Cloud and Web Application

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 deploying web applications on cloud
- Getting your hands dirty on web applications and cloud

Cloud and Web Application

Part 1: Introduction

Table of Contents

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

PREFACE

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

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?

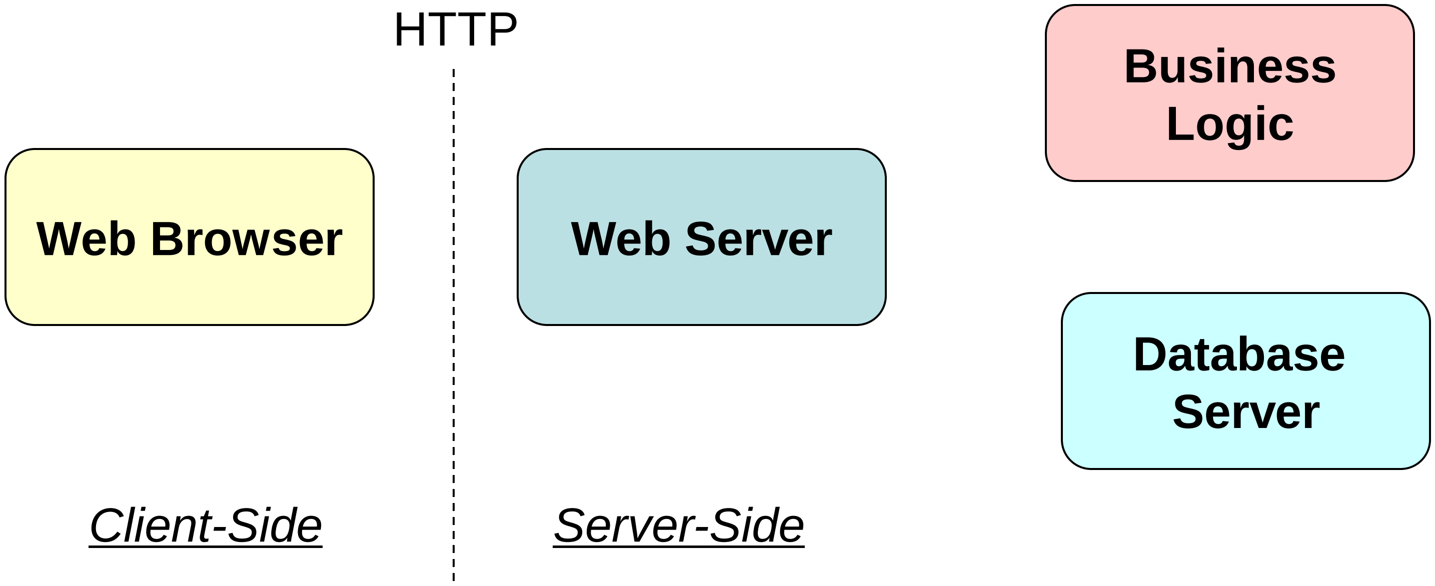
- “Who has my data?” Essentially, not you.
- Issues of trust. Many users do not trust other people, even within the same company, to keep their data safe and secure.
- Response time. While the actual execution of the app may be much quicker, user response time can be noticeably slower than a local app.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

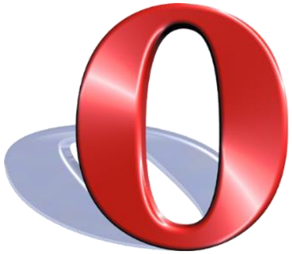
Different Browsers

Browsers driven by product differentiation

IE

Firefox

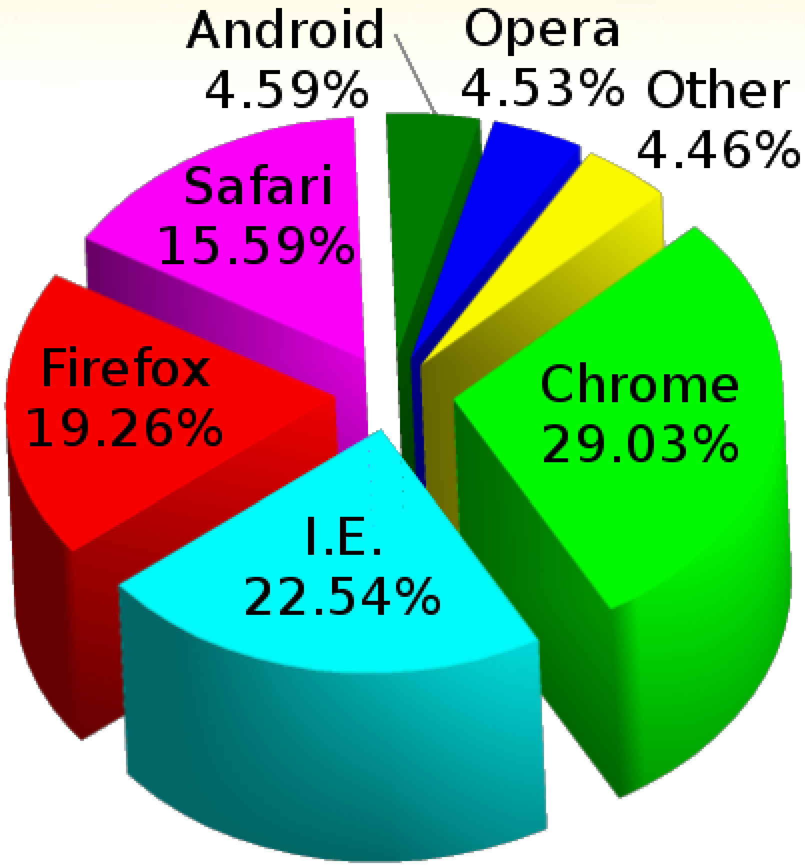
Chrome



Opera

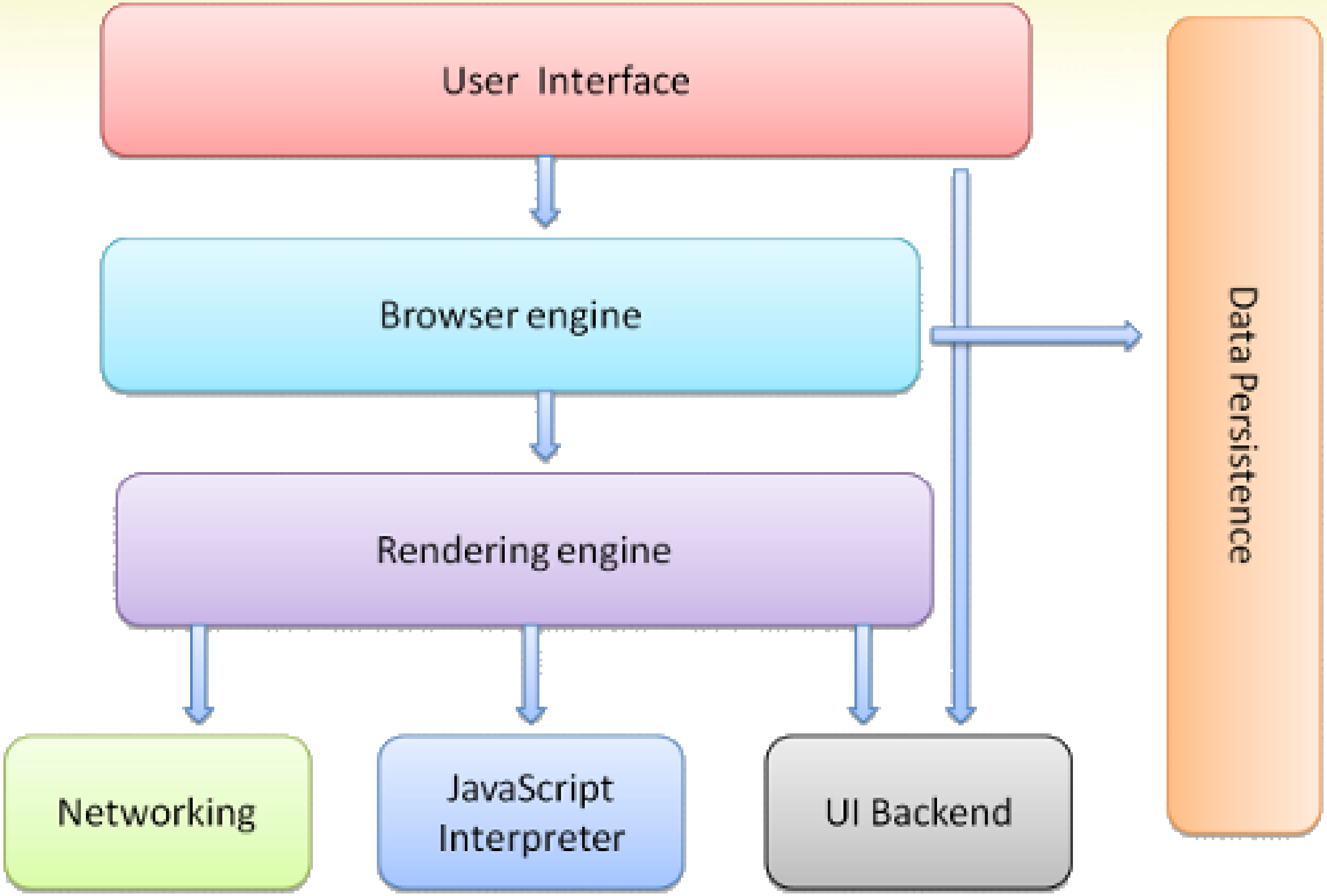
Safari

Browser's Market



Browser usage on Wikimedia
September 2012

Browser Structure

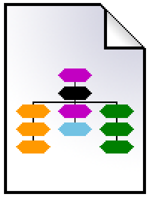


http://taligarsiel.com/Projects/howbrowserswork1.htm

Browsers components

Scripting engine:
interprets JavaScript

Rendering engine:
draws text, images, etc



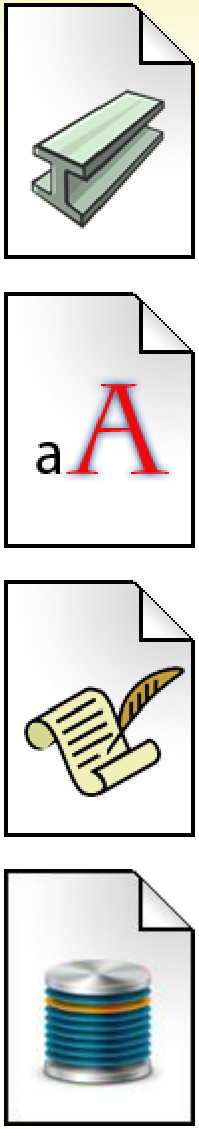
DOM:
Document Object Model

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

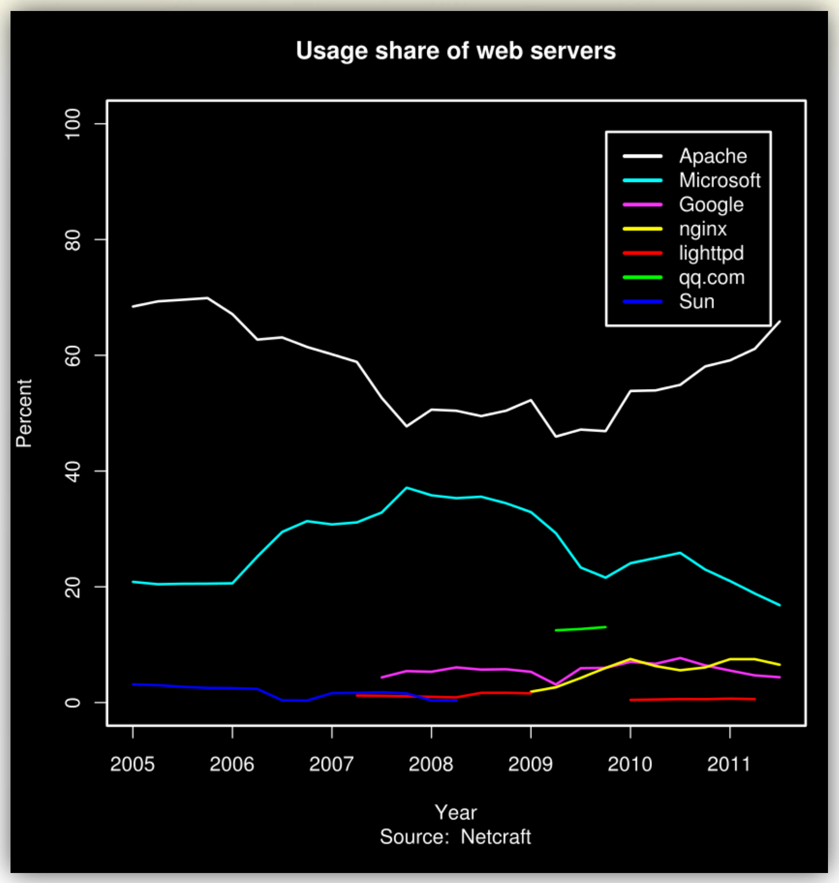
- Web servers are software products that handle web requests and manage connections
- These requests are redirected to other software products (ASP.NET, PHP, etc.), depending on the web server settings

Web Servers

Apache, IIS, Nginx, Lighttpd, etc.

Web Servers Market Share 2014

- Apache - 60.4%
- Nginx - 21.0%
- IIS -12.3%
- LiteSpeed -2.0%
- Google Server -5.09%
- Tomcat – 0.4%
- Lighttpd - 0.3%
- Node.js – 0.1%



W3Techs.com, 1 July 2014

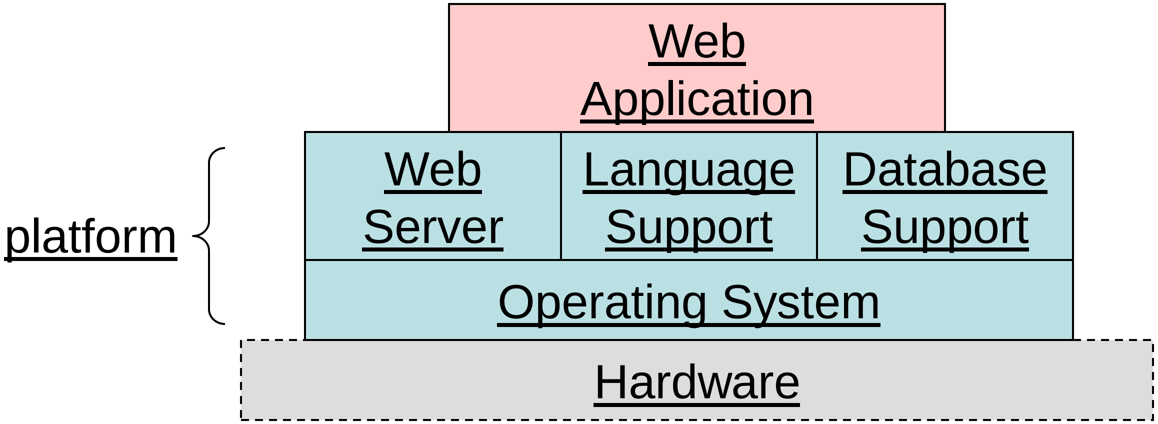
Database Server Responsibilities

- Relational Database Management System (RDBMS)

  NoSQL
- Maintains data storage for an application
  - 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

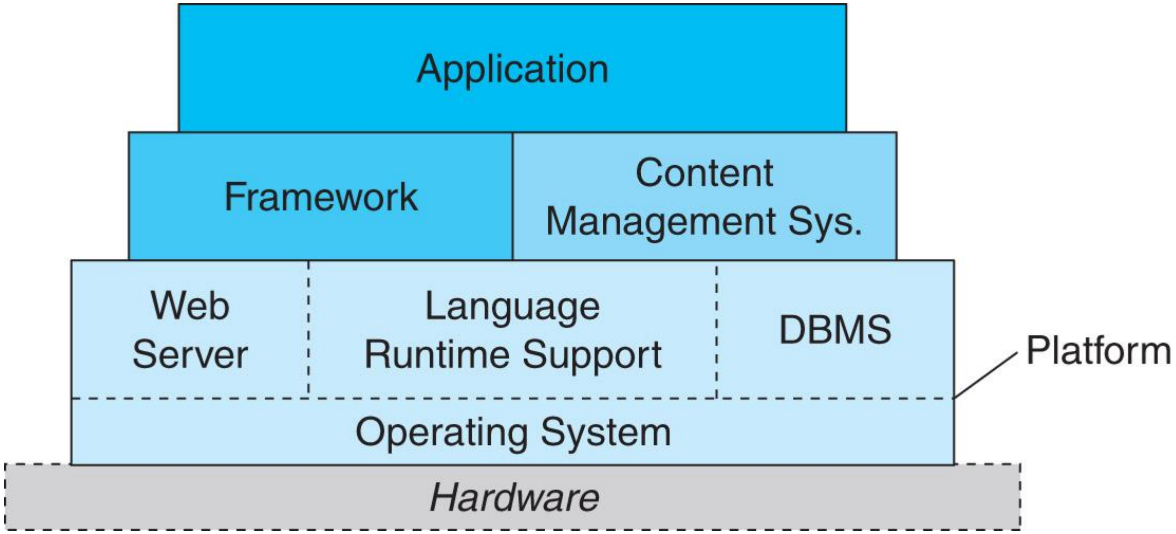


Web App Framework

- 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

Web Application Framework

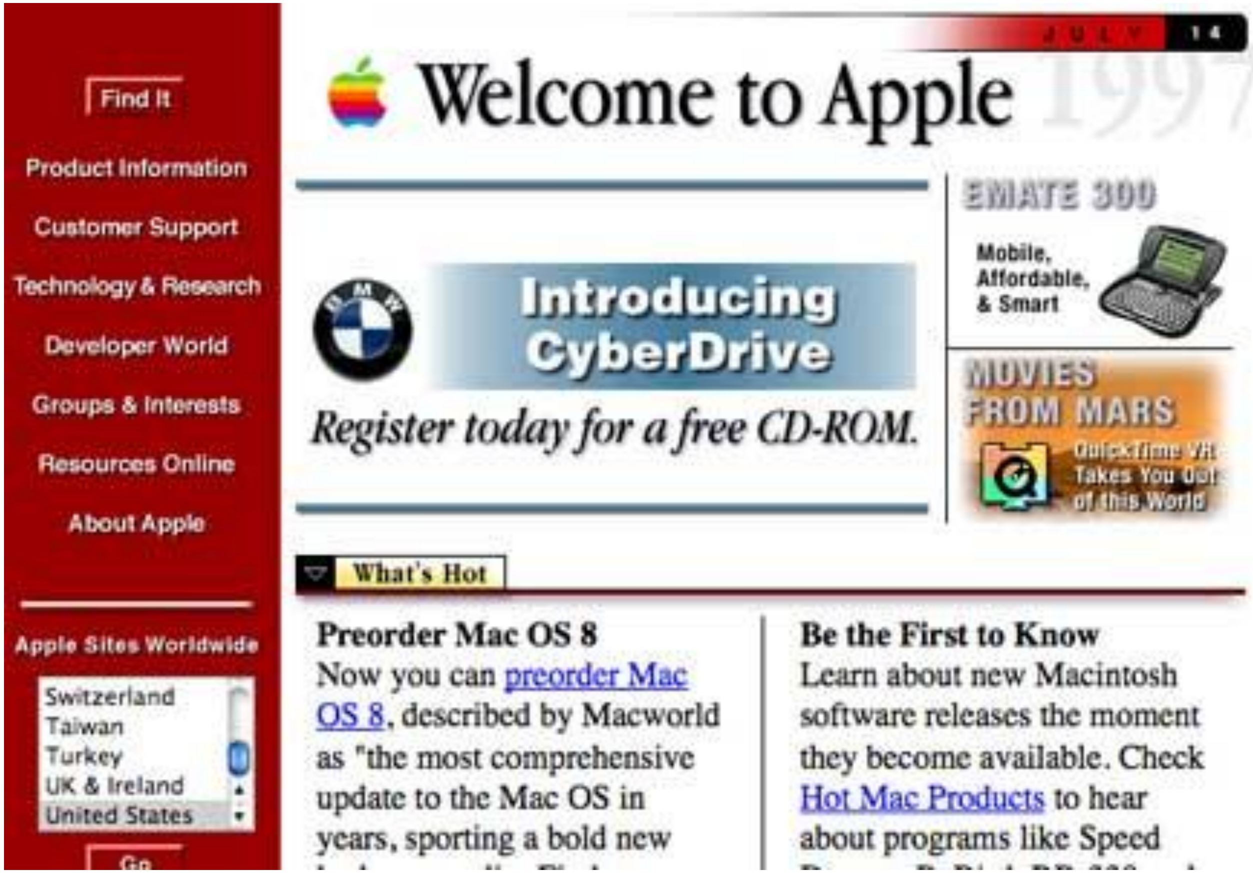
- Frameworks give application developers more powerful building blocks to work with



