Cloud Computing

SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

The Vision

- In 1969, Leonard Kleinrock, one of the chief scientists of the one of the chief scientists of the original Advanced Research Projects Agency Network (ARPANET) predicted Computer utilities like water, electricity
- don't care where my servers are, who manages them, where my documents are stored, or where my applications are hosted. I just want them always available and access them from any device connected through Internet. And I am willing to pay for this service for as much as I need it.
- IT services are traded as utilities in an open market
- No infrastructure needed on consumer side

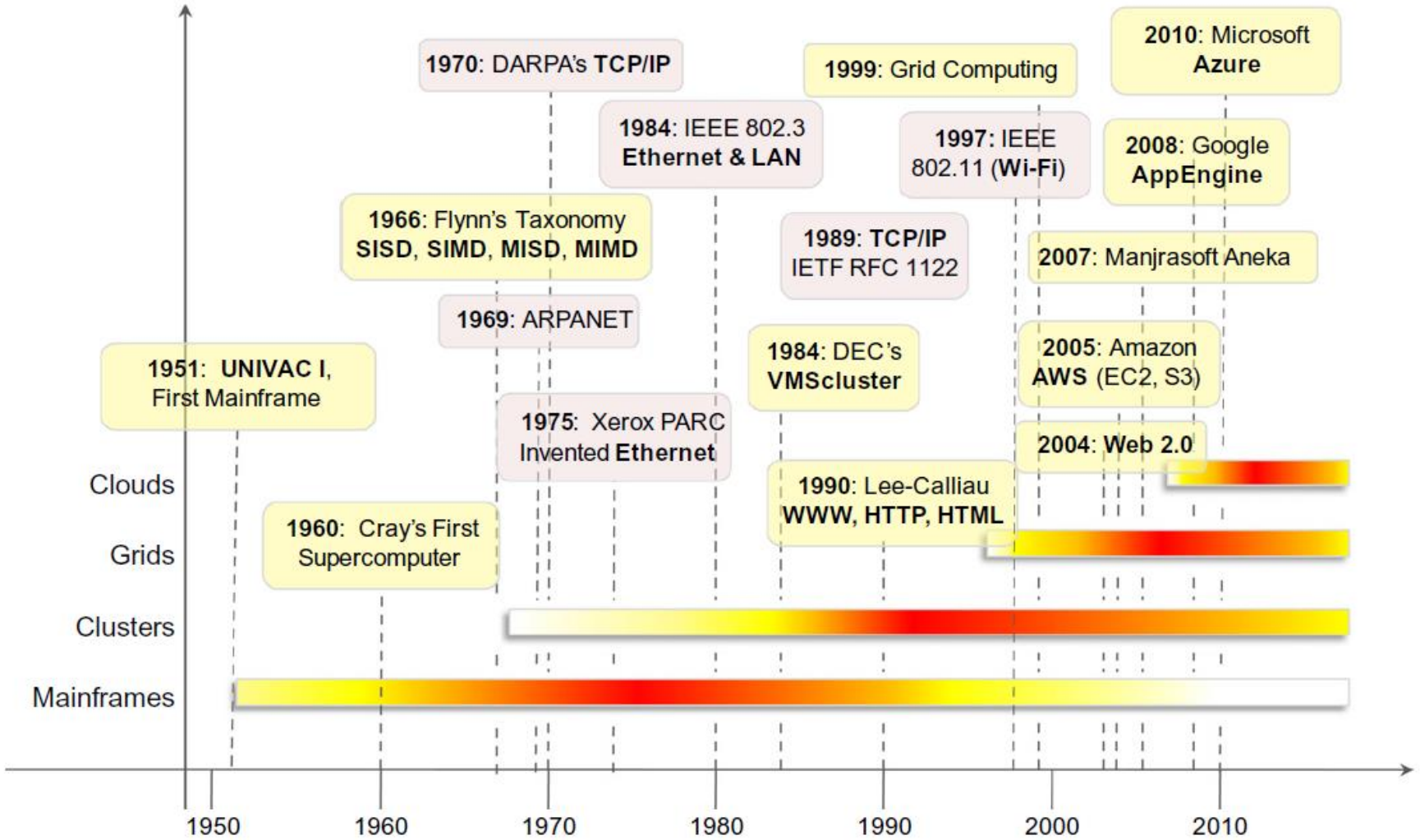
Definition

- Internet plays important role
- Armbrust: Cloud computing refers to both the applications delivered as services over the Internet and the hardware and system software in the datacenters that provide those services.
- Gartner: a style of computing in which massively scalable IT-enabled capabilities are delivered as a service to external customers using Internet technologies
- U.S. National Institute of Standards and Technology (NIST): Cloud computing is a model for enabling ubiquitous, 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.
- Zero capital expenditure is necessary to get started.

Benefits

- No up-front commitments
- On-demand access
- Nice pricing
- Simplified application acceleration and scalability
- Efficient resource allocation
- Energy efficiency
- Seamless creation and use of third-party services

Evolution


1970: DARPA's TCP/IP
1999: Grid Computing
2010: Microsoft Azure
1984: IEEE 802.3 Ethernet & LAN
1997: IEEE 802.11 (Wi-Fi)
2008: Google AppEngine
1966: Flynn's Taxonomy SISD, SIMD, MISD, MIMD
1989: TCP/IP IETF RFC 1122
2007: Manjrasoft Aneka
1969: ARPANET
1951: UNIVAC I, First Mainframe
1984: DEC's VMScluster
2005: Amazon AWS (EC2, S3)
1975: Xerox PARC Invented Ethernet
2004: Web 2.0
1990: Lee-Calliau WWW, HTTP, HTML
Clouds
1960: Cray's First Supercomputer
Grids
Clusters
Mainframes
1950
1960
1970
1980
1990
2000
2010

Mainframe

Web 2.0

- Web is the primary interface
- Interactivity and flexibility into web pages
- XML, AJAX, Web Services
- Variety of devices: mobile phones, car dashboards, TV sets, IoT
- Extremely dynamic applications, lightweight, loose coupling
- Wide availability via media accessibility or affordability
- Google maps, flickr, facebook, twitter, youtube

Evolution (contd)

- Service oriented
- Utility oriented
- Grid computing
- HPC
- Parallel Programming
- Distributed Computing

Challenges

- How many and how long for maximizing benefit
- Tradeoff between availability and price
- Longevity – how long your provider will be in business?
- Business Continuity: 9/11, multi region
- SLA
- Quality of Service (QoS): measure of user experience
- Security
- Regulatory, Legal

Summary

- Vision
- Definition
- Benefits
- Evolution
- Challenges

Assignment

- Signup for AWS
- Create Ubuntu VM
- Create private – public keypair
- Connect to VM using putty and .ppk
- Sudo apt install apache2
- Edit /var/www/html/index
- Edit inbound rules to allow port 80
- Connect to <your VM's public address> from browser

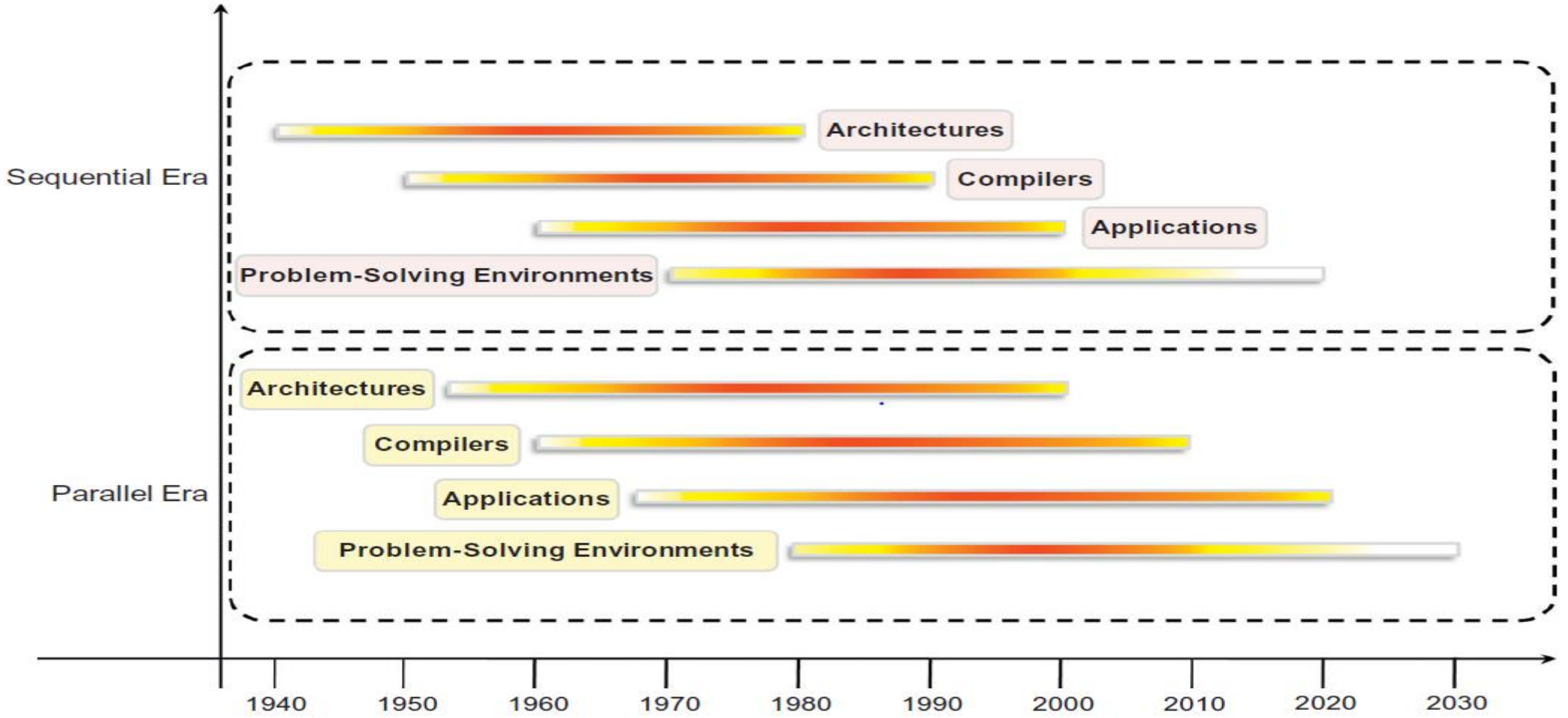
Complete video: https://youtu.be/RLeQ9lNROr0

Parallel and Distributed Computing

SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

Computing Era



Factors for parallel and distributed computing

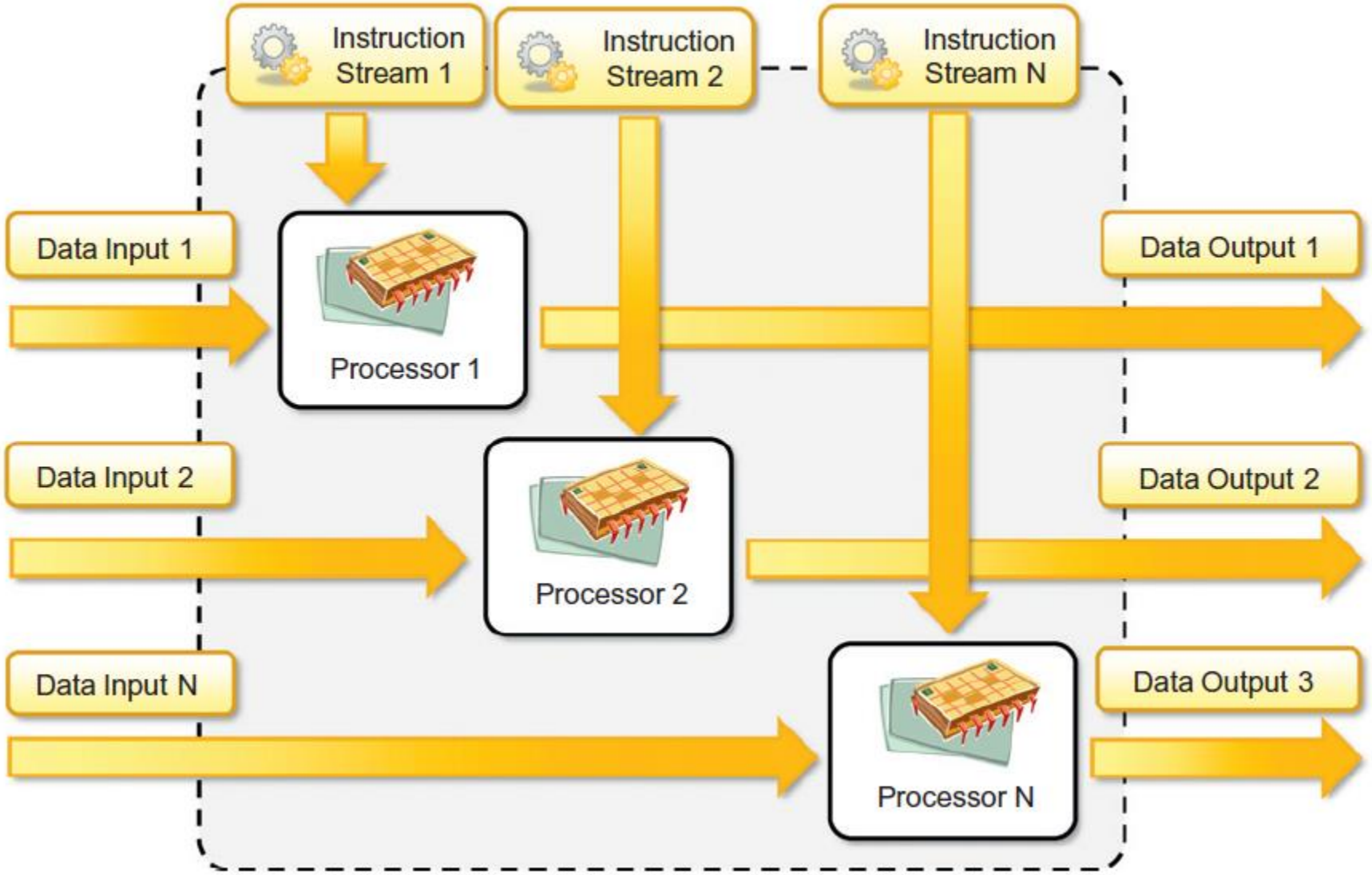
- Ever increasing computation requirements
- Saturation reached of sequential computation
- Hardware improvements in pipelining are non scalable, required compiler technologies are difficult
- Vector processing is useful only for scientific applications with matrices and not generic e.g. for databases
- Improvements in networking
- Parallel processing technology is mature, commercially viable, wide support with development tools and environments
- Scaling: scale up, scale out

Parallel and Distributed Computing

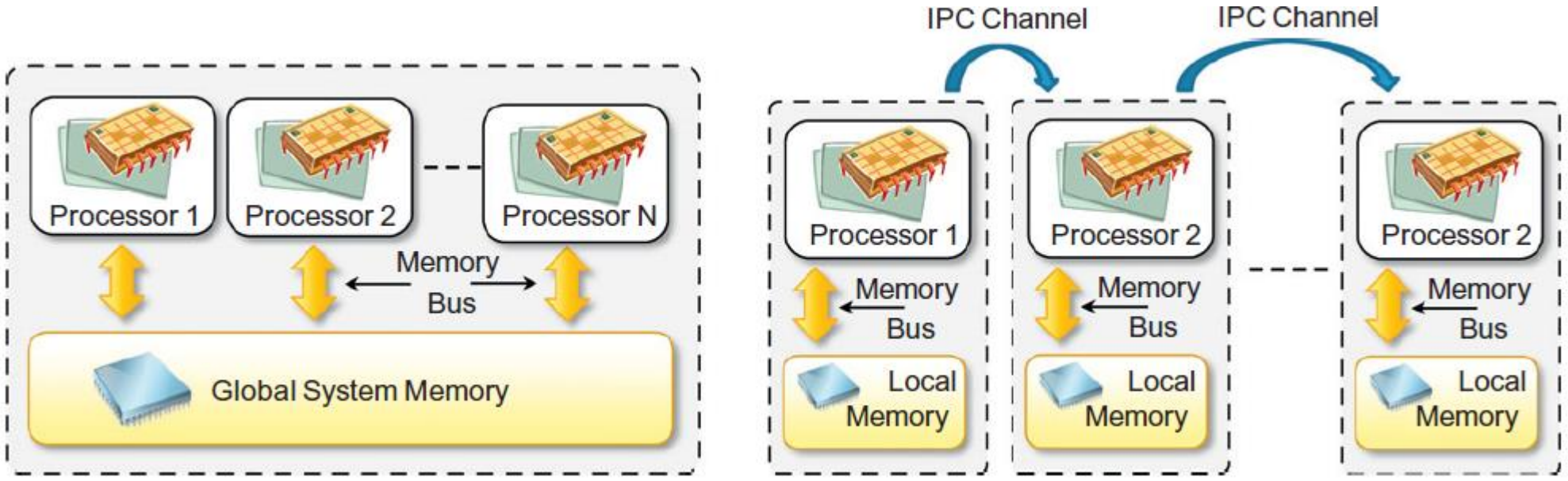
- Parallel
  - Computation is divided among several processors
  - Sharing the same memory
  - Homogeneity of components
  - Distributed shared memory connected by InfiniBand
- Distributed
  - Computation broken down into units and executed concurrently
  - Wider range of systems and applications
  - Heterogeneous

Hardware architectures for parallel processing

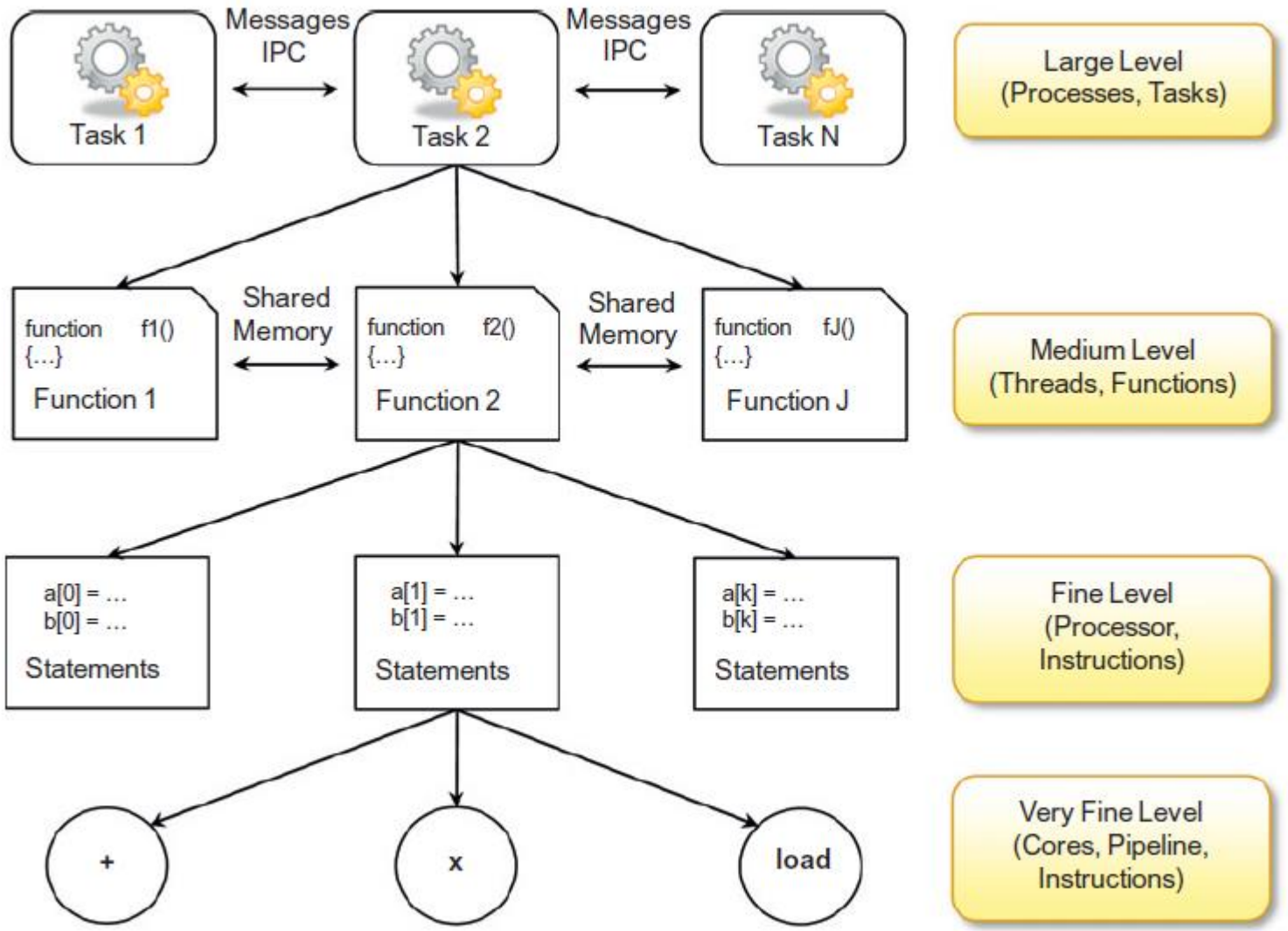
- SISD
- SIMD
- MISD
- MIMD
- Data / Process Parallelism



