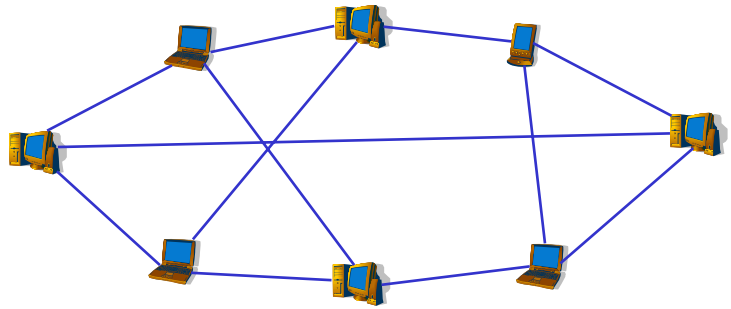
QuaP2P
Improving the Quality of P2P Systems (QuaP2P)

www.quap2p.de

Prof. Alejandro Buchmann, Ph.D.
Prof. Dr. rer. nat. Claudia Eckert
Dr.-Ing. Oliver Heckmann
Prof. Dr. Jussi Kangasharju
Prof. Dr. rer. nat. Max Mühlhäuser
Prof. Dr. rer. nat. Andreas Schürr
Prof. Dr.-Ing. Ralf Steinmetz

Technische Universität Darmstadt

Motivation



Peer-to-Peer (P2P)
- Internet traffic
  - ~80% of total Internet traffic

The most popular applications
- File sharing (eMule, Kazaa, etc)
- Content distribution platform (BitTorrent, MS Avalanche)
- Telephony (Skype)

Research
- Since quite a long time represented on conferences
  - ACM SIGCOMM, Infocom, ..
- Numerous (relatively new) dedicated conferences
- Research platform PlanetLab

However: barrier for broader usage
- Inadequate i.e. unexplored quality attributes in many areas

Goals of the Research Group

Improving the quality of P2P-systems

- Identifying and defining quality attributes and metrics
- Dependencies and trade-offs between the attributes
  - Examine and (as far as possible) quantify them
- How superior/inferior are P2P-Systems to existing systems?
- Effect of applying P2P-paradigm to different application area
- Proof-of-Concept of the investigated mechanisms using two scenarios

