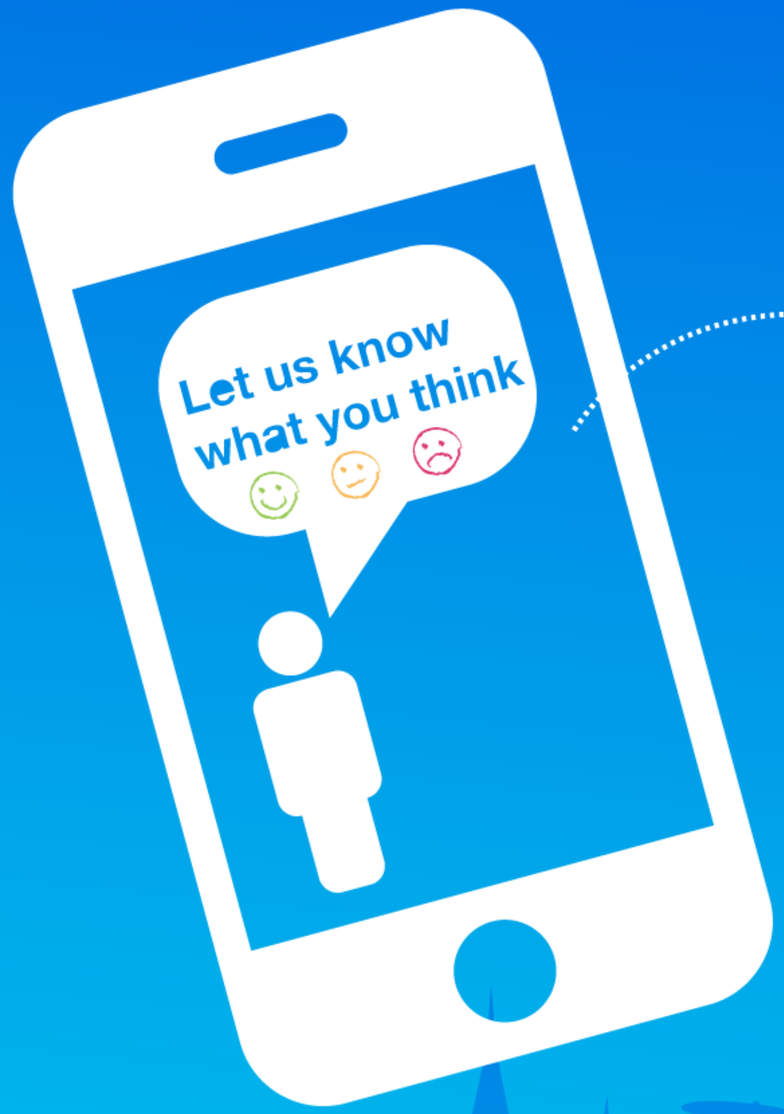
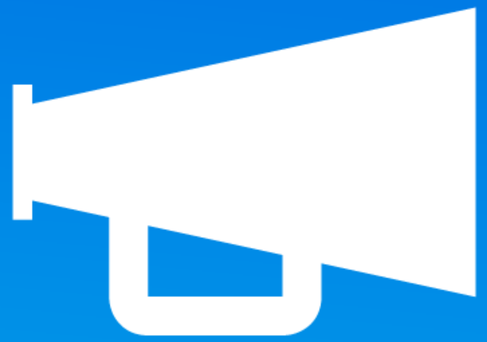
goto; conference

Reactive Systems

Dave Farley
http://www.davefarley.net
@davefarley77

goto; conference





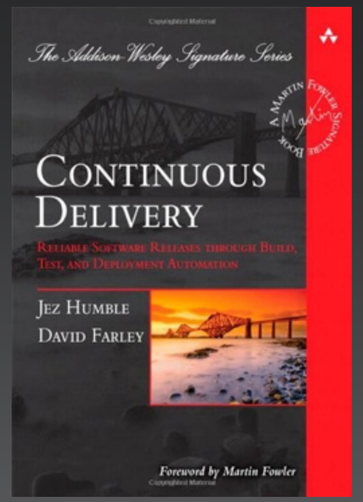
Click ‘engage’
to rate sessions
and ask questions

Join the conversation #gotocph

Reactive Systems

21st Century Architecture for 21st Century Problems

Dave Farley
http://www.davefarley.net
@davefarley77



Continuous Delivery ltd

http://www.continuous-delivery.co.uk

Our World Is Changing

Large Applications circa 2005:

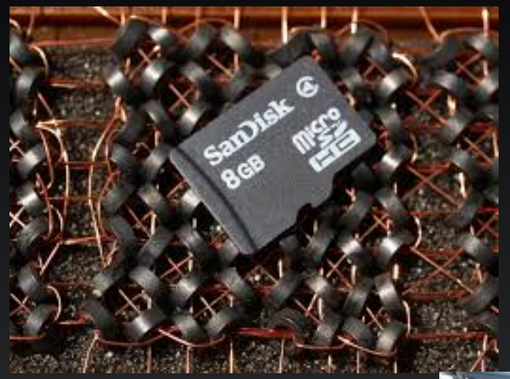
- 10's of Servers
- Seconds of Response Time
- Hours of Offline Maintenance
- Gigabytes of Data

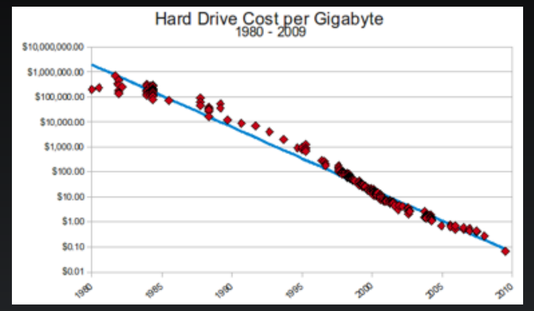
Large Applications Now:

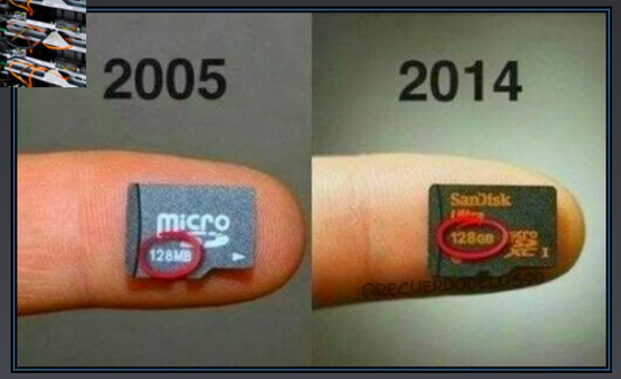
- Handheld Devices to 1000's of multi-core processors
- Millisecond Response Time
- 100% Uptime
- Petabytes of Data

