Cloud Computing & its Applications

Course Code: SWE4004

Dr Sunil Kumar Singh
Assistant Professor
School - SCOPE
VIT-AP University
sunil.singh@vitap.ac.in
Cabin - AB2 (124D)

Cloud Infrastructure Mechanisms

Outline

- Network Perimeter
- Virtual Server
- Cloud Storage Device
- Cloud Usage Monitor
- Resource Replication

Network Perimeter

The isolation of a network environment from the rest of communications network

- The logical network perimeter establishes a virtual network boundary that can encompass and isolate a group of related cloud-based IT resources that may be physically distributed.

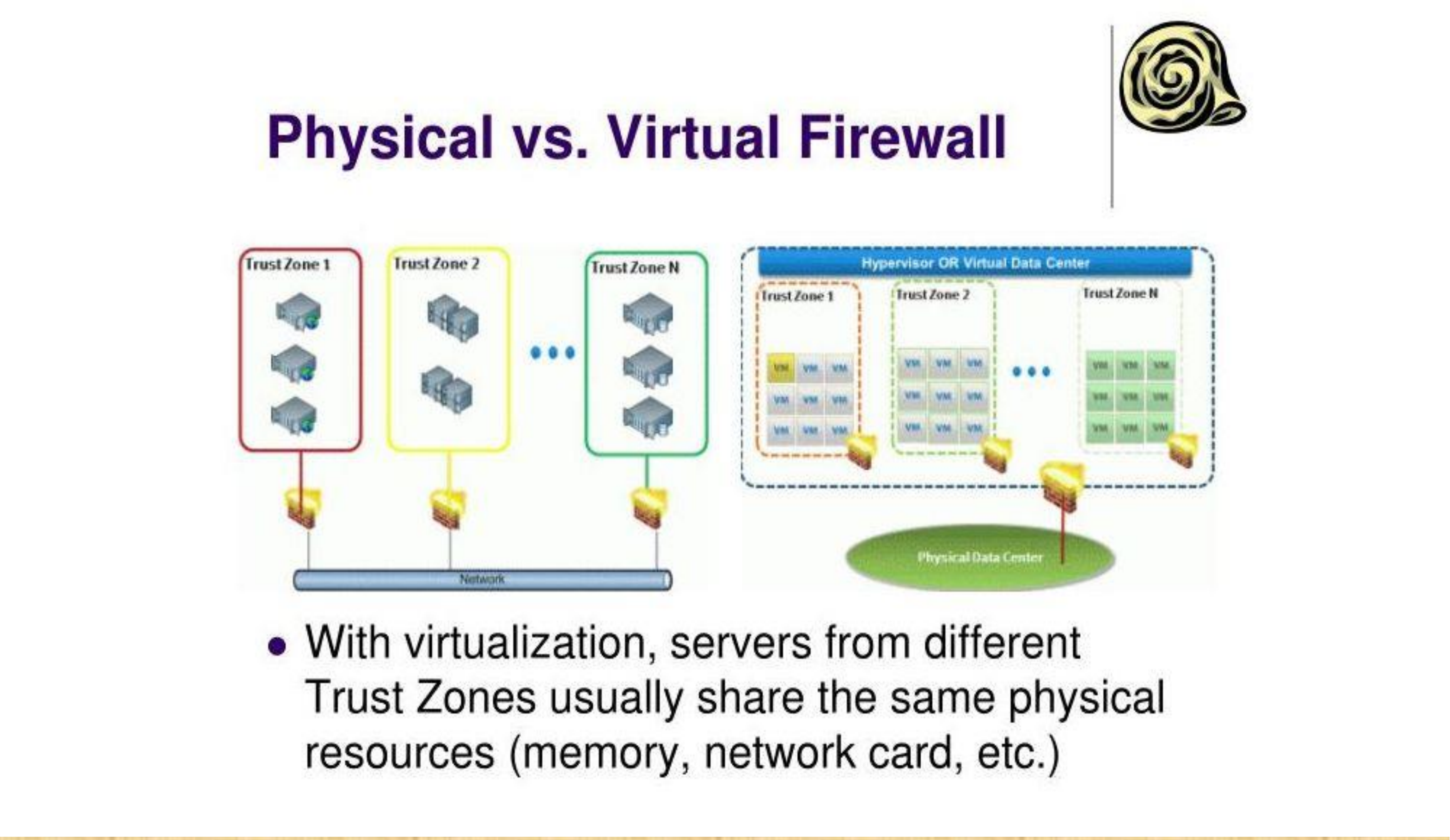
Logical network perimeter

Logical network perimeter can be implement to

- Isolate IT resources in a cloud from non-authorized users
- Isolate IT resources in a cloud from cloud users
- Control the bandwidth via network devices by deploying virtual firewall and virtual network

Virtualized IT environment

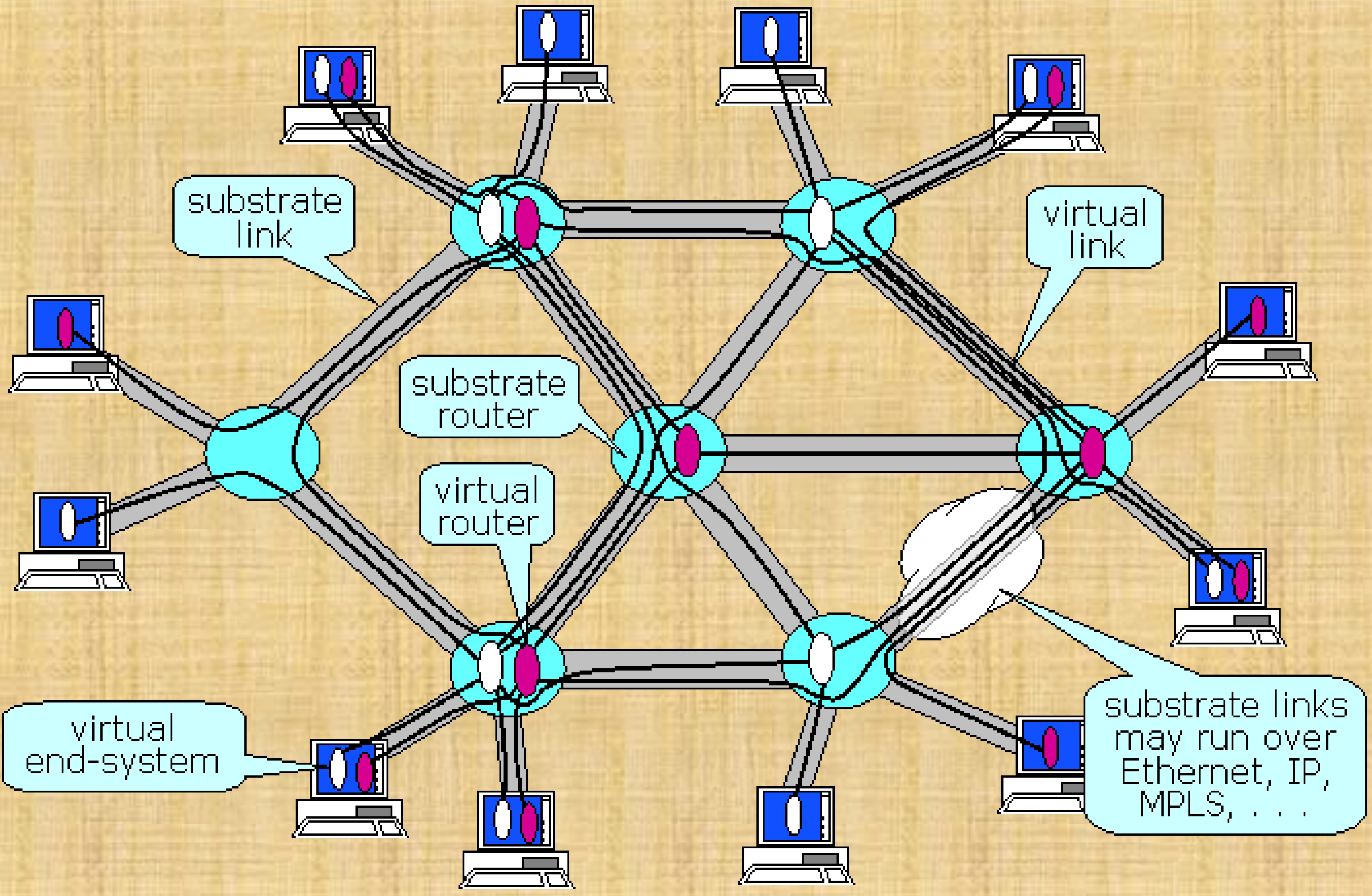
Virtual Firewall



Physical vs. Virtual Firewall

- With virtualization, servers from different Trust Zones usually share the same physical resources (memory, network card, etc.)

Virtual network



Virtual IT Resources

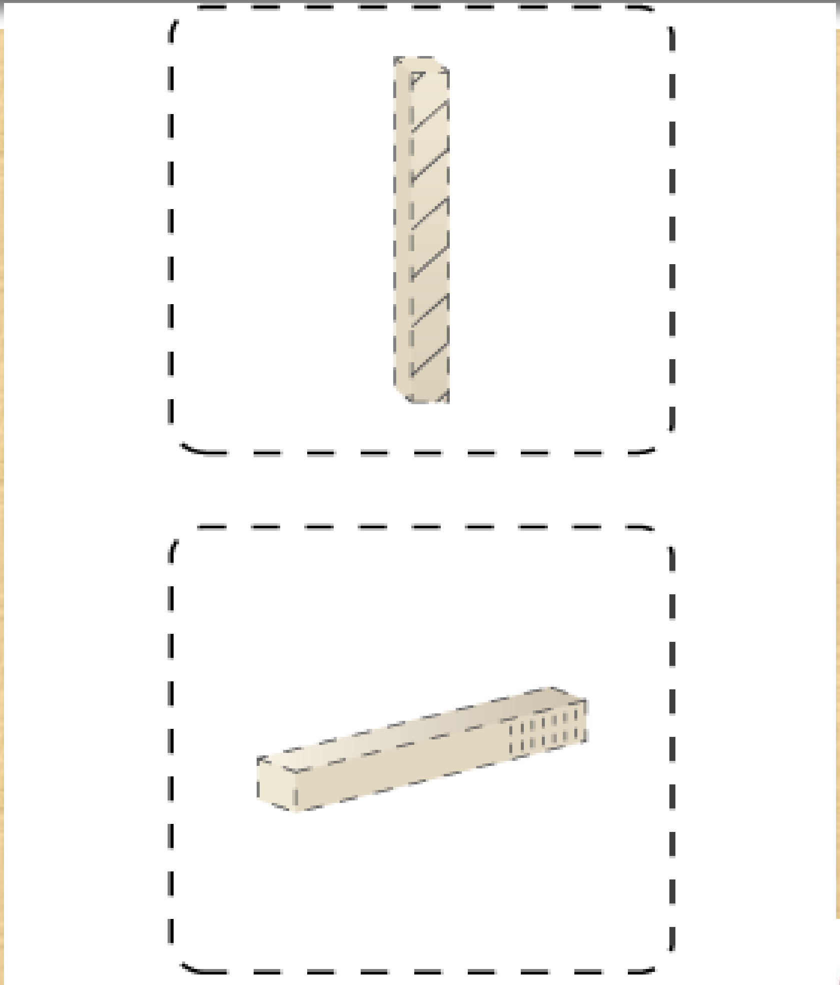
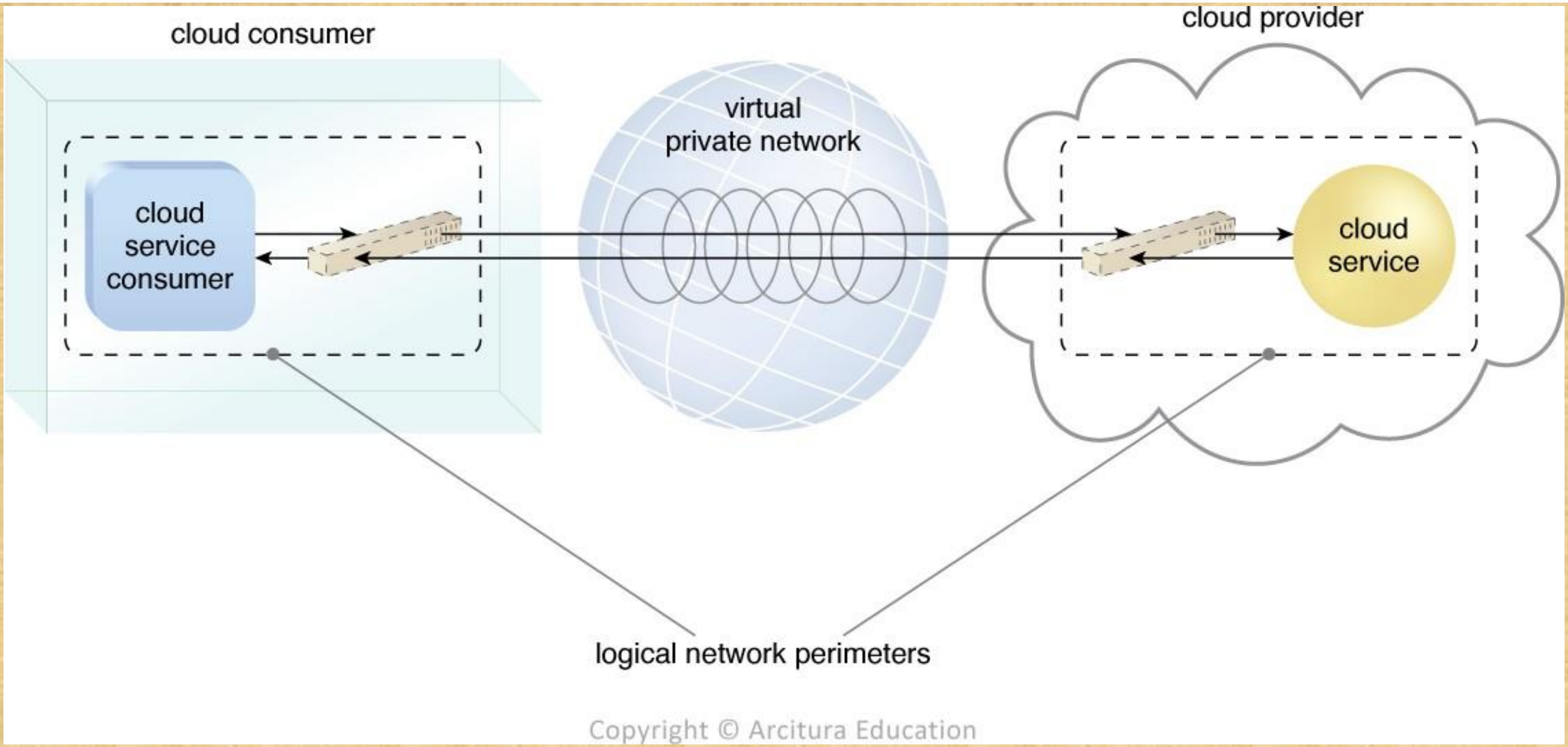
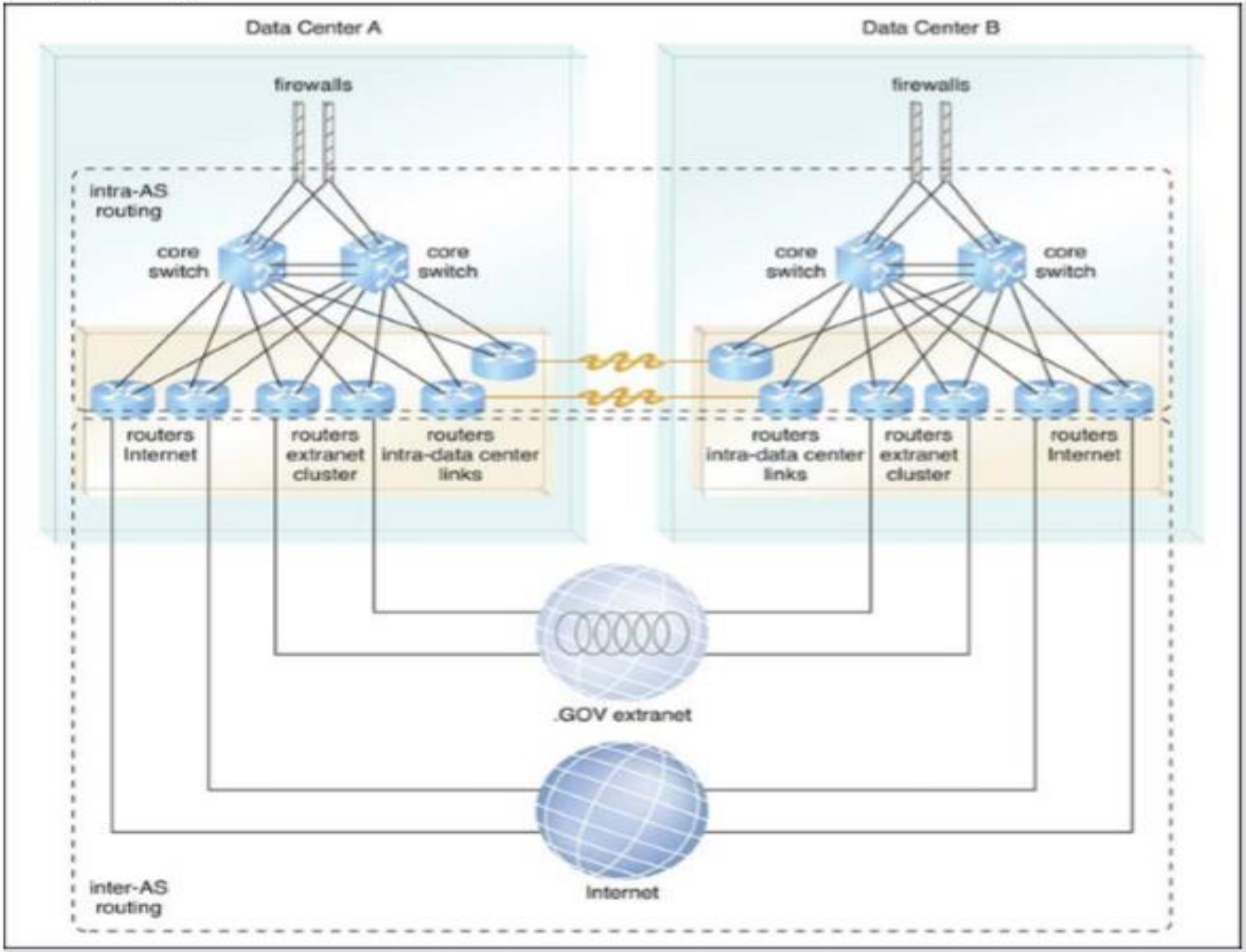


