

NServiceBus or MassTransit: .NET ServiceBus Frameworks Compared



Roland Guijt

MVP | CONSULTANT | TRAINER

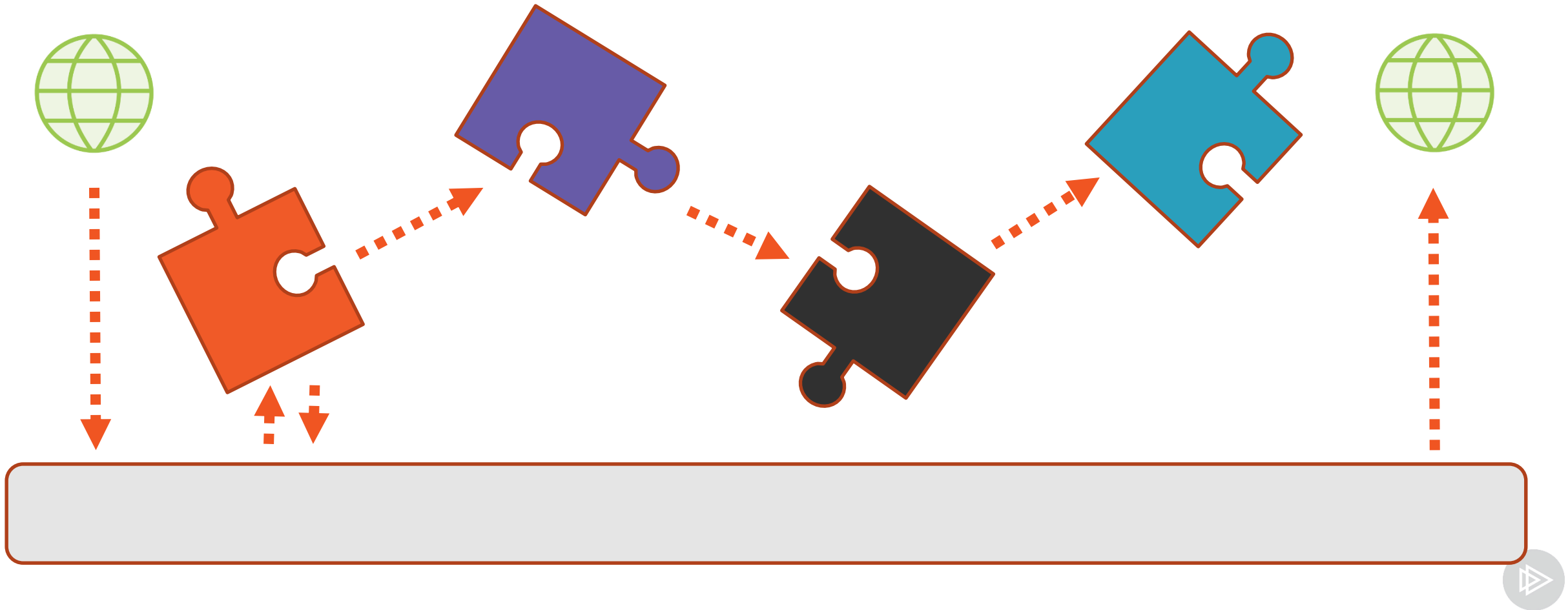
@rolandguijt rolandguijt.com



Context: Distributed Applications



Service-oriented Architecture



Coupling

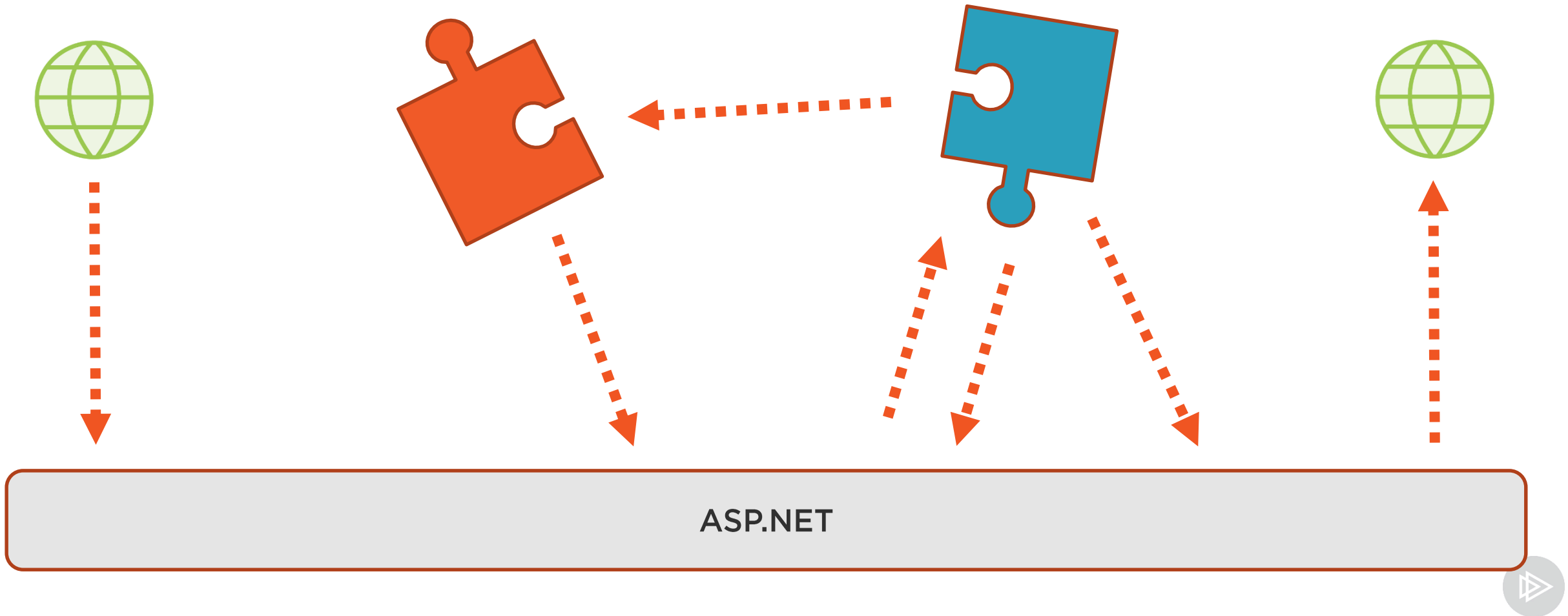
Platform