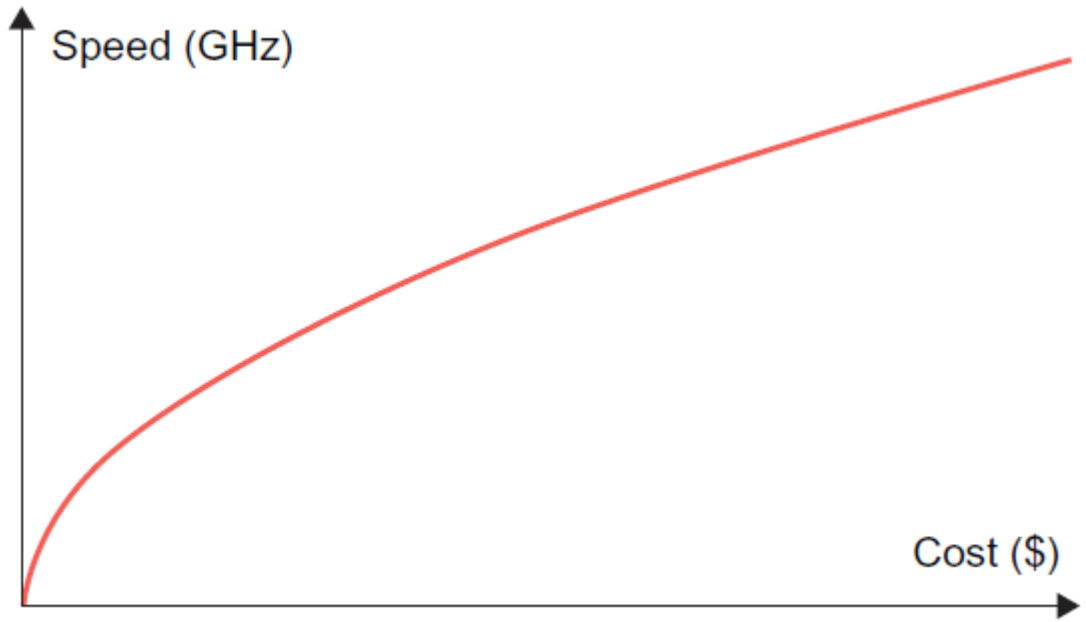
Shared and Distributed Memory MIMD



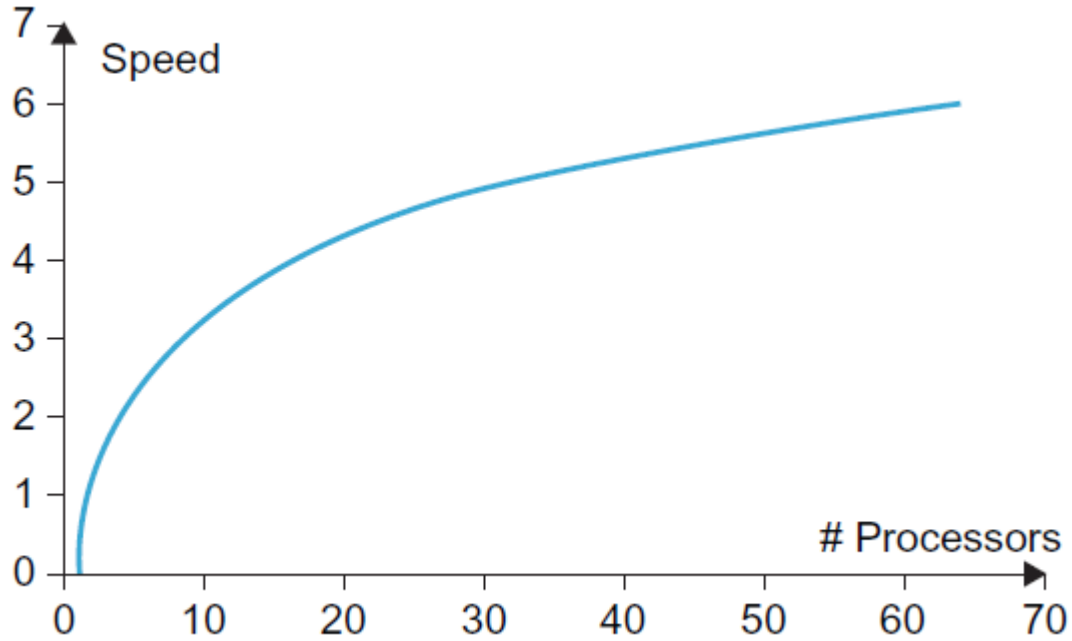
Levels of Parallelism



Speed up grapahs

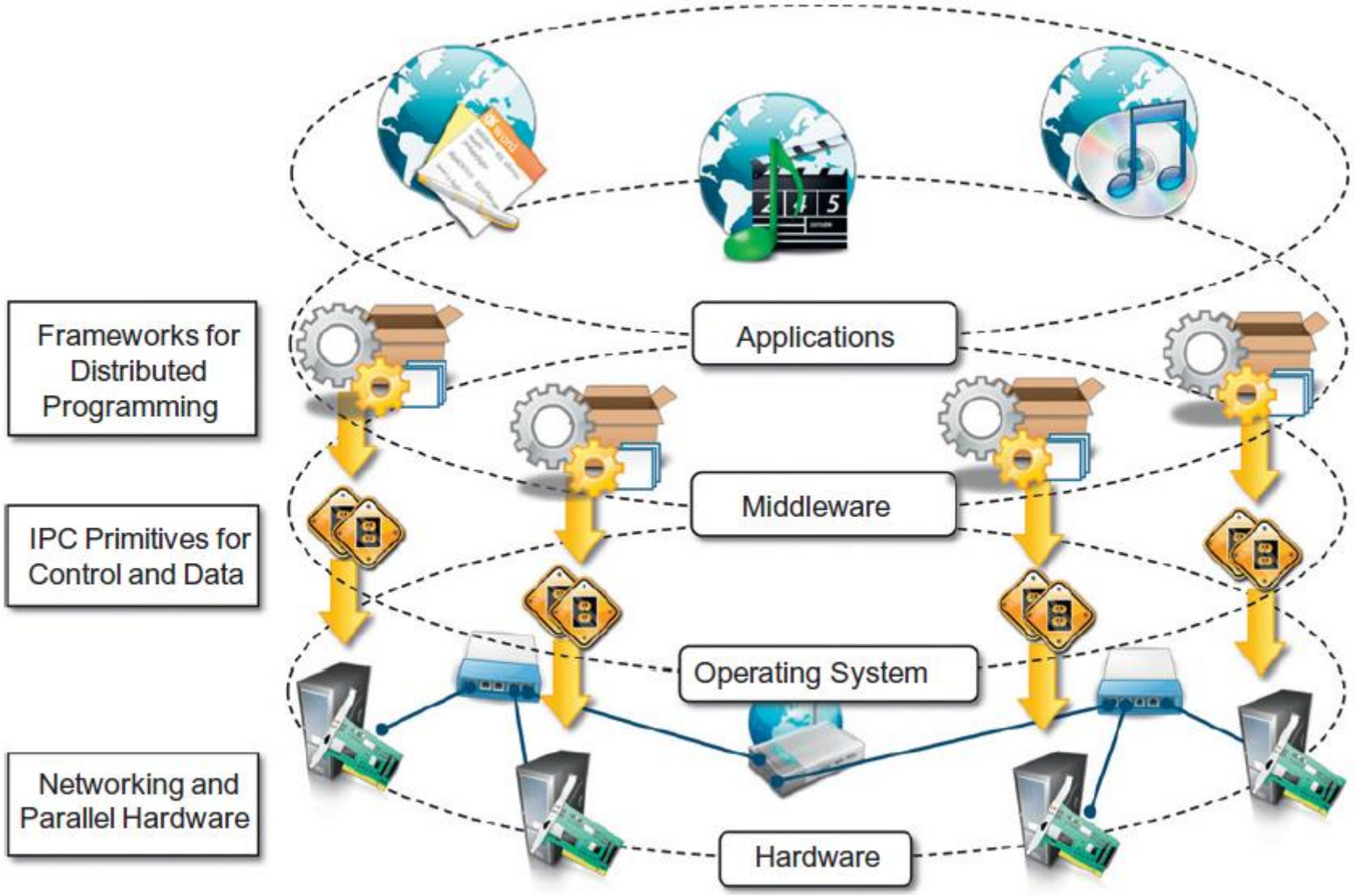


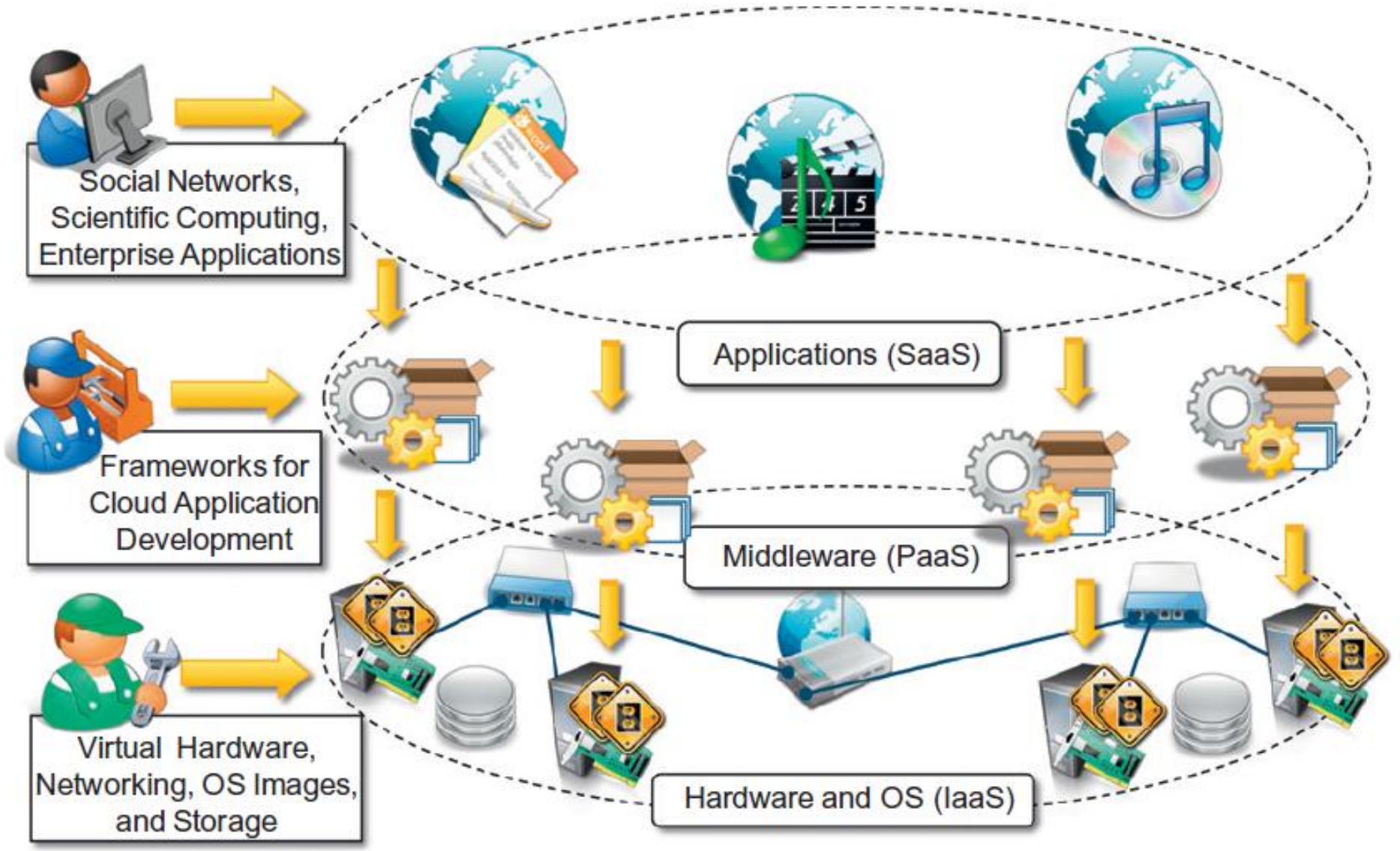
Square root



Logarithmic

Distributed Computing: Collection of independent computers that appears to its users as a single coherent system



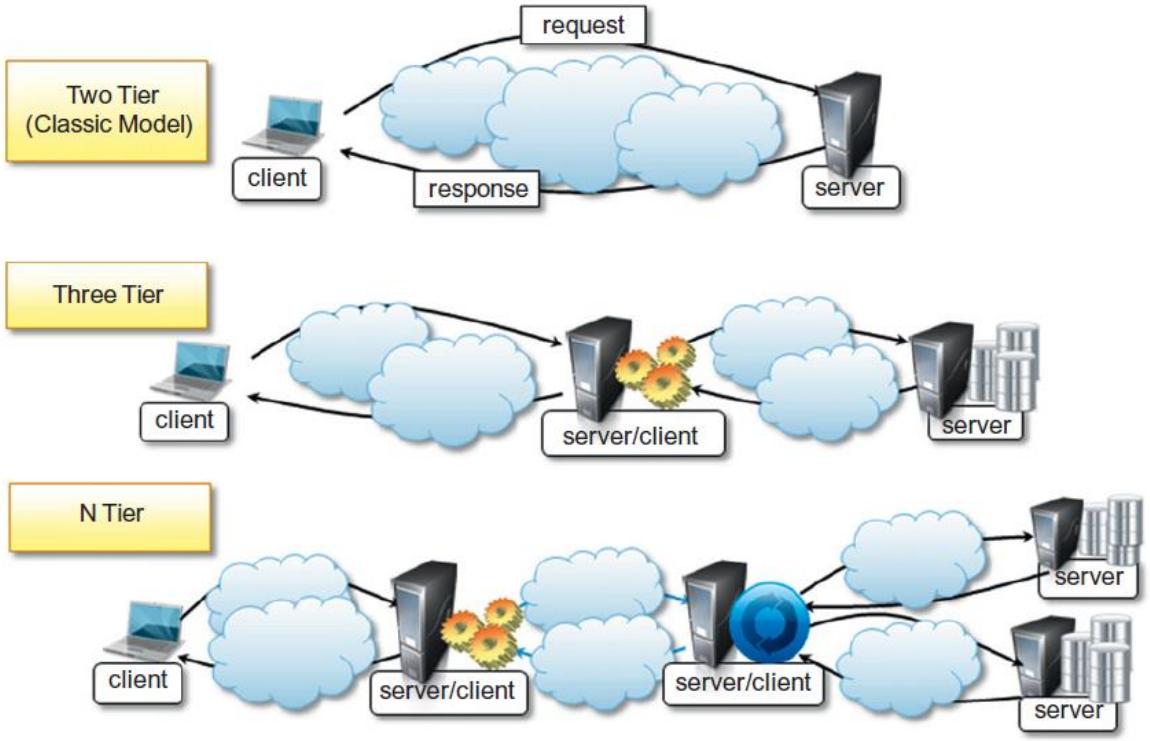

Social Networks, Scientific Computing, Enterprise Applications
Frameworks for Cloud Application Development
Virtual Hardware, Networking, OS Images, and Storage
Applications (SaaS)
Middleware (PaaS)
Hardware and OS (IaaS)

Software Architectural Styles

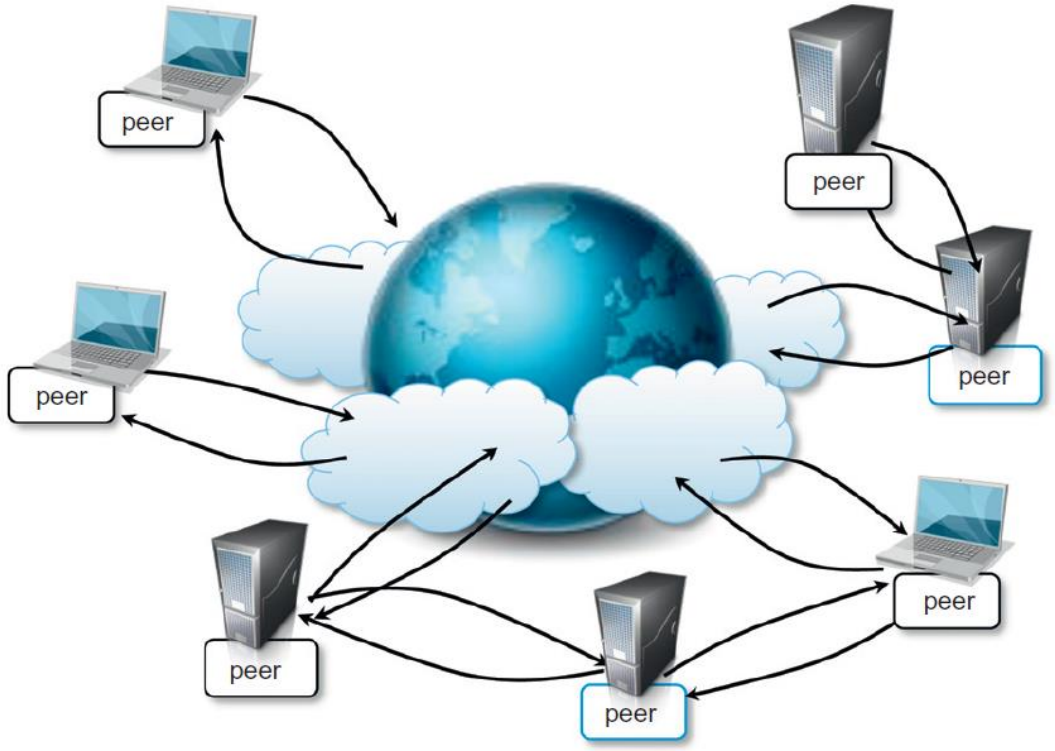
Category	Most Common Architectural Styles
Data-centered	Repository Blackboard
Data flow	Pipe and filter Batch sequential
Virtual machine	Rule-based system Interpreter
Call and return	Main program and subroutine call/top-down systems Object-oriented systems Layered systems
Independent components	Communicating processes Event systems

System Architectural Styles

Client - Server



Peer to Peer



Inter Process Communication (IPC)

Message: discrete amount of information that is passed from one entity to another

- Message passing: MPI, OpenMP
- Remote Procedure call (RPC): marshaling parameters and return values
- Distributed objects: state management and lifetime – CORBA, COM, .Net Remoting
- Distributed agents and active objects: objects with control thread
- Web Services: RPC over HTTP, Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) protocols

Messaging Models

- Point to point: direct addressing to identify the message receiver, no central infrastructure, initiated by sender
  - Direct
  - Queue based
- Publish and subscribe: topic or event, one to many, kafka
  - Push: publisher notifies all interested subscribers
  - Pull: subscriber poll for the messages of their interest
- Request-reply: each request has a reply, mostly point to point and not publish subscribe

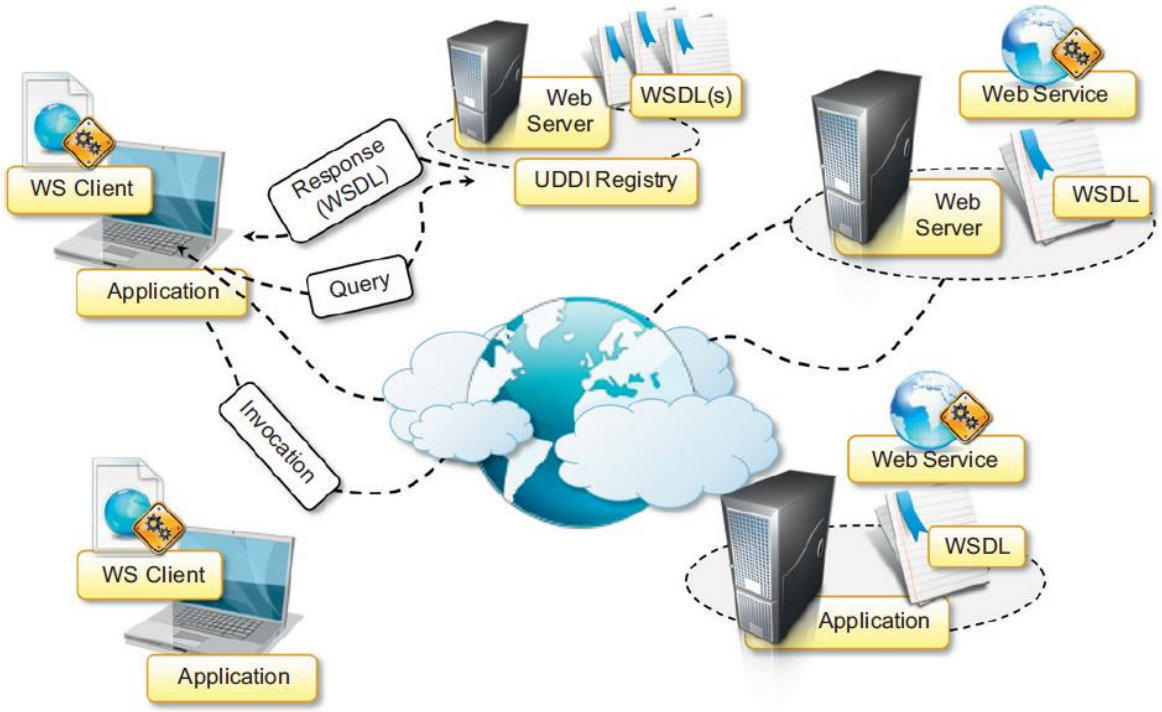
Technologies

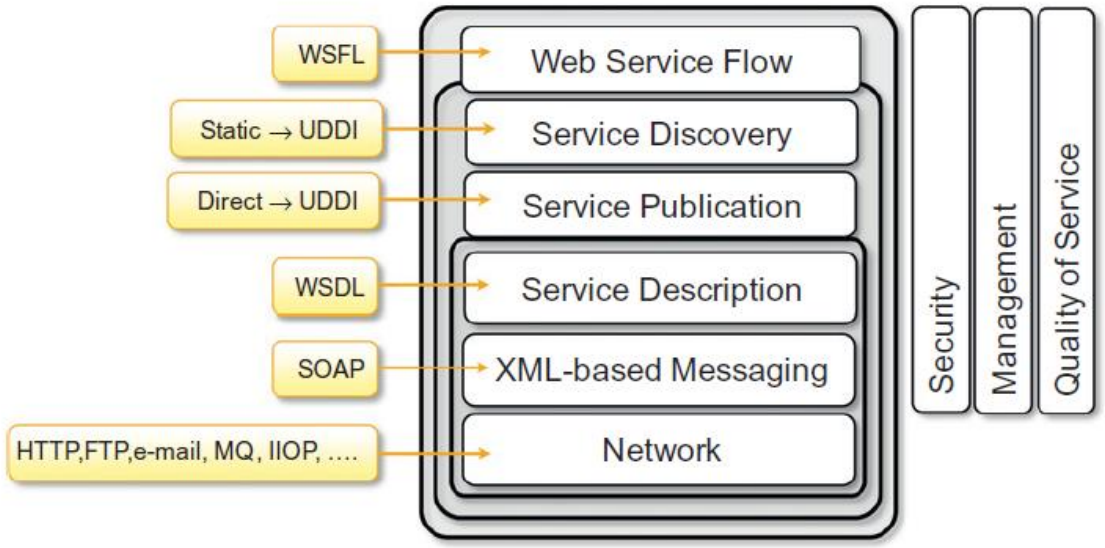
- RPC: gRPC
  - Server procedures
  - Register remote procedures with RPC Server
  - Client invoking remote procedures
- Distributed Object Frameworks: proxy skeleton
  - Objects distributed across heterogeneous network acting coherently
  - Server registry of active objects is published
  - Client accesses these objects using interfaces and invokes methods

Service Oriented Architecture (SOA)

- Service is a unit of functionality (Don Box)
  - Explicit boundaries
  - Autonomous
  - Expressed as schema and contracts, not class or interface definitions
  - Sematic compatibility is determined based on policy
- Guiding Principles
  - Standardized service contracts
  - Loose coupling
  - Abstraction
  - Reusability
  - Autonomy
  - Stateless
  - Discoverable
  - Composable

Web Services





Abstraction and Virtualization

SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

Abstraction

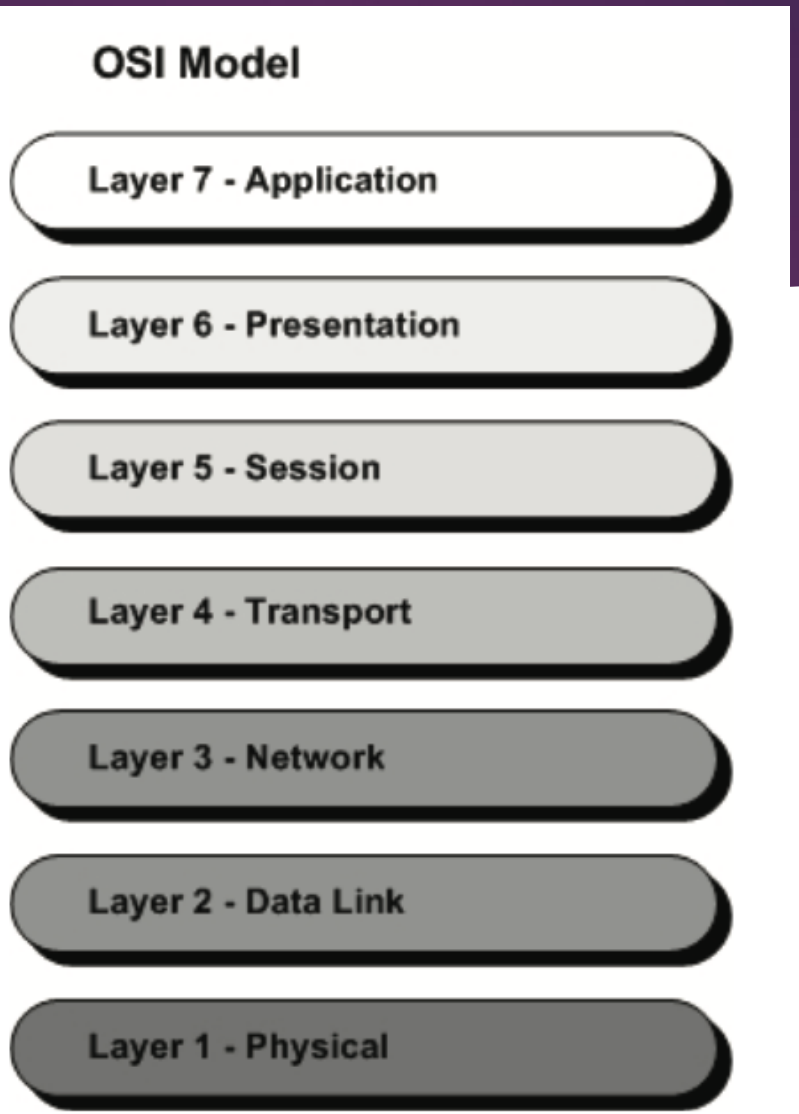
- Connecting multiple systems together leveraging their interfaces, not worrying about internal details
- Uber / Ola as an example for transportation service
- OSI – 7 layers of networking
- ODBC – abstracting databases
- OpenGL – images
- Internet is an abstraction layer to cloud computing
- Translation from abstraction layer to underlying physical layer

