Socially enhanced Services Computing

Novel models and algorithms for distributed systems

Schahram Dustdar

Distributed Systems Group

Institute of Information Systems

TU Wien

Joint work with:

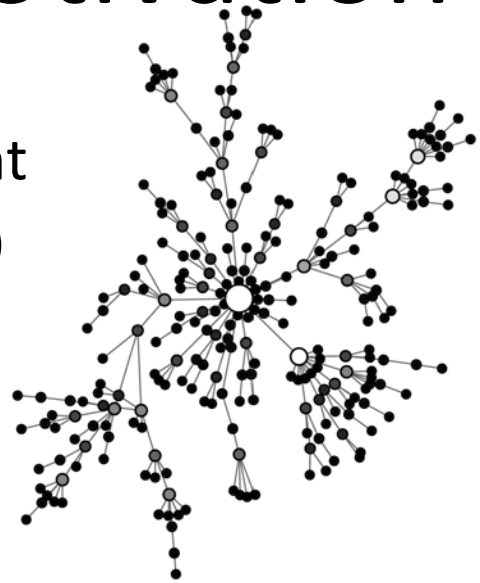
Daniel Schall, Florian Skopik, Harald Psaier, Lukasz Juszczyk, Linh Truong

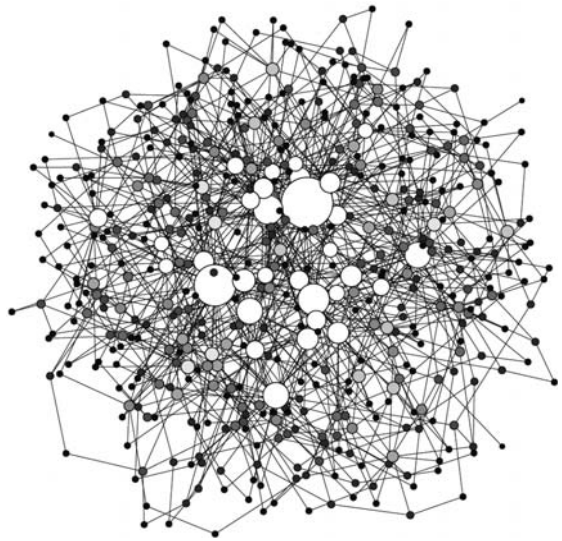
VITALAB Vienna Internet Technologies Advanced Research Lab

