eXtensible Access Control Language (XACML)

Fatih Turkmen

fturkmen(at)disi.unitn.it

fturkmen(at)mit.edu

Visiting PhD Student, CSAIL, MIT

DISI, University of Trento

Massachusetts Institute of Technology

UNIVERSITÀ DEGLI STUDI DI TRENTO

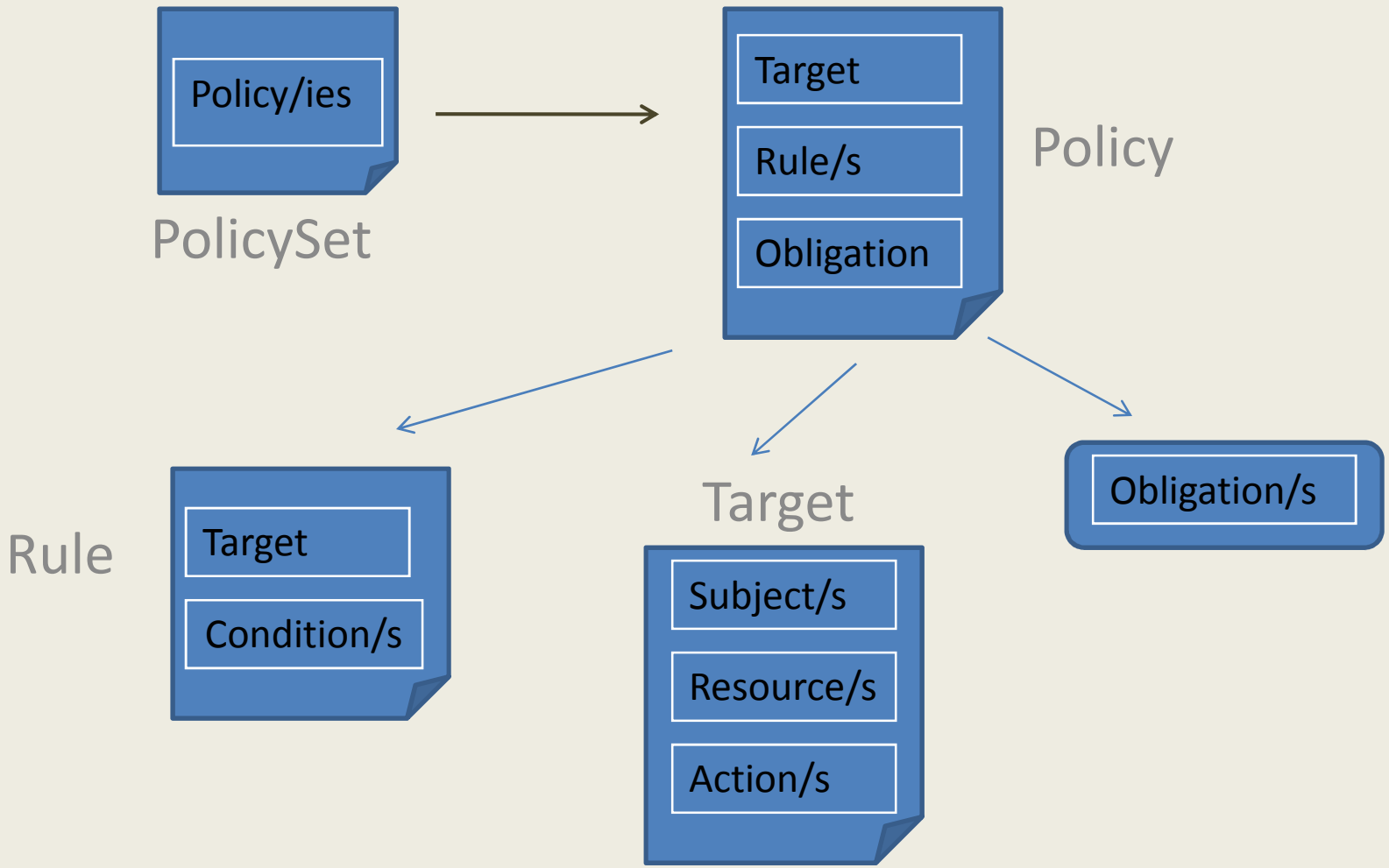
Outline

- eXtensible Access Control Markup Language (XACML)
  - Syntax
  - Architecture
- Industry Practices
- Weaknesses
- Available Implementations
- XACML & AIR
- References