Open Systems Interconnection

Vendor Interconnection

Layer Interconnection

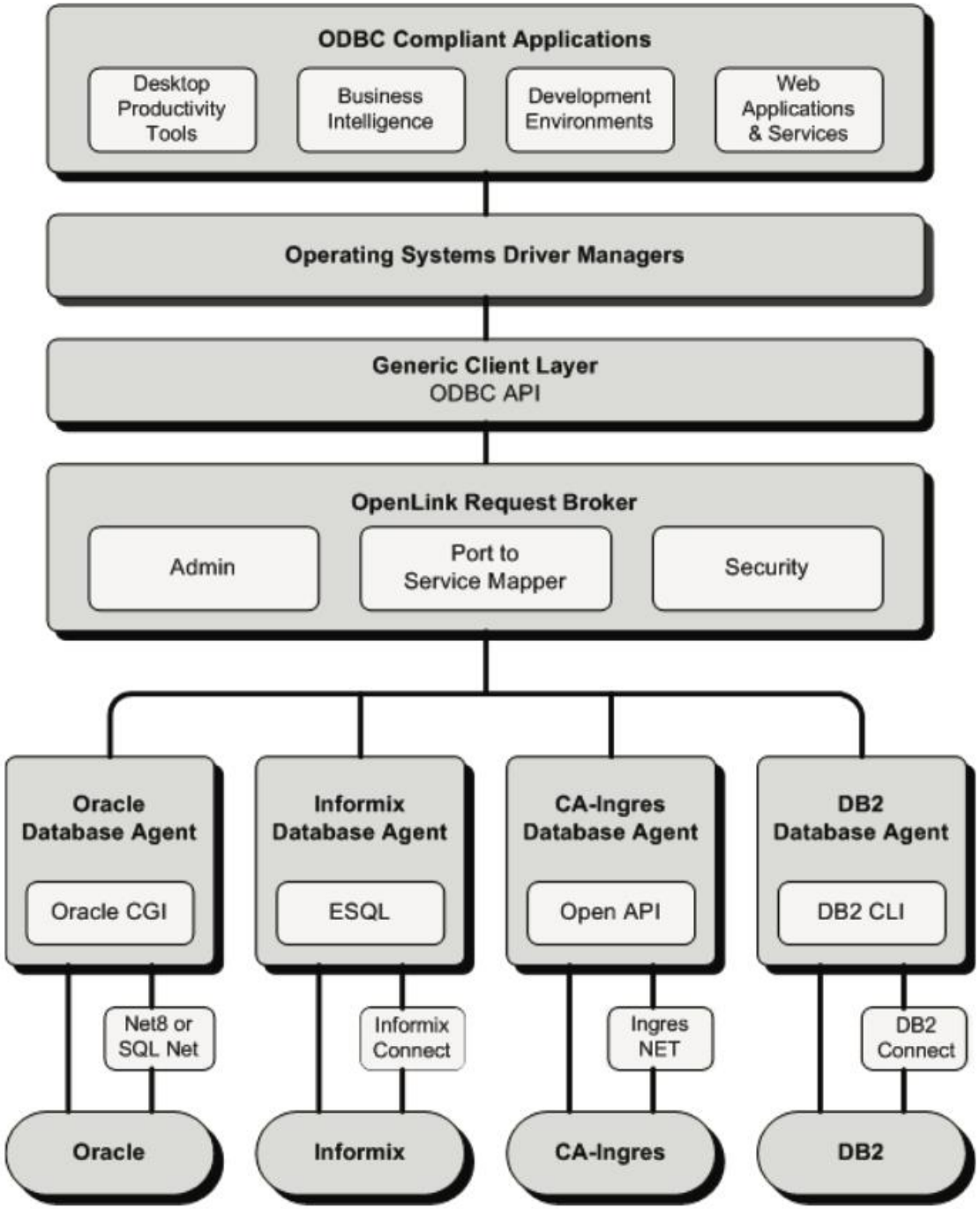
Independent development at each layer



Open DataBase Connection

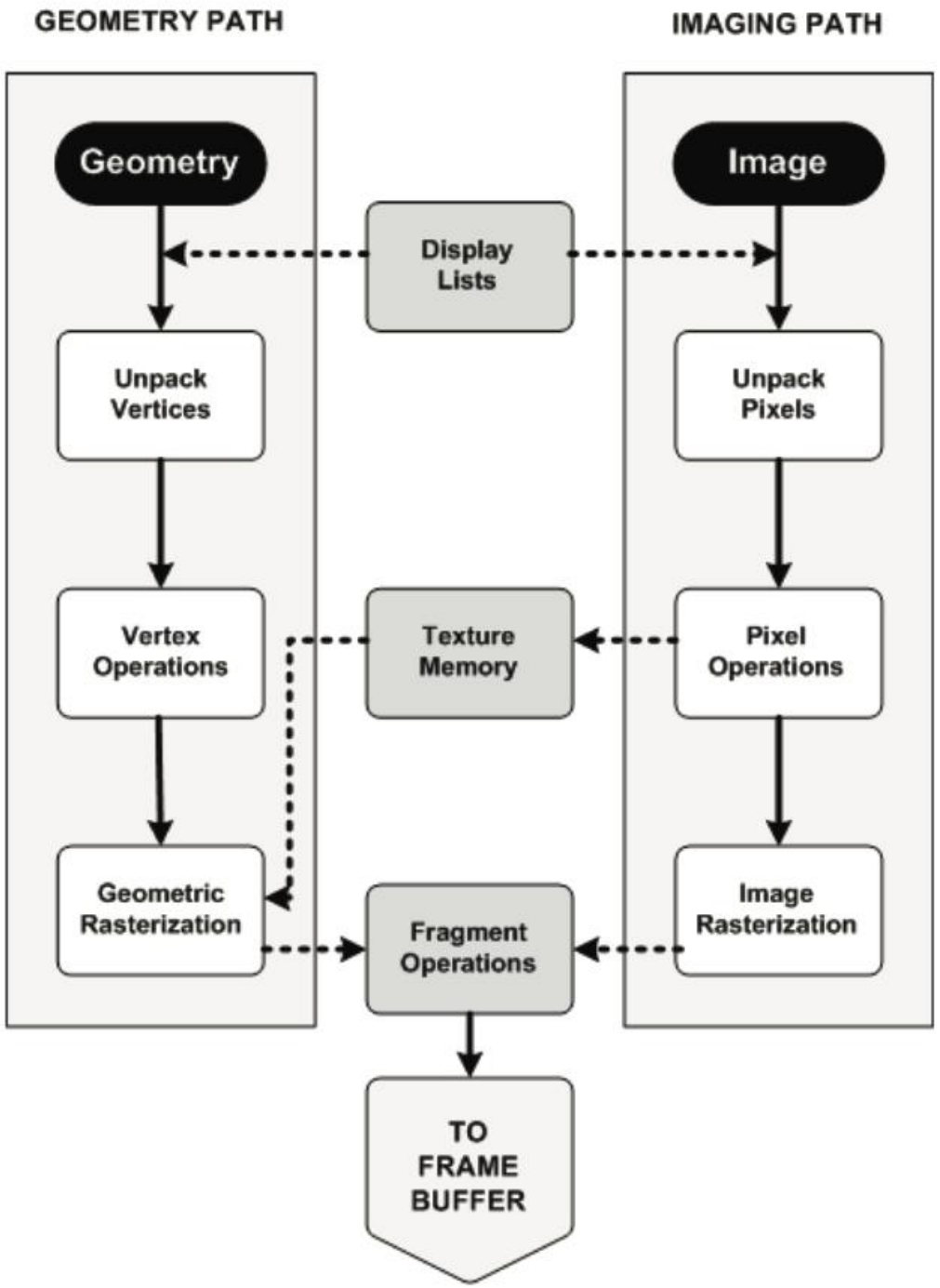
Config change

ODBC driver



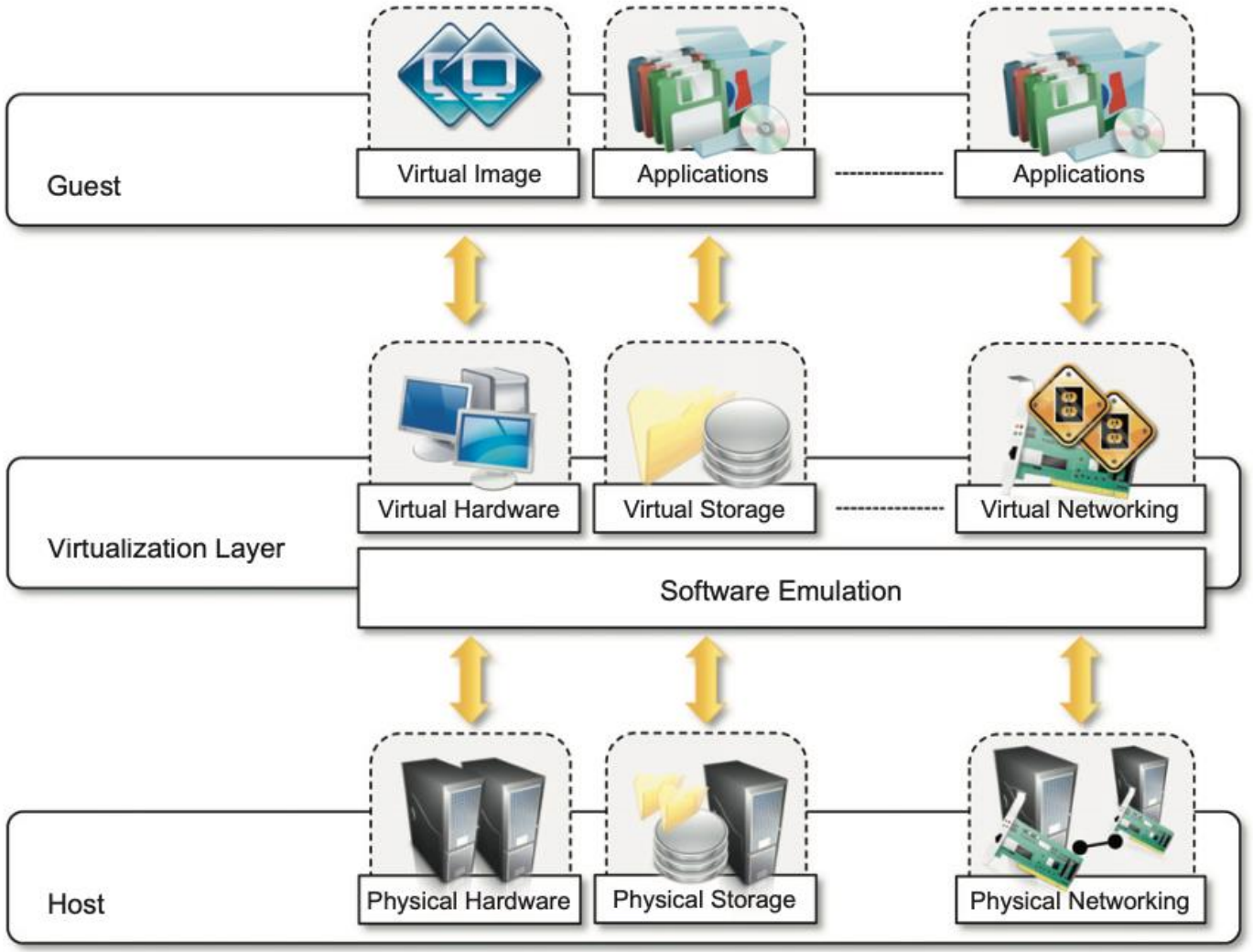
Open Graphics Library

Application development and Display adapter



Virtualization

- Abstraction of hardware, runtime environments, storage and networking
- Began in 1960s but widely adopted since 2000s
- Physical Host – virtual guest, 1 physical server hosting tens of virtual guests
- Virtual machine (VM) instances operate in complete isolation of each other
- Customized software stack on demand in each VM instance


Guest
Virtual Image
Applications
Applications
Virtualization Layer
Virtual Hardware
Virtual Storage
Virtual Networking
Software Emulation
Host
Physical Hardware
Physical Storage
Physical Networking

Factors

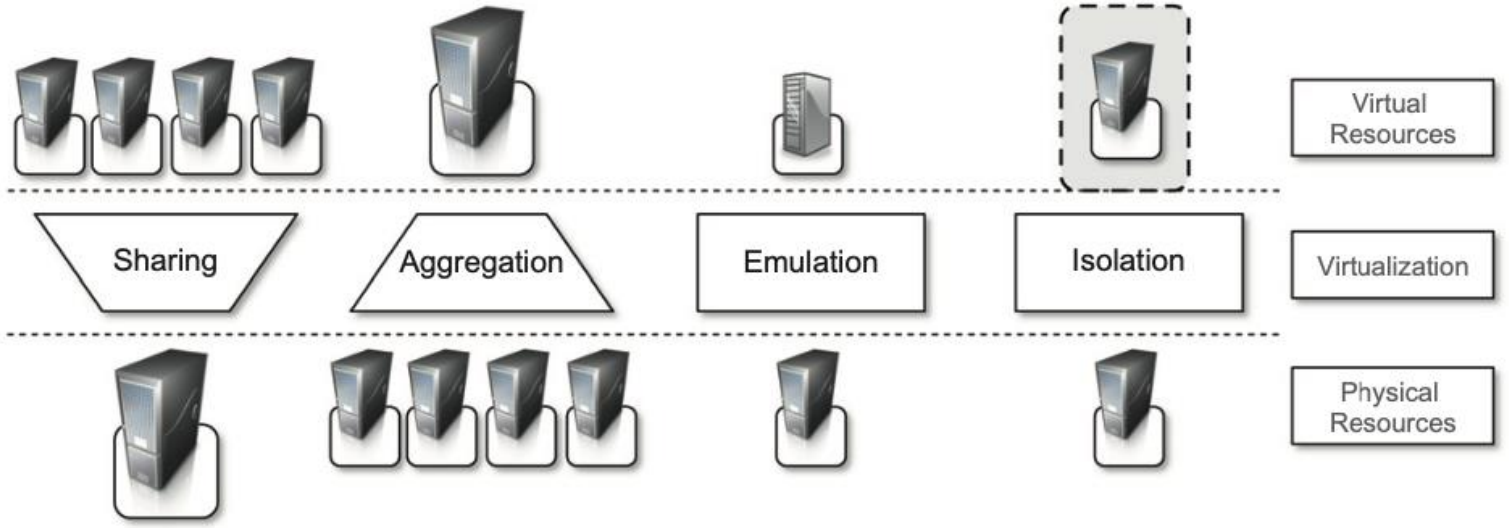
- Increased computing capacity: server can host tens of VMs
- Underutilized resources: using office desktops during night without affecting day work
- Lack of space – server consolidation
- Energy efficiency / Green initiatives
- Increased administrative costs
- 1995 – Sun Java, 2002 - .Net

Characteristics

- hardware virtualization
- physical machine is the host
- virtualization layer / virtual machine manager / hypervisor – software
- virtual machines / guests interact with virtualization layer

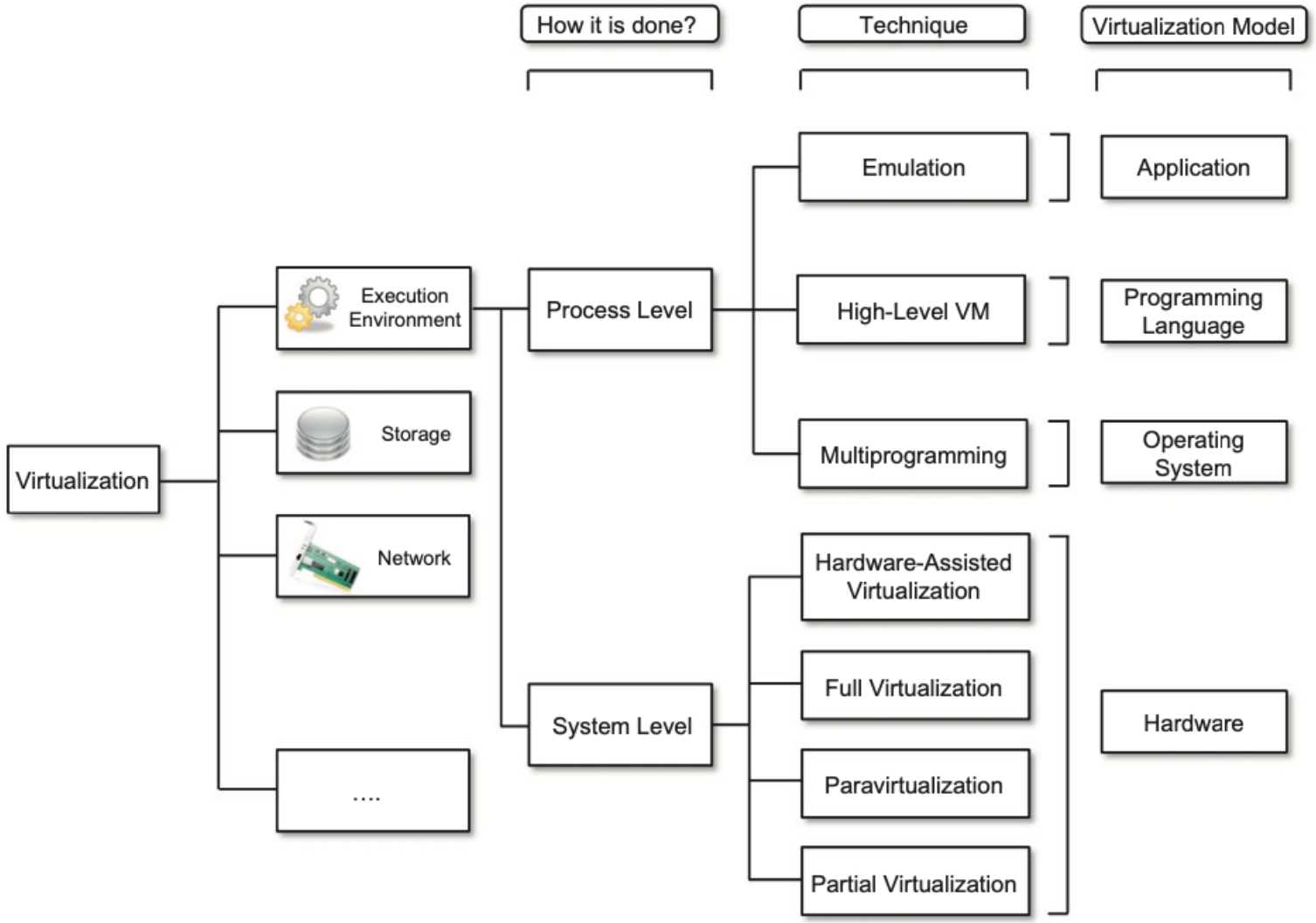
Advantages

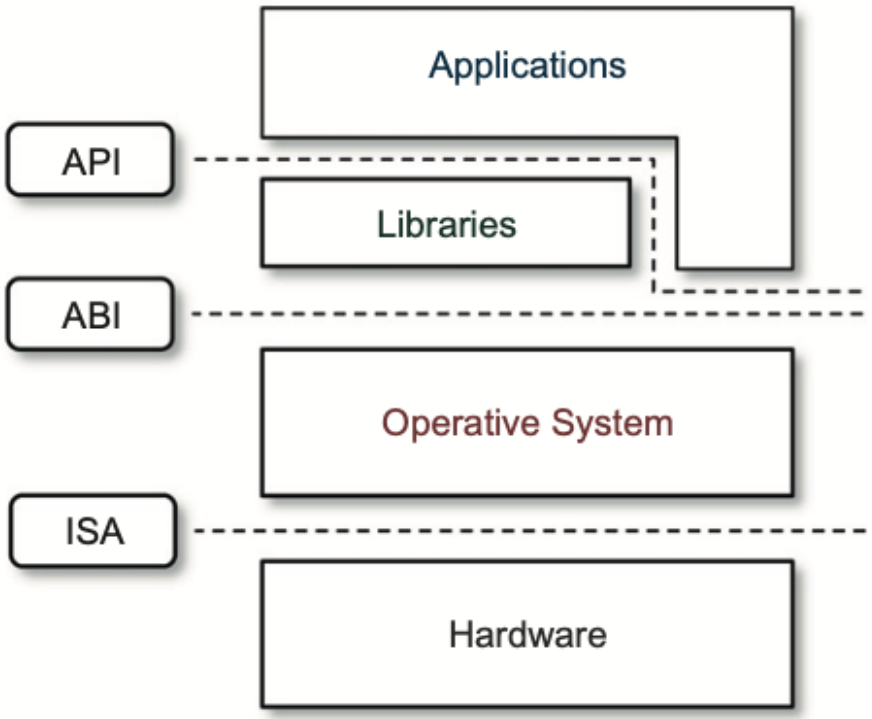
- Increased security – translation, control and filter guest activity, hide things on host from guest(s), limited access, separate FS
- Managed execution

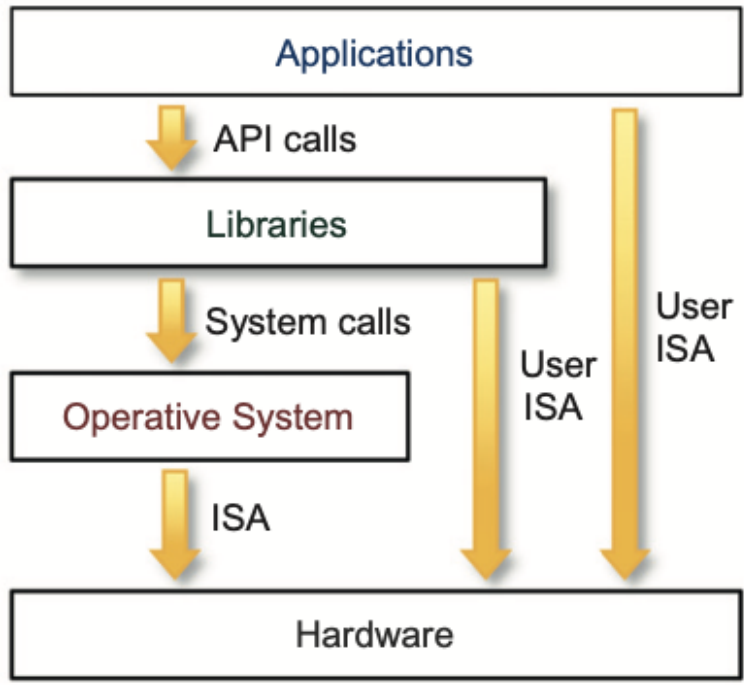


Advantages (contd)

- Performance tuning
- VM snapshot, migration
- Portability : JVM jar, .Net assembly, VMWare VMDK – images; don’t need all applications and services but just VM manager


How it is done?
Technique
Virtualization Model
Virtualization
Execution Environment
Storage
Network
....
Process Level
System Level
Emulation
High-Level VM
Multiprogramming
Hardware-Assisted Virtualization
Full Virtualization
Paravirtualization
Partial Virtualization
Application
Programming Language
Operating System
Hardware


Applications
API
Libraries
ABI
Operative System
ISA
Hardware

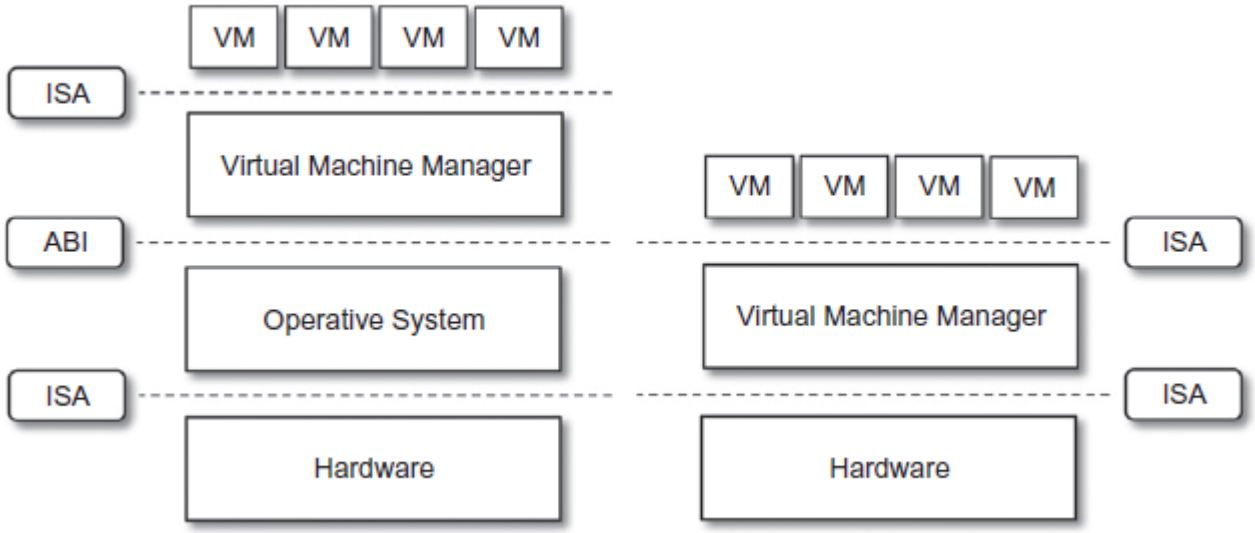

Applications
API calls
Libraries
System calls
Operative System
ISA
User ISA
User ISA
Hardware

Levels of virtualization

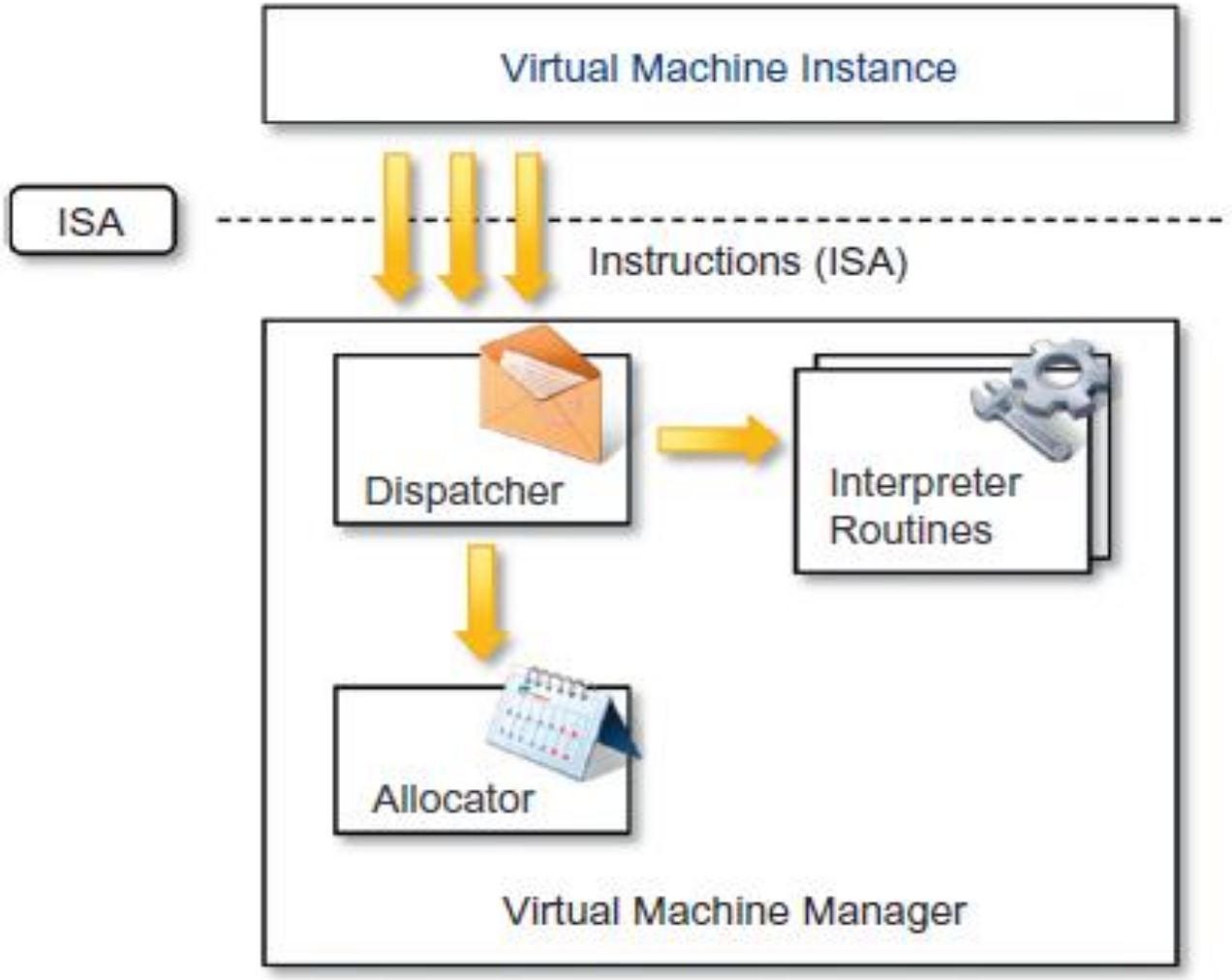
- Hardware level
- Operating system level
- Programming language level
- Application level
- Storage
- Network
- Desktop
- Application server

Hardware level virtualization

- Hypervisor
  - Type I : on bare hardware, like OS, interact directly with ISA, native
  - Type II : require support of OS, interact with ABI, hosted



Hypervisor Reference Architecture



Virtual Machine Manager Criteria

- **Equivalence**: should be same as that of phyisical host
- **Resource control:** should be complete control of virtualized resource
- **Efficiency**. A statistically dominant fraction of the machine instructions should be executed without intervention from the VMM.

