



# COS 448: Innovating Across Technology, Business, & Markets

Spring 2011

MW 1100-1220 in CS105

Prof. JP Singh

TA: Muneeb Ali

# Road Map: Weekly

---



## Trends-1: Market Places

- What are online market places
- Case Study: Online advertising

## Trends-2: Cloud Computing

- How to build large market places
- What technology is available, where is it heading

## Market Places: continued ...

# Road Map: Today's Lecture

---



## Case Study: Online Advertising:

- How to implement at scale?
- Different needs (e.g., data analytics vs. real-time ad serving)
- Design goals (e.g., performance vs. scalability)

## Leveraging the Cloud:

- What is Cloud Computing?
- 1999 vs. 2009
- Why should you, the CTOs, care?

## Technologies behind the Cloud:

- Datacenters; history, benefits
- Virtualization; why important, benefits
- Design tradeoffs

# Online Advertising

---



## Consider the Functionality:

- Display ads
- Apps/UI for advertisers,
- Apps/UI for publishers and ad networks
- Ad serving (including real-time auctions)
- Response prediction (computed offline, used real-time)
- Billing systems, etc.

## Example: DoubleClick

- DART for Advertisers
- DART for Publishers
- DoubleClick Ad Exchange
- Performics (now DART Search)



# Online Advertising

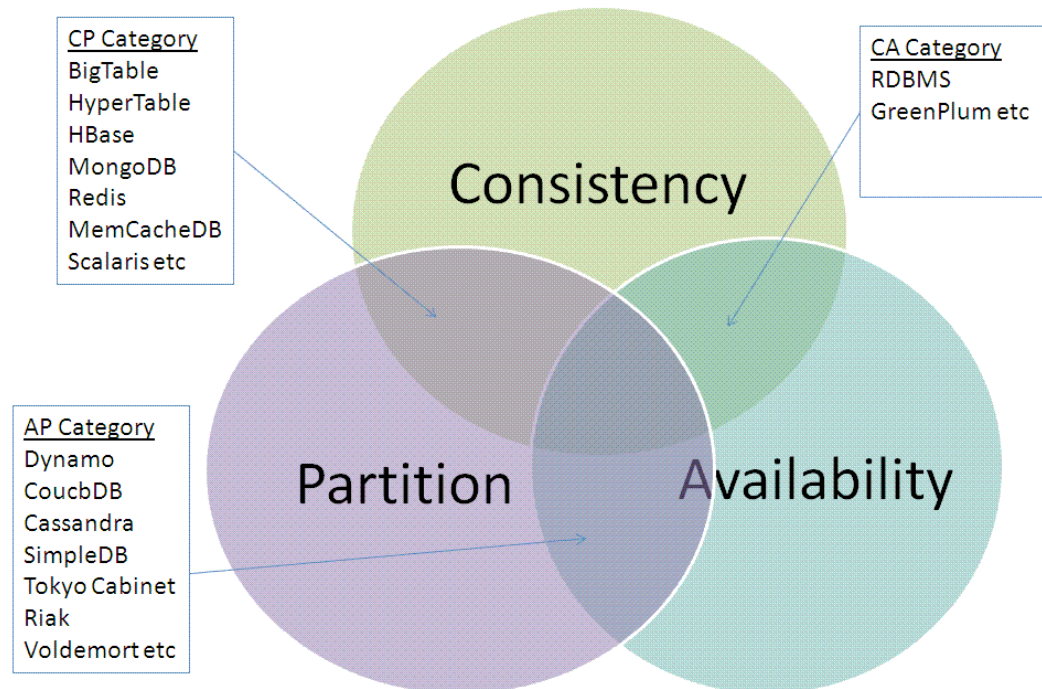


## Design Goals:

- Performance (speed, response time)
- Scalability (how performance grows with nodes added)
- Uptime (service availability)
- Reliability (recovery from failures)
- Cost

## Conflicting Goals:

- Cost vs. Performance
- Scalability vs. Reliability
- CAP Theorem



# Online Advertising



## 1999 vs. 2009:

- What has changed in terms of technology?
- What new options are available?

## 1999:



Buy enterprise servers

+



Proprietary software

+ Infrastructure management

+ Manual scale

# Online Advertising



1999:



Buy enterprise servers

+



Proprietary software

- + Infrastructure management
- + Manual scale

2009:



Lease computing + storage

+



Open source

- + Don't manage infrastructure
- + Scale on the fly
- + Other ...

# Online Advertising



1999:



Buy enterprise servers

+



Proprietary software

+ Infrastructure management

+ Manual scale

2009:

Welcome to the cloud!

