Engineering Cloud Applications 1

Cloud Computing

Dr. Philipp Leitner

University of Zurich, Switzerland

Universität Zürich[UZH]

What is Cloud Computing? (1)

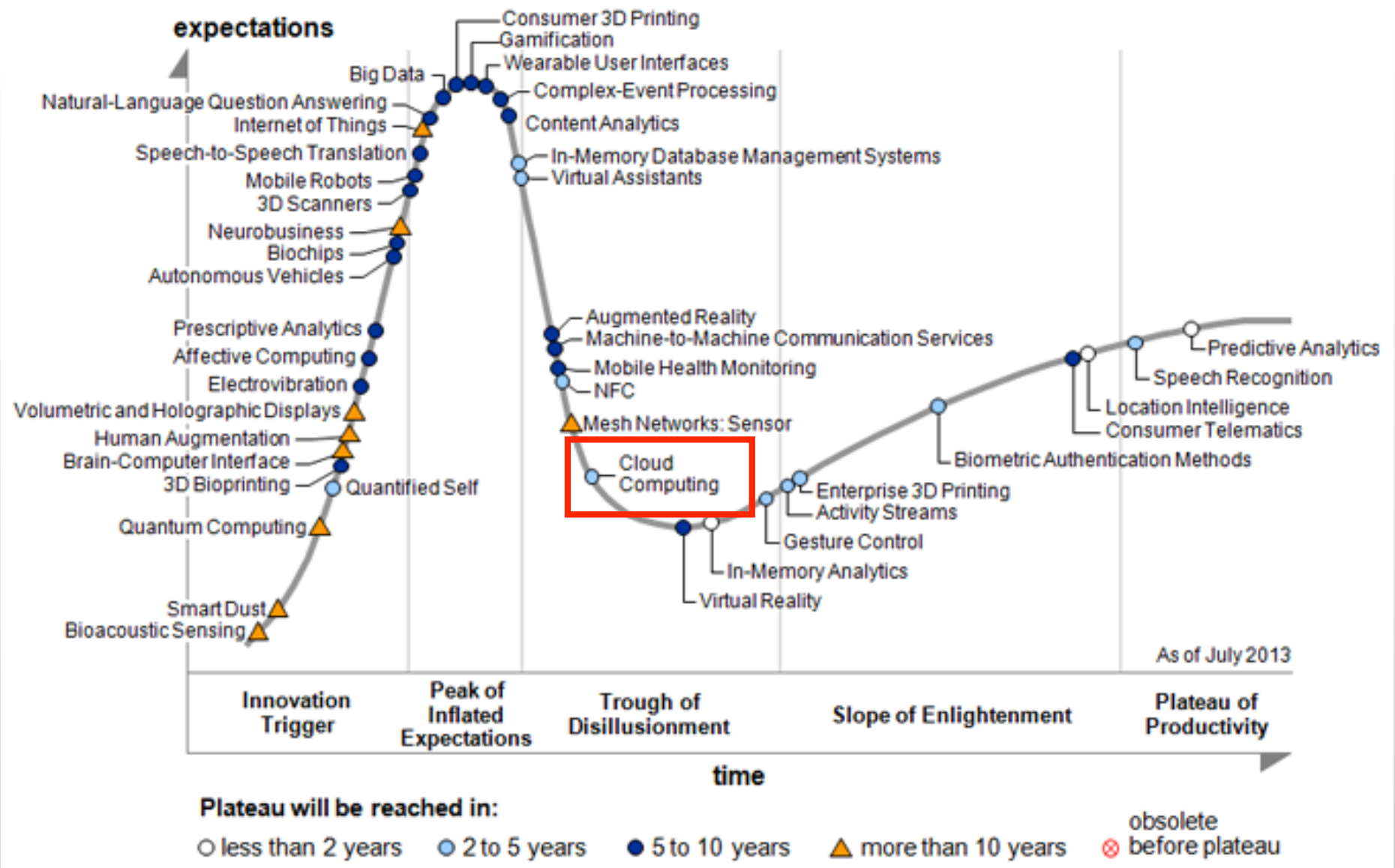
- From a user perspective:
  - Software and data is access over the Internet

What is Cloud Computing? (2)

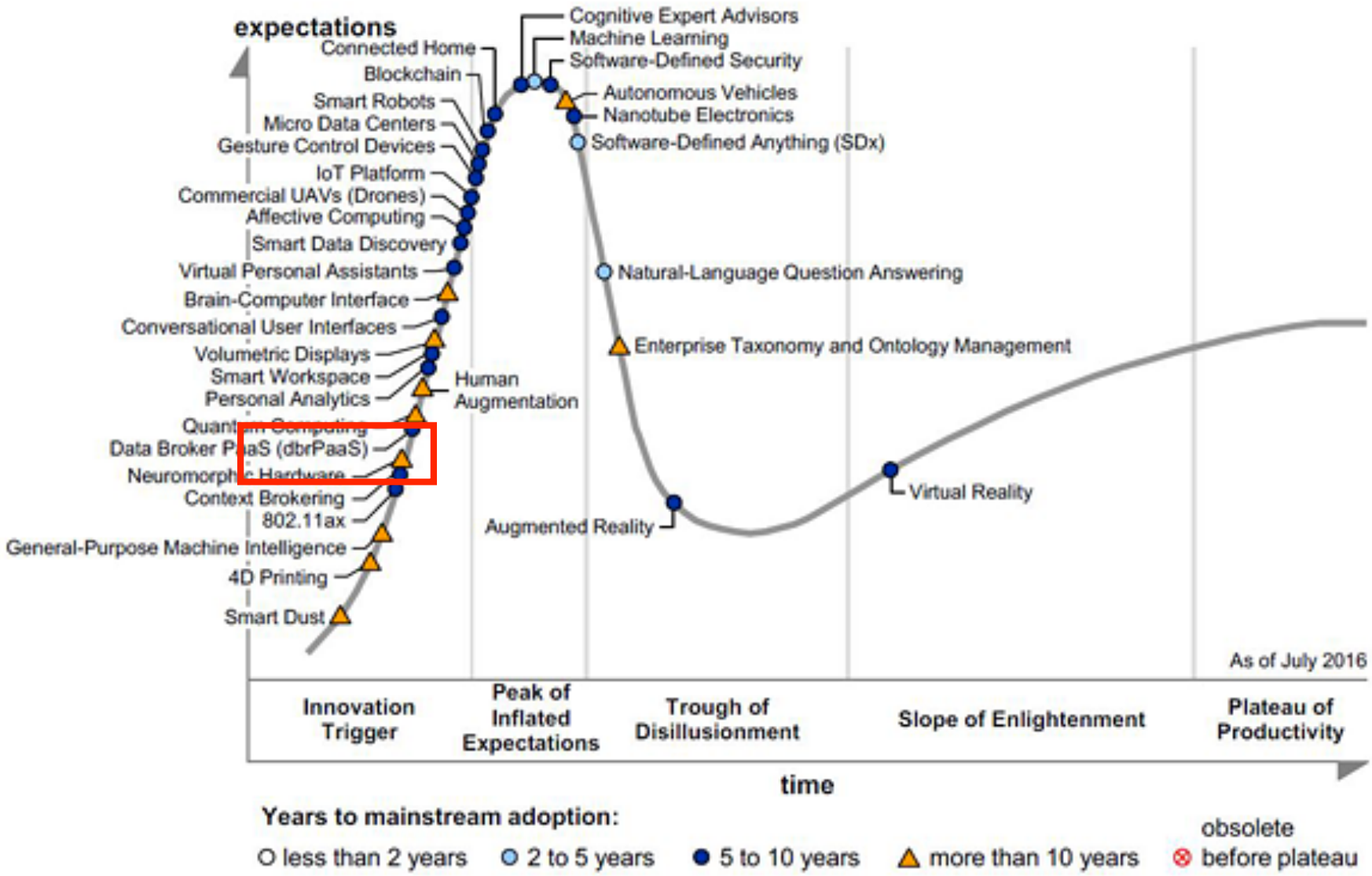
- From a user perspective:
  - Software and data is access over the Internet
- From a software developer perspective:
  - Developer “rents” computing resources
  - Application is running on the cloud provider’s machines
  - Fundamentally about **loss of control**

Gartner Hype Cycle 2011

Gartner Hype Cycle 2013



Gartner Hype Cycle 2016



Why is CC so popular?