Classification of instructions set

- Sensitive instructions are subset of privileged instructions
- Recursive virtualization: A VMM without any timing dependencies can be constructed for it
- In hybrid VMM user sensitive instructions are subset of privileged instructions

Hardware Virtualization Techniques

- Hardware assisted : architectural support, Intel x86-64 VT-x, AMD V (2006), less overhead
- Full virtualization: running complete OS in VM like physical machine
- Paravirtualization: thin virtualization layer, guest needs to be modified to use VMM interface, need OS source, Linux on Xen
- Partial virtualization: partial execution of guest, Address space virtualization

Levels of Virtualization

- OS Level: no VMM, within OS, evolution of Unix chroot, less overhead, solaris zones, containers (namespace)
- PL level: JVM, .Net assembly
- Application level: not installed on runtime hardware but run as if they were
  - Interpretation: every instruction is emulated, low startup but high overhead
  - Binary translation: block converted to native instructions, cached and reused

  Wine: Unix like on Windows

  WABI: Win16 on Solaris

  CrossOver: Windows on Mac OS X

  VMWare ThinApp

Other than execution virtualization

- Storage Virtualization: decouple physical from logical, Storage Area Networks (SAN), NAS, enabled SDS
- Network Virtualization: VLAN, Network Address Translation (NAT) enabled SDN
- Desktop Virtualization: making the same desktop environment accessible from everywhere, VDI
- Application server virtualization: achieved using load balancers

Enablers for Cloud Computing

- Hardware level – IaaS
- PL level – PaaS
- Migrate VMs to free up servers, server consolidation
- Storage services
- Network services
- VDI: your entire desktop can come from cloud

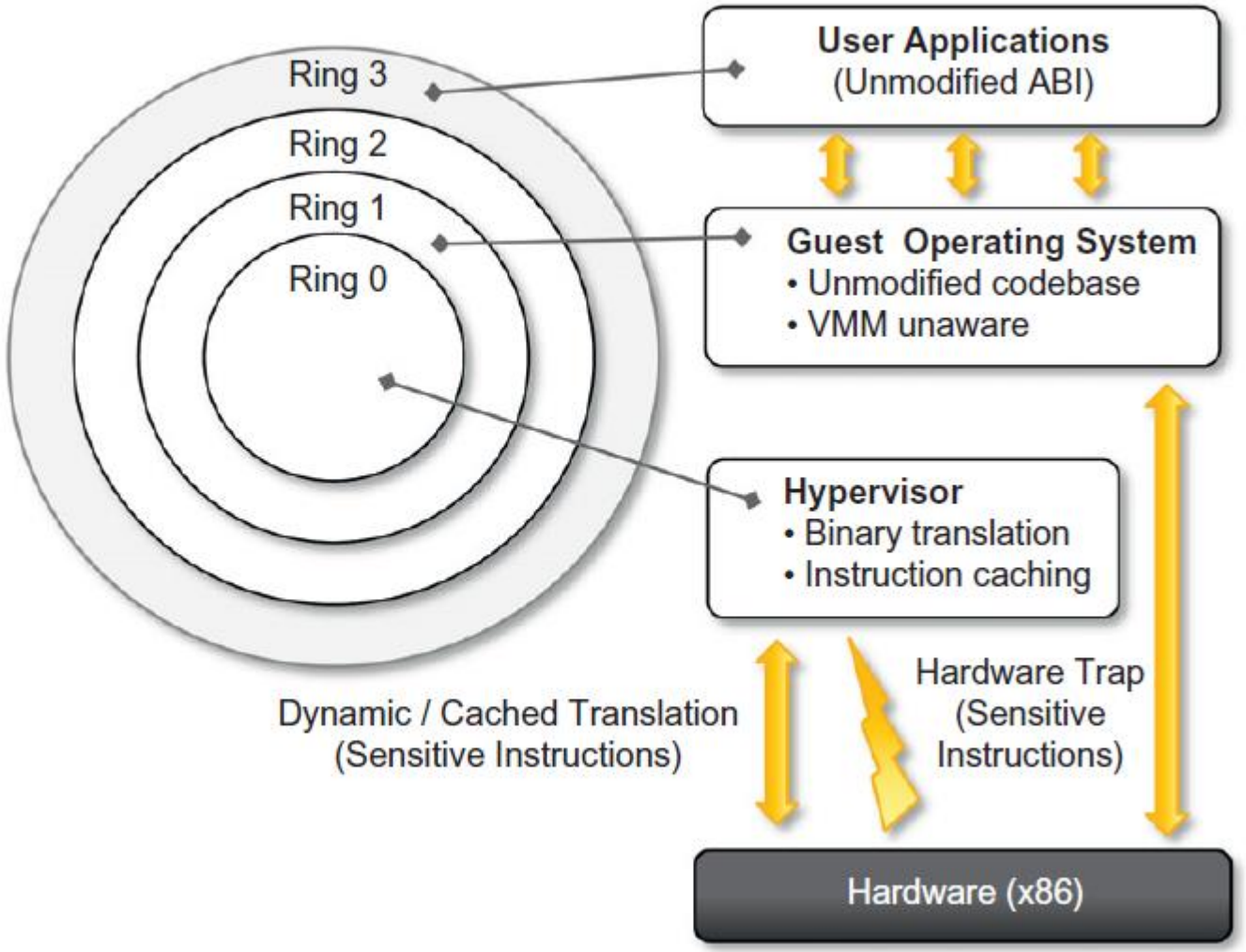
Advantages & Disadvantages

- Sandbox, Prevent any harmful operation cross the boundary
- Portable across physical systems, self contained without any dependencies, represented by one or two files
- Low maintenance

- Performance degradation
- Inefficiency and degraded user experience: inaccessible features
- Security holes, threats, phishing: extract sensitive information from guest, BluePil, SubVirt

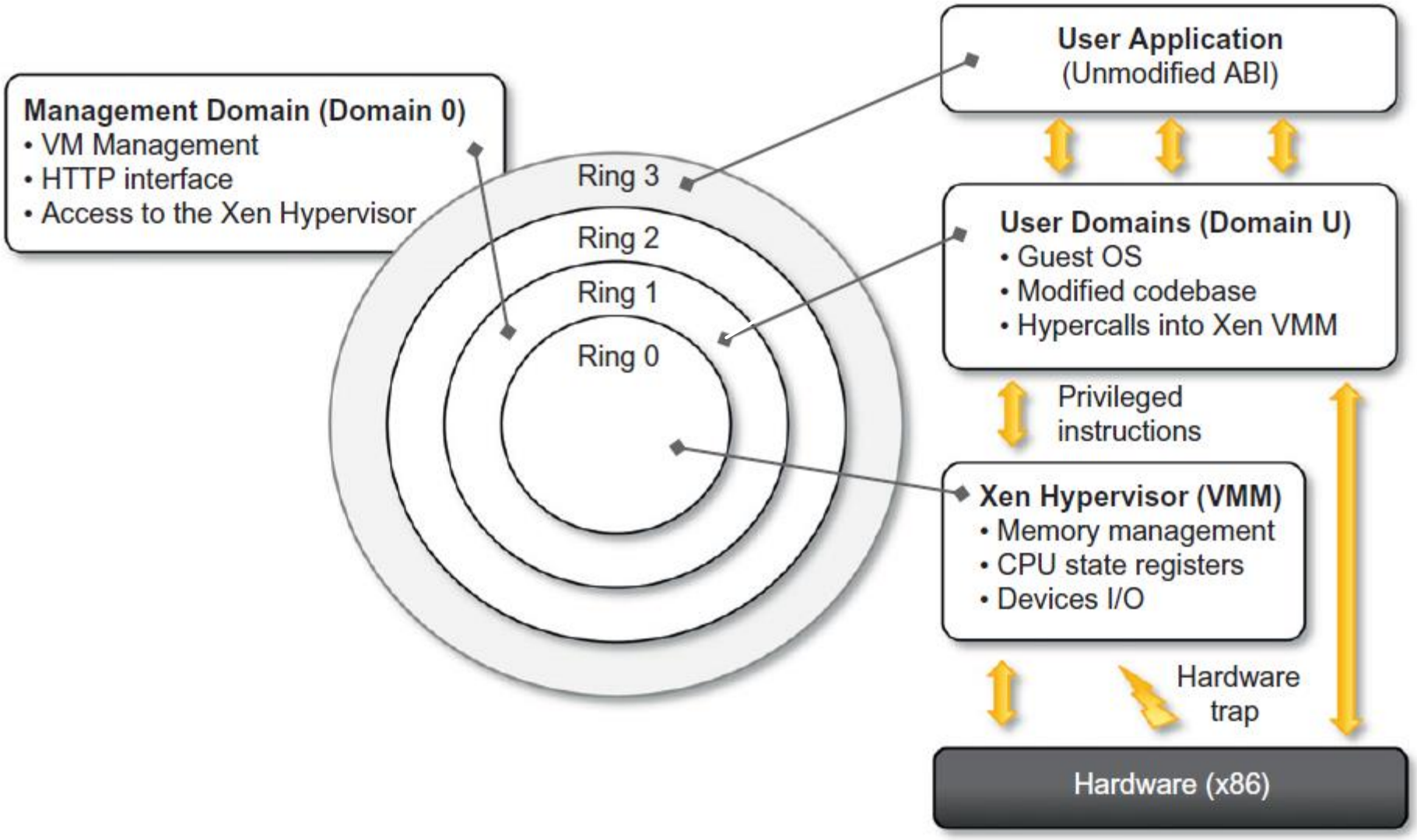
VMWare: full virtualization

- Hardware replicated, guest not modified and not aware of abstraction layer in between, desktop type II, server type I, direct execution for non sensitive instructions, binary translation for sensitive instructions, ecosystem
- X86 architecture does not meet theorem I, dynamic binary translation, trap is generated for privileged instructions and translation performed, caching of translation to improve performance
- CPU, memory and I/O devices (keyboard, mouse, USB, network controllers) virtualization
- Translation Look-aside Buffer (TLB) to improve MMU performance

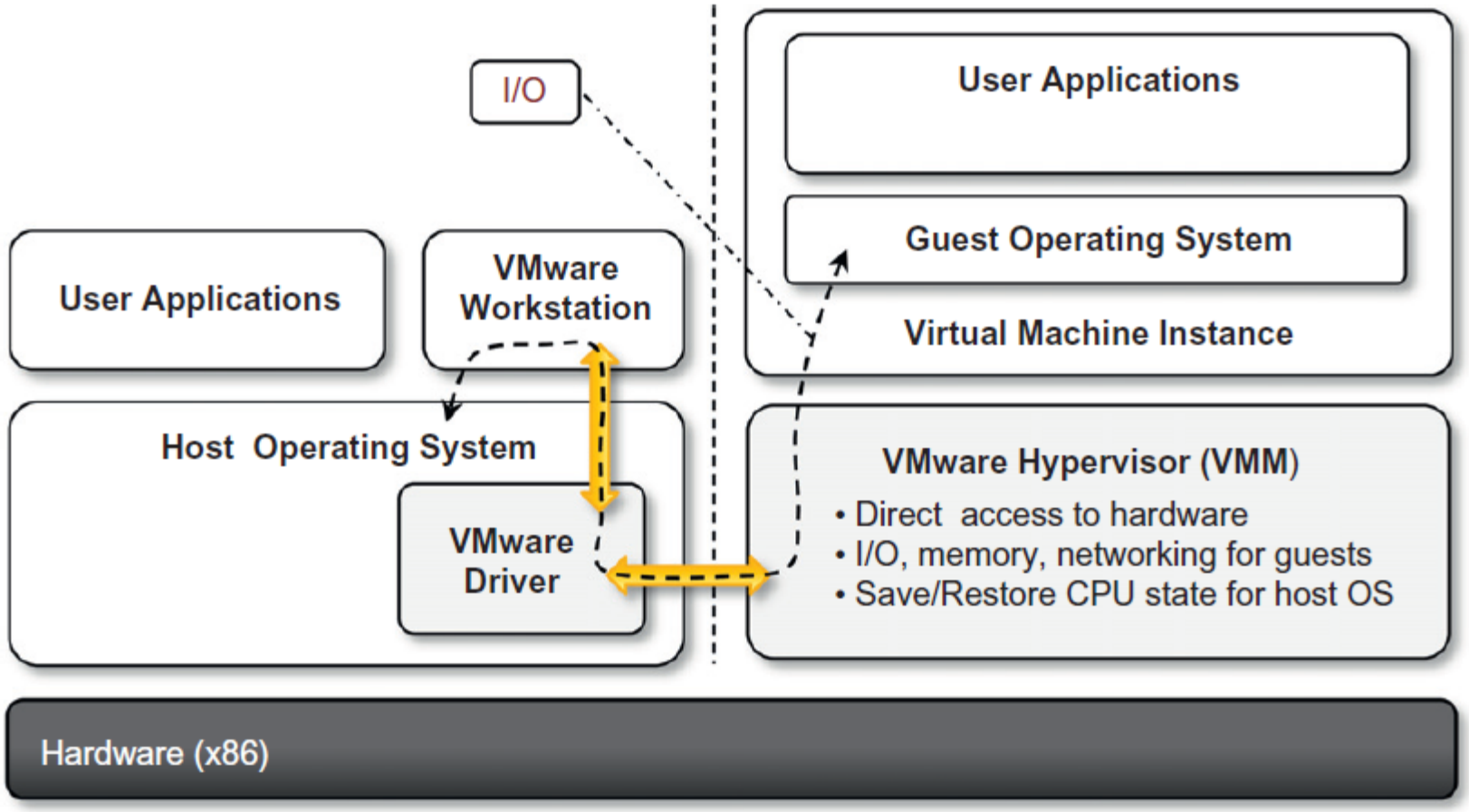

Ring 3
Ring 2
Ring 1
Ring 0
User Applications
(Unmodified ABI)
Guest Operating System
• Unmodified codebase
• VMM unaware
Hypervisor
• Binary translation
• Instruction caching
Dynamic / Cached Translation
(Sensitive Instructions)
Hardware Trap
(Sensitive
Instructions)
Hardware (x86)

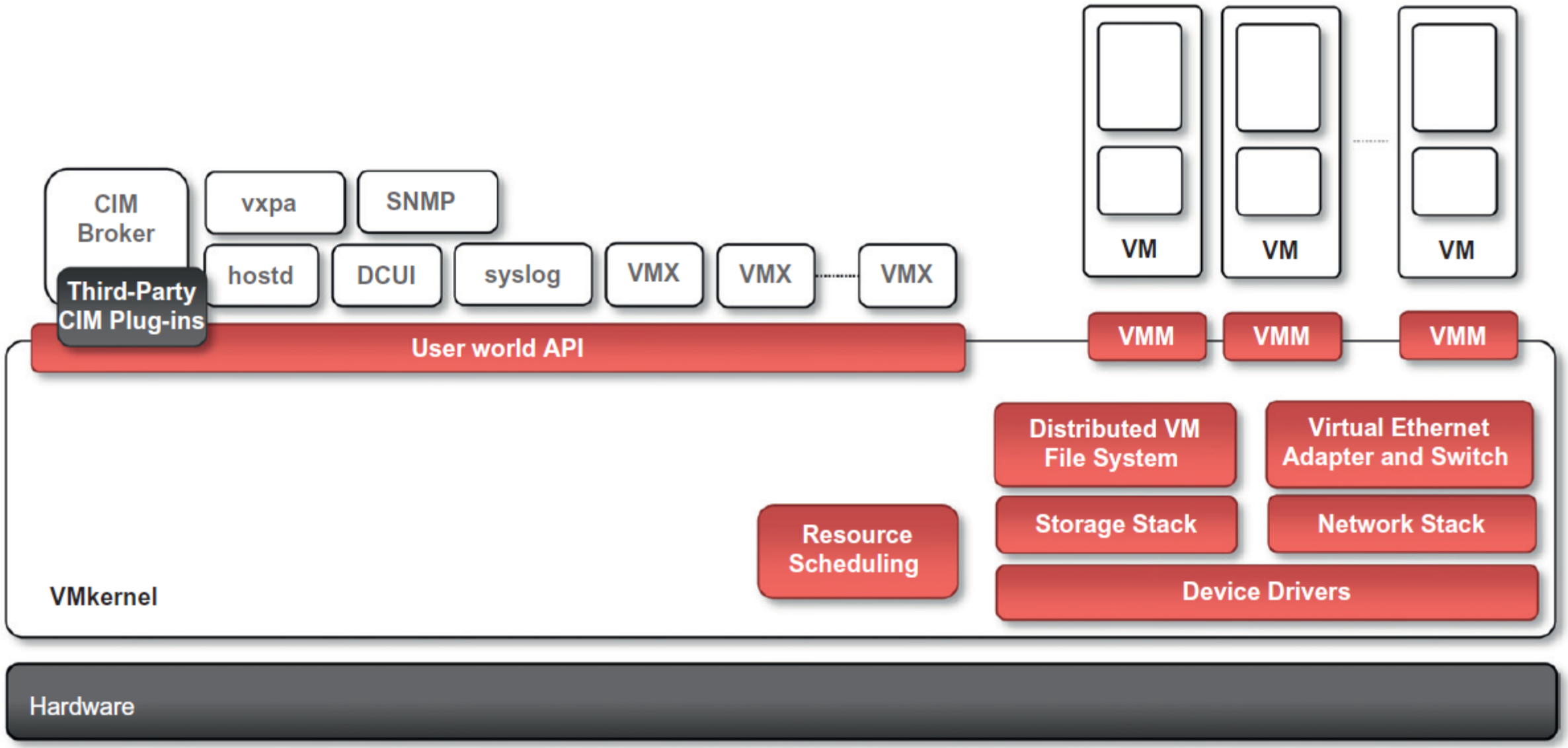
Xen: paravirtualization

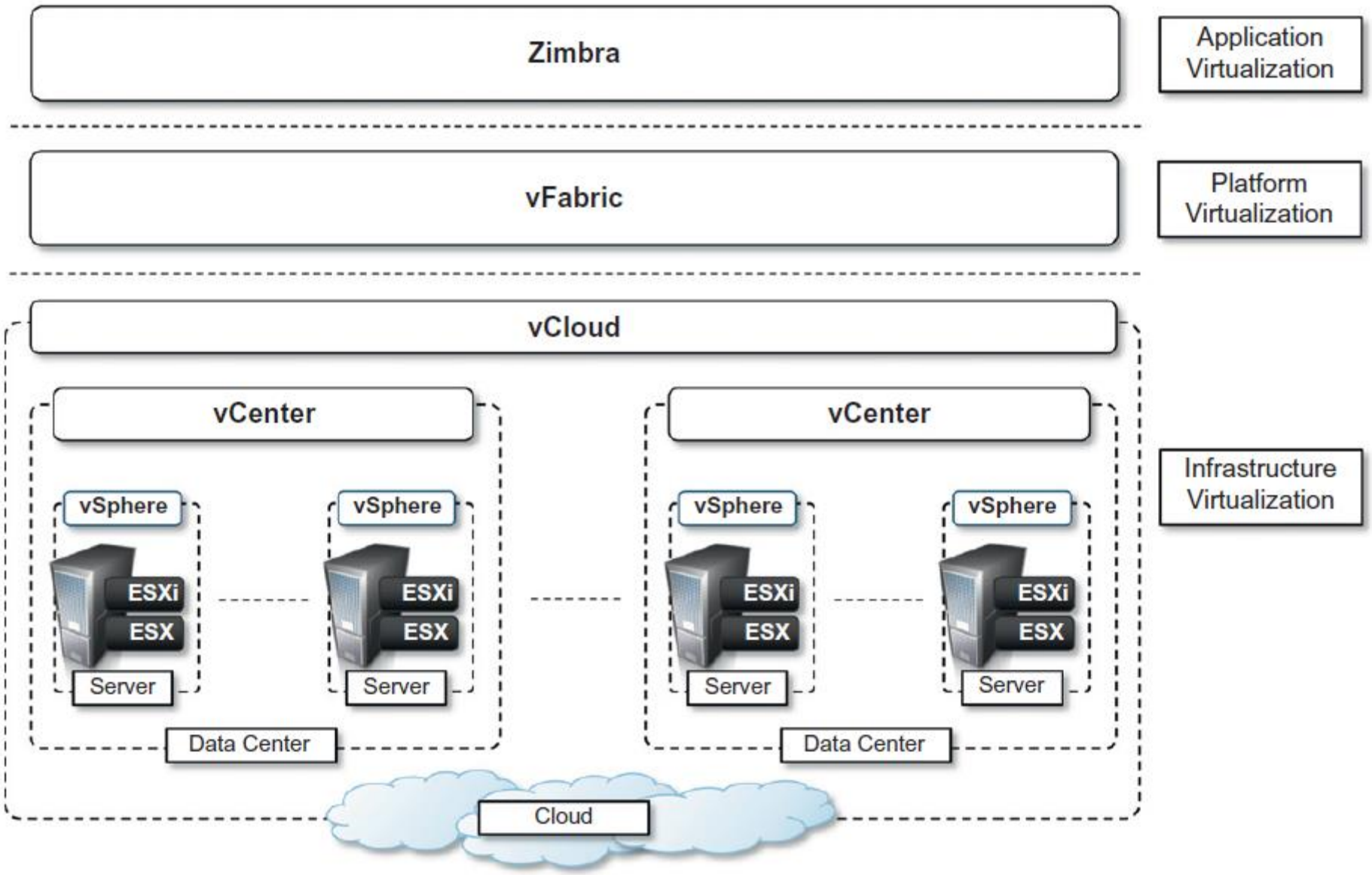
- Research by Cambridge University, UK
- Open source community
- Citix offers commercial offering of XenSource
- Desktop or server virtualization
- Xen Cloud Platform XCP
- Full virtualization with hardware assistance
- Xen Hypervisor, domains, hypercalls

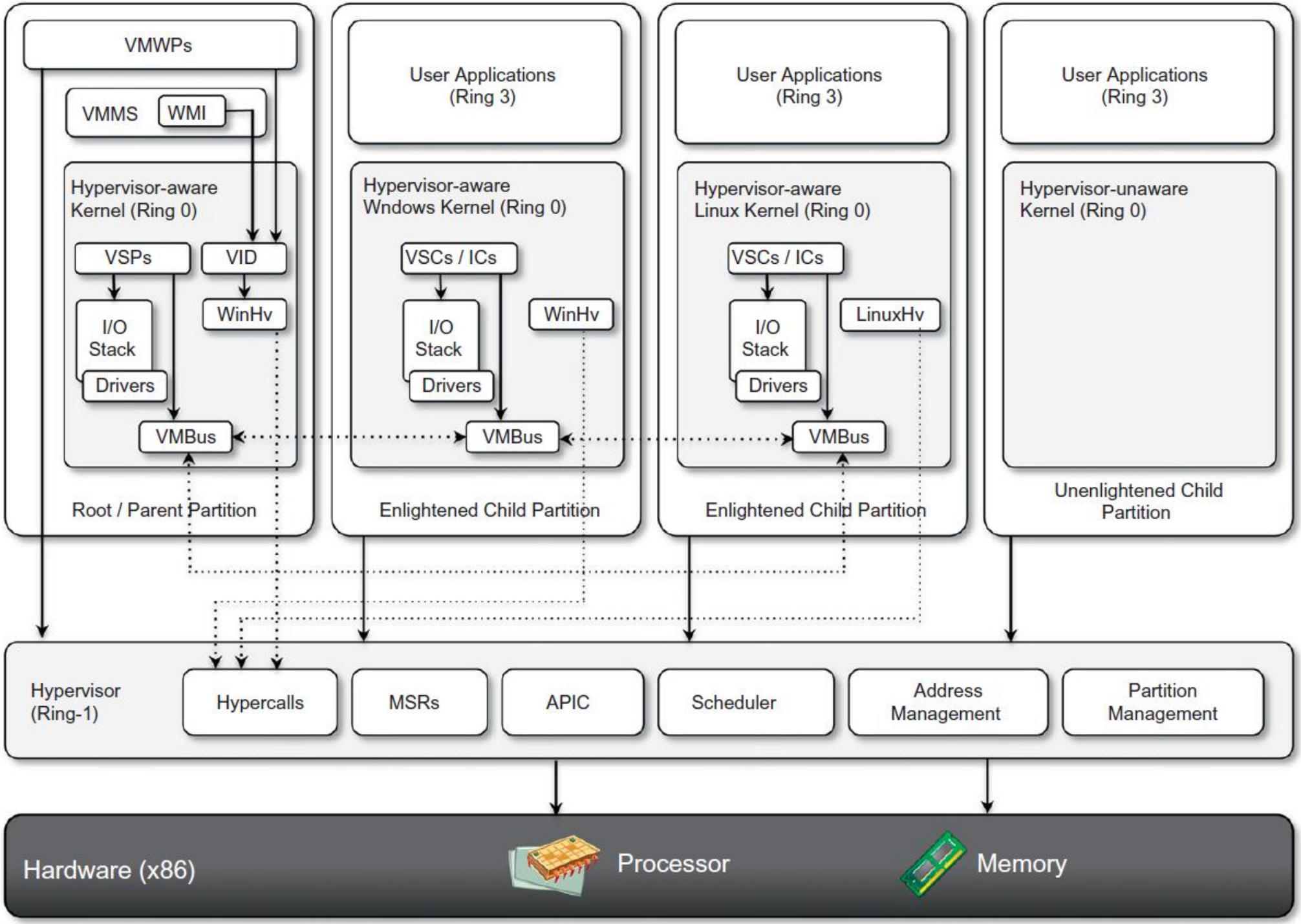

Management Domain (Domain 0)
• VM Management
• HTTP interface
• Access to the Xen Hypervisor
Ring 3
Ring 2
Ring 1
Ring 0
User Application
(Unmodified ABI)
User Domains (Domain U)
• Guest OS
• Modified codebase
• Hypercalls into Xen VMM
Privileged
instructions
Xen Hypervisor (VMM)
• Memory management
• CPU state registers
• Devices I/O
Hardware
trap
Hardware (x86)

End user VMWare Workstation Architecture




CIM Broker
Third-Party CIM Plug-ins
vxpa
SNMP
hostd
DCUI
syslog
VMX
VMX
VMX
User world API
VM
VM
VM
VMM
VMM
VMM
Distributed VM File System
Virtual Ethernet Adapter and Switch
Storage Stack
Network Stack
Resource Scheduling
Device Drivers
VMkernel
Hardware


Zimbra
Application Virtualization
vFabric
Platform Virtualization
vCloud
vCenter
vCenter
vSphere
vSphere
vSphere
vSphere
ESXi
ESX
ESXi
ESX
ESXi
ESX
ESXi
ESX
Server
Server
Server
Server
Data Center
Data Center
Cloud
Infrastructure Virtualization


VMWPs
VMMS
WMI
Hypervisor-aware
Kernel (Ring 0)
VSPs
VID
I/O
Stack
Drivers
WinHv
VMBus
Root / Parent Partition
User Applications
(Ring 3)
Hypervisor-aware
Wndows Kernel (Ring 0)
VSCs / ICs
I/O
Stack
Drivers
WinHv
VMBus
Enlightened Child Partition
User Applications
(Ring 3)
Hypervisor-aware
Linux Kernel (Ring 0)
VSCs / ICs
I/O
Stack
Drivers
LinuxHv
VMBus
Enlightened Child Partition
User Applications
(Ring 3)
Hypervisor-unaware
Kernel (Ring 0)
Unenlightened Child
Partition
Hypervisor
(Ring-1)
Hypercalls
MSRs
APIC
Scheduler
Address
Management
Partition
Management
Hardware (x86)
Processor
Memory

Releases

- Hyper-Viscurrently shipped as a component of Windows Server 2008 R2 that installs the hypervisor as a role within the server
- Windows Server Core to increase performance with smaller footprint, reduced maintenance, reduced management, use of “removed” rich features by remote management through WMI (e.g. powershell)
- SystemCenter Virtual MachineManager(SCVMM)2008
  - Management portal
  - V2V, P2V conversions
  - Deep powershell integration

Assignment

- Install VMWare ESXi server
- Create Ubuntu VM
- Connect to Ubuntu VM

Cloud Computing Architecture

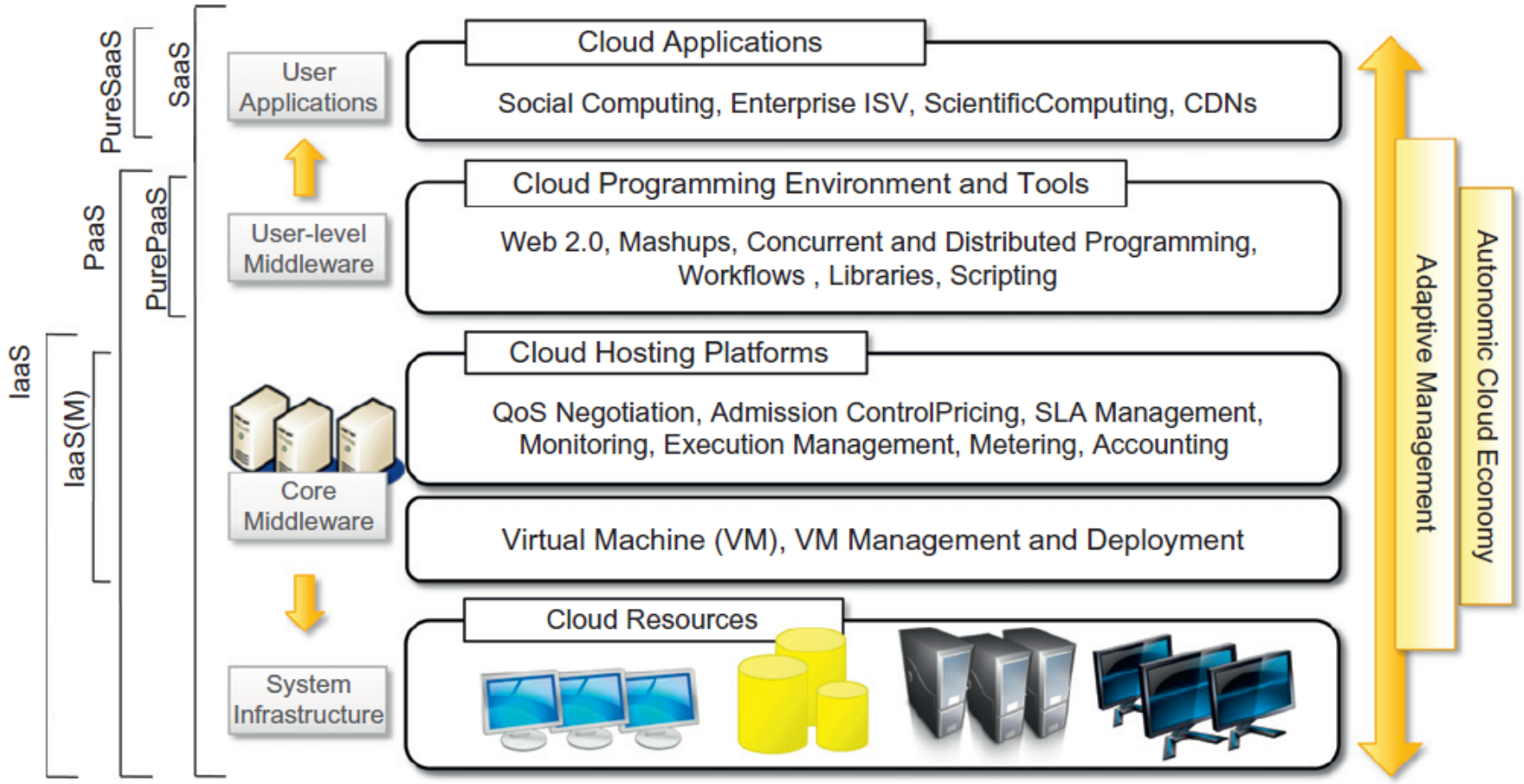
SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

