Data marketplaces: core models and concepts

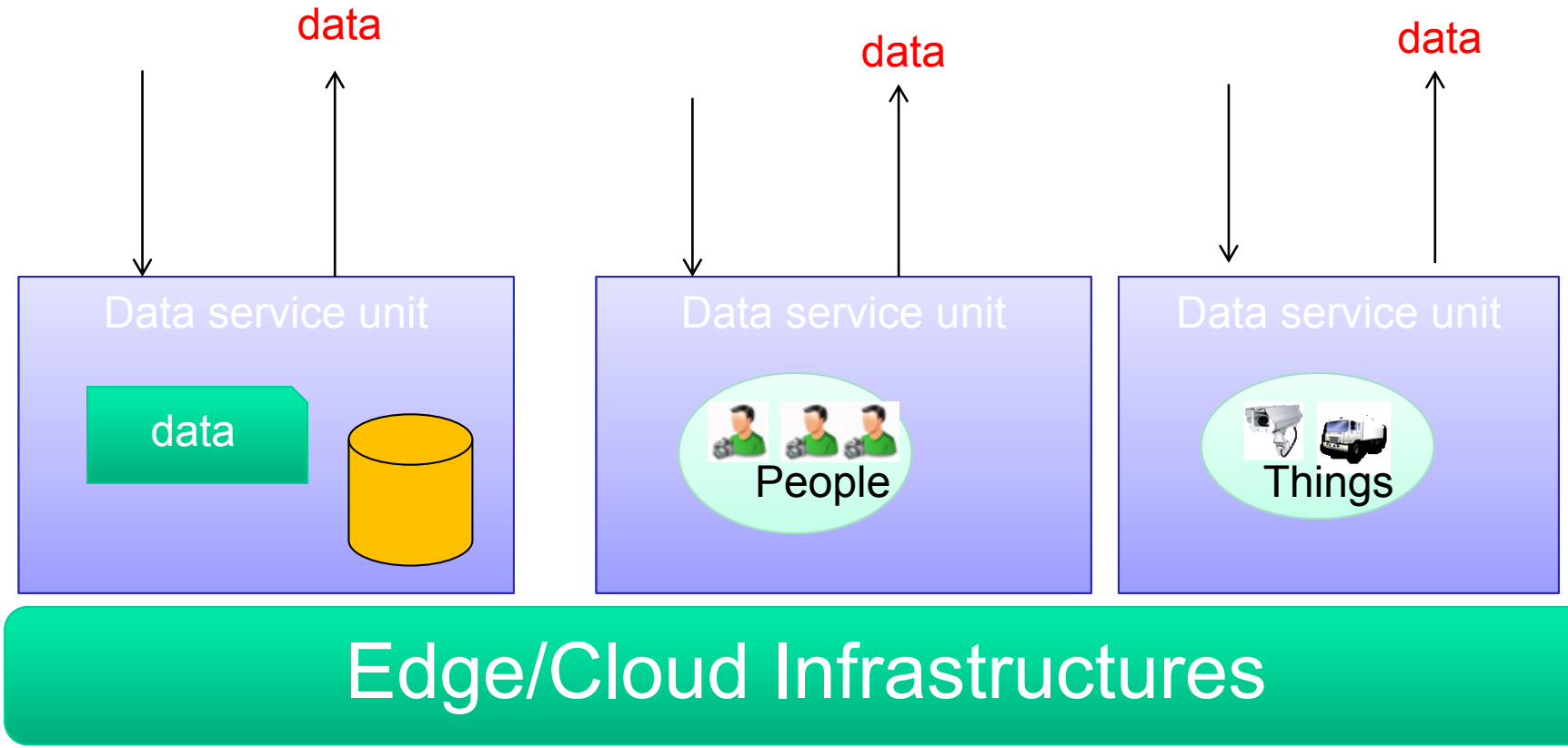
Hong-Linh Truong
Distributed Systems Group, TU Wien

truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong
@linhsolar

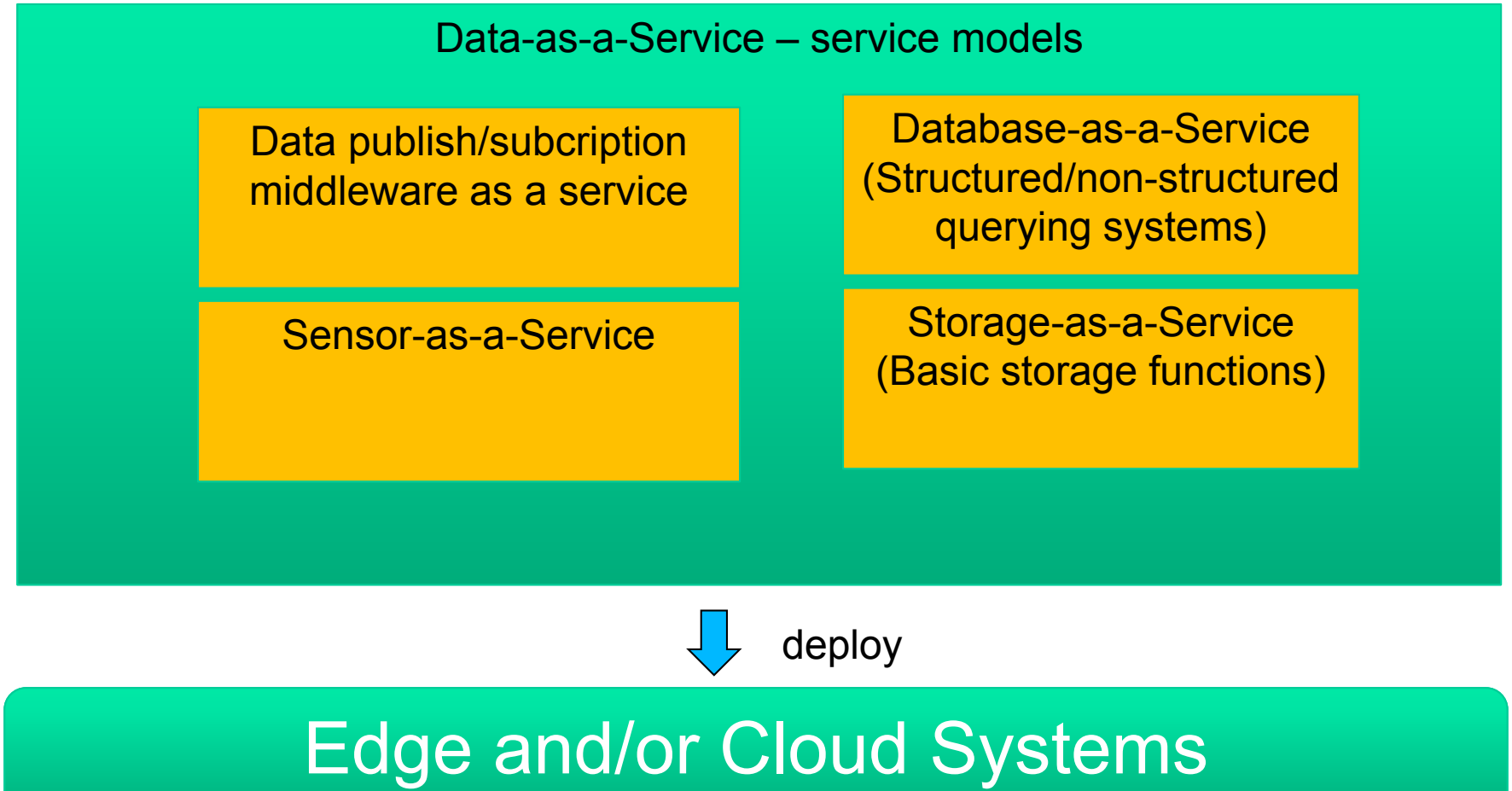
Outline

- Data marketplaces
- Description models
- Data contract model and evaluation
- Data contract exchange models and architectures
- Case study – design a data marketplace

Recall – data service units in clouds/internet

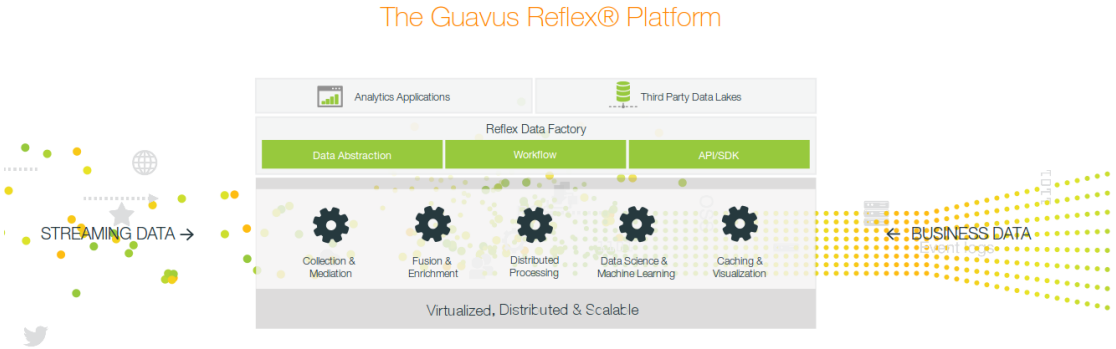


Recall – data as a service

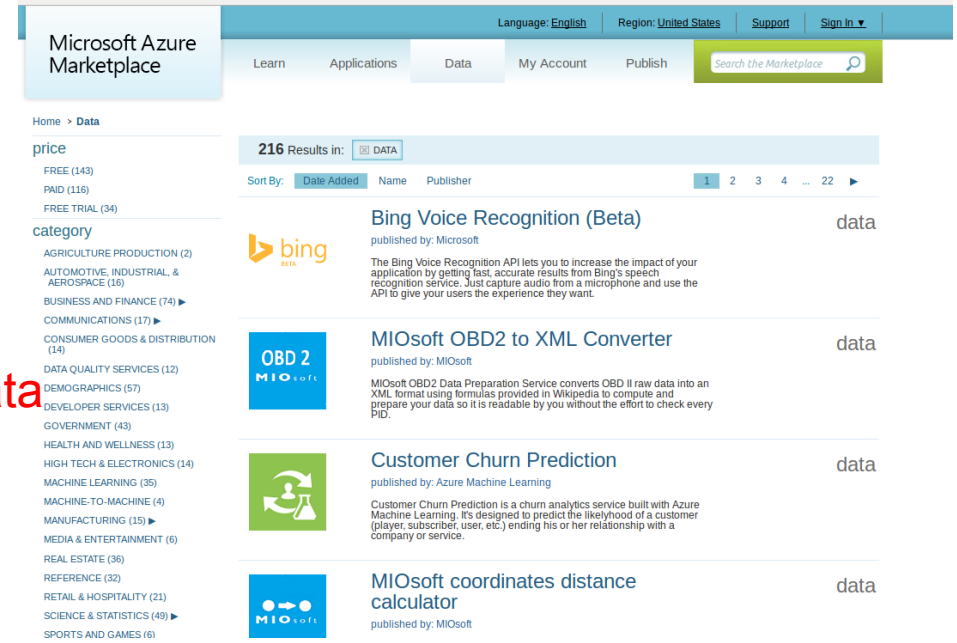


Data platform or marketplace?