HISTORY OF WEB APPLICATION

The Good Old Days

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



Welcome to Apple

Introducing CyberDrive

Register today for a free CD-ROM.

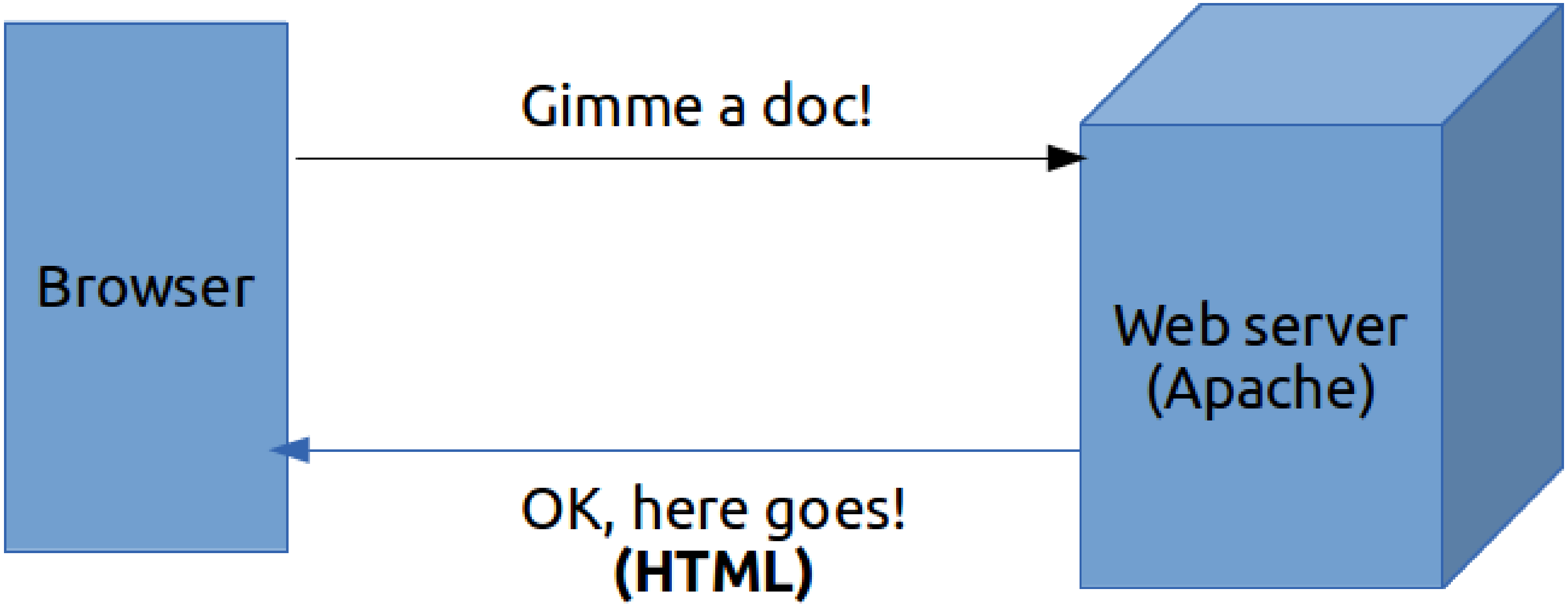
Preorder Mac OS 8

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

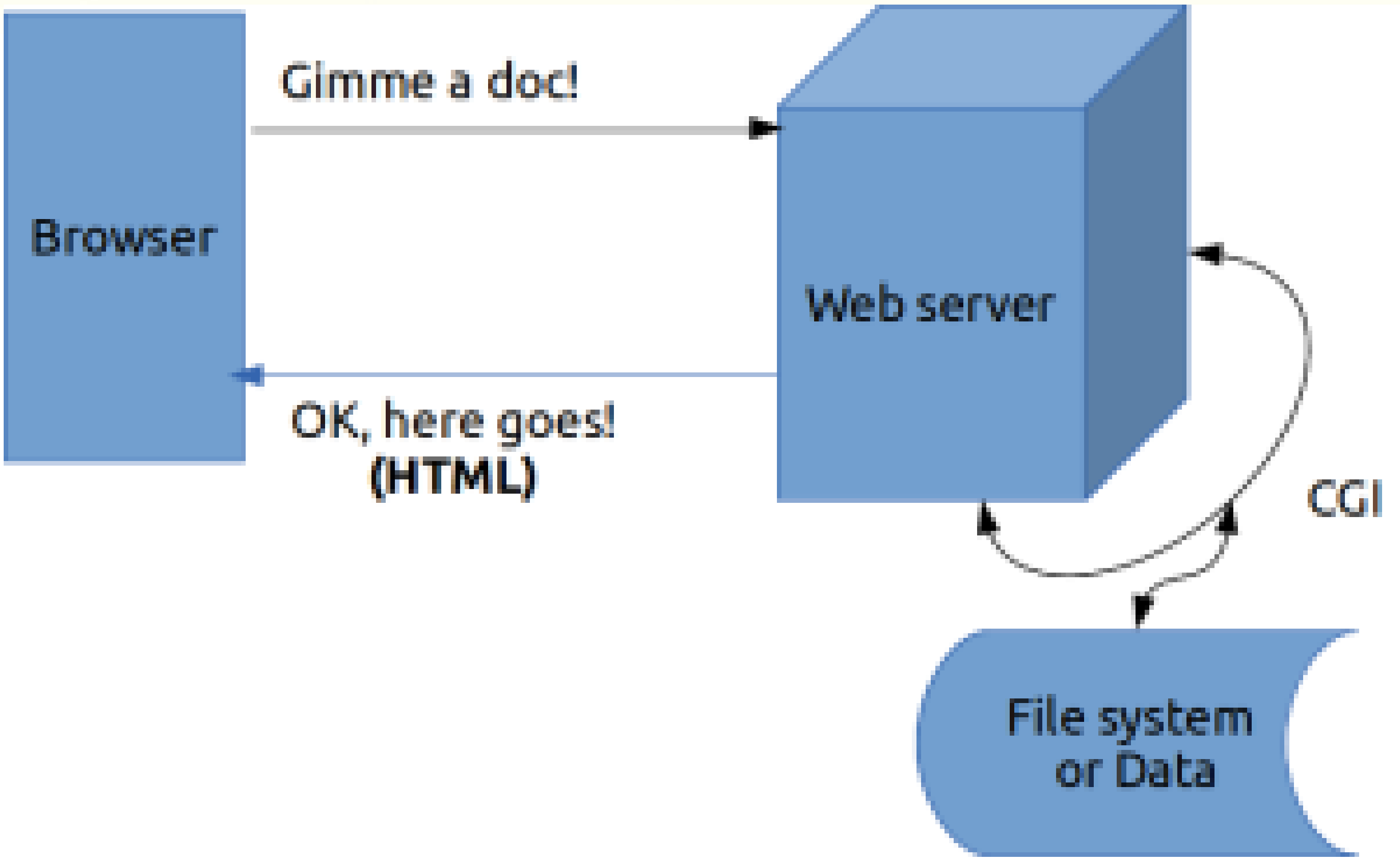
Be the First to Know

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

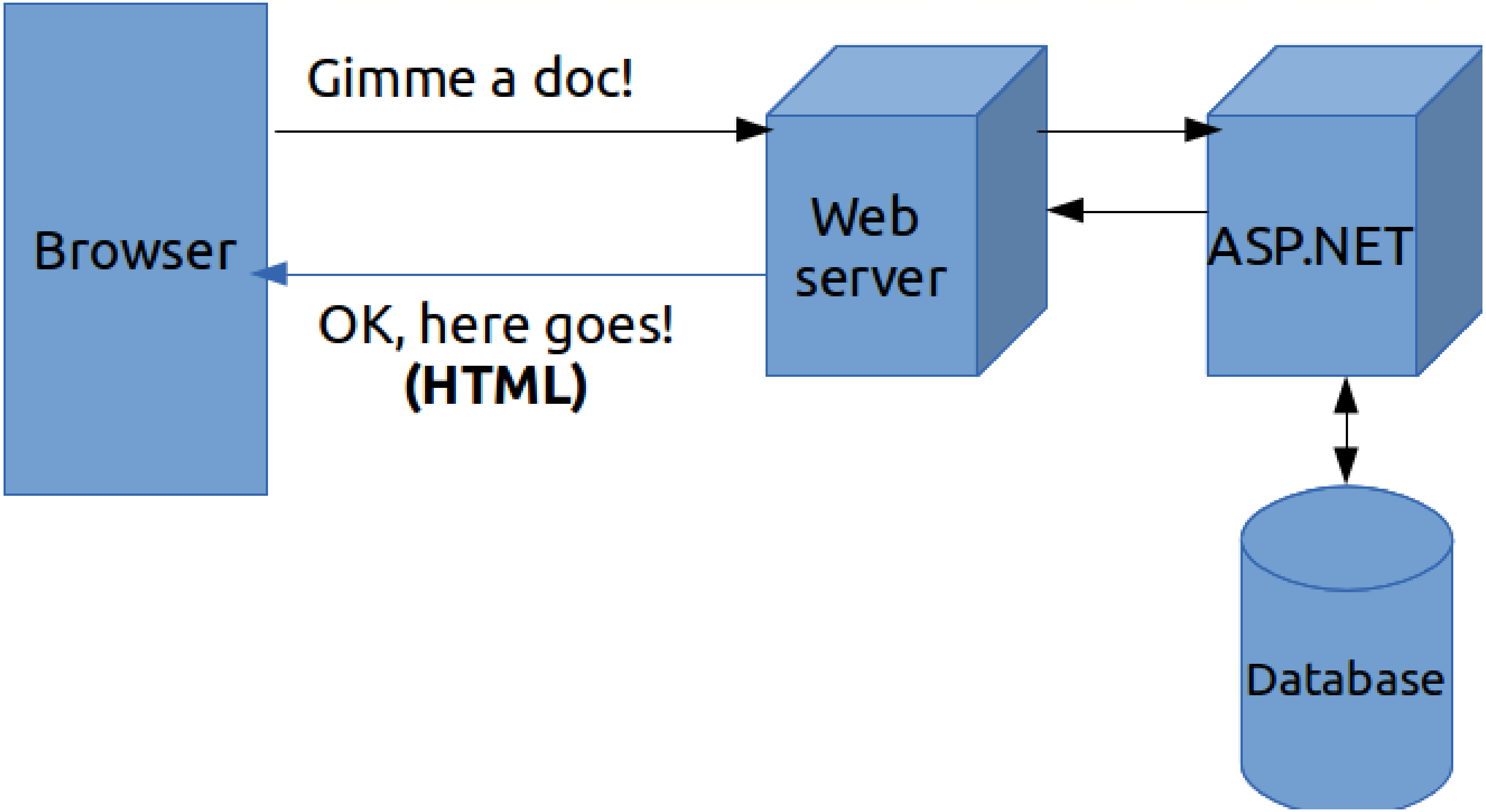
Static Web Site



CGI/Perl –> till 2005

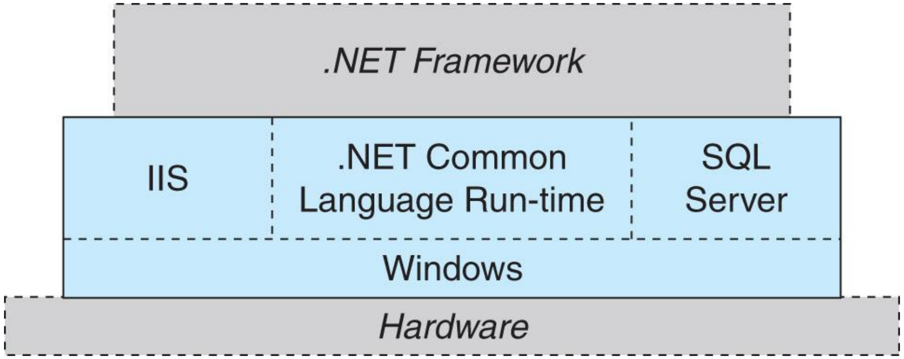


IIS & ASP


Gimme a doc!
Browser
Web server
ASP.NET
OK, here goes!
(HTML)
Database

Microsoft / .NET

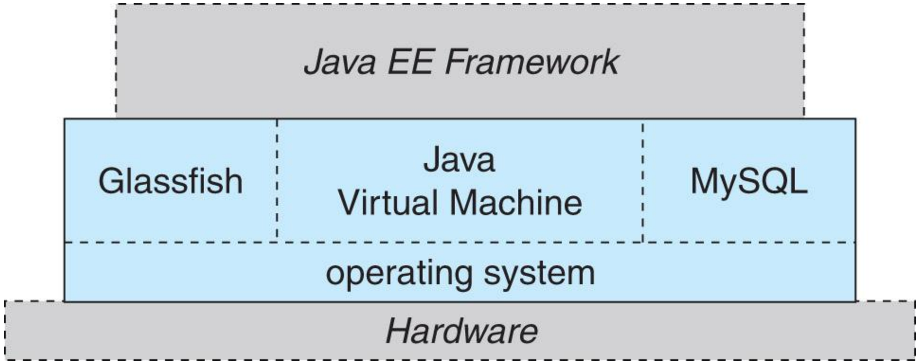
- All Microsoft products (licensed)



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

Java EE

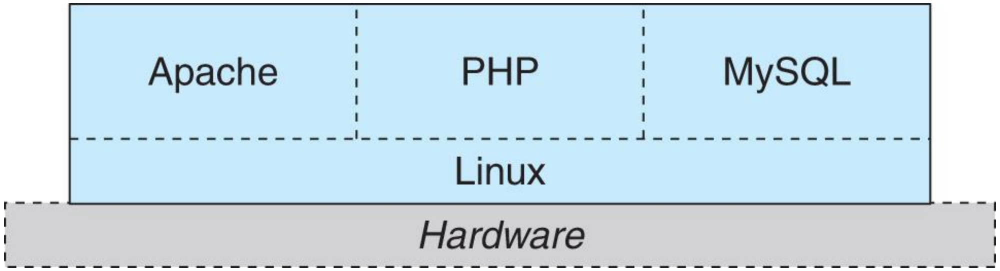
- Supports Java language development



- Supported by multiple operating systems
- Proprietary, free license

LAMP Stack

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

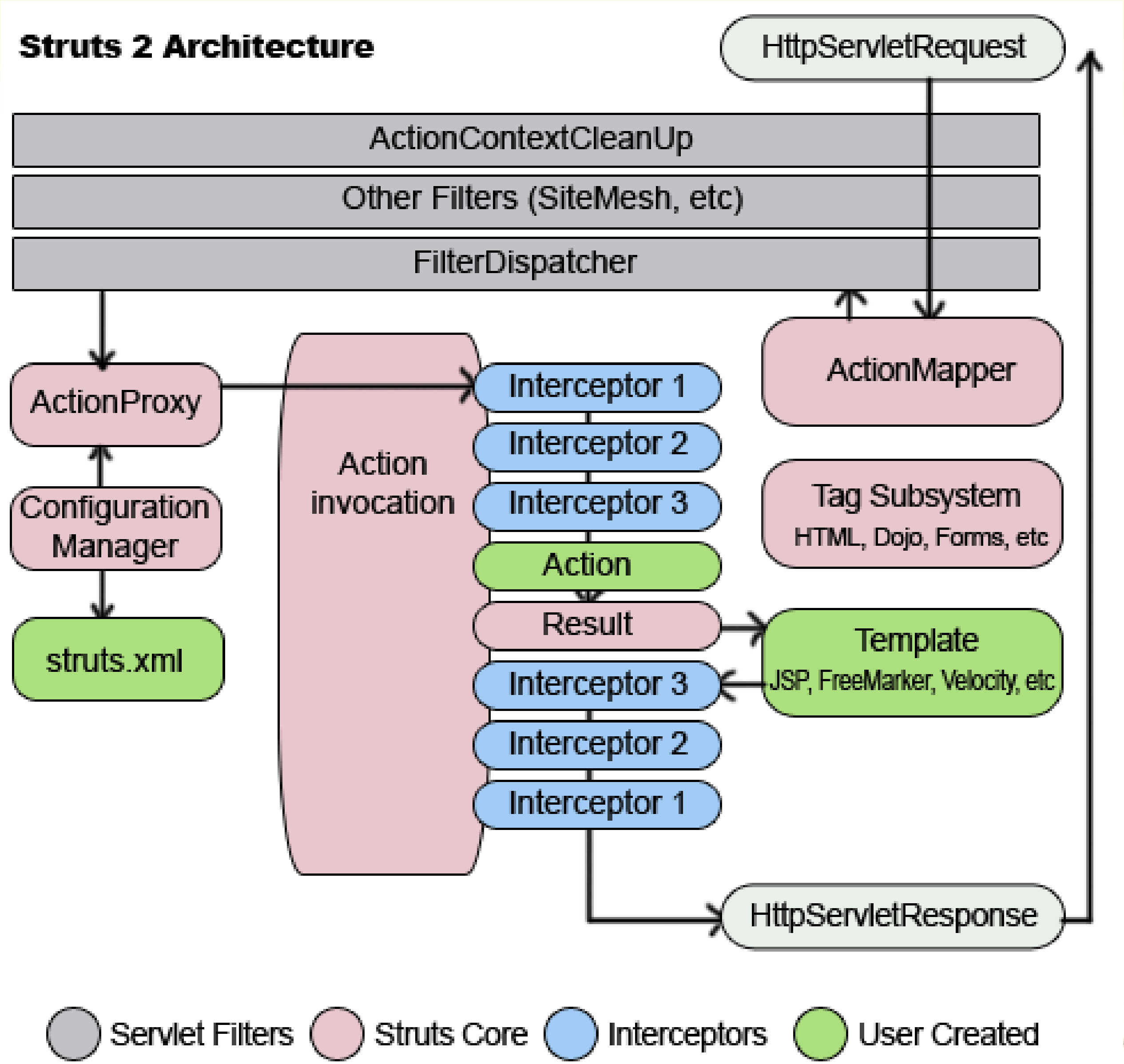


- The LAMP platform appeared in mid 1990's and has become very popular
- LAMP is open-source free software, which is one reason for its popularity