Cloud Computing Architecture

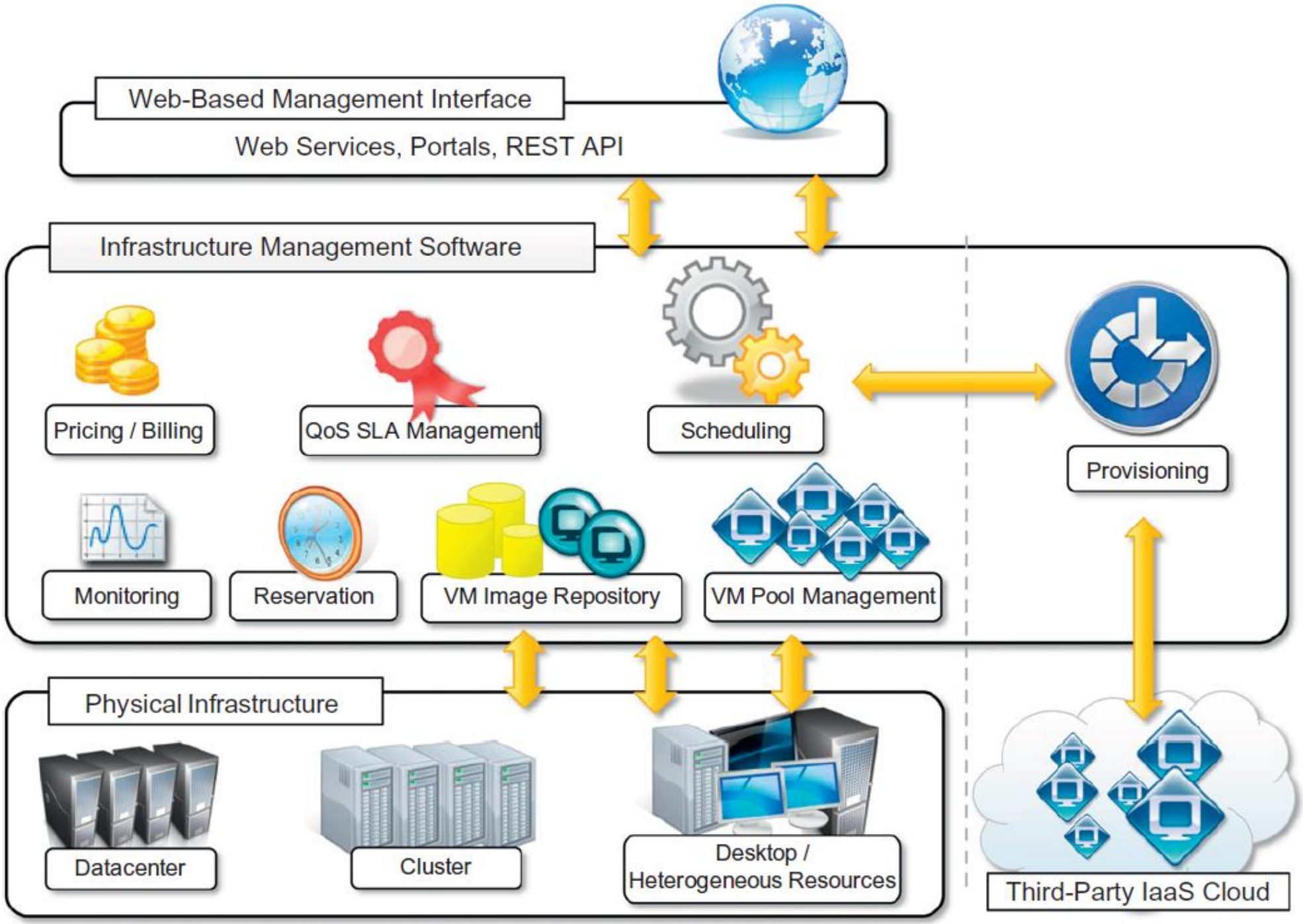
- Distributed infrastructure
- Data center(s), cluster(s), composed of desktops, workstations, servers
- Infrastructure can be owned by provider or rented from third party
- Typically virtualized for workload isolation and cost optimization
- Different layers stacked on each other
- Utility oriented, internet centric, IT services on demand, covering the entire computing stack from hardware infrastructure to development platforms and distributed applications

Reference Model



IaaS

- Infrastructure or hardware as a service
- Most popular and developed market segment of cloud computing
- from single servers to entire infra- structures, including network devices, load balancers, and database and Web servers
- Main technology used: hardware virtualization
- Atomic component of VM priced by memory, #processors, and disk size
- workload partitioning, application isolation, sandboxing, and hardware tuning
- LAMP: Linux Apache MySql and PHP


Web-Based Management Interface
Web Services, Portals, REST API
Infrastructure Management Software
Pricing / Billing
QoS SLA Management
Scheduling
Provisioning
Monitoring
Reservation
VM Image Repository
VM Pool Management
Physical Infrastructure
Datacenter
Cluster
Desktop / Heterogeneous Resources
Third-Party IaaS Cloud

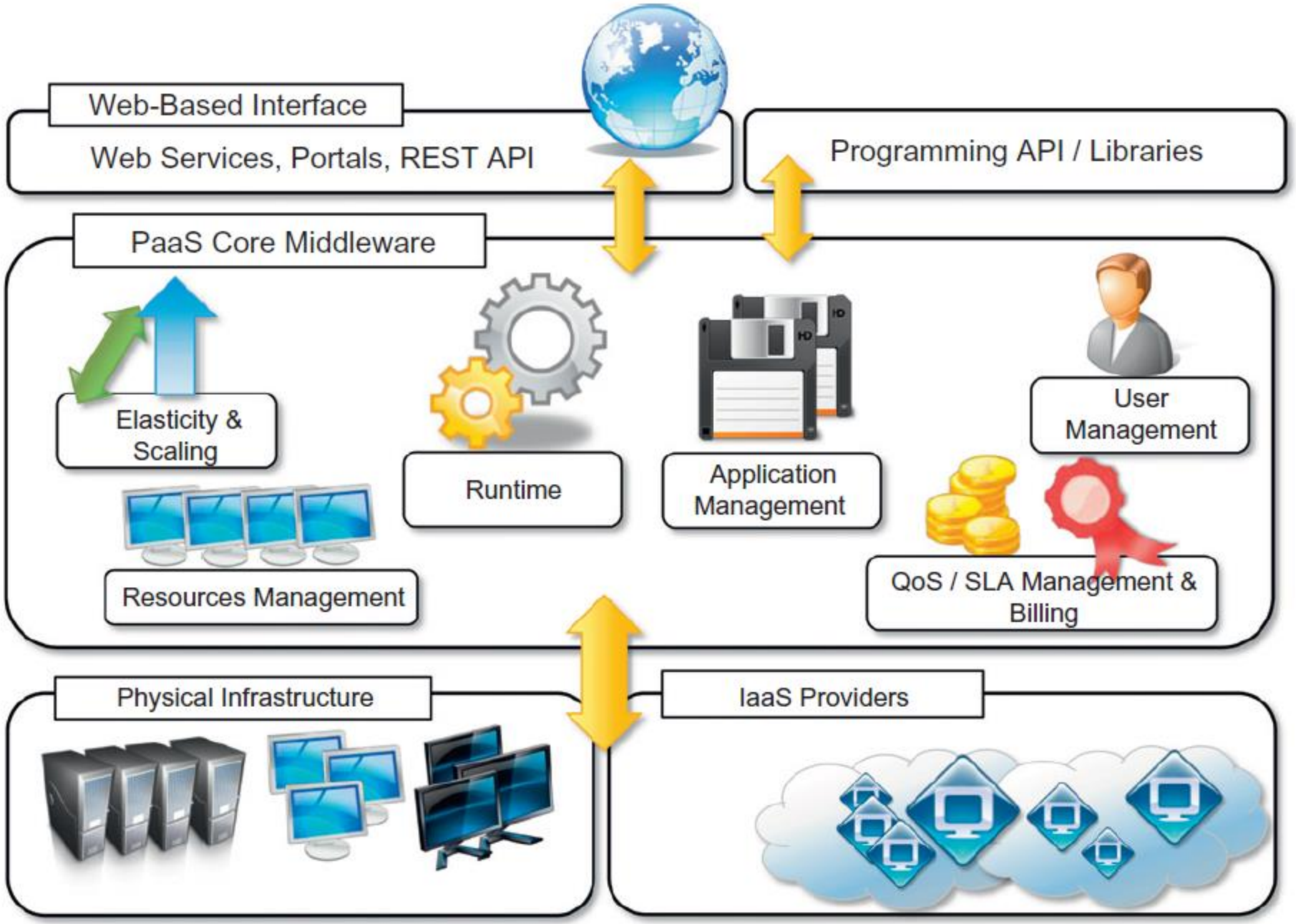

Web-Based Interface
Web Services, Portals, REST API
Programming API / Libraries
PaaS Core Middleware
Elasticity & Scaling
Runtime
Application Management
User Management
Resources Management
QoS / SLA Management & Billing
Physical Infrastructure
IaaS Providers

Table 4.2 Platform-as-a-Service Offering Classification

Category	Description	Product Type	Vendors and Products
PaaS-I	Runtime environment with Web-hosted application development platform. Rapid application prototyping.	Middleware + Infrastructure Middleware + Infrastructure	Force.com Longjump
PaaS-II	Runtime environment for scaling Web applications. The runtime could be enhanced by additional components that provide scaling capabilities.	Middleware + Infrastructure Middleware Middleware + Infrastructure Middleware + Infrastructure Middleware + Infrastructure Middleware	Google AppEngine AppScale Heroku Engine Yard Joyent Smart Platform GigaSpaces XAP
PaaS-III	Middleware and programming model for developing distributed applications in the cloud.	Middleware + Infrastructure Middleware Middleware Middleware Middleware Middleware	Microsoft Azure DataSynapse Cloud IQ Manjrasof Aneka Apprenda SaaSGrid GigaSpaces DataGrid

Essential Characteristics of PaaS

- Runtime framework
- Higher level of abstraction
- Automating deployment and scaling
- Cloud Services and APIs to simplify the creation and delivery of elastic and highly available cloud applications
- Integration of third party cloud services leveraging SOA
- Vendor lockin
- Optimize development, deployment and management of applicaitons

SaaS

- Complete applications as a service
- No hardware, software cost, pay as you go subscriptions
- One-to-many, hosted, some customization
- Multitenant
- SaaS coined in 2001, predates cloud computing, Application Service Providers (ASP)
- SaaS 2.0 rapid achievement of business objectives, integration, interconnected
- Salesforce (CRM), Facebook (social networking), linkedin,Google Documents

Types

Clouds are a type of parallel and distributed system harnessing physical and virtual computers presented as a unified computing resource

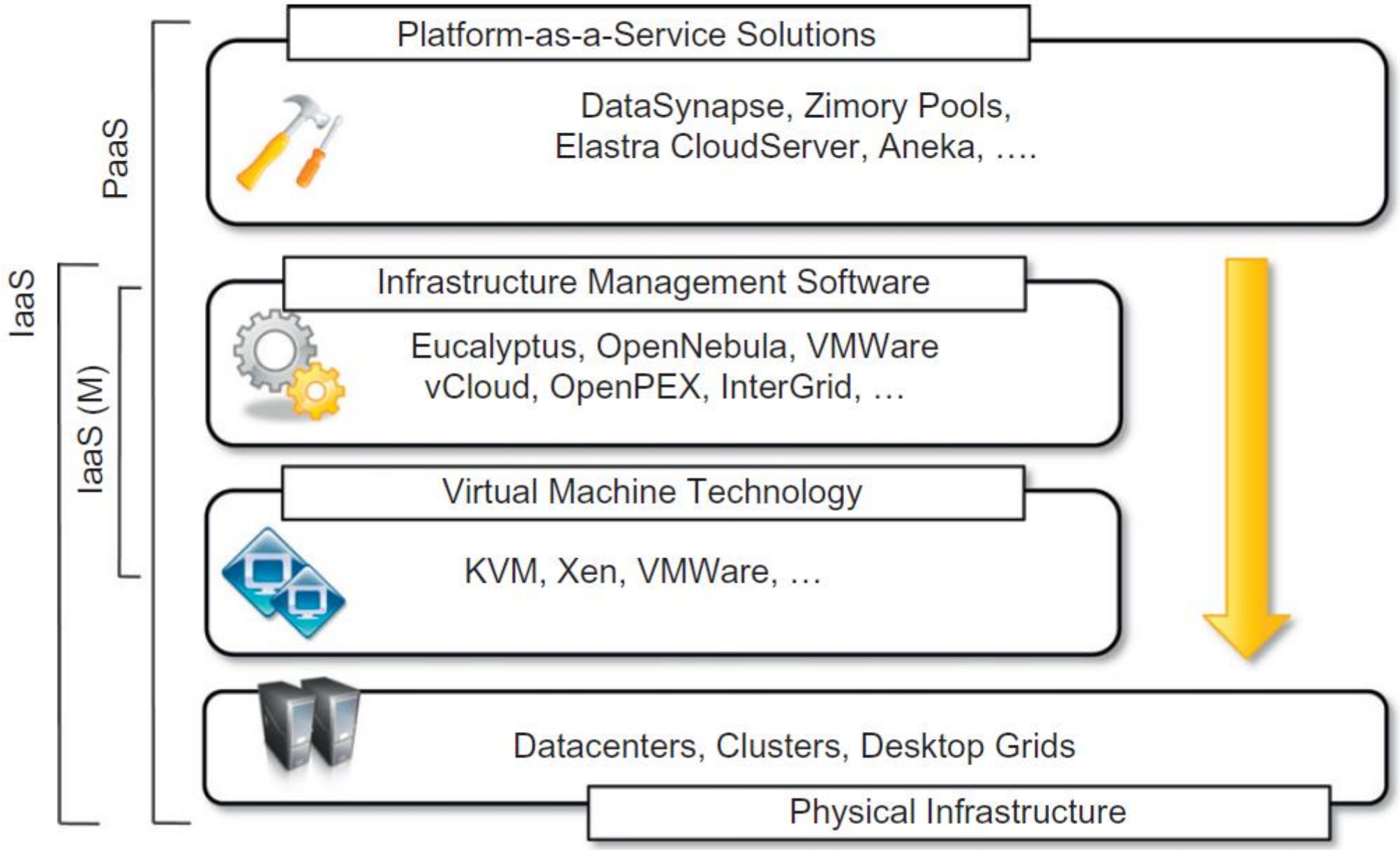
- Public: open to the wider public
- Private: limited to an institution, intranet
- Hybrid or heterogeneous: combination of the above two
- Community: for a specific industry
- Multi: combination of on-premises, private, public, and edge

Public Cloud

- First expression of cloud computing
- Anyone, anywhere, any time via Internet
- Useful to start small enterprises without incurring any capex
- Grow or shrink according to business needs, sustain peak load
- Renting infrastructure or subscribing to services
- Replace or extend
- Multitenancy, QoS
- Monitoring: uptime, usage (for billing), activity
- AWS EC2: IaaS, Google AppEngine: PaaS, salesforce.com: SaaS

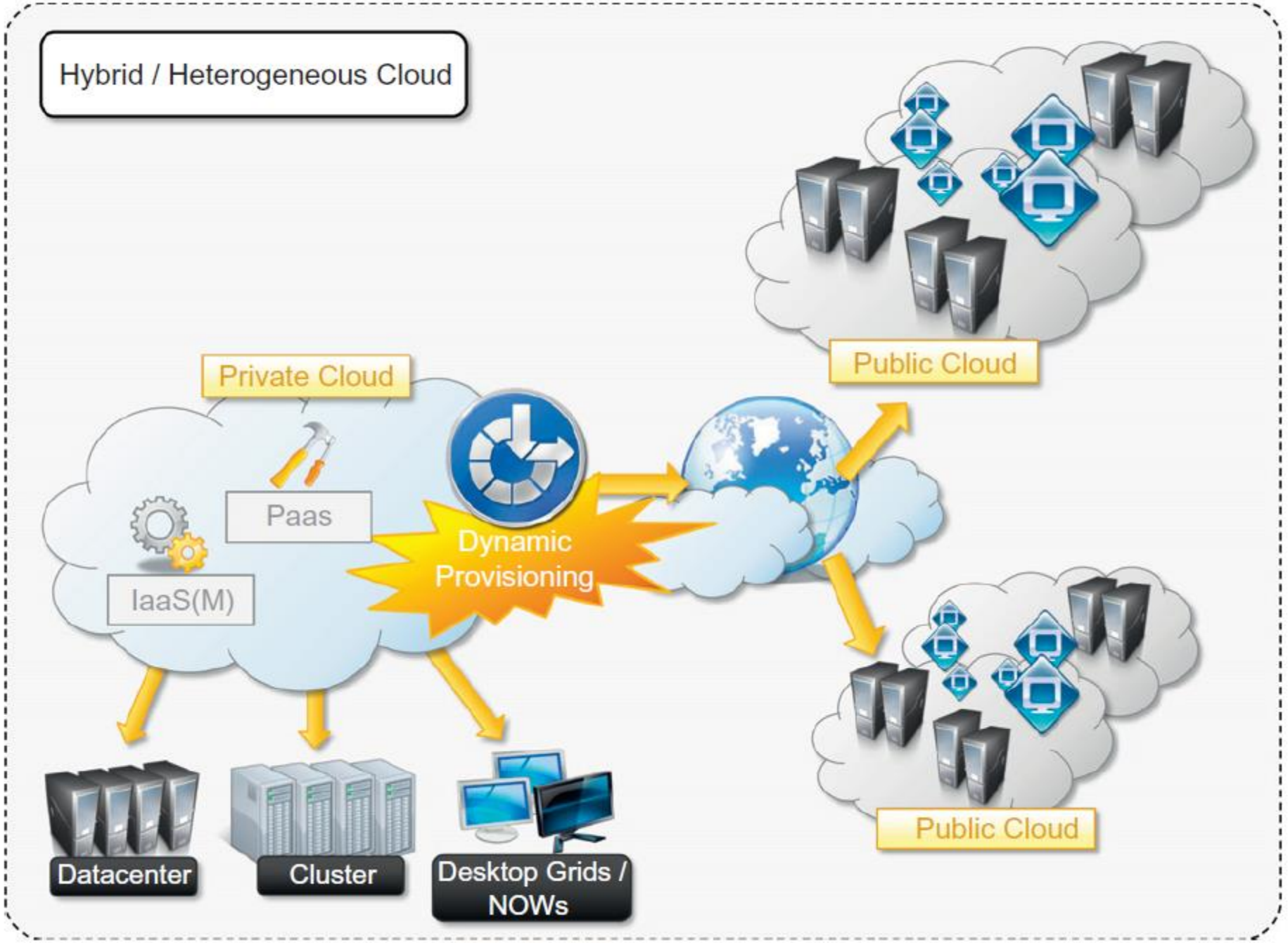
Private Cloud

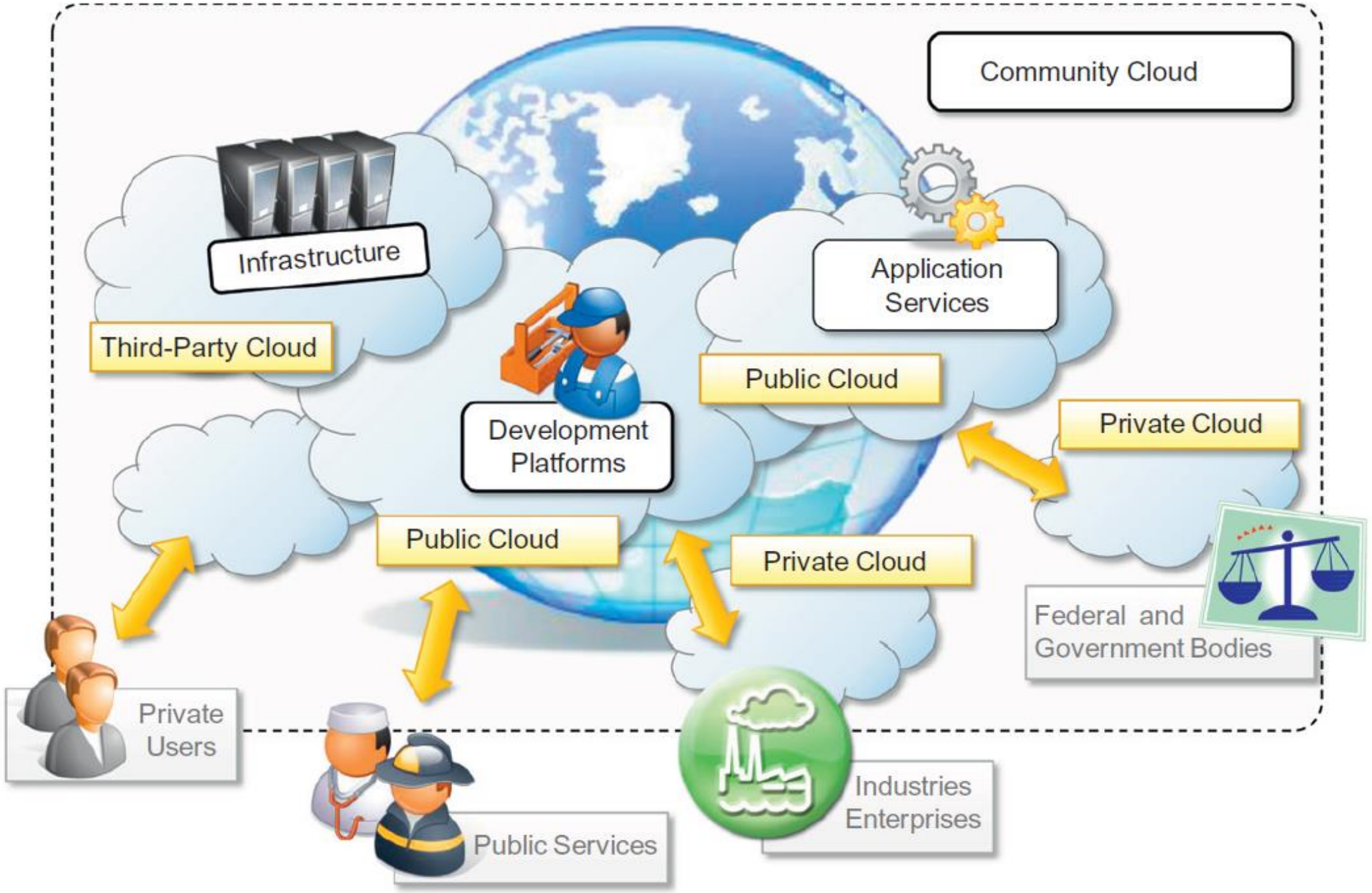
- Control, privacy, government, military agencies
- Depending on where data resides: US APATRIOT Act provides its government and other agencies access to information, including that belonging to any company that stores information in the U.S. territory
- Can leverage existing on prem hardware and software
- Migration of existing infra
- IT services on demand
- Customer information protection, Infra SLAs, Compliance
- Openstack, VMWare vCloud


Platform-as-a-Service Solutions
DataSynapse, Zimory Pools,
Elastra CloudServer, Aneka, ….
PaaS
IaaS
IaaS (M)
Infrastructure Management Software
Eucalyptus, OpenNebula, VMWare
vCloud, OpenPEX, InterGrid, …
Virtual Machine Technology
KVM, Xen, VMWare, …
Datacenters, Clusters, Desktop Grids
Physical Infrastructure

Hybrid Cloud

- Keep processing of information within premises
- Leverage existing hardware and software
- Scale depending on needs and sustain peak load
- private cloud that integrates additional services or resources from one or more public clouds
- Heterogeneous cloud
- Dynamic provisioning to meet demand – cloudbursting
- Mostly applies to hardware
- OpenNebula: cost based advanced Haizea scheduler uses QoS and budget


Hybrid / Heterogeneous Cloud
Public Cloud
Private Cloud
Paas
IaaS(M)
Dynamic Provisioning
Public Cloud
Datacenter
Cluster
Desktop Grids / NOWs


Community Cloud
Infrastructure
Third-Party Cloud
Application Services
Public Cloud
Development Platforms
Private Cloud
Public Cloud
Private Cloud
Federal and Government Bodies
Private Users
Public Services
Industries Enterprises

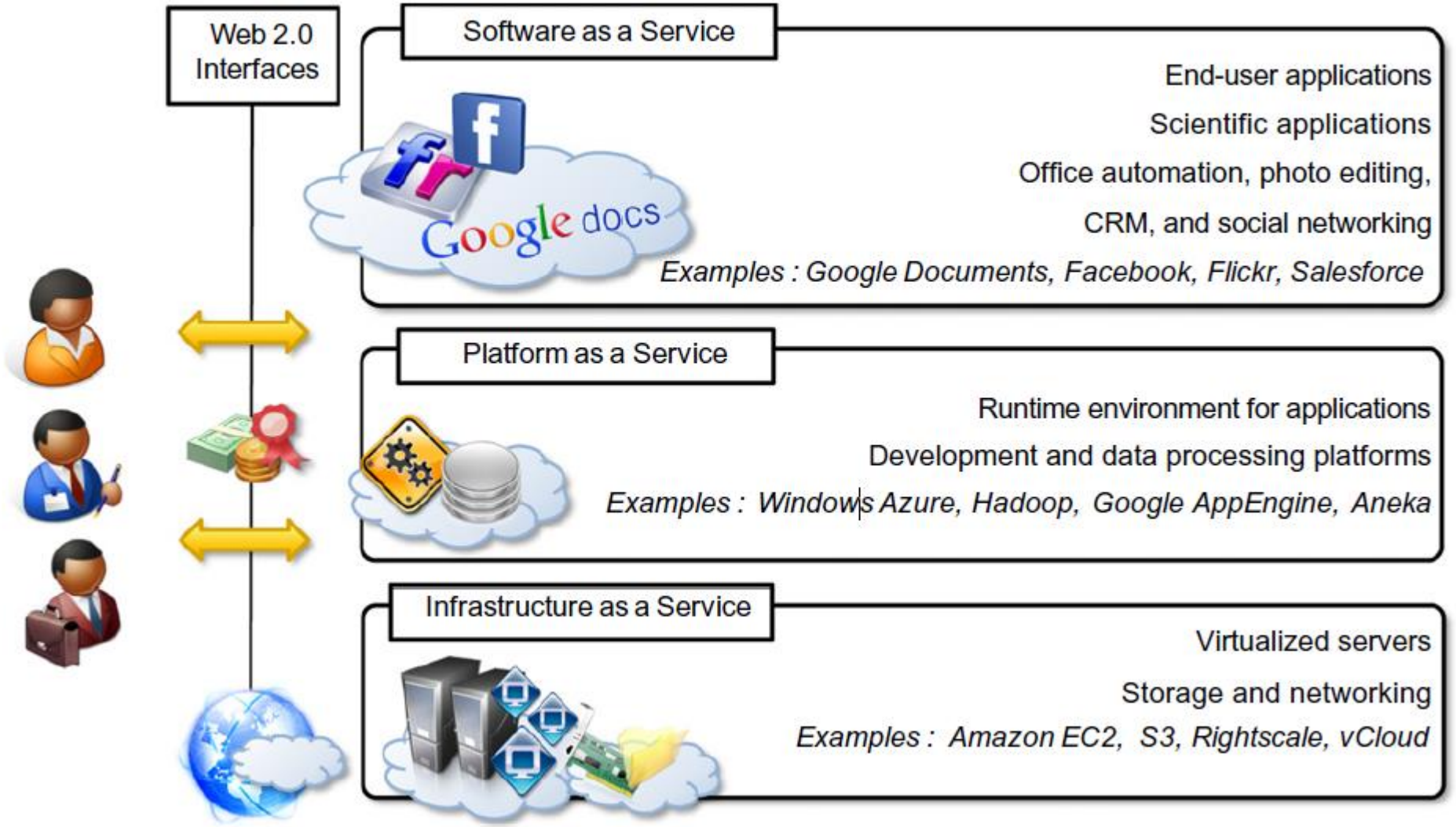
Pricing

- Tiered: different types of EC2 instances
- Per unit: data transfer, per GB storage
- Subscription based: SaaS, for application(s) / service, per month
- Combination of above: per GB per month for s3fa

