Curso de Especialização em Desenvolvimento Java

Arquiteturas e Infrastrutura de Sistemas Corporativos

Prof. Emílio Dias
emiliodias@gmail.com
http://www.github.com/emiliodias

Conteúdo da disciplina

- Introdução a arquitetura
- Arquitetura orientada a serviços
- Microservices
- Cloud Computing
- Serverless
- Infrastructure as a code
- Implementações
  - ESB, BPEL, etc...

Avaliação

- Presença e participação em sala de aula.

  - **Aula 1:** Apresentação da disciplina, e introduções, os alunos não serão avaliados.
  - **Aula 2 à 6:** 20 pontos por presença + participação

    Participação = Interesse no conteúdo apresentado, questionamentos, execução de exercícios quando propostos.

Arquitetura

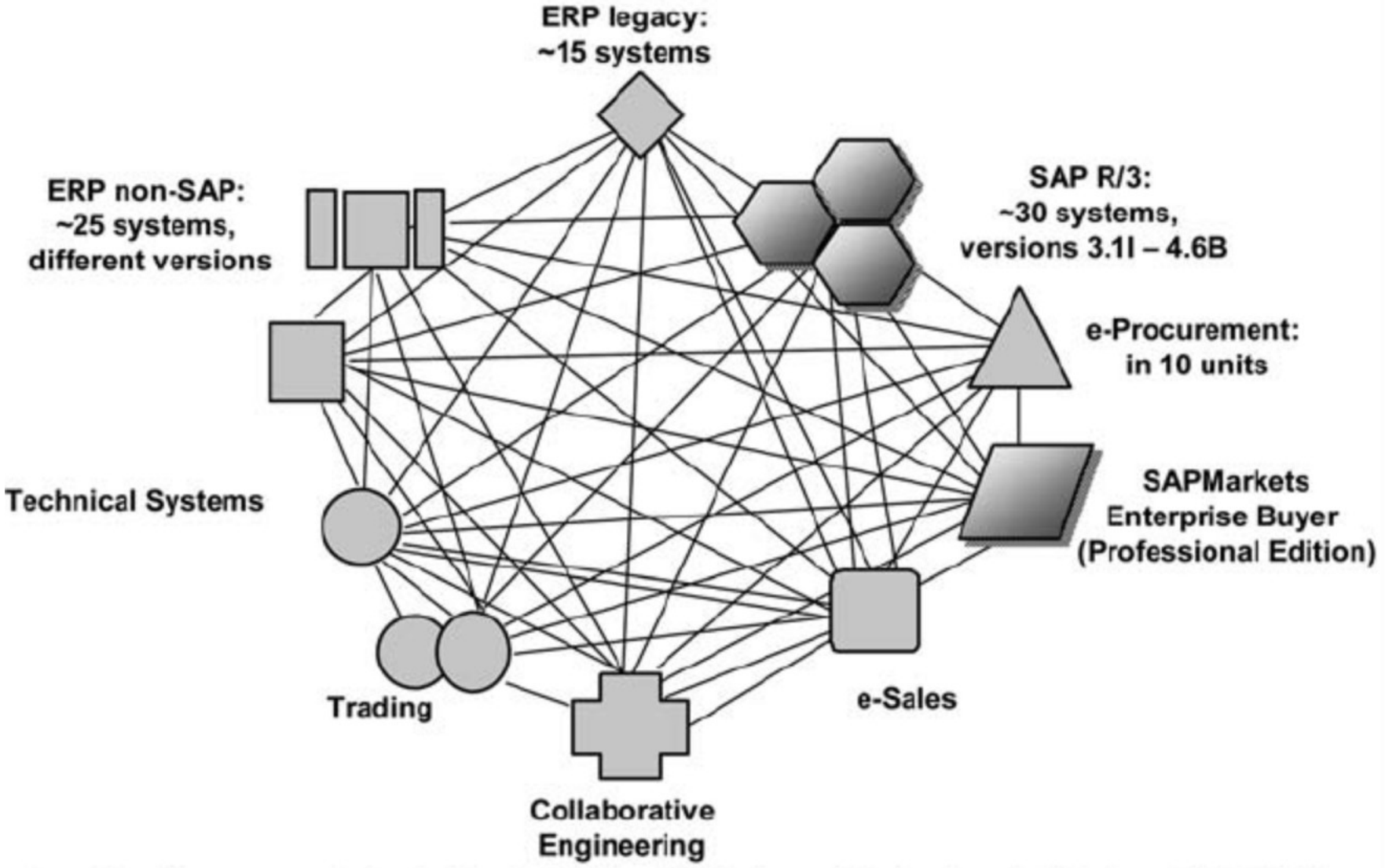
“A software architecture is an abstraction of the run-time elements of a software system during some phase of its operation. A system may be composed of many levels of abstraction and many phases of operation, each with its own software architecture.”

Roy Thomas Fielding

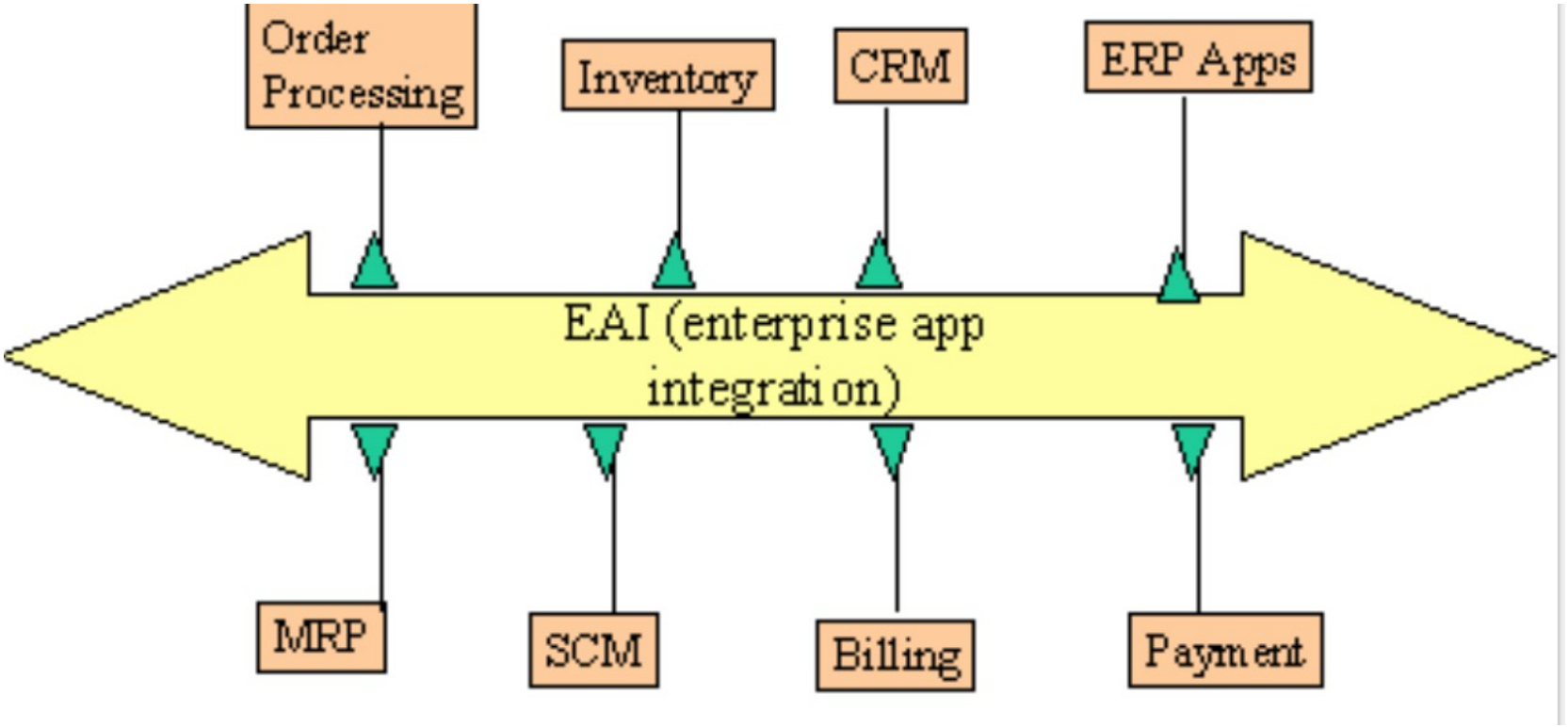
“A software architecture is defined by a configuration of architectural elements--components, connectors, and data--constrained in their relationships in order to achieve a desired set of architectural properties.”

Roy Thomas Fielding

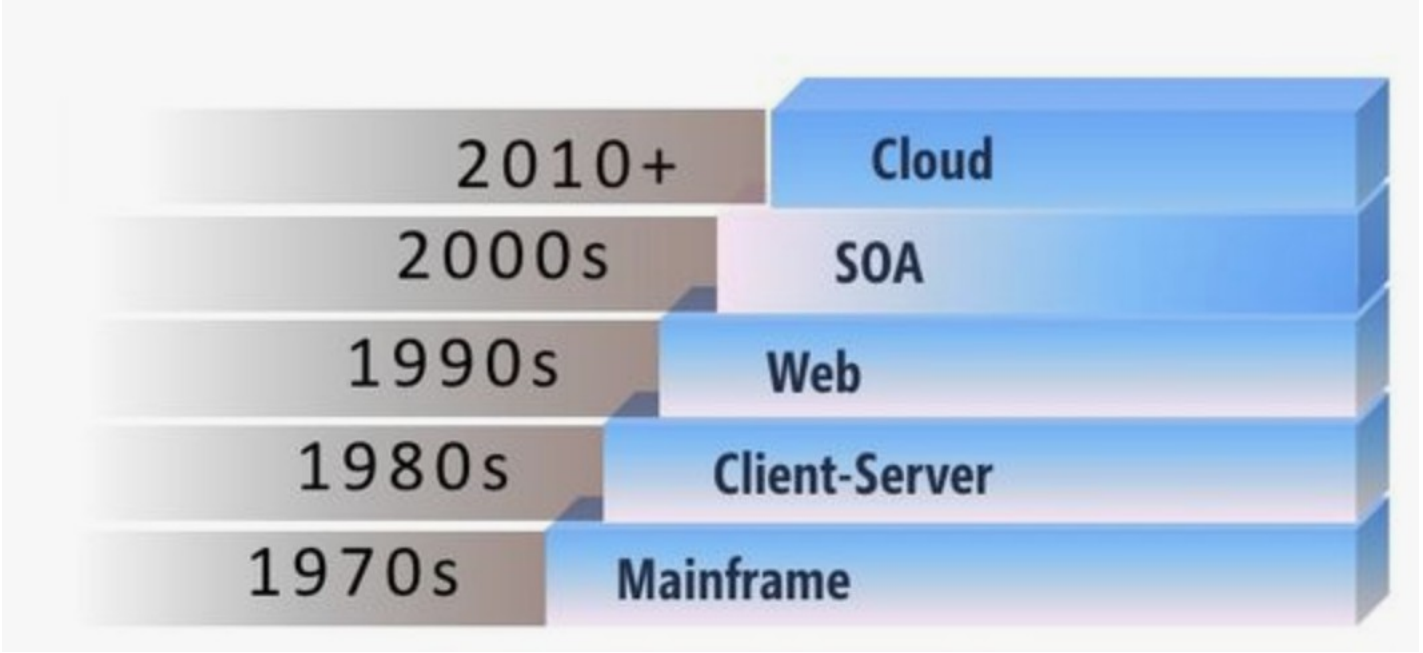
Point-to-Point



EAI (Enterprise Application Integration)



Evolução


2010+
Cloud
2000s
SOA
1990s
Web
1980s
Client-Server
1970s
Mainframe

SOA

“A Service-oriented architecture (SOA) é um padrão de mercado vagamente definido que apresenta todos os processos de negócios de uma maneira voltada para serviço. Dependências entre serviços como serviços da web, ativos de serviço do Enterprise Information System (EIS), fluxos de trabalho e bancos de dados são minimizados e a implementação de qualquer serviço é oculta.

O objetivo da arquitetura orientada a serviços é separar a lógica de integração de negócios da implementação para que um desenvolvedor de integração possa focar na montagem de um aplicativo integrado em vez de nos detalhes da implementação. Para alcançar esse objetivo, os componentes de serviço que contêm a implementação de serviços individuais requeridos pelos processos de negócios são criados.”

Fonte: https://www.ibm.com/support/knowledgecenter/pt-br/SSFPJS_8.6.0/com.ibm.wbpm.wid.main.doc/prodoverview/topics/csoa.html

SOA Manifesto

Service orientation is a paradigm that frames what you do.
Service-oriented architecture (SOA) is a type of architecture
that results from applying service orientation.

We have been applying service orientation to help organizations
consistently deliver sustainable business value, with increased agility
and cost effectiveness, in line with changing business needs.

Through our work we have come to prioritize:

Business value over technical strategy

Strategic goals over project-specific benefits

Intrinsic interoperability over custom integration

Shared services over specific-purpose implementations

Flexibility over optimization

Evolutionary refinement over pursuit of initial perfection

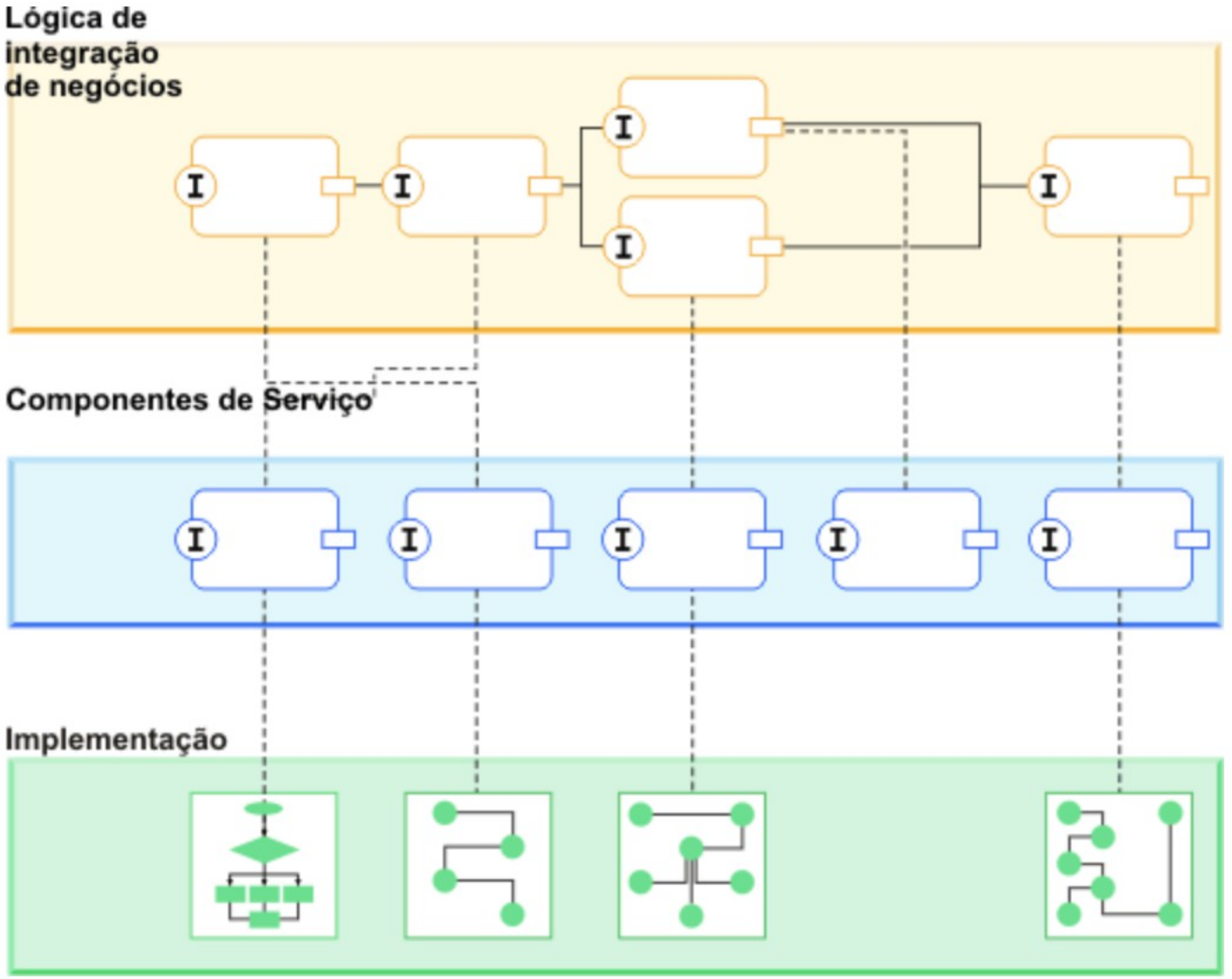
That is, while we value the items on the right, we value the items on the left more.

