Web Services Profile of XACML (WS-XACML)

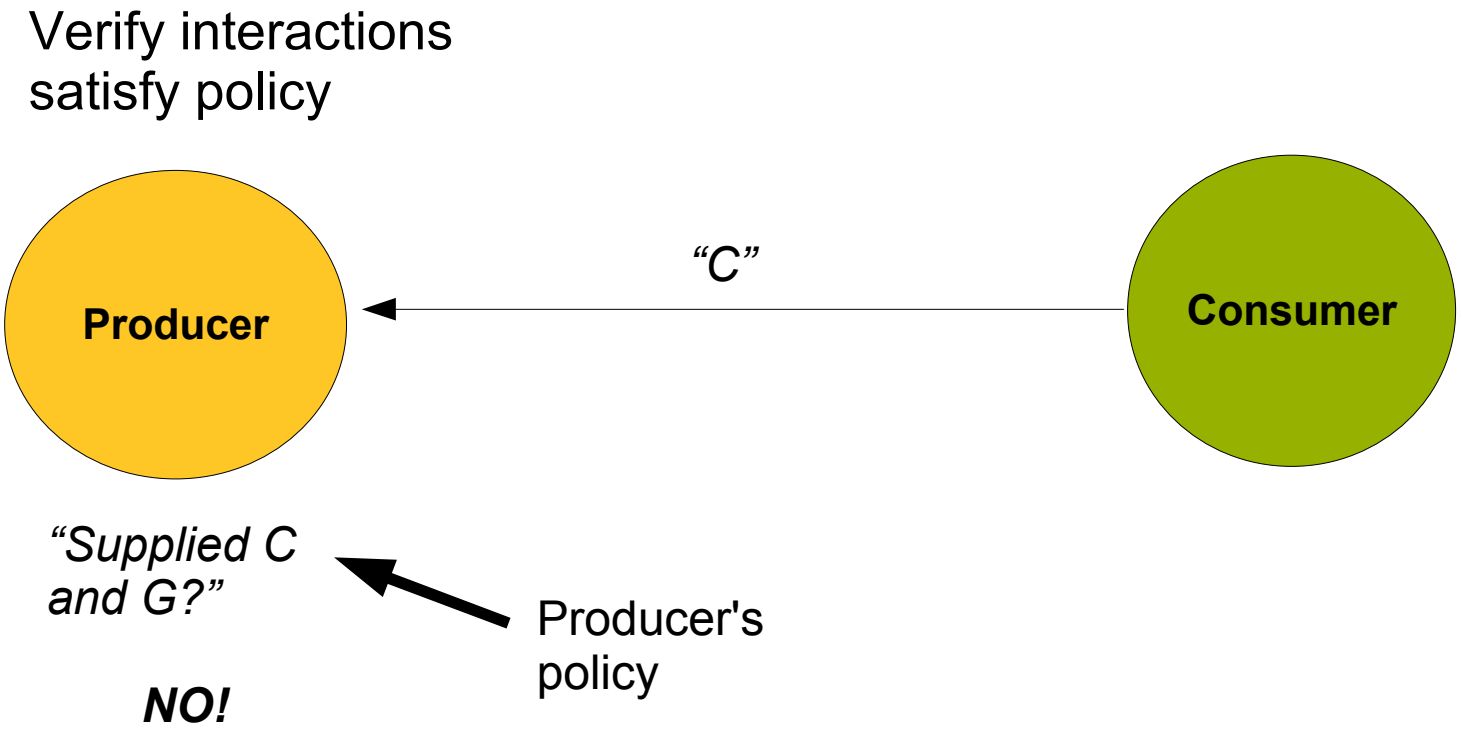
Anne Anderson
Sun Microsystems, Inc.

