







Windows Azure Architecture

I think, therefore I am

Fernando Machado Píriz
Enterprise Architect, Microsoft



@netconfuy
@fmachadopiriz

> 57%

Fortune 500 using Azure

> 250k

Active websites

Greater than
1,000,000

SQL Databases in Azure

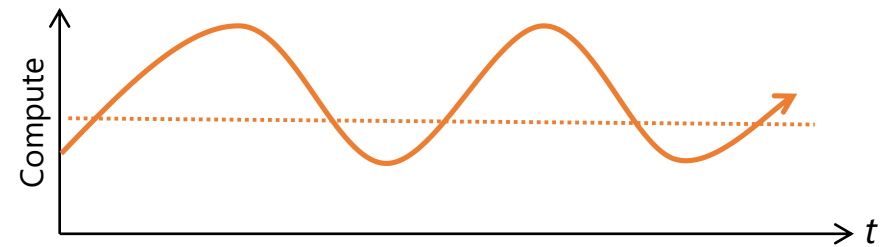
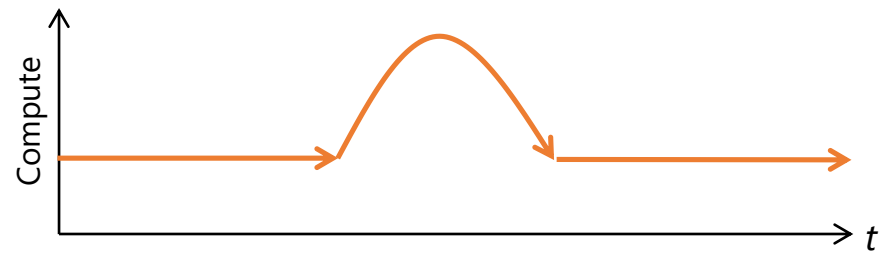
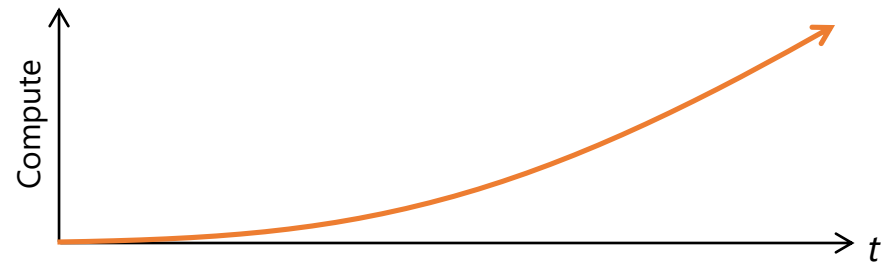
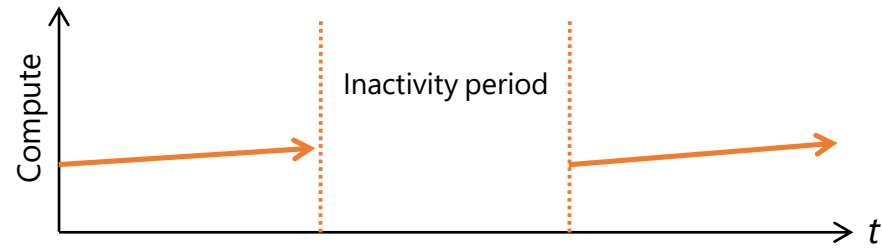
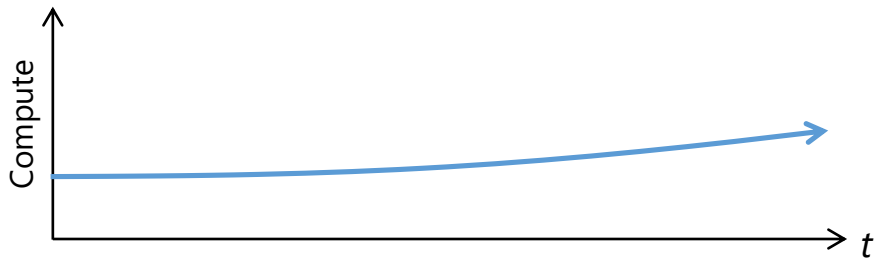
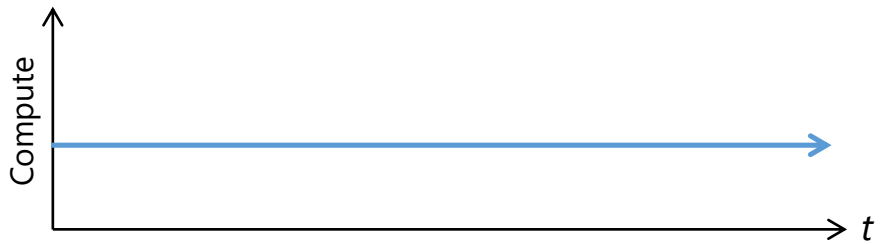
> 20 TRILLION
storage
objects

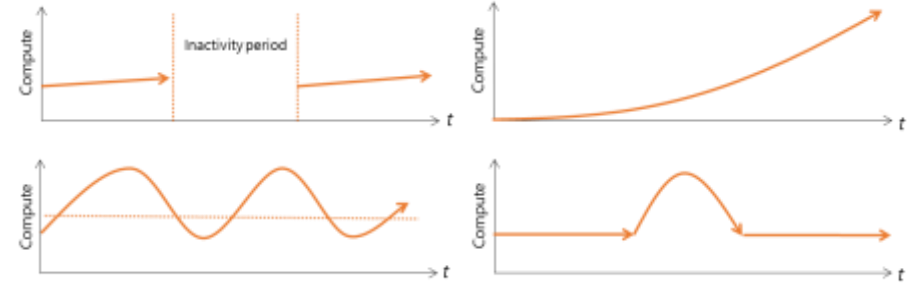
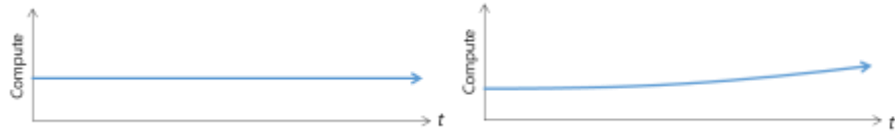
> 300 MILLION
AD users

