A RELOAD Usage for Distributed Conference Control

Alexander Knauf
alexander.knauf@haw-hamburg.de

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Outline

for a RELOAD Usage for Distributed Conference Control

1. Problem Statement
2. Objectives
3. Starting Situation
4. Architecture & Technologies
5. Risks & Proposed Result
6. Conclusion & Outlook

Problem Statements

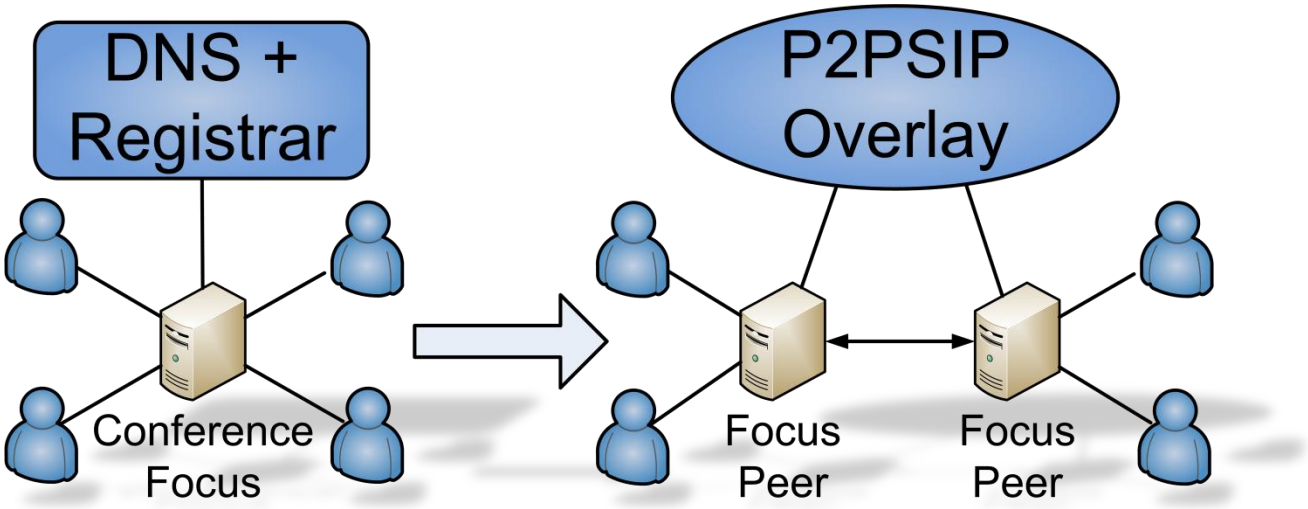
for a P2P Conferencing Approach

- Tightly coupled conferences are managed by a *single* entity called *Focus*:
  - Maintains signaling and media parameter negotiation
  - May perform media mixing functions
- **Problem (1)**: The Conference URI
  - *Identifies* the multiparty session, but
  - *locates* the conference focus
    - Single point of failure
- **Problem (2):** No dedicated server architecture in P2PSIP
  - Media mixing performed at the end-user devices
    - Scaling problem within large conferences
  - Conference must be registered and globally accessible
    - Demands a registrar, e.g., available through DNS

Objectives (1)

towards a Distributed Conference Control

- *Separate* the logical conference ID from the locator of the controller
  - Allows *multiple* focus peers to appear as a single controller
  - Increases robustness against focus failures
- *Replace* Registrars/DNS and mixer servers by a P2PSIP overlay
  - Independent of dedicated hardware by using RELOAD



Objectives (2)

towards a Distributed Conference Control

- **Proximity-aware Focus selection**
  - Peers determine their relative position
  - Choose closest Focus Peer
  - Reduces Delay and Jitter

ID/Locator Split

for a Transparent distribution of the conference focus

- Participants in role of *focus peers* maintain signaling and/or media connections to a subset of conference members
- Signaling messages sent from *several* focus peers appear as originating from *one* 'virtual' conference focus
  - Routing decision based on an additional *Record-Route* header pointing to the responsible focus peer

```
INVITE sip:bob@dht.example.com SIP/2.0
Call-ID: 0815@141.22.26.55
CSeq: 1 INVITE
From: <sip:conference@dht.example.com>;tag=134652
To: <sip:bob@dht.example.com>;tag=643684
...
Contact: <sip:conference@dht.example.com>;isfocus
Record-Route: <sip:alice@dht.example.com>
...
```

