

ZopRA LIMS Framework

Security Concept

Ingo Keller

Peter Seifert

Genetics, Biology Department

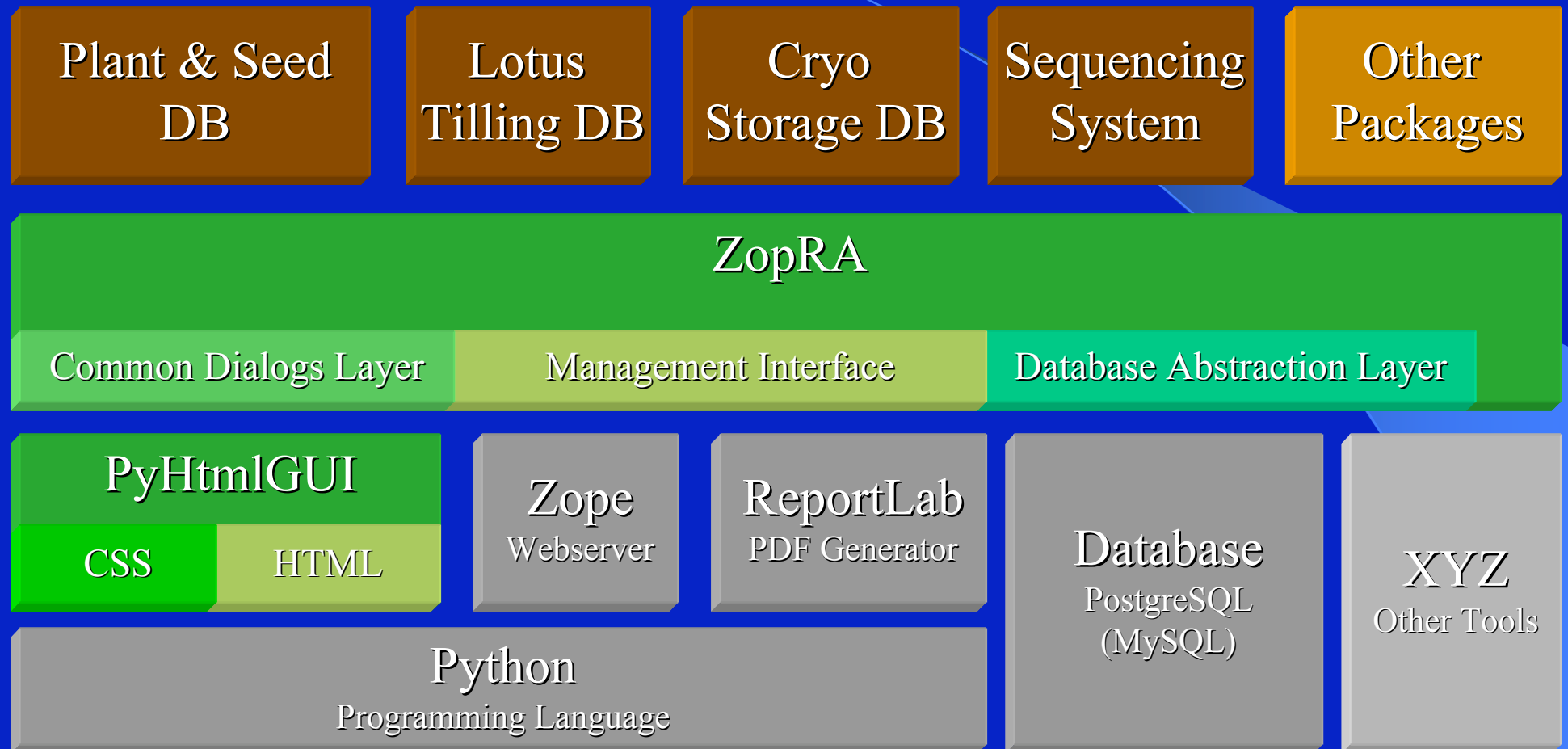
Ludwig-Maximilians-University of Munich

Motivation
Architecture
Security Concepts

Motivation

- Open-Source Software Stack
- Cooperating Laboratories
 - combined data -> aggregated information
 - personalised permission settings
- LIMS Development Framework
 - support of Web 2.0 technology
- Components Based Software Development
 - high degree of generic components
 - layered structure

