

Introduction to service-based, cloud-based and fog computing

Antonio Brogi

Dipartimento di Informatica
Università di Pisa

This series of seminars aims at introducing some of the main aspects in service-based, cloud-based and fog computing by covering both modelling and analysis aspects (e.g., app deployment, management, QoS prediction) and implementation aspects (e.g., Docker, microservices, DevOps practices).

The seminars will also illustrate some of the ongoing research carried on by the SoCC research group, and they will point to some open problems on which the possible cooperation of seminar attendees is welcome.



Service-based computing



Everything as a service

- Economy trend: from “*goods*” to “*services*”
- A few (random) examples



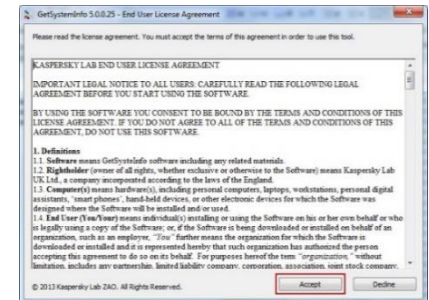
Service contracts

- Customers do not (want to) know how the service they use is actually implemented



- Customers (should) choose whether to use a service or not on the basis of the *contract* that is (or should) be exposed by the service provider

... but ... do customers
read contracts at all?



QoS matters, though



vs.



QoS matters (money is not all)

To store **all the data** of my school, I have chosen the storage service offered by the new local software company *StoreForLess*.

We will save **50 USD** per year if compared for instance with using Amazon S3!



Service Level Agreements



Effective Date: June 1, 2014



This Amazon S3 Service Level Agreement ("SLA") is a policy governing the use of Amazon Simple Storage Service ("Amazon S3") under the terms of the Amazon Web Services Customer Agreement (the "AWS Agreement") between Amazon Web Services, Inc. and its affiliates ("AWS", "us" or "we") and users of AWS' services ("you"). This SLA applies separately to each account using Amazon S3. Unless otherwise provided herein, this SLA is subject to the terms of the AWS Agreement and capitalized terms will have the meaning specified in the AWS Agreement. We reserve the right to change the terms of this SLA in accordance with the AWS Agreement.

Service Commitment

AWS will use commercially reasonable efforts to make Amazon S3 available with a Monthly Uptime Percentage (defined below) of at least 99.9% during any monthly billing cycle (the "Service Commitment"). In the event Amazon S3 does not meet the Service Commitment, you will be eligible to receive a Service Credit as described below.

Definitions

- "Error Rate" means: (i) the total number of internal server errors returned by Amazon S3 as error status "InternalError" or "ServiceUnavailable" divided by (ii) the total number of requests during that five minute period. We will calculate the Error Rate for each Amazon S3 account as a percentage for each five minute period in the monthly billing cycle. The calculation of the number of internal server errors will not include errors that arise directly or indirectly as a result of any of the Amazon S3 SLA Exclusions (as defined below).
- "Monthly Uptime Percentage" is calculated by subtracting from 100% the average of the Error Rates from each five minute period in the monthly billing cycle.
- A "Service Credit" is a dollar credit, calculated as set forth below, that we may credit back to an eligible Amazon S3 account.

Service Credits

Service Credits are calculated as a percentage of the total charges paid by you for Amazon S3 for the billing cycle in which the error occurred in accordance with the schedule below.

Monthly Uptime Percentage

Service Credit Percentage

Equal to or greater than 99% but less than 99.9%	10%
Less than 99%	25%

Q: How much *Service Credit Percentage* will you get if S3 was not available for two whole days (April 2nd and 3rd)?

Q: How many "InternalError" or "ServiceUnavailable" errors did you get? And When?

A: I tried to access S3 at 9am, 11am, 1pm, and 3pm both days, getting a "ServiceUnavailable" error every time

- 8 "five minute periods" with Error Rate = 100%
- Remaining $(12 \times 24 \times 30) - 8$ five minute periods with Error Rate = 0%
- "monthly Uptime Percentage" = $100\% - (8 / (12 \times 24 \times 30) \times 100\%) = 99,90740741\%$

Microsoft Services Agreement

Effective July 31, 2014

6. Services disruptions and backup

We strive to keep the Services up and running; however, all online services suffer occasional disruptions and outages. **You should regularly back up the Content that you store on the Services.** Having a regular backup plan and following it can help you prevent loss of your Content. We make an effort to provide detailed service status on the Service status website (<http://status.live.com>).