- Let's assume you are a small(ish) company

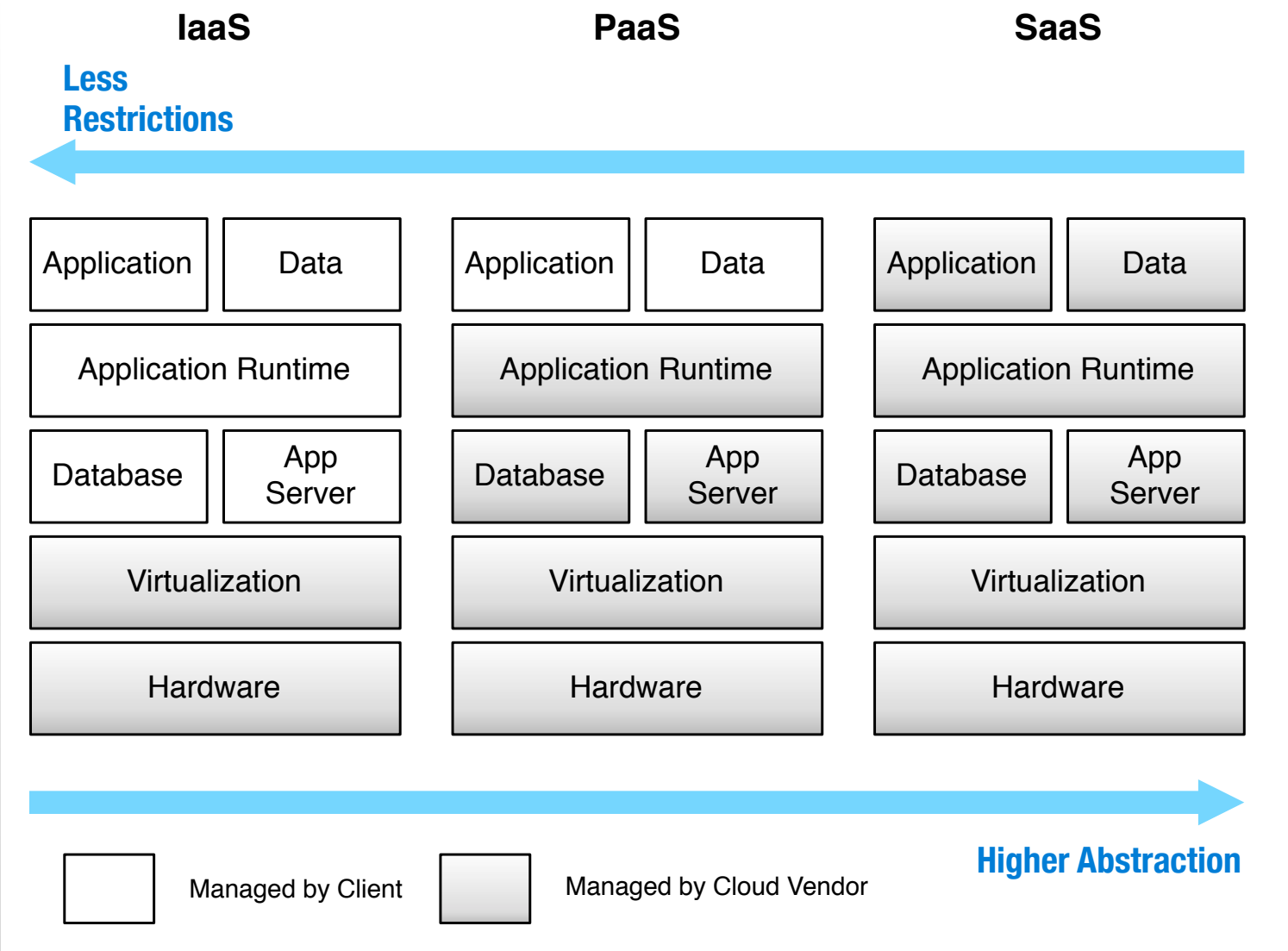
CODESHIP

- You probably have a Web-based product
- You probably don't have too much money

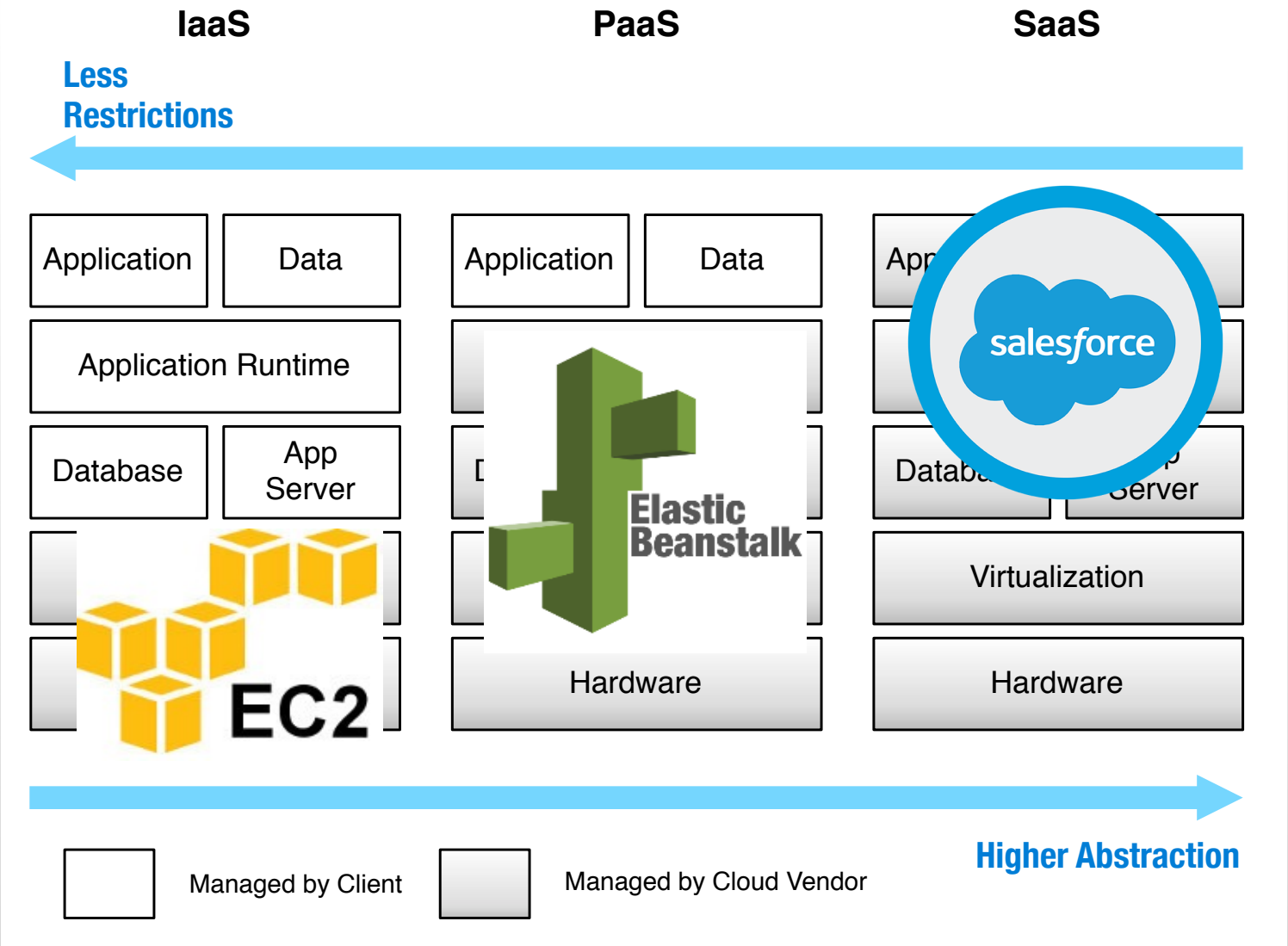
"In the past, if your application became popular and your systems or your infrastructure did not scale you became a victim of your own success. Conversely, if you invested heavily and did not get popular, you became a victim of your failure."

