

Project/ Open



Architecture

27.7.2003

Frank Bergmann <fraber@fraber.de>

- The Big Picture
- GUI Structure
- Component Architecture
 - Component Use Overview
 - Component Conventions
- Localization
- Permission Model
- Customization Concept
 - Application Lifecycle
 - Customization Options (1)
 - Customization Options (2)



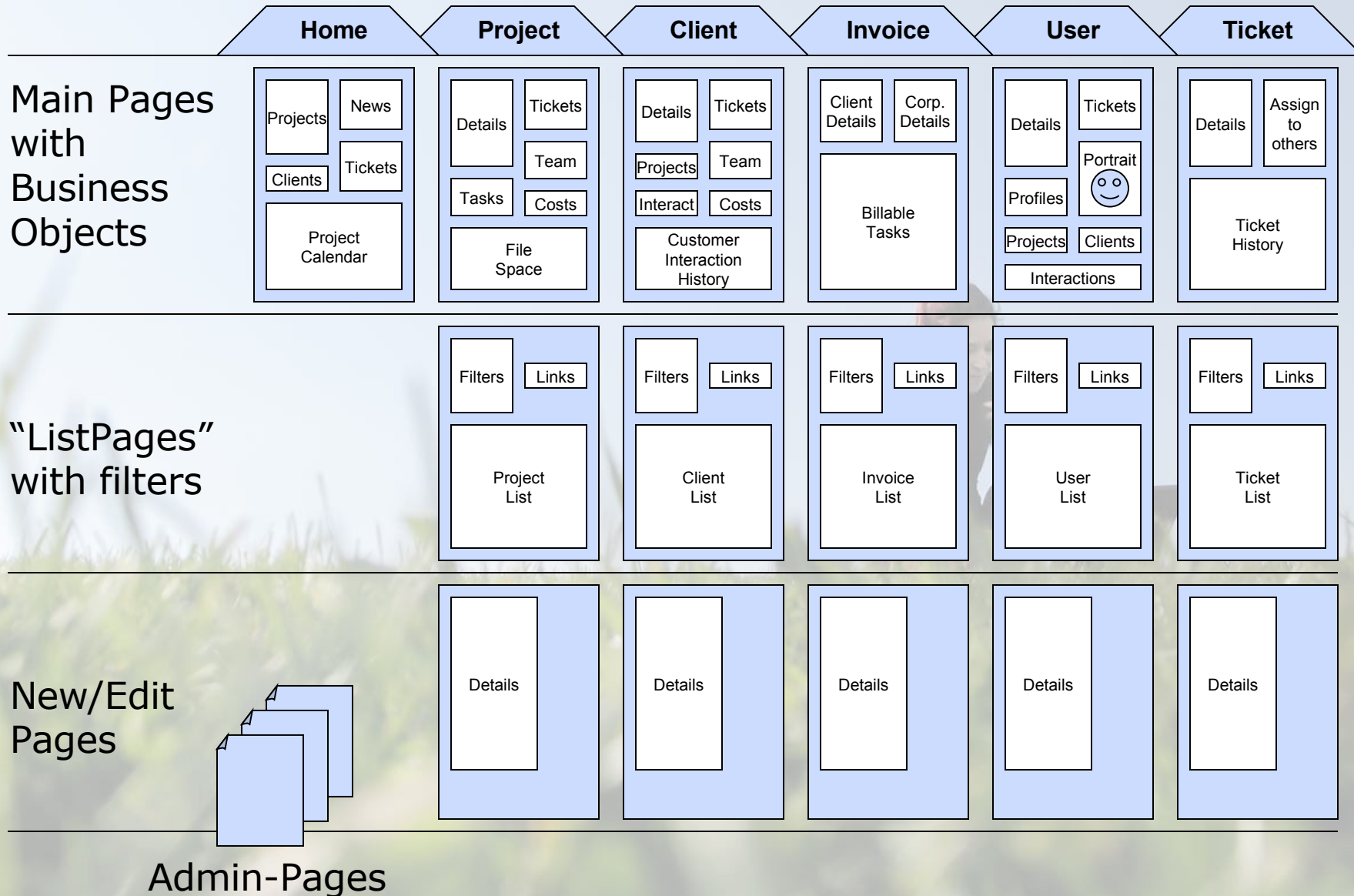
The Big Picture

- Project/Open is a web-based "Project-ERP".
- It combines project collaboration (content & community) with financial and management functionality (costs, invoicing, accounting, ...)
- Project/Open is based on the OpenACS (Open ArsDigita Community System) platform.
- Most Project/Open modules are free or GPLed, except for specific customer functionality

