INFO116

Semantics for Cloud Computing

Ch. 8

Cloud Computing

- Utility style, pay-as-you-go, elastic computing paradigm
- Model for enabling convenient, on-demand network access to a shared pool of configurable computing resources
- Problem of interoperability caused by heterogeneity and vendor lock-ins
- Can a little semantics help?

Service Models

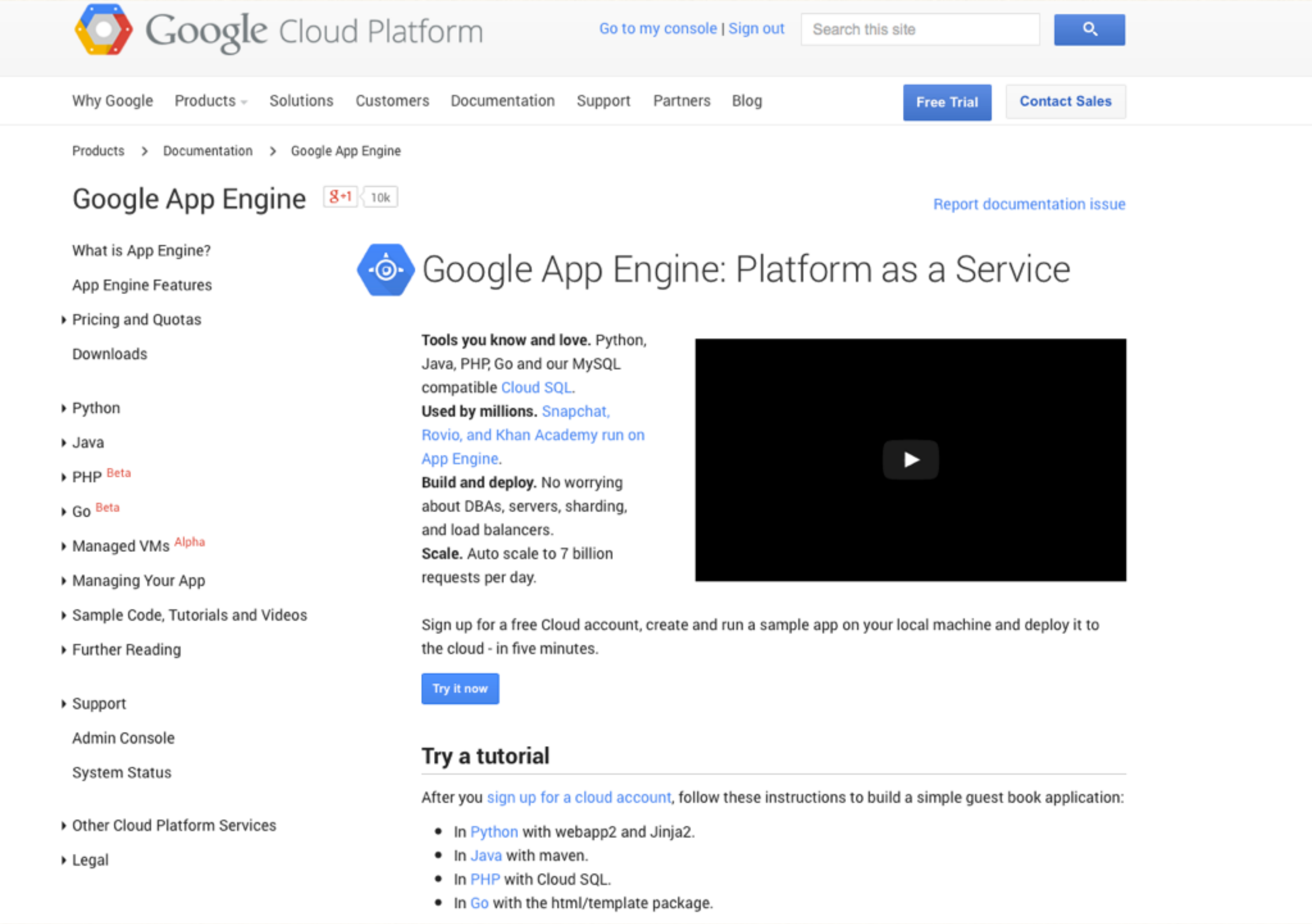
- Not just Dropbox style file storage
- NIST (National Institute of Standards and Technology) defines three service models:
  - Cloud Software as a Service (SaaS)
  - Cloud Platform as a Service (PaaS)
  - Cloud Infrastructure as a Service (IaaS)

SaaS

- The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

PaaS

- The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application hosting environment.



