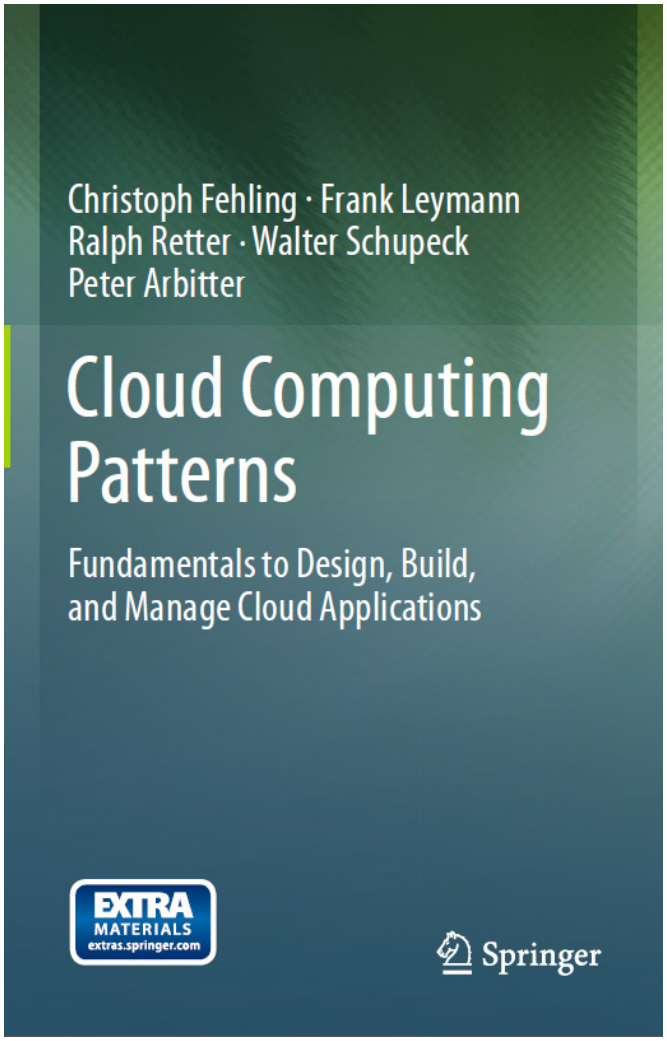
Cloud Computing Patterns

http://www.cloudcomputingpatterns.org

Cloud Computing Patterns



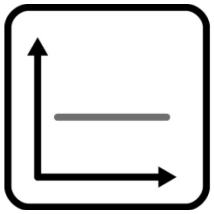
Patterns are a widely used concept in computer science to describe good solutions to reoccurring problems in an abstract form. Such conceptual solutions can then be applied in concrete use cases regardless of used technologies, such as software, middleware, or programming languages. We employ patterns to describe cloud service models and cloud deployment types in an abstract form to categorize the offerings of cloud providers. Furthermore, we give reoccurring cloud application architectural patterns on how to design, build, and manage applications that use these cloud offerings. The abstraction of these patterns make them applicable to challenges faced by developers regardless of the actual technologies and cloud services that they are using.

Cloud Computing Fundamentals
Application Workloads

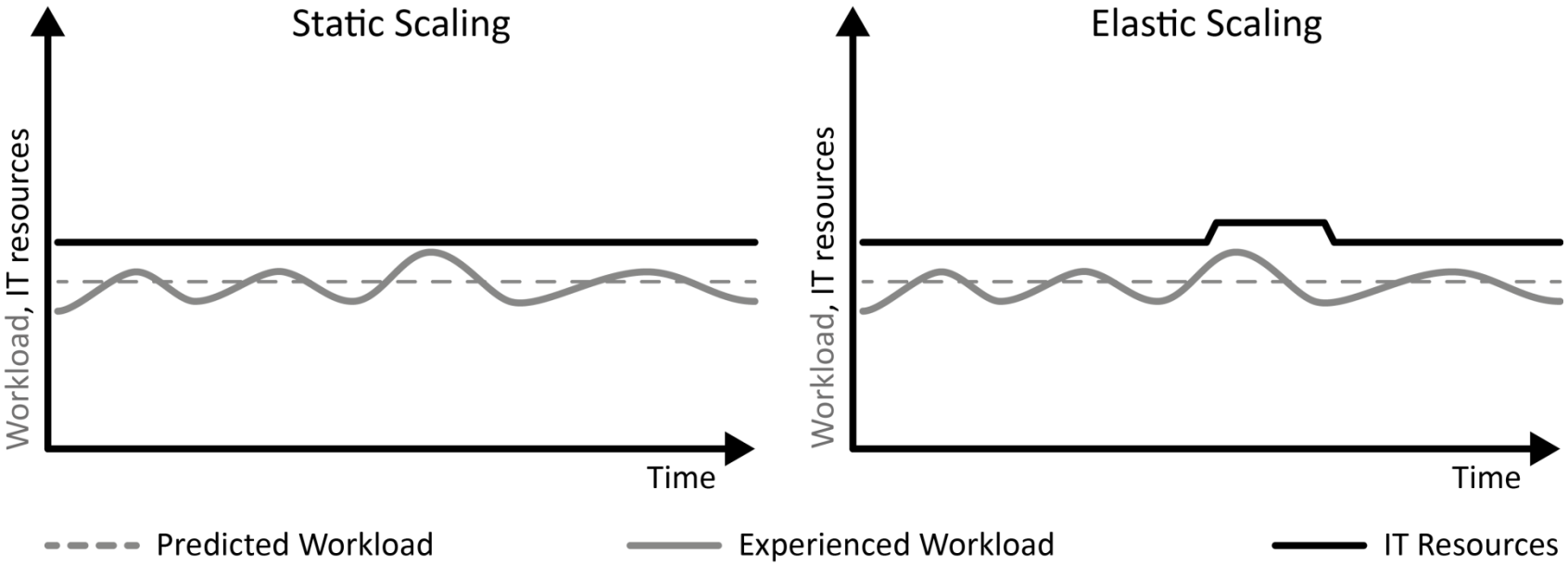
http://www.cloudcomputingpatterns.org

Static Workload

IT resources with an equal utilization over time experience static workload.

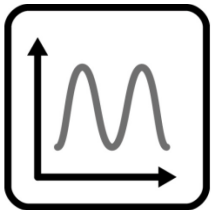


How can an equal utilization be characterized and how can applications experiencing this workload benefit from cloud computing?

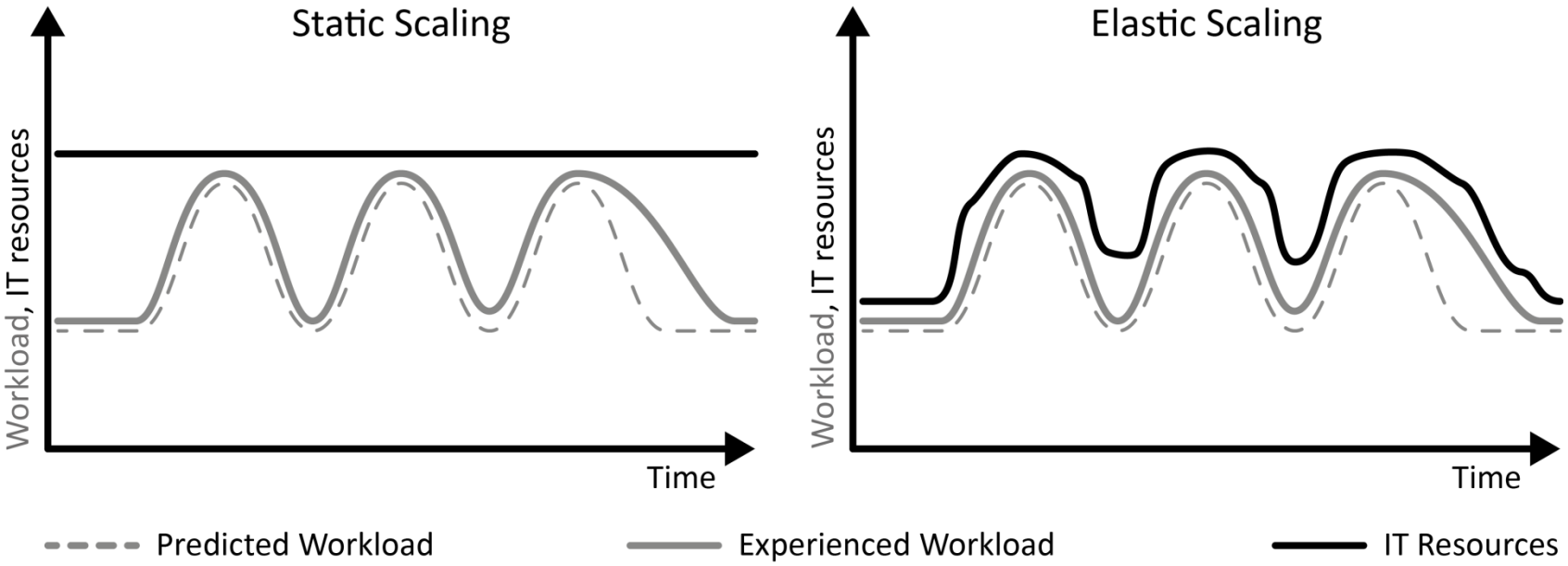


Periodic Workload

IT resources with a peaking utilization at reoccurring time intervals experience periodic workload.

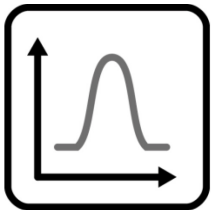


How can an equal utilization be characterized and how can applications experiencing this workload benefit from cloud computing?

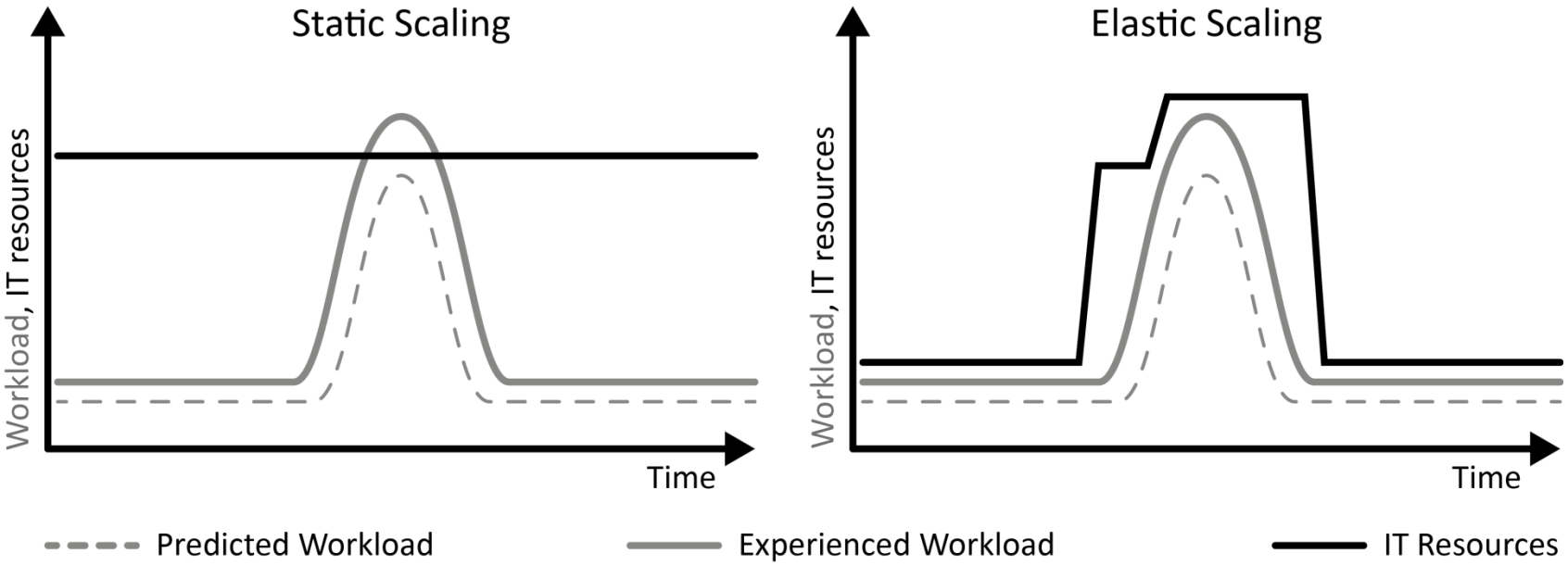


Once-in-a-lifetime Workload

IT resources with an equal utilization over time disturbed by a strong peak occurring only once experience once-in-a-lifetime workload.

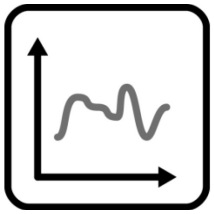


How can equal utilization with a one-time peak be characterized and how can applications experiencing this workload benefit from cloud computing?

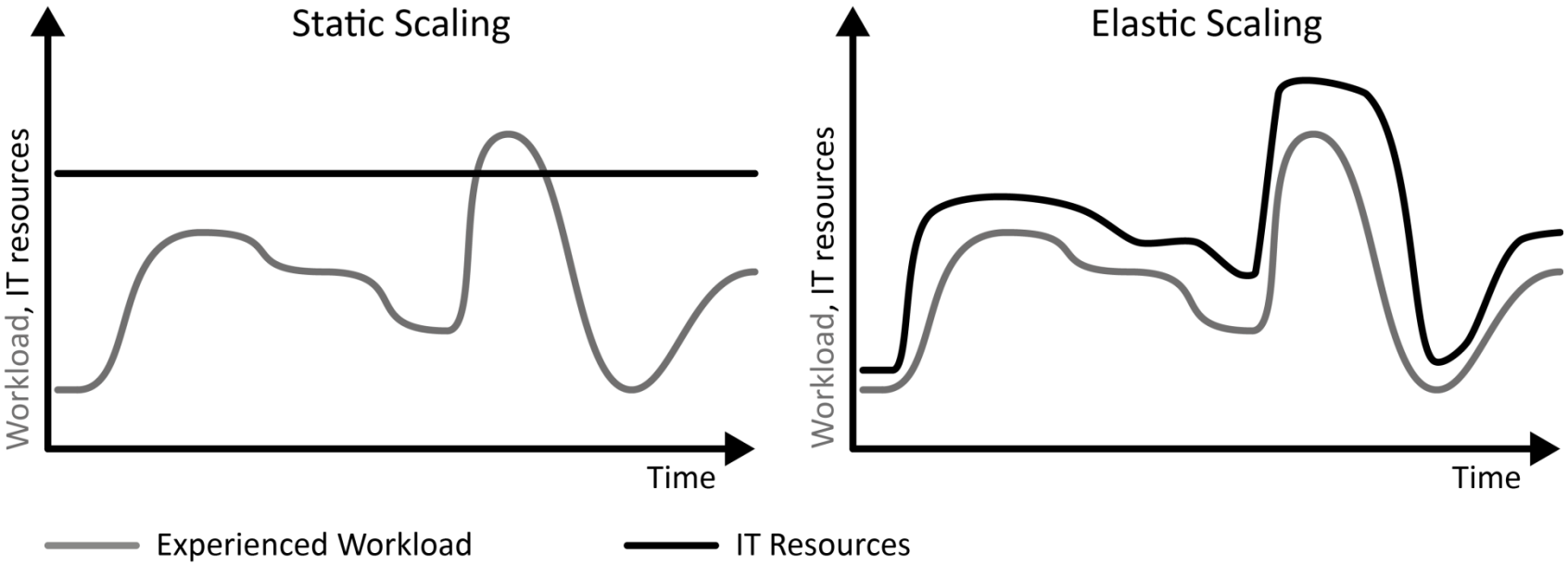


Unpredictable Workload

IT resources with a random and unforeseeable utilization over time experience unpredictable workload.

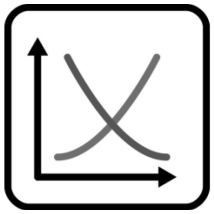


How can random and unforeseeable utilization be characterized and how can applications experiencing this workload benefit from cloud computing?

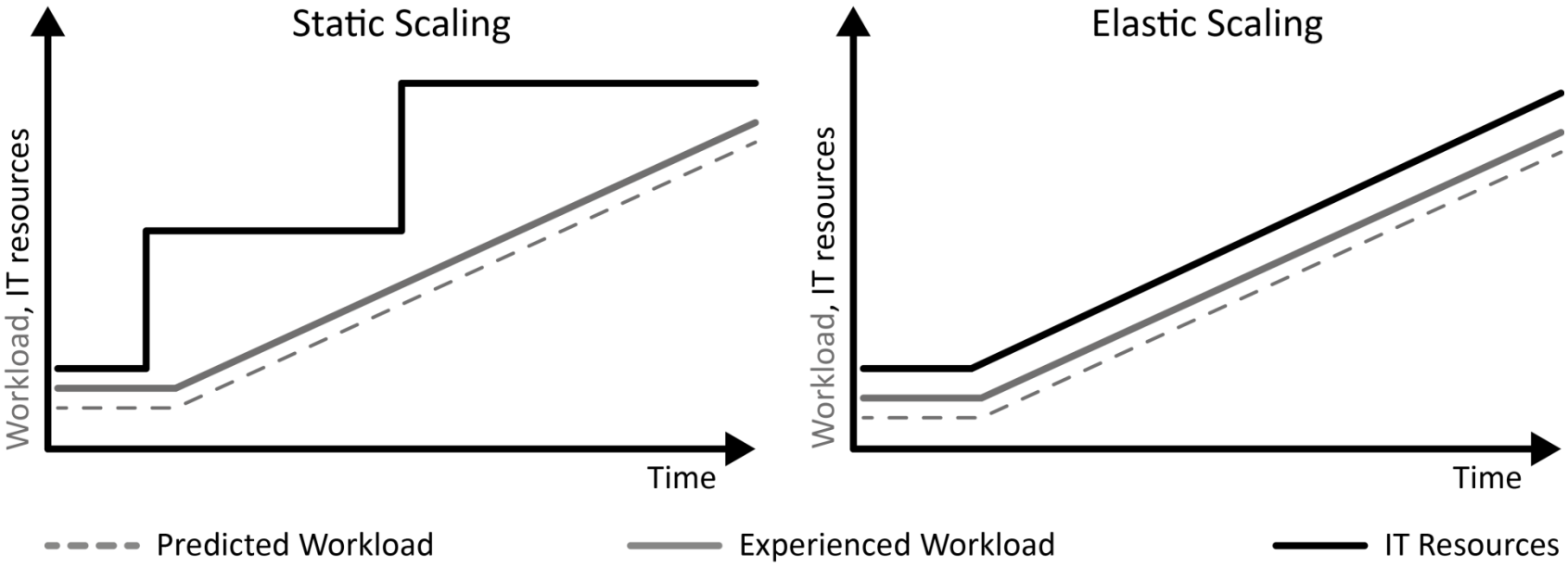


Continuously Changing Workload

IT resources with a utilization that grows or shrinks constantly over time experience continuously changing workload.



How can a continuous growth or decline in utilization be characterized and how can applications experiencing this workload benefit from cloud computing?



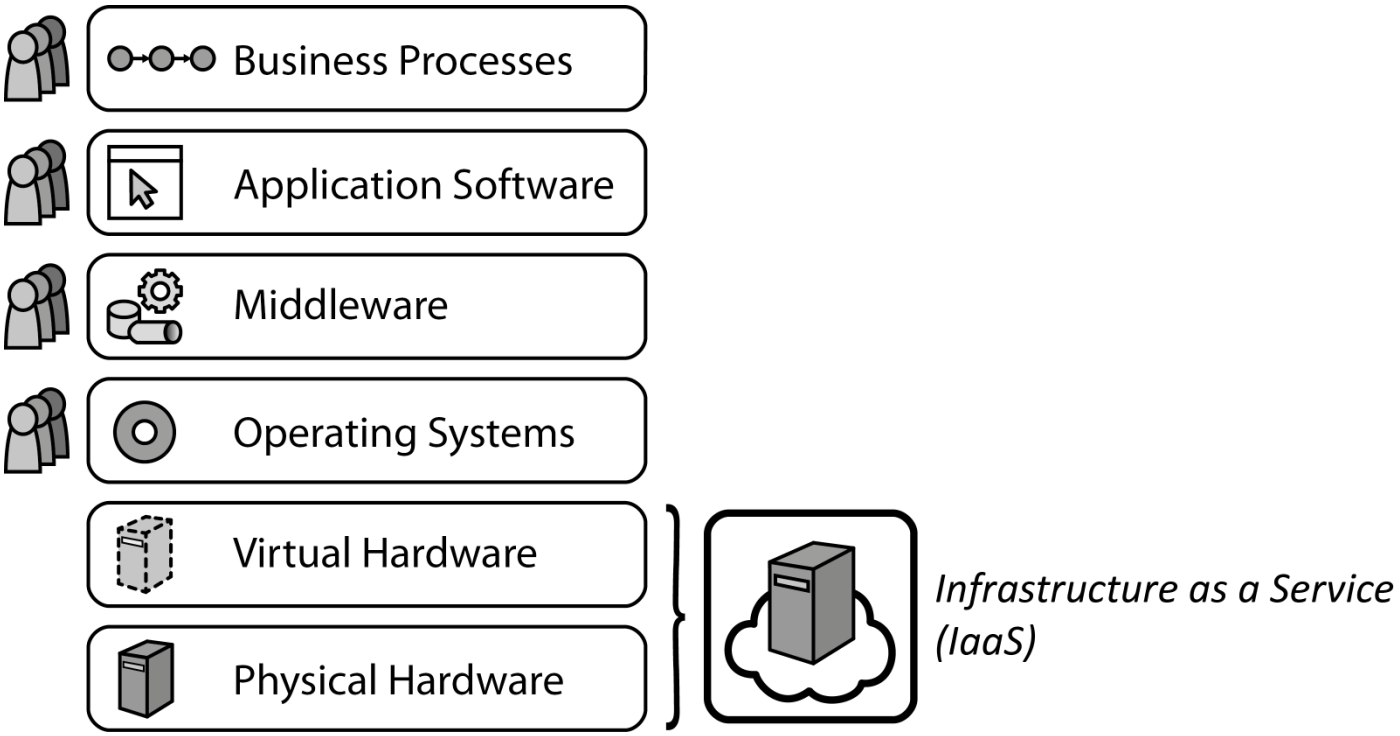
Cloud Computing Fundamentals
Cloud Service Models

http://www.cloudcomputingpatterns.org

Infrastructure as a Service (IaaS)

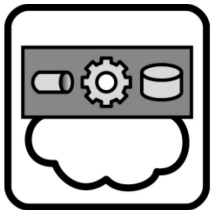
Providers share physical and virtual hardware IT resources between customers to enable self-service, rapid elasticity, and pay-per-use pricing.

How can different customers share a physical hosting environment so that it can be used on-demand with a pay-per-use pricing model?

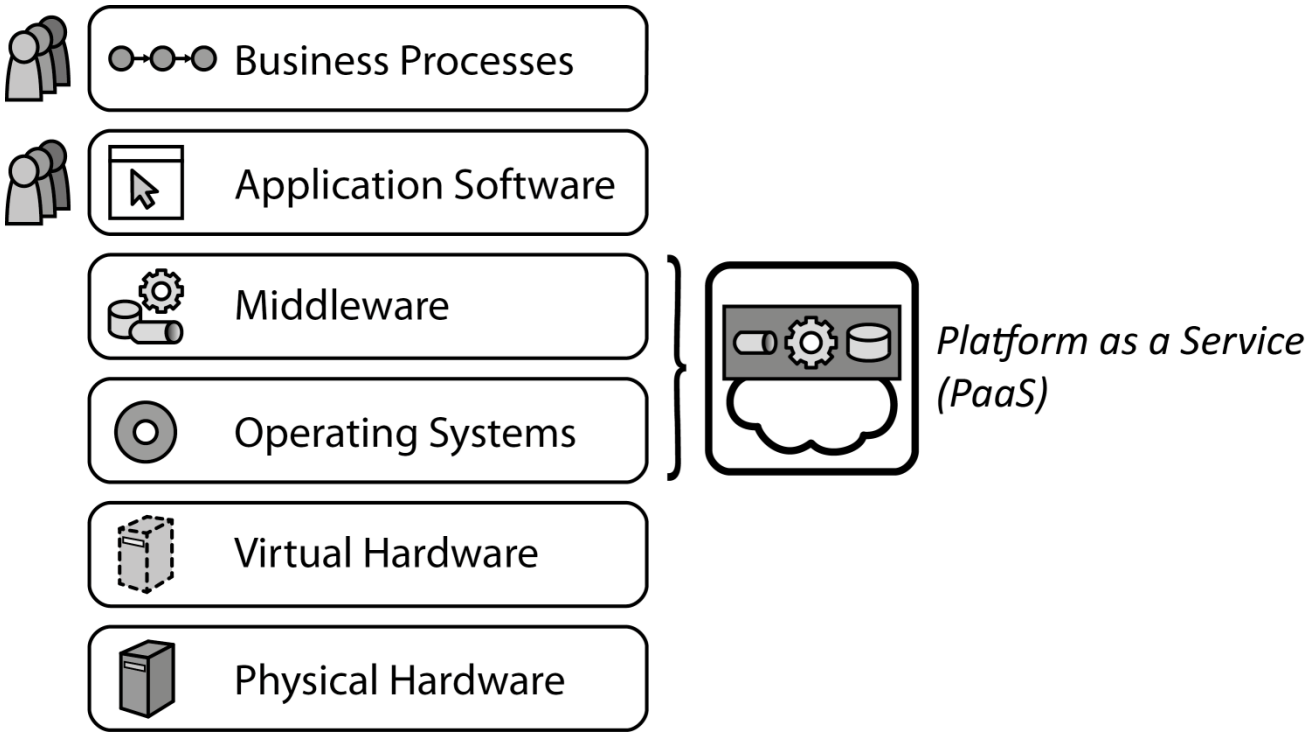


Platform as a Service (PaaS)

Providers share IT resources providing an application hosting environment between customers to enable self-service, rapid elasticity, and pay-per-use pricing.



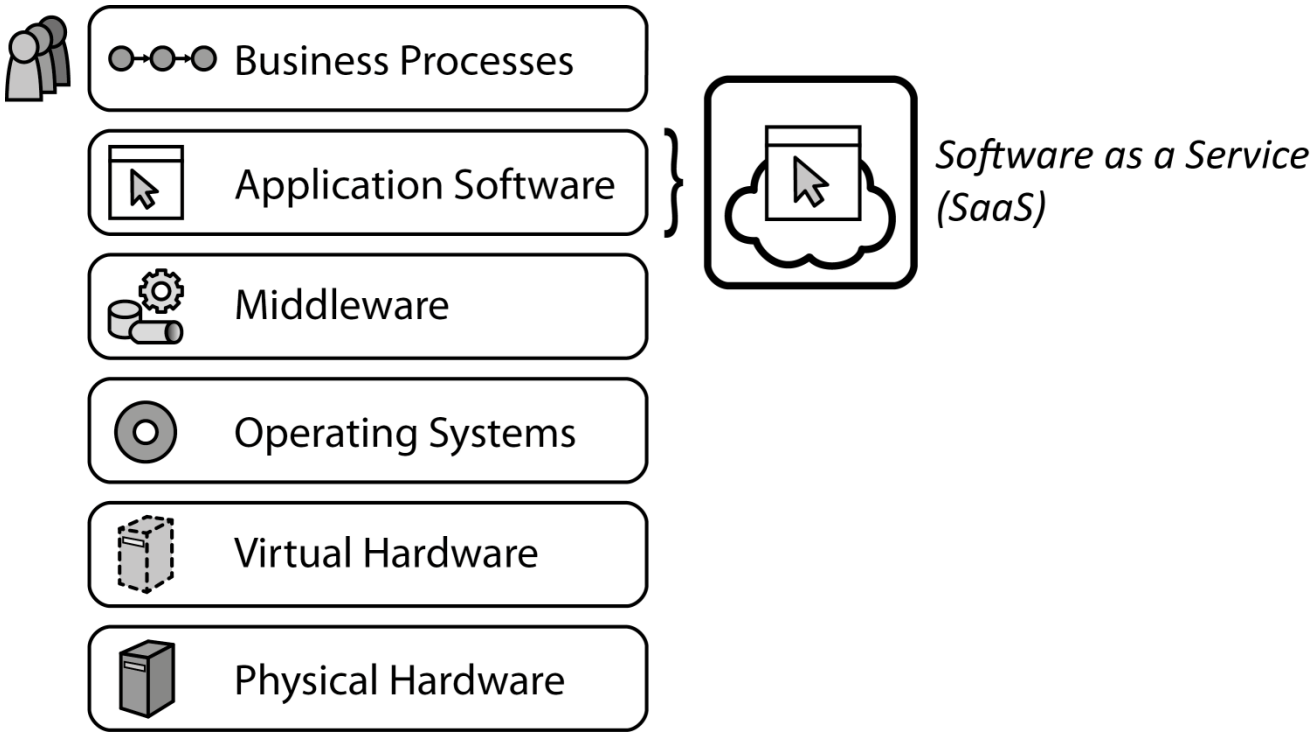
How can custom applications of the same customer or different customers share an execution environment so that it can be used on-demand with a pay-per-use pricing model?



Software as a Service (SaaS)

Providers share IT resources providing human-usable application software between customers to enable self-service, rapid elasticity, and pay-per-use pricing.

How can customers share a provider-supplied software application so that it can be used on-demand with a pay-per-use pricing model?



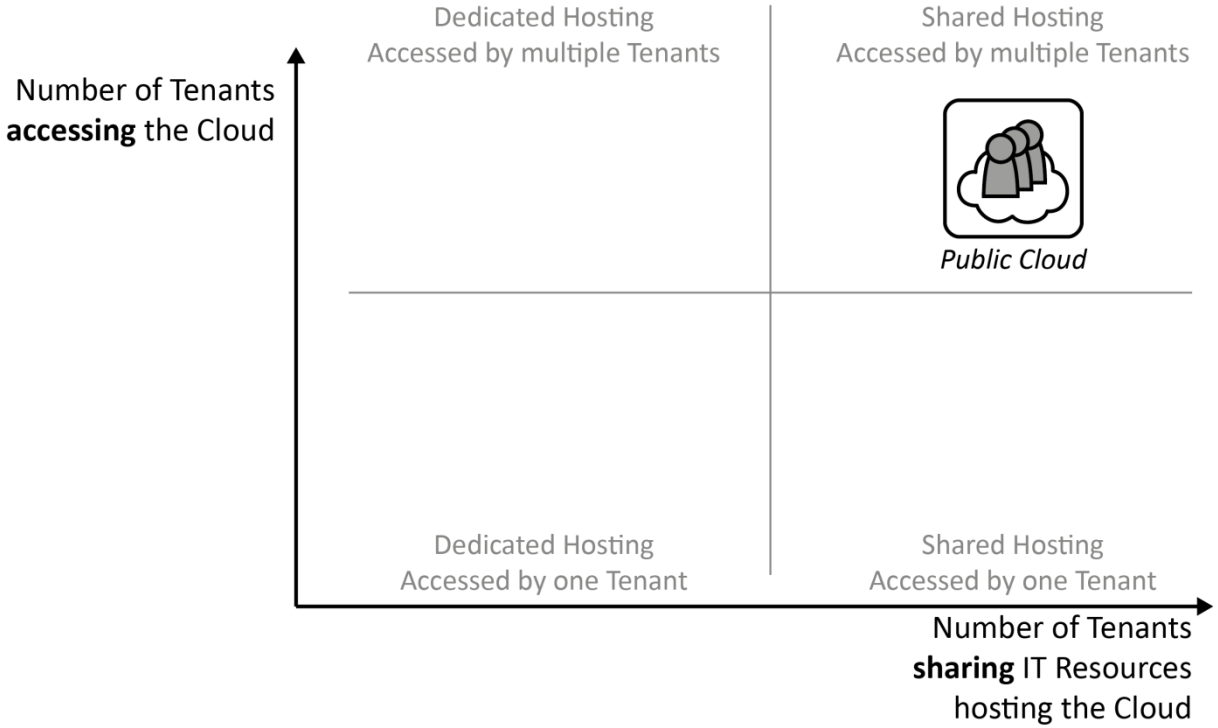
Cloud Computing Fundamentals
Cloud Deployment Models

http://www.cloudcomputingpatterns.org

Public Cloud

IT resources are provided as a service to a very large customer group in order to enable elastic use of a static resource pool.

