

Architectural styles for cloud computing



Prof. Paolo Ciancarini
Software Architecture
CdL M Informatica □
Università di Bologna

Agenda

- Introduction to Cloud Computing
- Cloud computing architectural styles
- Computing as a service: the MapReduce style

What is Cloud Computing?

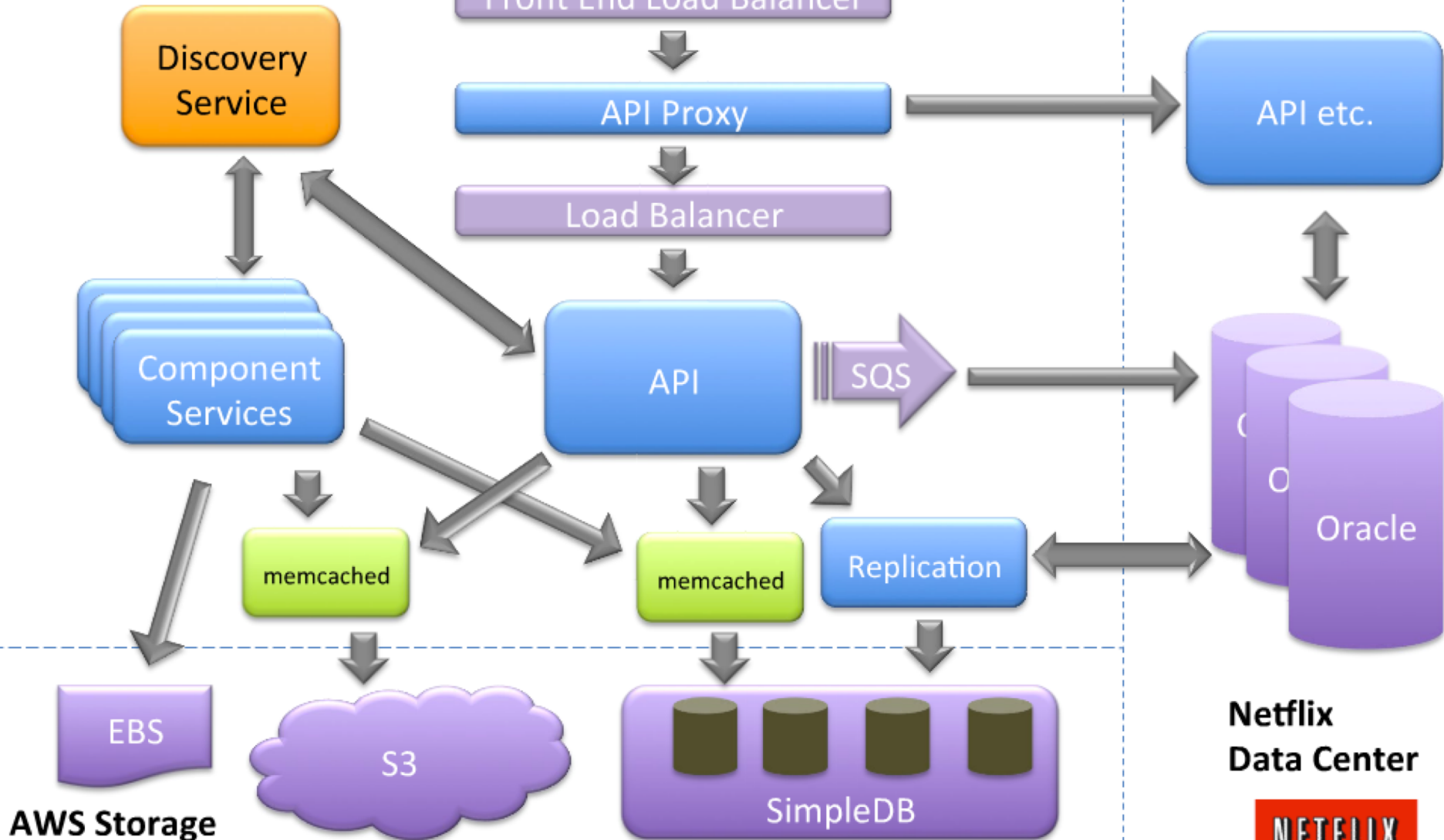
- **Cloud Computing** is a new kind of **computing commodity** over the Internet
 - *Cloud platform*: a collection of integrated hardware, software, and network infrastructure
 - Offering to clients virtual hardware, software or networking services
- The platforms (*clouds*) hide the complexity and the details of the underlying physical infrastructure by providing a simple graphical interface and/or some API (Applications Programming Interface)

API



Example: Netflix

AWS EC2



<http://www.programmableweb.com/api/netflix>



Amazon Elastic Cloud Computing EC2

- Provider of cloud infrastructure
- Data centers worldwide
- Many types of Virtual Machines
 - e.g. Linux, Windows, FreeBSD Unix
- Free tier for one year
- Per-hour charging (NB: complex pricing model)
 - Small 0.044\$/h
 - Large 0.14\$/h
 - Xlarge 0.28\$/h

<http://aws.amazon.com/ec2/pricing/>

AWS free tier

AWS Free Tier Details

★ FEATURED 12 MONTHS FREE ALWAYS FREE PRODUCT CATEGORIES ALL

<p>12 months free and always free products</p> <hr/> <p>AWS Free Tier includes offers that expire 12 months following sign up and others that never expire.</p> <p>Learn more »</p>	<p>COMPUTE Amazon EC2</p> <p>750 Hours per month</p> <hr/> <p>Resizable compute capacity in the Cloud</p> <p>Learn more about Amazon EC2 »</p> <hr/> <p>EXPAND DETAILS ^</p>	<p>ANALYTICS Amazon QuickSight</p> <p>1 GB of SPICE capacity</p> <hr/> <p>Fast, easy-to-use, cloud-powered business analytics service at 1/10th the cost of traditional BI solutions</p> <p>Learn more about Amazon QuickSight »</p> <hr/> <p>EXPAND DETAILS ^</p>
<p>DATABASE Amazon RDS</p> <p>750 Hours per month of db.t2.micro database usage (applicable DB engines)</p> <hr/> <p>Managed Relational Database Service for MySQL, PostgreSQL, MariaDB, Oracle BYOL, or SQL Server</p> <p>Learn more about Amazon RDS »</p> <hr/> <p>EXPAND DETAILS ^</p>	<p>STORAGE & CONTENT DELIVERY Amazon S3</p> <p>5 GB of standard storage</p> <hr/> <p>Secure, durable, and scalable object storage infrastructure</p> <p>Learn more about Amazon S3 »</p> <hr/> <p>EXPAND DETAILS ^</p>	<p>COMPUTE AWS Lambda</p> <p>1 Million free requests per month</p> <hr/> <p>Compute service that runs your code in response to events and automatically manages the compute resources</p> <p>Learn more about AWS Lambda »</p> <hr/> <p>EXPAND DETAILS ^ 6</p>

A cloud reference stack

Cloud Clients

Presentation Layer

Example: browsers, mobile devices

Cloud Applications

Software as a Service

Example: Google docs or calendar

Cloud Services

Components as a Service

Example: SOA via Web Service standards

Cloud Platform

Platform as a Service

Example: web server, app server

Cloud Storage

Storage as a Service

Note: formerly utility computing

Cloud Infrastructure

Distributed Multi-site Physical Infrastructure

Note: enabled by server virtualization

Cloud

- A cloud platform provides **services** that are always on, and on demand can be exploited anywhere, anytime
- Pay for use and elastic (= as needed)
 - scale up and down in capacity and functionalities
- The services are available to generic users, enterprises, corporations, or businesses markets
 - Thus, Cloud Computing can be considered a step on from Service Oriented Computing

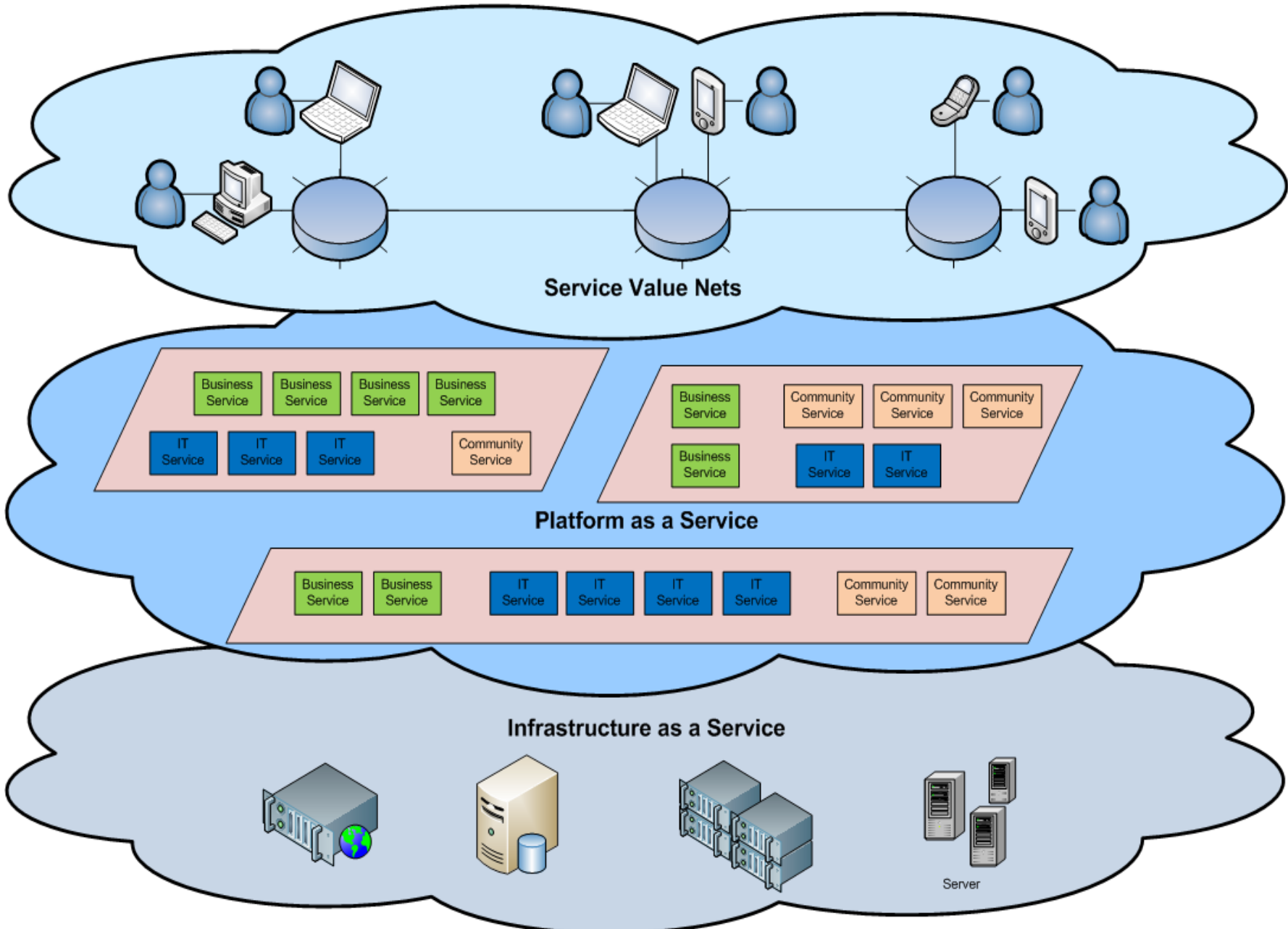
Cloud Computing Service Layers

	Services	Description
Application Focused	Services	Complete business services such as PayPal, OpenID, OAuth, Google Maps, Alexa
	Application	Cloud based software that eliminates the need for local installation such as Google Apps, Microsoft Online
	Development	Software development platforms used to build custom cloud based applications (PAAS & SAAS) such as Salesforce
Infrastructure Focused	Platform	Cloud based platforms, typically provided using virtualization, such as Amazon ECC, Sun Grid
	Storage	Data storage or cloud based NAS such as CTERA, iDisk, CloudNAS
	Hosting	Physical data centers such as those run by IBM, HP, NaviSite, etc.

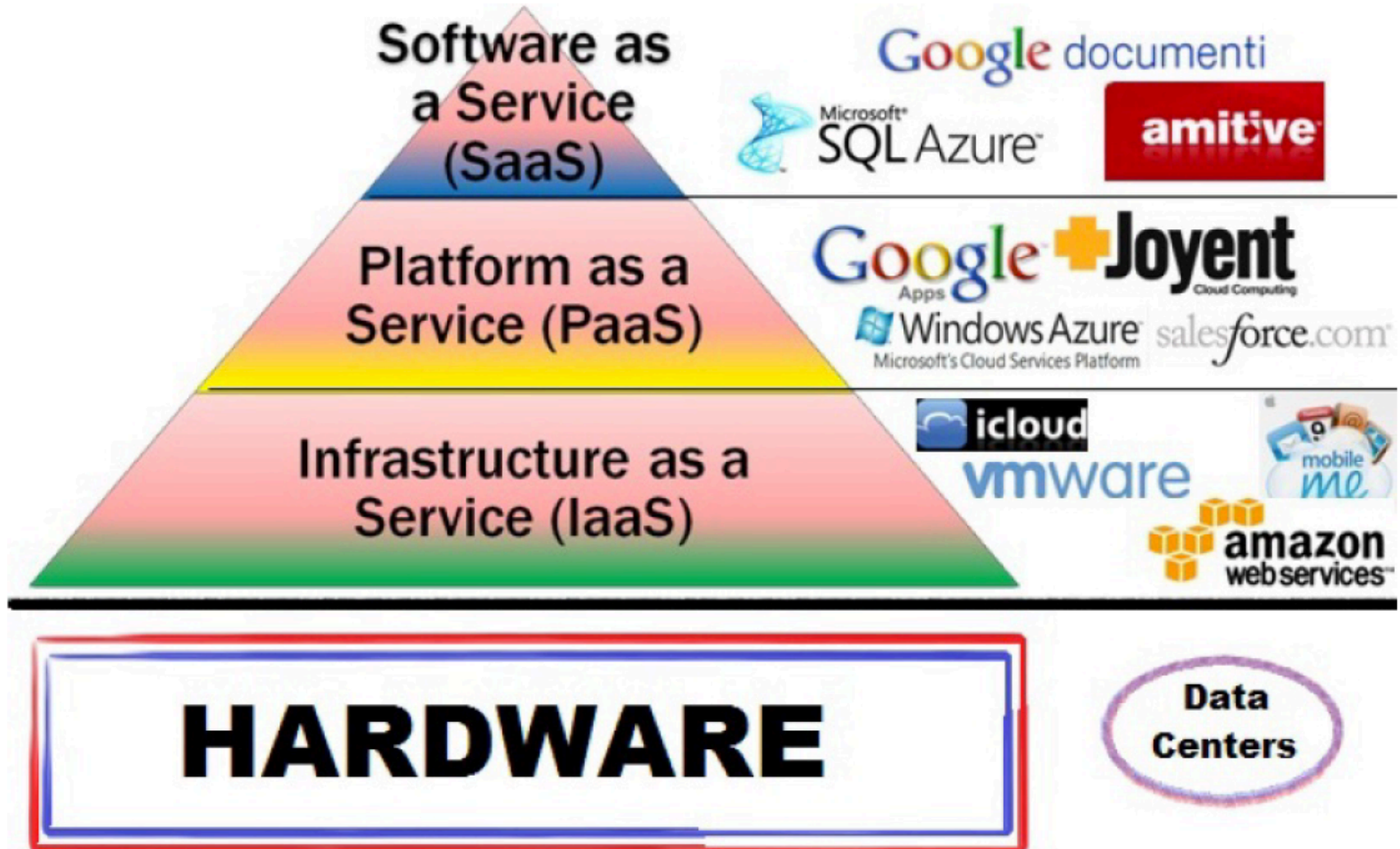
Features in-the-cloud

- Cloud Computing is a generic term used to refer to **Internet-based deployment of services**
- A number of features define data, applications, services, and infrastructures *in-the-cloud*:
 - **Remotely hosted**: Services or data are hosted remotely
 - **Ubiquitous**: Services or data are available from anywhere
 - **Commodified**: The result is a utility computing model similar to that of traditional utilities, like gas and electricity - you pay for what you use