Figure 7.2 Virtual firewall (top)and virtual network. (bottom)

Network Perimeter

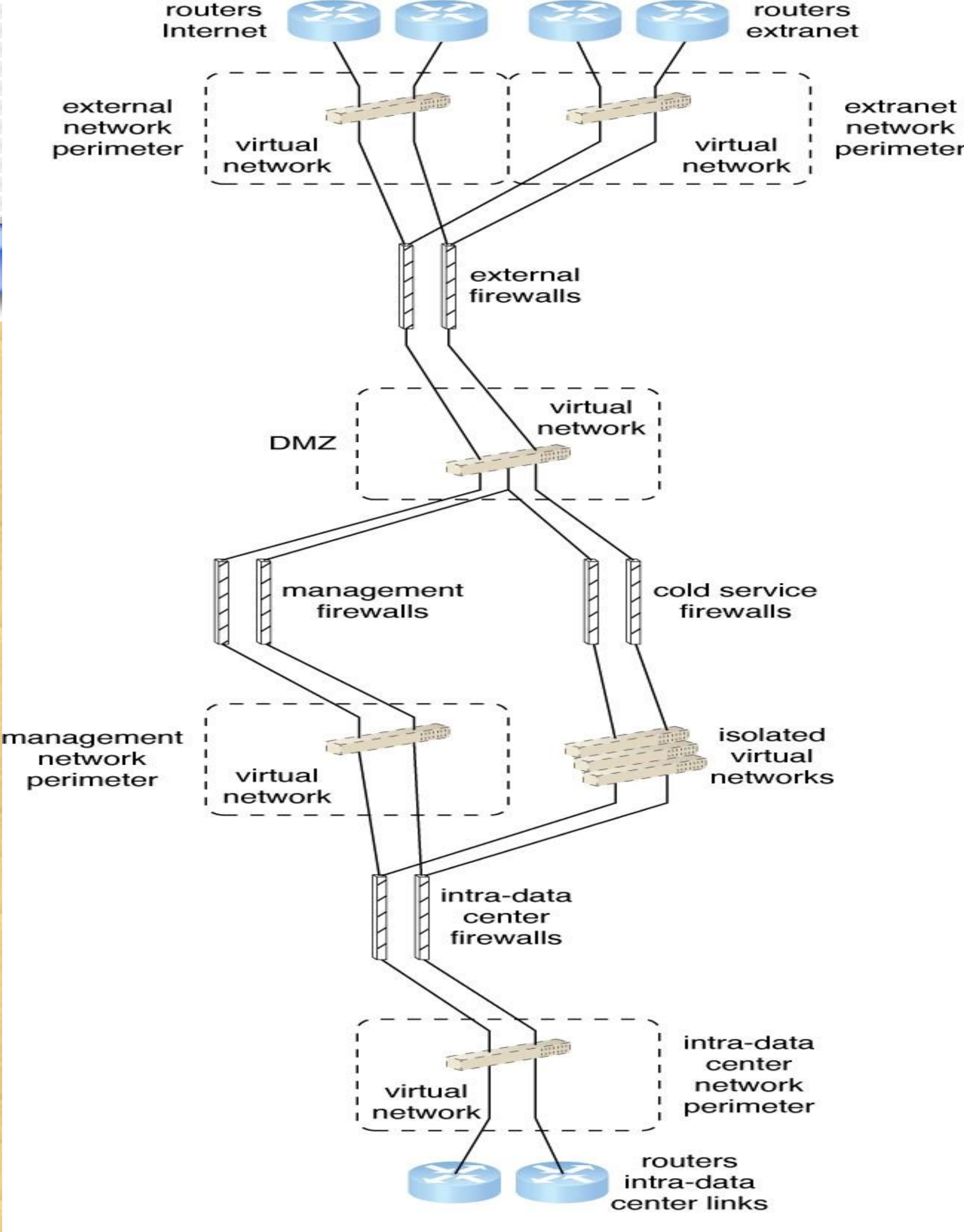


Two logical network perimeters surround the cloud consumer and cloud provider environments.



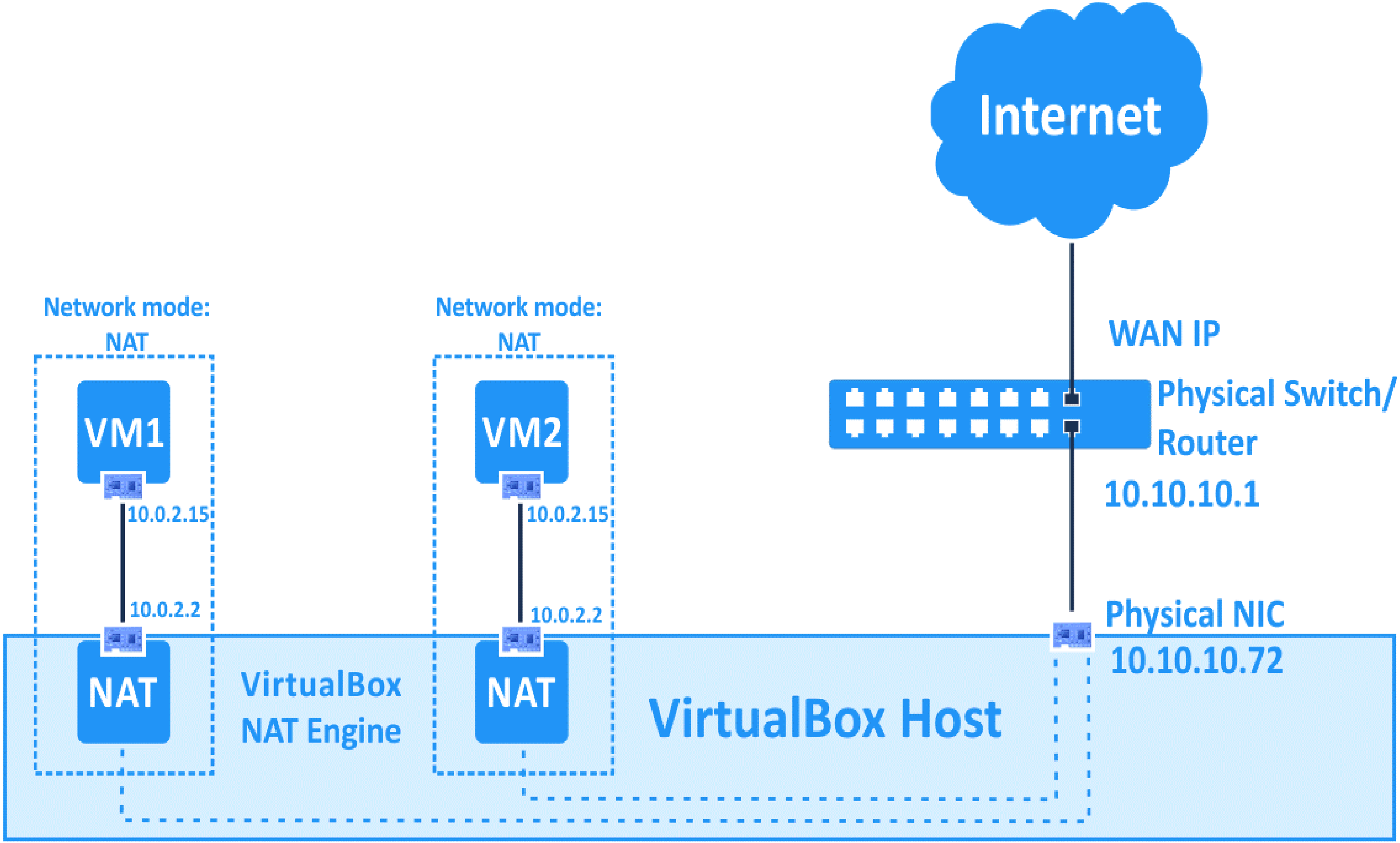
Data Center A
Data Center B
firewalls
firewalls
intra-AS routing
core switch
core switch
core switch
core switch
routers Internet
routers extranet cluster
routers intra-data center links
routers intra-data center links
routers extranet cluster
routers Internet
.GOV extranet
Internet
inter-AS routing

logical network layout is established through a set of logical network perimeters using various firewalls and virtual networks.

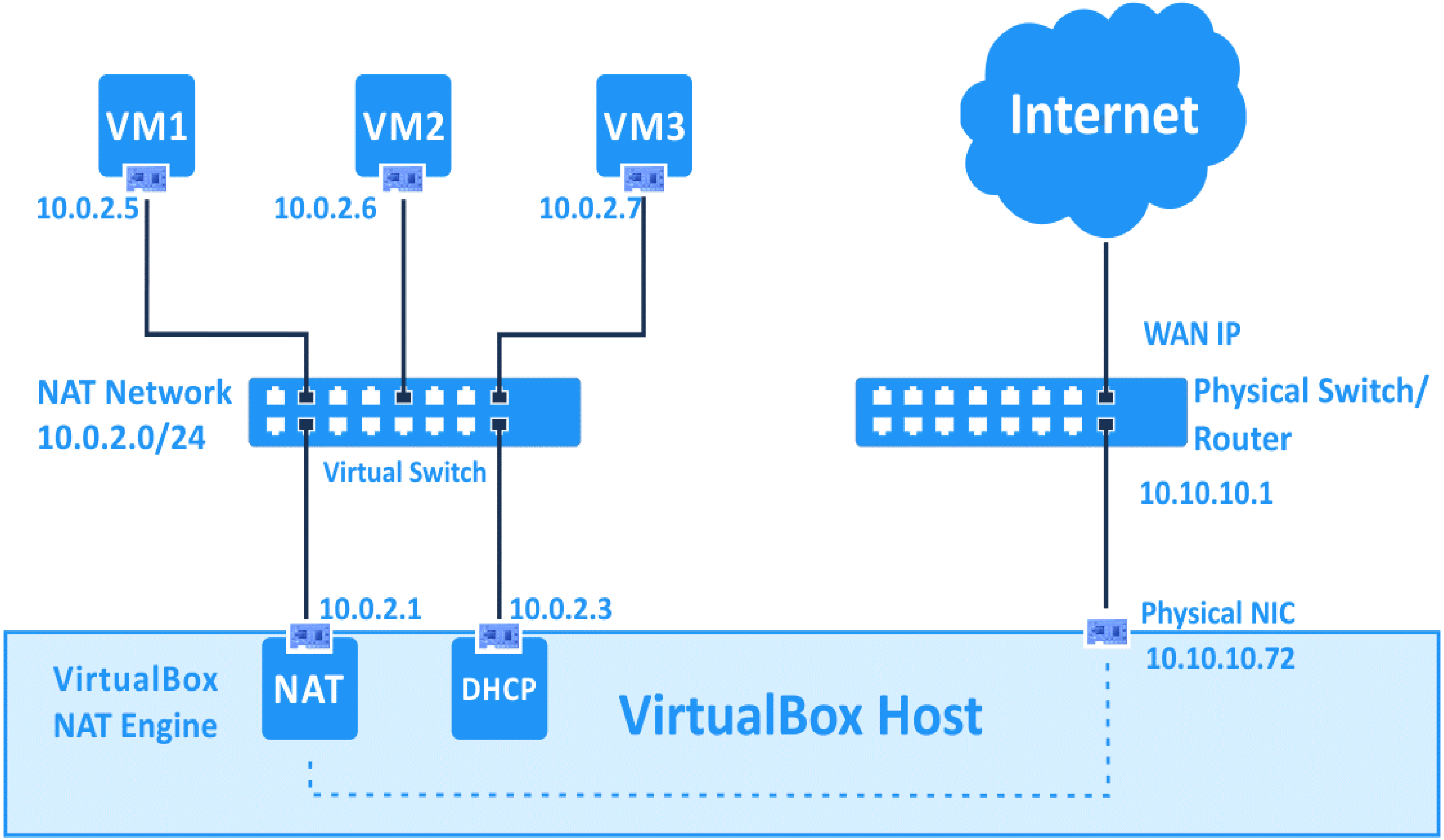


Copyright © Arcitura Education

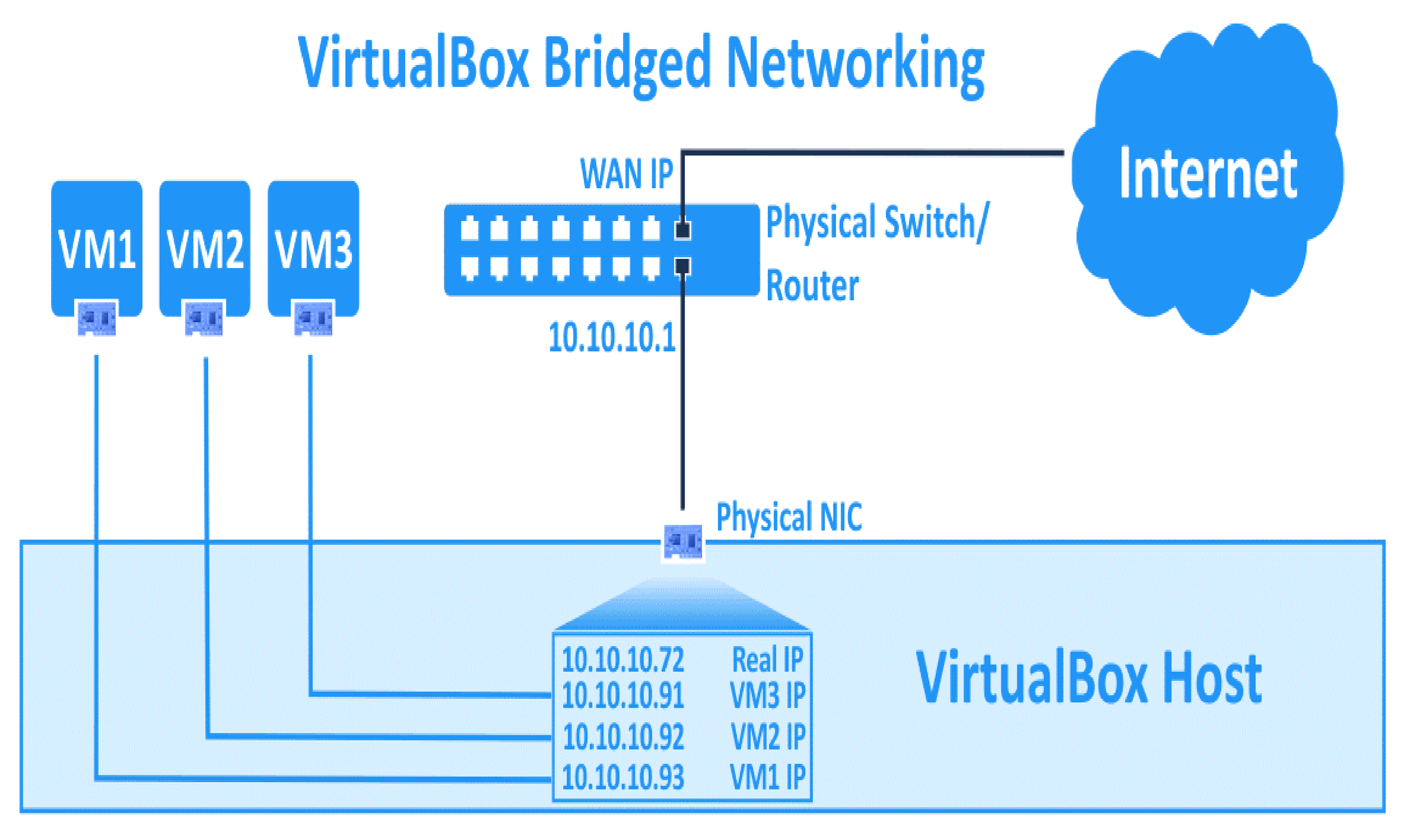
NAT


Internet
WAN IP
Physical Switch/
Router
10.10.10.1
Network mode:
NAT
VM1
10.0.2.15
10.0.2.2
NAT
Network mode:
NAT
VM2
10.0.2.15
10.0.2.2
NAT
VirtualBox
NAT Engine
VirtualBox Host
Physical NIC
10.10.10.72