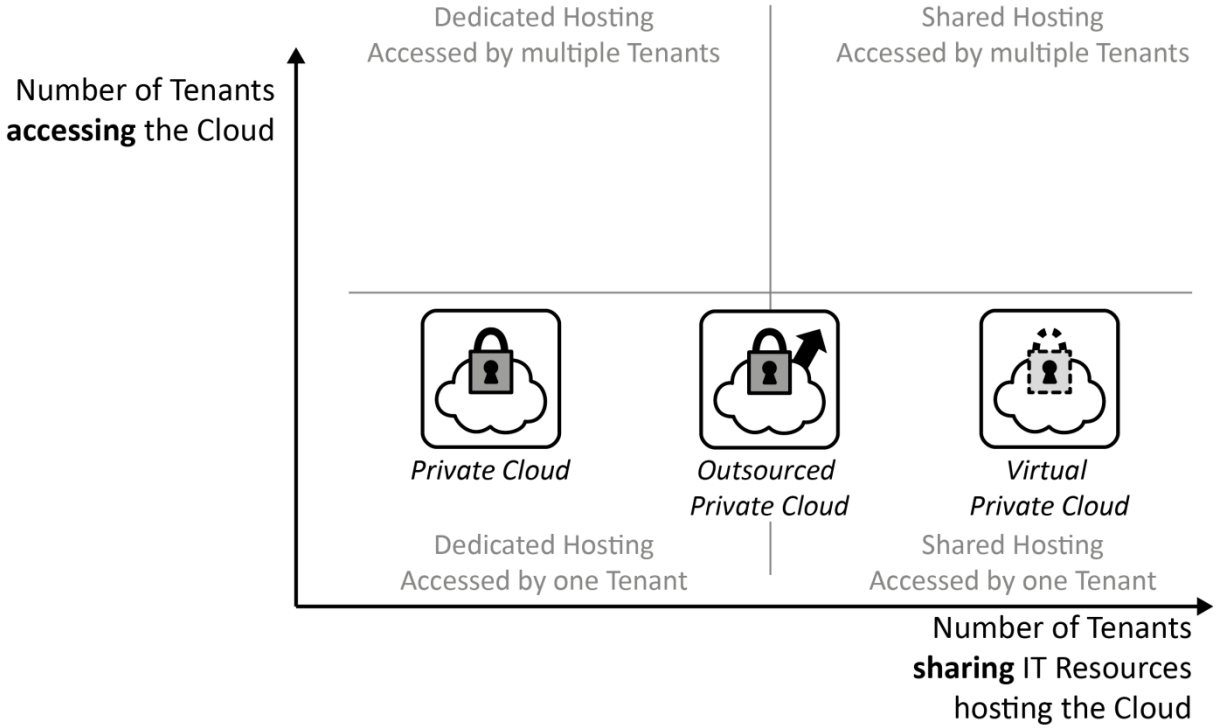
How can the cloud properties be provided to a large customer group?



Private Cloud

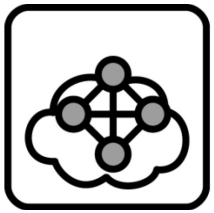
IT resources are provided as a service exclusively to one customer to meet requirements on privacy, security, and trust while enabling elastic use of a static resources.

How can the cloud properties be provided in environments with high privacy, security and trust requirements?

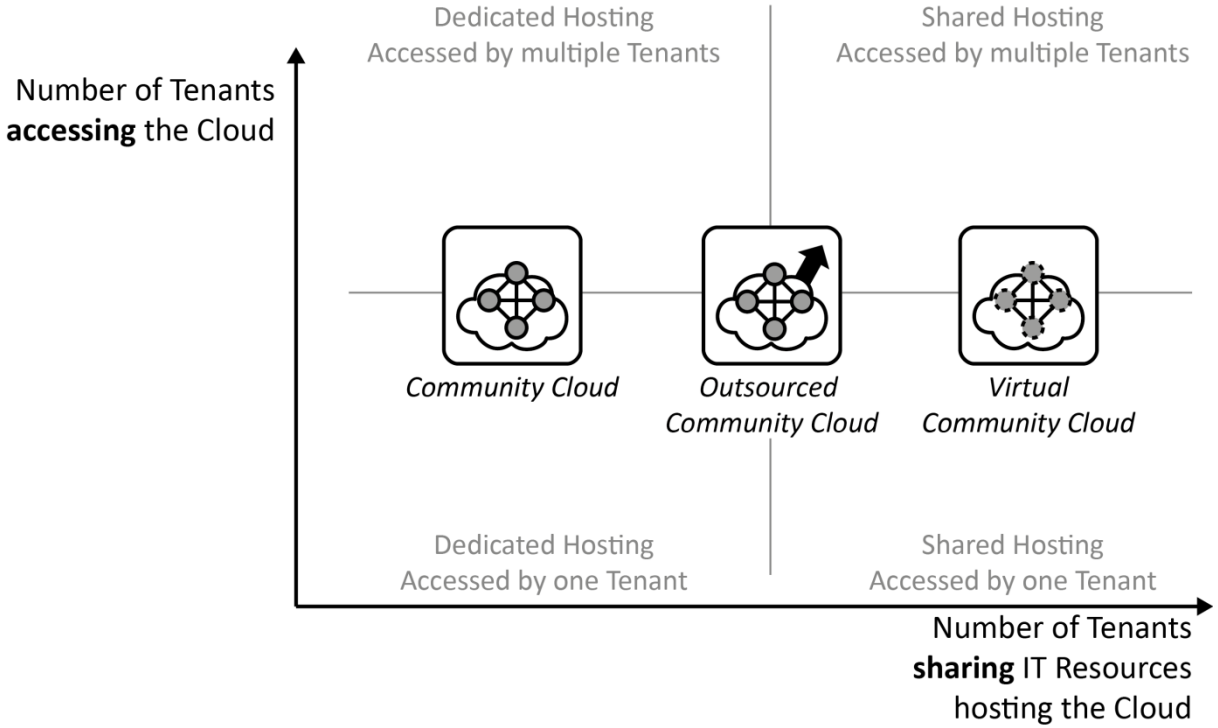


Community Cloud

IT resources are provided as a service to a group of customers trusting each other in order to enable collaborative elastic use of a static resource pool.



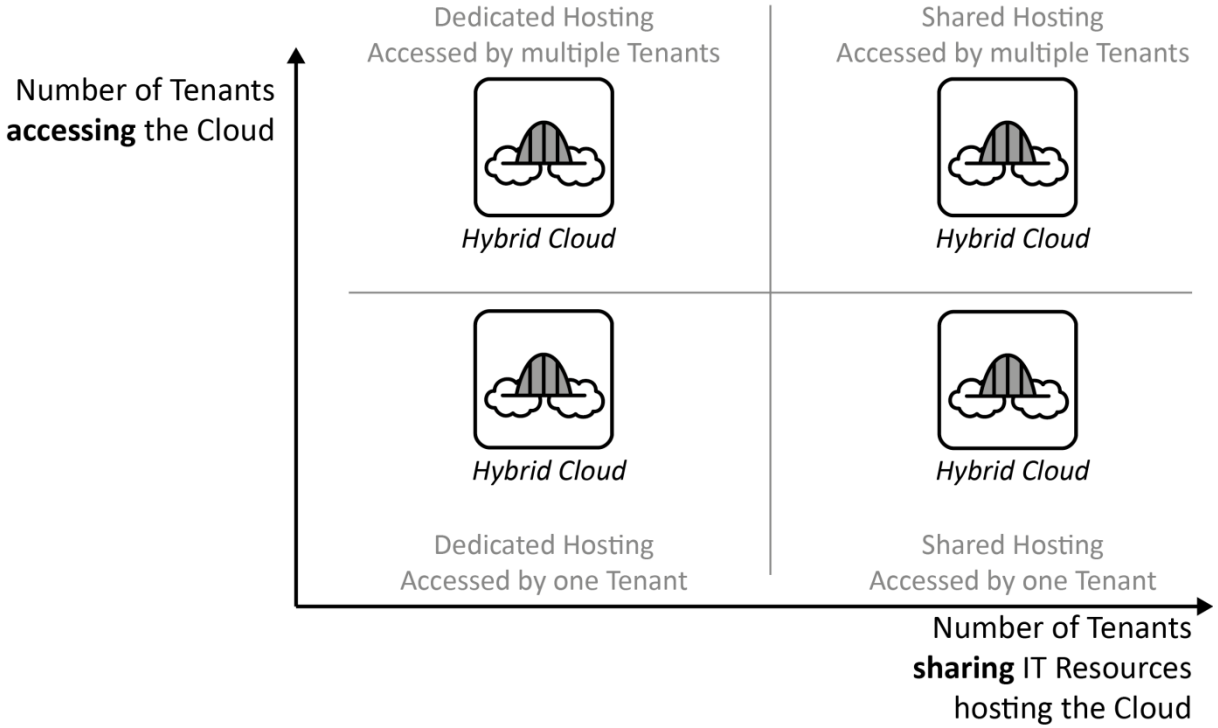
How can the cloud properties be provided exclusively to a group of customers forming a community of trust?



Hybrid Cloud

Different clouds and static data centers are integrated to form a homogeneous hosting environment.

How can the cloud properties be provided across clouds and other environments?

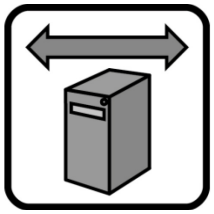


Cloud Offering Patterns
Cloud Environments

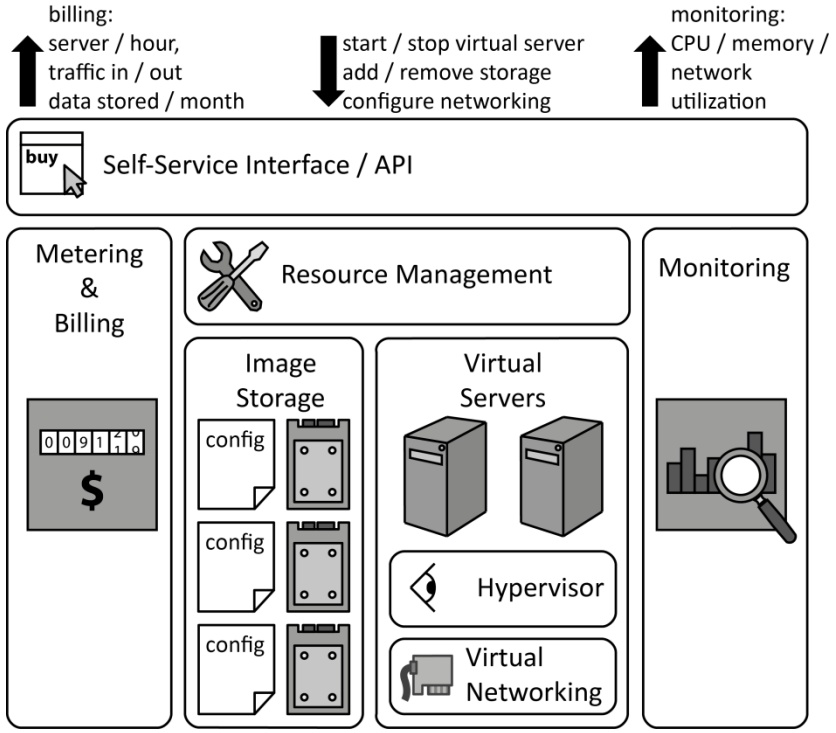
http://www.cloudcomputingpatterns.org

Elastic Infrastructure

Hosting of virtual servers, disk storage, and configuration of network connectivity is offered via a self-service interface over a network.

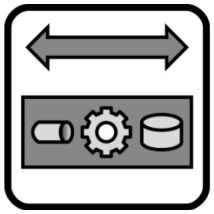


How do cloud offerings providing infrastructure resources behave and how should they be used in applications?

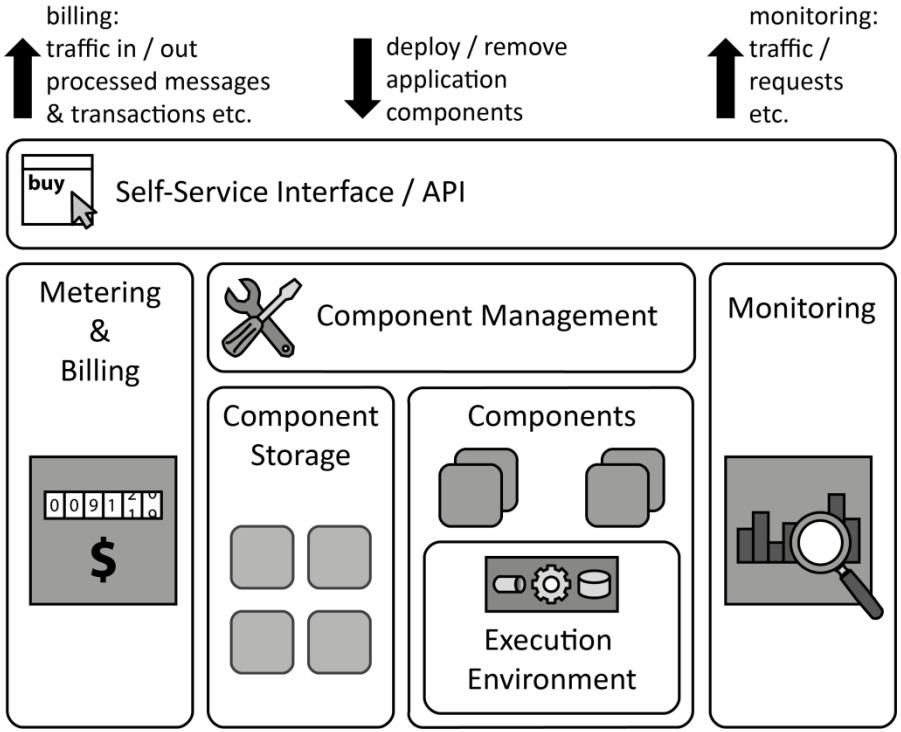


Elastic Platform

Middleware for the execution of custom applications, their communication, and data storage is offered via a self-service interface over a network.

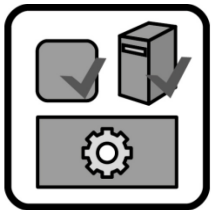


How do cloud offerings providing execution environments behave and how should they be used in applications?

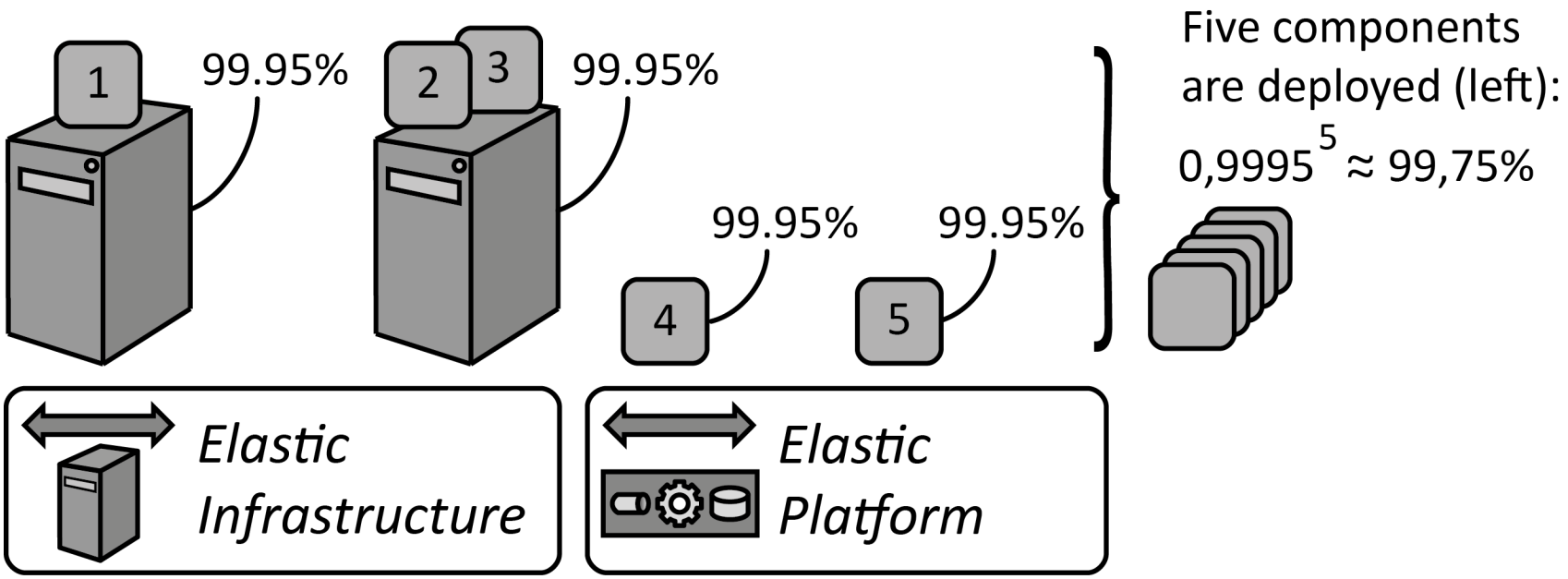


Node-based Availability

A cloud provider guarantees the availability of individual nodes, such as individual virtual servers, middleware components or hosted application components.

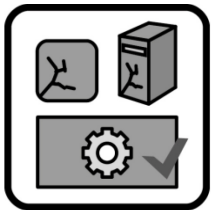


How can providers express availability in a node-centric fashion, so that customers may estimate the availability of hosted applications?

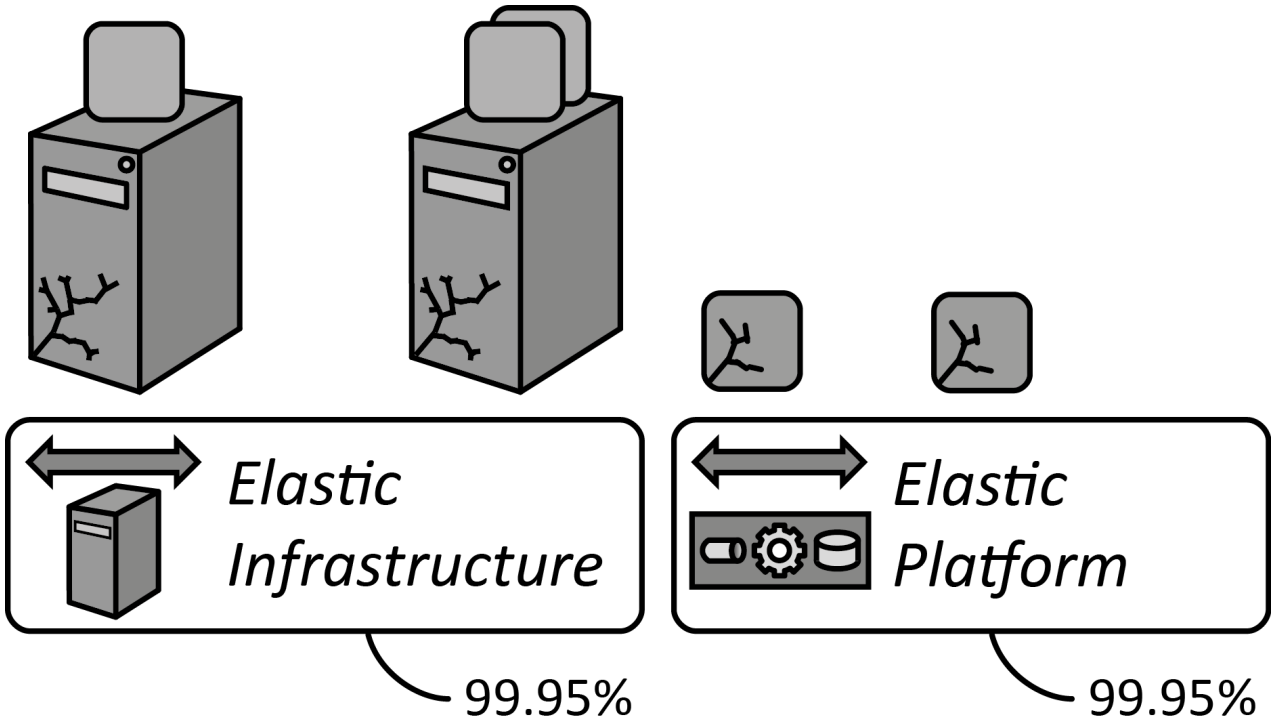


Environment-based Availability

A cloud provider guarantees the availability of the environment hosting individual nodes, such as virtual servers or hosted application components.



How can providers express availability in an environmental-centric fashion, so that customers may estimate the availability of hosted applications?

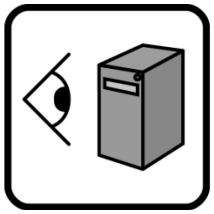


Cloud Offering Patterns
Processing Offerings

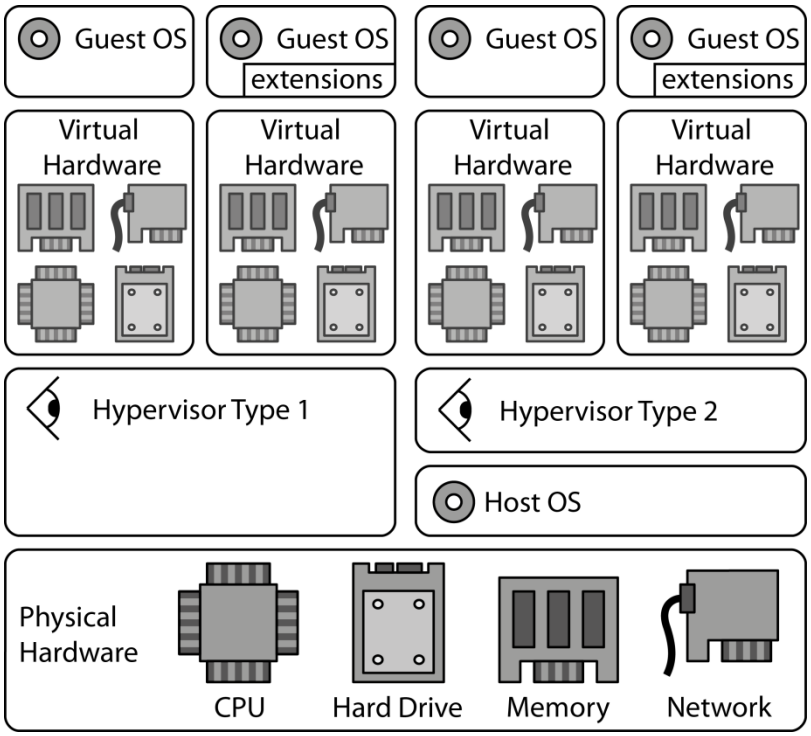
http://www.cloudcomputingpatterns.org

Hypervisor

To enable the elasticity of clouds, the time required to provision and decommission servers is reduced through hardware virtualization.

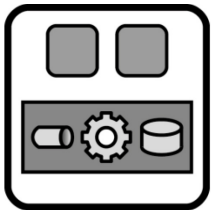


How can virtual hardware that has been abstracted from physical hardware be used in applications?

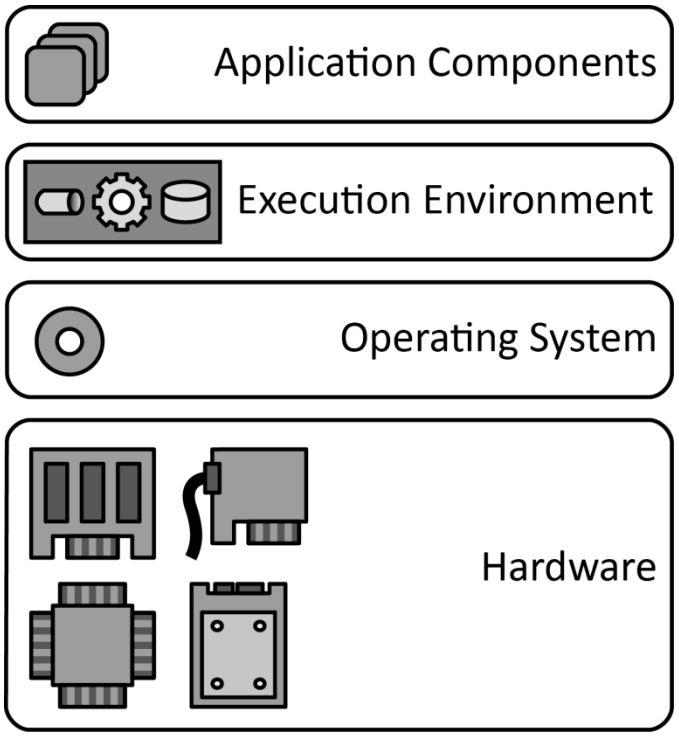


Execution Environment

To avoid duplicate implementation of functionality, applications are deployed to a hosting environment providing middleware services and common functionality.

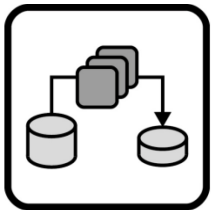


How can multiple application components share a hosting environment efficiently?

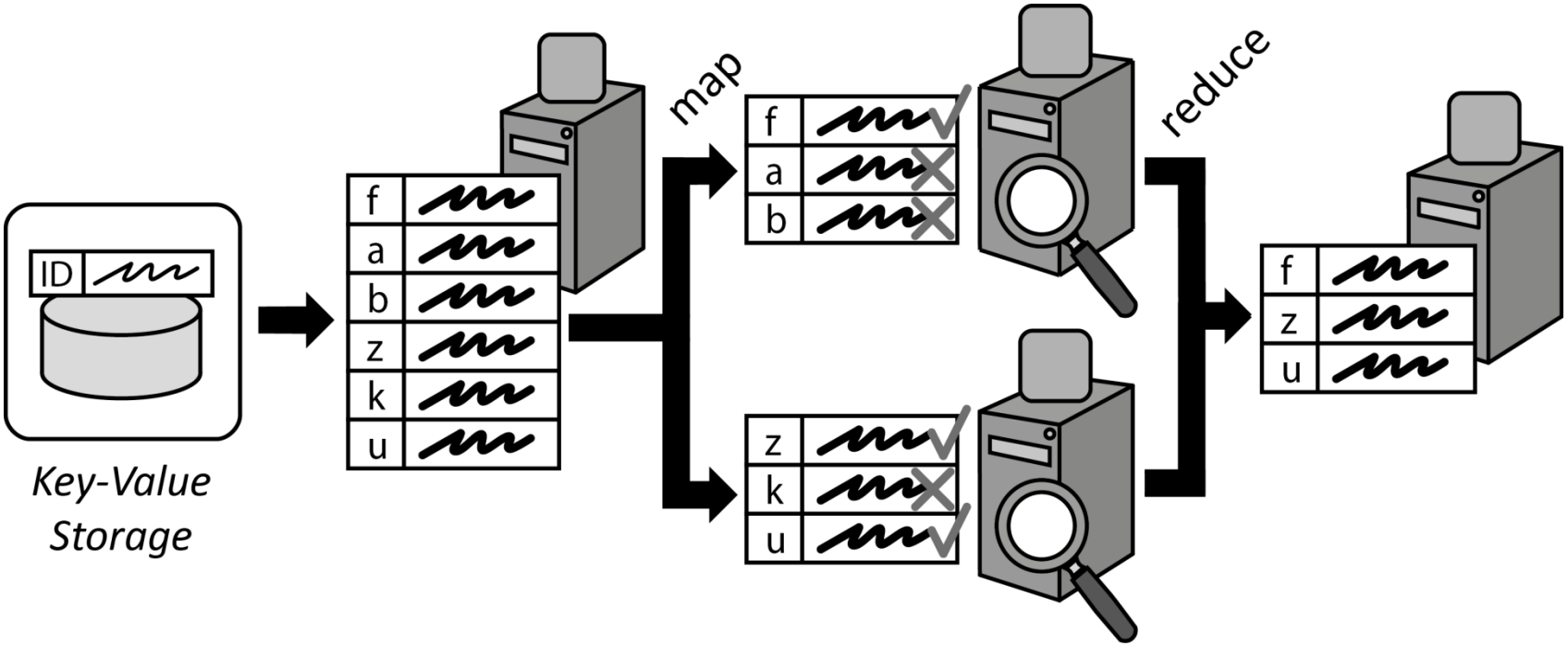


Map Reduce

Large data sets to be processed are divided into smaller data chunks and distributed among processing application components. Individual results are later consolidated.



How can the performance of complex processing of large data sets be increased through scaling out?

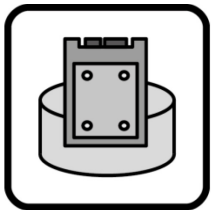


Cloud Offering Patterns
Storage Offerings

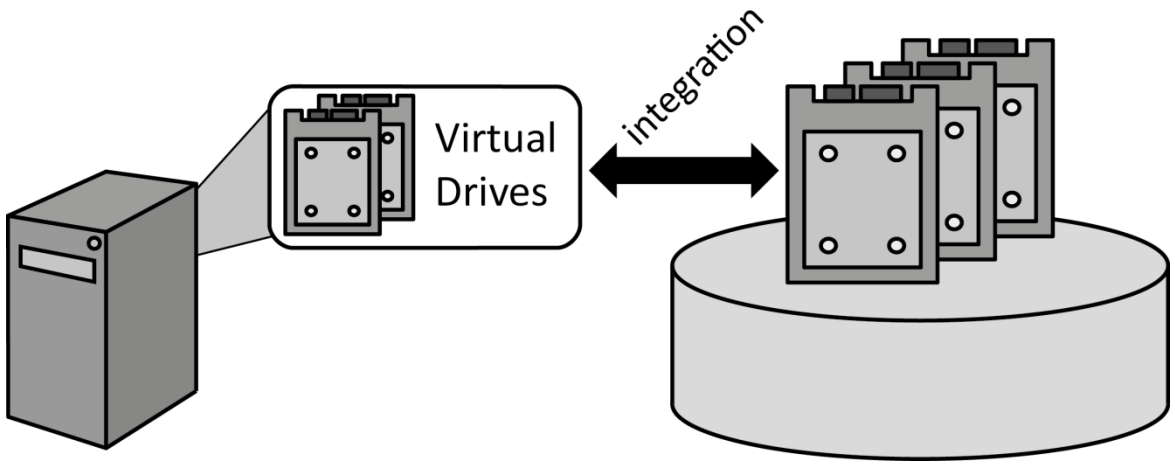
http://www.cloudcomputingpatterns.org

Block Storage

Centralized storage is integrated into servers as a local hard drive to enable access to this storage via the local file system.

