LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

Semantic WS-Agreement Partner Selection (SWAPS)

WWW2006, Edinburgh, May 2006.

Nicole Oldham, Kunal Verma, Amit Sheth, Farshad Hakimpur

LSDIS Lab, University of Georgia,
Athens, Georgia, UGA

paper

SWAPS Project Page

A component of the METEOR-S project

Outline

- Introduction
- WS-Agreement – Details and Ontology
- Matching Algorithm
- Architecture
- Detailed Example
- Conclusion

Introduction

- Technology View: SOA brings exciting new possibilities
  - Dynamic discovery of new partner services
  - Dynamic binding of service
- Business View: What about the contracts or agreements
  - Issues of service level guarantees
  - Need to represent business objectives, penalties etc.
- WS-Agreement
  - Provides a mechanism to represent contracts
    - Chose WS-Agreement over WS-Policy since its more expressive

Research Problem

- Agreement provides rich constructs to represent contracts
  - Service Level Objectives, Qualifying conditions, Business Values
- Focus so far on XML based representation
  - Efficient matchmaking is the key for dynamic partner selection
  - Matching based on XML is syntactic and limited to string matching
- We propose using semantics for flexible matchmaking
  - Using domain knowledge stored in ontologies and rules for better matches

Challenges

1. Heterogeneous Service Level Objectives: different ways to say the same thing

For Example: 98% of responses < 2s

responseTime < 2s with 96% as threshold

2. Objectives can only be met under certain conditions

How can we determine which conditions are more suitable for consumer?

For Example: transactionRate vs weekday

3. Tradeoffs exist for different consumers

A Consumer may prefer certain business values over other factors.

For Example: provider1 rt < 10 s and penalty 15 USD

provider2 rt < 5s and penalty 1 USD

*consumer may prefer slower with higher penalty

4. Requirements can be specified by both parties

Matching require that both consumer and provider's requirements are met. (Or it may only be imposed on one of the parties)

5. Alternatives

Each agreement may contain alternative sets of guarantees

Proposed Solution

- Use knowledge stored in OWL ontologies and SWRL rules for flexible matchmaking. WSDL-S for semantic annotation of WSDL.

- Three types of rules used
  - Domain Specific Rules
    - To handle heterogeneities between provider and requester agreements
    - These rules help matching assertions that are syntactically heterogeneous but semantically similar
  - Semantics of Predicate Rules
    - To precisely define semantics of the predicates used for making assertions
    - These rules help make matchmaking precise
  - User Specific Rules
    - To capture user preferences for matchmaking
    - Declaratively specifying preferences instead of having to write extra code

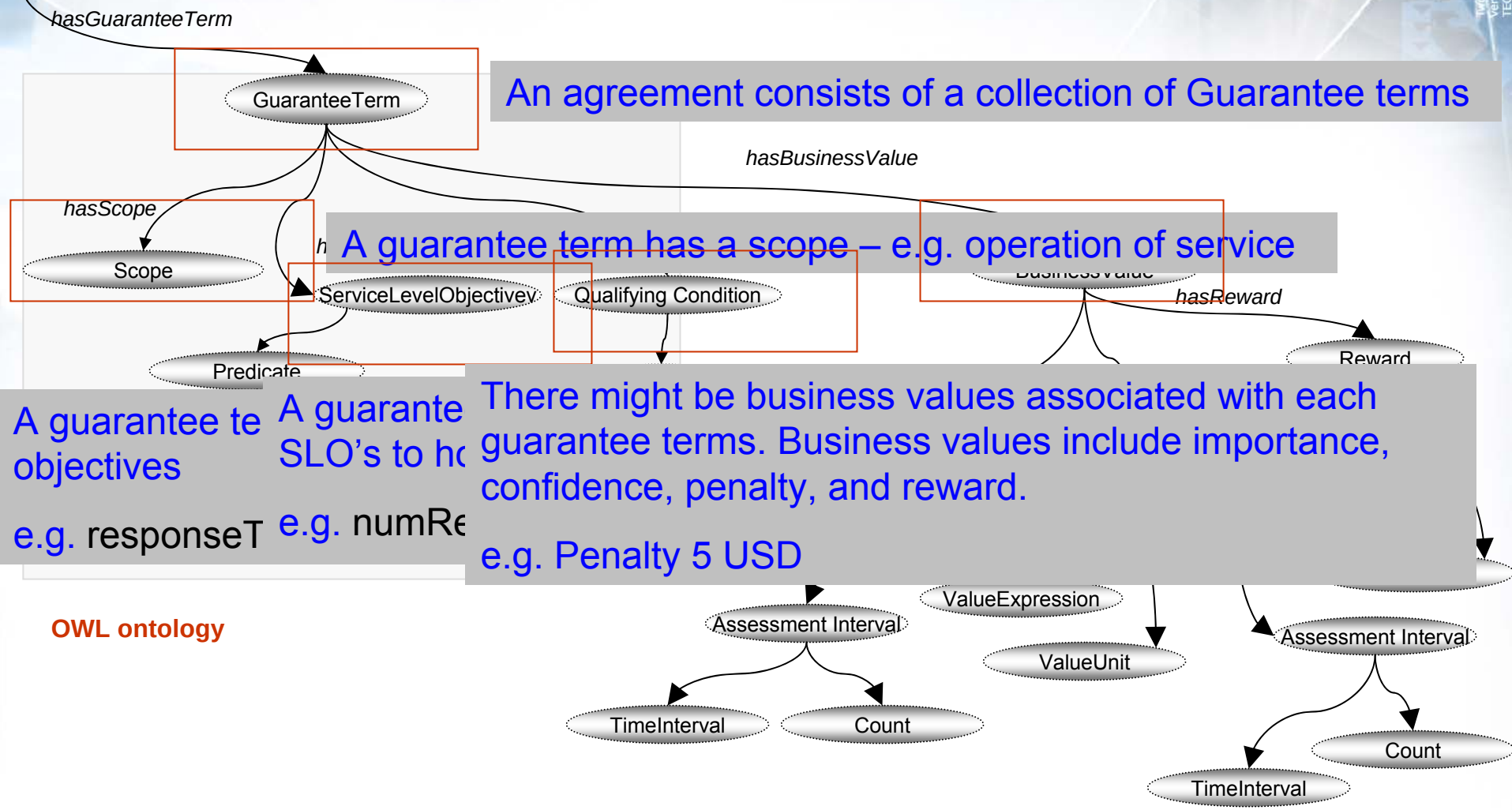
Outline

- Introduction
- WS-Agreement – Details and Ontology
- Matching Algorithm
- Architecture
- Detailed Example
- Conclusion

WS-Agreement

WS-Agreement Tag	Purpose
Obligated	States the party responsible for the fulfillment of the guarantee. Value will be ServiceProvider or ServiceConsumer
ServiceScope	Describes to what service element specifically a service applies.
ServiceLevelObjective (SLO)	An assertion over the terms of the agreement which represents the QoS aspect of the agreement. Usually defines bounds usually over QoS concepts such as response time, fault rate or cost.
QualifyingCondition	Optional condition which must exist in order for the SLO to be satisfied. Usually over external factors such as time of day.
BusinessValueList	Optional values which represent the strength of commitment by stating penalties, rewards and importance

WS-Agreement Definition and Ontology

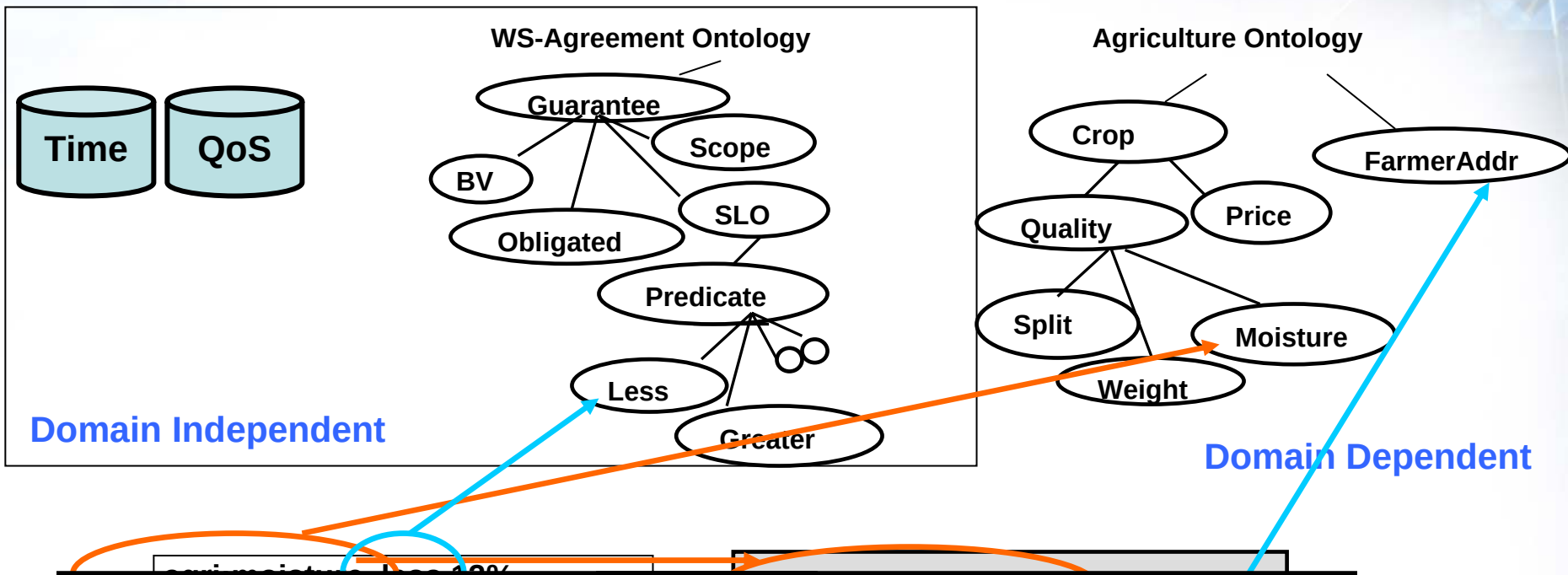


OWL ontology

SWAPS Ontologies

- **WS-Agreement:** individual agreements are instances of the WS-Agreement ontology
- **Temporal Concepts:** time.owl (OWL version of DAML time http://www.isi.edu/~pan/damltime/time.owl)
  - Concepts: seconds, dayOfWeek, ends
- **Quality of Service:** Max Maximilien's QoS ontology (IBM) -> Ont-Qos
  - Concepts: responseTime, failurePerDay
- **Domain Ontology:** an ontology used to represent the domain

Using Semantic Agreements with WSDL-S



Adding Semantics to Agree

Improves Monitoring and N

Improves the accuracy of n

Adding Semantics to Web Services:

Enables more accurate discovery and composition.

