

Architecture – Styles and Hypes

Michael Hilton Claire Le Goues

October 10, 2019

Administrivia

- Homework 4 has been released. -feedback?
- Midterm, Oct 17th

Learning Goals

- Understand history of Microservices
- Reason about tradeoffs of Microservices architectures.
- Practice Building Design Docs

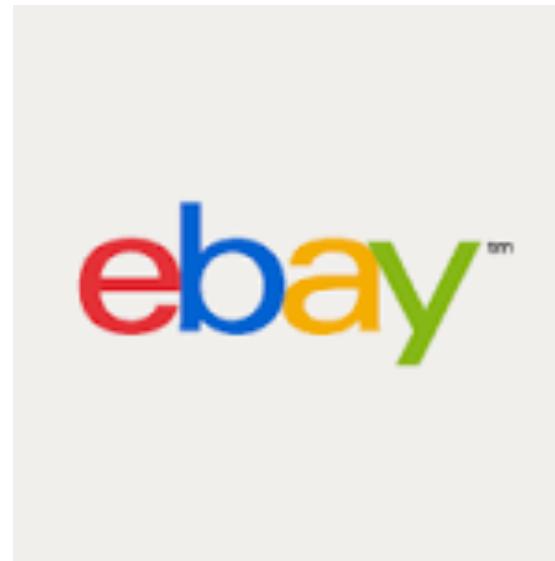
MICROSERVICES







Microservices

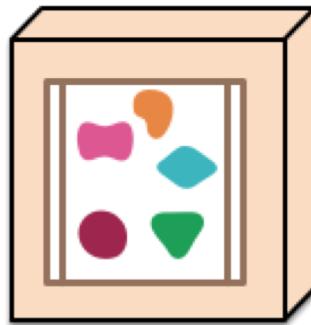
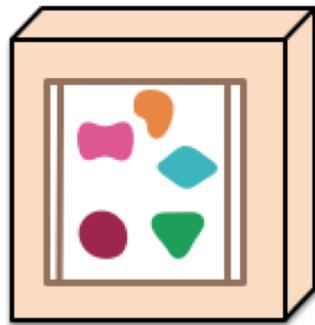


U B E R G R O U P O N®

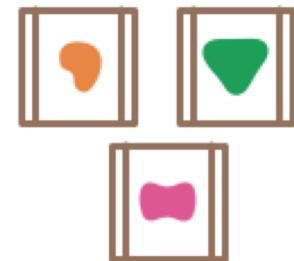
A monolithic application puts all its functionality into a single process...



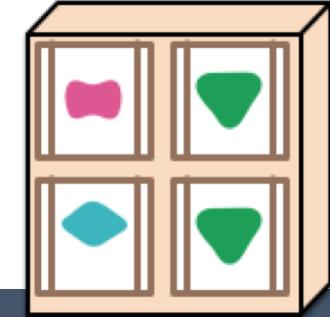
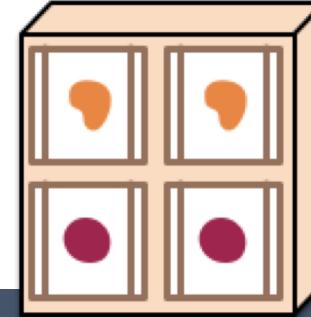
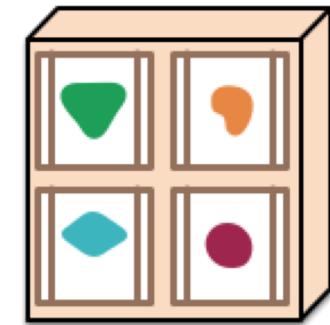
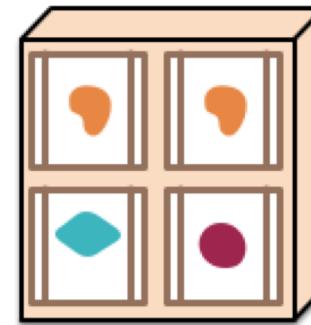
... and scales by replicating the monolith on multiple servers