Jinesh Varia, Amazon

Cloud Technology Stack



Example Services



Some Example Cloud Services

AWS EC2 (IaaS)

AWS | Services | Edit | Philipp Leitner | Oregon | Support

EC2 Dashboard
Events
Tags
Reports
Limits

INSTANCES
Instances
Spot Requests
Reserved Instances
Scheduled Instances
Commands
Dedicated Hosts

IMAGES
AMIs
Bundle Tasks

ELASTIC BLOCK STORE
Volumes
Snapshots

NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

LOAD BALANCING
Load Balancers

AUTO SCALING
Launch Configurations
Auto Scaling Groups

Launch Instance | Connect | Actions

Filter by tags and attributes or search by keyword — 1 to 4 of 4

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitc
viepepProxy	i-0d799f01	t2.micro	us-west-2b	stopped		None			ViepepBackend	disable
	i-b1cdac74	r3.8xlarge	us-west-2a	stopped		None			calexandru	dis
Default-Envir...	i-bf447578	t1.micro	us-west-2a	terminated		None				disable
ilia_test	i-fd1658f1	t1.micro	us-west-2b	stopped		None			openworld	disable

Instance: i-b1cdac74 Public DNS: -

Description | Status Checks | Monitoring | Tags

Instance ID	i-b1cdac74	Public DNS	-
Instance state	stopped	Public IP	
Instance type	r3.8xlarge	Elastic IP	-
Private DNS	-	Availability zone	us-west-2a
Private IPs		Security groups	launch-wizard-2. view rules
Secondary private IPs	-	Scheduled events	-
VPC ID	-	AMI ID	ubuntu-trusty-14.04-amd64-server-20150325 (ami-5189a661)
Subnet ID	-	Platform	-
Network interfaces	-	IAM role	-
Source/dest. check	False	Key pair name	calexandru
ClassicLink	Unlinked	Owner	284430493654
EBS-optimized	False	Launch time	August 24, 2015 at 1:32:49 PM UTC+2 (5302 hours)

AWS Elastic Beanstalk (Paas)

Elastic Beanstalk | Create New Environment

Info

Elastic Beanstalk is now creating your environment. When it has finished it will be running Sample Application.

My First Elastic Beanstalk Application ▸ Default-Environment (**Environment ID:** e-srih5ikspy, **URL:**) | Actions

- Dashboard
- Configuration
- Logs
- Health NEW
- Monitoring
- Alarms
- Events
- Tags

Elastic Beanstalk is launching your environment.
View Events

Overview | Refresh

Health
Pending
Causes

Running Version
Upload and Deploy

Configuration
64bit Amazon Linux 2015.09
v2.0.8 running Ruby 2.2 (Puma)
Change

Recent Events | Show All

Time	Type	Details
2016-04-01 11:35:10 UTC+0200	INFO	Using elasticbeanstalk-us-west-2-284430493654 as Amazon S3 storage bucket for environment data.
2016-04-01 11:35:09 UTC+0200	INFO	createEnvironment is starting.

AWS Lambda (PaaS as well)

AWS | Services | Edit | Philipp Leitner | Oregon | Support

Lambda > Functions > slackfunction

ARN - arn:aws:lambda:us-west-2:284430493654:function:slackfunction

Test | Actions

Code | Configuration | Event sources | API endpoints | Monitoring

Code entry type: Edit code inline | Upload a .ZIP file | Upload a .ZIP from Amazon S3

```
61 from base64 import b64decode
62 from urlparse import parse_qs
63 import logging
64
65 ENCRYPTED_EXPECTED_TOKEN = "CiAic8RDrptZy1oL9OfOThqVjJValprlU2KREse6GbQk0xKfAQEBAgB4InPEQ66bWctaC/Tnzk4alYyVWpaa5VNikRLHuhm0JNMAAAB2MHQGCSqGSIb3DQEHBqBnMGUCAQAwYAYJKoZIhvcNA(
66
67 # kms = boto3.client('kms')
68 # expected_token = kms.decrypt(CiphertextBlob = b64decode(ENCRYPTED_EXPECTED_TOKEN))['Plaintext']
69
70 logger = logging.getLogger()
71 logger.setLevel(logging.INFO)
72
73 def lambda_handler(event, context):
74
75     req_body = event['body']
76     params = parse_qs(req_body)
77     token = params['token'][0]
78     #if token != expected_token:
79     #    logger.error("Request token (%s) does not match exptected", token)
80     #    raise Exception("Invalid request token")
81
82     user = params['user_name'][0]
83     command = params['command'][0]
84     channel = params['channel_name'][0]
85     command_text = params['text'][0]
86
87     return "%s invoked %s in %s with the following text: %s" % (user, command, channel, command_text)
```

Other Cloud Providers

Basic Cloud Computing Concepts

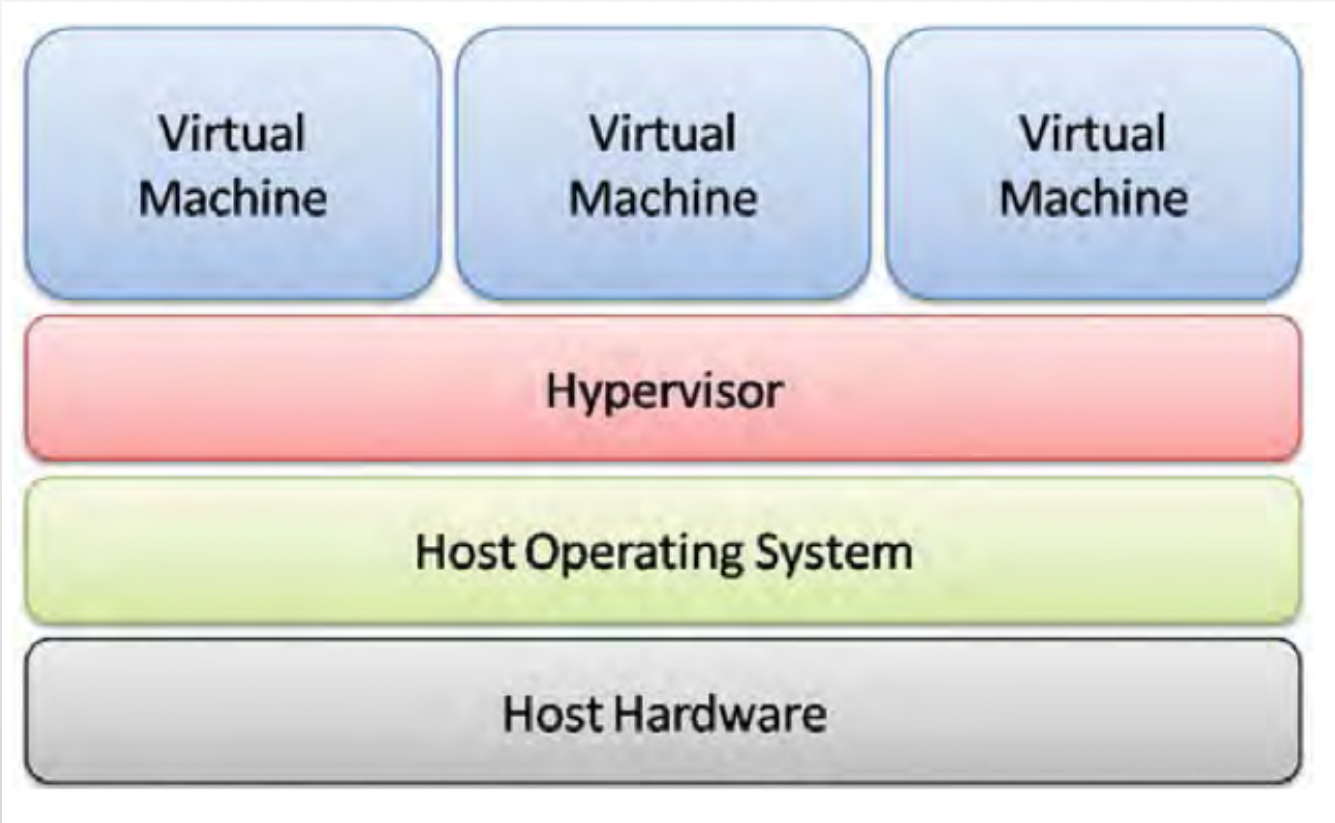
- **Core concepts:**
  - Resource virtualization / pay-per-use
  - Elasticity
  - Multi-Tenancy
  - Opaque infrastructure
  - “Illusion of Infinite Resources”

