CLOUD COMPUTING

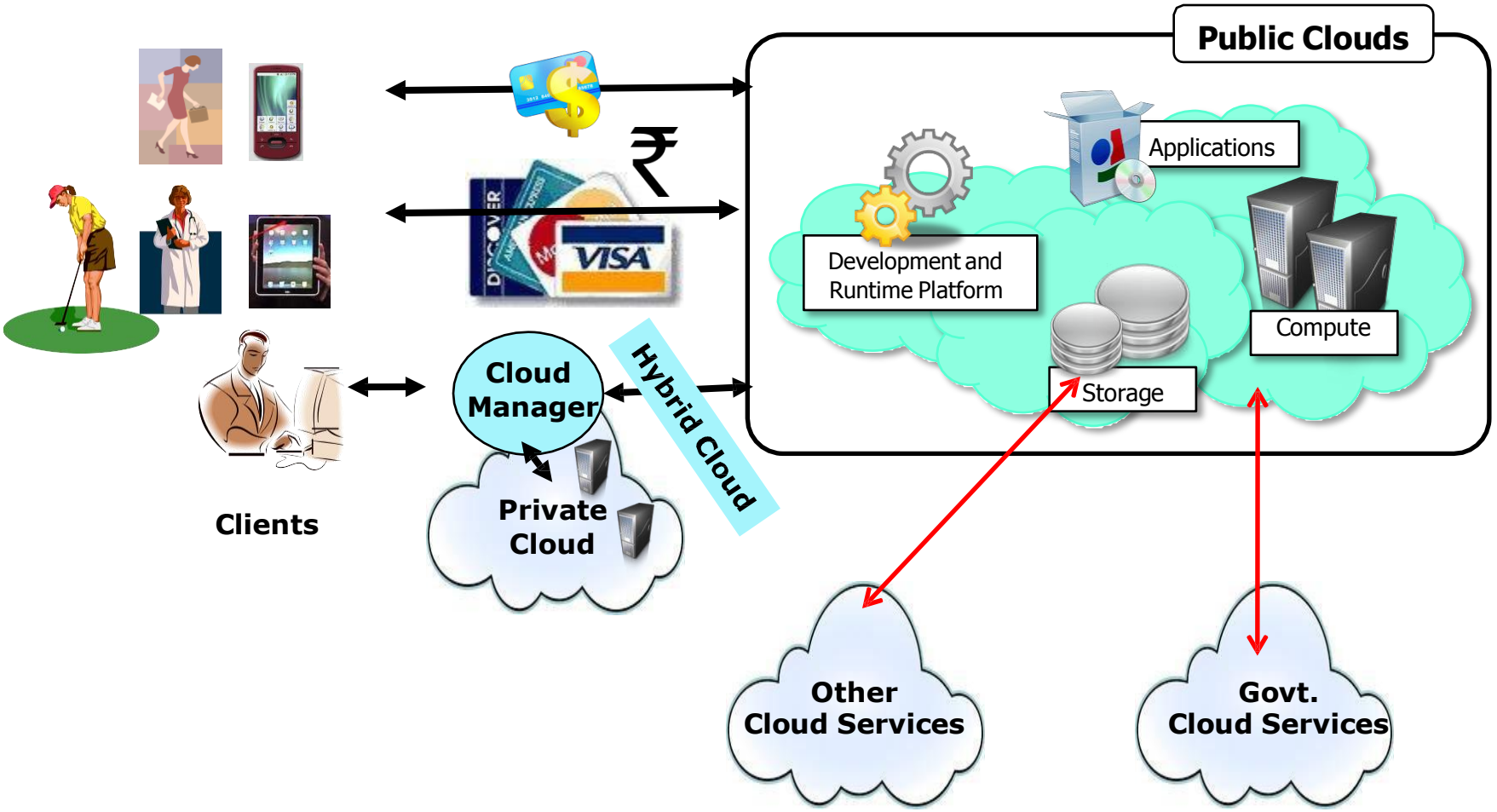
Prepared by,

Ms. Ch Sri Vidya, Assistant Professor
Ms. A Jayanthi, Assistant Professor
Ms. S Swarajya Laxmi, Assistant Professor
Ms. Tejaswi , Assistant Professor

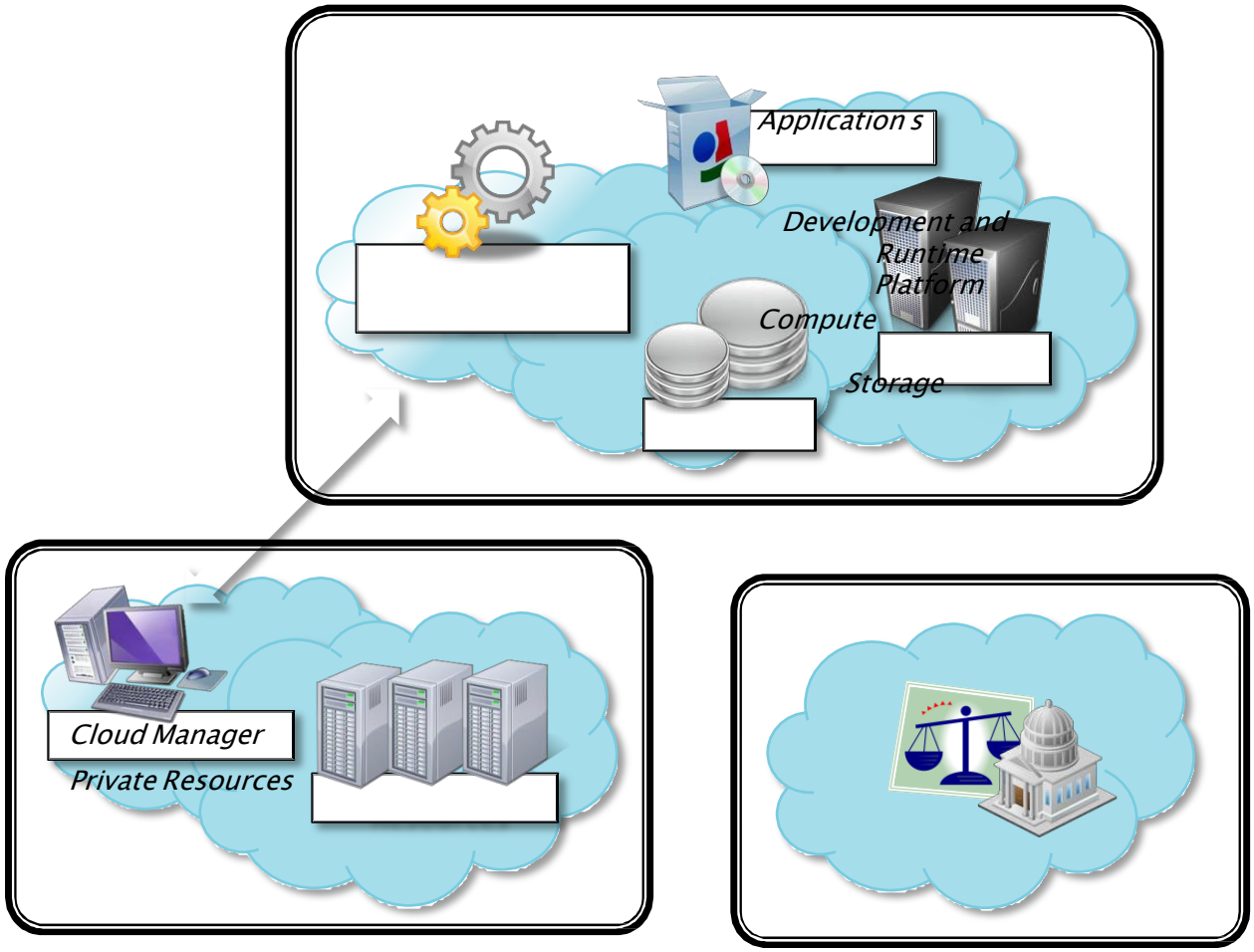
UNIT-I

System modelling, Clustering, Virtualisation

Subscription-Oriented Cloud Services:
X{compute, apps, data, ..} as a Service (..aaS)



Subscription-Oriented Cloud Services:
X{compute, apps, data, ..} as a Service (..aaS)



Cloud Deployment Models

Public/Internet Clouds

** 3rd party, multi-tenant Cloud infrastructure & services:*

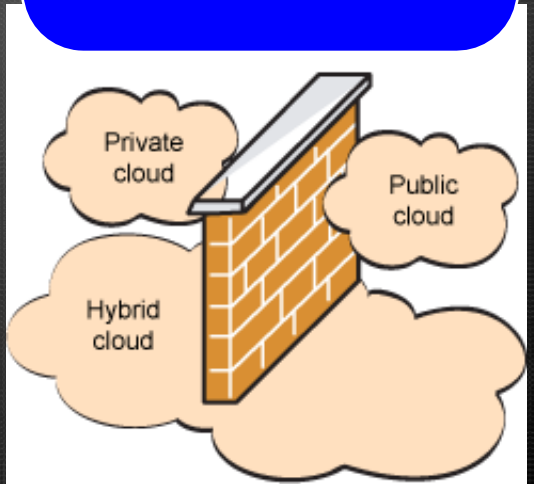
** available on subscription basis to all.*

