

Cloud and Web Applications

- 1. Introduction**
- 2. Client-Side Techniques**
- 3. Server-Side Techniques**
- 4. Lab: Web App for Cloud**

Objectives

- Understand the basic concepts and techniques of web applications
- Understand the basic ideas of cloud computing and the deployment of web applications on cloud
- Getting your hands dirty on web applications and cloud

Cloud and Web Applications

Part 1: Introduction

Table of Contents

- Web Applications
- History of Web Applications
- Cloud Computing
- OpenShift Online

WEB APPLICATIONS

What is a Web Application?

- “*A web application is a software package that can be accessed through the web browser. The software and database reside on a central server rather than being installed on the desktop system and is accessed over a network.*” (NetSity corporate homepage)
- A distributed application that accomplishes a certain business need based on the technologies of WWW and that consists of a set of web-specific resources
- Software services delivered over Internet standard technologies

Web Applications

- Complex distributed, client/server applications
- High interactivity, high accessibility (Cloud)
- Applications are usually broken into logical chunks called "tiers", where every tier is assigned a role
- Client side run inside a web browser
- Using HTTP for communication
- Rapid development, requires more planning, design, and control than “conventional” projects.

What are the Advantages?

- App runs server side, no install, packaging, CDs, upgrades, configurations or tweaking of settings on the client side.
- Greater responsibilities and control placed in the hands of the system administrators (as opposed to the users)
- Data is likely more secure (stored server side, proper security measures and backup)

What are the Advantages? (con't)

- Machine independent (any user can log in from any computer). Lower client side system requirements (machine only needs network access and the ability to run a compliant web browser)
- One application will run on any and all platforms, assuming standards compliant code and browsers.
- Reduced external network traffic (ex. Database heavy applications)

What are the Disadvantages?

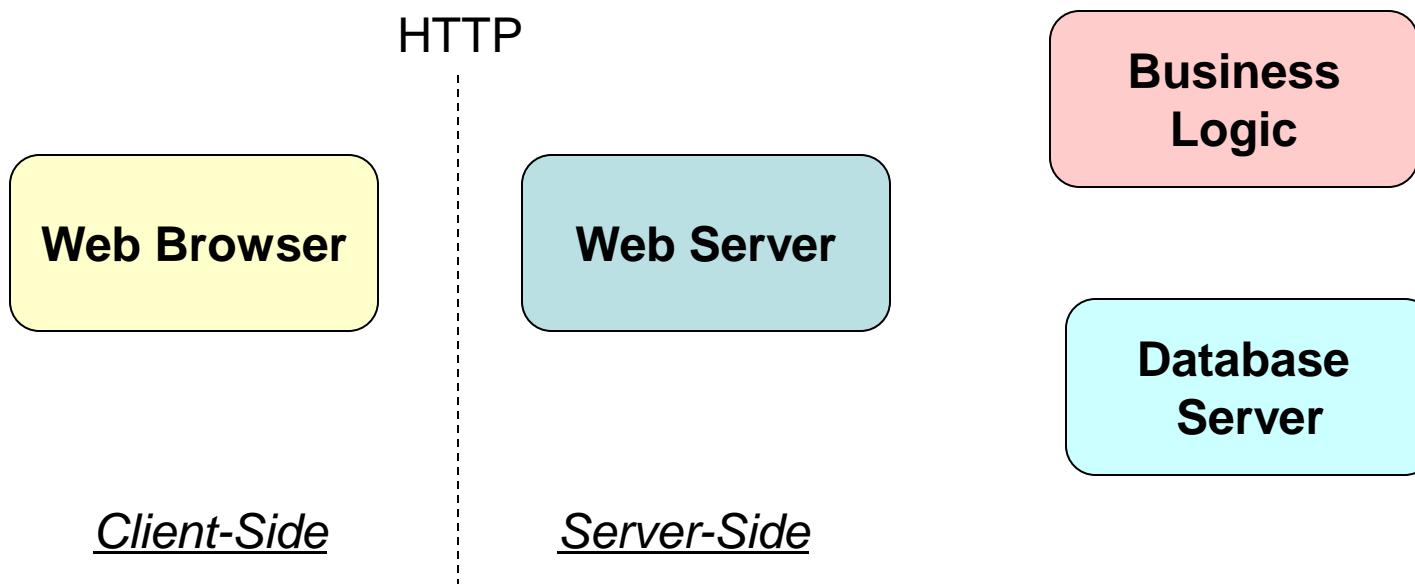
- “Who has my data?” Essentially, not you.
- Issues of trust. Many users do not trust other people, even within the same company, to keep their data safe and secure.
- Response time. While the actual execution of the app may be much quicker, user response time can be noticeably slower than a local app.
- Internet (or network) connectivity is not (yet) ubiquitous.

What are the Disadvantages? (2)

- Browser compatibility can still be a problem.
- Some tasks that are simple in traditional application development, are quite complicated from a web application (ex, local printing)
- Security concerns limit what you can accomplish (limited access to the users local machine).
- Depending on the application, usability can be very bandwidth sensitive.

Web Application Components

- Four important components of a web application:



Web Application Components (2)

- **Web Browser**: presents the user interface
- **Web Server**: processes HTTP requests
- **Business Logic**: processes requests at the application level by providing a service
- **Database Server**: maintains the database by processing query and update requests from the application

What are the Sides?

- Client side (Front-end):
 - runs after page is displayed
 - page/content generation
 - user interaction
 - send data back to server
 - depends on browser/DOM
- Server side (Back-end):
 - runs before page is displayed
 - page/content generation
 - handle returning data
 - concerned with web server/database

Web Browsers

- Program designed to enable users to access, retrieve and view documents and other resources on the Internet
- Main responsibilities:
 - Bring information resources to the user (issuing requests to the web server and handling any results generated by the request)
 - Presenting web content (render HTML, CSS, JS)
 - Capable of executing web applications.

Different Browsers

Browsers driven by product differentiation



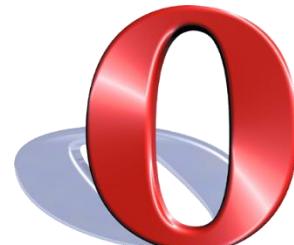
IE



Firefox



Chrome

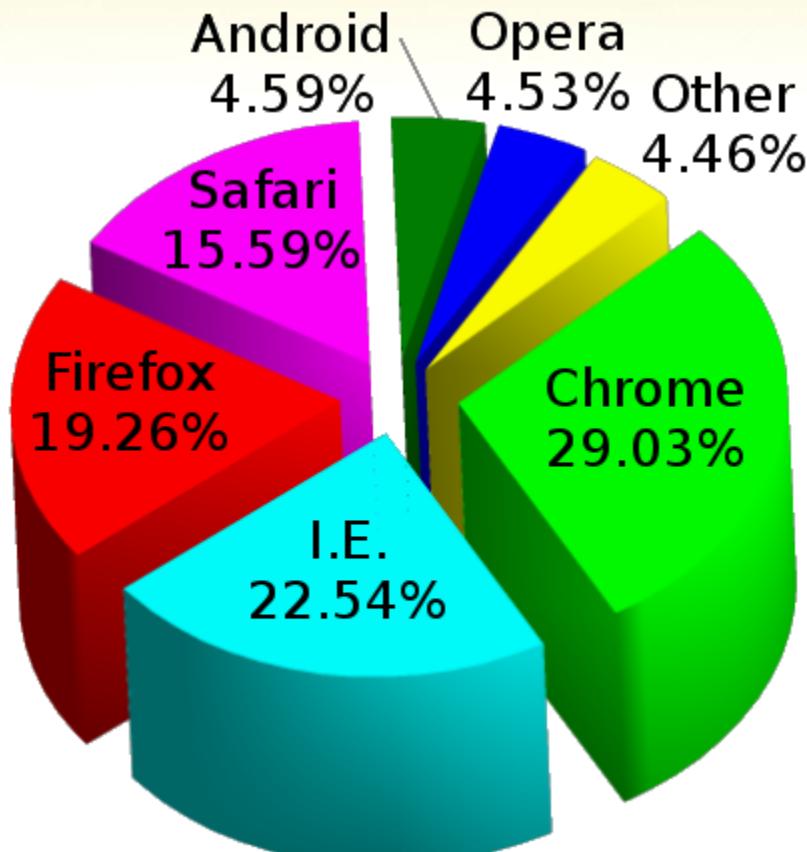


Opera



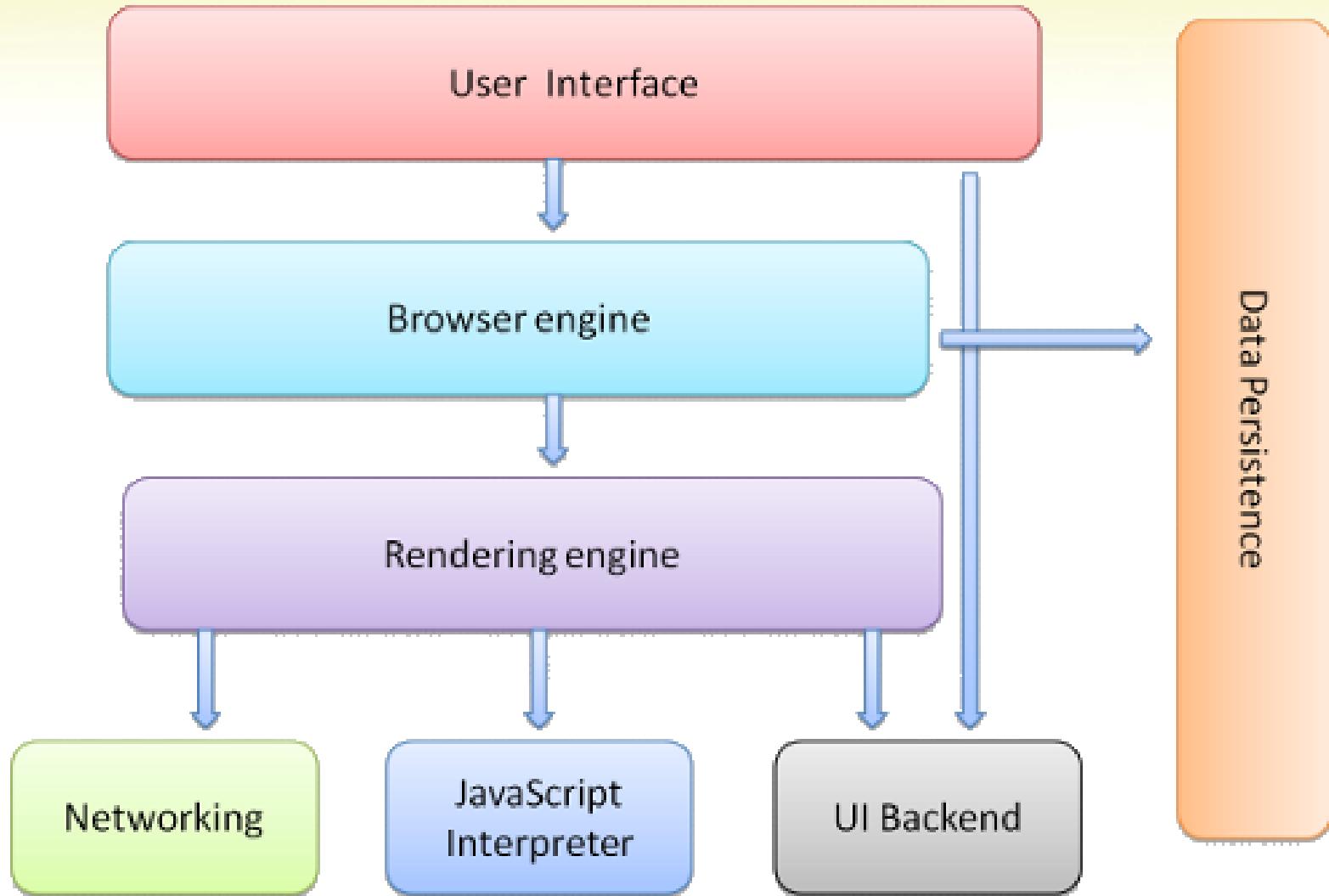
Safari

Browser's Market

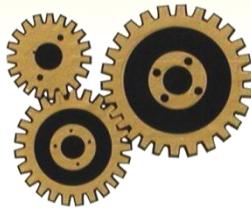


Browser usage on Wikimedia
September 2012

Browser Structure



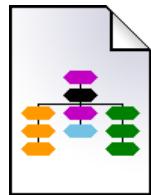
Browsers components



Scripting engine:
interprets JavaScript



Rendering engine:
draws text, images, etc



DOM:
Document Object Model

Rendering engines

- Trident-based
 - Internet Explorer, Netscape, Maxthon, etc.
- Gecko-based
 - Firefox, Netscape, SeaMonkey, etc.
- WebKit-based
 - Chrome, Safari, Maxthon, etc.
- Presto-based
 - Opera



Inside Browsers

- Separation of concerns:

- structure (.html)
- presentation (.css)
- logic (.js)
- data



User Interface Presentation

- Parse HTML and CSS code
 - handle errors
- Format and present a graphical display
- Handle user interactions
 - scroll, mouse movement, click, etc.

Script Interpretation

- Most browsers interpret JavaScript and its variants (ECMAScript, JScript, etc.)
- Scripting languages are powerful, so interpreters are necessarily complex
- Script interpreters are not entirely standardized across browsers, so script programmers must test scripts on many browser versions
 - The “write once, test many” principle in action

What is a server?

Software that provides services:

- Web server
- Database server
- File server
- Print server
- Mail server



Web Server Responsibilities

- Web servers are software products that handle web requests and manage connections
- These requests are redirected to other software products (ASP.NET, PHP, etc.), depending on the web server settings

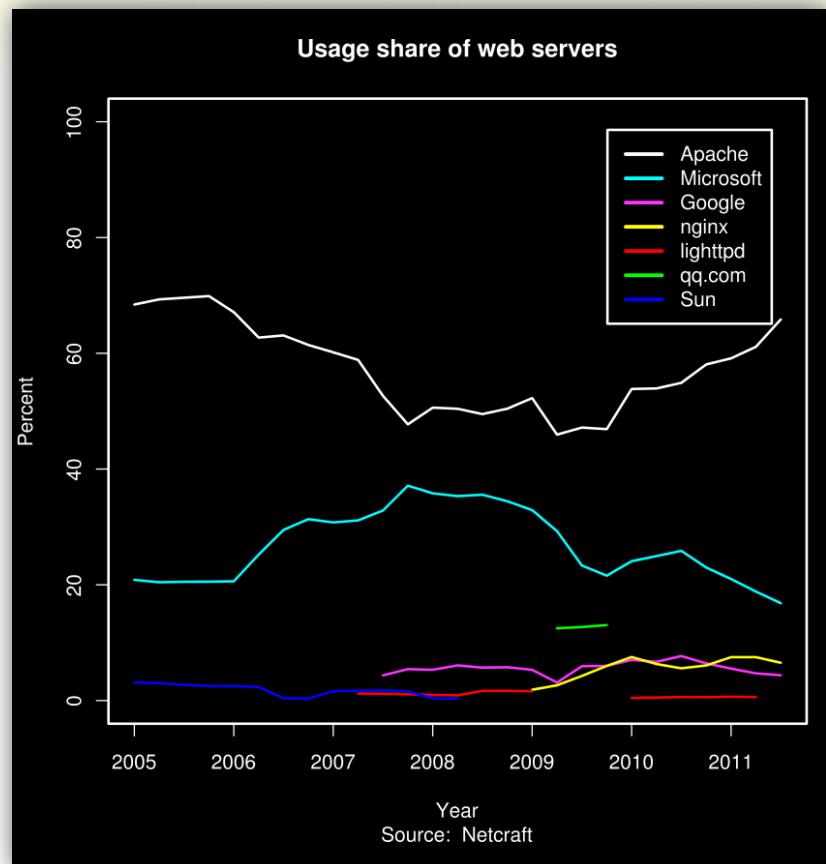
Web Servers

Apache, IIS, Nginx, Lighttpd, etc.



Web Servers Market Share 2016

- Apache - 52.3%
- Nginx – 30.4%
- IIS - 12.0%
- LiteSpeed -2.3%
- Google Server -1.3%
- Tomcat – 0.5%
- Lighttpd – 0.1%
- Node.js – 0.2%
- Others

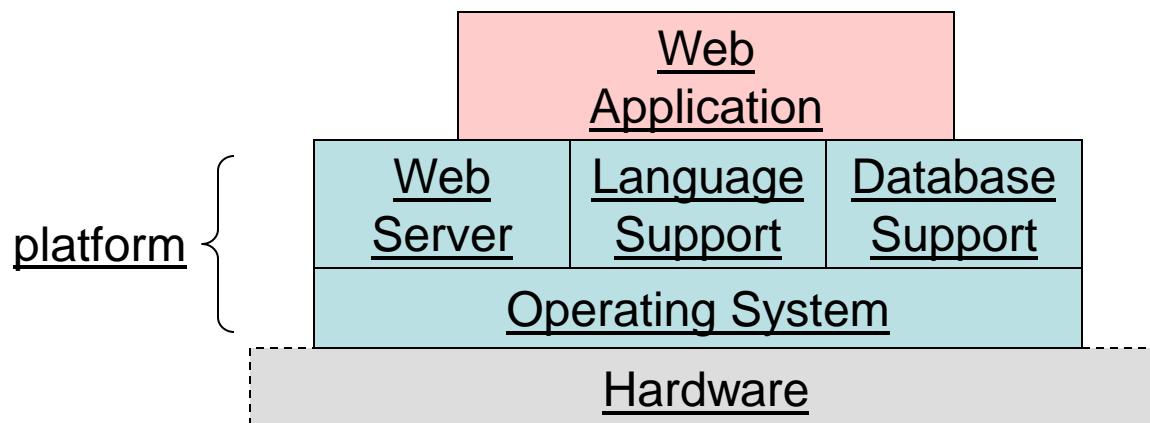


Database Server Responsibilities

- Relational Database Management System (RDBMS)
NoSQL
- Maintains data storage for applications
 - processes queries and updates
- Provides a standard interface for application programs
 - Open DataBase Connectivity (ODBC)
 - Java DataBase Connectivity (JDBC)
- Supports standard query language for data query and manipulation
 - Structured Query Language (SQL)

Web App Platform

- A webapp **platform** is the host environment for application development and operation
- The platform includes
 - operating system, web server, language support, database support

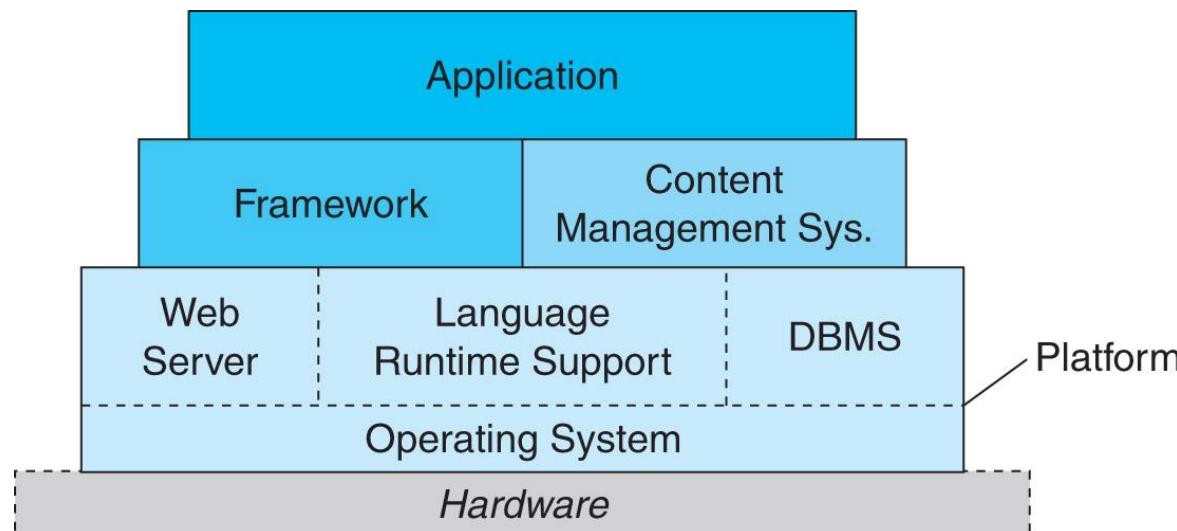


Web App Framework

- A web app **framework** is a set of tools that support web app development with:
 - A standard design model (e.g., MVC)
 - User interface toolkit
 - Reusable components for common functions (authentication, e-commerce, etc.)
 - Database support
 - Support for distributed system integration

Role of Web App Framework

- Frameworks give application developers more powerful building blocks to work with



HISTORY OF WEB APPLICATIONS

The Good Old Days

- Simple, static web pages (with animated gifs).

