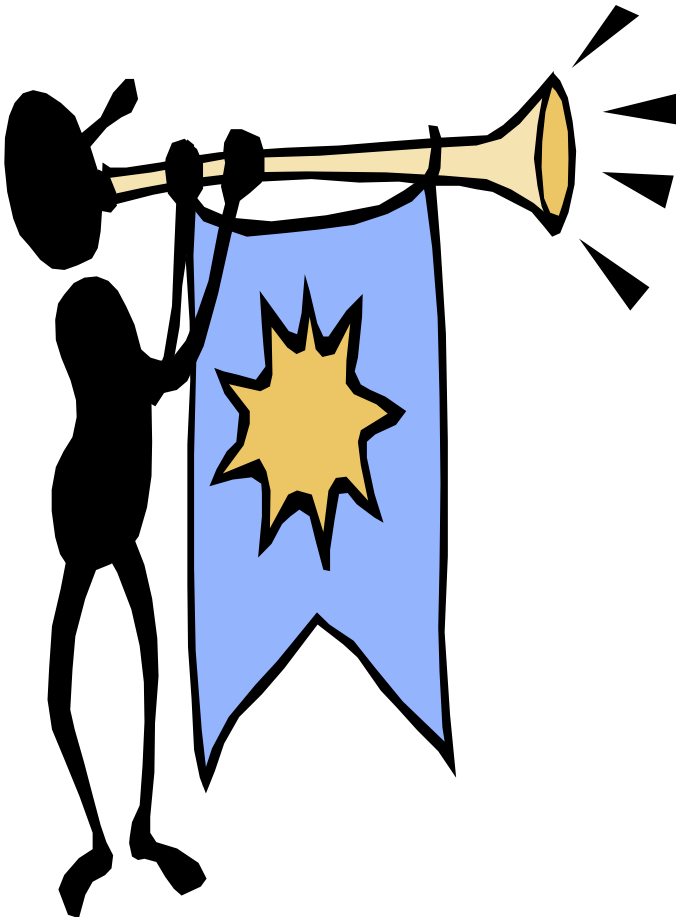
Telecommunication Services Engineering (TSE) Lab


# Chapter IX –
# Presence Applications and Services

http://users.encs.concordia.ca/~glitho/

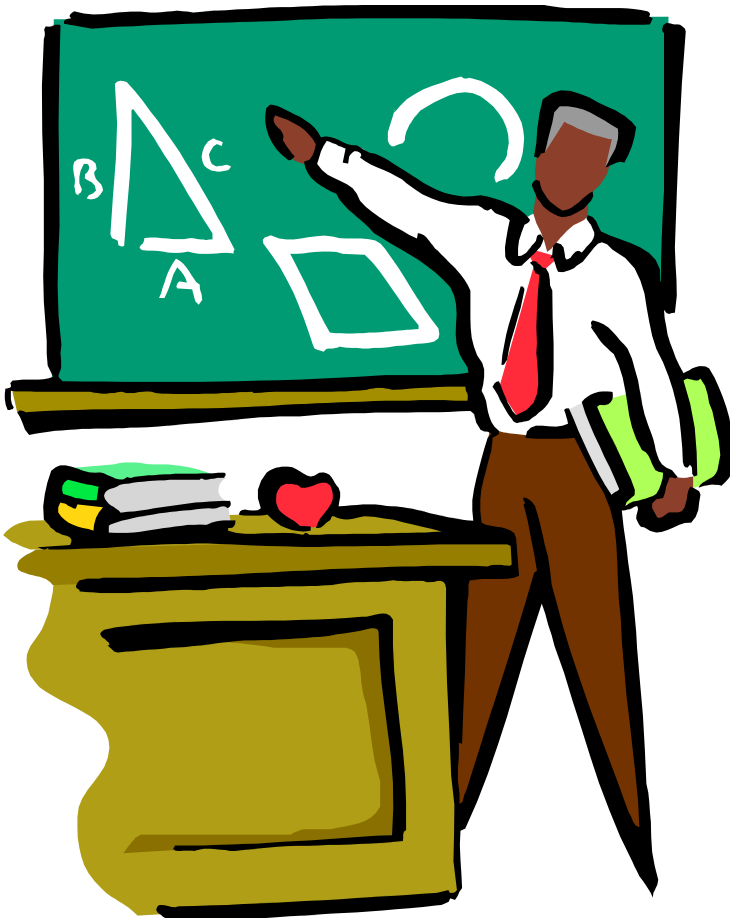
Concordia University
Engineering and Computer Science
Concordia Institute for Information Systems Engineering

# Outline



1. Basics
2. Interoperability
3. Presence service in clouds

# Basics

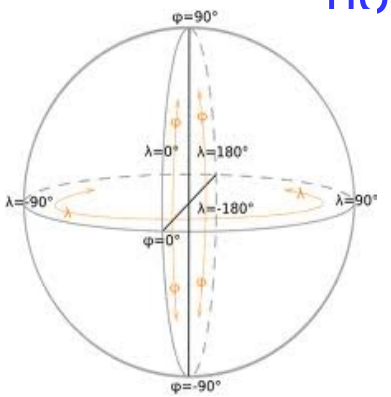



- 1 - IETF abstract model
- 2 - An example of implementation

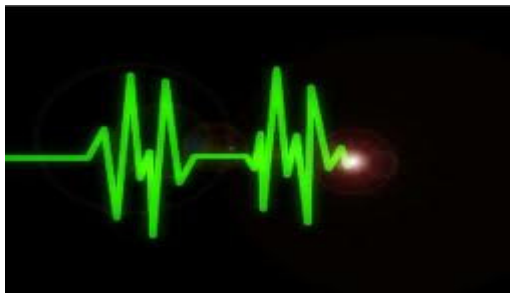
# IETF Abstract model

Presence service

- Enable the publication and the retrieval of context information (e.g. online/offline, willingness to communicate) or more generally
  - Space
  - Environment
  - Physiology
- Note: The information is collected by sensors that are not part of the presence framework







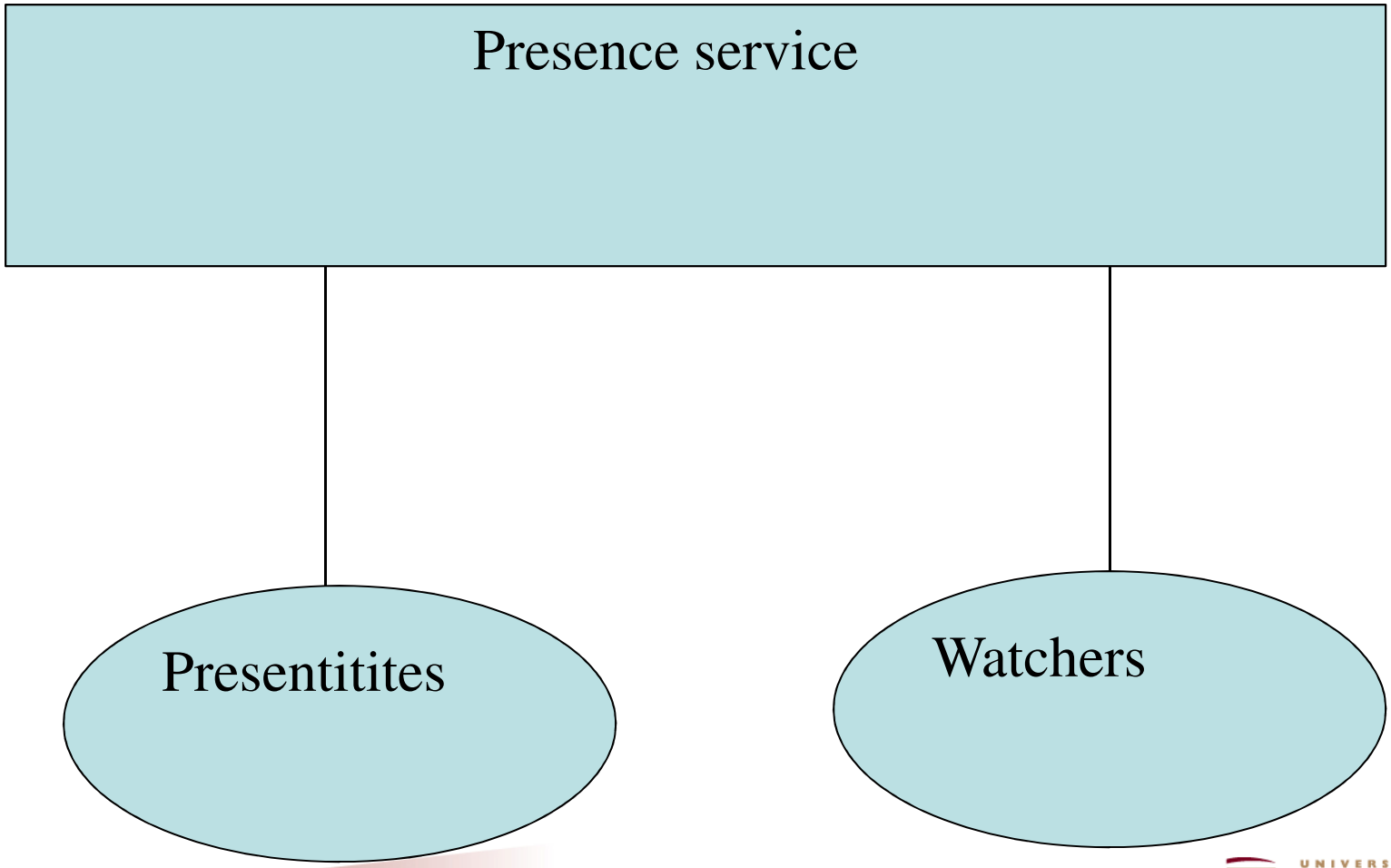
# IETF Abstract model

- Clients
  - Presentities: publish information
  - Watchers: Retrieve the information (pull, push)
- Presence service: accept the information (maybe centralized or distributed)