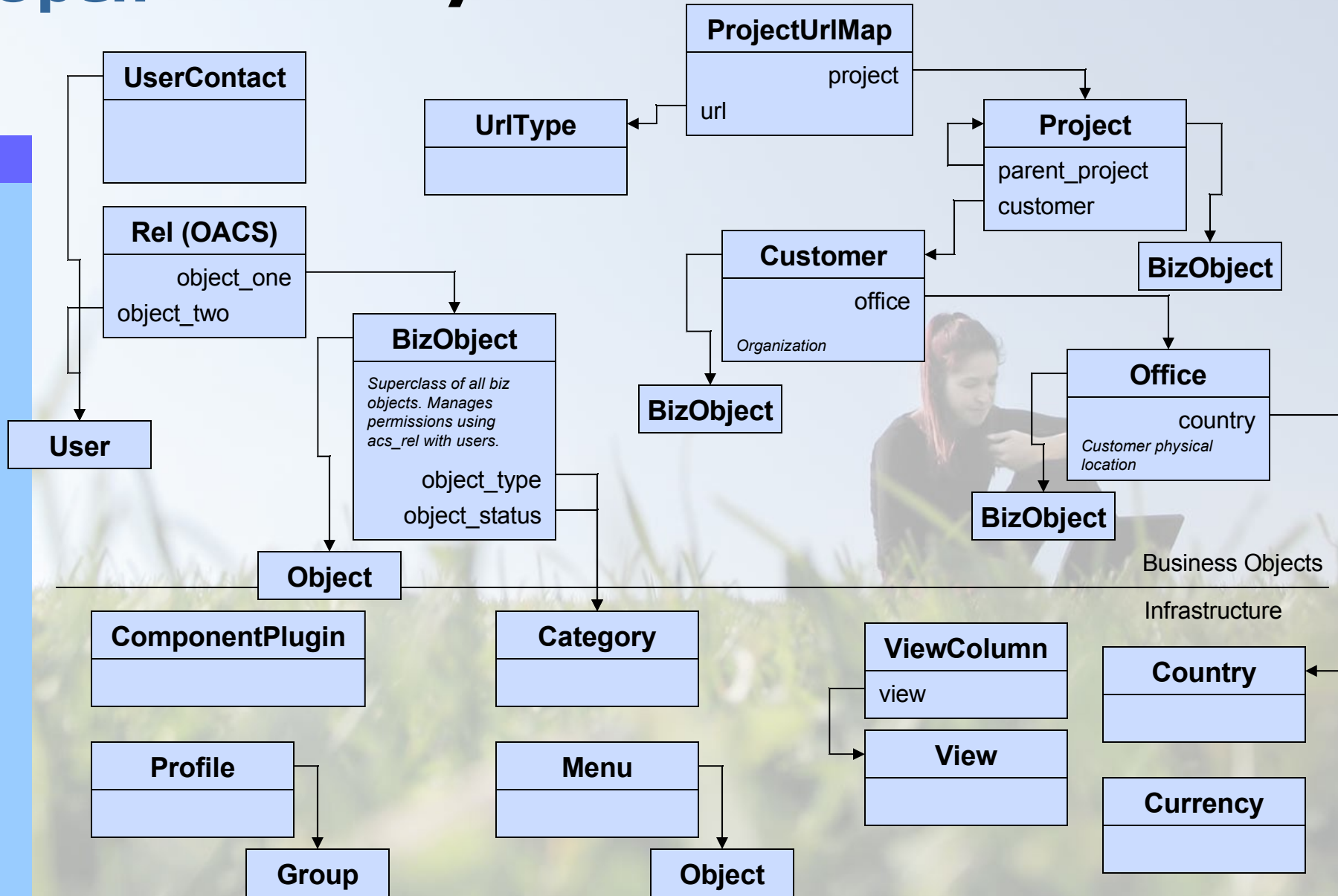
P/O	Open-ACS	Data base	Date
V 1.0	ACS 3.4.9	Oracle 8i (8.1.7)	11/2003
V 2.0	OpenACS 5.0	8i	3/2004
V 3.0	OpenACS 5.1	8i, 9i, Postgres	10/2004

