DISTRIBUTED SERVICE DEVELOPMENT IN PANS

Miklós Aurél Rónai, Kristóf Fodor, Gergely Biczók,
Zoltán Turányi, András Valkó

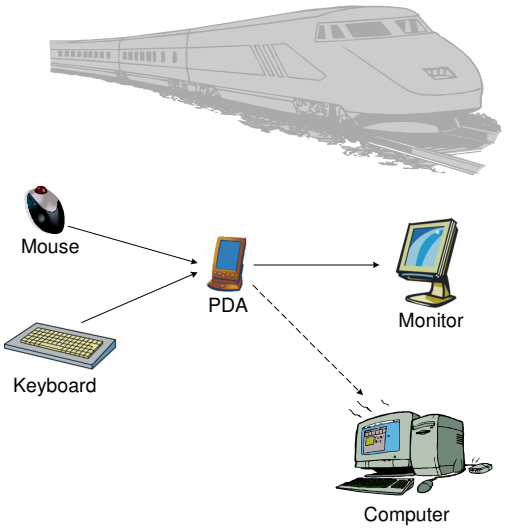
Miklos.Ronai@ericsson.com

ERICSSON
TAKING YOU FORWARD

Contents

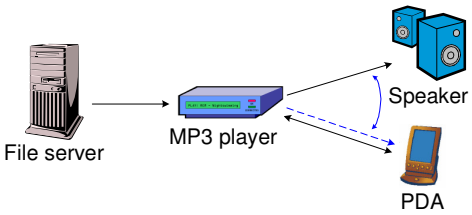
- PAN service scenarios
- Goals in PAN service development
- Proposing a middleware solution -> MAIPAN
- Architecture of MAIPAN
- Access control in MAIPAN
- API example

Scenario 1 (WORKING)



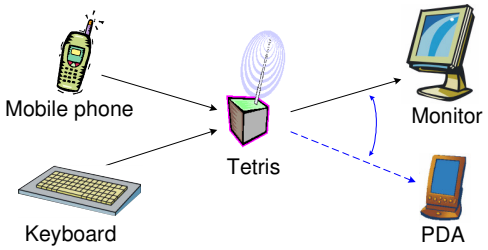
- Keyboard & mouse built in the seat of the train
- PDA
- Monitor built in the seat of the train
- Computer at office

Scenario 2 (MP3-PLAYER)



- MP3 files are located on the file server
- PDA controls the mp3 player
- MP3-player decodes the mp3 files
- MP3-player sends the audio stream to the speakers

Scenario 3 (TETRIS)



- Tetris game running on the Tetris-box
- Multiplayer game inputs: mobile's keypad and keyboard
- Monitor is located on the wall of the room

Goals in service development

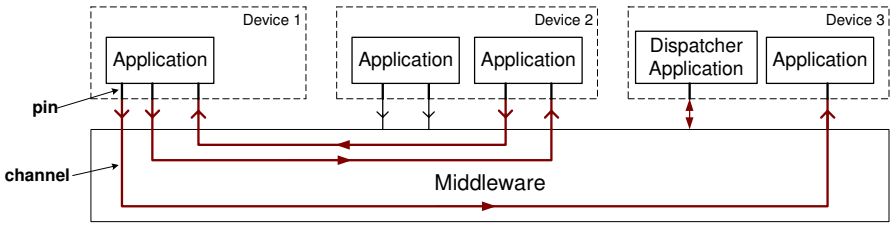
- Easily create and reconfigure PAN services
  - Interconnect applications
  - Reconfigure these connections if necessary
- Hide the physical scatteredness and dynamically changing configuration of the PAN
- Present PAN capabilities as a single computer
- Provide uniform API for service developers

Middleware for PANs

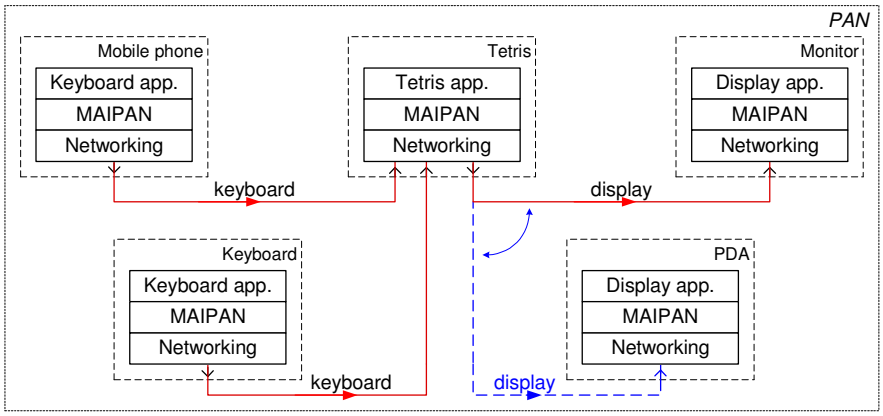
- Running under the applications
- Provides various services for applications
- Takes over some responsibilities of the applications (e.g., access control)
- Ease the development of PAN services
- Can span over multiple layers
- We propose Middleware for Application Interconnection in PANs (MAIPAN)

