THE UNIVERSITY OF BRITISH COLUMBIA

Security Engineering for Large Scale Distributed Applications

Konstantin Beznosov
Electrical and Computer Engineering
University of British Columbia
http://konstantin.beznosov.net

airplanes vs. cars

- flying is fast
- driving is slow
- why isn’t everybody flying?

why aren't secure systems everywhere?

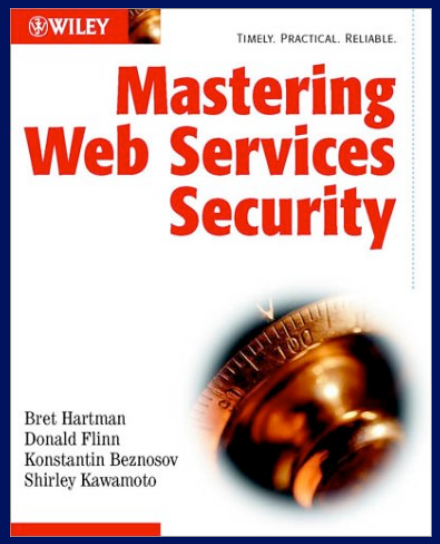
almost completely insecure, or "secure" **but**

- too expensive and error-prone to build
- too complex to administer
- inadequate for real-world problems
- forever

examples

examples

- CORBA Security
  - no compliant system
  - over 600 pages
  - 3 days to install and configure a toy set up
- Web services security
  - harder than RPC-based CORBA



outline

- research direction
- access control mechanisms overview
- some things that can be done about it
- some specific things: attribute function, composable policy engines
- other research projects

what can be done about it?

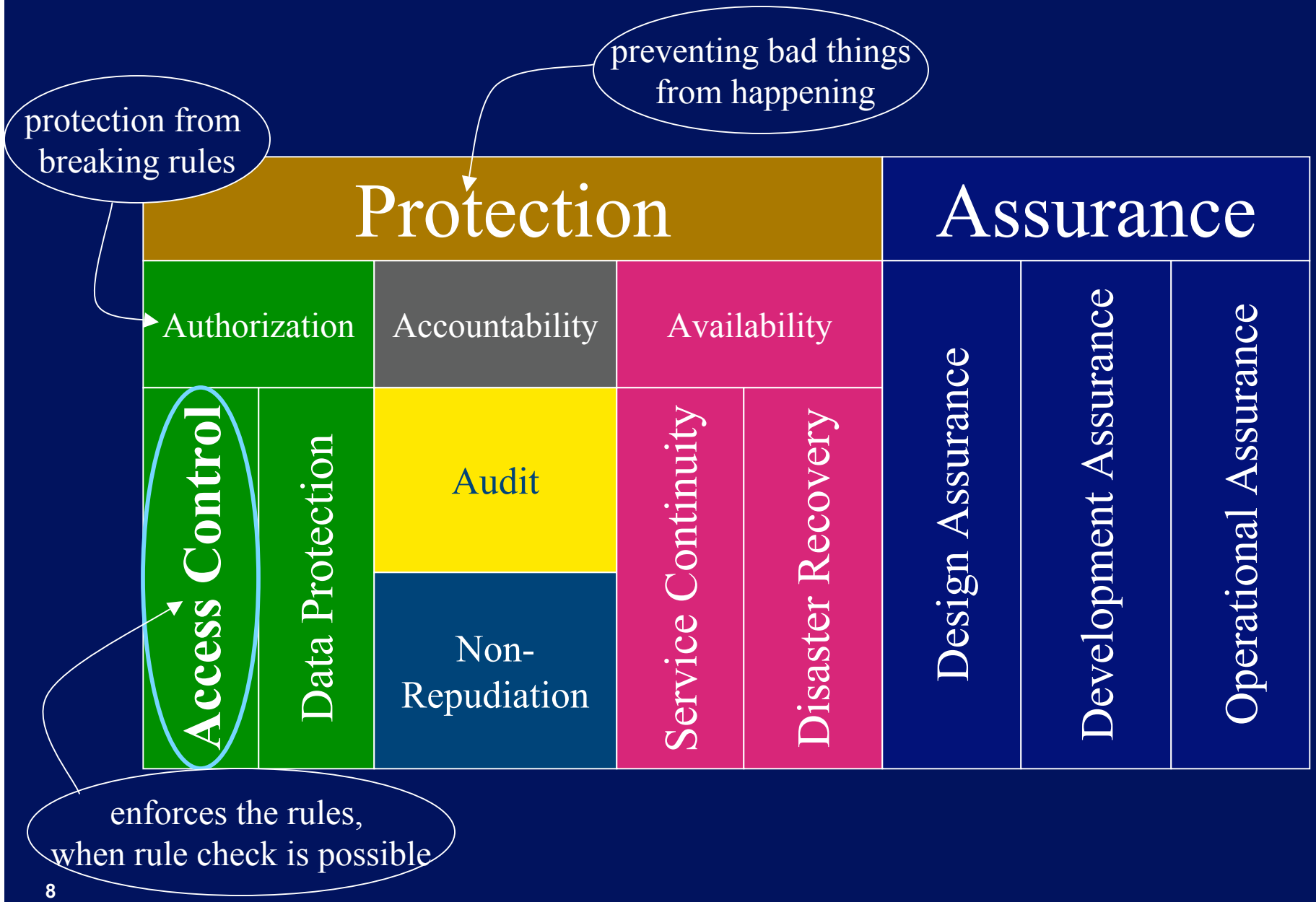
improvements towards

- inexpensive and error-proof to build
- effective and inexpensive in administration
- adequate for problem domains
- easy and inexpensive to change and integrate

