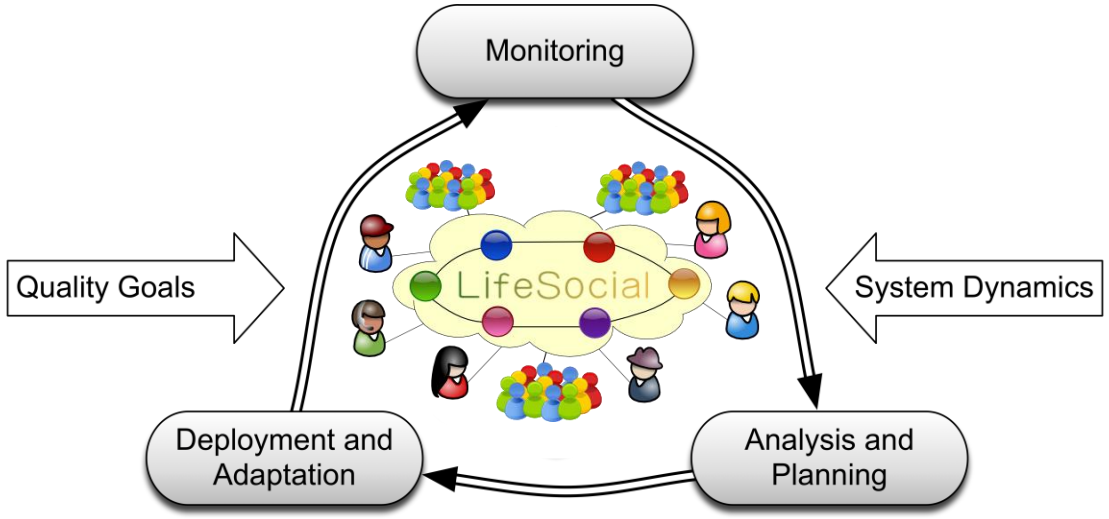
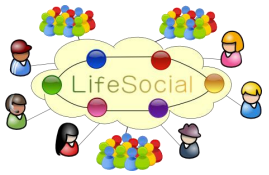
Monitoring and Managing Quality in P2P-based Online Social Networks

TECHNISCHE UNIVERSITÄT DARMSTADT





LifeSocial - www.lifesocial.org

QuaP2P

QuaP2P
Improving the Quality of P2P Systems
DFG research group 733

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Fachgruppe Theorie verteilter Systeme
Prof. Dr. Christian Scheideler
UPB – Universität Paderborn

Dr.-Ing. **Kálmán Graffi**

Graffi@KOM.TU-Darmstadt.de
Graffi@Uni-Paderborn.de

KOM - Multimedia Communications Lab
Prof. Dr.-Ing. Ralf Steinmetz (Director)
TUD – Technische Universität Darmstadt

27. April 2026

Communication Paradigms in the Internet

TECHNISCHE UNIVERSITÄT DARMSTADT

Internet application trends

- Large-scale applications
- User-generated content
- User-to-user communication

Example

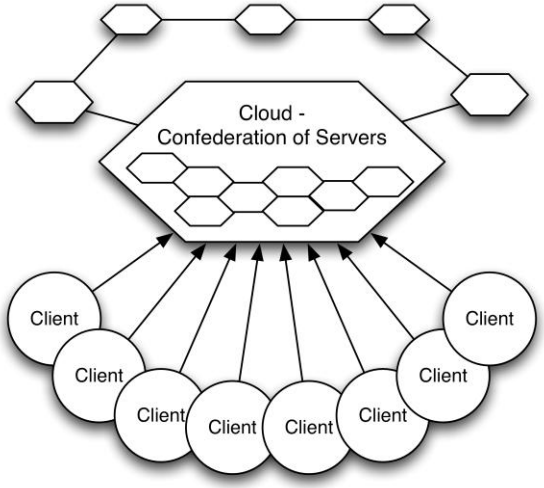
- Online social networks
- Access to user profiles and pictures

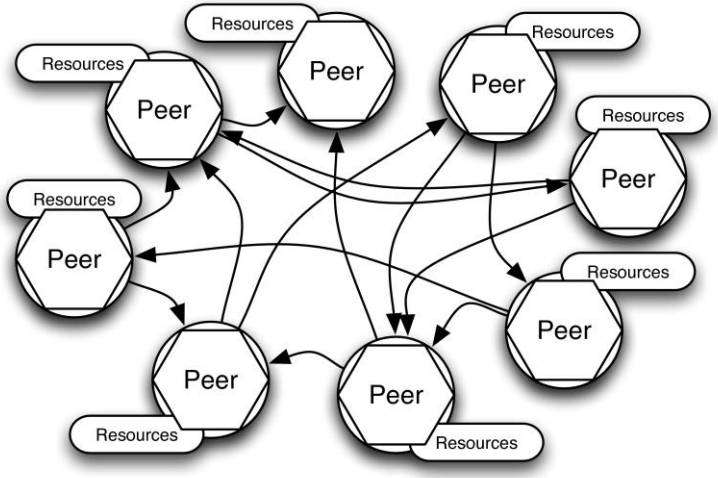
Client / server paradigm

- Server cluster as provider
- Good: controlled quality
- Bad: costs

Peer-to-peer (p2p) paradigm

- Infrastructure hosted by peers
- Good: shared costs
- Bad: quality?





Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Typical motivation for p2p applications

- Client / Server is bad → P2P is needed

My motivation: P2P is dying

- BitTorrent, edonkey, …, file sharing → One click hosting
- Skype → technological issues, although simple functionality
- Wuala, data storage → very centralized
- Joost (p2p tv), Groove (groupware) → Not anymore P2P

The Cloud is killing the purpose of P2P on user devices

- Quality and costs are guaranteed

To survive, P2P needs:

- Controlled quality and new applications

Challenges for Quality of P2P Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

Distributed character

- Distributed solutions and overlays needed

Undefined scale

- From tens to several millions

Peer fluctuation (churn)

- Peers join / leave the system autonomously

Peer heterogeneity

- Varying capacities and connectivity

Static configurations are insufficient

- Requirements known when system is deployed
- Updates are difficult

Approaches to Address Quality Challenges

TECHNISCHE UNIVERSITÄT DARMSTADT

Building a quality-aware p2p application

- Identify: application scope, scenario, expected dynamics
- Build software
  1. relying on peer capacities
  2. to meet (quality) expectations

1. Simple access to peer capacities in network

- Reliable usage of the capacities
- Support for mechanism design

2. Observe and control system quality

- Maintain high service quality
- Interest of users and application provider

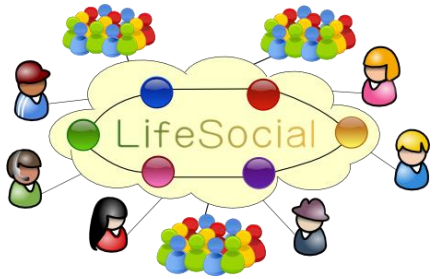
Agenda of the talk

TECHNISCHE UNIVERSITÄT DARMSTADT

Quality in p2p systems: *General mechanisms for* ***monitoring*** *the system and peer information in p2p systems and* ***managing*** *the quality of the p2p system and capacity reservations*

My contributions

- **SkyEye.KOM**: Monitoring
  - Statistics on system behavior
  - Capacity-based peer search
- **SkyNet.KOM**:
  Autonomic management cycle
- **P^{3}R^{3}O.KOM**:
  Reliable capacity reservation
- **LifeSocial.KOM**:
  Platform for online social networks



Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction

Overview on my System Management Approach

- System Monitoring – SkyEye
- System Management – SkyNet

P2P-based Online Social Networks

- LifeSocial – Secure, Distributed, Controlled

Conclusions

System Management Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Goal

- System automatically adapts to meet predefined target metric intervals

Traditional approach

- Build mechanism / overlay for
  - Specific scenario
  - Expected dynamics
- Limited reusabilty

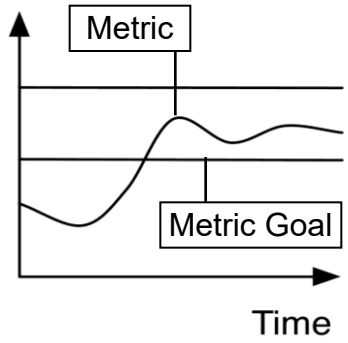
Approach

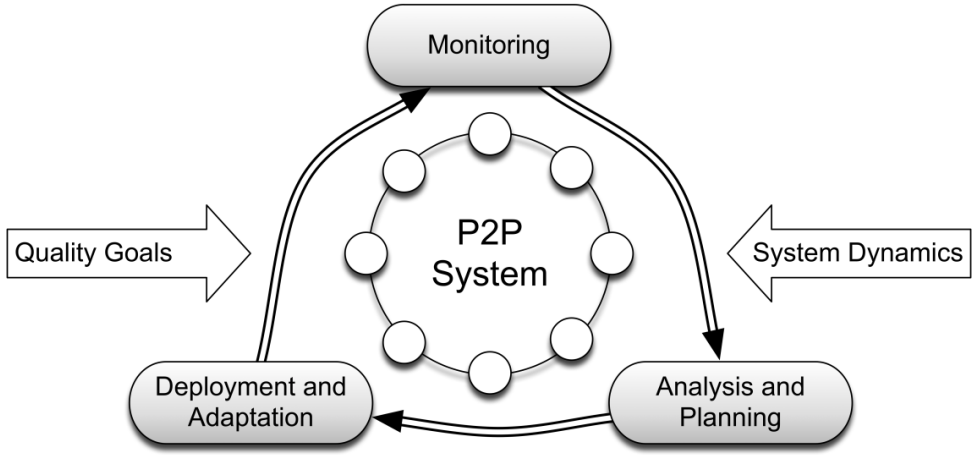
- Research on viable concepts

→ **SkyNet.KOM**:
Automated self-configuration framework

Challenges

- Precise monitoring of system behavior
- Quick information transport
- Closing the management cycle



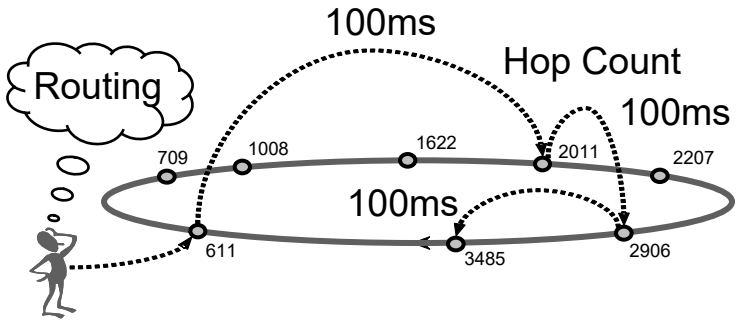


Running Example: Improving the Lookup Delay

TECHNISCHE UNIVERSITÄT DARMSTADT

P2P overlay functions

- Overlay provides ID-based routing
- Peers are responsible for ID ranges
- Used for
  - ID-based document storage
  - ID-based role assignment

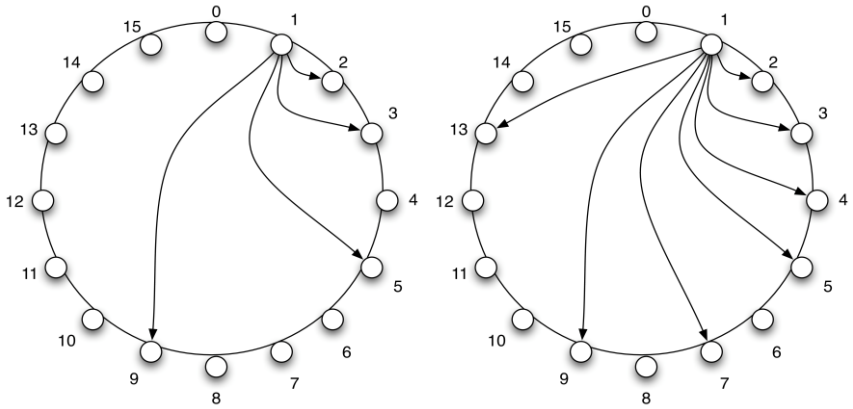


Classic p2p overlay “Chord”

- Low routing delay desired
- Routing delay influenced by hop count

→ Target metric for hop count: [7, 10]

- Influencing parameter: finger table size



→ Approach for adapting hop count to [7,10]

- Monitor metric status
- Analyze derivation from target metric
- Re-configure system if needed

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction

Overview on my System Management Approach

- System Monitoring – SkyEye
- System Management – SkyNet

P2P-based Online Social Networks

- LifeSocial – Secure, Distributed, Controlled

Conclusions

SkyEye.KOM: Tree Topology

TECHNISCHE UNIVERSITÄT DARMSTADT

Concept of new layer

- Decouples from specific p2p overlay
- Unified ID space [0,1]

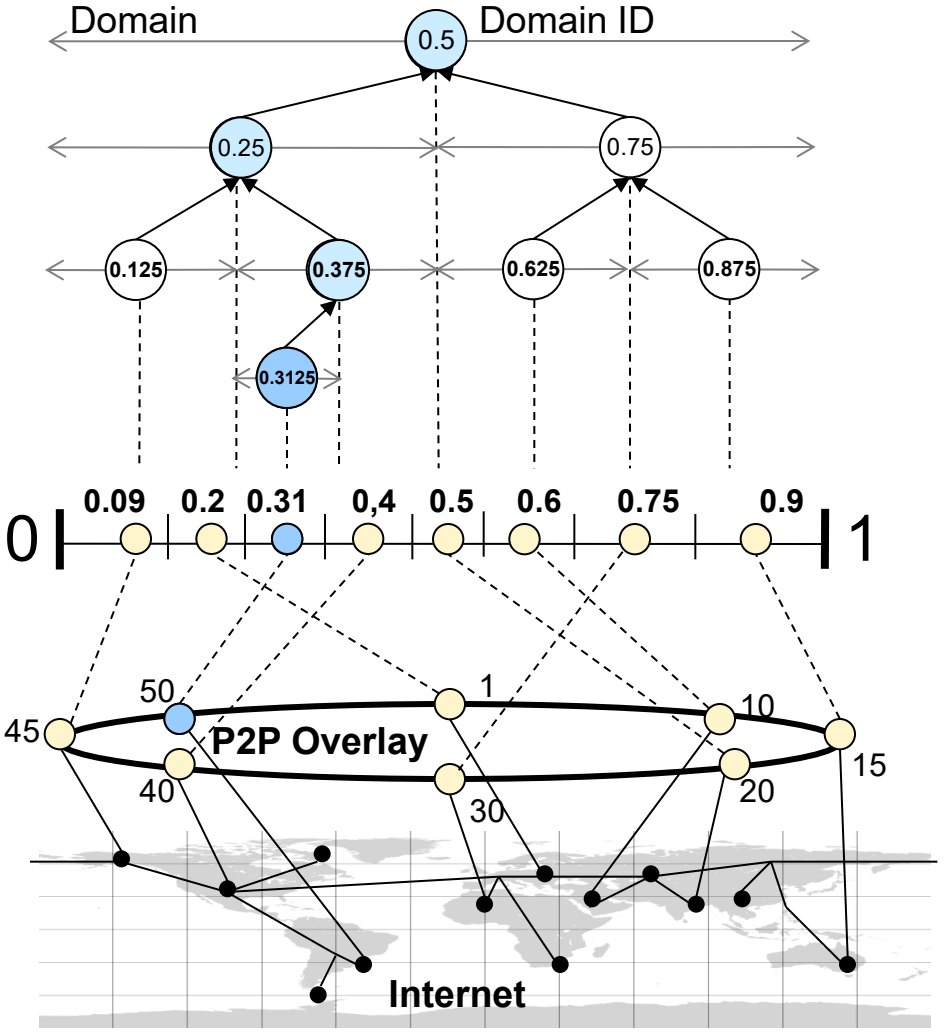
Tree of information domains

- Domain: ID range for monitoring
- Domain size split in **β** parts per level
- Domain IDs build tree topology

Peers to Domain ID assignment

- Peer ID_p, level l → Domain IDs
- If peer is responsible: position defined

$$K_p^l := \frac{\frac{\lfloor ID_p \cdot \beta^l \rfloor + 1}{\beta^l} + \frac{\lfloor ID_p \cdot \beta^l \rfloor}{\beta^l}}{2}$$



SkyEye.KOM: Communication Protocol

TECHNISCHE UNIVERSITÄT DARMSTADT

Gathering global view

- Information type: Lookup time, tree and network structure, overhead (types), …
- Every peer measures local status
- Periodically sent to parent peer
  - → Update Interval (**UI**)

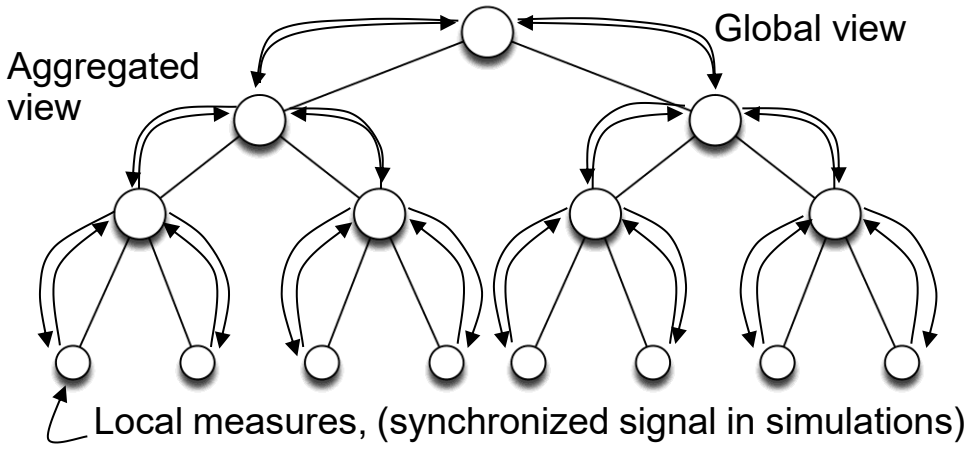
Aggregation

- Direct: count, sum, minimum, maximum, sum of squares
- Derived: mean, variance, std. deviation

Dissemination of global view

- Global view in root
- Every update message is acknowledged
- Contains global view from level above

Metric update protocol:



Gather and disseminate time

$$O(\log_{\beta} (N) \cdot UI)$$

Evaluation of SkyEye.KOM

TECHNISCHE UNIVERSITÄT DARMSTADT

Evaluation goal

- Understanding SkyEye.KOM's behavior
- Measuring performance and costs

Evaluation methodology

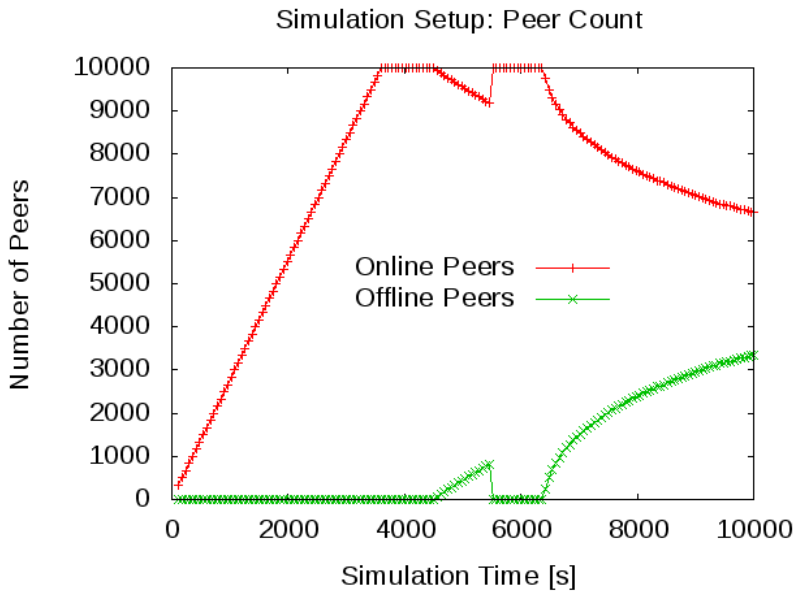
- Analytical model:
  - Parameter study
  - Tree characteristics
- Simulation:
  - Parameter study
  - Detailed single run
- Testbed evaluation:
  - Churn behavior
  - Accurate costs

Simulator – PeerfactSim.KOM

- Delay: global network positioning model
- Churn: based on KAD, exponential churn

