NAC 2007 Spring Conference

OASIS XACML Update

Hal Lockhart
Office of the CTO
BEA Systems
hlockhar@bea.com

Hal Lockhart

- Senior Principal Technologist, OCTO
- Co-chair XACML TC, SAML TC
- Co-chair OASIS TAB
- Vice Chair WS-I Basic Security Profile
- Also Member:
  - Provisioning TC, Digital Signature Services TC, Web Services Secure Exchange (WS-SX), WS-I Reliable Secure Profile WG
- OASIS Coordinator for WSS Interop Demo, OASIS XACML Interop Demo

Topics

- Overview of Policy and Authorization
- XACML Overview
- XACML Concepts
- Policy Evaluation
- XACML Profiles
- XACML 3.0
- XACML Interop Demo

Information Security Definition

Technologies and procedures intended to implement organizational policy in spite of human efforts to the contrary.

- Suggested by Authorization
- Applies to all security services
- Protection against accidents is incidental
- Suggests four areas of attention

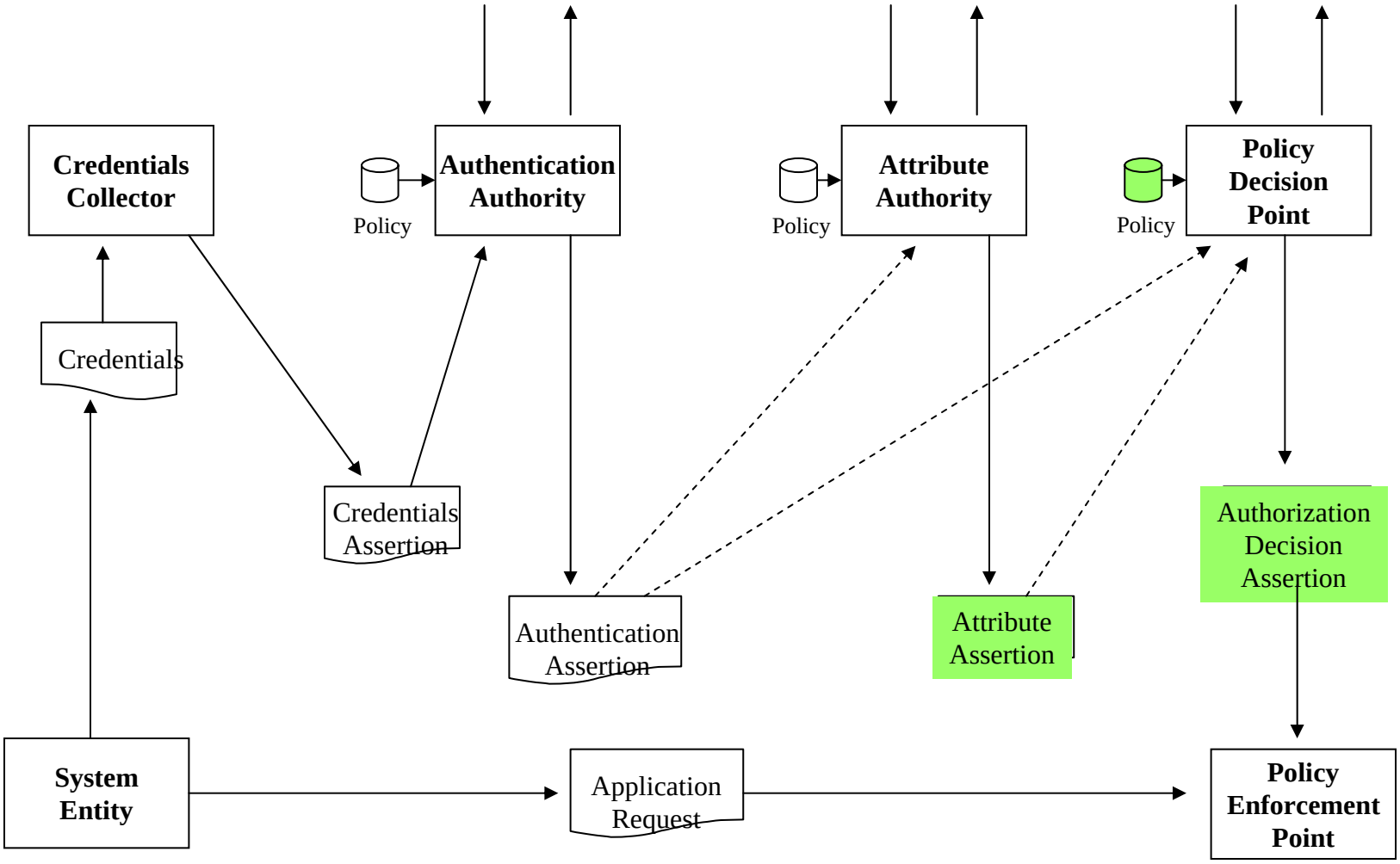
Information Security Areas

- Policy determination
  - Expression: code, permissions, ACLs, Language
  - Evaluation: semantics, architecture, performance
- Policy enforcement
  - Maintain integrity of Trusted Computing Base (TCB)
  - Enforce variable policy

Infrastructural Service

- Consistent enforcement of security policies
- Minimize user inconvenience
- Ensure seamless implementation
  - Coherent, interdependent services
  - Not just list of products
- Avoid reimplementation
- Simplify management and administration

Authorization Theory



Types of Authorization Info – 1

- Attribute Assertion
  - Properties of a system entity (typically a person)
  - Relatively abstract – business context
  - Same attribute used in multiple resource decisions
  - Examples: X.509 Attribute Certificate, SAML Attribute Statement, XrML PossessProperty
- Authorization Policy
  - Specifies all the conditions required for access
  - Specifies the detailed resources and actions (rights)
  - Can apply to multiple subjects, resources, times…
  - Examples: XACML Policy, XrML License, X.509 Policy Certificate

Types of Authorization Info – 2

- AuthZ Decision
  - Expresses the result of a policy decision
  - Specifies a particular access that is allowed
  - Intended for immediate use
  - Example: SAML AuthZ Decision Statement, IETF COPS

Implications of this Model