Outline

- Introduction
- WS-Agreement – Details and Ontology
- Matching Algorithm
- Architecture
- Detailed Example
- Conclusion

Agreement Matching

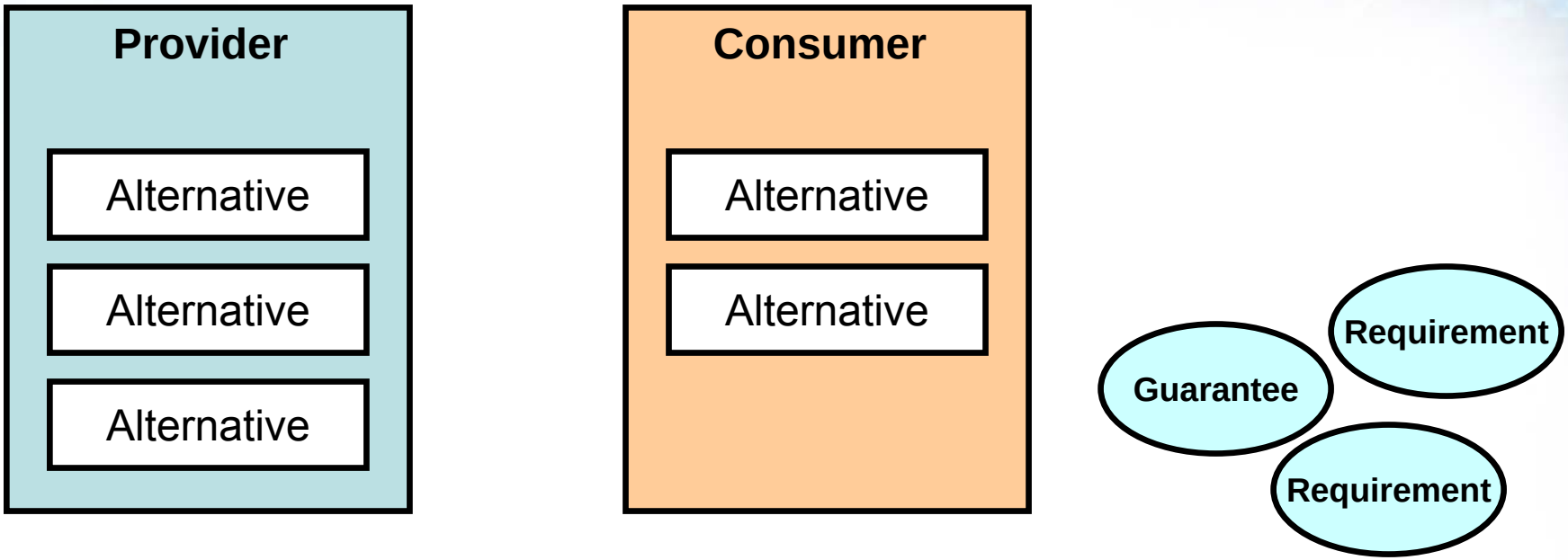
An agreement is a collection of alternatives.
A={Alt1, Alt2, …, AltN}

An alternative is a collection of guarantees.
Alt={G1, G2, …GN}

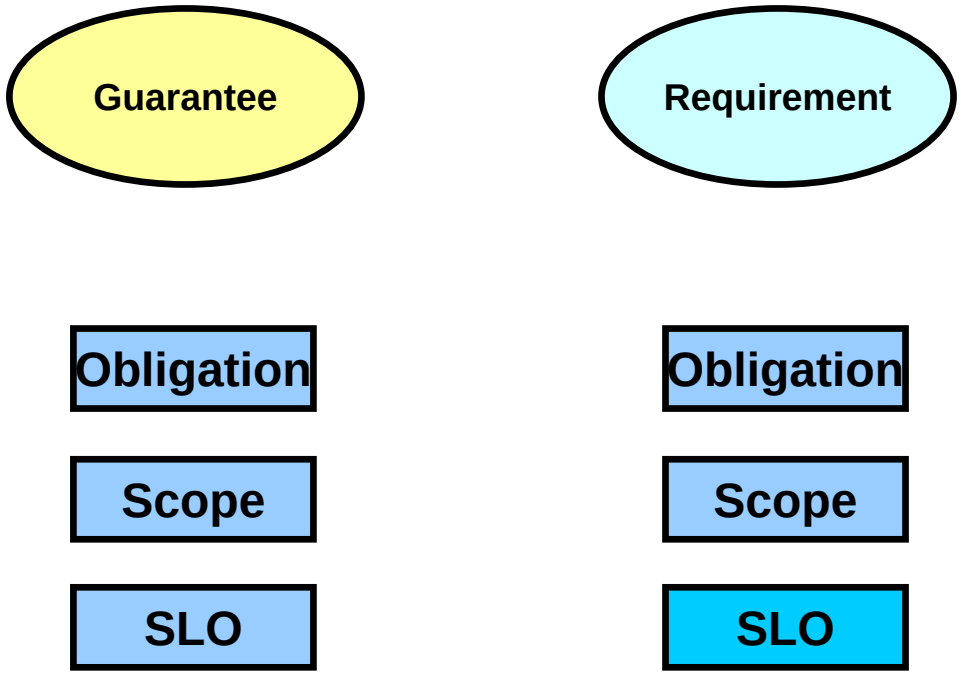
"**requirement(Alt, G)**" returns true if G is a requirement of Alt
"**capability(Alt, G)**" returns true if G is an assurance of Alt
"**scope(G)**" returns the scope of G
"**obligation(G)**" returns the obligated party of G
"**satisfies(Gj, Gi)**" returns true if the SLO of Gj is equivalent to or stronger than the SLO of Gi

An alternative Alt1 is a suitable match for Alt2 if:
("Gi) such that Gi $\in$ Alt1 $\wedge$ requirement(Alt1, Gi) $\wedge$ ($Gj)
such that Gj $\in$ Alt2 $\wedge$ capability(Alt2, Gj) $\wedge$ scope(Gi)
= scope(Gj) $\wedge$ obligation(Gi) = obligation(Gj) $\wedge$ satisfies(Gj, Gi)

WS-Agreement Matching



WS-Agreement Matching



The SLO of the guarantee should meet or exceed the SLO of the requirement

Three Types of Rules

- Domain Specific Rules
  - Used to resolve heterogeneities between SLOs
  - Allow matching of syntactically heterogeneous but semantically same SLOs
- Semantics of Predicates Rules
  - Used for normalizing different kinds of predicates
- User Defined Rules
  - Allow users to customize matchmaking declaratively

Domain Specific Rules – Example 1

Consumer:
Availability is greater than 95%

Provider:
Mean Time to Recover equals 5 minutes
Mean Time between failures equals 15 hours

Domain Specific Rules allow matching of Syntactically heterogeneous but semantically similar rules

Domain Specific Rules – Example 2

- ARL Rule contains a threshold and conversion instructions
- Threshold can apply to all parameters or can be defined separately.

---

when:

Agreement (A) and hasGuarantee (A,G) and hasSLO (G, SLO) and hasExpression(SLO, E) and hasPredicate(E, P) and hasType(P, “PercentageLessThanThreshold”) and hasPercentage(E, percent)

do:

```
if (percent<= THRESHOLD) then
    assert hasType(P, “less”)
else
    assert hasType(P, “greater”)
```

Semantics of Predicates Rules

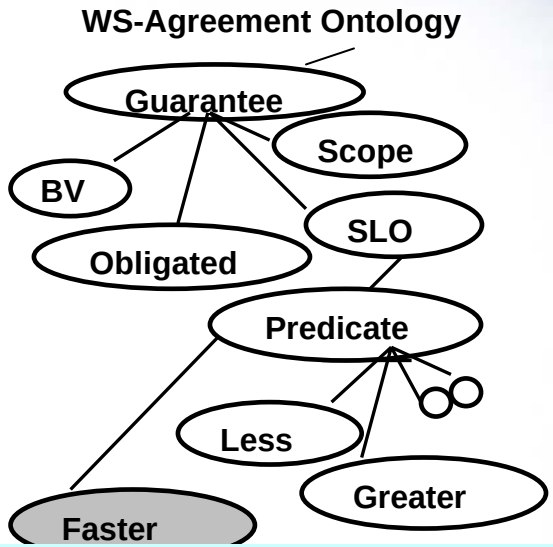
- Normalize matching using two predicates
  - *isStronger* or *isEquivalent*
  - SLO1: x < 5 sec SLO2: x < 10 sec
  - It is asserted that SLO1 *isStronger* SLO2
  - This is done by defining semantics of predicate “less” in terms of *isStronger* and *isEquivalent*

when: Agreement (A1) and hasGuaranteeTerm(A1, G1) and hasSLObjective(G1, SLO1) and hasExpression (SLO1, E1) and hasPredicate(E1, P1) and hasType(P1, “less”) and hasParameter(E1, p1) and hasValue(E1, V1) and Agreement (A2) where A1 != A2 and hasGuaranteeTerm(A2,G2) and hasSLO(G2, SLO2) and hasExpression (SLO2, E2) and hasPredicate(E2, P2) and hasType(P2, “less”) and hasParameter(E2, p2) and p2 == p1 and hasValue(E2, V2)

do: if (V1<V2) assert [E1 *isStronger* E2]
else if (V1>V2) assert [E2 *isStronger* E2]
else assert [E1 *isEquivalent* E2]

Semantics of Predicate Rules

- Semantics of predicates in the ontology are well defined using rules



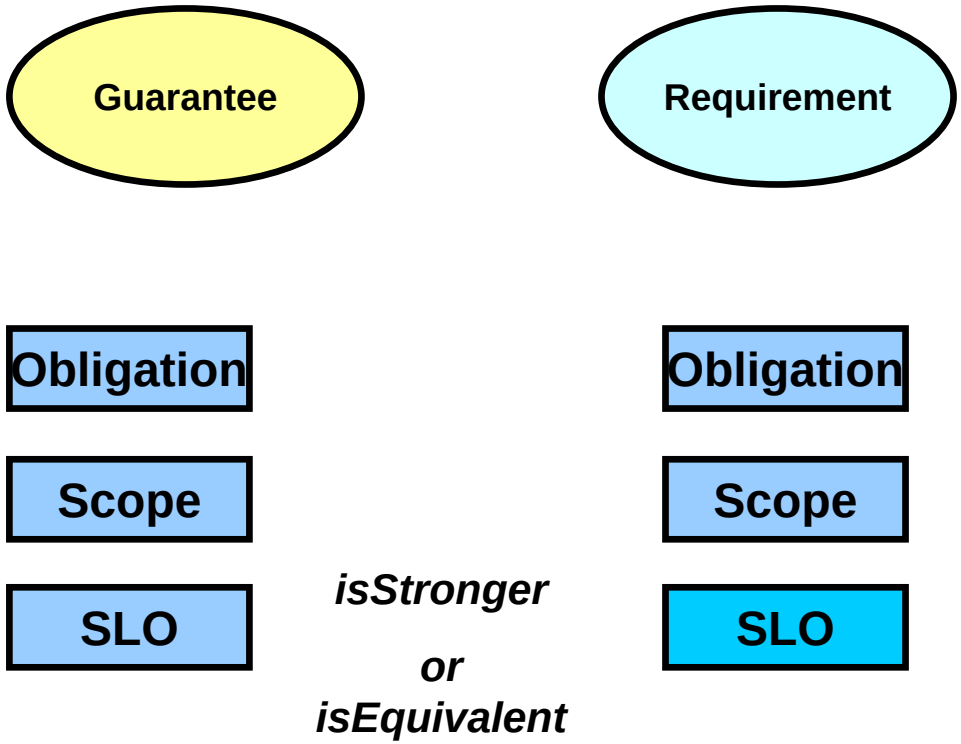
Semantics of Predicate Rules allow precise matching because the semantics of the predicates are well defined using rules

User Preference Rules

- Capture user prefernces using rules
  - Two predicates: *notSuitable* and *isPreferred*
- It is *notSuitable* if the qualifying condition states that the txRate must be less than 1000
  - Any alternative containing a *notSuitable* assertion will be excluded from the result set.