The screenshot shows the Apple website homepage from July 14, 1997. The header features the Apple logo and the text "Welcome to Apple". A banner in the center promotes "Introducing CyberDrive" and encourages users to "Register today for a free CD-ROM." Below the banner is a "BMW" logo. To the right, there are sections for "iMATE 300" (described as "Mobile, Affordable, & Smart") and "MOVIES FROM MARS" (featuring "QuickTime VR Takes You Out of this World"). On the left, a sidebar includes links for "Find It", "Product Information", "Customer Support", "Technology & Research", "Developer World", "Groups & Interests", "Resources Online", and "About Apple". A dropdown menu for "Apple Sites Worldwide" lists "Switzerland", "Taiwan", "Turkey", "UK & Ireland", and "United States", with "United States" currently selected. A "What's Hot" section at the bottom highlights "Preorder Mac OS 8" and "Be the First to Know".

JULY 14
1997

Welcome to Apple

Introducing CyberDrive

BMW

Register today for a free CD-ROM.

iMATE 300
Mobile, Affordable, & Smart

MOVIES FROM MARS
QuickTime VR Takes You Out of this World

What's Hot

Preorder Mac OS 8

Now you can [preorder Mac OS 8](#), described by Macworld as "the most comprehensive update to the Mac OS in years, sporting a bold new

Be the First to Know

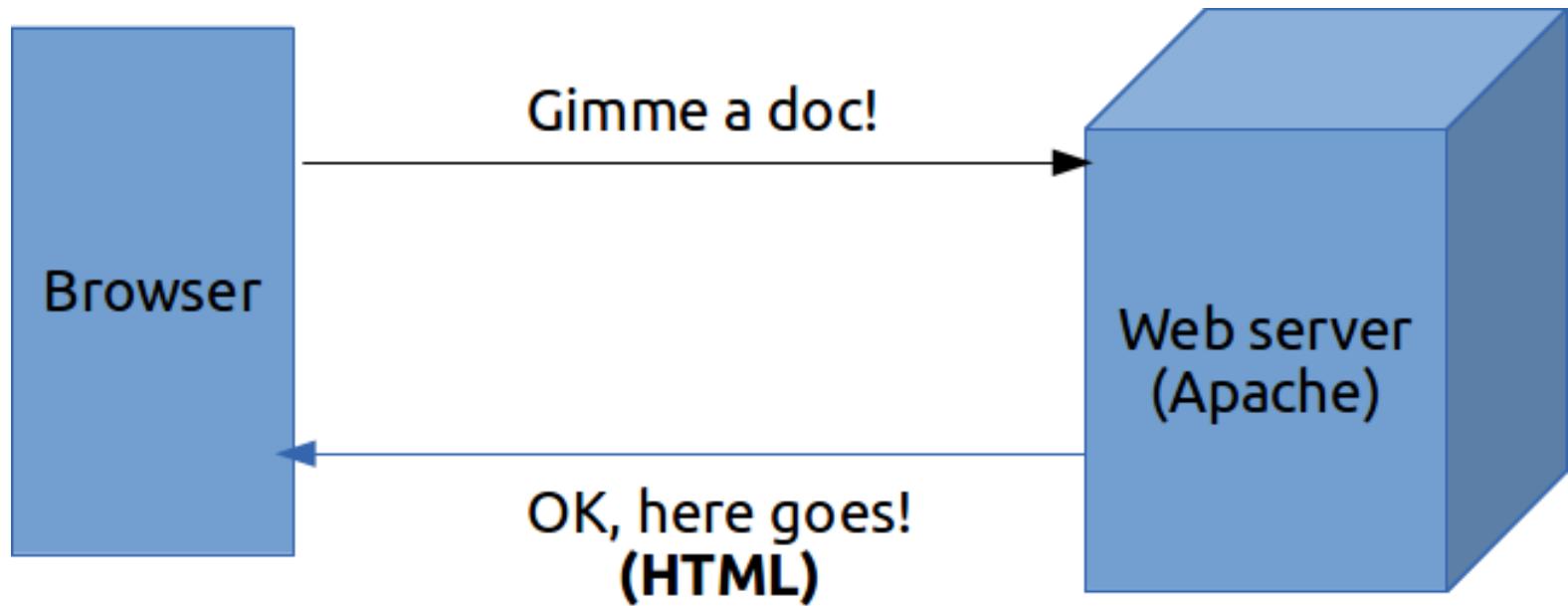
Learn about new Macintosh software releases the moment they become available. Check [Hot Mac Products](#) to hear about programs like Speed

Apple Sites Worldwide

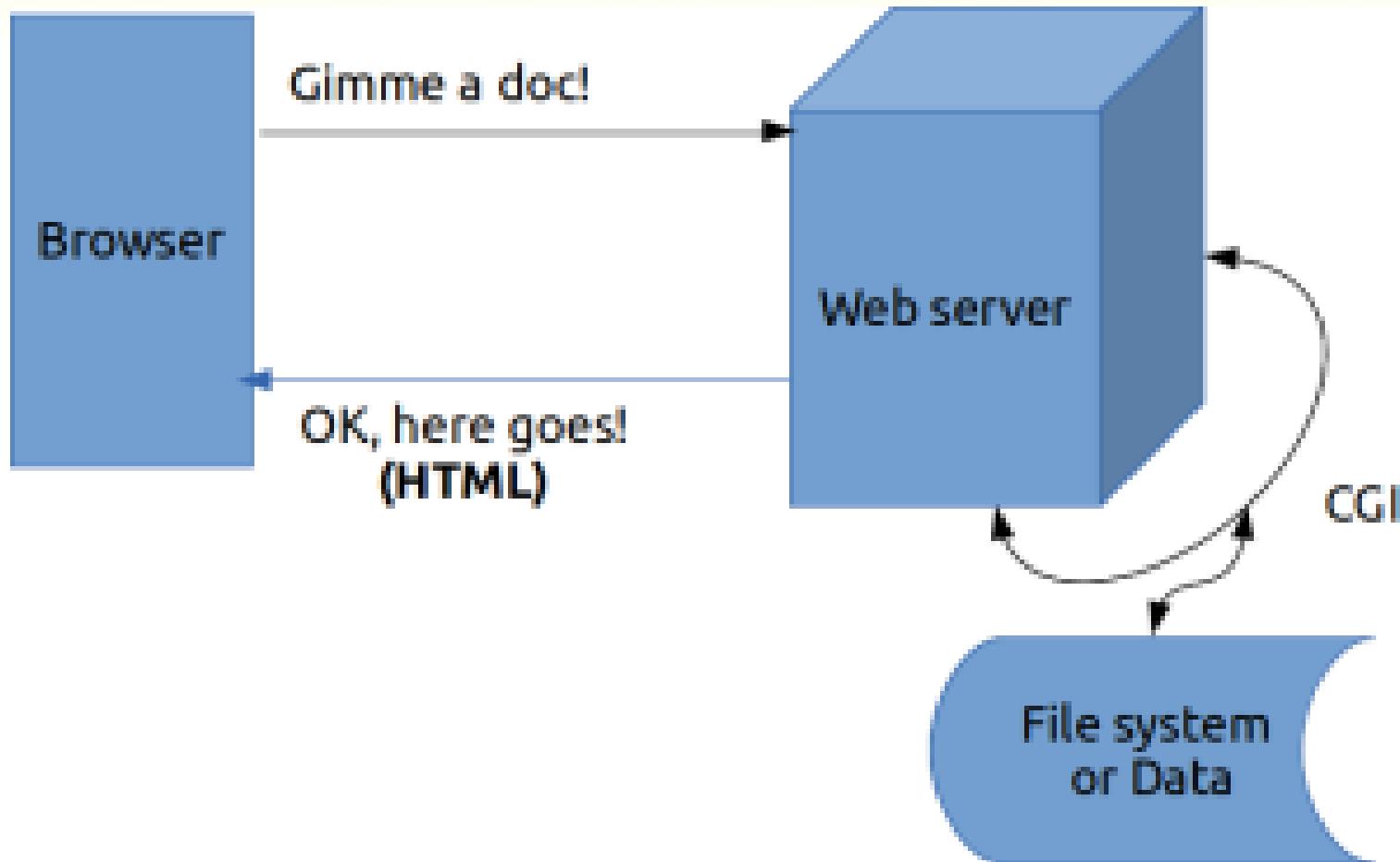
Switzerland
Taiwan
Turkey
UK & Ireland
United States

Go

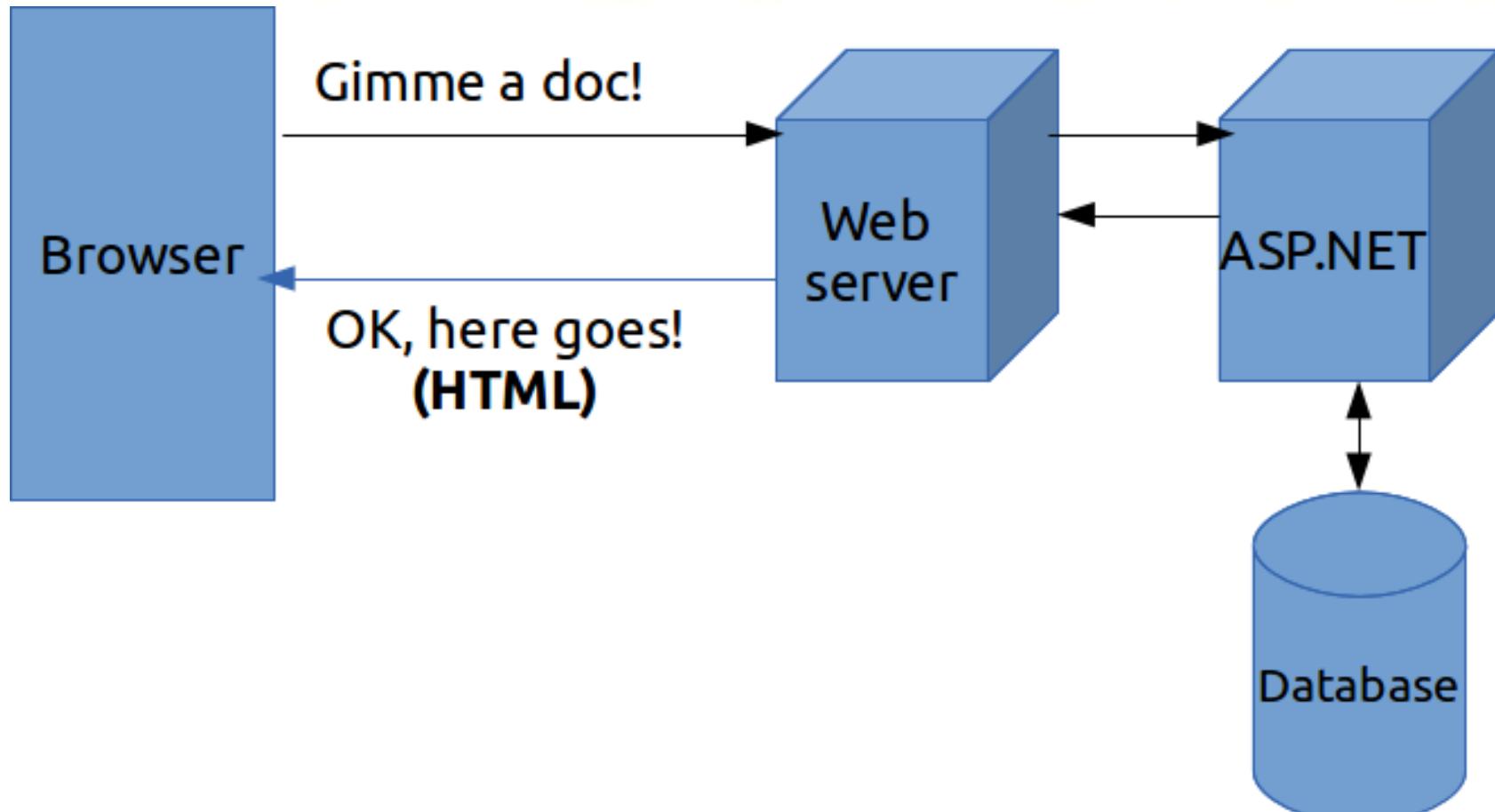
Static Web Site



CGI/Perl → till 2005

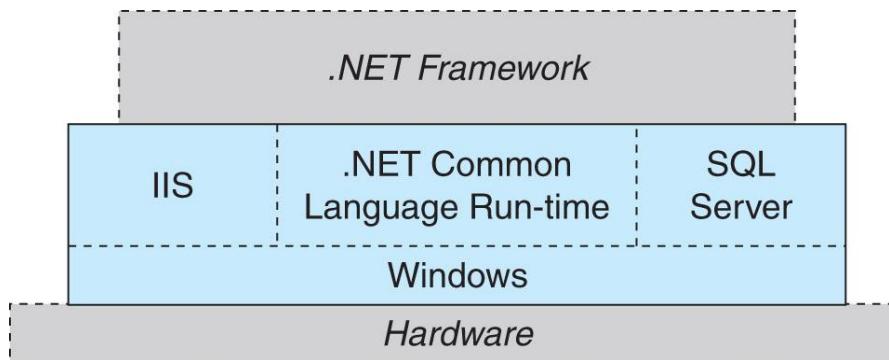


IIS & ASP



Microsoft / .NET

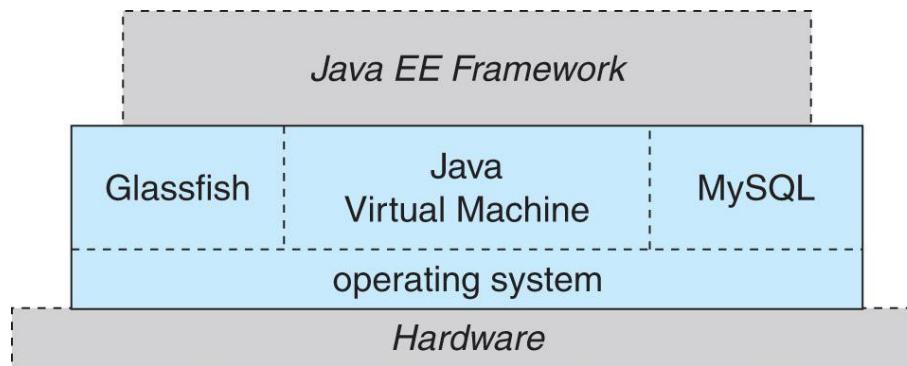
- All Microsoft products (licensed)



- .NET supports multiple languages
- Runs primarily on Windows Server O/S

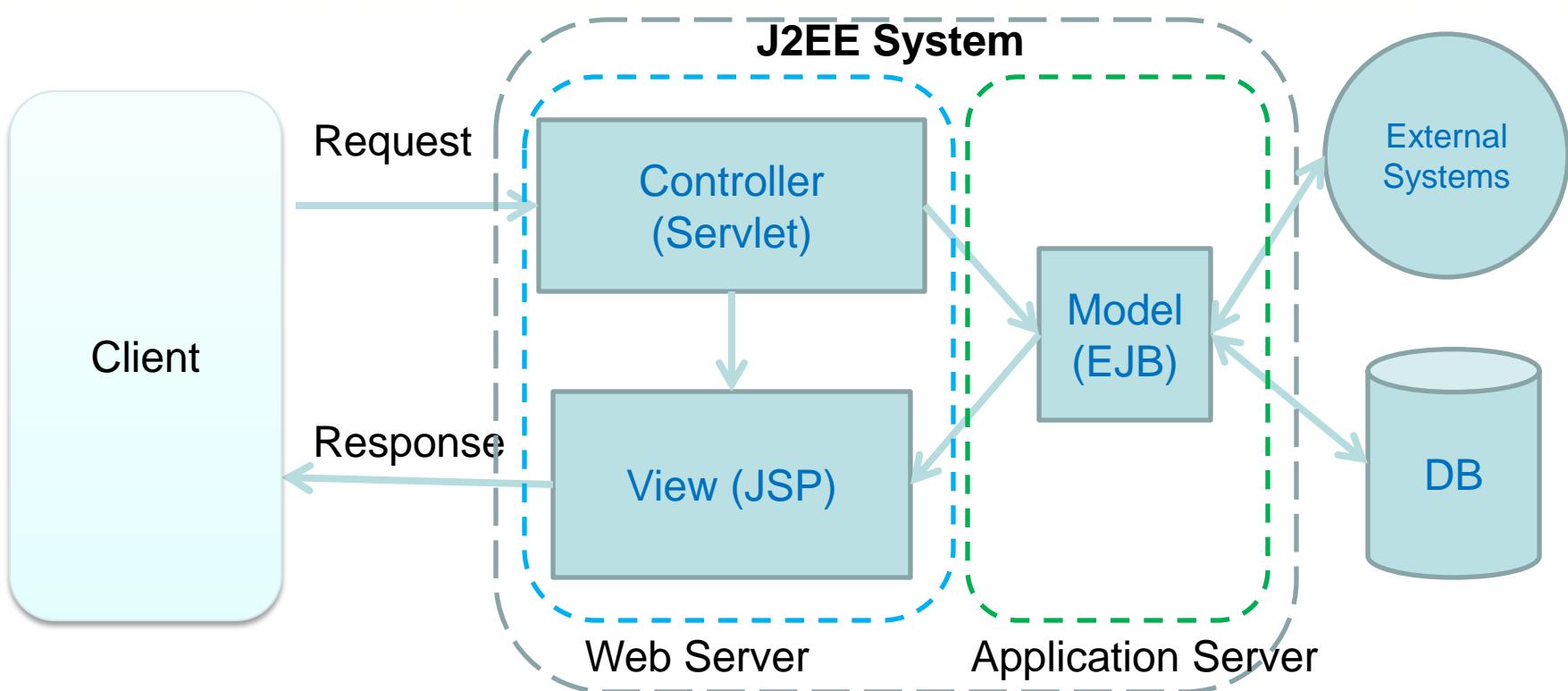
Java EE(J2EE)