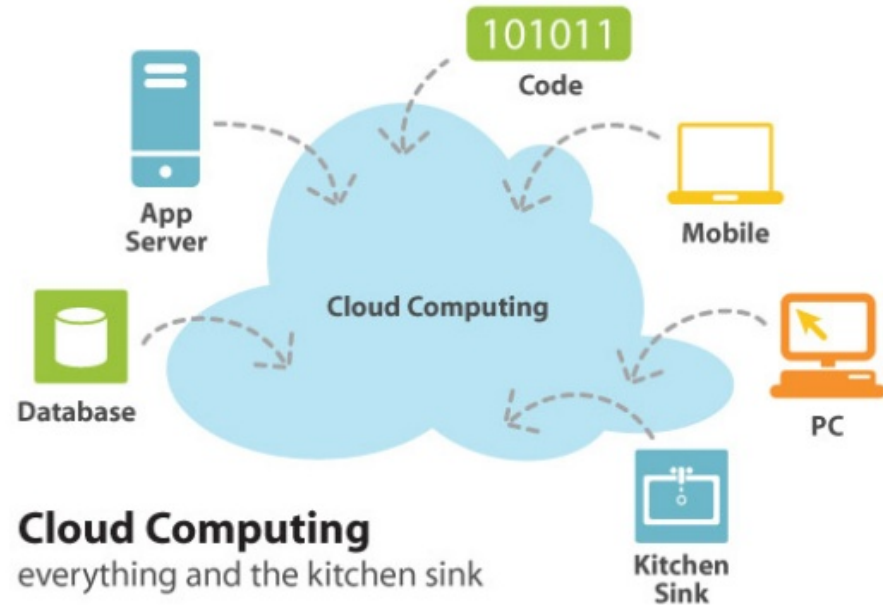




# Leveraging the Cloud



“Magic in the sky” view →



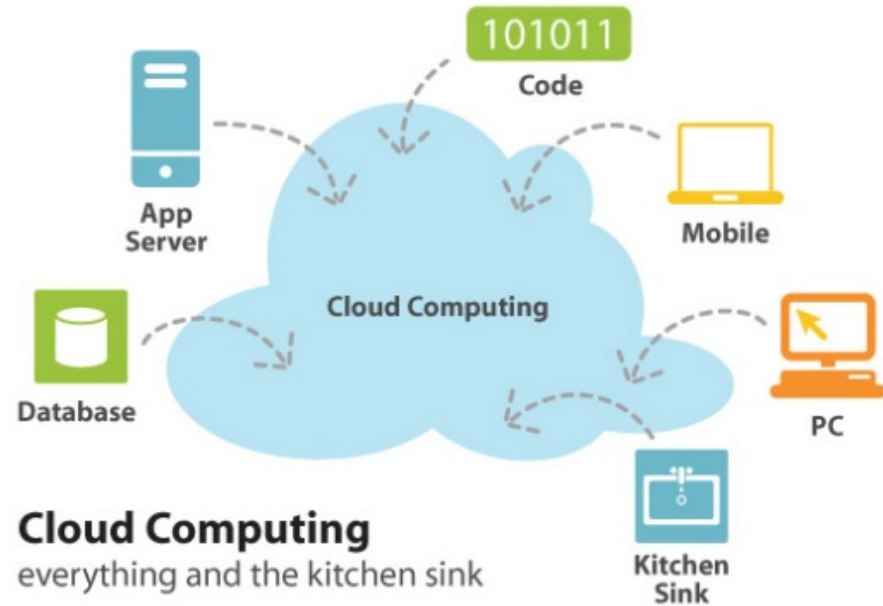
## Other Views:

- Platform as a Service + Software as a Service = Cloud Computing
- Datacenters + Virtualization + Services (APIs) = Cloud Computing

# Leveraging the Cloud



Let's decode this  
"magic in the sky" picture ...



## Cloud Computing:

- What exactly is Cloud Computing?
- Why should you, the CTOs, care?
- How to build marketplaces using these new technologies?



