A Crash Course on Building Microservice-based Systems

@ShawnWallace

Favorite Movie

My Son

My Daughter

My Family

I live here

I a Marine Vet

My Favorite Team

Where I work

I'm

SAFETY FIRST

SAFETY FIRST

CAREFUL

THIS MACHINE
HAS NO BRAIN
USE YOUR OWN

⚠ WARNING

Heavy Object.
Can Cause
Back Injury.
Do not lift
with back.

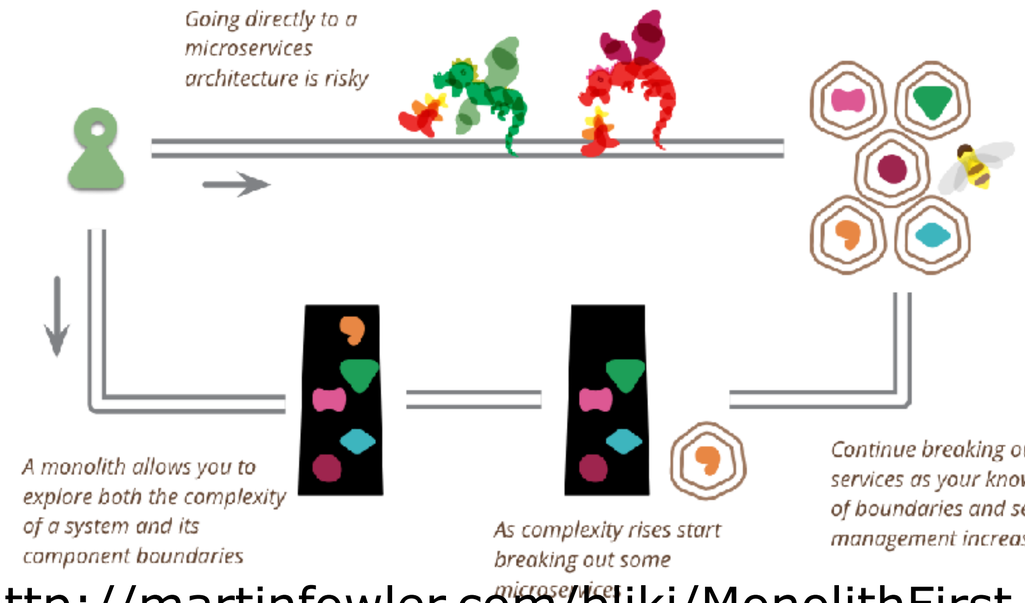
A word about ‘productizing’ patterns and practices

((CENTRIC))

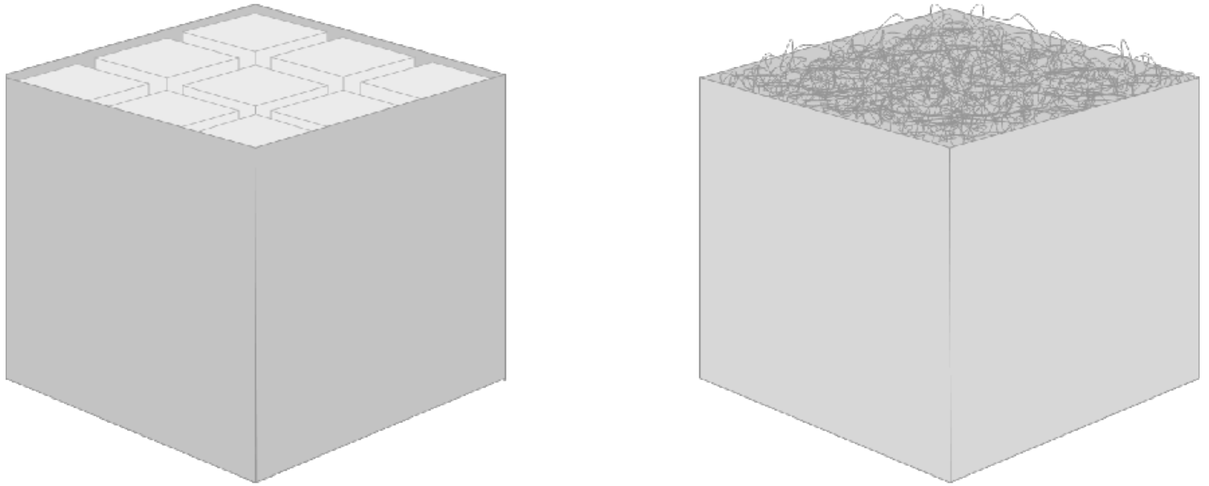
The Great Debate…

http://martinfowler.com/articles/dont-start-monolith.htm

Monolith First?



http://martinfowler.com/bliki/MonolithFirst.html

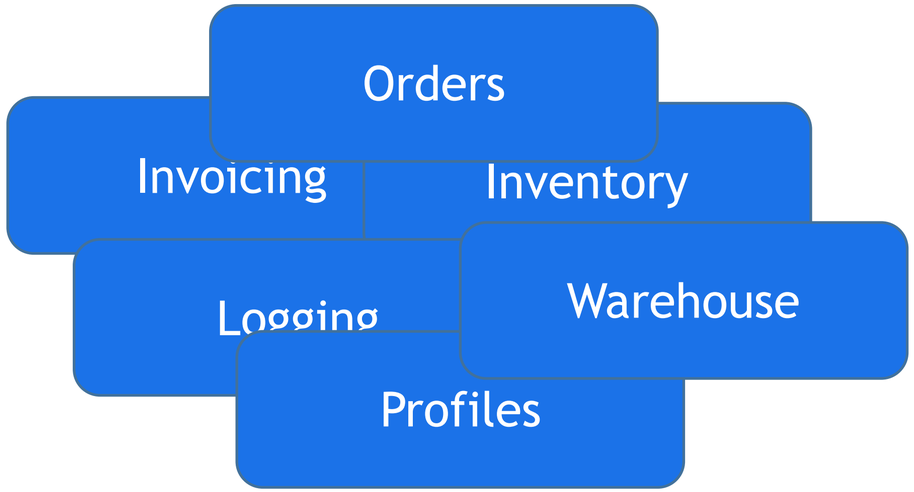


Hope vs. Reality

Micro services First?

((CENTRIC))

Monolithic



A problem with ‘Monolith First’

A product of our agile approach to our projects…smaller features finished earlier.

The goal is to deploy more often…this is hard.

One solution is to have smaller applications.

((CENTRIC))

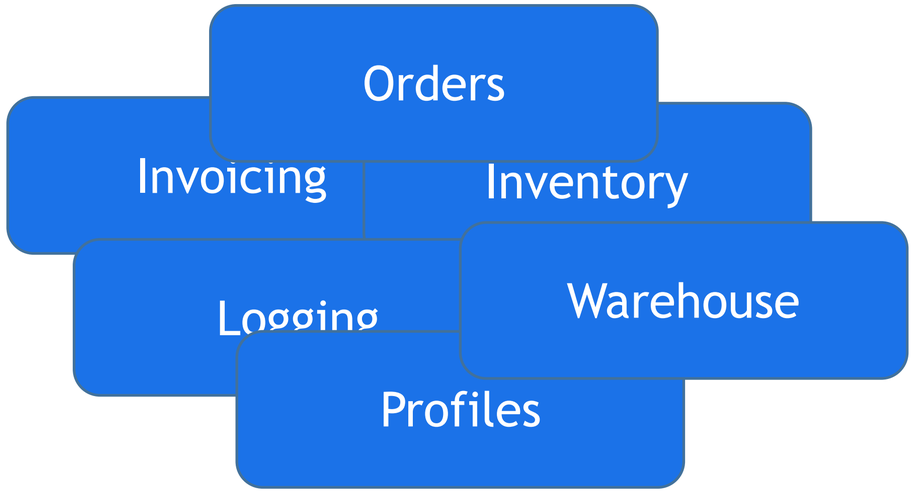
A problem with ‘Monolith First’

A product of our agile approach to our projects…smaller features finished earlier.

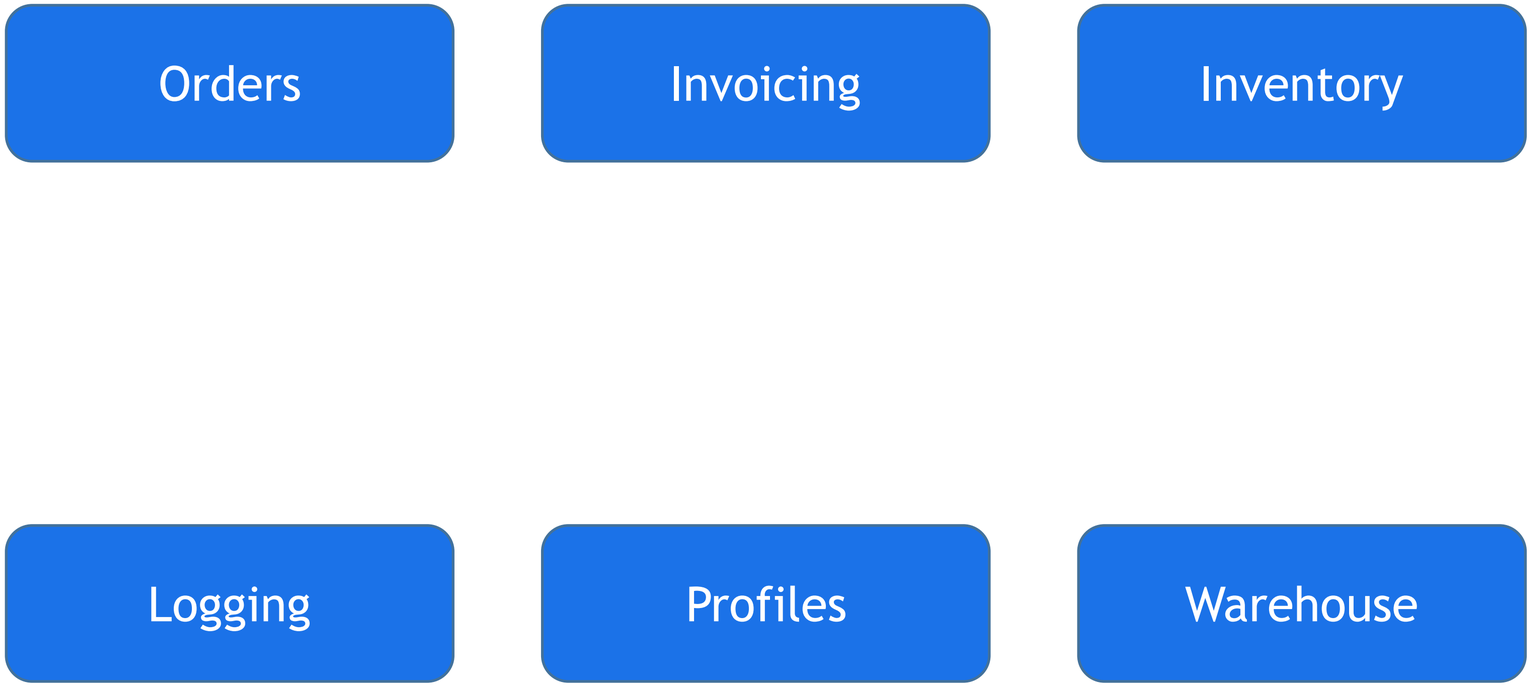
The goal is to deploy more often…this is hard.

