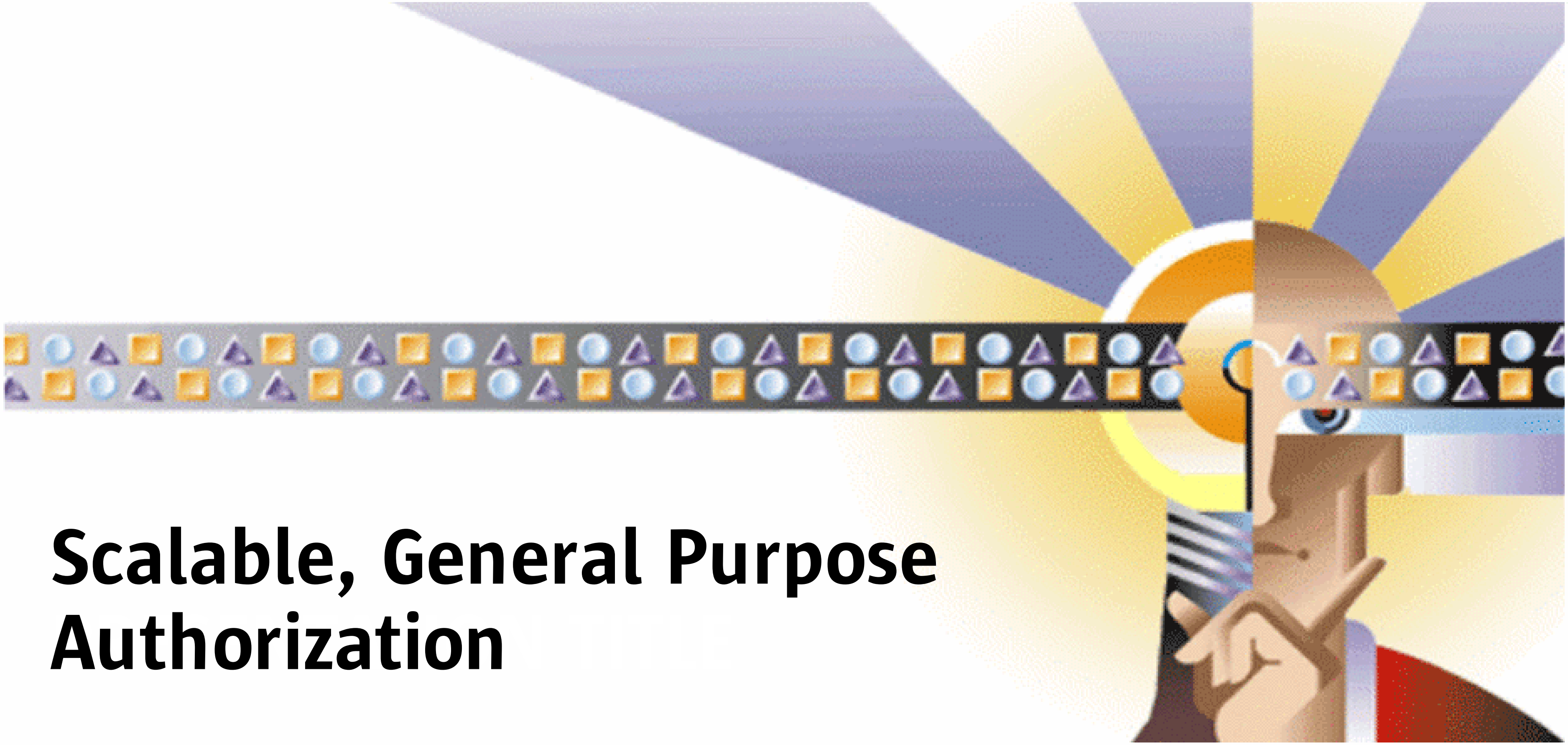


Scalable, General Purpose Authorization

Seth Proctor

Member of Technical Staff

Sun Microsystems Laboratories

Sun microsystems

We make the net work.

Outline

- Introduction & Current Challenges
- XACML Overview
- XACML in Detail: Policies & Queries
- Sun's XACML Implementation
- XACML 2.0 & Related Standards
- Future Research & Implementation Directions

Some Security Basics

- Authentication
  - Determining who you are, semantics aside
- Authorization
  - Determining what you can do, and under which conditions
  - Authorization criteria is generally based on Authentication
- Confidentiality, Integrity, Privacy...