11. NO WARRANTIES

MICROSOFT, AND OUR AFFILIATES, RESELLERS, DISTRIBUTORS, AND VENDORS, MAKE NO WARRANTIES, EXPRESS OR IMPLIED, GUARANTEES OR CONDITIONS WITH RESPECT TO YOUR USE OF THE SERVICES. YOU UNDERSTAND THAT USE OF THE SERVICES IS AT YOUR OWN RISK AND THAT WE PROVIDE THE SERVICES ON AN “AS IS” BASIS “WITH ALL FAULTS” AND “AS AVAILABLE.” **MICROSOFT DOESN'T GUARANTEE THE ACCURACY OR TIMELINESS OF INFORMATION AVAILABLE FROM THE SERVICES.** TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAW, WE EXCLUDE ANY IMPLIED WARRANTIES, INCLUDING FOR MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, WORKMANLIKE EFFORT, AND NON-INFRINGEMENT. YOU MAY HAVE CERTAIN RIGHTS UNDER YOUR LOCAL LAW. NOTHING IN THIS AGREEMENT IS INTENDED TO AFFECT THOSE RIGHTS, IF THEY ARE APPLICABLE. YOU ACKNOWLEDGE THAT COMPUTER AND TELECOMMUNICATIONS SYSTEMS AREN'T FAULT-FREE AND OCCASIONAL PERIODS OF DOWNTIME OCCUR. **WE DON'T GUARANTEE THE SERVICES WILL BE UNINTERRUPTED, TIMELY, SECURE, OR ERROR-FREE OR THAT CONTENT LOSS WON'T OCCUR.**



For content that is covered by intellectual property rights, like photos and videos (IP content), you specifically give us the following permission, subject to your privacy and application settings: you grant us a non-exclusive, transferable, sub-licensable, royalty-free, worldwide license to use any IP content that you post on or in connection with Facebook (IP License). This IP License ends when you delete your IP content or your account unless your content has been shared with others, and they have not deleted it.

Service Level Agreements

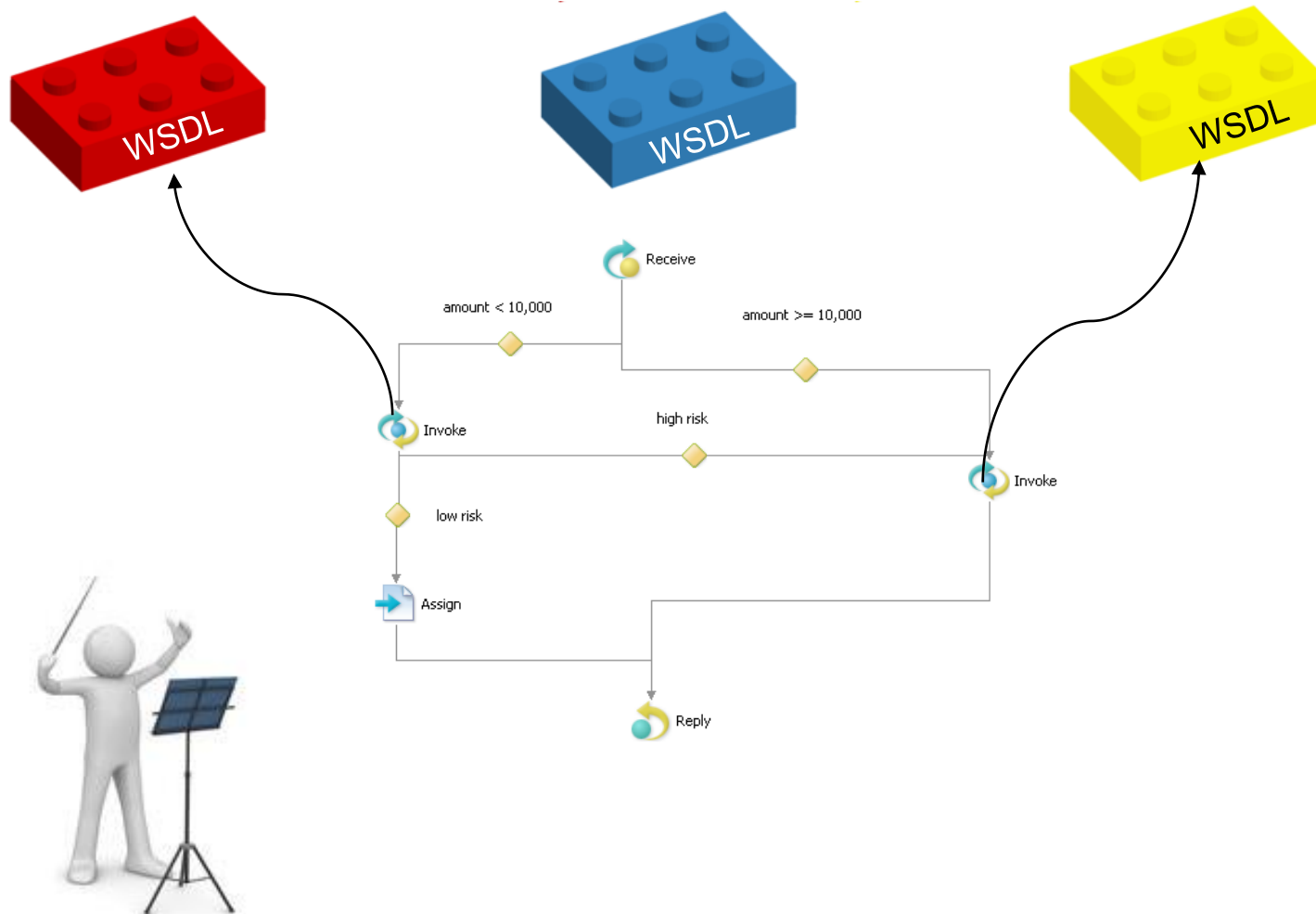


Which reliability can we guarantee for our app?



