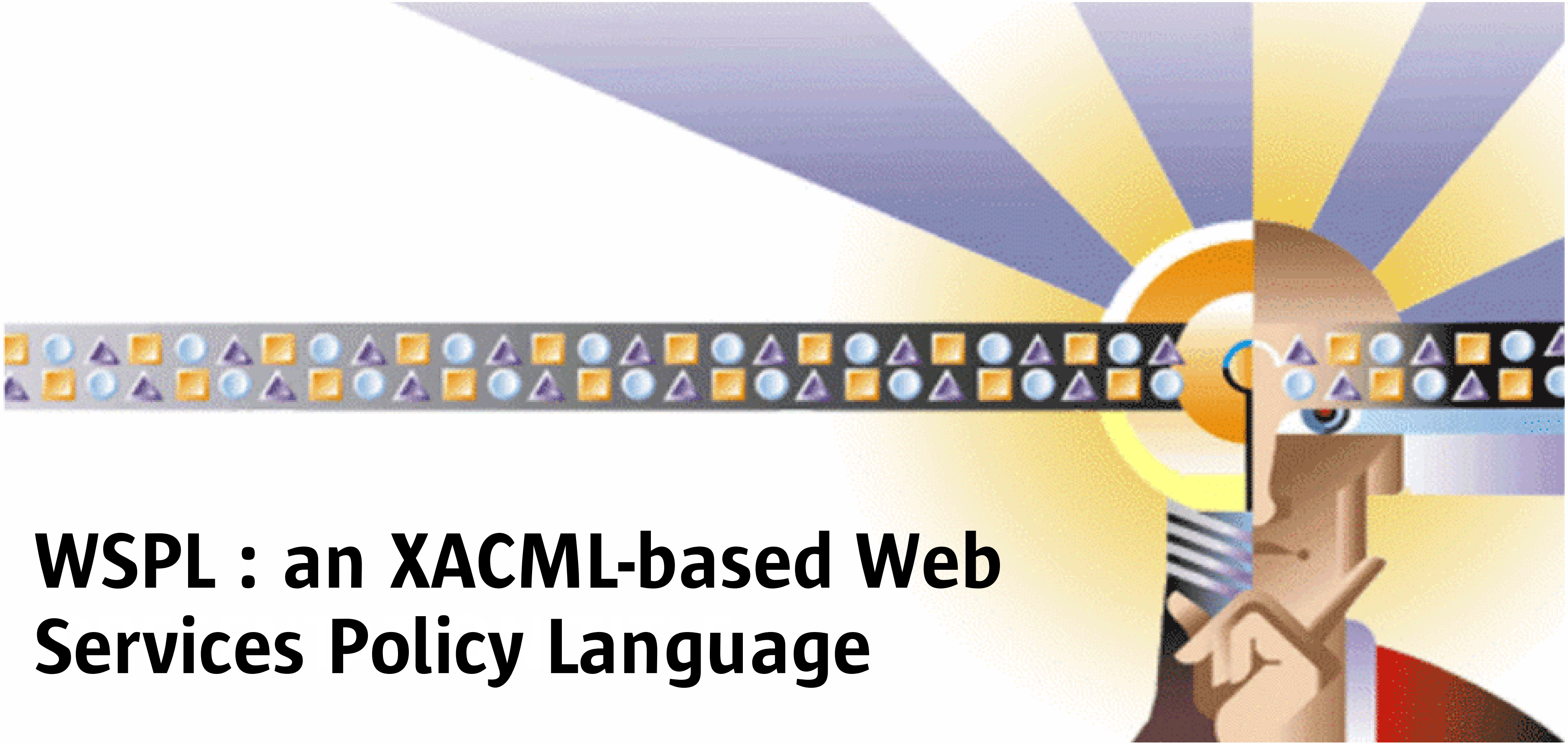


WSPL : an XACML-based Web Services Policy Language

Anne Anderson

Staff Engineer

Sun Microsystems Laboratories

Outline

- Introduction
- Use cases and requirements
- Overview of the WSPL language
- Some design decisions
- Current status
- Conclusions

