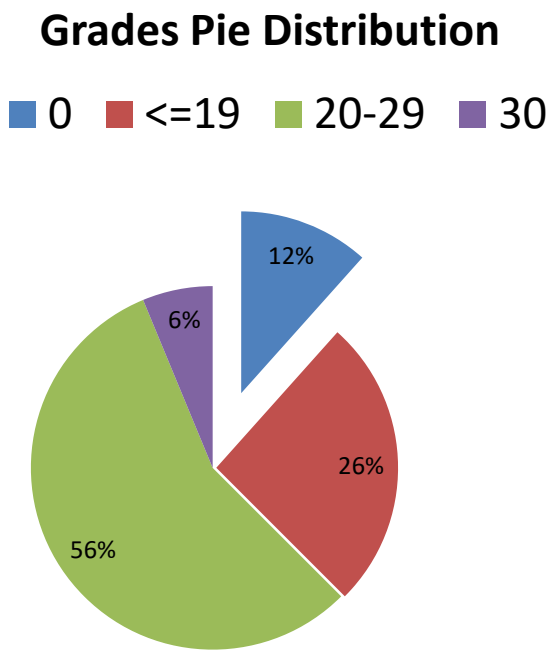
Z6110X0035: Introduction to Cloud Computing - Service

Lecturer: Prof. Zichen Xu

Recap from last Lecture

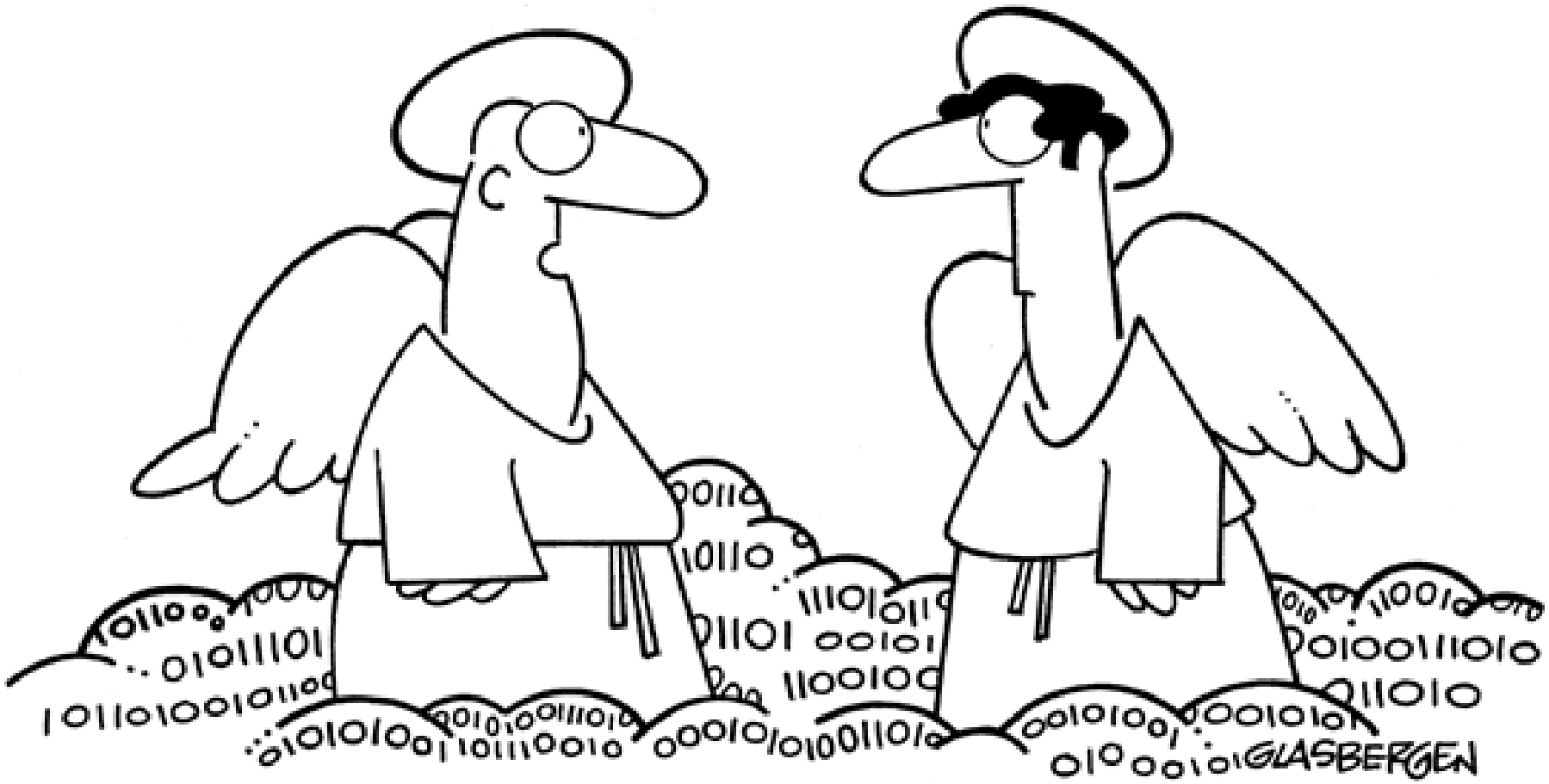
- Bad News
- Good News



Outline

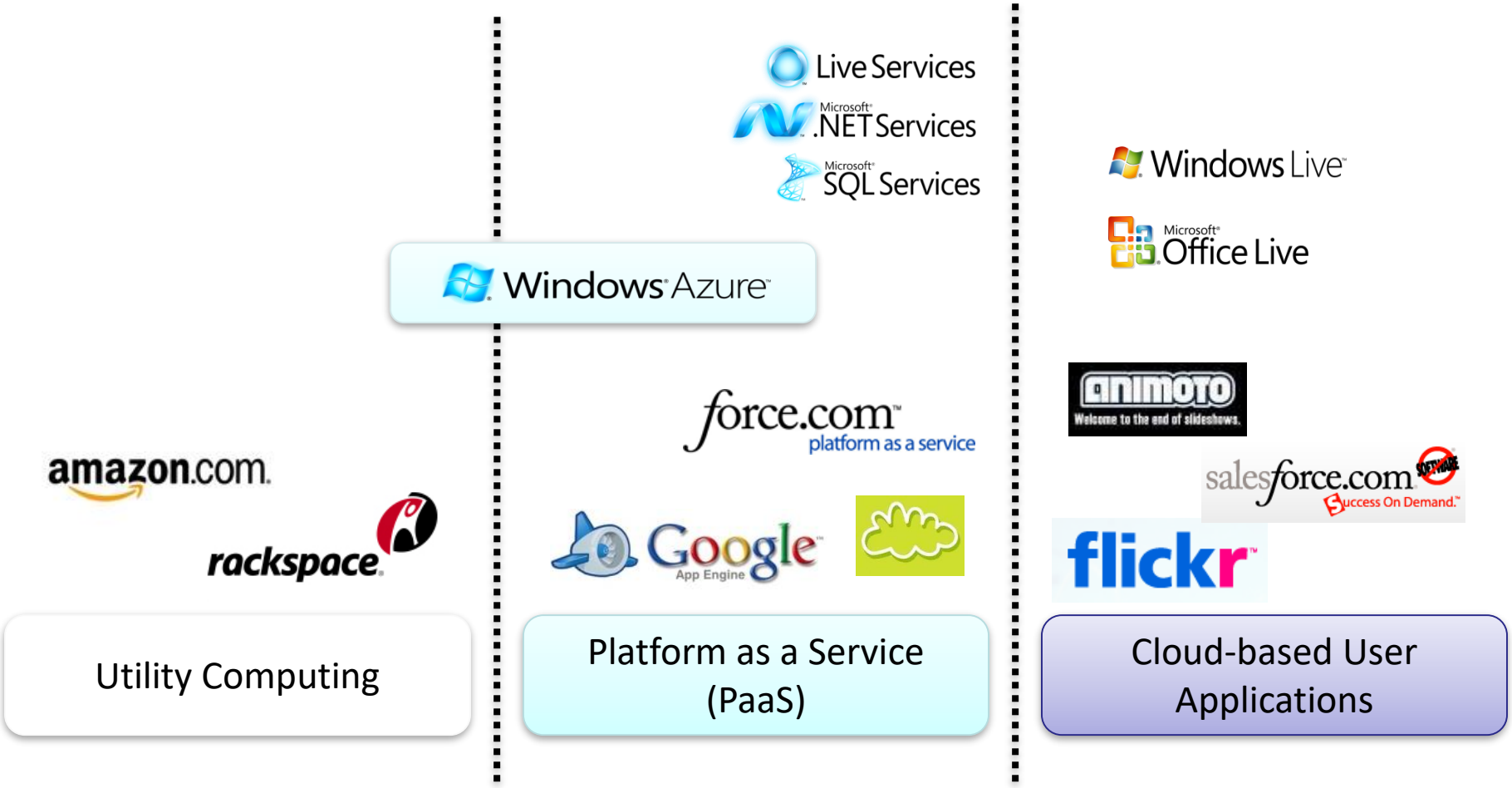
- Service Overview
- Highlights on PaaS
- Services on IaaS
- Pros and Cons on SaaS

Commercial Cloud Formation



"You should have been here back in the old days before cloud computing."