> 1 MILLION
Developers
registered with
Visual Studio
Online

> 2 MILLION
requests/sec

> 13 BILLION
authentication/wk





Cloud attributes

Pooled and shared resources

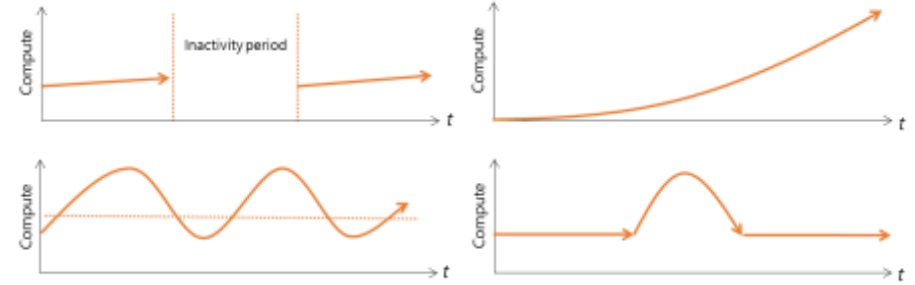
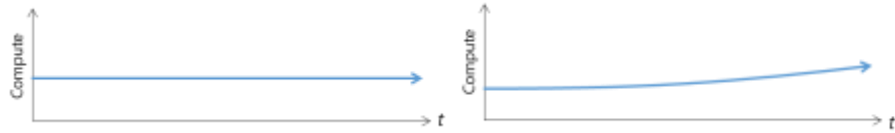
Metered

Cheaper

Elasticity

Pay as you go

New scenarios



Traditional World

Pooled and shared resources

Metered

Cheaper

New, Innovative World

Pooled and shared resources

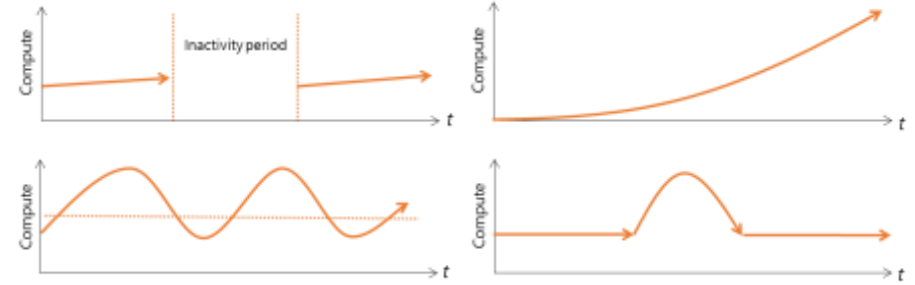
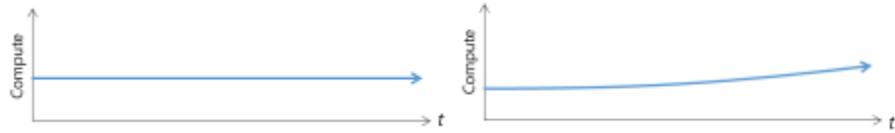
Metered

