Design and Implementation of a Flexible RBAC-Service in an Object-Oriented Scripting Language

Mark Strembeck, Gustaf Neumann
Vienna University of Economics and BA
{strembeck|neumann}@wu-wien.ac.at

Presentation Overview

- Roles in general, Role Modeling and Role-Based Access Control
- Object-Oriented Implementation of Dynamic Role Concepts
- The xoRBAC component:
  - Conceptual structure
  - Features
  - Implementation
- Summary and Outlook

What are Roles ?

- Roles are *conceptual entities* used in many different areas, e.g:
  - Sociology and Psychology
  - Object-Oriented Software Construction
  - Computer System Security
- *No common definition* for the Role concept exists
- In general:
  - Roles are used in behavioral modeling
  - Roles enrich the entities they are assigned to with additional behavioral capabilities and/or knowledge

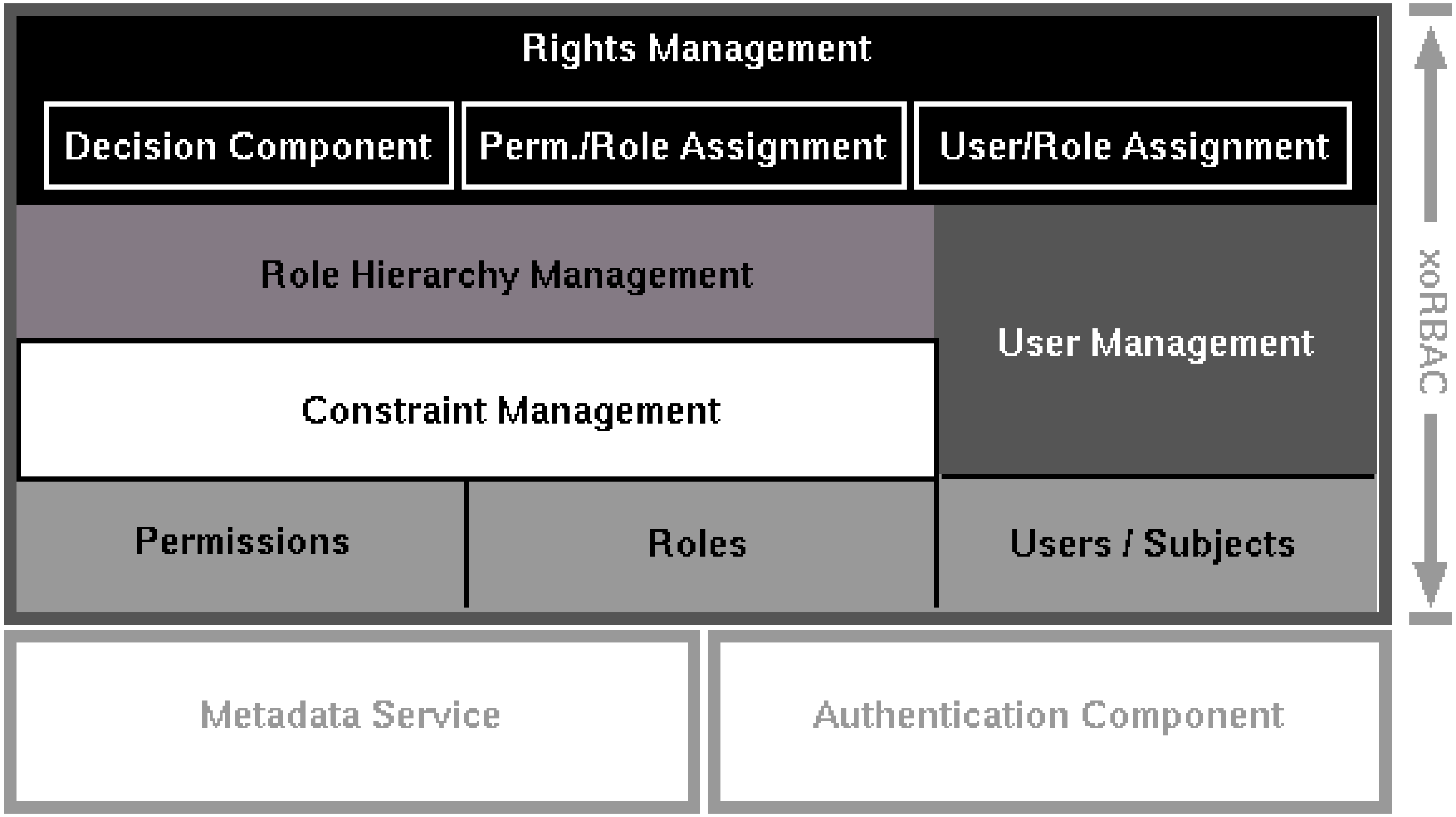
Current Situation in Role Modeling

- Modeling concepts for behavioral models are often role-based
- *Several approaches* for role modeling exist (e.g. in oo-modeling or business process modeling)
- None of the major (OO-)languages offers a *native language construct* for roles
- Implementing role concepts without proper language constructs is comparable to the imitation of OO-concepts in a non-oo-language
- No smooth transition from models to source code ("*semantic-gap*" arises, lack of traceability)

Role-Based Access Control (RBAC)

- RBAC-Roles are:
  - modeled for different work-place profiles and scopes of duty
  - equipped with a number of permissions
  - assigned to users or other "active" entities
- A central RBAC strength: *administration of access rights*
- Recent RBAC concepts comprise:
  - Base Concepts: Users, Roles and Permissions
  - Role-Hierarchies
  - Constraints (esp. separation of duties constraints)

xoRBAC: Conceptual Structure



(Current) Main Features of xoRBAC

- *Many-to-many* user-role and permission-role *assignment* (and revocation)
- Definition of *arbitrary role-hierarchies* (permission-inheritance and constraint-inheritance)
- Definition of *static separation of duties constraints* for both roles and permissions
- Definition of maximum and minimum *cardinalities* for both roles and permissions
- *User-role review* and *permission-role review*
- *Serialization* (export and import) of xoRBAC elements as RDF metadata in XML Syntax

