

Service Systems

Modeling and Applications

IFIP WG 2.14
Amsterdam, 10-11 October, 2013

Jorge Cardoso
Information Systems Group

Karlsruhe Service Research Institute, Karlsruhe Institute of Technology, Germany

jorge.cardoso@kit.edu

Dept. Engenharia Informatica/CISUC, University of Coimbra, Portugal

jcardoso@dei.uc.pt

The Present

Light Utilization Reserved Instances

Region: US East (N. Virginia)

1 yr Term

Upfront

Hourly

Standard Reserved Instances

Small (Default)	\$61	\$0.034 per Hour
Medium	\$122	\$0.068 per Hour
Large	\$243	\$0.136 per Hour
Extra Large	\$486	\$0.271 per Hour

Second Generation Standard Reserved Instances

Extra Large

Double Extra Large

Micro Reserve

Micro

High-Memory

Extra Large

Double Extra Large

Quadruple Extra Large

High-CPU Reserve

Medium

Extra Large

- A "Service Credit" is a dollar credit, calculated as set forth below, that we will apply to your account.

Service Commitments and Service Credits

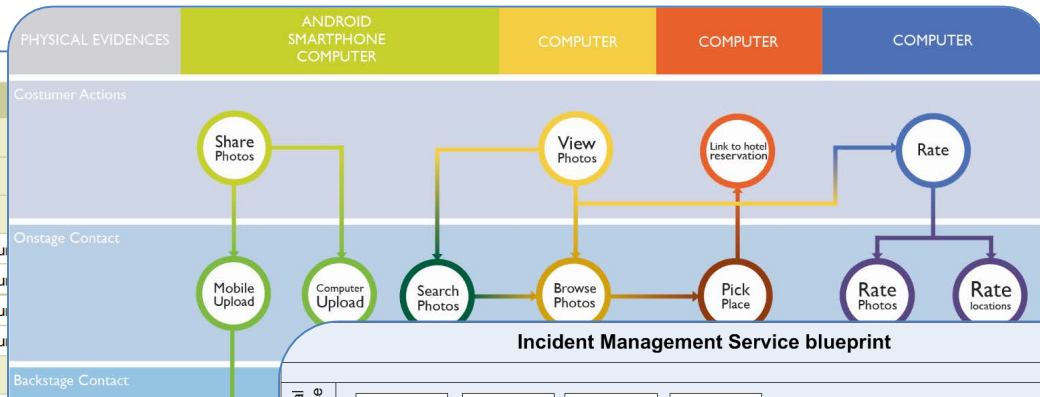
Service Credits are calculated as a percentage of the total charges paid by you (excluding any upfront payments made for Reserved Instances) for either Amazon EC2 or Amazon S3, if either was Unavailable, or both if both were Unavailable) in the Region affected for the month in which the Unavailability occurred in accordance with the schedule below.

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.95% but equal to or greater than 99.0%	10%
Less than 99.0%	30%

We will apply any Service Credits only against future Amazon EC2 or Amazon EBS charges. At our discretion, we may issue the Service Credit to the credit card you used to purchase the Reserved Instance. Service Credits will be applicable and issued only if the Unavailability occurred during the month in which the Unavailability occurred (\$1 USD). Service Credits may be applied to your account only if you have an AWS Agreement, your sole source of computing power is Amazon EC2 or Amazon S3, and you are not using an Amazon SLA.

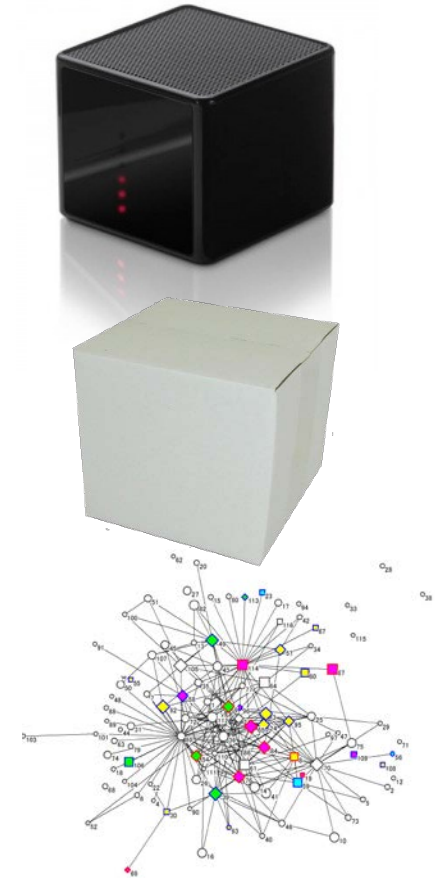
Credit Request and Payment Procedures

To receive a Service Credit, you must submit a claim by opening a case in the AWS Support Center. The credit request must be received by us by the end of the second billing cycle after the Unavailability occurred.



Service Systems

- Black box modeling
 - Linked USDL
- White box modeling
 - LSS4USDL
- Service Networks
 - Linked USDL and OSSR



Linked USDL family

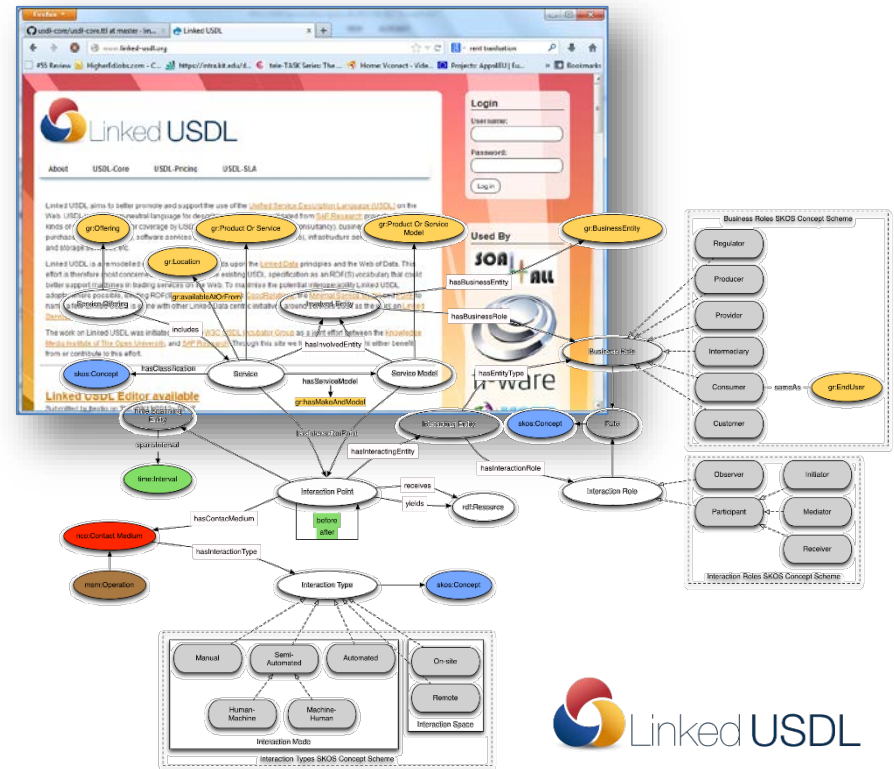
Service Description Language

- History

- a-USDL (2009), USDL (2011), Linked USDL (2012)

