Building simple Node.js Microservices using Hapi and Redis

Jeff Barczewski
Building simple Node.js Microservices using Hapi and Redis

- codewinds.com/nc2015
  - slides, code, resources, notes
- @jeffbski and @codewinds
- jeff@codewinds.com
Who is Jeff Barczewski?

- CodeWinds Online Training
- 26 yrs Software Engineer
- USAF, RGA, Elsevier, MasterCard
- @jeffbski jeff@codewinds.com
Story of a monolith

- Large highly coupled codebase
- Stuck on old technology
- Minimal tests
- Limited documentation
- Siloed knowledge
Story of a monolith - p2

- Slow feature cycle time
- Custom forks for clients
- Hard to scale
- Defects and regressions
Learning from History

• Do One Thing and Do It Well - Unix
  ▪ Piping data between programs
• SOA w/ESB, Soap - complexity
• REST - simplicity

Credits: Avatar Surfing On Soap Bar by schoschie - CC BY-SA 2.0
Requirements Change

- Real-time web/apps - push
- Web for transactions, RPC style
  - large number of small resources
  - dynamic tuning of data by device
  - FalcorJS w/JSON Graph
  - Relay w/GraphQL
Microservices

- Aka: fine grained SOA, Replaceable Component Architecture
- Term emerged for this style architecture 2011-2012
- Many opinions
Microservices (MSA)

Architectural style that generally utilizes small focused processes communicating over lightweight means.
Microservice Benefits

- Independently deployable, replaceable, upgradeable
- Elastic and scalable
- Allows disparate programming languages and technology (db)
- Managed by different teams
Microservice Benefits - p2

- Can enable resiliency
- Composable - caching, aggregation, proxies, gateways
- Allows fine grained monitoring and resource tuning
Microservice Risks

- Decomposing w/o domain knowledge
- Siloed knowledge / skills
- Rapid change vs quality
- Deploy / log complexity
- Version compatibility
Does it work?
Monolith to Microservices

- Netflix - 37% web bandwidth peak, 1B API calls/day, 6 services/call
- Amazon - 100-150 service calls per web page
- Ebay - split over 75 business functional areas
Microservice Optimism

Microservices are becoming the default style for building enterprise applications.
Balance Risk vs Reward

- Evolve architecture
- Automate deployment, log management, and monitoring
- Resilient design
- Plan for scalability (human and hardware)
Evolving Architecture

- Decomposition - Functions, Modules
- Redis Queue - Mail service
- Redis Reliable Queue
- Redis Cache - Ad generator
Evolving Architecture - p2

- Hapi API Gateway - Proxy
- Hapi Microservice
- Hapi API Gateway - Enhanced
- Hapi Microservice using plugin
Going Beyond

- Frameworks - SenecaJS
- Messaging - Kafka
- Stream services - AWS Lambda
- Dynamic scaling / deploy - Kubernetes
Resources

- codewinds.com/nc2015
  - slides, code, resources, notes
- @jeffbskki and @codewinds
- jeff@codewinds.com