Presence Aggregation in Endpoints

Jörg Ott <jo@tzi.org>

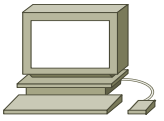
15 January 2003

Upperside SIP 2003 - Paris

Reminder: Presence in "VoIP"

Awareness of other users: availability, location, …

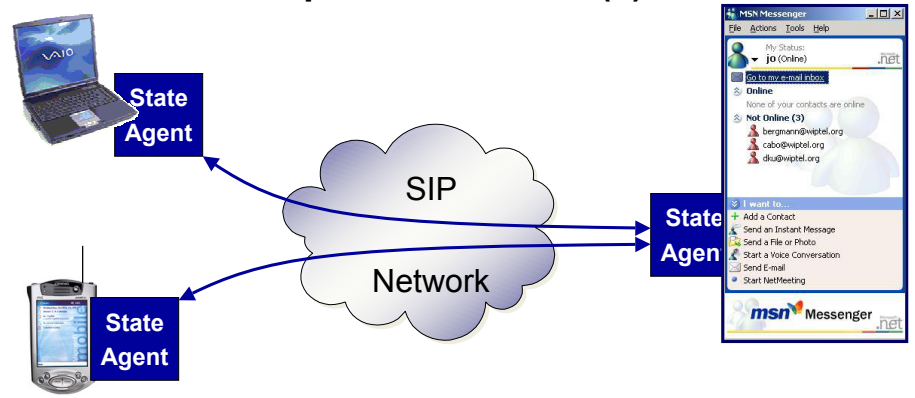
- To perform existing functions
  - User location, call routing, follow-me, ...
- To improve existing services
  - Call completion ratio
  - Indicate availability
- To enable new services
  - Presence per se: Simplify meeting people
  - Messaging per se: "SMS"
  - Presence and Messaging as basis for other applications
  - Location-based services
- To rescue the "VoIP" industry…



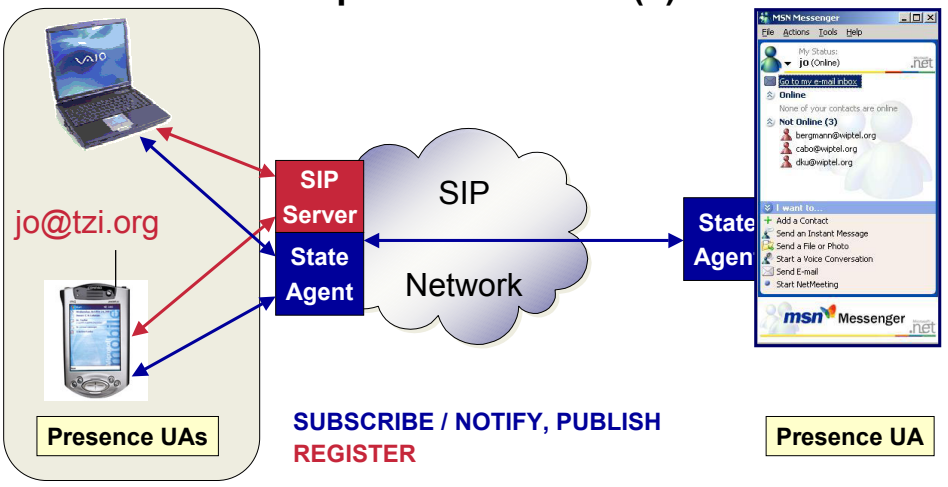
Presence in SIP: SIMPLE

- Roles
  - Presence User Agent (PUA)
  - Presence Agent: Intermediary or co-located with PUA
    - co-located with a PUA: Edge Presence Server
  - Presence Server: physical box implementing SIMPLE
  - Compositor
    - Aggregate state information from multiple PUAs
- SUBSCRIBE / NOTIFY Methods, PUBLISH
- Event Packages
  - Distribute user state (presence)
  - Manage subscriptions for a user (watcher-info)
  - For other purposes (dialog-info, conference-info, ...)

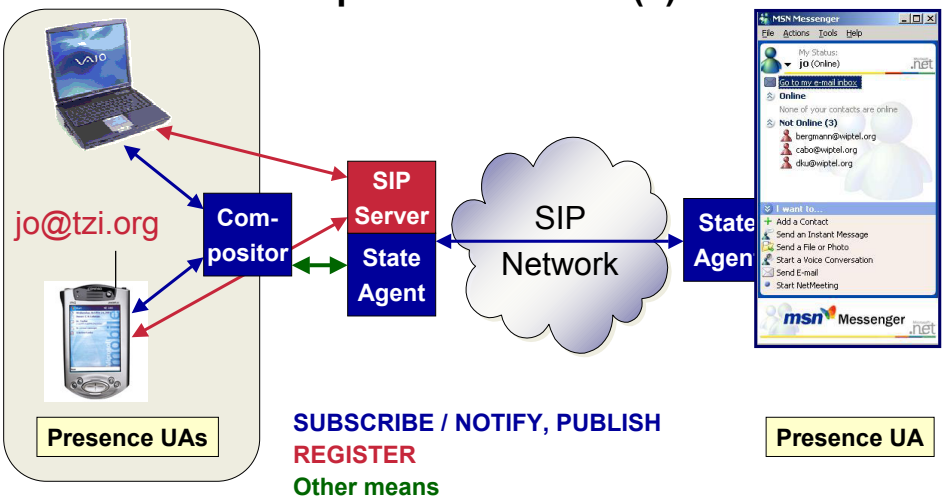
Sample Architecture (1)