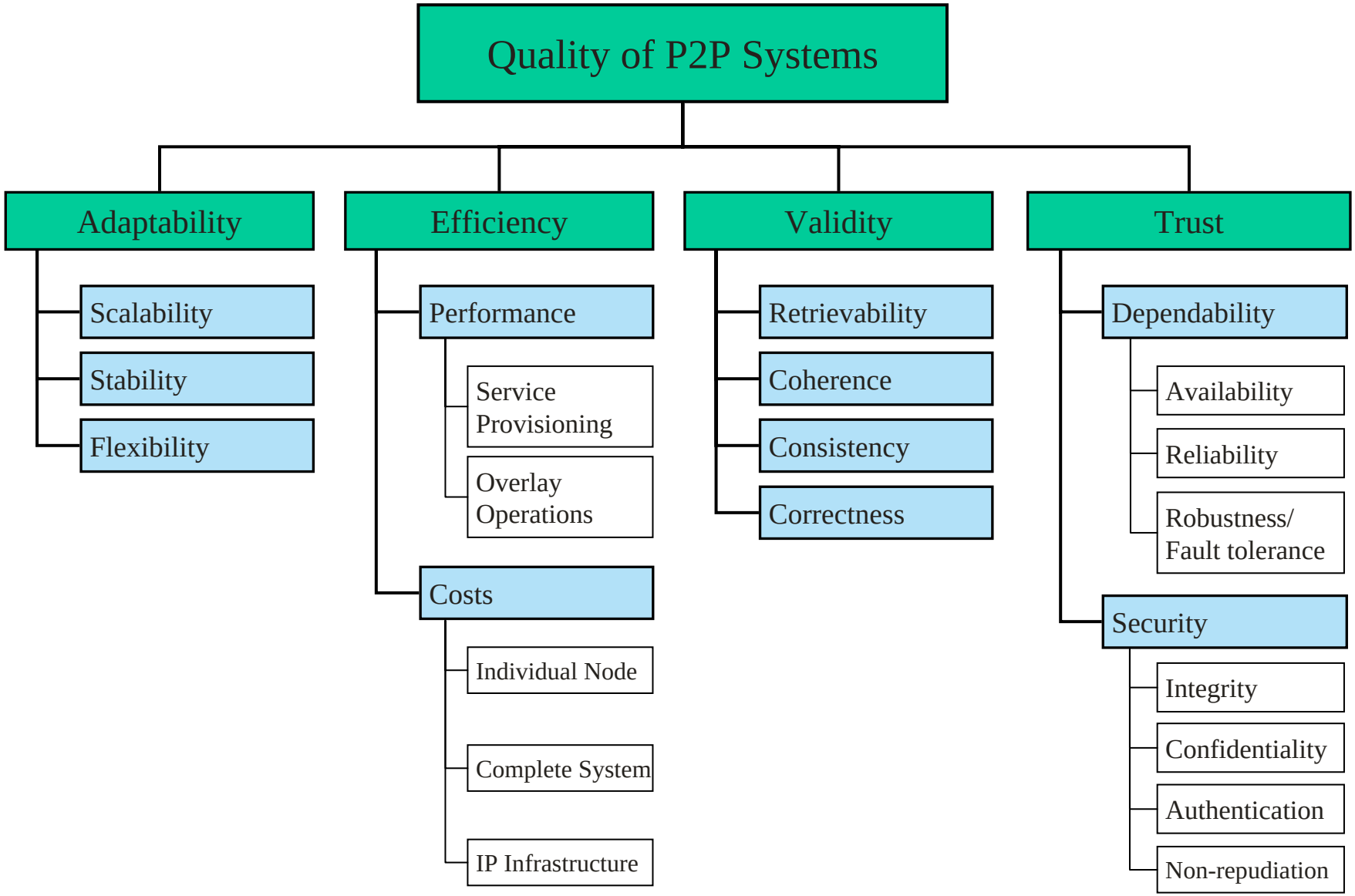
First phase

- Focus on bilateral interdependences between quality criteria
- Prototype implementation of scenarios

Second phase

- Examination of multilateral interdependences
- P2P Framework implementation

Quality Attributes



Mechanisms

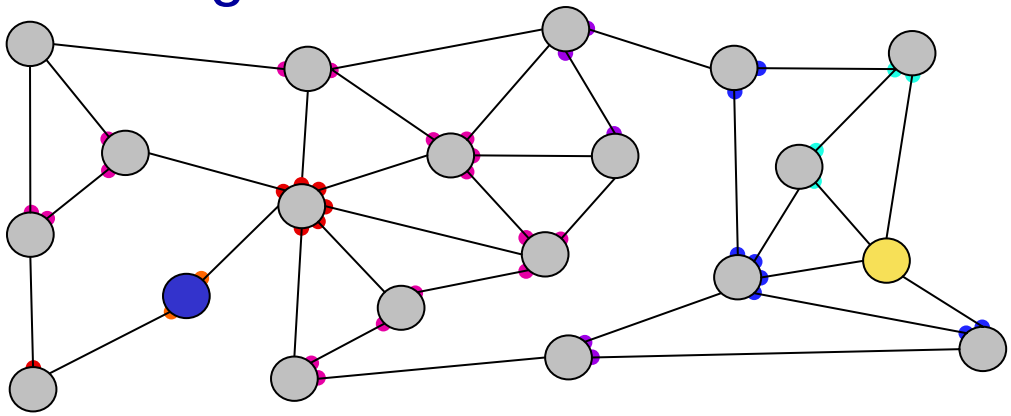
- A P2P-System consists of mechanisms for achieving the required functionality of the system
- New mechanisms or modifications of existing ones are necessary to improve the quality of P2P systems
- A mechanism is described by a number of (constructive) rules

Caution: There are **interdependences** and **goal conflicts** between the different quality attributes

