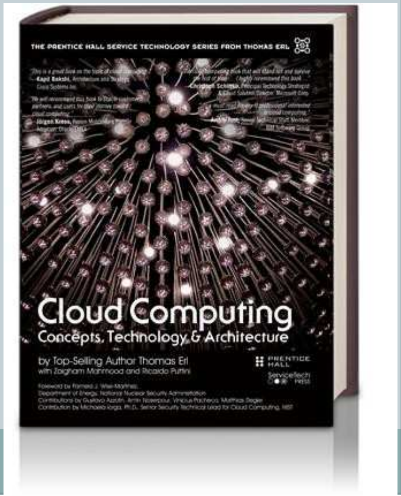
Cloud Computing

Concept, Technology & Architecture

Chapter 08
Specialized Cloud Mechanisms



Cloud Computing
Concepts, Technology & Architecture

Content
s

- 10 mechanisms fulfill a specific runtime function in support of cloud characteristics including:
  - 8.1 Automated Scaling Listener
  - 8.2 Load Balancer
  - 8.3 SLA Monitor
  - 8.4 Pay-Per-Use Monitor
  - 8.5 Audit Monitor
  - 8.6 Failover System
  - 8.7 Hypervisor
  - 8.8 Resource Cluster
  - 8.9 Multi-Device Broker
  - 8.10 State Management Database

8.1 Automated Scaling Listener

- The automated scaling listener mechanism is a service agent that monitors and tracks communications between cloud service consumers and cloud services for dynamic scaling purposes.
- Automatic scaling listeners are usually deployed near the firewall to track the volume of customer-generated requests or processing demands.
- Automatic scaling listeners provide different types of responses to workload fluctuation conditions, such as:

  (i)Automatically scaling IT resources out or in based on parameters defined by the cloud consumer. (auto-scaling)

  (iii)Automatically notification of cloud consumer when the workloads exceed current thresholds or fall below allocated resources. (auto-notification)

Figure

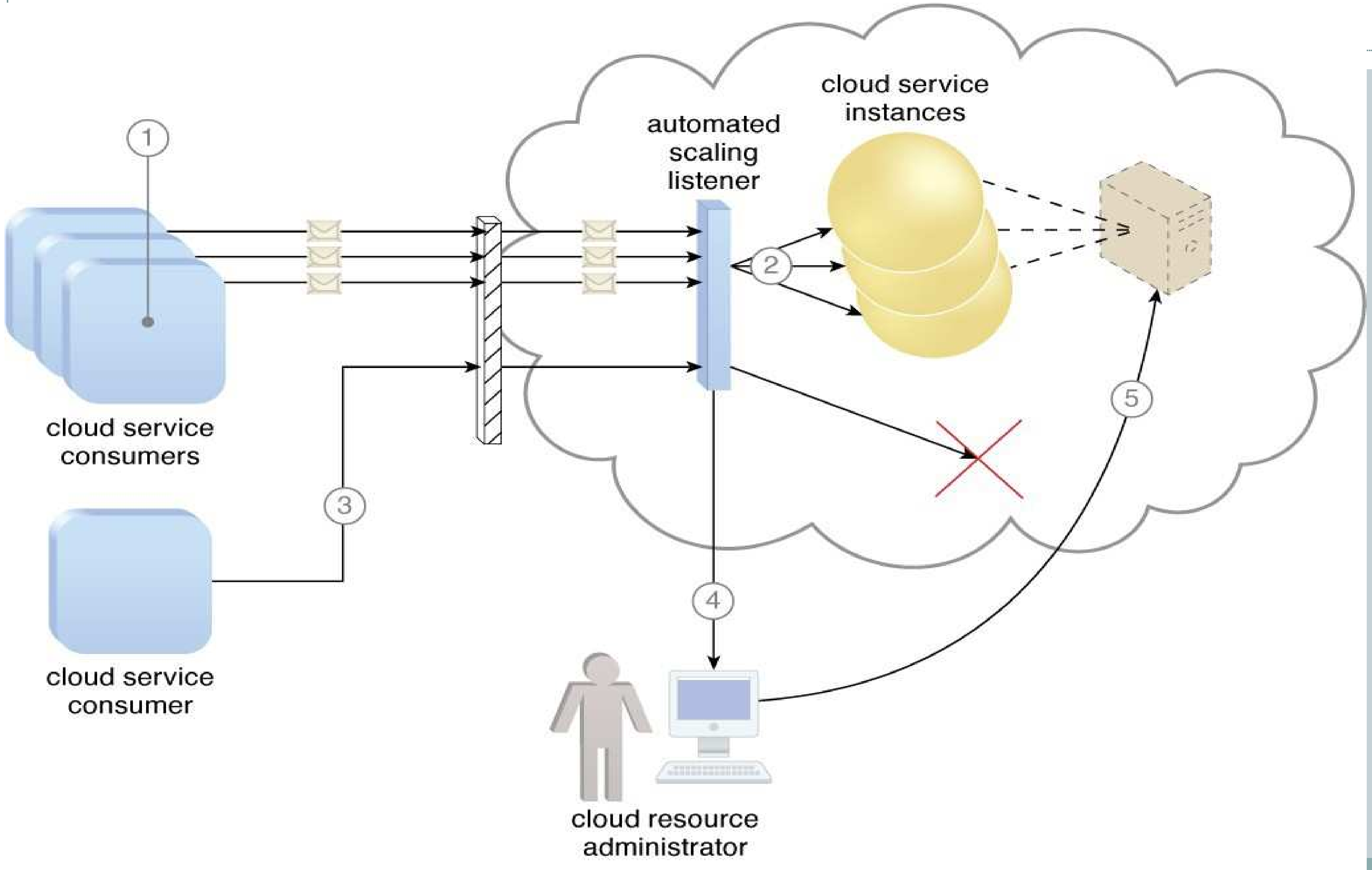

cloud service instances
automated scaling listener
1
2
3
4
5
cloud service consumers
cloud service consumer
cloud resource administrator

Figure 8.1 Automated Scaling Listener

1) Three cloud service consumers attempt to access one cloud service simultaneously (1).
2) The automated scaling listener scales out and initiates the creation of three redundant instances of the service (2).
3) A fourth cloud service consumer attempts to use the cloud service (3).
4) Programmed to allow up to only three instances of the cloud service, the automated scaling listener rejects the fourth attempt and notifies the cloud consumer that the requested workload limit has been exceeded (4).
5) The cloud consumer accesses the remote administration environment to adjust the provisioning setup and increase the redundant instance limit (5).

Figure 8.2 (1/2)

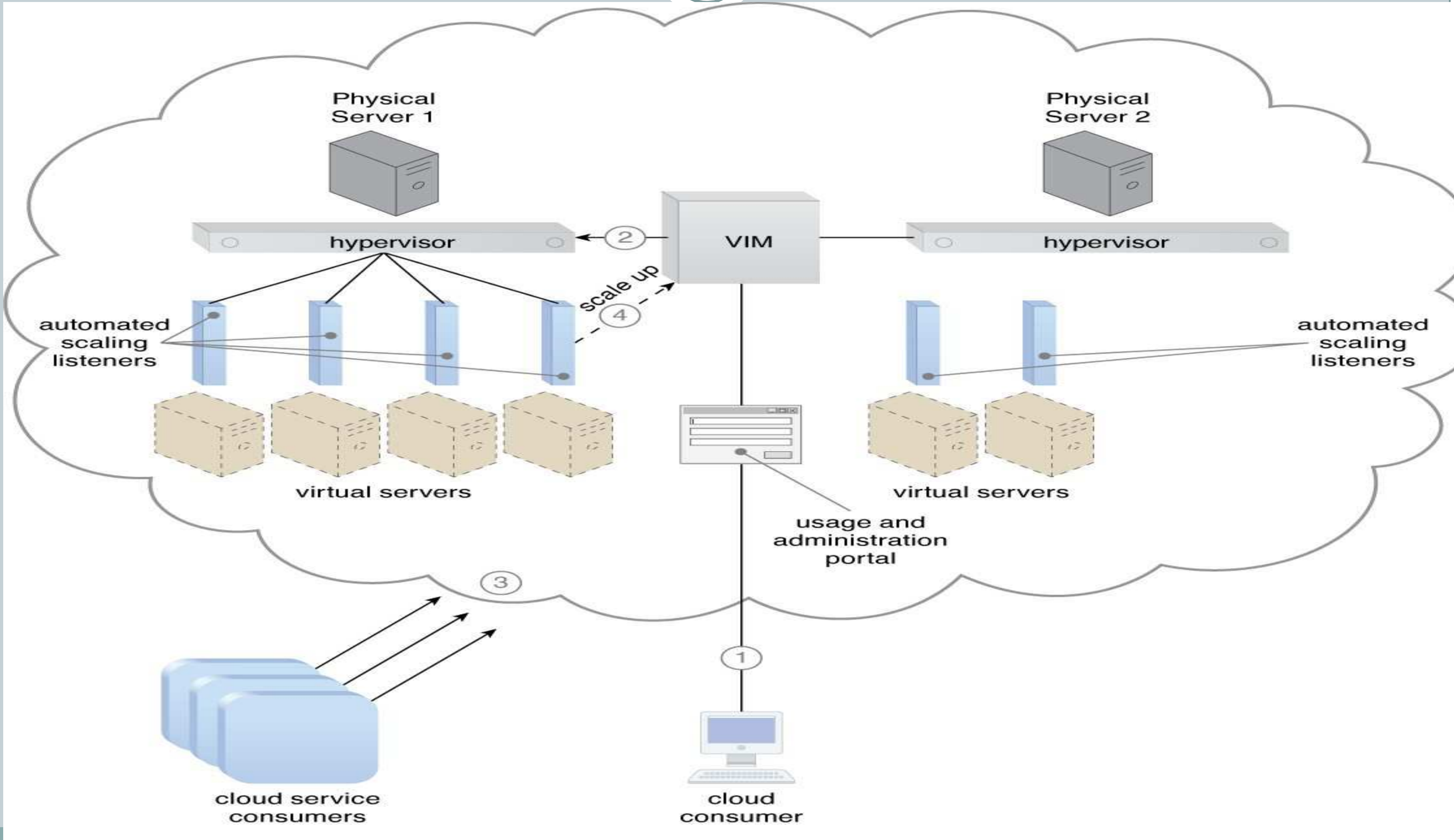

Physical Server 1
Physical Server 2
hypervisor
2
VIM
hypervisor
scale up
4
automated scaling listeners
automated scaling listeners
virtual servers
virtual servers
usage and administration portal
3
1
cloud service consumers
cloud consumer

Figure 8.2 Case of DTGOV

1) A cloud consumer creates and starts a virtual server with 8 virtual processor cores and 16 GB of virtual RAM (1).
2) The VIM creates the virtual server at the cloud service consumer's request, and the corresponding virtual machine is allocated to Physical Server 1 to join 3 other active virtual machines (2).
3) Cloud consumer demand causes the virtual server usage to increase by over 80% of the CPU capacity for 60 consecutive seconds (3).
4) The automated scaling listener running at the hypervisor detects the need to scale up and commands the VIM accordingly (4).

Figure 8.3

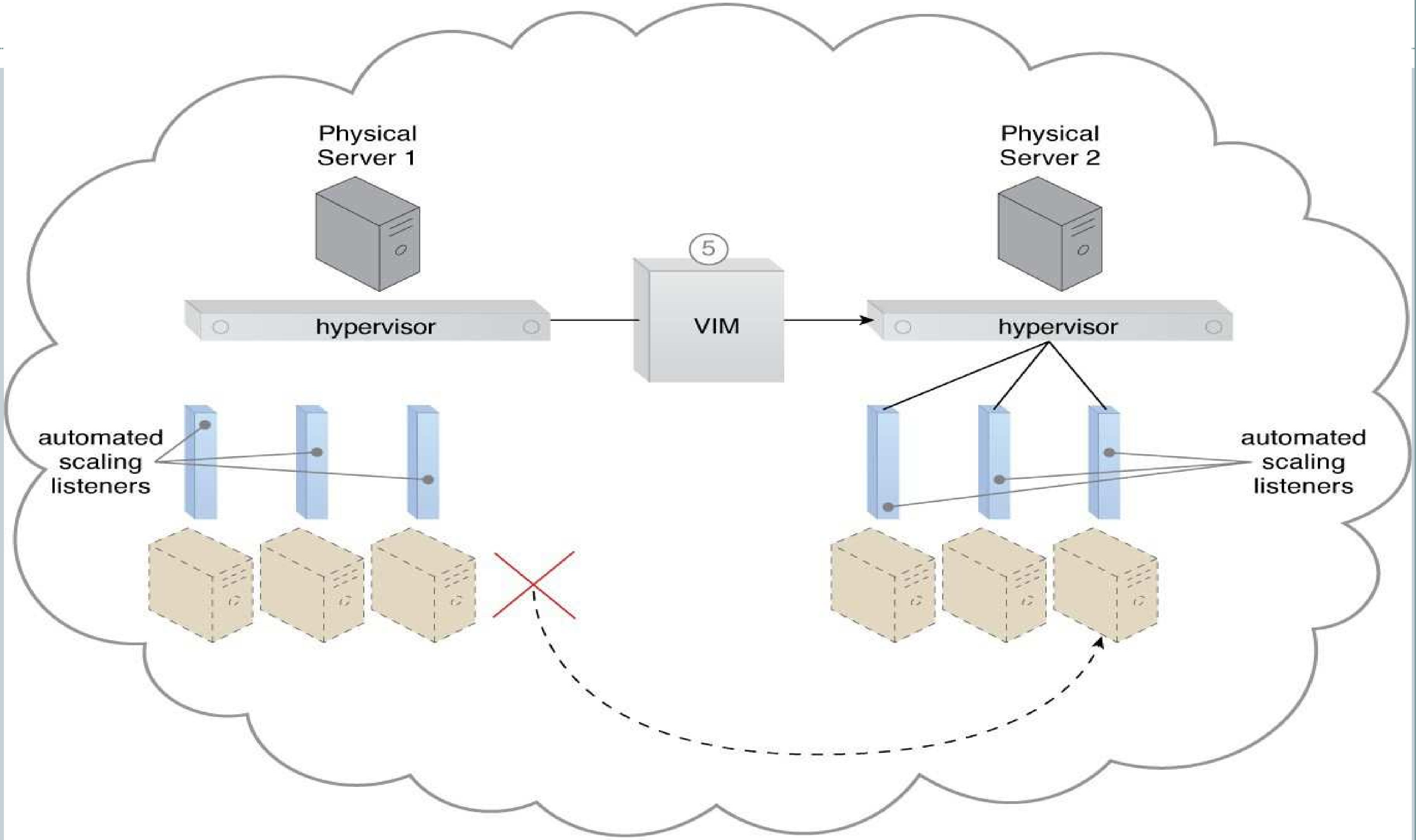
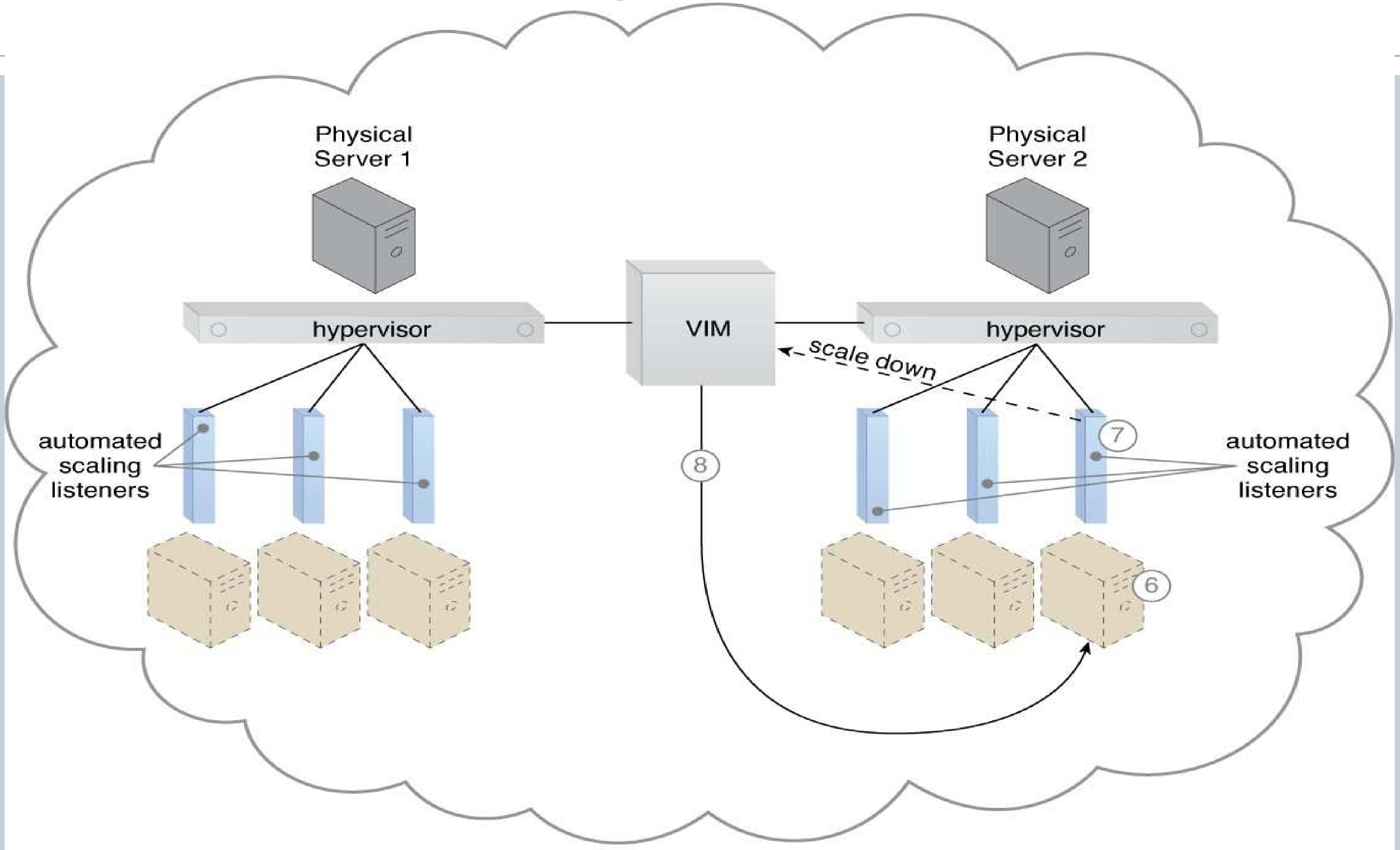

Physical Server 1
Physical Server 2
5
VIM
hypervisor
hypervisor
automated scaling listeners
automated scaling listeners

Figure 8.4



Physical Server 1
Physical Server 2
hypervisor
VIM
hypervisor
scale down
automated scaling listeners
8
7
automated scaling listeners
6

Figure 8.3 & 8.4 Case of DTGOV

- Figure 8.3 - The VIM determines that scaling up the virtual server on Physical Server 1 is not possible and proceeds to live migrate it to Physical Server 2.

6) The virtual server's CPU/RAM usage remains below 15% capacity for 60 consecutive seconds (6).
7) The automated scaling listener detects the need to scale down and commands the VIM (7), which scales down the virtual server (8) while it remains active on Physical Server 2.

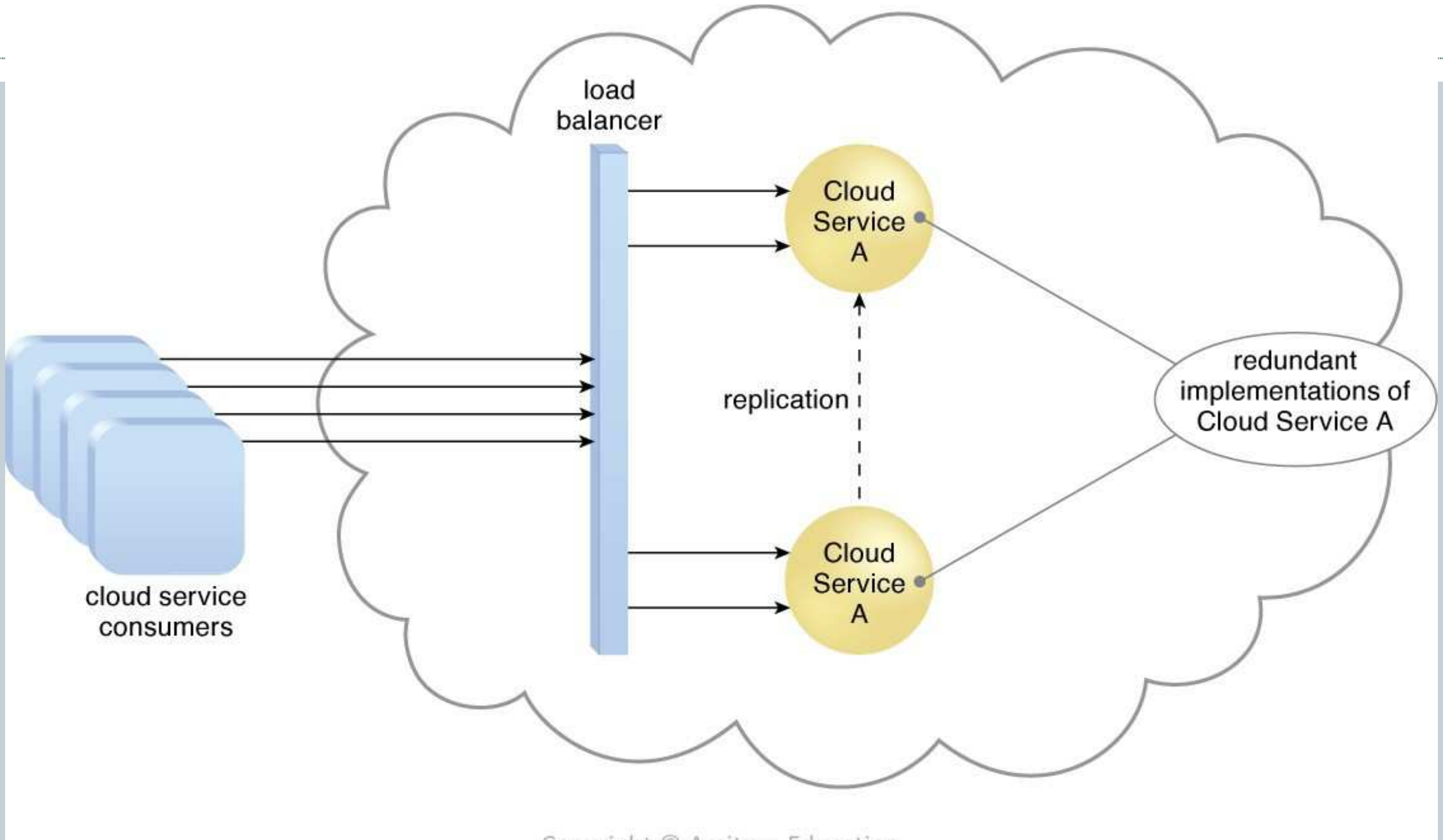
8.2 Load Balancer (1/2)

- A common approach to horizontal scaling is to balance a workload across two or more IT resources to increase performance and capacity beyond what a single IT resource can provide. The load balancer mechanism is a runtime agent with logic fundamentally based on this premise.
- Load balancers perform a range of specialized runtime workload distribution functions:
  - Asymmetric distribution
  - Workload prioritization
  - Content-aware distribution

8.2 Load Balancer (2/2)

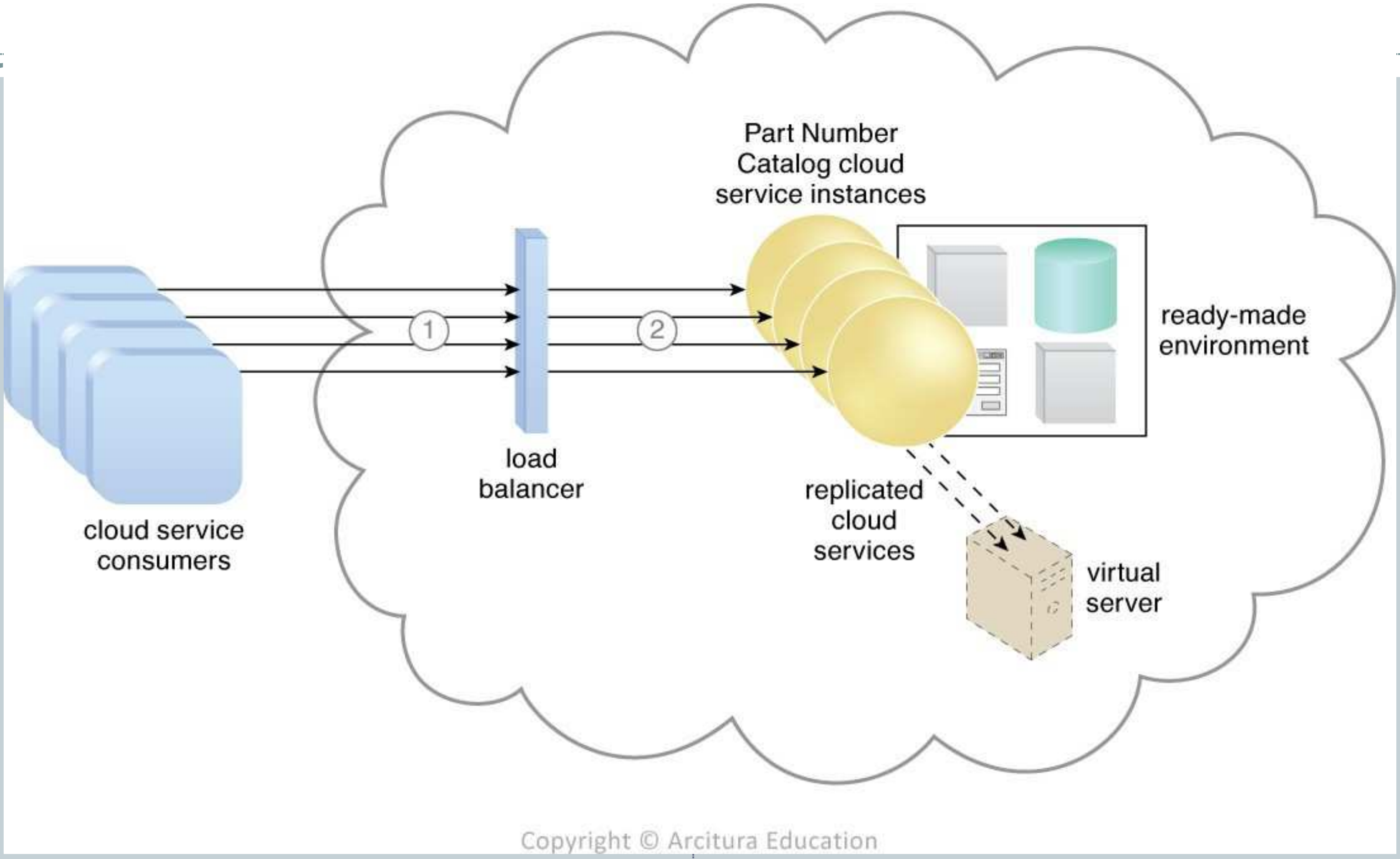
- The load balancer mechanisms can exist as:
  - multi-layer network switch
  - dedicated hardware appliance
  - dedicated software-based system (in server OS)
  - service agent (controlled by cloud management software)
- The load balancer mechanism is typically located on the communication path between the IT resource generating and workload processing.
- The load balancer can be designed as a transparent agent or as a proxy component.