XACML TC F2F, March 2007

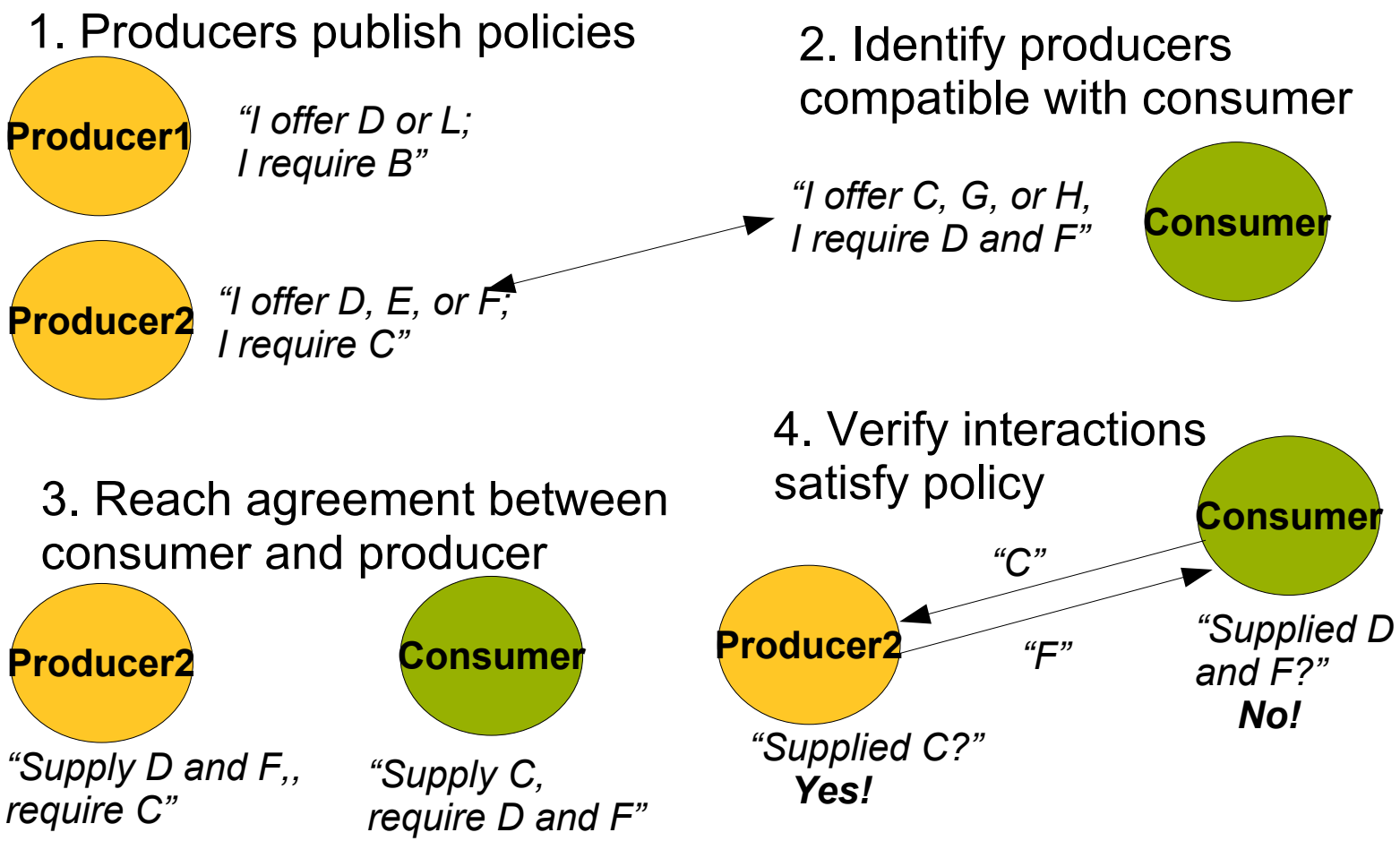
Outline

- **Web Services Policy Background**
- XACML Web Services Policy Assertions
- XACML Assertion Format
- XACML Assertion Matching
- Defined XACML Assertions
  - XACMLAuthzAssertion
  - XACMLPrivacyAssertion
- New XACML Functions and Attribute Identifiers
- Open Issues