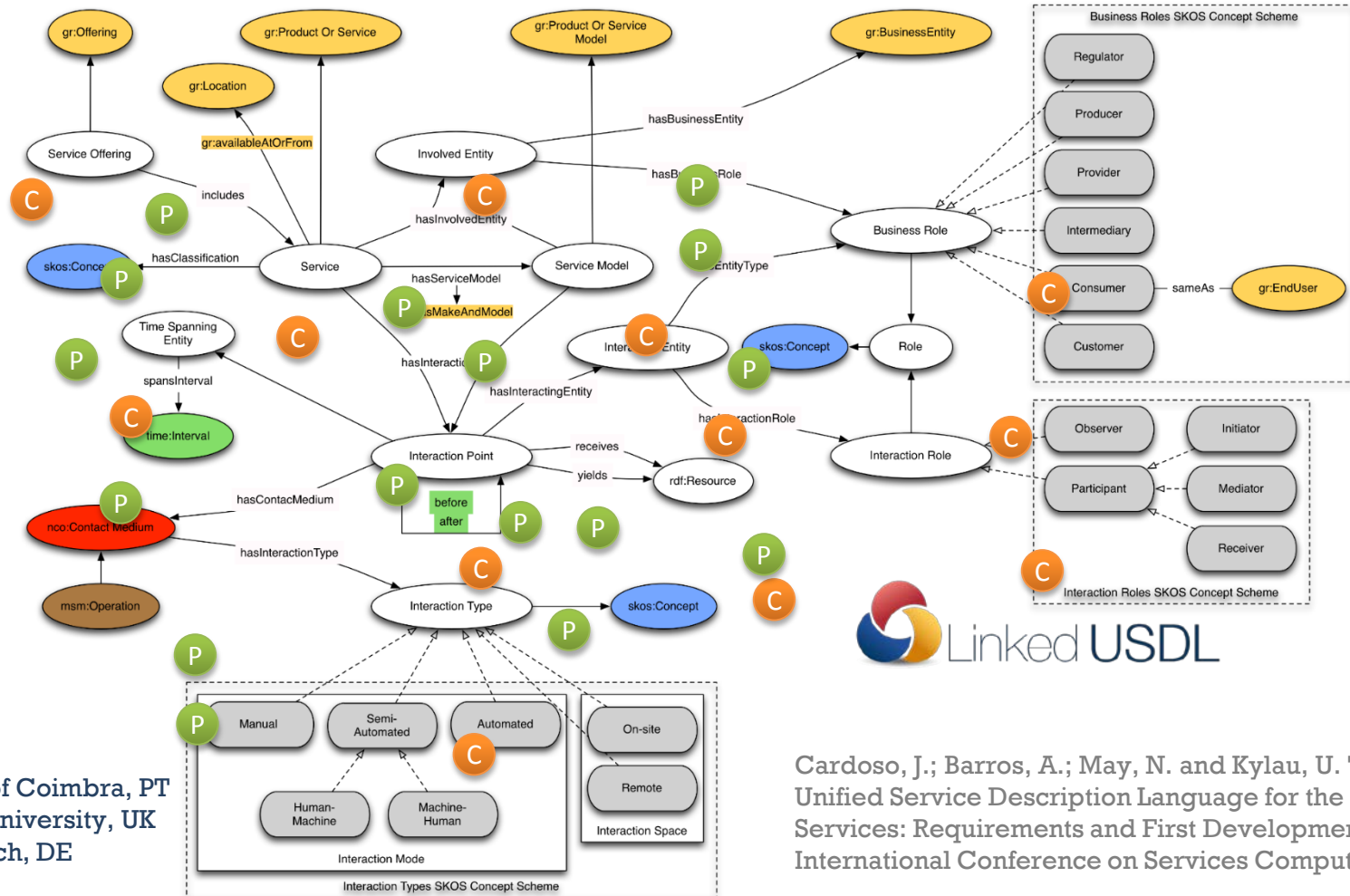
- New models (2013)

- Linked USDL core
 - The Open University, UK
 - SAP Research, DE
- Service System
 - University of Gent, BE
- Service Networks



<https://github.com/linked-usdl/>

Linked USDL Core (2013)



University of Coimbra, PT
The Open University, UK
SAP Research, DE

Cardoso, J.; Barros, A.; May, N. and Kylau, U. Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments. In IEEE International Conference on Services Computing, 2010.

USDL:INTERACTIONPOINT



- Blueprint
 - line of interaction
- E.g. face-to-face actions between employees and customers

NAME:

usdl:InteractionPoint

DESCRIPTION:

`rdfs:comment ""`

An InteractionPoint represents an actual step in accessing and performing operations of the service. On a technical level this could translate into calling a Web Service operation.

On a professional level, it could mean that consumer and provider meet in person to exchange service parameters or resources involved in the service delivery (e.g. documents that are processed by the provider).

An InteractionPoint can be initiated by the consumer or the provider. Since InteractionPoints may take time and have an ordering with respect to other InteractionPoints, this is a subclass of TimeSpanningEntity. One can therefore express temporal relationships between InteractionPoints such as before or after. For richer expressions the time ontology constructs could be used.