One solution is to have smaller applications.

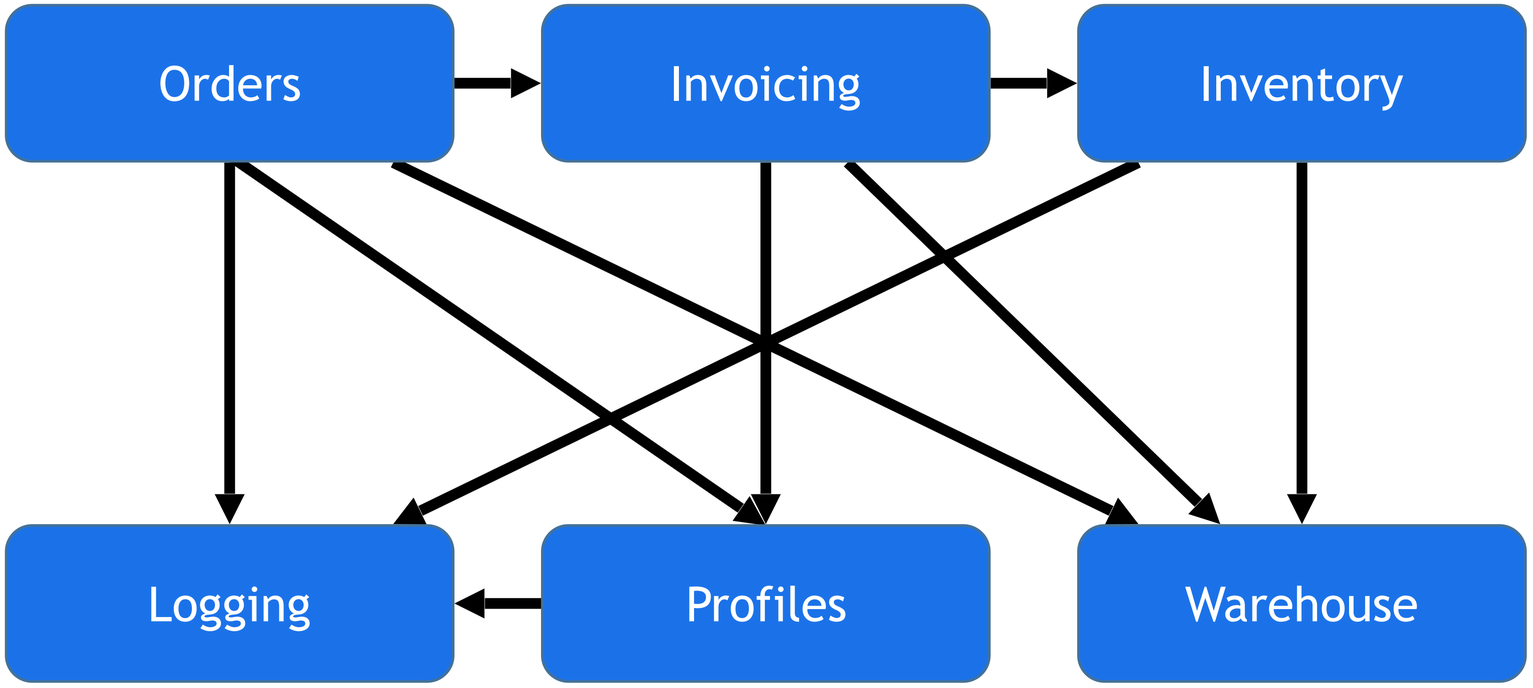
Monolithic



SOA



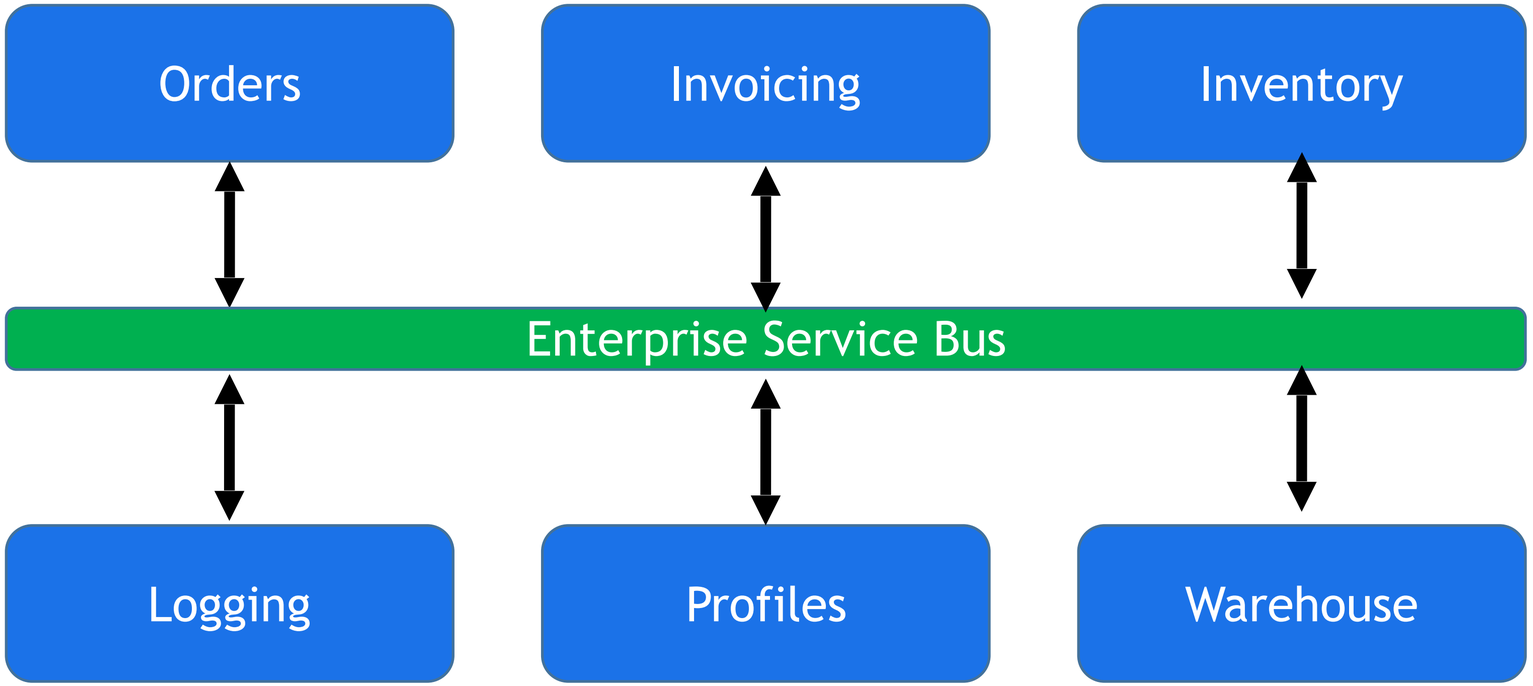
SOA



Tightly Coupled

((CENTRIC))

SOA with Enterprise Service Bus

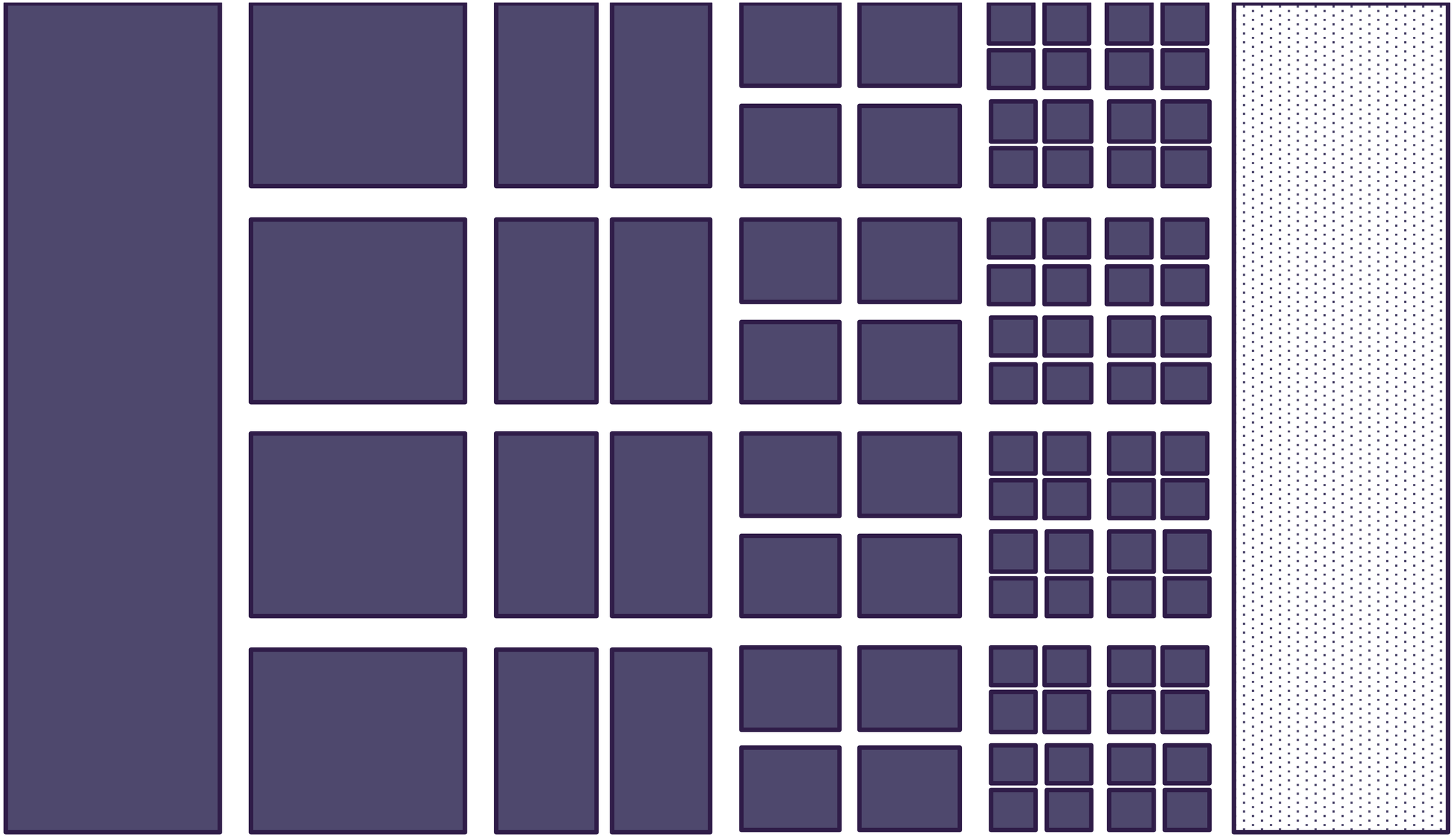


STILL Tightly Coupled

((CENTRIC))

