FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

This lecture contains materials under submission. For personal use only.

On Specifying and Providing Data Concerns for Adaptive Service Composition and Execution

Hong-Linh Truong

Distributed Systems Group, Vienna University of Technology

truong@infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at/Staff/truong

Joint work with: Schahram Dustdar, Marco Comerio, Atif Manzoor, Andrea Maurino, Michael Mrissa, Flavio De Paoli

Overview

- 9:30-11:30
  - Motivation
  - Data Concerns
  - Data Concerns Evaluation and Publishing
- 13:00-14:30
  - Service Contract Compatibility Evaluation
  - Service Information Overloading
  - Conclusion and Future Work

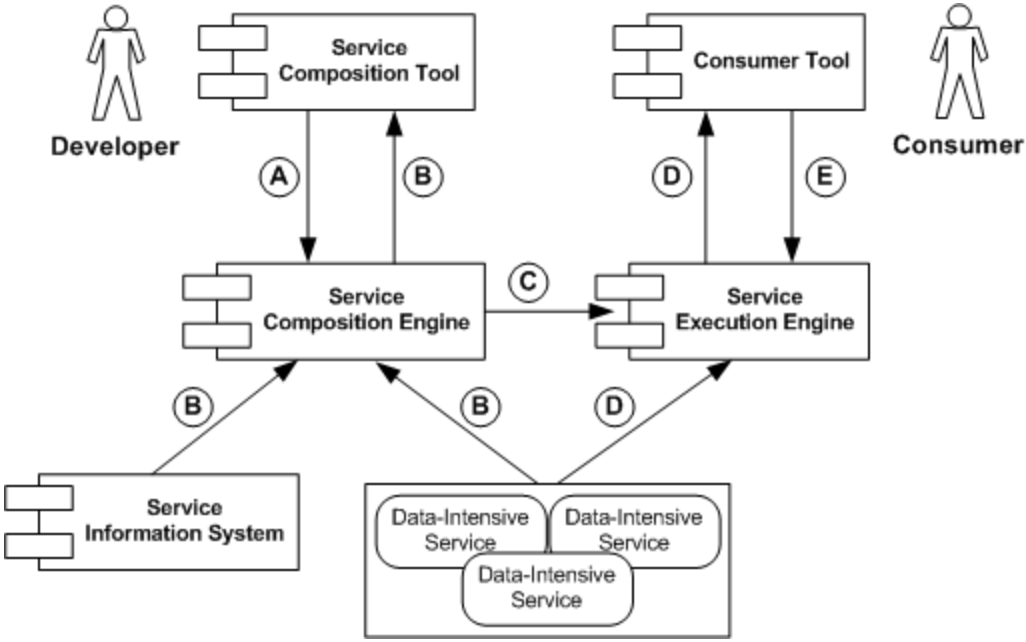
Motivation

Service Composition and Execution

- The ultimate goal:

  provide *relevant* results in an *acceptable quality*
- Solutions
  - Adaptive/context-aware service composition and execution
- But several challenging issues

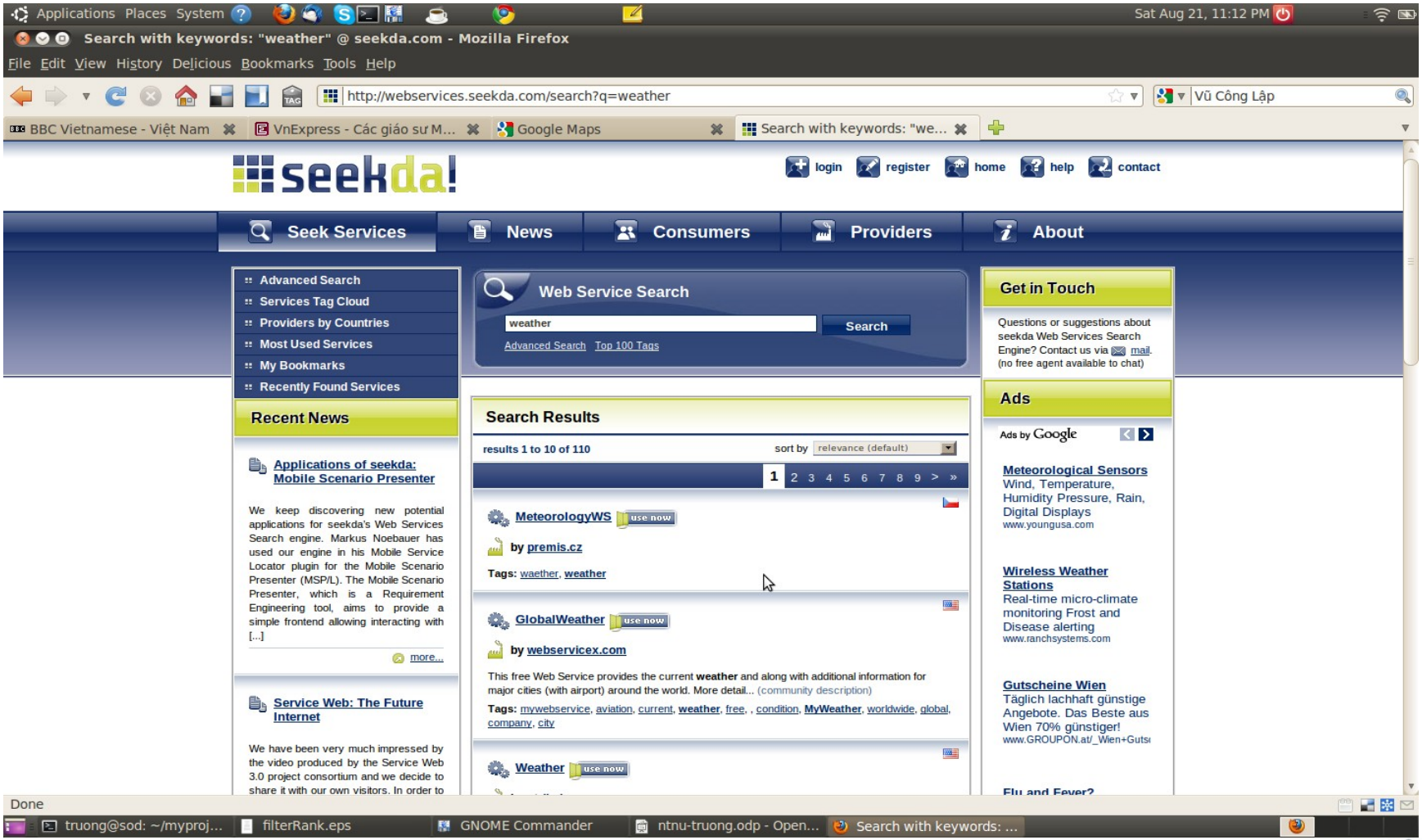
Basic information flows in service composition and execution



Examples of problems

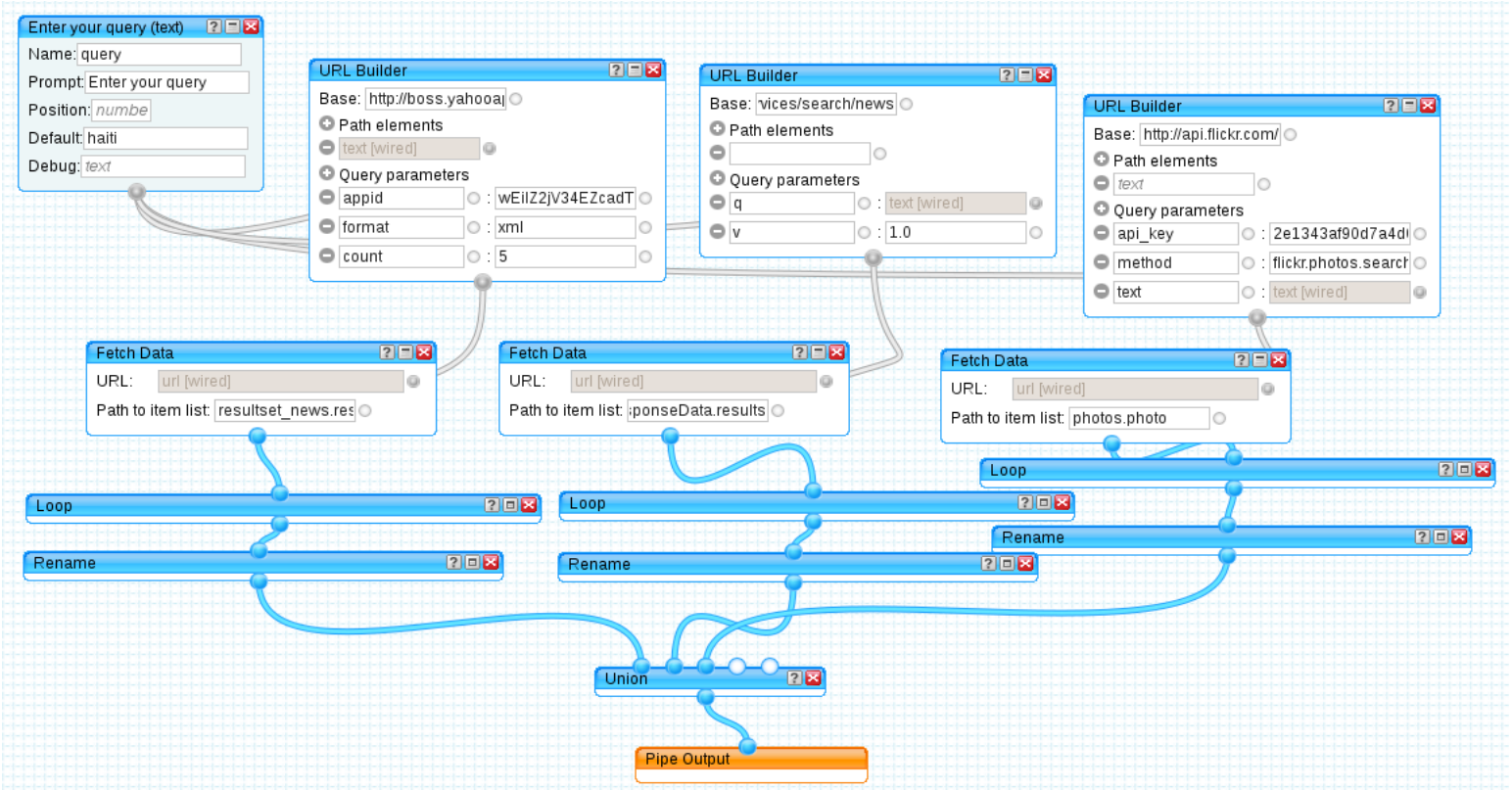
- Irrelevant service information in service composition
  - Too much service information returned to the developer or (semi-)automatic algorithms
- Irrelevant information between service composition and execution
  - Temporal distance between the composition time and execution time
    - The expected quality of service is invalid, triggering quality-aware service adaptation
- Irrelevant information in service usage
  - Results are returned without a clear usage and ownership causing data compliance problems

Example 1: Too much service information



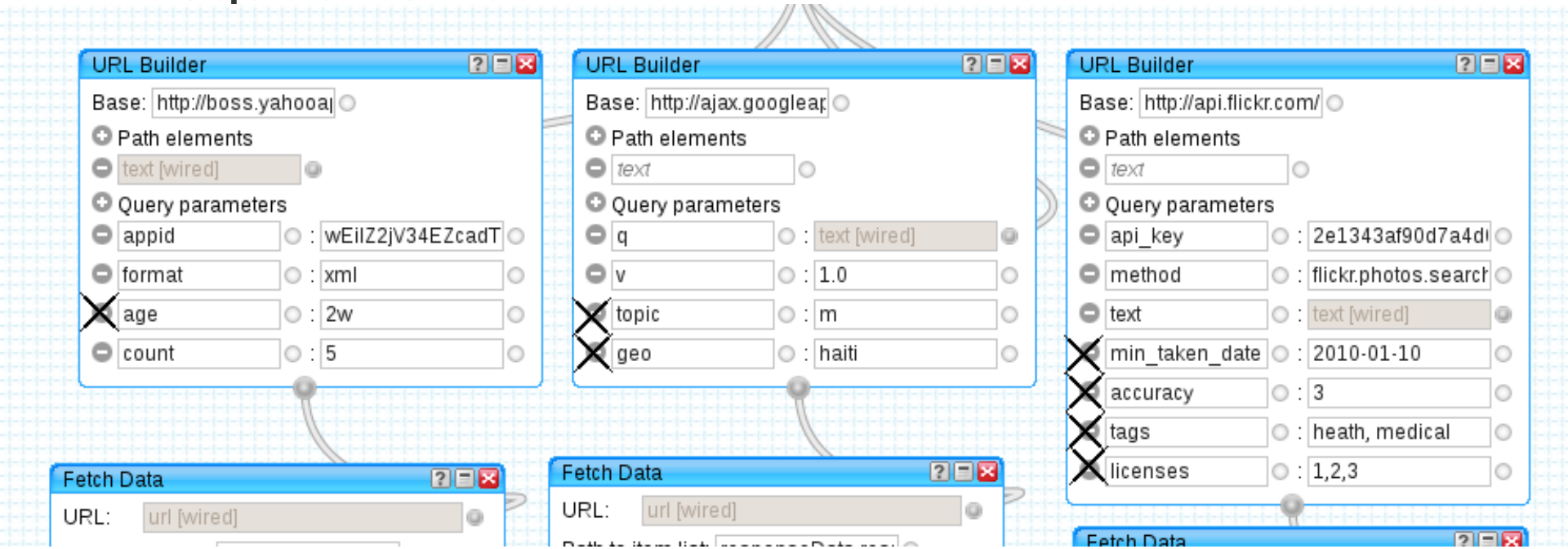
Example 2: Mashup

- Composition of Yahoo! Boss News Search, Google News Search , and Flickr
  - recent news and high-qualified images, but free-of charge, related to "Haiti earthquake"



If the composer is aware of context and quality parameters

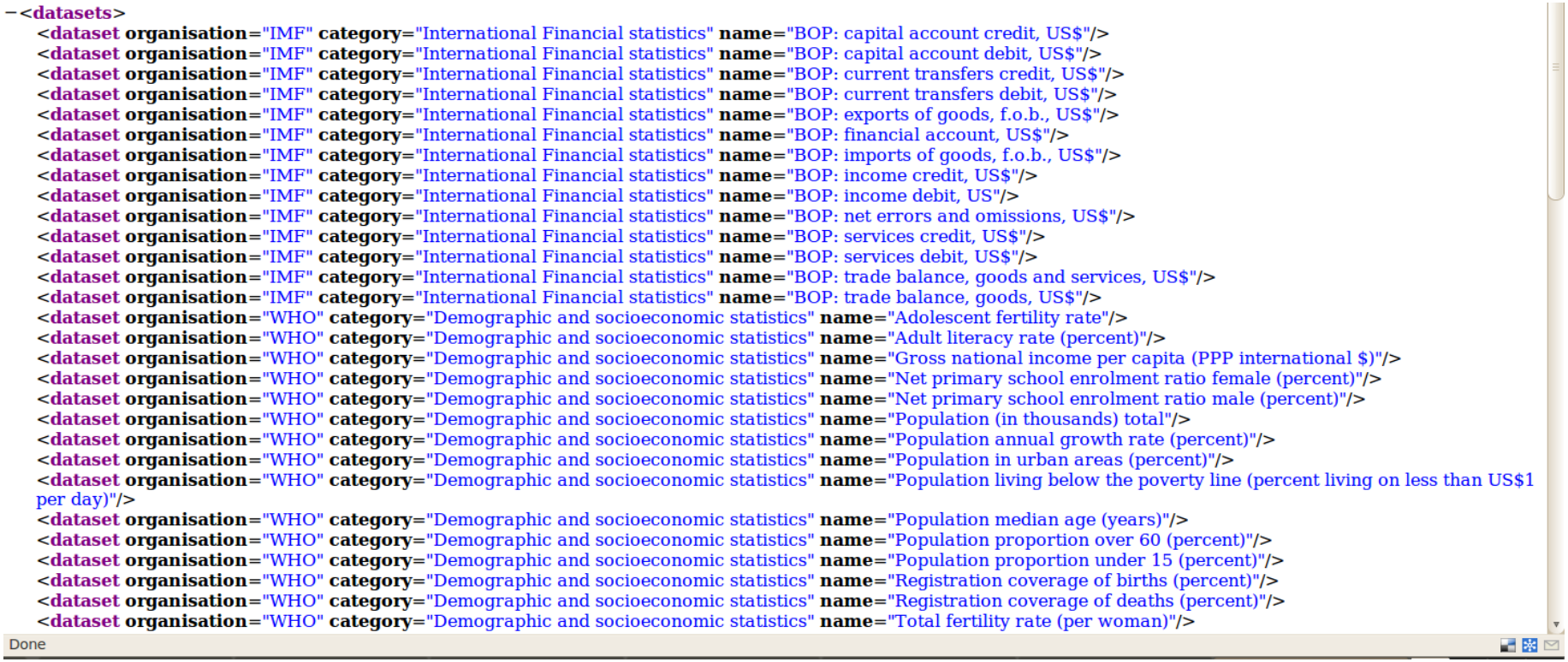
- Possible mappings of context and quality requirements



but it is a tedious task and hard to be automated and we are not sure we have a correct mapping.