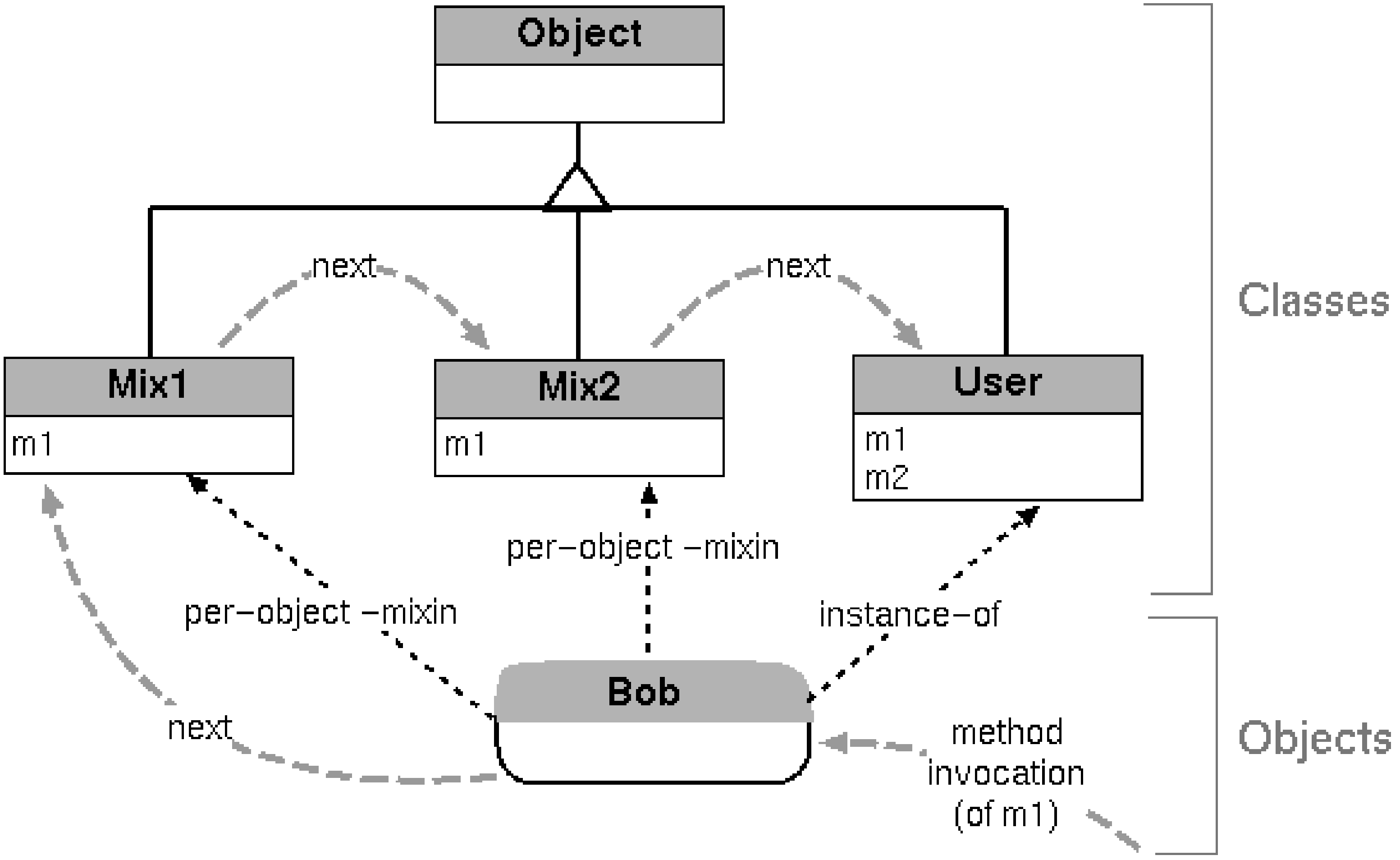
The Need for Dynamic Role Concepts

- RBAC contains many dynamic (implementation level) relations, e.g.:
  - dynamic generation of new roles, permissions or users
  - dynamic user-role and permission-role assignment
  - dynamic definition and deletion of constraints
  - user-role and permission-role review (introspection)
- Benefits of dynamic language constructs for role implementations:
  - more efficient and easier to implement (lessen the "semantic" gap)
  - better traceability of design decisions into source code
  - more comprehensive: improved maintainability, changeability

