Foundations of Software Engineering

Architecture – Styles and Hypes

Michael Hilton
Administrivia

• Overall reflections were well done
  – Review slides for suggestions for improvement
• Teams seems to be working well
  – Please reach out if you experience issues
• Resume advice if you want
Learning Goals

• Recognize architectural styles and their implications
• Reason about system structures and their tradeoffs with architectural views and styles
• Reason about tradeoffs of Microservice architectures
• Understand the key ideas of DevOps
• Appreciate challenge of architecture in practice
Microservices
Microservices

Microservices Everywhere
WHAT ARE MICROSERVICES?

SERVICES FOR ANTS??!!
MICROSERVICES...

...SO HOT RIGHT NOW

memegenerator.net
Netflix Discussion
Microservices

Twitter
COMCAST

NETFLIX

eBay

Amazon

UBER

GROUPON
A monolithic application puts all its functionality into a single process...

... and scales by replicating the monolith on multiple servers

A microservices architecture puts each element of functionality into a separate service...

... and scales by distributing these services across servers, replicating as needed.

source: http://martinfowler.com/articles/microservices.html
Microservices

• Building applications as suite of small and easy to replace services
  – fine grained, one functionality per service
    (sometimes 3-5 classes)
  – composable
  – easy to develop, test, and understand
  – fast (re)start, fault isolation
  – Modelled around business domain
• Interplay of different systems and languages, no commitment to technology stack
• Easily deployable and replicable
• Embrace automation, embrace faults
• Highly observable
Exactly. We will bid it out this year. A “route smart” system will not only create micro-districts for drivers to be accountable, it will also determine salt domes and fuel stations to keep them in the road. Today, we operate on clipboards & paper & expect them to succeed.

Daniel Winne @thedigitalpit
Replying to @billpeduto

I wonder if @Uber or @googlemaps could give the city an algorithm for where the trucks should drive most efficiently.
Technical Considerations

• HTTP/REST/JSON communication
• Independent development and deployment
• Self-contained services (e.g., each with own database)
  – multiple instances behind load-balancer
• Streamline deployment
monolith - single database

microservices - application databases

source: http://martinfowler.com/articles/microservices.html
Microservices overhead

for less-complex systems, the extra baggage required to manage microservices reduces productivity

as complexity kicks in, productivity starts falling rapidly

the decreased coupling of microservices reduces the attenuation of productivity

but remember the skill of the team will outweigh any monolith/microservice choice
Drawbacks

• Complexities of distributed systems
  – network latency, faults, inconsistencies
  – testing challenges
• Resource overhead, RPCs
• Shifting complexities to the network
• Operational complexity
• Frequently adopted by breaking down monolithic application
• HTTP/REST/JSON communication
Discussion of Microservices

• Are they really “new”? 
• Do microservices solve problems, or push them down the line?
• What are the impacts of the added flexibility?
• Beware “cargo cult”
• “If you can’t build a well-structured monolith, what makes you think microservices is the answer?” – Simon Brown
• Leads to more API design decisions
Interview