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



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

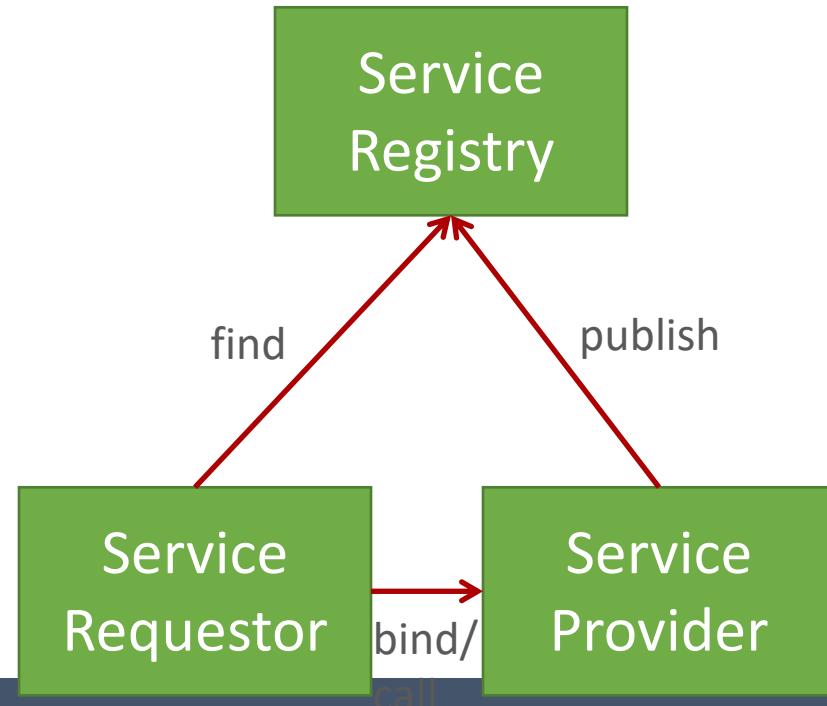


Microservices

- 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

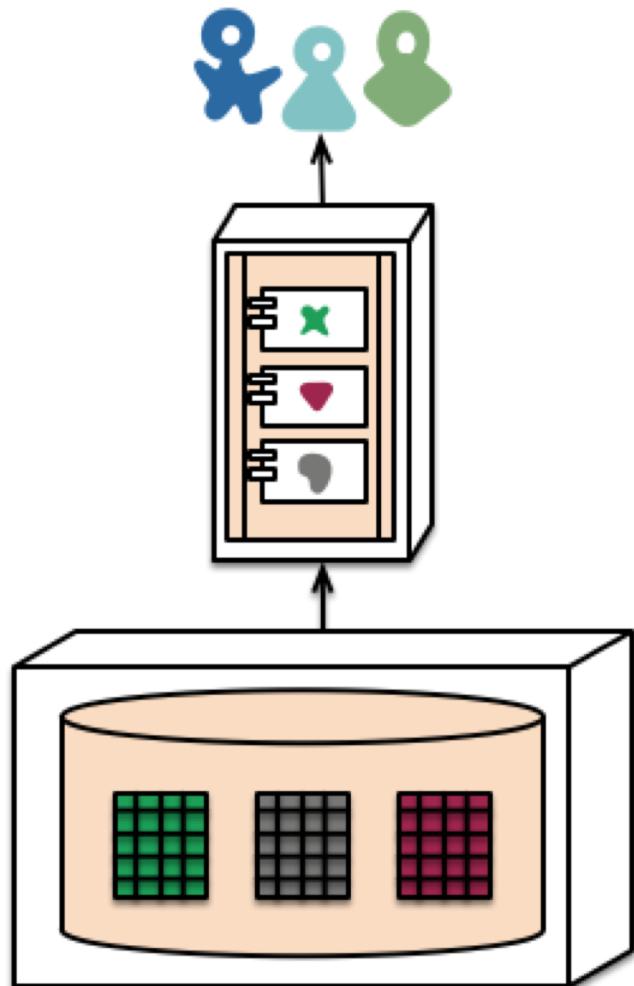
Service Oriented Architectures (SOA)

