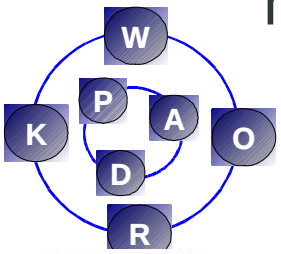
ESCAPE: An Adaptive Framework for Managing and Providing Context Information in Emergency Situations

Hong-Linh Truong, Lukasz Juszczyk, Atif Manzoor, Schahram Dustdar

Distributed Systems Group, Vienna University of Technology

truong@infosys.tuwien.ac.at

http://www.vitalab.tuwien.ac.at/autocompwiki



European Commission

Information Society

VITALAB Distributed Systems Group

Outline

- Scenario, requirements and motivation
- Approach
- Levels of context information
- ESCAPE framework overview
- Implementation status
- Experiments and current applications
- Conclusion and future work

Scenario

- Emergency situations
  - E.g., Natural disasters (earthquakes, hurricanes, floods)
- Multiple support teams
  - Established on demand
  - Deployed at front-end (disaster fields) and back-end sites
  - Conduct various processes to respond to the emergency situation
    - Collecting information, performing relief tasks, etc.

→ How to achieve effective response processes?

Requirements and Motivation

- Response processes to emergency situations, such as natural disasters:
  - Established on demand and changed rapidly, depending context of the situations

→ context information associated with entities inherent in the situation is critical to effective response processes

- Context information in emergency situation is complex
  - Reflects situation, responses, sites, teams and individuals
  - Related to people, resources and services, their status and activities.

Requirements and motivation(cont.)

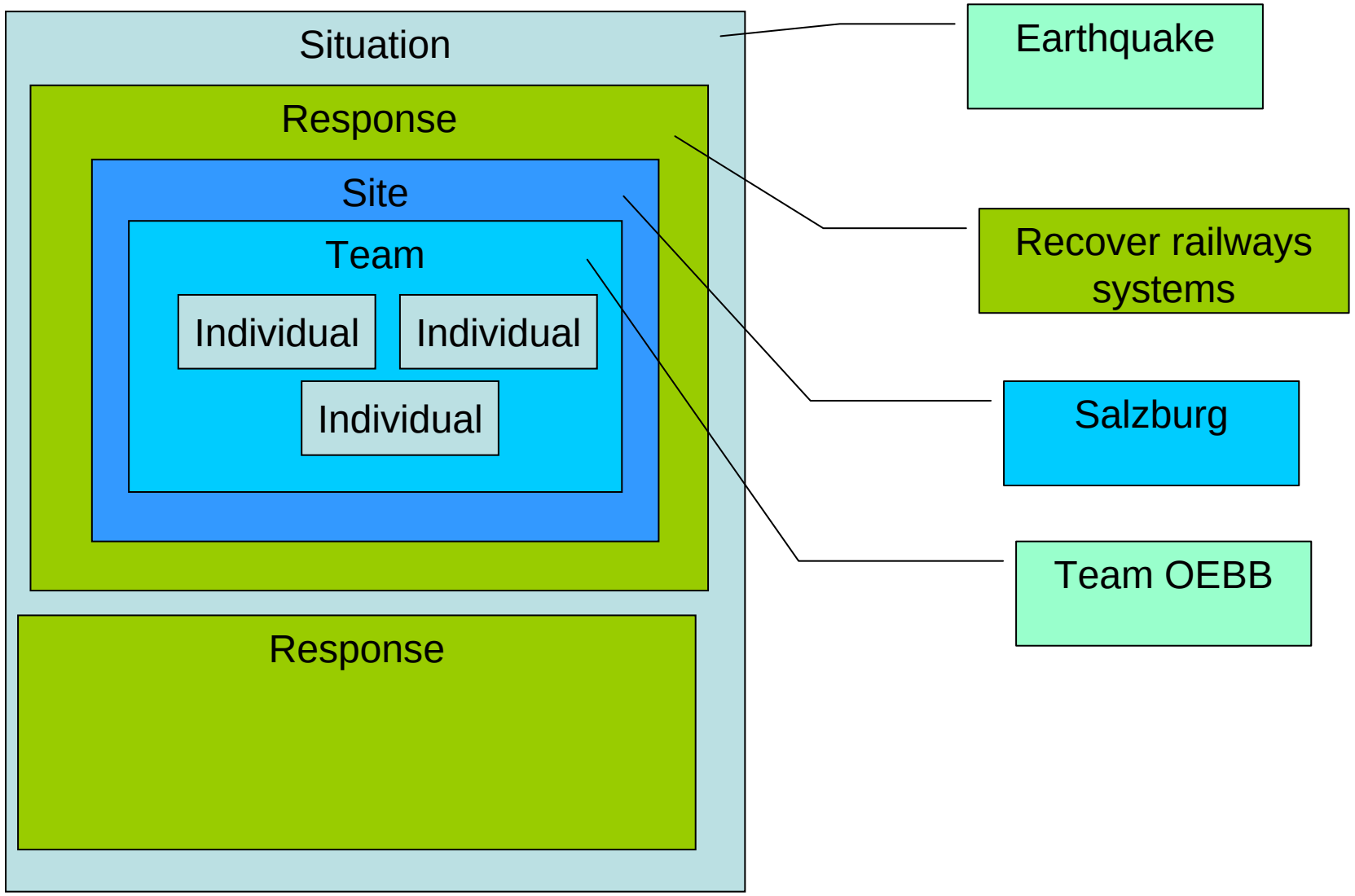
- Many context management frameworks exist
  - Mostly for normal environments, domotics, HCI for emergencies,lack of integration with crisis management systems
- Some scientific and engineering issues: how to
  - Support multiple teams collaborating in pervasive Grid environments
  - Make context information available to many crisis management supporting tools and services
  - Ensure that context information in emergency situations are extensible and interoperable
  - Runtime interpretation and engineering of context info