“The Role of the ESB still has its place — now in the form of a modern scalable message queue”

Source: Reactive Micoservices Architecture (Design Principles for Distributed Systems p. 34
Jonas Bonér


MONOLITH
CLIENT SERVER
WEB & INTERNET
CLOUD & WEB SERVICES
MICRO SERVICES
2020 - 2030

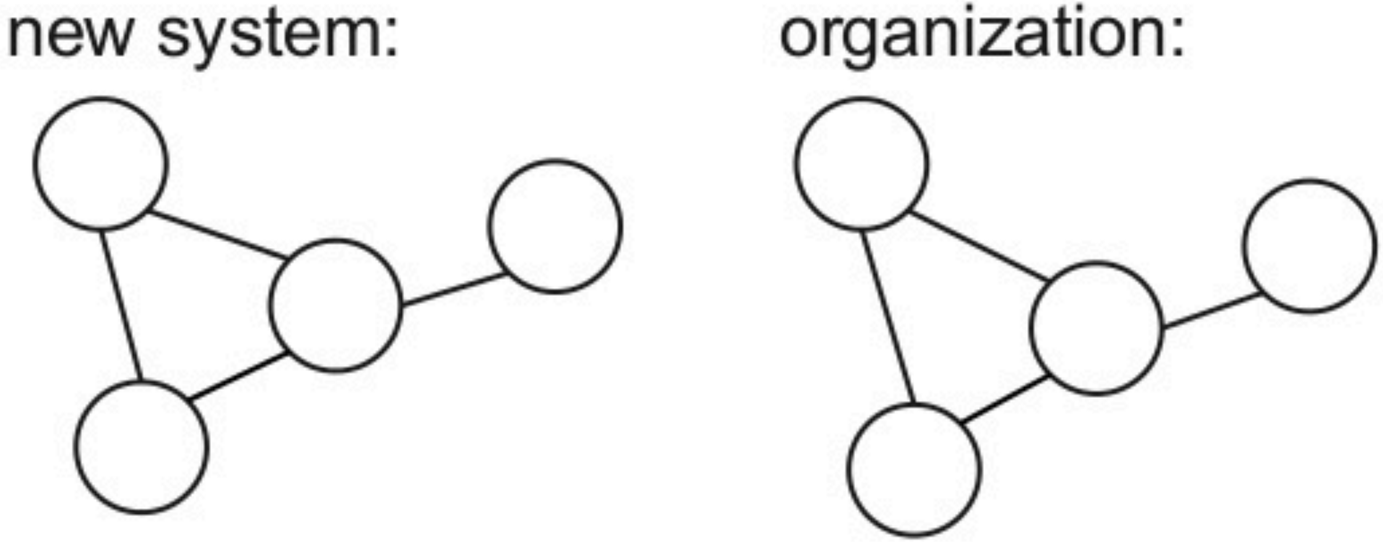
What are Microservices

- Independently testable/deployable
- Operate in isolation (loosely coupled)
- Maintains own state
- Asynchronous external interactions that favor message passing

Important Early Decisions

- Embrace Asynchronous Interactions
- Communication Pattern
- Logging Infrastructure

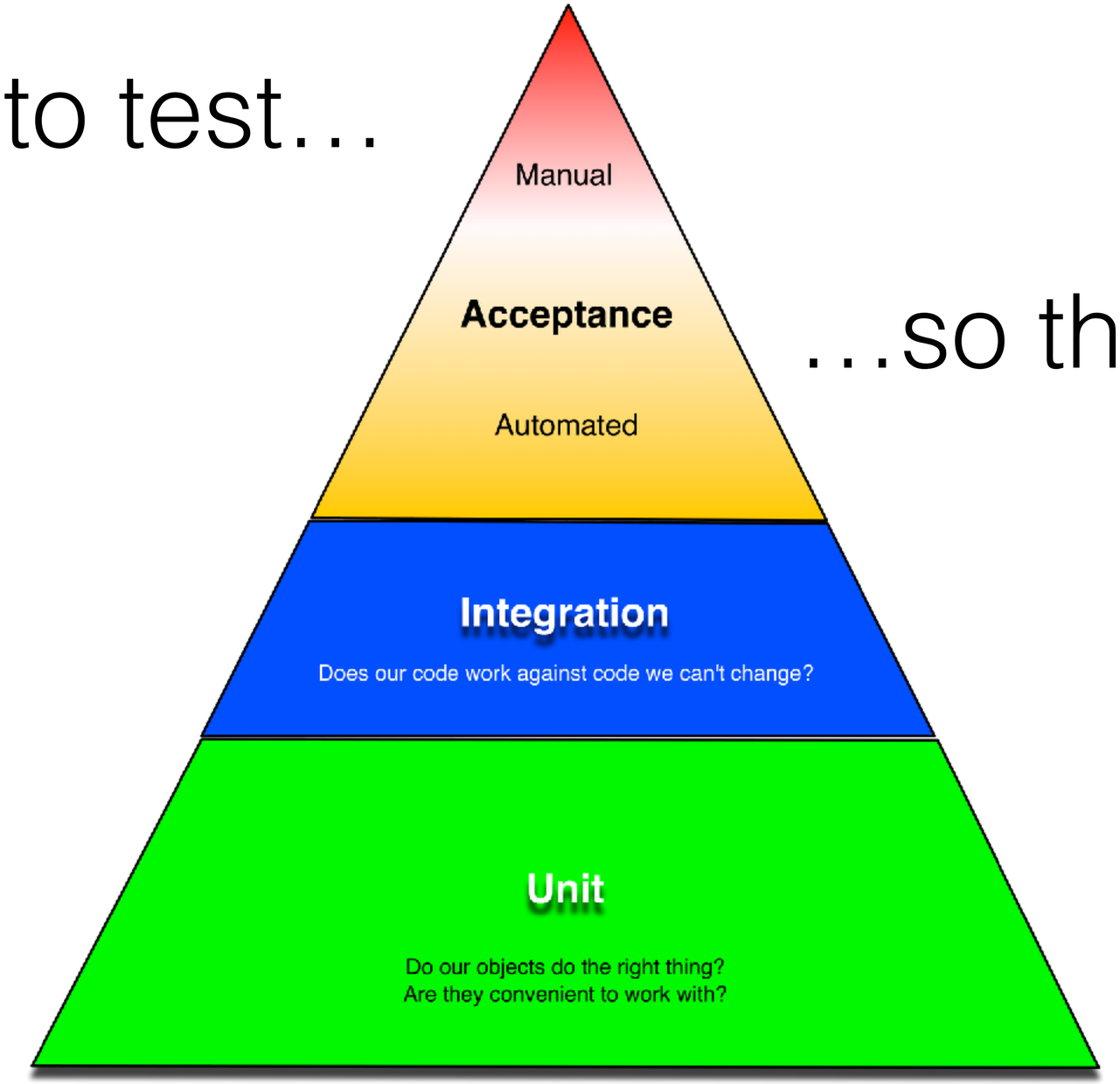
conway's law Embrace it



The basic thesis of this article is that organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations.

--- Melvin Conway

Be easy to test…



…so that you do.

((CENTRIC))

ReST is NOT your friend

Code Reuse is Overrated

Don’t I need Transactions?

Should Client Communication Still be Asynchronous?

((CENTRIC))

We have to think about our data differently

((CENTRIC))

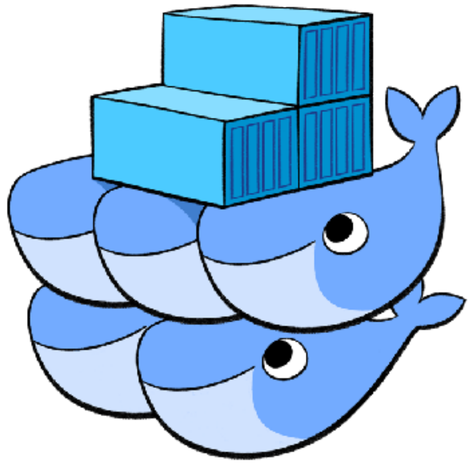
We have to think about our data differently

- Taking data from being OFFLINE to ONLINE
- Moved from “data at rest” to “data in motion”


DEMO

Scaling Model

((CENTRIC))



Scaling Model

Azure
Container Service

((CENTRIC))