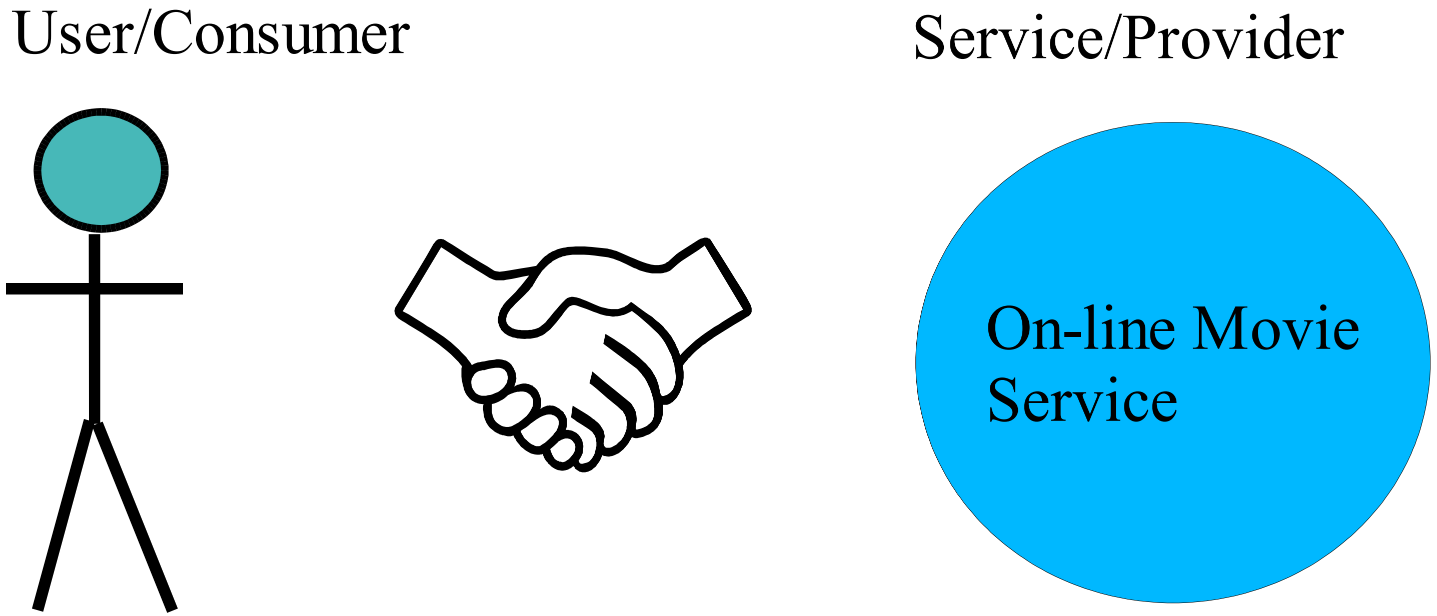
Introduction

- “Policy” - many things to many people
  - Guiding principles and procedures
  - Management policy
    Event -> Condition -> Action (ECA)
  - Interaction parameters
  - Authorization (access control) policy

Web services policies

- Authentication
- Authorization
- Quality of Protection (QoP)
- Quality of Service (QoS)
- Privacy
- Reliable messaging
- Service-specific options

Use cases (1)

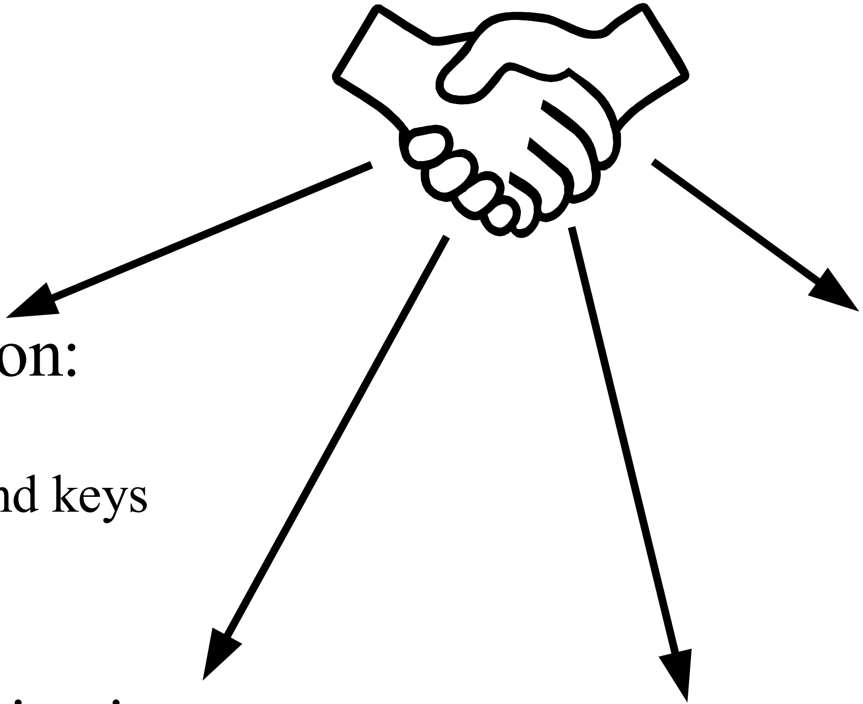


User/Consumer

Service/Provider

On-line Movie Service

Use cases (1)



Authentication:

- Method
- Algorithms and keys

Privacy:

- Share info?
- Store user info?
- Delete user info?