Platform Roadmap: V1.0 depends on Oracle 8i, but V3.0 will feature a free DB alternative.

Base GUI Structure



P/O Core Data Model



Project/ Open Component Architecture

Component Use Overview

	Home	Project	Client	Invoice	User	Ticket
Project Components	-Project List	-ViewPage -ListPage -New/Edit	-View: Project List (short)	-New: Project List -List: Filter -View: Renderer -Print: Renderer	-Project List	-Project Select -Project Render
Client Components	-Client List	-List: Renderer -List: Filter	-Project List	-Project/Task List -Project Filter	-Project List	-Project Select -Render Comp
Invoice Components	- ...	- ...	- View: Invoice List	-View Page -List Page -New/Edit	- ...	- ...
User Components	- ...	- ...	- ...	- ...	-View Page -List Page -New/Edit	- ...
Ticket Components	- ...	- ...	- ...	- ...	- ...	-View Page -List Page -New/Edit
Permissions	- ...	-View: Perm+Estim	-View: Key Accounts (perm) -View: Customer employees (perm)	- ...	- ...	- ...

Project/ Open **Component Architecture**

Component Conventions

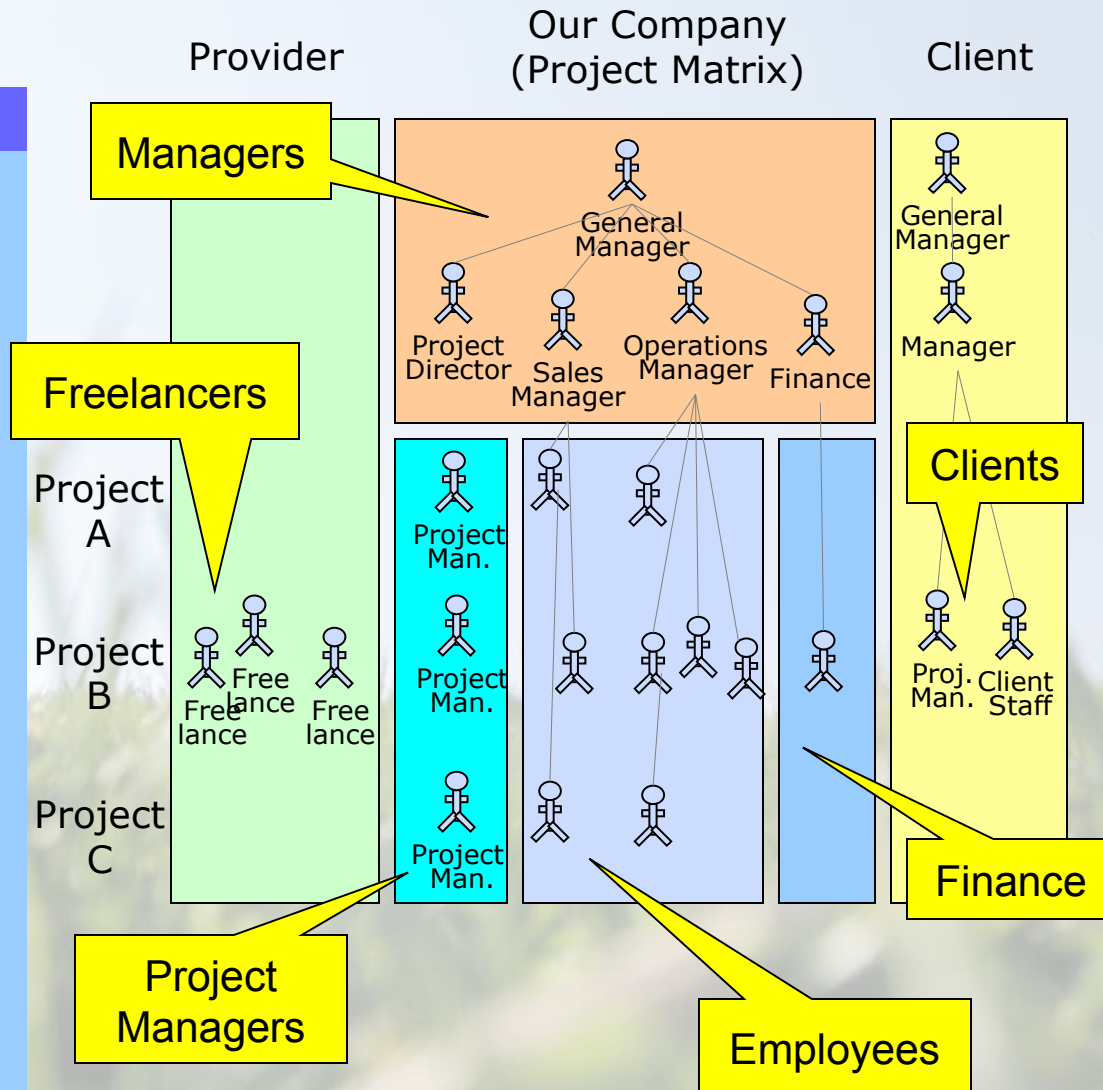
- (GUI-) Components are used to render lists of objects, to render individual objects (in various variations) and to “filter” objects (restrict the elements of a SQL “where” clause).
- These components have to deal with all important functions such as permissions, localization and design templates.
- HTML pages are used to validate input variables, generate the page layout, and to “wire” the GUI-components together (glue-code).