User preference Rules allow declaratively customization of the matcher using rules. No special modules or code is needed

Search Algorithm



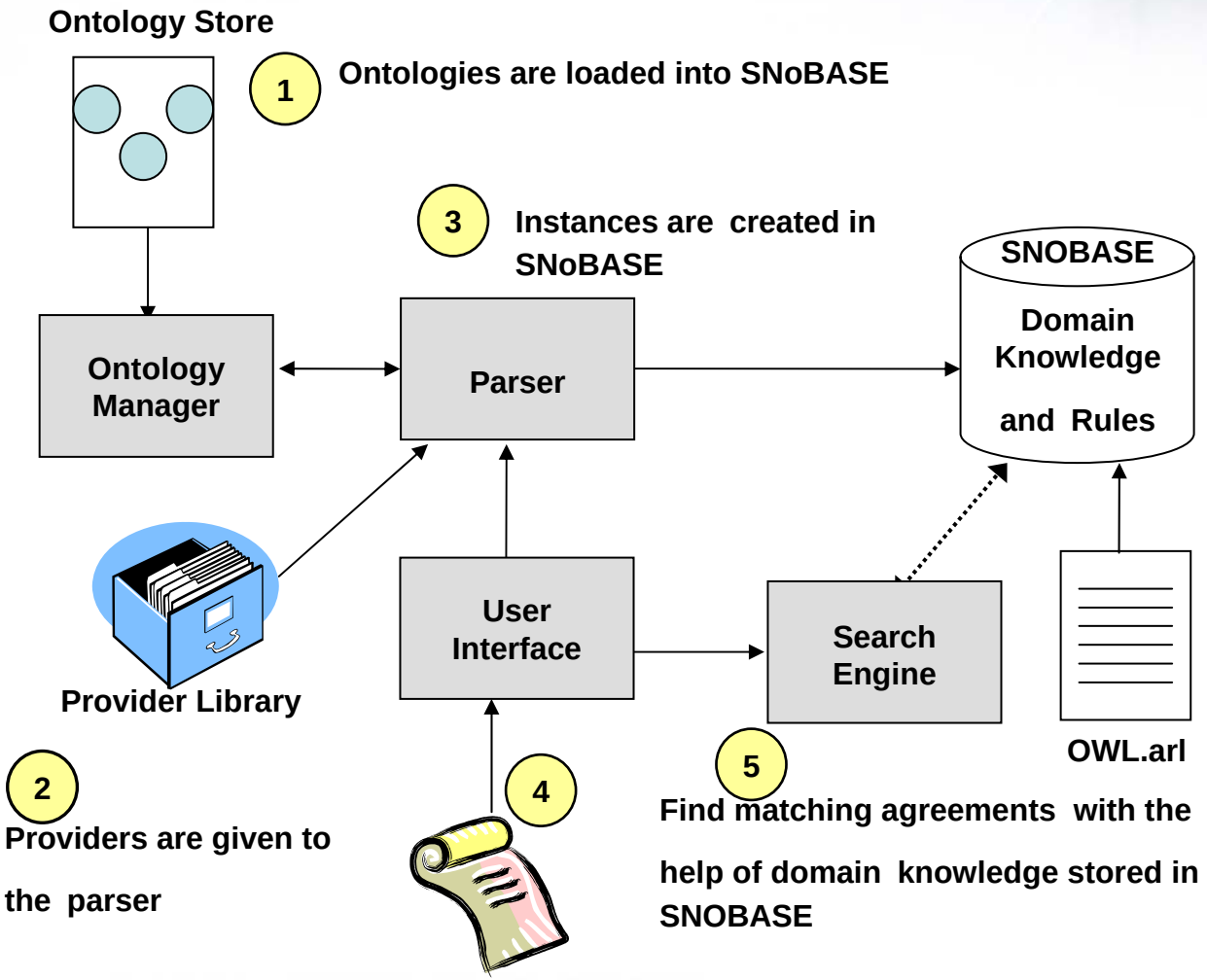
The SLO of the guarantee should meet or exceed the SLO of the requirement

AND There is no *notSuitable* assertion for the alternative

Outline

- Introduction
- WS-Agreement – Details and Ontology
- Matching Algorithm
- Architecture
- Detailed Example
- Conclusion

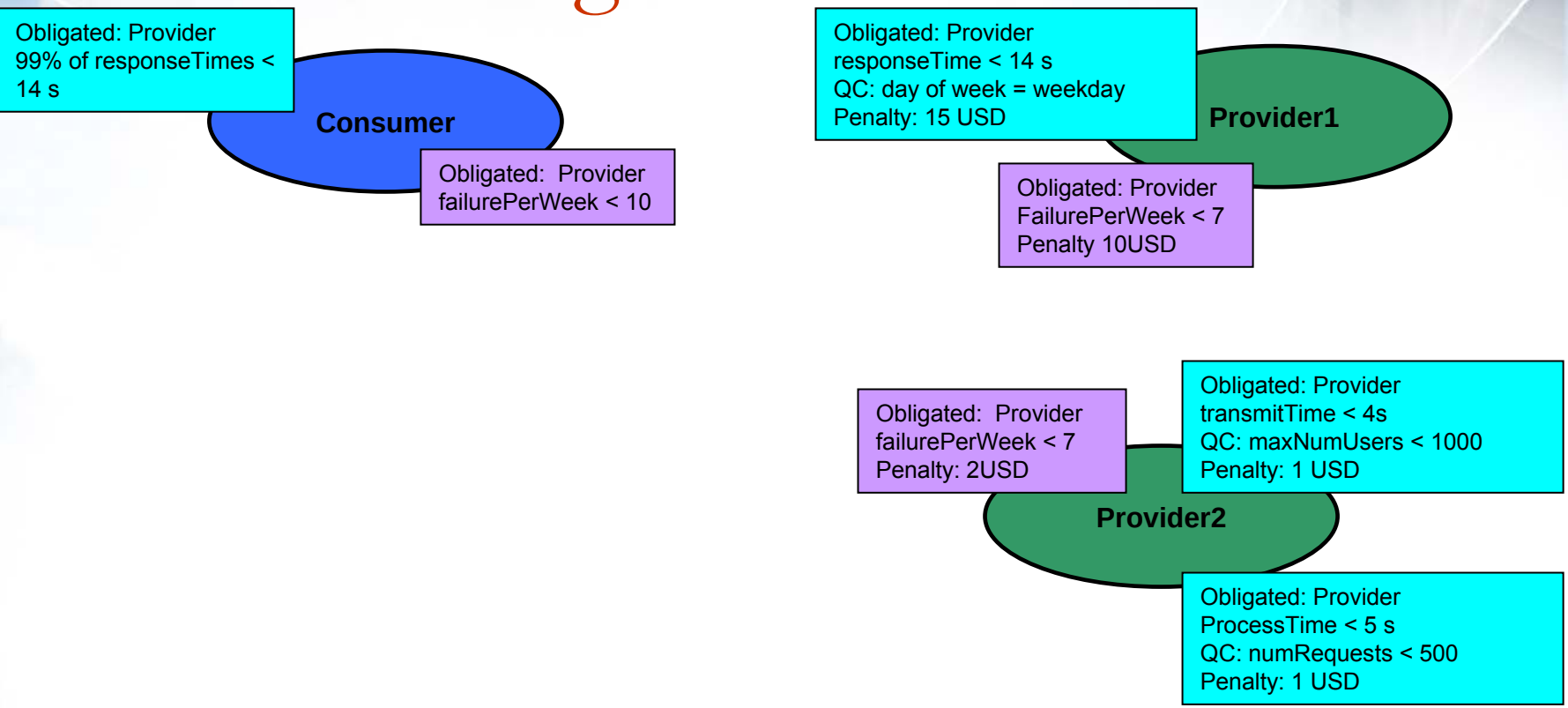
Architecture



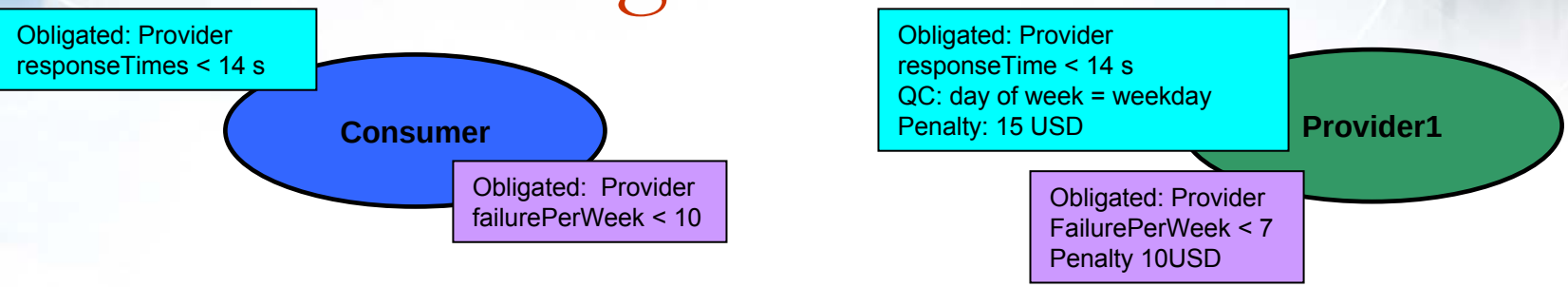
Outline

- Introduction
- WS-Agreement – Details and Ontology
- Matching Algorithm
- Architecture
- Detailed Example
- Conclusion