Mechanisms: Example

2 different mechanisms for routing in unstructured overlays

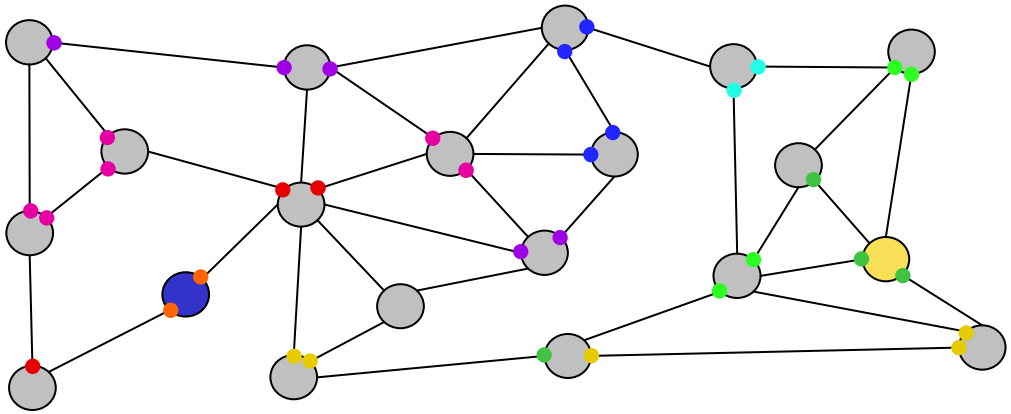
- Flooding



Overhead
- High
  e.g. 43 messages

Path-length
- 5 hops

- Random walks, n=2



Overhead
- Low
  e.g. 30 (36) messages

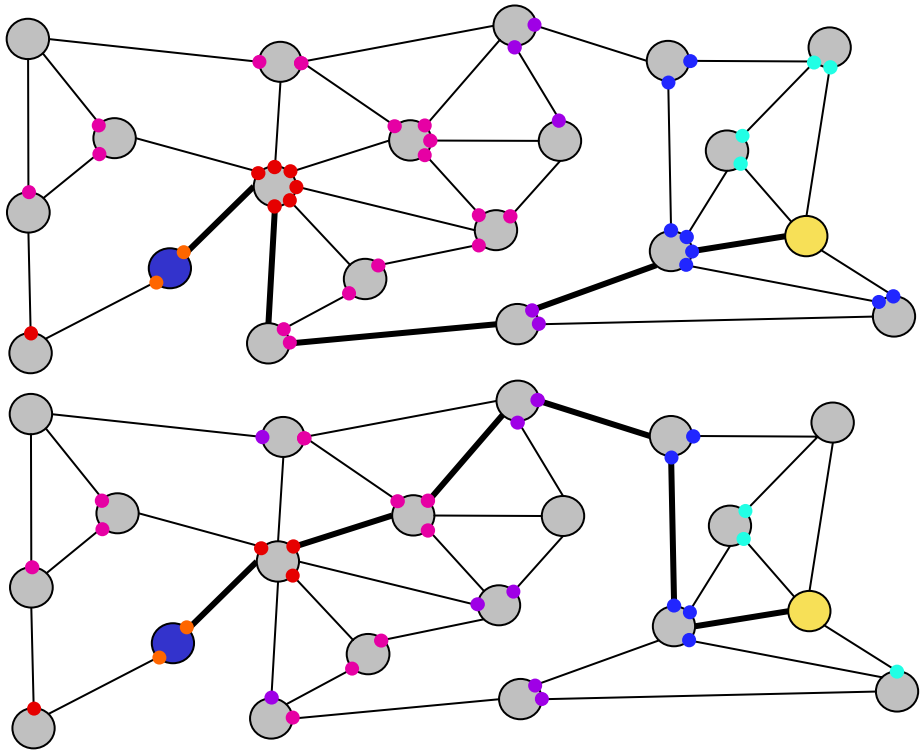
Path-length
- 6 (8) hops

Mechanisms: Example

Interdependency between efficiency (routing overhead) and retrievability (and dependability)

+ Efficiency in terms of less overhead

- Retrievability, dependability



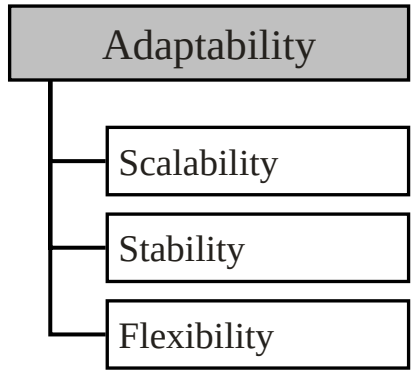
Flooding

- Here are e.g. 44 messages
- Path-length
  - 5 hops

Random Walks

- Here are e.g. 30 (36) messages
- Path-length
  - 6 (8) hops

Quality Attribute - Adaptability



Scalability
- quantitative adaptability of the system to a changing a number of
  - entities/nodes or
  - services in the system

Stability
- ability of the P2P system to maintain its functionality under changing basic conditions,
  - especially in the case of frequent adaptation procedures under “normal” operation
  - e.g. in contrast to content related robustness

Flexibility
- qualitative adaptability of P2P system
  - to changing circumstances in the “environment” of the system
  - compares to current research in *context aware computing*

Quality Attribute - Efficiency

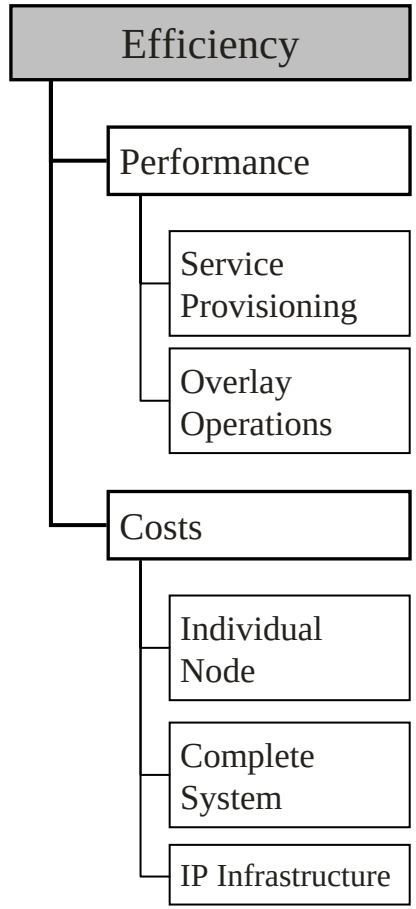
$$\text{Efficiency} = \frac{\text{Performance}}{\text{Costs}}$$

Performance

- performance of service provisioning
- performance of overlay operations

Costs´ point of view

- costs at an individual node
- costs of the whole P2P-Systems
- costs from the view of supporting IP infrastructure



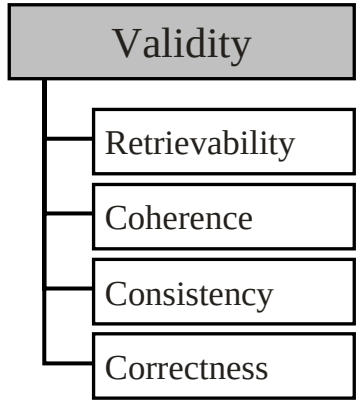
Quality Attribute Validity

Retrievability
- describes how well information/data stored in the P2P system can be found and retrieved by other peers