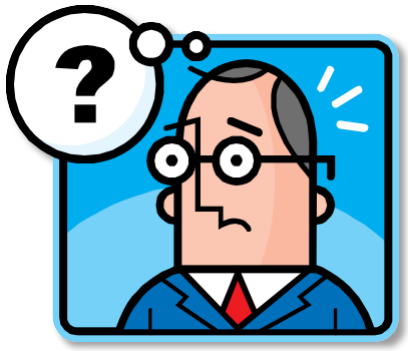
Private/Enterprise Clouds

** A public Cloud model within* **a company's** *own Data Center / infrastructure for internal and/or partners use.*

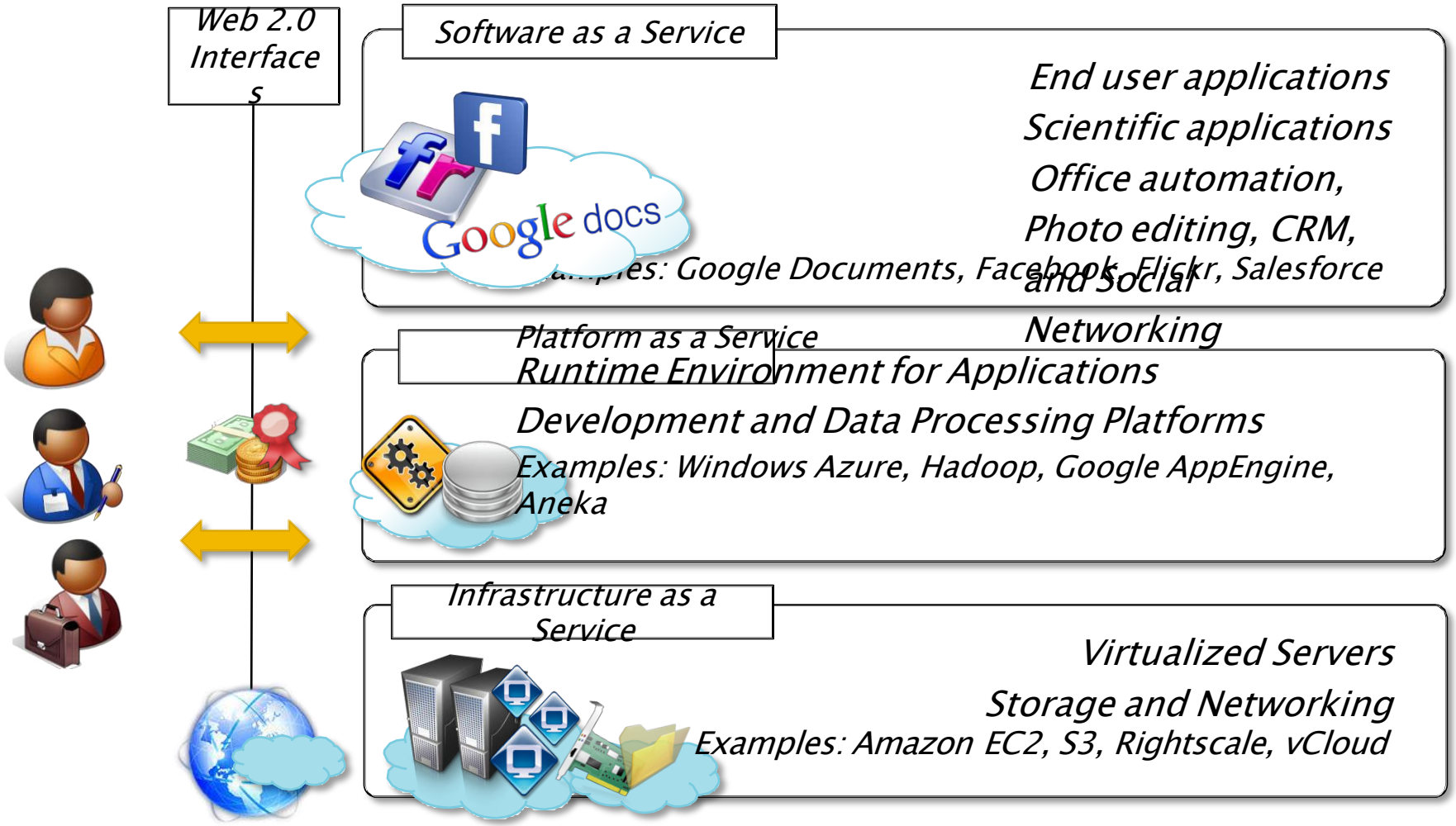
Hybrid/Inter Clouds

** Mixed usage of private and public Clouds: Leasing public cloud services when private cloud capacity is insufficient*

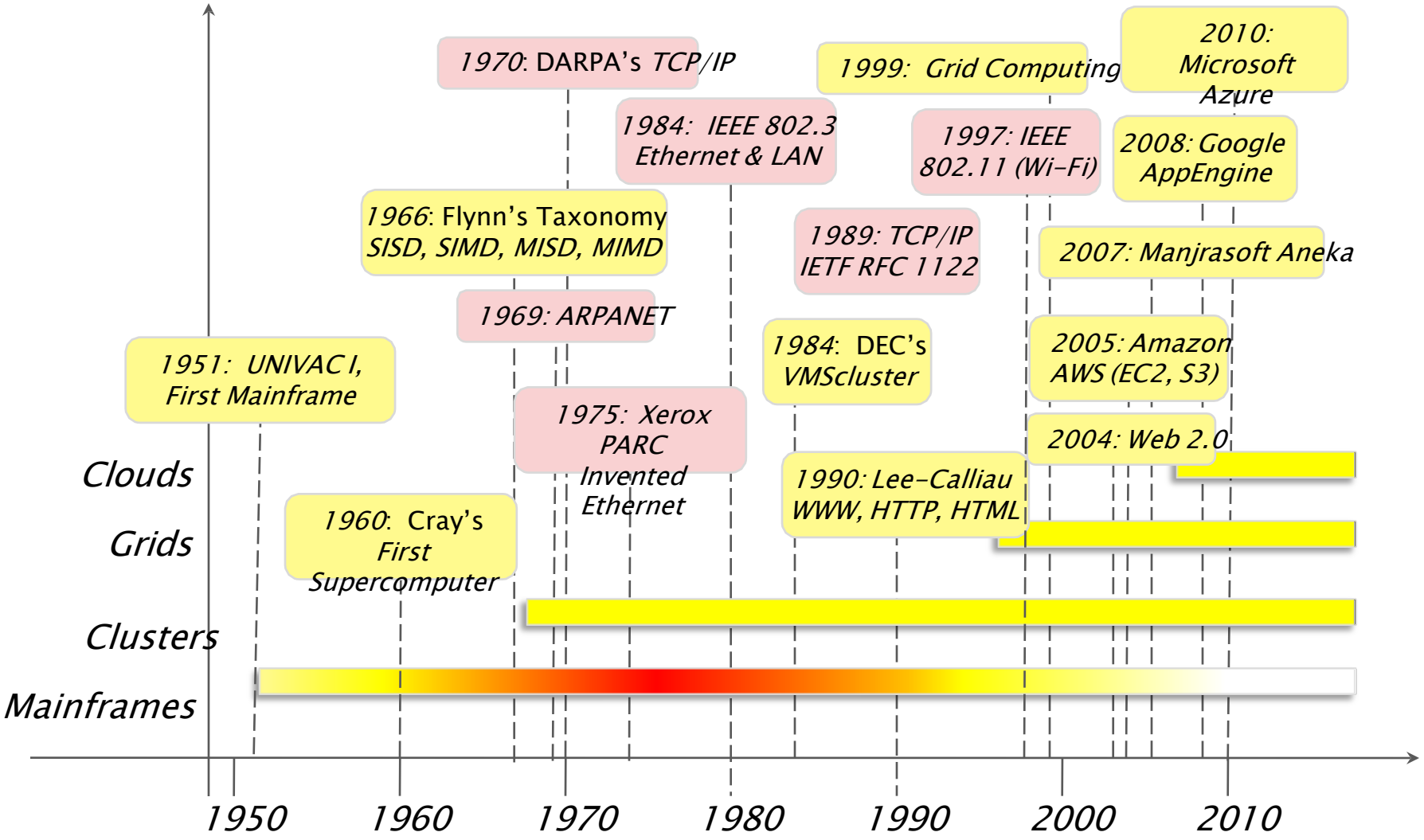




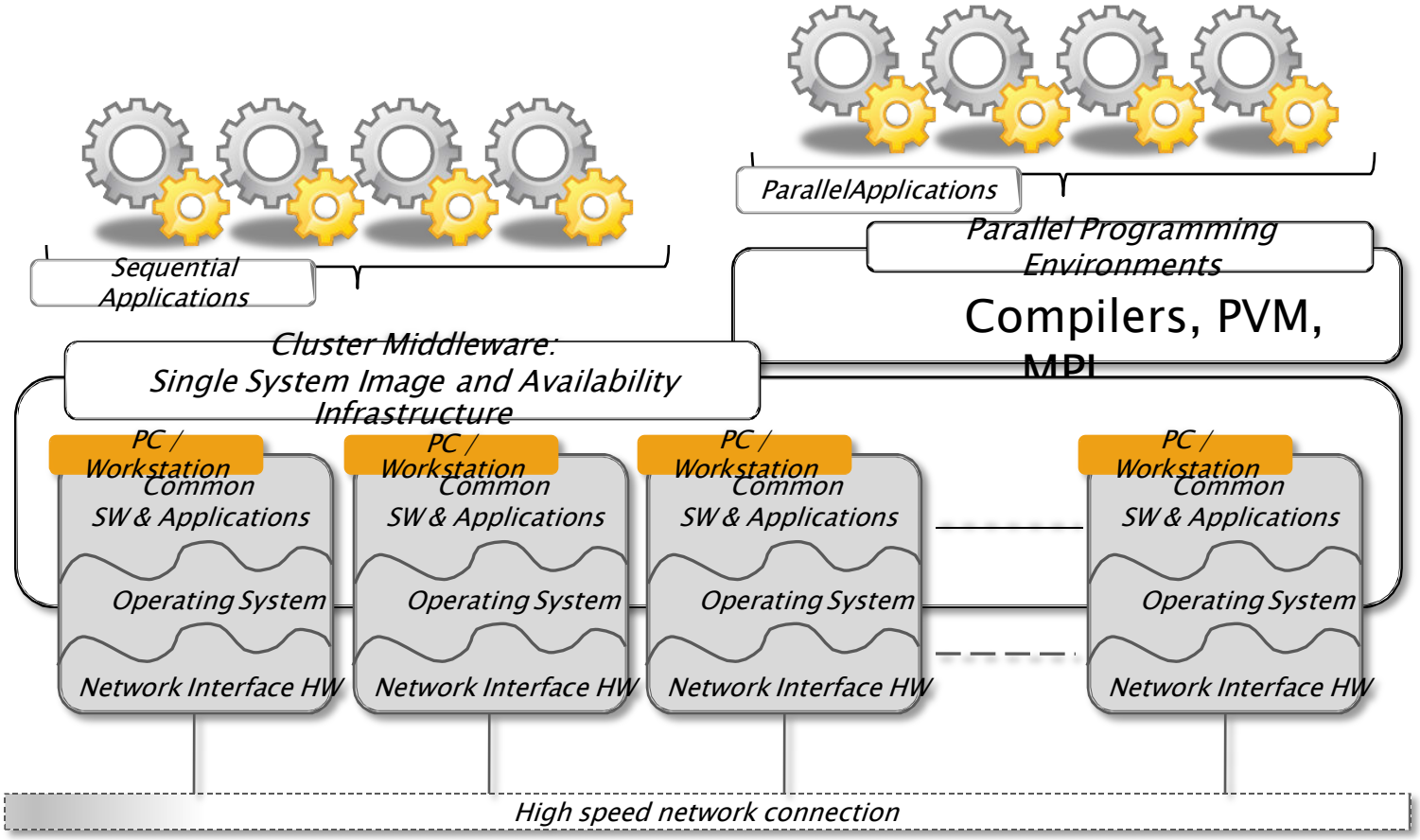
CLOUD COMPUTING DEPLOYMENT



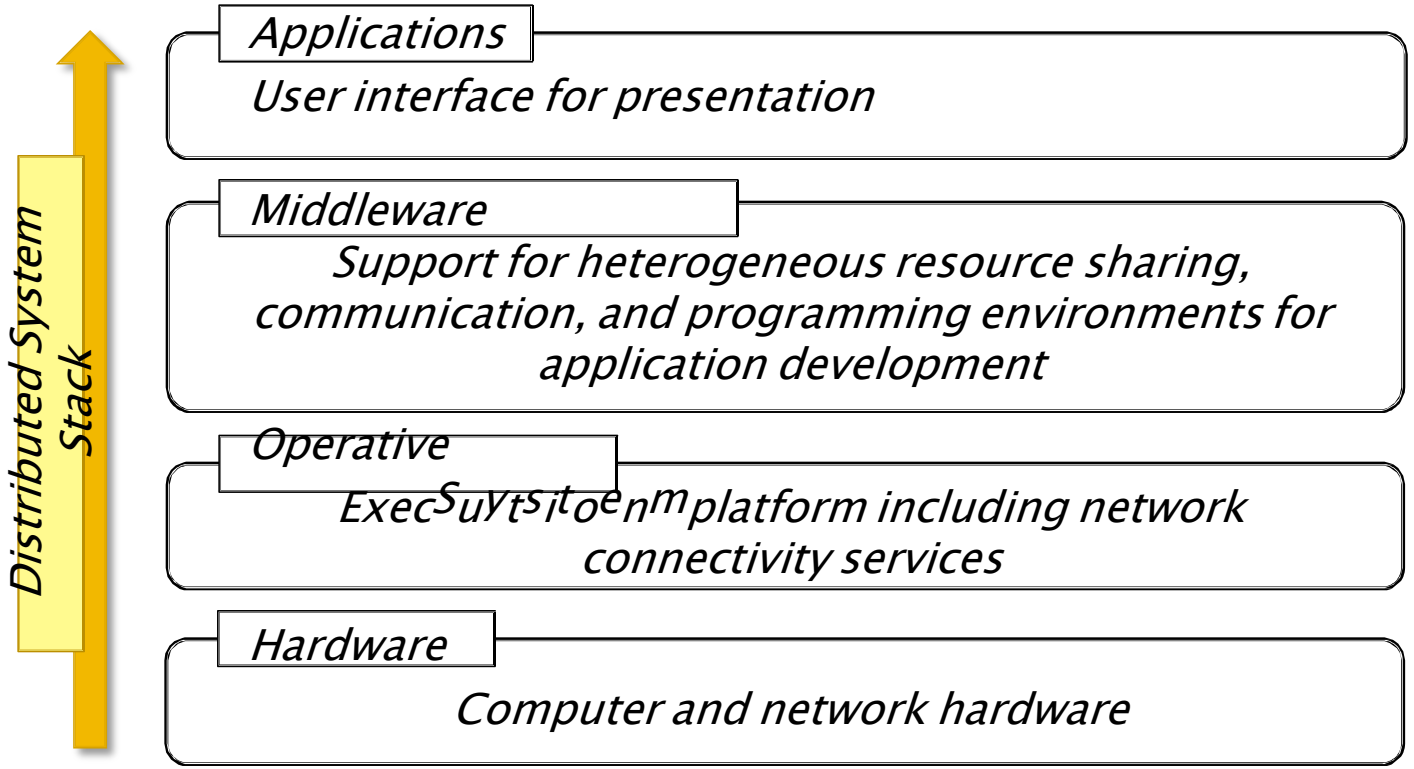
CLOUD COMPUTING DEPLOYMENT

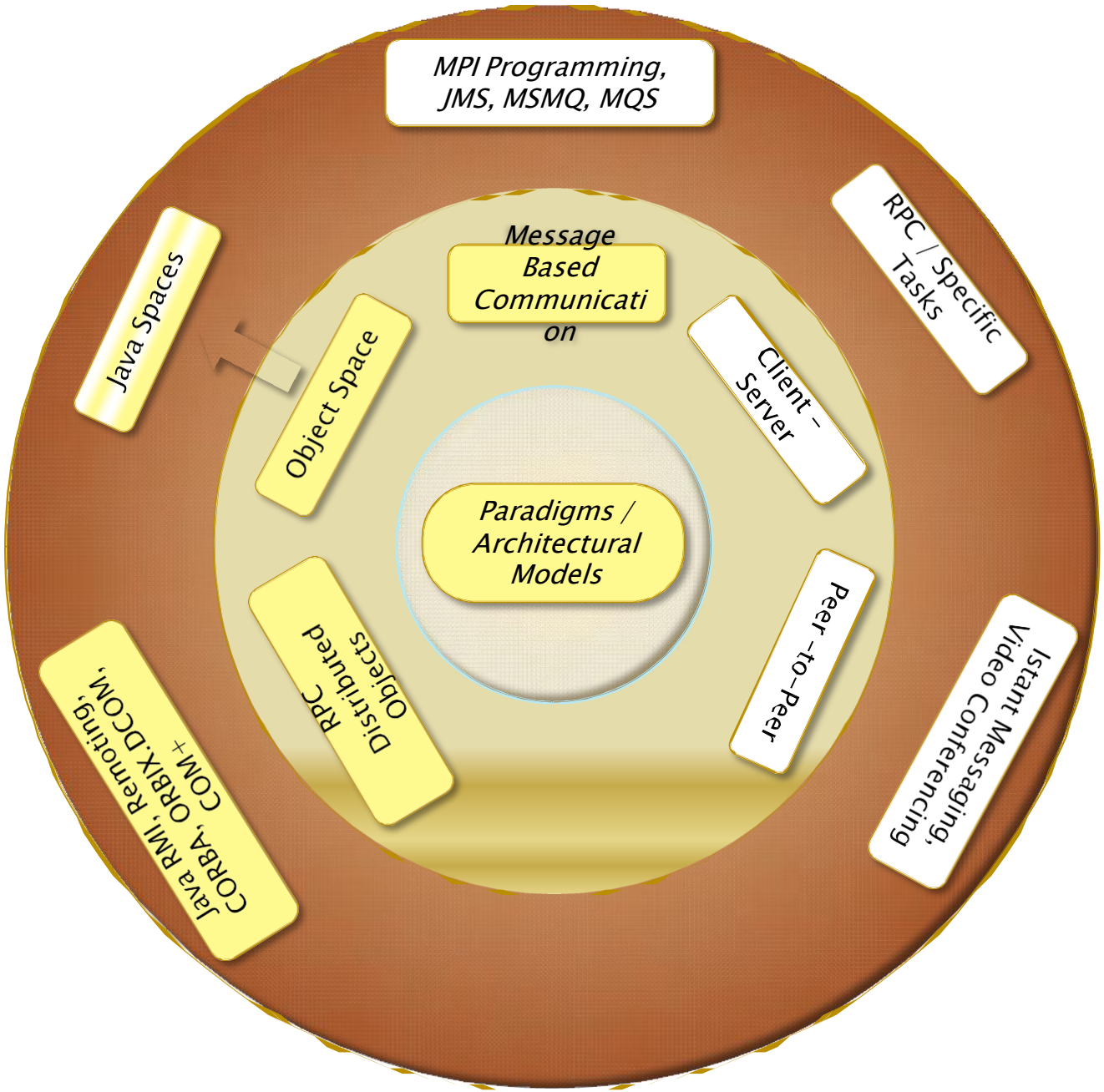


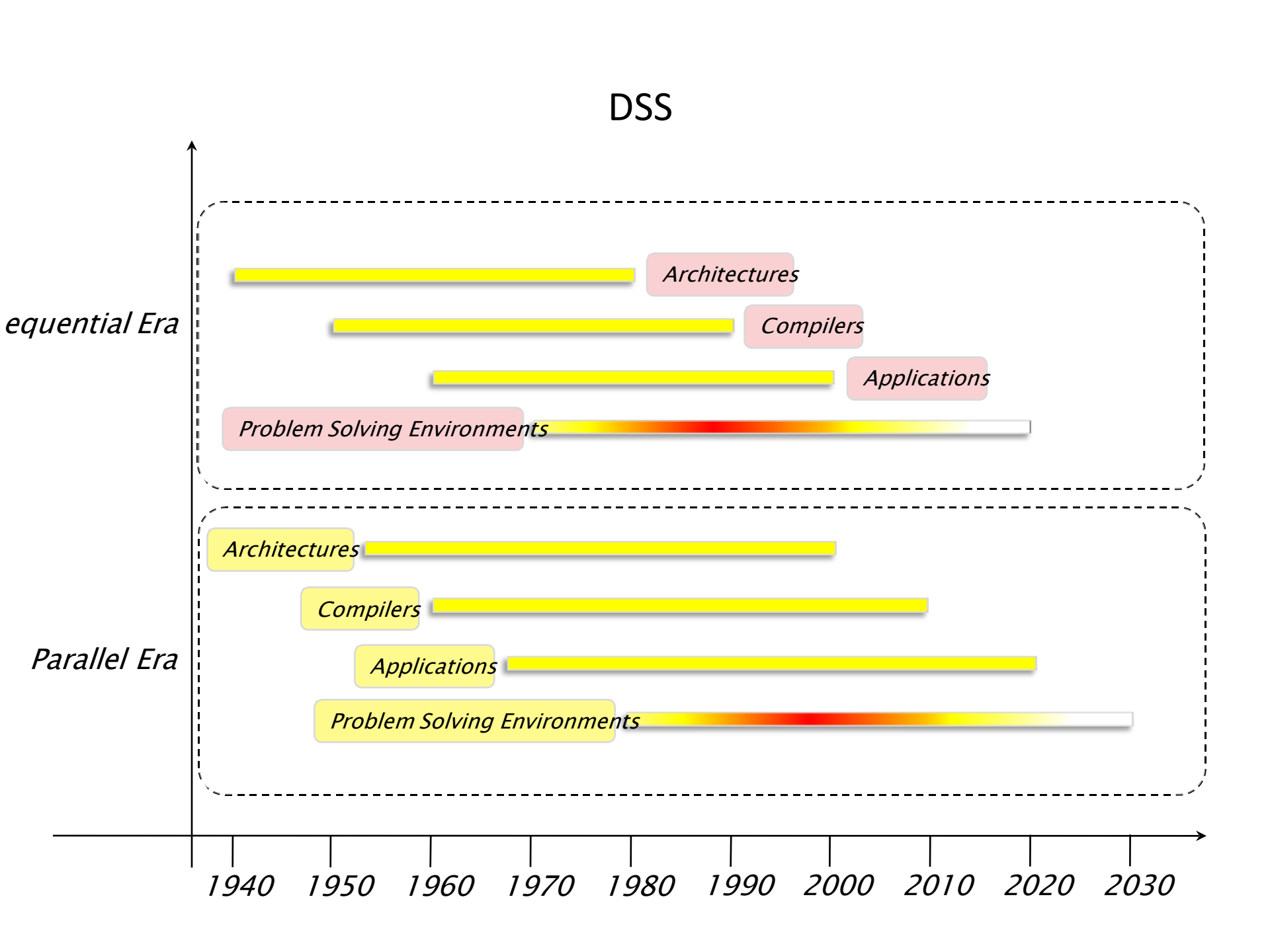
DSS


Sequential Applications
ParallelApplications
Parallel Programming Environments
Compilers, PVM, MPI
Cluster Middleware:
Single System Image and Availability Infrastructure
PC / Workstation
Common SW & Applications
Operating System
Network Interface HW
PC / Workstation
Common SW & Applications
Operating System
Network Interface HW
PC / Workstation
Common SW & Applications
Operating System
Network Interface HW
PC / Workstation
Common SW & Applications
Operating System
Network Interface HW
High speed network connection

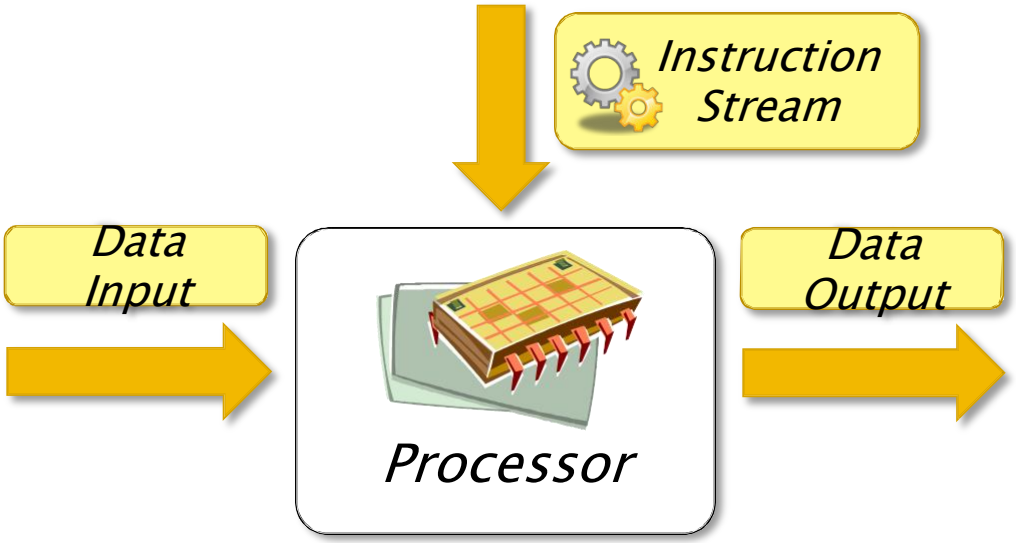
DSS

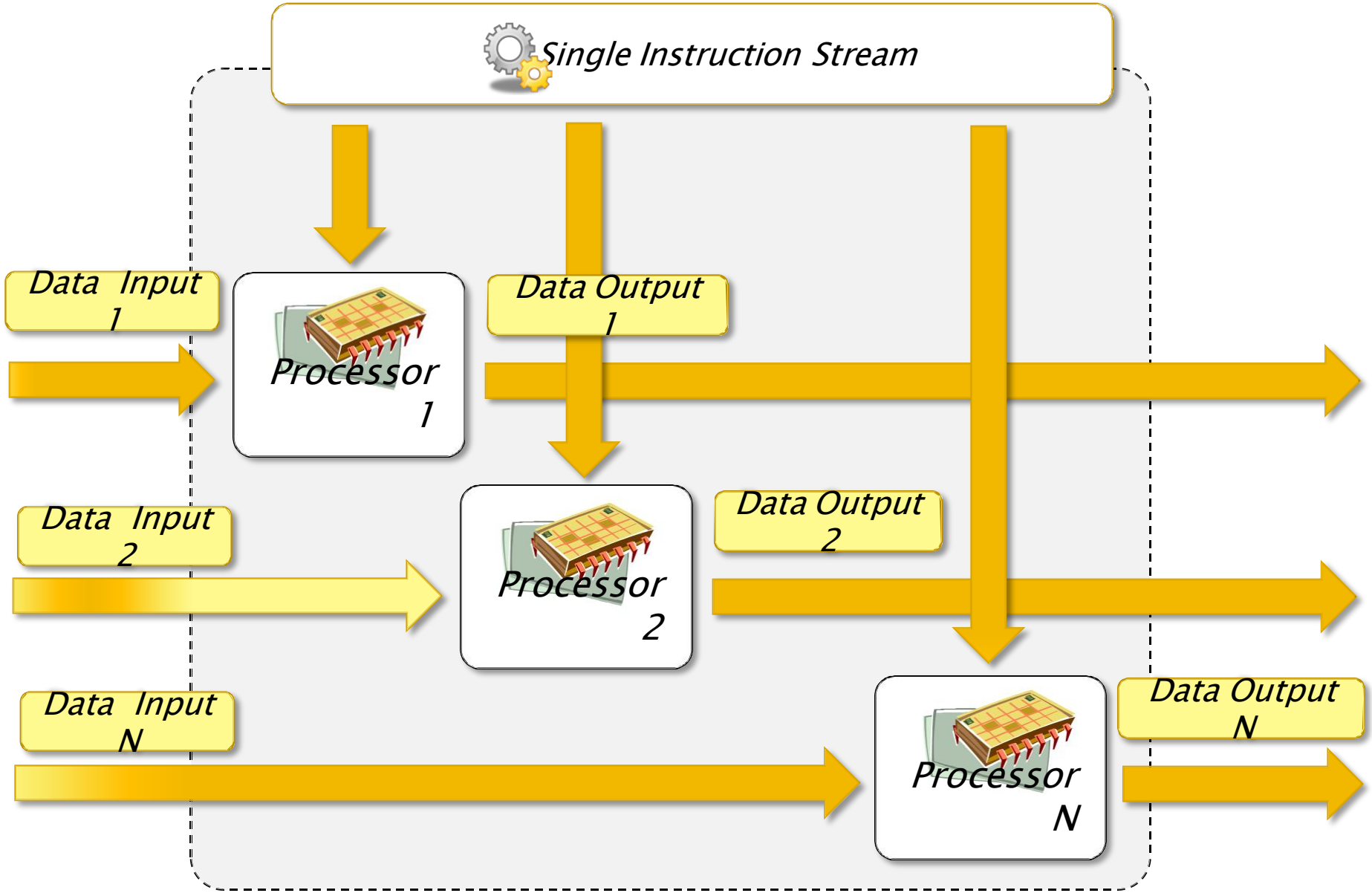

Distributed System Stack
Applications
User interface for presentation
Middleware
Support for heterogeneous resource sharing, communication, and programming environments for application development
Operative System
Execution platform including network connectivity services
Hardware
Computer and network hardware

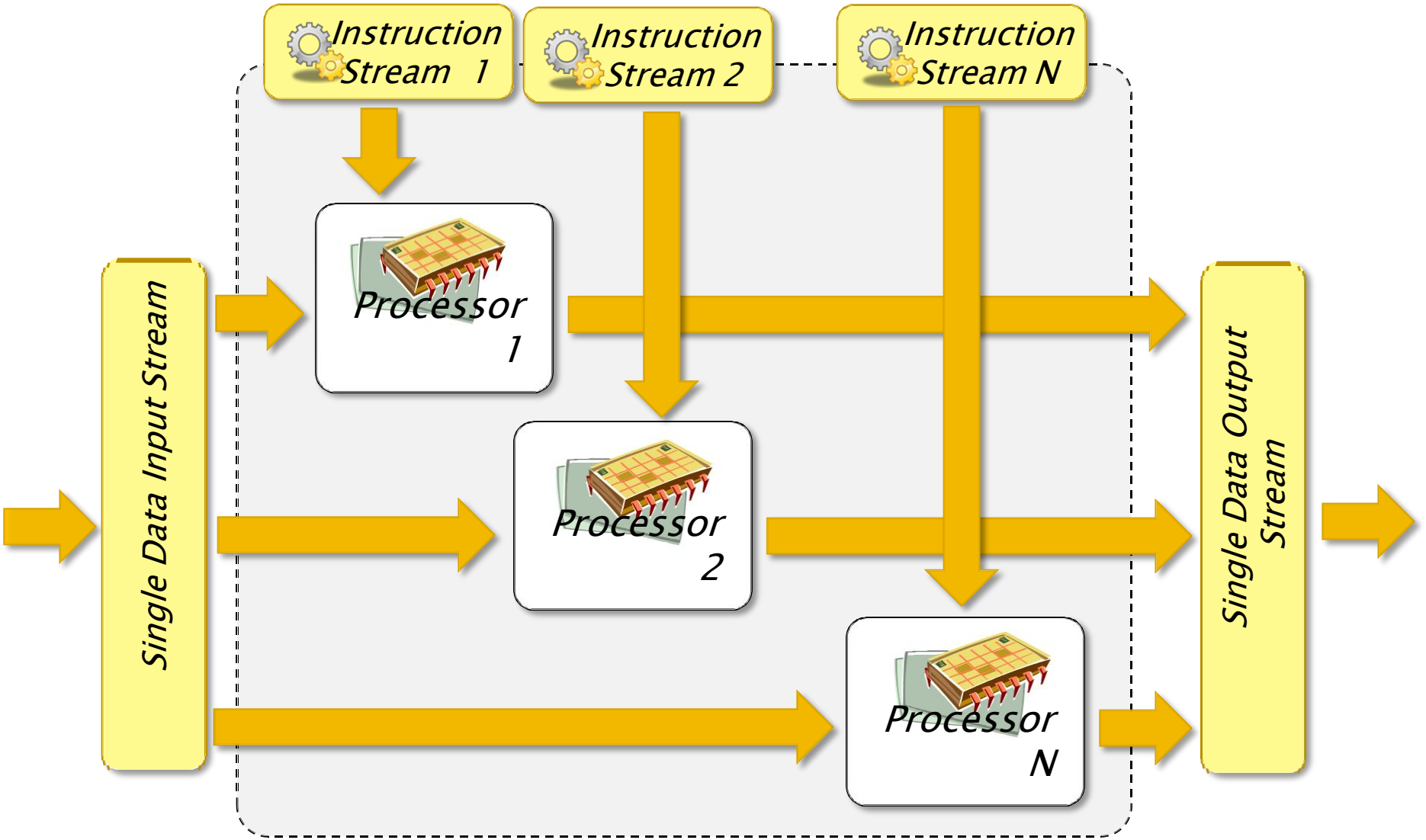

MPI Programming, JMS, MSMQ, MQS
RPC / Specific Tasks
Istant Messaging, Video Conferencing
Peer – to–Peer
Client – Server
Message Based Communication
Object Space
Java Spaces
Distributed Objects RPC
Java RMI, Remoting, CORBA, ORBIX.DCOM, COM+
Paradigms / Architectural Models

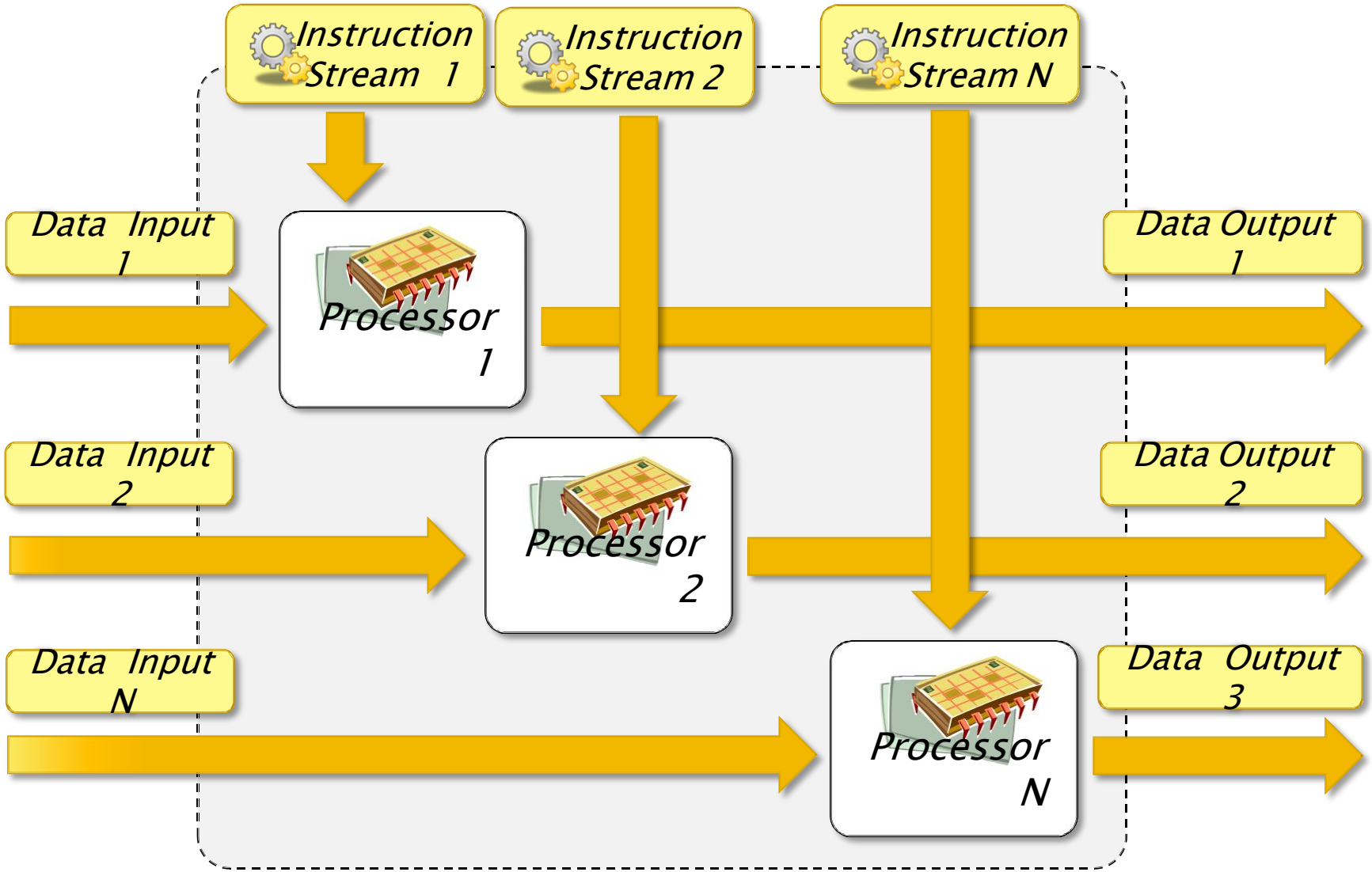

DSS
equential Era
Architectures
Compilers
Applications
Problem Solving Environments
Parallel Era
Architectures
Compilers
Applications
Problem Solving Environments
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030

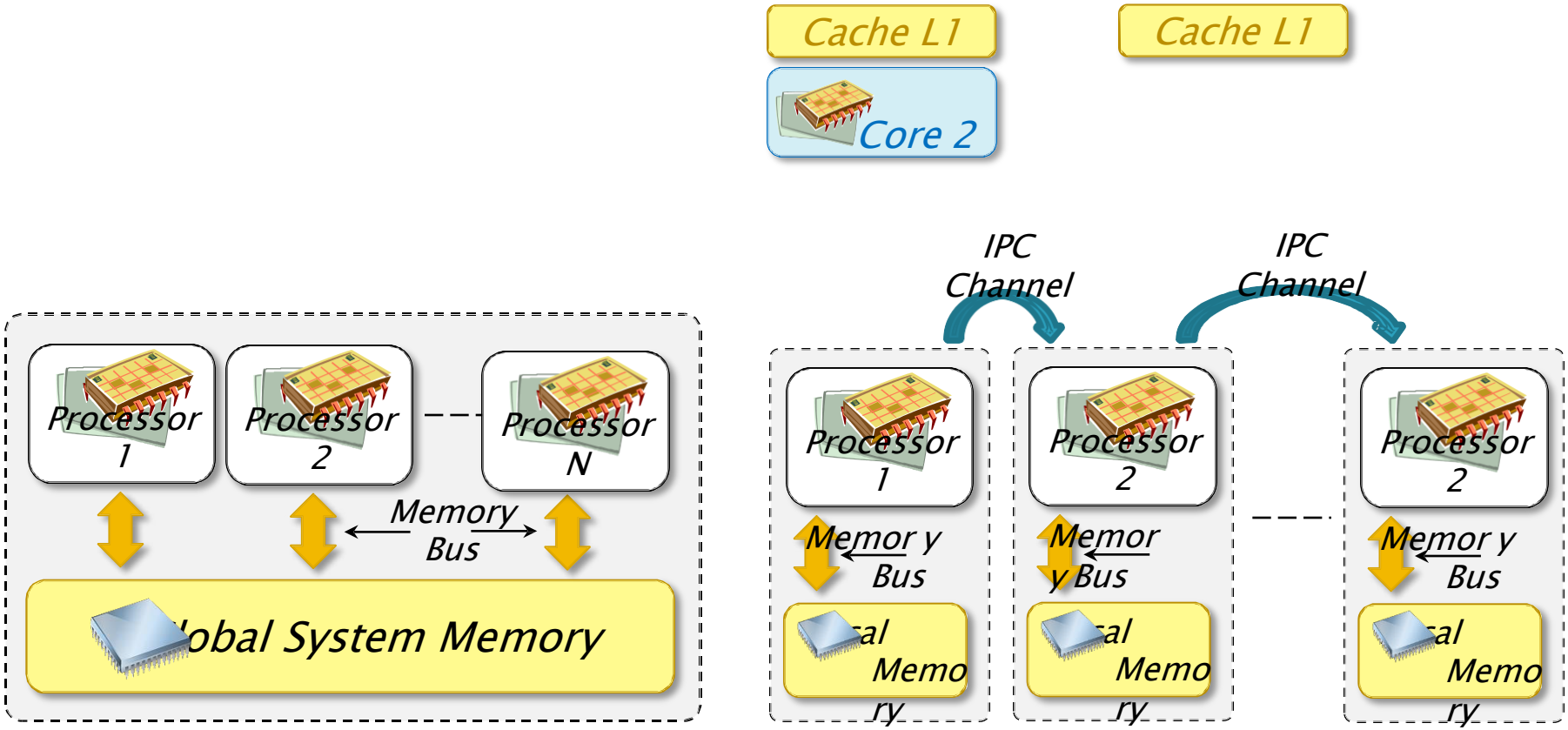
DSS

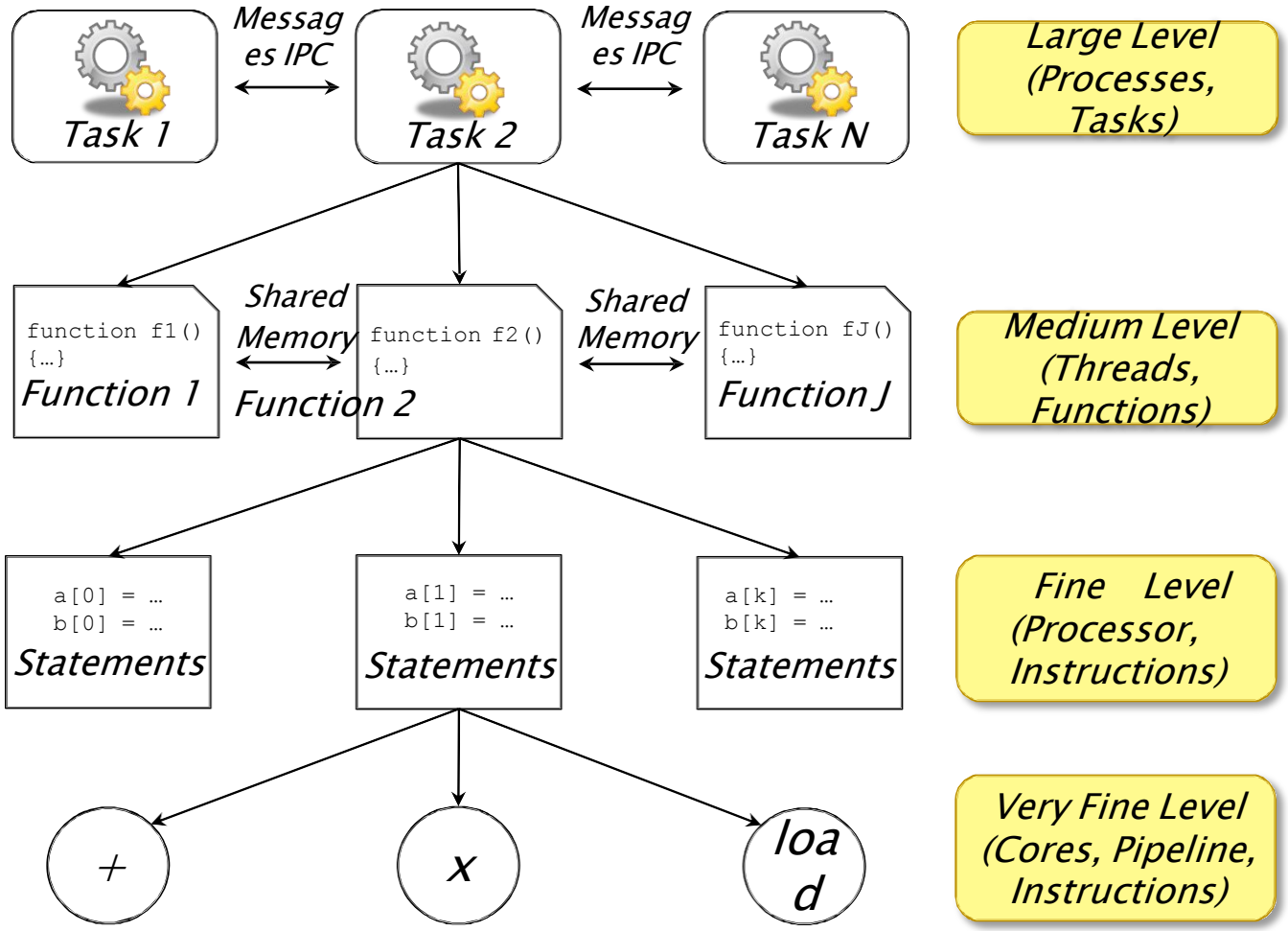


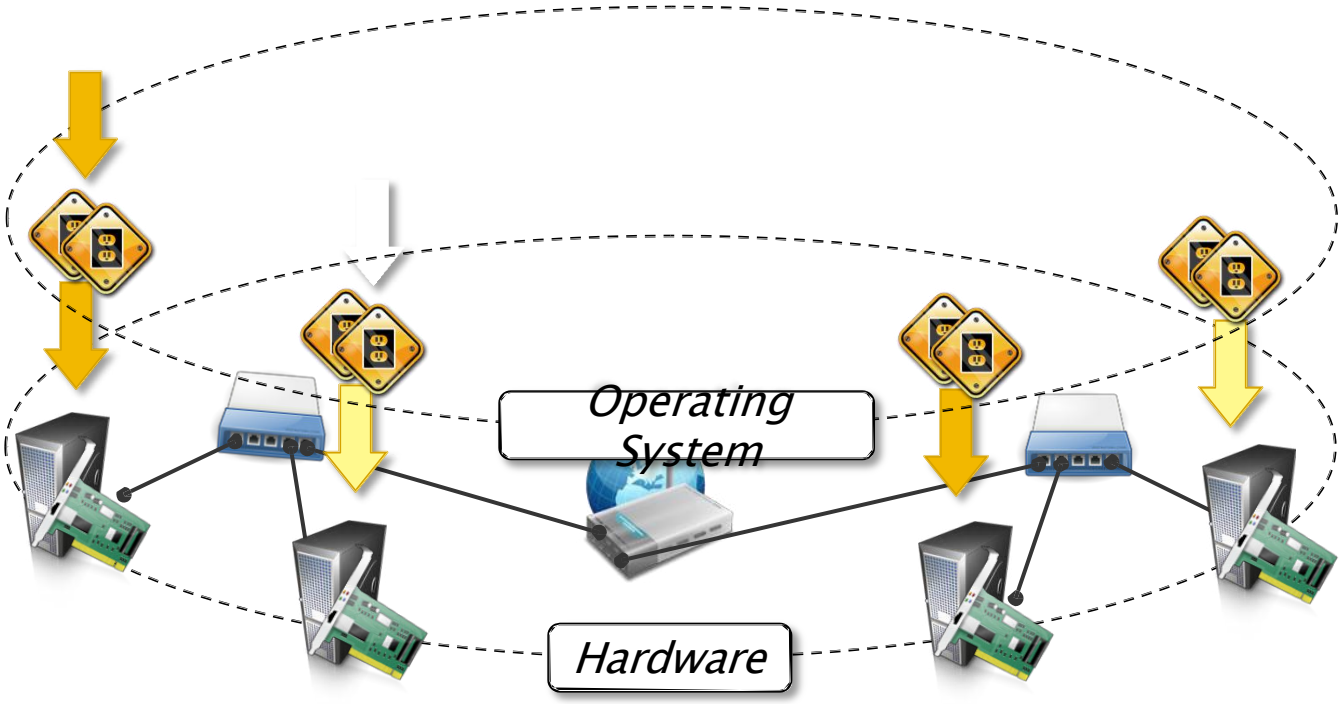

Single Instruction Stream
Data Input 1
Processor 1
Data Output 1
Data Input 2
Processor 2
Data Output 2
Data Input N
Processor N
Data Output N