Simulation Setup

- N=10,000, exponential and KAD churn



Simulation Run – N=10000, UI=60s, β=4

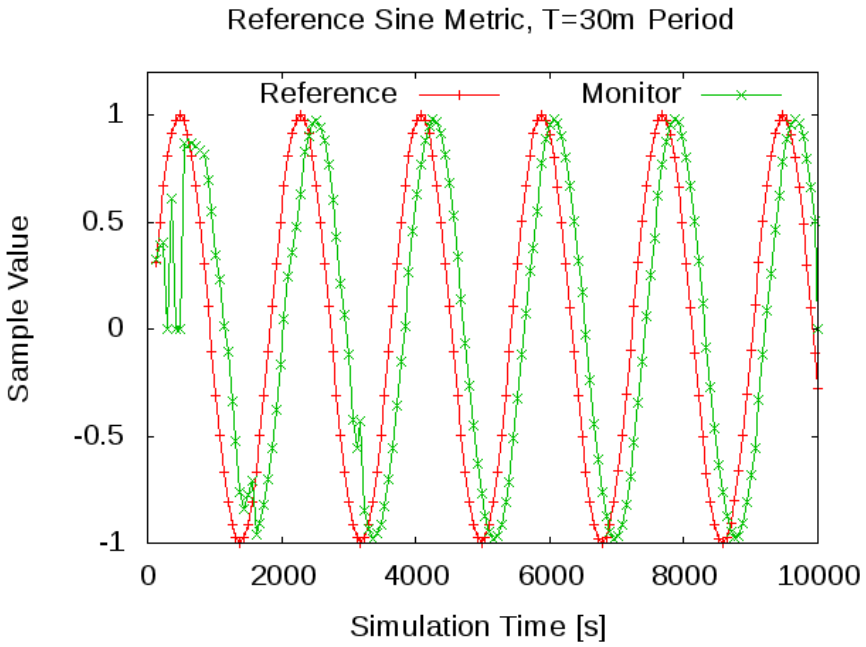
TECHNISCHE UNIVERSITÄT DARMSTADT

Precision: synchronized signal

- Sine, periodicity: 1800s
- Shows monitoring error and freshness

Results

- Precise measurement
- Age of information: ca. 200s

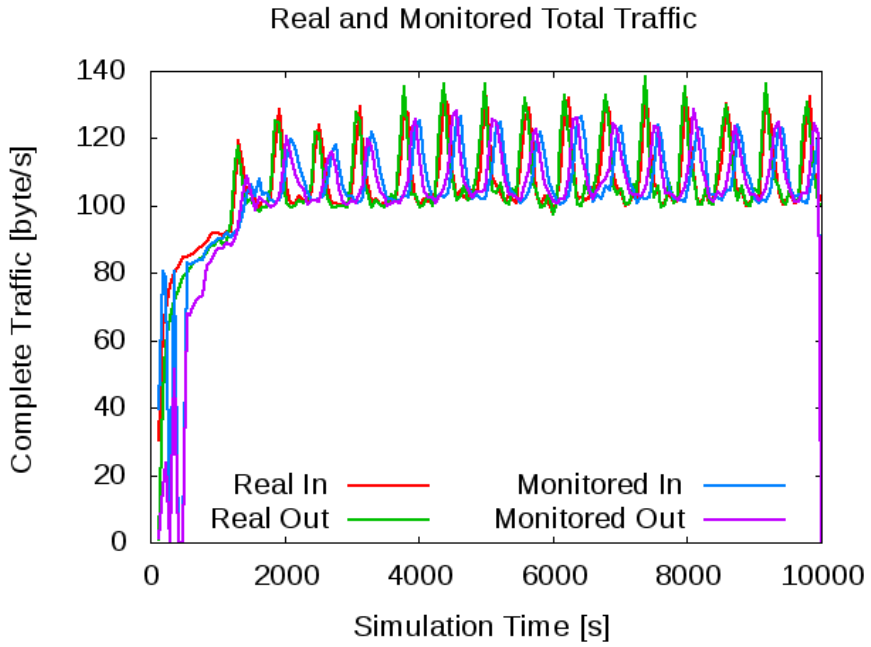


Costs: traffic overhead

- Shows average load on peers
- Load is well balanced

Results

- Traffic overhead: 100–140 bytes/s
- Calculation of average is churn-resistant



Testbed Evaluation: Validation of Results

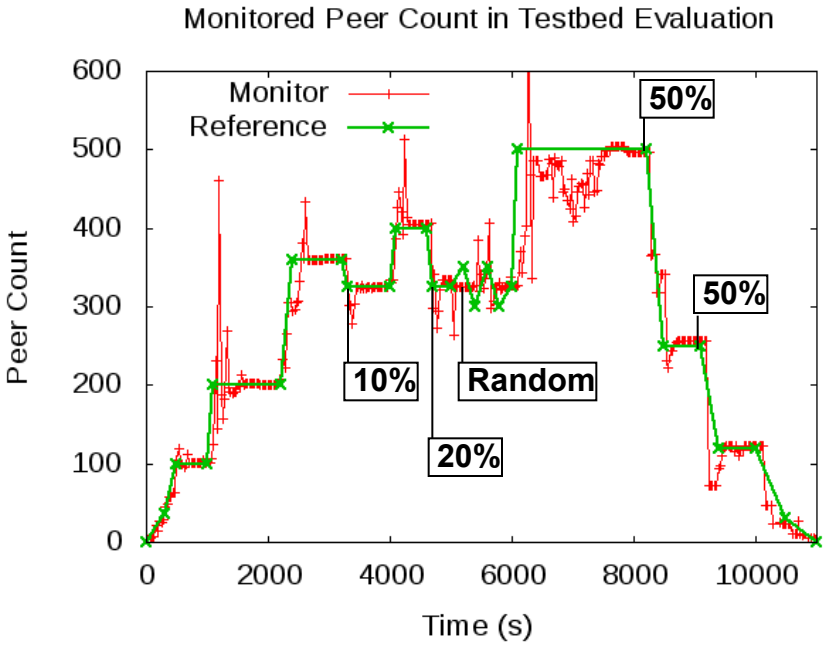
TECHNISCHE UNIVERSITÄT DARMSTADT

Testbed evaluation

- Testing behavior under strong churn
- Metrics: peer count and traffic overhead

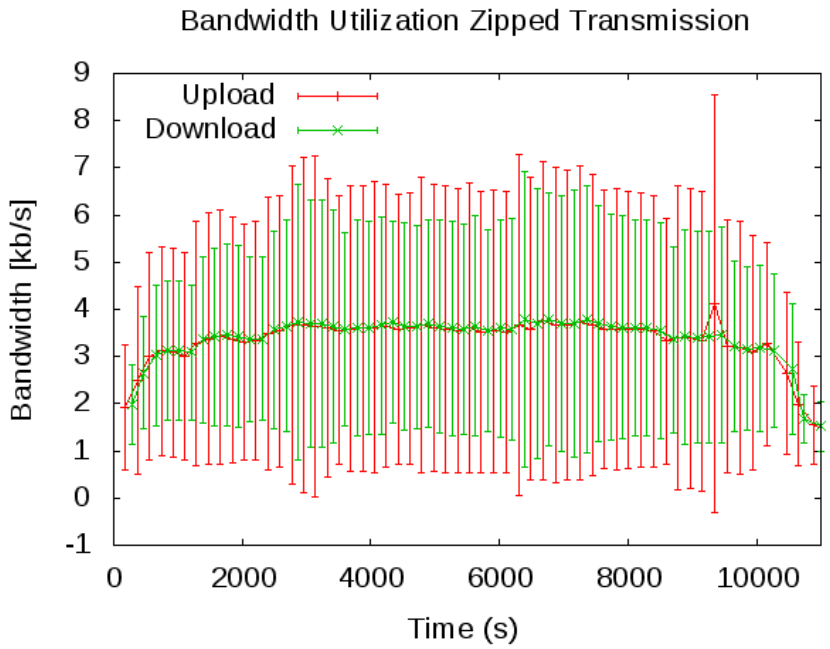
Setup:

- **N**=500 (on 37 PCs), **UI**=5 sec, **β** = 2
- 10%, 20%, 50% and random churn



Results

- Previous good results validated
- Even 50% peer fail is not crucial
- Short outliers due to joining peers
- 3kb/s bandwidth consumption (UI=5s)
- Average calculation not affected by churn



Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction

Overview on my System Management Approach

- System Monitoring – SkyEye
- System Management – SkyNet

P2P-based Online Social Networks

- LifeSocial – Secure, Distributed, Controlled

Conclusions

Control Theory in P2P Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

Service quality of the p2p system

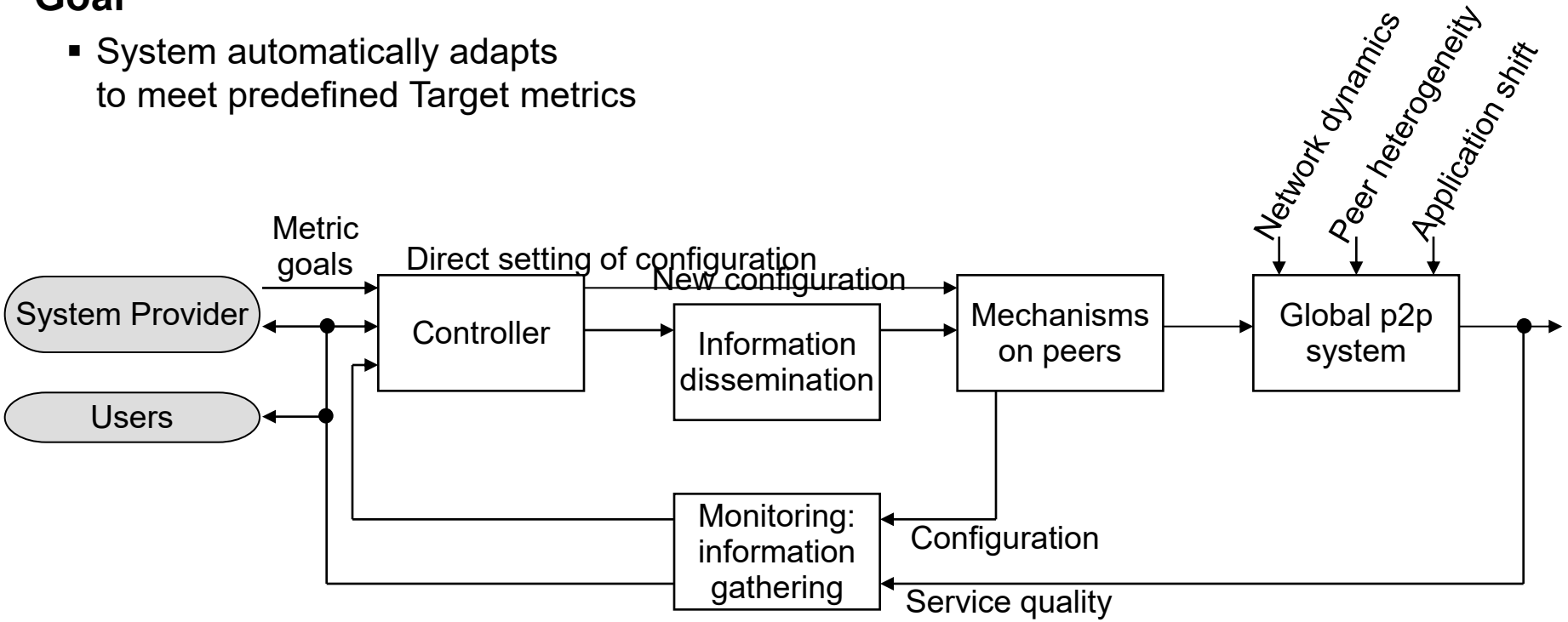
- Application-specific metric intervals
- *Response time [100ms,400ms]*

Goal

- System automatically adapts to meet predefined Target metrics

Main idea

- Research on viable approaches
- Autonomic computing for p2p systems

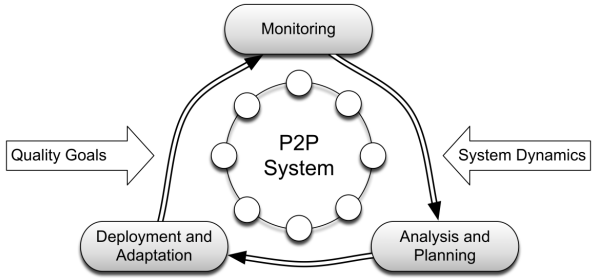


Planning a new Configuration

TECHNISCHE UNIVERSITÄT DARMSTADT

Analysis and planning step

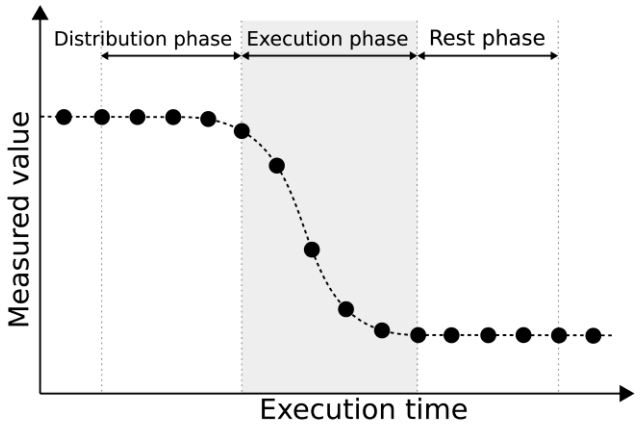
- Detects metric violation
- Step-wise adaptation of configuration
- Approaching target metric interval

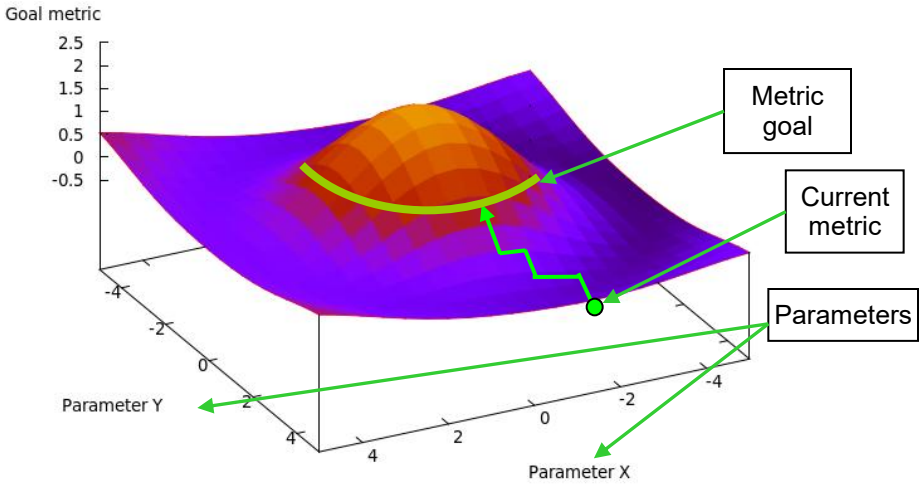


Changes need time

- Analyze slope of metric history
- Act only if changes settled
- Prevent configuration oscillation





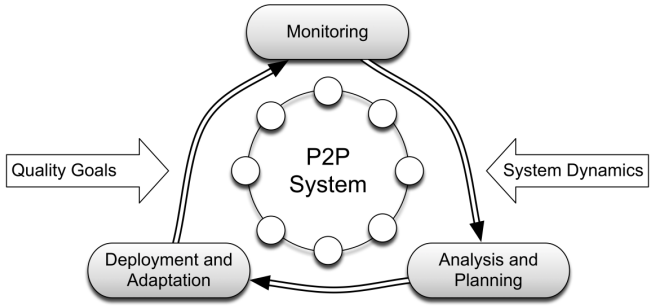


Re-Configuration of the System

TECHNISCHE UNIVERSITÄT DARMSTADT

Management protocol

- Gather current system parameters
- Distribute new parameter configuration

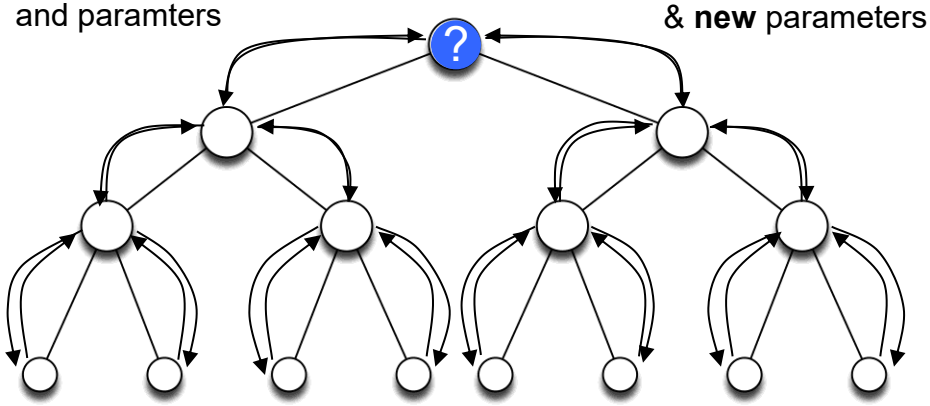


Approach

- Extension of SkyEye.KOM
  - Gather statistics on parameter settings
- Root:
  detects violation, plans new configuration
- New configuration:
  - Spread with SkyEye.KOM in ACKs
  - Applied by peers locally

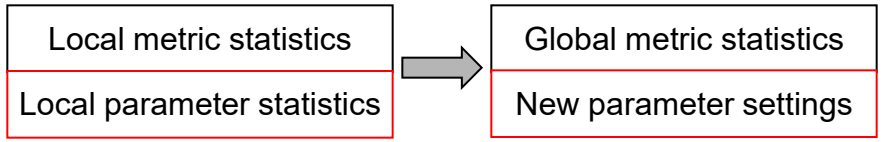
Statistics: metrics and paramters

Global view on metrics & **new** parameters



Planning step

- Various approaches possible
- My goal: closing the management cycle



Metric Update

Metric ACK

Management of the Hop Count in Chord

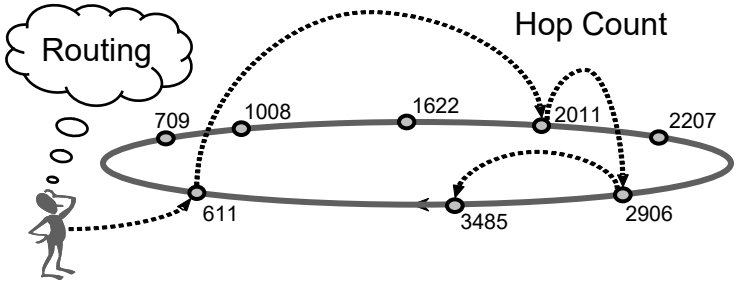
TECHNISCHE UNIVERSITÄT DARMSTADT

Managing the behavior of Chord

- Main metric: hop count

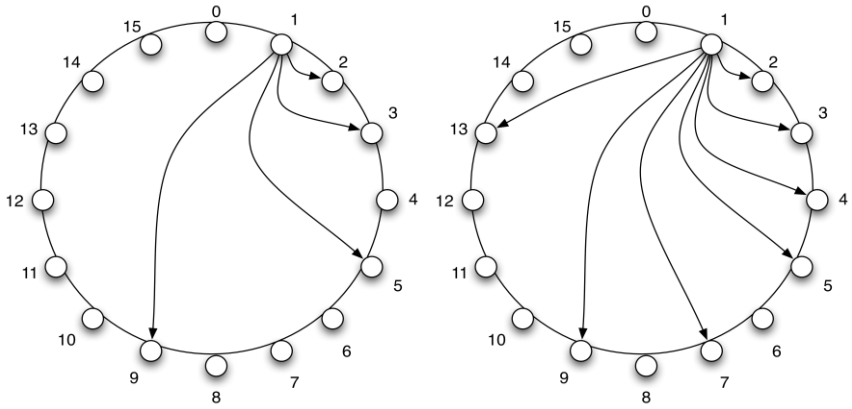
→ Target metric: [7, 10]

- Parameter: finger table size



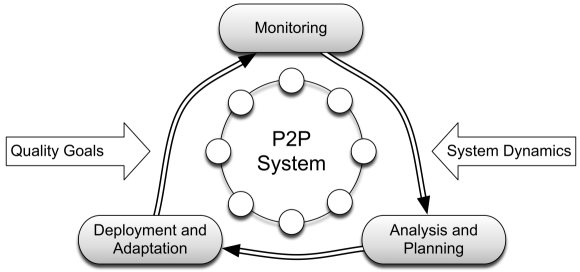
Rules for violation of the goal metric

Static Rule Approach	Finger table size
Hop count too small	- 10 %
Hop count too large	+ 100 %



Evaluation goal

- Target metrics are reached and held
- Feasibility of management cycle



Simulation Setup

- **N**=10000, **UI**=30s, **β**=4

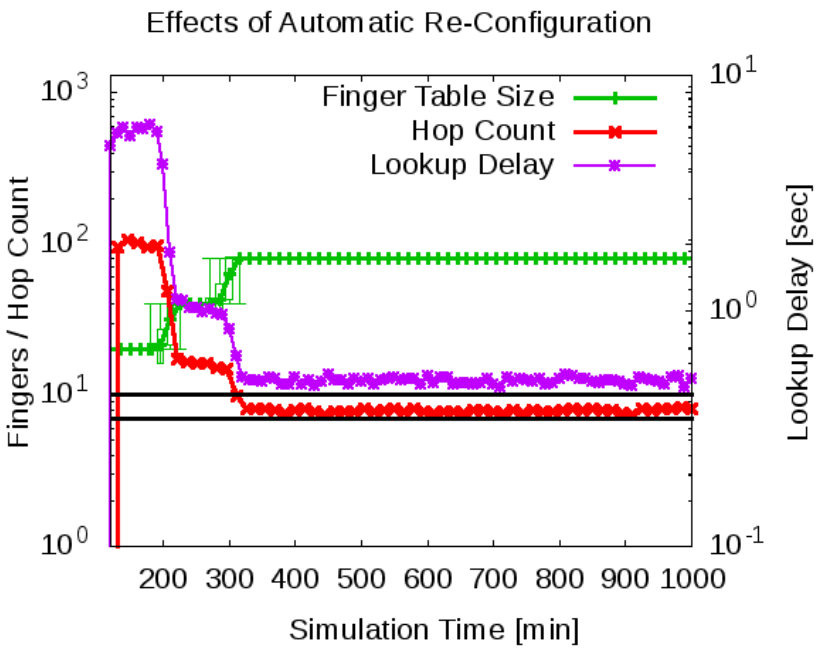
Evaluation Results on SkyNet.KOM

TECHNISCHE UNIVERSITÄT DARMSTADT

Results

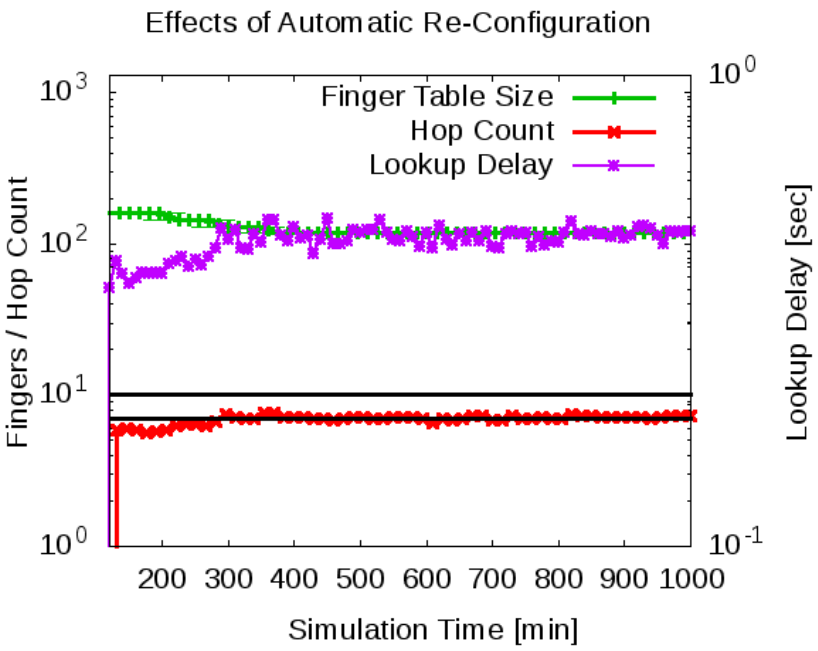
- Metric interval [7,10] is reached and held
- Management cycle is closed

Start: too large	Initial	2 Steps
Finger Table Size	20	80
Hop Count	100	8



- Target metric approximation from 2 sides
- Corresponding effects on lookup delay

Start: too small	Initial	3 Steps
Finger Table Size	160	117
Hop Count	5.7	7.1



Summary and Implications

TECHNISCHE UNIVERSITÄT DARMSTADT

Monitoring and management of p2p systems

- Always up-to-date view on system statistics
- Control of quality metrics through distributed control loop

Usage of p2p overlays “off the shelf”