MAIPAN: Basic Concepts

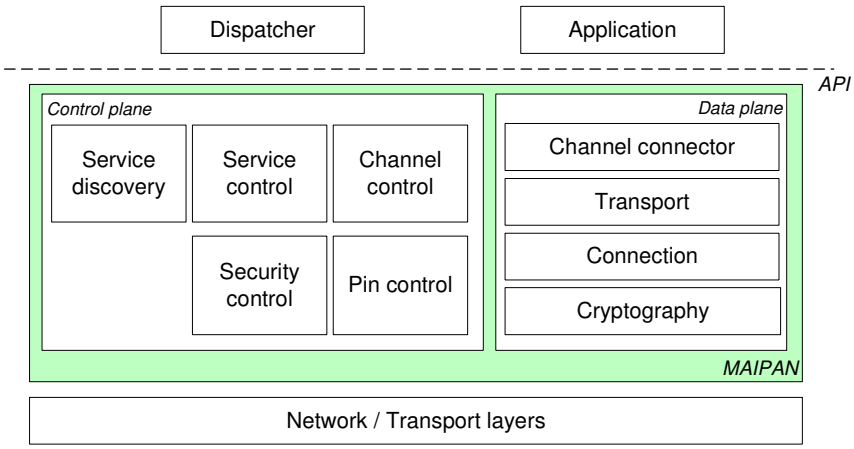
- Pins: outlets of applications
- Channels: interconnect pins
- Session: group of channels that are necessary for a given service
- Dispatcher application: PAN configurator



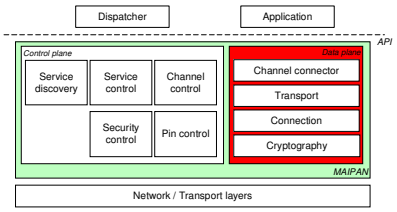
Scenario 3 - Detailed



MAIPAN Architecture

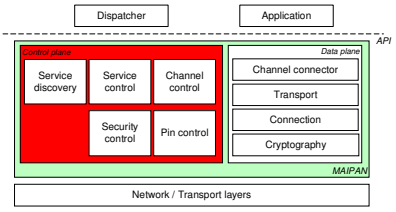


Data Plane



- Channel connector layer:
  - Forwards data from pins to adequate channels
- Transport layer:
  - Create packets
  - Different transport modules with different functions (e.g., flow control, reordering, automatic retransmission, QoS)
  - Applications should indicate the right modules for each of their pins
- Connection layer:
  - Registers the remote endpoints of channels
  - Forwards packets based on this registry
- Cryptography layer:
  - Encrypts / decrypts packets

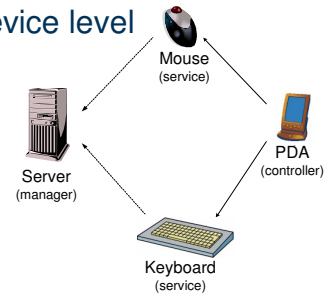
Control Plane



- Service control:
  - Registers local services
  - Sets access rights for services
- Service discovery
- Channel control:
  - Manages the channels & sessions
  - Changes focus on pins
- Pin control
  - Answers on channel creation requests
  - Instructs the layers of the data plane
- Security control
  - Registers service access rights
  - Manages device authentication

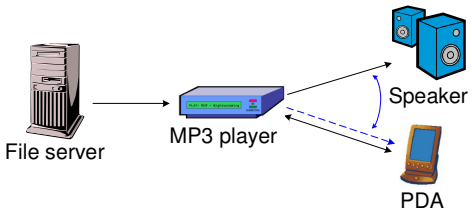
Access Control in MAIPAN

- Functions defined and handled on device level
- Basically service access rights are set locally
- Special case: access rights may be delegated to *managers*
- Accessing a service:
  1. Control entity sends a request
  2. Security control checks whether the control entity has adequate rights to access the service
  3. If the right management is delegated to one or more manager then the manager is / managers are asked
  4. If access is granted to the control entity then the requested operation can be carried out