# IETF Abstract model

- Clients
  - Presentities: publish information
  - Watchers: Retrieve the information (pull, push)
- Presence service: accept the information (maybe centralized or distributed)

# IETF Abstract model



# IETF Abstract model

Two types of watchers

- Fetcher
  - Fetch the current value of some presentity presence information
- Subscriber
  - Request the notification of changes in some presence entity presence information

# An example of implementation

Two standard IETF implementations

- XMPP
- SIP SIMPLE
- Focus in this course: SIP SIMPLE
  -

# An example of implementation

SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions)

- Set of specifications for presence and instant messaging with SIP
- Cover among other things:
  - Core protocol machinery
  - Presence documents (XML based)
  - Privacy and policy
  - Provisioning

# An example of implementation

SIP SIMPLE

Protocol machinery

- Extended SIP event framework
  - Publish (New message)
  - Subscribe
  - Notify

SIMPLE presence documents

XML based

# An example of implementation

SIP SIMPLE

- Entities
  - Presence User Agent (PUA):
    - client (presentity or watcher)
  - Presence Agent (PA)
    - : stores presence information
- Presence Server (PS):
  - Acts as a PA or a proxy to forward requests to the appropriate destination

# An example of implementation

SIP SIMPLE

- Related messages:
  - SIP Publish, Subscribe and Notify
- Information model: XML based known as PIDF

Note: A plethora of other implementations exist (standard and proprietary)

# Examples of applications

Examples of applications areas

- Instant Messaging (IM)
  - Rely on a SIP extension (i.e. MESSAGE)
    - Do not require SIP sessions establishment although they may be sent within on-going SIP sessions.

# Examples of applications

Beyond instant messages

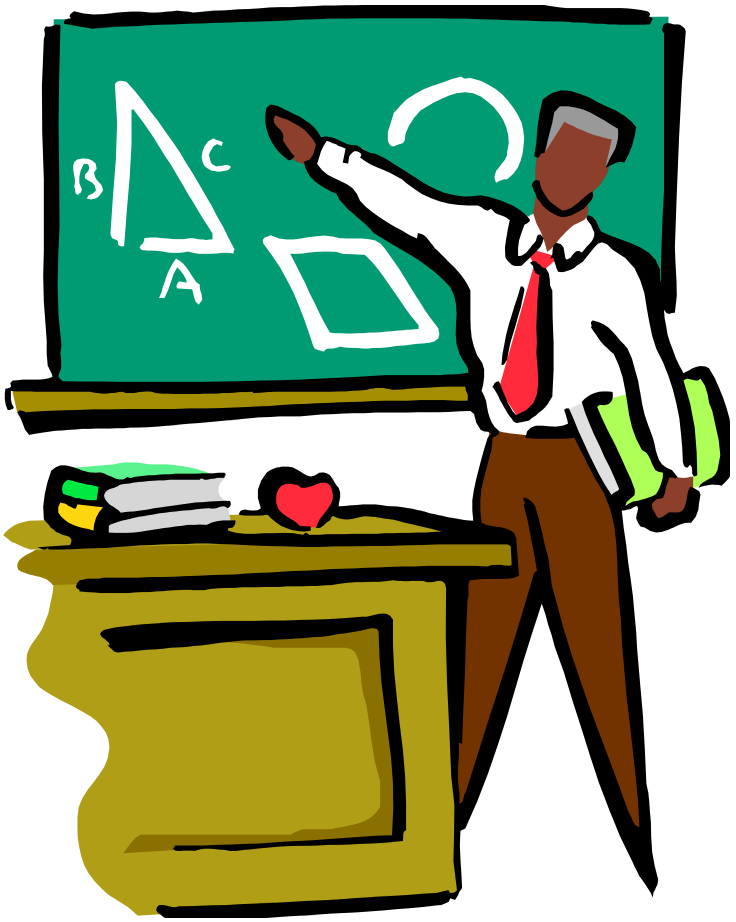
- Presence enabled conferencing (ie. Start a conference when a quorum is online)
- Unified Communications (e.g. Email, SMS, FM radio)
  - Use most appropriate communication mean
- Social networks

# Examples of applications

Illustration

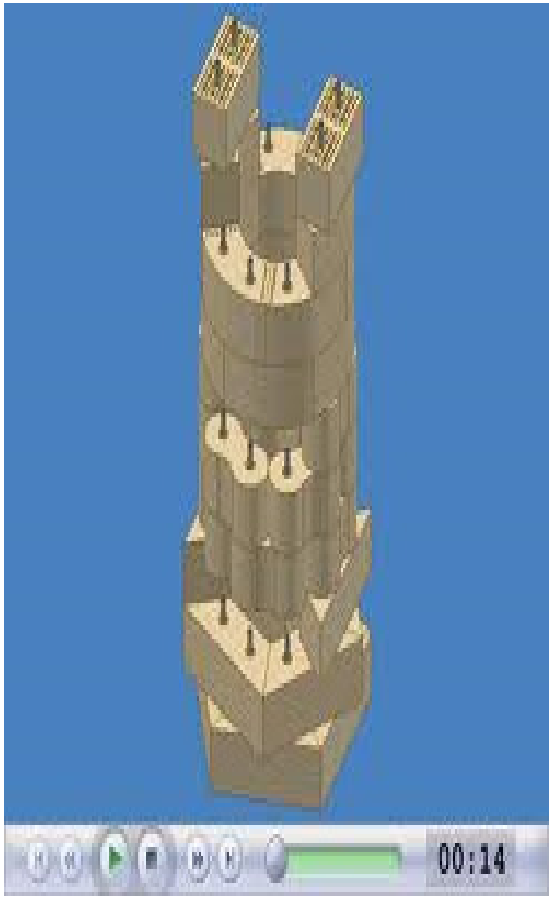
- Draw a sequence diagram with participants, P1 P2, P3, P4, P5, a conference server, and a presence server where the conference starts automatically when at least 3 participants are on-line and let us assume, P2, P4, and P5 get successively on-line

# Interoperability

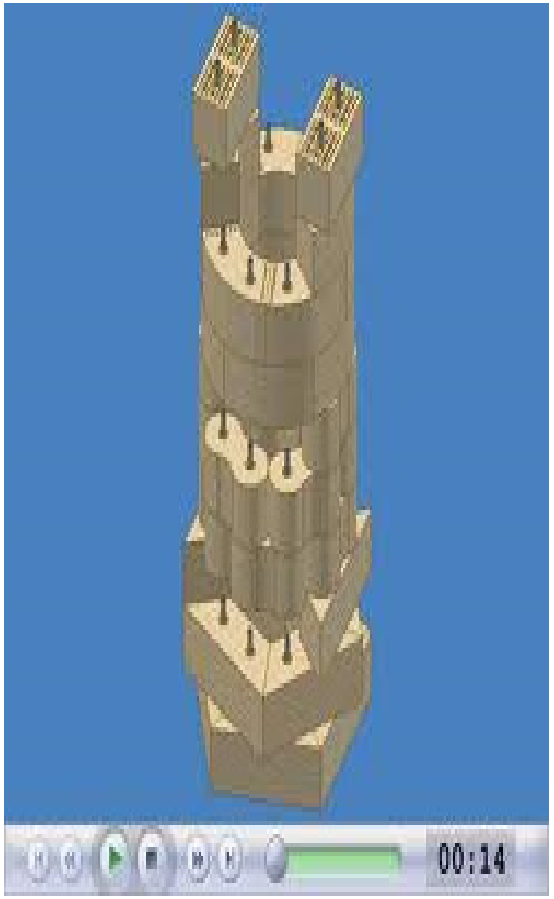



- 1 - Problem statement
- 2 - The state of the art
- 3. Our recently proposed solution

# Interoperability




# Interoperability




# Problem statement

The problem

- Several non interoperable standards (e.g. SIP SIMPLE, XMPP, Wireless Village)
- Several non interoperable proprietary implementations (e.g. Facebook, Skype)
- How to bridge the different worlds?

# Problem statement

The problem

- More concretely
  - Users may have several accounts with several providers with non interoperable implementations
  - How can a presentity with multiple accounts publish context information via the account she/he is using at a given time, with the possibility for watchers to retrieve this information via any of the other accounts?

# The state of the art

- Bidirectional mapping (e.g. IETF spec for XMPP / SIP SIMPLE mapping)
  - Exponential growth with number of standard and non standard presence implementations
  - Few specifications exist (XMPP / SIP SIMPLE only known standard)

# The State of the Art

- Open source SIP communicator
  - Support all protocols
  - Not realistic

# The State of the Art

- SOAP based Web services used as common denominator / glue
  - Example: YooHoo
    - Enable end-user to access different presence services through a Big Web services
  - Problem: small footprint devices

# Our proposed solution

## Our proposed centralized solution

- **C. Fu, F. Belqasmi, R. Glitho, RESTful Web Services fro Bridging Presence Service across Technologies and Domains: An Early Feasibility Prototype, *IEEE Communications Magazine*, December 2010, Vol. 48, No12.**

# Centralized solution

Our design goals:

- No restriction on the business model
- Avoid one to one mapping
- Presence  service (e.g. XMPP, SIP) interface neutral
- Unique and ligthweight interface to clients
- Accommodate clients behind firewalls
- Deployment on  Web servers with no additional server
- No modification to existing presence servers