The Matching Process



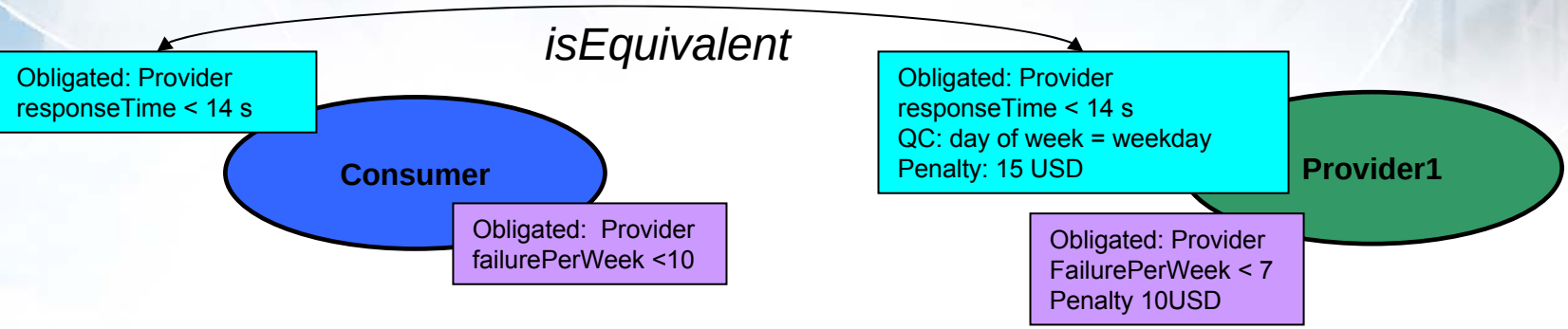
The Matching Process



Knowledge from Domain Specific Rules:

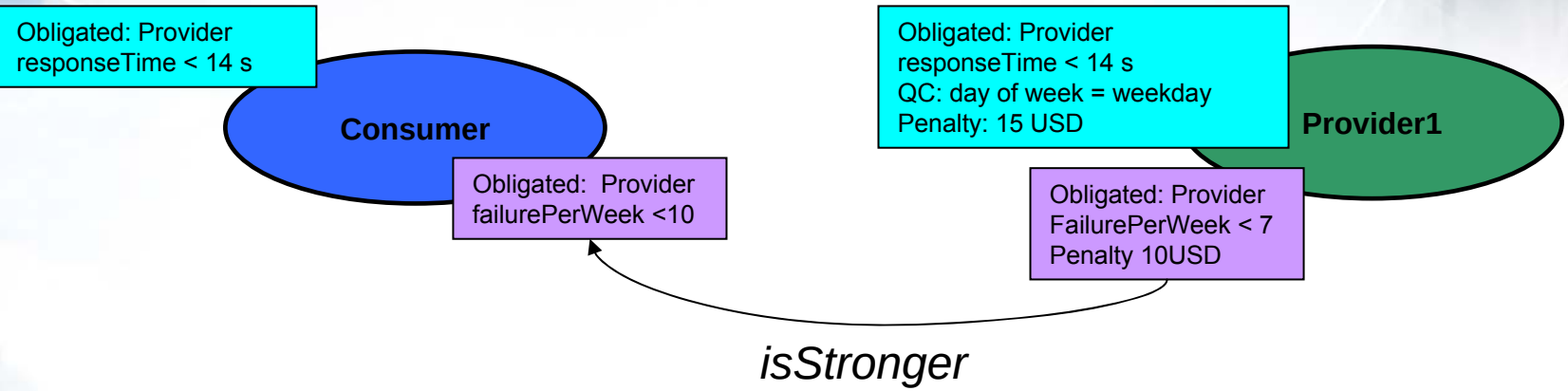
```
if (x >= 96)
  responseTime < y
else
   responseTime > y
```

The Matching Process



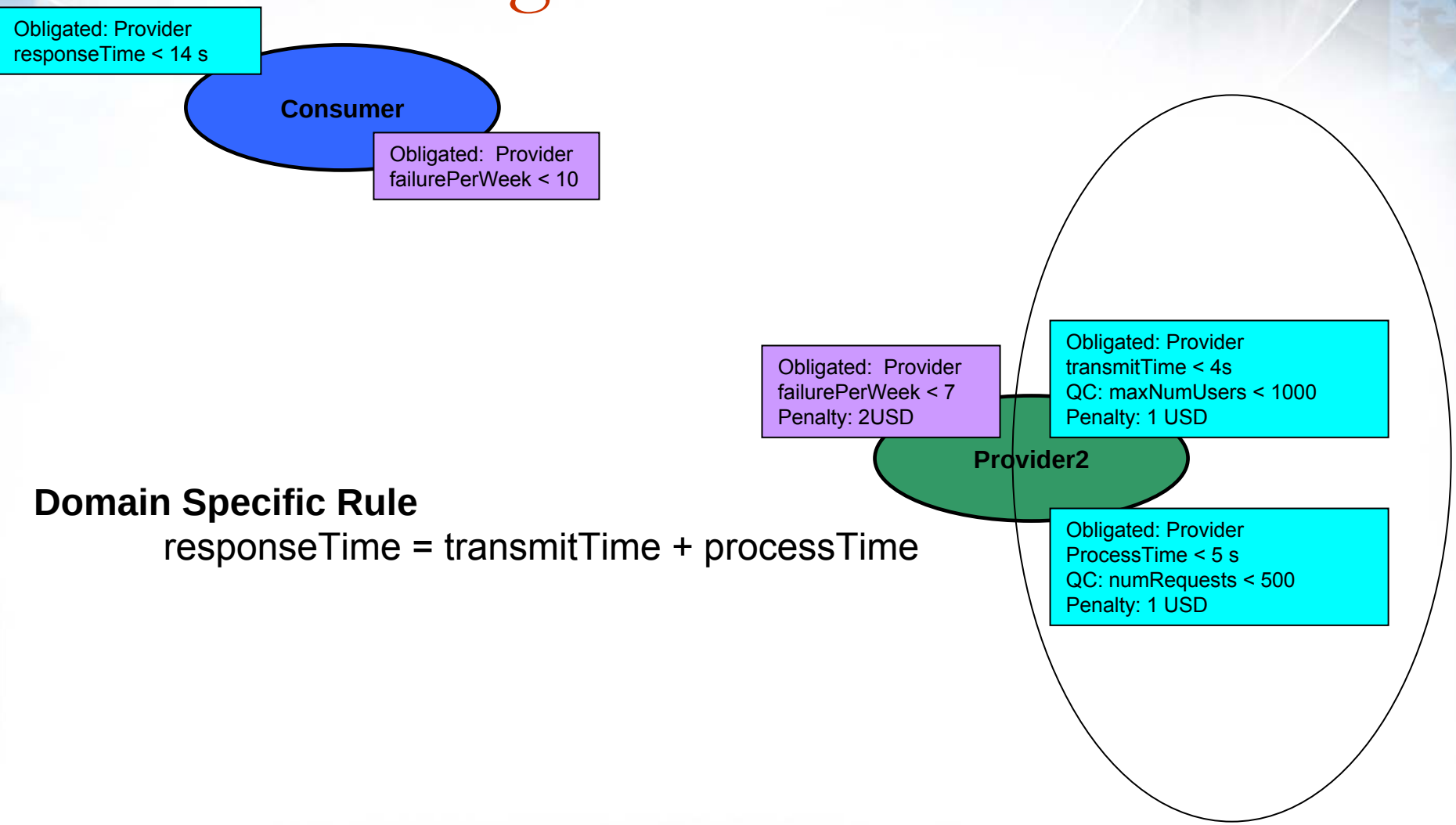
Knowledge from Semantics of Predicate Rules