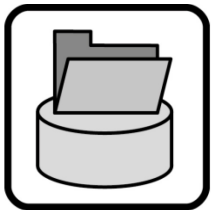


How can central storage be accessed as a local drive by servers and hosted applications?

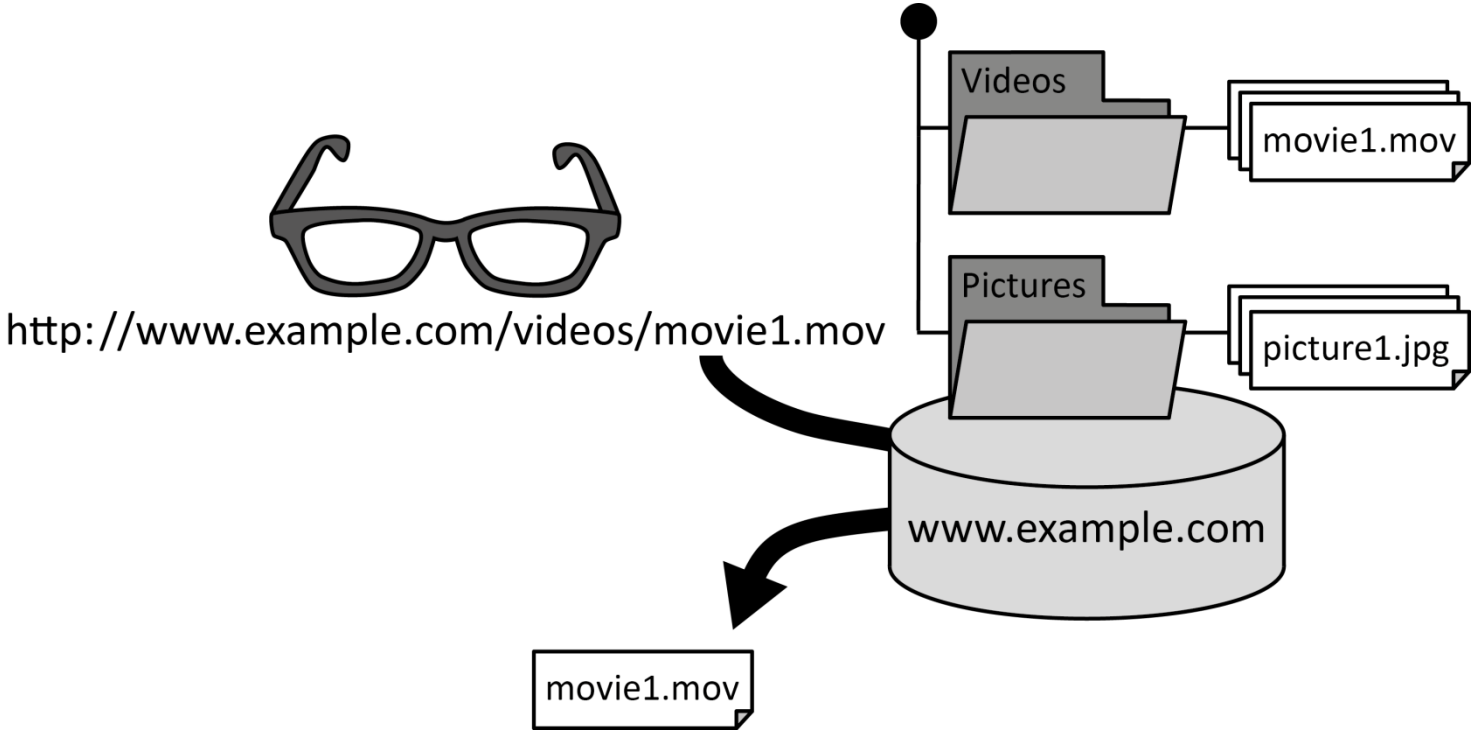


Blob Storage

Data is provided in form of large files that are made available in a file system-like fashion.

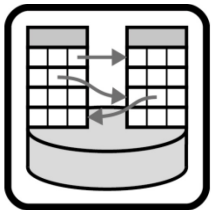


How can large files be stored, organized, and made available over a network?

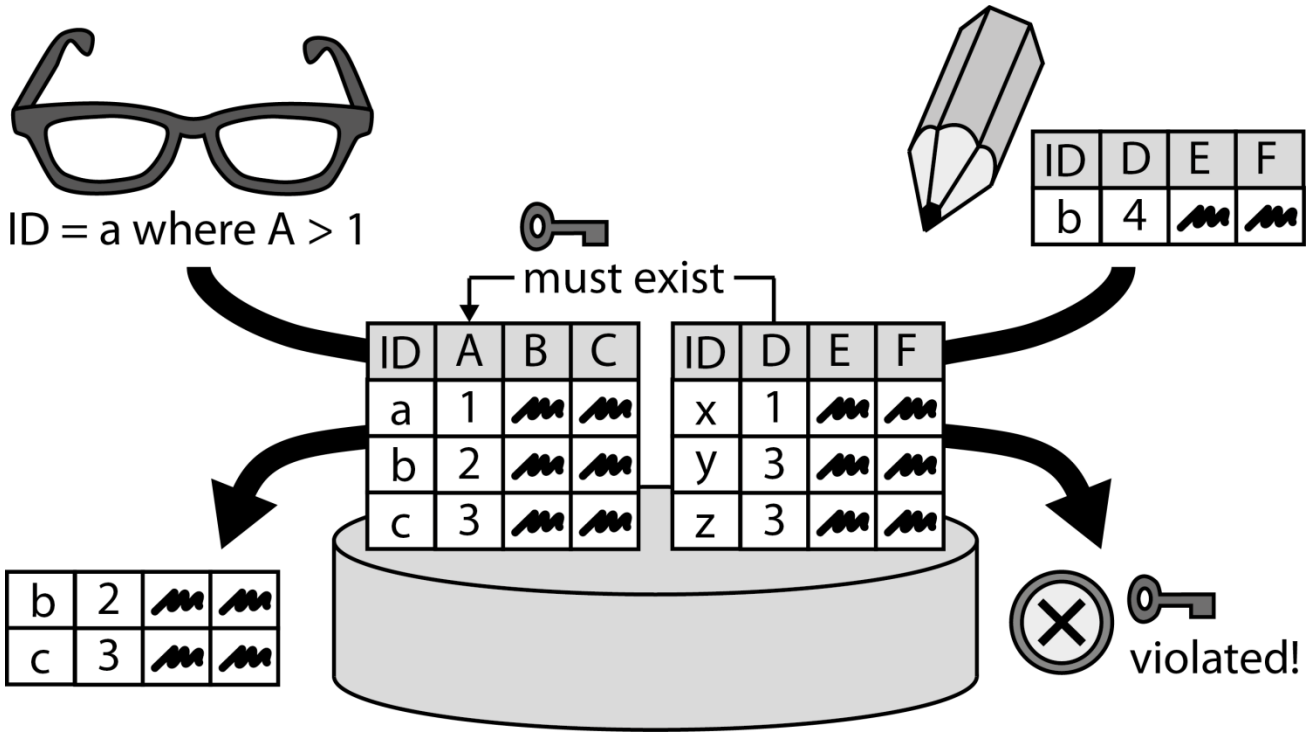


Relational Database

Data is structured according to a schema that is enforced during data manipulation and enables expressive queries of handled data.



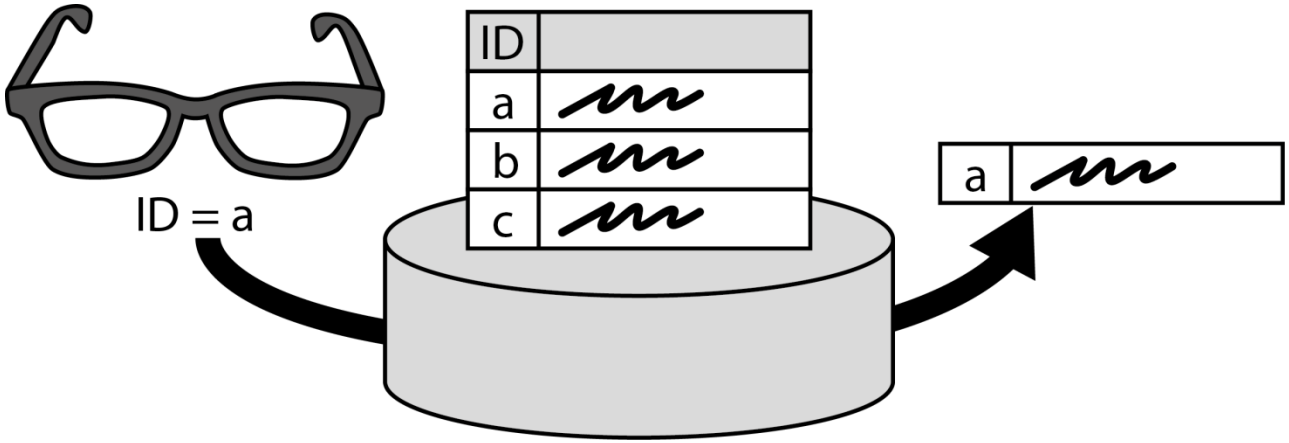
How can data elements be stored so that relations between them can be expressed and expressive queries are enabled to retrieve required information effectively?



Key-value Storage

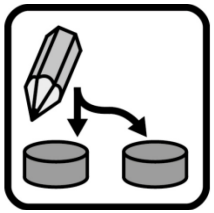
Semi-structured or unstructured data is stored with limited querying support but high-performance, availability, and flexibility.

How can key-value elements be stored to support scale out and an adjustable data structure?

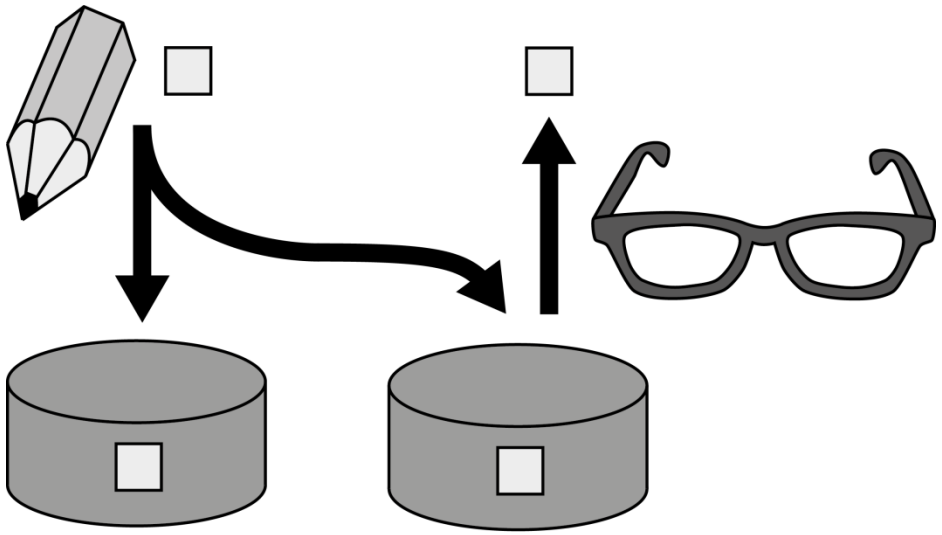


Strict Consistency

Data is stored at different locations to improve response time and to avoid data loss in case of failures while consistency of replicas is ensured at all times.



How can data be distributed among replicas to increase availability, while ensuring data consistency at all times?



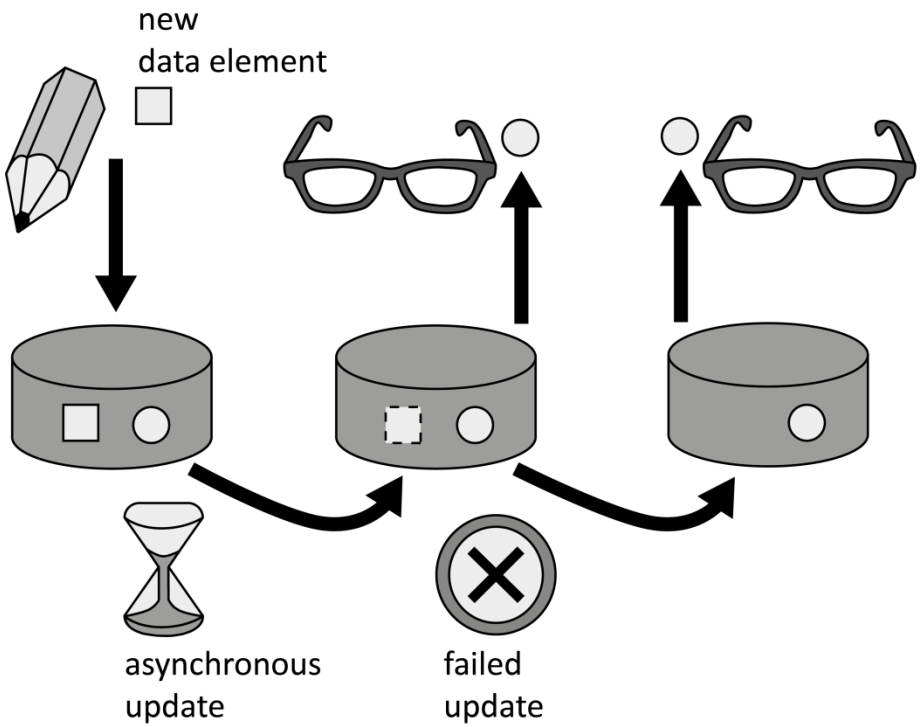
number of replicas (n) = 2
replicas accessed to write (w) = 2
replicas accessed to read (r) = 1

$n < w + r$

Eventual Consistency

Performance and availability of data in case of network partitioning are enabled by ensuring data consistency eventually and not at all times.

How can data be distributed among replicas with focus on increased availability and performance, while being resilient towards connectivity problems?

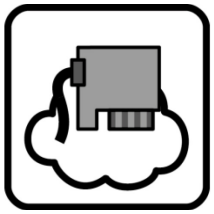


Cloud Offering Patterns
Communication Offerings

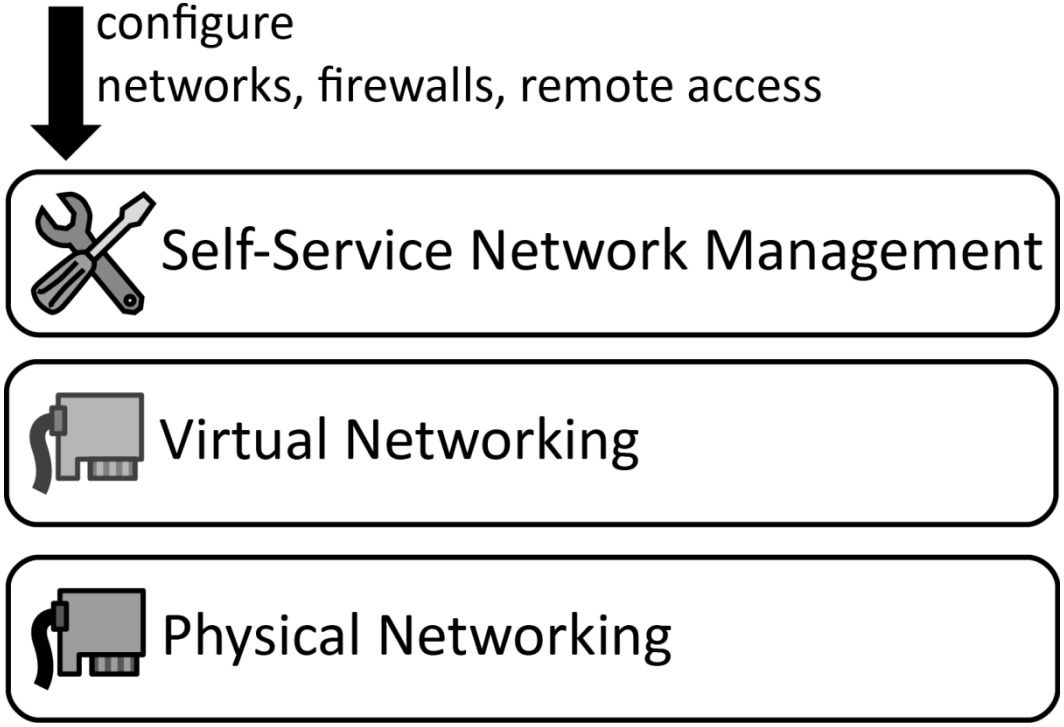
http://www.cloudcomputingpatterns.org

Virtual Networking

Networking resources are virtualized to empower customers to configure networks, firewalls, and remote access using a self-service interface.

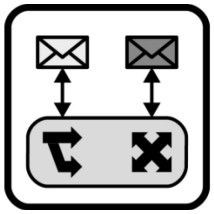


How can network connectivity between IT resources hosted in a cloud be configured dynamically and on-demand?

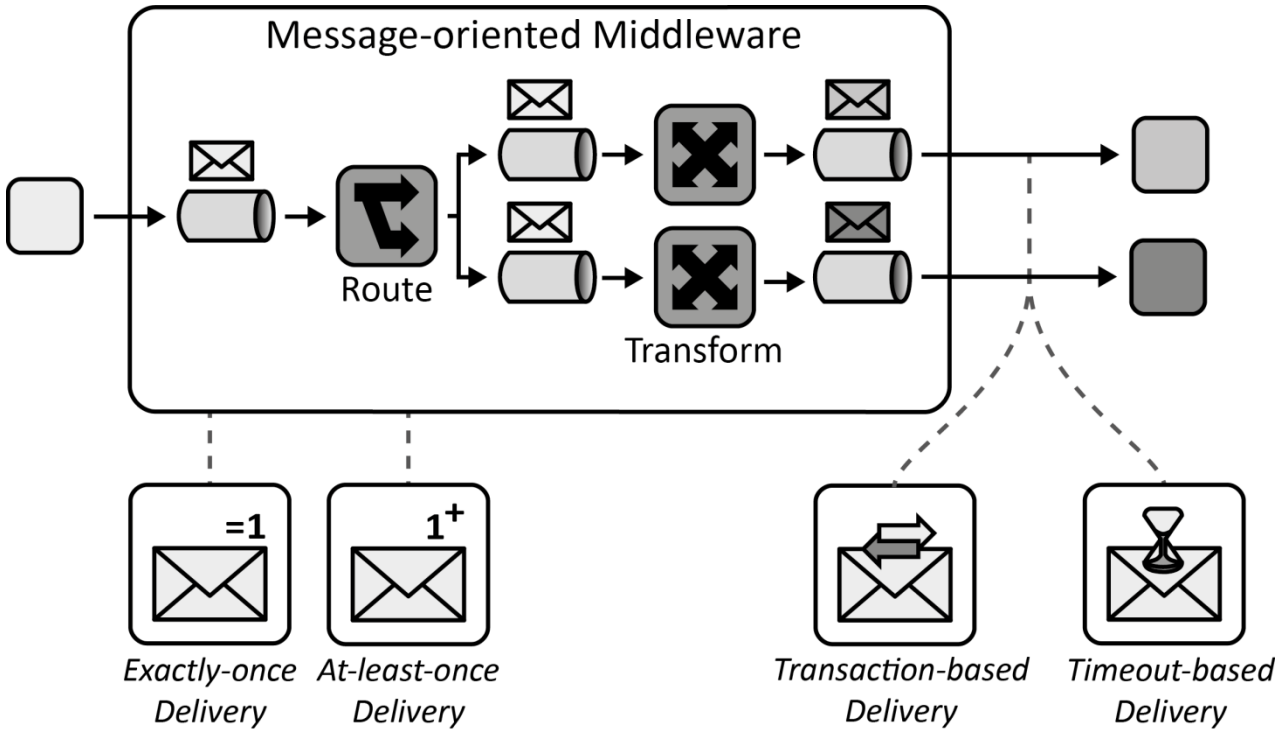


Message-oriented Middleware

Asynchronous communication is provided while hiding complexity of addressing, routing, or data formats to make interaction robust and flexible.



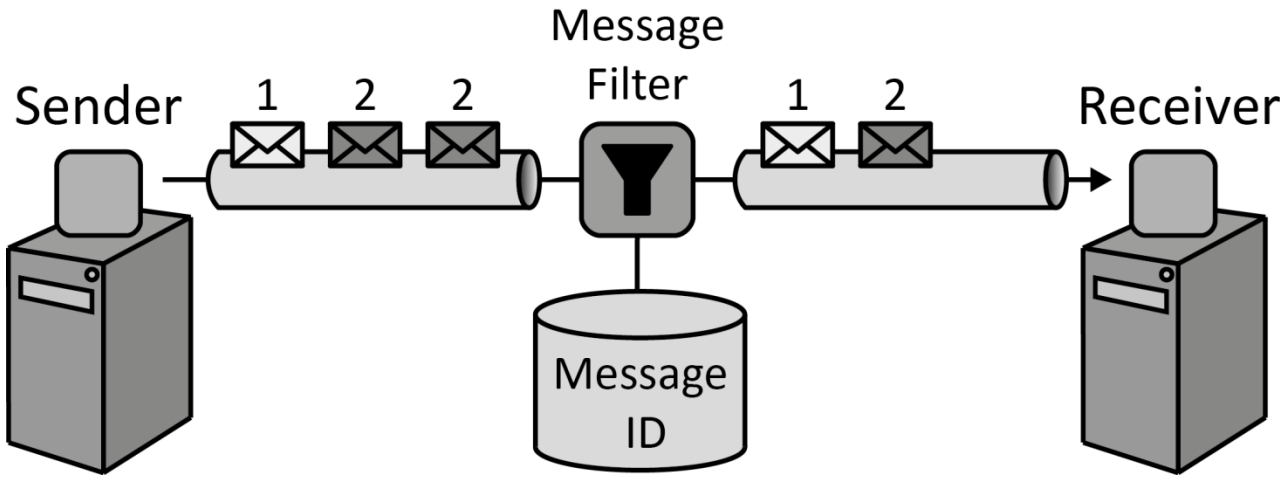
How can communication partners exchange information asynchronously with a communication partner?



Exactly-once Delivery

The messaging system ensures that each message is delivered exactly once by filtering possible message duplicates automatically.

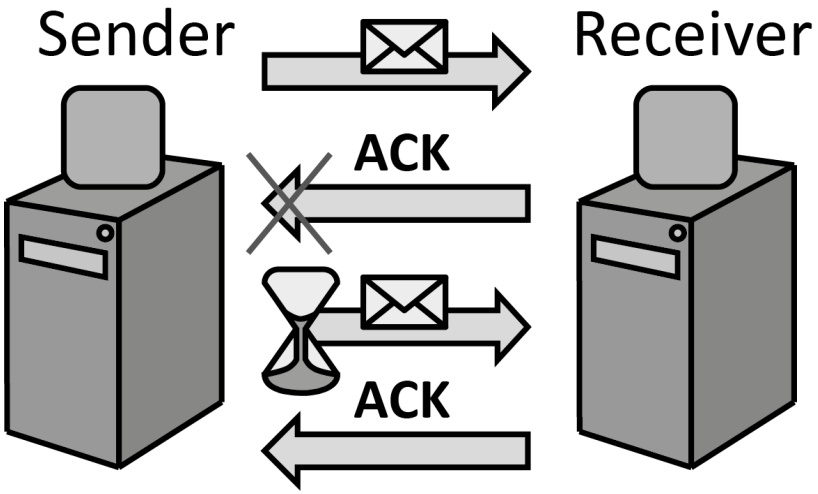
How can it be assured that a message is delivered only exactly-once to a receiver?



At-least-once Delivery

In case of failures that lead to message loss, messages are retransmitted to assure they are delivered at least once.

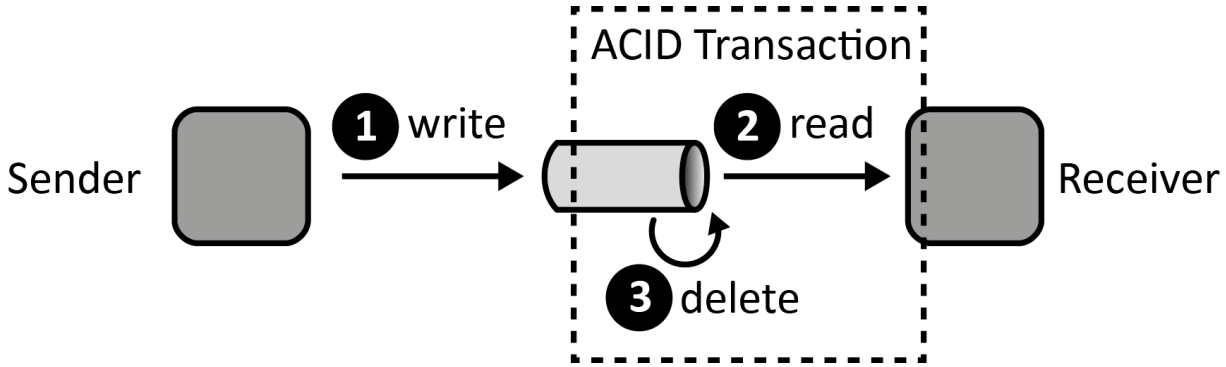
How can a message-oriented middleware ensure that messages are received successfully at least once?



Transaction-based Delivery

Clients retrieve messages under a transactional context to ensure that messages are received by a handling component.

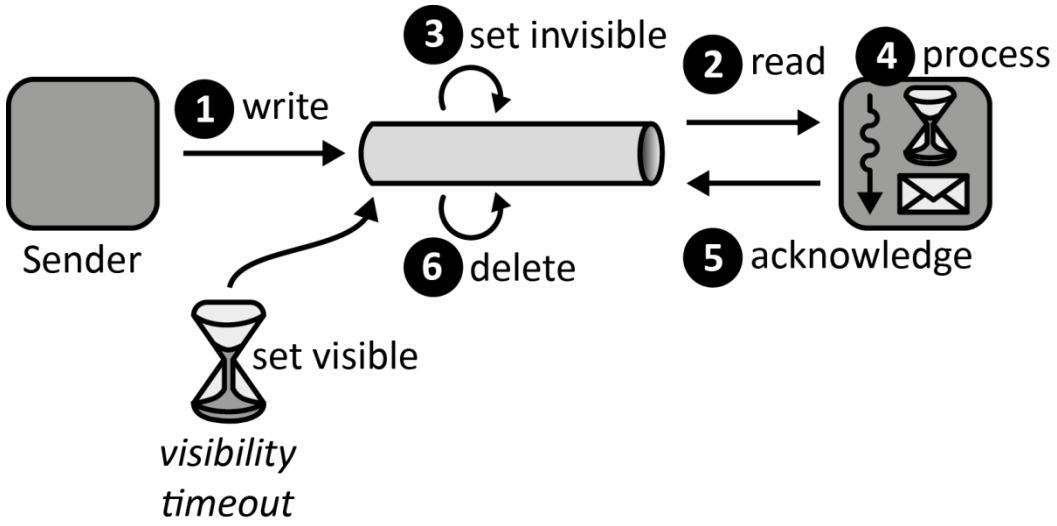
How can it be ensured that messages are only deleted from a message queue if they have been received successfully?



Timeout-based Delivery

Clients acknowledge message receptions to ensure that messages are received properly.

How can it be ensured that messages are only deleted from a message queue if they have been received successfully at least once?

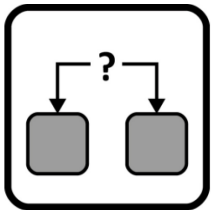


Cloud Application Architecture Patterns
Fundamental Cloud Architectures

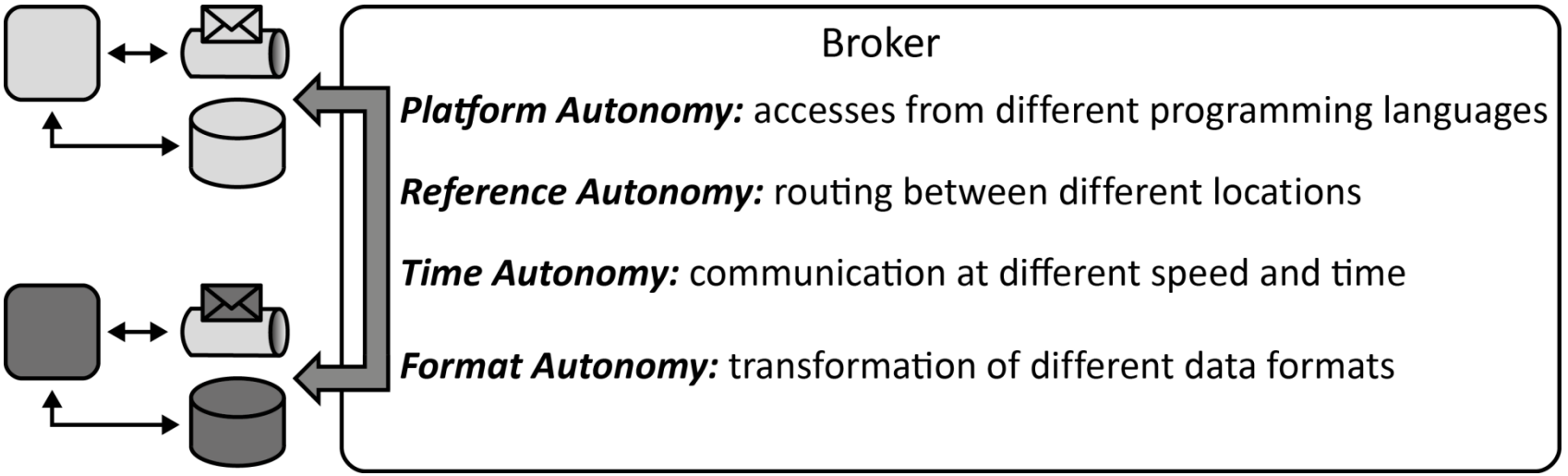
http://www.cloudcomputingpatterns.org

Loose Coupling

A broker encapsulates concerns of communication partner location, implementation platform, time of communication, and data format.

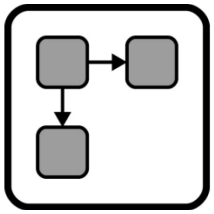


How can dependencies between distributed applications and between individual components of these applications be reduced?

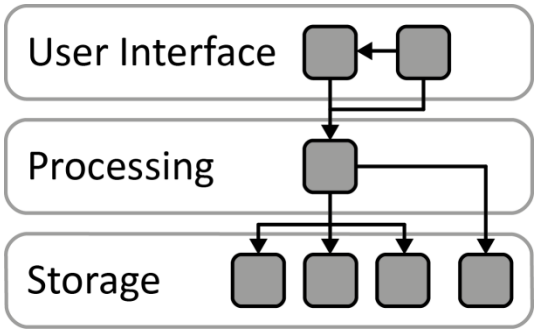


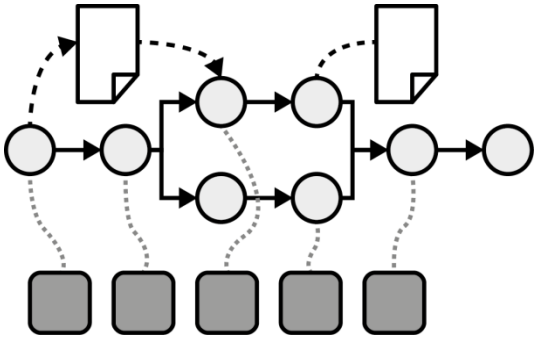
Distributed Application

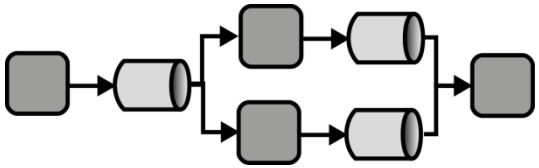
A cloud application divides provided functionality among multiple application components that can be scaled out independently.



