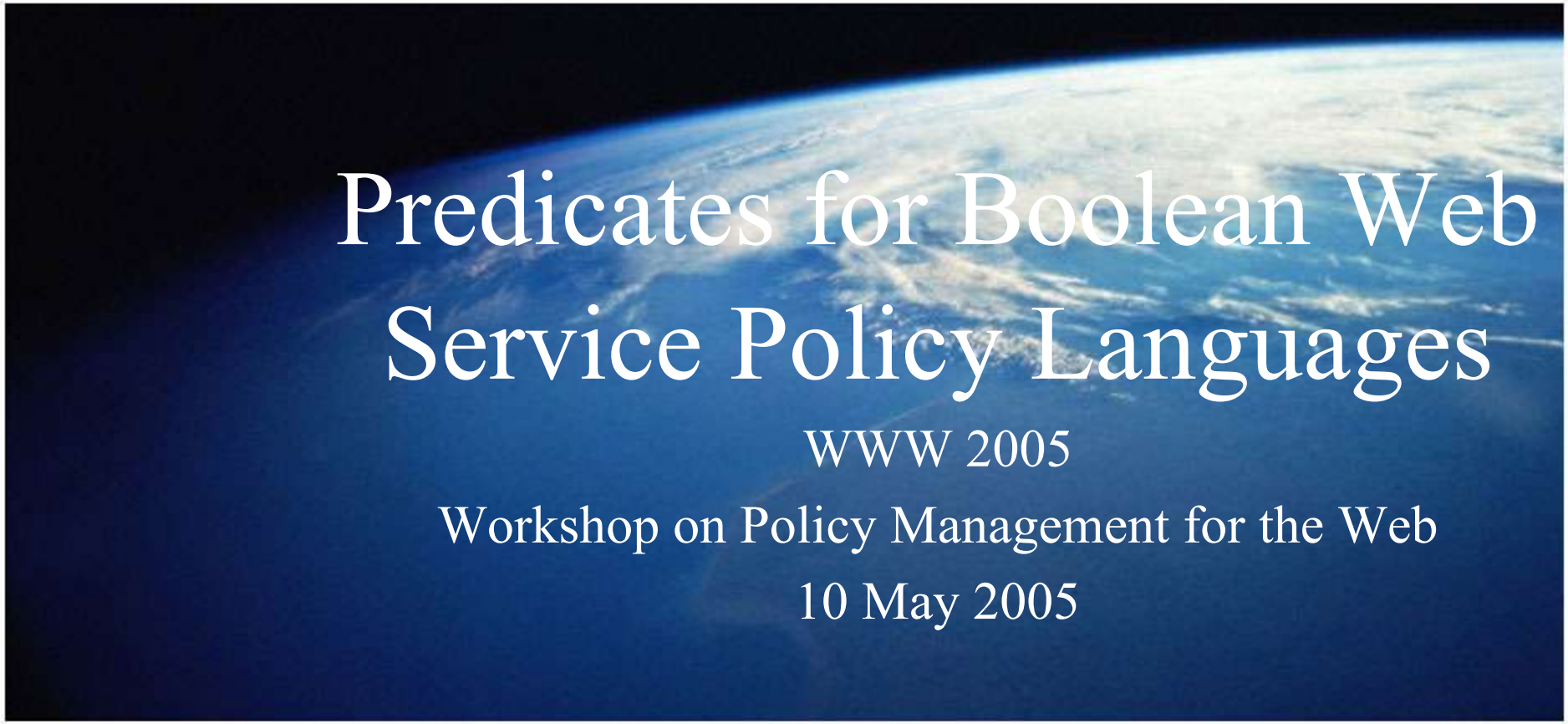


Predicates for Boolean Web Service Policy Languages

WWW 2005

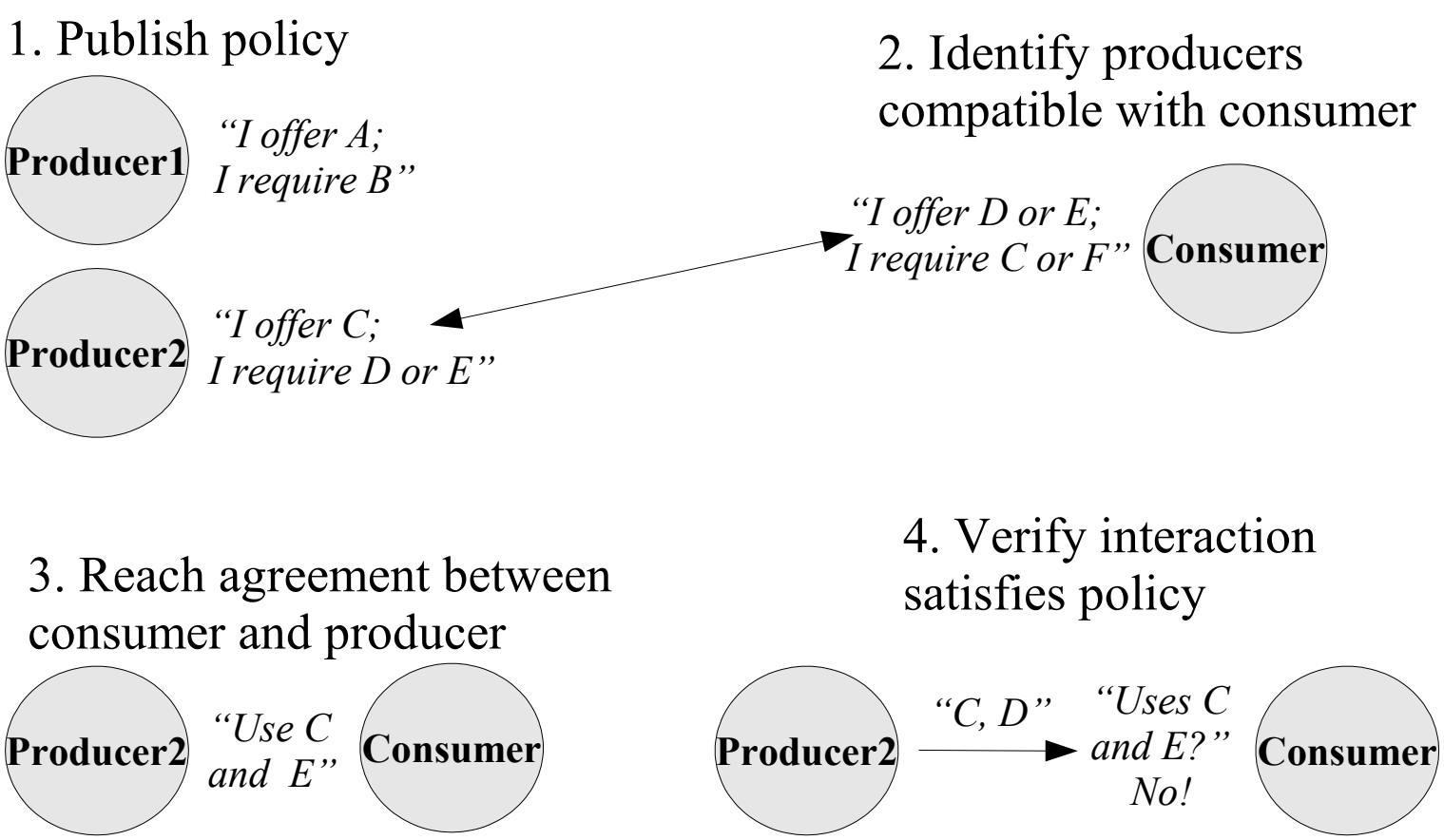
Workshop on Policy Management for the Web