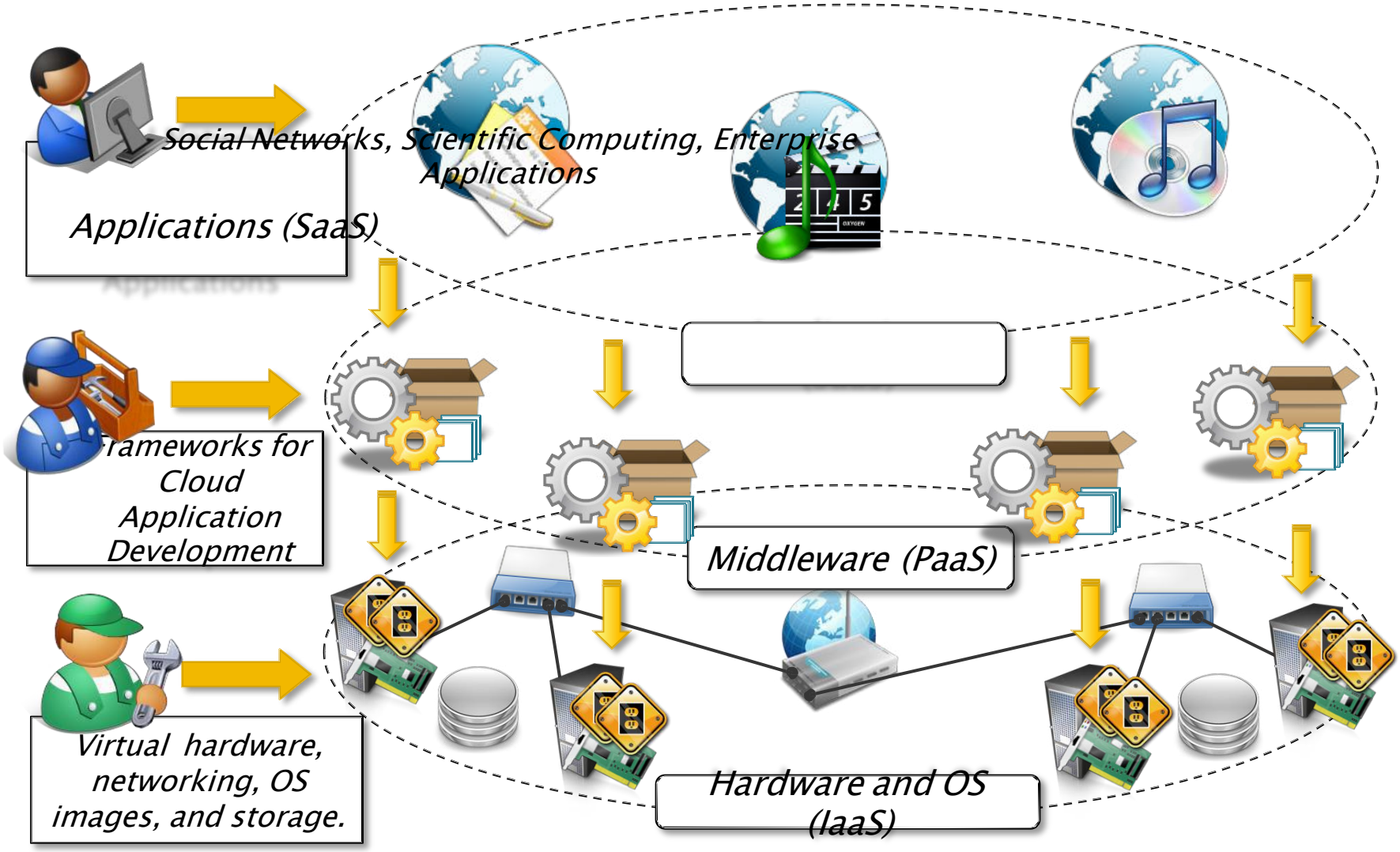

Instruction Stream 1
Instruction Stream 2
Instruction Stream N
Single Data Input Stream
Processor 1
Processor 2
Processor N
Single Data Output Stream


Instruction Stream 1
Instruction Stream 2
Instruction Stream N
Data Input 1
Data Input 2
Data Input N
Processor 1
Processor 2
Processor N
Data Output 1
Data Output 2
Data Output 3

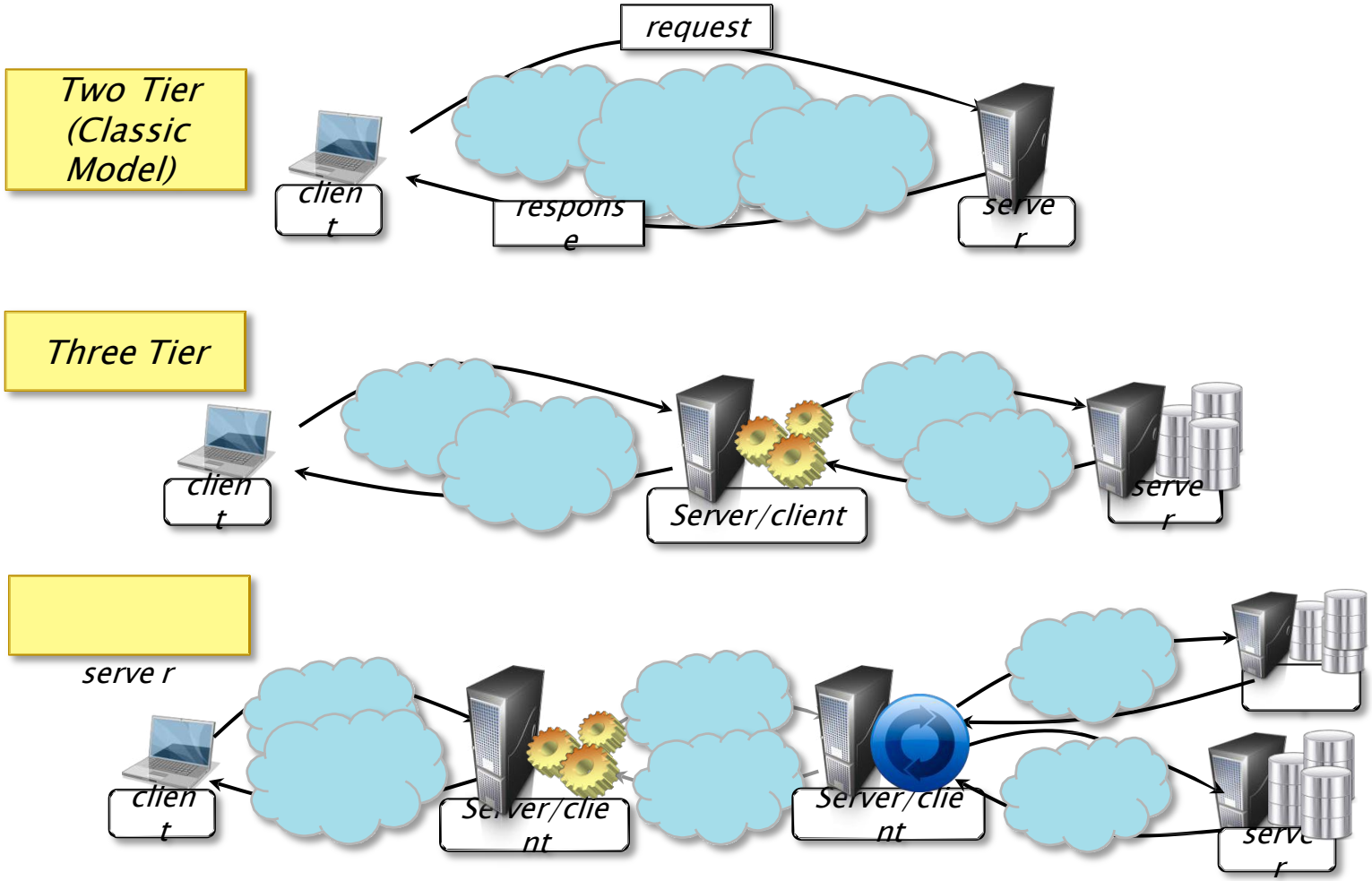

Cache L1
Cache L1
Core 2
IPC Channel
IPC Channel
Processor 1
Processor 2
Processor N
Memory Bus
obal System Memory
Processor 1
Processor 2
Processor 2
Memor y Bus
Memor y Bus
Memor y Bus
al Memo ry
al Memo ry
al Memo ry


Task 1
Messag es IPC
Task 2
Messag es IPC
Task N
Large Level (Processes, Tasks)
function f1() {…}
Function 1
Shared Memory
function f2() {…}
Function 2
Shared Memory
function fJ() {…}
Function J
Medium Level (Threads, Functions)
a[0] = …
b[0] = …
Statements
a[1] = …
b[1] = …
Statements
a[k] = …
b[k] = …
Statements
Fine Level (Processor, Instructions)
+
x
load
Very Fine Level (Cores, Pipeline, Instructions)

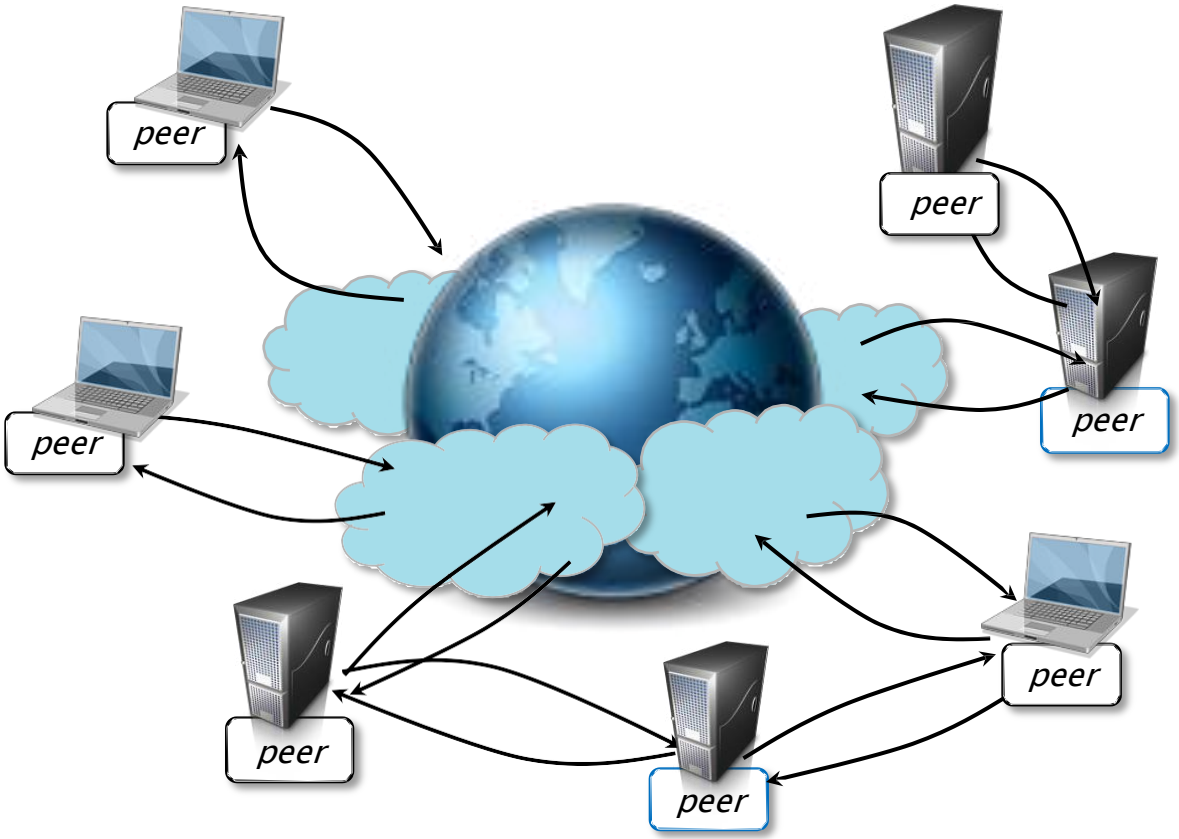

Operating System
Hardware


Social Networks, Scientific Computing, Enterprise Applications
Applications (SaaS)
Frameworks for Cloud Application Development
Middleware (PaaS)
Virtual hardware, networking, OS images, and storage.
Hardware and OS (IaaS)

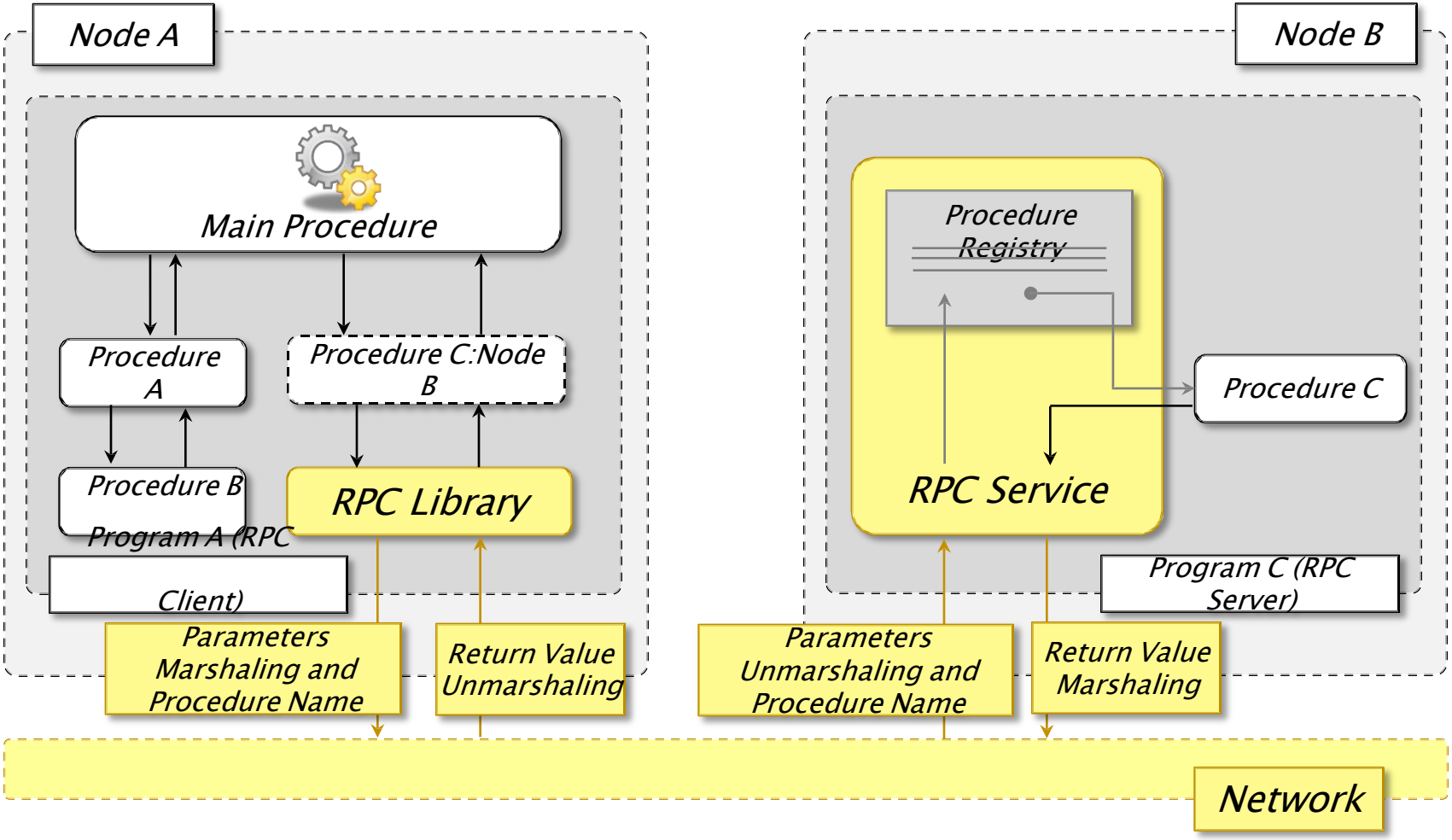
Client-server



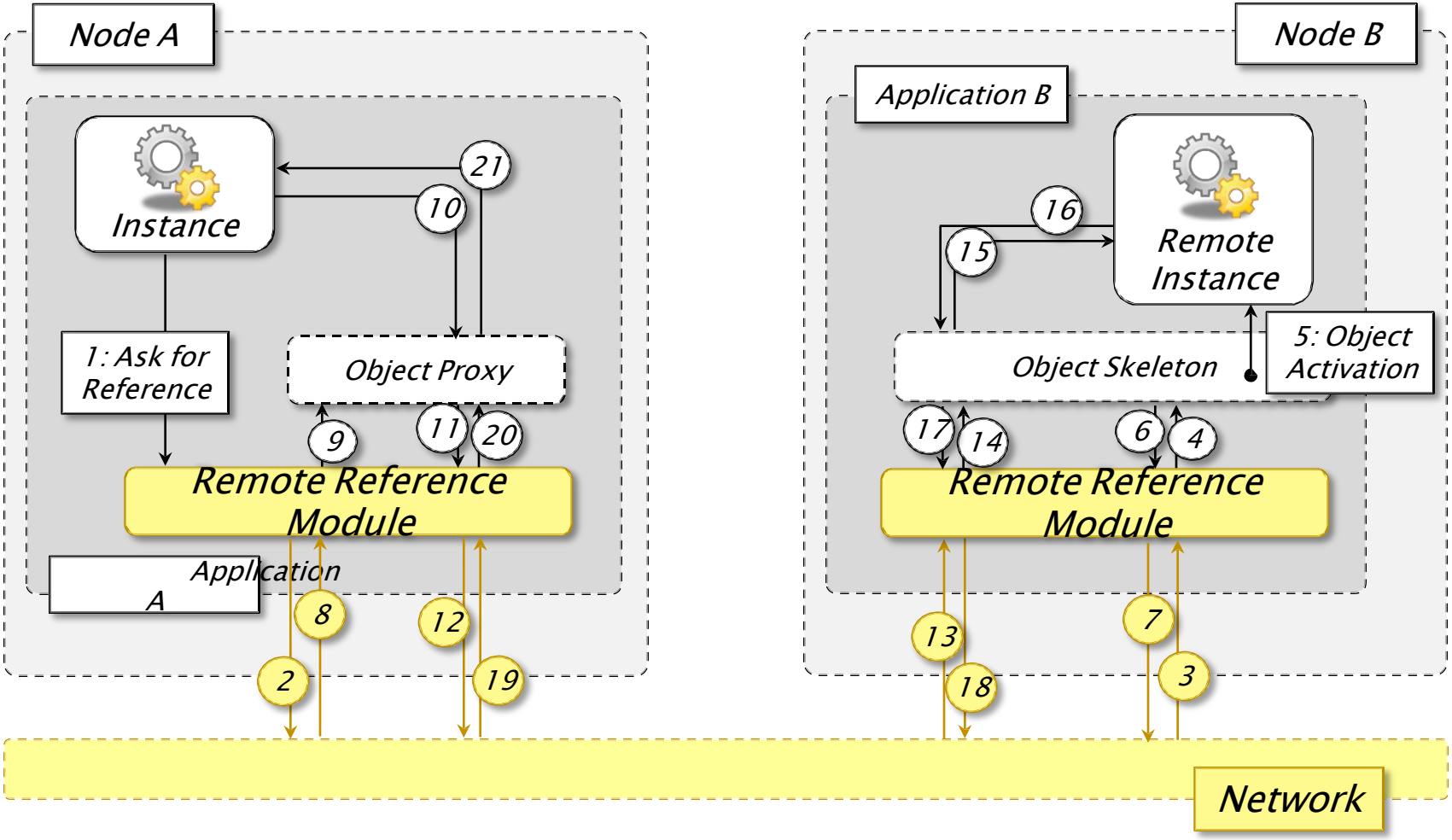
P2P

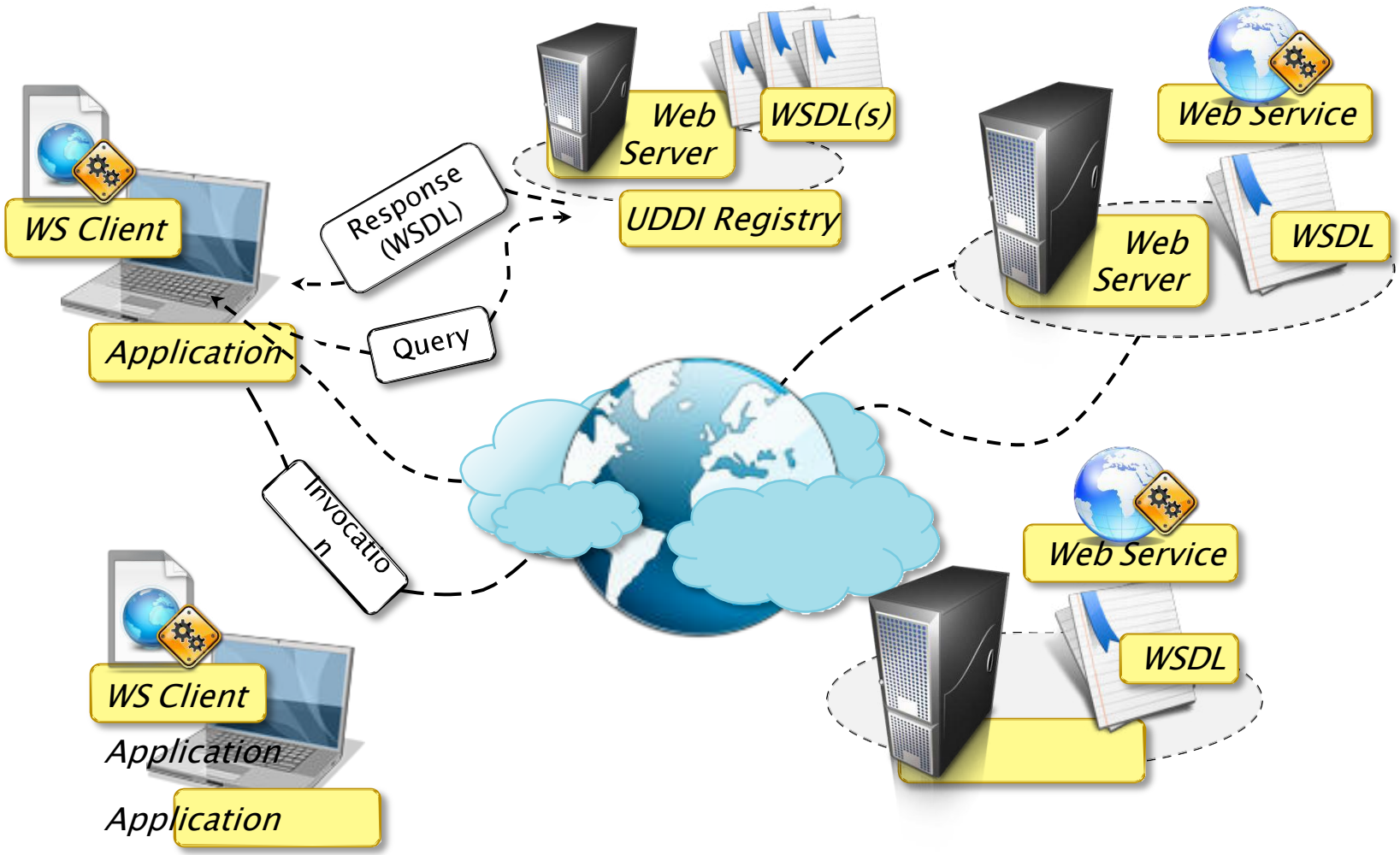


RPC

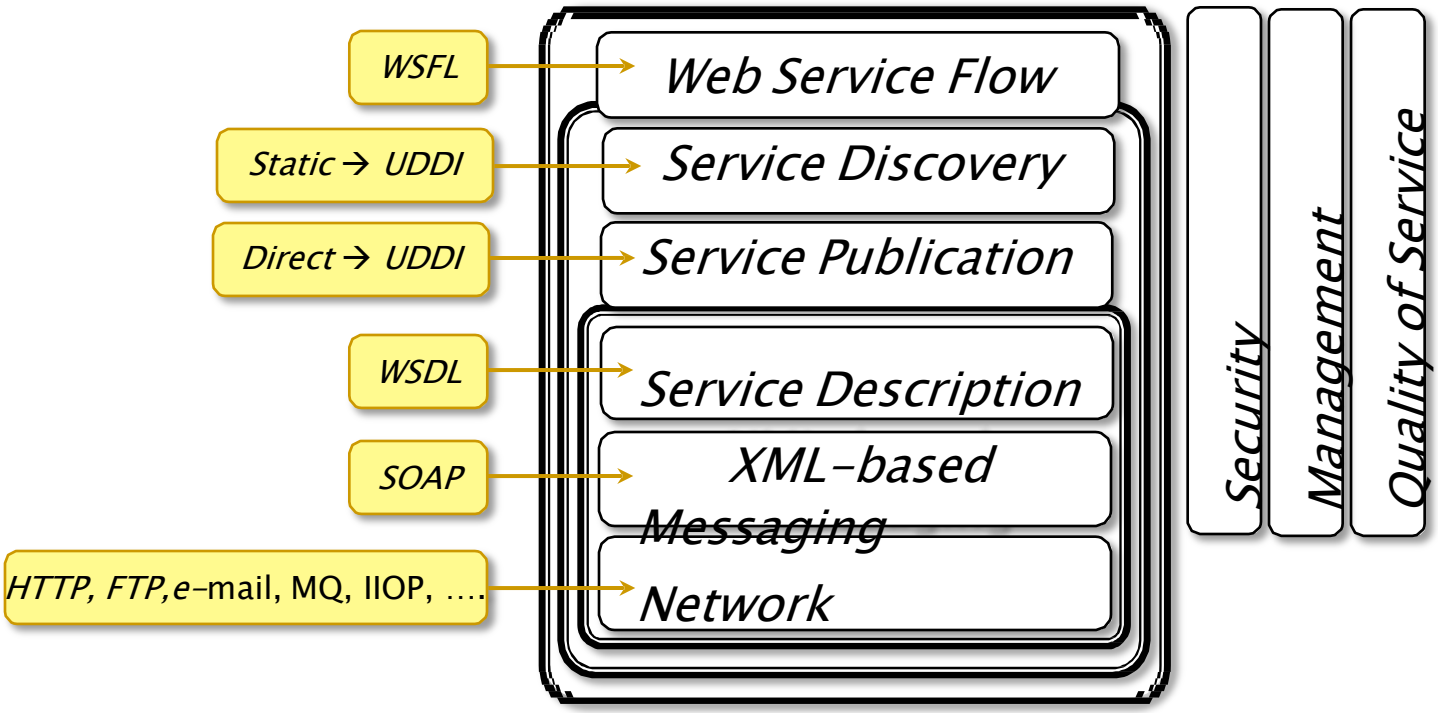

Node A
Main Procedure
Procedure A
Procedure C:Node B
Procedure B
RPC Library
Program A (RPC Client)
Parameters Marshaling and Procedure Name
Return Value Unmarshaling
Node B
Procedure Registry
Procedure C
RPC Service
Program C (RPC Server)
Parameters Unmarshaling and Procedure Name
Return Value Marshaling
Network