Example 3: access to data-as-a-service (DaaS) in the cloud

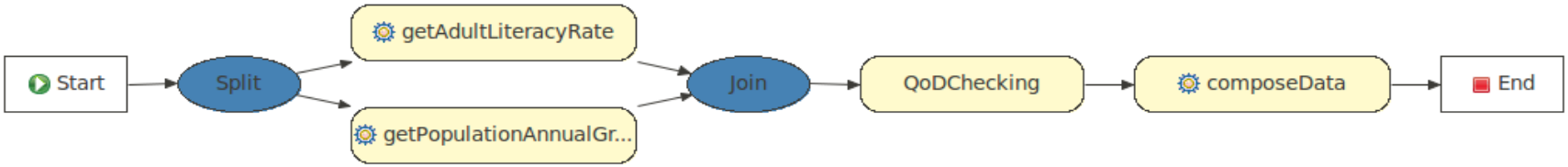
- Retrieve big datasets from RESTful services for further extraction, transform or data composition activities



```
−<datasets>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: capital account credit, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: capital account debit, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: current transfers credit, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: current transfers debit, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: exports of goods, f.o.b., US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: financial account, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: imports of goods, f.o.b., US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: income credit, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: income debit, US"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: net errors and omissions, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: services credit, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: services debit, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: trade balance, goods and services, US$"/>
  <dataset organisation="IMF" category="International Financial statistics" name="BOP: trade balance, goods, US$"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Adolescent fertility rate"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Adult literacy rate (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Gross national income per capita (PPP international $)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Net primary school enrolment ratio female (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Net primary school enrolment ratio male (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Population (in thousands) total"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Population annual growth rate (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Population in urban areas (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Population living below the poverty line (percent living on less than US$1
  per day)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Population median age (years)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Population proportion over 60 (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Population proportion under 15 (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Registration coverage of births (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Registration coverage of deaths (percent)"/>
  <dataset organisation="WHO" category="Demographic and socioeconomic statistics" name="Total fertility rate (per woman)"/>
Done
```

http://www.undata-api.org/

Example 3: quality problem

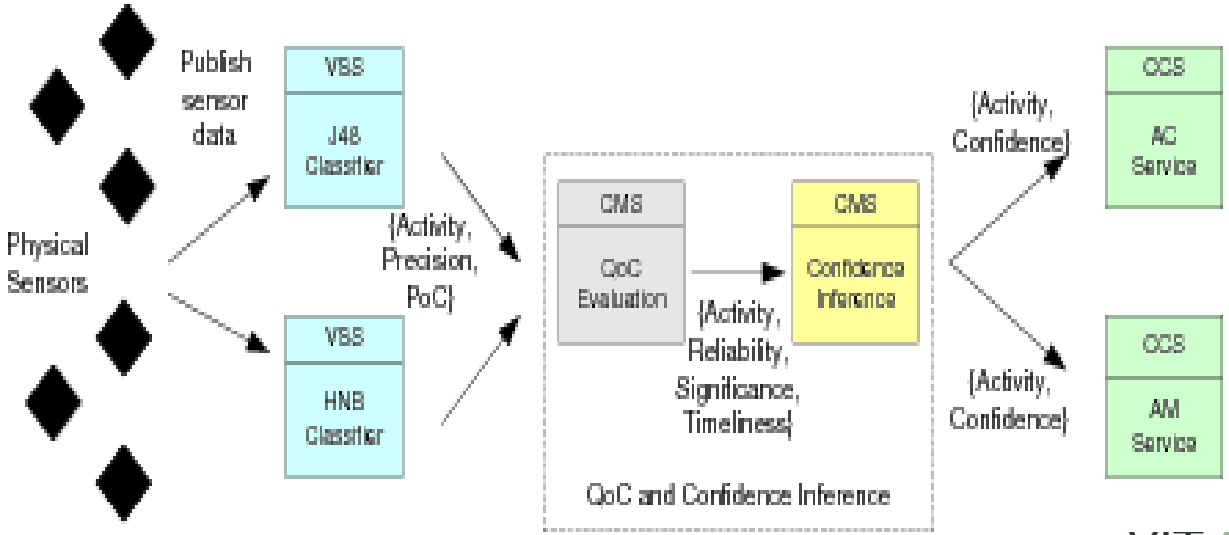


- Example: study the population growth and literacy rate from 1990-2009 for all countries in the world
- Without QoD: get datasets and perform mashup
- With QoD support:
  - Population annual growth rate (percent):
    - dataelementcompleteness= 0.8654708520179372, datasetcompleteness=0.7356502242152466;
  - Adult literacy rate (percent):
    - dataelementcompleteness=0.5874439461883408,datasetcompleteness=**0.0434 9775784753363**

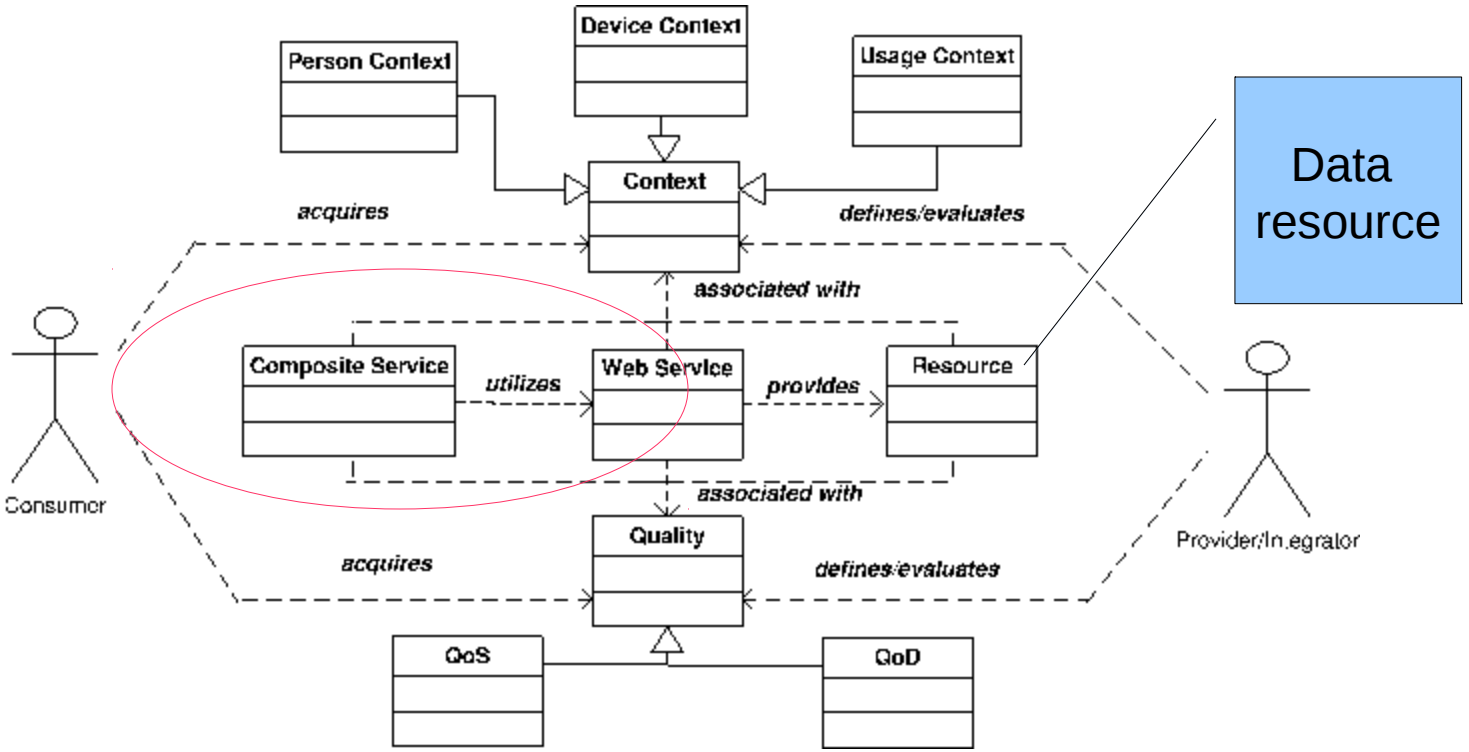
→ ***Should we retrieve the data and perform data composition?***

Example 4: Sensor-as-a-Service in Smart Environments

- Smart environments with several low level sensors:
  - Recognize human activities: idle, relaxing, cleaning up,
  - Provide context information for adaptive service discovery and execution
  - E.g., FP7 SM4All, FP7 EU OPPORTUNITY
- Virtual Sensor-as-a-Service provides human activities



Context, quality, and relevance dependencies in data service composition



- Only supporting **QoS and context awareness** at the **service level as a whole** is not enough

Our big picture of problems

- Context and quality information models
  - Unstructured description of context, QoS and quality of data (QoD)
  - Different specifications and terminologies
  - Mismatching semantics of information about services and data
- Context and quality information access APIs
  - No/Limited description of data and service usage
  - No API for retrieving quality and context information
  - No quality and context information associated with requested data
- Context and quality evaluation techniques
  - Missing evaluation of compatibility of context and quality of multiple services
  - Large/irrelevant data quantity

Require a „holistic integration“ of information models, APIs and evaluation techniques to support adaptive composition and execution

Current research focuses

- Context and quality information models
  - Often used only a fraction of QoS or context, several specifications that cannot be easily linked
- Access APIs
  - Mainly static publishing, few metrics at runtime but typically at the service as a whole level
- Adaptive and context-aware algorithms
  - Mainly for adapting individual services in a composition
  - Either consumer-service flow or composite service-service flow

The role of data concerns? Context and quality associated with data resources? Dependency chain: consumer-service-service-resource?

Our suggested roadmap

- Developing a meta-model and domain-dependent semantic representations for quality and context information specifications
  - Reconciliation of context/quality terms
  - Linked data
- Developing context and quality information that can be accessed via open APIs
  - On-the-fly content/quality access
- Developing techniques for context and quality evaluation
  - context and quality compatibility evaluation and composition

Data Concern Specification

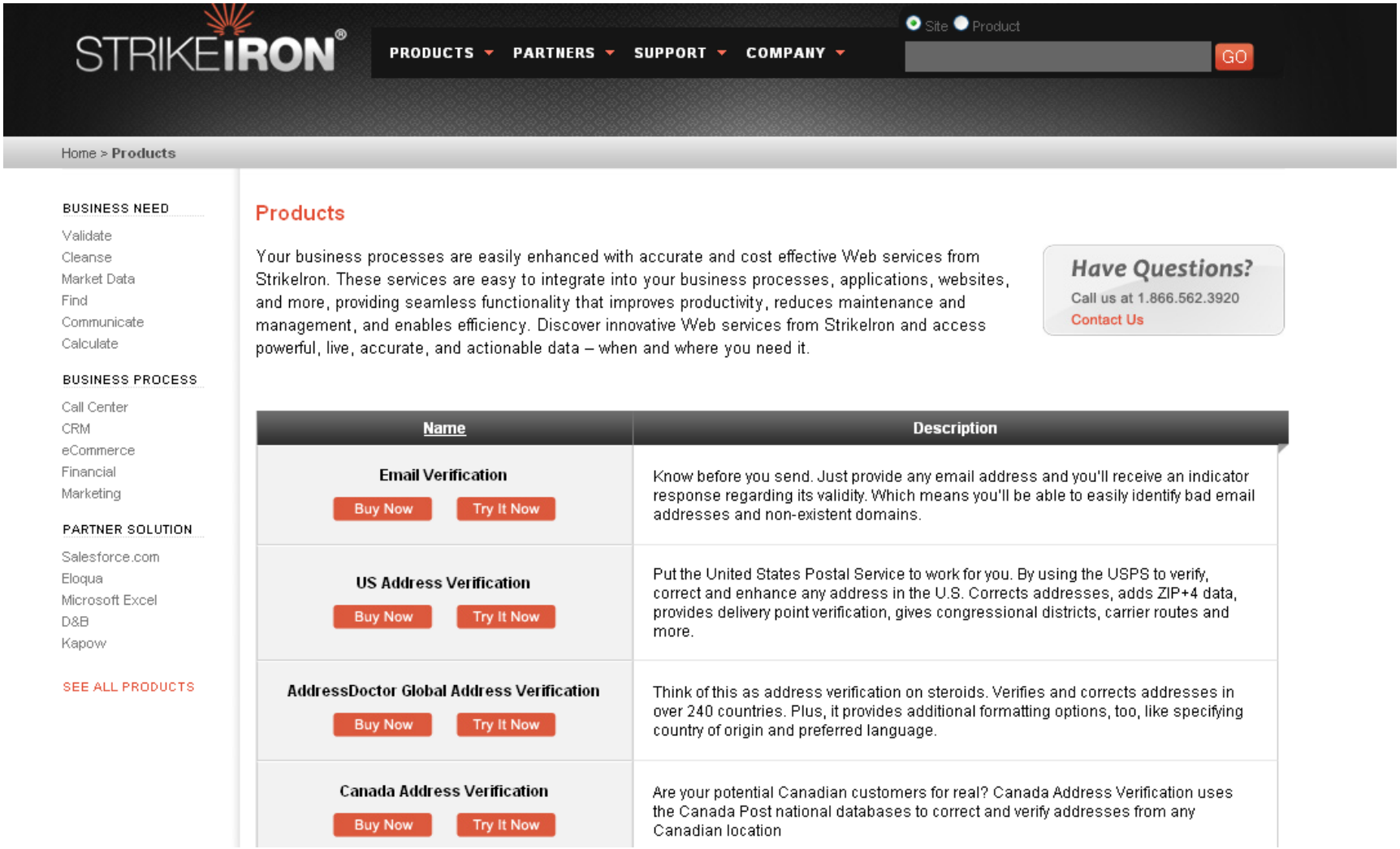
Background on DaaS

- Web services technologies, the SaaS model and the cloud computing model foster the concept of data/information as a service (DaaS)
- No precise definition but DaaSs
  - *Provide data capabilities* rather than provide computation on data or data based on computation
- Providing DaaS is an increasing trend
  - In both business and e-science environments
    - Bio data, weather data, company balance sheets, etc., via Web services
  - Academic research and industrial relevant research topics

Background - our view on DaaS

- Read-only DaaS versus CRUD DaaS
  - Read-only DaaS:
    - Service consumer can only read the data
  - CRUD DaaS
    - Service consumer can read/write, using their own data format
- Service APIs versus Data
  - *Service APIs* are used to CRUD *data*
  - They are not the same wrt concerns
- The data provider is not the same as the DaaS service provider