- For various scenarios / environments

Generalization

- Applicable on any metrics and parameters

Research on

- Identification and modeling of interdependencies

Monitoring and management

- Towards p2p as reliable IT architecture

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction

Overview on my System Management Approach

- System Monitoring – SkyEye
- System Management – SkyNet

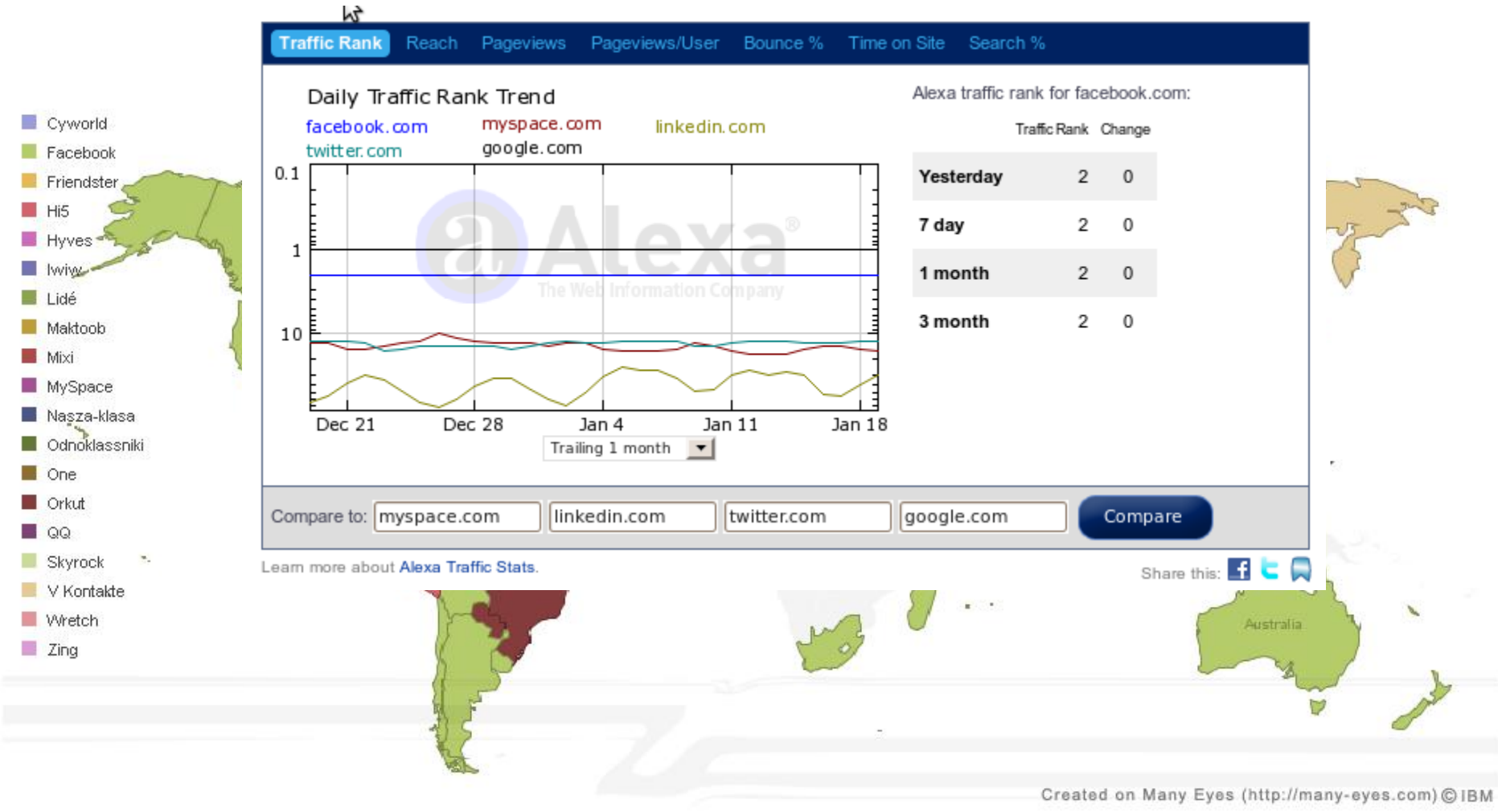
P2P-based Online Social Networks

- LifeSocial – Secure, Distributed, Controlled

Conclusions

What might be the next P2P application?

TECHNISCHE UNIVERSITÄT DARMSTADT



Online Social Networks

TECHNISCHE UNIVERSITÄT DARMSTADT

What are ‘Online Communities’ technically?

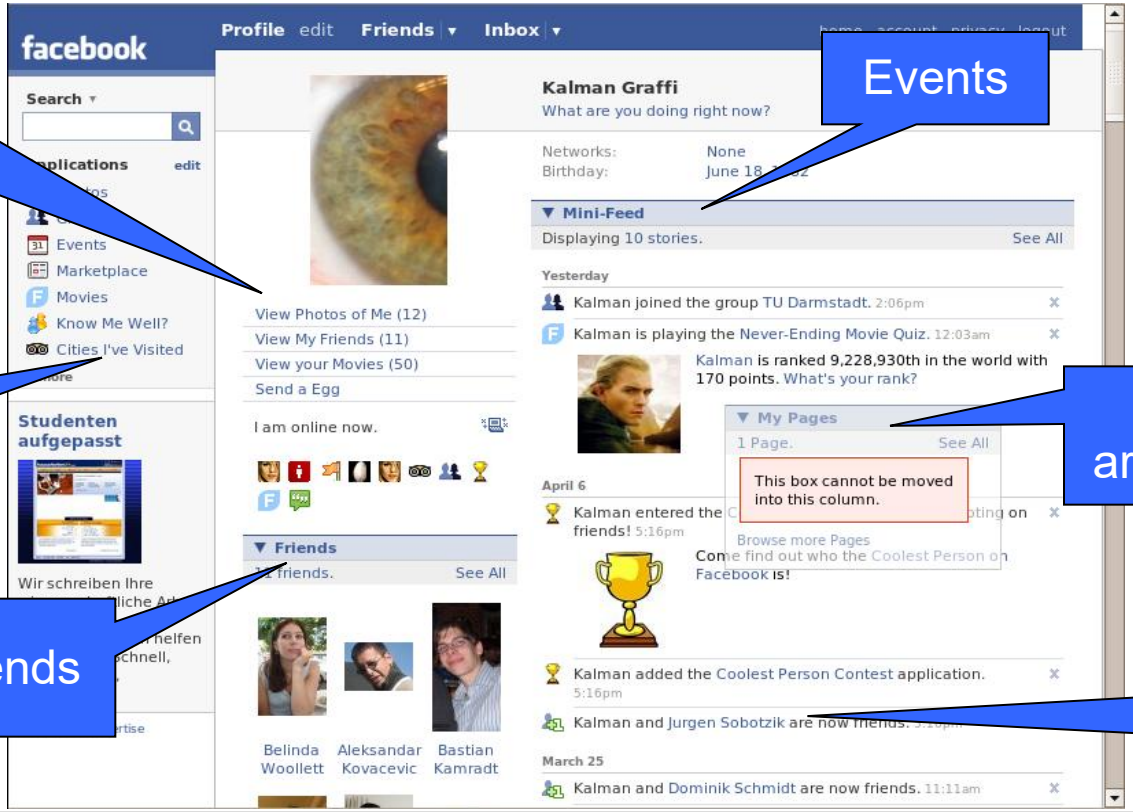
- Web-based applications (StudiVZ, Facebook, MySpace, Xing)
- Provide different services for community members

facebook

BETA 2.0 studiVZ DAS STUDIVERZEICHNIS

XING COMING SOON

myspace.com



Personal information and photos

Events

Games

Plugin architecture

Friends

Social interaction

Goals and Motivations

TECHNISCHE UNIVERSITÄT DARMSTADT

Users want

Storing and searching for content
- Profiles, friend lists, …
- Pictures, shared “Wall” editing, …



User to user interaction
- Chatting, VoIP, …
- Games



Security
- Access control on their data
- Secure, confidential communication



→ **Fun!**

System providers want

High profit
- Many users
- Personalized advertisements



Low operational costs
- For servers, electricity, cooling …
- For personnel, legal issues



Controlled Quality of Service
- To attract and keep users
- Providing reliable, high quality services



→ **Money!**

Our goal: all of the above following the P2P paradigm

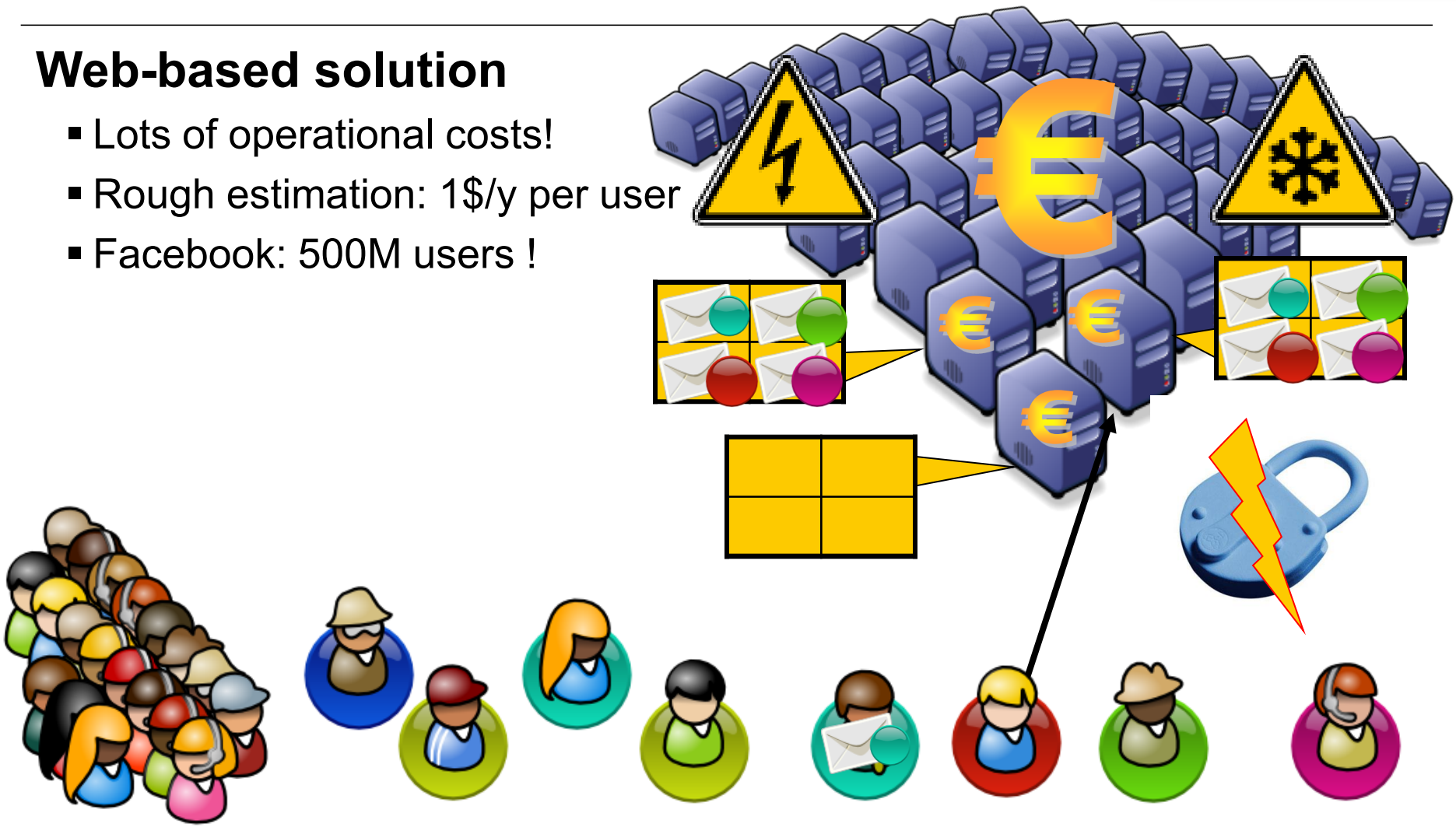
How do they work?
What is the architecture beneath?

Current IT Paradigm: Client / Server

TECHNISCHE UNIVERSITÄT DARMSTADT

Web-based solution

- Lots of operational costs!
- Rough estimation: 1$/y per user
- Facebook: 500M users !



Alternatives? – Peer-to-Peer based Platforms

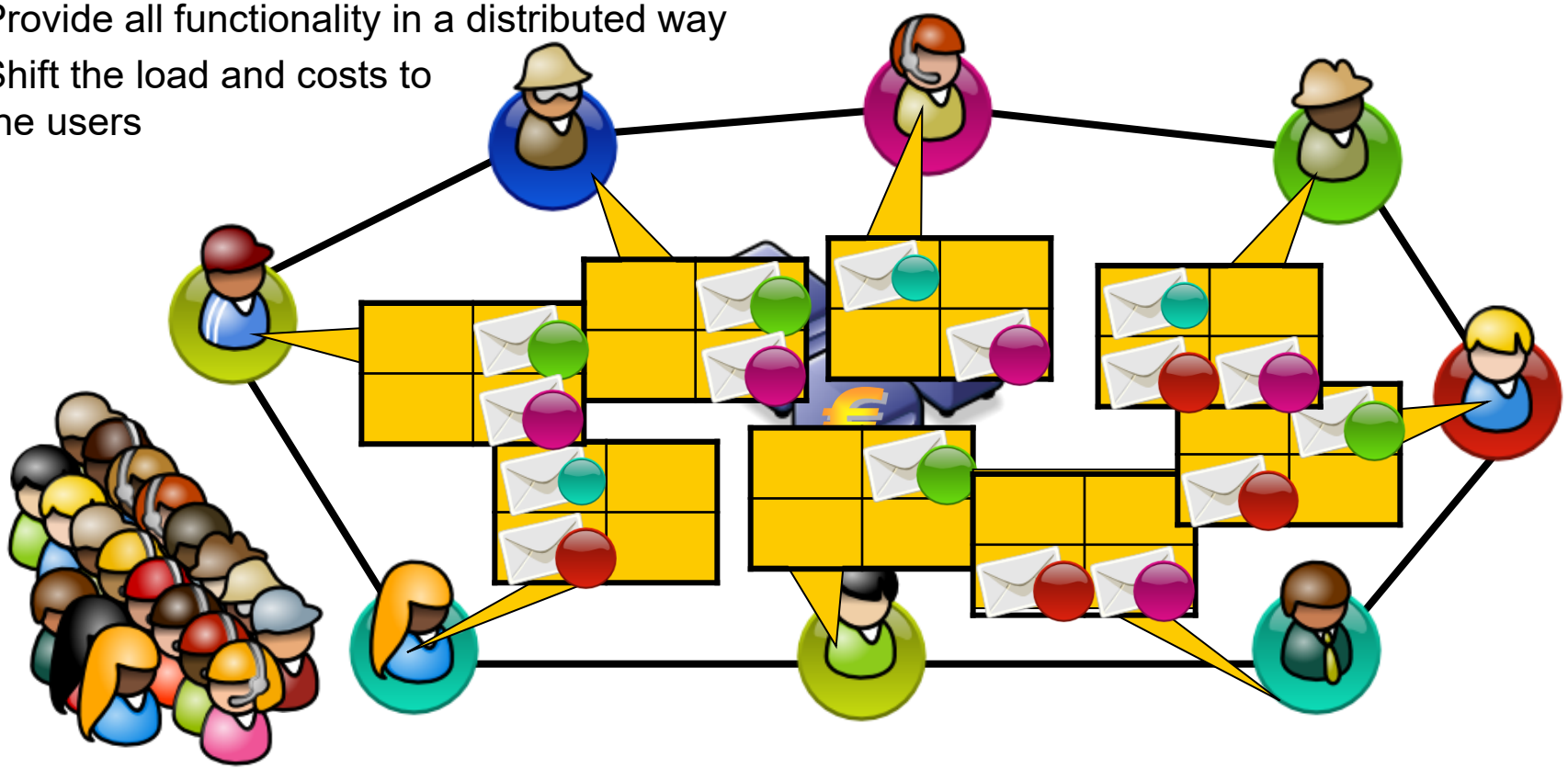
TECHNISCHE UNIVERSITÄT DARMSTADT

Idea:

- Use capacities of user devices (Moore's law!)
- Interconnect users with p2p-overlay
- Provide all functionality in a distributed way
- Shift the load and costs to the users

Platforms:

- LifeSocial.KOM
- SafeBook, PeerSon



Our Solution: LifeSocial.KOM

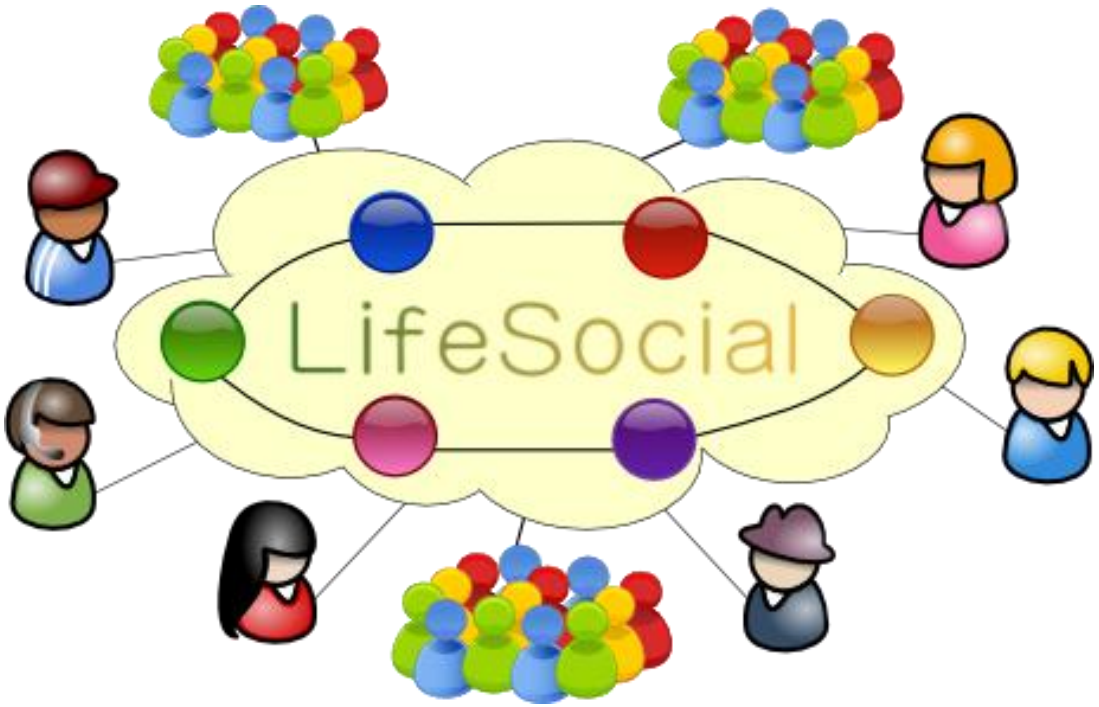
TECHNISCHE UNIVERSITÄT DARMSTADT

Researched since end of 2007

- Ca. 10 diploma / bachelor theses on this topic
- Ca. 20 students programming plugins / GUIs in “Praktika” / project seminars

See: www.lifesocial.org





How does it look like?

What can you do?