Service-oriented computing

- Idea: Use software services to develop distributed applications rapidly and at low cost



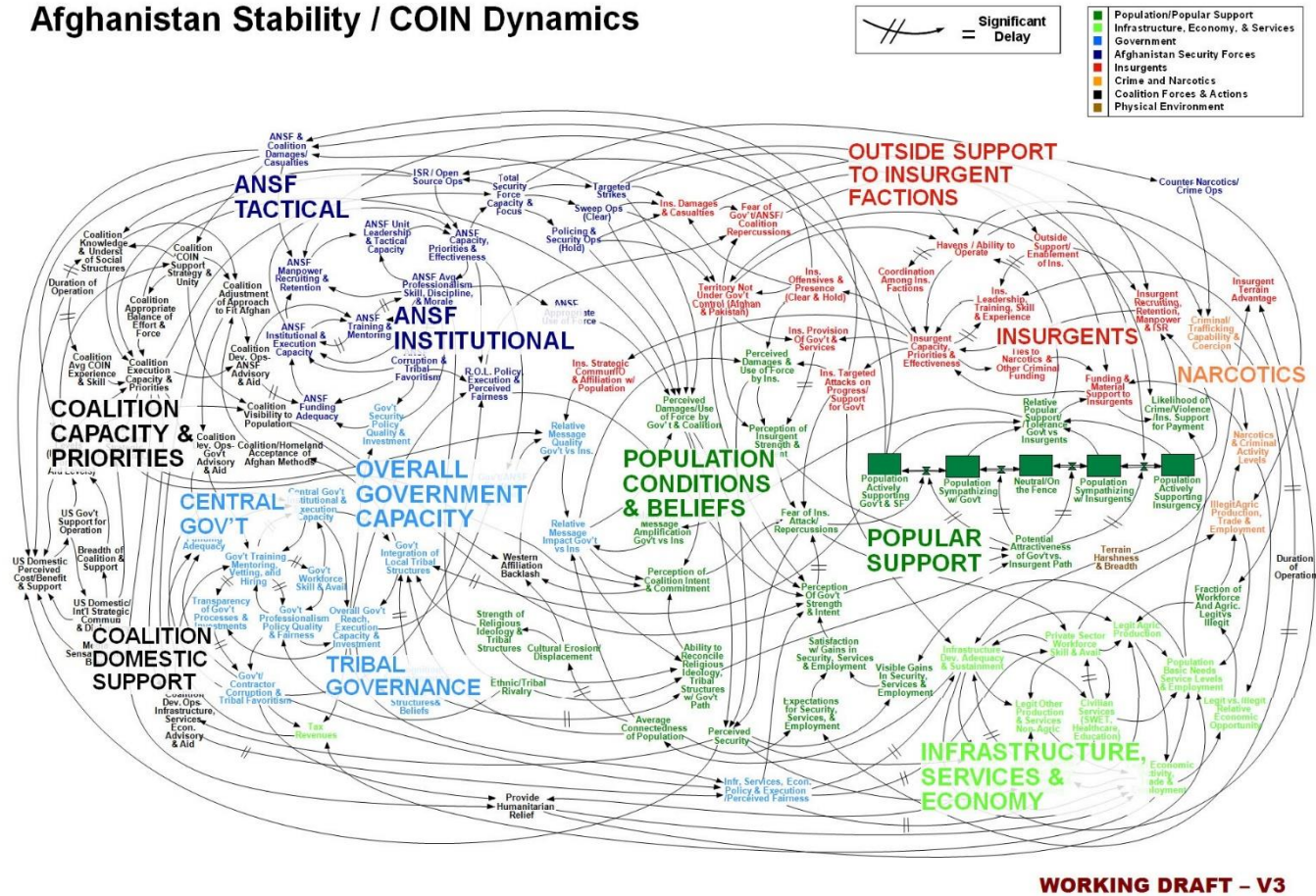
Service-oriented computing



- The QoS of a service-based app does depend on the QoS of the services it invokes
 - *What is the QoS of app A invoking S1 and S2?*
 - *What happens if S1 is replaced by S1' ?*
- **QoS prediction** difficult because of
 - *different results of service invocations*
 - *non-determinism in composition workflow*
 - *correlations & dependencies among activities*



