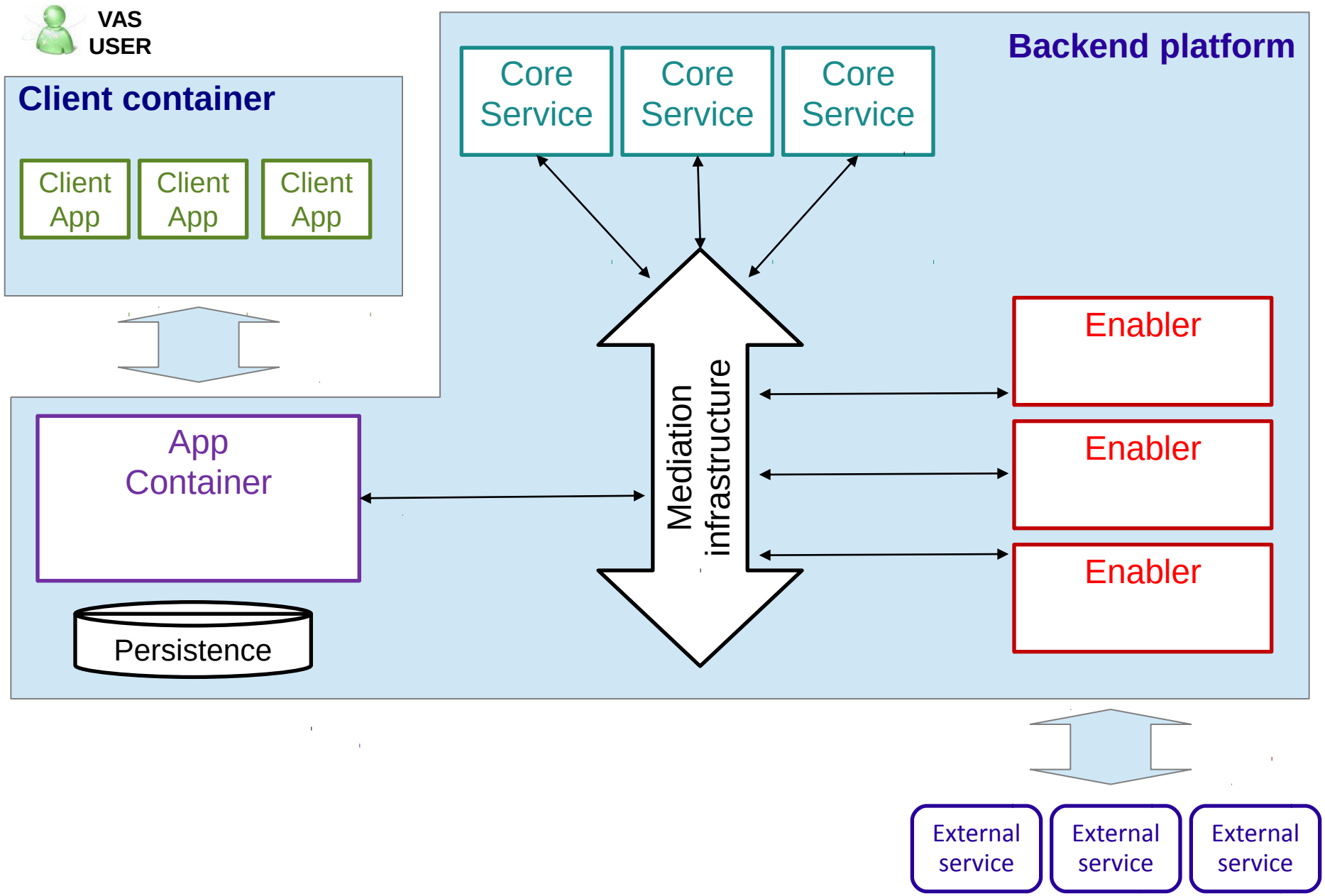
