

μ Service Architecture: A Personal Journey of Discovery

Fred George

fedgeorge@acm.org

@fgeorge52 

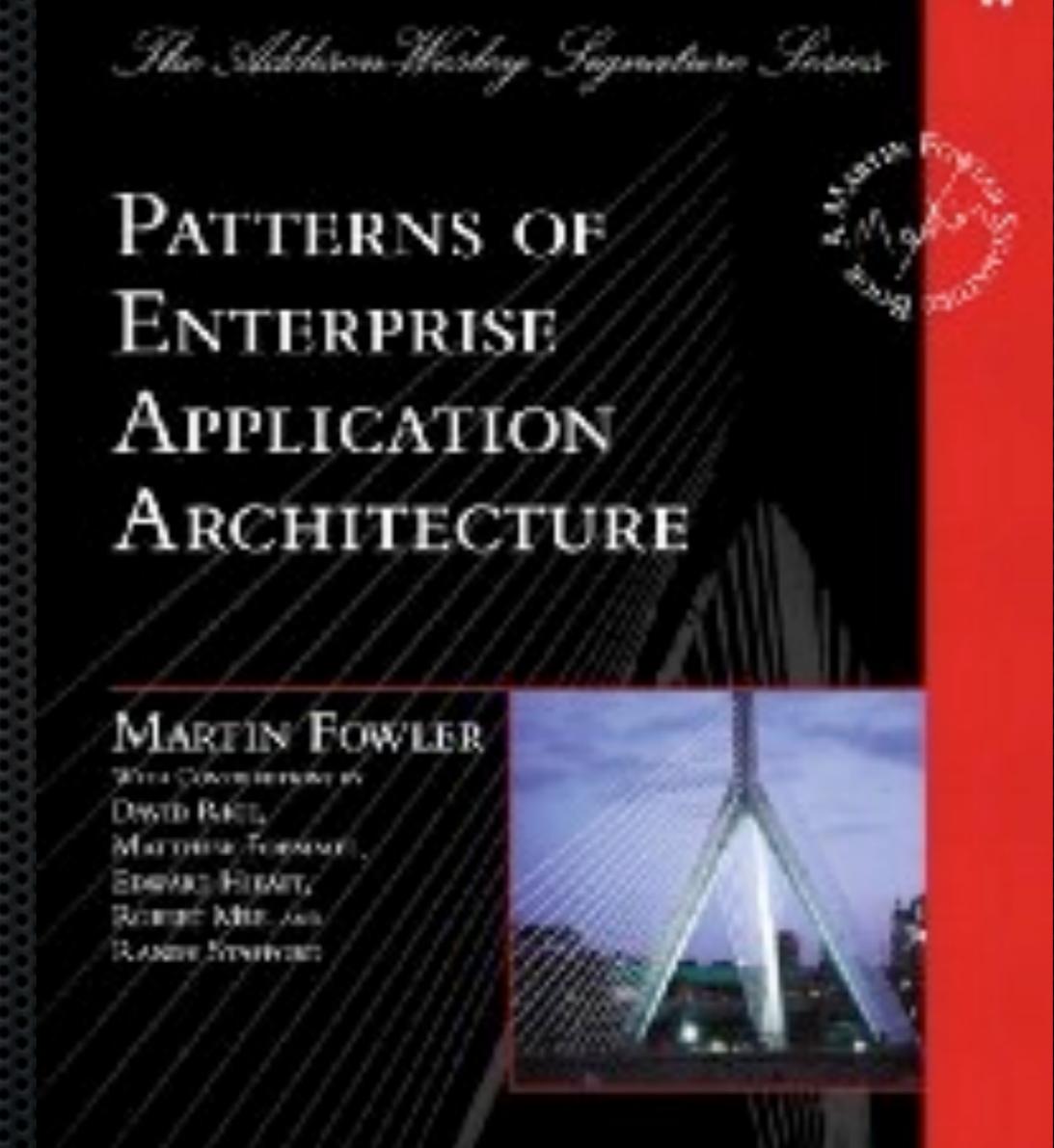
In the Beginning...