Perspectives

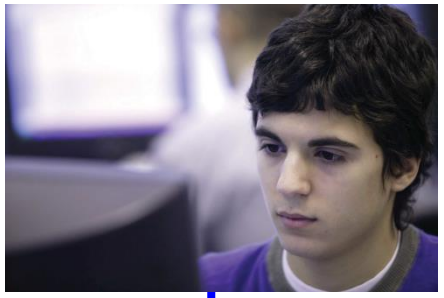


Cloud Computing-What is it?

moving services, computation and to an internal or external, centralized facility or contractor.

Reason

- easy and ubiquitous accessibility
- cost
- collaboration, integration, and analysis on a shared common platform.



SLAs

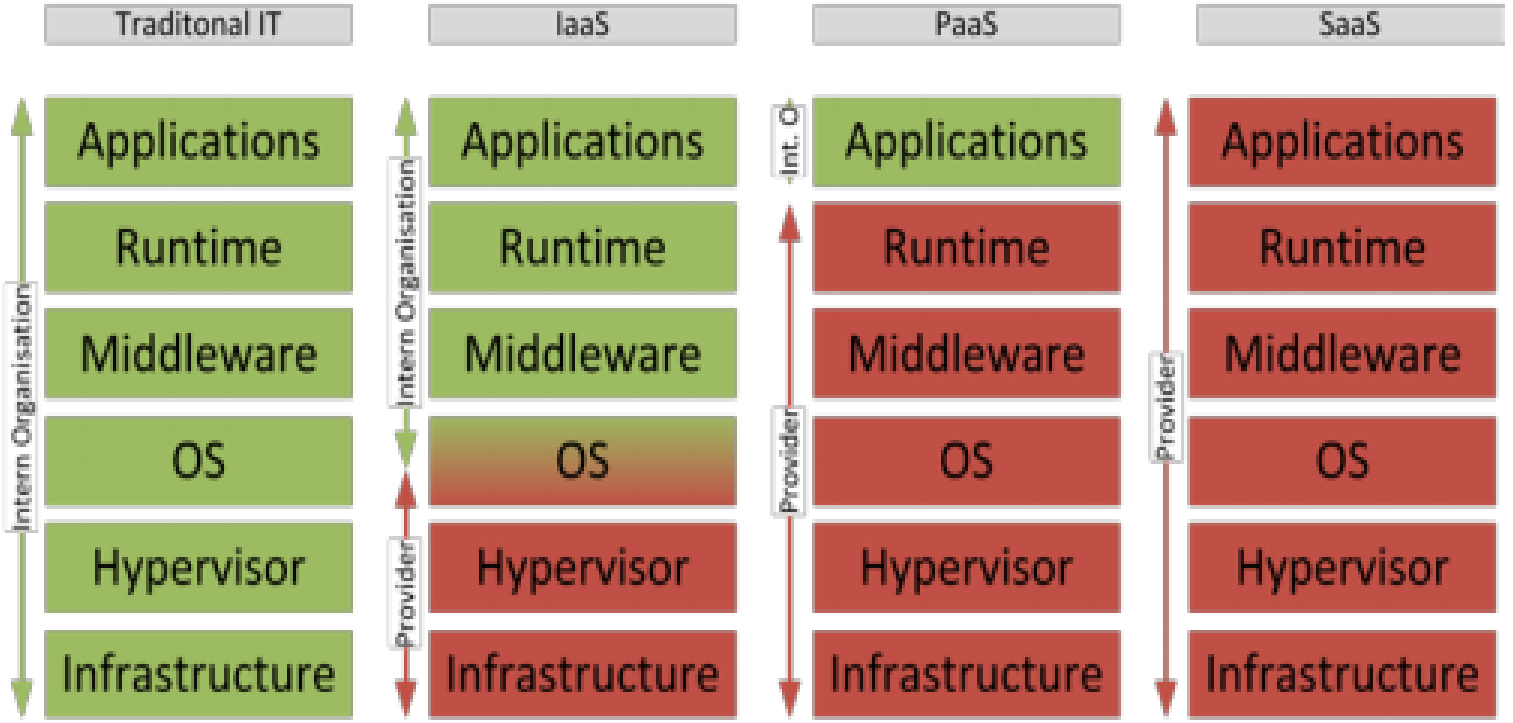
Web Services

Virtualization

What is it?

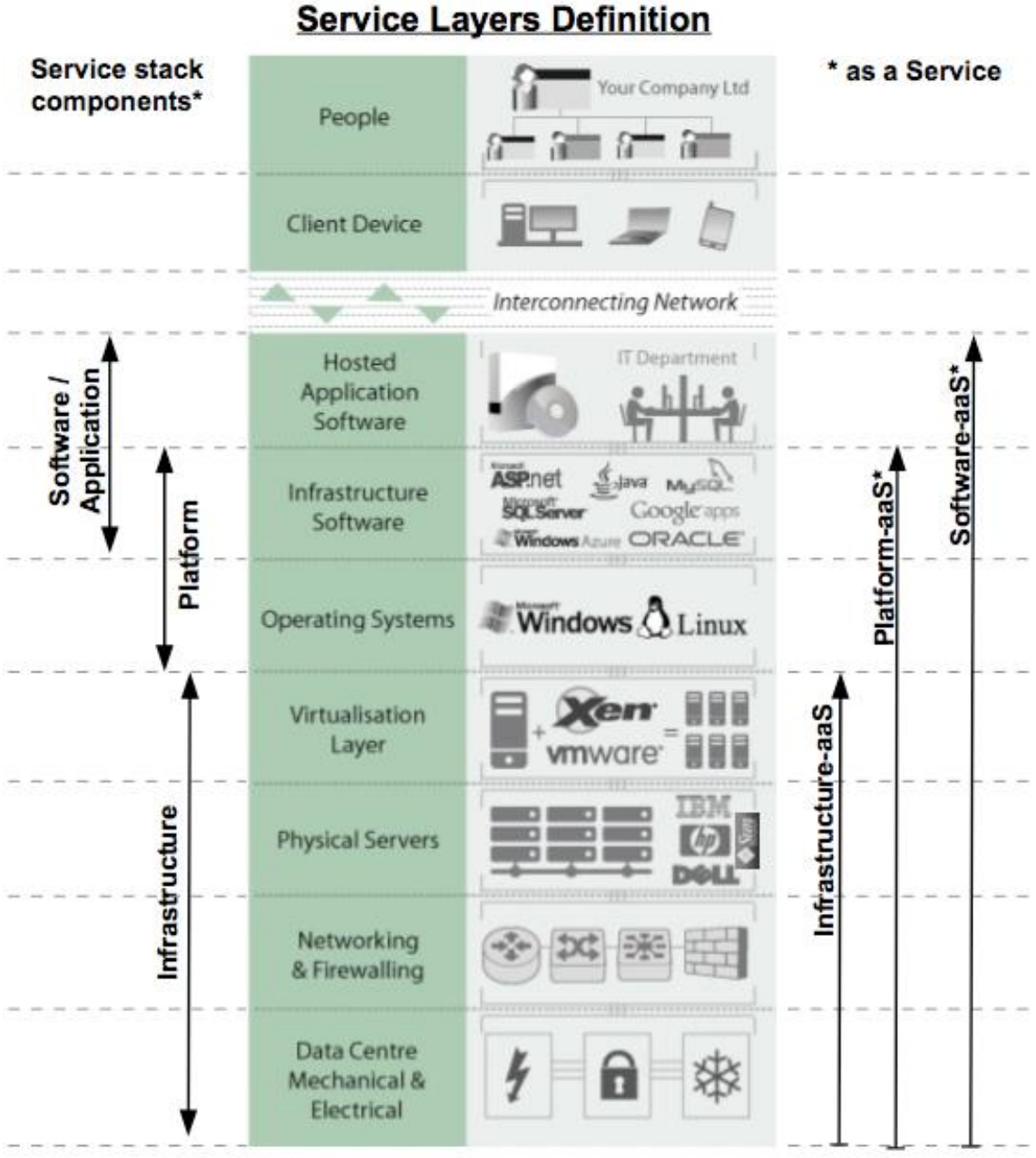
- SaaS (software-as-a-service). WAN-enabled application services (e.g., Google Apps, Salesforce.com, WebEx)
- PaaS (platform-as-a-service). Foundational elements to develop new applications (e.g., Coghead, Google Application Engine)
- IaaS (infrastructure-as-a-service). Providing computational and storage infrastructure in a centralized, location-transparent service (e.g., Amazon)

What is it?