http://www.guavus.com/platform/



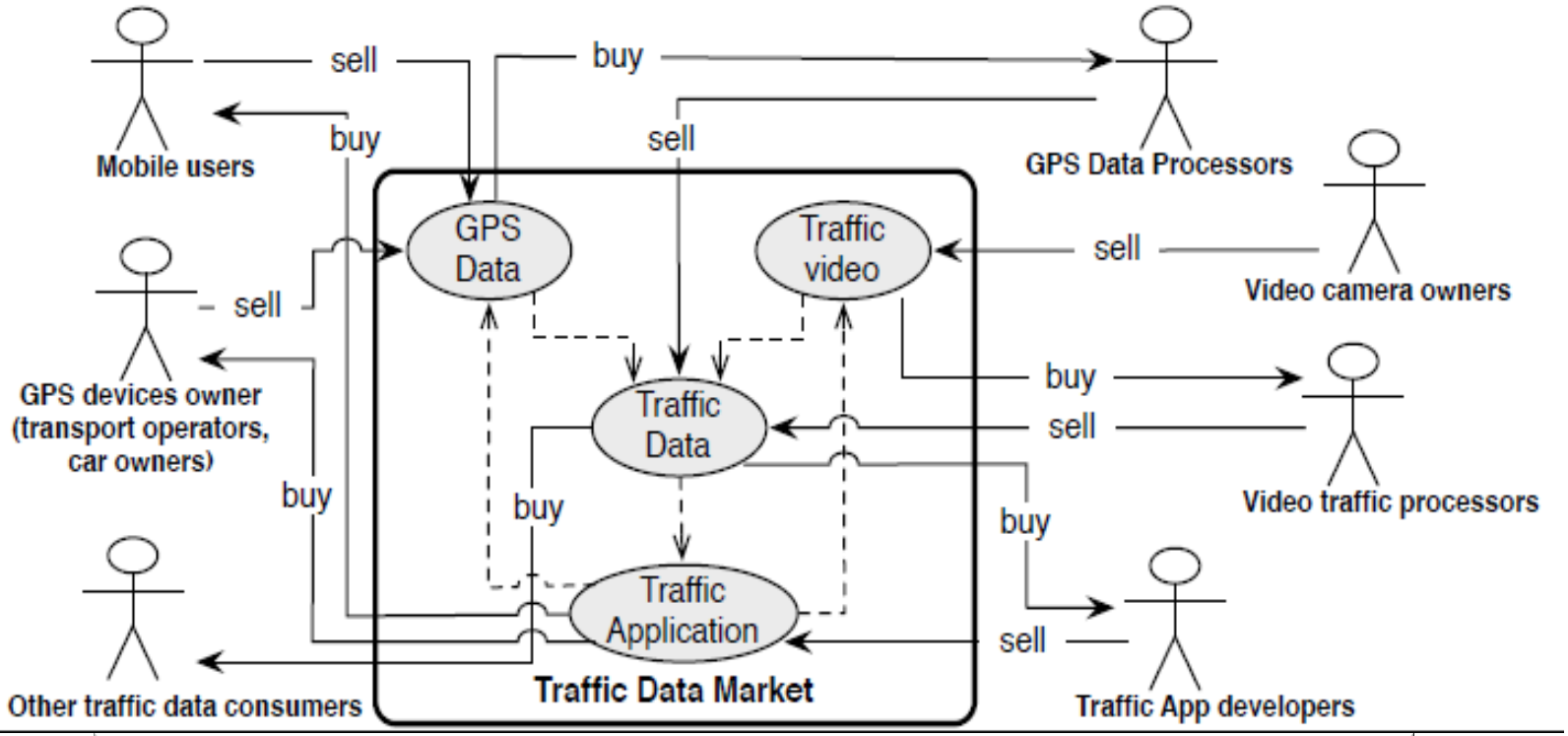
http://datamarket.azure.com/browse/data



Data marketplaces

- More than just DaaS
  - DaaS focuses on data provisioning features
- Stakeholders in data marketplaces
  - Multiple data providers and consumers
  - Marketplace providers
  - Marketplace authorities
  - Analytics providers
  - Data transportation providers
  - Billing and payment providers

Example of stakeholders



Tien-Dung Cao, Quang-Hieu Vu, Duc-Hung Le, Hong-Linh Truong, Schahram Dustdar: MARSA: A Marketplace for Realtime Human-Sensing Data.
http://dungcao.github.io/marsa/

Specific data market (Tokyo Tsukiji) or generic data market (Donau Zentrum)?

Technical services, protocols, mechanisms in data marketplaces

- Multiple DaaS provisioning
  - Access models and interfaces
- Complex interactions among DaaS providers, data providers, data consumers, marketplace providers, etc.
  - Data exchange as well as payment
- Complex billing and pricing models
- Market dynamics
- Service and data contracts

Some important issues

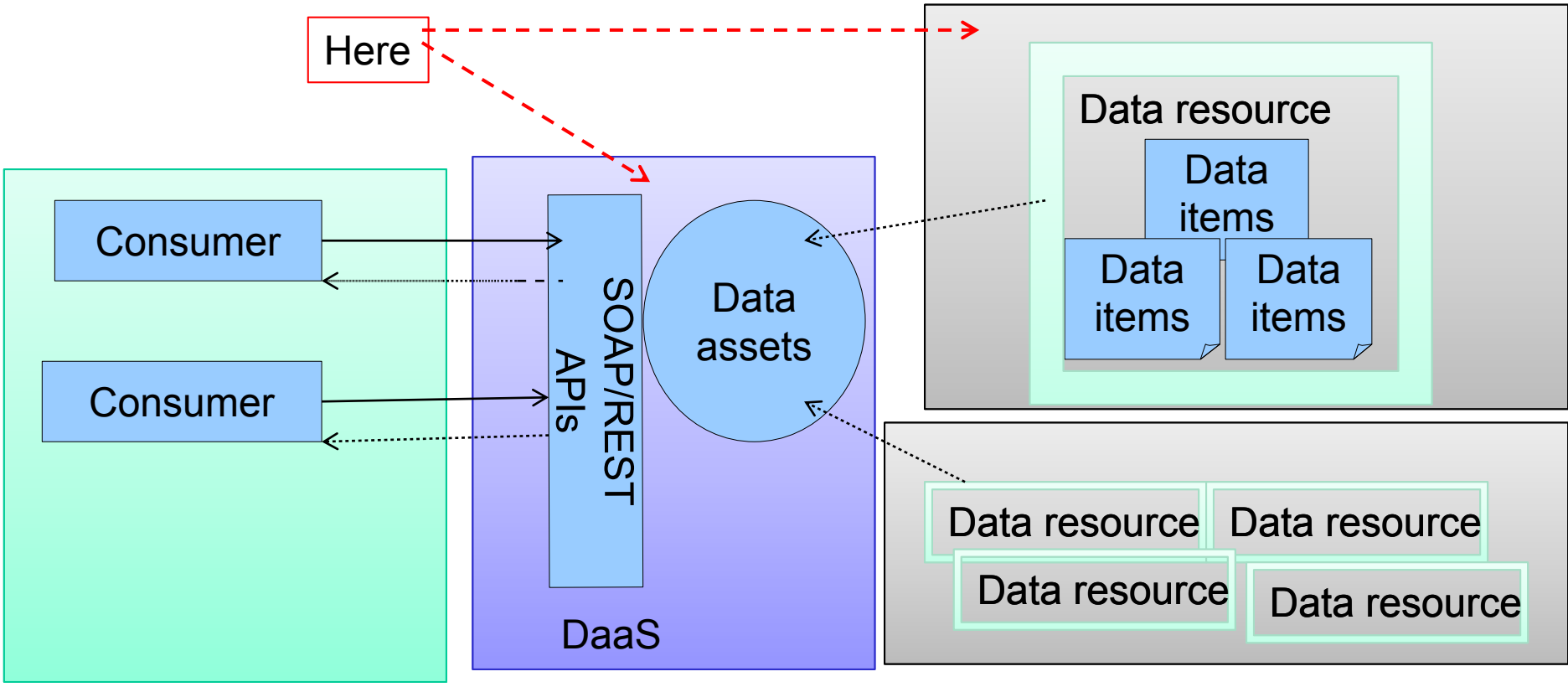
DAAS DESCRIPTION MODEL

DATA CONTRACT

DATA CONTRACT EXCHANGE

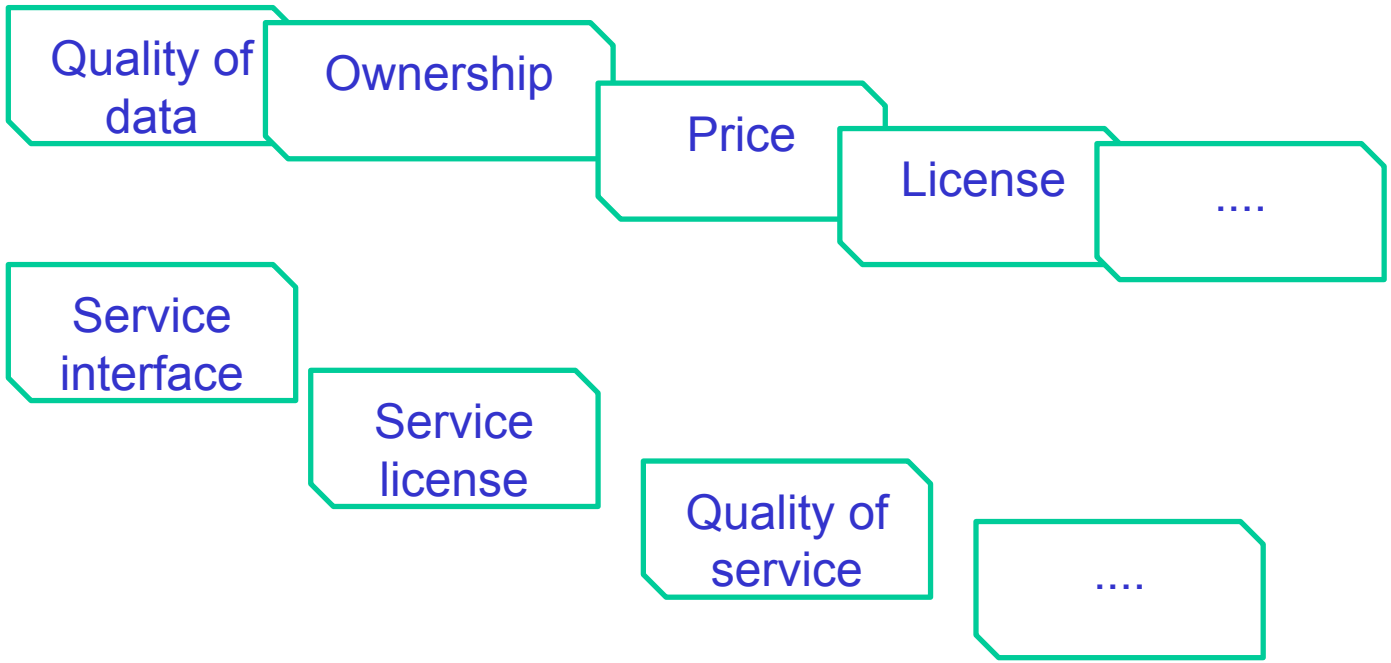
Description Model for DaaS (1)

Which levels must be covered?



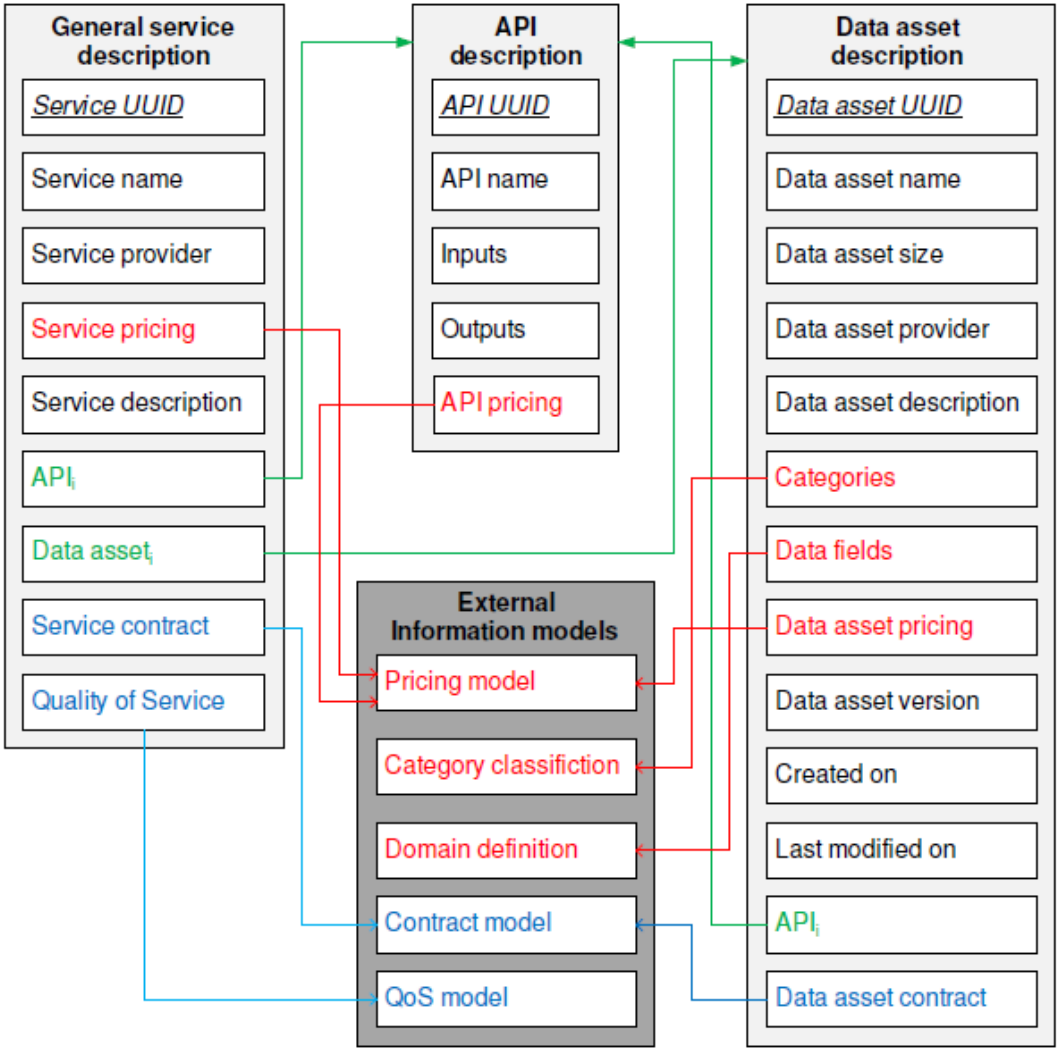
Description model for DaaS – types of information

Which types of information must be covered?



