CLAFIS Project
Knowledge Processing Framework and Security Implementations

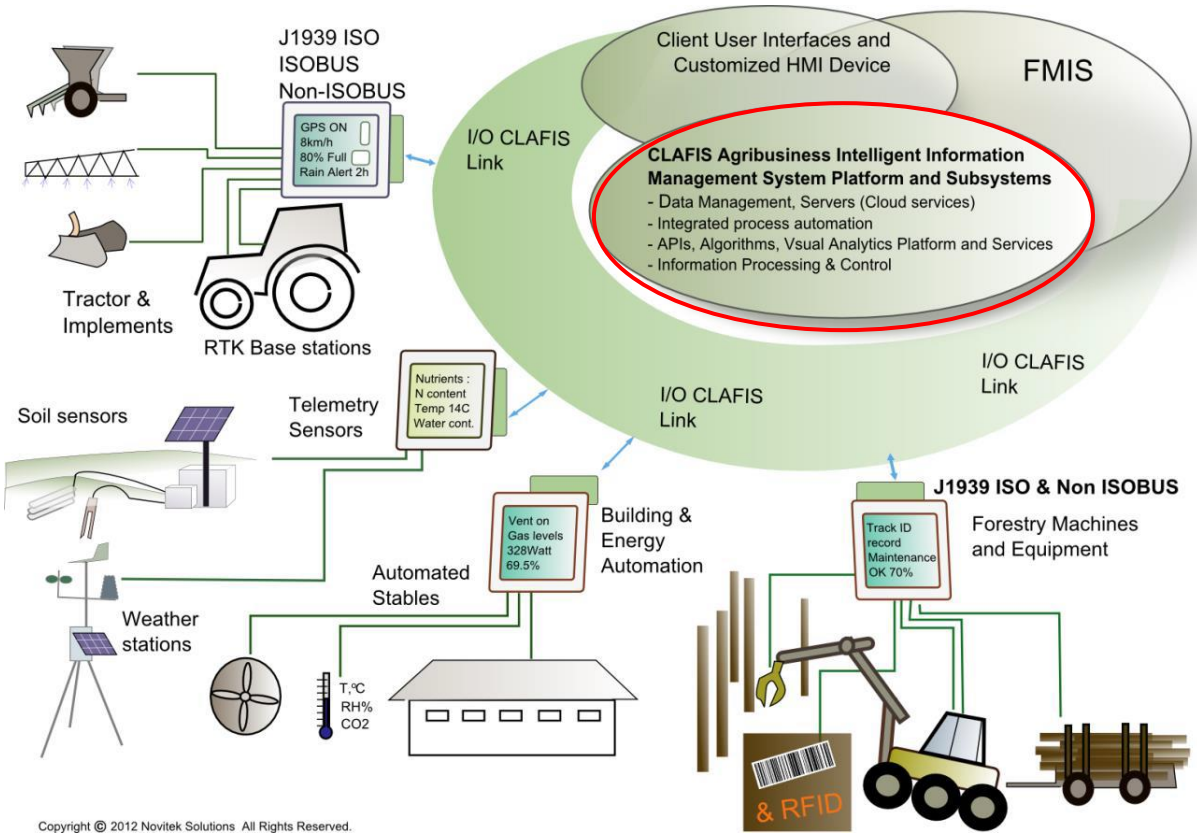
European Project Space

Nov 10th, 2016, Porto

Overview

- Work Package 4 of CLAFIS Project
- Knowledge Processing Framework (KPF)
- Central Authentication and Authorization System (CAAS)
- Conclusion

WP4 of CLAFIS Project



WP4 of CLAFIS Project

- 4A – Analysis of Data and Knowledge sources and Basic Knowledge Modelling
- 4B – Knowledge import, Knowledge extraction
- 4C – Knowledge Management and Processing
- 4D – Integration of Knowledge, openness for new types of knowledge
- 4E – Parallelism and Cloud
- 4F – Computational and Intelligent Signal Processing algorithms, tools and models
- 4G – Security, trust and risk