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

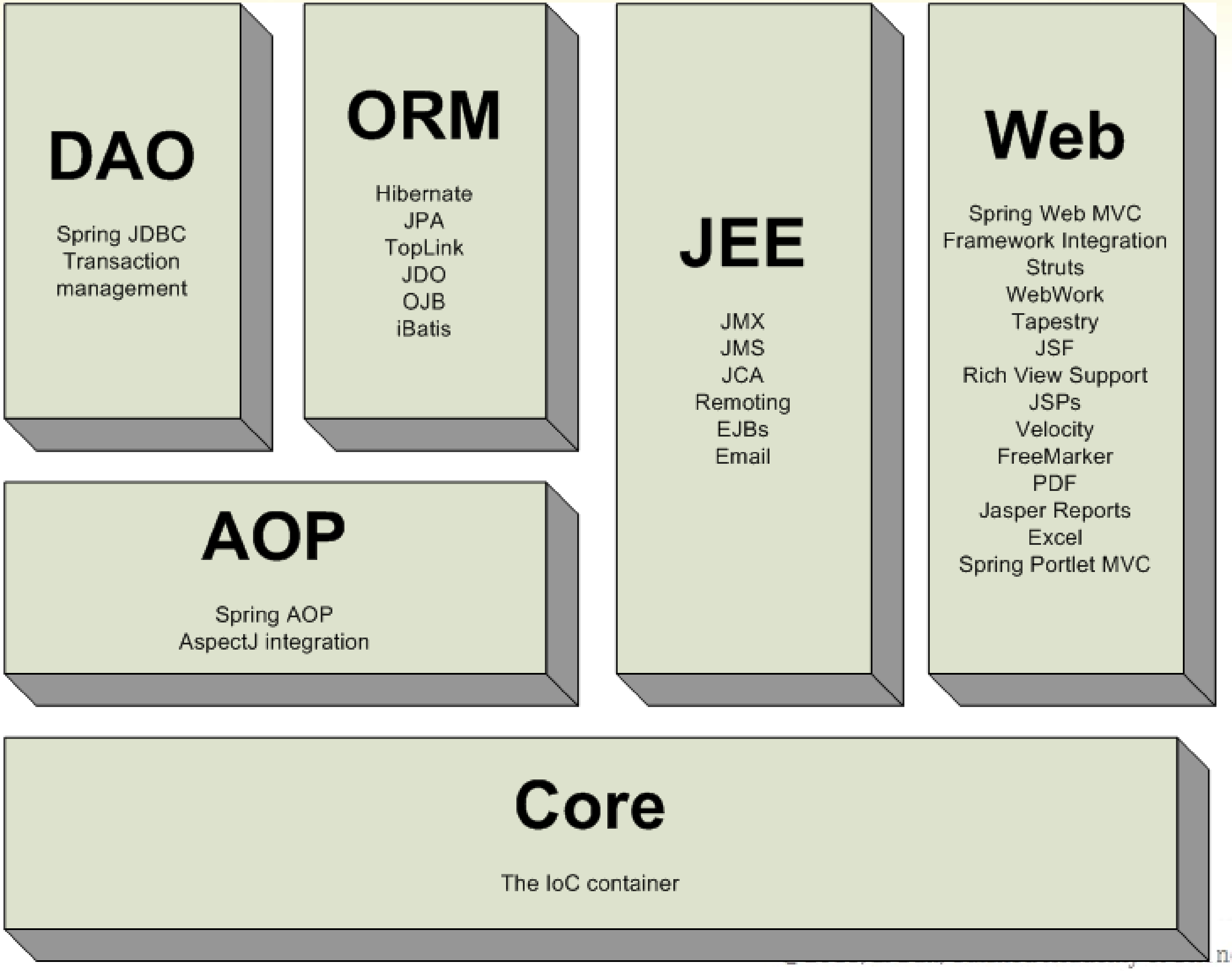
Struts Architecture



Spring

- Lightweight container and framework
  - Most of your code will be unaware of the Spring framework
  - Use only the parts you 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



DAO

Spring JDBC
Transaction management

ORM

Hibernate
JPA
TopLink
JDO
OJB
iBatis

JEE

JMX
JMS
JCA
Remoting
EJBs
Email

Web

Spring Web MVC
Framework Integration
Struts
WebWork
Tapestry
JSF
Rich View Support
JSPs
Velocity
FreeMarker
PDF
Jasper Reports
Excel
Spring Portlet MVC

AOP

Spring AOP
AspectJ integration

Core

The IoC container

nces

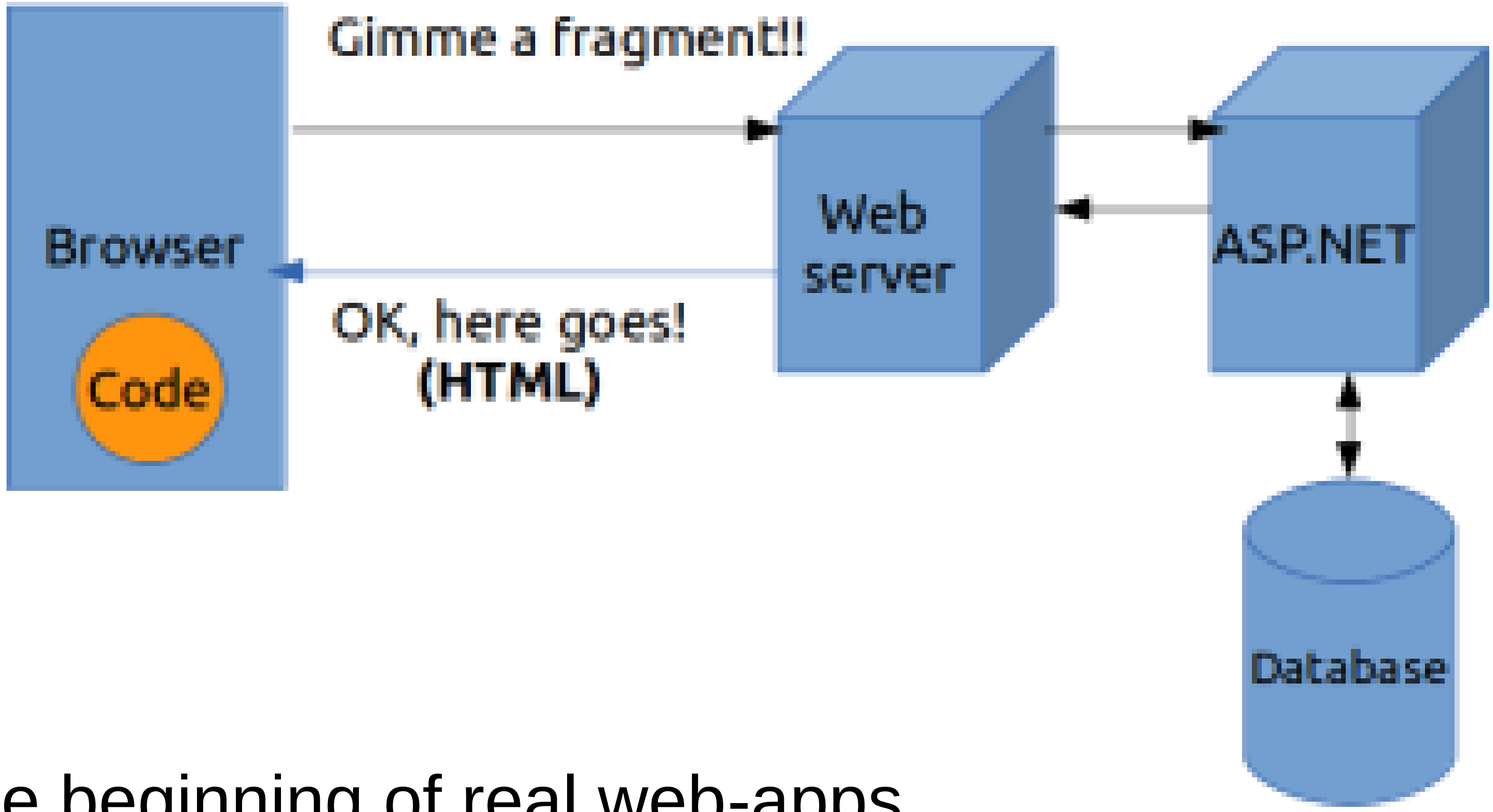
Hibernate

- 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?

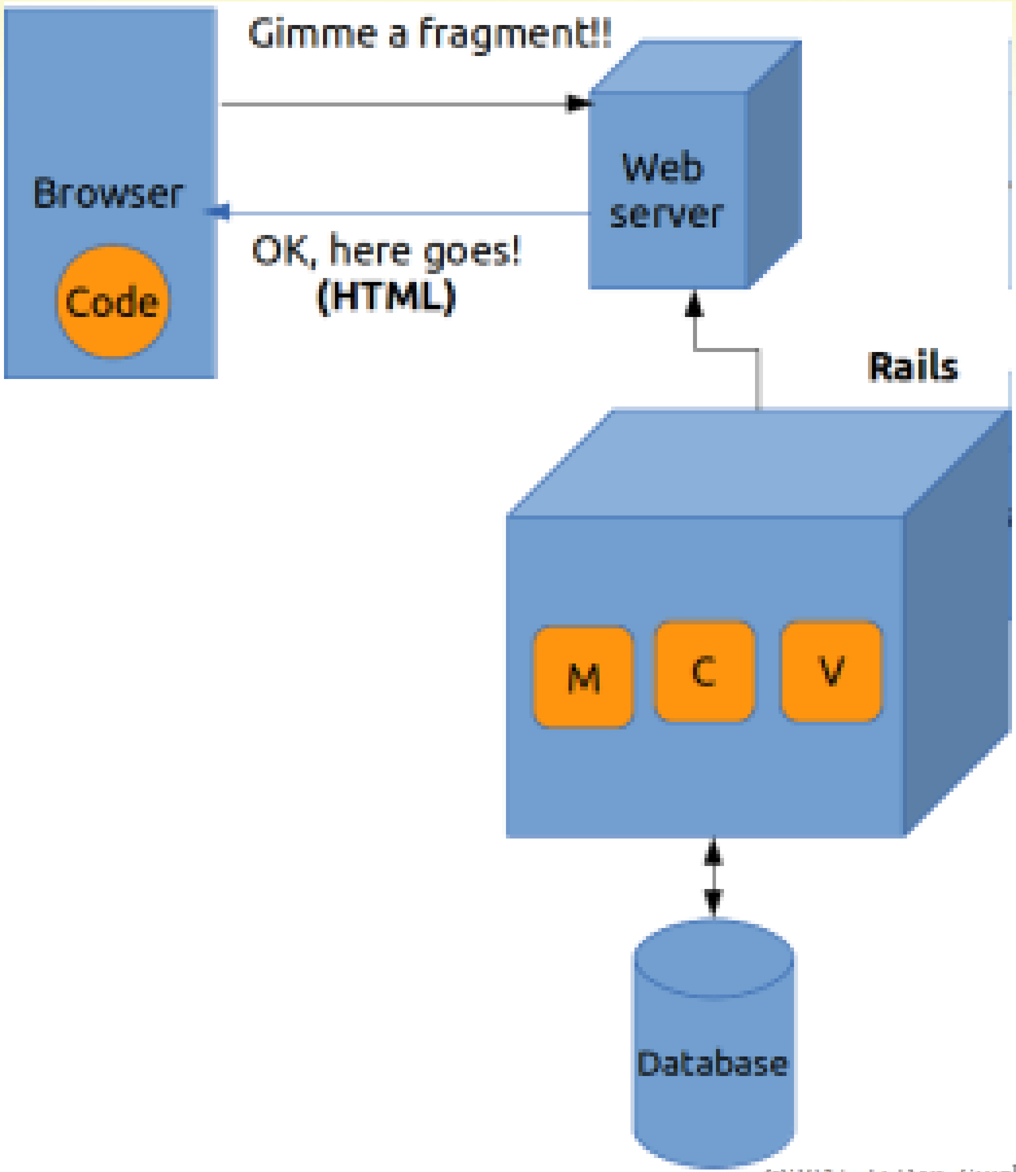
- Simpler data persistence
  - Automatically handles mapping SQL to Object and vice versa
  - Automatic creation of database schemas
  - Automatic updating of database schemas
    - Add a field to an object; Hibernate converts your existing database for you.
  - Provides search functionality
- Simpler 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)



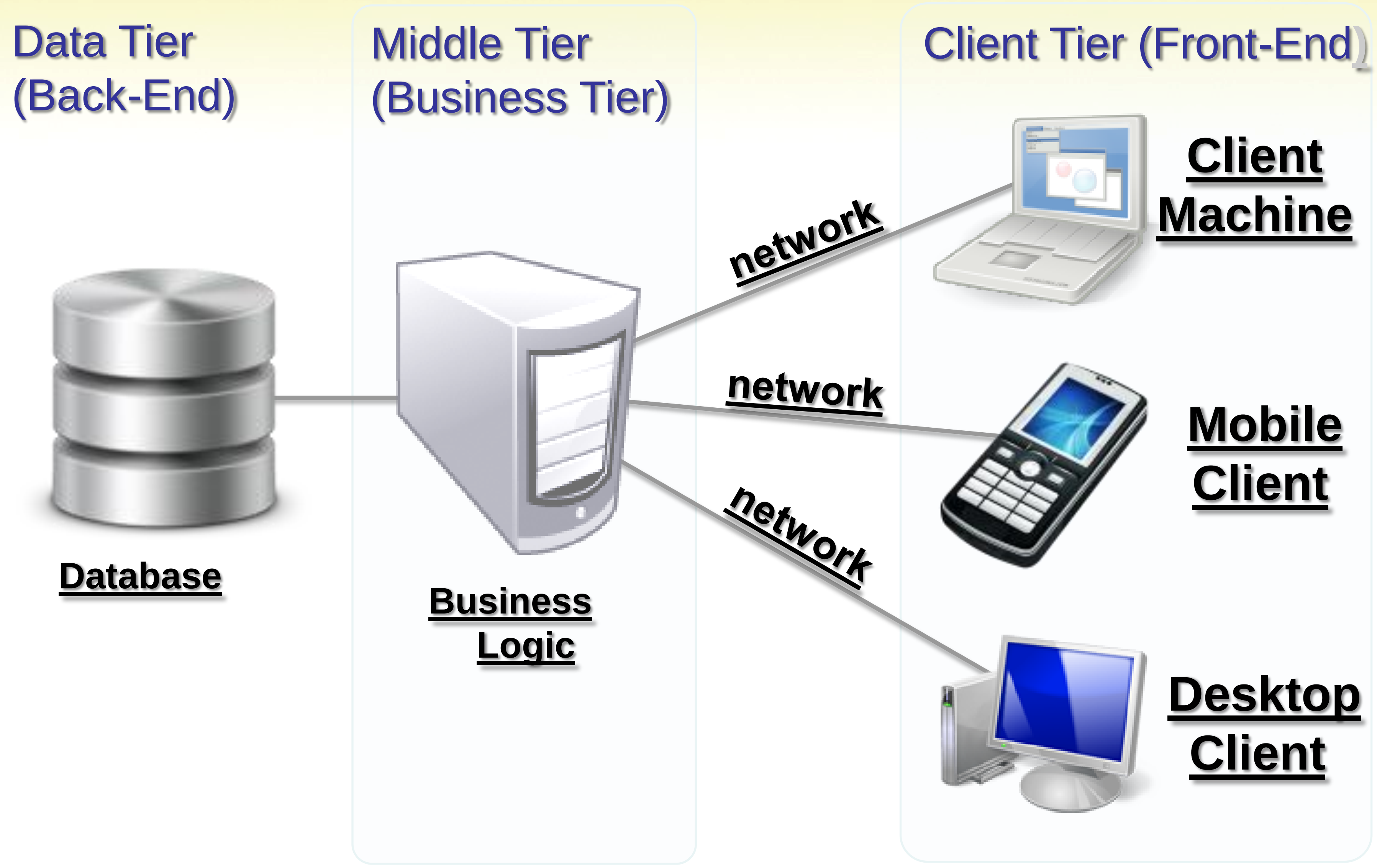
Ruby On Rails

- Ruby: a dynamically typed object-oriented programming language
- Rails: a webapp 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)