- Supports Java language development



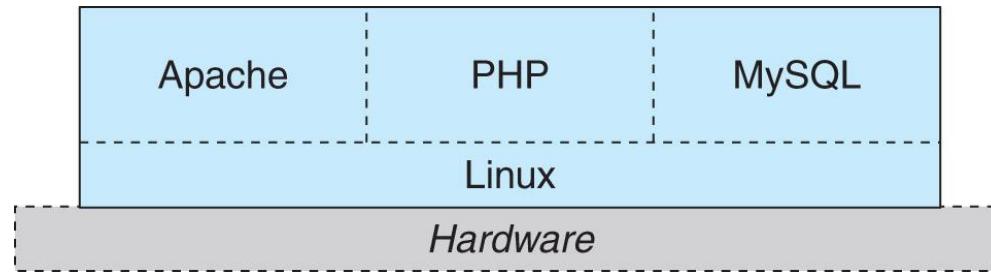
- Supported by multiple operating systems
- Proprietary, free license

J2EE Technology Overview



LAMP Stack

- Linux, Apache, MySQL, PHP/Perl/Python

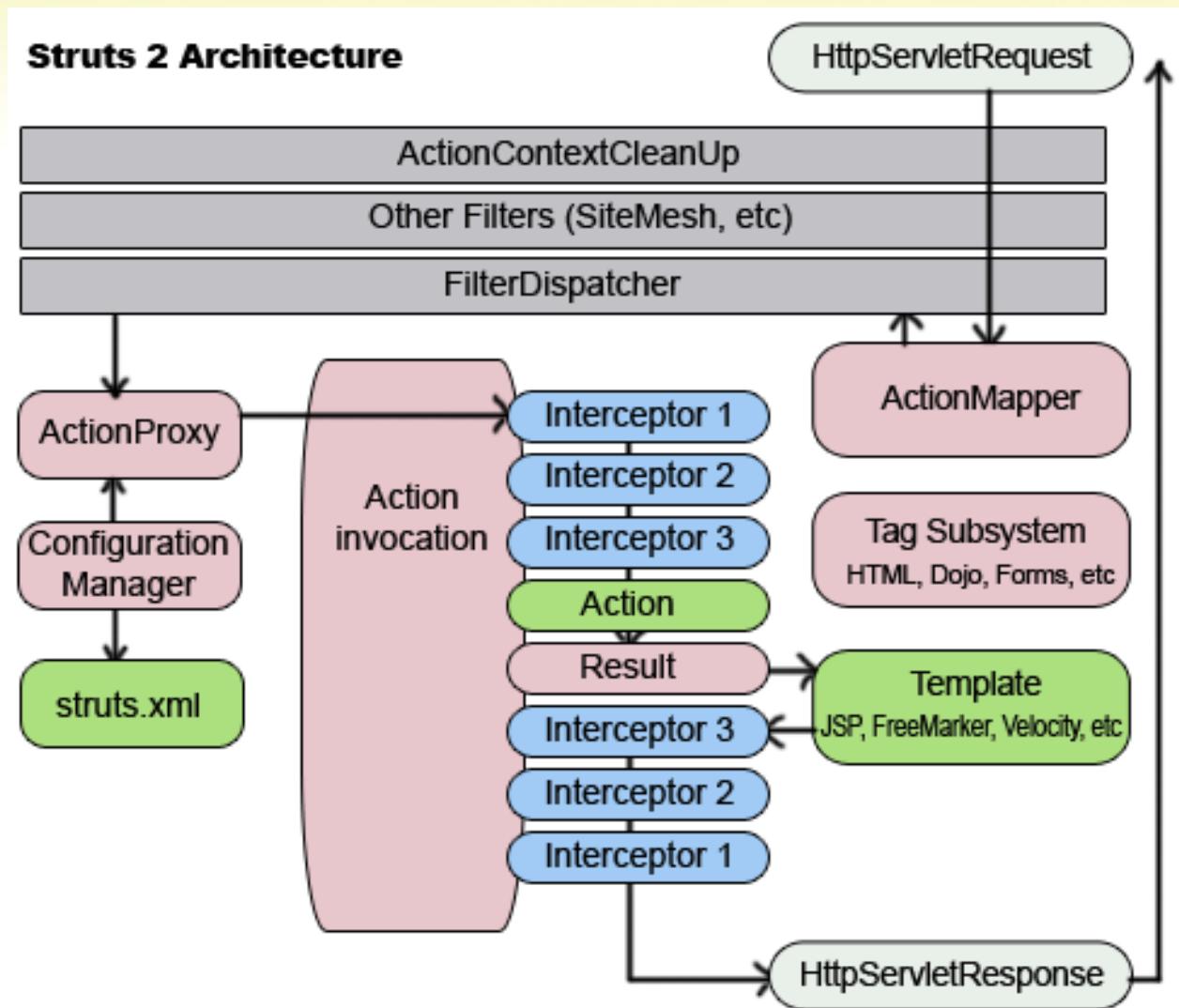


- The LAMP platform appeared in mid 1990's and has become very popular
- LAMP is open-source free software, which is one reason for its popularity

Struts

- A webapp framework based on Java EE
- Features:
 - use of MVC design paradigm
 - Centralized XML-based application configuration that can define many functions
 - Action definitions link user interface events to Controller and View modules

Struts Architecture



Servlet Filters



Struts Core



Interceptors

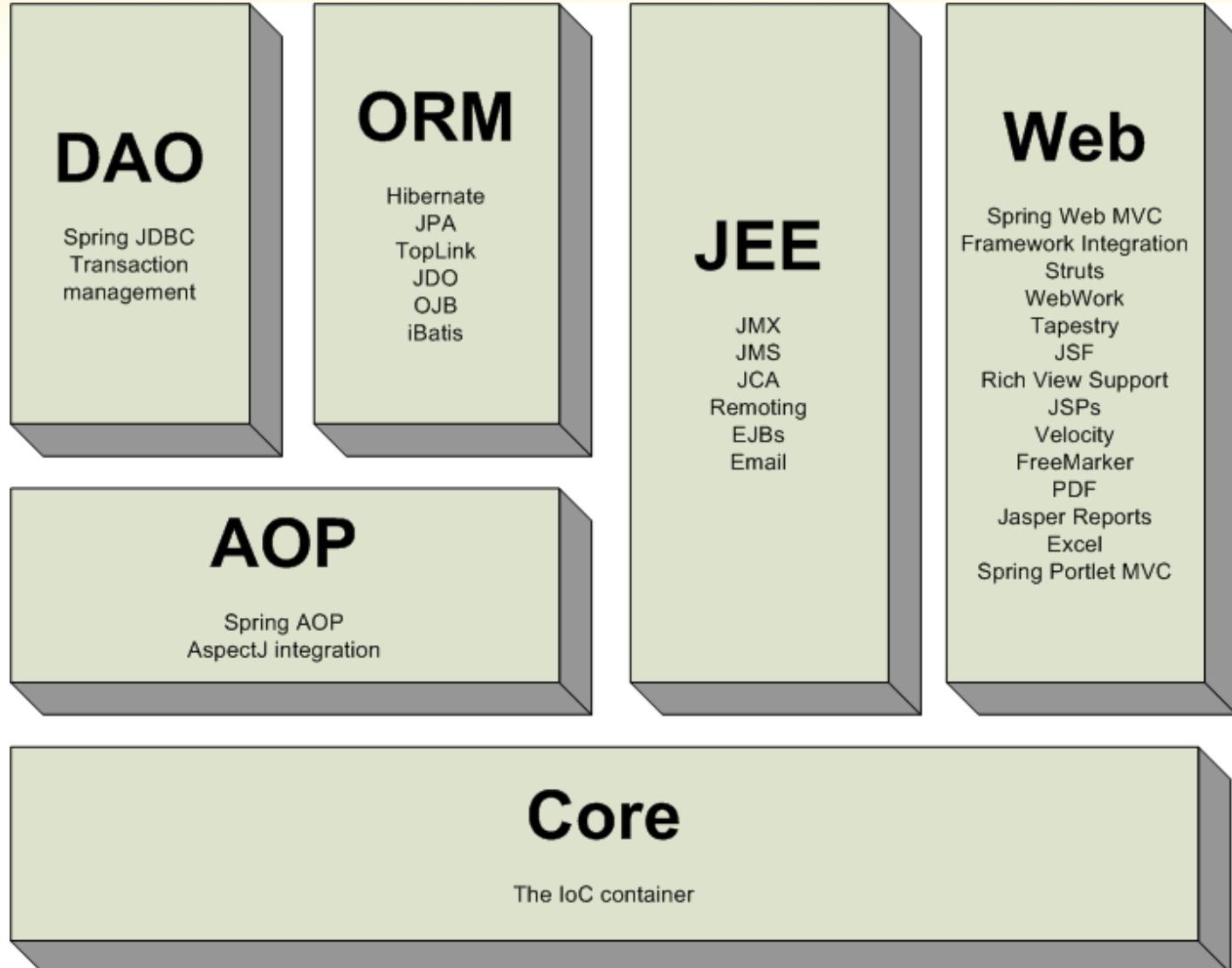


User Created

Spring

- Lightweight container and framework
 - Most of your code will be unaware of the Spring framework
 - Use only the parts you want of Spring you want
- Manages dependencies between your objects
 - Encourages use of interfaces
 - Lessens “coupling” between objects
- Cleaner separation of responsibilities
 - Put logic that applies to many objects in one single place
 - Separate the class’s core responsibility from other duties
- Simplifies database integration
 - Spring JDBC
 - Hibernate
 - Java Persistence

Spring Architecture



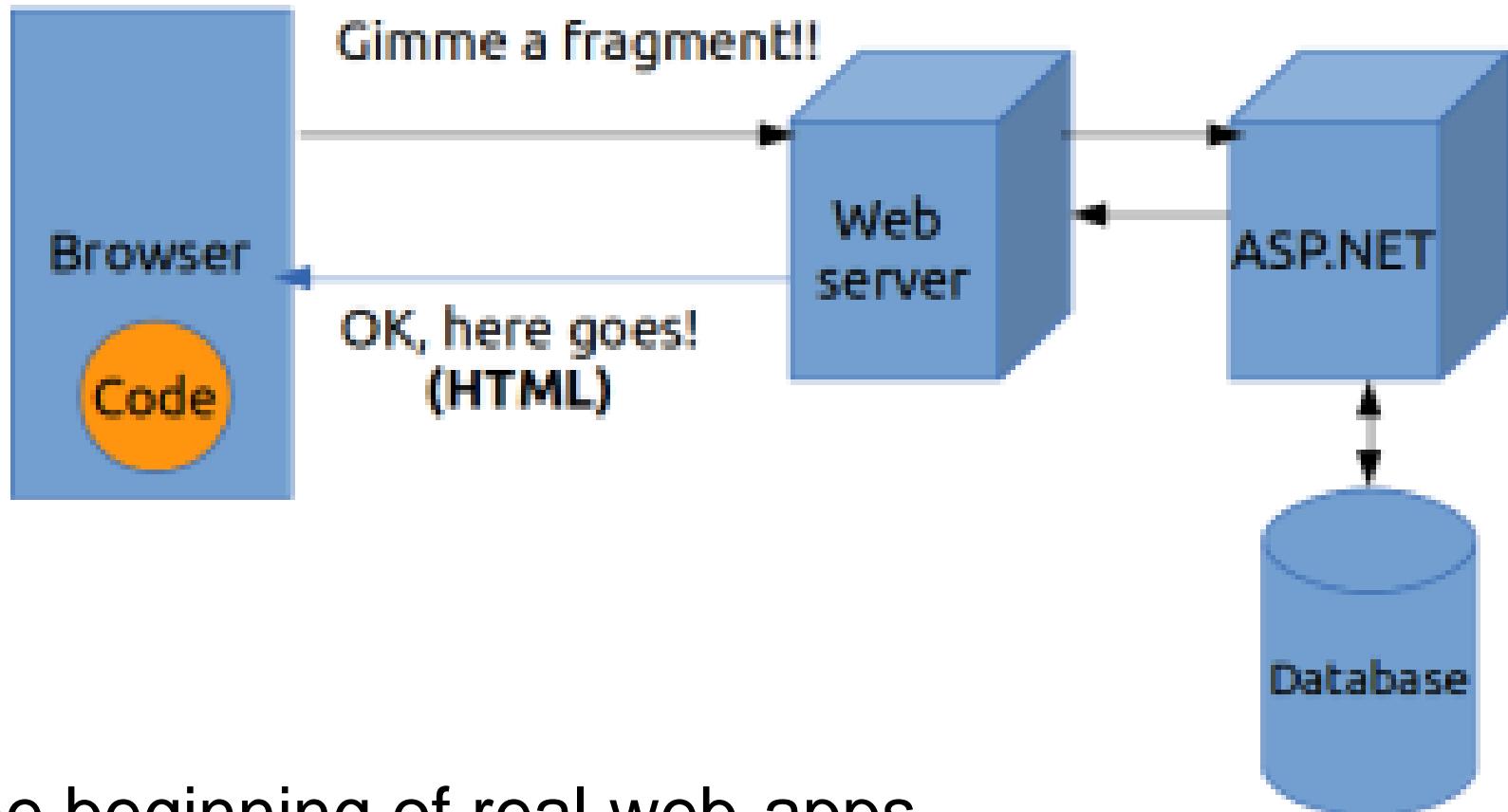
Hibernate

- A ORM (Object/Relation Mapping) framework
- Provides a high-performance Object/Relational persistence and query service
- Traditional (historical) use
 - Mapping Java objects to relational databases
- Today
 - Collection of projects/frameworks for extended use of POJO (plain old Java objects)
- <http://www.hibernate.org/>

Why Use Hibernate?

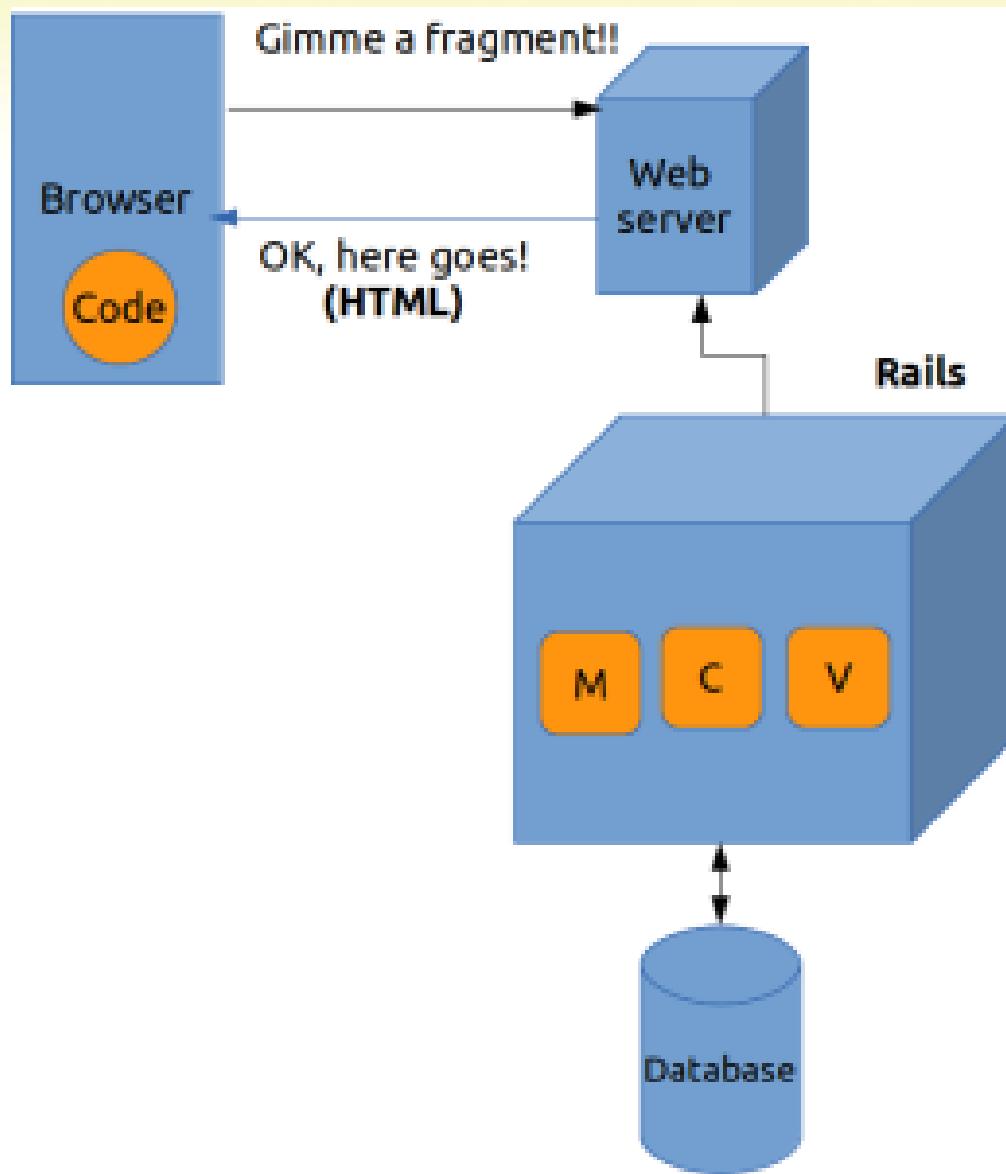
- Simpler data persistence
 - Automatically handles mapping SQL to Object and vice versa
 - Automatic creation of database schemas
 - Automatic updating of database schemas
 - Add a field to an object; Hibernate converts your existing database for you.
 - Provides search functionality
- Simpler database management
 - No JDBC code or SQL code needed
 - Easy to swap out database engines by a simple configuration change
 - No need to create the schema on the new database

Appearance of Ajax (2005)



The beginning of real web-apps

Ruby on Rails (2007)



Ruby on Rails

- Ruby: a dynamically typed object-oriented programming language
- Rails: a web app framework, featuring:
 - automatic code skeletons
 - built-in testing features
 - object-relation mapping
 - default implementation of common webapp functions

The 3-Tier Architecture

- The 3-tier architecture consists of the following tiers (layers):
 - Front-end (client layer)
 - Client software – provides the UI of the system
 - Middle tier (business layer)
 - Server software – provides the core system logic
 - Implements the business processes / services
 - Back-end (data layer)
 - Manages the data of the system (database / cloud)

The 3-Tier Architecture Model

Data Tier
(Back-End)



Database

Middle Tier
(Business Tier)



Business Logic

Client Tier (Front-End)