Fonte: http://www.soa-manifesto.org/

SOA


Lógica de
integração
de negócios
Componentes de Serviço
Implementação

SOA


Lógica de
integração
de negócios
Componentes de Serviço
Implementação

SOA

ServiceOrientedAmbiguity

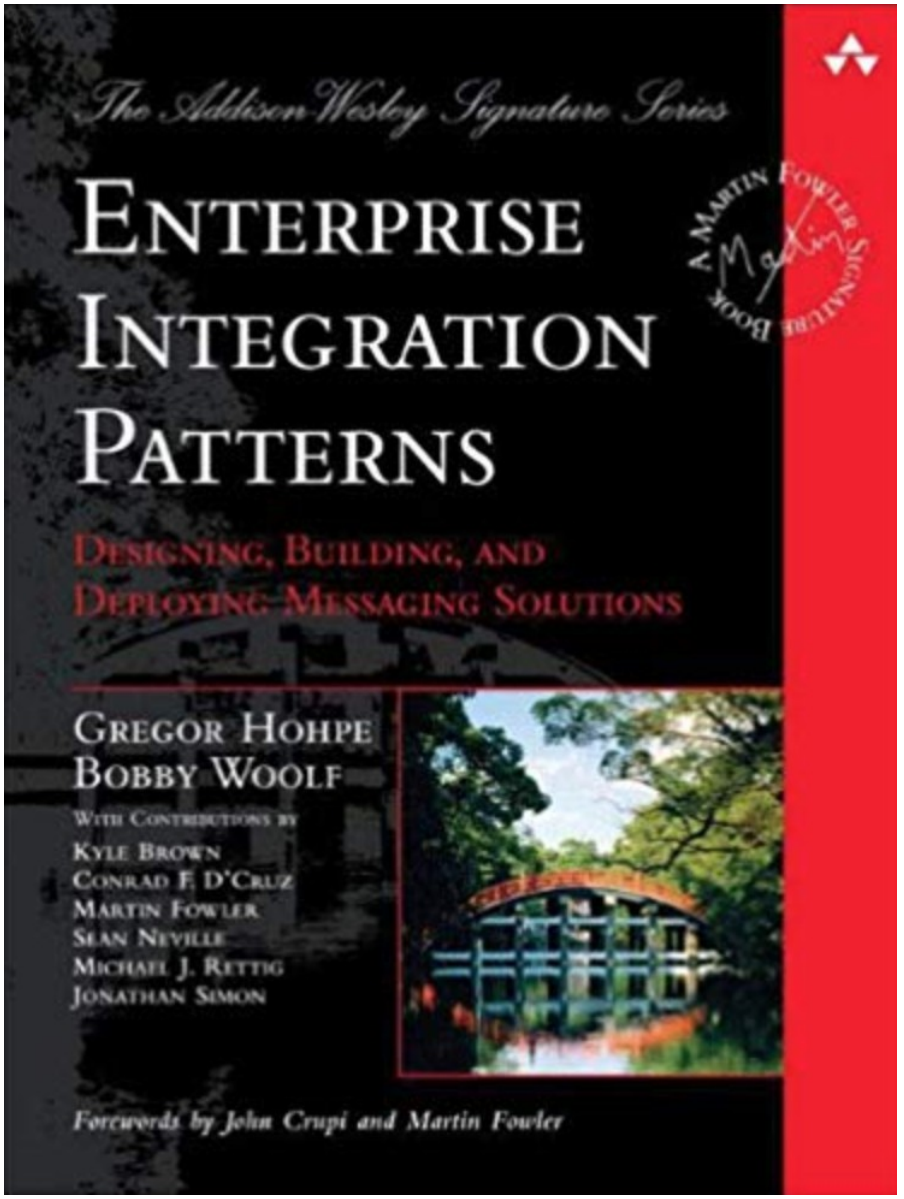
Martin Fowler

1 July 2005

Whenever ThoughtWorks rashly lets me out in front of a client, one question I'm bound to be asked is "what do you think of SOA (Service Oriented Architecture)?" It's a question that's pretty much impossible to answer because SOA means so many different things to different people.

- For some SOA is about exposing software through web services. This crowd further sub-divides into those that expect the various WS-* standards and those that will accept any form of XML over http (and maybe not even XML).
- For some SOA implies an architecture where applications disappear. Instead you have core services that supply business functionality and data separated by UI aggregators that apply presentations that aggregate together the stuff that core services provide.
- For some SOA is about allowing systems to communicate over some form of standard structure (usually XML based) with other applications. In it's worse form this is "CORBA with angle brackets". In more sophisticated forms this involves coming up with some form of standard backbone for an organization and getting applications to work with this. This backbone may or may not involve http.
- For some SOA is all about using (mostly) asynchronous messaging to transfer documents between different systems. Essentially this is EAI without all the expensive EAI vendors locking you in.

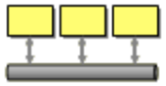
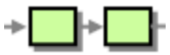
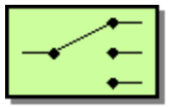
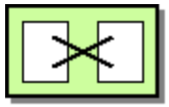
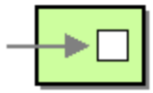
Estilos de integração



Estilos de integração

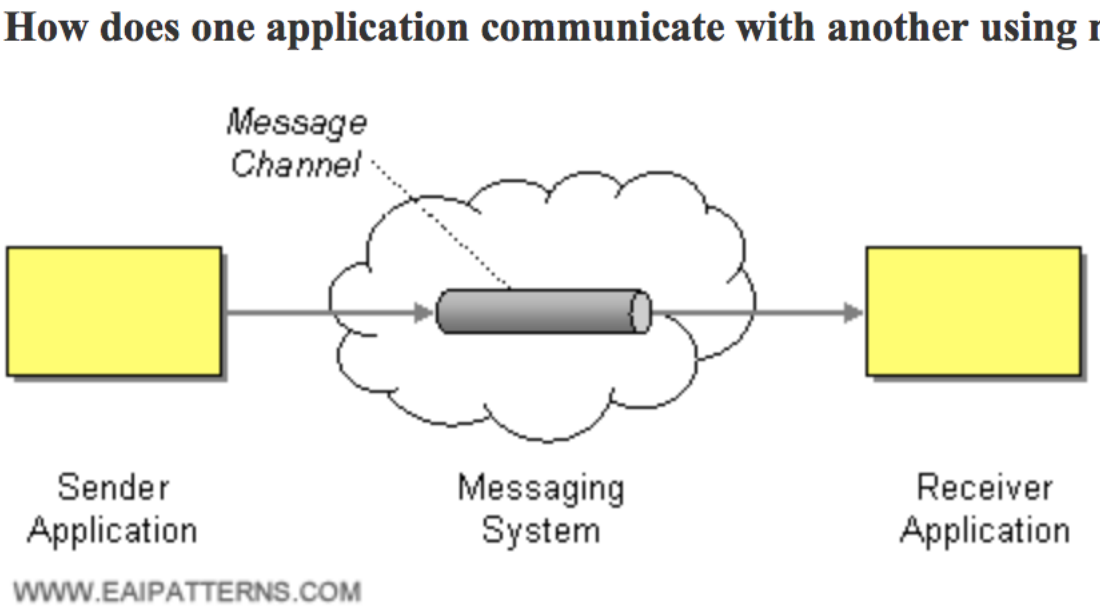
- Transferência de arquivos;
- Banco de Dados compartilhado;
- RPC;
- Mensageria.

Sistemas de mensagem

Messaging Systems		
	Introduction to Messaging Systems	
	Message Channel	How does one application communicate with another using messaging?
	Message	How can two applications connected by a message channel exchange a piece of information?
	Pipes and Filters	How can we perform complex processing on a message while maintaining independence and flexibility?
	Message Router	How can you decouple individual processing steps so that messages can be passed to different filters depending on a set of conditions?
	Message Translator	How can systems using different data formats communicate with each other using messaging?
	Message Endpoint	How does an application connect to a messaging channel to send and receive messages?

Sistemas de mensagem

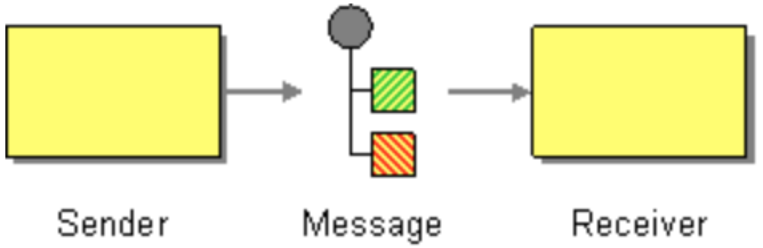
Message channel



Sistemas de mensagem

Message

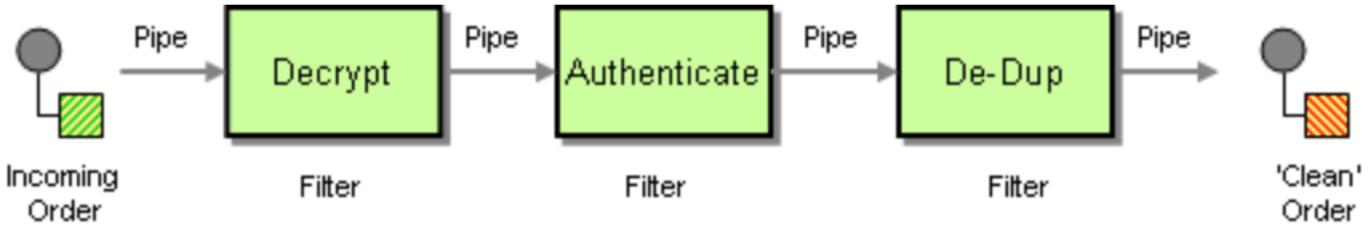
How can two applications connected by a message channel exchange a piece of information?



Sistemas de mensagem

Pipe and Filters

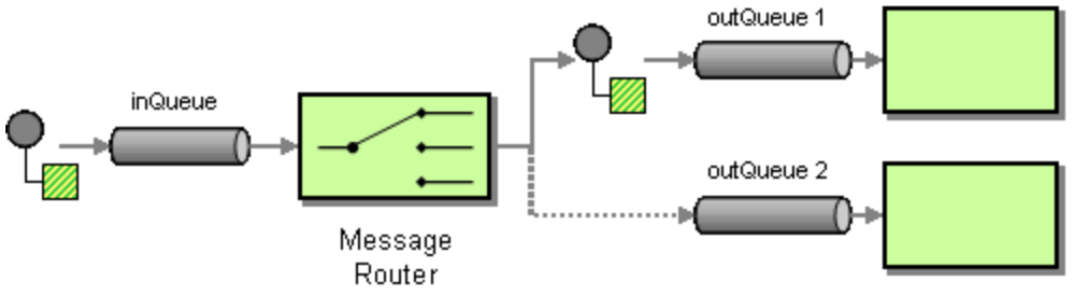
How can we perform complex processing on a message while maintaining independence and flexibility?



Sistemas de mensagem

Message Router

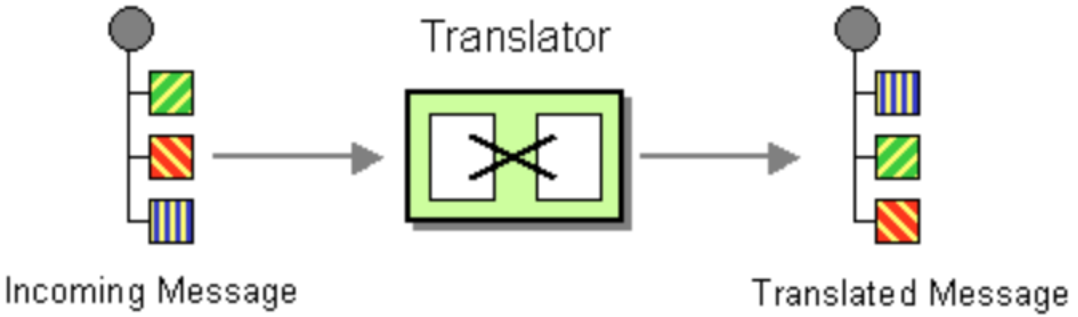
How can you decouple individual processing steps so that messages can be passed to different filters depending on a set of conditions?



Sistemas de mensagem

Message Translator

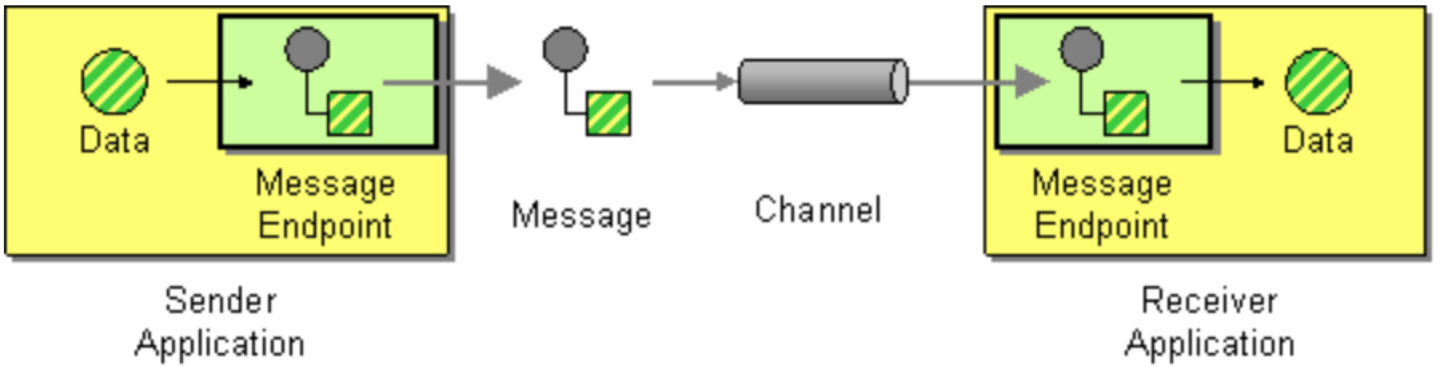
How can systems using different data formats communicate with each other using messaging?



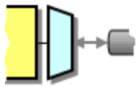
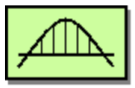
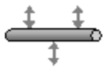
Sistemas de mensagem

Message Endpoint

How does an application connect to a messaging channel to send and receive messages?



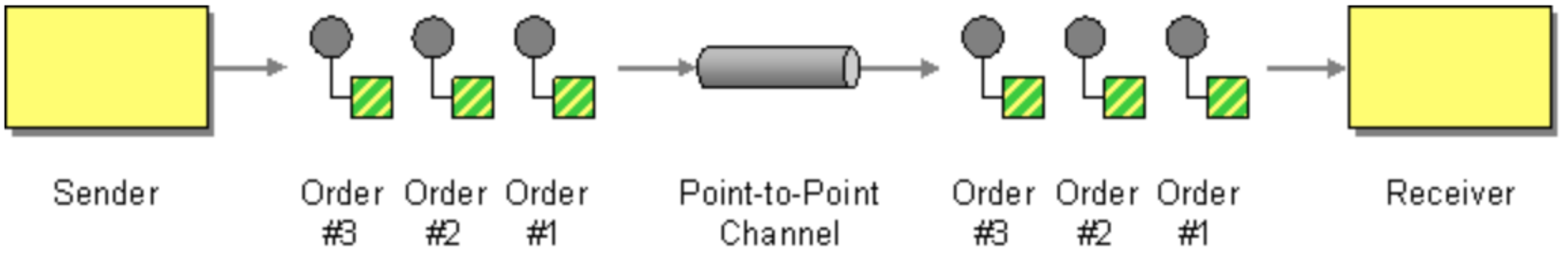
Canais de mensagens

Messaging Channels		
	Introduction to Messaging Channels	
	Point-to-Point Channel	How can the caller be sure that exactly one receiver will receive the document or perform the call?
	Publish-Subscribe Channel	How can the sender broadcast an event to all interested receivers?
	Datatype Channel	How can the application send a data item such that the receiver will know how to process it?
	Invalid Message Channel	How can a messaging receiver gracefully handle receiving a message that makes no sense?
	Dead Letter Channel	What will the messaging system do with a message it cannot deliver?
	Guaranteed Delivery	How can the sender make sure that a message will be delivered, even if the messaging system fails?
	Channel Adapter	How can you connect an application to the messaging system so that it can send and receive messages?
	Messaging Bridge	How can multiple messaging systems be connected so that messages available on one are also available on the others?
	Message Bus	What is an architecture that enables separate applications to work together, but in a decoupled fashion such that applications can be easily added or removed without affecting the others?

Canais de mensagens

Point-to-Point

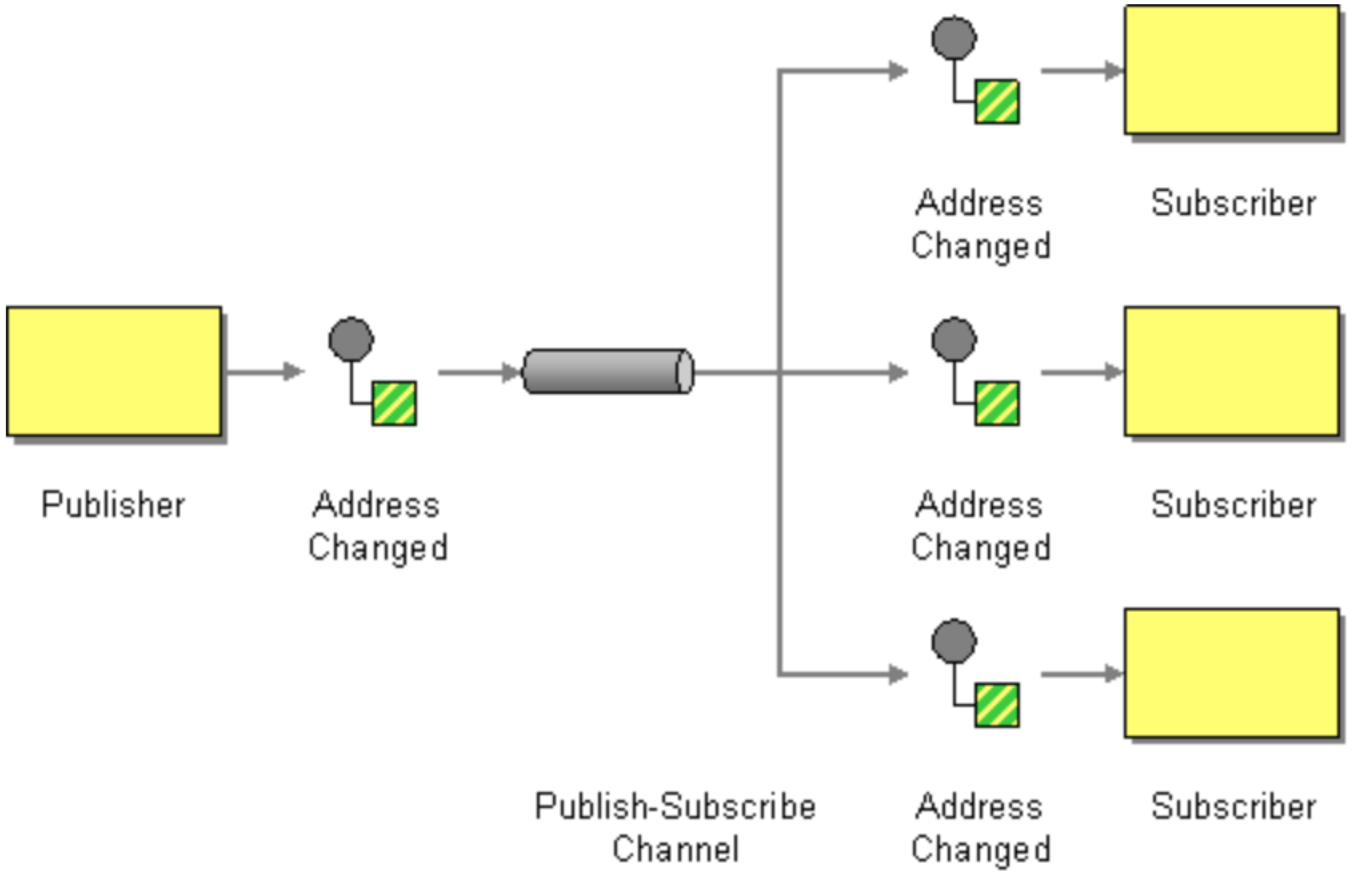
How can the caller be sure that exactly one receiver will receive the document or perform the call?