Read-only DaaS



Simple CRUD DaaS

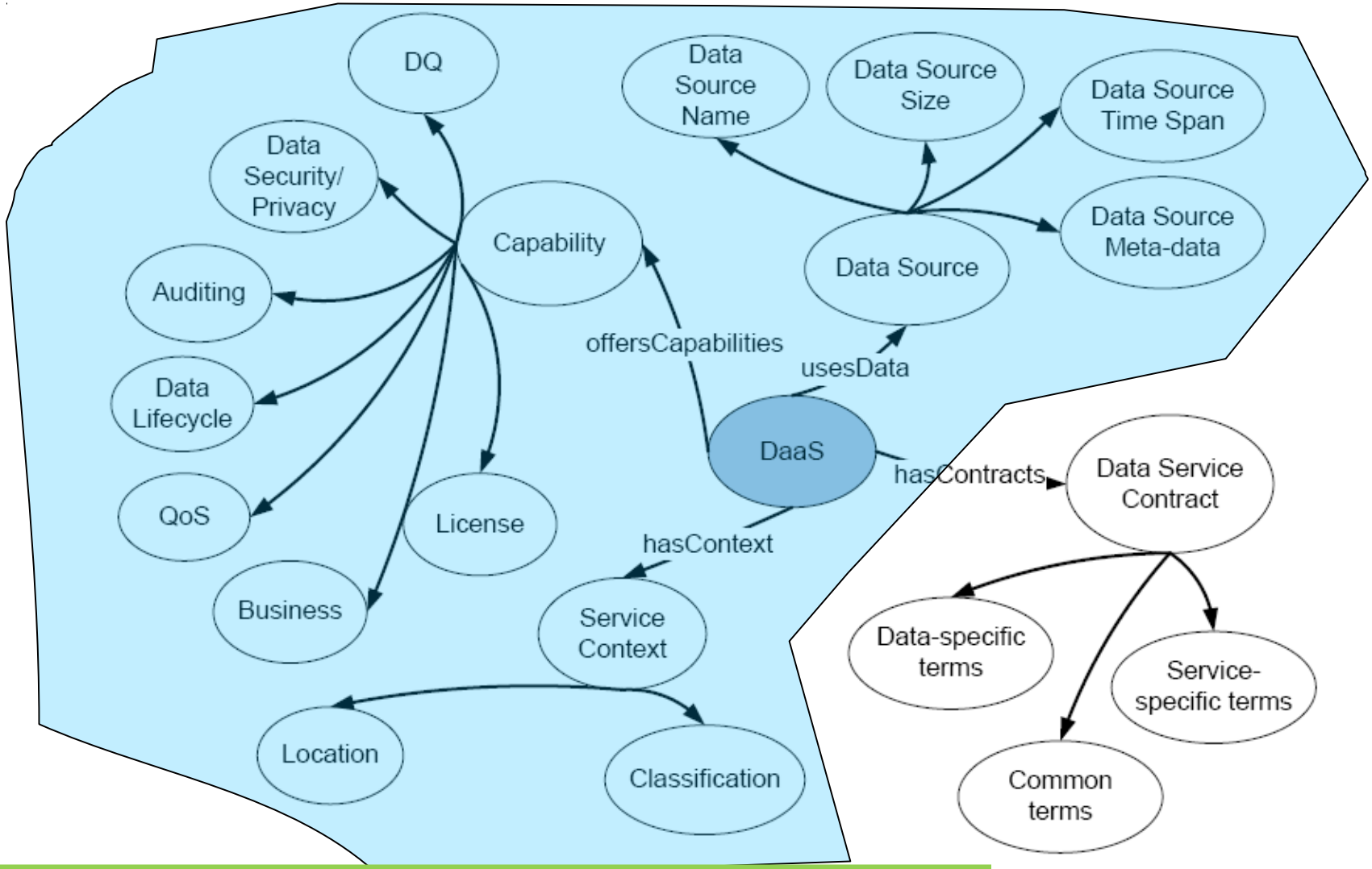
Issues

- DaaS concerns include QoS, DQ, service licensing, data licensing, data governance, etc.
- There is a lack of techniques for the publishing, discovery, selection and evaluation of data concerns
- There is a lack of techniques for integrating concerns for DaaSs
  - Data concerns and Service APIs concerns

The Importance of Concerns in Data Consumer‘s View

Concerns	Read-only DaaS	CRUD Daas
Data Quality	Important factor for the selection of DaaS. For example, the accurary and compleness of the data, whether the data is up-to-date	Expected some support to control the quality of the data in case the data is offered to other consumers
Data source	Important factor for the trustworthiness of the DaaS.	
Data & Service Usage	Important factor, in particular, price, data and service APIs licensing, law enforcement, and IPRs	Important factor, in paricular, price, service APIs licensing, and law enforcement
Data Governance		Important factor, for example, the security and privacy compliance, data distribution, and auditing
QoS	Important factor, in particular availability and response time	Important factor, in particular, availability, response time, depability, and security
Service Context	Useful factor, such as classification and service type (REST, SOAP), location	Important factor, e.g. location (for regulation compliance) and versioning

Conceptual Model for DaaS Concerns and Contracts



Check http://www.infosys.tuwien.ac.at/prototyp/SOD1/dataconcerns/

Capability Concerns

- Data Quality capabilities
  - Based on well-established research on data quality
    - Timelineness, uptodate, free-of-error, cleaning, consistency, completeness, domain-specific metrics, etc.
  - We mainly support the specification of DQ metrics for the whole DaaS but possible to extend to the service operation level
- Data Security/Privacy capabilities
  - Data protection within DaaS, e.g. encryption, sensitive data filtering, and data privacy
  - Many terms are based on the W3C P3P

Capability Concerns (cont.)

- Auditing capabilities
  - Logging, reporting (e.g., daily, weekly, and monthly), and warning
  - Support system maintenance, SLA monitoring, billing, and taxation
- Data lifecycle
  - Backup/recovery, distribution (e.g., a service is in Europe but data is stored in US), and disposition
  - Support system maintenance but also regulation on data

Capability Concerns (cont.)

- Data and service license
  - Usage permission: for data (distribution, transfer, personal use, etc.) and for service APIs (adaptation, composition, derivation, etc.)
    - We utilize some terms from ODRL/ODRL-S
  - Copyrights
  - Liability: e.g., who is reponsible for the loss due to a network disruption?
  - Law enforcement (e.g., US or European court)
  - Domain specific IRPs

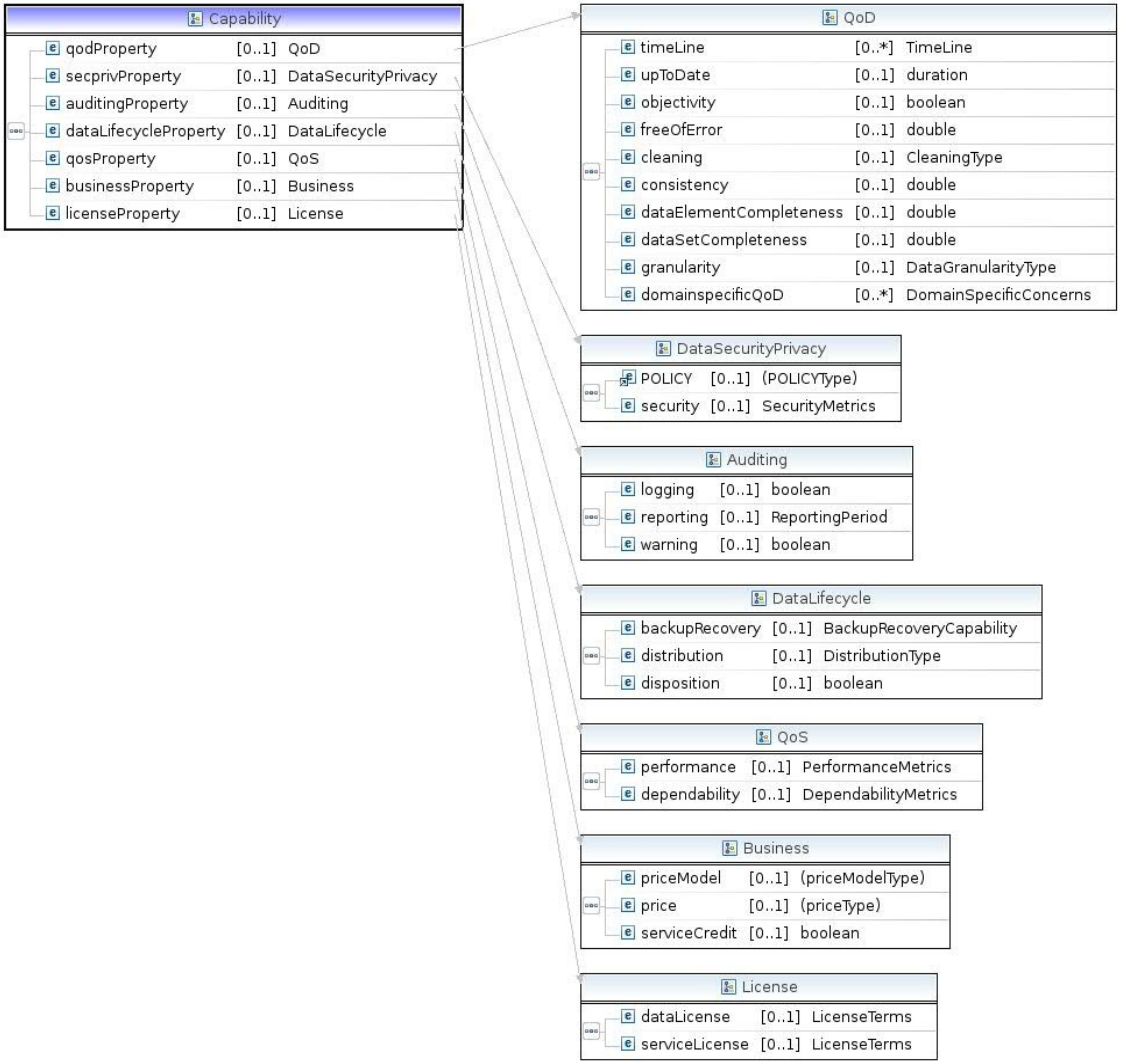
Data Source Concerns

- A DaaS may utilize data from many sources.
- Similar DaaSs may utilize data from the same source
- Data source properties
  - Name: e.g. ddfFlus or DataFlux or Mr A
  - Size
  - Timespan: the duration of collected data, e.g., more than 4 years in the eBay Data License
  - Update Frequency: how offen the data is updated
  - Etc.

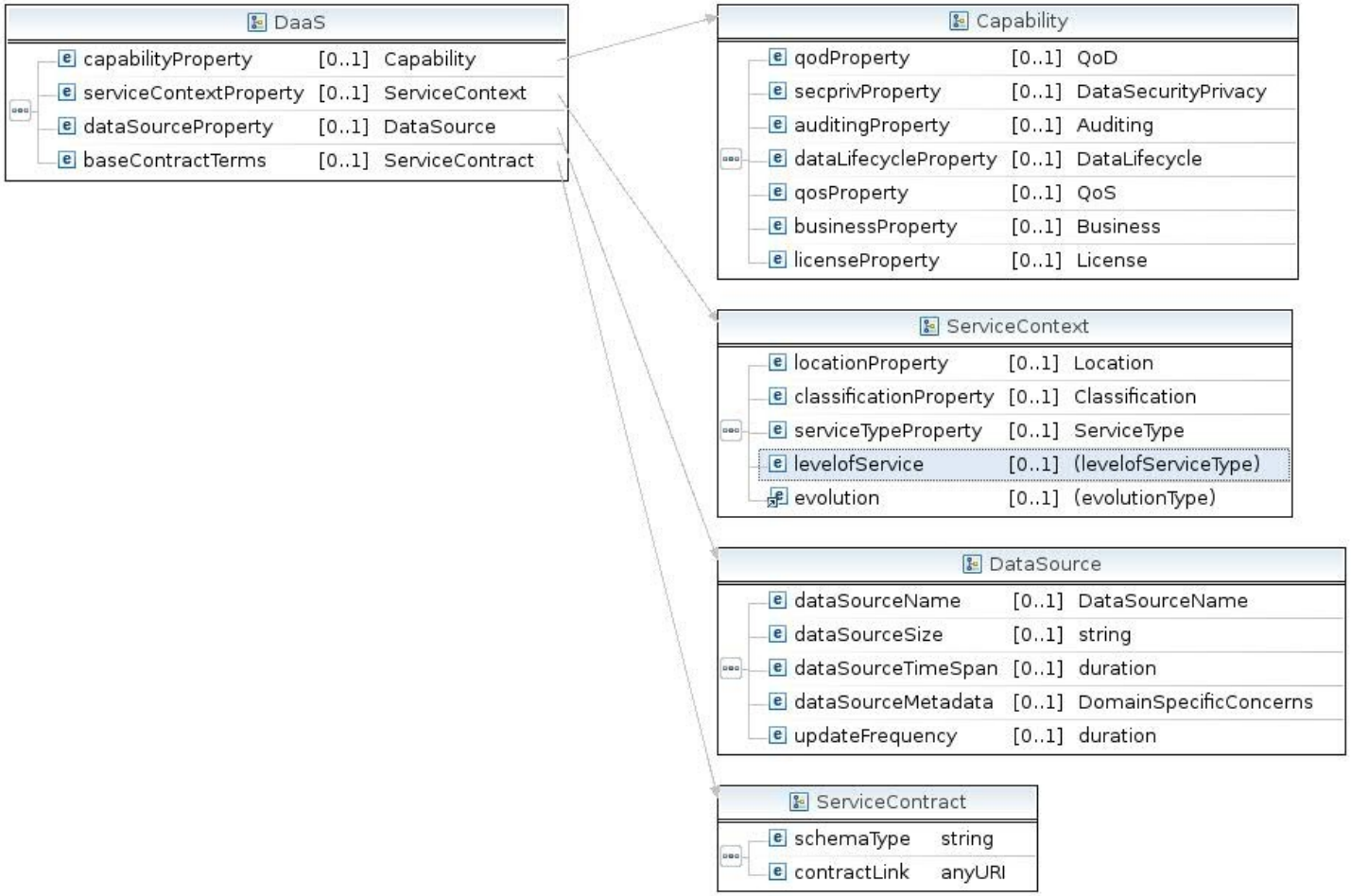
Service Context Concerns

- Location:
  - Selecting a DaaS in Amazon US Zone or European Zone?
- Service Type: REST or SOAP?
  - E.g., mobile client daas
- Level of Service
- Service Classification
  - Based on UNSPSC Code Classification Services
- Data Classification
- Service/data versioning

XML Diagram for the DaaS Capability Specification

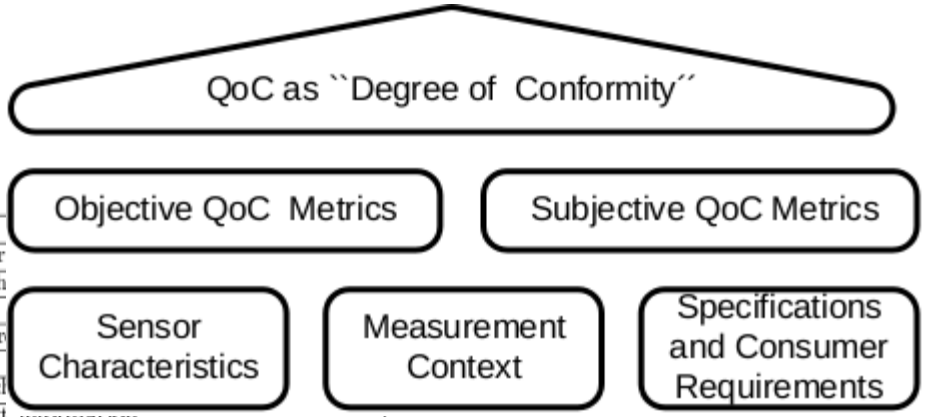


XML Diagram for DaaS Specification



Domain-specific aspect: Data Concerns for Context Information in Smart Environments

- Quality of context (QoC)

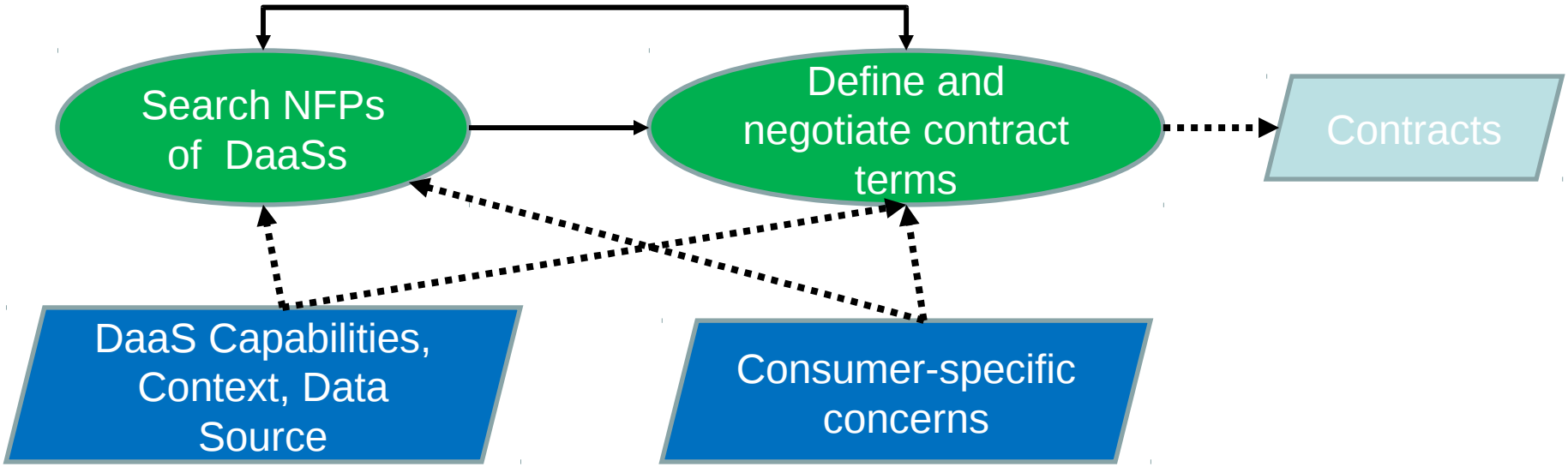


Classification	*Metric*	*Definition*
Sensor Characteristics	Accuracy	Extent to which data is corr
	Precision	Degree of exactness with wh
	Granularity	Degree of detail with which
	Time Period	Time interval between two r
	Sensor State	Physical state of sensor
	Sensor Range / Span	Maximum distance for whic
Measurement Context	Measurement Time	Time of collection of context information
	Sensor Location	Location of sensor when context information is collected
	Information Entity Location	Location of the real world entity about which context is collected at the time of collection of context
	Available Attributes	Number of attributes that have a value for that context object
Specifications and Consumer Requirements	Validity Time	Maximum length of time for which a specific type of context information is stable
	Required Attributes	Number of attributes that are required to have a value for that type of context information
	Critical Value	Level of importance of context information of a specific type
	Access Level	Information about the rights of context consumer to access certain type of information
QoC Metric	Reliability	Indicates the extent to which context can be considered credible
	Timeliness	Indicates validity of context to use considering its freshness
	Completeness	All aspects of phenomenon in the environment have been shown
	Significance	Critical value of context information for a specific application
	Usability	Indicates suitability of for use for an intended purpose
	Access Right	Indicate the extent to which owner of context allows the context consumer to access context
	Representation Consistency	Extent to which context representation format is consistent to consumer requirements