Google App Engine: Platform as a Service

Tools you know and love. Python, Java, PHP, Go and our MySQL compatible Cloud SQL.
Used by millions. Snapchat, Rovio, and Khan Academy run on App Engine.
Build and deploy. No worrying about DBAs, servers, sharding, and load balancers.
Scale. Auto scale to 7 billion requests per day.

Sign up for a free Cloud account, create and run a sample app on your local machine and deploy it to the cloud - in five minutes.

Try a tutorial

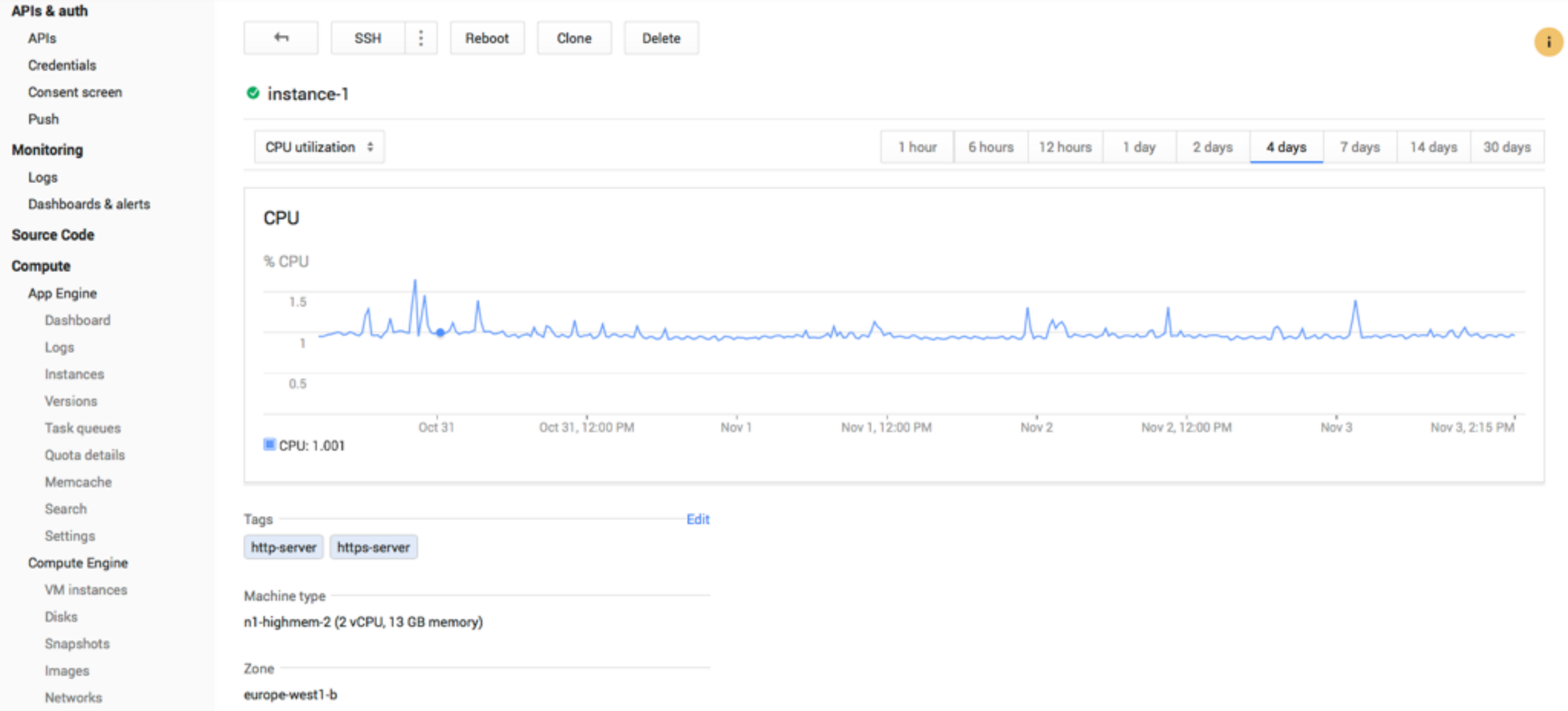
After you sign up for a cloud account, follow these instructions to build a simple guest book application:

- In Python with webapp2 and Jinja2.
- In Java with maven.
- In PHP with Cloud SQL.
- In Go with the html/template package.

IaaS

- The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Google developer console



instance-1

CPU: 1.001

Tags

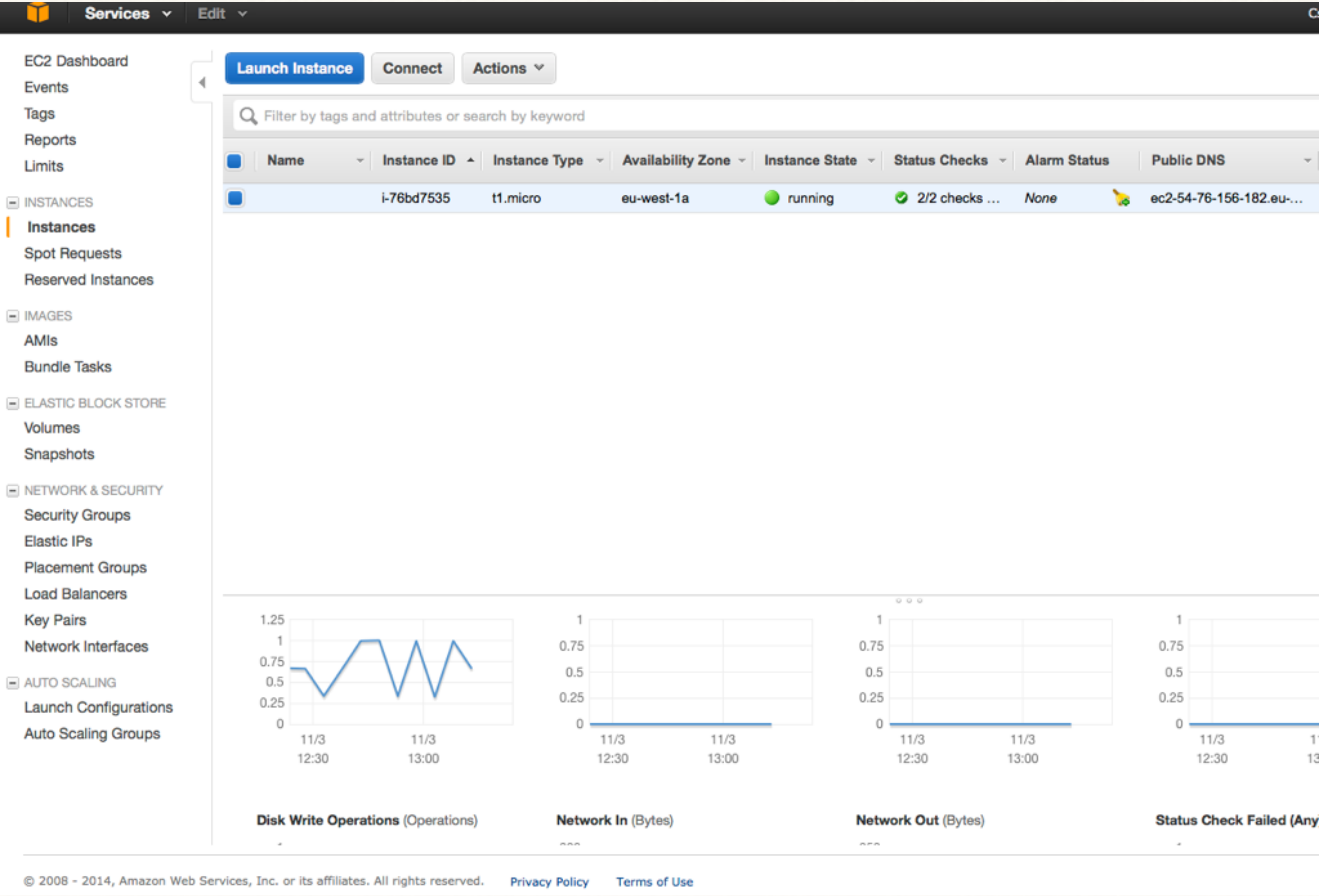