The Matching Process



Knowledge from Semantics of Predicate Rules

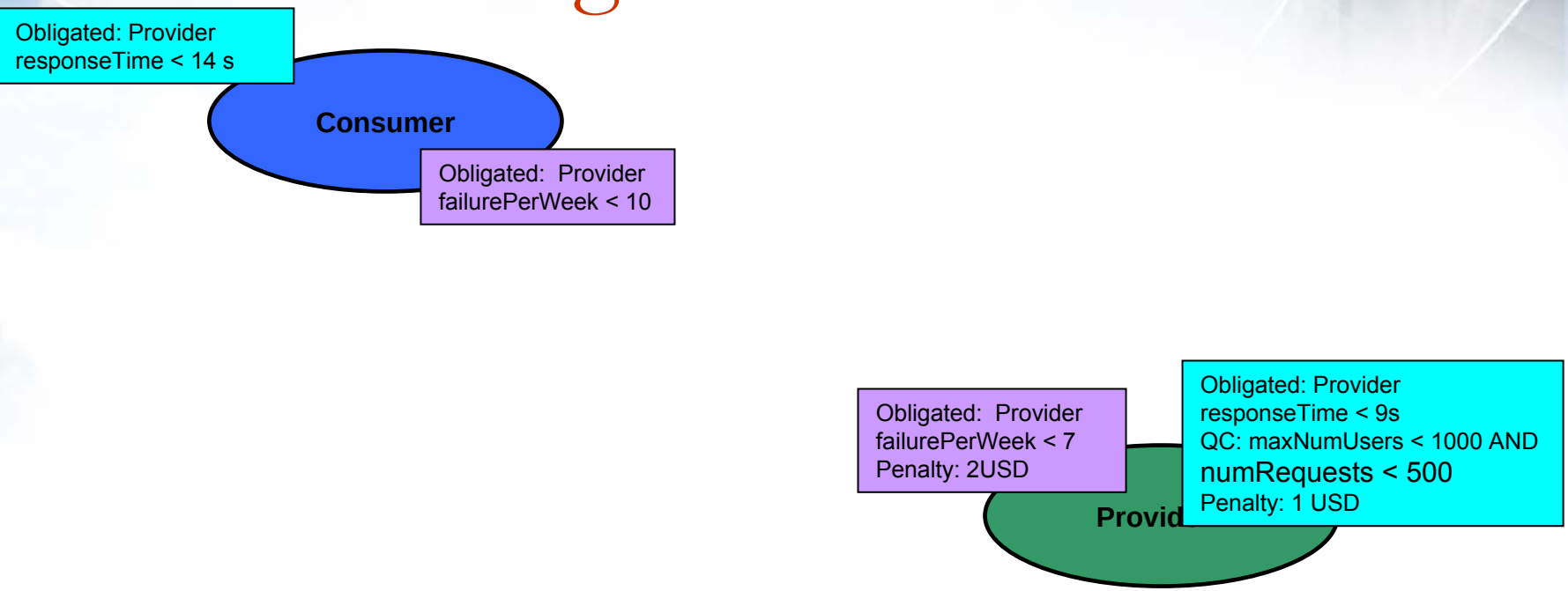
The Matching Process



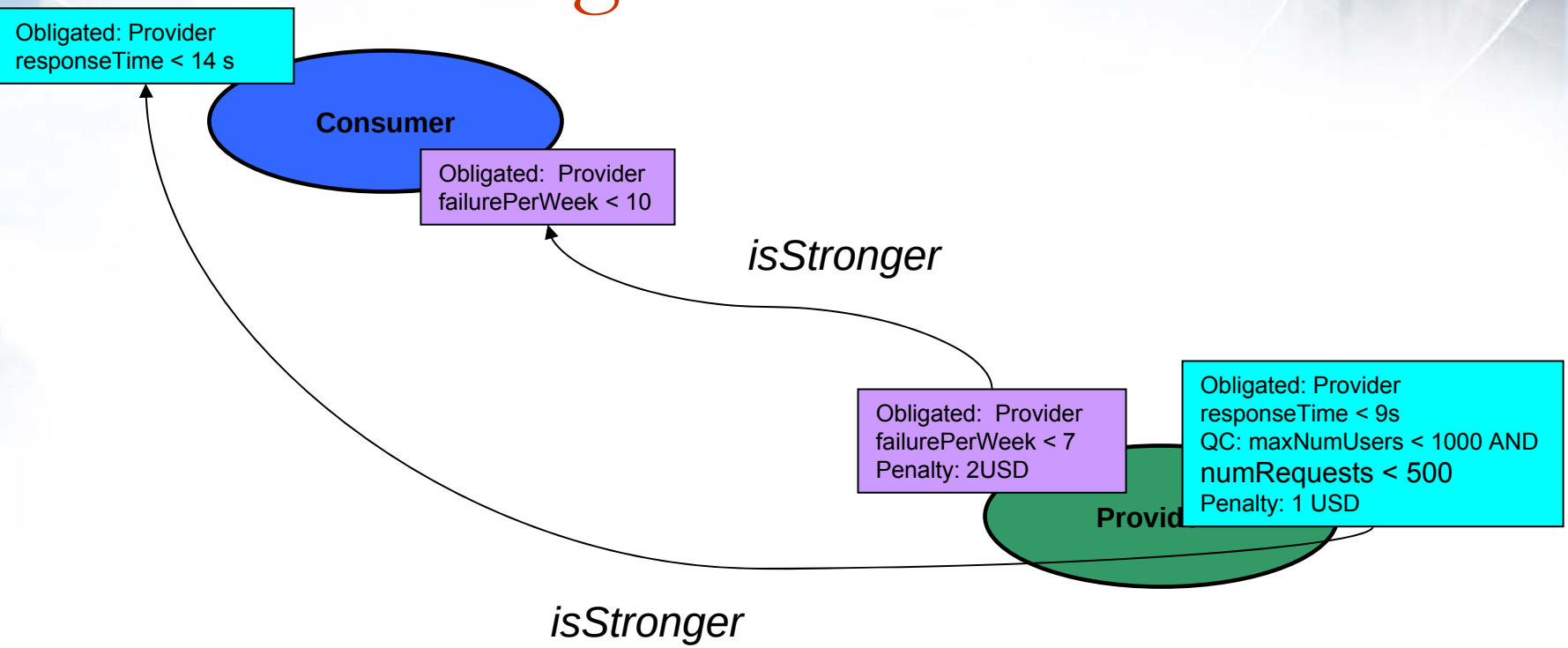
Domain Specific Rule

responseTime = transmitTime + processTime

The Matching Process



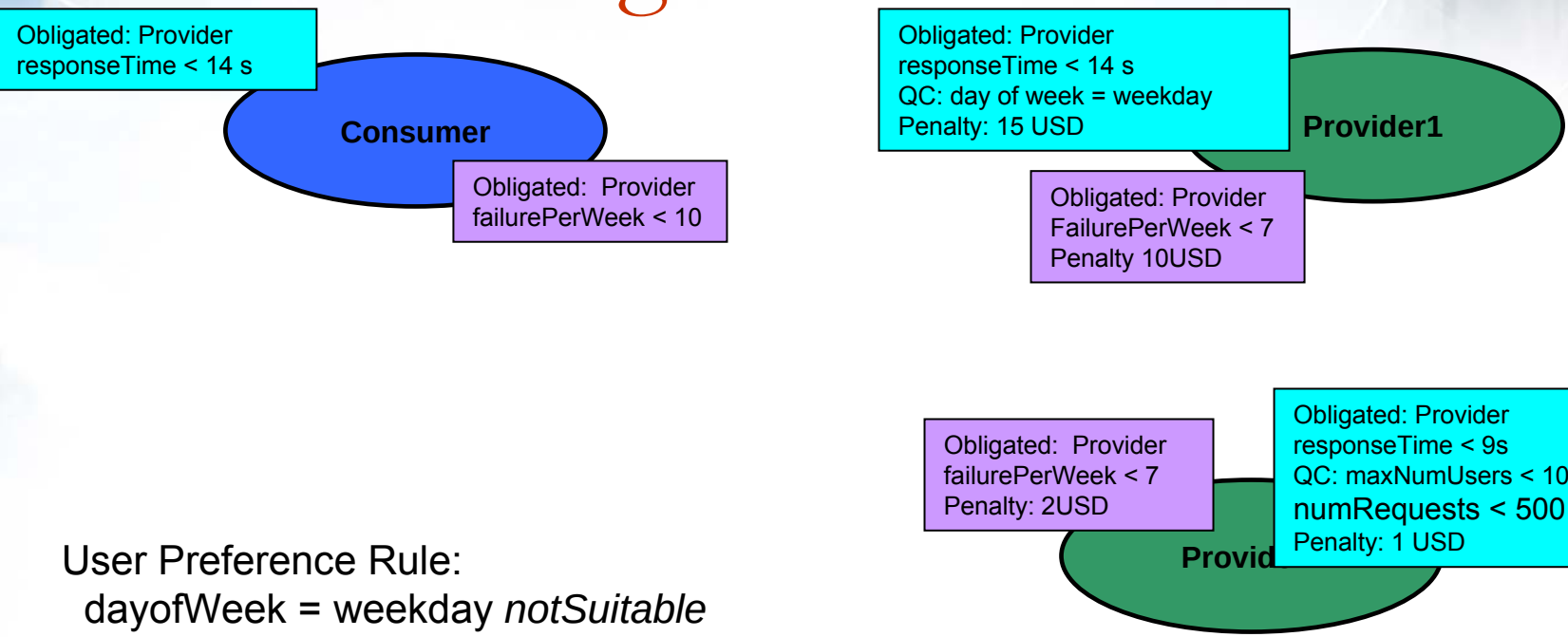
The Matching Process



Steps #5-6: Comparison Rules

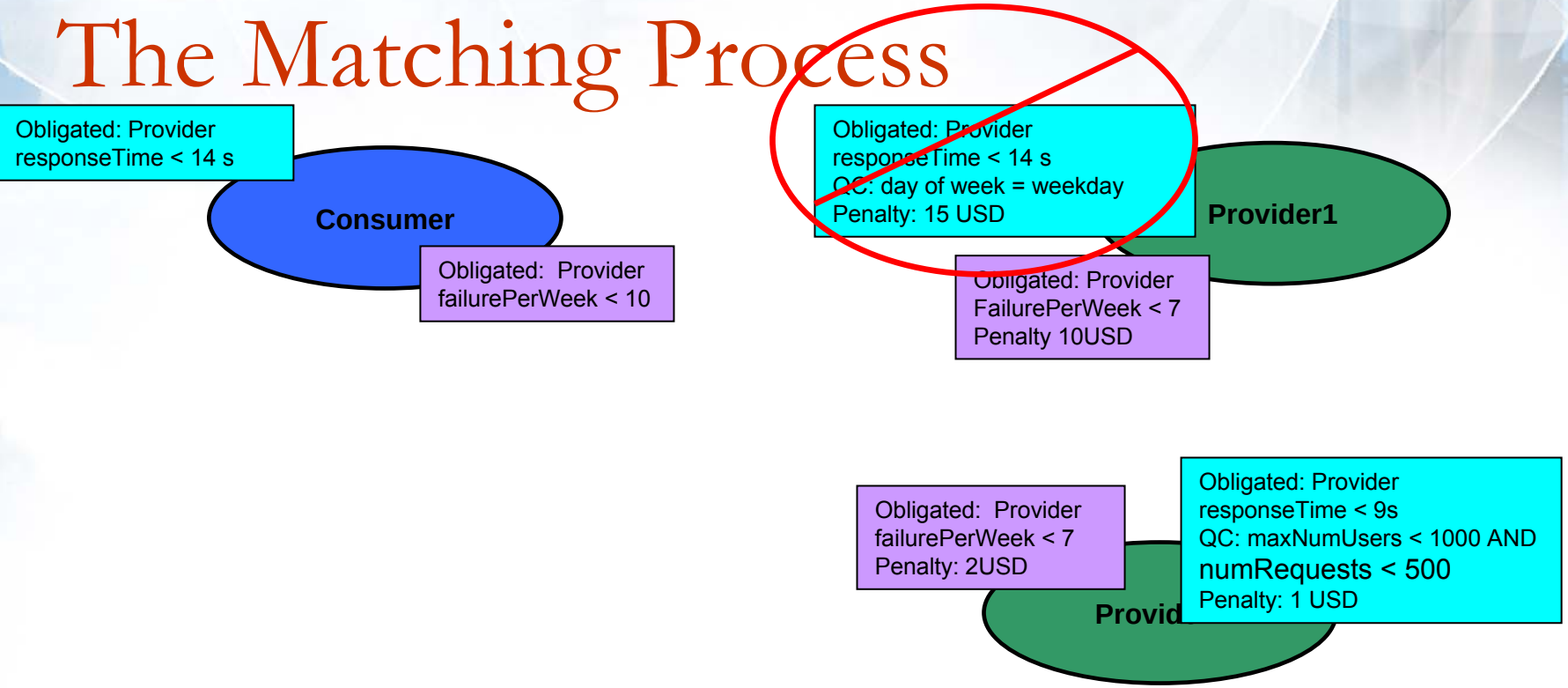
The Matching Process

notSuitable



User Preference Rule:
dayofWeek = weekday *notSuitable*

The Matching Process



Outline

- Introduction
- WS-Agreement – Details and Ontology
- Matching Algorithm
- Architecture
- Detailed Example
- Conclusion

Comparison of Alternatives

Consumer Requirement	Provider Capability	Approach 1: Ontology and Rules	Approach 2: Ontology without Rules	Approach 3: Rules without Ontologies	Approach 4: No Rules and No Ontology
responseTime < 5	responseTime < 4	YES	YES	YES, but only if parameters are named similar syntactically	YES, but only if parameters are named similar syntactically
responseTime < 5	(duration1 + duration2) < 4	YES	NO	YES, but only if the parameters are named similar syntactically to the rule criteria	NO
responseTime < 5	rt < 4	YES	YES	NO	NO
responseTime < 5	networkTime < 2 executionTime < 1	YES	NO	YES, but only if the parameters are named similar syntactically to the rule criteria	NO

Conclusion

- ✓ Designed and Implemented a tool to automate the matching of WS-Agreements.
- ✓ Semantic approach combined with rules yields the most accurate and effective matches which are tailored to user preferences.
  - ✓ Ability to match syntactically heterogeneous but semantically similar assertions
- ✓ Categories of rules allow for customizable matching process independent of code.