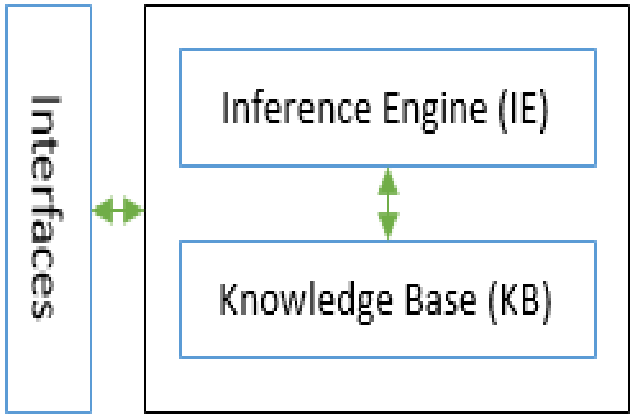
Knowledge Processing Framework – KPF

KPF Motivation

- No development model exists for the knowledge-based systems. [Akerkar]
- Developing a KBS is costly and tedious. [Akerkar]
- Above all, the knowledge is voluminous and continuously changing, so the typical data structure will not be able to completely possess the knowledge. [Akerkar]
- Most KBS development shells claim to provide hooks to connect an external resource to a KBS after it has been constructed. However, the communication among the KBS and external entities requires additional functionality to truly integrate them. [Davis et al.]
- Modern enterprise systems bring together various systems built on different platforms and enable them to communicate with each other. Similarly, knowledge systems will be more attractive if they can be integrated with existing conventional systems. [Syazwan et al.]
- Most descriptions of a KBS architecture are very high-level!

Knowledge Based Systems (KBS)

- Knowledge **Management** (data structure, storage technology, etc.)
  vs. Knowledge **Processing** (reasoning, processing of existing knowledge, etc.).
- Most KBS resemble Expert Systems.
- Main components
  - Knowledge Base,
  - Inference Engine,
  - Explanation Component,
  - Working Memory,
  - Interfaces.
- Expert System Shells or custom KBS.

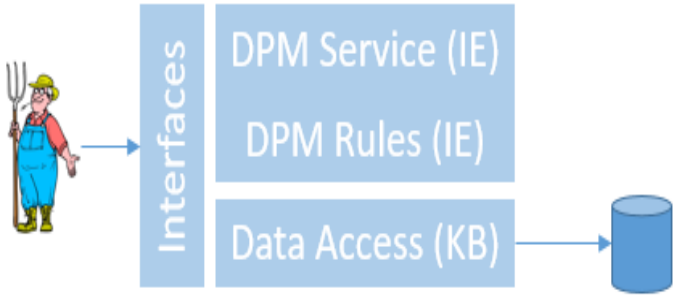


Knowledge Based Systems

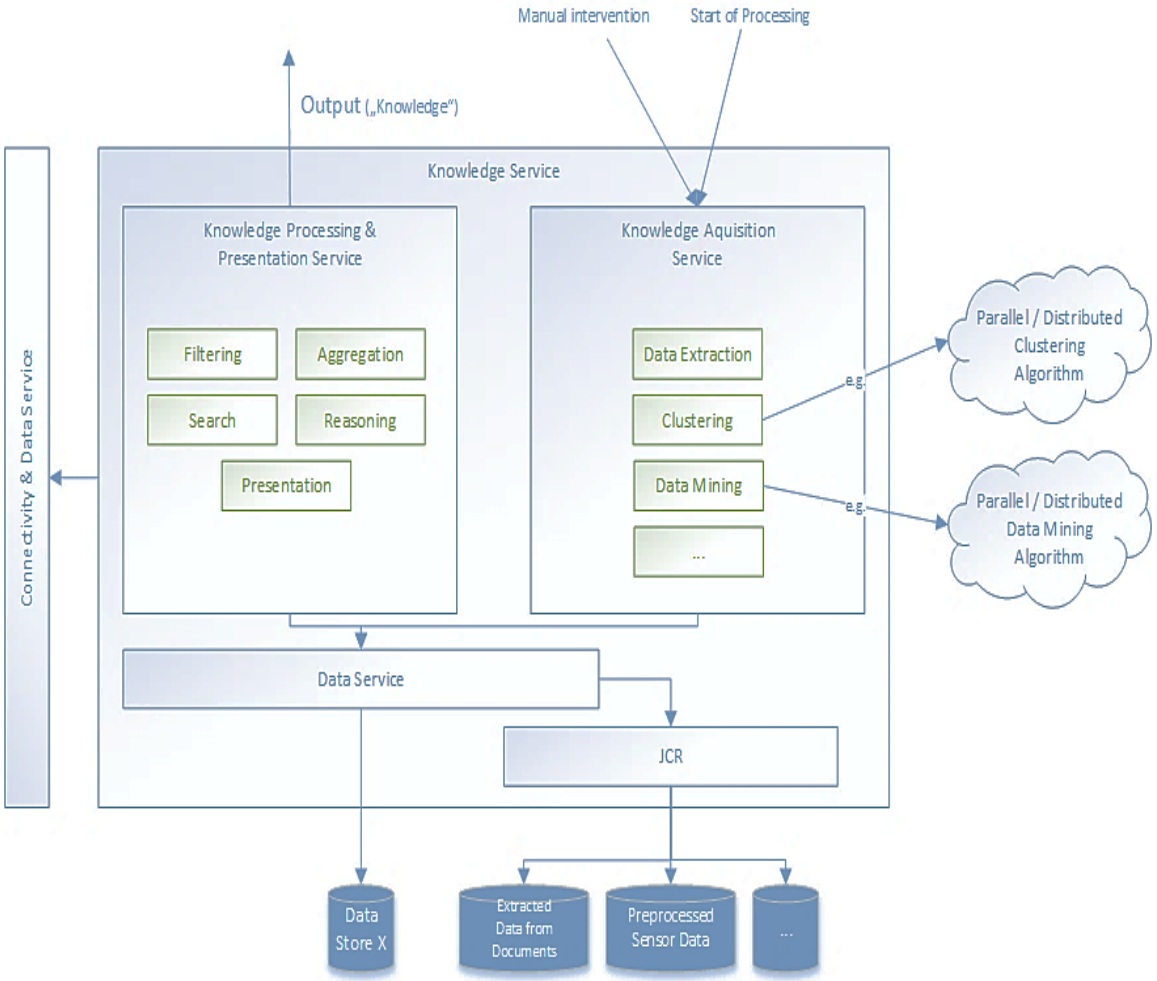
- Functional Requirements:
  - Management,
  - Search,
  - Import / Export of knowledge,
  - Collaboration and Sharing,
  - Personalization and Presentation,
  - Knowledge Processing (reasoning, etc.).
- Non-Functional Requirements
  - Performance and Scalability,
  - Extensibility,
  - Storage and openness to new knowledge,
  - Security,
  - Timeliness & Relevancy.