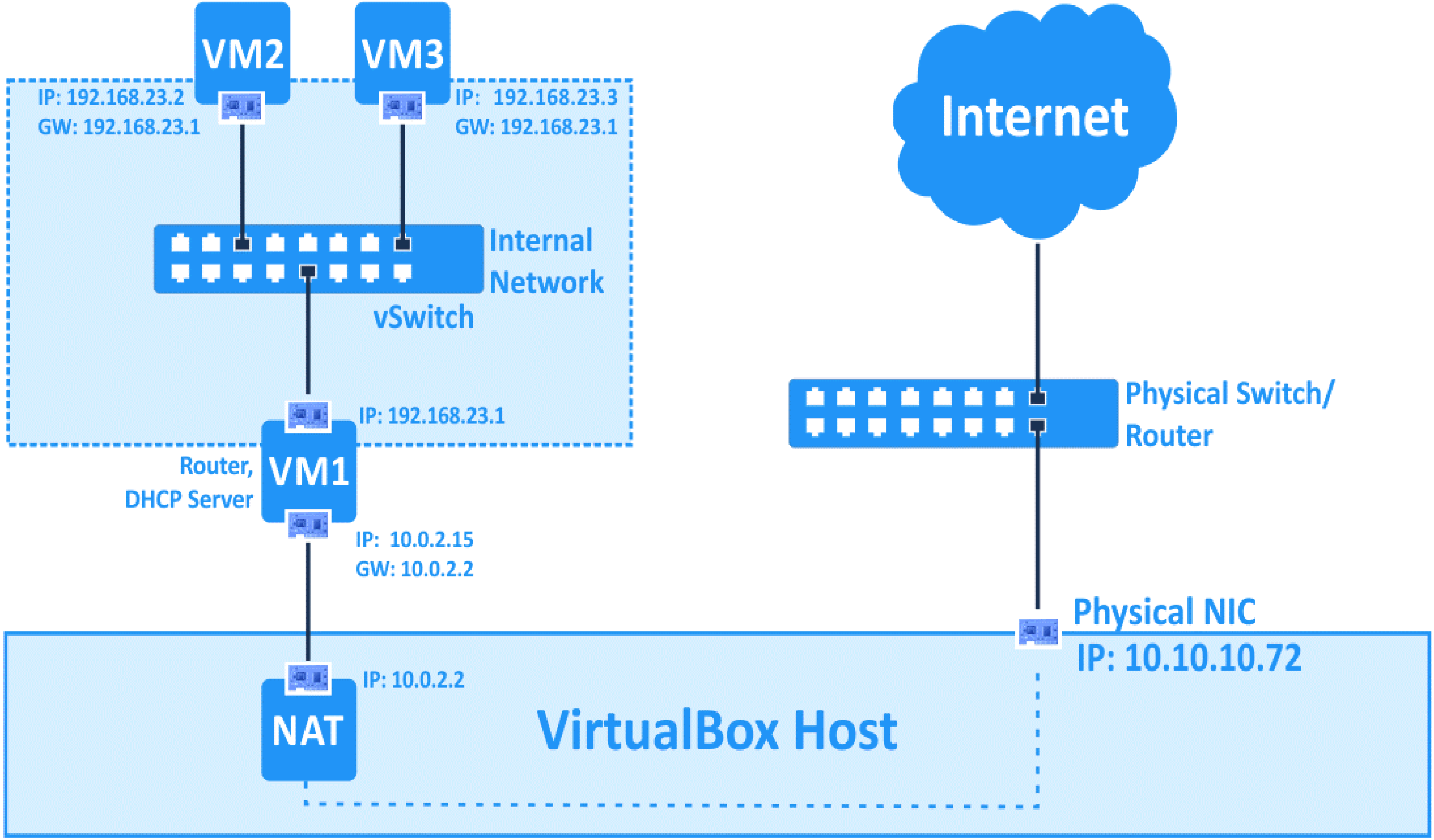
NAT Network

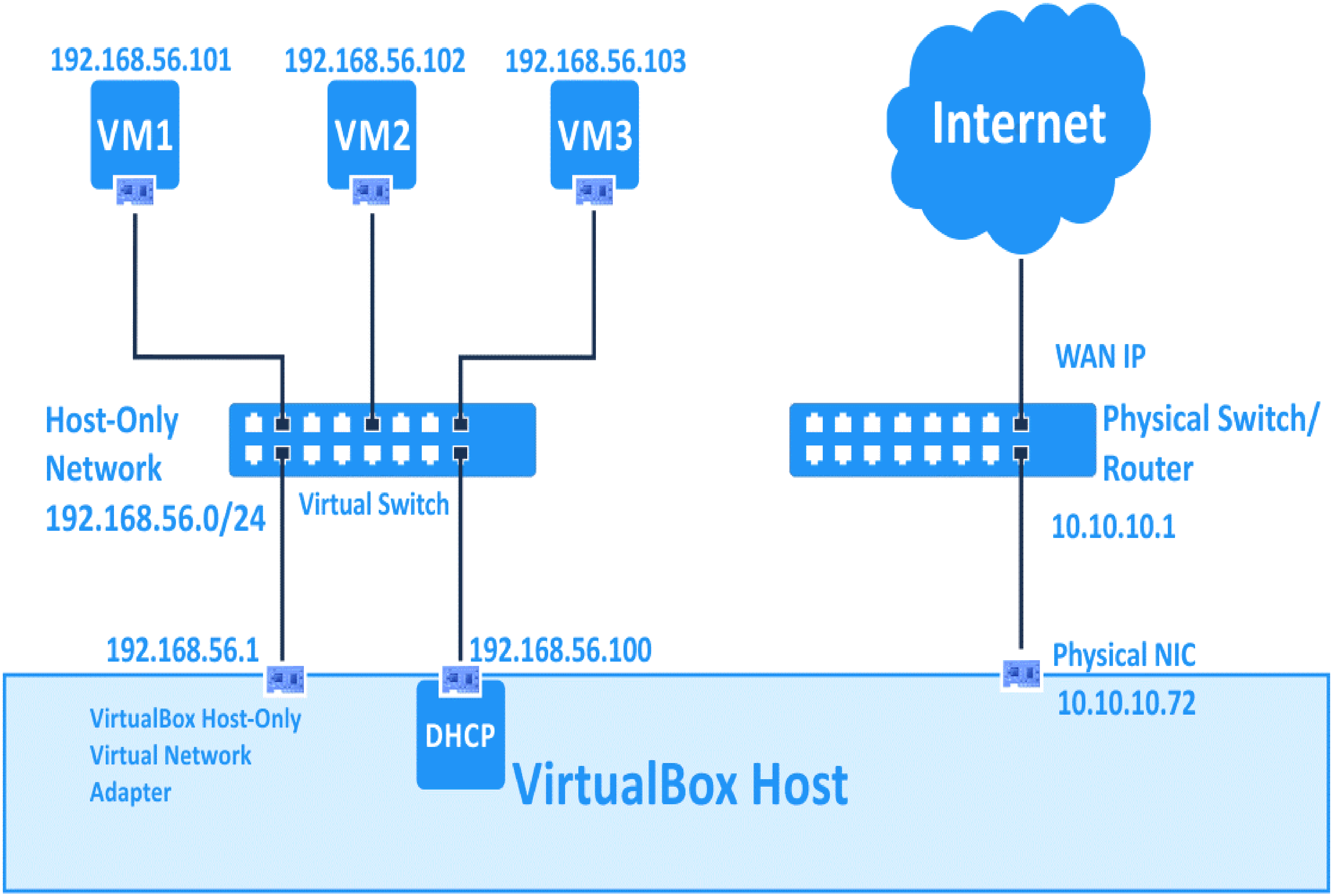

VM1
VM2
VM3
10.0.2.5
10.0.2.6
10.0.2.7
Internet
WAN IP
NAT Network
10.0.2.0/24
Virtual Switch
Physical Switch/
Router
10.10.10.1
10.0.2.1
10.0.2.3
Physical NIC
10.10.10.72
VirtualBox
NAT Engine
NAT
DHCP
VirtualBox Host

Virtual Network



Virtual Network




192.168.56.101
192.168.56.102
192.168.56.103
VM1
VM2
VM3
Internet
WAN IP
Host-Only
Network
192.168.56.0/24
Virtual Switch
Physical Switch/
Router
10.10.10.1
192.168.56.1
192.168.56.100
Physical NIC
10.10.10.72
VirtualBox Host-Only
Virtual Network
Adapter
DHCP
VirtualBox Host

Virtual Server

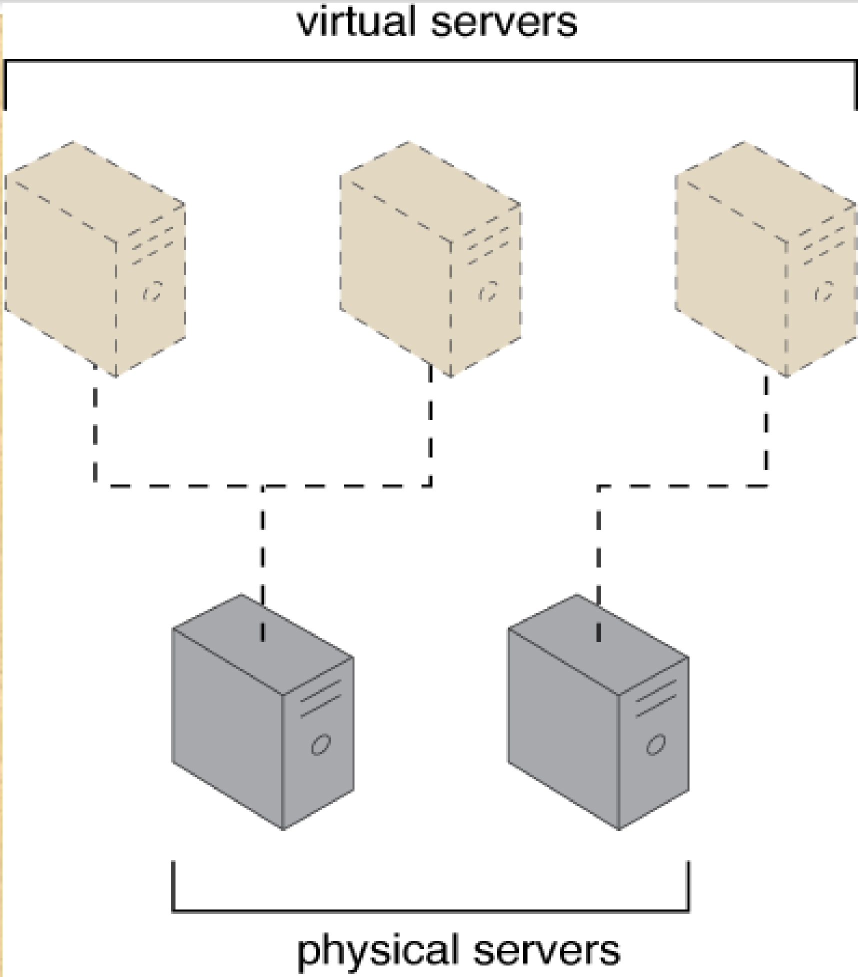
- A virtual server is a form of virtualization software that emulates a physical service.

- The virtual server represents the mode fundamental building block of cloud environment. The instantiation of virtual servers from image files is a resource allocation process that can be completed rapidly and on-demand.

- Cloud customers that install or lease virtual servers can customize their environments independently from other customers.

Virual Server (Cont..)

The first physical server hosts two virtual servers, while the second physical server hosts one virtual server.

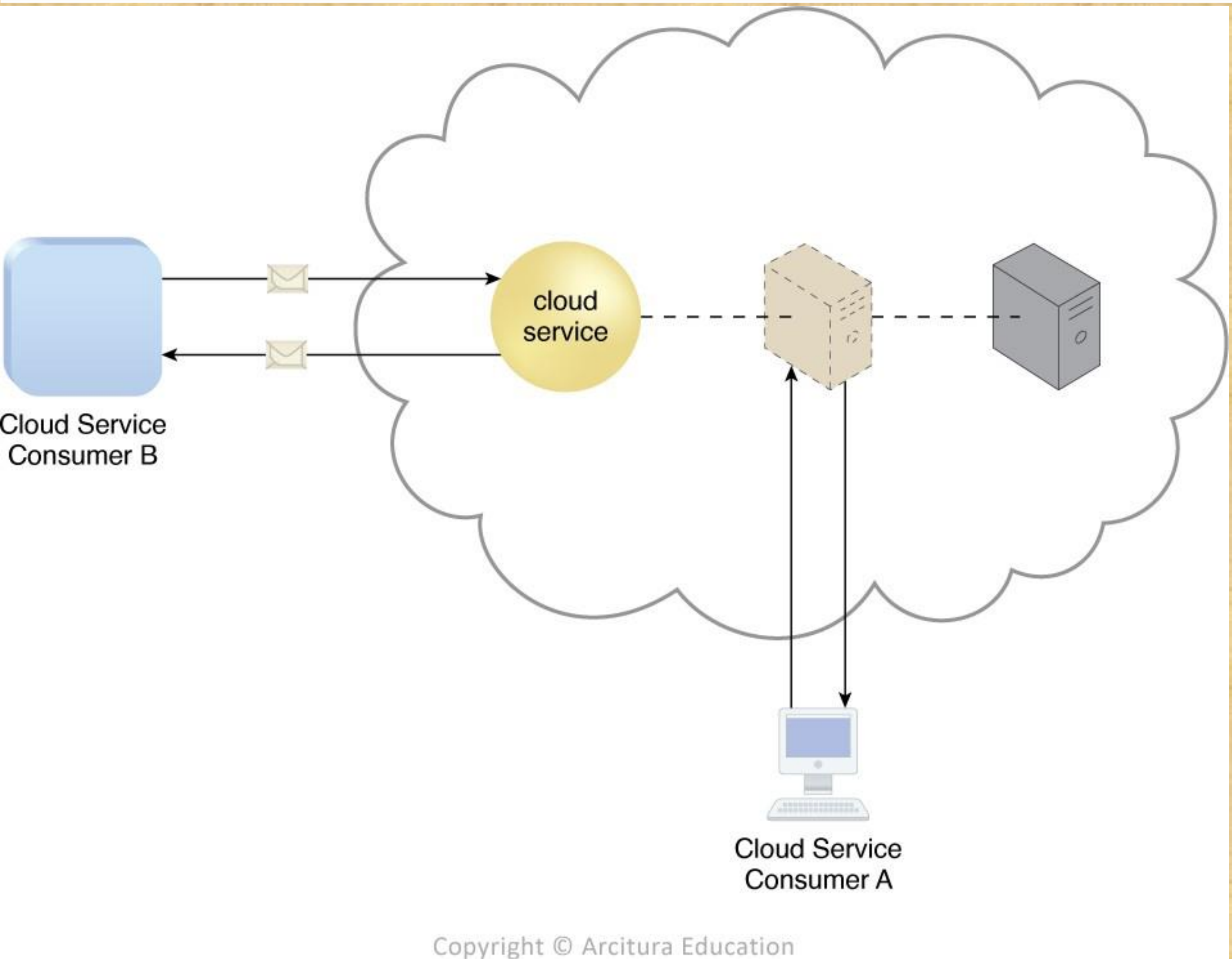


Copyright © Arcitura Education

Virtual Server (Cont..)

- Each virtual server can host numerous IT resources, cloud-based solutions, and various other cloud computing mechanisms.
- The instantiation of virtual servers from image files is a resource allocation process that can be completed rapidly and on-demand.
- Cloud consumers that install or lease virtual servers can customize their environments independently from other cloud consumers that may be using virtual servers hosted by the same underlying physical server.

Virtual Server (Cont..)



virtual server hosts an active cloud service and is further accessed by a cloud consumer for administrative purposes.

Case Study Example DTGOV's IaaS environment

- DTGOV's IaaS environment contains hosted virtual servers that were instantiated on physical servers running the same hypervisor software that controls the virtual servers.
- Their VIM is used to coordinate the physical servers in relation to the creation of virtual server instances.
- This approach is used at each data center to apply a uniform implementation of the virtualization layer.

Case Study Example DTGOV's IaaS environment

- In order to enable the on-demand creation of virtual servers, DTGOV provides cloud consumers with a set of template virtual servers that are made available through pre-made VM images.

- These VM images are files that represent the virtual disk images used by the hypervisor to boot the virtual server. DTGOV enables the template virtual servers to have various initial configuration options that differ, based on operating system, drivers, and management tools being used.

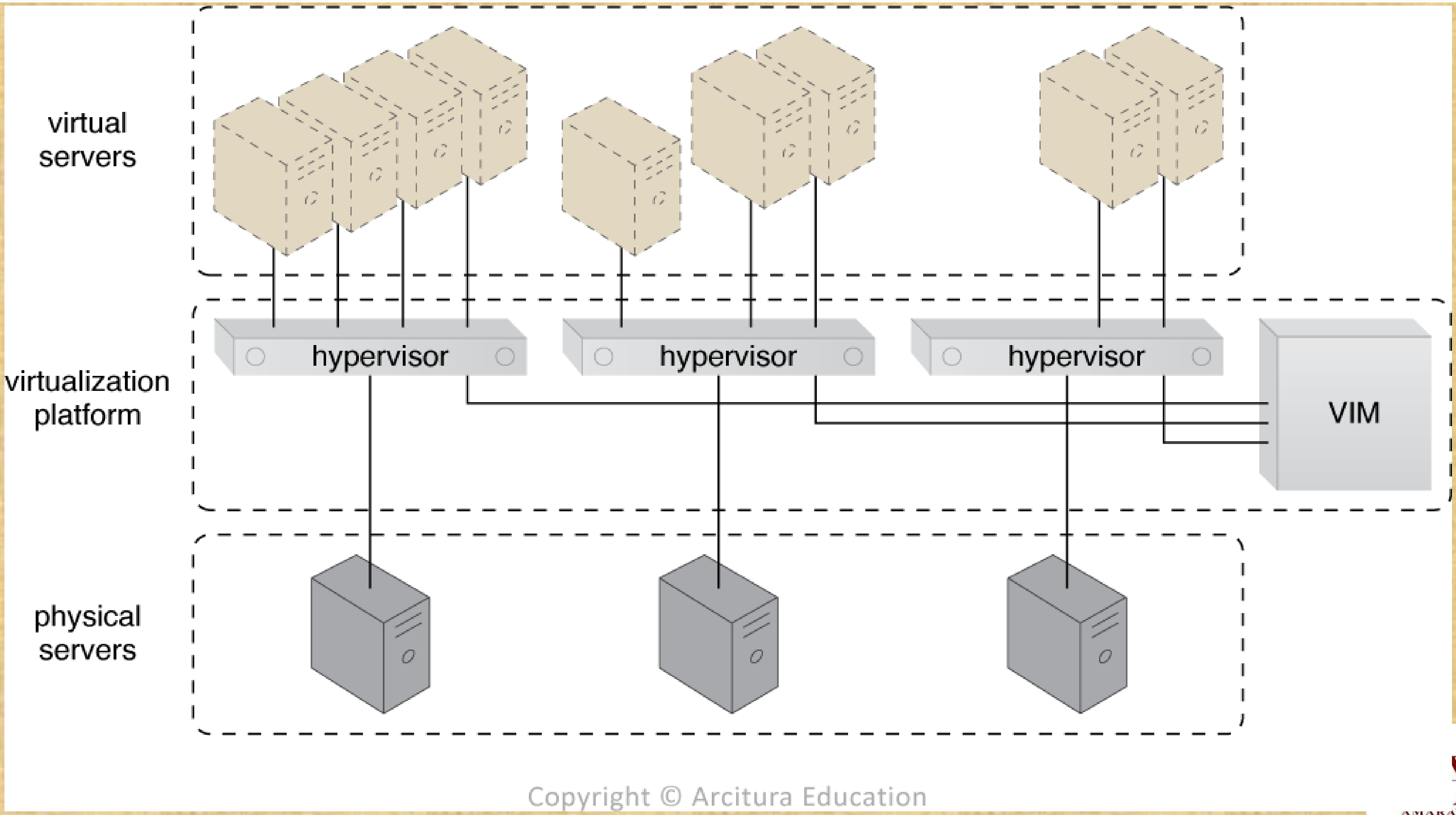
- Some template virtual servers also have additional, pre-installed application server software.