DOF interaction




WS Client
Application
Response (WSDL)
Query
Invocation
Web Server
WSDL(s)
UDDI Registry
Web Service
Web Server
WSDL
Web Service
WSDL
WS Client
Application
Application

WS technology Stack



SOAP Messages

```
POST /InStock HTTP/1.1 Host: www.stocks.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: <Size>

<?xml version="1.0">

<soap:Envelope xmlns:soap="http//www.w3.org/2001/12/soap-envelope"
 soap:encondingStyle="http//www.w3.org/2001/12/soap-encoding">
 <soap:Header></soap:Header>
 <soap:Body xmlns:m=http://www.stocks.org/stock>
 <m:GetStockPrice>
 <m:StockName>IBM<m:StockName>
 </m:GetStockPrice>
 </soap:Body>
</soap:Envelope>
```

Envelope

Header: Metadata & Assertions

Body: Method Call

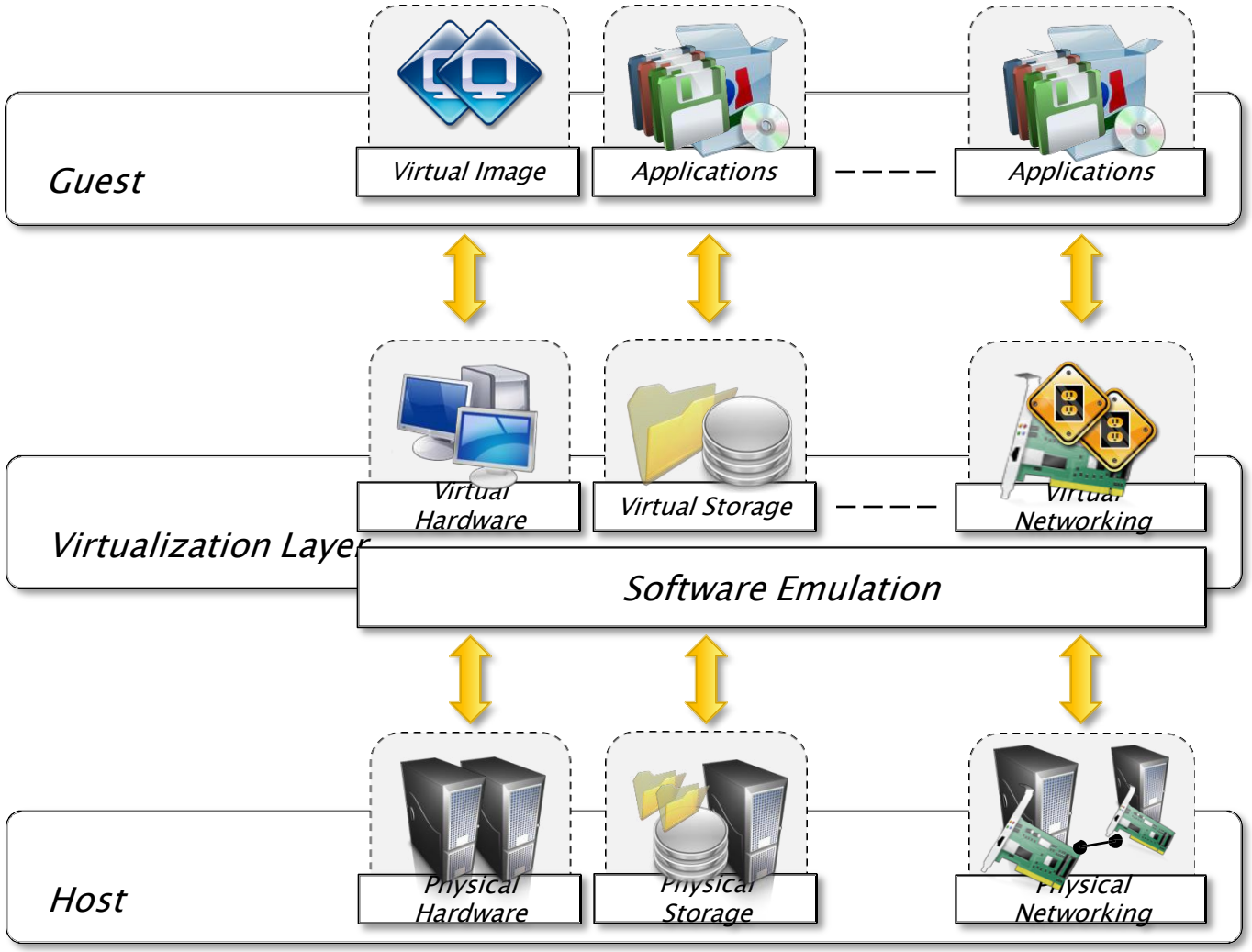
SOAP Messages

```
POST /InStock HTTP/1.1 Host: www.stocks.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: <Size>

<?xml version="1.0">

<soap:Envelope xmlns:soap="http//www.w3.org/2001/12/soap-envelope"
 soap:encondingStyle="http//www.w3.org/2001/12/soap-encoding">
 <soap:Header></soap:Header>
 <soap:Body xmlns:m=http://www.stocks.org/stock>
                          <m:GetStockPriceResponse>
                            <m:Price>34.5<m:Price>
 </m:GetStockPriceResponse>
 </soap:Body>
</soap:Envelope>
```

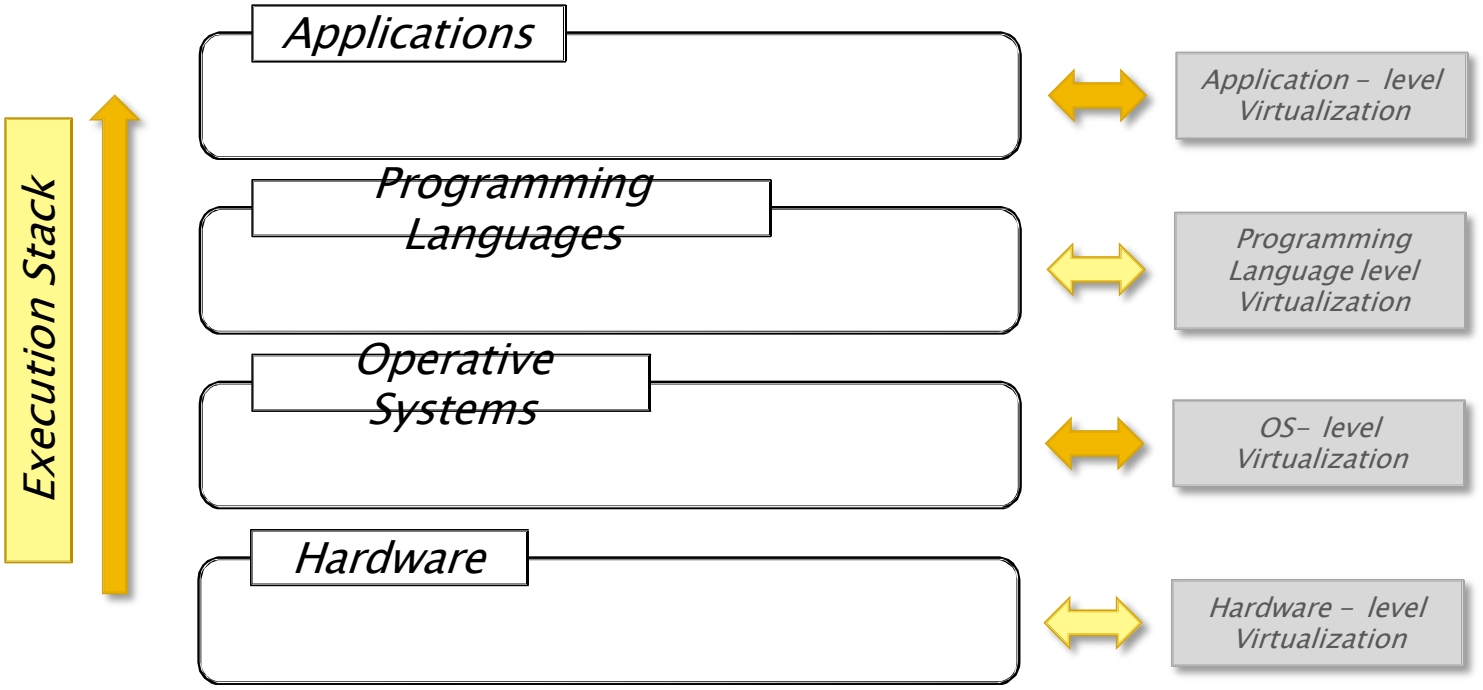
Envelope

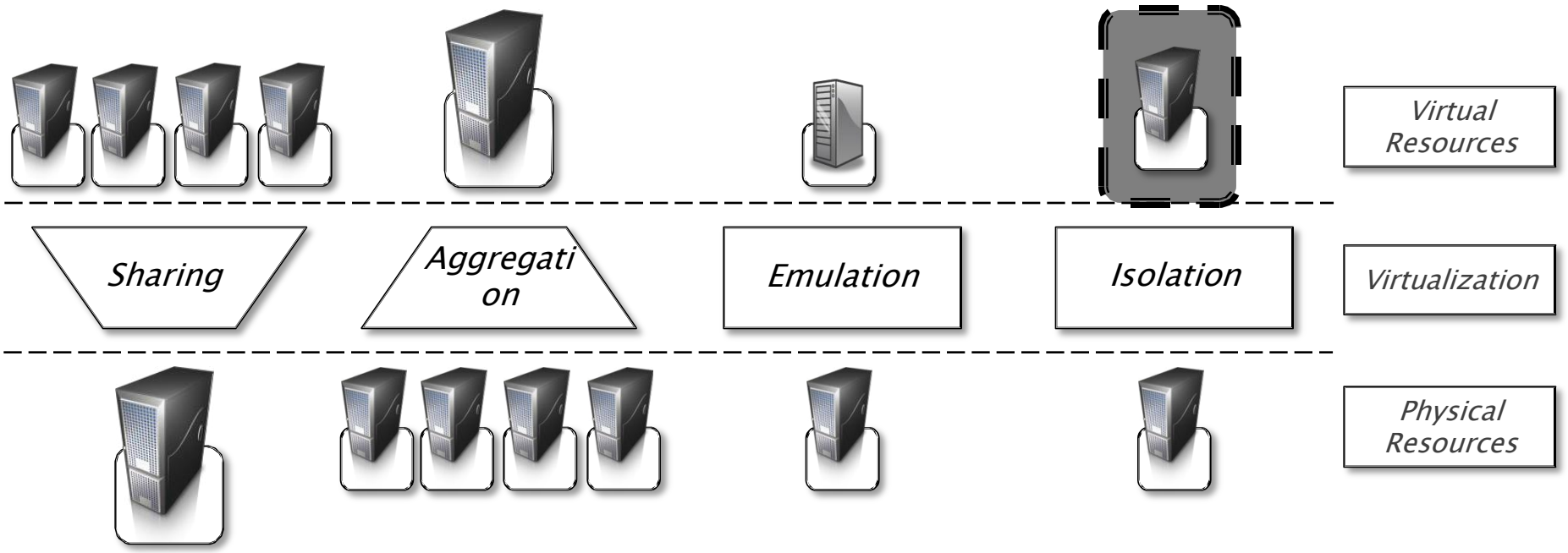
Header: Metadata & Assertions

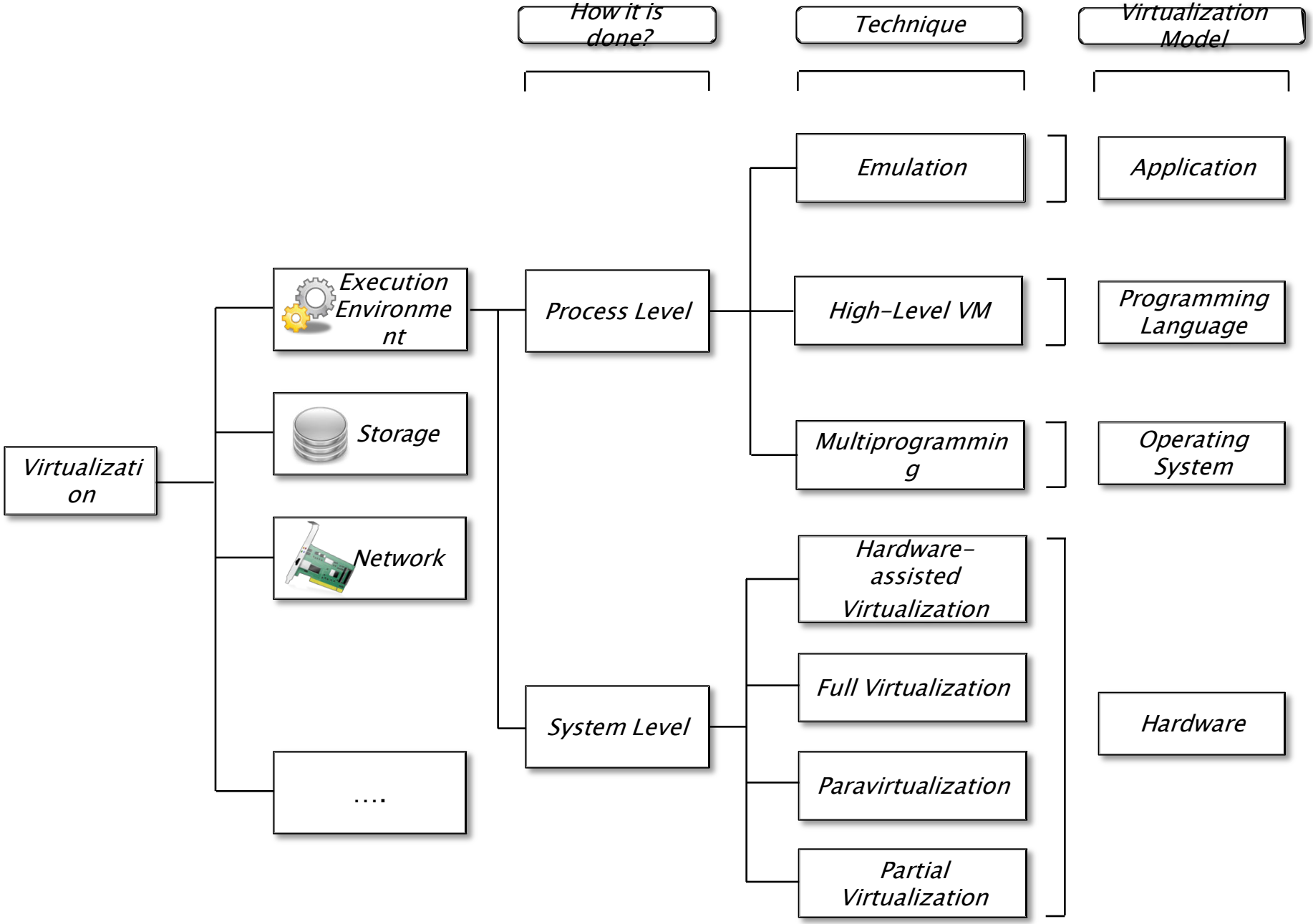
Body: Execution Result

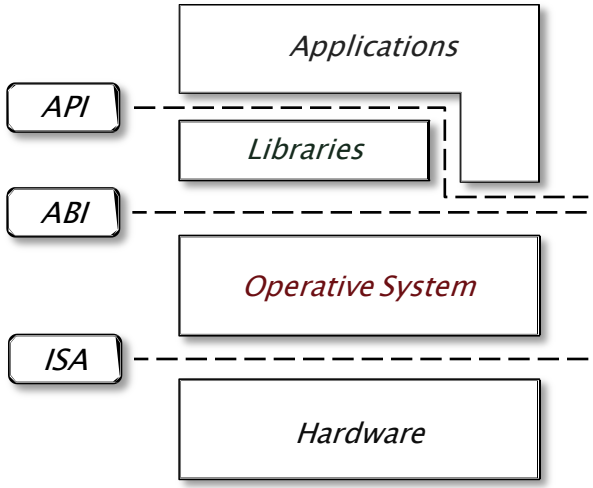

Guest
Virtual Image
Applications
Applications
Virtualization Layer
Virtual Hardware
Virtual Storage
Virtual Networking
Software Emulation
Host
Physical Hardware
Physical Storage
Physical Networking

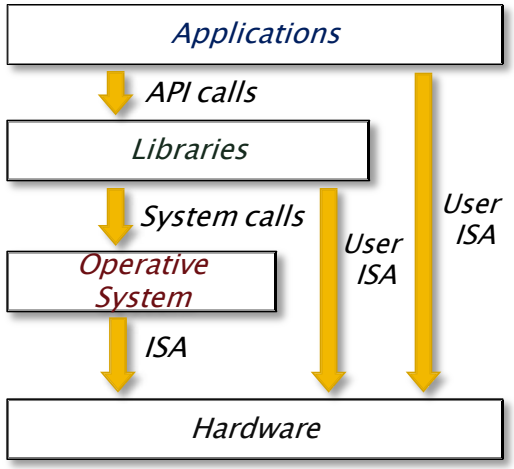
UNIT-II

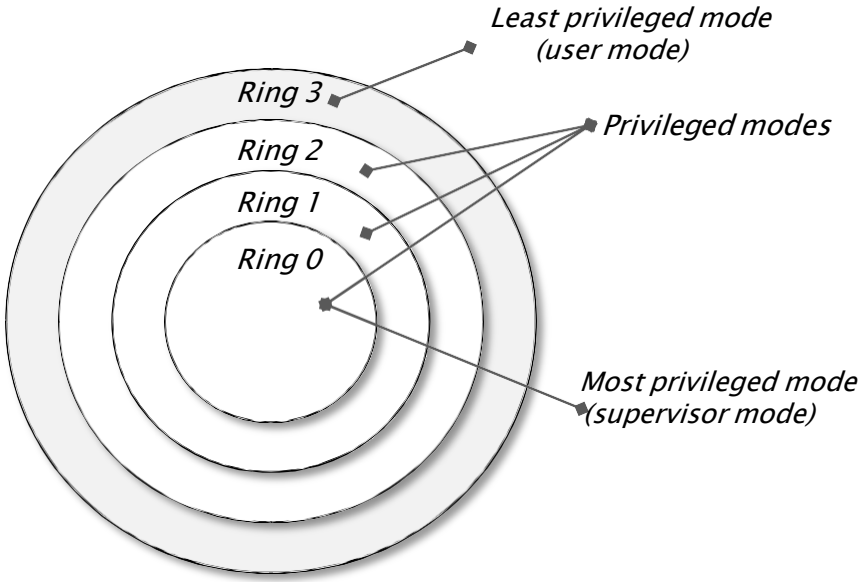

Execution Stack
Applications
Programming Languages
Operative Systems
Hardware
Application – level Virtualization
Programming Language level Virtualization
OS– level Virtualization
Hardware – level Virtualization

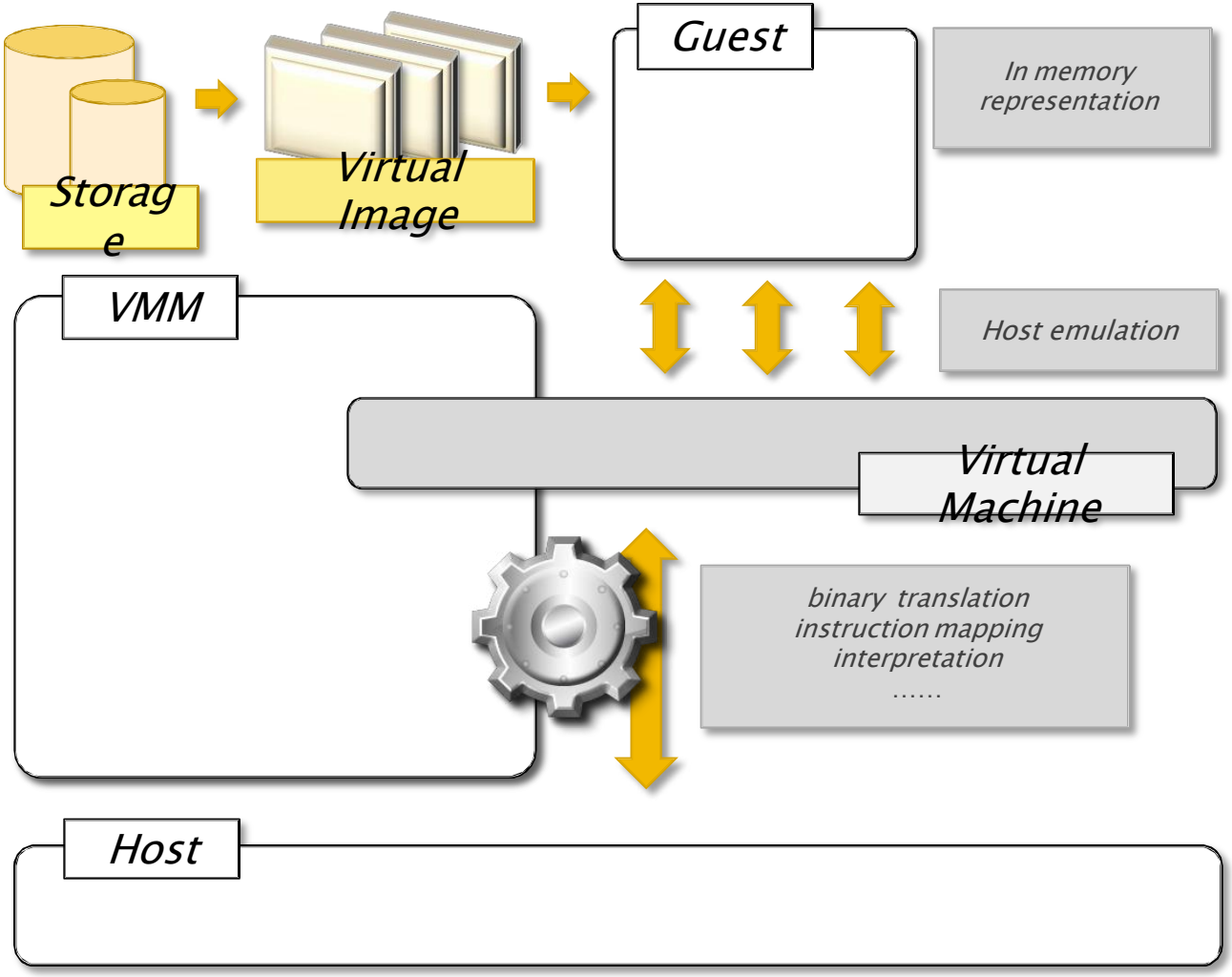

Virtual Resources
Sharing
Aggregation
Emulation
Isolation
Virtualization
Physical Resources

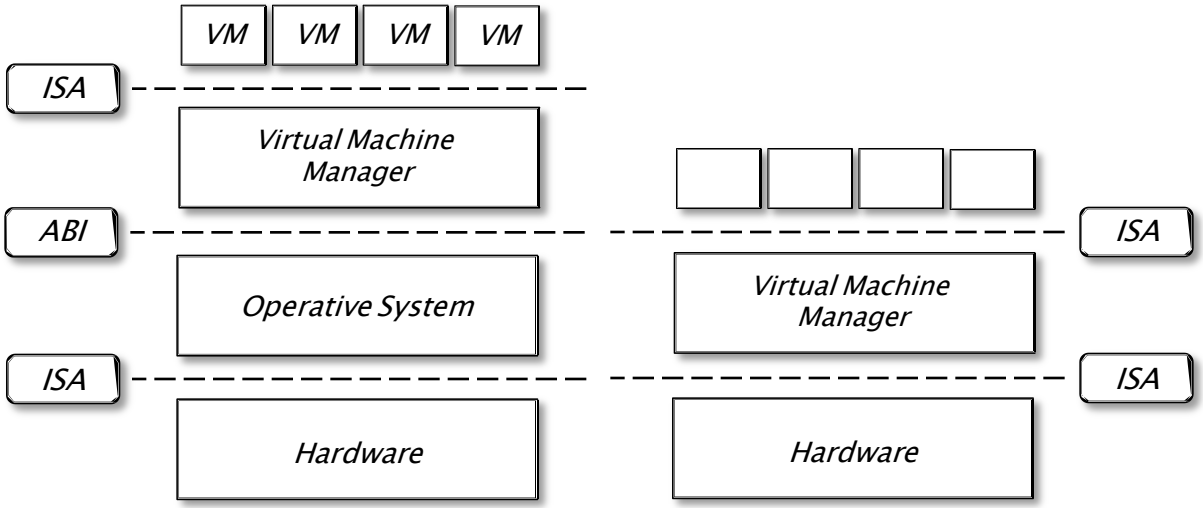

How it is done?
Technique
Virtualization Model
Virtualization
Execution Environment
Storage
Network
....
Process Level
System Level
Emulation
High-Level VM
Multiprogramming
Hardware-assisted Virtualization
Full Virtualization
Paravirtualization
Partial Virtualization
Application
Programming Language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