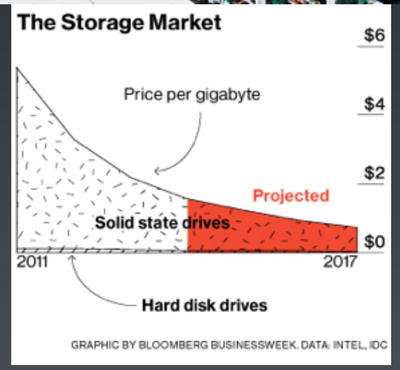
Continuous Delivery ltd

Our World Is Changing





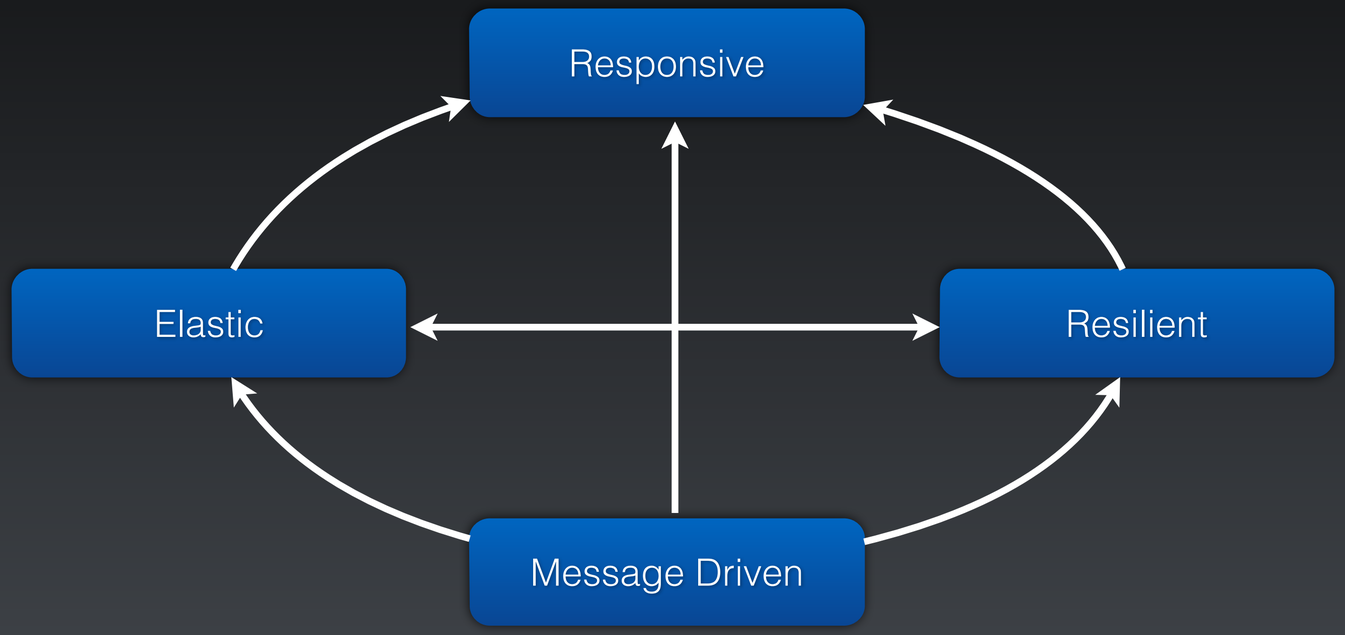




Continuous Delivery ltd

The Reactive Manifesto

"21st Century Problems are not best solved with 20th Century Software Architectures"



The Evolution of modern hardware has changed many of the common assumptions of software development

Source: www.reactivemanifesto.org

Continuous Delivery ltd

Reactive Systems Are:

Responsive:

- Responds in a Timely Manner
- Cornerstone of Usability
- Also Quick to Detect Problems

Reactive Systems Are:

Resilient:

- Remains Responsive in the Face of Failure
- Resilience Depends on - Replication, Containment, Isolation and Delegation

Continuous Delivery ltd

Reactive Systems Are:

Elastic:

- Remains Responsive Under Varying Workload
- Responds to Change in the Input Rate By Increasing or Decreasing Resources that Service the Input
- Decentralised Architecture, No Contention Points, No Central Bottlenecks

Continuous Delivery ltd

Reactive Systems Are:

Message Driven:

- Asynchronous Message Passing is the foundation for all of these properties
- Loose-Coupling, Isolation, Location Transparency
- Ability to Delegate Errors

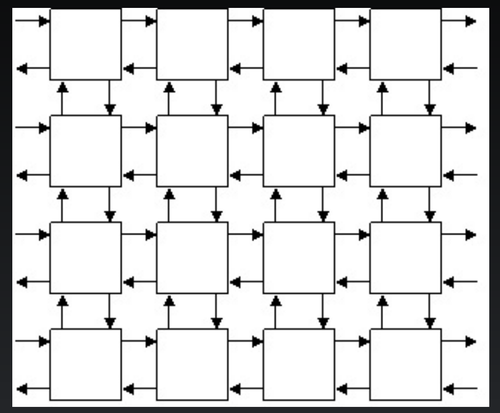
Continuous Delivery ltd

Properties of Reactive Systems

- Flexible
- Loosely-Coupled
- Scalable
- Easier to Develop
- More Tolerant of Failure
- Respond to Failure Gracefully
- Responsive to Users

Continuous Delivery ltd

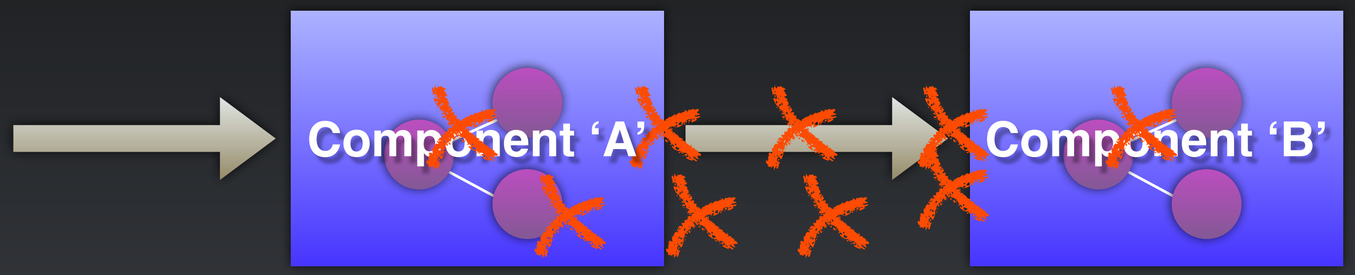
Fractal Architecture



- Large Systems Are Composed of Smaller Ones
- They Depend on the Reactive Properties of Their Constituents
- These Benefits Operate At All Scales
- Such Systems are Composable