Monitoring

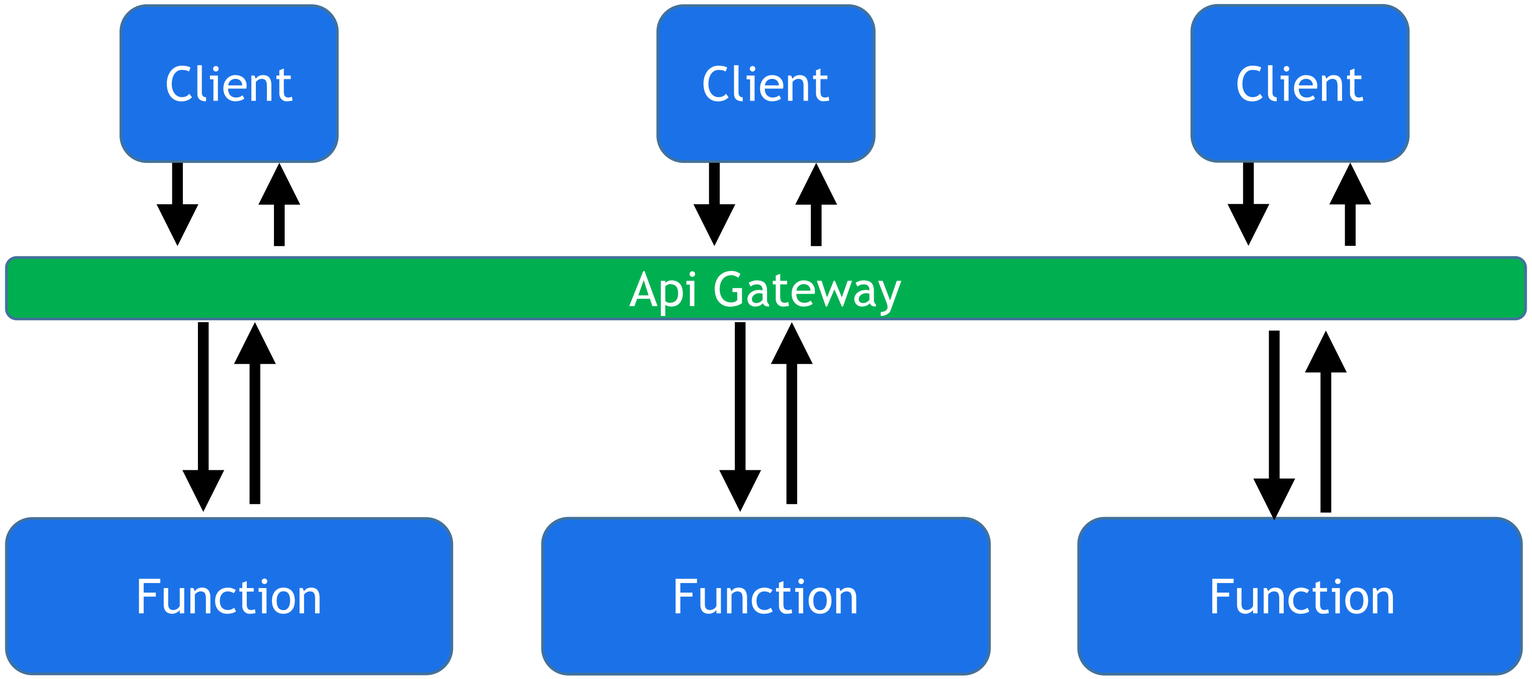
((CENTRIC))

Composition

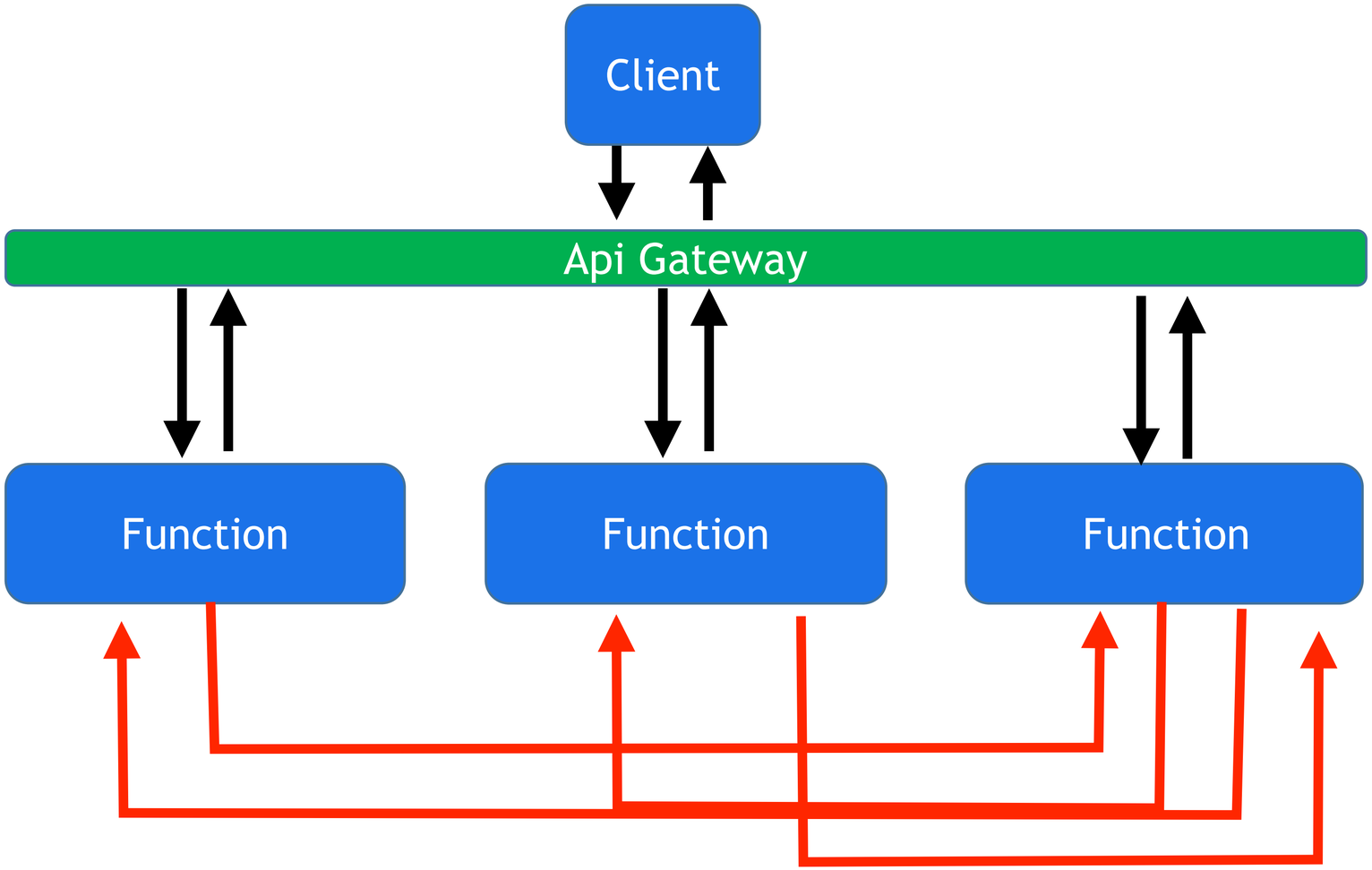
API Gateway

((CENTRIC))

Function as a Service

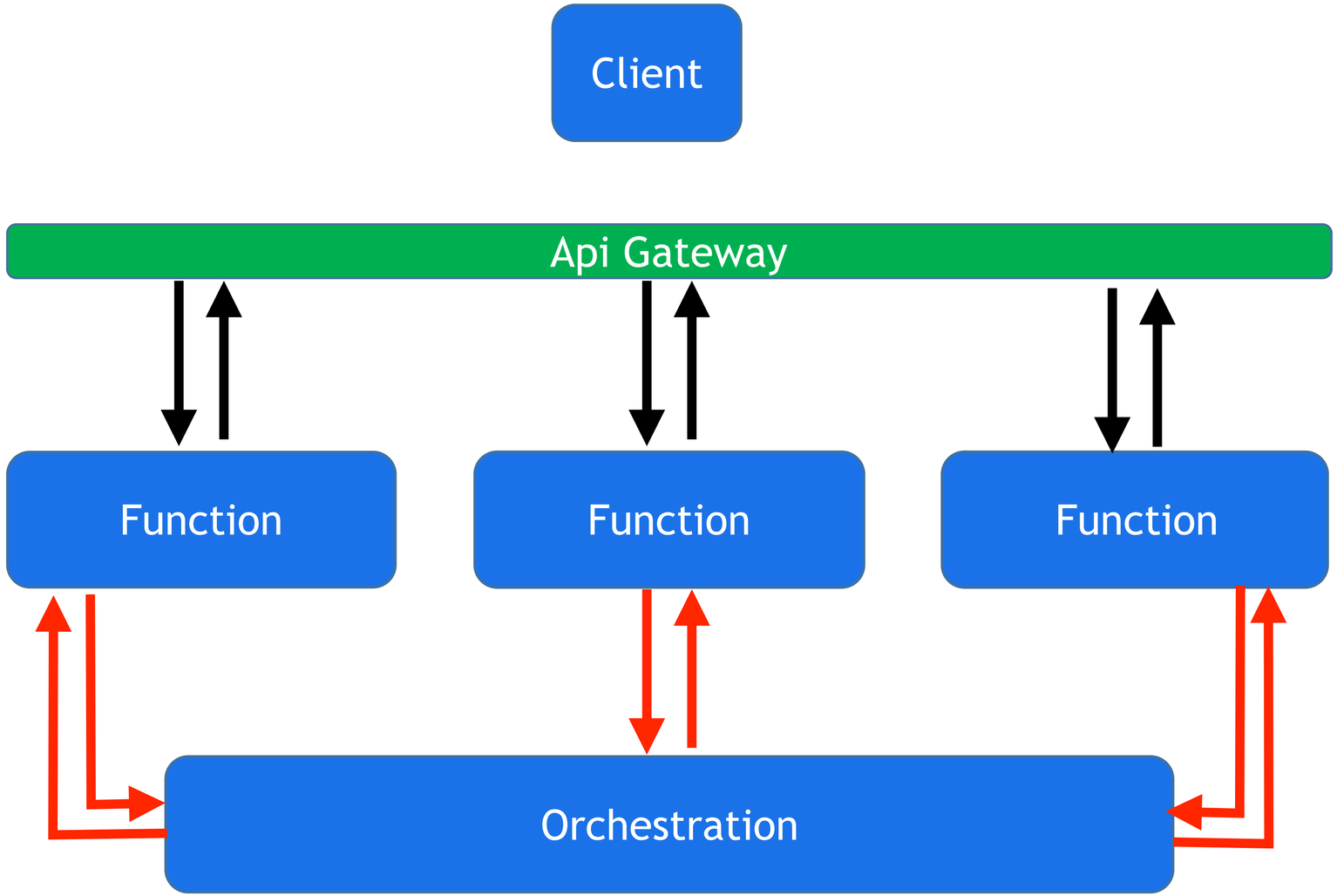


Function as a Service



((CENTRIC))

Function as a Service



Event-Driven

((CENTRIC))