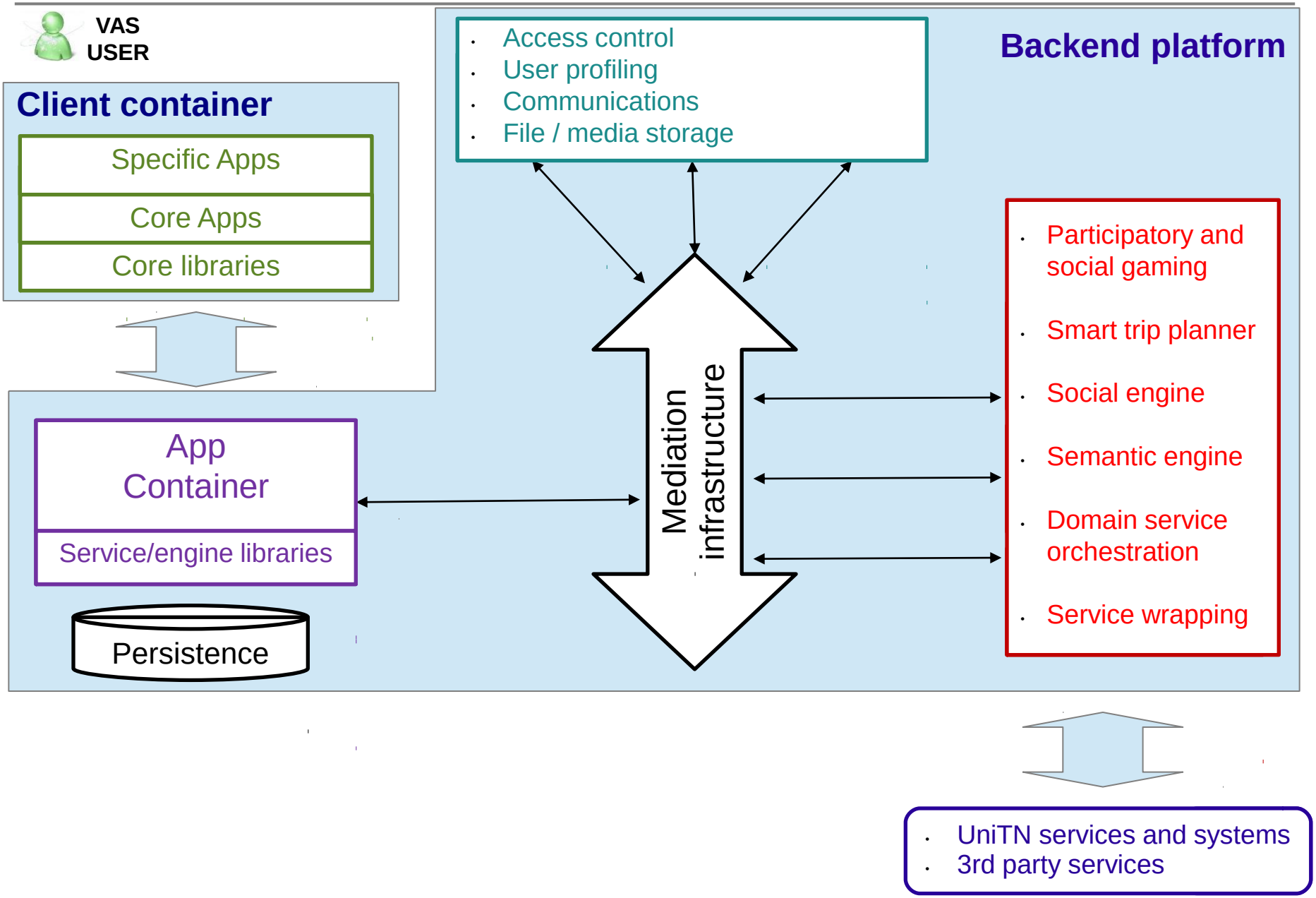
Platform Architecture Overview