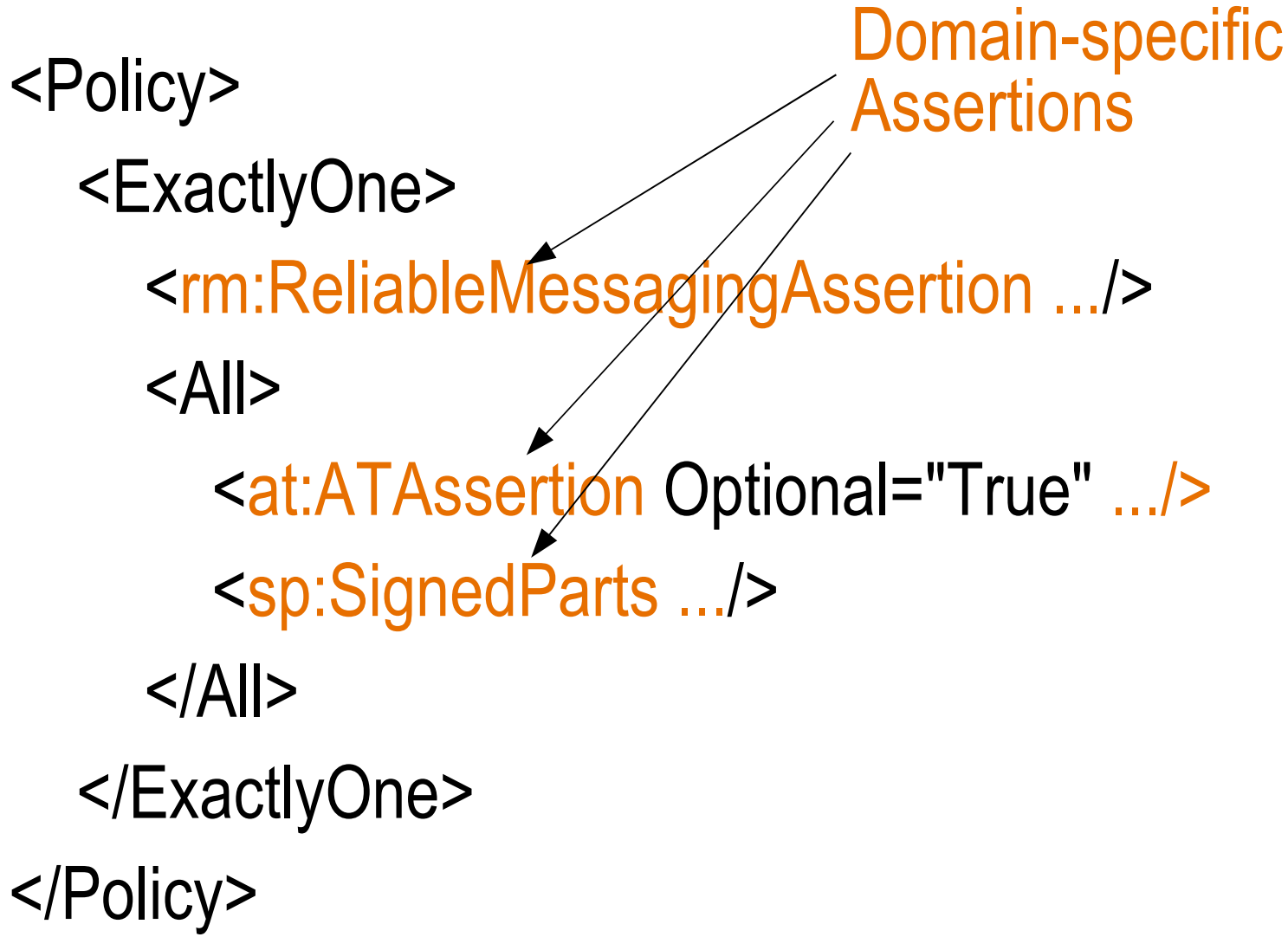
Typical XACML policy usage today



Policy usage in Web Services



WS-Policy Policies



Outline

- Web Services Policy Background
- **XACML Web Services Policy Assertions**
- XACML Assertion Format
- XACML Assertion Matching
- Defined XACML Assertions
  - XACMLAuthzAssertion
  - XACMLPrivacyAssertion
- New XACML Functions and Attribute Identifiers
- Open Issues