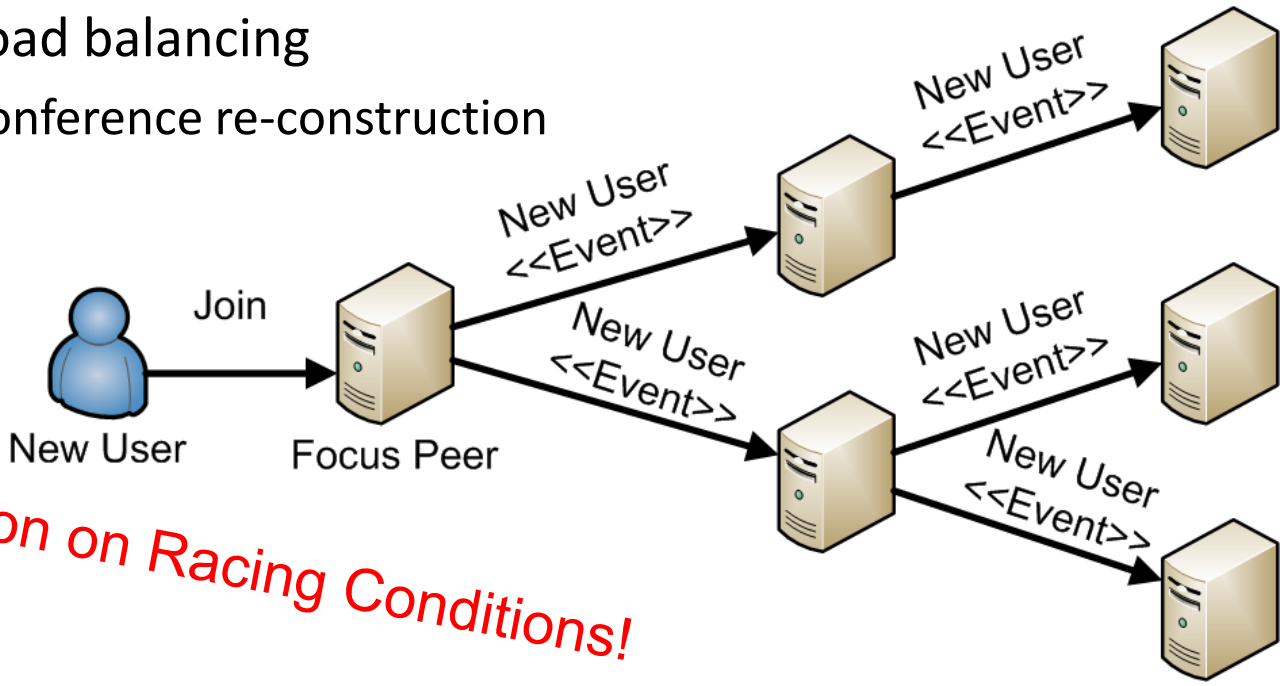
Here: Alice is Bob's responsible focus

- Alice receives message through the *Record-Route* and – as responsible focus peer - intercepts message from Bob

Synchronizing the Global State

to Provide a Coherent Knowledge

- Focus Peers maintain SIP Dialogs independently
- Change events must be advertised (via XML document)
- Global knowledge used for:
  - User information (e.g., Buddy List)
  - Load balancing
  - Conference re-construction