http-server https-server

Machine type

n1-highmem-2 (2 vCPU, 13 GB memory)

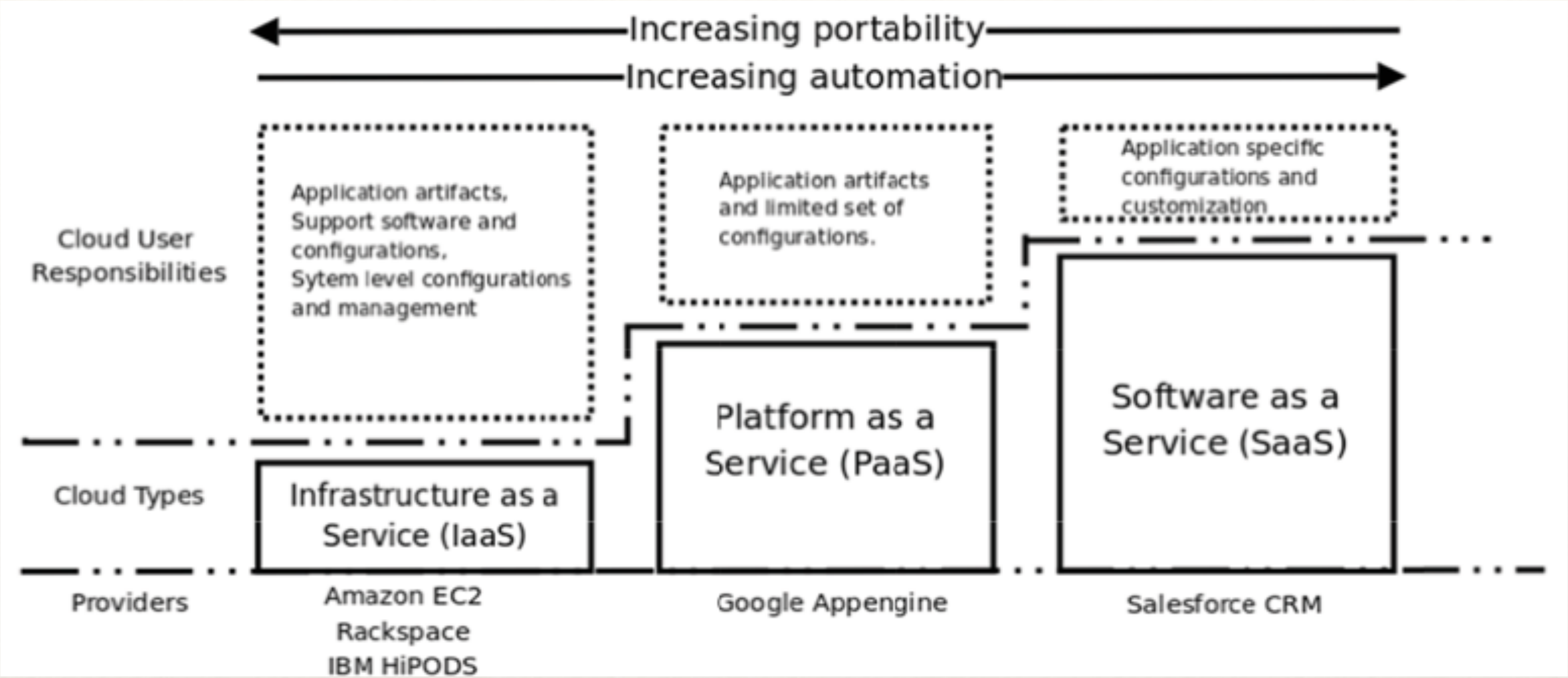
Zone

europe-west1-b



☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
☐		i-76bd7535	t1.micro	eu-west-1a	● running	✓ 2/2 checks ...	None	ec2-54-76-156-182.eu-...