- **Platform overview**
- How-to example
- Platform components detailed

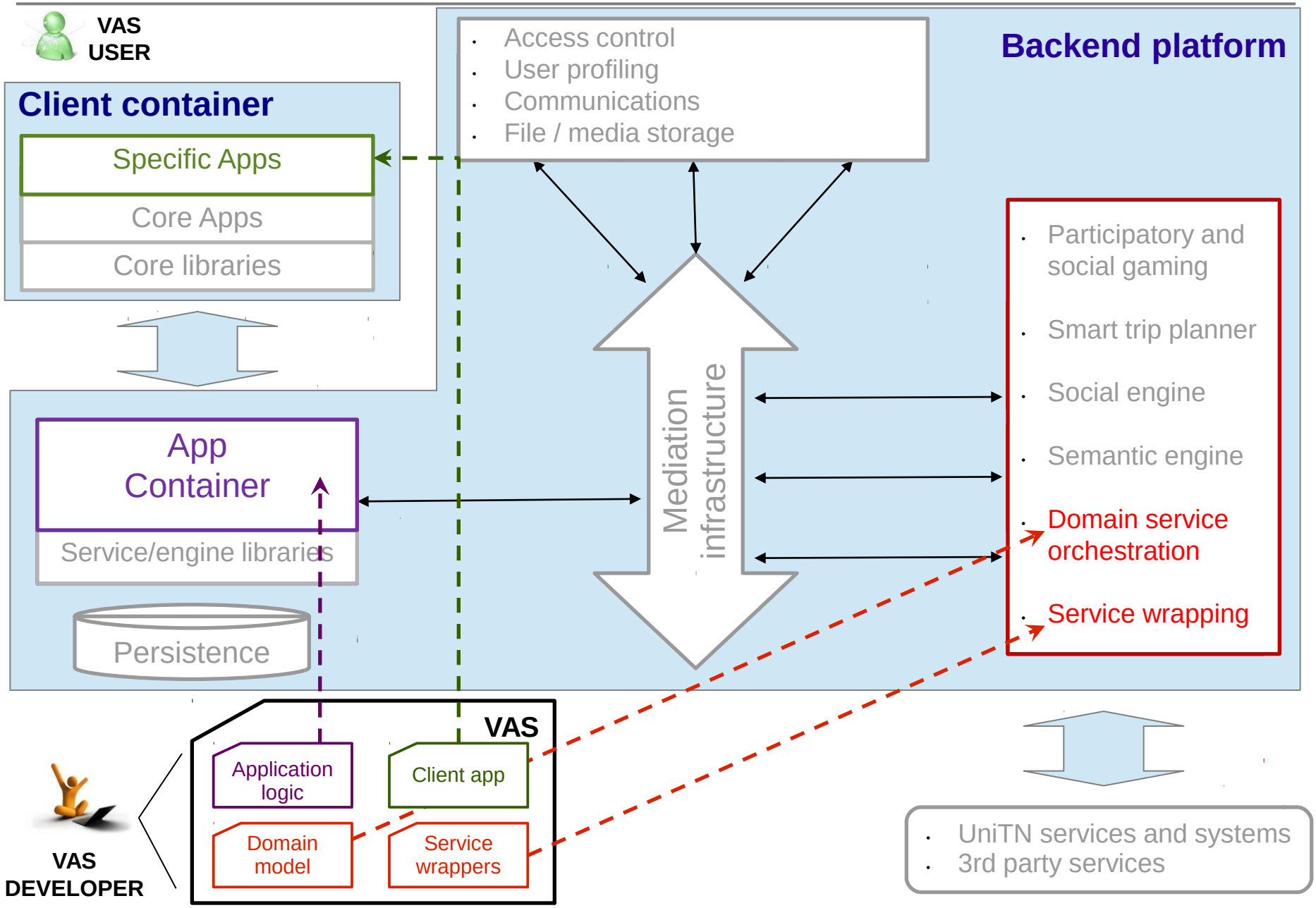
Architecture: overall



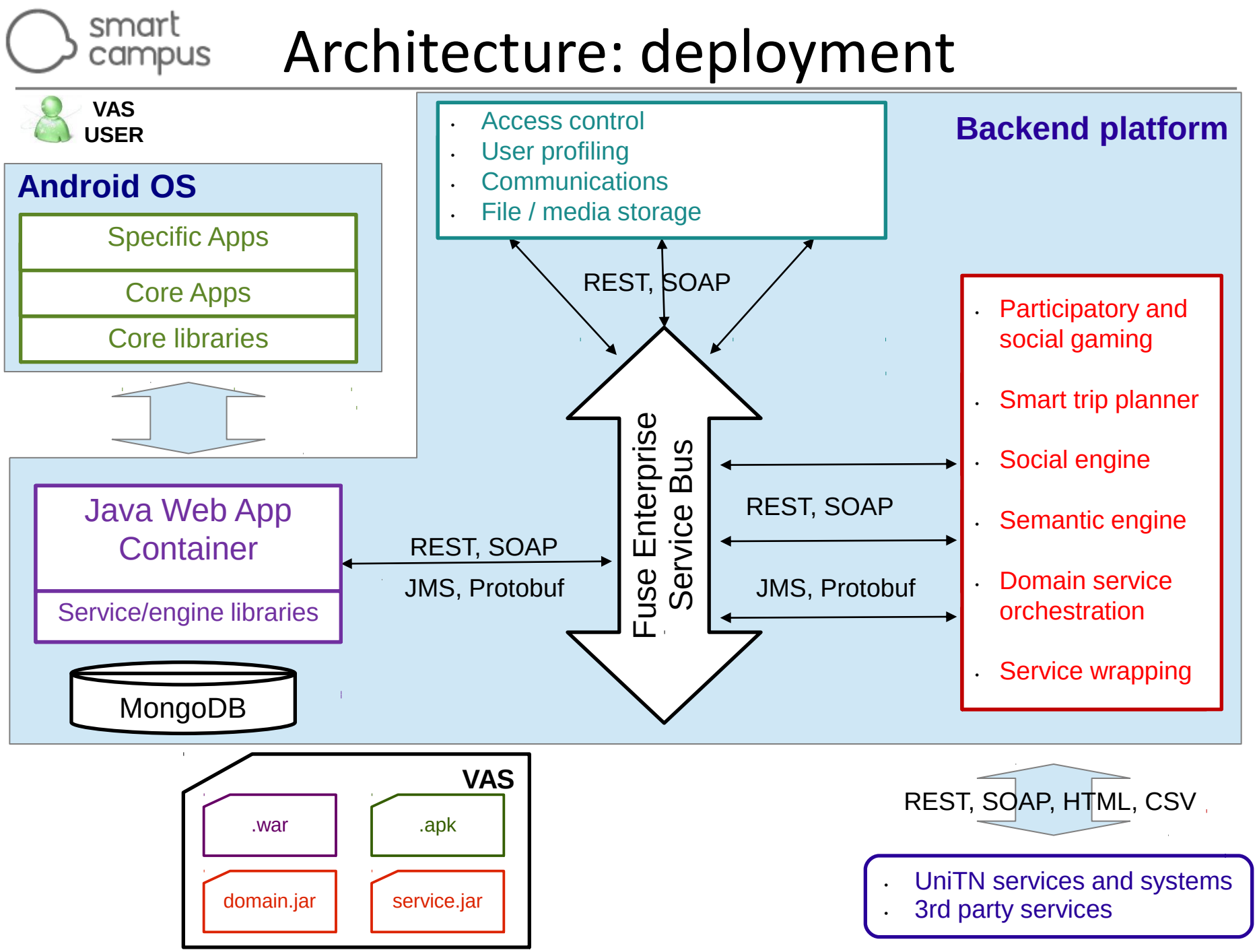
Architecture: concretely



Developer perspective



Architecture: deployment

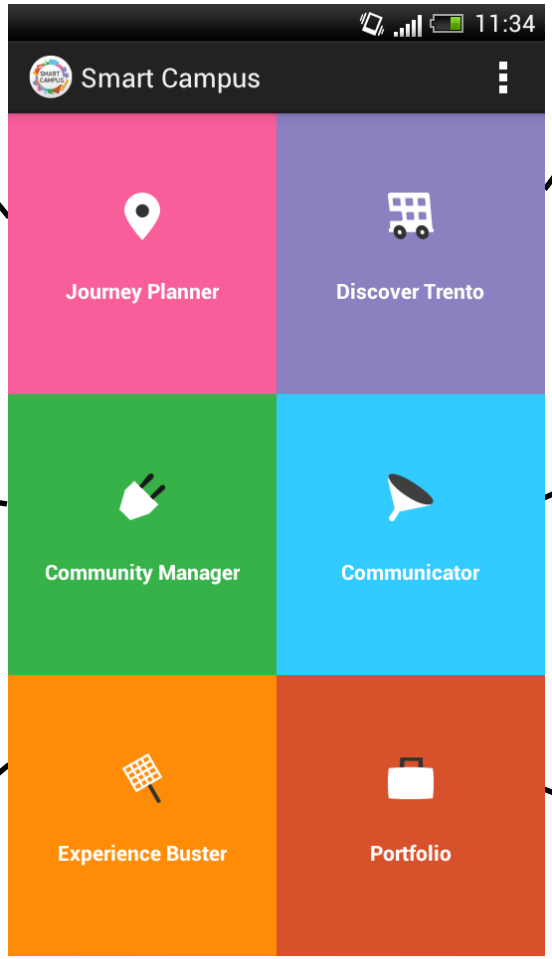


Existing apps

Plan your **trips** in Trento, and monitor your recurrent trips