Figure 8.5



- *Figure 8.5 - A load balancer implemented as a service agent transparently distributes incoming workload request messages across two redundant cloud service implementations, which in turn maximizes performance for the cloud service consumers.*

Figure 8.6 Case of ATN Part number Ca



- *Figure 8.6 - New instances of the cloud services are automatically created to meet increasing usage requests. The load balancer uses round-robin scheduling to ensure that the traffic is distributed evenly among the active cloud services.*

8.3 SLA Monitor

- The SLA monitor mechanism is used to specifically observe runtime performance of cloud services to ensure that they are fulfilling the contractual QoS requirements that are published in SLAs.
- The data collected by the SLA monitor is processed by an SLA management system to be aggregated into SLA reporting metrics.

Figure 8.7 (1/2)

- The SLA monitor polls the cloud service by sending over polling request messages (MREQ1 to MREQN). The monitor receives polling response messages (MREP1 to MREPN) that report that the service was "up" at each polling cycle (1a).
- The SLA monitor stores the "up" time-time period of all polling cycles 1 to N-in the log database (1b).

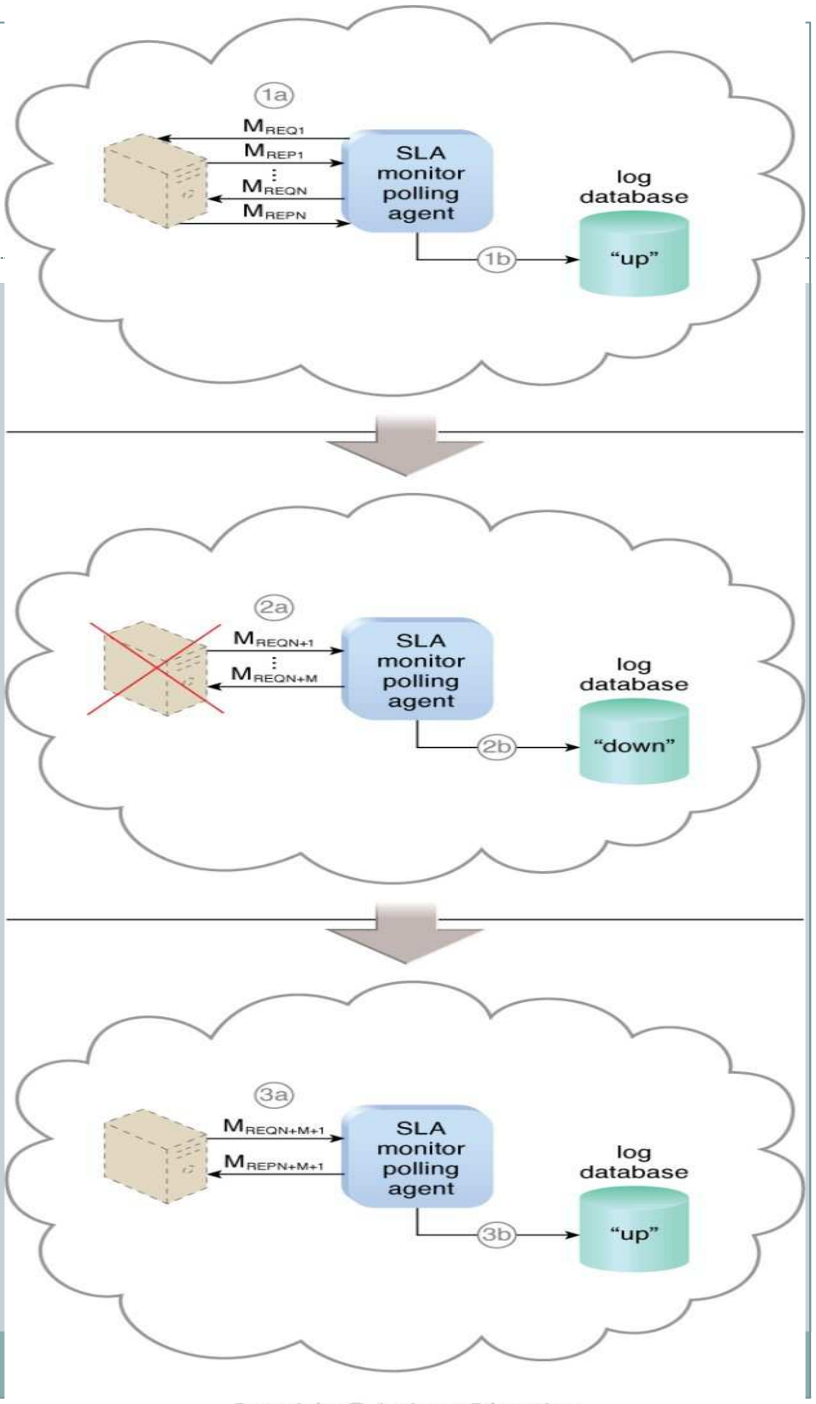
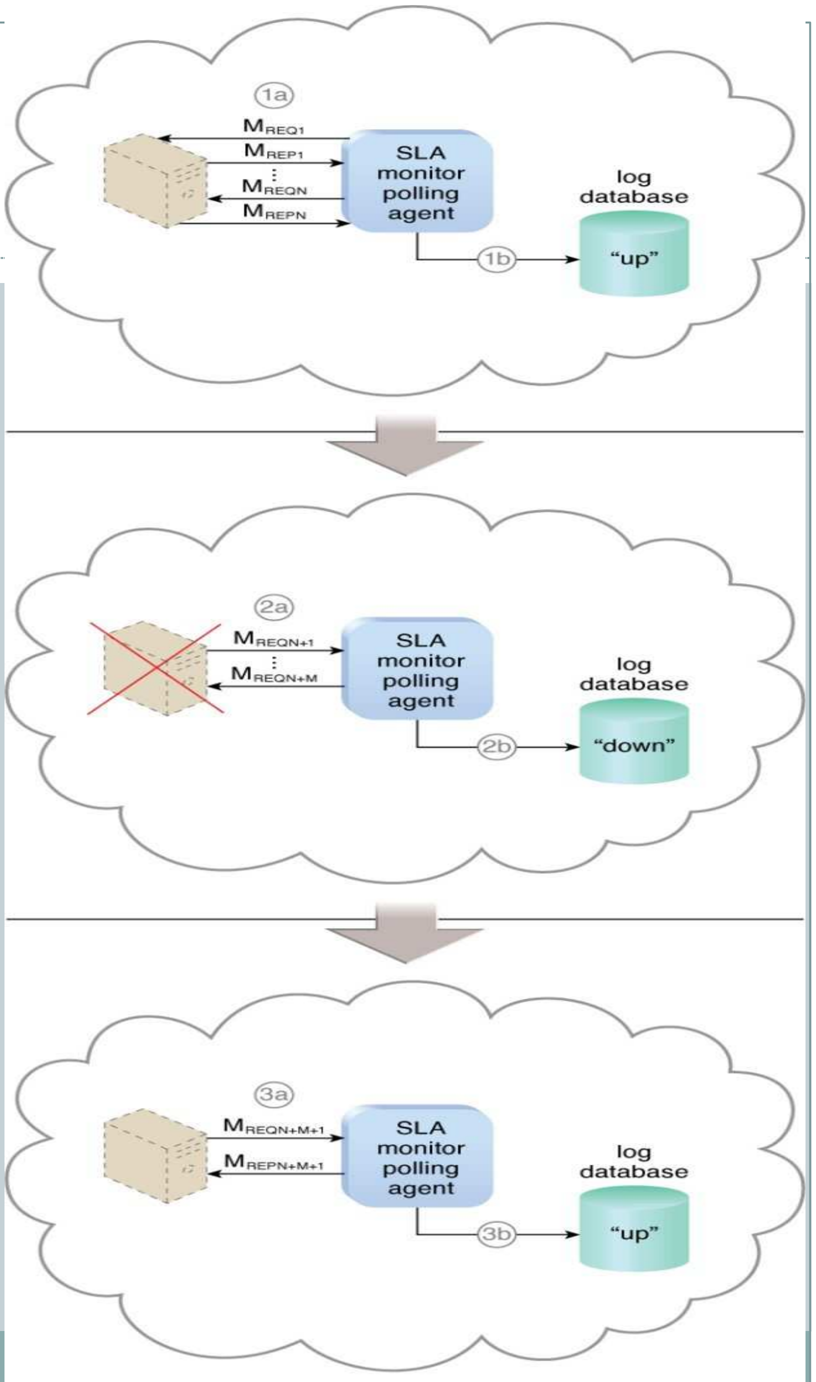


Figure 8.7 (2/2)

- The SLA monitor polls the cloud service that sends polling request messages (MREQN+1 to MREQN+M). Polling response messages are not received (2a).
- The response messages continue to time out, so the SLA monitor stores the "down" time-time period of all polling cycles N+1 to N+M-in the log database (2b).
- The SLA monitor sends a polling request message (MREQN+M+1) and receives the polling response message (MREPN+M+1) (3a).
- The SLA monitor stores the "up" time in the log database (3b).



SLA for Virtual Servers in DTGOV

- The standard SLA for virtual servers in DTGOV's leasing agreements defines a minimum of 99.95% by using 2 tracking agent: monitor polling agent and monitoring agent.
- 3 types of events generated by monitor polling agent:
  1. PS_Timeout
  2. PS_Unreachable
  3. PS_Reachable
- 3 types of events generated by monitoring agent:
  1. VM_Unreachable
  2. VM_Failure
  3. VM_Reachable

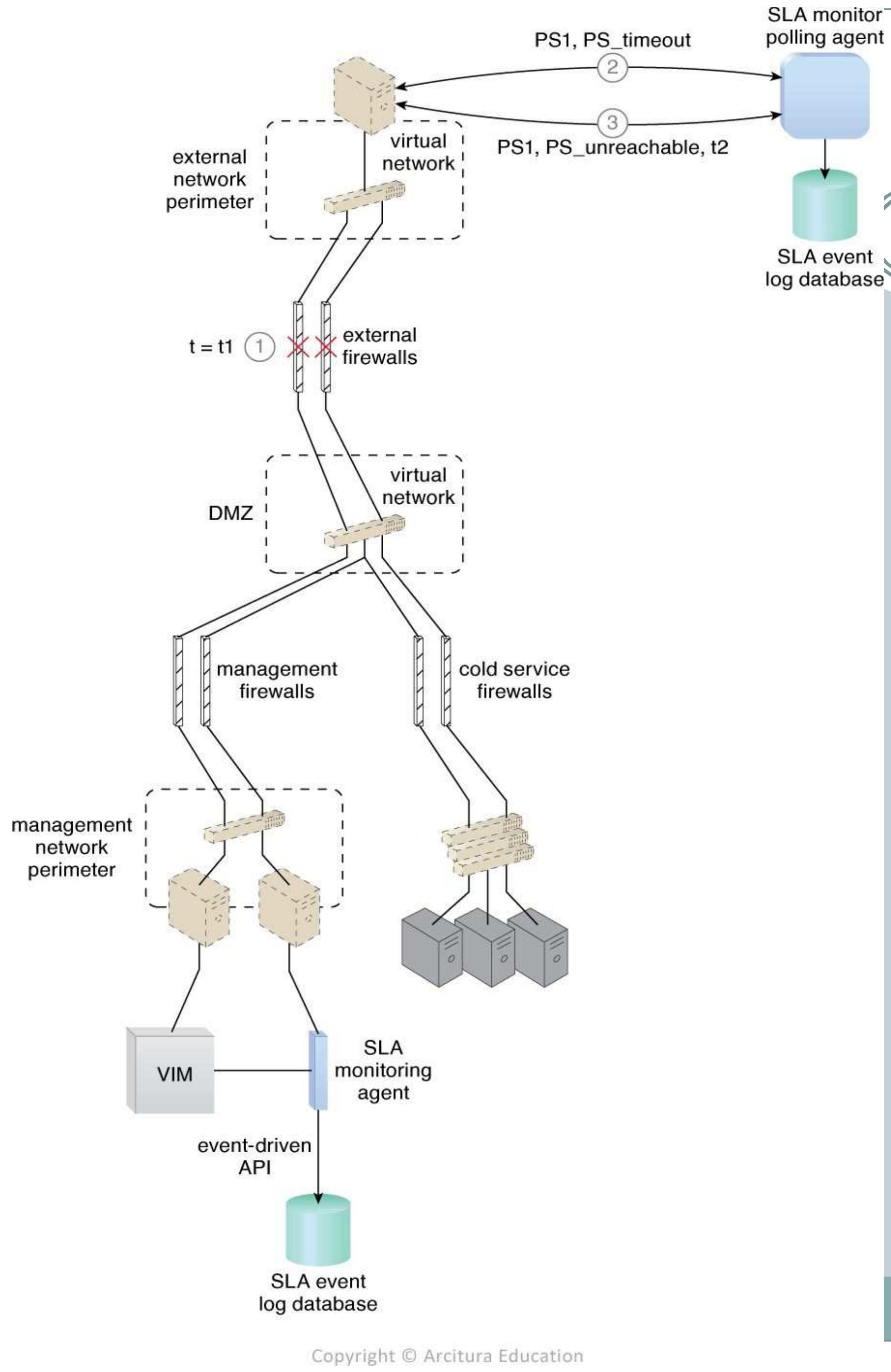


Figure 8.8

- At timestamp = t1, a firewall cluster has failed and all of the IT resources in the data center become unavailable (1).
- The SLA monitor polling agent stops receiving responses from physical servers and starts to issue PS_timeout events (2).
- The SLA monitor polling agent starts issuing PS_unreachable events after three successive PS_timeout events. The timestamp is now t2 (3).

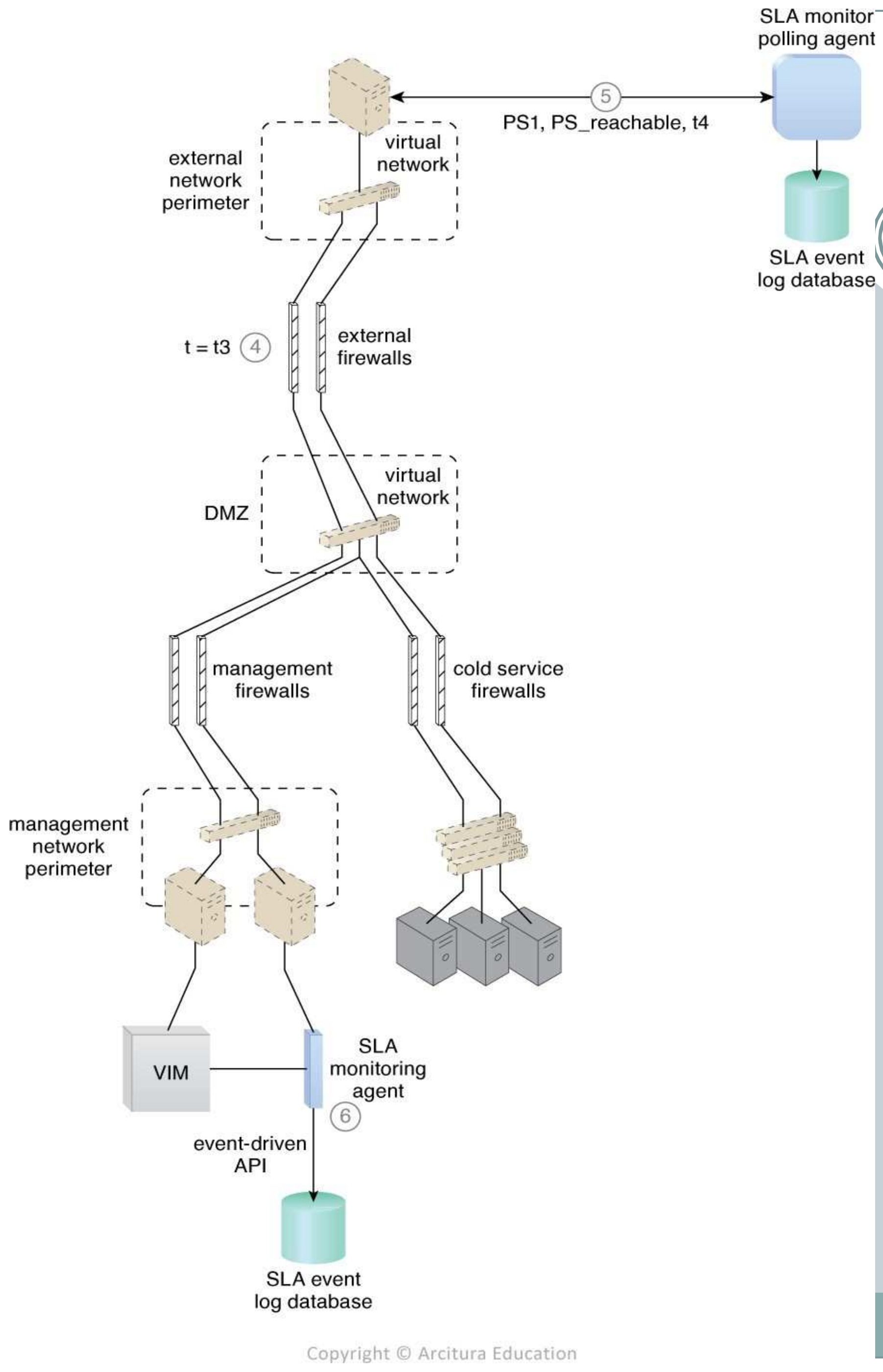
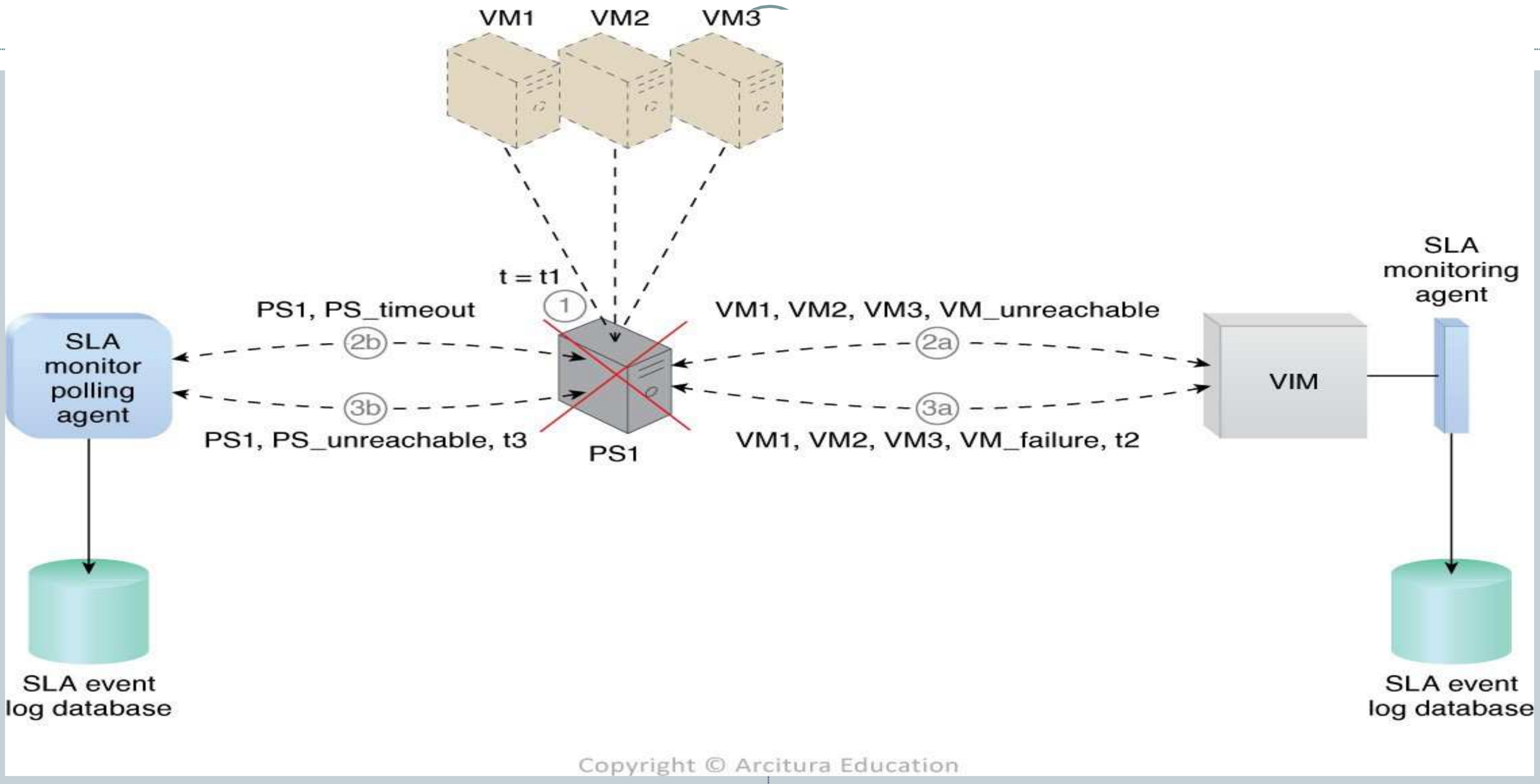


Figure 8.9

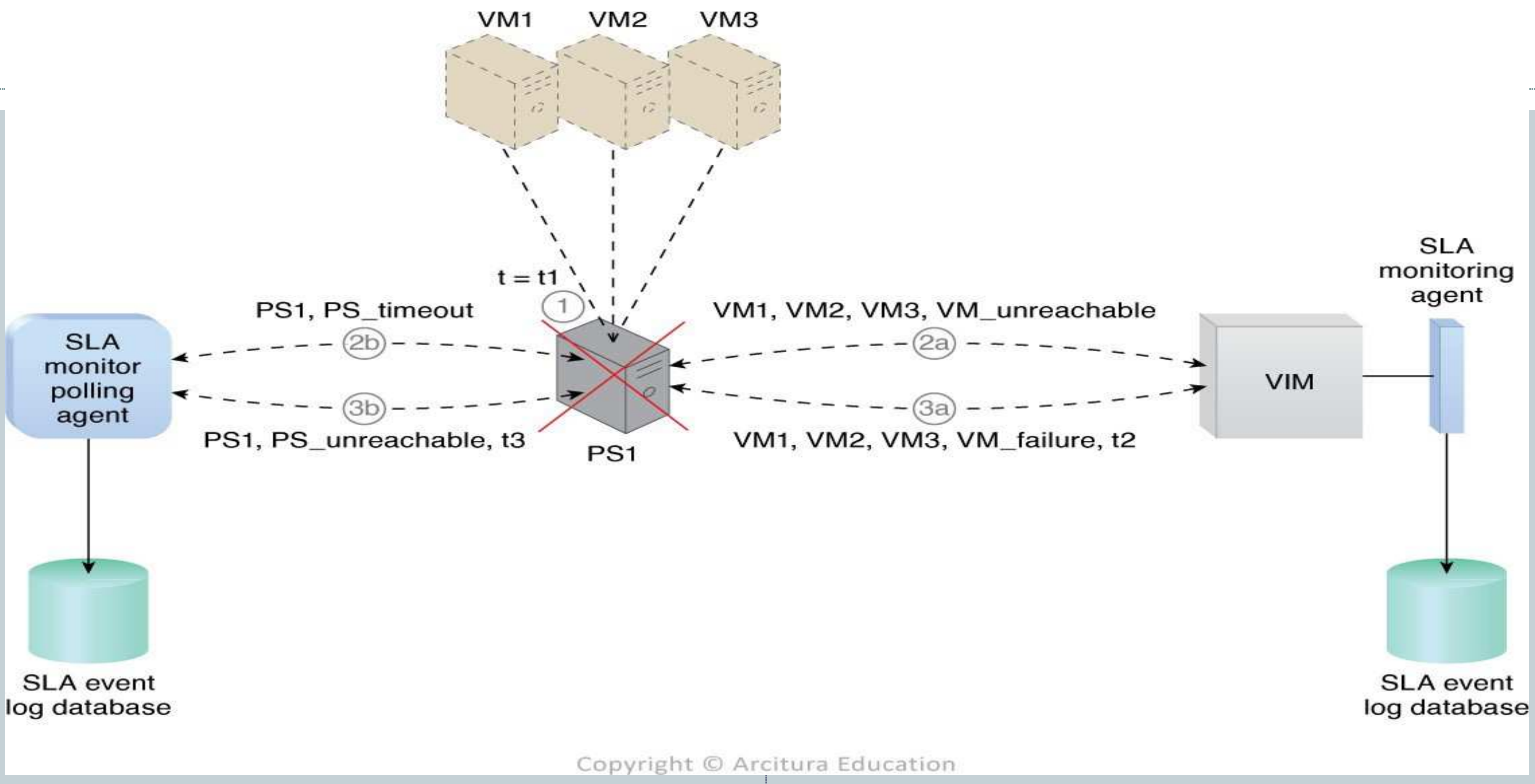
- The IT resource becomes operational at timestamp = t3 (4).
- The SLA monitor polling agent receives responses from the physical servers and issues PS_reachable events. The timestamp is now t4 (5).
- The SLA monitoring agent did not detect any unavailability since the communication between the VIM platform and physical servers was not affected by the failure (6).