Cheaper

Elasticity

Pay as you go

New scenarios



Traditional World

Many features

Mean time to failure

Slow update pace

Development lifecycle

Unknown consumption habits

On time, on budget

New, Innovative World

Reliability and agility over features

Mean time to resolution

Continuous updates

Continuous development

Detailed usage patterns

Up and running at appropriate costs

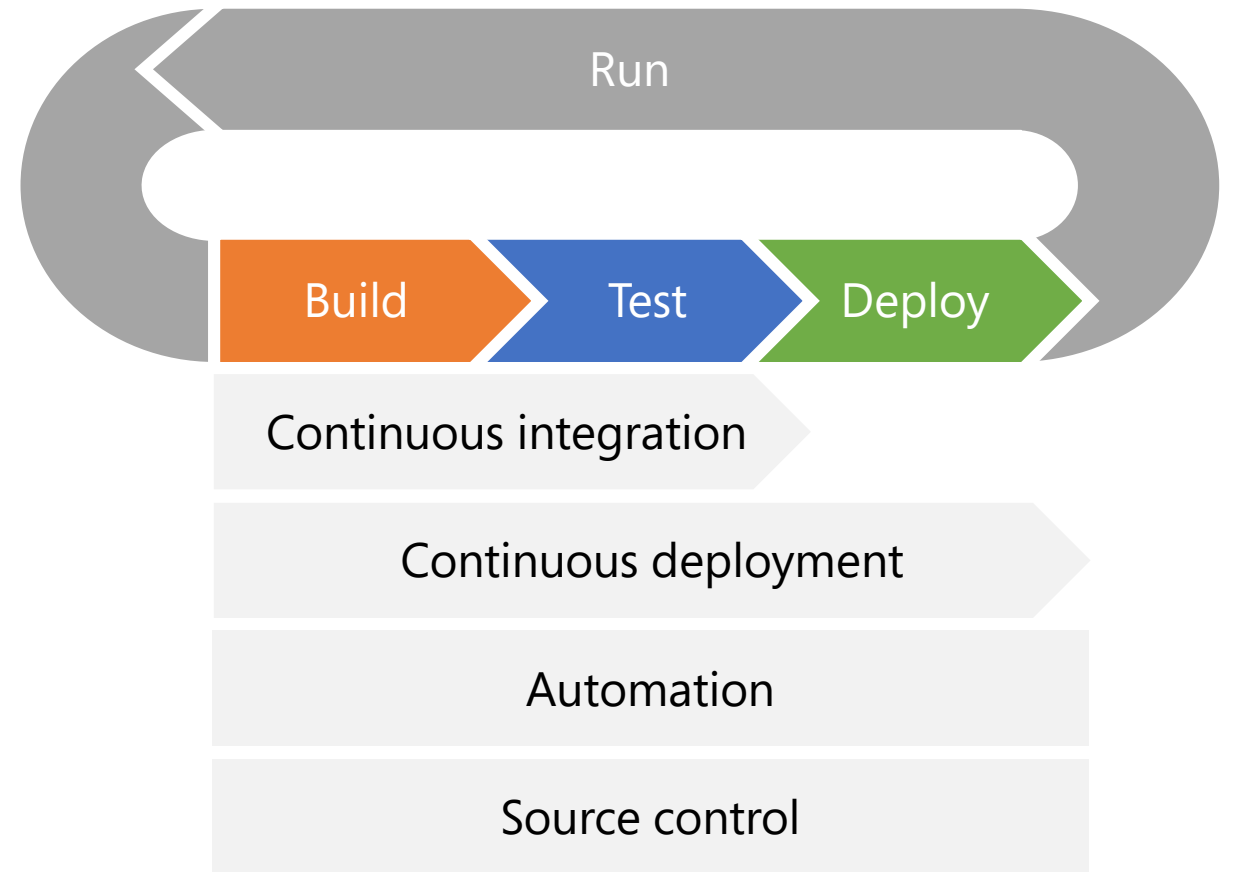
Agility over features

Mean time to resolution

Continuous updates

Continuous development

Up and running at appropriate costs



PowerShell
API



Visual Studio
Online



Team Foundation
Server



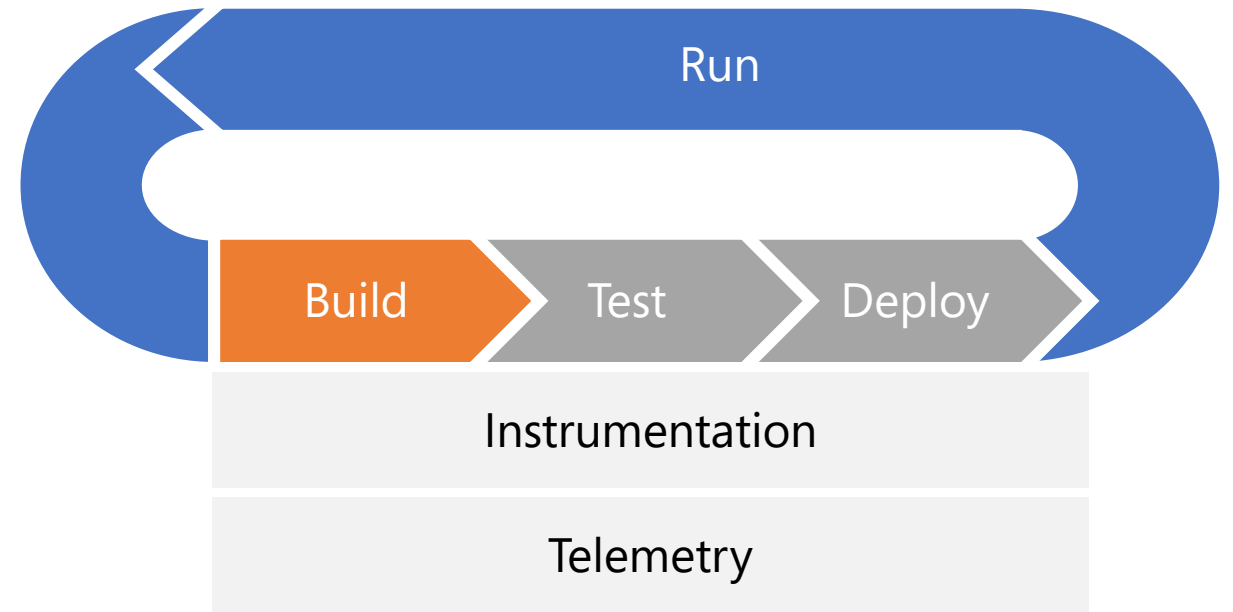
Visual Studio
Git support

Reliability over features

Mean time to resolution

Detailed usage patterns

Up and running at appropriate costs



AppDynamics



Compuware Dynatrace



Microsoft System Center



New Relic



MetricsHub

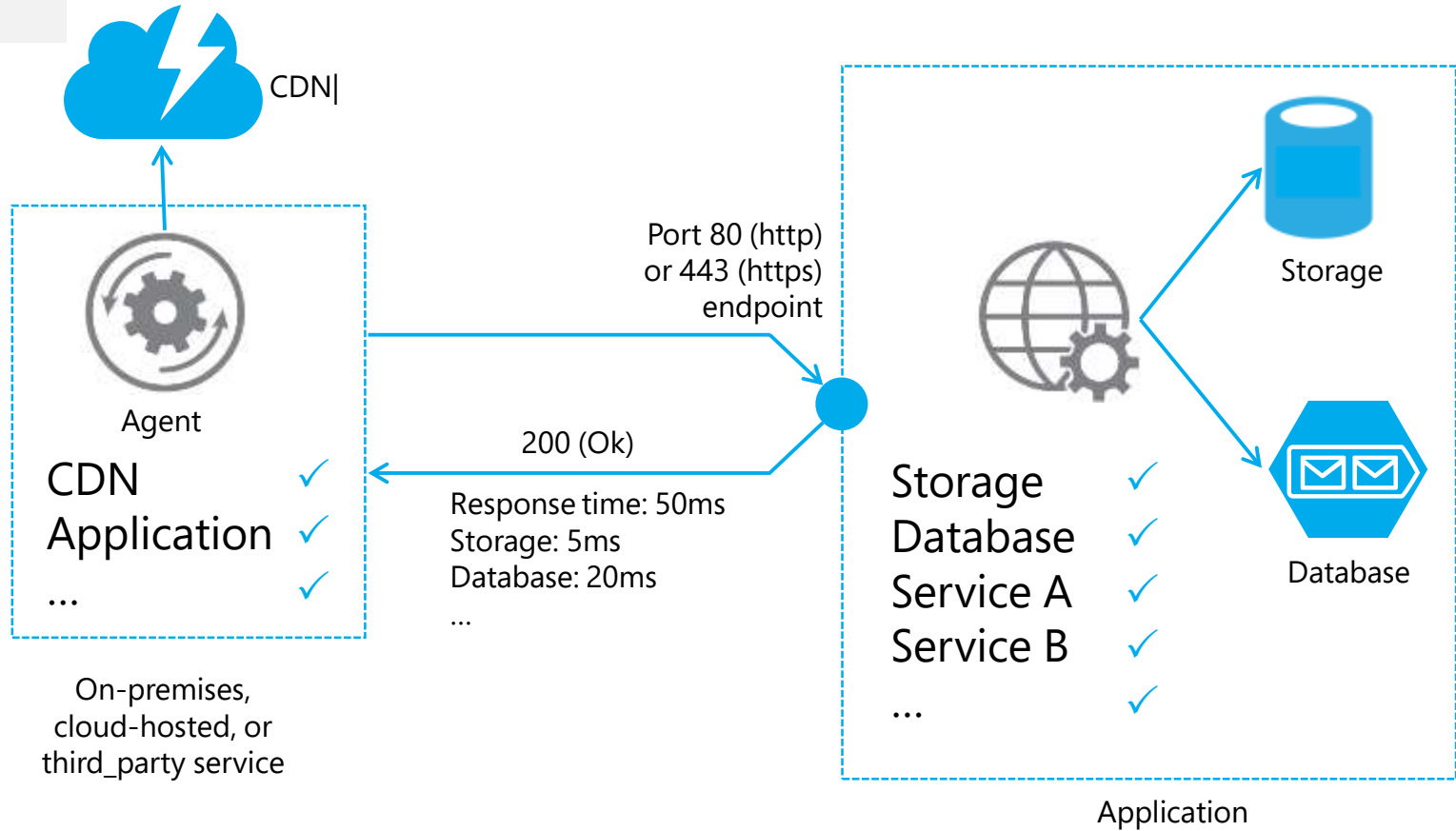


Microsoft Application Insights for Visual Studio Online

Reliability over features

Up and running at appropriate costs

Health check



Health Endpoint Monitoring

Reliability over features

Mean time to resolution