Canais de mensagens

Publish-Subscriber

How can the sender broadcast an event to all interested receivers?

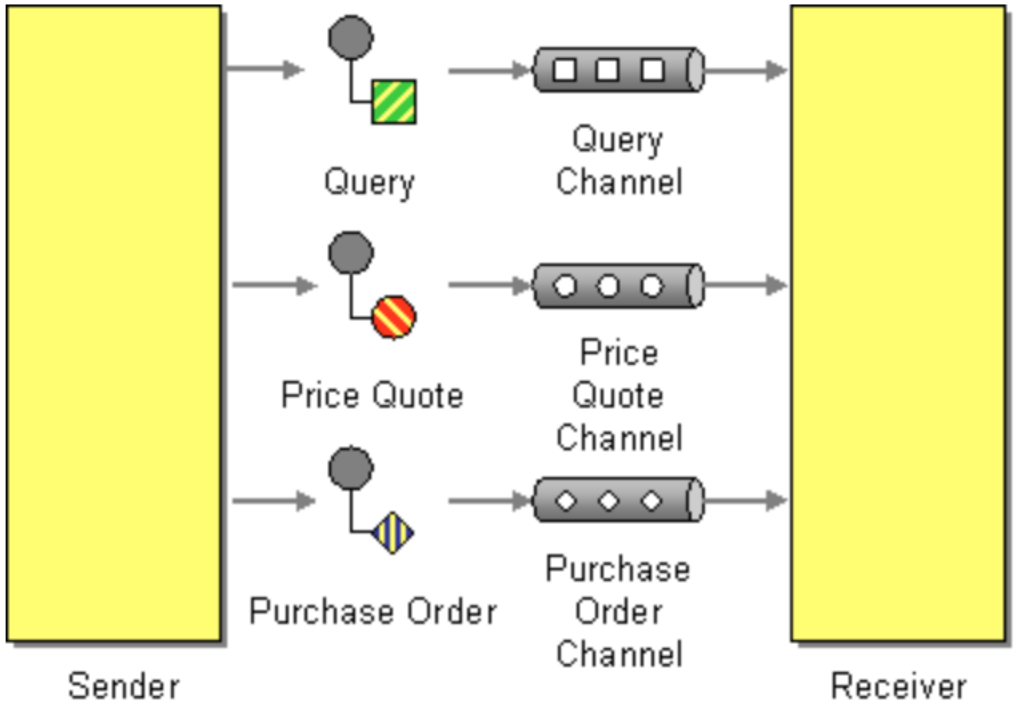


Send the event on a *Publish-Subscribe Channel*, which delivers a copy of a particular event to each receiver.

Canais de mensagens

Data-type channel

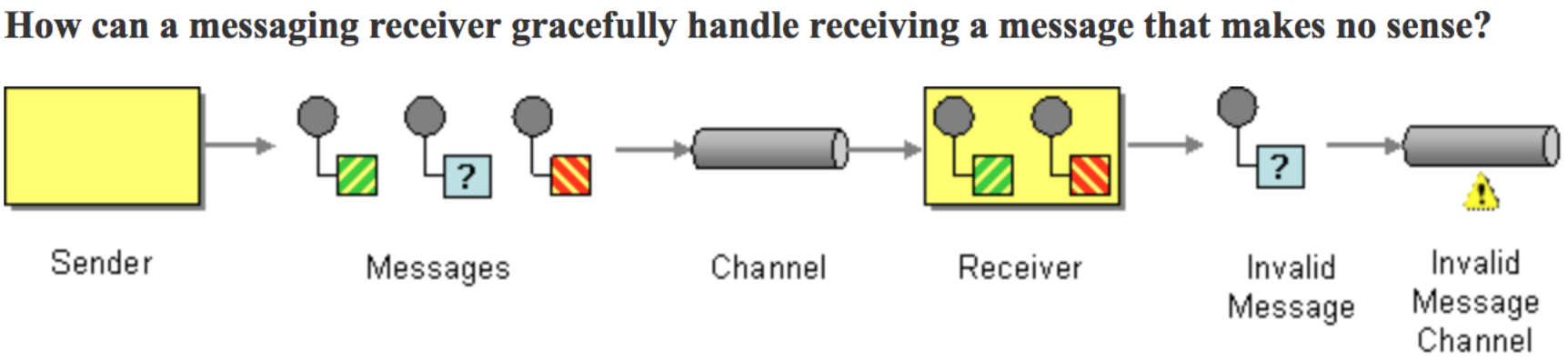
How can the application send a data item such that the receiver will know how to process it?



Use a separate *Datatype Channel* for each data type, so that all data on a particular channel is of the same type.

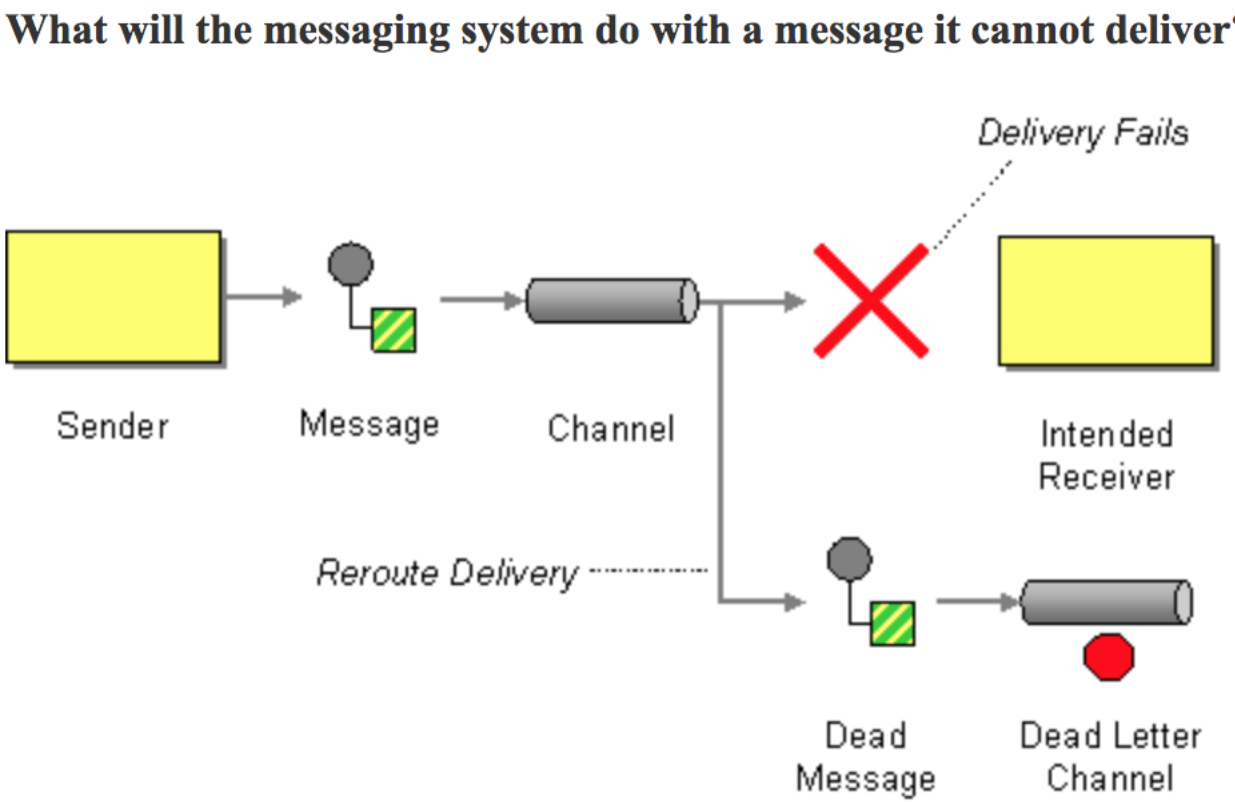
Canais de mensagens

Invalid message channel



Canais de mensagens

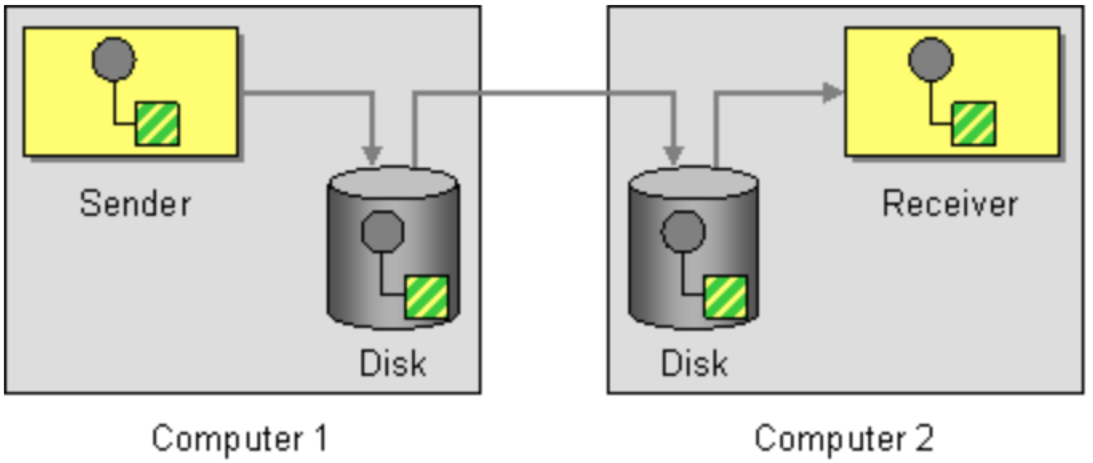
Dead Letter channel



Canais de mensagens

Guaranteed Delivery

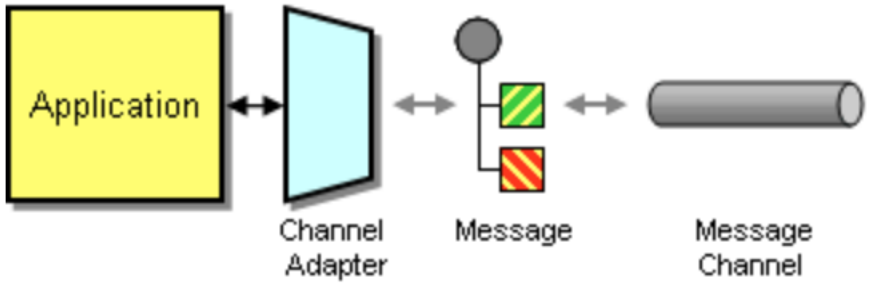
How can the sender make sure that a message will be delivered, even if the messaging system fails?



Canais de mensagens

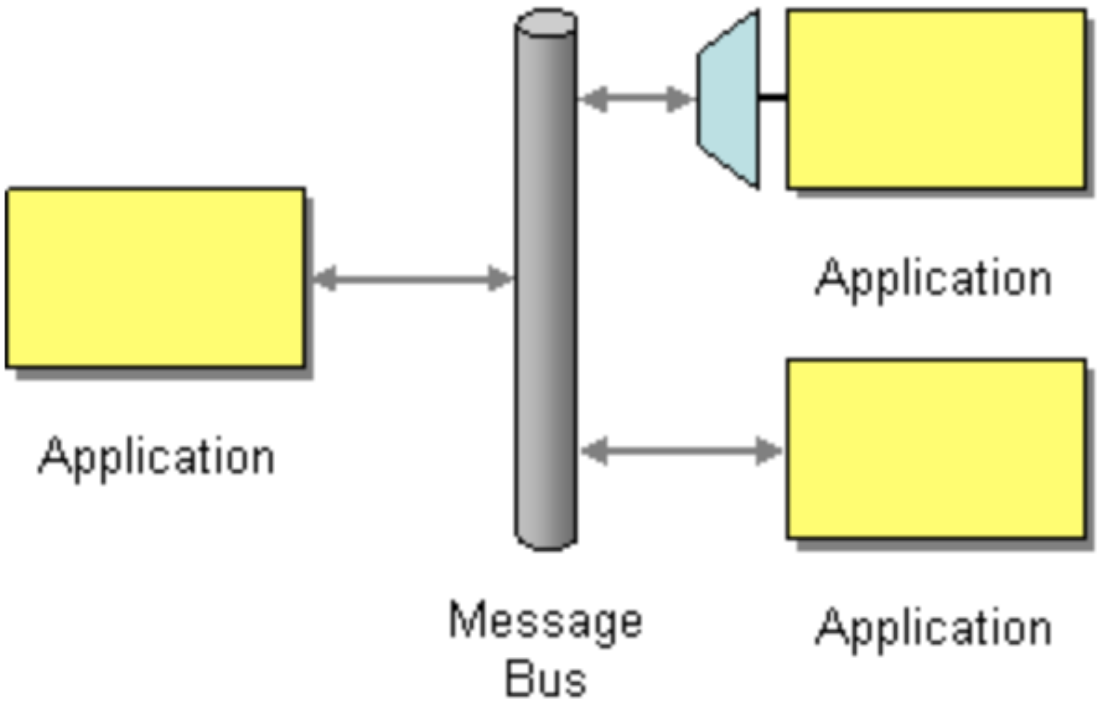
Channel Adapter

How can you connect an application to the messaging system so that it can send and receive messages?

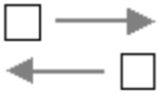
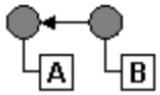
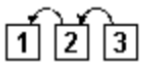


Canais de mensagens

Message Bus



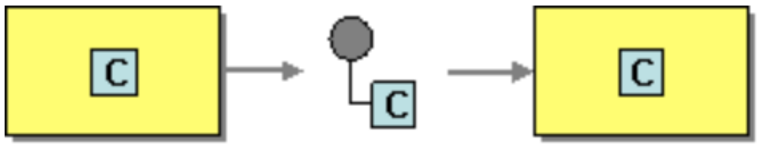
Construção de mensagens

Message Construction		
	Introduction to Message Construction	
	Command Message	How can messaging be used to invoke a procedure in another application?
	Document Message	How can messaging be used to transfer data between applications?
	Event Message	How can messaging be used to transmit events from one application to another?
	Request-Reply	When an application sends a message, how can it get a response from the receiver?
	Return Address	How does a replier know where to send the reply?
	Correlation Identifier	How does a requestor that has received a reply know which request this is the reply for?
	Message Sequence	How can messaging transmit an arbitrarily large amount of data?
	Message Expiration	How can a sender indicate when a message should be considered stale and thus shouldn't be processed?
	Format Indicator	How can a message's data format be designed to allow for possible future changes?

Construção de mensagens

Command Message

How can messaging be used to invoke a procedure in another application?



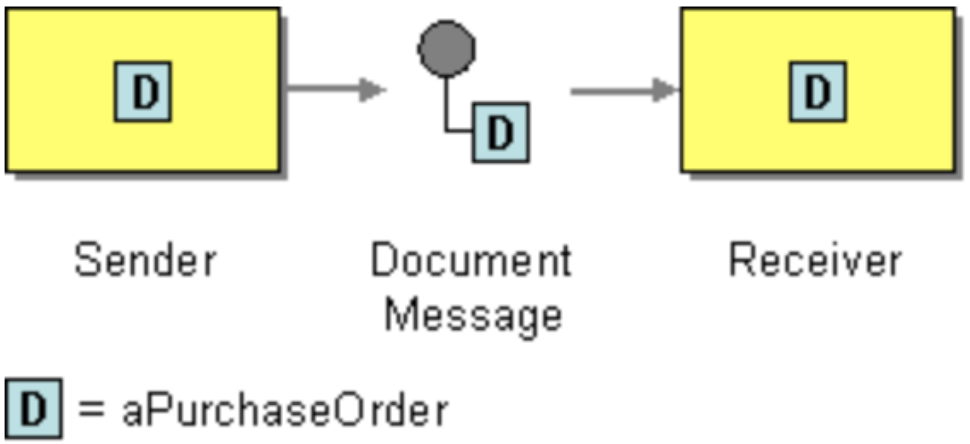
C = getLastTradePrice("DIS");

Use a *Command Message* to reliably invoke a procedure in another application.