Afghanistan Stability / COIN Dynamics



PA Consulting Group
© PA Knowledge Limited 2009

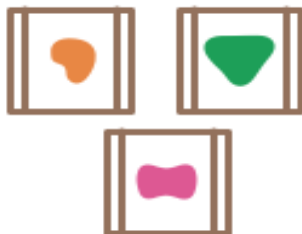
Page 22

WORKING DRAFT – V3

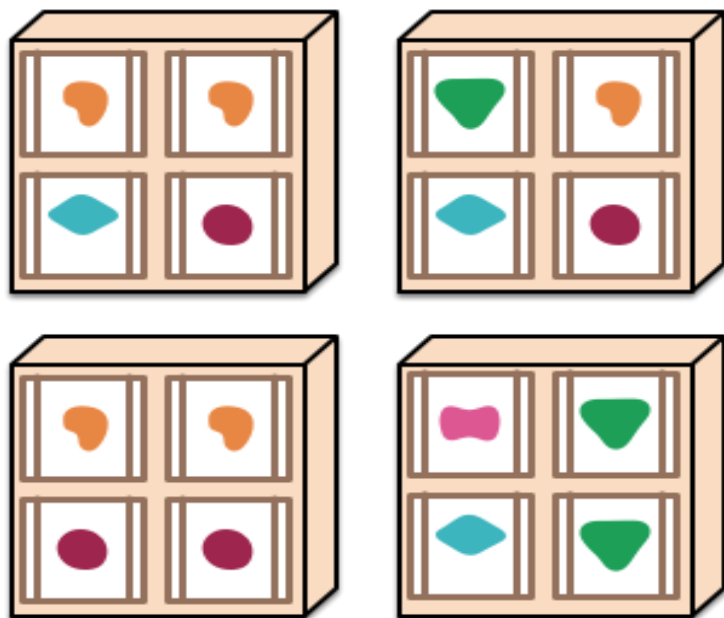
Correctness of service compositions?

Service-oriented computing

A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.