10 May 2005

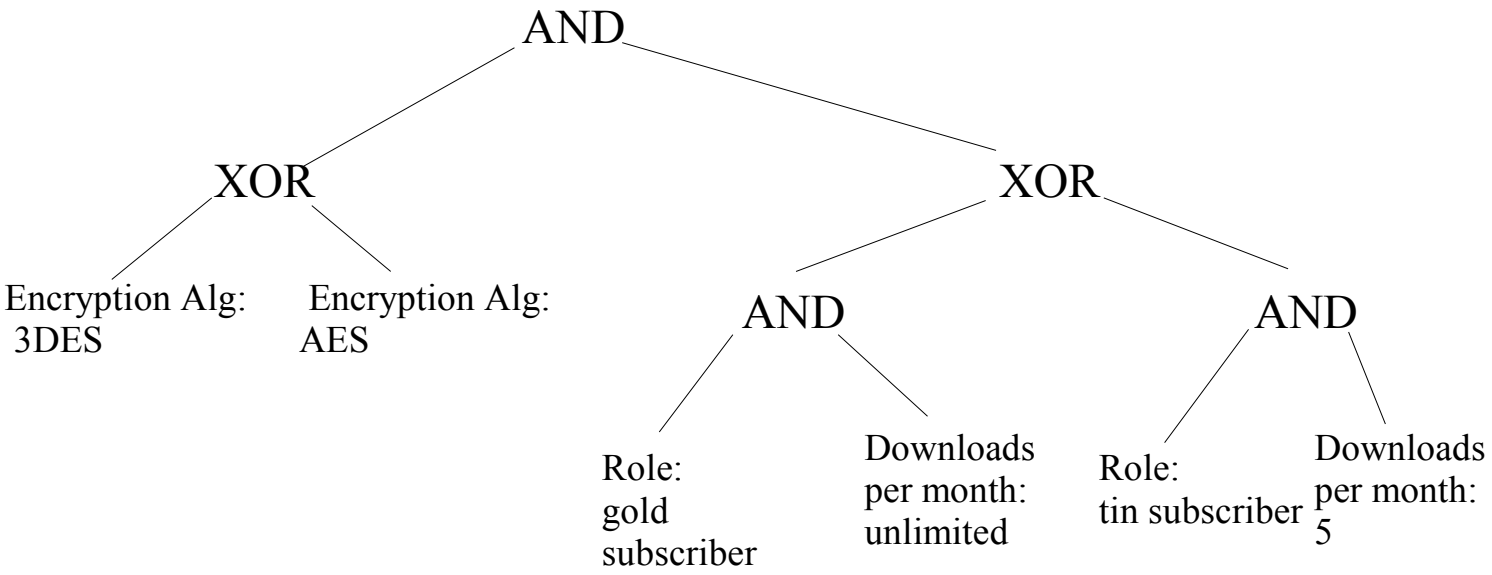
Anne Anderson
Staff Engineer
Sun Microsystems Labs
Burlington, MA, USA
Anne.Anderson@sun.com