KPF in CLAFIS

- Crop, Livestock and Forests Integrated System for Intelligent Automation.
- Cloud based Knowledge **Management** (weather data, field history, rules, etc.) & **Processing** (calculation) combined.
- Proof of Concept Knowledge Services:
  - Disease Pressure Calculation
  - Cereal Maturation Calculation
  - Knowledge Documents Import / E:



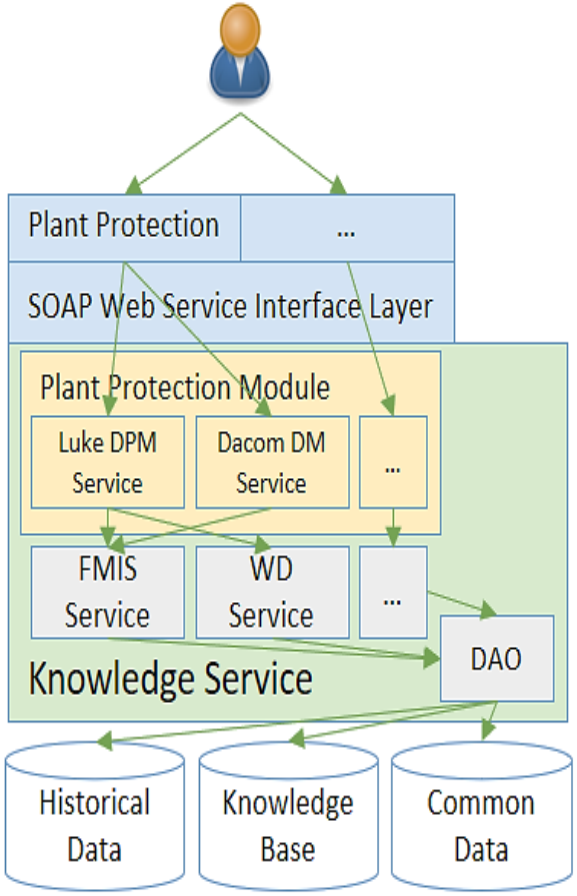
General Architecture of KPF



CLAFIS Framework

- Logical **modules** and **services**.

- Architecture layers:
  - (optional) User Interface,
  - Webservice,
  - Services,
  - Data Access.



KPF – Cloud and KBS

- PaaS, IaaS or SaaS
- Performance and Scalability:
  - Stateless services:
    - Parallelization and physical distribution.
    - Parallel instances.
  - Parallel versions of reasoning algorithms.
- Extensibility:
  - *Dependency Injection* pattern:
    - Configuration via XML.
    - Supporting strong cohesion and loose coupling.
    - Changes without redeployment, configurable.

KPF – Cloud and KBS

- Resilience:
  - Fallback strategies (see *Change of Responsibility*, *Retry,* or *Circuit-Breaker* pattern).
- Security:
  - CAAS.
- Deployment:
  - Java, use Apache Maven.

KPF – Summary

- A more detailed architecture for Knowledge Based Systems conforming to modern Enterprise Systems was developed.
- Design Patterns, usable in this context were identified (see EuroPLoP publication).
- Cloud Platforms ideally support modern requirements of Knowledge Based Systems.
  - Performance,
  - Scalability,
  - Extensibility.