**Selected “bridging” technology: RESTFul Web services**

# Centralized solution

## Our centralized approach: The business model

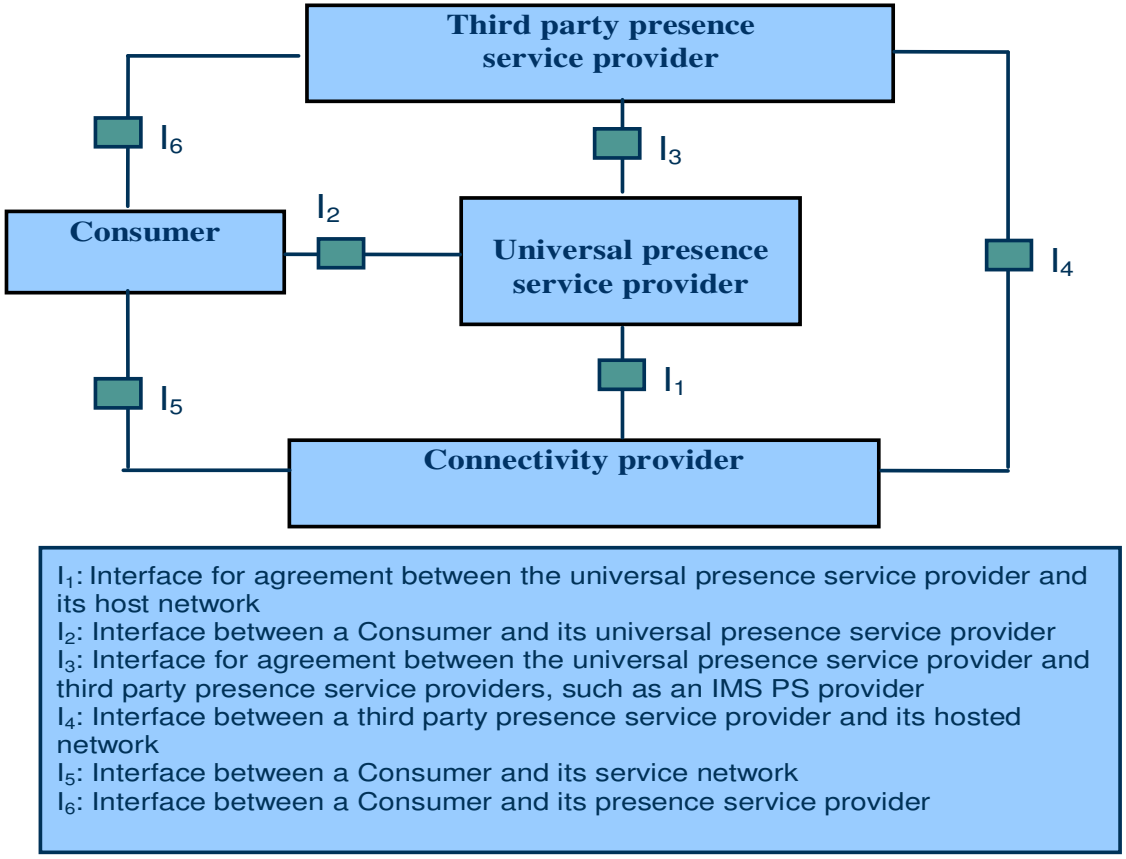



Figure 1: Business Model

# Centralized solution

## Our centralized approach: The architecture

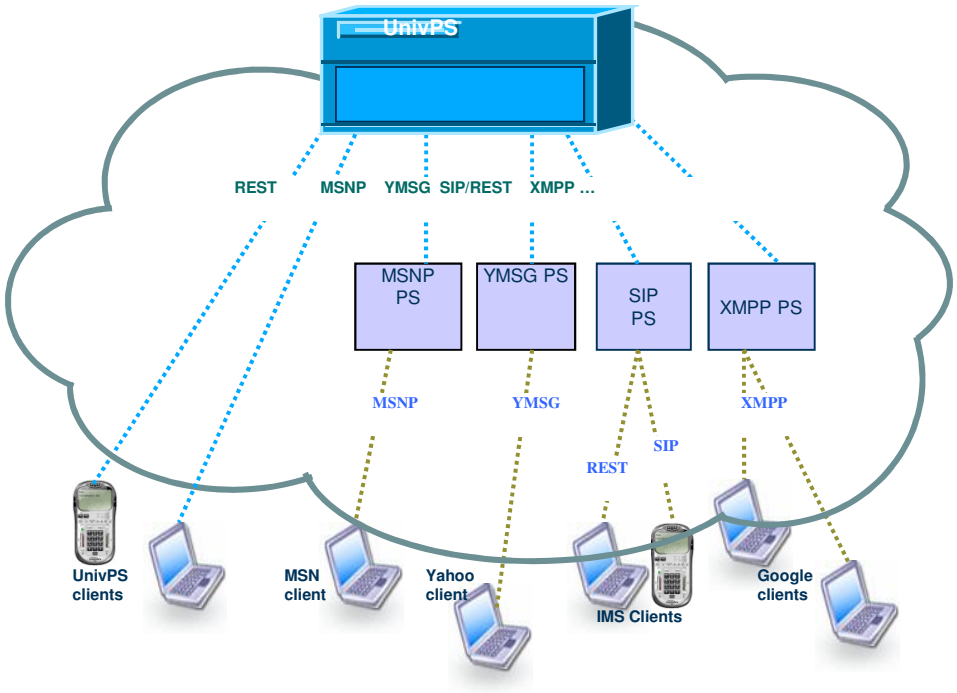



Figure 2: Overall Architecture

# Centralized solution

## Our centralized approach: Universal presence server

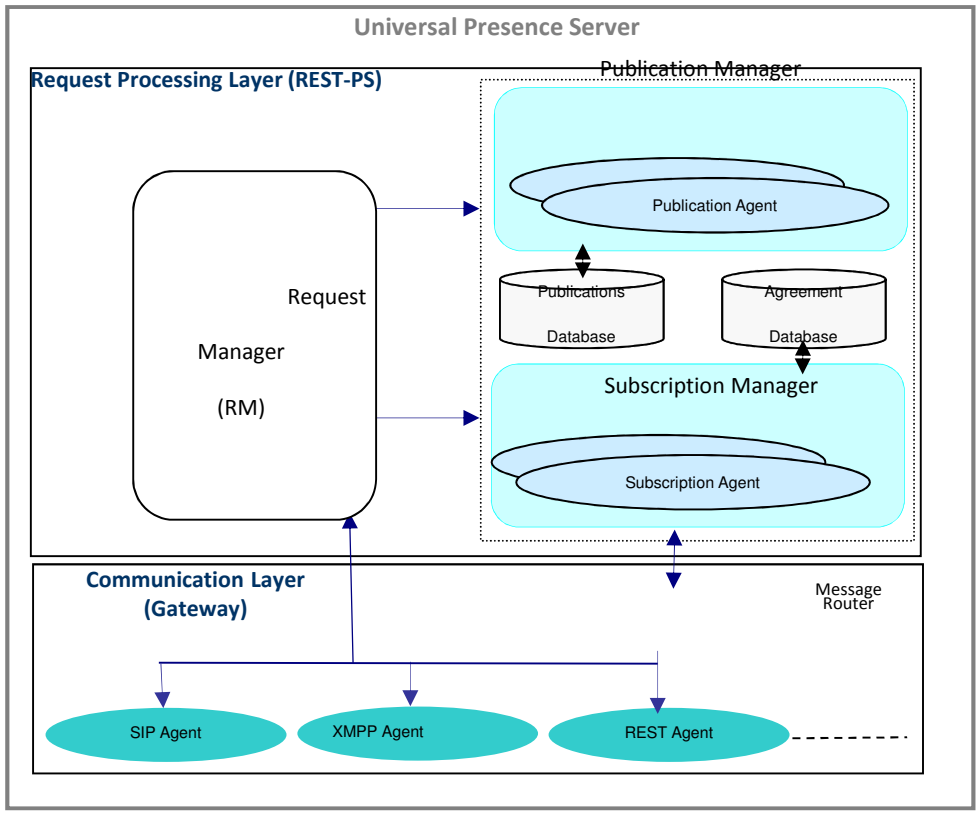
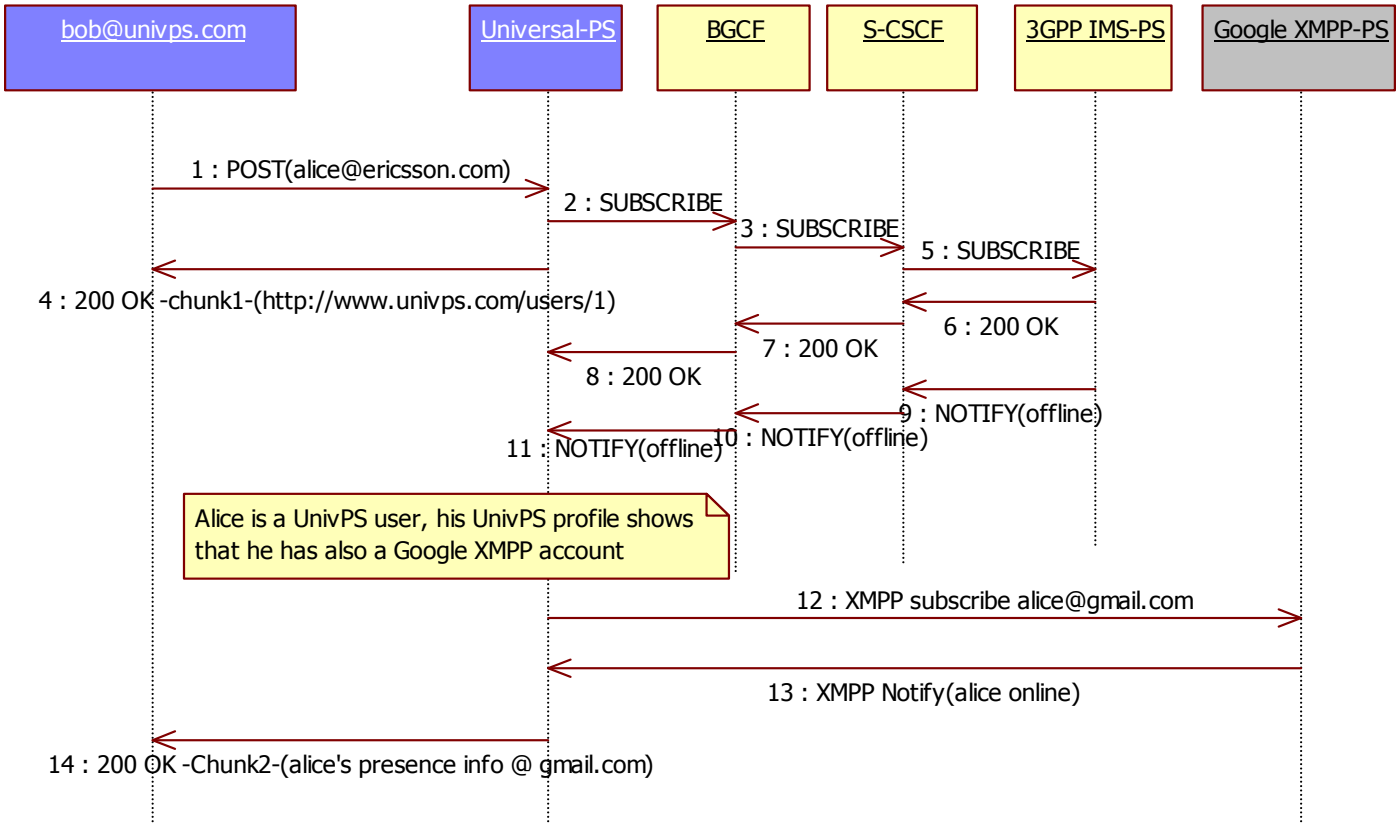