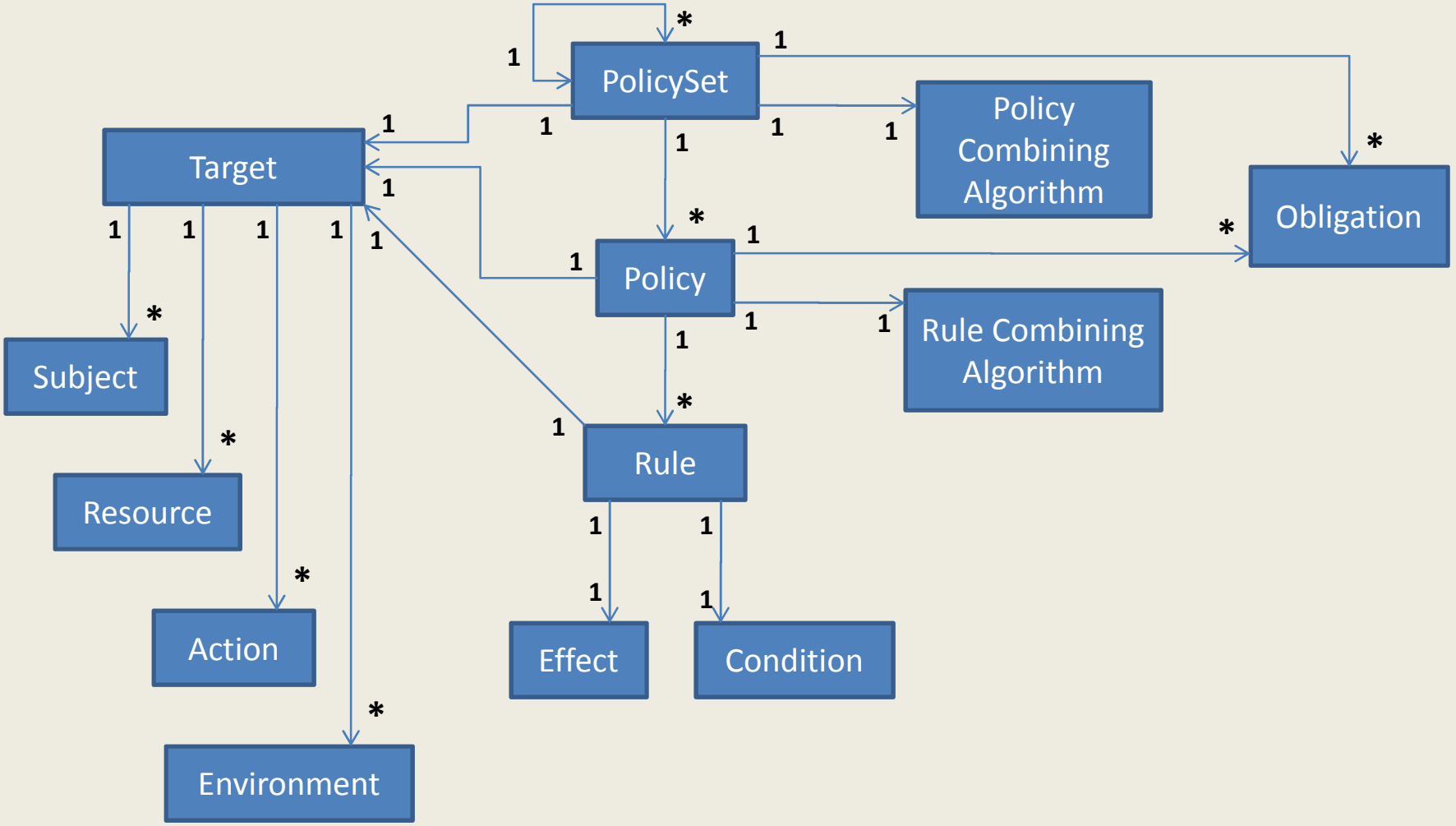
XACML

- XML based access control language
- Simple Syntax, Strong Expressivity, Machine Processable
- OASIS standard
- Widely adopted both in industry and academia
- Many implementations (both open source and proprietary)

XACML (Cont.)


Policy/ies
PolicySet
Target
Rule/s
Obligation
Policy
Rule
Target
Condition/s
Target
Subject/s
Resource/s
Action/s
Obligation/s