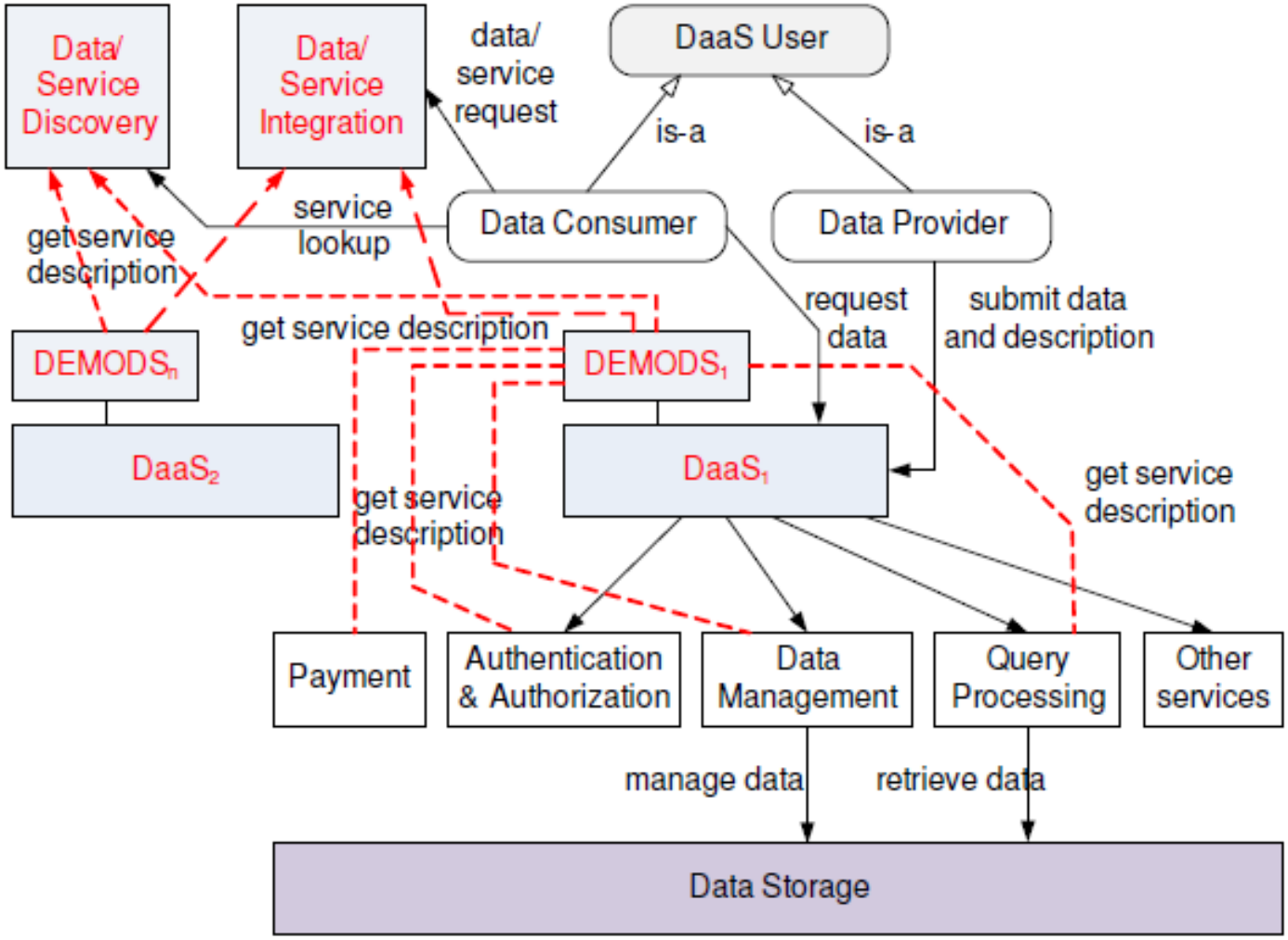
DEMOS – a description model for Data-as-a-Service

Quang Hieu Vu, Tran Vu Pham, Hong Linh Truong,, Schahram Dustdar, Rasool Asal: DEMODS: A Description Model for Data-as-a-Service. AINA 2012: 605-612



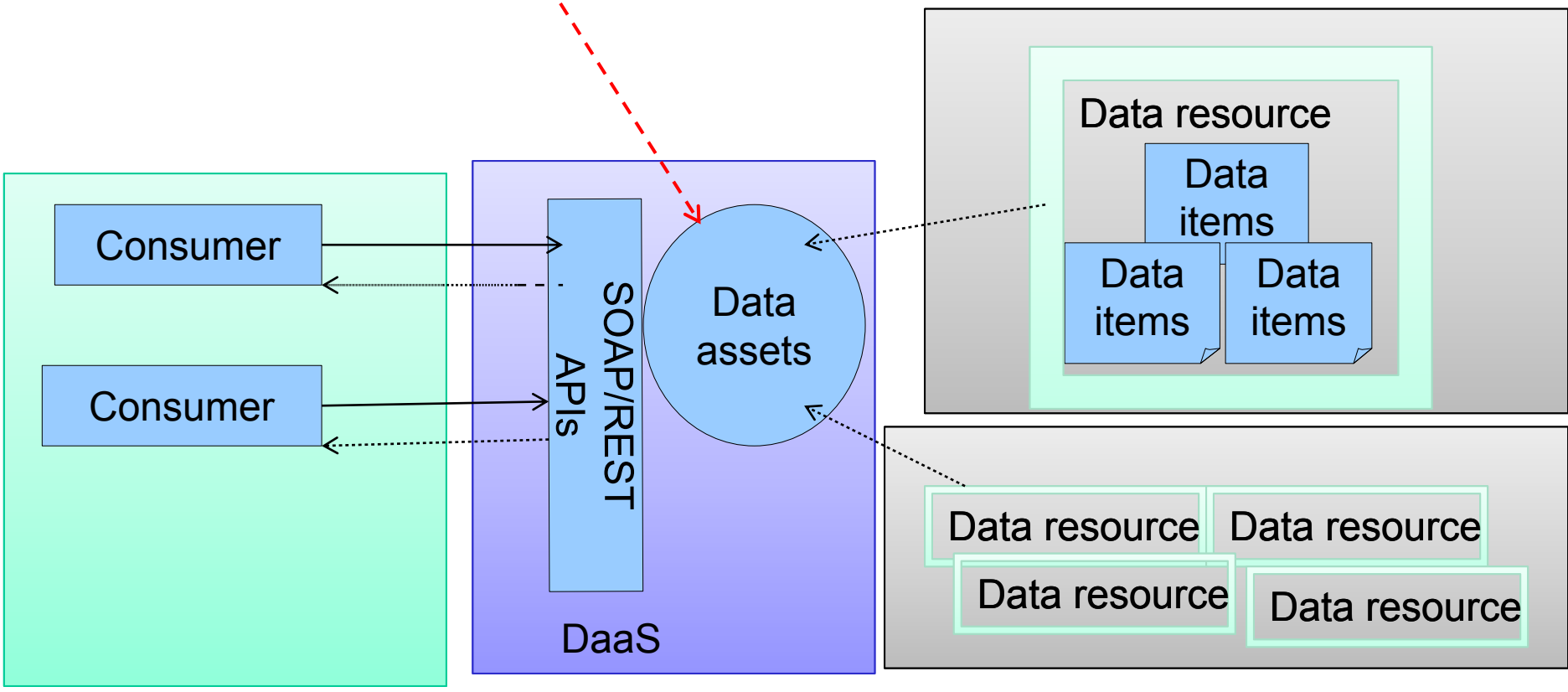
See prototype:
http://www.infosys.tuwien.ac.at/prototype/SOD1/demods/

Data marketplaces and related components/services



Data contract

How to specify data contract?



Data contracts

- Give a clear information about data usage
- Have a remedy against the consumer for illegal data usage
- Limit the liability of data providers in case of failure of the provided data;
- Specify information on data delivery, acceptance, and payment

Data contracts

- Well-researched contracts for services but not for DaaS and data marketplaces
  - But **service APIs != data APIs =! data assets**
- Several open questions
  - Right to use data? Quality of data in the data agreement? Search based on data contract? Etc.

→ Require extensible models
  - → Capture contractual terms for data contracts
  - → Support (semi-)automatic data service/data selection techniques.

Hong-Linh Truong, Marco Comerio, Flavio De Paoli, G.R. Gangadharan, Schahram Dustdar, "**Data Contracts for Cloud-based Data Marketplaces** ", International Journal of Computational Science and Engineering, 2012 Vol.7, No.4, pp.280 - 295

Study of main data contract terms

- Data rights
  - Derivation, Collection, Reproduction, Attribution
- Quality of Data (QoD)
  - Not mentioned, Not clear how to establish QoD metrics
- Regulatory Compliance
  - Sarbanes-Oxley, EU data protection directive, etc.
- Pricing model
  - Different models, pricing for data APIs and for data assets
- Control and Relationship
  - Evolution terms, support terms, limitation of liability, etc

Most information is in human-readable form

Data contract studv

Table 1 Example of data contracts in real-world DaaS

Contracts	*Data rights*					*Quality of data*		*Compliance*	*Pricing model*		*Control and relationship*			
	Derivation	*Collection*	*Reproduction*	*Attribution*	*Non-commercial use*	*Completeness*	*Accuracy*		*Transaction*	*Subscription*	*Warranty*	*Indemnity*	*Liability*	*Laws, Jurisdiction*
AvianKnowledge.net (AKN, 2011)				+	+				+	+	+	+	+	
Building model products (BMP, 2011)		+			+			+		+	+	+	+	+
Creative common: attribution-ShareAlike 2.0 Generic (CCAS, 2011)	+	+	+	+							+	+	+	+
Consumer expenditure data (CED, 2011)		+			+			+		+	+	+	+	+
Freebase data dump (FDD, 2011)	+	+	+	+							+	+	+	+
GBIF Data Usage Agreement (GBIF, 2011)	+	+	+	+				+			+	+	+	+
Infochimps Twitter Census: Stock Twittes (TCST, 2011)	+	+						+	+		+		+	+
Open Data Commons Attribution License (ODCAL, 2011)	+	+	+	+							+	+	+	
Open Government License (OGL, 2011)	+	+	+	+									+	
US Consumer Price Index: 1913 to current (USCPI, 2011)		+			+			+		+	+	+	+	+

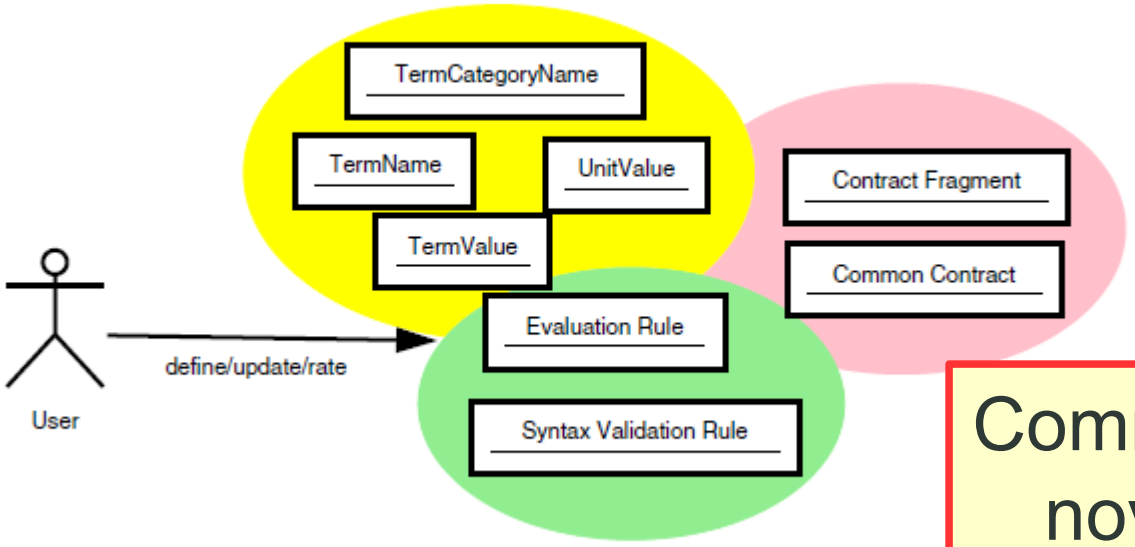
Hong-Linh Truong, Marco Comerio, Flavio De Paoli, G.R. Gangadharan, Schahram Dustdar, **"Data Contracts for Cloud-based Data Marketplaces "**, International Journal of Computational Science and Engineering, 2012 Vol.7, No.4, pp.280 - 295

Developing data contracts in cloud-based data marketplaces

- Follow community-based approach for data contract
- Propose generic structures to represent data contract terms and abstract data contracts
- Develop frameworks for data contract applications
- Incorporate data contracts into data-as-a-service description
- Develop data contract applications

Community view on data contract development

- Community users can develop:
  - Term categories, term names, values, and units
  - Rules for data contracts
  - Common contract and contract fragments