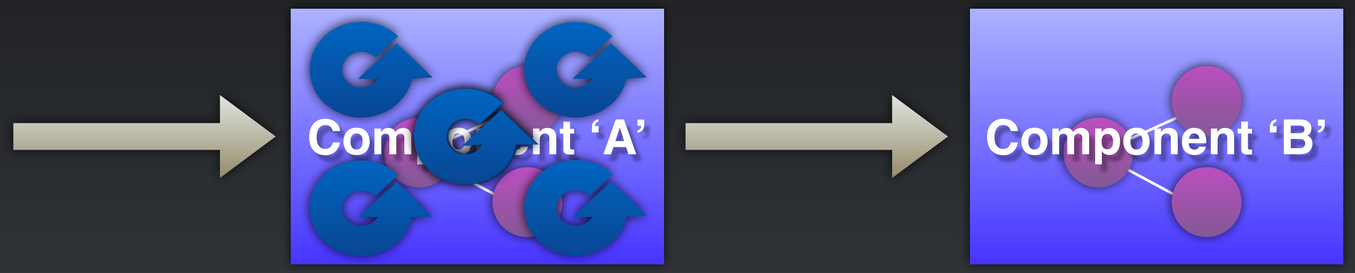
Continuous Delivery ltd

Failure Modes in Synchronous Messaging



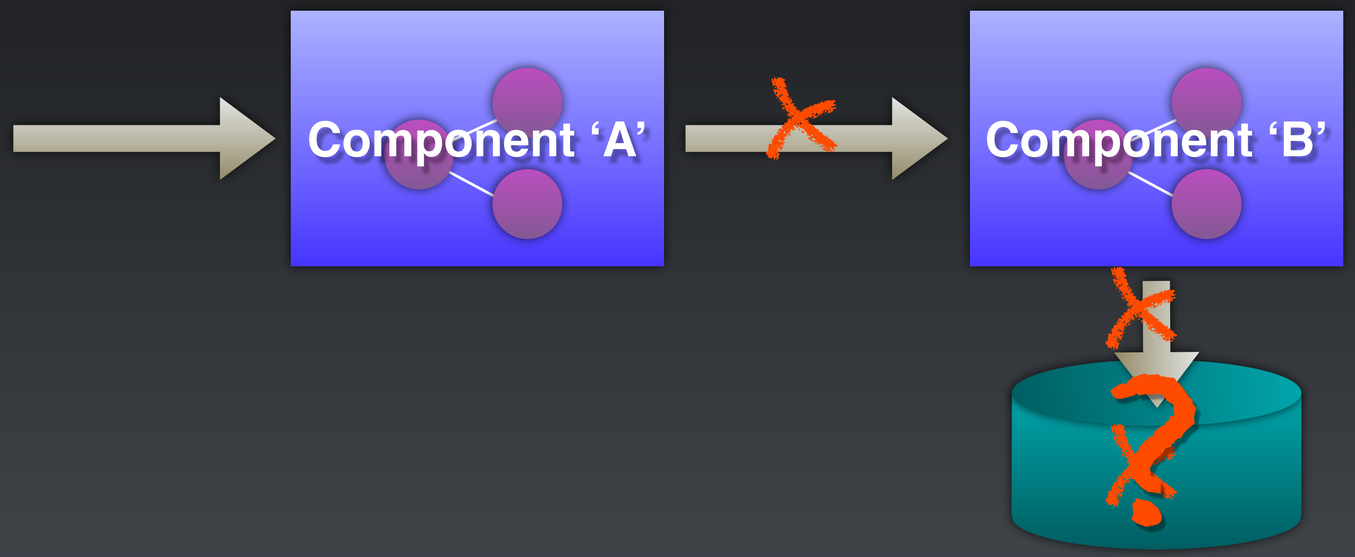
Continuous Delivery ltd

Synch Messaging Breeds Complexity



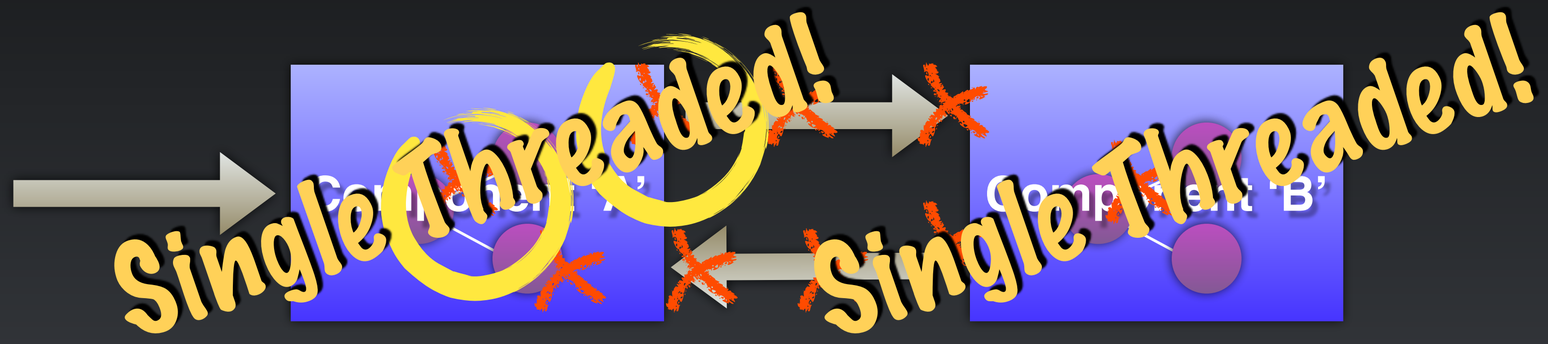
Synchronous Comms Increases Coupling in Location and Time