Client Machine

network

network

network

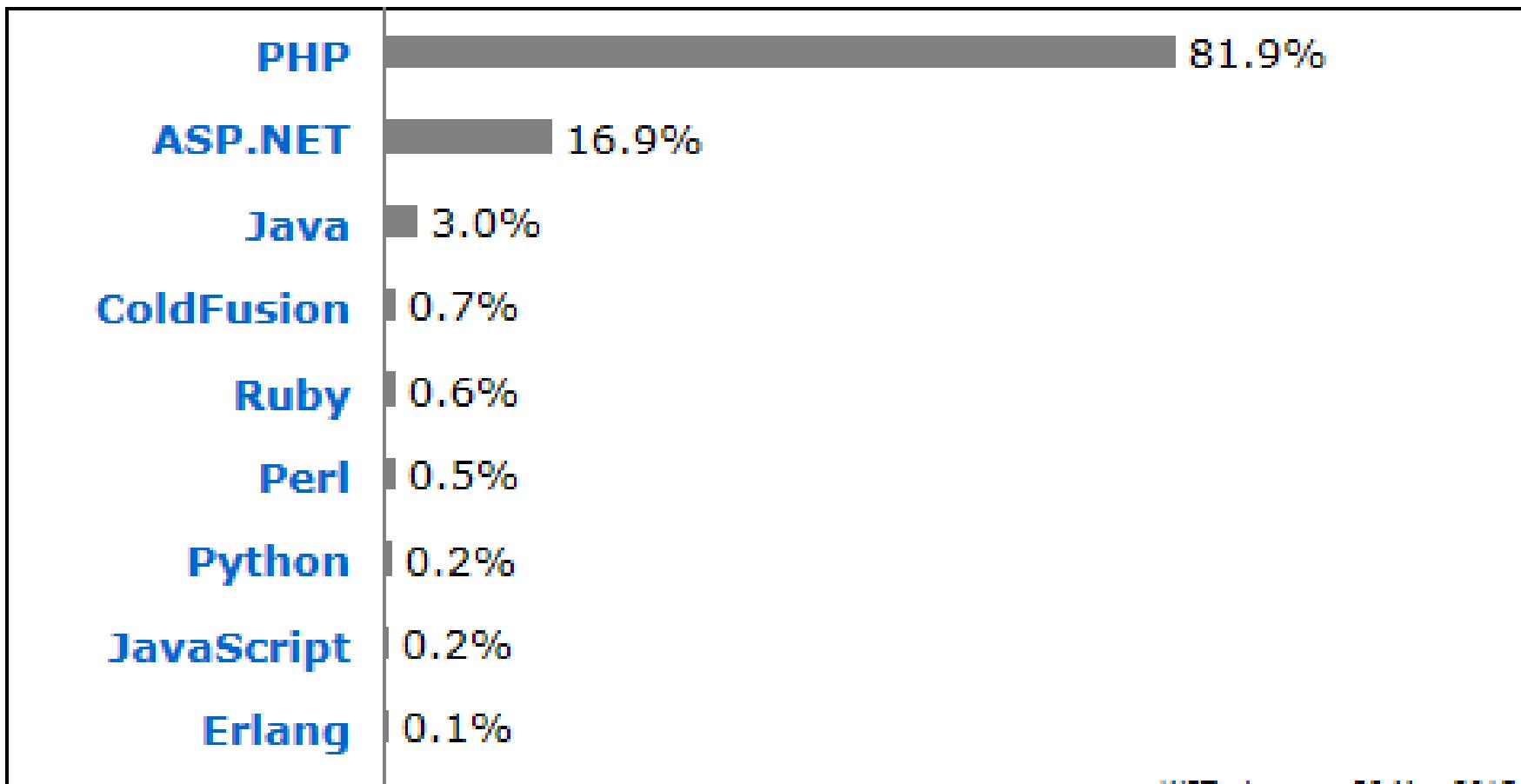


Mobile Client



Desktop Client

Market Shares of Server Languages



W3Techs.com, 29 May 2015

Percentages of websites using various server-side programming languages
Note: a website may use more than one server-side programming language

Three Trends between 2007-2010

- The rise of smart phones and mobile apps. Many applications had a web version and a mobile phone app for it.
- The rise of jQuery – a JavaScript library to build dynamic, beautiful web apps – and made Ajax easy!
- The release of Node.js –high performance JavaScript on the server.

MEAN Stack



A complete stack for Javascript, comprised of **MongoDB**, **Express**, **Angular**, and **Node**.



MongoDB for a schema-optional, "NoSQL" data store - perfectly suited for working with JSON data

express

The Express Javascript framework, with a large library of middleware and add-ons - including Mongoose, an ORM for MongoDB



AngularJS for declarative templating, and easy client-side application design



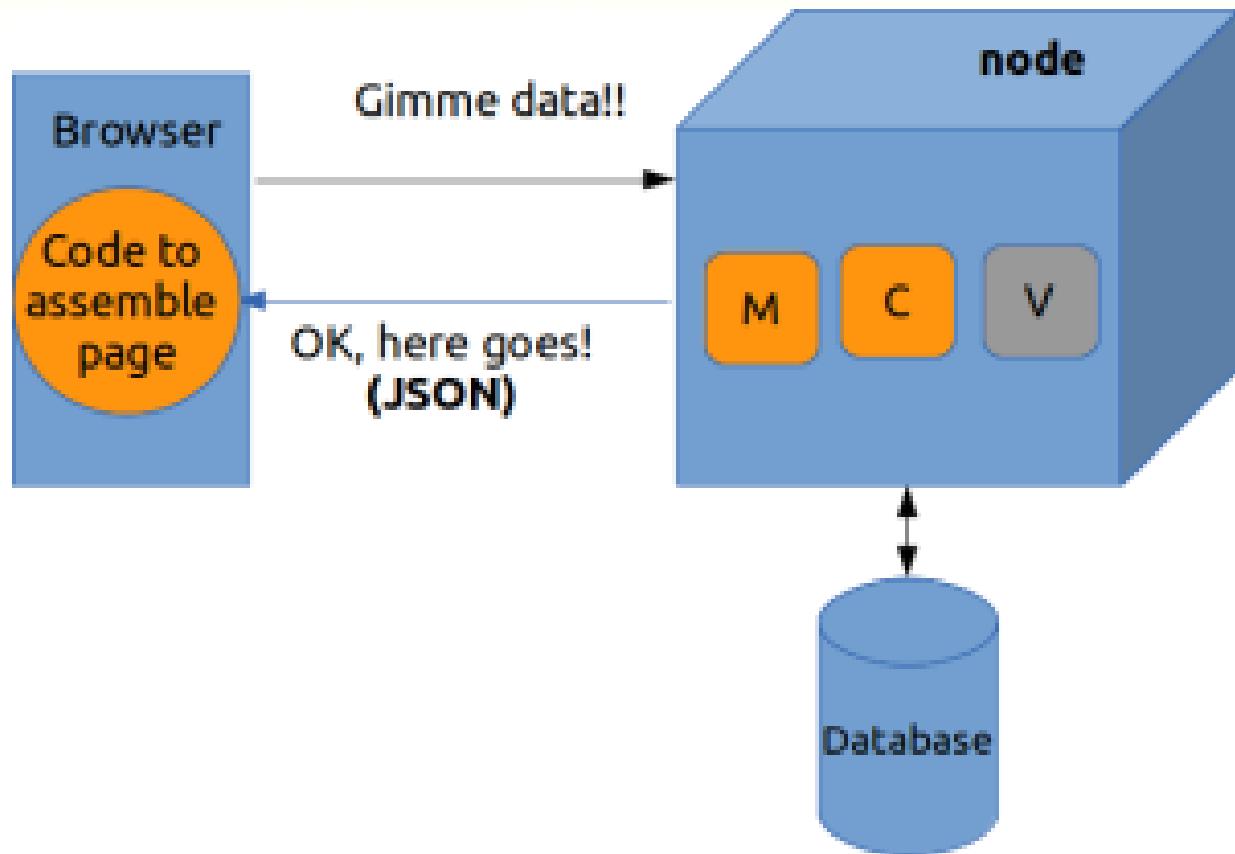
Nodejs, for an event-driven, server-side Javascript runtime - powered by the V8 engine

Introduction to Node.js

“Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.”

- nodejs.org

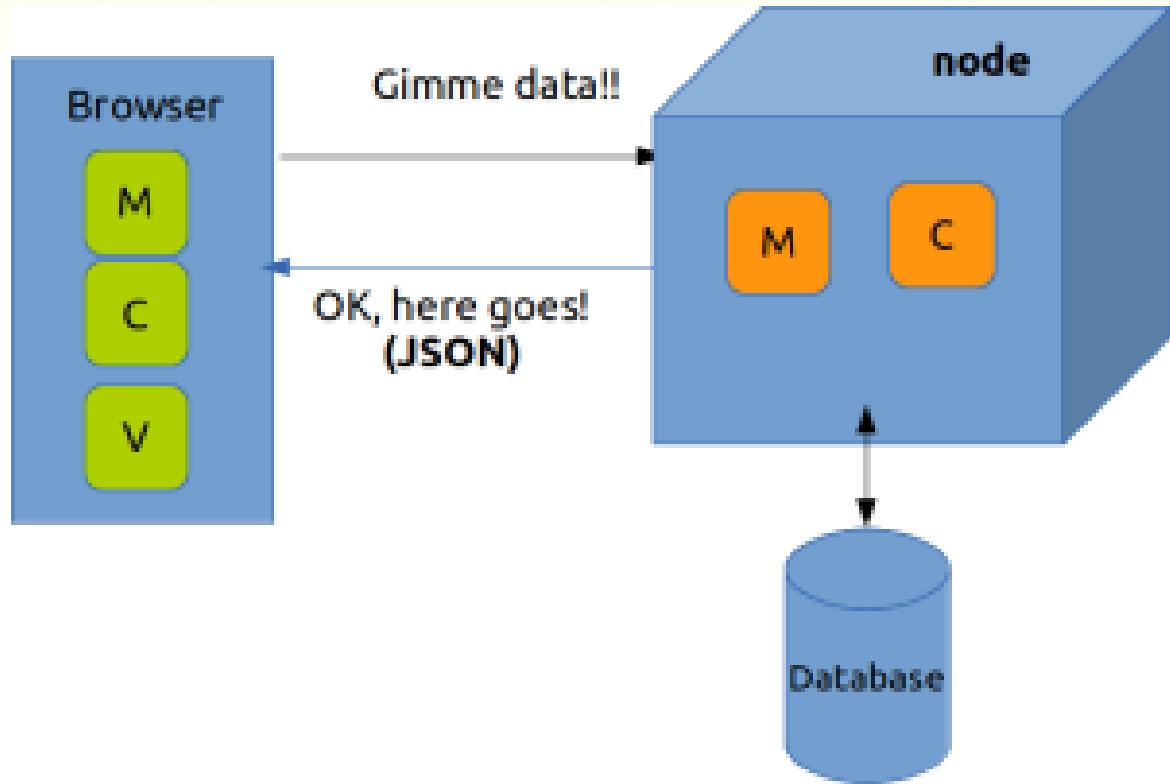
Node.js



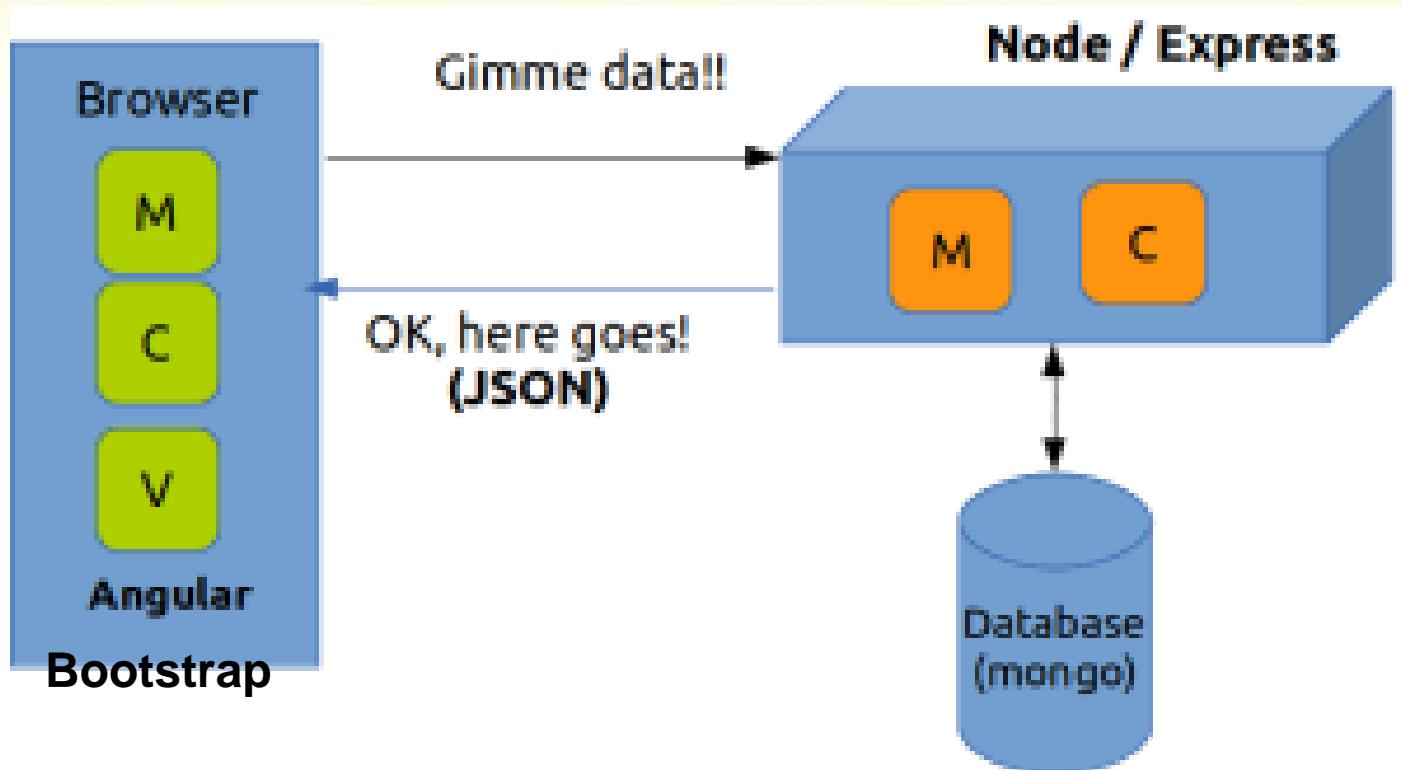
Focus on Client Side

- MVC Frameworks:

- ✓ Backbone
- ✓ Ember
- ✓ Knockout
- ✓ AngularJs



A Modern Web App Architecture



Mobility

- Everything
 - Web sites
 - Information
 - Services
- Everywhere
 - You only need your phone or tablet
- All the time



Cloud Apps



CLOUD COMPUTING

The Cloud is Coming !

- The cloud technologies are becoming inseparable part of our life.
- Cloud is a metaphor for the internet
- The world is moving towards the cloud!
- Software developers will also jump into the cloud: now or later, it will happen
 - This year, or few years later, everyone will develop applications for the cloud

Cloud Computing

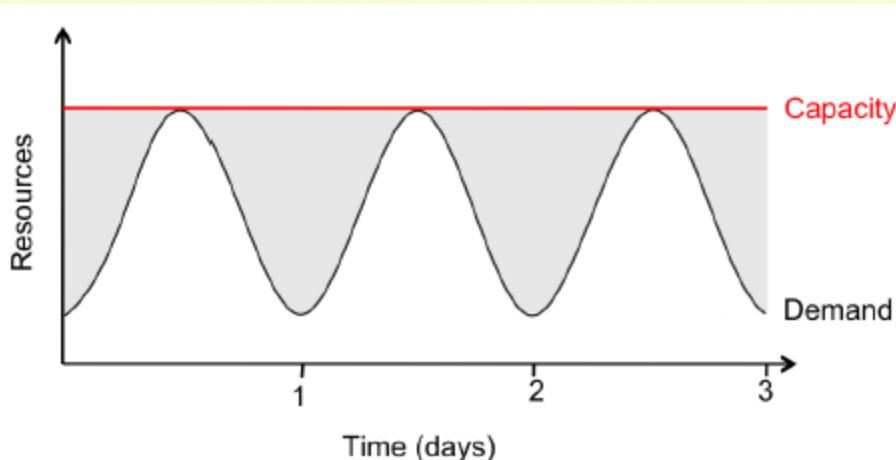
- Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. --- [NIST Definition](#)
- Cloud computing has five essential **characteristics**, three **deployment models**, and three **service models**.

Before the cloud

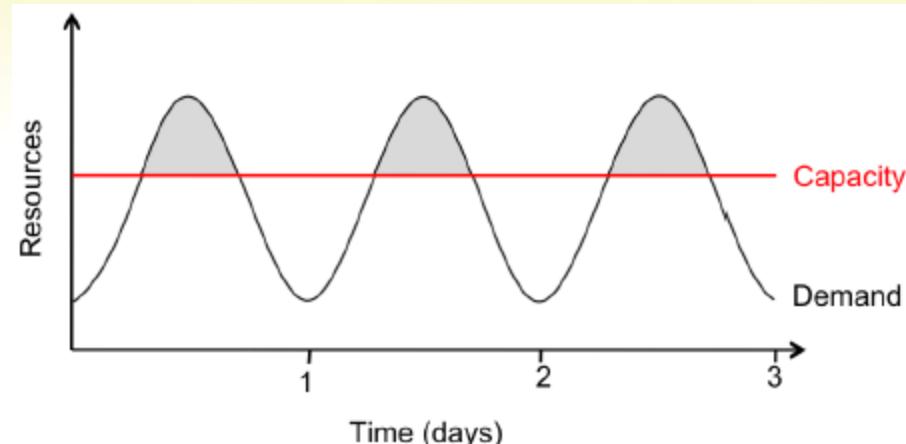
- If you wanted to start an enterprise app, you needed an IT shop
- Massive costs in hardware, software, power, administrative staff
- Prohibitive cost to entry



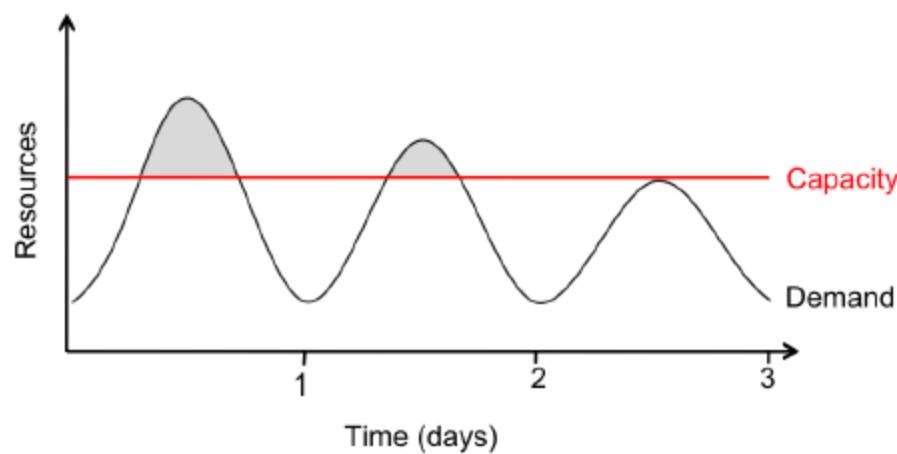
Variability for IT Resources



(a) Provisioning for peak load



(b) Underprovisioning 1

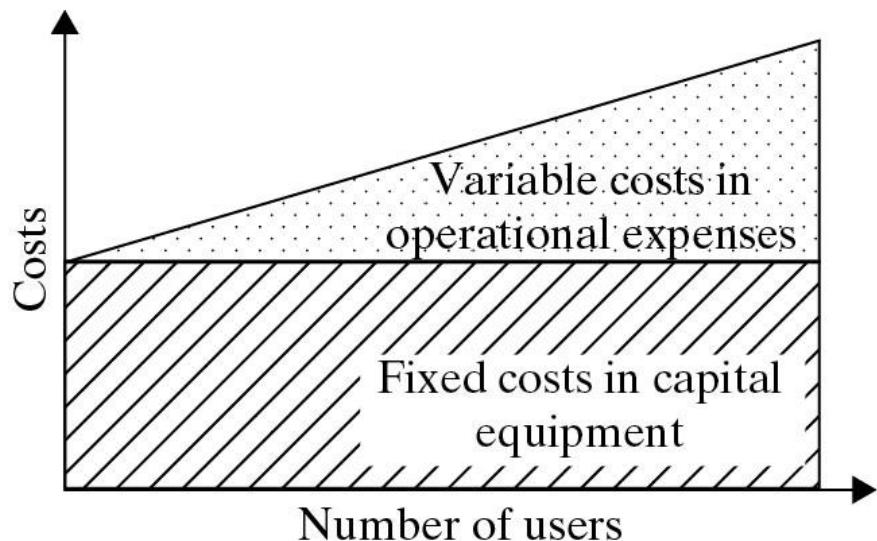


(c) Underprovisioning 2

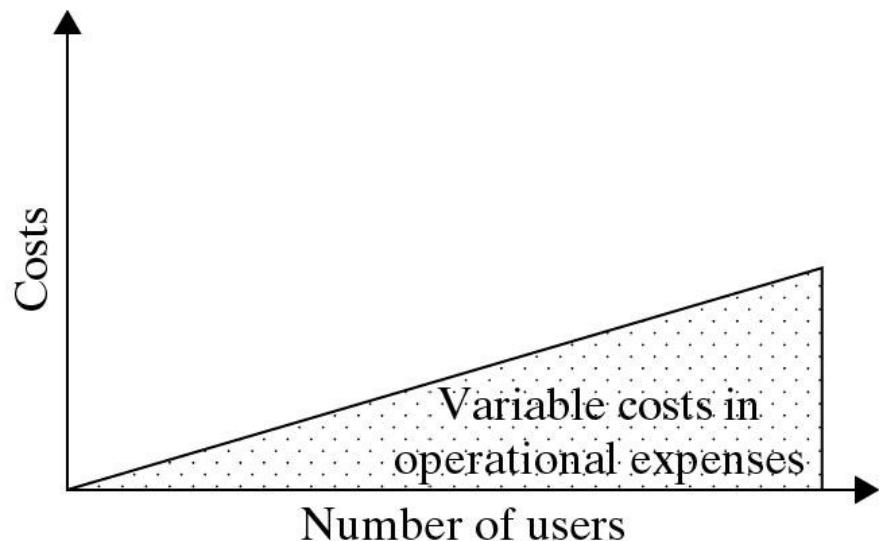
How the Cloud Works?