Manage your **communities**, share and follow relevant topics

Capture your **experiences** and share with your friends



Discover interesting events, places and itineraries in Trento

Organize the **messages** from the campus, the city, and the other apps

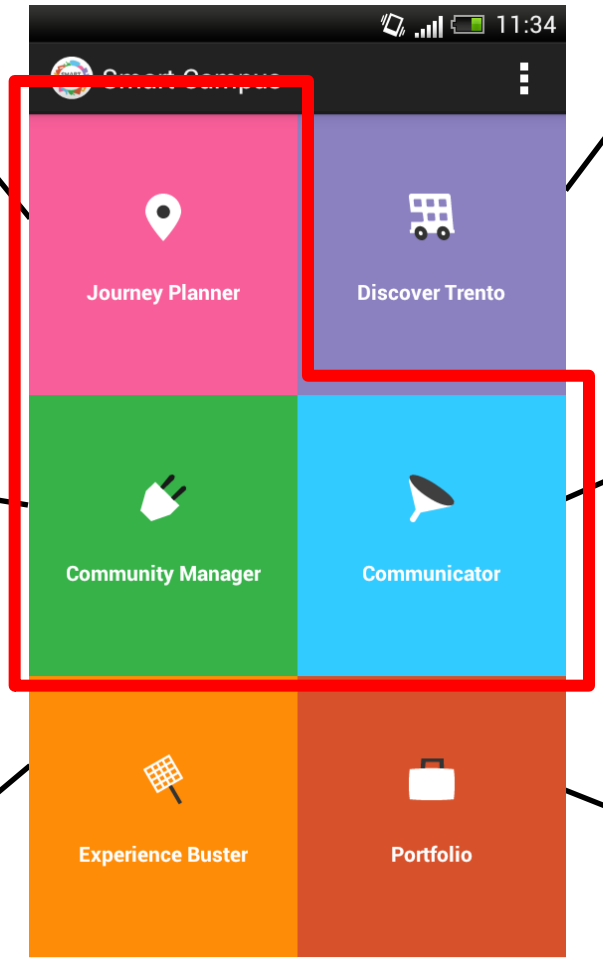
Present **yourself** by combining certified infos from Uni with your own content