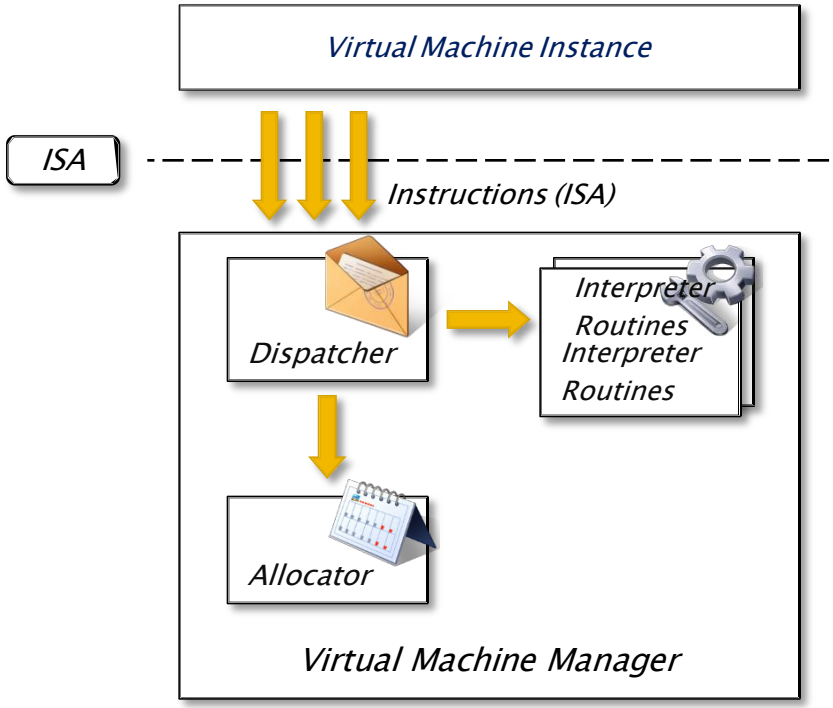

Applications
API
Libraries
ABI
Operative System
ISA
Hardware

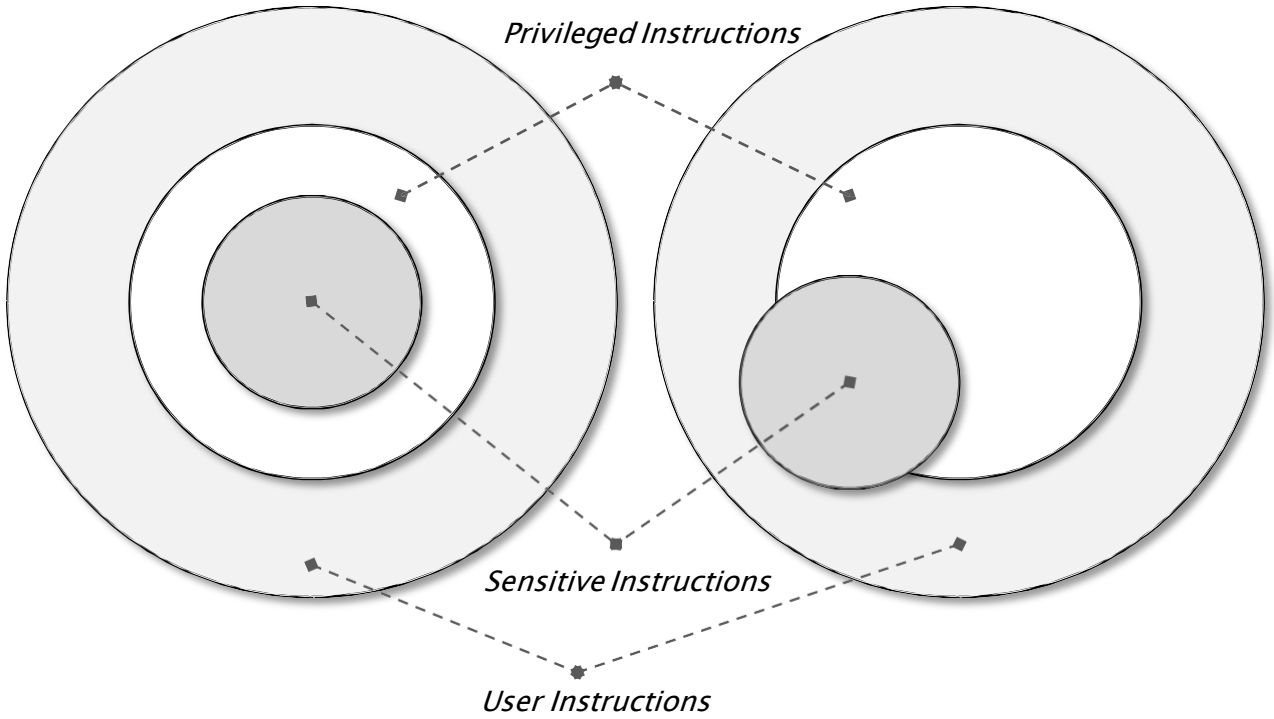

Applications
API calls
Libraries
System calls
Operative System
ISA
User ISA
User ISA
Hardware

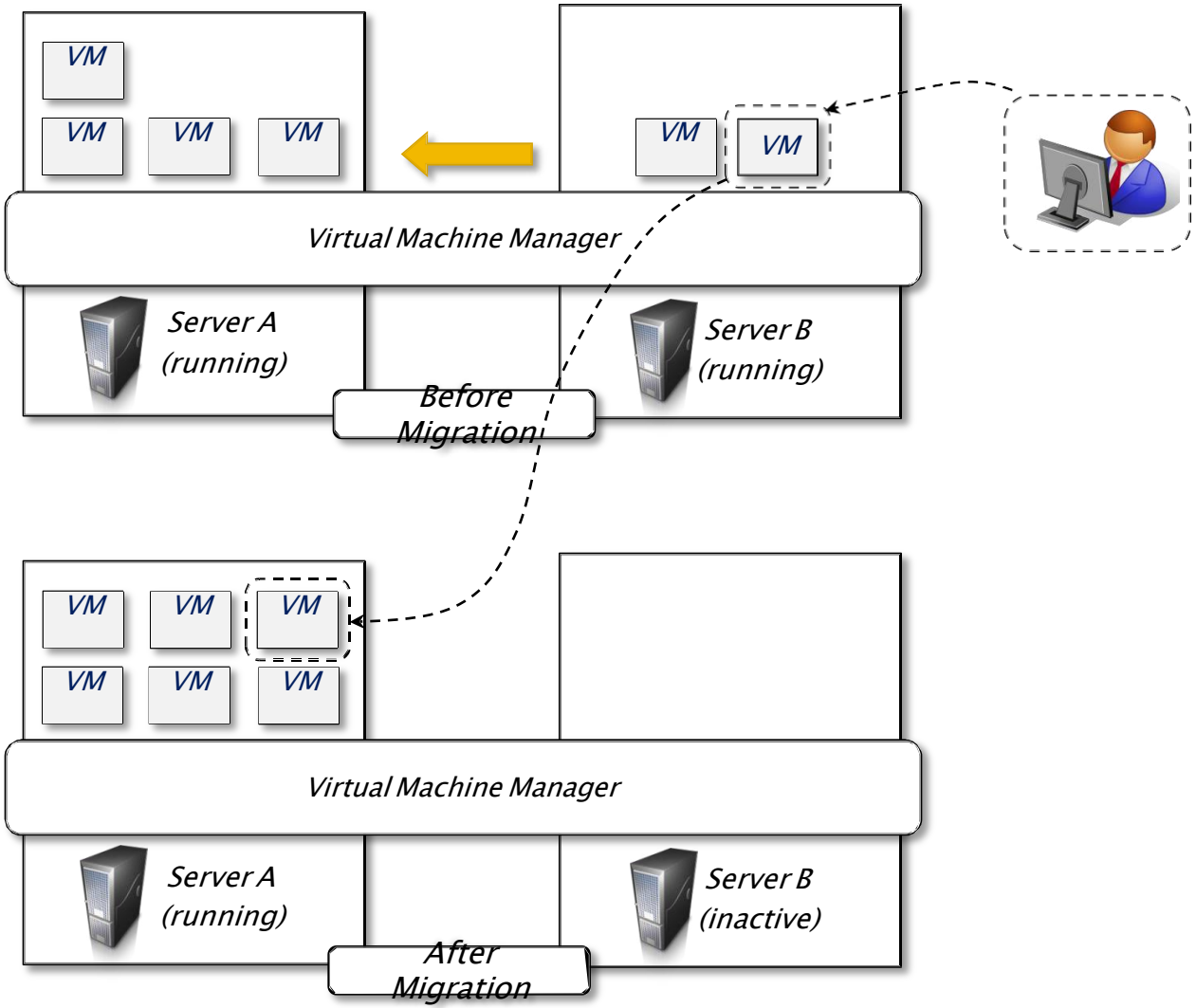

Least privileged mode
(user mode)
Ring 3
Privileged modes
Ring 2
Ring 1
Ring 0
Most privileged mode
(supervisor mode)

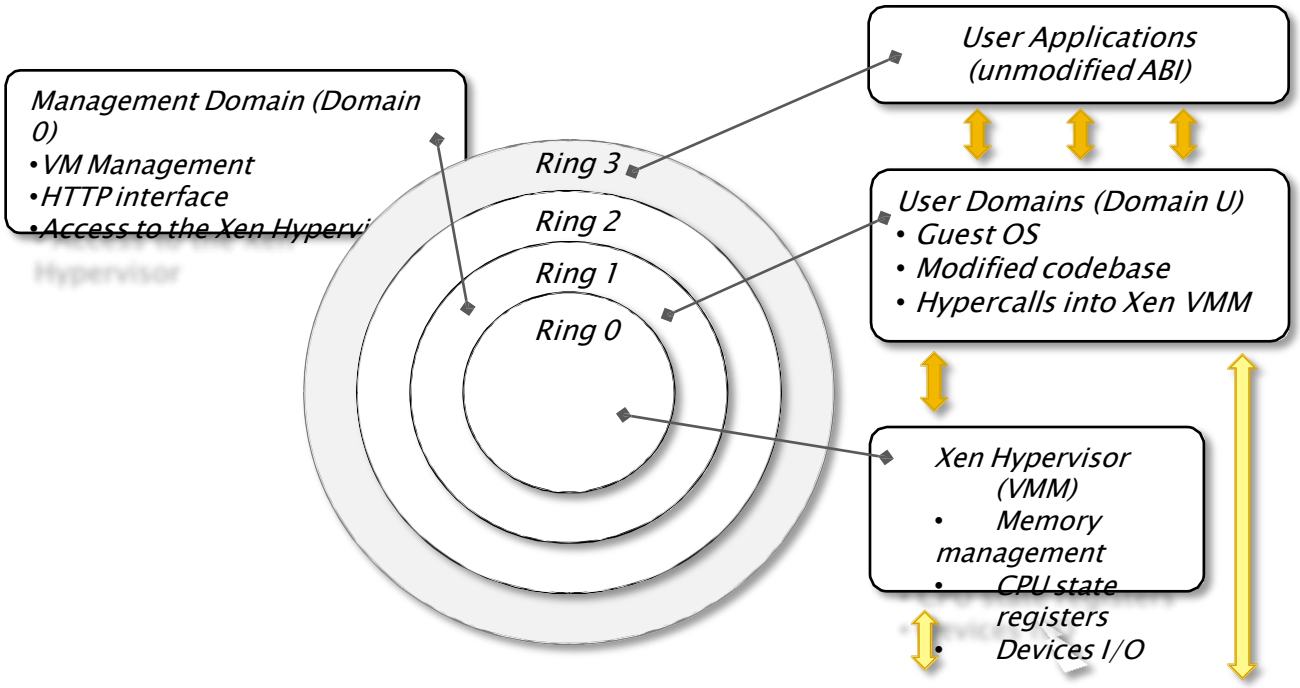

Storag
e
Virtual
Image
Guest
In memory
representation
VMM
Host emulation
Virtual
Machine
binary translation
instruction mapping
interpretation
……
Host

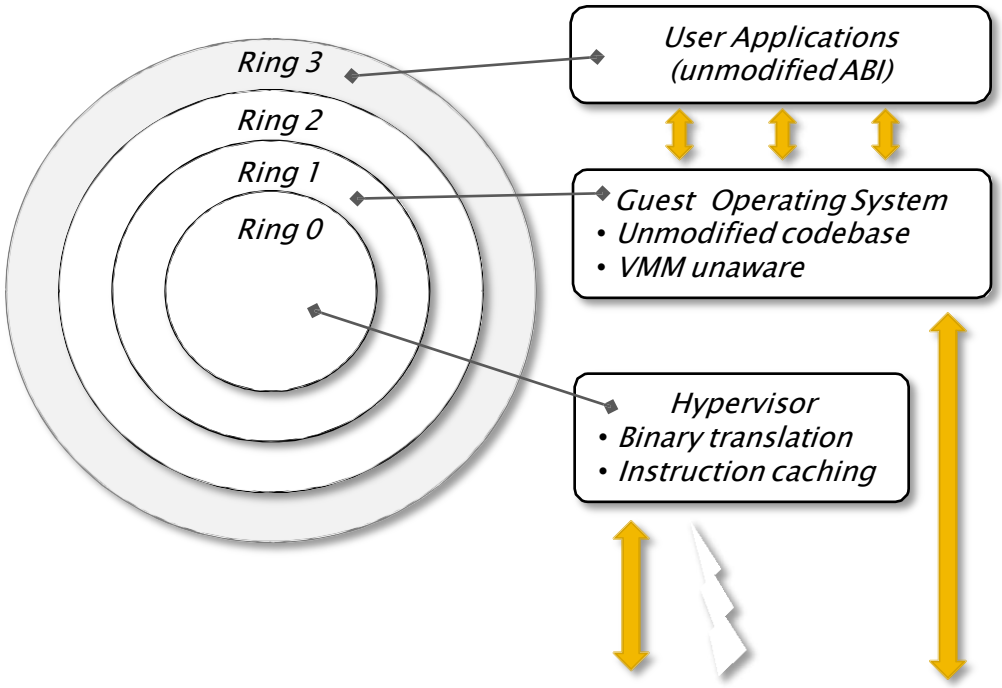

VM
VM
VM
VM
ISA
Virtual Machine Manager
ABI
ISA
Operative System
Virtual Machine Manager
ISA
ISA
Hardware
Hardware

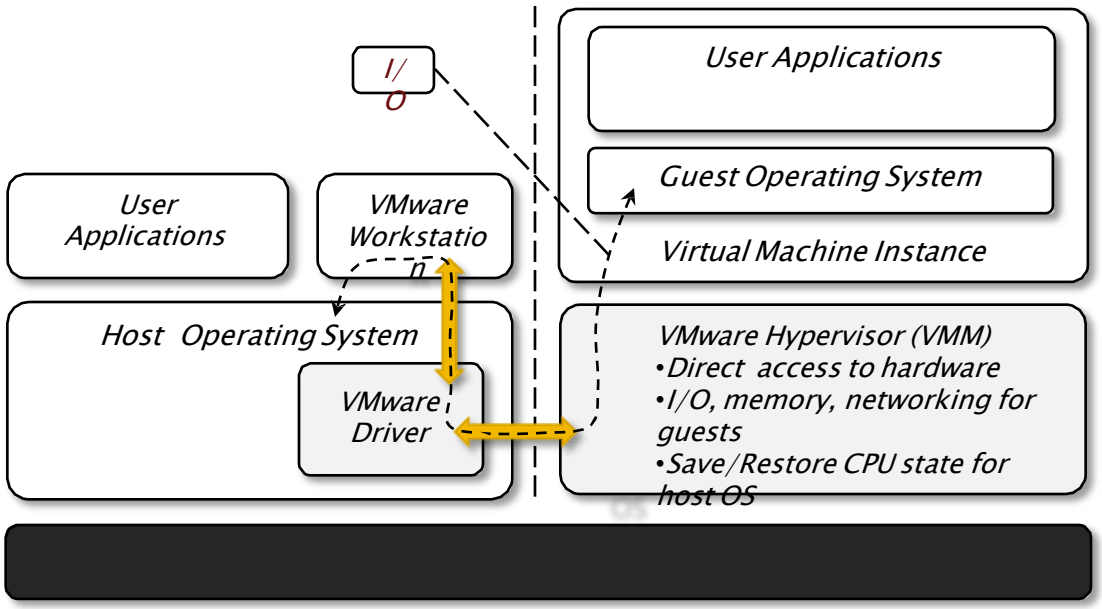

Virtual Machine Instance
ISA
Instructions (ISA)
Dispatcher
Interpreter Routines
Interpreter Routines
Allocator
Virtual Machine Manager

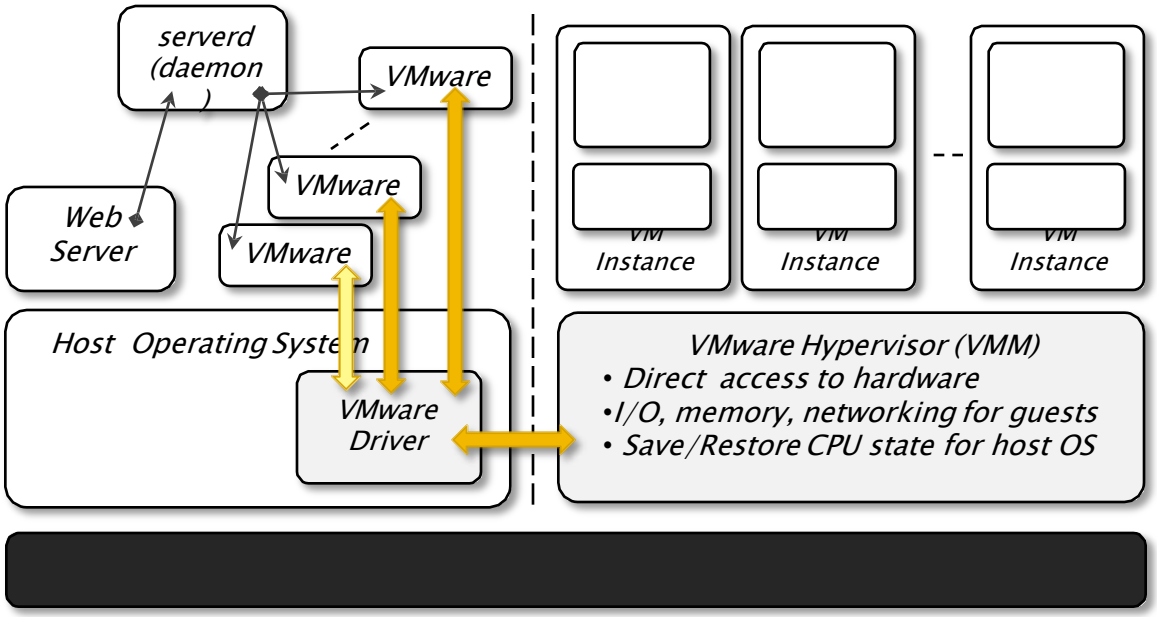

Privileged Instructions
Sensitive Instructions
User Instructions

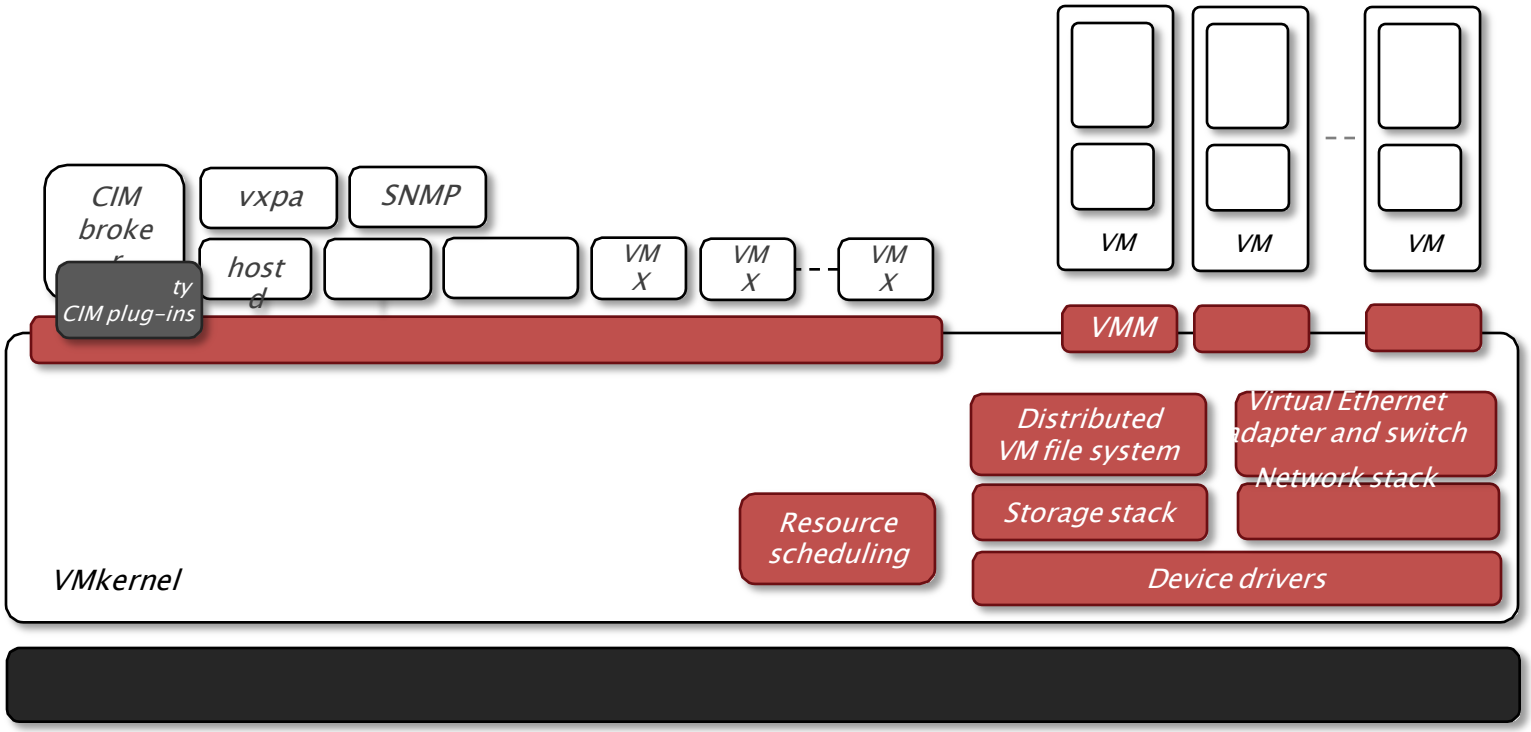

VM
VM
VM
VM
VM
VM
Virtual Machine Manager
Server A
(running)
Server B
(running)
Before
Migration
VM
VM
VM
VM
VM
VM
Virtual Machine Manager
Server A
(running)
Server B
(inactive)
After
Migration

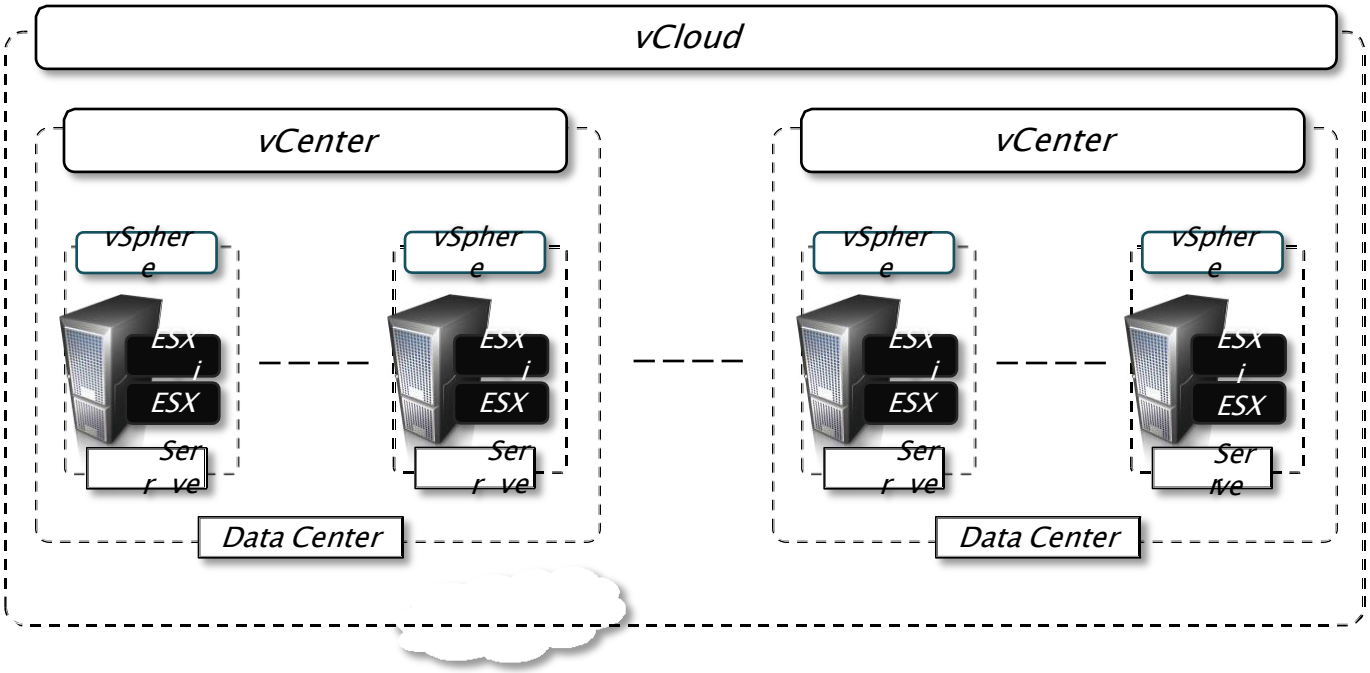

User Applications
(unmodified ABI)
Management Domain (Domain 0)
• VM Management
• HTTP interface
• Access to the Xen Hypervi
Hypervisor
Ring 3
Ring 2
Ring 1
Ring 0
User Domains (Domain U)
• Guest OS
• Modified codebase
• Hypercalls into Xen VMM
Xen Hypervisor
(VMM)
• Memory management
• CPU state registers
• Devices I/O

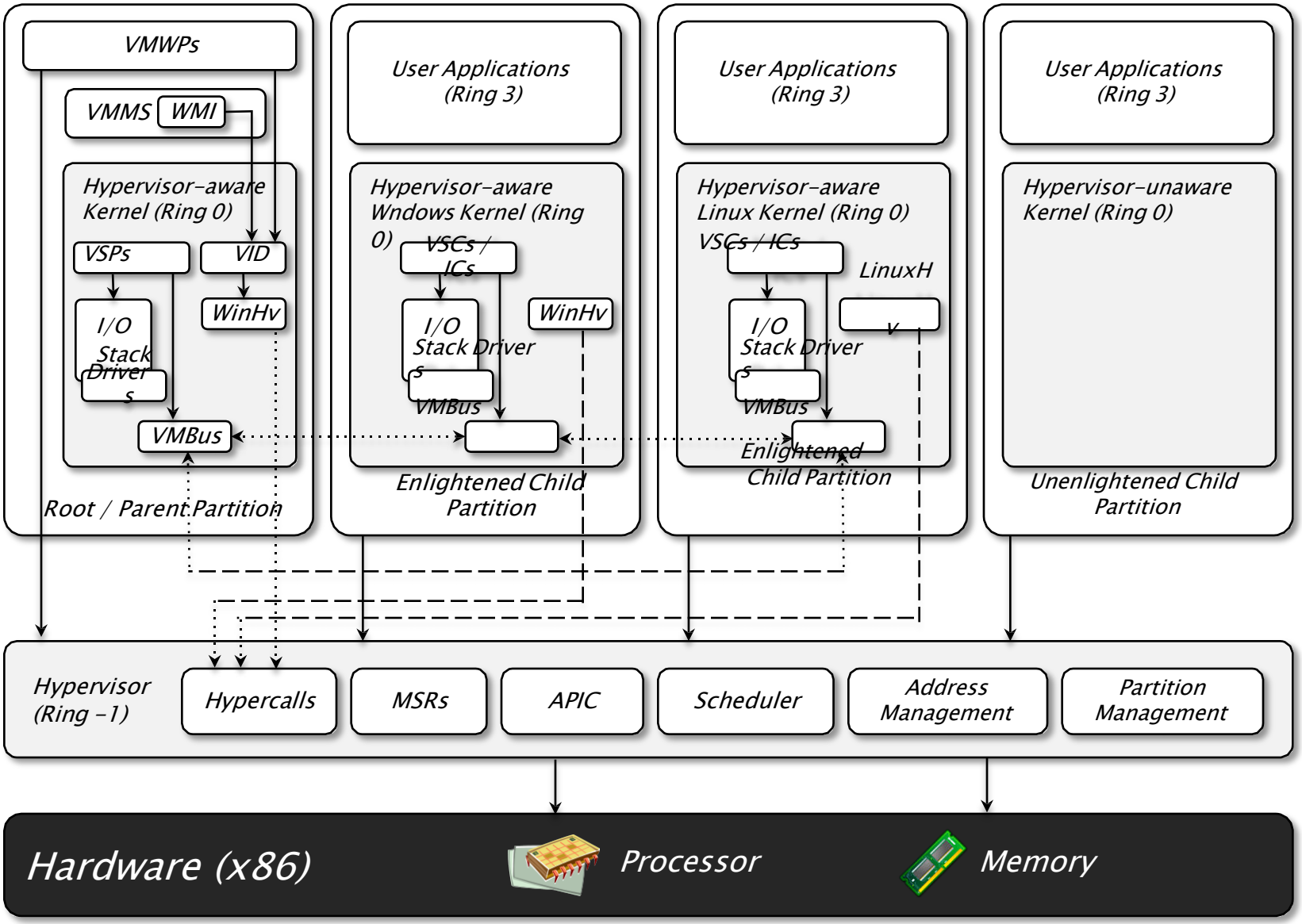

Ring 3
Ring 2
Ring 1
Ring 0
User Applications
(unmodified ABI)
Guest Operating System
• Unmodified codebase
• VMM unaware
Hypervisor
• Binary translation
• Instruction caching


I/O
User Applications
Guest Operating System
Virtual Machine Instance
User Applications
VMware Workstation
Host Operating System
VMware Driver
VMware Hypervisor (VMM)
•Direct access to hardware
•I/O, memory, networking for guests
•Save/Restore CPU state for host OS


serverd
(daemon
)
VMware
VMware
VMware
Web
Server
Host Operating System
VMware
Driver
VM
Instance
VM
Instance
VM
Instance
VMware Hypervisor (VMM)
• Direct access to hardware
•I/O, memory, networking for guests
• Save/Restore CPU state for host OS


CIM broke
vxpa
SNMP
host d
VM X
VM X
VM X
ty
CIM plug-ins
VM
VM
VM
VMM
Distributed VM file system
Virtual Ethernet adapter and switch
Network stack
Storage stack
Resource scheduling
Device drivers
VMkernel


vCloud
vCenter
vCenter
vSphere
vSphere
vSphere
vSphere
ESXi
ESX
Server
ESXi
ESX
Server
ESXi
ESX
Server
ESXi
ESX
Server
Data Center
Data Center


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
LinuxHv
I/O Stack Drivers
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

UNIT-III

Migration into a cloud

Introduction

- *disruptive techno-commercial model*
- *Answer the following questions*
  - *when and how to migrate* one's application *into a cloud?*
  - *what part or component of the IT application to migrate into a cloud and what not to migrate into a cloud?*
  - *what kind of customers really benefit from migrating their IT into the cloud?*
  - *Definition*
  - *It is a techno-business disruptive model of using distributed large-scale data centers either private or public or hybrid offering customers a scalable virtualized infrastructure or an abstracted set of services qualified by service-level agreements (SLAs) and charged only by the abstracted IT resources consumed.*

Promise of the cloud

Cloudonomics

- 'Pay per use' – Lower Cost Barriers
- On Demand Resources –Autoscaling
- Capex vs OPEX – No capital expenses (CAPEX) and only operational expenses OPEX.
- SLA driven operations – Much Lower TCO
- Attractive NFR support: Availability, Reliability

Technology

- 'Infinite' Elastic availability – Compute/Storage/Bandwidth
- Automatic Usage Monitoring and Metering
- Jobs/Tasks Virtualized and Transparently 'Movable'
- Integration and interoperability 'support' for hybrid ops
- Transparently encapsulated & abstracted IT features.