Basic Cloud Computing Concepts

- **Core concepts:**
  - **Resource virtualization / pay-per-use**
  - Elasticity
  - Multi-Tenancy
  - Opaque infrastructure
  - “Illusion of Infinite Resources”

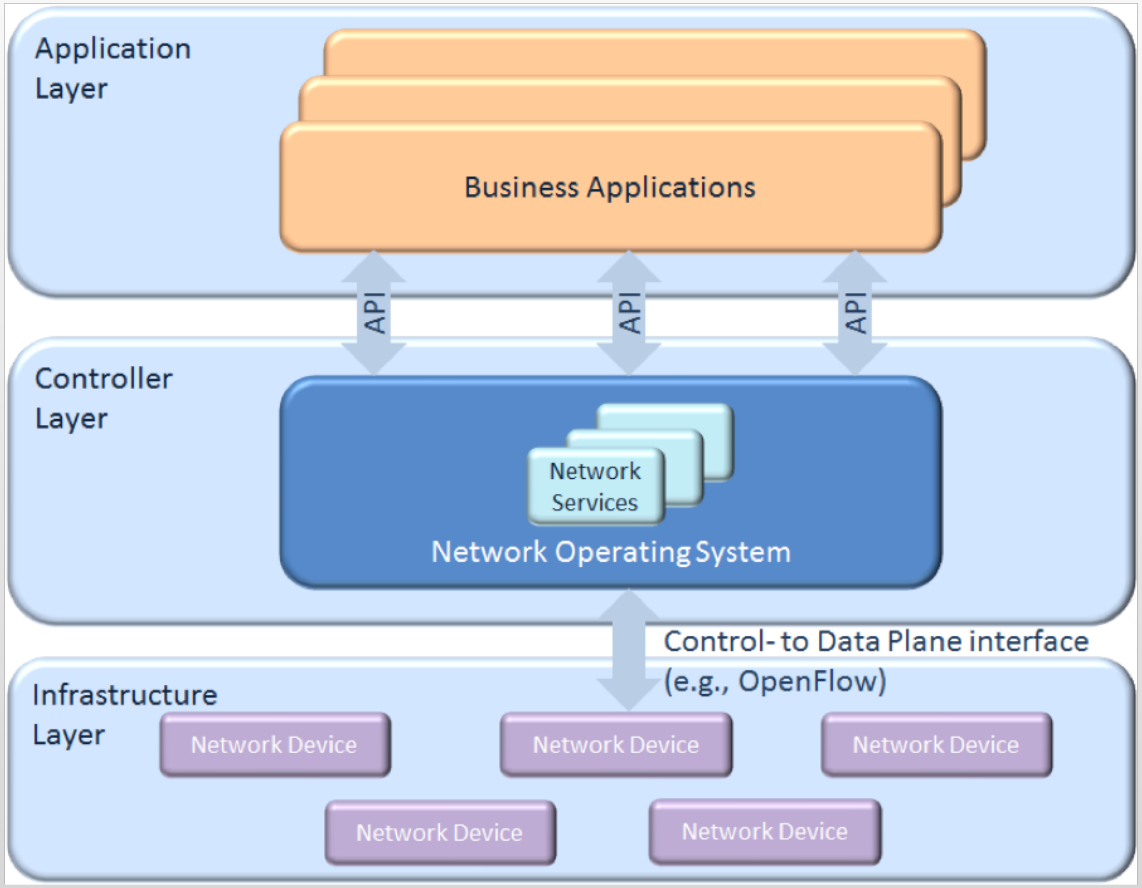
Virtualization of Computing Resources

Virtualization is at the core of cloud computing



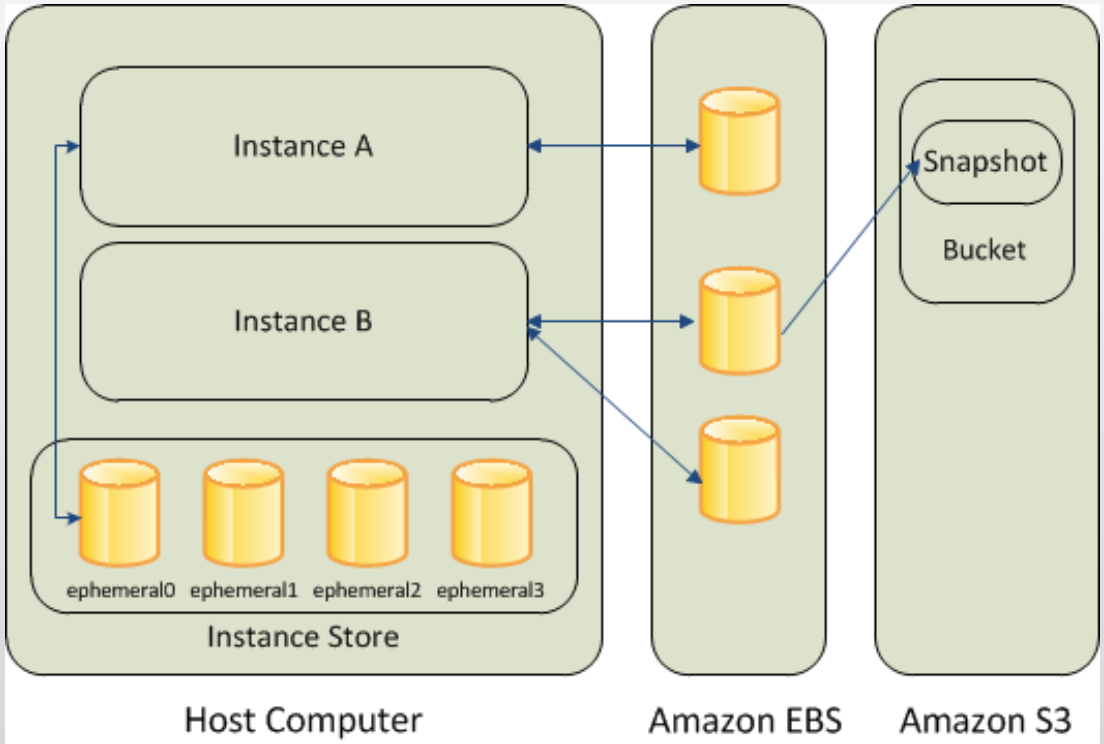
Network Virtualization

- **Software-Defined Networks (SDN)**



Virtualization of Storage

- **"Ephemeral" Storage**
  - Storage on the same physical host
- **Block Storage**
  - Virtualized NAS



