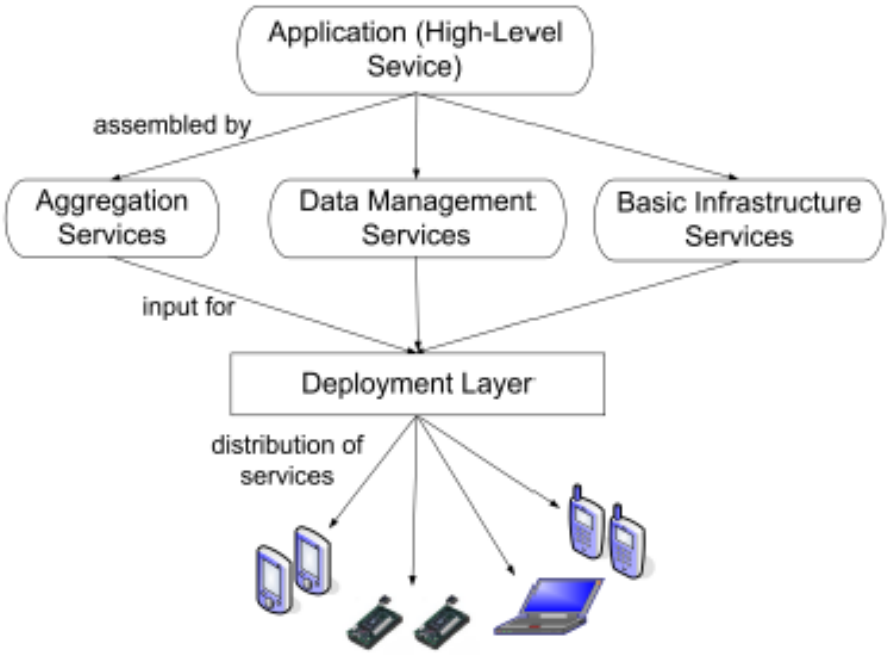
The remainder of this paper is organized as follows. In Section 2 we describe the main building blocks of our system architecture and discuss the different roles of each component. We explain the distributed service deployment in Section 3. Depending on the deployment request, we propose several re-

Introduction

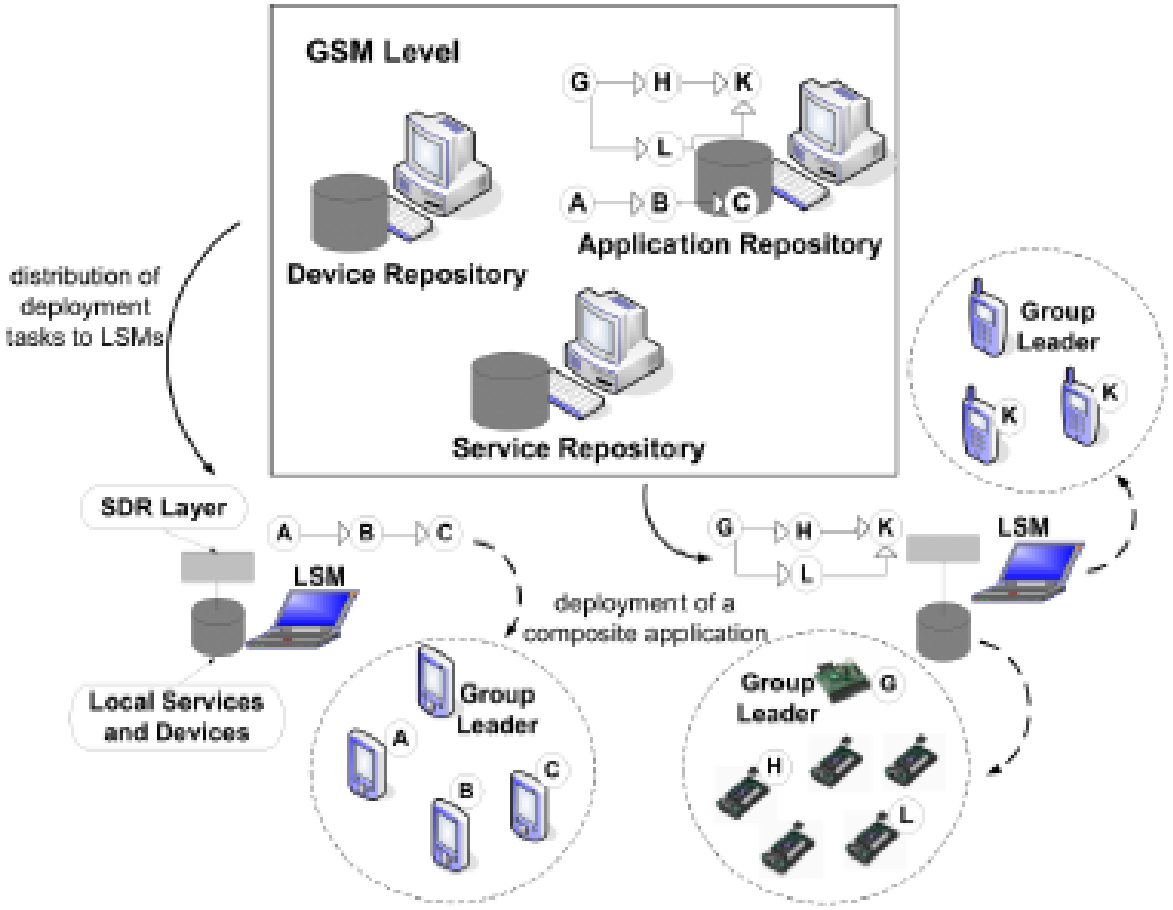
- Advantages of Distributed Service Deployment
  - Higher system scalability
  - Better system response time
  - Higher data accuracy
- New Challenges
  - Increased Dynamics
  - Limited Resource Capabilities
  - Limited Reliability
  - Higher Demand for Scalability