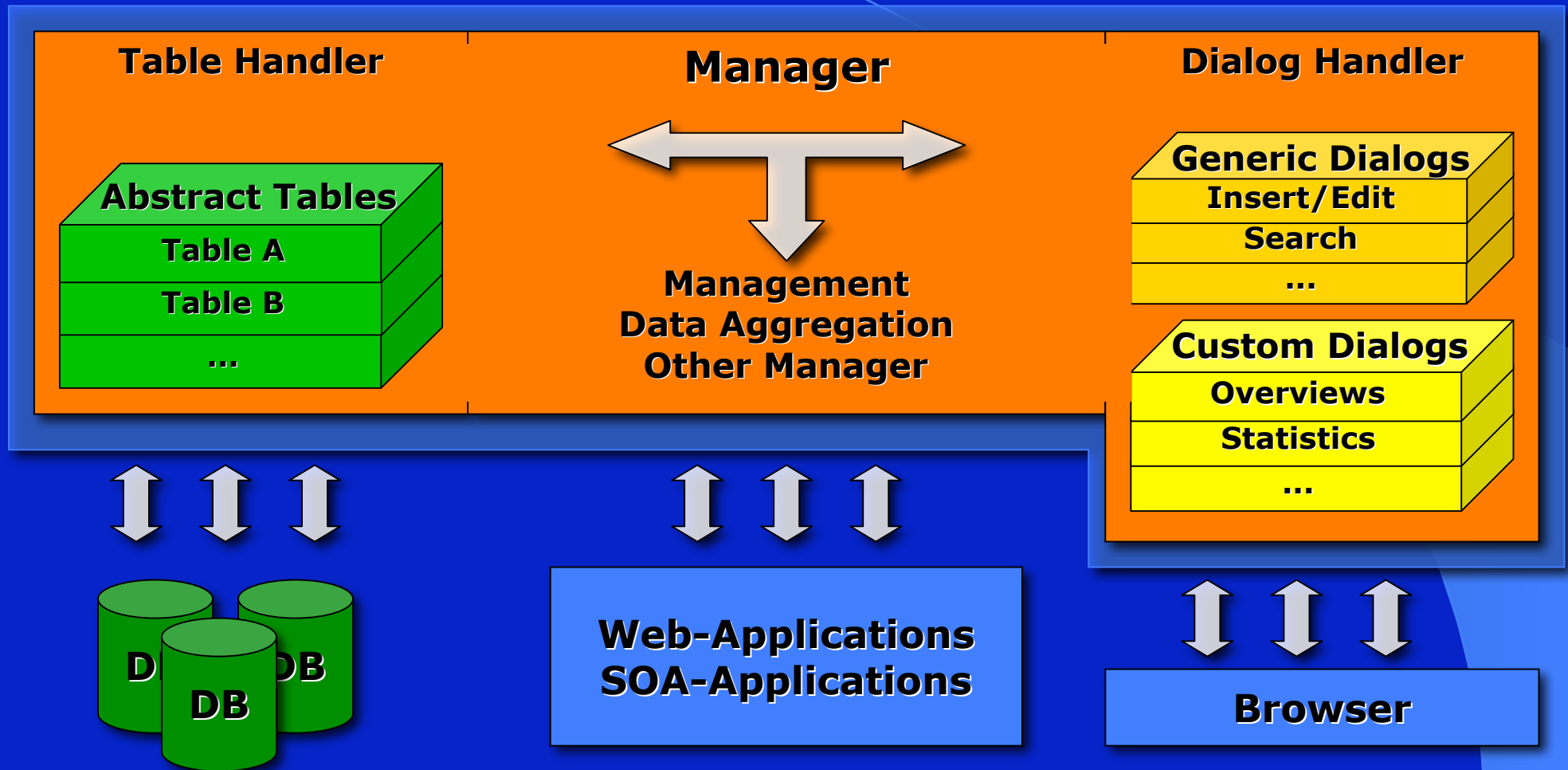
Zope Research Architecture Software Stack