The XOTcl Language

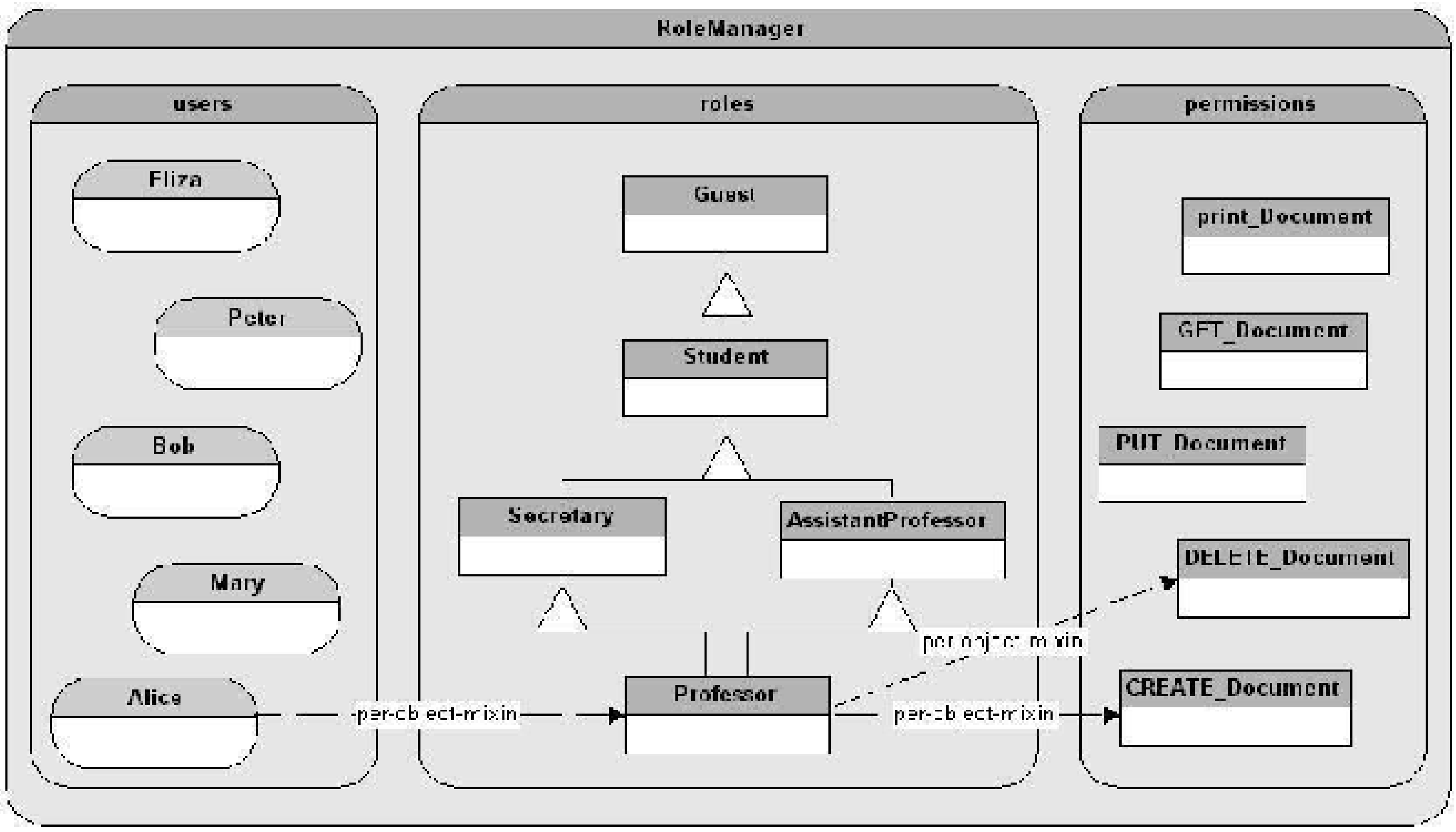
- XOTcl (eXtended Object Tcl) is a *general purpose object oriented programming language*.
- Offers *novel language constructs* originally developed for the *support of design patterns*.
- All language constructs can be applied in a *dynamic fashion.*
  - e.g. redefinition of class/class and class/object relations or
  - the defnition of new classes at runtime
- Support of multiple inheritance and per-object mixins:
  - use of an unambigous "next-path" (essential for name resolution)
  - rich introspection mechanism (e.g. to keep track of dynamic changes)

