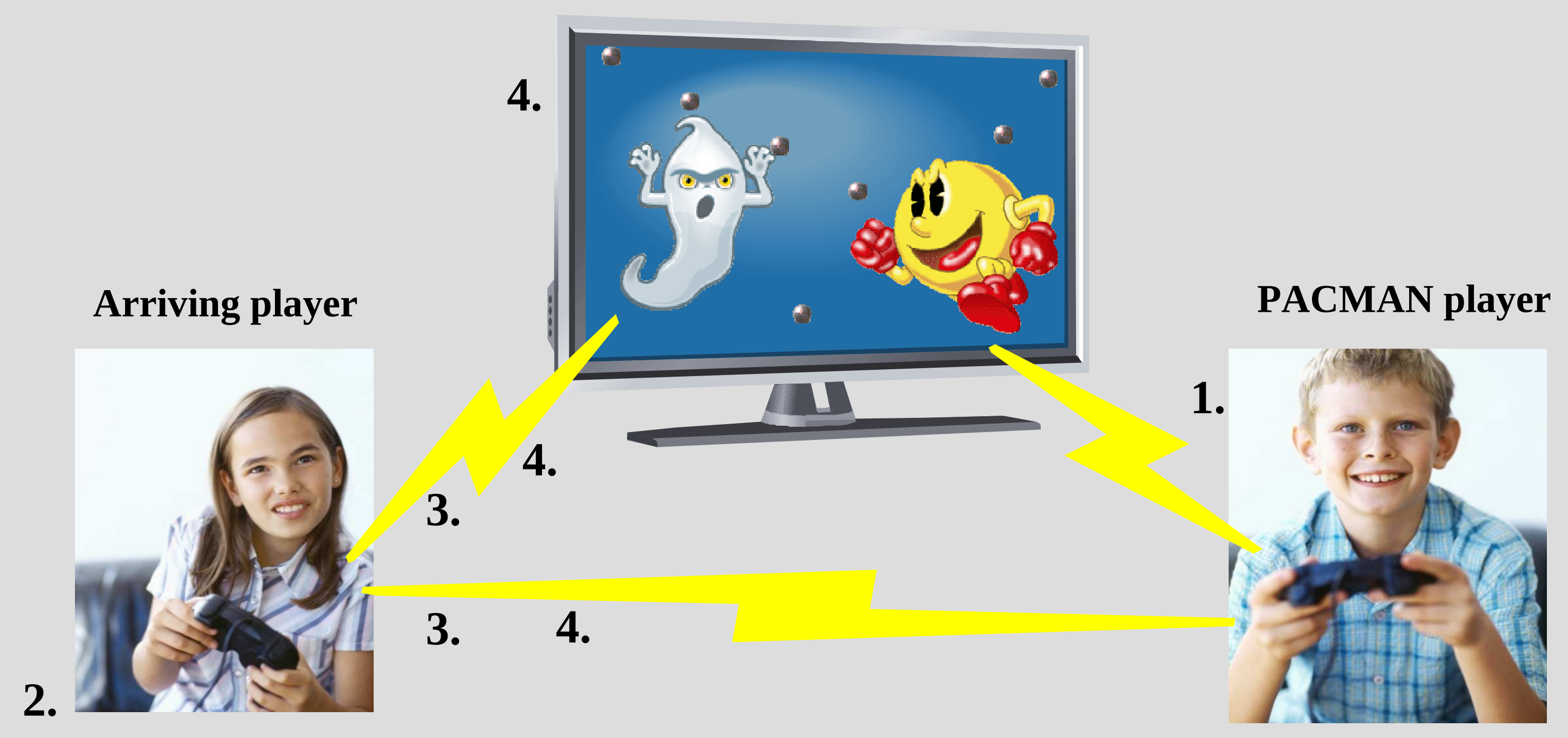
Open Pervasive Environments for migratory iNteractive services