Coherence
- describes freshness/timeliness of the search result
- results set of the query includes the last (actual) version of the data

Consistency
- describes whether all replica of one piece of data in the system are equal

Correctness
- specifies whether objects stored in the P2P system fulfill integrity constraints according to their specification and semantic
  - internal correctness
    - internal structure of the single (possibly versioned) data object
  - external correctness
    - constraints and relations between data structure objects which can be versioned and changed separately to each other



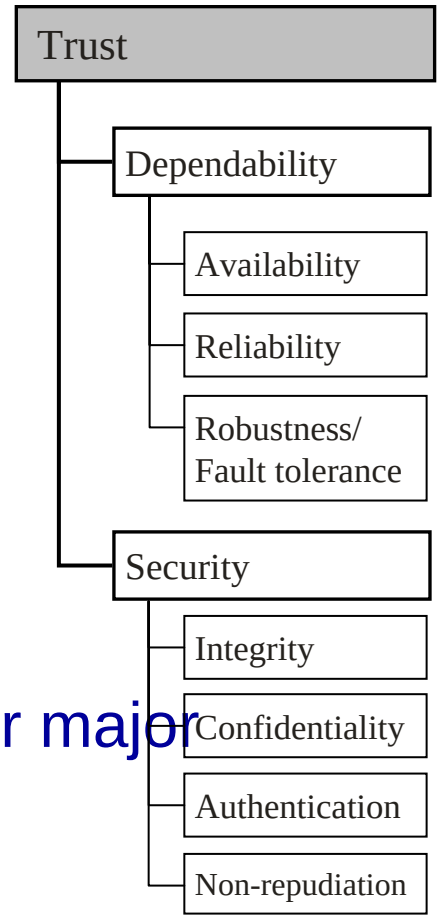
Quality Attribute – Trust (Dependability)

Dependability– Availability
- the system is ready to deliver the correct services at any time

Dependability– Reliability
- system continuously delivers services correctly as specified

Dependability– Robustness
- continue operation under or recover from severe system dynamics (like peak loads or major disruptions in network connectivity)
  - related to stability

Dependability– Fault tolerance
- Extension of robustness to occurence of error



Quality Attribute – Trust (Security)

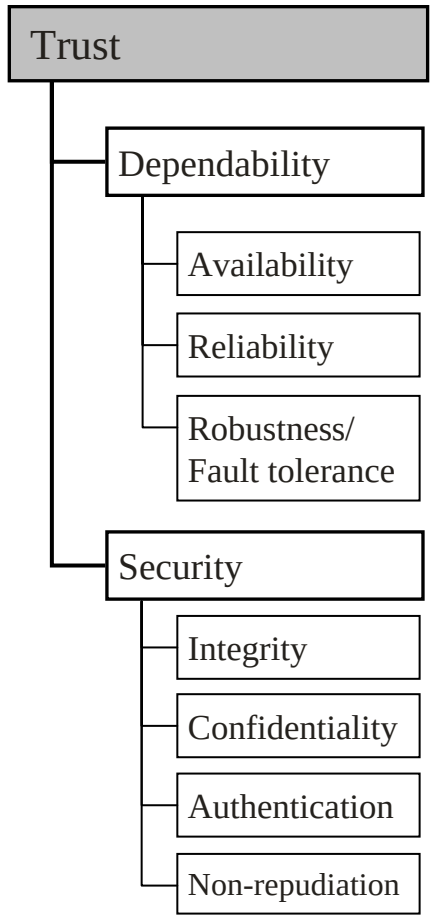
Security – (Data) integrity
- no unauthorized data modification