- All numbers, currency amounts, dates, times and physical locations are rendered as a function of the chosen user locale
- All currency amounts are preceeded by a currency identifier. A system currency determines the default.
- All static text used in pages and components is translated before rendering the HTML page. An English text in the target location is used as a default value in case of incomplete translations. The translation interface also specifies the target location to provide a context for translators.
- Dynamic text (names and descriptions of business objects) are not translated.
- Future versions of Project/Open may contain accounting modules specific for individual countries.

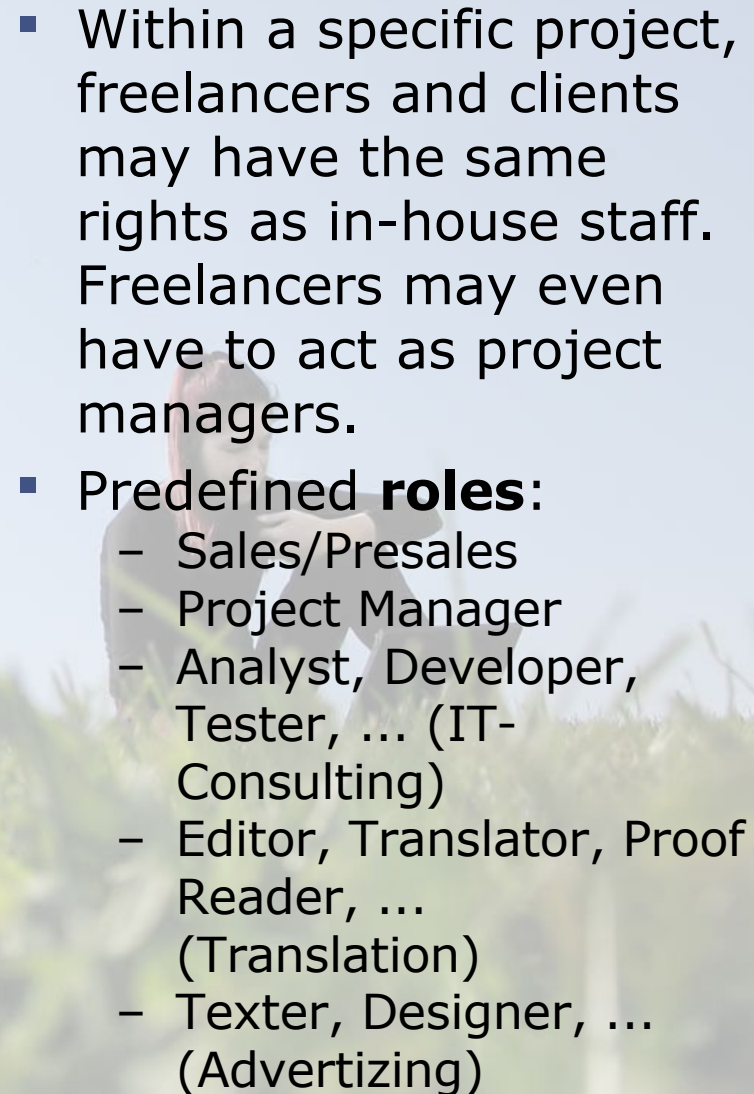
Permission Model