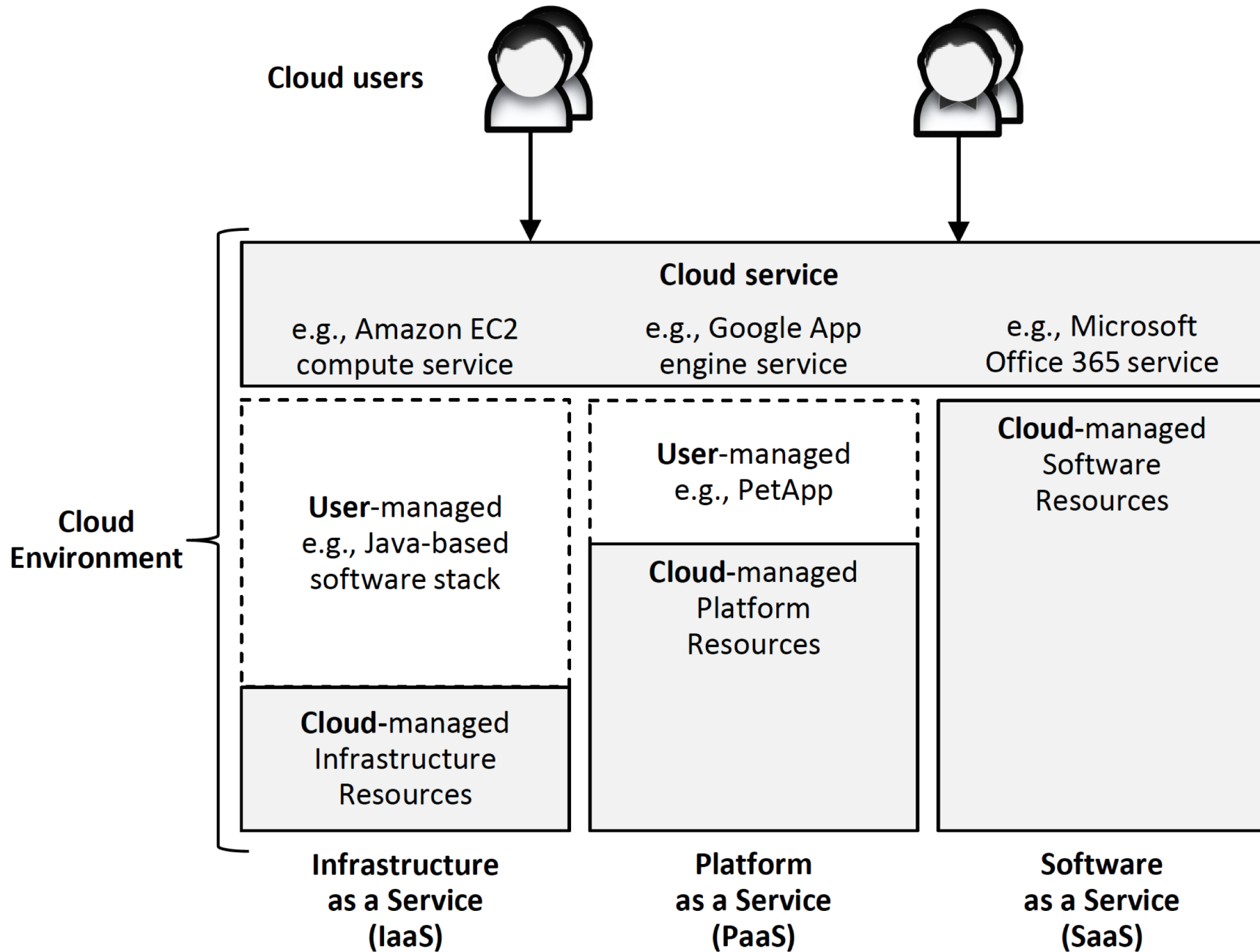
Cloud Architecture



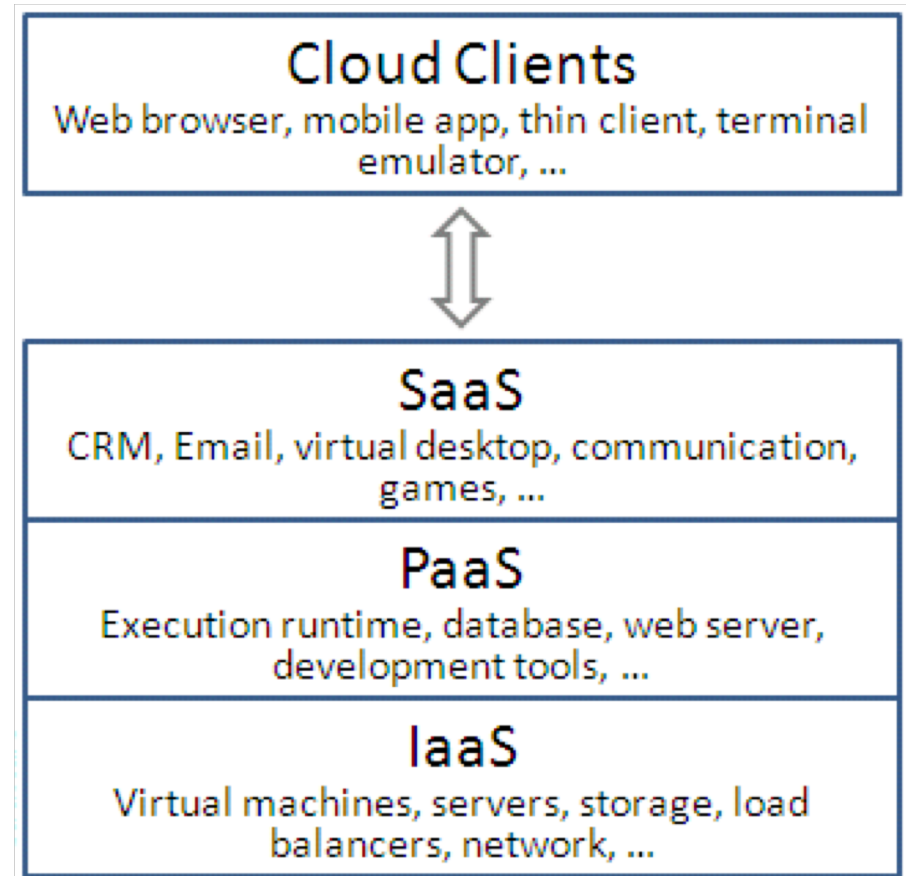
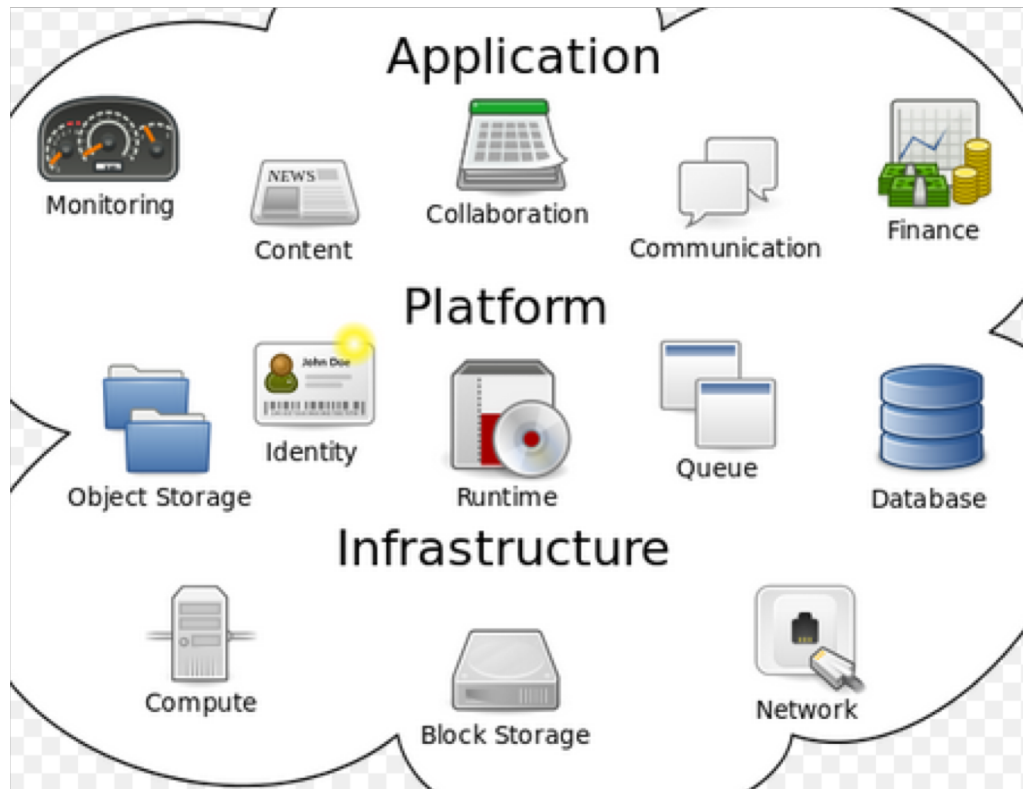
Clouds: examples