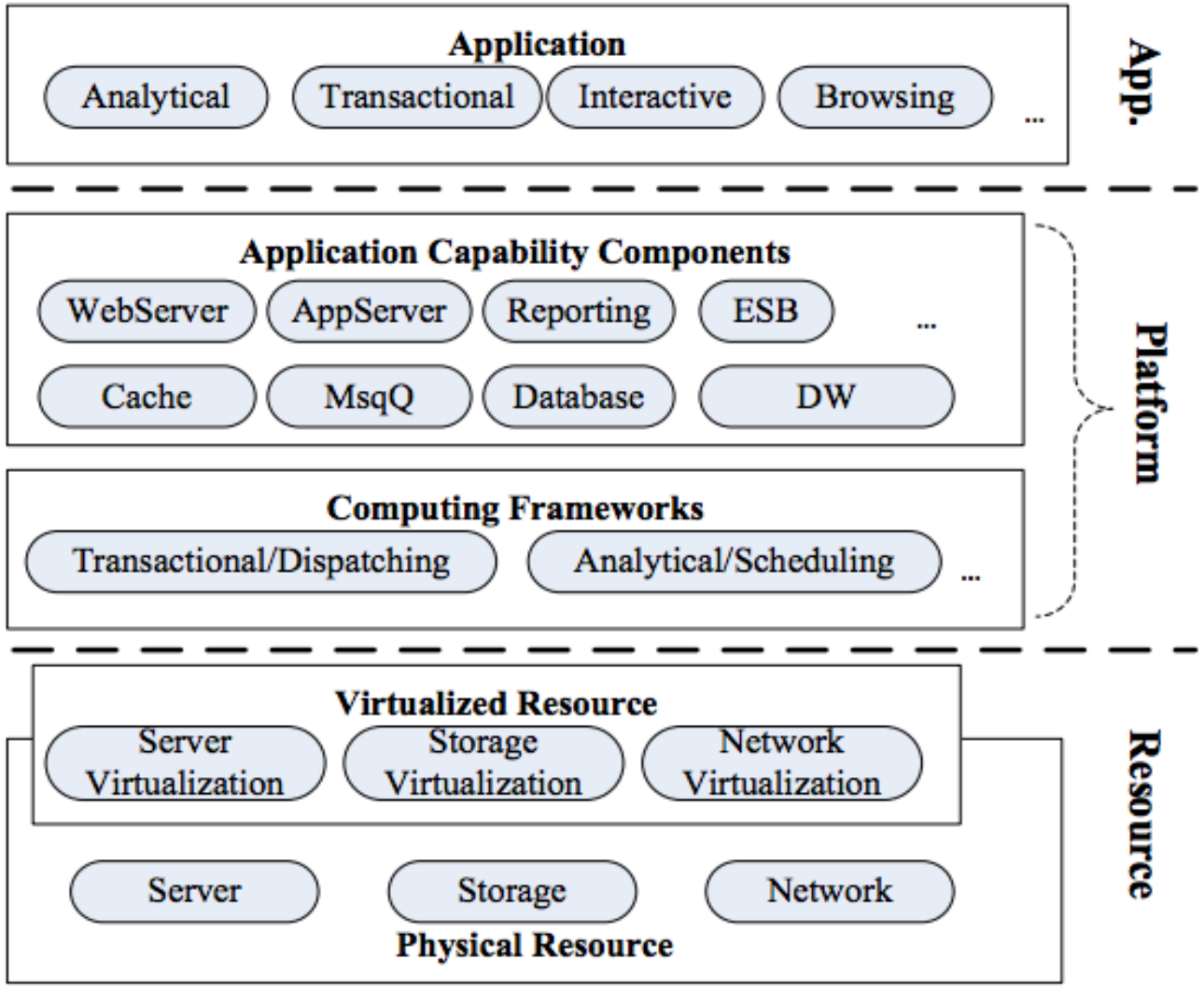
Source: Based on the model developed by NIST (2011)

- Green indicates the levels owned and operated by the organization
- Red levels are run and operated by the service provider



Source:
http://www.katescomment.com/iaas-paas-saas-definition/

Core Stacks



Source: Goel, Pragati, and Mayank Kumar. "An overview of Cloud Computing."

Resource Layer

infrastructure layer which is composed of physical and virtualized computing, storage and networking resources.

Platform Layer

computing framework manages the transaction dispatching and task scheduling.
storage sub-layer provides storage and caching capability

Application Layer

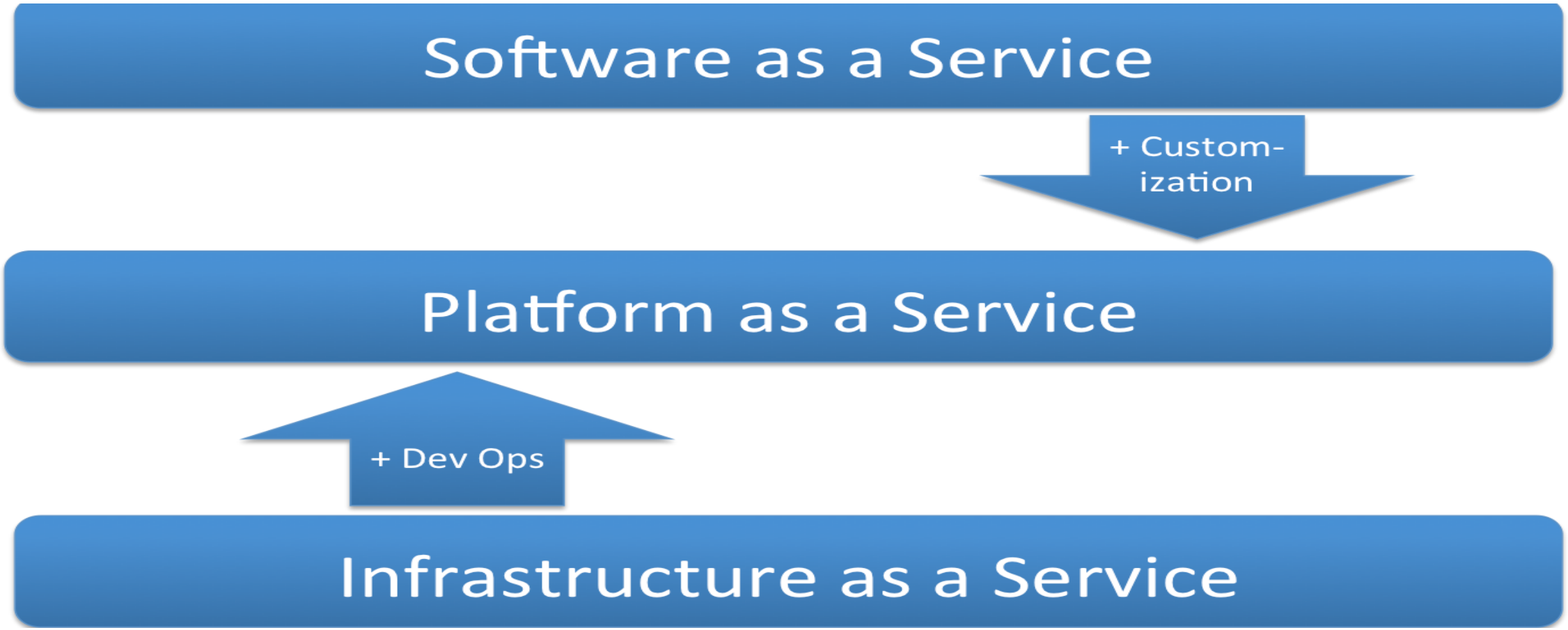
general application logic
either on-demand capability or flexible management.
no components will be the bottle neck of the whole system.
large and distributed transactions and management of huge volume of data.
All the layers provide external service through web service or other open interfaces.

Introduction to PaaS

What is a Platform?

- A platform is anything you can leverage to accomplish something in a simpler, faster, or otherwise better way than you could without.
- As a programmer, you leverage pre-existing code rather than starting from scratch and writing everything.
- The most well-known software platforms for desktop software are Windows and Mac OS

Platform as a Service (PaaS)



Web Platforms

- The infrastructure or hosting layer is analogous to desktop computer hardware and the platform layer is analogous to a desktop operating system.
- Additional features such as email distribution lists, contact form handlers, e-commerce options and other tools that make it easier to build and run a website are part of almost every hosting service

Goals of PaaS

- The ultimate goal of a PaaS is to make it easier for you to run your website or web application no matter how much traffic it gets.
- You just deploy your application and the service figures out what to do with it.
- A platform as a service should handle scaling seamlessly for you so you can just focus on your website and the code running it.

Why PaaS?

- There is always a conflict between the developers and the System Engineers
- Developers are keen on getting their environments up without waiting.
- System Engineers care about performance and stability
  Creates a peaceful environment for both parties