Sample Architecture (2)



Sample Architecture (3)



Issue: Keeping Presence State Up-to-date

Providing input timely and always

- **Explicitly via GUIs**
  - Need to remember
  - "How much do I invest – how much do I get in return…?"
- **Implicitly derived**
  - Idle time, Call / phone status, databases, ...
- **Various sources of input**
  - Databases, calendar, building information system, car park, ...
  - Workstation, laptop, PDA, ...
  - IP phone, legacy phone (via PBX), cell phone?
  - Other devices: camera, chair sensor, motion detector, ...

TZi Technologie-Zentrum Informatik

Issue: Keeping Presence State Up-to-date

Need to collect input from many sources

Need to aggregate input in a meaningful way

Need to disseminate results appropriately

Universität Bremen
© 2003 Jörg Ott

Collecting Presence Input…

- Single source of input is rarely sufficiently accurate alone
- Enable many sources
- Keep user in control (and keep data under user control!)

- No collection: independent (e2e) delivery of state info
- SIMPLE Role: Compositor for presence state collection
  - Within "user environment" or centralized
- Other local collection:
  - Group entities contributing to presence state
  - (Dynamically) choose representative ("presence proxy")
  - Integrate with non-local PAs (e.g. via Compositor)

Aggregating & Distributing Presence Information

- Watcher (remote PA)
  - May not know how to weigh information
  - May receive too much and/or too detailed information
  - Not allowed by SIMPLE anyway
- (Centralized) Presence Server
  - Requires specification language for aggregation
    - e.g. similar to CPL
  - MUST be trusted!
- Edge Presence Server / Compositor
  - Requires specification language for aggregation
  - Allows for more user flexibility
- Or some combination of the above...

Approach: Local Coordination

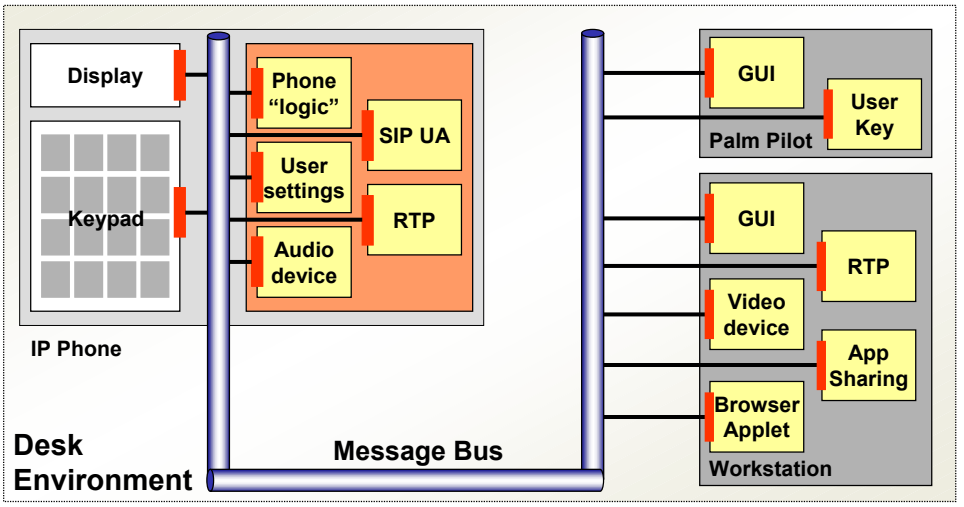
- Background: local communication environment in place
  - Mbus links various personal and desktop devices and applications

Message Bus

Mbus is a local message-oriented coordination mechanism for component-based systems.

- Multiparty peer-to-peer communications
  - Automated peer and liveness detection
  - Secure, reliable information exchange
  - Flexible addressing (unicast, multicast, broadcast)
  - Lightweight, scalable
- Packages for multimedia conferencing
  - Media and device control
  - Call and conference control
  - Presence status

Integration of User Devices



Approach: Local Coordination

- **Background: local communication environment in place**
  - **Mbus links various personal and desktop devices and applications**
- **Extend to collect information from multiple local sources**
  - **Use Mbus to convey presence status (presence profile)**
  - **Choose one device as the user's presence proxy (≈ compositor)**
  - **Collect input and locally decide which "presence state" to export**
- **Also: locally distribute state received from other parties**
- **Talk to an external compositor / presence server**
  - **To integrate state information from non-local entities**
  - **To provide state independent of the operation and availability of personal or desktop appliances**