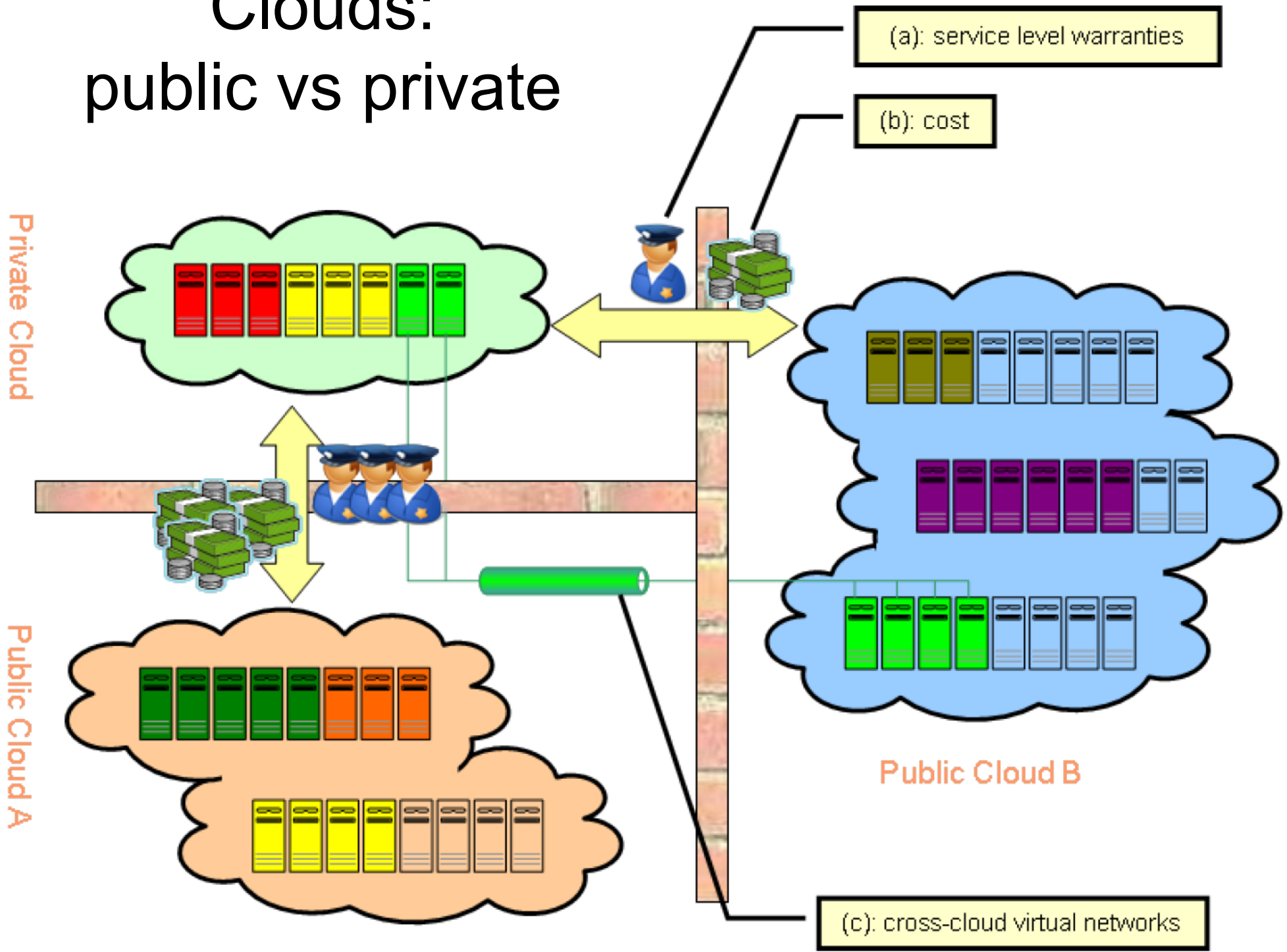
SaaS, PaaS, IaaS

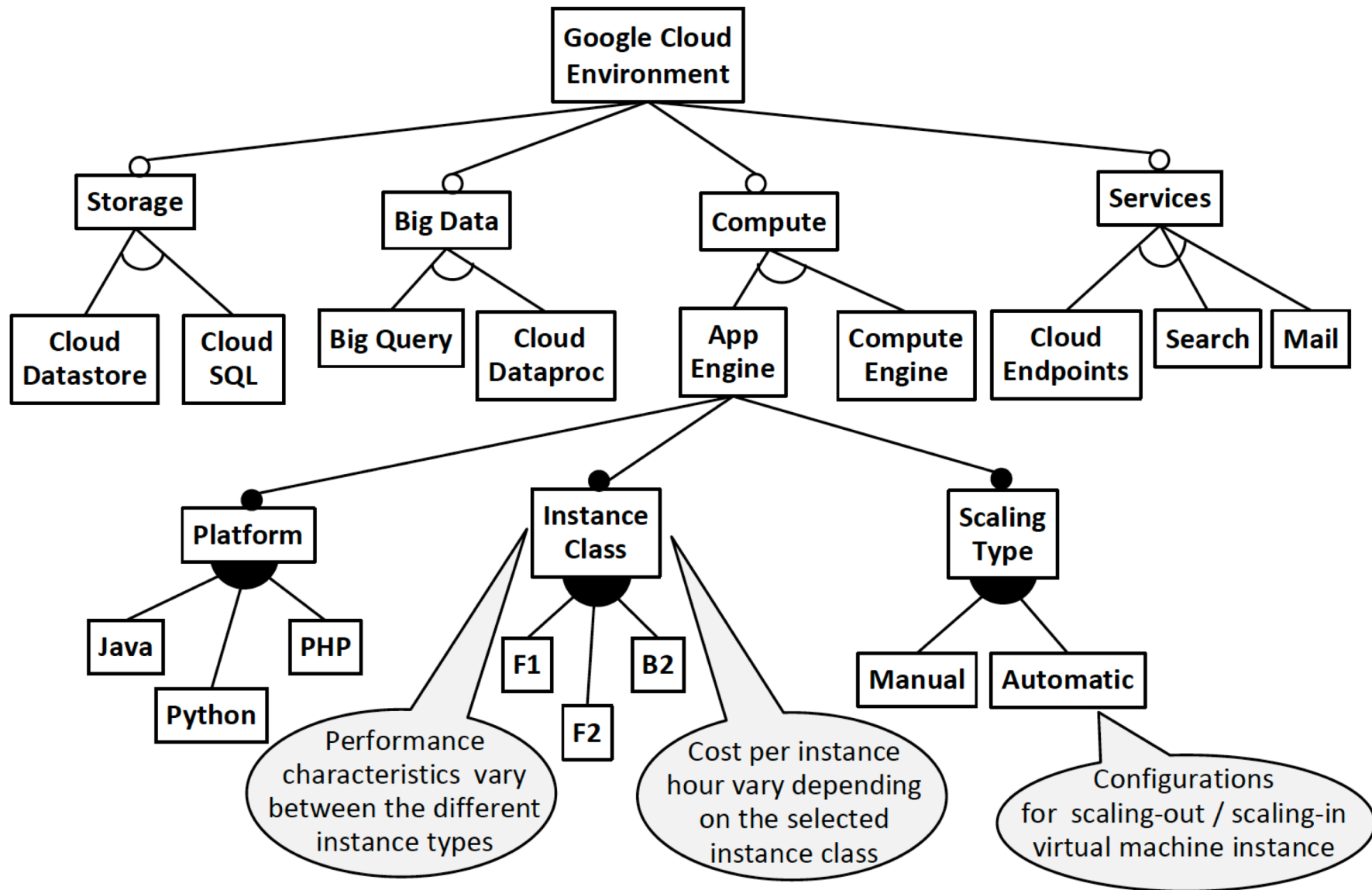


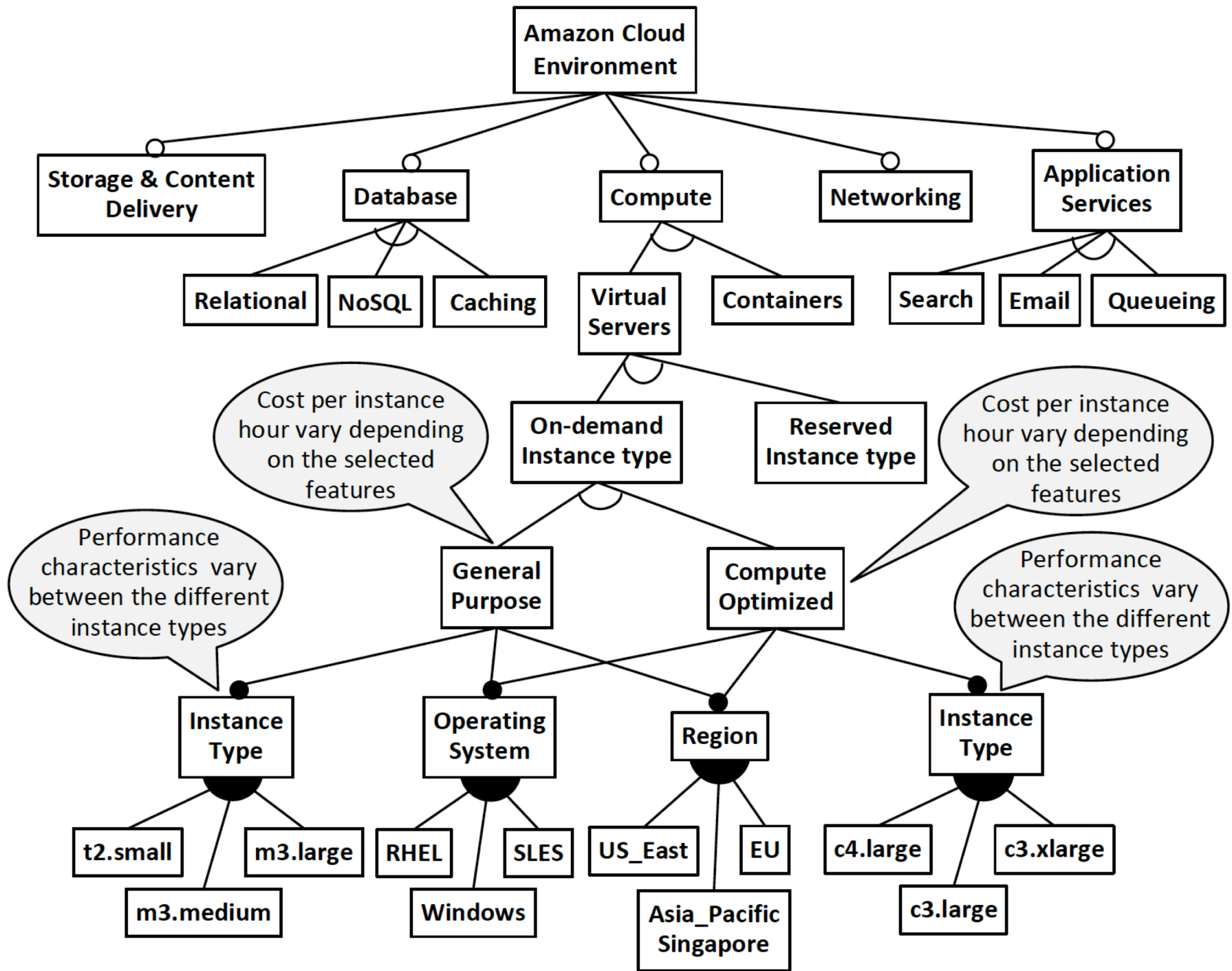
IaaS, PaaS, SaaS

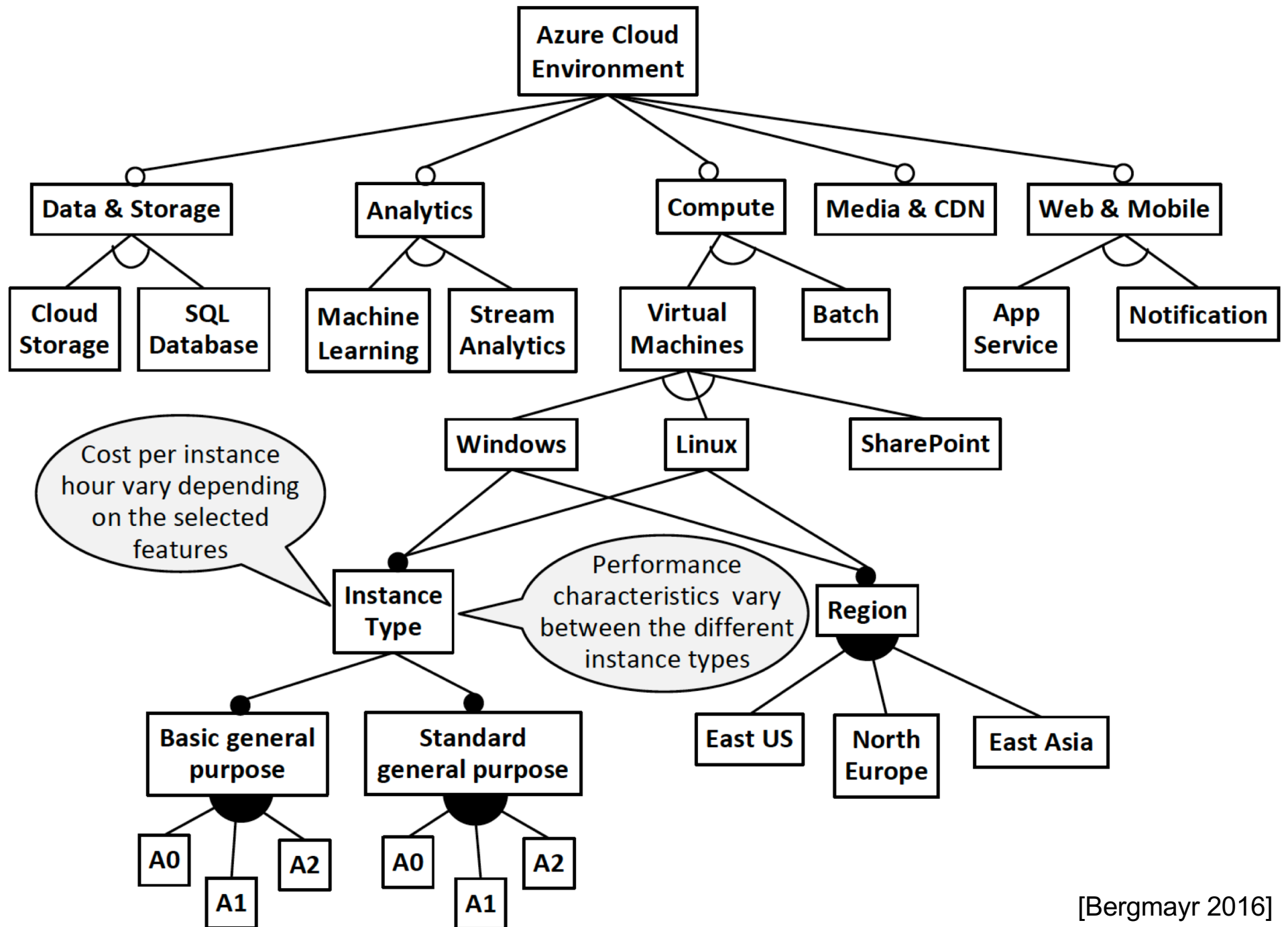


Clouds: public vs private







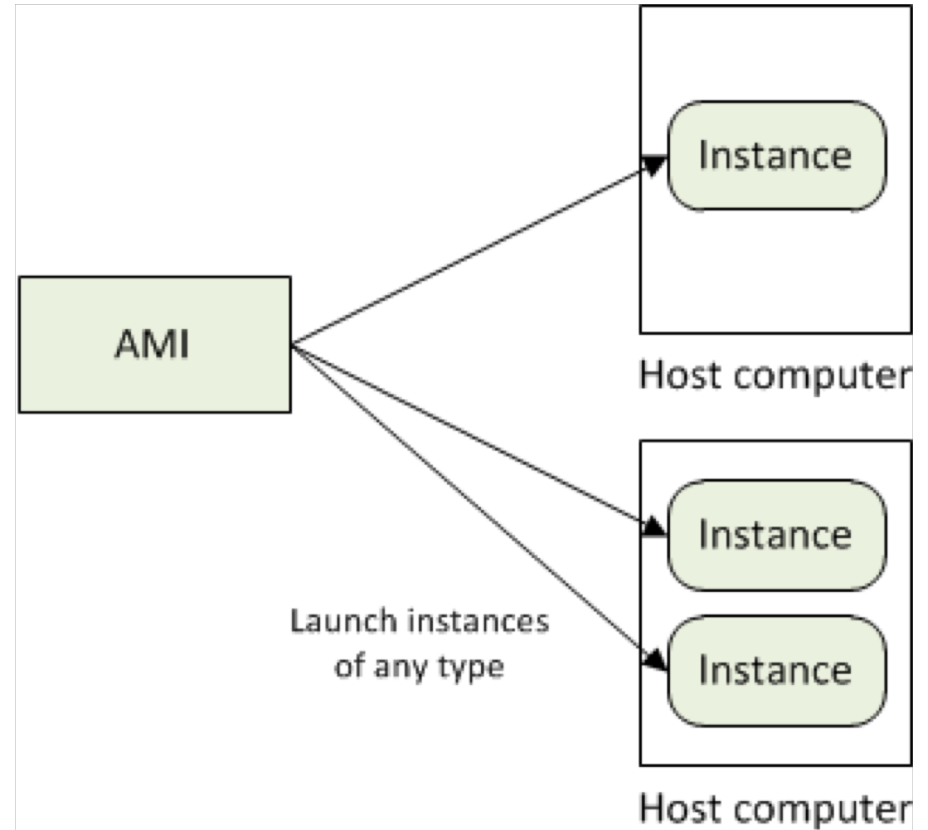


Main types of clouds: IaaS

- **Infrastructure as a service**: Resource Clouds, provide resources as services to the user
- Pay per use based on SLA
- Different resources may be provided :
 - AMI (Amazon Machine Image)
 - Data & Storage Clouds examples:
Amazon S3 (Simple Storage Service), SQL Azure.
 - Compute Clouds examples:
Amazon EC2 (Elastic Cloud Computing), Zimory, ElasticHosts.

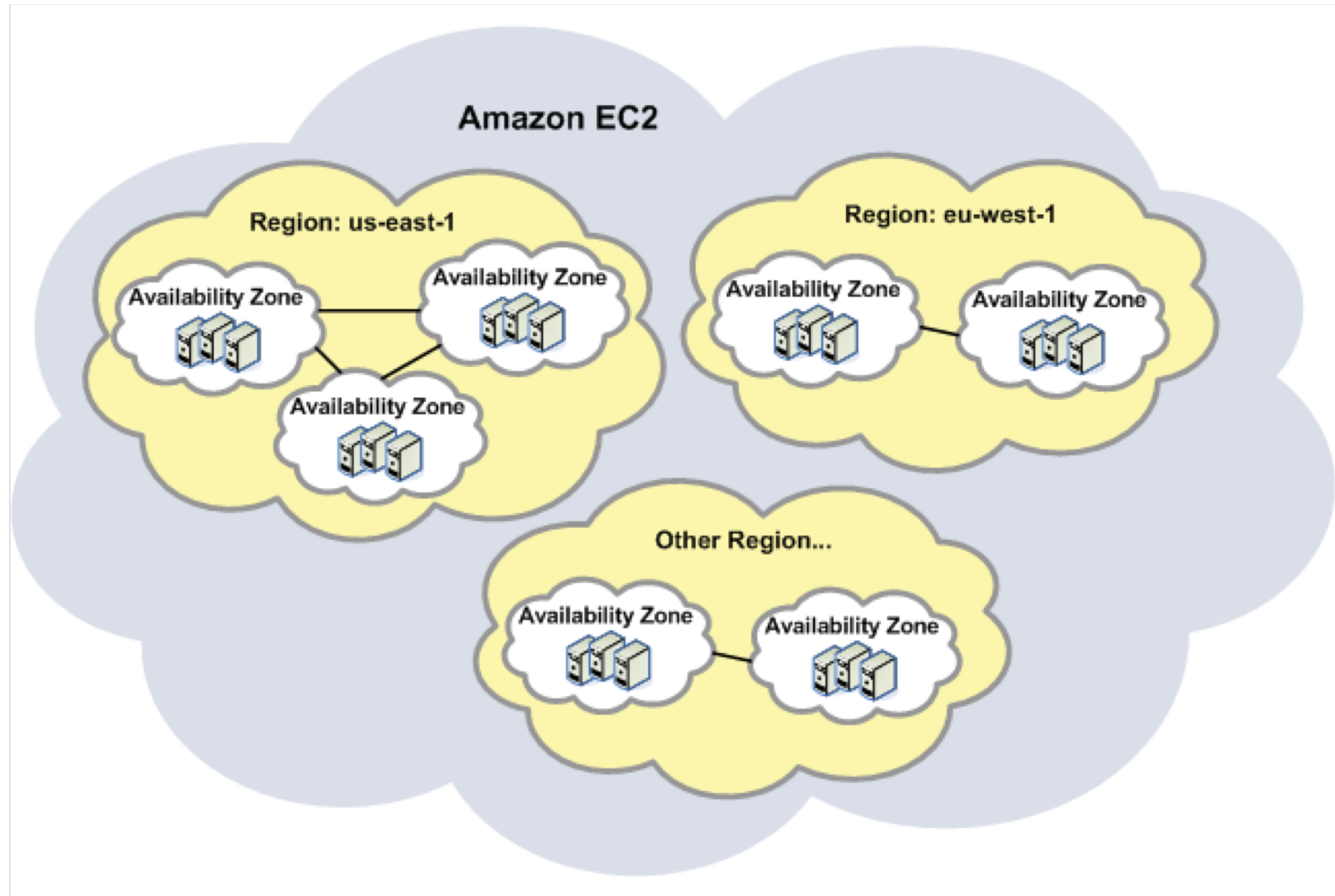
Amazon Machine Images

- An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications)
- From an AMI, you launch *instances*, which are copies of the AMI running as virtual servers in the cloud
- You can launch multiple instances of an AMI

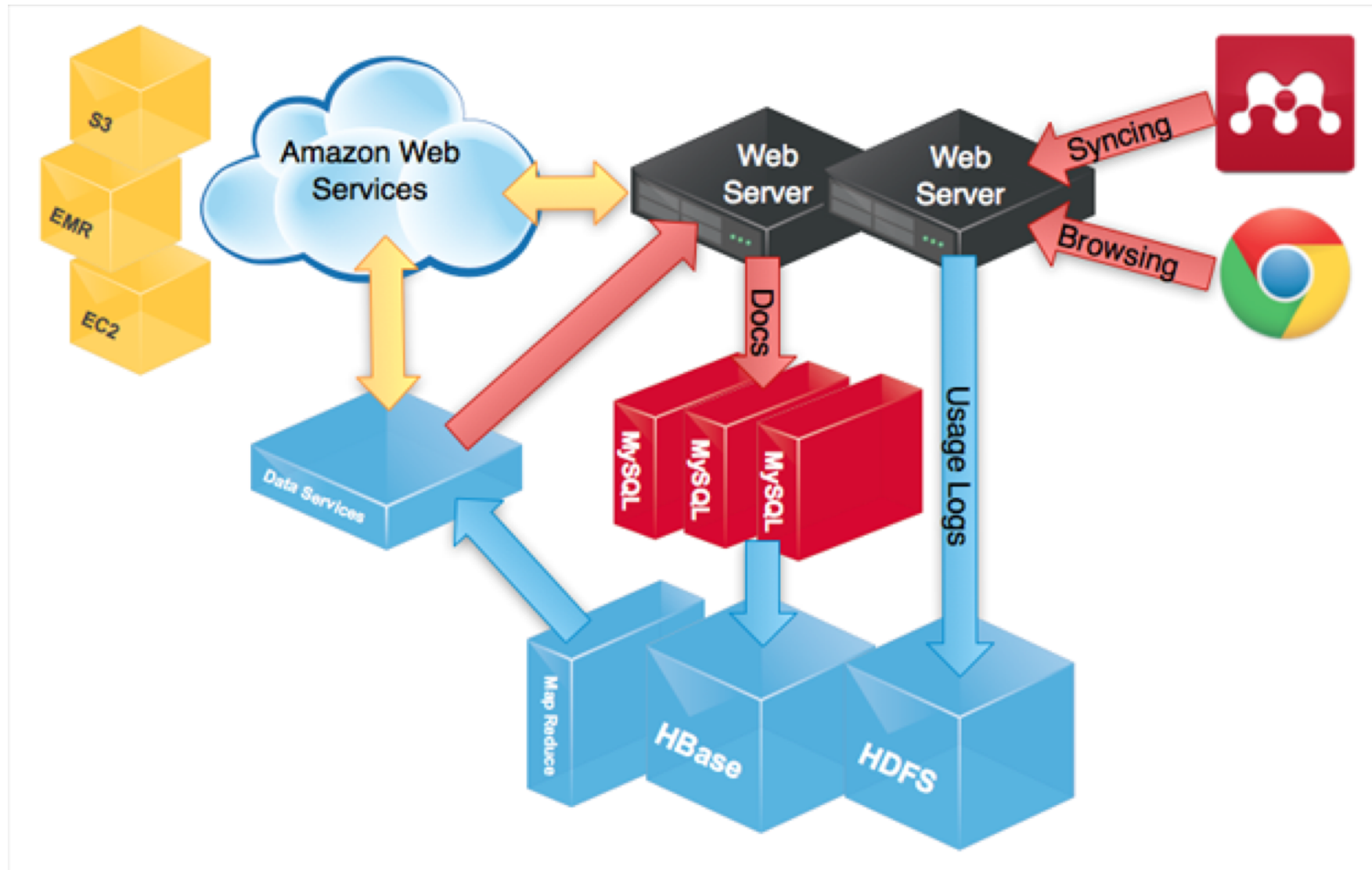


<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

Amazon EC2



Mendeley (Elsevier) case study



Main types of clouds: PaaS

- **Platform as a Service** provides computational resources via a platform upon which applications and services can be developed and hosted.
- PaaS makes use of dedicated APIs to control the behaviour of a server hosting engine which executes and replicates the execution according to user requests (e.g. access rate).
- As each provider exposes its own API, applications developed for one specific cloud provider cannot be moved to another cloud host
- Examples: Force.com, Google App Engine, Heroku (SalesForce), CloudBees, OpenShift, Windows Azure (Platform)
- Pattern: MapReduce

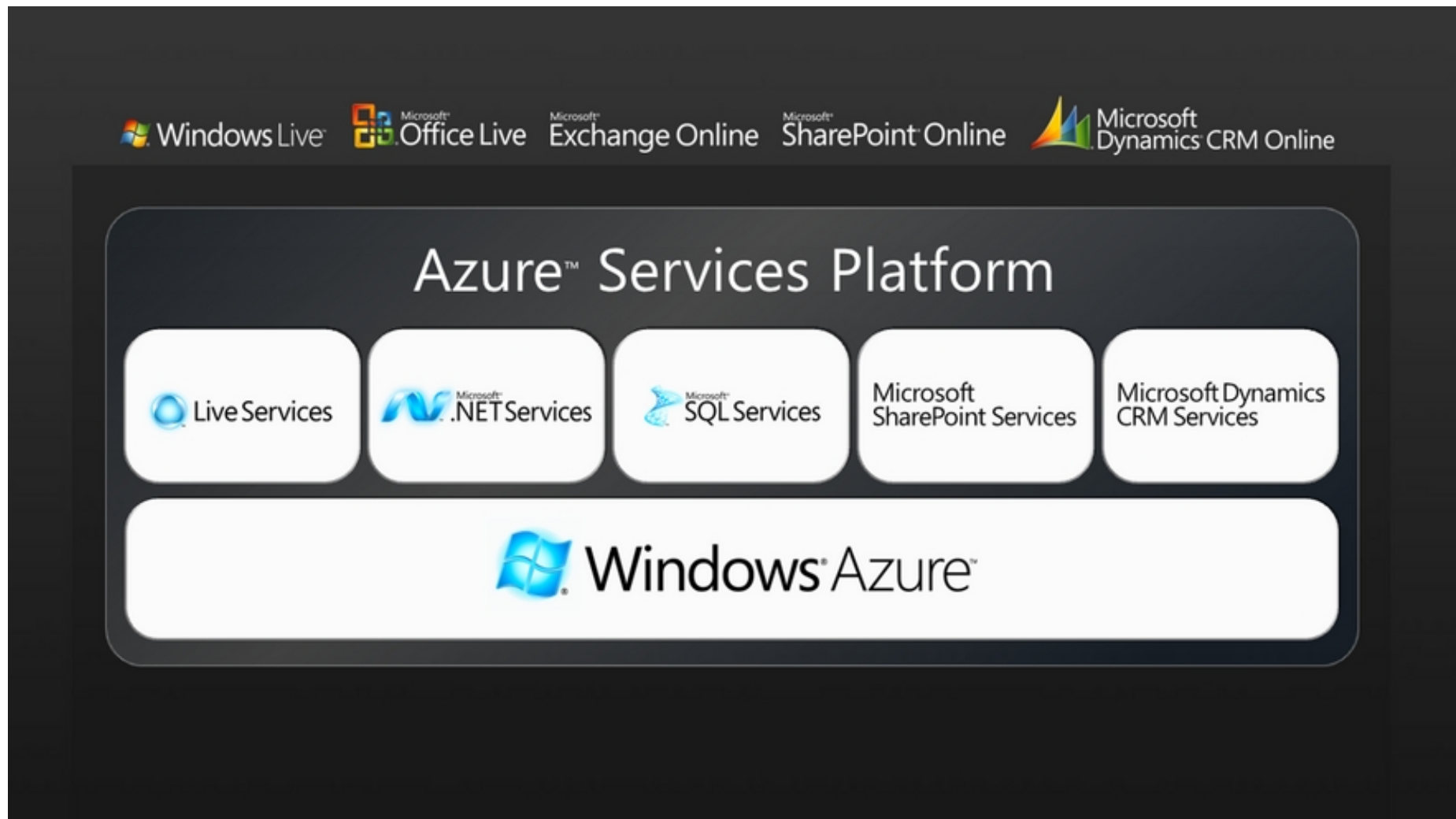
Deployment

- “Deployment-as-a-service” increasingly common
 - monthly pay-as-you-go curated environment (Heroku)
 - hourly pay-as-you-go cloud computing (EC2)
 - **hybrid**: overflow from fixed capacity to elastic capacity
 - Remember administration costs when comparing!
- Good framework can help at deployment time
 - Separate abstractions for different types of state: session state, asset server, caching, database
 - ORM – natural fit for social computing, and abstracts away from SQL (vs Web 1.0 PHP, e.g.)
 - REST – make your app RESTful from start, so that “SOA”-ifying it is trivial
- Scaling structured storage: open challenge