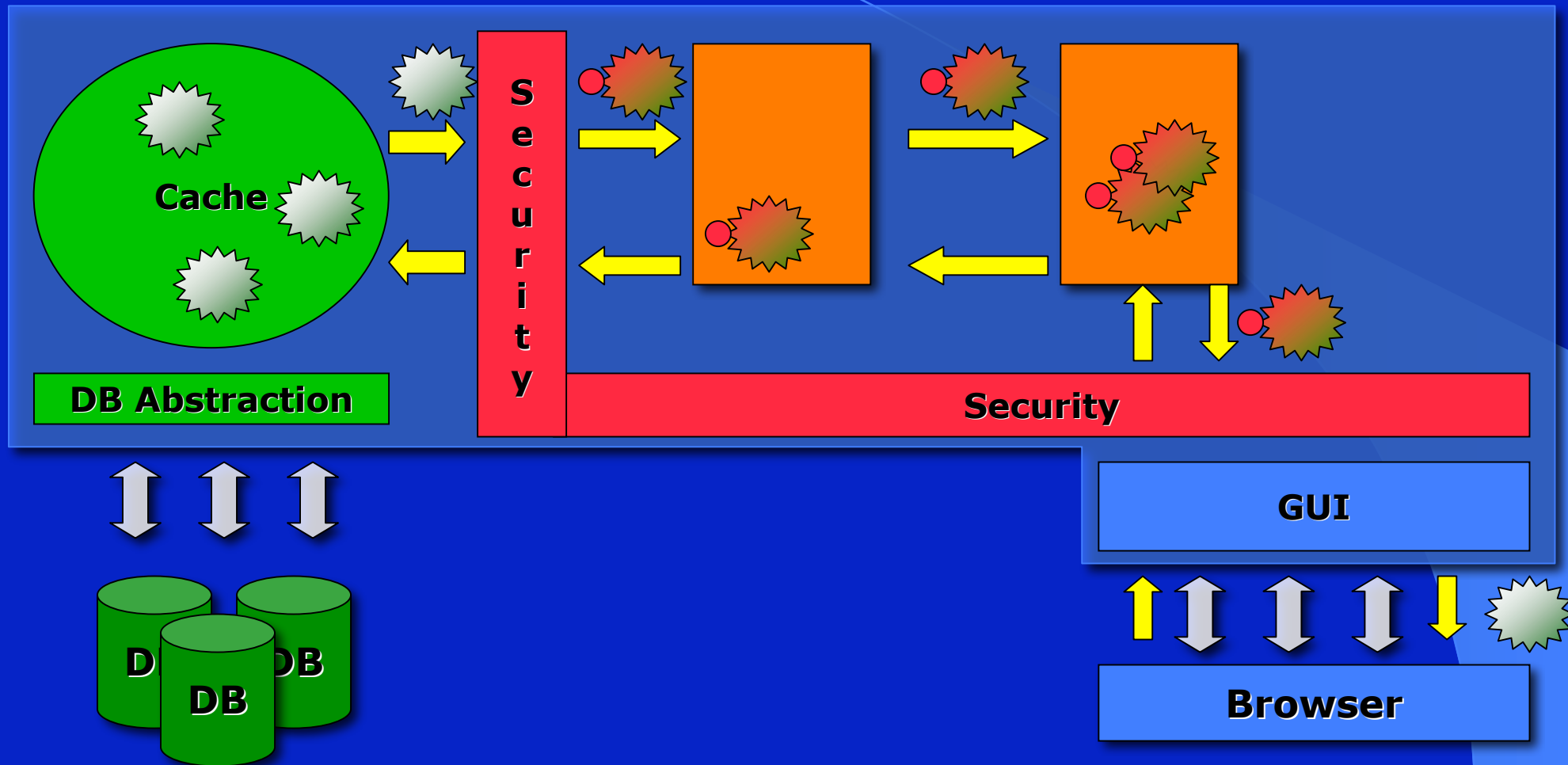
Security Aspects

- Possible Intruder Classes
 - Anonymous, User, Administrator, Developer
- Separation of data and security information
 - integration of different data sources
- 2 Levels of security checking
 - (Web-) GUI Level, Data Source Level
- Introduction of security objects attached to data objects
 - secGUIPermission - user dependend
 - secEntryPermission - entry dependend

ZopRA Core Components