Middle Tier (Business Tier)

Client Tier (Front-End)

Database

Business Logic

network

network

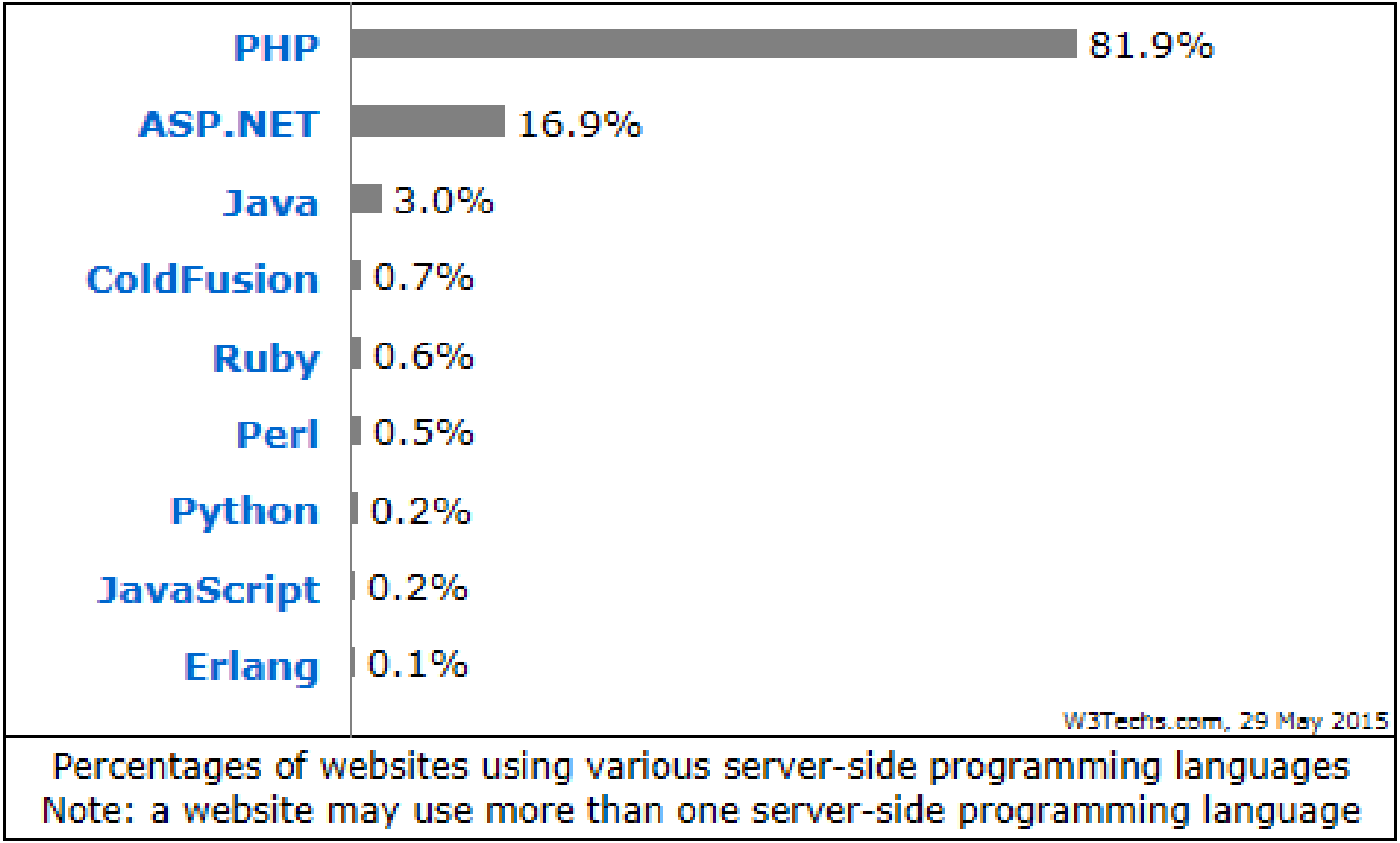
network

Client Machine

Mobile Client

Desktop Client

Market Shares of Server Languages



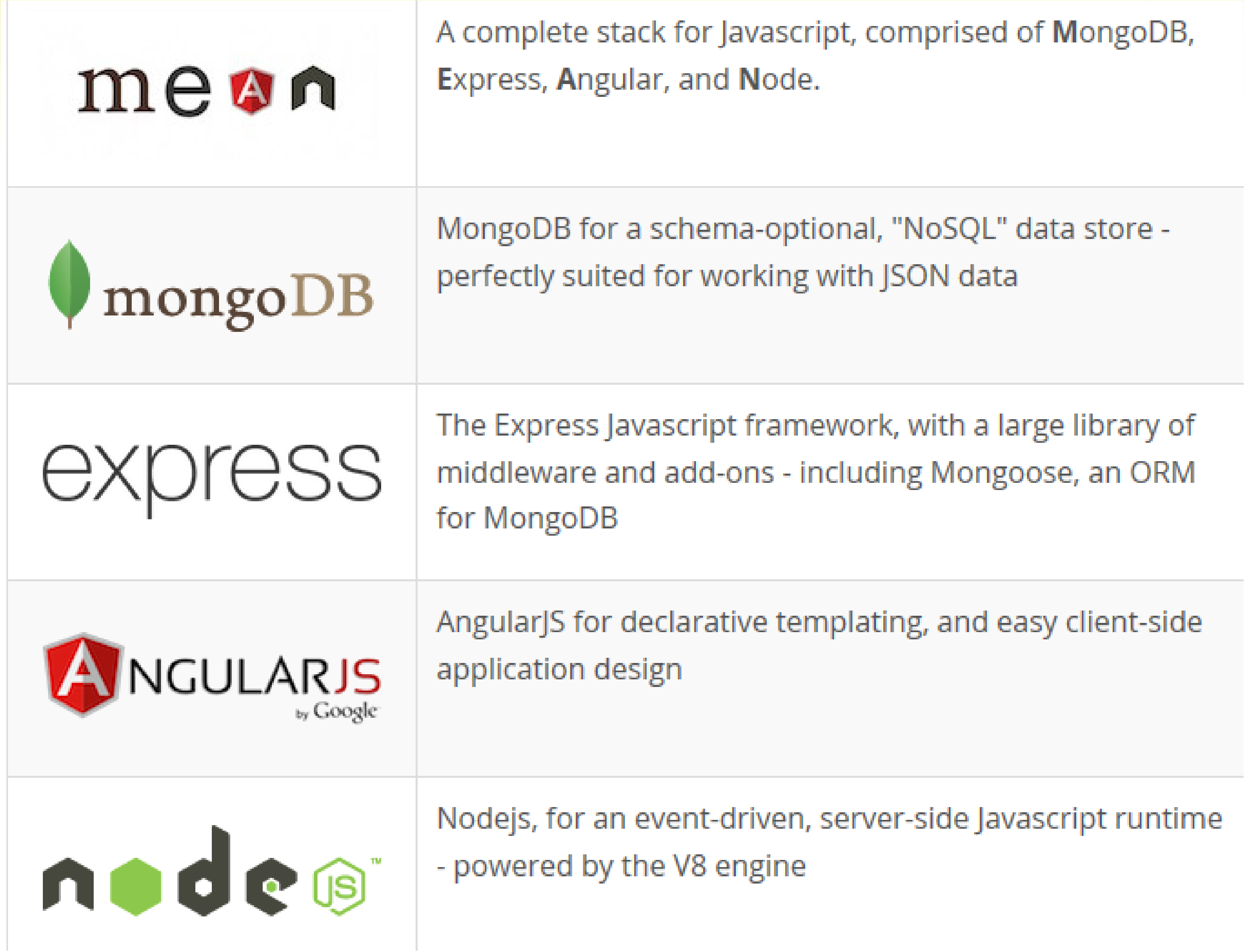
W3Techs.com, 29 May 2015

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



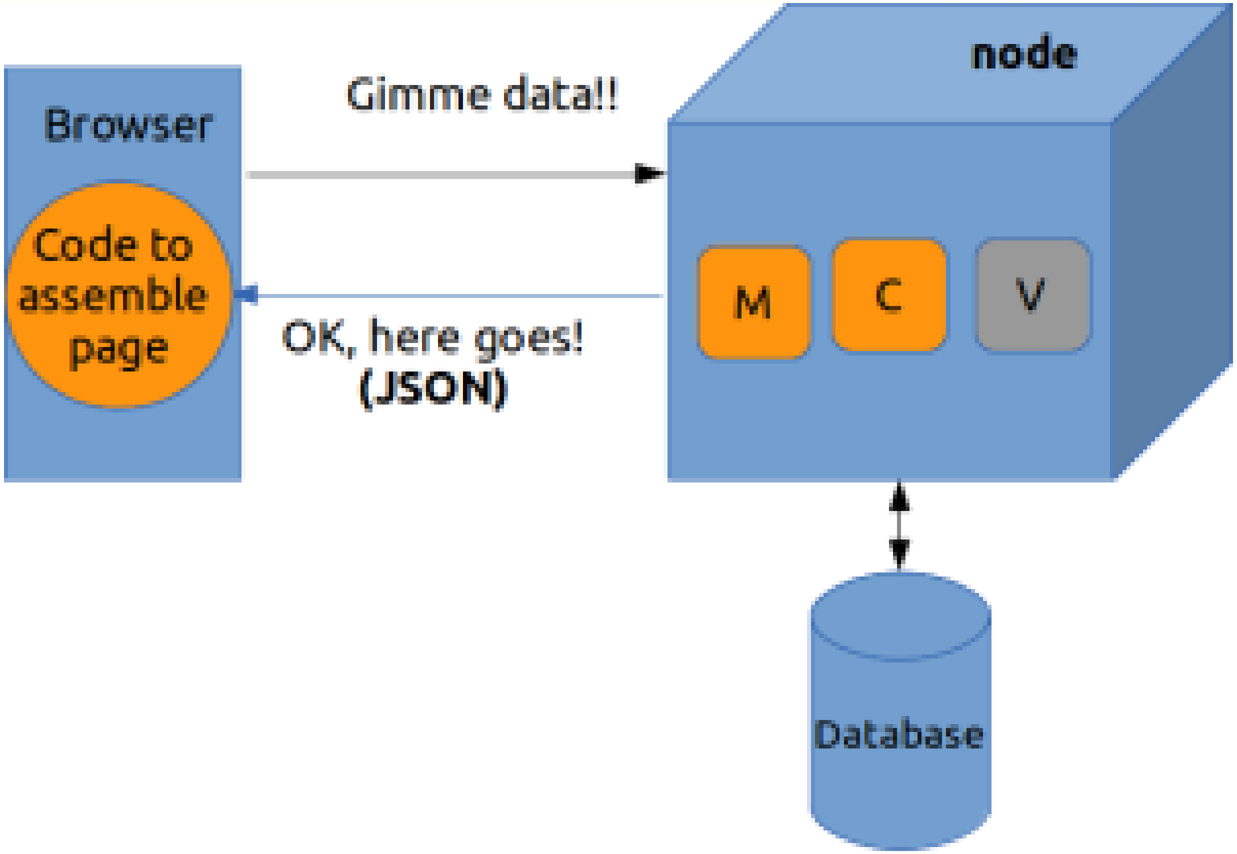
mean	A complete stack for Javascript, comprised of **M**ongoDB, **E**xpress, **A**ngular, and **N**ode.
mongoDB	MongoDB for a schema-optional, "NoSQL" data store - perfectly suited for working with JSON data
express	The Express Javascript framework, with a large library of middleware and add-ons - including Mongoose, an ORM for MongoDB
ANGULARJS by Google	AngularJS for declarative templating, and easy client-side application design
node JS™	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.”

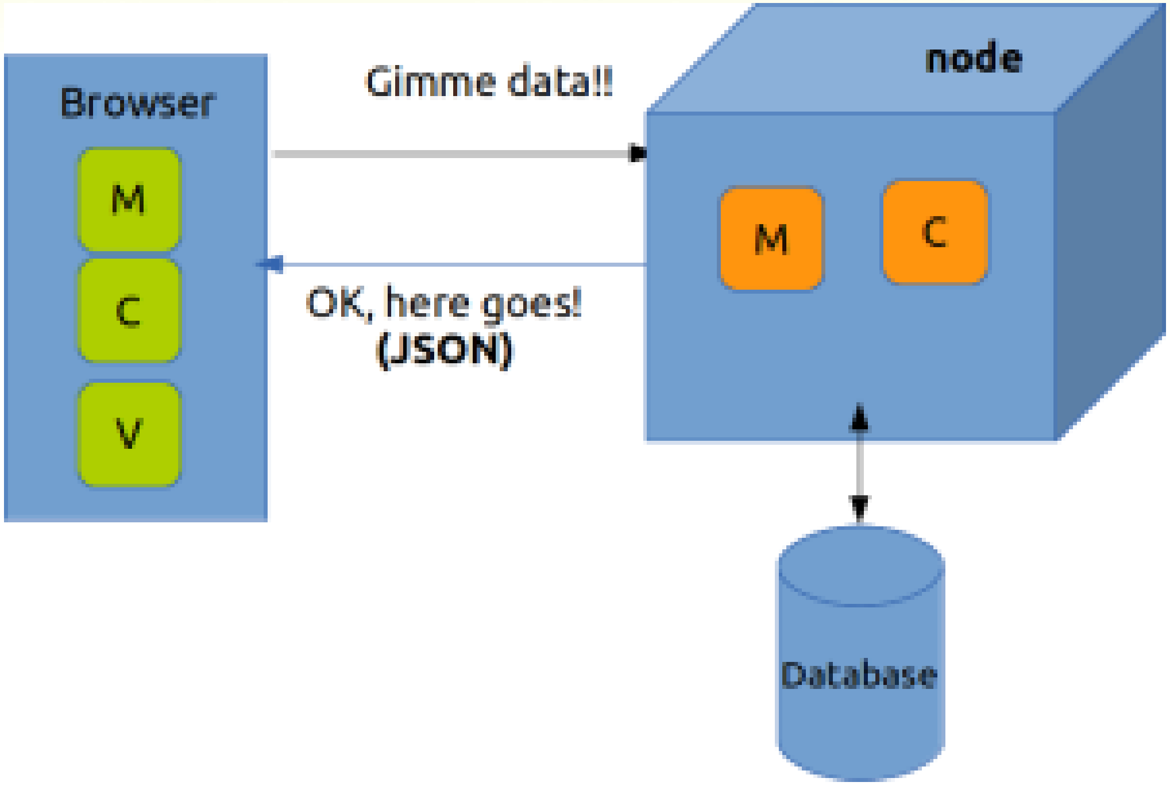
- nodejs.org

Node.js

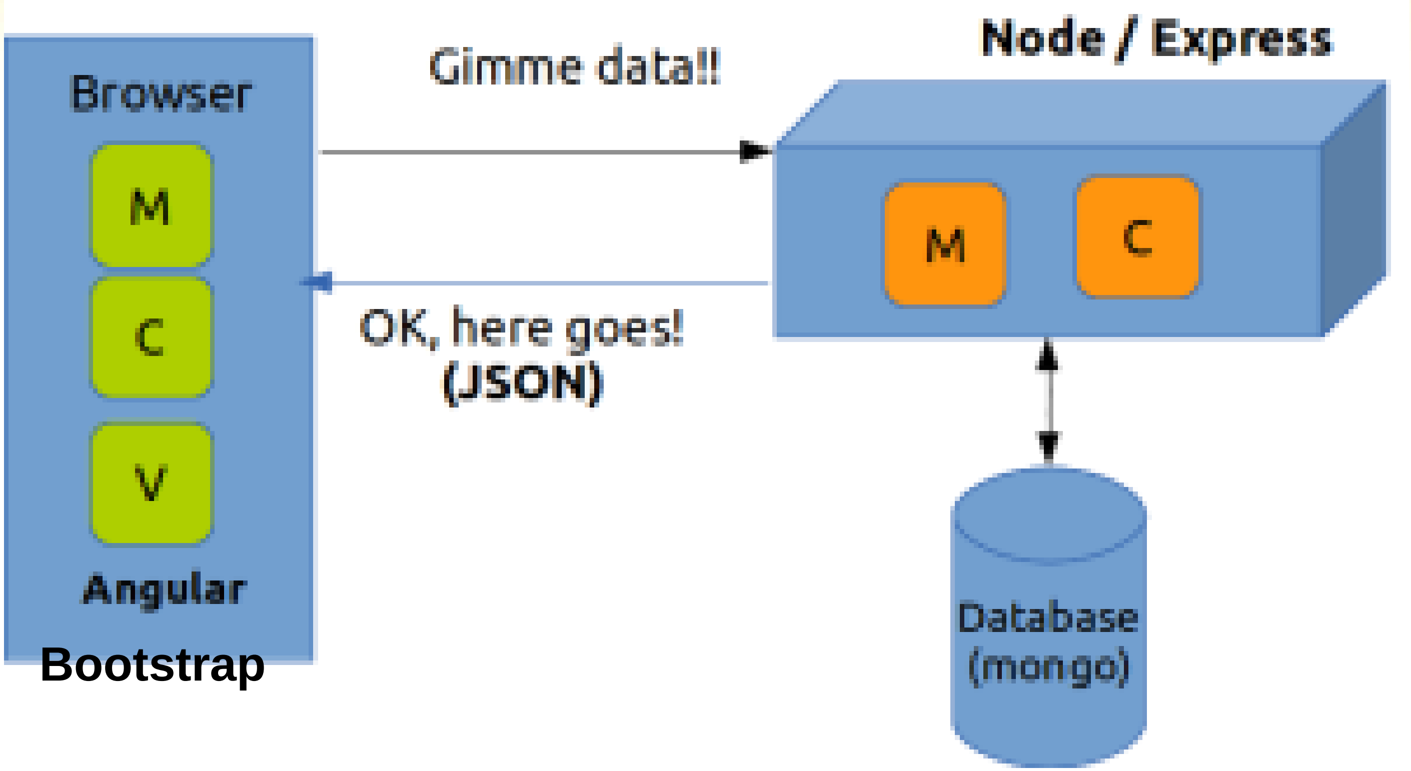

node
Browser
Gimme data!!
Code to assemble page
OK, here goes! (JSON)
M
C
V
Database

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

Google
app engine
HTML 5
coding
PHP
Java
Python
cloud
computing
$
Windows Azure
C#
Ruby
amazon
web services™

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**.