Event Driven Architecture

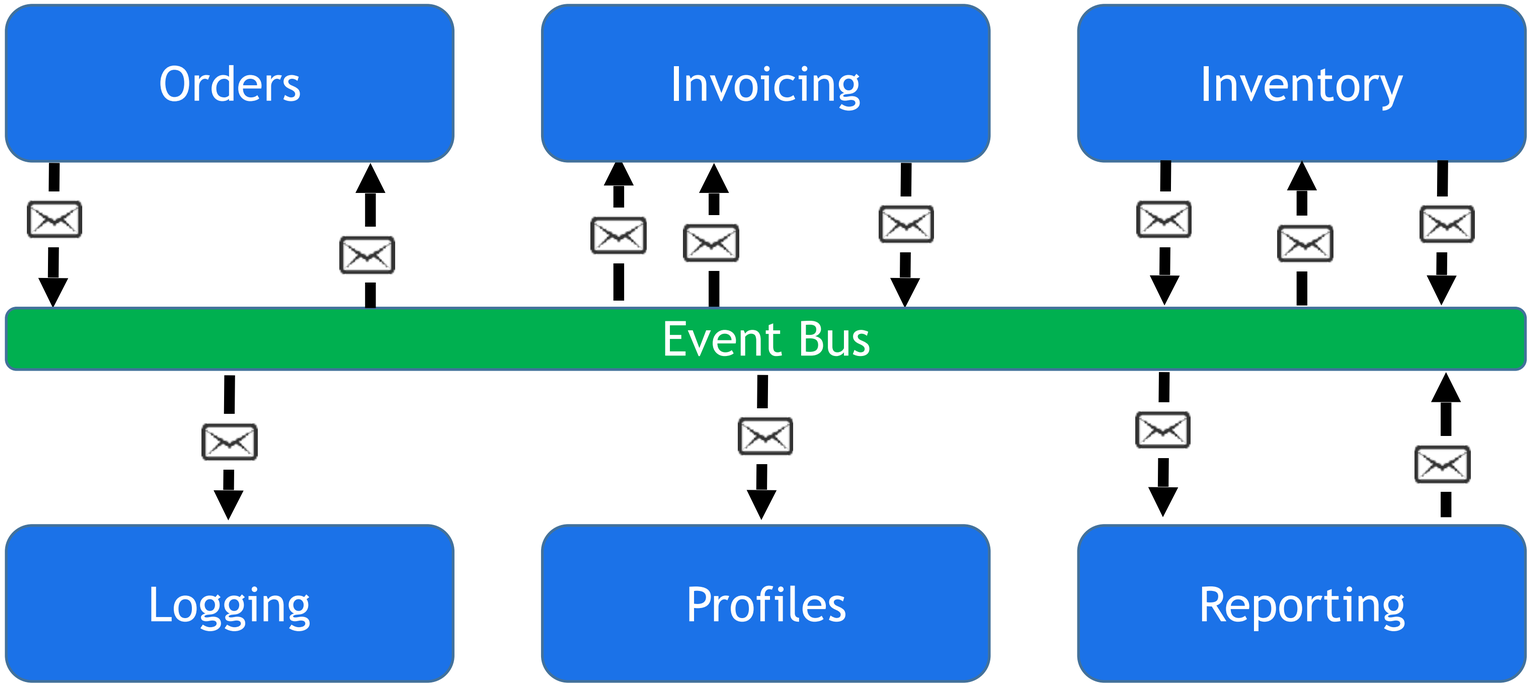
- An Event is represents something that happens in a domain
  - Customer Submits Order
  - Customer Billed
  - Payment Received
  - Order Ready for Shipment
  - Order Shipped
- While Events and their payload are designed at the enterprise system level, their implementations are left to the specific subsystems.

Event Driven Architecture

- No point-to-point integrations
- Loosely coupled, highly scalable systems
- Loosely coupled, TEAMS
- Easier to test
- Easier to change
- Topology agnostic

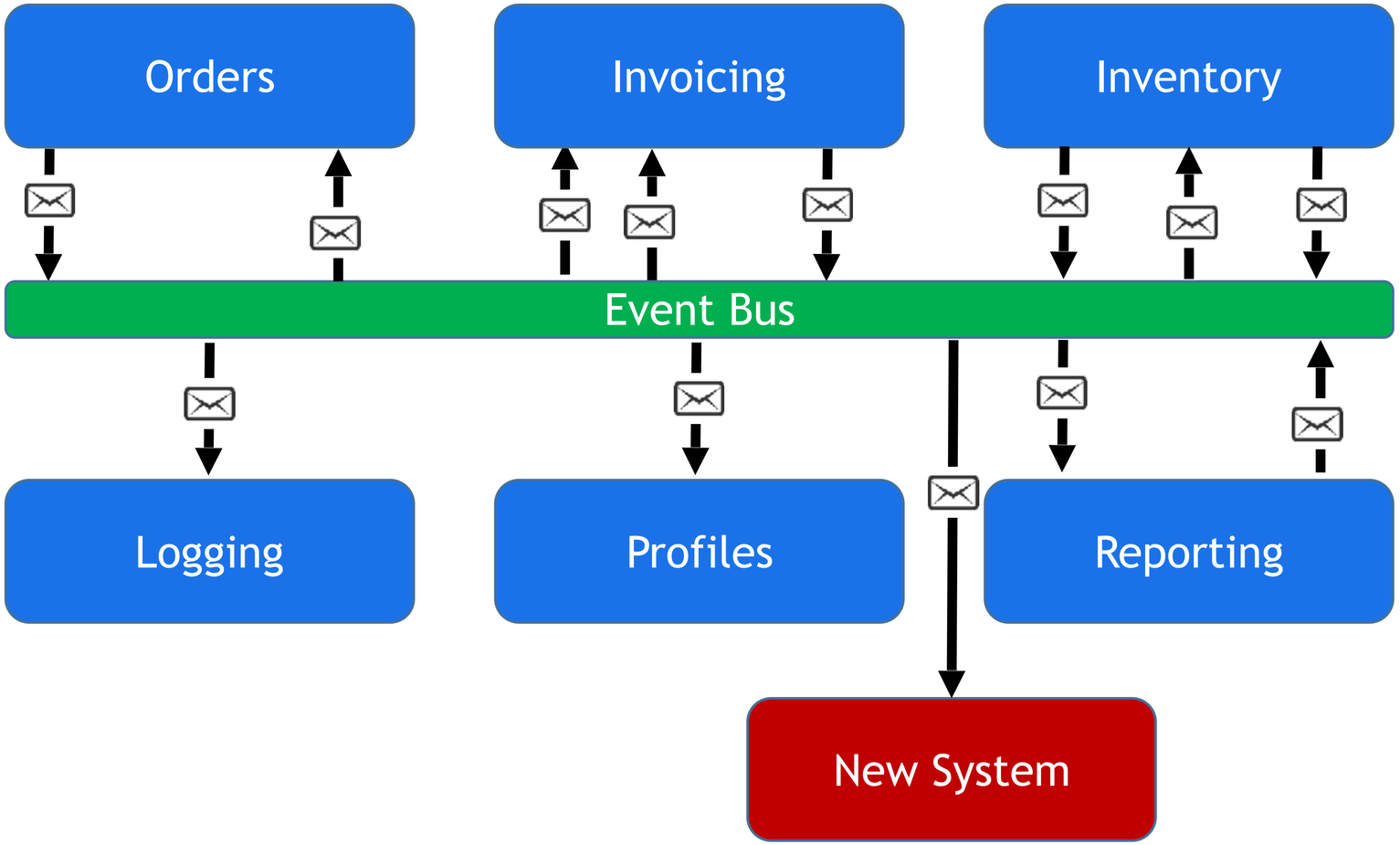
((CENTRIC))

Loosely Coupled



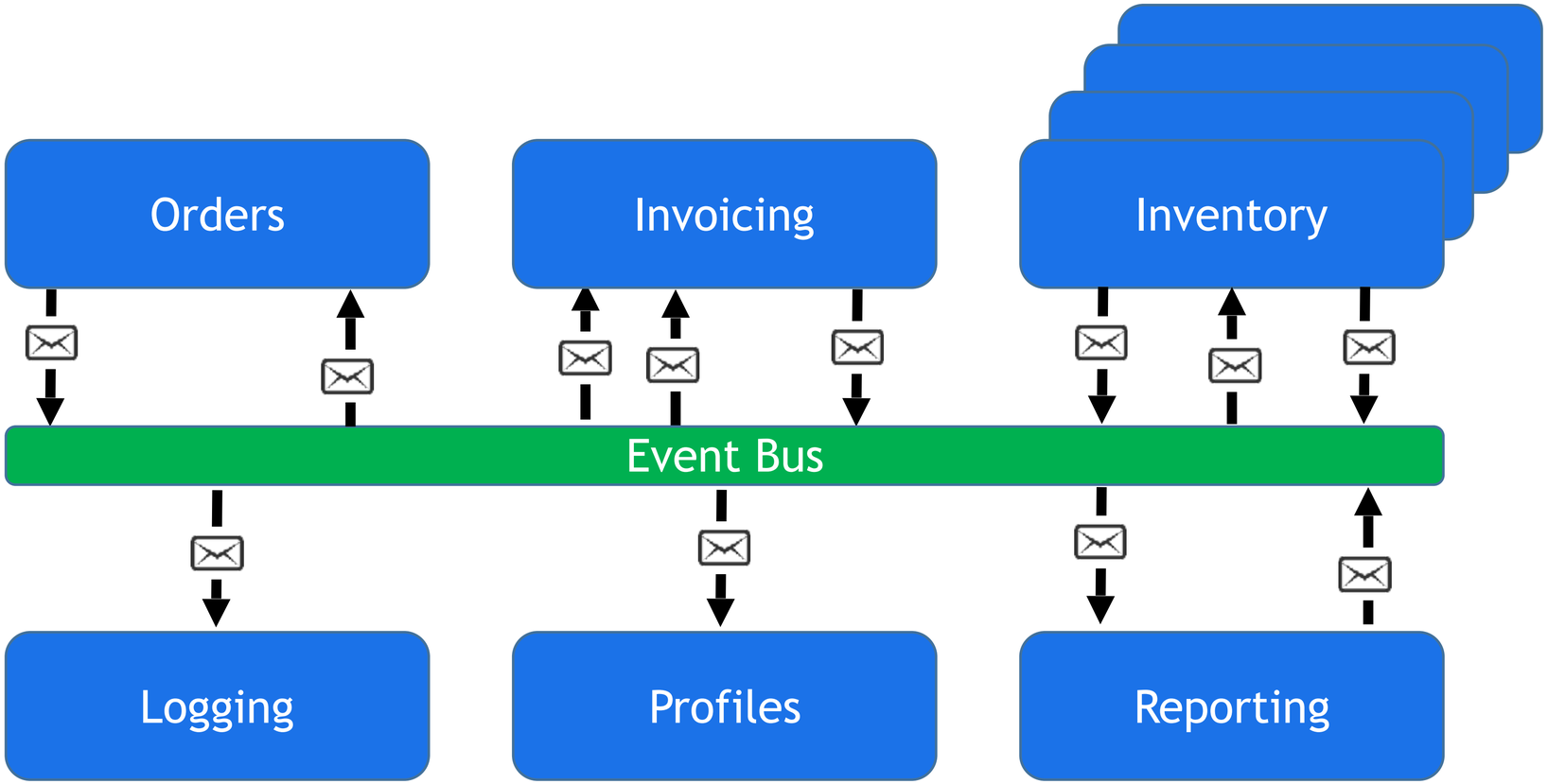
((CENTRIC))