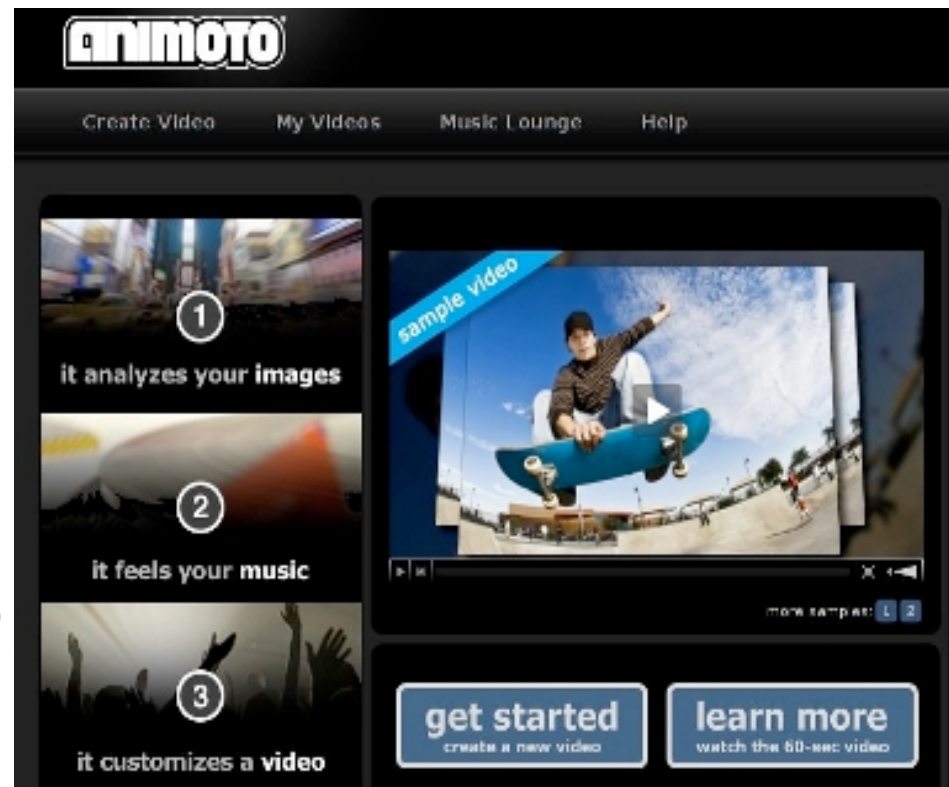
# Leveraging the Cloud

- Advances in cloud computing are (largely) powered by:

- a) Datacenters
- b) Virtualization
- c) Remote Services (APIs)

## Example: Animoto

- Personalized videos
- 25k to 250k users in 3 days!
- Amazon's infrastructure (AWS)



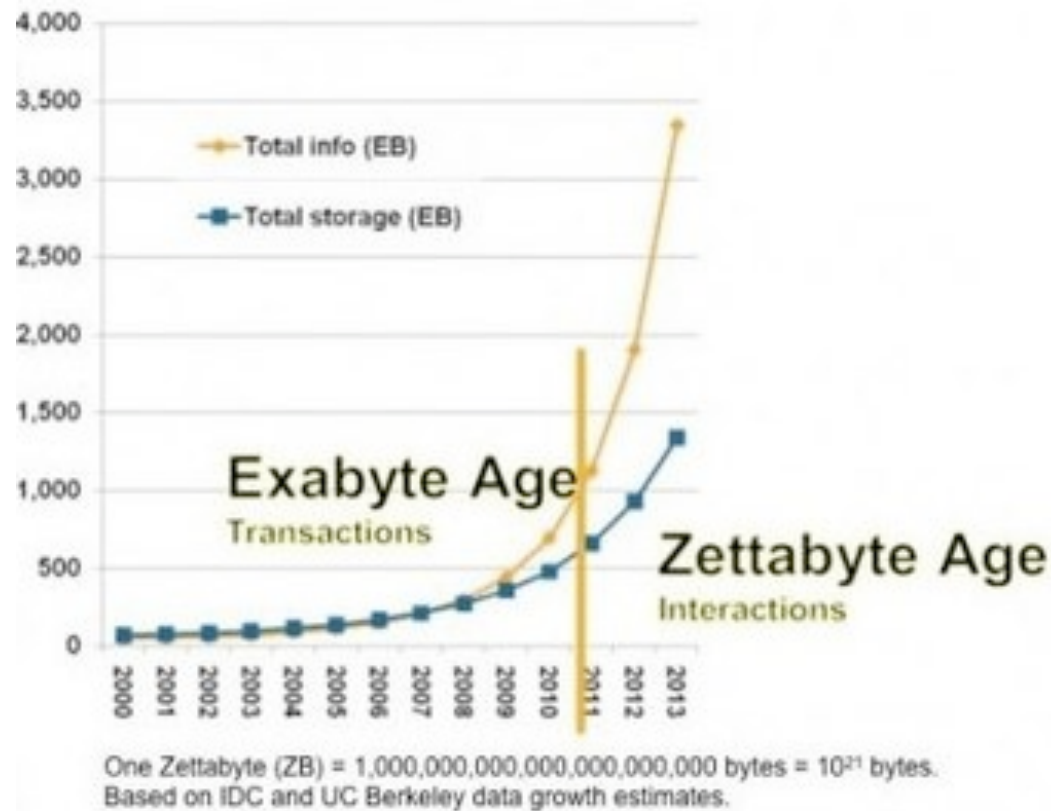
# What is a Datacenter

---



Google's datacenter in the Dalles, Oregon (350k servers)  
Each the size of a football field!

# Why Need Datacenters



More data has been created in the last 3 years than last 40,000!

# Inside Datacenters

---



There is a mini version of this in the CS department's basement!