Security– Confidentiality
- no unauthorized gain of information

Security– Authentication
- ensuring that peers are who they claim to be

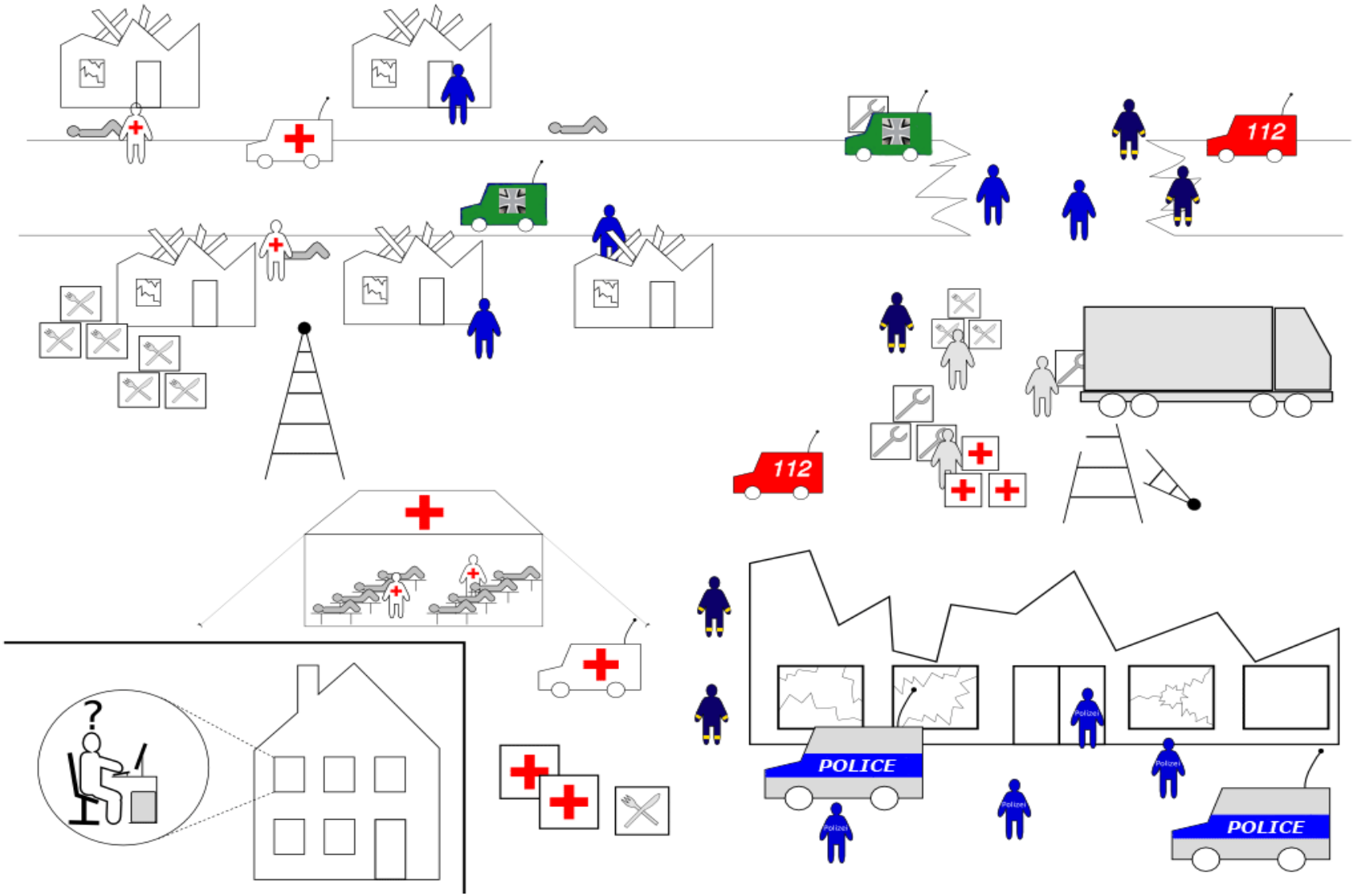
Security– Non-repudiation
- executed actions cannot later be denied by one of the peers involved

Maybe: security– Privacy, Anonymity
- personal identity of a user or peer remains unknown