OpenShift

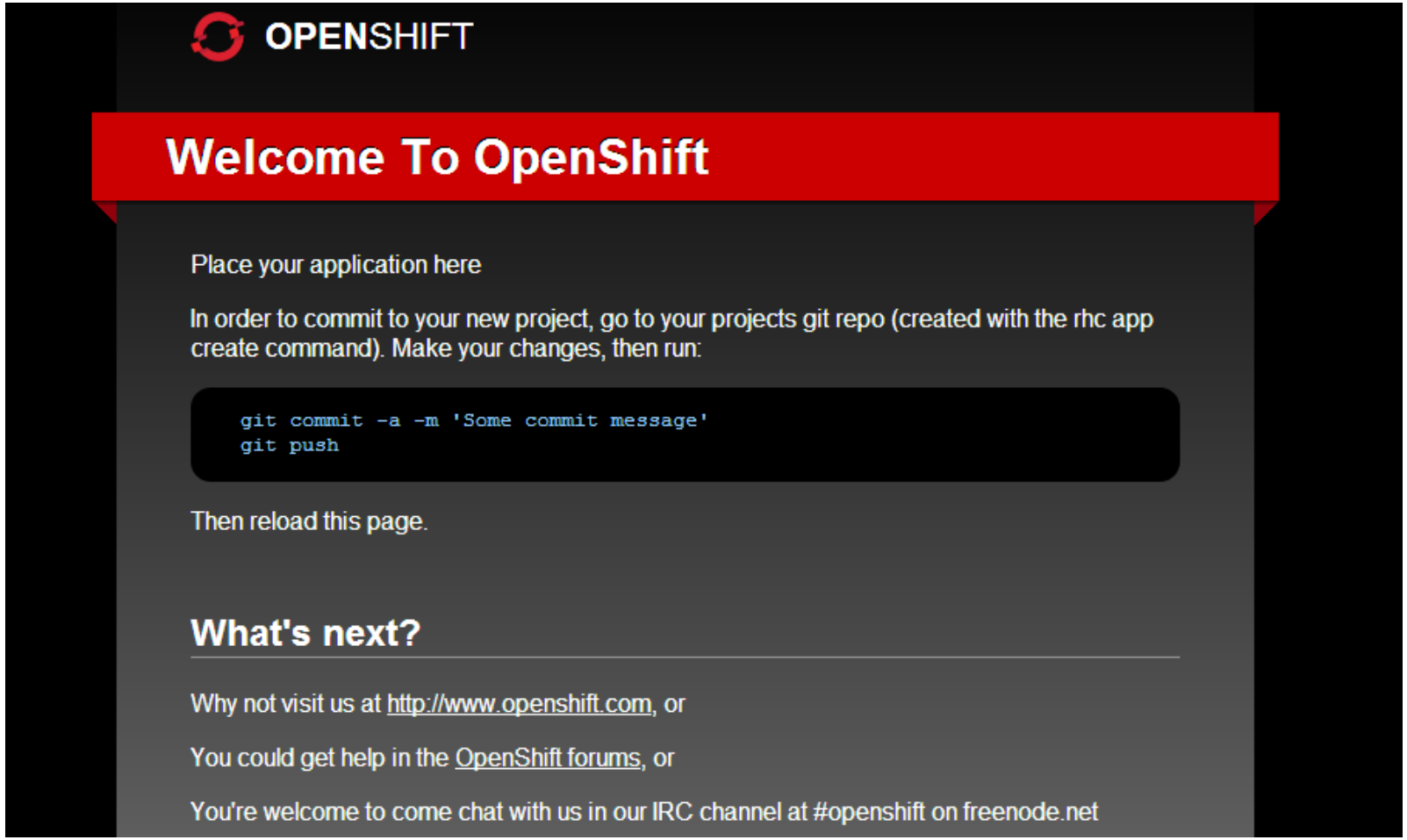
- The open source PaaS from RedHat
- OpenShift runs on top of OpenStack Companies can deploy OpenShift on top of their infrastructure
- OpenStack is the infrastructure and OpenShift is the platform that run on top of it
- This analogous to Apache and MySQL that run on top of a Linux machine

OpenShift Example

What is a cartridge?

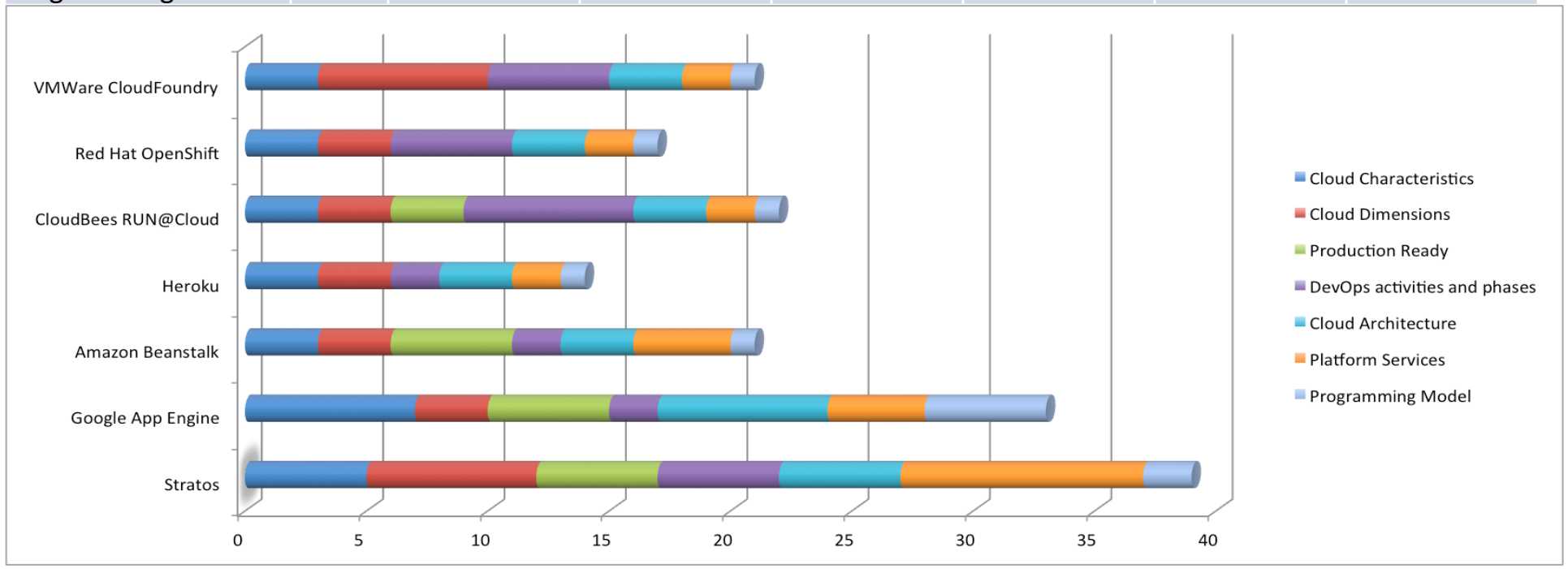
- OpenShift offers cartridges to deploy on to Red Hat's infrastructure
- Sign up for OpenShift
- Create your own namespace
- Deploy Apps
- Sample cartridges – Java, PHP, Ruby, Python
- Framework support- CakePHP, CodeIgniter

OpenShift PHP Cartridge



Platform Comparison

	Stratos	Google App Engine	Amazon Beanstalk	Heroku	CloudBees RUN@Cloud	Red Hat OpenShift	VMWare CloudFoundry
Cloud Characteristics	5	7	3	3	3	3	3
Cloud Dimensions	7	3	3	3	3	3	7
Production Ready	5	5	5	0	3	0	0
DevOps activities and phases	5	2	2	2	7	5	5
Cloud Architecture	5	7	3	3	3	3	3
Platform Services	10	4	4	2	2	2	2
Programming Model	2	5	1	1	1	1	1



Overview for IaaS

What is the problems in conventional case ?

- Companies IT investment for peak capacity
- Lack of agility for IT infrastructure
- IT maintain cost for every company
- Usually suffered from hardware failure risk
- …etc

These IT complexities force company back !!

Overview for IaaS

How to solve these problem ?

Let's consider some kind of out-sourcing solution

- Somebody will handle on demand capacity for me
- Somebody will handle high available resource for me
- Somebody will handle hardware management for me
- Somebody will handle system performance for me
- Somebody will …

Frankly, that would be a great solution IF there were "somebody".
But who can be this "somebody", and provide all these services ?

Overview for IaaS

Infrastructure as a Service will be the salvation.

IaaS cloud provider takes care of all the IT infrastructure complexities.
IaaS cloud provider provides all the infrastructure functionalities.
IaaS cloud provider guarantees qualified infrastructure services.
IaaS cloud provider charges clients according to their resource usage.

But, what make all of these happen so magically ?

Virtualization

Assume that you are going to be an IaaS cloud provider.

Then, what are the problems you are facing ?

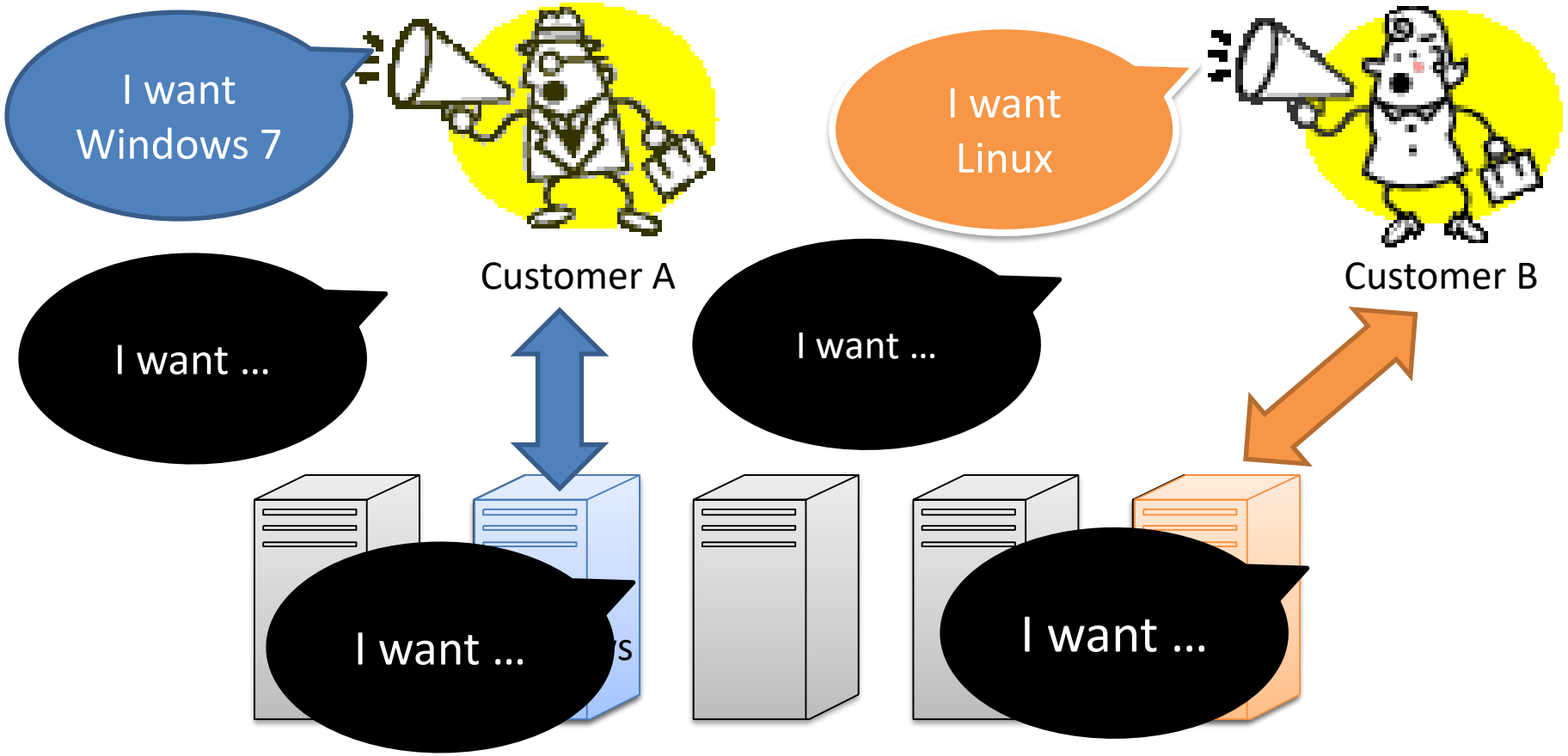
- Clients will request different operating systems.
- Clients will request different storage sizes.
- Clients will request different network bandwidths.
- Clients will change their requests anytime.
- Clients will …