Construção de mensagens

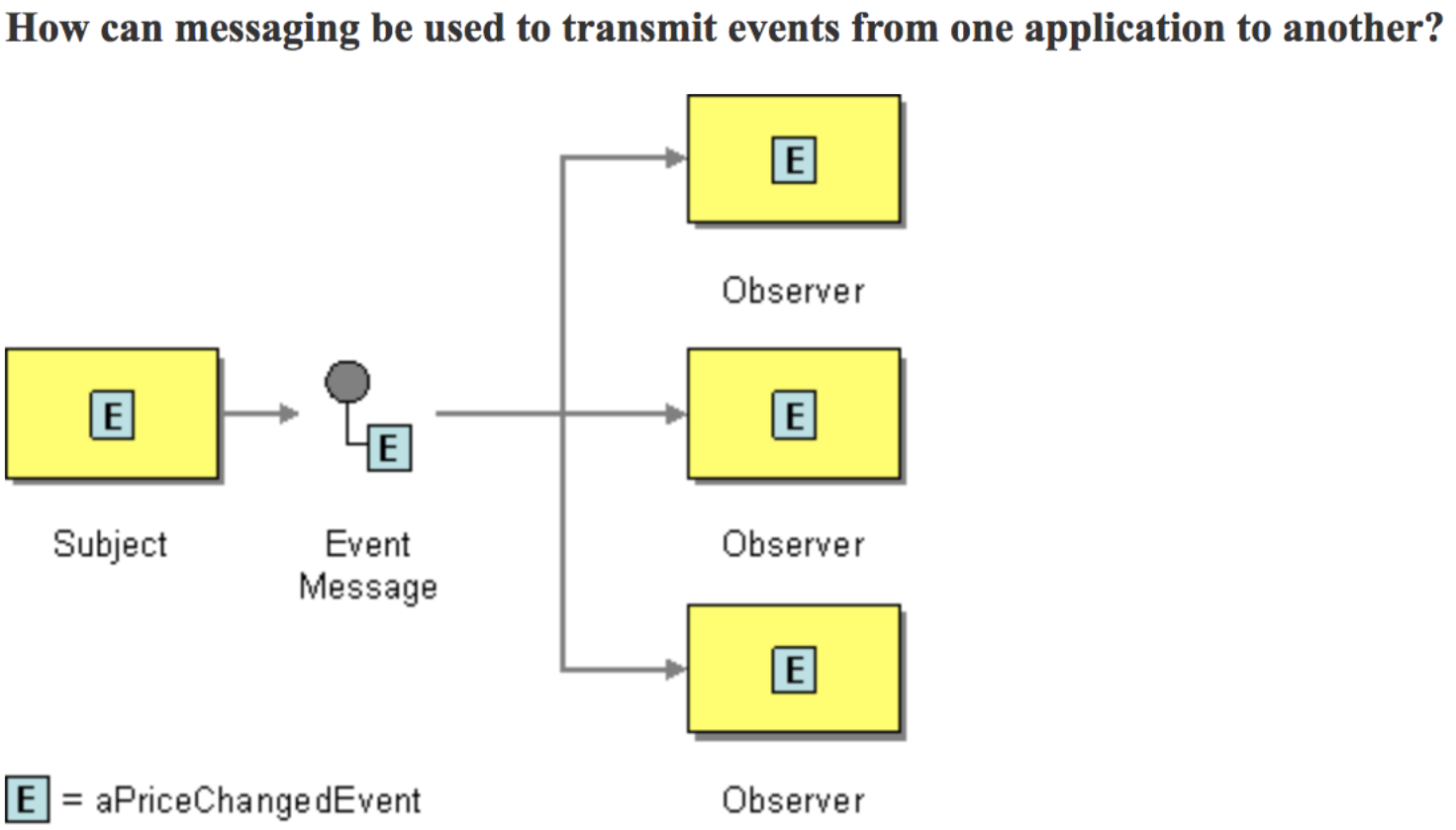
Document Message

How can messaging be used to transfer data between applications?



Construção de mensagens

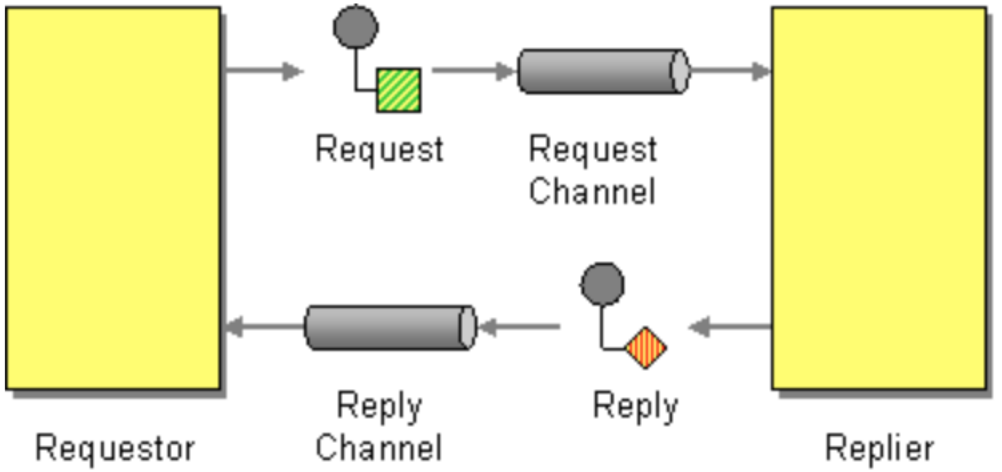
Event Message



Construção de mensagens

Request/Reply

When an application sends a message, how can it get a response from the receiver?

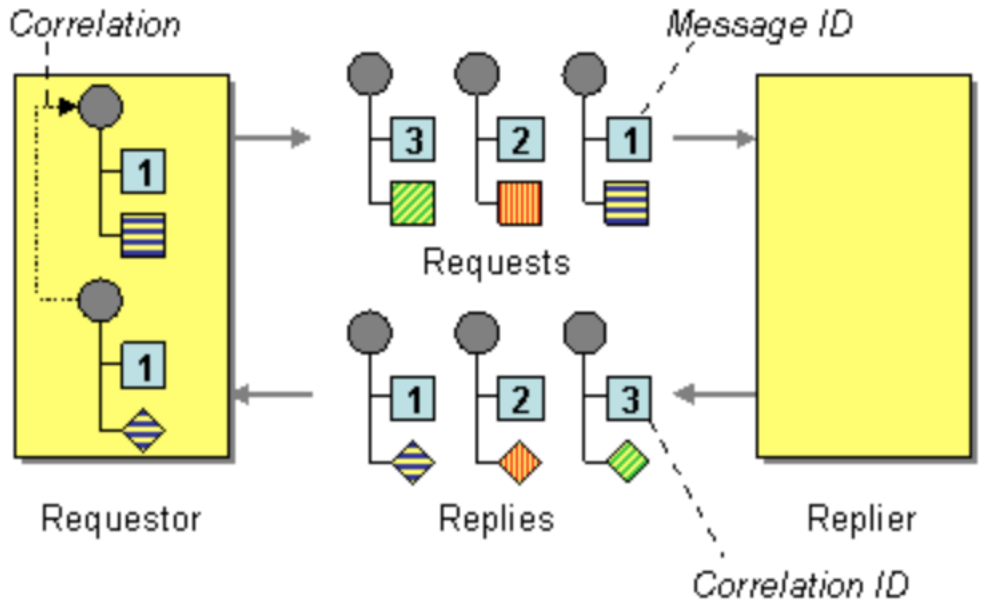


Send a pair of *Request-Reply* messages, each on its own channel.

Construção de mensagens

Correlation Identifier

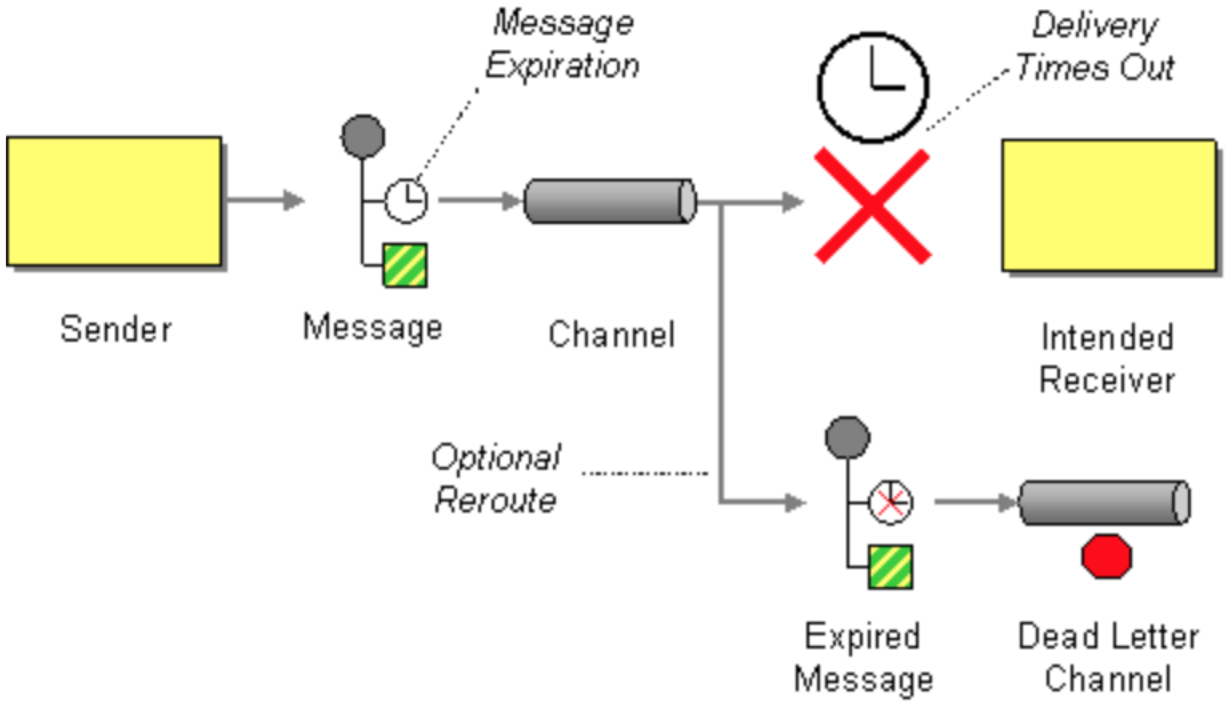
How does a requestor that has received a reply know which request this is the reply for?



Construção de mensagens

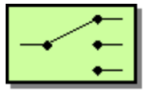
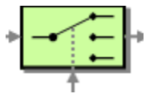
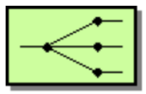
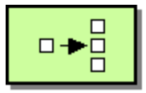
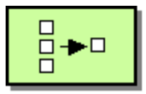
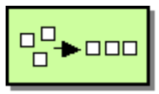
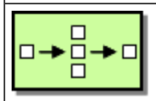
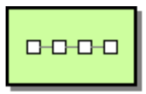
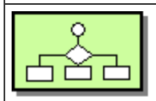
Message Expiration

How can a sender indicate when a message should be considered stale and thus shouldn't be processed?



Set the *Message Expiration* to specify a time limit how long the message is viable.

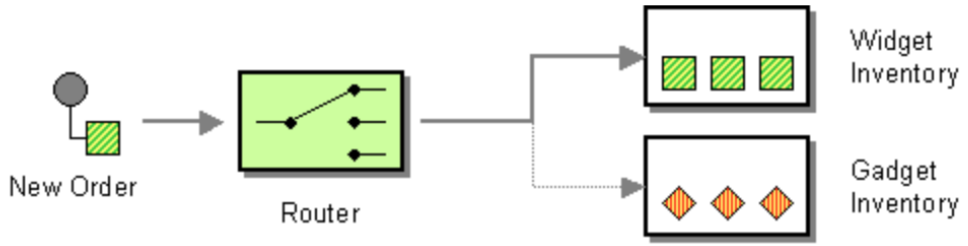
Roteamento de mensagens

Message Routing		
	Introduction to Message Routing	
	Content-Based Router	How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?
	Message Filter	How can a component avoid receiving uninteresting messages?
	Dynamic Router	How can you avoid the dependency of the router on all possible destinations while maintaining its efficiency?
	Recipient List	How do we route a message to a list of dynamically specified recipients?
	Splitter	How can we process a message if it contains multiple elements, each of which may have to be processed in a different way?
	Aggregator	How do we combine the results of individual, but related messages so that they can be processed as a whole?
	Resequencer	How can we get a stream of related but out-of-sequence messages back into the correct order?
	Composed Message Processor	How can you maintain the overall message flow when processing a message consisting of multiple elements, each of which may require different processing?
	Scatter-Gather	How do you maintain the overall message flow when a message needs to be sent to multiple recipients, each of which may send a reply?
	Routing Slip	How do we route a message consecutively through a series of processing steps when the sequence of steps is not known at design-time and may vary for each message?
	Process Manager	How do we route a message through multiple processing steps when the required steps may not be known at design-time and may not be sequential?
	Message Broker	How can you decouple the destination of a message from the sender and maintain central control over the flow of messages?

Roteamento de mensagens

Content-based

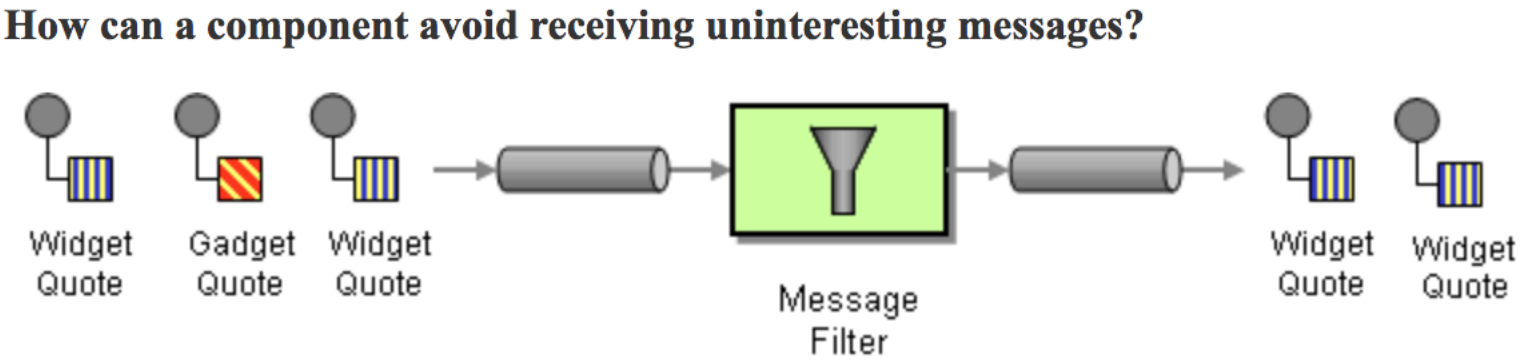
How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?



Use a *Content-Based Router* to route each message to the correct recipient based on message content.

Roteamento de mensagens

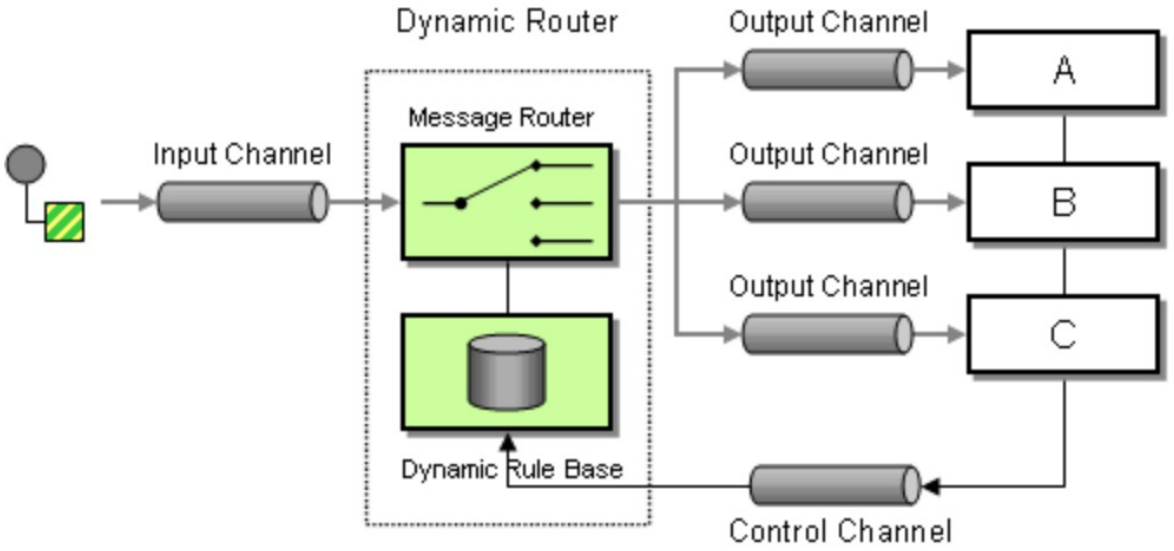
Message filter



Roteamento de mensagens

Dynamic router

How can you avoid the dependency of the router on all possible destinations while maintaining its efficiency?

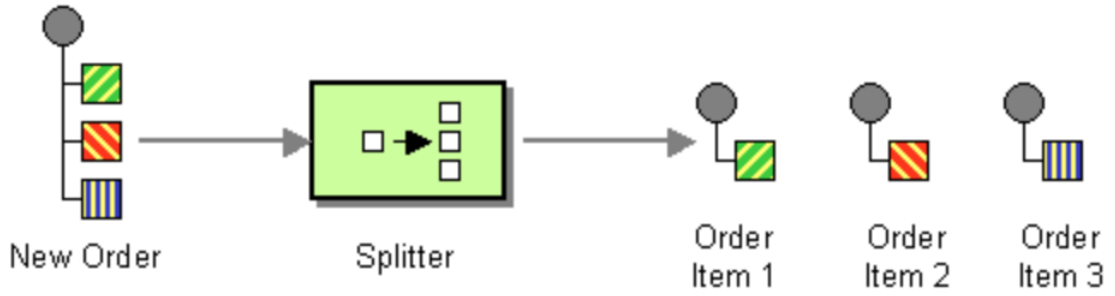


Use a *Dynamic Router*, a Router that can self-configure based on special configuration messages from participating destinations.

Roteamento de mensagens

Splitter

How can we process a message if it contains multiple elements, each of which may have to be processed in a different way?

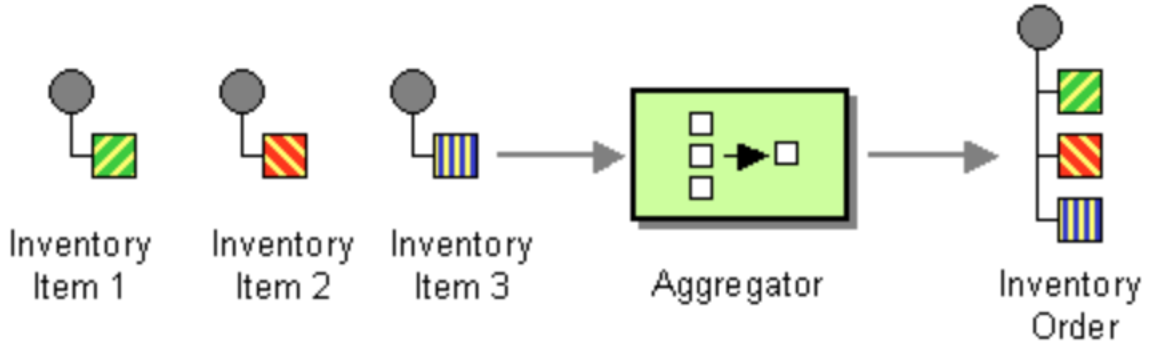


Use a *Splitter* to break out the composite message into a series of individual messages, each containing data related to one item.