Our Approach

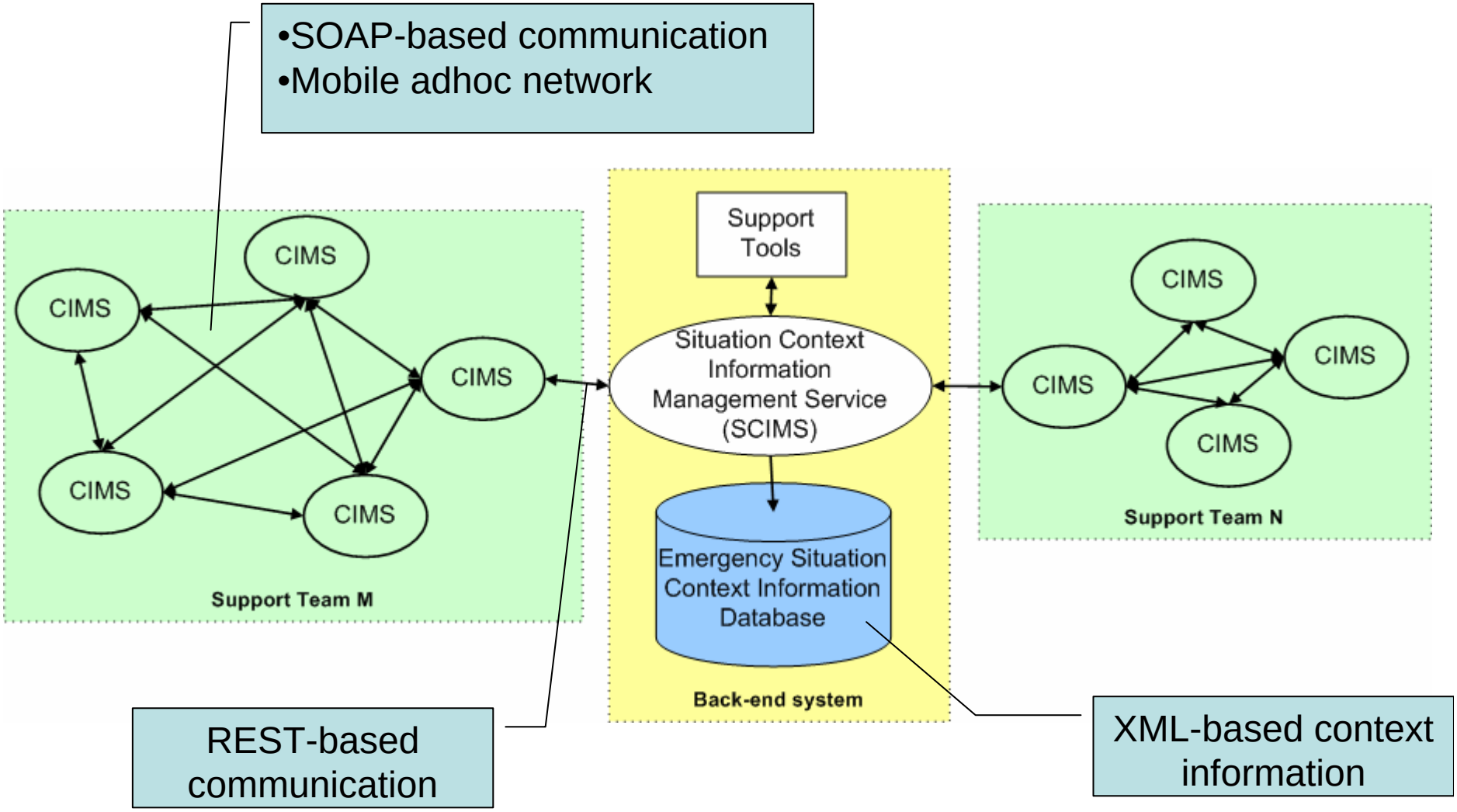
- Integration with other tools/services
  - SOAP-based and REST (Representational State Transfer)-based models
  - Different techniques for front-end and back-end integration
- Extensible context information representations
  - Support XML-based context information
  - Accept any representations as long as the information is in XML
- Flexibility in pervasive environments
  - Customizable software components
- Tracing capability
  - Context provenance