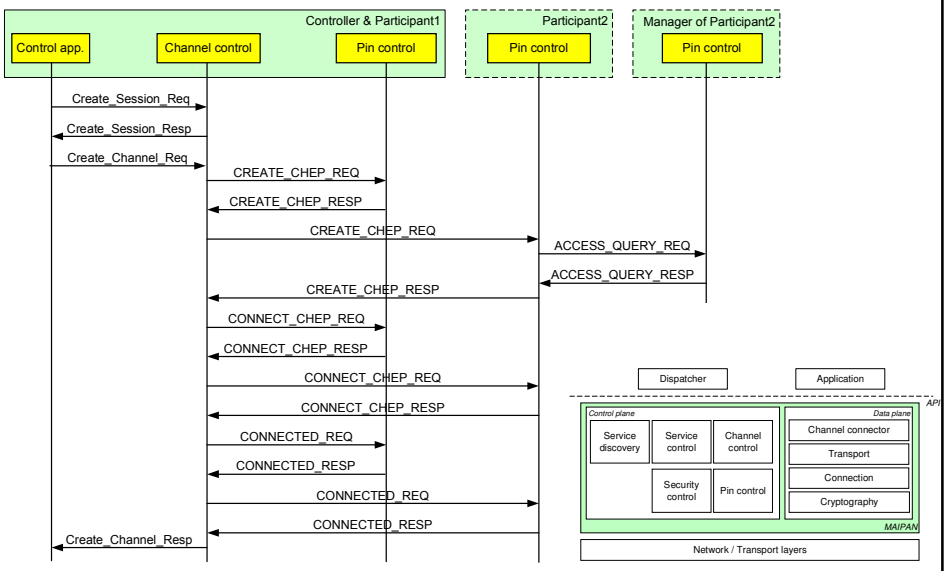


Focus Change

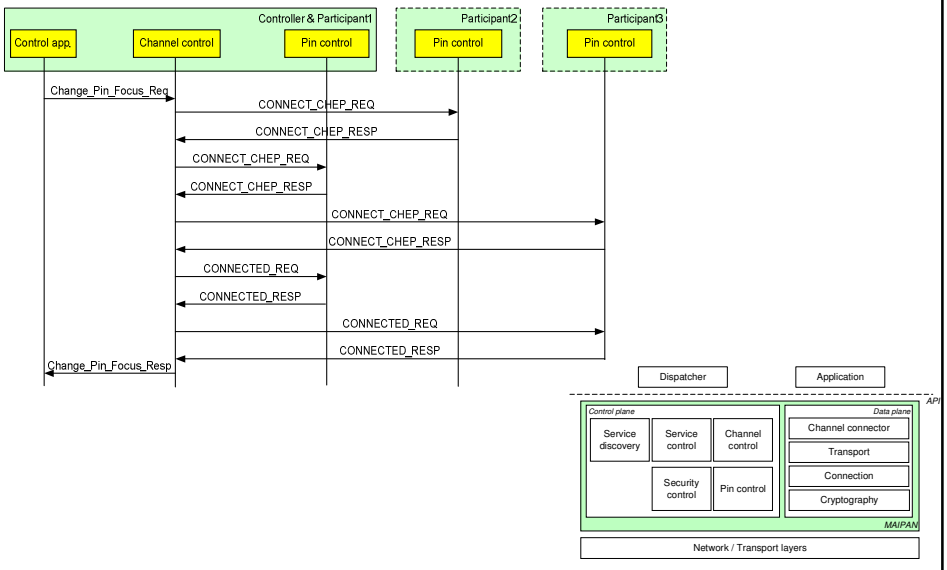
- Like ALT-TAB switching in MS Windows
- Redirecting the output of a service or changing the input source of a service
- Doing it without restarting the service



Example: Creating a Session



Example: Focus Change



Conclusion

- Proposing MAIPAN for PAN service development, to
  - Easily create and reconfigure PAN services
  - Hide the physical scatteredness and dynamically changing configuration of the PAN
  - Present PAN capabilities as a single computer
  - Provide uniform API for service developers
- MAIPAN main features are:
  - Session and channel creation and reconfiguration;
  - Access control and data encryption;
  - Focus change;

ERICSSON

TAKING YOU FORWARD