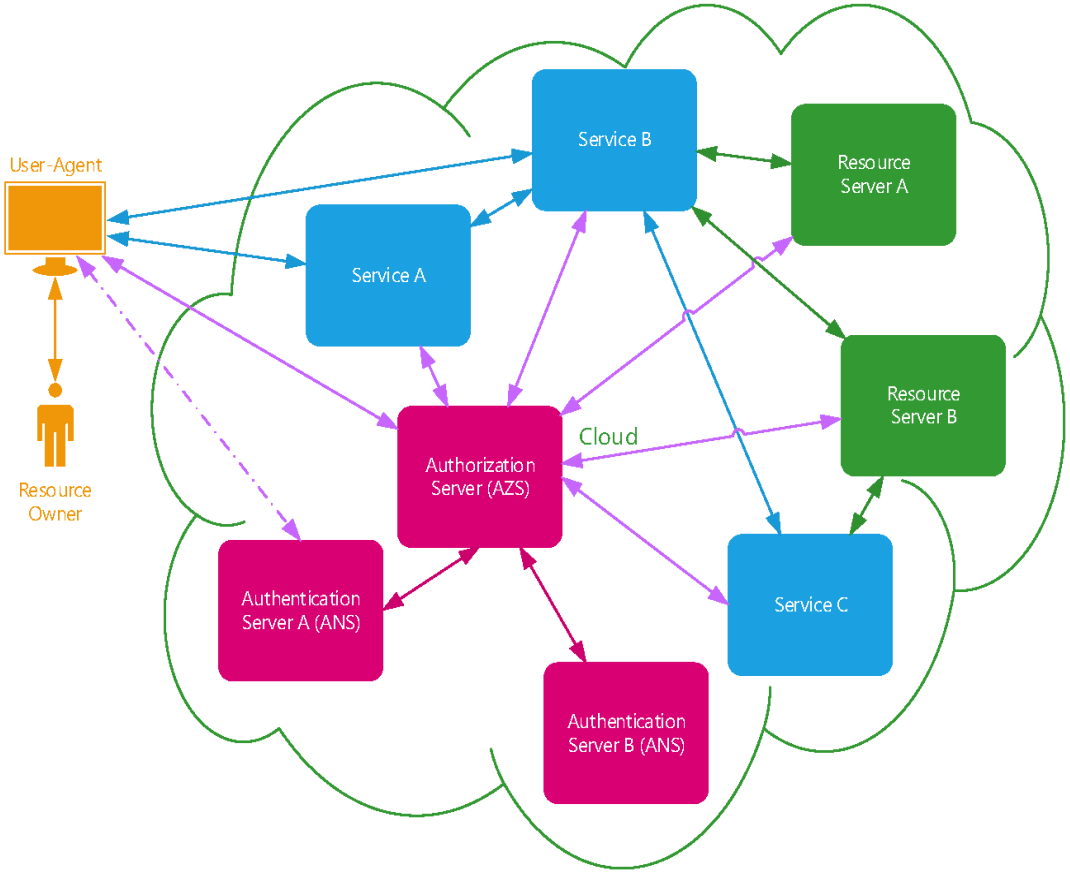
Central Authentication and Authorization Framework– CAAS

CAAS – Introduction

- Security solution for cloud environments
- Configurable for arbitrary domains
- Distinct Authentication Server and Authorization Server
- Authentication based on OpenID
- Authorization based on OAuth2.0

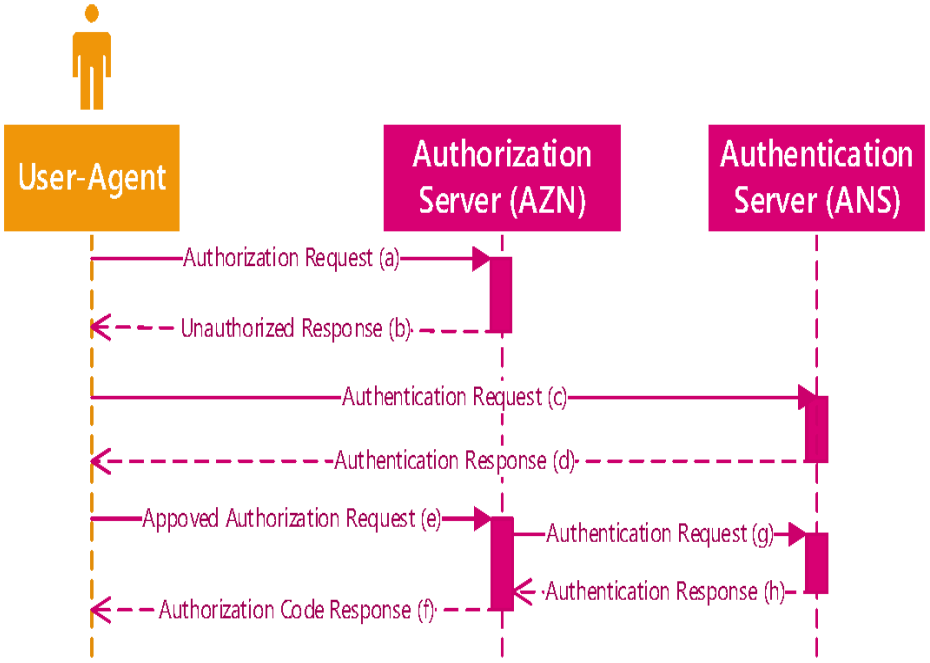
CAAS Introduction (con‘)