Continuous updates

Up and running at appropriate costs

Web development

Stateless web servers

No session state; or distributed caching

Content distribution networks

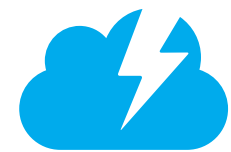
Default to async



Websites



Virtual Machines



CDN

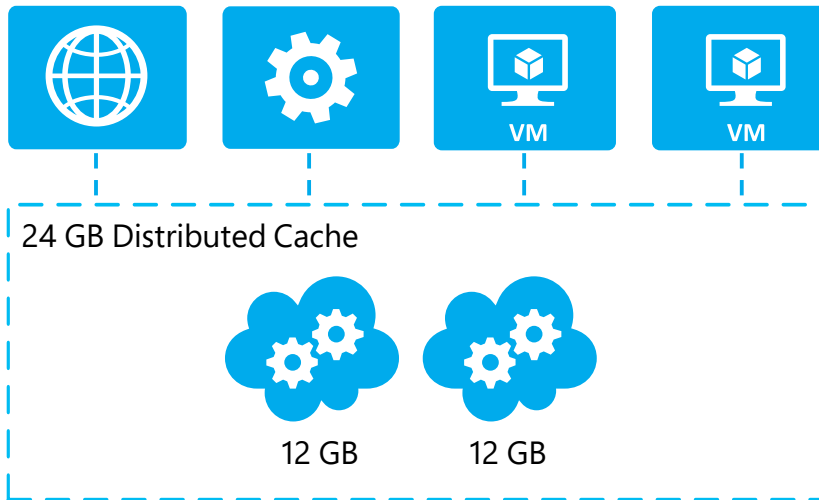
Reliability over features

Continuous updates

Up and running at appropriate costs

Caching

Web roles, worker roles, virtual machines

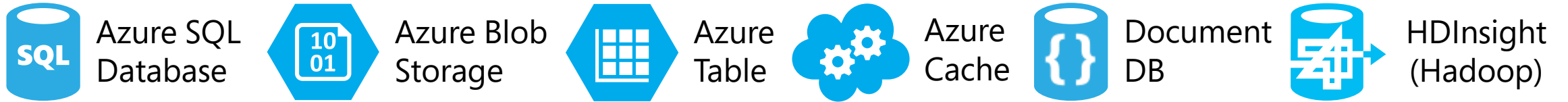


Reliability over features

Up and running at appropriate costs

Storage

ACID vs BASE



Relational	Key/Value	Column Family	Document	Graph
<ul style="list-style-type: none">Azure SQL DatabaseSQL ServerOracleMySQL <ul style="list-style-type: none">SQL CompactSQLitePostgres	<ul style="list-style-type: none">Azure Blob StorageAzure Table Storage <ul style="list-style-type: none">Azure CacheRedisMemcachedRiak	<ul style="list-style-type: none">CassandraHBase	<ul style="list-style-type: none">DocumentDBMongoDBRavenDBCouchDB	<ul style="list-style-type: none">Neo4J

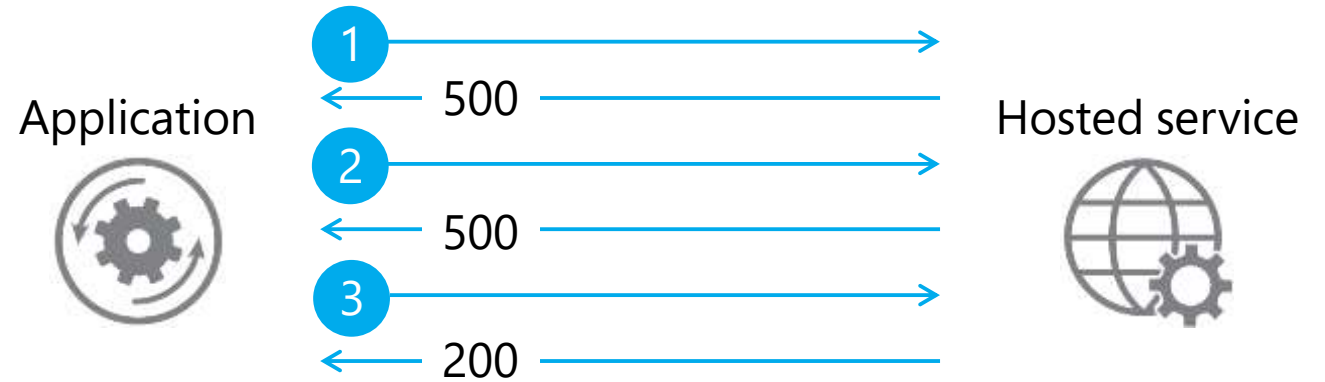
■ PaaS (managed services)

■ IaaS (virtual machines)