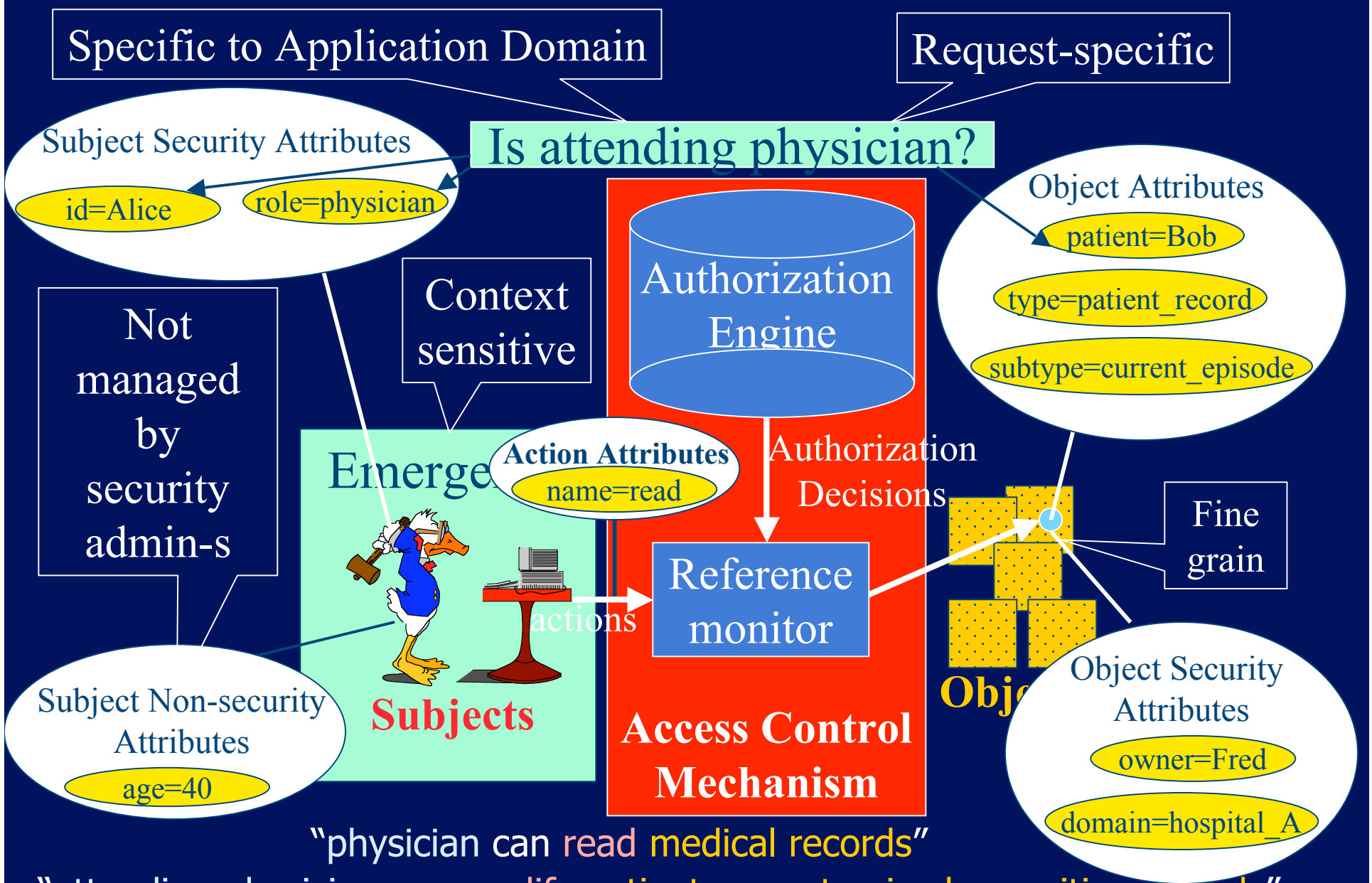
THE UNIVERSITY OF BRITISH COLUMBIA

access control mechanisms overview

conventional computer security



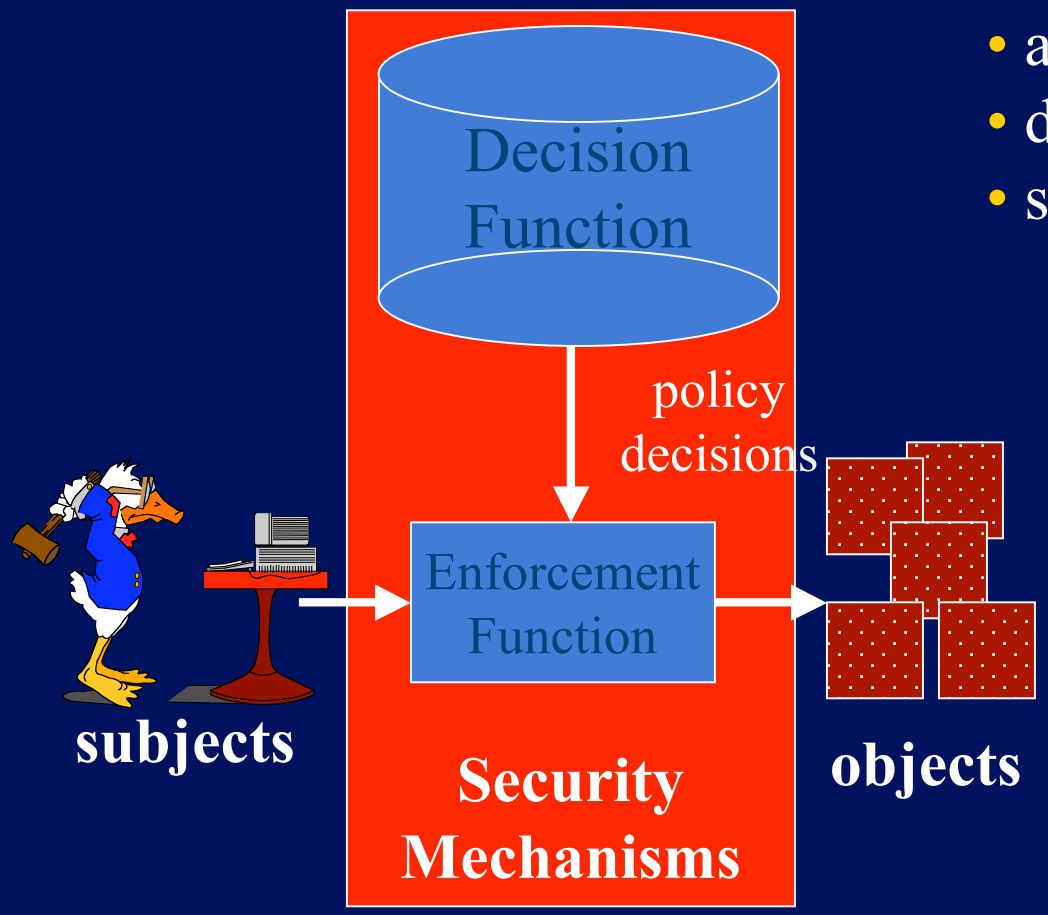
overview of access control



"physician can read medical records"

"attending physician can modify patient current episode sensitive records"

decision-enforcement paradigm



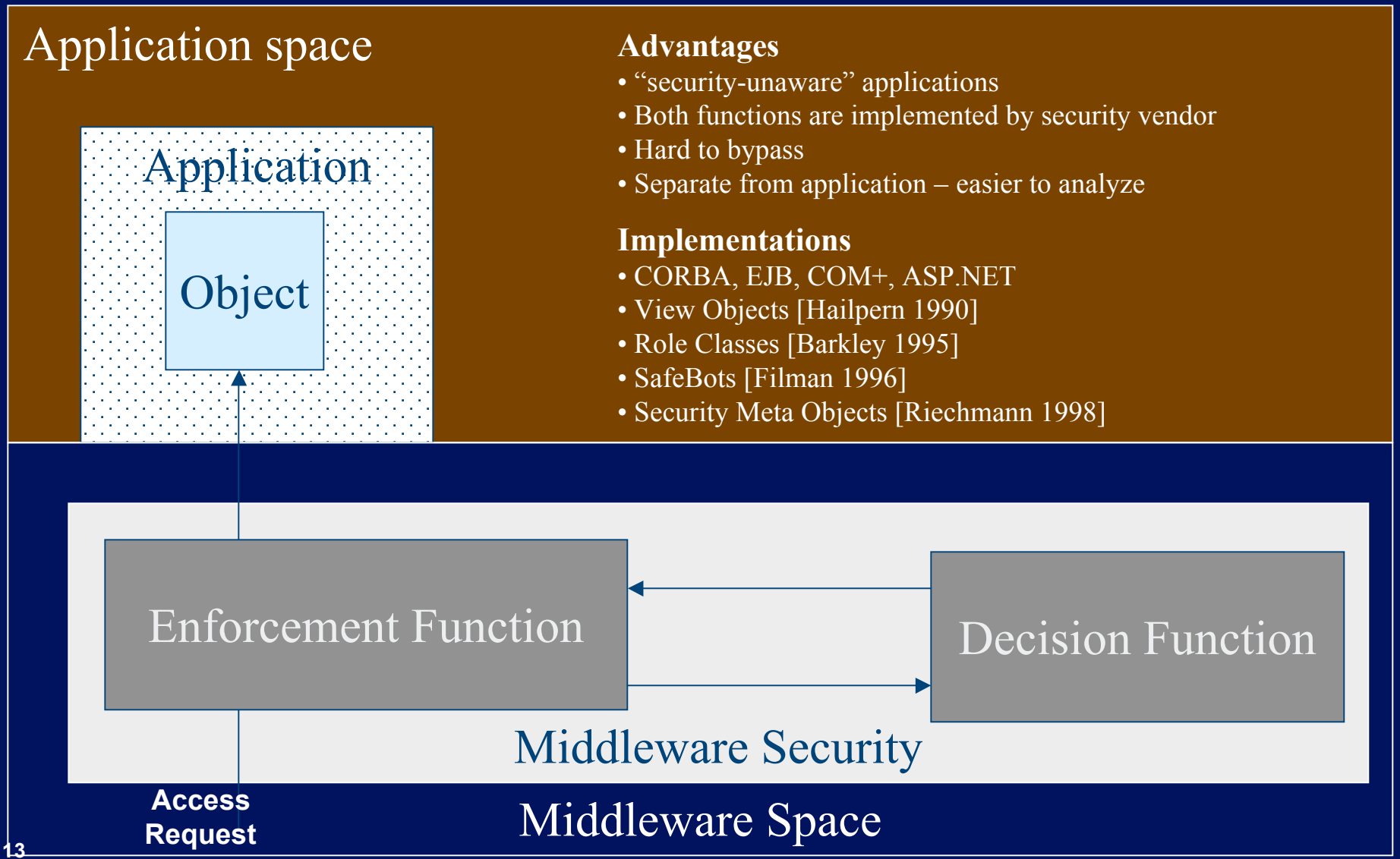
- access control
- data protection
- security audit

some things that can make it better

separation of concerns

- application vendors – sell application(s) product
- middleware vendors – sell middleware products
- security vendors – sell security products
- application owners – sell service(s)

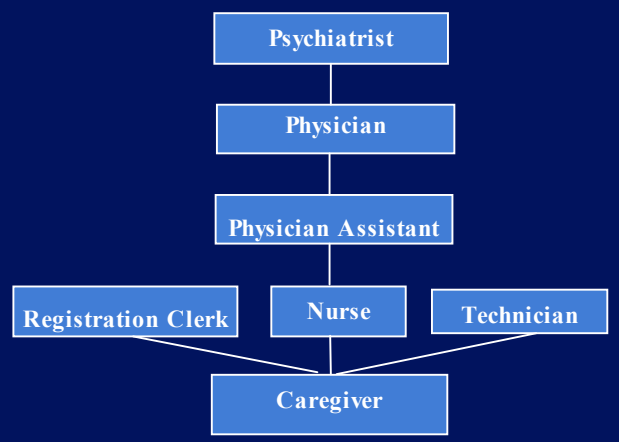
all security in middleware

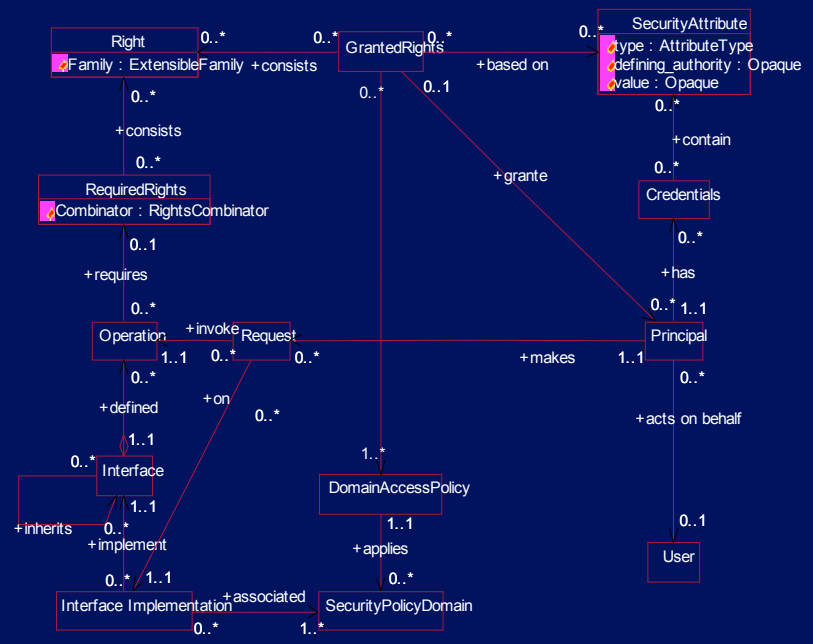


Advantages
- "security-unaware" applications
- Both functions are implemented by security vendor
- Hard to bypass
- Separate from application – easier to analyze