Authorization

- Authorization might be based on Identity, Groups, Attributes, Roles, etc.
- N-tuple requests and (generally) boolean responses
- Policies drive authorization decisions
  - Example: -rwxr-x--- stp other a.out

Current Challenges

- Many Application and Environment specific languages and data sources
  - If there are no tools for a custom language...
- No standard, general languages
  - Some non-standard languages do exist
  - No standard solutions for developers
- Key Problem: No good support for distributed & decentralized policy

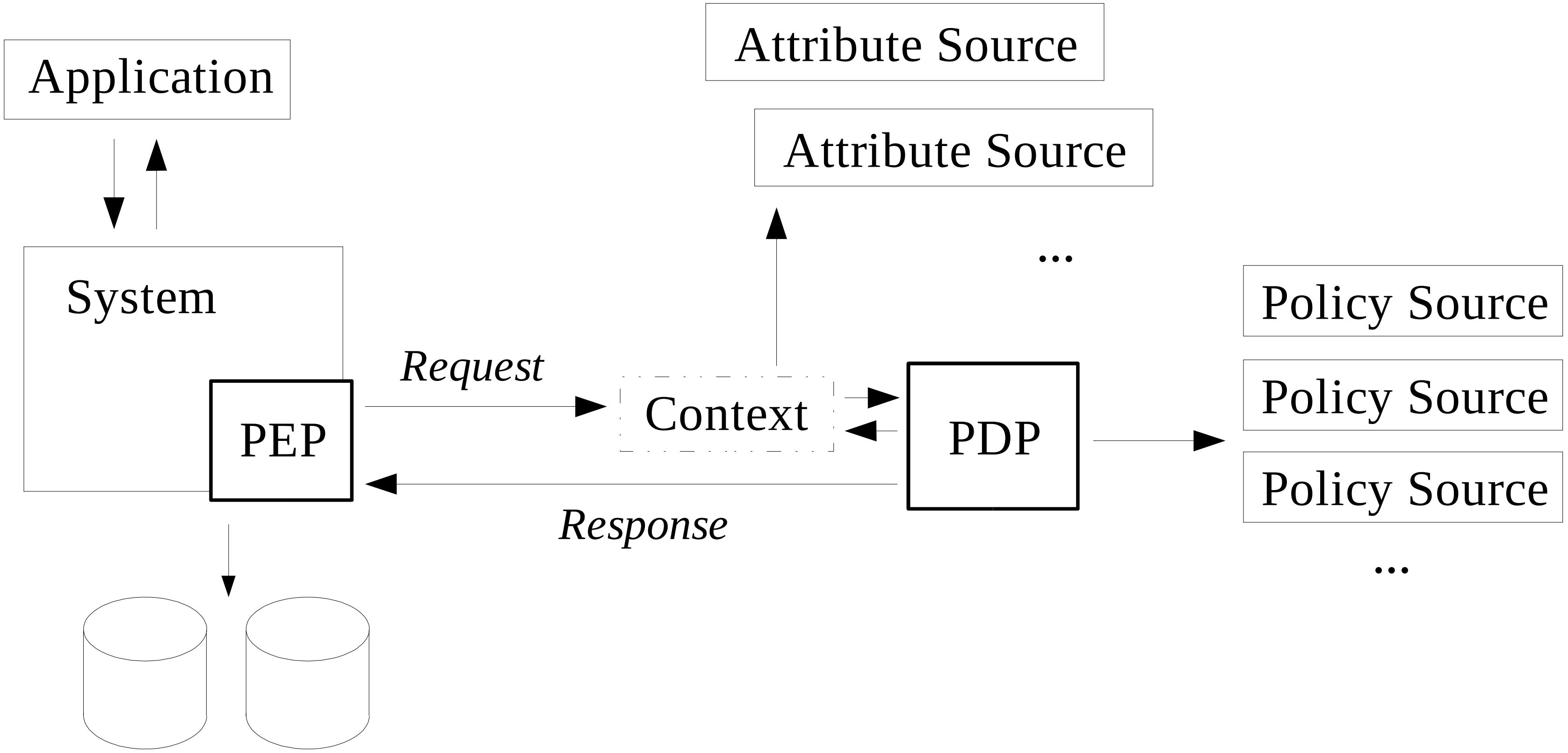
XACML 1.1

- The eXtensible Access Control Markup Language is an OASIS Open Standard
- Defined in XML, though not tied to Web Services exclusively
- General Access Control Policy language
- General enough to be an intermediate language too