Reliability over features

Up and running at appropriate costs

Resiliency

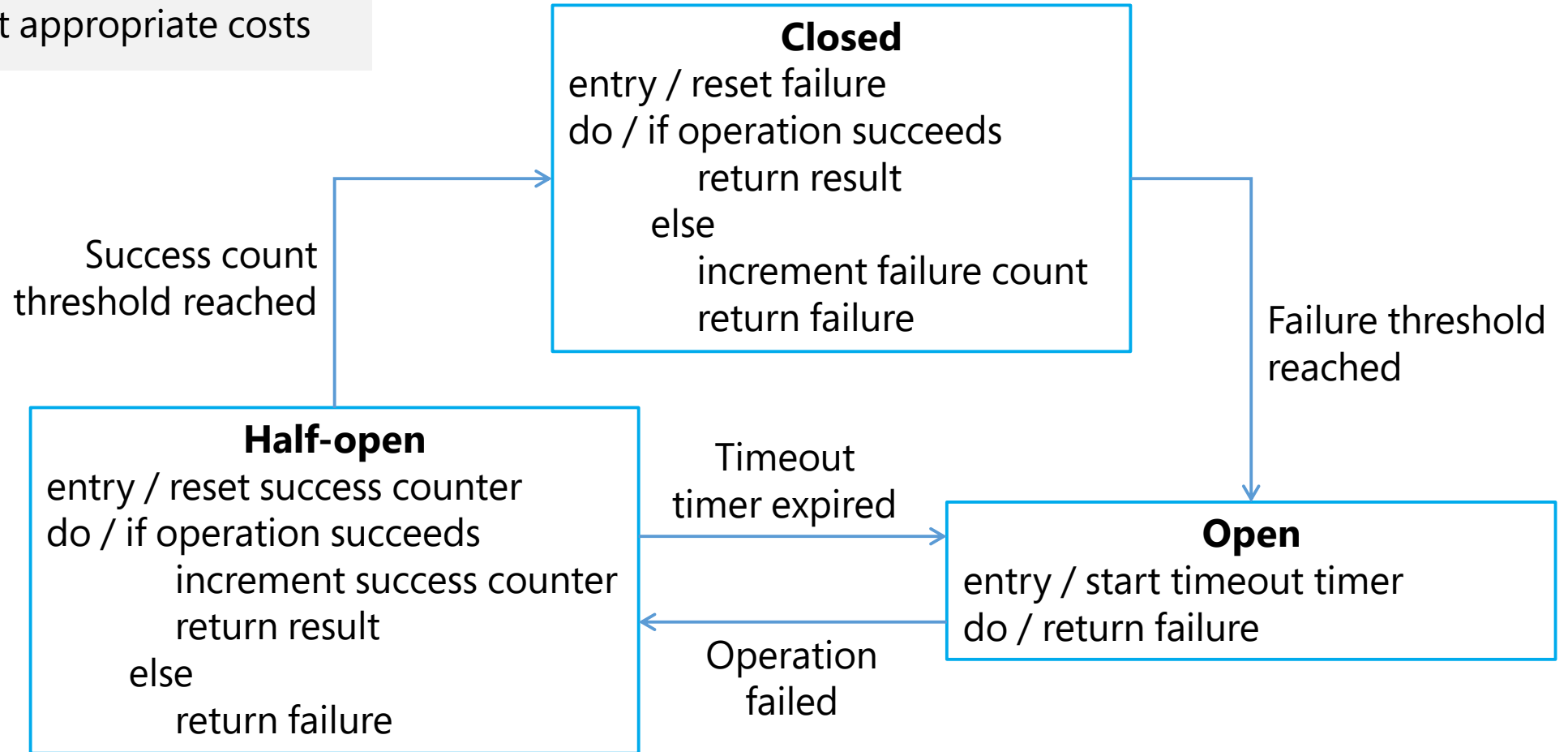


Retry

Reliability over features

Up and running at appropriate costs

Resiliency

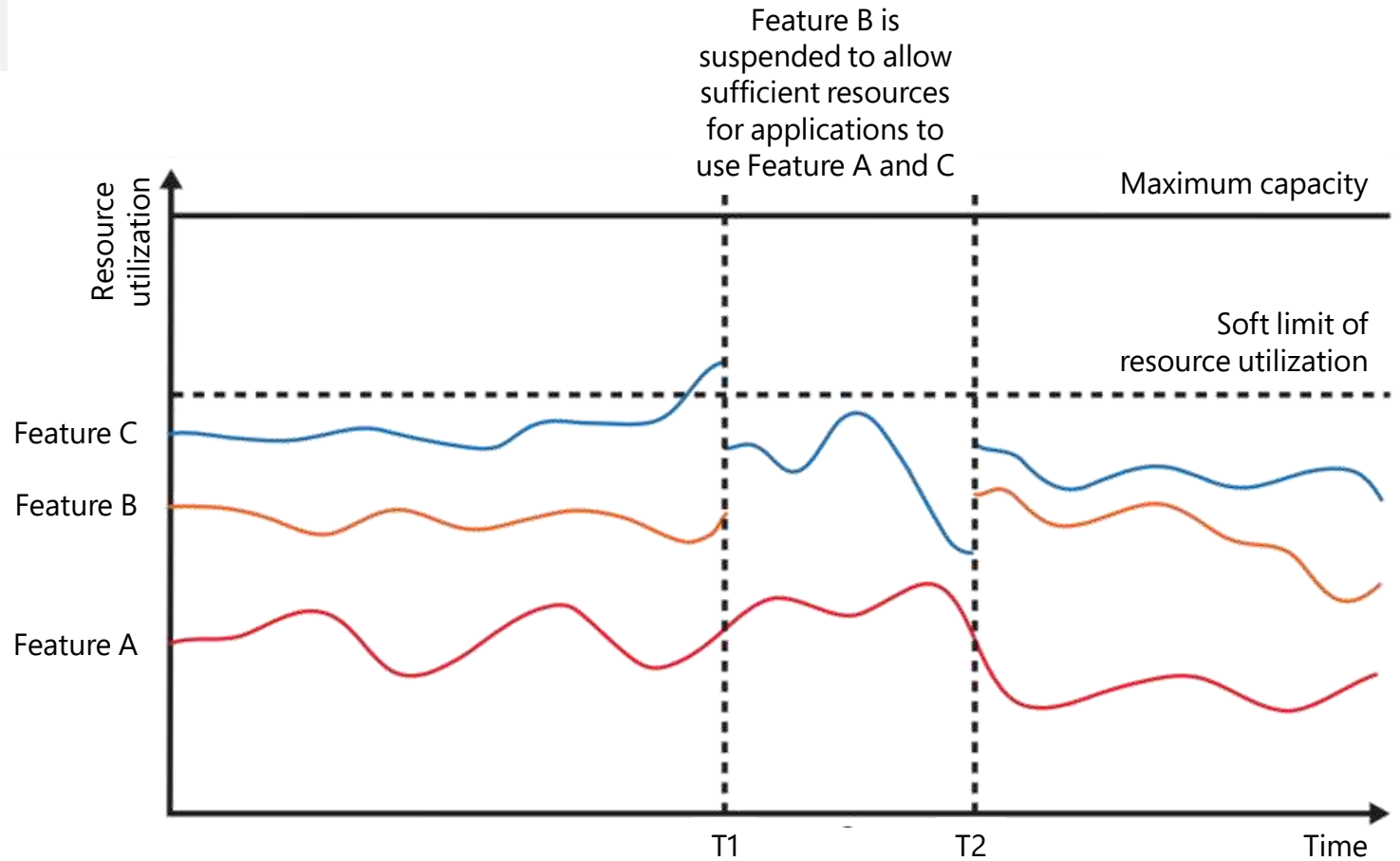


Circuit breaker

Reliability over features

Up and running at appropriate costs

Resiliency

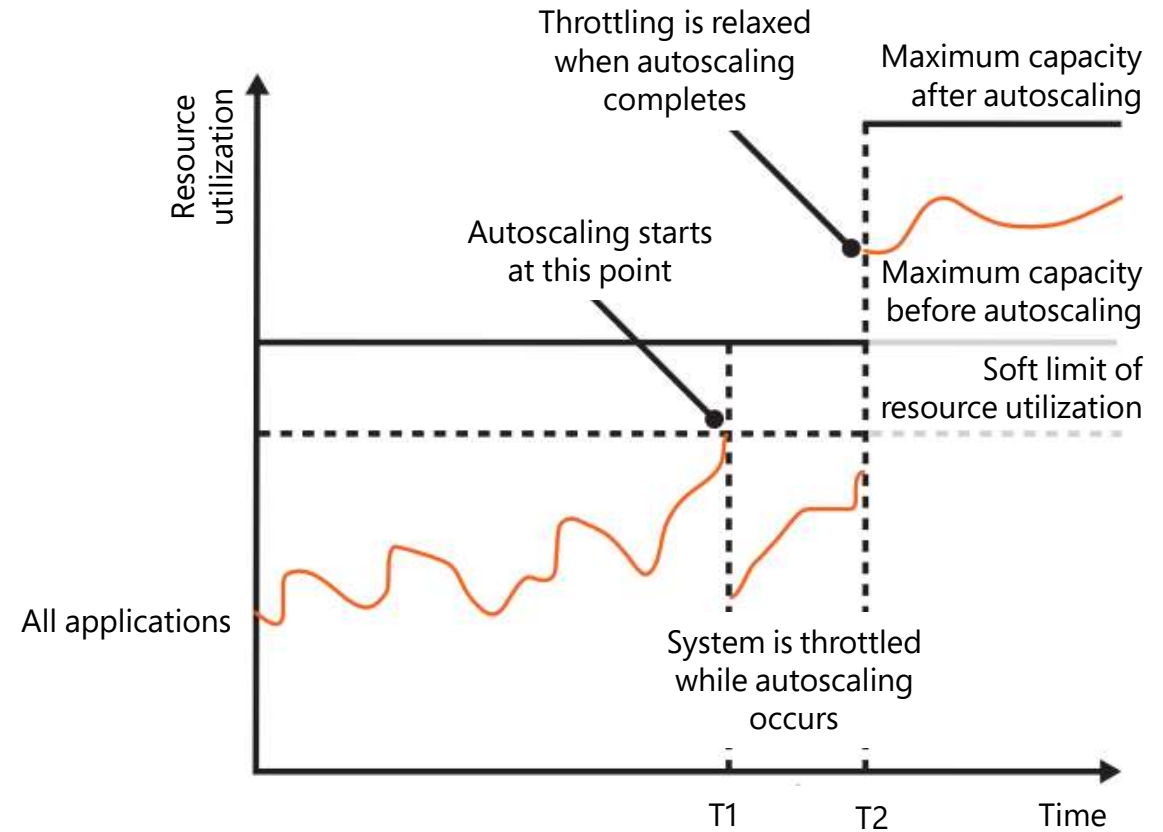


Throttling

Reliability over features