Roteamento de mensagens

Aggregator

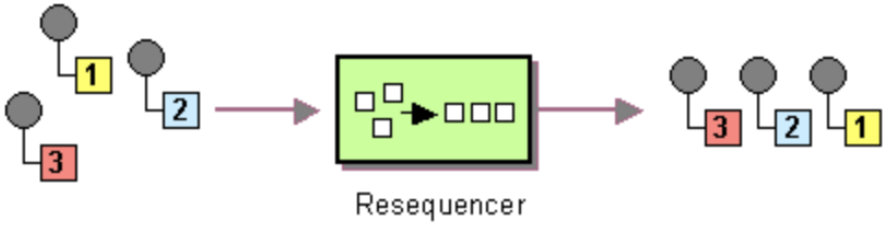
How do we combine the results of individual, but related messages so that they can be processed as a whole?



Roteamento de mensagens

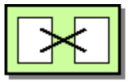
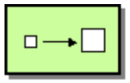
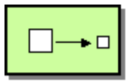
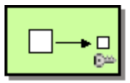
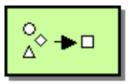
Resequencer

How can we get a stream of related but out-of-sequence messages back into the correct order?

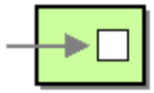
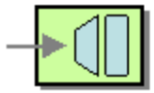
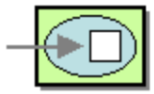
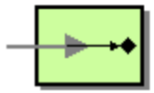
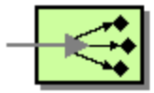
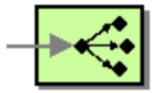
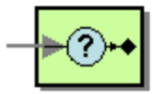
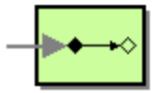


Use a stateful filter, a *Resequencer*, to collect and re-order messages so that they can be published to the output channel in a specified order.

Transformaçao de mensagens

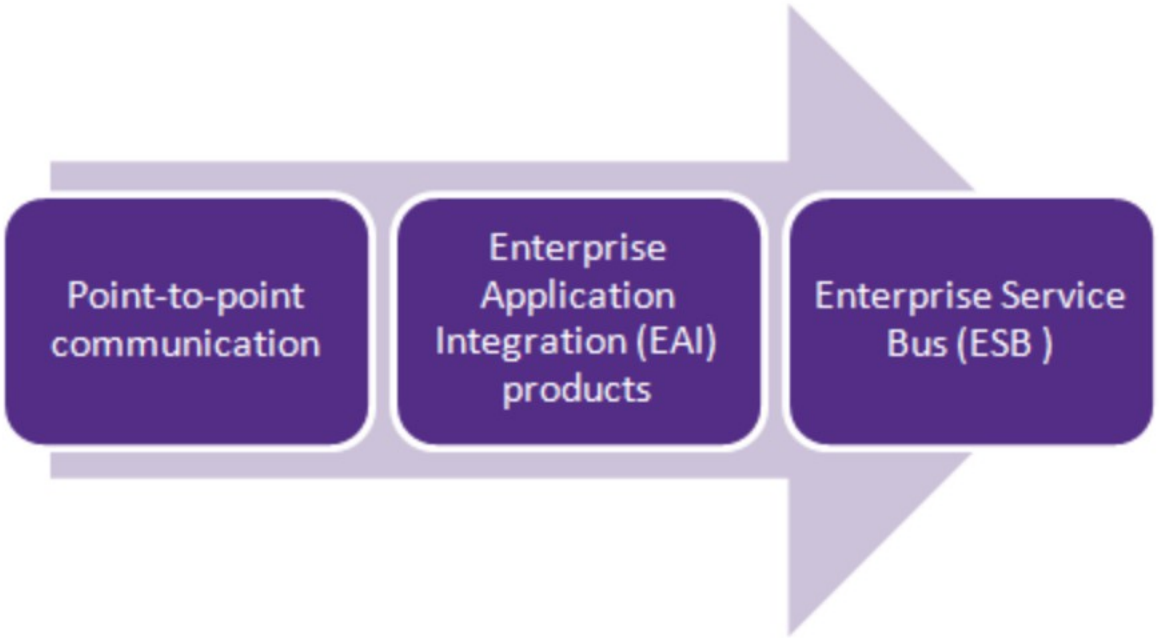
Message Transformation		
	Introduction to Message Transformation	
	Envelope Wrapper	How can existing systems participate in a messaging exchange that places specific requirements on the message format, such as message header fields or encryption?
	Content Enricher	How do we communicate with another system if the message originator does not have all the required data items available?
	Content Filter	How do you simplify dealing with a large message, when you are interested only in a few data items?
	Claim Check	How can we reduce the data volume of message sent across the system without sacrificing information content?
	Normalizer	How do you process messages that are semantically equivalent, but arrive in a different format?
	Canonical Data Model	How can you minimize dependencies when integrating applications that use different data formats?

Endpoints

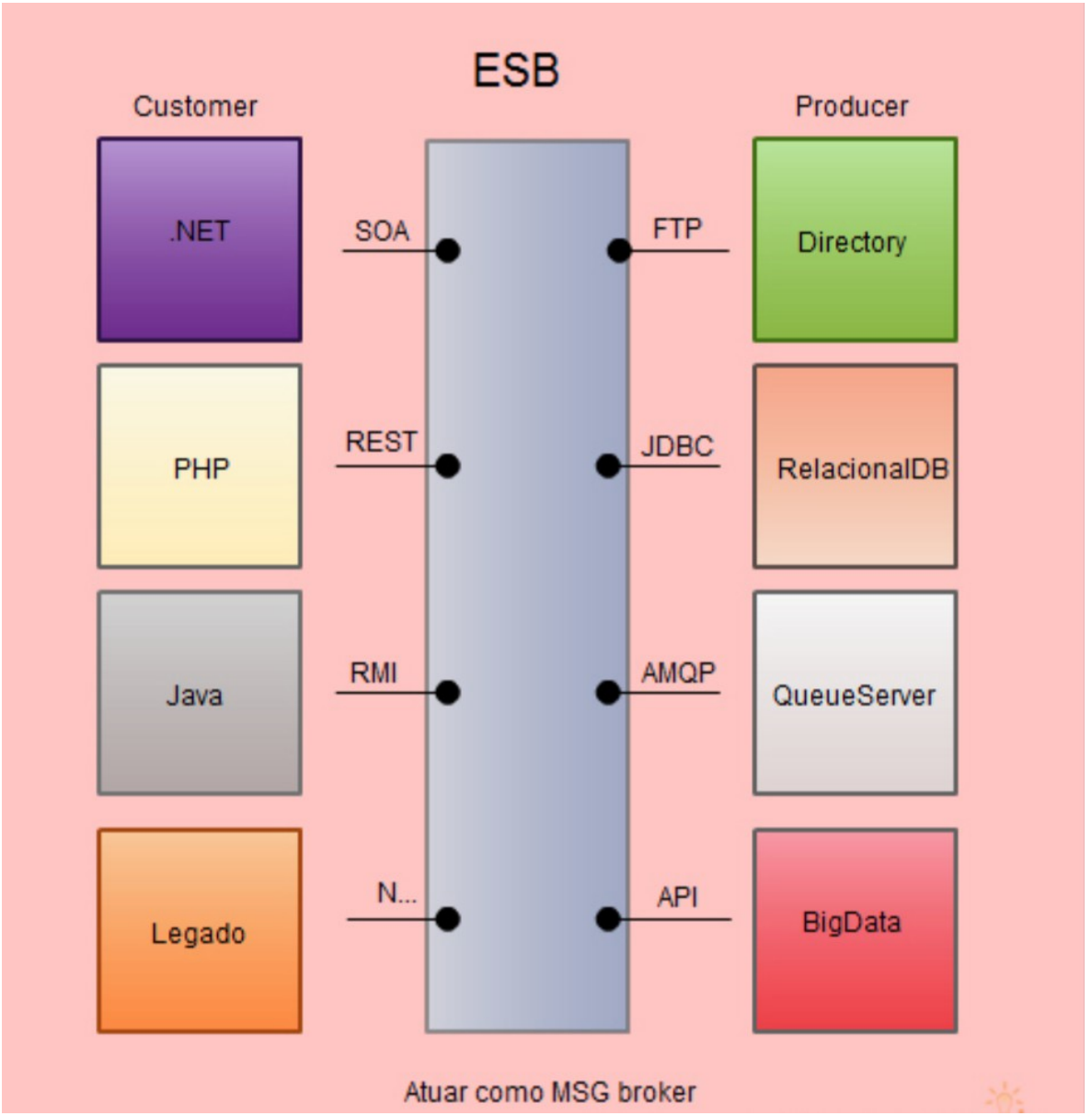
Messaging Endpoints		
	Introduction to Messaging Endpoints	
	Messaging Gateway	How do you encapsulate access to the messaging system from the rest of the application?
	Messaging Mapper	How do you move data between domain objects and the messaging infrastructure while keeping the two independent of each other?
	Transactional Client	How can a client control its transactions with the messaging system?
	Polling Consumer	How can an application consume a message when the application is ready?
	Event-Driven Consumer	How can an application automatically consume messages as they become available?
	Competing Consumers	How can a messaging client process multiple messages concurrently?
	Message Dispatcher	How can multiple consumers on a single channel coordinate their message processing?
	Selective Consumer	How can a message consumer select which messages it wishes to receive?
	Durable Subscriber	How can a subscriber avoid missing messages while it's not listening for them?
	Idempotent Receiver	How can a message receiver deal with duplicate messages?
	Service Activator	How can an application design a service to be invoked both via various messaging technologies and via non-messaging techniques?

Implementações

ESB


Point-to-point communication
Enterprise Application Integration (EAI) products
Enterprise Service Bus (ESB)

ESB

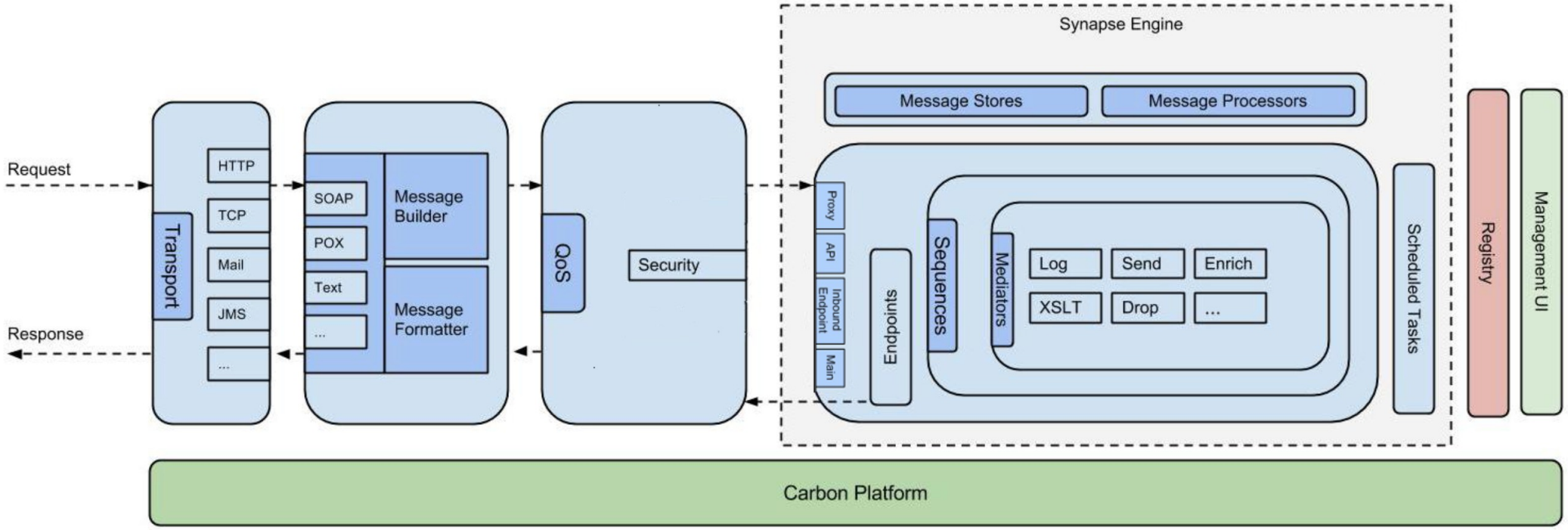

ESB
Customer
Producer
.NET
PHP
Java
Legado
SOA
REST
RMI
N...
FTP
JDBC
AMQP
API
Directory
RelacionalDB
QueueServer
BigData
Atuar como MSG broker

ESB

- WSO2 ESB:
  - Download:
    - https://wso2.com/integration/previous-releases utm_source=esb_page&utm_medium=esb_page& utm_campaign=esb_page
    - Versão 5.0.0

ESB

Arquitetura WSO2 ESB



ESB

- WSO2 ESB:
  - Após feito o download e a descompactação, entre no diretório raiz do WSO2 ESB, depois no diretório bin e execute:
    - ./wso2server.sh (linux) ou wso2server.bat (windows)
    - Exemplo: $WSO2_ESB_HOME/bin/wso2server.sh

ESB

WSO2 Documentation
Training
Community

Search this documentation

Expand all Collapse all

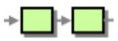
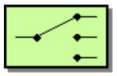
- Setting up the Environment
- Messaging Systems
- Messaging Channels
- Message Construction
- Message Routing
- Message Transformation
- Messaging Endpoints
- System Management

Enterprise Integration Patterns with WSO2 ESB

Enterprise Application Integration (EAI) is key to connecting business applications with heterogeneous systems. Over the years, architects of integration solutions have invented their own blend of patterns in a variety of ways. But most of these architectures have similarities, initiating a set of widely accepted standards in architecting integration patterns. Most of these standards are described in the **Enterprise Integration Patterns Catalog** available at: http://www.eaipatterns.com/toc.html.

In this guide, we have shown how each pattern in the patterns catalog can be simulated using various constructs in **WSO2 ESB 4.9.0**. Click on a topic in the list below for details.

Messaging Systems

	Message Channels	How one application communicates with another using messaging.
	Message	How two applications connected by a message channel exchange a piece of information.
	Pipes and Filters	How to perform complex processing on a message while maintaining independence and flexibility.
	Message Router	How to decouple individual processing steps so that messages can be passed to different filters depending on conditions.

Evoluções

Evolução Tecnológica dos Últimos 50 anos

- Redes Sociais, virtualização.
- Internet of Everything

- Computação Distribuídas
- Redes de Computadores

- Browsers Netscape, Internet Explorer, Google
- E-mail, E-commerce, , Wi-Fi, Web 2.0, Hosting

- Pacote MS Office, Windows
- Computação "stand-alone".

- Empresas de Grande Porte;
- Computação Centralizada.