- In the cloud everyone consumes a portion of the shared computing resources
 - CPU, memory, storage, IO, networking, etc.
- If your business is small, you consume less
 - If your business is growing, you consume more resources from the cloud
- Pay as you go
 - Start for free, pay when you grow and need more resources

Cost-Effective of Cloud vs. Traditional

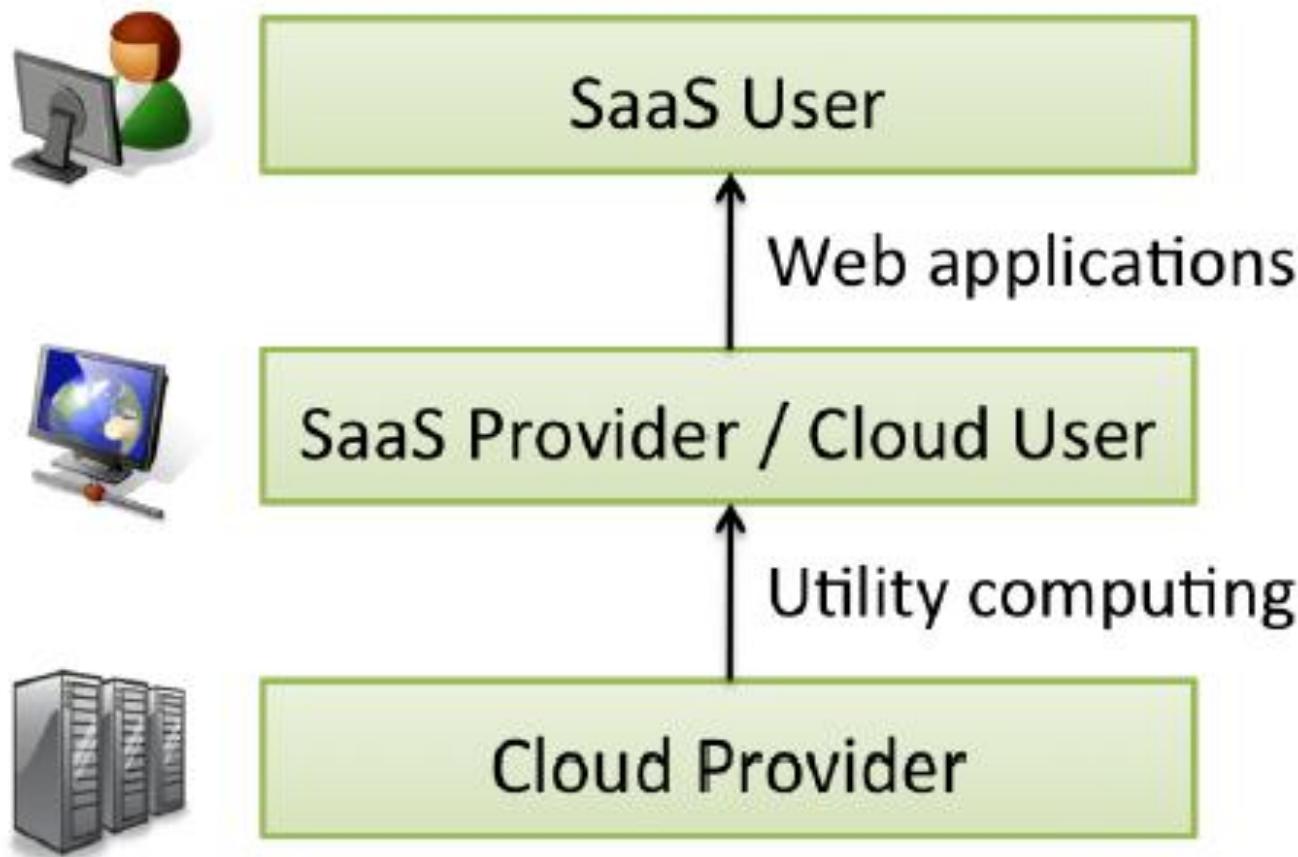


(a) Traditional IT cost model

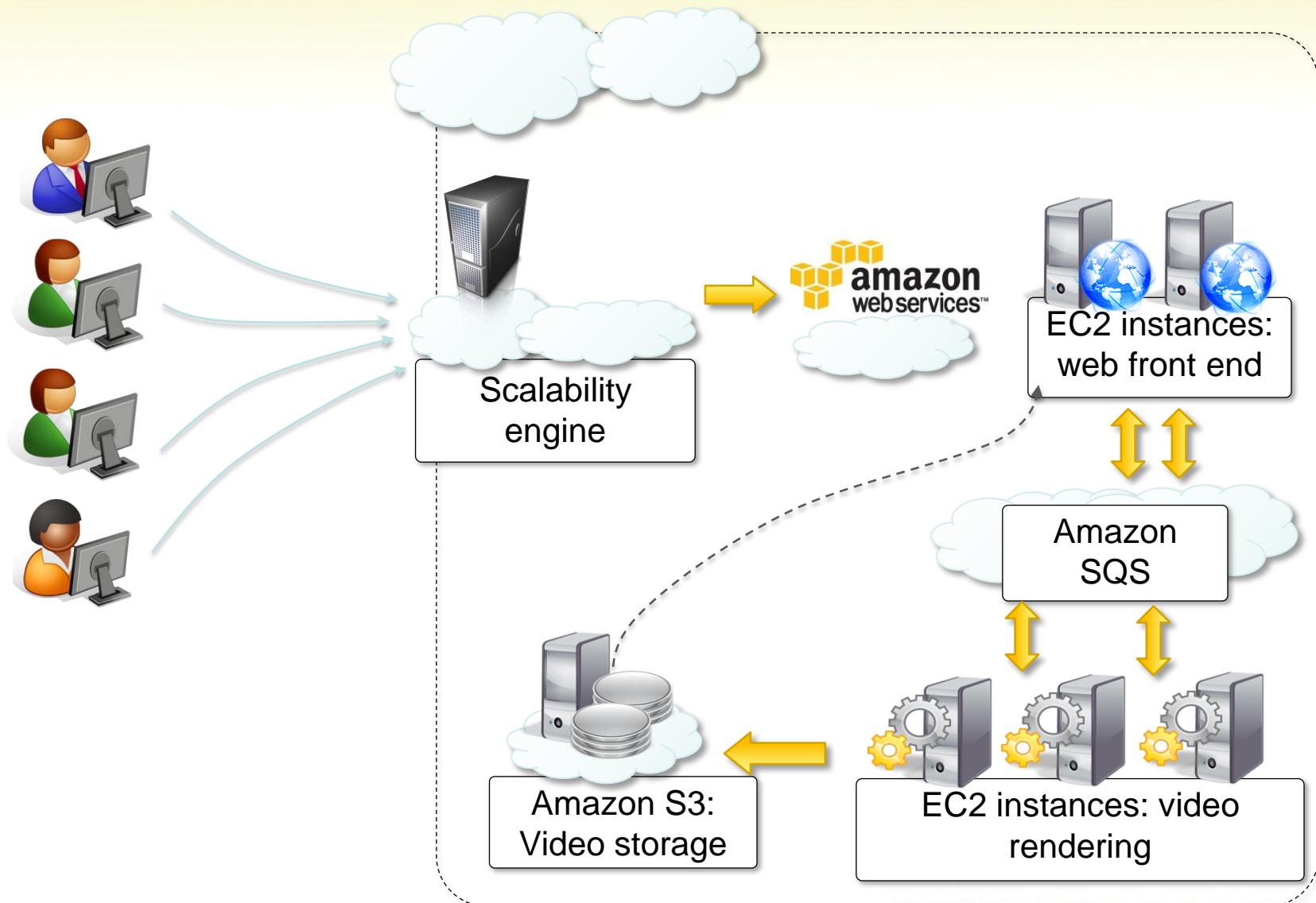


(b) Cloud computing cost model

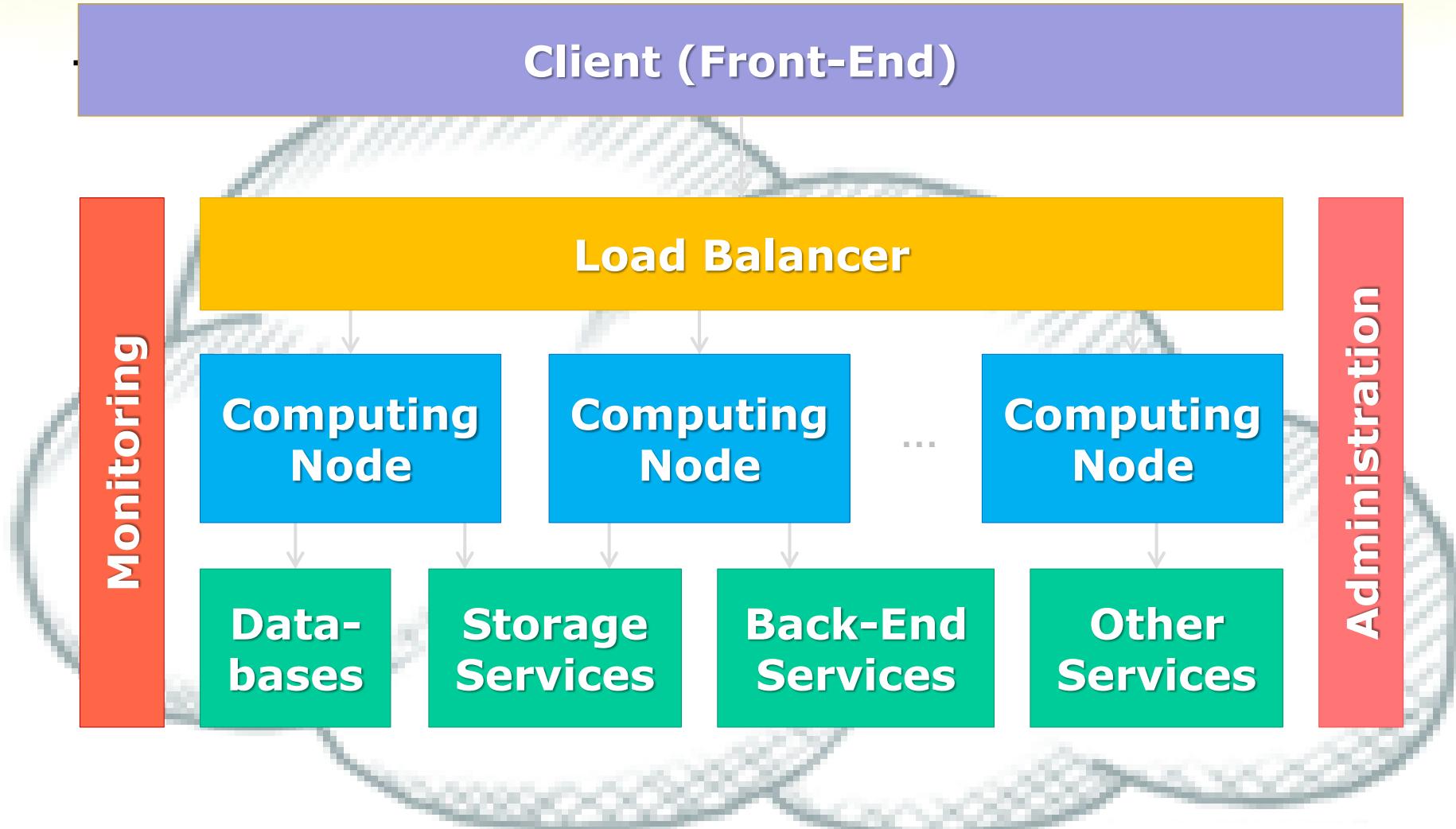
Users and Providers of Cloud



Example Cloud Application



Typical Cloud App Architecture



Why Cloud Computing is Appealing

- Cost
- Consumerization of IT
- Scalability and availability of services
- Sustainability or green IT

Benefits of Cloud Computing

Availability

- Access data anywhere, anytime through standard internet connection

Cost reductions

- Pay as you go model
- Savings on maintenance, repairs, and upgrades

Mobility

- Can access data and application from various devices
- Connecting employees, partners, and suppliers globally

Scalability

- Users have access to resources that scale quickly based on their demand

http://www.businessweek.com/magazine/toc/09_24/B4135cloud_computing.htm

<http://www.sei.cmu.edu/library/assets/whitepapers/Cloudcomputingbasics.pdf>

Green Computing of Cloud

Technology	Cost in Medium-sized DC	Cost in Very Large DC	Ratio
Network	\$95 per Mbit/sec/month	\$13 per Mbit/sec/month	7.1
Storage	\$2.20 per GByte / month	\$0.40 per GByte / month	5.7
Administration	≈140 Servers / Administrator	>1000 Servers / Administrator	7.1

- Data for 2006
- Medium-sized datacenter: 1000 servers
- Very large datacenter: 50,000 servers

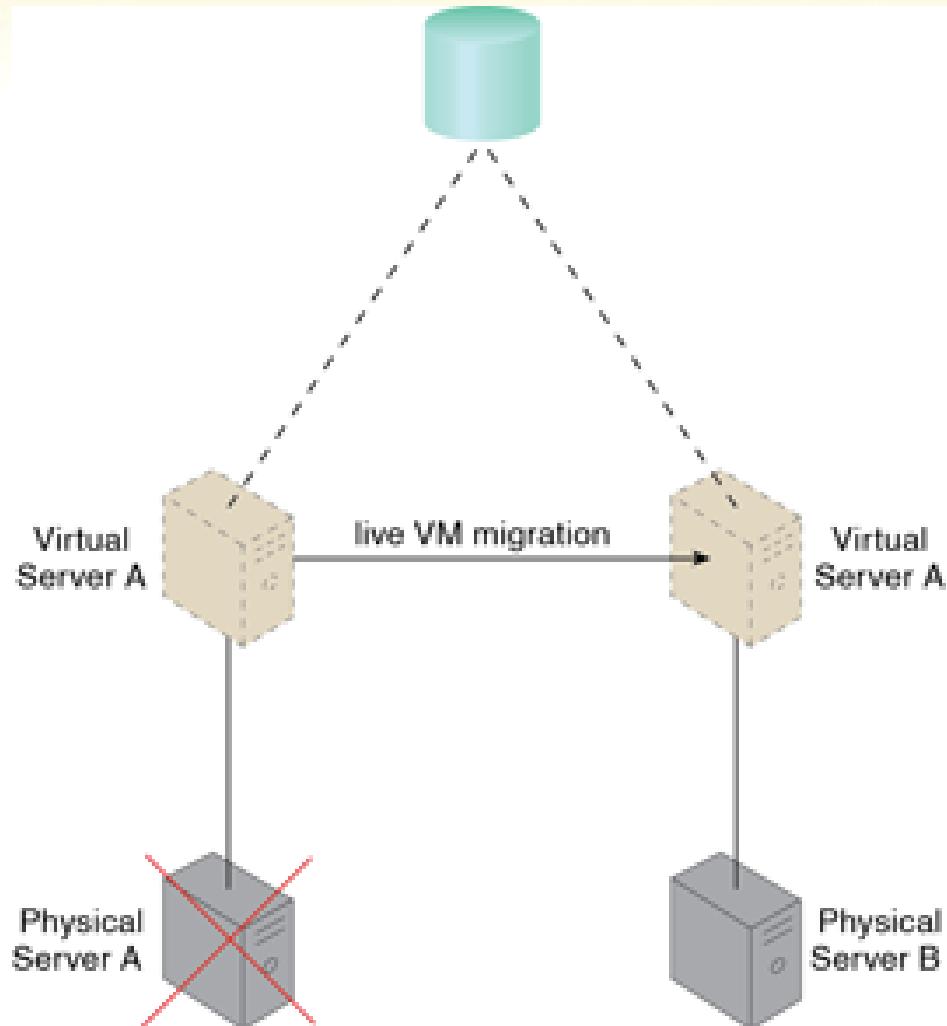
Source: Above the Clouds: A Berkeley View of Cloud Computing