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction

Overview on my System Management Approach

- System Monitoring – SkyEye
- System Management – SkyNet

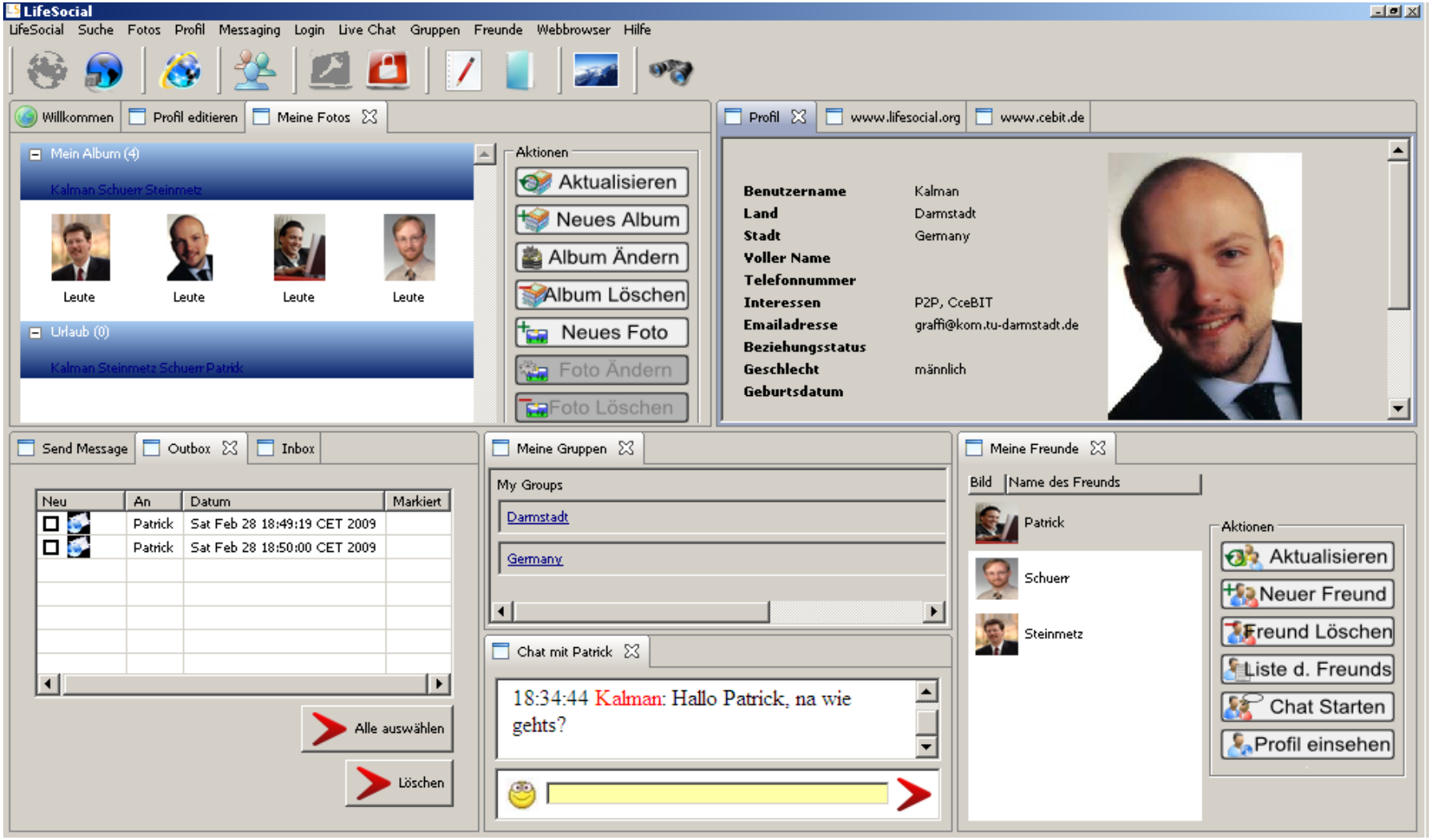
P2P-based Online Social Networks

- LifeSocial – Secure, Distributed, Controlled

Conclusions

Screenshots

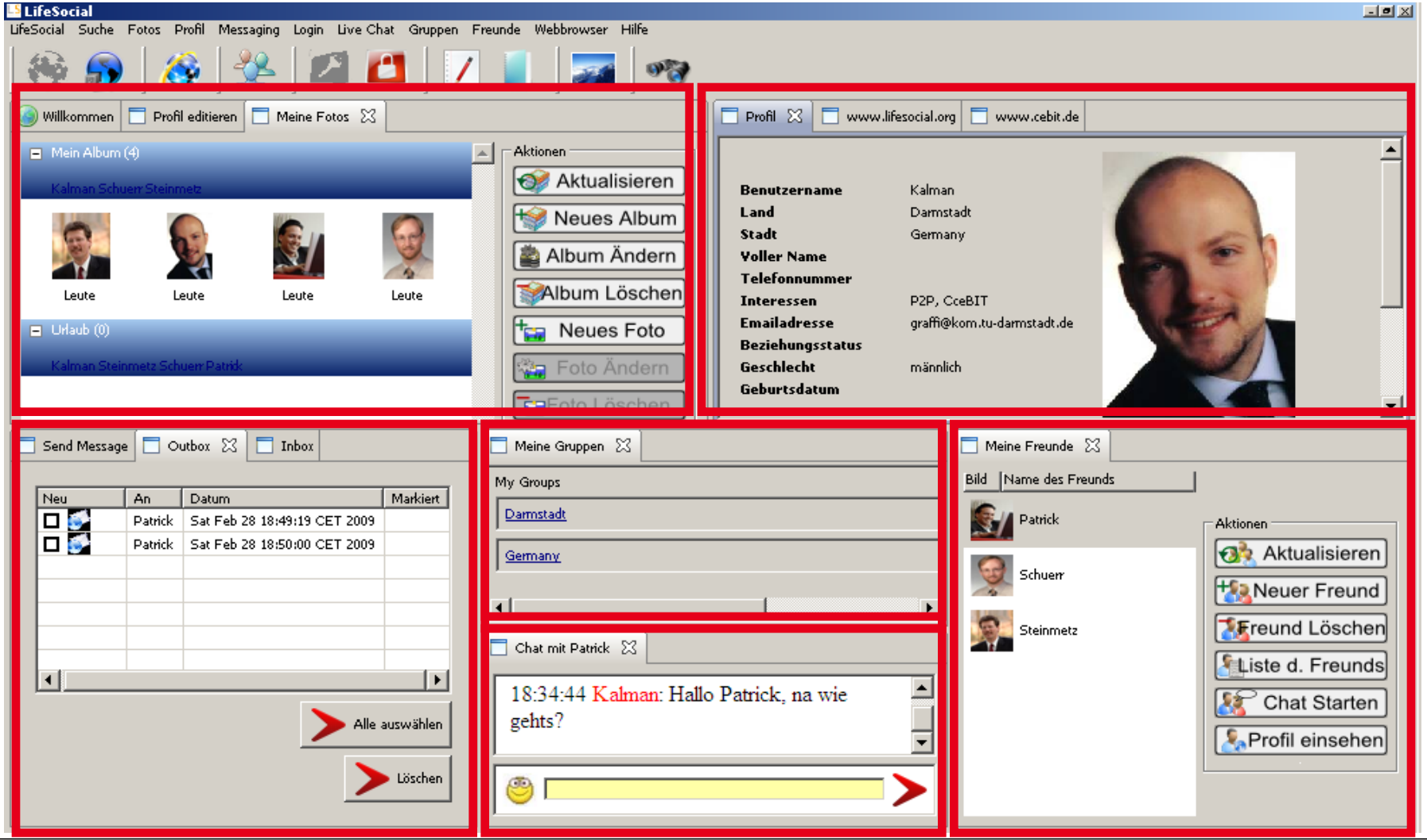
TECHNISCHE UNIVERSITÄT DARMSTADT



→ See: www.lifesocial.org

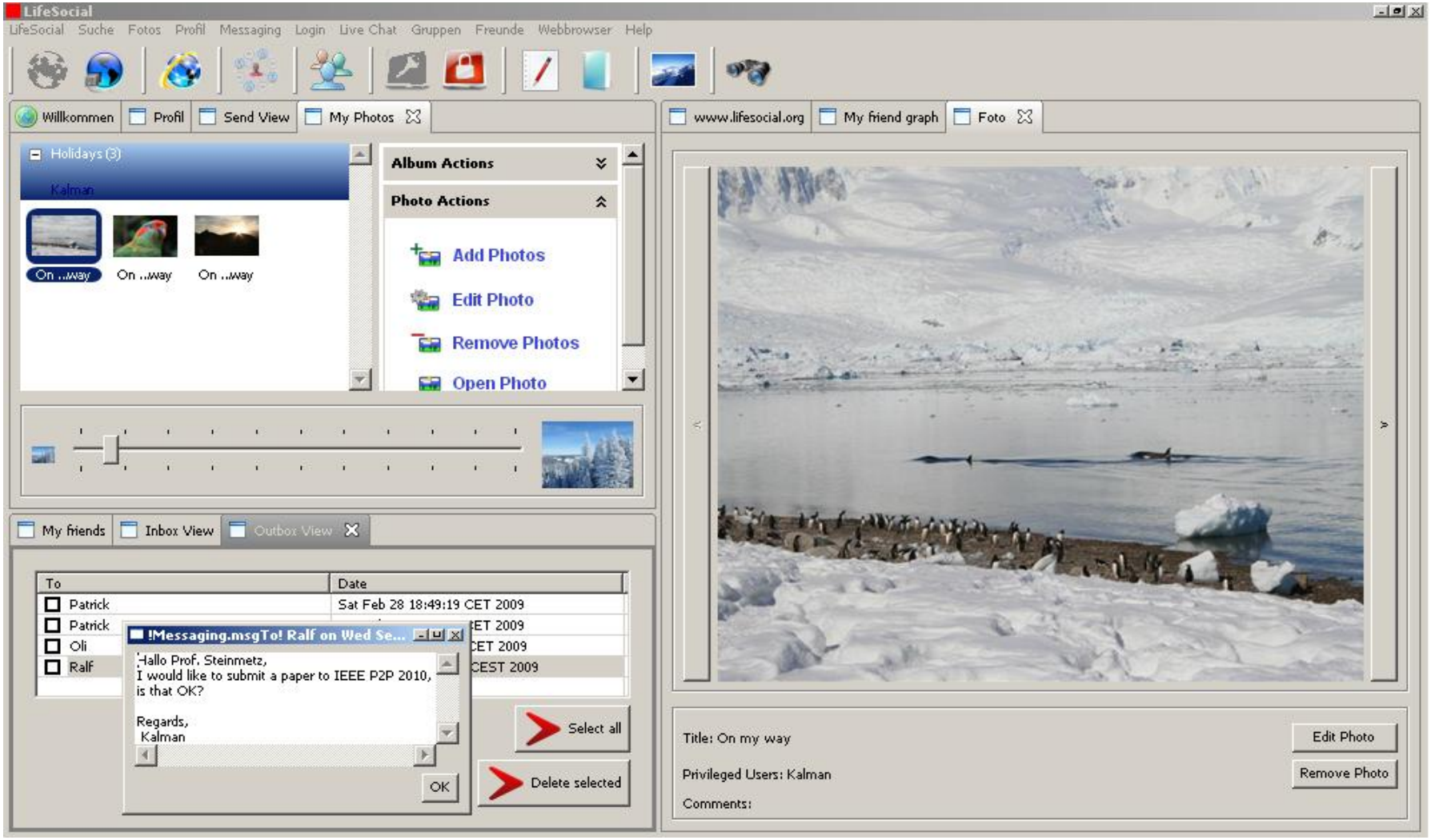
See: www.lifesocial.org

TECHNISCHE UNIVERSITÄT DARMSTADT



Screenshots

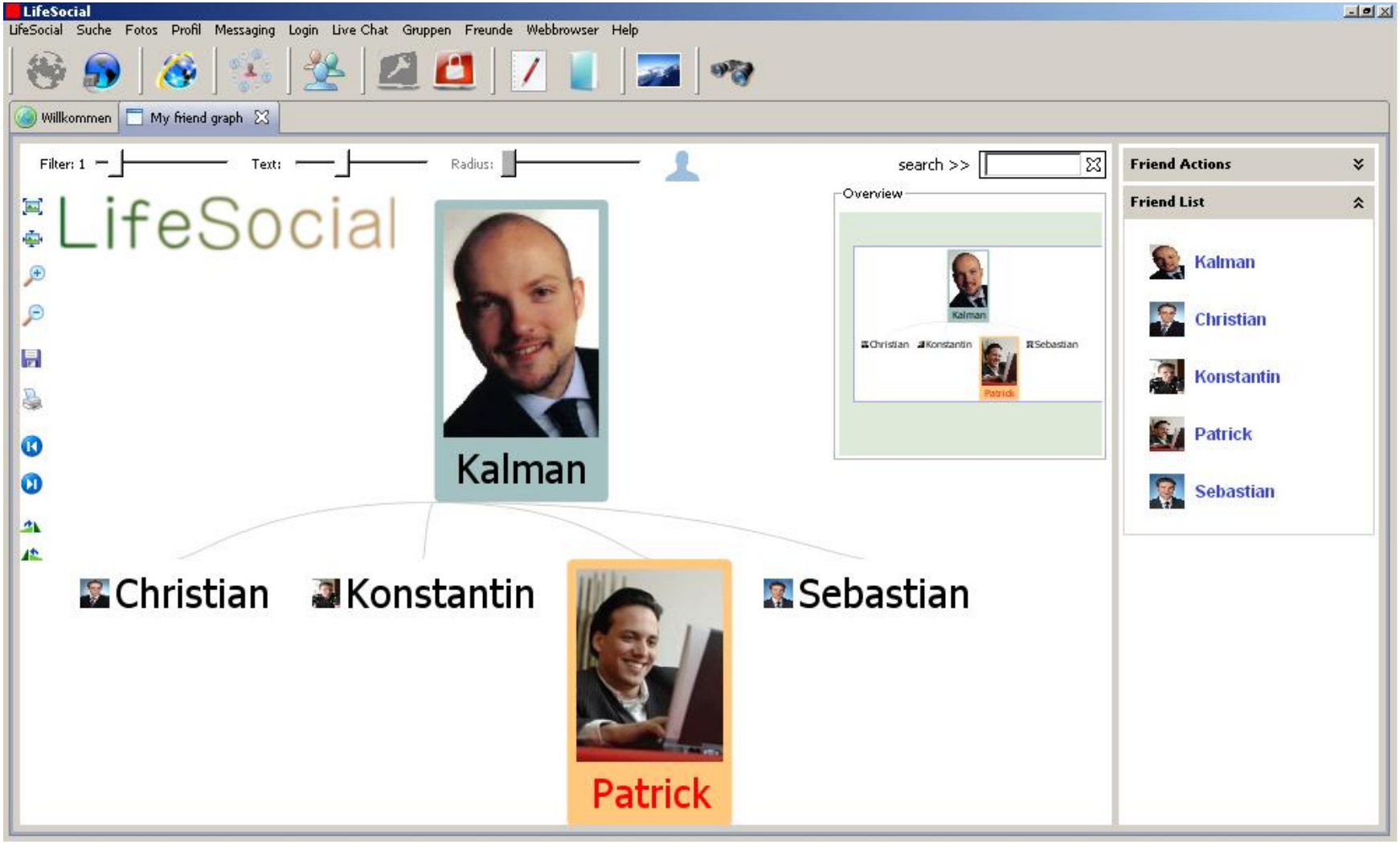
TECHNISCHE UNIVERSITÄT DARMSTADT



→ See: www.lifesocial.org

Screenshots

TECHNISCHE UNIVERSITÄT DARMSTADT



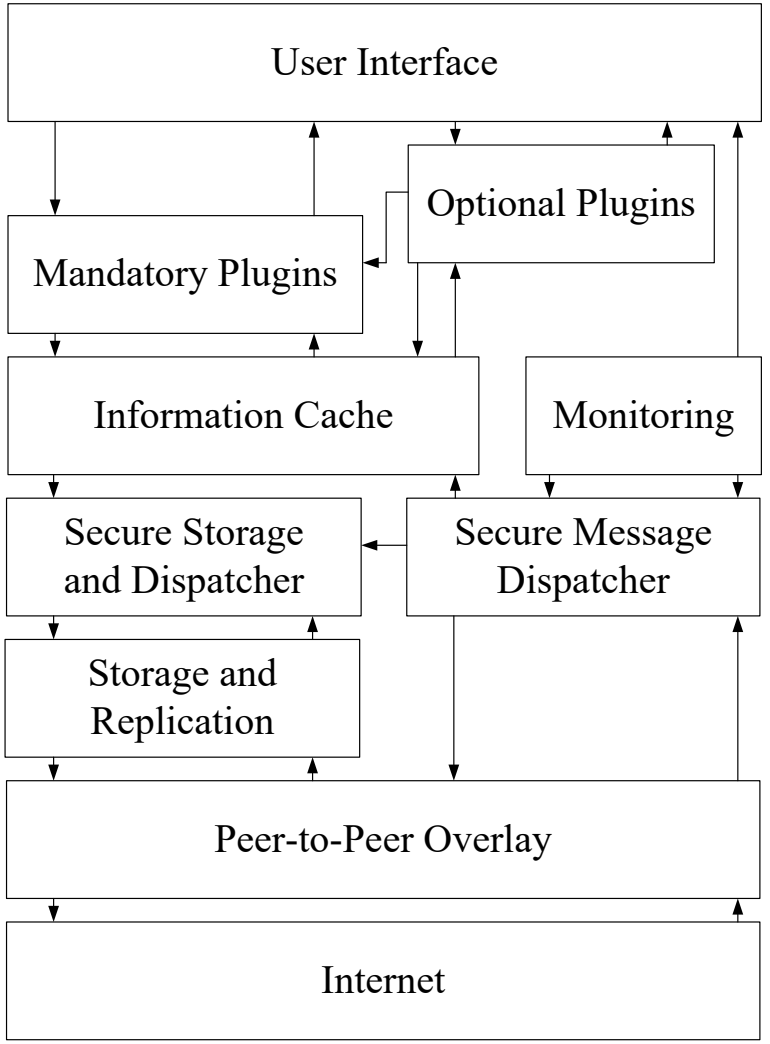
→ See: www.lifesocial.org

How does this work?
What is the architecture beneath?

Architecture Overview on LifeSocial.KOM

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Extendable framework for user interface components**
- **Stand-alone applications, core functionality and optional functionality of the system. Extendable.**
- **Caching of data objects and messages**
- **Monitoring of the quality of service**
- **Low-delay user-to-user communication**
- **Storage (store, modify, retrieve, delete)**
- **Distributed storage and replication**
- **Organization of nodes in an overlay network**
- **Standard Internet protocols**



Challenges and Lessons Learned

TECHNISCHE UNIVERSITÄT DARMSTADT

Interconnecting the peers

- Overlay needed for ID-based, consistent routing
- Issues:
  - For academia (Chord, CAN)
  - Different purpose (Kademlia, unstructured overlays)
  - Homebrew: design and evaluation takes time

→ **FreePastry**

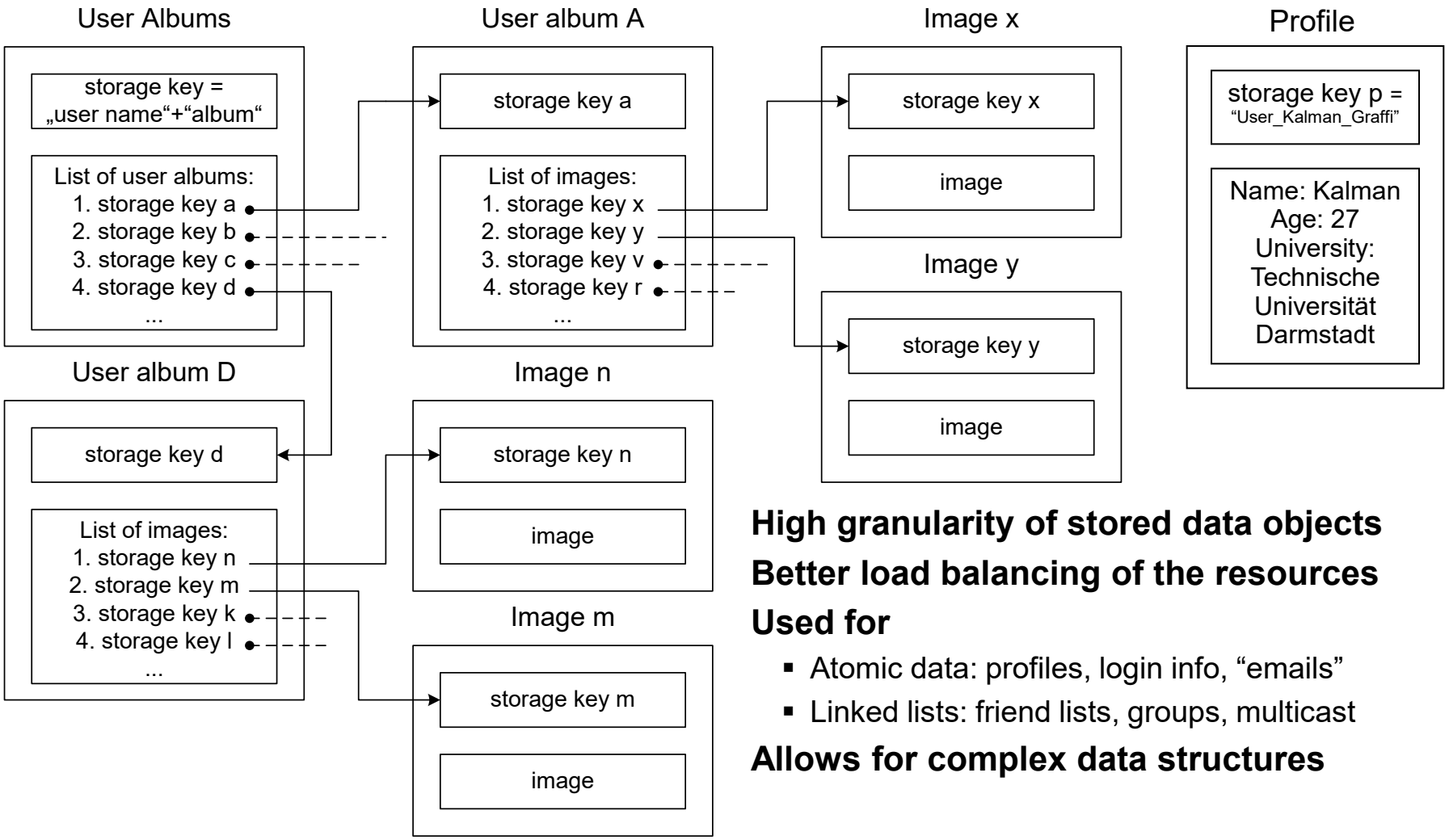
Data Storage / Replication

- Reliable + consistent data storage: read, write, update
- Load balancing?
- Even more complicated

→ **PAST, comes with FreePastry**

- ID-based storage and retrieval

Document Types, Obvious Storage Keys



High granularity of stored data objects

Better load balancing of the resources

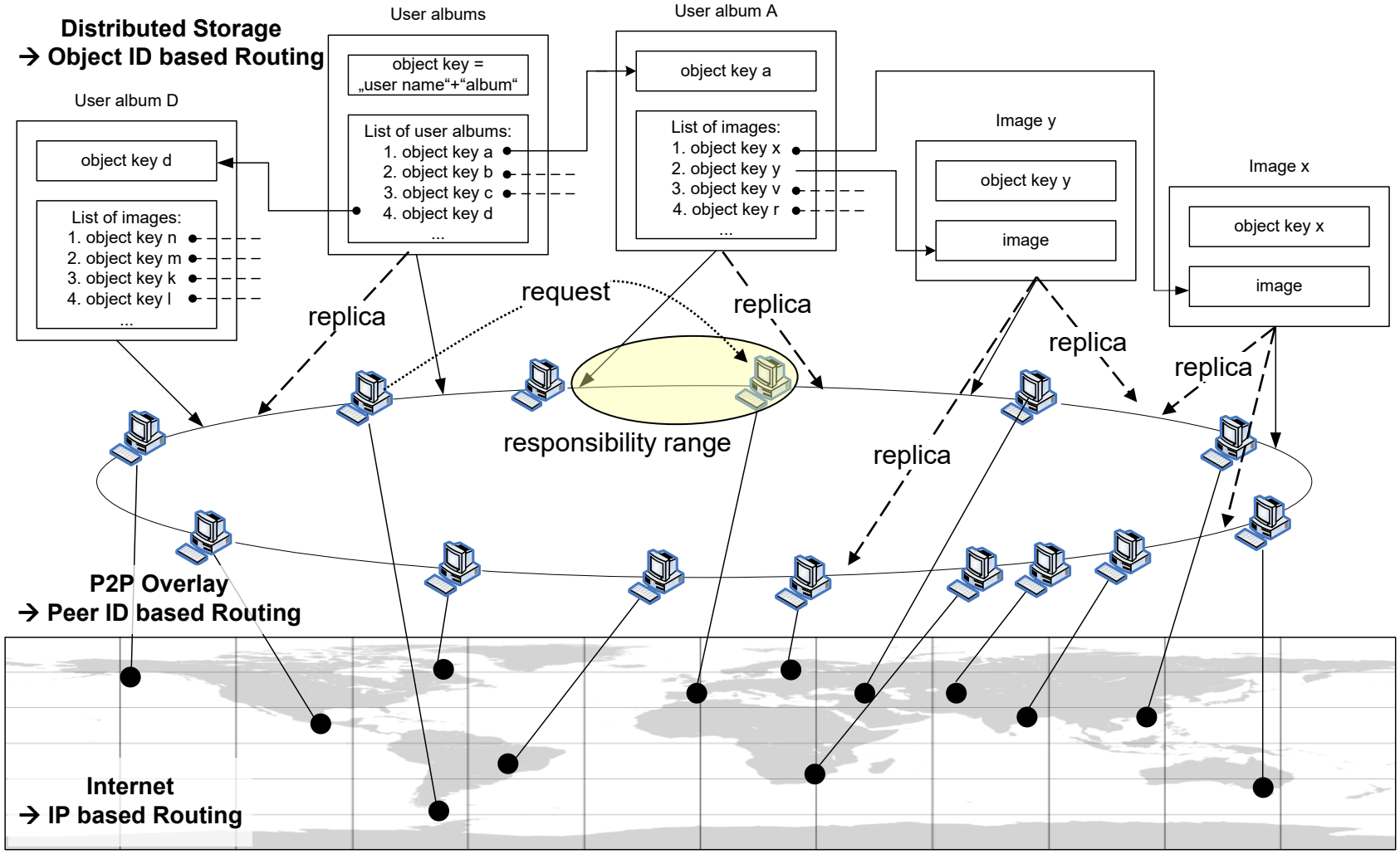
Used for

- Atomic data: profiles, login info, “emails”
- Linked lists: friend lists, groups, multicast

Allows for complex data structures

→ See: K. Graffi et al., “A Distributed Platform for Multimedia Online Communities” In: IEEE International Symposium on Multimedia '08 (IEEE ISM’09), December 2008.