Synch Messaging Breeds Complexity

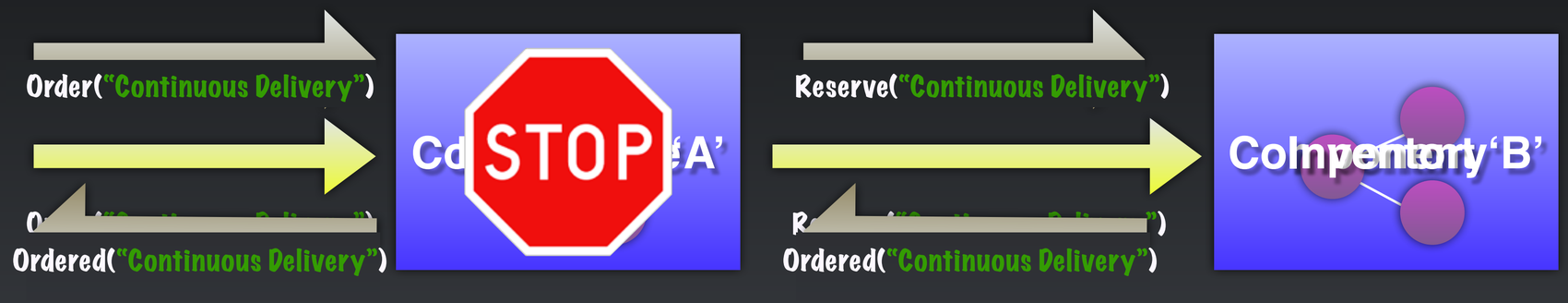


Continuous Delivery ltd

The Benefits of Asynchrony

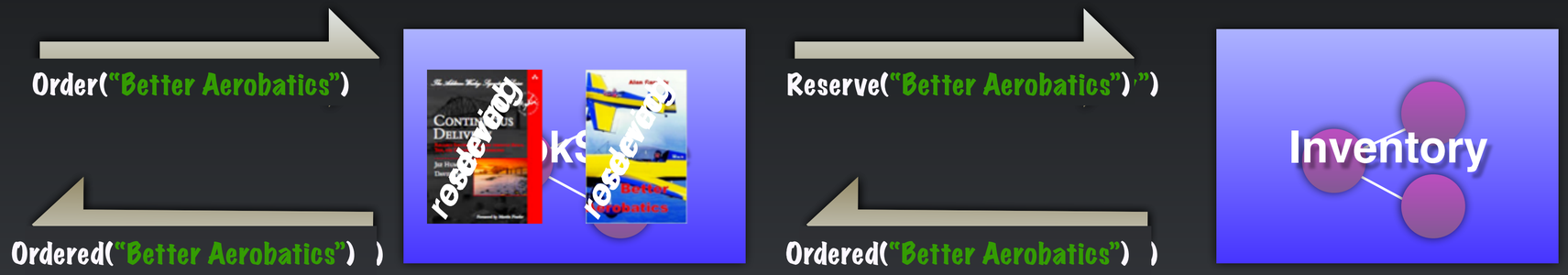


An Example

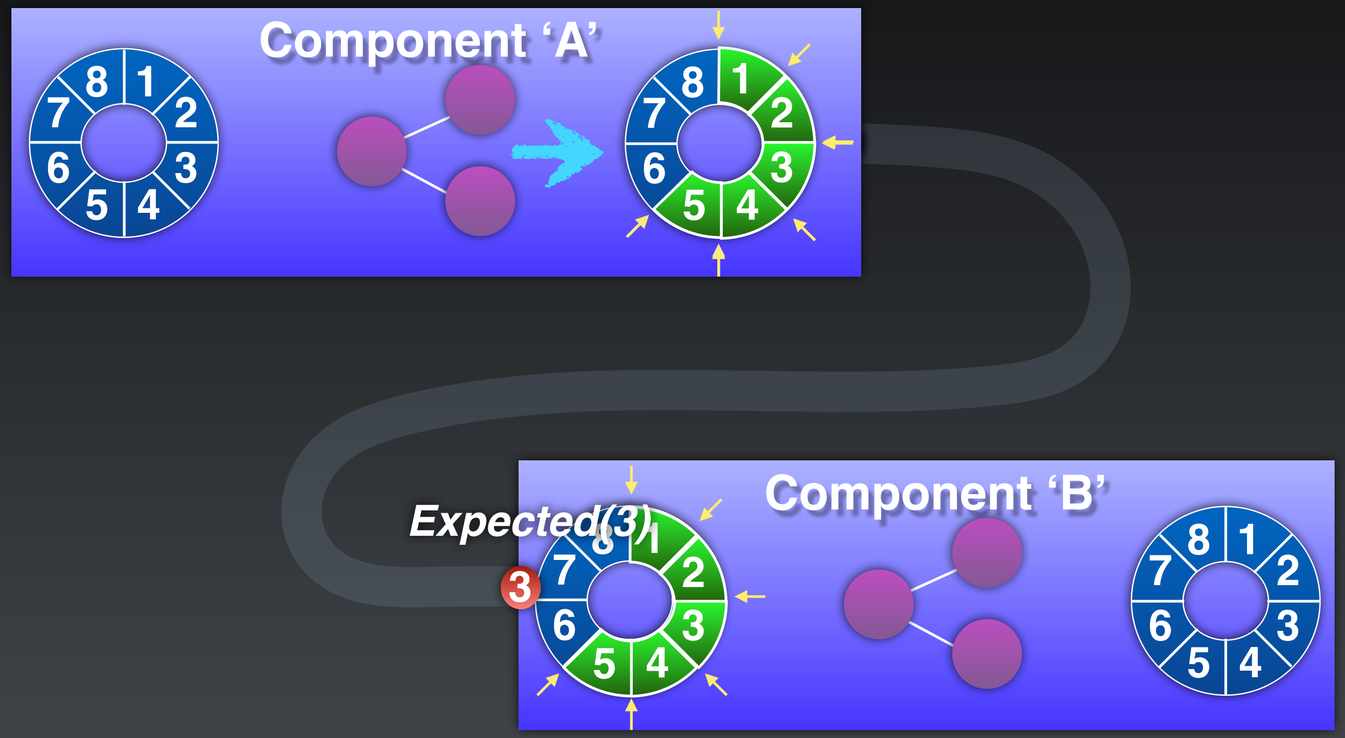


Continuous Delivery ltd

An Example



An Example of Idempotence

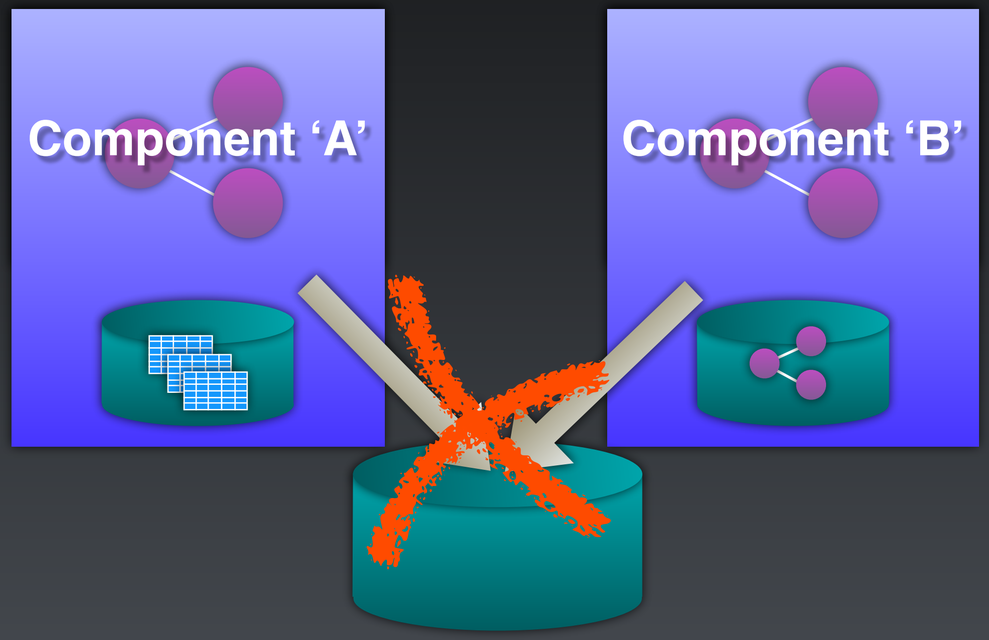


Continuous Delivery ltd

Isolation

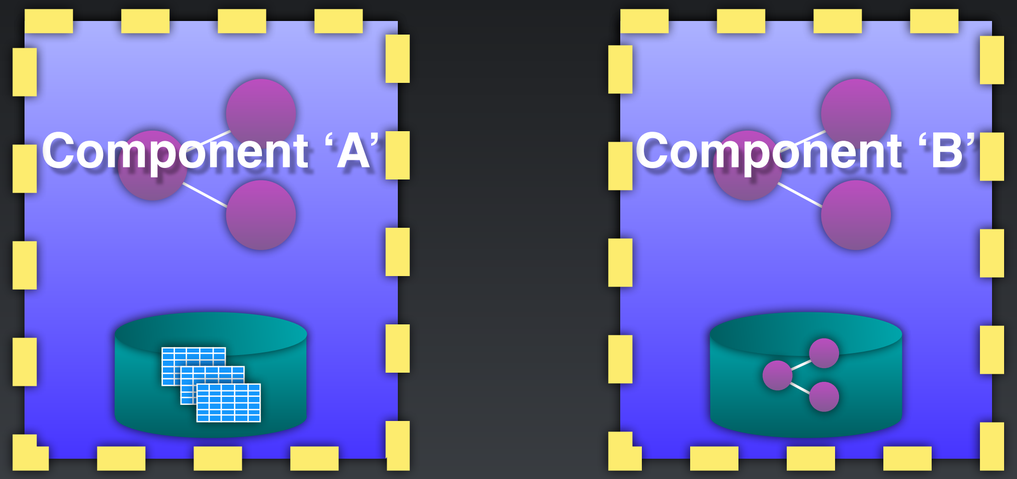
- Decoupling in Time and Space
  - Time - Sender and Receiver have independent lifecycle
  - Space - Location Transparency
- Share Nothing!
- Built on Inter-Component Communication over Well Defined Protocols