“Micro”services
independently
deployable and scalable

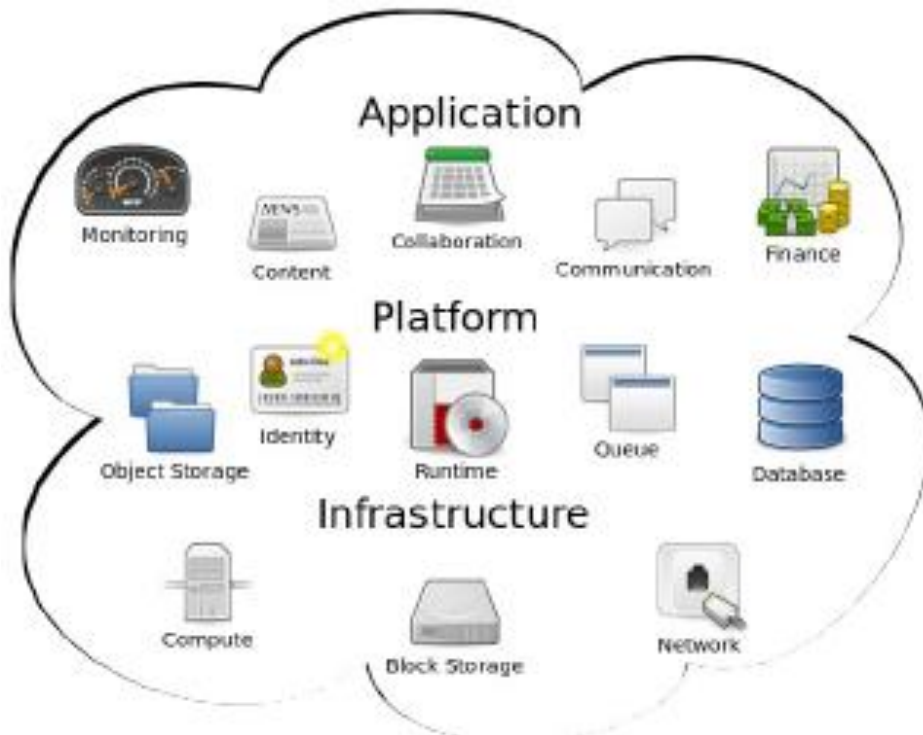


Service-based computing

Cloud-based computing



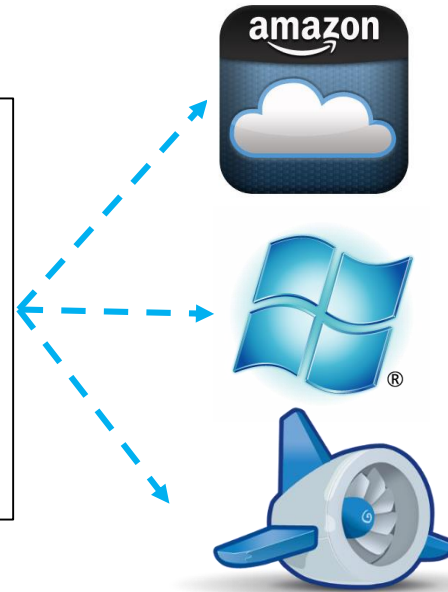
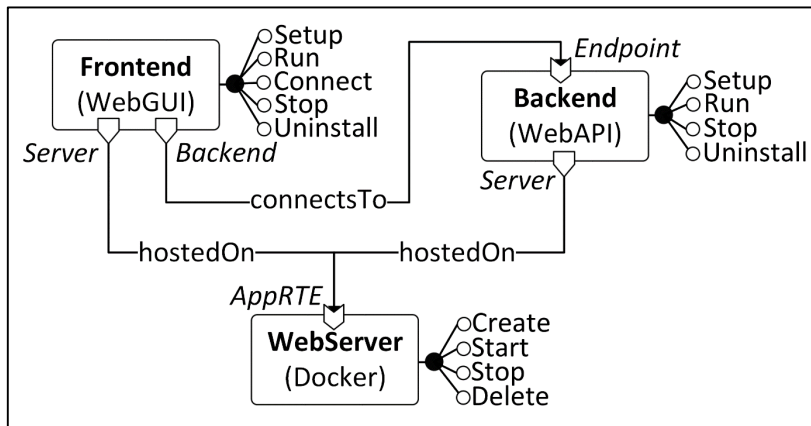
Cloud computing



Management of cloud-based apps



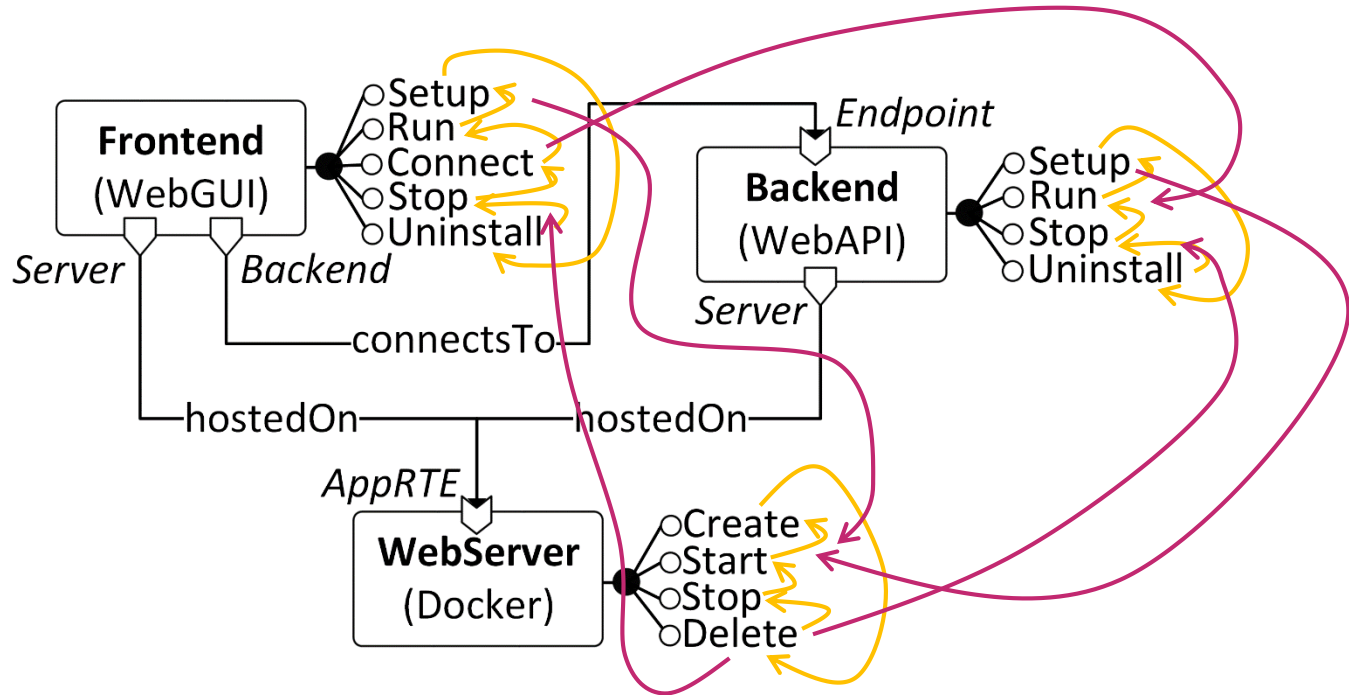
How to flexibly manage complex composite applications over heterogeneous cloud platforms?



OASIS  TOSCA



Management of cloud-based apps



How to manage composite applications
by suitably taking into account
intra- and *inter-*component dependencies?