Data Positioning in the Network



SECURITY

TECHNISCHE UNIVERSITÄT DARMSTADT

Security is 2nd most important

- After efficiency!

Goals:

- Authentification of hosts
- Encrypted messaging
- Access control lists (on sensible data)

Idea:

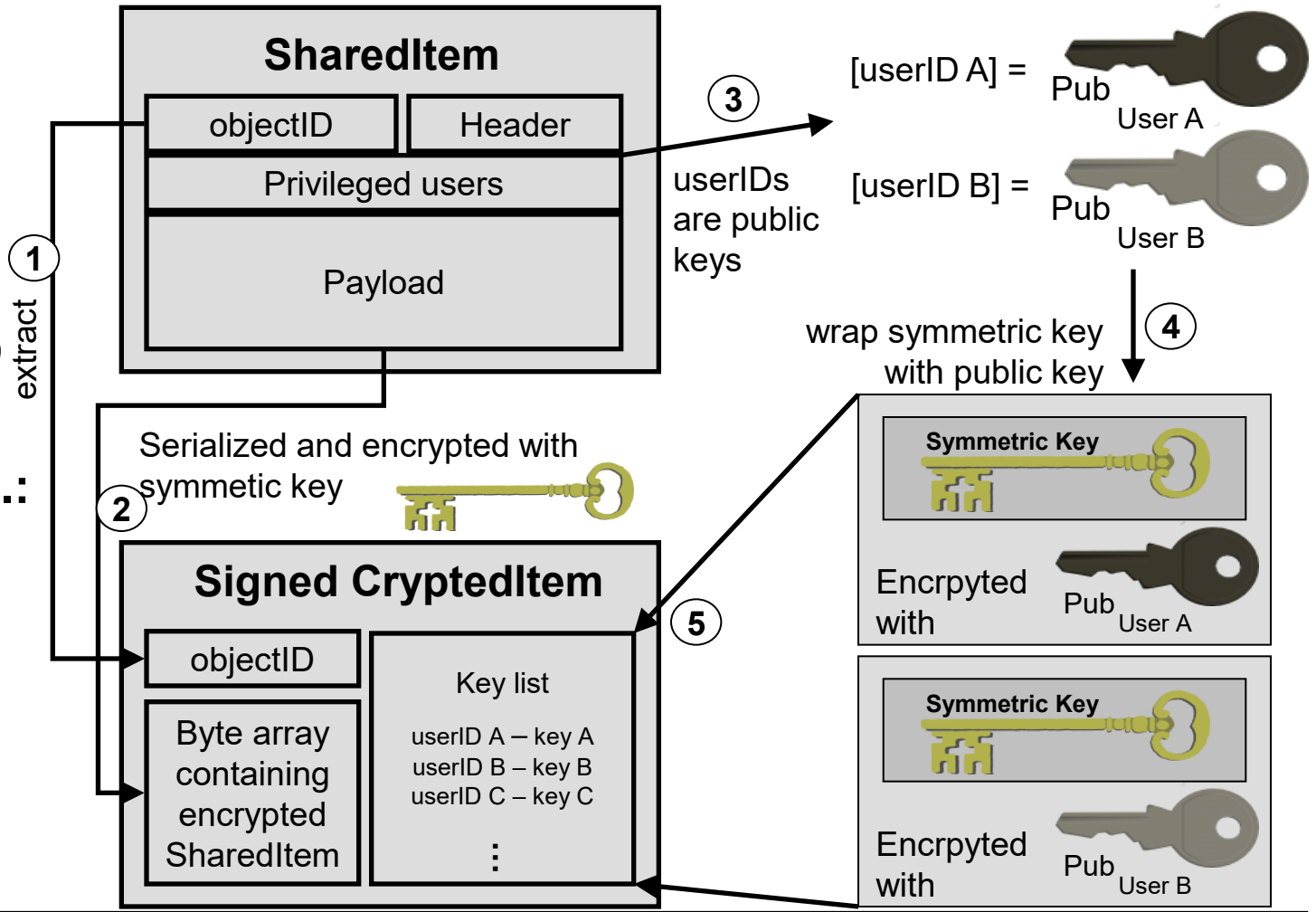
- Use PublicKeys as NodeIDs
  → allows instant authentication and encrypted communication
- Encrypt all stored data with unique symmetric key
  - Encrypt the symmetric key for all privileged reader
  - Attach the ENCRYPTED symmetric key to the encrypted data

Detailed Idea of Distributed Access Control

TECHNISCHE UNIVERSITÄT DARMSTADT

Step 1:
PeerID
= UserID
= PublicKey
= f(username, passphrase)

Secure comm.:
- encrypted
- signed



→ See: K. Graffi et al., "Practical Security in P2P-based Social Networks" In: IEEE Local Computer Networks '09 (IEEE LCN'09), October 2009.

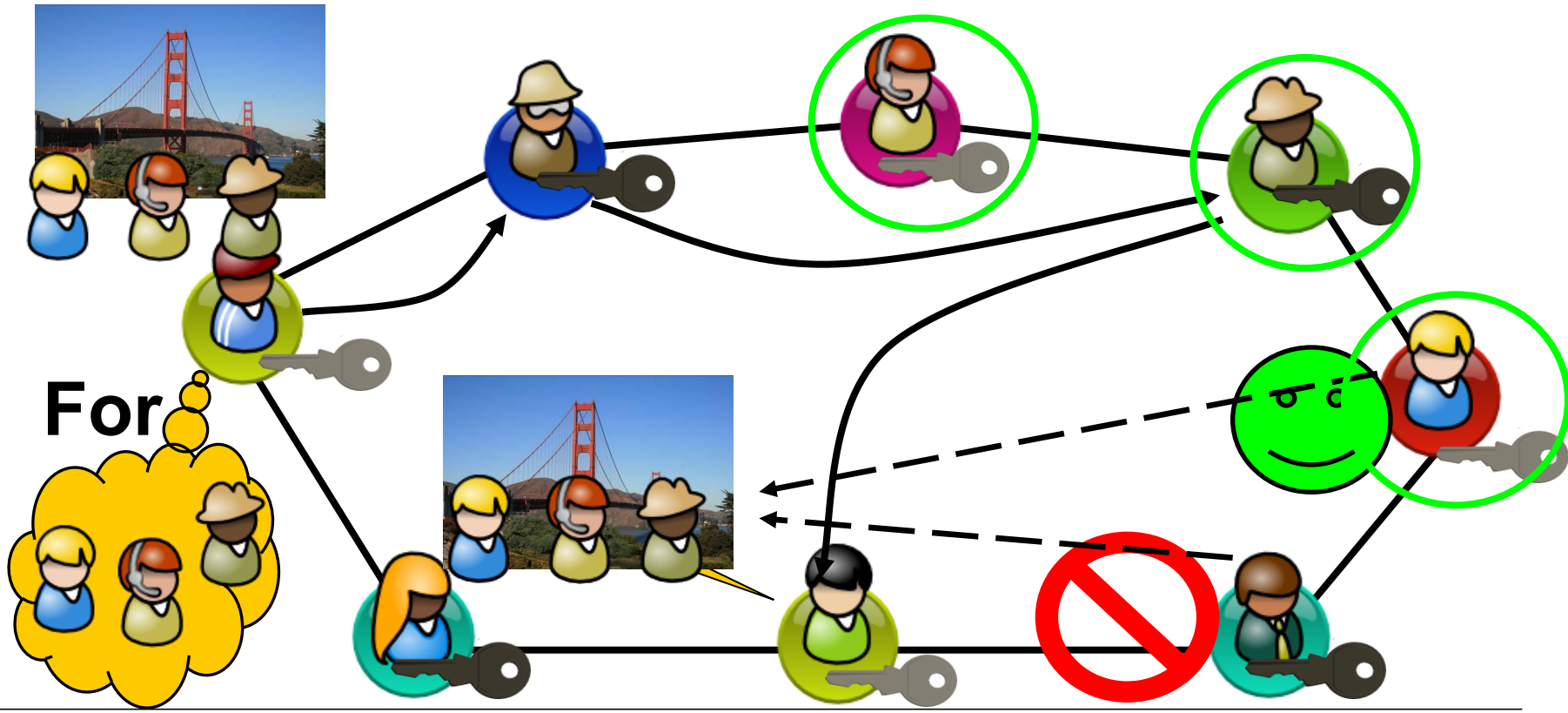
Simple Idea of Distributed Access Control

TECHNISCHE UNIVERSITÄT DARMSTADT

How to provide Access Control in a distributed environment?

Goal: Assign read-rights on objects to privileged users

Mechanism: Sym. encrypted objects, asym. encrypted sym. keys



→ See: K. Graffi et al., "Practical Security in P2P-based Social Networks" In: IEEE Local Computer Networks '09 (IEEE LCN'09), October 2009.

When it is distributed,

how do you know that it works?

What is the quality?

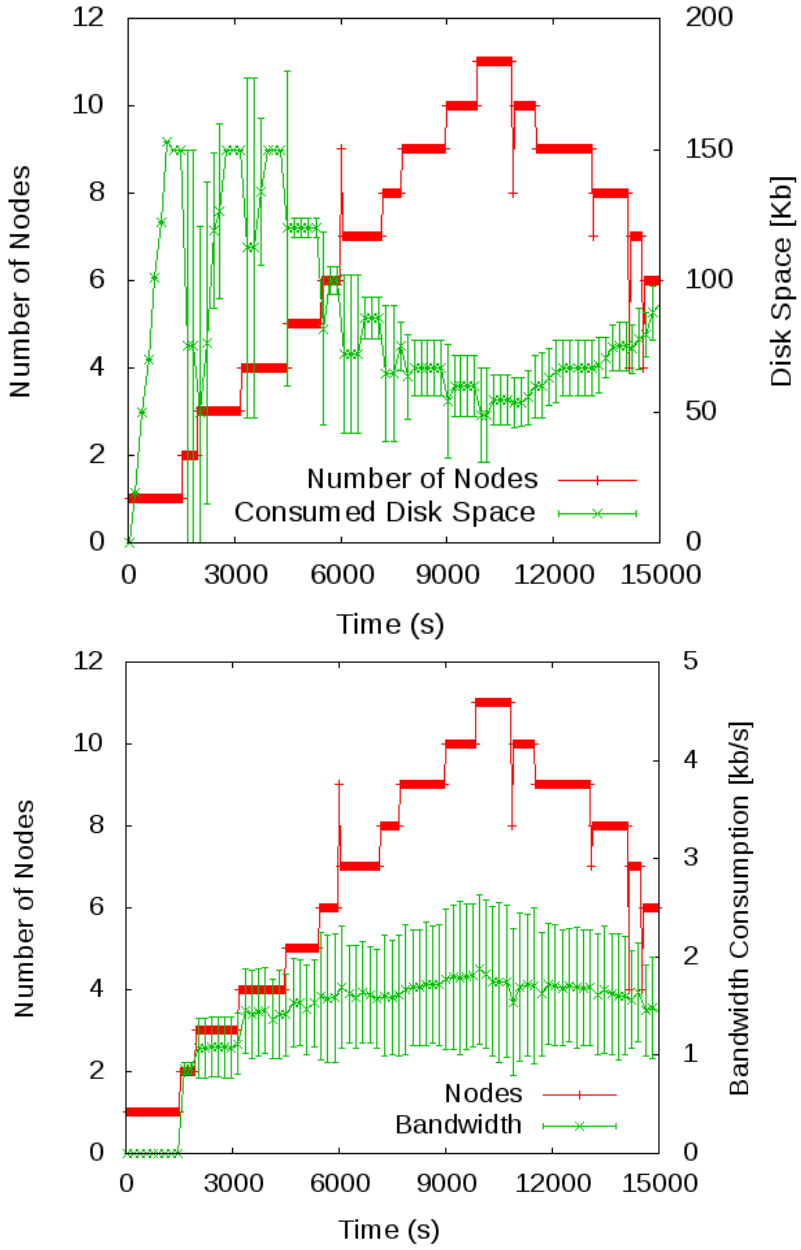
Monitoring and Evaluation

Integration of a monitoring solution

- Totally distributed, precise and cheap

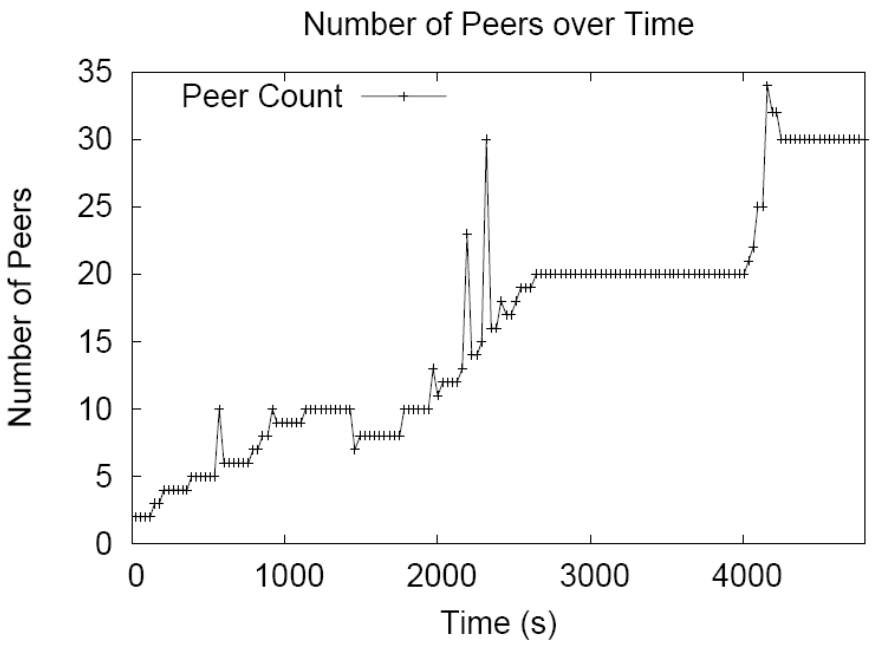
Global system statistics

- Statistics on
  - CPU / bandwidth usage
  - Data retrieval delays
  - Messages sent / received
  - Number of peers
  - Objects in Cache
  - Friends and clustering coefficient
  - …
- Statistical information:
  avg, min, max, standard dev., sum,...

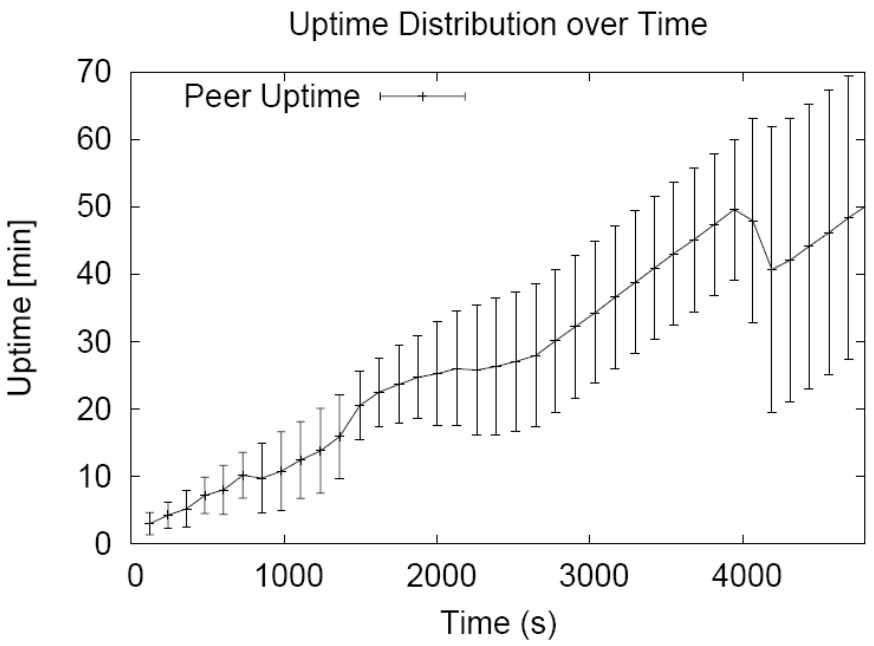


→ See: K. Graffi et al., "Monitoring and Management of Structured Peer-to-Peer Systems" In: IEEE Peer-to-Peer Computing '09 (IEEE P2P'09), September 2009.

Some Statistics (1)

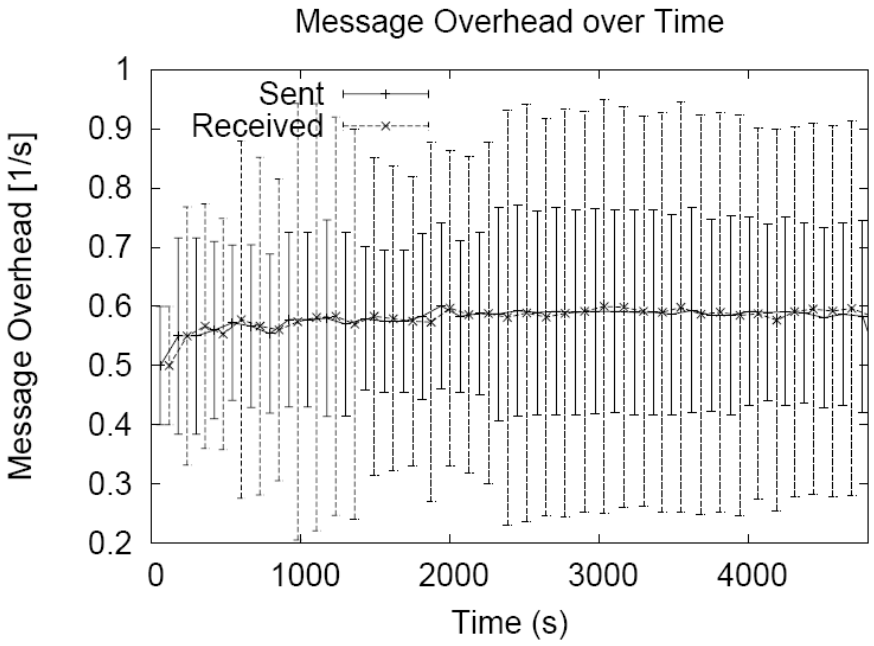


(a) Peer Count

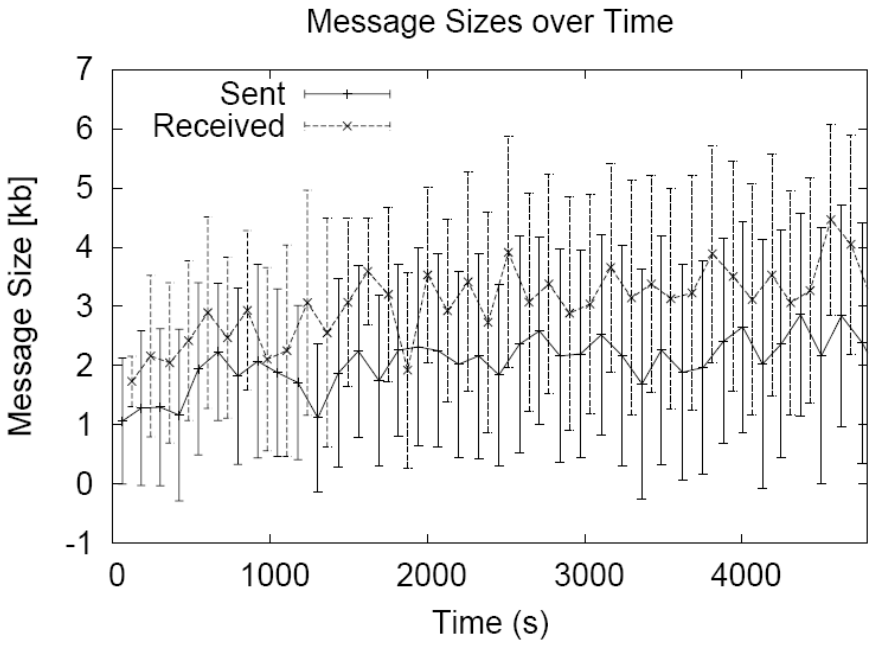


(b) Online Time Distribution

Some Statistics (2)

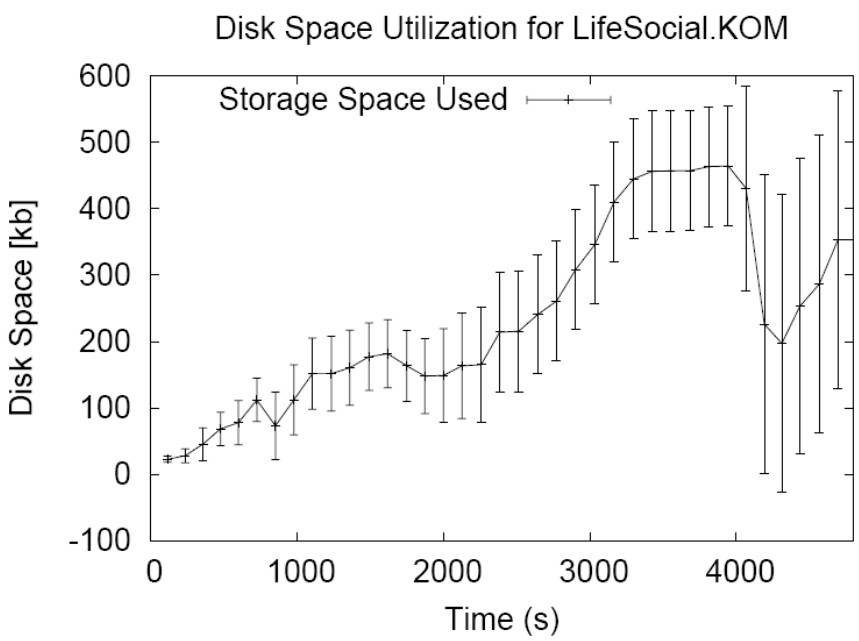


(a) Messaging Overhead

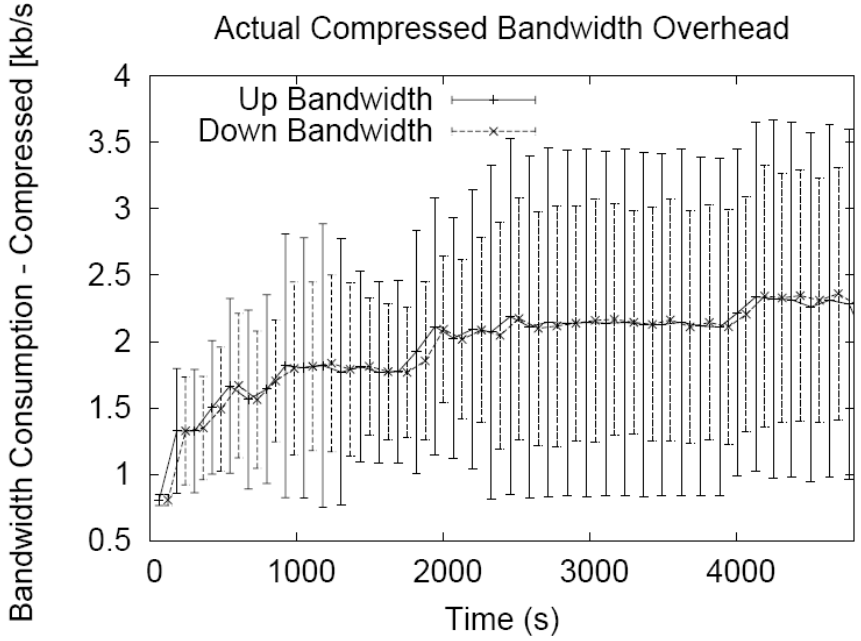


(b) Message Size Distribution

Some Statistics (3)