- Allows multiple Authentication Server
- Allows multiple Authorization Server
- User delegates permissions over resources to service provider
- Token based communication between servers (Token encryption independent from TLS)

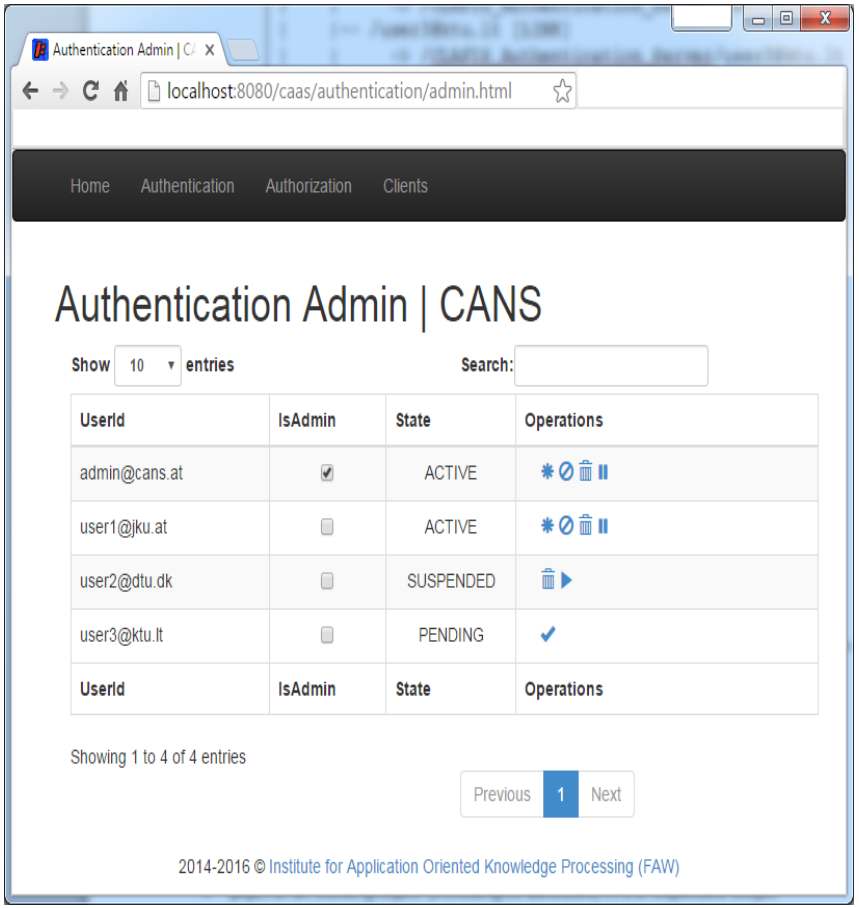


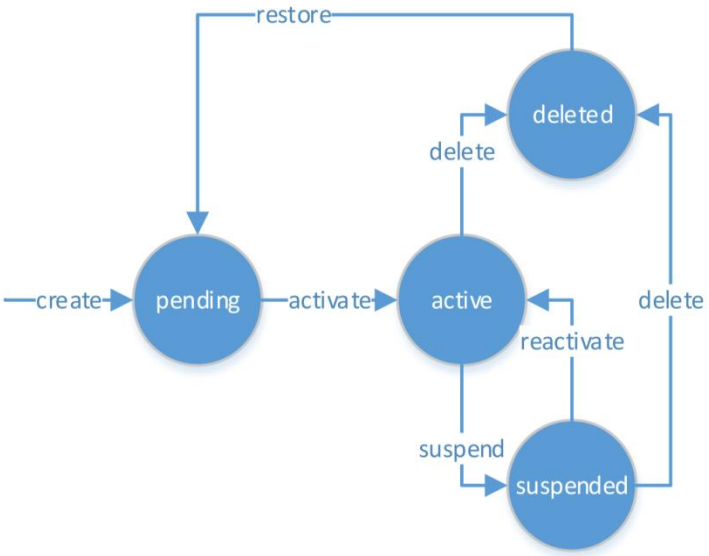
CAAS – Authentication

- Approves the identity of a user for the authorization process
- Every user has to be approved (by administrator or activation mail)
- Only active users can request resources



CAAS – Authentication interface





CAAS – Authorization

- Approves the permissions of an authenticated user
- Resources and users are represented as nodes in the ScopeTree of an Authorization Server
- Permissions are defined using the SCRUDL scheme