How can application functionality be decomposed to be handled by separate application components?





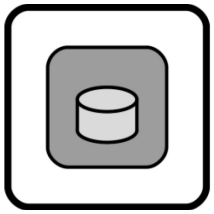


Cloud Application Architecture Patterns
Cloud Application Components

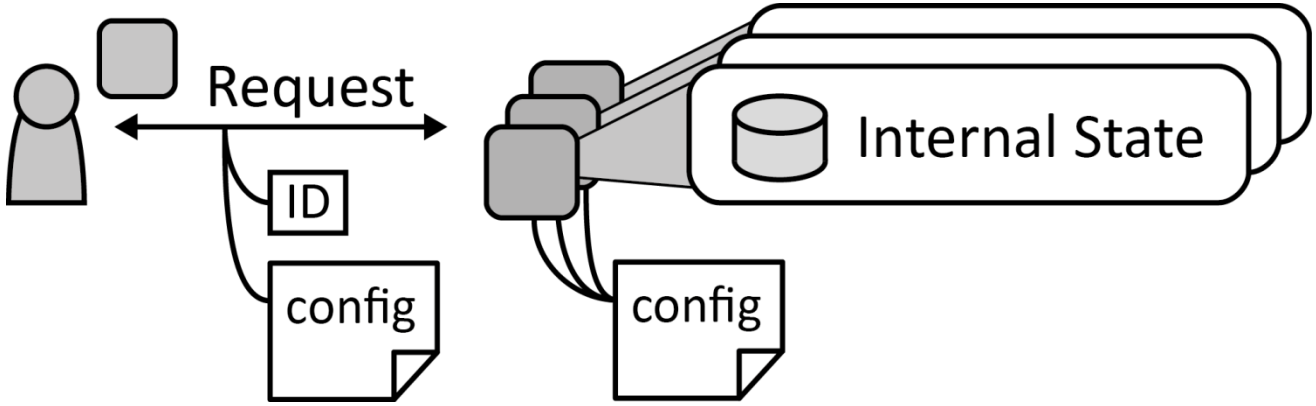
http://www.cloudcomputingpatterns.org

Stateful Component

Multiple instances of a scaled-out application component synchronize their internal state to provide a unified behavior.



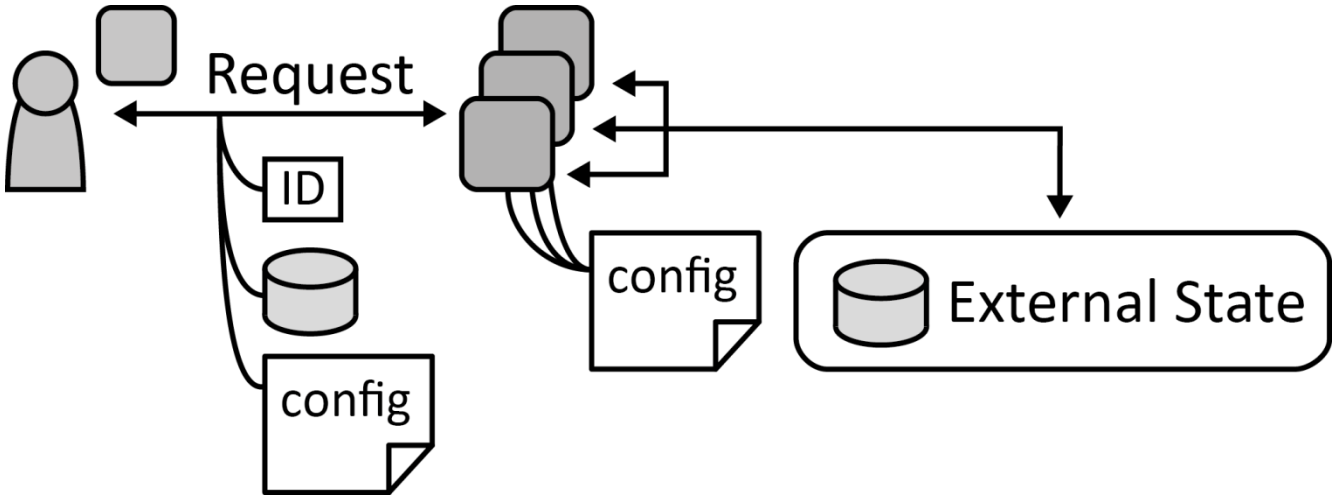
How can applications components that are scaled-out maintain a synchronized internal state?



Stateless Component

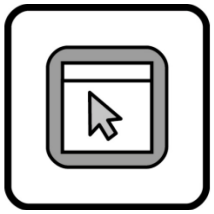
State is handled external of application components to ease their scaling-out and to make the application more tolerant to component failures.

How can elasticity and robustness of an application component be increased?

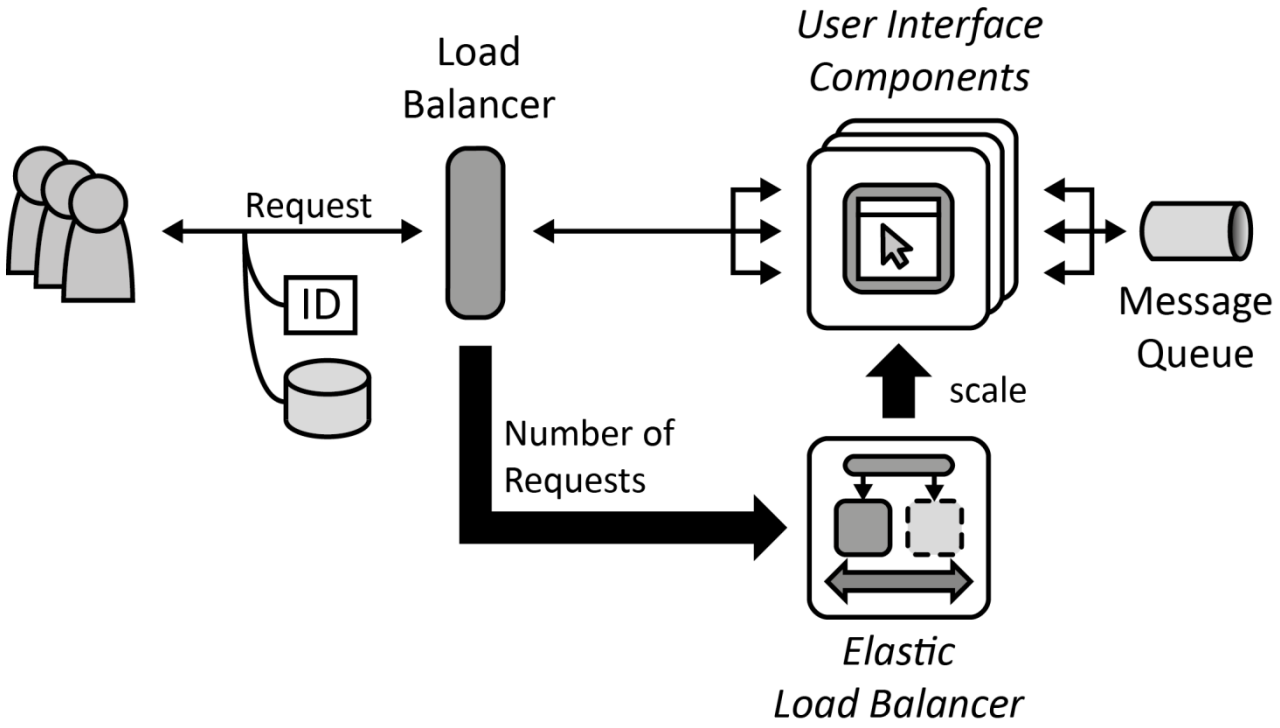


User Interface Component

Customizable synchronous user interfaces are accessed by humans, while application-internal interaction is realized asynchronously to ensure loose coupling.

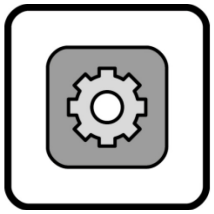


How can user interface components be accessed interactively by humans while being configurable and decoupled from the remaining application?

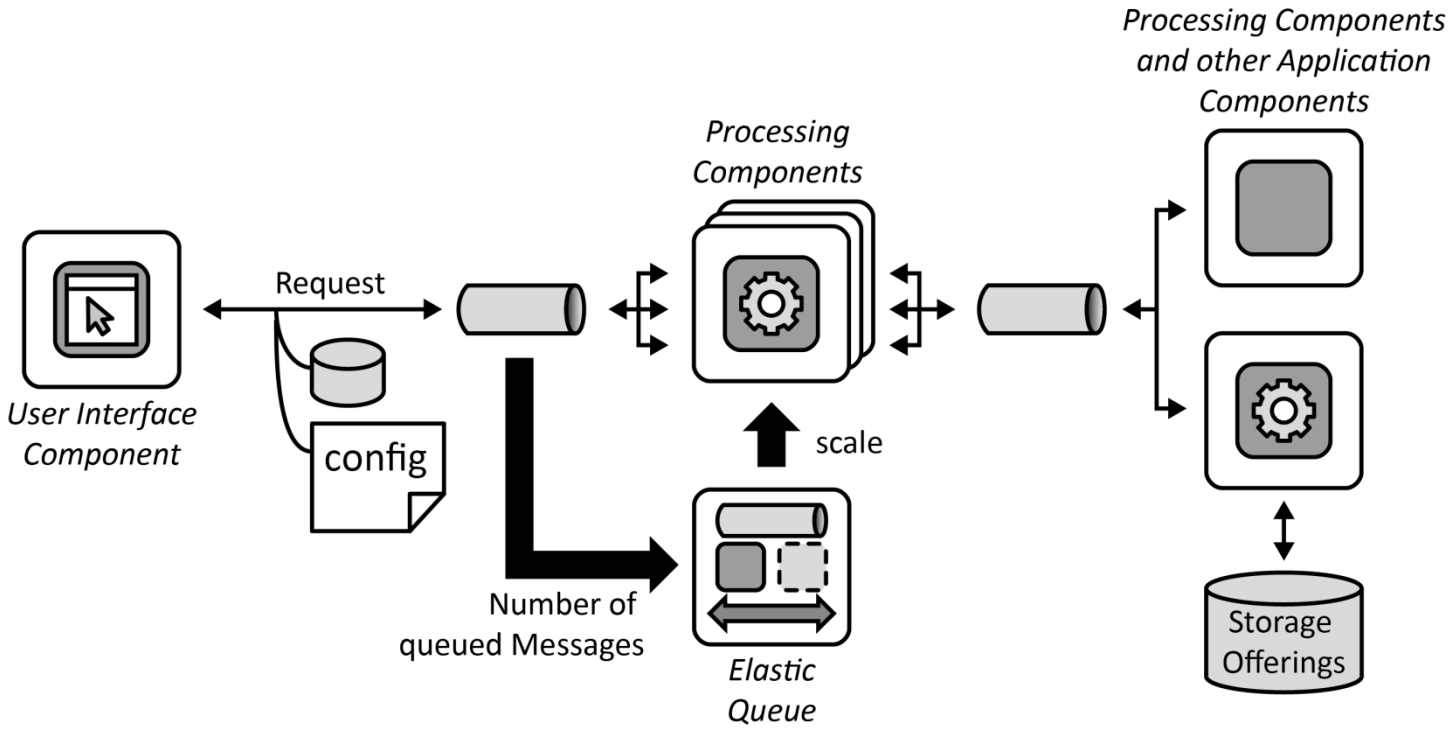


Processing Component

Processing functionality is handled by elastically scaled components. Functionality is made configurable to support different customer requirements.

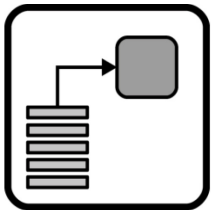


How can processing be scaled out elastically while being configurable regarding the supported functions?

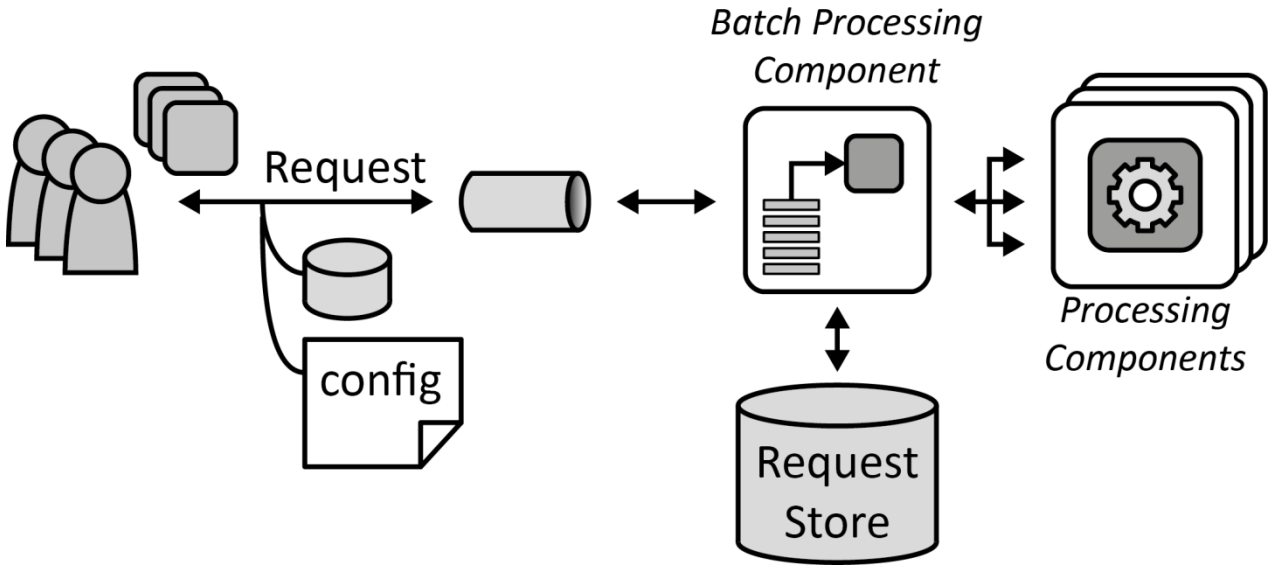


Batch Processing Component

Requests are delayed until environmental conditions make their processing feasible.



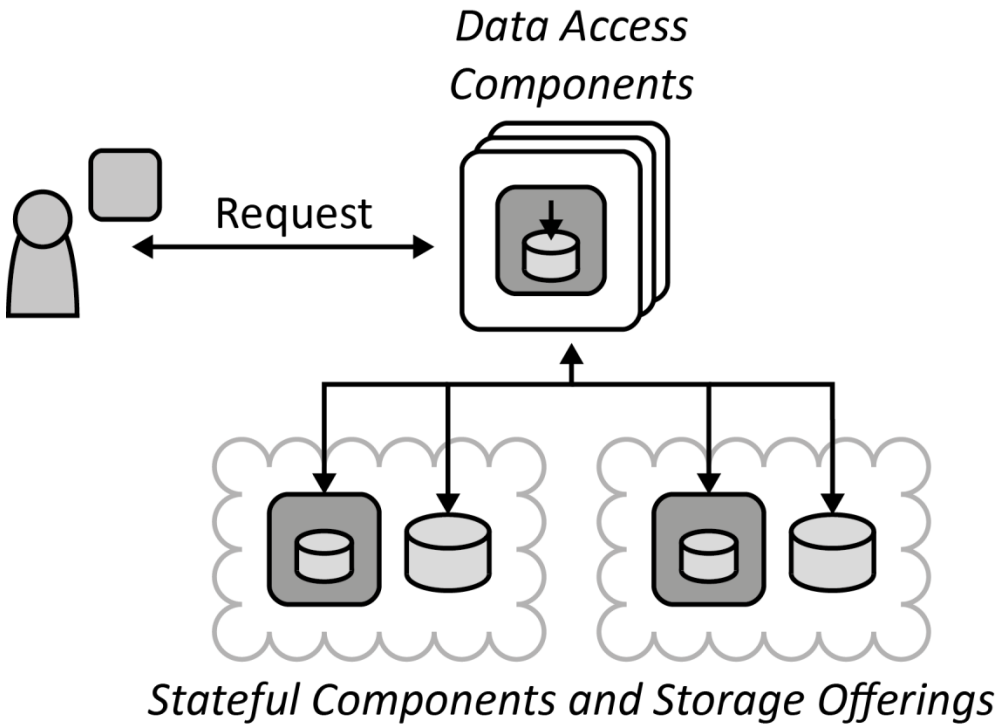
How can asynchronous processing requests be delayed to be handled when conditions for their processing are optimal?



Data Access Component

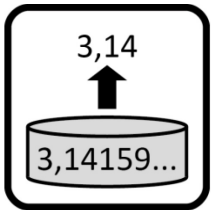
Access to data is provided by special components that isolate complexity, enable additional data consistency, and ensure adjustability of data elements.

How can the complexity of data storage be hidden while ensuring data structure configurability?

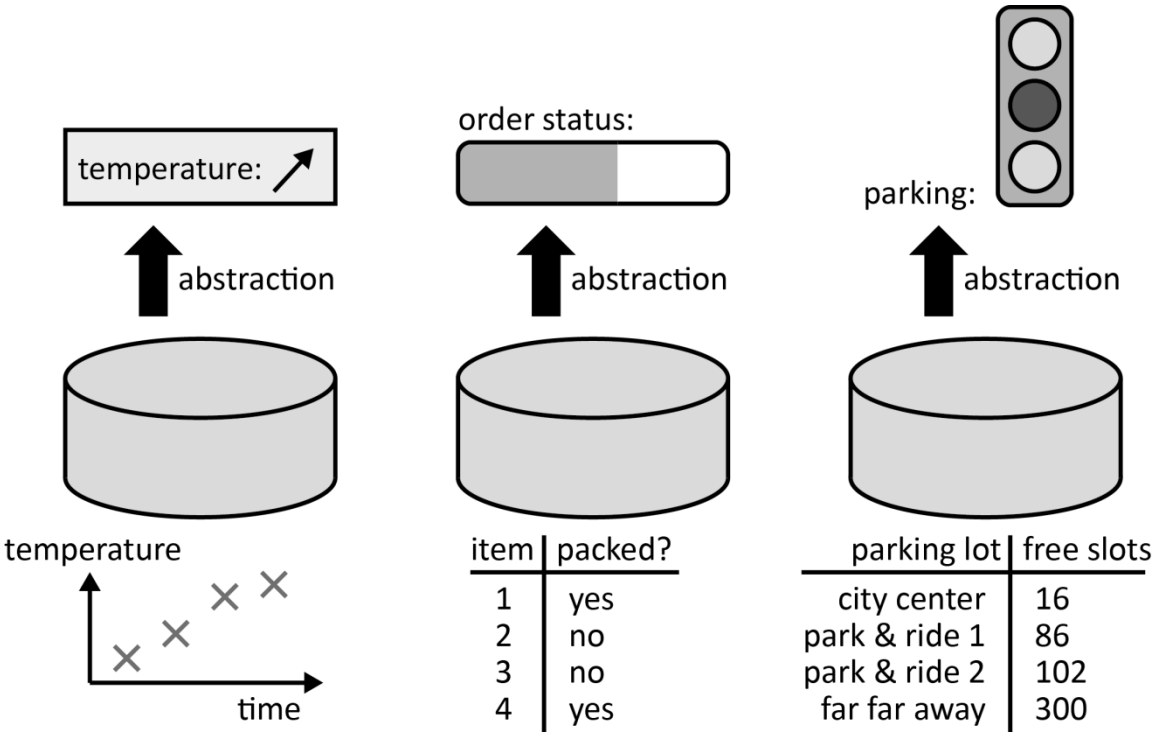


Data Abstractor

Data is abstracted to inherently support eventually consistent data storage through the use of abstractions and approximations.



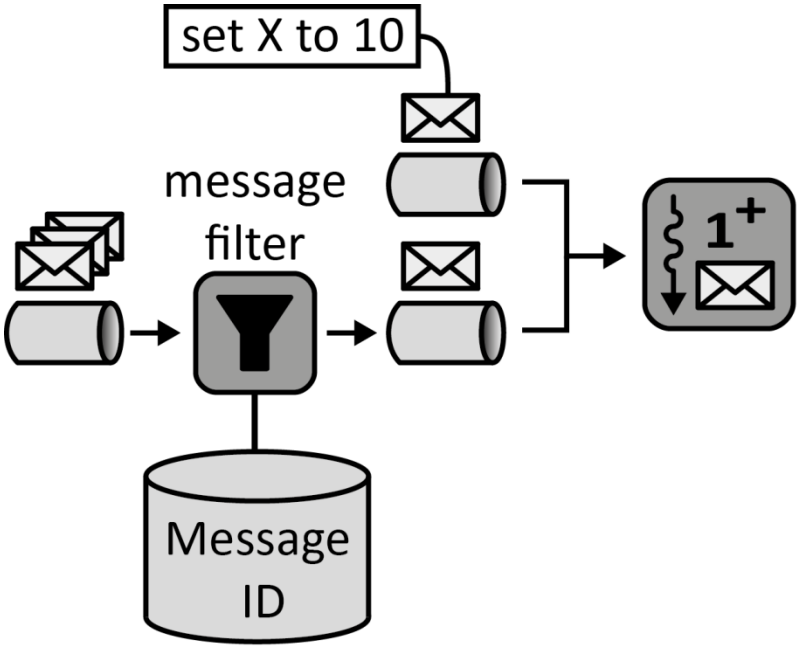
How can eventually consistent data be presented, so that possible inconsistencies are hidden?

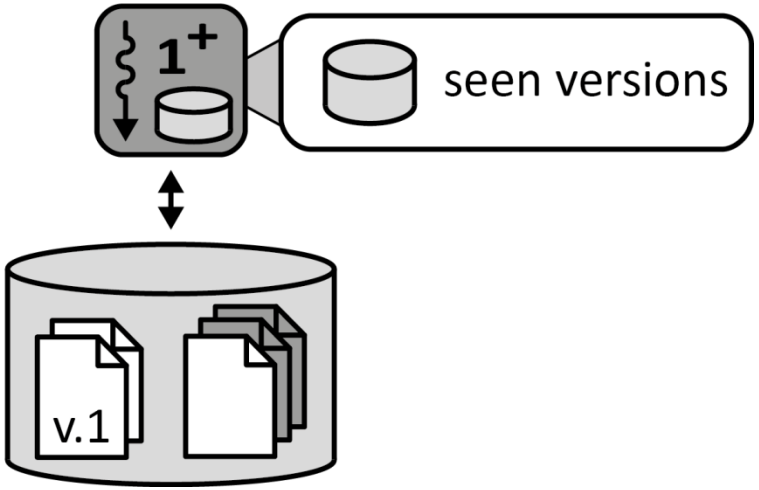


Idempotent Processor

Application functions detect duplicate messages and inconsistent data or are designed to be immune to these conditions.

How can an application component cope with message duplicates or data inconsistencies that could lead to duplicate function execution?

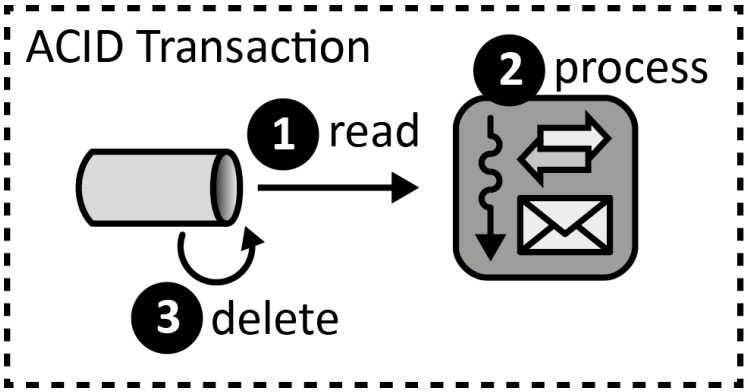


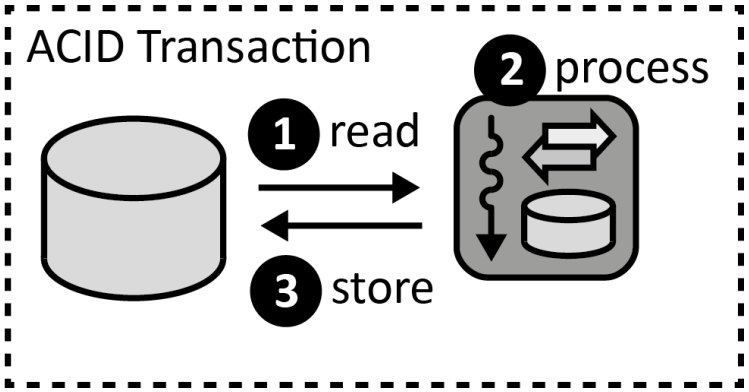


Transaction-based Processor

Components receive messages or read data and process the obtained information under a transactional context to ensure successful processing.

How can a processing component ensure that messages are processed and data is persisted successfully?

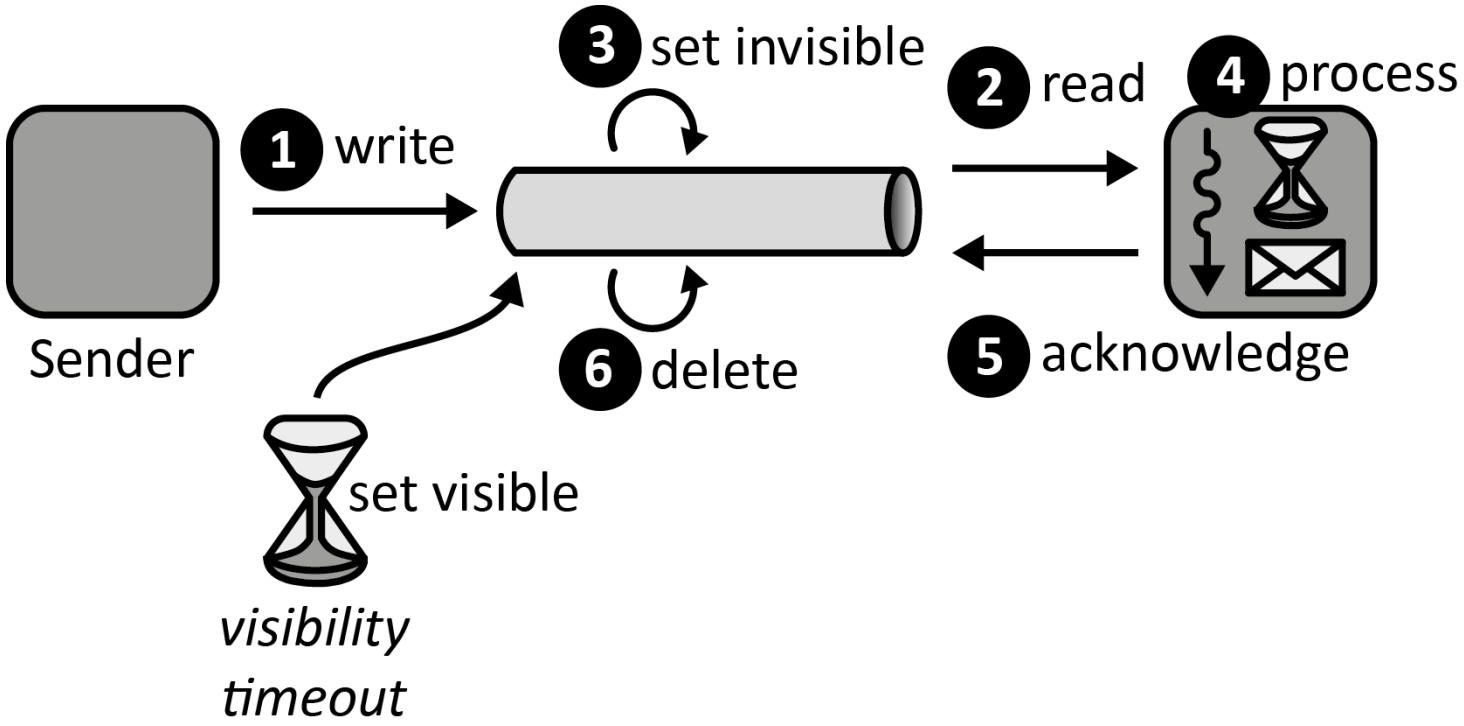




Timeout-based Message Processor

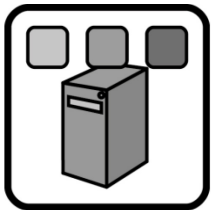
Clients acknowledge message processing to ensure that messages are handled. If a message is not acknowledged after a certain timeout, it is processed again.

How can an application guarantee that all handled messages are processed at-least-once?

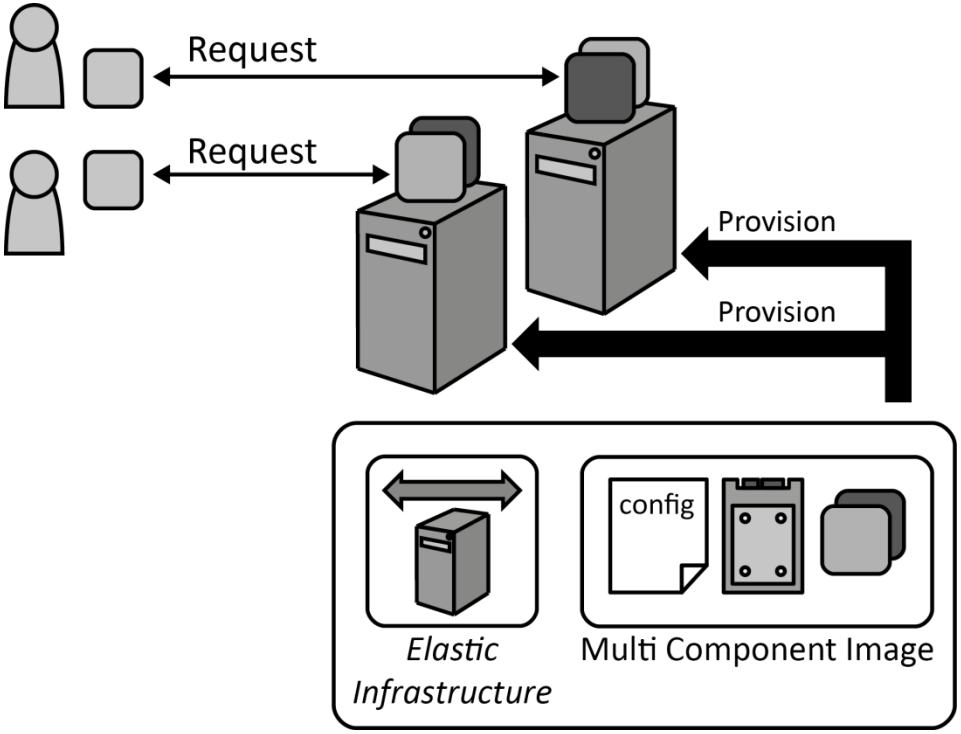


Multi-component image

Virtual servers host multiple application components that may not be active at all times to reduce provisioning and decommissioning operations.



How can a virtual server provide the functionality of multiple application components to be used flexibly in applications?

