

# UIT

## Greek Vice President Office

# Records and Document Management Systems

## **SCRIPTUM**

### ELLAK

### Athens 2010



# Records and Document Management System



# Agenda

- Records and Document Management System  
= Πρωτόκολλο = **SCRIPTUM**
- Presentation of OpenKM
- Scope of The Protocol System
- Interconnected Systems
- The Technology Stack
- Implementation Methodology



# Objectives

After this presentation, we should be able to

- Identify the Basic Systems that the RDMS interconnects.
- Understand the proposed Technology Stack for the Electronic Protocol System.
- Understand the Architecture of the System and identify the basic components.
- Follow the Implementation Methodology
- Specify the deliverables



# Scope

- To implement an RDMS that will assist the Greek Vice President Office to archive Protocol Documents. Must be:
  - Reliable and Precise: obey current Hellenic Republic Laws
  - Easy to use – something for everyone.
  - Open to third party systems
  - Modular and Expandable – adapt best practices in design.
  - Open Source / Osor



# Definitions

- Protocol Node: a set of the following items
  - Metadata: relevant tags like: unique protocol id, direction, sender...
  - Master document.
  - Attachment Documents
- User Profiles:
  - A set of users of the system sharing the same attributes
- Disctionary:
  - Contacts list
  - Templates



# Use Cases

- Administrator: The owner of the platform:
  - Create and Alter and Users.
  - Edit protocol nodes
  - Change Platform's parameters
- Writer User: Can create new protocol node
  - Complete metadata
  - Set master document
  - Add Attachements (MIMEs...)
- Writer User: Can print the protocol book