The Cloud Service Offerings and Deployment Models

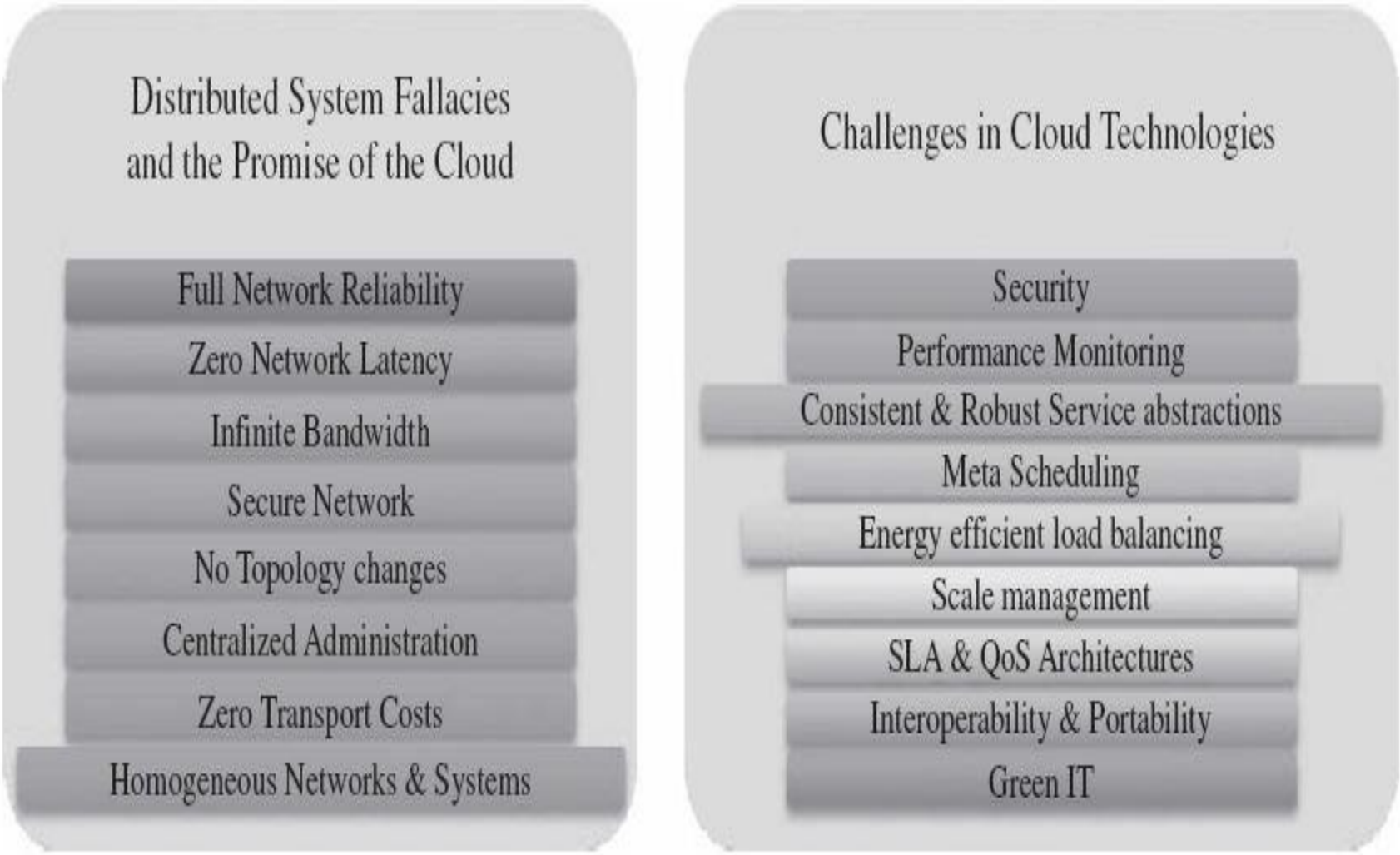
IaaS **IT Folks**	• Abstract Compute/Storage/Bandwidth Resources • Amazon Web Services[10,9] – EC2, S3, SDB, CDN, CloudWatch
PaaS **Programmers**	• Abstracted Programming Platform with encapsulated infrastructure • Google Apps Engine(Java/Python), Microsoft Azure, Aneka[13]
SaaS **Architects & Users**	• Application with encapsulated infrastructure & platform • Salesforce.com; Gmail; Yahoo Mail; Facebook; Twitter

Cloud Application Deployment & Consumption Models		
Public Clouds	Hybrid Clouds	Private Clouds

Examples

- *IaaS*
  - *Amazon services*
  - *Elastic Cloud Compute (EC2)*
    - *small-instance*
    - *large-instance*
    - *extra-large instance*
    - *high-cpu instance*
    - *high-cpu medium instance*
    - *high-cpu extra-large instance*
- *SaaS*
  - *Gmail*
  - *Scalable storage space*

Challenges in cloud



Why Migrate

- *Reasons*
  - *Economic*
  - *Business*
  - *Technologic*
- *Five level of migration*
  - *Application*
  - *Code*
  - *Design*
  - *Architecture*
  - *Usage*
- *Clean and independent a*

$$P \rightarrow P'_C + P'_l \rightarrow P'_{\mathrm{OFC}} + \mathrm{P}'_l$$

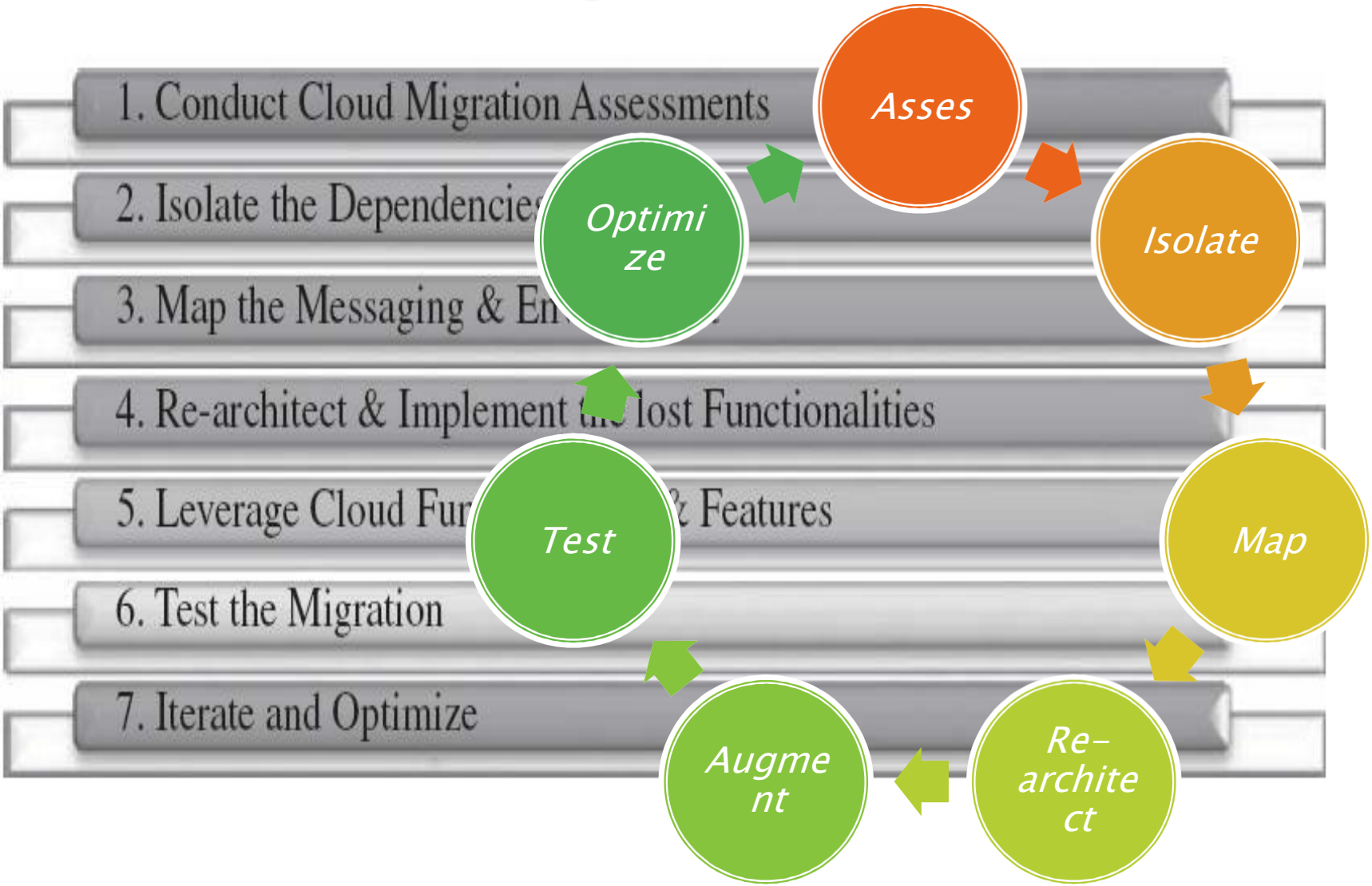
- *Code(design) needs to be modified and adapted*
- *Usage of application needs to be modified and adapted*
- *Hybrid Cloud*

Cloudonomics

- *economics and the associated trade-offs, of leveraging the cloud computing services*
- *Factors to migrate*

- *Economic*
  - *CaPex*
  - *Opex*
  - *When?*
    - *cost of using cloud + cost of migration < cost of using captive dc*
- *Licensing issues*
- *SLA compliances*
- *pricing of the cloud service offerings*
  - *Elasticity and pricing variability*

Model of Migration



Migration in AWS

- *Assessment*
  - *Isolate dependency*
- *Reference migration architecture*
- *Data migration*
- *Application migration*
- *Leveraging AWS features*
- *Optimize for Cloud*

Migration risk

- *Is a challenge*
- *Identify in test phase*
- *Mitigate in optimization phase*
- *Types*
  - *General*
    - *Performance monitoring & tuning*
    - *Disaster recovery*
    - *Compliance with standards and governance issues*
    - *Licensing issues*
    - *QoS*
    - *Portability and interoperability*
    - …
  - *Security-related*
    - *Issues of security at various level of app*
    - *issues of trust and issues of privacy*
    - *Right execution logs*
    - *Consistent identity management*
    - …

Enriching the integration as a service paradigm for the cloud era

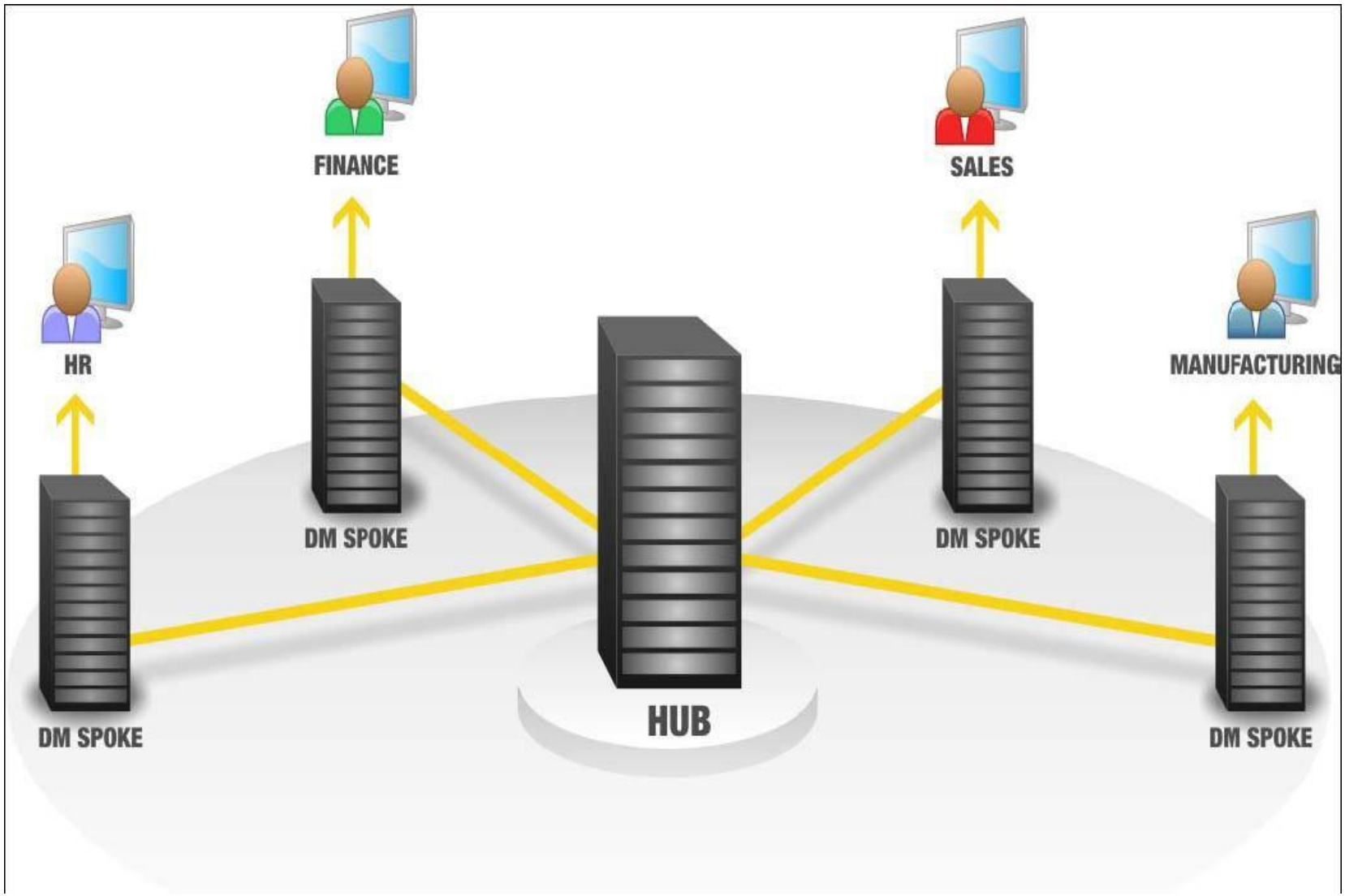
UNIT-IV

Introduction

- *Integration*
  - *E2E (Enterprise to Enterprise)*
  - *E2C (Enterprise to Cloud)*
  - *C2C (Cloud to Cloud)*

- *Systems*
  - *One-to-many*
  - *One-to-one*
  - *Many-to-one*

Evolution of SaaS



Challenges of SaaS paradigm

- *Challenges*
  1. *Controllability*
  2. *Visibility & flexibility*
  3. *Security and Privacy*
  4. *High Performance and Availability*
  5. *Integration and Composition*
  6. *Standards*
- Private cloud,…
- *Integration*
  - *Real time data and functionality*
- *API*
  - *Too coding*
- *Data Transmission security*
- *Impact of the clouds*

Approaching the SaaS integration

- *Integration middleware*
  - *EAI*
    - *Enterprise Application Integration*
  - *ESB*
    - *Enterprise Service Bus for service integration*
    - *loosely coupled, in a cloud*
  - *EDB*
    - *Enterprise Data Bus for data integration*
  - *MOM*
    - *Message Oriented Middleware for integration application via Message passing*
  - *CEP*
    - *Complex Event Processing engines*
    - *decoupled*

Why SaaS Integration is hard?

- *Data synchronization*
- *Constraint of SaaS*
  - *Dynamic nature of the SaaS interfaces*
  - *Dynamic nature of the metadata*
  - *Managing assets outside of the firewall*
  - *Move Massive amounts of information*
- *Complicated integration*
  - *New integration scenarios*
  - *Limited access*
    - *Controllability, Flexibility, Visibility*
  - *Dynamic resources*
    - *Tightly coupled*
  - *Performance*

Integration Scenario

- *Integration model*
  - *Local to local*
  - *Local to cloud*
  - *Cloud to cloud*

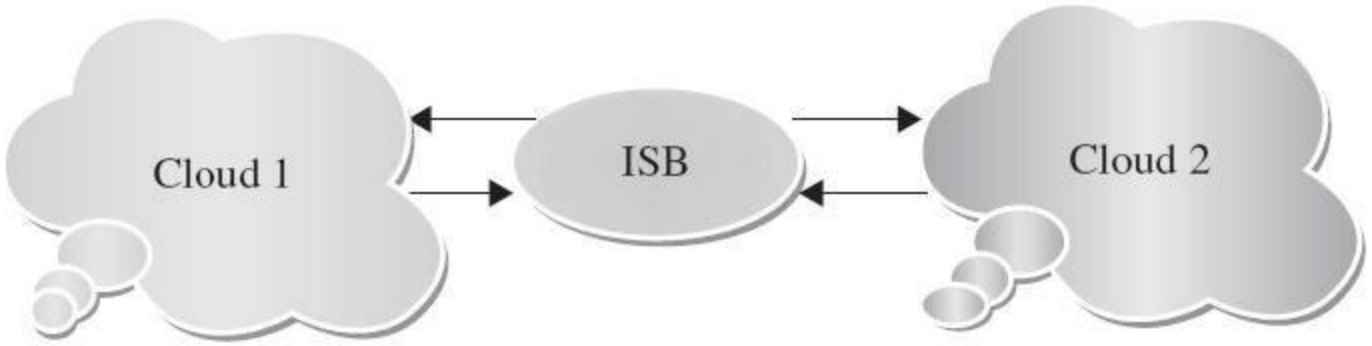
- *Three major scenario*
  - *Public cloud*
  - *Homogeneous cloud*
  - *Heterogeneous cloud*

Integration with public cloud

- *In one cloud*
- *Integration middleware*
  - *ESB or ISB*
- *Two app owned by different companies*
- *May be in single server*

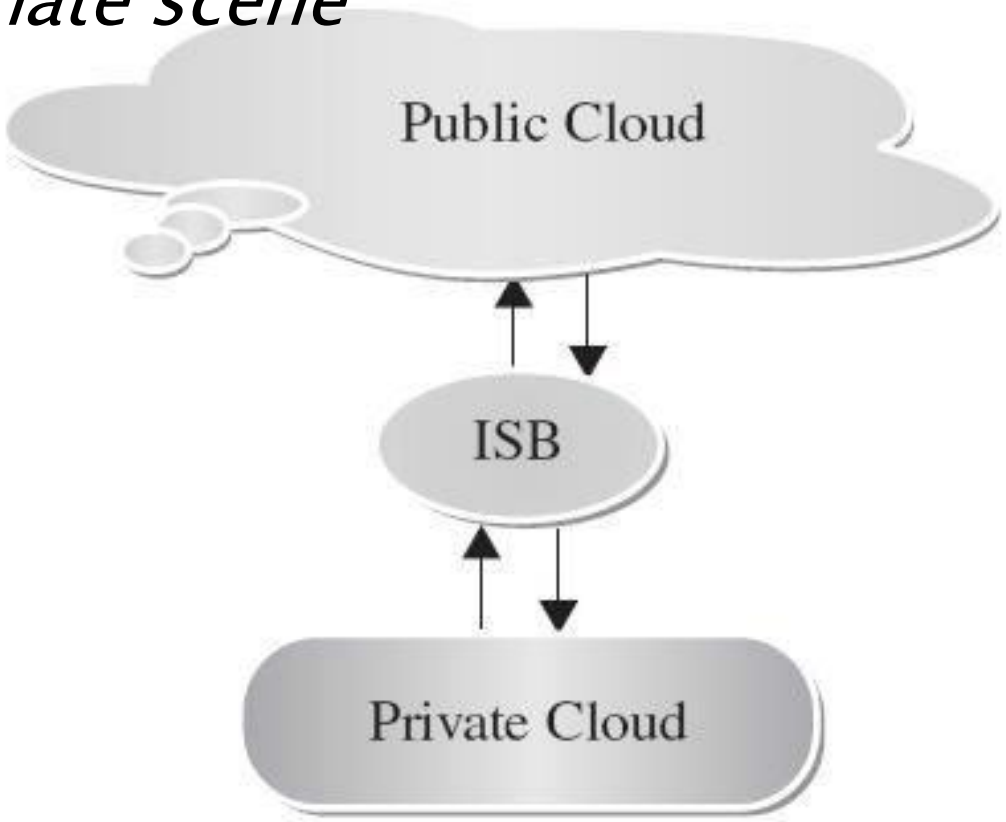
Integration homogeneous clouds

- *Two geographically separate*
- *Middleware in 1 or 2 or another cloud*



Integration Heterogeneous clouds

- *Public and private*
- *Dominate scene*



Integration Methodology

- *Three type cloud integration*
  - *i. Traditional Enterprise Integration Tools can be empowered with special connectors to access Cloud-located Applications*
    - *the most likely approach for IT organizations*
  - *ii. Traditional Enterprise Integration Tools are hosted in the Cloud*
    - *Good for C2C*
  - *iii. Integration-as-a-Service (IaaS) or On-Demand Integration Offerings*
    - *On-premise to cloud, cloud to cloud and on-premise to on-premise*
    - *Informatica on demand is an example*

Characteristic of integration

- *Connectivity*
- *Semantic mediation*
- *Data mediation*
  - *Data transformation*
- *Data Migration*
- *Data Integrity*
- *Data Security*
- *Governance*

Integration Engineering Lifecycle

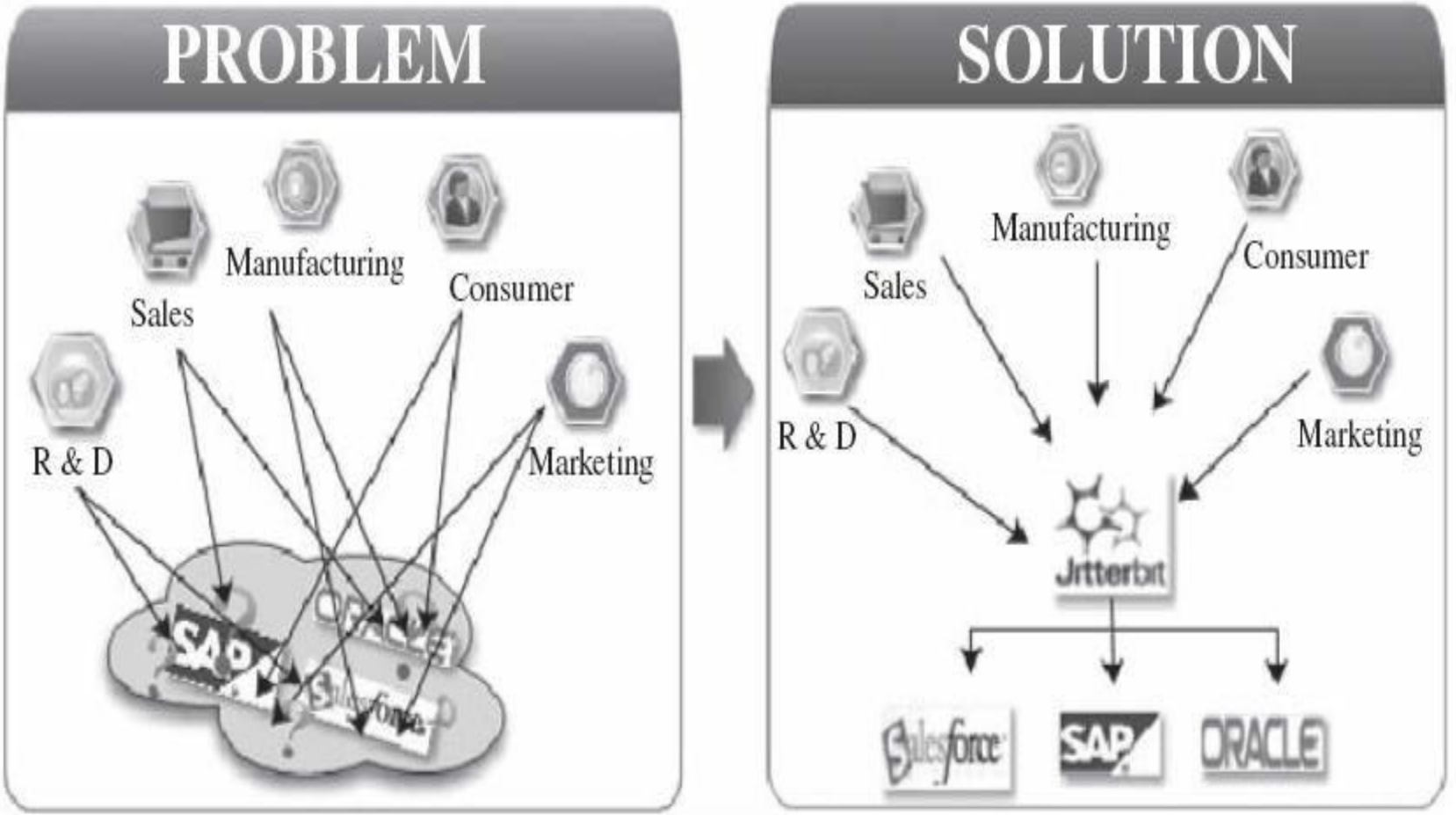
- *Understanding*
  - *Semantic understanding of source and target system*
- *Definition*
  - *Information represent, ownership, physical attributes*
- *Design*
  - *Visual mapping technology*
- *Implementation*
  - *Connecting source and target systems*
- *Test*
  - *integration is properly designed and implemented*