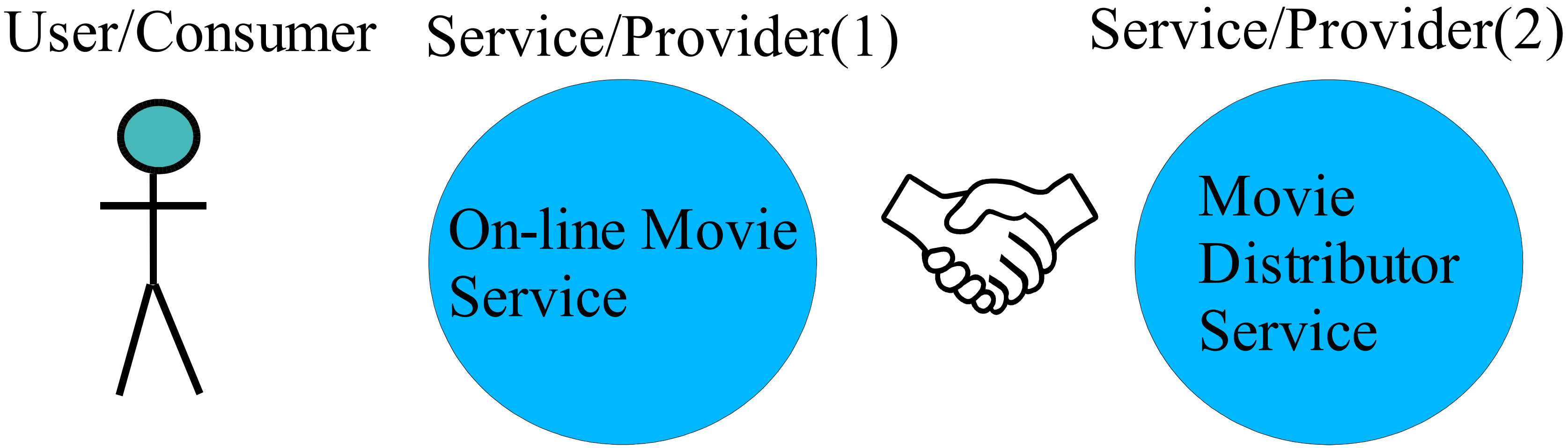
Authorization

- Subscribe/unsubscribe
- Download
- Manage

Service options

- # of movies/month
- Bandwidth guarantees
- Fee

Use cases (2)



User/Consumer

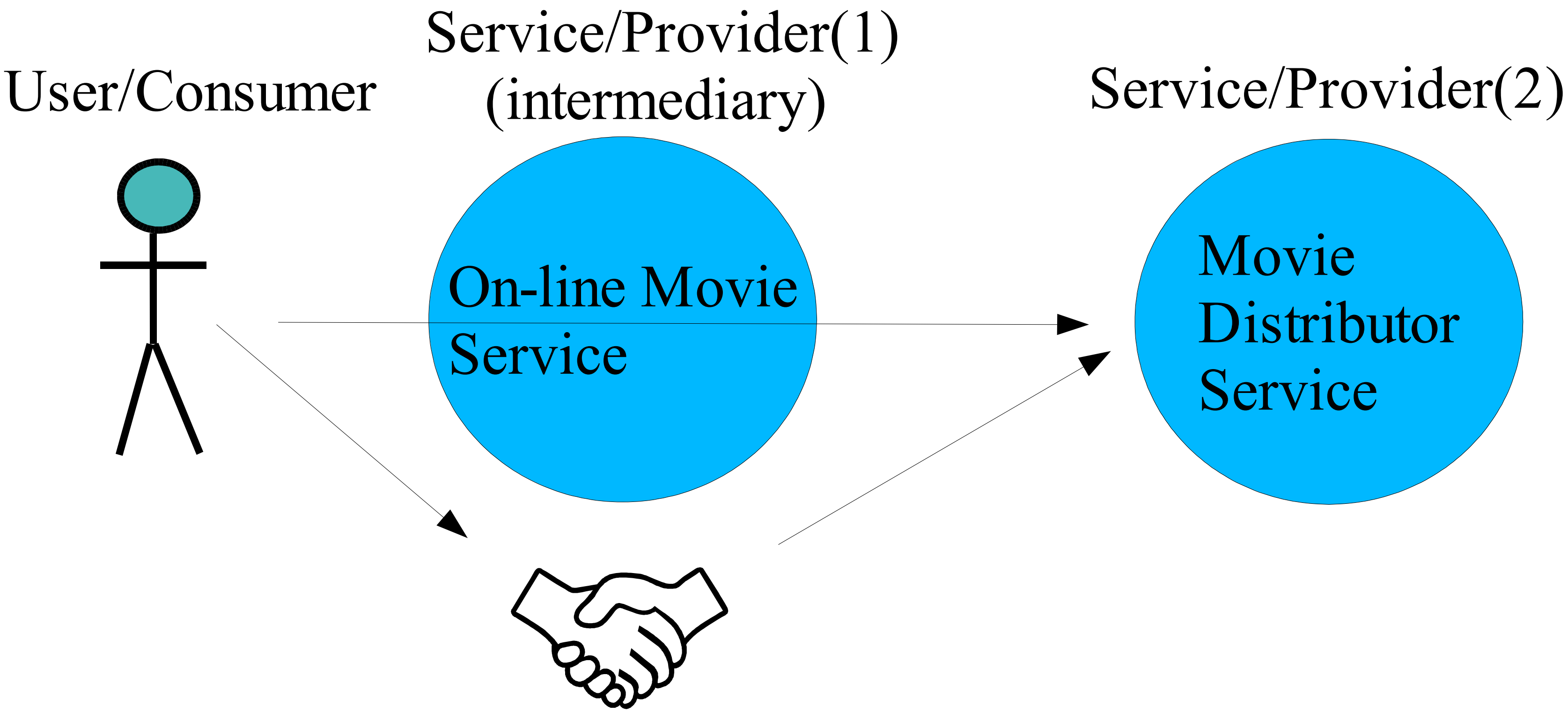
Service/Provider(1)