Data Object Workflow

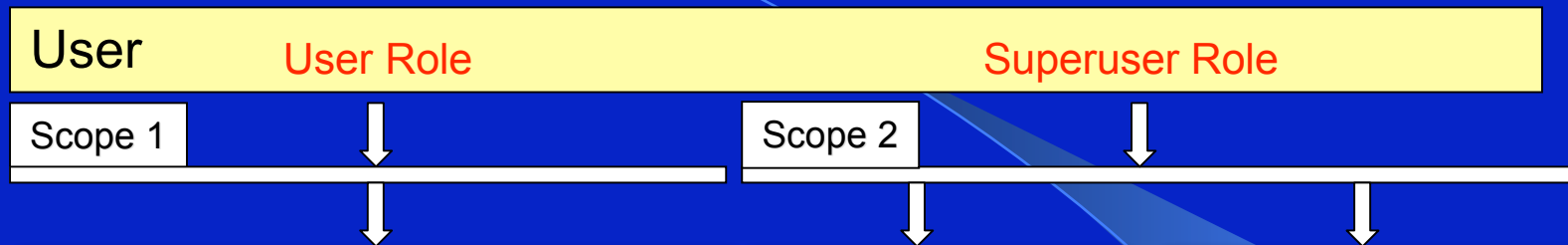


Global Access / Access Roles

- Simple Access Control
 - application wide handling
 - steering of GUI
- Basic Permissions
 - insert, view, edit, delete
- Level based user groups
 - Visitor < User < SuperUser < Admin
- Custom Permissions
 - create custom group
- Complex Access Control
 - Scope: defined application parts
- Overwrite Global Permissions
 - same permissions, not system wide