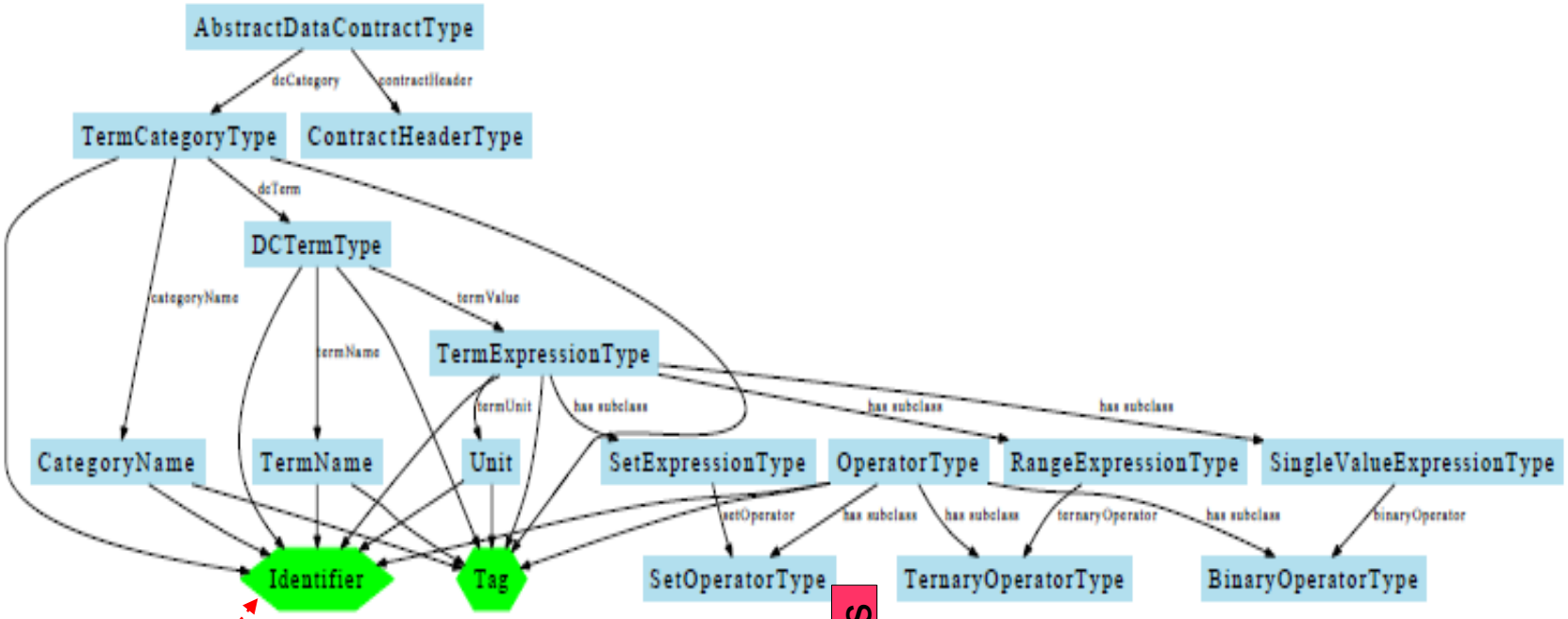
Community users =! novice users

Representing data contract terms

- Contract term: (termName,termValue)
  - Term name: common terms or user-specific terms
  - Term value: a single value, a set, or a range

Category	*Term representation*	*Examples*
Data rights	$termName = \{val_1, val_2, \cdots, val_n\}$	$termName$={Derivation, Collection, Reproduction, Attribution, Noncommercialuse}, val_i = {Undefined, Null, Allowed, Required, True, False}
Quality of data	$val_l \leq termName \leq val_u$	$termName$={Accuracy, Completeness, Uptodateness}, val_l and $val_u \in [0,1]$
Compliance	$termName = \{val_1, val_2, \cdots, val_n\}$	$termName$ and val_i are any string, e.g., $termName$={PrivacyCompliance} and $termValue$={Sarbanes-Oxley (SOX) Act}
Pricing model	$termName = (cost = val_1, usagetime = val_2, , maximumuse = val_3)$	$termName$ is any string, e.g., MonthlyPayment; $val_1 \in R$, e.g., $cost = 50$ €, $val_2 = \{(end_t - start_t); UNLIMITED\}$ where $end_t, start_t \in datetime$, e.g., $usagetime = 30$ days; $val_3 \in N$, e.g., $maximumuse = 1,000$ calls
Control and relationship	$termName = val$	$termName$ and val are any string, e.g., $termName$={Liability, LawandJurisdiction} and $val = \{US, Austria\}$

Structuring abstract data contracts



Use Identifiers and Tags for identifying and searches

generates

Concrete data contracts can be in RDF, XML or JSON

Development of contract applications

- Main applications:
  - Data contract compatibility evaluation, data contract composition
- Some common steps
  - Extract DCTermType in TermCategoryType
    - Extact comprable terms from all contracts,
      - e.g., dataRight: Derivation, Composition and Reproduction
  - Use evaluation rules associated with DCTermType from rule repositories
  - Execute rules by passing comparable terms to rules
  - Aggregate results

Evaluating Data Contracts

- Goal

  Check the quality and reputation of a data contract
- We can check data contracts using quality of data metrics
  - Timeliness, Completeness, Reputation, Consistency metrics
- Examples
  - Free-per-use but cost = 100EUR
  - Missing „data accuracy“ concern

Data Contract Compatibility

- Goal

  If multiple data contracts are compatible with the consumer needs

  → The consumer requires multiple data associated with different contracts

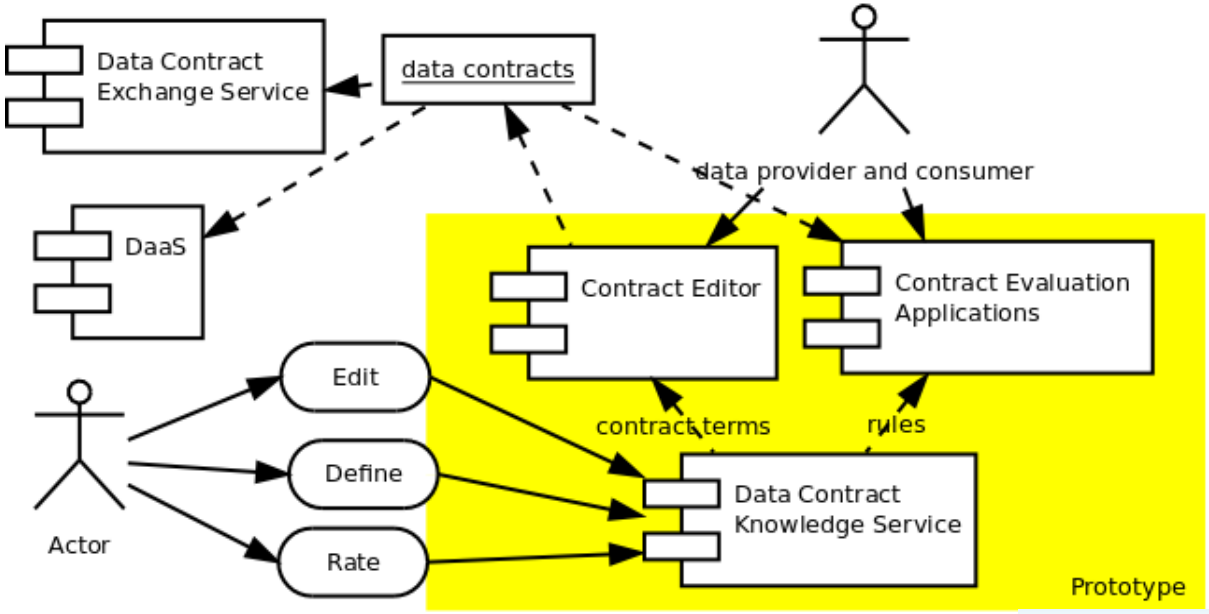
- Contract compatibility
  - Matching contract terms
  - Evaluating contract term compatibility and completeness w.r.t. application needs
  - Making decision in using data

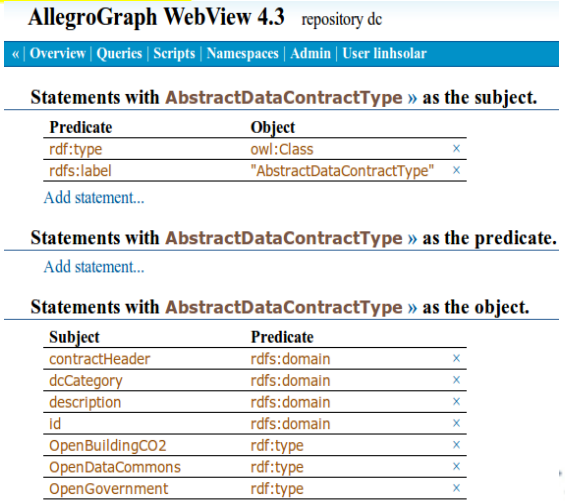
Example of contract compatibility evaluation

Table 3 Possible steps in making decisions based on contract compatibility evaluation

	Action	*Case*	*Solution*
Compatibility = 1	Check reputation, consistency and timeliness	Reputation = LOW	Change the data composition. Substitute the data provided by the untrustworthy DaaS.
		Consistency = LOW	Interact, if possible, with the DaaS providers to solve inconsistent contract terms.
		Timeliness = LOW	Interact, if possible, with the DaaS providers to update contract terms.
Compatibility < 1	Check completeness and timeliness	Completeness = LOW	Interact, if possible, with the DaaS providers to have additional contract terms.
		Timeliness = LOW	Interact, if possible, with the DaaS providers to update contract terms.

Conceptual architecture for contract management and evaluation



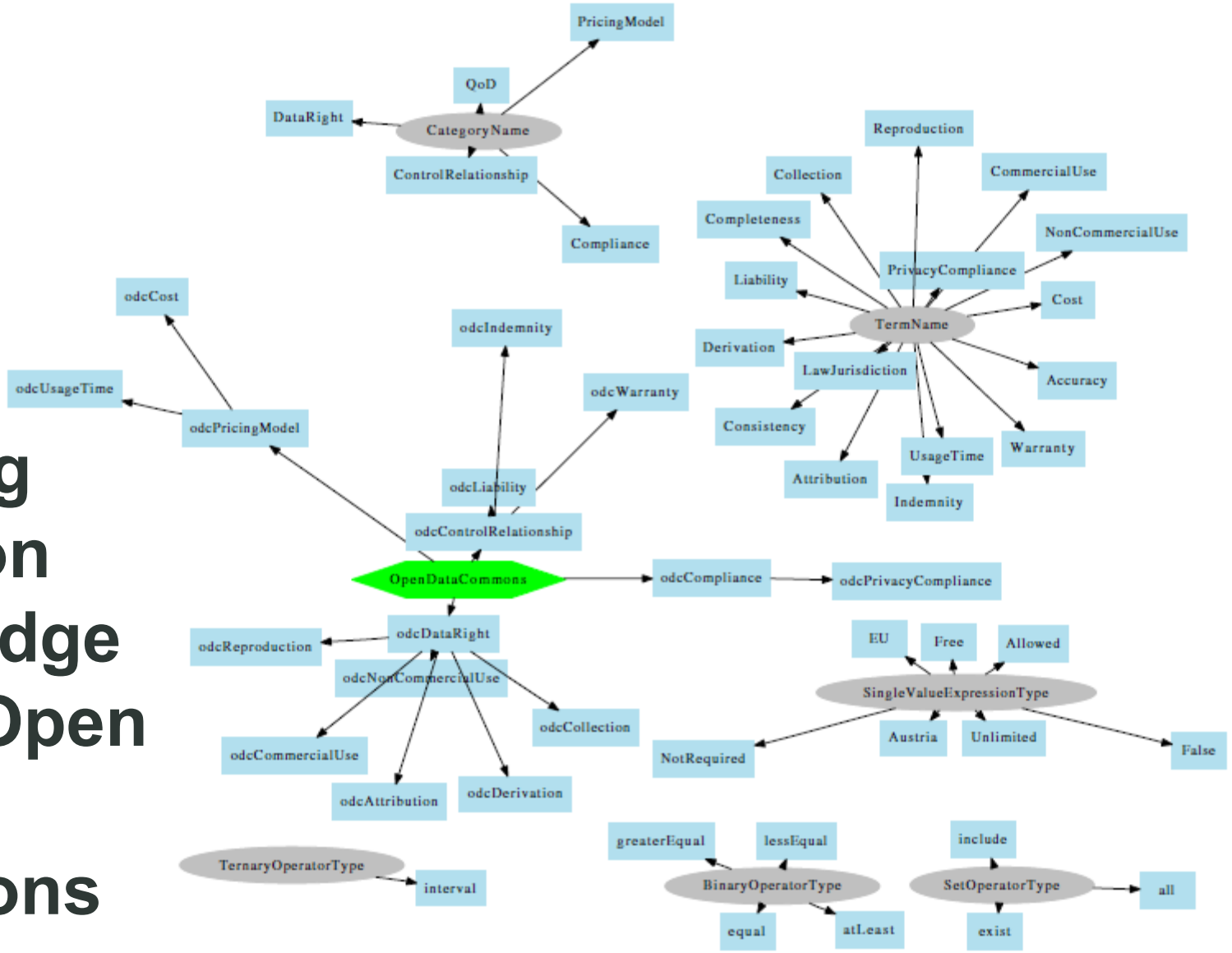


- Prototype
  - RDF for representing term categories, term names, term values, units
  - Allegro Graph for storing contract knowledge

Illustrating examples

- A large sustainability monitoring data platform shows how green buildings are
  - Real-time total and per capita of CO2 emission of monitored building
  - Open government data about CO2 per capita at national level
- We created contracts from
  - Open Data Commons Attribution License
  - Open Government License

Existing common knowledge about Open Data Commons



Step 2: provide OpenBuildingCO2

OpenBuildingCO2 by modifying quality of data and data right

OpenGov for government data

Statements with OpenBuildingCO2 » as the subject.