Table 1 Brief description of concepts in QoC processing model

From Capability/Context to Service Contract

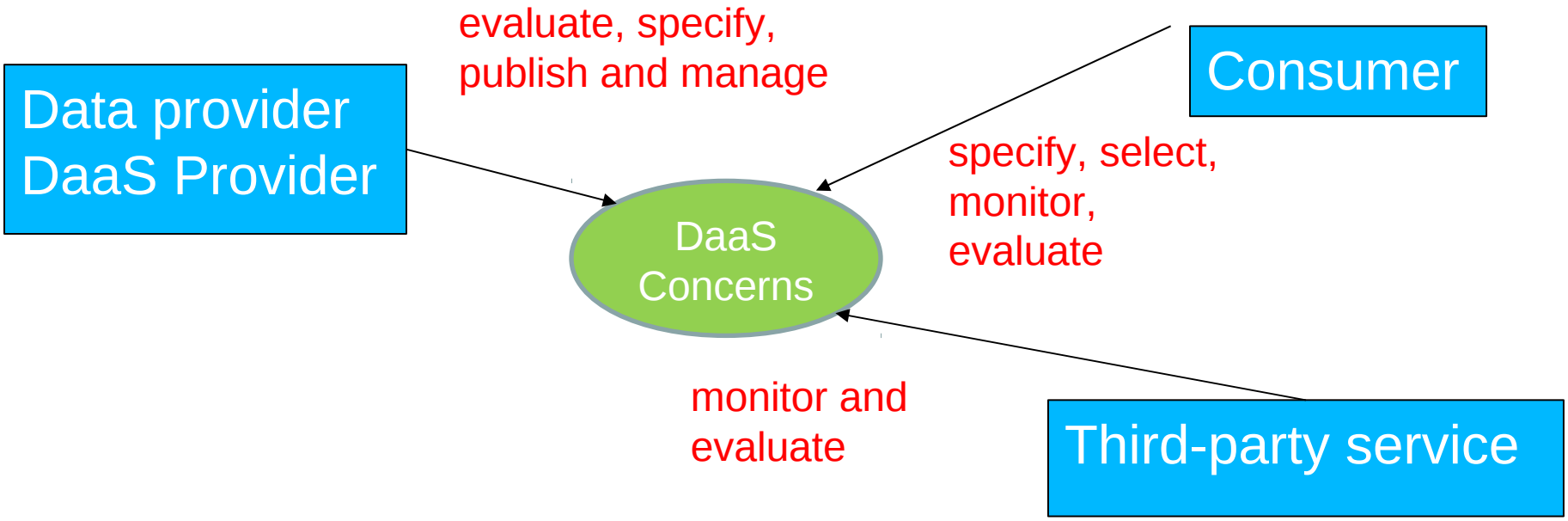
Non-functional parameters (NFPs) to Service Contracts



- A service contract includes a set of generic, data-specific and service-specific conditions established based on concerns