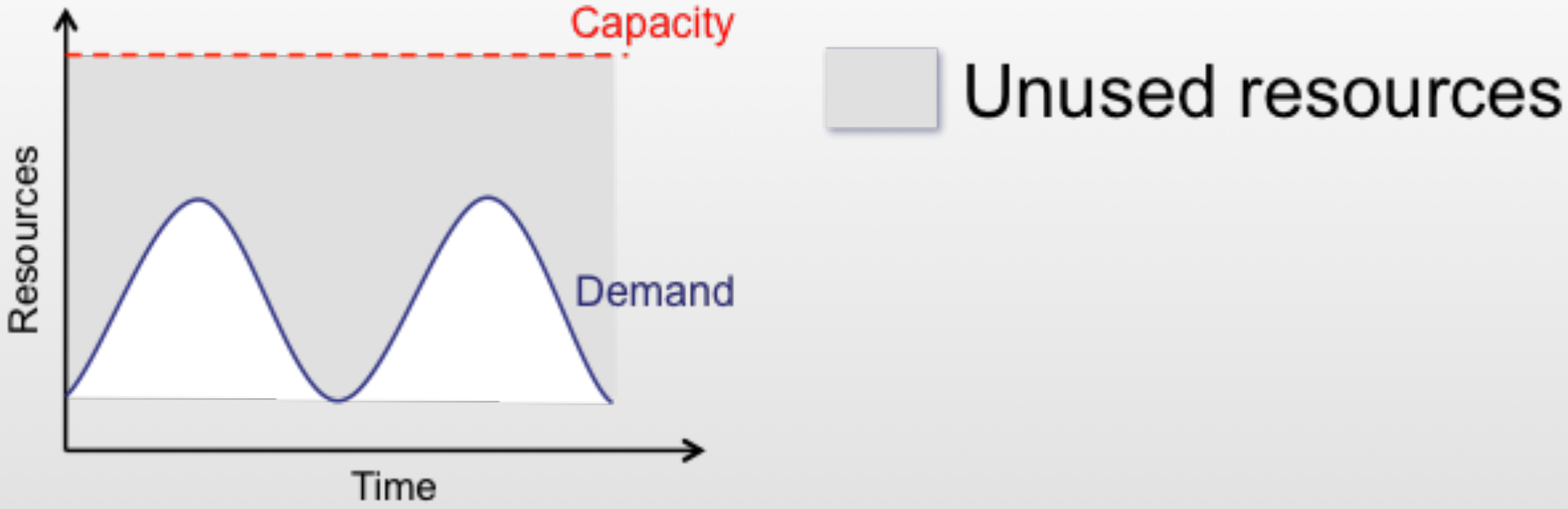
Basic Cloud Computing Concepts

- **Core concepts:**
  - Resource virtualization / pay-per-use
  - **Elasticity**
  - Multi-Tenancy
  - Opaque infrastructure
  - “Illusion of Infinite Resources”

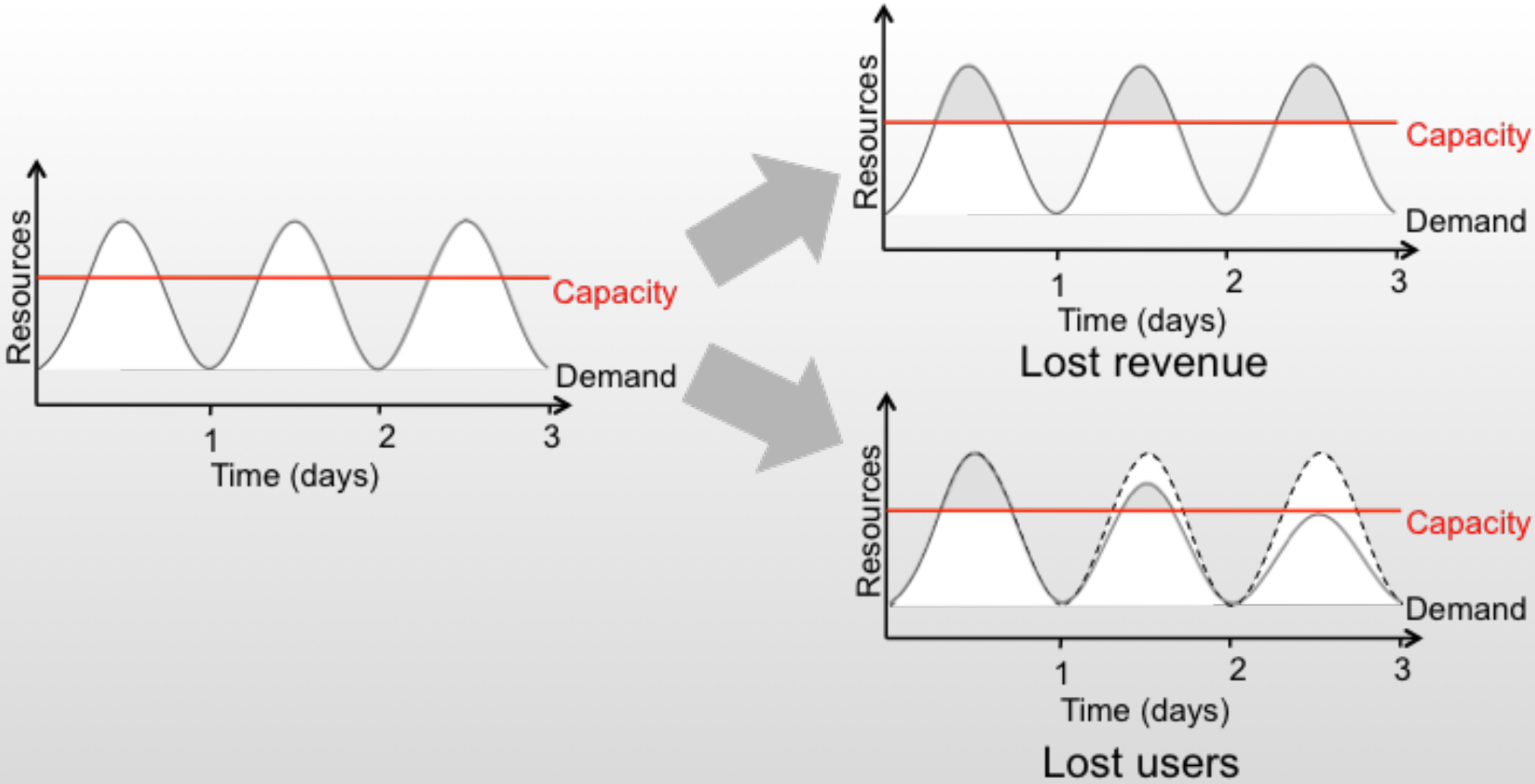
On Elasticity

"elasticity (or stretchiness) is the physical property of a material that **returns to its original shape** after the stress (e.g. external forces) that made it deform or distort is removed" – http://en.wikipedia.org/wiki/Elasticity_(physics)