Sun microsystems
We make the net work.

Uses of a web service policy language



Boolean policy languages



Two approaches

	WS-Policy	XACML WSPL
Predicate combiners	*Boolean Operators*	
Predicates	***Domain-specific***	***Standard expressions***
Domain-specific policy vocabularies	*WS-Security*	*WS-RX* ...

Example item for policy: WS-Security UsernameToken

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
   <S:Header>
        ...
      <wsse:Security>
        <wsse:UsernameToken>
          <wsse:Username>Zoe</wsse:Username>
          <wsse:Password>ILoveDogs</wsse:Password>
        </wsse:UsernameToken>
      </wsse:Security>
        ...
   </S:Header>
   ...
</S:Envelope>
```

WS-Policy predicate example

Defined in: WS-SecurityPolicy

```
<wssp:Integrity>
    <wssp:TokenInfo>
      <wsse:SecurityToken wsu:id="..."/>
        <wsse:TokenType>wsse:UsernameToken</TokenType>
        <wsse:Claims>
          <wssp:SubjectName MatchType=
                    "wsse:Prefix">Zoe</SubjectName>
          <wssp:UsePassword Type="wsse:PasswordDigest"/>
        </wsse:Claims>
      </wsse:SecurityToken>
    </wssp:TokenInfo>
</wssp:Integrity>
```

WS-Policy predicate example semantics

Defined in WS-SecurityPolicy

When the TokenType is wsse:UsernameToken, the TokenIssuer element in a SecurityToken assertion MUST be absent.

...

/SecurityToken/Claims/SubjectName/@MatchType

The value of this optional attribute MAY be one of wsse:Exact, wsse:Prefix, and wsse:Regexp. The interpretation of the matching operation is given in the table below. If this attribute is omitted, the default value is wsse:Prefix.

QName	Description
wsse:Exact	The values must be exactly the same.
wsse:Prefix (default)	The specified value must be the prefix of the value in the certificate.
wsse:Regexp	The specified value is an regular expression that matches the value in the token.

XACML WSPL predicate example

Concept:

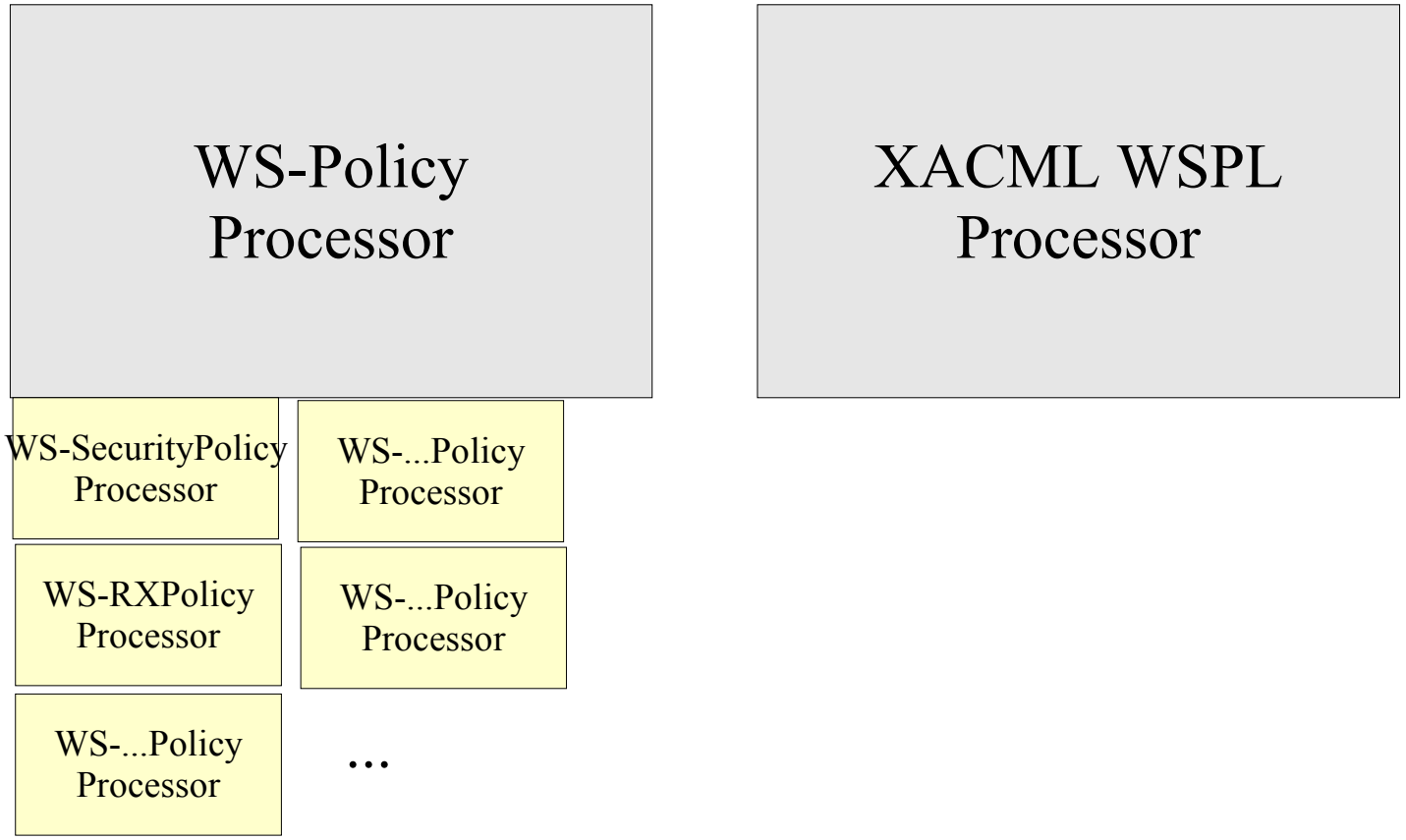
```
AND (
    Regular-expression-match (
        “Zoe.*”,