Implementations
- CORBA, EJB, COM+, ASP.NET
- View Objects [Hailpern 1990]
- Role Classes [Barkley 1995]
- SafeBots [Filman 1996]
- Security Meta Objects [Riechmann 1998]

studying DF expressiveness





supports

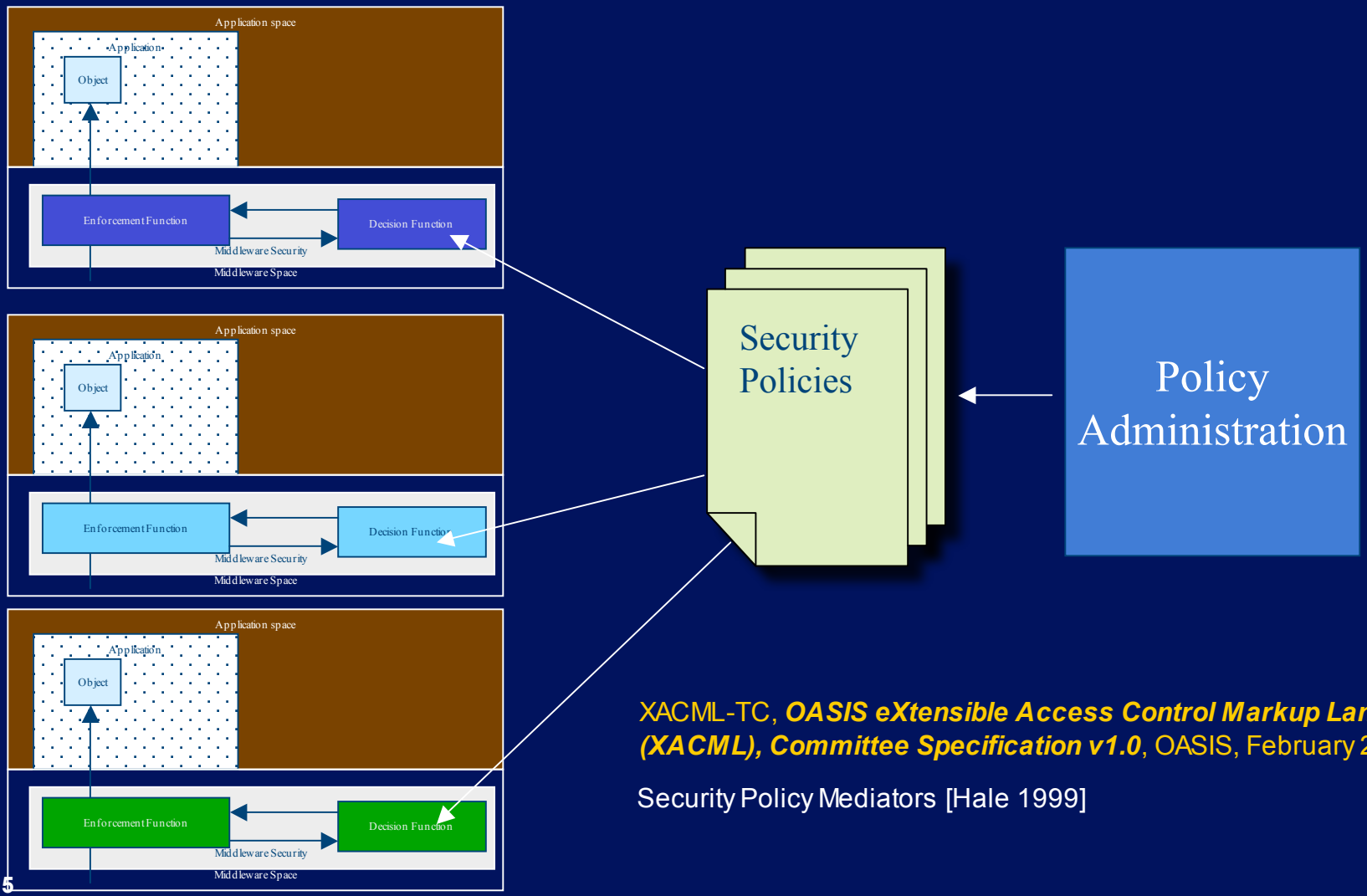
RBAC ← CORBA Security

[Sandhu 1998] → Owner-based DAC

[Sandhu 1996] → Lattice-based MAC

K. Beznosov and Y. Deng, "**A Framework for Implementing Role-based Access Control Using CORBA Security Service**," Fourth ACM Workshop on Role-Based Access Control, Fairfax, Virginia, USA, 1999.

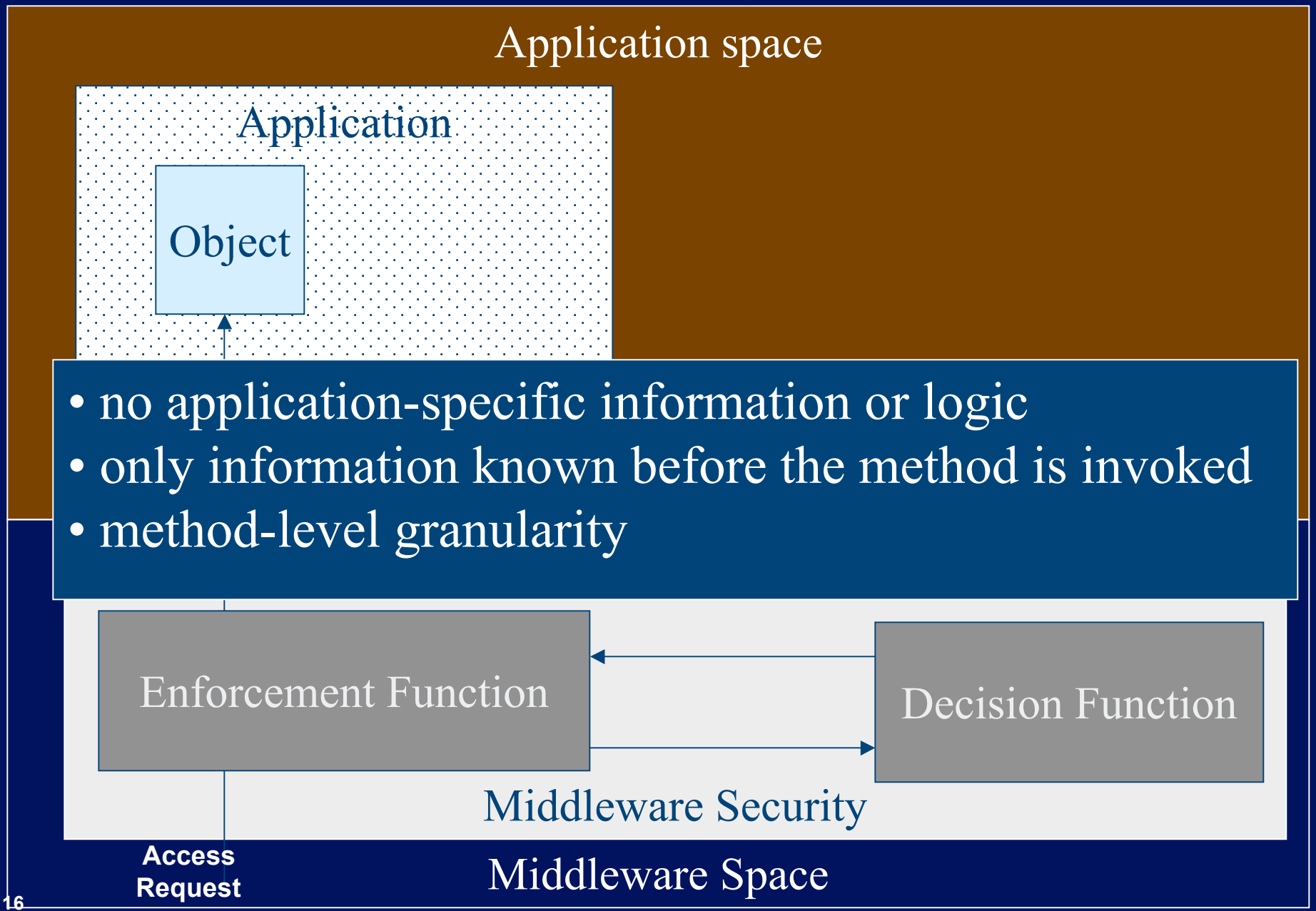
making better to administer



XACML-TC, ***OASIS eXtensible Access Control Markup Language (XACML), Committee Specification v1.0***, OASIS, February 2003.