2004:
1M loc - Largest Certified
J2EE Application



It Started Well...

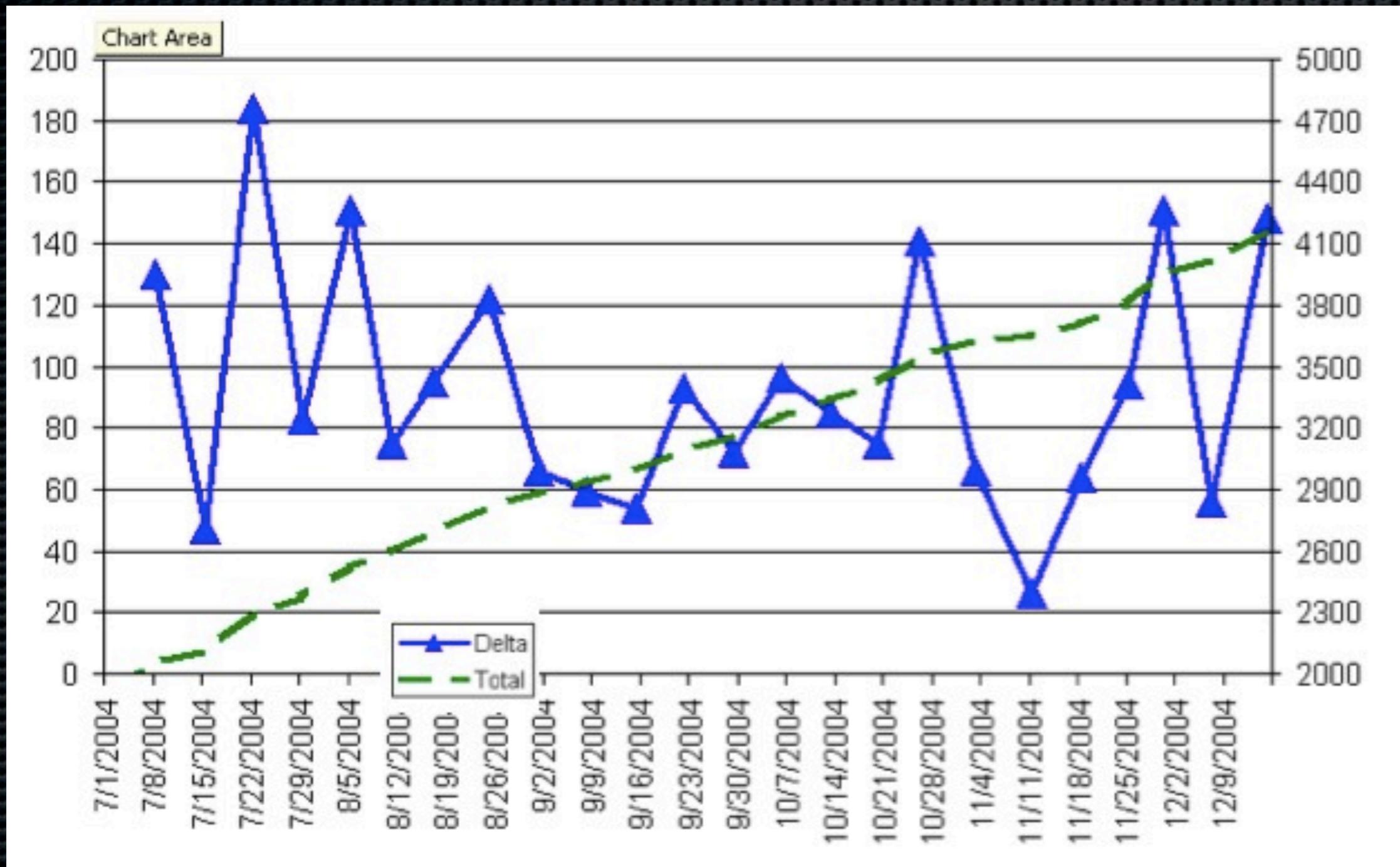
Martin Fowler
consulted, then
joined ThoughtWorks



What I Found in 2004

- ◆ 1M locs
- ◆ 2,000 tests
- ◆ 70% success rate on Acceptance Test was Acceptable
- ◆ Bug database with over 1000 entries
- ◆ Still being changed
- ◆ New tests almost non-existent

Unit Test Counts



```
assertNotNull(new Loan(50));
```

- ♦ Easy (reasonably) unit test
- ♦ On Saturday without interruptions

5 hours later, it passed!