`""@en .`

SUBCLASS:

`rdfs:subClassOf usdl:TimeSpanningEntity;`



Linked USDL Pricing (2013)

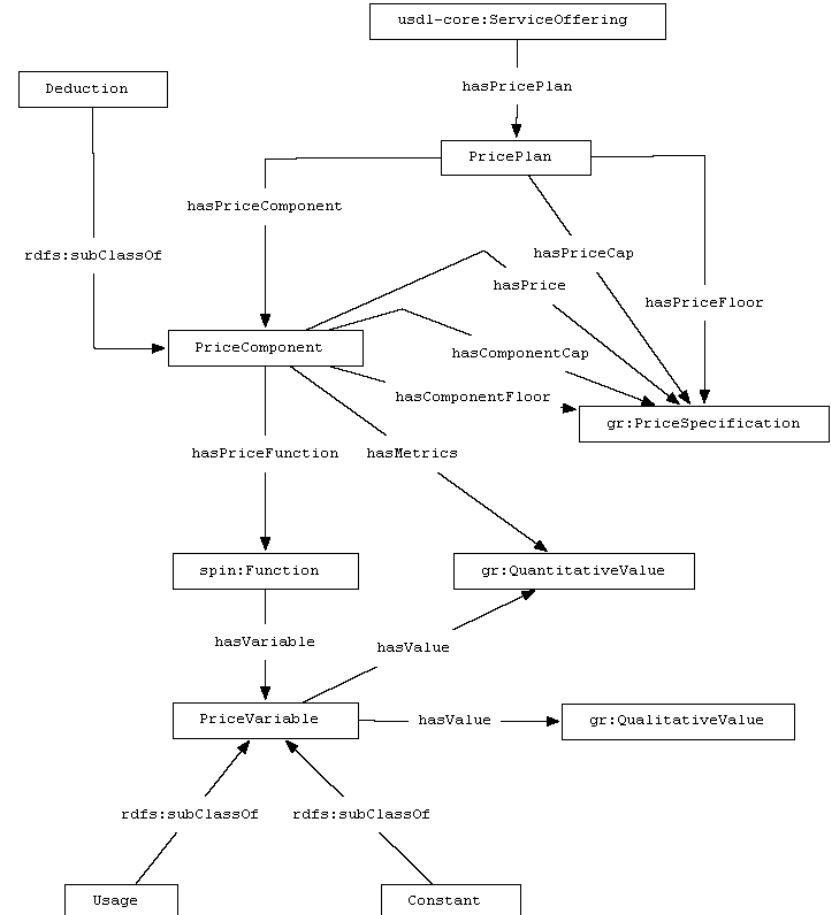
- ↓ On-Demand Instances
- ↓ Reserved Instances
- ↓ Reserved Instance Volume Discounts
- ↓ Spot Instances
- ↓ Data Transfer
- ↓ EBS-Optimized Instances
- ↓ Amazon Elastic Block Store
- ↓ Elastic IP Addresses
- ↓ Amazon CloudWatch
- ↓ Auto Scaling
- ↓ Elastic Load Balancing
- ↓ AWS GovCloud Region

Light Utilization Reserved Instances

Region:	US East (N. Virginia)			
	1 yr Term		3 yr Term	
	Upfront	Hourly	Upfront	Hourly
Standard Reserved Instances				
Small (Default)	\$61	\$0.034 per Hour	\$96	\$0.027 per Hour
Medium	\$122	\$0.068 per Hour	\$192	\$0.054 per Hour
Large	\$243	\$0.136 per Hour	\$384	\$0.108 per Hour
Extra Large	\$486	\$0.271 per Hour	\$768	\$0.215 per Hour
Second Generation Standard Reserved Instances				
Extra Large	\$517	\$0.299 per Hour	\$807	\$0.236 per Hour
Double Extra Large	\$1034	\$0.598 per Hour	\$1614	\$0.472 per Hour
Micro Reserved Instances				
Micro	\$23	\$0.012 per Hour	\$35	\$0.012 per Hour
High-Memory Reserved Instances				
Extra Large	\$272	\$0.169 per Hour	\$398	\$0.136 per Hour
Double Extra Large	\$544	\$0.338 per Hour	\$796	\$0.272 per Hour
Quadruple Extra Large	\$1088	\$0.676 per Hour	\$1592	\$0.544 per Hour
High-CPU Reserved Instances				
Medium	\$161	\$0.09 per Hour	\$243	\$0.079 per Hour
Extra Large	\$644	\$0.36 per Hour	\$972	\$0.316 per Hour

2013

Dynamic Pricing

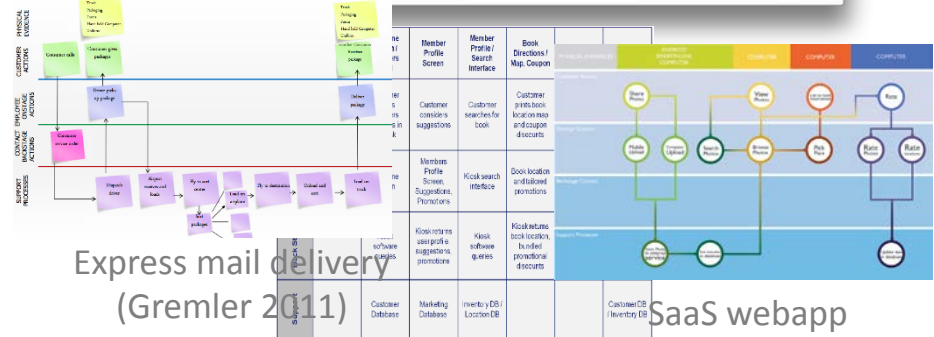
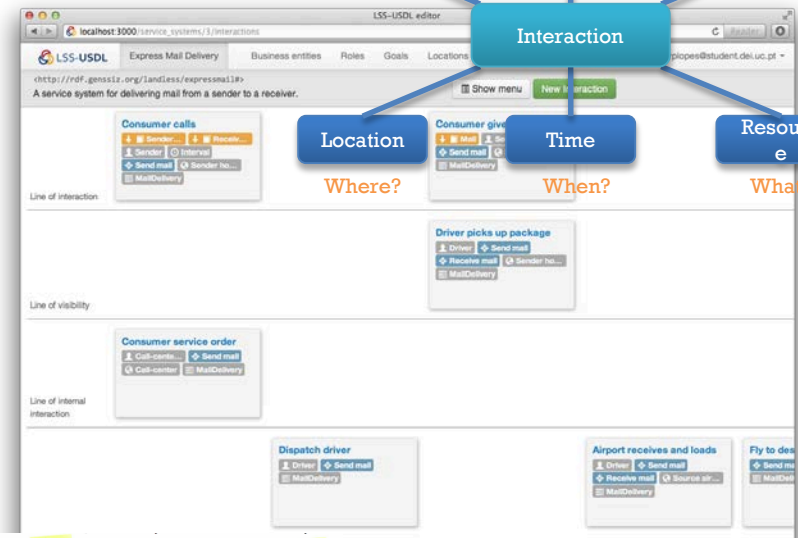
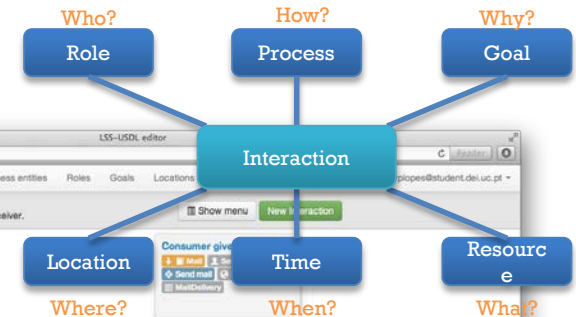