→ This paper focuses on middleware aspects

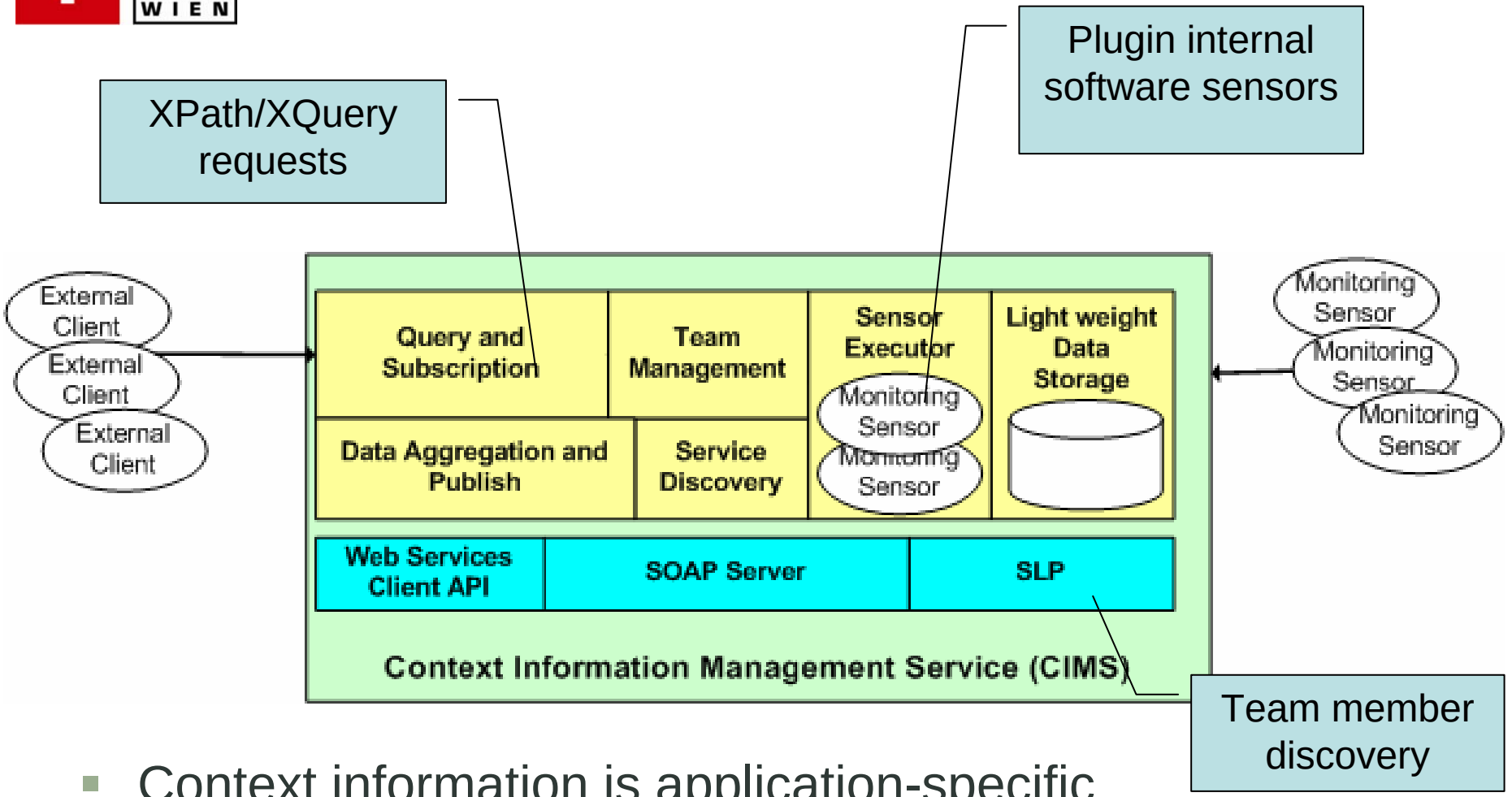
Levels of Context Information in Emergency Situation