2004:

1M loc - Largest Certified
J2EE Application

“**100K** loc trying to get out”

Jeff Bay

Why? What Happened?

Lazy?

Sloppy

Technical Debt

Inexperienced!

No power to refuse!

Evolution...

2005: SOA in Chinese Bank

Cash,
Balances

Loans

Mortgage

Insurance



Solution: Pub/Sub

- ♦ Events “published”
- ♦ Interested applications “subscribe”
- ♦ Applications that “publish” interaction requests
- ♦ UI elements render appropriate interactions

Rejected

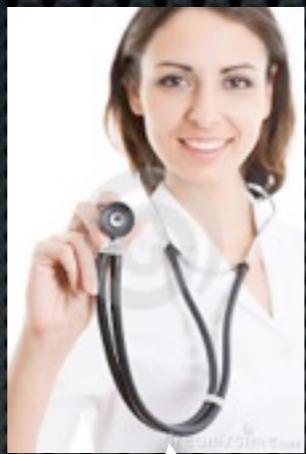
~~2004~~, 2005

**1M loc - Largest Certified
J2EE Application**

**“20 5K loc trying to get
out”**

~~“100K loc trying to get out”~~

2005: Medical Systems



Information “Nuggets”



From:	CAT Scan
About:	Jane Doe
Time:	23 Apr 2012 20:13:45
Urgency:	Concern
Validity:	26 Apr 2012 20:13:45

Rej

Information “Nuggets”

Rejected

Summary

Only relevant
until...

2005: Services for Investment Management

- ♦ Prototyping new system architecture
- ♦ Bayesian Service Principles (letter)
- ♦ It's okay to run more than one version at the same time
- ♦ You can only deploy one service at a time
- ♦ Decoupling result: Deploying 3 times a day!

2006: Batch Processing Replacement Orders

- ♦ Client needed to replace parts on cars
 - ♦ Many variations based on car, location of car
- ♦ Vendor estimated 15-18 months
- ♦ Unacceptable to business

2006: Batch Processing Replacement Orders



Refined Packet Design

?	Addr	?	?	?	?
---	------	---	---	---	---

?	?	?	VIN	?	?
---	---	---	-----	---	---

Date	?	?	?	Name	?
------	---	---	---	------	---

? Addr ? ? ? ?