Isolation



Continuous Delivery ltd

Share Nothing



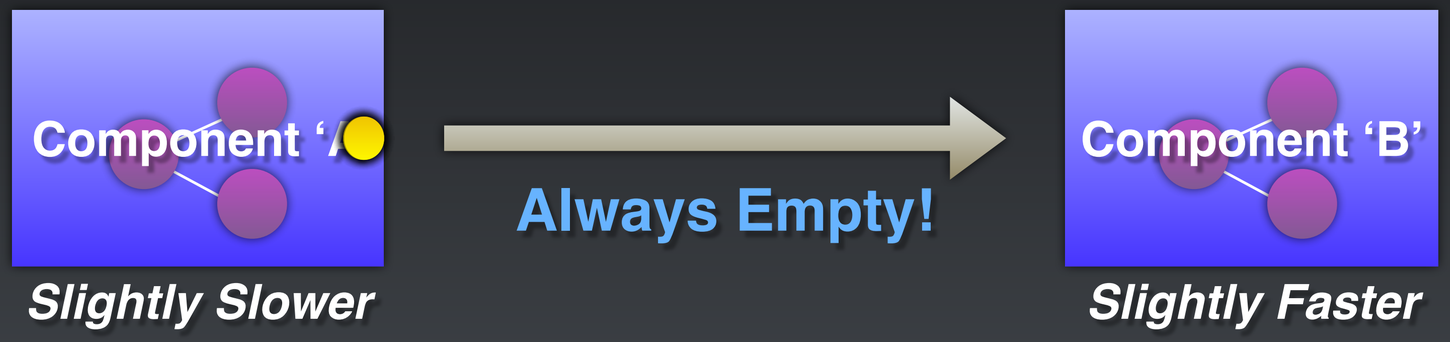
Back-Pressure

- You Can’t Isolate Stress
- The System as a Whole Needs to Respond Sensibly
- Unacceptable For a Stressed Component to Fail Catastrophically or Loose Messages
- Queues Represent An Unstable State - Load
- Components Under Stress Need to Reflect This By Applying Back-Pressure, Slowing Upstream Inputs

Continuous Delivery ltd

Queues Represent an Unstable State

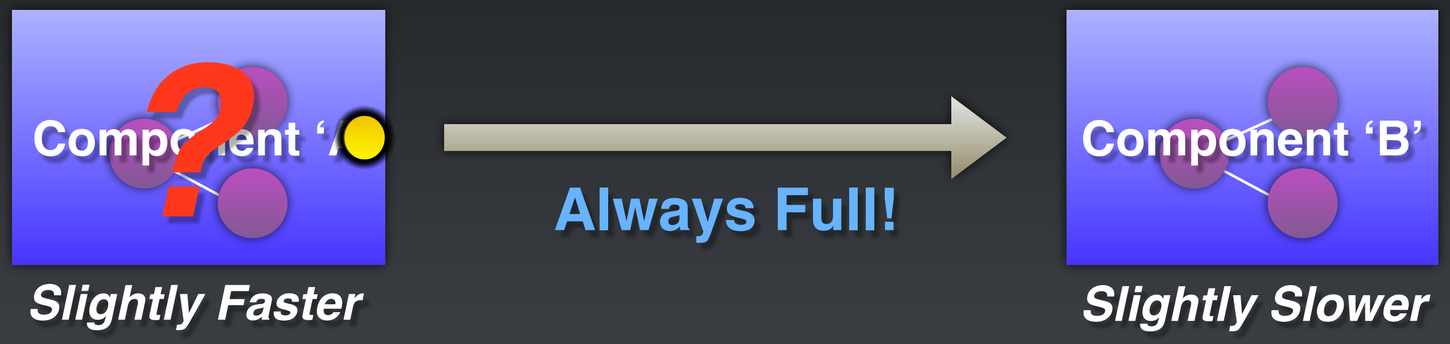
Queues are always full or always empty.
Anything else is transitional, on its way to full or empty.