Easy to Integrate New Systems



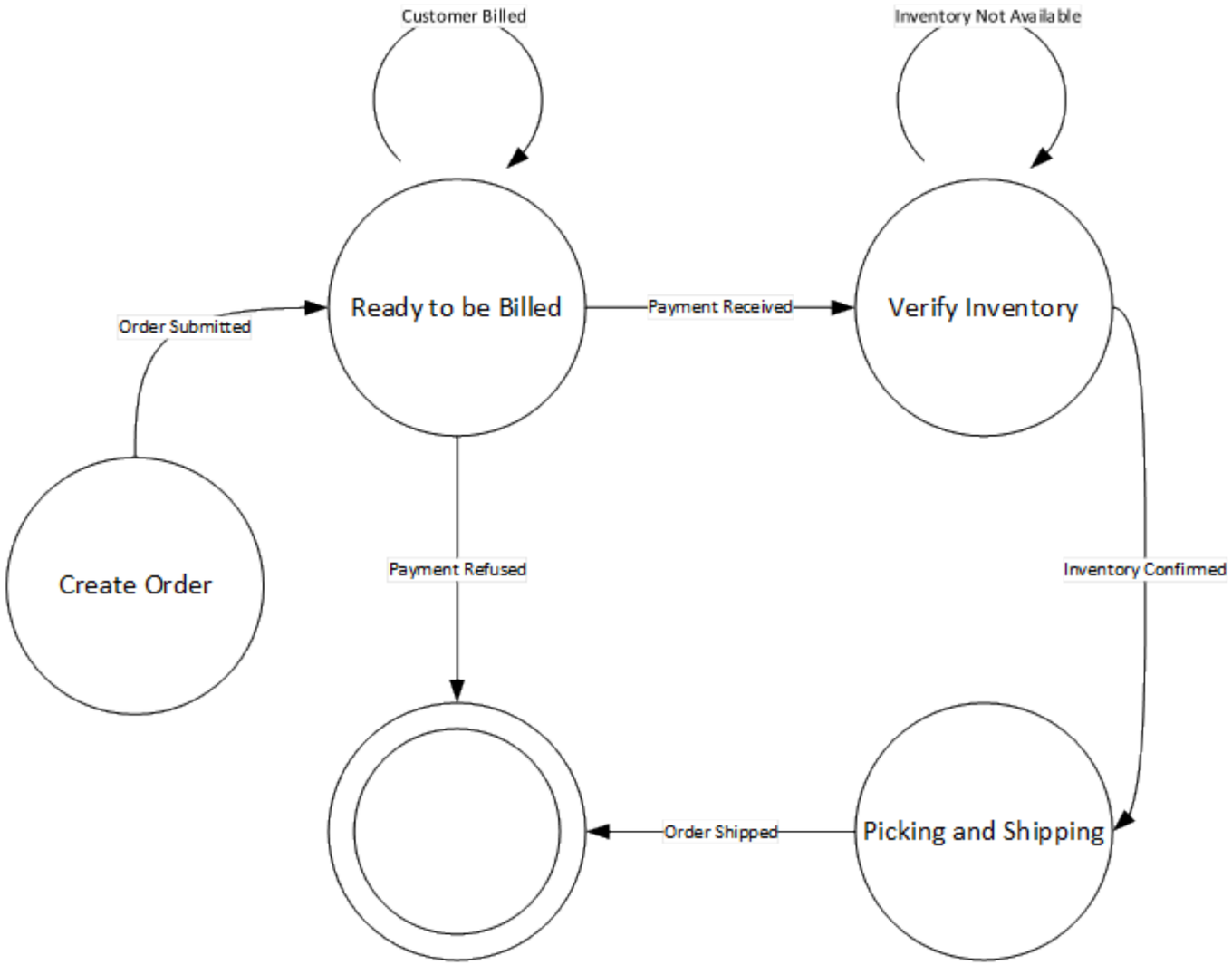
((CENTRIC))

Scales



((CENTRIC))

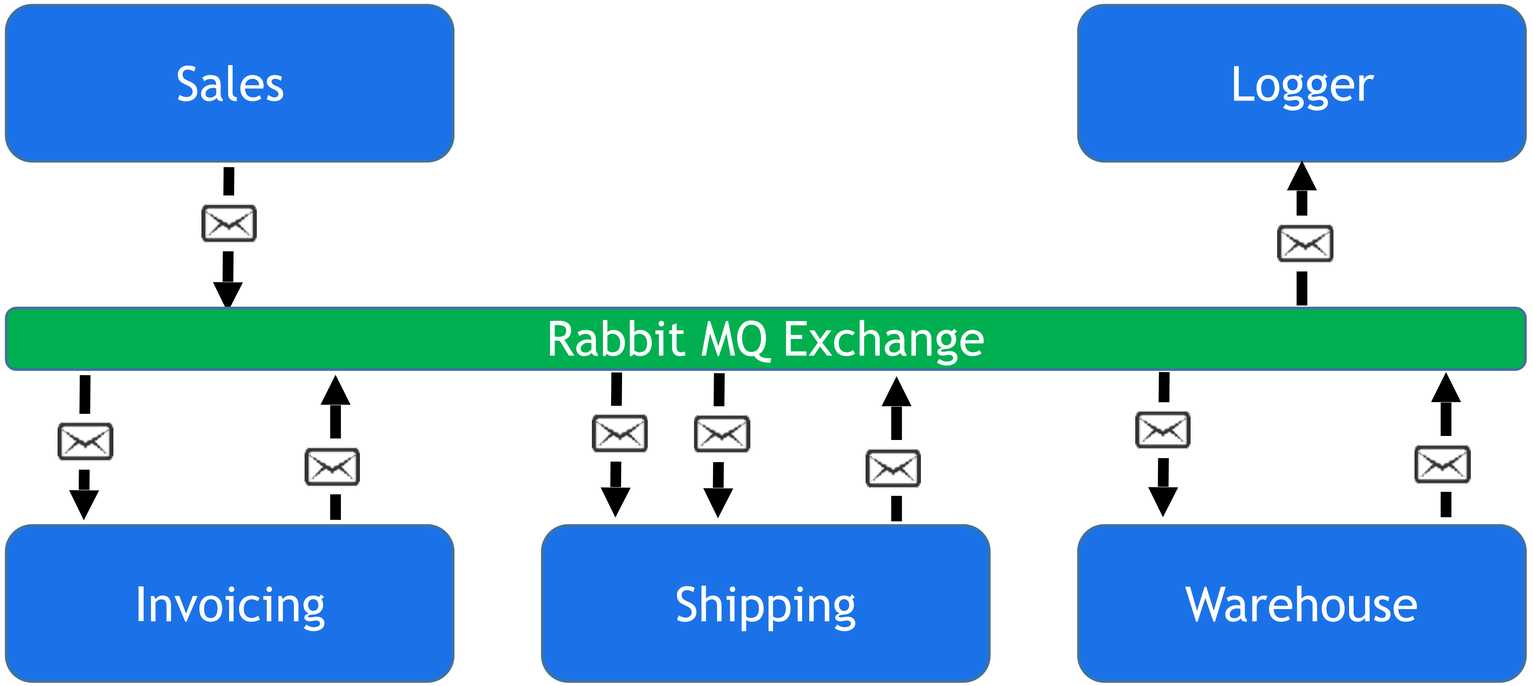
Easily Modeled as a finite state diagram