Service Classes

- Three Major Service Classes
  - Business Logic Services
  - Aggregation Services and Data Management Services
  - Basic Infrastructure Services



Tiered System Architecture



Service Deployment

- Service Mapping
- Context-Aware Determination of Service Requirements
- Group-Based Resource Tracking
- Priority Assignment and Query Processing
- Distributed Service Injection

Service Deployment

- Service Mapping
  - Mapping Function
- Context-Aware Determination of Service Requirements
- Group-Based Resource Tracking
- Priority Assignment and Query Processing
- Distributed Service Injection

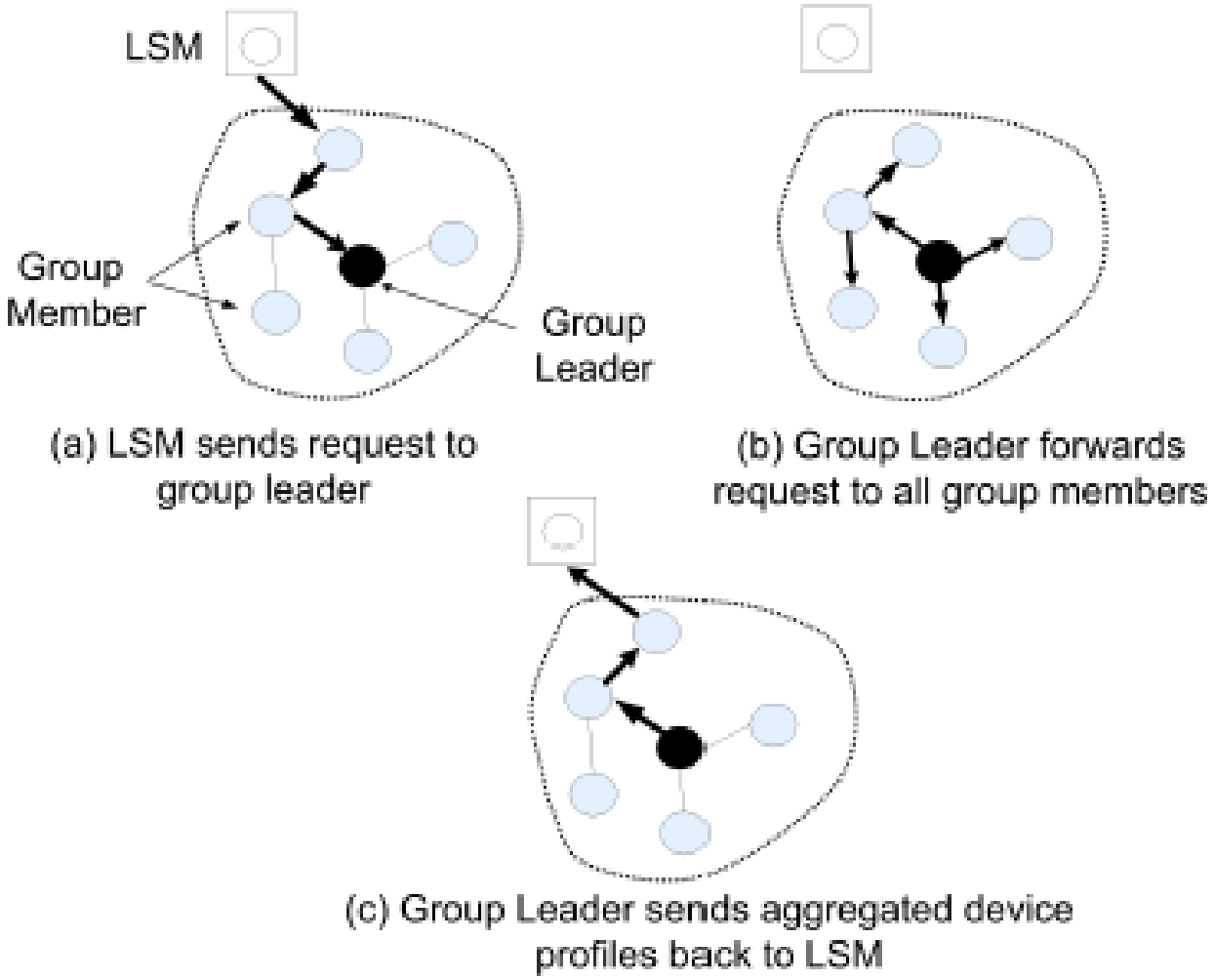
Service Deployment

- Service Mapping
- Context-Aware Determination of Service Requirements
  - Translation Process
- Group-Based Resource Tracking
- Priority Assignment and Query Processing
- Distributed Service Injection

Service Deployment

- Service Mapping
- Context-Aware Determination of Service Requirements
- Group-Based Resource Tracking
  - Group-Based Retrieval Algorithm
- Priority Assignment and Query Processing
- Distributed Service Injection