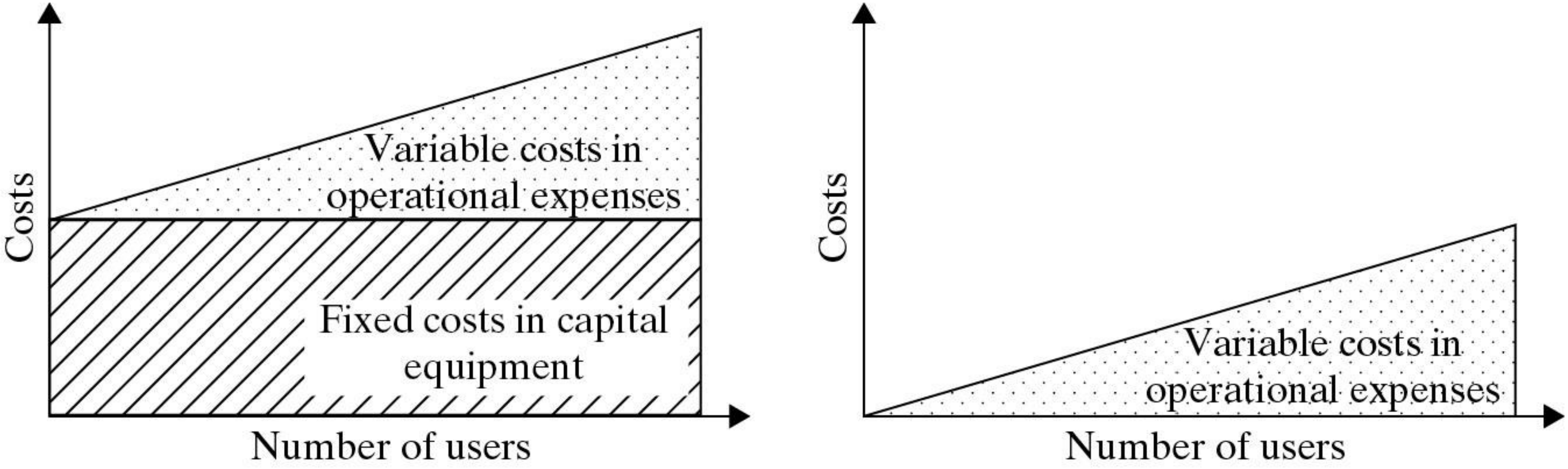
Before the cloud

- 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

How the Cloud Works?

- In the cloud everyone consumes a portion of the shared computing resources
  - CPU, memory, storage, IO, networking, etc.
- If you 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

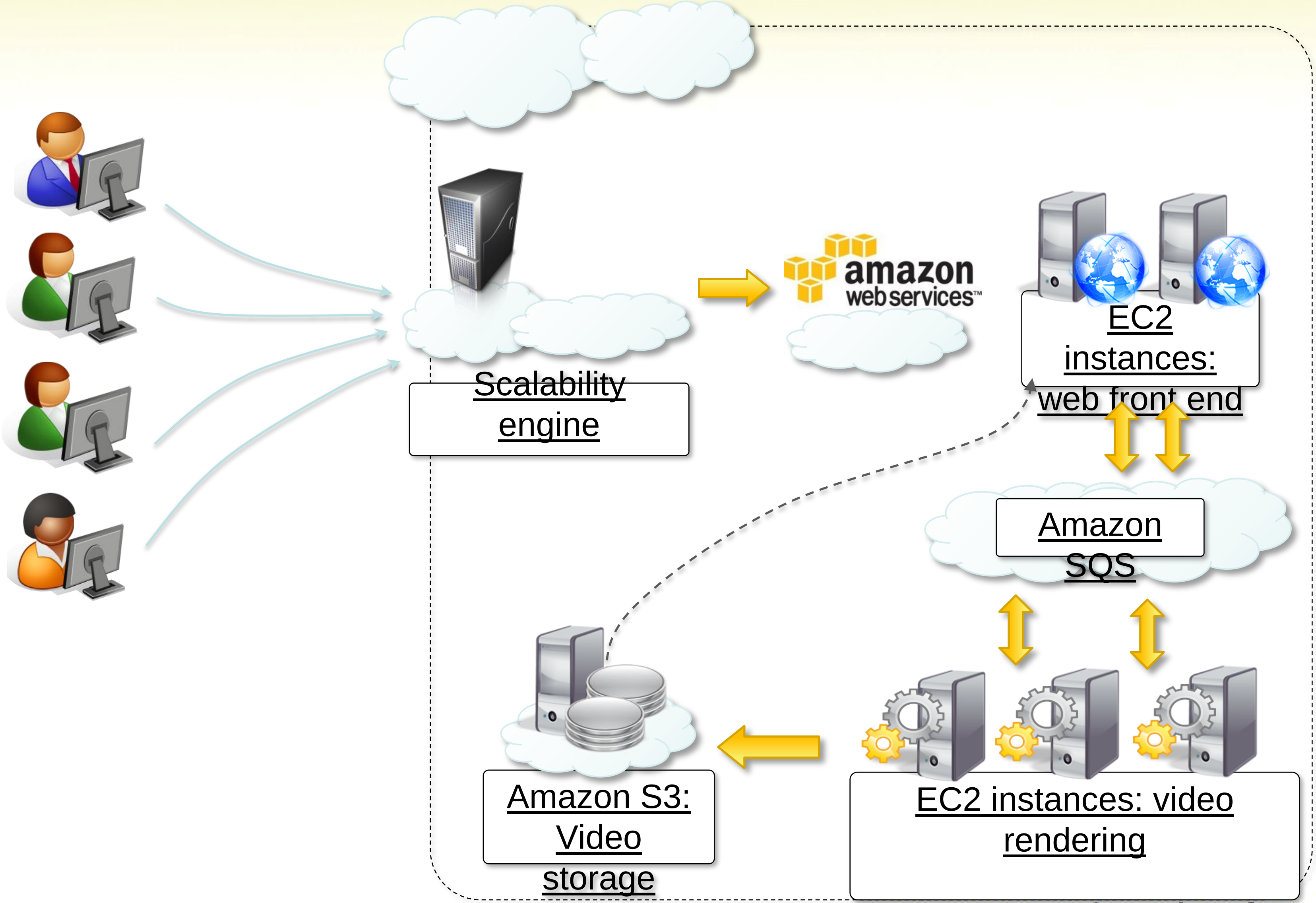
Cost-Effective of Cloud vs. Traditional


Costs
Variable costs in operational expenses
Fixed costs in capital equipment
Number of users
Costs
Variable costs in operational expenses
Number of users

(a) Traditional IT cost model

(b) Cloud computing cost model

Cloud Application



Scalability engine

EC2 instances: web front end

Amazon SQS

Amazon S3: Video storage

EC2 instances: video rendering

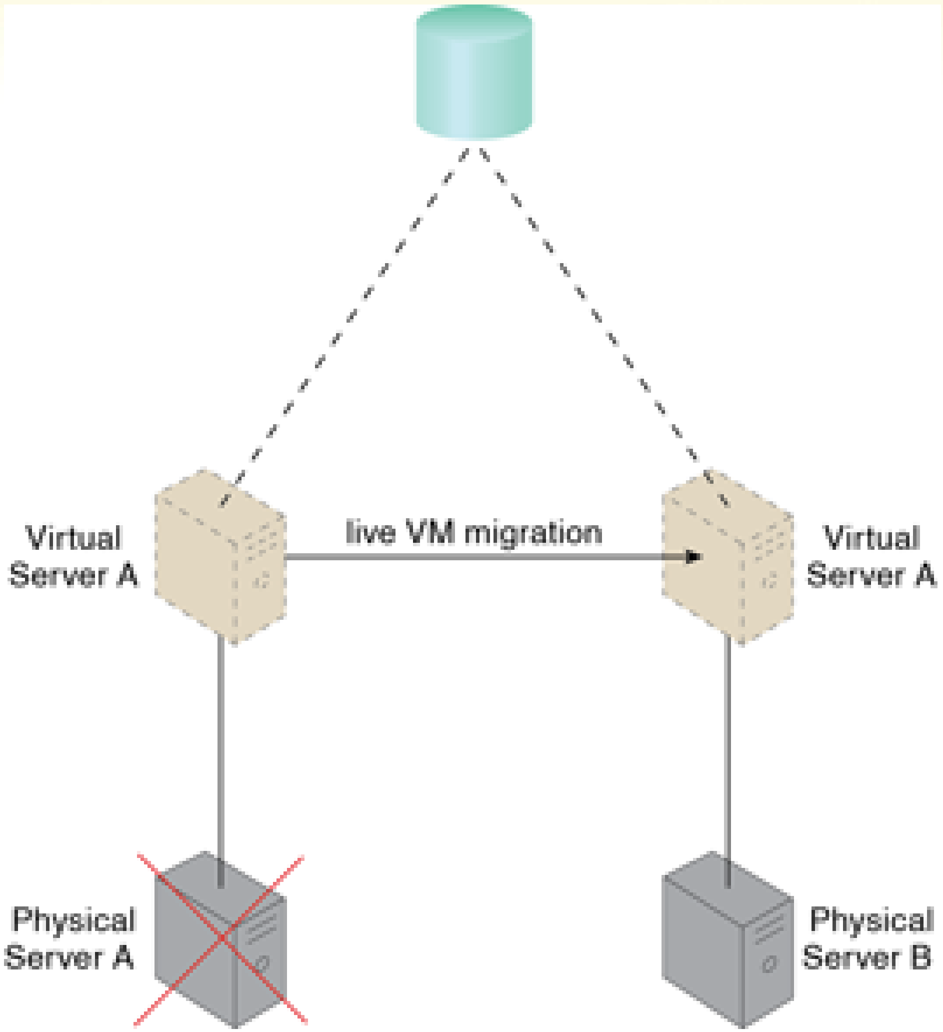
@2015, Li Dan, Guizhou Academy of Sciences

Essential Characteristics of Cloud

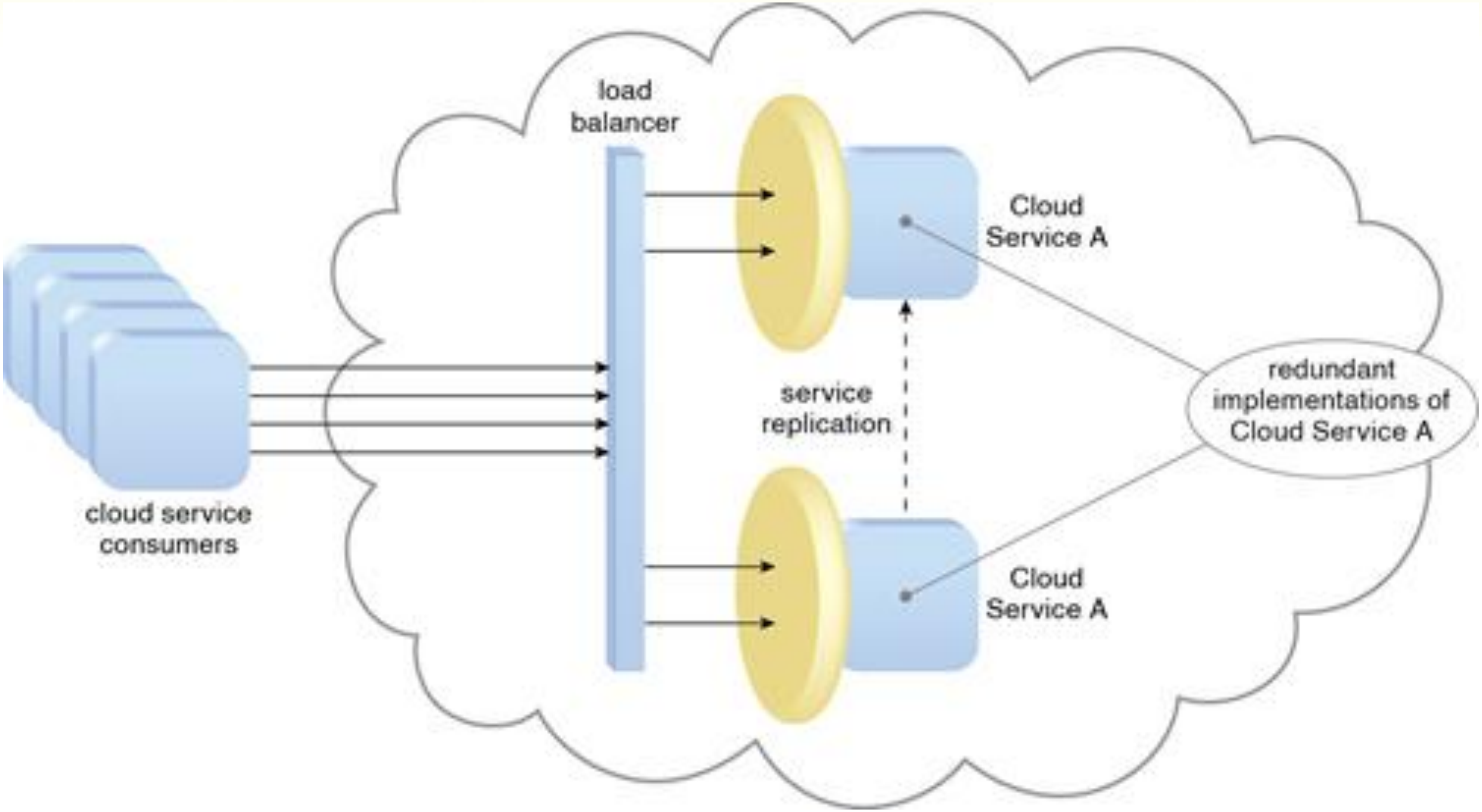
1. Broad network access
   - ➢ You can access the cloud from anywhere
2. Resource pooling
   - ➢ You work with virtual machines that could be hosted anywhere
3. Rapid elasticity
   - ➢ Easily go from 5 servers to 50 or from 50 servers to 5
4. On-demand self-service
   - ➢ You get elasticity automatically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

Source: NIST Working Definition of Cloud Computing

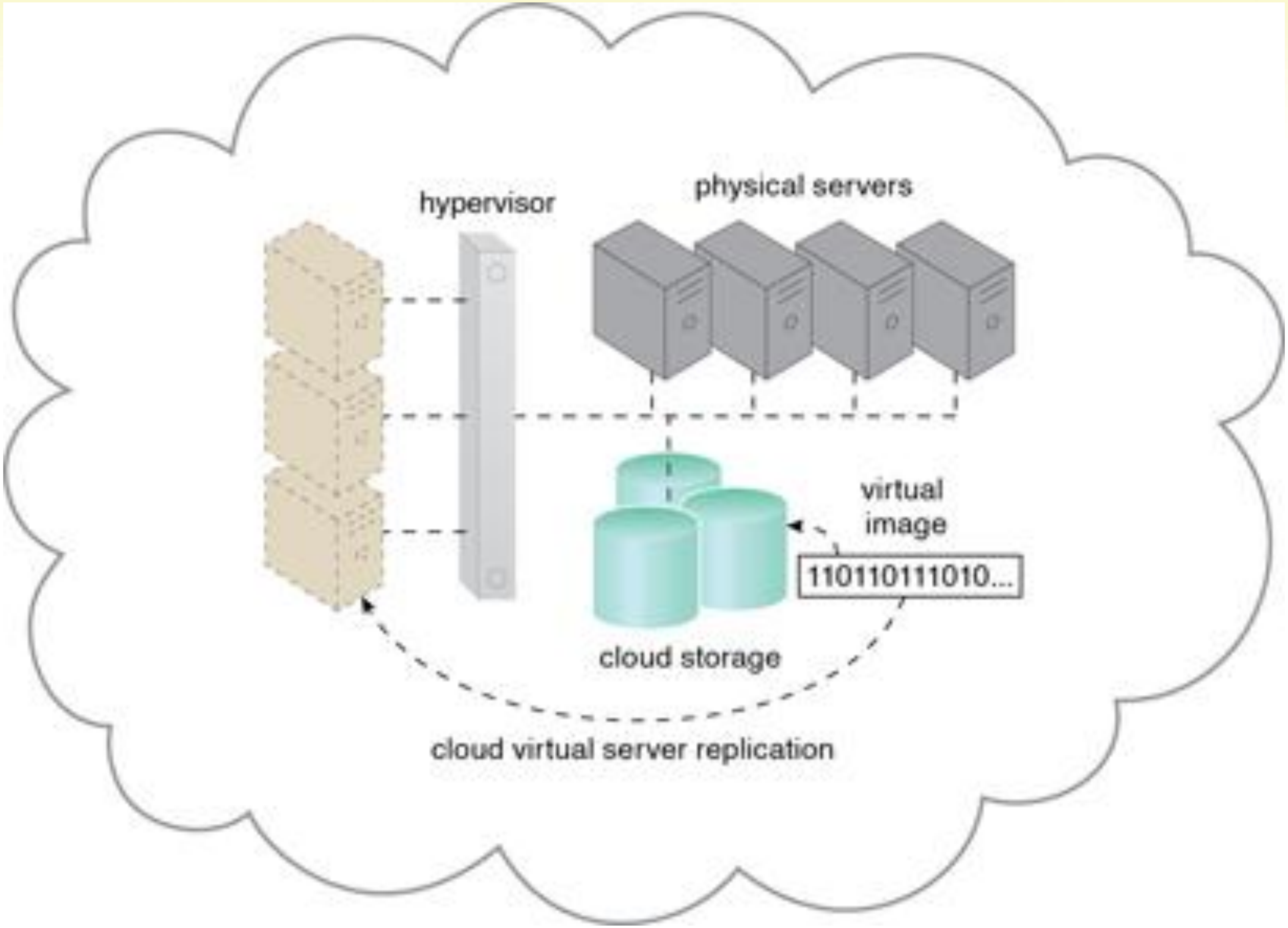
Zero Downtime



Load Balancer



Resource Replication



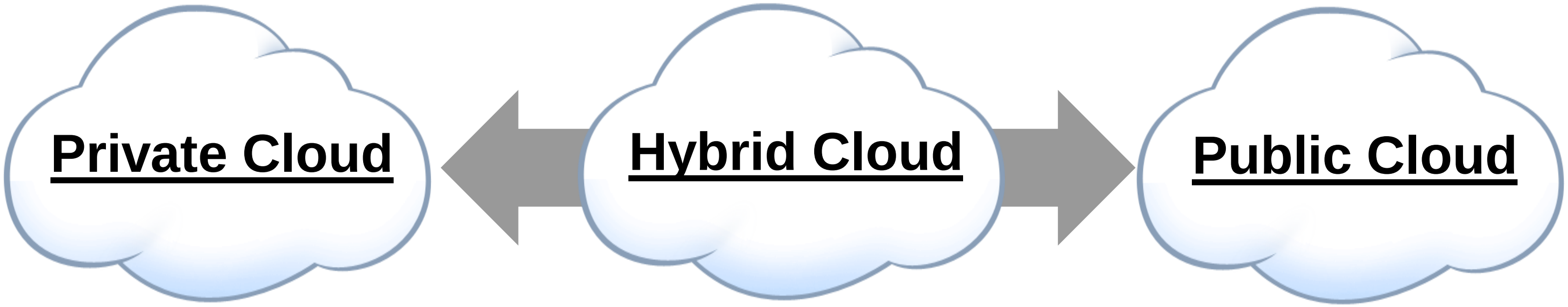
Deployment Models

- 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

Exclusive

Shared by multiple organizations



Private Cloud

Hybrid Cloud

Public Cloud

- Lower total costs
- Control & visibility
- Multiple apps sharing resources

- Cloudbursting – overdraft for peak loads
- Dev/test vs. production
- B2B integration