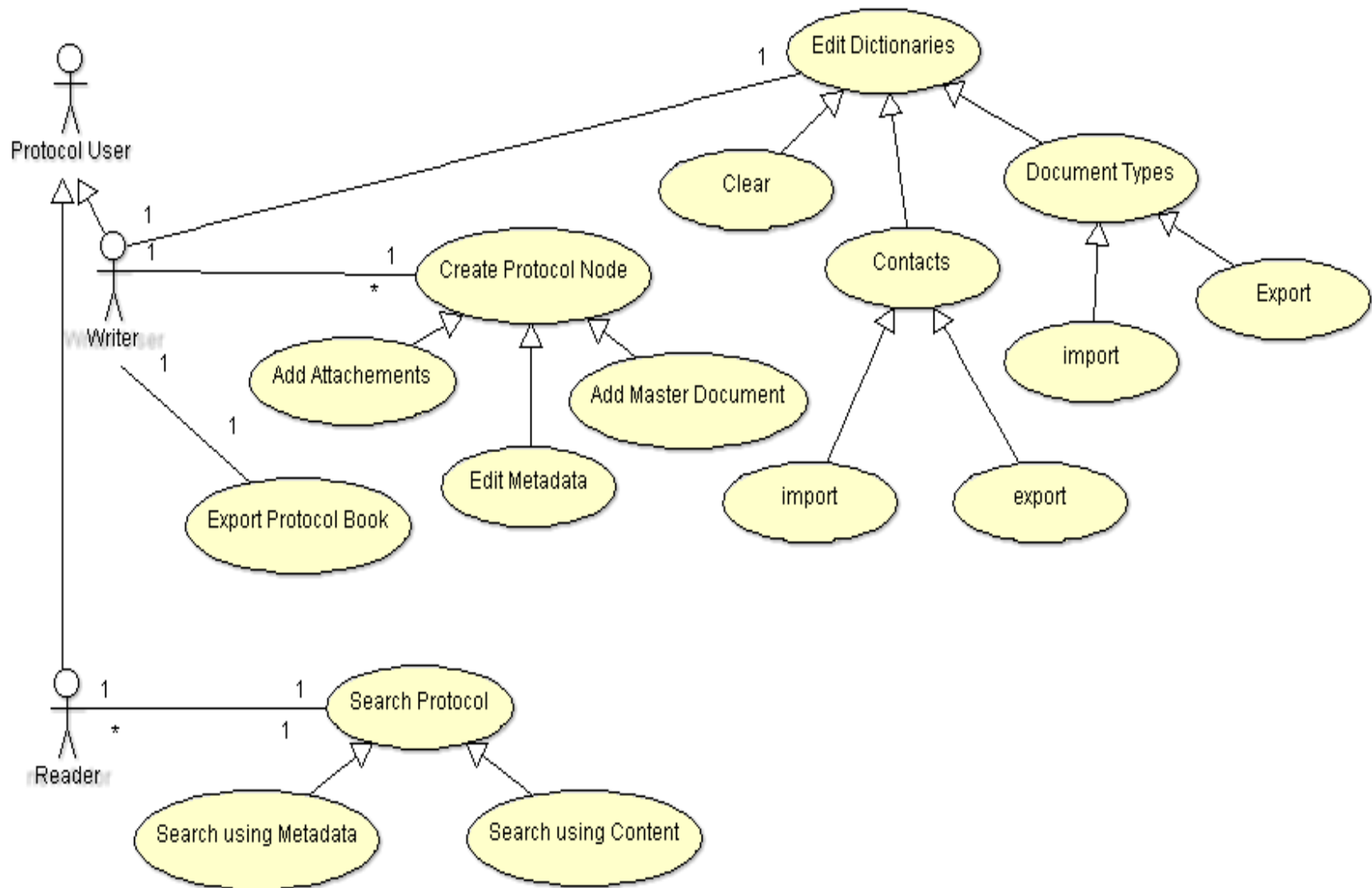
# Use Cases

- Writer User: Can import, export dictionaries
  - Contacts
  - Document Types
- Reader & Writer User: Can search the protocol
  - Using metadata
  - Using content of master
  - Using content of attachments