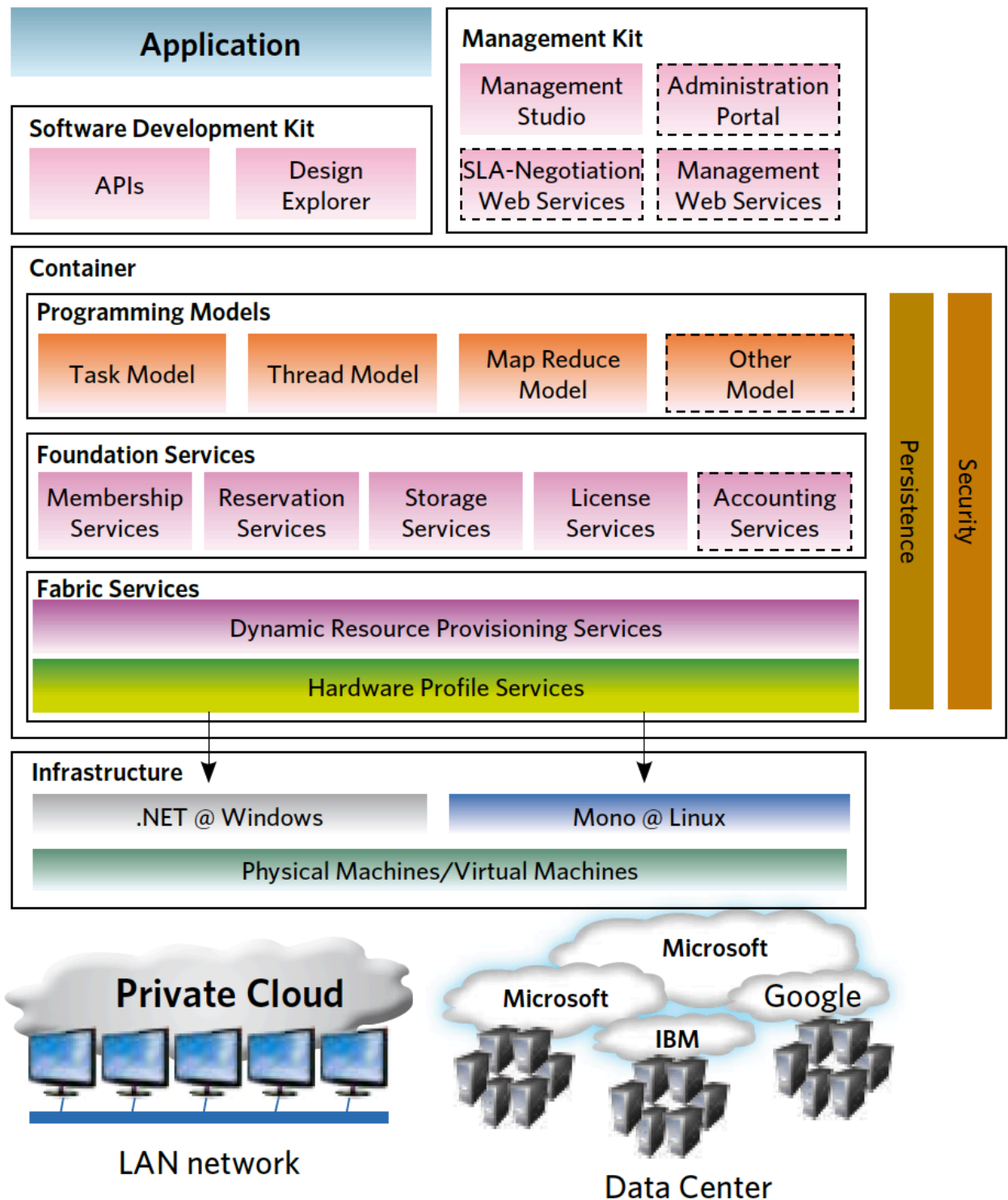
Microsoft Azure

- Windows Azure
 - Compute: Virtualized computing based on Windows
 - ServerStorage: Durable, scalable, & available storage
 - Management: Automated management of the service
- SQL AzureDatabase:
 - Relational processing for structured/unstructured data
- .Net Services
 - Service Bus: General purpose application bus
 - Access Control: Rule-driven
- Azure provides a complete cloud computing stack. The administration interface is simple. It allows to allocate a server or database capacity, hook in the service bus, and configure an application

MS Azure services platform



Aneka



Ranjan, Decentralized Overlay for Federation of Enterprise Clouds, *Handbook of Research on Scalable Computing Technologies*, 2009, IGI Global

Data-as-a-Service

Enterprise

Single users

Common needs:

- Huge quantities of data
- Synchronization, collaboration
- Privacy and security

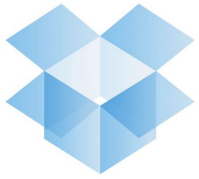
- Need flexibility
- No big initial investments
- Pay for use

- Each user owns several devices
- Social services and sharing
- Usability, user-friendliness
- Enjoy free services

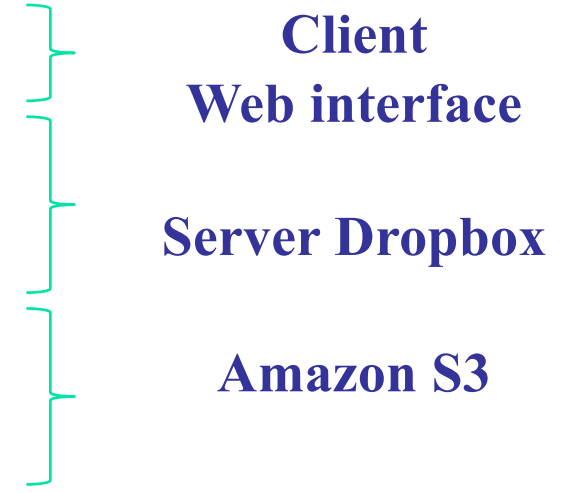
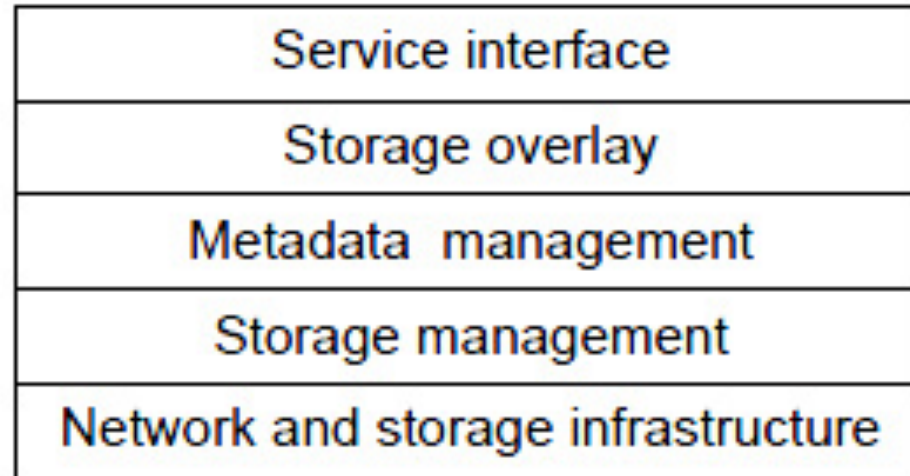
Storage utility



PRODUCT FEATURES	Dropbox	Google Drive	MS Skydrive	Apple iCloud
Free storage (GB)	2 up to 18	5	7 up to 25	5
Monthly price plans (GB)	50 for 10 \$ 100 for 20 \$ more customizable	25 for 2.49 \$ 100 for 4.99 \$ 200 for 9.99 \$ 400 for 19.99 \$	20 for 0.83 \$ 50 for 2.08 \$ 100 for 4.17 \$	20 for 1.67 \$ 40 for 3.33 \$ 100 for 8.33 \$
Max file size (GB)	0.3 (web) 2 (client)	10	2	0.025 (web) 2 (client)
Desktop apps	PC, Mac, Linux	PC, Mac	PC, Mac	Mac
Mobile apps	Android, iOS, Blackberry, Symbian	Andrian, iOS	Windows Phone, iOS	iOS



Dropbox architecture



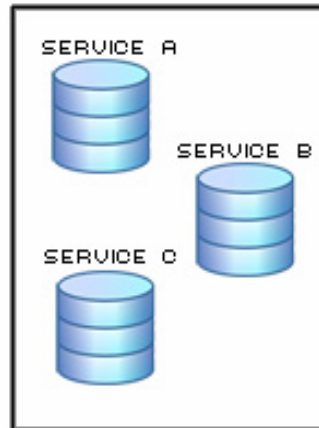
FONTE: [1]