# Cloud Services

---



- Two models of cloud services

a) Application-specific



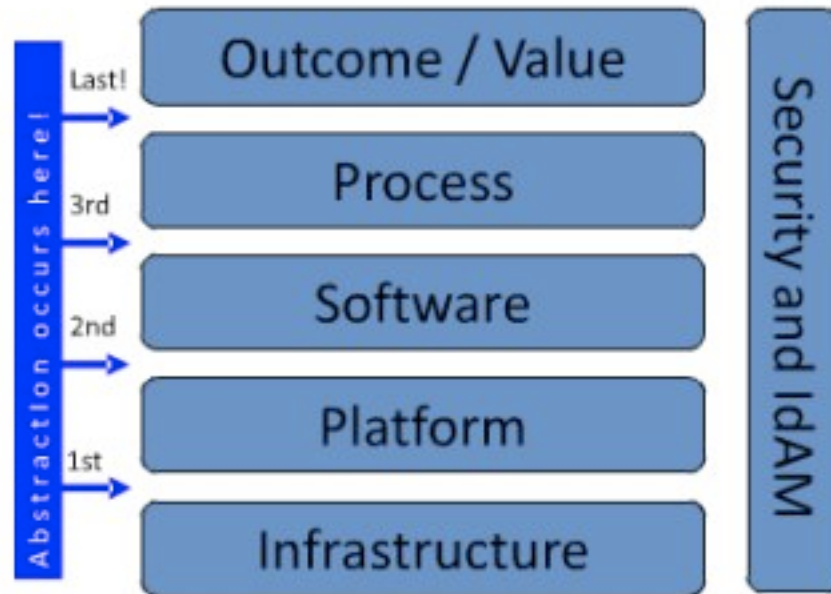
b) Generic



# Cloud Services: Abstraction

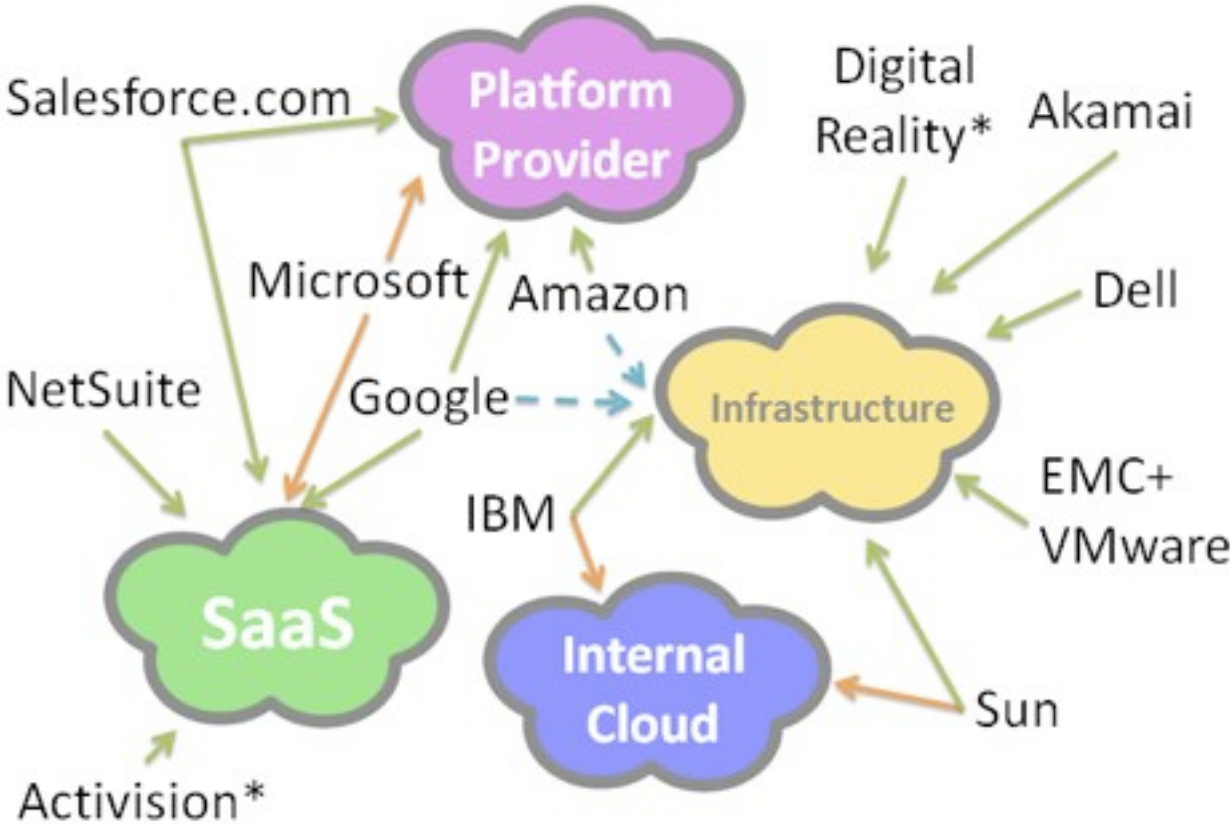


## Cloud Layers

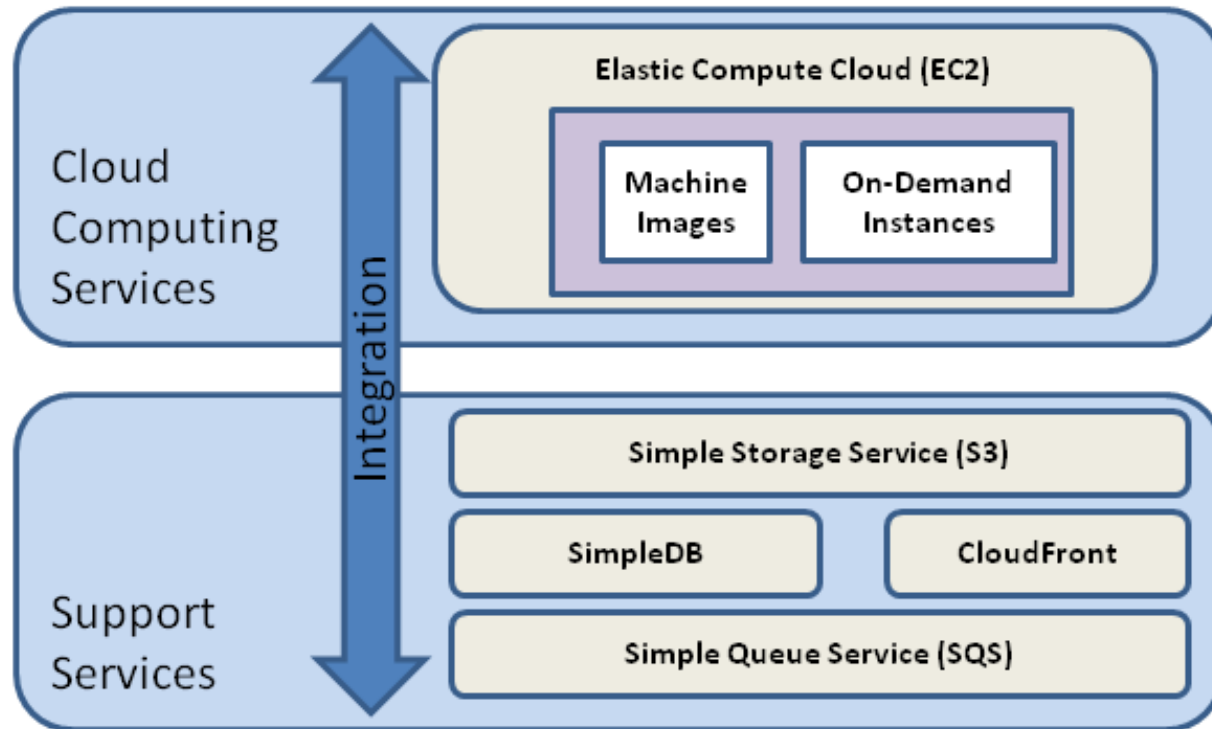




# Cloud Services: Ecosystem



# Cloud Services: Example



# Cloud Services

---



Before we move forward, need to understanding a very important concept:

VIRTUALIZATION