ESCAPE Framework Overview



ESCAPE Framework Overview

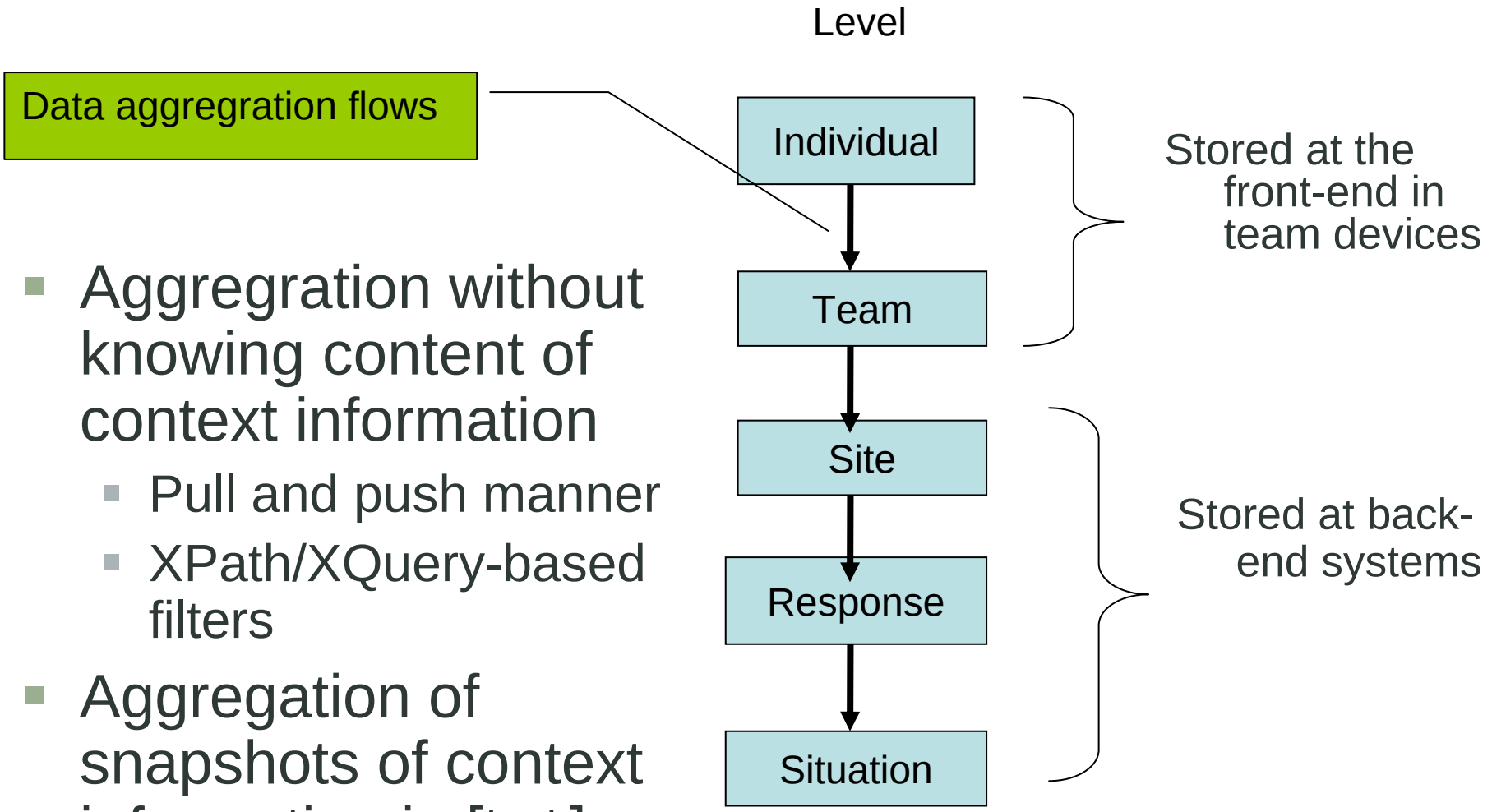


- Context information is application-specific
- The middleware accepts any XML-based context information

Query/Subscription and Storage

- Query and Subscription APIs
  - Core APIs support requests based on XPath/XQuery
  - Application-specific query and subscription APIs built atop core APIs
- Lightweight Data Storage
  - High-end devices: eXist database with XQuery, XUpdate, etc.
  - PDAs: round-robin model in which XML data is stored in files