Existing apps: core

Plan your **trips** in Trento, and monitor your recurrent trips

Manage your **communities**, share and follow relevant topics

Capture your **experiences** and share with your friends



Discover interesting events, places and itineraries in Trento

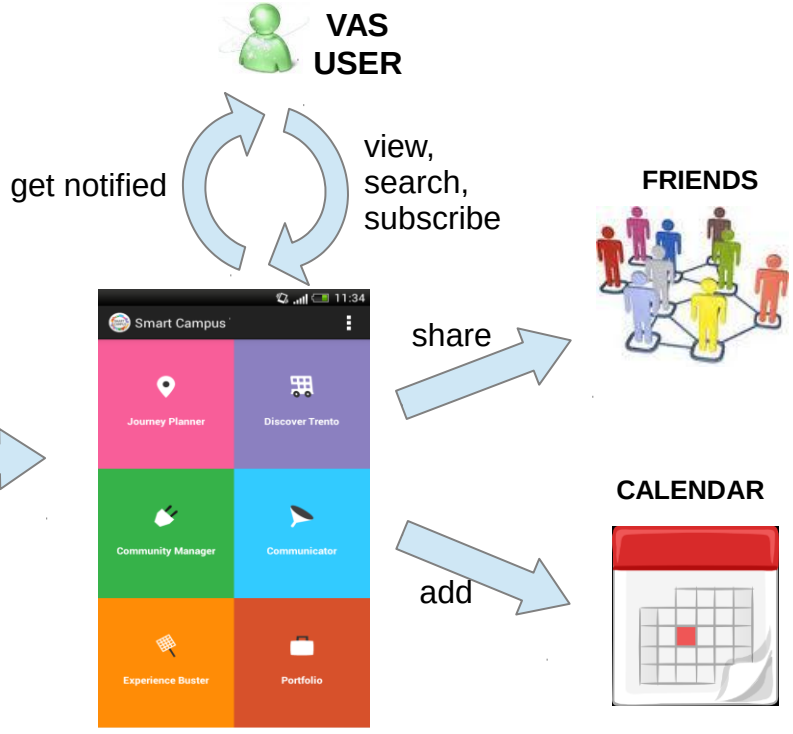
Organize the **messages** from the campus, the city, and the other apps