Service/Provider(2)

On-line Movie Service

Movie Distributor Service

Use cases (3)


User/Consumer
Service/Provider(1)
(intermediary)
Service/Provider(2)
On-line Movie
Service
Movie
Distributor
Service

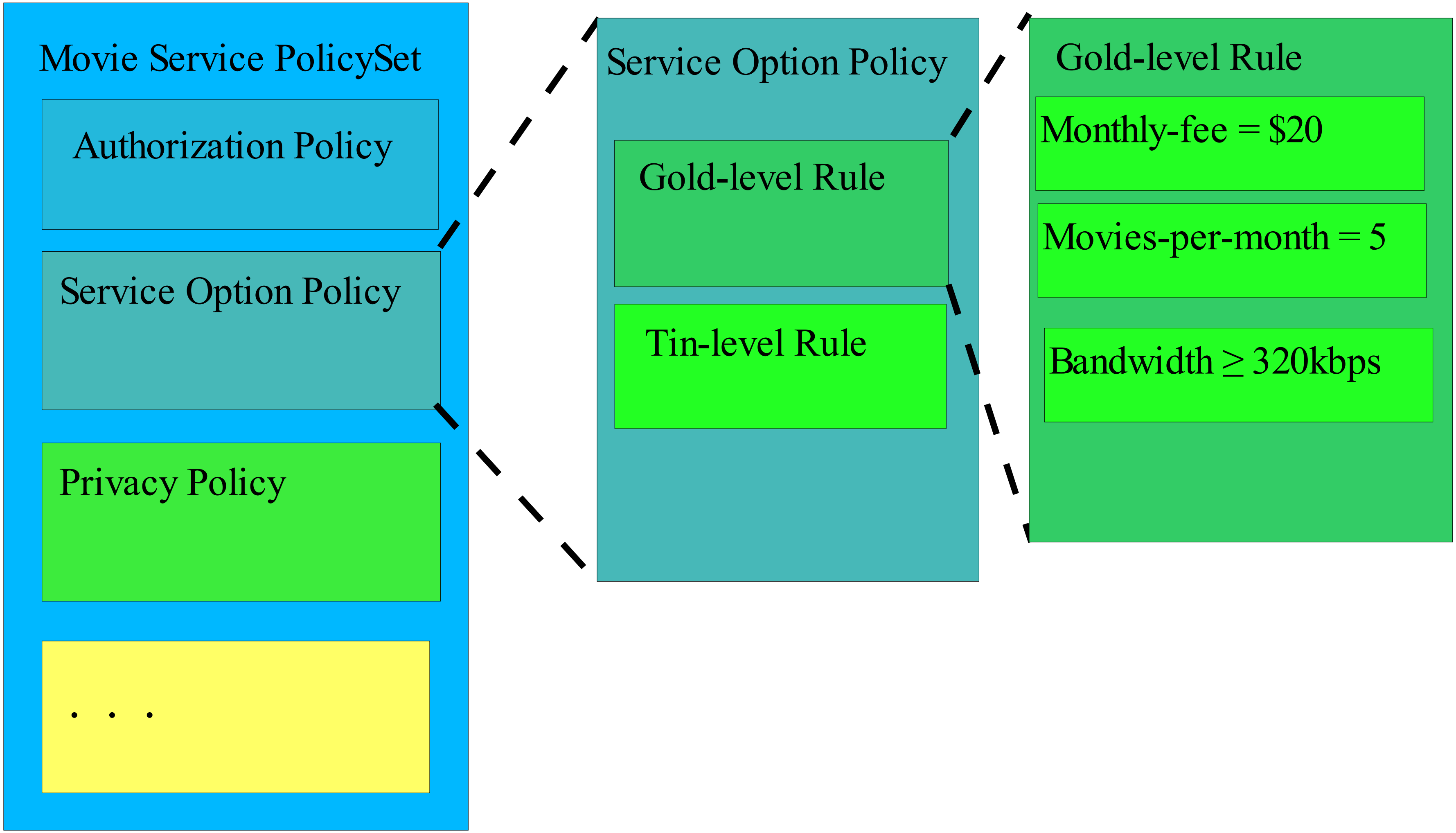
Negotiation is KEY

- Needed when choices exist
- Both sides have preferences, capabilities, requirements
- Needed to automate service discovery and connection
- WSPL supports this

WSPL Policy Structure

- A WSPL policy is a tree of
  - PolicySet
    - represents the policies of a particular service
    - contains multiple Policies
  - Policy
    - represents a single aspect of the service
    - contains sequence of Rules
  - Rule
    - represents an acceptable set of Attributes
    - contains predicates

WSPL policy example



Movie Service PolicySet

Authorization Policy

Service Option Policy

Privacy Policy

. . .