Figure 4: Software architecture of the UnviPS

# Centralized solution

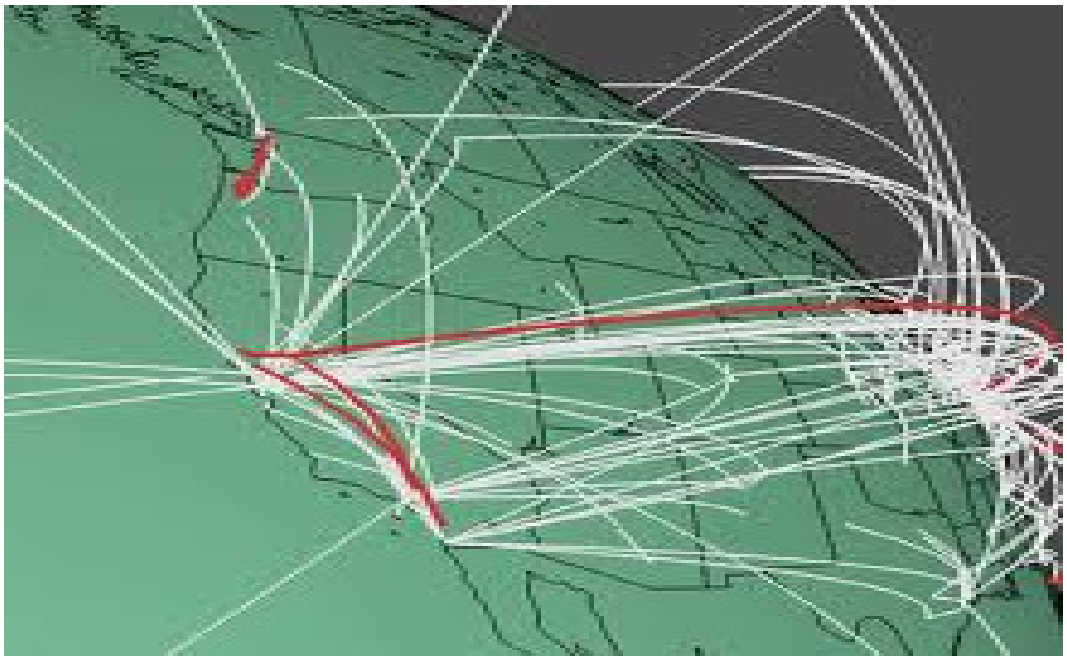
Our centralized approach: Scenario (Alice off-line in IMS but online in Google)



# Interoperability

## Our de-centralized approach: A P2P Approach

- C. Fu, F. Beqasmi, M. Shasvelayati, H. Khlifi, R. Glitho, A Peer to Peer Architecture for Enabling Universal Presence Service, *Next Generation Mobile applications, Services and Technologies (NGMAST 2010)*, 26 – 29 July 2010, Amman, Jordan



# Our decentralized solution

Differences with previous approach:

- Service offered by a federation of providers instead of a single universal presence service provider

**Selected "bridging" technology: P2P overlays**

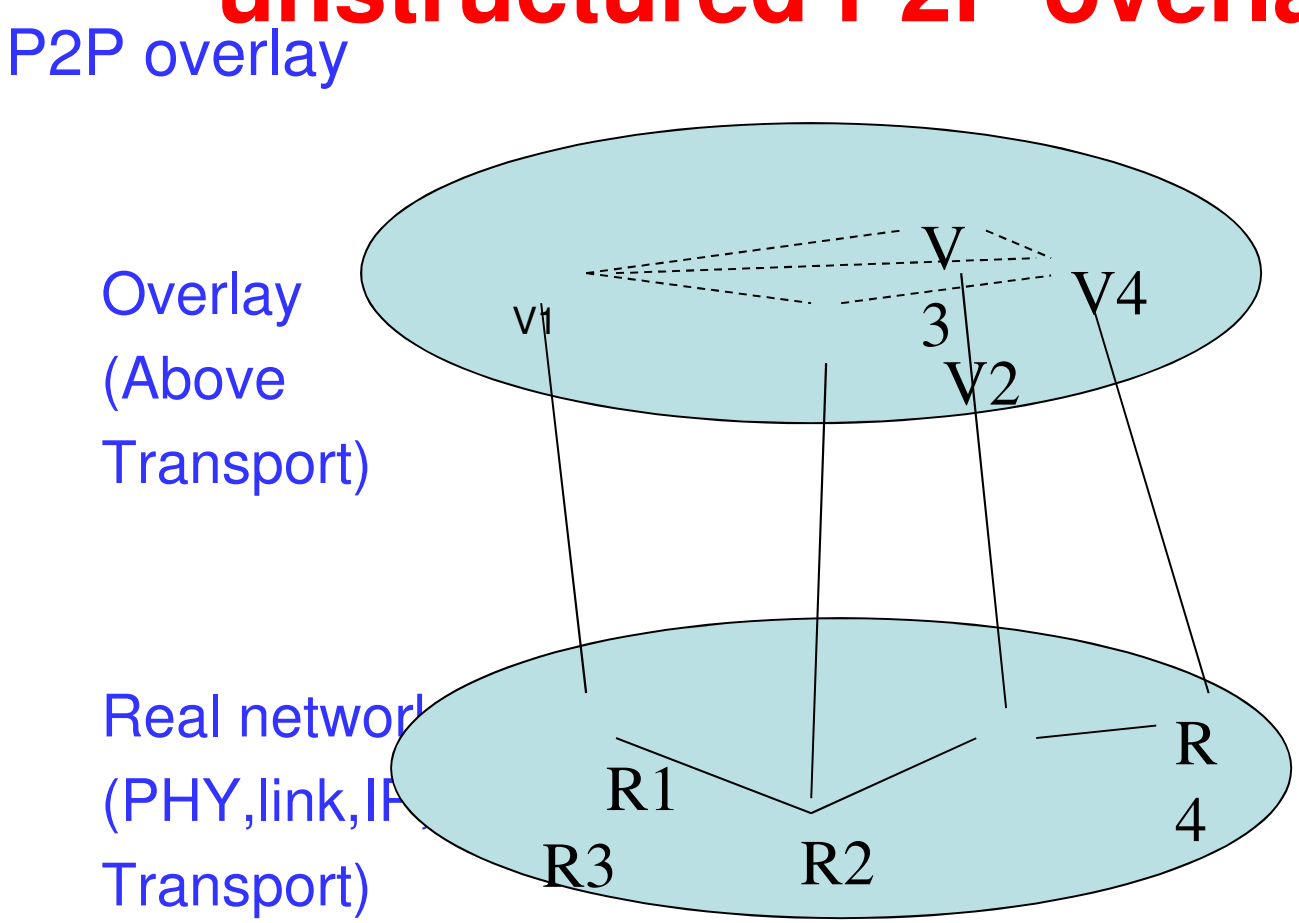
Overlay built on top of real presence servers and real services nodes (e.g. protocol translators)