Present **yourself** by combining certified infos from Uni with your own content

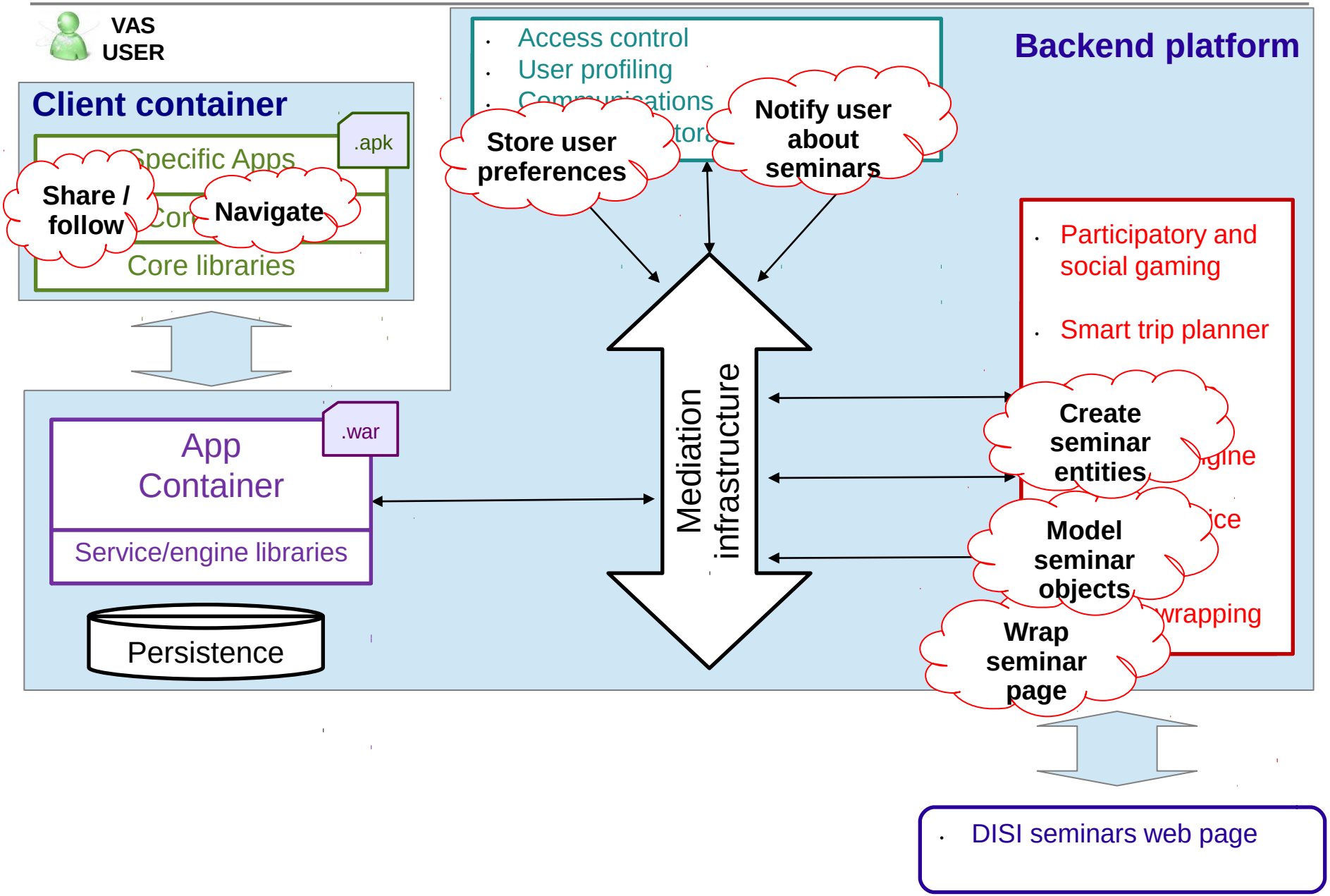
- Platform overview
- **How-to example**
- Platform components detailed