Is there any good strategy ?

- Allocate a new physical machine for each incomer.
- Prepare a pool of pre-installed machines for different requests.
- or …

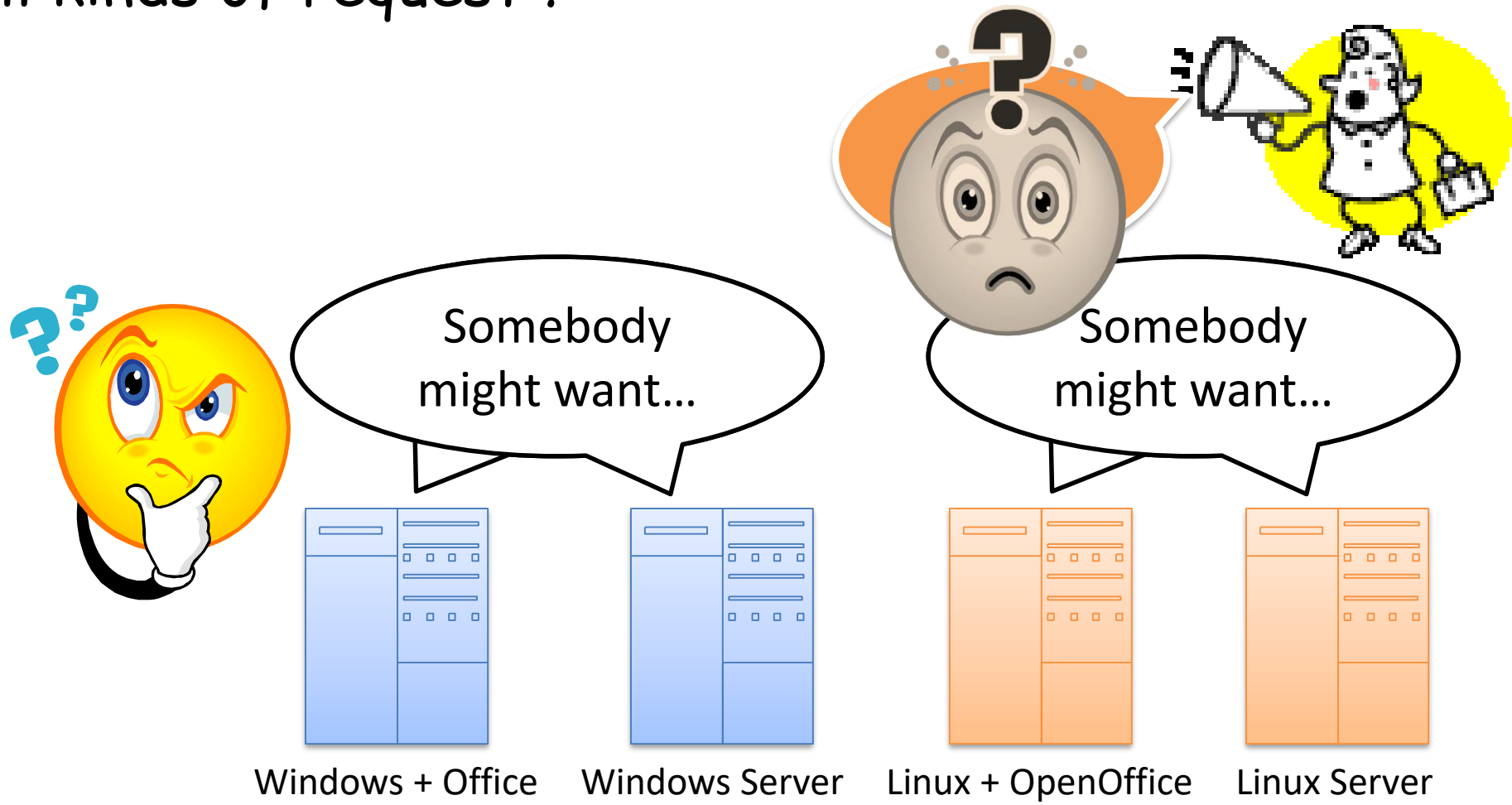
Virtualization

What if we allocate a new physical machine for each incomer ?



Virtualization

How about preparing a pool of pre-installed physical machines for all kinds of request ?



Virtualization

Obviously, neither of previous strategies will work.
We need more powerful techniques to deal with that.

Virtualization techniques will help.

- For computation resources
  - Virtual Machine technique
- For storage resources
  - Virtual Storage technique
- For communication resources
  - Virtual Network technique

Properties and Characteristics

As a cloud provider, all of the fundamental properties and characteristics stated in previous lectures should be concerned and implemented.

Scalability & Elasticity

What do scalability and elasticity mean in IaaS ?

Clients should be able to dynamically increase or decrease the amount of infrastructure resources in need.
Large amount of resources provisioning and deployment should be done in a short period of time, such as several hours or days.
System behavior should remain identical in small scale or large one.

Scalability & Elasticity

How to approach scalability and elasticity in IaaS ?

For computation resources :

- Dynamically create or terminate virtual machines for clients on demand.
- Integrate hypervisors among all physical machines to collaboratively control and manage all virtual machines.

For storage resources :

- Dynamically allocate or de-allocate virtual storage space for clients.
- Integrate all physical storage resources in the entire IaaS system
- Offer initial storage resources by thin provisioning technique.

For communication resources :

- Dynamically connect or disconnect the linking state of virtual networks for clients on demand.
- Dynamically divide the network request flow to different physical routers to maintain access bandwidth.

Availability & Reliability

What do availability and reliability mean in IaaS ?

Clients should be able to access computation resources without considering the possibility of hardware failure.
Data stored in IaaS cloud should be able to be retrieved when needed without considering any natural disaster damage.
Communication capability and capacity should be maintained without considering any physical equipment shortage.

Availability & Reliability

How to approach availability and reliability in IaaS ?

For computation resources :

- Monitor each physical and virtual machine for any possible failure.
- Regularly backup virtual machine system state for disaster recovery.
- Migrate virtual machine among physical machines for potential failure prevention.

For storage resources :

- Maintain data pieces replication among different physical storage devices.
- Regularly backup virtual storage data to geographical remote locations for disaster prevention.

For communication resources :

- Built redundant connection system to improve robustness.

Manageability & Interoperability

What do manageability and interoperability mean in IaaS ?

Clients should be able to fully control the virtualized infrastructure resources which allocated to them.
Virtualized resources can be allocated by means of system control automation process with pre-configured policy.
States of all virtualized resource should be fully under monitoring.
Usage of infrastructure resources will be recorded and then billing system will convert these information to user payment.

Manageability & Interoperability

How to approach manageability and interoperability in IaaS ?

For computation resources :

Provide basic virtual machine operations, such as creation, termination, suspension, resumption and system snapshot.

Monitor and record CPU and memory loading for each virtual machine.

For storage resources :

Monitor and record storage space usage and read/write data access from user for each virtual storage resource.

Automatic allocate/de-allocate physical storage according to space utilization.

For communication resources :

Monitor and record the network bandwidth consumption for each virtual link.

Automatically reroute the data path when computation and storage are duplicated.

Performance & Optimization

What do performance and optimization mean in IaaS ?

Physical resources should be highly utilized among different clients.

Physical resources should form a large resource pool which provide high computing power through parallel processing.

Virtual infrastructure resources will be dynamically configured to an optimized deployment among physical resources.

Performance & Optimization

How to approach performance and optimization in IaaS ?

For computation resources :

- Deploy virtual machine with load balancing consideration.
- Live migrate virtual machines among physical ones to balance the system loading.

For storage resources :

- Deploy virtual storage with hot spot access consideration.
- Live migrate virtual storage among physical ones with different performance level.

For communication resources :

- Consider network bandwidth loading when deploying virtual machines and storage.
- Dynamically migrate virtual machines or storage to balance network flow.

Accessibility & Portability

What do accessibility and portability mean in IaaS ?

Clients should be able to control, manage and access infrastructure resources in an easy way, such as the web-browser, without additional local software or hardware installation.