# Our decentralized solution

On P2P overlay

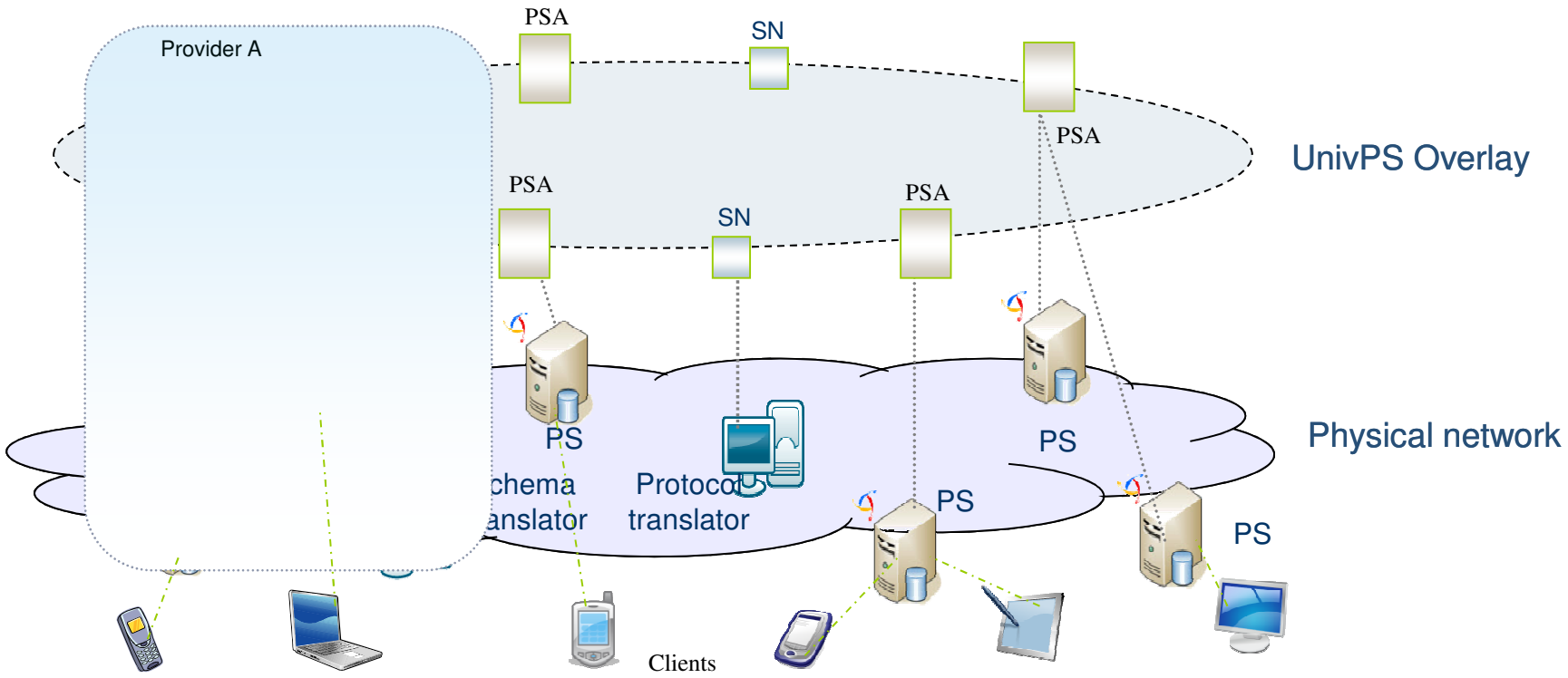
- Current way of implementing P2P computing
  - Application layer virtual networks that provide storage, processing, connectivity and routing
    - Network built by peers that federate to offer storage and processing capabilities to applications
      - Built on top of existing networks, thus the name of overlay
        - » Applications running on top of transport protocols of real network
        - » Real network nodes become virtual nodes in the overlay