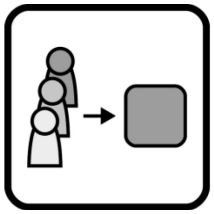


Cloud Application Architecture Patterns
Multi-Tenancy

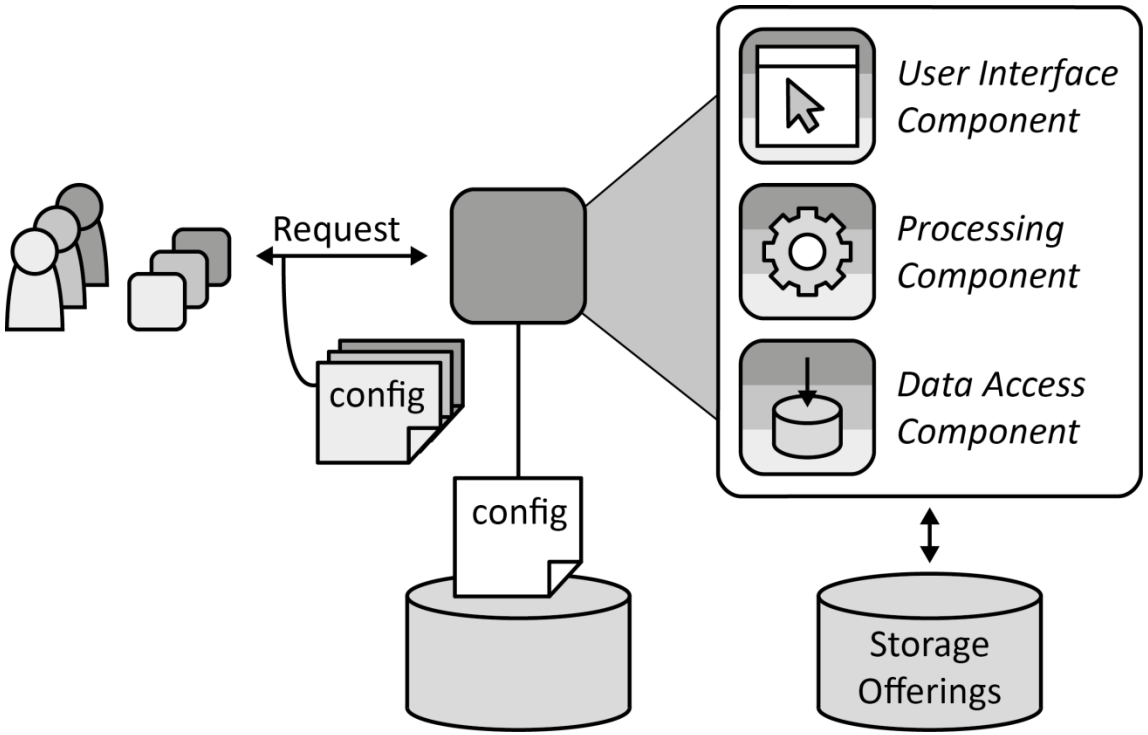
http://www.cloudcomputingpatterns.org

Shared Component

A component is accessed by multiple tenants to leverage economies of scale.

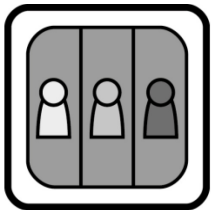


How can an application component be shared between multiple tenants enabling some individual configuration?

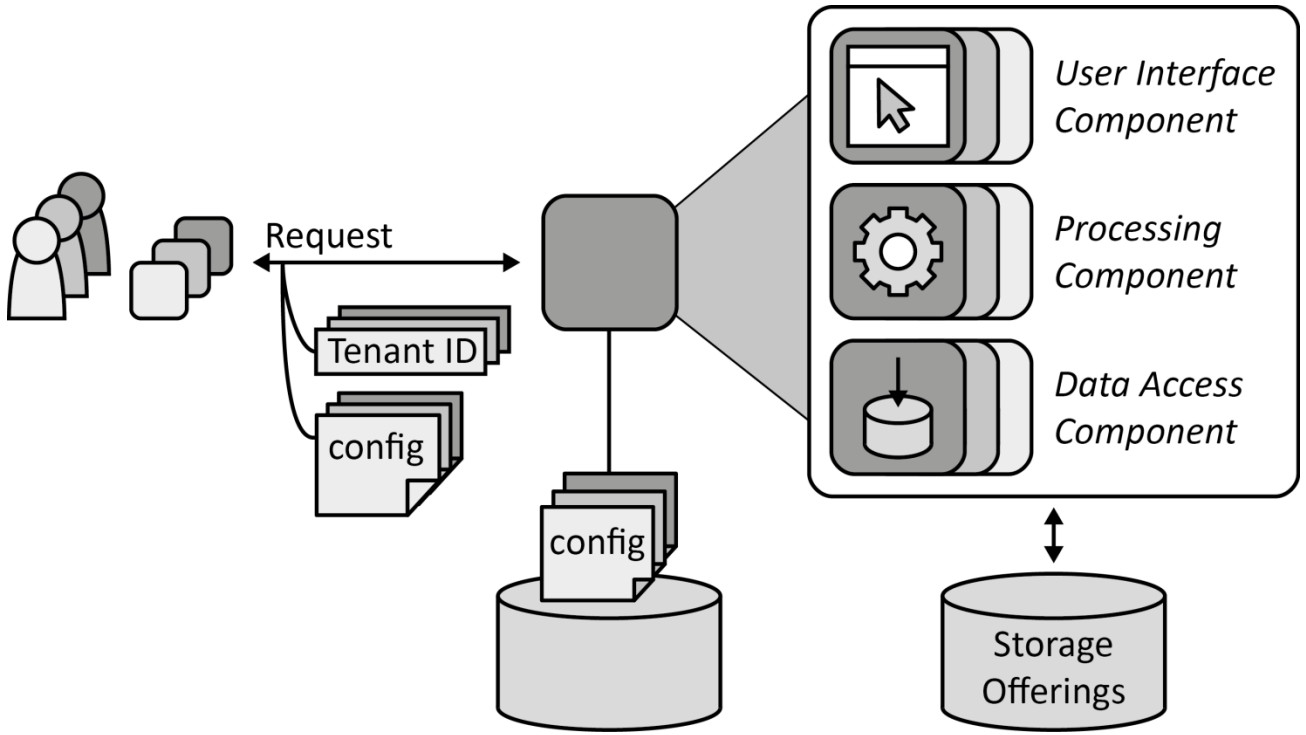


Tenant-isolated Component

A shared component avoids influences between tenants regarding assured performance, available storage capacity, and accessibility of functionality and data.

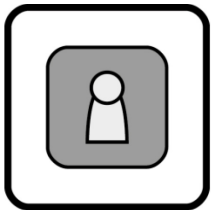


How can an application component be shared while enabling individual configuration and tenant-isolation regarding performance, data volume, and access privileges?

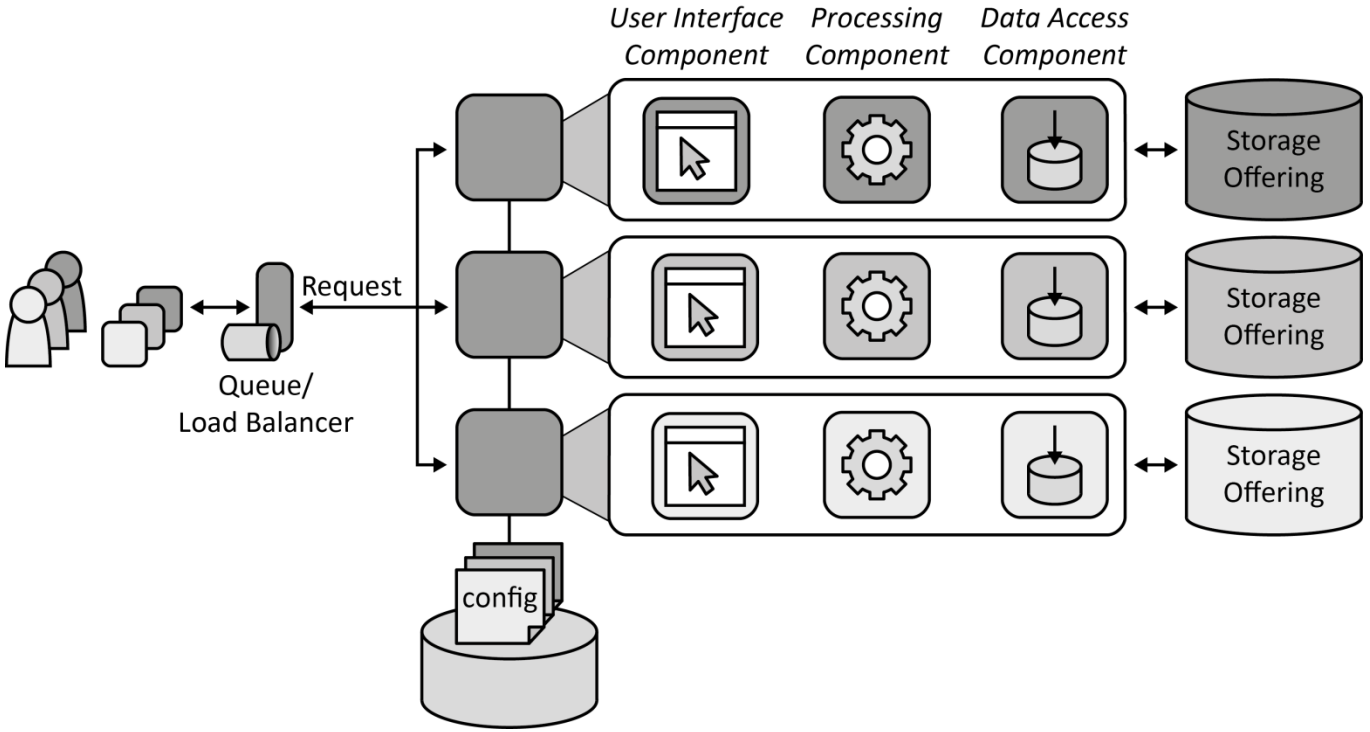


Dedicated Component

Components providing critical functionality are provided exclusively to tenants while still allowing other components to be shared between tenants.



How can application components that cannot be shared be integrated into a multi-tenant application?



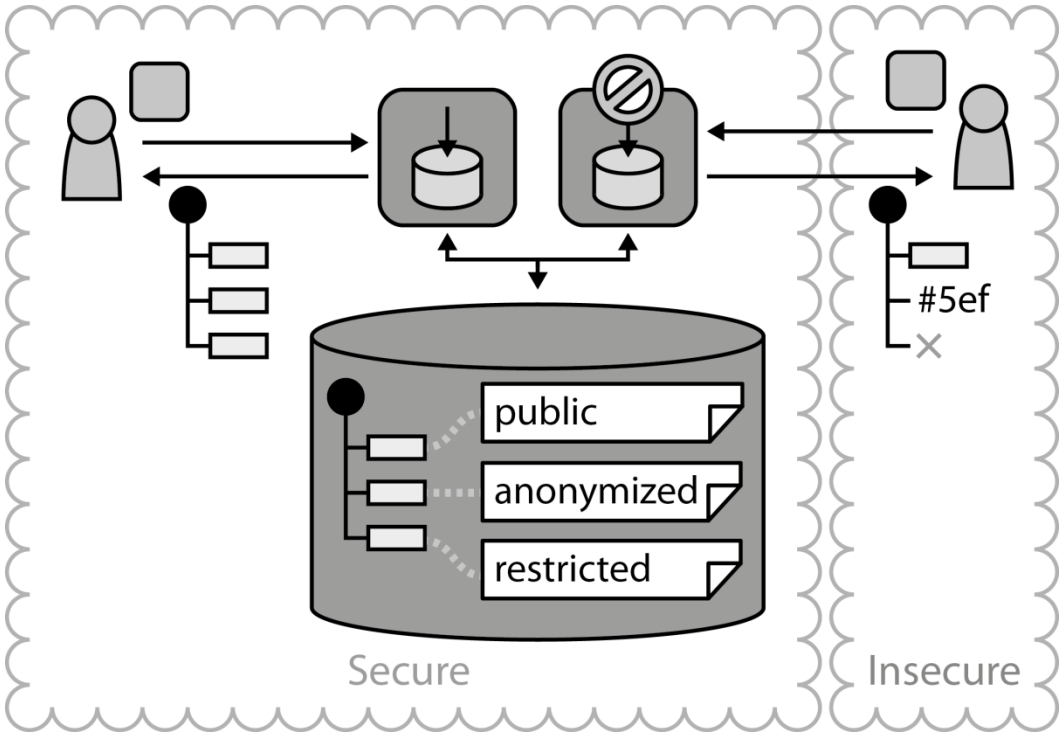
Cloud Application Architecture Patterns
Cloud Integration

http://www.cloudcomputingpatterns.org

Restricted Data Access Component

Data provided to clients from different environments is adjusted based on access restrictions.

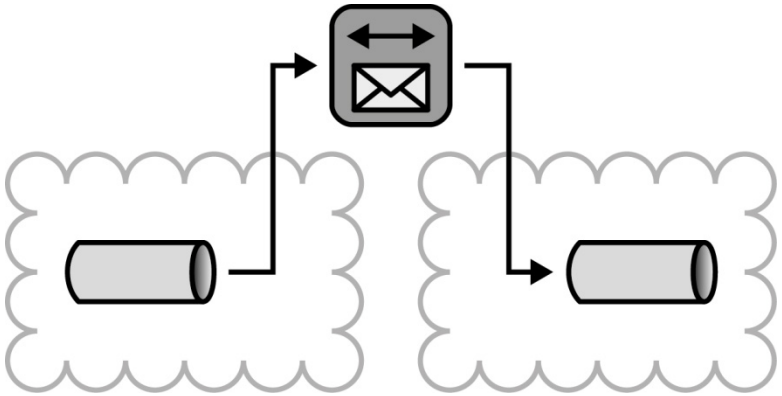
How can an application component alter provided data based on access restrictions imposed on different environments?



Message Mover

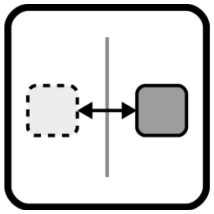
Messages are moved automatically between different cloud providers to provide unified access to application components using messaging.

How can message queues of different providers be integrated without an impact on the application components using them?

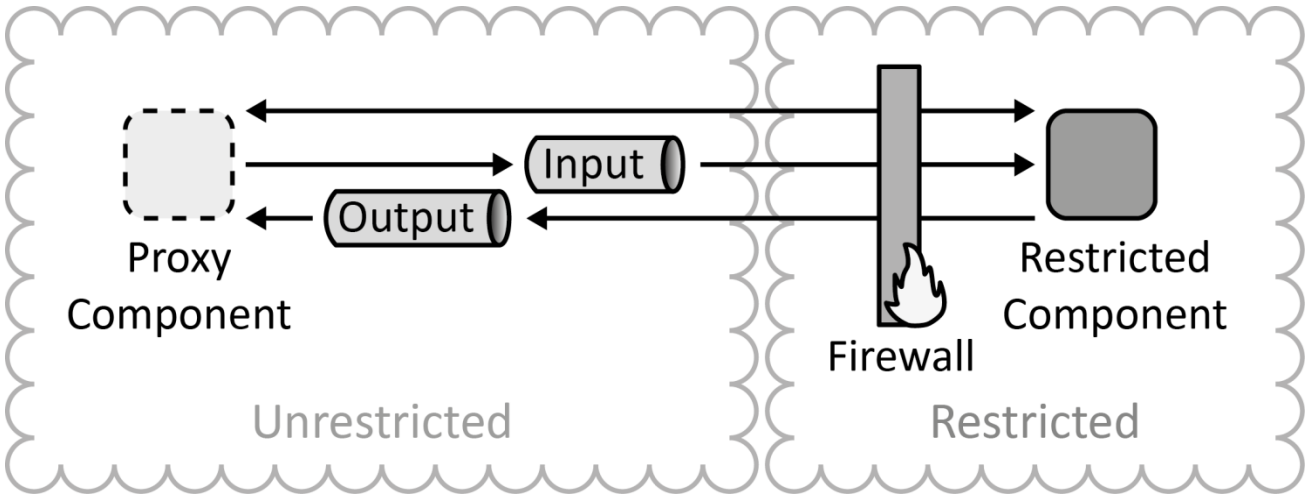


Application Component Proxy

An application component is made available in an environment from where it cannot be accessed directly by deploying an application component proxy.

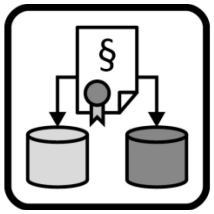


How can an application component be accessed if direct access to its hosting environment is restricted?

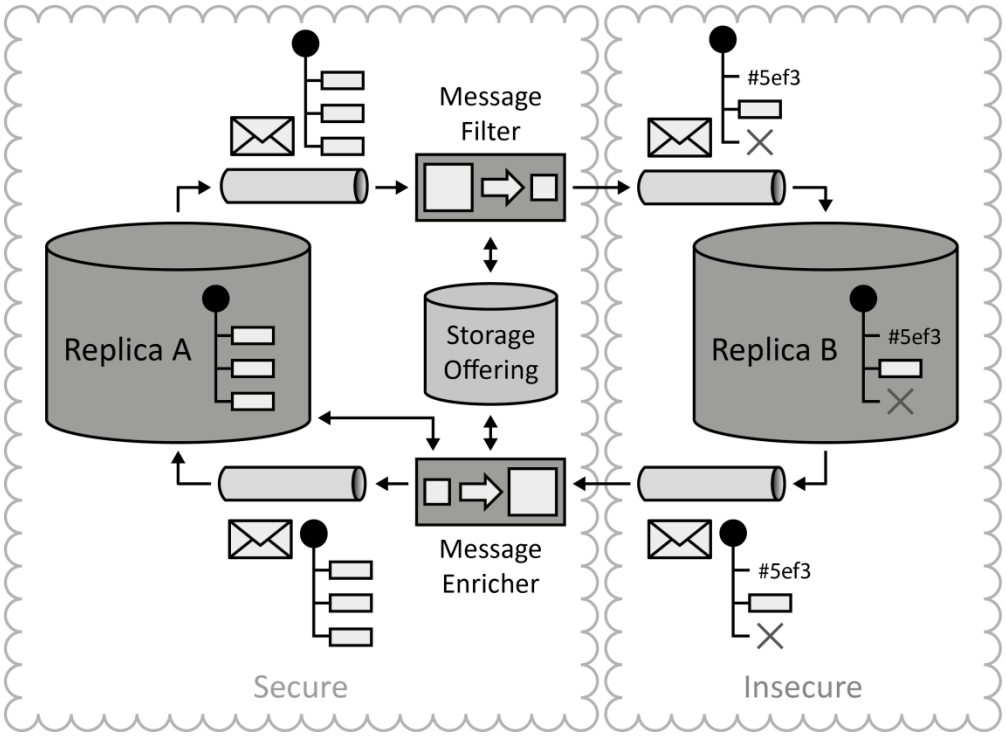


Compliant Data Replication

Data is replicated among multiple environments. Data is automatically obfuscated and deleted to meet laws and security regulations.

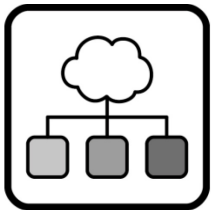


How can data be replicated between environments if some environments may only handle subsets of the data due to laws and corporate regulations?

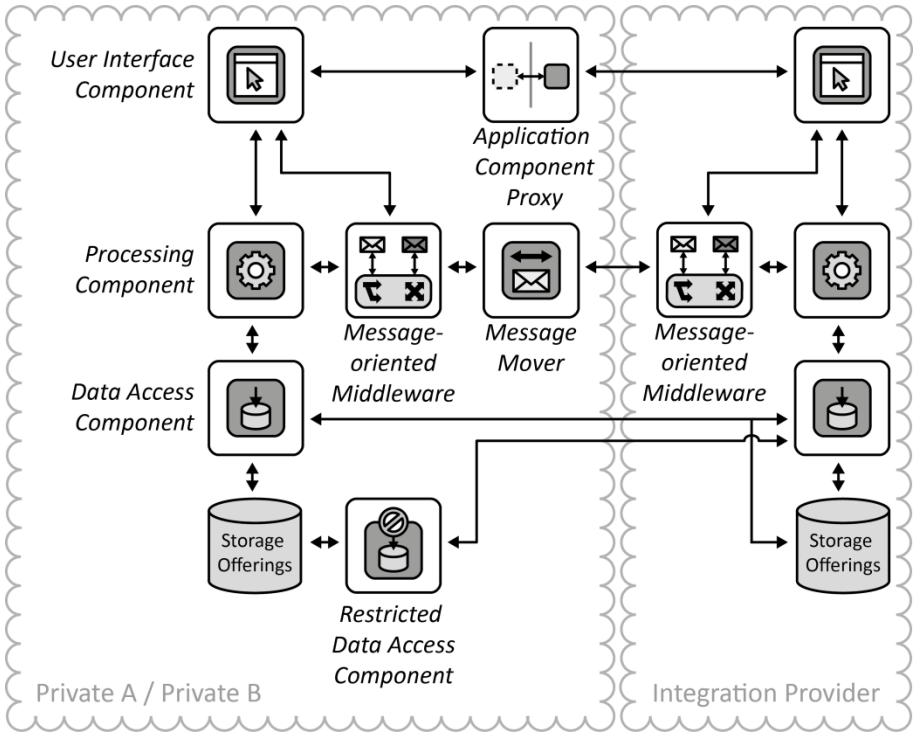


Integration Provider

Integration functionality such as messaging and shared data is hosted by a separate provider to enable integrate of otherwise separated hosting environments.



How can application components that reside in different environments, possibly belonging to different companies, be integrated through a third-party provider?

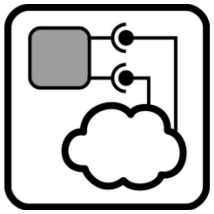


Cloud Application Management Patterns
Management Components

http://www.cloudcomputingpatterns.org

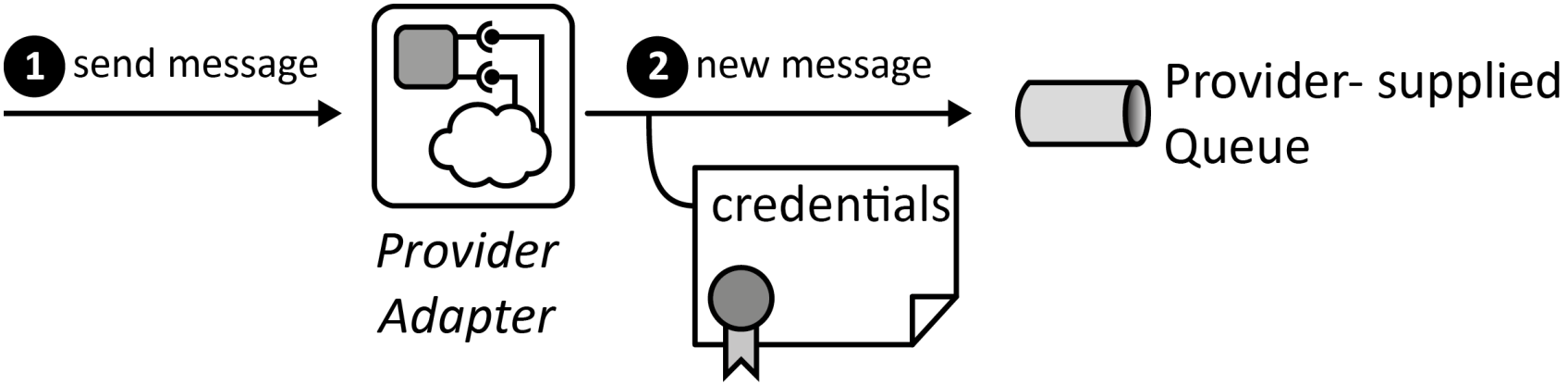
Provider Adapter

Provider interfaces are encapsulated and mapped to unified interfaces to separate concerns of interactions with the provider from application functionality.



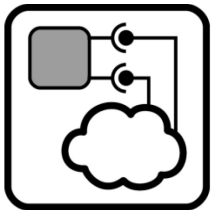
How can the dependencies of an application component on a provider-specific interface be managed?

Synchronous to Asynchronous Adapter



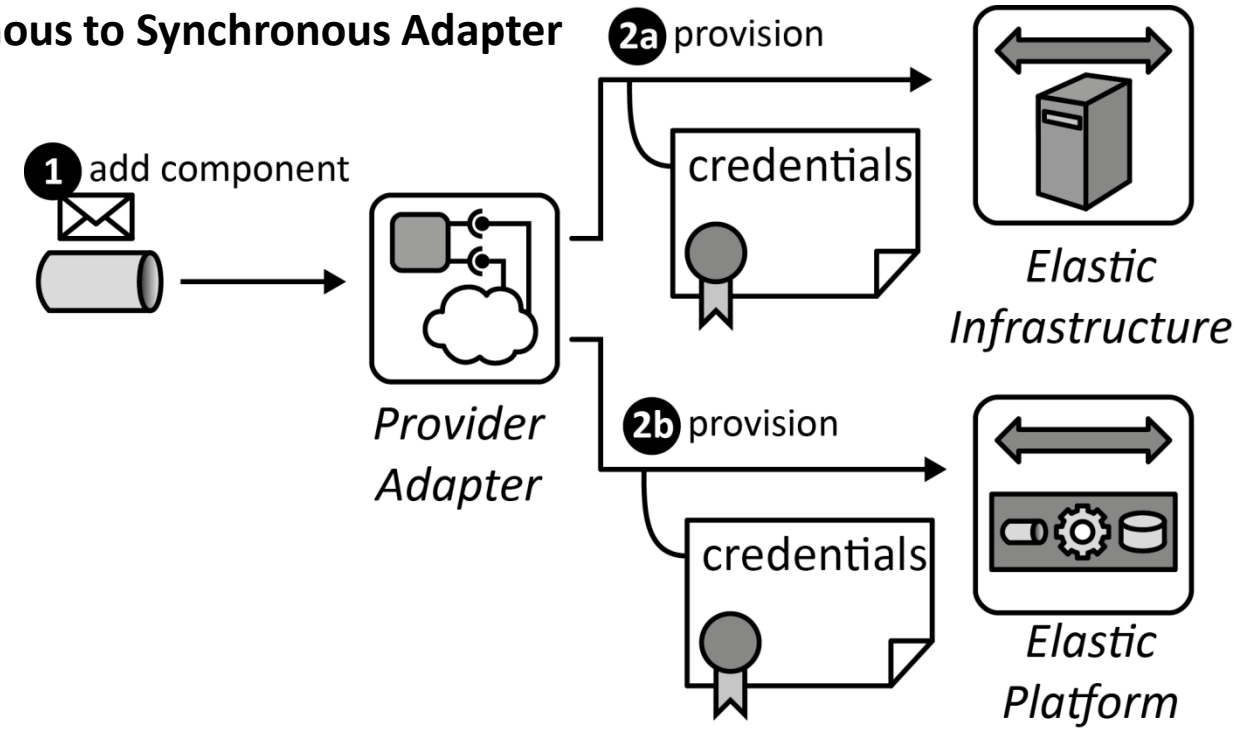
Provider Adapter

Provider interfaces are encapsulated and mapped to unified interfaces to separate concerns of interactions with the provider from application functionality.



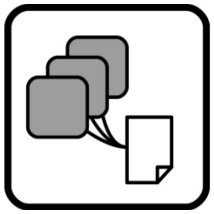
How can the dependencies of an application component on a provider-specific interface be managed?

Asynchronous to Synchronous Adapter



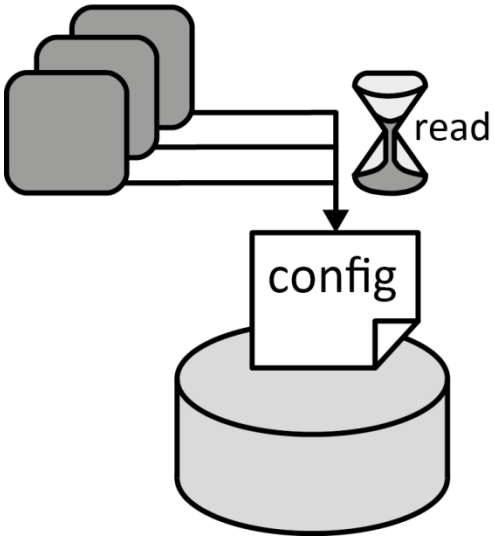
Managed Configuration

Scaled-out application components use a centrally stored configuration to provide a unified behavior that can be adjusted simultaneously.

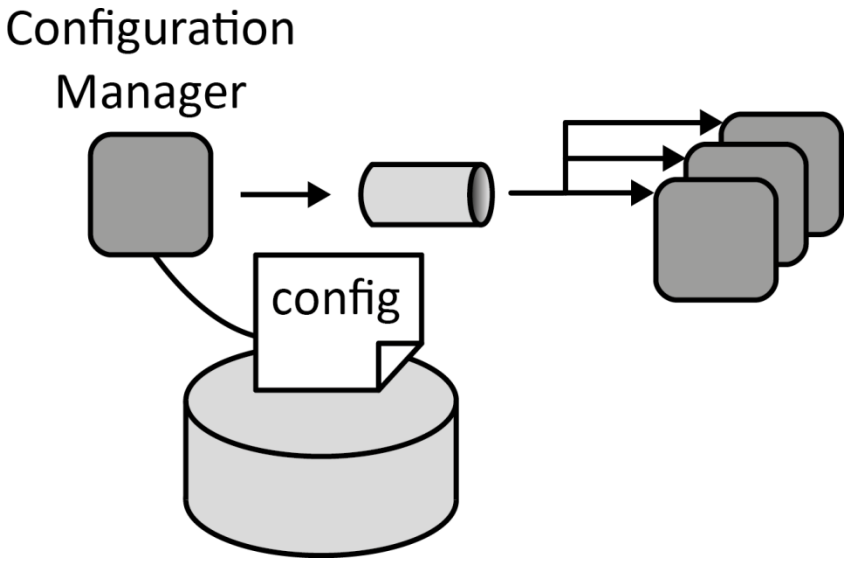


How can the configuration of scaled out application component instances be controlled in a coordinated fashion?