Products and platform

- *Jitterbit*
- *Bommi software*
  - *On demand*
- *Bungee connect*
- *OpenSource connect*
- *SnapLogic*
  - *Free community*
  - *Professional*
- *Pervasive DataCloud*
- *Bluewolf*
  - *Proactive monitoring and consulting services*
- *Online MQ*
- *CloudMQ*
- *Linxter*

Jitterbit

Enable integration among data, apps, web services



Pervasive DataCloud

- *Multi-tenant platform*
- *Deliver*
  - *Integration as a Service*
  - *Package turnkey integration*
  - *Support every integration scenario*
  - *Connectivity to hundreds of different application and data stores*
- *Is platform for deploy applications that are*
  - *Scalable*
    - *multi-tenant architecture*
  - *Flexible*
    - *SaaS-to-SaaS, SaaS to on*-premise ,…
  - *Easy to access and configure*
    - *Via web browser*
  - *Robust*
  - *Secure*
    - *automatic update, monitoring,…*
  - *Affordable*
    - *Pay-as-you-go model*

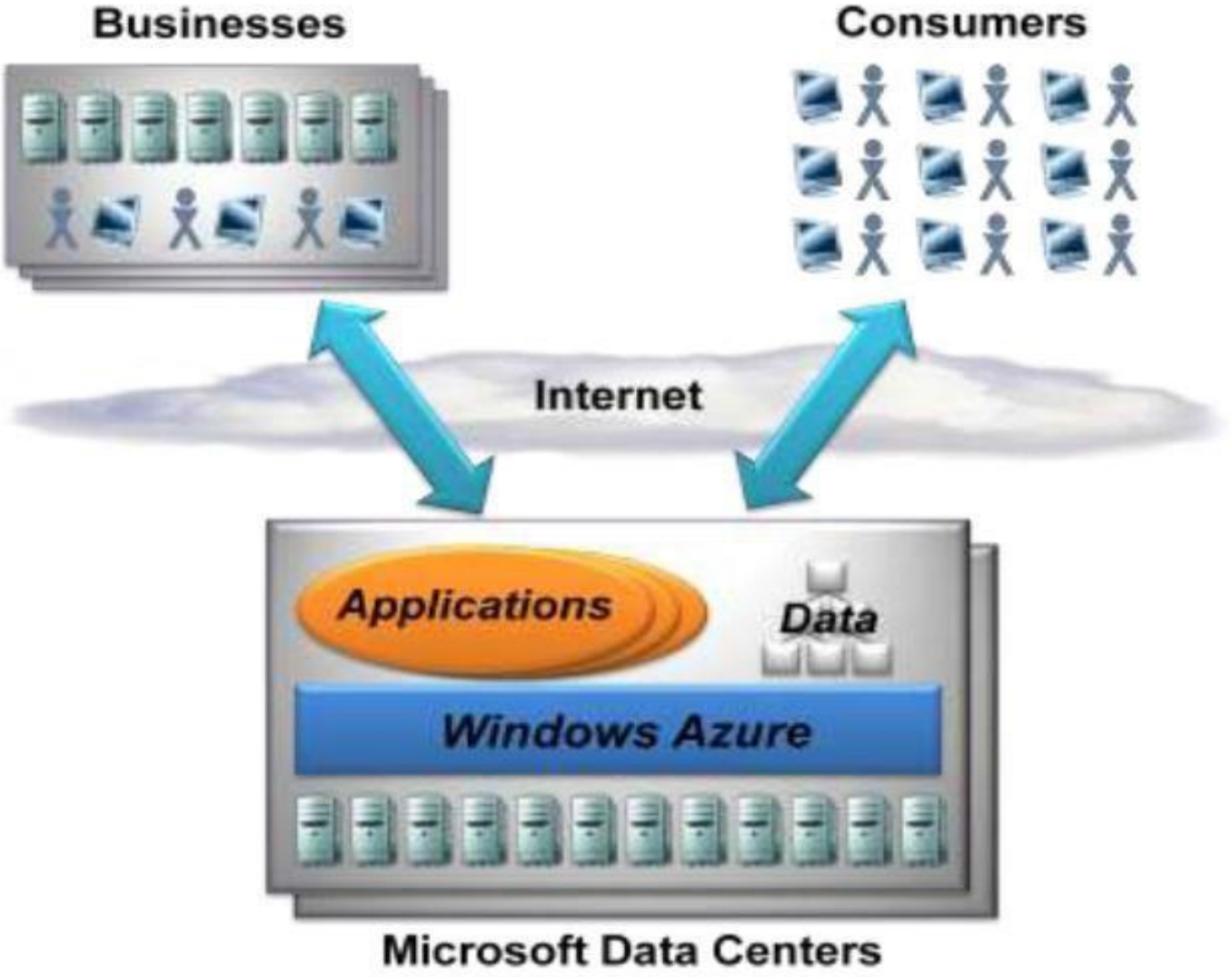
Online MQ

- *Internet based queuing system*
- *Send/Receive message over network*
- *Cloud messaging queuing service*
- *Advantages*
  - *Ease of use*
  - *No maintenance*
  - *Load balancing*
    - *Multiple instance*
  - *High availability*
    - *clustering*
  - *Easy integration*
    - *SOAP, JMS-compatible*

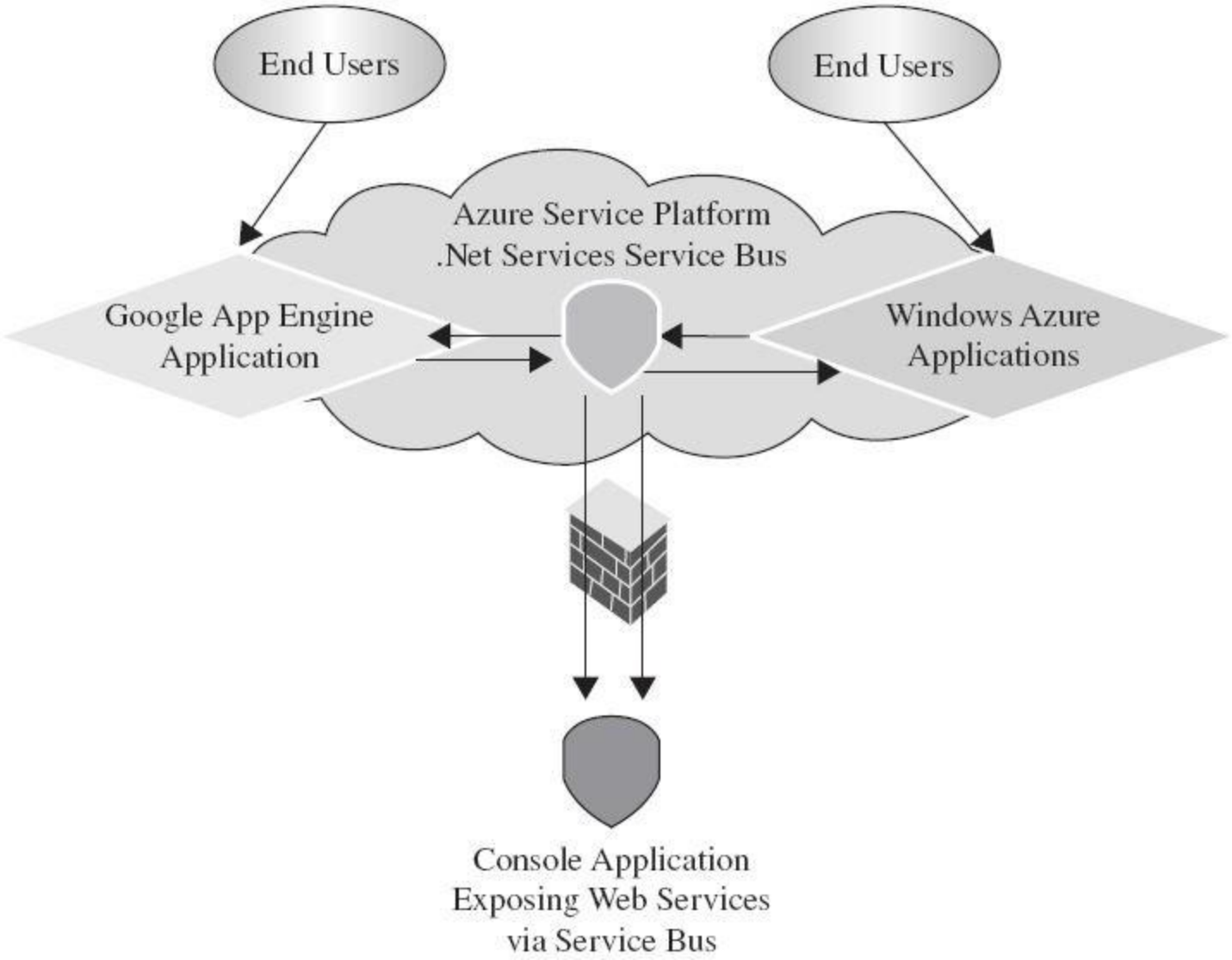
Informatica on-demand service

- *innovative on-demand data integration solutions*
- *As-a-service delivery model*
- *Benefits*
  - *Rapid development and deployment*
    - *with zero maintenance of the integration technology*
  - *Automatically upgraded and continuously enhanced by vendor*
  - *Proven SaaS integration solutions*
  - *Proven data transfer and translation technology*
- *No complex software update*
- *No additional fee*
- *Patching, versioning has no cost*