- Benefits
  - Improved scalability
  - Separation of concerns
  - Enables federation
- Distinctions not absolute
  - Attributes can seem like rights
  - A policy may apply to one principal, resource
  - Systems with a single construct tend to evolve to treating principal or resource as abstraction

OASIS XACML History

- First Meeting – 21 May 2001
- Requirements from: Healthcare, DRM, Registry, Financial, Online Web, XML Docs, Fed Gov, Workflow, Java, Policy Analysis, WebDAV
- XACML 1.0 - OASIS Standard – 6 February 2003
- XACML 1.1 – Committee Specification – 7 August 2003
- XACML 2.0 – OASIS Standard – 1 February 2005
- XACML 2.0 – ITU/T Recommendation X.1142

XACML TC Charter

- Define a core XML schema for representing authorization and entitlement policies
- Target - any object - referenced using XML
- Fine grained control, characteristics - access requestor, protocol, classes of activities, and content introspection
- Consistent with and building upon SAML

Policy Examples

- “Anyone can use web servers with the ‘spare’ property between 12:00 AM and 4:00 AM”
- “Salespeople can create orders, but if the total cost is greater that $1M, a supervisor must approve”
- “Anyone view their own 401K information, but nobody else’s”
- “The print formatting service can access printers and temporary storage on behalf of any user with the print attribute”
- “The primary physician can have any of her patients’ medical records sent to a specialist in the same practice.”

XACML Objectives

- Ability to locate policies in distributed environment
- Ability to federate administration of policies about the same resource
- Base decisions on wide range of inputs
  - Multiple subjects, resource properties
- Decision expressions of unlimited complexity
- Ability to do policy-based delegation
- Usable in many different environments
  - Types of Resources, Subjects, Actions
  - Policy location and combination

General Characteristics

- Expect it to be generated by programs
- Defined using XML Schema
- Strongly typed language
- Extensible in multiple dimensions
- Borrows from many other specifications
- Features requiring XPath are optional
- Obligation feature optional
- Language is very “wordy”
  - Many long URLs
- Complex enough that there is more than one way to do most things