Anders Nickelsen, Rasmus Løvenstein Olsen, Aalborg University, Denmark
Carmen Santoro, ISTI-CNR, Pisa, Italy

http://www.ict-open.eu

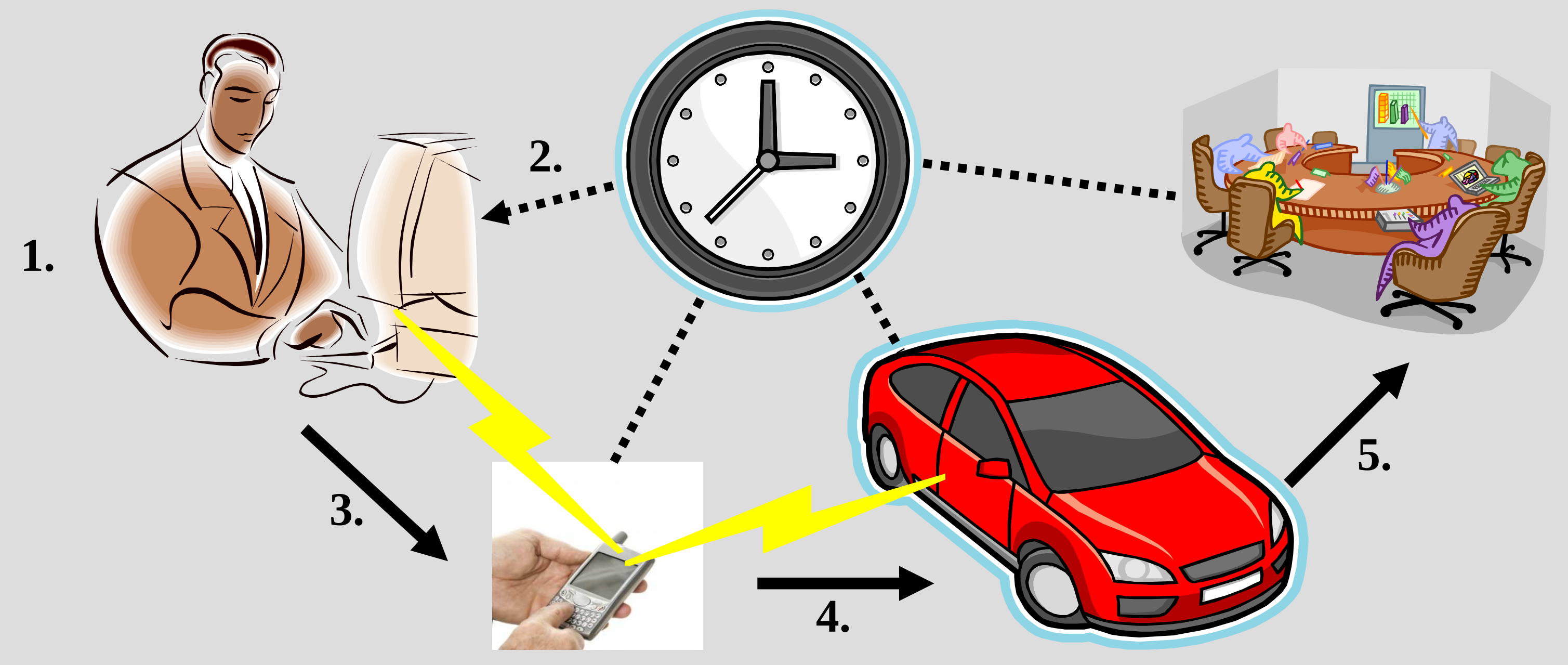
Migration = Device Change + Adaptation + Continuity