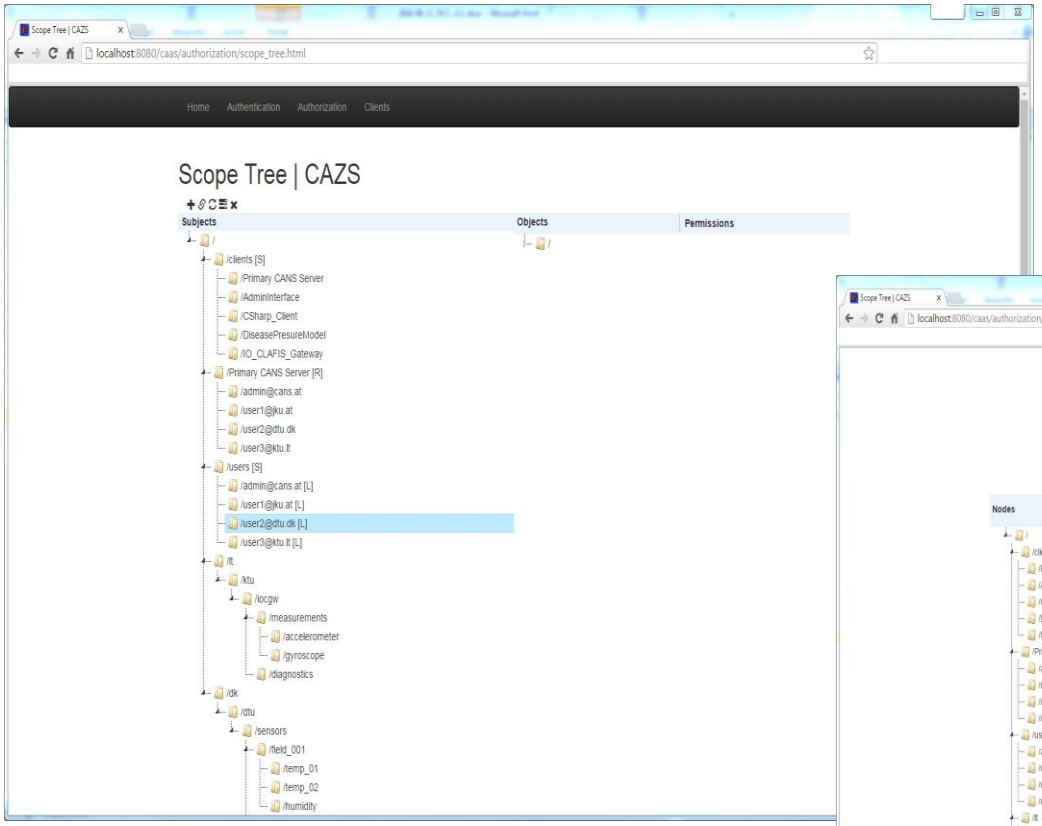
```
permissions := ('S' | '.' | '–')('C' | '.' | '–')
               ('R' | '.' | '–')('U' | '.' | '–')
               ('D' | '.' | '–')('L' | '.' | '–')
```

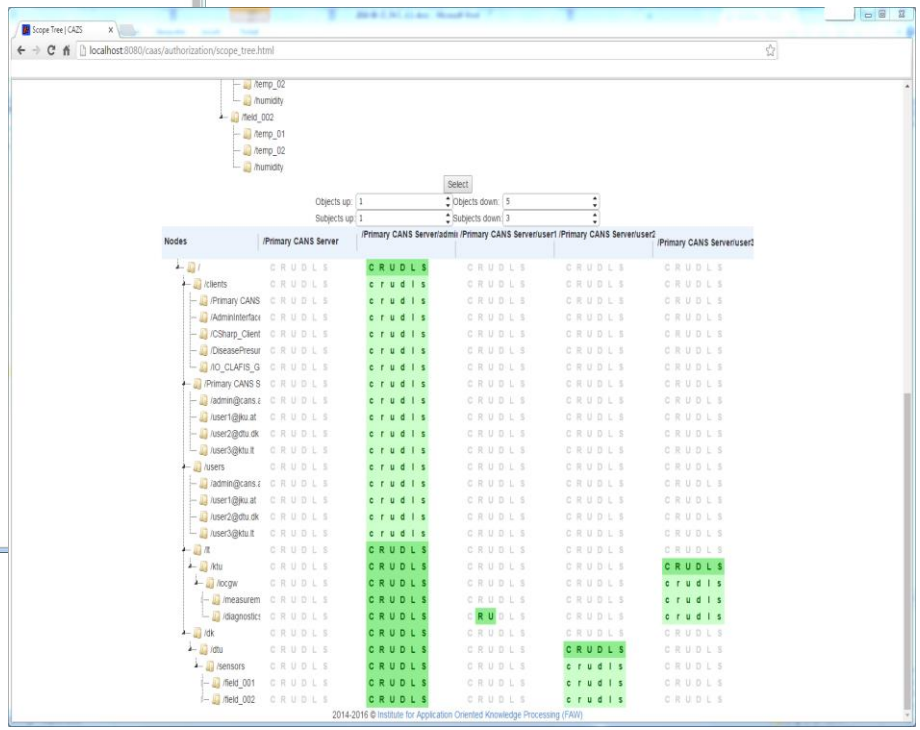
Subject: /CLAFIS_Authentication_Server/user1@jku.at
Object: /lt/ktu/iocgw/diagnostics
Permissions: [..RU..]

