Monolith to Microservices

How HomeDepot.com made the switch

Christopher Grant
Sr Architect HomeDepot.com
Building the monolith
Its an old story

• Start out small
• Lean and ready to take on the world
• Add some new features
• One day suddenly you realize

...you’ve got a monolith
2011

- 2 GB Java Deploy that took an hour
- Locking SCM (Clear Case) with no CI
- 3 Month long release Cycles
- 2-4 hour deploy outages
- Infrastructure drift (Pets) and config sprawl
Breaking Down the Monolith
Our Objective

- Increase the rate of change
- Increase number developers
- Maintain and enhance operations
Agile
Domain Spaces
Continuous Integration Tooling
APIs
Feature Throttles
Key Take Aways
Start with a vision

- Have a vision
- Understand your objectives
- Choose what matters
- Implement in stages
Culture change can be hard

Schneider Culture Model

http://agilitrix.com/2011/03/how-to-make-your-culture-work/
Iterative Architecture?
Changing the wheels while driving?
Evolutionary Architecture
Architect for Change

- Prepare for future changes
- Be proactive not reactive
- Delay until the last responsible moment
- Expect but minimize future rework
Domain Driven Design
Decompose & Define Boundaries

- Understand business products & domains
- Identify people structures
- Review data models
- Look for the hard things
8 fallacies of distributed computing

- The network is reliable
- Latency is zero
- Bandwidth is infinite.
- The network is secure
- Topology doesn't change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

L Peter Deutsch and others at Sun Microsystems
Implement Safely

• Automate early and often
• Consider in-place and green-field
• Utilize feature switches and traffic throttles
• Encourage separation and independence
## Result

<table>
<thead>
<tr>
<th></th>
<th>2011</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apps</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>Sales</td>
<td>$1b</td>
<td>$5b</td>
</tr>
<tr>
<td>Developers</td>
<td>170</td>
<td>300</td>
</tr>
<tr>
<td>Deploy frequency</td>
<td>3 month</td>
<td>Weekly</td>
</tr>
</tbody>
</table>
2015 began moving to cloud & micro services
Microservices in the Cloud
Microservices for everyone

- Everyone’s trying to break down their monolith these days
  - More maintainable
  - Easier deployment
  - More isolated concerns
  - Not just for unicorns anymore
How is THD approaching it
Microservices Patterns

Chris Richardson
http://microservices.io/

Great resource &
Great way to talk about what we did
Core Patterns

• Monolith
• Microservice
Core Patterns

- Monolith
- Microservice
Microservices

• How big should they be?
• How many should we have?
conway’s law

new system:

organization:
Two Pizza Rule

If a team can’t be fed by two pizzas then its too big

-Jeff Bezzos
1 Microservice per two pizza team
Domain Services
Cloud Native Apps

http://12factor.net/

**The Twelve-Factor App**

- Code Base
- Dependencies
- Config
- Backing Services
- Build Release Run
- Process
- Port Binding
- Concurrency
- Disposability
- Dev/Prod Parity
- Logs
- Admin Process
Communication

- API Gateways
- Discovery
  - Server side
  - Client Side
    - *Ambassador (proxy)*
- Registration
  - Self
  - Third Party
Communication

- API Gateways
- Discovery
  - Server side
  - Client Side
  - Ambassador (proxy)
- Registration
  - Self
  - Third Party
Ambassador Pattern

1. The new server registers itself in the registry
2. Registry updates all LB configurations
3. The client App calls a local proxy
4. The Local proxy routes to the new Service App
Deployment patterns

• Single Service Per Host
• Multiple Service Per Host
• Single Service Per VM
• Single Service per container
Deployment patterns

- Single Service Per Host
- Multiple Services Per Host
- **Single Service Per VM**
- Single Service per container
Immutable Architecture

- Assets are rebuilt, redeployed but never modified
- SSH is only for diagnostic uses
- Base images used for speed and maintainability
- Immutable application assets added separately
Single Instance

productpage-1-5-5.jar
Java 8
Proxy
Registry Agent  Registry Template  Logs Collector

App

product-app.yaml
sourceImage: java8-10141019
properties:
  app_name: productpage
  app_version: 1-5-5

Base
(java8-10141019)
Building with Hammer & Nail
Data Management

- One Database per service
- Event-driven Architecture
Data Management

- One Database per service
- Event-driven Architecture
Decoupling Strategies

Shared Database

App1 → DB1 → App2

Shared Domain

App1 → Dom1 → App2

Decentralized Data

App1 → DB1 → App2

Queue
7 deadly sins of microservices

Daniel Bryant
@danielbryantuk

https://opencredo.com/the-seven-deadly-sins-of-microservices-redux/
Thank You