What XACML Provides

- Policy and Query language
- Semantics for processing policies and determining applicability to requests
- Standard data types & functions
- Extensible & Flexible
- Remote referencing and inclusion

Conceptual Model



XACML Policy Structure

- An XACML policy is a tree of
  - PolicySet: contains multiple policies and policy references
  - Policy: contains multiple Rules and represents a single ACP
  - Rule: contains decision logic in Conditions and an Effect (Permit or Deny)
- Each node has a Target for determining applicability

Conditions

- Single boolean combination of predicates
  - Predicates can be nested to any depth using any supported functions and datatypes
- If `true`, the Effect (Permit or Deny) is returned
- If `false`, the Rule is NotApplicable

Attributes

- All values in XACML are Attributes
- All values are of a known type
  - Standard types include boolean, string, integer, date, email, etc.
  - New datatypes can be defined as needed
- Sets (unique collections) and Bags (non-unique collections) are supported

Functions (and the Apply tag)

- All functions have a known identifier
  - Standard functions include equal, greater-than, match, add, is-in, etc.
  - New functions can be defined as needed
- Functions have well defined parameter and return types (as Attribute types)
- Higher-order functions are supported
- Used in Targets and Conditions

Combining Algorithms

- If multiple Policies or Rules are used, you have multiple decisions
- Combining Algorithms resolve single decisions out of multiple decisions
- Different ones for Rules and Policies
  - Standard algorithms include first-applicable, deny-overrides, etc.
  - New algorithms can be defined as needed

Requests and Responses

- Request is a 4-tuple of attributes
  - Subjects, Resource, Action, and Environment
  - Request Attributes include an identifier
  - Subjects are labeled with category identifiers
  - The resource content may be included
- Result is a 4-valued “boolean”
  - Permit, Deny, Indeterminate, NotApplicable
  - Optional: Status, Messages, and Obligations

Designators & Selectors

- Used to reference values in Request
- Included in Conditions and Targets
  - Designators use Identifiers and specify subject, resource, action, or environment
  - Selectors use XPath queries
  - Both specify a datatype
  - Both may require that a value must be present
- May look outside the Request

SunXACML Features

- Open Source & written in the Java™ Programming Language
- Full support for XACML 1.0/1.1
- Classes for generating, parsing, and evaluating policies
- Classes for generating, parsing, and using Requests and Responses

SunXACML Features (2)

- Supports all the standard functions and datatypes
- Supports most optional features (like Obligations and AttributeSelectors)
- Provides a PDP interface for easy evaluation
- Well documented with many examples

SunXACML Features (3)

- Pluggable support for new Functions, Data Types, and Combining Algorithms
- Finder Modules for retrieving policies and attribute values
  - Makes for easy integration with current and future systems
- Extension points only as the Standard allows

XACML 2.0

- Syntax cleanups & syntactic sugar
- Updated descriptions & examples
- Hierarchical resources
- Time handling
- Versions for policies & references
- Delegated administration (maybe)

Related Standards/Groups

- WSPL
  - Communications criteria
- SAML 2.0
  - Assertions and online exchanges
- ebXML Registry 2.x
  - Securing registry data
- Global Grid Forum & Internet2

Future Directions for XACML

- XACML 2.0 draft (spring 2004-ish)
- Profiles
  - RBAC
  - Retrieval mechanisms & configuration
  - LDAP and other attribute/policy sources
  - Delegated Administration
  - Mappings for filesystems and other hierarchical resources

Future Implementation Work

- Support XACML 2.0
- Support new profiles (as they emerge)
- Tools, tools, and tools
- Work with other systems (J2SE, Apache, etc.)
- Concrete tasks are on the project web page...get coding!