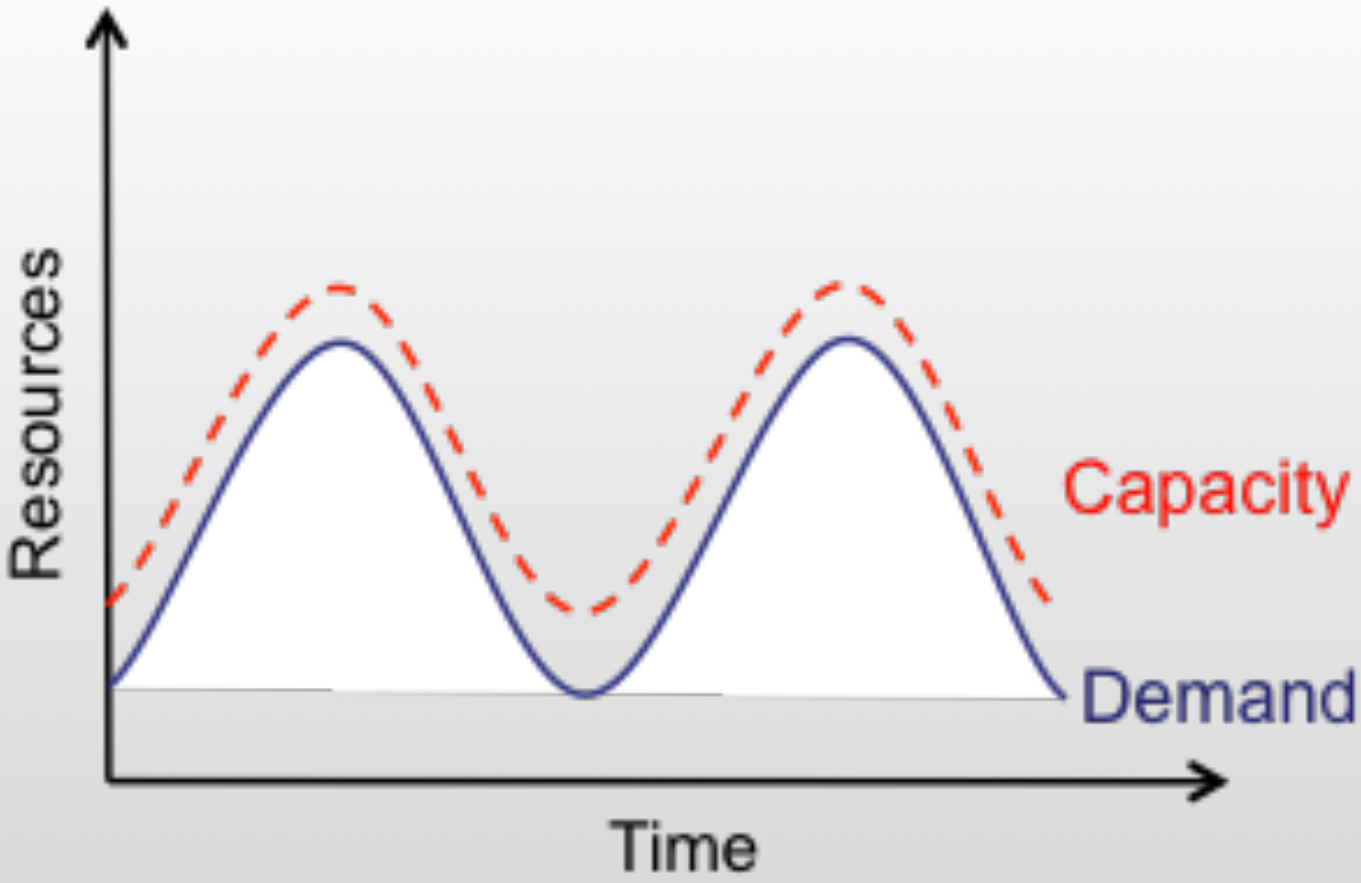
Overprovisioning



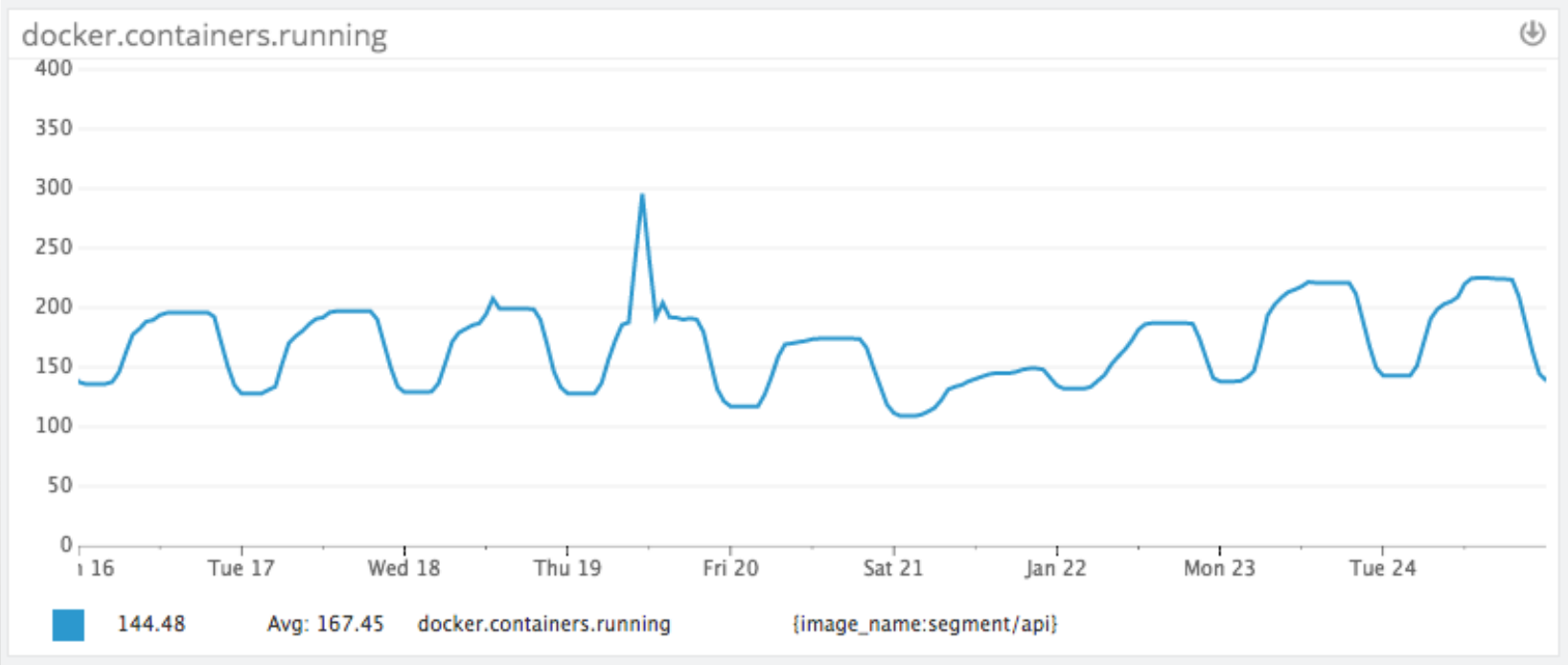
Underprovisioning



Elastic Provisioning



Real Example (from Segment)



https://segment.com/blog/the-million-dollar-eng-problem/

Pay-Per-Use Granularity

- In practice, how “elastic” one can really be depends on three factors:
  - Resource granularity (how fine-grained can we get resources?)
  - Billing granularity (“billing time unit”, for instance 1 hour)
  - Provisioning time (how long does it take to get and release resources?)
- **—> all of those depend on the service !**

Basic Cloud Computing Concepts

- **Core concepts:**
  - Resource virtualization / pay-per-use
  - Elasticity
  - **Multi-Tenancy**
  - Opaque infrastructure
  - “Illusion of Infinite Resources”

Multi-Tenancy

- Resources are shared among many independent parties
  - *e.g., the physical host that runs your application also hosts VMs for Netflix*
- Fundamentally **how the cloud makes money**
  - Allows cloud to efficiently use its physical infrastructure and leverage **economies of scale**

Main Issues

- **“Noisy Neighbours”** (resource inference)
  - Not all computational resources can be cleanly divided
  - Most importantly: **I/O and network access**
- **“Nosy Neighbours”** (information leakage)
  - E.g., VM placement attacks

Potential Fixes

- "**Dedicated**" instances are guaranteed to be only co-located with resources of the same tenant
  - Not the same as non-virtualized resources
- **Bare Metal** Cloud
  - Non-virtualized resources that are served on demand

Basic Models

IaaS vs. PaaS vs. SaaS

What level of abstraction is provided by the cloud?

