OF MONOLITHS
AND MICROSERVICES

Adventures in structuring a web app

Shane Russell,
United States Digital Service
First, a confession...
I’ve never worked at a fancy microservices company.
How *micro* does a service have to be in order to be called a microservice?
But I am a guy who helped architect a pretty complex government web app...
Microservices or a monolith?

You must choose, but choose wisely
Dan  Mountain View, CA
⭐⭐⭐⭐⭐
Microservices worked very well for us! Make sure you have good monitoring and you’ll love it.

David  Chicago, IL
⭐⭐
Microservices are OVERRATED, would not go there again.
All my services are microservices.
all my services are microservices
Decomposition Strategy
“I remain convinced that it is much easier to partition an existing, "brownfield" system than to do so up front with a new, greenfield system.

Sam Newman, *Microservices for Greenfield?*
Microservices

Monolith
Netflix microservices diagram
I think that anybody who’s at all worked with scale and anybody who’s evolved systems to scale knows that architectures are context-specific.

John Allspaw – Etsy
Microservices, monoliths and laser nail guns
OUR STORY
Minimum Viable Product
Feature #2
“Want better apps?
Make smaller things.

Sandi Metz, *All the Little Things*
Small methods 😊
Small methods 😊
Small classes 😊
Small methods 😊
Small classes 😊
Small services?
Small methods 😊
Small classes 😊
Small services? 😊
Feature #4
Feature #4
WHY DID WE RE-MONOLITH?
Reason 1:

It’s easier to change
“Let's change the data store for this feature to MongoDB

Monolith or Microservices
Let's change the data store for this feature to MongoDB

Monolith or Microservices
“This feature could be written best in Go

Monolith or Microservices
This feature could be written best in Go

Monolith or Microservices
We need to change the format of all our application logs

Monolith or Microservices
“We need to change the format of all our application logs

Monolith or Microservices
“Let's record all our errors in Sentry, this great error tracking system we just found out about.

Monolith or Microservices
“Let's record all our errors in Sentry, this great error tracking system we just found out about.”
Microservices make it easier to change any one service without disrupting any others.
Microservices make it easier to change any one service without disrupting any others.
But changes to the platform far more difficult.
And consistency will be almost impossible to maintain.
And consistency will be almost impossible to maintain.
Each time we change our shared library, we need 3 of these...
“Shared code is a coupling between two services that otherwise have nothing to do with each other. The whole point of microservices was to decouple!

Jessica Kerr, *Reuse*
“No shared code.”

Chad Fowler, *Kill “Microservices” Before It’s Too Late*
Hold your shared code to high standards.
Hold your shared code to high standards.

- Focused
- Stable
- Well Documented
Lets refactor functionality between these two features.

Monolith or Microservices
“Let’s refactor functionality between these two features.

Monolith or Microservices
“
Let’s refactor functionality between these two features.

Monolith or Microservices

By a long shot
Refactoring functionality between...

Small methods
Small classes
Small services?
Refactoring functionality between...

Small methods 😊
Small classes
Small services?
Refactoring functionality between...

Small methods 😊

Small classes 😊

Small services?
Refactoring functionality between...

Small methods 😊
Small classes 😊
Small services? 😲
Any refactoring of functionality between services is much harder than it is in a monolith.

Martin Fowler, Monolith First
def establish_claim_in_vbms(end_product):
    - Appeal.repository.establish_claim!
    + VBMSService.establish_claim!
        claim_hash: end_product.to_vbms_hash,
        veteran_hash: appeal.veteran.to_vbms_hash
-class Api::V1::AppealsController < Api::V1::ApplicationController
+class Api::V2::AppealsController < Api::V2::ApplicationController

  before_action :verify_feature_enabled

  rescue_from Caseflow::Error::InvalidSSN, with: :invalid_ssn
Reason 1:
It’s easier to change

Reason 2:
Less operational complexity
Distributed systems are hard.
“Every time you extract a collaboration between objects to a collaboration between systems, you’re accepting a world of hurt with a myriad of liabilities and failure states.

DHH, *The Majestic Monolith*
<table>
<thead>
<tr>
<th><strong>Dependency</strong></th>
<th><strong>Scale</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit breakers, fallbacks,</td>
<td>Auto-scaling</td>
</tr>
<tr>
<td>chaos</td>
<td>Redundancy – avoid SPoF</td>
</tr>
<tr>
<td>Simple clients</td>
<td>Partitioned workloads</td>
</tr>
<tr>
<td>Eventual consistency</td>
<td>Failure-driven design</td>
</tr>
<tr>
<td>Multi-region failover</td>
<td>Chaos under load</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Variance</strong></th>
<th><strong>Change</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineered operations</td>
<td>Automated delivery</td>
</tr>
<tr>
<td>Understood cost of variance</td>
<td>Integrated practices</td>
</tr>
<tr>
<td>Prioritized support by impact</td>
<td></td>
</tr>
</tbody>
</table>
Continuous Integration
Open Pull Request