PULL:

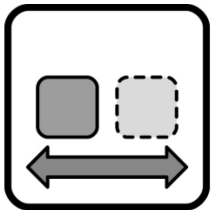


PUSH:

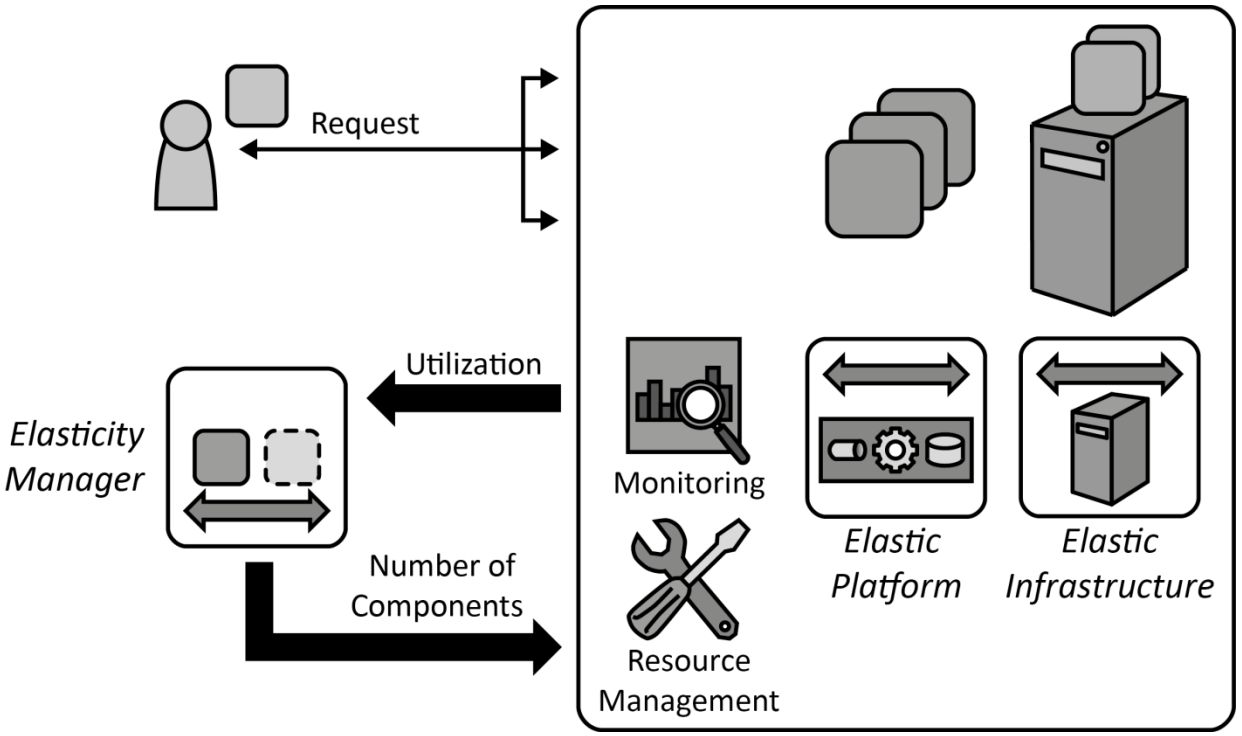


Elasticity Manager

The utilization of IT resources on which an application is hosted is used to determine the number of required application component instances.

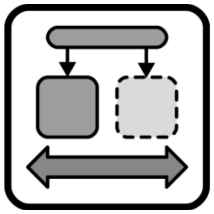


How can the number of required application component instances be determined based on the utilization of hosting IT resources?

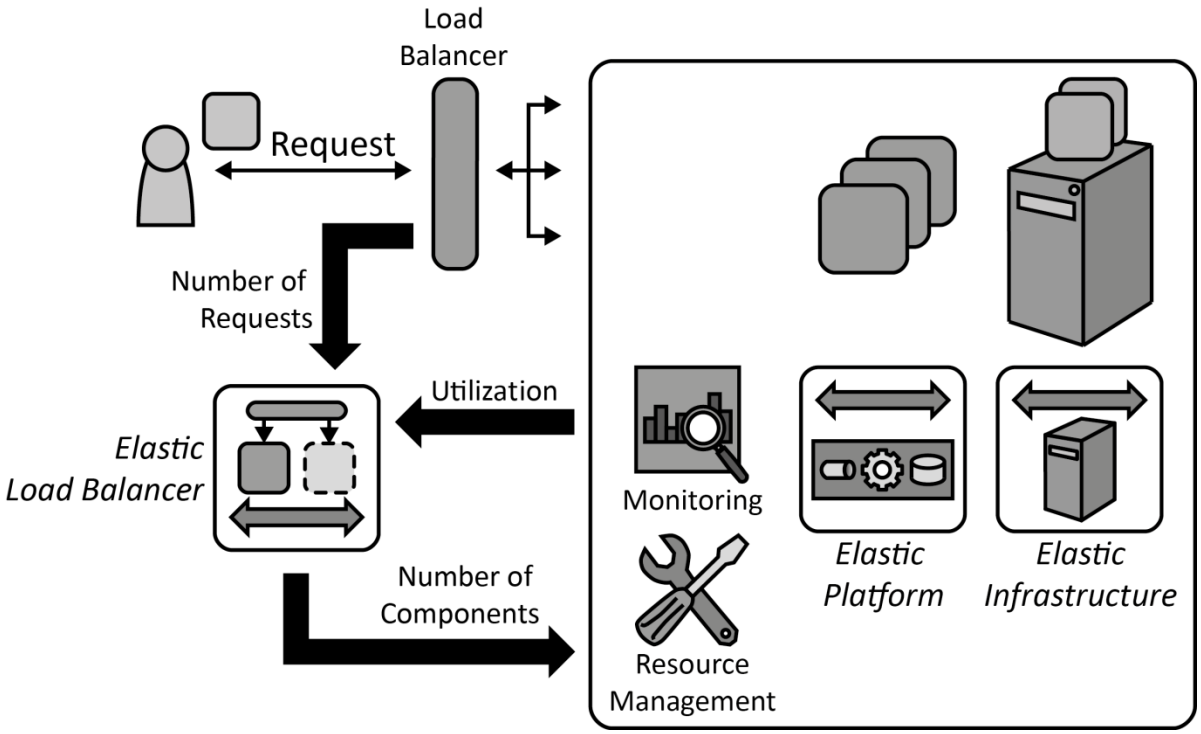


Elastic Load Balancer

The number of synchronous accesses to an elastically scaled-out application is used to determine the number of required application component instances.

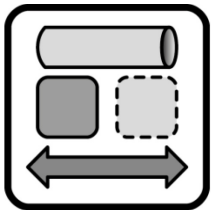


How can the number of required application component instances be determined based on monitored synchronous accesses?

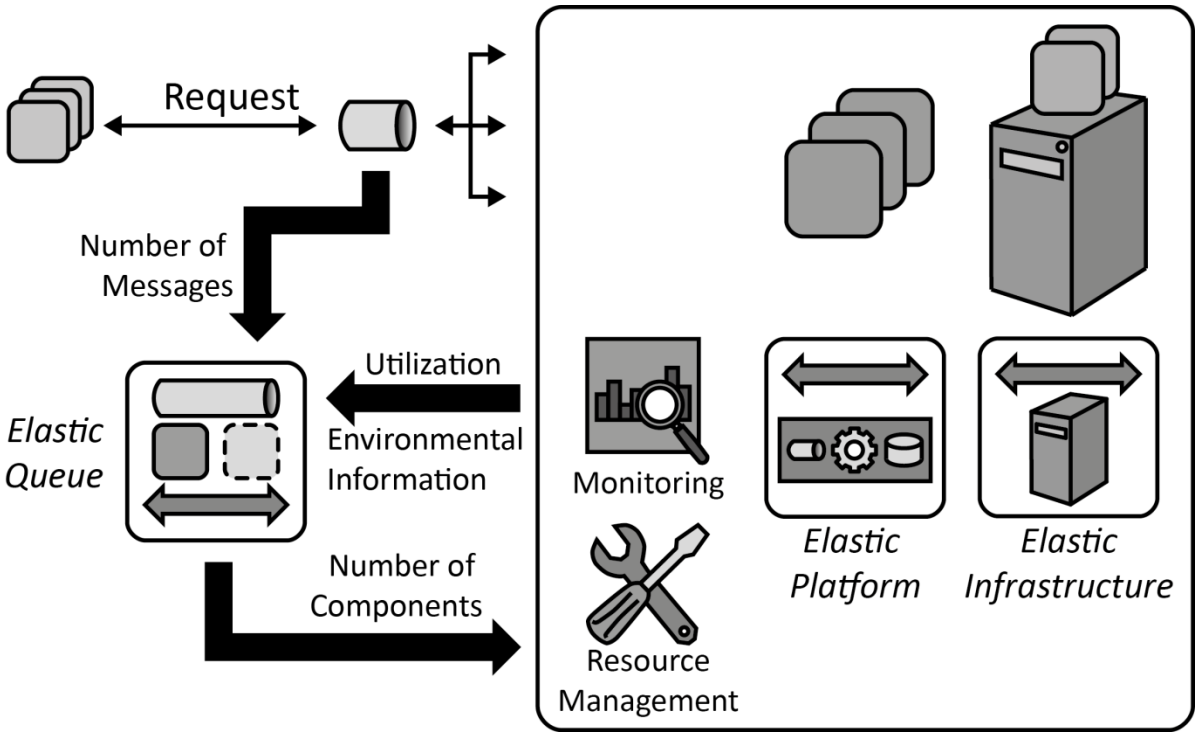


Elastic Queue

The number of accesses via messaging is used to adjust the number of required application component instances.



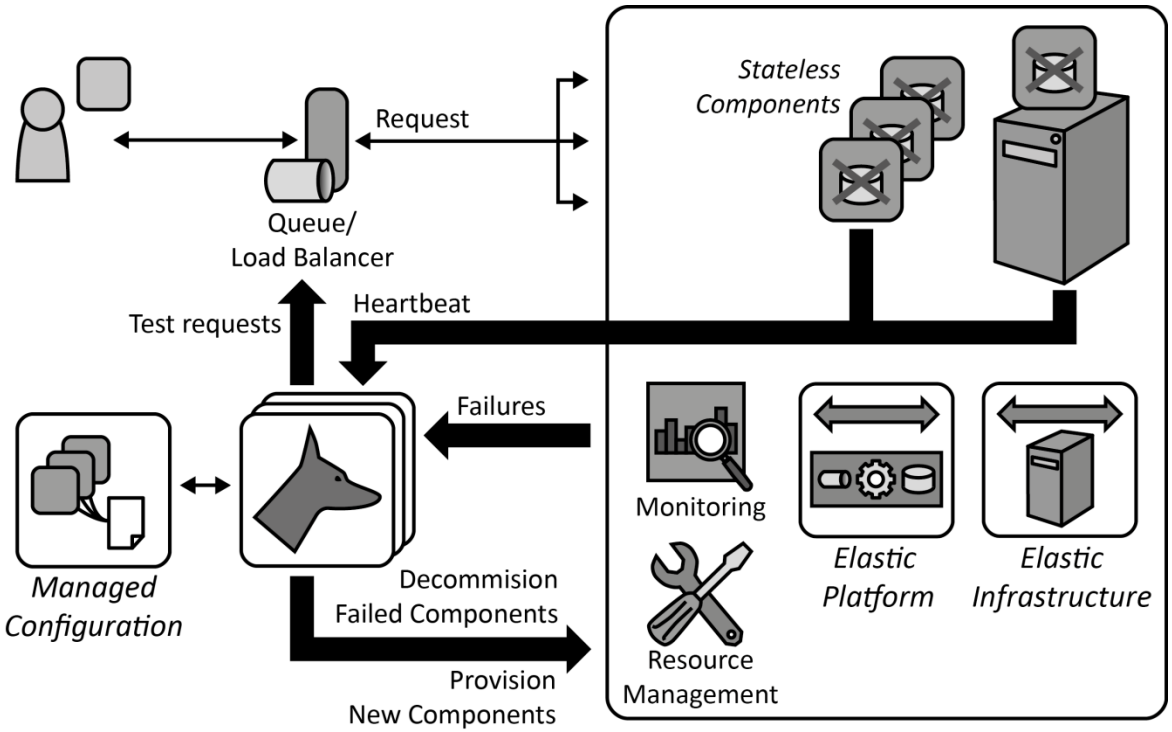
How can the number of required application component instances be adjusted based on monitored asynchronous accesses?



Watchdog

Applications cope with failures automatically by monitoring and replacing application component instances if the provider-assured availability is insufficient.

How can applications automatically detect failing application components and handle their replacement?

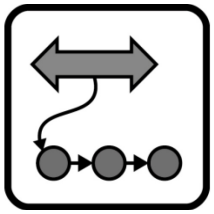


Cloud Application Management Patterns
Management Processes

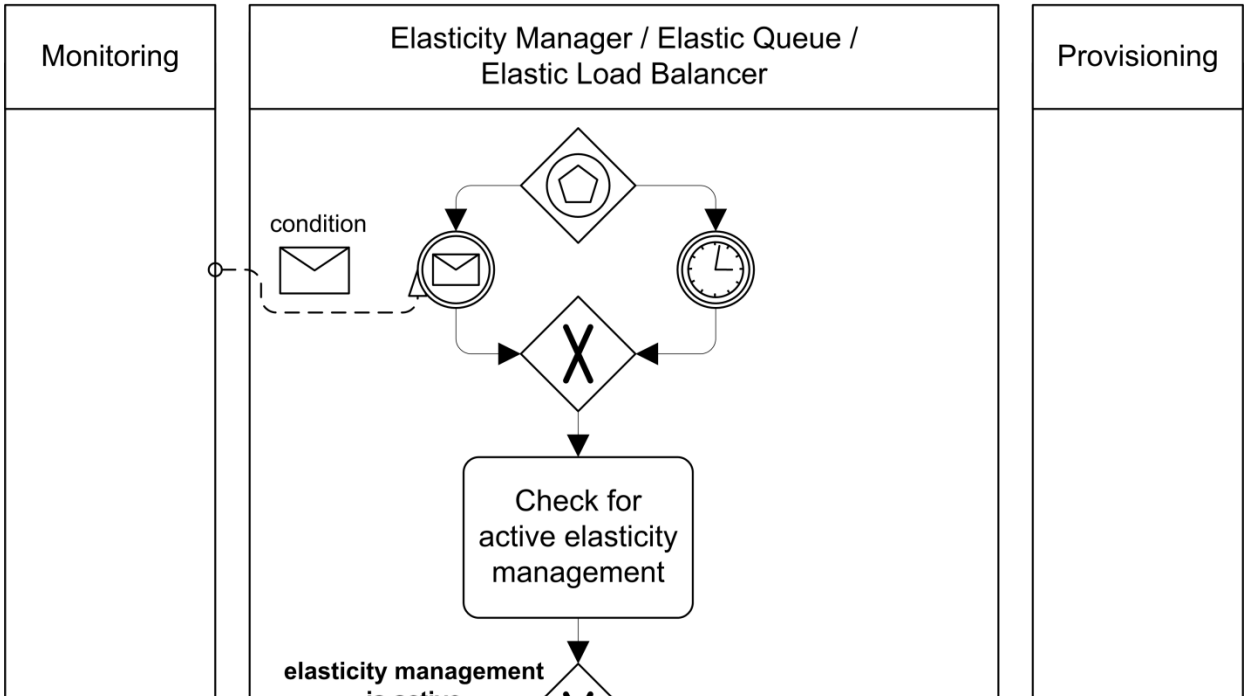
http://www.cloudcomputingpatterns.org

Elasticity Management Process

Application component instances are added and removed automatically to cope with increasing or decreasing workload, respectively.

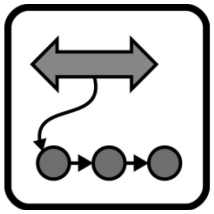


How can the number of resources to which application components are scaled-out be adjusted efficiently to the experienced workload and anticipated future workload?

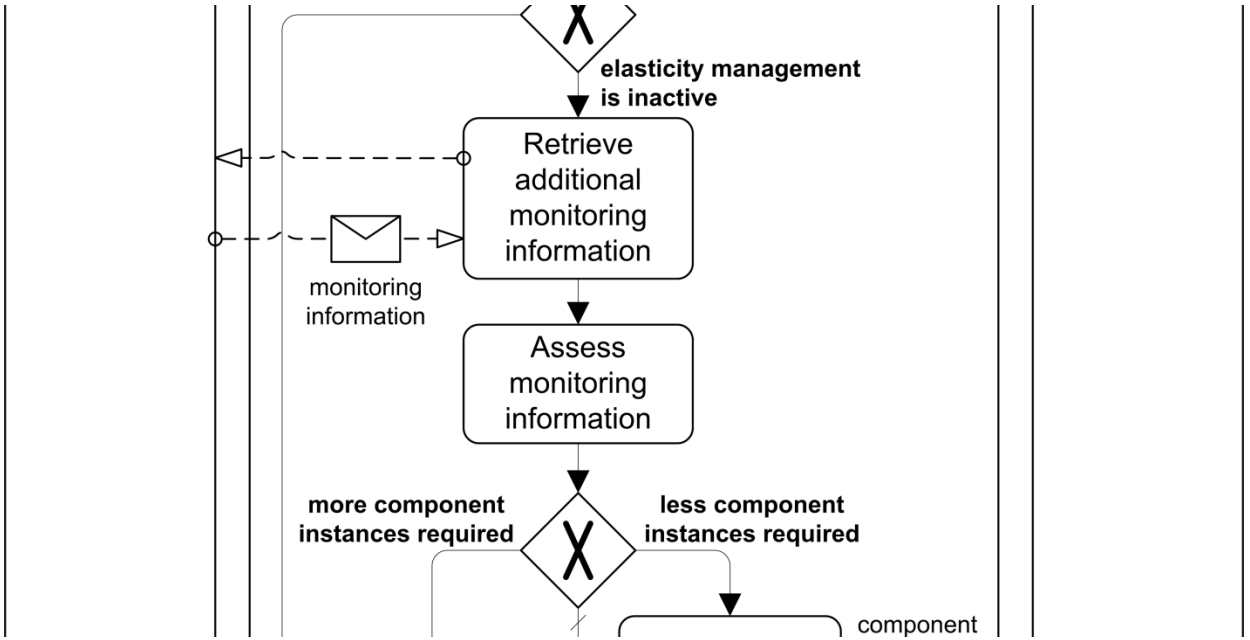


Elasticity Management Process

Application component instances are added and removed automatically to cope with increasing or decreasing workload, respectively.

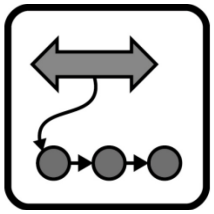


How can the number of resources to which application components are scaled-out be adjusted efficiently to the experienced workload and anticipated future workload?

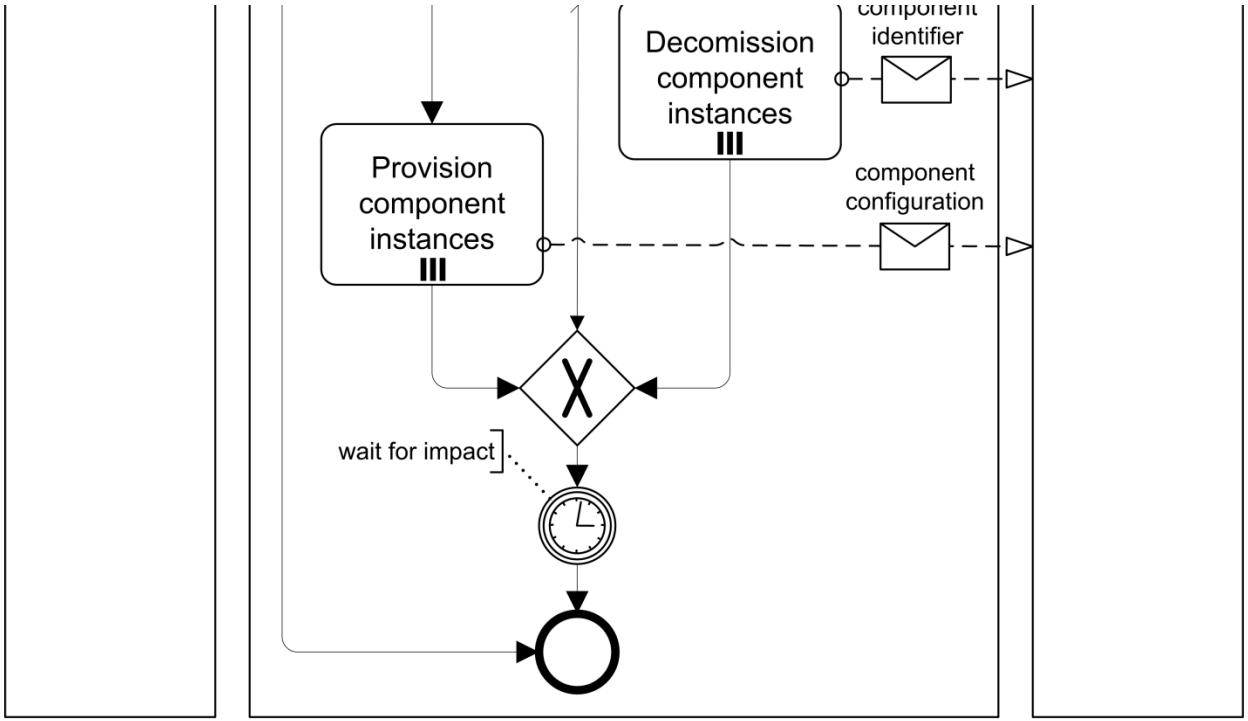


Elasticity Management Process

Application component instances are added and removed automatically to cope with increasing or decreasing workload, respectively.

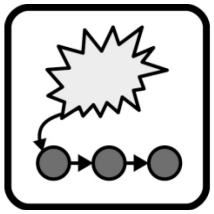


How can the number of resources to which application components are scaled-out be adjusted efficiently to the experienced workload and anticipated future workload?

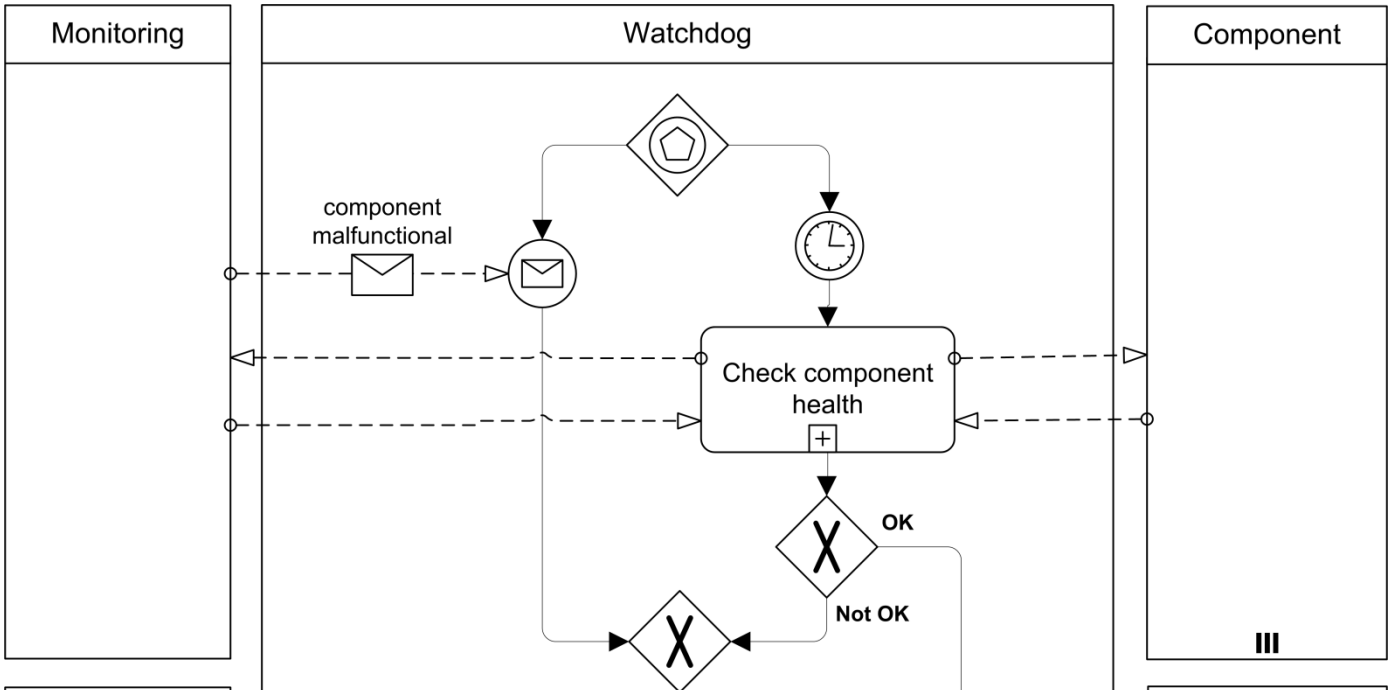


Resiliency Management Process

Application components are checked for failures and replaced automatically without human intervention.

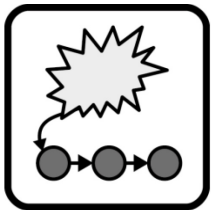


How can the overall availability of an application be ensured automatically even if individual application component instances fail?

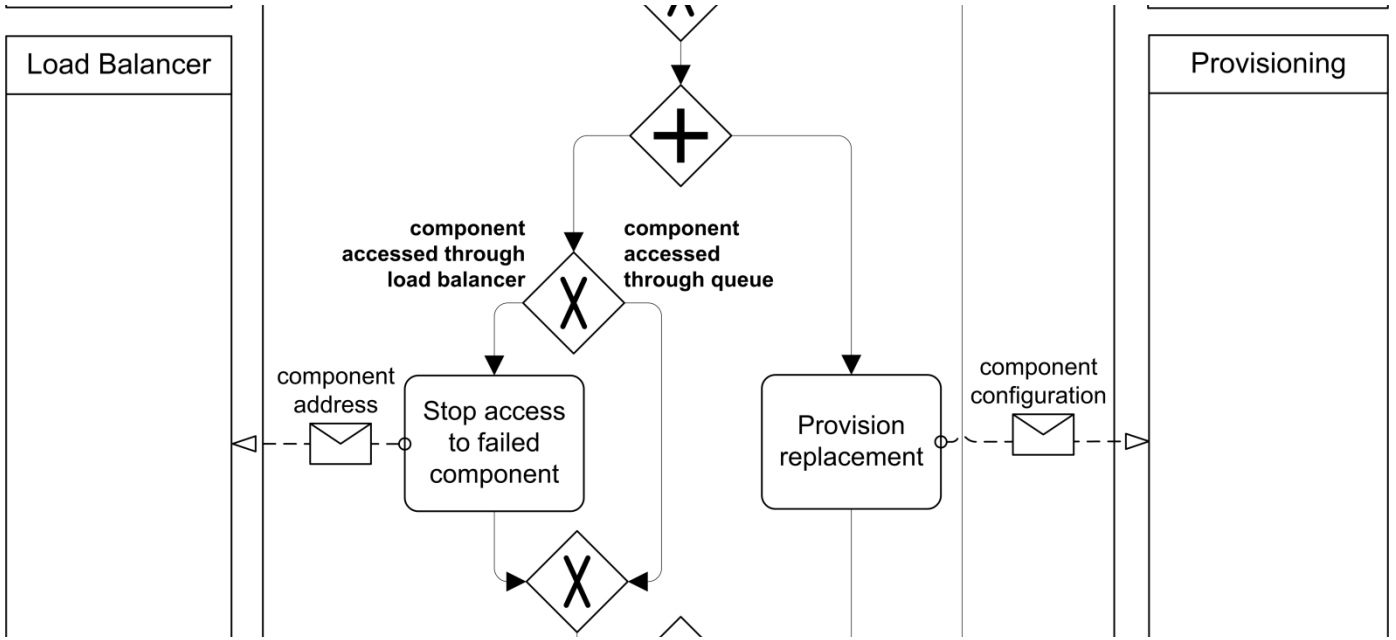


Resiliency Management Process

Application components are checked for failures and replaced automatically without human intervention.

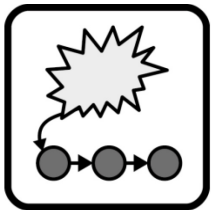


How can the overall availability of an application be ensured automatically even if individual application component instances fail?

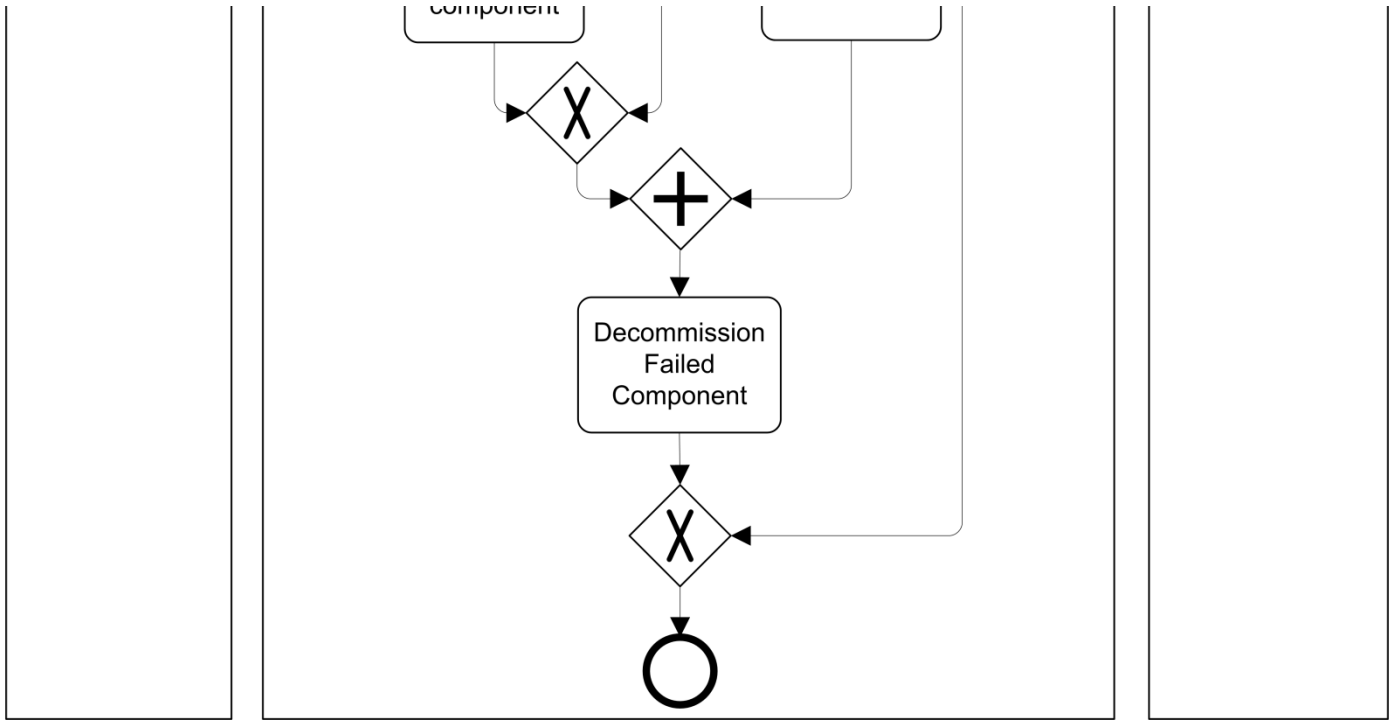


Resiliency Management Process

Application components are checked for failures and replaced automatically without human intervention.

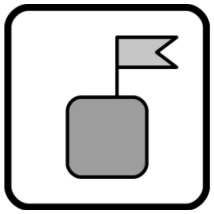


How can the overall availability of an application be ensured automatically even if individual application component instances fail?