# Virtualization



## Virtualization: 30,000 feet overview

- Hardware that is actually software!
- Software still thinks it's hardware



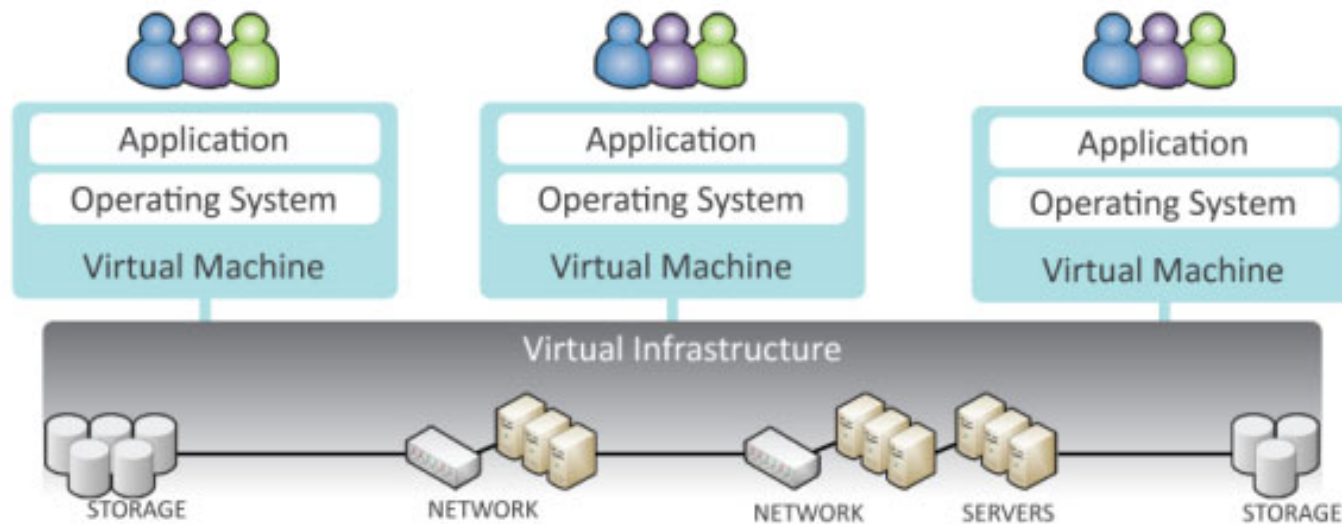
# Virtualization



Before Virtualization:



After Virtualization:



# Virtualization: Benefits

---



## Isolation:

- Limits security exposure
- Other people's bugs don't hurt you on the shared infrastructure

## Backups:

- VM state is usually just a file, easy to save it and have backups
- Easy to create "sandbox" environments for testing