Migrating games



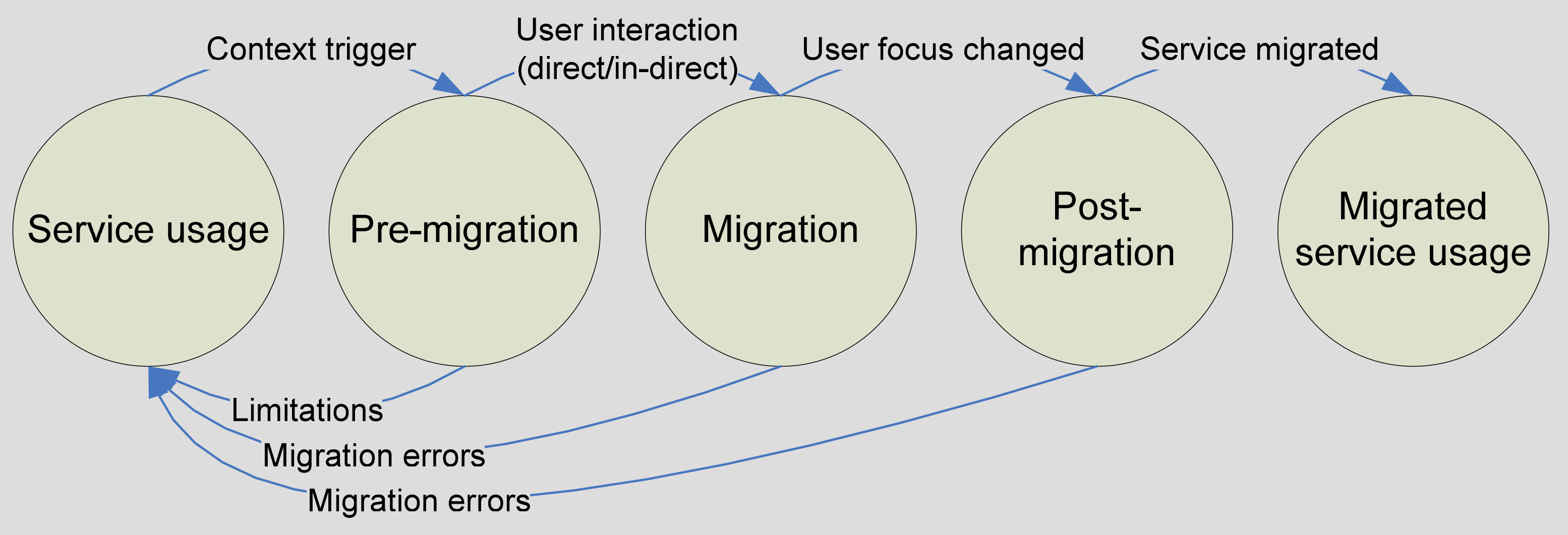
1. Peter is playing a game of Pacman projected on a large screen. He is using his mobile phone to control Pacman.
2. Another user, Laura, enters the room and wants to join the game.
3. The system has become aware of the possibility for a migration and is aware that Peter and Laura knows each other and likes to play games together.
4. So Laura initiates a migration process of the game from her mobile phone, which she intends to use as control device of a ghost instead of the game.
5. The system and game executes the migration, and Laura can now play the game with Peter, using another game controller and through an interface adapted to such device.