ScopeTree:

```
/
|-- /clients
|     |-- /AdminInterface
|     |-- /DiseasePresureModel
|     |-- /IO_CLAFIS_Gateway
|     |-- /CSharp_Client
|     |-- /CLAFIS_Authentication_Server
|-- /users
|     |-- / admin@cans.at [LINK]
|     |      -> /CLAFIS_Authentication_Server/admin@cans.at
|     |-- /user1@jku.at [LINK]
|     |      -> /CLAFIS_Authentication_Server/user1@jku.at
|     |-- /user2@dtu.dk [LINK]
|     |      -> /CLAFIS_Authentication_Server/user2@dtu.dk
|     |-- /user3@ktu.lt [LINK]
|     |      -> /CLAFIS_Authentication_Server/user3@ktu.lt
|-- /CLAFIS_Authentication_Server
|     |-- /admin@cans.at
|     |-- /user1@jku.at
|     |-- /user2@dtu.dk
|     |-- /user3@ktu.lt
|-- /lt/ktu/iocgw
|           |-- /measurements
|           |     |-- /accelerometer
|           |     |-- /gyroscope
|           |-- /diagnostics
|           /dtu/sensors
|           |-- /field_001
|           |     |-- /temp_01
|           |     |-- /temp_02
|           |     |-- /humidity
|           |-- /field_002
|           |     |-- /temp_01
|           |     |-- /temp_02
|           |     |-- /humidity
```