University of Coimbra, PT
The Open University, UK
SAP Research, DE

Service Systems

LSS USDL

- Service System model
 - Machine-understandable and processable
- Existing work
 - External perspective: WSDL, SAWSDL, USDL
 - Black-box
 - How can a service system be integrated with other services?
- Internal perspective
 - White-box approach and
 - Service management, optimization, and analytics.
 - How does the service works internally?



Express mail delivery (Gremier 2011)

Bookstore kiosk (Lopes et al. 2012)
(Glushko 2010)

Model @ GitHub

The screenshot shows the GitHub interface for the repository 'Genessiz / lss-usdl'. The repository is public and has 2 watchers, 0 stars, and 0 forks. The current branch is 'master'. The commit history shows a recent commit by 'rplopes' 9 days ago, titled 'Added vectors for the model figures', with the latest commit hash 'e033c34375'. Below the commit history, a list of files is shown, including 'images', 'README.md', 'lss-usdl-interactions.ttl', 'lss-usdl-resources.ttl', and 'lss-usdl.ttl'. The 'README.md' file is currently selected and displayed at the bottom of the page.

branch: master ▾ lss-usdl / model / + History

Added vectors for the model figures

rplopes authored 9 days ago latest commit e033c34375

images	Added vectors for the model figures	9 days ago
README.md	Added the model, use cases and editor	18 days ago
lss-usdl-interactions.ttl	Added the model, use cases and editor	18 days ago
lss-usdl-resources.ttl	Added the model, use cases and editor	18 days ago
lss-usdl.ttl	Added the model, use cases and editor	18 days ago

README.md

Getting Started Tutorial (6)

@prefix : <http://genssiz.org/lss-usdl/expressmail#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix lss-usdl: <http://genssiz.dei.uc.pt/lss-usdl#> .

:ExpressMailDelivery a lss-usdl:ServiceSystem;
 rdfs:label "Express Mail Delivery";
 rdfs:comment "A service system for delivering express mails";
 lss-usdl:hasInteraction :CustomerCalls,
 :CustomerDeliversPackages .

:CustomerCalls a lss-usdl:CustomerInteraction;
 rdfs:label "Customer calls";
 lss-usdl:hasGoal :SendMail;
 lss-usdl:isPerformedBy :Sender;
 lss-usdl:hasLocation :SenderHome .

:CustomerDeliversPackages a lss-usdl:CustomerInteraction;
 rdfs:label "Customer delivers packages";
 lss-usdl:isPerformedBy :Sender .

:SendMail a lss-usdl:Goal;
 rdfs:label "Send mail" .

:Sender a lss-usdl:Role;
 rdfs:label "Sender" .

:SenderHome a lss-usdl:Location;
 rdfs:label "Sender's home" .

Remote Access

```
from rdflib import Graph, Literal, BNode, RDF, URIRef, RDFS
from rdflib.namespace import FOAF, DC
import rdflib

g = Graph()
g.parse("https://raw.githubusercontent.com/rplopes/lss-usdl/master/use%20cases/1%20-%20Express%20Mail%20Delivery.ttl", format='n3')

print("--- printing LSS ---")
for lss in g.subjects(RDF.type, URIRef("http://genssiz.dei.uc.pt/lss-usdl#ServiceSystem")):
    print "Service Ssystem Name: ", lss.rsplit("#", 2)[1]
    for ss_description in g.objects(lss, RDFS.comment):
        print "Description:", ss_description

print "Interaction Points: "
for sub, obj in g.subject_objects(URIRef("http://genssiz.dei.uc.pt/lss-usdl#hasInteraction")):
    interaction = obj.rsplit("#", 2)[1]
    print interaction

qres = g.query(
    """PREFIX lss-usdl: <http://genssiz.dei.uc.pt/lss-usdl#hasInteraction>
    SELECT DISTINCT ?a ?b
    WHERE {
        ?a lss-usdl:hasInteraction ?b .
    }""")

for row in qres:
    print("%s hasInteraction %s" % row)
```

Service Costing

- Identify hidden patterns associated with costs for usdl concept usdl:InteractionPoints: who, how and what:
- Who (CQ1). Who is involved during the provisioning of a service or a particular interaction point?
- How (CQ2). How is an interaction conducted?
- What (CQ3). What resources were used during interactions?

Wolfgang Seiringer, Jorge Cardoso, Johannes Kunze von Bischhoffshausen, Service System Analytics: Cost Prediction, (PRO-VE'13) 14TH IFIP Working Conference on Virtual Enterprises, 30 Sep- 2 Oct 2013, Dresden, Germany.