Online registration



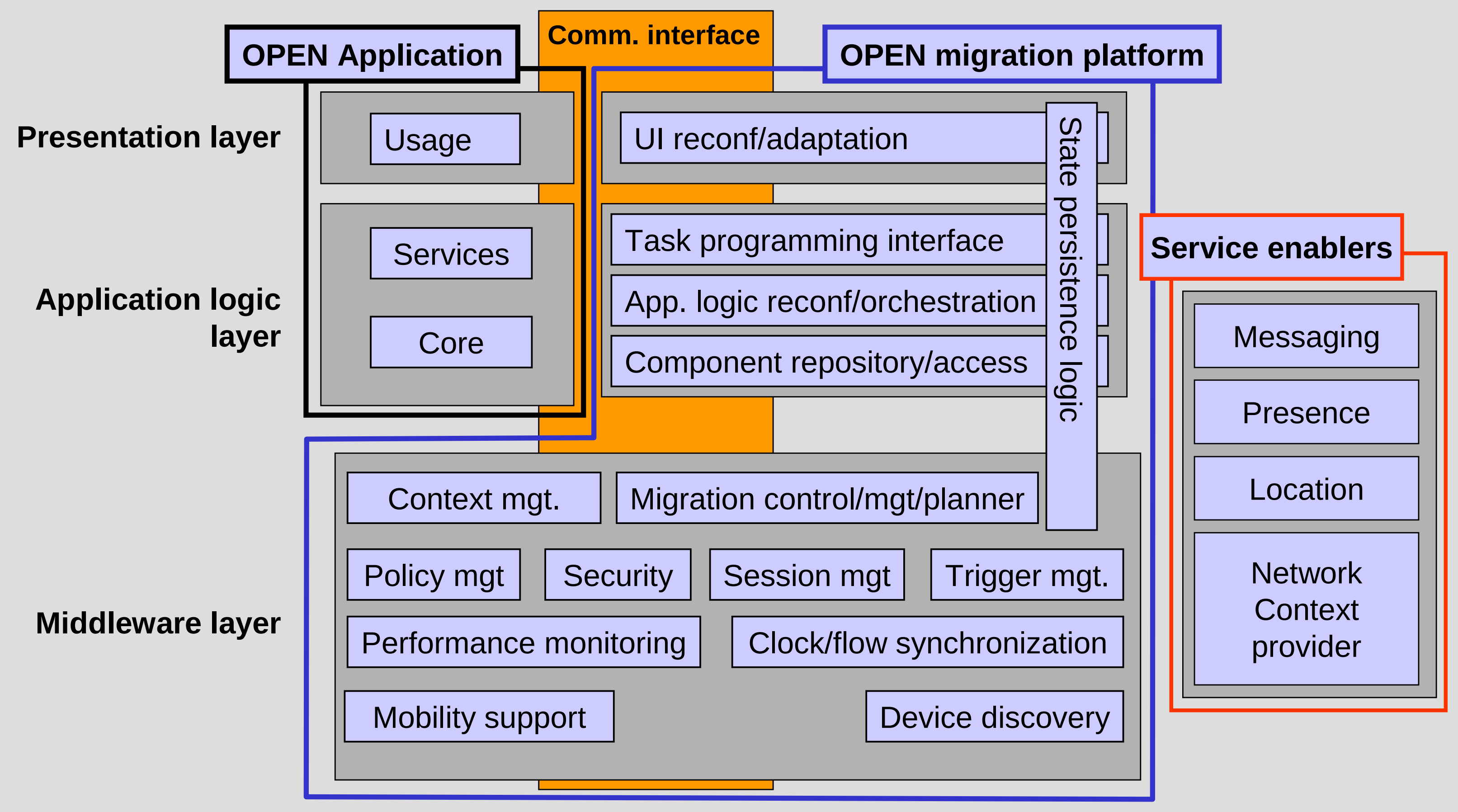
1. John is sitting at home in the morning browsing the internet on his desktop. He is planning to go on vacation and would like to buy a new camera, and wants to search for a bargain at ”e-Bet” website.
2. He needs to register, and start doing this, when his calendar indicates he needs to hurry to his office for a meeting.
3. He triggers a migration of the website registration to his PDA, from where he continues the registration for the bet.
4. As he reaches his car, he has finished the registration process, and is ready to put a bet. The PDA now shifts to vocal input for accepting Johns betting.
5. John continues interacting with the betting system now through the vocal interface and eventually gets the camera for a good price.

The migration proces



1. **Initially the service is in use**
2. **Some external event triggers the system to initiate migration**
   - Context has changed (migration is automatically proposed to the user)
   - User initiates a migration (user-initiated migration)
3. **The system enters a pre-migration phase:**
   - Discovery and selection of possible entities to migrate to
   - Establishment of connection to entities
   - Ensurance that QoS requirements can be met
4. **Once an entity has been selected as migration target, the migration can take place**
   - Startup of target service/application
   - Transfer of application state
   - Adaptation to the new device
5. **At some point in time, the user will have to change focus as interaction device is changing**
6. **When user has changed focus, the ”old” application/service may be shut down or proceed in a different mode of operation.**
7. **The user may now use the application in a fully migrated state**

Functional architecture



• **Presentation layer –** Includes support functionality for reconfigurable multi-modal interfaces, enabling the users to interact with various types of services (at home and on the move) on different devices in different service environments.

• **Application logic layer –** Provides an service platform, which allows service components to be dynamically reconfigured as context, device capabilities and migration allows.

• **Middleware layer –** OPEN middleware will support the migration process, by several network functionalities, such as trigger management to ensure the migration is triggered at the right time, synchronization for data streams when being migrated, device and service discovery, access to context through a context management system and many more required to ensure a reliable, secure and efficient service/application migration.

About

The OPEN project is a multidisciplinary consortium combining the expertise of three technological world leaders, three well known research organizations, and one SME. The partners are as follows CNR-ISTI (Pisa, Italy), AAU (Aalborg, Denmark), NEC Research Center (Germany), SAP (Germany), Vodafone (Milano, Italy), Arcadia (Italy), Clausthal University (Clausthal, Germany)

References

[1] http://www.ict-open.eu, Jul 2008
[2] OPEN D1.1: Requirements for OPEN service platform, May 2008
[3] OPEN D1.2: Initial OPEN service platform architectural framework, Sep 2008