How-to Example: MySeminars

- *Goal: improve my participation to seminars*
  - **Extract** seminar information
  - Capture seminar **life-cycle** (cancellation, changes)
  - View, search, and get notifications for seminars of interest
  - **Personalize** and **share**
  - Be able to **reach the place**
  - **Integrate** with other apps



MySeminars: What?



smart campus

MySeminars: How?

- Extract seminar information
  - Create and deploy DISI seminar page wrapper with **service engine**
- Capture seminar **life-cycle** (cancellation, changes)
  - Model seminar data, evolution, and important changes using **domain engine**
- View and search seminars of interest
  - Store and classify entities using **semantic engine**
  - Store user preferences using **profile service**
- Get notifications about seminars of interest
  - Transform domain changes into user notification and use **communications service** to notify the user
- Personalize and share
  - Create user copies of seminar entities using **social engine**
  - Use **CommunityManager** Android app to share seminars with friends
- Be able to reach the place
  - Use **JourneyPlanner Android app** to plan the trip to the seminar's place
- Integrate with other apps
  - Extend **DiscoverTrento app domain** to integrate seminars with other events
  - Export personalized events to **Google Calendar app**

- Platform overview
- How-to example
- **Platform components detailed**

Enablers: service wrapping

*Goal: provide **continuous** and **reliable** access to external **fragmented** and **heterogeneous** services and information sources*

- Non-standard service formats (e.g., HTML, CSV)
- Format and data validation
- Check detection
- Data caching
- Data updates publish/subscribe

Realization

- Markup language for HTML/SML data scrapping and validation
- Enterprise mashup language for server data aggregation and transformation
- Maven tools for artifact assembly
- Service wrapper container with hot deployment and management console
- APIs for service access and event subscription/publishing

smart campus

Enablers: service orchestration

*Goal: model the evolution of the application **domain** on top of other objects and services, their **operations** and **events***

- Capture object data
- Define simple and composite object operations
- Capture domain evolution triggered by client operations or by domain events
- Execution environment
- Domain persistence
- Domain updates publish/subscribe

Realization

- Specification language for server data aggregation and transfromation
- Maven tools for Java code generation and artifact assembly
- Domain container with hot deployment and management console
- APIs for domain access and event subscription/publishing

Enablers: social / semantic engine

- *Semantic engine: support creation, population,* ***semantic search*** *and reasoning on* ***user****- and* ***community****-tailored knowledge bases containing typed* ***information entities***
  - CRUD operations on entities, their reliationships, and on knowledge bases
  - Semantic entity classification and search
  - Natural language conceptualization
- *Social Engine: provide* ***social networking*** *functionalities over the user and community knowledge bases*
  - User and community management
  - Organizing users in groups
  - Management entity permissions with respect to user groups and communities

Realization

- RESTful service APIs
- Java client library

Enablers: smart planner

*Goal: perform multimodal **trip planning** and analysis with **real-time** transport information support*

- Plan trips supporting variety of transportation means and modes
- Real time alerts about transport information (delays, cancellations) and the consecutive analysis

Realization

- Relies upon the open source Open Trip Planner engine
- RESTful APIs
- Tightly integrated in the Journey Planner application

Core Services

- Authentication and access control
  - Token-based user authentication
  - Externalized sign-in (Shibboleth)
  - Spring-security -compatible back-end library
  - Android account managment provided by SmartCampus app
  - Exposed as Web service and as a Web application
- User profiling
  - Access to public user profiles
  - CRUD operations to provide custom user profiling elements
  - REST API
- Communications
  - Send communications to the user on behalf of an application
  - The notifications are visualized in Communicator application
  - REST API
- File / media storage
  - Create and store files on behalf of the user
  - Sharing / access control using Social Engine
  - REST API
  - Android client library

Thank you