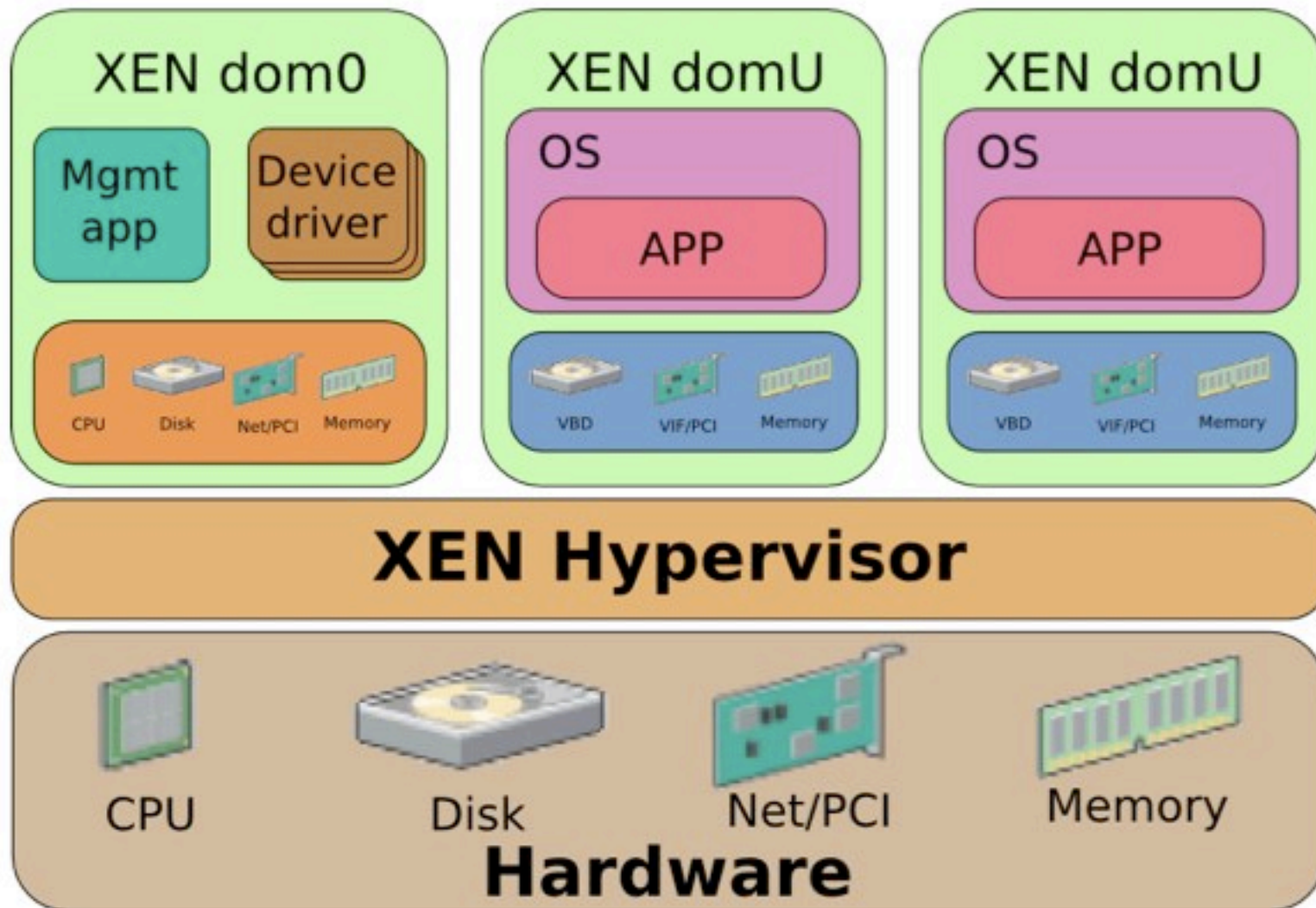
## Deployment:

- Easy to deploy new VMs from base images
- Easy to manage deployed VMs by using resource pools