# Structured P2P overlays vs. unstructured P2P overlays



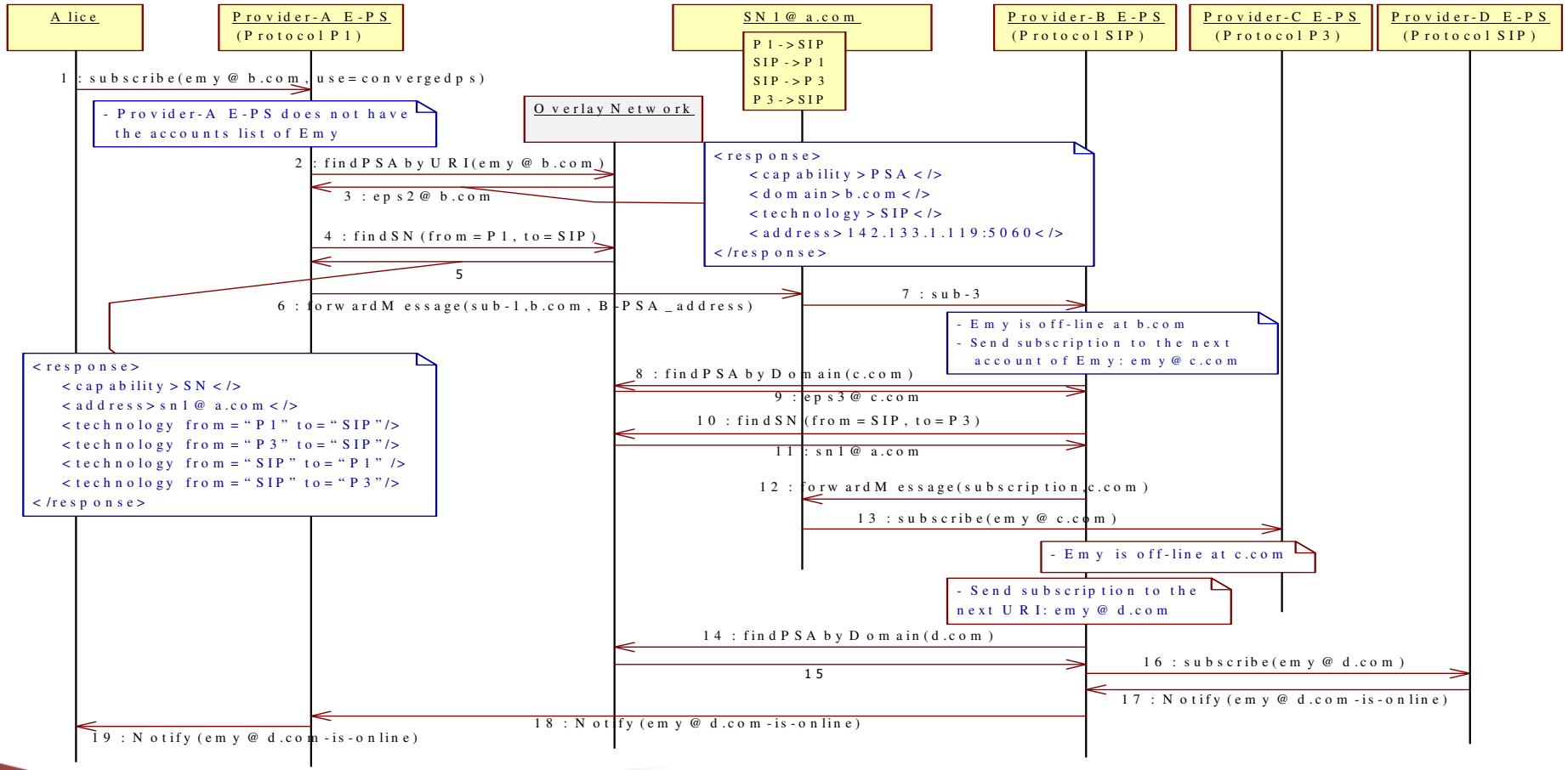
# Our decentralized solution

## Our de-centralized approach: A P2P Approach

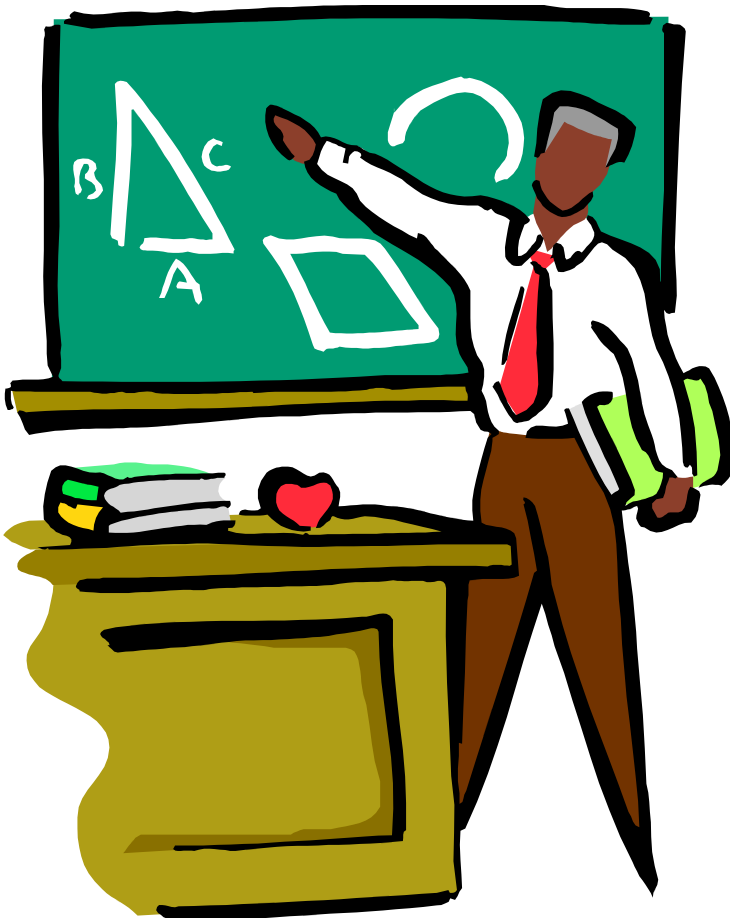


# Our decentralized solution

## Our de-centralized approach: A P2P Approach



# Presence services in clouds



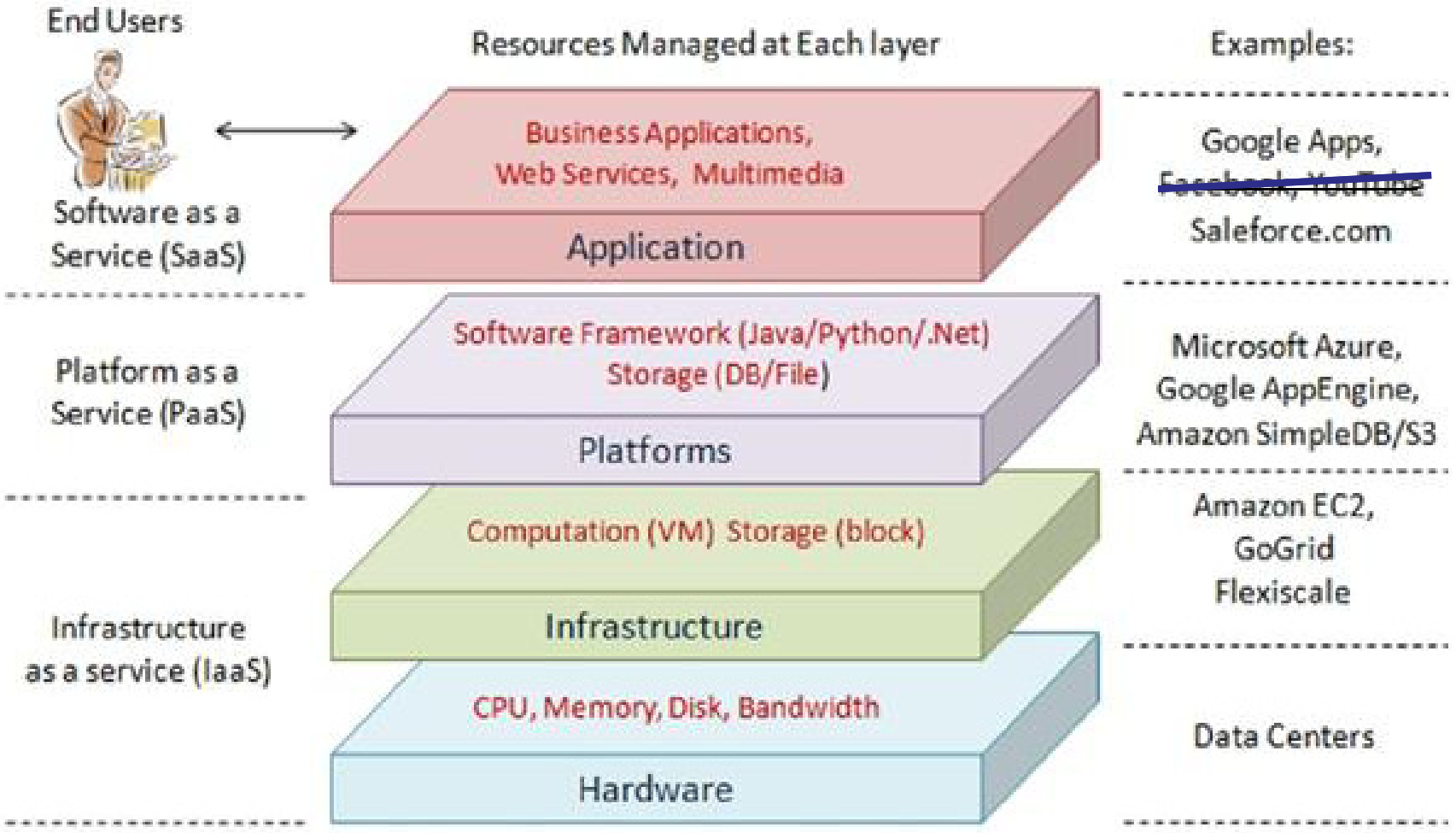

- 1 - Cloud basics
- 2 - Virtualization
- 3 - Problem statement
- 4 - Virtualized presence service

# Cloud basics

Motivations / expected benefits

- Cost reduction (hardware/software resources and their management)
- Consumption/usage -based pricing
- Elasticity
- Flexibility in service deployment

# Cloud basics



# Cloud basics

## Software as a Service (SaaS)

- Accessible via GUI but also via programmatic interfaces (i.e. APIs)
- Alternative to running services locally

## Platform as a Service (PaaS)

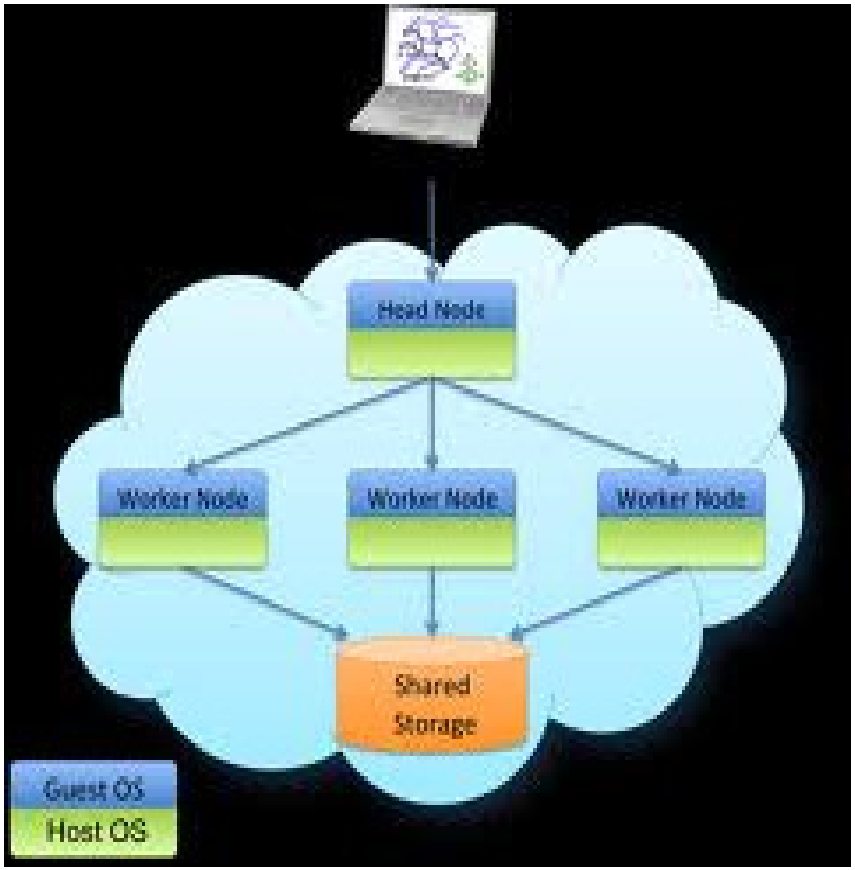
- Development and management of services hosted in the cloud (e.g. Google Apps Engine)

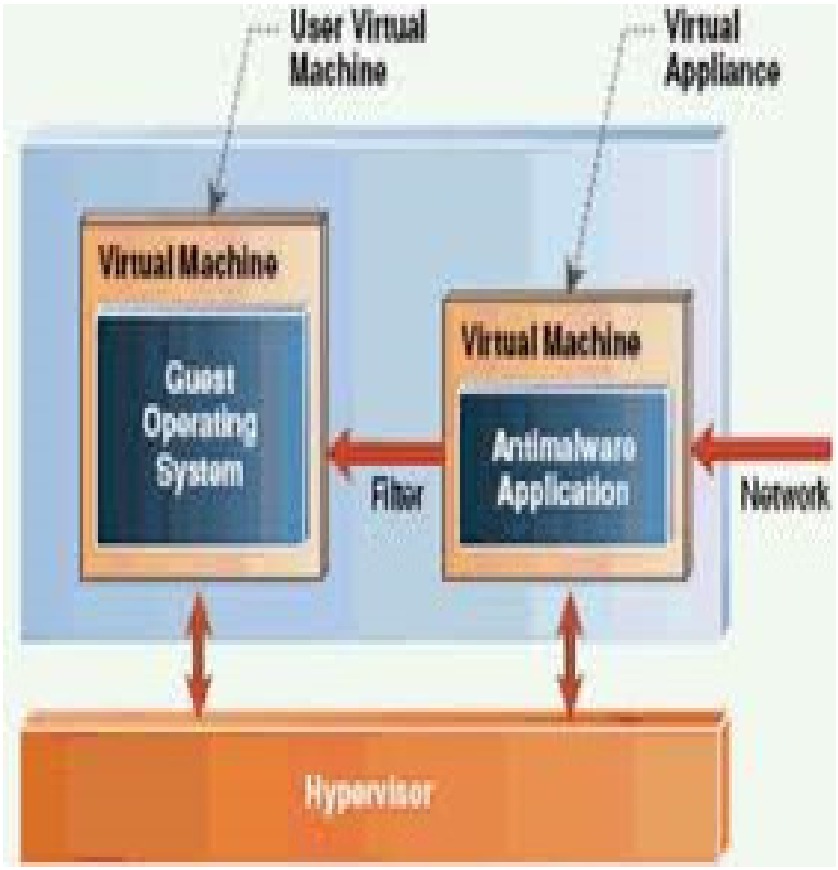
## Infrastructure as a Service (IaaS)

- Computing resources (e.g. storage and processing) that are dynamically split, assigned and re-sized.