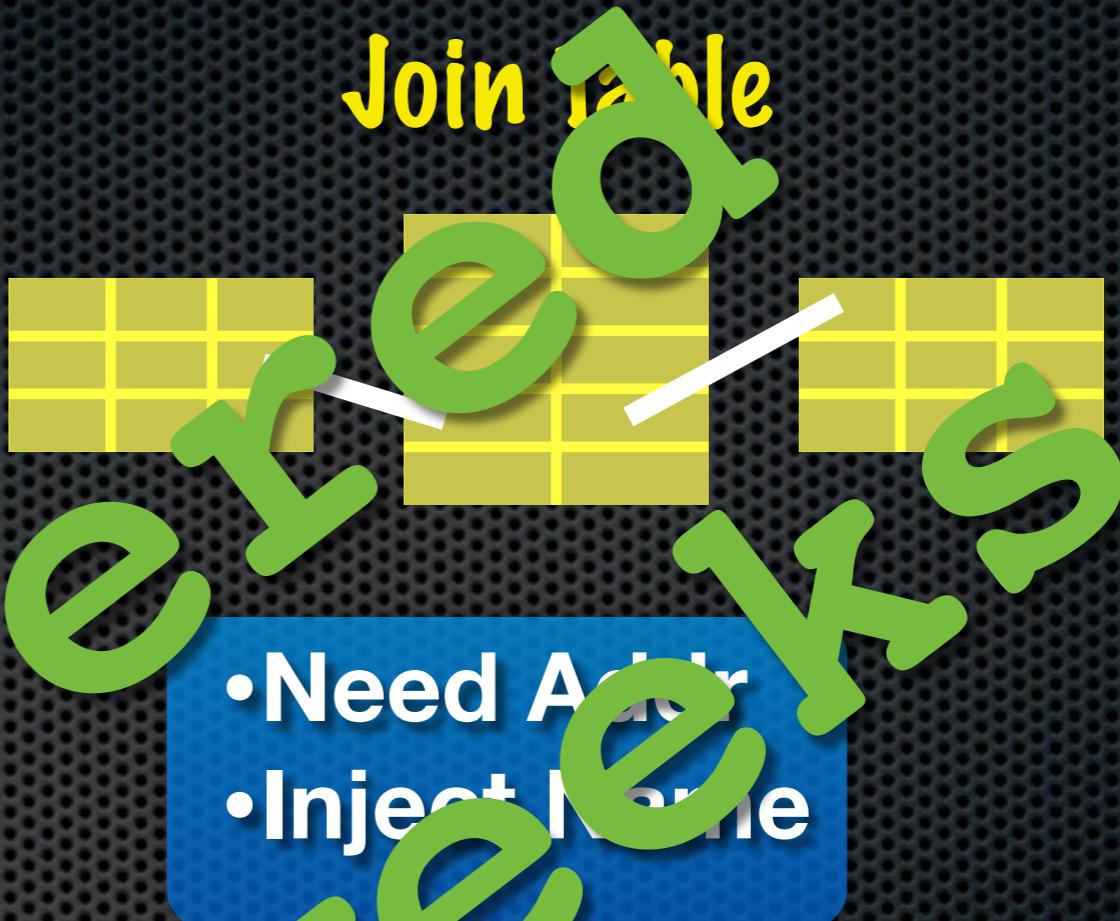


- Need VIN
- Need zzz
- Inject aaa



- Need Name
- Need rrr
- Inject VIN

in



New Observations and Revelations

- ♦ Services are like Classes
 - ♦ Small, crisp conceptualization
 - ♦ Services got very tiny (100 loc)
- ♦ Smalltalk message passing perfect for services
- ♦ Encapsulation
 - ♦ Database segregated among services w/o sharing
 - ♦ Service publishes conclusions, not raw data

New Problems and Challenges

- ♦ “WTF is going on ?”
 - ♦ Cycles, lost packets
 - ♦ Needed to redefine tracking, logging
- ♦ Inexperienced team in defining services
 - ♦ Approach compromised in Release 2
 - ♦ Won an award at TW Tech Day... for being a bad idea!

~~Delayed?~~

~~2004~~ 2005 2006

1M loc - Largest Certified
J2EE Application

200 500 loc
“~~20~~ **5K** loc trying to get
out”

Experiences at...



Forward



PetVillas

2007: Forward Needs to Monitor AdWords Accounts

- ♦ Pub/sub model based on Linda Spaces (Apache Beamline)
 - ♦ Segregate databases to services
 - ♦ Define agent services for each user
 - ♦ Start to automate activities and recommendations
 - ♦ Off-shore for implementation
 - ♦ CTO killed it (former Oracle executive)
 - ♦ Replaced with a more traditional, SQL DB-driven solution
- Rejected**

2008: CardWall

- ♦ Front-end automated Card Wall in Agile
- ♦ Back-end emerged as a set of services analyzing data
 - ♦ One service, one role
 - ♦ Post alerts (with recommendations) to users
- ♦ Migrated to Hadoop Cluster as it grew

New Observations and Revelations

- ♦ Services became disposable
- ♦ Loosely coupled via RESTful Json packets or DB
 - ♦ Albeit still knowledgeable about flow
- ♦ Self-monitoring services replaces Unit Tests
- ♦ Business monitoring replaces Acceptance Tests
- ♦ Services language-agnostic

Current State...