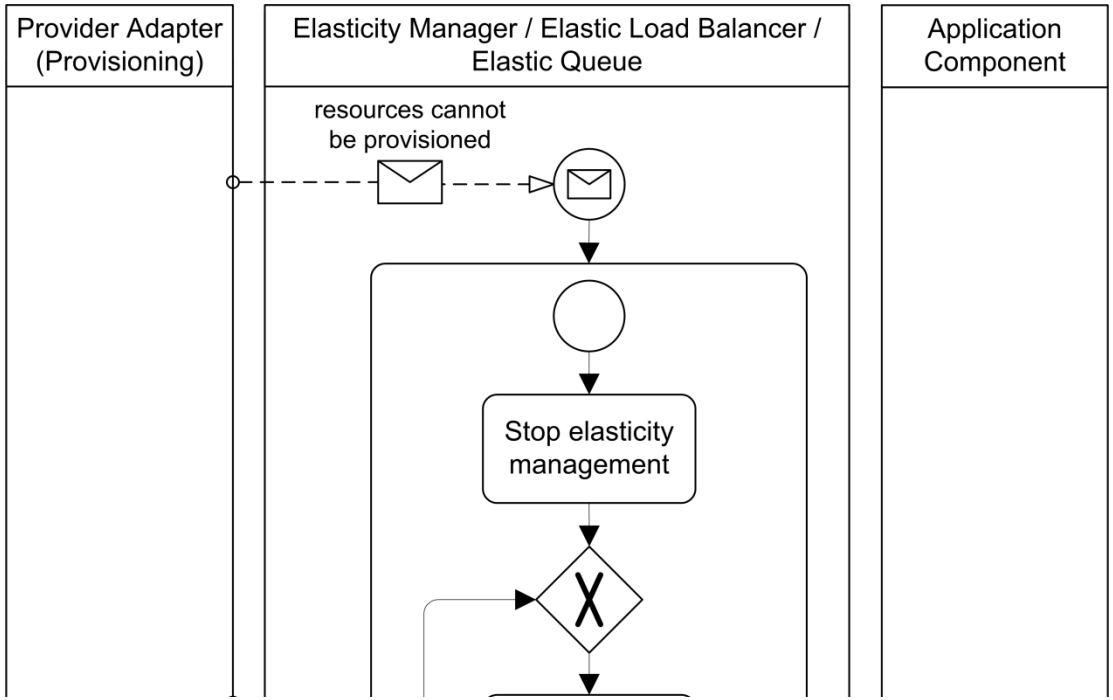


Feature Flag Management Process

If the cloud cannot provide required resources in time, application features by are degraded gracefully in order to keep vital features operational.

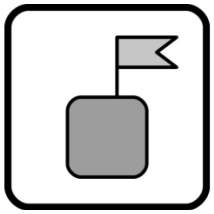


How can the performance of an application degrade gracefully, if the experienced workload increases but additional cloud resources are unavailable or take too long to provision?

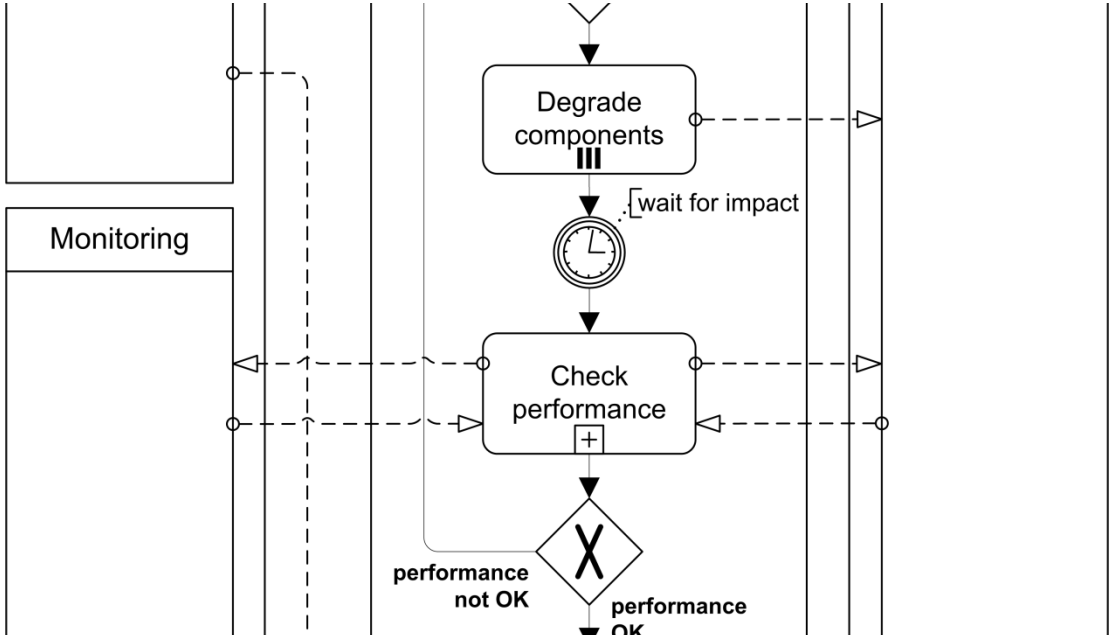


Feature Flag Management Process

If the cloud cannot provide required resources in time, application features by are degraded gracefully in order to keep vital features operational.

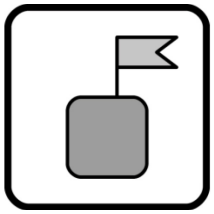


How can the performance of an application degrade gracefully, if the experienced workload increases but additional cloud resources are unavailable or take too long to provision?

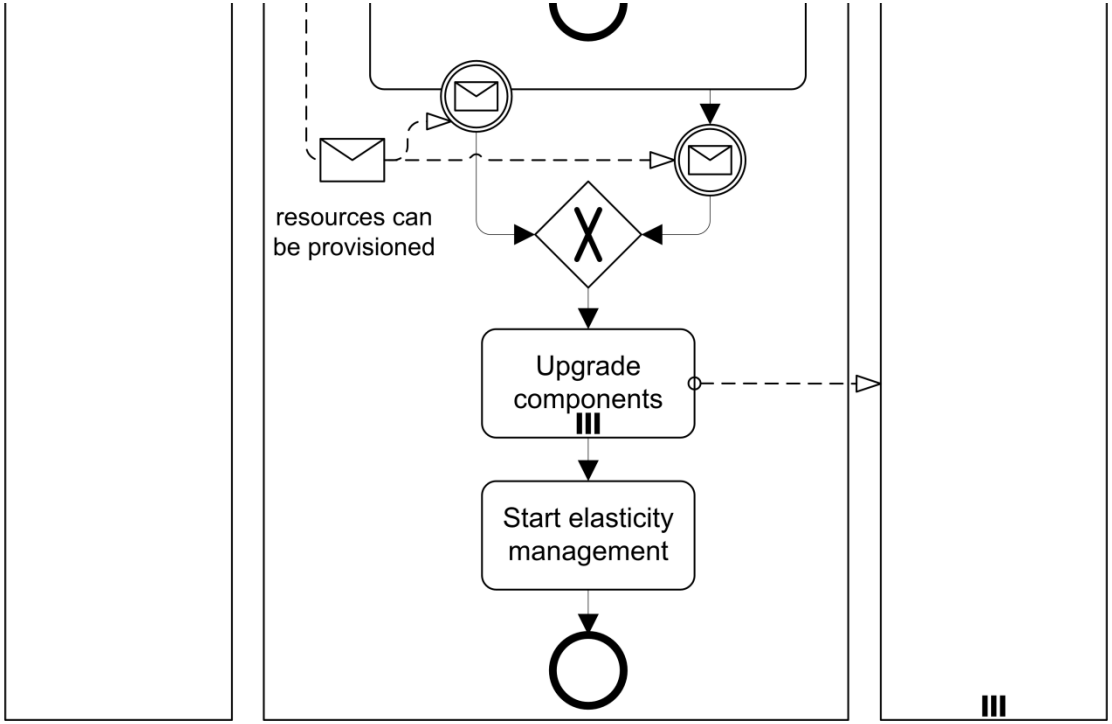


Feature Flag Management Process

If the cloud cannot provide required resources in time, application features by are degraded gracefully in order to keep vital features operational.

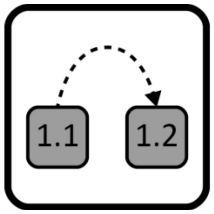


How can the performance of an application degrade gracefully, if the experienced workload increases but additional cloud resources are unavailable or take too long to provision?

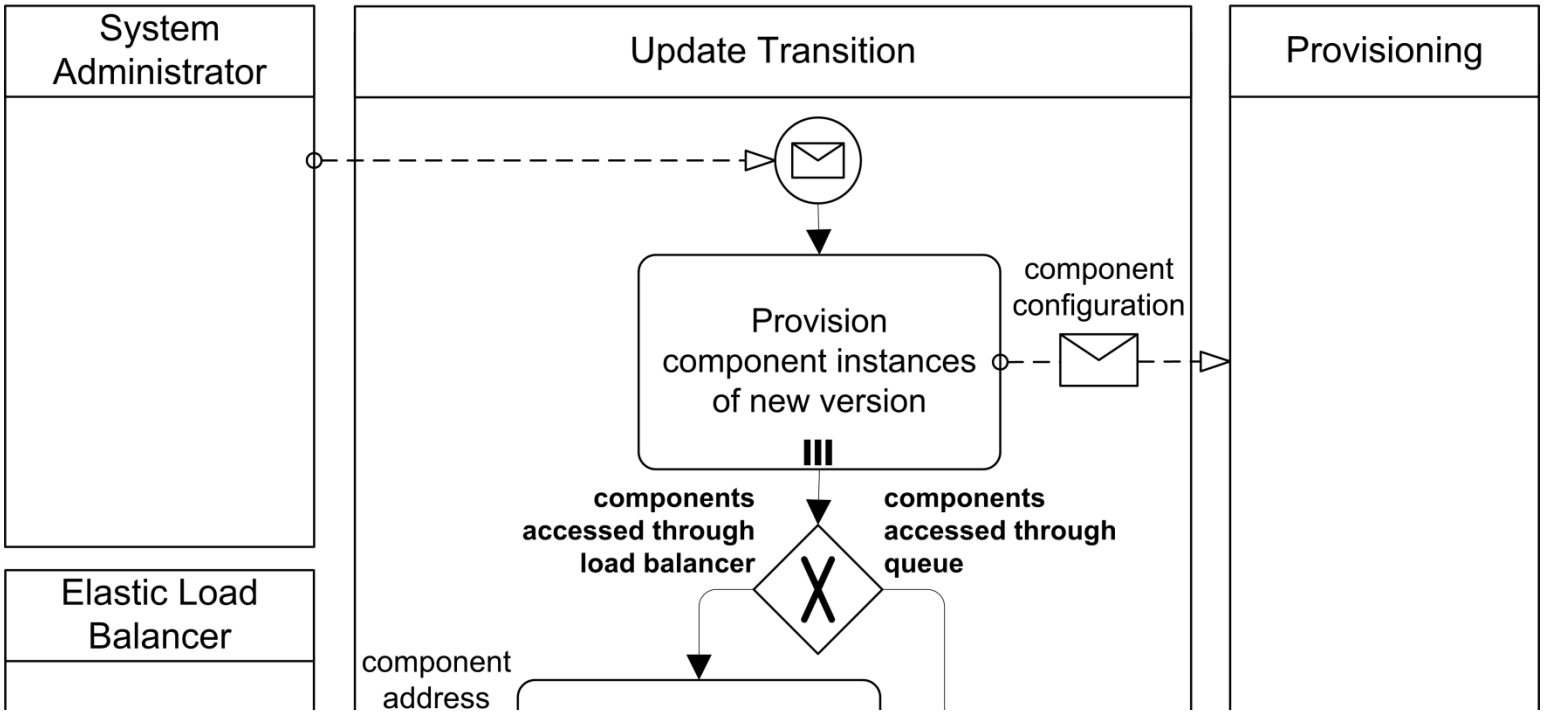


Update Transition Process

When a new application component version, middleware versions etc. become available, running application components are updated seamlessly.

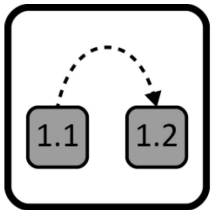


How can application components of a distributed application be updated seamlessly?

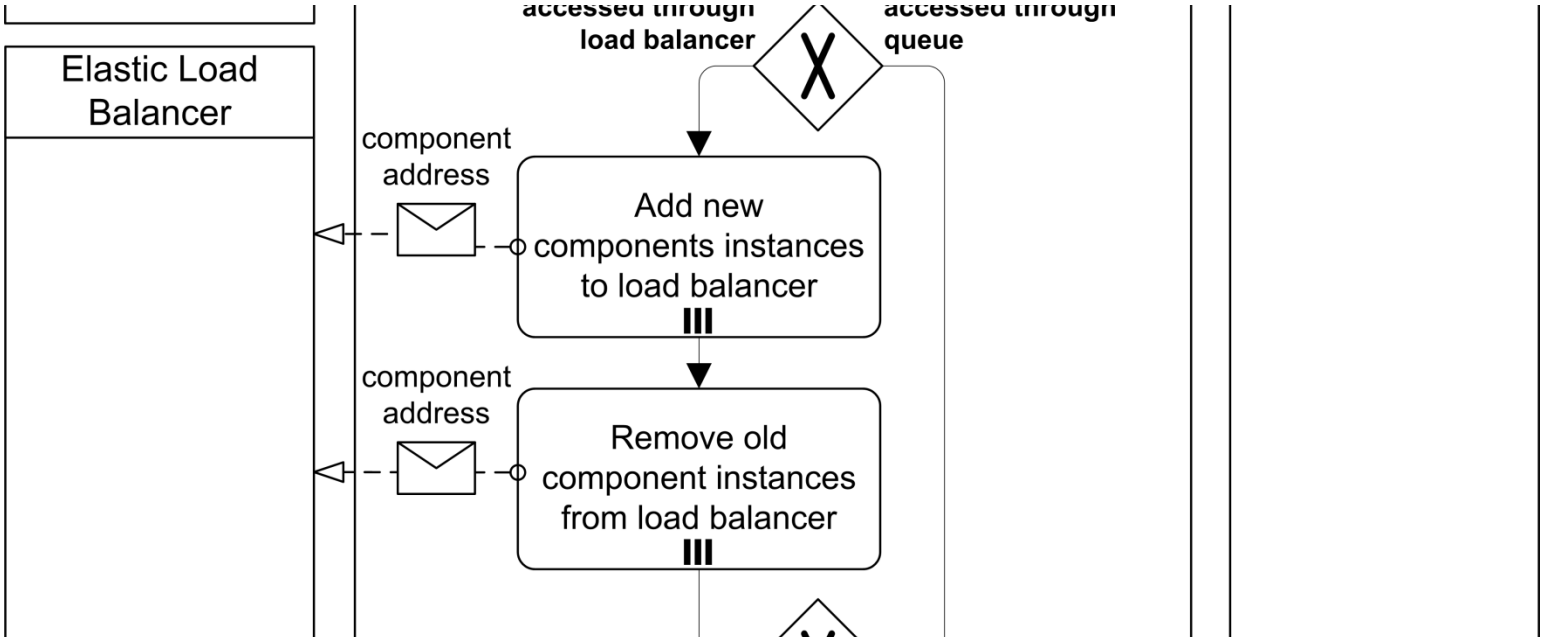


Update Transition Process

When a new application component version, middleware versions etc. become available, running application components are updated seamlessly.

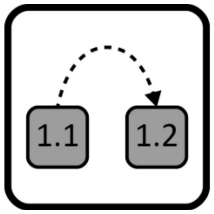


How can application components of a distributed application be updated seamlessly?

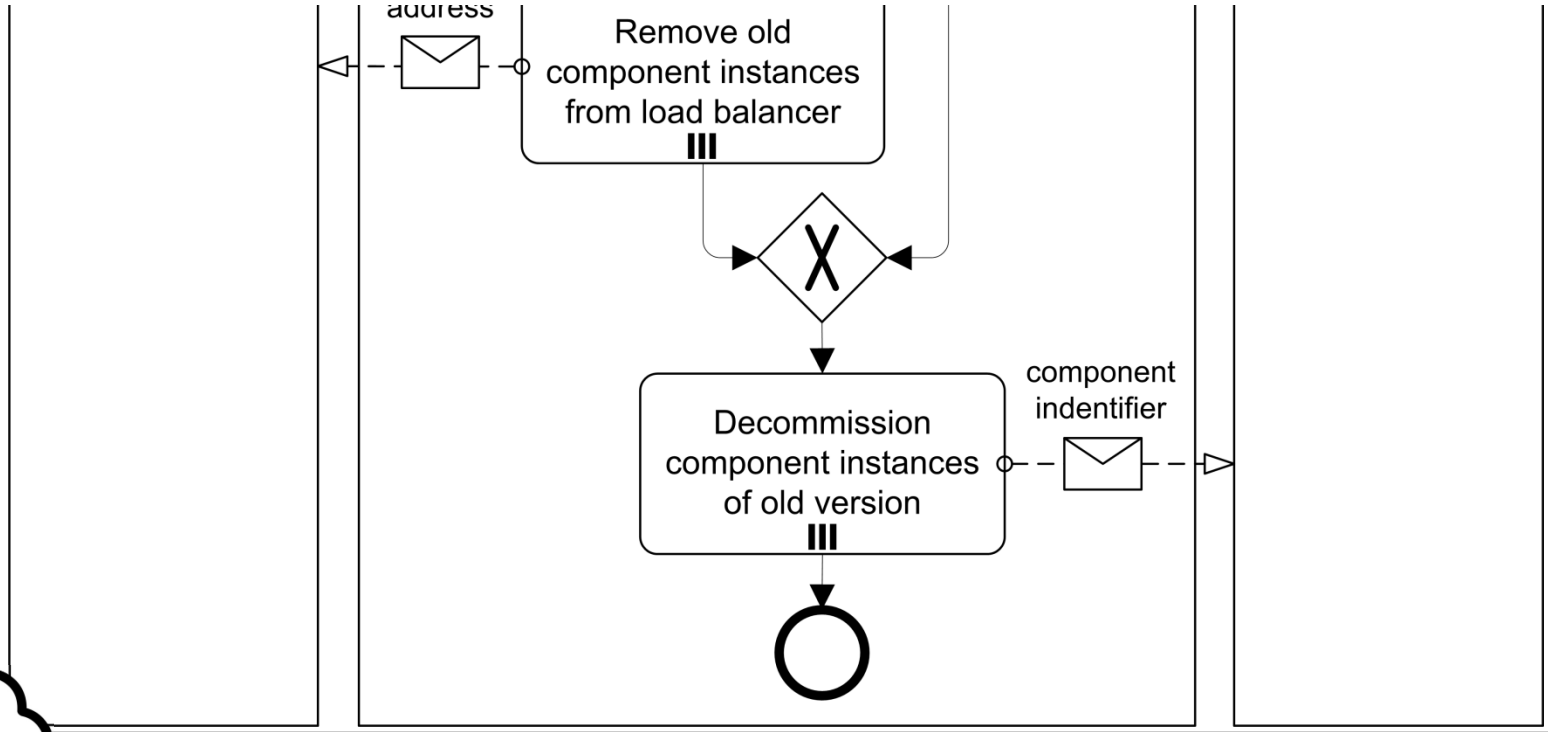


Update Transition Process

When a new application component version, middleware versions etc. become available, running application components are updated seamlessly.



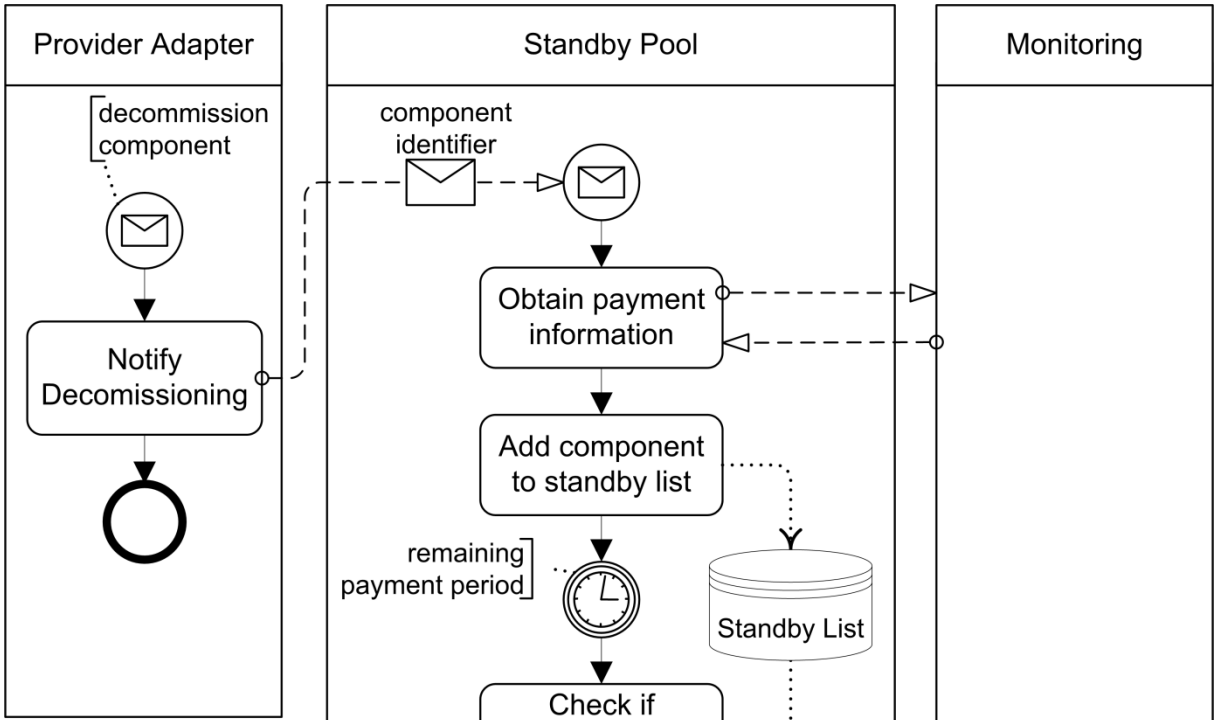
How can application components of a distributed application be updated seamlessly?



Standby Pooling Process

Application component instances are kept on standby to increase provisioning speed and utilize billing time-slots efficiently.

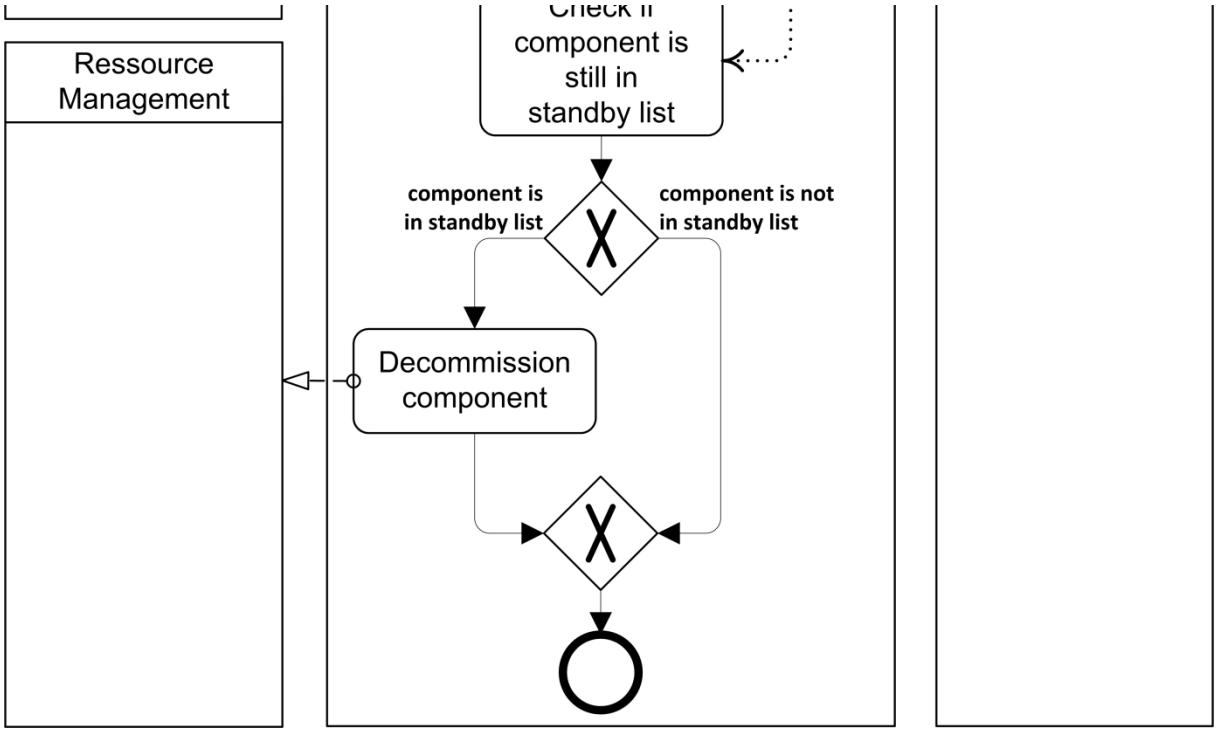
How can required provisioning times for application component instances be ensured while utilizing pay-per-use resources in an optimal fashion?



Standby Pooling Process

Application component instances are kept on standby to increase provisioning speed and utilize billing time-slots efficiently.

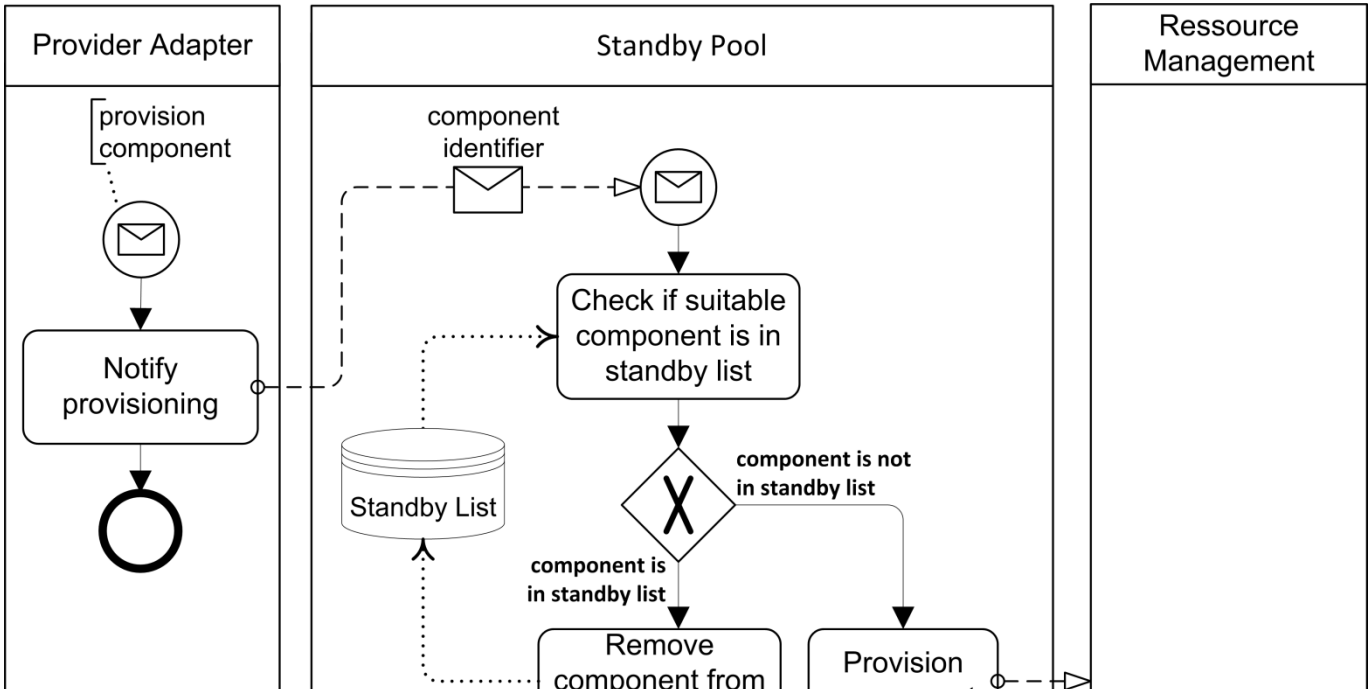
How can required provisioning times for application component instances be ensured while utilizing pay-per-use resources in an optimal fashion?



Standby Pooling Process

Application component instances are kept on standby to increase provisioning speed and utilize billing time-slots efficiently.

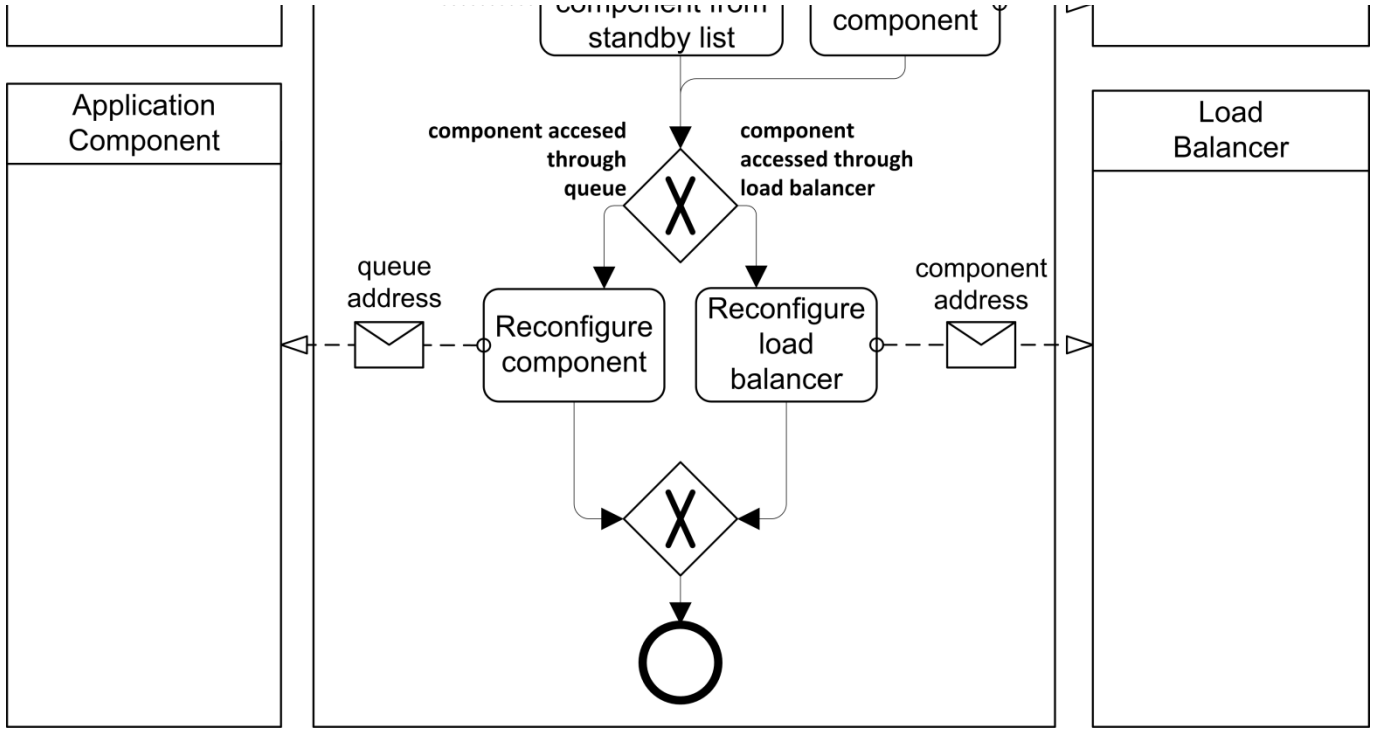
How can required provisioning times for application component instances be ensured while utilizing pay-per-use resources in an optimal fashion?



Standby Pooling Process

Application component instances are kept on standby to increase provisioning speed and utilize billing time-slots efficiently.

How can required provisioning times for application component instances be ensured while utilizing pay-per-use resources in an optimal fashion?