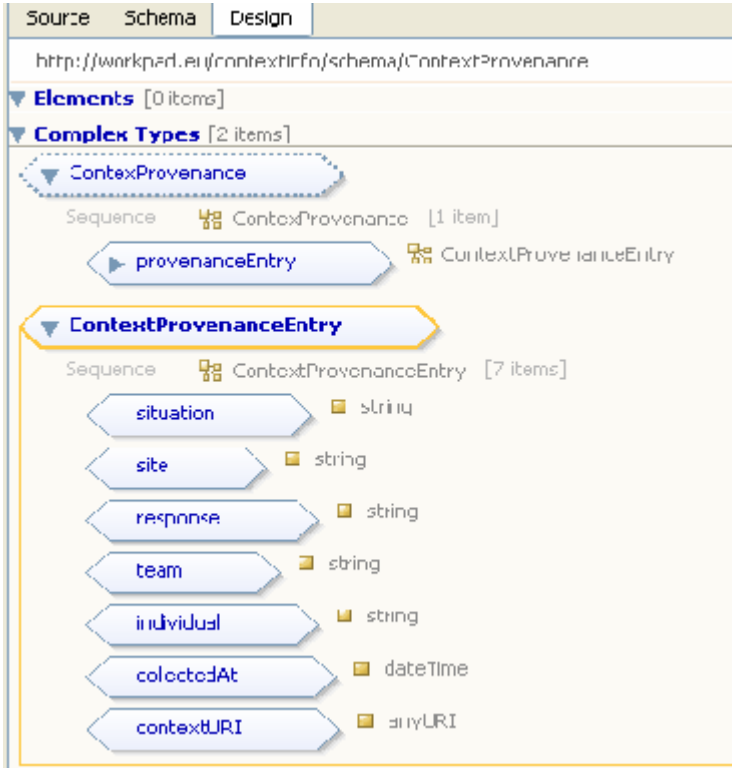
Management of Context Information



- Aggregration without knowing content of context information
  - Pull and push manner
  - XPath/XQuery-based filters
- Aggregation of snapshots of context information in $[t_i, t_j]$

Context Provenance Support

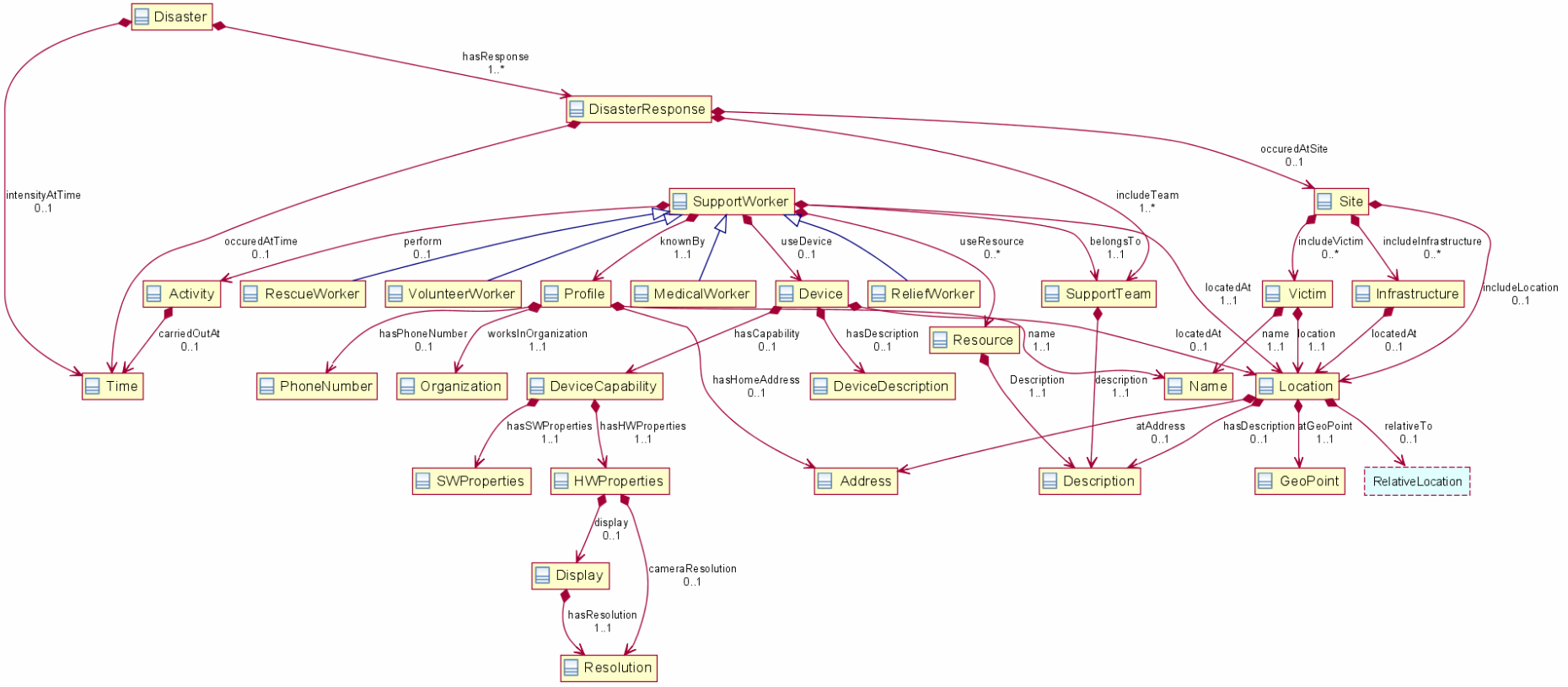
- For supporting tracing capability
- Types of context provenance information
  - Application-specific information
  - Middleware-specific information
- ESCAPE supports middleware-specific context provenance
- Storing provenance information into the back-end system