Thank You!

Google: METEOR-S -> SWAPS

or

Yahoo!: WS Agreement

References

[Aiello et al] What's in an Agreement? An Analysis and an Extension of WS-Agreement, Proc. 3rd *ICSOC, 2005*

[Andrieux et al] *WebServices Agreement Specification (WS-Agreement).* June 29th 2005

[Bigus et al] ABLE: A toolkit for building multiagent autonomic systems, IBM Systems Journal, 41 (3), 2002

[Chaudhary et al] Architecture of Sensor based Agricultural Information System for Effective Planning of Farm Activities. IEEE SCC 2004: 93-100

[Eaton et al] Contract Farming Partnerships for Growth FAO Agricultural Services Bulletin 145

[Kagal et al] Authorization and Privacy for Semantic Web Services, AAAI Spring Symposium on SW S, 2004

[Kagal et al] Declarative Policies for Describing Web Service Capabilities and Constraints, Proceedings of W3C Workshop on Constraints and Capabilities for Web Services, 2005

[Lee et al] Snobase: A Semantic Network-based Ontology Ontology Management http://alphaWorks.ibm.com/tech/Snobase 2003

[Li et al] Design and Application of Rule Based Access Control Policies. Proc of the Semantic Web and Policy Workshop, 2005, Galway, IR.

References

[Ludwig et al] Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements. Proc 2nd *ICSOC,*, New York, 2004.

[Maxemilien et al] A Framework and Ontology for Dynamic Web Services Selection. IEEE Internet Computing

OWL-S, http://www.daml.org/services/owl-s/

[Pan et al] OWL Time http://www.isi.edu/~ pan/damltime/time-entry.owl

[Parsia et al] Expressing WS-Policies in OWL. Policy Management for the Web Wkshp, May 2005

[Paschke et al] A Logic Based SLA Management Framework. Proc. of the Semantic Web and Policy Workshop, November, 2005.

References

- Sorathia, V., Laliwala, Z., and Chaudhary, S. Towards Agricultural Marketing Reforms: Web Services Orchestration Approach, IEEE SCC 2005.
- Uszok, A., Bradshaw, J.M., Jeffers, R., Johnson, M., Tate, A., Dalton, J., Aitken, S. Policy and Contract Management for Semantic Web Services, Proc. of the AAAI Spring Symposium on Semantic Web Services, 2004
- Verma, K., Akkiraju, R., Goodwin, R. Semantic Matching of Web Service Policies, SDWP Workshop, 2005.,
- The Web Service Policy Framework, http://www-106.ibm.com/developerworkds/library/ws-polfram
- Wohlstadter, E., Tai, S., Mikalsen, T., Rouvello, I., Devanbu, P. GlueQoS: Middleware to Sweeten Quality-of-Service Policy Interactions, The Proc ICSE 2004, pp. 189-199
- The WSLA Specification, http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf
- WSDL-S, http://www.w3.org/Submission/WSDL-S/
- W. Yang, H. Ludwig, A. Dan: Compatibility Analysis of WSLA Service Level Objectives. Workshop on the Design of Self-Managing Systems. Supplemental, 2003

Backup Slides

Robodemo

Sample Agreement

```
<wsag:GuaranteeTerm wsag:Name="TransferTimeJob1">
        <wsag:Obligated>ServiceProvider</wsag:Obligated>
        <wsag:ServiceScope>
                <wsag:ServiceName>rosettaNet:getInvoice</wsag:ServiceName>
        </wsag:ServiceScope>
        <wsag:ServiceLevelObjective>
                <wsag:predicate type="less">
                <wsag:parameter>qos:ResponseTime</wsag:parameter>
                <wsag:value>5</wsag:value>
                <wsag:unit>time:seconds</wsag:unit>
                </wsag:predicate>
        </wsag:ServiceLevelObjective>
        <wsag:BusinessValueList>
                <wsag:Penalty>
                        <wsag:AssessmentInterval>
                        <wsag:Count>1</wsag:Count>
                        </wsag:AssessmentInterval>
                        <wsag:ValueExpression>5</wsag:ValueExpression>
                        <wsag:ValueUnit>USD</wsag:ValueUnit>
                </wsag:Penalty>
        </wsag:BusinessValueList>
</wsag:GuaranteeTerm>
```

Categories of Rules

1. Conversions for Heterogeneous SLOs ie: PercentageLessThanThreshold, etc.

when: Agreement (A) and hasGuarantee (A,G) and hasSLO (G, SLO) and hasExpression(SLO, E) and hasPredicate(E, P) and hasType(P, “PercentageLessThanThreshold”) and hasPercentage(E, percent)

do: if (percent<=x) then assert hasType(P, “less”) else assert hasType(P, “greater”)

Categories of Rules

2. Semantics of Predicates

when: Agreement (A1) and hasGuaranteeTerm(A1, G1) and hasSLObjective(G1, SLO1) and hasExpression (SLO1, E1) and hasPredicate(E1, P1) and hasType(P1, “less”) and hasParameter(E1, p1) and hasValue(E1, V1) and Agreement (A2) where A1 != A2 and hasGuaranteeTerm(A2,G2) and hasSLO(G2, SLO2) and hasExpression (SLO2, E2) and hasPredicate(E2, P2) and hasType(P2, “less”) and hasParameter(E2, p2) and p2 == p1 and hasValue(E2, V2)

do: if (V1<V2) assert [E1 *isStronger* E2]

else if (V1>V2) assert [E2 *isStronger* E2]

else assert [E1 *isEquivalent* E2]

Categories of Rules

3. Domain Specific Rules

MTBF is the Mean Time Between Failures

MTTR is the Mean Time To Recover

Availability = MTBF/(MTBF + MTTR)

Guarantee1: SLO: qos:MTBF=150 time:minutes, Qualifying Condition: numRequests<1000, Penalty: 5 USD, Importance 8

Guarantee2: SLO: qos:MTTR<5 time:minutes, Qualifying Condition: numUsers<500, Penalty: 3 USD, Importance 4

Guarantee3: SLO: qos:Availability=96.8, Qualifying Condition: numUsers<500 AND numRequests<1000, Penalty: 5 USD, Importance: 6

Categories of Rules

4. User Preference Rules

when: Agreement (A) and hasGuarantee (A, G1) and hasQualifyingCondition(G1, QC1) which hasExpression(QC1, E1) and hasParameter(E1, “time:dayOfWeek”) and hasValue(E1, “time:weekday”)