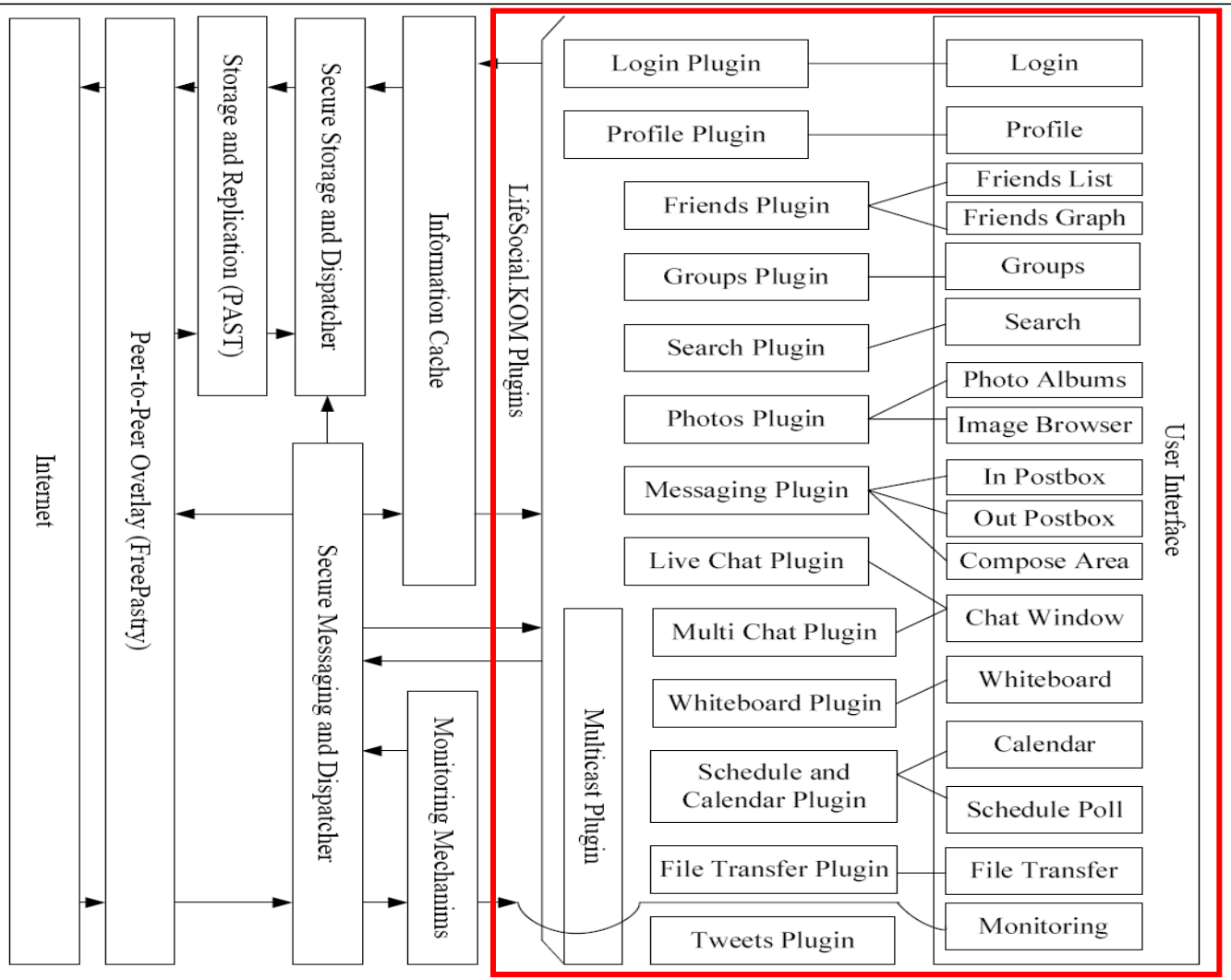
(a) Disk Usage of LifeSocial.KOM



(b) Compressed Bandwidth Usage

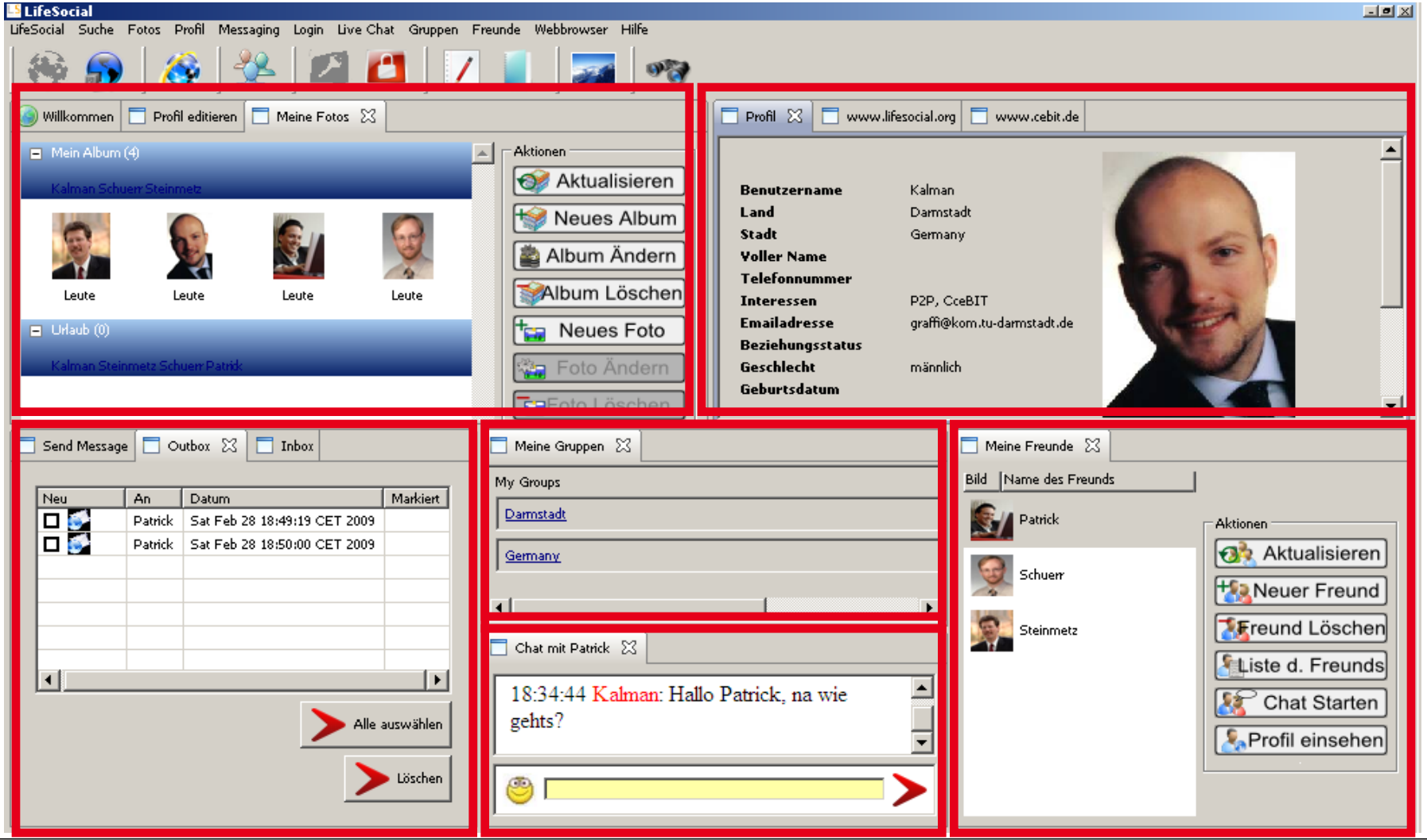
Plugin Architecture Overview

TECHNISCHE UNIVERSITÄT DARMSTADT



See: www.lifesocial.org

TECHNISCHE UNIVERSITÄT DARMSTADT



Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

IT solutions for social networks

- Currently centralized and very costly
- Scales only with high monetary invests

Distributed, p2p-based platforms

- Data storage is totally distributed
- Costs are shared among the users

LifeSocial.KOM

- Operational prototype
- Secure, reliable storage and messaging
- Monitoring mechanism to observe (and control) the quality of service
- Rich, extendable functionality through Plugin-based architecture
- See videos on www.lifesocial.org

Analysis of needs:

Users want

- Storing and searching for content 
- User to user interaction 
- Security 

System provider want

- Low operational costs 
- Controlled quality of service 
- High profit 

Next steps:

- Integrate management mechanisms
- Run Internet-wide beta-test
- Deploy

Issues and Challenges in Academia

TECHNISCHE UNIVERSITÄT DARMSTADT

Engineering a prototype is not considered as research

Programming effort hard to mount

How to test large-scale distributed systems?

Successfull standalone P2P application known?

- Filesharing?
- Skype? → is it working now?
- ???

Who needs P2P – we have the cloud!

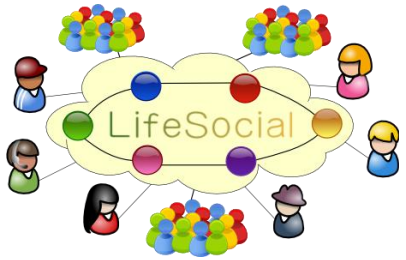
- Guaranteed quality! Controllable costs!
- Easier to maintain / operate

Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

System management: SkyNet.KOM

- Reach and hold preset metric intervals
- Through monitoring and self-configuration

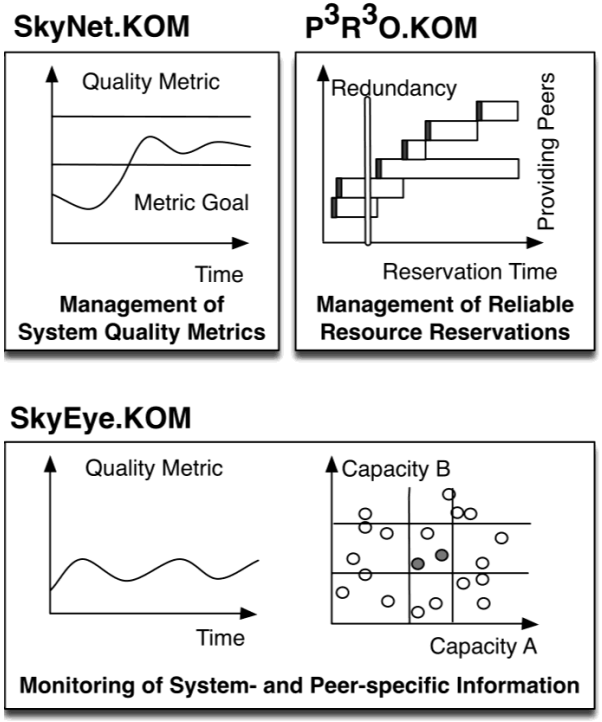


System monitoring: SkyEye.KOM

- Global view on statistics of running system
- Precise yet cost effective monitoring

Resource management

- Convenient access to peer capacities
- Reliable capacity reservation



Application scenario: LifeSocial

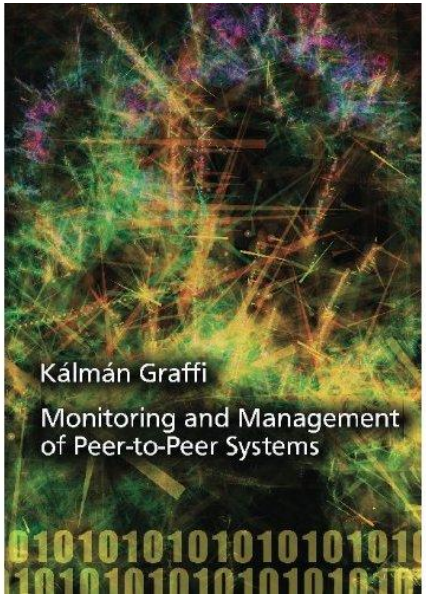
- P2P-based online social network
- Demonstration of feasibility

Questions?

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

QuaP2P

TECHNISCHE UNIVERSITÄT DARMSTADT



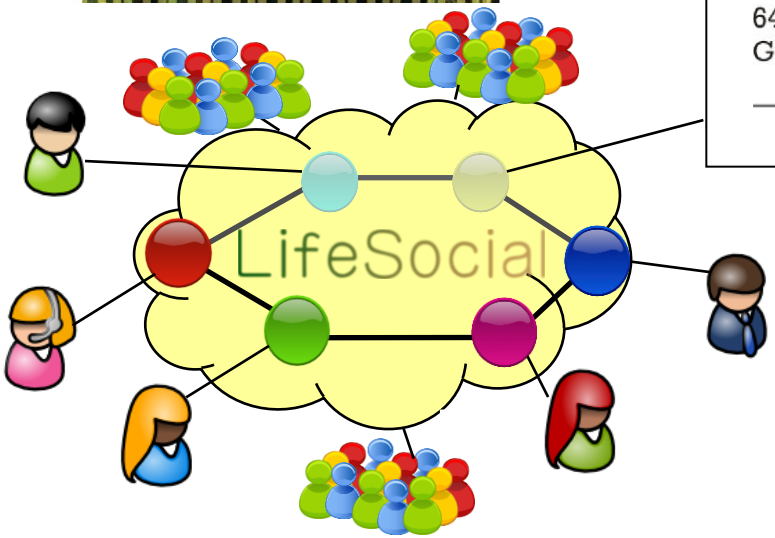
Department of Electrical Engineering
and Information Technology
Multimedia Communications Lab - KOM

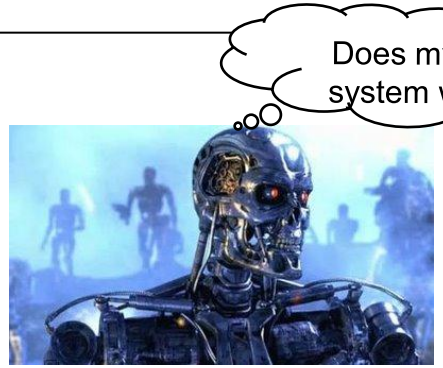
TECHNISCHE UNIVERSITÄT DARMSTADT

Dr.-Ing. Kalman Graffi

Kalman.Graffi@KOM.tu-darmstadt.de
Rundeturmstr. 10
64283 Darmstadt
Germany

Phone +49 (0) 6151/164959
Fax +49 (0) 6151/166152
www.kom.tu-darmstadt.de





Have a look at:

www.lifesocial.org

www.skynet-project.com

www.kom.tu-darmstadt.de

Backup Slides

Monitoring of P2P Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

System view:

- Global statistics on set of metrics
- Avg., std. dev., sum, min., max.
- Long list of aggregatable metrics

Example:

- Global statistics on bandwidth usage

Use:

- Observe quality (performance, costs) of running p2p system

Peer view:

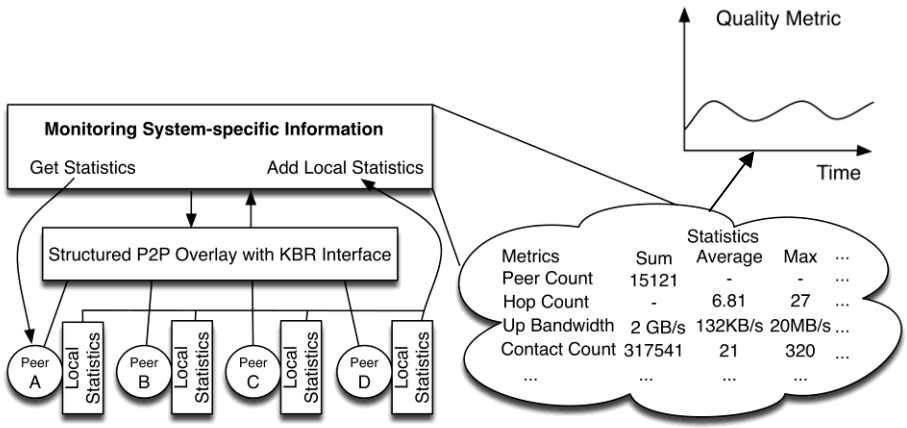
- Overview on the peers' capacities
- Peer specific capacity information
- Allows capacity-based peer search

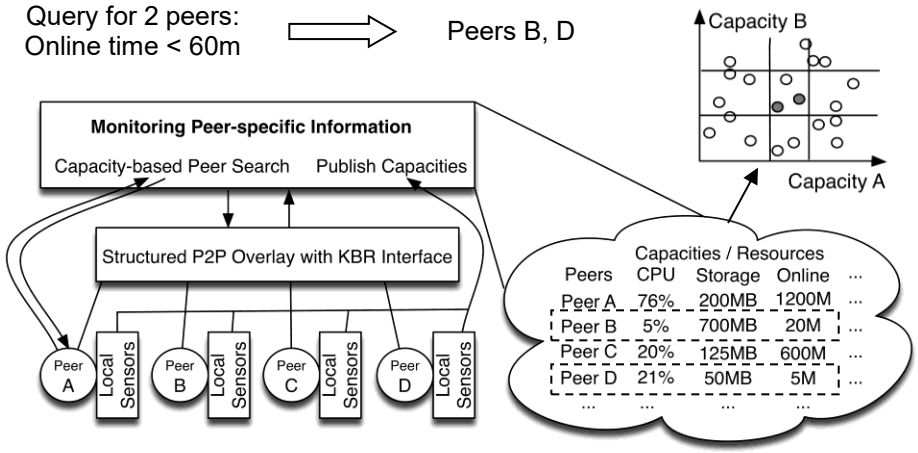
Example:

- Bandwidth capacities of a single peer

Use:

- Mechanisms find desired capacities



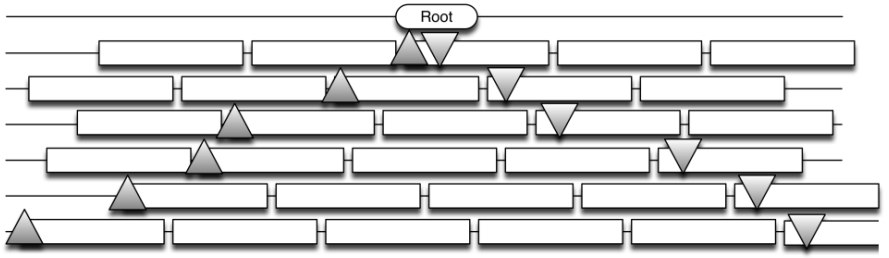


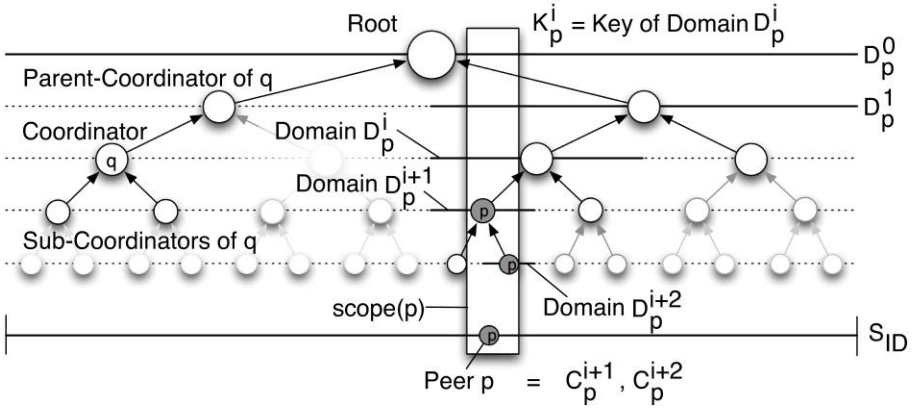
SkyEye.KOM: Protocols

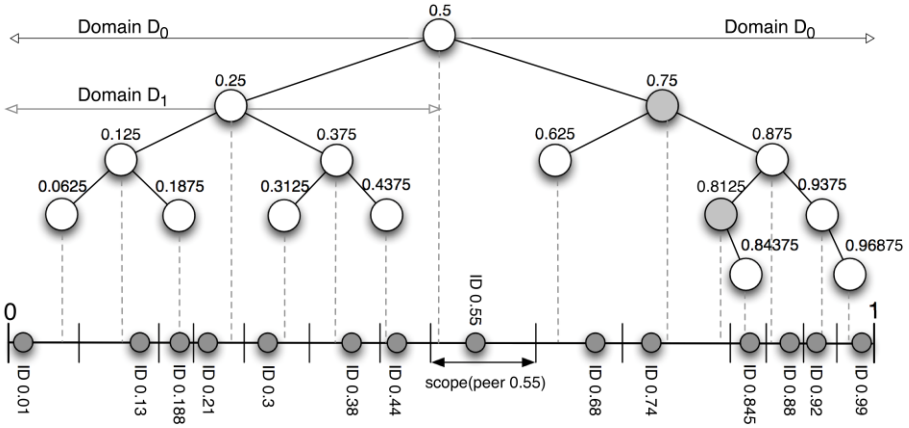
$$f(v_1, v_2) = f(v_2, v_1)$$
$$f(v_1, v_2, v_3) = f(f(v_1, v_2), v_3) = f(v_1, f(v_2, v_3))$$

Function	Description
count	Counts the number of metric values
min	Calculates the minimal value of a metric
max	Calculates the maximal value of a metric
sum	Sums the values of a metric
sum_squares	Sums the squares of the values of a metric
mean	Calculates the average of a metric
variance	Calculates the variance of a metric
standard_deviation	Calculates the sample standard deviation of a metric

Aggregation Functions
$count(a_1, \ldots, a_n) = n$
$min(a_1, \ldots, a_n) = min(a_1, min(a_2, \ldots min(a_{n-1}, a_n) \ldots))$
$max(a_1, \ldots, a_n) = max(a_1, max(a_2, \ldots max(a_{n-1}, a_n) \ldots))$
$sum(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i$
$sum_squares(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i^2$
$mean(a_1, \ldots, a_n) = \frac{sum(a_1, \ldots, a_n)}{count(a_1, \ldots, a_n)}$
$variance(a_1 \ldots a_n) = \frac{sum_squares(a_1, \ldots, a_n)}{count(a_1, \ldots, a_n)} - \left(\frac{sum(a_1, \ldots, a_n)}{count(a_1, \ldots, a_n)}\right)^2$
$standard_deviation(a_1, \ldots, a_n) = \sqrt{variance(a_1, \ldots, a_n)}$







