THE UNIVERSITY OF BRITISH COLUMBIA

Experience Report: Design and Implementation of a Component-Based Protection Architecture for ASP.NET Web Services

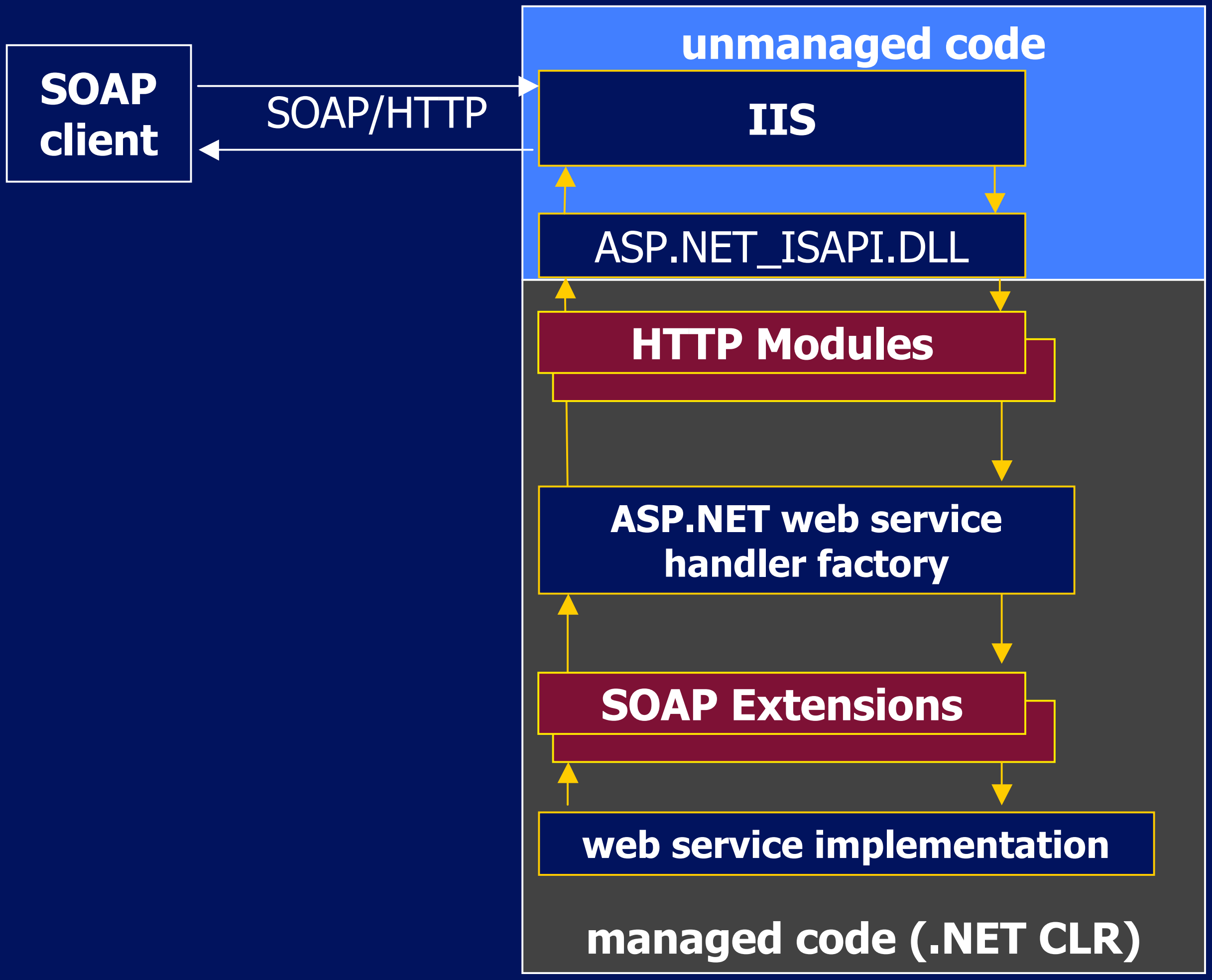
Konstantin Beznosov

Laboratory for Education and Research in Secure Systems Engineering (LERSSE)

Electrical and Computer Engineering

University of British Columbia

How ASP.NET Web Services Work



ASP.NET Web Services Security

Disclaimer: Biased, qualitative, unsupported comparison

	Out-of-the-box	Reported Architecture
granular	***	*****
scalable	****	*****
extensible	*	*****
reusable	***	*****

Outline

- System architecture
- Examples
- Lessons learned
- Summary

Reported System

What is it?