Remote Service Querying

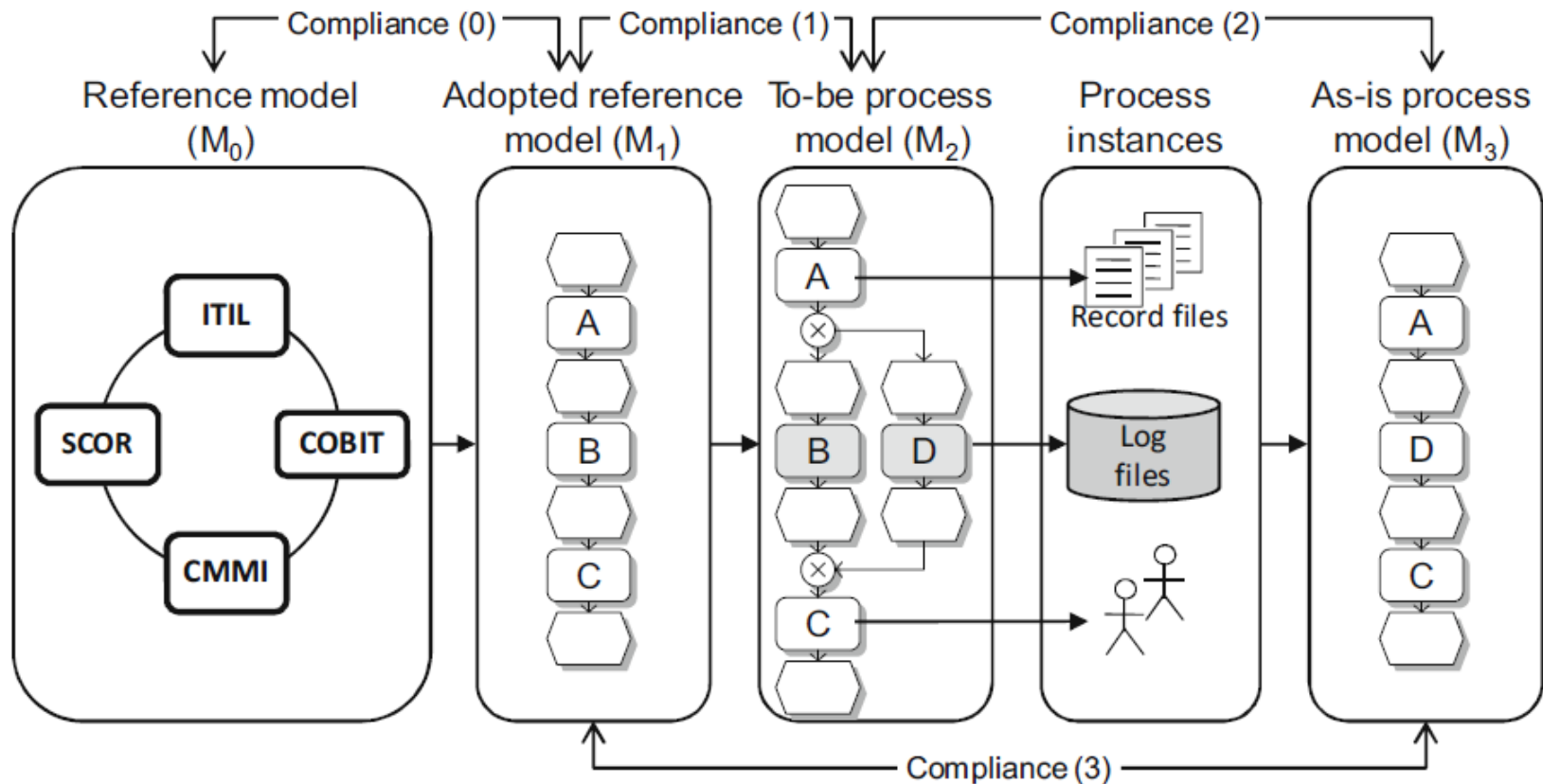
```
PREFIX usdl: <http://www.linked-usdl.org/ns/usdl-core>  
prefix gr: <http://purl.org/goodrelations/v1>  
PREFIX rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