XACML Policy Assertions? Some use cases:

- Shibboleth service requires a particular Attribute for access
  - > Role="Full-time Student"
- Service requires Attributes to be signed by a trusted 3rd party
  - > AttributeSigner in {Sun, Liberty, JCP}
- Client requires that Service not reveal PII to any 3rd party
  - > P3P/Recipient="Ours"

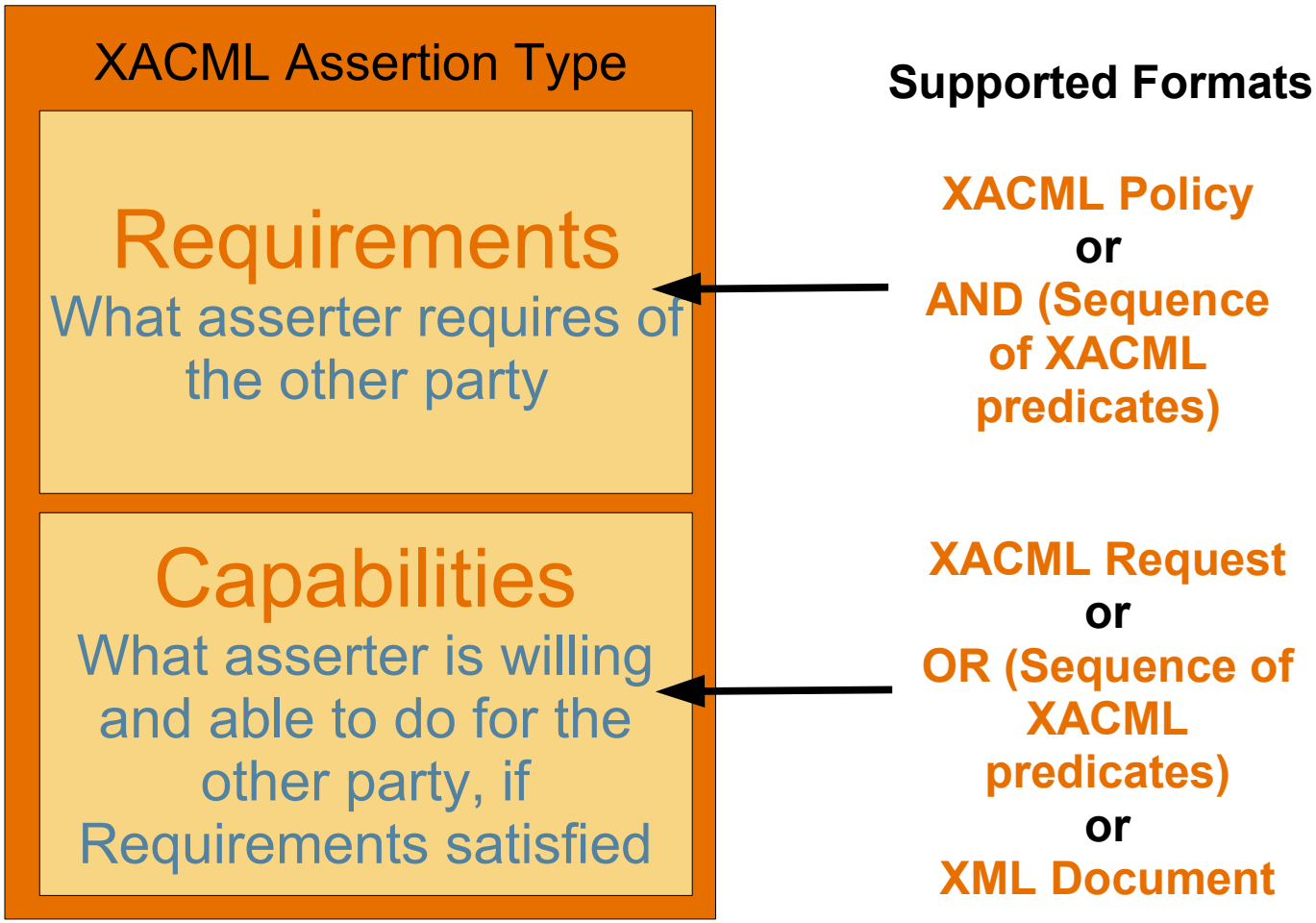
XACML Policy Assertion Options

- Separate Assertion for each constraint or capability
  - Which domain-specific policy engine to use for each?
- <xacml:Policy> as Assertion
  - How to express capabilities (offerings)?
  - How to match client and service Assertions?
- <xacml:Policy> and <xacml-context:Request> as Assertions
  - How does generic policy engine (e.g. WS-Policy) know to match them?

Outline

- Web Services Policy Background
- XACML Web Services Policy Assertions
- **XACML Assertion Format**
- XACML Assertion Matching
- Defined XACML Assertions
  - XACMLAuthzAssertion
  - XACMLPrivacyAssertion
- New XACML Functions and Attribute Identifiers
- Open Issues

XACMLAssertionAbstractType



XACML Assertion Requirements and Capabilities

- Each includes a “Vocabulary” element
  - One or more URIs associated with Attribute vocabulary
- Requirements formats