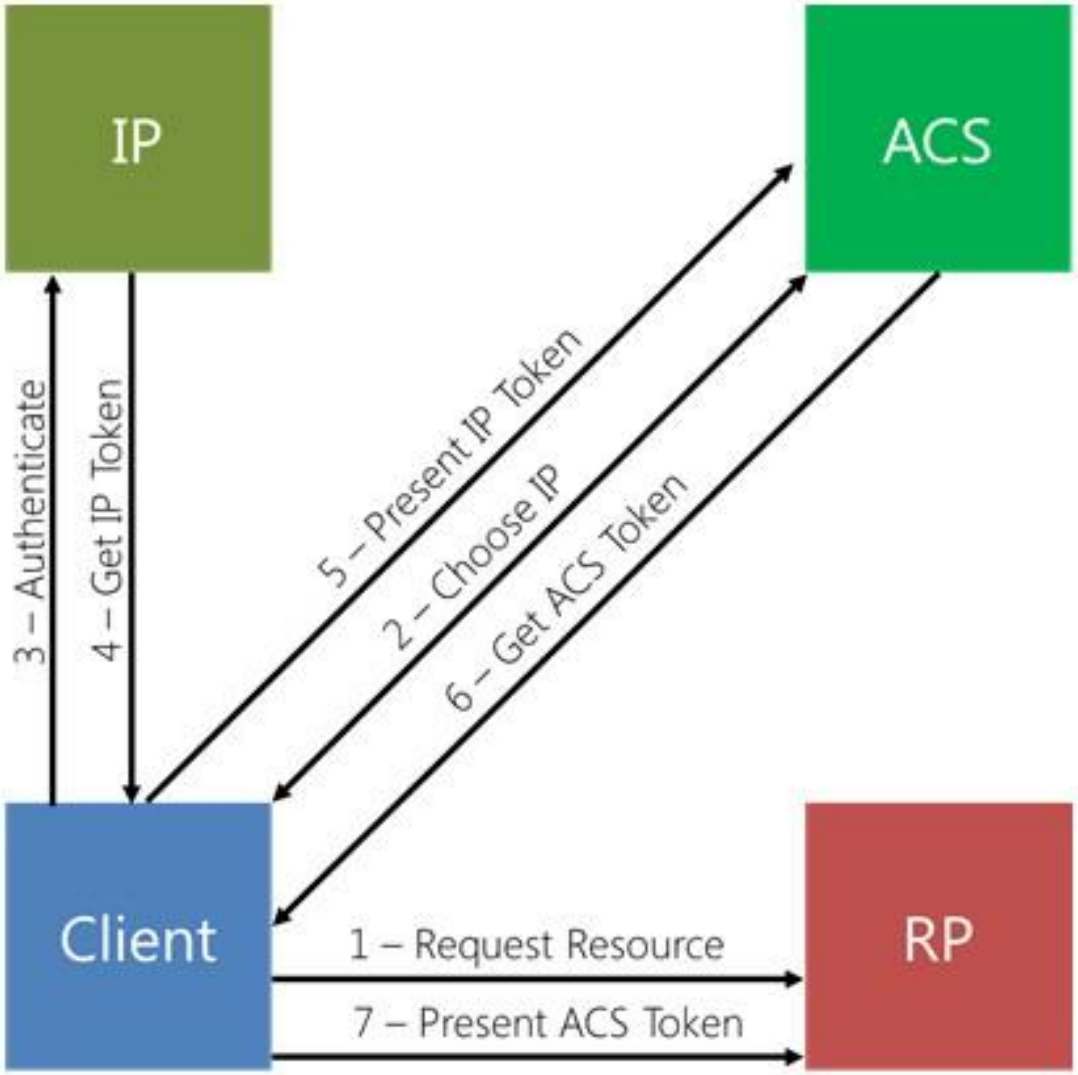
Microsoft ISB



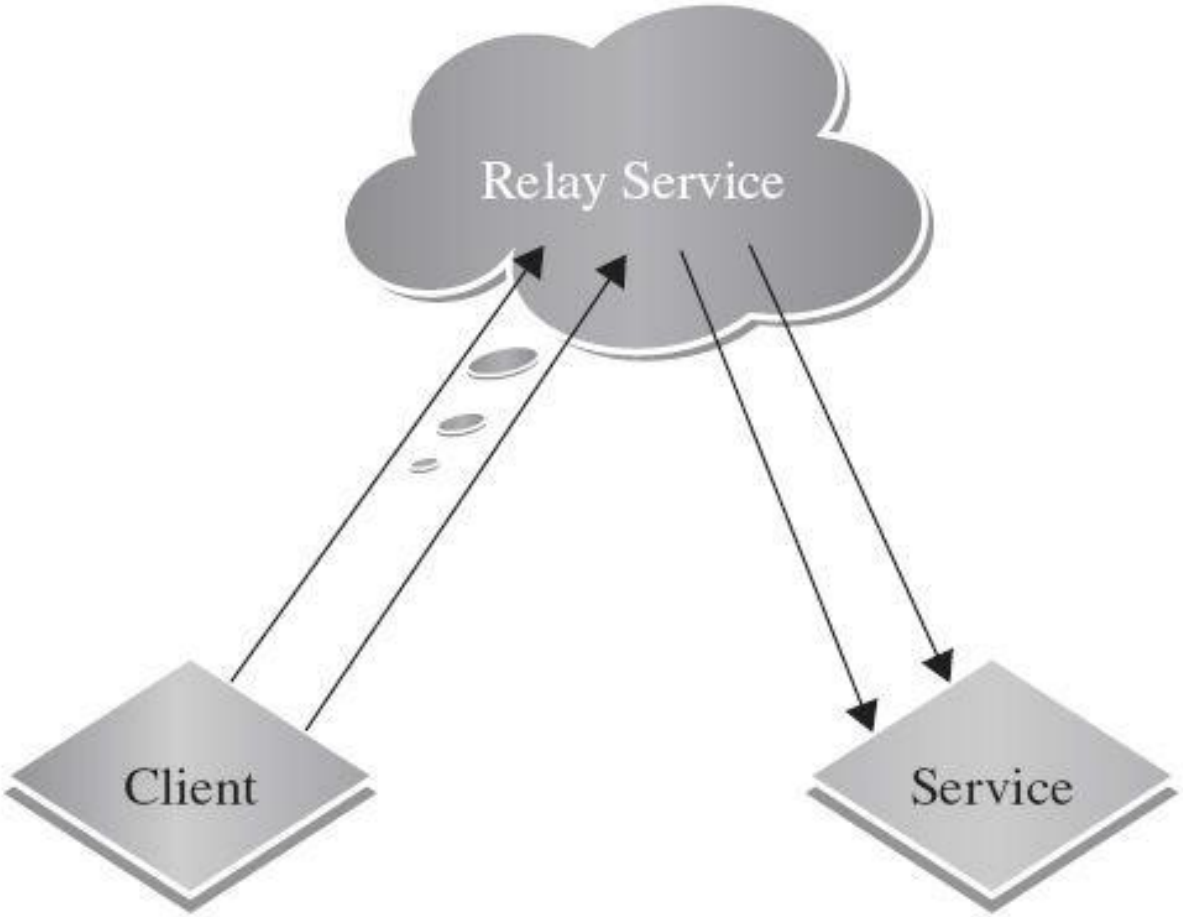
.Net Service Bus



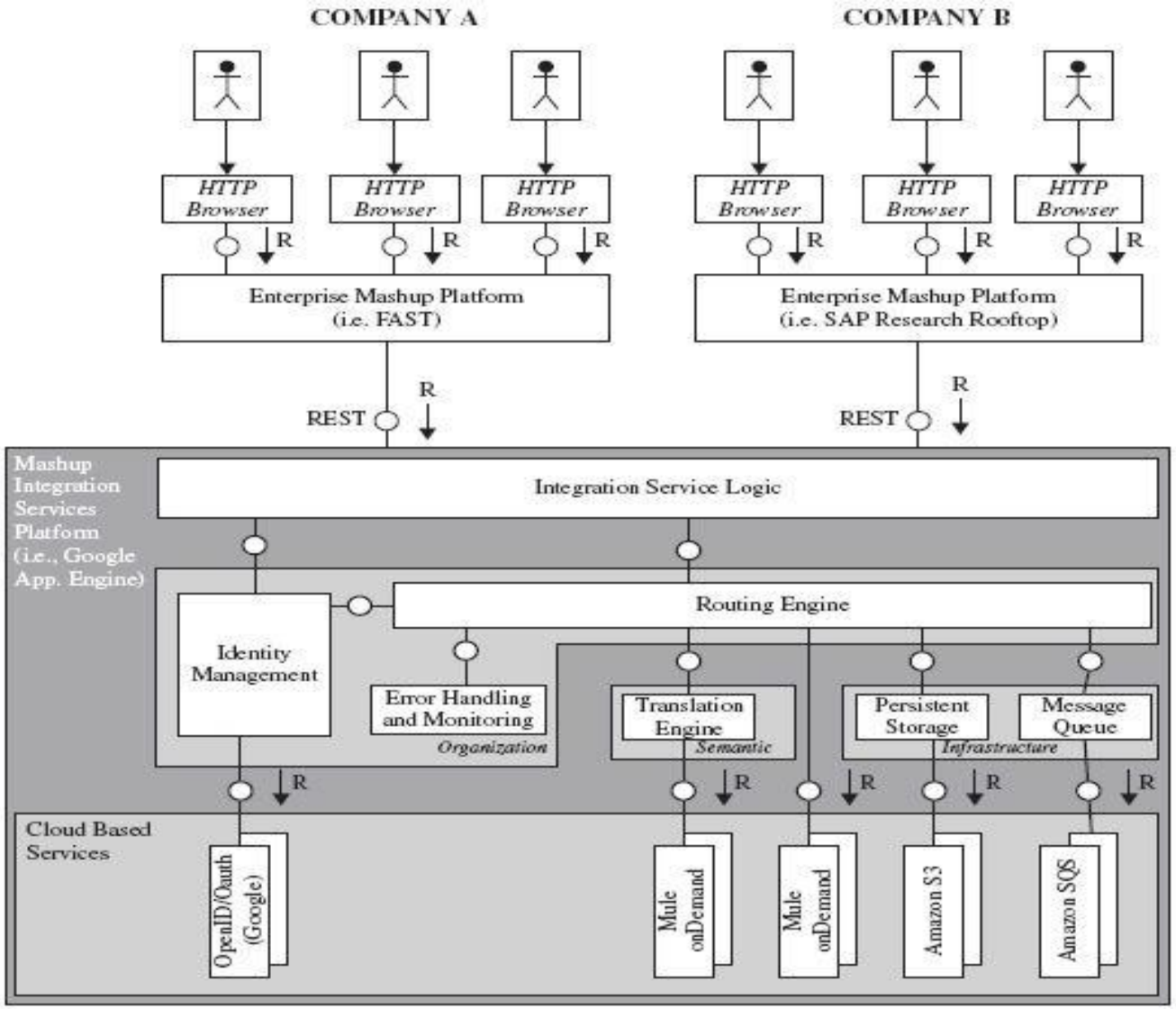
.Net Access Control Service



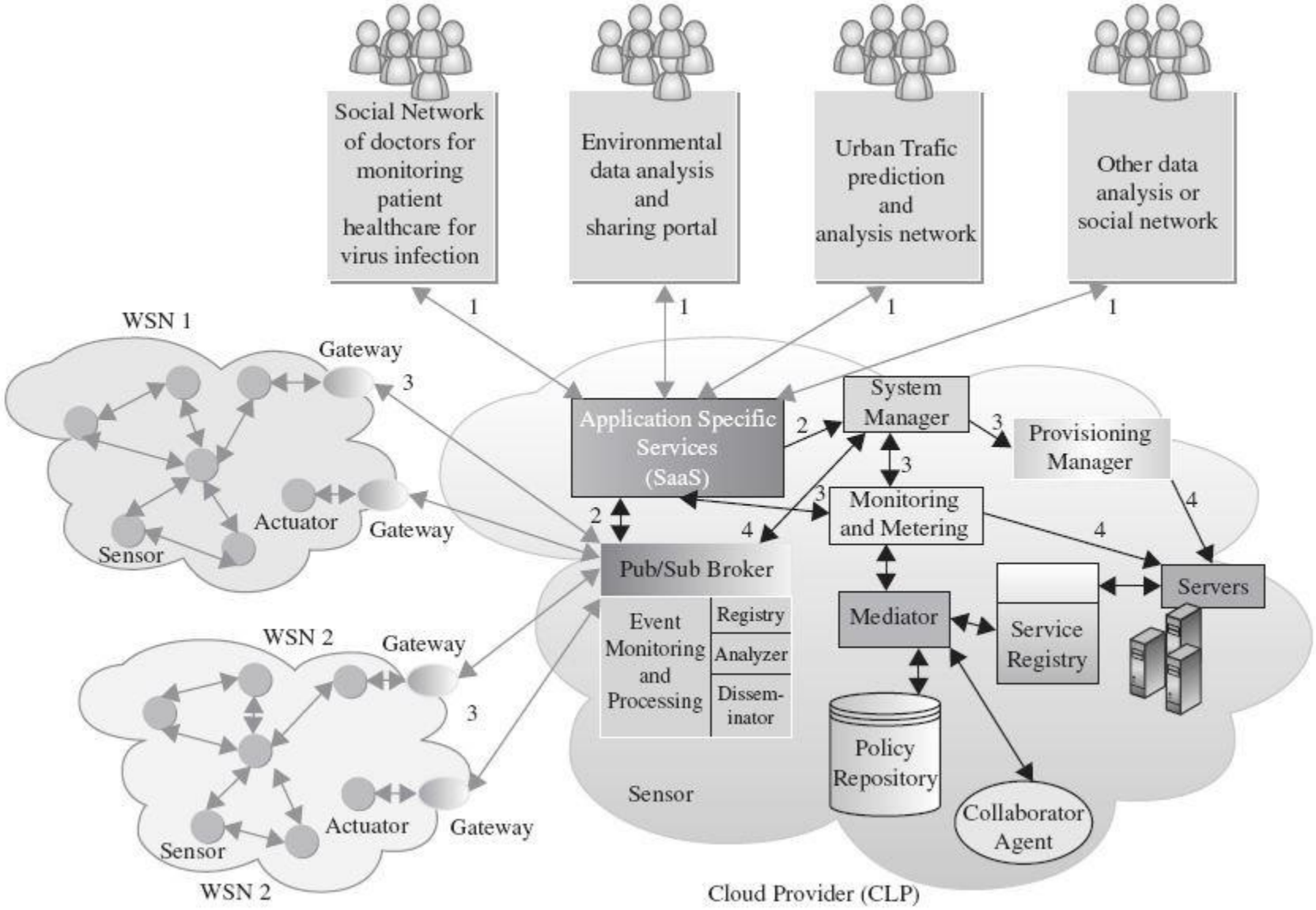
Relay Service



Enterprise mashup platform



Sensor-Cloud Integration

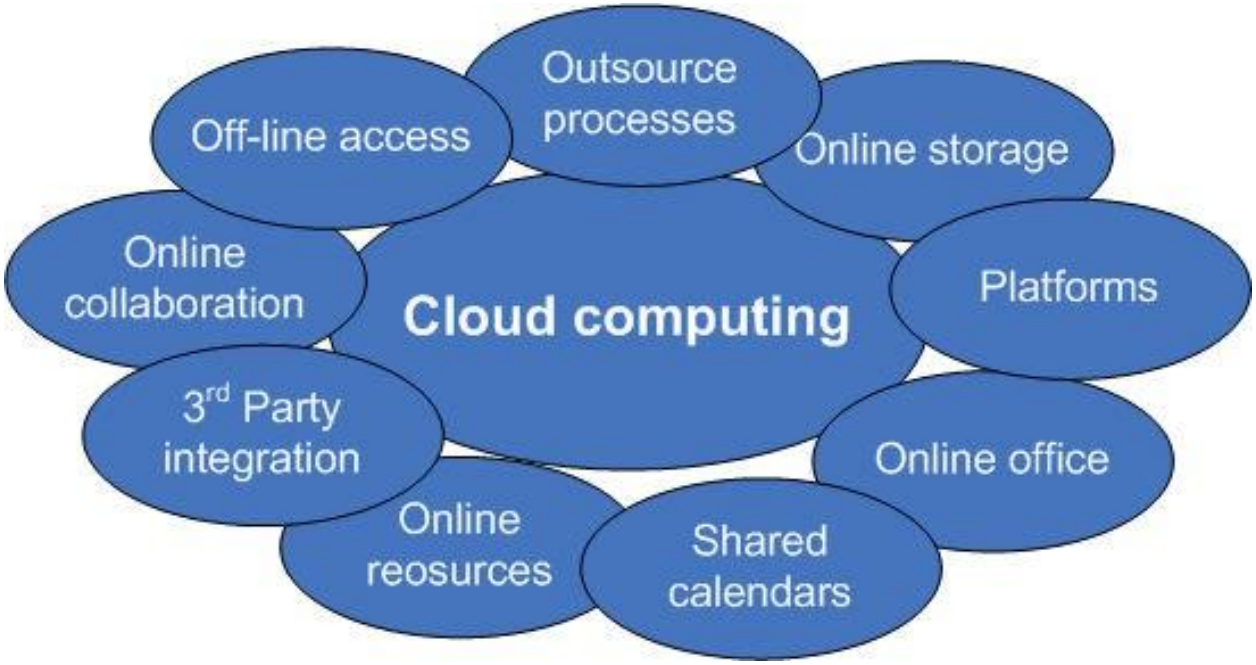


UNIT-V

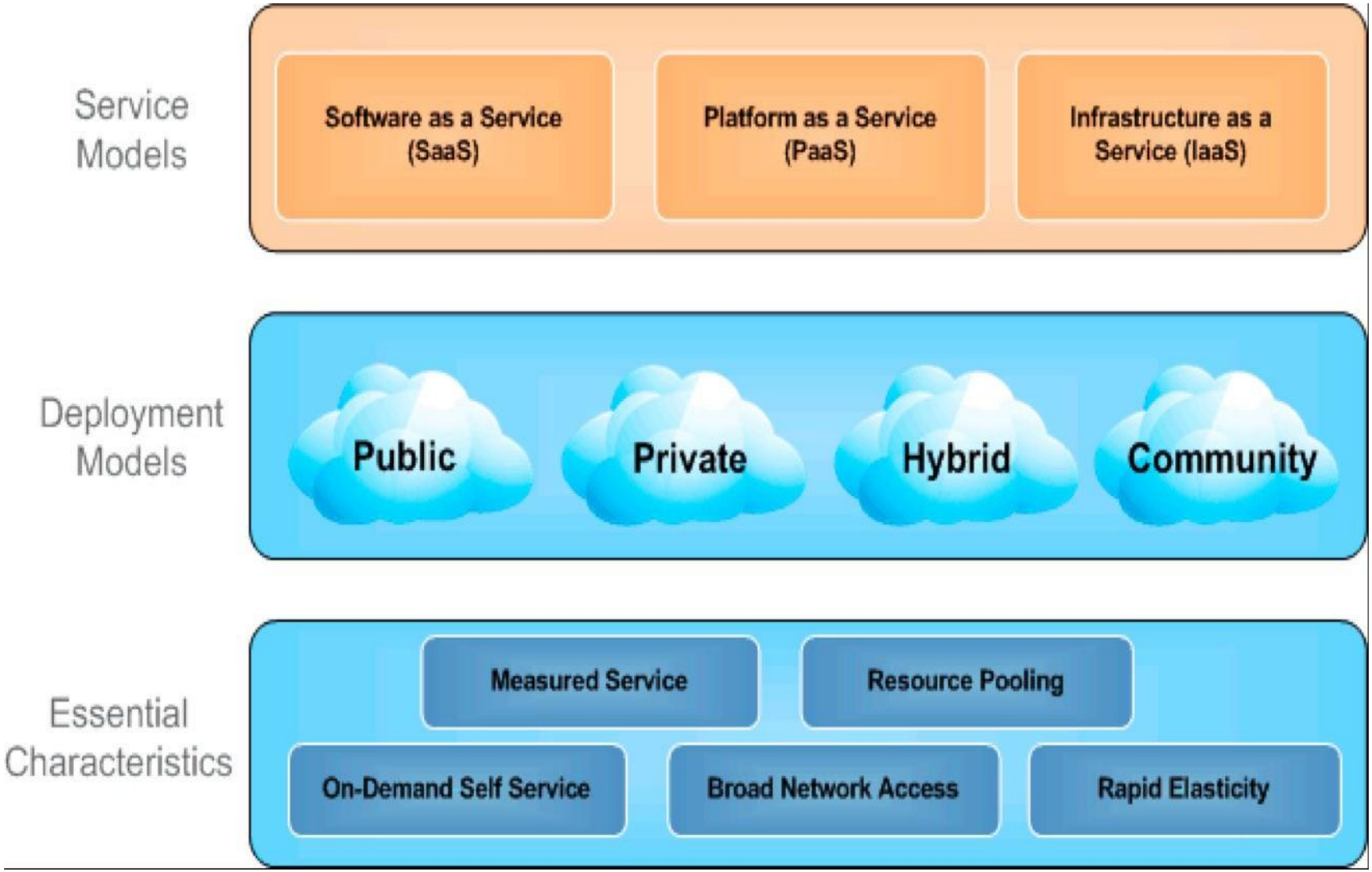
The Enterprise Cloud Computing Paradigm

Introduction

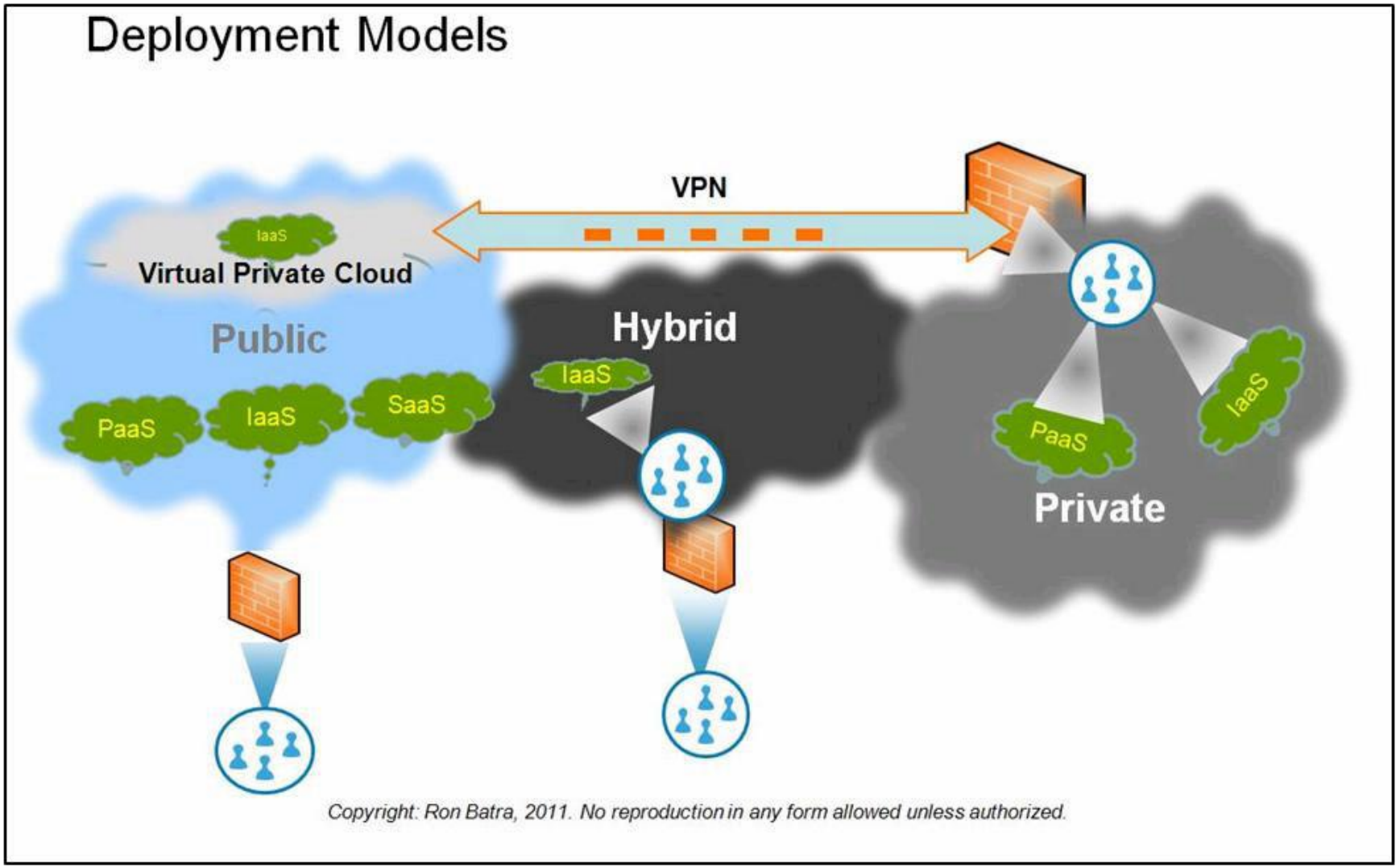
- *Enterprise Cloud Computing Paradigm*



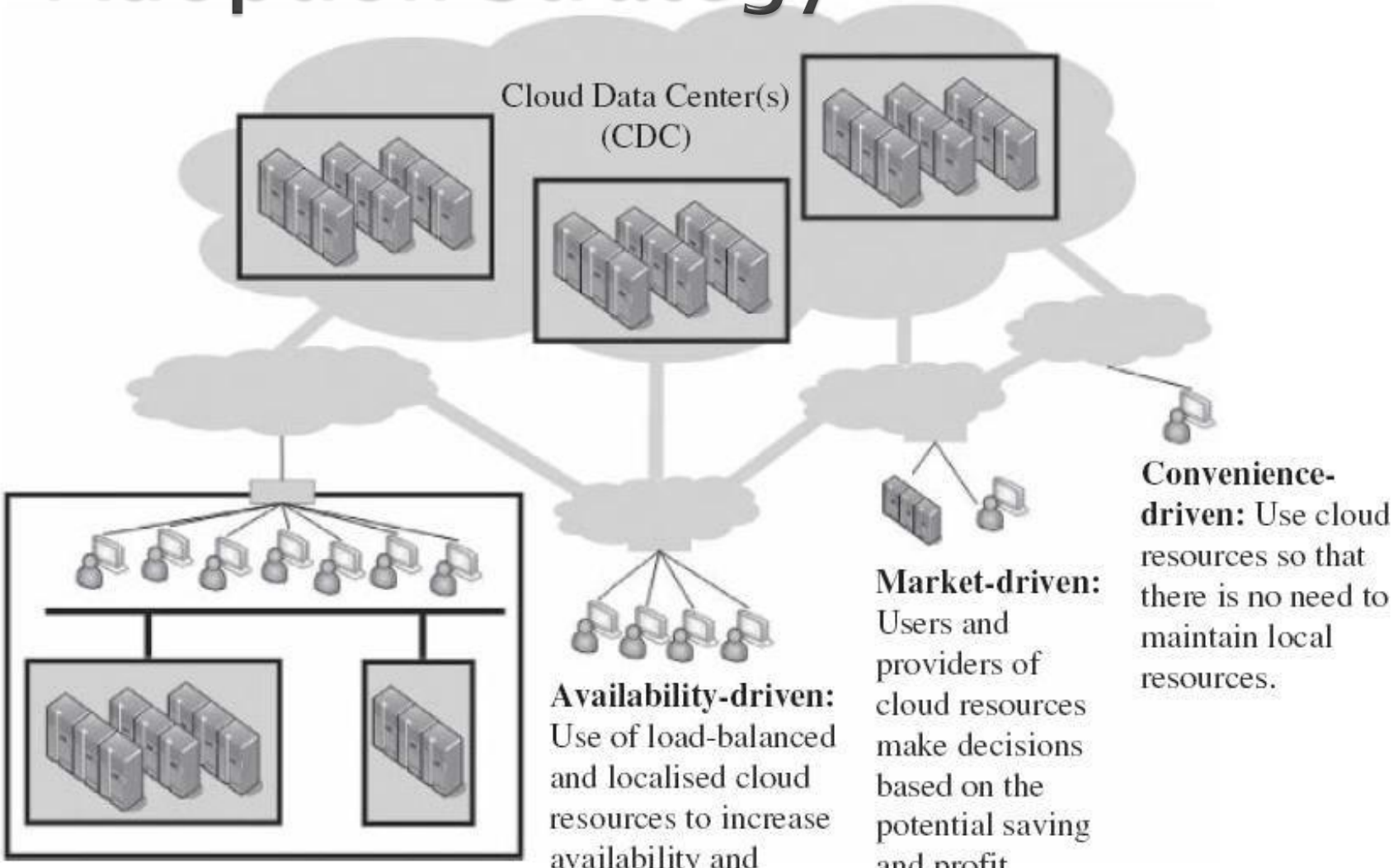
Deployment Models



Deployment Models



Adoption Strategy



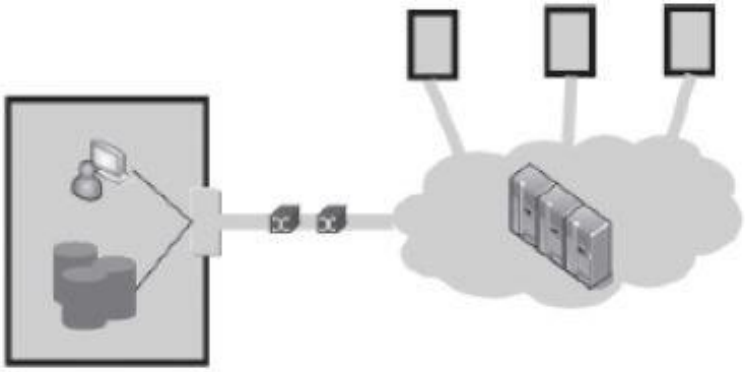
Scalability-driven: Use of cloud resources to support additional load or as back-up.

Availability-driven: Use of load-balanced and localised cloud resources to increase availability and reduce response time

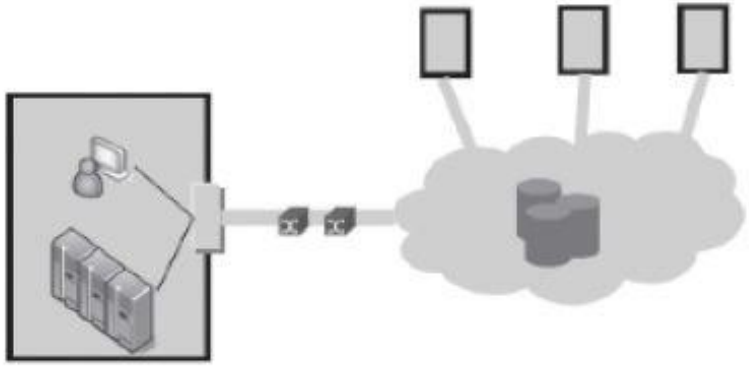
Market-driven: Users and providers of cloud resources make decisions based on the potential saving and profit

Convenience-driven: Use cloud resources so that there is no need to maintain local resources.

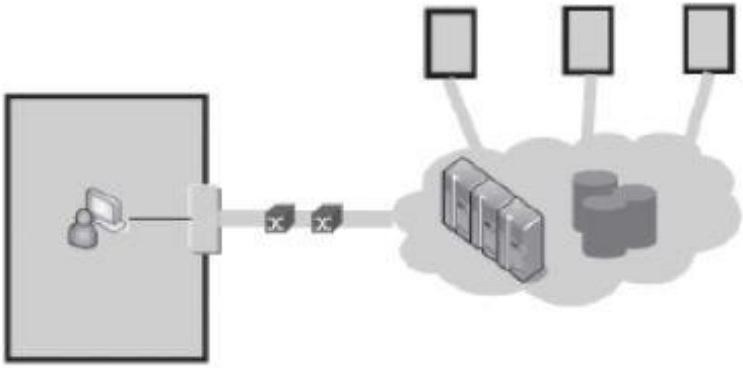
Consumption Strategy



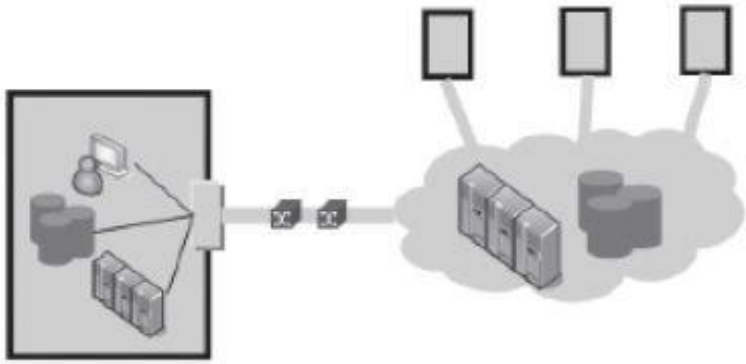
(1) Software Provision: Cloud provides instances of software but data is maintained within user's data center



(2) Storage Provision: Cloud provides data management and software accesses data remotely from user's data center



(3) Solution Provision: Software and storage are maintained in cloud and the user does not maintain a data center



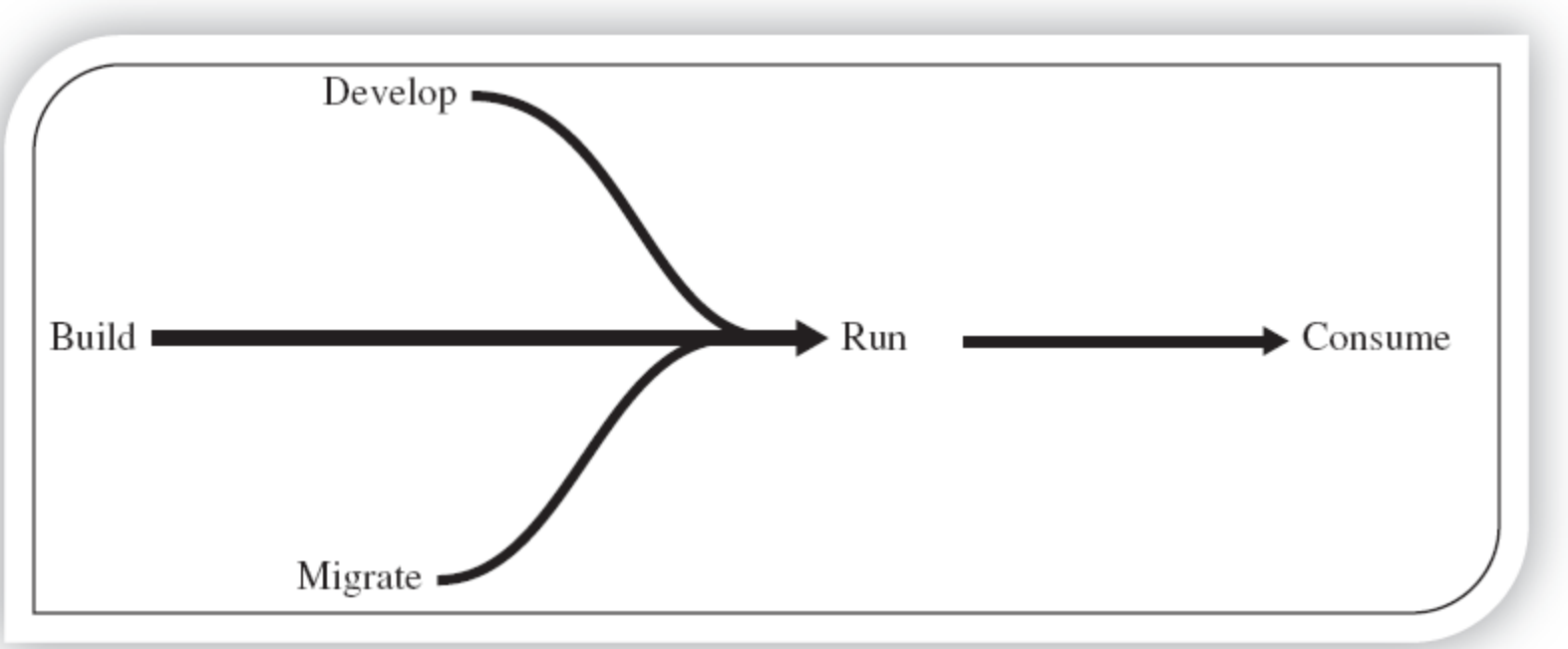
(4) Redundancy Services: Cloud is used as an alternative or extension of user's data center for software and storage

Issues for Enterprise Applications in cloud

- *ERP*
- *Capabilities of ERP*
  - *Transactional Capabilities*
    - *OLTP (Online Transaction Processing)*
    - *manage transaction oriented applications(relational databases)*
    - *ACID properties, write/update-intensive*
    - *CRM (Customer Relationship Management)*
  - *Analytical Capabilities*
    - *OLAP (Online Analytical Processing)*
    - *Analysis, reporting, decision support*
    - *Read only*
    - *Data-intensive*
    - *BI (Business Intelligence)*

(ERP) Transition Challenges

- *Five stage of the cloud*



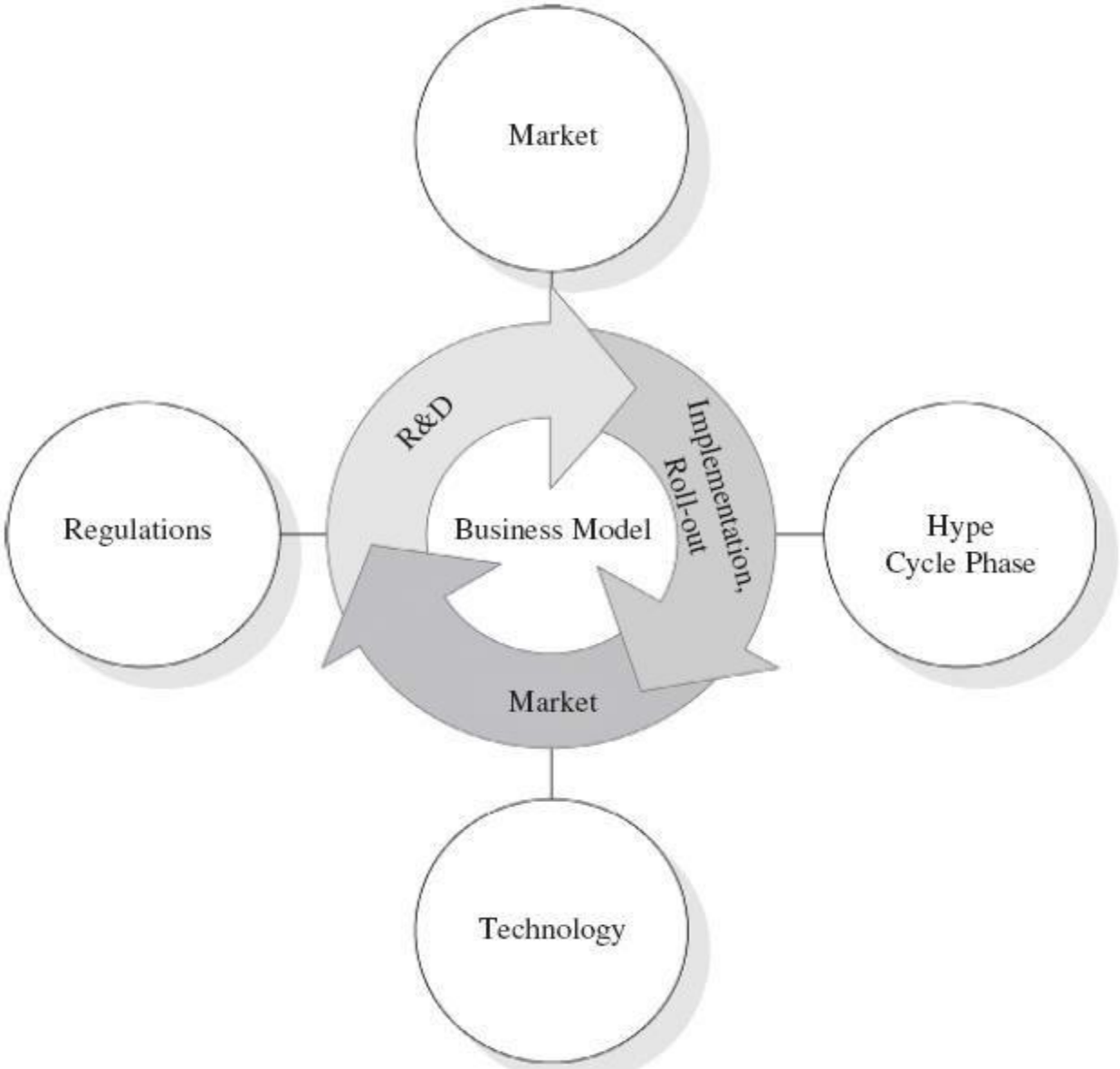
Enterprise Cloud Technology & Market Evolution

- *Standard*
  - *Vendor lock-in*
  - *OGF OCCI for compute clouds*
  - *SNIA CDMI for storage and data management*
  - *DMTF Virtualization Management (VMAN)*
  - *DMTF Cloud Incubator*
  - *Drives adoption, Drives the market, Third party vendor*
- *SLA*
  - *Lack of control*
  - *Primitive vs. Sophisticated*
- *Cloud Service Brokerage (CSB)*
  - *Cloud Service Intermediation*
  - *Aggregation*
  - *Cloud service Arbitrage*

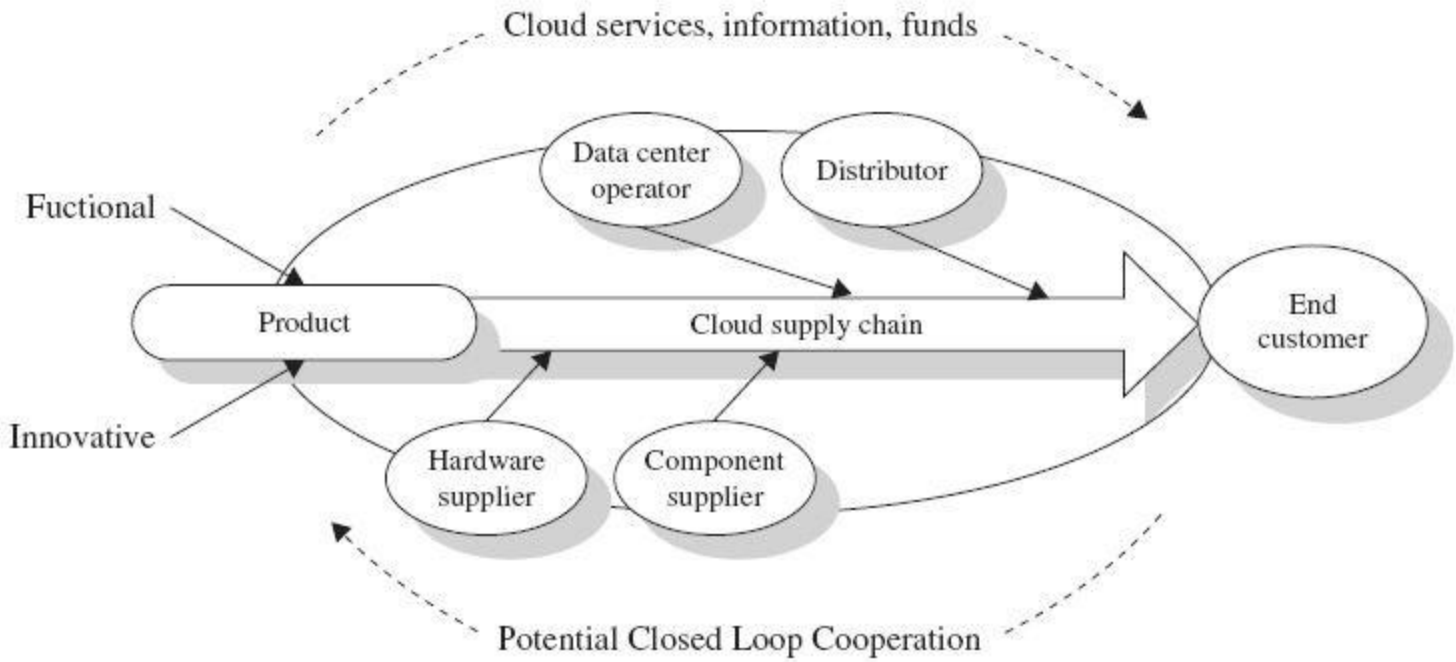
Marketplace

- *industry-specific characteristics*
  - *Rivalry*
  - *Comparable Products*
    - *Market share*
  - *Federation*
    - *Small companies*
  - *Switching cost*
    - *Standardization*

Dynamic Business Model



Cloud Supply Chain



ON THE MANAGEMENT OF VIRTUAL MACHINES FOR CLOUD INFRASTRUCTURES

IaaS Anatomy

- *IaaS provider characteristic*
  1. *on-demand provisioning of computational resources*
  2. *Virtualization technologies to lease resources*
  3. *Provide public and simple remote interfaces to manage resources*
  4. *use a pay-as-you-go cost model*
  5. "infinite capacity" or "unlimited elasticity"
- *Private and Public difference*
- *Role of Virtualization*
  - *Key of these characteristic*
  - *Allocating resources efficiently*
  - *Taking* into account an organization's goals
  - *Reacting to changes in the physical infrastructure*

IaaS Anatomy

- *Problems In VM Solutions*
  - *Distributed management of virtual machines*
  - *Reservation-based provisioning of virtualized resource*
  - *Provisioning to meet SLA commitments*
- *RESERVOIR project*
  - *Resources and Services Virtualization without Barriers*
  - *Addressed above problems*

Distributed Management

- Manage the virtual infrastructures themselves
- Efficiently selecting or scheduling
  - computational resources
- VM-based resource scheduling
  - Static approach
  - Efficiency approach
- Solution
  - Virtual Infrastructure Manager
    - *Managing VMs in a pool of distributed physical resources*
- Case Study
  - OpenNebula

VM Model and Life Cycle (OpenNebula)

- *VM model attributes*
  - *A capacity in terms of memory and CPU*
  - *A set of NICs attached to one or more virtual networks*
  - *A set of disk images*
  - *A state file (optional) or recovery file*

- *Life Cycle*
  - *Resource Selection*
  - *Resource Preparation*
    - *Contextualization*
  - *VM Creation*
  - *VM Migration*
  - *VM Termination*