The following virtual server packages are offered to DTGOV's cloud consumers. Each package has different pre-defined performance configurations and limitations:

Case Study Example DTGOV's IaaS environment

- Small Virtual Server Instance – 1 virtual processor core, 4 GB of virtual RAM, 20 GB of storage space in the root file system
- Medium Virtual Server Instance – 2 virtual processor cores, 8 GB of virtual RAM, 20 GB of storage space in the root file system
- Large Virtual Server Instance – 8 virtual processor cores, 16 GB of virtual RAM, 20 GB of storage space in the root file system
- Memory Large Virtual Server Instance – 8 virtual processor cores, 64 GB of virtual RAM, 20 GB of storage space in the root file system
- Processor Large Virtual Server Instance – 32 virtual processor cores, 16 GB of virtual RAM, 20 GB of storage space in the root file system
- Ultra-Large Virtual Server Instance – 128 virtual processor cores, 512 G virtual RAM, 40 GB of storage space in the root file system

Case Study Example DTGOV's IaaS environment



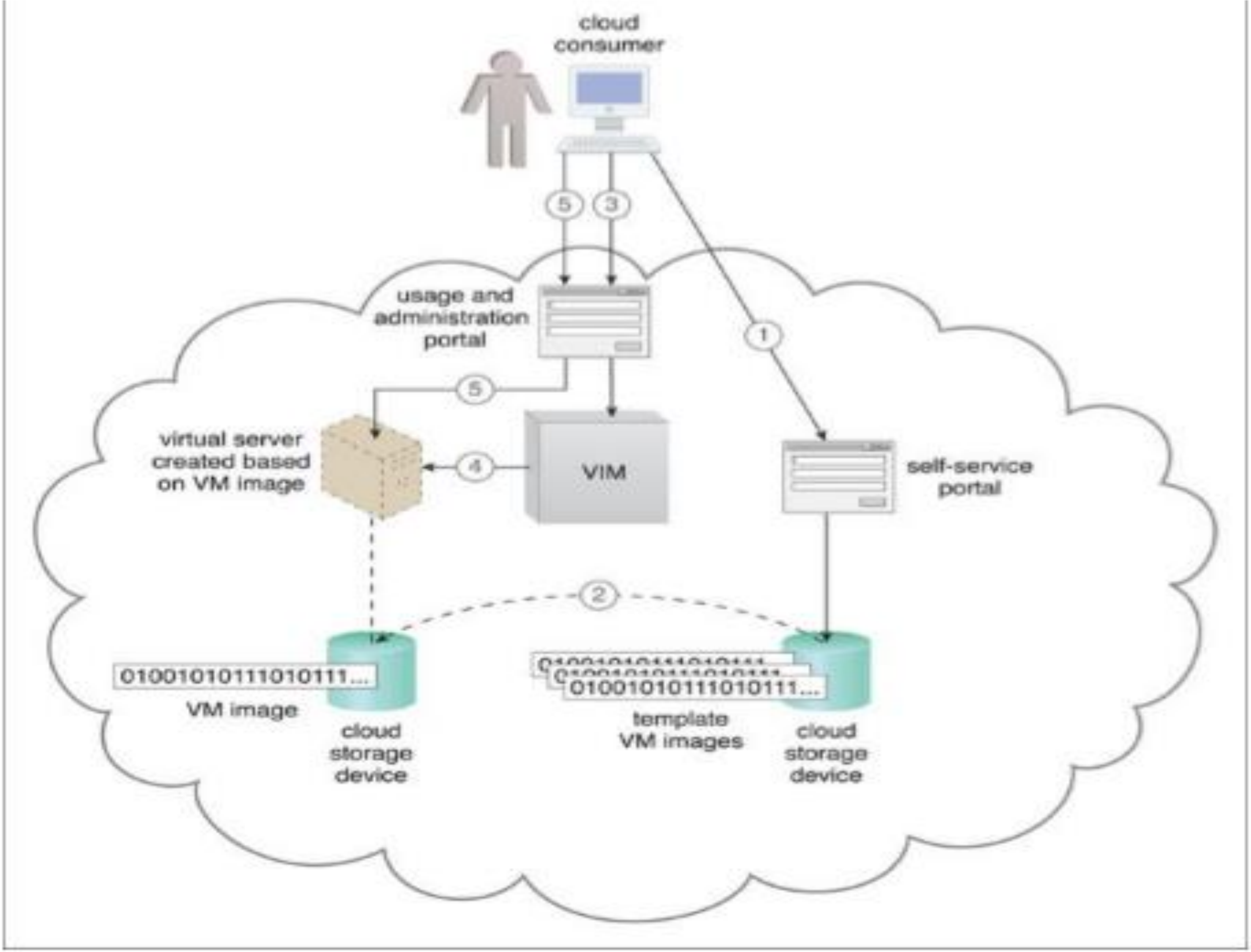
Virtual servers are created via the hypervisor and VIM.

Case Study Example DTGOV's IaaS environment

- Additional storage capacity can be added to a virtual server by attaching a virtual disk from a cloud storage device. All of the template virtual machine images are stored on a common cloud storage device that is accessible only through the cloud consumers' management tools that are used to control the deployed IT resources.

- Once a new virtual server needs to be instantiated, the cloud consumer can choose the most suitable virtual server template from the list of available configurations. A copy of the virtual machine image is made and allocated to the cloud consumer, who can then assume the administrative responsibilities.

- The allocated VM image is updated whenever the cloud consumer customizes the virtual server. After the cloud consumer initiates the virtual server, the allocated VM image and its associated performance profile is passed to the VIM, which creates the virtual server instance from the appropriate physical server. DTGOV uses the process described in Figure 7.8 to support the creation and management of virtual servers that have different initial software configurations and per characteristics


cloud consumer
5
3
usage and administration portal
1
5
virtual server created based on VM image
4
VIM
self-service portal
2
01001010111010111...
VM image
cloud storage device
01001010111010111...
template VM images
cloud storage device

Cloud Storage Device

- Storage devices that are designed specifically for cloud-based provisioning.

- Provides remote access via cloud storage services.

A primary concern related to cloud storage is

- Availability
- Integrity
- confidentiality of data

Cloud Storage Levels

Cloud storage device mechanisms

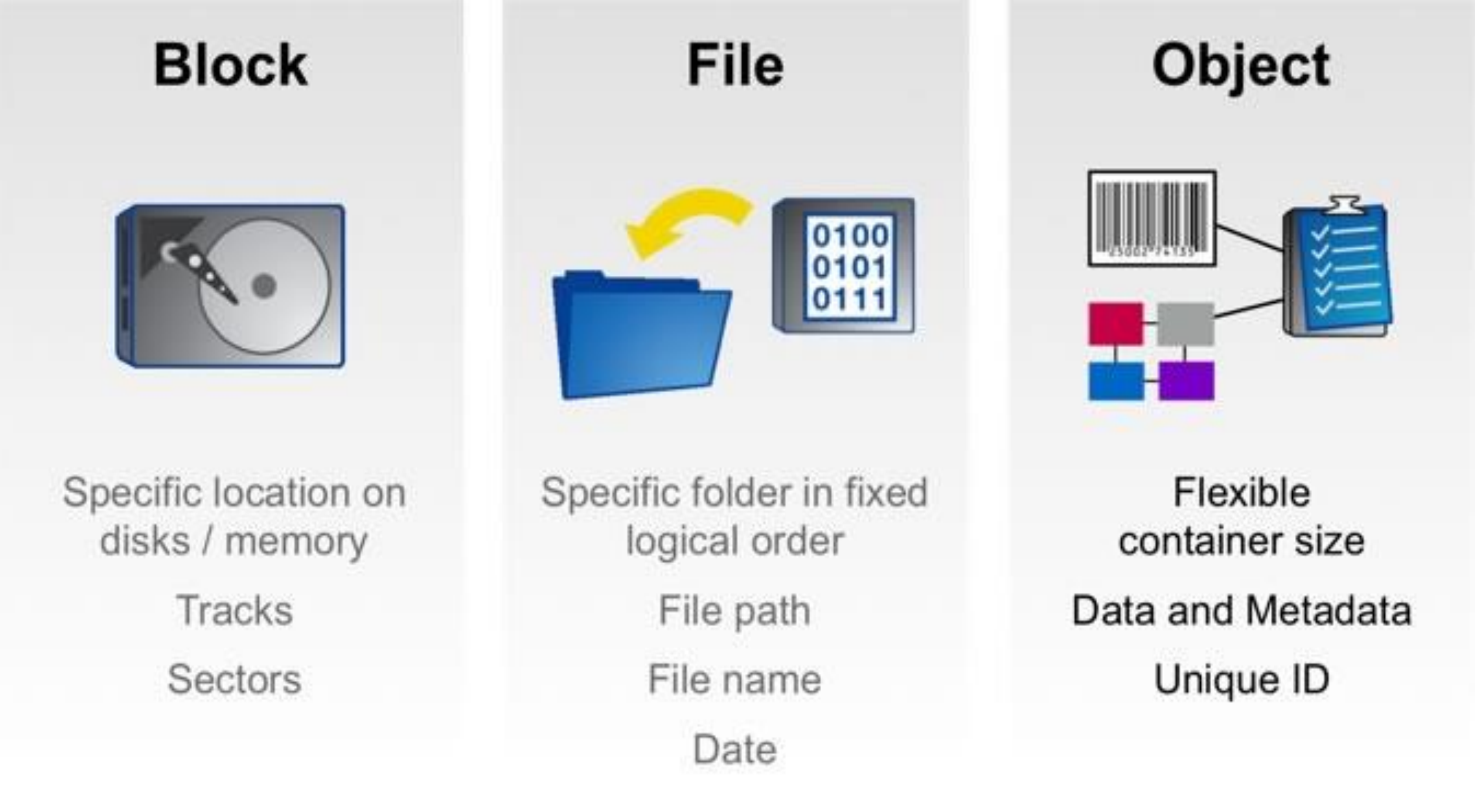
a) File

b) Database sets

c) Block

d) Object

Cloud Storage Levels



Block

Specific location on disks / memory

Tracks

Sectors

File

Specific folder in fixed logical order

File path

File name

Date

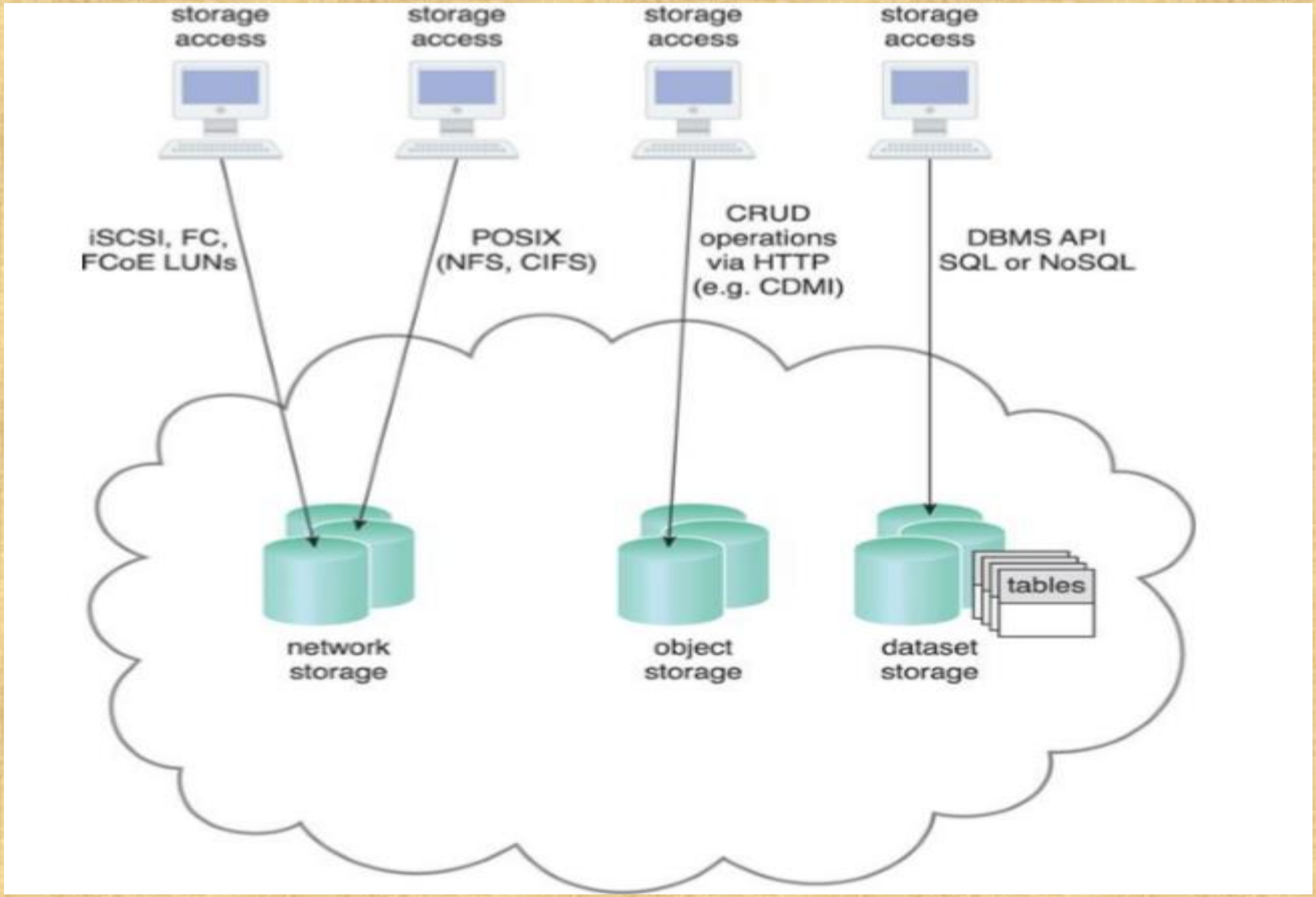
Object

Flexible container size

Data and Metadata

Unique ID

Cloud storage device access



AWS Simple Storage Service (S3)

- Storage abstraction: “buckets”
    - Unlimited number of objects per bucket; 5TB limit per object
- SLAs:
    - Standard:
        - Availability: 99.99% on yearly basis
        - Durability: 99.999999999% (11 nines) on yearly basis
- Encryption: in-flight and at-rest

Google Cloud Storage

- Storage Abstraction: “buckets”
  - Unlimited number of objects per bucket; 5TB limit per object
- SLAs:
  - Standard: 99.9% monthly for standard
  - Durable reduced availability (DRA): 99.0% monthly
  - Latency for both is milliseconds.
- Encryption: Same as AWS, but in alpha

Azure Storage

- Storage Abstraction: “containers” and “blobs”
  - Unlimited number of objects, 500TB limit per storage account; can have multiple storage accounts
- Service Levels:
  - Locally Redundant Storage (LRS), Zone Redundant Storage (ZRS), Geographically Redundant Storage (GRS) (more comparable to AWS and Google), Read-Access Geo-Redundant (RA-GRS)
- Encryption: Same via Azure Encryption Extensions (run it on your VM); can be used with Azure Key Vault

Demands and issues

- Volume size: 1GB to 16TB (in 1GB increments)
- Volume types:
  - Magnetic: 100 IOPS on average, bursting to several hundred IOPS (used mostly for storage/snapshotting)
  - General Purpose (SSD): 3 IOPS/GB up to 10,000 IOPS. Throughput limit of 128MB/sec, up to 160MB/sec on larger (> 170GB) volumes
  - Provisioned IOPS (SSD): Up to 20,000 IOPS/volume. Max throughput of 320MB/sec (when used with EBS-optimized instances)
- Snapshots available across Availability Zones (AZs) but not regions
- Encrypted EBS volumes of all types are supported