- Permissions are divided into profile-based permissions and role-based permissions
- **Profile-based permissions**
 - Restrict the visibility of pages and modules to a user
 - Are modified as part of system administration
 - API: `im_permission(user_id, token)`: Validates that a user has access rights to the permission token.
- **Role-based permissions**
 - Restrict the access to business objects (currently “Project” and “Client”)
 - Permissions can be modified by the creators of the respective objects, to configure project/client team.
 - API: `im_has_role(user_id, object_id, role)`: Validates that a user has been assigned the specific role.
 - Roles are ordered hierarchically (administrator -> member -> specific_roles).

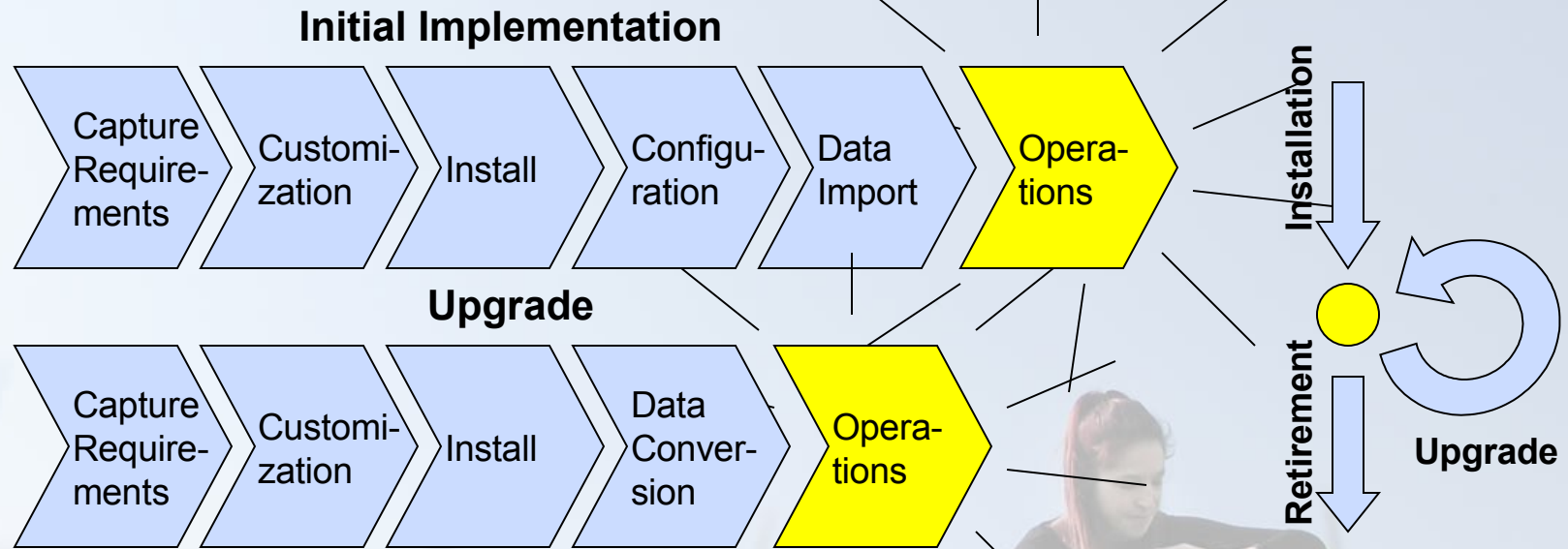
Profile Based Permissions „Vertical“ Permissions