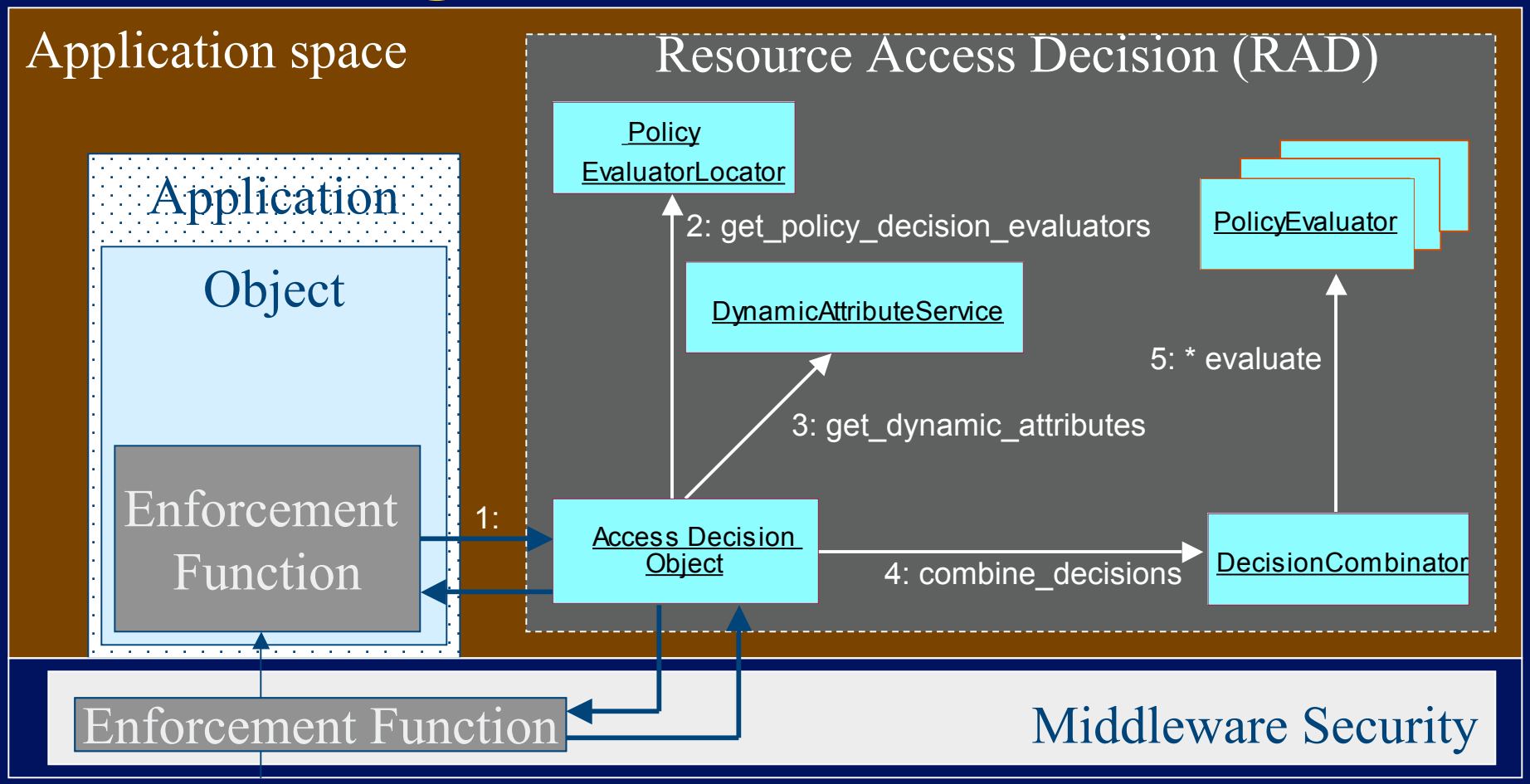
Security Policy Mediators [Hale 1999]

middleware security limitations



- no application-specific information or logic
- only information known before the method is invoked
- method-level granularity

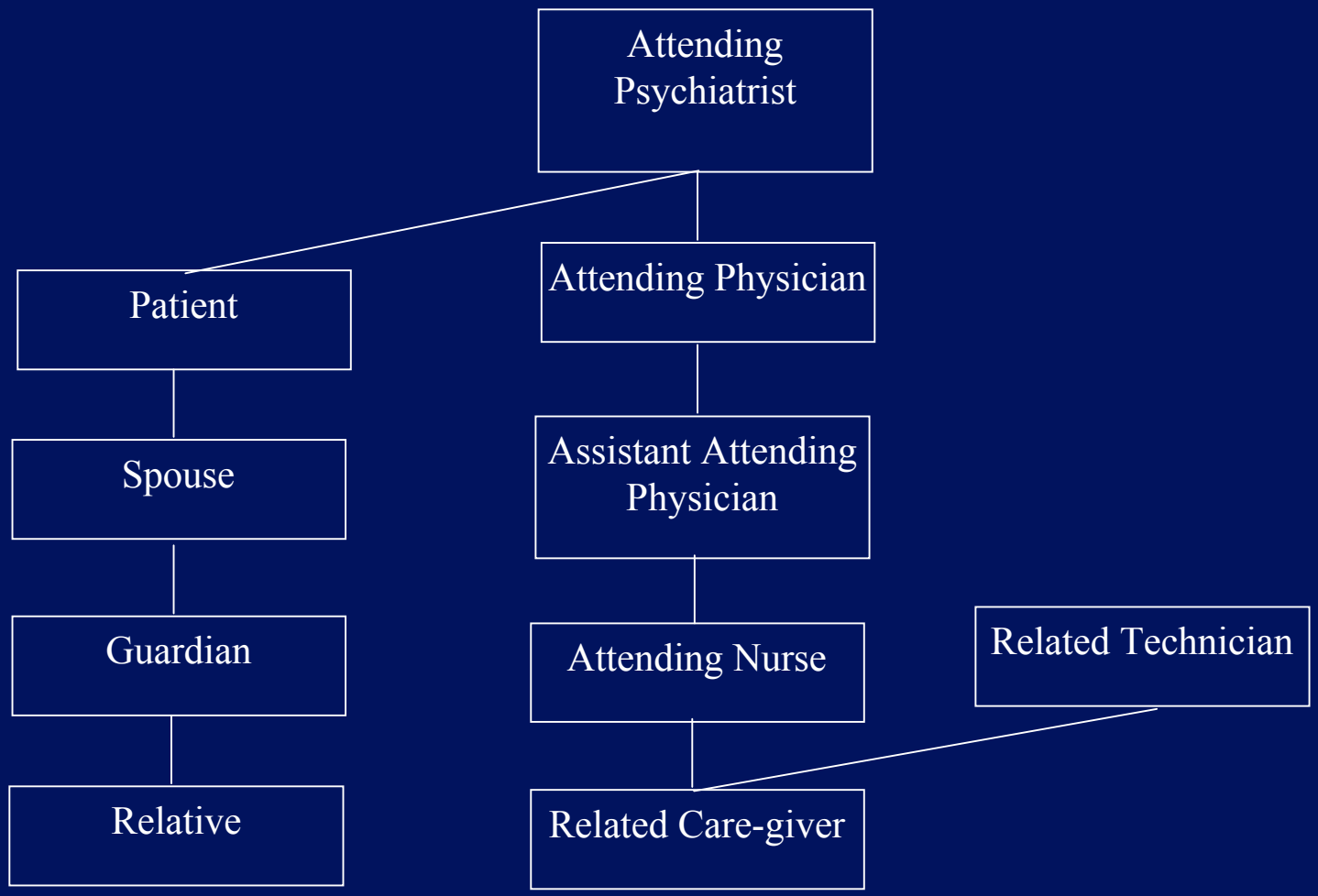
reconfigurable decision function



– K. Beznosov, Y. Deng, B. Blakley, C. Burt, and J. Barkley, "**A Resource Access Decision Service for CORBA-based Distributed Systems**," Annual Computer Security Applications Conference (ACSAC), Phoenix, Arizona, USA, 1999.