Assignment

- Google account
- Google drive
- Spreadsheet
- Note

Layers / Reference Model



Cloud Computing Platforms

SAMEER MAHAJAN

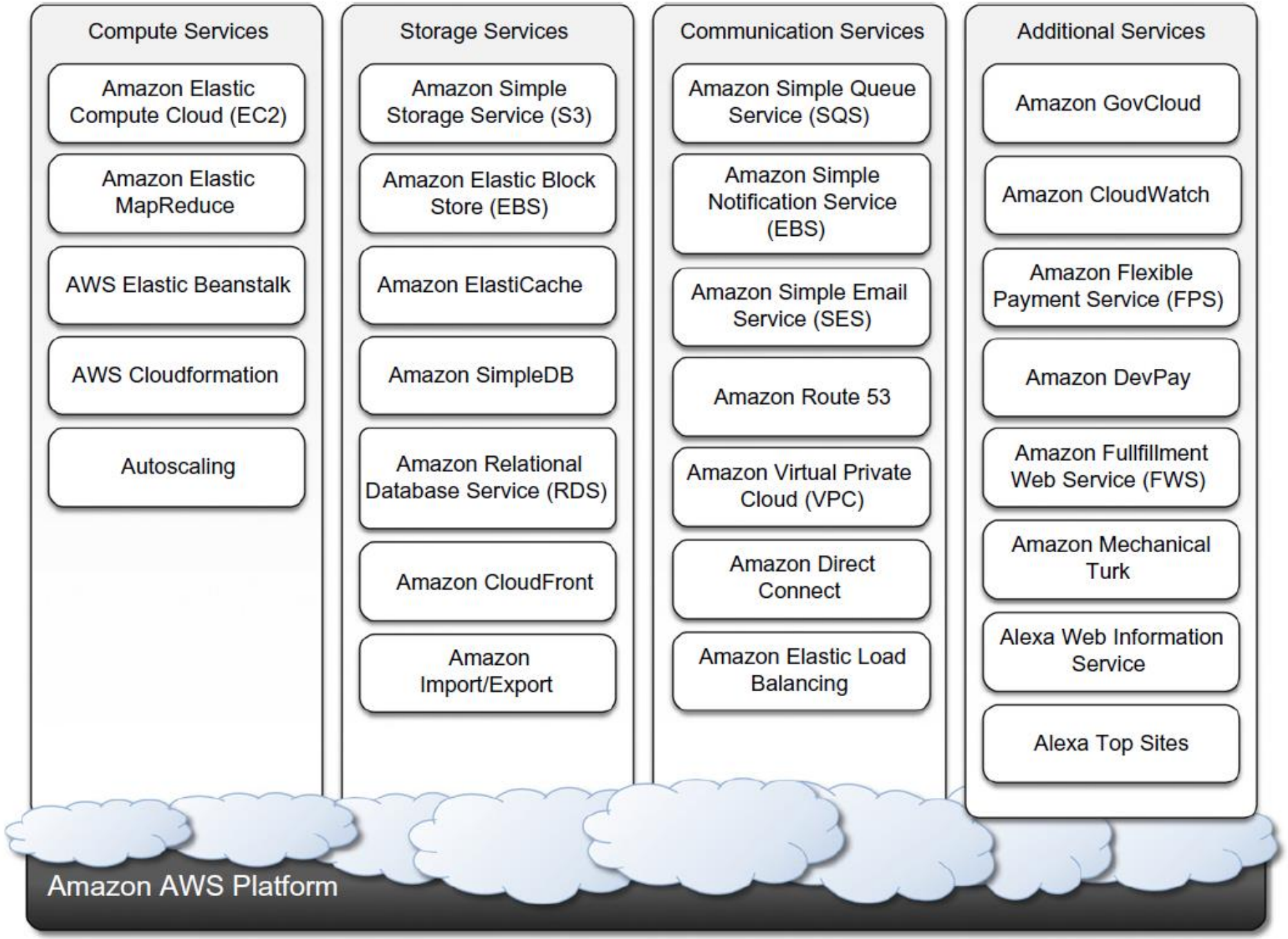
HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

Table 9.1 Some Example Cloud Computing Offerings

Vendor/Product	Service Type	Description
Amazon Web Services	IaaS, PaaS, SaaS	Amazon Web Services (AWS) is a collection of Web services that provides developers with compute, storage, and more advanced services. AWS is mostly popular for IaaS services and primarily for its elastic compute service EC2.
Google AppEngine	PaaS	Google AppEngine is a distributed and scalable runtime for developing scalable Web applications based on Java and Python runtime environments. These are enriched with access to services that simplify the development of applications in a scalable manner.
Microsoft Azure	PaaS	Microsoft Azure is a cloud operating system that provides services for developing scalable applications based on the proprietary Hyper-V virtualization technology and the .NET framework.
SalesForce.com and Force.com	SaaS, PaaS	SalesForce.com is a Software-as-a-Service solution that allows prototyping of CRM applications. It leverages the Force.com platform, which is made available for developing new components and capabilities for CRM applications.
Heroku	PaaS	Heroku is a scalable runtime environment for building applications based on Ruby.
RightScale	IaaS	Rightscale is a cloud management platform with a single dashboard to manage public and hybrid clouds.

Amazon Web Services (AWS)

- SOAP or RESTful web service interface, web based console, CLI, SDK
- Raw compute: EC2, raw storage: s3
- EC2 VM with image having predefined software stack, instance configured with memory, processors and storage, remote access
- Amazon Machine Image (AMI) templates to create a VM, stored in s3 and have unique identifiers, contains Amazon Ramdisk Image (ARI) and Amazon Kernel Image (AKI)
- vCPU / virtual cores and ECU =1.0 -1.2 GHz 2007 Opteron or Xeon processor
- Volatile spot instances with user defined price thresholds


Compute Services
Amazon Elastic Compute Cloud (EC2)
Amazon Elastic MapReduce
AWS Elastic Beanstalk
AWS Cloudformation
Autoscaling
Storage Services
Amazon Simple Storage Service (S3)
Amazon Elastic Block Store (EBS)
Amazon ElastiCache
Amazon SimpleDB
Amazon Relational Database Service (RDS)
Amazon CloudFront
Amazon Import/Export
Communication Services
Amazon Simple Queue Service (SQS)
Amazon Simple Notification Service (EBS)
Amazon Simple Email Service (SES)
Amazon Route 53
Amazon Virtual Private Cloud (VPC)
Amazon Direct Connect
Amazon Elastic Load Balancing
Additional Services
Amazon GovCloud
Amazon CloudWatch
Amazon Flexible Payment Service (FPS)
Amazon DevPay
Amazon Fullfillment Web Service (FWS)
Amazon Mechanical Turk
Alexa Web Information Service
Alexa Top Sites
Amazon AWS Platform

Instance Type	ECU	Platform	Memory	Disk Storage	Price (U.S. East) (USD/hour)
Standard instances					
Small	1(1 × 1)	32 bit	1.7 GB	160 GB	$0.085 Linux $0.12 Windows
Large	4(2 × 2)	64 bit	7.5 GB	850 GB	$0.340 Linux $0.48 Windows
Extra Large	8(4 × 2)	64 bit	15 GB	1,690 GB	$0.680 Linux $0.96 Windows
Micro instances					
Micro	< = 2	32/64 bit	613 MB	EBS Only	$0.020 Linux $0.03 Windows
High-Memory instances					
Extra Large	6.5(2 × 3.25)	64 bit	17.1 GB	420 GB	$0.500 Linux $0.62 Windows
Double Extra Large	13(4 × 3.25)	64 bit	34.2 GB	850 GB	$1.000 Linux $1.24 Windows
Quadruple Extra Large	26(8 × 3.25)	64 bit	68.4 GB	1,690 GB	$2.000 Linux $2.48 Windows
High-CPU instances					
Medium	5(2 × 2.5)	32 bit	1.7 GB	350 GB	$0.170 Linux $0.29 Windows
Extra Large	20(8 × 2.5)	64 bit	7 GB	1,690 GB	$0.680 Linux $1.16 Windows
Cluster instances					
Quadruple Extra Large	33.5	64 bit	23 GB	1,690 GB	$1.600 Linux $1.98 Windows
Cluster GPU instances					
Quadruple Extra Large	33.5	64 bit	22 GB	1,690 GB	$2.100 Linux $2.60 Windows

EC2 Environment

- Storage: Ephemeral or EBS
- Networking: Internal IP (within EC2, client to internet), public IP (remote access over internet), elastic IP (server, failover)
- Security: Access control
- Domain name ec2-xxx-xxx-xxx-xxx.compute-x.amazonaws.com, xxx-xxx-xxx-xxx represents external IP compute-x gives availability zone
- 69 availability zones priced differently, 23 regions (Nov 2019)
- Key pair, security groups, firewall rules

Advanced Compute Services

- CloudFormation: templates, describe resources and services and relations between them required to run an application, integrate EC2 with other services like S3, SimpleDB, SQS, SNS, Route 53, Elastic Beanstalk etc.
- Elastic Beanstalk: package and deploy applications, provision instances, Web applications with Java / Tomcat, WAR file, flexible, higher level
- Elastic MapReduce: Hadoop, EC2, S3, Support Pig, Hive etc., dynamic sizing, selects appropriate configurations of EC2 instances, basic web applications to quickly run data-intensive applications without writing code

Storage Services: Simple Storage Service (S3)

- distributed object store
- Buckets: virtual containers to store objects, cannot be further partitioned, top level, no nesting
- Objects: stored content
- RESTful interface: HTTP Requests (GET, PUT,DELETE,HEAD, and POST)
- Renmaing, modifying or relocating not allowed
- Eventual consistency
- Requests fail occasionally

S3 (contd)

- three different URI ways of addressing
  - Canonical form: http://s3.amazonaws.com/bukect_name/[object_name]
  - Subdomain form: http://bucket-name/s3.amzonaws.com/[object_name]
  - Virtual hosting form: http://bucket-name.com/[object_name] needs DNS
- S3 is flat data store so unique bucket names across all users
- Object ACL: http://s3.amazonaws.com/bukect_name/object_name?acl
- Bucket server logging: http://s3.amzonaws.com/bucket_name?logging
- Metadata are manipulated as attributes of in the request
- Bucket and all its objects belong to a specific region

S3 (further contd)

- List content of bucket with GET
- Name cannot be longer than 1024 UTF-8 encoded bytes
- Names can have '/', way to achieve folder hierarachy
- Maximum object size is 5 GB
- Access Control Policies (ACP): READ, WRITE, READ_ACP, WRITE_ACP, FULL_CONTROL
- Advanced features
  - Server access logging
  - Torrent

Elastic Block Store (EBS)

- Persistent storage in the form of volumes
- Can be mounted at EC2 instance startup
- Up to 1 TB storage, block device interface (raw, FS etc.)
- Stored in s3, cloning, snapshot
- Typically in same AZ, lazily loaded, not shared, multiple volumes
- Cost by storage size in S3, #I/O requests and throughput
- SSD / HDD, gp3 / gp2 / io2 / st1 / sc1
- $0.08/GB-month, 3,000 IOPS free, $0.005/provisioned IOPS-month

ElastiCache

- Elastic in memory cache based on cluster of EC2 instances
- Mecached compatible protocol through web services
- ElastiCache cluster is dynamically (re)sized based on demand
- Automatic patch management, failure detection, recovery
- Priced according to EC2 pricing model with some differences due to the use of the caching service installed on such instances

Structured Storage Solutions

- Preconfigured EC2 AMIs
  - IBM DB2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, Sybase, and Vertica
  - Priced hourly according to the EC2 cost model
  - EC2 user has to configure, maintain, and manage the DBMS
- Relational Data Storage (RDS)
  - Managed MySQL or Oracle
  - Advanced features of multi-AZ deployment and read replicas
- SimpleDB: lightweight, not fully relational, flexible, fully managed
- Aurora

Communication Services

- Among applications and services residing within the AWS infrastructure

Virtual Networking

- Between compute and storage
- Virtual Private Cloud (VPC) and Direct Connect (user network to AWS), Route 53 for naming (dynamic DNS), Identity Access Management (IAM)

Messaging

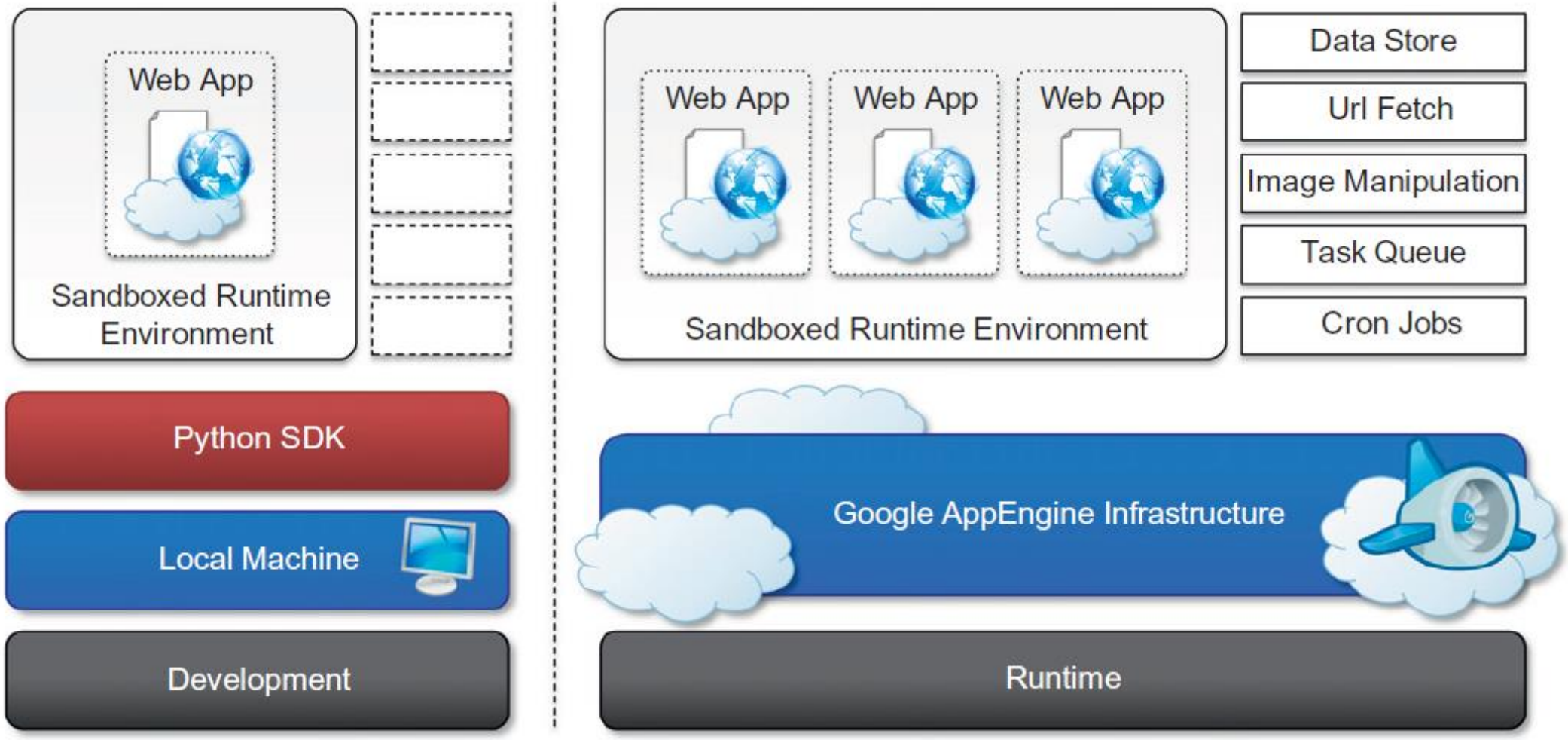
- Simple Queue Service (SQS), Simple Notification Service(SNS):publish-subscribe, Simple Email Service (SES)
- Pay as you go

Additional Services

- **CloudFront**: content delivery network (CDN) on distributed storage infra, leverages strategically located edge servers, reduced transfer time, Web service APIs, distribution with origin server can be in AWS or outside, static or streaming content, access rules, cheaper than S3
- **CloudWatch**: monitoring, earlier subscription, now free
- **Flexible Payment Service (FPS):** sell goods and services to other AWS users
- **Athena**
- **Glue**
- **QuickSight**
- AWS SDK for Python (**Boto3**)

Google AppEngine

- PaaS for developing and hosting scalable web applications
- Java, Python, Golang
- Managed or interpreted languages
- Sandboxing: Cannot write to server's FS, execute more than 30 seconds
- Java Server Pages (JSP), Python Web Application framework (webapp)


Web App
Sandboxed Runtime Environment
Python SDK
Local Machine
Development
Web App
Web App
Web App
Sandboxed Runtime Environment
Data Store
Url Fetch
Image Manipulation
Task Queue
Cron Jobs
Google AppEngine Infrastructure
Runtime

Storage

- **Static File Servers**: CSS, html, javascript static content
- **DataStore**: sem istructured data, object database, BigTable based, entities and properties, entity has immutable key, indexes, transactions, query by specifying either the key or conditions on the values of the properties, pre computed indexes (based on queries) – query execution is independent of data size but influenced only by result set size, entity is updated atomically, multiple updates or multiple entities (in the same entity group) in a transaction, optimistic concurrency control, no exception retrieving an entity

Application Services

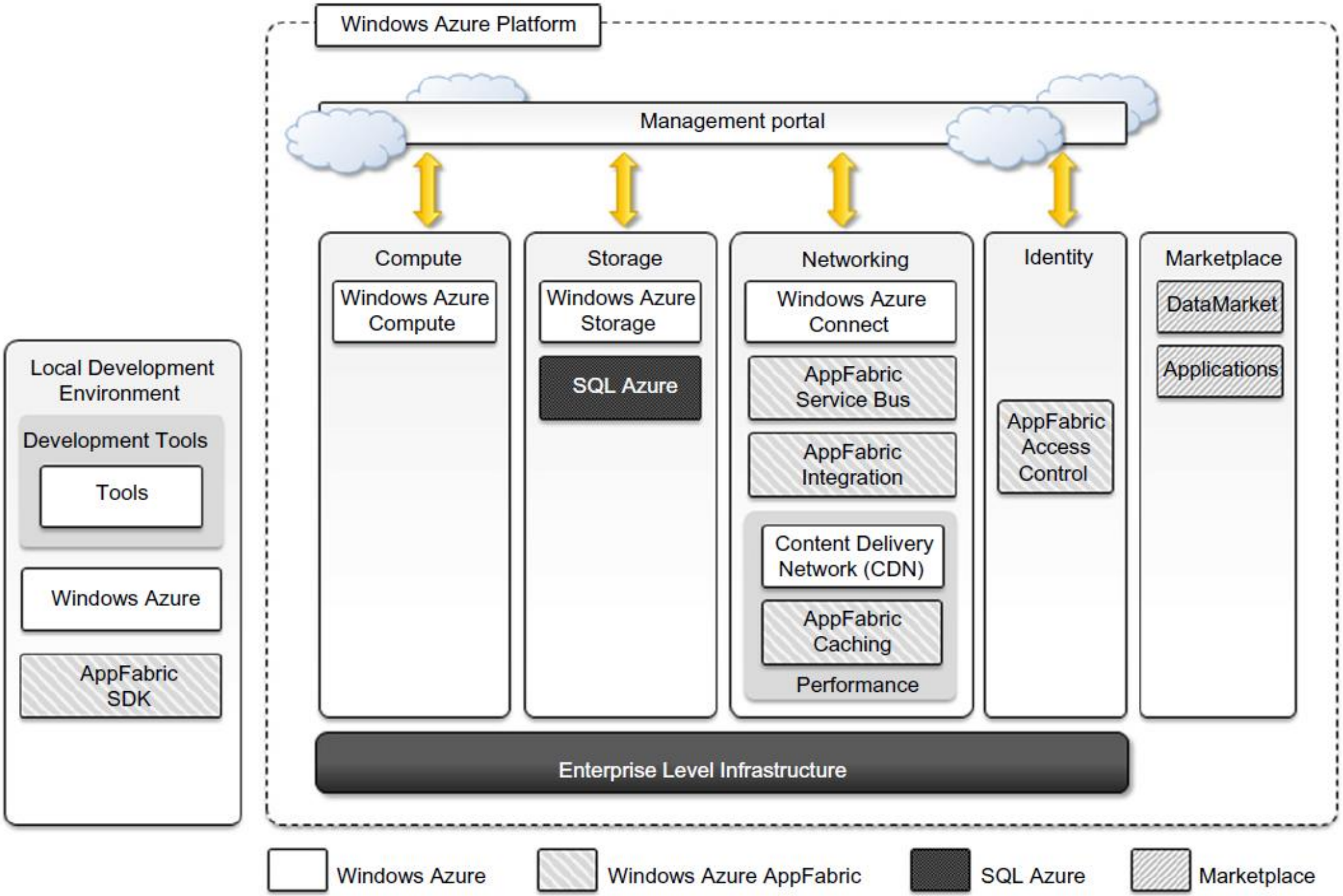
- **UrlFetch**: retrieve remote resources, request can specify deadline, async, integrate meshes into web pages, leverage remote web service
- **MemCache**: distributed in memory cache for fast access
- **Mail and IM**: communication, XMPP for IM
- **Account Management**: Google Accounts, user profiles, authentication
- **Image manipulation**: light weight image processing with speed

Compute Services

- Real time and synchronous within web request timeframe
- **Task queues**: submit a task for later execution, long computations, up to 10 queues, queue invokes request handler for task execution, queue re executes failed task until it is successful
- **Cron jobs**: to be performed at specific time (e.g. nightly), does not re execute in case of failure, maintenance or notifications

Application Lifecycle

- Develop on local development environment, simulates AppEngine runtime by providing mock implementations of DataStore, MemCache, services, hosting web applications, monitoring and tuning
- **Java SDK**: Eclipse with google appengine plugin, servlet, Eclipse web platform, CLI
- **Python SDK**: standalone GoogleAppEngineLauncher tool, integrated webapp, Django, CLI
- Create application identifier (.appspot.com), deploy / upload, manage and monitor by administrative console, versioning, billing
- Free service with limited quotas reset every 24 hours
- billable, fixed, and per-minute quotas; 403 error once quota reached


Windows Azure Platform
Management portal
Compute
Windows Azure Compute
Storage
Windows Azure Storage
SQL Azure
Networking
Windows Azure Connect
AppFabric Service Bus
AppFabric Integration
Content Delivery Network (CDN)
AppFabric Caching
Performance
Identity
AppFabric Access Control
Marketplace
DataMarket
Applications
Enterprise Level Infrastructure
Local Development Environment
Development Tools
Tools
Windows Azure
AppFabric SDK
Windows Azure
Windows Azure AppFabric
SQL Azure
Marketplace

Compute Services

- **Web role**: web applications, hosted on IIS, auto scaled, native support since .Net 3.5, ASP.NET and WCF applications, PHP runtime via FastCGI, non IIS technologies like JSP on Tomcat can also be hosted
- **Worker role**: general compute, do not communicate with the external world through HTTP, background processes, continuously running service, any technology that runs on a Windows Server stack
- **Virtual Machine role**: fully control the computing stack, Hyper-V based, Virtual Hard Disk (VHD) with custom stack

Storage Services

- Durable, not local, can be accessed by multiple clients at the same time and from everywhere
- **Blobs**: binary large objects, metadata, snapshots, CDN
  - **Block blobs**: sequential access, media streaming, 4 MB block, 200 GB blob
  - **Page blobs**: random access, 1 TB
- **Azure drive**: A page blob mounted as part of an NTFS tree
- **Tables**: semi structured storage, entities as rows, properties as columns, key / index, insert, update, delete, select a subset of rows, no schema, partial result sets, partitions for load balancing across servers, 100 TB, 255 properties, 1 MB row, 1 kb keys
- **Queues**: message exchange across applications, FIFO, read message -> invisible, processed message -> deleted, peeking
- Triple Replication

Core Infrastructure

- **AppFabric**: middleware for developing, deploying, and managing applications, language agnostic interfaces
- **Access Control**: rules to access resources, integrates with authentication providers like Active Directory (AD), Windows Live, Google, single coherent identity management framework
- **Service bus**: messaging and connectivity infra, services as URLs, pub-sub, full duplex, peer to peer, multicast, async, connections per month pricing
- **Azure cache**: elastic distributed in memory cache, .Net managed objects, any format, easily integrated with applications, priced by allocated size

Other Services

- **Virtual Network**: Connect – IP based, VM roles and Traffic Manager – load balancing
- **CDN**: static or streaming, 24 locations distributed across the world
- **SQL Azure**: RDBMS, accessible from cloud or any other location having access to cloud, Tabular Data Stream (TDS) protocol, ODBC or ADO.NET, billed by space and edition, Web edition (1 or 5 GB), Business edition (10-50 GB)
- **Windows Azure Platform Appliance**: windows, SQL, configuration of network, storage, hardware

Assignment

- Setting up big data fabric in AWS
  https://www.youtube.com/watch?v=4VJvjnkDDfs

Cloud Applications

SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

Cloud Applications

- Migrate to the cloud
- Bootstrap (SDK, APIs, programming environments) for the cloud
- Develop for the cloud
- Cloud first / Cloud born
- Rich Internet Applications (RIA) with and without browser
- Abstraction based at programming language, environment (e.g. browser) level