Figure 8.10 (1/3)



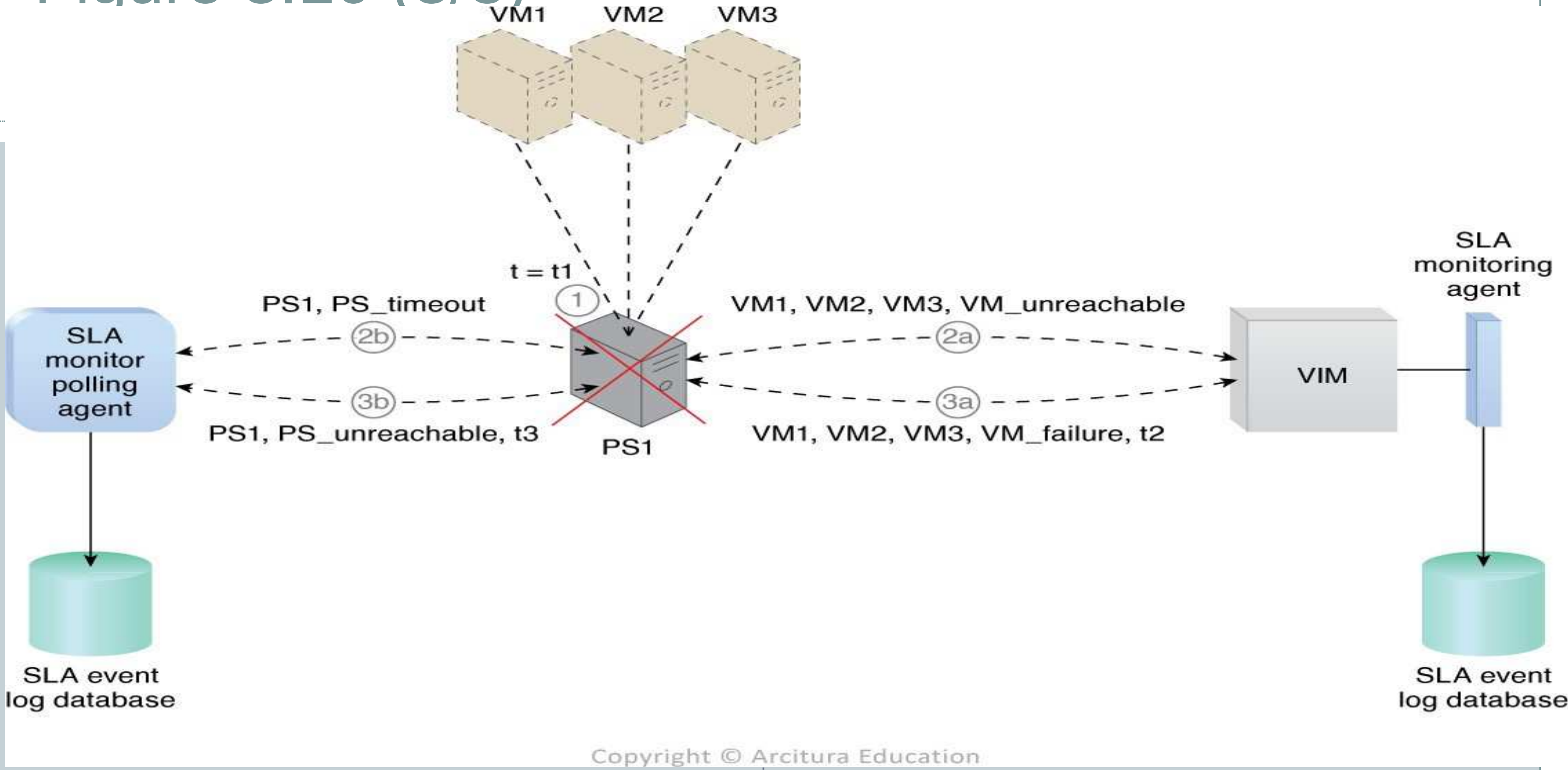
- *At timestamp = t1, the physical host server has failed and becomes unavailable (1).*

Figure 8.10 (2/3)



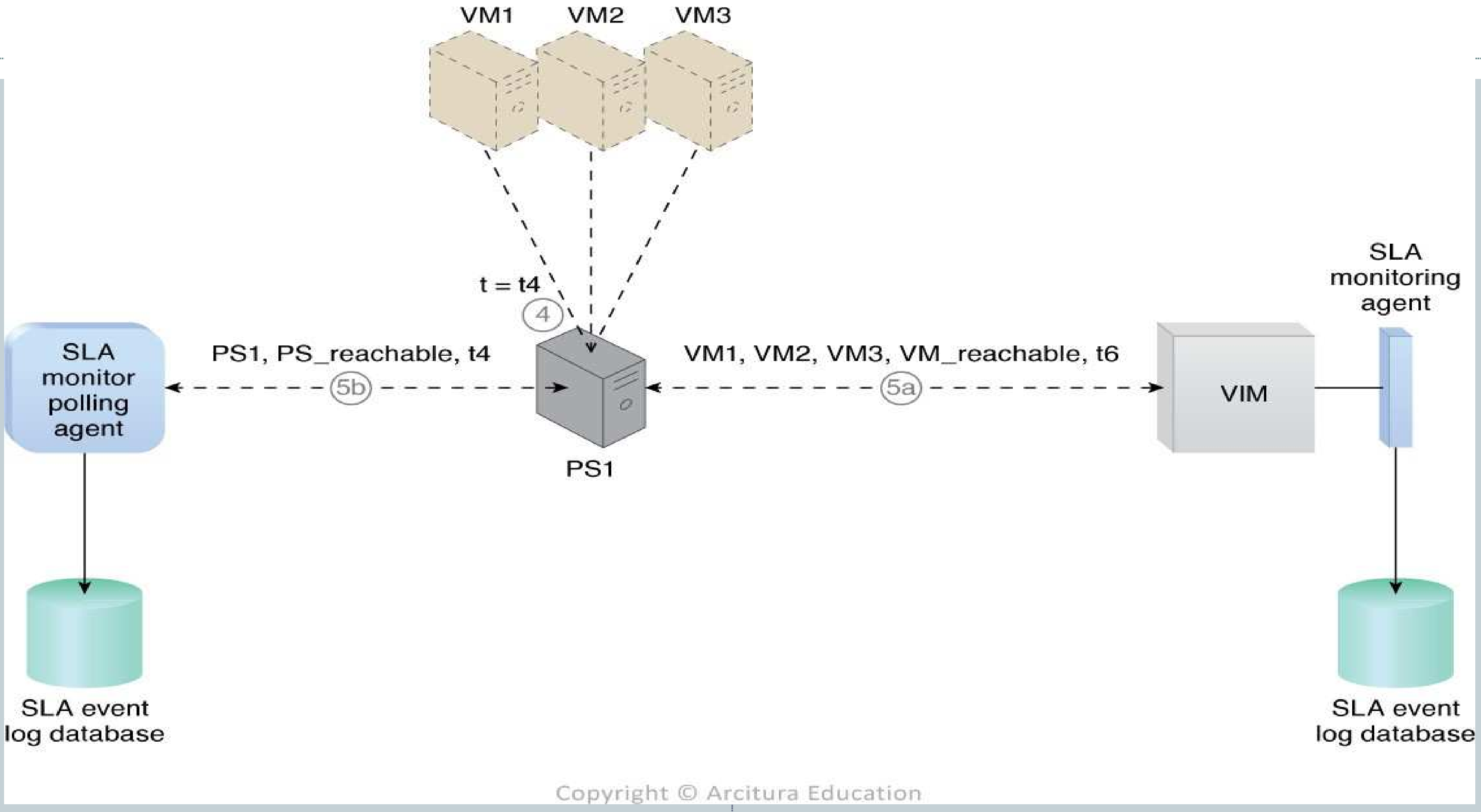
- *The SLA monitoring agent captures a VM_unreachable event that is generated for each virtual server in the failed host server (2a).*
- *The SLA monitor polling agent stops receiving responses from the host server and issues PS_timeout events (2b).*

Figure 8.10 (3/3)



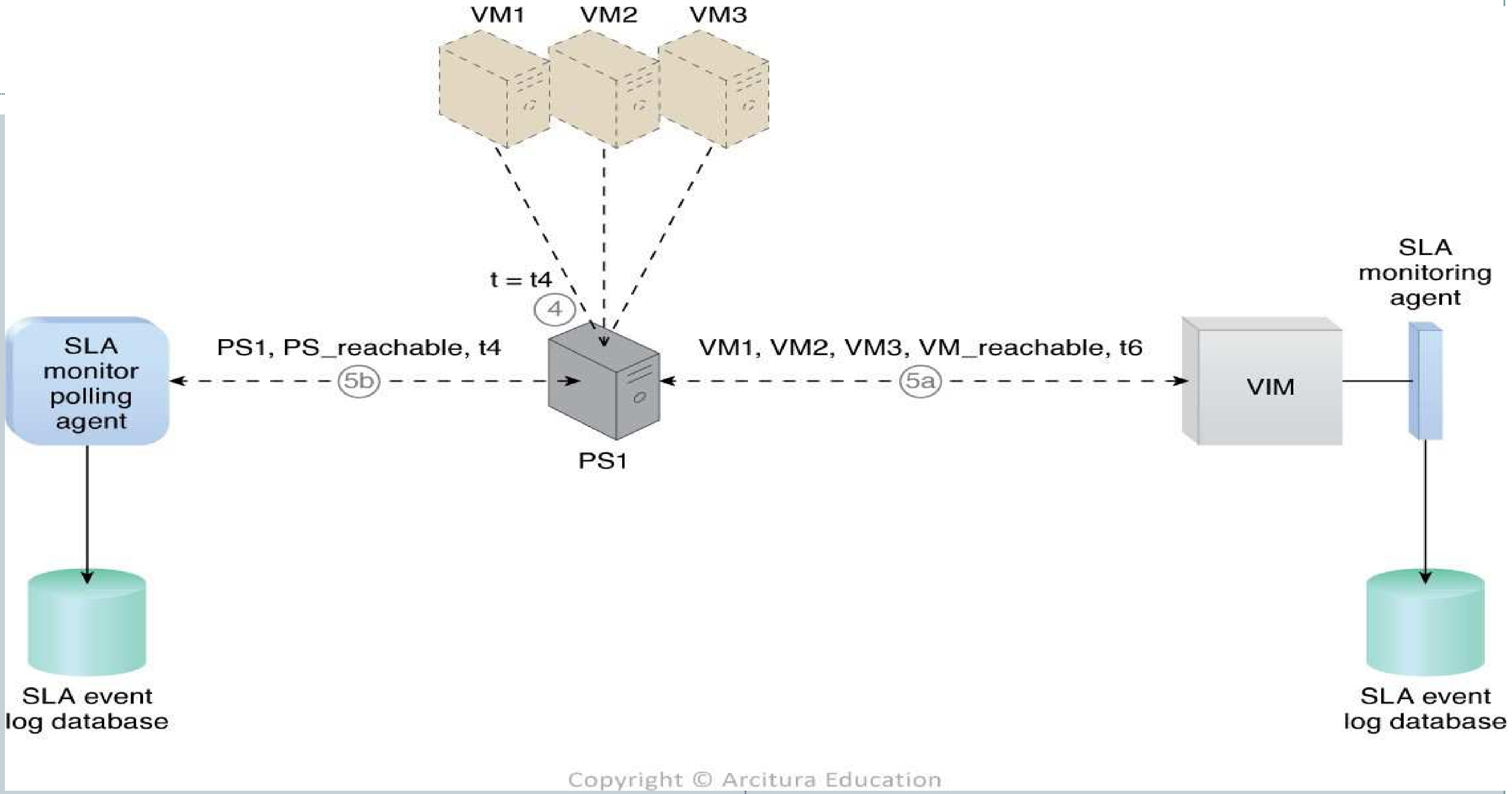
- *At timestamp = t2, the SLA monitoring agent captures a VM_failure event that is generated for each of the failed host server's three virtual servers (3a).*
- *The SLA monitor polling agent starts to issue PS_unavailable events after three successive PS_timeout events at timestamp = t3 (3b).*

Figure 8.11



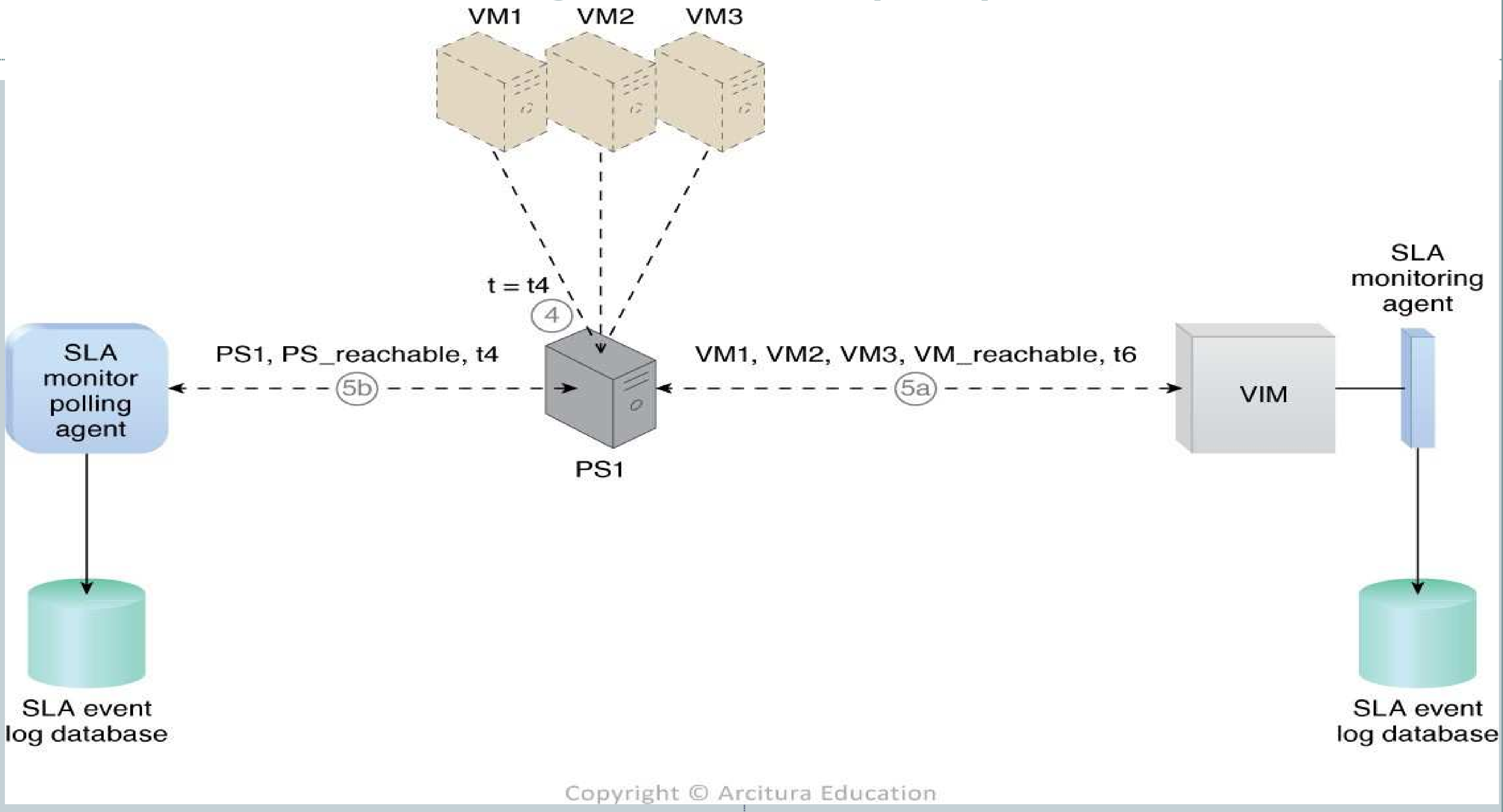
- *The host server becomes operational at timestamp = t4 (4).*

Figure 8.11 (2/3)



- *The SLA monitor polling agent receives responses from the physical server and issues PS_reachable events at timestamp = t5 (5a).*
- *At timestamp = t6, the SLA monitoring agent captures a VM_reachable*

Figure 8.11 (3/3)



- *The SLA management system calculates the unavailability period that affected all of the virtual servers as t6 - t2 (6).*

8.4 Pay-Per-Use Monitor

- The pay-per-use monitor mechanism measures cloud-based IT resource usage in accordance with predefined pricing parameters and generates usage logs for fee calculations and billing purposes.
- Some typical monitoring variables are:
  - request/response message quantity
  - transmitted data volume
  - bandwidth consumption
- The data collected by the pay-per-use monitor is processed by a billing management system that calculates the payment fee.

Figure 8.12 (1/2)

Pay-per-use implemented as a resource agent.

- A cloud consumer requests the creation of a new instance of a cloud service (1).
- The IT resource is instantiated and the pay-per-use monitor receives a "start" event notification from the resource software (2).
- The pay-per-use monitor stores the value timestamp in the log database (3).

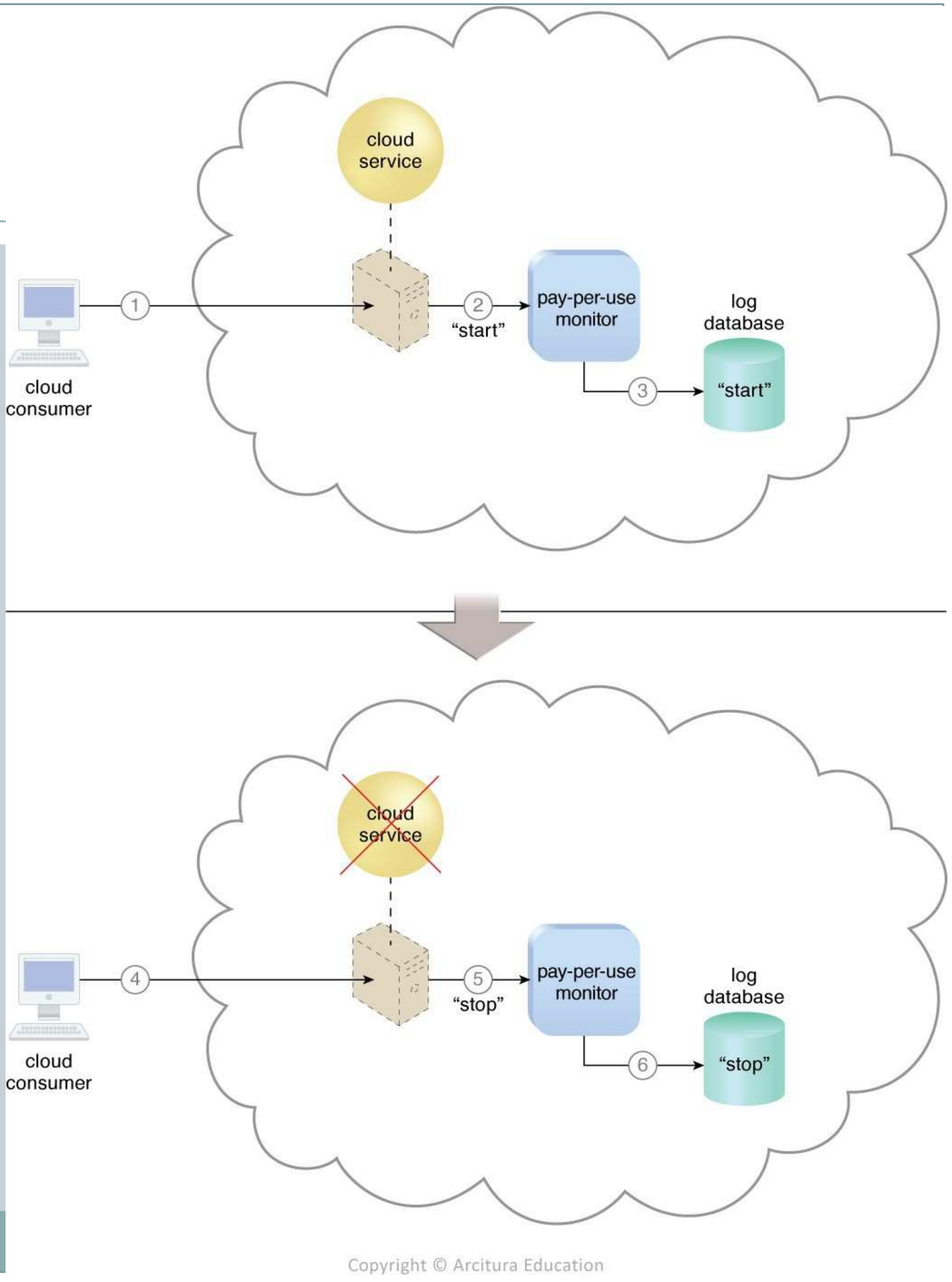
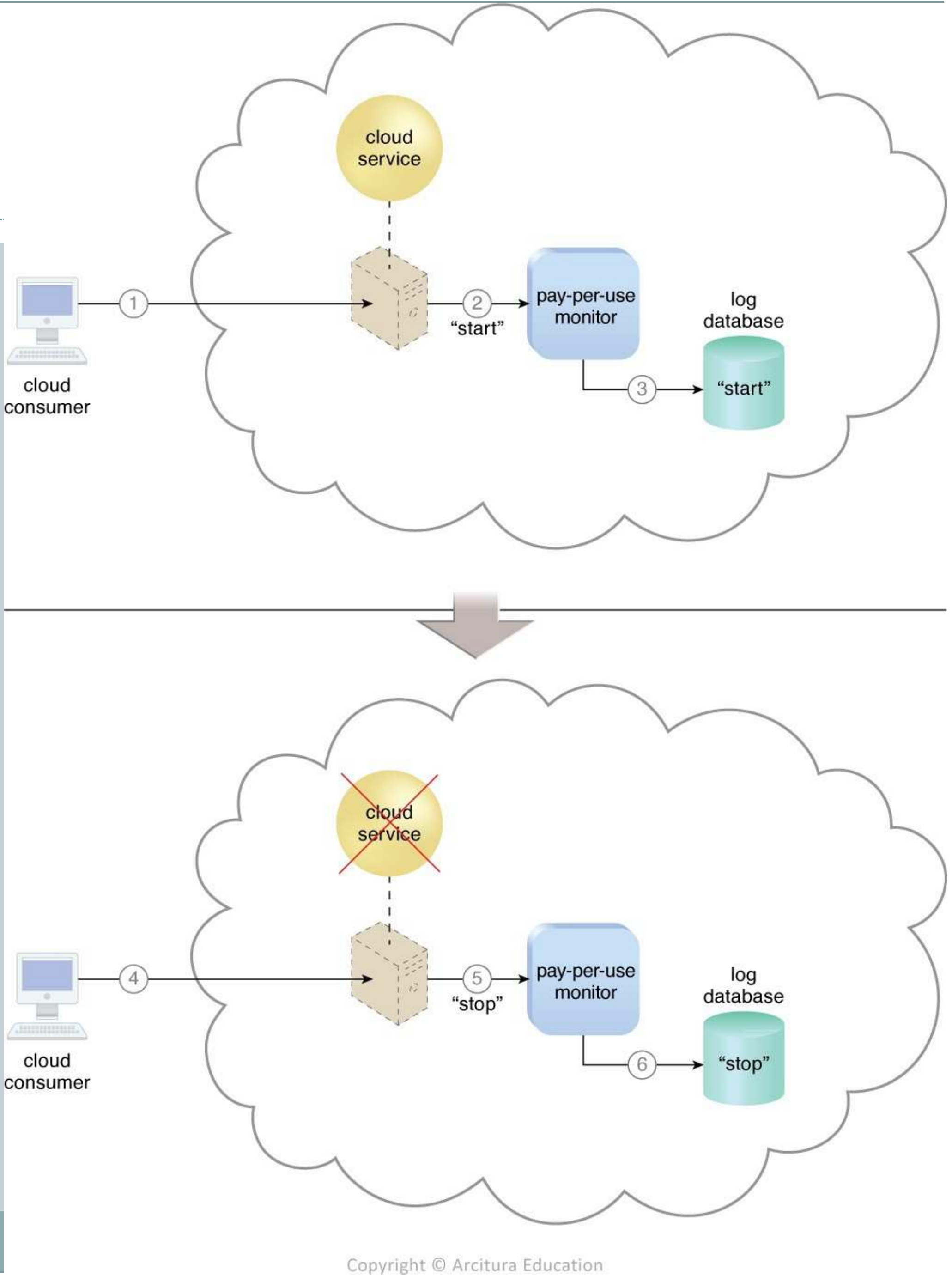