- Service: self-contained functionality
- Remote invocation, language-independent interface
- Dynamic lookup possible
- Often used to wrap legacy systems

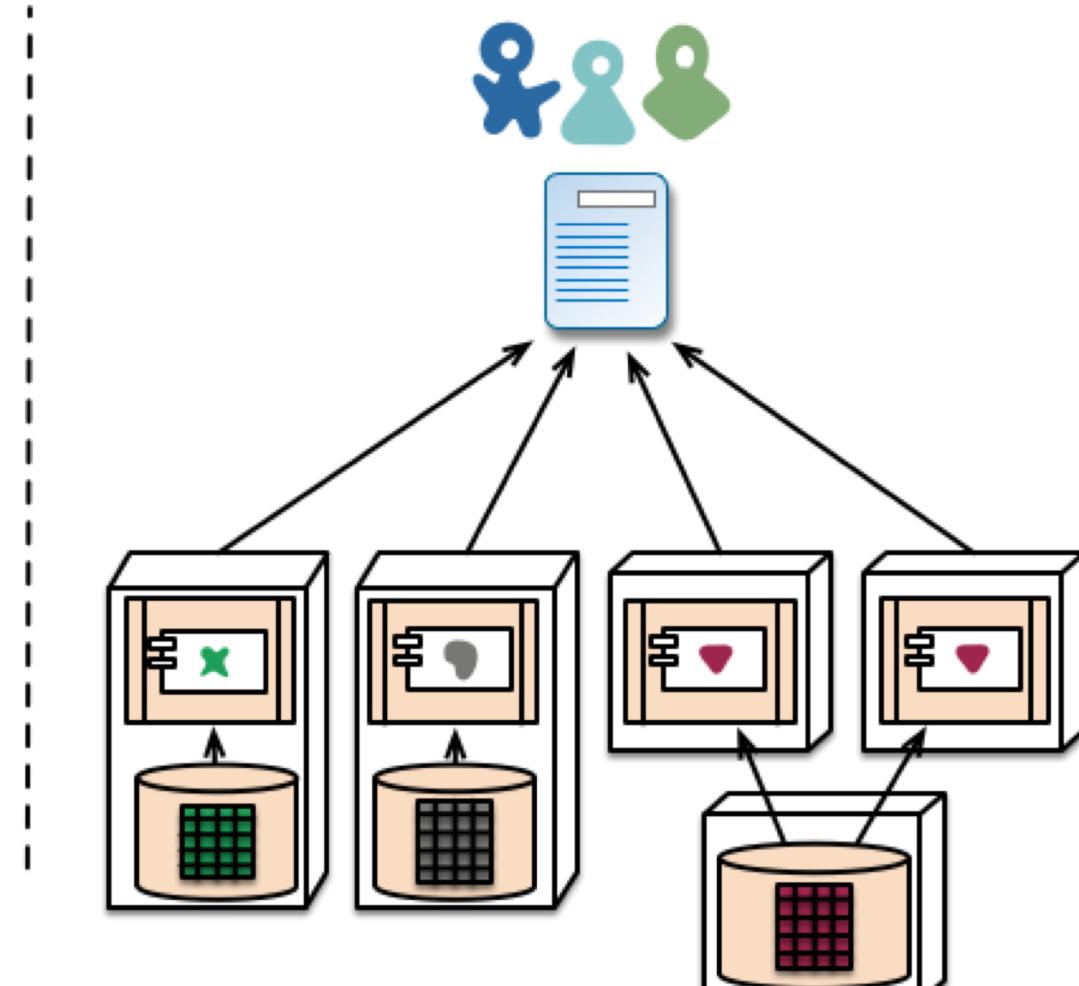


Technical Considerations

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