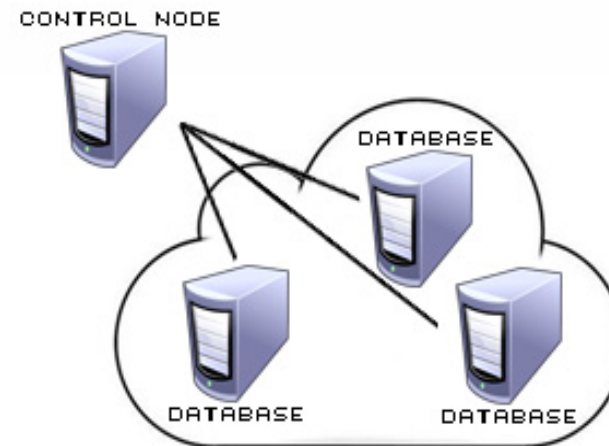
Client devices



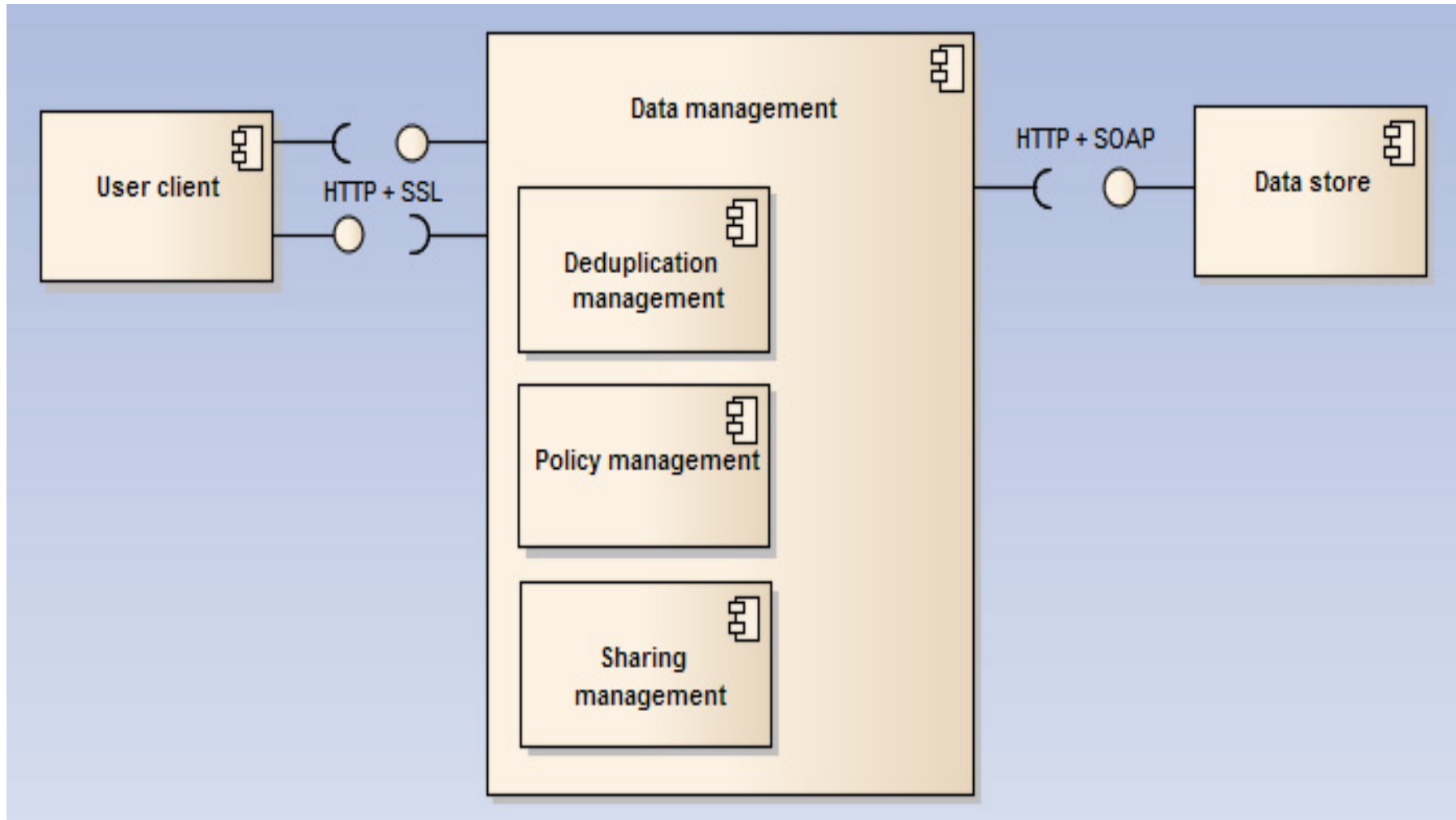
Dropbox services



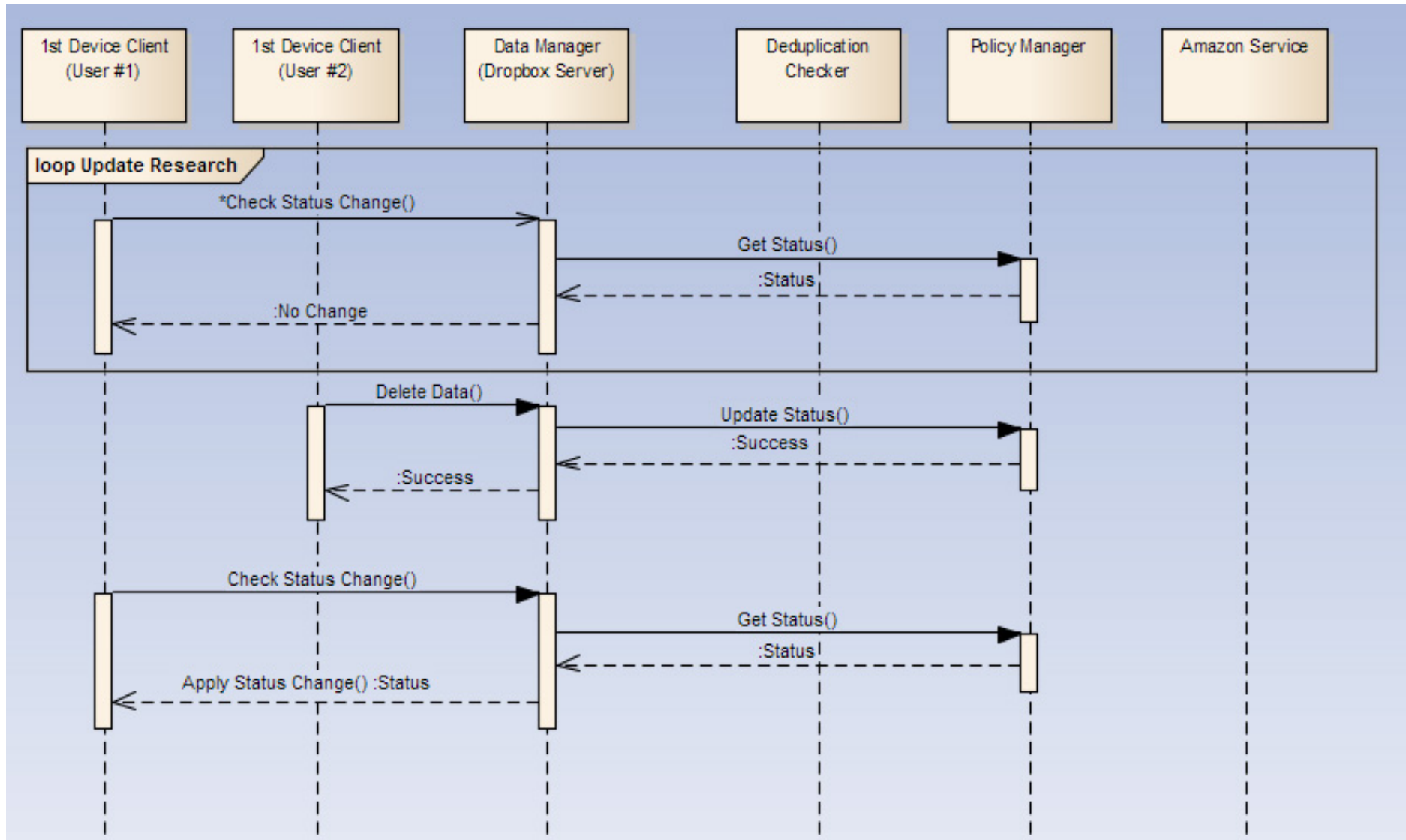
Amazon S3 service



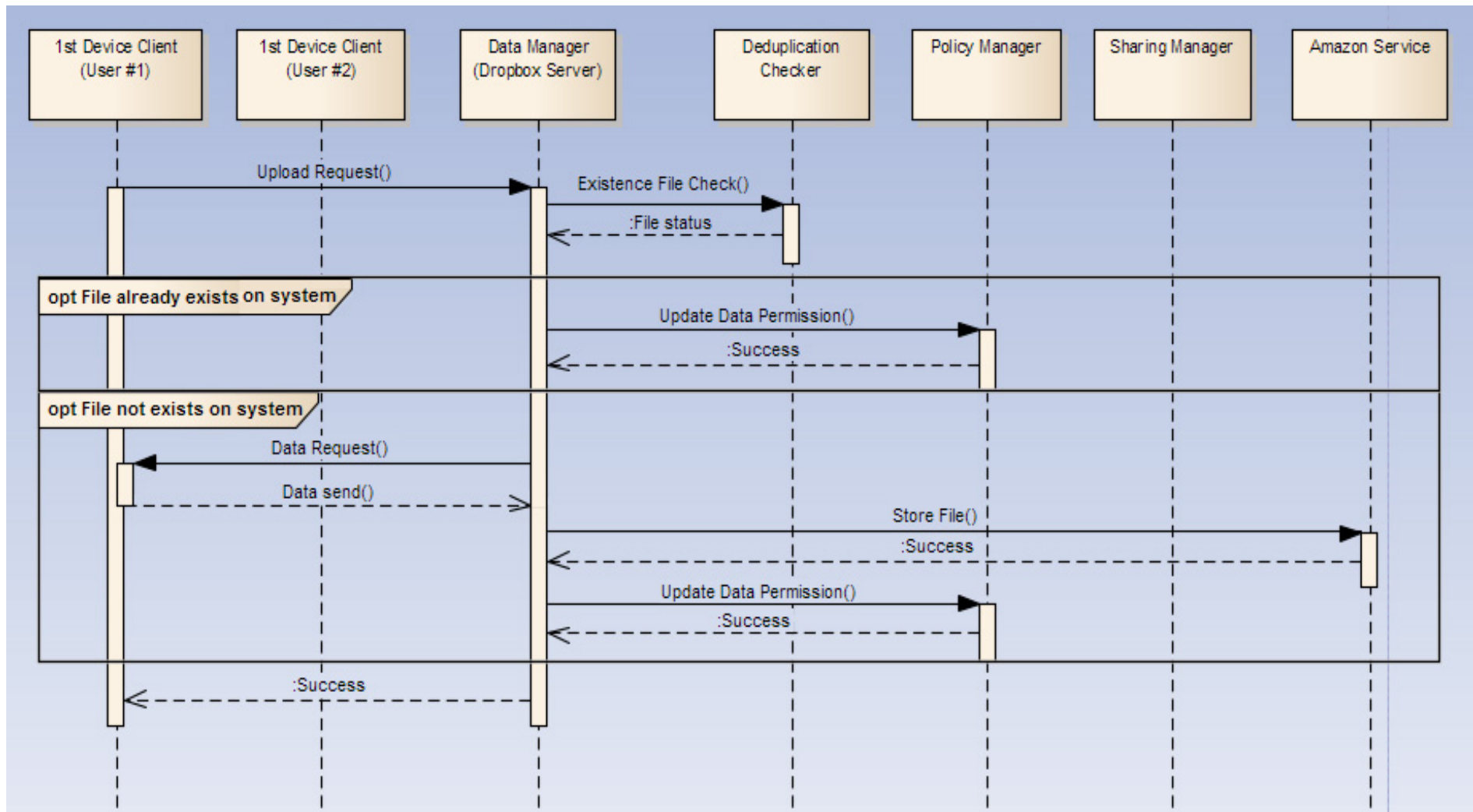
Component diagram



use case: file delete and synchronization

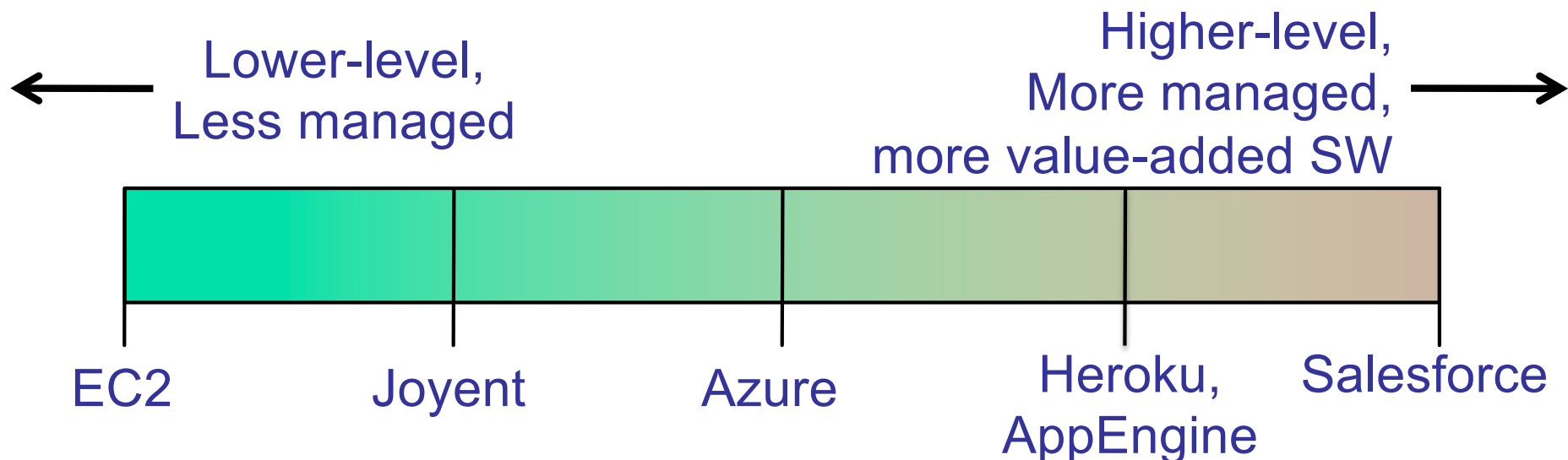


Use case: file upload



Classifying Clouds for Web 2.0

- Instruction Set VM (Amazon EC2)
- Managed runtime VM (Microsoft Azure)
- Curated “IDE-as-a-service” (Heroku)
- Platform as service (Google AppEngine, Force.com)
- *flexibility/portability vs. built-in functionality*

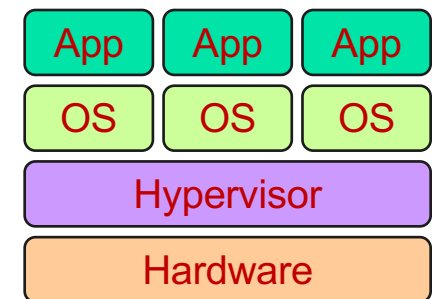


Basic cloud features

- The “**no-need-to-know**” in terms of the underlying details of infrastructure, applications interface with the infrastructure via the APIs
- The “**flexibility and elasticity**” allows these systems to scale up and down at will
 - utilising the resources of all kinds
 - CPU, storage, server capacity, load balancing, and databases
- The “**pay as much as used and needed**” type of utility computing and the “**always on, anywhere and any place**” type of network-based computing

Virtualization

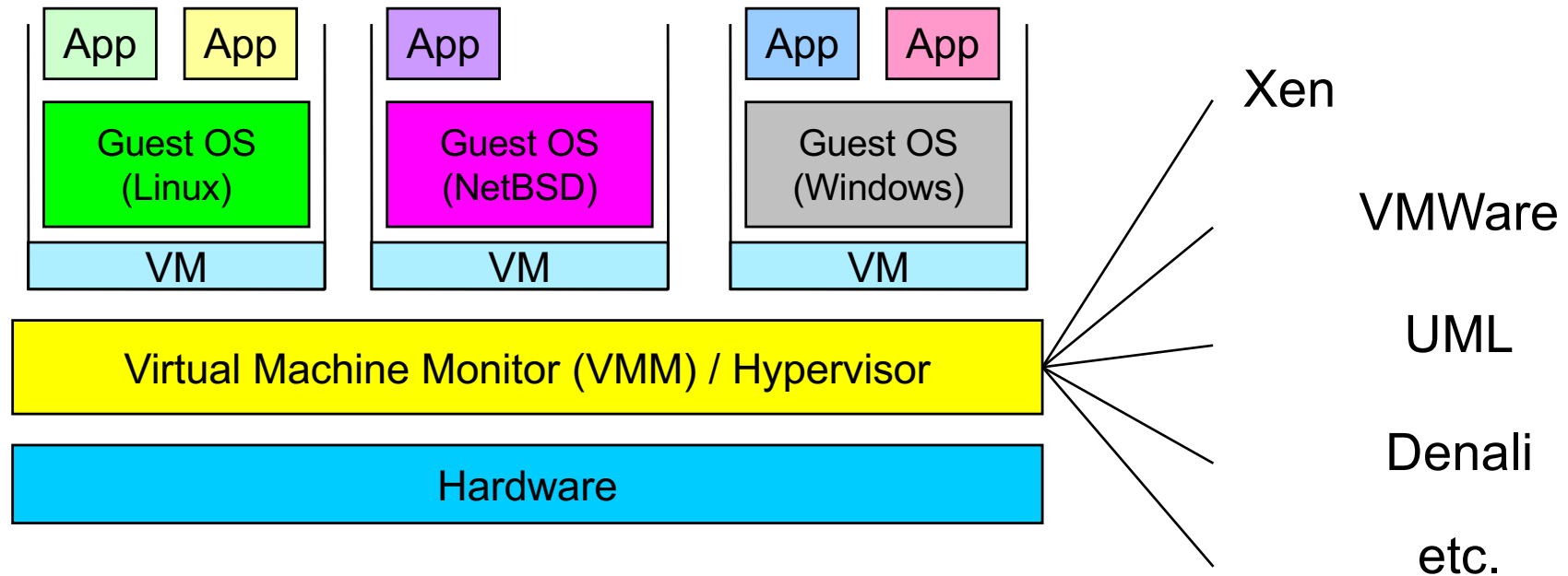
- Virtual workspaces:
 - An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols,
 - Resource quota (e.g. CPU, memory share),
 - Software configuration (e.g. O/S, provided services).
- Implement on Virtual Machines (VMs):
 - Abstraction of a physical host machine,
 - Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs,
 - VMWare, Xen, etc.
- Provide infrastructure API:
 - Plug-ins to hardware/support structures



Virtualized Stack

Virtual Machines

- VM technology allows multiple virtual machines to run on a single physical machine.

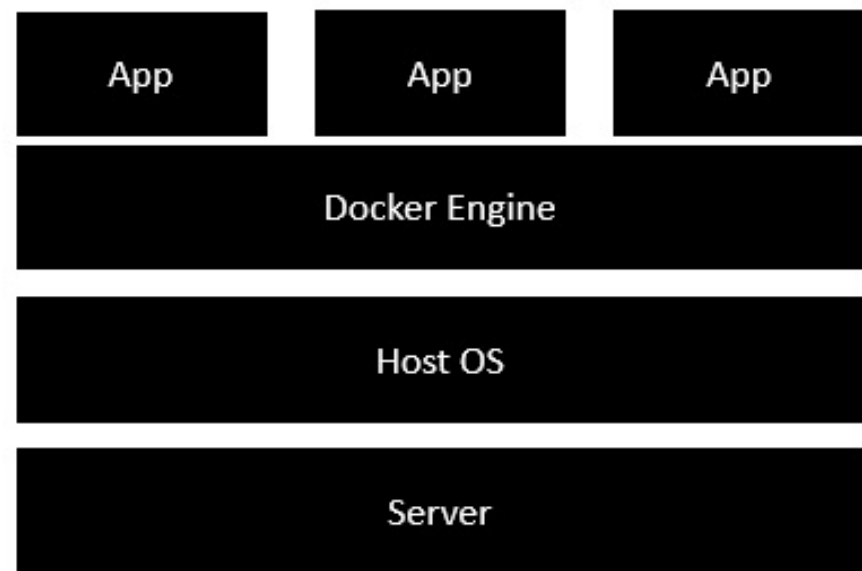
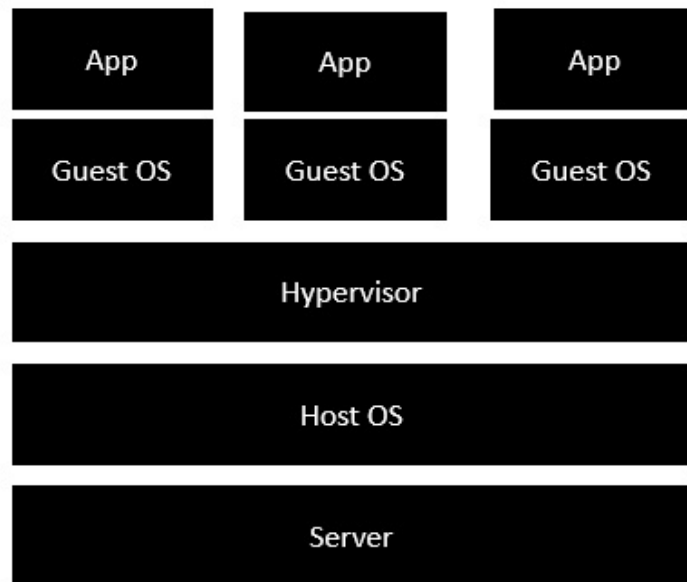


Performance: Para-virtualization (e.g. Xen) is very close to raw physical performance!

Docker

Docker containers wrap a software in a complete filesystem that contains everything needed to run: code, runtime, system tools, libraries – anything that can be installed on a server.

This guarantees that the software will always run the same, regardless of its environment.



A design pattern for cloud computing: MapReduce

Challenges of a Cloud

- Cheap nodes fail, especially when you have many
 - Mean time between failures for 1 node = 3 years
 - MTBF for 1000 nodes = 1 day
 - **Solution:** Build fault tolerance into system
- Commodity network = low bandwidth
 - **Solution:** Push computation to the data
- Programming distributed systems is hard
 - **Solution:** Restricted programming model: users write data-parallel “map” and “reduce” functions, system handles work distribution and failures

File systems for big data

- A new challenge in the form of web logs, or web crawler's data: large scale “peta scale”
- Big data are very different from transactional or “customer order” data : “write once read many (WORM)”; for instance
 - Google copies of web pages;
 - Privacy protected healthcare and patient information;
 - Historical financial data;
 - Climate data
- A solution: Google File System (GFS)

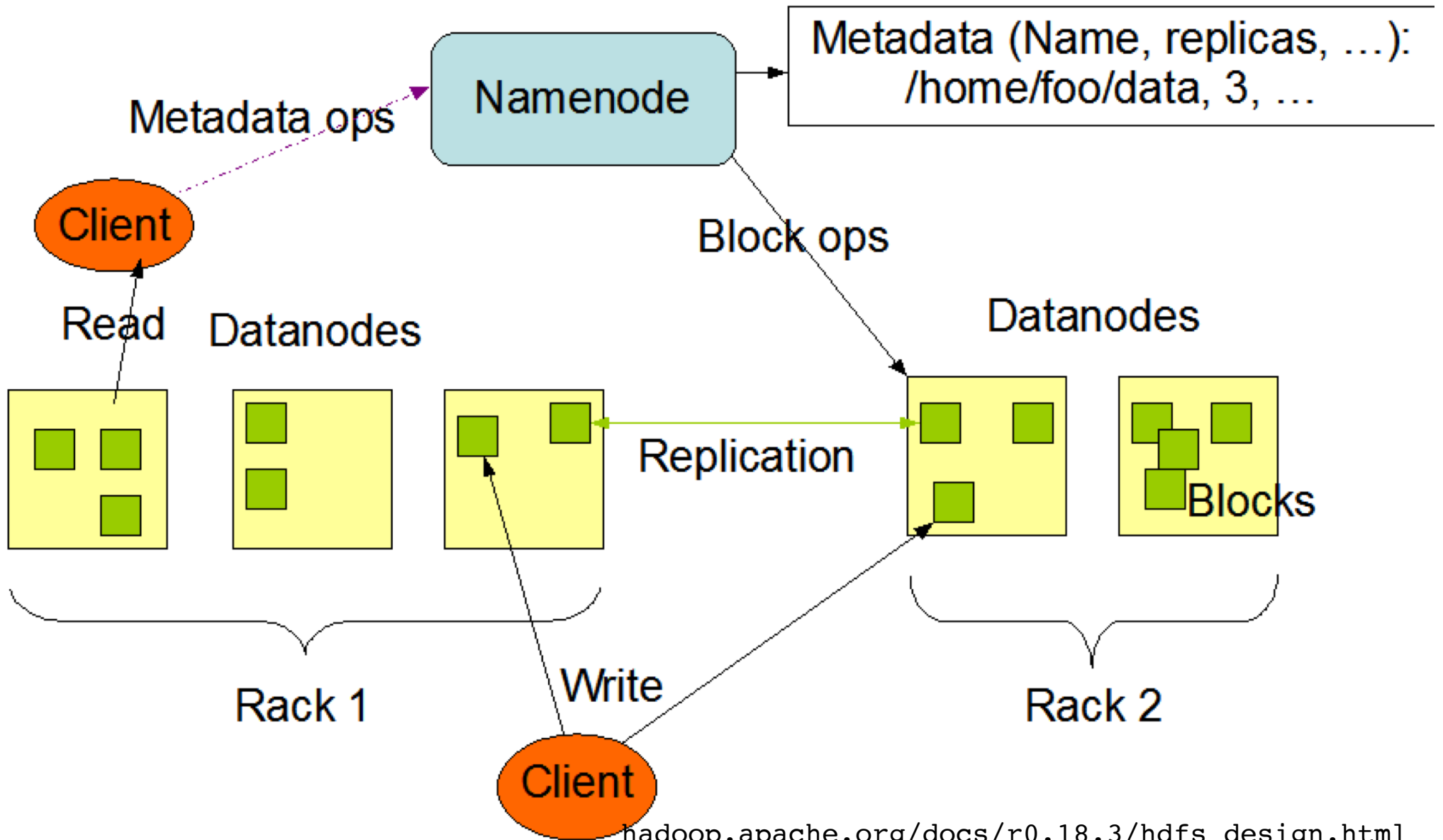
What is Hadoop?