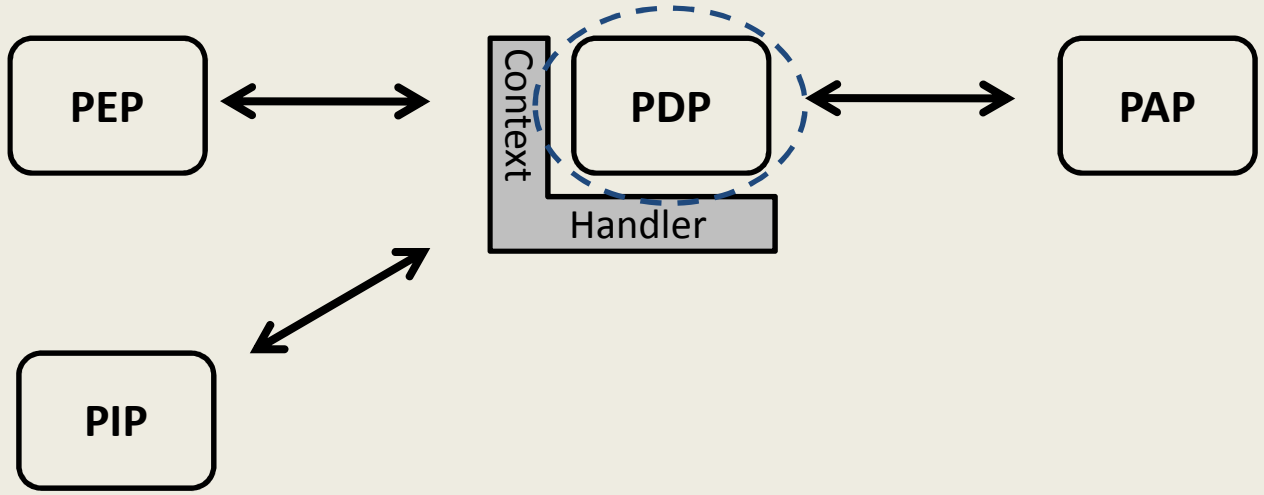
XACML (Cont.)



Redrawn from "Anne Anderson, Sun Microsystems Laboratories, XML Community of Practice, 21 June 2006"

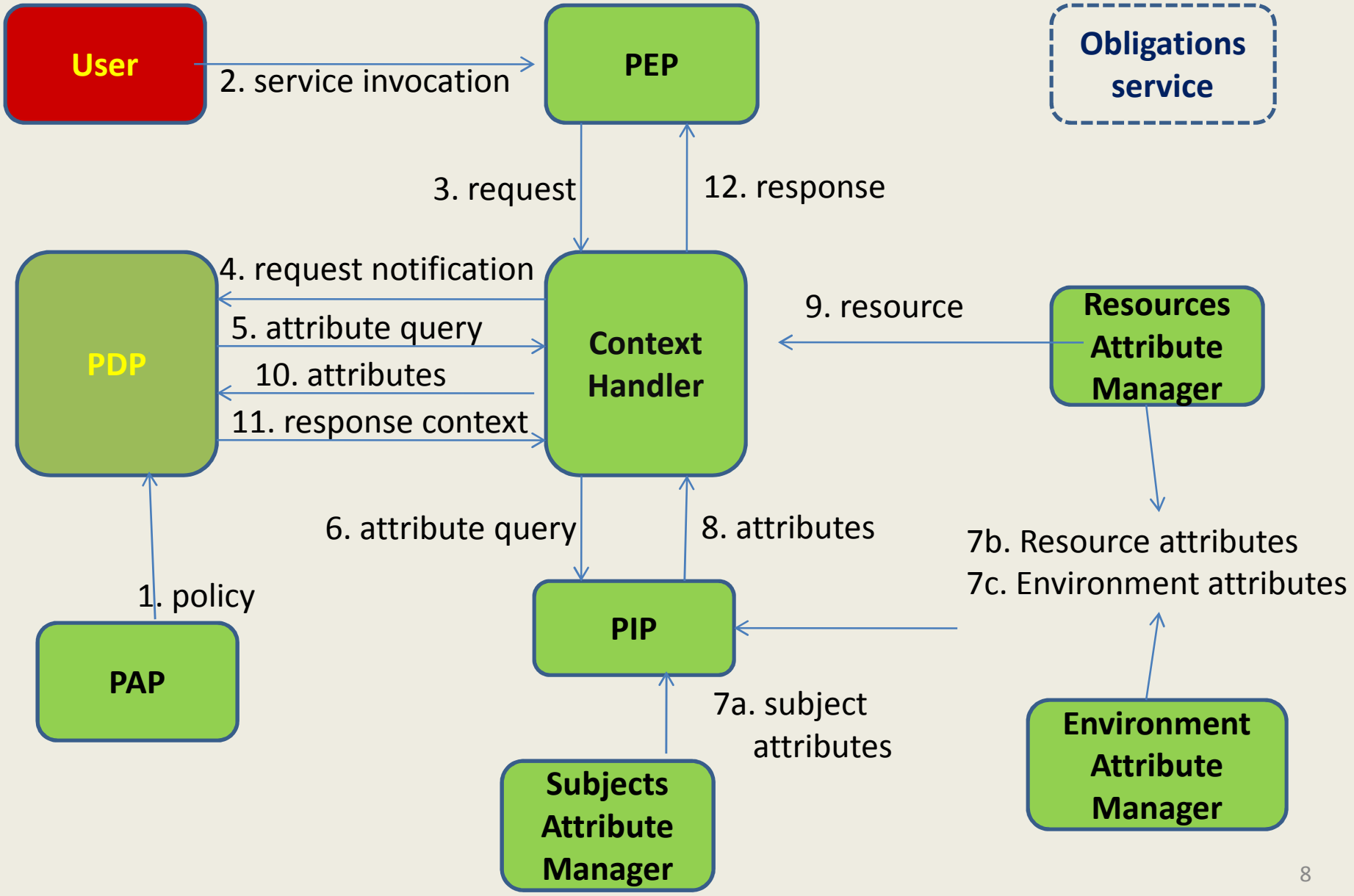
```
<Policy PolicyId="Policy0" RuleCombiningAlgId="Permit-Overrides">
<Description>Sales Report Policy</Description>

<Target/>

<Rule RuleId="Report_Access" Effect="Permit">
    <Target>
            <Subjects>
                    <Subject>        Manager        </Subject>
            </Subjects>

            <Resources>
                    <Resource>       Sales Report   </Resource>
            </Resources>

            <Actions>
                    <Action>         Modify         </Action>
            </Actions>
    </Target>
    <Condition>
            <SubjectAttributeDesignator AttributeId="Division“ /> Sales Department
   </Condition>
</Rule>

<Rule RuleId="FinalRule" Effect="Deny"/>

</Policy>
```