- Fast & inexpensive to start
- Outsourced services
- Multiple tenants sharing resources

Public Clouds

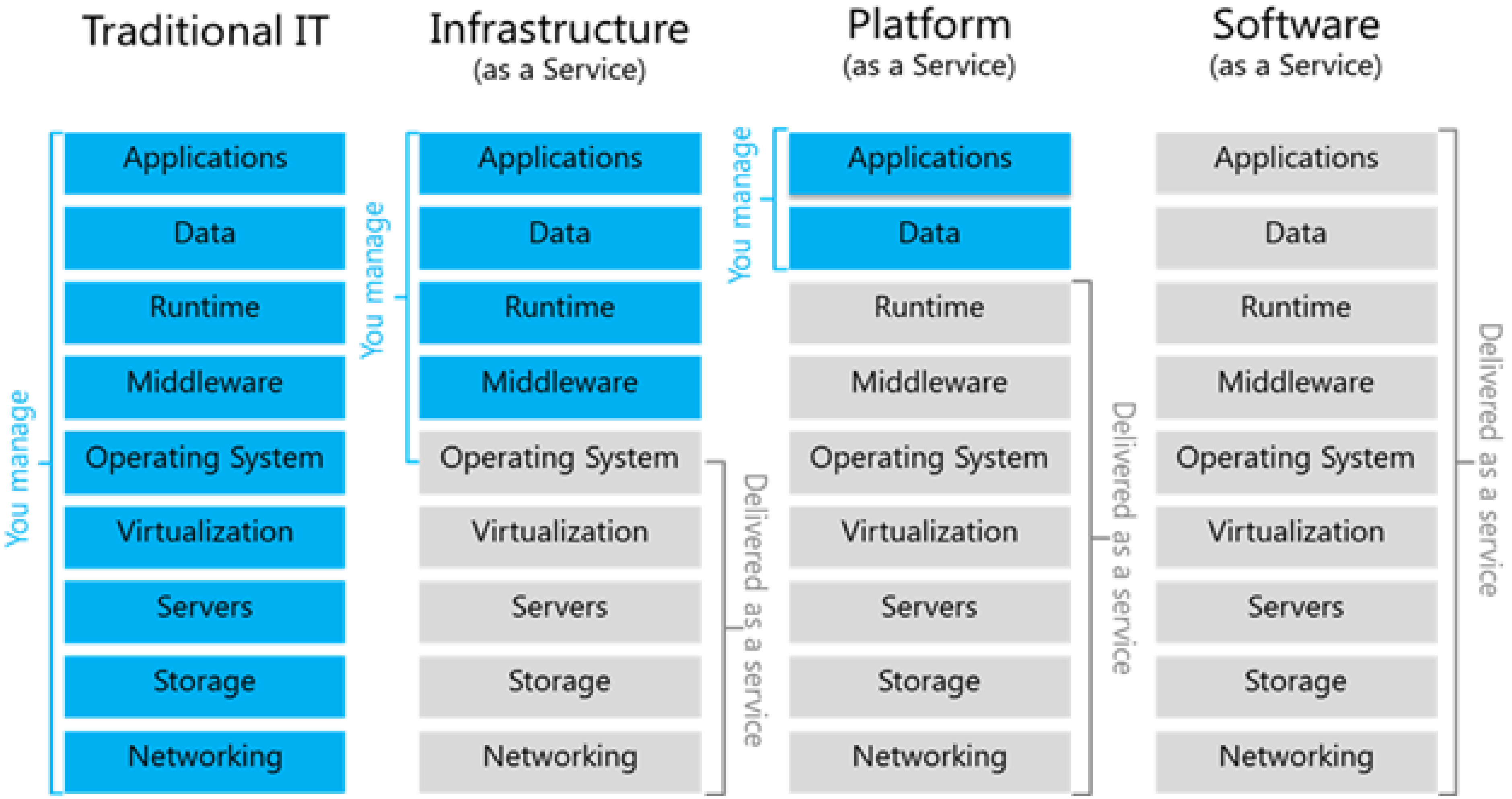
- Provide computing resources on demand
  - Publicly in Internet, for everyone
  - Paid on usage of resources
  - Could be IaaS, PaaS, SaaS or mix of them

- 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)

Levels of Service



Traditional IT

You manage

Applications

Data

Runtime

Middleware

Operating System

Virtualization

Servers

Storage

Networking

Infrastructure (as a Service)

You manage

Applications

Data

Runtime

Middleware

Delivered as a service

Operating System

Virtualization

Servers

Storage

Networking

Platform (as a Service)

You manage

Applications

Data

Delivered as a service

Runtime

Middleware

Operating System

Virtualization

Servers

Storage

Networking

Software (as a Service)

Delivered as a service

Applications

Data

Runtime

Middleware

Operating System

Virtualization

Servers

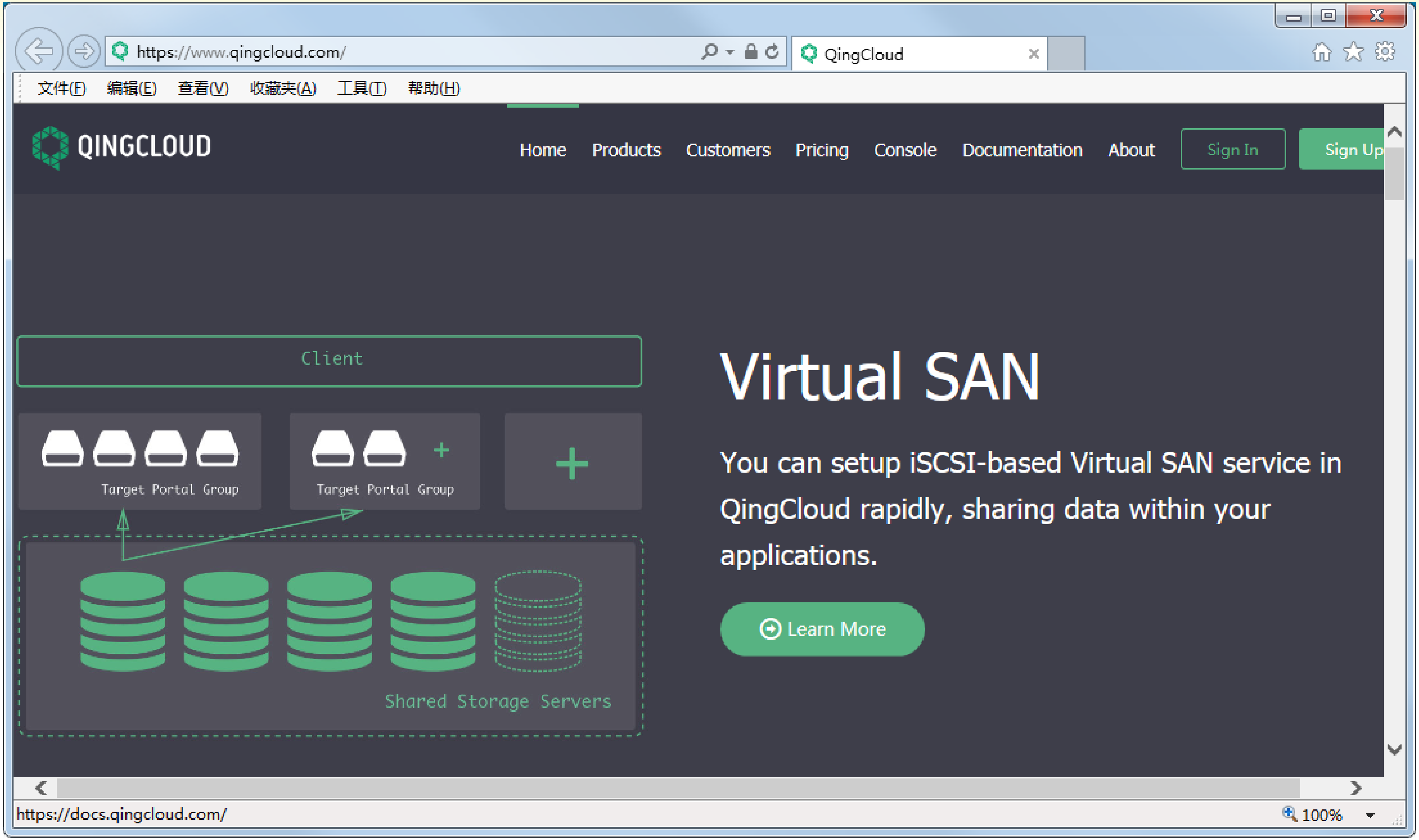
Storage

Networking

IaaS (Infrastructure as a Service)

- PaaS ≈ 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.

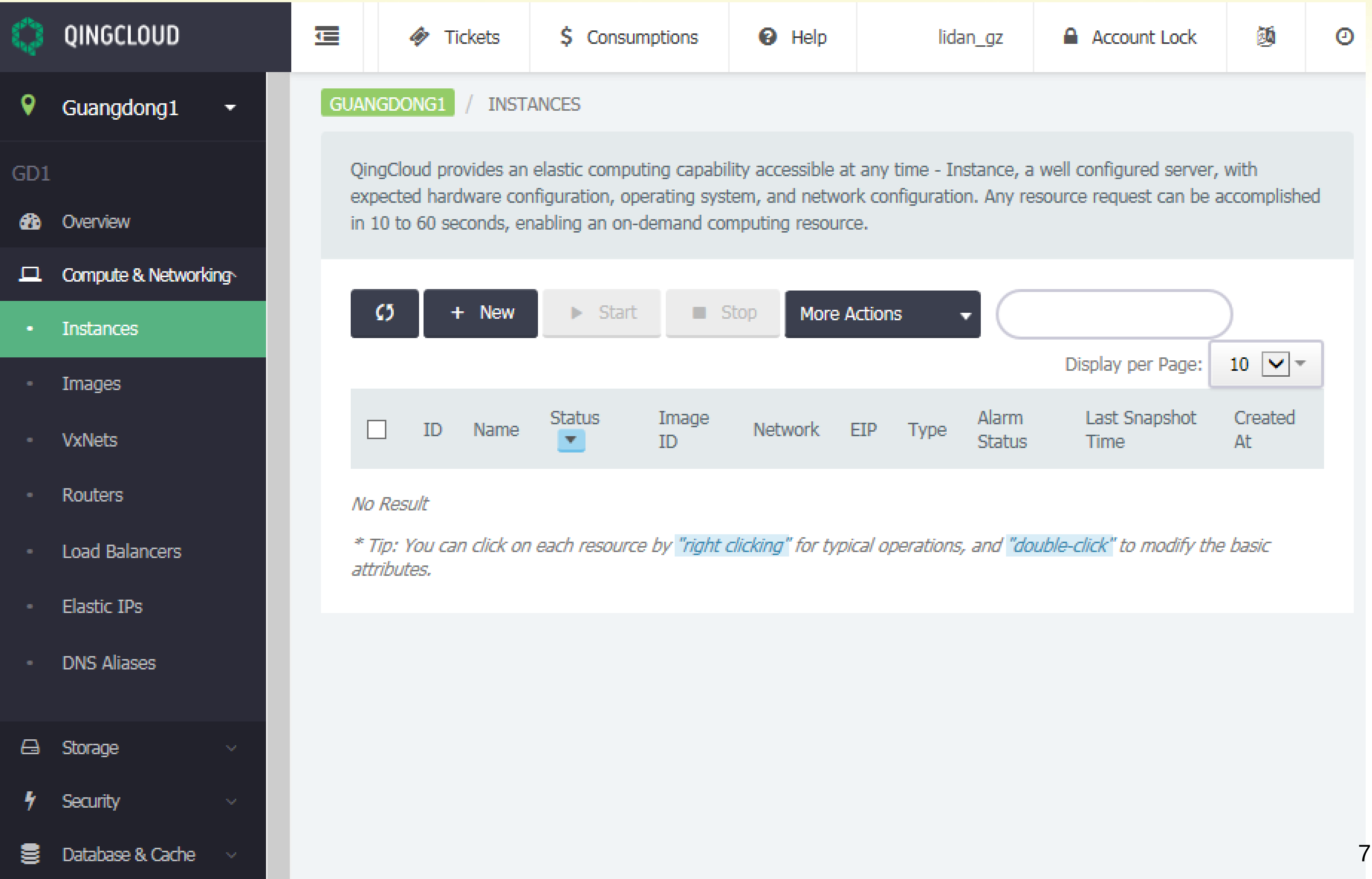
Example of IaaS: QingCloud



Virtual SAN

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

Console of QingCloud



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

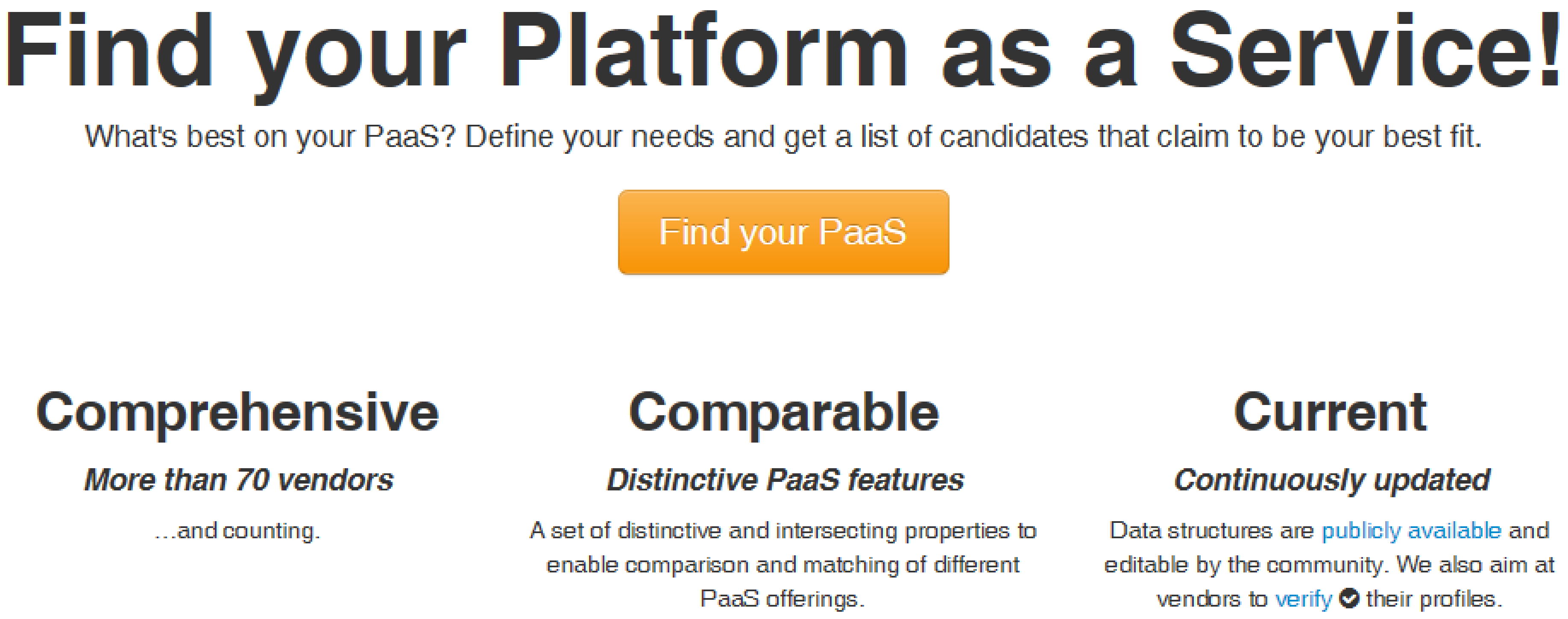
Operating Systems: Linux / Windows / other

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.