[Gas, electricity & water ▾](#)[Broadband, TV & home phone ▾](#)[Mobiles ▾](#)[Money & insurance ▾](#)[About ▾](#)

Save £1,000+ today with uSwitch!*



Save up to £420 on energy bills*

Compare gas and electricity tariffs with uSwitch and start saving today.

[Switch your gas & electricity ▶](#)[Gas & electricity ▶](#)[Free solar panels ▶](#)[Boiler cover ▶](#)

Broadband deals from £2.50

Compare the best broadband deals and the cheapest offers. Save £200+.

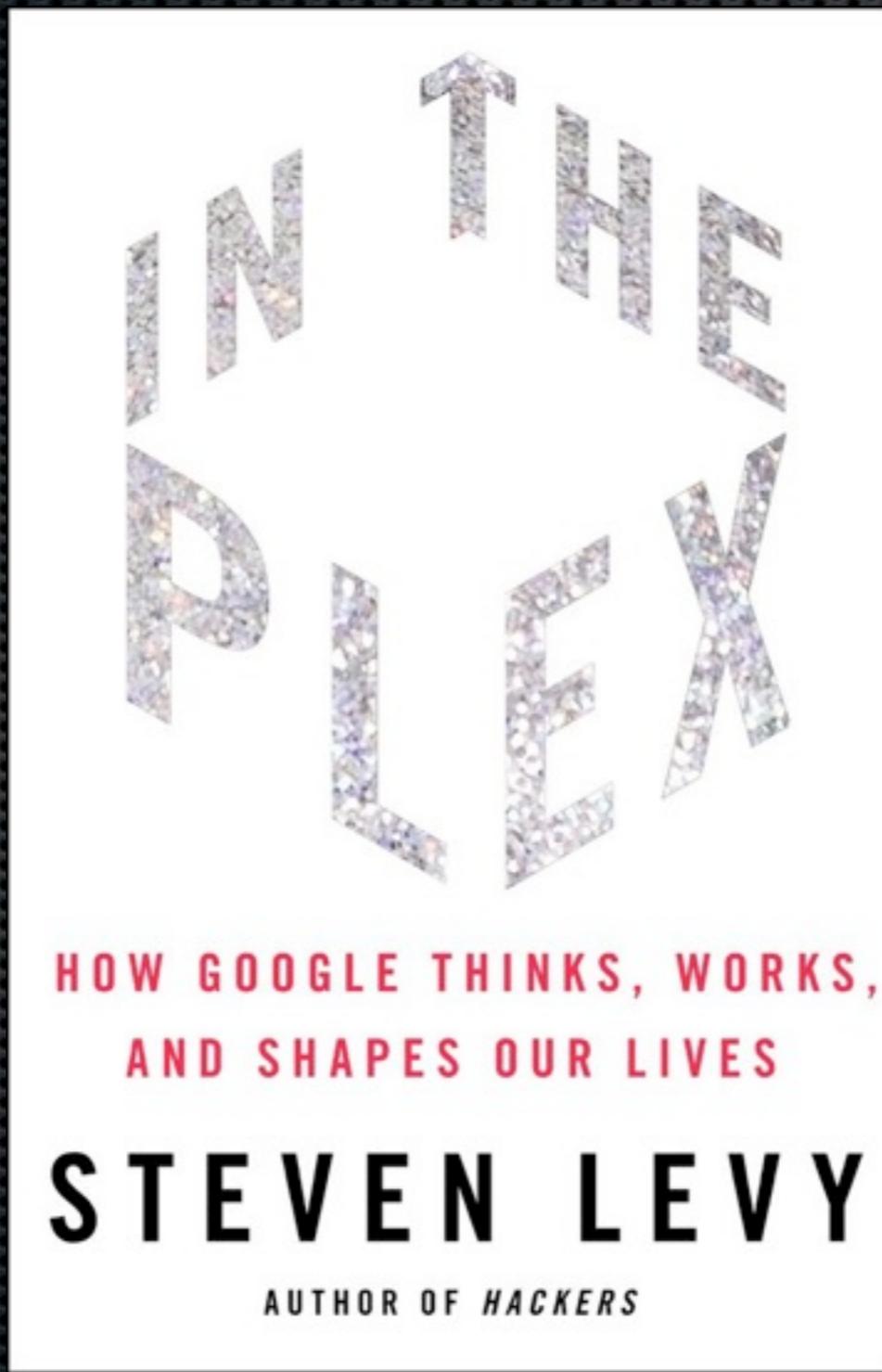
[Compare broadband deals ▶](#)[Broadband deals ▶](#)[Broadband & home phone ▶](#)[Broadband & TV packages ▶](#)