XACML Policy Enforcement



Policy Enforcement Point (PEP): Responsible for making access control decision requests to PDP and the enforcement of the given decisions.
Policy Decision Point (PDP): Makes access decisions by evaluating the given request against matched policies.
Context Handler: Responsible for conversions between XACML canonical format and native formats
Policy Information Point (PIP): Source of content values for XACML attributes.
Policy Administration Point (PAP): Creates and manages the policy and policy sets.

XACML Runtime



Industry Practices

- Adaptation to Business Requirements via Profiles (e.g. RBAC Profile, Web Services Profile, Privacy Profile)

- SAML supported authorization services

- Together with an Identity Management System (e.g. LDAP, OpenID)

Industry Practices (Cont.)

- Integration of XACML to products

Industry Practices (Cont.)

- FedoraCommons
  - General purpose repository system

- Health-care Systems
  - National Swedish Health Care (Axiomatics startup)

- Geospatial XACML
  - Protecting access to distributed geographic information

GeoXACML
protecting geodata

XACML Weaknesses

- Verbose syntax (XML)
  - Can easily get complex.
  - Scalability Issues
- Very basic structure
  - Needs profiles and schemas
- Some Issues listed for v3.0:
  - More general conclusions (yes, no ...)
  - Really generic architecture (so much to be done)

XACML Weaknesses (Cont.)

- Delegation problem
  - Administrative delegation is available
- Informal Syntax, difficult to analyze
  - Verification of Properties (e.g. SoD, Permissions)
  - Compatability among different policies
- Enforcement is difficult

Available Implementations

- XEngine
- Permis
- XACML Enterprise
- XACMLLight
- Sun XACML

  (Many implementations based on it: GlobusXACML, Axiomatics)
- HerASAF
- Some out-of-date or proprietary ones

(XACML.Net, Parthenon, AXESCON XACML 2.0 Engine)

AIR vs XACML

- AIR is logic-based, XACML is not (informal)

→ AIR (Phyton) can be serialized to XML??

- XACML is dedicated for access control, AIR seems more generic

AIR vs XACML (Cont.)

- Obligations can not be addressed in AIR?
  - The requirements to be met after the decision.

- No architectural (enforcement) model provided with AIR

AIR vs XACML (Cont.)

	XACML	AIR
Constructs	PolicySet, Policy, {*Subject, Resource, Action, Environment*}, Rule, Condition, Obligation	Policy, Pattern (Variable), Assertion, Rule, MatchedGraph
Usage	Suitable for offline/online control	More suitable for offline control (analysis)
Evaluation Mechanism	Request against Policy (Request - Response)	Forward Chaining (based on Policy and the generated data) Reasoning
Complexity	Low (both advantage and disadvantage)	High (both advantage and disadvantage)
Extensibility	Yes	?????
Conflict Resolution	Flexible Combining Algorithms	Left to the reasoner

References

- http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
- http://www.oasis-open.org/committees/download.php/27298/xacmlRefs-V1-84-1.htm
- http://www.cse.msu.edu/~feichen/xengine.html
- http://sec.cs.kent.ac.uk/permis/downloads/download.shtml
- http://code.google.com/p/enterprise-java-xacml/downloads/list
- http://sourceforge.net/project/showfiles.php?group_id=224175
- http://sourceforge.net/cvs/?group_id=73884
- http://www.herasaf.org/development.html
- http://www.fedora-commons.org/about/

Shoot with Questions !!!