  - XACML Policy, or
  - AND (Predicate1, Predicate2, ...)
- Capabilities formats
  - XACML Request, or
  - OR (Predicate1, Predicate2, ...), or
  - XML Document (e.g. a P3P Policy)

XACML Predicates Usage

- Support matching two policies
  - client <=> service or service <=> service
- Fit WS-XACML use cases well
  - Fairly simple policies
- Also good for describing required Attributes

XACML Predicates Format

- Restricted form of <Apply> element
  - Exactly one AttributeDesignator/Selector, compared to literal(s)
    - role string-is-in {“Vice President”, “Director”}
    - tokenType anyURI-equals “urn:tokens:x509Token”
    - maxRetentionDays integer-less-than “100”
    - current-time time-in-range “8am”,”5pm”
  - Restricted, but powerful set of comparison functions
  - No nested functions except for one-and-only, limit-scope
- One predicate per Attribute
  - Exception: ranges retentionDays >= 5, retentionDays <= 10

Outline

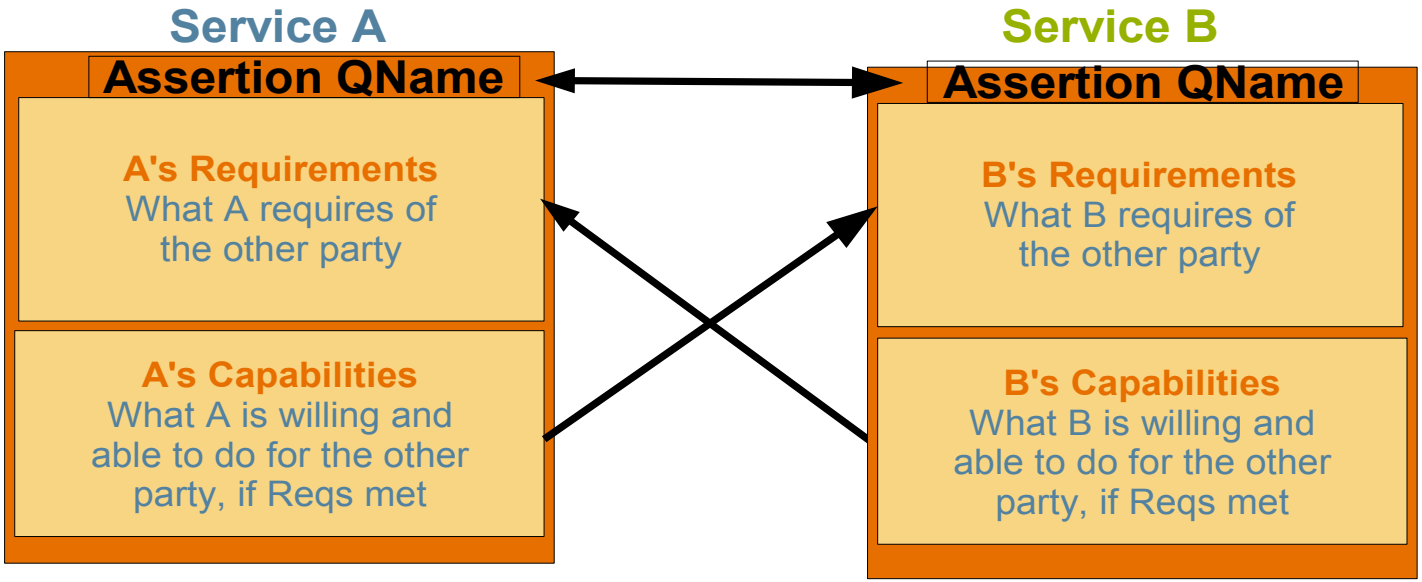
- Web Services Policy Background
- XACML Web Services Policy Assertions
- XACML Assertion Format
- **XACML Assertion Matching**
- Defined XACML Assertions
  - XACMLAuthzAssertion
  - XACMLPrivacyAssertion
- New XACML Functions and Attribute Identifiers
- Open Issues