Evolution of Abstraction

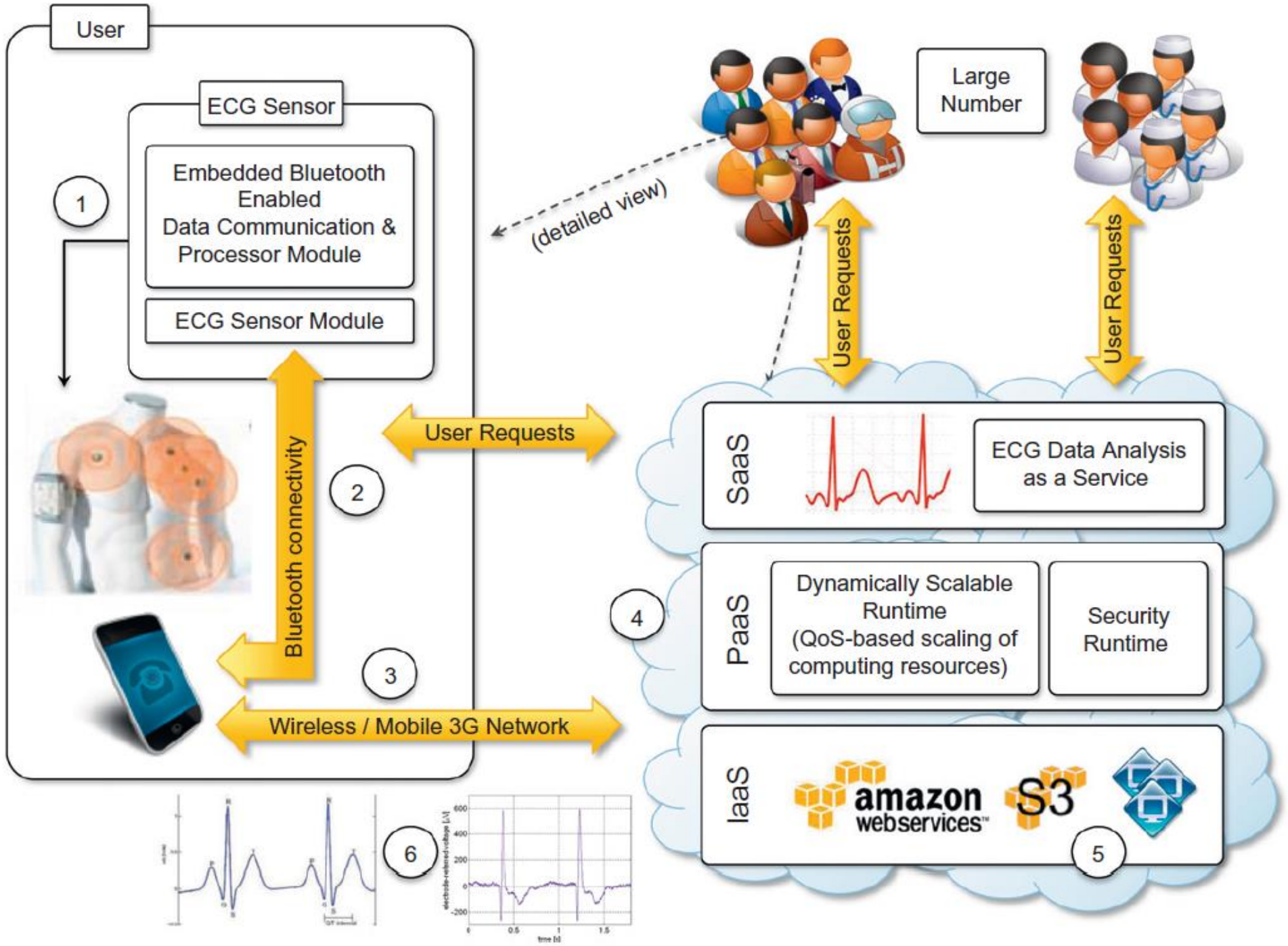
- Programming Language: assembly -> macro assembler -> cobol -> fortran -> C -> Java -> python, Go -> Stats / no code ML / AutoML
- Browser tightly integrated with OS and hardware, browser plugins instead of applications, simple installation, small footprints
- Adobe Integrated Runtime (AIR) : Java, Flash, Windows, Linux; Microsoft Silverlight
- Advantages: ease of development and maintainability
- Disadvantages
  - system overhead (speed, final machine code), not tight, concise
  - security, toolbars have full access to your desktop

Clouds Use Cases

- Times: converting 11 million articles into .pdf for online archive. Estimated 100 servers with 4 TB storage taking months. AWS 100 EC2, 4 TB s3 in a day with cost of only $240
- Washington Post: 200 EC2 server instances to process 17,481 pages of non-searchable PDF images into a searchable online library. 9 hours, a minute per page. $144.62
- Above two are processing and throwaway clouds, use only once
- Storage Clouds: s3, dropbox, google drive, backup, smart switch
- Traveling clouds: USB drive with 5 preinstalled windows servers, mounted locally in AWS for a conference demo
- Occasional use clouds: interopnet VM images / virtual appliance, spun up once for Vegas and once for New York events every year
- DR: storage side synced up at DR site not incurring compute side cost but still warm to able to be brought up immediately in case of disaster without any delay

Scientific Applications

- Researchers and academics
- Availability of compute and sustainable price of storage compared to complete in-house deployment, minimal changes
- High performance (HPC), high throughput (HTC), data intensive
- IaaS, PaaS, MapReduce – data intensive
- Healthcare – diagnostics, health monitoring, ECG analysis


User
ECG Sensor
Embedded Bluetooth Enabled Data Communication & Processor Module
ECG Sensor Module
1
(detailed view)
Large Number
User Requests
User Requests
Bluetooth connectivity
User Requests
2
SaaS
ECG Data Analysis as a Service
PaaS
Dynamically Scalable Runtime (QoS-based scaling of computing resources)
Security Runtime
4
3
Wireless / Mobile 3G Network
IaaS
amazon webservices
S3
5
6

Case Study – Data science based technology platform

About the client

Our client is a technology healthcare startup developing a platform to diagnose mental & physical health of patients based on their voice samples. They have partnered with a premier research lab and licensed speech features pertaining to some health conditions such as depression.

Challenges

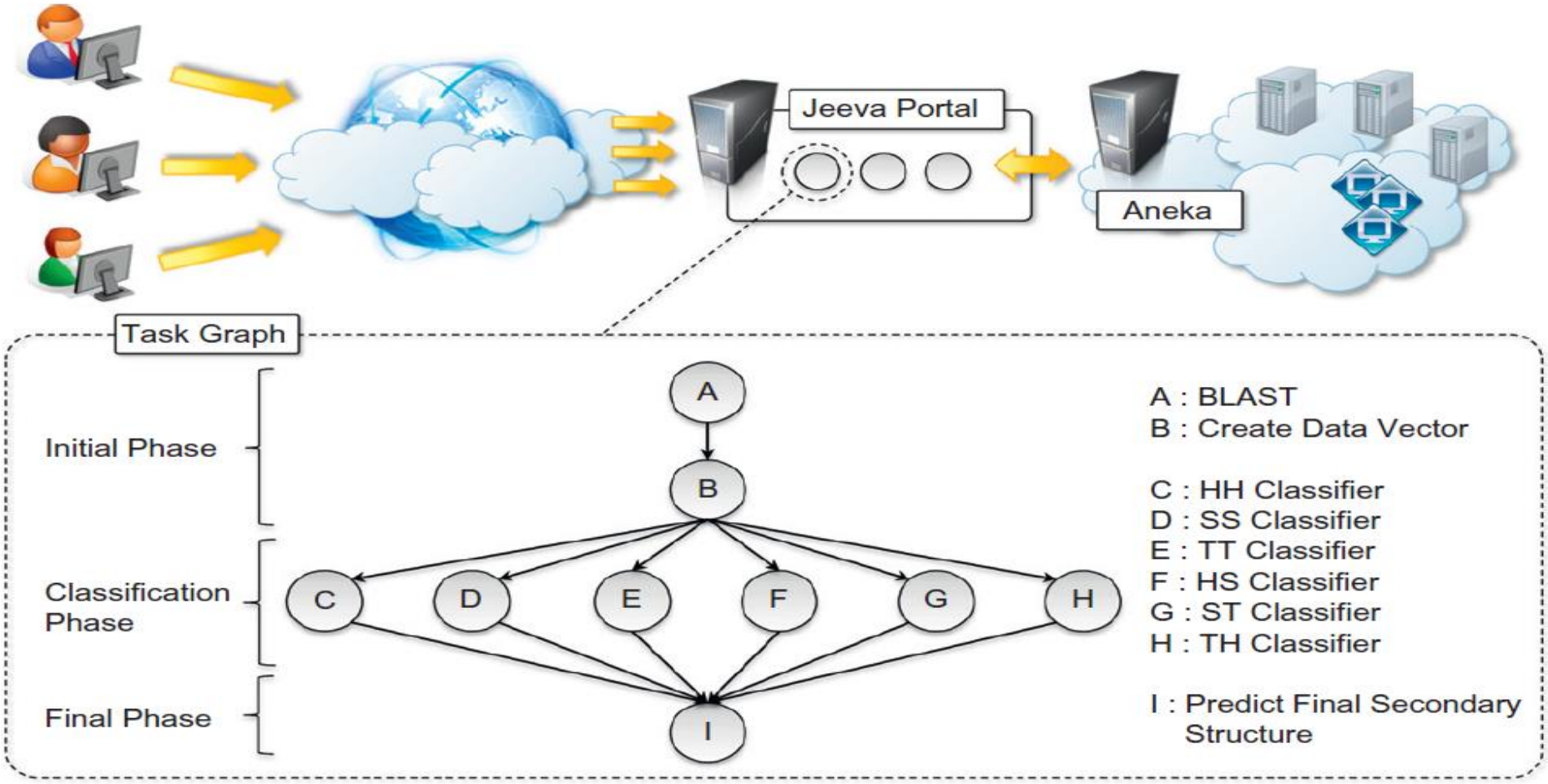
- Create a platform that can score a participant on clinical (depression, Alzheimer's etc.) or wellness (nasality, stress, sleepiness etc.) based on few seconds of voice.
- Develop mobile apps for
  1. Clinical studies with hospitals or pharma partners for discovery of new vocal biomarkers
  2. Organizations to monitor wellness of their employees or end-customers
- Power partner apps through APIs and SDKs
- Accelerate discovery of models for new conditions with help of internal tools

Solution & outcome

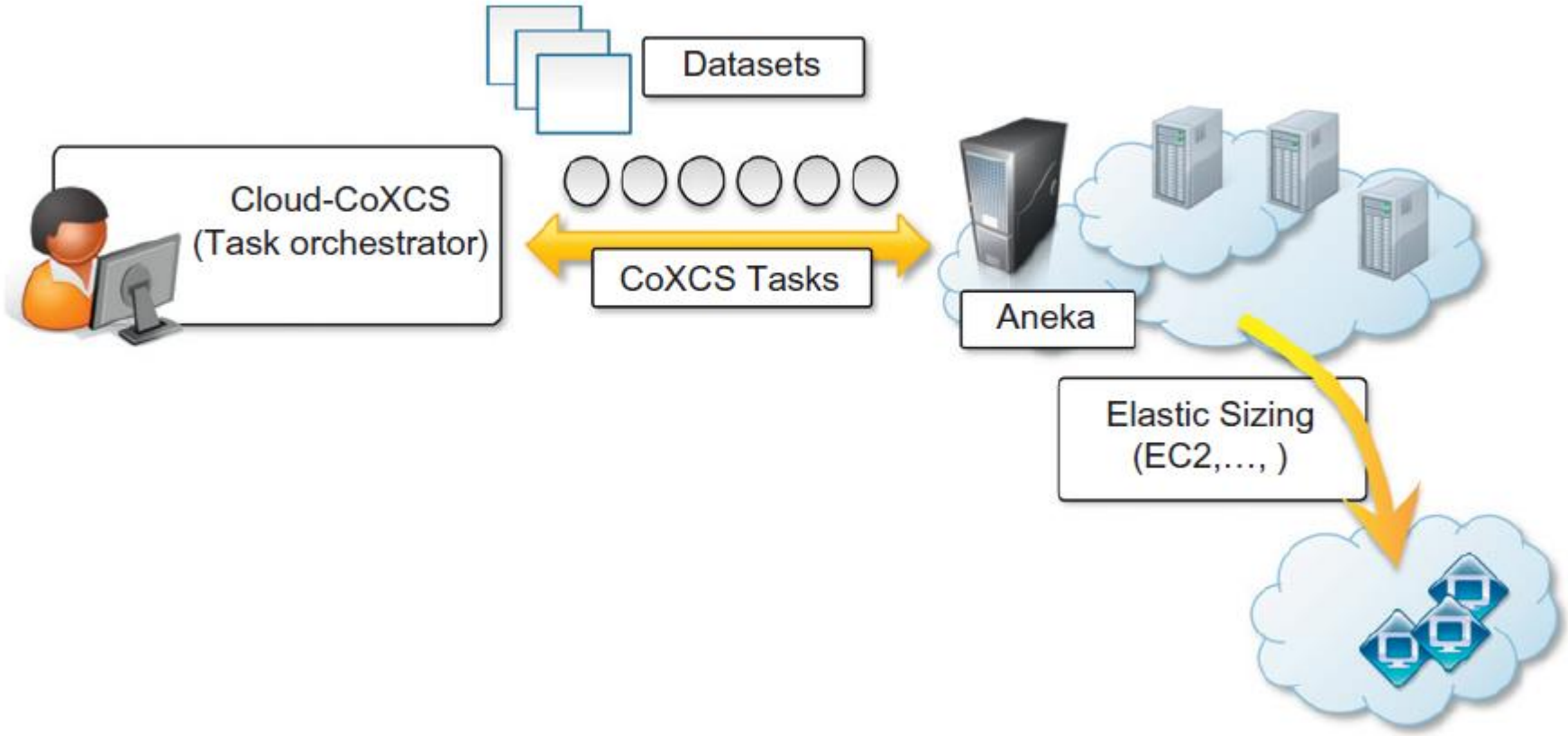
Scalable platform to power mobile apps and partner apps adhering to region-specific legal constraints

'Discovery' of models for new health conditions reduced from few months to few weeks leveraging rapid experimentation and self-service tools and processes

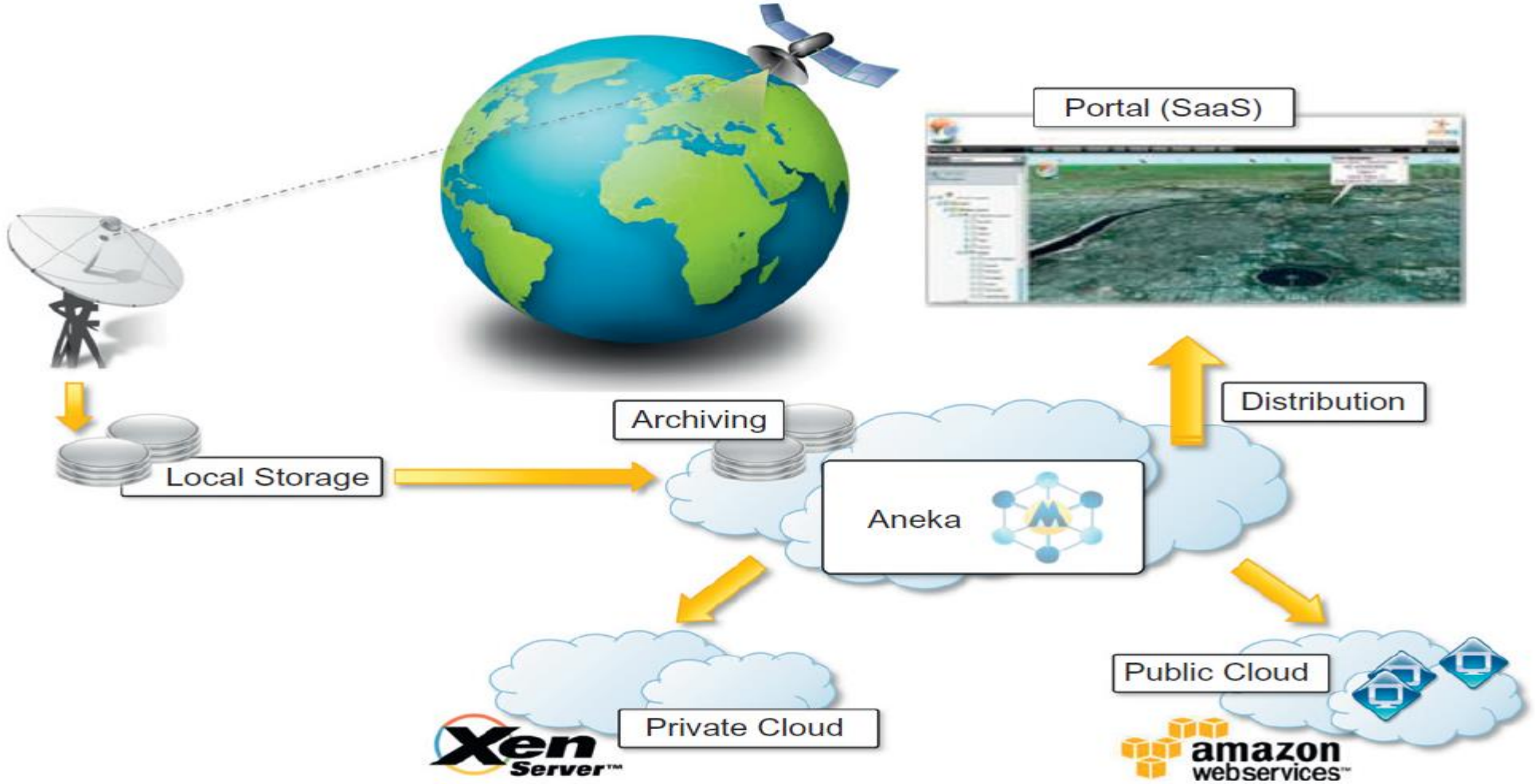
Biology: protein structure prediction



Biology: gene expression data analysis for cancer diagnosis



Geoscience: satellite image processing



Business and consumer applications

- Transform capital costs into operational costs
- Data and services can be accessed from anywhere, anytime, anyone
- Elastic nature with auto scaling – quickly release products and services
- Customer Relationship Management (CRM): salesforce.com – 100k customers, Microsoft dynamics CRM – 99.9% SLA, NetSuite
- Enterprise Resource Planning (ERP): finance, accounting, HR, manufacturing, supply chain, project management, NetSuite

More applications

- Productivity
  - Dropbox: synchronize any file across any platform, any device, through browser or client
  - Google docs
  - Cloud web desktop: EyeOS, Xcerion XML Internet OS/3 (XIOS/3)
- Social networking: facebook, twitter, Instagram, linkedin
- Media applications: Netflix, youtube
- Multiplayer online gaming

Provider Support

- AWS: supports all languages, environments
- GCP: web and google centric
- Azure: .Net and windows centric
- Blurring lines between mobile and desktop,1000+ contacts on mobile phone backed up in cloud

Cloud Security

SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

The Spectrum

- Two categories
  - Assets (hardware, software, network infrastructure) – traditionally
  - Data – more important today
    - At rest
    - In motion
- User Authentication: only authorized users can access resources
- Filtering: only authorized data flows into or out of network, unauthorized data should be unusable
  - Firewall: rule based (devices, appliances)
  - Deep packet and pattern inspection: signatures
- Monitoring: preventing any breach

Authentication

- Logically centralized but physically distributed authentication directory
- HQ Master – cloud / sites slaves
- Read only copies at sites
- Identity federation, Single Sign On – single authentication to identify and authorize across multiple systems
- Network Access Control (NAC) – endpoints are secure, especially mobile
- Agent based, communicates status and configuration to the central server, receives and performs any necessary updates to keep the endpoint latest and secure, polling from server based
- No access at all if not authenticated in case of NAC / cloud
- Microsoft's version is called NAP

Cloud Filtering

- Firewall operates at low network level, ports, protocols, IP addresses, filter incoming traffic
- IDS, Intrusion Prevention System (IPS), Unified Threat Management (UTM) look for patterns, filter incoming as well as outgoing traffic
- Web filters operate at higher level and look at data stream content
- Data leakage prevention systems filter outgoing traffic
- IDS is like burglar alarm, IPS is real time
- Email filtering, application blocking
- Takes significant computing power
- Only tap at source to redirect and process elsewhere, process at source leveraging auto scale and reduce traffic

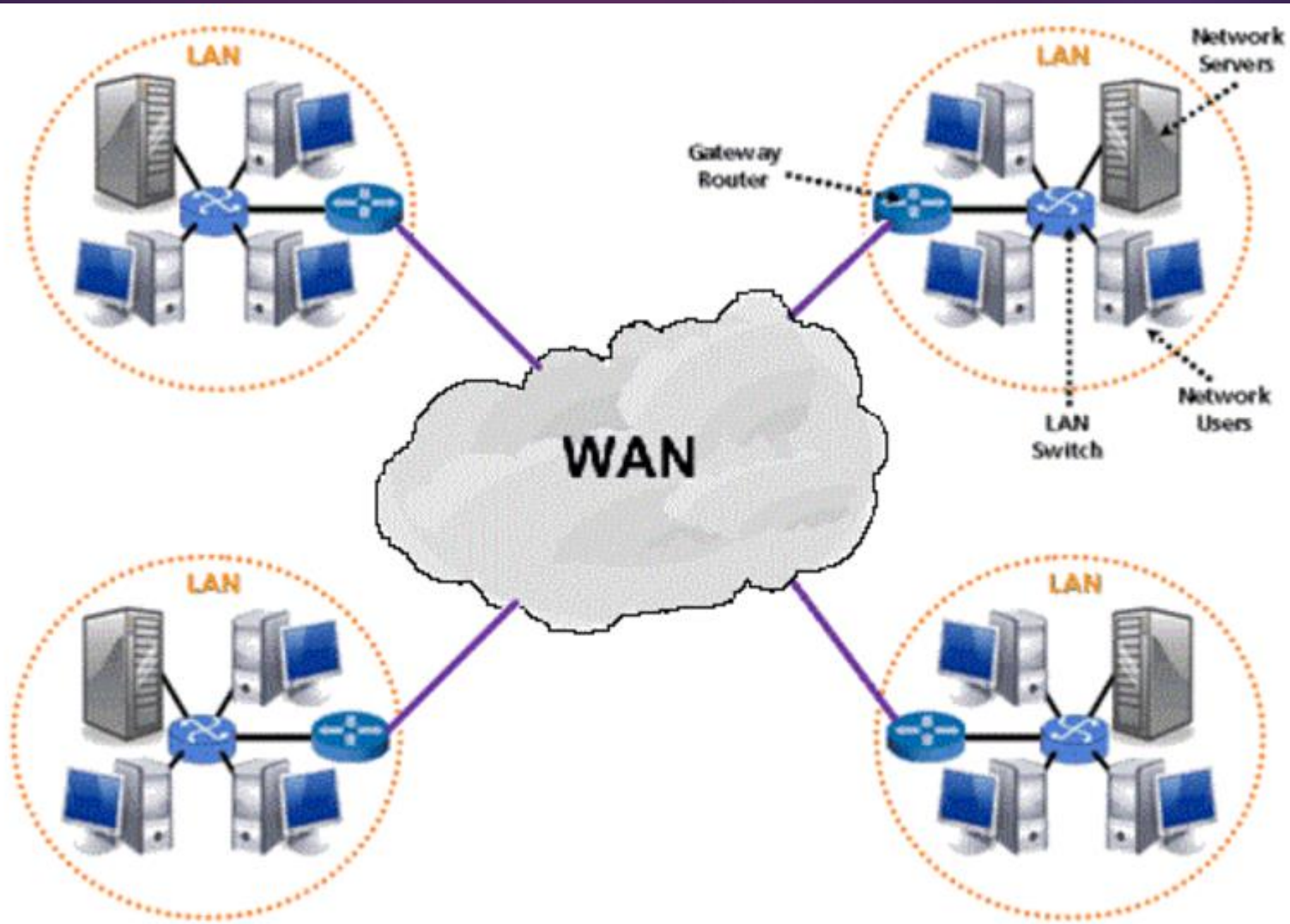
Internal Network Firewall Rules (Drag and drop rows by the left to change rule order)

Rule	Protocol	Source	Destination	Action	
AlexJumpBox	Any	192.168.10.72	192.168.10.0/24	⊘	✖
AdminAllowAll	Any	192.168.10.104/29	Any	✔	✖
CLJumpBox	Any	192.168.10.104/29	10.88.11.4	✔	✖
R-CELive	Any	192.168.10.157	10.88.11.4	✔	✖
N-CEDemo	Any	192.168.10.158	10.12.0.12	✔	✖
LocalAllow	Any	192.168.10.128/26	192.168.10.192/26	✔	✖
Default	Any	Any	Any	⊘	