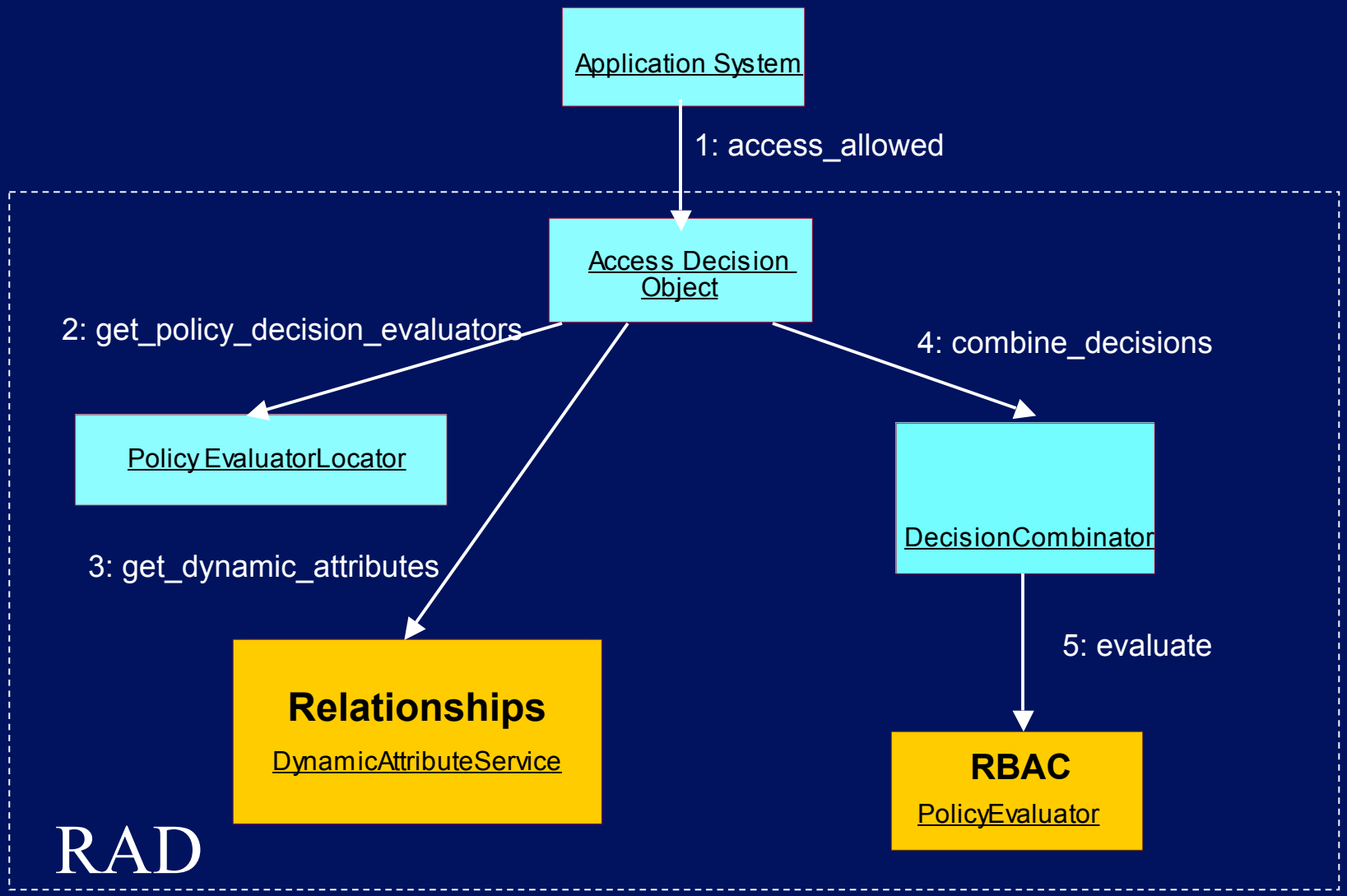
– OMG, ***Resource Access Decision Facility***, Object Management Group, OMG document number: formal/2001-04-01, August 2001.

relationship-based access control



J. Barkley, K. Beznosov, and J. Uppal, "**Supporting Relationships in Access Control Using Role Based Access Control**," Fourth ACM Role-based Access Control Workshop, Fairfax, Virginia, USA, 1999.

RBAC ⊗ RAD == RelBAC



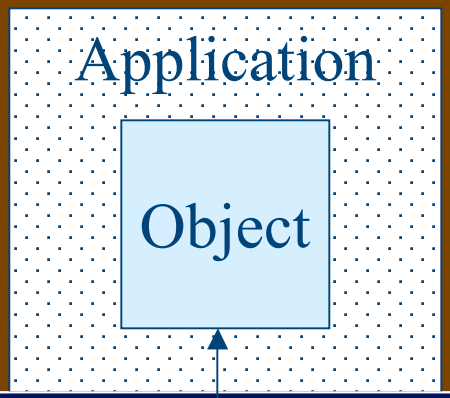
one specific thing:
attribute function

enforcement in middleware

Application space

Advantages
- EF stays outside of the application
- DF can use application-specific policies and/or information

Examples
- CORBA Security replaceable mech.
- Java Authorization Contract for Containers, (JACC) in J2EE v1.4
- getAccess, Access Manager, SiteMinder
- Legion [Grimshaw 1998]

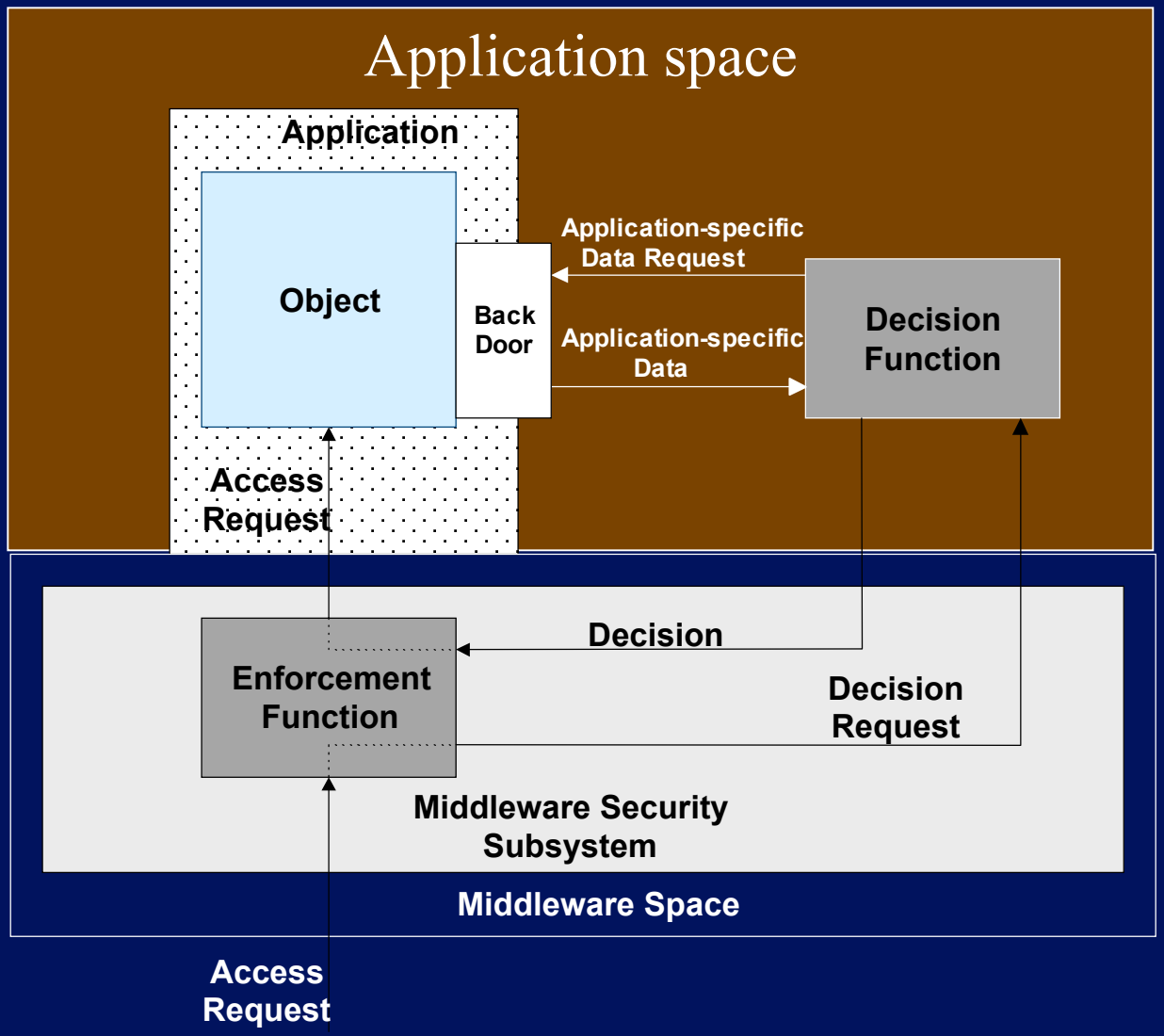


Middleware Security

Access Request

Middleware Space

how to get application data for decisions?