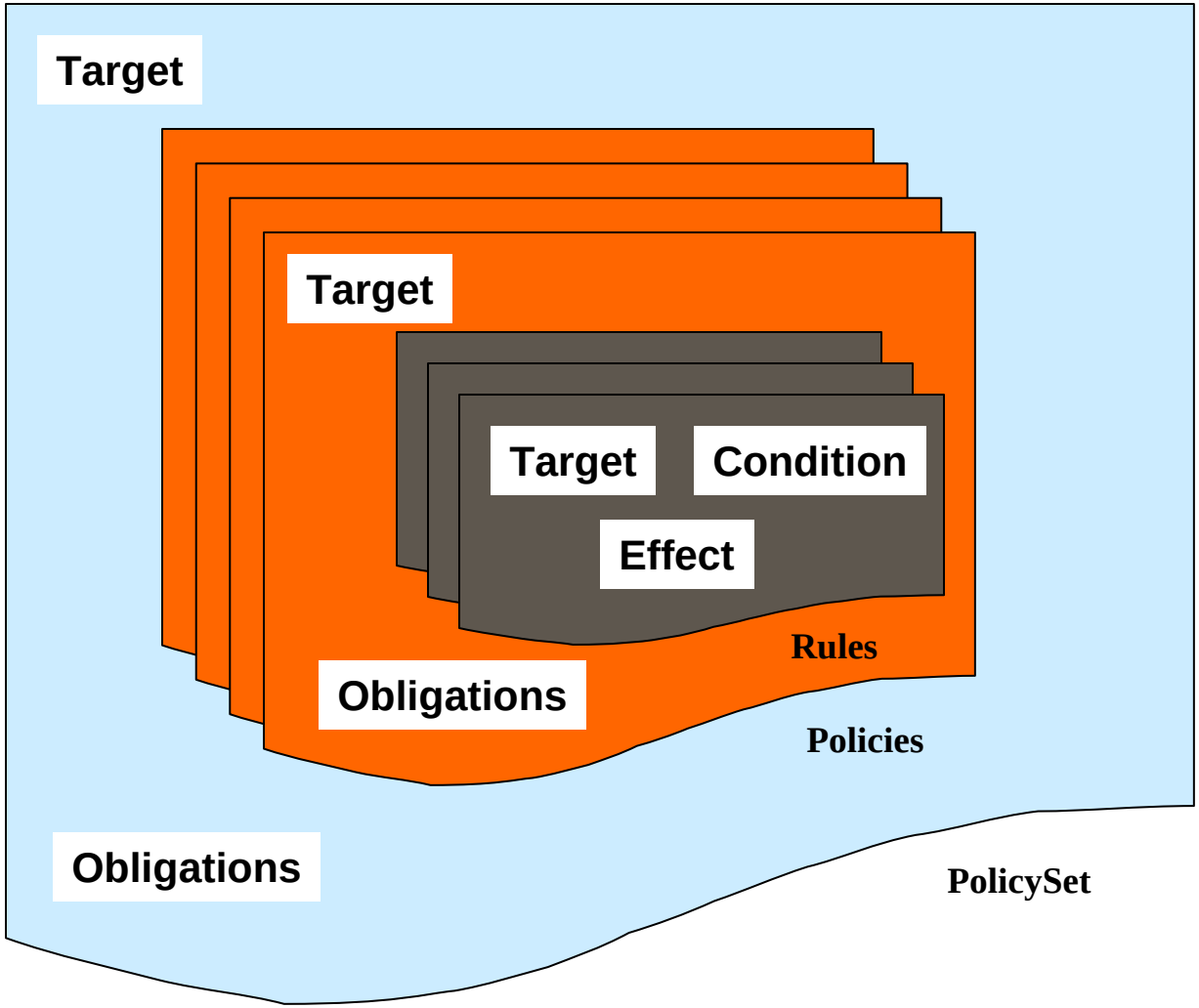
Novel XACML Features

- Large Scale Environment
  - Subjects, Resources, Attributes, etc. not necessarily exist or be known at Policy Creation time
  - Multiple Administrators - potentially conflicting policy results
  - Combining algorithms
- Request centric
  - Use any information available at access request time
  - Zero, one or more Subjects
  - No invented concepts (privilege, role, etc.)
- Dynamically bound to request
  - Not limited to Resource binding
  - Only tell what policies apply in context of Request
  - Two stage evaluation