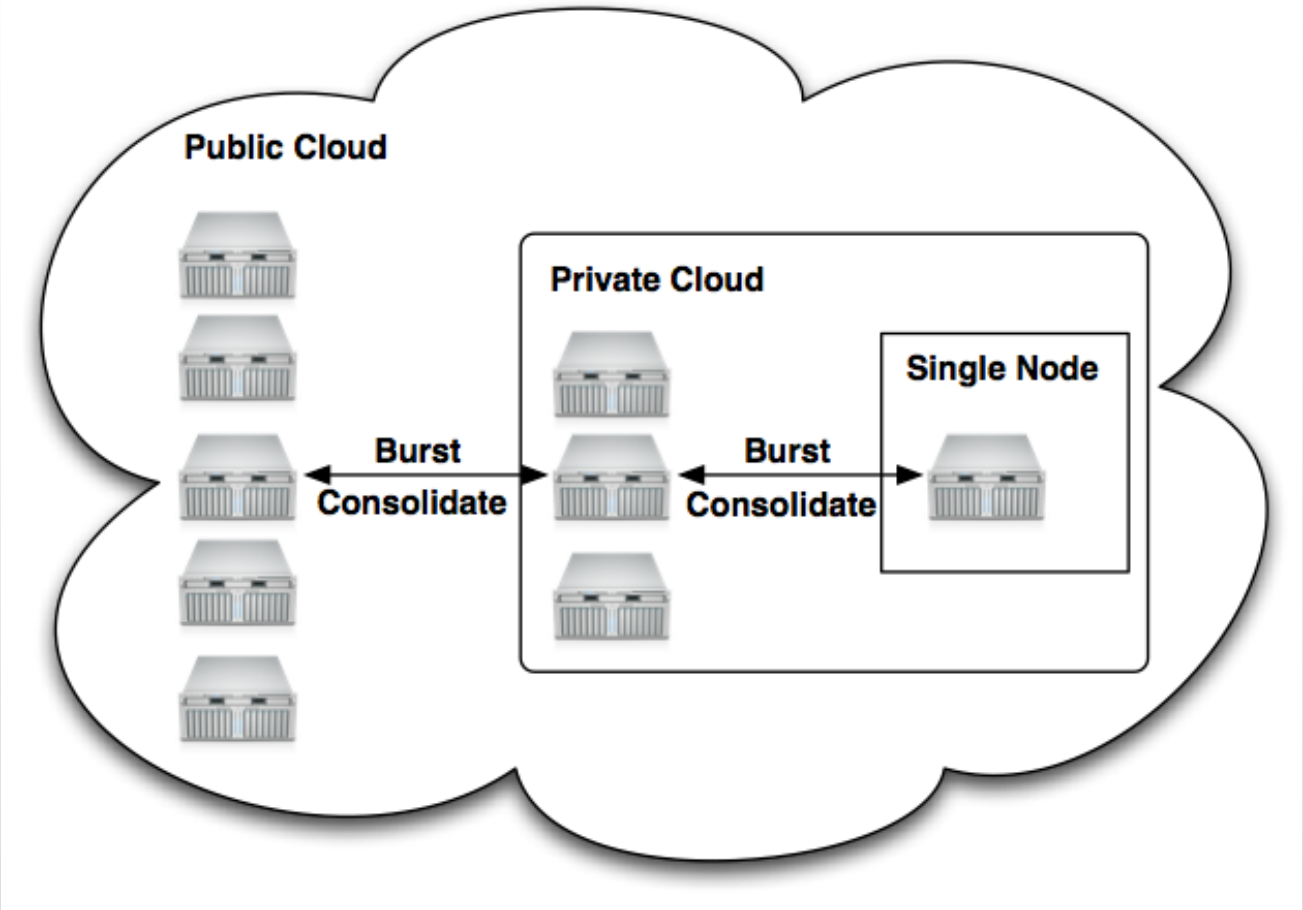
Public vs. Private vs. Hybrid Cloud

Who is hosting the cloud?

Private Cloud

- So far we have always assumed that “cloud computing” means **public cloud**
  - The entity providing the cloud is separate from the consumer
- **Private cloud**:
  - Use cloud principles on your own hardware
  - Largely out of scope in this lecture

Cloud Bursting



Hybrid Cloud

A hybrid cloud system is a system that makes use of multiple clouds at the same time

(typically a combination of private and public clouds)

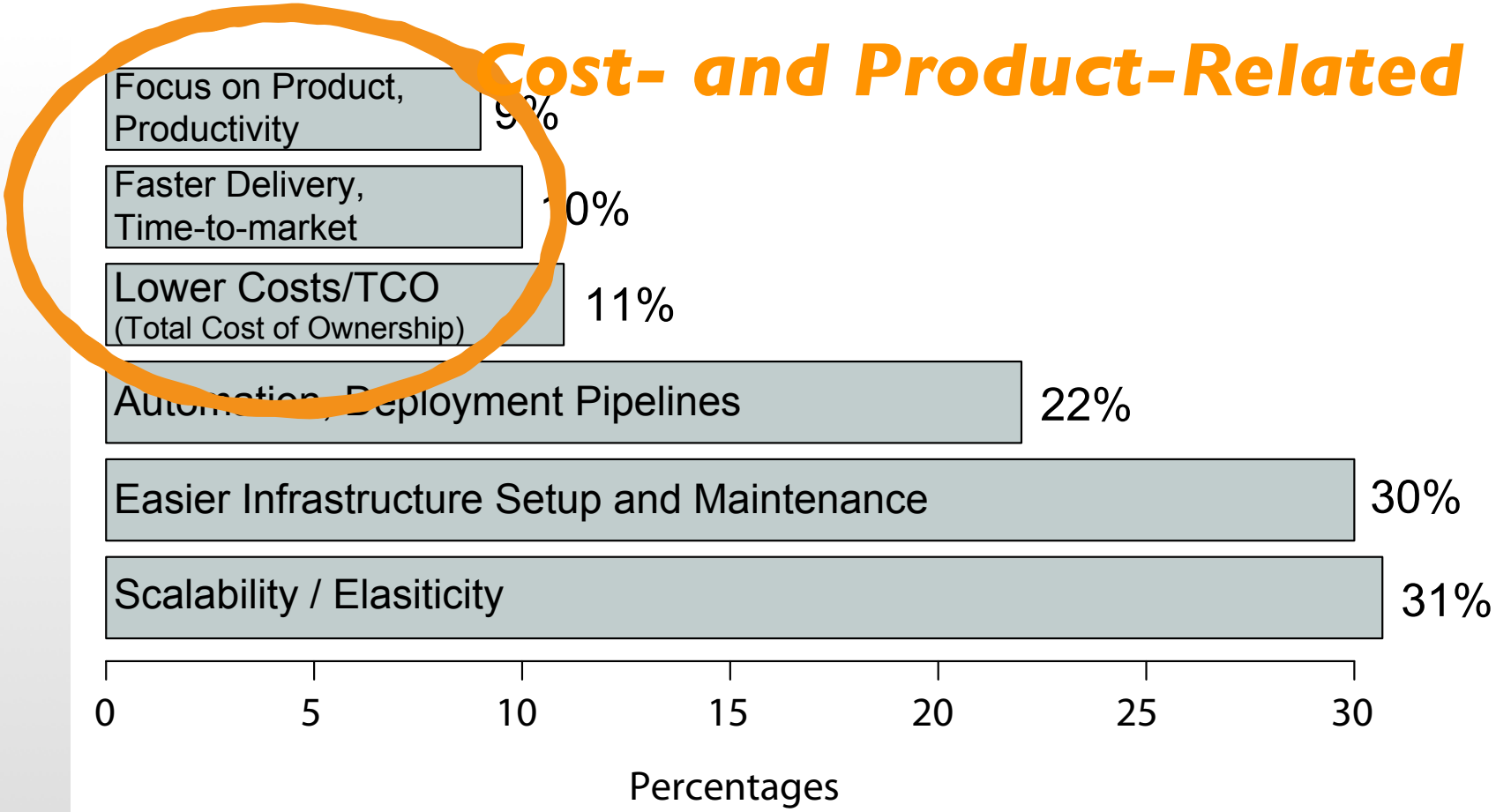
Cloud Bursting Advantages

- Best of two worlds:
  - (1) application mostly runs on top of (cheap, fast and relatively reliable) internal data centre —> private cloud
  - (2) load spikes can be covered by resources from a public cloud provider —> public cloud

Cloud Bursting Disadvantages

- **Technical issues:**
  - Network latency between instances in your private cloud and those in the public cloud is very large
  - Technical capabilities of the different clouds may be different
- **Legal issues:**
  - Oftentimes, not all data is allowed to be transferred to a public cloud

Opportunities / Benefits



(survey responses of 294 cloud developers)

J. Cito, P. Leitner, T. Fritz, and H. C. Gall, "**The Making of Cloud Applications – An Empirical Study on Software Development for the Cloud**," in Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE), 2015.