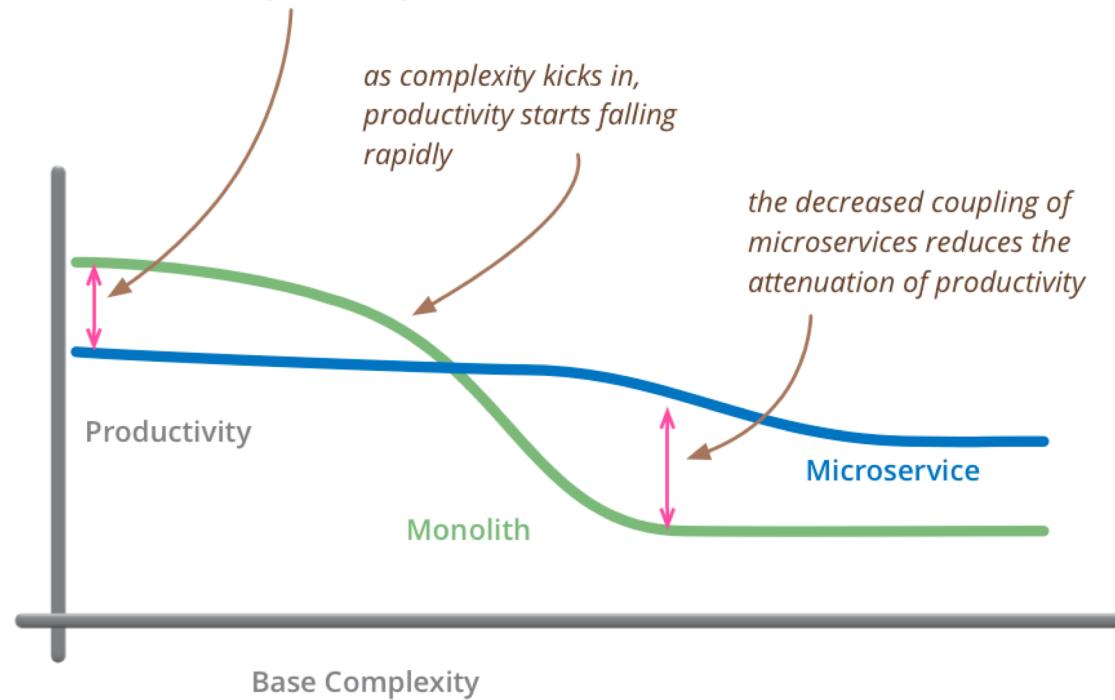
monolith - single database



microservices - application databases

Microservices overhead

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



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

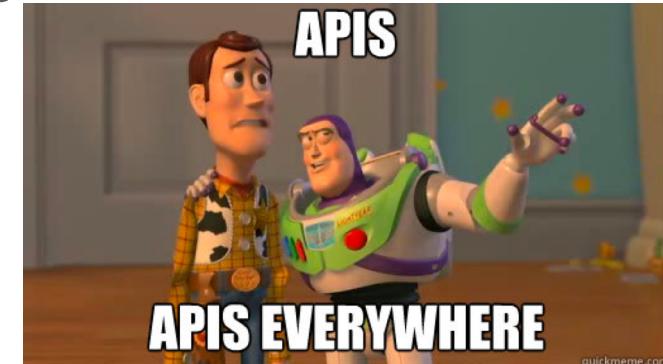