Management of cloud-based apps

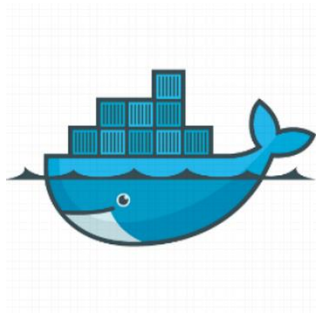
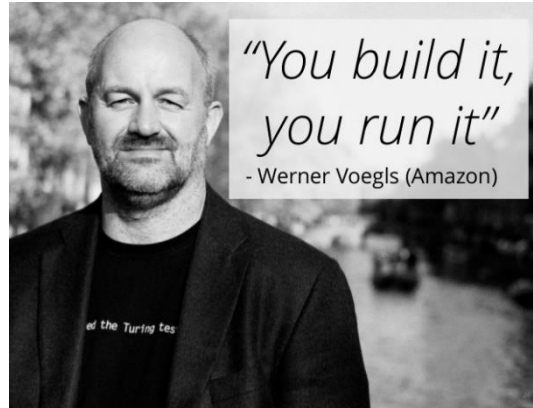
How to manage composite applications
by suitably taking into account
intra- and *inter-*component dependencies?



How to manage (unexpected) faults and hard recover?

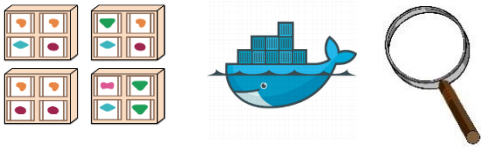


DevOps/Containers



- Building, shipping, and running applications, together with their dependencies, in lightweight virtual environments (containers)
- Docker containers run from Docker images, which are read-only templates used to create them

DevOps/Containers



Service-based computing

Cloud-based computing

Fog computing

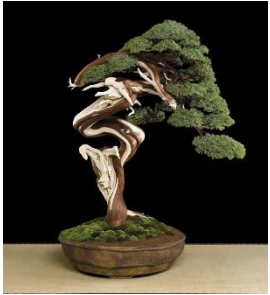


IoT and Cloud Computing



- The Cloud alone cannot support the **IoT momentum**
- Need to **filter** and **process** data *before* the Cloud

So much data, really?



My IoT Setup

1	Umidità del terreno	SIZE: 8 bytes	1min	×
1	pH Sensor	SIZE: 8 bytes	1min	×
1	Thermometer	SIZE: 8 bytes	1h	×
1	Water Quality Monitor	SIZE: 56 bytes	1h	×
1	Videocamera	SIZE: 56250 bytes	1s	×

Add another Thing +

Resources Used

Data Transfer
145800.74
MB / month

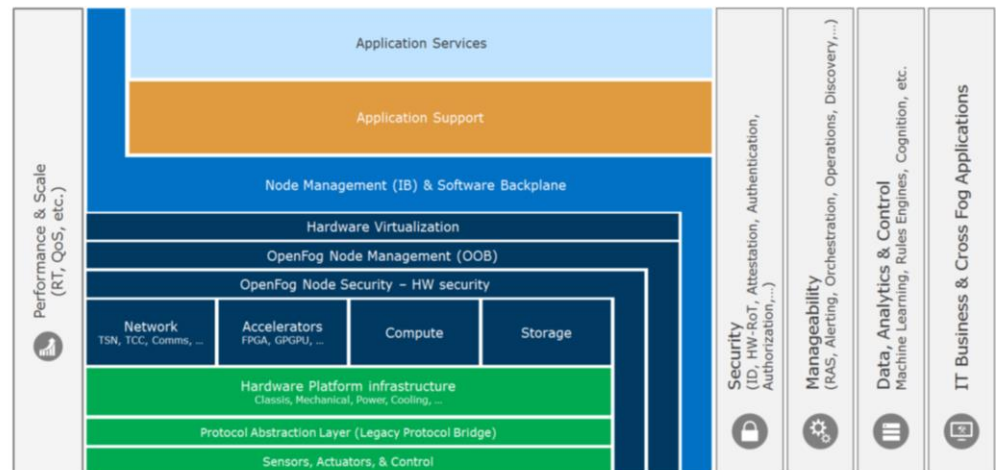
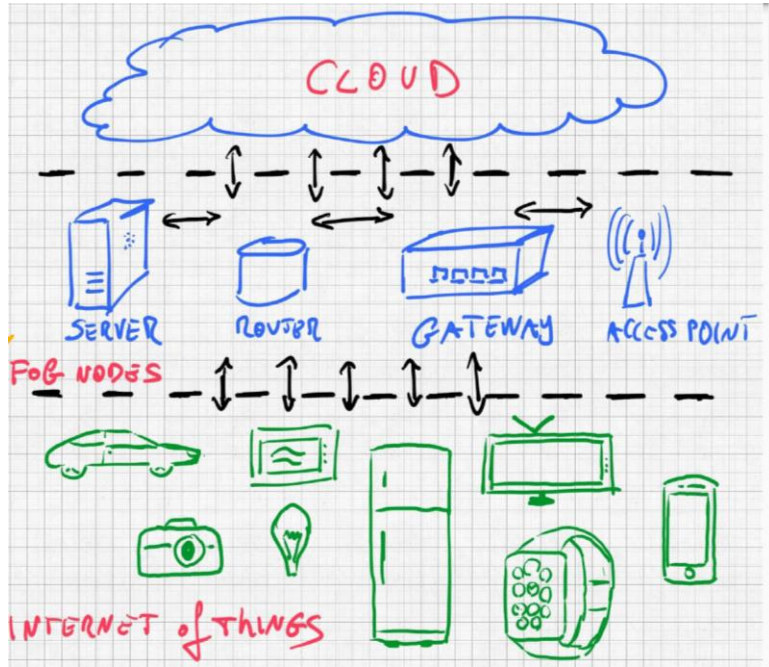
146 GB/month