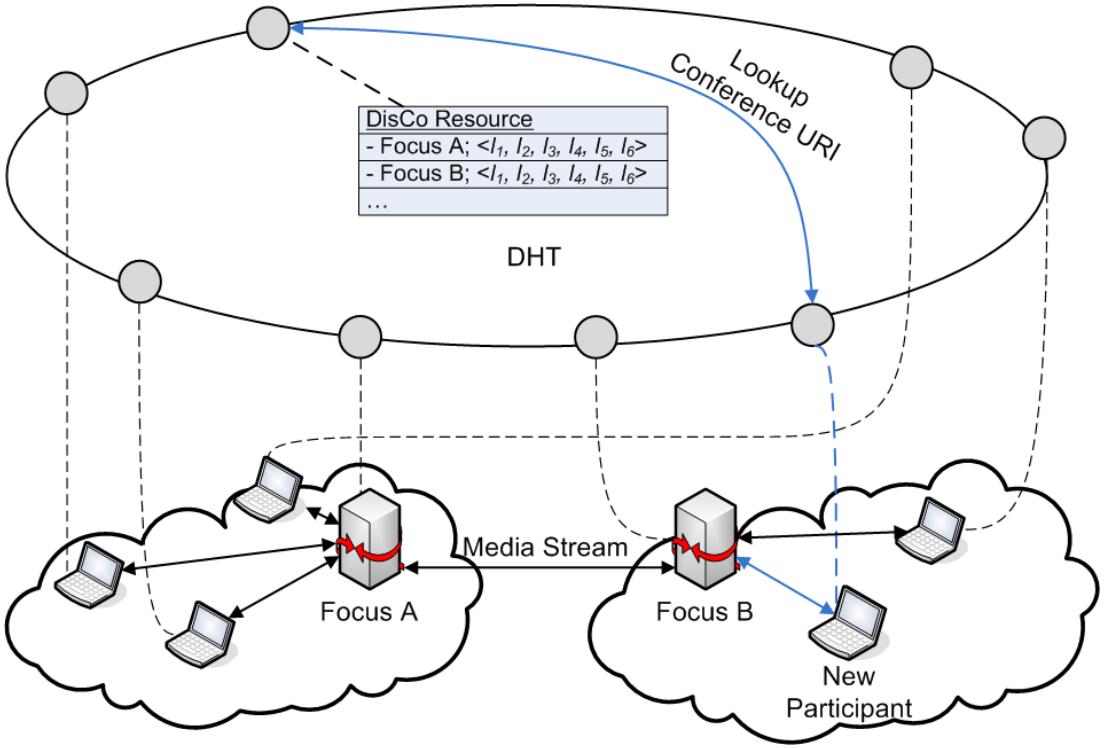


Attention on Racing Conditions!

Conference ID Registration

Using a RELOAD Overlay

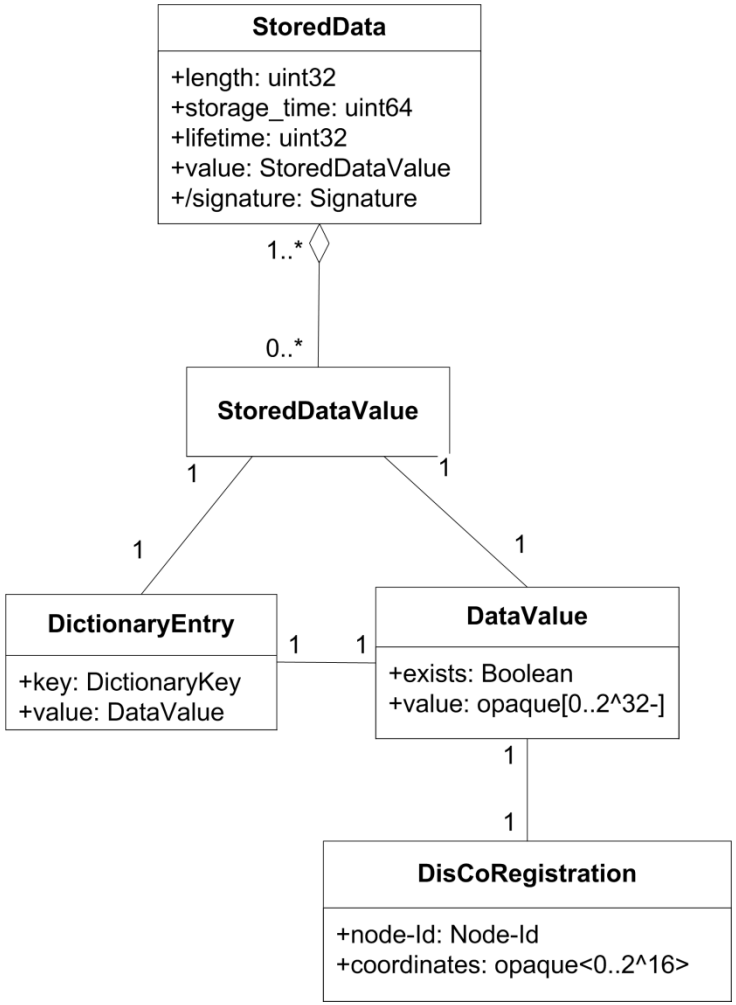
- **Problem:** How to announce several *entry points* to a single conference?
- **Idea:** Register URI in a **RELOAD** overlay as a *key* for several focus peers
- Detach the Conf-ID from any physical instance
  - Allows selection of a focus peer based on network proximity