Predicate	Object
rdf:type	owl:NamedIndividual
rdf:type	AbstractDataContractType
rdfs:label	"OpenBuildingCO2"
dcCategory	obcDataRight
dcCategory	obcQoD
dcCategory	odcCompliance
dcCategory	odcControlRelationship
dcCategory	odcFinancial

Statements with OpenGovernment » as the subject.

Predicate	Object
rdf:type	owl:NamedIndividual
rdf:type	AbstractDataContractType
rdfs:label	"OpenGovernment"
dcCategory	ogCompliance
dcCategory	ogControlRelationship
dcCategory	ogDataRight
dcCategory	ogFinancial
contractHeader	ogHeader

Data contract for green building data

Example – composing data contract terms

Query Editor | Query Library

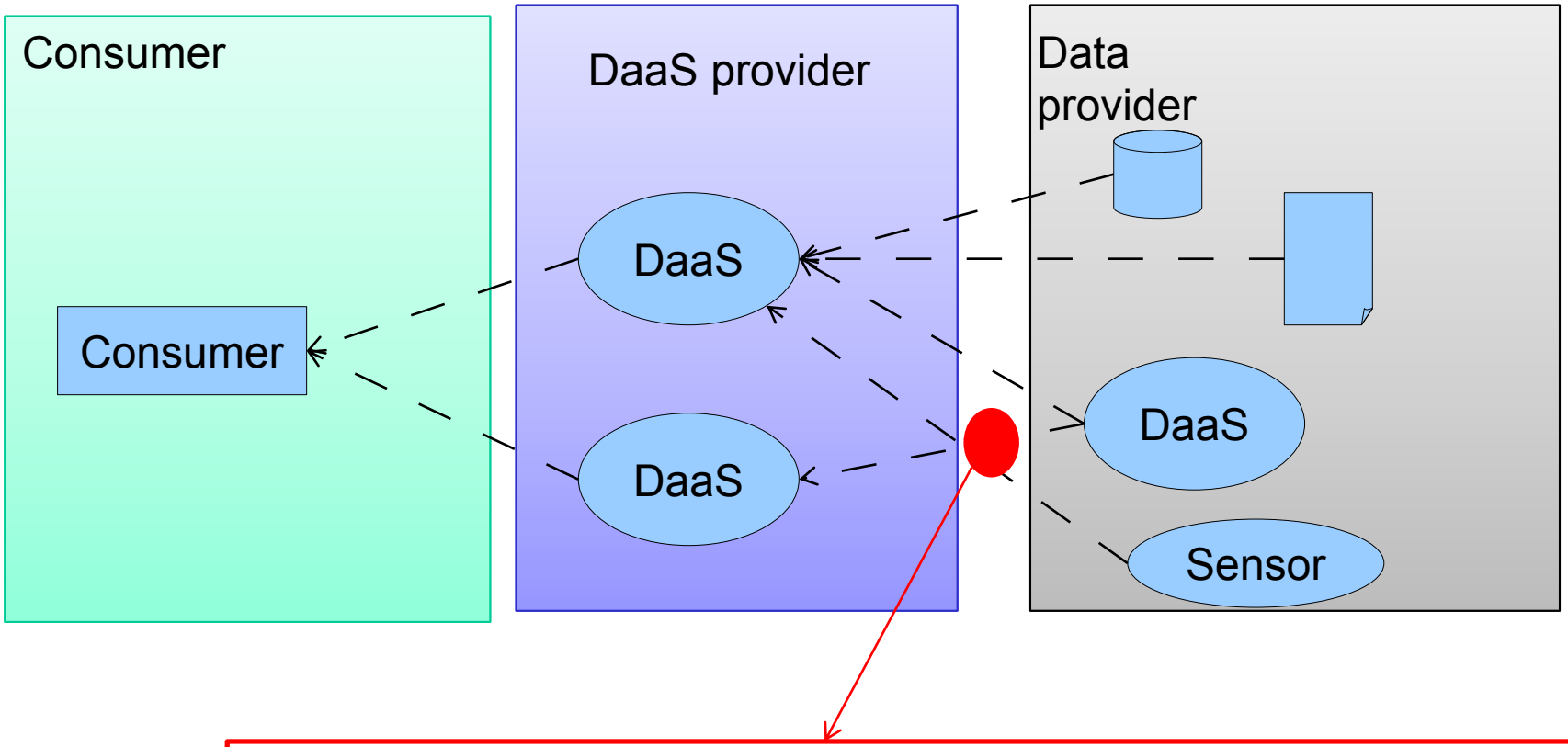
```
PREFIX
adcm: <http://www.infosys.tuwien.ac.at/SOD1/adcm#>
CONSTRUCT {
  adcm:compositeAccuracy adcm:lowerBound ?compositeLowerBound .
  adcm:compositeAccuracy adcm:upperBound ?compositeUpperBound .
}
WHERE {
   ?varAcc1 rdf:type adcm:SingleValueExpressionType .
   ?varAcc1 adcm:numericValue ?value .
   ?varAcc1 adcm:binaryOperator adcm:atLeast .
   ?varAcc2 rdf:type adcm:RangeExpressionType .
   ?varAcc2 adcm:lowerBound   ?lowerBound .
   ?varAcc2 adcm:upperBound   ?upperBound .
FILTER (?value <= ?upperBound) .
LET (?compositeLowerBound := afn:max(?value, ?lowerBound)) .
LET (?compositeUpperBound :=?upperBound) .
}
```

Subject	[Predicate]	Object
ns1:compositeAccuracy	ns1:lowerBound	F 0.7
ns1:compositeAccuracy	ns1:upperBound	F 0.95

Discussion time

CAN WE AUTOMATICALLY GENERATE DATA CONTRACTS FOR NEAR-REALTIME DATA?

Exchange data contract (1)



How do they interact w.r.t. data concerns?
How do their data contracts look like?

Exchange data contract (2)

- Lack of models and protocols for data contract exchange in data marketplaces
  - Constraints for data usage are not clear
  - Inadequate data/service description → hindering automatic (near realtime) data selection and integration
- Existing techniques are not adequate for dynamic data contract exchange in data marketplaces

Need generic exchange models suitable for different ways of data provisioning in data marketplaces

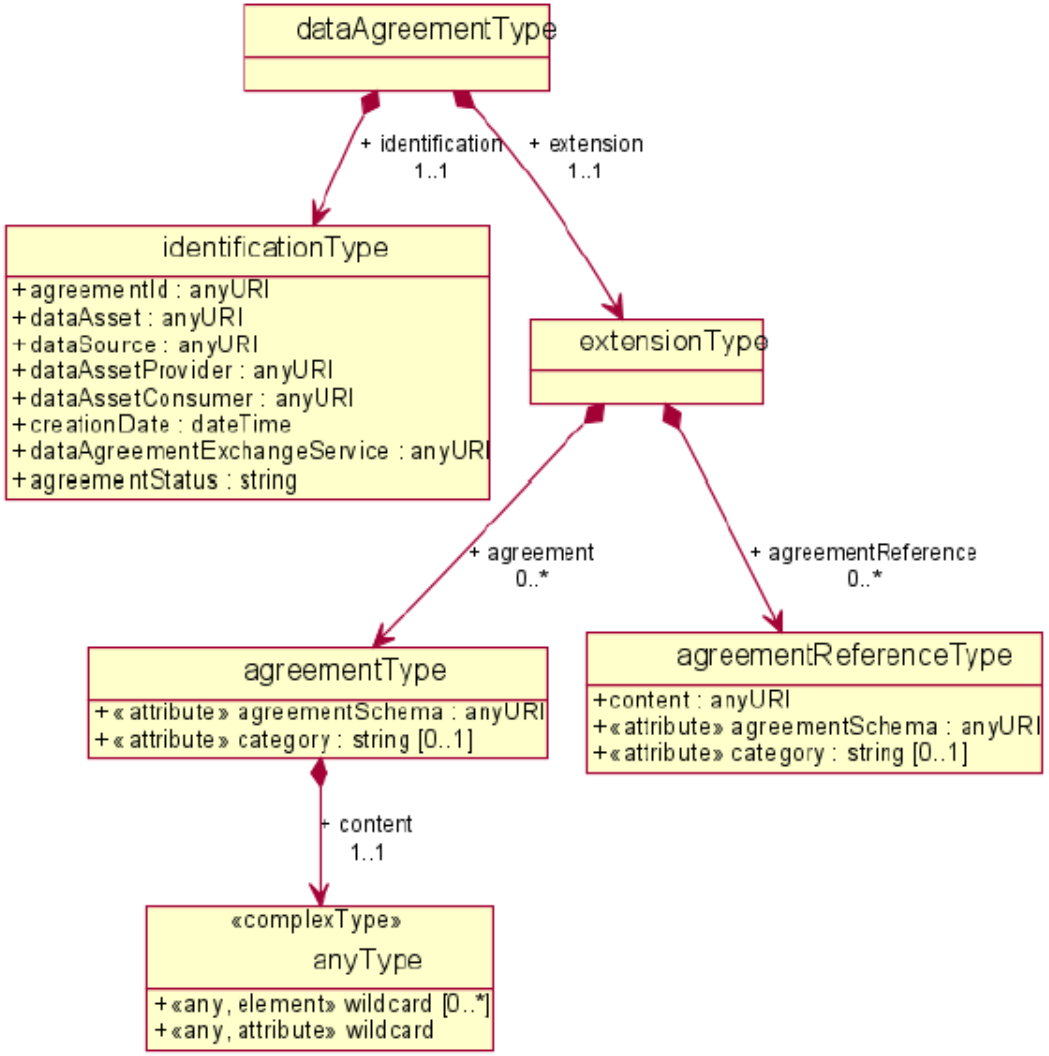
Data Contract Exchange as a Service

- Metamodel for data contract exchange
  - More than data contracts themselves
- Techniques for enriching and associating data assets with contract terms
- Interaction models for data contract exchange

Hong Linh Truong, Schahram Dustdar, Joachim Götze, Tino Fleuren, Paul Müller, Salah-Eddine Tbahriti, Michael Mrissa, Chirine Ghedira: Exchanging Data Agreements in the DaaS Model. APSCC 2011: 153-160

Metamodel for data contract exchange

- Different category of agreements
  - Licensing, privacy, quality of data
- Extensions
  - Languages
  - Different types of agreements
  - Different specifications

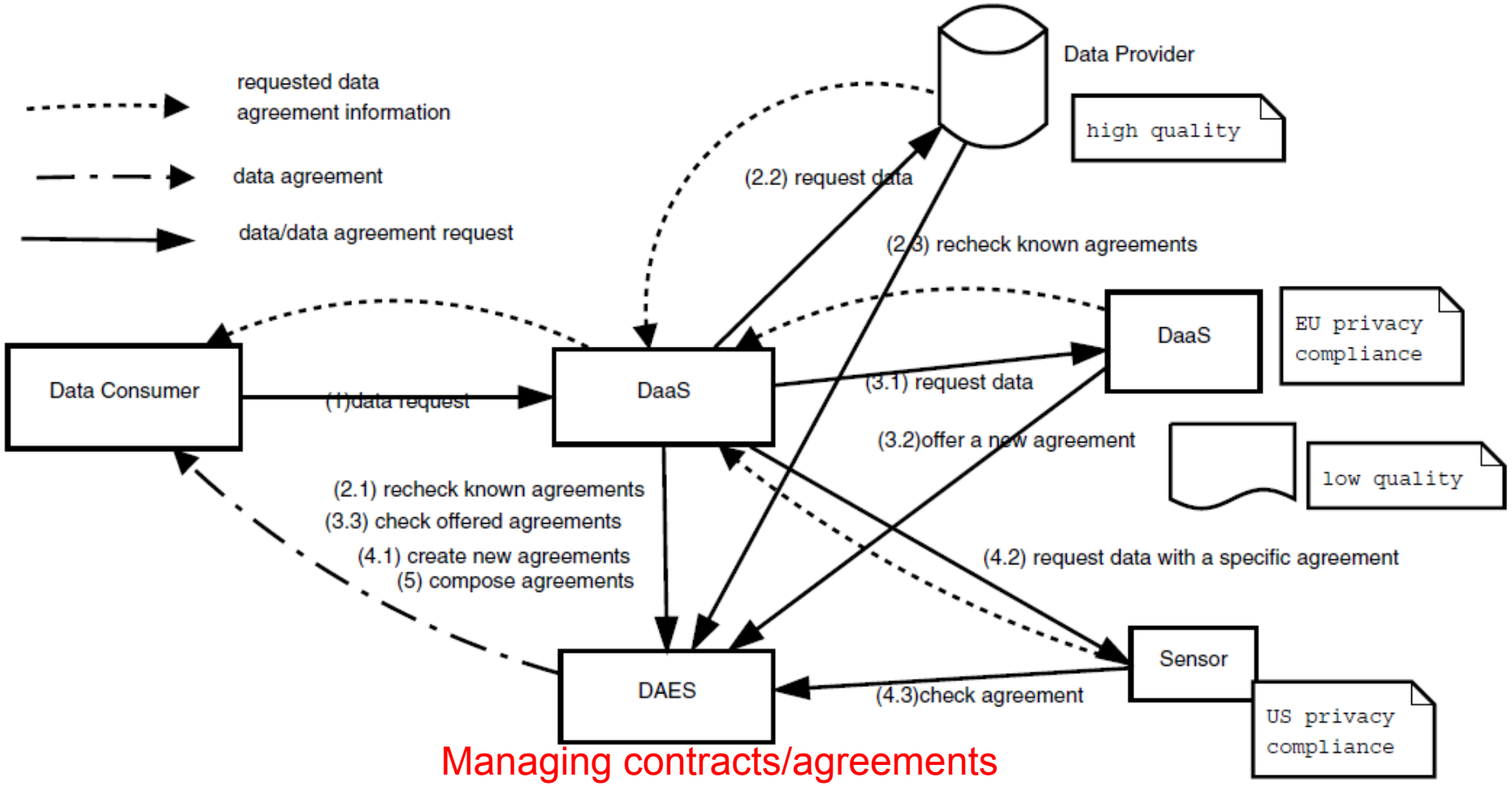


Associating data with data contracts

- Solutions
  - (a) directly inserting contracts into data assets
  - (b) providing two-step access to contracts and data assets
  - (c) linking data contracts to the description of DaaS
  - (d) linking data contracts to the message sent by DaaS