Essential Characteristics of Cloud

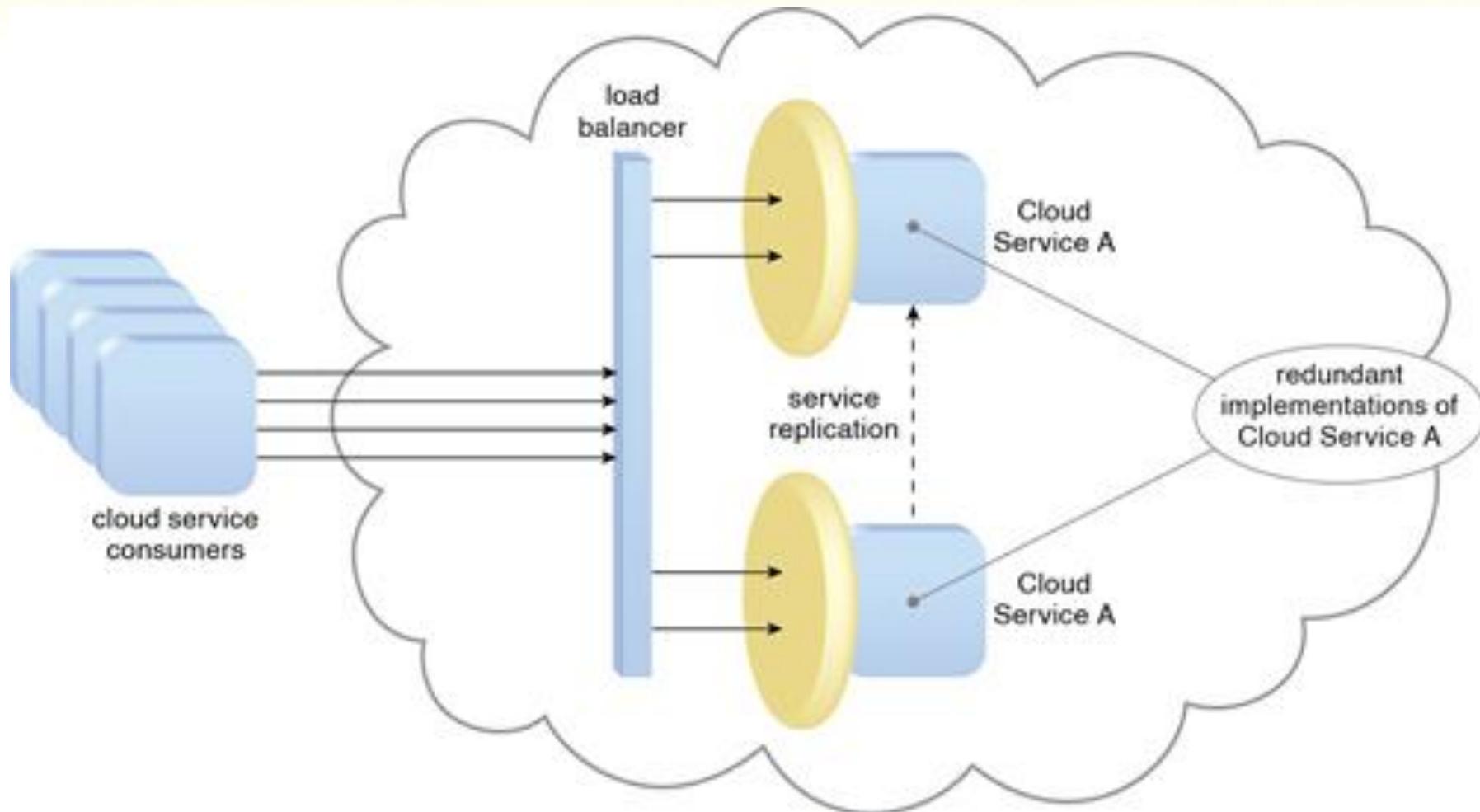
1. Broad network access
 - You can access the cloud from anywhere
2. Resource pooling
 - You work with virtual machines that could be hosted anywhere
3. Rapid elasticity
 - Easily go from 5 servers to 50 or from 50 servers to 5
4. On-demand self-service
 - You get elasticity automatically
5. Measured service
 - You pay for what you use

Source: NIST Working Definition of Cloud Computing

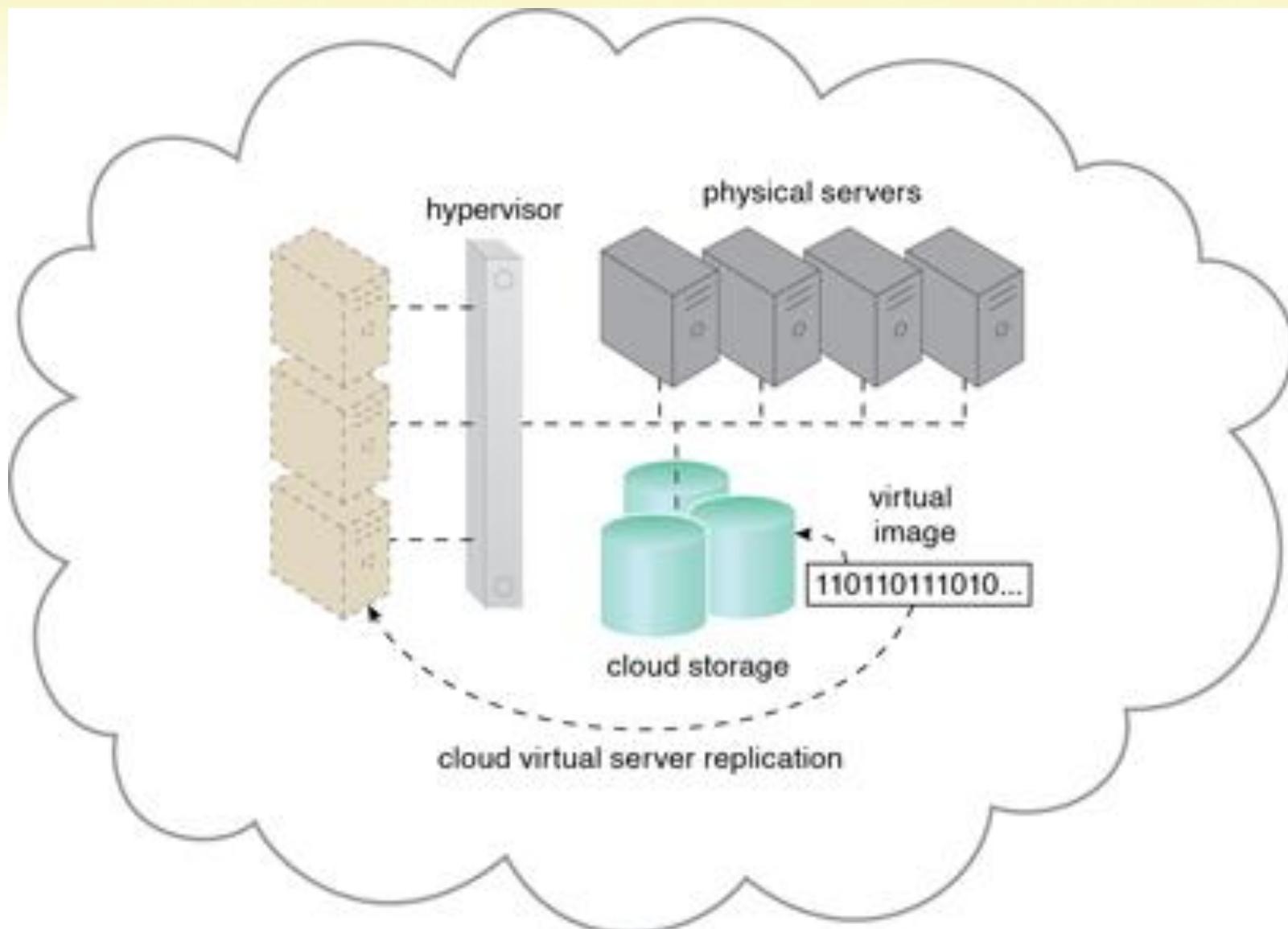
Zero Downtime



Load Balancer



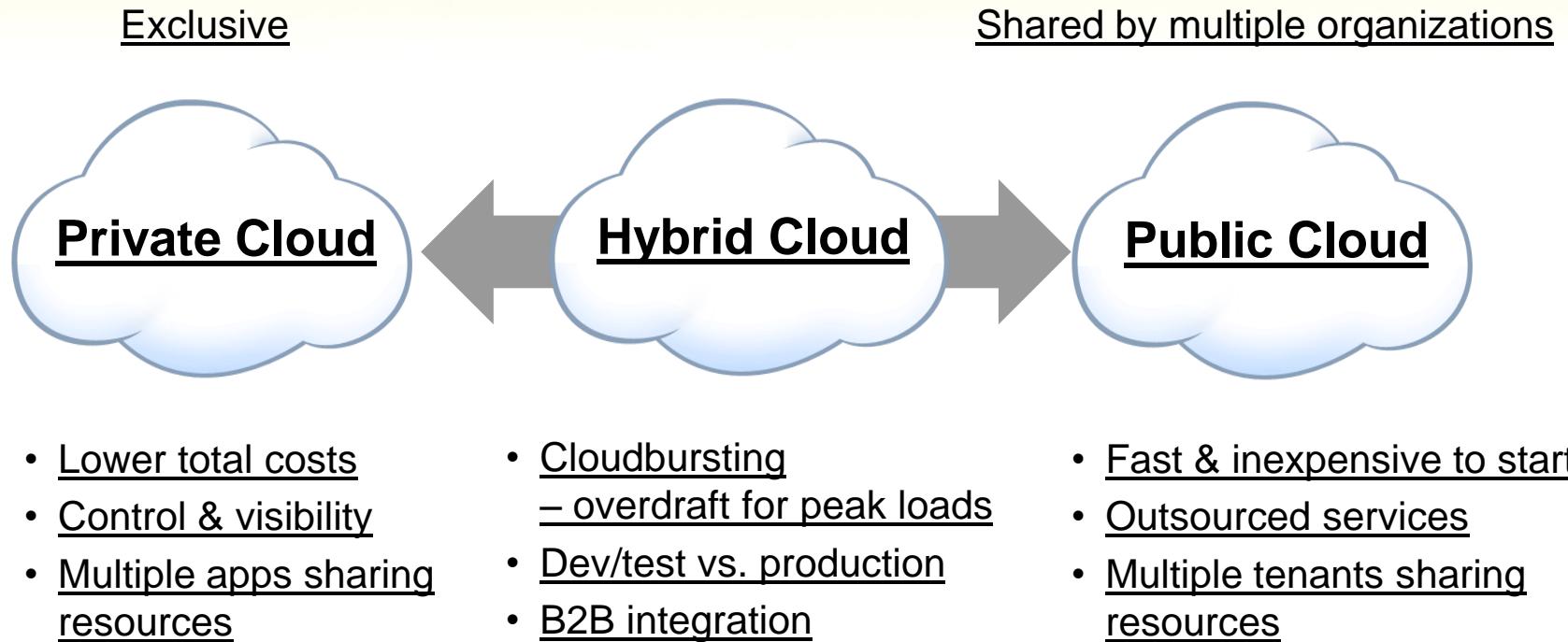
Resource Replication



Deployment Models

- Public clouds
 - Service provider owned and managed.
 - Access by subscription.
 - Standardized services
- Private clouds
 - Users owned and managed.
 - Access limited to client and its partner network.
 - Retaining greater customization and control.
- Hybrid clouds
 - Mix of private and public cloud

Choice of Deployment Models



Public Clouds

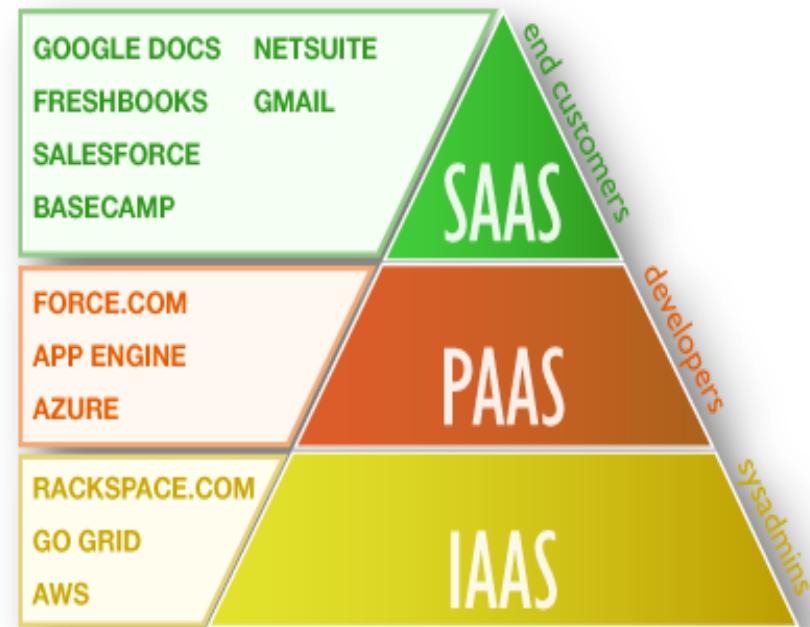
- Provide computing resources on demand
 - Publicly in Internet, for everyone
 - Paid on usage of resources
 - Could be IaaS, PaaS, SaaS or mix of them
- Examples of public clouds
 - **IaaS**: Aliyun, Qingyun, Amazon EC2, ...
 - **PaaS**: OpenShift online, IBM Bluemix, Heroku, Google App Engine, Amazon AWS, Windows Azure, Baidu BAE, JD JAE, Sina SAE...
 - **SaaS**: Google Apps, Microsoft Office 365, Salesforce.com, Adobe Creative Cloud ...

Cloud Service Models

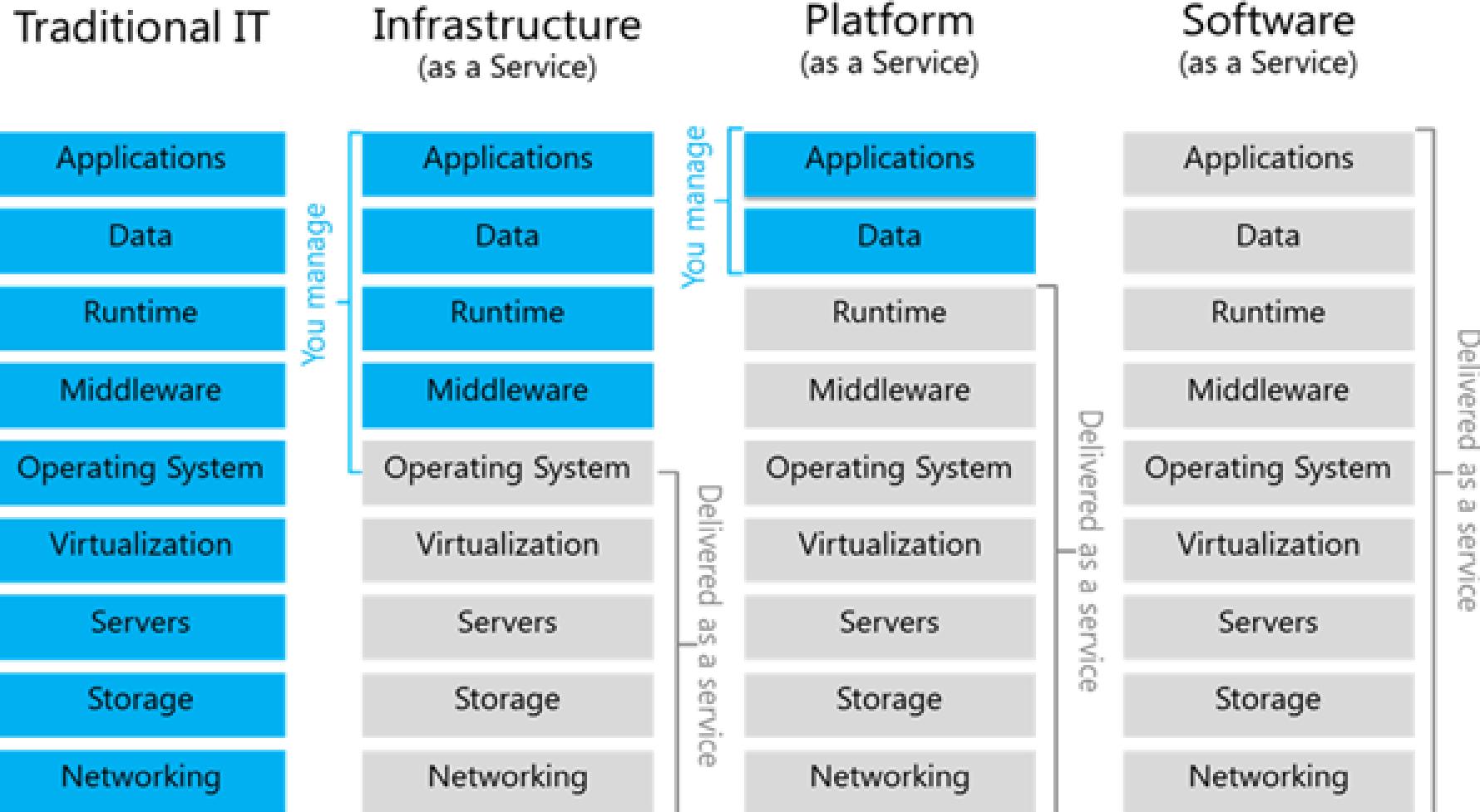
- Infrastructure as a Service (**IaaS**)
 - Virtual machines in the cloud on demand
 - Users install the OS and software they need
- Platform as a Service (**PaaS**)
 - Platform, services and APIs for developers
 - E.g. Java + JBoss + JSF + JPA + MySQL or JavaScript + Node.js + MongoDB + Express
- Software as a Service (**SaaS**)
 - Hosted application on demand (e.g. WordPress)

Types of Cloud Computing

- Software as a Service (SaaS)
 - Access to resources and applications
- Platform as a Service (PaaS)
 - Access to development and operational components
- Infrastructure as a Service (IaaS)
 - Completely outsources needed storage and resources



Levels of Service



IaaS (Infrastructure as a Service)

- IaaS ≈ rent virtual computers and other resources, such as networks and storeages.
- Most basic cloud service model
- Cloud users deploy their applications by then installing operating system images on the machines as well as their application software.
- You could modify your resources as you go
 - E.g. add more 100 GB HDD storage + 2 GB RAM
- IaaS pricing on the amount of resources allocated and consumed.

Example of IaaS: QingCloud

The screenshot shows a web browser window for the QingCloud website (<https://www.qingcloud.com/>). The page features a navigation bar with links for Home, Products, Customers, Pricing, Console, Documentation, and About, along with Sign In and Sign Up buttons. On the left, there's a diagram titled 'Client' showing two 'Target Portal Group' icons connected to a dashed box labeled 'Shared Storage Servers'. On the right, a large section is dedicated to 'Virtual SAN', with text explaining its setup and sharing capabilities, and a 'Learn More' button.

Client

Target Portal Group

Target Portal Group

+

+

Shared Storage Servers

Virtual SAN

You can setup iSCSI-based Virtual SAN service in QingCloud rapidly, sharing data within your applications.

Learn More

<https://docs.qingcloud.com/>

Console of QingCloud

The screenshot shows the QingCloud console interface. The top navigation bar includes links for Tickets, Consumptions, Help, and Account Lock, along with user information for lidan_gz. The left sidebar menu is under the 'Compute & Networking' category, with 'Instances' selected. The main content area displays the 'GUANGDONG1 / INSTANCES' page. It contains a descriptive text about elastic computing, followed by a toolbar with buttons for Refresh, New, Start, Stop, More Actions, and a dropdown for 'Display per Page' set to 10. Below this is a table header with columns: ID, Name, Status, Image ID, Network, EIP, Type, Alarm Status, Last Snapshot Time, and Created At. A message 'No Result' is displayed, indicating no instances are currently listed.

QINGCLOUD

Guangdong1

GD1

Overview

Compute & Networking

Instances

Images

VxNets

Routers

Load Balancers

Elastic IPs

DNS Aliases

Storage

Security

Database & Cache

Tickets

\$ Consumptions

Help

lidan_gz

Account Lock

GUANGDONG1 / INSTANCES

QingCloud provides an elastic computing capability accessible at any time - Instance, a well configured server, with expected hardware configuration, operating system, and network configuration. Any resource request can be accomplished in 10 to 60 seconds, enabling an on-demand computing resource.

Display per Page: 10

ID	Name	Status	Image ID	Network	EIP	Type	Alarm Status	Last Snapshot Time	Created At
----	------	--------	----------	---------	-----	------	--------------	--------------------	------------

No Result

* Tip: You can click on each resource by "right clicking" for typical operations, and "double-click" to modify the basic attributes.

PaaS (Platform as a Service)

- PaaS ≈ rent a complete development platform;
- Cloud vendors deliver a computing platform typically including operating system, programming language execution environment, database, and web server.
 - E.g. Linux + Python + Django + MongoDB + Nginx load balancer + web server
- Users develop and run their software on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.