RELOAD Data Structure

To Register a Distributed Conference (DisCo)

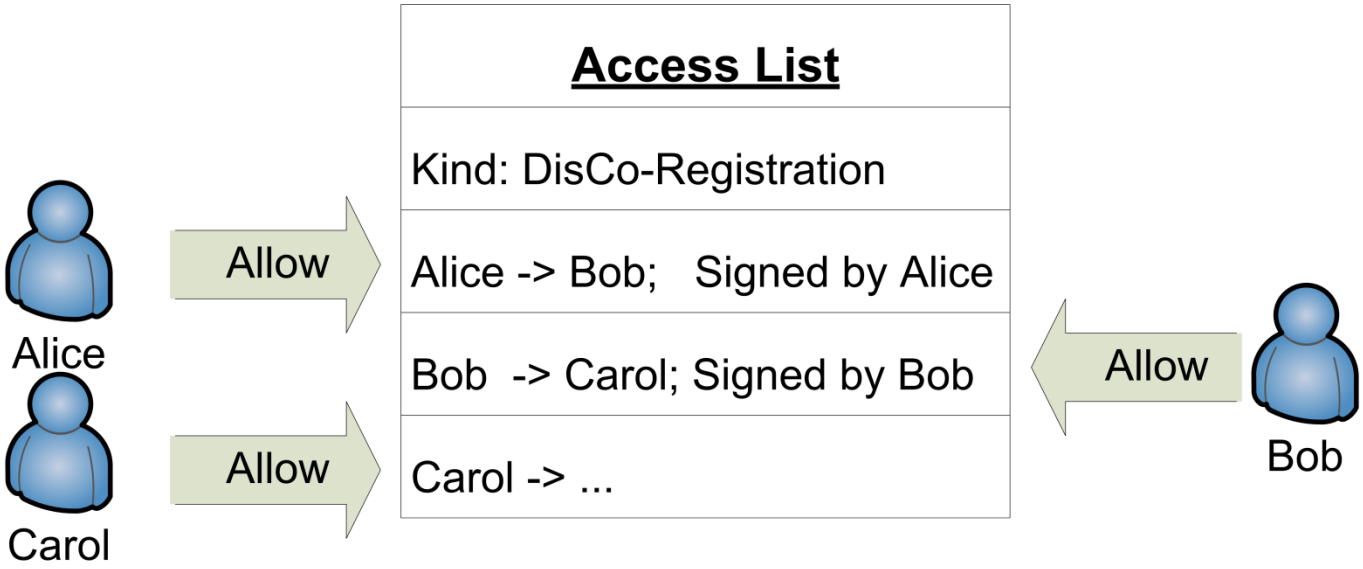
- DisCo-Registration stores a dictionary of :
  - *Node-IDs* of a Focus peer
  - *Coordinates vector* relative network position
- DisCo-Registration is a *shared* resource of all focus peers



Shared Resources

in a RELOAD Overlay

- **Problem:** How to share a Resource in a secure manner?
  - RELOADs access policy: Only owner can write, but
  - Usages allowed to *define new access policies*
- **Proposal:** Access List



Topological Descriptors

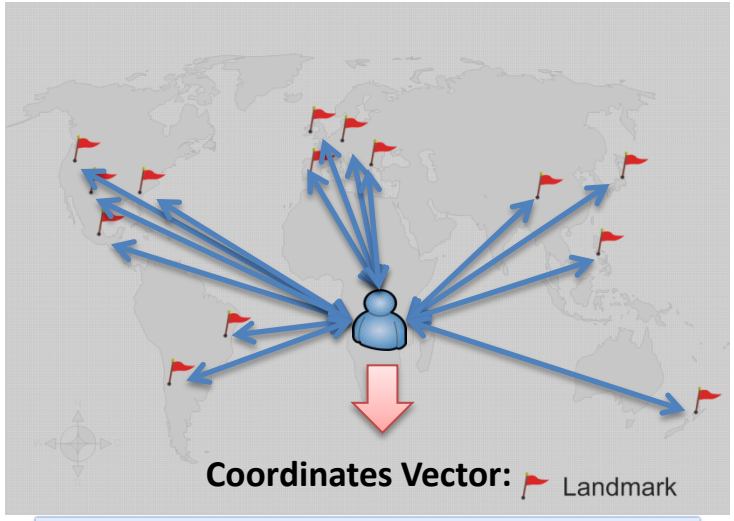
optimizing Conference Topology

- Each peer determines a *coordinate vector* - position in *n-dim* Cartesian space