- **Predefined Profiles:**
 - Manager
 - Project Manager
 - Employee
 - Finance
 - Client
 - Freelance
- Access permissions to all critical data can be assigned to profiles
- Additional profiles can be defined if necessary



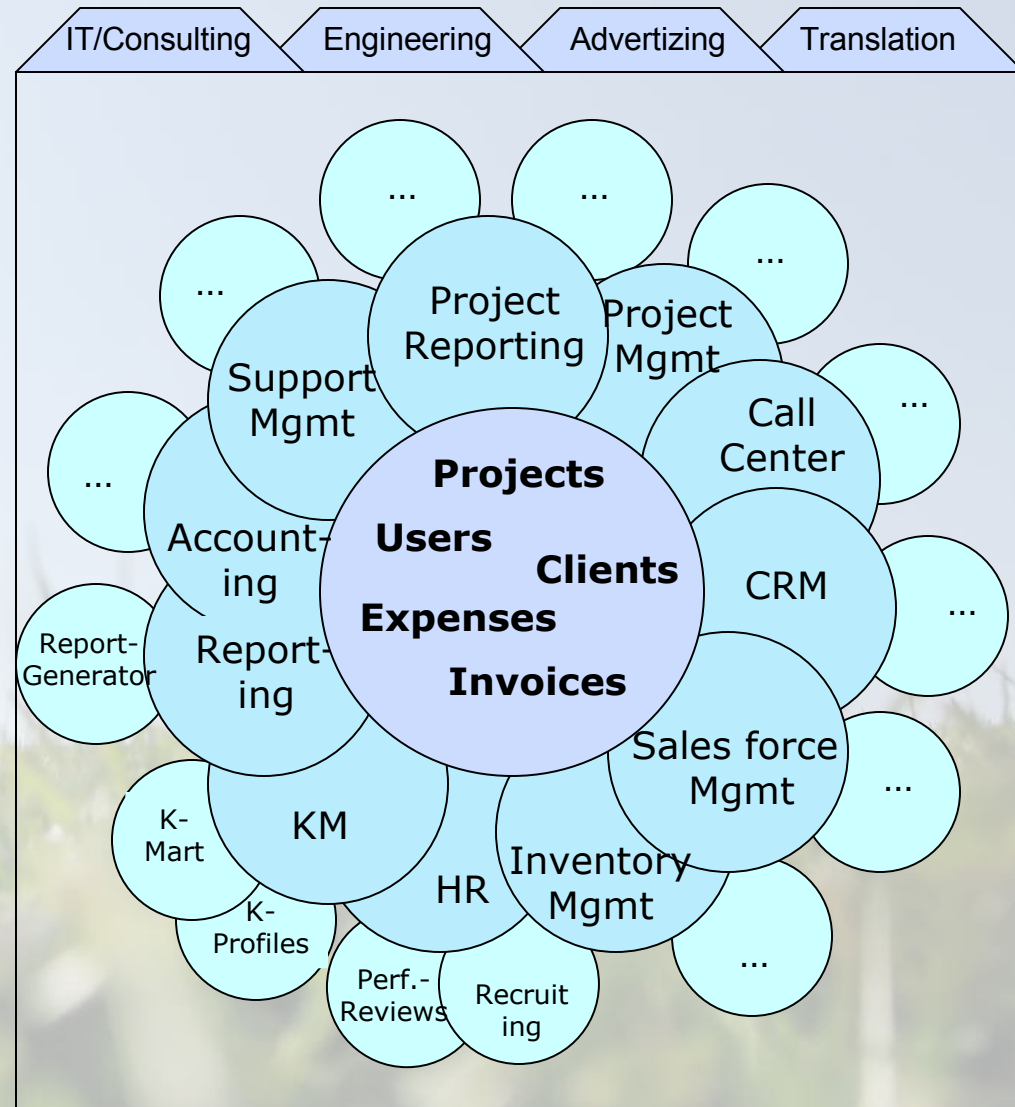
Application Lifecycle



- ERP applications are not a static systems, but need to evolve continually with changes in the business processes of the client (customizations)
 - ERP applications need to integrate/communicate with other corporate applications
 - ERP applications have life-cycles of 5-20 years
- ➔ **The challenge is to maintain customizations when upgrading to the next release.**

Extensible Architecture

- A central core allows for extensions with different application modules
- Different industry sectors need different configurations and modules
- Customizations should be portable to the next version as easily as possible.



Extensible Architecture

- Not all modules are useful for all sectors
- Some modules are different for every sector (for example invoicing)
- Some of these modules may exist in several variants specific for each sector.

	IT	Consulting	Engineering	Advertising	Translation	Intl. IT Dept.	Intl. Product Dev.
Presales	*	*	*	*	*	(*)	(*)
Project Collaboration	*	*	*	*	*	*	*
Project File Sharing	*	*	*	*	*	*	*
Customer Management	*	*	*	*	*	*	*
Employee Management	*	*	*	*	*	*	*
Timesheet	*	*	*	*	*	*	*
Travel Costs	*	*	*	*	*	*	*
Project Reporting	*	*	*	*	*	*	*
Support Management	*	*	*	*	*	*	*
Recruiting	*	*	*	*	*	*	
Invoicing	+	+	+	+	+		
CRM light'	*	*	*	*	*		

=applicable, ()=maybe applicable, +=customized modules

- ➔ The Module architecture has to deal with the interdependencies between each other and over time.

Customizable Items

How can additional module change the “core”?

- **GUI**
 - Home-Page (module-specific status boxes)
 - The main menu
 - Submenus of business objects
 - Views of biz objects (Project, Client, User, ...)
- **Categories** (=workflow states and types)
 - Object status categories
 - Object type categories
- **Data Model**
 - Add columns to existing BizObj tables (system modules)
 - BizObj extension tables (sectorial configuration modules)