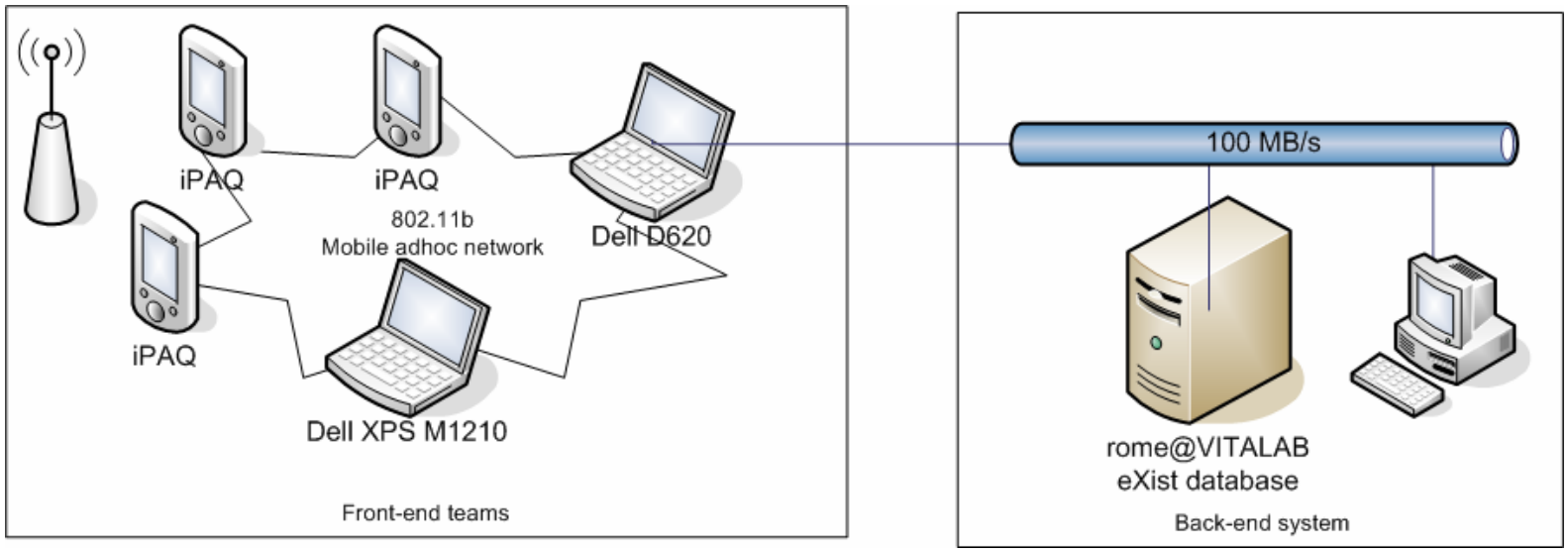
Implementation Status

- Web services
  - kSOAP for handling SOAP-based Web services, between CIMSs
  - REST model between CIMS and SCIMS
- XPath/XQuery/XUpdate for querying and updating information
- Hosting environments:
  - At the front-end: normal laptops and PDAs
  - At the back-end: SMP machines