XOTcl Per-Object Mixins

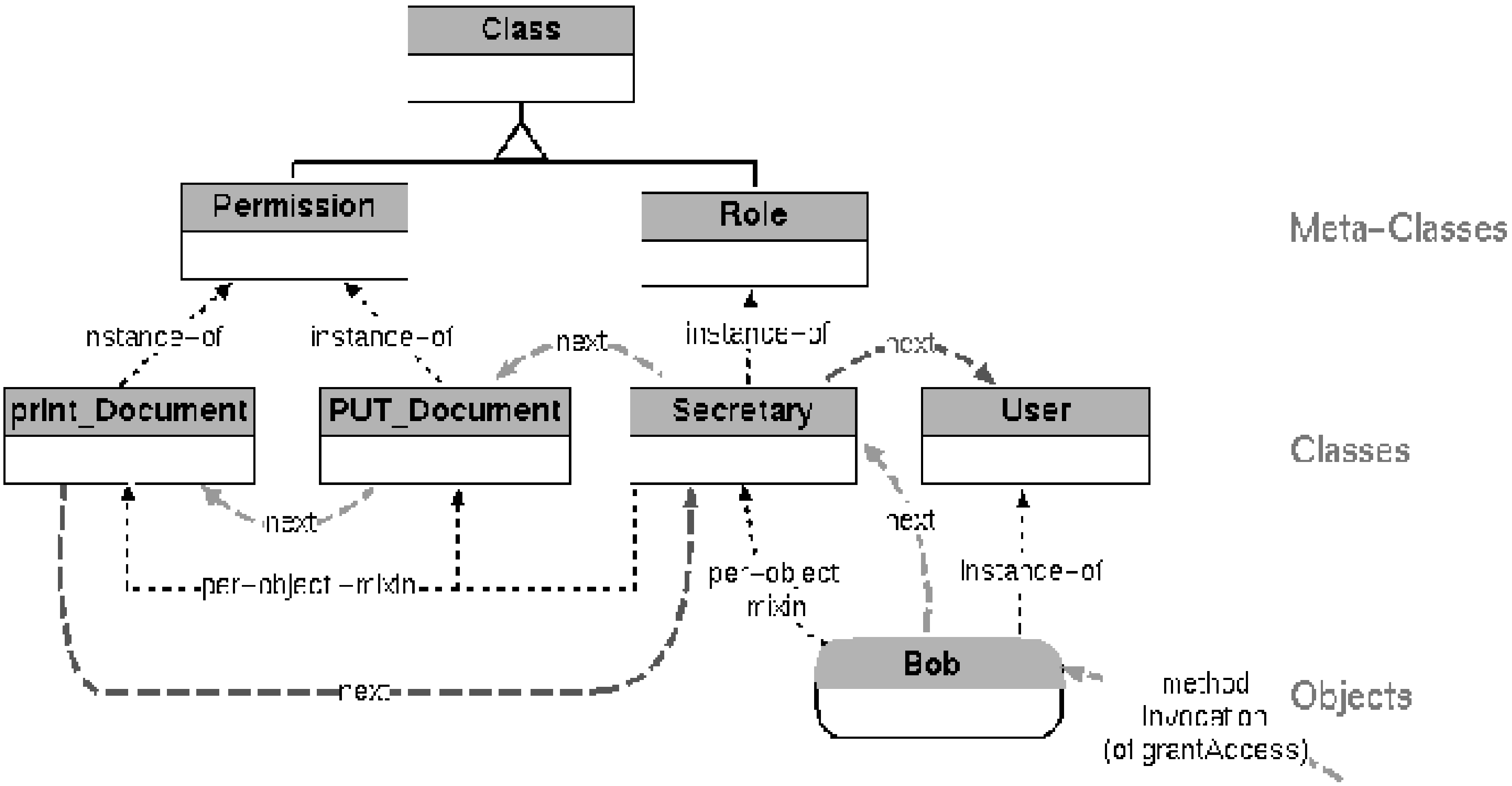


The XOTcl next-path with per-object mixins

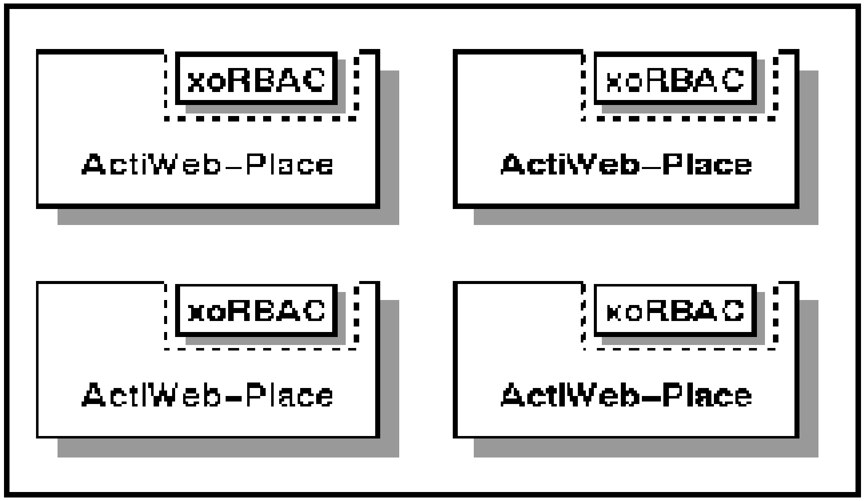
xoRBAC: Runtime View

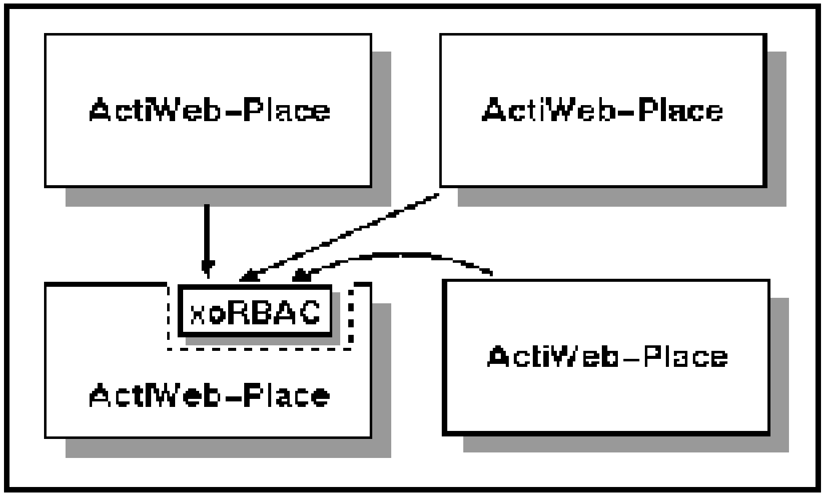


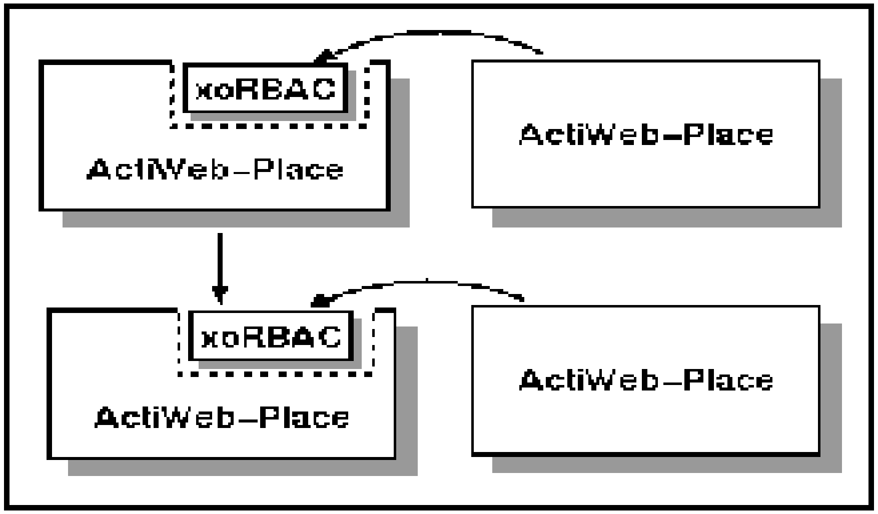
The "grantAccess" Method



xoRBAC for mobile Agents







a) independent services on each ActiWeb-Place

b) central xoRBAC service for several ActiWeb-Places

c) cascading xoRBAC services

Summary and Outlook

- Presentation Summary:
  - xoRBAC provides a flexible RBAC-service implemented with XOTcl.
  - xoRBAC can be reused for arbitrary applications with a C or Tcl linkage on Unix and Windows systems.
  - XOTcl and xoRBAC are publicly available (www.xotcl.org).
  - the current implementation has about 3000 lines of code without comments and blank lines and is subject to a constant improvement and extension process.
- Outlook:
  - SOAP-binding to make xoRBAC available for arbitrary (web) applications
  - Graphical user interface for xoRBAC instances and the corresponding RDF files
  - Support of dynamic separation of duties and other types of constraints