What does @adrianco do?

- Presentations at Conferences
- Presentations at Companies
- Program Committee for Conferences
- Maintain Relationship with Cloud Vendors
- Technology Due Diligence on Deals
- Technical Advice for Portfolio Companies
- Tinkering with Technologies
- Networking with Interesting People

Previously: Netflix, eBay, Sun Microsystems, CCL, TCU London BSc Applied Physics
Agenda

Why now?
Microservice Architectures
What’s Missing?
Migration and Simulation
What’s Next?
Key Goals of the CIO?
Align IT with the business
Develop products faster
Try not to get breached
Insecure applications hidden behind firewalls make you feel safe until the breach happens…

http://peanuts.wikia.com/wiki/Linus'_security_blanket
“Web scale” vs. “Enterprise”
“Webscale”

Freedom and responsibility

High trust
“Enterprise”

Bureaucracy and blame

Low trust
How can everyone get speed, low cost, and better usability?
Mixed methods: Disaggregation into microservices helps!
Example Monolith:

Monolithic application
Complex mix of queries
Monolithic “kitchen sink” database

Because one part of the monolithic application and database holds sensitive data all of it is subject to the most rigorous policies
Microservices version:

- Segregated team owns secure data sources and infrequent updates
- Microservices separation of concerns
- Isolated single purpose connections
- Optimized datastores

User

- Sign Up
- Login
- Home Page

Payment Method

Personal Data

Reports

Because each microservice can conform to the appropriate policy, demands for agility can be separated from requirements for security

Segregated team owns rapid improvement of most common use cases
In Search of Segmentation

Ops

- Datacenters
- AWS Accounts
- AD/LDAP Roles
- IAM Roles
- VLAN Networks
- VPC
- Hypervisor
- Security Groups
- IPtables
- Calico Policy
- Docker Links
- Docker Net/Weave

Dev
Hierarchical Segmentation

AWS Account - Manage across multiple accounts

VPC Z - Manage a small number of large network spaces

Homepage Team Security Group

- A
- B
- C

Reports Team Security Group

- D
- E
- F

An AWS oriented example...
“You build it, you run it.”

Werner Vogels 2006
Developer responsibilities: 
Faster, cheaper, safer
Speeding Up The Platform

Datacenter Snowflakes
• Deploy in months
• Live for years
Speeding Up The Platform

- Datacenter Snowflakes
  - Deploy in months
  - Live for years

- Virtualized and Cloud
  - Deploy in minutes
  - Live for weeks
Speeding Up The Platform

Datacenter Snowflakes
• Deploy in months
• Live for years

Virtualized and Cloud
• Deploy in minutes
• Live for weeks

Container Deployments
• Deploy in seconds
• Live for minutes/hours
Speeding Up The Platform

Datacenter Snowflakes
• Deploy in months
• Live for years

Virtualized and Cloud
• Deploy in minutes
• Live for weeks

Container Deployments
• Deploy in seconds
• Live for minutes/hours

Lambda Deployments
• Deploy in milliseconds
• Live for seconds
AWS Lambda is leading exploration of serverless architectures in 2016
What Happened?

Rate of change increased

Cost and size and risk of change reduced
Microservices
A Microservice Definition

Loosely coupled service oriented architecture with bounded contexts
A Microservice Definition

Loosely coupled service oriented architecture with bounded contexts

If every service has to be updated at the same time it’s not loosely coupled
A Microservice Definition

Loosely coupled service oriented architecture with bounded contexts

If every service has to be updated at the same time it’s not loosely coupled.

If you have to know too much about surrounding services you don’t have a bounded context. See the Domain Driven Design book by Eric Evans.
Next Generation Applications
Fill in the gaps, rapidly evolving ecosystem choices

Lambda
Docker
Spinnaker

Tooling

Archaius
LaunchDarkly
Habitat

Configuration

Etcd
Eureka
Consul

Discovery

Compose
Linkerd
Weave

Routing

Zipkin
Prometheus
Hystrix

Observability

Datastores: Orchestrated, Distributed Ephemeral e.g. Cassandra, or DBaaS e.g. DynamoDB

Operational: Mesos, Kubernetes, Swarm, Nomad for private clouds. ECS, Mesos, GKS for public

Development: components interfaces languages e.g. Docker Hub, Artifactory, Datawire Quark, Go, Rust

Policy: Security compliance e.g. Docker Content Trust. Architecture compliance e.g. Cloud Foundry
What could go wrong?
Timeouts and Retries

Bad config: Every service defaults to 2 second timeout, two retries
Timeouts and Retries

Bad config: Every service defaults to 2 second timeout, two retries

- Edge Service
- Good Service
- Good Service
Timeouts and Retries

Bad config: Every service defaults to 2 second timeout, two retries

If anything breaks, everything upstream stops responding
Timeouts and Retries

Bad config: Every service defaults to 2 second timeout, two retries

If anything breaks, everything upstream stops responding

Retries add unproductive work
Timeouts and Retries

Budgeted timeout, one retry

- Edge Service
- Good Service
- Failed Service
Timeouts and Retries

Budgeted timeout, one retry

Edge Service

Good Service

Failed Service

3s

1s

Fast fail response after 2s

Upstream timeout must always be longer than total downstream timeout * retries delay

No unproductive work while fast failing
Timeouts and Retries

Budgeted timeout, failover retry

For replicated services with multiple instances never retry against a failed instance

No extra retries or unproductive work
Timeouts and Retries

Budgeted timeout, failover retry

For replicated services with multiple instances never retry against a failed instance

No extra retries or unproductive work
Cloud Native Monitoring and Microservices
Interesting architectures have a lot of microservices! Flow visualization is a big challenge.