  Key technology: Virtualization

# Virtualization



# Virtualization

Basics of virtualization

- Enable the co-existence of entities on same substrates/building blocks - e.g.
- Several operating systems on a same machine
- several networks on a same set of routers / links
- Advantages
  - Efficient usage of resources
  - Easy introduction of new functionalities

# Virtualization

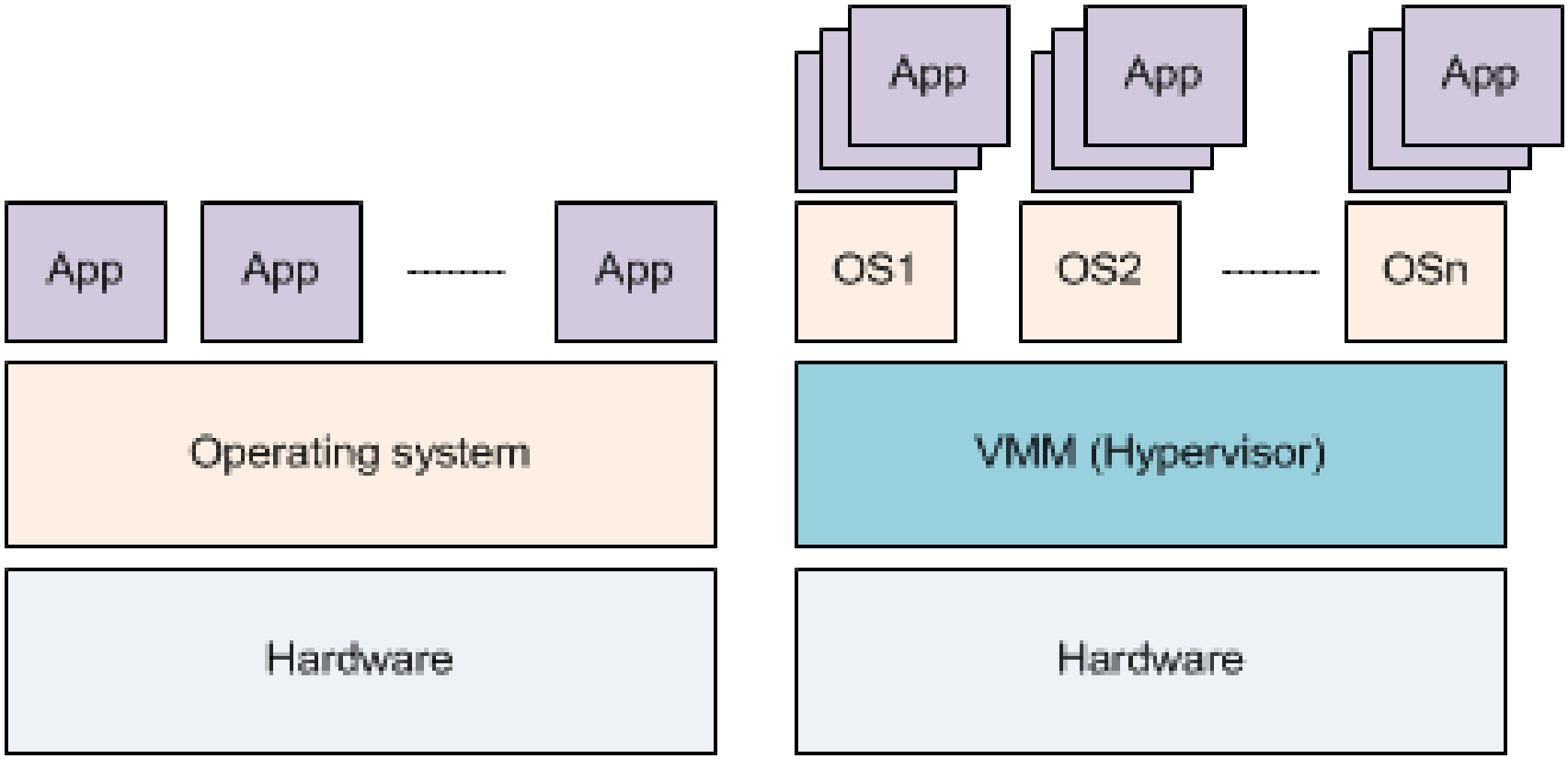



**Figure 2:** Non virtualized environment vs. non virtualized environment: An example from the computer world

# Virtualization

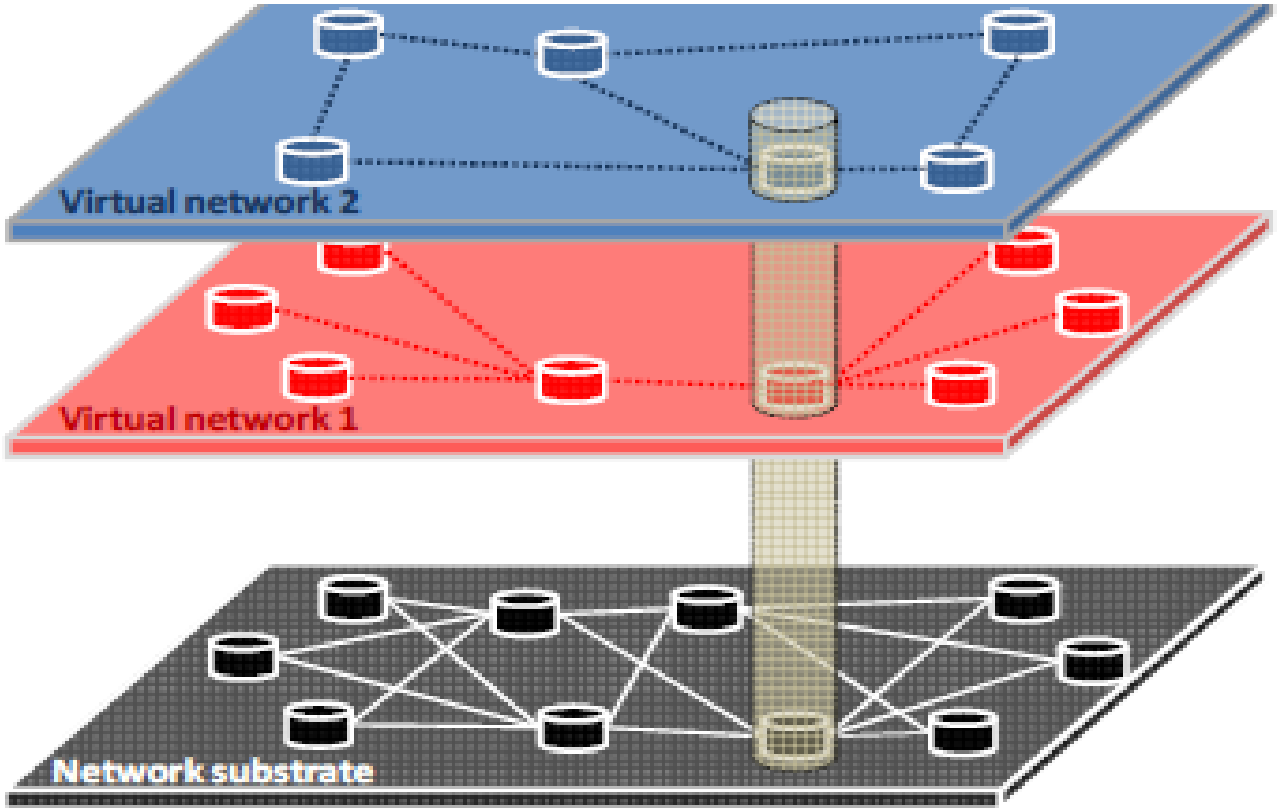



**Figure 2:** 2 virtual networks using a same substrate network)

http://conferences.sigcomm.org/sigcomm/2009/workshops/visa/papers/p73.pdf

# Virtualization

Basics of virtualization

- Differences between virtualization and VPN
  - VPN does not allow heterogeneous networks on a same infrastructure because based on a same technology
- Differences between virtualization and overlay networks
  - Overlays are designed independently of each other and may interact in an harmful way
  - Clear separation between substrate and virtual networks in case of virtualization

# Problem statement

The problem

- Presence service is expected to grown significantly in the future (e.g. social networks), thus the need to:
  - Re-use presence substrates / building blocks across standard and proprietary implementations for cost sharing
  - Easily introduce new presence functionality

**Potential technological basis for a solution:**

**Virtualization**

# Problem statement

Our ultimate goal: Presence service in the cloud

- On-going step: Infrastructure as a Service (IaaS)
  - Virtualized presence service
- Next steps
  - Platforms as a Service (Paas)
    - Multi-level abstraction APIs
      - Experienced developers
      - Novice developers
  - Software as a Service (SaaS)
    - Access by other applications and access by end-users

# Virtualized presence service

Our proposed solution for presence service virtualization

F. Belqasmi, N. Kara, R. Glitho, A Novel Virtualized Presence Service Architecture for Future Internet, accepted, IEEE ICC 2011 Workshop on Future

