Building Reactive Microservices

DESIGNING FAILURE TOLERANT MICROSERVICES

Matthew Alexander
SOFTWARE ENGINEER

@alexandermj

Overview

Common architectural patterns

Monoliths and microservices

Resilience

Reactive Systems

Globomantics’ document translation platform

Architectural Patterns

Hexagonal architecture

Layered architecture

Hexagonal Architecture

Provides users with a mechanism to consume electricity

Implements a standard interface for consuming electricity

Same idea applies to application design

Layered Architecture

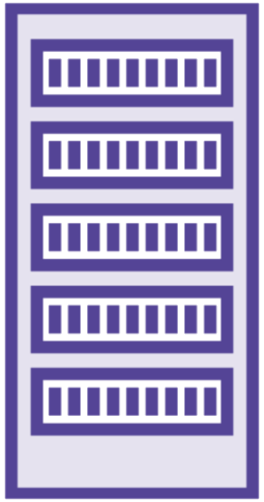
Presentation

Integration

Business

Database

Schools of Thought



Monoliths

Microservices

Characteristics of Monoliths

Single unit

Complications with Continuous Delivery

Cascading failures

Characteristics of Microservices

Modular components

Isolated database

Requires careful coordination

Resilience Through Reactive Architectures

Almost every architectural decision includes a trade off

Resilience

The capacity to recover quickly from difficulties.

Difficulties with Microservices

Microservices now require multiple external services to provide the same functionality that was originally guaranteed by a single host

Microservices tend to suffer from external dependency failures due to the underlying network or application

Due to the new requirement for external functionality resource exhaustion may occur

Reactive Manifesto

Reinventing the wheel

Published September 2014

Principles from the Reactive Manifesto

Responsive

Resilient

Elastic

Message Driven

Reactive Manifesto: Responsive

Breaking Down Responsiveness

Responsive systems provide timely responses against customer interaction

Through these timely responses, consistent and reliable upper bounds are established which greatly enhance service quality and correspondingly increase customer expectations

Responsive systems operate with consistency and consequently enhance visibility into anomalies

Responsiveness In Practice

First byte latency as an indication of customer experience

Server and client side metrics

Reactive Manifesto: Resilient

Resilience

A system's ability to stay responsive under failure.

Resilience Components

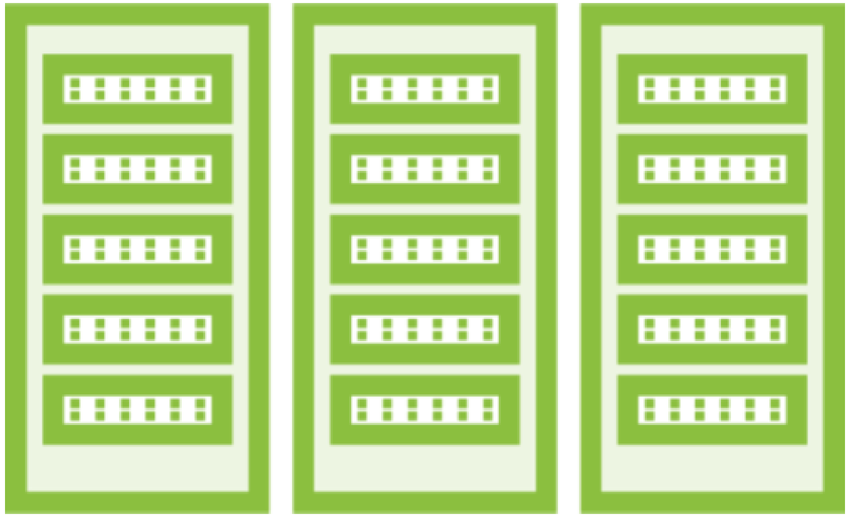
Replication

Containment

Isolation

Delegation

Replication



Distribute workload evenly across multiple servers

Single failure is isolated against other servers

Containment and Isolation

Supported by a decoupled architecture

HTTPS

Implemented via protocols

Delegation

Execution responsibility passes from one component to another

Enables focusing on core competencies

Resilience in Practice

Circuit breaker

Bulkhead

Reactive Manifesto: Elastic

What is Elasticity?

Elasticity is NOT scalability

Elasticity fails under bottlenecks and resource contention

Elasticity is generally achieved by sharding, replication, and workload distribution

Elastic Systems in Practice

Distributed hash table used by BitTorrent

Distributed hash table concepts implemented by AWS S3

Reactive Manifesto: Message Driven

Message Driven and Non-Blocking

Builds upon containment, isolation, and delegation

Optimizes system resources through concurrency

Message Driven in Practice

Accepted status is asynchronous by design

Callback mechanisms eliminate resource wait

Summary

Reactive Manifesto core concepts

Responsive

Resilient

Elastic

Message Driven

Apply concepts to Globomantics' document translation platform