- At Google operations are run on a special file system called Google File System (GFS) that is highly optimized for big chunks (1 file: 100 GB)
- GFS is not open source
- Doug Cutting and others at Yahoo! reverse engineered the GFS and called it Hadoop Distributed File System (HDFS).
- The software framework that supports HDFS and related entities (eg MapReduce) is called Hadoop
- This is open source and distributed by Apache

Fault tolerance in HDFS

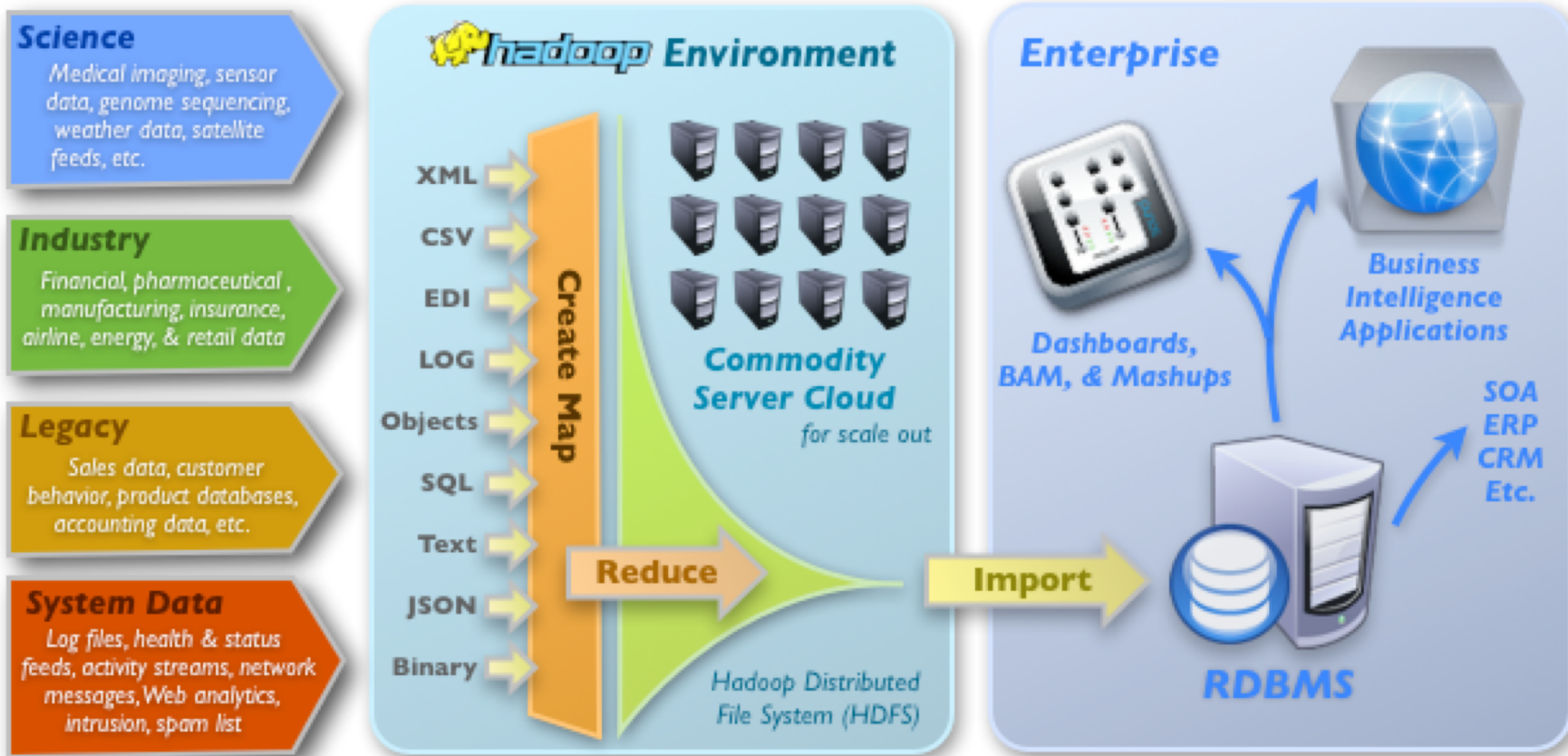
- Failure is the norm rather than exception
- A HDFS instance may consist of thousands of server machines, each storing part of the file system's data.
- Since we have huge number of components and that each component has non-trivial probability of failure means that there is always some component that is non-functional.
- Detection of faults and quick, automatic recovery from them is a core architectural goal of HDFS

HDFS Architecture



Hadoop (Apache)

Using Hadoop in the Enterprise



1 **High Volume Data Flows**

2 **MapReduce Process**

3 **Consume Results**

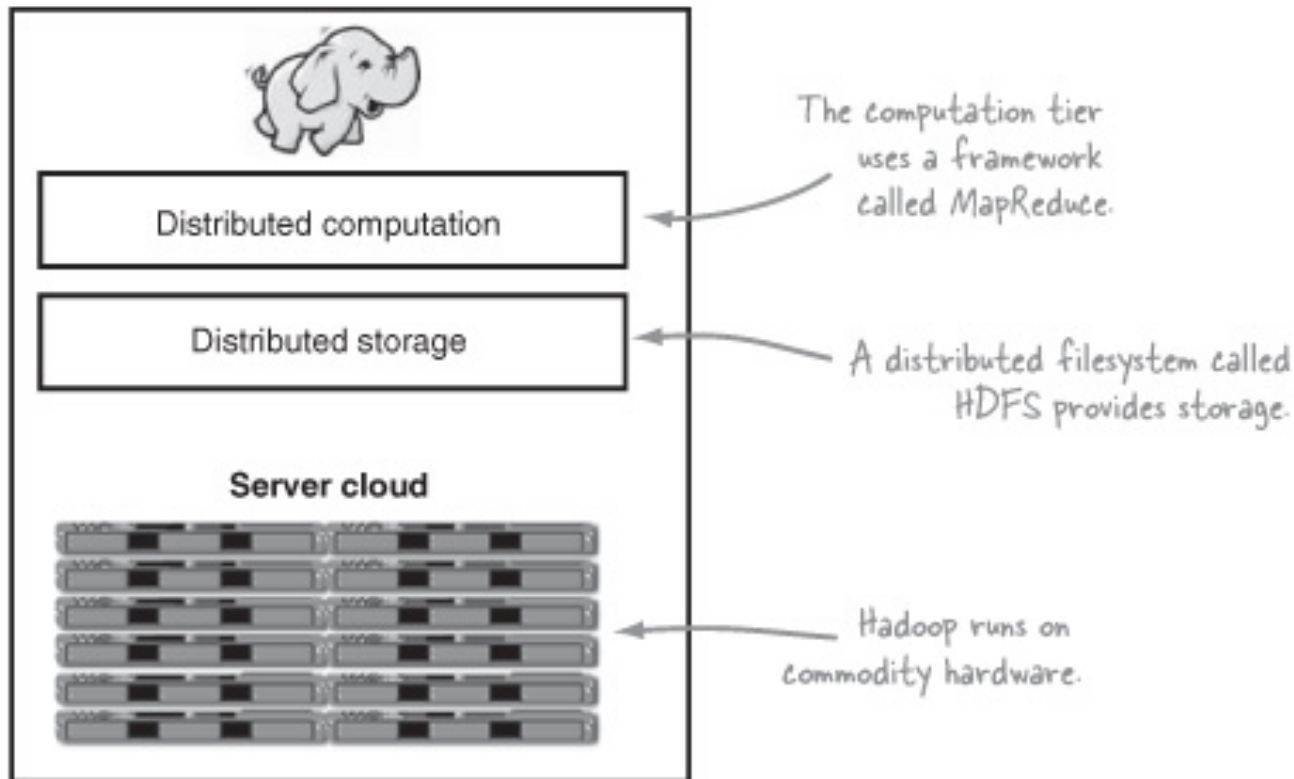
From <http://www.ebizq.net/blogs/enterprise>

Typical Hadoop Cluster

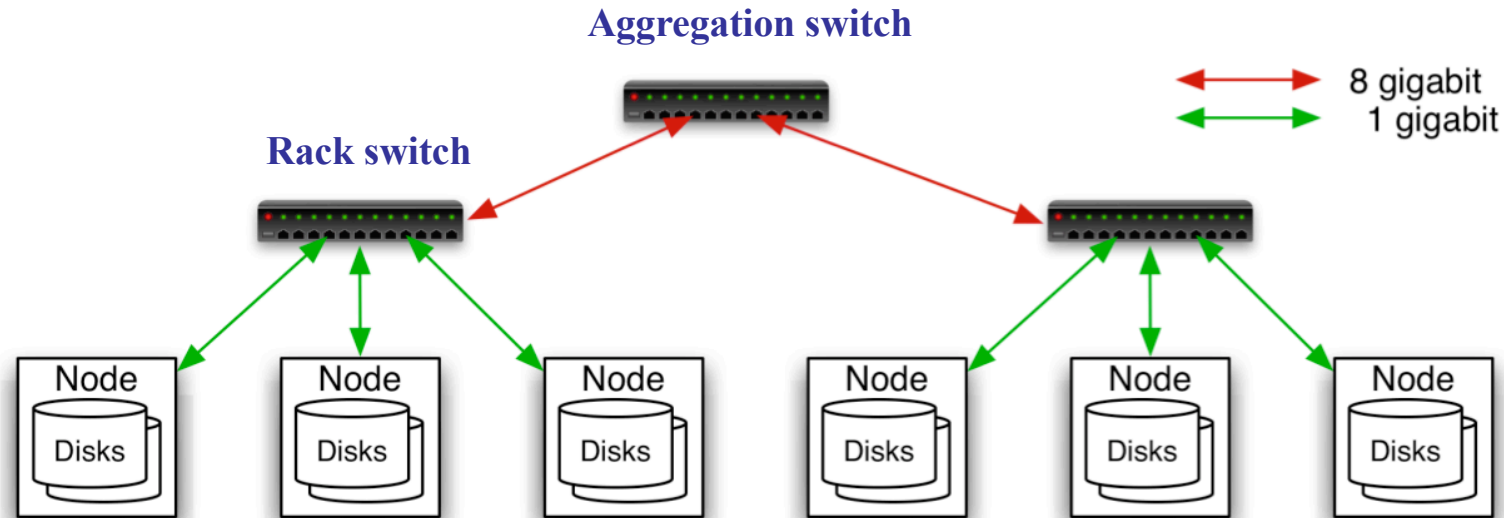


Hadoop Components

- Distributed file system (HDFS)
 - Single namespace for entire cluster
 - Replicates data 3x for fault-tolerance
- MapReduce framework
 - Runs jobs submitted by users
 - Manages work distribution & fault-tolerance
 - Colocated with file system



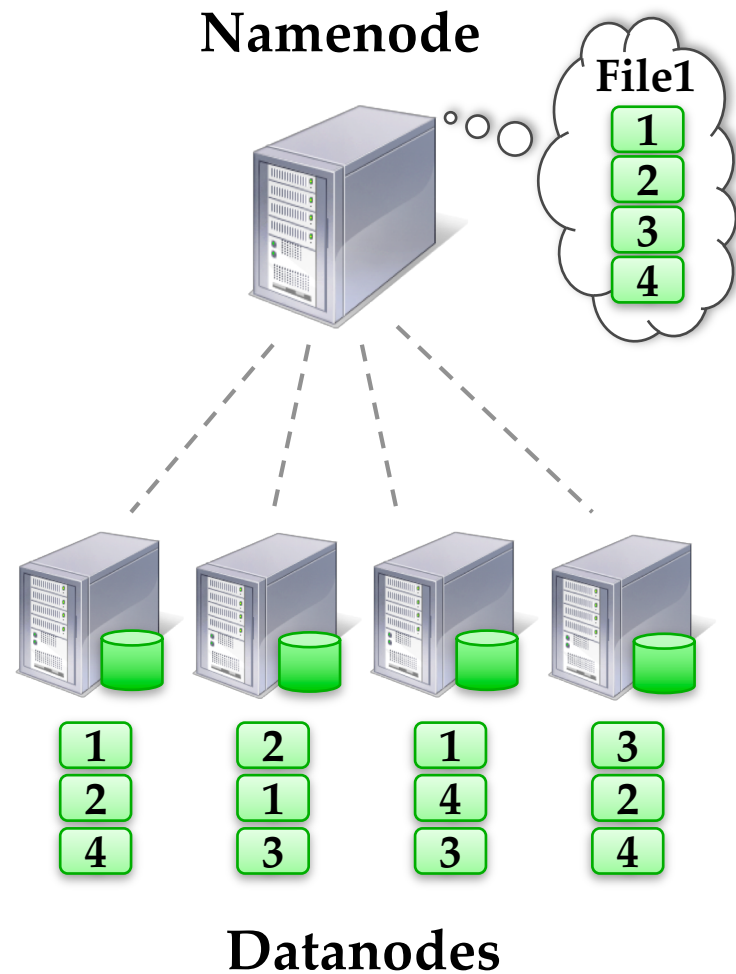
Hadoop



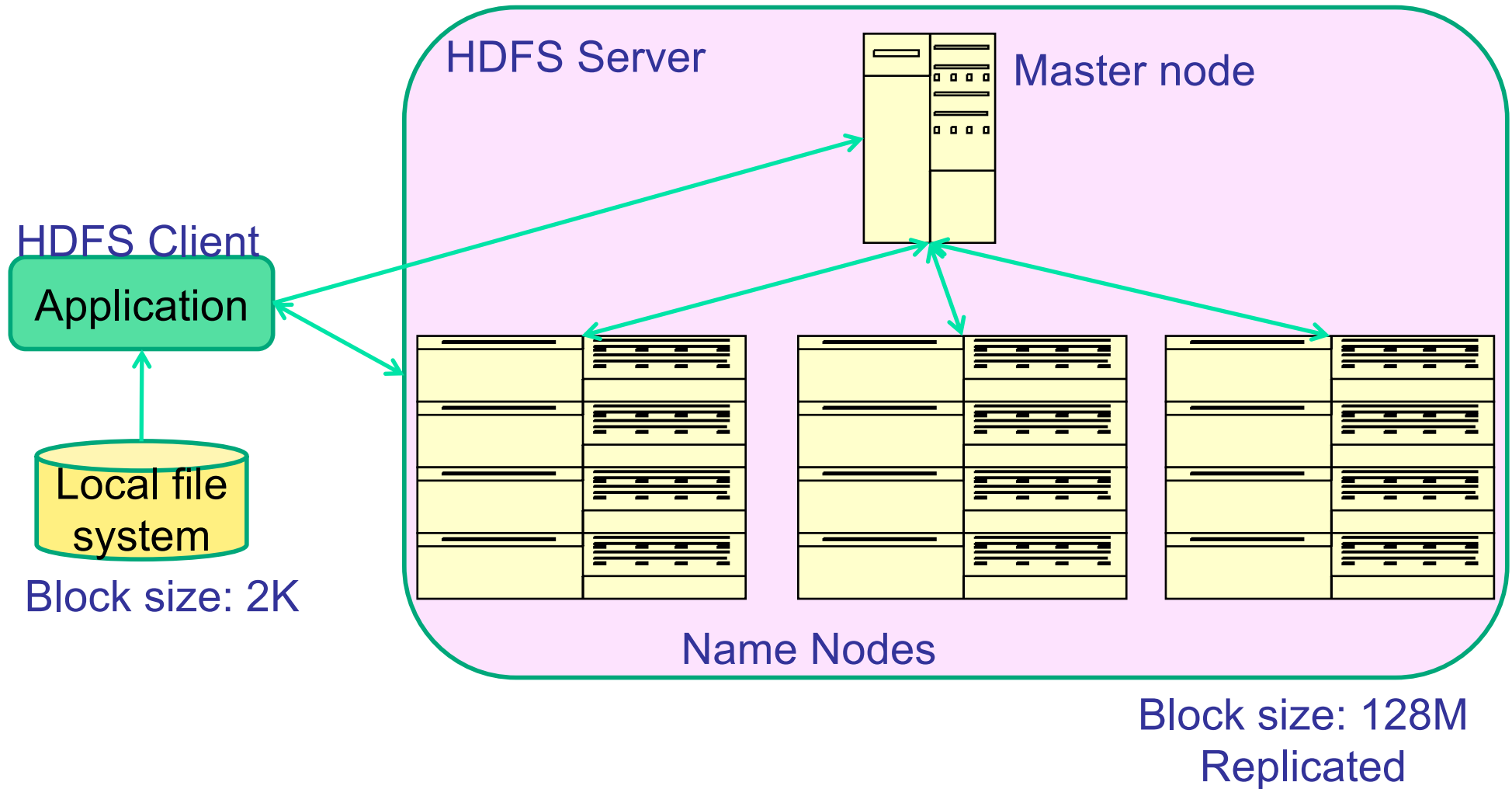
- 40 nodes/rack, 1000-4000 nodes in cluster
- 1 Gbps bandwidth in rack, 8 Gbps out of rack
- Node specs (Facebook):
8-16 cores, 32 GB RAM, 8 × 1.5 TB disks, no RAID

Hadoop Distributed File System

- Files split into 128MB blocks
- Blocks replicated across several datanodes (often 3)
- Namenode stores metadata (file names, locations, etc)
- Optimized for large files, sequential reads
- Files are append-only



Hadoop Distributed File System



MapReduce: an architectural style for CC

- MapReduce: a data-intensive programming paradigm and style for computing on clouds
- Pioneered by Google on GFS
 - Processes 20 PB of data per day
- Popularized by the Apache Hadoop project
 - Used by Yahoo!, Facebook, Amazon, ...

What is MapReduce used for

- At Google:
 - Index building for Google Search
 - Article clustering for Google News
 - Statistical machine translation
- At Yahoo!:
 - Index building for Yahoo! Search
 - Spam detection for Yahoo! Mail
- At Facebook:
 - Data mining
 - Ad optimization
 - Spam detection

What is MapReduce used for?