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

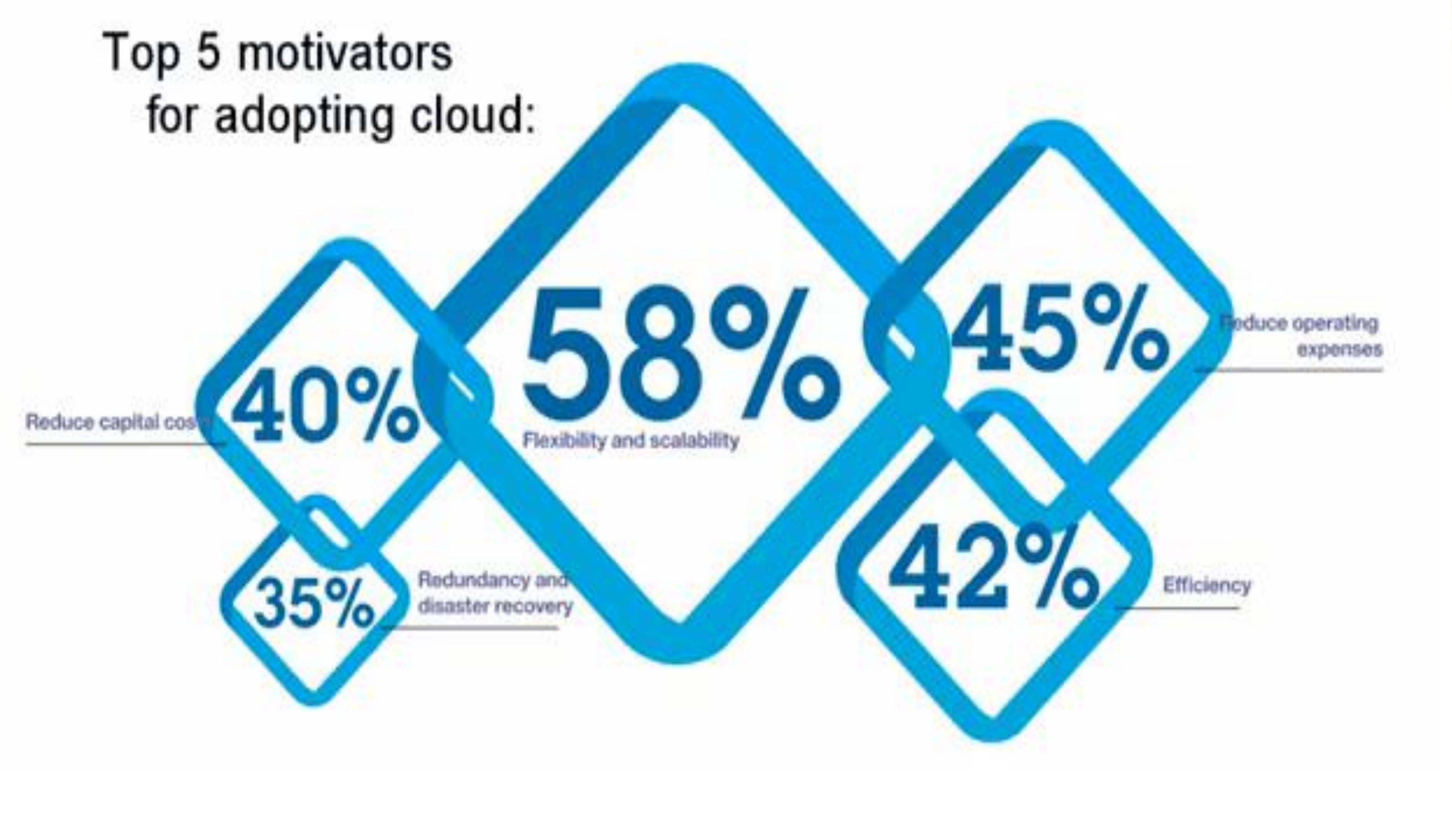
Continuously updated

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

Top Motivators for Adopting Cloud



Source: IBM Tech Trends Report 2011

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

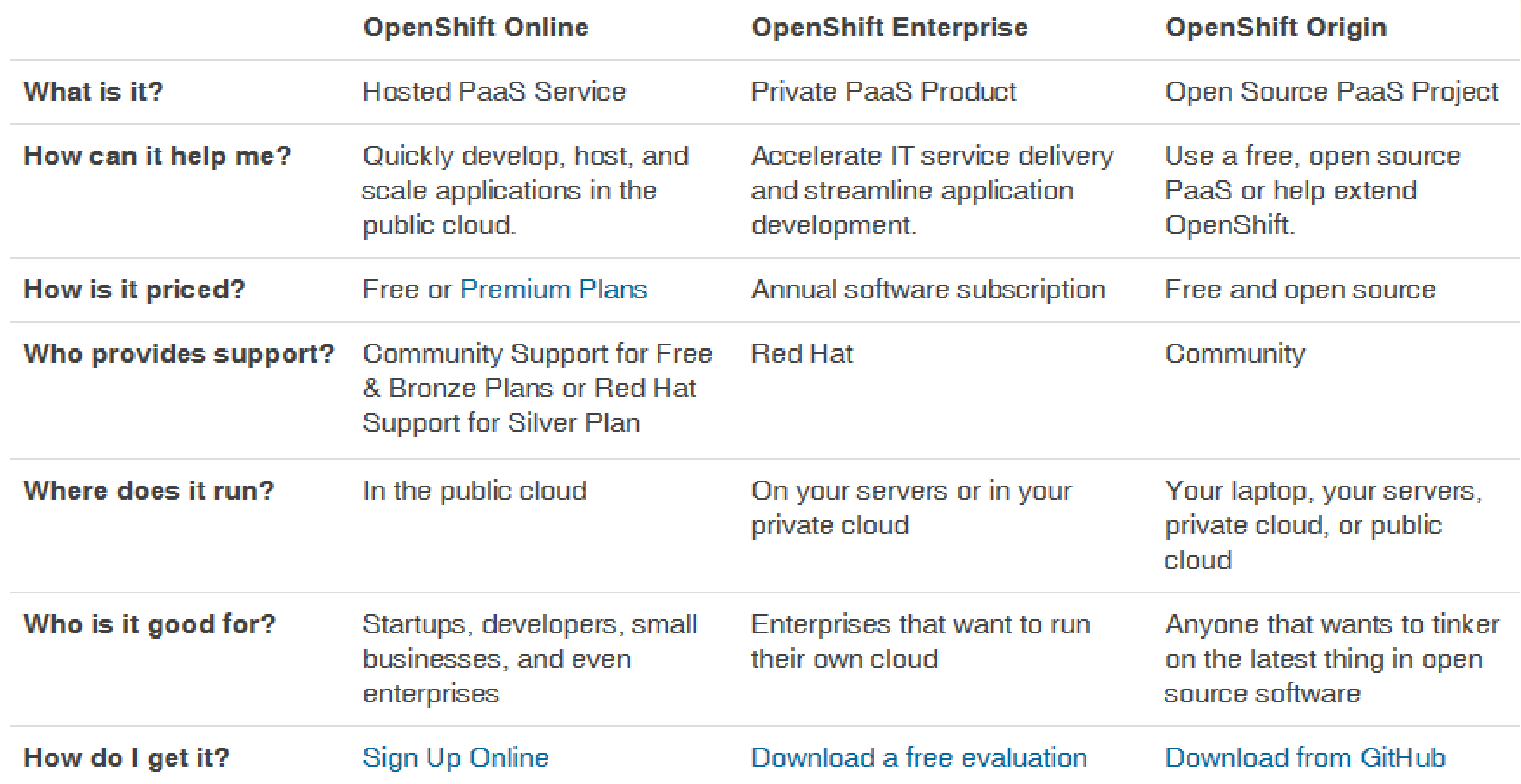
Web Apps
from Browser/Server to Browser/Cloud

OPENSHIFT ONLINE

OPENSHIFT

- 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 **Open**Shift



	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 6 languages: Java, Ruby, PHP, Node.js, Python and Perl;
- 3 Middleware: Jboss, Tomcat, Zend server
- 6 Frameworks: Django ,Drupal ,Flask,Rails ,Switchyard ,Vert.x
- 7 Native Services: Jenkins, Mongodb, Mysql, Openshift Metrics, Pgrouting, Postgis, Postgresql
-

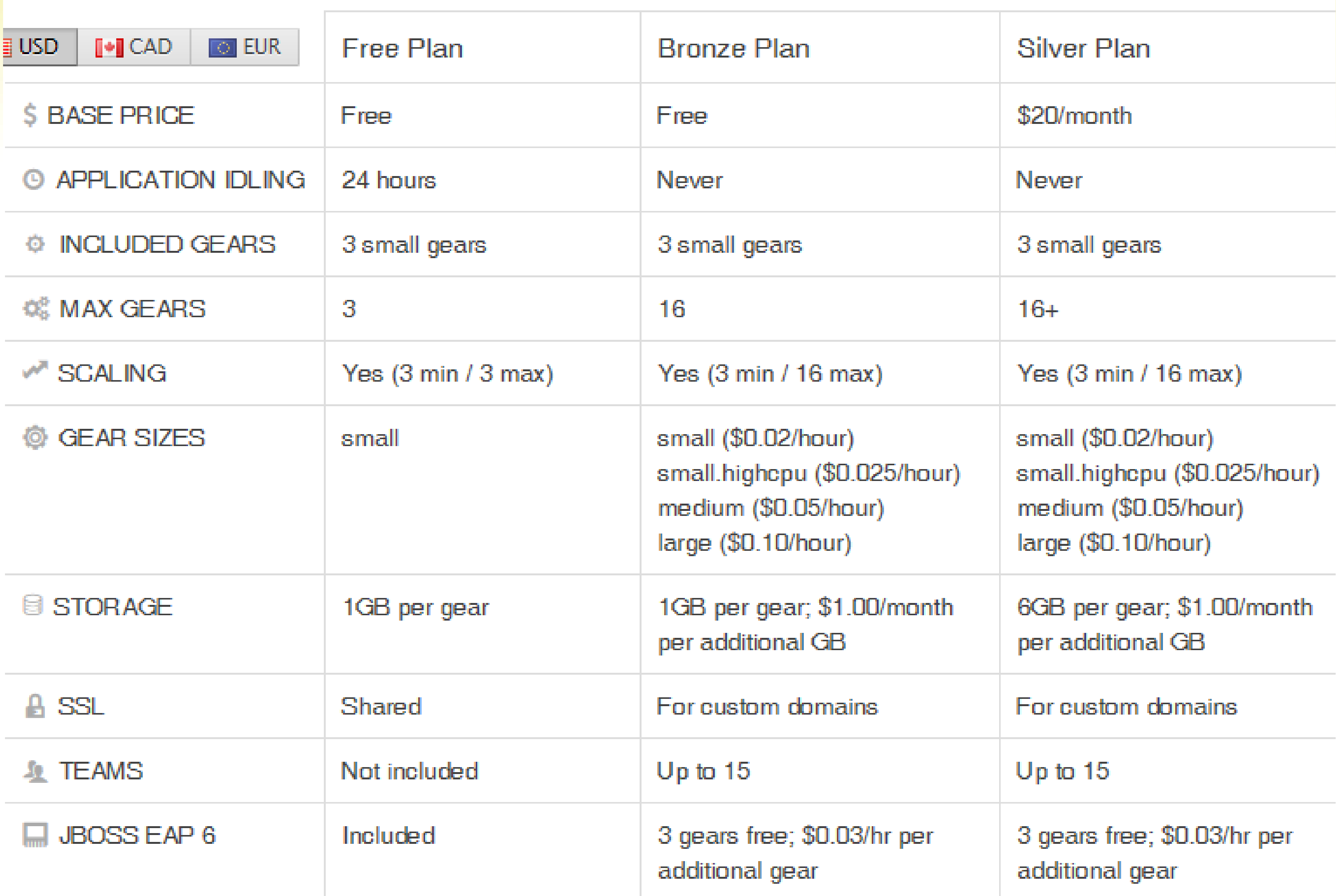
Services and Features

- **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 gears 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 you application needs it.

Monthly Pricing

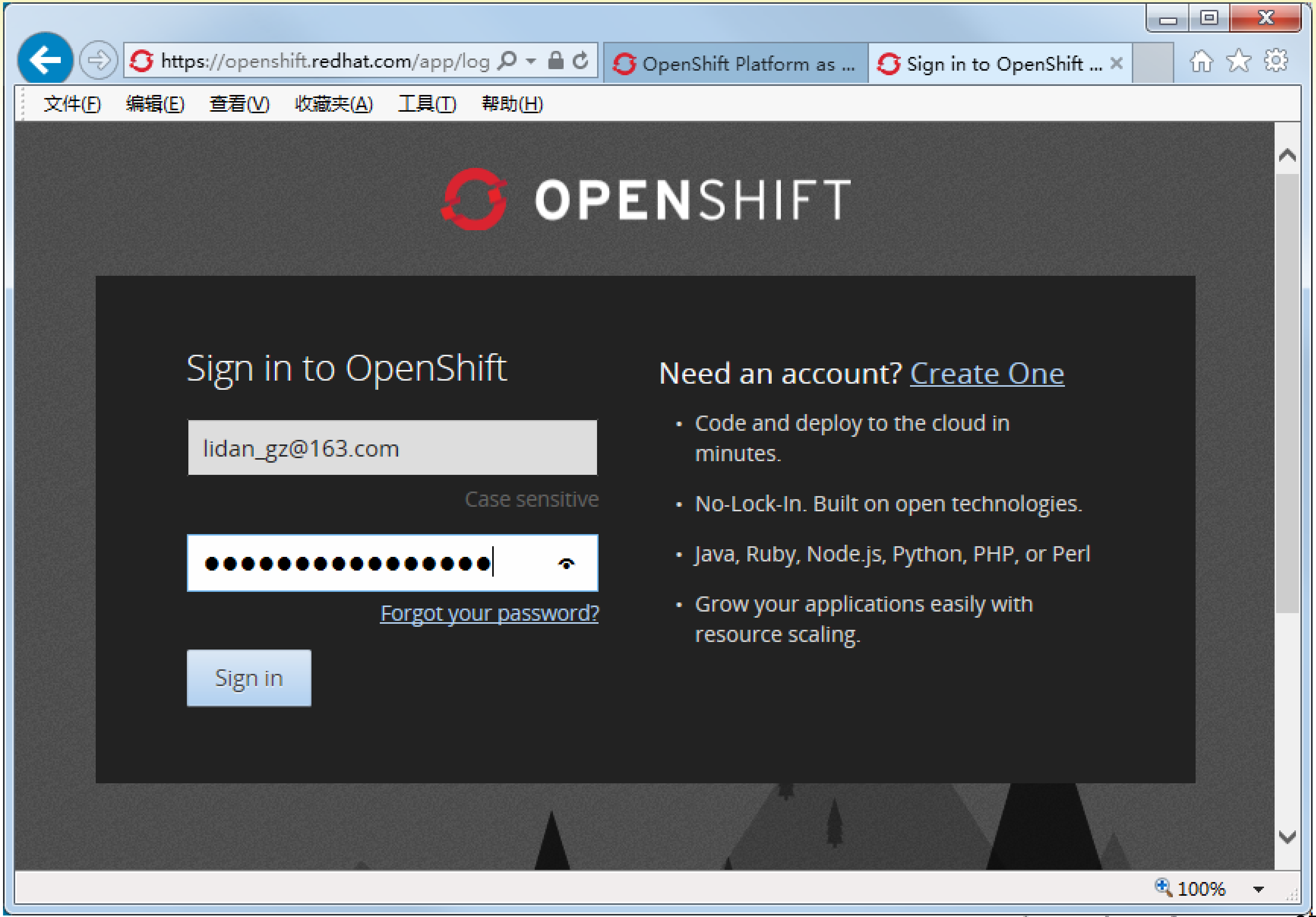


USD CAD EUR	Free Plan	Bronze Plan	Silver Plan
$ BASE PRICE	Free	Free	$20/month
APPLICATION IDLING	24 hours	Never	Never
INCLUDED GEARS	3 small gears	3 small gears	3 small gears
MAX GEARS	3	16	16+
SCALING	Yes (3 min / 3 max)	Yes (3 min / 16 max)	Yes (3 min / 16 max)
GEAR SIZES	small	small ($0.02/hour) 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



https://openshift.redhat.com/app/log
OpenShift Platform as ...
Sign in to OpenShift ...
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
OPENSHIFT
100%

Sign in to OpenShift

lidan_gz@163.com

Case sensitive

●●●●●●●●●●●●●●●●

Forgot your password?

Sign in

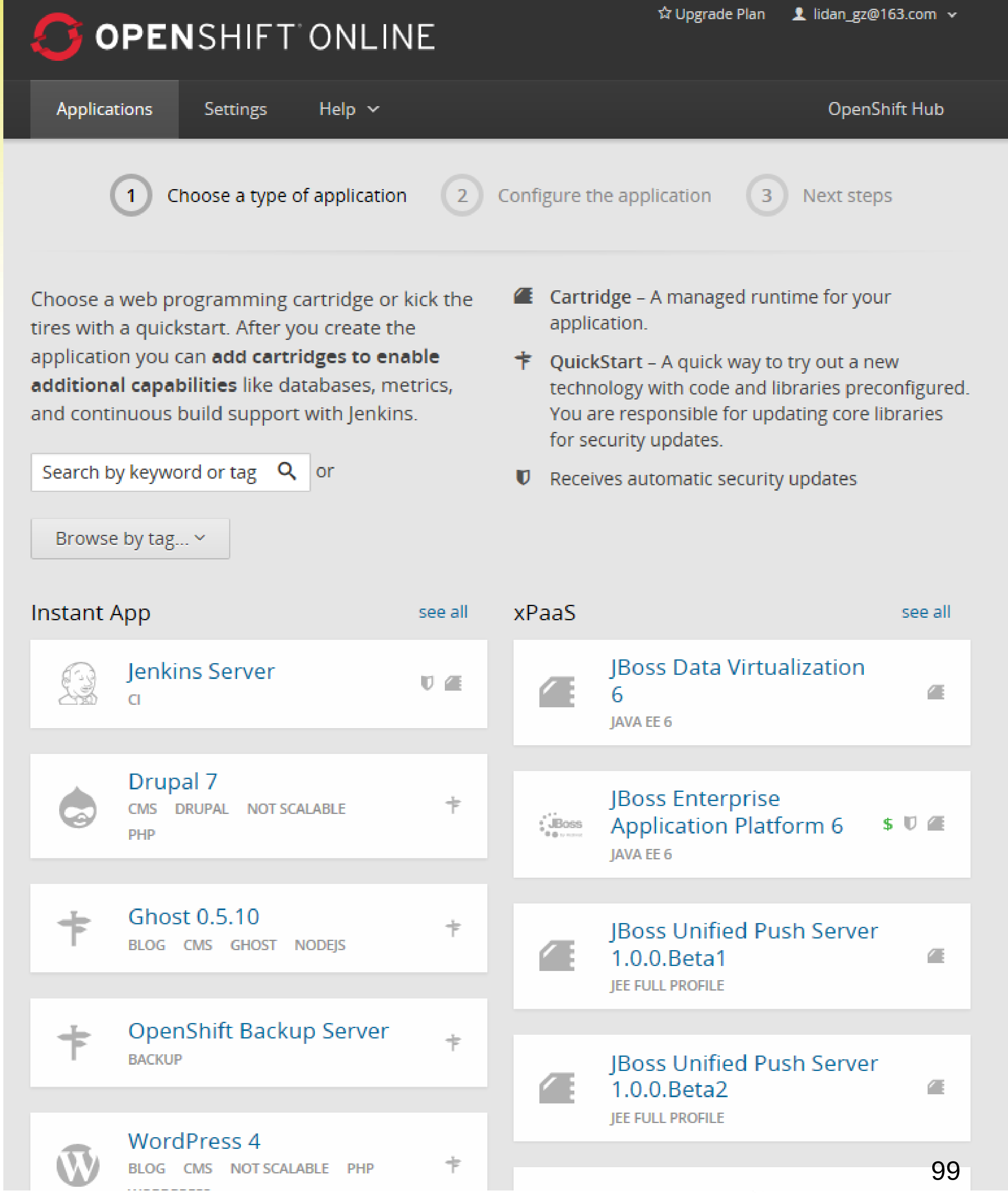
Need an account? Create One

- Code and deploy to the cloud in minutes.
- No-Lock-In. Built on open technologies.
- Java, Ruby, Node.js, Python, PHP, or Perl
- Grow your applications easily with resource scaling.