Switch to a 0% Credit Card

Find the best balance transfer, cashback and 0% on purchases cards now.

[Compare credit cards ▶](#)[0% balance transfer cards ▶](#)[0% purchase cards ▶](#)[Credit builder cards ▶](#)

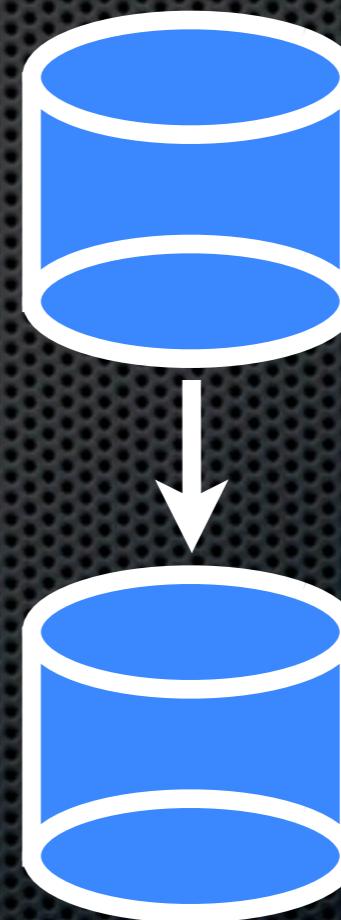
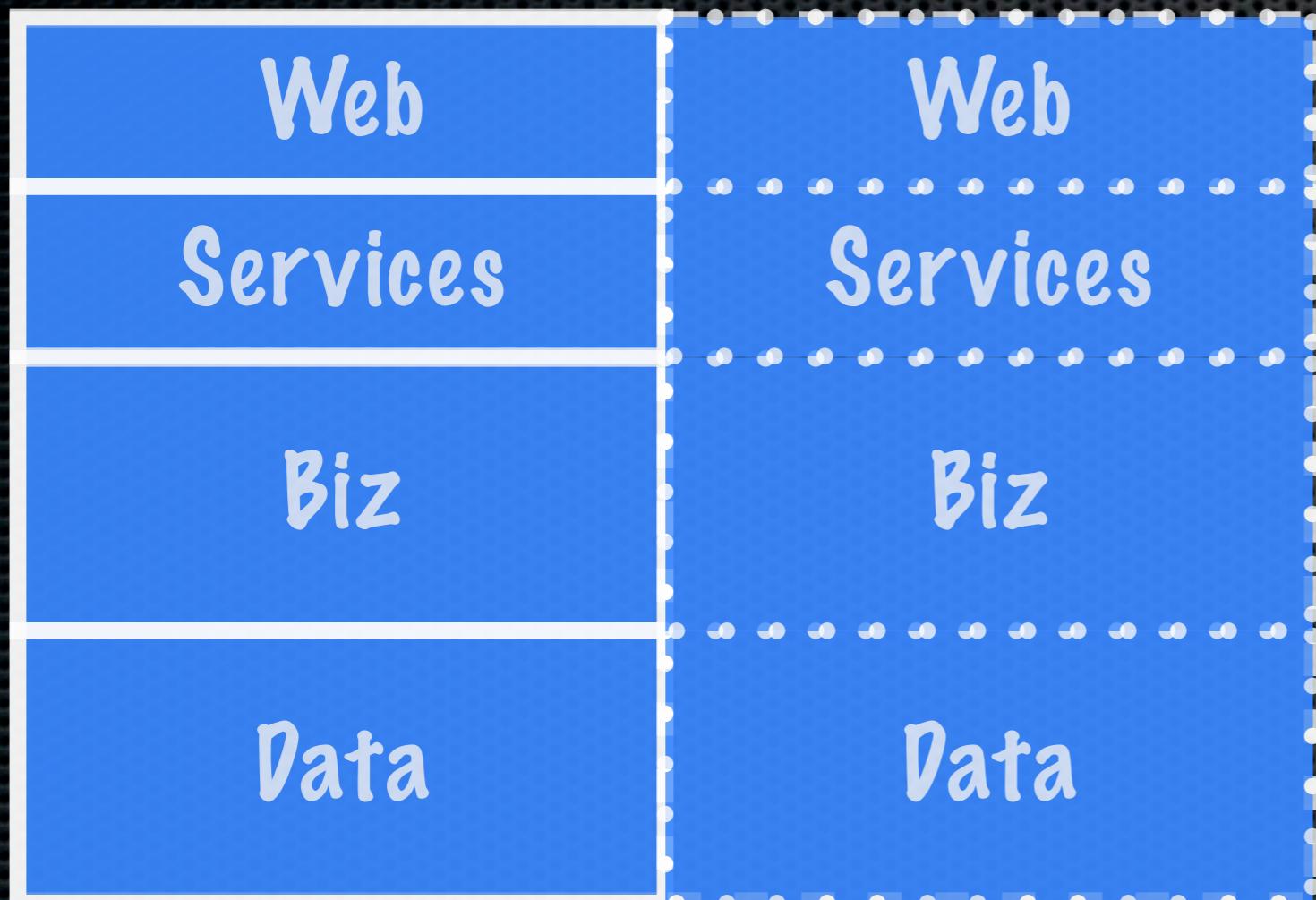
Design: Events, not Entities