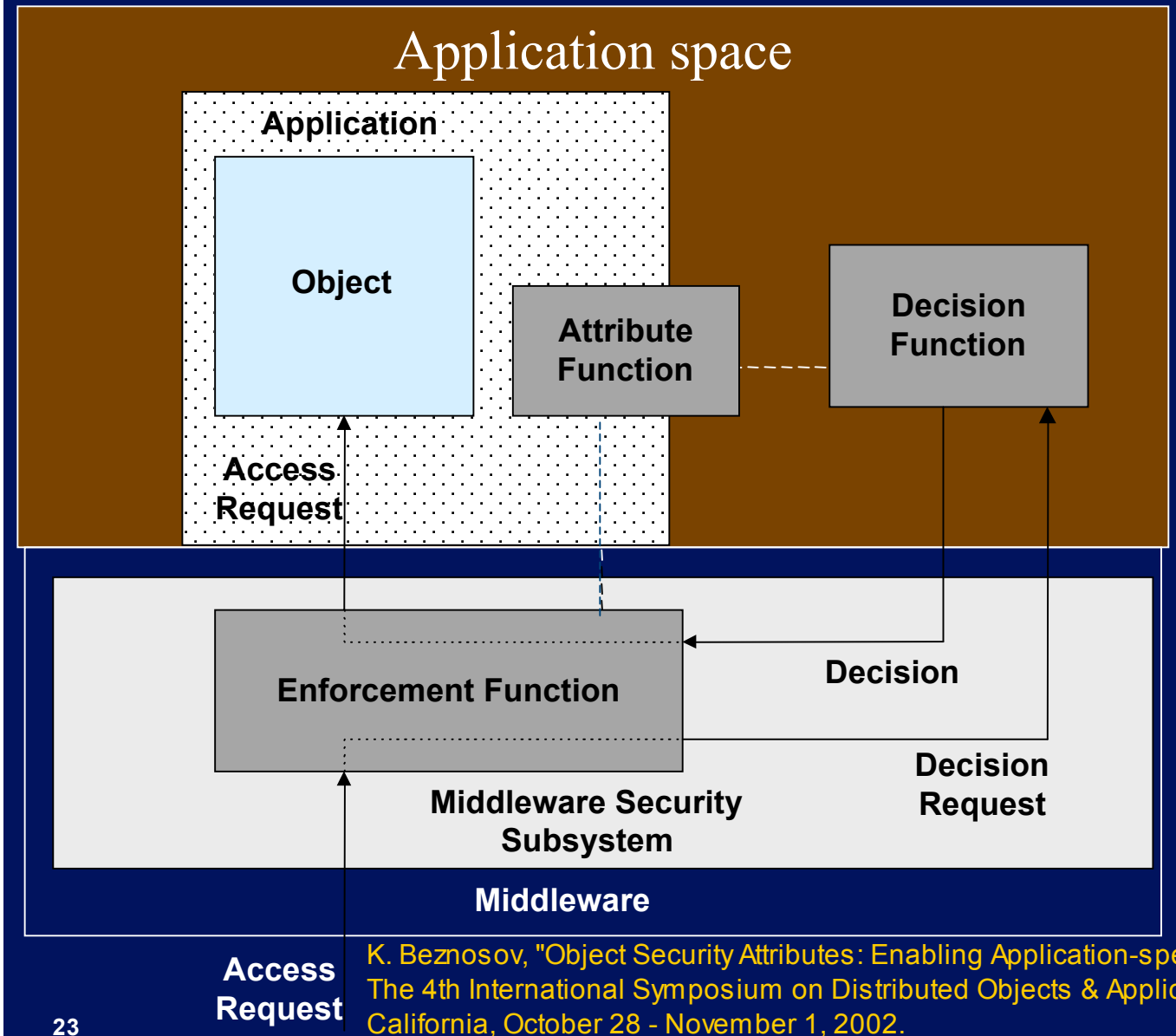


Coupled with the object

Disadvantages

- Each business object has to implement the backdoor
- Could be inefficient on expensive to activate objects
  - Weak in the face of denial of service attacks

a better way – Attribute Function



Advantages

+ security out
+ application data in
+ separation of concerns
  - EF – middleware vendor
  - DF – authorization vendor
  - AF – application owner

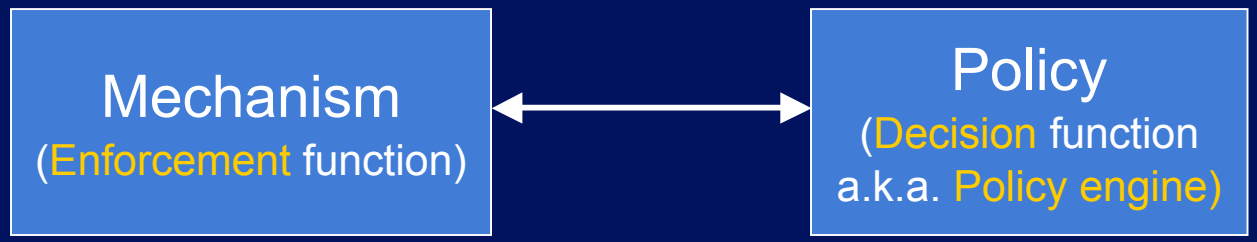
K. Beznosov, "Object Security Attributes: Enabling Application-specific Access Control in Middleware," The 4th International Symposium on Distributed Objects & Applications, pp. 693-710, Irvine, California, October 28 - November 1, 2002.

THE UNIVERSITY OF BRITISH COLUMBIA

another specific thing: composable policy engines

K. Beznosov, "**On the Benefits of Decomposing Policy Engines into Components**," in *Proceedings of The 3rd Workshop on Reflective and Adaptive Middleware*, Toronto, Canada. October 19 2004.

problem motivation



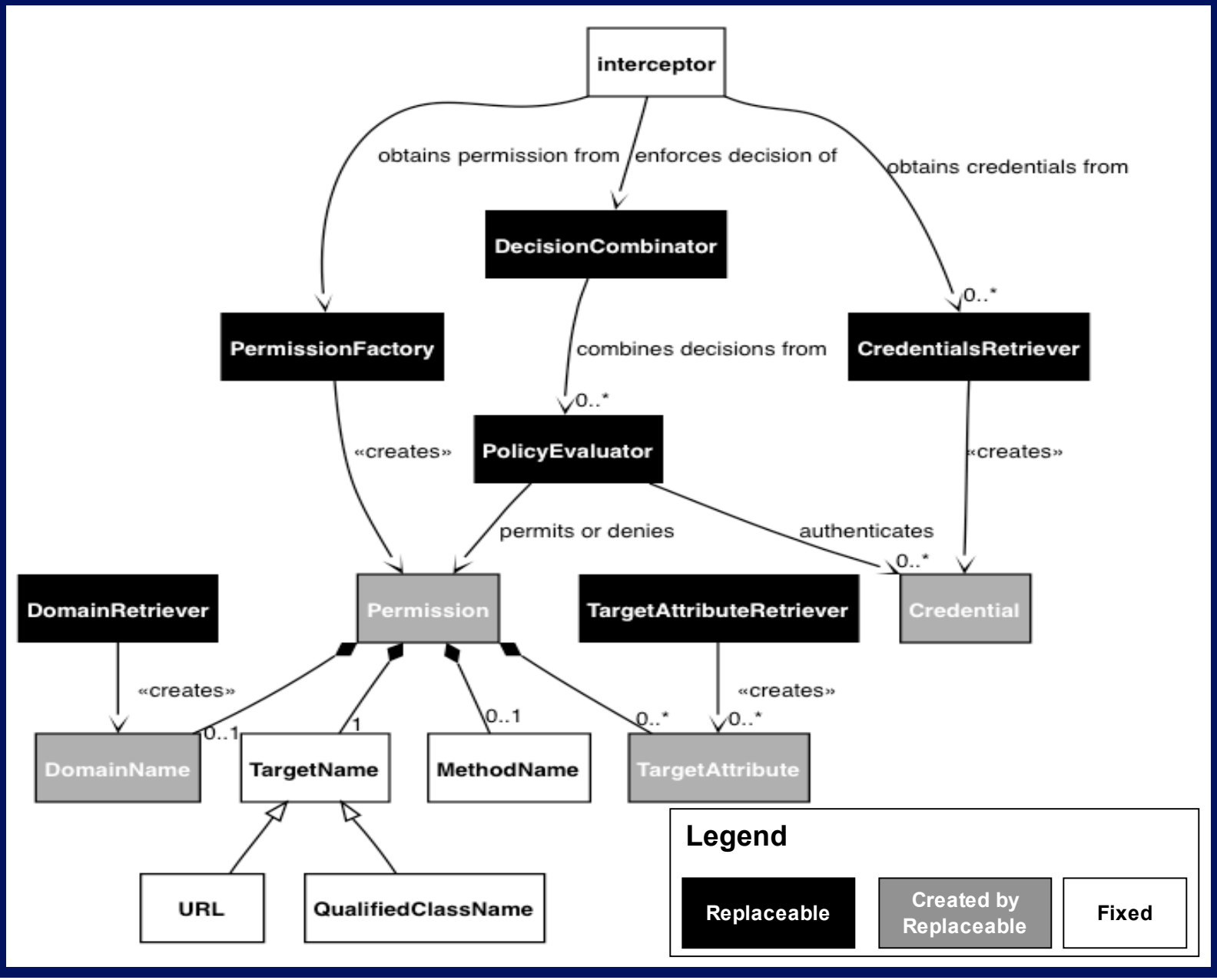
Distributed app. developers/admins have limited choices:

1. Pre-built policy engines with limited capabilities
   - e.g., JAAS default policy file, COM+, EJB authorization
   - Limited support for non-trivial or application-specific policies
2. Pre-built policy engines "one size fits all" generic
   - e.g., CORBA
   - Unnecessary complex and expensive to use
3. "plug-in" APIs for creating custom "do-it-yourself" engines
   - e.g., CORBA Sec. Replaceable, JSR 115, SiteMinder and alike
   - Hard to do it right

premise

- common policy elements
  - e.g., authorizations based on roles, groups, location
- differences in
  - the weight and composition
    - e.g., location || (role && group) vs.
      role || (location && group)
  - application-specific factors
    - e.g., relations, certification, license

component framework for A&A policy engine



expected benefits

- wide range of supported policies
- "pay as you go" cost of supporting a policy
  - determined by required policy
    - not by policy engine complexity
  - incremental changes proportional to policy Δ-s
    - addition/removal/re-composition of policy components
    - re-use of existing policy logic by developers/administrators