        //S:Envelope/S:Header/wsse:UsernameToken/wsse:Username/text()
    )
    Exact-match (
        “wsse:PasswordDigest”,
        //wsse:SecurityToken/wsse:Claims/wssp:UsePassword@Type
    )
 )
```

XACML WSPL predicate example

Defined in: XACML Standard

```
<xacml:Apply FunctionId="&xacml-function;and"
   <xacml:Apply FunctionId="&xacml-function;string-regexp-match">
      <xacml:AttributeValue
            DataType="&xsd;string">Zoe*</xacml:AttributeValue>
      <xacml:AttributeSelector RequestContextPath=
            "//S:Envelope/S:Header/wsse:UsernameToken/wsse:Username/text()"
            DataType="&xsd:string"/>
   </xacml:Apply>
   <xacml:Apply FunctionId="&xacml-function;string-equal"
      <xacml:AttributeValue DataType="&xsd;string">
            wsse:PasswordDigest</xacml:AttributeValue>
            DataType="&xsd:string"/>
      <xacml:AttributeSelector RequestContextPath=
            "/wssp:UsePassword@Type"DataType="&xsd;string"/>
   </xacml:Apply>
</xacml:Apply>
```

Implementation of functions



XACML WSPL Issues

- All related to finding compatible policies
  - XPath expression intersections not well-defined
    - Example: /X[2] =? //Y/X[@Z]
    - Solution: define an XPath subset
  - Comparing non-XML data
    - Example: extracting fields from an X.509 certificate
    - Solution: additional functions: getCertExtValue
    - Hope: most can be standard
    - Basis: legacy data only; new data XML
  - Multiple alternative syntaxes in specifications
    - Example: `/ds:SignedInfo/ds:Reference[@URI]`
    - Solution: profile specification for use with policies

Conclusion

- Two approaches to representing predicates
  - WS-Policy:

    Domain-specific vocabulary syntax and semantics

    Domain-specific policy expression syntax and semantics;
  - XACML WSPL:

    Domain-specific vocabulary syntax and semantics

    Standard policy expression syntax and semantics;
- Standard expression syntax advantages
  - Standard policy processors
  - No need for domain-specific modules
- Solutions exist for standard expression issues