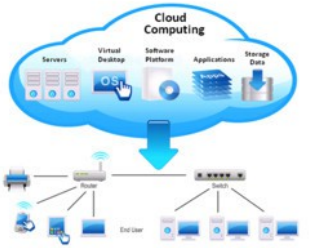
Mainframe
- Anos 60 e 70

PC e Aplicações
- 1980 a 1985

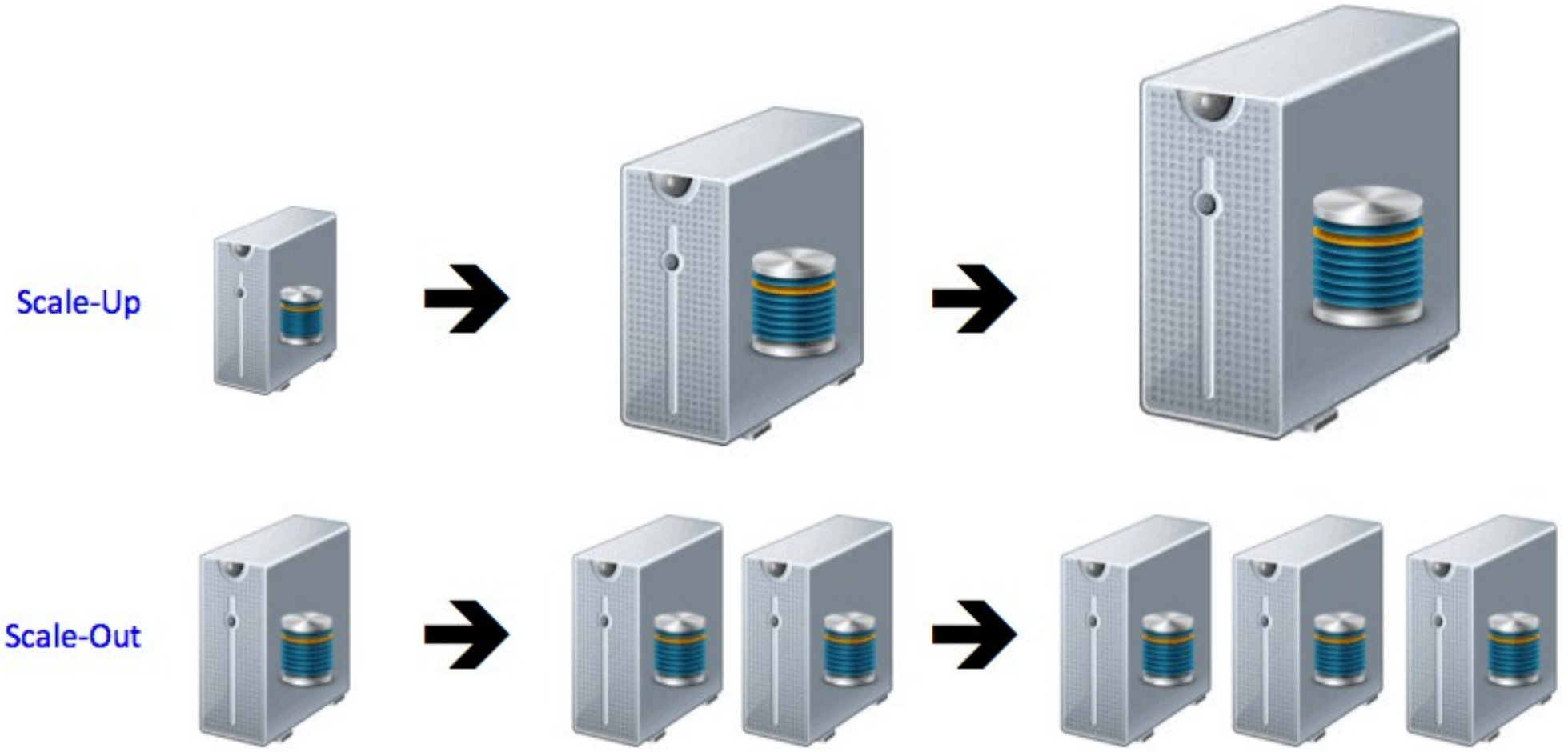
Client / Server
- 1985 a 1995

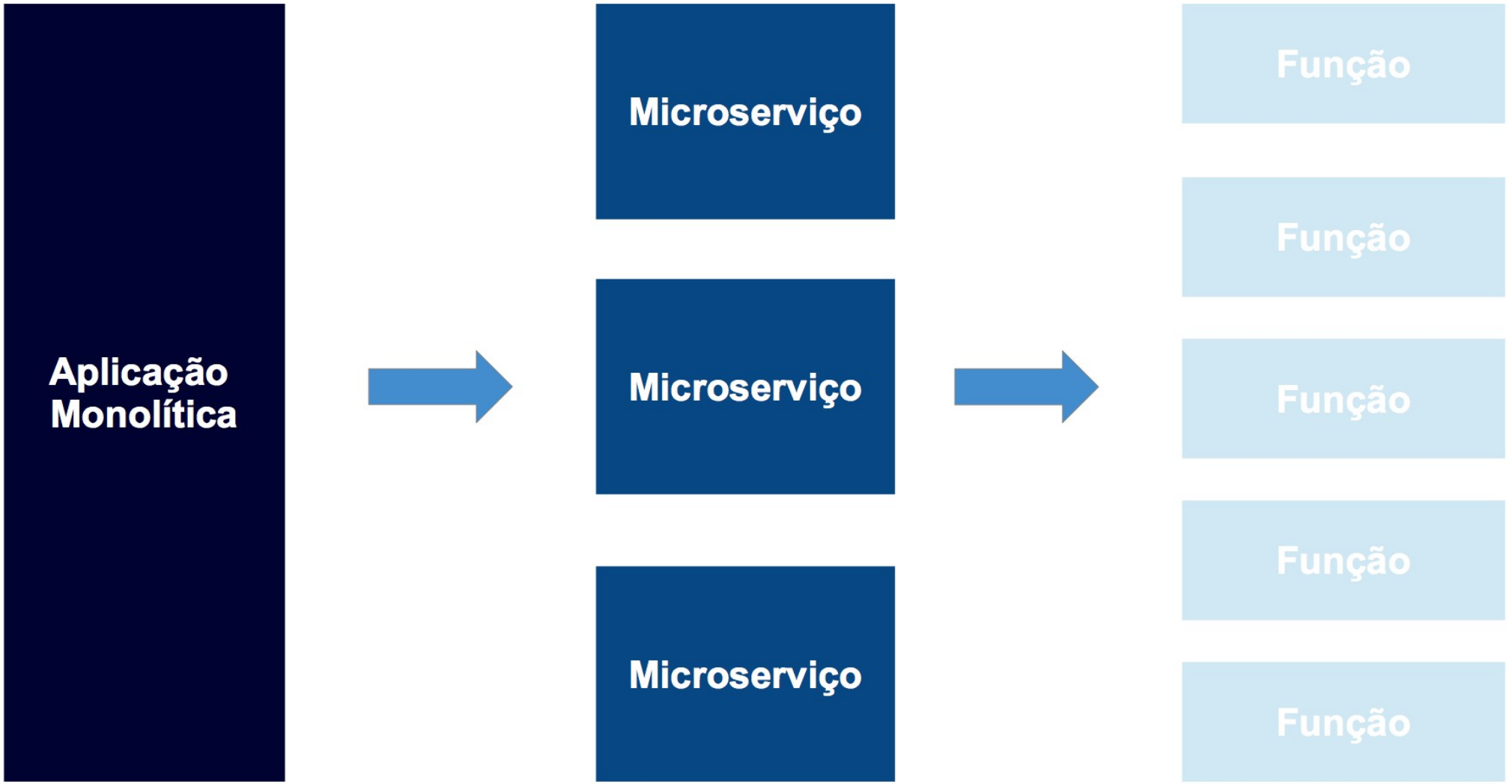
Internet
- Após 1995 Inícios dos anos 2000

Cloud Computing
- Dias atuais



Escalabilidade

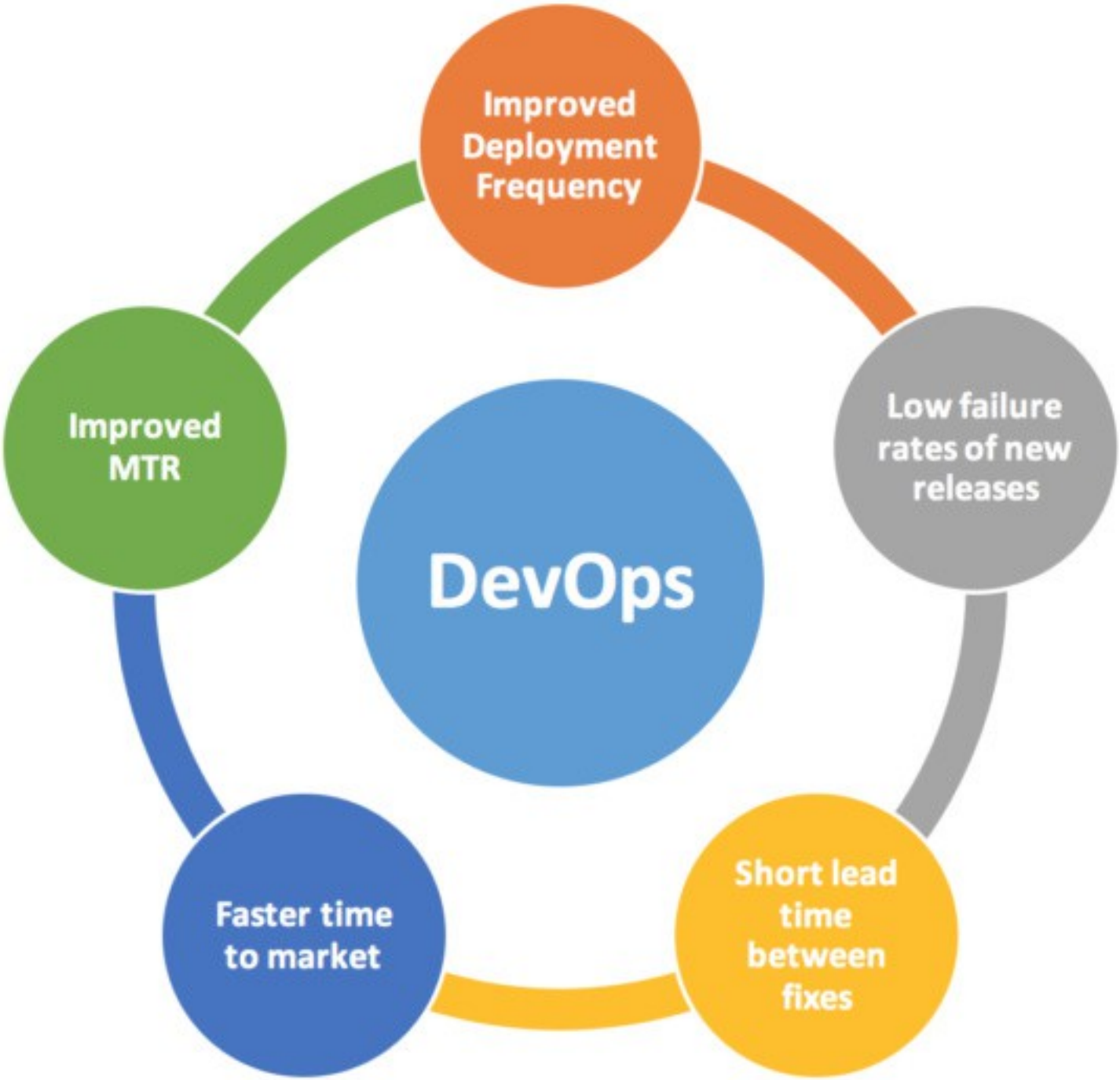

Scale-Up
Scale-Out

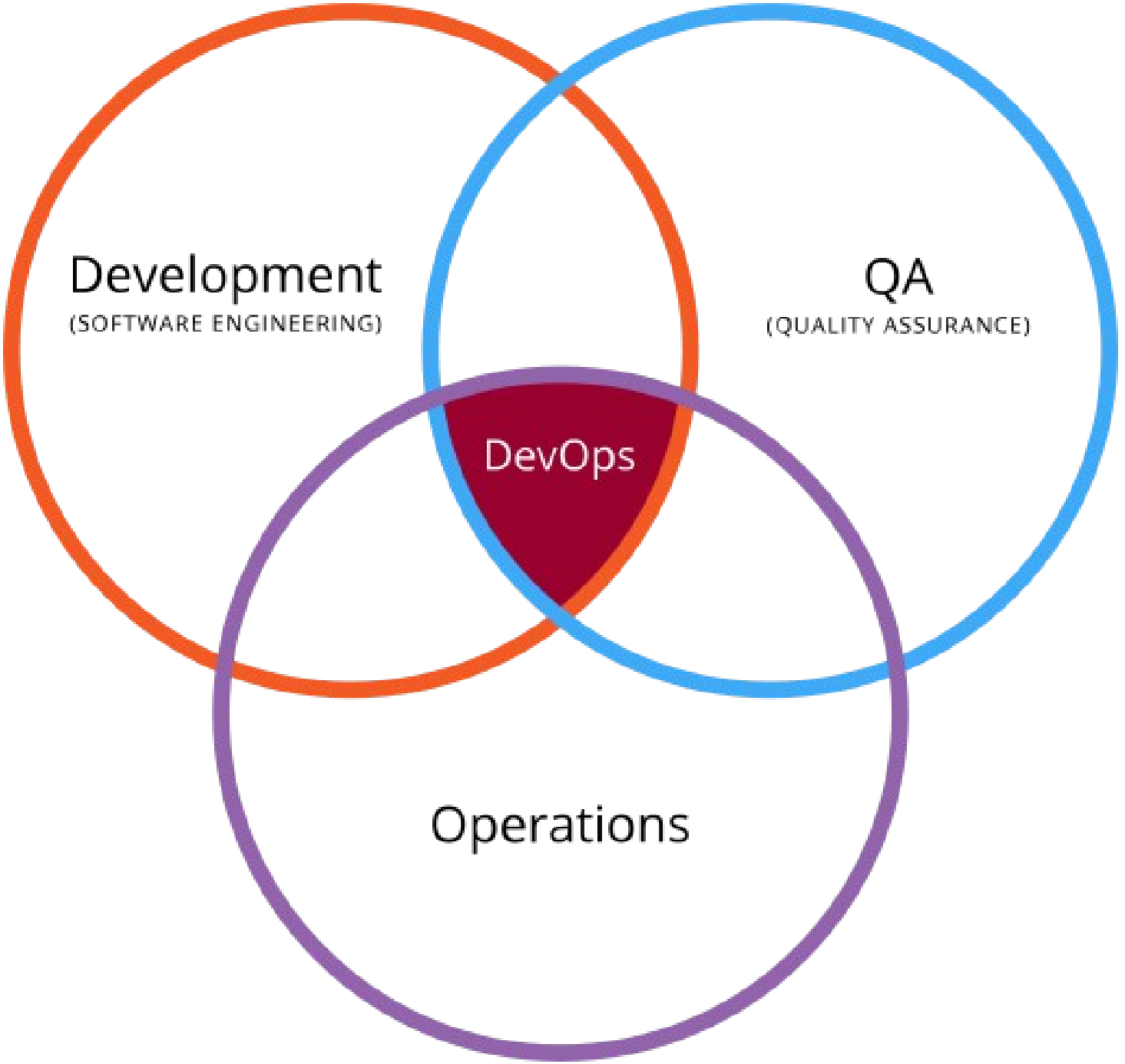

Aplicação Monolítica
Microserviço
Microserviço
Microserviço
Função
Função
Função
Função
Função

Desafios

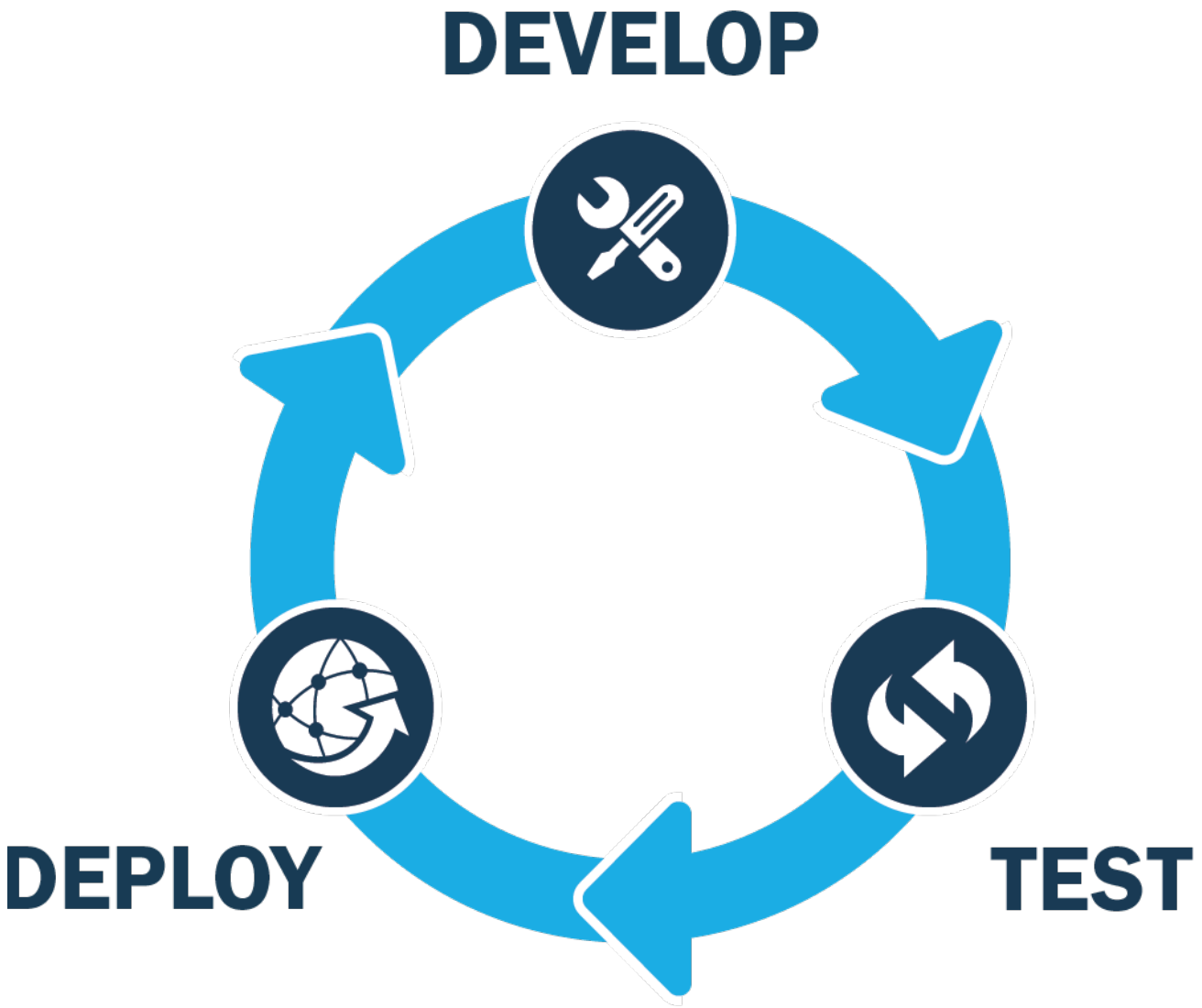
- Monitoramento
- Logging
- Manipulação de estado
  - Aplicações Stateful e Stateless
- Controle transacional
- Autenticação / Autorização
- Relação Operação e Desenvolvimento

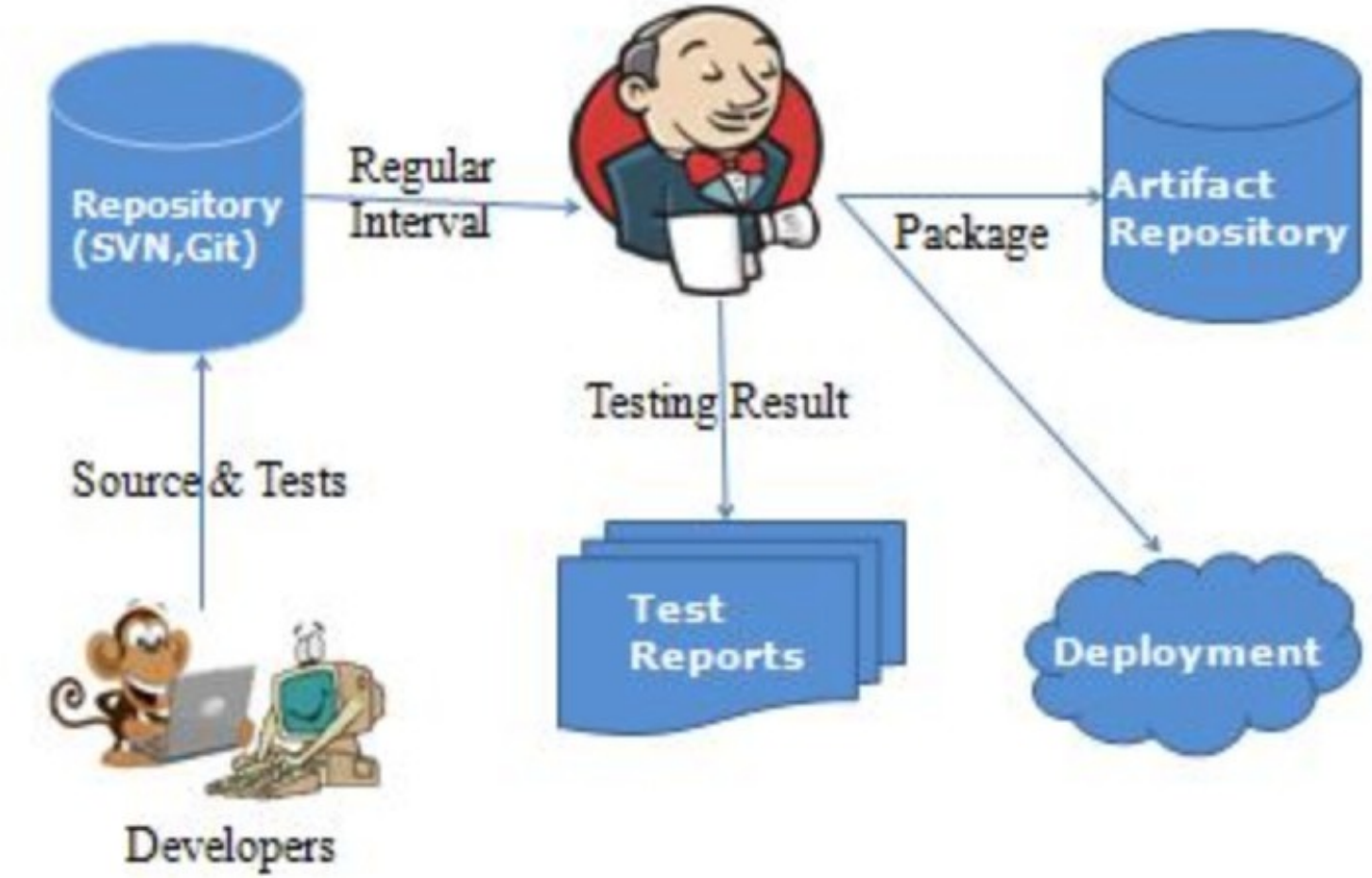
DevOps


Improved Deployment Frequency
Low failure rates of new releases
Short lead time between fixes
Faster time to market
Improved MTR
DevOps