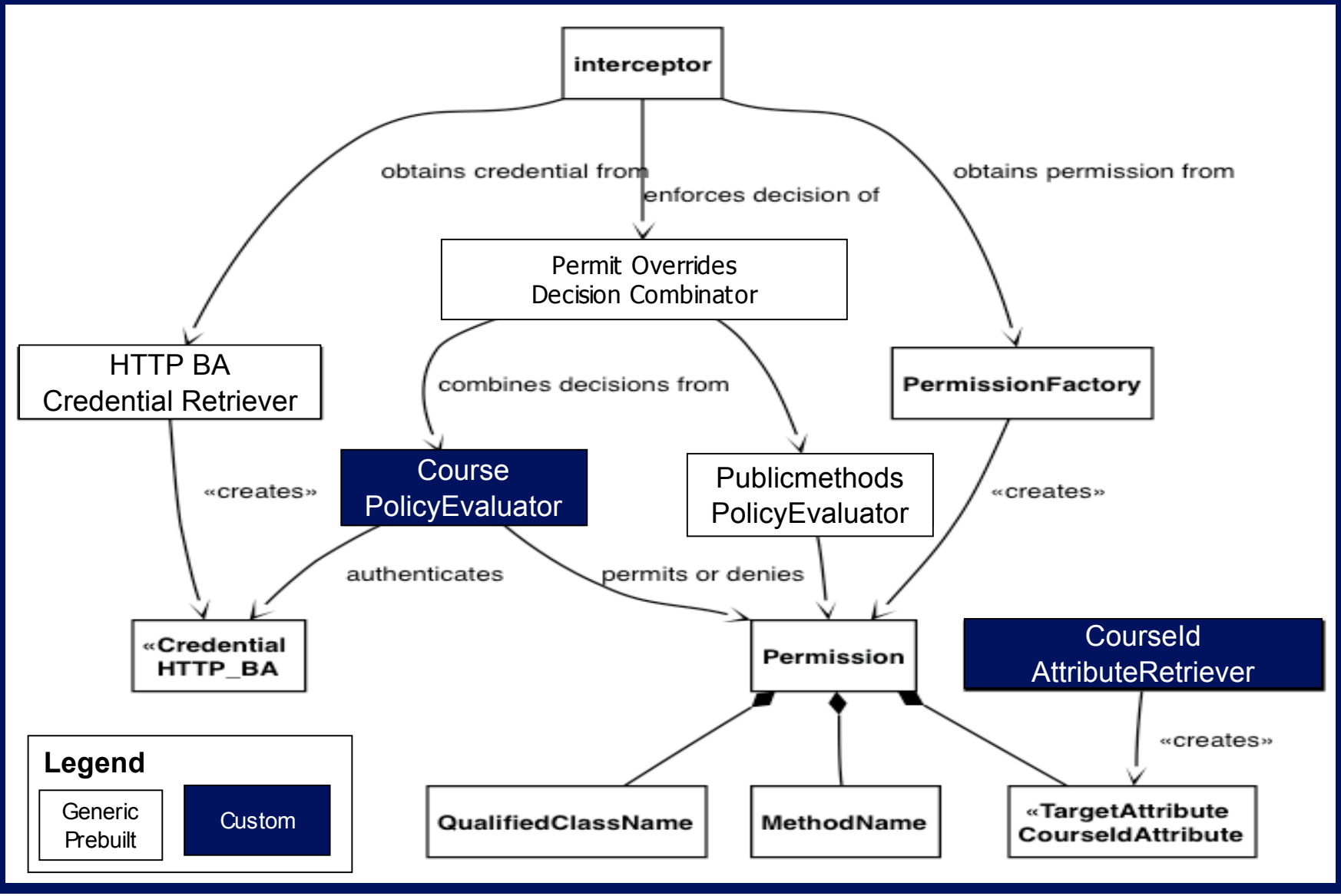
example 1

university course web service

university course web service **policy**

1. Anyone can lookup course descriptions.
2. All users should authenticate using HTTP-BA.
3. Registration clerks can list students registered for the course and (un)register students.
4. The course instructor can list registered students as well as manage course content.
5. Registered for the course students can download assignments and course material, as well as submit assignments.

policy engine assembly for example 1



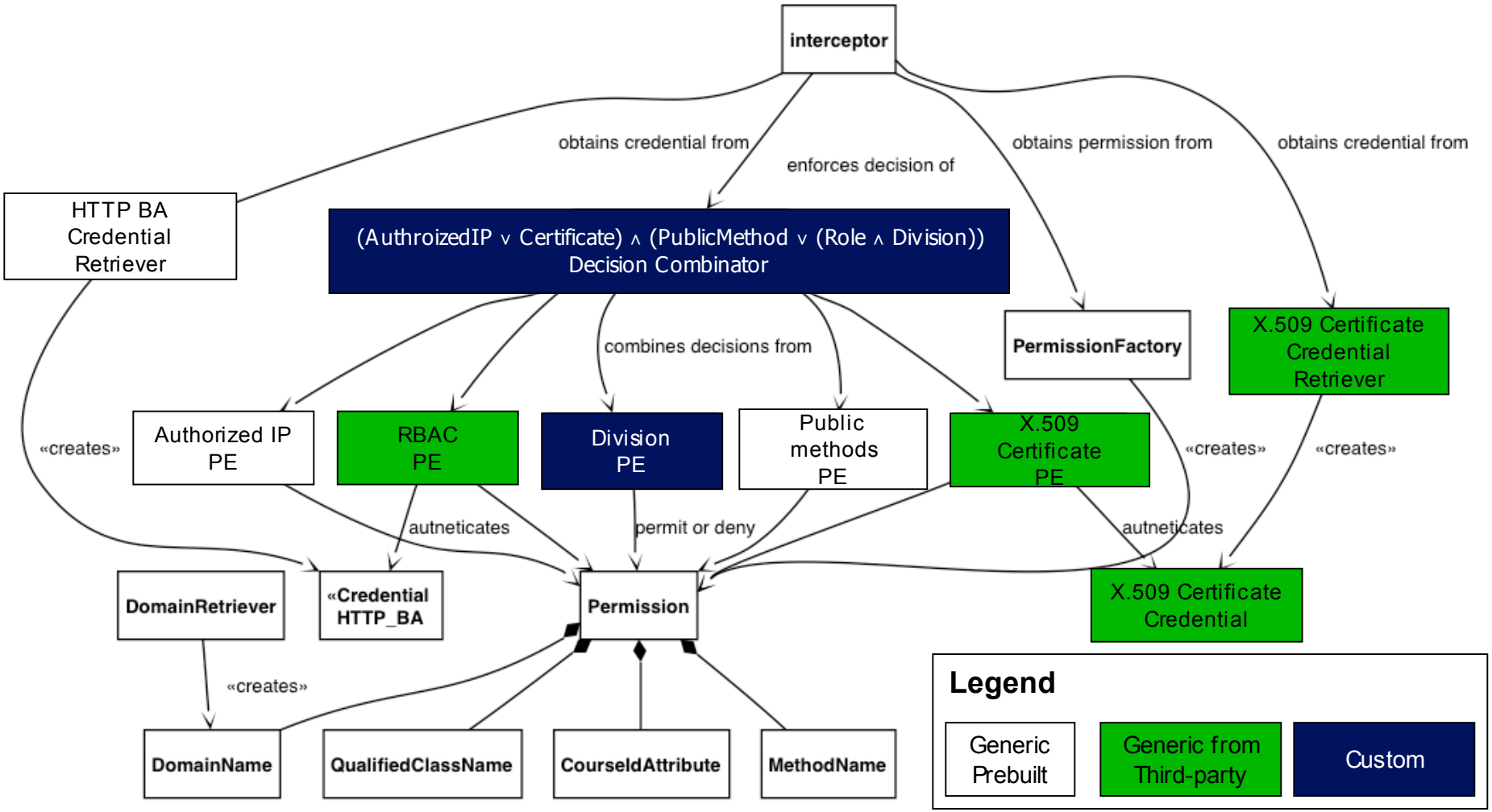
example 2

human resources web service for an international organization

HR web service policy

1. Only users within the company's intranet or those who access the service over SSL and have valid X.509 certificates issued by the company should access.
2. Anybody in the company can look up any employee and get essential information about her/him.
3. HR employees can modify contact information and review salary information of any employee from the same division.
4. HR managers can modify any information about the employees of the same division.