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 a Type of Application



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

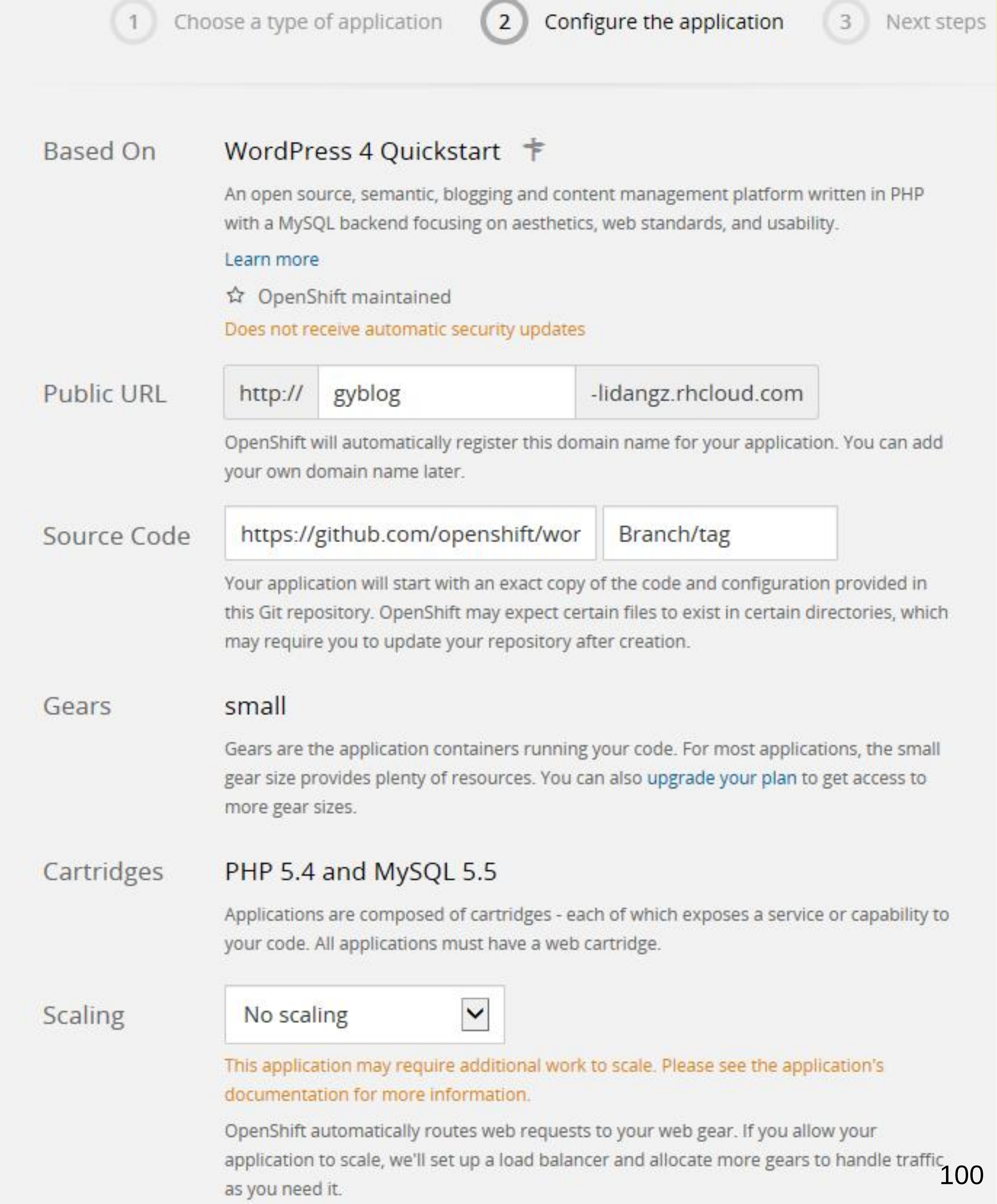
Instant App   see all

- 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

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

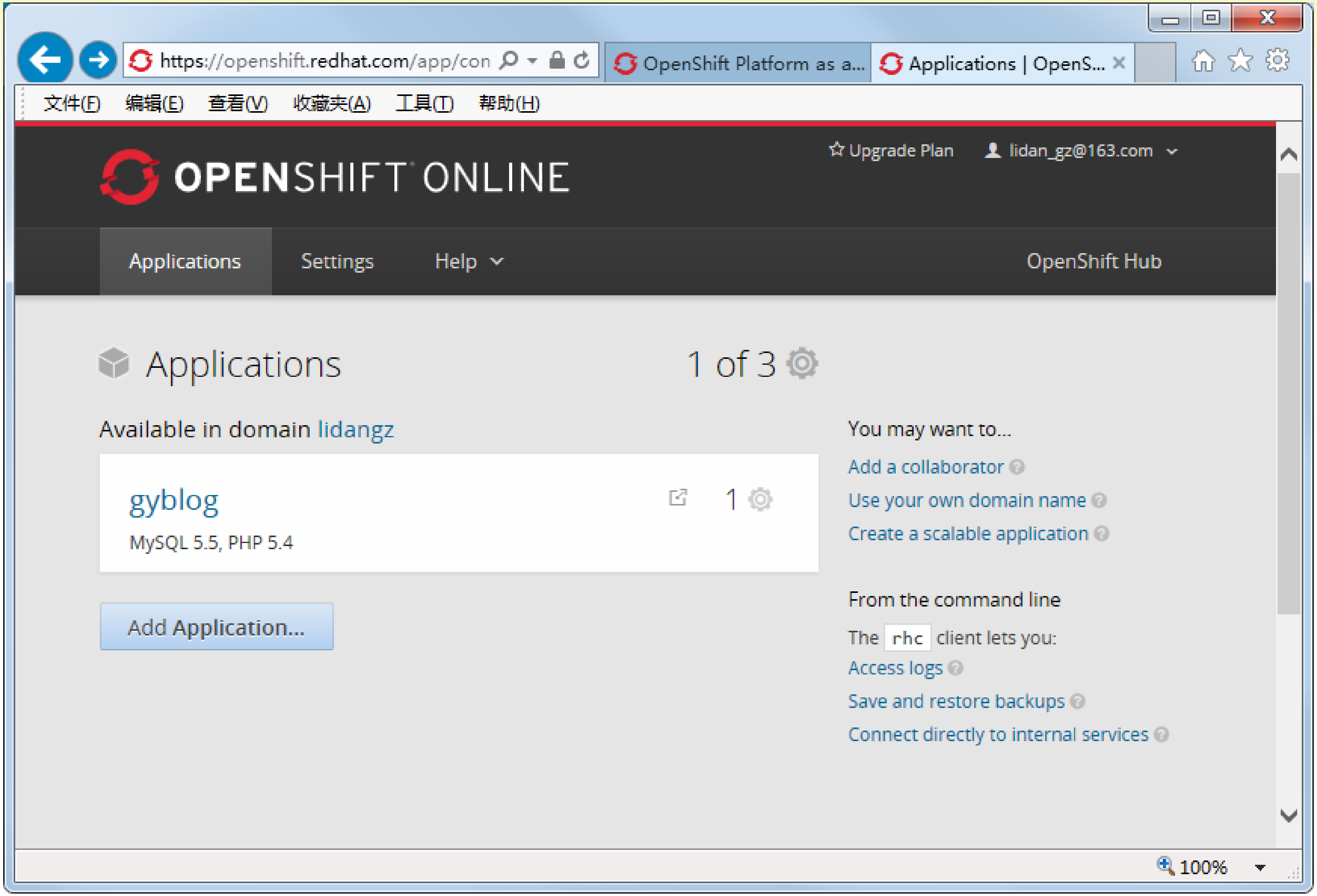
Applications are composed of cartridges - each of which exposes a service or capability to your code. All applications must have a web cartridge.

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



https://openshift.redhat.com/app/con
OpenShift Platform as a...
Applications | OpenS...
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
OPENSHIFT ONLINE
Upgrade Plan
lidan_gz@163.com
Applications
Settings
Help
OpenShift Hub
100%

Applications 1 of 3

Available in domain lidangz

gyblog 1

MySQL 5.5, PHP 5.4

Add Application...

You may want to...

Add a collaborator

Use your own domain name

Create a scalable application

From the command line

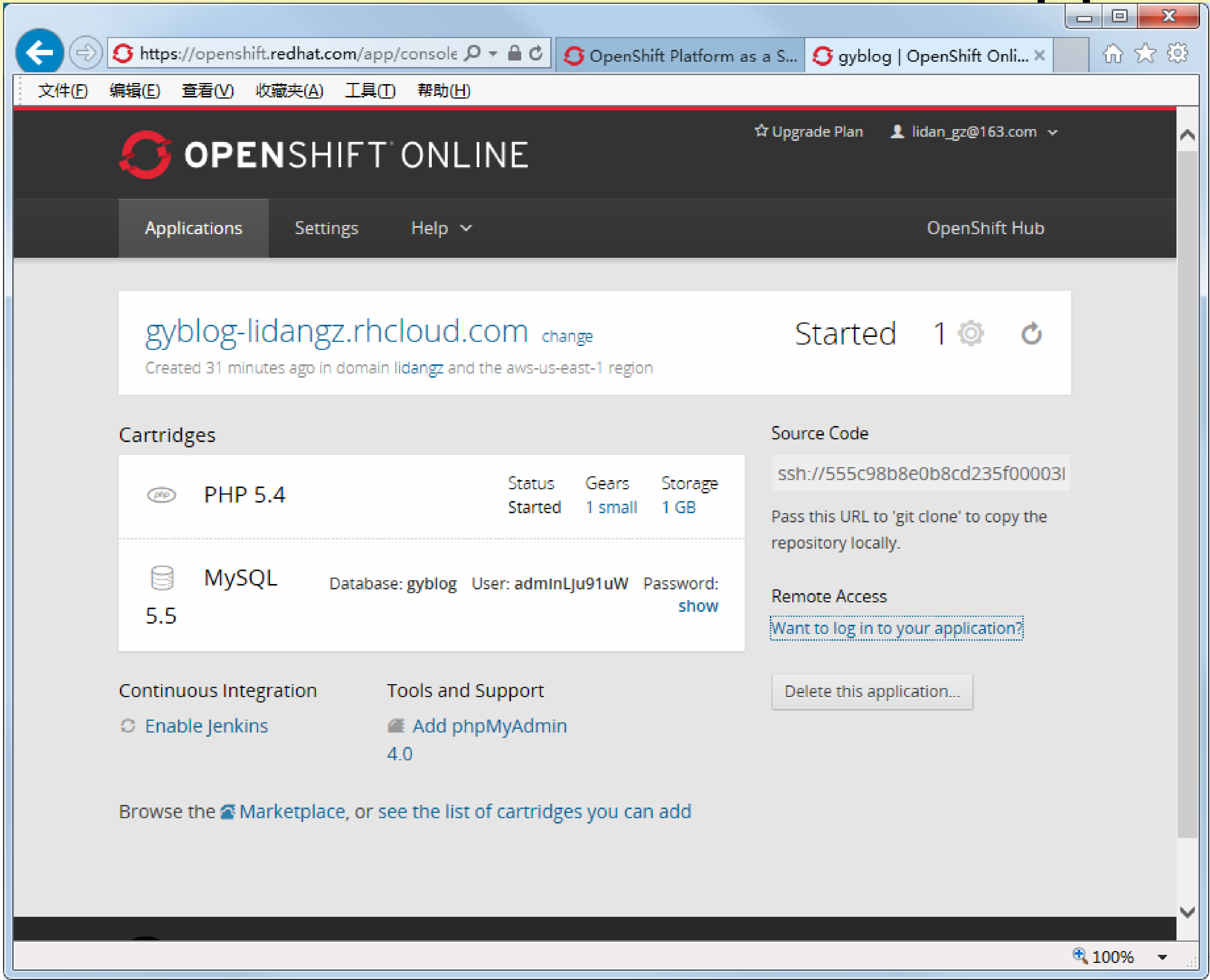
The `rhc` client lets you:

Access logs

Save and restore backups

Connect directly to internal services

Overview of WordPress App


https://openshift.redhat.com/app/console
OpenShift Platform as a S...
gyblog | OpenShift Onli...
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
OPENSHIFT ONLINE
Upgrade Plan
lidan_gz@163.com
Applications
Settings
Help
OpenShift Hub
change
100%

gyblog-lidangz.rhcloud.com

Created 31 minutes ago in domain lidangz and the aws-us-east-1 region

Started 1

Cartridges

PHP 5.4 — Status: Started — Gears: 1 small — Storage: 1 GB

MySQL 5.5 — Database: gyblog — User: adminLJu91uW — Password: show

Source Code

ssh://555c98b8e0b8cd235f00003l

Pass this URL to 'git clone' to copy the repository locally.

Remote Access

Want to log in to your application?

Delete this application...

Continuous Integration

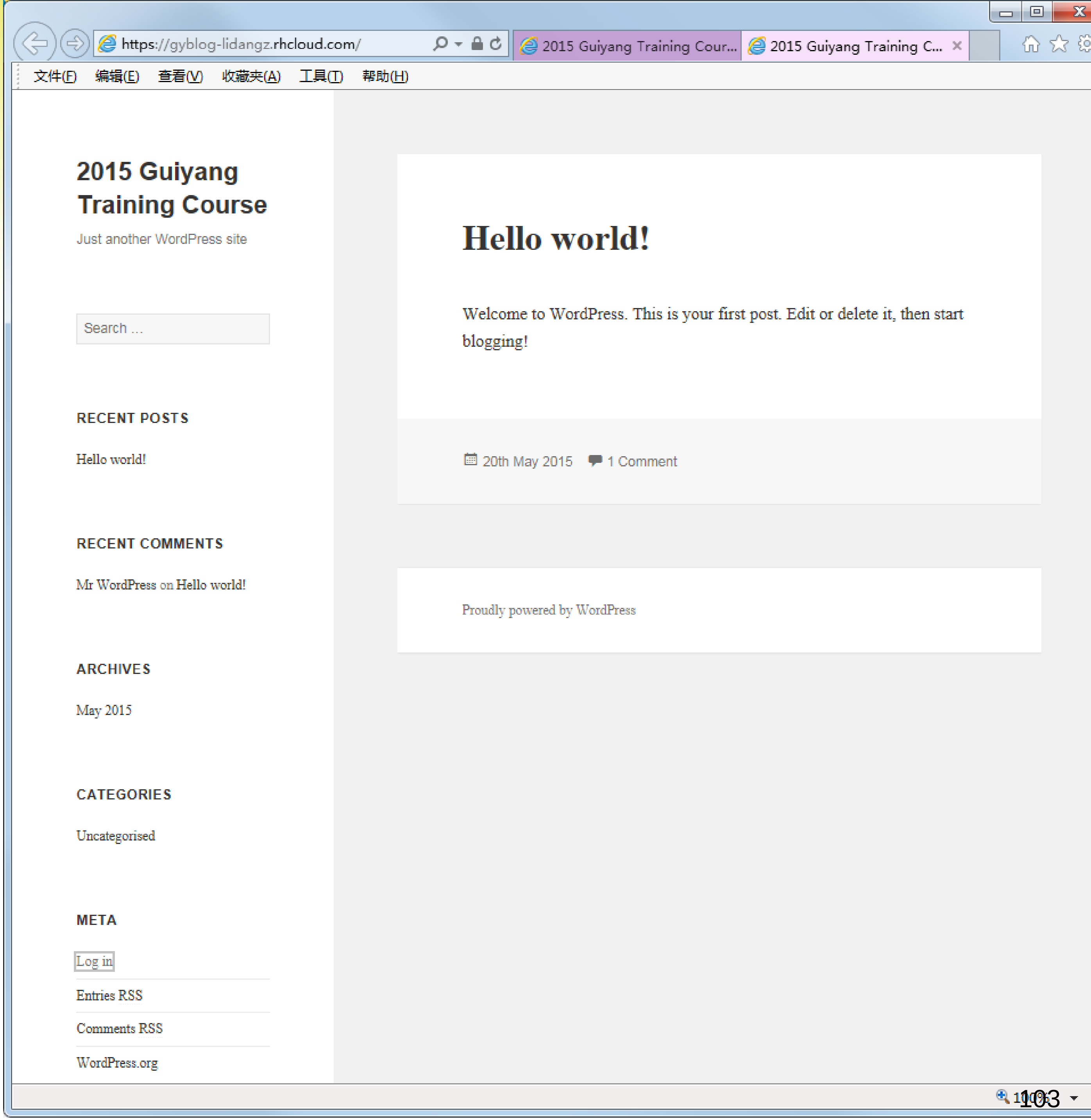
Enable Jenkins

Tools and Support

Add phpMyAdmin 4.0

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

WordPress Blog


https://gyblog-lidangz.rhcloud.com/
2015 Guiyang Training Cour...
2015 Guiyang Training C...
文件(F)
编辑(E)
查看(V)
收藏夹(A)
工具(T)
帮助(H)
100%

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

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!

20th May 2015 1 Comment

Proudly powered by WordPress

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.