```
SELECT ?ip ?interactionRole  
WHERE{  
  ?service gr:name ?name .  
  ?service usdl:hasInteractionPoint ?ip  
  ?ip usdl:hasInteractingEntity ?ie .  
  ?ie usdl:hasEntityType ?businessRol  
  ?ie usdl:hasInteractionRole ?interact  
  FILTER regex(?name, "Maintenance"  
}
```

Query selects all
interactionPoints
associated with
„Maintenance“

The obtained result provides
input for methods of service
analytics
Can be extended to other LINK
USDL elements

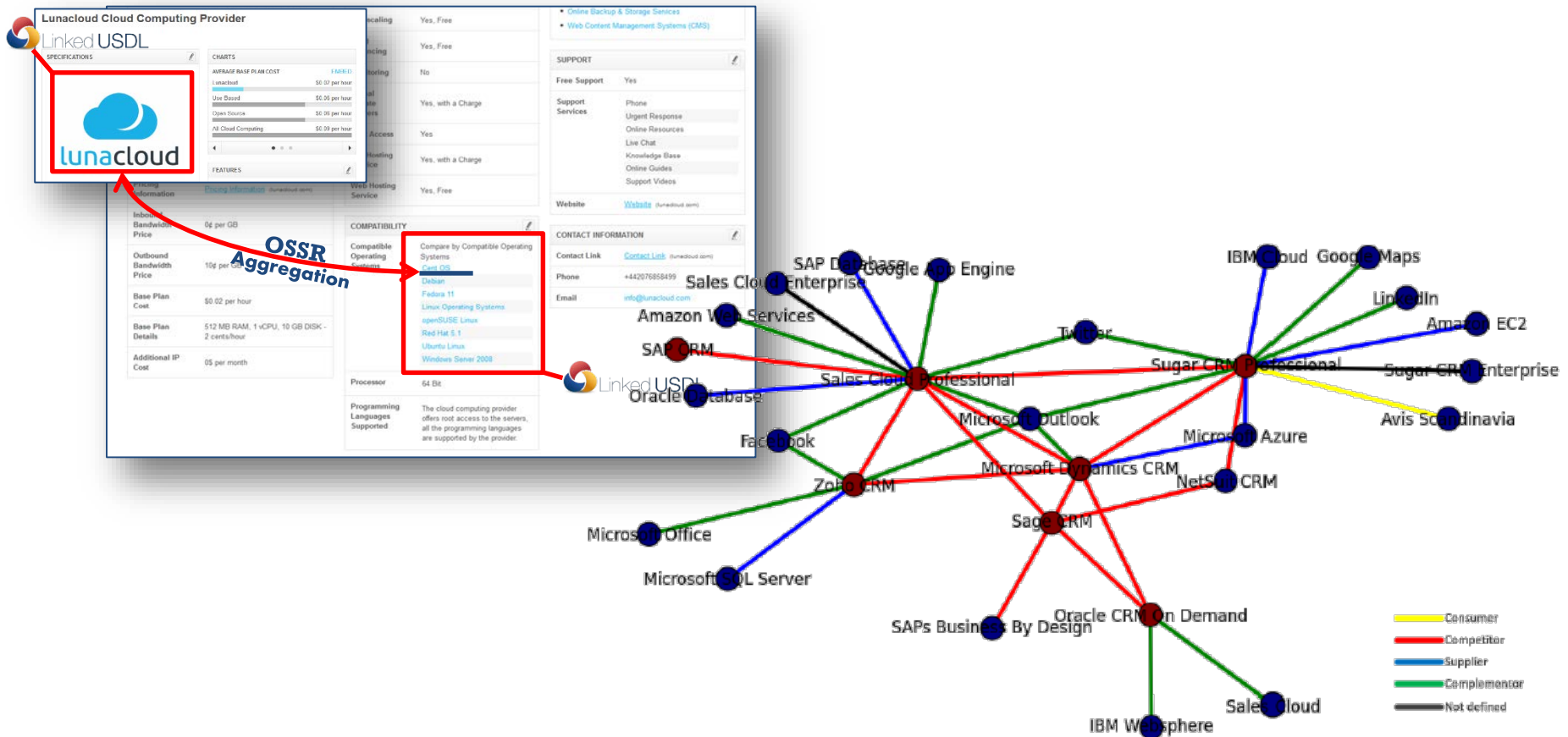
Process Compliance



Gerke, K.; Cardoso, J. and Claus, A. Measuring the Compliance of Processes with Reference Models. In 17th International Conference on Cooperative Information Systems (CoopIS 2009), Springer, Algarve, Portugal, 2009.

Service Networks

Linked USDL + OSSR = OSSN

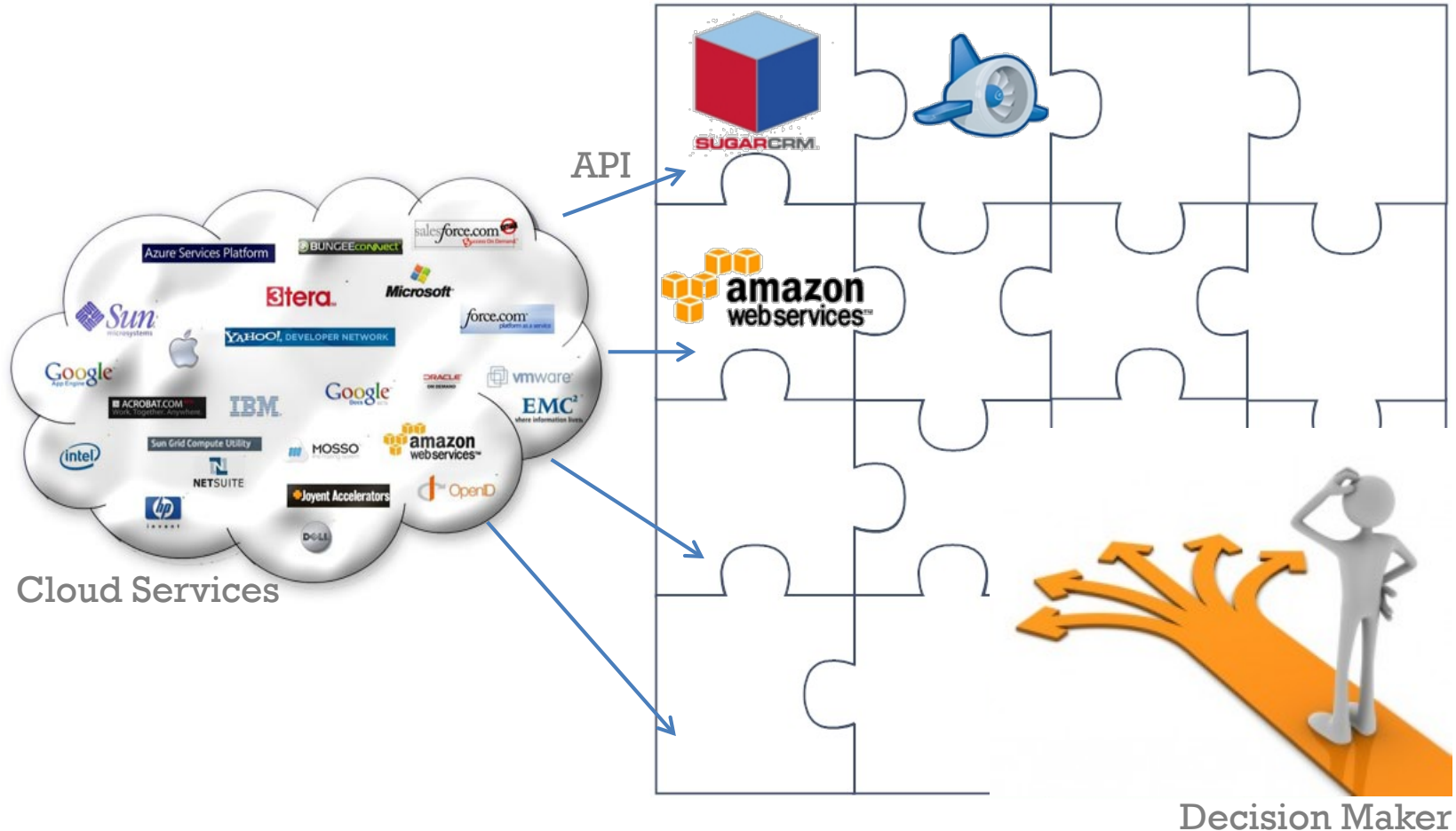


Cardoso, J. Modeling Service Relationships for Service Networks. In 4th International Conference on Exploring Service Science (IESS 1.3), pages 114-128, Springer, Porto, Portugal, LNBIP, Vol. 143, 2013.

Applications (2013)

Linked USDL and Cloud Services

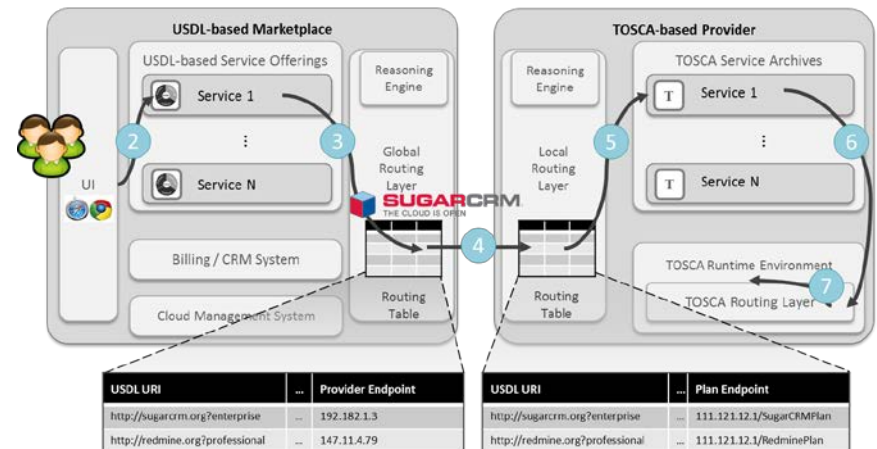
Consider cost, compatibility, space, speed, etc.



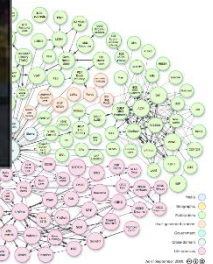
Applications (2013)

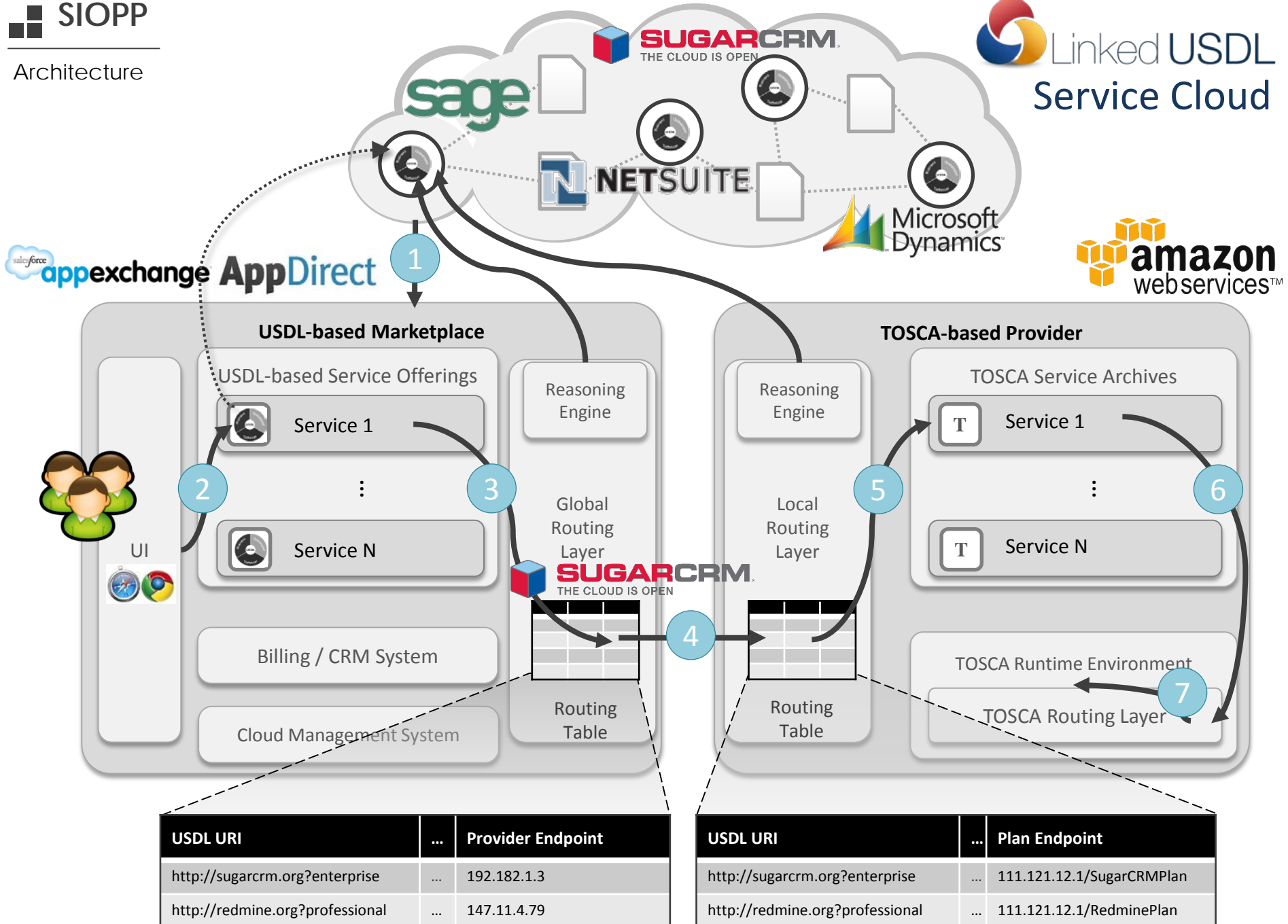
- Cloud Services (USDL & TOSCA)
 - University of Stuttgart, DE
- Could Service Aggregation
 - INESC, PT
- Service System Costing
 - Vienna Univ. of Technology, AT