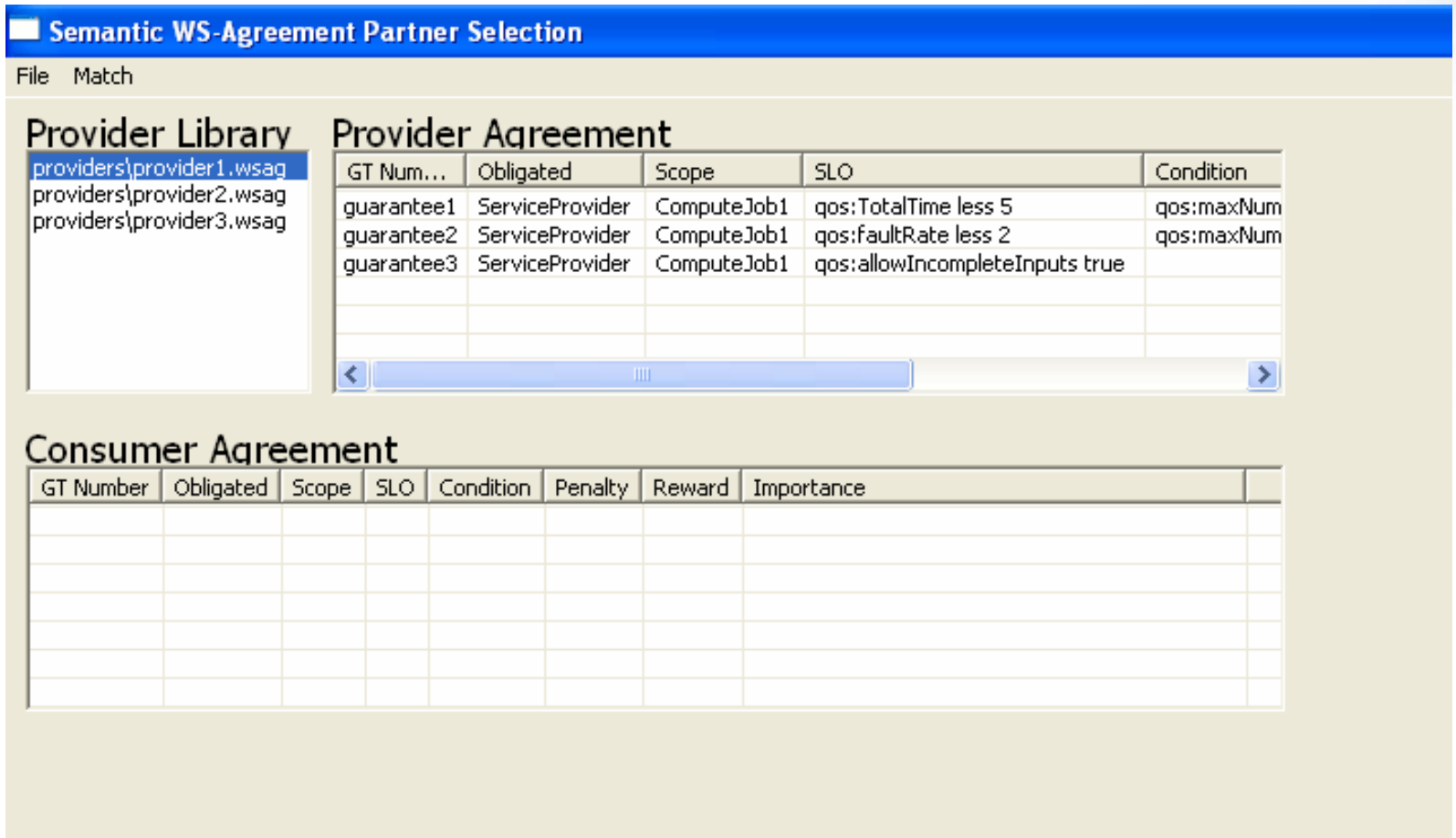
do: assert Guarantee *notSuitable G1*

Categories of Rules

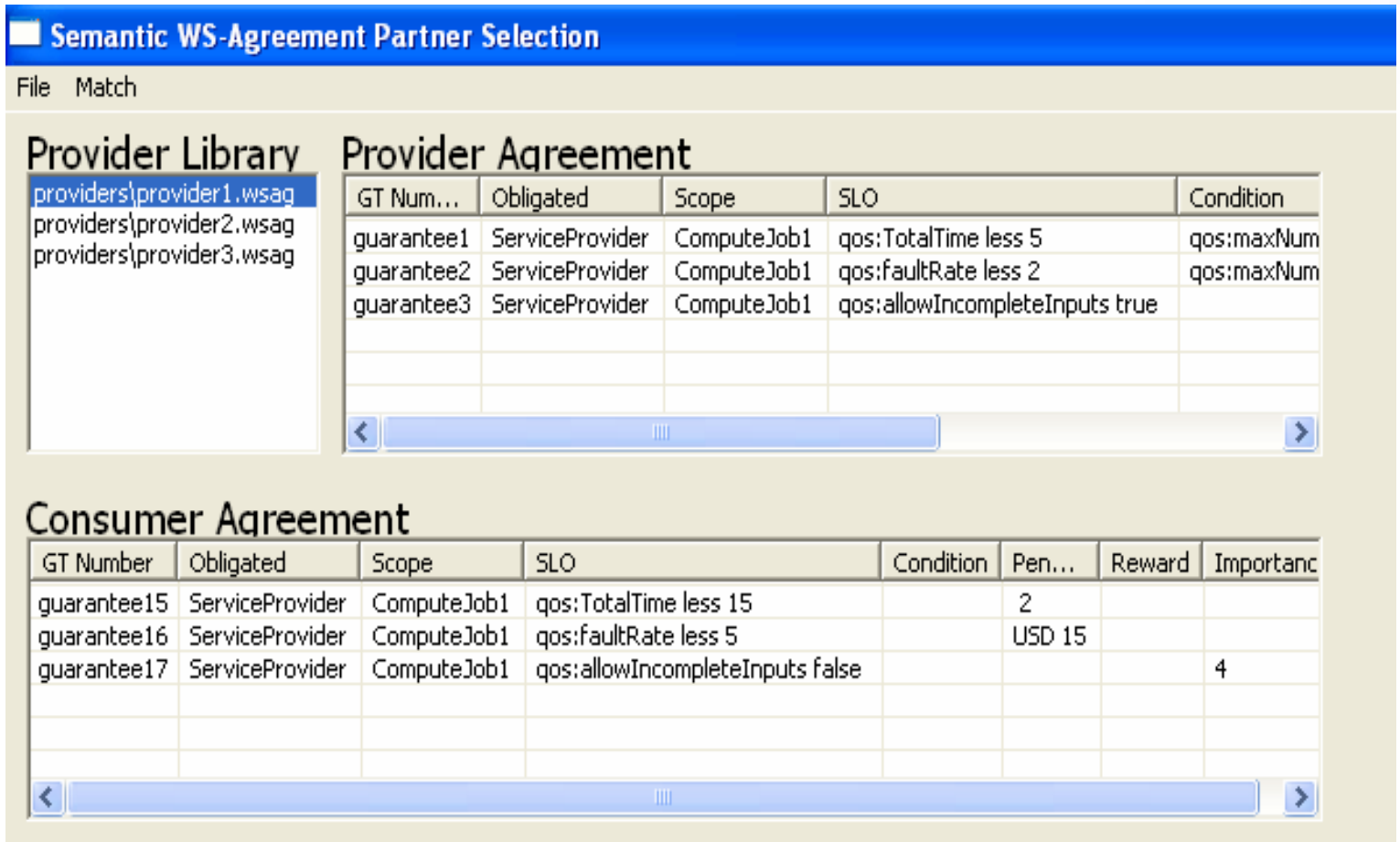
when: Agreement (A) and hasGuarantee (A, G1) and hasSLO (G1, SLO1) and hasQualifyingCondition(G1, QC1) and hasPenalty(G1, P1) and hasImportance(G1, I1) and hasExpression (SLO1, E1) and hasParameter(E1, "*qos:MTBF"*) and hasValue(E1, X) and hasGuarantee (A, G2) and hasSLO (G2, SLO2) and hasQualifyingCondition(G2, QC2) and hasPenalty(G2, P2) and hasImportance(G2, I2) and hasExpression (SLO2, E2) and hasParameter(E2, *"qos:MTTR"*) and hasValue(E2, Y)

do: hasGuarantee (A,G3) and hasSLO(G3, SLO3) and hasExpression(SLO3, E3) and hasParameter(E3, "*qos:Availability"*) and hasVaule(E3, X+Y) and hasPenalty (G3, max(P1, P2)) and hasImportance(avg(I1,I2))