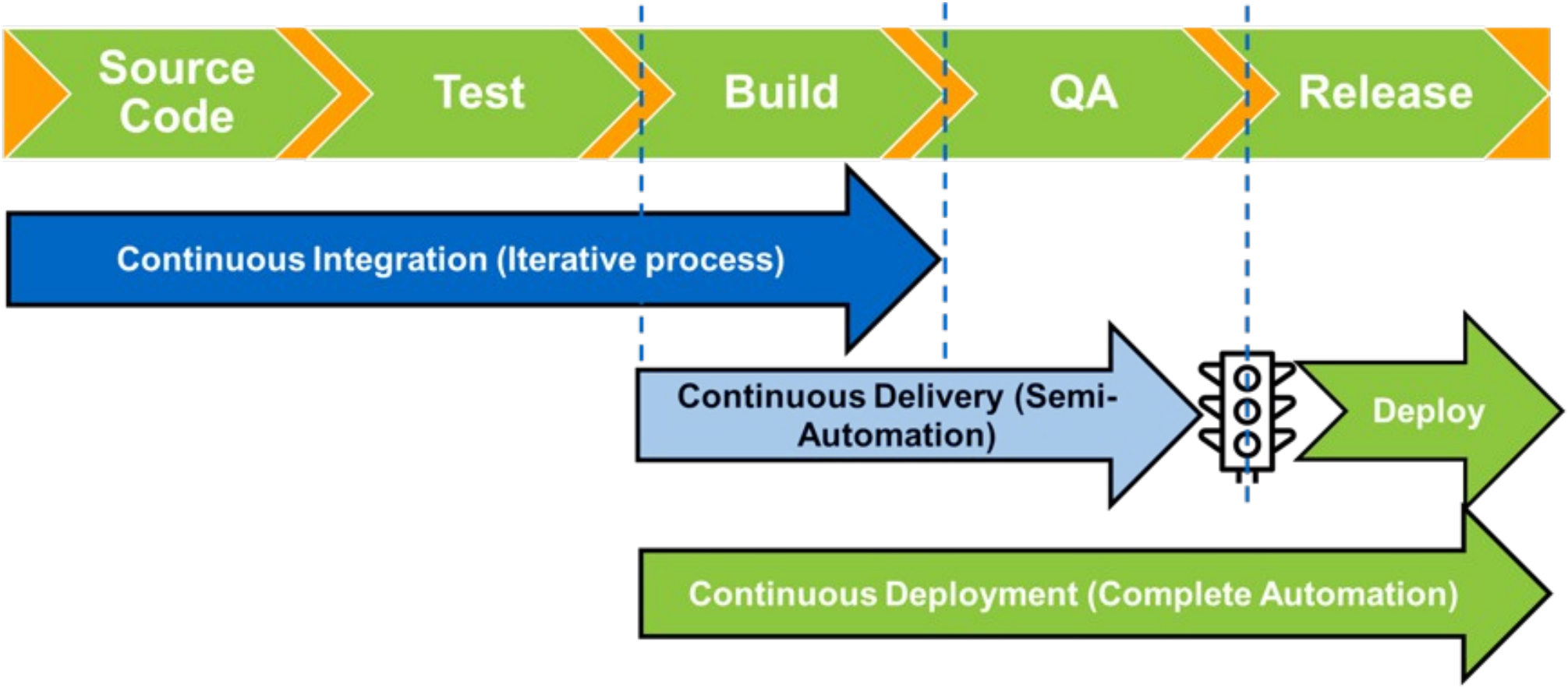

Development
(SOFTWARE ENGINEERING)
QA
(QUALITY ASSURANCE)
DevOps
Operations

Continuous Integration


DEVELOP
DEPLOY
TEST


Repository
(SVN,Git)
Regular
Interval
Artifact
Repository
Package
Testing Result
Source & Tests
Test
Reports
Deployment
Developers

Continuous Delivery


Source Code
Test
Build
QA
Release
Continuous Integration (Iterative process)
Continuous Delivery (Semi-Automation)
Deploy
Continuous Deployment (Complete Automation)

Jenkins

Infrastructure as a Code

Infraestrutura como código é o processo de gerenciamento e provisionamento de centros de processamentos dados usando arquivos de configuração ao invés de configurações físicas de hardware ou ferramentas de configuração interativas.

A infraestrutura de TI envolvida consiste tanto de equipamentos físicos (servidores de metal), assim como máquinas virtuais e outros recursos associados.

A principal característica da IaC é o uso de scripts ou definições declarativas ao invés de processos manuais, mas o termo é usado com mais frequência para promover abordagens declarativas. Como se tratam de arquivos, as definições podem ser armazenadas em um sistema de controle de versões.

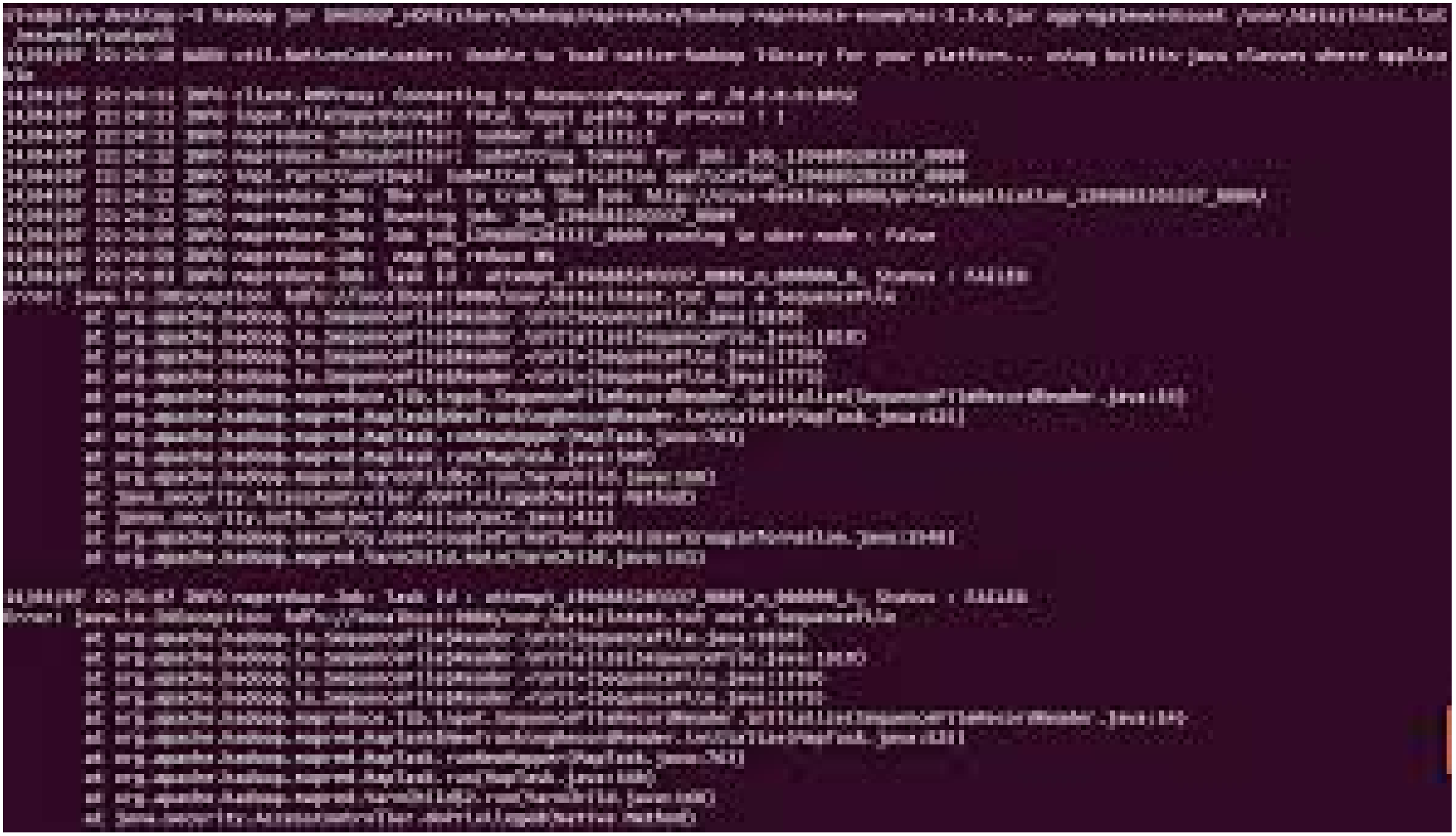
HashiCorp
Terraform

CHEF™

puppet

ANSIBLE

Logging




Apache - Total Visitors
4,931,584
Apache - Unique Visitors
29,740
Apache - Unique Visitors ...
City
Unique Visitors
Beijing 562
Redmond 445
Ashburn 400
Chicago 373
Los Angeles 245
Seattle 233
San Jose 232
Singapore 208
Apache - Bytes and Count
US FR DE NL CN
Count
30,000
20,000
10,000
2017-09-23 20:00
2017-09-30 20:00
2017-10-07 20:00
@timestamp per 3 hours
Apache - Country and Status
Apache - Country traffic by hour
Country
US
CN
DE
GB
FR
Hour of Day
Apache - Visitor Map (geocentroid)
Apache - Browser to Country (vega)
4,500,000
4,000,000
3,500,000
3,000,000
2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
Chrome
Firefox
Googlebot
IE
Other
CN
DE
FR
GB
NL
US
Agent
Country Code


LOG
logstash
Kibana

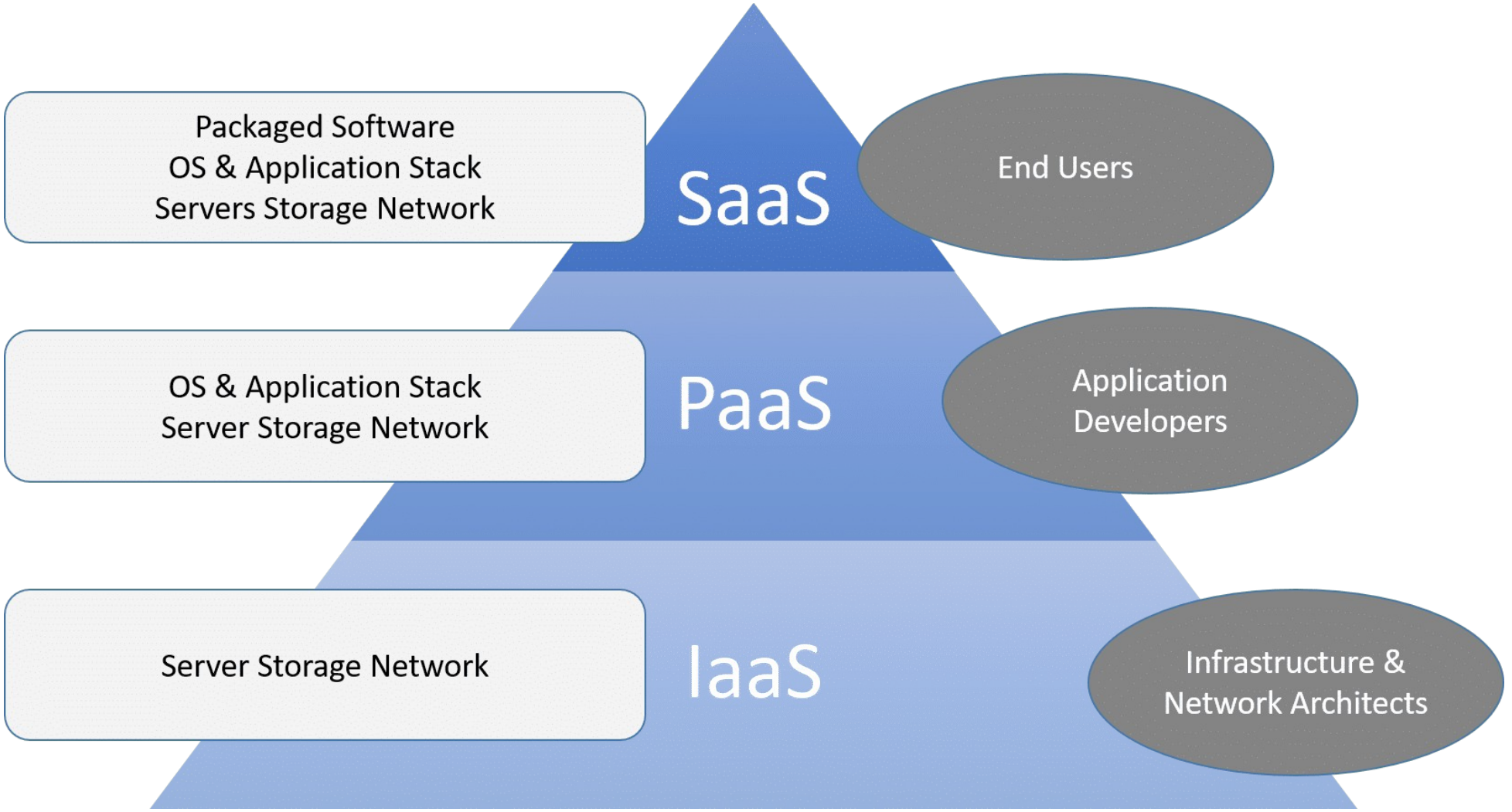
Elasticsearch

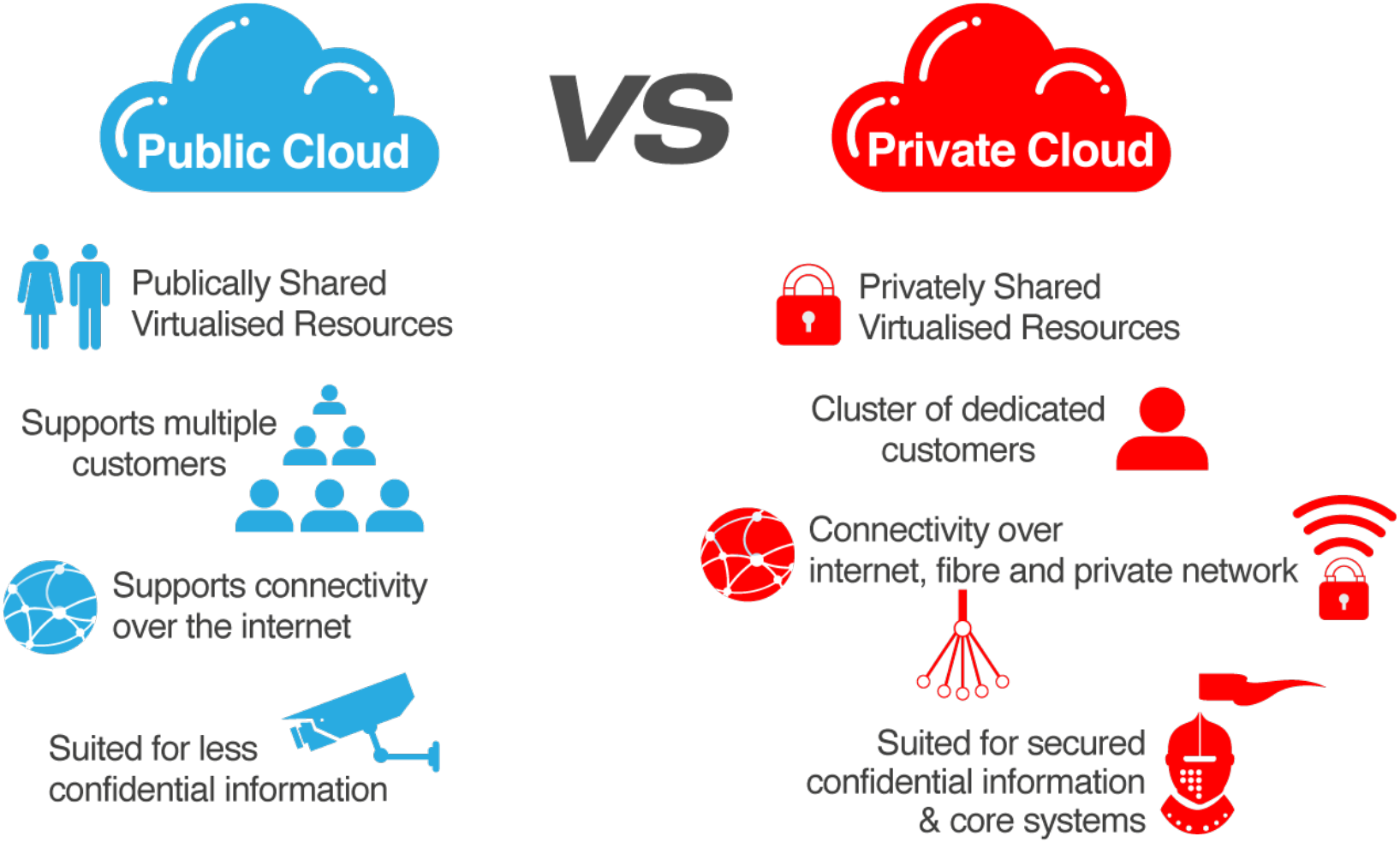
Logstash

Kibana

Cloud Computing

Cloud Service Models




Public Cloud
VS
Private Cloud
Publically Shared
Virtualised Resources
Supports multiple
customers
Supports connectivity
over the internet
Suited for less
confidential information
Privately Shared
Virtualised Resources
Cluster of dedicated
customers
Connectivity over
internet, fibre and private network
Suited for secured
confidential information
& core systems