XACML Concepts

- Request and Response Contexts – Input and Output
- Policy & PolicySet – combining of applicable policies using CombiningAlgorithm
- Target – Rapidly index to find applicable Policies or Rules
- Conditions – Complex boolean expression with many operands, arithmetic & string functions
- Effect – “Permit” or “Deny”
- Obligations – Other required actions
- Bag – unordered list which may contain duplicates

XACML Concepts



Rules

- Smallest unit of administration, cannot be evaluated alone
- Elements
  - Description – documentation
  - Target – select applicable policies
  - Condition – boolean decision function
  - Effect – either “Permit” or “Deny”
- Results
  - If condition is true, return Effect value
  - If not, return NotApplicable
  - If error or missing data return Indeterminate
    - Plus status code

Target

- Designed to efficiently find the policies that apply to a request
- Enables dynamic binding
- Makes it feasible to have very complex Conditions
- Attributes of Subjects, Resources, Actions and Environments
- Matches against value, using match function
  - Regular expression
  - RFC822 (email) name
  - X.500 name
  - User defined
- Attributes specified by Id or XPath expression
- Normally use Subject or Resource, not both

Condition

- Boolean function to decide if Effect applies
- Inputs come from Request Context
- Values can be primitive, complex or bags
- Can be specified by id or XPath expression
- Fourteen primitive types
- Rich array of typed functions defined
- Functions for dealing with bags
- Order of evaluation unspecified
- Allowed to quit when result is known
- Side effects not permitted

Datatypes

- From XML Schema
  - String, boolean
  - Integer, double
  - Time, date
  - dateTime
  - anyURI
  - hexBinary
  - base64Binary
- From Xquery
  - dayTimeDuration
  - yearMonthDuration
- Unique to XACML
  - rfc822Name
  - x500Name

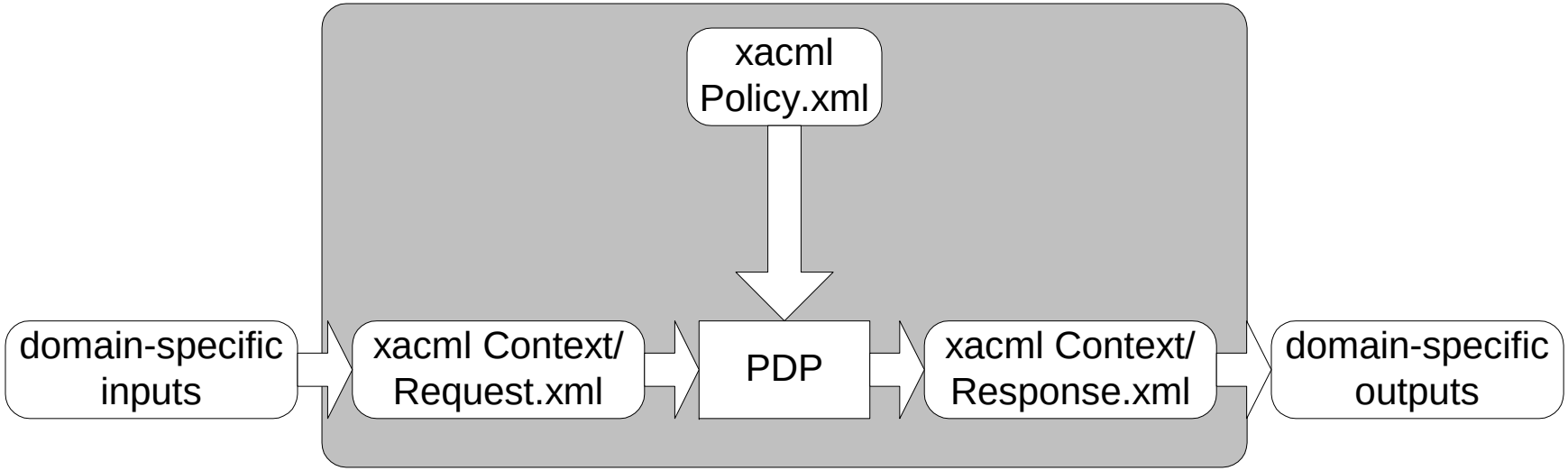
Functions

- Equality predicates
- Arithmetic functions
- String conversion functions
- Numeric type conversion functions
- Logical functions
- Arithmetic comparison functions
- Date and time arithmetic functions
- Non-numeric comparison functions
- Bag functions
- Set functions
- Higher-order bag functions
- Special match functions
- XPath-based functions
- Extension functions and primitive types