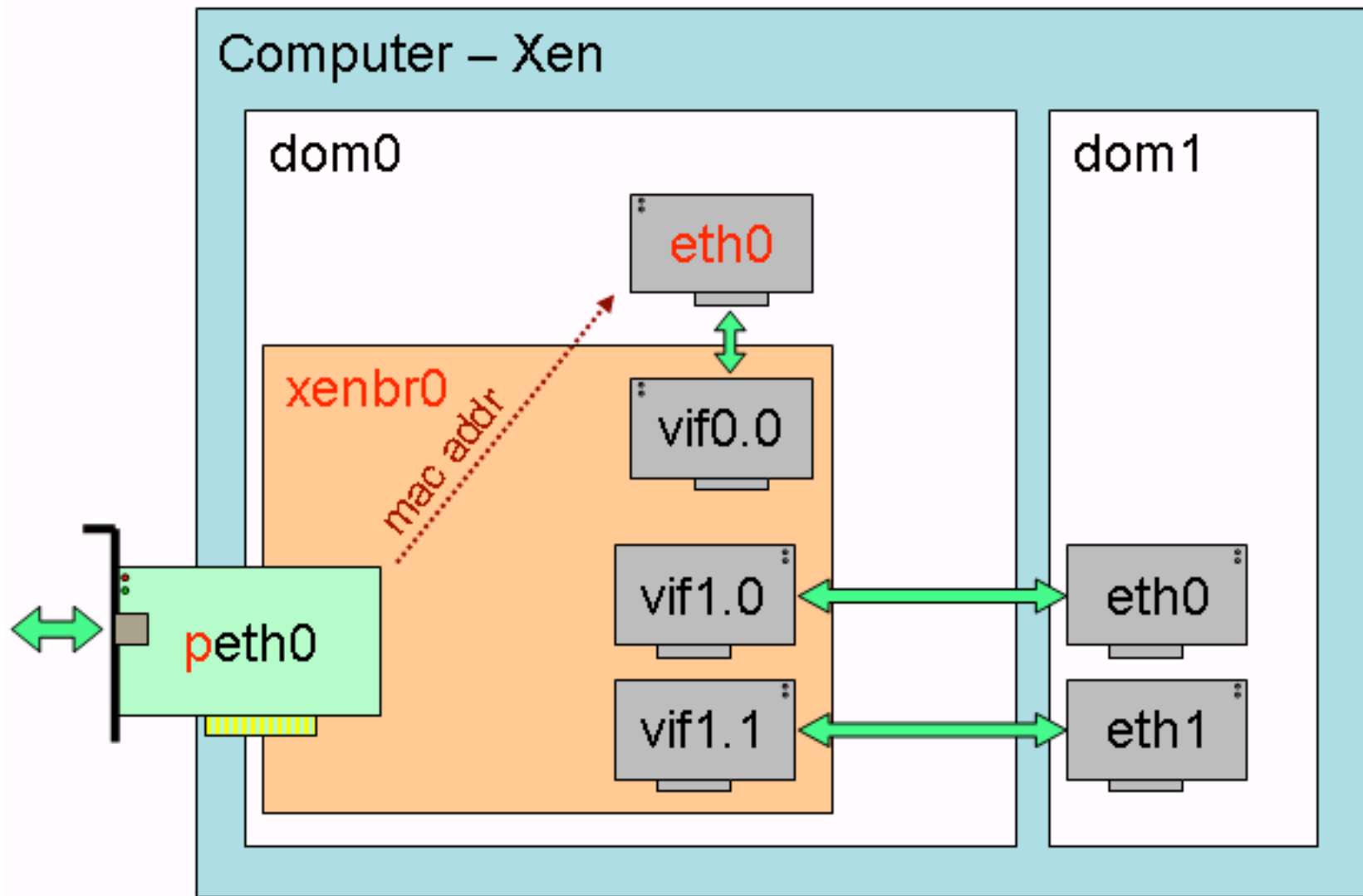
## Portability:

- Reduce dependencies, as each service can run it's own environment
- Move from one physical machine to another with ease