Service Option Policy

Gold-level Rule

Tin-level Rule

Gold-level Rule

Monthly-fee = $20

Movies-per-month = 5

Bandwidth ≥ 320kbps

Disjunctive Normal Form

- Policy logic
  - “Rule 1” OR “Rule 2” OR “Rule 3” ...
- Rule logic
  - “Predicate 1” AND “Predicate 2” AND “Predicate 3”...
- An “OR” of “AND”s

Predicates

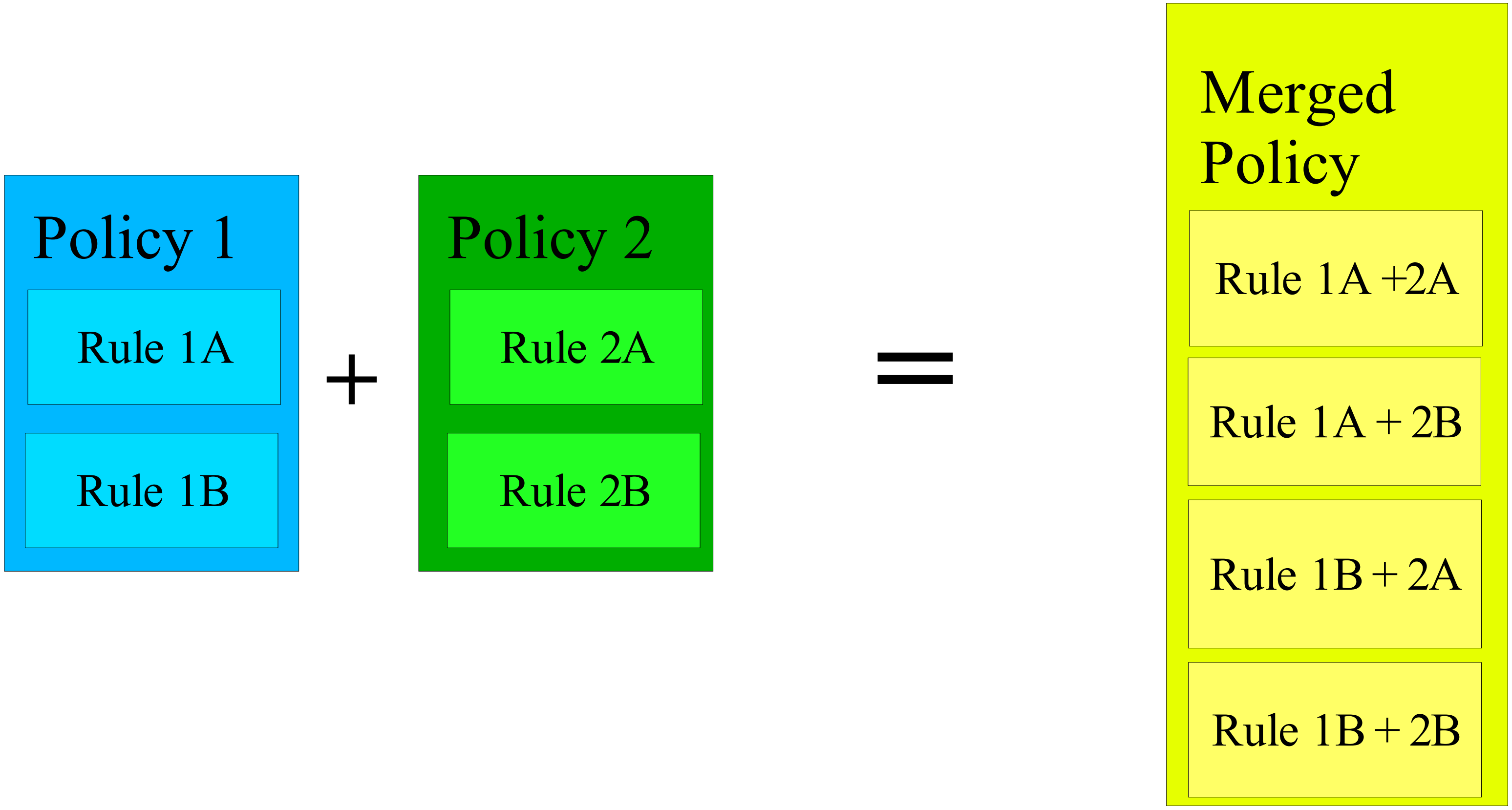
- Attribute DataTypes
  - xsi:integer, xsi:string, xsi:dateTime, xacml:x500Name, xacml:rfc822Name, ...
- Predicate operators
  - equal, greater-than, ..., set-equals, subset
- Can compare Attribute to literal or Attribute to Attribute