Policies and Policy Sets

- Policy
  - Smallest element PDP can evaluate
  - Contains: Description, Defaults, Target, Rules, Obligations, Rule Combining Algorithm
- Policy Set
  - Allows Policies and Policy Sets to be combined
  - Use not required
  - Contains: Description, Defaults, Target, Policies, Policy Sets, Policy References, Policy Set References, Obligations, Policy Combining Algorithm
- Combining Algorithms: Deny-overrides, Permit-overrides, First-applicable, Only-one-applicable

Request and Response Context



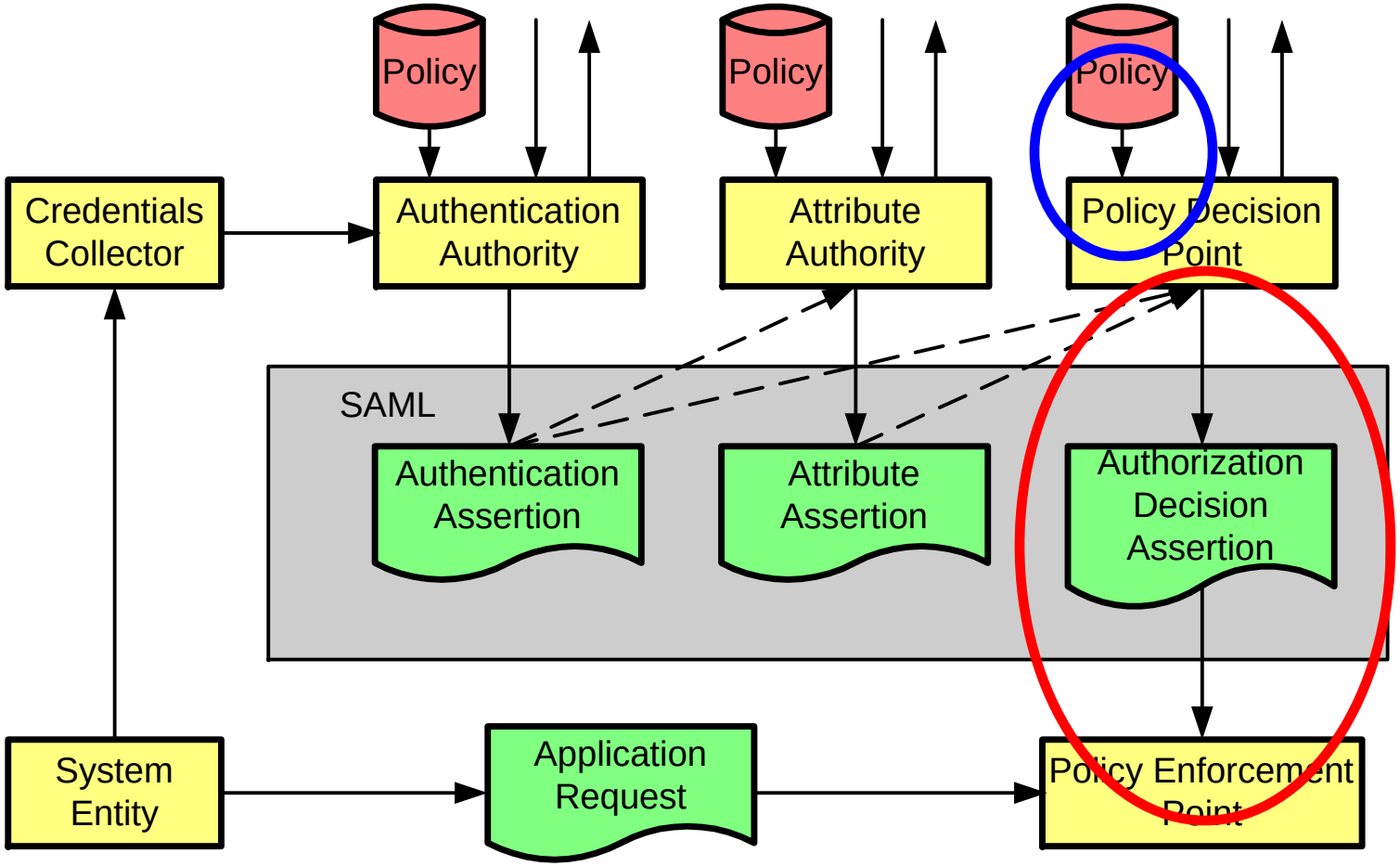
OASIS

XACML 2.0 Profiles

- Digital Signature
  - Integrity protection of Policies
- Hierarchical Resources
  - Using XACML to protect files, directory entries, web pages
- Privacy
  - Determine “purpose” of access
- RBAC
  - Support ANSI RBAC Profile with XACML
- SAML Integration
  - XACML-based decision request
  - Fetch applicable policies
  - Attribute alignment

OASIS

XACML 2.0 Uses SAML Features



XACML Performance

- Some public comments based on ignorance
- Many optimization opportunities
  - Policy encoding
  - Request context
  - Partial evaluation
  - Decision Caching
  - Precomputed admin chaining
- Complex policies cost more to evaluate than simple
  - But is the difference more significant that other factors?

Current Work - XACML 3.0

- Administration/Delegation
- Schema generalization
- WS-XACML
- Obligation combining rules
- Policy provisioning
- Metadata/vocabulary advertisement
- Closely coupled PDP/PEP

Delegation with XACML 2.0

- Use of Intermediary Subject Category
  - Print Format Service can read any file a user wants printed, but not otherwise
  - Access Subject + Intermediary Subject
- Delegation by modifying attributes
  - User can enable family member's access
  - Policy protects subject repository
- Policies protecting each policy repository

Administration/Delegation

- Two primary use cases
  - “HR-Admins can create policies concerning the Payroll servers”
  - “Jack can approve expenses while Mary is on vacation”
- Backward compatible
- Likely to define two compliance levels
- Policies can contain Issuer
- Policies can be Access or Admin
- Admin policies enable policy creation

Administration/Delegation

- Situation – all information values used as policy inputs
- If policy issued by trusted issuer – use
- If not, look for Admin policy for Issuer covering current Situation
- Chain back to Trusted Issuer
- Actual processing is complex, because of interplay with policy combining