- Isolation Tests,
  Linters,
  Security Scans

- Integration Tests

- Shared between repos
Open Pull Request

Isolation Tests, Linters, Security Scans

Integration Tests

Integration Test PR
Local Development
Deployment configuration

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>caseflow-certification</td>
<td>Added VVA environments and certs to bake scripts (#642)</td>
</tr>
<tr>
<td>caseflow-decisions</td>
<td>Added Static Proxy for all apps (#509)</td>
</tr>
<tr>
<td>caseflow-feedback</td>
<td>Added VVA environments and certs to bake scripts (#642)</td>
</tr>
<tr>
<td>caseflow-monitor</td>
<td>Added VVA environments and certs to bake scripts (#642)</td>
</tr>
<tr>
<td>creds/tasks</td>
<td>Consolidate all appeals team devops-related repositories</td>
</tr>
<tr>
<td>dsva-user-config</td>
<td>Added Shade to devops (#593)</td>
</tr>
<tr>
<td>dsva-user/meta</td>
<td>deploy_env-specific SSH access (#299)</td>
</tr>
<tr>
<td>efolder-express</td>
<td>Added VVA prod vault info (#642)</td>
</tr>
<tr>
<td>gemstash</td>
<td>Gem Server (#198)</td>
</tr>
<tr>
<td>jenkins</td>
<td>Consolidate all appeals team devops-related repositories</td>
</tr>
<tr>
<td>nessus</td>
<td>Consolidate all appeals team devops-related repositories</td>
</tr>
<tr>
<td>pdftk</td>
<td>Consolidate all appeals team devops-related repositories</td>
</tr>
<tr>
<td>rpm/tasks</td>
<td>clarify task name, it's generic (#458)</td>
</tr>
</tbody>
</table>
## Deployment configuration

<table>
<thead>
<tr>
<th>Branch: master</th>
<th>appeals-deployment / ansible / roles /</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>askldjd committed on GitHub</th>
<th>Added VVA prod vault info (#642)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>caseflow-certification</th>
<th>Added VVA environments and certs to bake scripts (#642)</th>
</tr>
</thead>
<tbody>
<tr>
<td>caseflow-decisions</td>
<td>Added Static Proxy for all apps (#509)</td>
</tr>
<tr>
<td>caseflow-feedback</td>
<td>Added VVA environments and certs to bake scripts (#642)</td>
</tr>
<tr>
<td>caseflow-monitor</td>
<td>Added VVA environments and certs to bake scripts (#642)</td>
</tr>
<tr>
<td>creds/tasks</td>
<td>Consolidate all appeals team devops-related repository</td>
</tr>
<tr>
<td>dsva-user-config</td>
<td>Added Shade to devops (#593)</td>
</tr>
<tr>
<td>dsva-user/meta</td>
<td>deploy_env-specific SSH access (#299)</td>
</tr>
<tr>
<td>efolder-express</td>
<td>Added VVA prod vault info (#642)</td>
</tr>
<tr>
<td>gemstash</td>
<td>Gem Server (#198)</td>
</tr>
<tr>
<td>jenkins</td>
<td>Consolidate all appeals team devops-related repository</td>
</tr>
<tr>
<td>nessus</td>
<td>Consolidate all appeals team devops-related repository</td>
</tr>
<tr>
<td>pdftk</td>
<td>Consolidate all appeals team devops-related repository</td>
</tr>
<tr>
<td>rpm/tasks</td>
<td>clarify task name, it's generic (#458)</td>
</tr>
</tbody>
</table>
Additional Overhead

- DNS
- Certificates
- IP configuration
- Monitoring setup
Additional Overhead

- DNS
- Certificates
- IP configuration
- Monitoring setup

X (Number of Environments)
Additional Overhead

- DNS
- Certificates
- IP configuration
- Monitoring setup

X (Number of Environments)
Reason 1: It’s easier to change

Reason 2: Less operational complexity

Reason 3: For our team, the payoff wasn’t big enough
- Increased fault tolerance
- Independent deployments and rollbacks
- Forced modularity/isolation
- Faster builds
- Allows differing technology stacks
Microservices enable development teams to get big and work like they're still small.
Conway’s Law

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.
“Microservices is as much about a team organizational attitude, as it is about a technological attitude.

Martin Fowler, *Microservices*
**Dependency**
- Circuit breakers, fallbacks, chaos
- Simple clients
- Eventual consistency
- Multi-region failover

**Scale**
- Auto-scaling
- Redundancy – avoid SPoF
- Partitioned workloads
- Failure-driven design
- Chaos under load

**Variance**
- Engineered operations
- Understood cost of variance
- Prioritized support by impact

**Change**
- Automated delivery
- Integrated practices
WHERE ARE WE NOW?
Biggest pain:

The build is longer... and sometimes flakey
+10 minutes per year
Another pain:

One missing index or memory leak can bring down the whole app.
Unexpected win:

Keeping stable code separate.
These were relatively stable.
These were relatively stable.
When you monolith,
make it a modular monolith.
RESOURCES

- Reuse
  http://blog.jessitron.com/2017/02/reuse.html

- Monolith First
  https://martinfowler.com/bliki/MonolithFirst.html

- The Majestic Monolith
  https://m.signalvnoise.com/the-majestic-monolith-29166d022228

- Microservices Prerequisites
  https://martinfowler.com/bliki/MicroservicePrerequisites.html

- Microservices for Greenfield?
  http://samnewman.io/blog/2015/04/07/microservices-for-greenfield/

- Microservices – Please don’t
  http://basho.com/posts/technical/microservices-please-dont/
RESOURCES

- Mastering Chaos – A Netflix Guide to Microservices
  https://www.youtube.com/watch?v=CZ3wIuvmHeM

- What’s Your Decomposition Strategy?
  https://builttoadapt.io/whats-your-decomposition-strategy-e19b8e72ac8f

- The New New Software Development Game
  http://www.ustream.tv/recorded/61477219

- Microservices, monoliths and laser nail guns: Etsy tech boss on finding the right focus

- Microservices
  https://www.youtube.com/watch?v=2yko4TbC8cI

- Kill "Microservices" before its too late
  https://www.youtube.com/watch?v=-UKEPd2ipEk