Distributed Systems Group

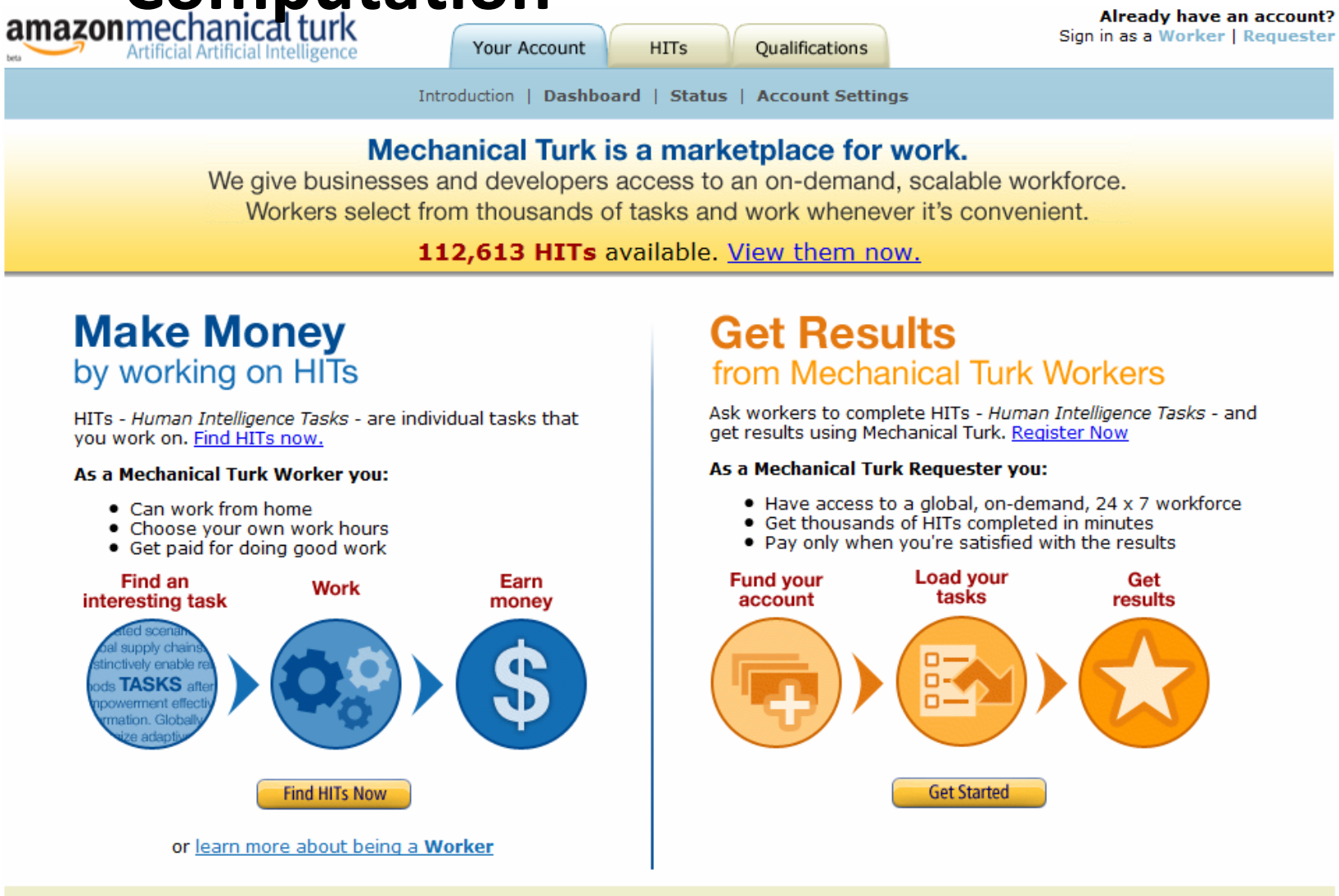
Environment and Motivation

- **Open** and dynamic Internet-based environment
  - Humans **and** software resources (e.g., Web services)
  - **Joining/leaving** the environment **dynamically**
  - Humans perform **activities**
- Massive **collaboration** in SOA/Web 2.0
  - Large sets of **humans** and software **resources**
  - Dynamic **compositions**
  - Distributed communication and coordination
- Understanding the **dynamics**
  - Future interactions
  - Resource selection
  - Compositions & Adaptation of actors
  - Disclosure of information

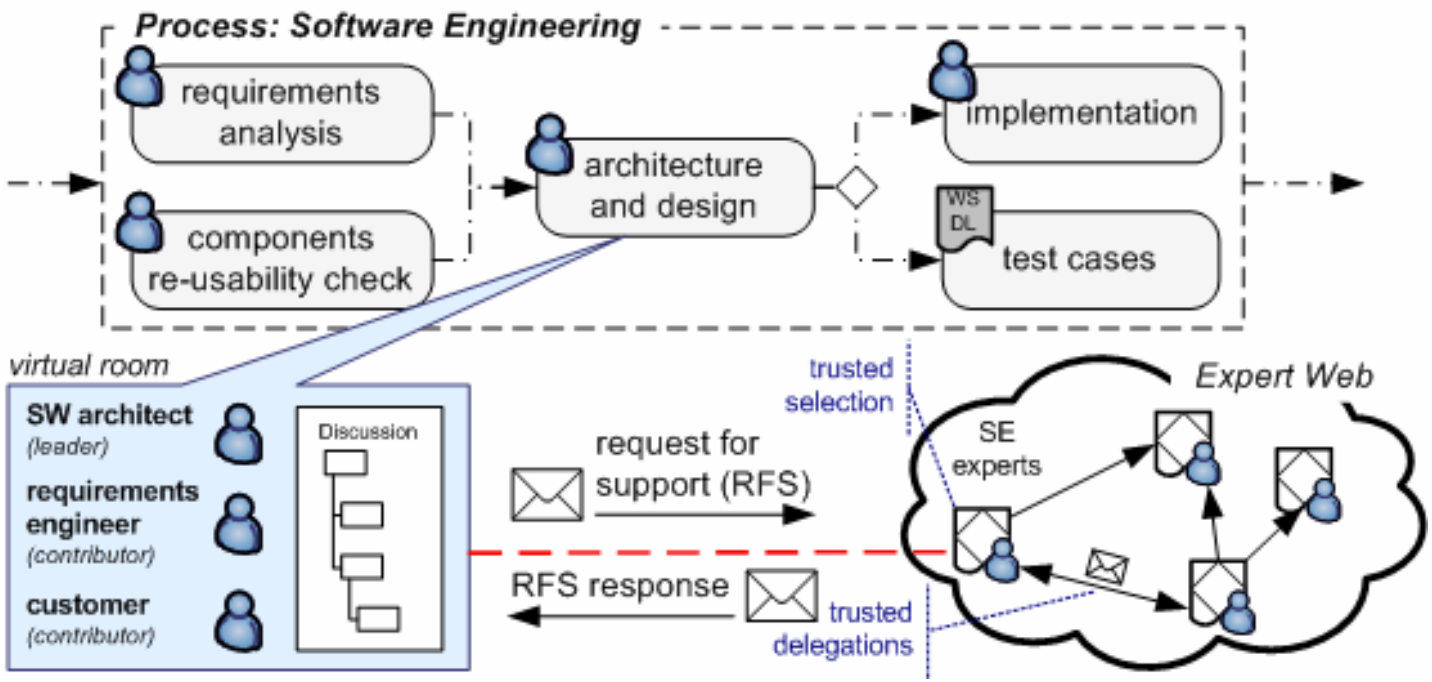




Crowdsourcing & Human Computation



Motivating Scenario



Q1: How do actor discovery and selection mechanisms work?