Continuous Delivery ltd

Queues Represent an Unstable State

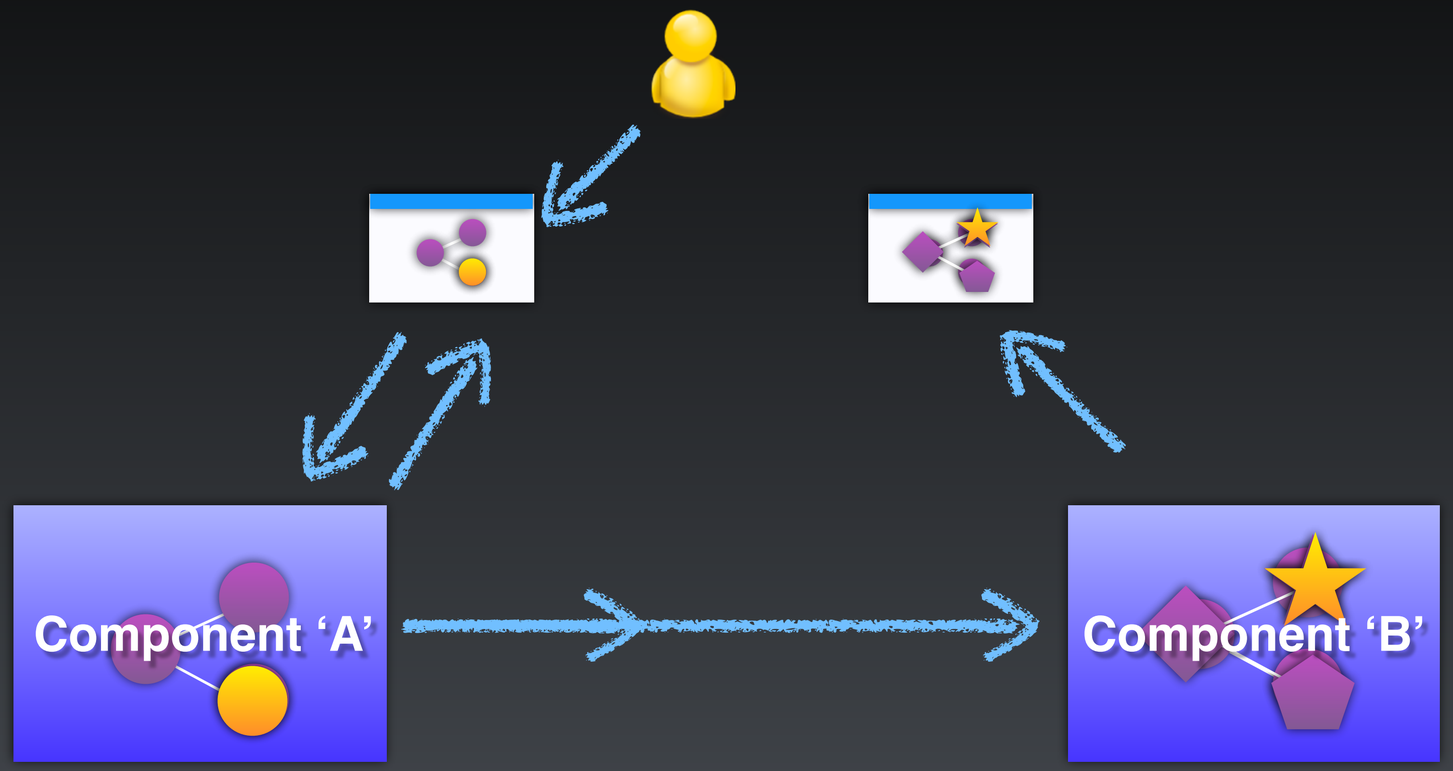
Queues are always full or always empty.
Anything else is transitional, on its way to full or empty.