Behavioral

Temporal



Services with Messaging



Message Queueing Systems



MSMQ

Native to Windows

Decentralized: each machine has its own queues

Store and forward

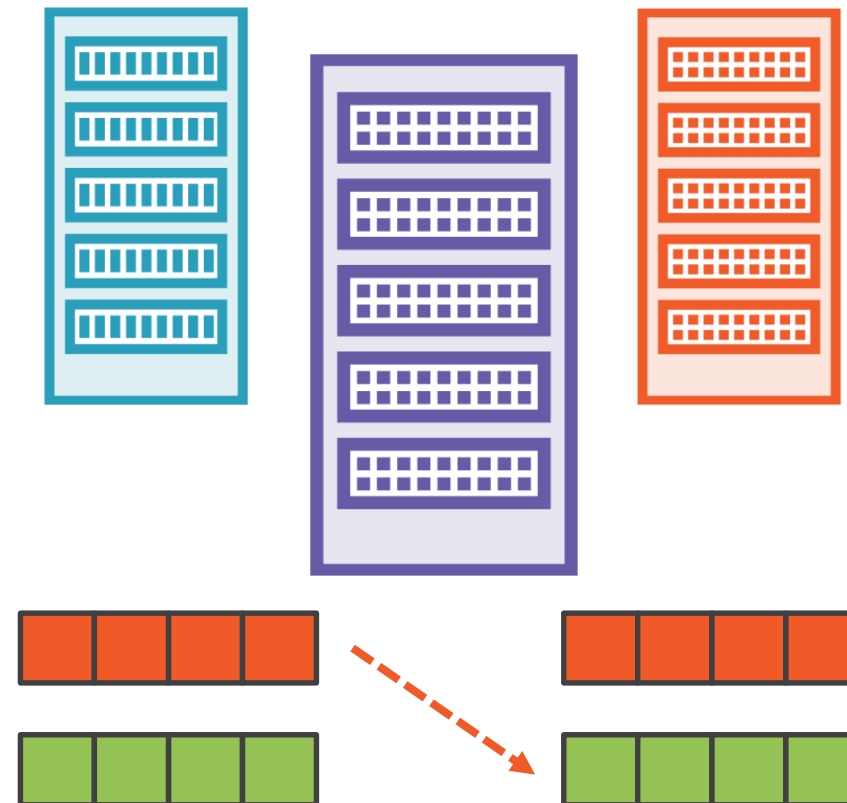


RabbitMQ

Broker style

Multiple platforms

Supports AMQP



Microsoft Azure

Storage Queues