Q2: How can actors be flexibly involved (ranked)?

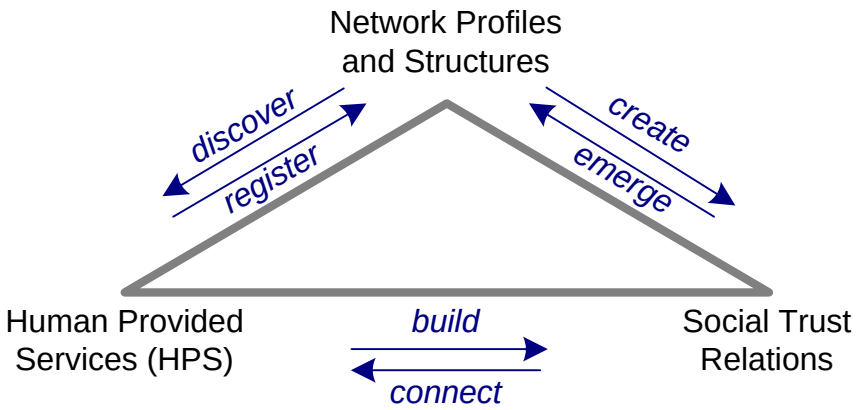
Q3: How can interactions and service compositions become adaptive?

Skopik, F., Schall, D., Dustdar, S. *Trusted Interaction Patterns in Large-scale Enterprise Service Networks.* 18th International Conference on Parallel, Distributed, and Network-Based Computing. Pisa, Italy, 2010. IEEE.

General Principles

- Interface
- Protocols
- Composition
- Behavior dynamics
- Overlay network
- Monitoring & Metrics

Socially enhanced Services Computing

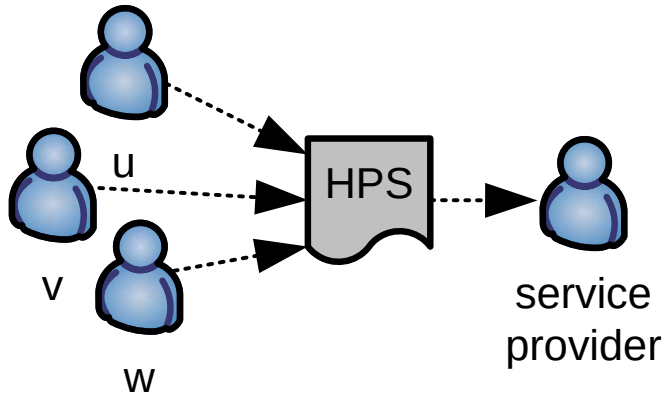


Mixed Systems with the human in the loop

- Traditional perspective on SOA not sufficient anymore
- Considering social influences and relations
  - Humans provide services (HPSs)
  - HPSs build social relations (Trust)
  - Emerging network structures and communities
  - Services are discovered based on partner recommendations

Human-Provided Services (HPS)

- User contributions modeled as services
  - Users define their own services
  - Reflect willingness to contribute
- Technical realization
  - Service description with WSDL (capabilities)
  - Communication via SOAP messages
- Example: Document Review Service
  - Input: document, deadline, constraints
  - Output: review comments



Schall, D., Dustdar, S., Blake, B.M. A Programming Paradigm for Integrating Human-Provided and Software-Based Web Services
IEEE Computer, July 2010

Schall, D., Truong, H.-L., Dustdar, S. *The Human-Provided Services Framework*. IEEE 2008 Conference on Enterprise Computing, E-Commerce and E-Services (EEE), Crystal City, Washington, D.C., USA, 2008. IEEE.