Design: History, not Current



Legacy System



ETL/
Data
Muddling

Reporting

Cloud of Signals

Postcode

Name

Server Load

Email Address

Email Read

Postal Address

URL Request

Data Ecosystem

Producers

App

App

Service

Web Server

3rd Party
Web Hooks

Kafka

Postal Address

Email Address

URL Request

Name

Consumers

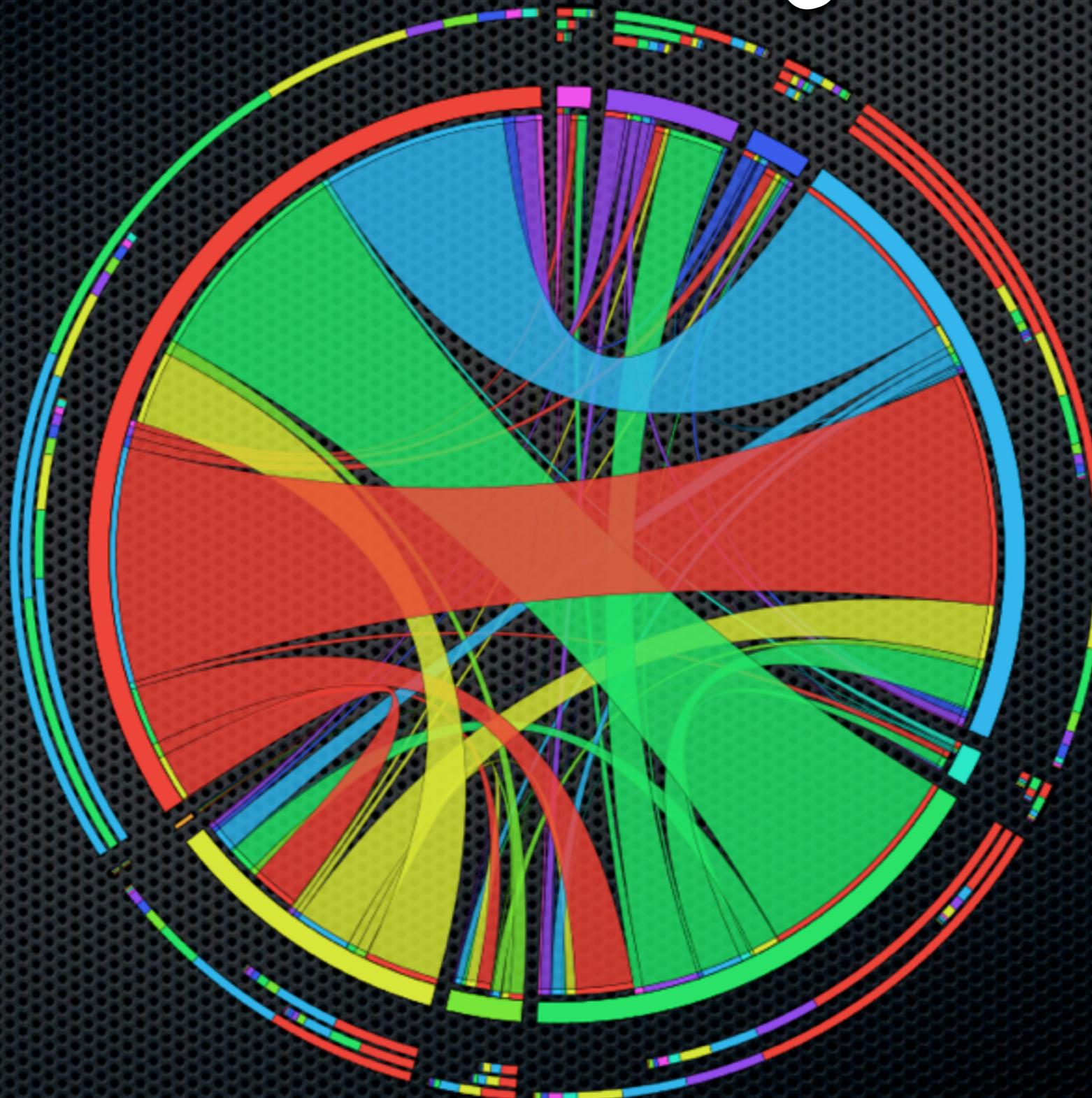
R

Hive

Monitoring

Apps/
Services

Cross-sell Tracking



User Paths Through Site



Agile Best Practices **Not Used**

- ◆ ~~Stand ups~~ Trust with collocation
 - ◆ ~~Story narratives~~
 - ◆ ~~Retrospectives~~
 - ◆ ~~Estimates~~ Results, not blame
 - ◆ ~~Iterations~~
 - ◆ ~~Mandatory pairing~~

Results, not blame

- ◆ ~~Unit tests~~
 - ◆ ~~Acceptance tests~~
 - ◆ ~~Refactoring~~ Small, short-lived apps
 - ◆ ~~Patterns~~
 - ◆ ~~Continuous integration~~

Continuous deployment

What About Technical Debt?

Lazy?

Sloppy

Technical Debt

Inexperienced!

No power to refuse!

Conclusions (so far)...

Summary Principles of μ Services

- ♦ Very, very small
- ♦ Loosely coupled (including flow)
- ♦ Multiple versions acceptable (encouraged?)
- ♦ Self-execution monitoring of each service
- ♦ Publish interesting “stuff” (w/o requirement)
- ♦ “Application” seems to be poor conceptualization

“Living Software” System

- ♦ Long-lived system; short-lived services (human body)
- ♦ Extremely dynamic with continuous deployments
 - ♦ 5-10 minutes between deployments typical
- ♦ Accept it is complex (especially for testing)
 - ♦ Acceptance test on business outcomes instead
- ♦ Radical impact to processes (Anarchy)
- ♦ There will be a learning curve for developers!

μ Service Architecture: A Personal Journey of Discovery

Fred George

fedgeorge@acm.org

@fgeorge52 