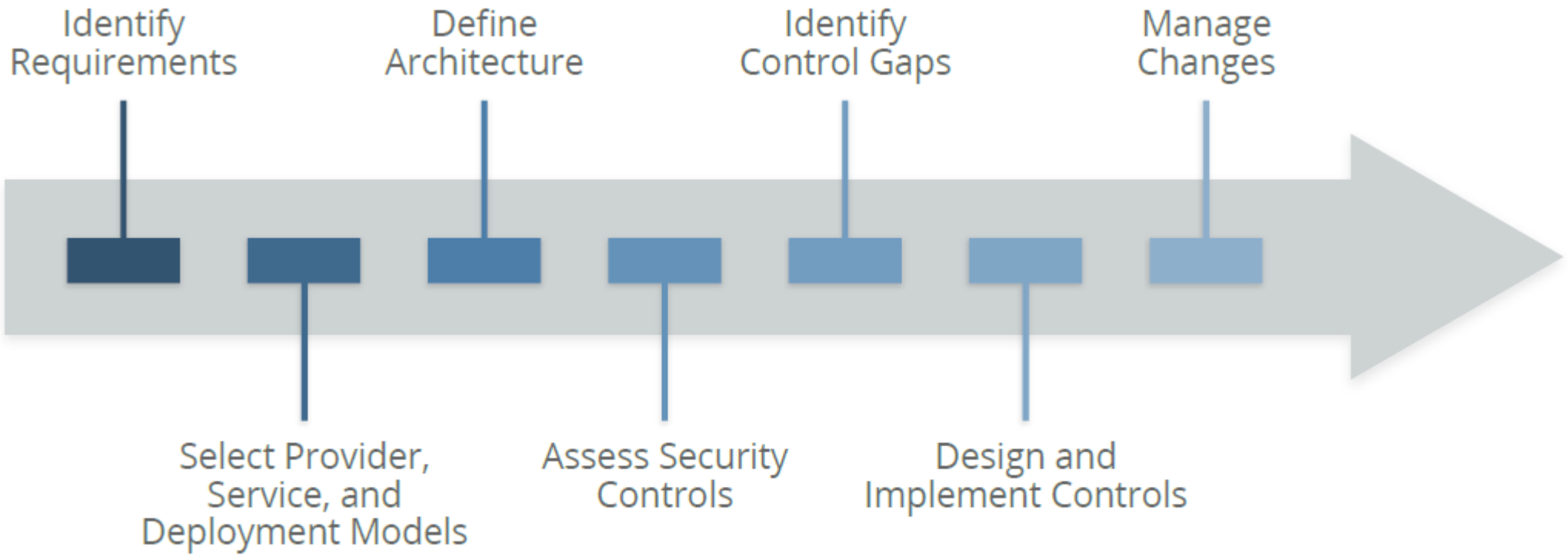
Add Rule

Firewalls

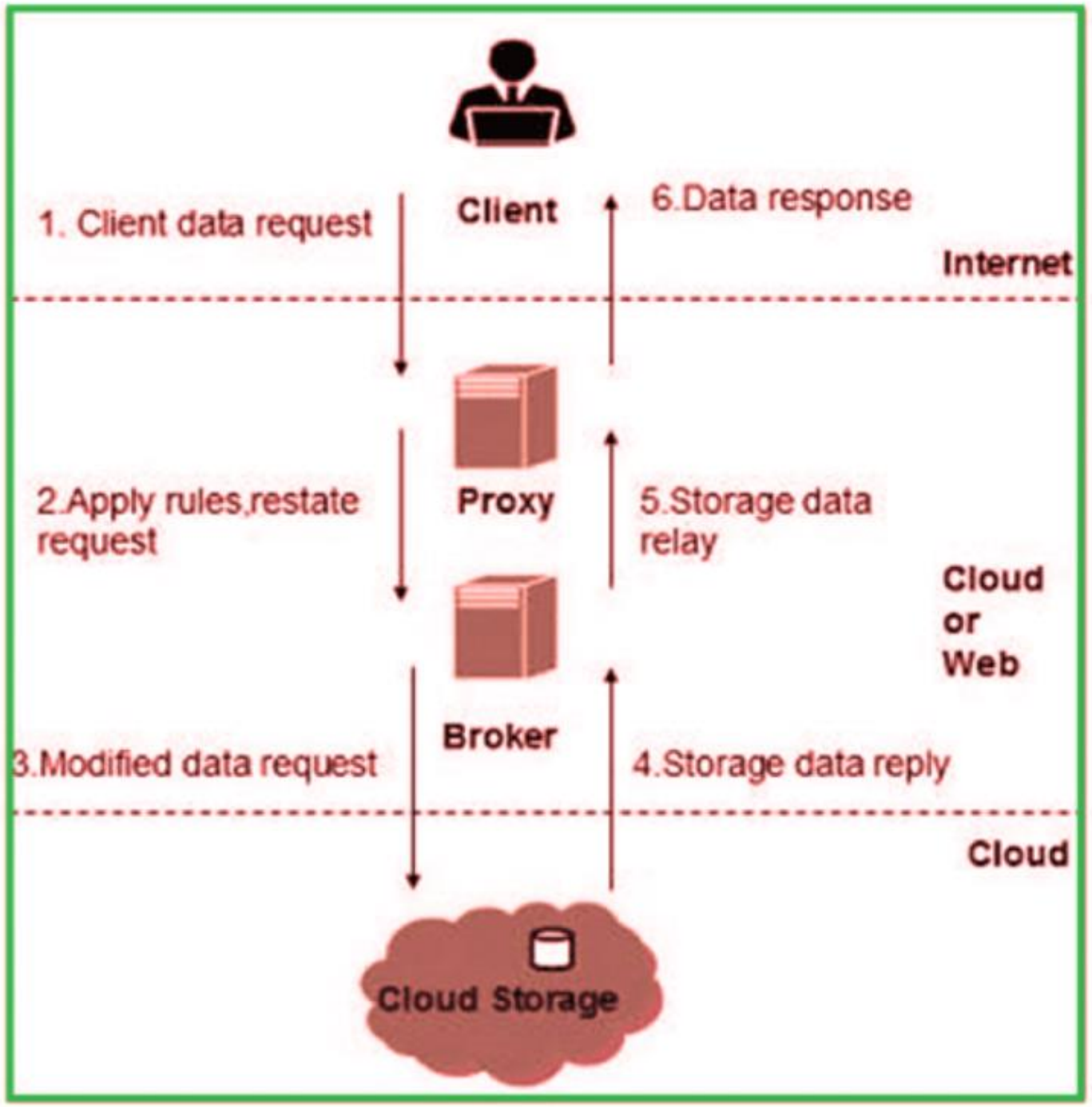
- Cisco
- SonicWall NSA 16 core caveum
- Astaro VM
- Autoscale firewall VMs from load balancers
- Autoscale servers and firewalls in front of them
- VM in cloud can also be monitored for its performance using Simple Network Monitorning Protocol (SNMP) or Windows Management Instrumentation (WMI)
- Virtual Private Network (VPN) to connect to protected virtual networks


LAN
LAN
Network
Servers
Gateway
Router
WAN
LAN
Switch
Network
Users
LAN
LAN

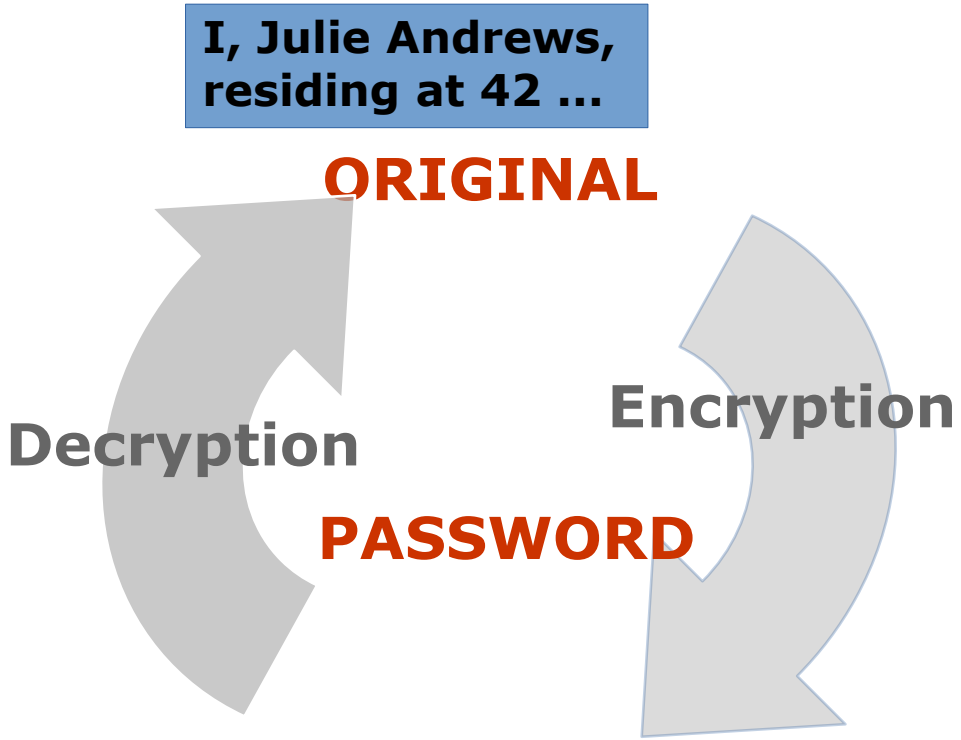
Cloud Security Alliance



Broker
Cloud
Storage



Encryption Using Normal Passwords – data at rest



How it works:

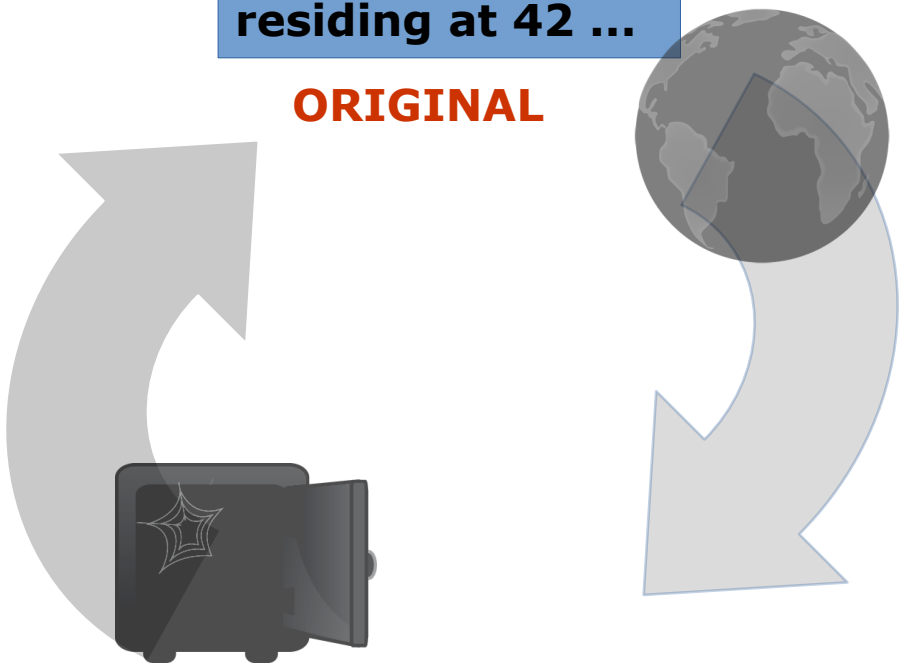
- At encryption
  - » Software asks for password
  - » You can pick any password
- At Decryption
  - » Software asks for password
  - » Must provide same password

Encryption Using 2 Passwords – data in motion

I, Julie Andrews, residing at 42 …

ORIGINAL

PASSWORD1

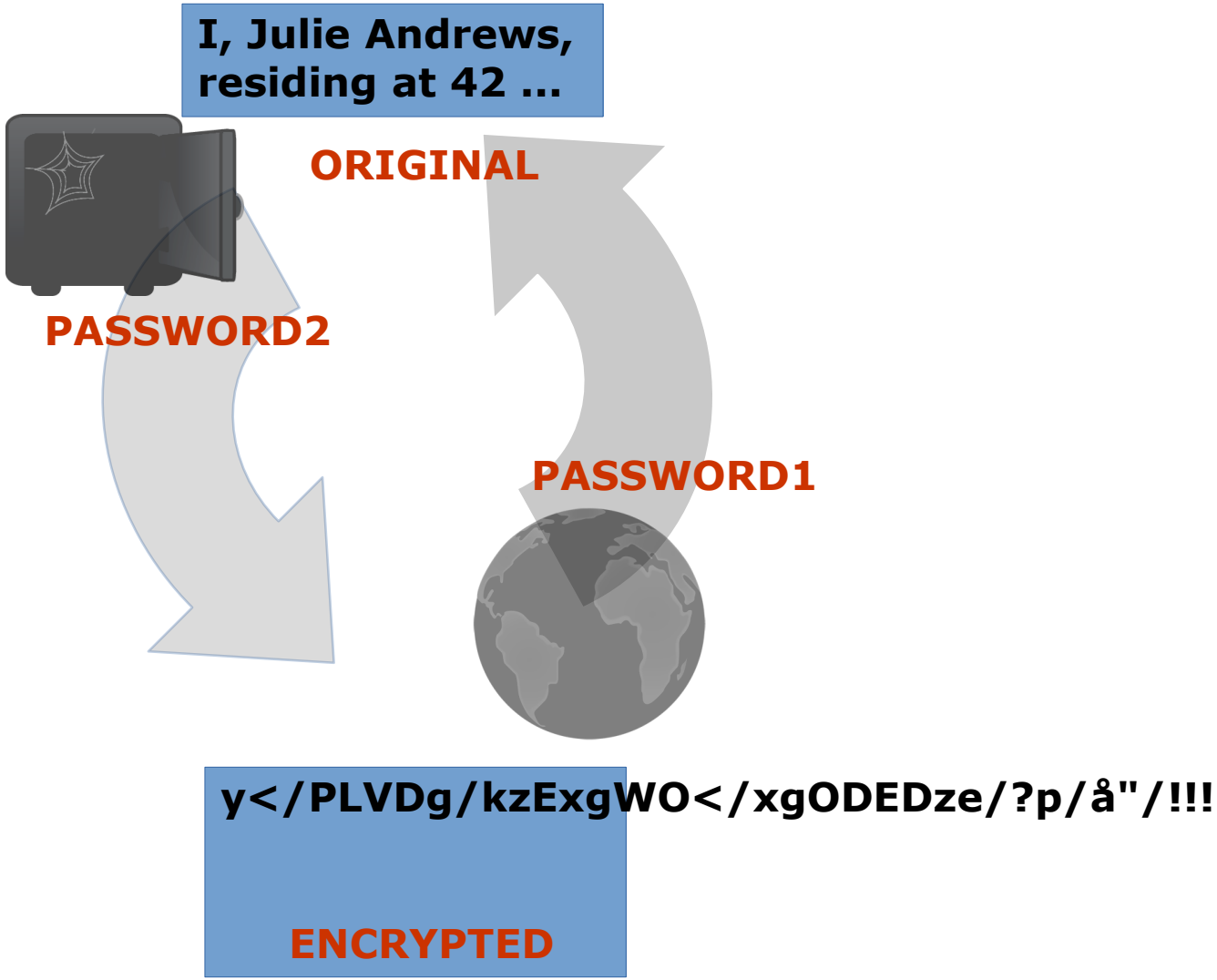


PASSWORD2

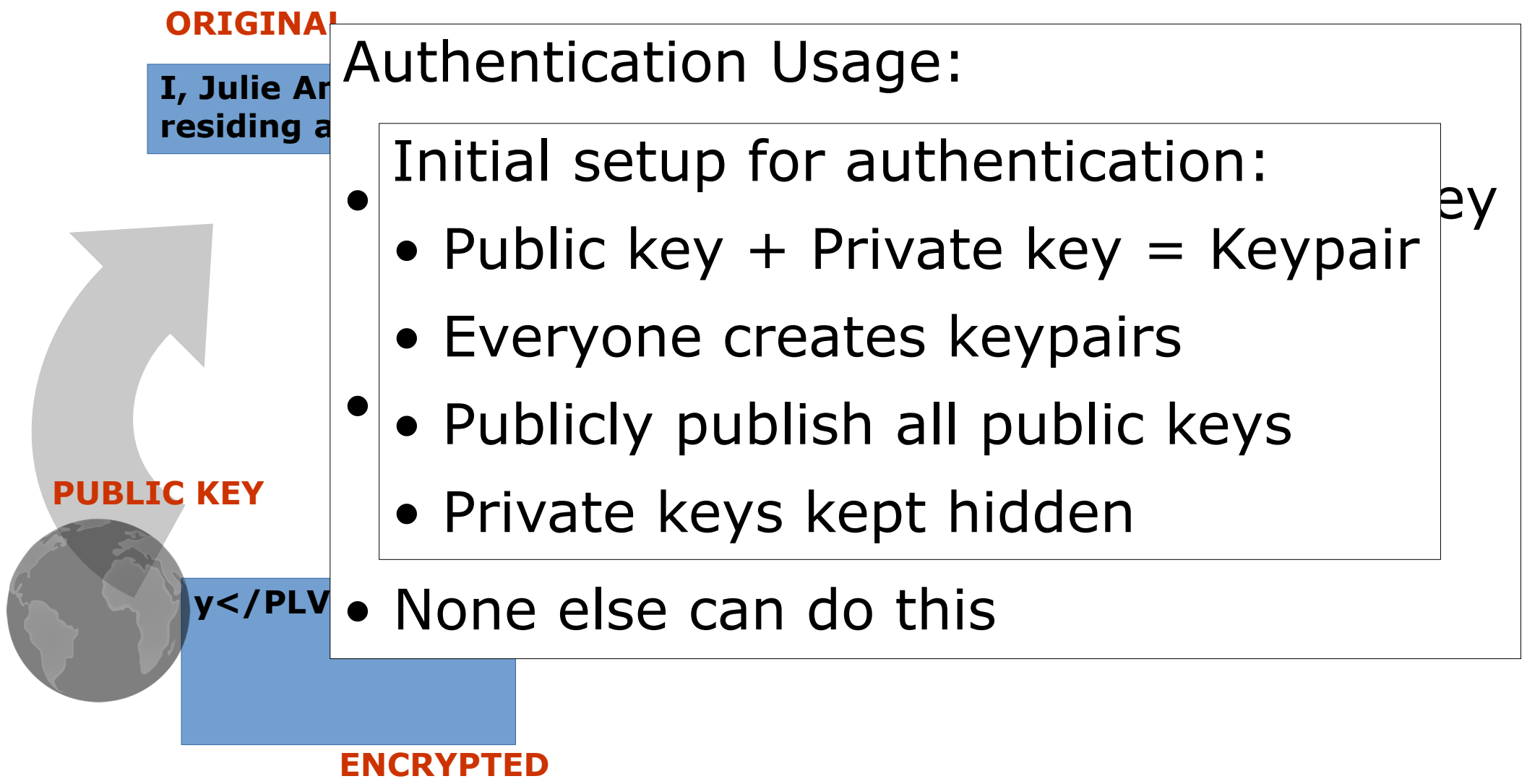
y</PLVDg/kzExgWO</xgODEDze/?p/å"/!!!

ENCRYPTED

Works in Reverse Too



Authentication



Authentication Usage:

- Initial setup for authentication:
  - Public key + Private key = Keypair
  - Everyone creates keypairs
- Publicly publish all public keys
- Private keys kept hidden
- None else can do this

Advantages

- Bad traffic can be intercepted and stopped before it ever reaches the enterprise network to avoid any liability, threats
- Mobile devices are vulnerable to malware, not many antimalware for mobile and have overhead, cleaner data for lighter clients
- Filter at source, spam email filters, easy to catch in transit malware
- Security systems as building blocks, can be changed easily to adapt
- Spin up another IDS when malware detected requiring more resources
- Eliminates single point of failure
- Abilities on demand – download karate like Neo in The Matrix

Limits

- Speed or bandwidth related
- Willingness of management to trust security in cloud
- Regulators' reluctance, consider abstraction layers as obfuscation
- Existence of on prem firewalls preventing cloud based firewalls adoption
- Customization of authentication systems (granularity of identifying groups of users, user defined fields)
- Regulators and cloud providers working together towards cloud security

Future

- Improving compute and bandwidth is helping
- Fuzziness of network boundary due to mobile devices, telecommuting or work from home (WFH) employees
- Use case: Card swipe for physical entry and VPN login for the same user from another country at the same time
- Single security infrastructure for various OS, platforms, from anywhere
- More security
- Ease of management with single pane

Cloud Challenges

SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

Stability

- How many internet feeds you have? Do they come into your data centers through different cable paths? Consider road digging and fiber cut!
- Kind of backup power. Kind of fuel for generators and how much they have. Are they tested and serviced regularly.
- Where is DNS hosted? Its availability on site as well as in Internet.
- NAT, network demarcation point, out of band network
- Remote access to consoles
- Change management system to track power down
- Physical security like access cards etc.
- Administrators, on site spare parts

Partner Quality

- Provider's relationship with its partners
- (Not so normal) circumstances like slammer
- How fast provider can work with partners setting up ACLs to block attacks
- Inadvertent changes in internet routing e.g. BGP peering table
- .edu to .com within city of Honolulu went all the way through Chicago!

Longevity

- How long your provider can support you
- Cloud provider shuts down: PictureBug closed in October 2007
- Cloud failure can bring your business to standstill
- Imagine google (along with its gmail and google accounts) close down!
- Can I get my data, apps, images back?
- Advance notice, contracts, enforcements
- Look out by provider's financial reports
- Cash rich like Amazon or backed by experience and trustworthy
- Provider's staffing, automation, available templates, documentation

Service Level Agreement (SLA)

- Few risks and many benefits as possible
- Troubleshooting capabilities
- Should / should not be negotiable
- Cover everything that can possibly go wrong
- Cover damages in case of business impact
- As quantitative and objective as possible
- Exercise regularly and not only when things go wrong
- Keep data ownership with you
- Have plan B ready
- Ensure physical jurisdiction, tech and legal hand-in-hand

Reaching agreement

- Communication between VMs on the same host, tapping traffic for analysis
- Promiscuous mode, peeking into vSwitch traffic
- Quality of Service: understandable voice over VoIP, not much interruptions due to buffering while watching a video, performance expectations, user experience, profiling, measurable metrics
- Don't rely completely on the provider for security but have your own layer
- Control over failover and load balancing in cloud
- Regulatory : putting in cloud considered as publishing giving up patenting rights, auditing

Business factors

- Who has control over data
- Training and personnel cost, hidden costs
- Regulations, where is my data (load balancing)
- Abstraction layers make it difficult to enforce policy (proxied internet access)
- Does cloud support your applications or is migration required?
- Bandwidth support (cost, apps, firewalls) to and from external cloud
- Speedtest.net, iperf, ixia chariot application
- Load balancers: traditional, dedicated servers for peek load, vMotion
- On prem – cloud, salesforce - SugarCRM combos
- Websites, liability, SMBs partnering with VARs
- Collaboration through cloud, shared s3 storage without compromising privacy
- Identity federation (LDAP, eDirectory, Active Directory)

Planning and Future for Cloud

SAMEER MAHAJAN

HTTPS://WWW.LINKEDIN.COM/IN/SAMEERSMAHAJAN/

Licensing

- Per CPU, VM has single CPU but physical machine may have 8
- Multiple copies on VMs on the same physical machine
- License dongles
- License servers
- Some versions may not support virtualization
- Overhead of abstraction, tuning for optimal throughput

First Steps

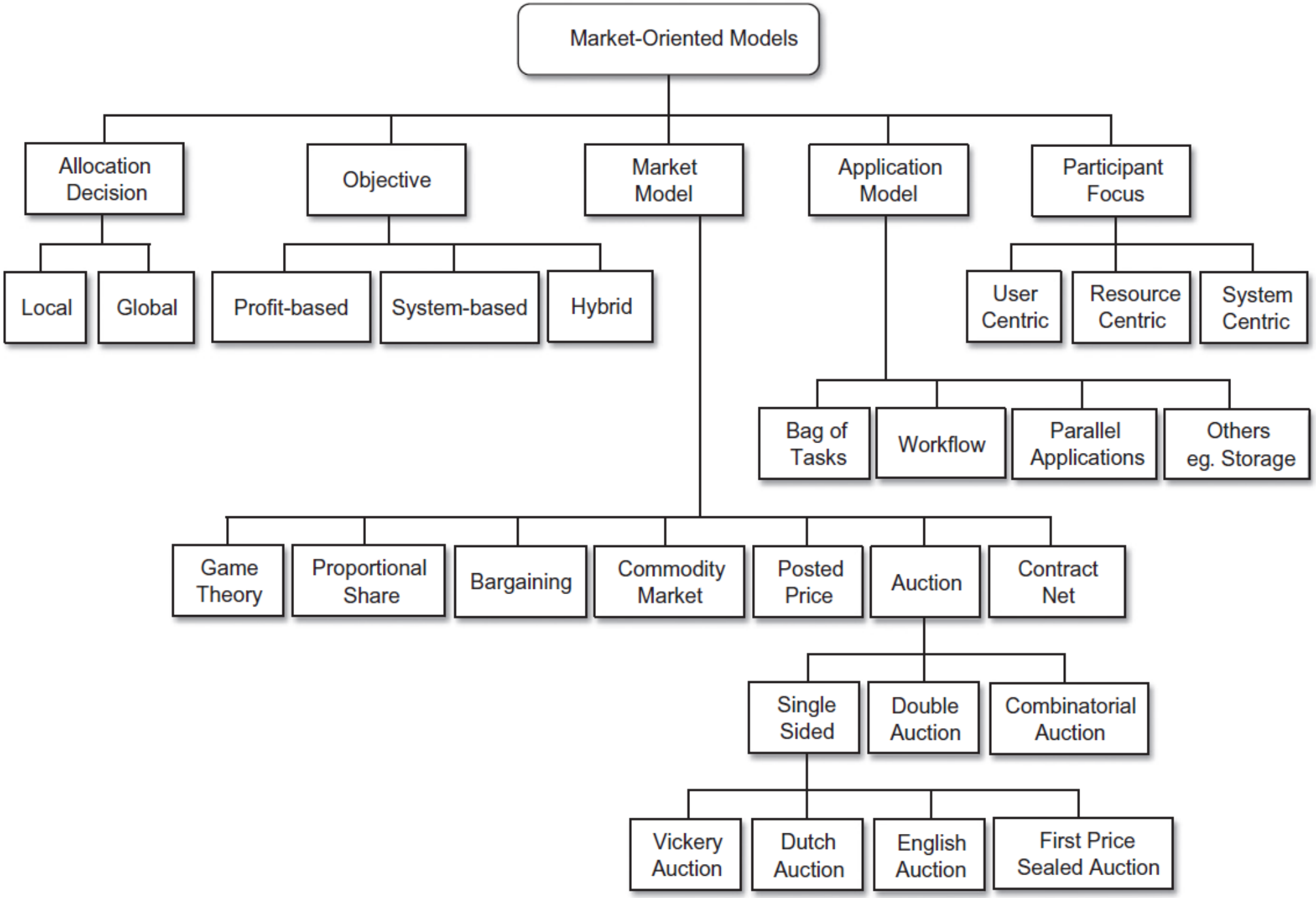
- Short term and long term, revisit at every technology refresh cycle
- Keep asking: “does it make sense for my organization?”
- Might have to adopt some cloud due to partners / vendors etc.
- Test drive / trial
- If no 24/7 data centers in house, you can shutdown at nights, weekends
- Collocation as first step towards moving to cloud completely
- Remote access, IP KVM, console access, VMWare, Hyper-V support

Further steps

- Peaks and Valleys – auto scale, blade servers can run full load all the time
- Energy efficiency: power supplies are most efficient with in the middle of their load range, green devices, green incentives and initiatives
- Starter kit, migration utilities, eat learning curve / training / getting used to
- Virtualization as a stepping stone towards cloud, free trial software, old / retired servers
- Planning for success, cloud just an aid, well understood application
- Websites for contests, globally accessed CRM

Advancements

- Energy efficiency: single datacenter consumes as much energy as 25000 households, doubles every 5 years, 42% of Amazon's budget
- The Green Grid: increase energy efficiency, reduce carbon footprint, $23 (2010) -> $16 (2020) billion, 28% less emission
- Green scheduling
- Market oriented, broker, bid / negotiate
  - EC2 spot instances
  - SpotCloud
  - AppSpot


Market-Oriented Models
Allocation Decision
Objective
Market Model
Application Model
Participant Focus
Local
Global
Profit-based
System-based
Hybrid
User Centric
Resource Centric
System Centric
Bag of Tasks
Workflow
Parallel Applications
Others eg. Storage
Game Theory
Proportional Share
Bargaining
Commodity Market
Posted Price
Auction
Contract Net
Single Sided
Double Auction
Combinatorial Auction
Vickery Auction
Dutch Auction
English Auction
First Price Sealed Auction

Advancements (contd)

- Federated clouds
  - ad hoc
  - agreement between cloud providers to use each other's services
  - Proprietary interfaces
- InterCloud
  - Standards
  - Interoperability
- Third party cloud services: MetaCDN (over storage clouds), SpotCloud

Future

- Standardization
- BOINC, Blockchain, leverage distributed available compute
- Specialized clouds
- Google docs, Microsoft office live, quora
- Mobile cloud: heavy lifting by back end cloud, maps
- Multi programming -> multiple VMs
- Cloud extensions: AutoCAD -> Building Information Modeling (BIM)
- Cloud Descriptor Language (CDL): capturing resource requirements
- Cost, control, bottom line – profit
- Full cycle; mainframe – dumb terminals, cloud – thin clients
- Ubiquitous computing, abstraction -> focus on business rather than tools

Thank You

&

Good Luck