Service Bus



The Service Bus



Service Bus vs Native Transport API

Service Bus

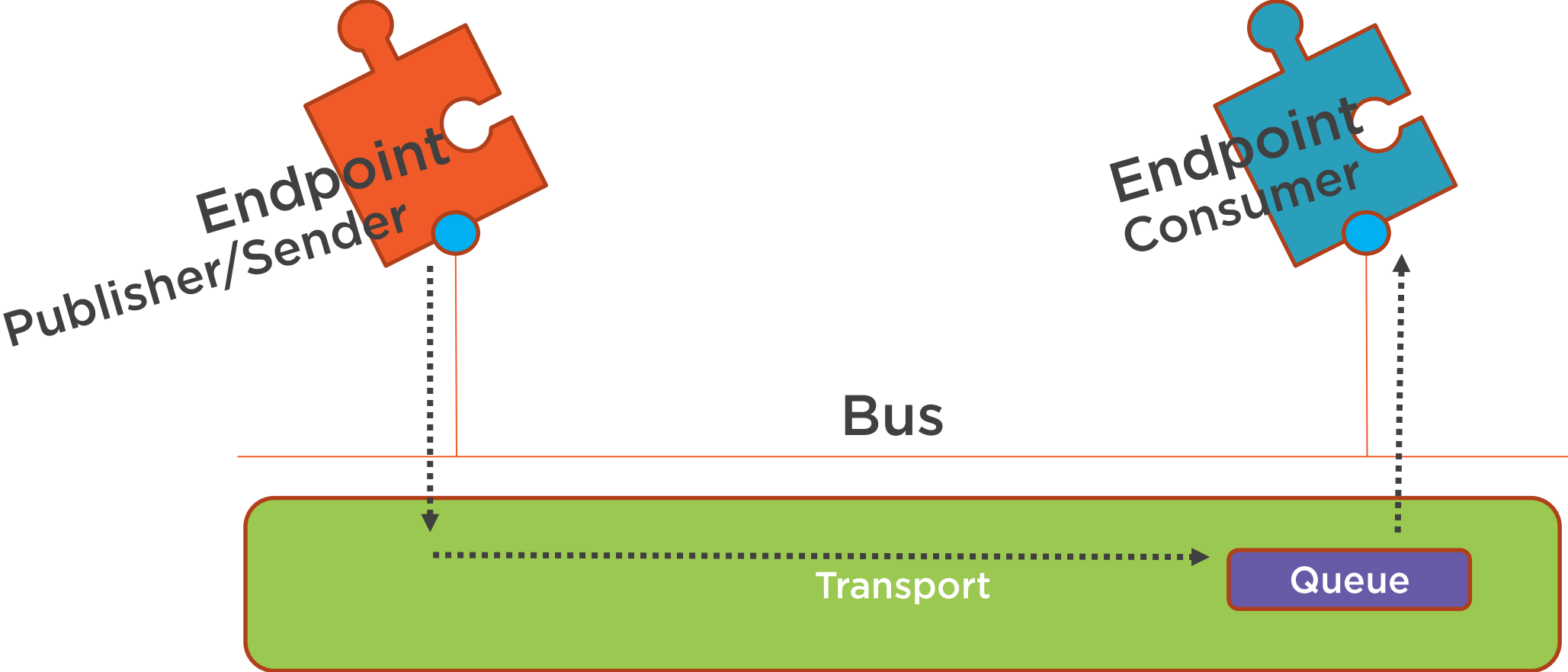
- A ready to go framework for messaging
- Hides complexity of transport
- Part of transport features supported
- More geared towards type system
- Supports multiple transports
- Easy to unit test