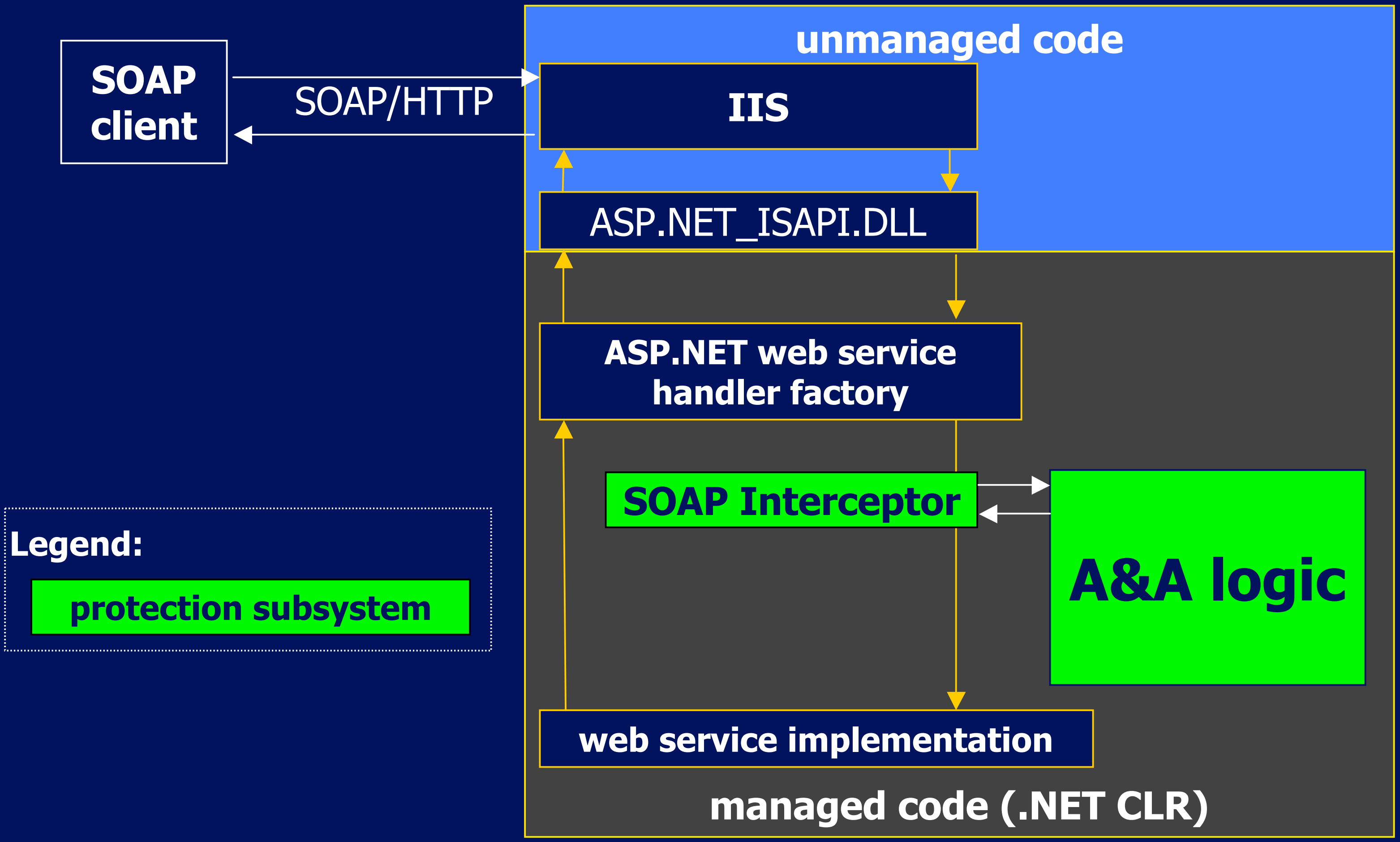
Component-based
Authentication and **A**uthorization (A&A) architecture
for ASP.NET Web services