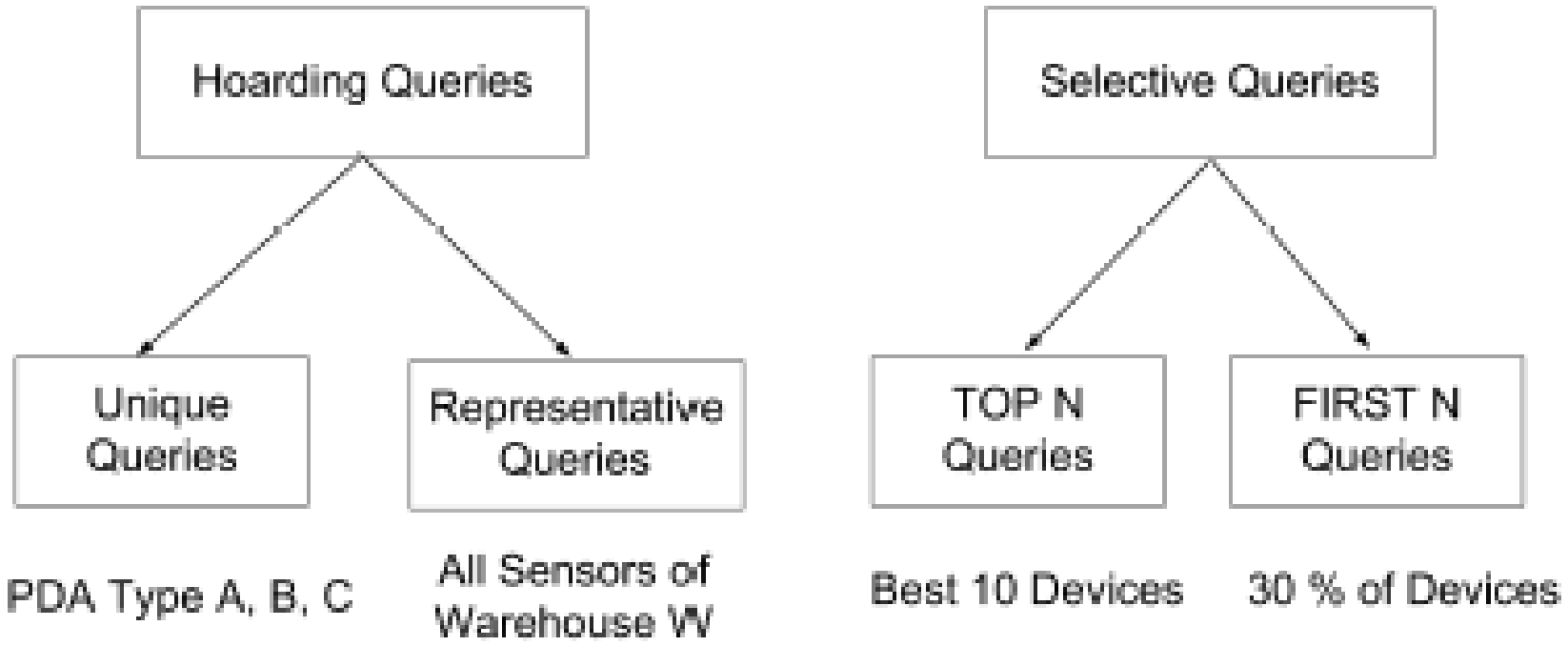
Group-Based Retrieval Algorithm



Service Deployment

- Service Mapping
- Context-Aware Determination of Service Requirements
- Group-Based Resource Tracking
- Priority Assignment and Query Processing
  - Two major request classes
  - Three strategies to determine priorities
- Distributed Service Injection

Priority Assignment and Query Processing



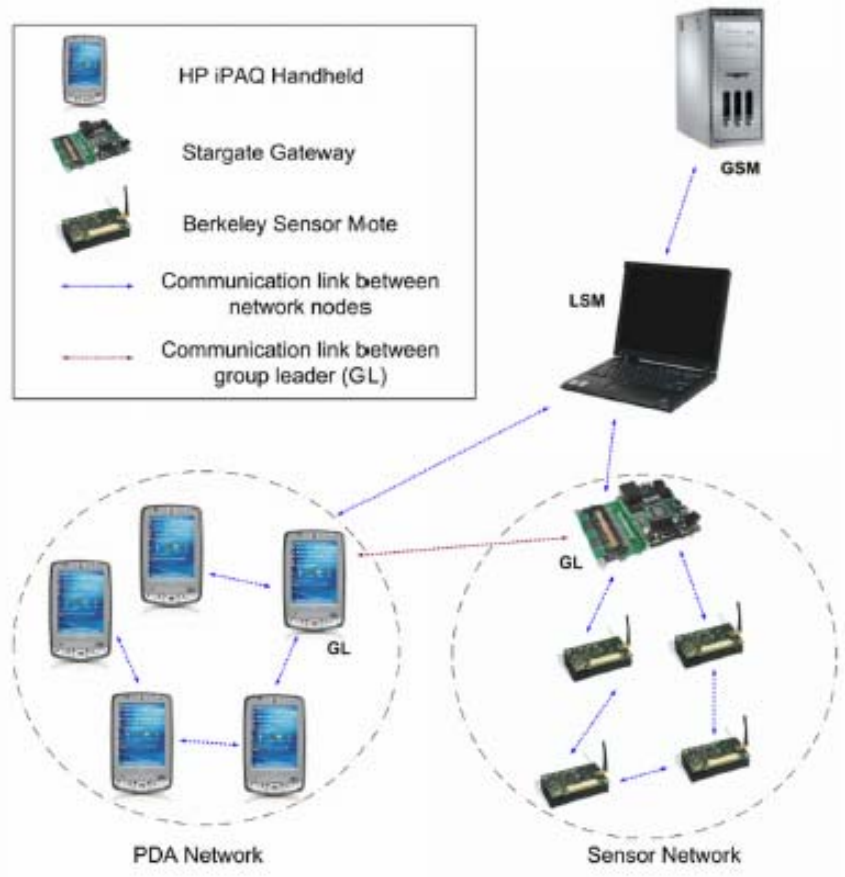
Service Deployment

- Service Mapping
- Context-Aware Determination of Service Requirements
- Group-Based Resource Tracking
- Priority Assignment and Query Processing
- Distributed Service Injection
  - Pair matching

OSGi Prototype Implementation

- Example Scenario:

  "A wireless sensor network measures the temperature in specific areas of a warehouse, whereas the current temperature values are periodically forwarded to the display of a worker's PDA. Each worker is equipped with such a PDA and can immediately react if the temperature exceeds a certain threshold to avoid damage of goods or machines."



Performance Evaluation

- List Topology

$$\leq O(N) + O(N^2 + N * H) \rightarrow\!\!\ll O(n^2 + n * H)$$

(no packet merging)

$$\leq O(N) + O(N) \rightarrow\!\!\ll O(n)$$

(with packet merging)

- Star Topology

$$\ll O(n)$$

- Binary Tree Topology

$$\rightarrow O(\ln N * N^{\ln(2)}) \leq O(\ln N * N^{0.7}) \ll O(\ln n * n^{0.7})$$

Discussion

- How could using a Distributed System Deployment in Mobile Ad-Hoc Networks apply to usable security?
- What privacy issues are at risk in distributed system deployment schemes?