Hardware-Based Virtualization

- Volume size: 1GB to 10TB
- Volume Types:
  - HDD (standard magnetic).
    - IOPS: Up to 3,000 read IOPS/15,000 write IOPS
    - Throughput: 180MB/sec read, 120MB/sec write
  - SSD
    - IOPS: Up to 15,000 IOPS.
    - Throughput: Up to 240MB/sec
- Snapshots available across all datacenters in the zone, but not across regions
- All data encrypted in-flight and at-rest by default on all volumes

Azure Block Storage

- Volume size: 1GB to 1TB (significantly smaller on the high end because of how Azure does the back end)
- Implemented as “Page Blobs.” Reads/writes translated to GETs/PUTs on backend
- Volume Types:
  - Standard storage
    - IOPS: 500 IOPS/attached disk
    - Throughput: 60MB/sec
  - Premium storage: SSD based (only available to Azure Virtual Machines, not other services)
    - IOPS: Up to 80,000 IOPS
    - Throughput: 2,000MB/sec
- Snapshots replicated across multiple data centers in the zone, with option for cross-region replication
- All data encrypted in-flight and at-rest via Azure Encryption Extensions

Database

Name	**Amazon DynamoDB X**	**Google Cloud Bigtable X**	**Microsoft Azure Cosmos DB X**
Description	Hosted, scalable database service by Amazon with the data stored in Amazons cloud	Google's NoSQL Big Data database service. It's the same database that powers many core Google services, including Search, Analytics, Maps, and Gmail.	Globally distributed, horizontally scalable, multi-model database service

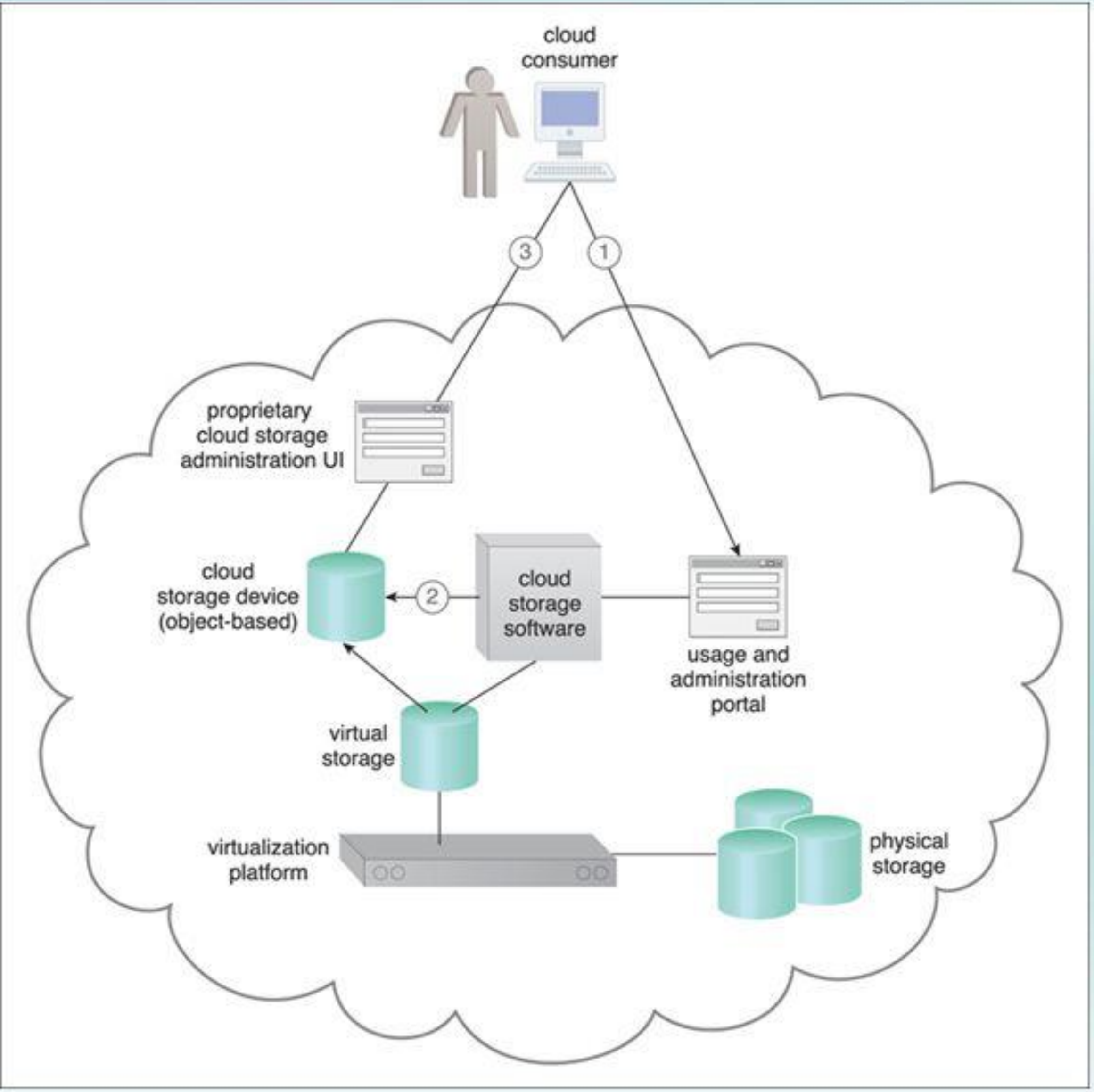

cloud consumer
3
1
proprietary cloud storage administration UI
cloud storage device (object-based)
2
cloud storage software
usage and administration portal
virtual storage
virtualization platform
physical storage

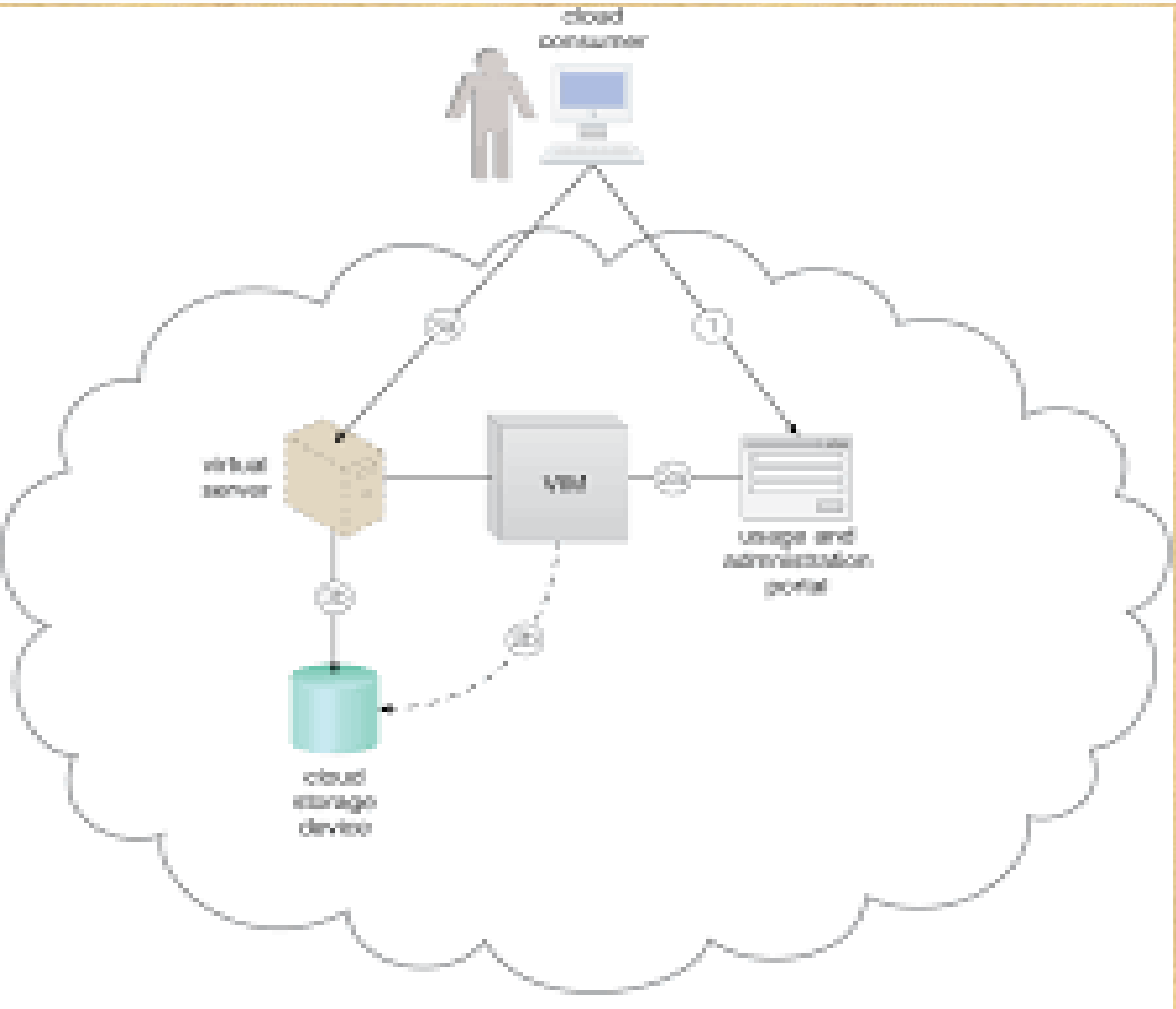
Figure 7.10. The cloud consumer interacts with the usage and administration portal to create a cloud storage device and define access control policies

(1). The usage and administration portal interact with the cloud storage software to create the cloud storage device instance and apply the required access policy to its data objects

(2). Each data object is assigned to a cloud storage device and all of the data objects are stored in the same virtual storage volume. The cloud consumer uses the proprietary cloud storage device UI to interact directly with the data objects

(3). (Note that the usage and administration portal is explained in Chapter 9.)

Other considerations



Copyright © Arcitura Education

Cloud Usage Monitor

- The cloud usage monitor mechanism is a lightweight and autonomous software program responsible for collecting and processing IT resource usage.

Three common agent-based implementation formats:

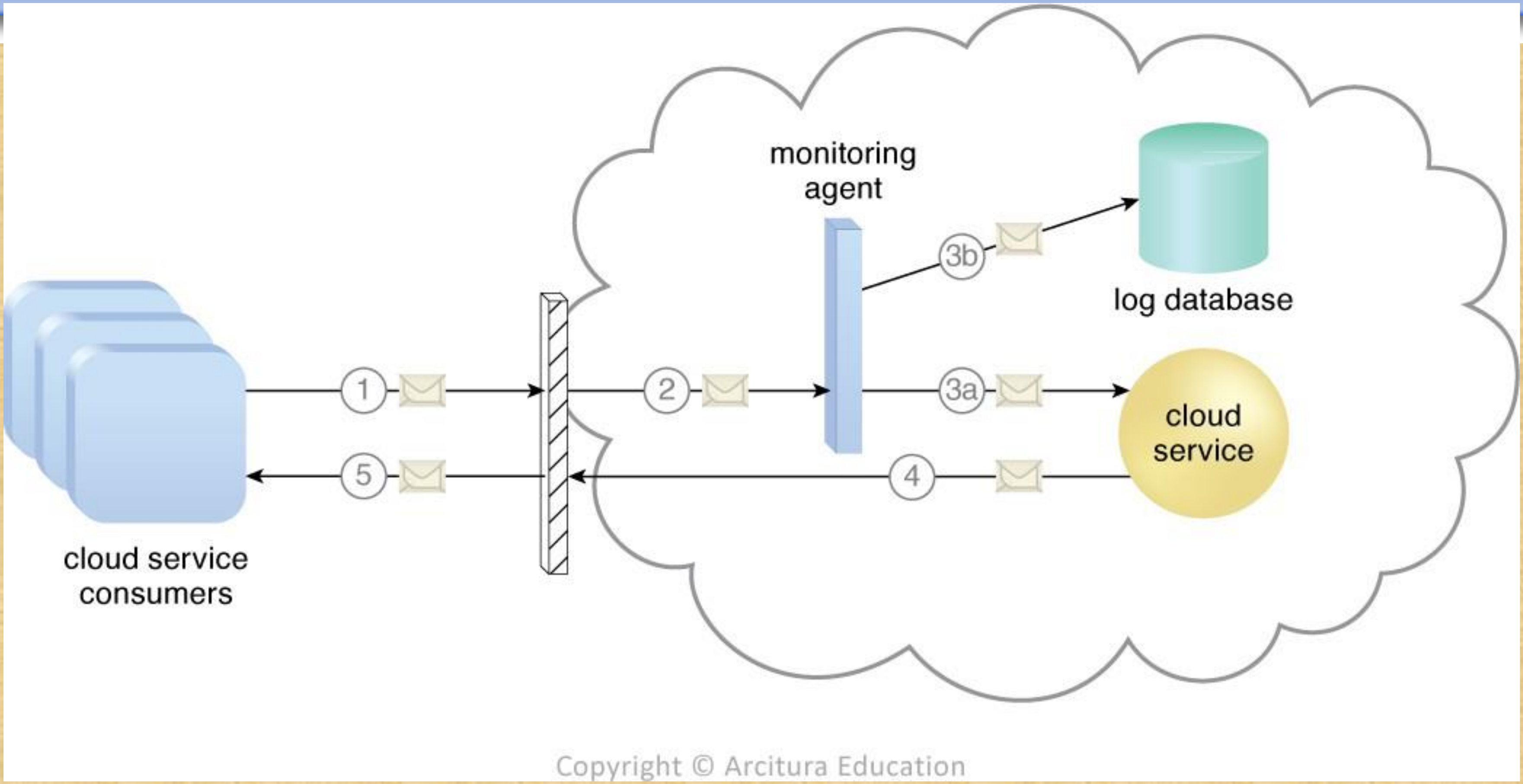
- Monitoring agent
- Resource agent
- Polling agent

Monitoring Agents

- A monitoring agent is an intermediary, event-driven program that exists as a service agent and resides along existing communication paths to transparently monitor and analyze dataflows.

- This type of cloud usage monitor is commonly used to measure network traffic and message metrices.

Monitoring Agents (Cont..)



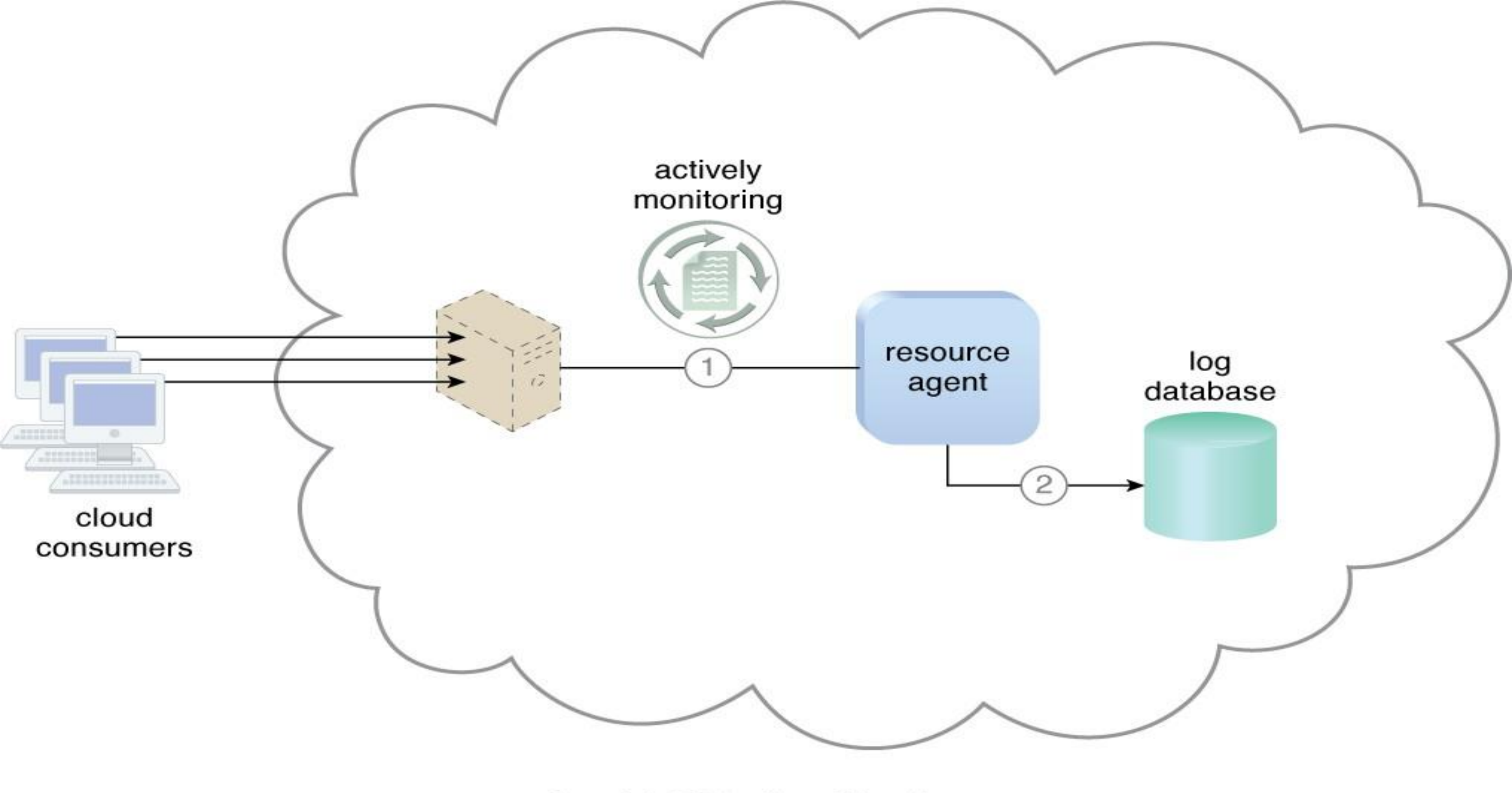
Copyright © Arcitura Education

- *A cloud service consumer send a request message to a cloud service (1).*
- *The monitoring agent intercepts the message to collect relevant usage data (2) before allowing it to continue to the cloud service (3a).*

Resource Agent

- Resource agent monitors usage metrics based on pre-defined, observable events at the resource software level, such as initiating, suspending, resuming, and vertical scaling.