- Distance between two peers p_1, p_2 is Euclidian distance between p_1's and p_2's coordinates vector:

$$d(p_1, p_2) = \sqrt{\sum_{i=1}^{n} (p_{1i} - p_{2i})^2}$$

- New participants select a focus peer whose Euclidian distance in minimal



<105ms,120ms,110ms,...,320ms,175ms>

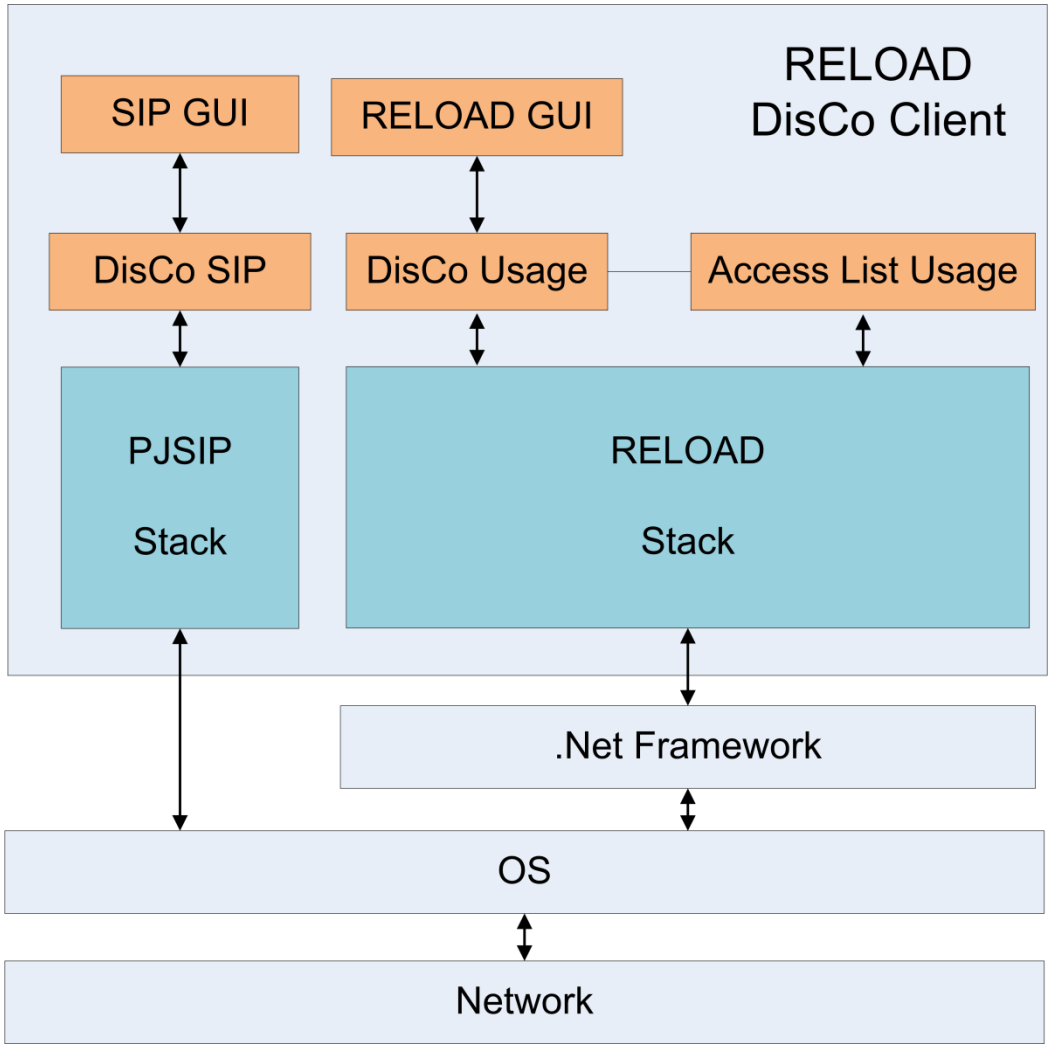
Starting Situation

towards Master Thesis

- Preliminary Works:
  - DisCo Concept: **“stable”**
  - ID/Locator Split: **works**
  - State Synchronization: **XSD Schema defined**
  - Proximity-awareness concept: **evaluated**