Up and running at appropriate costs

Resiliency

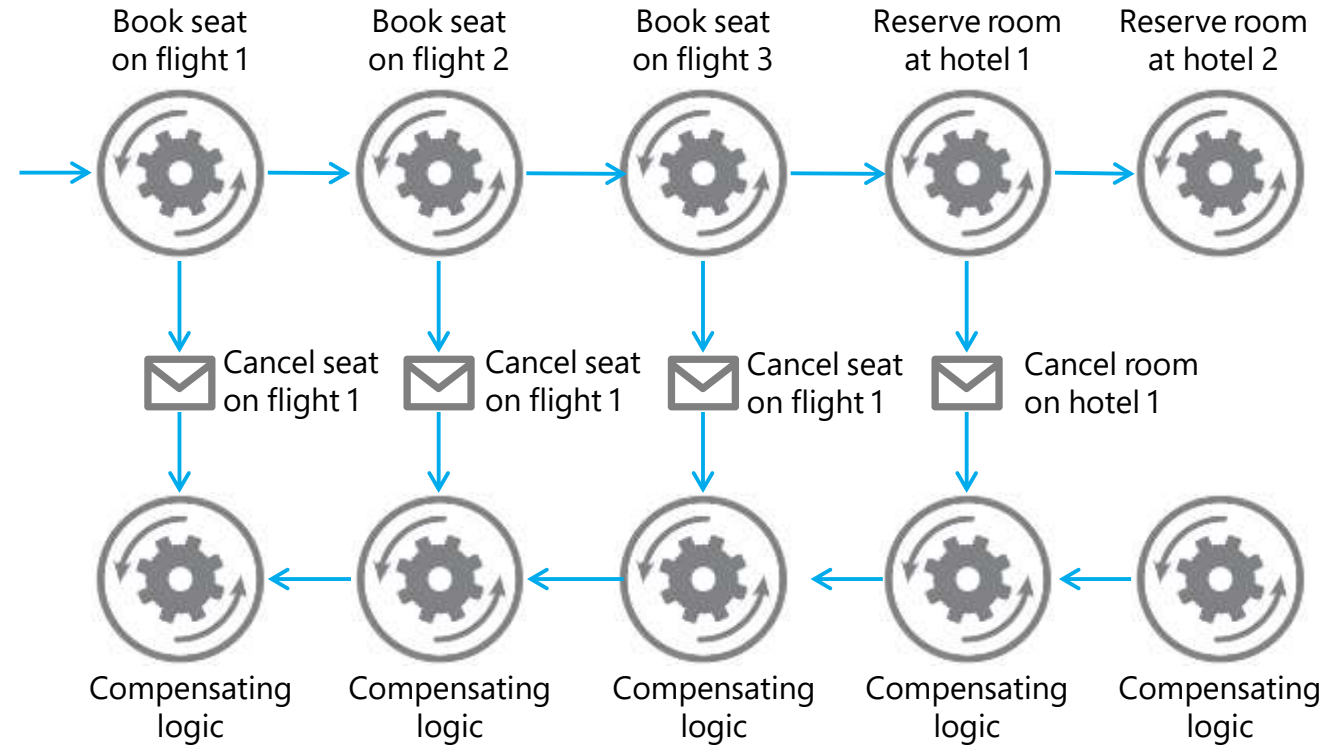


Throttling

Reliability over features

Up and running at appropriate costs

Resiliency



Compensating transactions

Reliability over features

Up and running at appropriate costs

Default to async

```
private async Task MyTaskAsync(SomeType param)
{
    var result = await object.ProcessMessagesAsync(param);
}
```