Provided infrastructure resources should be able to be reallocated or duplicated easily.

Accessibility & Portability

How to approach accessibility and portability in IaaS ?

For computation resources :

Cloud provider integrates virtual machine management and access through web-based portal.

Comply the virtual machine standard for portability.

For storage resources :

Cloud provider integrates virtual storage management and access through web-based portal.

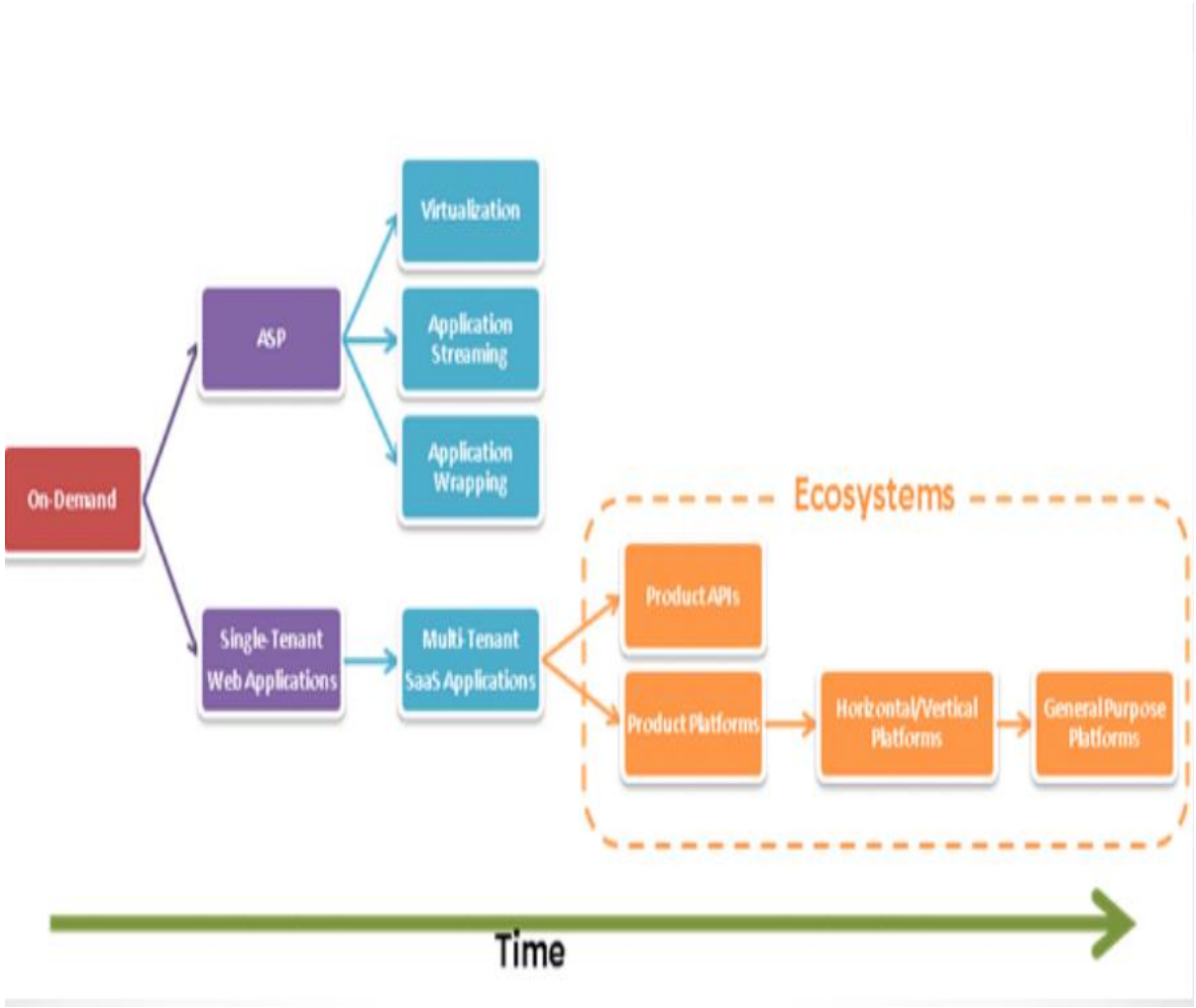
For communication resources :

Cloud provider integrates virtual network management and access through web-based portal.

SaaS Definition

- Software as Service is a software delivery model in which software and data are hosted centrally and accessed via web. It can be rented .
- Configuration and customization according to the customer
- Accelerated feature delivery weekly updates
- Multi tenant architecture single instance of the software for multiple client organizations
- Open integration protocols API'S and protocols for accessing company's internal systems

A Brief History



1960-IBM and other main frame providers came up with time sharing or utility computing services.

1990- ASP with advent of internet

2000- Increased speed of internet resulted in more popularity of SaaS

ADVANTAGES

USER BENEFITS:

- *Lower Cost of Ownership*
- *Focus on Core Competency .*
- *Access Anywhere*
- *Freedom to Choose (or Better Software)*
- *Faster Product Cycles*

VENDOR BENEFITS:

- *Increased Total Available Market*
- *Lower Development Costs & Quicker Time-to-Market*
- *Improved Customer Relationships*

Applicability and types of SaaS

- Enterprise Software application:
  - Sharing of data between internal and external users e.g. : Salesforce CRM application
  - Single user Software application
- Runs on single user computer and serves 1 user at a time e.g. : Microsoft office
- Business Utility SaaS - Applications like Salesforce automation are used by businesses and individuals for managing and collecting data, streamlining collaborative processes and providing actionable analysis. Popular use cases are Customer Relationship Management (CRM), Human Resources and Accounting.
- Social Networking SaaS - Applications like Facebook are used by individuals for networking and sharing information, photos, videos, etc.

Important factors for a good design

Three key differentiators that separate well-designed SaaS application from a poorly designed one

- *scalability*
- *Multi tenant efficient*
- *Configurable*

Scaling the application - maximizing concurrency, and efficient use of resources

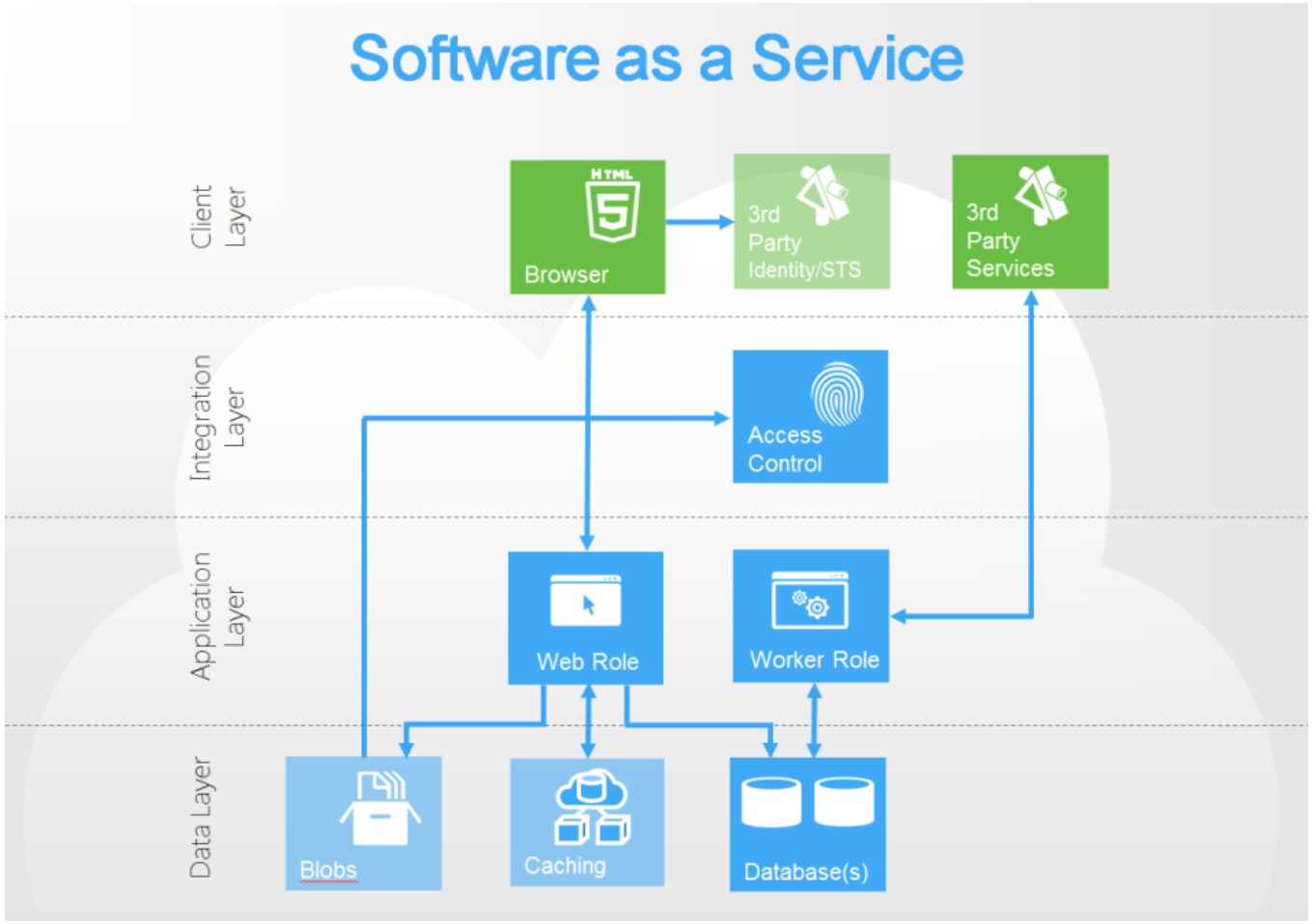
i.e. optimizing locking duration, statelessness, sharing pooled resources such as threads and network connections, caching reference data, and partitioning large databases