PaaS vs. IaaS

- IaaS better for migrating existing applications
 - More flexible, you install your environment
- PaaS has lower demands on administration
- PaaS will take care of scaling if applications use correct frameworks, also redundancy and CDN
- PaaS better for new applications
- BUT has dangers of vendor lock in if platform specific functions are used

More about PaaS

- Public solution stacks for web applications
 - OS, web server, language interpreters, provisions for automatic scaling, all shielded from the user
- Each system only has a few supported languages
 - Automatic deployment and scaling not trivial
- Offers development tools
 - Libraries for specific services
 - IDE plugins, deployment tools

Typical PaaS Architecture

Front-End: HTML5, CSS3 JavaScript / Mobile

Middle-Tier Languages and Frameworks:

PHP, Java, C#, Python, Ruby, JavaScript,
Symfony, Zend Framework, JSF, ADF, Django, Rails,
ASP.NET, ASP.NET MVC, Node.js

Computing Nodes:

Amazon EC2, Azure
Compute, App Engine

Back-End :

Relational DBs, NoSQL
(MongoDB), Blob Storage,
Message Queues, CDN,
Notifications

Operating Systems:

Linux / Windows / other

Classical PaaS Stacks

- Java + JBoss app server + Java ServerFaces + JBoss Rich Faces + Java Persistence API + Oracle database
- Python + Django + MongoDB + Linux cron jobs + Nginx load balancer + Gunicorn web server
- .NET Framework + C# + ASP.NET + WCF + SQL Server + Nginx load balancer + IIS web server
- PHP + Zend Framework + Cassandra DB + Nginx load balancer + Apache web server
- **JavaScript + Node.js + MongoDB + RabbitMQ**
- Ruby + Ruby on Rails + MySQL + Sphinx + Memcache + Unicorn HTTP server

Find your Platform as a Service

- <http://www.paasify.it>

Find your Platform as a Service!

What's best on your PaaS? Define your needs and get a list of candidates that claim to be your best fit.

Find your PaaS

Comprehensive

More than 70 vendors

...and counting.

Comparable

Distinctive PaaS features

A set of distinctive and intersecting properties to enable comparison and matching of different PaaS offerings.

Current

Continuously updated

Data structures are [publicly available](#) and editable by the community. We also aim at vendors to [verify](#)  their profiles.

SaaS (Software as a Service)

- SaaS ≈ rent an application in cloud.
- Cloud vendors install and operate application software in the cloud.
- One application instance may be serving hundreds of customers.
- Customers access the software from cloud clients.
- A customer can configure the application through metadata

SaaS Concepts

- Simply renting an application instead of setting it up on your own server
- Examples:
 - Exchange hosting (\$10/user/month)
 - Wordpress hosting (\$20-\$150 / month)
 - Web hosting (\$90 / year)
 - Quickbooks (\$50 / month)
 - Salesforce (\$125/user/month)
 - World of Warcraft (\$20/month)
- These are all cloud apps (computing as a utility)

SaaS Example



- **Up to now:** on-demand customer relationship management (CRM), marketing, Web analytics as hosted service
- **Now:** platform to enable developers to create and provide arbitrary business applications on-demand



Why SaaS is Attractive

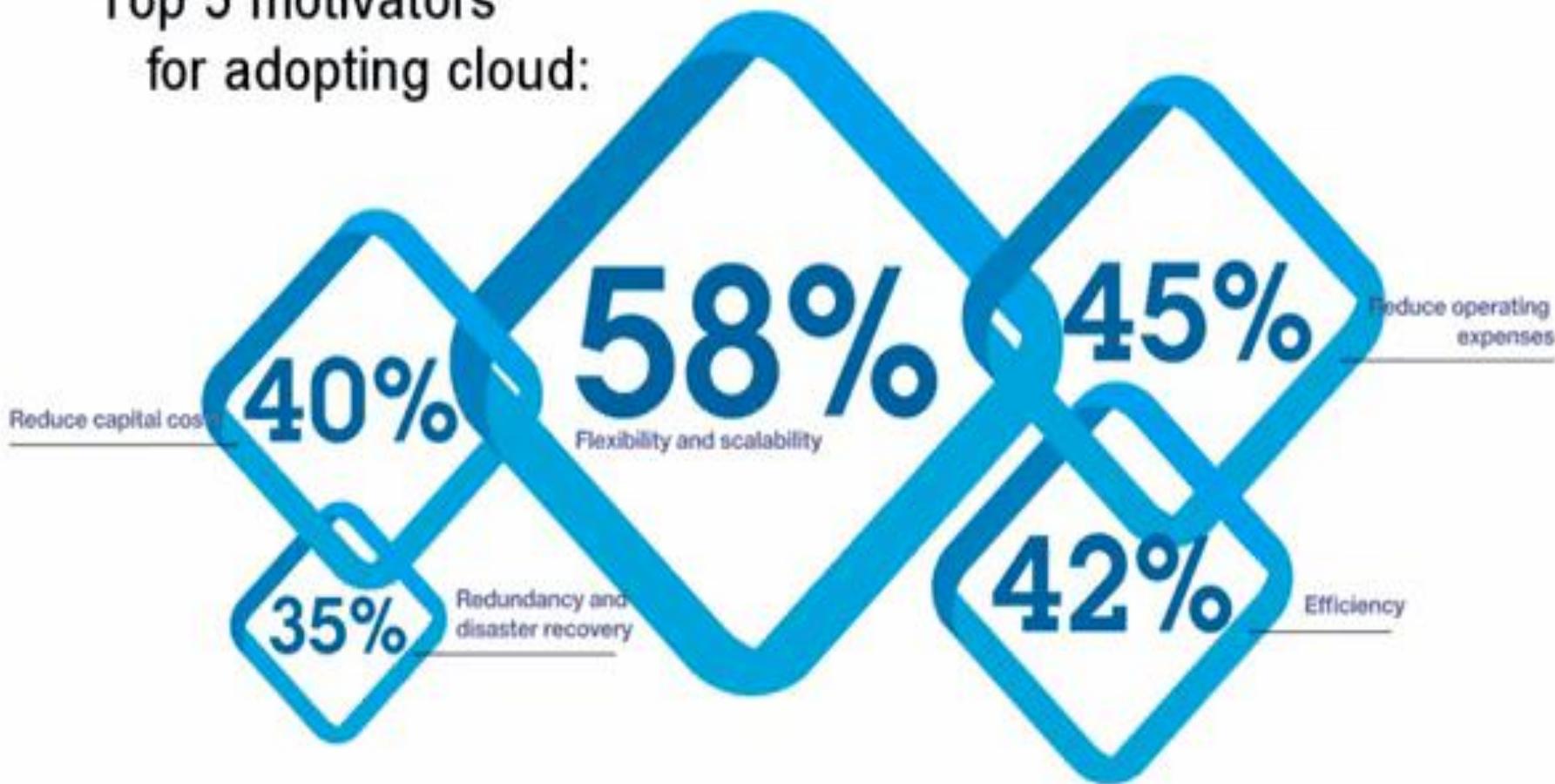
- No software installation on private PC, no upgrades, no patches, no service packs, no maintenance
- No license renewal when computer is exchanged
- All data resides on the Web
- “Pay-as-you-go” business model, i.e., payment based on usage

Periodic Table of SaaS Ecosystem 2015



Top Motivators for Adopting Cloud

Top 5 motivators
for adopting cloud:



Issues with Cloud Computing: Security and Privacy

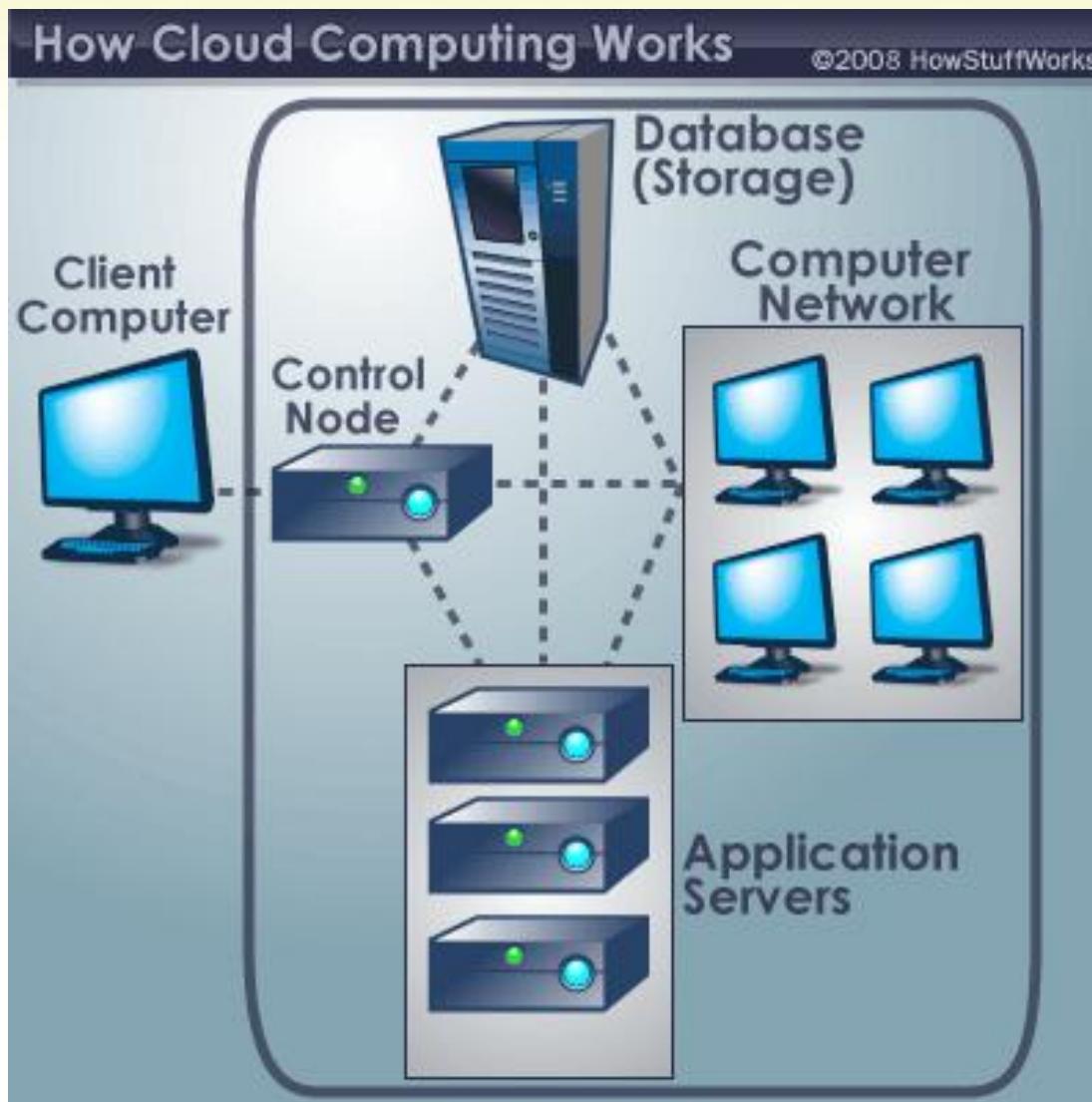
- Does the cloud back up your data?
- Is your data always safe?
- Can you conduct business abroad?
- Who is given access to your data?
- Who else is on your server?



Disadvantages of cloud computing

- Dependent on internet connections
- Users are subject to terms and conditions
- Data in hands of a 3rd party
- No worldwide accepted standards

Cloud Computing Architecture



Techniques of Cloud Computing

- Multi-tenancy
- Virtualization
- Distributed Computing
- Parallel Computing
- Utility Computing
- Network Storage
- High Available
- Load Balance
- Service & Interface

Web Apps from Browser/Server to Browser/Cloud



IaaS Example: Aliyun, Ecosystem

SaaS



[Alibaba](#)



[Taobao](#)



[Alipay](#)

PaaS



Computing



Storage



Searching



Maps



Hosting



vCluster

IaaS



Virtual Computing



Virtual Storage



Virtual Networks

Hardware



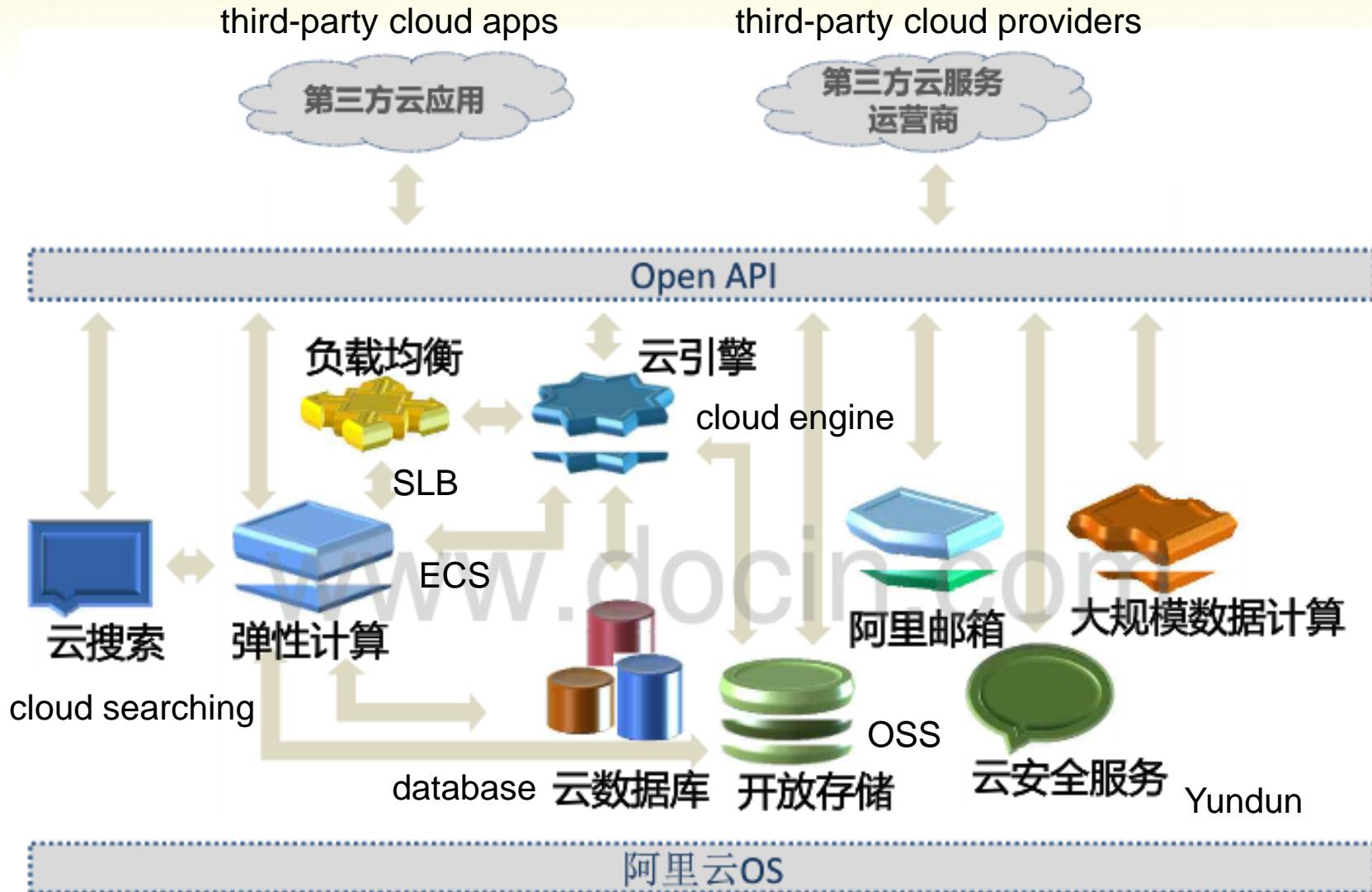
Data Centers of Aliyun



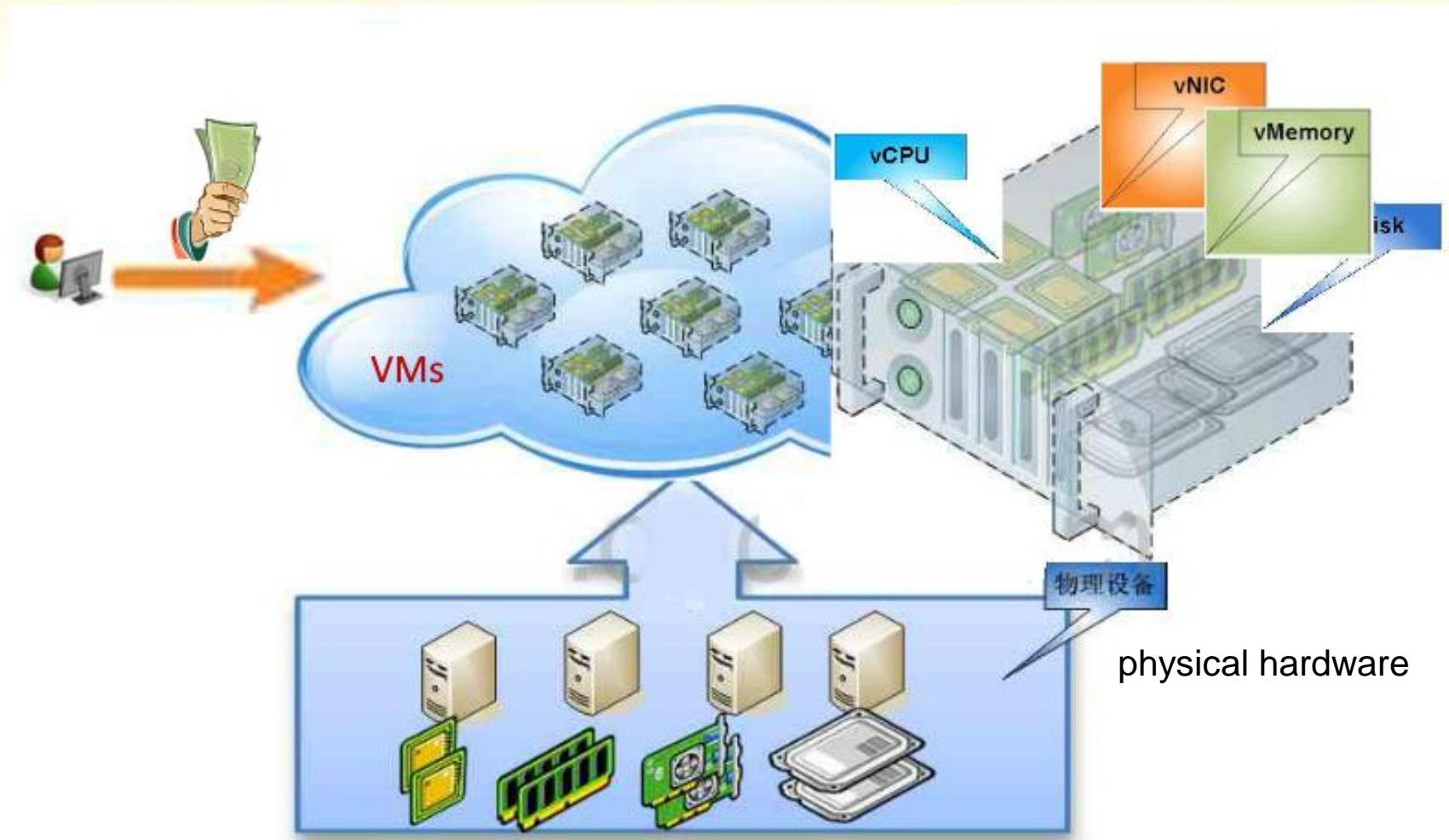
Main Services of Aliyun

- Elastic Computing:
 - **ECS**: Elastic Compute Service. Scalable capacity to build stable and secure applications.
 - **SLB**: Server Load Balancer. Distributes traffic for multiple cloud servers to improve the availability.
- Storage and CDN
 - **OSS**: Open Storage Service.
 - **CDN**: Content Delivery Network
- Database:
 - **RDS** (MySQL), OTS (Open Table Service (OTS), a NoSQL database)
 - **OCS**: Open Cache Service . An online caching service.
- Security and Management
 - **Yundun**: avoid heavy-traffic DDoS attacks
 - **CMS**: Cloud Monitor System. A platform to monitor sites and servers in real time.