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Increasing portability

Increasing automation

Cloud User Responsibilities

Application artifacts, Support software and configurations, Sytem level configurations and management

Application artifacts and limited set of configurations.

Application specific configurations and customization

Cloud Types

Infrastructure as a Service (IaaS)

Platform as a Service (PaaS)

Software as a Service (SaaS)

Providers

Amazon EC2
Rackspace
IBM HiPODS

Google Appengine

Salesforce CRM

Deploying a Cloud Service

- Select which cloud to use. Each cloud vendor exposes these details in different formats and at different levels of granularity.
- Technical aspects (service interface, scaling configuration, etc.) and the workflow of the particular cloud provider.
- Choice of programming language and limitations in the application runtime

Challenges

- Changing providers involves heterogeneity
  - Software
  - Data
- Changing services
  - e.g. PaaS -> IaaS

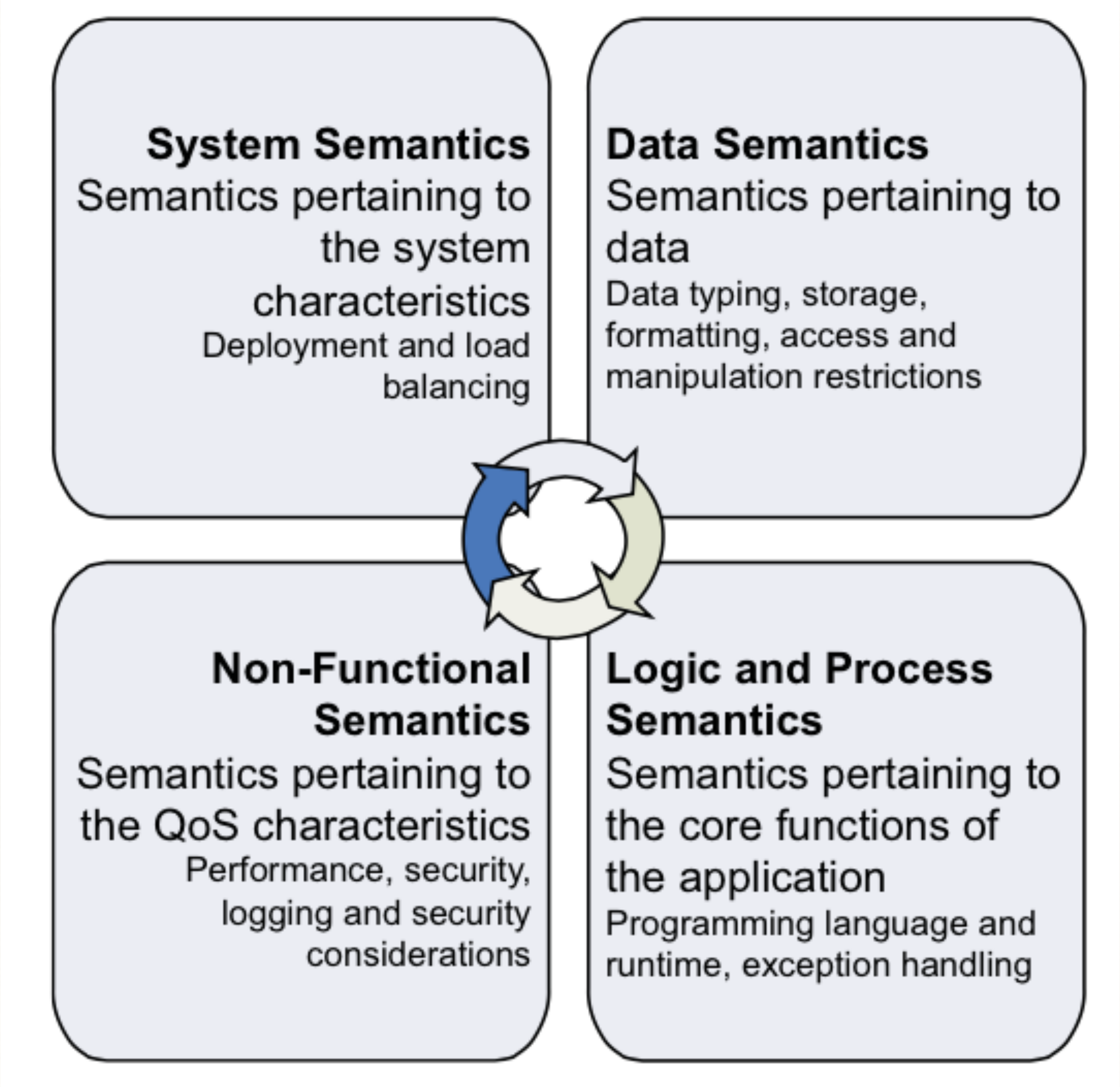
The Role of Semantics

- Many of the core data and services causing these heterogeneities follow the same semantic concepts
  - almost all the IaaS clouds follow conceptually similar workflows when allocating resources, although the actual service implementations and tools differ significantly
  - the modeling space for a PaaS is a subset of the modeling space for IaaS, from a semantic perspective

Semantic Annotation

- Type of semantics
  - data
  - logic/process
  - non-functional (performance, security, etc.)
  - system (load balancing, etc.)
- Level of abstraction (EER vs. Tables, UML vs. Java code)
- Software life cycle state (EER vs. Tables, UML vs. Java code)