Other 3.0 Work

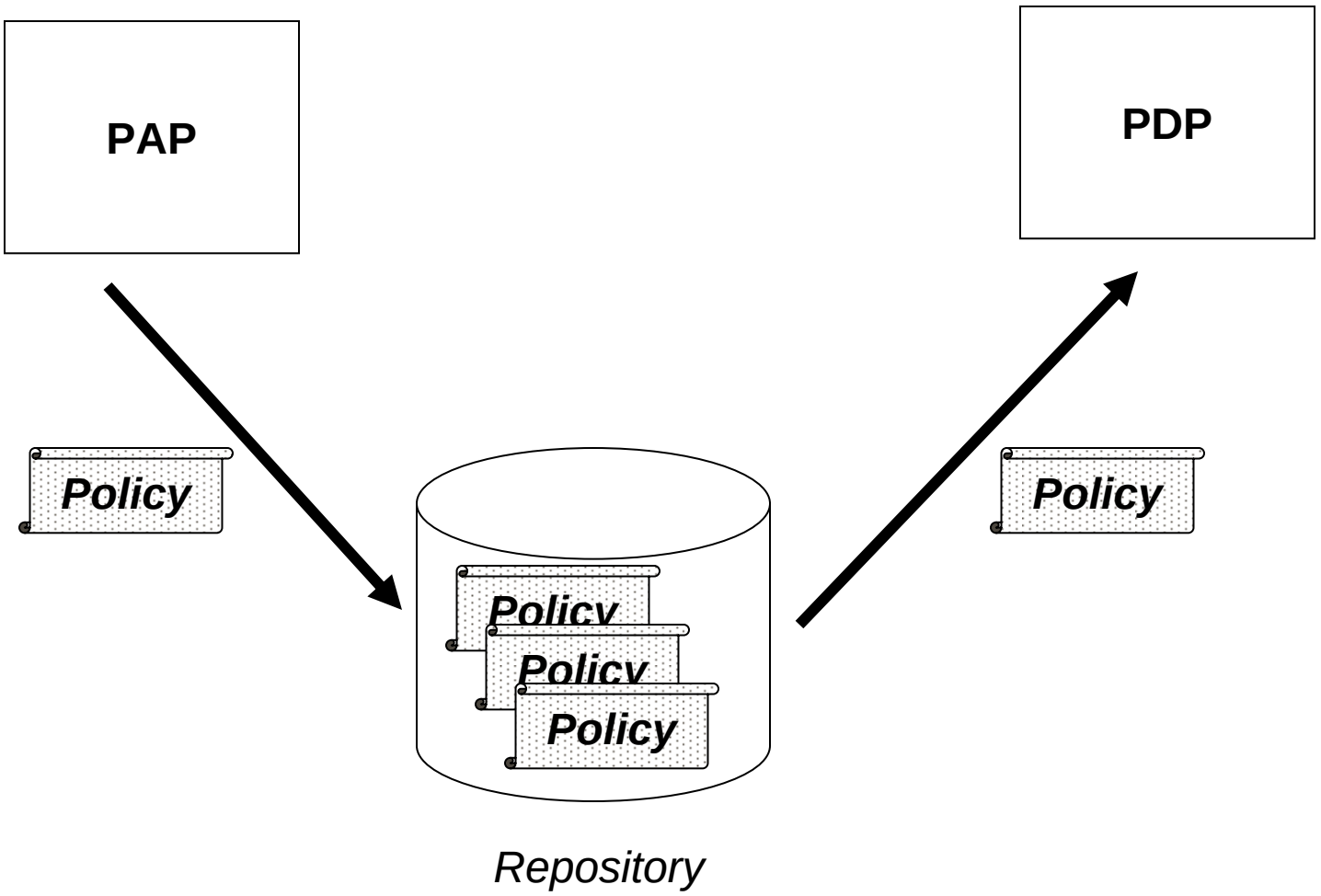
- Schema generalization
  - Improve extensibility
- WS-XACML
  - Builds on WS-Security Policy – more fine grained
  - Good for privacy policies
- Obligation combining rules
  - XACML 2.0 accumulates all Obligations
  - Characterize Obligation types – enable different treatments
- Policy provisioning
  - From repository distribute distinct policy subsets

OASIS

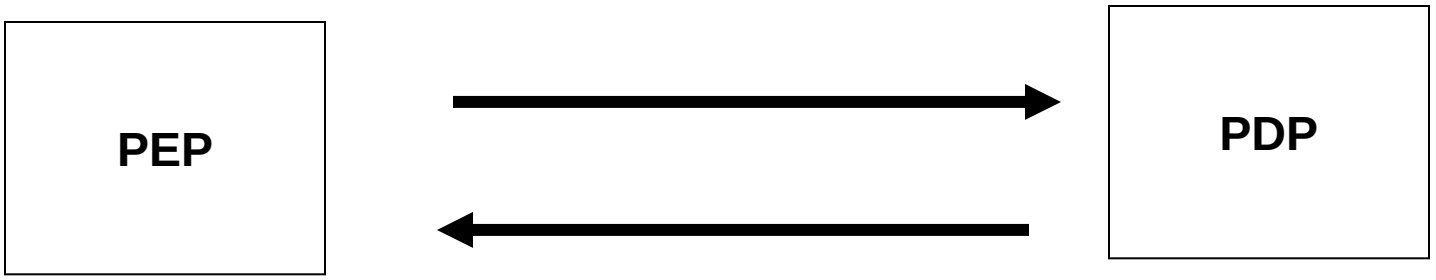
XACML Interop Demo

- Burton Catalyst Conference
  - San Francisco, June 25-29, 2007
- Tentative participants
  - BEA, IBM, Jericho Systems, Oracle, Redhat, Securent, Symlabs
- Approach under discussion
  - Two Usecases (Policy Exchange, Decision)
  - Four Stock Trading Scenarios
- Weekly concalls

Policy Exchange Scenario



Decision Request Scenario



Interop Challenges

- Minimize extraneous components
- Agree on items unspecified by XACML
- Motivating business cases
- Present understandable demo
- Repeatable scenarios
- Human error
- Opportunity for ad hoc variants

Questions?