Reliability over features

Up and running at appropriate costs

Default to async



Storage Queue



BizTalk Services



Service Bus



Service Bus Queue



Service Bus Relay



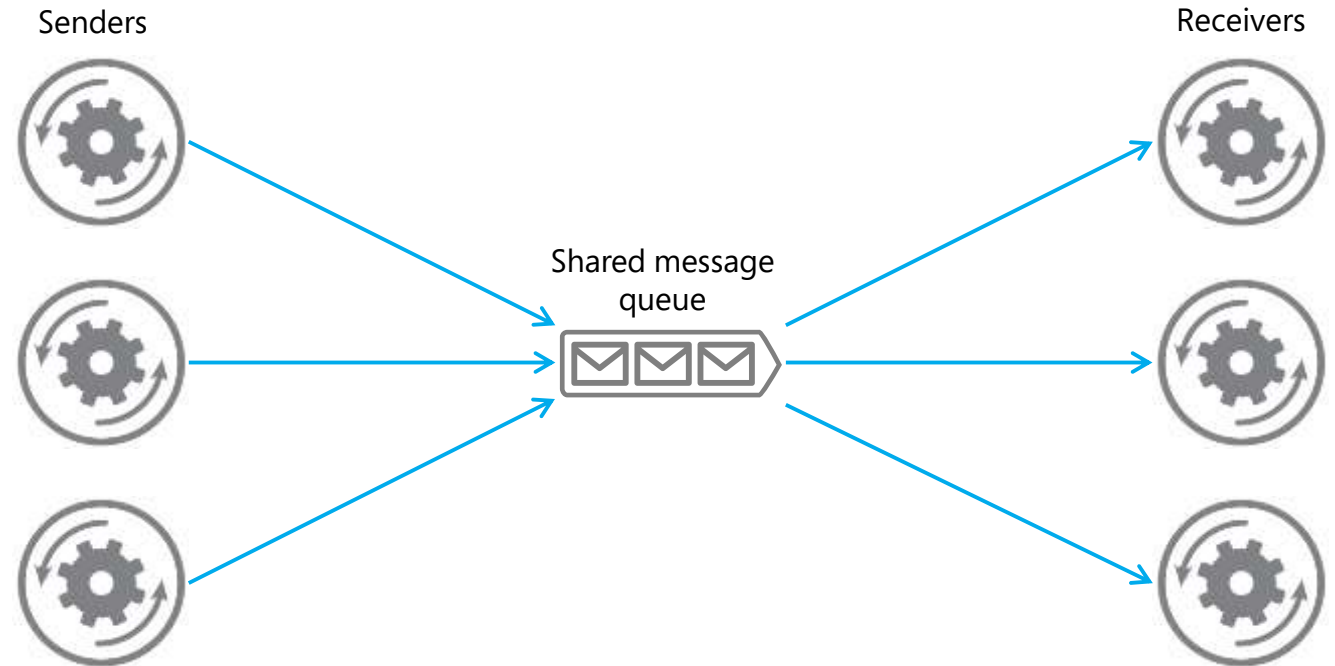
Service Bus Topic

Asynchronous Messaging Primer

Reliability over features

Up and running at appropriate costs

Default to async

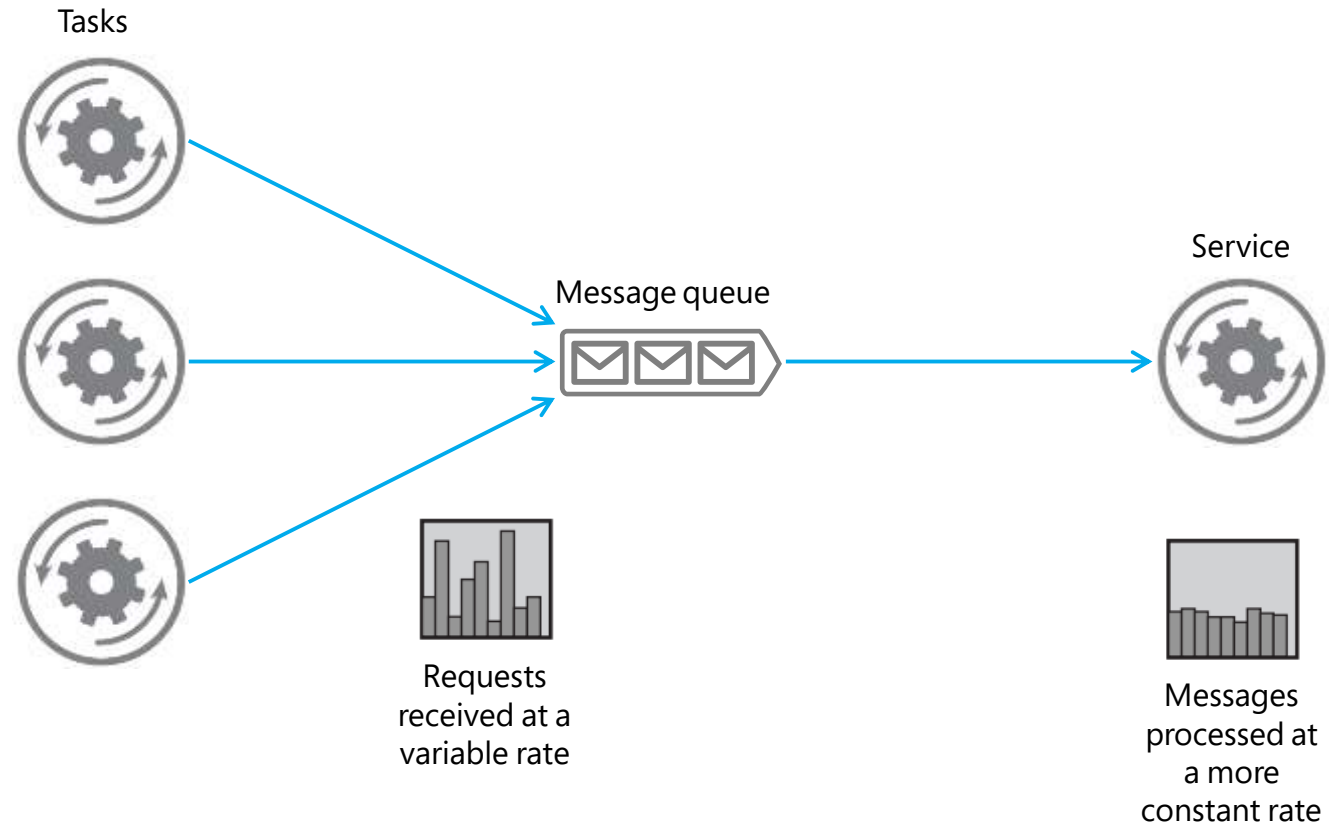


Asynchronous Messaging Primer

Reliability over features

Up and running at appropriate costs

Default to async

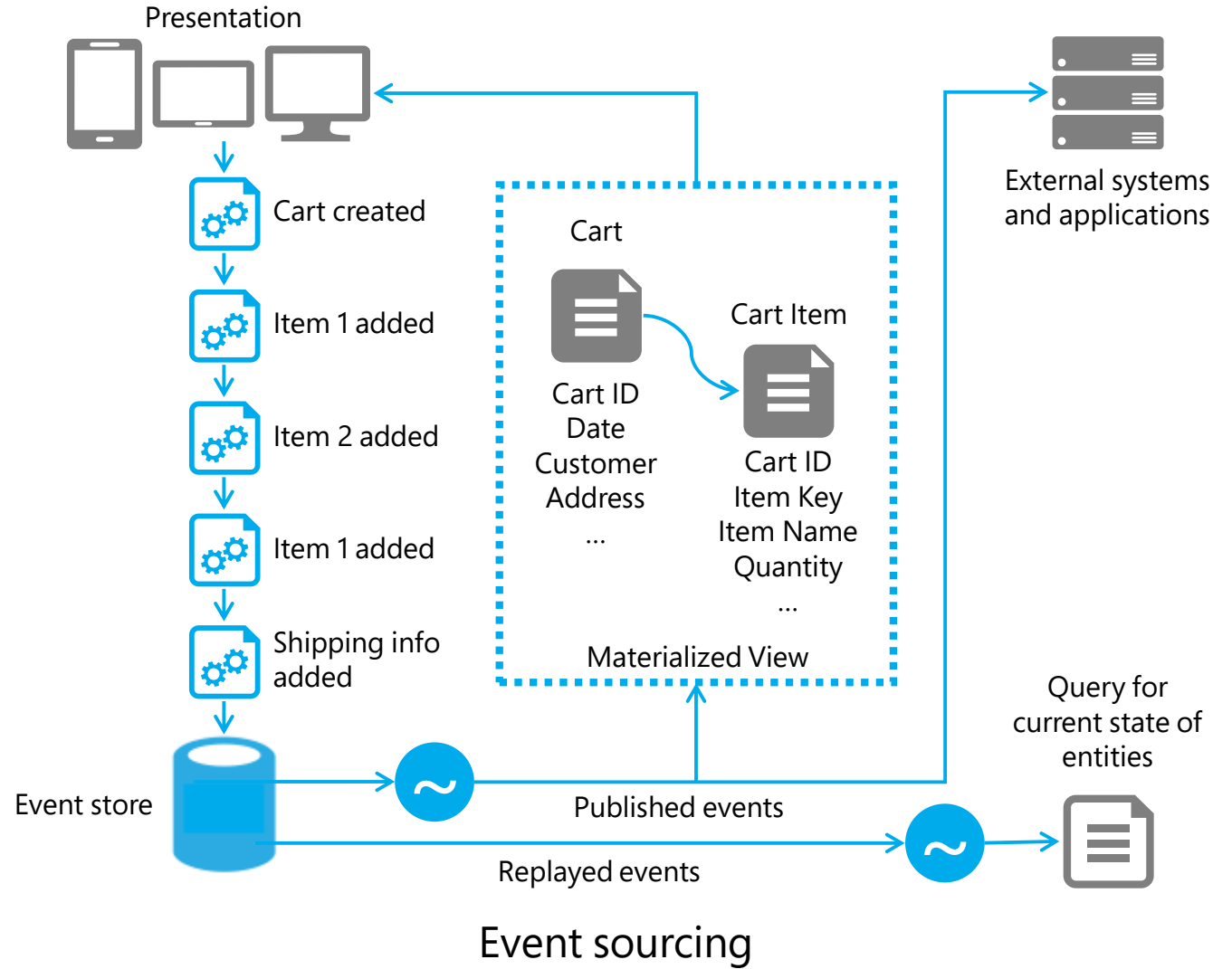


Queue-Based Load Leveling

Reliability over features