Key features

Less effort to integrate into enterprise security

1. More granularity and scalability: scalable and fine-grained configuration of machine-wide A&A functions
2. More extensibility: easy to add new A&A logic
3. Better reusability: A&A components can be combined

Separation of Enforcements & Decisions



SOAP client

SOAP/HTTP

unmanaged code

IIS

ASP.NET_ISAPI.DLL

ASP.NET web service handler factory

SOAP Interceptor

A&A logic

web service implementation

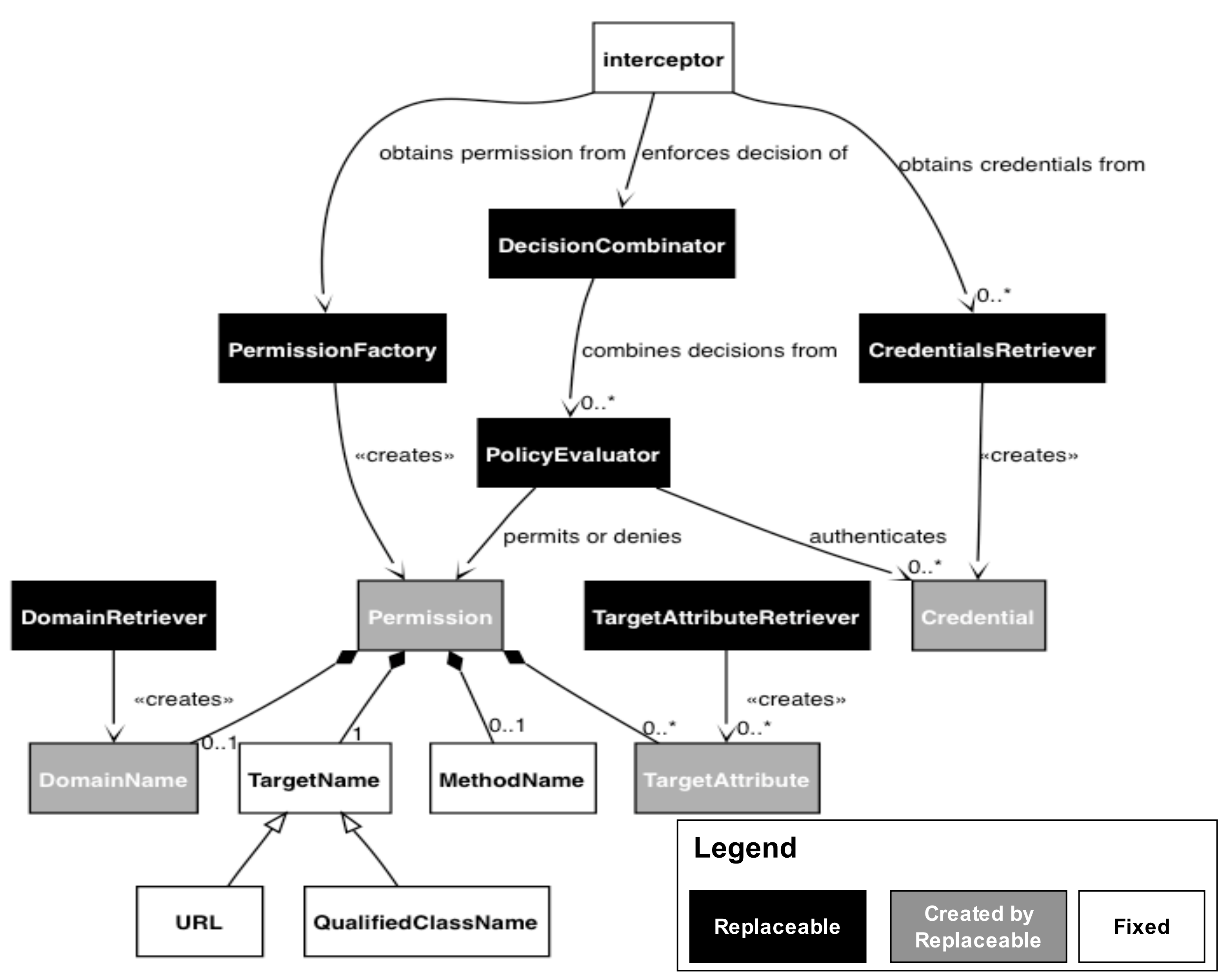
managed code (.NET CLR)