Policy negotiation

- Use cases
  - User policy <-> Service 1 policy
  - Service 1 policy <-> Service 2 policy
  - User policy <-> Service 2 policy (where Service 1 is an intermediary)
- Goal: find a single policy consistent with both input policies

Policy merging (1)

- Pair rules in all possible combinations



Policy 1

Rule 1A

Rule 1B

+

Policy 2

Rule 2A

Rule 2B

=

Merged Policy

Rule 1A +2A

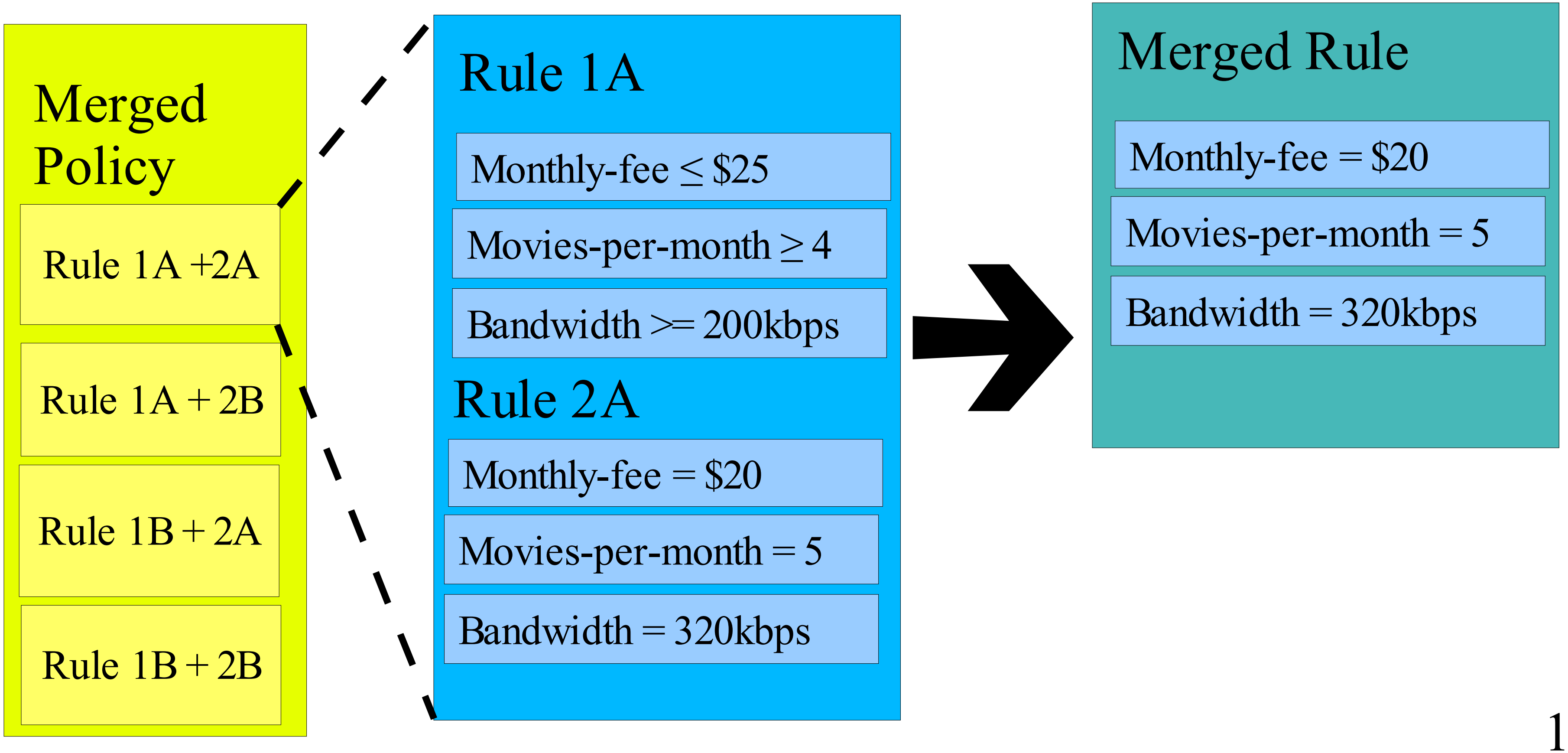
Rule 1A + 2B

Rule 1B + 2A

Rule 1B + 2B

Policy merging (2)