# RDMS

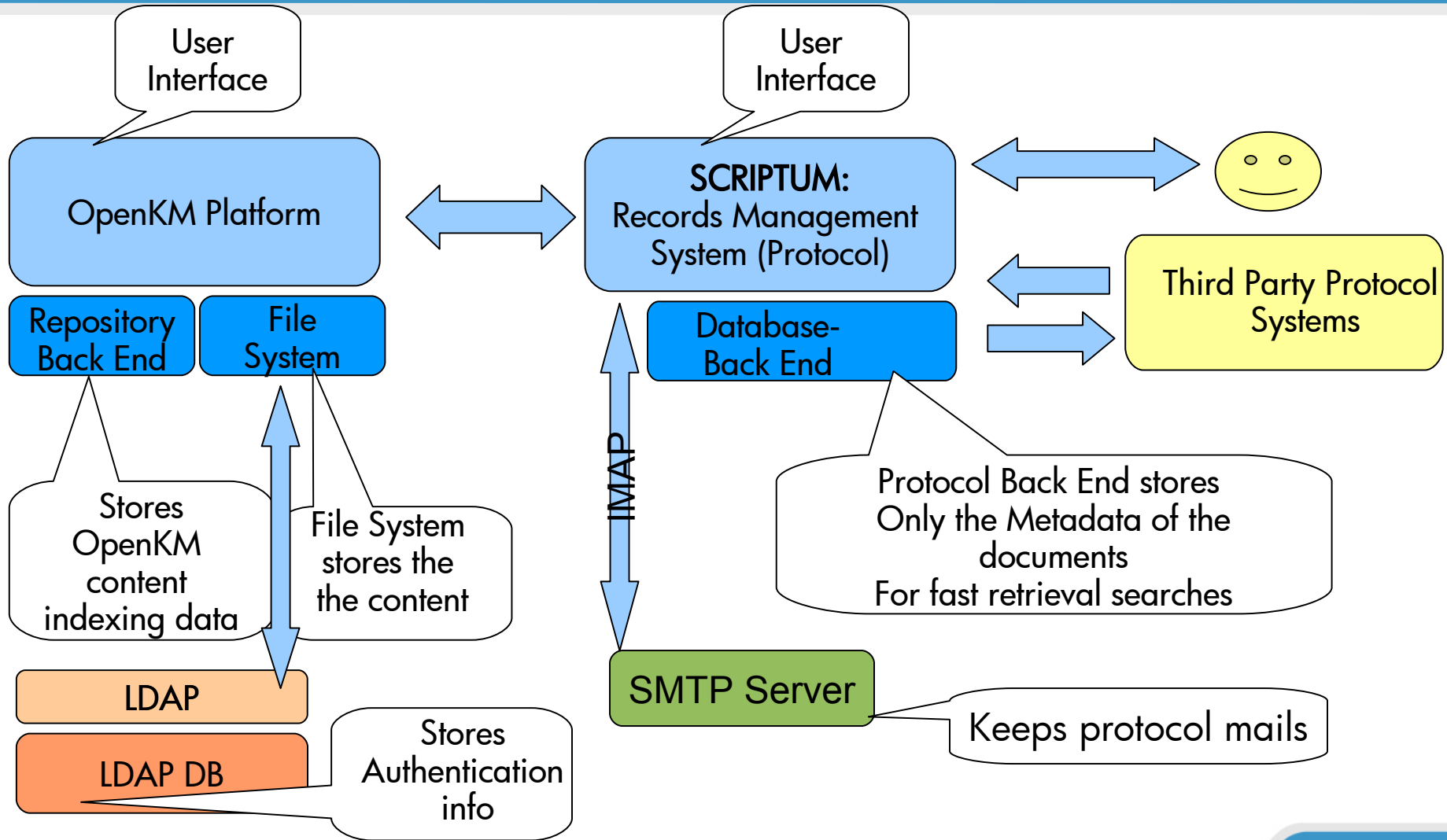


# Basic Systems

- **Scriptum:**
  - The Electronic Record Management System (Πρωτόκολλο)
- **Open KM**
  - Document Management Platform
- **LDAP/AD**
  - Authentication domain
- **Third party systems**
  - Any vendor's system wishing to communicate with Scriptum.