- Main Problem: No RELOAD implementation!
  - RELOAD spec. too large
  - Begun to emulate behavior...
- **Obtained RELOAD implementation** through Research Association with: ...

Software Architecture

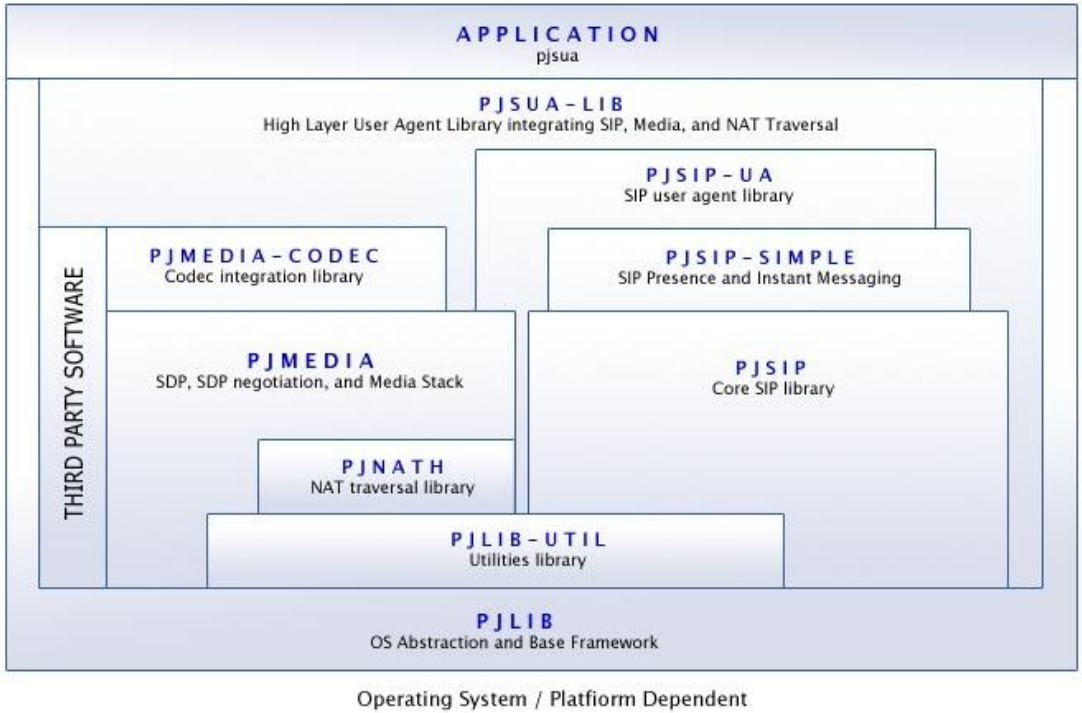
overview



PJSIP

SIP Stack

- Open Source
- Full SIP with Extensions
- Includes media library



RELOAD – Stack

P2P REsource LOcation And Discovery

- Almost a “full” RELOAD implementation
- Based on .NET
- Provides port for mobile devices
- Uses Google Maps for visualization