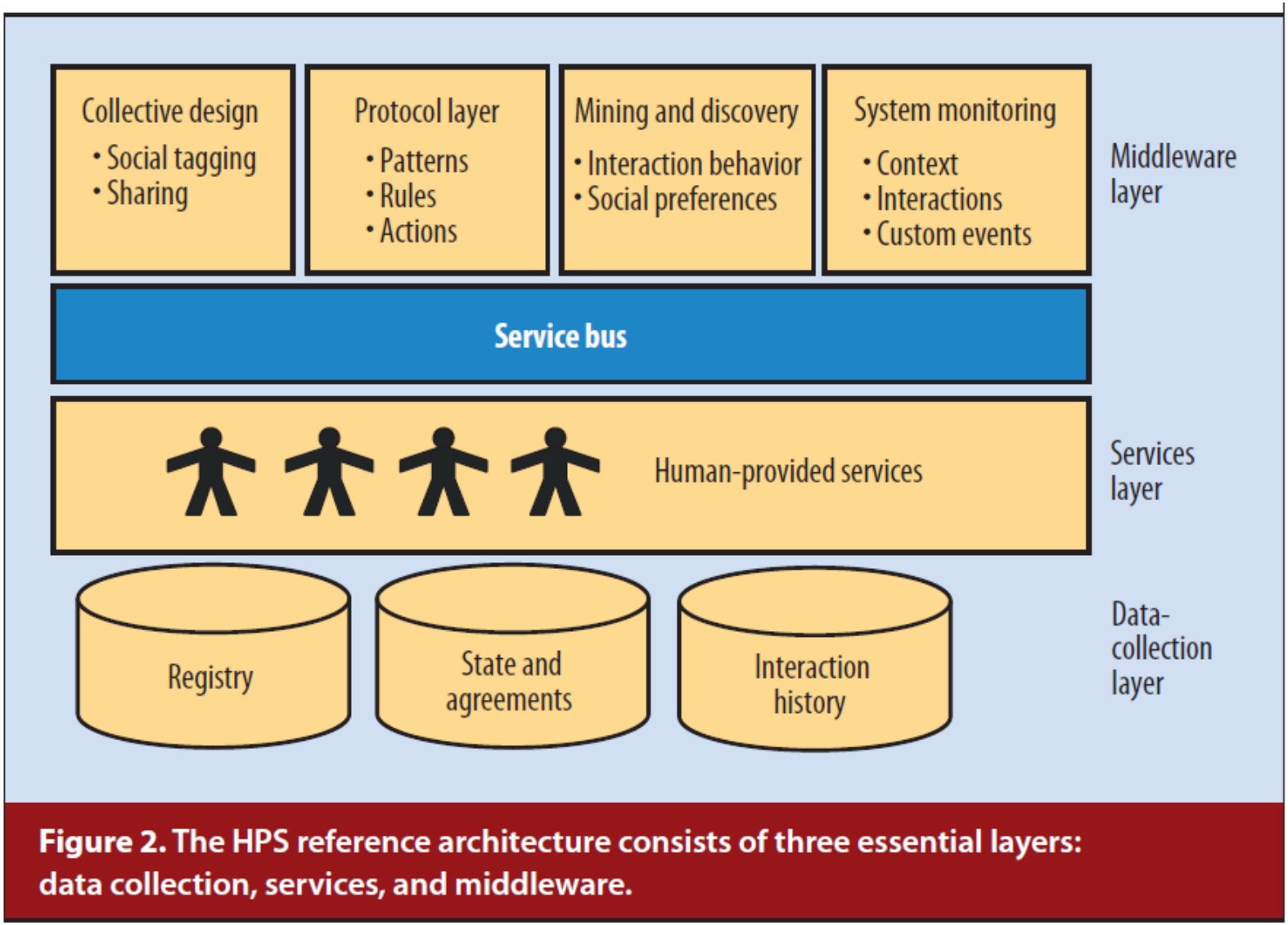
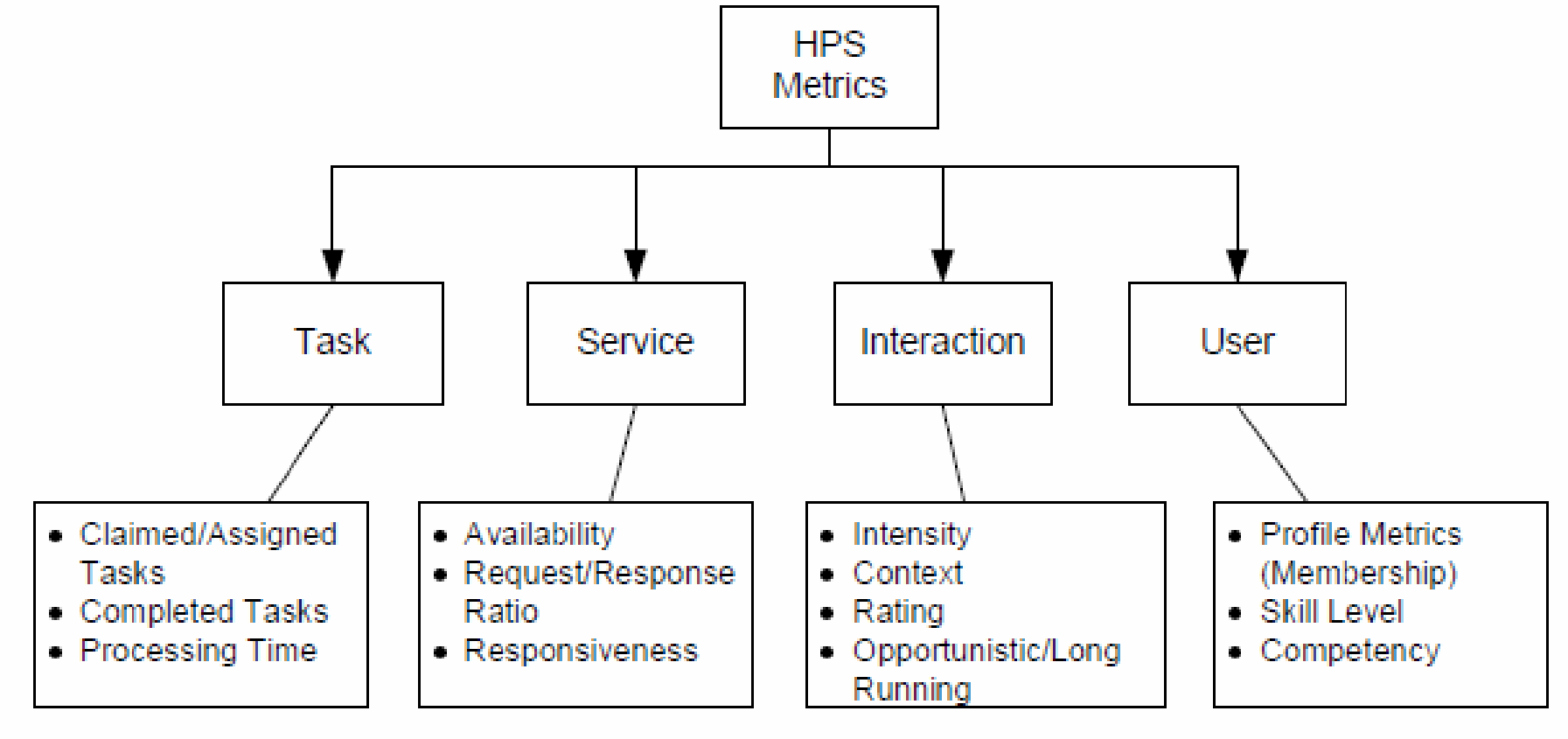