Figure 8.12 (2/2)

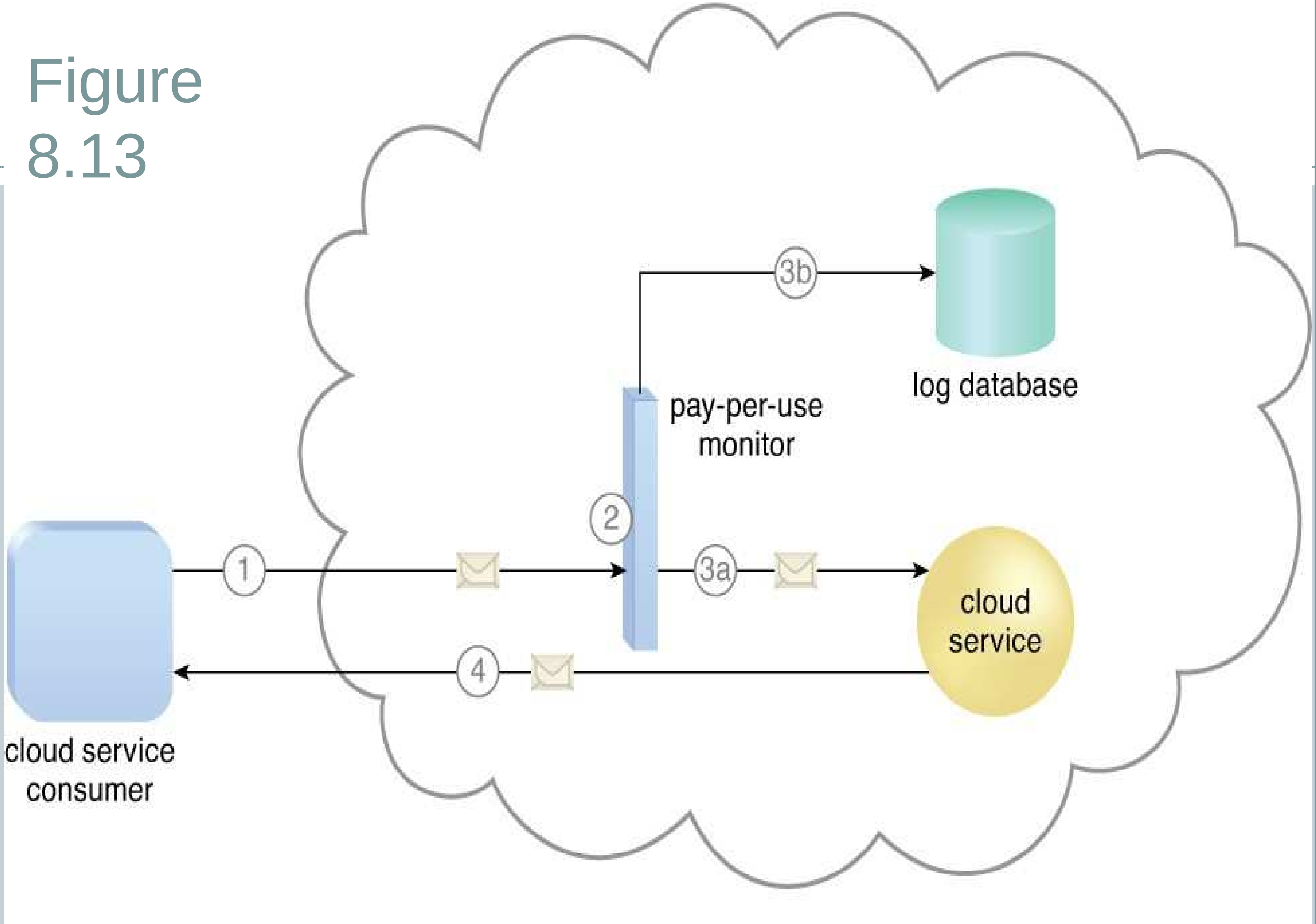
- The cloud consumer later requests that the cloud service instance be stopped (4).
- The pay-per-use monitor receives a "stop" event notification from the resource software (5) and stores the value timestamp in the log database (6).



Pay-Per-Use Monitor Alternatives

- In Figure 8.12, pay-per-use is implemented as a resource agent. An alternative is to implement it as a monitor agent, as shown in Figure 8.13.
- In Figure 8.13,
  1. *A cloud service consumer sends a request message to the cloud service (1).*
  2. *The pay-per-use monitor intercepts the message (2), forwards it to the cloud service (3a), and stores the usage information in accordance with its monitoring metrics (3b).*
  3. *The cloud service forwards the messages back to the cloud consumer to provide the requested service (4).*

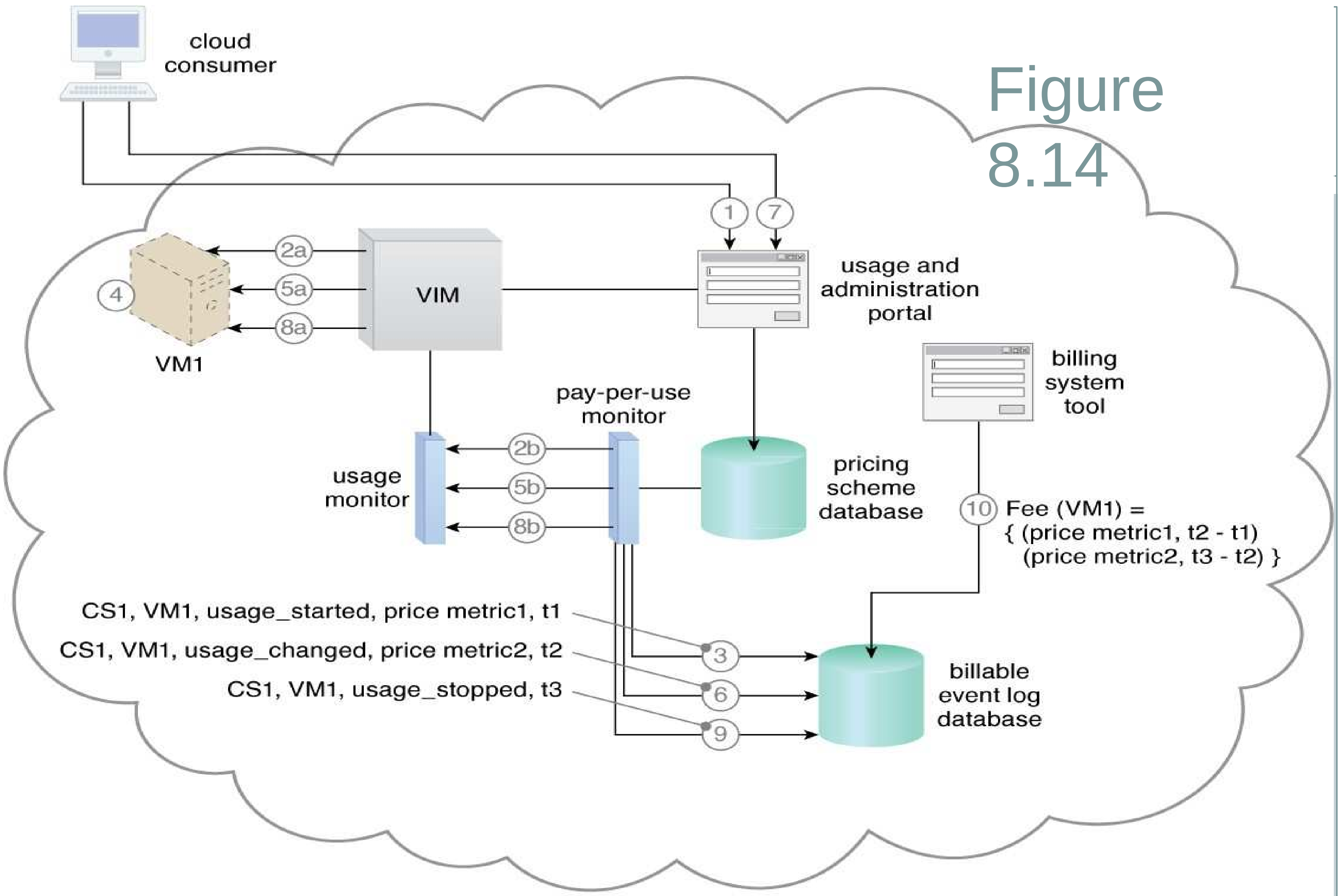
Figure 8.13


log database
3b
pay-per-use monitor
2
1
3a
4
cloud service
cloud service consumer

Pay-Per-Use Case in DTGOV

- DTGOV install 2 database: the billing event database and the pricing scheme database to support billable events and customizable pricing model.
- The pay-per-use monitor polling agent periodically supplies the billing system with billable event information.
- A separate monitoring agent provides further supplemental billing-related data:
  - Cloud consumer subscription type
  - Resource usage category
  - Resource usage quota consumption

Figure 8.14


cloud consumer
1
7
2a
5a
8a
4
VM1
VIM
usage and administration portal
billing system tool
pay-per-use monitor
usage monitor
2b
5b
8b
pricing scheme database
10
Fee (VM1) = { (price metric1, t2 - t1) (price metric2, t3 - t2) }
CS1, VM1, usage_started, price metric1, t1
CS1, VM1, usage_changed, price metric2, t2
CS1, VM1, usage_stopped, t3
3
6
9
billable event log database

Explanation of Figure 8.14 (1/3)

1. The cloud consumer (CS_ID = CS1) creates and starts a virtual server (VM_ID = VM1) of configuration size type 1 (VM_TYPE = type1) (1).
2. The VIM creates the virtual server instance as requested
   (2a).
5. The VIM's event-driven API generates a resource usage event with timestamp = t1, which is captured and forwarded to the pay-per-use monitor resource agent by the usage monitoring agent (2b).
6. The pay-per-use monitor resource agent interacts with the pricing scheme database to identify the chargeback and usage metrics that apply to the resource usage. A "started usage" billable event is generated and stored in the billable event log database (3).

Explanation of Figure 8.14 (2/3)

5. The virtual server's usage increases and reaches the auto-scaling threshold (4).
6. The VIM scales up the virtual server (VM1) (5a) 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 forwarded to the pay-per-use monitor resource agent by the usage monitoring agent (5b).
7. The pay-per-use monitor resource agent interacts with the pricing scheme database to identify the chargeback and usage metrics that apply to the updated IT resource usage. A "changed usage" billable event is generated and stored in the pay-per-use log database (6).
8. The cloud consumer shuts down the virtual server (7) and the VIM stops Virtual Server (VM1) (8a).

Explanation of Figure 8.14 (3/3)

9 The VIM's event-driven API generates a resource usage event with timestamp = t3, which is captured and forwarded to the pay-per-use monitor resource agent by the usage monitoring agent (8b).

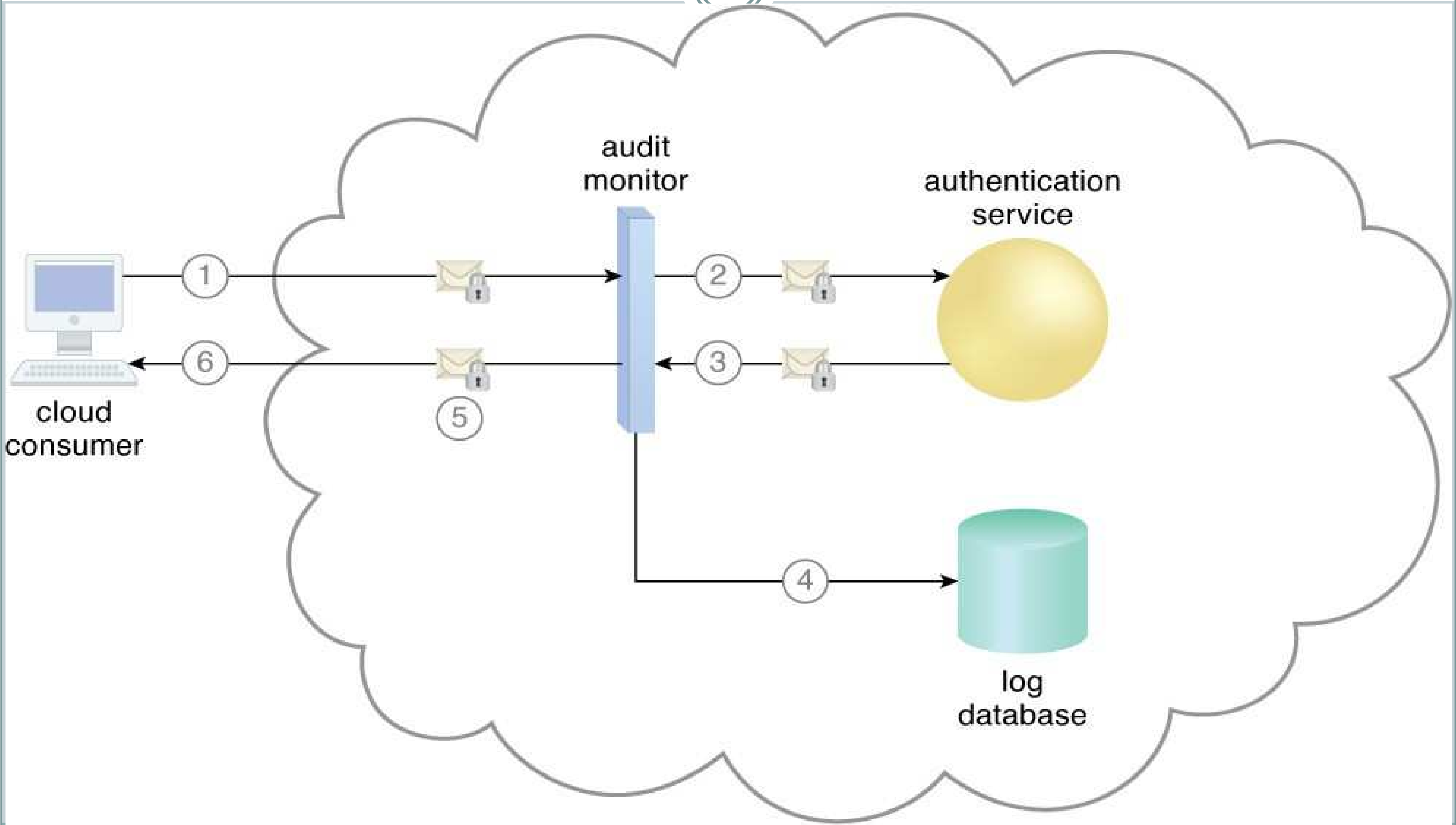
10. The pay-per-use monitor resource agent interacts with the pricing scheme database to identify the chargeback and usage metrics that apply to the updated IT resource usage. A "finished usage" billable event is generated and stored in the pay-per-use log database (9).

11. The billing system tools access the log database and calculate the total usage fee for the virtual server as (Fee(VM1)) (10).

8.5 Audit Monitor

- The audit monitor mechanism is used to collect audit tracking data for networks and IT resources in support (or dictated by) regulatory and contractual obligations.
- Figure 8.15 depicts an audit monitor implemented as a monitoring agent that intercepts login requests and stores the requestor's security credentials, as well as failed and successful login attempts, in a log database.

Figure 8.15


audit monitor
authentication service
1
2
3
4
5
6
cloud consumer
log database

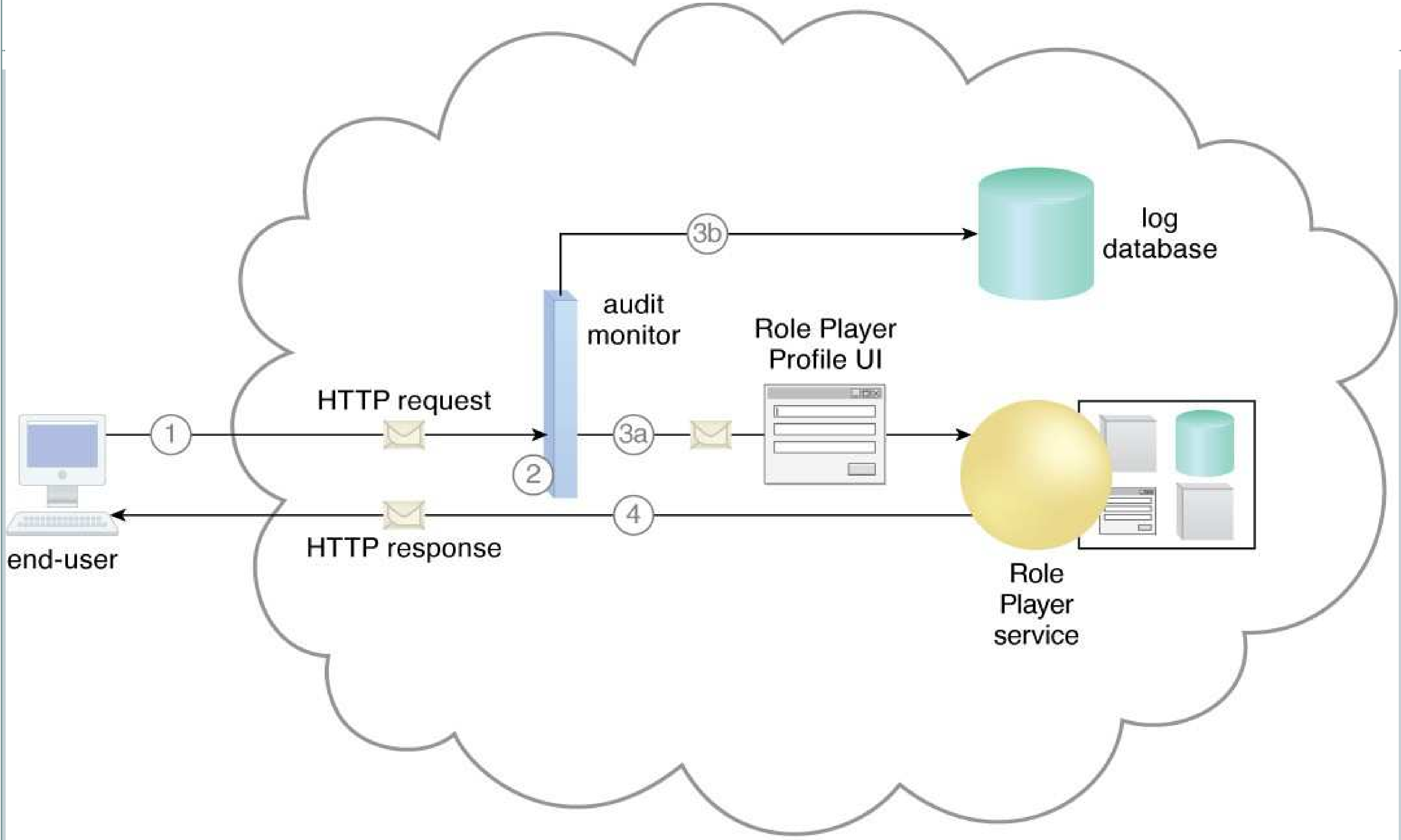
Procedures of an Audit Monitor

1. A cloud service consumer requests access to a cloud service by sending a login request message with security credentials (1).
2. The audit monitor intercepts the message (2) and forwards the message to the authentication service (3).
3. The authentication service processes the security credentials. A response message is generated for the cloud service consumer, in addition to the results from the login attempt (4).
4. The audit monitor intercepts the response message and stores the entire collected login event details in the log database, as per the organization's audit policy requirements (5).
5. Access has been granted, and a response is sent back to the cloud service consumer (6).

Audit Monitor in Innovartus

- Innovartus' legal department is working on the issue that the application can either not be accessed by users or access is free of charge depending on various geographical regions.
- Innovartis asks its cloud provider to establish an audit monitoring system to intercept each inbound message, analyze the HTTP header, and collect the origin of the end-user.

Figure


3b
log database
audit monitor
Role Player Profile UI
HTTP request
1
3a
2
4
end-user
HTTP response
Role Player service

Audit Monitor Procedures in Fig 8.16

- An end-user attempts access to the Role Player cloud service (1).
- An audit monitor transparently intercepts the HTTP request message and analyzes the message header to determine the geographical origin of the end-user (2).
- The audit monitoring agent determines that the end-user is from a region that Innovartus is not authorized to charge a fee for access to the application. The agent forwards the message to the cloud service (3a) and generates the audit track information for storage in the log database (3b).
- The cloud service receives the HTTP message and grants the end-user access at no charge (4).

8.6 Failover System (1/2)

- The failover system mechanism is used to increase the reliability and availability of IT resources by using established clustering technology to provide redundant implementations.
- A failover system is configured to automatically switch over to a redundant or standby IT resource instance whenever the current active IT resource becomes unavailable.