Continuous Delivery ltd

Back-Pressure

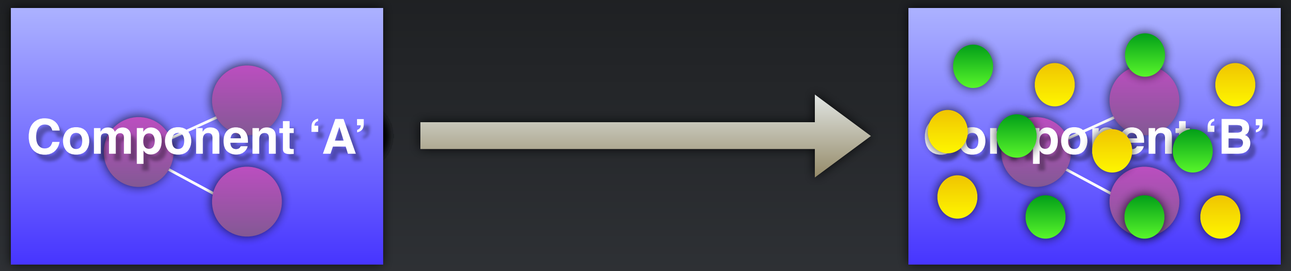
Eventual Consistency



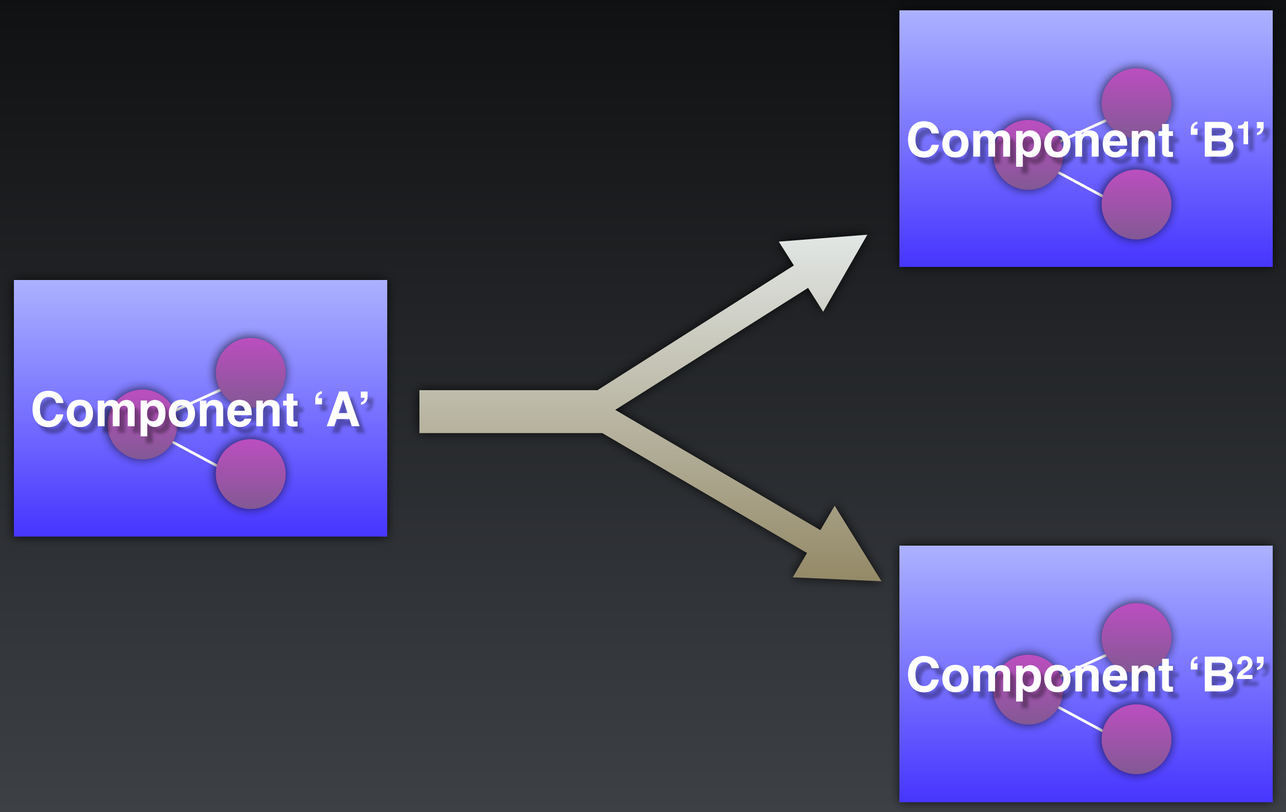
Location Transparency

- Elastic Systems Need To React To Changes In Demand
- We Are All Doing Distributed Computing
- Embracing This Means There Is No Difference Between Horizontal (Cluster) and Vertical (Multicore) Scalability
- Components Should Be Mobile
- One Pattern For Communications
  - Local Communications Is An Optimisation

Linear Scalability Through Sharding



Linear Scalability Through Sharding



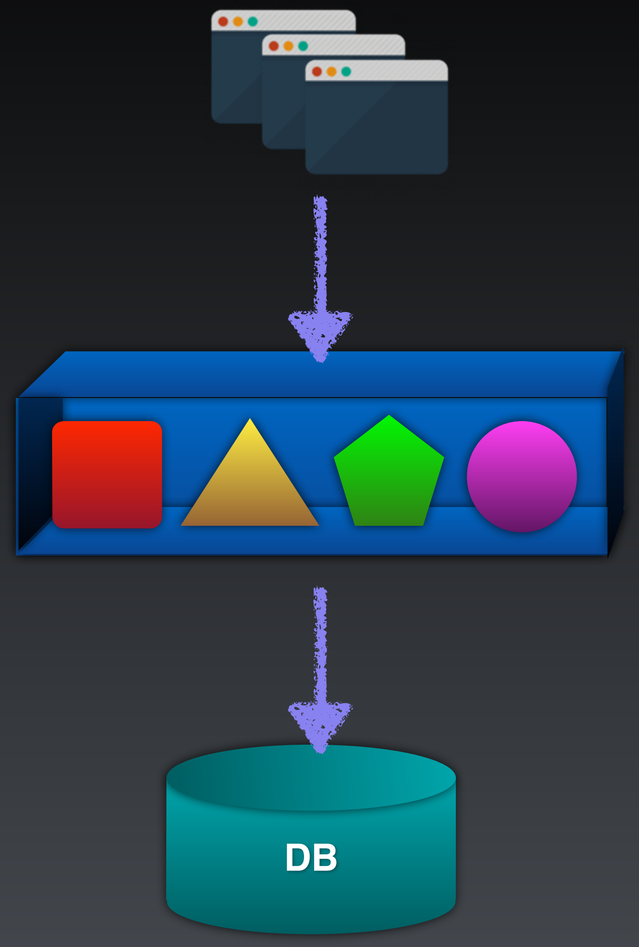
Continuous Delivery ltd

Modern Hardware Should Change Our Assumptions

- For Efficient Software, The Biggest Cost is Shifting Data
- RAM is Not Random Access
- Disk is Not Random Access
- SSD is Not Random Access
- RAM is Slow, Network is Slow, Disk is Slower