Native Transport API

- Low level: Create framework yourself
- Exposure to complexity of transport
- Full transport feature support
- Heavy use of strings
- One transport supported
- Not designed with unit testing in mind



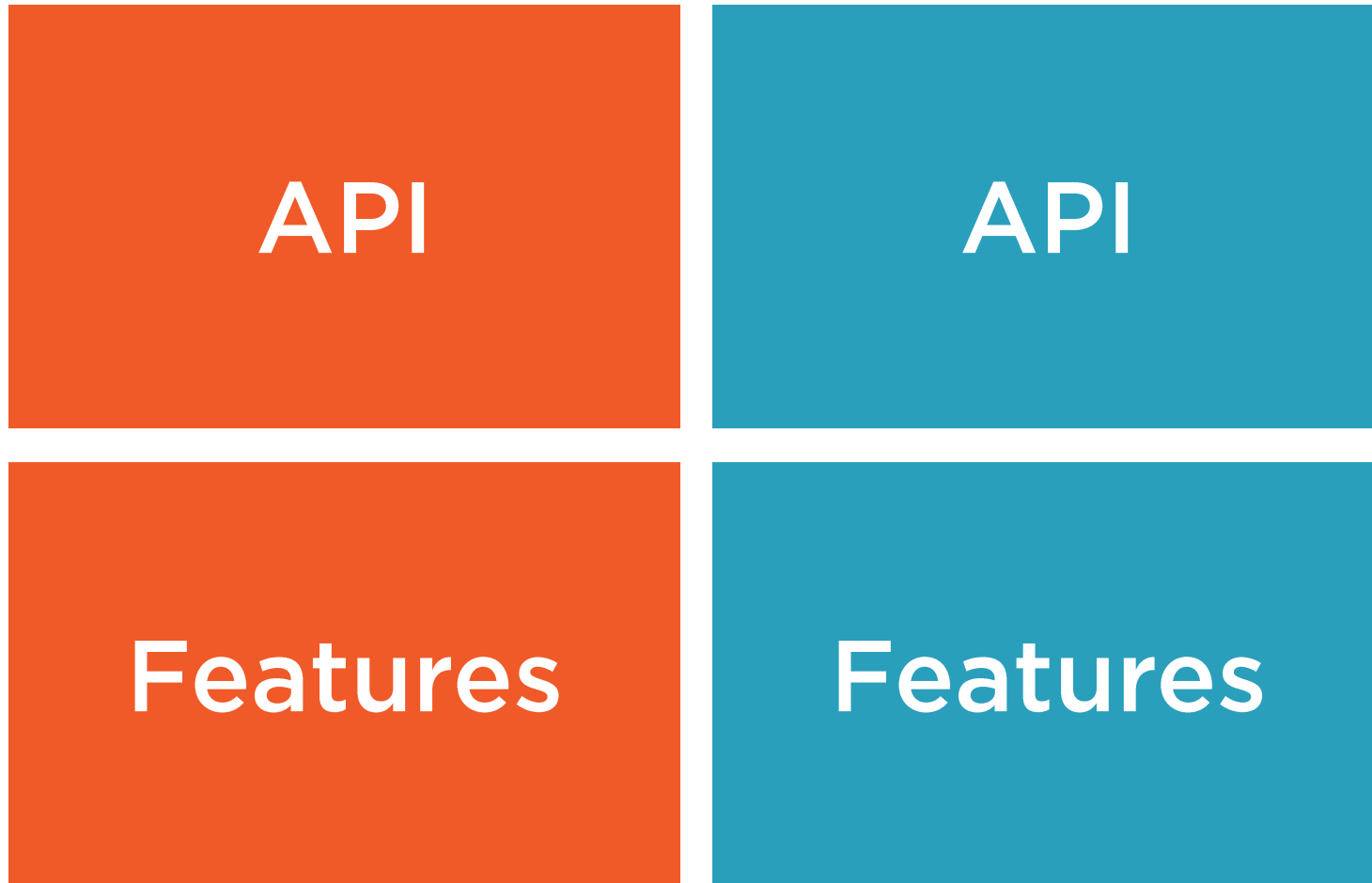
Message-based Architecture Terminology



NServiceBus and MassTransit API



MassTransit vs NServiceBus



MassTransit vs NServiceBus

**RabbitMQ
and ASB
transports**

**MSMQ and
many more**



MassTransit vs NServiceBus

OSS

OSS

Chris
Patterson