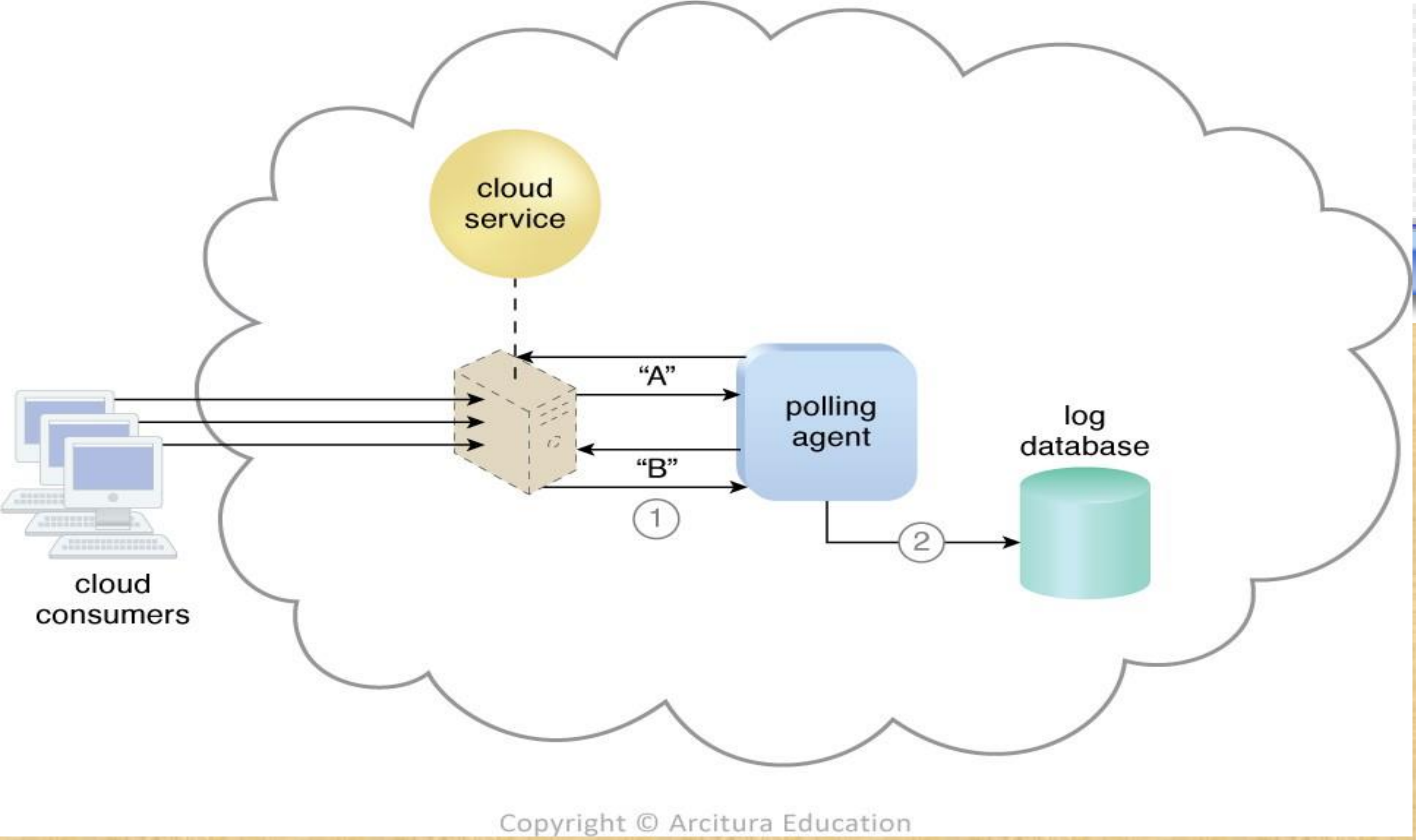
- It is a processing module that collects usage data by having event-driven interactions with specialized resource software.



- *The resource agent is actively monitoring a virtual server and detects an increase in usage (1).*
- *The resource agent receives a notification from the underlying resource management program that the virtual server is being scaled up and stores the collected usage data in a log database, as per its monitoring metrics (2).*

Polling Agent

- Polling agent polls IT resources to periodically monitor IT resource status, eg. up or down time.
- It is a processing module that collects cloud service usage data by polling IT resources.



- *Figure 7.14 - A polling agent monitors the status of a cloud service hosted by a virtual server by sending periodic polling request messages and receiving polling response messages that report usage status "A" after a number of polling cycles, until it receives a usage status of "B" (1), upon which the polling agent records the new usage status in the log database (2).*

A case Study Example

- The resource agent event that is generated by the VIM software can be obtain the following data.

1. Event Type (EV_TYPE) – Generated by the VIM platform, there are five types of events:

- *VM Starting (creation at the hypervisor)*
- *VM Started (completion of the boot procedure)*
- *VM Stopping (shutting down)*
- *VM Stopped (termination at the hypervisor)*
- *VM Scaled (change of performance parameters)*

2. Usage measurements are recorded for every virtual server that a cloud consumer creates.

3. Usage measurements are recorded for a measurement period whose length is defined by two timestamps called t_{start} and t_{end}. The start of the measurement period defaults to the beginning of the calendar month (t_{start} = 2012-12-01T00:00:00-08:00) and finishes at the end of the calendar month (t_{end} = 2012-12-31T23:59:59-08:00). Customized measurement periods are also supported.

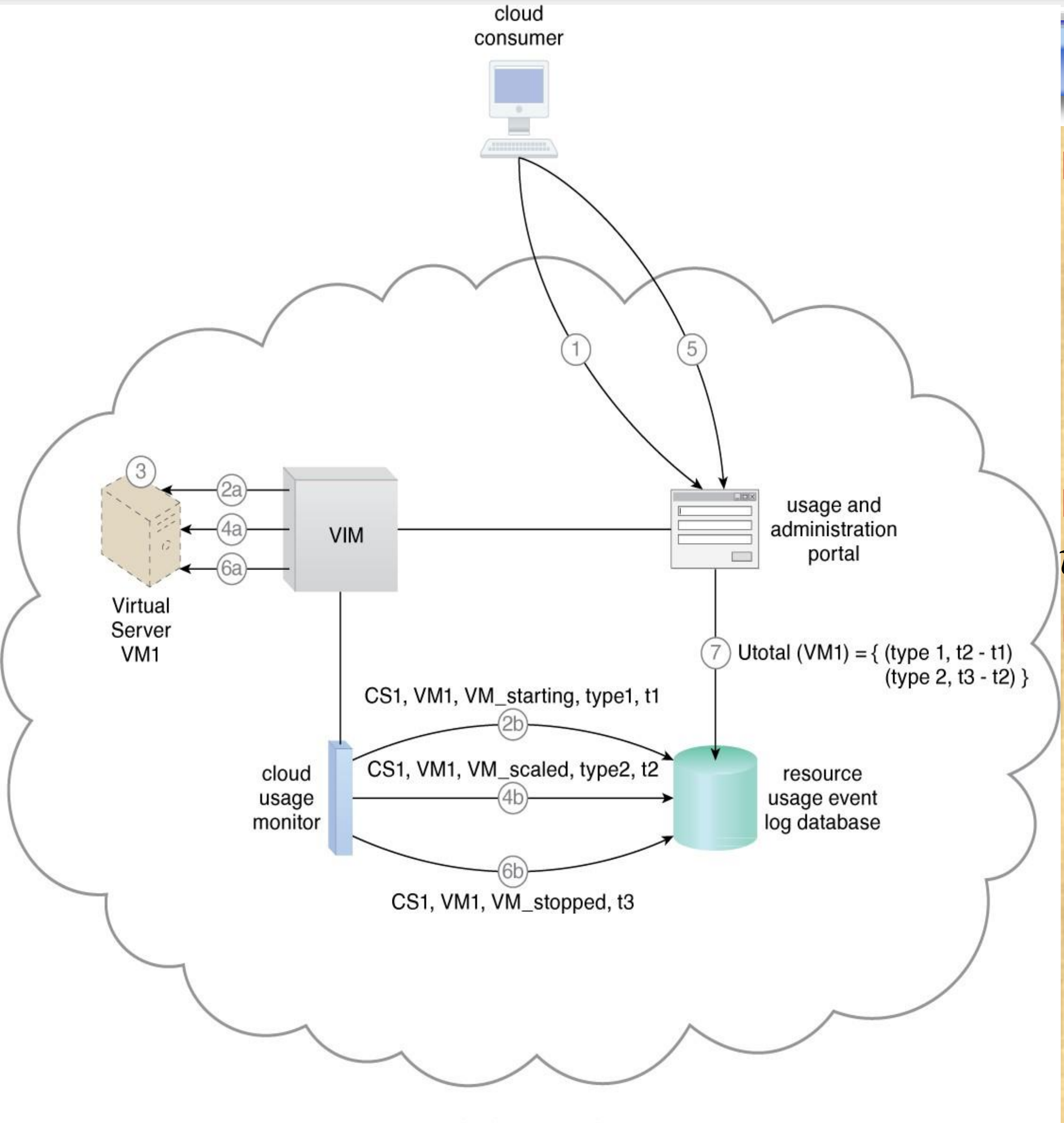
4. Usage measurements are recorded at each minute of usage. The virtual server usage measurement period starts when the virtual server is created at the hypervisor and stops at its termination.

5. Virtual servers can be started, scaled, and stopped multiple times during the measurement period. The time interval between each occurrence i ($i = 1, 2, 3, \ldots$) of these pairs of successive events that are declared for a virtual server is called a usage cycle that is known as T_{cycle_i}:
   - VM_Starting, VM_Stopping – VM size is unchanged at the end of the cycle
   - VM_Starting, VM_Scaled – VM size has changed at the end of the cycle
   - VM_Scaled, VM_Scaled – VM size has changed while scaling, at the end of the cycle
   - VM_Scaled, VM_Stopping – VM size has changed at the end of the cycle

6. The total usage, U_{total}, for each virtual server during the measurement period is calculated using the following resource usage event log database equations:
   - For each VM_TYPE and VM_ID in the log database: $U_{total\_VM\_type\_j} = \sum_{t_{start}}^{t_{end}} T_{cycle_i}$
   - As per the total usage time that is measured for each VM_TYPE, the vector of usage for each VM_ID is U_{total}: U_{total} = {type 1, $U_{total\_VM\_type\_1}$, type 2, $U_{total\_VM\_type\_2}$, ...}

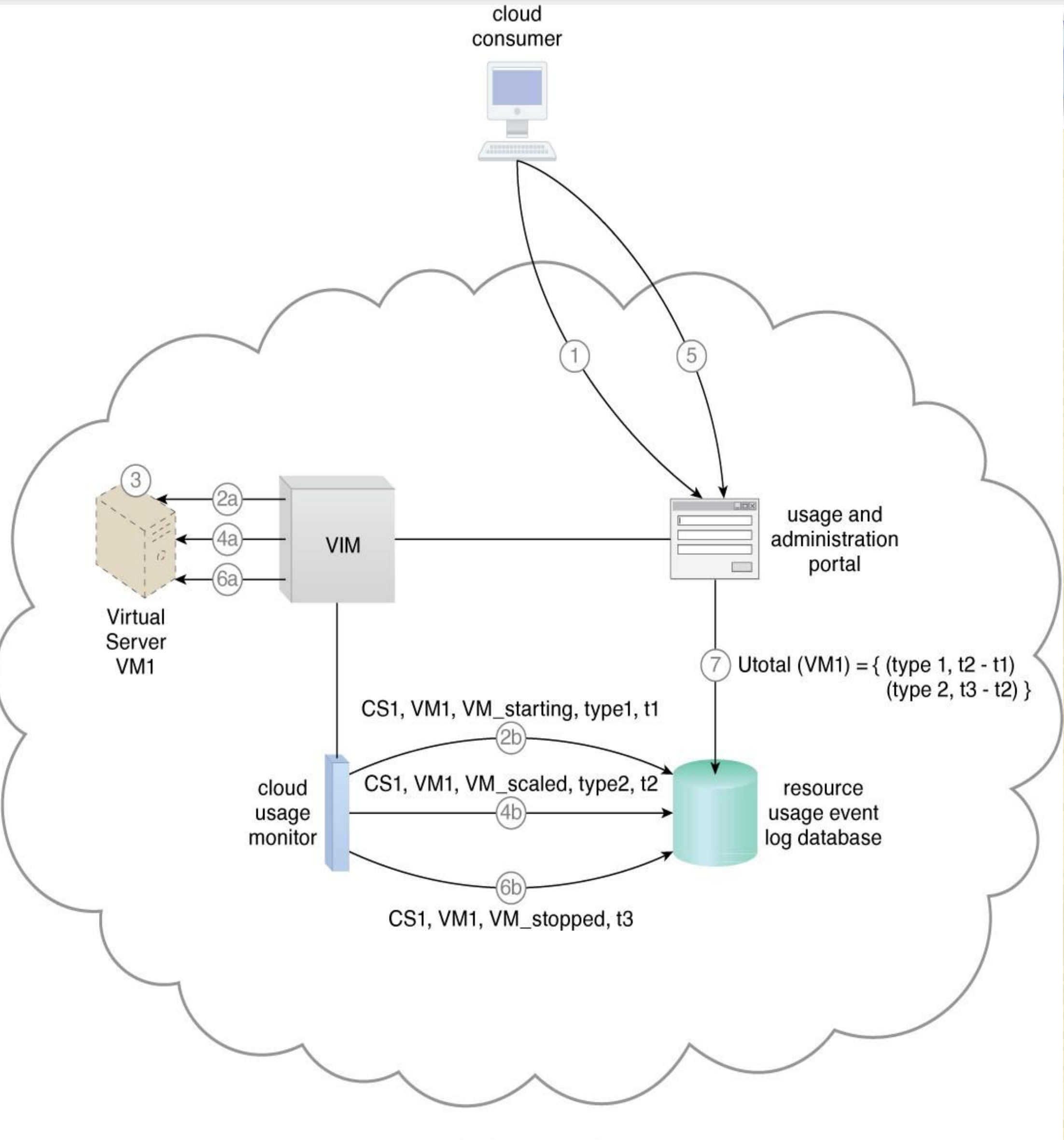
Figure 7.15 depicts the resource agent interacting with the VIM's event-driven API.

A case Study Example



- *The cloud service consumer (CS_ID = CS1) requests the creation of a virtual server (VM_ID = VM1) of configuration size type 1 (VM_TYPE = type1) (1).*
- *The VIM creates the virtual server (2a).*
- *The VIM's event-driven API generates a resource usage event with timestamp = t1, which the usage monitor software agent captures and records in the resource usage event log database (2b).*

A case Study Example

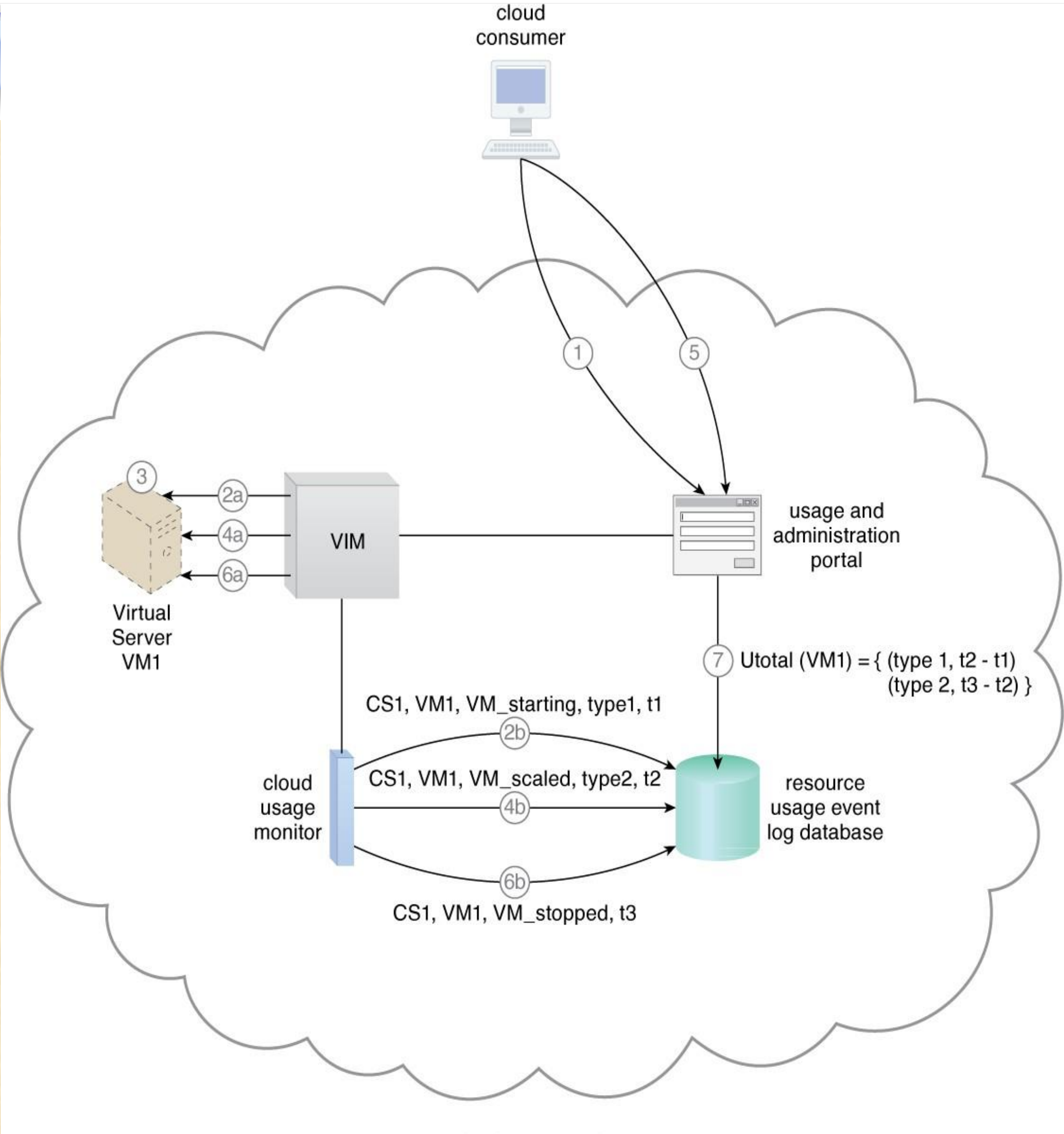

cloud consumer
1
5
3
2a
4a
6a
VIM
usage and administration portal
Virtual Server VM1
7
Utotal (VM1) = { (type 1, t2 - t1)
(type 2, t3 - t2) }
CS1, VM1, VM_starting, type1, t1
2b
CS1, VM1, VM_scaled, type2, t2
4b
cloud usage monitor
resource usage event log database
6b
CS1, VM1, VM_stopped, t3