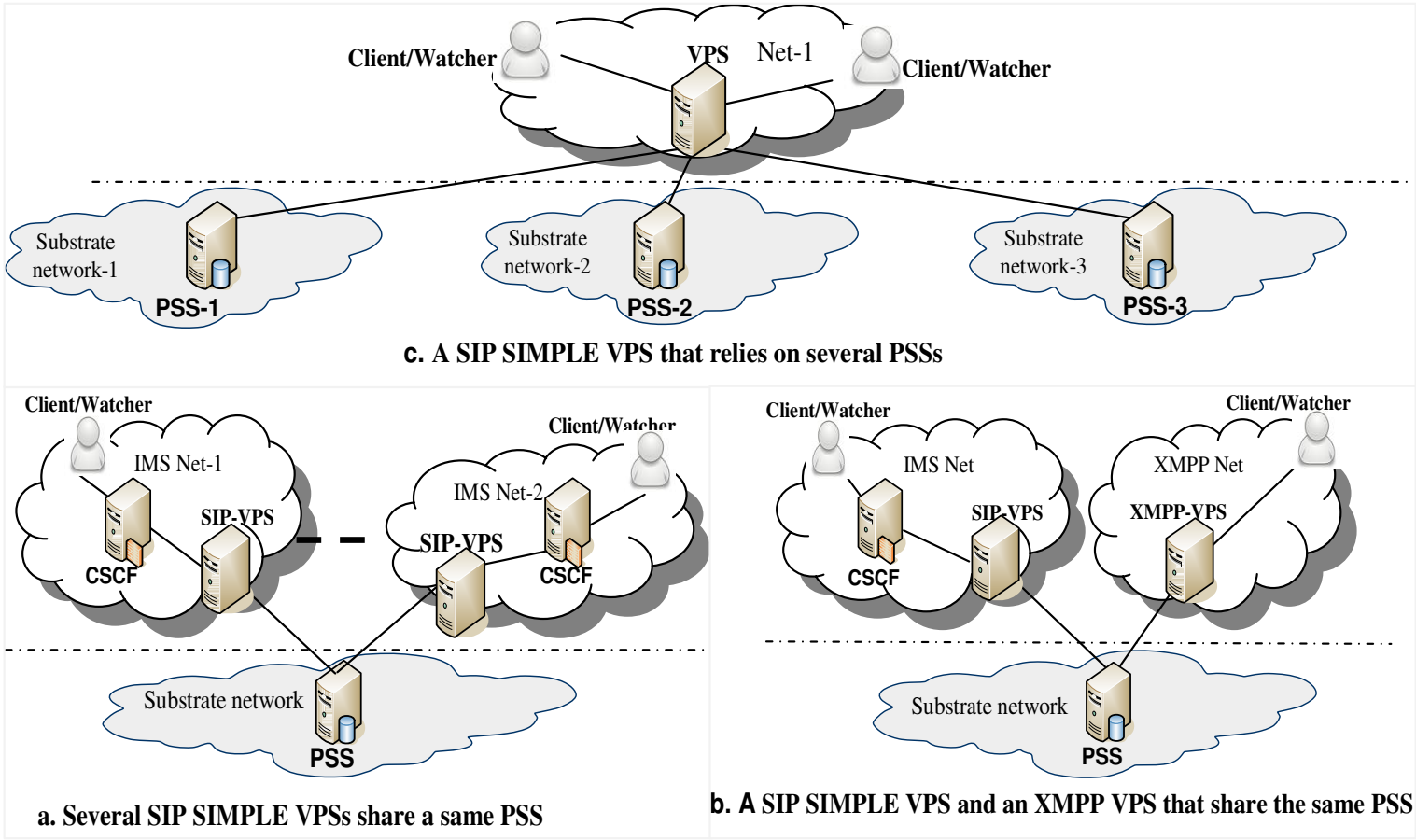
# Virtualized presence service

Our terminology

- Virtual Presence Service (VPS)
  - Presence service as deployed by service providers (e.g. SIP SIMPLE, XMPP, Facebook)
- PSS (Presence Service Substrate)
  - Sharable building blocks that could be re-used across standard and non standard VPSs

# Virtualized presence service

## Our proposed solution: Scenarios



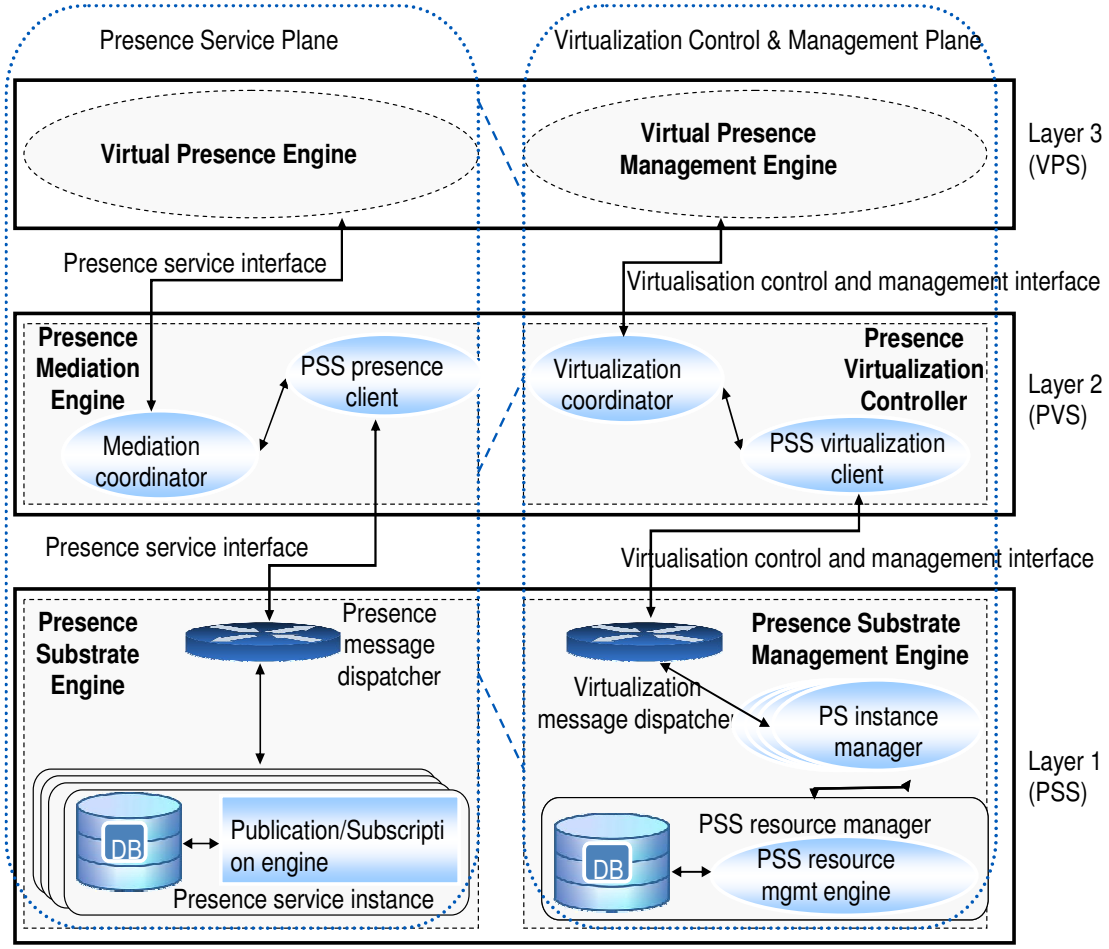
# Virtualized presence service

Our design goals

- Use of a same PSS by several VPS that may be in separate domains
- Use of several PSS by a same VPS for scalability purpose
- VPS and PSS should interact dynamically
- PSS should enable the rapid deployment of a very wide range of VPS.

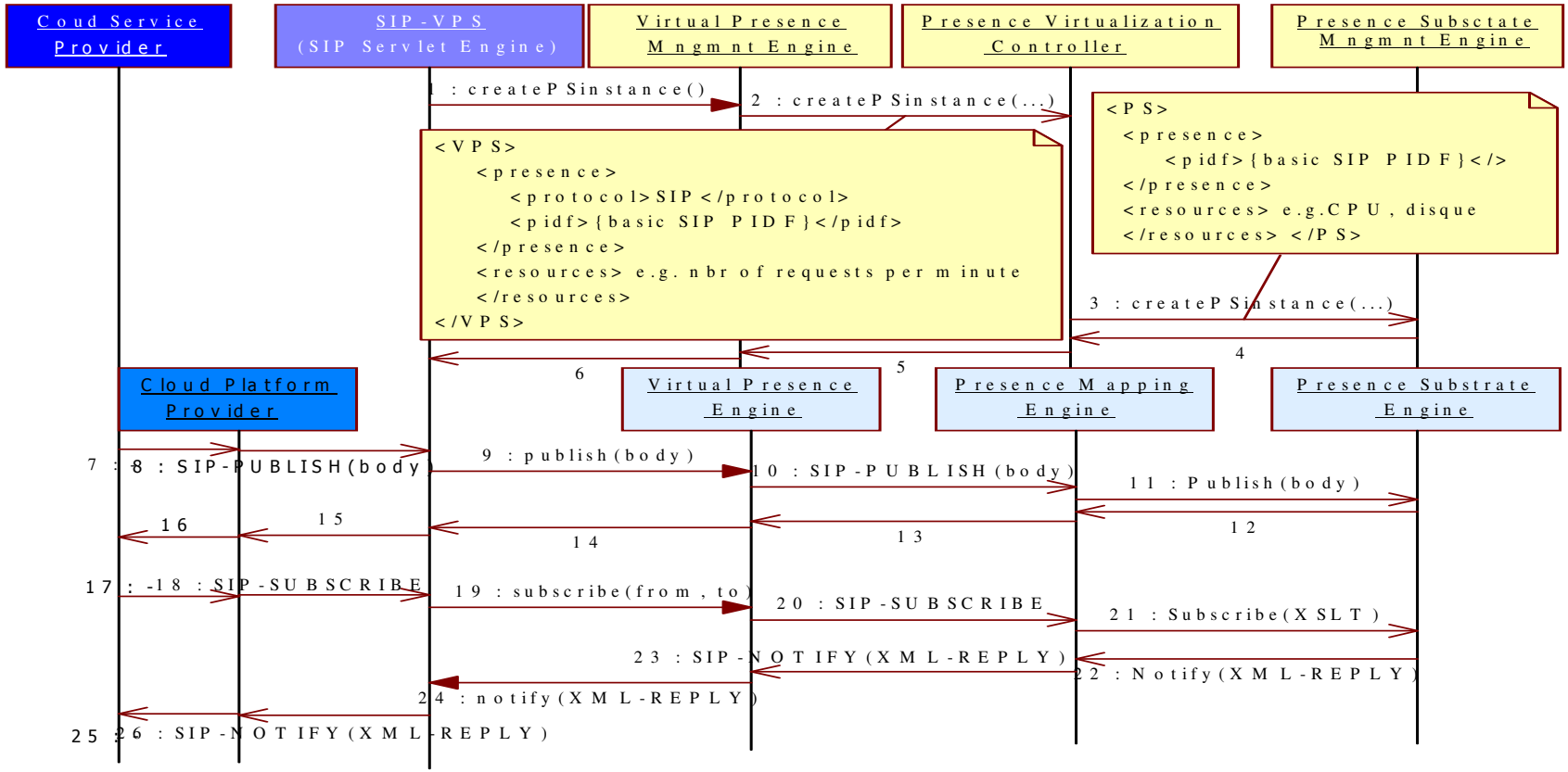
# Virtualized presence service

## Our proposed solution: architecture



# Virtualized presence service

## Our proposed solution: architecture (SIP VPS instantiation and use)



# References

1. L. M. Vaquero et al., “A Break in the Clouds: Towards a Cloud Definition”, ACM SIGCOMM Computer Communication Review, Vol. 39, No1, January 2009
2. J. Carapinha; J. Jiménez, Network virtualization: a view from the bottom, Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures (VISA-09), pp. 73-80, 2009