See http://www.slideshare.net/LappleApple/gilt-from-monolith-ruby-app-to-micro-service-scala-service-architecture
Simulated Microservices

Model and visualize microservices
Simulate interesting architectures
Generate large scale configurations
Eventually stress test real tools

Code: [github.com/adrianco/spigo](github.com/adrianco/spigo)
Simulate Protocol Interactions in Go
Visualize with D3
See for yourself: [http://simianviz.surge.sh](http://simianviz.surge.sh)
Follow @simianviz for updates
Definition of an architecture

See for yourself: http://simianviz.surge.sh/lamp
Running Spigo

$ ./spigo -a lamp -j -d 2
2016/01/26 23:04:05 Loading architecture from json_arch/lamp_arch.json
2016/01/26 23:04:05 lamp.edda: starting
2016/01/26 23:04:05 Architecture: lamp Simple LAMP stack
2016/01/26 23:04:05 architecture: scaling to 100%
2016/01/26 23:04:05 lamp.us-east-1.zoneB.eureka01....eureka.eureka: starting
2016/01/26 23:04:05 lamp.us-east-1.zoneA.eureka00....eureka.eureka: starting
2016/01/26 23:04:05 lamp.us-east-1.zoneC.eureka02....eureka.eureka: starting
2016/01/26 23:04:05 Starting: {rds-mysql store 1 2 []}
2016/01/26 23:04:05 Starting: {memcache store 1 1 []}
2016/01/26 23:04:05 Starting: {webserver monolith 1 18 [memcache rds-mysql]}
2016/01/26 23:04:05 Starting: {webserver-elb elb 1 0 [webserver]}
2016/01/26 23:04:05 Starting: {www denominator 0 0 [webserver-elb]}
2016/01/26 23:04:05 lamp.*.*.www00....www.denominator activity rate 10ms
2016/01/26 23:04:06 chaosmonkey delete: lamp.us-east-1.zoneC.webserver02....webserver.monolith
2016/01/26 23:04:07 asgard: Shutdown
2016/01/26 23:04:07 lamp.us-east-1.zoneB.eureka01....eureka.eureka: closing
2016/01/26 23:04:07 lamp.us-east-1.zoneA.eureka00....eureka.eureka: closing
2016/01/26 23:04:07 lamp.us-east-1.zoneC.eureka02....eureka.eureka: closing
2016/01/26 23:04:07 spigo: complete
2016/01/26 23:04:07 lamp.edda: closing
Open Zipkin

A common format for trace annotations
A Java tool for visualizing traces
Standardization effort to fold in other formats
Driven by Adrian Cole (currently at Pivotal)
Extended to load Spigo generated trace files
Trace for one Spigo Flow
Migrating to Microservices

See for yourself: http://simianviz.surge.sh/migration

Next step
Controls node placement distance
Select models

Endpoint
ELB
MySQL
PHP
MySQL
Migrating to Microservices

See for yourself: http://simianviz.surge.sh/migration

Step 1 - Add Memcache

Step 2 - Add Web Proxy Service
Migrating to Microservices

See for yourself: http://simianviz.surge.sh/migration

Step 3 - Add Data Access Layer

Step 4 - Add Microservices

Data Access

memcache per zone

node.js
Migrating to Microservices

See for yourself: http://simianviz.surge.sh/migration

Step 5 - Add Cassandra

Step 6 - Remove MySQL

12 node cross zone Cassandra cluster
Migrating to Microservices
See for yourself: http://simianviz.surge.sh/migration

Step 7 - Add Second Region
Step 8 - Connect Cassandra Regions

Endpoint with location routed DNS
Migrating to Microservices

See for yourself: http://simianviz.surge.sh/migration

Step 9 - Add Third Region

Endpoint with location routed DNS
Simple Architecture Principles

Symmetry
Invariants
Stable assertions
No special cases
What’s Next?
Serverless
Serverless Architectures

*AWS Lambda getting some early wins*

*Google Cloud Functions, Azure Functions alpha launched*

*IBM OpenWhisk - open sourced*

*Startup activity: iron.io, serverless.com, apex.run toolkit*
Serverless Architecture

API Gateway

DynamoDB  Kinesis  S3
Serverless Architecture

API Gateway

DynamoDB

Kinesis

S3
Serverless Architecture

API Gateway

DynamoDB

Kinesis

S3
Serverless Programming Model

- Event driven functions
- Role based permissions
- Whitelisted API based security
- Good for simple single threaded code
Serverless Cost Efficiencies

100% useful work, no agents, overheads
100% utilization, no charge between requests
No need to size capacity for peak traffic
Anecdotal costs ~1% of conventional system
Ideal for low traffic, Corp IT, spiky workloads
Serverless Work in Progress

Tooling for ease of use
Multi-region HA/DR patterns
Debugging and testing frameworks
Monitoring, end to end tracing
DIY Serverless Operating Challenges

Startup latency
Execution overhead
Charging model
Capacity planning
“We see the world as increasingly more complex and chaotic because we use inadequate concepts to explain it. When we understand something, we no longer see it as chaotic or complex.”

Jamshid Gharajedaghi - 2011
Q&A

Adrian Cockcroft @adrianco
http://slideshare.com/adriancockcroft
Technology Fellow - Battery Ventures

See www.battery.com for a list of portfolio investments
Visit [http://www.battery.com/our-companies/](http://www.battery.com/our-companies/) for a full list of all portfolio companies in which all Battery Funds have invested.