Business Opportunities

- **Lower TCO (presumably)**
- **CapEx —> OpEx**
  - Pay little as long as you don’t have users
- **DevOps / NoOps**
  - Less administration, more building features
- **Business Agility**
  - Faster feedback, continuous delivery

TCO

- **“Total Costs of Ownership”**
  - **Acquistion + Operation + Change**

  - Costs of buying the hardware
  - + Operational costs (utilities, admin salaries)
  - + Maintenance and repair

Simple Calculation Example

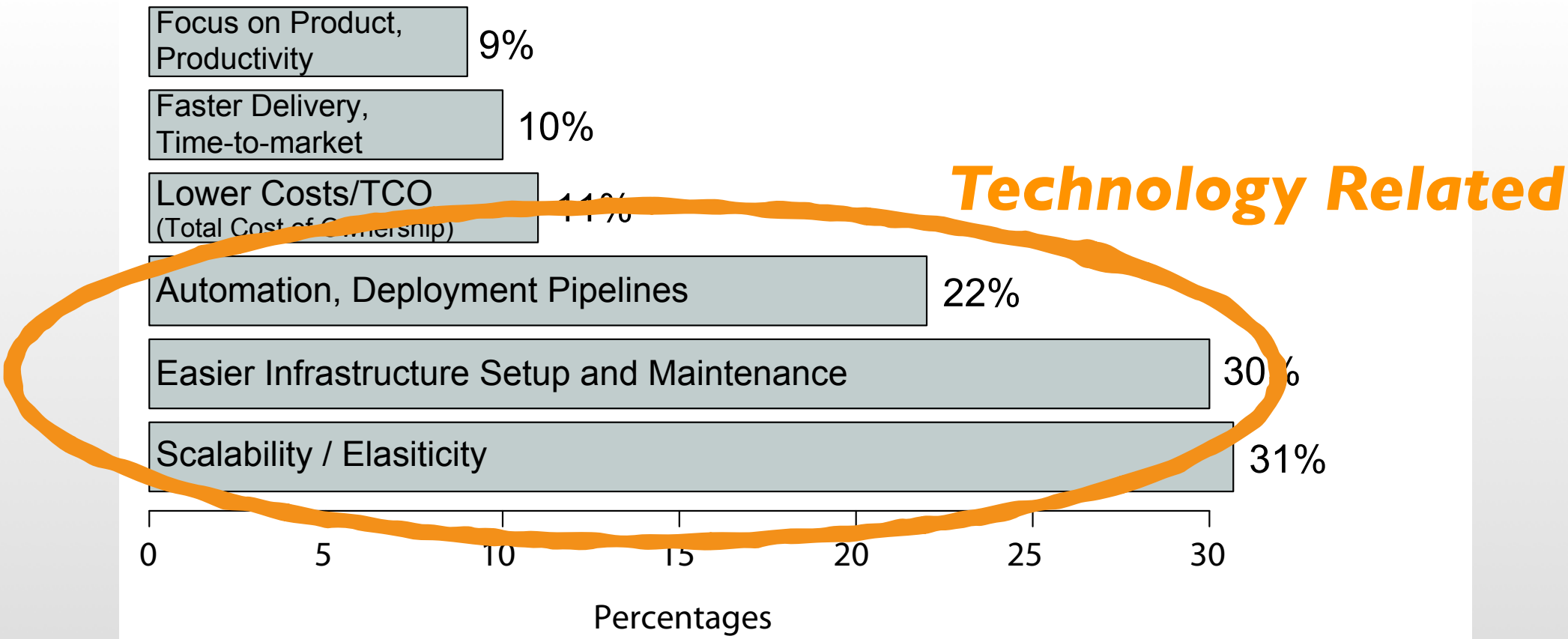
$ in 1,000s	Acquisition Costs	Operating Costs	Change Costs	Total	% of TCO
Software	274	82	138	**494**	2.9%
Hardware	539	97	71	**707**	4.1%
Personnel	55	8,873	5,952	**14,879**	86.2%
NW & Comm	146	543	459	**1,149**	6.7%
Facilities	0	15	15	**29**	0.2%
Total	**1,104**	**9,610**	**6,634**	**17,258**	—
% of TCO	5.9%	55.7%	38.4%	—	100.0%

https://www.business-case-analysis.com/total-cost-of-ownership.html

CapEx -> OpEx

- Acquisition costs -> **capital expenses** (CapEx)
  - Requires money upfront
  - Cashflow problem (especially for startups)
- Through pay-per-use, you essentially have no CapEx anymore, only operational expenses (OpEx)
- Compare: car or personell leasing

Opportunities / Benefits



(survey responses of 294 cloud developers)

J. Cito, P. Leitner, T. Fritz, and H. C. Gall, "**The Making of Cloud Applications – An Empirical Study on Software Development for the Cloud**," in Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE), 2015.

Automation

- Central concept in IaaS:
  - Instances are **volatile**
  - No data stored on instance
  - Can be terminated any time
  - Setup completely automated

> *"One interesting thing that is very cloud specific and influenced our architecture is, that the cloud provider tells you, we can kill your machine any time we want."* -$P9_{IaaS}$

> *"We have now moved to strict Immutable Infrastructure in our deployment. We don't even put SSH keys into instances anymore, making changes to existing infrastructure impossible"* -$P12_{IaaS}$

J. Cito, P. Leitner, T. Fritz, and H. C. Gall, "**The Making of Cloud Applications – An Empirical Study on Software Development for the Cloud**," in Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE), 2015.

Example Technology - Opscode Chef

```
$ knife ec2 server create \
  --availability-zone us-east-1d \
  --node-name redis.learnchef.demo \
  --flavor t1.micro \
  --image ami-fd20ad94 \
  --identity-file ~/.ssh/aws.pem \
  --run-list "role[redis]" \
  --ssh-user ubuntu
```

Interacts with the IaaS cloud via API

```
name 'redis'
description 'A redis master on slave (on a single machine)'
default_attributes({
  'redisio' => {
    'servers' => [
      { 'port' => '6379' },
      { 'port' => '6380', 'slaveof' => { 'address' => '127.0.0.1', 'port' => '6379' } }
    ]
  }
})
run_list(
  'recipe[redisio::install]',
  'recipe[redisio::enable]'
)
```

Configures the server via Ruby code

Summary (1)

- (Public) cloud computing is about doing your computing on somebody else's hardware
  - —> **Loss of control**
- Basic models:
  - IaaS vs. PaaS vs. SaaS
  - Private vs. Public Cloud

Summary (2)

- Core advantages are **technical** and **economical**
  - **Economical advantages:**
    - Reduced TCO
    - No investment necessary
  - **Technical advantages:**
    - Elasticity
    - Automation