Particular
team



MassTransit vs NServiceBus

Free
Apache 2.0

Licensed

\$0

From
\$25/month



MassTransit vs NServiceBus

**Do-it-
yourself
support**

**Support
>=\$50
/month**



MassTransit vs NServiceBus

**No
monitoring
software**

**Monitoring
software
from \$50**



MassTransit vs NServiceBus

**Tested by
unit tests**

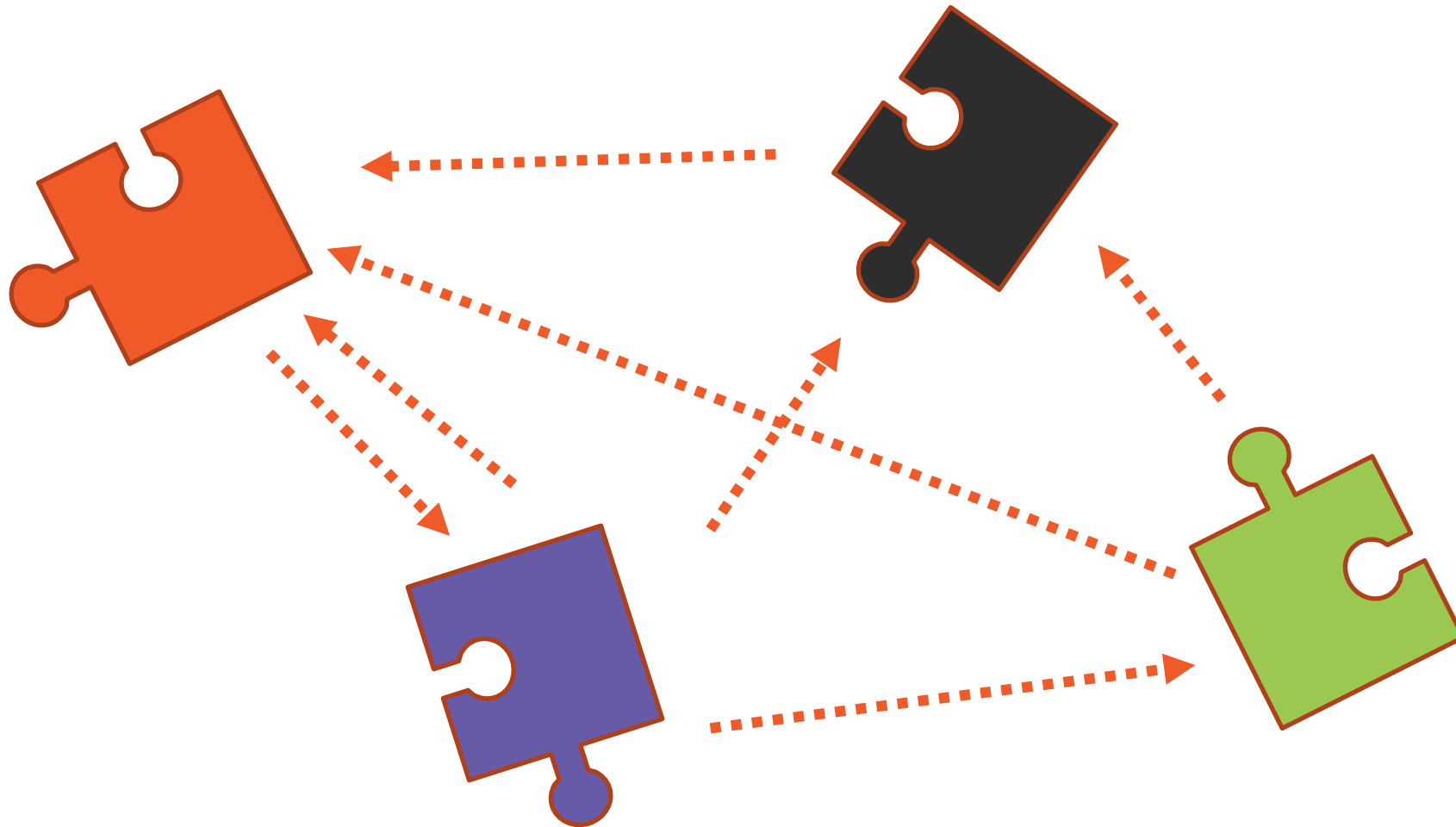
**Production
scenario
testing**



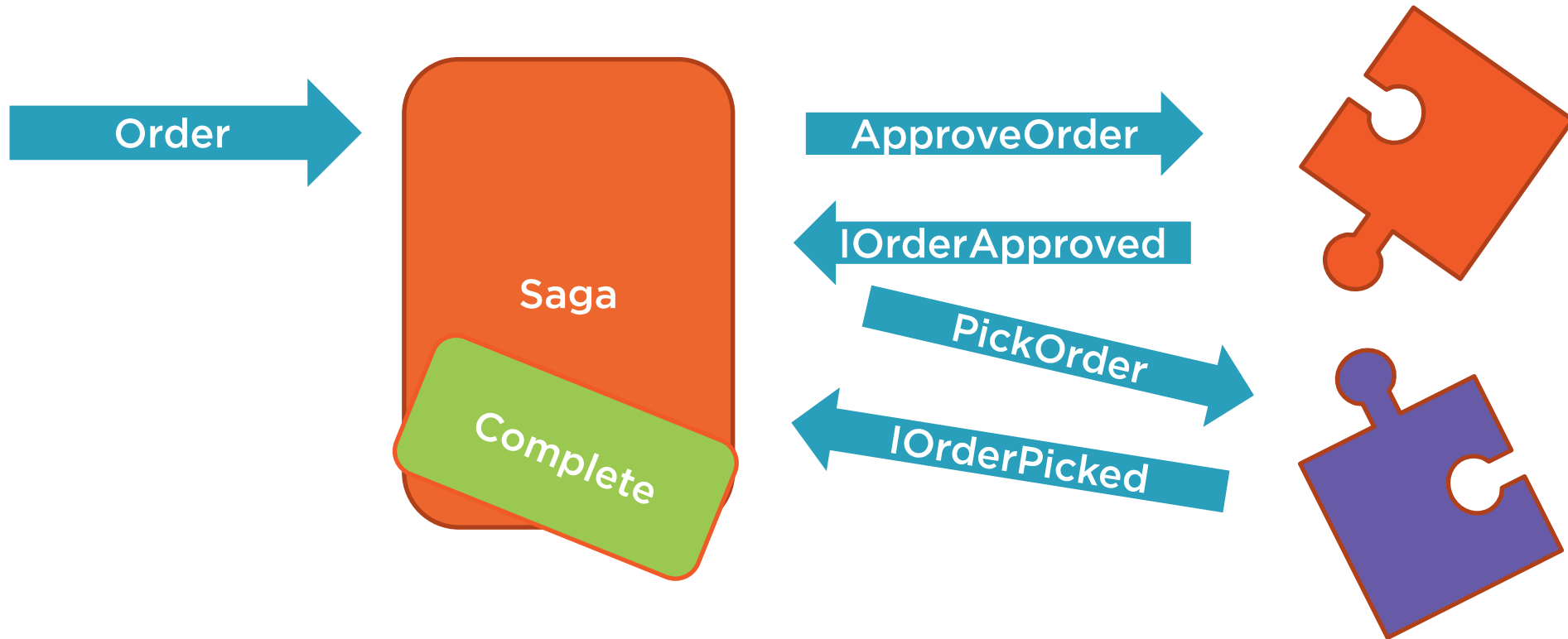
Sagas



Why Sagas?



What Are Sagas?



When to Use a Saga?



Processes with more than one message roundtrip

Time related process requirements



Sagas: API



Other Features

Retries

Transactions

Manipulate pipeline

Attachments

Encryption

Scheduling

Request/Response support

Performance counters



Bottom Line

Do you want to pay or not

Paying gives you

- More transport choice (MSMQ)
- A thoroughly tested product
- Support
- A couple of enterprise features

And maybe your prefer one API over the other



Thanks

roland.guijt@gmail.com
@rolandguijt