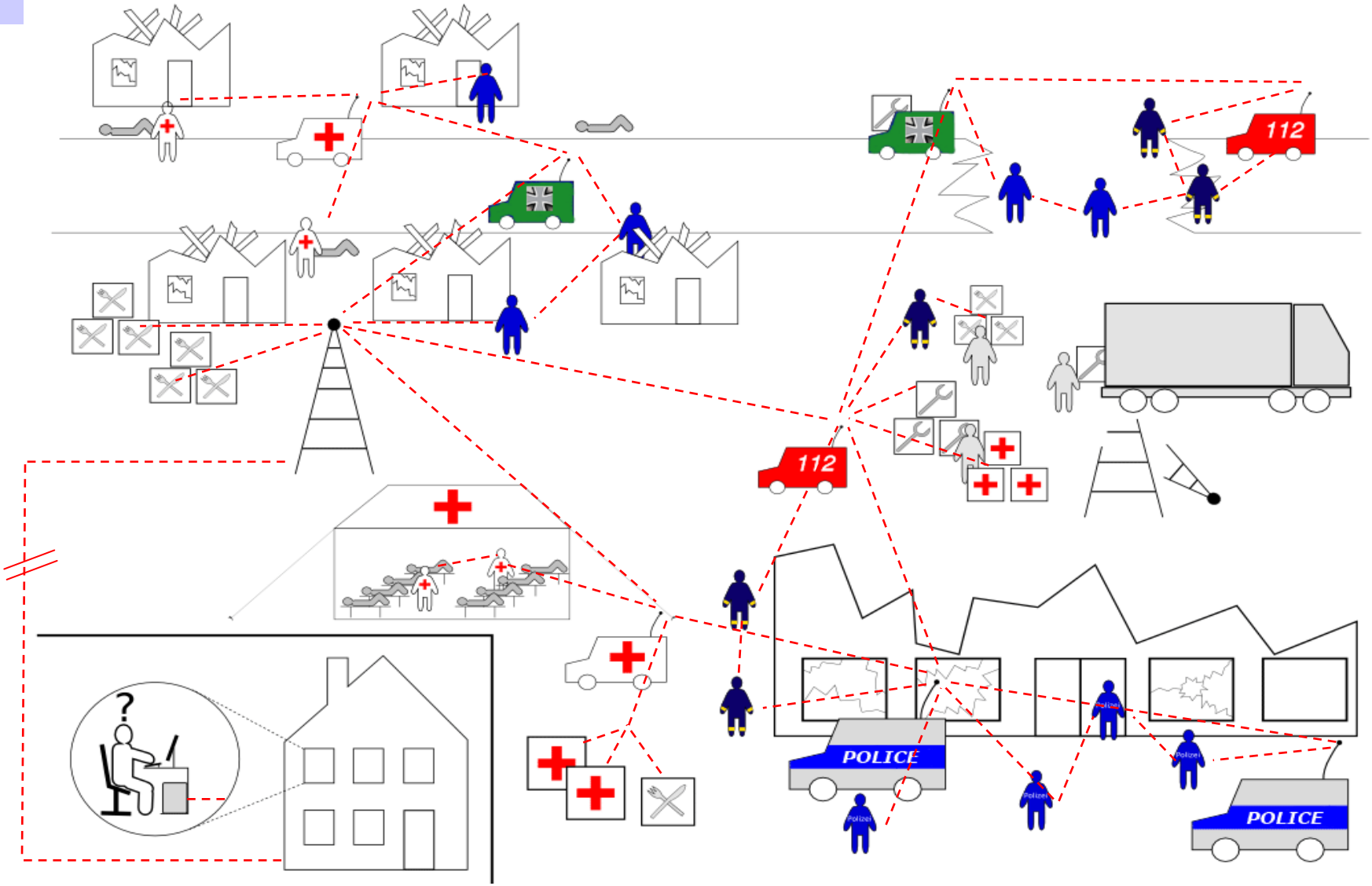
Scenario A

Communication System for Rescue Operation



Scenario A

Communication System for Rescue Operation



Scenario A

Communication System for Rescue Operation

Different types of traffic and service

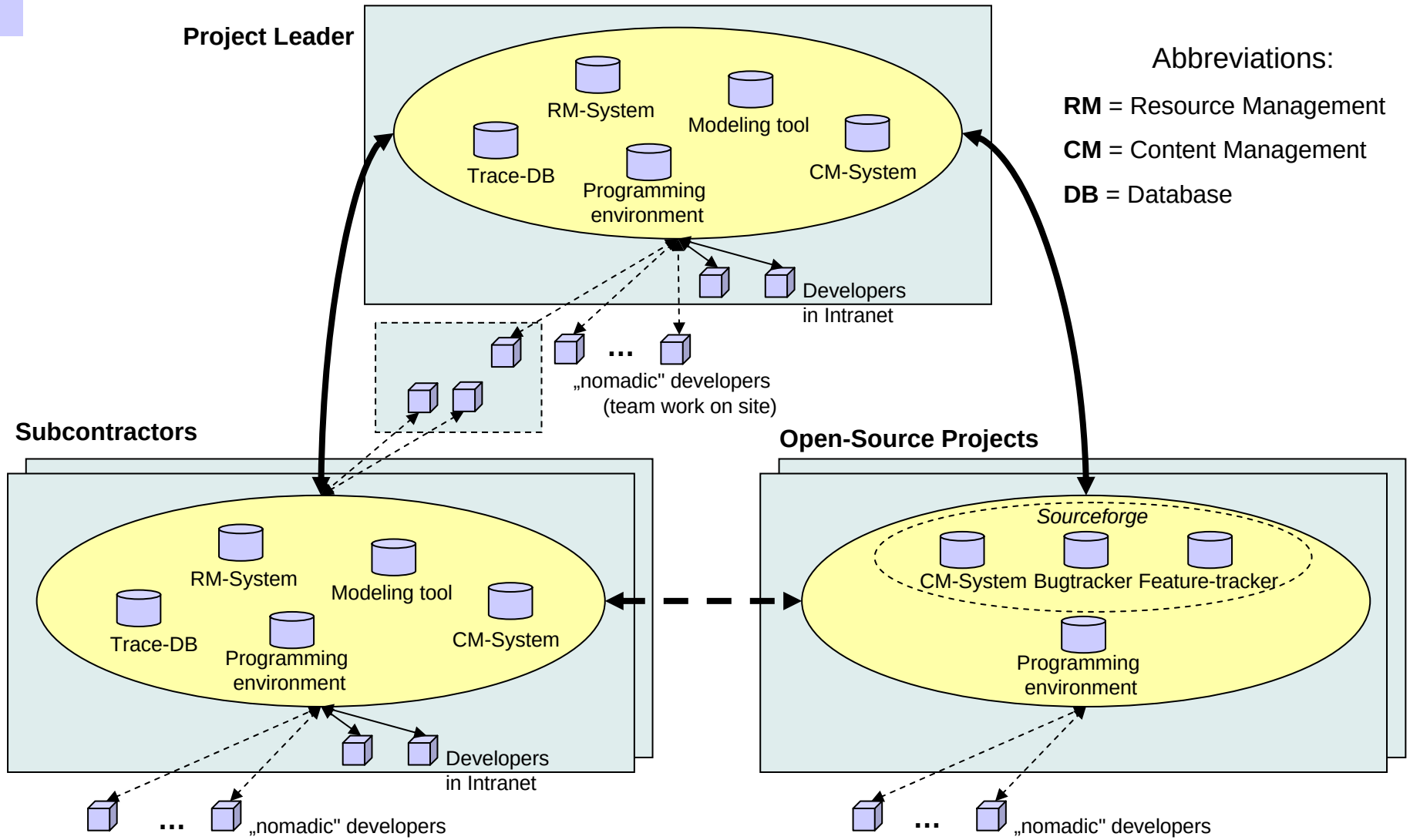
- Traffic: Audio- und video communication, live chat, data transfer, …
- Service: Synchronisation of databases, locating people, devices, …