- Merge rules



Merged Policy

Rule 1A +2A

Rule 1A + 2B

Rule 1B + 2A

Rule 1B + 2B

Rule 1A

Monthly-fee ≤ $25

Movies-per-month ≥ 4

Bandwidth >= 200kbps

Rule 2A

Monthly-fee = $20

Movies-per-month = 5

Bandwidth = 320kbps

Merged Rule

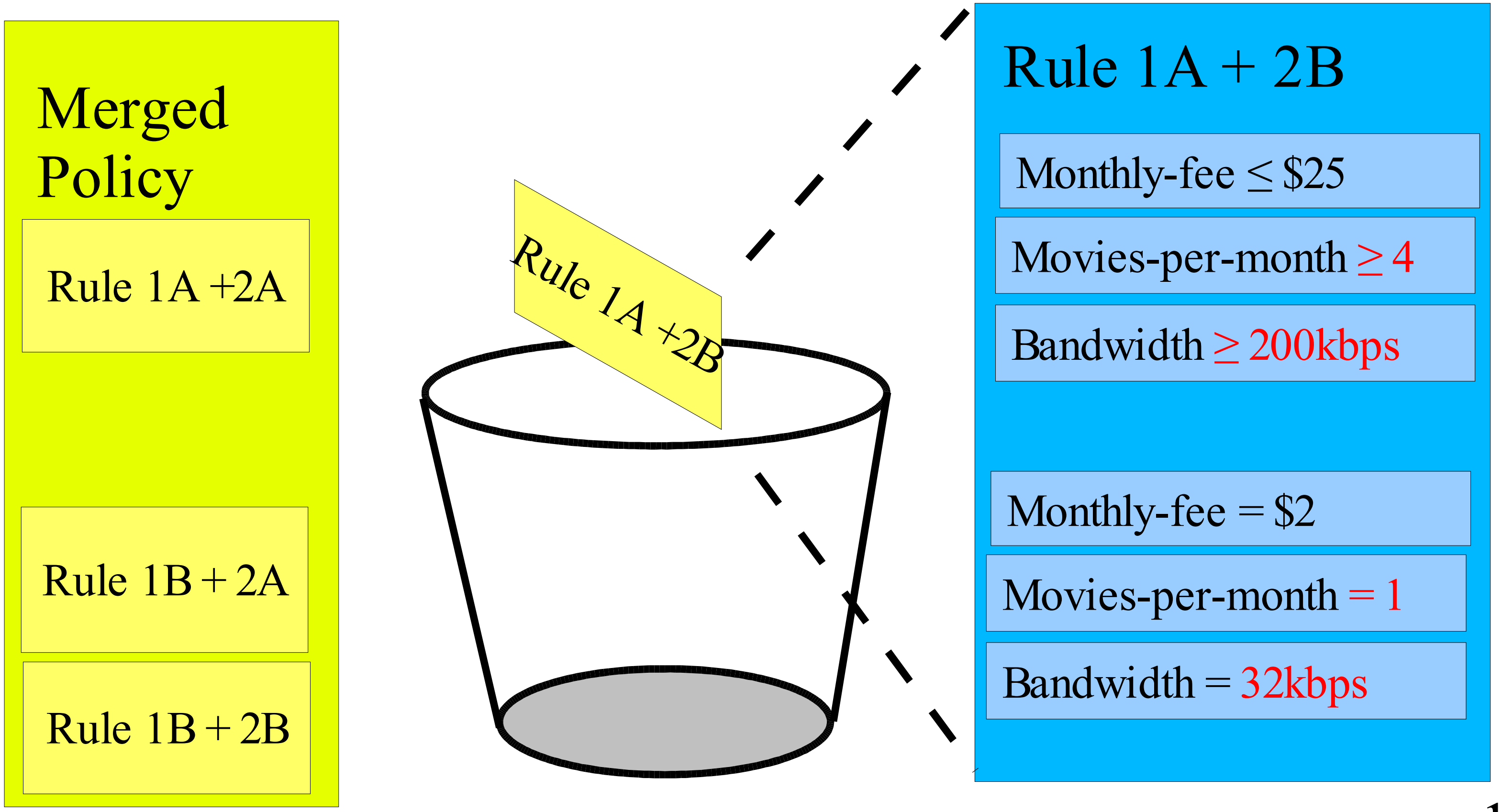
Monthly-fee = $20

Movies-per-month = 5

Bandwidth = 320kbps

Policy merging (3)

- Eliminate incompatible rules



Merged Policy

Rule 1A +2A

Rule 1B + 2A

Rule 1B + 2B

Rule 1A +2B

Rule 1A + 2B

Monthly-fee ≤ $25

Movies-per-month ≥ 4

Bandwidth ≥ 200kbps

Monthly-fee = $2

Movies-per-month = 1

Bandwidth = 32kbps

Policy merging (4)

- Eliminate unusable rules

  Currently:

  timeOfDay == 6pm

  Rule:

  timeOfDay ≥ 9am

  timeOfDay ≤ 5pm

Preferences

- Policy Rules are in preference order
- Preserve combiner's preference order, then other Policy's order
- Requester/client is usually the combiner

Relationship to XACML

- Strict subset of XACML[*] syntax
- Different evaluation engines
  - XACML: given a set of Attributes and a Policy, is the set acceptable or not?
  - WSPL: given two Policies, what are the acceptable sets of Attributes?