# Basic Interconnected Systems

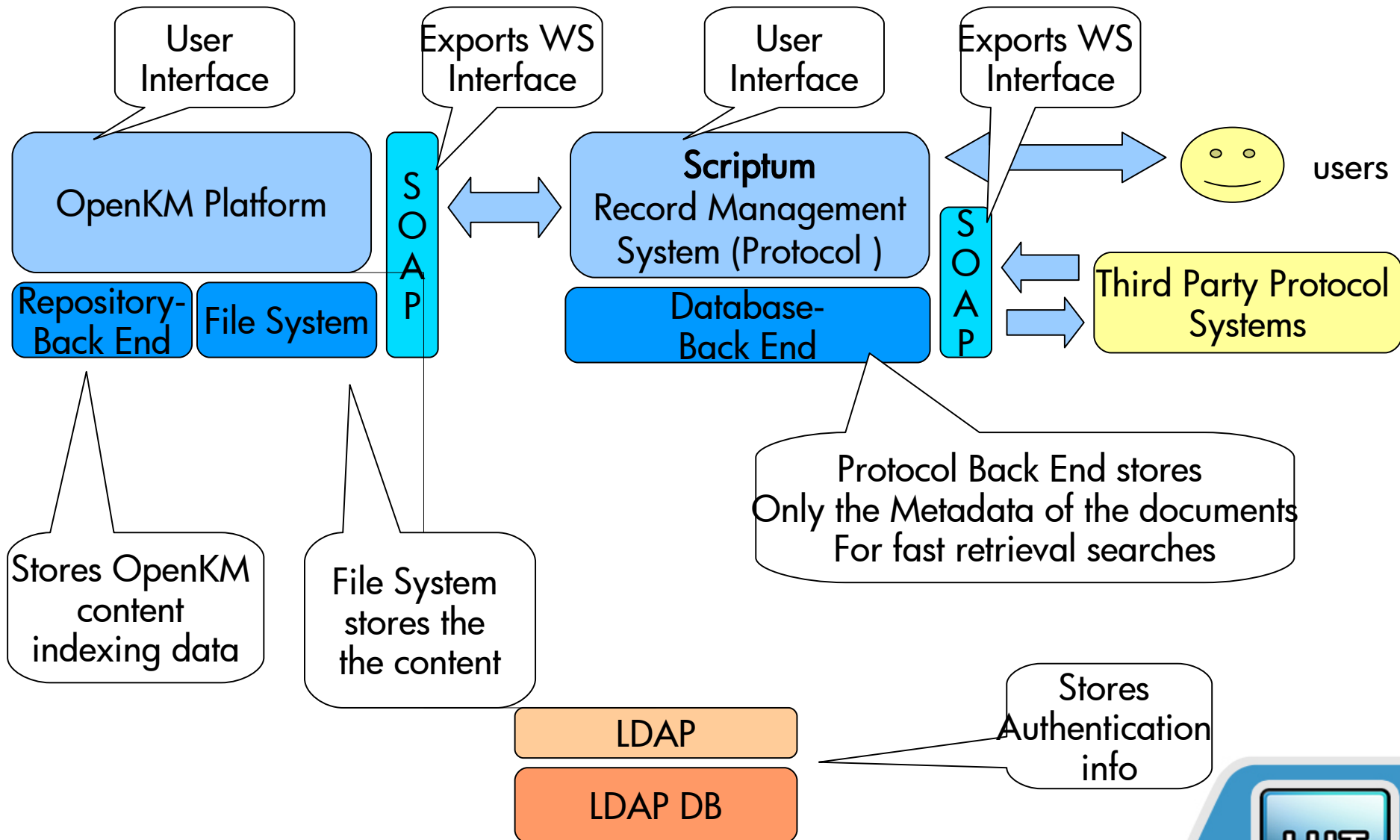


# OpenKM Client Solution

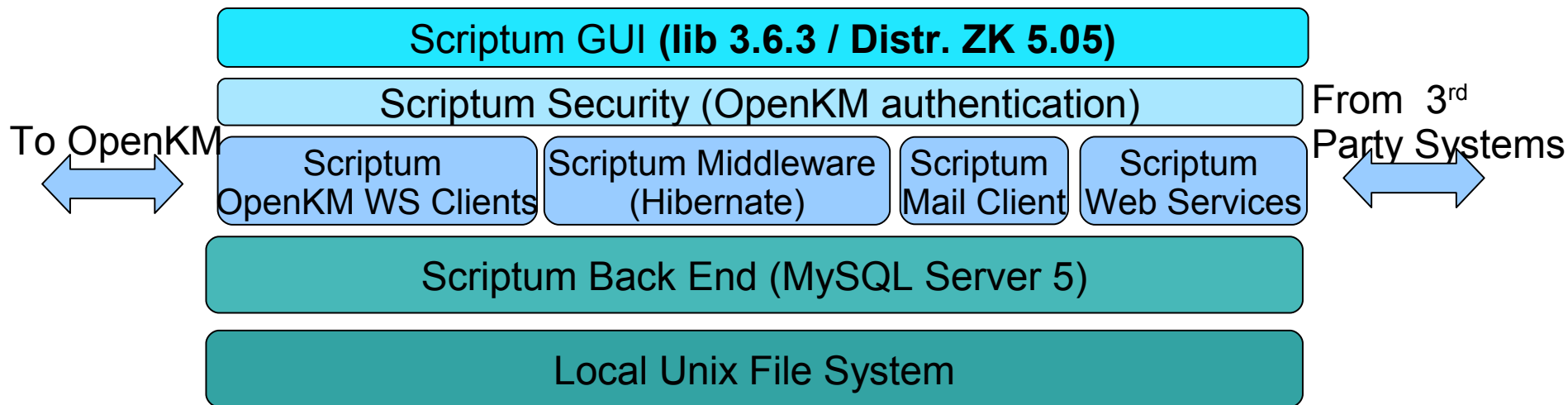
- Scriptum is implemented as a cooperative but independent system with OpenKM:
  - Designed with OpenKM look and feel but using different platform (Java ZK)
  - Utilizes the document repository of OpenKM. Communicates with the OpenKM's Web Service API
  - Has a database to store metadata of protocol documents for faster retrieval.
  - Also has the ability to save content on local filesystem.