Requirements

- Scalability & Stability: Highly dynamic, large and heterogeneous network
- Efficiency: Scarce resources
- Validity: Nodes can be disconnected any time
- Robustness: Can decide about survival

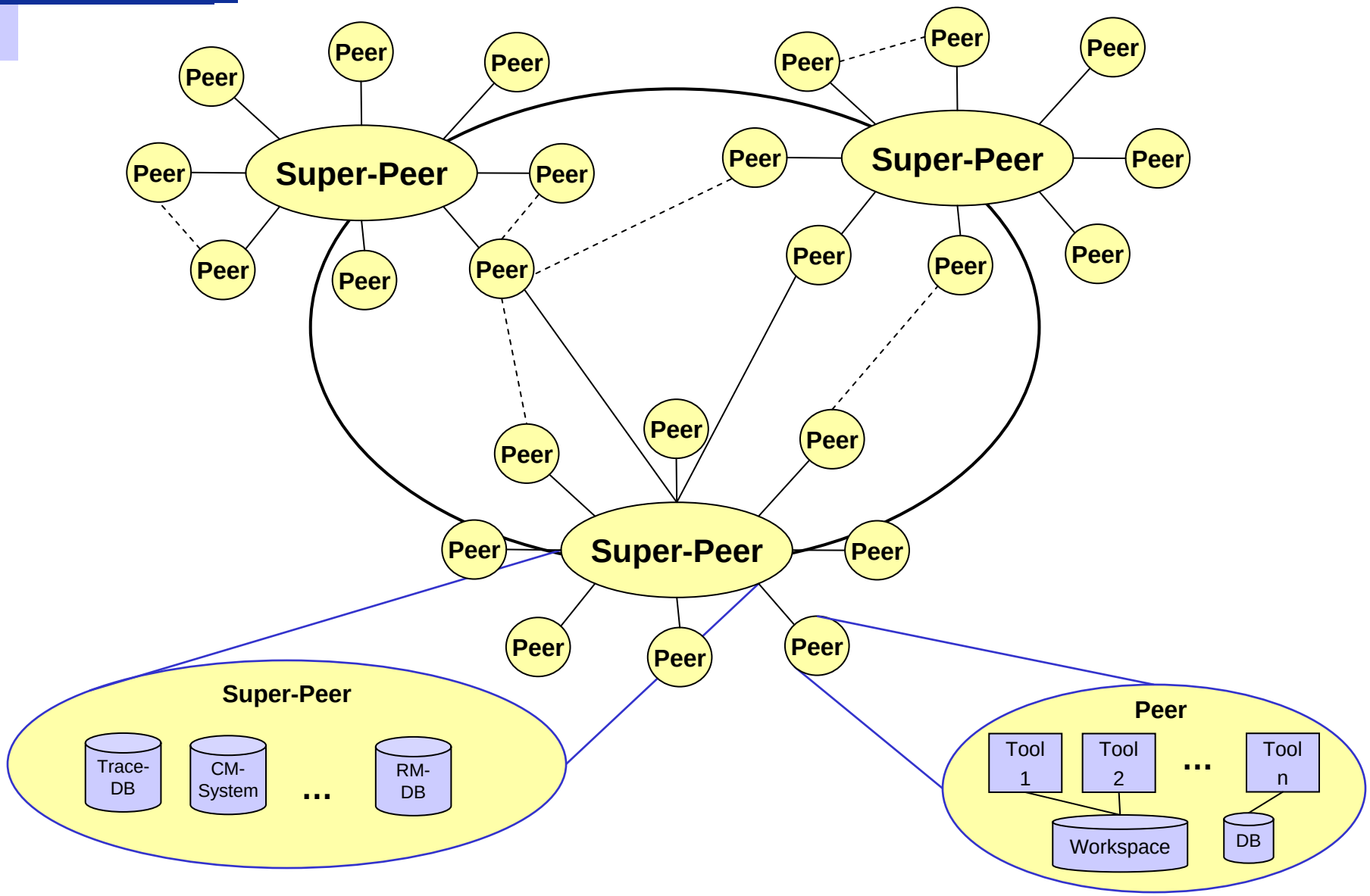
Scenario B

Globally Distributed Software Development



Scenario B

Integration of Development Tools with P2P



Characteristics of Scenario B

- Geographical distribution of complex structured development data with metadata (models, source-code, tables, …)
  - ➢ guarantee of retrievability, availability
- Concurrent write and read access to versioned development data
  - ➢ guarantee on coherence, consistence
- Multiple dependencies to development data (traceability links)
  - ➢ guarantee on correctness
- Developer has to be able to access all versions of (needed) development data
  - ➢ guarantee on reliability, robustness
- Development process and –organization create access rights for security critical data
  - ➢ guarantee on integrity, trust and authenticity …

From Client/Server to P2P-Approach

Software-development is traditionally on client/server architecture

- However: Communication is over border of an organization
- However: „Ad-hoc“ team work nomadic developer

First approach to P2P-support

- Computer Supported Cooperative Work (CSCW): Groove
- Process-/Workflow-Management: PeCo, SwinDeW
- Version-Management: Co-Op, Gram

The big challenge of the scenario

How far can P2P technology replace the “classic” Client-/Server-Technologies for coordinating software development processes? What are the chances for the development processes, how much loss of quality is unavoidable? What can be gained on the other hand?