XACML Assertion Matching Overview

1. Match QNames of Assertions
   - XACMLAuthzAssertion matches XACMLAuthzAssertion
   - XACMLPrivacyAssertion matches XACMLPrivacyAssertion

   > Matching done by generic policy engine (e.g. WS-Policy)
2. Match vocabularies of Requirements against vocabularies of Capabilities
3. Match Requirements against Capabilities
   - Standard XACML Request/Policy evaluation, or
   - Predicate matching

XACML Assertion Matching



- Domain-independent (WS-Policy Engine):
  QNames match?
- Domain-dependent (XACML Assertion Engine):
  Vocabularies match?
  B's Capabilities satisfy A's Requirements?
  A's Capabilities satisfy B's Requirements?

XACML Assertion Matching

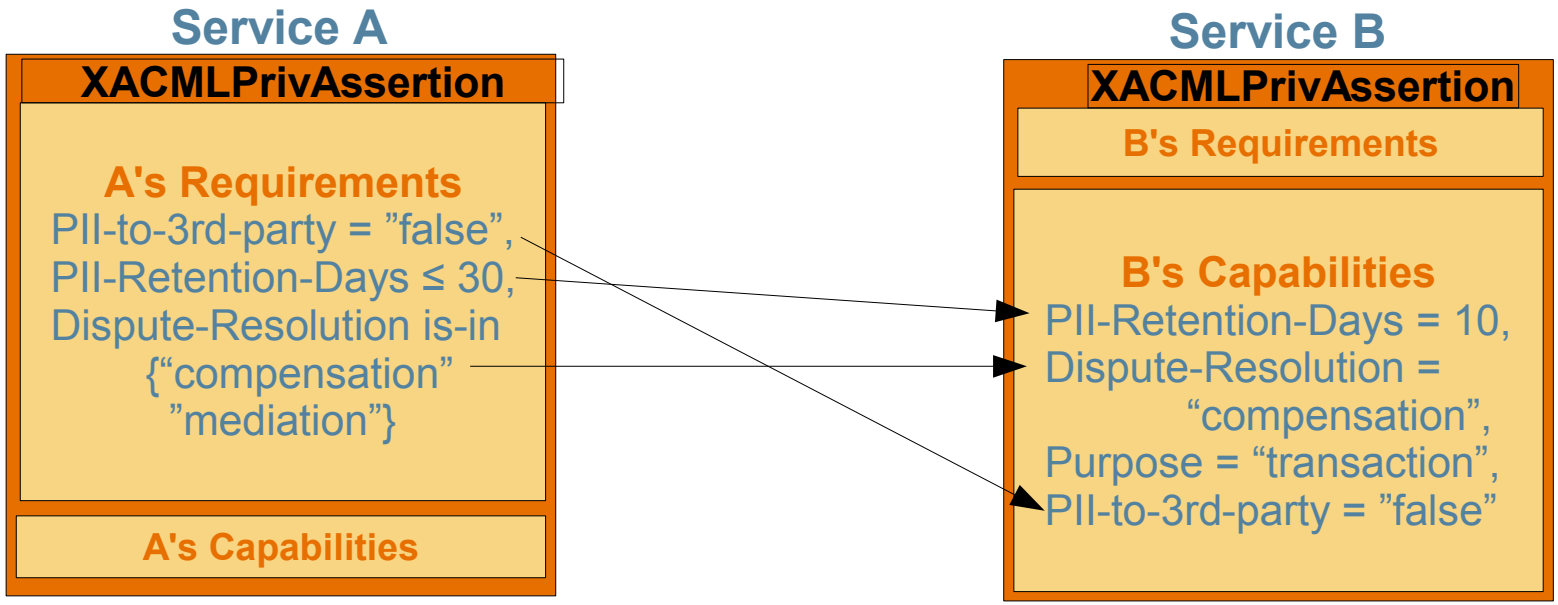
Service 1 Requirements Format	Service 2 Capabilities Format	Can Match?
XACML Policy	XACML Request or XML document	Yes
XACML Policy	Predicate list	**No**
Predicate list	XACML Request or XML document	Yes
Predicate list	Predicate list	Yes
Missing	Any	Always true
Any	Missing	Always true
Empty	Any	Yes, always true