- **TCL Libraries**
- **Permission Model**
 - Add new permission tokens
 - New system roles
 - New Object/Group specific roles

Customization (1)

- **Configurable Customization (100% preservation)**
The system has been designed for the particular customizations. The customization involves changes in the configuration file and/or database contents.
→ Future version of Project/Open maintain compatibility with these configurable options, possibly extending them.
- **Business Object Extensions (90% preservation)**
Special database tables are reserved to hold customer extensions to business objects. These tables are not replaced during a system upgrade. Rendering of these objects is handled by (modified) components in a new library.
→ Future version of Project/Open must not overwrite the extension tables. The client will have to integrate the modified components into the new View/List/Edit pages.

Customization (2)

- **New Custom Modules (80% preservation)**
The user may add new modules to the system, including their own data model and components.
 - ➔ Future version of Project/Open will not take into account the compatibility with custom modules.
However, it is likely that the integration of the new modules with the new version will be easy to manage.
- **Custom Extensions (50% preservation)**
In some cases the client wants to modify the core business objects or component libraries.
 - ➔ Future version of Project/Open will not take into account the compatibility with custom extensions.
That means that the client will have to port the extensions to the new version.

- Build `_useful_` software
- We teach you how to build real software
 - Learn about high-performance server applications
 - See why Java doesn't work for `_real_` software
 - Learn basics of SQL and Database Management
 - We have prepared some exercises that guide you through the process
 - Learn how to keep a web application running
- Choose a module to concentrate
- Study some background material and become an expert
- Design a user-friendly GUIs and a solid data model
- Being a leading developer with your module, clients will ask for your experience