coupling	tight-coupling	loose-coupling		
technical solution	(a)	(b)	(c)	(d)
structured data	(+)ciphering possible (-)requires specific client (-)not scalable (-)modifies data structure	(+)data-independent (-)manual access only	(-)service-specific data agreement (-)no enforcement possible	(+)message-specific data agreement (+)ciphering possible
unstructured data	(+)data agreement enforcement possible (+)ciphering possible (-)requires specific consumer (-)costly data agreement injection	(+) data-independent (-)manual access only	(-)service-specific data agreements (-)no enforcement possible	(+)message-specific data agreement (+)ciphering possible

Possible interaction models for data enriched with data contracts



Illustrating examples – insert agreement into data asset

- A pay-per-use consumer uses **dataAPI** of DaaS search for data
  - The consumer pays the use APIs
  - Each call can return different types of data

Example of searching people

```
{"results":[
dataagreement:{
    ...
    extension: {
       agreementReference: {
            category:''licensing'',
            agreementSchema  :''urn:at:act:tuwien:infosys:
                 license:twitter'',
            content:''http://.../DAES/da/references/
                 retrieve/peoplesearch-license'',
       }
    }
},
{"description":"Student: spatial planning, music. Location:
     Vienna. Interests: Everything
 nerdy.","location":"Vienna,
 Austria","time_zone":"Vienna","user_id":REMOVED,"
     utc_offset":"3600","name":"REMOVED","scraped_at"
     :1259592694000,"screen_name":"REMOVED"},
...}}
```

But a strong consequence for data service engineering techniques: dealing with elastic requirements!

Illustrating examples – link agreements to geospatial data

- Domain-specific DaaS: different agreements for different data requests
  - Vector data of geographic features via Web-Feature-Service (WFS)
  - Terrain elevation data via Web-Coverage Services (WCS)

```
−<identification>
    <agreementId>urn:de:icsyagreement:1</agreementId>
    <dataAsset>urn:de:icsy:asset:wcs:1</dataAsset>
  −<dataSource>
      http://gdi-arc1.gridlab.uni-kl.de/arcgis01/services/Hamburg_WFS/MapServer/WCSServer?service=WCS
    </dataSource>
  −<dataAssetProvider>
      http://gdi-arc1.gridlab.uni-kl.de/arcgis01/services/Hamburg_WFS/MapServer/WCSServer?service=WCS
    </dataAssetProvider>
    <dataAssetConsumer>urn:de:icsy:j_goetze</dataAssetConsumer>
    <creationDate>2011-07-04T20:11:15.029Z</creationDate>
  −<dataAgreementExchangeService>
      http://sod.infosys.tuwien.ac.at:7101/services/jersey/DAES
    </dataAgreementExchangeService>
    <agreementStatus>AGREED</agreementStatus>
  </identification>
−<extension>
  −<agreementReference agreementSchema="urn:de:icsy:license:wcs" category="licensing">
    −<content>
        http://sod.infosys.tuwien.ac.at:7101/services/jersey/DAES/da/references/retrieve/license_wcs
      </content>
    </agreementReference>
  </extension>
</ns0:dataAgreement>
```

Illustrating examples – link agreements to geospatial data

```
−<o-ex:rights xsi:schemaLocation="http://www.opengis.net/wcs http://schemas.opengis.net/wcs/1.0.0/getCoverage.xsd htt
 EX-11.xsd http://odrl.net/1.1/ODRL-DD http://odrl.net/1.1/ODRL-DD-11.xsd">
 −<o-ex:offer>
  −<o-ex:asset o-ex:id="ASSET1">
   −<o-ex:context>
      <o-dd:uid>WCS_ASSET1</o-dd:uid>
     −<wcs:request>
      −<wcs:getCoverage>
       −<wcs:listOfLayers wcs:setFunction="
          <wcs:layer>composite</wcs:layer>
          <wcs:layer>amp</wcs:layer>
        </wcs:listOfLayers>
       −<wcs:defaultZoomIn>
          <wcs:unrestricted/>
        </wcs:defaultZoomIn>
      </wcs:getCoverage>
     </wcs:request>
     −<o-dd:service>
        http://gdi-arc1.gridlab.uni-kl.de/arcgis01/services/Hamburg_WFS/MapServer/WCSServer?service=WCS
      </o-dd:service>
    </o-ex:context>
  </o-ex:asset>
  −<o-ex:asset o-ex:id="ASSET2">
   −<o-ex:context>
      <o-dd:uid>WMS_ASSET2</o-dd:uid>
     −<wcs:request>
      −<wcs:getCoverage>
       −<wcs:listOfLayers wcs:setFunction="include">
          <wcs:layer>criobe</wcs:layer>
        </wcs:listOfLayers>
       −<wcs:defaultZoomIn>
```

Software can interpret and reason if the data can be used for specific purposes

Illustrative examples – develop an app for policy compliance (1)

```
public class TwitterCompatibilityApp implements
     DataAgreementAppInterface {
    //...
    public String getResult() {
        return output;
    }
    public void setDataAgreements(List dataAgreements,
         boolean reference) {
        this.dataAgreements = dataAgreements;
        //....
    }
    public void execute() {
        ModelManager mm = new ModelManager();
        String agreementReference1 = (String)dataAgreements
            .get(0);
        String agreementReference2 = (String)dataAgreements
            .get(1);
        OntModel m1 = mm.loadFromAgreementContent(
            agreementReference1);
        OntModel m2 = mm.loadFromAgreementContent(
            agreementReference2);
        boolean valid1= mm.validateModel(m1);
        boolean valid2= mm.validateModel(m2);
        boolean compatible=mm.isCompatible(m1, m2);
        output="<?xml version=\"1.0\" encoding=\"UTF-8\"
            ?>\n" +
        //...
    }
    public void setDescription(DataAgreementApp description
         ) {
    //...
```

Illustrative examples – develop an app for policy compliance (2)

Configuration

```
<dataAgreementApp>
  <withPlugin>RDF-Policy-Twitter-Mashup-CompApp</withPlugin
      >
  <withCapabilities>
    <dataAgreementAppCapability name="compatibility" value=
        "true"/>
    <dataAgreementAppCapability name="validation" value="
        true"/>
    <worksWithSpects>
      <dataAgreementSpecification>
        <contentURI>http://sod.infosys.tuwien.ac.at:7101/
            services/jersey/DAES/daspecs/retrieve/
            twitterpolicy</contentURI>
        <language>RDF</language>
      </dataAgreementSpecification>
    </worksWithSpects>
  </withCapabilities>
</dataAgreementApp>
```

Results

```
<compatibilityresult>
  <validation agreementReference="twitterpolicy1" value="true"/>
  <validation agreementReference="twitterpolicy2" value="true"/>
  <compatibility>false</compatibility>
</compatibilityresult>
```

Discussion time

HOW DOES NEAR-REALTIME DATA IMPACT ON DATA CONTRACT EXCHANGE?

Data Market without Marketplace?

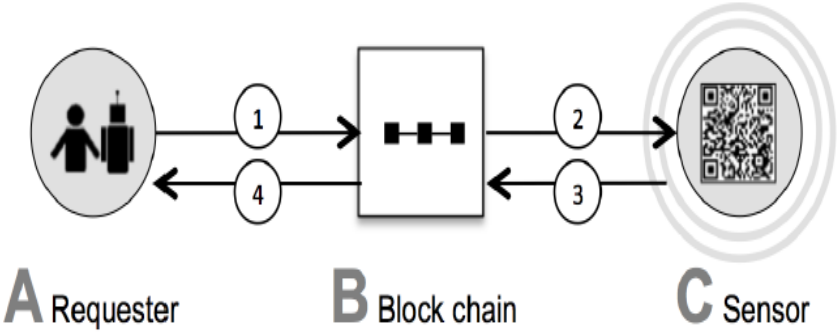


Fig. 1. Schema for the atomic S²aaS process of exchanging a single datum for cash using Bitcoin.

Kay Noyen, Dirk Volland, Dominic Wörner, Elgar Fleisch:
When Money Learns to Fly: Towards Sensing as a Service Applications Using Bitcoin.

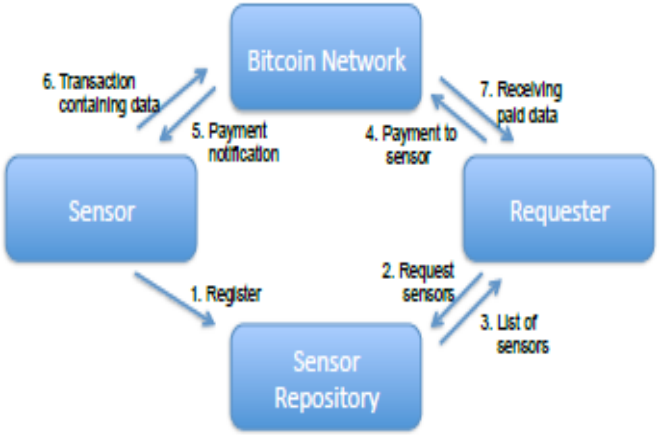


Figure 1: Process for exchanging data for bitcoin.

Dominic Wörner and Thomas von Bomhard. 2014. **When your sensor earns money: exchanging data for cash with Bitcoin.** In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct). ACM, New York, NY, USA, 295-298.

But what about data contract?

Some data marketplaces and their features

Products	Data type	Data source	Publishing /delivery	Cost model	Auto-lookup	Data con-tract	Payment
MARSA	Realtime, streaming	IoT devices	MOM[5], IoT platform	Flexible[6]	yes	yes	online billing
Xignite	Datasets, realtime	Range, finance	Files, API	Asset, deliv-ery	Yes	N/A	N/A
Amazon	Datasets	Range	Files	Free	N/A	N/A	N/A
Azure	Datasets	Range	OData API	Subscription	N/A	Publisher offer terms	N/A
Factual	Datasets	Geography	Files, API	Free/ sub-scription	Yes	Terms of services	N/A
Trimble insphere	Datasets	Geography	Files	per user/ de-vice/ data	N/A	License Agreement	N/A
Gnip	Realtime, historical	Social network	API	N/A	Yes	N/A	N/A
Sense2Web	Realtime, streaming	IoT devices	MOM, IoT platform	N/A	Yes	N/A	N/A

Tien-Dung Cao, Tran-Vu Pham, Quang-Hieu Vu, Hong-Linh Truong, Duc-Hung Le, Schahram Dustdar, "MARSA: A Marketplace for Realtime Human-Sensing Data," Transactions on Internet Technology, 2016

CASE STUDY – DESIGN DATA MARKETPLACE

MARSA: A Marketplace for Realtime Human-Sensing Data

Cao, Tien-Dung ; Pham, Tran-Vu ; Vu, Quang-Hieu ; Le, Duc-Hung ; Truong, Hong-Linh ; Dustdar, Schahram