XACML Assertion Vocabulary Matching

- Requirements Vocabulary MUST be a subset of the other Service's Capabilities Vocabulary
- Standard vocabulary URNs defined for
  - > XACML standard AttributeIds
  - > P3P 1.0 full schema
- Applications, products, consortia, sites define their own specific vocabularies

XACML Predicate Matching (1)

For each Service A Requirements predicate, find a Service B Capabilities predicate that references the same Attribute



PII: Personal Identifying Information

XACML Predicate Matching (2)

- There MUST be a Service B Capabilities predicate matching each Service A Requirements predicate
  - > At this step, “match” means references same Attribute
- If not, policies are incompatible. **STOP**
- There MAY be additional Service B Capabilities not needed to meet Service A Requirements

XACML Predicate Matching (3)

- Compute intersection of ranges of corresponding predicates
- If intersection empty, policies incompatible. **STOP**
- Express intersection as new <Apply> function
  - PII-to-3rd-party = “false” ∩ PII-to-3rd-party = “false”
    = PII-to-3rd-party = “false”
  - PII-Retention-Days ≤ “30” ∩ PII-Retention-Days = “10”
    = PII-Retention-Days = “10”
  - Dispute-Resolution is-in {“compensation”, “mediation”}
    ∩ Dispute-Resolution = “compensation”
    = Dispute-Resolution = “compensation”

XACML Predicate Matching (4)

- Result is new set of predicates
- Asymmetric results
  - OWN Requirements intersected with OTHER Capabilities => new OWN Requirements
  - OWN Capabilities intersected with OTHER Requirements => new OWN “required” Capabilities
- Computation of intersections not complex
  - Usually floor/ceiling operations
  - Table of intersections of all permitted function types explicitly specified in Appendix A

XACML Assertion Requirements and Capabilities Revisited

- Capabilities need not be published until “agreed-upon”
- Minimal disclosure protocol:
  1. Publish own Requirements
  2. Retrieve other Service's Requirements
  3. Match other Service's Requirements against own Capabilities (internally)
  4. Respond with intersection as own “agreed-upon” “required” Capabilities for interaction with this other Svc
  5. Use “agreed-upon” Capabilities from other Service as own Requirements for interaction with this other Svc

Outline

- Web Services Policy Background
- XACML Web Services Policy Assertions
- XACML Assertion Format
- XACML Assertion Matching
- **Defined XACML Assertions**
  - XACMLAuthzAssertion
  - XACMLPrivacyAssertion
- New XACML Functions and Attribute Identifiers
- Open Issues