Quality Requirements of the Scenarios

QuaP2P

Scenario A:
P2P-based communication system for rescue operations

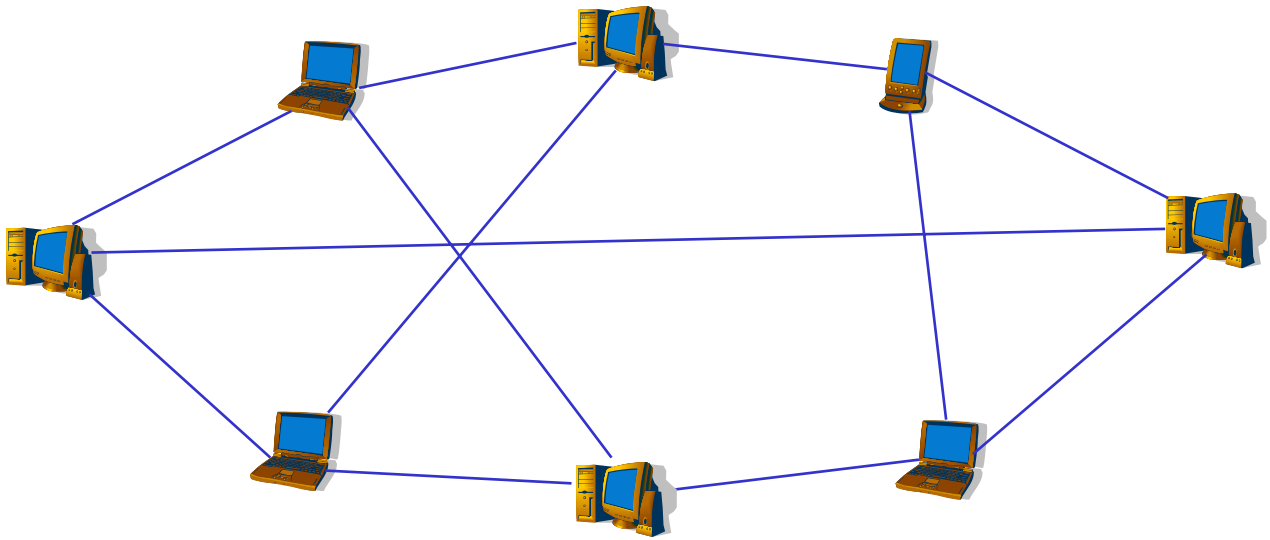
Scenario B:
Integration of Software development tools with P2P techniques

		Scenario A	Scenario B
Adaptability	Scalability	++	+
	Stability	++	++
	Flexibility	+++	+
Efficiency	Performance– Services Provisioning	+++	+
	Performance - Overlay Operations	++	(+)
	Costs - Individual Node	+++	(+)
	Costs - Complete System	+++	(+)
	Costs - IP Infrastructure	+++	+
Validity	Retreivability	+++	++
	Coherence	++	+++
	Consistency	++	+++
	Correctness	+	+++
Trust	Dependability– Availability	++	+++
	Dependability– Reliability	+	+++
	Dependability– Robustness/Fault tolerance	+++	++
	Security – Integrity	++	+++
	Security– Confidentiality	+	+++
	Security– Authentication	+	+++
	Security– Non-repudiation	+	++

Example Efficieny Workpackage: Optimization of Resource Allocation

- Load balancing and incentive systems
  - support the heterogeneity of the peers
  - transparency and minimum fairness

- Knowledge about system state is necessary
  - data location: decentralised
  - type of data collection: decentralised
  - therefore decentralised resource accounting necessary



Example Efficieny Workpackage: Optimization of Resource Allocation

- Load balancing and incentive systems
  - support the heterogeneity of the peers
  - transparency and minimum fairness
- Knowledge about system state is necessary
  - data location: decentralized
  - type of data collection: decentralized
  - therefore decentralised resource accounting necessary
- Strong influence on trust
  - e.g. fraud possibility
  - positioning of replicas
- First results: AMCIS 2005, effects of unequal usage/supplying of resources
  - influence on stability as well

Summary and Conclusions

DFG Forschergruppe 733 (QuaP2P)

- Insufficient / unpredictable quality is one obstacle for broader use of P2P technology in „serious“ applications
- Goal: Improving the quality of P2P systems
- Different quality attributes and their mutual interdepencies are investigated
- Improved mechanisms for P2P systems are developed
- Proof-of-concept with 2 prototypical scenarios
  - P2P communication system for rescue operations
  - Integration of Software development tools with P2P techniques

More information @ www.quap2p.de