Copyright © Arcitura Education

- *Virtual server usage increases and reaches the auto-scaling threshold (3).*
- *The VIM scales up Virtual Server (VM1) (4a) from configuration type 1 to type 2 (VM_TYPE = type2).*
- *The VIM's event-driven API generates a resource usage event with timestamp = t2, which is captured and recorded at the resource usage event log database by the usage monitor software agent (4b).*

A case Study Example

- *The cloud service consumer shuts down the virtual server (5).*
- *The VIM stops Virtual Server (VM1) (6a) and its event-driven API generates a resource usage event with timestamp = t3, which the usage monitor software agent captures and records at the log database (6b).*
- *Management tools access the log database and calculate the total usage for Virtual Server Utotal (VM1) (7).*

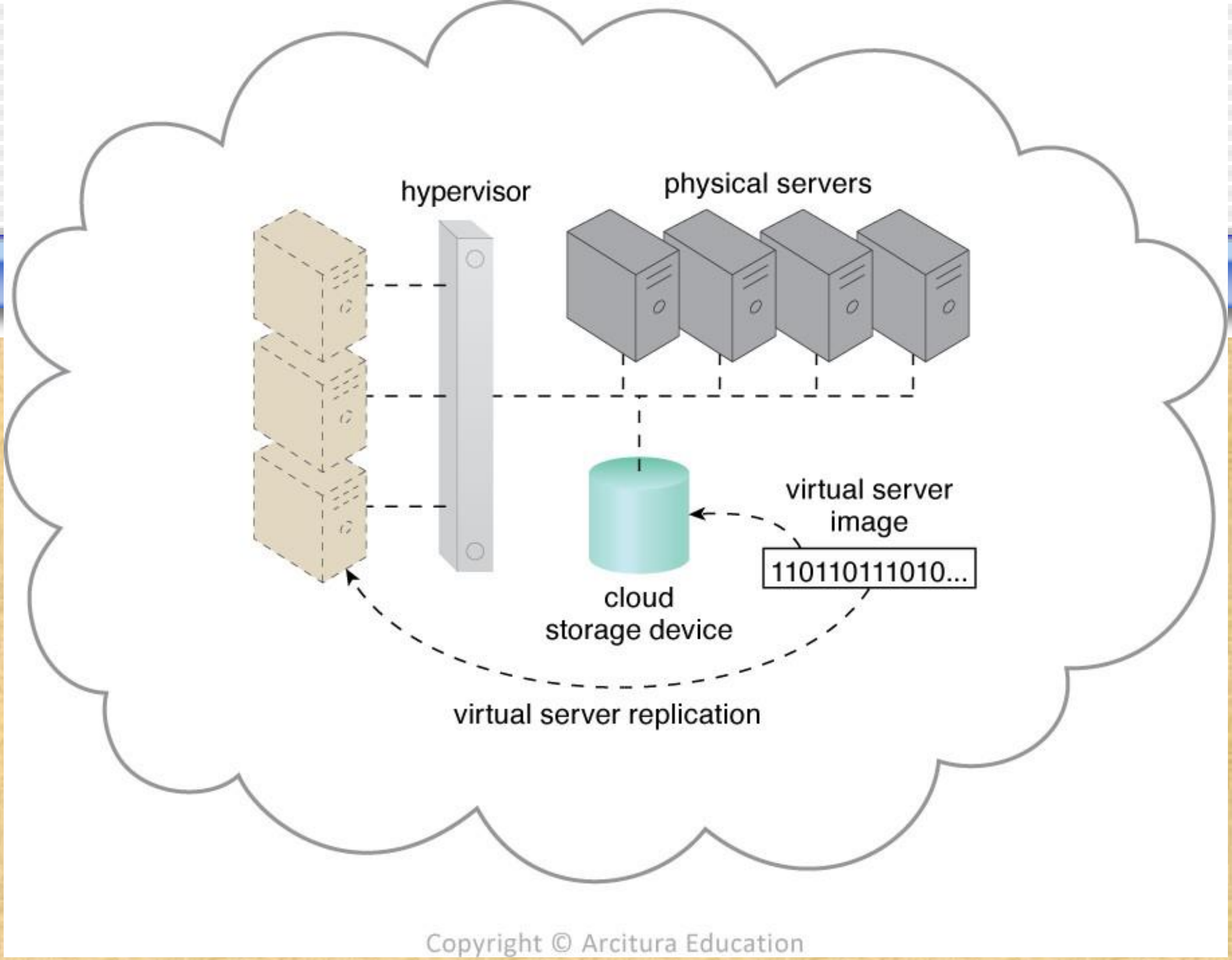

cloud consumer
1
5
3
2a
4a
6a
VIM
usage and administration portal
Virtual Server VM1
7
Utotal (VM1) = { (type 1, t2 - t1)
(type 2, t3 - t2) }
CS1, VM1, VM_starting, type1, t1
2b
CS1, VM1, VM_scaled, type2, t2
4b
6b
CS1, VM1, VM_stopped, t3
cloud usage monitor
resource usage event log database

Copyright © Arcitura Education

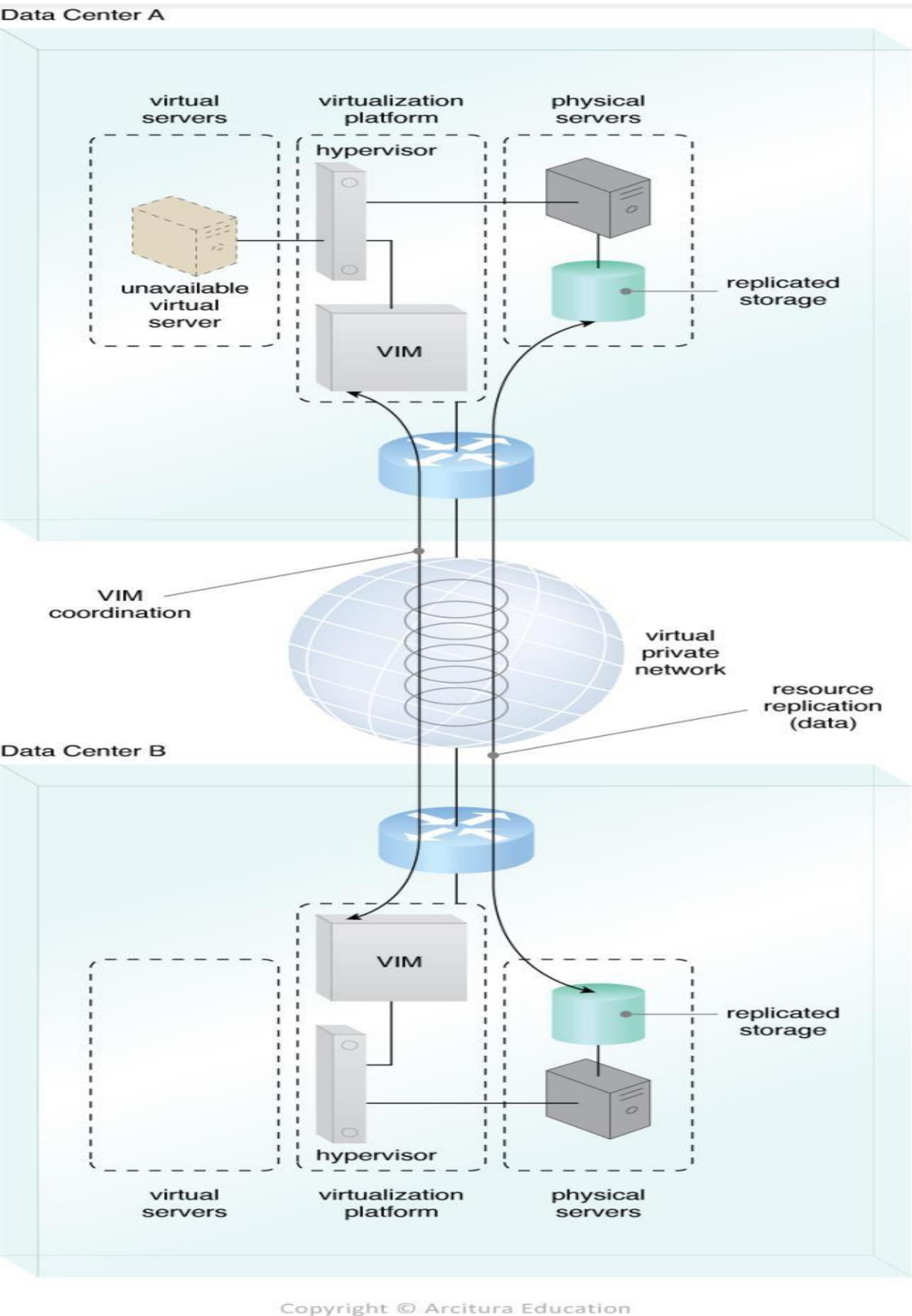
Resource Replication

- Replication is usually performed when resource’s availability and performance need to be enhanced.

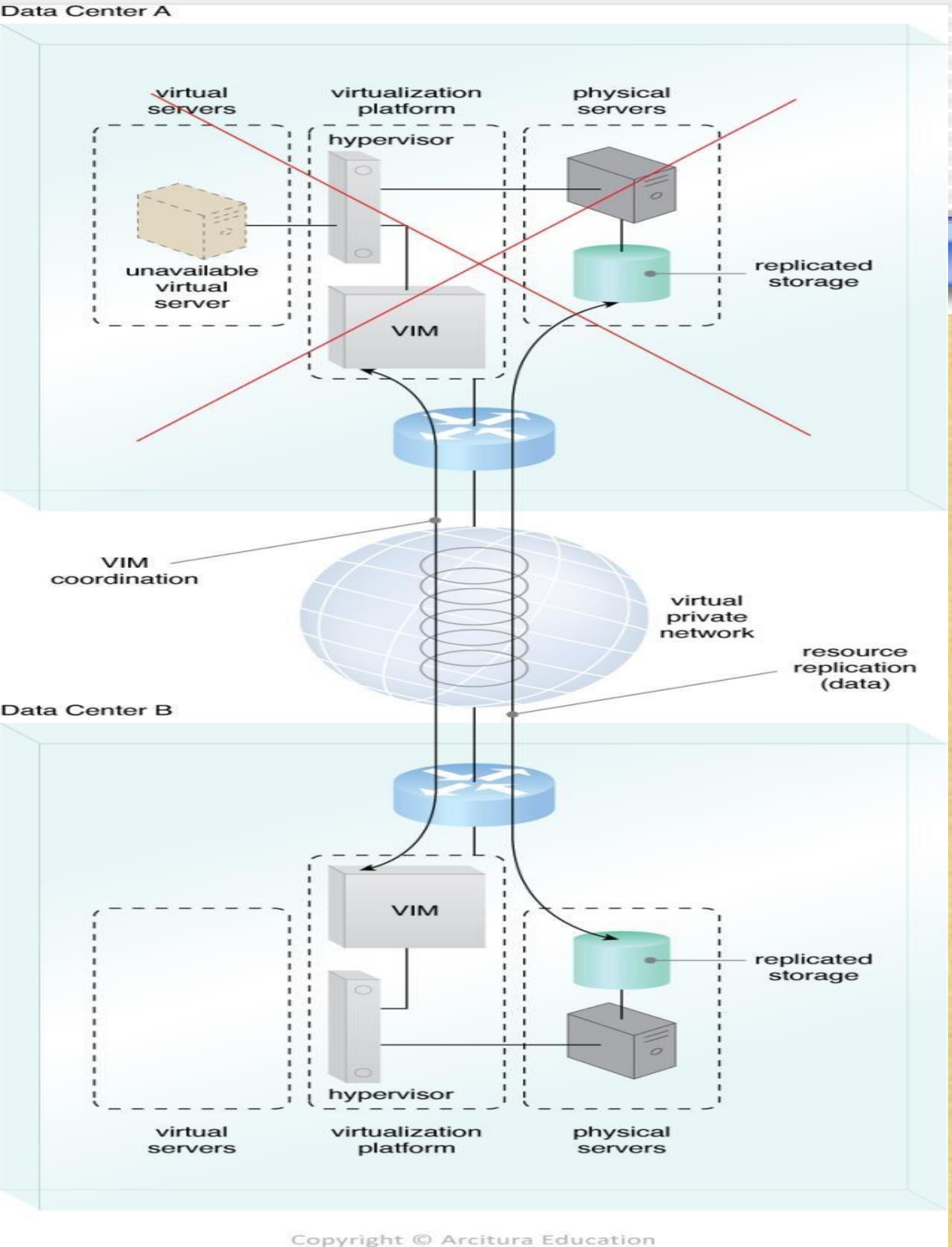
- Resource replication mechanism usually uses irtualization technology to replicate cloud-based IT resources.



- *Figure 7.16 - The hypervisor replicates several instances of a virtual server.*

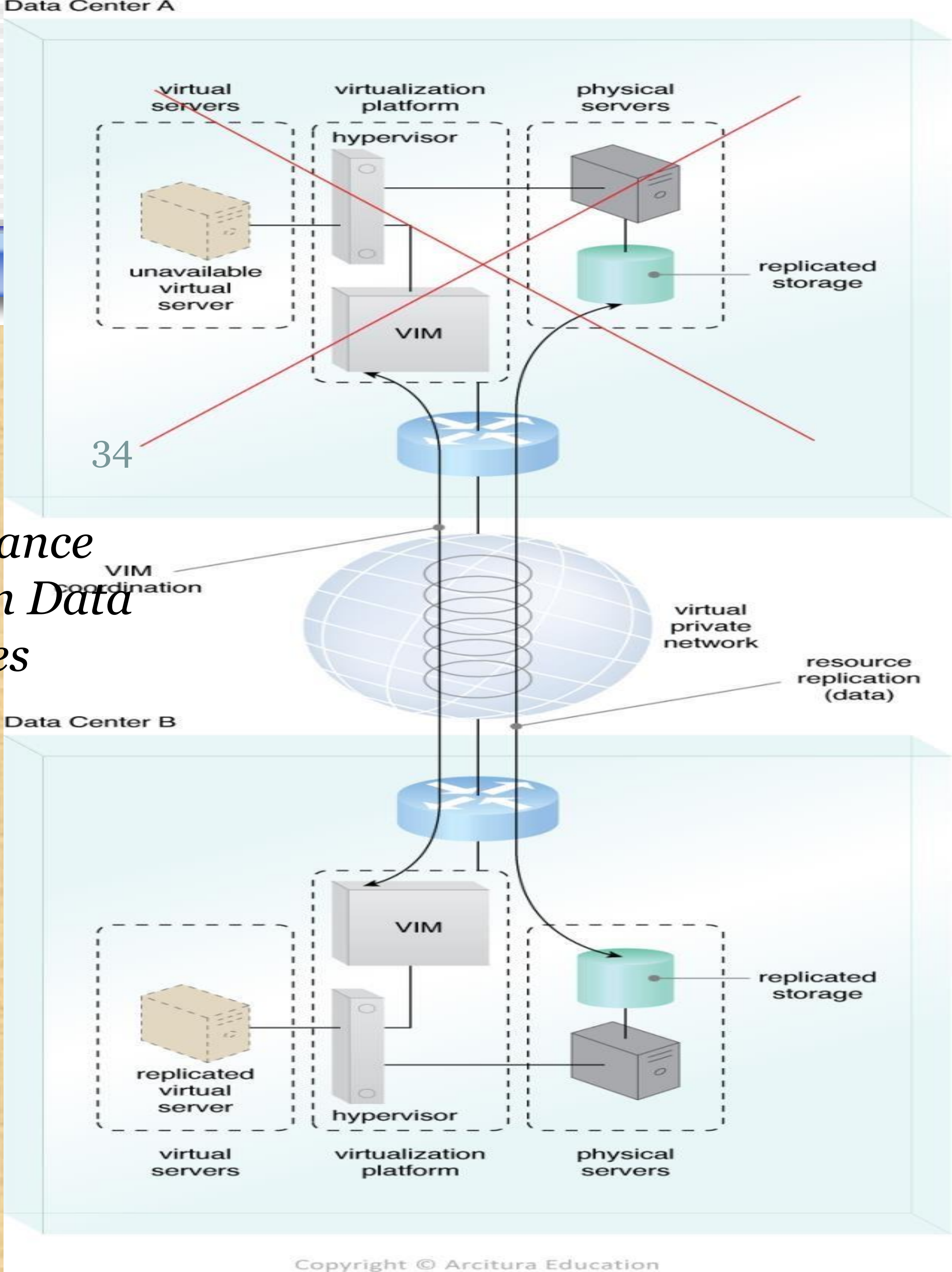


- *Figure 7.17 - A high availability virtual server is running in Data Center A. VIM instances, in Data Center A and Data Center B, are executing the coordination function that allows detection of failure conditions. Storage of VM images is replicated between data centers, as a consequence of the high availability configuration.*