Products of Aliyun



ECS: Elastic Compute Service



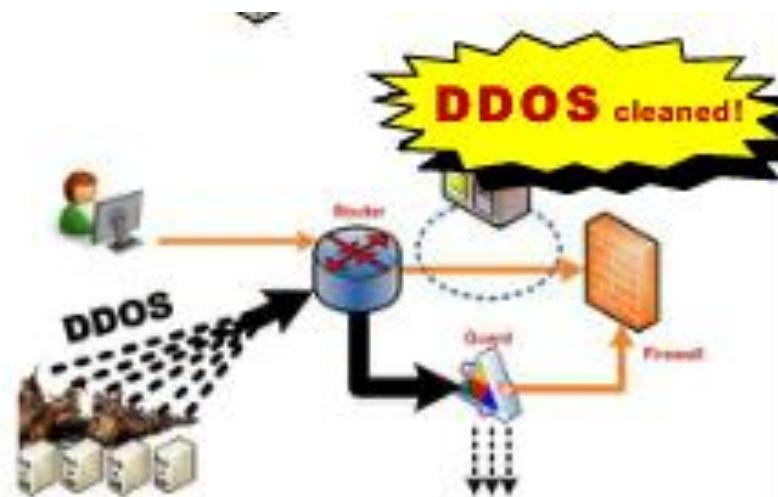
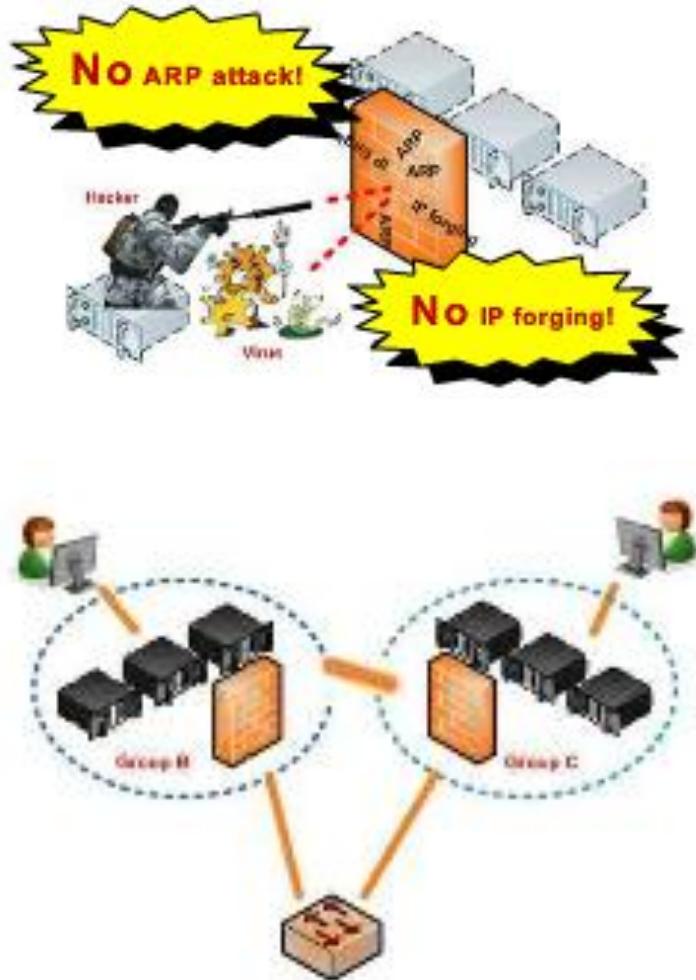
ECS: Elastic Compute Service

- Ability to increase or decrease capacity within minutes
- Ability to commission one, hundreds, or even thousands of server instances simultaneously
- A web service API to control the scaling of instances depending on your needs
- A “pay only for what you use” pricing model

OSS: Open Storage Service.

- Offers secure online storage system for any kind of data
- Automatically replicated 3 copies
- Data is universally accessible over the web
- No limits on amount of data, longevity or bandwidth for transfer or publish
- Scalable, distributed system
 - Stores data redundantly across geographically separated data centers

Security of Aliyun: Cloud Shield



Prices of Aliyun

香港区域ECS包年包月定价				
CPU(核)	内存(GB)	香港年价Linux (元)	香港年价Windows (元)	
1	1	1,160	1,760	
1	2	1,550	2,360	
1	4	2,340	3,550	
2	2	2,310	3,520	
2	4	3,100	4,710	
2	8	4,680	7,110	
4	4	4,630	7,030	
4	8	6,200	9,420	
4	16	9,350	14,210	
8	8	9,250	14,060	
8	16	12,400	18,850	
8	32	18,700	28,420	
16	64	37,400	56,850	

计费项	除Windows外的操作系统			Windows操作系统		
	包年包月		按量付费	包年包月		按量付费
	月价	年价	小时价	月价	年价	小时价
数据盘 (元/GB)	0.5	5	0.0009	0.7	7	0.001
按固定带宽计费(元/Mbps)	232	2320	0.58	232	2320	0.58
按流出流量付费 (元/GB)	n/a	n/a	1.3	n/a	n/a	1.3

OPENSHIFT ONLINE

OPENSIFT

- <https://www.openshift.com/>
- Red Hat's open source Platform-as-a-Service (PaaS)
- Allow developers to quickly develop, host, and scale applications in a cloud
- Offer online, on premise, and open source project options.

Three Versions of OpenShift

	OpenShift Online	OpenShift Enterprise	OpenShift Origin
What is it?	Hosted PaaS Service	Private PaaS Product	Open Source PaaS Project
How can it help me?	Quickly develop, host, and scale applications in the public cloud.	Accelerate IT service delivery and streamline application development.	Use a free, open source PaaS or help extend OpenShift.
How is it priced?	Free or Premium Plans	Annual software subscription	Free and open source
Who provides support?	Community Support for Free & Bronze Plans or Red Hat Support for Silver Plan	Red Hat	Community
Where does it run?	In the public cloud	On your servers or in your private cloud	Your laptop, your servers, private cloud, or public cloud
Who is it good for?	Startups, developers, small businesses, and even enterprises	Enterprises that want to run their own cloud	Anyone that wants to tinker on the latest thing in open source software
How do I get it?	Sign Up Online	Download a free evaluation	Download from GitHub

OpenShift Online

- Red Hat's public cloud application development and hosting platform
- Automate the provisioning, management and scaling of applications
- Supporting languages: Java, Ruby, PHP, Node.js, Python, Perl, and more;
- Supporting Middleware: Jboss, Tomcat, Zend server
- Supporting Frameworks: Django ,Drupal ,Flask,Rails ,Switchyard ,Vert.x
- Supporting Native Services: Jenkins, Mongodb, MySql, Openshift Metrics, Pgrouting, Postgis, PostgreSQL
-

Services and Features

- **World-class Support**
 - Supported by Red Hat's award-winning technical support.
- **Auto Scaling**
 - Automates the scaling of your application as your user traffic increases. Pay-as-you-go access to more and faster servers.
- **Custom SSL**
 - Secure traffic to your custom domains with SSL and your own certificates.
- **Extra Storage**
 - Access to more fast local storage for your applications.

Terminology of OpenShift

- **Application:** OpenShift is focused on hosting web applications.
- **Gear:** a gear is a service container running the application. Gear size: small (512M RAM, 1G disk), medium(1G RAM, 1G disk) and large(2G RAM, 1G disk).
- **Cartridge:** plug-ins a gear to run an application, such as [Tomcat](#), [Node.js](#), [MySQL](#).
- **Scaling:** If you allow your application to scale, a load balancer allocates more gears to handle traffic as your application needs it.

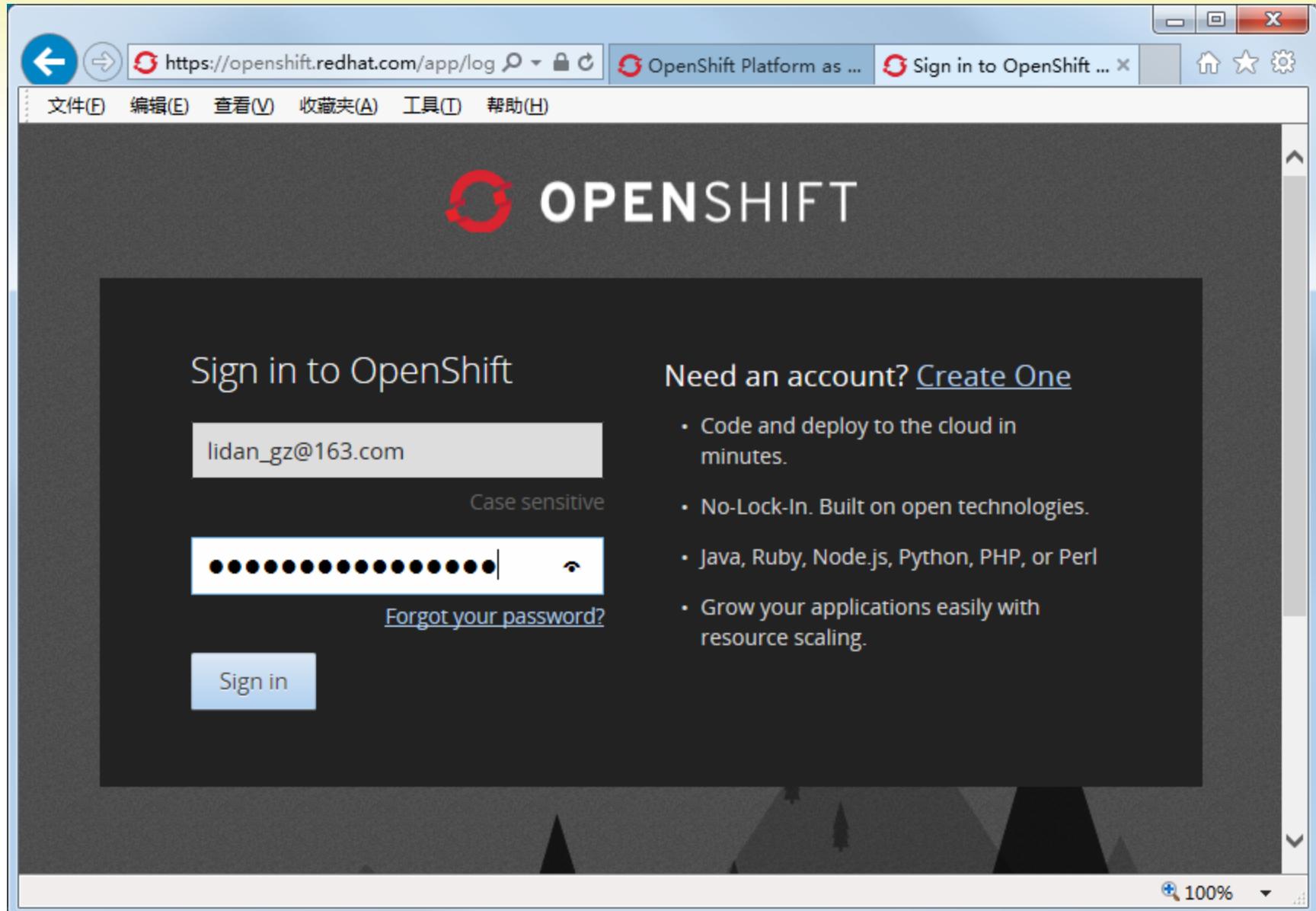
Monthly Pricing

	USD	CAD	EUR	Free Plan	Bronze Plan	Silver Plan
\$ BASE PRICE				Free	Free	\$20/month
⌚ APPLICATION IDLING				24 hours	Never	Never
⚙ INCLUDED GEARS				3 small gears	3 small gears	3 small gears
⚙ MAX GEARS				3	16	16+
↗ SCALING				Yes (3 min / 3 max)	Yes (3 min / 16 max)	Yes (3 min / 16 max)
⚙ GEAR SIZES				small	small (\$0.02/hour) small.highcpu (\$0.025/hour) medium (\$0.05/hour) large (\$0.10/hour)	small (\$0.02/hour) small.highcpu (\$0.025/hour) medium (\$0.05/hour) large (\$0.10/hour)
💾 STORAGE				1GB per gear	1GB per gear; \$1.00/month per additional GB	6GB per gear; \$1.00/month per additional GB
🔒 SSL				Shared	For custom domains	For custom domains
👥 TEAMS				Not included	Up to 15	Up to 15
JBoss EAP 6				Included	3 gears free; \$0.03/hr per additional gear	3 gears free; \$0.03/hr per additional gear

Steps to Use OpenShift Online

1. Create an "Application" in OpenShift Online (with the web console, command-line tools, or your IDE)
2. Code the application in your favorite environment, or use one of the available Quickstarts.
3. Push the application code to OpenShift Online (using [Git](#))

Log in to OpenShift web console



Create an Application

1. Choose a web framework or codebase to start from
 - Try JBoss, PHP, Python, Ruby, Node.js
 - or create a new Drupal or Wordpress site instantly.
2. Add **cartridges** like MySQL or MongoDB to your application
 - including databases, cache servers, management tools, and continuous integration servers.
3. Upload your code to OpenShift via Git
 - Your source code is stored with your application in a Git version control repository.

Choose Type of the Application

OPENSHIFT ONLINE

Applications Settings Help ▾ OpenShift Hub

1 Choose a type of application 2 Configure the application 3 Next steps

Choose a web programming cartridge or kick the tires with a quickstart. After you create the application you can **add cartridges to enable additional capabilities** like databases, metrics, and continuous build support with Jenkins.

Search by keyword or tag or

Browse by tag... ▾

Cartridge – A managed runtime for your application.
QuickStart – A quick way to try out a new technology with code and libraries preconfigured. You are responsible for updating core libraries for security updates.
Receives automatic security updates

Instant App see all

-  Jenkins Server CI
-  Drupal 7 CMS DRUPAL NOT SCALABLE PHP
-  Ghost 0.5.10 BLOG CMS GHOST NODEJS
-  OpenShift Backup Server BACKUP
-  WordPress 4 BLOG CMS NOT SCALABLE PHP

xPaaS see all

-  JBoss Data Virtualization 6 JAVA EE 6
-  JBoss Enterprise Application Platform 6 JAVA EE 6
-  JBoss Unified Push Server 1.0.0.Beta1 JEE FULL PROFILE
-  JBoss Unified Push Server 1.0.0.Beta2 JEE FULL PROFILE

123

Configure the Application

Based On

WordPress 4 Quickstart 

An open source, semantic, blogging and content management platform written in PHP with a MySQL backend focusing on aesthetics, web standards, and usability.

[Learn more](#)

 OpenShift maintained

 Does not receive automatic security updates

Public URL

<http://gyblog>

-lidangz.rhcloud.com

OpenShift will automatically register this domain name for your application. You can add your own domain name later.

Source Code

<https://github.com/openshift/wor>

Branch/tag

Your application will start with an exact copy of the code and configuration provided in this Git repository. OpenShift may expect certain files to exist in certain directories, which may require you to update your repository after creation.

Gears

small

Gears are the application containers running your code. For most applications, the small gear size provides plenty of resources. You can also [upgrade your plan](#) to get access to more gear sizes.

Cartridges

PHP 5.4 and MySQL 5.5

Applications are composed of cartridges - each of which exposes a service or capability to your code. All applications must have a web cartridge.

Scaling

No scaling



This application may require additional work to scale. Please see the application's documentation for more information.

OpenShift automatically routes web requests to your web gear. If you allow your application to scale, we'll set up a load balancer and allocate more gears to handle traffic as you need it.

List of the Applications

The screenshot shows the OpenShift Online web interface. At the top, there is a header bar with a back/forward button, a search bar containing the URL <https://openshift.redhat.com/app/console>, and tabs for "OpenShift Platform as a..." and "Applications | OpenS...". Below the header is a menu bar with Chinese options: 文件(F), 编辑(E), 查看(V), 收藏夹(A), 工具(I), 帮助(H). On the right side of the header, there are links for "Upgrade Plan" and a user account "lidan_gz@163.com". The main content area has a dark header with the "OPENSHIFT ONLINE" logo, "Applications", "Settings", "Help", and "OpenShift Hub" buttons. The main content area displays the "Applications" section, showing 1 of 3 applications named "gyblog". This application is described as using MySQL 5.5, PHP 5.4. To the right of the application list, there is a sidebar with links: "Add a collaborator", "Use your own domain name", and "Create a scalable application". Below these links, there is information about using the command line with the "rhc" client, including links for "Access logs", "Save and restore backups", and "Connect directly to internal services". A "Add Application..." button is located at the bottom left of the application list area.

Overview of WordPress App

The screenshot shows the OpenShift Online application console interface. At the top, the URL is https://openshift.redhat.com/app/console. The main header displays "OPENSHIFT ONLINE" with navigation links for Applications, Settings, and Help. A user profile is shown on the right.

Application Overview:

- Name:** gyblog-lidangz.rhcloud.com
- Status:** Started (1 gear)
- Created:** 31 minutes ago in domain lidangz and the aws-us-east-1 region

Cartridges:

Cartridge	Status	Gears	Storage
PHP 5.4	Started	1 small	1 GB
MySQL 5.5			

Source Code: ssh://555c98b8e0b8cd235f000031
Pass this URL to 'git clone' to copy the repository locally.

Continuous Integration: Enable Jenkins

Tools and Support: Add phpMyAdmin 4.0

Actions: Delete this application... (button)

Browse the Marketplace, or see the list of cartridges you can add

100% 126

WordPress Blog

The screenshot shows a WordPress blog site titled "2015 Guiyang Training Course". The main content area displays a single post titled "Hello world!" with the text "Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!". Below the post are the date "20th May 2015" and "1 Comment". The sidebar on the left contains several widgets: "RECENT POSTS" (listing "Hello world!"), "RECENT COMMENTS" (listing "Mr WordPress on Hello world!"), "ARCHIVES" (listing "May 2015"), "CATEGORIES" (listing "Uncategorised"), and "META" (listing "Log in", "Entries RSS", "Comments RSS", and "WordPress.org"). The browser's address bar shows the URL "https://gyblog-lidangz.rhcloud.com/".

2015 Guiyang Training Course

Just another WordPress site

Search ...

RECENT POSTS

Hello world!

RECENT COMMENTS

Mr WordPress on Hello world!

ARCHIVES

May 2015

CATEGORIES

Uncategorised

META

Log in

Entries RSS

Comments RSS

WordPress.org

2015 Guiyang Training C... 2015 Guiyang Training C...

1027

Lab: Start with OpenShift Online

- Sign up a free account in
<https://www.openshift.com/>
- Login in to OpenShift web console, and create a WordPress application.
- Test your WordPress application.