8.6 Failover System (2/2)

- Failover system come in two basic configurations:
  - Active-Active – redundant implementations of the IT resources is required, and load balancing is among active instance is necessary. (Figure 8.17-19)
  - Active-Passive – a standby or inactive implementation is activated to take over the processing from the IT resource that becomes unavailable and the corresponding workload is redirected to the instance taking over the operation. (Figure 8.20-22)

Figure

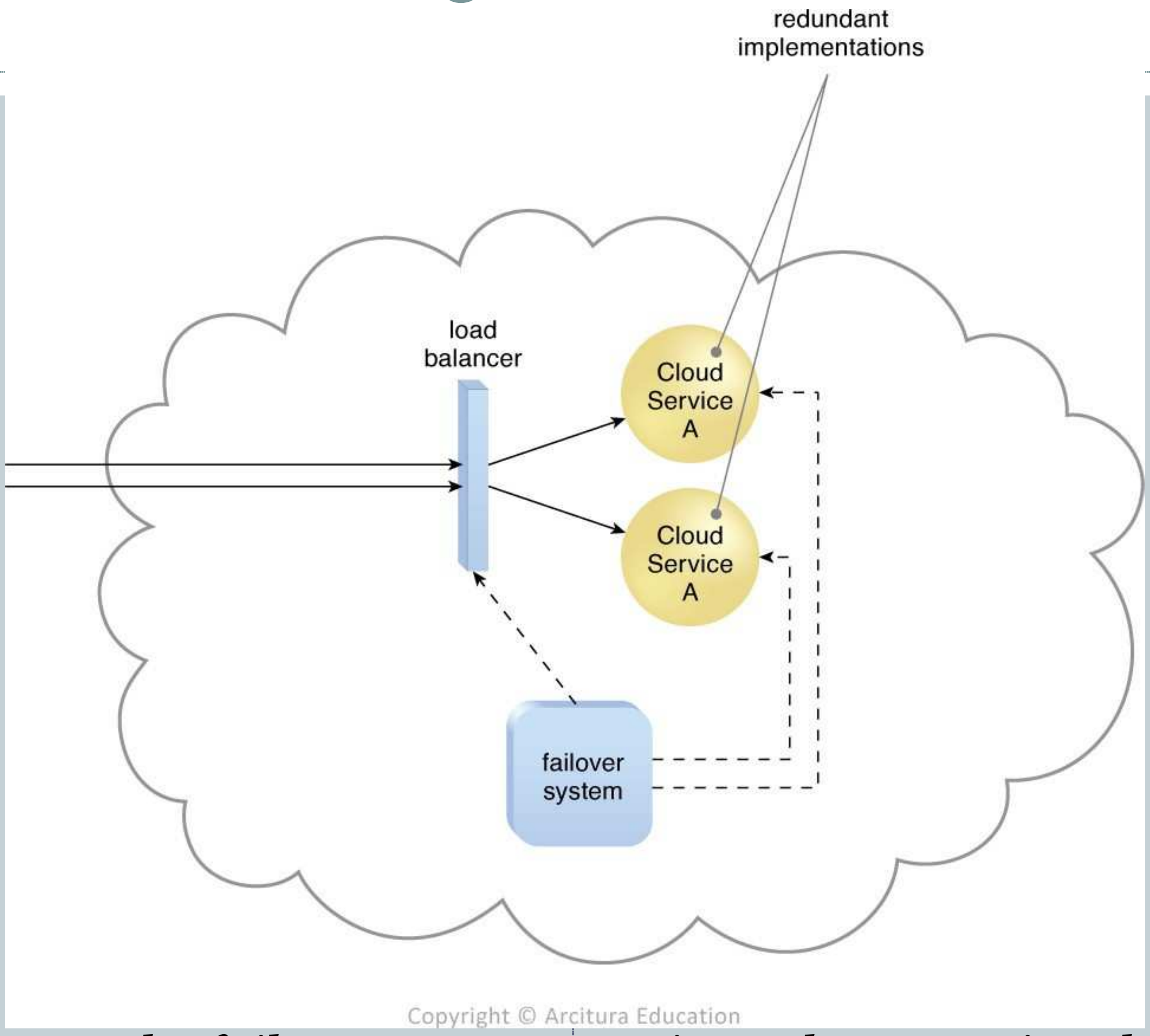


 Figure 8.17 - The failover system monitors the operational status of

Cloud Service A

Figure

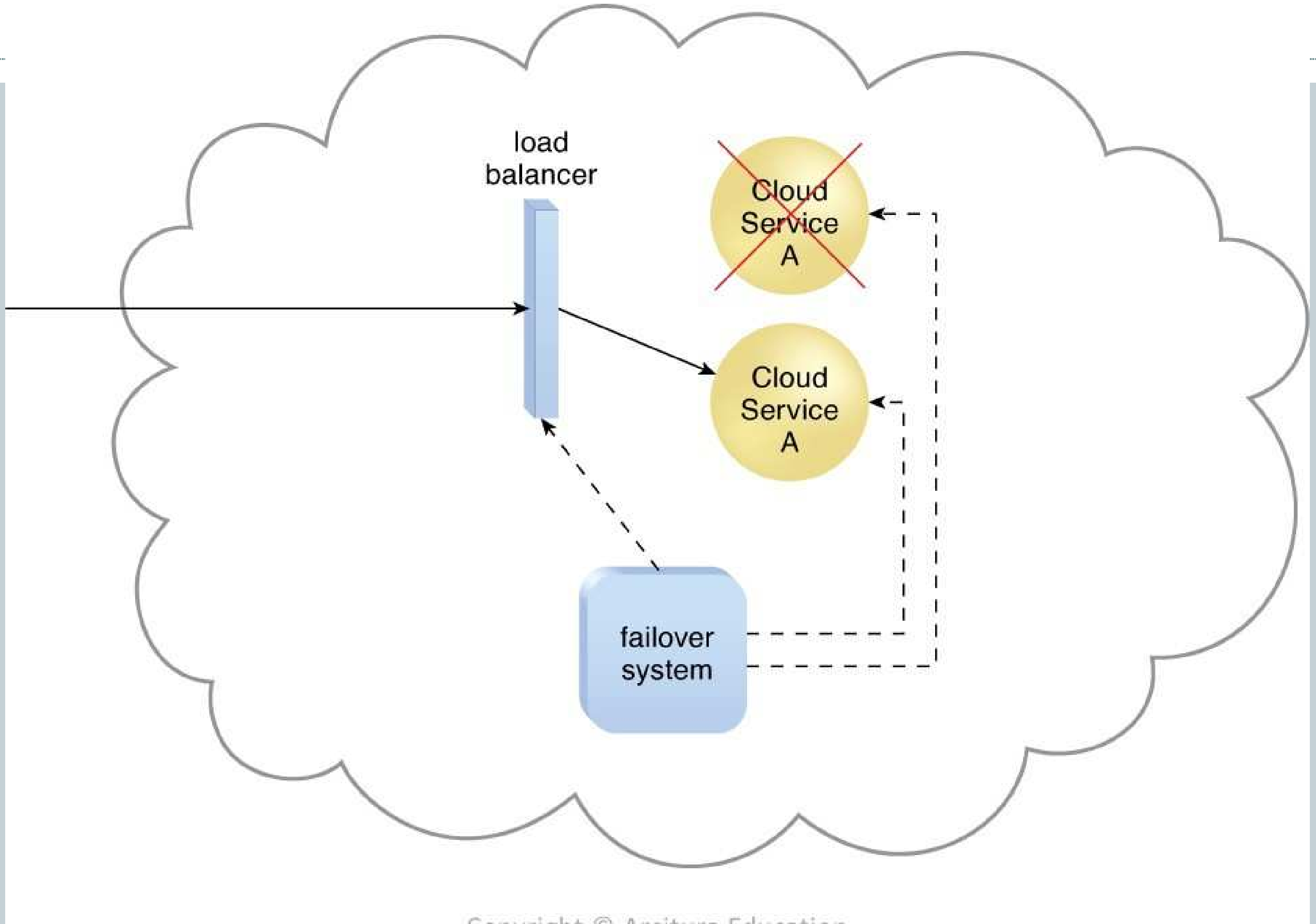
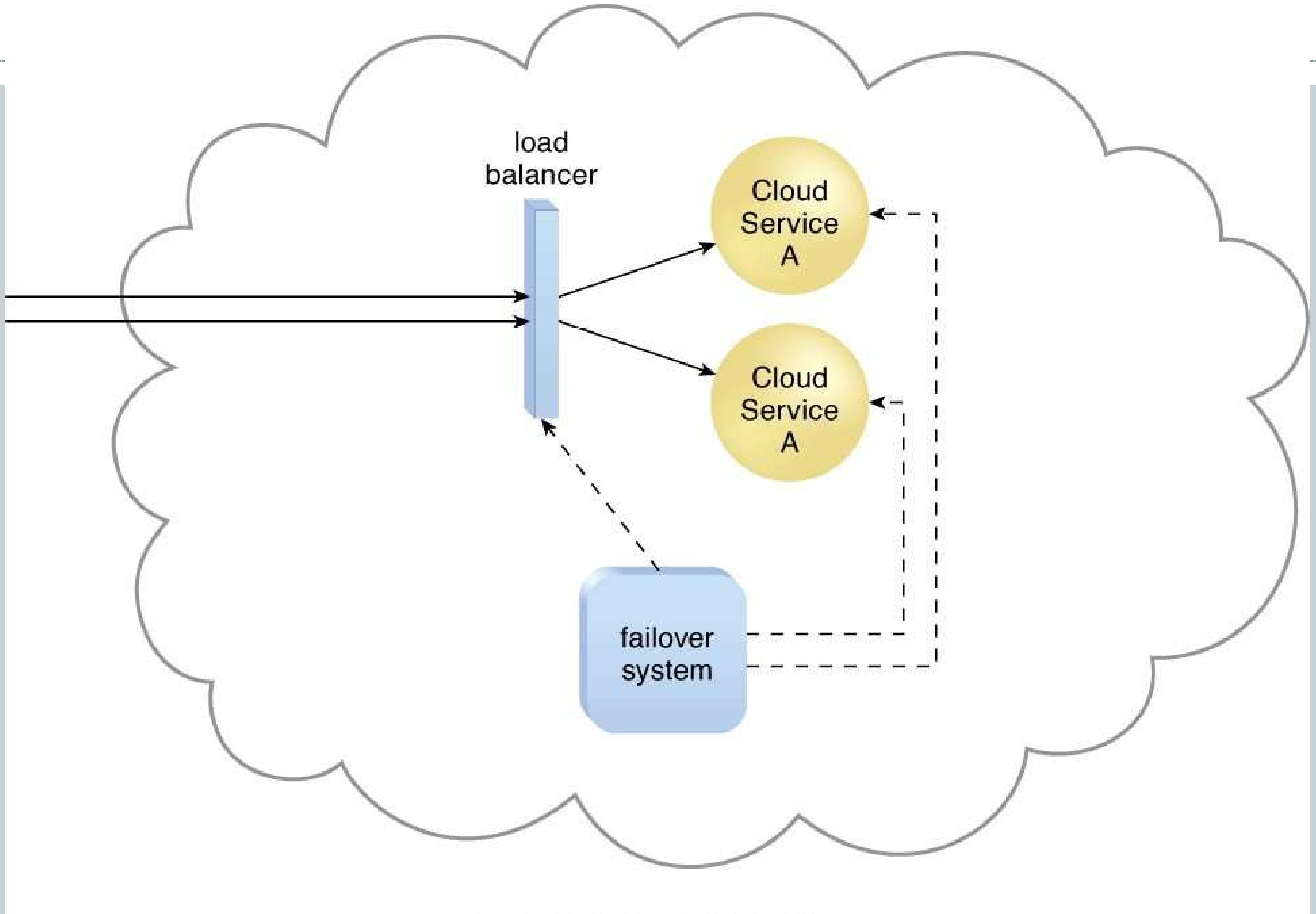


 Figure 8.18 - When a failure is detected, the failover system commands the load balancer to switch over the workload to the redundant Cloud Service A implementation.

Figure



- *Figure 8.19 - The failed Cloud Service A implementation is recovered or replicated into another operational resource. The failover system now commands the load balancer to distribute the workload again.*

Figure 8.20

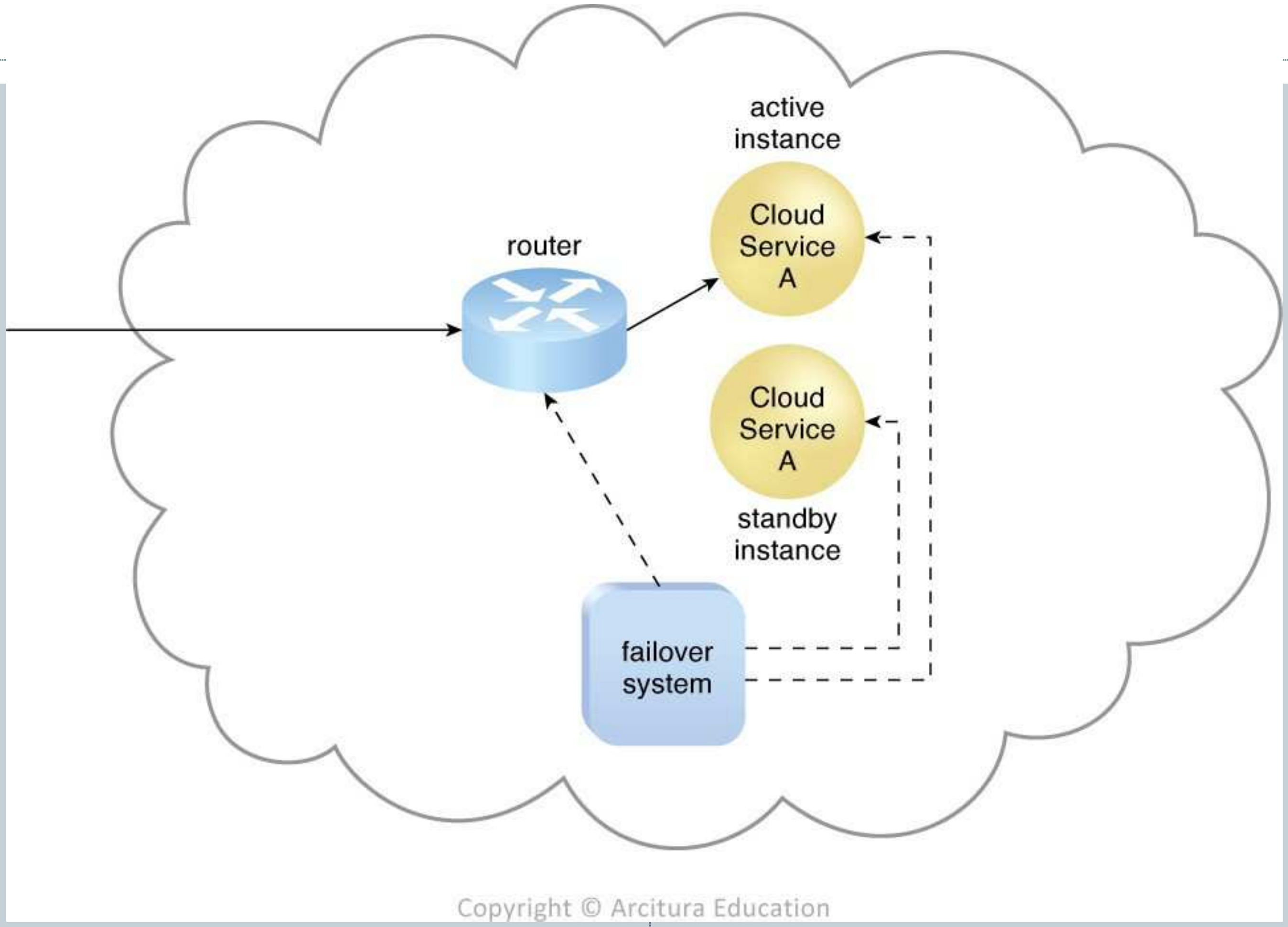
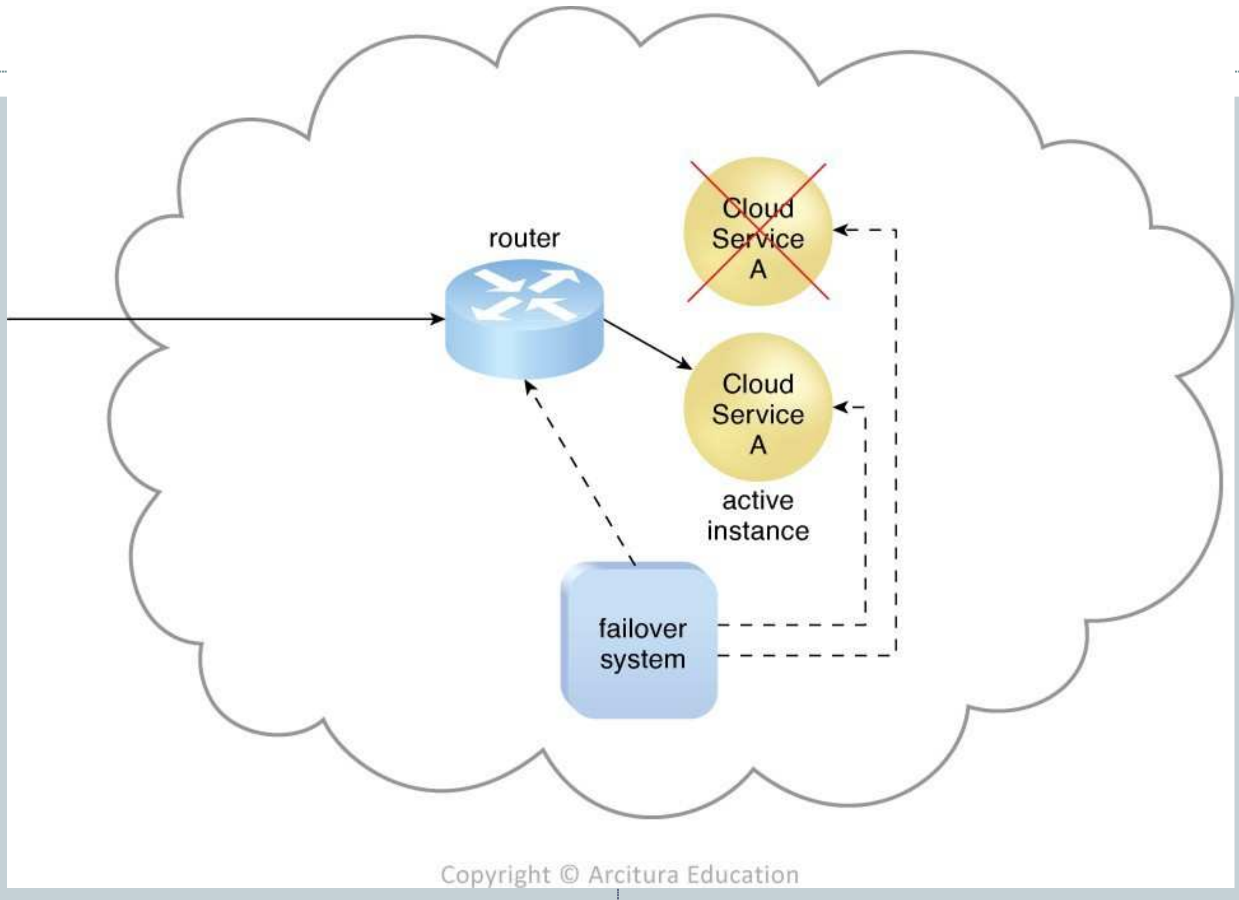


 Figure 8.20 - The failover system monitors the operational status of Cloud Service A. The Cloud Service A implementation acting as the active instance is receiving cloud service consumer requests.

Figure



- *Figure 8.21 - The Cloud Service A implementation acting as the active instance encounters a failure that is detected by the failover system, which subsequently activates the inactive Cloud Service A implementation and redirects the workload toward it. The newly invoked Cloud Service A implementation now assumes the role of active instance.*

Figure 8.22

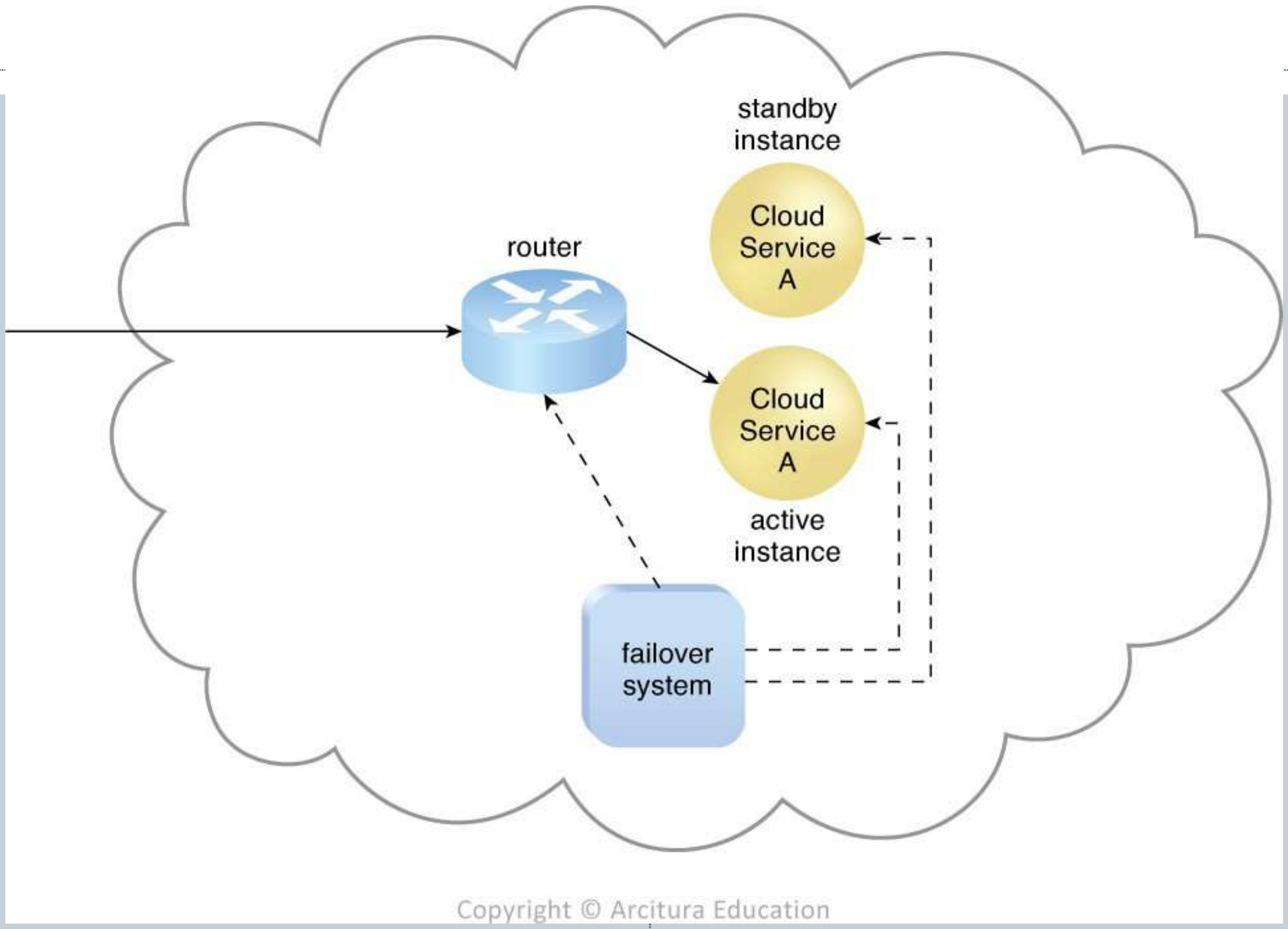
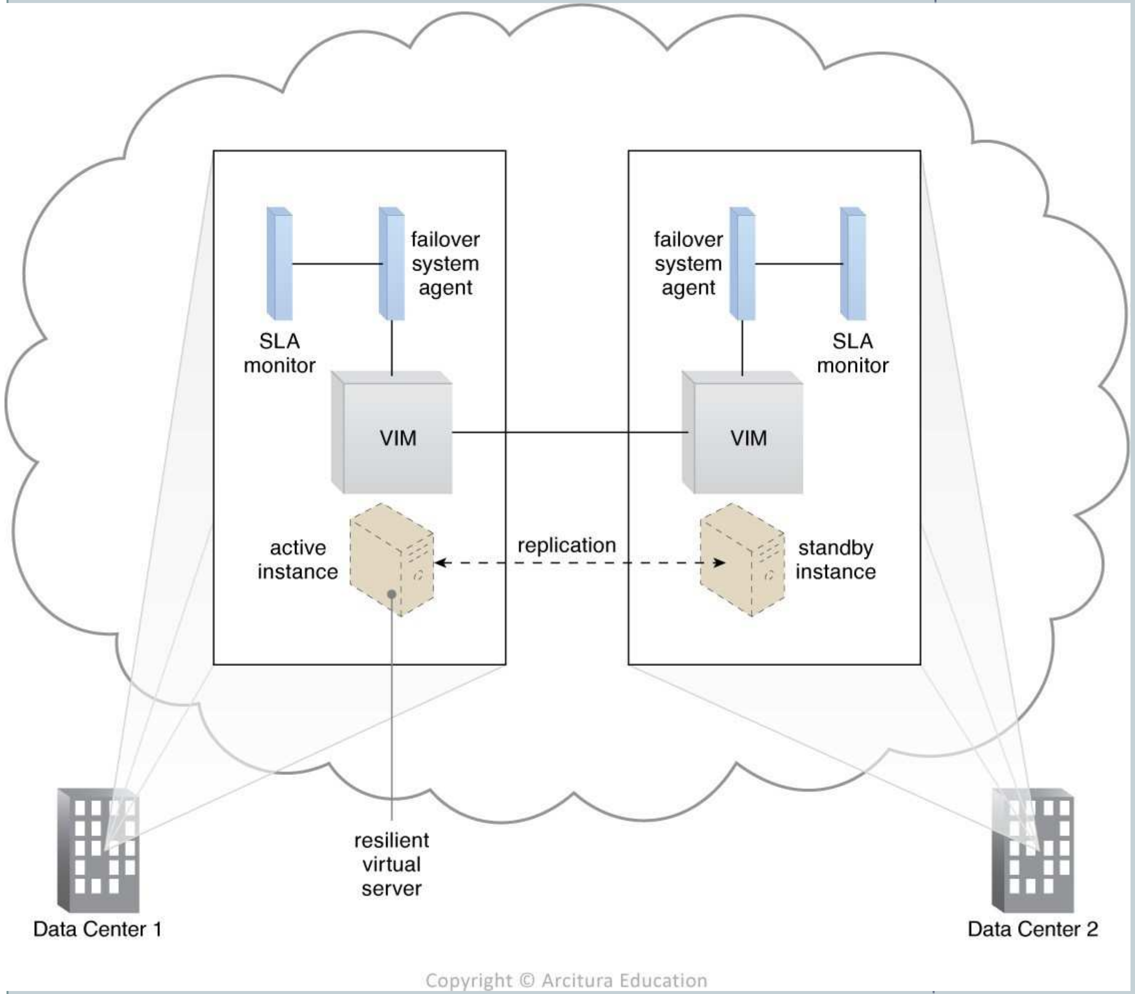


Figure 8.22 - The failed Cloud Service A implementation is recovered or replicated into another operational resource, and is now positioned as the standby instance while the previously invoked Cloud Service A continues to serve as the active instance.

