A Resource Access Decision Service for CORBA-based Distributed Systems

Konstantin Beznosov (FIU)
Yi Deng (FIU), Bob Blakley (DASCOM/IBM),
Carol Burt (2AB), John Barkley (NIST)

December 10, 1999
ACSAC ‘99
Presentation Overview

• Problem statement
• RAD logical design
• Discussion
• Status report
• Conclusions
Particular Problem

Application-level access control logic

– reasons to have
  • fine-grain access control (service-oriented systems)
  • policies are complex and/or dynamic
– mixed with application logic
– advantages
– disadvantages
Application AC -- headache for vendors and owners
Disadvantages of embedded authorization logic:

• need to administrate on an application-by-application basis
• multiple access control models
  – difficult to ensure correctness of mapping organizational policy into authorization mechanisms
• difficult to ensure the consistency of changes
• application has to be re-designed/re-implemented/re-tested if authorization logic changes
RAD high-level view

1. Application Request
2. Authorization request
3. Reply to authorization request
4. Reply to application request
RAD Component Collaboration

1: access_allowed(ResourceName, Operation, AttributeList)

2: get_policy_decision_evaluators(ResourceName)

3: get_dynamic_attributes(AttributeList, ResourceName, Operation)

4: combine_decisions(ResourceName, Operation, AttributeList, PolicyEvaluatorList)

5: evaluate(ResourceName, Operation, AttributeList)

6:
RAD Sequence Diagram

an Application Object : AccessDecision
an Access Decision Object : AccessDecision
a Locator : Policy EvaluatorLocator
an Attribute Service : DynamicAttributeService
a Combinator : DecisionCombinator
an Evaluator : PolicyEvaluator

access_allowed(ResourceName, Operation, AttributeList)

g_get_policy_decision_evaluators(ResourceName)

g_get_dynamic_attributes(AttributeList, ResourceName, Operation)

combine_decisions(ResourceName, Operation, AttributeList, PolicyEvaluatorList)

* evaluate(ResourceName, Operation, AttributeList)
Discussion

• Simplicity
  – simple interfaces and data structures
  – nominal amount of data is passed
  – complexity encapsulated in RAD components

• Generality
  – resource and operation names provide generic abstraction
  – generic framework for AC

• Flexibility
  – Existing authorization engines can be used
Discussion (cont’d)

• performance
• scalability
• “resource → resource name” abstraction
• semantics consistency among different RAD components
Prototype Implementations

• 2AB

• Telemed project at Los Alamos Labs
  http://www.acl.lanl.gov/TeleMed/

• FIU
  http://cadse.cs.fiu.edu
Current Status

• OMG pre-final **Resource Access Decision Facility** standard since August 24, 1999.

• DASCOM Inc. announced plans for commercial availability on September 7, 1999

• Center for Advanced Distributed Systems Engineering (CADSE) at FIU continues the research.
Conclusions

Main contributions

• Logical design of generic authorization service.
• Decoupling of authorization logic from application logic can be done.
• Dynamic factors can be supported in authorization process using traditional access matrix as an underlying implementation.