- In research:
 - Analyzing Wikipedia conflicts (PARC)
 - Natural language processing (CMU)
 - Climate simulation (Washington)
 - Bioinformatics (Maryland)
 - Particle physics (Nebraska)
 - **<Your application here>**

MapReduce goals

- **Scalability** to large data volumes:
 - Scan 100 TB on 1 node @ 50 MB/s = 24 days
 - Scan on 1000-node cluster = 35 minutes
- **Cost-efficiency:**
 - Commodity nodes (cheap, but unreliable)
 - Commodity network (low bandwidth)
 - Automatic fault-tolerance (fewer admins)
 - Easy to use (fewer programmers)

MapReduce

- Data type: key-value *records*

- Map function:

$$(K_{in}, V_{in}) \rightarrow \text{list}(K_{inter}, V_{inter})$$

- Reduce function:

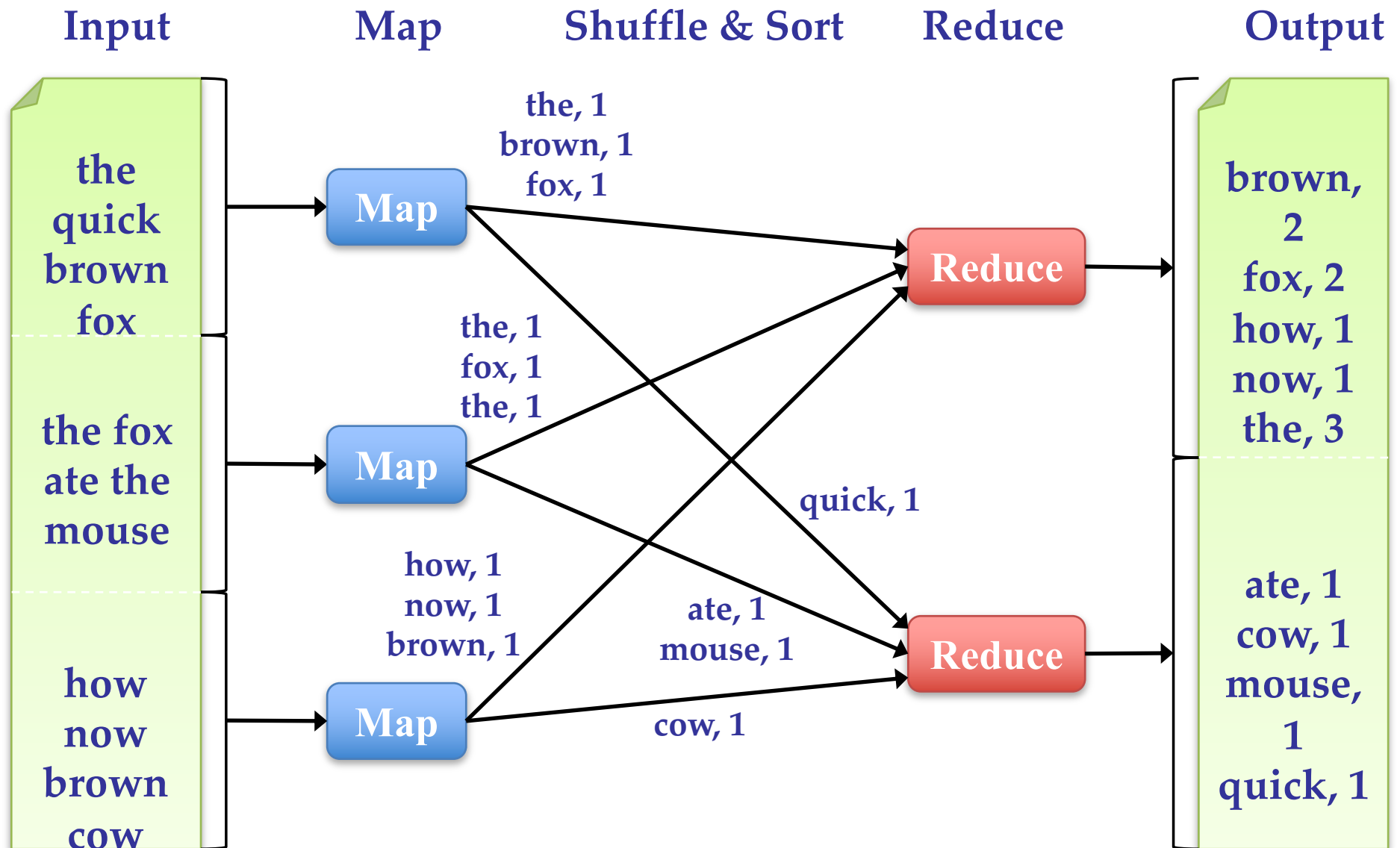
$$(K_{inter}, \text{list}(V_{inter})) \rightarrow \text{list}(K_{out}, V_{out})$$

Example: Word Count

```
def mapper(line):  
    foreach word in line.split():  
        output(word, 1)
```

```
def reducer(key, values):  
    output(key, sum(values))
```

Word Count Execution

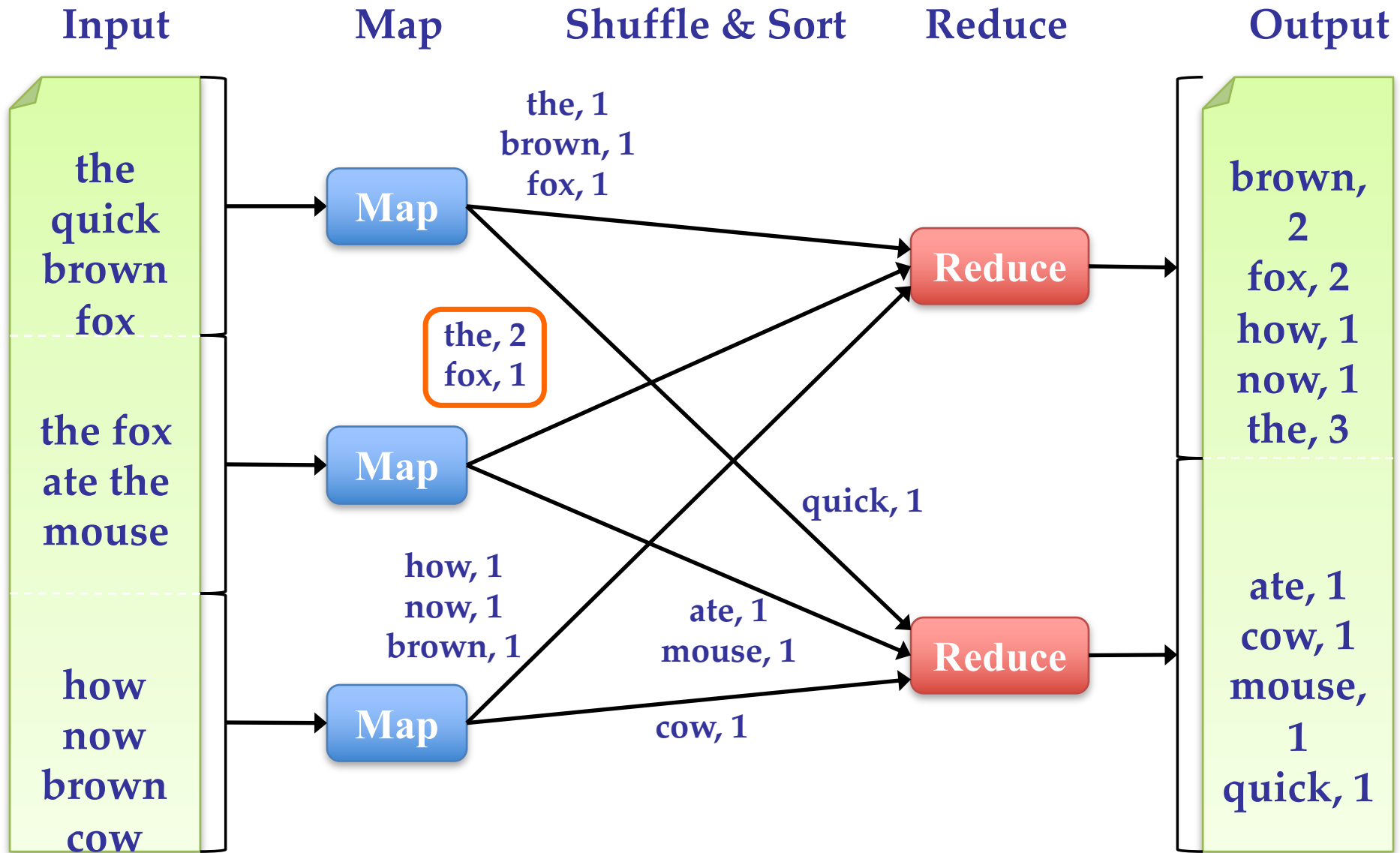


An optimization: the Combiner

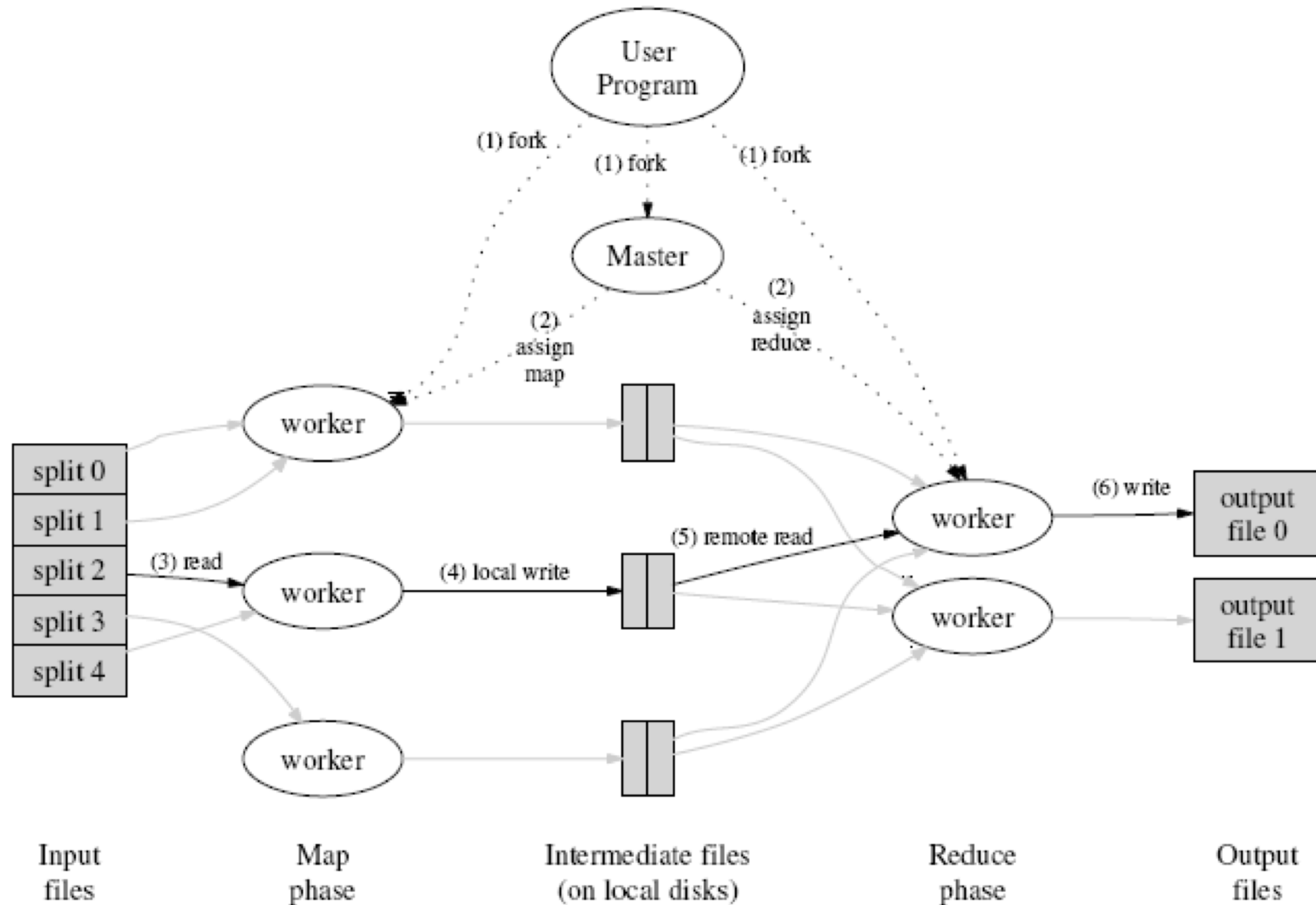
- Local reduce function for repeated keys produced by same map
- For associative ops. like sum, count, max
- Decreases amount of intermediate data
- Example: local counting for Word Count:

```
def combiner(key, values):  
    output(key, sum(values))
```

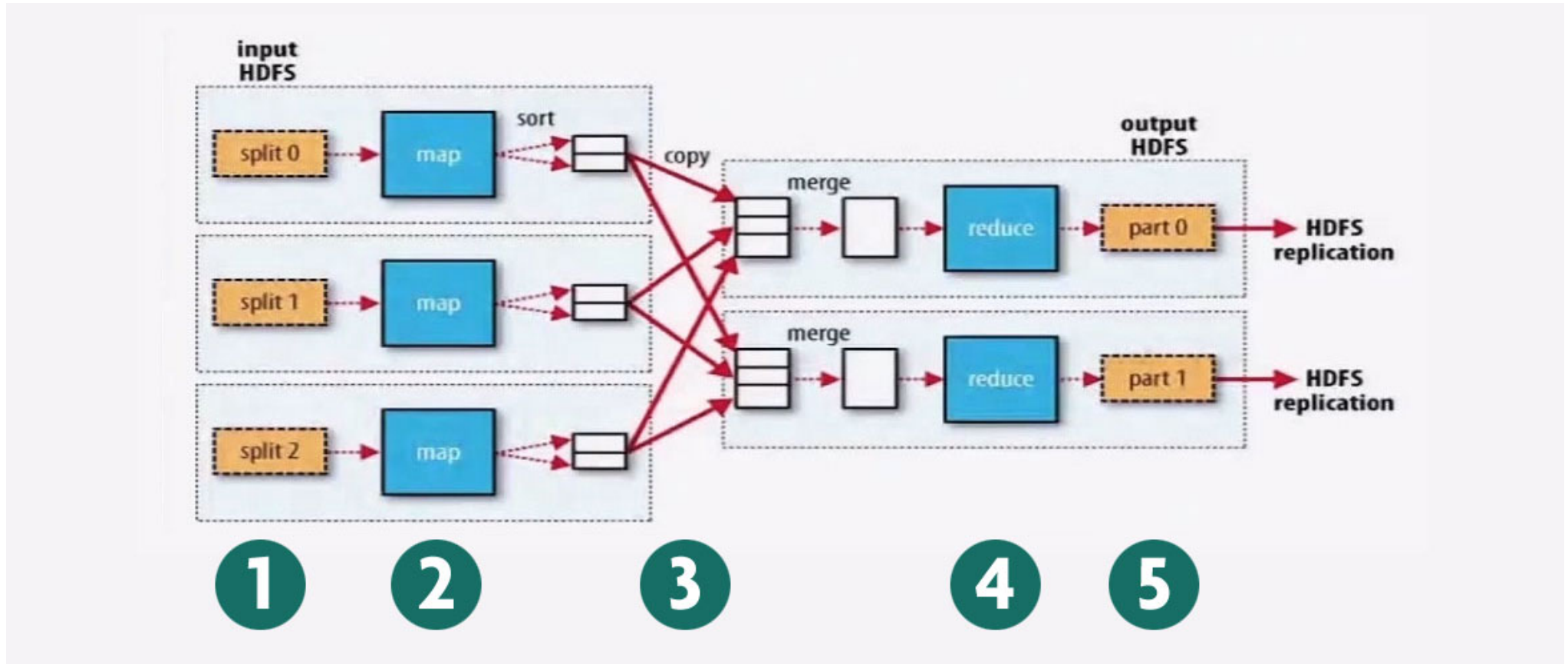
Word Count with Combiner



MapReduce behavior

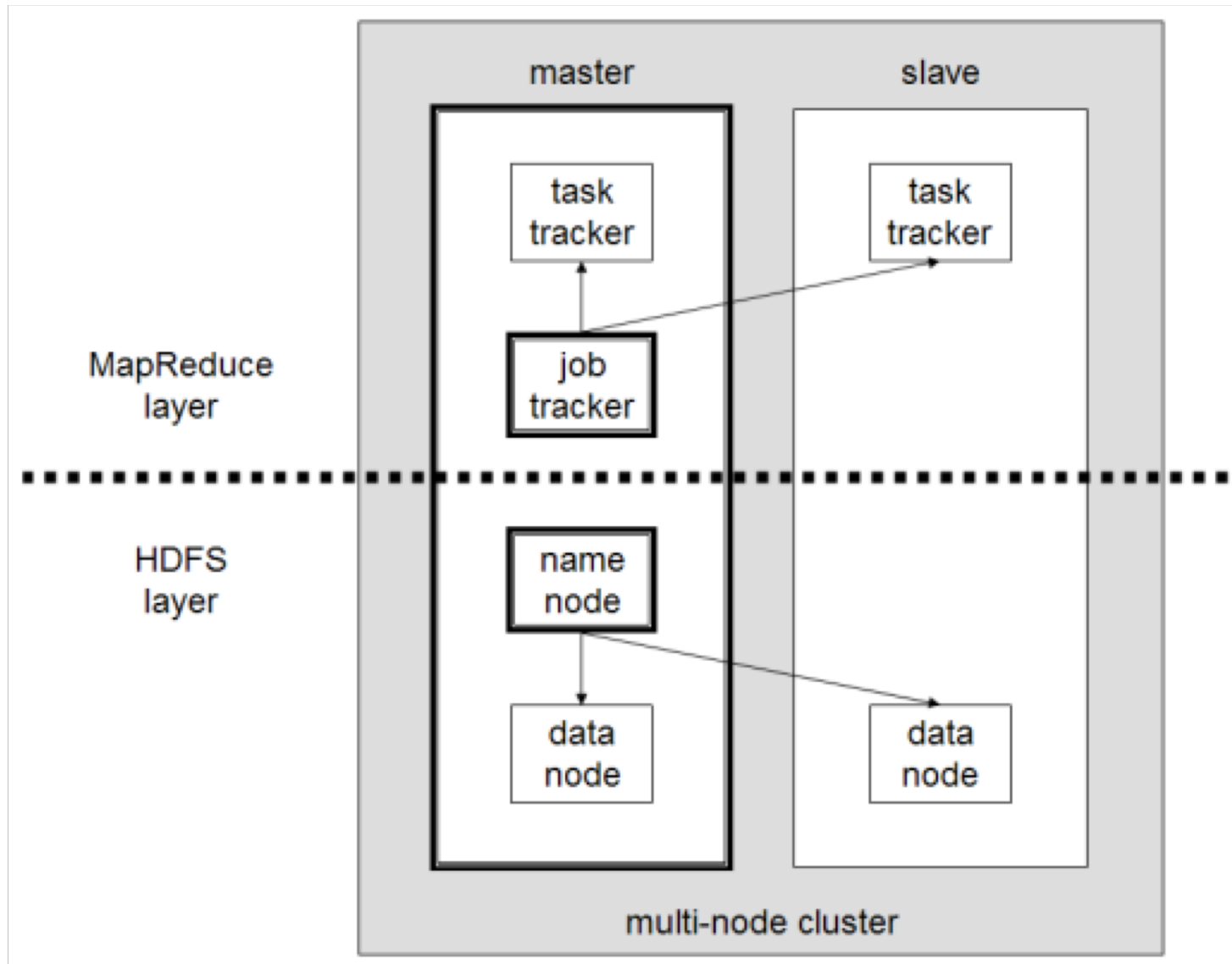


MapReduce behavior



blog.gopivotal.com/products/hadoop-101-programming-mapreduce-with-native-libraries-hive-pig-and-cascading

Hadoop and MapReduce



Execution details

- Mappers preferentially scheduled on the same node or the same rack as their input block
 - Minimize network use to improve performance
- Mappers save outputs to local disk before serving to reducers
 - Allows recovery if a reducer crashes
 - Allows running more reducers than # of nodes