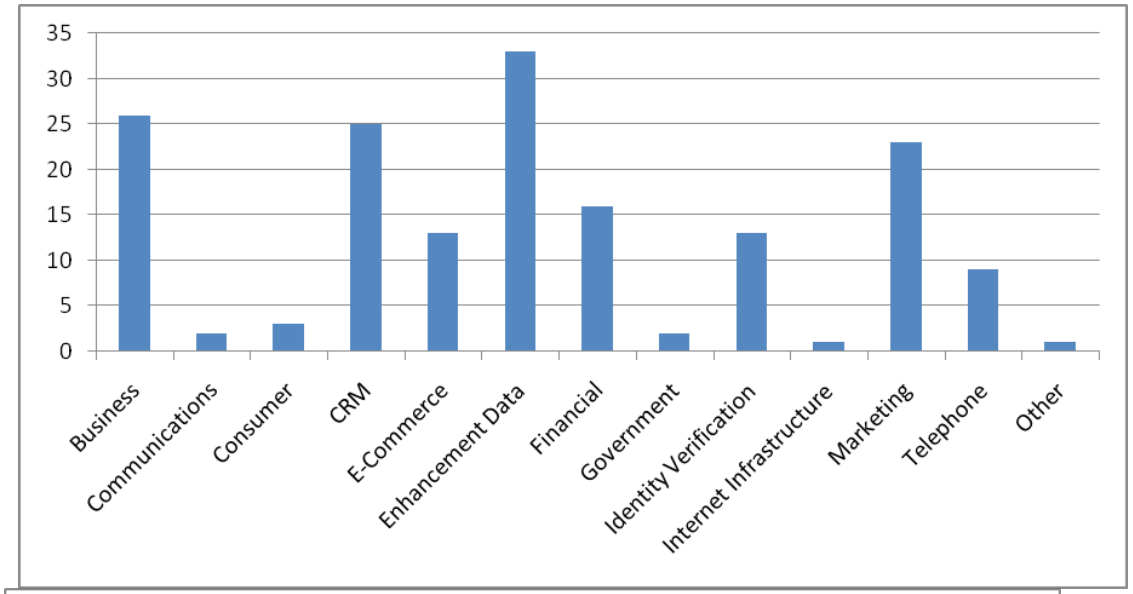
Populating DaaS Concerns

The role of stakeholders in the most trivial view

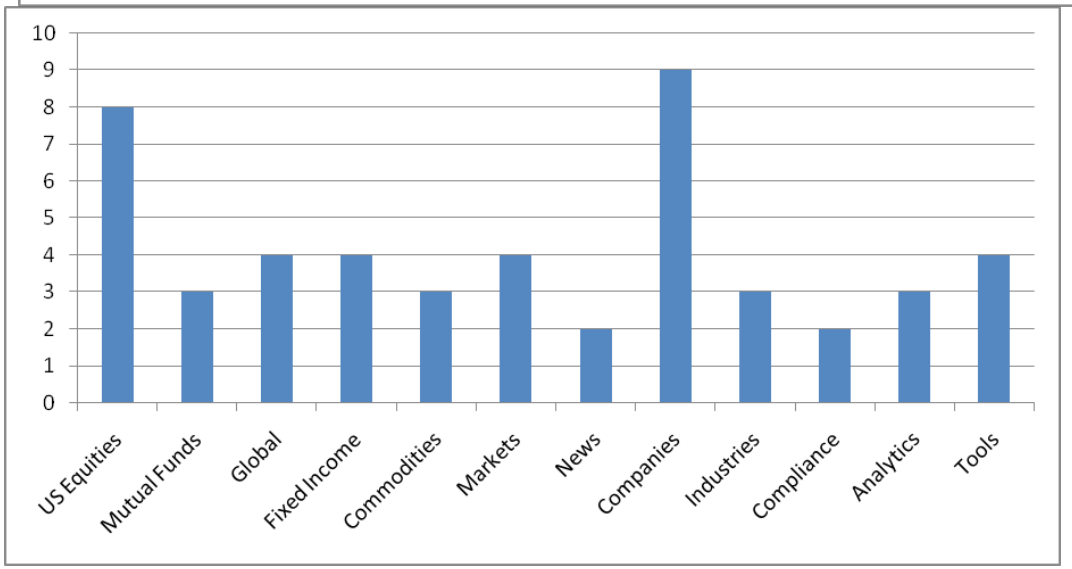


Service Classification

- StrikeIron Web services

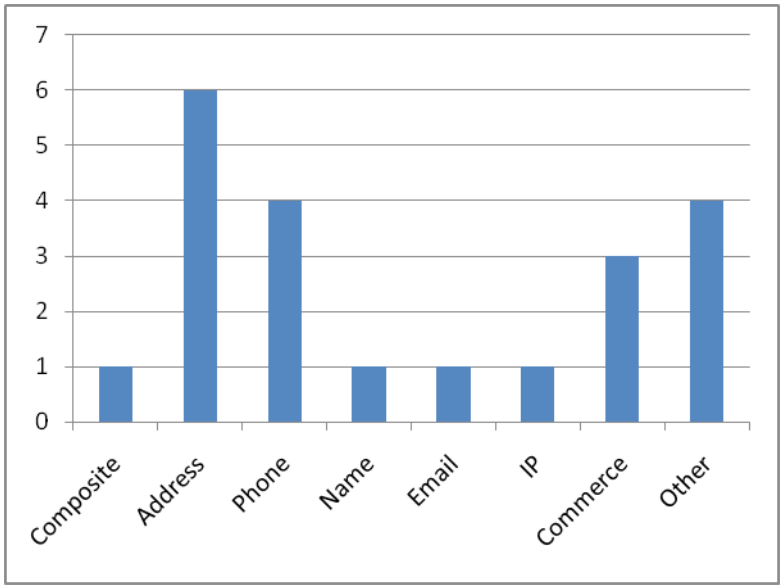


- Xignite Web services

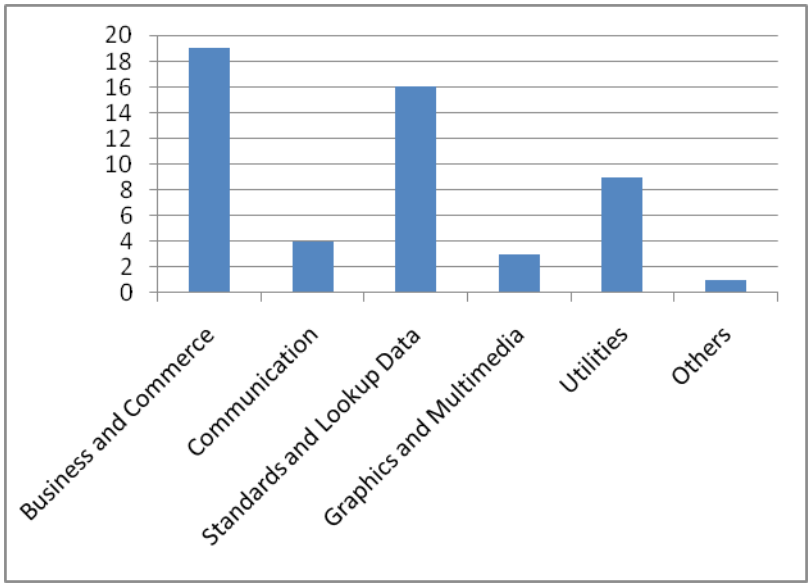


Service Classification

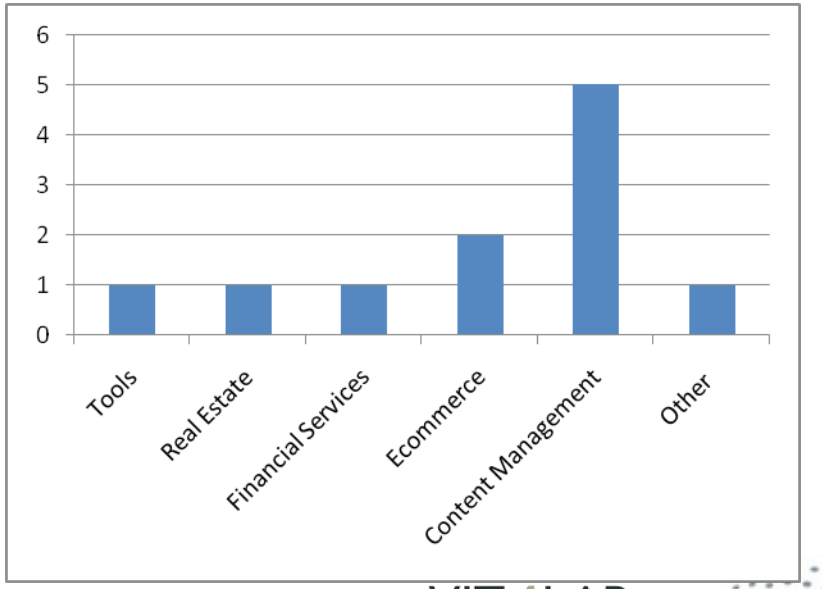
- ServiceObjects Web Services



- WebservicesX Web services

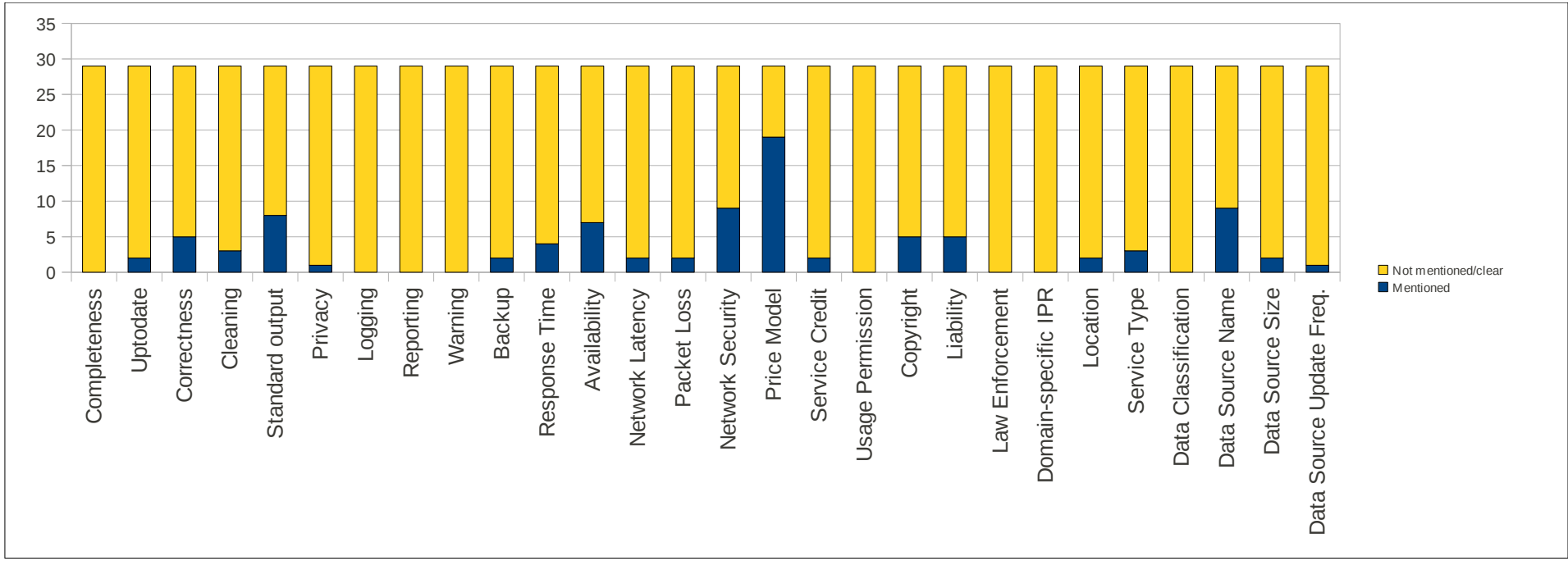


- XWebService Web services



Concerns in HTML descriptions

- 29 services from 7 providers, most are SOAP-based



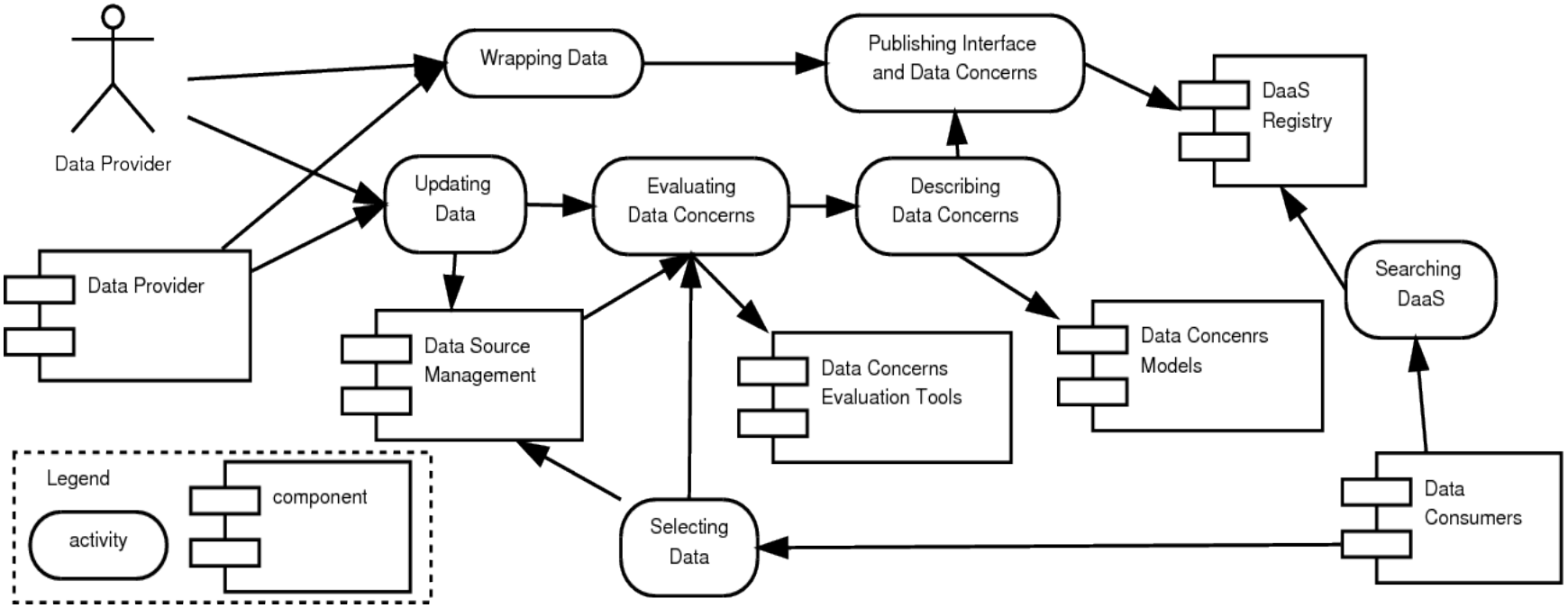
Concerns of DaaSs for Scientific Data

From the DaaS description point of view

Service Registries	DQ	QoS	Business	Licensing	
				Ownership	Usage permission
GBIF	No	No	No	unstructured	unstructured
EBI Web Services	No	No	No	No	No
EMBRACE Service Registry	No	No	No	No	No
BioCatalogue	No	No	unstructured	unstructured	unstructured

Data Concern Evaluation

Data concern-aware service engineering process

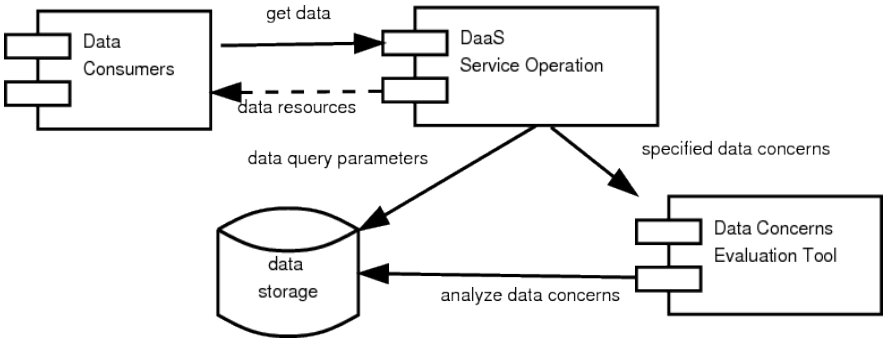


Data Concern Evaluation

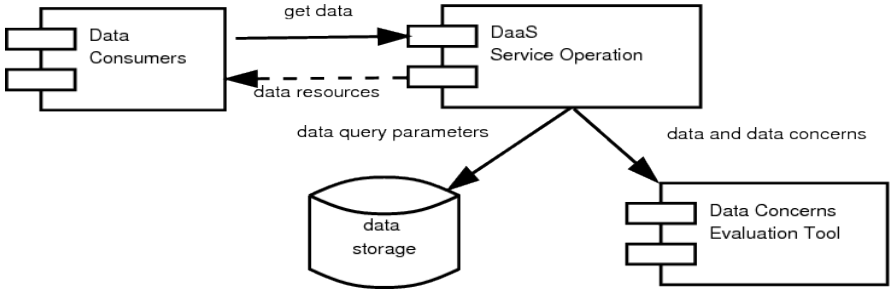
- Evaluation scope
  - Data resources
  - Service Operation
  - Service as a whole
- Evaluation modes
  - Off-line and on-the-fly
- Integration models
  - Push versus pull
  - Pass-by-value versus pass-by-reference

Possible Data Concern Evaluation

Pull, pass-by-references

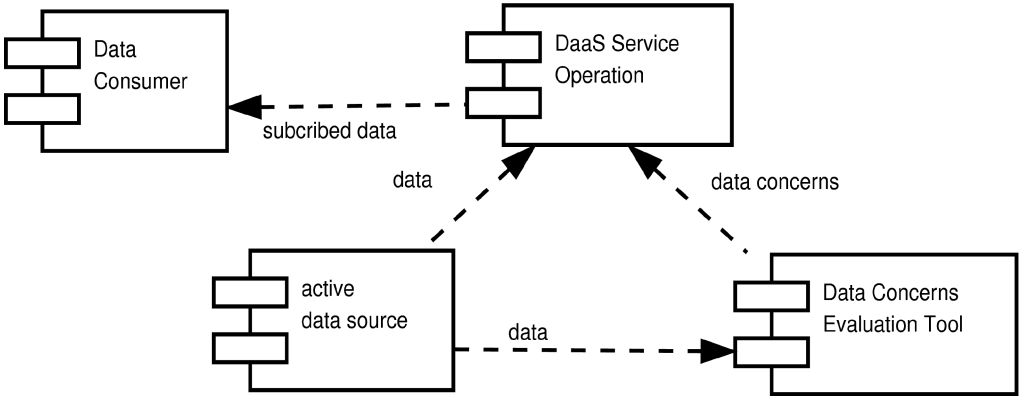


Pull, pass-by-values

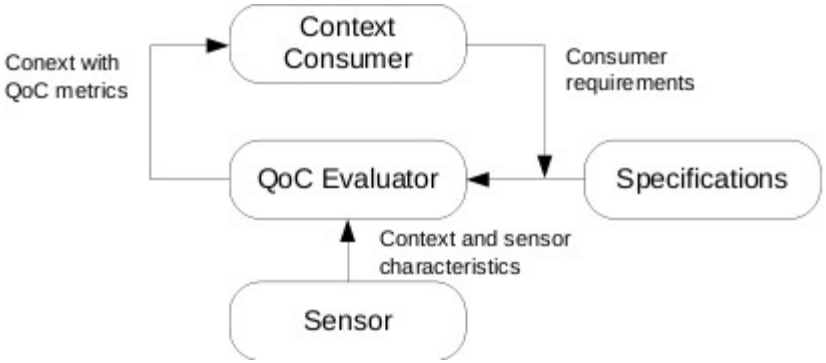


Active data sources, sensors

Push, pass-by-values



Domain-specific: smart environment

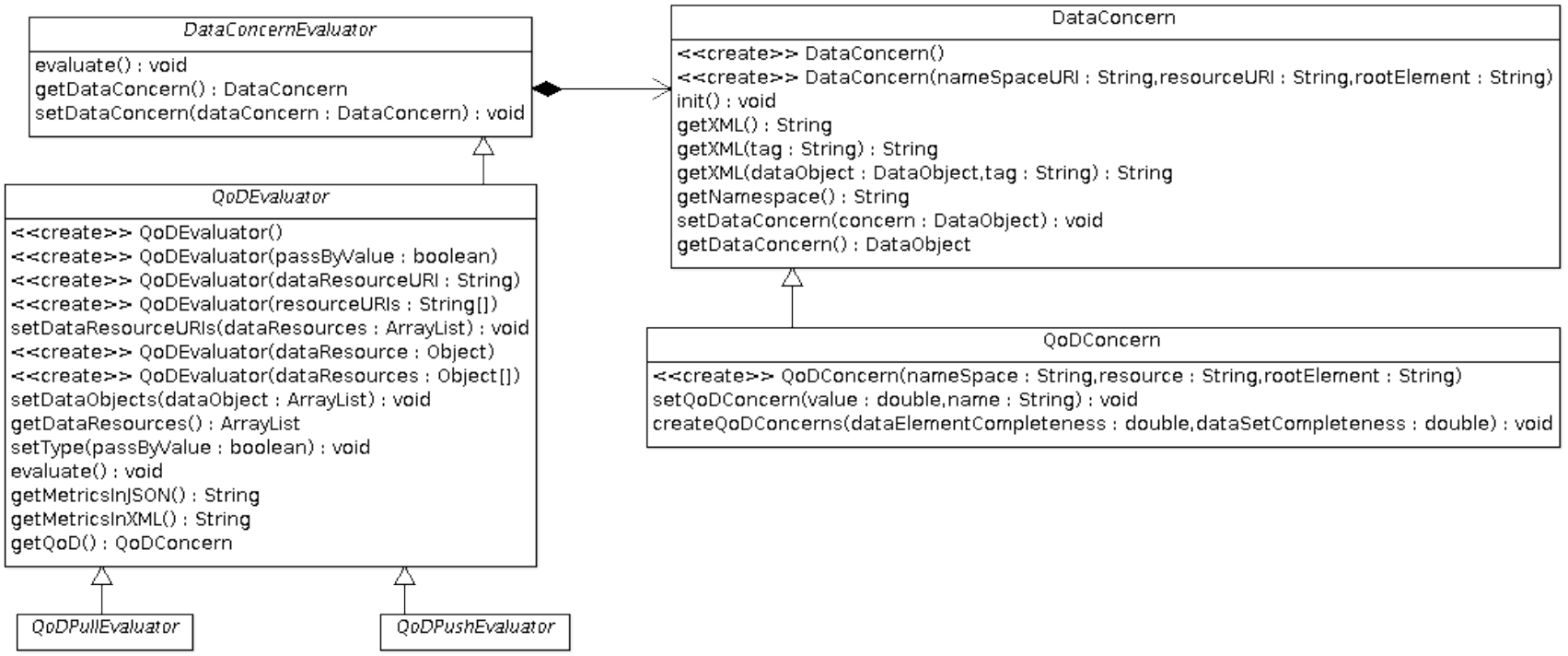


Data Concern Publishing

- Off-line publishing of data concerns
  - suitable for static data concerns
  - the publishing of data concerns of a data resource is separated from the service operation which provides the access to the data resource
- On-the-fly publishing of data concerns by associating concerns with retrieved data resources
  - The resulting data resources (e.g., via queries) are annotated with data concerns evaluated by data concerns evaluation tools.
  - suitable for providing dynamic data concerns
- On-the-fly publishing of data concerns through queries
  - the use of different service operation parameters to query data concerns of data resources
  - suitable for validating data concerns before accessing data resources

QoD Framework

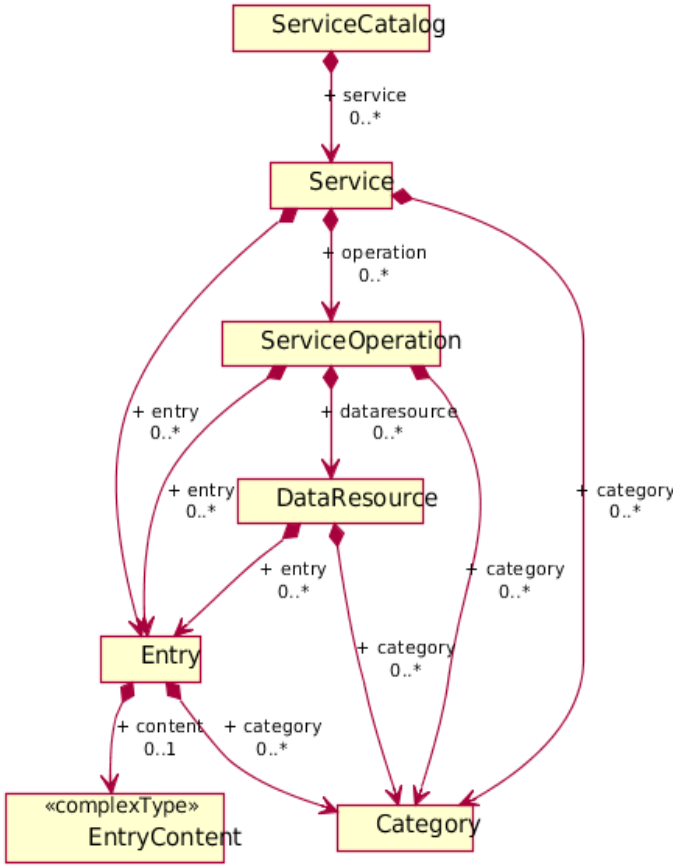
- A proof-of-concept implementation of data concern-aware service engineering process



QoD Framework: Publishing Concerns

off-line data concern publishing

- Common data concern publication specification
- A tool for providing data concerns according to the specification



QoD Framework: Publishing Concerns

On-the-fly querying context and quality information associated with resources

- Using parameter convention
  - Based on metric names in the data concern specification
- Specifying requests by using utilizing query parameters the form of metricName=value

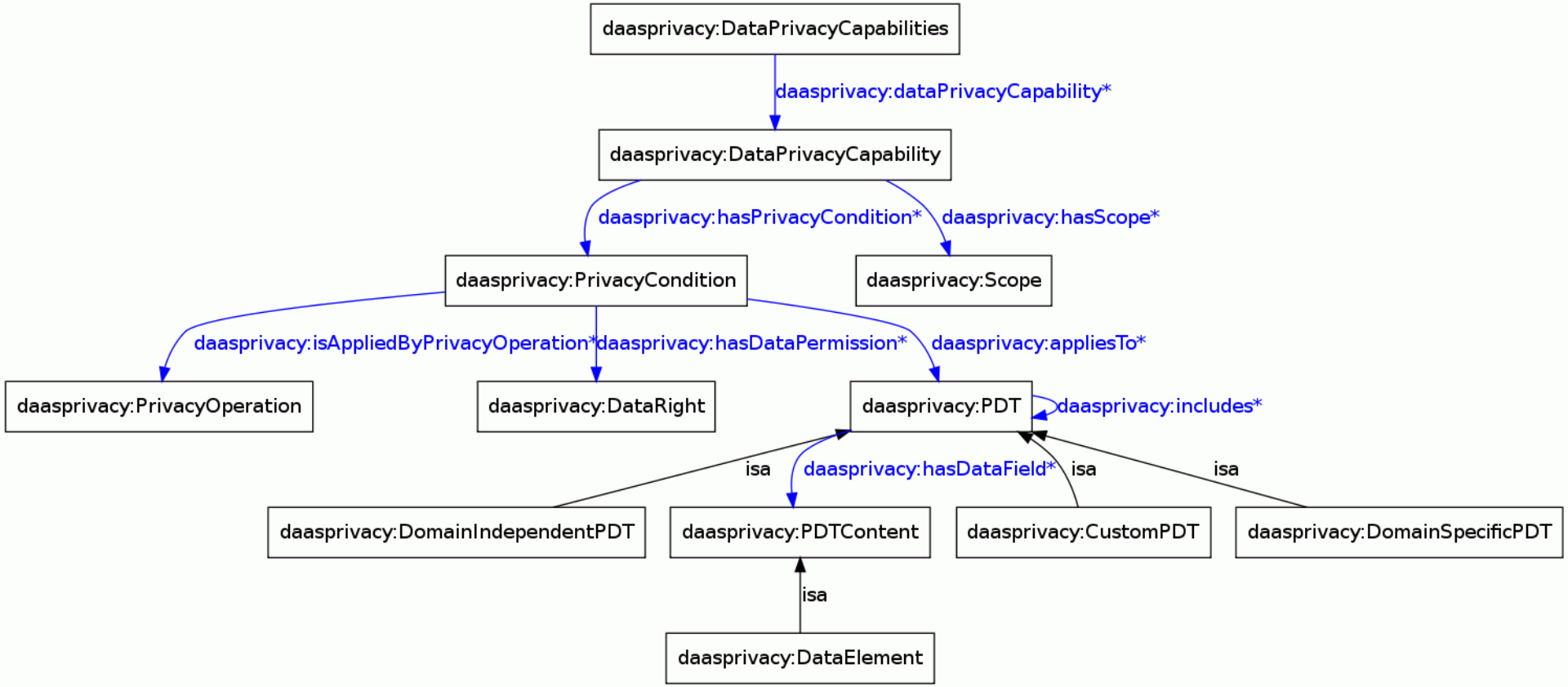
```
GET/resource?accuracy="0.5"&location=’’Europe”
```