[*] OASIS eXtensible Access Control Markup Language

Some design decisions

- XACML-based
  - Attributes are name/value pairs
- Limited datatypes and operators
- Disjunctive normal form

Current status

- Draft in OASIS XACML TC
- Prototype implementation done at Brown
  - Based on Sun's Open Source XACML Implementation
- XACML TC may progress WSPL for authorization oly
- Possible OASIS WSPL TC

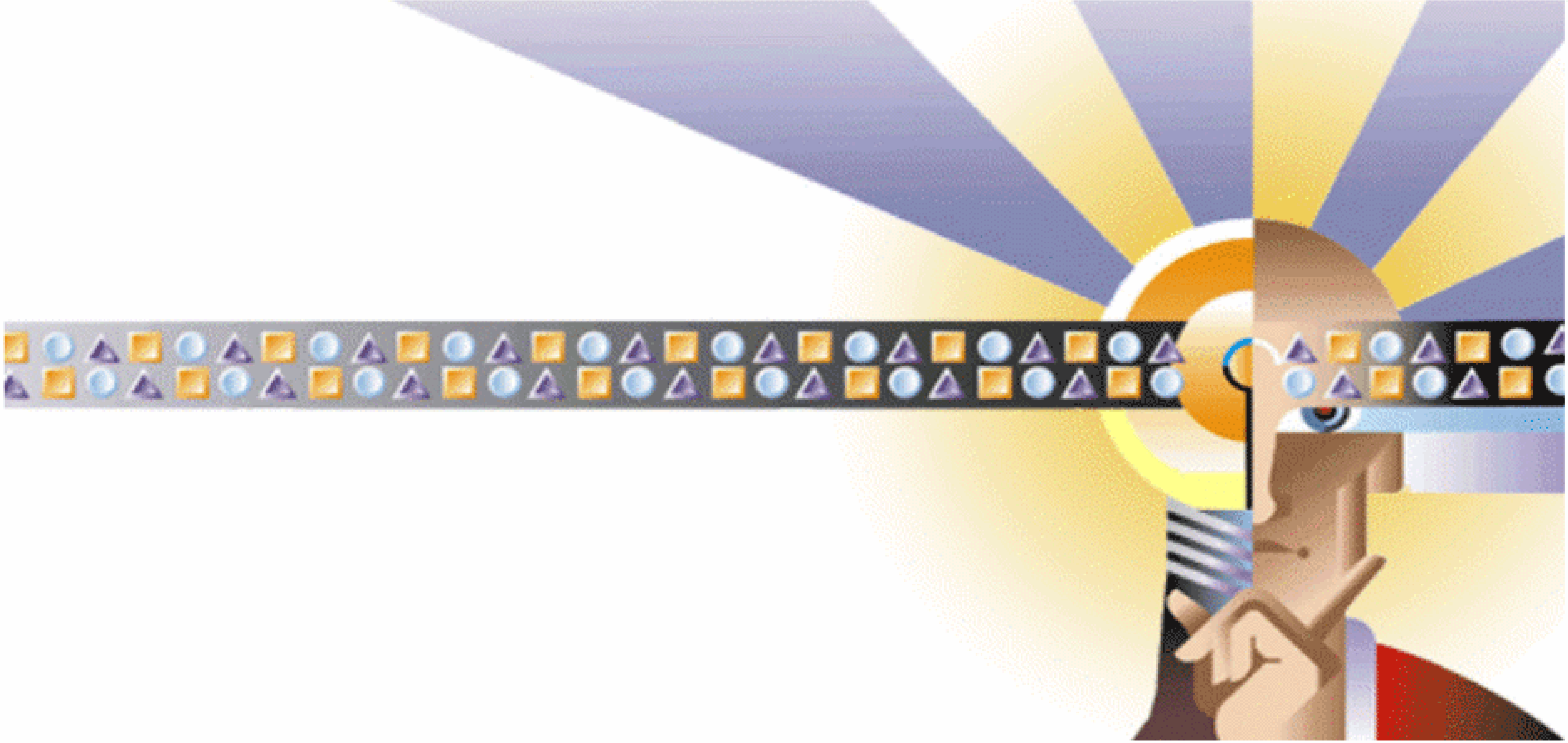
Conclusion

- WSPL
  - Requirements-based
  - Standards-based
  - Formally analyzed
  - Supports policy negotiation
  - Supports comparison-based requirements
- Good basis for a web services policy standard

References

- OASIS XACML TC Web Page
  - http://www.oasis-open.org/committees/xacml
    - “Web-services policy language use-cases and requirements”
    - “XACML profile for Web-services” (WSPL)
- Sun Labs

  http://research.sun.com



Anne Anderson

Anne.Anderson@sun.com

Sun microsystems

We make the net work.