Figure 2. The HPS reference architecture consists of three essential layers: data collection, services, and middleware.

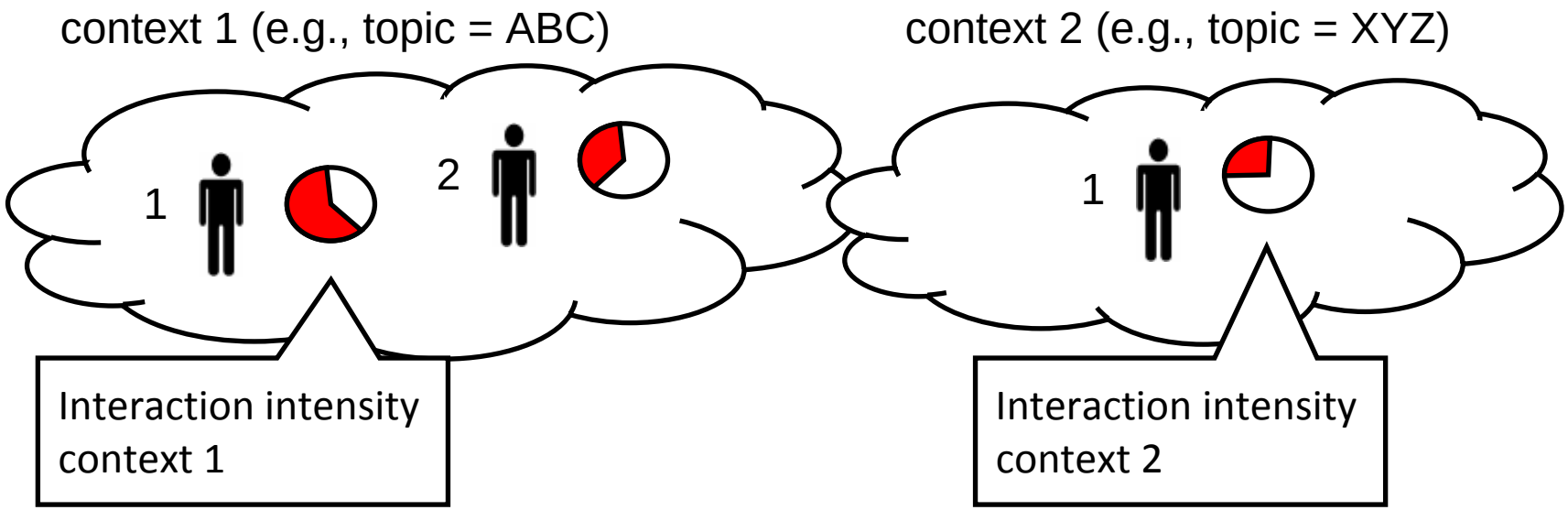
Overview Metrics



Metrics: ranking and selection of services

Ranking Algorithm:
Interaction context

- Users interact in different contexts with different intensities

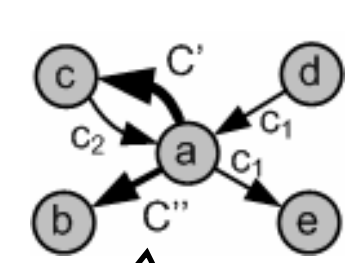


- Personalize ranking (i.e., expertise) for different contexts

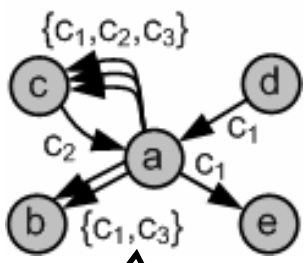
Schall D., Dustdar S. (2010) Dynamic Context-Sensitive PageRank for Expertise Mining, 2nd International Conference on Social Informatics (SocInfo'10), 27-29 October, 2010, Austria. Springer.