QoD Framework: Data Concern Annotation

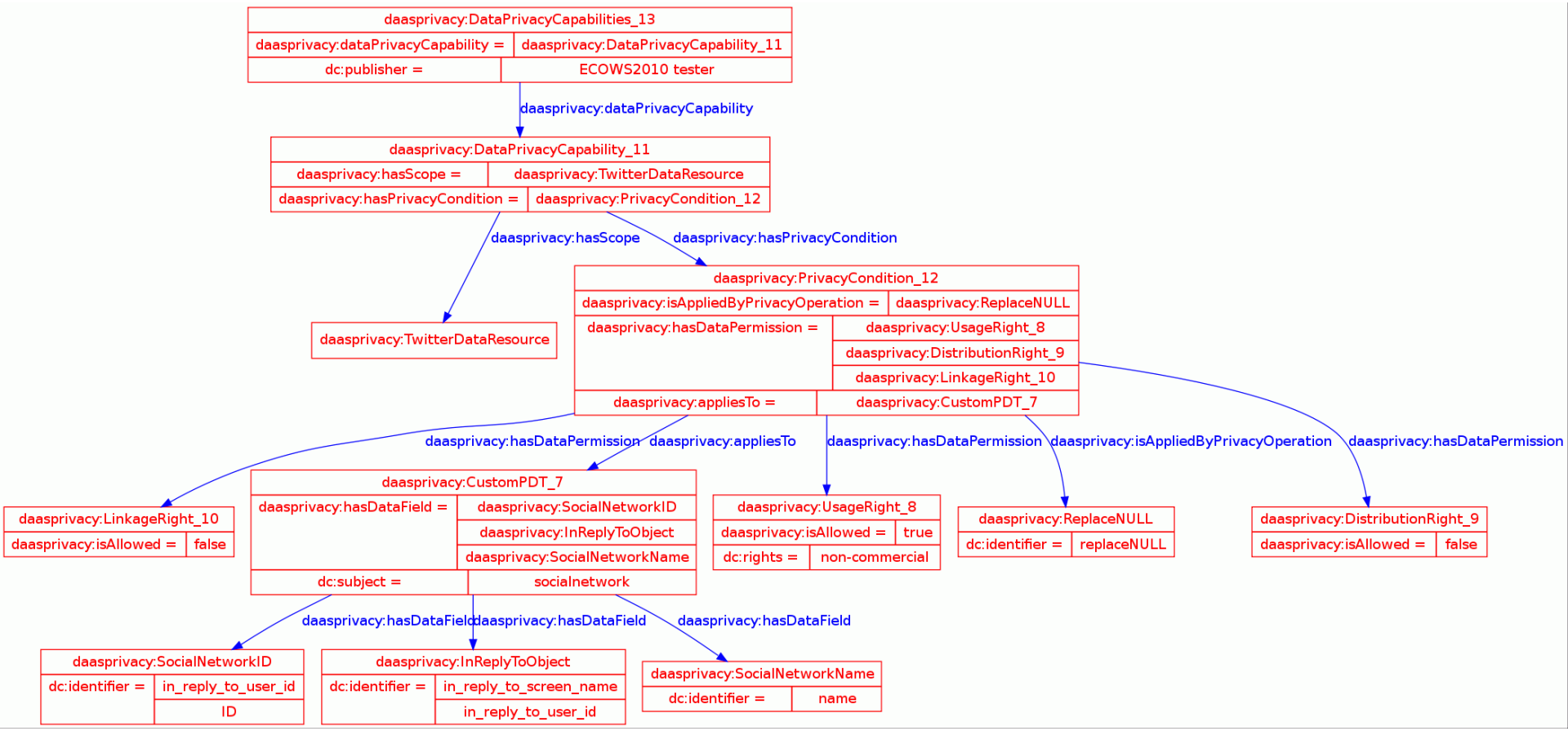
```
1   @GET
2   @Produces("application/xml")
3   public String getXml(@PathParam("id") String id, @QueryParam("QoD") String QoD) throws Exception {
4       //...
5       DataObject dataobject = getDataResourceByID(id);
6       //...
7       //return the requested data resource only
8       if (QoD == null) {
9           return toXML(dataobject, "http://www.undata-api.org", "results");
10      }
11      //evaluate and return only the quality of data of the requested data resource
12      if (QoD.isEmpty()) {
13          QoDUNDataEvaluator qodEval = new QoDUNDataEvaluator(dataobject);
14          qodEval.setType(true);
15          qodEval.evaluate();
16          return qodEval.getMetricsInXML();
17      }
18      //evaluate and return the requested data resource and its quality of data
19      if (QoD.equals("annotation")) {
20          QoDUNDataEvaluator qodEval = new QoDUNDataEvaluator(dataobject);
21          qodEval.setType(true);
22          qodEval.evaluate();
23          DataObject resultObject = dataobject;
24          //...
25          resultObject.getSequence().add("qod",qodEval.getDataConcern().getDataConcern());
26          return toXML(resultObject, "http://www.infosys.tuwien.ac.at/SOD1/undata-api", "results");
27      }
28      //...
29  }
```

Data privacy for DaaS: Privacy Concern Model

- Data privacy concerns are annotated with WSDL and MicroWSMO



Data Privacy for DaaS: Example for Twitter Data



http://infochimps.org/datasets/twitter-haiti-earthquake-data

Applications of Concerns

How can we utilize data concerns?

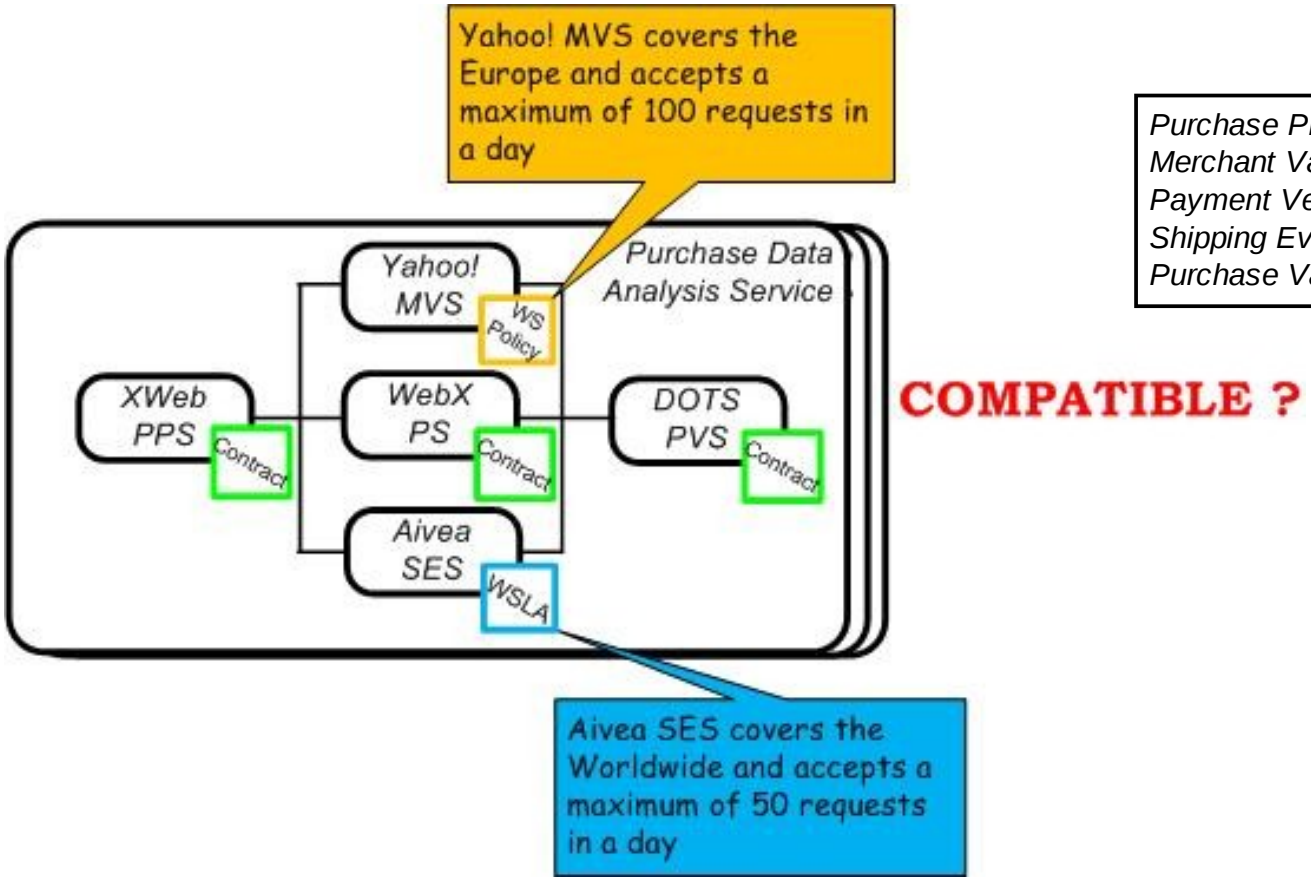
- Service composition and execution
  - Adaptive, context-aware service selection and execution algorithms
    - Can be extended to cover data concerns
  - Runtime data concerns access
    - Data quality-aware adaptation of services
  - Data and service contract compatibility
  - Filtering irrelevant service information
- Application domains
  - DaaSs in the Cloud, data composition, context-aware computing

Service Contract Compatibility

SECO2: Motivation and Background

- Besides a WSDL document stating the offered functionalities, a Web Service can be characterized by a *service contract* .
- A service contract
  - establishes the understanding between a service consumer and a service provider;
  - specifies conditions on non-functional parameters(NFPs)/concerns, such as:
    - *Quality of Service* (e.g., response time);
    - *Business terms* (e.g., service price);
    - *Context terms* (e.g., service coverage);
    - *License terms* (e.g., limitation of liability).
- No/several standard languages for service contract descriptions
  - Several proposals (e.g., WSLA, WSOL, ODRL-S, WS-Policy)

Motivation and Background (cont.)



Purchase Processing Service (**PPS**)
Merchant Validation Service (**MVS**)
Payment Verification Service (**PS**)
Shipping Evaluation Service (**SES**)
Purchase Validation Service (**PVS**)

COMPATIBLE ?

- The heterogeneity of languages specifying contracts
- The compatibility among services in a composition
- The compatibility between a (composite) service and a consumer's specific-conditions

Motivation and Background (cont.)

Past research…

- has neglected contracts of composite services when performing service composition
  - by considering mainly functional parameters
  - by assuming that contracts are described by a single language.
- has not focused on tools and algorithms dealing with contract compatibility evaluation when combining different services from different providers.
  - mainly contract negotiation between consumer and service in a point-to-point manner.

Motivation and Background (cont.)

- Some works address QoS-based compatibility for control flows of service compositions.
- Currently, no techniques to check contract compatibility for data (i.e., the input/output of services), whose contract terms are not always the same to that of the service operations.
  - An example is Google Maps: a *free-for-charge service* but the *copyrighted data* (i.e., the maps)
  - There is still a big debate on data licensing but you can sell your data, e.g., see http://infochimps.org/
- QoS, Business, License and Context terms differently influence data/control flows of the service composition.

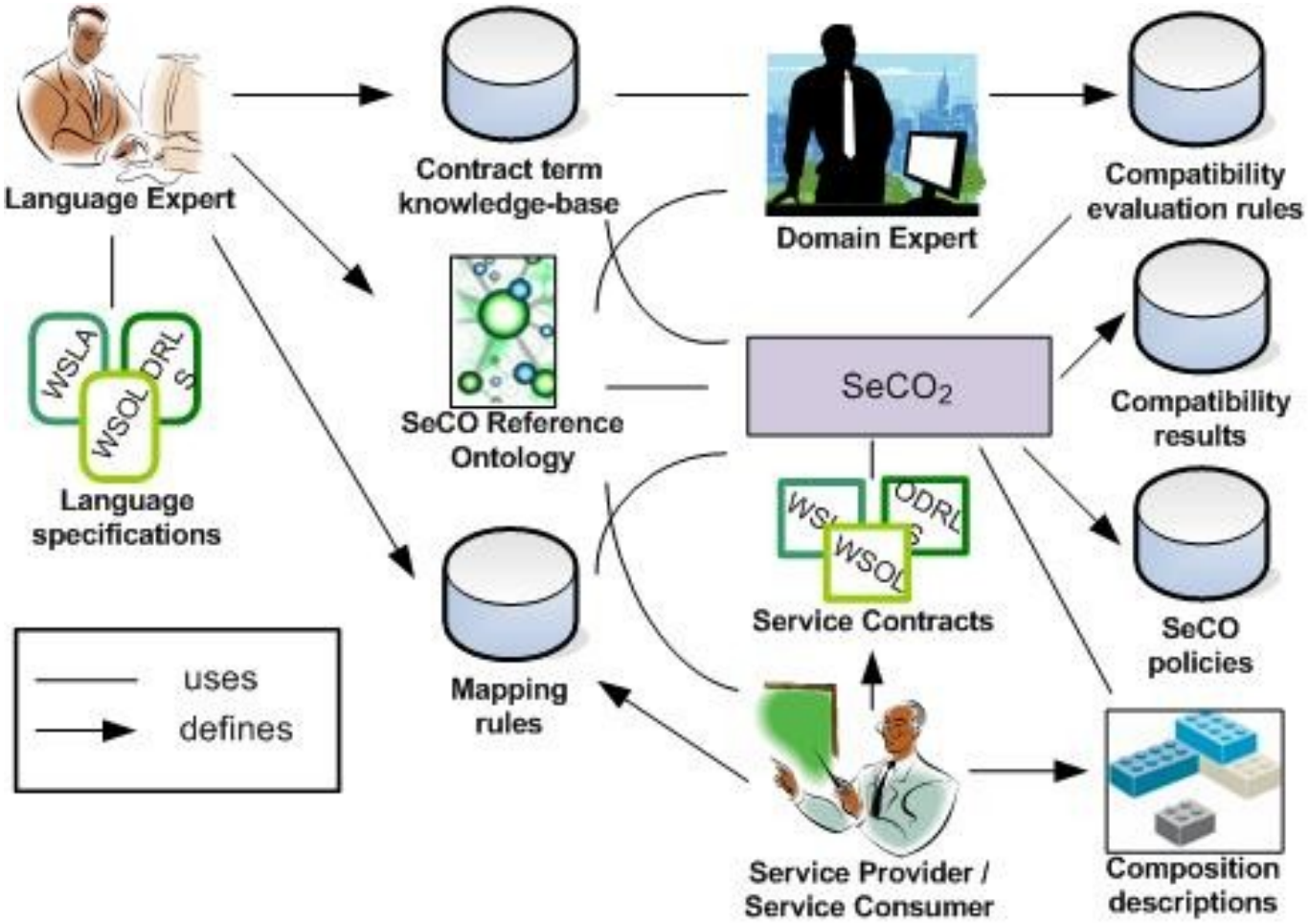
	control flow	data flow	independent
Quality of Service (QoS)	X		
Service Context			X
Business	X	X	
License	X	X	

Table 1. Data and control flows in contract compatibility evaluation

The SeCO$_2$ Framework

- SeCO2 deals with service contract compatibility by considering
  - two aspects – *service APIs* and *provided data* concerns;
  - a rich set of contract properties (e.g., *QoS, Data quality, Business, License* and *Context terms*);
  - several service contract specification languages (e.g., WSLA, WSOL, ODRL-S) together.
- SeCO2 supports
  - semantic service contract descriptions (namely, *SeCO policies*);
  - service contract compatibility evaluation and recommendation;
  - compatibility based on both data and control flows of the service composition;
  - an extensible reference ontology (namely, *SeCO reference ontology*) and a *Contract term knowledge-base;*
  - a rich set of mapping and compatibility evaluation rules.

The SeCO$_2$ Framework



Currently we deal with **modeling and mapping service contracts** and **contract compatibility evaluation among services in a composition**

Modeling and Mapping Service Contracts

- ***Problem***: Heterogeneity in service contract specifications.