The Twelve Factors

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

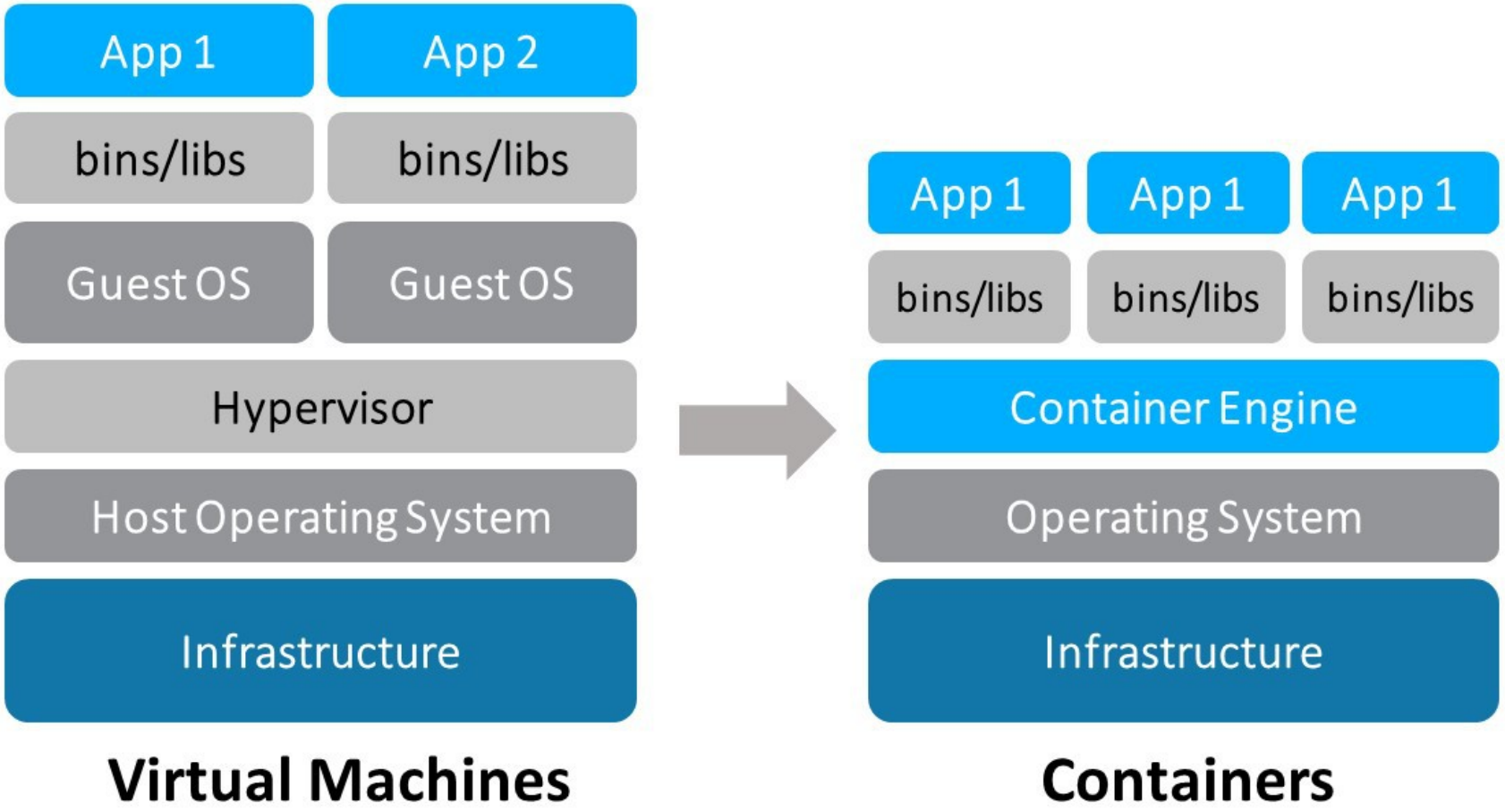
X. Dev/prod parity
Keep development, staging, and production as similar as possible

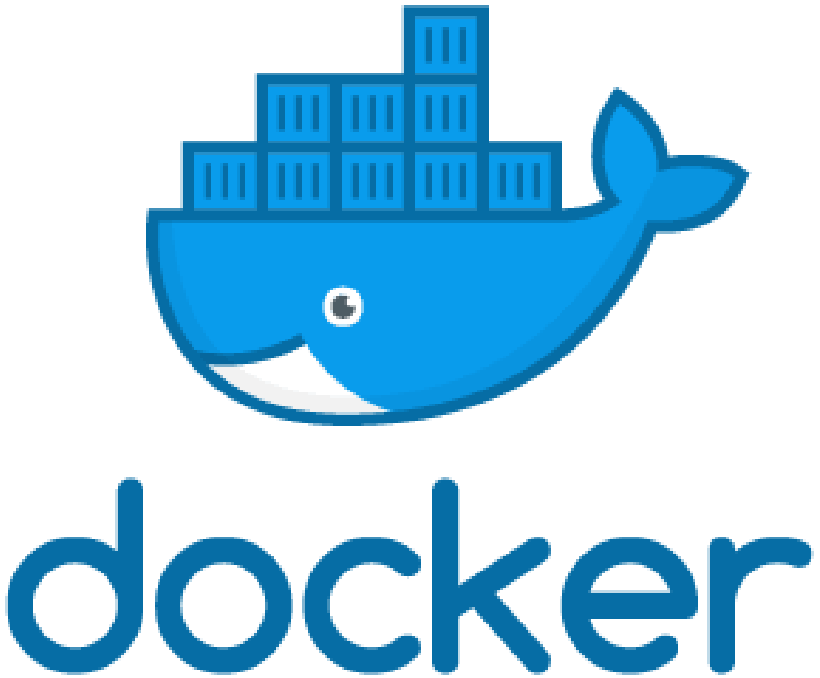
XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

https://12factor.net/

Containers

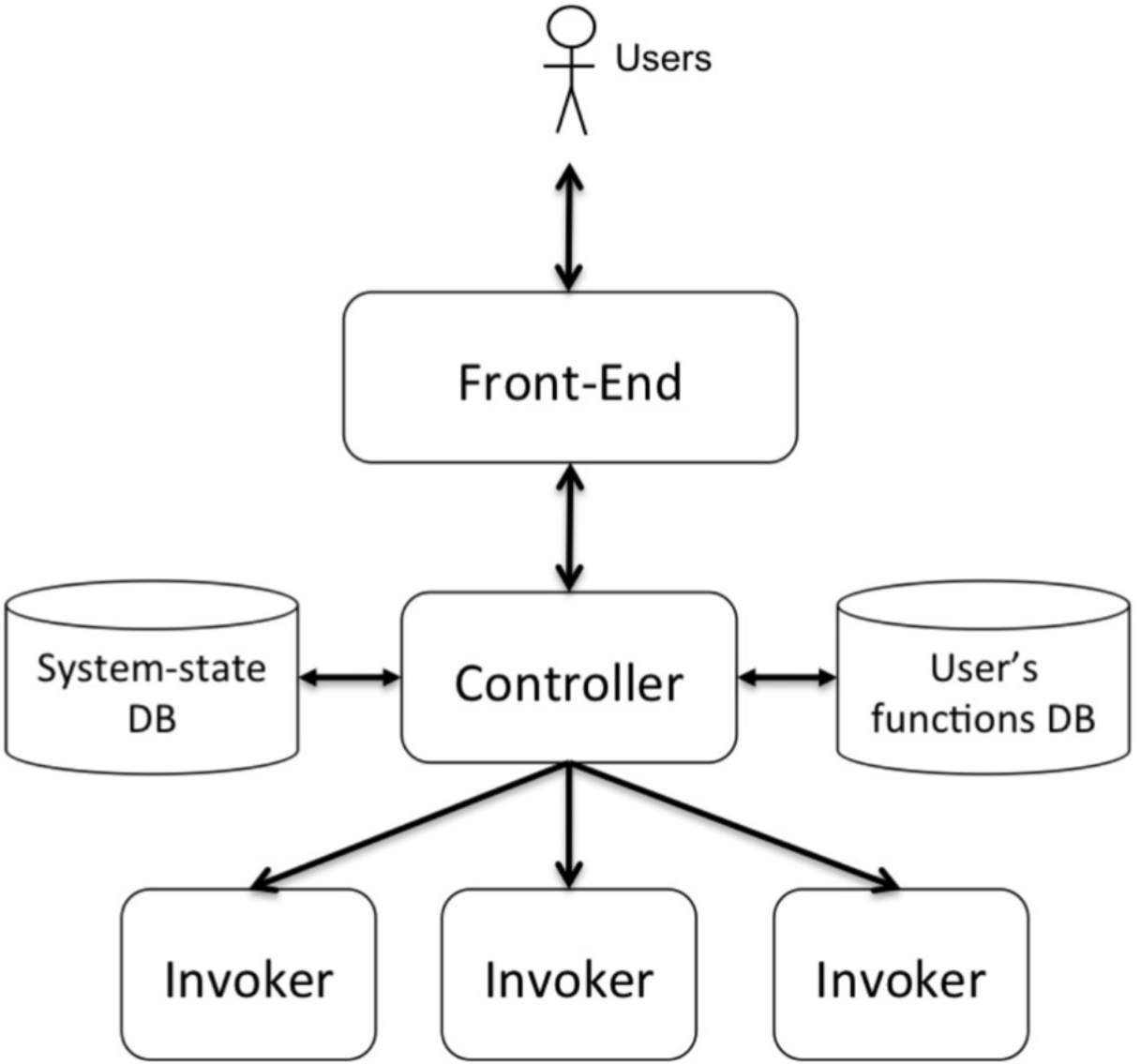

App 1
App 2
bins/libs
bins/libs
Guest OS
Guest OS
Hypervisor
Host Operating System
Infrastructure
Virtual Machines
App 1
App 1
App 1
bins/libs
bins/libs
bins/libs
Container Engine
Operating System
Infrastructure
Containers


docker

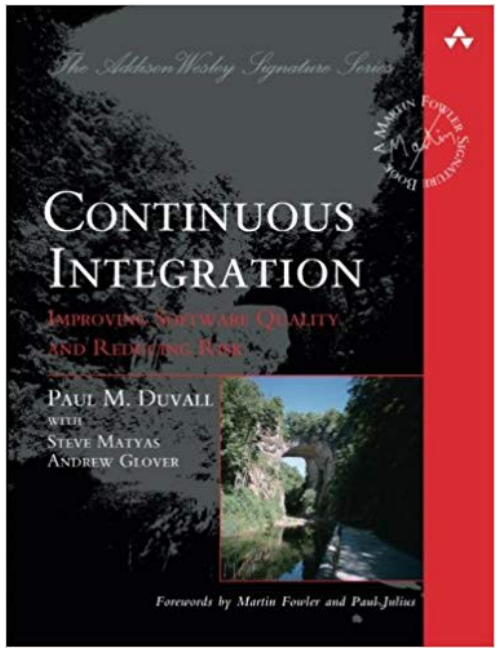
Serverless

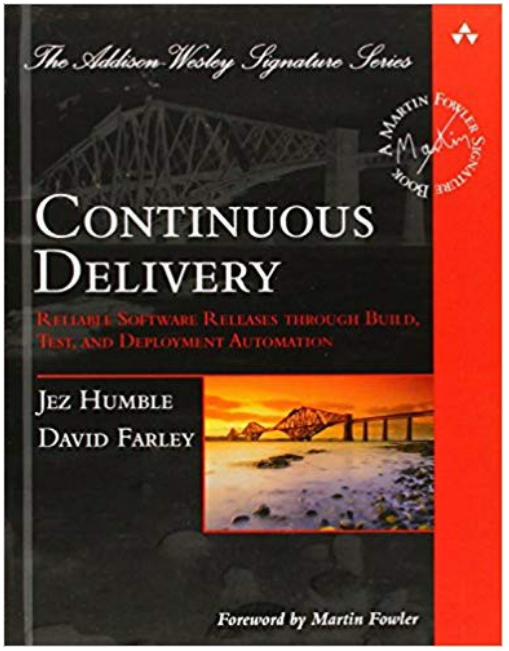
- **Baseado no modelo event driven**
- **Gerenciamento completo do ciclo de vida de uma requisição e container**
- **Diminuição do custo e complexidade operacional**
- **Auxilia na redução da superfície de possíveis ataques**
- **Possibilita um modelo de cobrança por tempo de processamento**
- **Evolução natural de PaaS**

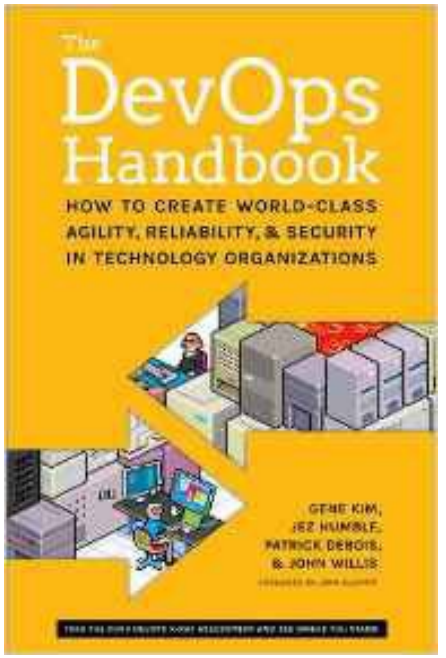
“[...] capacidade de obter funcionalidade sob demanda, de forma transparente, onde o usuário da referida funcionalidade não está ciente da complexidade subjacente envolvida em alcançá-lo.”

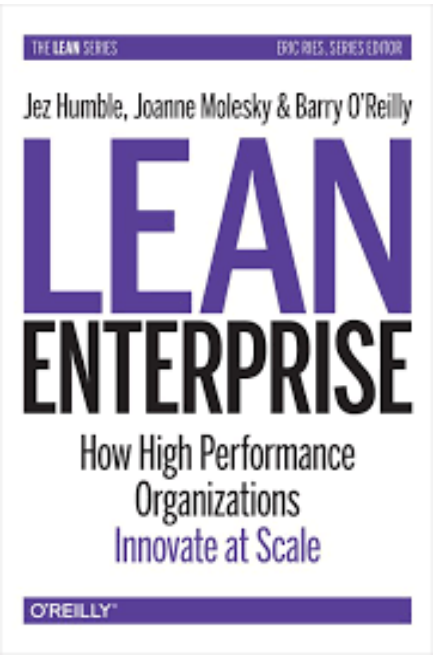

Users
Front-End
System-state DB
Controller
User's functions DB
Invoker
Invoker
Invoker

Leituras recomendadas

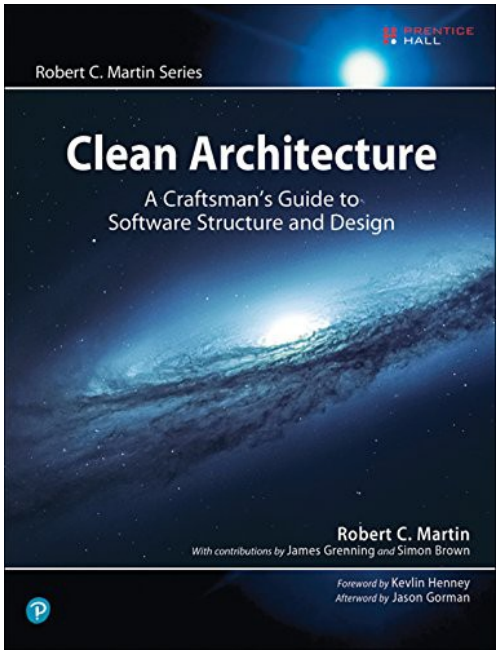

The Addison Wesley Signature Series
Continuous Integration
Improving Software Quality and Reducing Risk
Paul M. Duvall
with
Steve Matyas
Andrew Glover
Forewords by Martin Fowler and Paul Julius

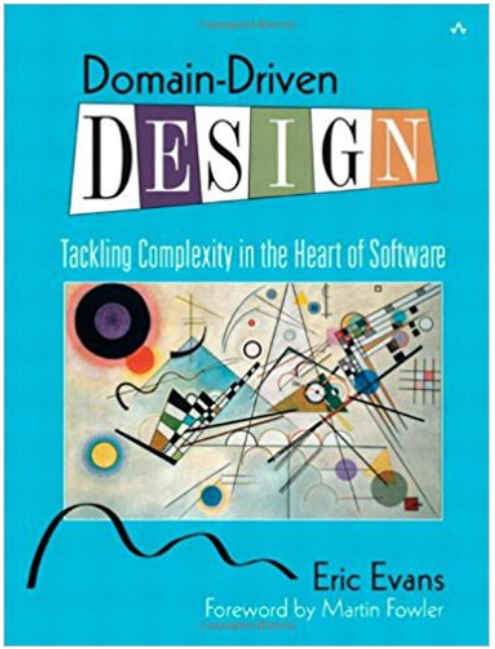

The Addison-Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler


The DevOps Handbook
How to Create World-Class Agility, Reliability, & Security in Technology Organizations
Gene Kim, Jez Humble, Patrick Debois, & John Willis


The Lean Series
Eric Ries, Series Editor
Jez Humble, Joanne Molesky & Barry O'Reilly
Lean Enterprise
How High Performance Organizations Innovate at Scale
O'Reilly


Scrum
A Arte de Fazer o Dobro do Trabalho na Metade do Tempo
Edição Revista
Jeff Sutherland
Cocriador do Scrum
LeYa


Robert C. Martin Series
Prentice Hall
Clean Architecture
A Craftsman's Guide to Software Structure and Design
Robert C. Martin
With contributions by James Grenning and Simon Brown
Foreword by Kevlin Henney
Afterword by Jason Gorman


Domain-Driven
Design
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

Obrigado!