Cloud Storage
437402.21
MB / month

437 GB/month
5.2 TB/year

Updates
~2.68 M
per month

Fog Computing



Fog Features



QoS-awareness

- App deployments dynamically adapt to the **state** of the network



Location-awareness

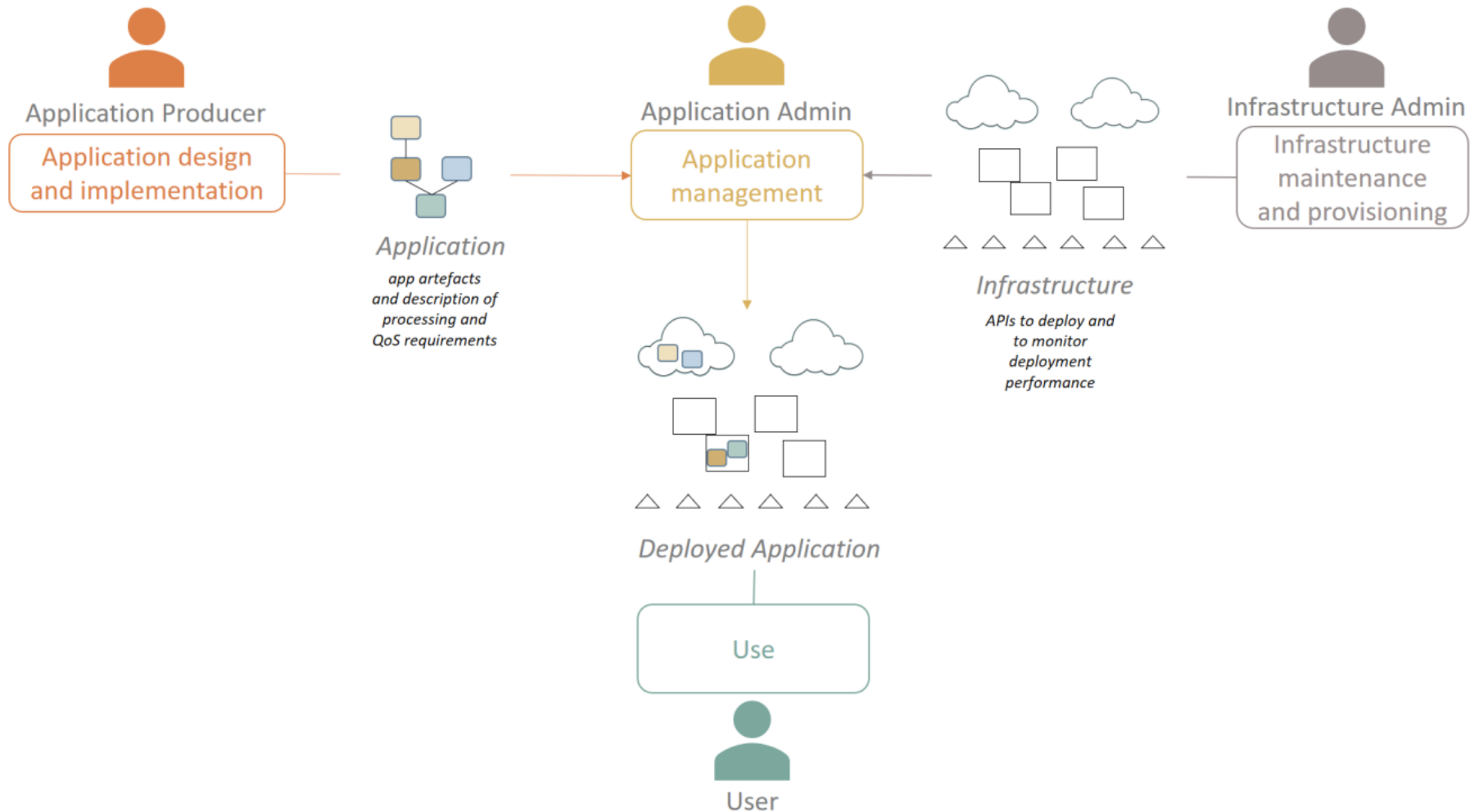
- **Position** is known so to handle fluid and mobile computation