Four Levels of Semantics

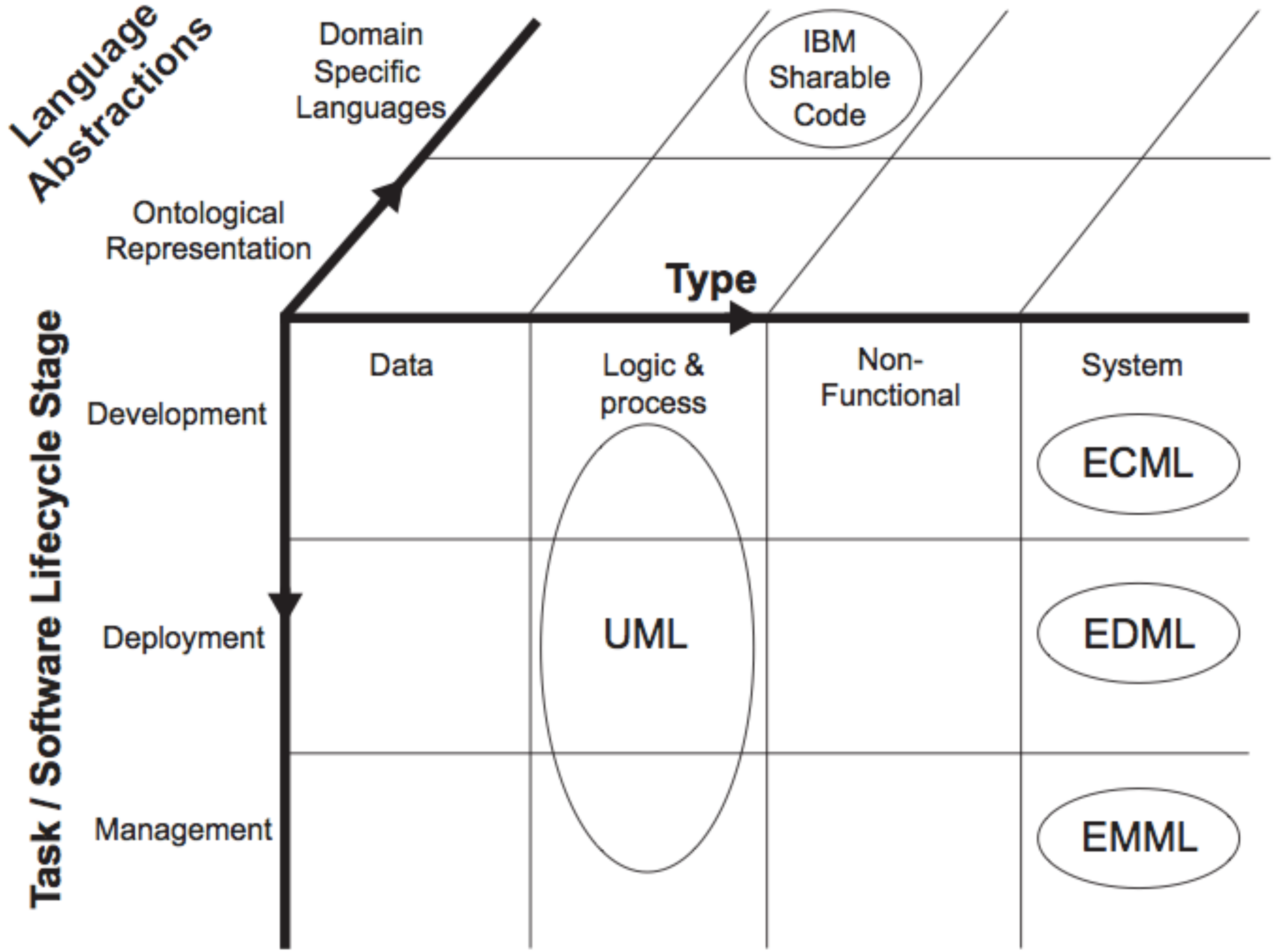


System Semantics
Semantics pertaining to the system characteristics
Deployment and load balancing

Data Semantics
Semantics pertaining to data
Data typing, storage, formatting, access and manipulation restrictions

Non-Functional Semantics
Semantics pertaining to the QoS characteristics
Performance, security, logging and security considerations

Logic and Process Semantics
Semantics pertaining to the core functions of the application
Programming language and runtime, exception handling



Language Abstractions

Domain Specific Languages

Ontological Representation

Type

IBM Sharable Code

Task / Software Lifecycle Stage

Data

Logic & process

Non-Functional

System

Development

Deployment

Management

UML

ECML

EDML

EMML

Four types of semantics (1/4)

- Data Semantics
  - Definitions of data structures, relationships across multiple data structures
  - Restrictions on the access of some of the data items
  - Semantic modelling of data to provide a platform independent data representation
- Ontologies

Semantics for Data Modelling

- RDF data modelling to overcome data lock-in
- Modelling data in RDF can support generation of different target representations, and in some cases, even code for data access layer of the application

Four types of semantics (2/4)

- Logic and process semantics
  - Core functionality (commonly referred to as the business logic) of an application
- Ontologies?

Four types of semantics (3/4)

- Non-functional semantics
  - quality of service (QoS)
  - access control
  - logging
    - e.g. INFO (informational content) vs. ERROR (only errors)
- Ontologies?