# Virtualization: Technology



# Virtualization: Technology

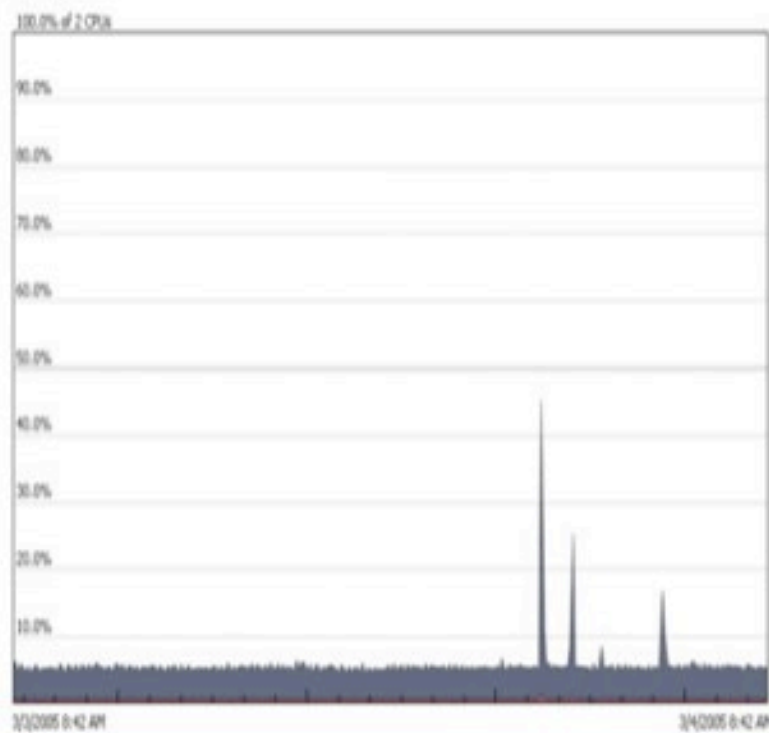




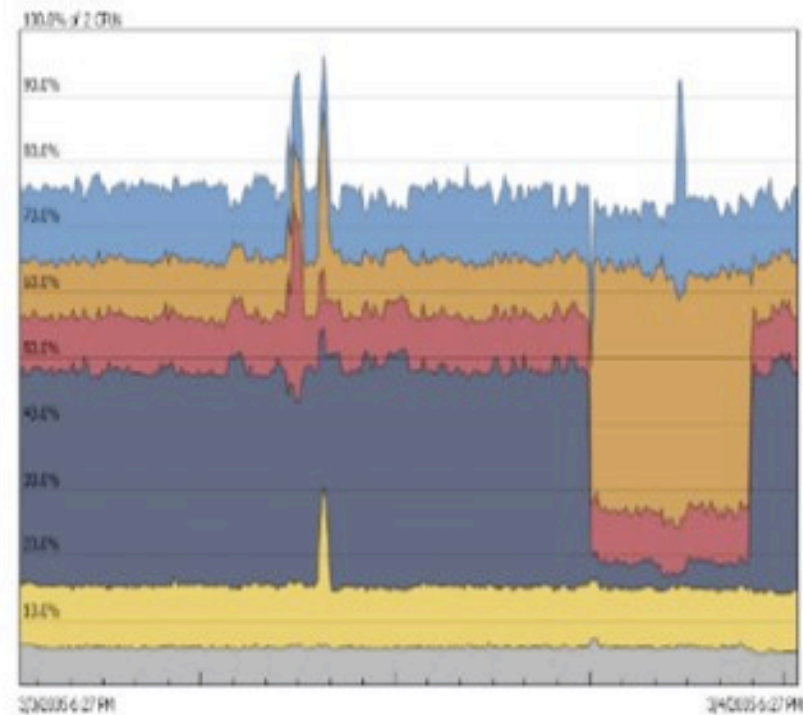
# Virtualization: Benefits



- Server consolidation
- Buy enterprise servers vs. Lease computing power (shared)
- Performance vs. Cost



**Dedicated Application**

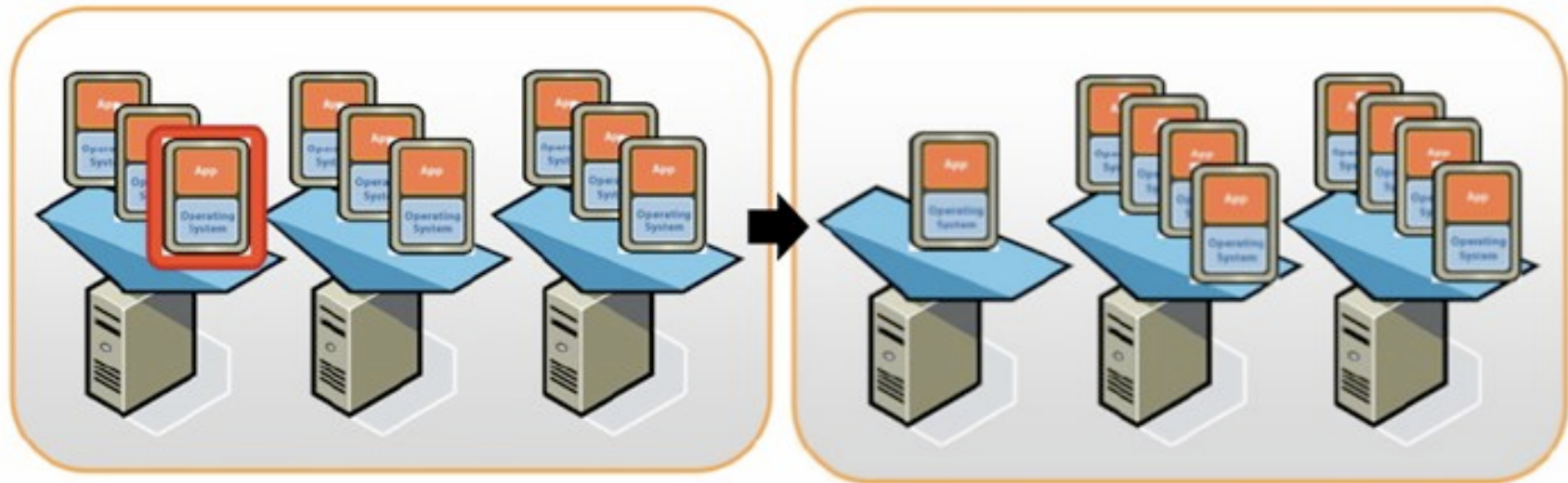


**Virtualized Applications**

# Virtualization: Benefits



- Dynamic load balancing
- For example “follow the moon” to save energy costs



# Virtualization: Benefits



- Failure



- Server is going down

- Move!

# Putting it All Together

---



Let's build a marketplace using the cloud!



# Putting it All Together

---



## Design Choice 1: Processing

- Own infrastructure or use processing in the cloud?
- Private cloud or platform as a service?
- And if you are a startup instead of a big company?



## Design Choice 2: Storage

- Use specialized storage (SAN boxes) or commodity hardware?
- Storage in the cloud or local/specialized?
- Co-locate storage and processing?
- How to perform backups?
- What consistency models are required?

# Putting it All Together

---



## Design Choice 3: OS Environment

- Virtualized or Native?
- Open source or proprietary?
- Who will maintain the environment and upgrade it?



## Design Choice 4: Networking

- Who will I need to communicate to? How often?
- How far are they?
- Replicate services near traffic sources or use CDNs?
- Consistency vs. performance?

# Putting it All Together

---



## Design Choice 5: Databases

- How to store data?
- Performance vs. consistency?



## Design Choice 6: Load Balancing

- How will I scale?
- Is availability more important or consistency?
- How can I use cloud services to help balance load with less cost?

## Design Choice 7: Failure

- How critical is uptime? Lose millions if service goes down for 1 min?
- Recover from network failure? Power? Server crash? Software bugs?