 - Karlsruhe Inst. of Technology, DE
- ITIL Service Management
 - Portugal Telecom, PT
- Linked Open Data Integration
 - Portugal Telecom, PT
- Process Navigation
 - University of Bayreuth, DE

USDL & TOSCA Integration

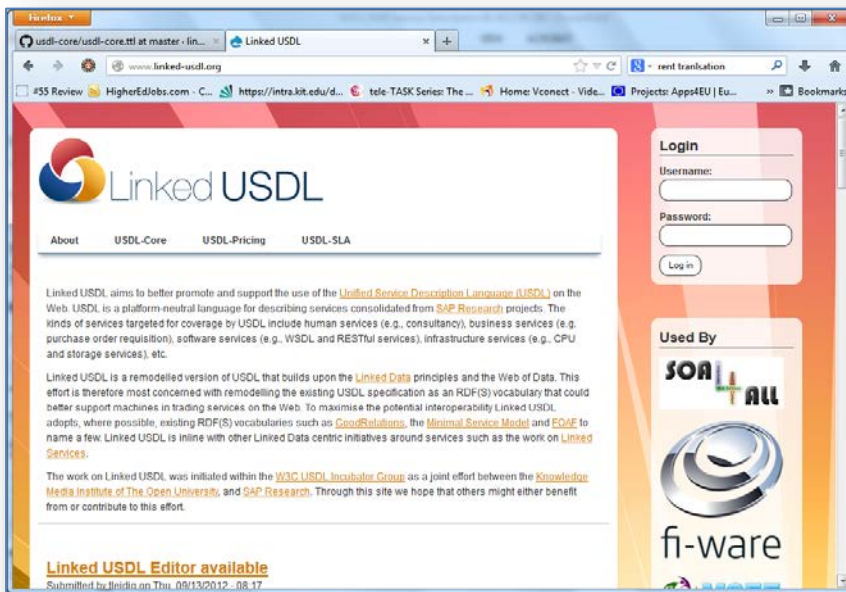


Linked Open Data Integration

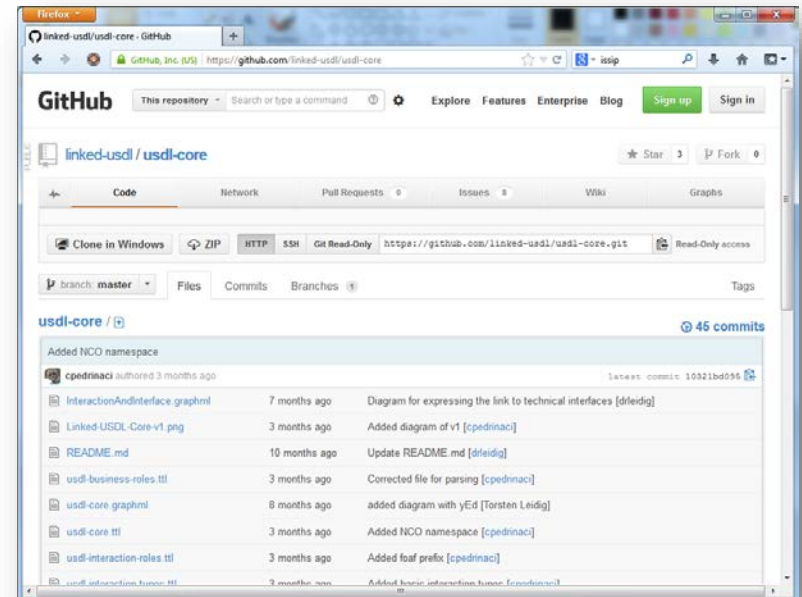




Resources



<http://www.linked-usdl.org/>

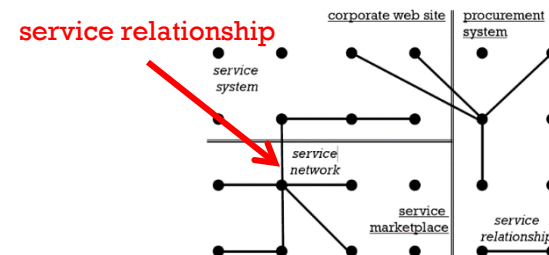
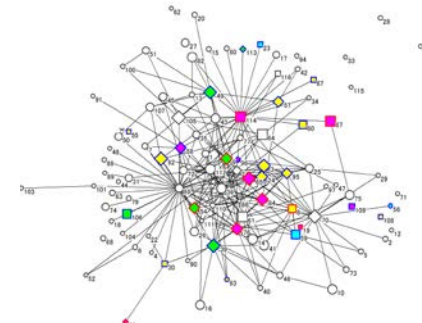
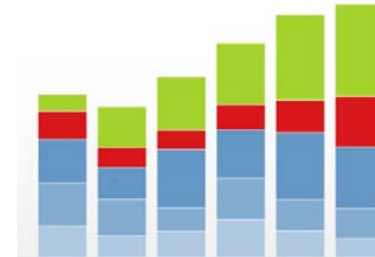


<https://github.com/linked-usdl/>
<https://github.com/Genssiz>

Linked USDL

Next Steps (2014)

- **Service Analytics**
 - Service system mining
 - Process mining
 - Data mining
- **Service Network Analysis**
 - Automated reconstruction
 - Domain-specific metrics
- **Service Relationships**
 - Evidence from Social Networks
 - Text mining techniques
 - Co-occurrence analysis



Useful links

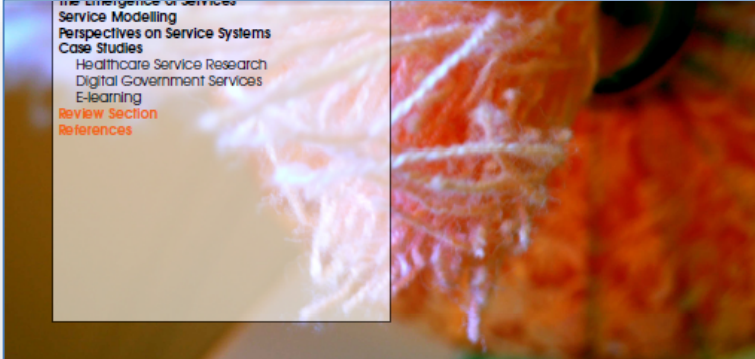
- [LSS-USDL Editor](#): Open source repository of the LSS-USDL graphical editor.
- [USDL Incubator Group](#): LSS -USDL is part of the research for service systems by the USDL research group.
- [Linked USDL](#): Similar project, focusing on service descriptions for customers. The third use case found in LSS-USDL's repository shows a service system modeled both in LSS-USDL and Linked USDL.
- [Linked USDL core](#): Repository for the core module of Linked USDL. The other modules may be found under the same Github profile.
- [Semantic Web](#): Technologies such as RDF are a core component of LSS-USDL.

Textbook on Service Systems



Contents

1	Fundamentals	13
1.1	The Emergence of Services	15
1.2	Service Modelling	15
1.3	Perspectives on Service Systems	17
1.4	Case Studies	21
1.4.1	Healthcare Service Research	21
1.4.2	Digital Government Services	25
1.4.3	E-learning	25
	Review Section	27
	References	28



1 — Fundamentals

Summary

This chapter provides an overview of the origins of services. Two important views are examined: services as a transformation process and services as a set of resources. The differences and complementarity between services and goods are examined. Since the development of digital services is rapidly emerging, the relationships between services, software, and ICT are framed by presenting a classification framework. The last sections present the running use cases that will be used throughout the textbook and the six perspective that will be used to study each use case.

Learning Objectives

1. Understand the historical evolution of services and their importance nowadays for societies.
2. Analyse the various views on services based on the emphases placed on processes and resources.
3. Explain how services from various industry domains can benefit from a service system discipline.
4. Describe various perspectives which can be taken to study services using scientific and systematic approaches.

Thank You
for Listening