Experiments: WORKPAD context information model



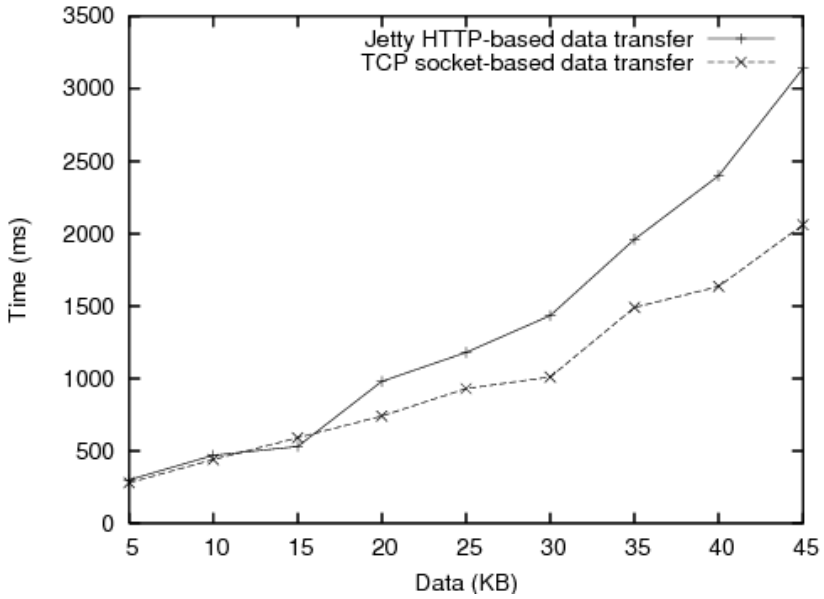
Experiments – Performance Tests



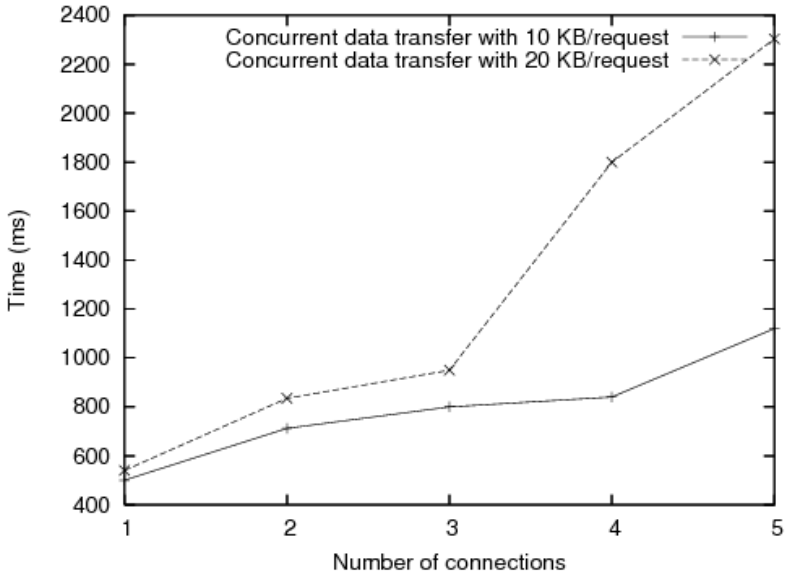
- Devices
  - 3 iPAQ 6915 PDAs (Intel PXA 270 416 Mhz, 64 RAM, Windows CE 5.0, 2GB external miniSD, IBM J9
  - Dell XPS M1210 (Intel Centrino Duo Core 1.83 Ghz, 2GB RAM, Windows XP)
  - Dell D620 (Intel Core 2 Duo 2Ghz, 2GB RAM, Debian Linux
  - Dell Blade (2 Xeon 3.2 Ghz CPUs with hyperthreading, 2GB RAM, Ubuntu Linux)
- Network
  - Mobile adhoc: setting 220Kbits/s (150Kbits/s, observed)
  - 100 MB/s to back-end
- Two modes
  - Single team and multiple team leaders

Experiments: Performance Tests

- Single data transfer between a CIMS member (PDA) and CIMS team leader (Laptop)