Multi-tenancy – important architectural shift from designing isolated, single-tenant applications

One application instance should accommodate users from multiple other companies at the same time while providing transparency
This requires an architecture that maximizes the sharing of resources efficiently across tenants

- *Configurable* - a single application instance on a single server has to accommodate users from several different companies
  - Customizing the application for one customer will change the application for other customers as well.
- Traditionally customizing an application would mean changes in the code.
- Each customer must use metadata to *configure* the way the application appears and behaves for its users.
- Customers configuring applications must be simple and easy without any extra development or operation costs

Windows Azure



Azure Architecture

Web role
One for each instance of software
Access control
Definition of users, groups and roles. A pre-built ASP.NET membership provider is included in the training kit.
Databases
Relational database for core operational data
Worker role
Autonomous background processing like billing
Caching
frequently used read-only, user specific, and application resource data in a high-speed distributed in-memory for faster response
Blobs
Blob storage provides a scalable, resilient way to store terabytes of user data

SaaS Drawbacks

Robustness

- Difference between Google docs and Microsoft office.

Privacy

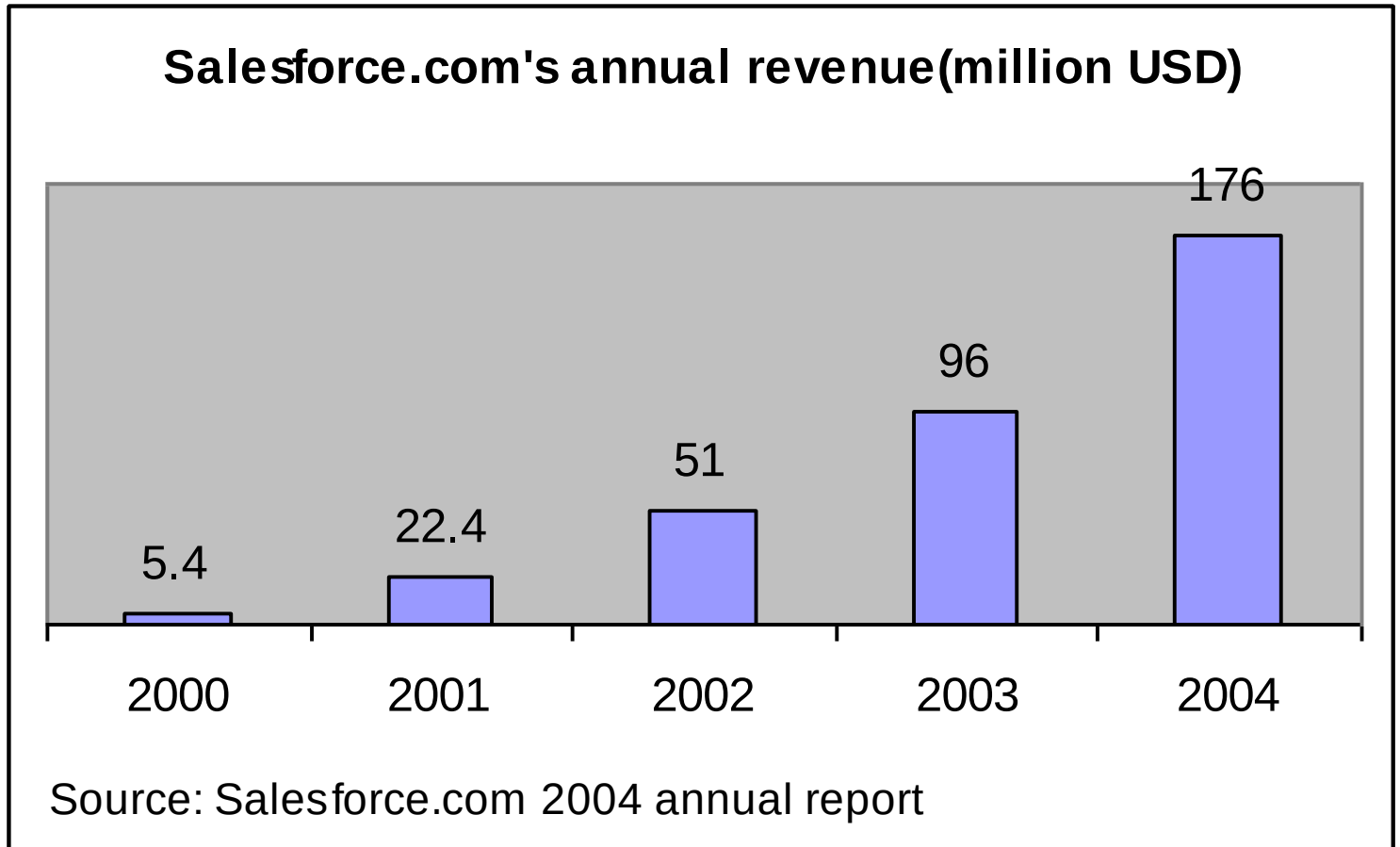
- Storing all data in cloud prone to hacks

Reliability

- Recovery during server downtime is difficult

Salesforce.com

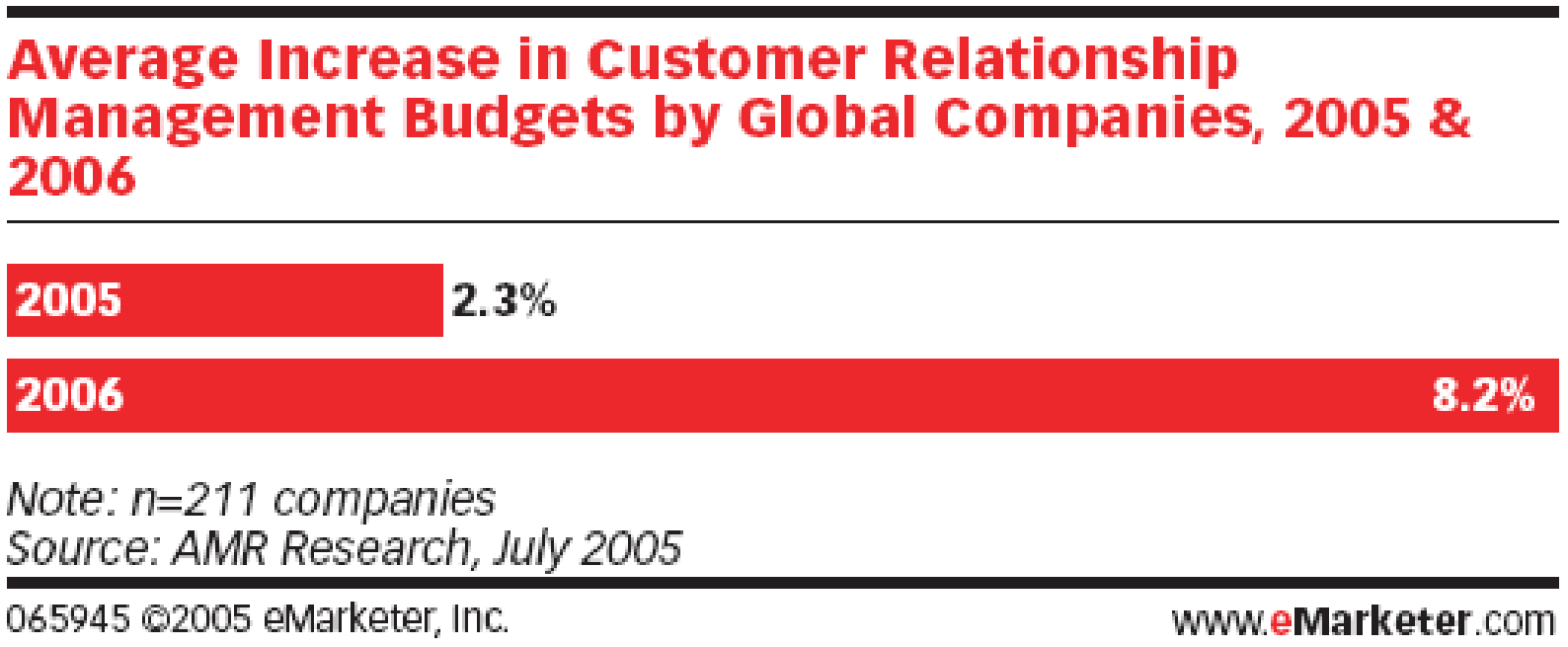
Founded in 1999
Annual growth rate is over 80%
Went to IPO in 2004