Defined WS-XACML Assertions

- XACMLAuthzAssertion
  - For public authorization requirements and capabilities
  - Additional internal policy may be imposed at time interaction occurs
- XACMLPrivacyAssertion
  - For privacy/confidentiality Requirements and Capabilities
  - Client typically has privacy Requirements
    - Don't disclose my personal identifying information
  - Service typically has confidentiality Requirements
    - Don't disclose the price-list I send you
  - Service Capabilities MAY be a W3C P3P Policy
  - Client Capabilities typically IDs of PII willing to disclose

Outline

- Web Services Policy Background
- XACML Web Services Policy Assertions
- XACML Assertion Format
- XACML Assertion Matching
- Defined XACML Assertions
  - XACMLAuthzAssertion
  - XACMLPrivacyAssertion
- **New XACML Functions and Attribute Identifiers**
- Open Issues

New WS-XACML Functions

- Must-be-present
  - AttributeDescriptors or XPath expressions that must be present, but value may be anything
- Must-not-be-present
  - AttributeDescriptors or XPath expressions that must not be present with any value
- Limit-scope:all
  - True if multiple constraints on descendants of all nodes selected by a specified XPath expression are true
- Limit-scope:atLeastOne
  - True if multiple constraints on descendants of at least one node selected by a specified Xpath expression are true

New WS-XACML AttributeIds

- Maximum data retention days
- Data disclosures allowed
  - > Values from those defined for P3P RECIPIENT element
  - > Two additional values
    - – None
    - – Named (may disclose to names in “Named recipient” attr)
- Named recipient
  - > Value is identity of a permitted data recipient
  - > Used with 'Data disclosures allowed = “named”'

Outline

- Web Services Policy Background
- XACML Web Services Policy Assertions
- XACML Assertion Format
- XACML Assertion Matching
- Defined XACML Assertions
  - XACMLAuthzAssertion
  - XACMLPrivacyAssertion
- New XACML Functions and Attribute Identifiers
- **Open Issues**

Open Issues #55-57

#55: Policy References in an XACMLAssertion Policy
- Resolution: referenced policies must be present in new "ReferencedPolicies" element in the Requirements

#56: "Preference" XML attribute in <Apply> element
- To indicate which end of a range or set is preferred, optional
- Open: and should it also go into Core?

#57: Restricted subset of XPath for WS-XACML predicate AttributeSelectors
- Necessary to know which predicates to intersect and to ensure nodes referenced in two predicates will be the same
- Open: Proposal: absolute only, no query operators

Open Issues #58-60

#58: P3P 1.0 POLICY/STATEMENT/NON-IDENTIFIABLE
- Interacts with other “capabilities” in a P3P policy
- Open: no proposal yet

#59: Regular expressions as predicate functions
- Can be somewhat expensive to intersect, requires “basic” regular expressions
- Resolution: allow “basic” regular expressions

#60: Move non-Assertion parts of WS-XACML to SAML Profile
- AuthzToken and conveying XACML Attributes in SOAP header
- Pending Review: SAML Profile WD3 has these. Not positive they should have be moved, though.

Open Issues #61

#61: XACMLAuthzAssertions in regular policies

- Can predicates for XACMLAuthzAssertions be marked for extraction from regular policies?
- Open: Proposal: No. Could be in Policies combined with others, however.

References

[1] *Web Services Profile of XACML (WS-XACML) Version 1.0*, Working Draft 8, 12 December 2006, OASIS XACML Technical Committee, http://www.oasis-open.org/committees/download.php/21490/xacml-3.0-profile-webservices-spec-v1.0-wd-8-en.pdf

[2] *SAML 2.0 Profile of XACML, Version 2*, Working Draft 3, 6 March 2007, http://www.oasis-open.org/committees/download.php/22765/xacml-profile-saml2.0-v2-wd-2.zip

Web Services Profile of XACML (WS-XACML)_

Anne Anderson
anne.anderson@sun.com