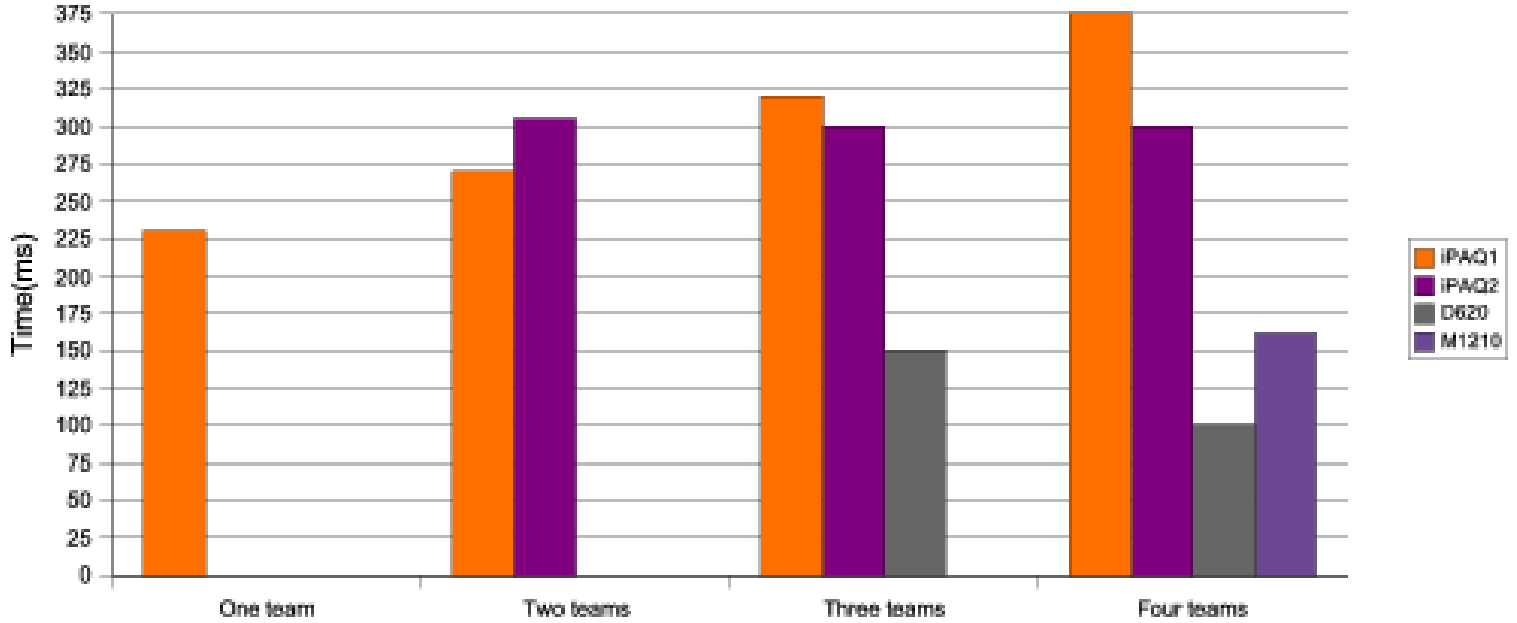


- Concurrent data transfers between a CIMS (PDA) and its client (Laptop)



- PDA is not suitable for team leader‘s device due to performance reasons

Experiments: Performance Tests



- Performance of transferring data to the back-end system
  - ~17KB, every 5 seconds
  - High variation between different runs in PDAs
    - 190 ms – 5170 ms
  - Should not update an XML document in the back-end with small documents
    - Store small XML documents and use back-end high capability to process information

Current applications

- Enrich GIS management tools with context information
  - Combine static and heavyweight GIS data (back-end) with volatile and in-situ context information
  - Emergency management tools show GIS information annotated with context information describing current status of entities within the emergency situation
- Provide context information for adaptive process management
  - Context information is used in adapting tasks performed in disaster responses

→ improving coordination and decision making of response activities based on real-time context information

Example: search for relevant context information

- Context information and GIS: *Team A wants to reach to a place „P“. Let‘s check unusable roads leading to „P“*

```
for $infrastructure in
collection('/db/contextinformation')//includeInfrastructure
where $infrastructure/category="ROAD" and
$infrastructure/condition="UNUSABLE"
return $infrastructure
```

- Context information and adaptive processes: *Let‘s send one worker to place „P“ to take a photo*

```
for $worker in collection('db/contextinformation')//SupportWorker
where $worker//hasCamera and $worker/belongsTo/description="Team 1"
and $worker//Activity/status="LOW"
return $worker
```

Conclusion and Future Work

- ESCAPE (Emergency Situations, Context Awareness, Pervasive Environments)
  - Supports extensible context information model and provenance
  - Makes context information available to front-end and back-end teams
  - Provides a basis framework for further developments of smart sensing and context in crisis situations
- Future work
  - Applications integration: GIS-based and in situ context information support and adaptive processes in disaster responses
  - Quality aspects of context information
  - Data aggregation based on rules and XML transformation techniques
  - Automatically process context information using Event-Condition-Action and Complex Event Processing techniques

Thanks for your attention!

Hong-Linh Truong

Distributed Systems Group
Vienna University of Technology

http://www.vitalab.tuwien.ac.at/autocompwiki