policy engine assembly for example 2

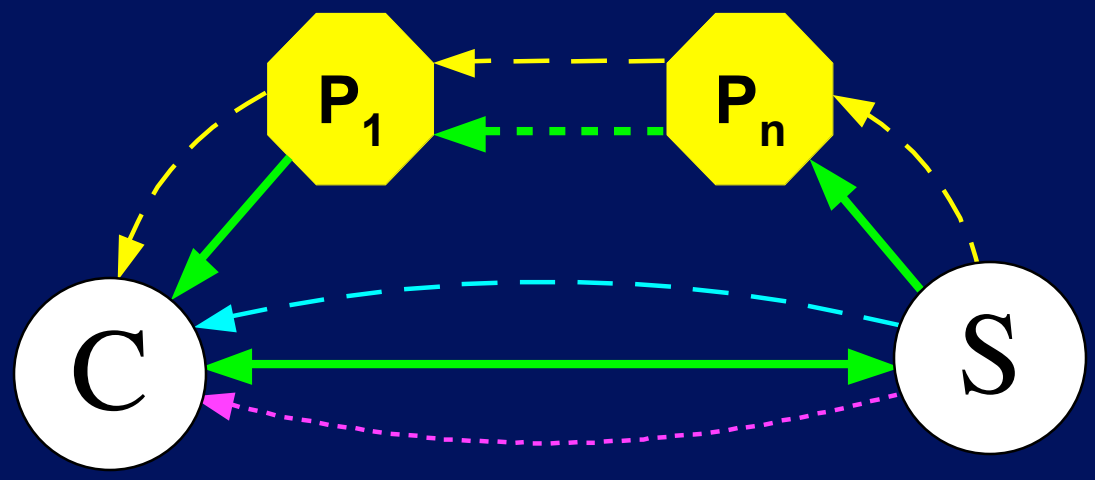


summary

- what
  - adequate for different application domains
  - inexpensive and error-proof to build
  - effective and inexpensive in administration and management
  - easy and inexpensive to change, and replace
- how
  - RBAC in CORBA
  - XACML
  - Resource Access Decision (RAD)
  - RelBAC
  - attribute function
  - composable policy engines

other research projects

multiple-channel SSL



- end-to-end security with partially trusted proxies
- selective data protection

usability of security administration

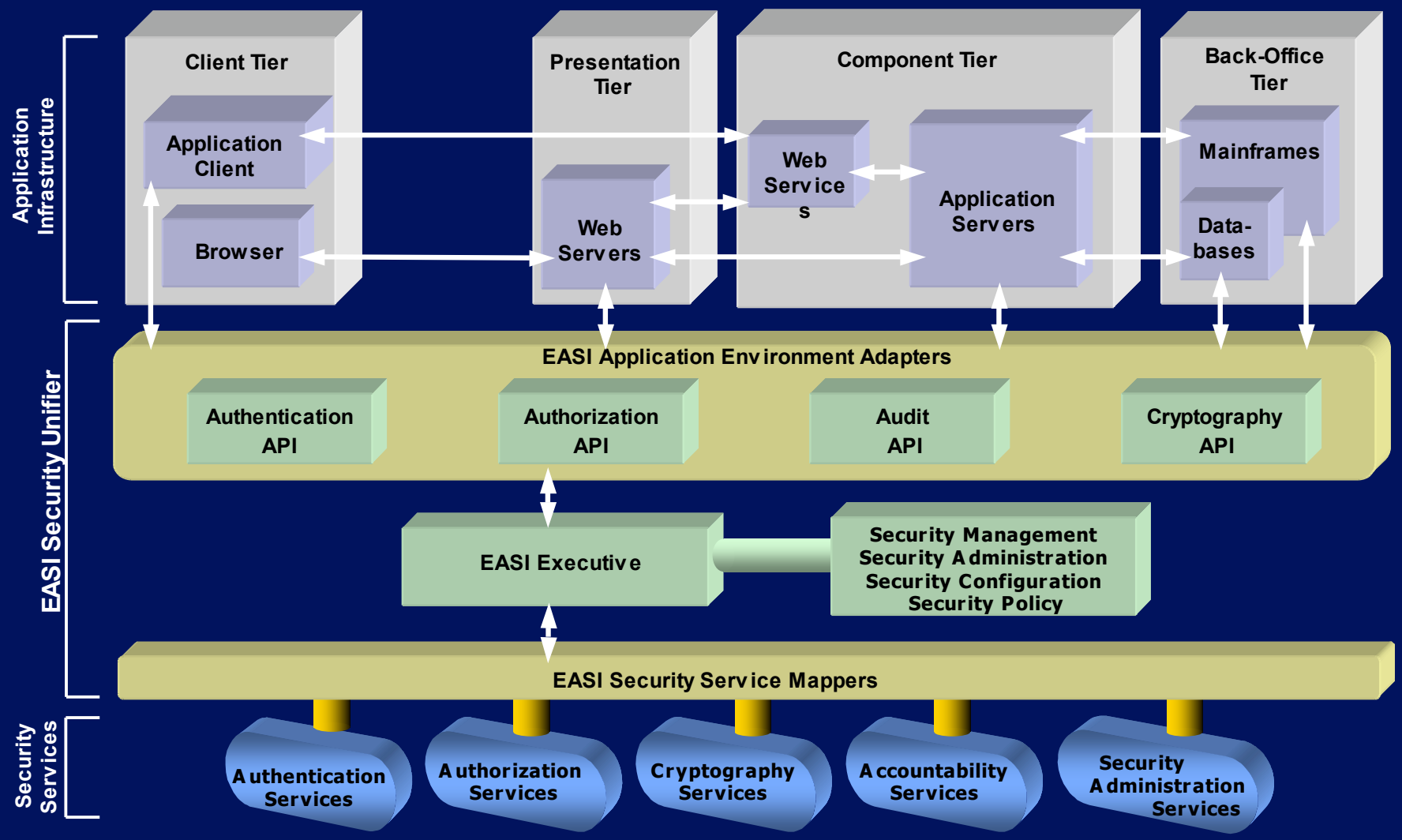
- improving visualization of the information
  - existing cognitive models of security administration
- improving feedback to security administrators
  - "what if" scenarios
  - safe staging playgrounds
  - testing system state
- better cognitive models
- mappings between different mental models/abstractions
  - application-specific model oriented on business processes
  - mechanism-specific technical model

agile security assurance

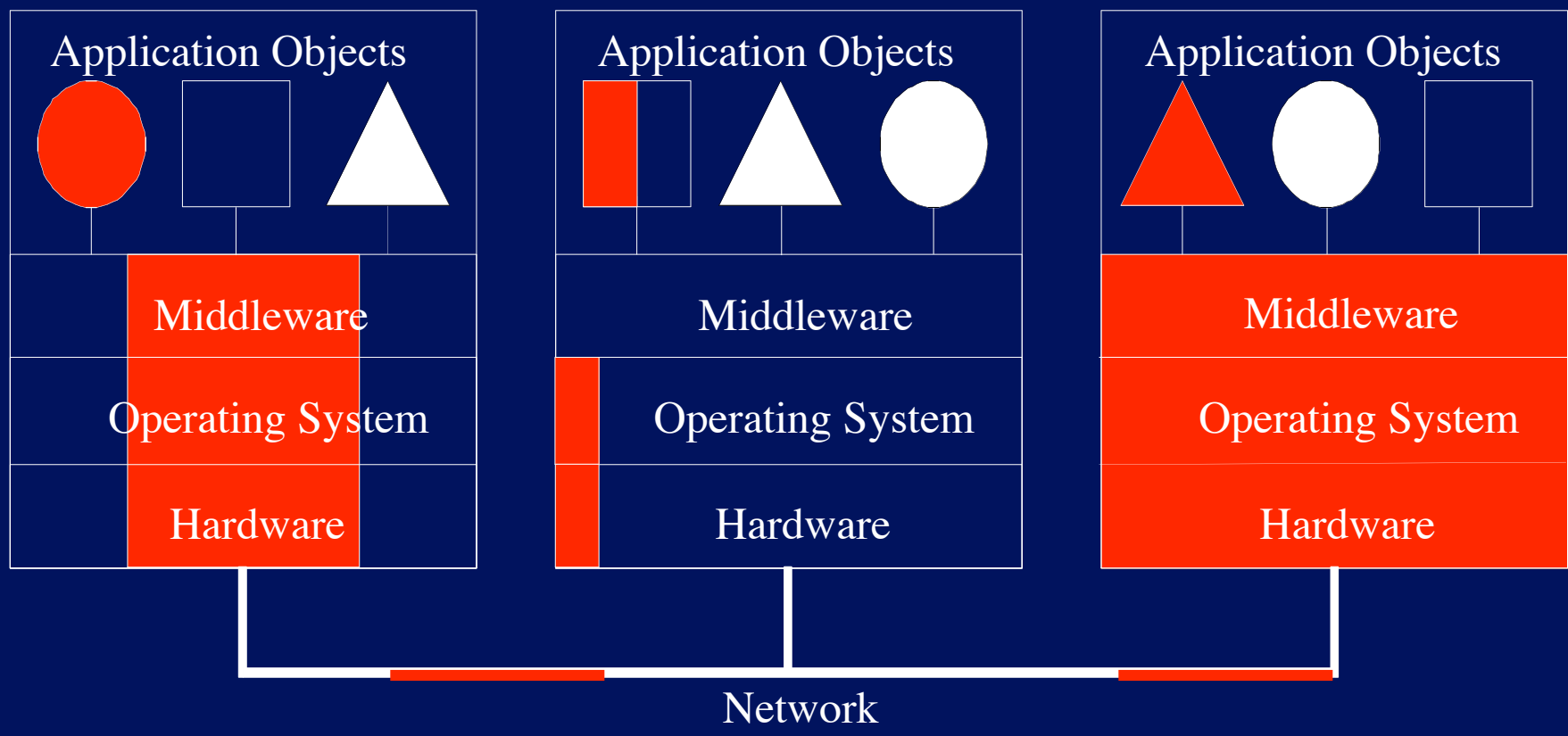
1. examined the mismatch between security assurance and agile methods

2. classified conventional security assurance practices according to the degree of clash
   - 1) natural match, 2) methodology nuetral, 3) (semi-)atomatable, 4) complete mismatch
3. suggested ways of alleviating the conflict
   - tool support, knowledge codification, agile-friendly assurance, intermittent assurance

Advanced ADAE/ADME Scheme



security diffuses in applications



attribute function in CORBA