Ranking Algorithm:
Context-aware DSARank (Dynamic Skill Activity)

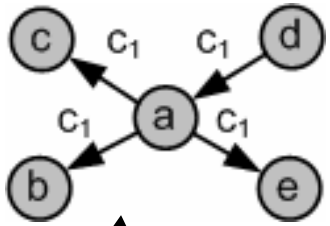
Approach: Expertise mining in weighted subgraph



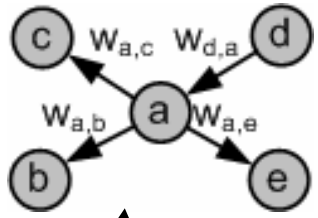
"Tags" identify the interaction context.



Each context tag may have different weights (e.g., frequency).



For a given context (e.g., c1) create a subgraph.



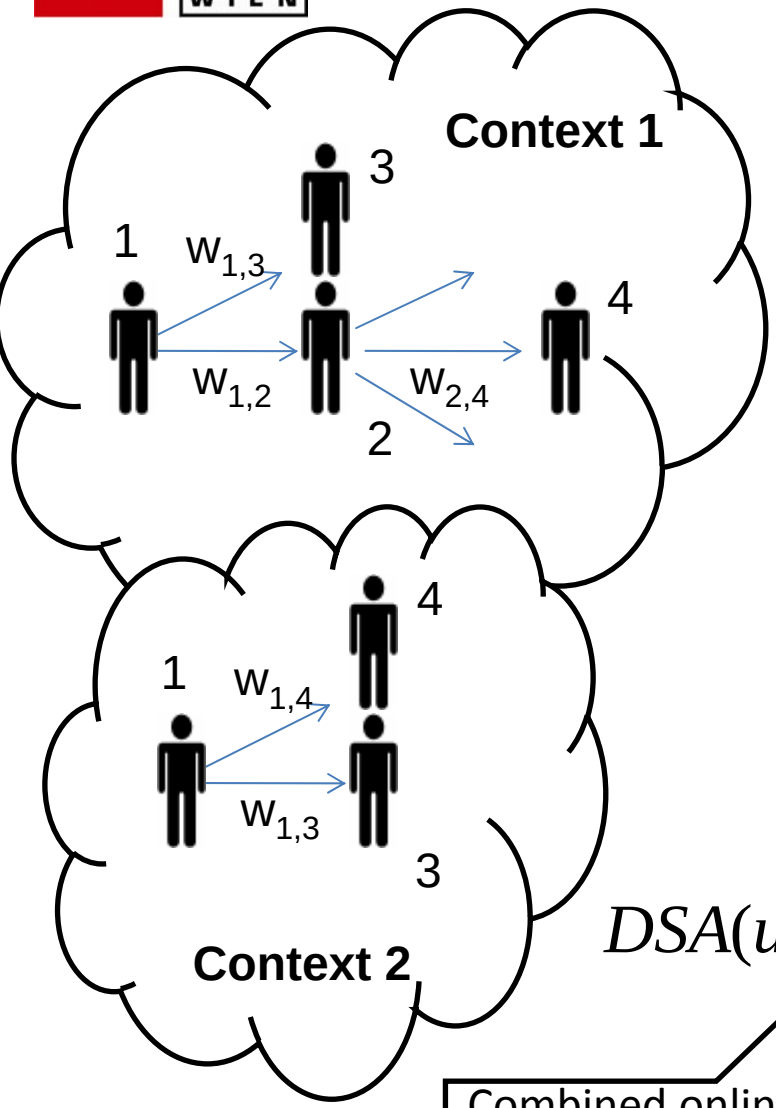
Perform ranking based on weighted links in subgraph.

- Linearity Theorem (Haveliwala 02):

$$w_1 PR(p_1) + w_2 PR(p_2) = PR(w_1 p_1 + w_2 p_2)$$

Schall D., Dustdar S. (2010) Dynamic Context-Sensitive PageRank for Expertise Mining, 2nd International Conference on Social Informatics (SocInfo'10), 27-29 October, 2010, Austria. Springer.

Context-dependent DSARank



- (1) Identify context of interactions ("tags")
- (2) Select relevant links and people
- (3) Create weighted subgraph (for context)
- (4) Perform mining


User 1's expertise in context 1


User 1's expertise in context 2

$$DSA(u;C') = \sum_{c \in C'} w_c \underbrace{DSA\left(w_1 p_1(u) + \ldots + w_n p_n(u)\right)}$$