Figure 8.23 (**DTGOV's Example**)

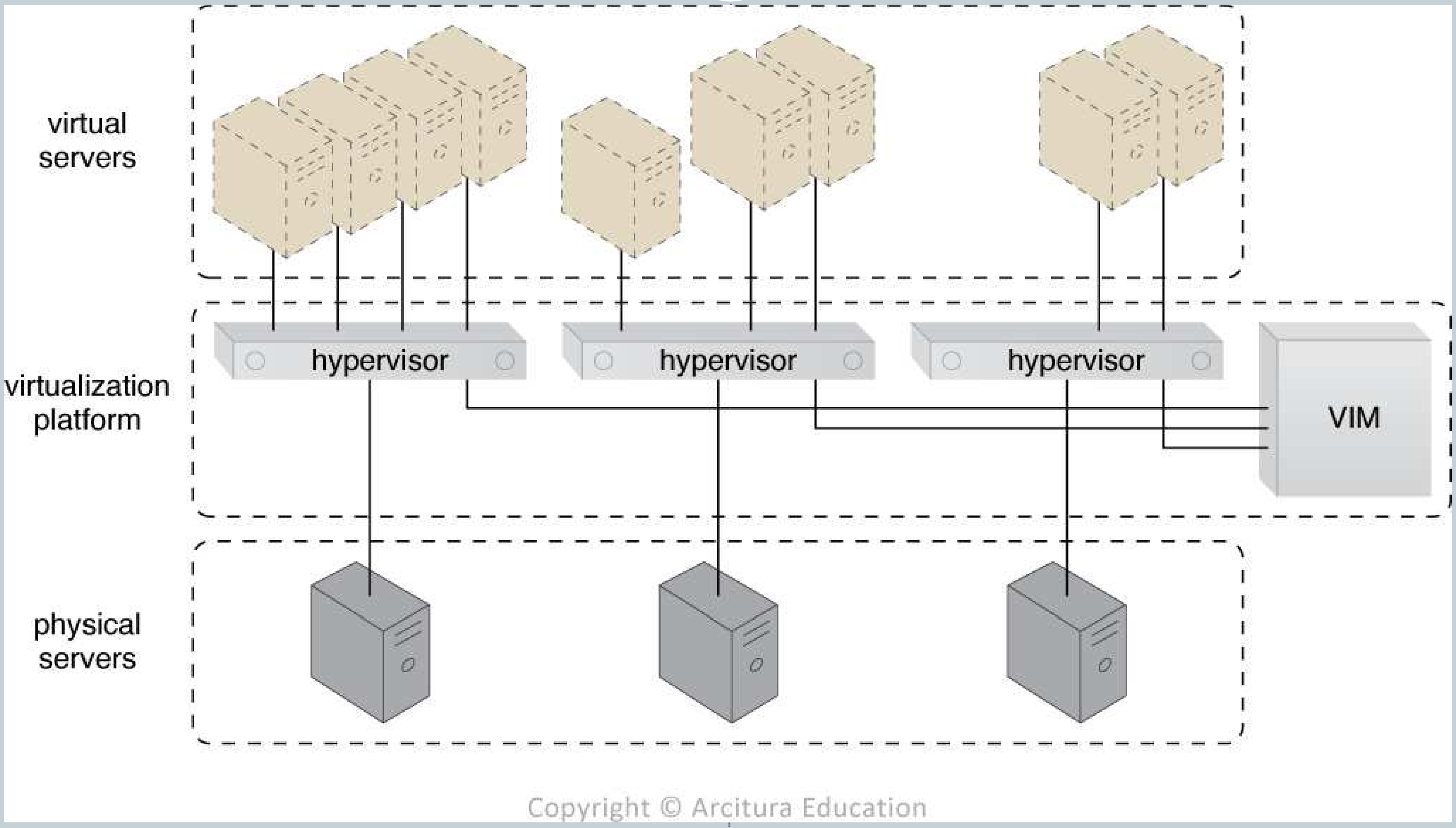


- *Figure 8.23 - A resilient virtual server is established by replicating the virtual server instance in two different data centers, as performed by the VIM that is running at both data centers. The active instance receives the network traffic and is vertically scaling in response, while the standby instance has no workload and runs at the minimum configuration.*

8.7 Hypervisor

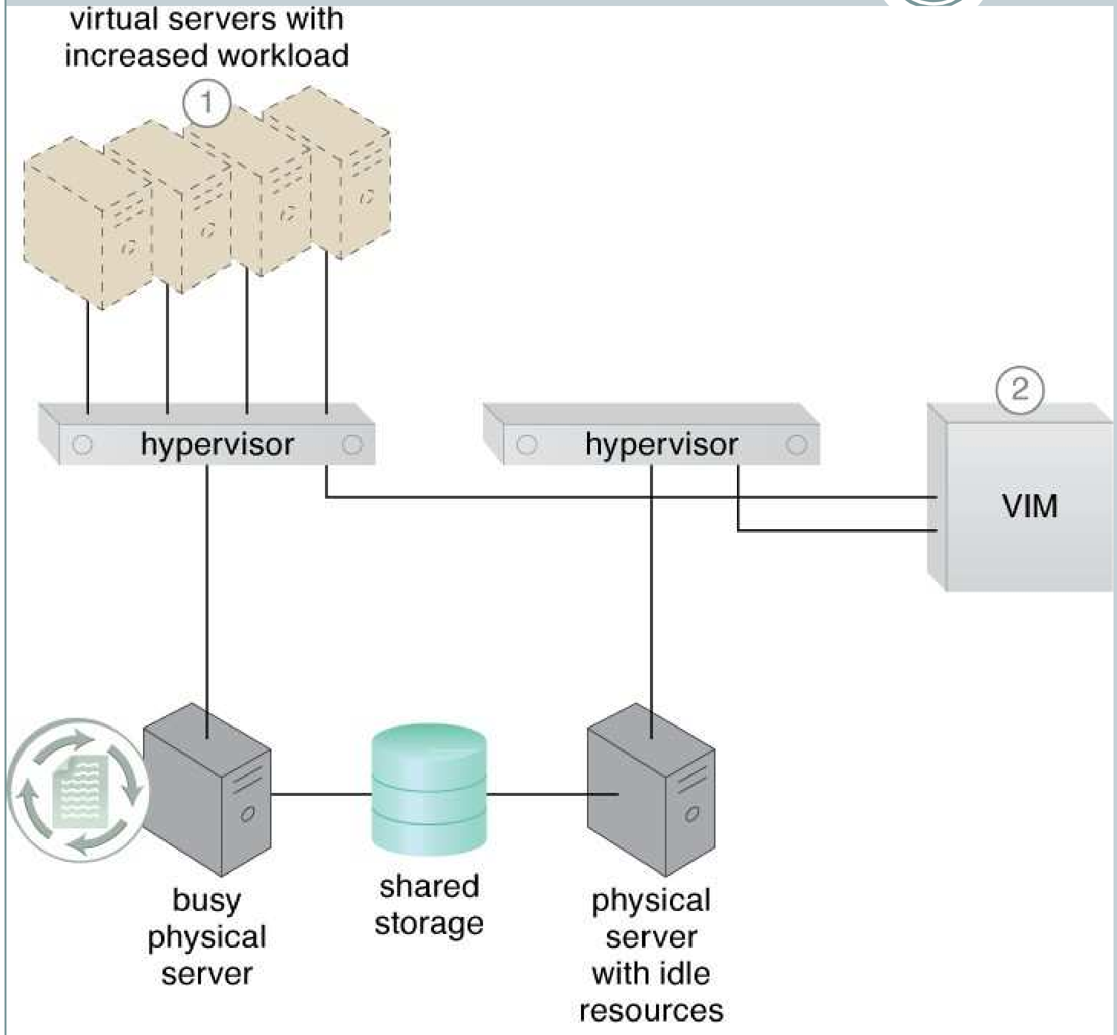
- The hypervisor mechanism is a fundamental part of virtualization infrastructure that is primarily used to generate virtual server instances of a physical server.
- A hypervisor is limited to one physical server, while VIM provides administration features for multiple hypervisors.

Figure 8.27



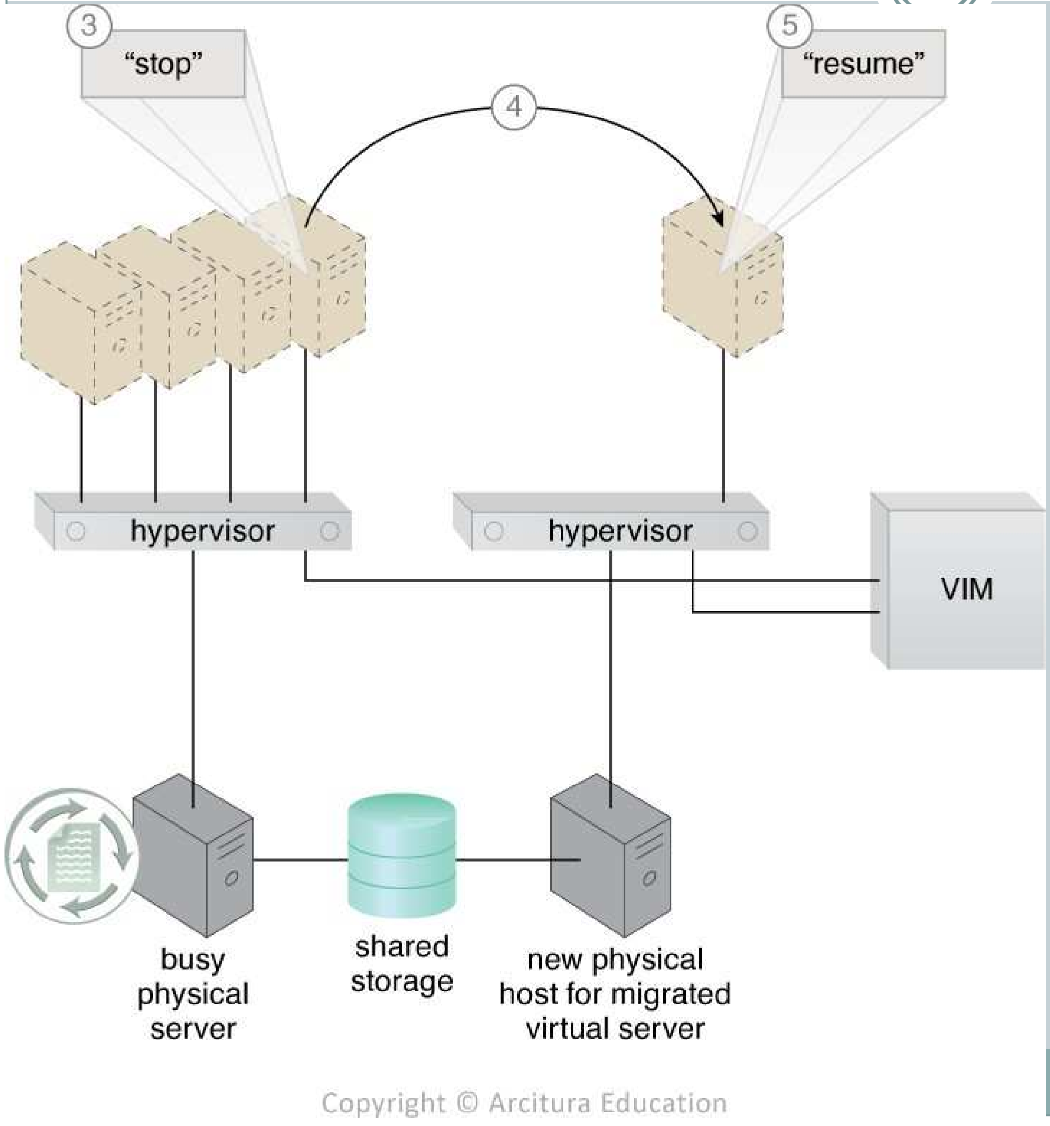
- *Figure 8.27 - Virtual servers are created via individual hypervisor on individual physical servers. All three hypervisors are jointly controlled by the same VIM.*

Figure 8.28 (**DTGOV's Example**)


virtual servers with increased workload
1
hypervisor
hypervisor
2
VIM
busy physical server
shared storage
physical server with idle resources

- *A virtual server capable of auto-scaling experiences an increase in its workload (1).*
- *The VIM decides that the virtual server cannot scale up because its underlying physical server host is being used by other virtual servers (2).*

Figure 8.29


3
"stop"
4
5
"resume"
hypervisor
hypervisor
VIM
busy physical server
shared storage
new physical host for migrated virtual server
Copyright © Arcitura Education

- *The VIM commands the hypervisor on the busy physical server to suspend execution of the virtual server (3).*
- *The VIM then commands the instantiation of the virtual server on the idle physical server. State information (such as dirty memory pages and processor registers) is synchronized (4).*
- *The VIM commands the hypervisor at the new physical server to resume the virtual server processing*

8.8 Resource Cluster (1/3)

- The resource cluster mechanism is used to group multiple geographically diverse IT resource instances so that they can be operated as a single IT resource.

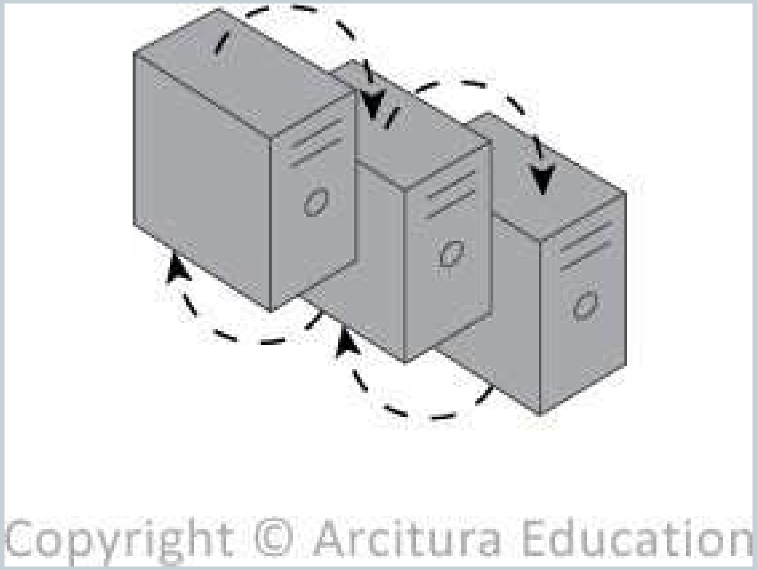


Figure 8.30 - The curved dashed lines are used to indicate that IT resources are clustered.

8.8 Resource Cluster (2/3)

- Resource clustering architectures rely on high-speed dedicated network connections, or cluster nodes, between IT resources to communicate about workload distribution, task scheduling, data sharing, and system synchronization.
- Common resource cluster types include:
  - Server Cluster – increase performance or availability
  - Database Cluster – improve data availability, requires synchronization feature for data consistency.
  - Large Dataset Cluster – data partitioning and distribution

Figure 8.31

- *Figure 8.31 - Load balancing and resource replication are implemented through a cluster-enabled hypervisor. A dedicated storage area network is used to connect the clustered storage and the clustered servers, which are able to share common cloud storage devices. This simplifies the storage replication process, which is independently carried out at the storage cluster.*

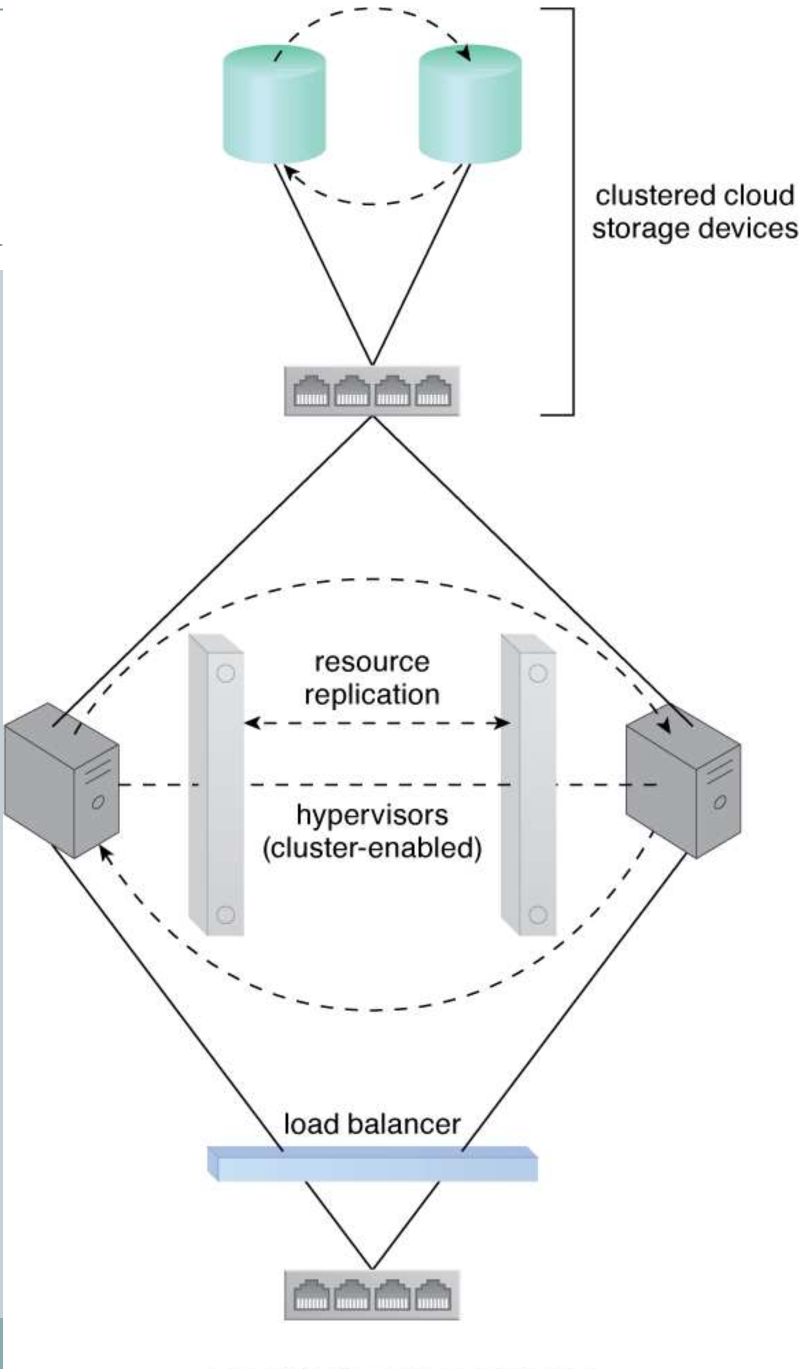

clustered cloud storage devices
resource replication
hypervisors (cluster-enabled)
load balancer

Figure 8.32

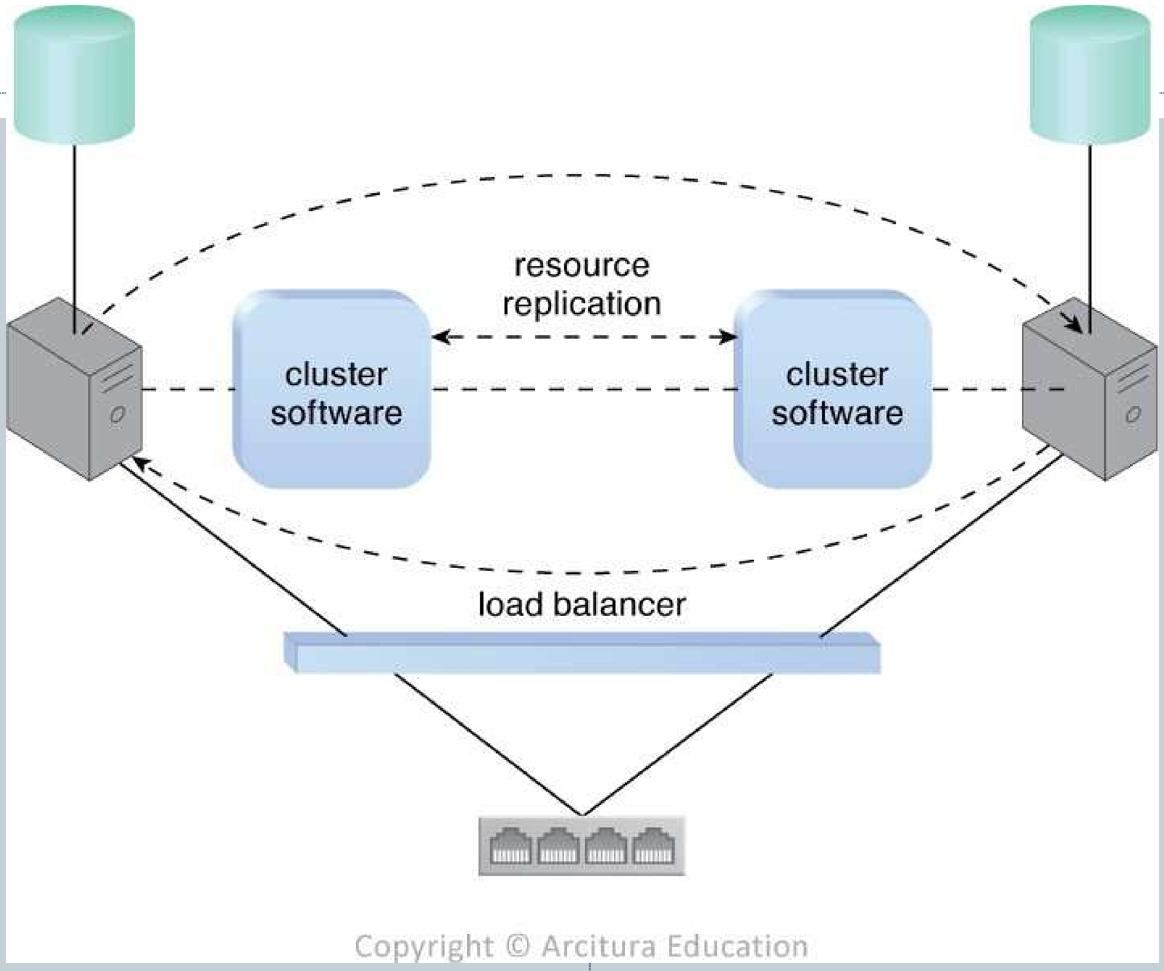
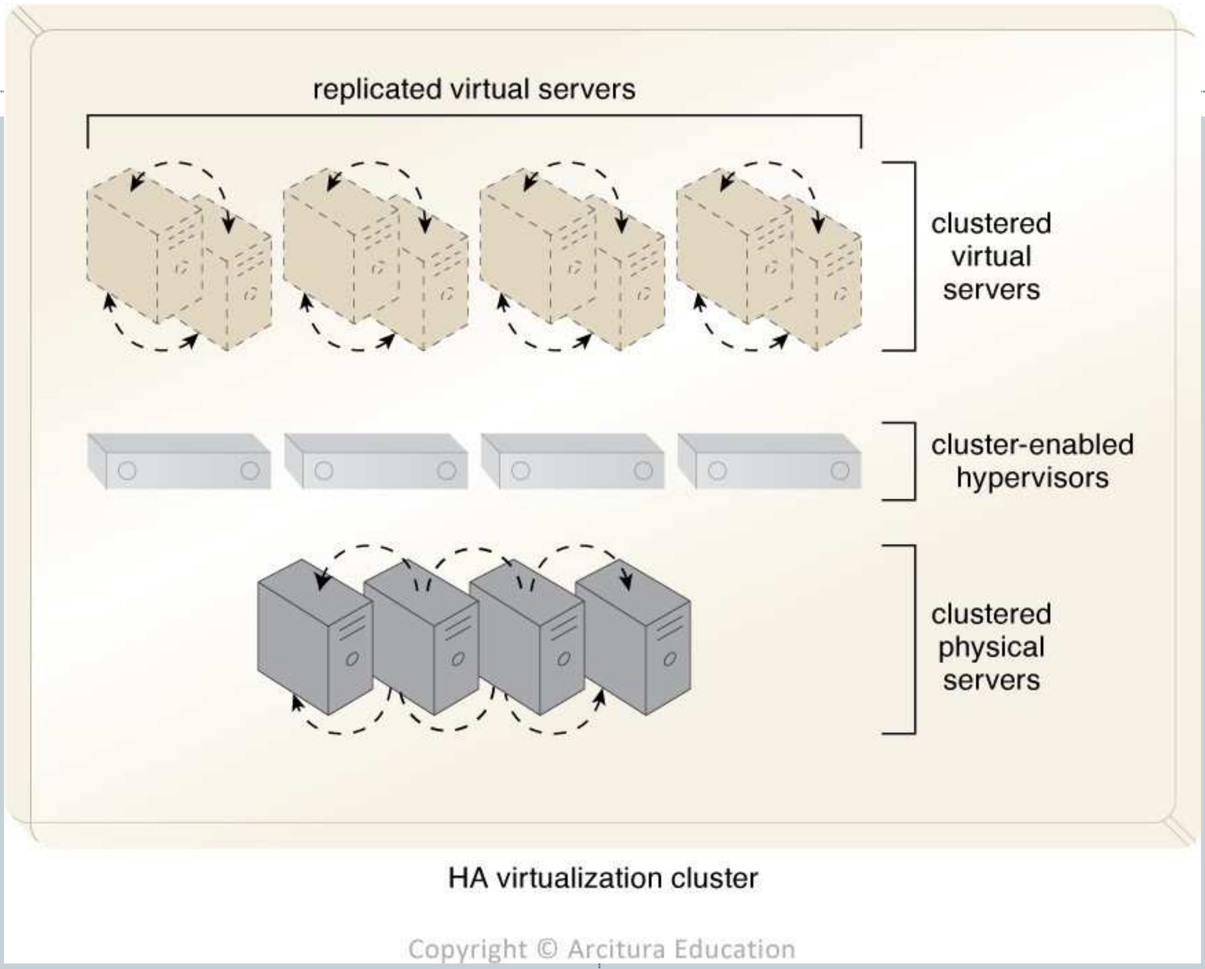


 Figure 8.32 - A loosely coupled server cluster that incorporates a load balancer. There is no shared storage. Resource replication is used to replicate cloud storage devices through the network by the cluster software.