# Systems in OpenKM Client Solution

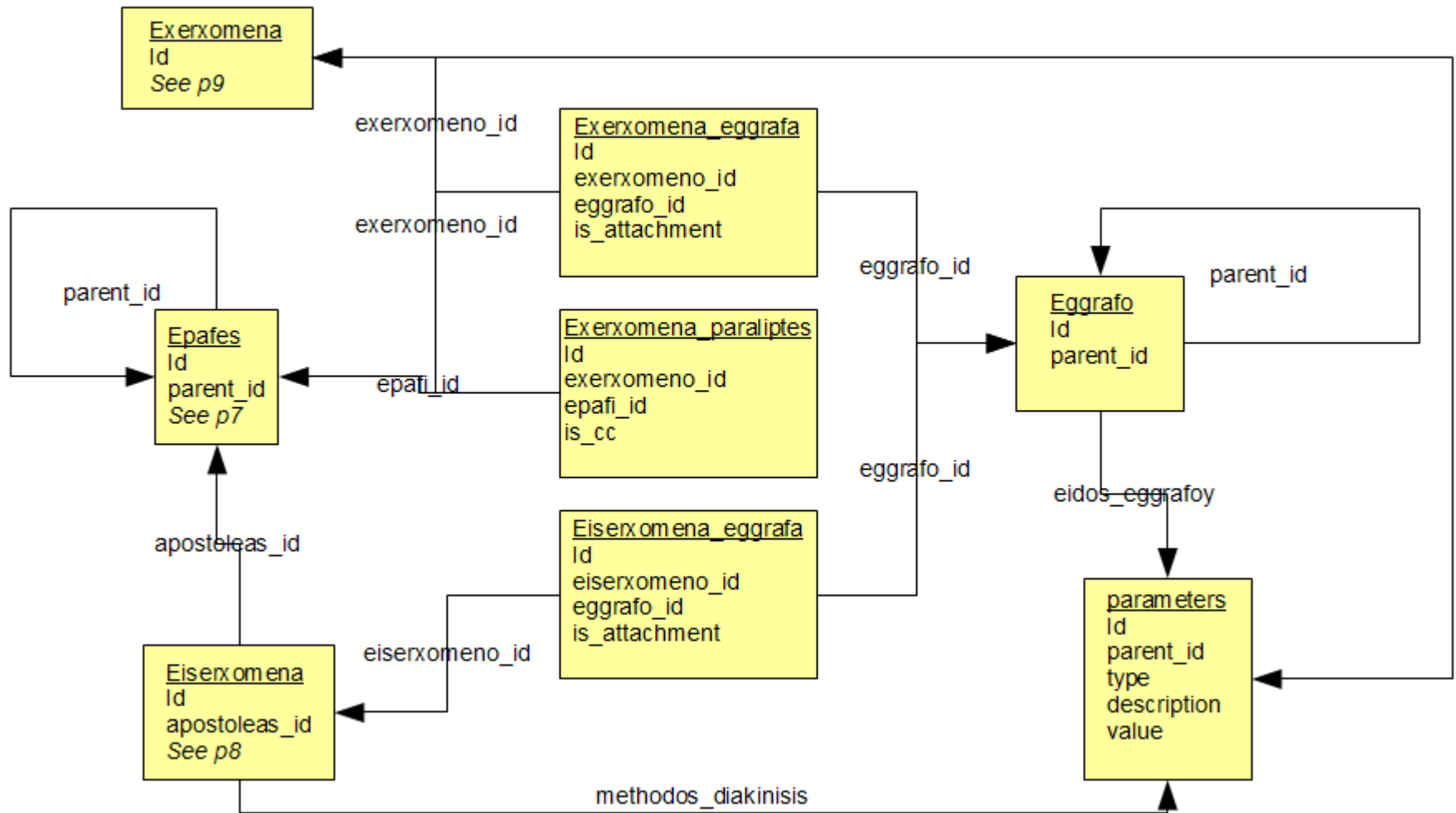


# OpenKM Client Architecture



- OpenKM client web services according to:
  - [http://wiki.openkm.com/index.php/Webservices\\_Guide](http://wiki.openkm.com/index.php/Webservices_Guide)
- Export Web Services to third party clients.
- Save contents also in local file system as folders in zip format.

# Protocol Schema



# Client Solution Tasks

- Design the Scriptum Back End
- Design the Scriptum Middleware (Hibernate)
- Design the Scriptum ZK GUI
- Design the OpenKM to Scriptum Interface
- Implementation of Local File System repository





# Implementation Method

- Use the RUP method:
- Last week: Specification Analysis
- Right now we are in the beginning of phase High Level Design.
- Next steps:
  - Produce a High Level Design Document.
  - Start working with the Sriptum Protocol detailed design towards implementation.



# Time Plan - Deliverables

- Community (2 weeks)
- RDMS (depends on implementation)
- Case Management (depends on implementation)
  
- Vice President Office Installation & Customization (On RDMS completion)
- Vice President Office Installation & Customization (On Case Management completion)



**UIT**

Greek Vice President Office  
Records and Document Management  
Systems

**SCRIPTUM**  
ELLAK  
Athens 2010



## Records and Document Management System



## Agenda

- Records and Document Management System  
= Πρωτόκολλο = **SCRIPTUM**
- Presentation of OpenKM
- Scope of The Protocol System
- Interconnected Systems
- The Technology Stack
- Implementation Methodology



## Objectives

- Identify the Basic Systems that the RDMS interconnects.
- Understand the proposed Technology Stack for the Electronic Protocol System.
- Understand the Architecture of the System and identify the basic components.
- Follow the Implementation Methodology
- Specify the deliverables



## Scope

President Office to archive Protocol Documents. Must be:

- Reliable and Precise: obey current Hellenic Republic Laws
- Easy to use – something for everyone.
- Open to third party systems
- Modular and Expandable – adapt best practices in design.
- Open Source / Osor



## Definitions

- Metadata: relevant tags like: unique protocol id, direction, sender...
- Master document.
- Attachment Documents
- User Profiles:
  - A set of users of the system sharing the same attributes
- Disctionary:
  - Contacts list
  - Templates



## Use Cases

- Administrator: The owner of the platform:
  - Create and Alter and Users.
  - Edit protocol nodes
  - Change Platform's parameters
- Writer User: Can create new protocol node
  - Complete metadata
  - Set master document
  - Add Atachements (MIMEs...)
- Writer User: Can print the protocol book

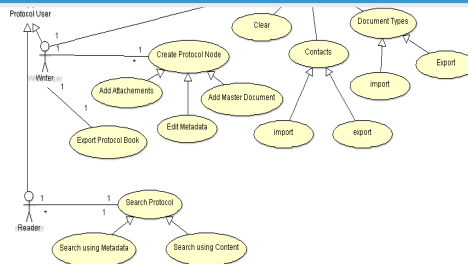


## Use Cases

- Writer User: Can import, export dictionaries
  - Contacts
  - Document Types
- Reader & Writer User: Can search the protocol
  - Using metadata
  - Using content of master
  - Using content of attachments



# RDMS

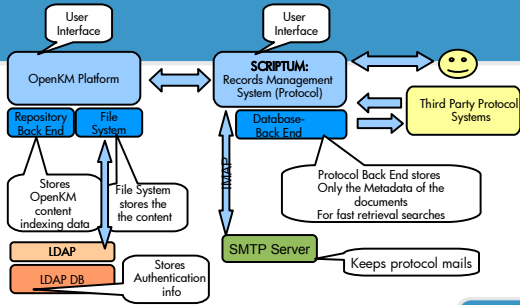


## Basic Systems

- The Electronic Record Management System (Πρωτόκολλο)
- Open KM
  - Document Management Platform
- LDAP/AD
  - Authentication domain
- Third party systems
  - Any vendor's system wishing to communicate with Scriptum.



## Basic Interconnected Systems



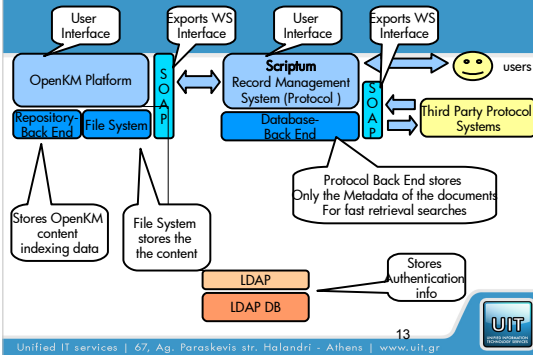
## OpenKM Client Solution

independent system with OpenKM:

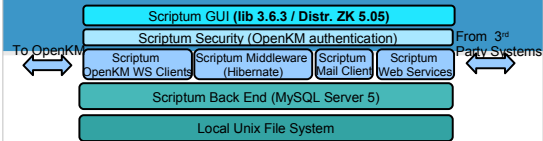
- Designed with OpenKM look and feel but using different platform (Java ZK)
- Utilizes the document repository of OpenKM. Communicates with the OpenKM's Web Service API
- Has a database to store metadata of protocol documents for faster retrieval.
- Also has the ability to save content on local filesystem.



## Systems in OpenKM Client Solution



## OpenKM Client Architecture



- OpenKM client web services according to:
  - [http://wiki.openkm.com/index.php/Webservices\\_Guide](http://wiki.openkm.com/index.php/Webservices_Guide)
- Export Web Services to third party clients.
- Save contents also in local file system as folders in zip format.





## Client Solution Tasks

- Design the Scriptum Back End
- Design the Scriptum Middleware (Hibernate)
- Design the Scriptum ZK GUI
- Design the OpenKM to Scriptum Interface
- Implementation of Local File System repository



## Implementation Method

- Last week: Specification Analysis
- Right now we are in the beginning of phase High Level Design.
- Next steps:
  - Produce a High Level Design Document.
  - Start working with the Sriptum Protocol detailed design towards implementation.

## Time Plan - Deliverables

- RDMS (depends on implementation)
- Case Management (depends on implementation)
- Vice President Office Installation & Customization (On RDMS completion)
- Vice President Office Installation & Customization (On Case Management completion)