Fault Tolerance in MapReduce

1. If a task crashes:

- Retry on another node
 - OK for a map because it had no dependencies
 - OK for reduce because map outputs are on disk
- If the same task repeatedly fails, fail the job or ignore that input block

➤ Note: For the fault tolerance to work, *user tasks must be deterministic and side-effect-free*

Fault Tolerance in MapReduce

2. If a node crashes:

- Relaunch its current tasks on other nodes
- Relaunch any maps the node previously ran
 - Necessary because their output files were lost along with the crashed node

Fault Tolerance in MapReduce

3. If a task is going slowly (straggler):

- Launch second copy of task on another node
- Take the output of whichever copy finishes first, and kill the other one

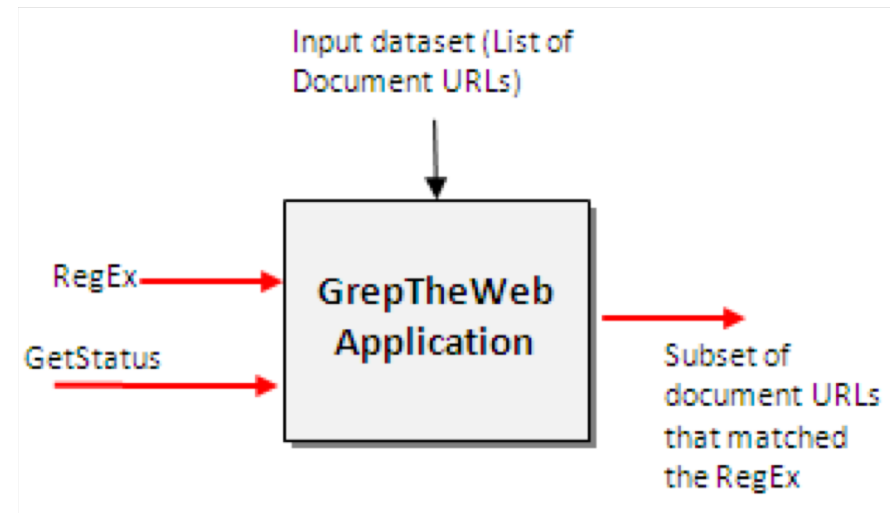
■ Critical for performance in large clusters (many possible causes of stragglers)

GrepTheWeb

The Alexa Web Search service allows developers to build customized search engines against the massive data that Alexa crawls every night

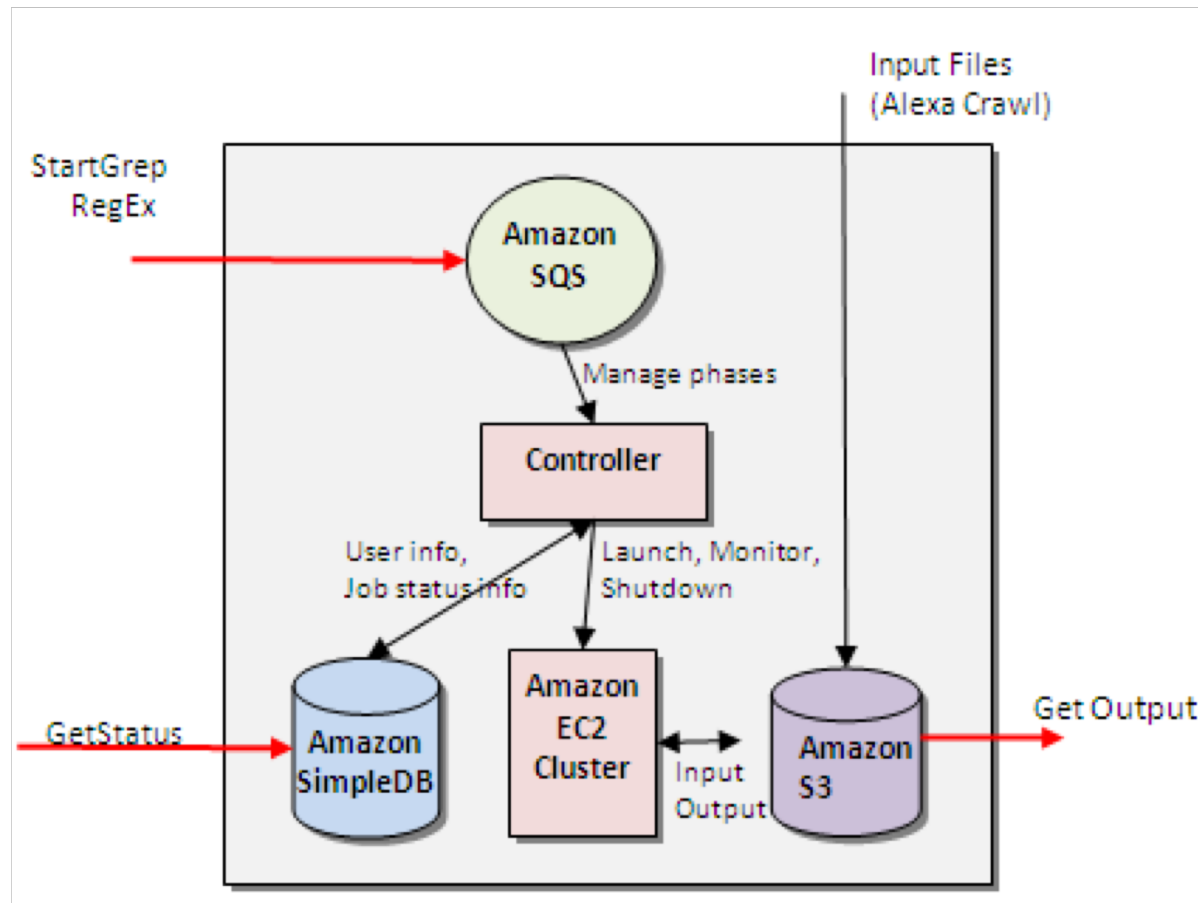
The web service allows users to query the Alexa search index and get results back as output

Developers can run queries that return up to 10 million results

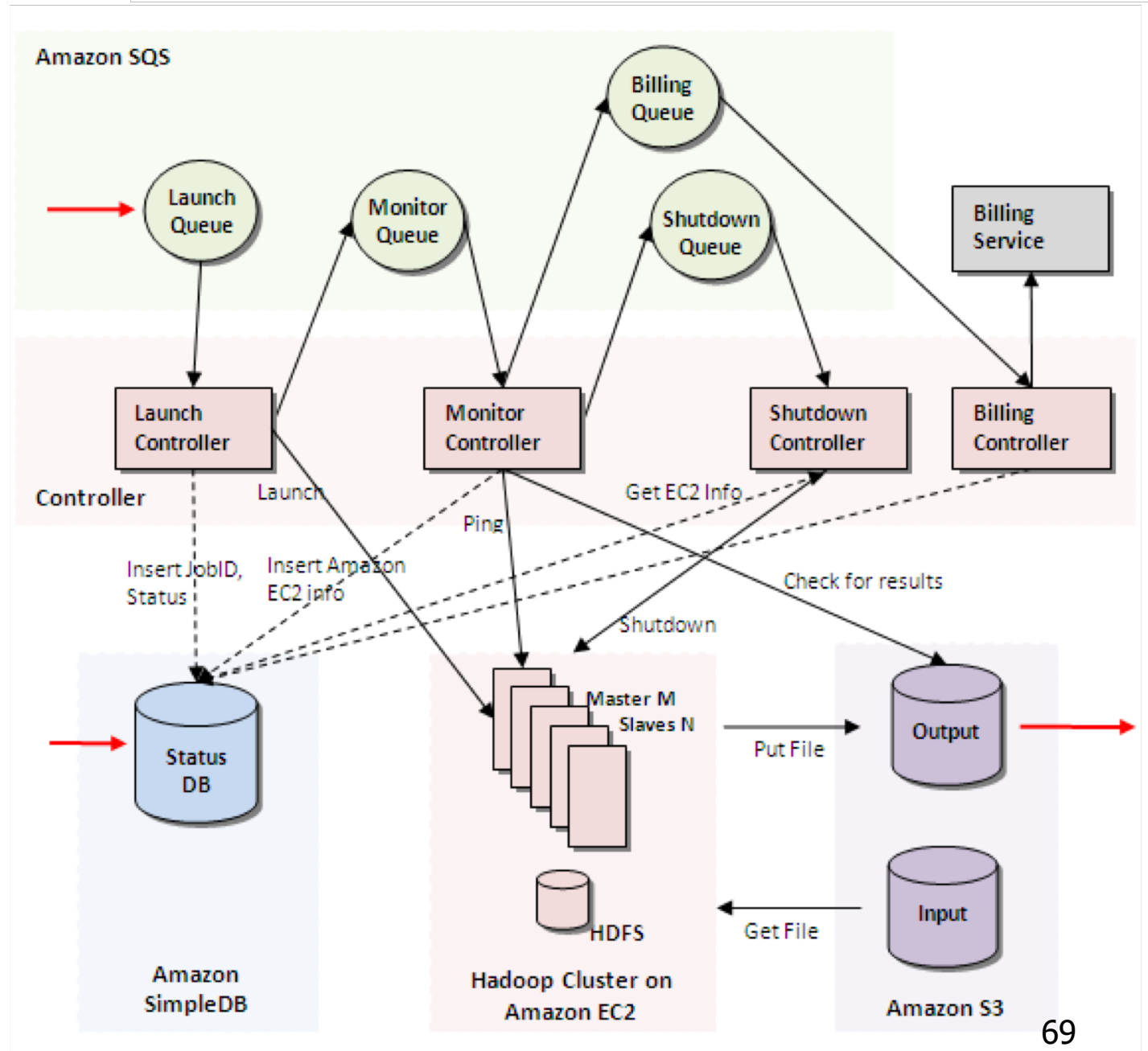
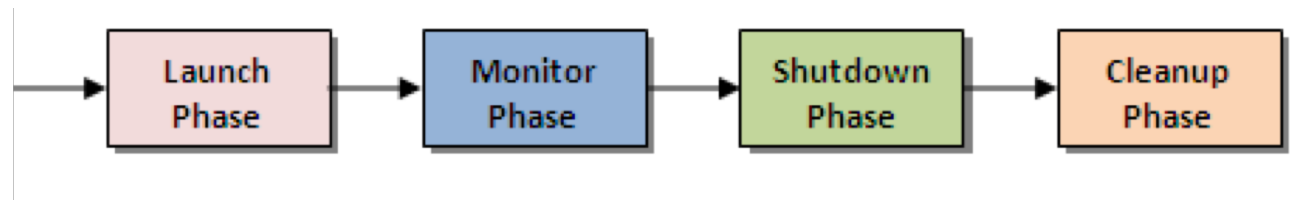


<http://aws.amazon.com/articles/Amazon-SQS/1632>

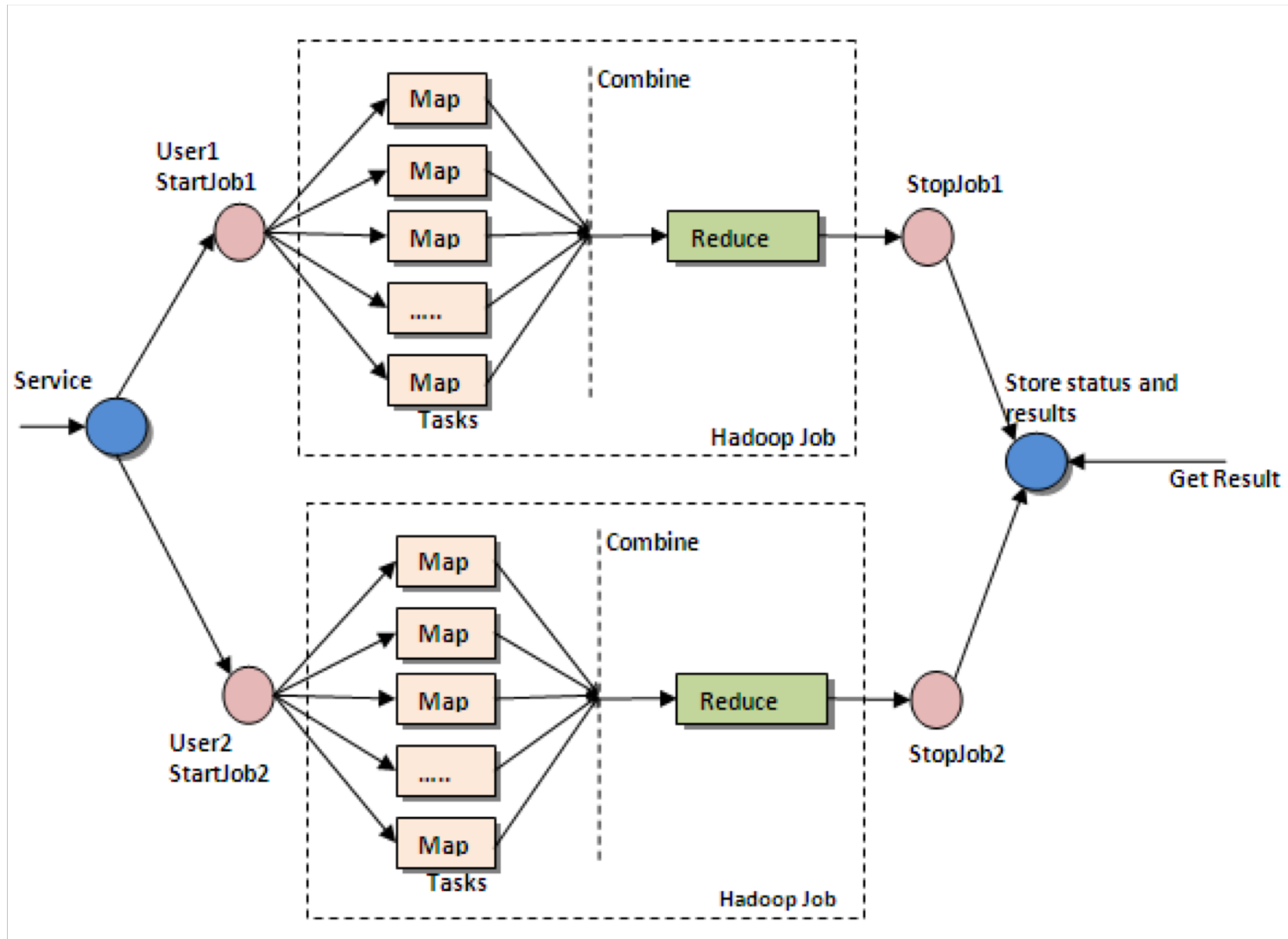
GrepTheWeb: zoom 1



GrepTheWeb: zoom 2



GrepTheWeb: mapreduce on hadoop



Apache Mesos

- Mesos is built using the same principles as the Linux kernel: it runs on every machine and provides applications (e.g., Hadoop, Spark, Kafka, Elastic Search) with API's for resource management and scheduling across entire datacenter and cloud environments.
- Scalability to 10,000s of nodes
- Fault-tolerant replicated master and slaves using ZooKeeper
- Support for Docker containers
- Native isolation between tasks with Linux Containers
- Multi-resource scheduling (memory, CPU, disk, and ports)
- Java, Python and C++ APIs for developing new parallel applications
- Web UI for viewing cluster state

Discussion

SOA vs Clouds

Horizontal (SOA) vs. vertical (Clouds) services

1. SOAs focus mainly on business
 - each service may represent one aspect of the business
2. Clouds layered according to software stacks
 - the lower services support the upper ones to deliver applications

SOA vs Clouds

1. SOA concerns the *application architecture*
 - Components designed for specific roles in a SOA application
 - Design starts with a business problem and then abstract out the services
 - Services can be re-used by other applications in the future
2. Clouds are for the *delivery of IT infrastructures*
 - Services based on their roles in a software stack, therefore mostly well defined
 - The cloud services are domain- or problem- independent
 - The cloud services can be easily re-used by any application

SOA vs Clouds

1. SOA is mainly for enterprise computing
 - Cloud computing is internet-based services
2. SOA provides services of computing
 - Cloud computing provides computing of services
3. SOA applies the service principle to application software
 - Cloud applies the service principle at all levels: infrastructure, platform, and application level
4. Cloud is in many ways beyond SOA: implement Cloud, also get to keep SOA

Common issues

- Network dependence
 - Same structural weakness when the network is not performing or is unavailable
- Producer/ consumer model
- Share concepts of service orientation
- Services can be shared by multiple applications/users
- Demand minimum coupling among different parts of the system
 - any change on one part of the system has limited impact on the overall system
- SOA and clouds complement each other

Summary

- SOA define the Web as a powerful application and legacy integrator
- The standards to support SOA based on WebServices - such as XML, WSDL, and SOAP - are stable
- However, technologies to support quality attributes of SOA-WS, such as security, transaction efficiency, and availability are still evolving
- Clouds foster novel ecosystems: they require changes to the development, deployment, and operational processes (everything-as-a-service)
- Novel interesting research problems

Self test

- What is a hybrid cloud architecture?
- In which ways are clouds scalable?
- Which are the main components and connectors of clouds?
- Compare a SOA with a cloud architectural stack
- Which architectural issues and patterns are typical of cloud computing systems?

Readings

- Erl & Cope, *Cloud computing design patterns*, Prentice Hall 2015
- Bergmayr, *An Architecture Style for Cloud Application Modeling*, PhD th, TUWien 2016
- Dean & Ghemawat, MapReduce: simplified data processing on large clusters, *CACM*, 51:1(107-113), 2008
- Miner & Shook, *MapReduce design patterns*, O'Reilly 2013
- Nurmi, The Eucalyptus Open-Source Cloud-Computing System, 9th IEEE/ACM Int. Symp. on Cluster Computing and the Grid, 2009
- Rochwerger, The Reservoir model and architecture for open federated cloud computing, *IBM Journal of Research and Development*. 53:4(1-11), 2009

Relevant sites

- `cloudcomputingpatterns.org`
- `docs.microsoft.com/en-us/azure/architecture/patterns/`
- `aws.typepad.com/`
- `developers.google.com/appengine/`
- `www.windowsazure.com`
- `www.armandofox.com/geek/teaching/`
- `radlab.cs.berkeley.edu`

Journals and conferences

- IEEE Transactions on Cloud Computing
- Journal of Cloud Computing: Advances, Systems and Applications
- IEEE/ACM Int. Conf. on Utility and Cloud Computing
- IEEE Int. Conf. on Cloud Computing
- ACM Symp. on Cloud Computing
- ACM Cloud and Autonomic computing Conf.

Questions?