8.8 Resource Cluster (3/3)

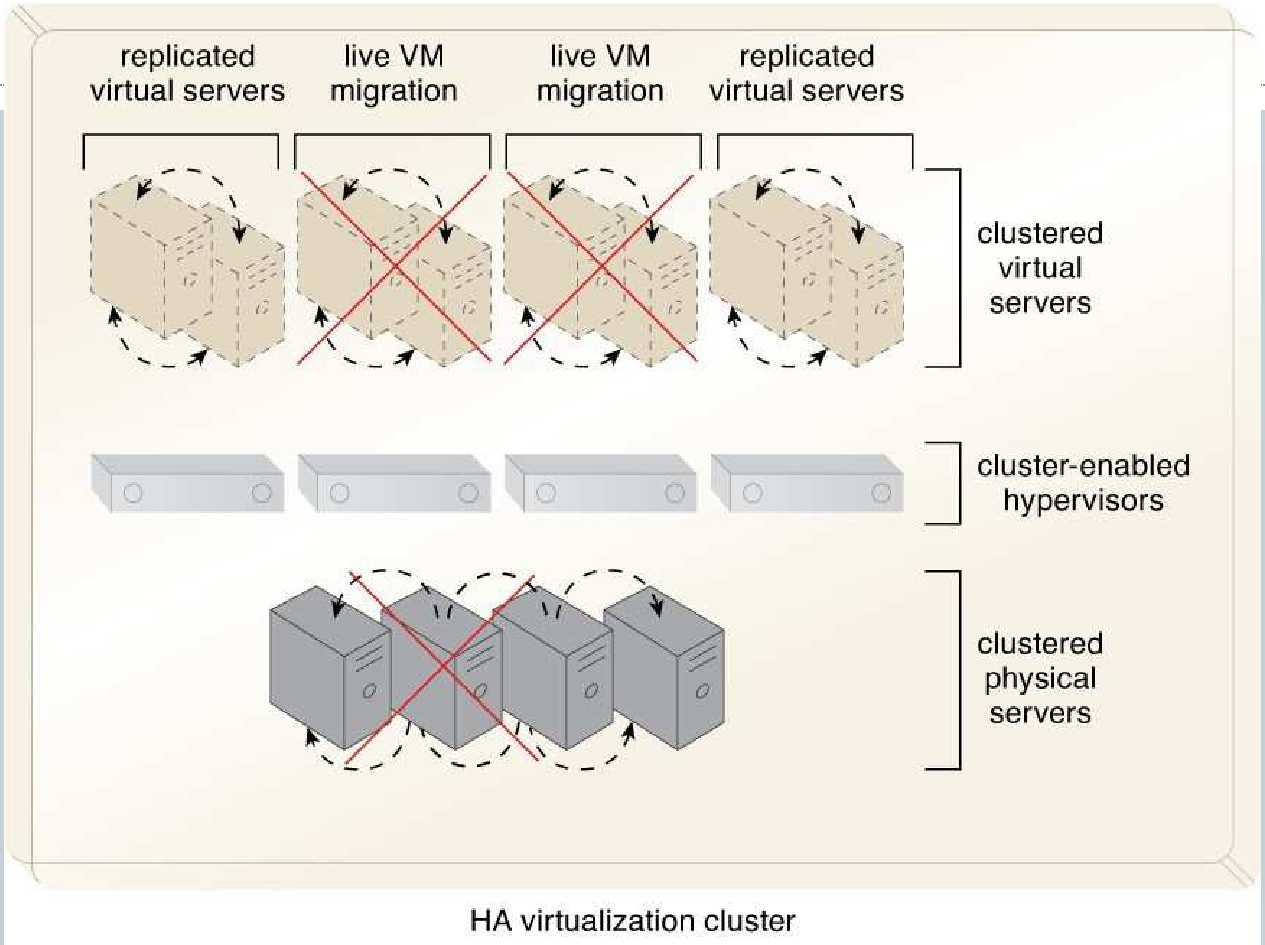
- Resource clusters can also be regarded as one of
  - Load Balanced Cluster – distribute workloads among cluster nodes, while preserving centralized cluster management platform.
  - HA Cluster – maintain system availability in the event of multiple node failures, requiring redundant implementation. Failover system mechanism is also necessary.

Figure 8.33 (**DTGOV's Example**)


replicated virtual servers
clustered virtual servers
cluster-enabled hypervisors
clustered physical servers
HA virtualization cluster

- *Figure 8.33 - An HA virtualization cluster of physical servers is deployed using a cluster-enabled hypervisor, which guarantees that the physical servers are constantly in sync. Every virtual server that is instantiated in the cluster is automatically replicated in at least two physical servers*

Figure 8.34

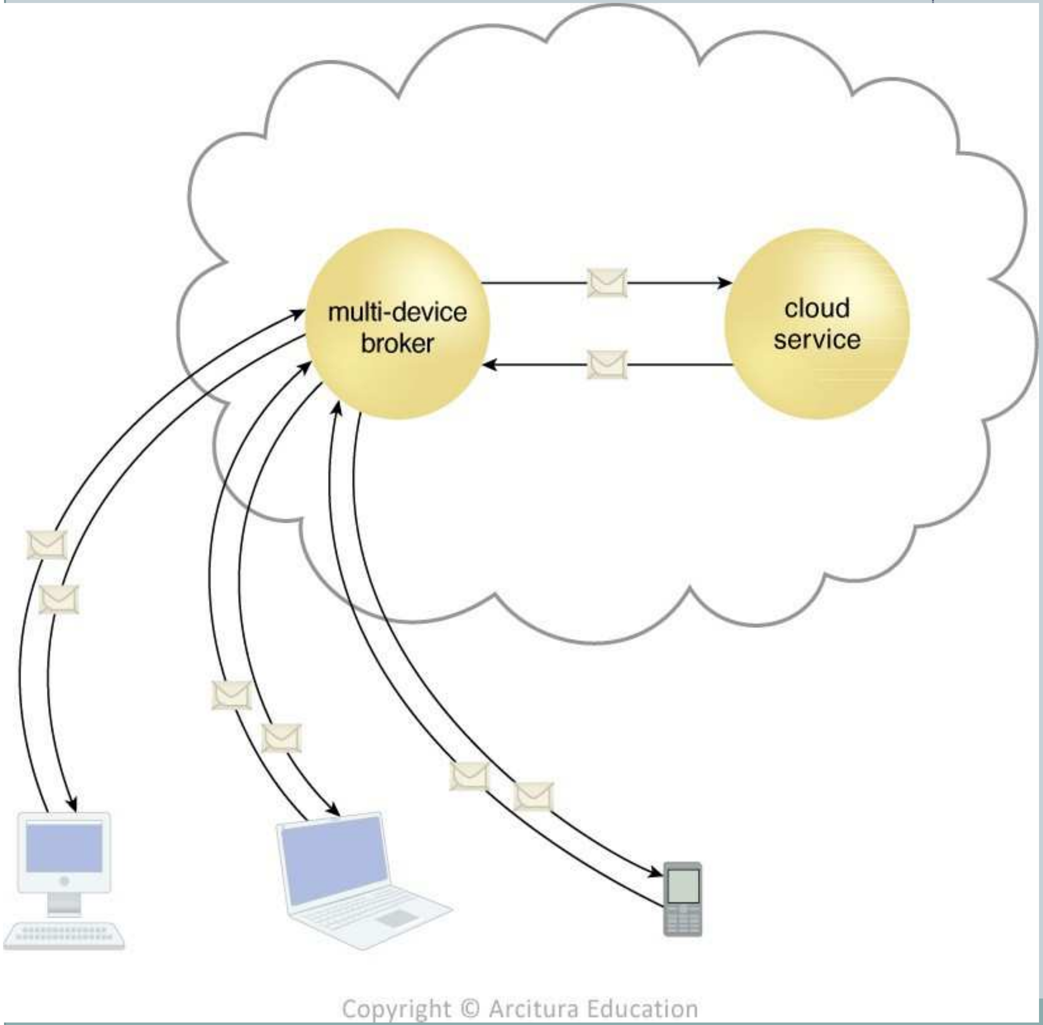

replicated virtual servers
live VM migration
live VM migration
replicated virtual servers
clustered virtual servers
cluster-enabled hypervisors
clustered physical servers
HA virtualization cluster

- *Figure 8.34 - All of the virtual servers that are hosted on a physical server experiencing failure are automatically migrated to other physical servers by the hypervisor.*

8.9 Multiple-Device Broker

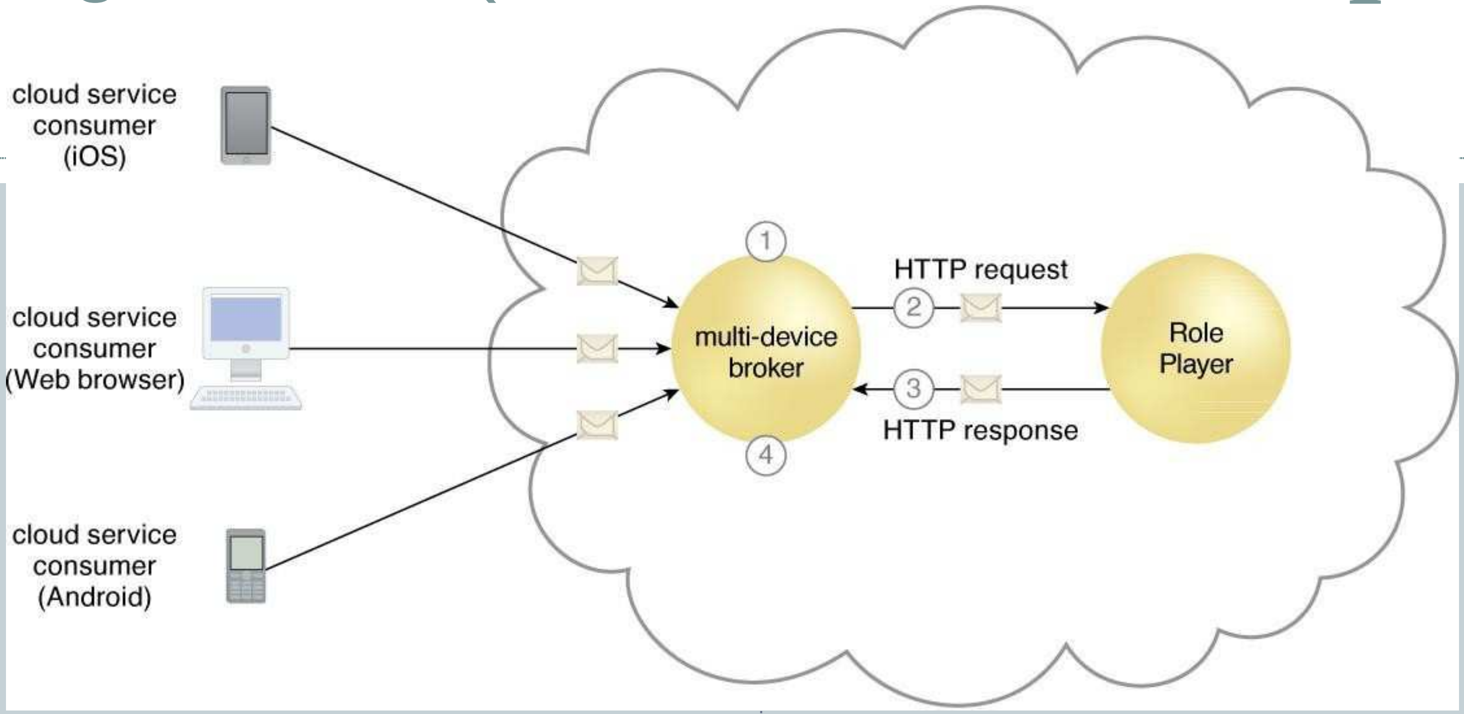
- To overcome incompatibilities between a cloud service and a disparate cloud service consumers (multitenancy), mapping logic is required.
- The multiple-device broker mechanism is used to facilitate runtime data transformation so as to make a cloud service accessible to a wider range of cloud service consumer programs and devices.
- Multi-device brokers commonly exist as gateways such as:
  - XML gateway
  - Cloud storage gateway
  - Mobile device gateway

Figure 8.35



- *Figure 8.35 - A multi-device broker contains the mapping logic necessary to transform data exchanges between a cloud service and different types of cloud service consumer devices.*
- *This mechanism can also be implemented as a service agent that intercepts messages at runtime to perform necessary transformations.*

Figure 8.36 (**Innovartus's Example**)


cloud service consumer (iOS)
cloud service consumer (Web browser)
cloud service consumer (Android)
1
multi-device broker
4
HTTP request
2
3
HTTP response
Role Player

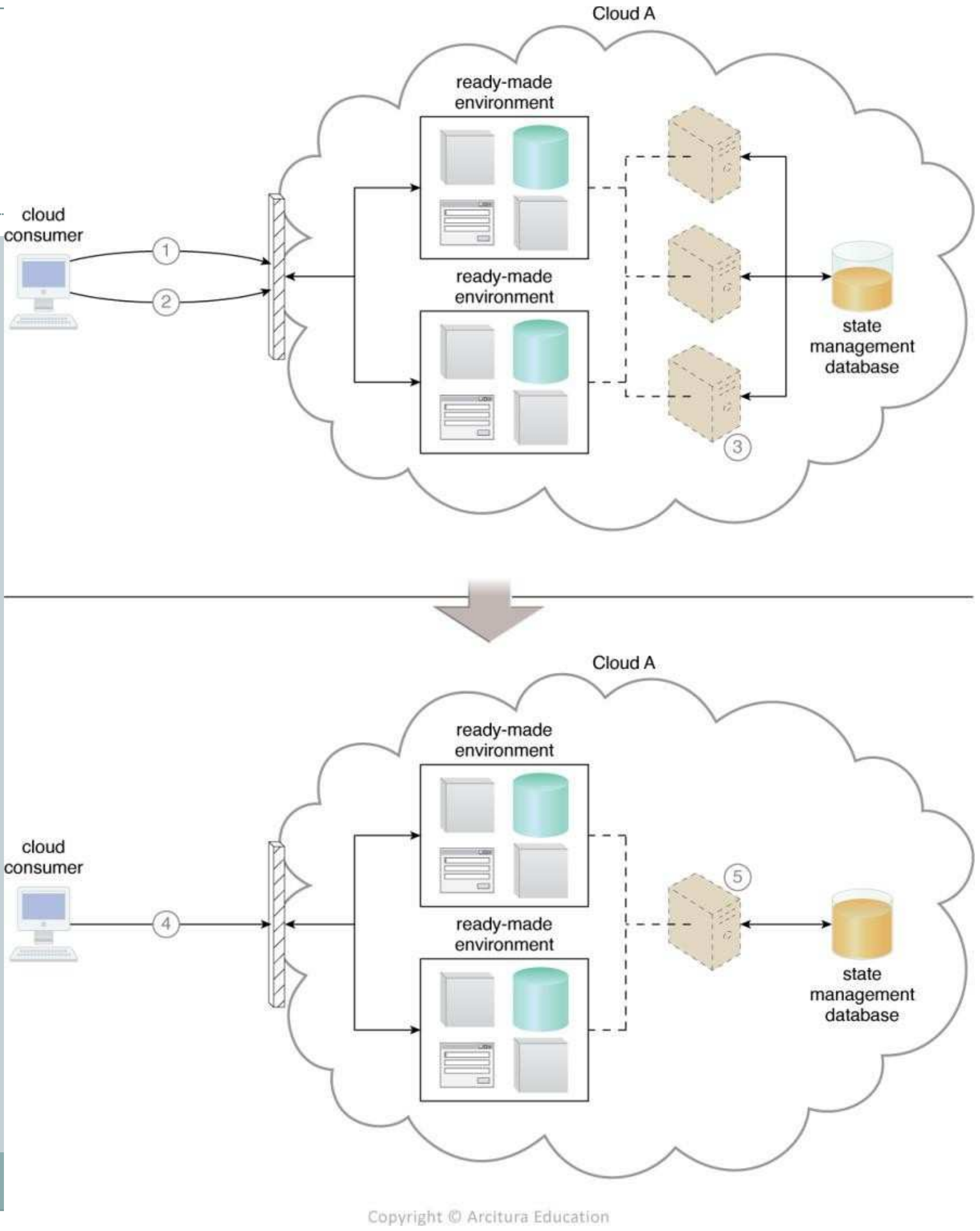
- *The multi-device broker intercepts incoming messages and detects the platform (Web browser, iOS, Android) of the source device (1).*
- *The multi-device broker transforms the message into the standard format required by the Innovartus cloud service (2).*
- *The cloud service processes the request and responds using the same standard format (3).*
- *The multi-device broker transforms the response message into the format required by the source device and delivers the message (4).*

8.10 State Management Database

- A state management database is a storage device that is used to temporarily persist state data for software programs.
- As an alternative to caching state data in memory, software programs can off-load state data to the database to reduce the amount of runtime memory.
- State management database allows software programs and the surrounding infrastructure to be more scalable.

Figure 8.39

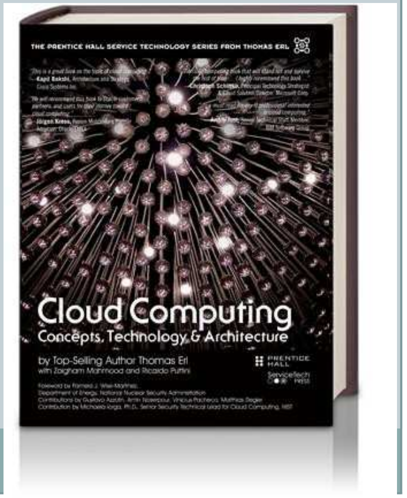
- *The cloud server accesses the ready-made environment and requires three virtual servers to perform all activities (1).*
- *The cloud consumer pauses activity. (2).*
- *Reducing the number of virtual servers for scale in. State data is saved in the state management database and one virtual server remains active to allow for future logins by the cloud consumer (3).*



Cloud Computing

Concept, Technology & Architecture

Chapter 09
Cloud Management Mechanics


Cloud Computing
Concepts, Technology & Architecture
by Top-Selling Author Thomas Erl

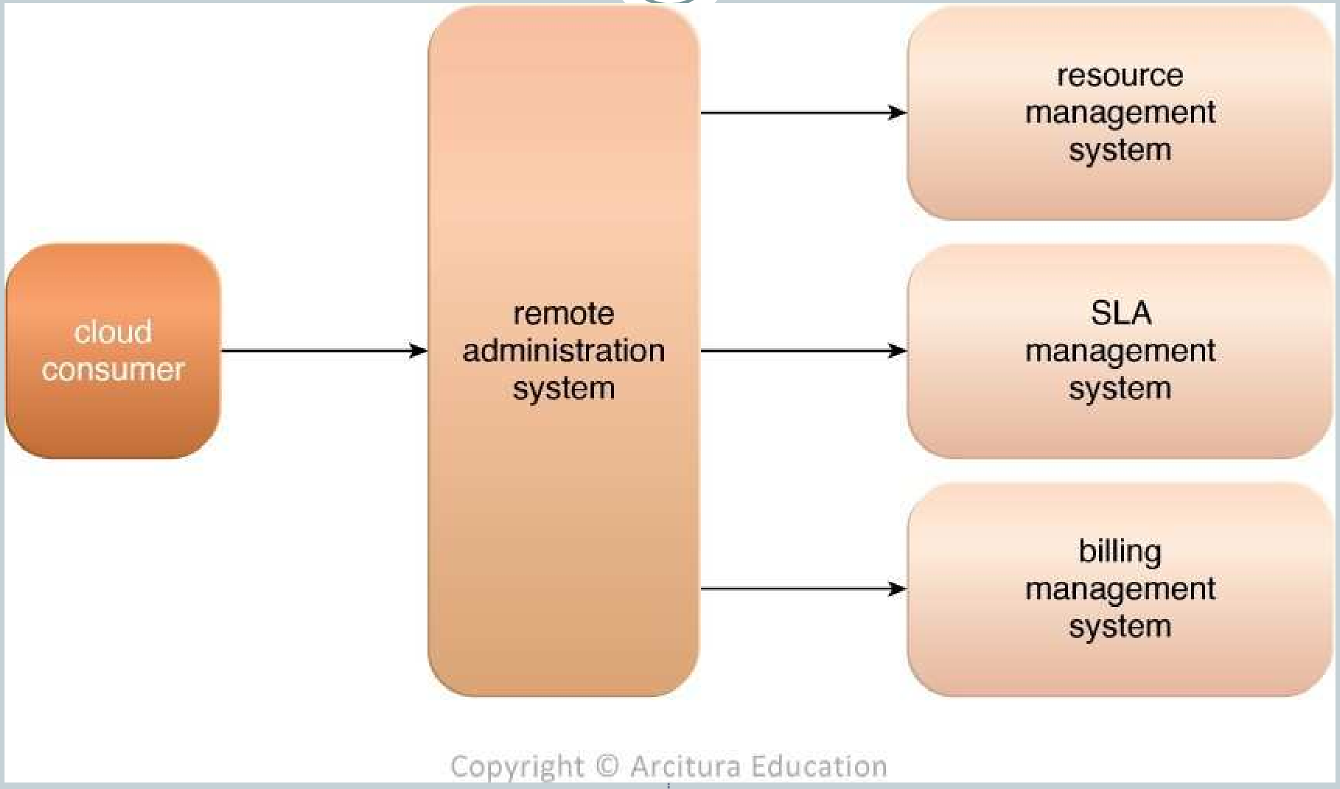
Cloud Management Mechanisms

- Cloud-based IT resources need to be set up, configured, maintained, and monitored.
- Cloud-based IT resources mechanisms for hands-on administration and management include the following:
  - 9.1 Remote Administration System
  - 9.2 Resource Management System
  - 9.3 SLA Management System
  - 9.4 Billing Management System

9.1 Remote Administration System (1/3)

- The remote administration system mechanism provides tools and user-interfaces for external cloud resource administrators to configure and administer cloud-based IT resources.
- A remote administration can establish a portal for access to administration and management features including the resource management, SLA management, and billing system.

Figure 9.2


cloud consumer
remote administration system
resource management system
SLA management system
billing management system
Copyright © Arcitura Education

- *Figure 9.2 - The remote administration system abstracts underlying management systems to expose and centralize administration controls to external cloud resource administrators.*

9.1 **Remote Administration System** (2/3)

- The tools and APIs provided by a remote administration system are generally used by the cloud provider to develop and customize online portals that provide cloud consumers with a variety of administrative controls.
- Two primary types of portals that are created with the remote administration system:
  - Usage and Administration Portal – A general purpose portal
  - Self-Service Portal – This is essentially a self service portal that allows cloud consumers too search an up-to-date list of cloud services.