Interdependencies: Finger Table Size, Contacts, Hop Count, Lookup Delay

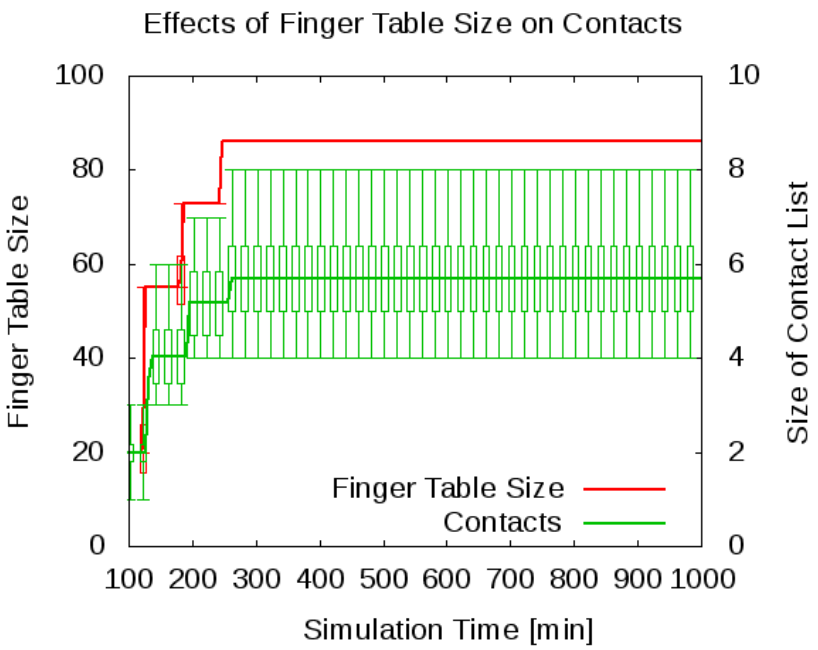
TECHNISCHE UNIVERSITÄT DARMSTADT

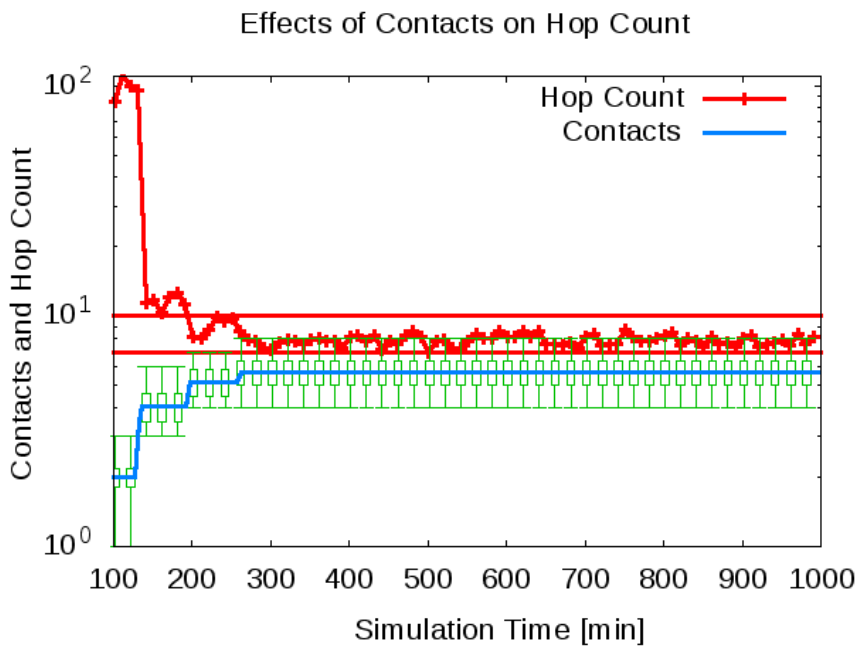
Contacts ~~ $\log(FT_{Size})$

Hop Count !~ $\log_{Contacts}(N)$

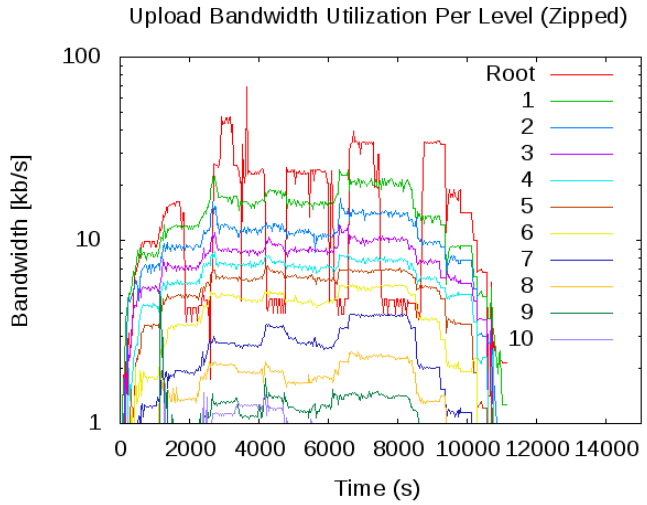
Approach:

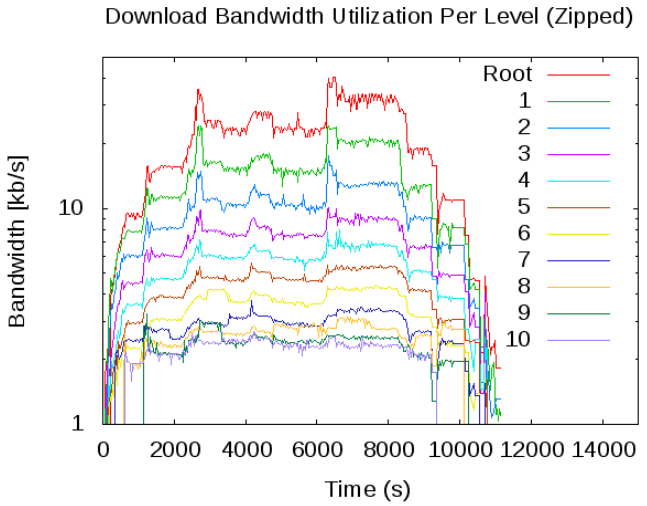
- Modify FT_{Size} → effect on Contacts
- Modify Contacts → effect on Hop Count

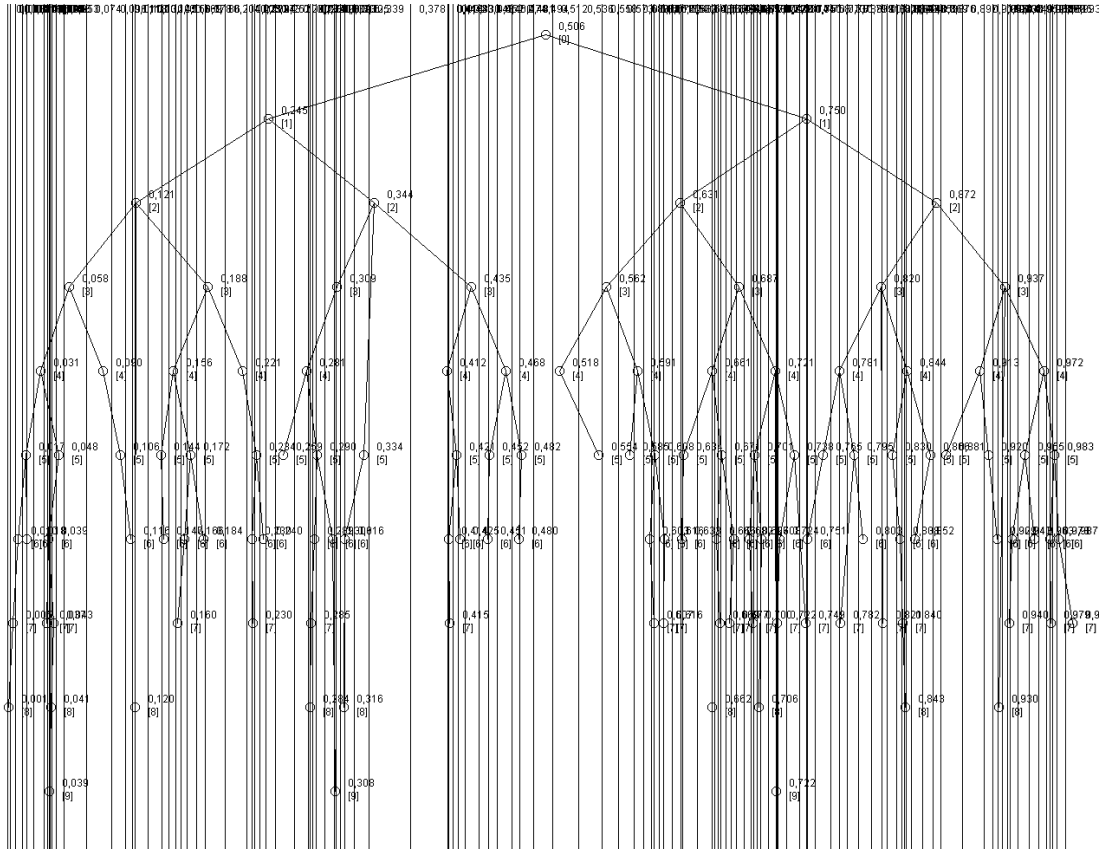


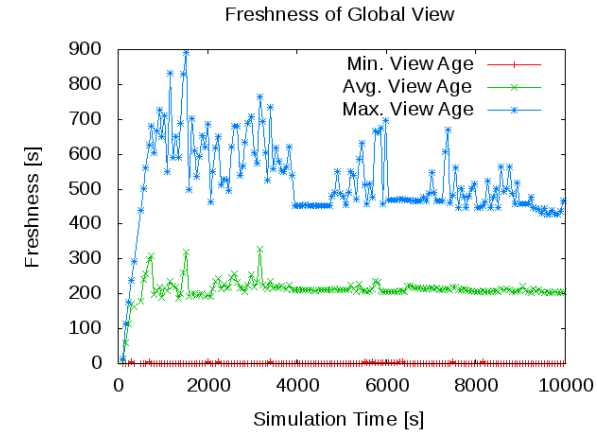


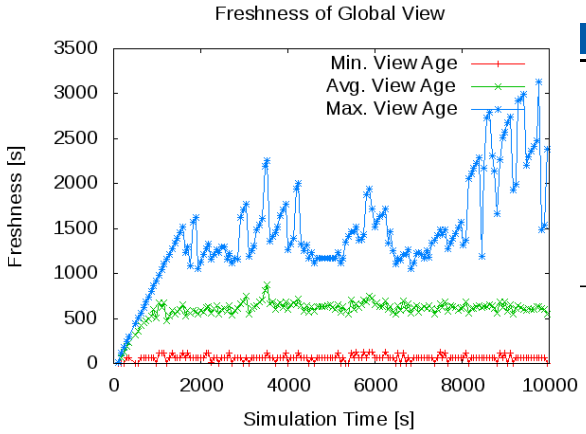
Selected Results from the Testbed Evaluation

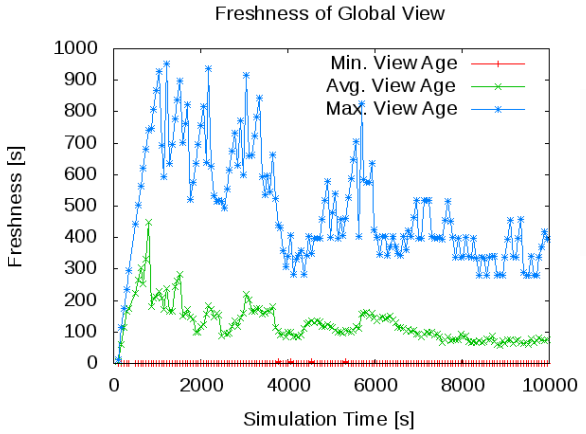


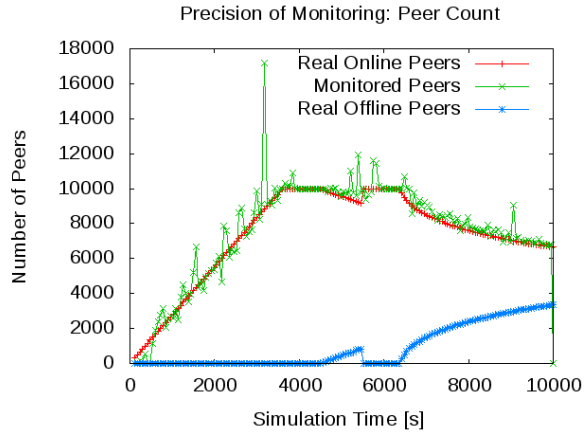


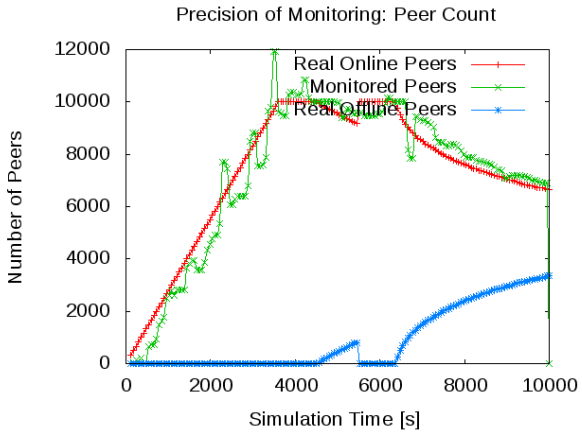


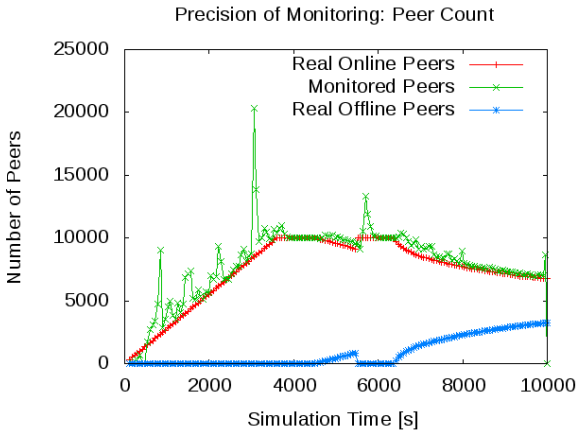

Freshness of Global View
Min. View Age
Avg. View Age
Max. View Age
Freshness [s]
Simulation Time [s]

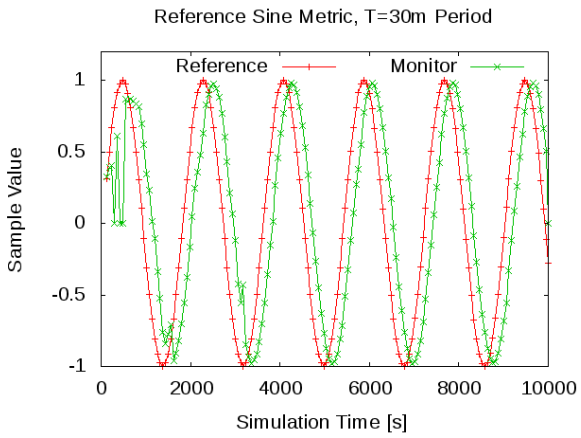

Freshness of Global View
Min. View Age
Avg. View Age
Max. View Age
Freshness [s]
Simulation Time [s]

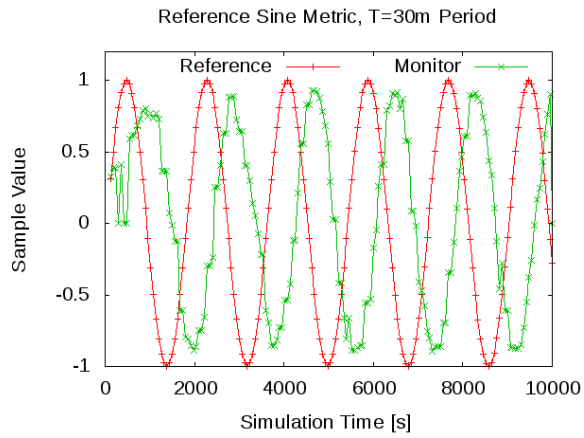

Freshness of Global View
Min. View Age
Avg. View Age
Max. View Age
Freshness [s]
Simulation Time [s]

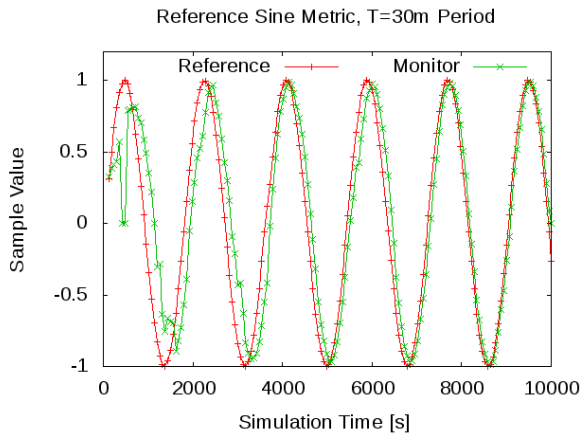

Precision of Monitoring: Peer Count
Real Online Peers
Monitored Peers
Real Offline Peers
Number of Peers
Simulation Time [s]


Precision of Monitoring: Peer Count
Real Online Peers
Monitored Peers
Real Offline Peers
Number of Peers
Simulation Time [s]

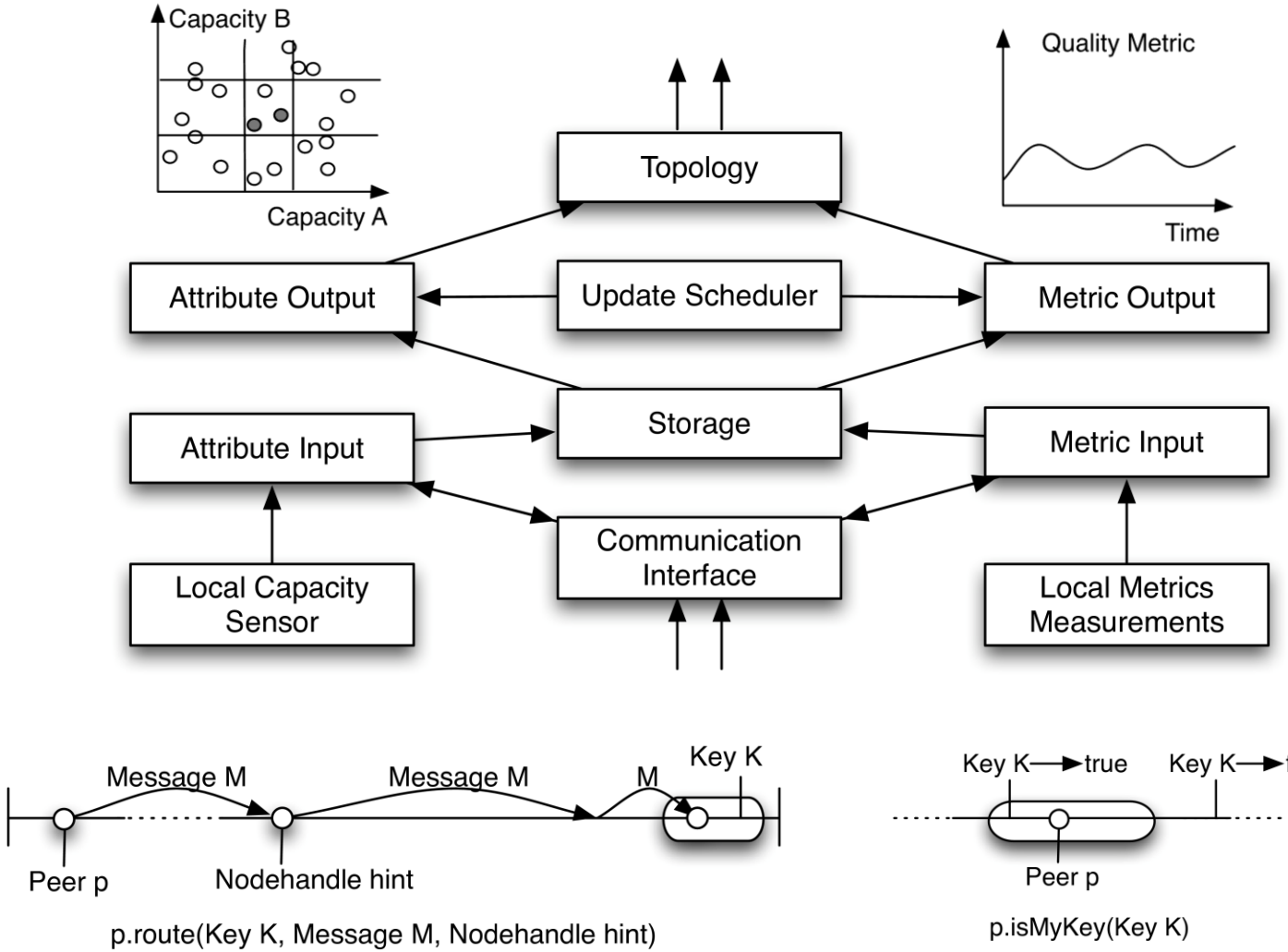

Precision of Monitoring: Peer Count
Real Online Peers
Monitored Peers
Real Offline Peers
Number of Peers
Simulation Time [s]


Reference Sine Metric, T=30m Period
Reference
Monitor
Sample Value
Simulation Time [s]


Reference Sine Metric, T=30m Period
Reference
Monitor
Sample Value
Simulation Time [s]


Reference Sine Metric, T=30m Period
Reference
Monitor
Sample Value
Simulation Time [s]

Modules in SkyEye.KOM



SkyEye.KOM – Variants

TECHNISCHE UNIVERSITÄT DARMSTADT

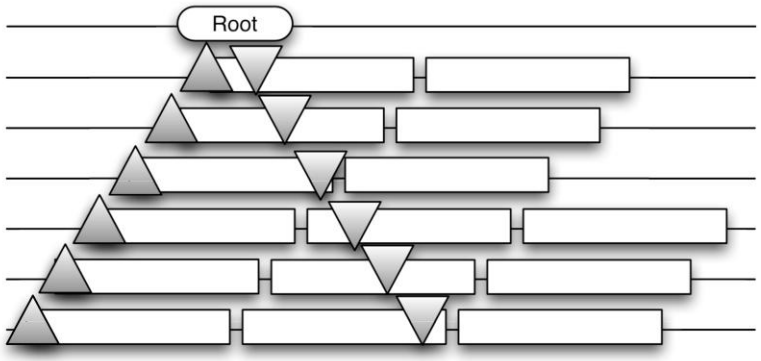
Variant: synchronized updates

Idea:

- Loosely synchronize updates messages
- Push results once global view is created
- Tradeoff: freshness vs. costs

Approach:

- Push with ACKs: synchronization delay
- Peers adapt their update offset
- Updates processed in a row
- Push global view with “special-ACK”



Variant: smoothing of results

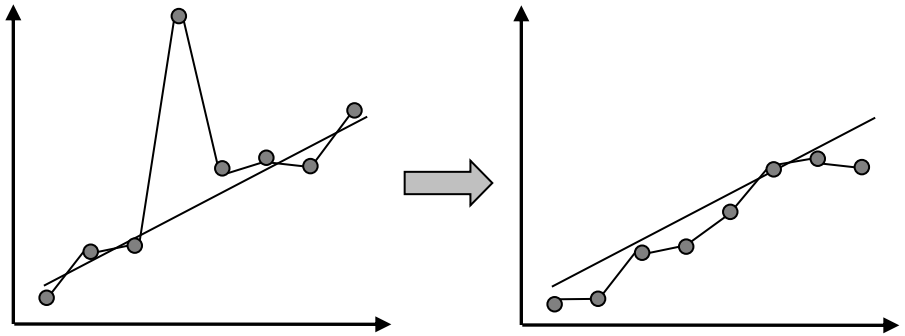
Idea:

- Eliminate monitoring outliers with history-based smoothing: median / exponential
- Tradeoff: freshness vs. less outliers

Approach:

- History of measurements: $H = \{m_1,\ldots,m_h\}$
- Final measure $s_H = \text{smooth}(H)$
- Median: $s_H = m_{(|H|+1)/2}$ with sorted m_i
- Exponential smoothing:

$$s_H = \alpha m_h + (1-\alpha)\, s_{H-1}$$



Quality Properties of SkyEye.KOM

TECHNISCHE UNIVERSITÄT DARMSTADT

Applicability

- Applicable on every (KBR-compatible) structured p2p overlay
- Independent of any application

→Unified ID space, using core DHT functions

Robustness

- Handling Churn, coping with link losses

→ If peer fails: automatically replaced in the DHT

→ Updates are routed to new peer for aggregation

Scalability

- Scaling in terms of participating peers and exchanged information

→Low costs per peer:
bound tree node degree (1+β)

→Costs independent of node's tree position, ~1Kb/update

Performance

- High precision, low outliers

→Information age limited by tree depth, $O(\log_\beta(N))$

→Influenced by update period

Retrievability

- Monitoring all peers in the network

→All peers are in the monitoring tree

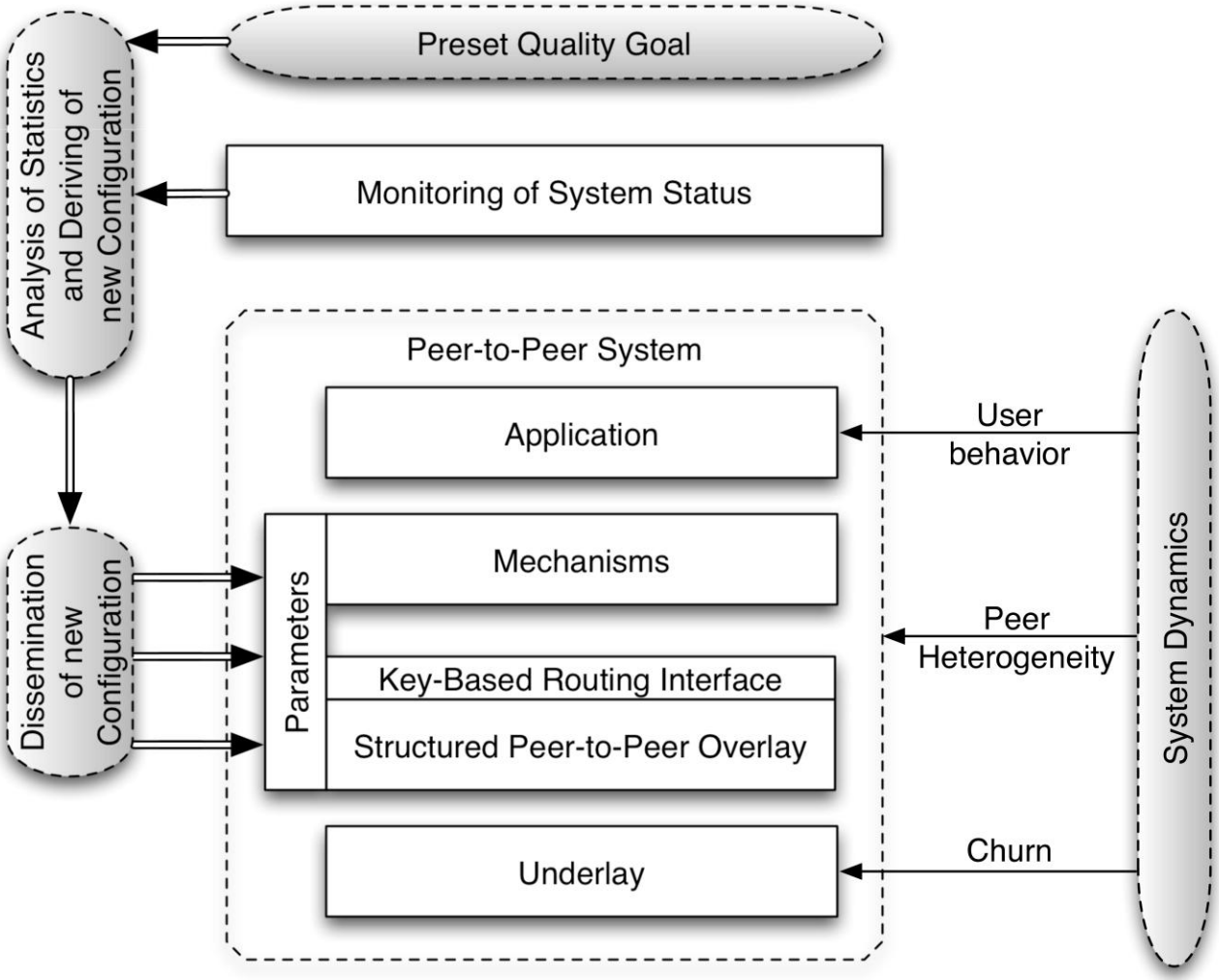
Efficiency

- Lightweight solution, low complexity: easier to use, more robust

→Just two message types: Update, ACK

→No signaling complexity and overhead

Additional Slides on Management



Comparison of C/S Solution with SkyEye.KOM

TECHNISCHE UNIVERSITÄT DARMSTADT

Category		Tree	Centralized	
			Server	Clients
Average Age		$(\sum_{i=1}^{N} E_i(N) \cdot i \cdot UI)/N$	UI	
Number of Messages per Peer	Updates sent	1	0	1
	Updates received	β	N	0
	ACKs sent	β	N	0
	ACKs received	1	0	1
	Total	$2 \cdot (\beta + 1)$	$2 \cdot N$	2
Traffic per peer	Total	$2 \cdot (\beta + 1) \cdot size_m$	$2 \cdot N \cdot size_m$	$2 \cdot size_m$
Example: $\beta = 8$, $N = 10^6$, $UI = 60s$, $size_m = 3kb$.				
Average age		6m47s	1m	
Number of messages per peer per $UI = 60s$		18	2,000,000	2
Traffic per $UI = 60s$		54kb/m	6,000,000kb/m	6kb/m
Traffic per second		0.9kb/s	100,000kb/s	0.1kb/s

Overview on Quantity of Monitored Statistics

TECHNISCHE UNIVERSITÄT DARMSTADT

A	S	P	Statistic - (A: Analytic Model, S: Simulation View, P: Prototype)	Unit
General Statistics - 5 values for all statistics: Count, Min, Max, Sum, Sum of Squares				
	X	X	Time	#
X	X	X	Number of peers	#
X	X	X	Online time of the peers	s
X	X		Information age at root, freshness	s
X	X	X	Peer level	#
Lookup Duration and Hops - 5 values each				
X	X		Delay in the overlay	s
X	X	X	Number of hops in the overlay	#
X	X		Number of forwarded messages	1/s
Overhead per Message Type - 20 values each: sent and received messages and traffic				
X	X	X	Complete overhead	1/s,KB/s
X	X	X	Complete overhead per level	1/s,KB/s
X	X	X	SkyEye.KOM overhead	1/s,KB/s
	X		Overlay overhead	1/s,KB/s
	X		Join and leave overhead	1/s,KB/s
	X		Lookup overhead	1/s,KB/s
General Overhead - 10 values each (global) and for each level: up- and download				
	X	X	Ratio of bandwidth used for unzipped data	%
		X	Ratio of bandwidth used for zipped data	%
Model-specific - 1 value each				
X			Number of Sub-Coordinators, free places, leafs, non-leafs	#
X			Probability to join a level	%
Simulation-specific - 15 values				
	X		Lookup operations: completed, succeeded, failed	1/s
Prototype-specific - 5 values each, 15 values for the memory-related metrics				
		X	CPU utilization	%
		X	Disk space	%
		X	Heap memory: maximum, initial, used	%
		X	Non heap memory: maximum, initial, used	%

A	S	P	Statistic - (A: Analytic Model, S: Simulation View, P: Prototype)	Unit
Reference Signal - 20 values each: T=1m,3m,10m,30m				
	X		Monitored ZigZag Signal	#
	X		Monitored Sine Signal	#
Overhead per Message Type - 20 values each: sent and received messages and traffic				
X	X	X	Metric Update overhead	1/s,KB/s
X	X	X	Metric Update ACK overhead	1/s,KB/s

A	S	P	Statistic - (A: Analytic Model, S: Simulation View, P: Prototype)	Unit
General Statistics - 5 values for all statistics: Count, Min, Max, Sum, Sum of Squares				
X	X	X	Number of Support Peers	#
X	X	X	Number of stored attribute entries in tree	#
X	X	X	Number of stored attribute entries at root	#
X	X	X	Number of stored attribute entries per level	#
X	X	X	Peer quality of Coordinators	#
X	X	X	Peer quality of Support Peers	#
Capacity-based Peer Search - 5 values each				
X	X		Delay for resolving queries	s
X	X	X	Number of hops for resolving	#
X	X	X	Number of initiated queries	1/s
X	X	X	Level of initiated queries	#
X	X	X	Level of solved queries	#
Overhead per Message Type - 20 values each: sent and received messages and traffic				
X	X	X	Attribute update overhead	1/s,KB/s
X	X	X	Query overhead	1/s,KB/s

Total metric quantity:

- Simulation: 370+10•Depth
- Prototype: 255+20•Depth

Comparison of Benchmark Signals

TECHNISCHE UNIVERSITÄT DARMSTADT

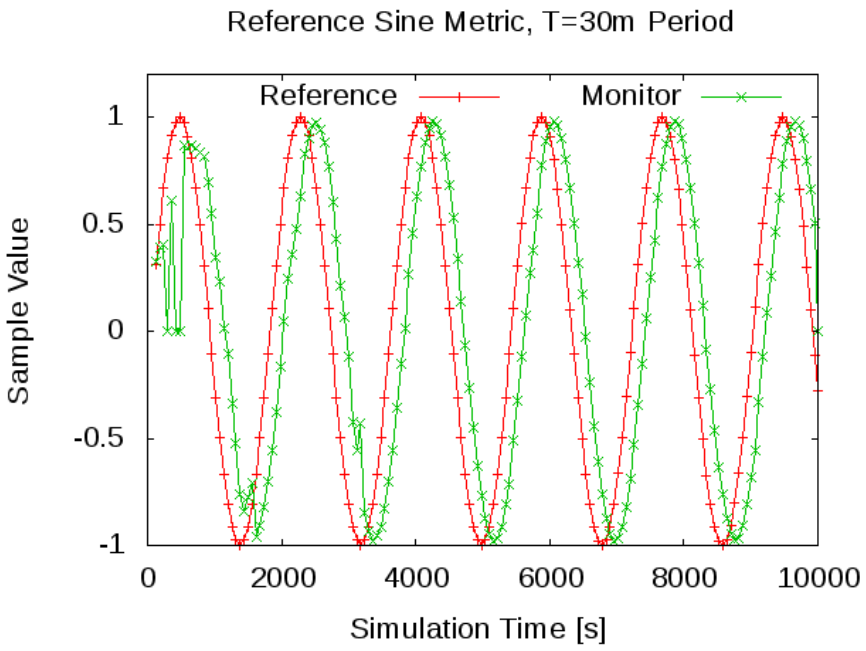
Benchmarking signals

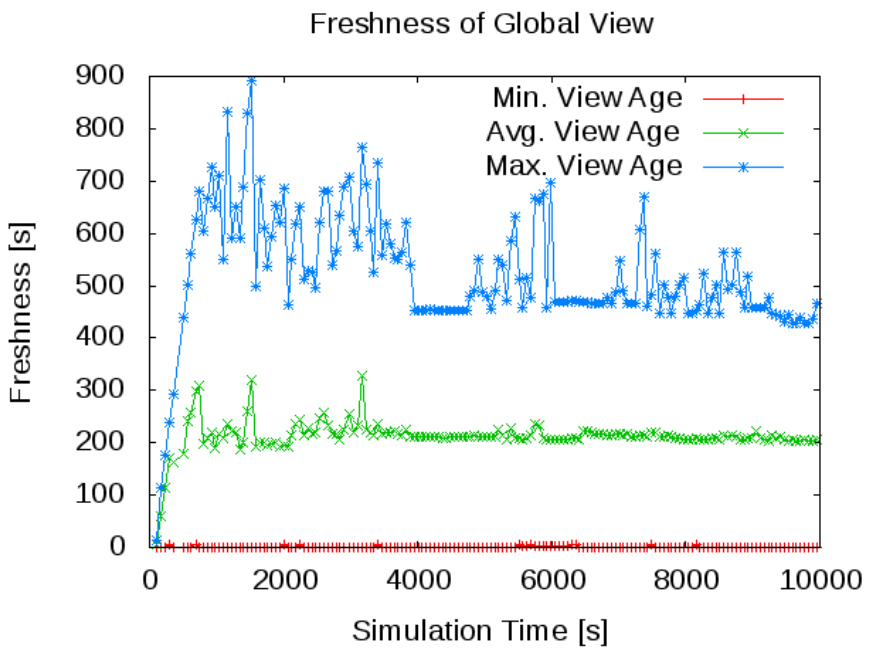
- Sine and ZigZag, periodicity: 1800s=30m
- Synchronized injection of signals
- Allows to benchmark monitoring precision
- Age of information: around 200s

Freshness

= Average age of aggregated information
= ½ of time to gather and disseminate
= ½ $\log_\beta$ (N) · UI
= ½ $\log_4$(10000) · 60s = ½ · 398s = 199s

- Model and simulations match





Parameter Studies: Identification of Tradeoffs

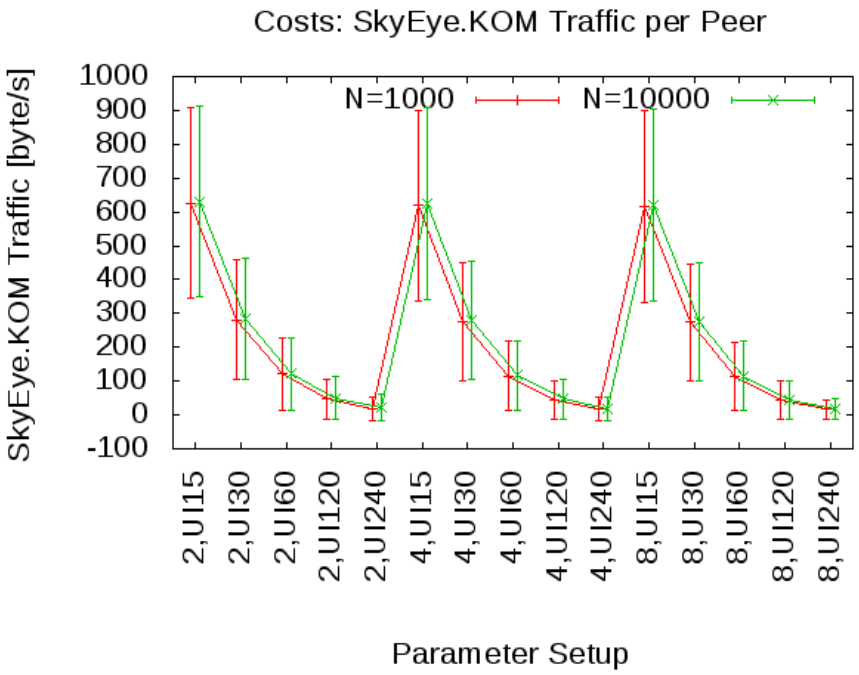
TECHNISCHE UNIVERSITÄT DARMSTADT

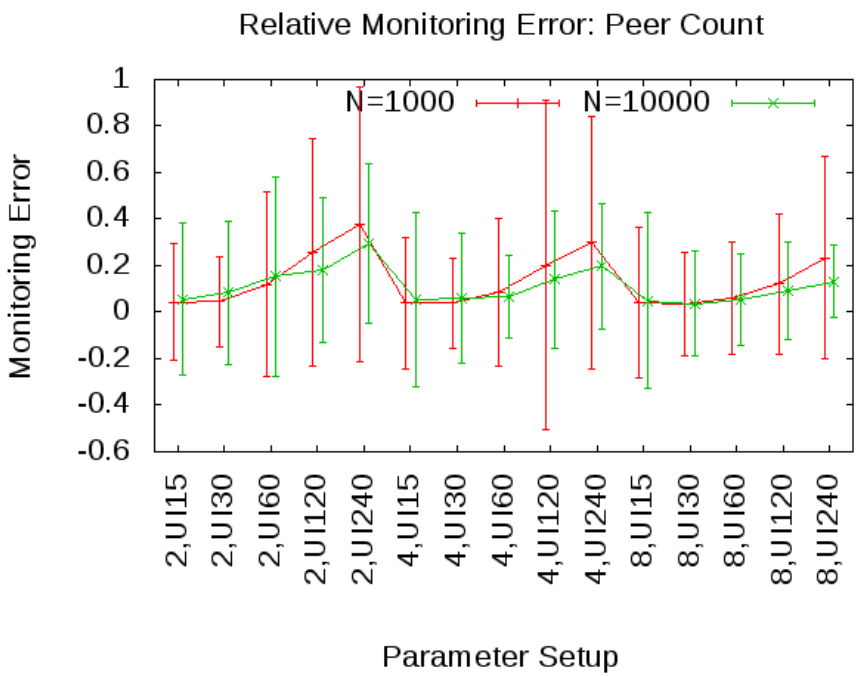
Varied parameters

- Update interval UI: 15, 30, 60, 120, 240s
- Branching factor β = 2, 4, 8
- Number of peers N = 1000, 10000
- (Mean) costs depend on UI (not β or N)

Tradeoff

- Monitoring precision (of sums) vs. costs
  - Monitoring error depends on UI, β, N
  - Low relative error requires more traffic
- Increase effort until precision is sufficient





Management Goals

TECHNISCHE UNIVERSITÄT DARMSTADT

Service quality of the p2p system

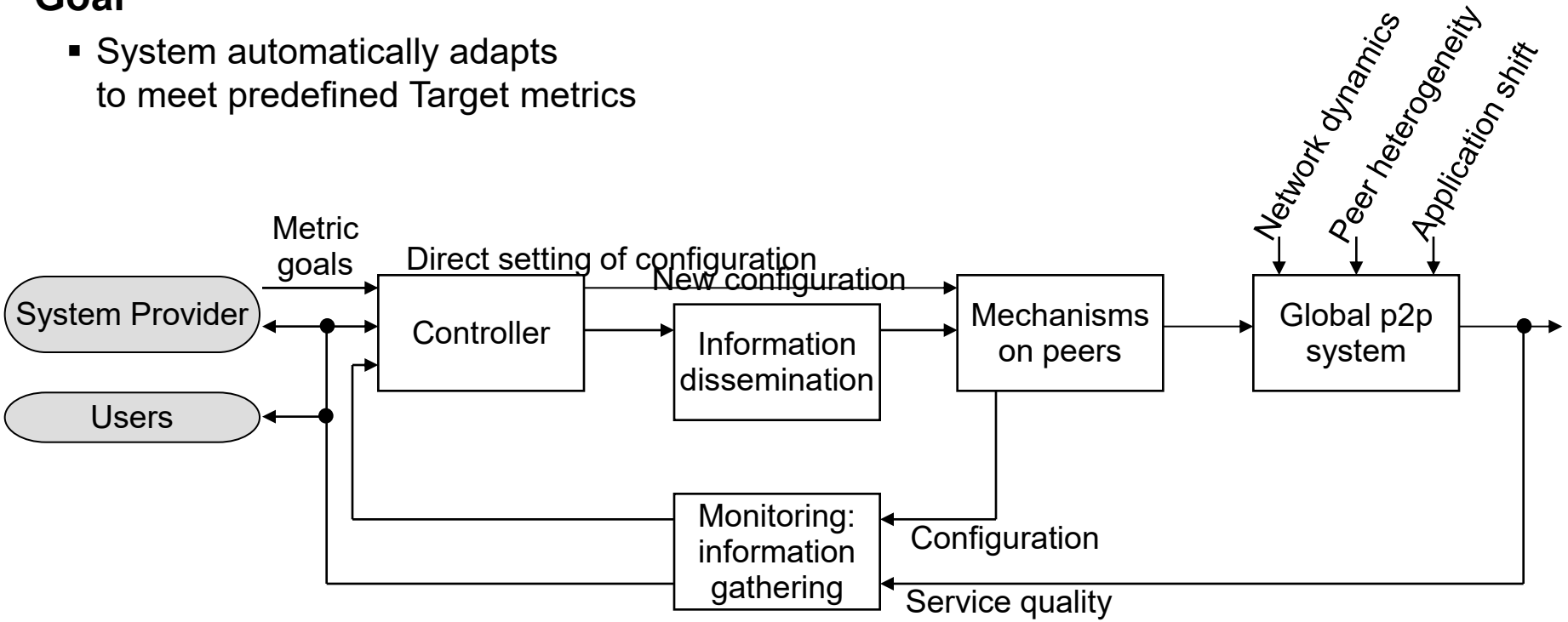
- Application-specific metric intervals
- *Response time [100ms,400ms]*

Goal

- System automatically adapts to meet predefined Target metrics

Main idea

- Research on viable approaches
- Autonomic computing for p2p systems



$$K_p^l := \frac{\frac{\lfloor ID_p \cdot \beta^l \rfloor + 1}{\beta^l} + \frac{\lfloor ID_p \cdot \beta^l \rfloor}{\beta^l}}{2}$$

$$level_p^{min} := min(i \in \mathbb{N} \text{ with } K_p^i \in scope(p))$$

$$E_i(N) = E_i(N-1) + \frac{\beta \cdot E_{i-1}(N-1) - E_i(N-1)}{\beta^i}$$

$$Age(i, UI) = i \cdot UI$$

$$AvAge(E_i(N), i, UI) = \frac{\sum_{i=1}^{N} E_i(N) \cdot Age(i, UI)}{N}$$