ACM Transactions on Internet Technology, 2016

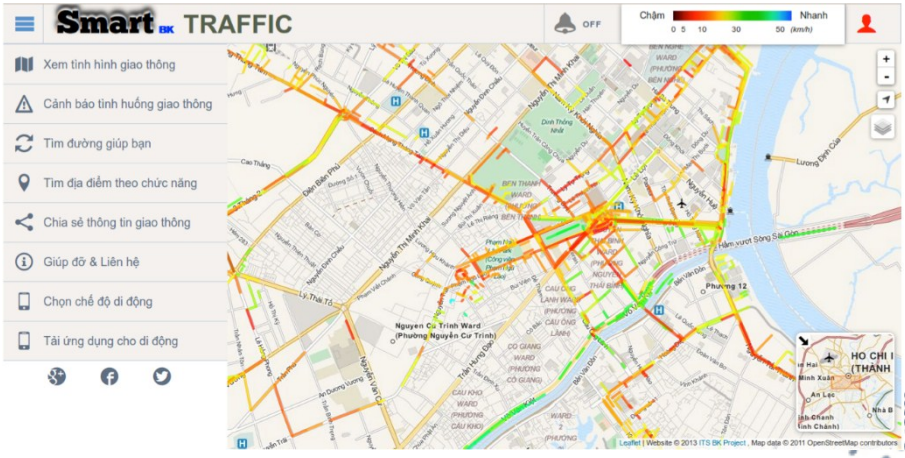
http://dungcao.github.io/marsa/

Traffic problems in HoChiMinh City

- Crowded and unpredictable
- Needs a lot of data to understand traffics

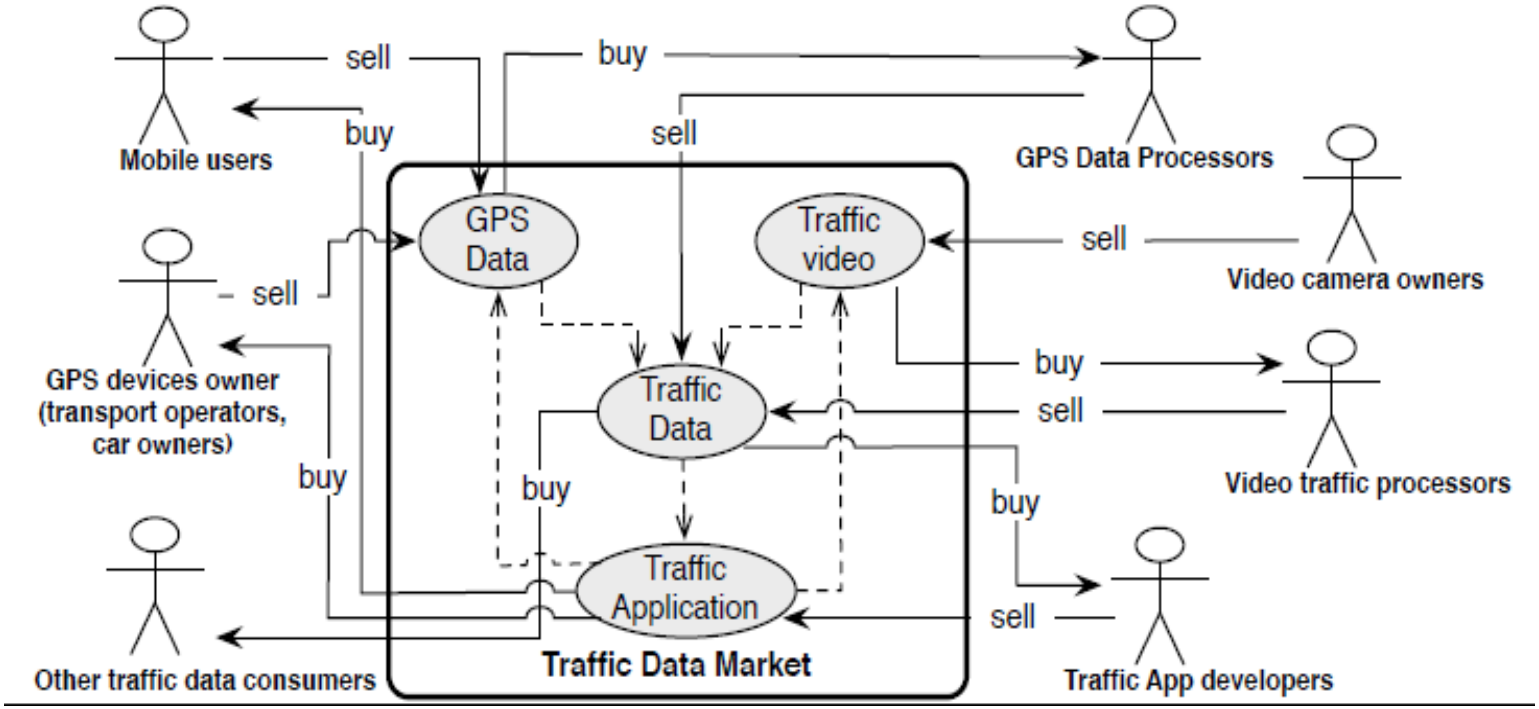
- Lack infrastructures for collecting traffic information
- Common problems in developing countries

Figure sources: Internet



Cannot buy expensive traffic data collection systems!

Market-oriented View of traffic data scenarios



4000 citybus fleet, 0.25MB per day per bus (7.5MB/month/bus), 30GB for the fleet

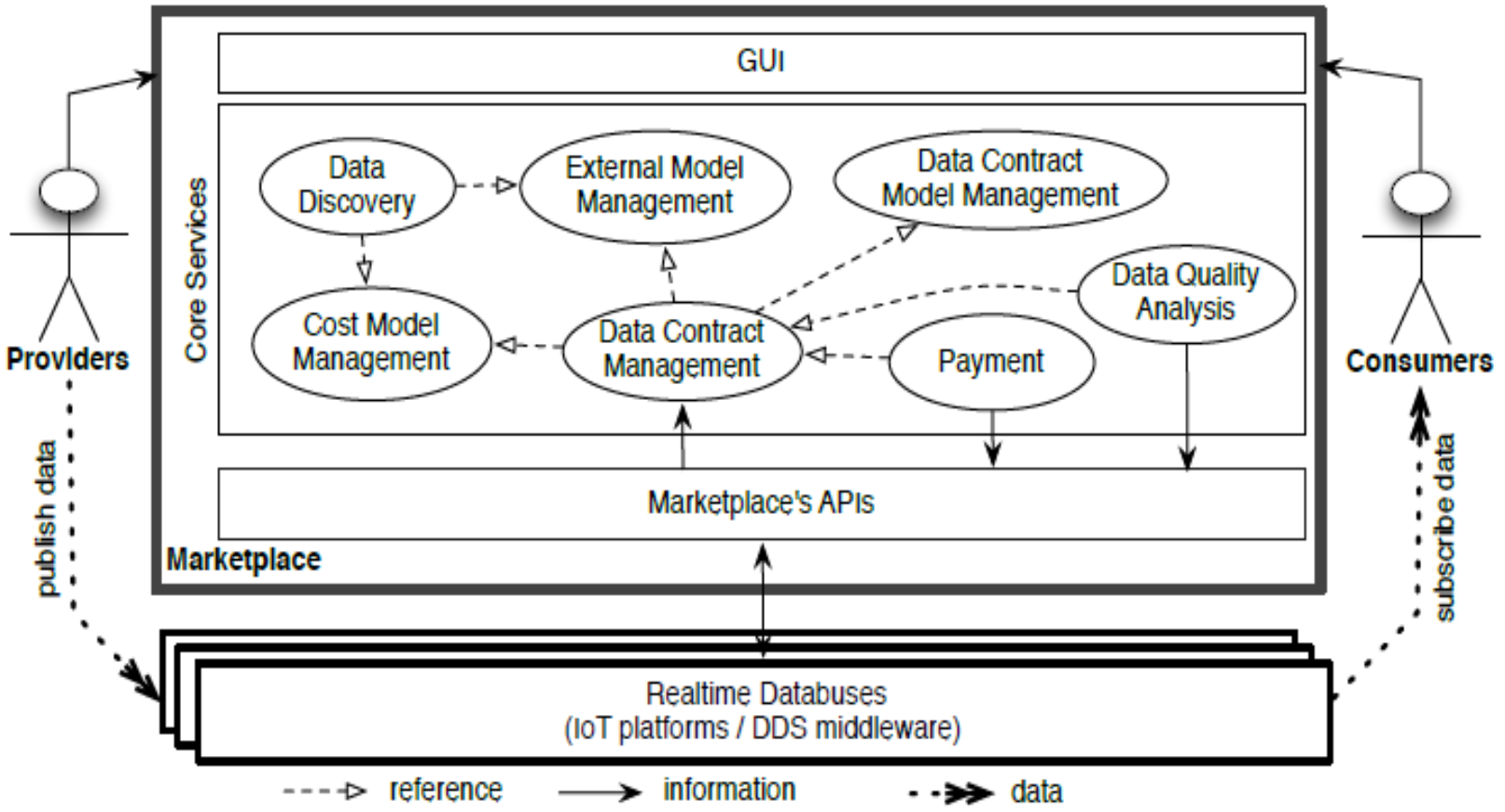
1MB of GPS data =20 USD cent → 6000 USD for the fleet operators

A mobile phone, like a bus, can receive 1.5 USD per month → ½ of 3G data bill

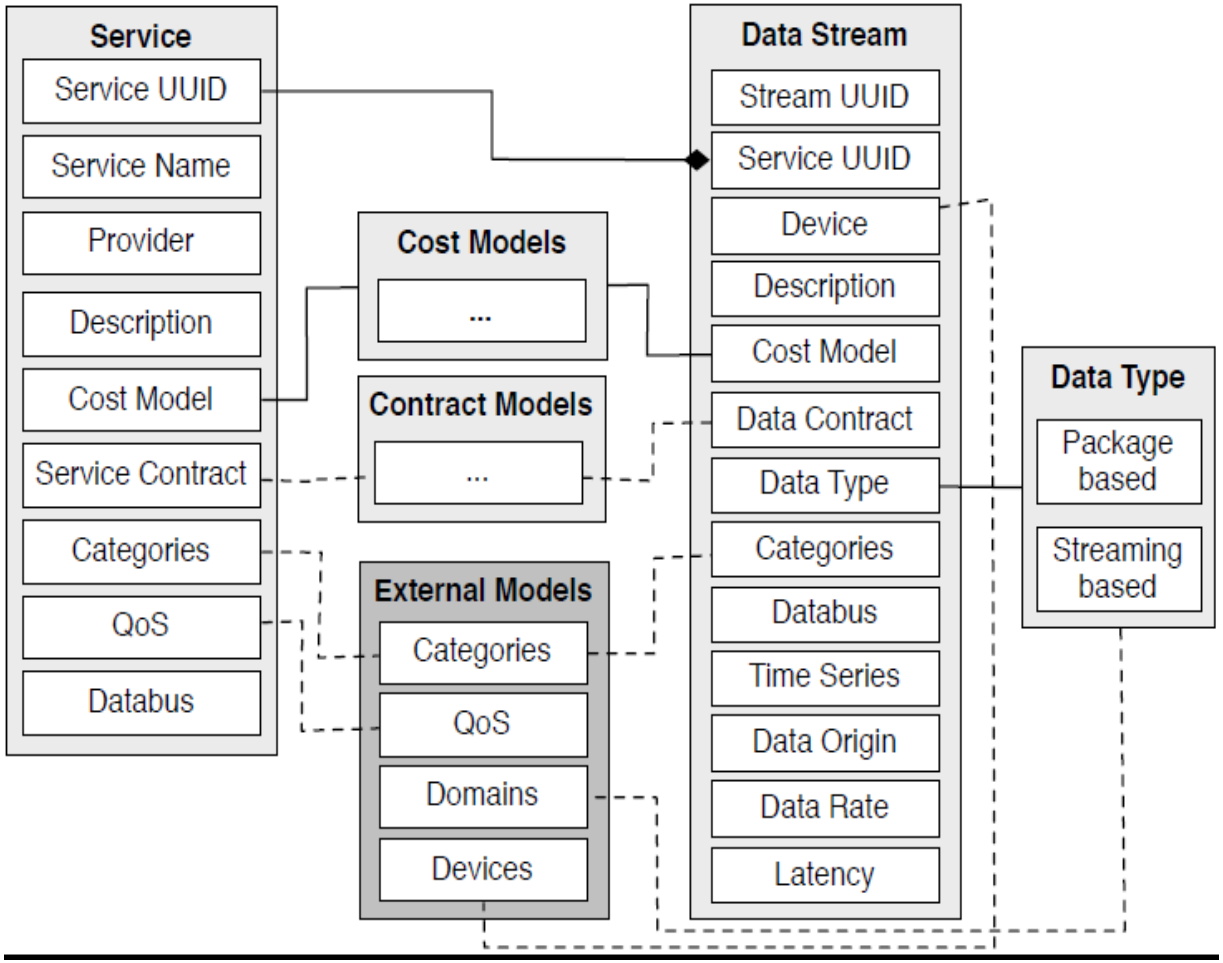
Costs and benefits

Parties	Costs of collecting raw data	Benefits from processed traffic data
Bus, taxi and truck operators	GPS devices, Internet and mobile network subscription fees, acquiring and maintaining data at servers	Able to track status of their buses, knowledge of current traffic conditions to better provide services to commuters
Private car owners	GPS devices, mobile network subscription fees	Knowledge of current traffic conditions to better navigate in cities
Mobile device owners	Mobile devices (e.g. smartphones, tablets), mobile network subscription fees and device battery time	Knowledge of current traffic conditions to better navigate in cities
Video camera owners	Video cameras and network connections to video cameras	Selling of video data and traffic information
Data processors	Cost of raw data, infrastructures for collecting and processing raw data	Selling traffic data
Traffic data users	Buying traffic data	Knowledge of current traffic conditions to better navigate in cities