Figure 9.3 (1/2)

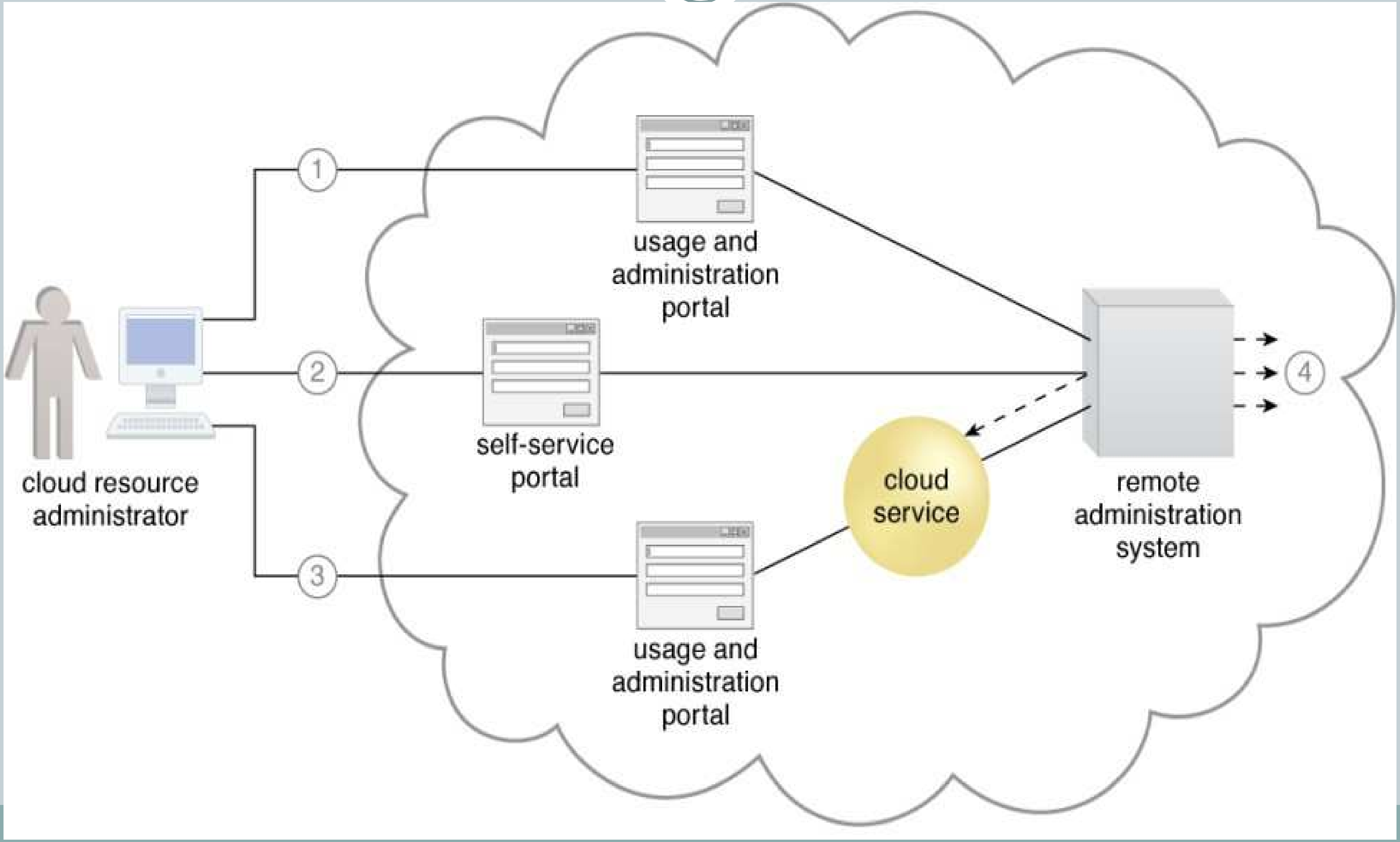

1
usage and administration portal
cloud resource administrator
2
self-service portal
4
remote administration system
cloud service
3
usage and administration portal

Figure 9.3 (2/2)

- A cloud resource administrator uses the usage and administration portal to configure an already leased virtual server to prepare it for hosting (1).
- The cloud resource administrator then uses the self-service portal to select and request the provisioning of a new cloud service (2).
- The cloud resource administrator then accesses the usage and administration portal again to configure the newly provisioned cloud service that is hosted on the virtual server (3).
- Throughout these steps, the remote administration system interacts with the necessary management systems to perform the requested actions.(4)

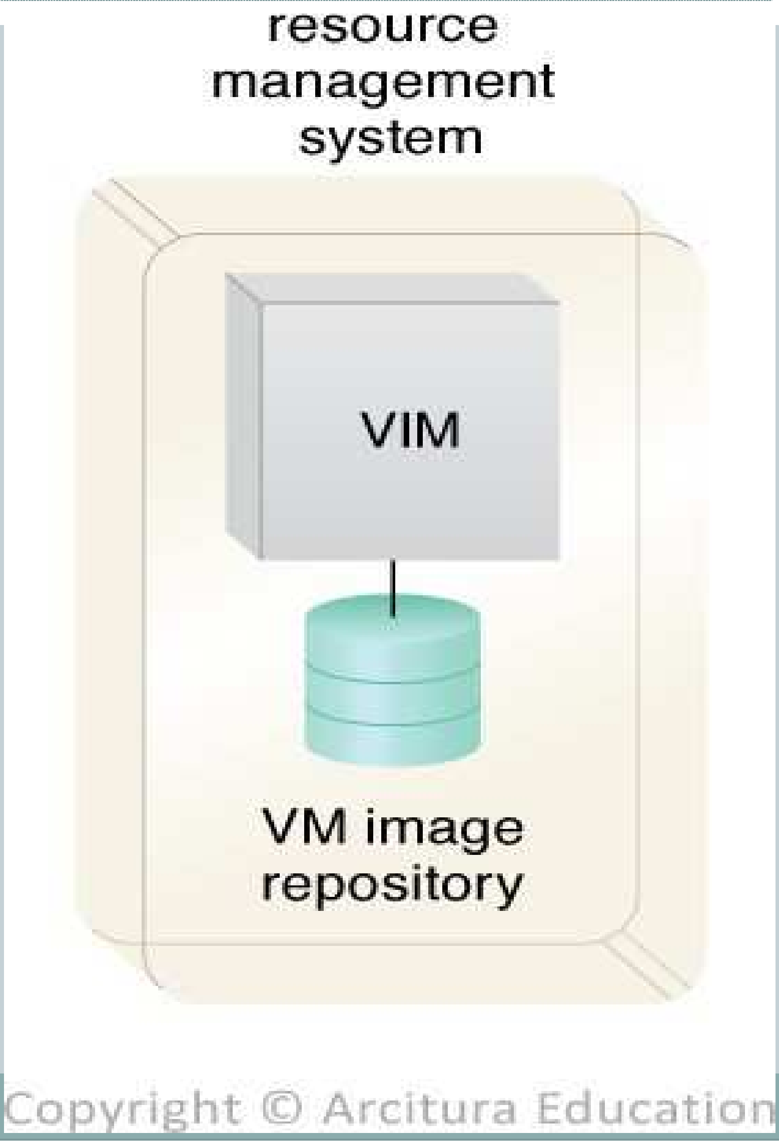
9.1 Remote Administration System (3/3)

- Tasks that can commonly be performed by cloud consumers via a remote administration console include:
  - configuring and setting up cloud services
  - provisioning and releasing IT resources for on-demand cloud services
  - monitoring cloud service status, usage, and performance
  - monitoring QoS and SLA fulfillment
  - managing leasing costs and usage fees
  - managing user accounts, security credential, authorization, and access control
  - tracking internal and external access to leased services
  - capacity planning

9.2 Resource Management System (1/2)

- The resource management system mechanism helps coordinate IT resources in response to management actions performed by both cloud consumers and cloud providers.
- Core to this system is the virtual infrastructure manager (VIM) that coordinates the server hardware to create virtual server instances.
- A VIM is a commercial product that can be used to manage a range of virtual IT resources across multiple physical servers.

Figure 9.5



- *Figure 9.5 - A resource management system encompassing a VIM platform and a virtual machine image repository. The VIM may have additional repositories, include one dedicated to storing operational data.*

9.2 Resource Management System (2/2)

- Tasks that are typically automated and implemented through the resource management system include:
  - managing virtual IT resource templates.
  - allocating and releasing virtual IT resources, such as starting, pausing, resuming, and termination.
  - coordinating IT resources in relation to the involvement of other mechanisms, such as resource replication, load balancer, and failover system.
  - enforcing usage and security policies through the lifecycle of cloud services instances.
  - monitoring operational conditions of IT resources.

Figure 9.6

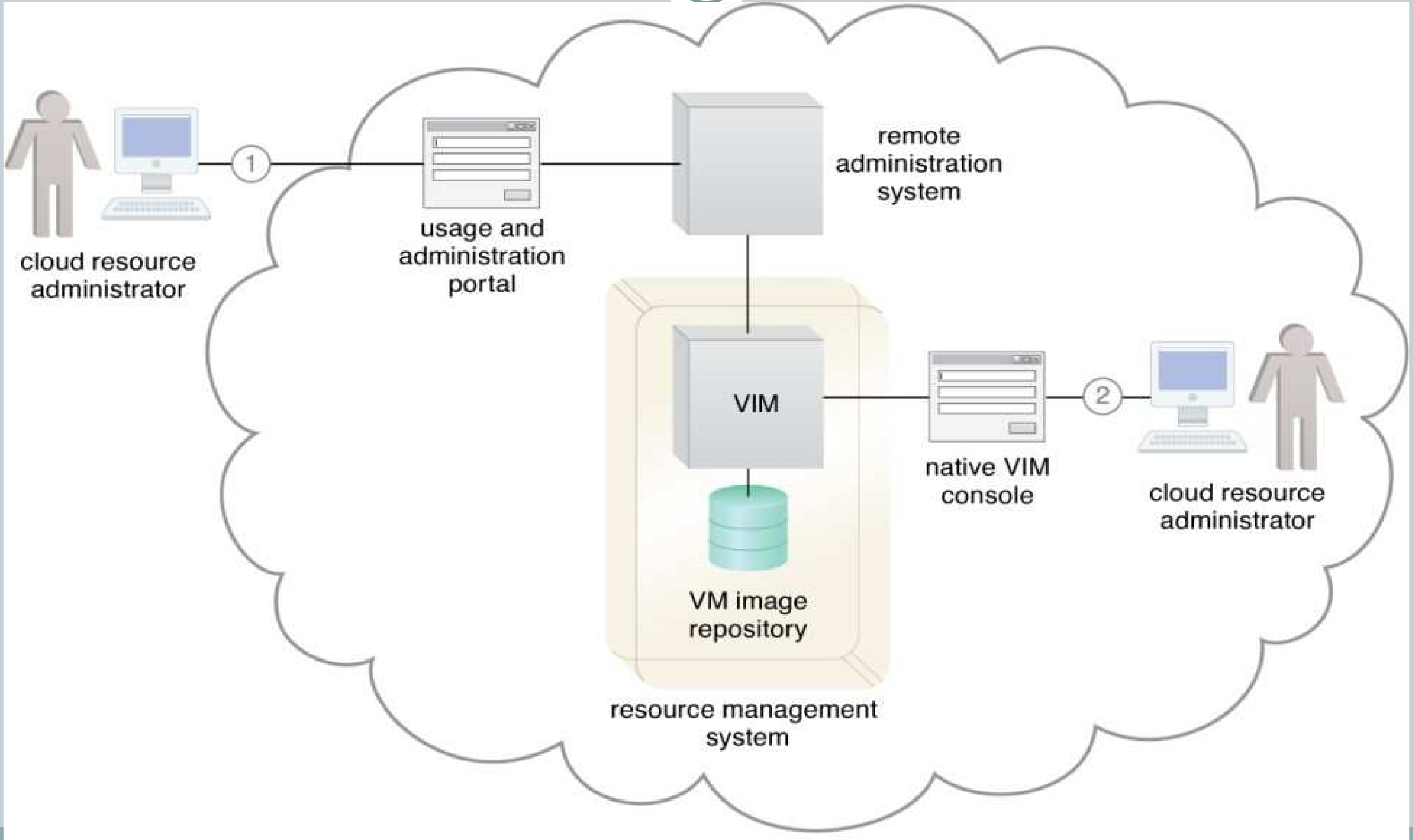

cloud resource administrator
1
usage and administration portal
remote administration system
VIM
VM image repository
resource management system
native VIM console
2
cloud resource administrator

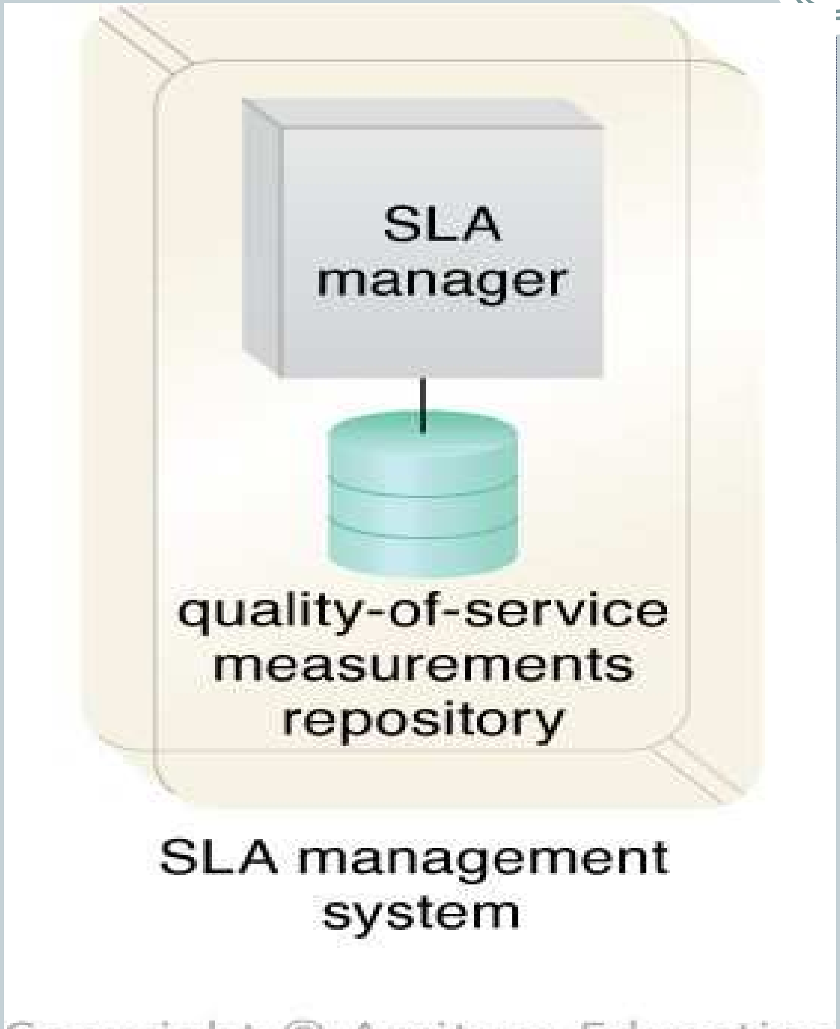
Figure 9.6 External or internal administration

- The cloud consumer's cloud resource administrator accesses a usage and administration portal externally to administer a leased IT resource (1).
- The cloud provider's cloud resource administrator uses the native user-interface provided by the VIM to perform internal resource management tasks (2).

9.3 SLA Management System

- The SLA management system mechanism represents a range of commercially available cloud management products that provide features pertaining to the administration, collection, storage, reporting, and runtime notification of SLA data.
- An SLA management system deployment generally include a repository used to store and retrieve collected SLA data based on pre-defined metrics and reporting parameters.

Figure 9.7



- *Figure 9.7 - An SLA* management system encompassing an SLA manager and QoS measurements repository.

Figure 9.8

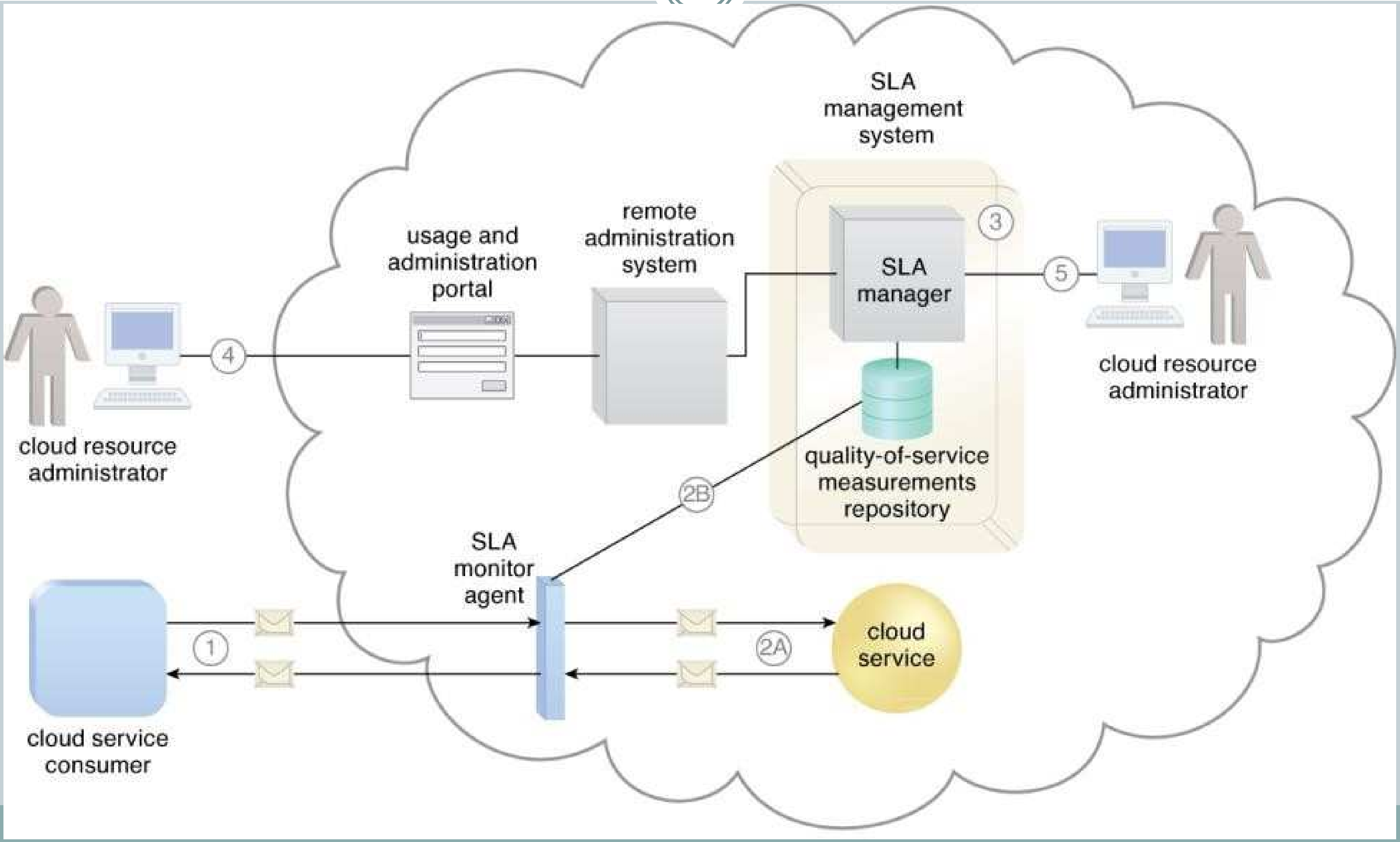

SLA management system
usage and administration portal
remote administration system
SLA manager
3
5
4
cloud resource administrator
cloud resource administrator
quality-of-service measurements repository
2B
SLA monitor agent
1
2A
cloud service
cloud service consumer

Figure 9.8

- A cloud service consumer interacts with a cloud service (1).
- An SLA monitor intercepts the exchanged messages, evaluates the interaction, and collects relevant runtime data in relation to quality-of-service guarantees defined in the cloud service's SLA (2A).
- The data collected is stored in a repository (2B) that is part of the SLA management system (3).
- Queries can be issued and reports can be generated for a cloud resource administrator with an external cloud consumer via a usage and administrator portal (4) or for an internal cloud resource administrator via the SLA management system's native user interface (5).

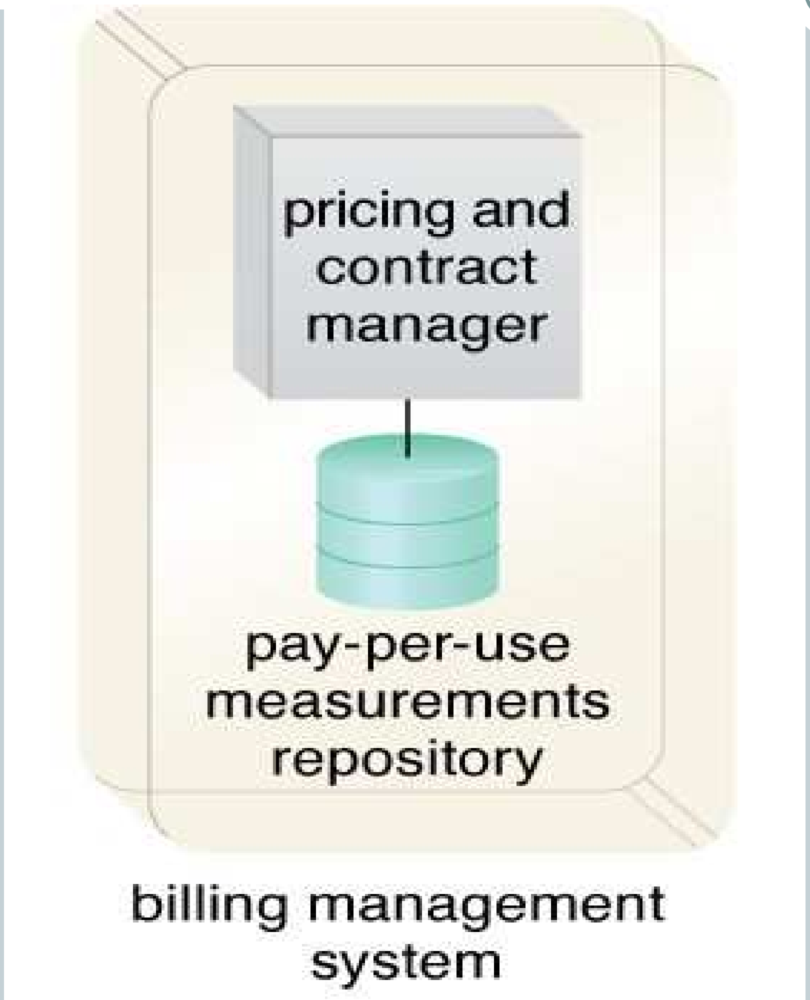
9.4 Billing Management System (1/2)

- The billing management system mechanism is dedicated to the collection and processing of usage as it pertains to cloud provider accounting and cloud consumer billing.
- It relies on pay-per-use monitors to gather runtime usage data for billing, reporting, and invoicing purpose.

9.4 Billing Management System (2/2)

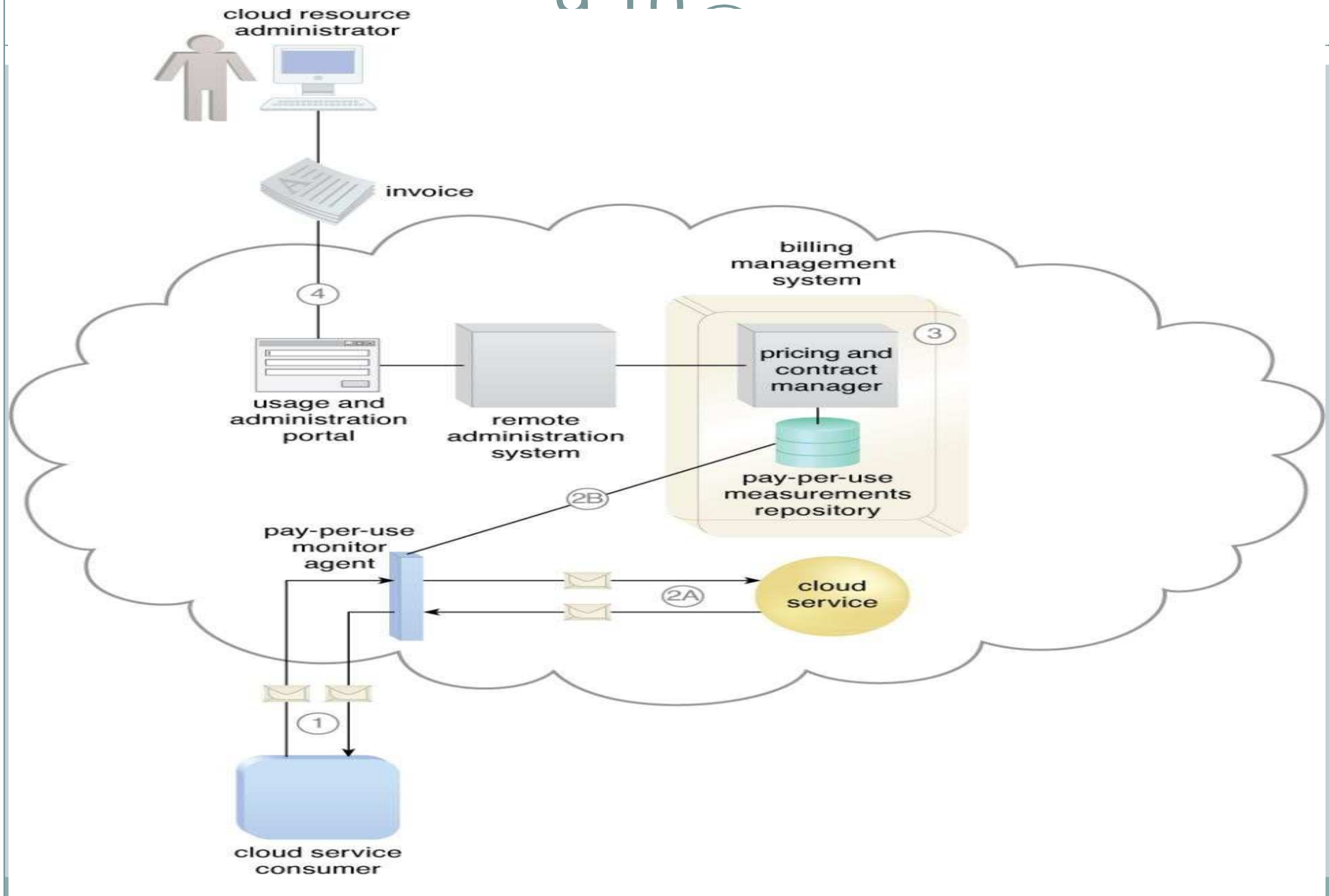
- The billing management system allows for definition of different pricing policies, as well as custom pricing model on per consumer or per IT resource basis.
- Pricing model can be pay-per-use, flat-rate, pay-per-allocation modes, or a combination.
- Billing arrangements can be based on pre-usage or post-usage payments.

Figure 9.9


pricing and contract manager
pay-per-use measurements repository
billing management system

- *Figure 9.9 - A billing* management system comprised of a pricing and contract manager and a pay-per-use measurements repository.

Figure 9.10



cloud resource administrator
invoice
4
billing management system
3
pricing and contract manager
usage and administration portal
remote administration system
pay-per-use measurements repository
2B
pay-per-use monitor agent
2A
cloud service
1
cloud service consumer

Figure 9.10

- Cloud service consumers exchange messages with a cloud service (1).
- A pay-per-use monitor keeps track of the usage and collects data relevant to billing (2A), which is forwarded to a repository that is part of the billing management system (2B).
- The system periodically calculates the consolidated cloud service usage fees and generates an invoice for the cloud consumer (3).
- The invoice may be provided to the cloud consumer through the remote administration system console (4).