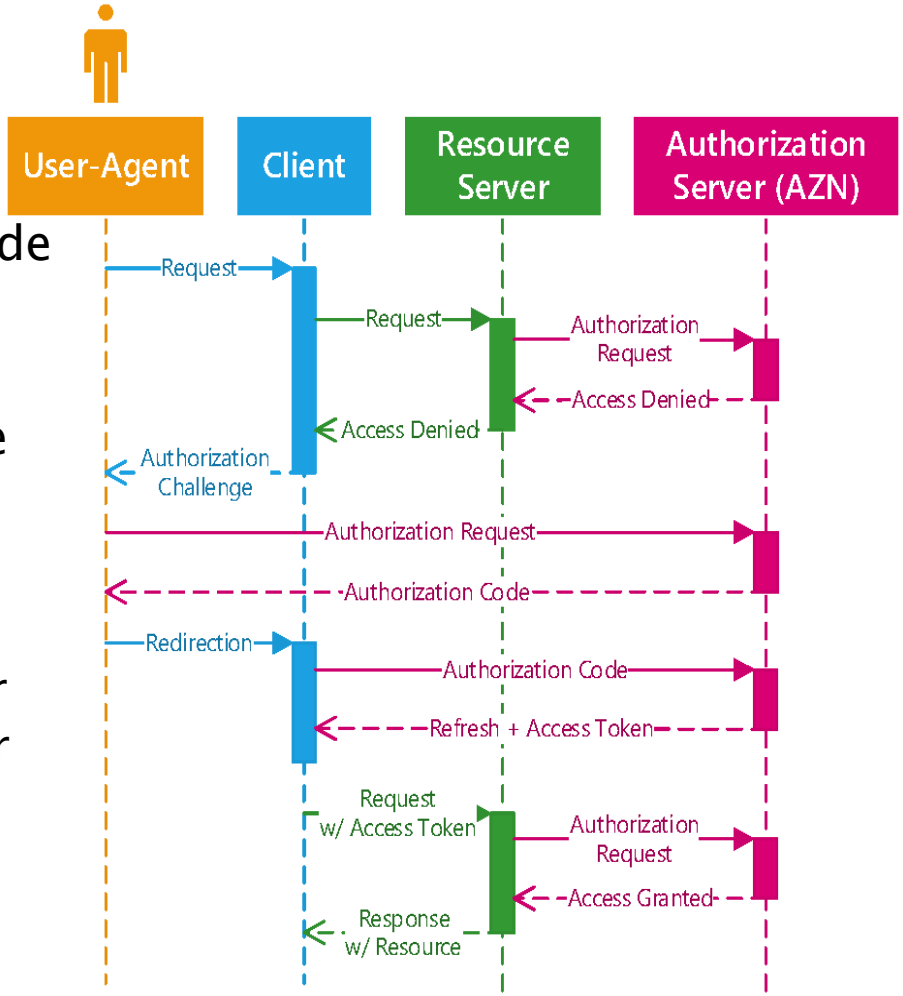
CAAS – Authorization interface





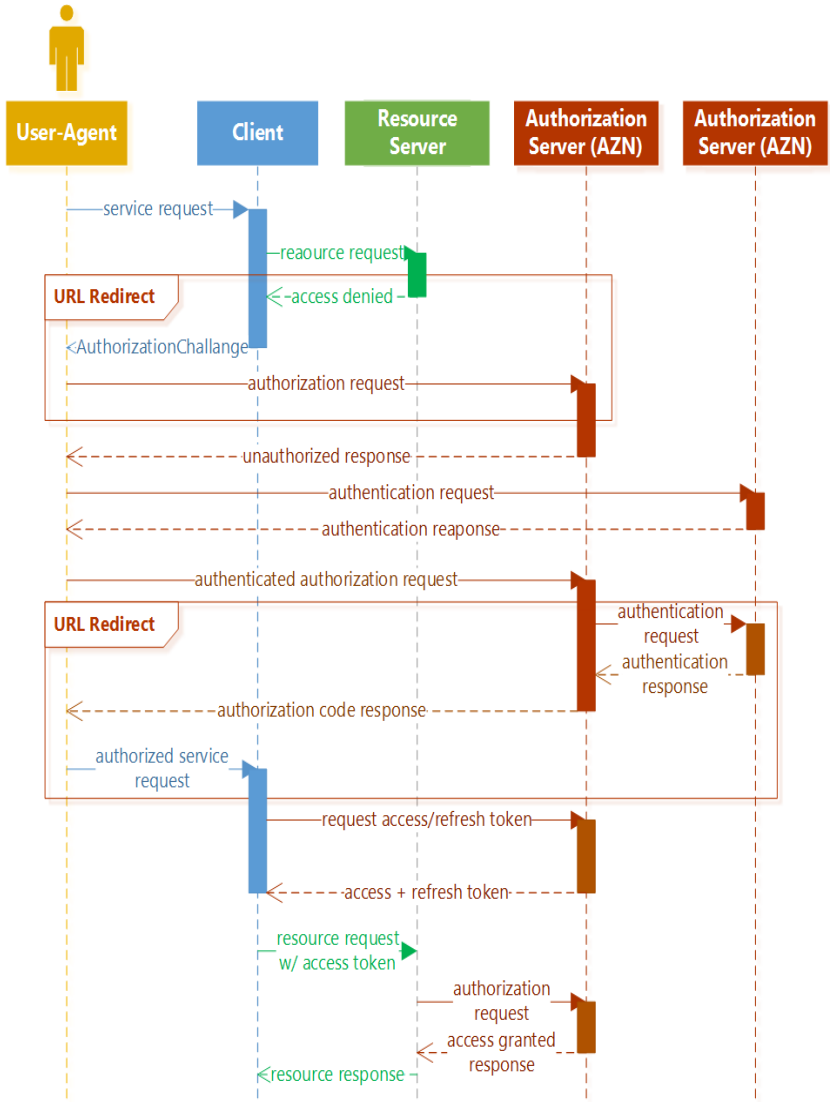
CAAS – Authorization Process

- Authenticated user requests a service from a client
- Client requires resources to provide the service
- User must request an access code to prove that he is authorized to access the resource
- Client obtains an access token for the resource on behalf of the user
- Resource server checks if the access token is valid



CAAS – Authorization + Authentication

- The user must prove his identity to the AuthenticationServer, before he is able to obtain an access code
- URL Redirects hide most of the complexity of the system
- For the User-Agent (e.g. Web browser) only 3 request are important
  - Initial resource request
  - Authentication request
  - Authenticated resource request



CAAS – Summary

- Comprehensive security solution for arbitrary cloud environments
- Centralized administration of users and permission for all cloud resources
- Implements token-based concept described in current state of the art standards (OpenId, OAuth2.0)
- Guaranties a secure communication between the involved parties even without a secure server configuration
- Scalable to arbitrary large cloud environments
- System complexity is completely hidden from the user
- Permission can be manage at any degree of granularity an complexity
- Resource provider need not to deal with user credentials

CLAFIS Project
Knowledge Processing Framework and Security Implementations

Conclusion

- Within work package 4 of this project two main parts were a Knowledge Processing Framework and according security mechanism
- For both of them a modern concept (architecture patterns, newest standards, open, cloud, …) has been developed and proof of concepts prototypes have been implemented.
- Prototypes have been successfully tested within the project
- Each of these frameworks could be further developed towards the CLAFIS system product, but also towards stand alone solutions.

Many thanks for your attention