Source: Salesforce.com 2004 annual report

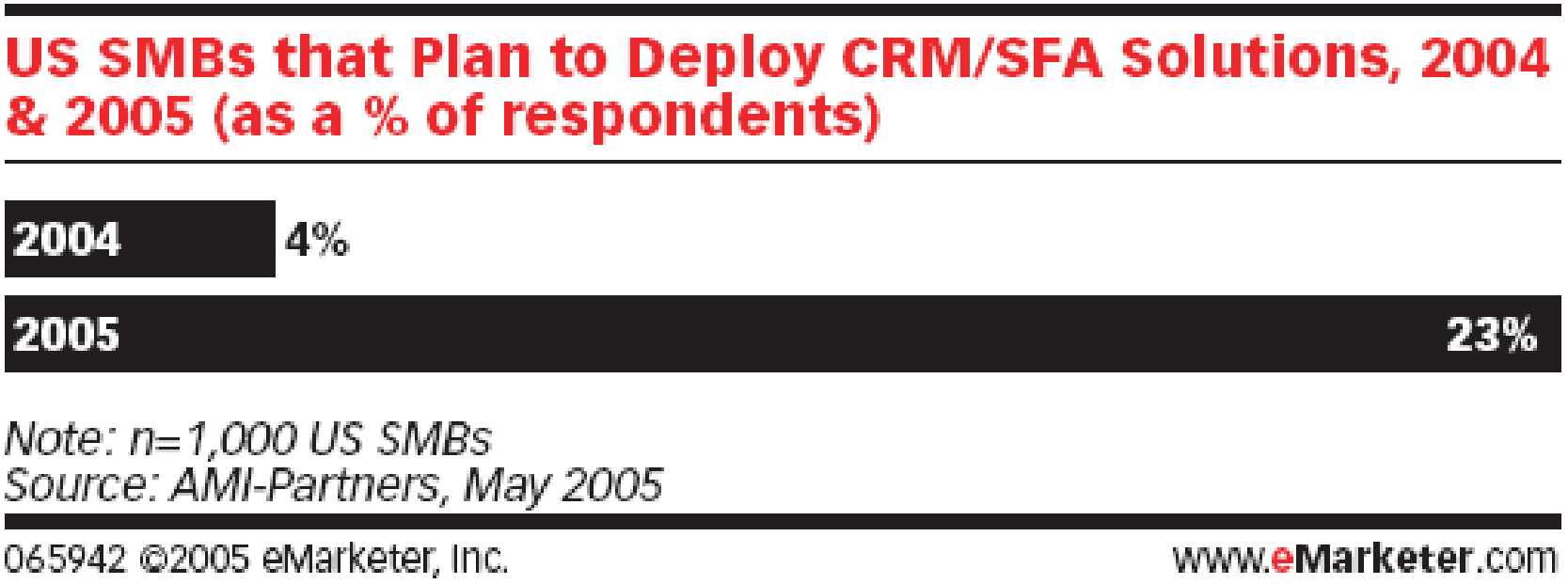
Software Market Trend

Large enterprises:



Software Market Trend(Continue)

Small-to-medium businesses:



Software Market Trend(continue)

Business Trend

Smaller package with
flexible coupled component

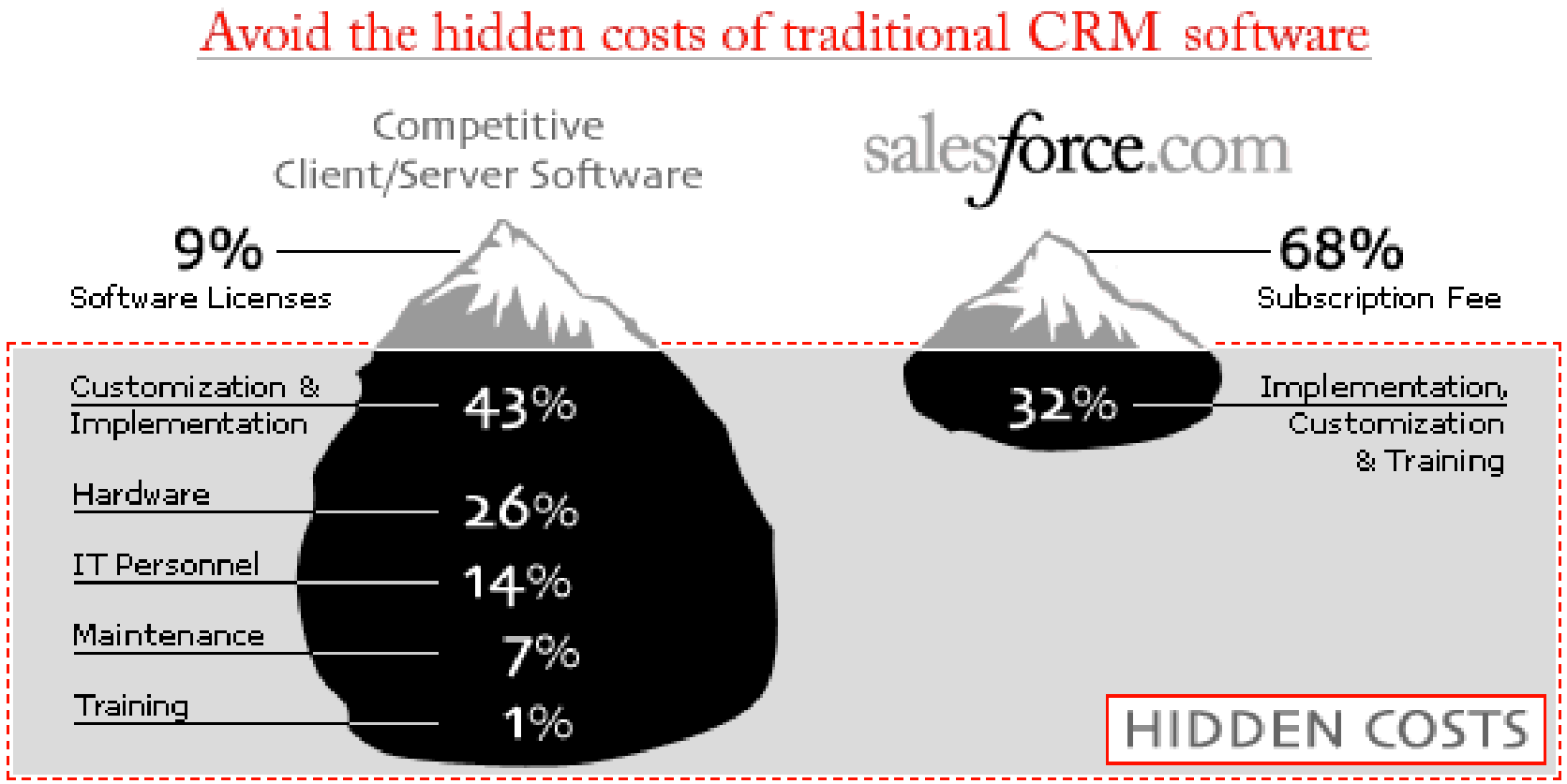
Cost effective
Easily deployed,
reconfigured and updated
Standardized industrial
application

Traditional enterprise Software

Delivered as a large package

Charge high license fee
High deployment and
maintenance fee

Hidden Cost


Avoid the hidden costs of traditional CRM software
Competitive
Client/Server Software
salesforce.com
9%
Software Licenses
68%
Subscription Fee
Customization &
Implementation
43%
Hardware
26%
IT Personnel
14%
Maintenance
7%
Training
1%
32%
Implementation,
Customization
& Training
HIDDEN COSTS

What kind of software is more amenable to be provided as a service?

Applicability of Saas

Enterprise Software Application

Perform business functions
Organize internal and external information
Share data among internal and external users
The most standard type of software applicable to Saas model
Example: Saleforce.com CRM application, Siebel On-demand application

Applicability of Saas(Continue)

Single-User software application

- Organize personal information
- Run on users' own local computer
- Serve only one user at a time
- Inapplicable to Saas model
  - Data security issue
  - Network performance issue
- Example: Microsoft office suite

Applicability of Saas(Continue)

Infrastructure software

- Serve as the foundation for most other enterprise software application
- Inapplicable to Saas model
  - Installation locally is required
  - Form the basis to run other application
- Example: Window XP, Oracle database

Applicability of Saas(Continue)

Embedded Software

Software component for embedded system
Support the functionality of the hardware device
Inapplicable to Saas model
    Embedded software and hardware is combined together and is inseparable
Example: software embedded in ATM machines, cell phones, routers, medical equipment, etc

Take-home Message

- We have introduced three basic service models, namely PaaS, IaaS, and SaaS
- The basic understanding shall be remembered as its pros and cons.
- Remember to do your homework, and no class next week.

References

- Creeger, M. (2009). Cloud computing: An overview. ACM Queue, 7(5), 3-4.
- OpenShift – www.openshift.com
- http://www.slideshare.net/cobiacomm/introduction-topaa-s
- http://www.ibm.com/developerworks/training/kp/cl-kp-cloudsaas/

References

- Glitten, Sandra. "Finance's role in controlling use of SaaS". CFOworld. 4 April, 2011. <http://www.itworld.com/saas/152477/finances-role-controlling-use- saas>.
- Gray, Mike. "Cloud Computing: Demystifying IaaS, PaaS, and SaaS". SPS Commerce. 21 October 2010. <http://www.zdnet.com/news/cloud- computing-demystifying-iaas-paas-and-saas/477238>.
- Gruman, Galen. "The Truth About Software as a Service (SaaS)". 21 May 2007, 1-8. <http://www.cio.com/article/109706/The_Truth_About_Software_as_a_Servi ce_SaaS_>.
- Software & Information Industry Association. "Software as a Service: Strategic Backgrounder". February 2001. Print

Acknowledgement

Some slides are based on papers in references and

- Operating System Concepts, 7th ed., by Silbershatz, Galvin, & Gagne
- Modern Operating Systems, 2nd ed., by Tanenbaum
- http://web.cs.wpi.edu/~cs502
- https://www.cs.purdue.edu/homes/bb/cloud/
- U of Buffalo cse487
- U of Florida eel6686