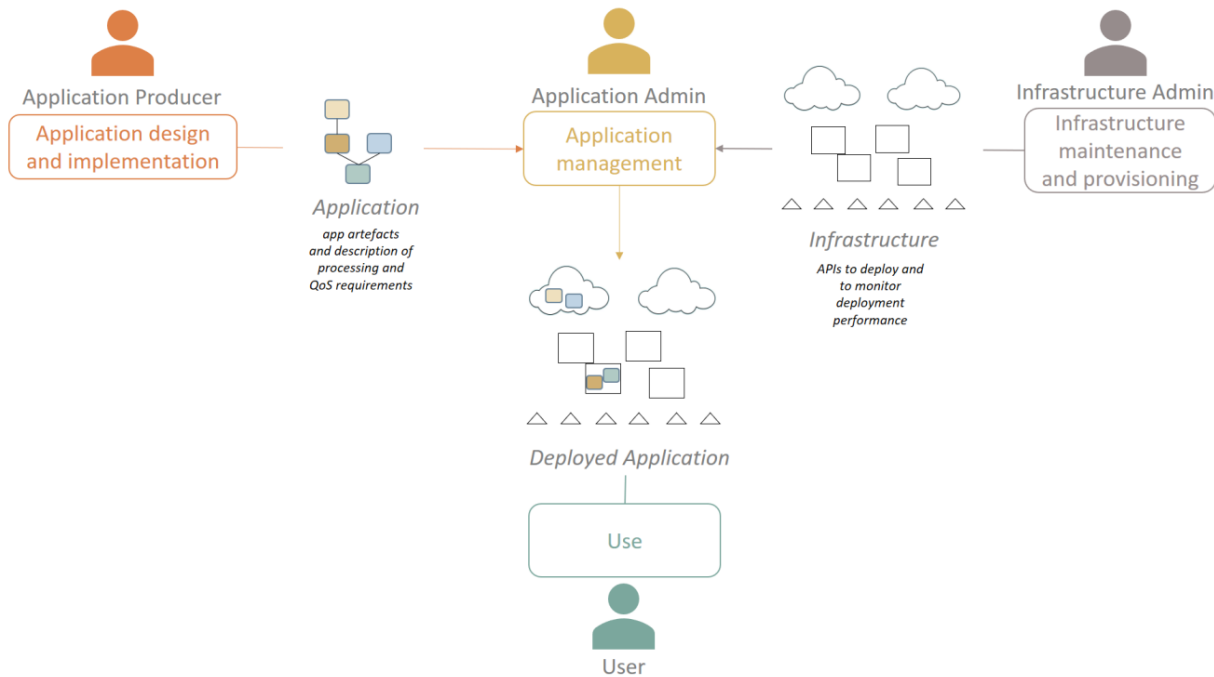
Context-awareness

- Discover and use available resources, **cooperating** horizontally

Fog application management



Fog application management



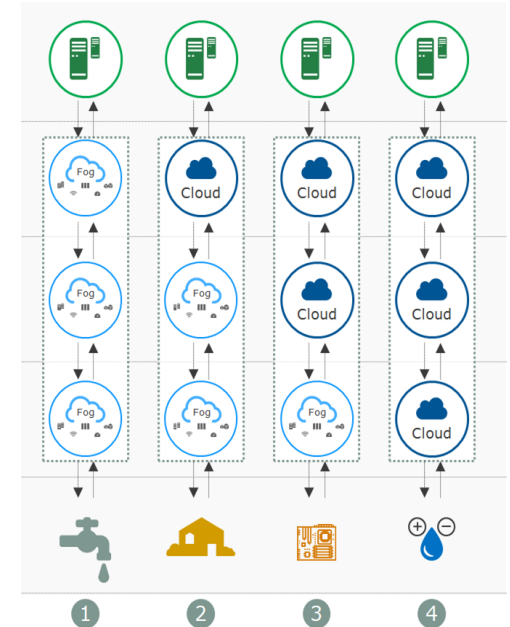
One stakeholder can play more than one role

- DevOps: roles of application producer and administrator collapse
- Telecom provider providing IoT+Edge+Cloud infrastructure to its customers, and managing apps and services over the very same infrastructure
- User can act as infrastructure provider when sharing her home router as Edge capability for application deployment

Fog application management



- How to decide *where* to «best» deploy each component of an application by exploiting QoS-, location-, and context-awareness?
- How to estimate the «goodness» of a candidate deployment?



fogtorch 



Service-based computing

Cloud-based computing

Fog computing



Next seminars

May 12 Introduction to service, cloud and fog computing

May 19 Fault-aware management of cloud-based applications



May 26 Microservices and Docker



June 01 Combining TOSCA and Docker



June 09 Predicting quality of service of software applications



June 16 App deployment through the fog



XX YY TBA