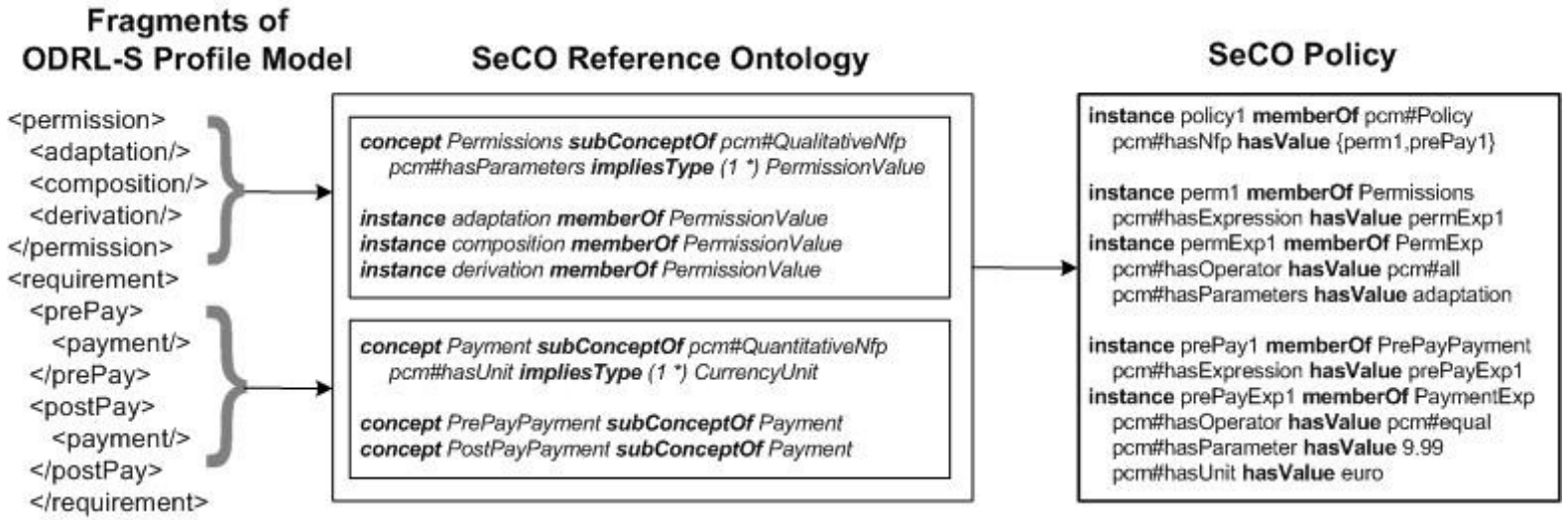
- Three types of languages for the specification of service contract properties:
  - **Type A** (e.g., ODRL-S): includes *languages allowing the specification of predefined properties*.
  - **Type B** (e.g., WSLA): includes *languages allowing the specification of user-defined properties*.
  - **Type C** (e.g., WSOL): includes *languages allowing the specification of properties defined in user ontologies*.
- Ontology alignment tools cannot be used to fully automate the mapping between different specifications.

Modeling and Mapping Service Contracts

- ***Solution:*** $SeCO_2$ makes service contracts comparable through the wrapping to specifications (i.e., *SeCO Policies)* built on a common meta-model
  - without loss of information;
  - by means of the *SeCO Reference Ontology* and predefined mapping rules;
  - supporting the use of lexical databases (e.g., *WordNet*) and ontology alignment tools (e.g., *H-match*).

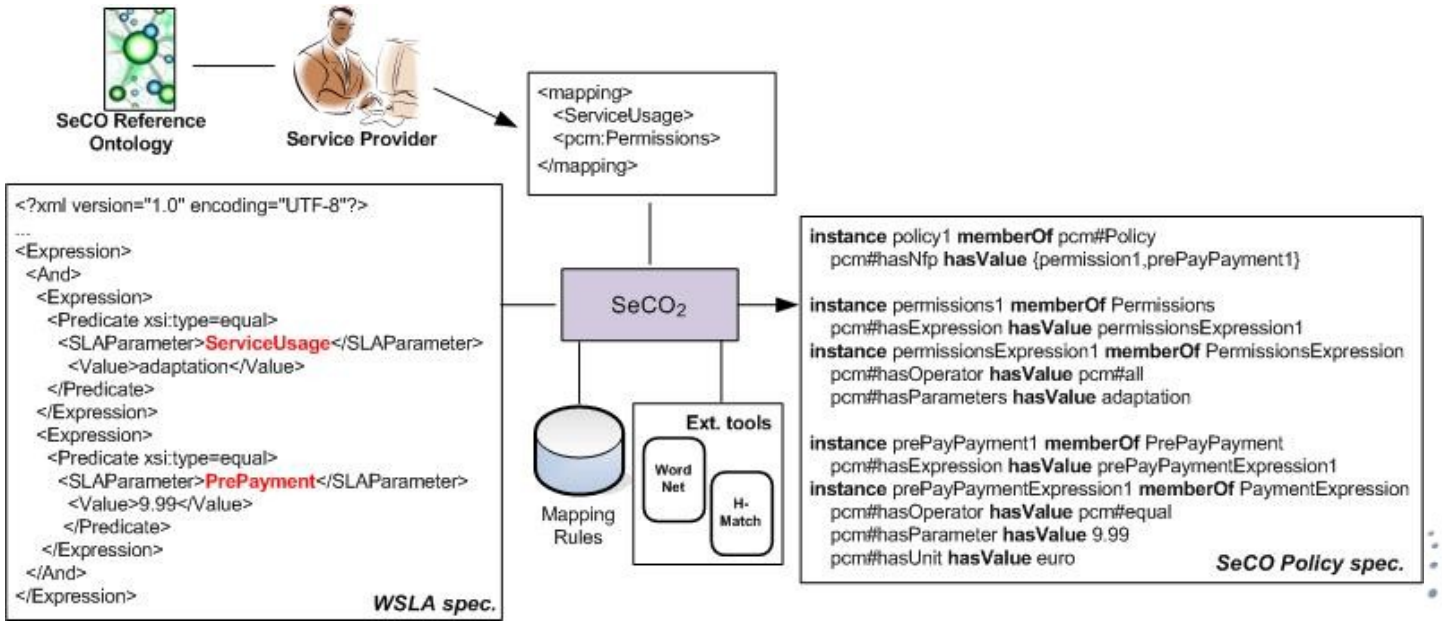
SeCO Reference Ontology and SeCO Policies

- *SeCO Reference Ontology* and *SeCO Policies*
  - built on the Policy Centered Meta-model (PCM) [DePaoli08].
- *SeCO Reference Ontology*
  - built applying general modeling rules to profile models;
  - defines expressive descriptions of contract properties.
- *SeCO Policies*
  - represent service contracts defined as clusters of contract property istances.

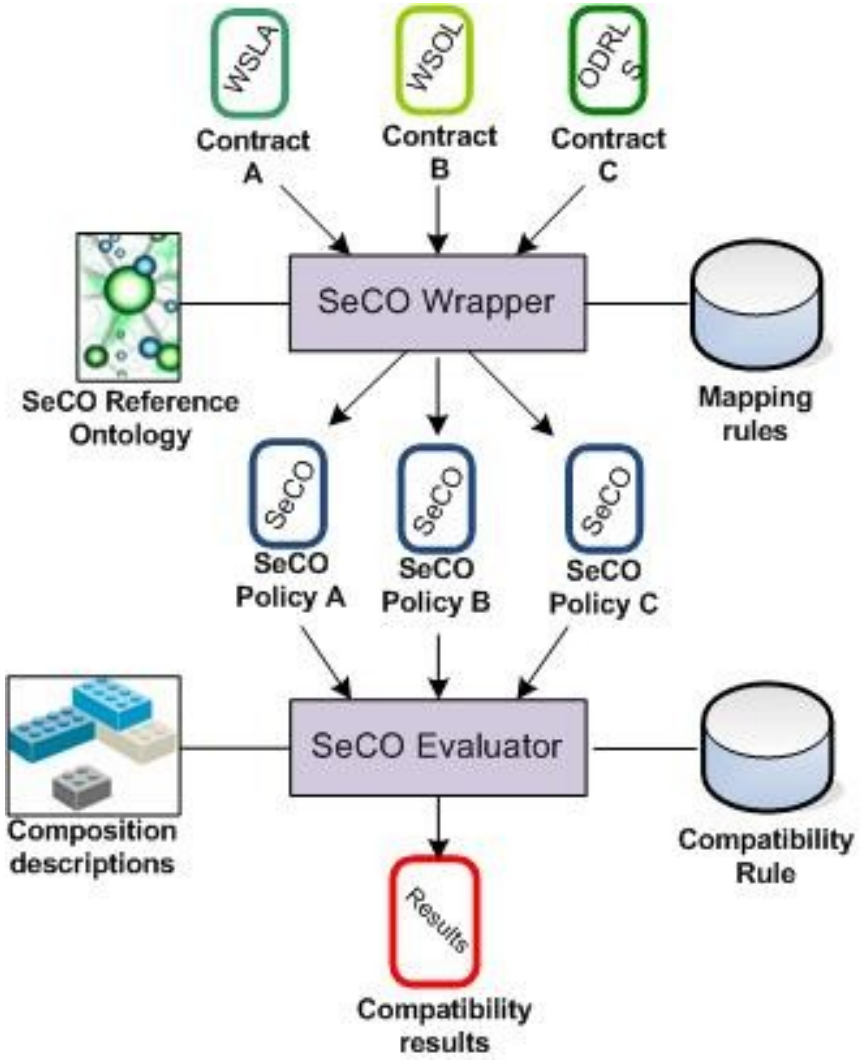


Mapping Service Contracts

- A proper technique for each type of language
  - Specifications in **Type A** are wrapped applying fixed mapping rules.
  - Specifications in **Type B** and **Type C** can require interactions with service providers to handle the absence of knowledge (i.e., mapping rules).
    - The definition of new mapping rules is supported by lexical databases and ontology alignment tools.



Evaluating Service Contract Compatibility: activities and flows



Service Contract Mapping

Service Contract Compatiblity Evaluation (at the service level as a whole)

Evaluating Service Contract Compatibility

- ***Problem:*** evaluation of contract compatibility in a service composition.
- Input:
  - service composition description in terms of data and control flows;
  - contracts of the services involved in the composition.
- Output:
  - compatible/incompatible service contract properties.

- The compatibility is checked considering
  - semantic relations among values associated with qualitative contract properties;
  - constraint operators used to define quantitative contract properties;
  - data and control flows of the service composition.

Compatibility Evaluation Rules

Property	*Type*	*Data Flow*	*Control Flow*	*Rule*
Location	Service Context			Partnership
Pricing	Business	X		Compatible value list
Payment (for data usage)	Business	X		Binary, Ternary
Payment (for service usage)	Business		X	Binary, Ternary
Scalability	QoS		X	Binary, Ternary
Permissions	License		X	Subsumption
Data Ownership	License	X		Compatible value list

Evaluating Service Contract Compatibility

Algorithm 1 Compatibility Evaluation

```
1: for all s_i ∈ S do
2:   for all s_j ∈ S(j ≠ i) do
3:     Ω(s_i, s_j) = φ where Ω(s_i, s_j) is a set of triples [p_w, p_z, λ(p_w, p_z)]
4:     for all p_w ∈ P(s_i) do
5:       for all p_z ∈ P(s_j) do
6:         λ(p_w, p_z) = φ, where λ(p_w, p_z) is a set of triples [pr_i, pr_j, result]
7:         ϒ(p_w, p_z) = φ, where ϒ(p_w, p_z) is a set of comparable properties [pr_1, pr_2]
8:         ϒ(p_w, p_z) = Matching(p_w, p_z)
9:         for all [pr_1, pr_2] ∈ ϒ(p_w, p_z) do
10:          rule = Extract(pr_1.name)
11:          if pr_1.type =' CF − inf' then
12:            λ(p_w, p_z) = λ(p_w, p_z) ∪ EvalRuleF(rule, pr_1, pr_2, cf_j ∈ CF(s_i))
13:          else
14:            if pr_1.type =' DF − inf' then
15:              λ(p_w, p_z) = λ(p_w, p_z) ∪ EvalRuleF(rule, pr_1, pr_2, df_j ∈ DF(s_i))
16:            else
17:              λ(p_w, p_z) = λ(p_w, p_z) ∪ EvalRule(rule, pr_1, pr_2)
18:            end if
19:          end if
20:        end for
21:        Ω(s_i, s_j) = Ω(s_i, s_j) ∪ [p_w, p_z, λ(p_w, p_z)]
22:      end for
23:    end for
24:  end for
25: end for
```

For all SeCO Policy couples

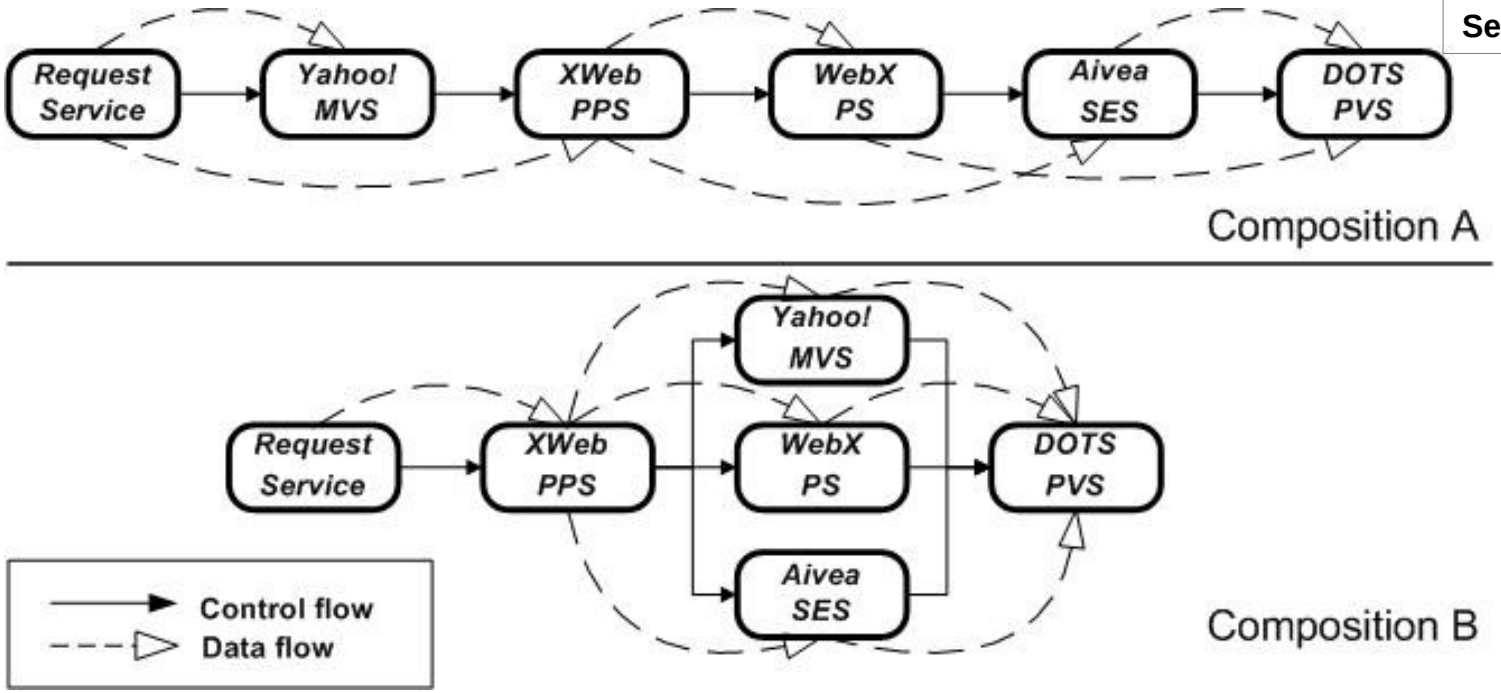
Identify comparable SeCO properties

Extract the evaluation rule

Evaluate according to flow influences

Illustrating Example

Purchase Data Analysis Service



	Data Ownership	Scalability
Request Service	Personal-use	100 tr/min
Yahoo! MVS	Copyrighted	100 tr/min
XWeb PPS	Free-distribution	100 tr/min
Aivea SES	Free-distribution	100 tr/min
WebX PS	Free-distribution	500 tr/min
DOTS PVS	Free-distribution	500 tr/min

Illustrating Example

- *Data Ownership* :
  - a *License* term stating how the data are protected;
  - influences the *data flow* of the service composition;
  - assumes values characterized by relations of compatibility/incompatibility
    - *copyrighted* is compatible with *personal-use*
    - *copyrighted* is incompatible with *free-distribution*
- *Scalability* :
  - a *QoS* term indicating the maximum number of transactions accepted per minute.
  - influences the *control flow* of the service composition;
  - assumes numeric values.