Conway's Law

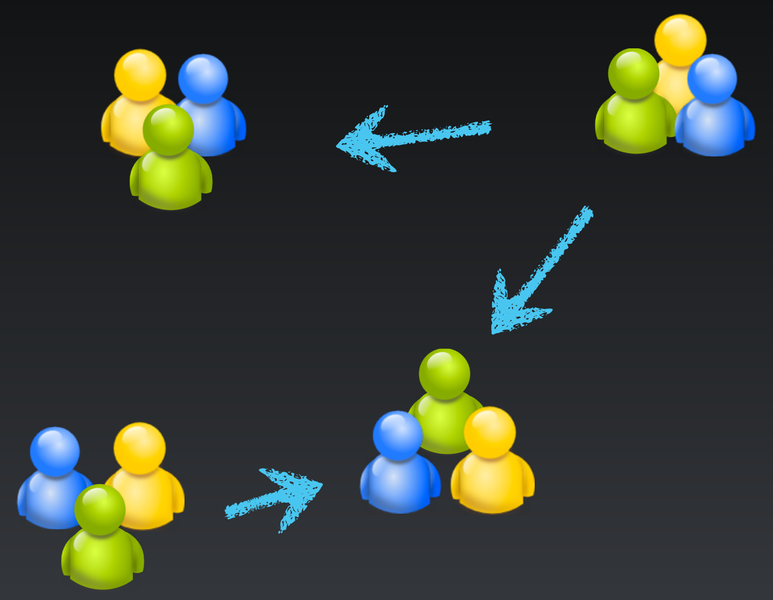
Siloed Teams



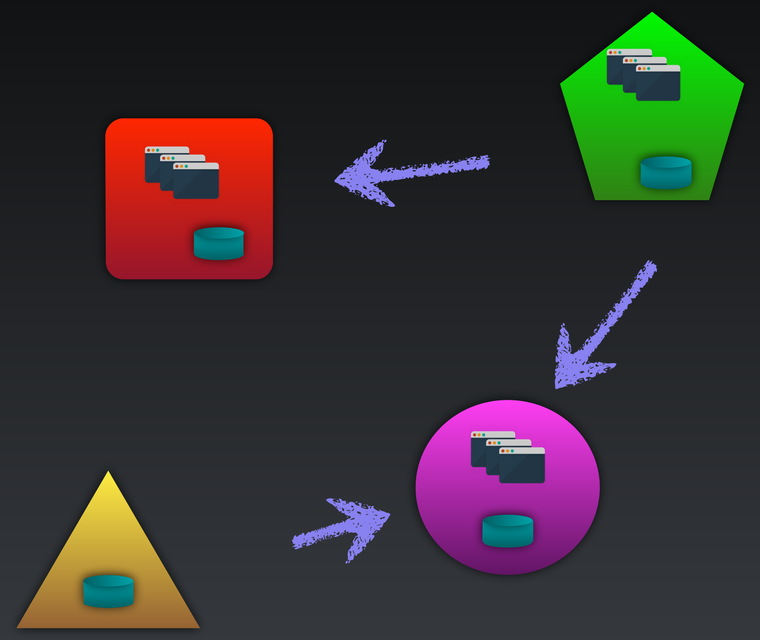
Rigid Architecture

Continuous Delivery ltd

Conway's Law



Cross-Functional Teams Organised by Business Function



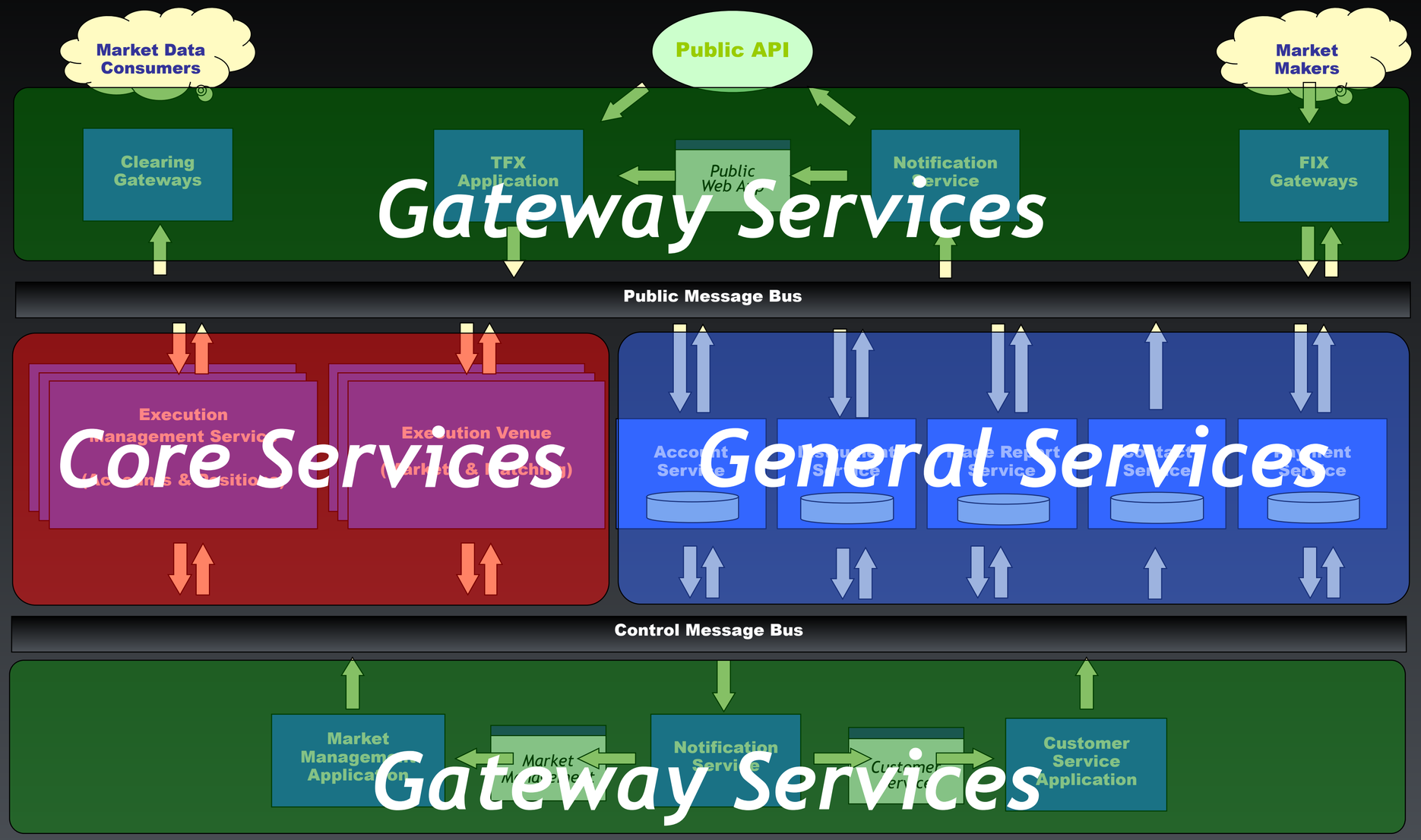
Distributed Service Architecture

Continuous Delivery ltd

Bounded Contexts

- Each Component Is Autonomous and Isolated
- It Only Communicates Through Well Defined Protocols
- Works Best When Components Are Aligned With Bounded Contexts
- Bounded Contexts Are A Concept From Domain Drive Design (DDD) - Eric Evans
- Bounded Context:
  - The Context Within Which A Model Of the Problem Domain Make Sense
  - There Should Be ‘Translations’ Between Bounded Contexts - No “One Model To Rule Them All”
  - Bounded Contexts Tend To Align With Business Functions
  - Best Way To Decompose Organisations And Systems

Example Reactive, MicroService architecture



Continuous Delivery ltd

Where to start?

Q&A

http://www.continuous-delivery.co.uk

Dave Farley

http://www.davefarley.net

@davefarley77

Continuous
Delivery ltd

goto; conference

Please

Remember to rate this session

Thank you!

Join the conversation #gotocph