Drawbacks

- 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

Apache Thrift™

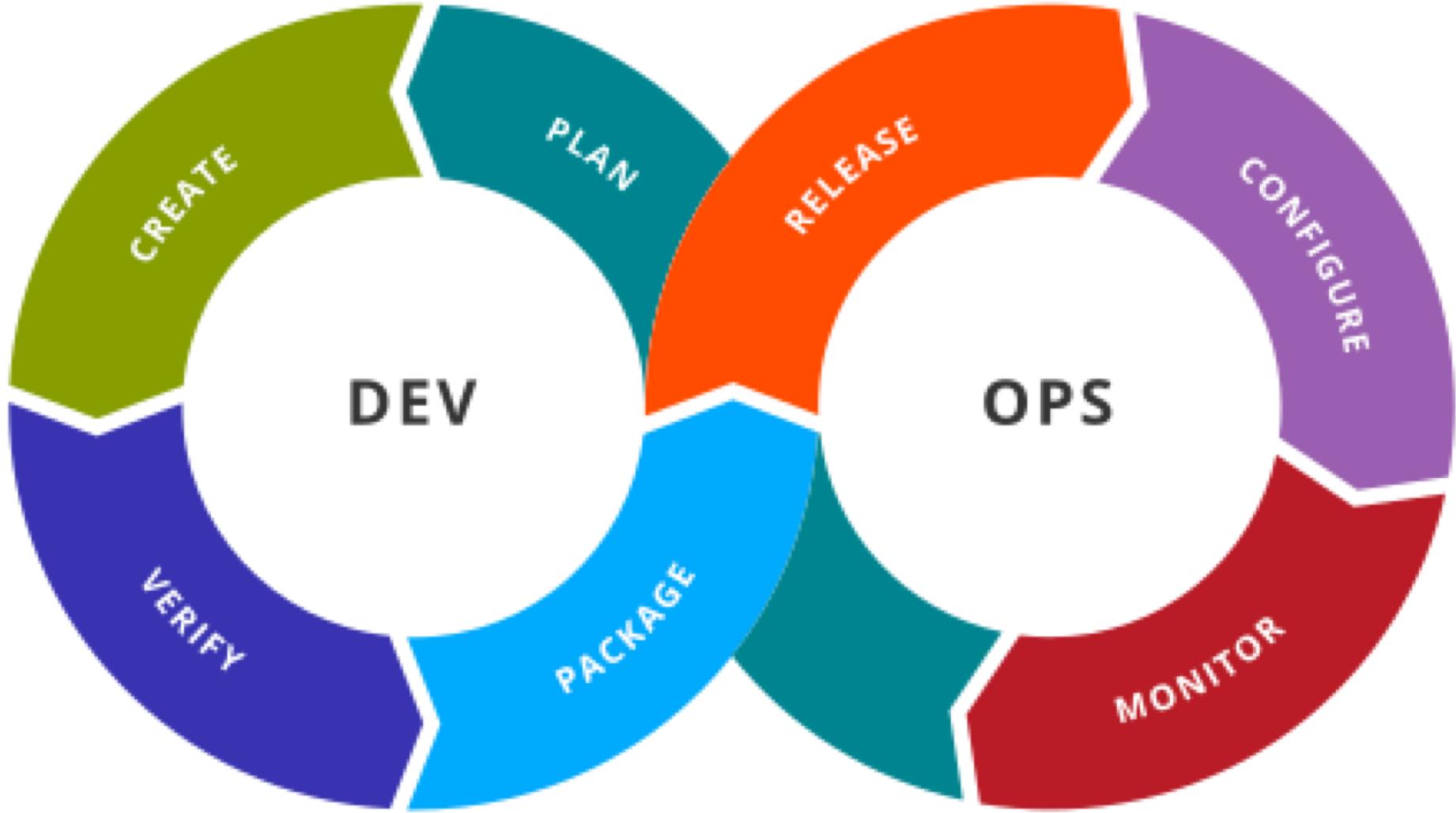
Discussion of Microservices

- 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



Exercise

Decompose system in reading



Claire: Design Doc exercise