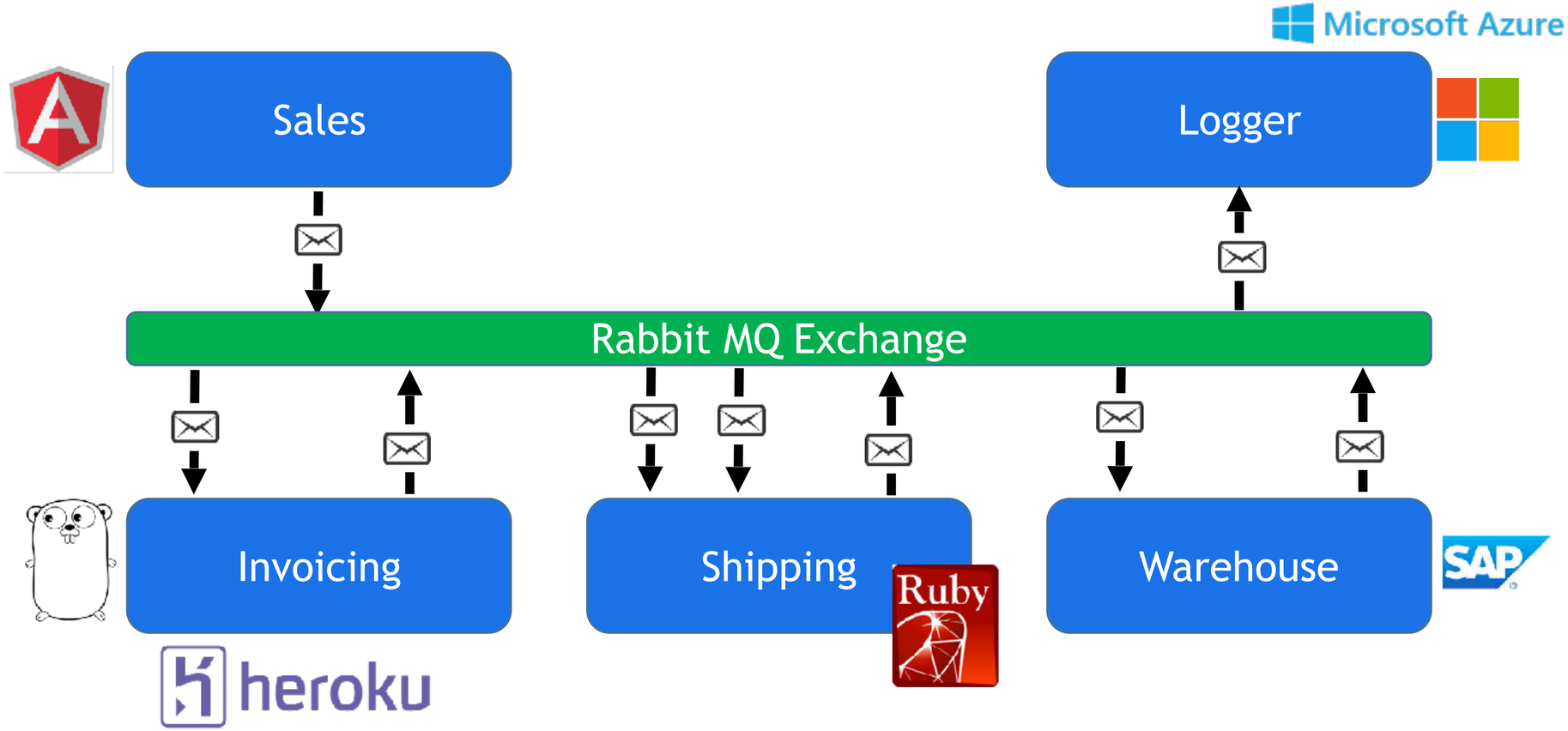
Other benefits

- Ease of 3rd party integration
- Existing systems can be ‘wrapped’
- Can be deliberate about scaling
- Fault tolerant
- Event messages can be logged and ‘replayed’
- Can test subsystems in isolation