Legend:

🟩 protection subsystem

Interceptor enforces, “A&A logic” decides

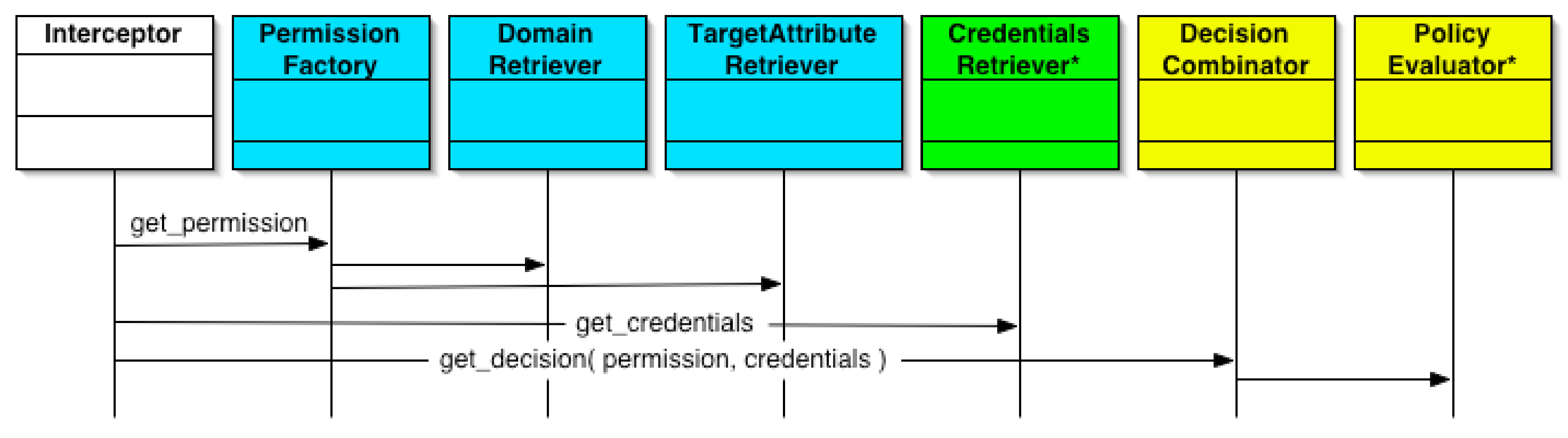
Component Framework for A&A Logic



Permission Examples

Permission Example	**Explanation**
http://foobank.com/bar.asmx	Only the URL is used
com.foobank.ws.Sbar/m1	Class and method names
D1/com.foobank.ws.Sbar/m1	Same but in domain “D1”
com.foobank.ws.Sbar/owner=smith	Class name and attribute
D1/com.foobank.ws.Sbar/owner=smith/m1	Domain / class / attribute / method