- *Figure 7.18 - The virtual server becomes unavailable in Data Center A. VIM in Data Center B detects the failure condition and starts to reallocate the high availability server from Data Center A into Data Center B.*

- *Figure 7.19 - A new instance of the virtual server is created in Data Center B and the service becomes available.*

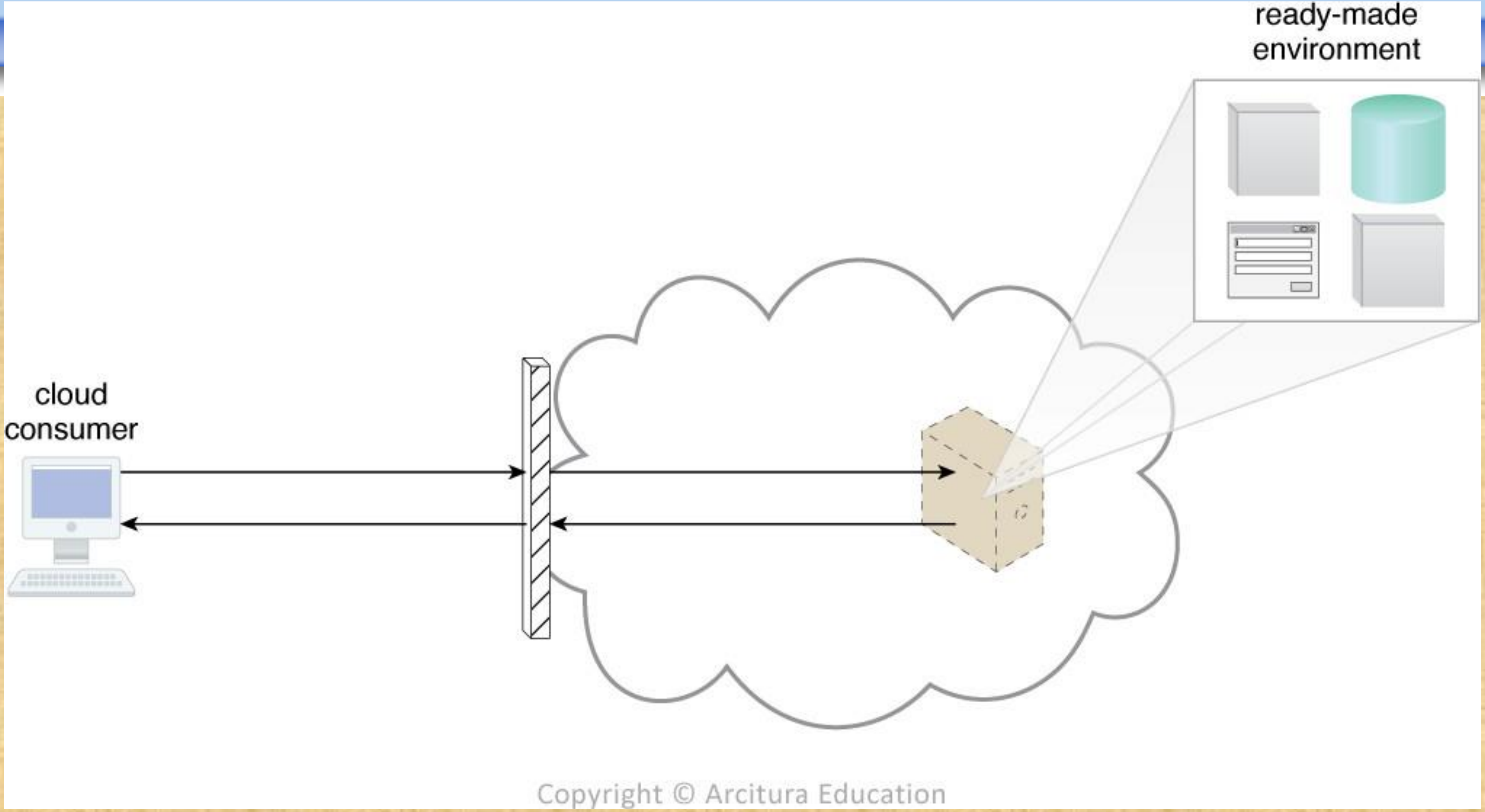


Ready-Made Environment

- The ready-made environment mechanism is a defining component of the PaaS cloud delivery model that represents a platform comprised of a set of already installed IT resources, ready to be used and customized by a cloud consumer.

- Ready-made environments are utilized by cloud consumers to remotely develop and deploy their own services and applications within a cloud by providing with a complete software development kit (SDK).

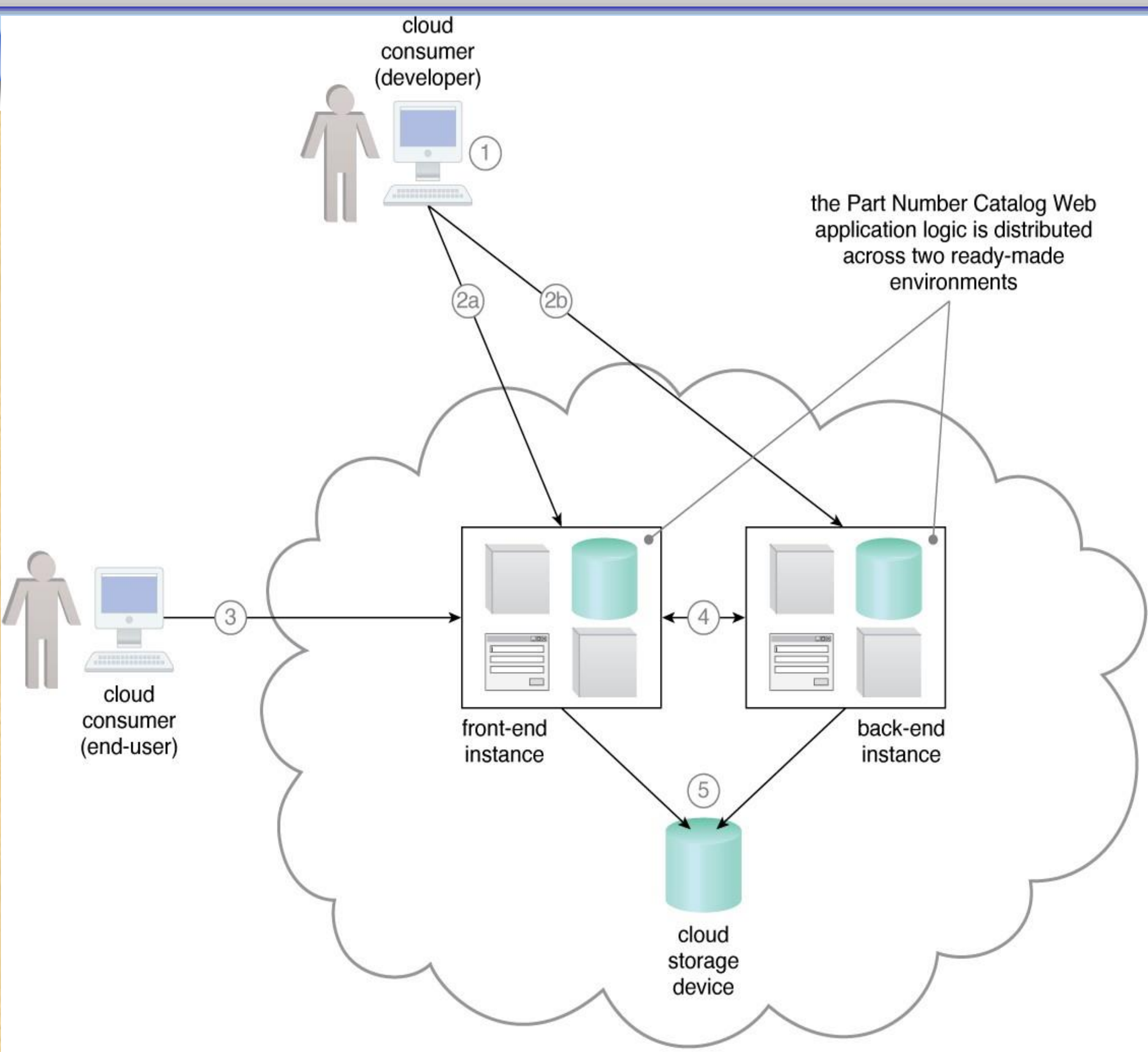
- Typical ready-made environments include pre- installed IT resources, eg. database, middleware, governance tools.



- *Figure 7.20 - A cloud consumer accesses a ready-made environment hosted on a virtual server.*

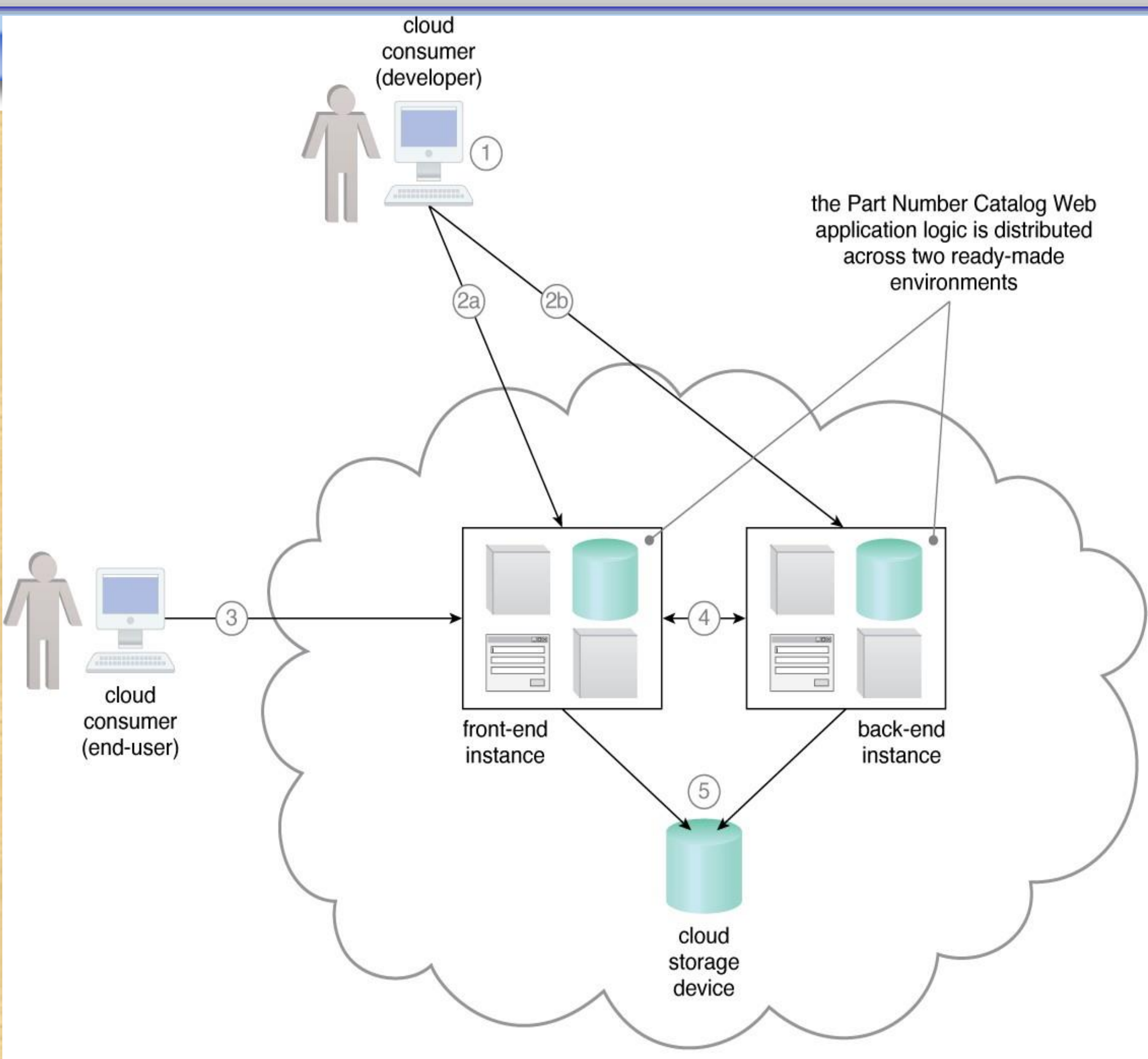
Customize features

- *The cloud developer uses the provided SDK to develop the Web application (1).*
- *The application software is deployed on the Web platform that was established by the two ready-made environments called the frontend instance (2a) and the backend instance (2b).*


cloud consumer (developer)
1
2a
2b
the Part Number Catalog Web application logic is distributed across two ready-made environments
3
4
cloud consumer (end-user)
front-end instance
back-end instance
5
cloud storage device

Copyright © Arcitura Education

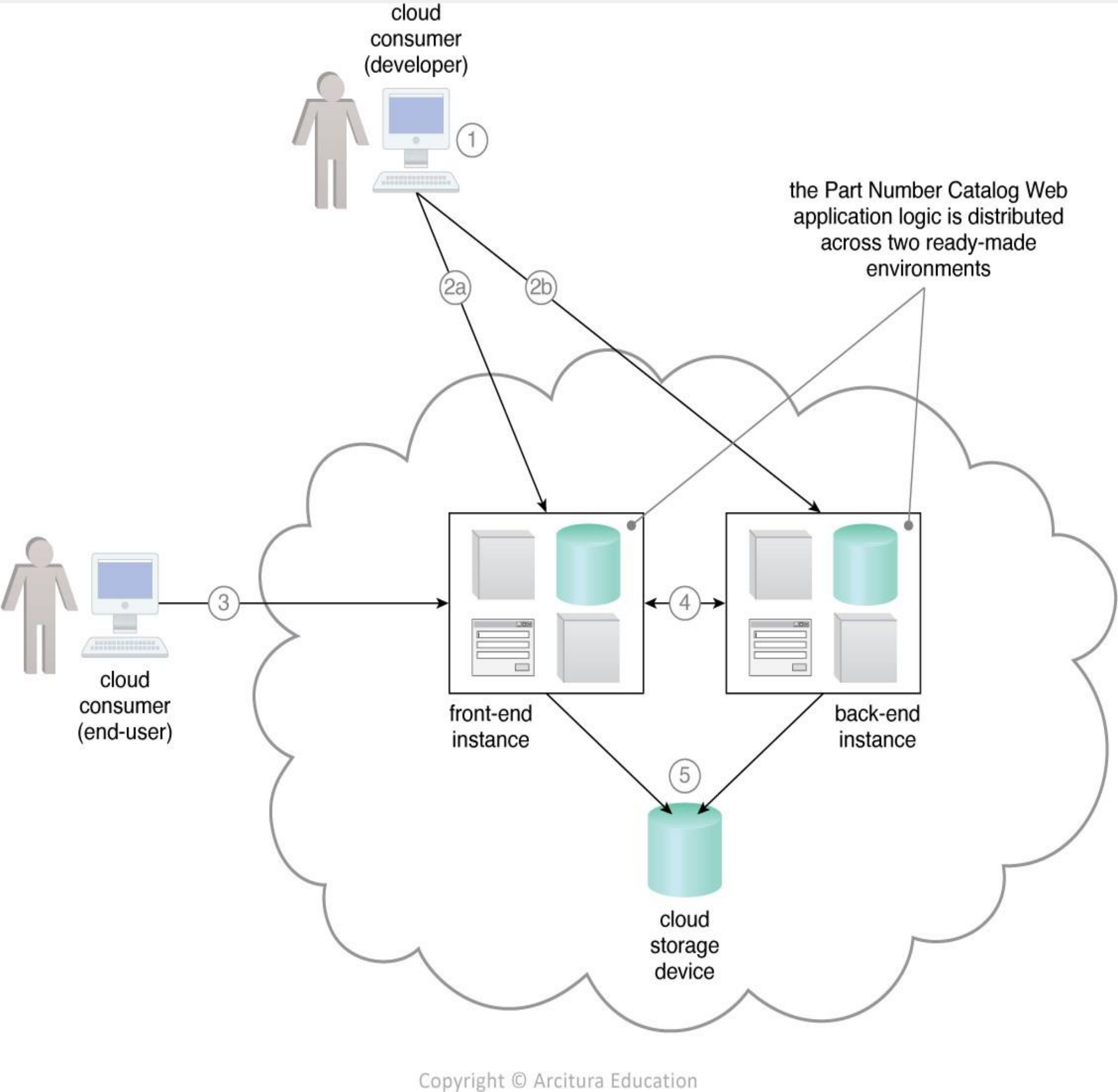
- *The application is made available for end-users and one accesses the application interface at the frontend instance (3).*
- *The software running in the frontend instance invokes a long-running task at the backend instance that corresponds to the processing required by the end-user (4).*


cloud consumer (developer)
1
2a
2b
the Part Number Catalog Web application logic is distributed across two ready-made environments
3
4
cloud consumer (end-user)
front-end instance
back-end instance
5
cloud storage device

Copyright © Arcitura Education

Multitenancy

- *The application software deployed at both the frontend and backend instances is backed by a cloud storage device that implements persistent storage of the application data (5).*


cloud consumer (developer)
1
2a
2b
the Part Number Catalog Web application logic is distributed across two ready-made environments
3
4
cloud consumer (end-user)
front-end instance
back-end instance
5
cloud storage device

Conclusion