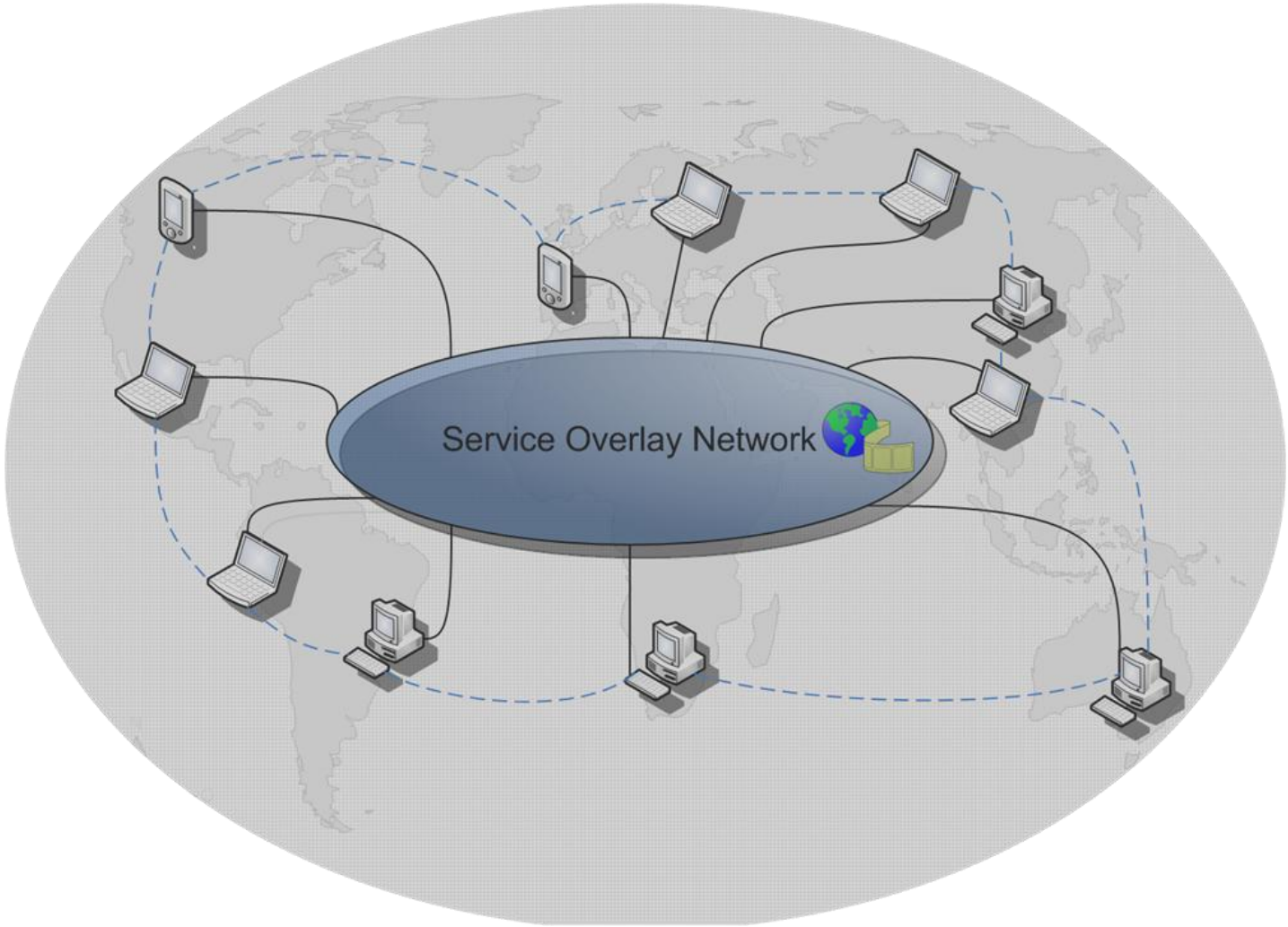
Risks

no Projects without them

- **DisCo Concept close to Internet standards:**
  - Partial aspects might conflict
  - Network stacks expect different behaviors
  - ➢ Change Concepts or Stacks
- **Evaluation of Proximity-awareness difficult:**
  - How to deploy several Node around the world?
  - Experimental Platforms (e.g., PlanetLab) do not support .NET
  - ➢ May need to emulate network topologies

Result Anticipation

of Distributed Conferencing



Conclusion & Outlook

- **Conclusion:**
  - Detailed Concept – presented at IETF78 in Maastricht
  - Partly implemented or evaluated
  - Finally – a RELOAD implementation

- **Outlook:**
  - Work in progress: -01 draft submission in P2PSIP WG
  - Separate document for a RELOAD *Usage for shared Resources*
  - Reference implementation for Distributed Conferencing
  - *Master Thesis* – for sure ☺

Questions?

Thanks for your attention!

http://inet.cpt.haw-hamburg.de/