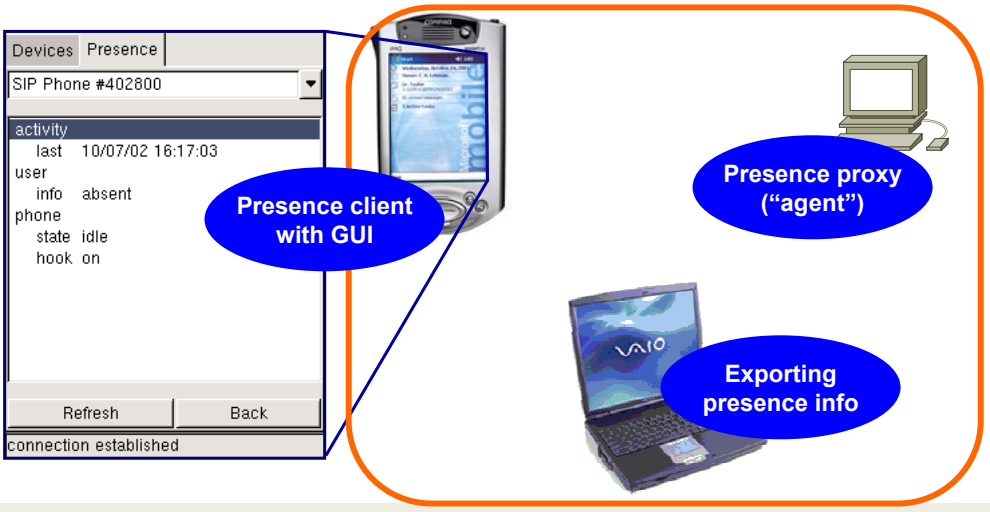
Mbus Presence

- Join several user devices / applications in an Mbus domain
  - Define an Mbus presence package
  - PIDF data + extensions for more fine-grained information
- Designate one entity as a Presence Proxy (= Compositor)
  - Aggregates presence state from all other entities
  - Combine Mbus presence information and convert to SIMPLE
  - Uses SIMPLE to talk to other users
  - Retrieves state of others and distributes it locally
- Locate and integrate new entities
- Monitor neighbor state (liveness, “presence”)

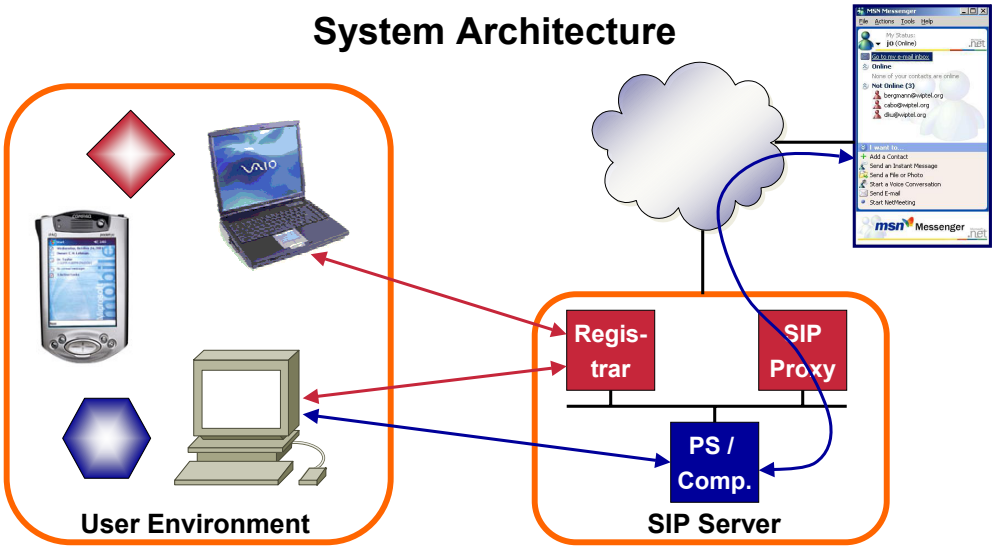
Presence Aggregation Language (PAL)

- Borrowed from CPL
- Source identification
  - Mbus address attributes
  - SIP URI
- Currently, rather simple aggregation approach:
  - Weighing factors for various inputs
    - Per input type (e.g. idle time, powered on)
    - Per device class (e.g. workstation, laptop, ...)
  - Define thresholds for discrete state information
  - Map combinations (simple formulas) to PIDF and other attributes
- Simple text notation for testing
  - Will get more sophisticated as we learn
  - Selecting distribution yet to be integrated

Sample Implementation



System Architecture



Conclusion

- Reasonably accurate presence state may require integration of information from many sources
- Aggregation of presence state best done close to the user
  - Mbus may provide an efficient infrastructure for collection
- User knows how to aggregate information properly
- Keep user in control
- Presence Aggregation Language (PAL) as means for describing aggregation functions