Up and running at appropriate costs

Default to async





Building Cloud Apps with Microsoft Azure

Best practices for DevOps, data
storage, high availability, and more



Professional

Scott Guthrie, Mark Simms, Tom Dykstra,
Rick Anderson, Mike Wasson



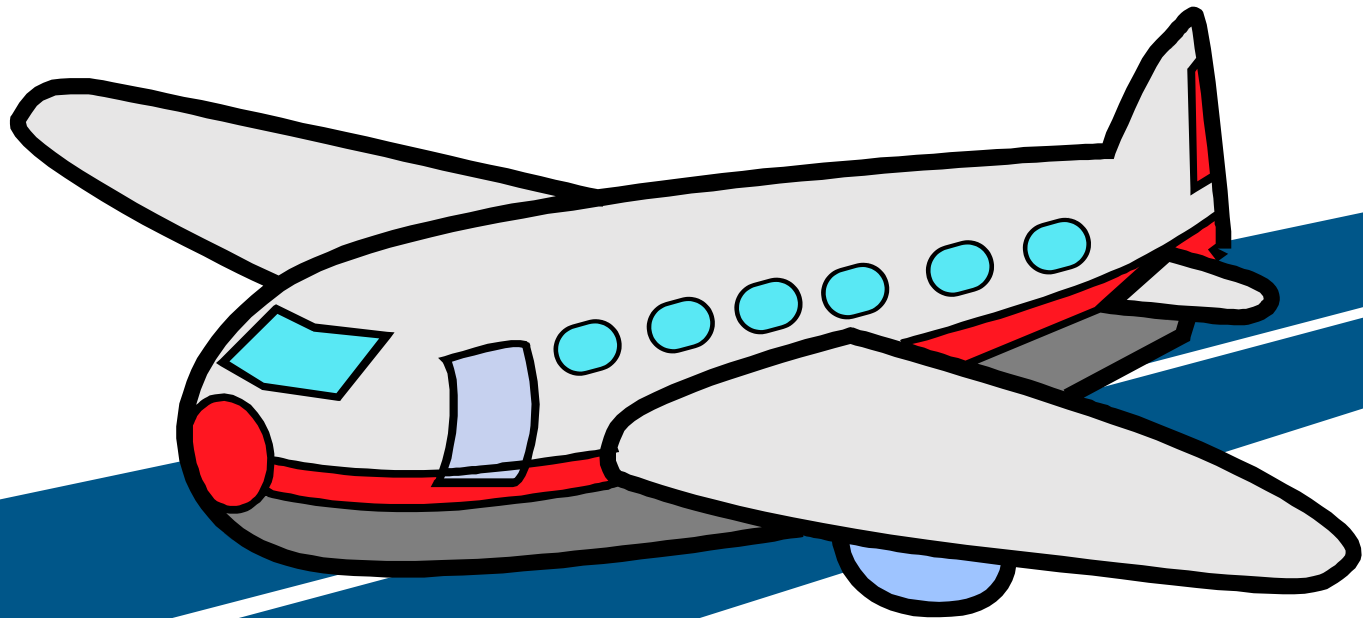
Cloud Design Patterns

PRESCRIPTIVE ARCHITECTURE GUIDANCE FOR CLOUD APPLICATIONS

Alex Homer
John Sharp
Larry Brader
Masashi Narumoto
Trent Swanson



patterns & practices





Questions?
Thanks!

Fernando Machado Píriz
Enterprise Architect, Microsoft



@netconfuy
@fmachadopiriz