Access Role vs. Entry Based Security

Access Roles

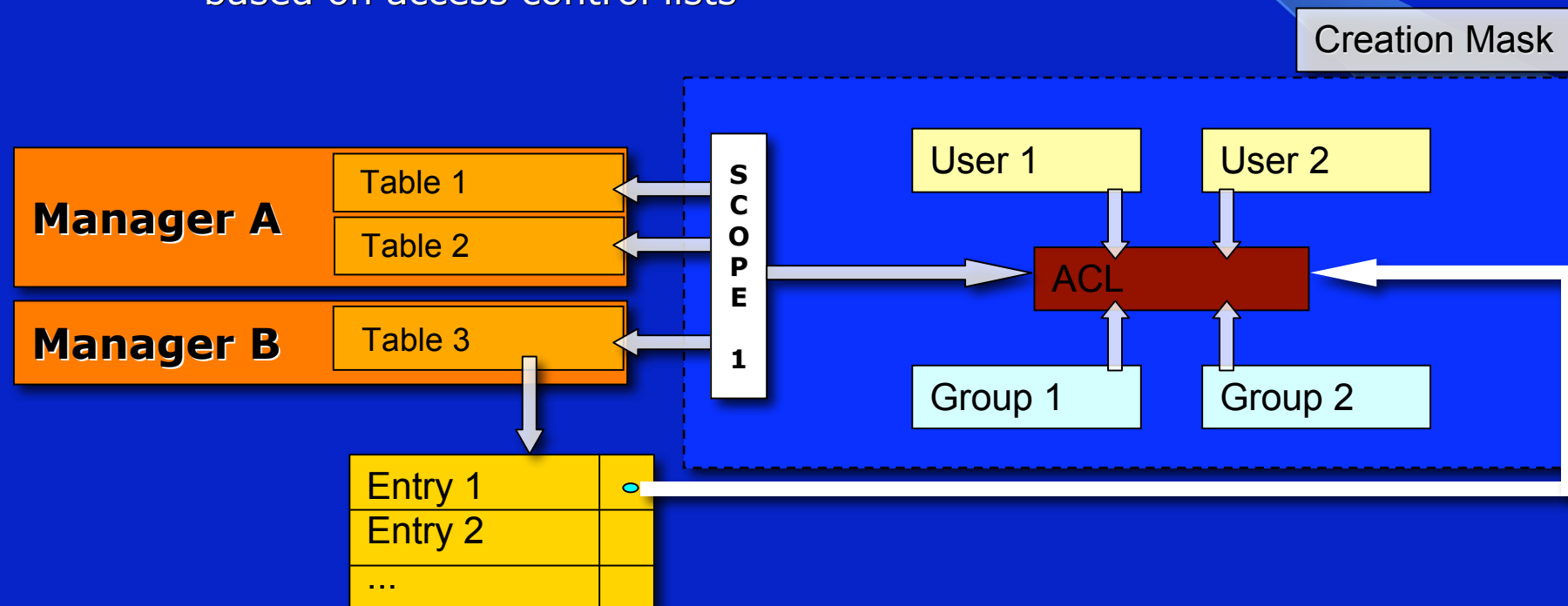


EBaSe



Entry Based Security (EBaSe)

- Fine granular permission control
 - using scopes
 - creation mask: default permissions for user and groups per scope
 - based on access control lists



ACL Implementation

- ACL String Composition

- Separator ":"
- Indicator of type "u" - user, "g" - group
- Identifier number (userID, groupID)
- Permissions "r" read
 "w" write
 "l" labelread
 "d" delete
- e.g. ":u23w:u23r:u21l:g3r:"

- Retrieval

- depends on data source -> LIKE search in databases (*:u23r:*)

Security Object Implementation

- secGUIPermission
 - hasRole()
 - hasMinimumRole()
 - hasPermission()
- secEntryPermission
 - isOwner()
 - hasPermission()
- Object Location
 - secGUIPermission within the SESSION object of the user
 - secEntryPermission within the data dictionary
- Securing secEntryPermission object with hash function
 - hash over content and permissions

Interaction With The User

Seed Bag Info Page for 65101

Entered by Peter Seifert
Entry Date 03.12.2006
Last edited by
Last edited on 03.12.2006
Owner Peter Seifert

Permissions

Users

	l	r	w
Jodie (Jodie Pike)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Brande (Brande Wulff)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
catharine (Catharine White)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Groups

	l	r	w
All	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[Edit permissions](#)

Logged Events

```
add entry 51380 to lo_smseed
update entry 51380 in lo_smseed
update entry 51380 in lo_smseed
update entry 51380 in lo_smseed
update entry 51380 in lo_smseed
update entry 51380 in lo_smseed
```

OK

Comparison

- Filesystem oriented approaches
 - limited overlapping group permission support
- Database oriented approaches
 - Database dependent
 - Fine granularity -> Large number of views for large number of groups
 - Row Level Security - RLS (e.g. Oracle) -> not available as open source
- Similar approach: eGK
 - uses encryption of data -> aggregation of data not possible
- solution: data source independent security approach

Thanks to

- Martin Parniske Lab
Genetics LMU Munich
Sainsbury Laboratory Norwich/UK
- Trevor Wang Lab
Jonn Innes Centre (JIC) Norwich/UK