Current & Future Research

- Protocols
  - PEP/PDP exchange and PDP/P*P exchanges
- Performance
  - How do we build an efficient system using XACML?
  - How do we think about caching and contention models for policies and attributes?
- Trusted Computing

Current & Future Research (2)

- Privacy & Trust negotiation
  - How do we establish trust for a given exchange or relationship?
  - What do we use to determine the value of our privacy?
- Delegation
  - Policies for what can be delegated
  - Policies to protect what has been delegated

Current & Future Research (3)

- Visualization and Usability
  - How do we visualize policy data as it gets large and decentralized?
  - How do average users interact with their policies?
- Reasoning
  - Understanding the meaning of a policy
  - Understanding the effect of a given change

Conclusion

- XACML is an open standard for generic, decentralized Access Control Policy
- An Open Source implementation makes it easy to get started
- Ongoing work to connect XACML with other authorization components
- Interesting research still left to do

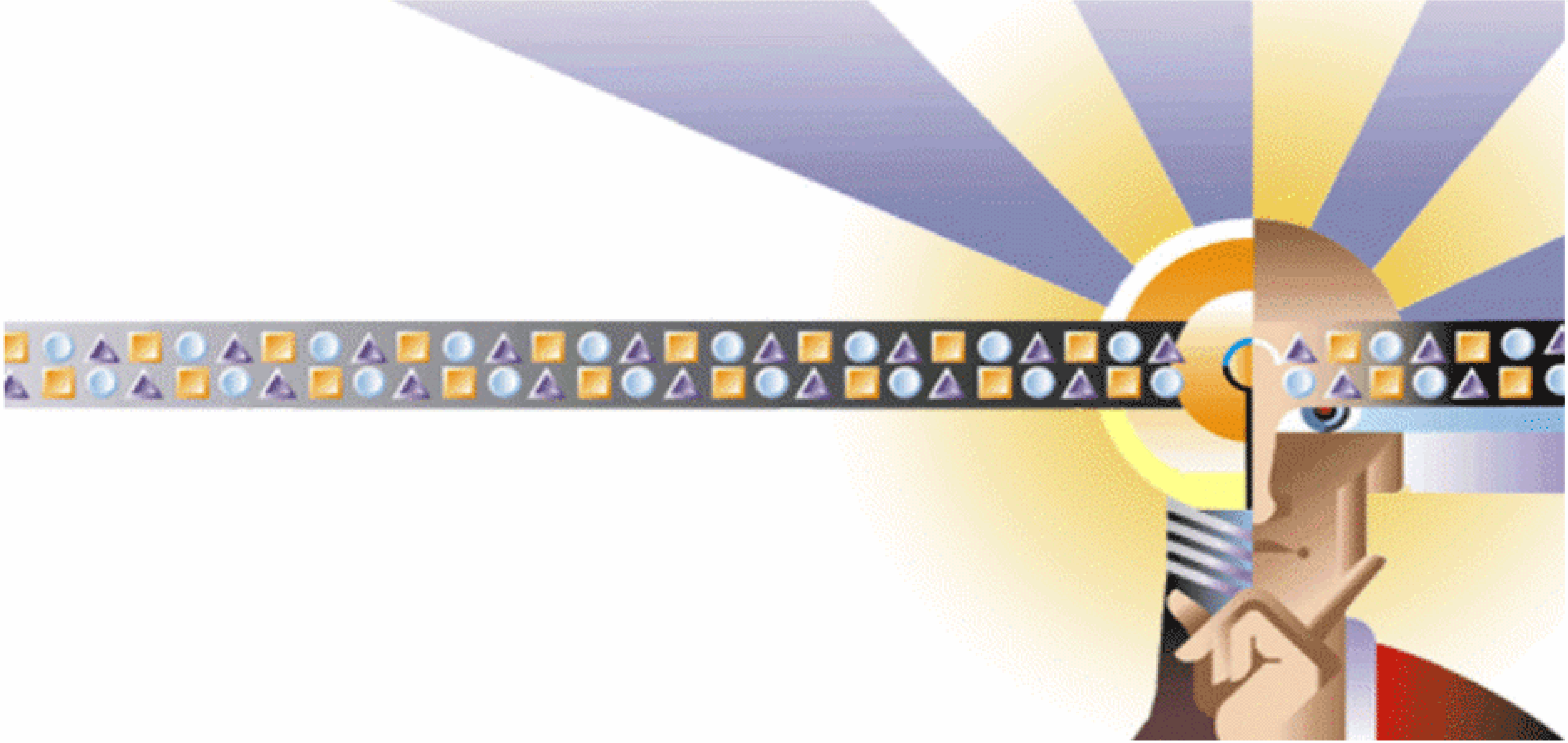
References

- The XACML TC

  http://www.oasis-open.org/committees/xacml
- Sun's XACML Implementation

  http://sunxacml.sourceforge.net
- SunLabs

  http://research.sun.com



Seth Proctor

seth.proctor@sun.com

Sun microsystems
We make the net work.