Call Sequence



THE UNIVERSITY OF BRITISH COLUMBIA

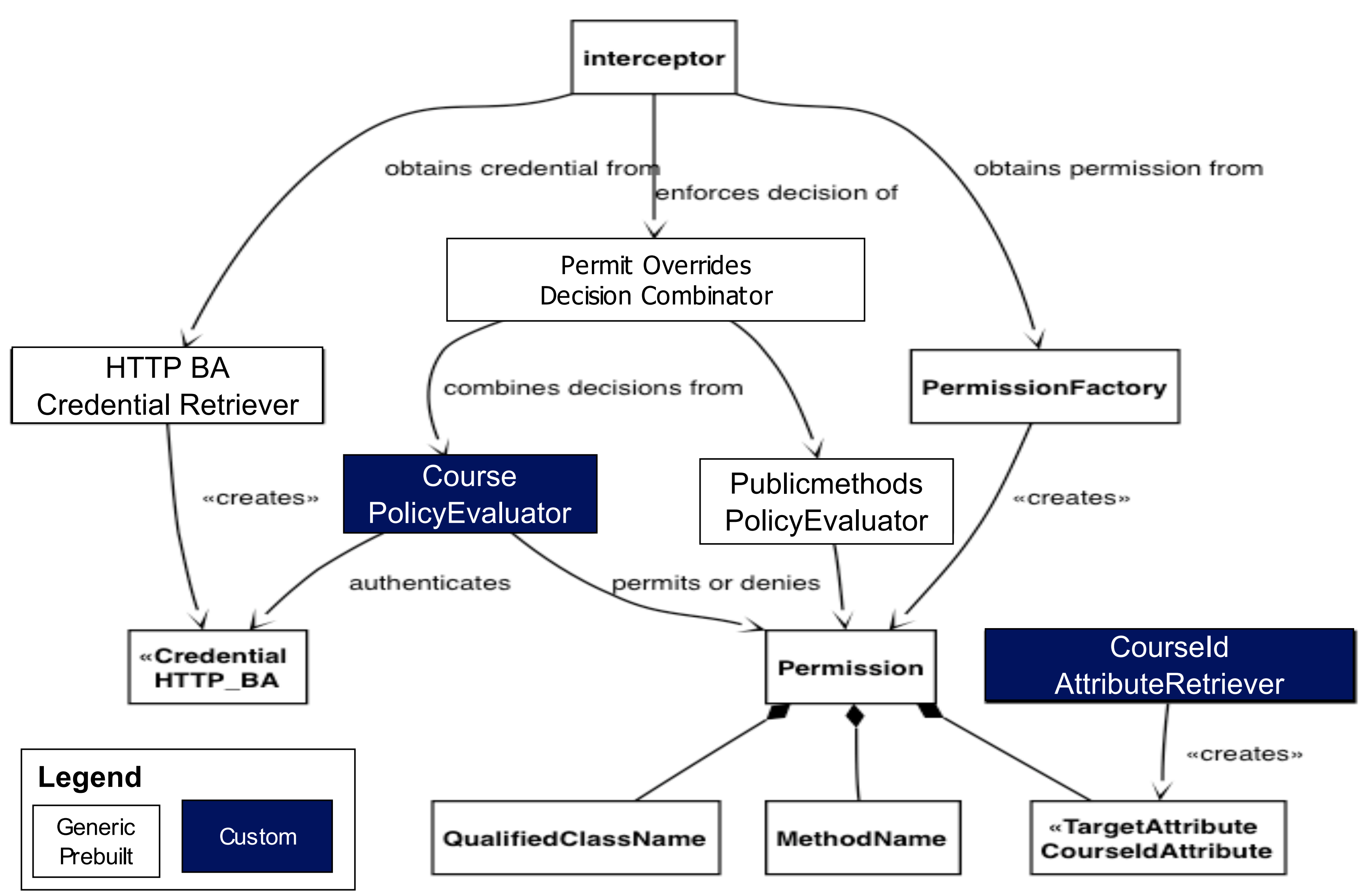
Example 1

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



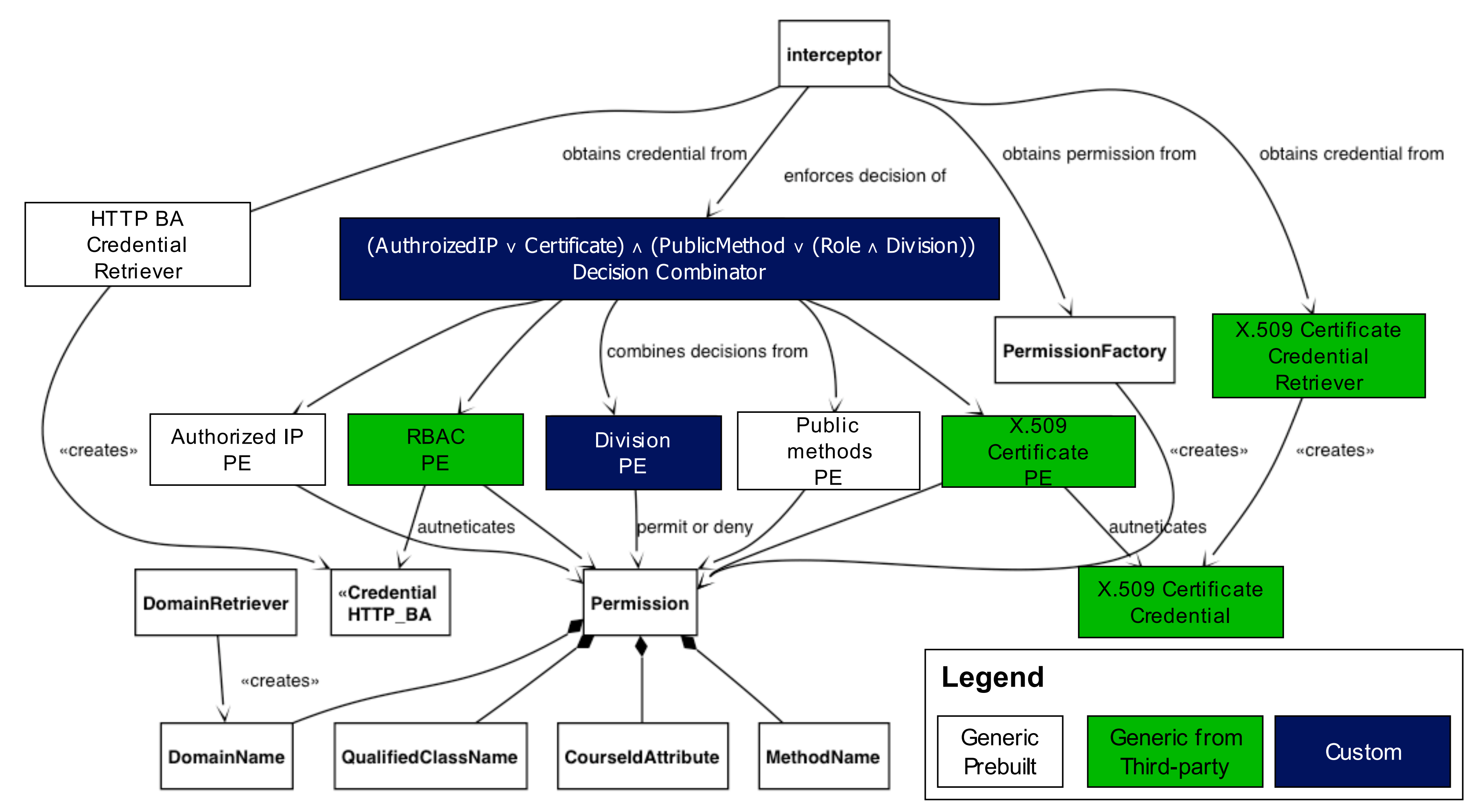
Example 2

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



Expected Lessons Learned

- It’s possible to design security decision logic as components
  - reusable from policy to policy
  - composable to support different policies
  - replaceable to allow new policies
- ASP.NET container is suitable for extensions (in the form of components)
- effective design required deep understanding of access control, Web services, and (ASP).NET
- effective configuration (packaging) crucial
- embracing (not ignoring or suppressing) ASP.NET idiosyncrasies lead to the success

Expected Lessons Learned

- It’s possible to design security decision logic as components
  - reusable from policy to policy
  - composable to support different policies
  - replaceable to allow new policies
- ASP.NET container is suitable for extensions (in the form of components)
- effective design required deep understanding of access control, Web services, and (ASP).NET
- effective configuration (packaging) crucial
- embracing (not ignoring or suppressing) ASP.NET idiosyncrasies lead to the success

Unexpected Lessons Learned

- customers did not care that much about standard compliance & interoperability
- hard to interpret very flexible WS-Security spec
- switching to XP-like User Stories too shocking
- avoid showing all the capabilities/flexibility
- unscalable life-cycle of interceptors
- SOAP interceptor intercepts only SOAP messages (duh!)

Summary

- experience report about designing and implementing protection framework for ASP.NET Web services
- (un)expected lessons learned
  - CB authentication and authorization mechanisms
    - feasable
    - evolve with policies
- details
  - in the paper
  - http://konstantin.beznosov.net