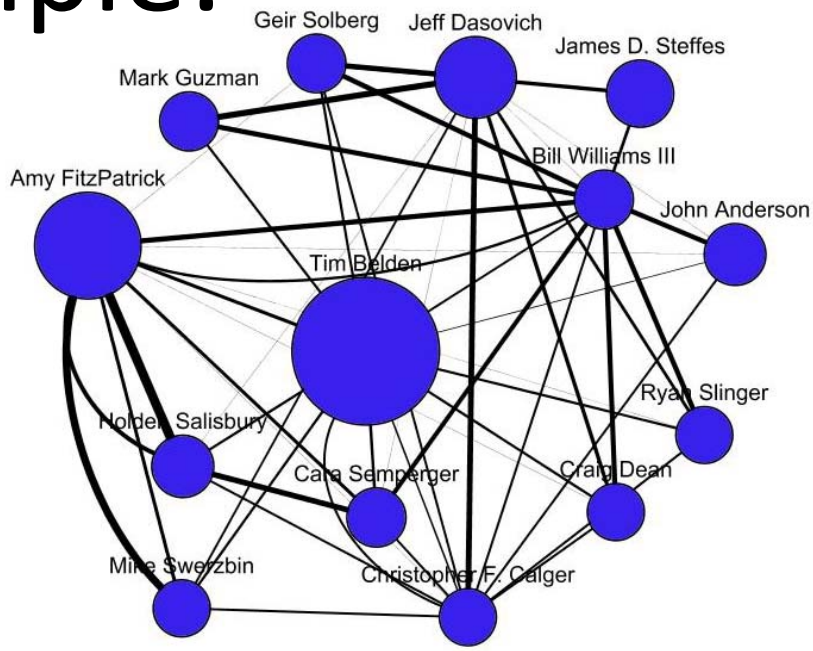
Combined online based on preferences

Calculated offline

E.g., p(u) = w1 IIL(u) + w2 availability(u)

Ranking Example:
Interaction Mining

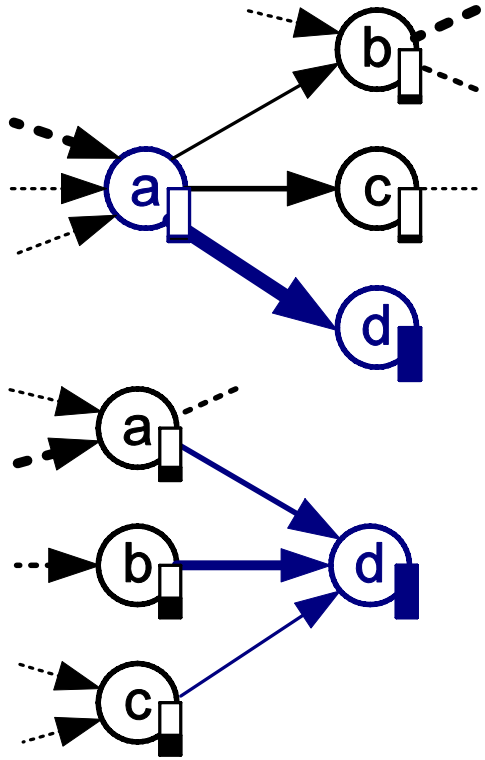
- Email Interaction Graph
- High interaction intensity influences importance rankings
- High interaction intensity reveals key people



ID	Rank (DSA)	Rank (PR)	Intensity Level
37	1	21	7.31
...			
253	4	170	2.07
347	5	282	1.39

Delegation Factory/Sink

- Factory
  - *a* accepts and delegates tasks frequently
  - *a* processes few tasks and has a low task-queue
- Sink
  - *d* accepts too many tasks
  - *d* processes slow (capability vs. overload)
- Misbehavior impact
  - Produces unusual amounts of task delegations
  - Tasks miss their deadline
  - Leads to performance degradations of the entire network



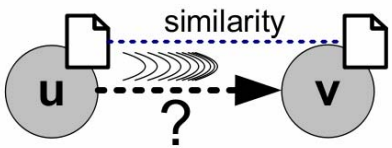
Psaier H., Juszczyk L., Skopik F., Schall D., Dustdar S. Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems, 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'10), 27 Sept.-01 Oct. 2010, Budapest, Hungary.

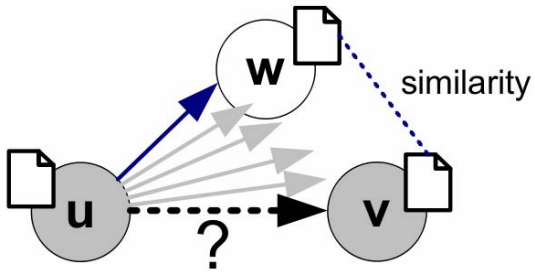
(Mis)behavior monitoring

- Open System with varying participation
- All services use the communication infrastructure
- Interaction logging:
  - Log the exchanged messages and process their content
- Logs provide information on:
  - Task properties: id, tags, etc.
  - Type, skills, and interests of services