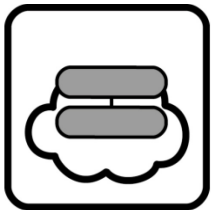


Composite Cloud Application Patterns
Native Cloud Applications

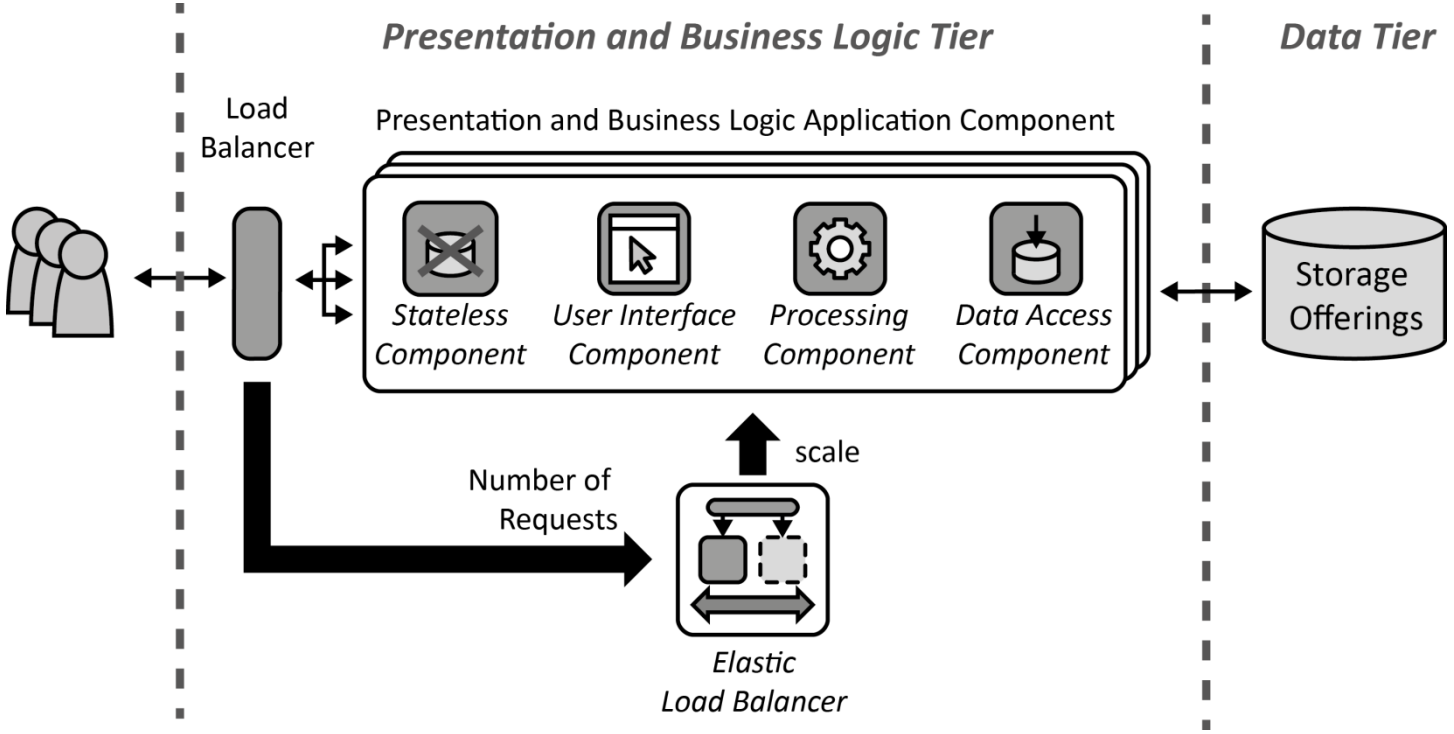
http://www.cloudcomputingpatterns.org

Two-Tier Cloud Application

Presentation and business logic is bundled to one stateless tier that is easy to scale. This tier is separated from the data tier that is harder to scale.

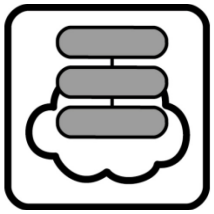


How can application functionality be separated from data handling to scale them independently?

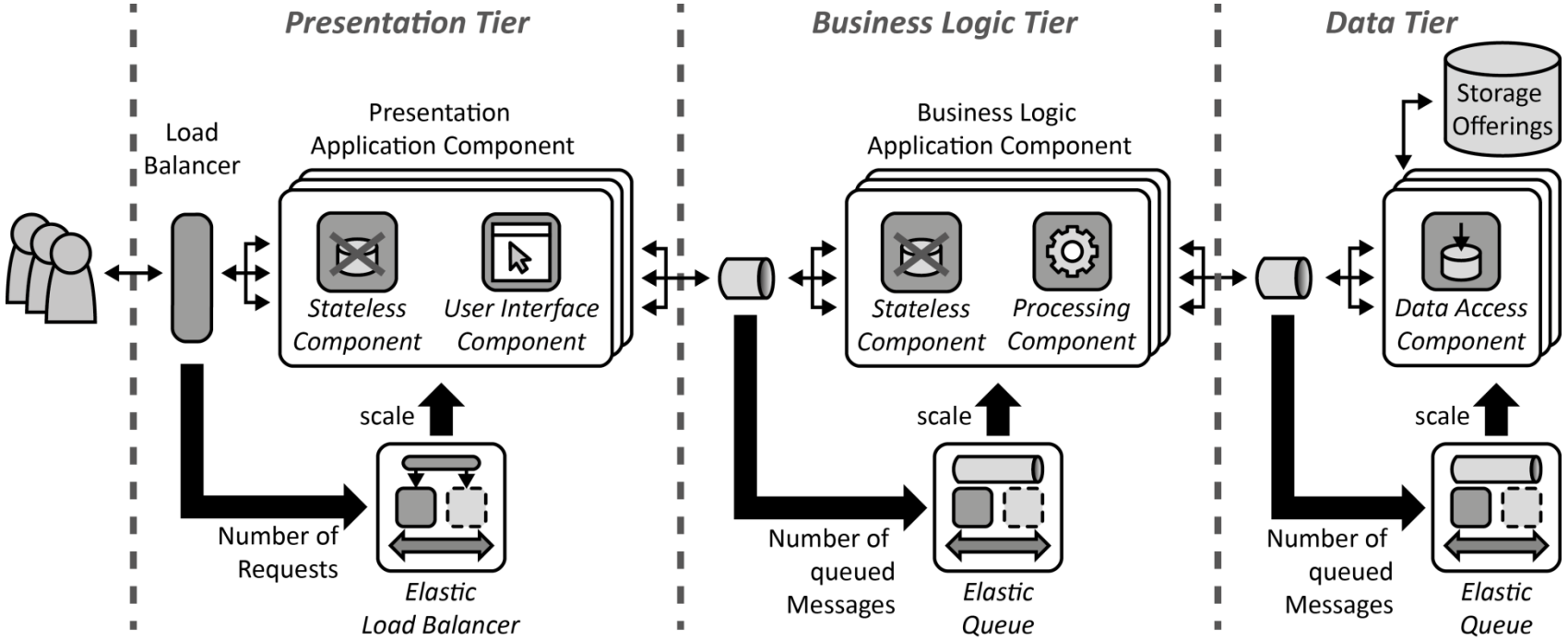


Three-Tier Cloud Application

Presentation, business logic, and data handling are separated to scale stateless presentation and compute-intensive processing independent of the data tier.

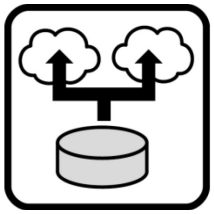


How can presentation logic, business logic, and data handling be decomposed into separate tiers that are scaled independently?

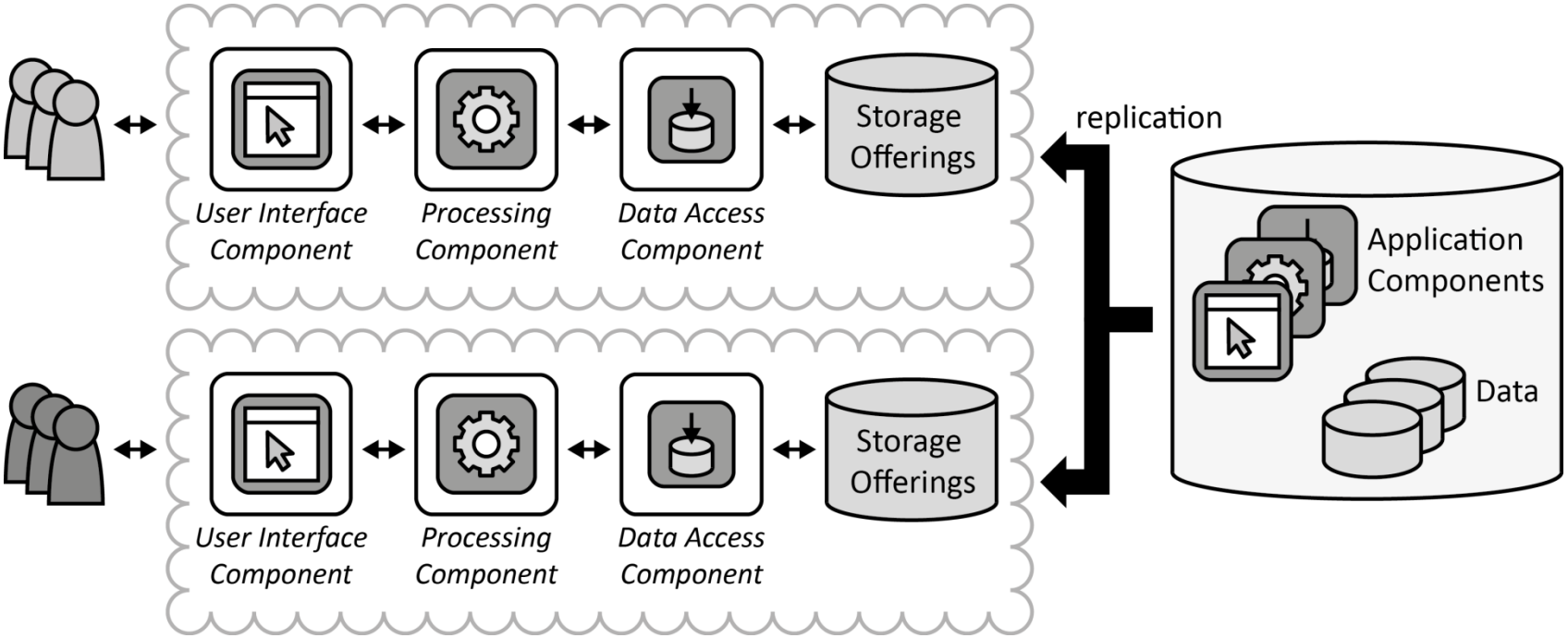


Content Distribution Network

Applications component instances and data handled by them are globally distributed to meet the access performance required by a global user group.



How can timely access to an application be ensured for a globally distributed user group?



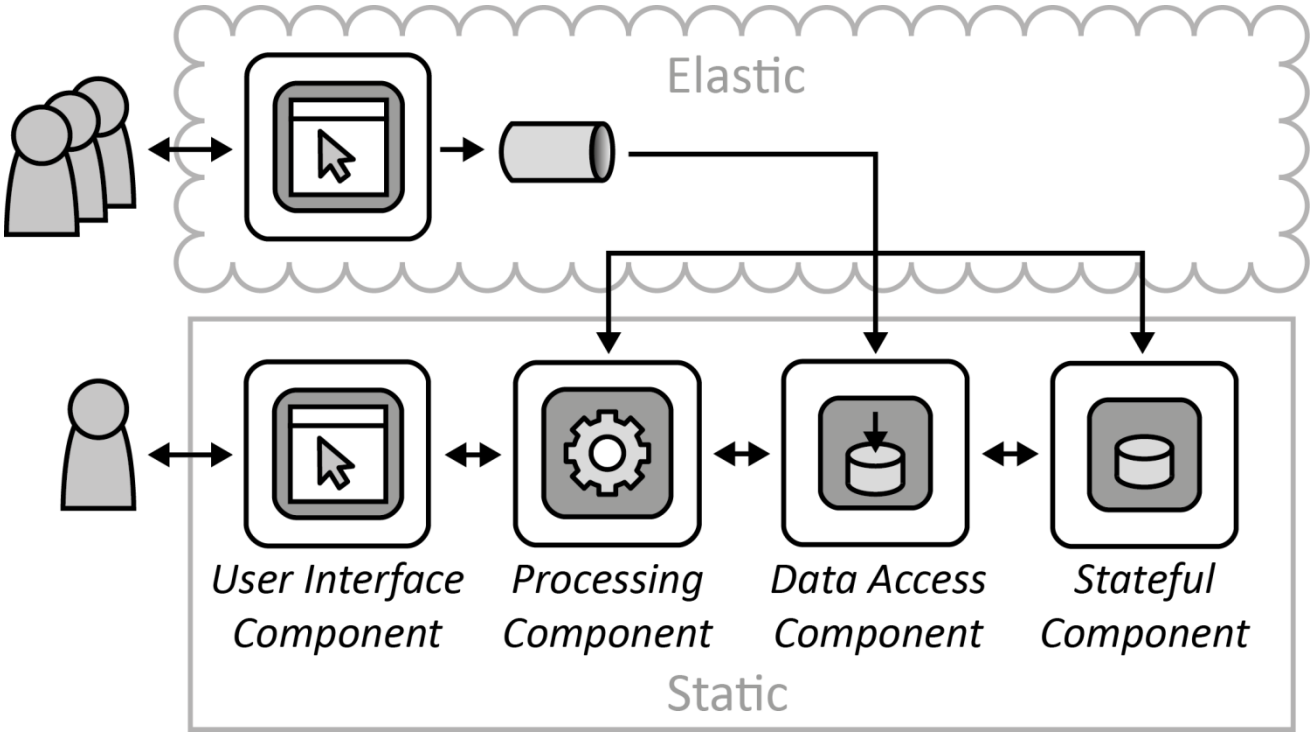
Composite Cloud Application Patterns
Native Cloud Applications

http://www.cloudcomputingpatterns.org

Hybrid User Interface

Varying workload from a user group interacting asynchronously with an application is handled in an elastic environment.

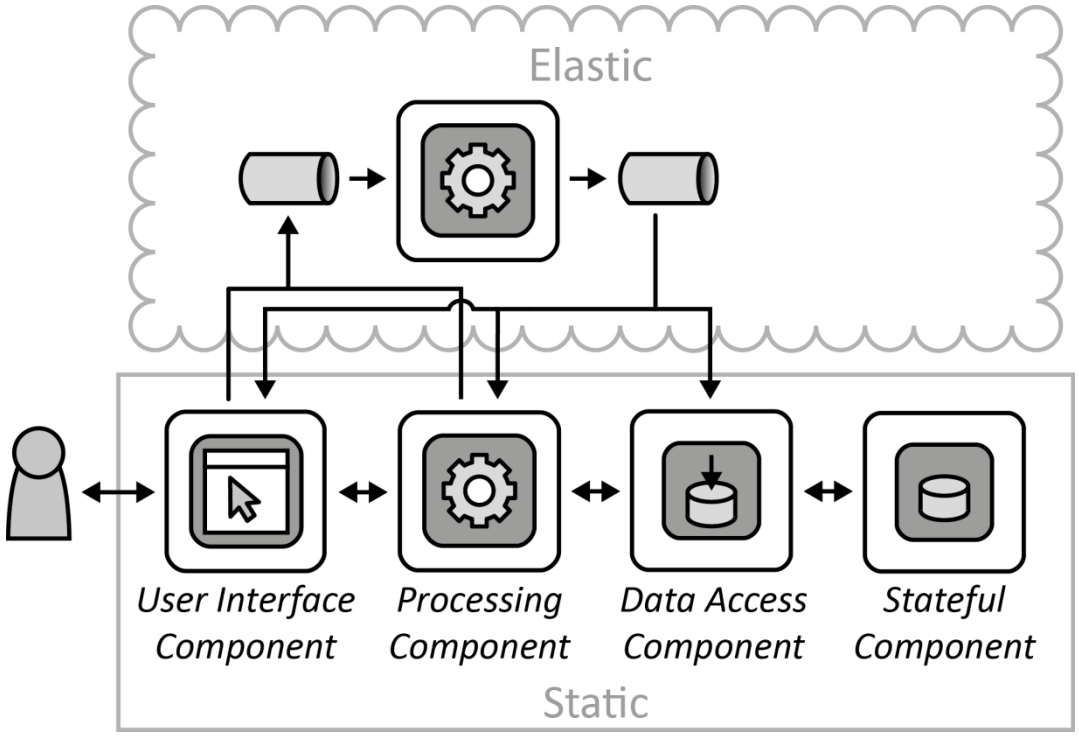
How can a user interface for asynchronous interaction be hosted in a cloud while being integrated with an application otherwise hosted in a static data center?



Hybrid Processing

Processing functionality that experiences varying workload is hosted in an elastic cloud while the remainder of an application resides in a static environment.

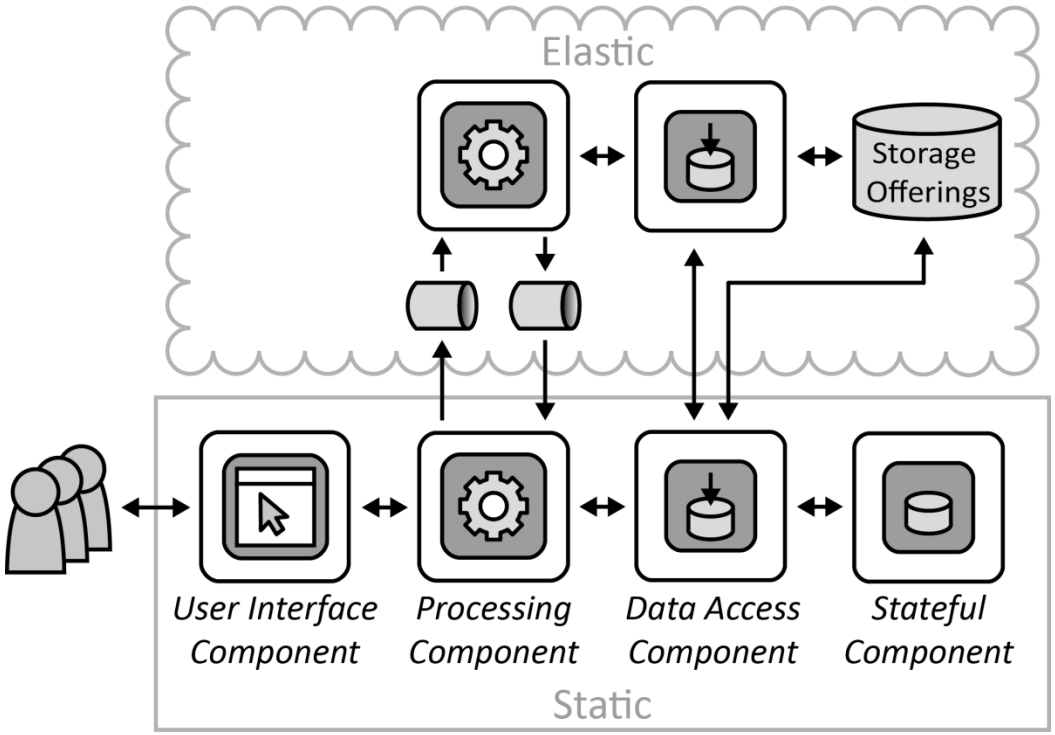
How can processing components that experiences varying workload be hosted in an elastic cloud while the remainder of an application is hosted in a static data center?



Hybrid Backend

Backend functionality (data-intensive processing and storage) is experiencing varying workloads and is hosted in an elastic cloud.

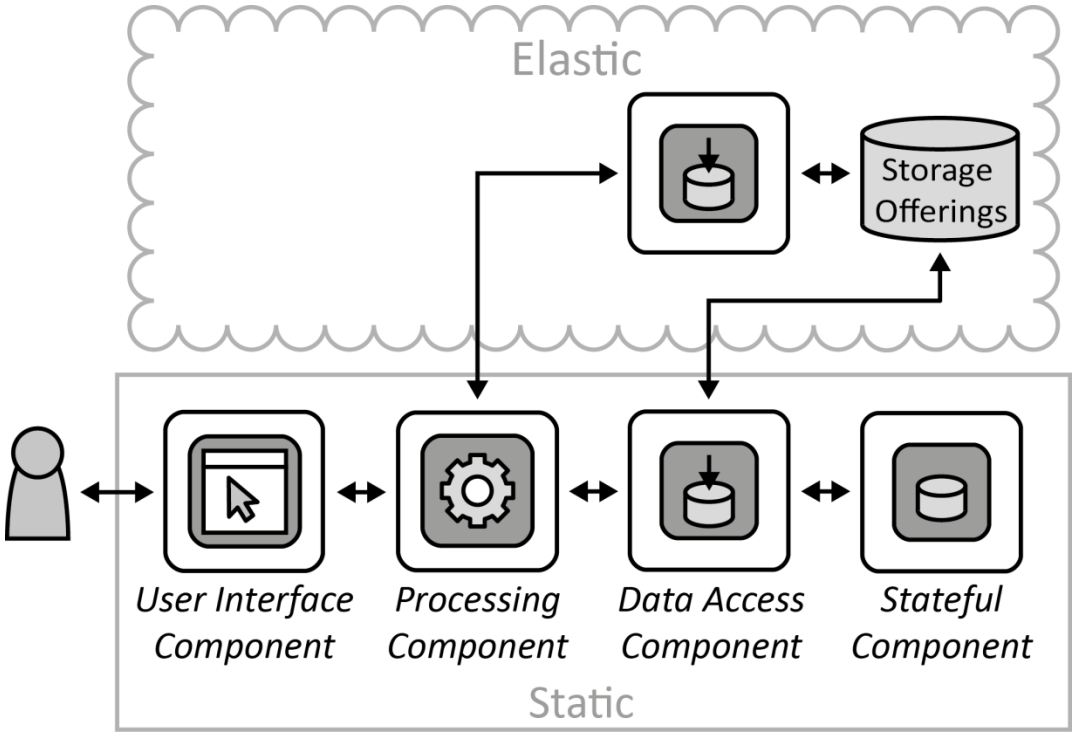
How can processing components that experience varying workload and access large amounts of data be hosted in an elastic environment while the remainder of the application is hosted in a static environment?



Hybrid Data

Data of varying size is hosted in an elastic cloud while the remainder of an application resides in a static environment.

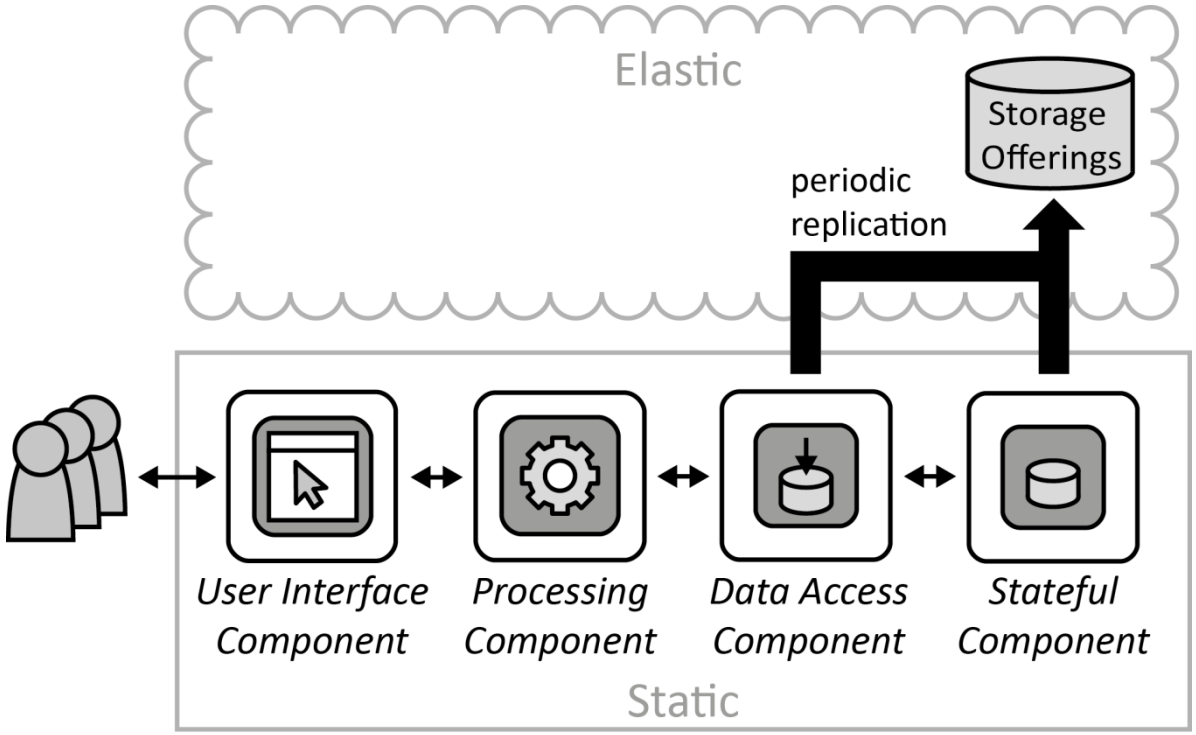
How can a data handling functionality that experiences varying workload be hosted in an elastic cloud while the rest of the application is located in a static data center?



Hybrid Backup

Data is periodically extracted from an application to be archived in an elastic cloud for disaster recovery purposes.

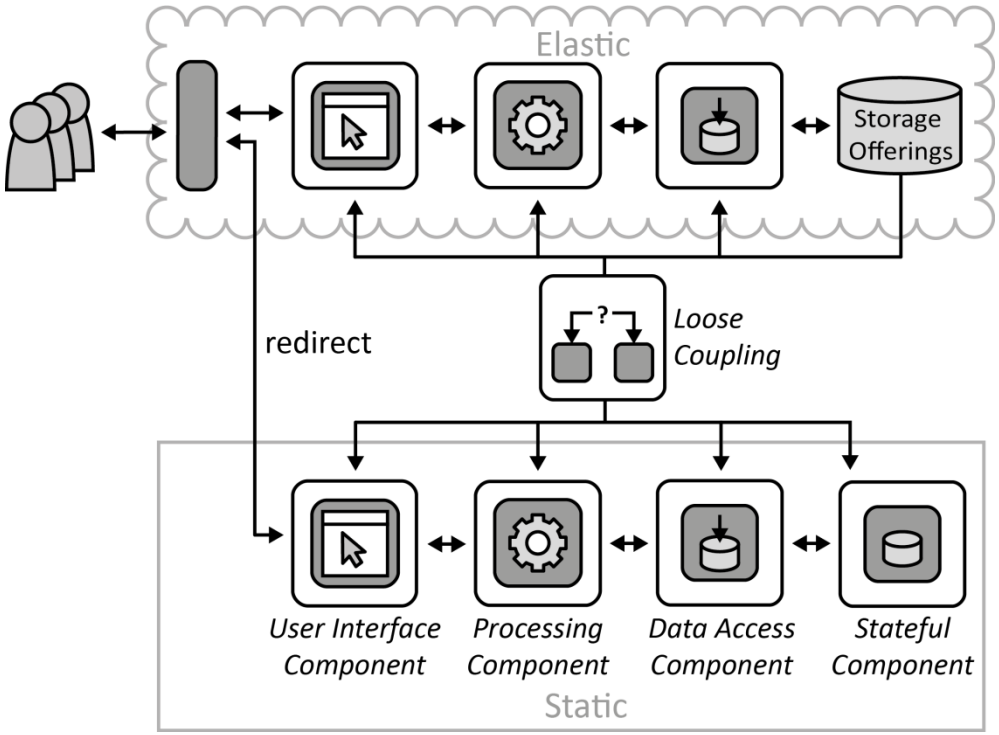
How can data be archived in a remote environment while the remainder of the application is hosted in a static environment?



Hybrid Application Functions

Some application functionality provided by user interfaces, processing, and data handling is experiencing varying workload and is hosted in an elastic cloud.

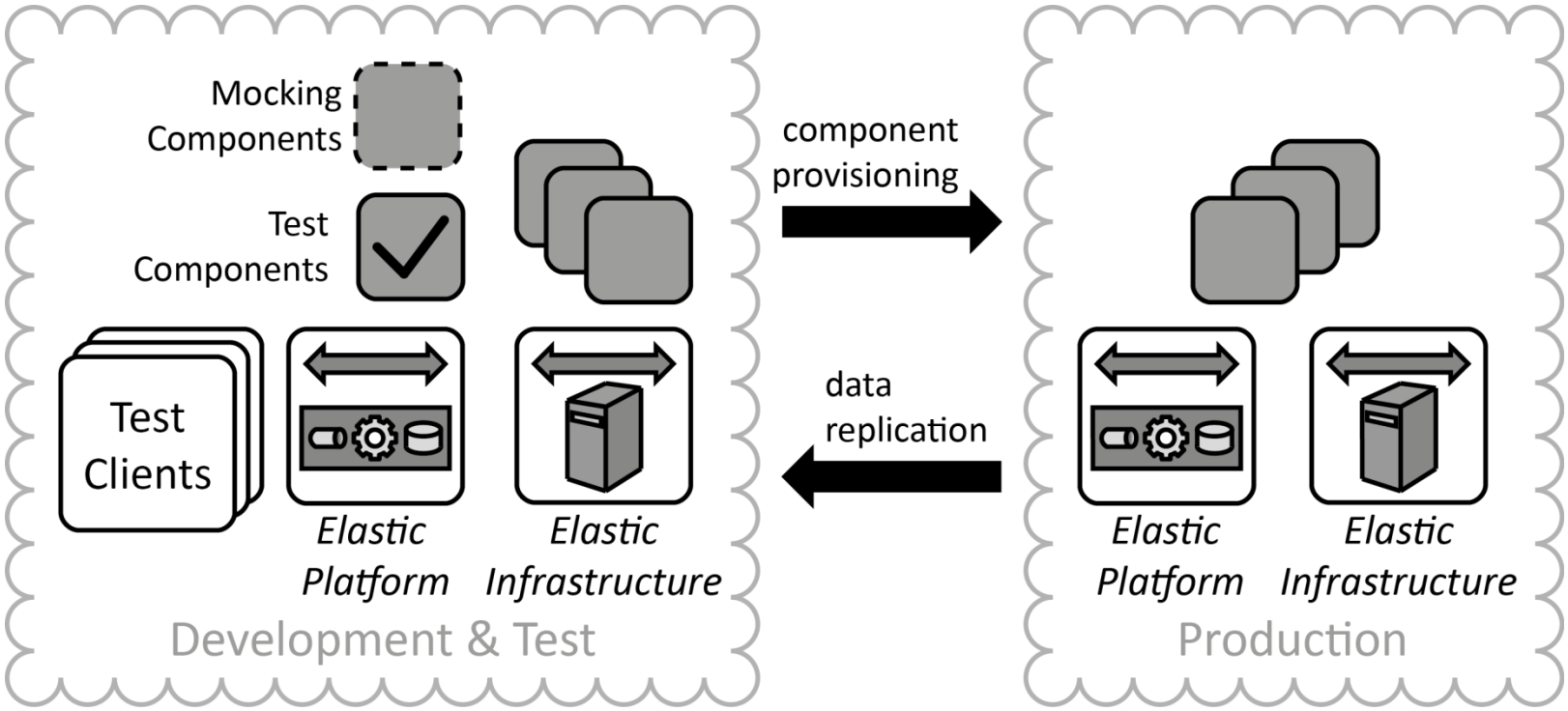
How can arbitrary functionality of an application be distributed among static data centers and elastic clouds best matching its requirements?



Hybrid Development Environment

A production runtime environment is replicated and mocked in an elastic environment where new applications can be developed and tested.

How can an application use different computing environments during its development, test, and production stages?



The End.

http://www.cloudcomputingpatterns.org