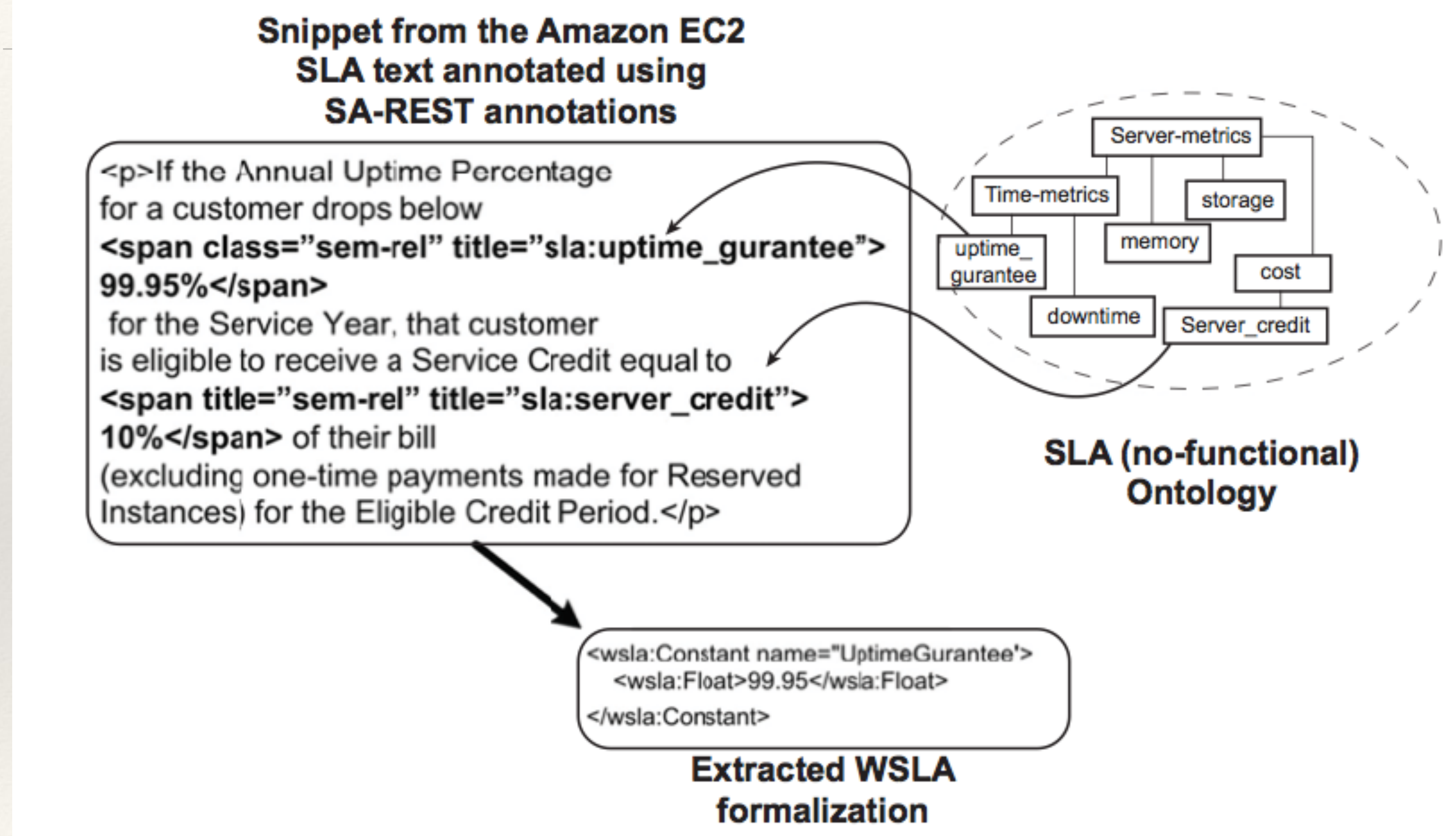
Four types of semantics (4/4)

- System semantics
  - Deployment descriptions and dependency management
  - Runtime considerations
  - Service interfaces
    - different from vendor to vendor
    - generic operational models would play a key role in consolidating these APIs
- Ontologies?

SEMANTICS FOR SERVICE ENRICHMENT

- Clouds normally provide Web services to manipulate resources
- SAWSDL standardizes semantic annotations in WSDL service descriptions
    - cloud space is still evolving and attaching formalisations via annotations is compatible with this trend

Example: embedded sla



Snippet from the Amazon EC2
SLA text annotated using
SA-REST annotations

```
<p>If the Annual Uptime Percentage
for a customer drops below
<span class=”sem-rel” title=”sla:uptime_gurantee”>
99.95%</span>
 for the Service Year, that customer
is eligible to receive a Service Credit equal to
<span title=”sem-rel” title=”sla:server_credit”>
10%</span> of their bill
(excluding one-time payments made for Reserved
Instances) for the Eligible Credit Period.</p>
```

SLA (no-functional)
Ontology

```
<wsla:Constant name="UptimeGurantee'>
   <wsla:Float>99.95</wsla:Float>
</wsla:Constant>
```

Extracted WSLA
formalization