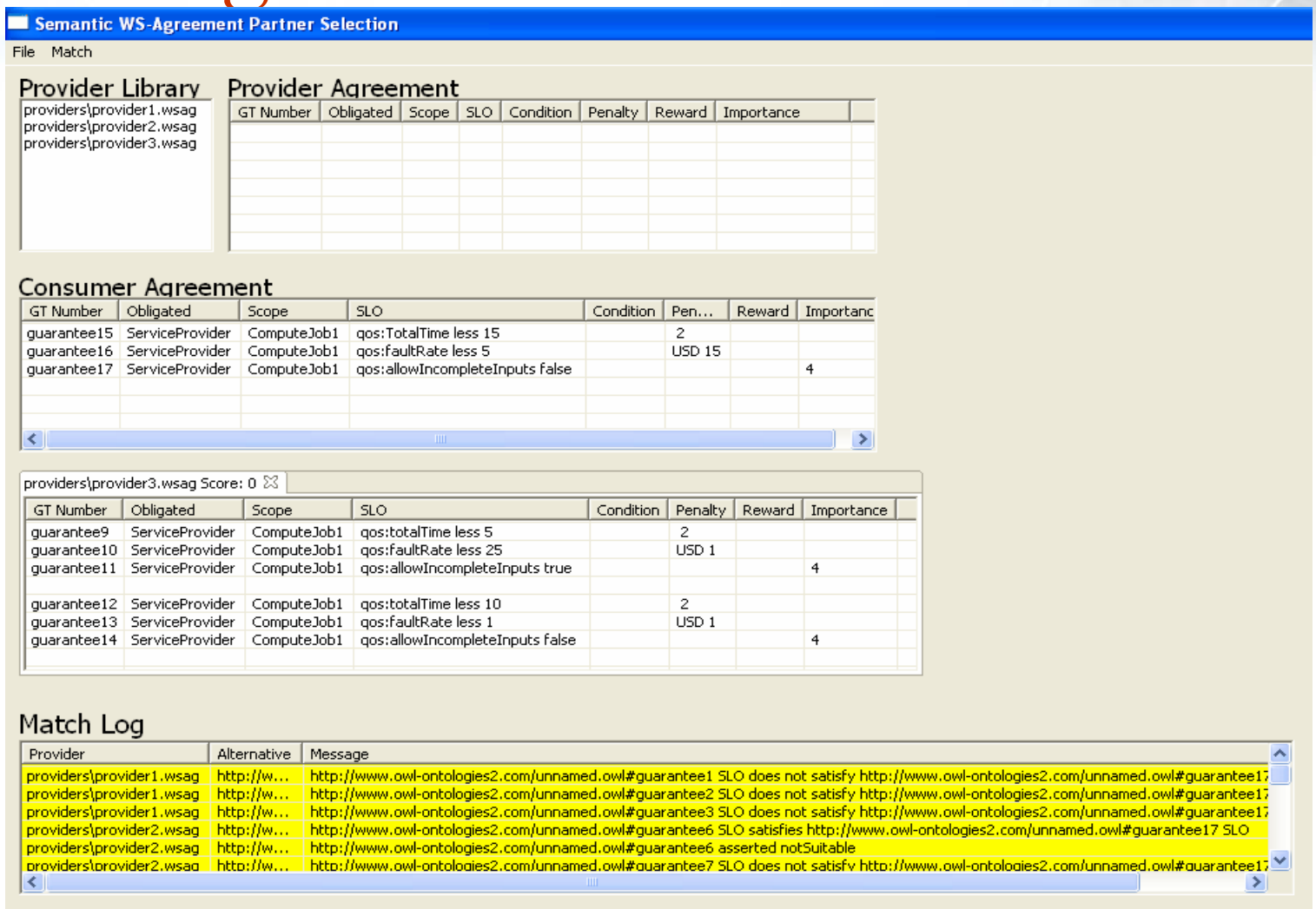
Provider Library



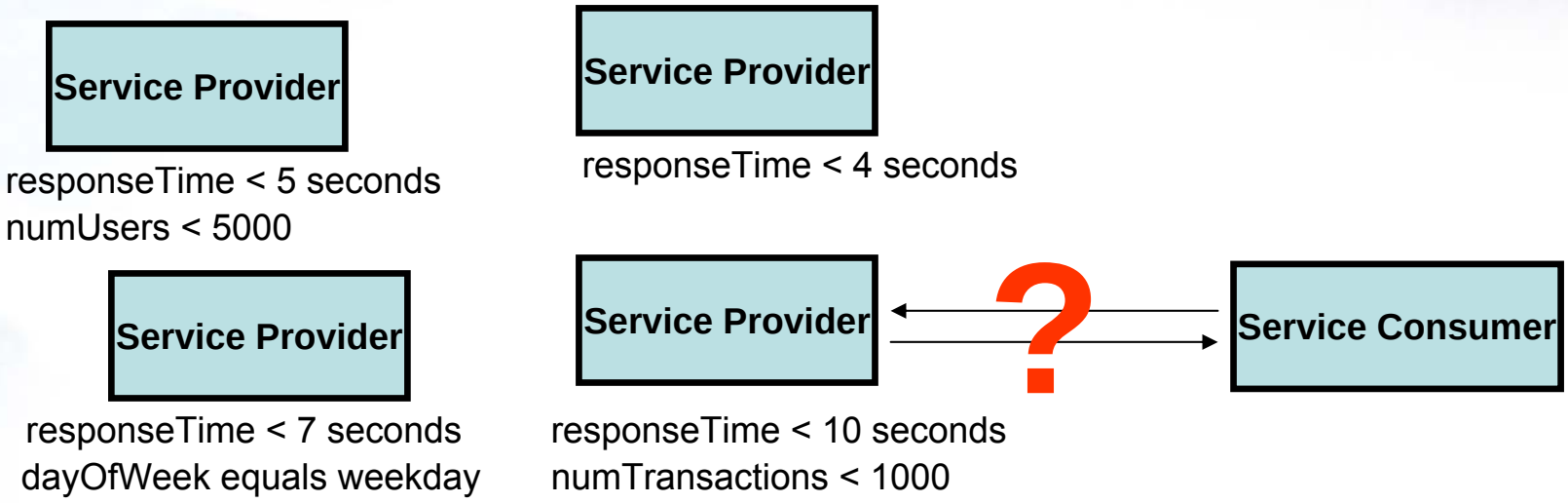
Selecting a Consumer



Matching


Semantic WS-Agreement Partner Selection
File
Match
Provider Library
providers\provider1.wsag
providers\provider2.wsag
providers\provider3.wsag
Provider Agreement
GT Number
Obligated
Scope
SLO
Condition
Penalty
Reward
Importance
Consumer Agreement
GT Number
Obligated
Scope
SLO
Condition
Pen...
Reward
Importanc
guarantee15
ServiceProvider
ComputeJob1
qos:TotalTime less 15
2
guarantee16
ServiceProvider
ComputeJob1
qos:faultRate less 5
USD 15
guarantee17
ServiceProvider
ComputeJob1
qos:allowIncompleteInputs false
4
providers\provider3.wsag Score: 0
GT Number
Obligated
Scope
SLO
Condition
Penalty
Reward
Importance
guarantee9
ServiceProvider
ComputeJob1
qos:totalTime less 5
2
guarantee10
ServiceProvider
ComputeJob1
qos:faultRate less 25
USD 1
guarantee11
ServiceProvider
ComputeJob1
qos:allowIncompleteInputs true
4
guarantee12
ServiceProvider
ComputeJob1
qos:totalTime less 10
2
guarantee13
ServiceProvider
ComputeJob1
qos:faultRate less 1
USD 1
guarantee14
ServiceProvider
ComputeJob1
qos:allowIncompleteInputs false
4
Match Log
Provider
Alternative
Message
providers\provider1.wsag
http://w...
http://www.owl-ontologies2.com/unnamed.owl#guarantee1 SLO does not satisfy http://www.owl-ontologies2.com/unnamed.owl#guarantee17
providers\provider1.wsag
http://w...
http://www.owl-ontologies2.com/unnamed.owl#guarantee2 SLO does not satisfy http://www.owl-ontologies2.com/unnamed.owl#guarantee17
providers\provider1.wsag
http://w...
http://www.owl-ontologies2.com/unnamed.owl#guarantee3 SLO does not satisfy http://www.owl-ontologies2.com/unnamed.owl#guarantee17
providers\provider2.wsag
http://w...
http://www.owl-ontologies2.com/unnamed.owl#guarantee6 SLO satisfies http://www.owl-ontologies2.com/unnamed.owl#guarantee17 SLO
providers\provider2.wsag
http://w...
http://www.owl-ontologies2.com/unnamed.owl#guarantee6 asserted notSuitable
providers\provider2.wsag
http://w...
http://www.owl-ontologies2.com/unnamed.owl#guarantee7 SLO does not satisfy http://www.owl-ontologies2.com/unnamed.owl#guarantee17

Service Partner Selection



- **Consumers benefit from obtaining guarantees regarding the required service.**
- **These guarantees usually pertain to the quality of service (QoS).**
- **Requirements and Capabilities of a service can be expressed using standards such as WS-Policy, WSLA, WS-Agreement.**
- **Finding the best provider for a consumer is tedious, time consuming, and error prone.**

WS-Agreement

Allows users to specify requirements and capabilities in the following domains/categories:

1. **Scope:** Service element to which a guarantee applies. A guarantee might only apply to one operation of a Web service at a particular end point

3. **Service Level Objectives**
   - responseTime < 2 seconds

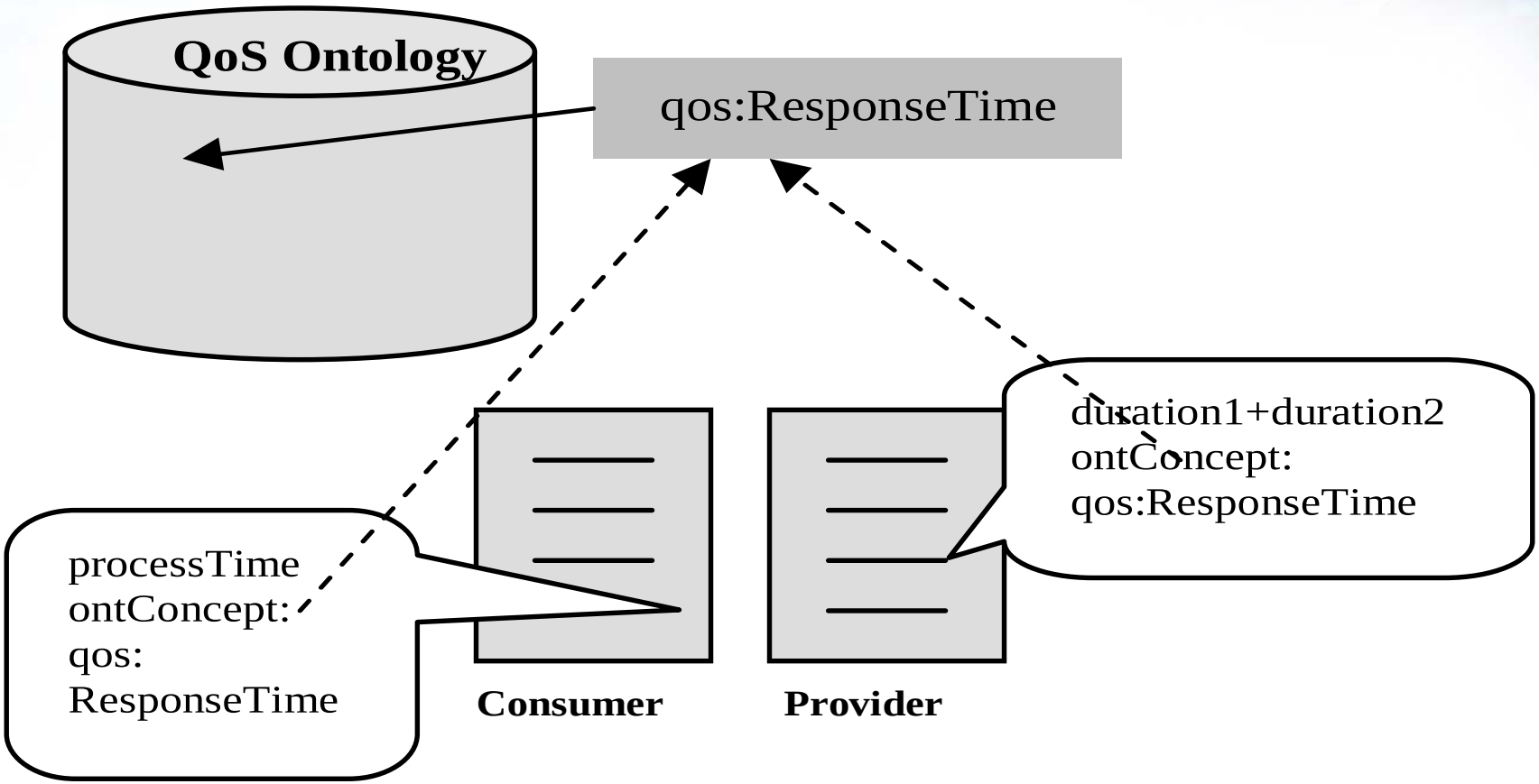
4. **Qualifying Conditions**
   - numRequests < 100

5. **Business Values:** expresses importance, confidence, penalty, and reward.
   - Penalty 5 USD

Semantic WS-Agreements

	WS-Agreement Schema	SWAPS Schema
SLO	<ServiceLevelObjective> duration1+duration2 < 5 s </ServiceLevelObjective >	<ServiceLevelObjective> <Expression> <Predicate type=”less”> <Parameter>duration1+duration2 < /Parameter> <OntConcept>qos:responseTime </OntConcept> <Value>5</Value> <Unit>time:seconds</Unit> </Predicate></Expression> </ServiceLevelObjective>
QC	<QualifyingCondition> day of week is a weekday </QualifyingCondition>	<QualifyingCondition> <Expression> <Predicate type=”equals”> <Parameter>dayOfWeek </Parameter> <OntConcept>time:dayOfWeek </OntConcept> <Value>time:weekday</Value> </Predicate></Expression> </QualifyingCondition>

Benefits of the OntConcept Annotation



Related Work

- WS-Agreement
- WSLA: Compatibility Analysis, Heiko Ludwig and Asit Dan
- [Wohlstadter et al., 2004] GlueQoS (Syntactic Policy Matching)
- [A. Paschke, et al.] A Logic Based SLA Management Framework (Policies with Rules)
- [Uszok et al., 2004] Policy and Contract Management for Semantic Web Services (Policies with Semantics)

Related Work

[K. Verma, et al.] Semantic Matching of Web Service Policies
(Semantic Policy Matching with Rules)

Example:

Provider: BusinessLevel of requestor must be Enterprise

Requestor has a Dun & Bradstreet rating of A3.

Rule: If a company has Dun &Bradstreet rating A3 then it is enterprise level

- What they don't do:
  - Reason over qualifying conditions or business values
  - Allow the user to specify tradeoffs and preferences

Use Case: The value of agreement in farming

Problem: Farmers cultivating goods without assurance that there will be a buyer.

- Wasted goods

Solution: Contract Farming

- Farmer provides an agricultural commodity of a certain type, at a time and a price, and in the quantity required by a known and committed buyer

Advantages:

- Farmer have guaranteed buyers, quantities, prices.
- Buyers have more consistent quality than purchasing in the open market

Why Agreements?

- Agreements cover
  - -the responsibilities and obligations of each party
  - -the manner in which the agreement can be enforced
  - -the remedies to be taken if the contract breaks down.

- Each merchant will have prices, stipulations, incentives.

- How does a farmer chose the best merchant?

Categories of Agreement

- Crop Delivery Arrangements
- Pricing Arrangements
- Cultivation Practices
- Quality and Quantity of Goods
- Payment Procedures
- Insurance Arrangements

Sample Contracts

Farmer Contract

Objective1: *Moisture is less inclusive 14%*
Penalty: discount $x each

Objective2: *splits is less inclusive 20%*
Penalty: splits of 5% or more, discount $y each

Objective3: *test weight is greater than inclusive 54 lbs*

Objective4: *oil content varies between x% and y%*
Conditions: variety of seed selected
planting date is between x and y dates
contaminating pollination by non-high oil corn variety

Buyer Contract

Objective1: *guarantees compensation of grower to be*
*(deliveryLocationPrice – discountPenalties) * netBushels*
Condition: market conditions may make deliveryLocationPrice higher or lower.

Objective2: *establishes delivery date.*

Objective3: *draws a sample oil content from each load.*