Illustrating Example

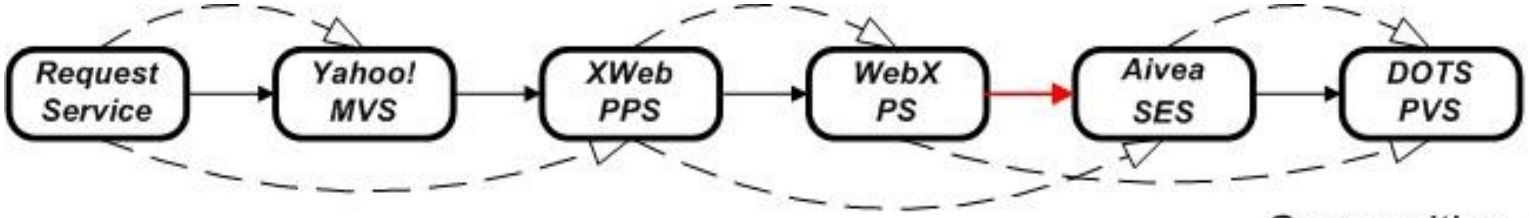
- *Data Ownership is evaluated exploiting the axiom:*

```
axiom dataOwnershipCompatibility
    definedBy
        compatible ( ?X , ?Y) :−
                ( ?X memberOf seco#DataOwnValue) and
                ( ?Y memberOf seco#DataOwnValue) and
                seco#compatible( ?X, ?Y)
```

- *Scalability is evaluated applying the algorithm*

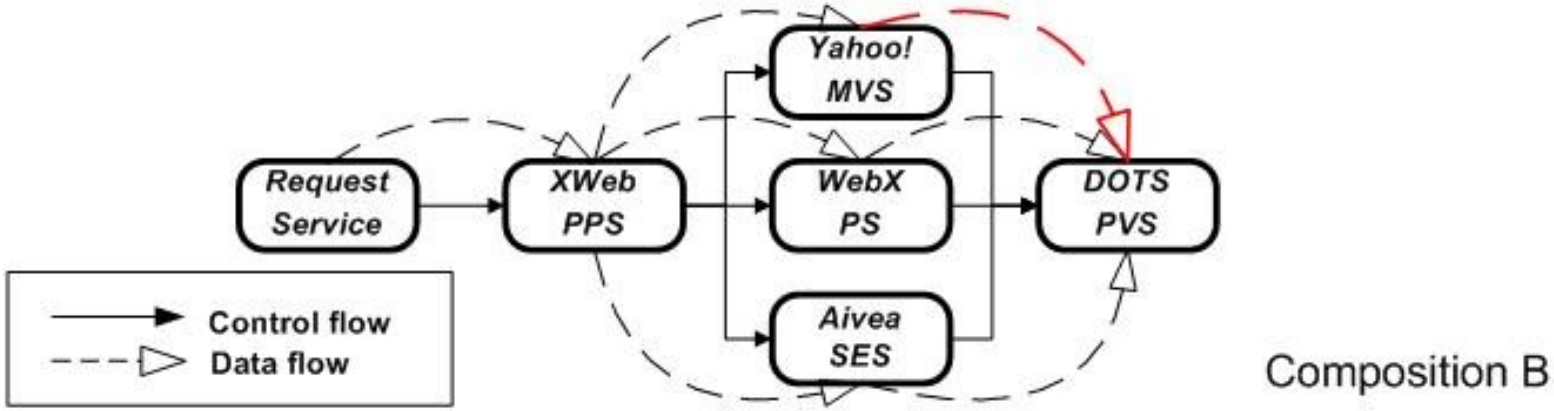
```
Given pr1,pr2
if(([pr1,pr2].equals("seq"))||([pr1,pr2].equals("par"))){
        if(pr2.value<pr1.value)
                result = "INCOMPATIBLE";
        else
                result = "COMPATIBLE"; }
```

Illustrating Example



Composition A

Service Contract Compatibility			
Property	**Caller Service**	**Callee Service**	**Incompatibity**
Scalability	WebX PS	Aivea SES	**500 tr/min** not compatible with **100 tr/min**



Composition B

Service Contract Compatibility			
Property	**Caller Service**	**Callee Service**	**Incompatibity**
Data Ownership	Yahoo! MVS	DOTS PVS	**Copyrighted** not compatible with **Free-distribution**

Service Information Overloading

Problems

- Too much service information pushed to the composition developer and (semi-)automatic service composition algorithms
- Our solutions
  - Use quality of data metrics to characterize service information
  - Filter service information based on consumers' requests.

Metrics

Interpretability specifies the availability of documentation and metadata for correct interpretation of service information

Category	Service information	Examples
schema	conceptual service and data schemas	WSDL, SAWSDL, pre/post conditions, data models
documentation	documents	APIs explanation, best practices
NFP	non-functional properties	categorization, location, QoS information
contract	service contracts and contract templates	service level agreements based on NFPs
provenance	Provenance information	versioning of schemas, NFPs, contracts

$$Interpretability = \frac{\sum (score(category_i) \times w_i)}{\sum w_i}$$

VITALAB Distributed Systems Group

Metrics

- Completeness specifies the ratio of missing values of provided NFP information, NFP_p to the expected minimum set of NFPs, NFP_{min}

$$Completeness = 1 - \frac{\|NFP_p \cap NFP_{min}\|}{NFP_{min}}$$

- Timeliness specifies how current a non-functional property is.

$$Timeliness = 1 - \left(\frac{Age}{ExpectedLifetime}, 1 \right)$$

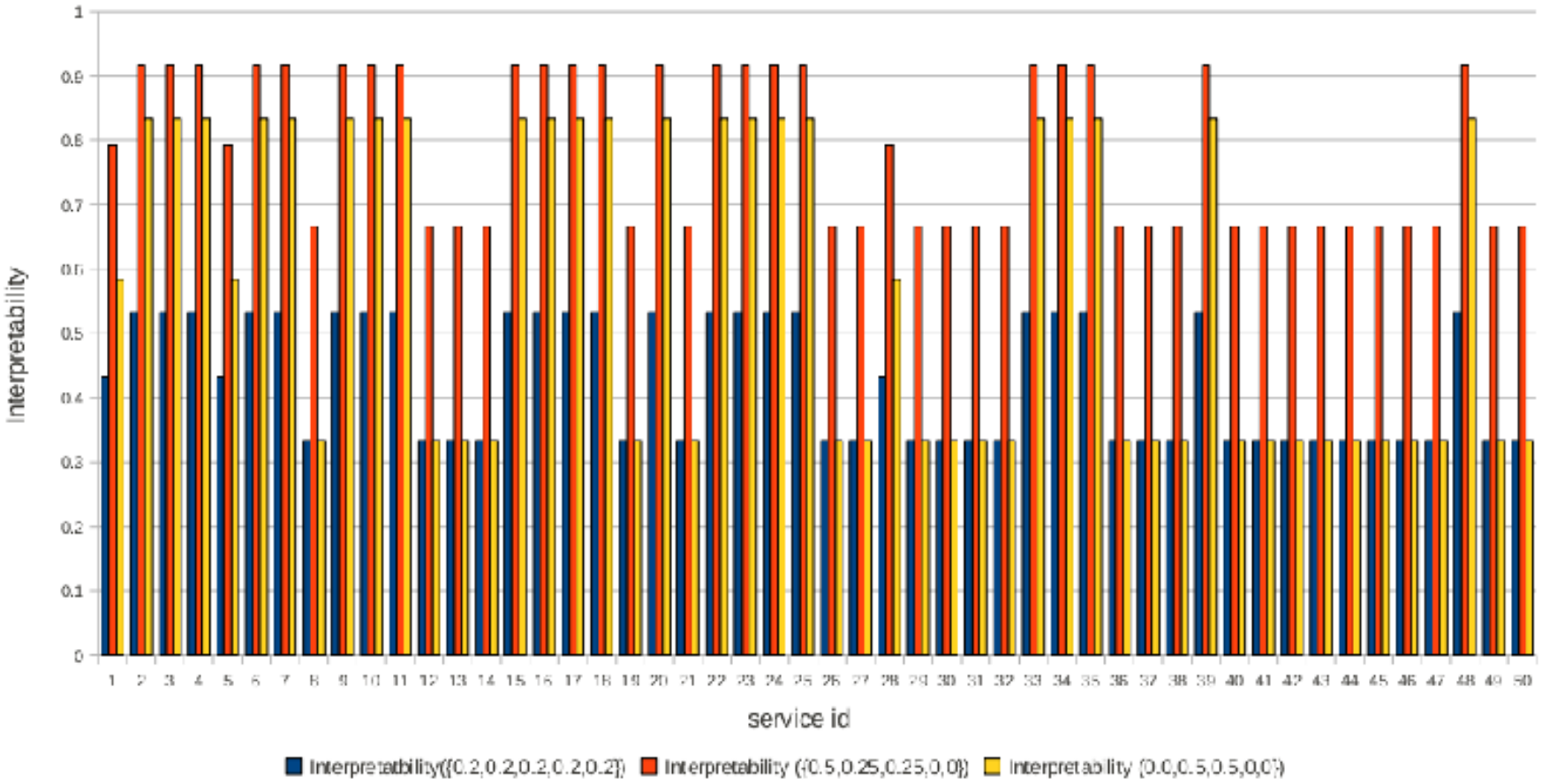
Filtering mechanism

- Two types of filtering
  - Interpretability and NFPs.
- NFP-based filtering:
  - Step 1: Extract and establish NFPmin and ExpectedLifetime from the developer's requirement;
  - Step 2: Evaluate QoD metrics, e.g., Completeness and T imeliness;
  - Step 3: Establish filtering thresholds based on QoD metrics;
  - Step 4: Eliminate services whose information does not meet conditions setup in Step 3;
  - Step 5: Refine the filtering by repeating Step 3.

Example 1: Weather Services

- Using Seekda! 50 weather services
  - service interface, information about documentation, availability, user rating, etc.
- Data preparation
  - score(schema) = 1 as their schemas are basically a WSDL file.
  - Seekda!'s {none, partially, good} =score(documentation) = {0, 0.5, 1}.
  - We assumed NFPmin ={availability, reliability, responsetime} whereas seekda! provides only availability and response time.
  - Provenance information and service contract are missing.

Filtering Data



Example 2: Service Contracts

- Having 100 idential services, each has 5 service contracts
  - 500 WSML service contracts in the PoliMaR framework (http://polimar.sourceforge.net/)
- Looking for a shipping service able to satisfy specified conditions on
  - payment method, payment deadline, insurance, base price and hours to delivery.
  - No older than 1 year before 19 June 2010
- Using following parameters:
  - NFPmin ={payment method, payment deadline, insurance, base price, hours to delivery}
  - ExpectedLifetime = 1year.

Filtering Evaluation

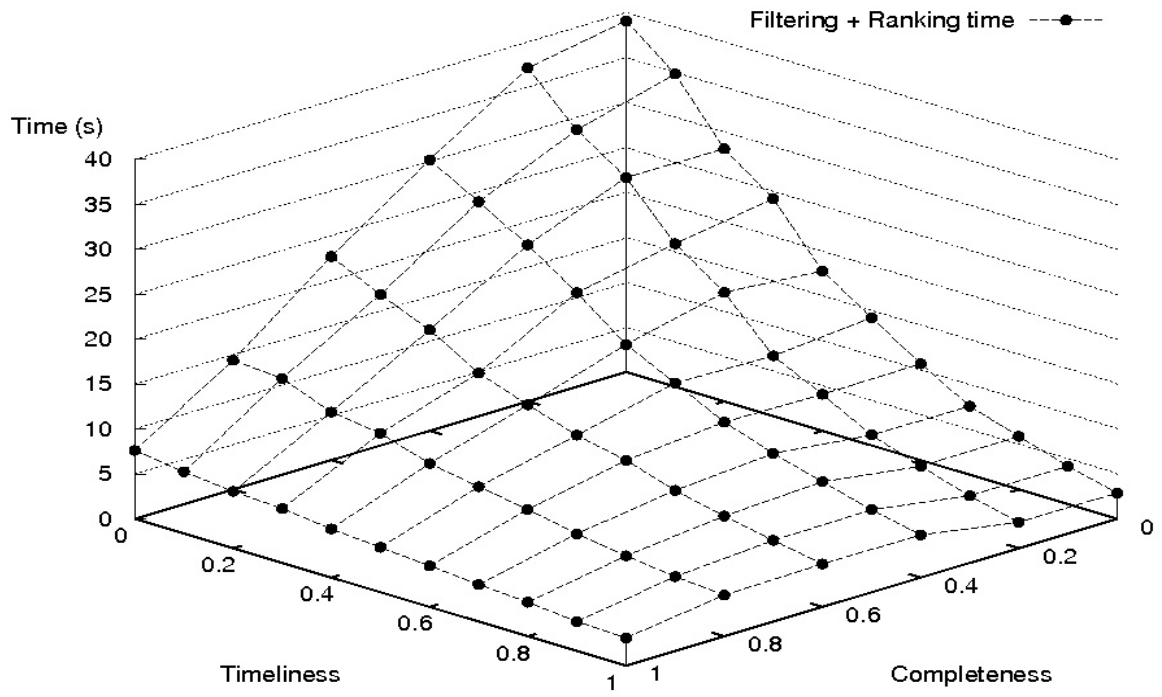
- Performance evaluation with threshold: Completeness ≥ 0.6 and Timeliness > 0.2.

	Filter 1	Filter 2	Filtered Contracts	Filtering Time	Ranking Time	Total Time
Exp. 1	no	no	500	0 sec	37.5 sec	37.5 sec
Exp. 2	yes	no	309	2.7 sec	19.9 sec	22.6 sec
Exp. 3	no	yes	395	2.2 sec	25.9 sec	28.1 sec
Exp. 4	yes	yes	246	3.5 sec	14.2 sec	17.7 sec

Table 4. Results of applying Completeness (Filter 1) and Timeliness (Filter 2) filtering activities.

Filtering Evaluation

- With thresholds= {0, 0.2, 0.4, 0.6, 0.8, 1} which are equivalent to {not required, optional, preferred, strong preferred, required, strict required }



Further Readings

- Hong-Linh Truong, Schahram Dustdar "**On Evaluating and Publishing Data Concerns for Data as a Service**", August 2010. On submission.
- Michael Mrissa, Salah-Eddine Tbahriti, Hong-Linh Truong, **"Privacy model and annotation for DaaS"** , July, 2010. On submission.
- Hong-Linh Truong, Marco Comerio, Andrea Maurino, Schahram Dustdar, Flavio De Paoli and Luca Panziera **"On Identifying and Reducing Irrelevant Information in Service Composition and Execution"**, June, 2010. On submission.
- Atif Manzoor, Hong-Linh Truong, Schahram Dustdar, **"Quality of Context: Models and Applications for Context-aware Systems in Pervasive Environments"**, Special issue on Web and Mobile Information Services, Editors: Ghita Kouadri Mostefaoui, Muhammad Younas, Patrick Brezillon, The Knowledge Engineering Review. To appear.
- Hong-Linh Truong, Schahram Dustdar, Andrea Maurino, Marco Comerio, **"Context, Quality and Relevance: Dependencies and Impacts on RESTful Web Services Design"**, Second International Workshop on Lightweight Integration on the Web (CommposableWeb's 2010), ICWE 2010, (c)Springer-Verlag, July 5-9, 2010, Vienna, Austria.
- Hong-Linh Truong, Schahram Dustdar **"On Analyzing and Specifying Concerns for Data as a Service"**, The 2009 Asia-Pacific Services Computing Conference (IEEE APSCC 2009), (c) IEEE Computer Society, December 7-11, 2009, Biopolis, Singapore.
- Marco Comerio, Hong-Linh Truong, Flavio De Paoli, Schahram Dustdar, **" Evaluating Contract Compatibility for Service Composition in The SeCO2 Framework "** The 9th International Conference on Service Oriented Computing (ICSOC 2009), (c) Springer-Verlag, November 24 - 27, 2009, Stockholm, Sweden.

Conclusions

- We present
  - Data concern specification, evaluation and publishing
  - Some applications of data concerns in service composition and execution
  - Data concerns are important for Web mashup, DaaS in the cloud, as well as Sensor-as-a-service
- Several examples are based on quality of data
  - But in principle it can also be applied to data privacy and other concerns
- But there are still several fragment research results that need to be integrated

Future Work

- Open issues
  - Adaptive algorithms based on data concerns are open
  - Evaluation of data concerns is challenging, especially for domain specific data concerns
  - The dependency among data concerns and service concerns
- Integration and experiment
- Data contract

Which topics we can collaborate? And how?

Thanks for your attention!

Hong-Linh Truong

Distributed Systems Group
Vienna University of Technology

Austria

truong@infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at