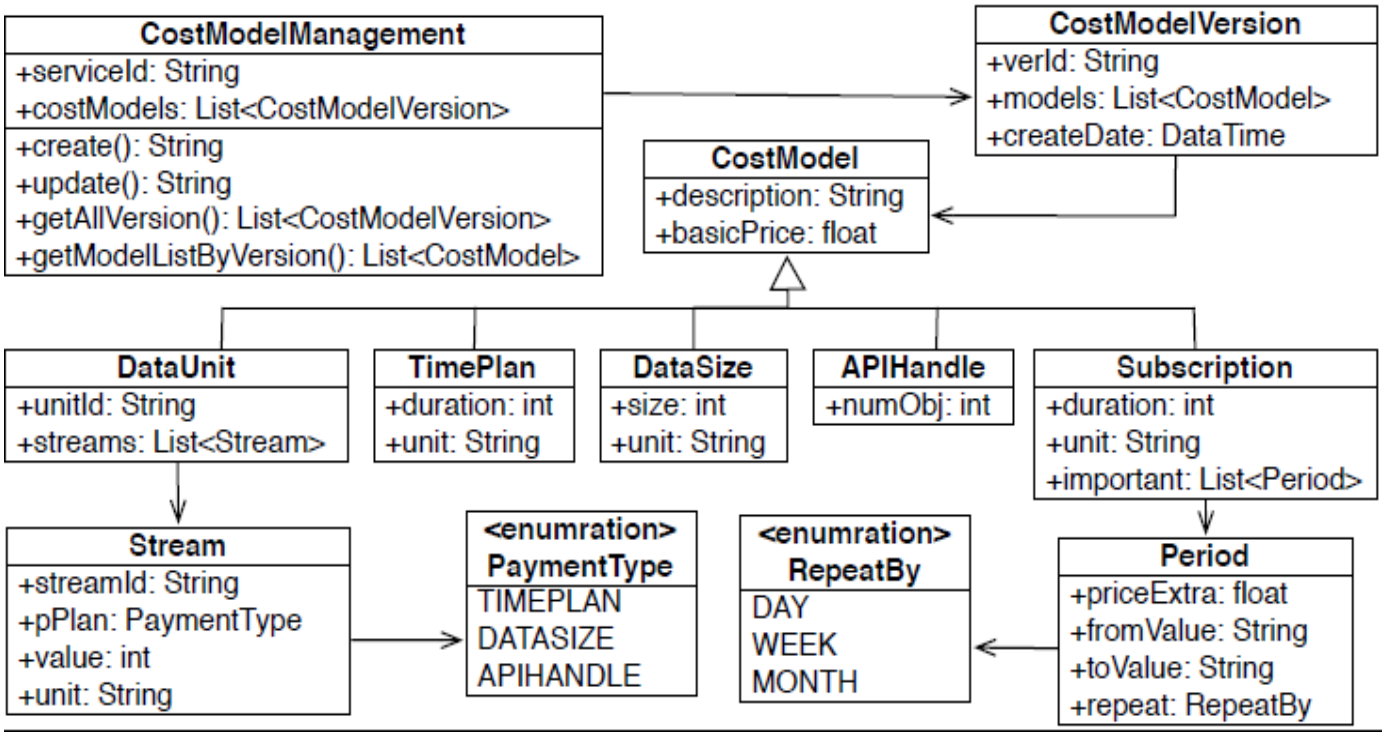
MARSA Design



MARSA description for human-sensing data marketplace

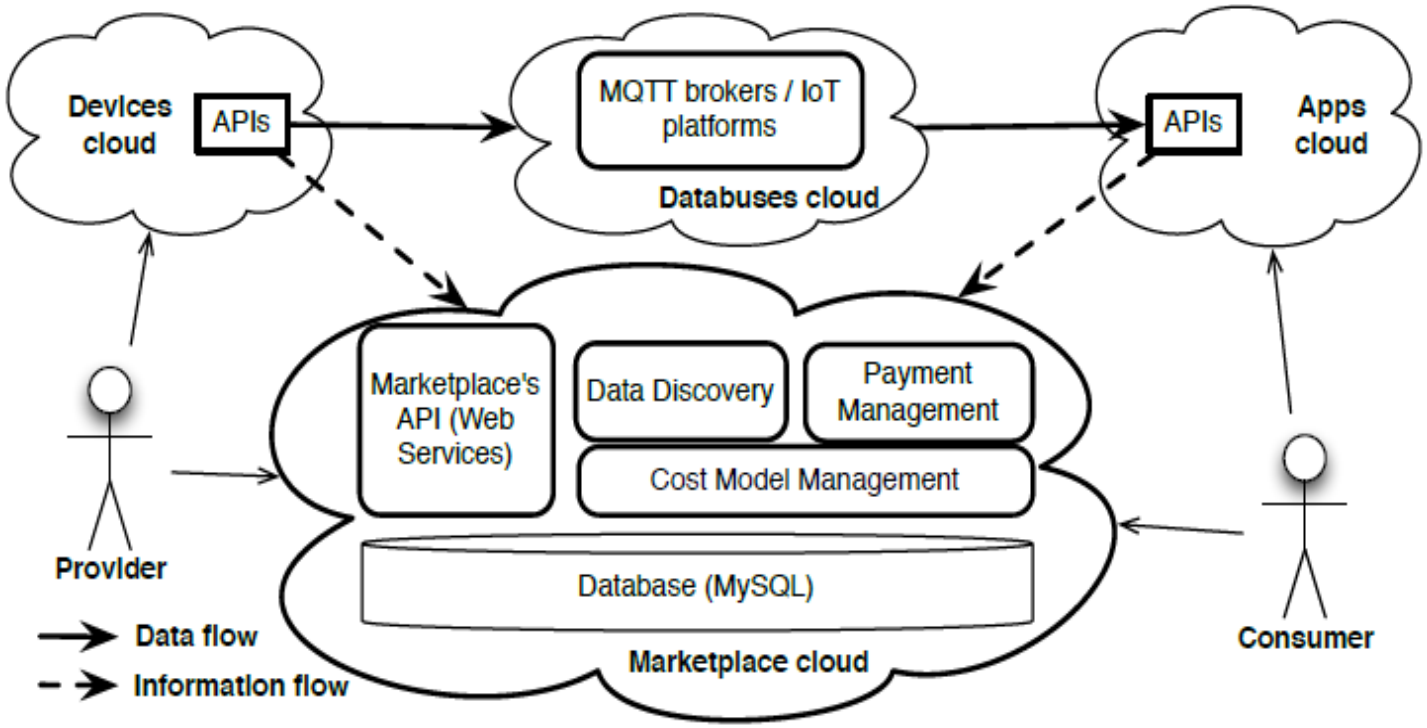


Cost model

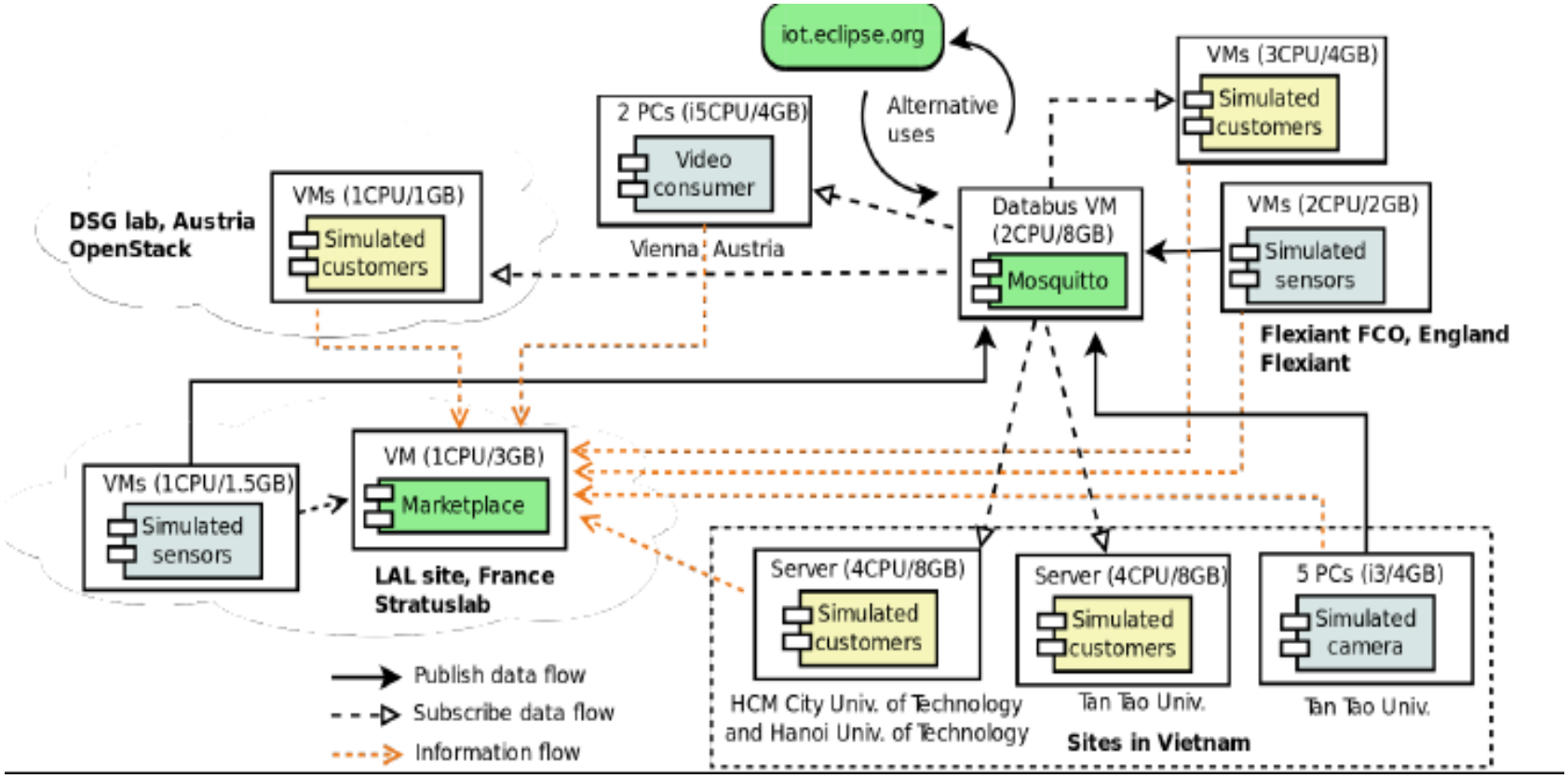


Quality of data has not supported yet

Implementation



Testbed


iot.eclipse.org
Alternative uses
2 PCs (i5CPU/4GB)
Video consumer
Vienna, Austria
VMs (3CPU/4GB)
Simulated customers
Databus VM (2CPU/8GB)
Mosquitto
VMs (2CPU/2GB)
Simulated sensors
Flexiant FCO, England
Flexiant
DSG lab, Austria
OpenStack
VMs (1CPU/1GB)
Simulated customers
VM (1CPU/3GB)
Marketplace
VMs (1CPU/1.5GB)
Simulated sensors
LAL site, France
Stratuslab
Server (4CPU/8GB)
Simulated customers
HCM City Univ. of Technology and Hanoi Univ. of Technology
Server (4CPU/8GB)
Simulated customers
Tan Tao Univ.
5 PCs (i3/4GB)
Simulated camera
Tan Tao Univ.
Sites in Vietnam
Publish data flow
Subscribe data flow
Information flow

Example of bills

Bill No.: 2015/03-5.1

From date: 2015-03-30 12:39:53 To date: 2015-03-30 18:40:57

Status: Not Payment

Payment on DATA_SIZE (5.0 $ / 1 GB)

List of streams

No.	Stream UUID	Size	Price
1	suuid1427702254973/sid1	0.219 GB	$ 1.1
2	suuid1427702254973/sid2	0.0217 GB	$ 0.11
3	suuid1427702254973/sid3	0.0550 GB	$ 0.28
4	suuid1427702254973/sid4	0.181 GB	$ 0.9
5	suuid1427702254973/sid5	0.205 GB	$ 1.02
			Total price: $ 3.41

Payment on SUBSCRIPTION (2.0 $ / 1 HOUR)

List of streams

No.	Stream UUID	Size	Price	Size Extra	Price Extra	Sum Price
1	suuid1427702254973/sid11	3.67 HOUR	$ 7.34	0	$ 0	$ 7.34
2	suuid1427702254973/sid12	6.02 HOUR	$ 12.04	0	$ 0	$ 12.04
						Total Price: $ 19.38

Total price of contract: $ 22.79

Exercises

- Read mentioned papers
- Examine existing data marketplaces and write DEMODS-based specification for some of them
- Develop some specific data contracts for open government data
- Work on some algorithms for checking data contract compatibility
- Incorporate data marketplaces concepts into your scenario
- Build your own mini data marketplace

Thanks for your attention

Hong-Linh Truong
Distributed Systems Group, TU Wien
truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong
@linhsolar