Similarity Service

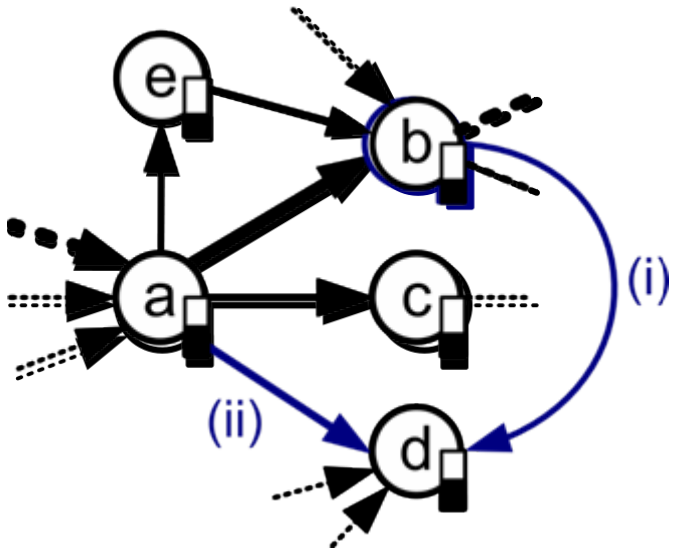
- Cos-similarity to determine the similarity of two services' profile vectors: $sim_{profile}(\mathbf{p_u}, \mathbf{p_v}) = \cos(\mathbf{p_u}, \mathbf{p_v})$
- **Trust mirroring:** "similar minded" nodes tend to trust each other more than random nodes
- **Trust teleportation:** the past trust relation (u,w) "teleports" to others having similar interests.
  - Note: u and w have different profile, e.g., different roles





Misbehavior adaptation

initial state
-> *b* queue overload detected
-> find alternative/similar service
-> *(i)* 1st support *b* **mirroring of trust**
-> *(ii)* 2nd avoid *b* **teleportation of trust**

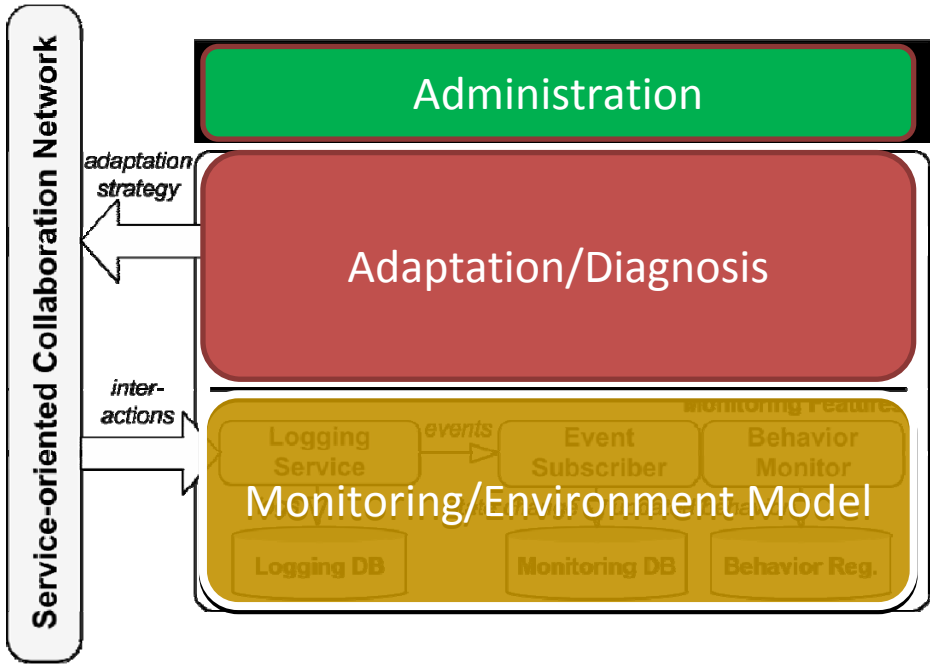


Self-adaptation concepts

- feedback loop design for misbehavior healing
- MAPE loop of autonomic computing:
  - **monitor** interactions and queue threshold
  - **analyze** behavior and compare to misbehavior models
  - update **behavior registry** (part of **knowledge**)
  - **plan** adaptive actions
  - **execute** channel regulations and redirections

VieCure framework

- Interaction logging updates monitoring db and behavior registry.
- Policy Store and Similarity Service determine the adaptations
- Admin tools allow to fine-tune the framework



Conclusions

- Mixed Dynamic Systems require novel "programming model" composing HPS and SBS
- Identification of (mis)behavior patterns and protocols and composition primitives in Mixed Systems
- Non-intrusive adaptation of misbehavior with self-healing

Thanks for your attention

1. **Trust-based Discovery and Interactions in Mixed Service-Oriented Systems**
   Schall D., Skopik F., Dustdar S. IEEE Transactions on Services Computing (TSC), Volume 3, Issue 3, pp. 193-205

2. **Modeling and Mining of Dynamic Trust in Complex Service-oriented Systems**
   Skopik F., Schall D., Dustdar S. Information Systems Journal (IS), Volume 35, Issue 7, November 2010, pp. 735-757. Elsevier.

3. **Programming Human and Software-Based Web Services**
   Schall D., Dustdar S., Blake M.B. IEEE Computer, vol. 43, no. 7, pp. 82-85, July 2010.

4. **Unifying Human and Software Services in Web-Scale Collaborations**
   Schall D., Truong H.-L., Dustdar S.
   IEEE Internet Computing, vol. 12, no. 3, pp. 62-68, May/Jun, 2008.

5. **Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems**
   Psaier H., Juszczyk L., Skopik F., Schall D., Dustdar S. 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'10), 27 Sept.-01 Oct. 2010, Budapest, Hungary.