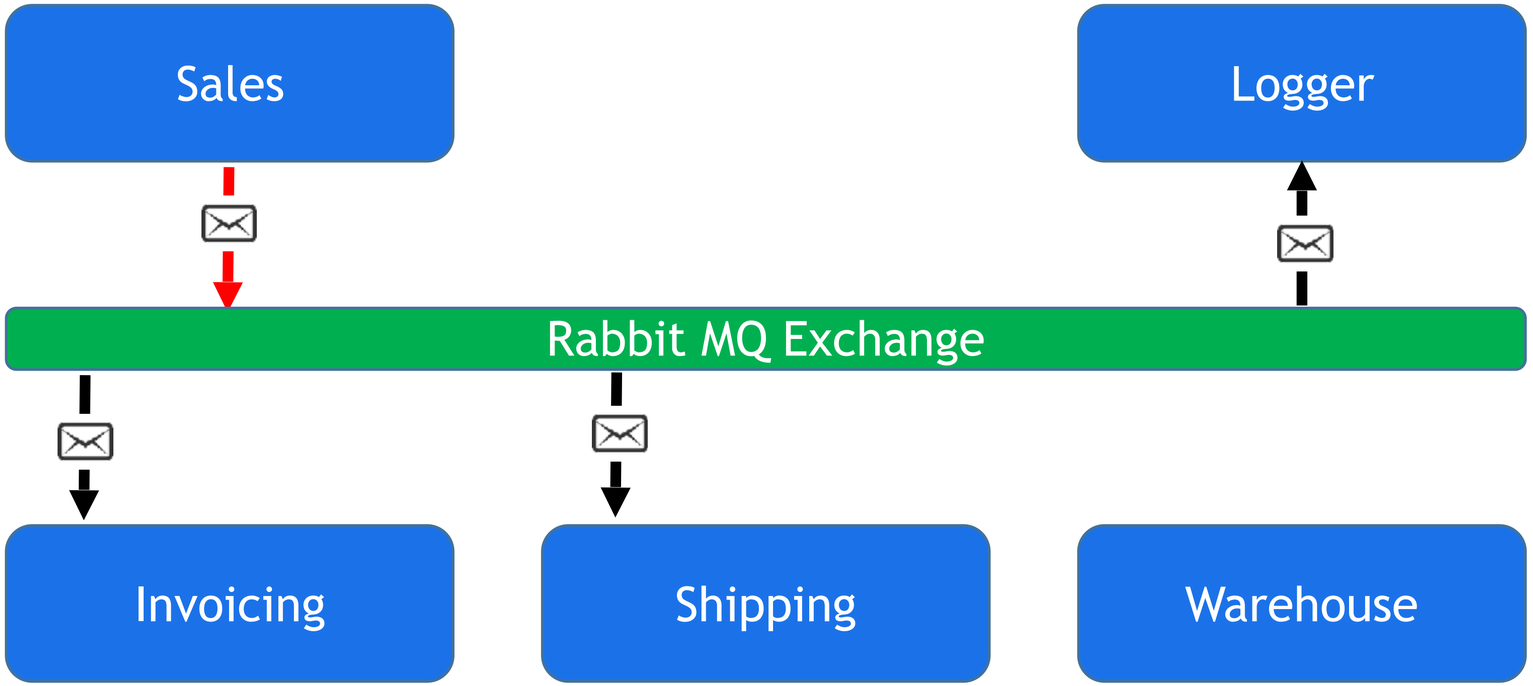
A Retail System



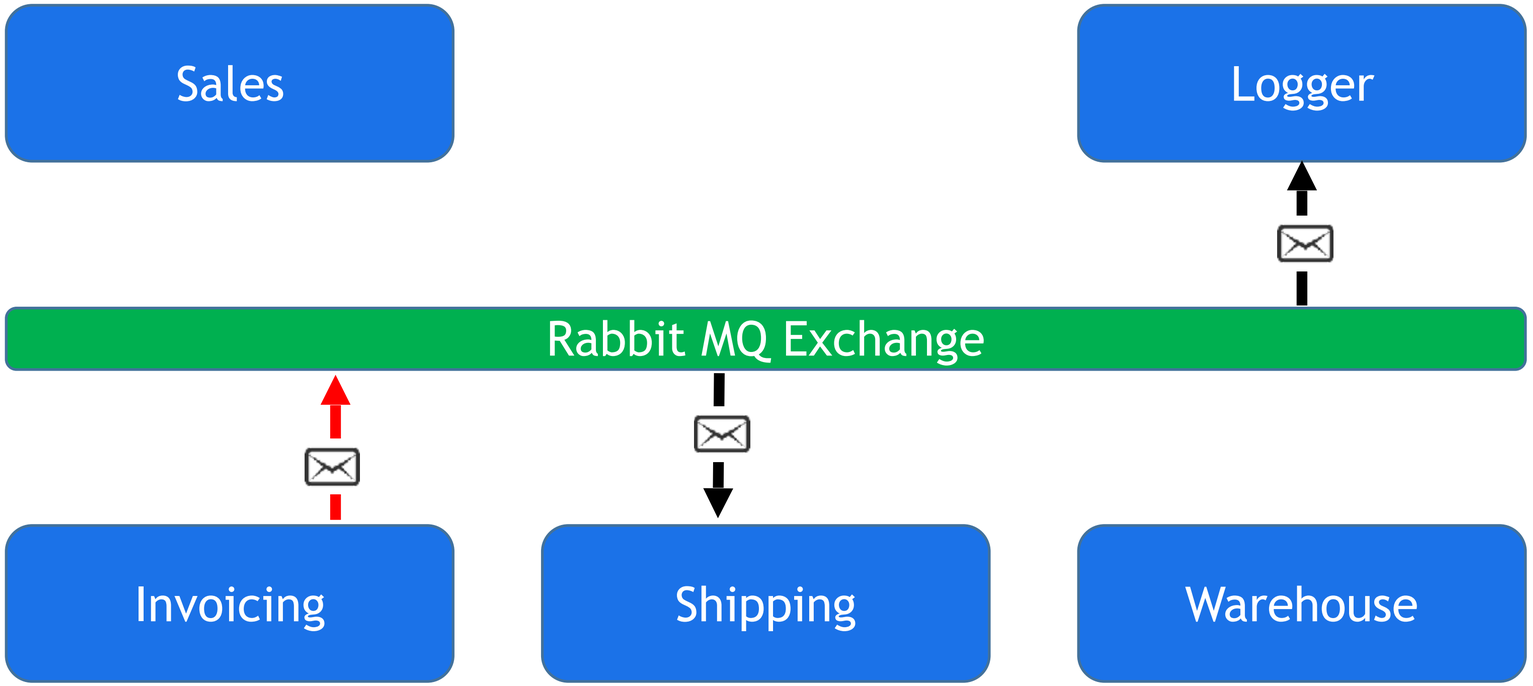
A Retail System – Platform Agnostic



Order Accepted Event is generated

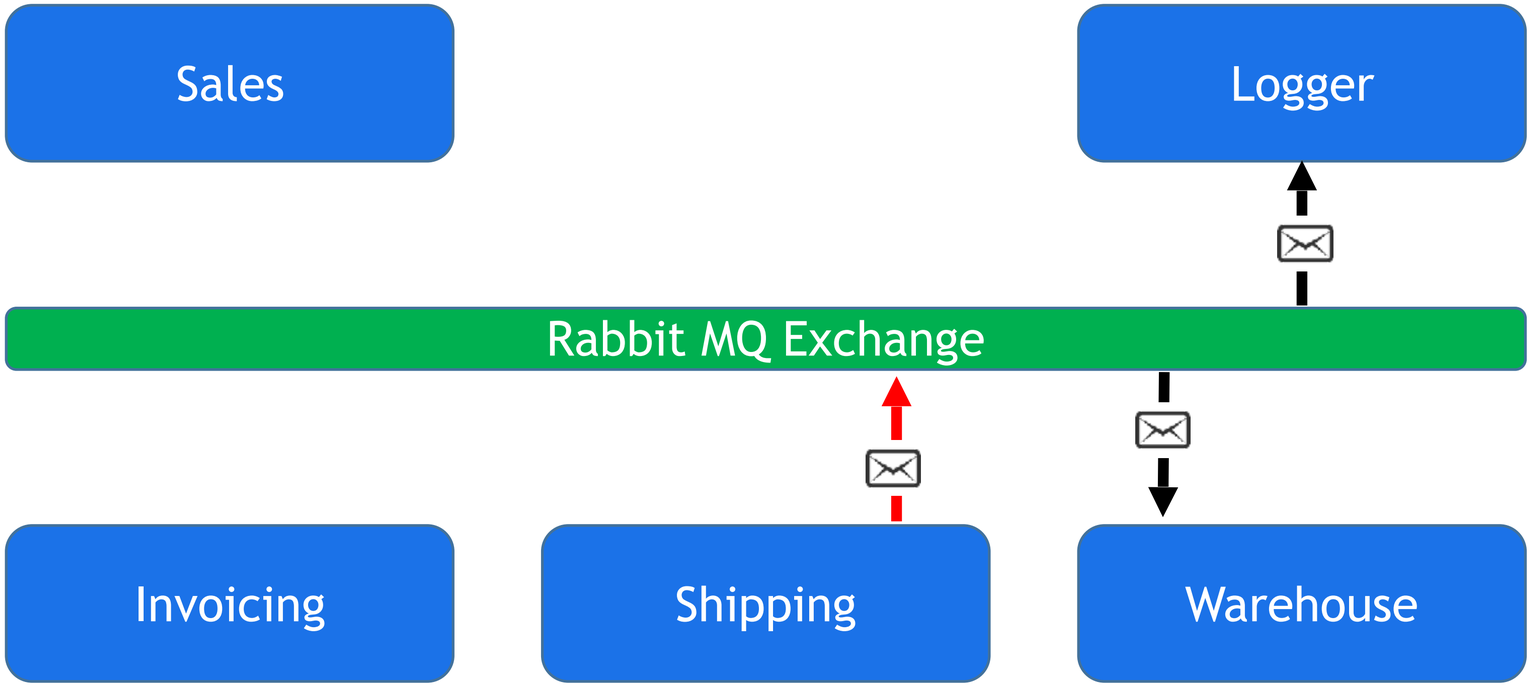


Customer Billed Event



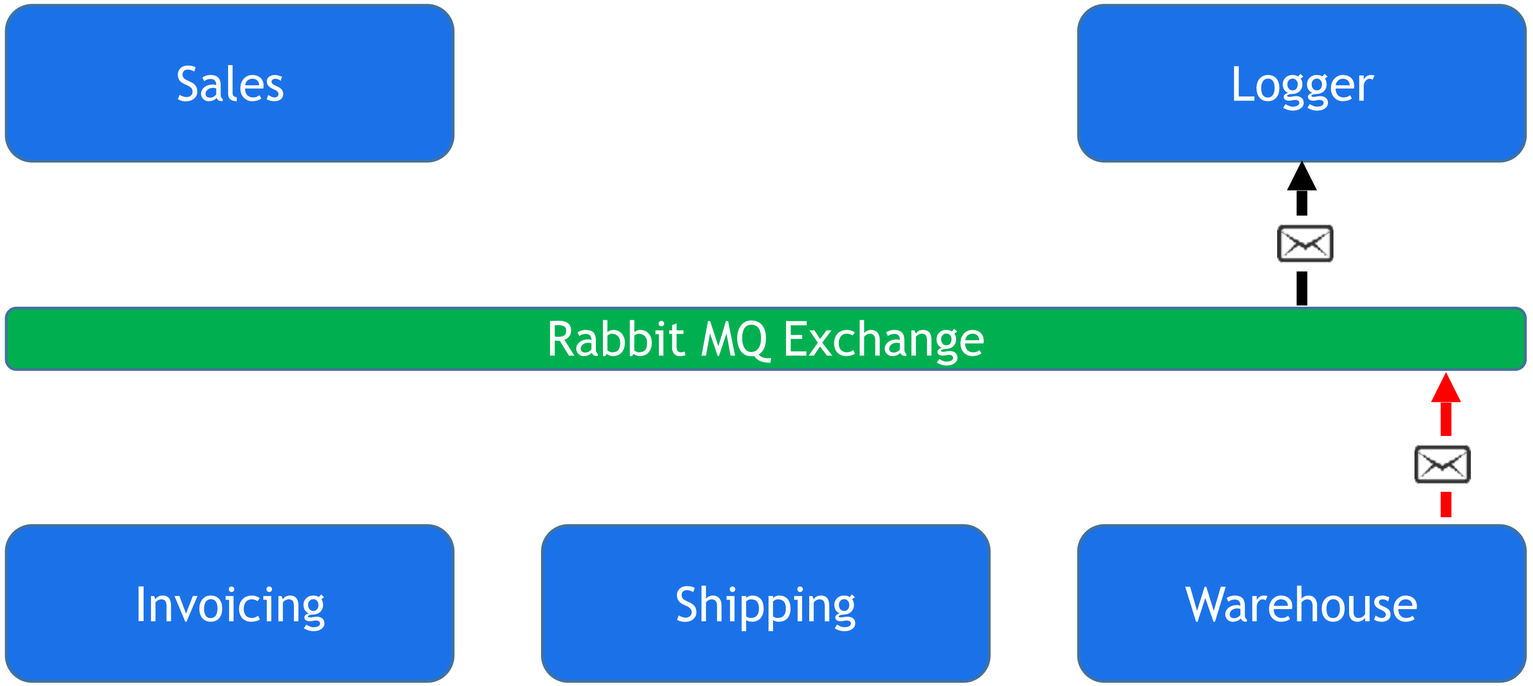
((CENTRIC))

Order Ready for Shipment Event

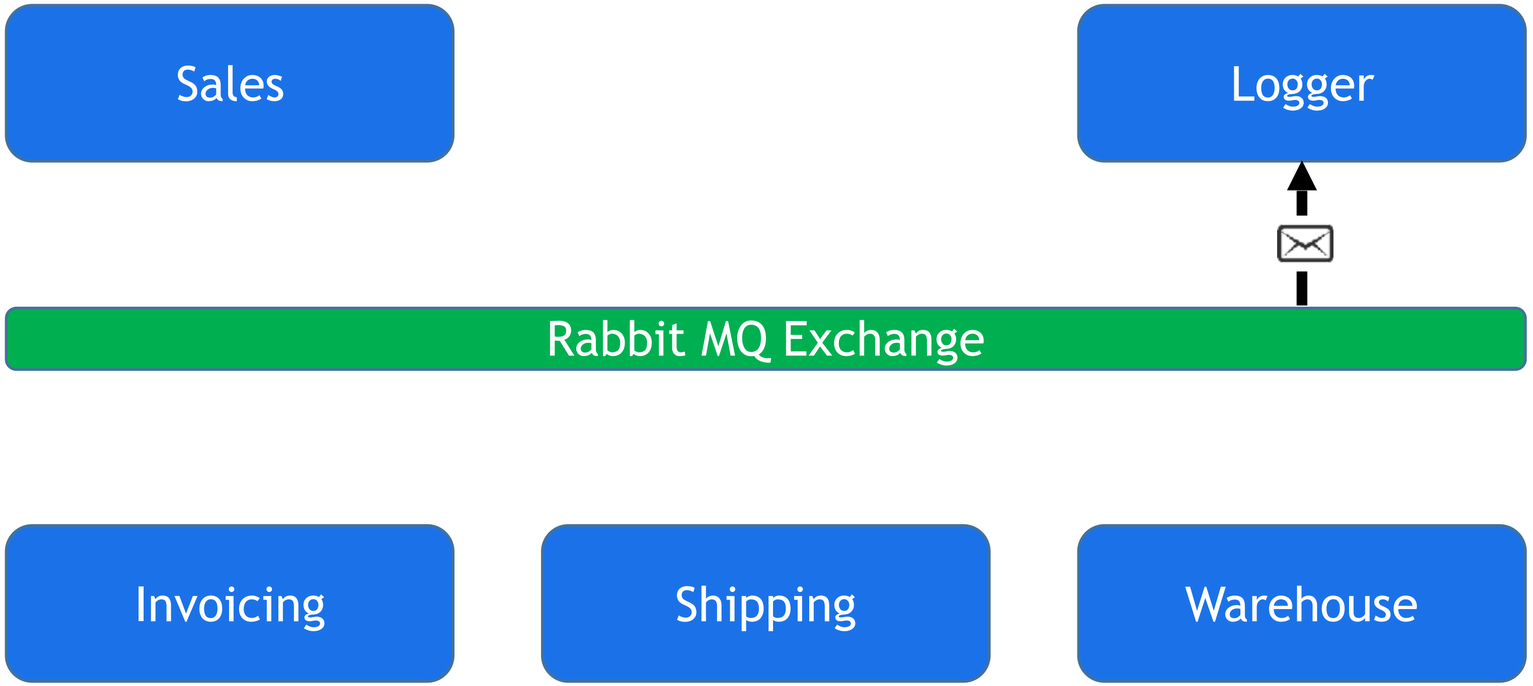


((CENTRIC))

Order Shipped



Logger Consumes EVERY Event



((CENTRIC))

Logger Consumes EVERY Event

Results | Message

Id	OrderId	RoutingKey	Content	WhenReceived
873	2D4BE6EB-6274-4074-8690-C014D3FCA645	order_accepted	{"CustomerId":"834de3b4-fe10-43c6-82a3-6443749c5...	2015-08-10 15:22:16.013
990	2D4BE6EB-6274-4074-8690-C014D3FCA645	customer_billed	{"OrderId":"2d4be6eb-6274-4074-8690-c014d3fca645"}	2015-08-10 15:24:57.343
1169	2D4BE6EB-6274-4074-8690-C014D3FCA645	order_ready_to_ship	{"OrderId":"2d4be6eb-6274-4074-8690-c014d3fca645"}	2015-08-14 16:04:00.827
1371	2D4BE6EB-6274-4074-8690-C014D3FCA645	order_shipped	{"OrderId":"2d4be6eb-6274-4074-8690-c014d3fca645"}	2015-08-14 16:07:05.793

((CENTRIC))

For Further Reading

- THIS presentation (and code) on GitHub

https://github.com/shawnewallace/choreographed_process

- **Stephan Norberg EDA**

http://www.infoq.com/presentations/Domain-Event-Driven-Architecture

- **“Programming Without a Call Stack”**

http://www.enterpriseintegrationpatterns.com/docs/EDA.pdf

- **“Event-Driven Architecture Overview”**
- http://www.omg.org/soa/Uploaded%20Docs/EDA/bda2-2-06cc.pdf

((CENTRIC))

Shawn Wallace

((CENTRIC))

www.centricconsulting.com