Architecture
Styles and Hypes

Michael Hilton Claire Le Goues

Christopher Meiklejohn

October 15, 2020

Administrativia

- Homework 4 will be released today.
- No recitation Friday.
- Wednesday recitation – bring questions or we end early!
  - Work through problems on the previous midterms – many students found this helpful.
  - Any questions on the previous midterm questions – bring them to recitation to discuss as a class.
- Midterm on October 22nd.

Learning Goals

- Understand history of Microservices
- Reason about tradeoffs of Microservices architectures.

MICROSERVICES

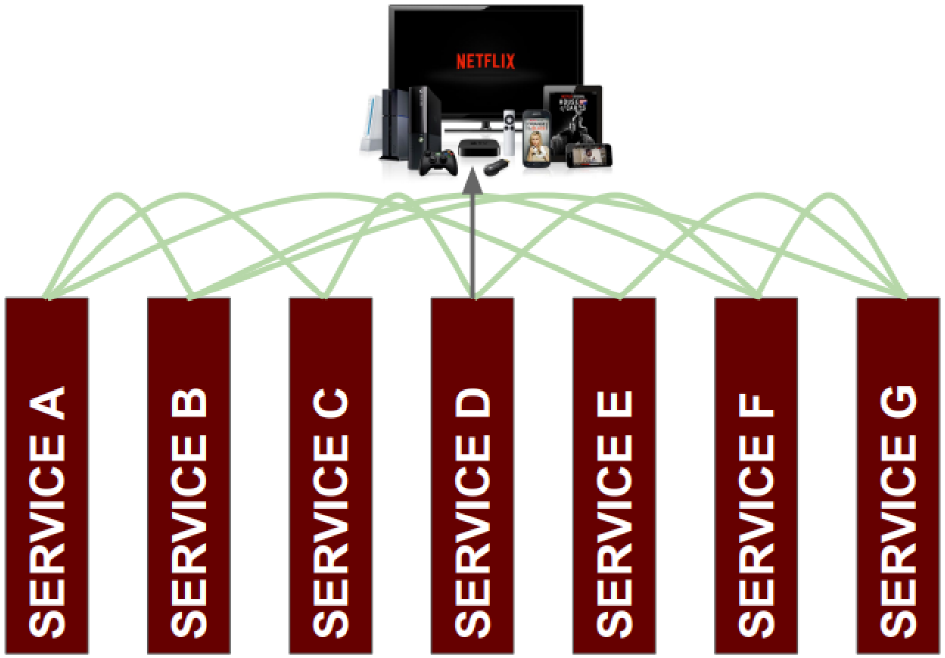
Microservices
Microservices Everywhere

WHAT ARE MICROSERVICES?
SERVICES FOR ANTS???!
makeameme.org

MICROSERVICES...
...SO HOT RIGHT NOW
memegenerator.net

Netflix

AppBoot



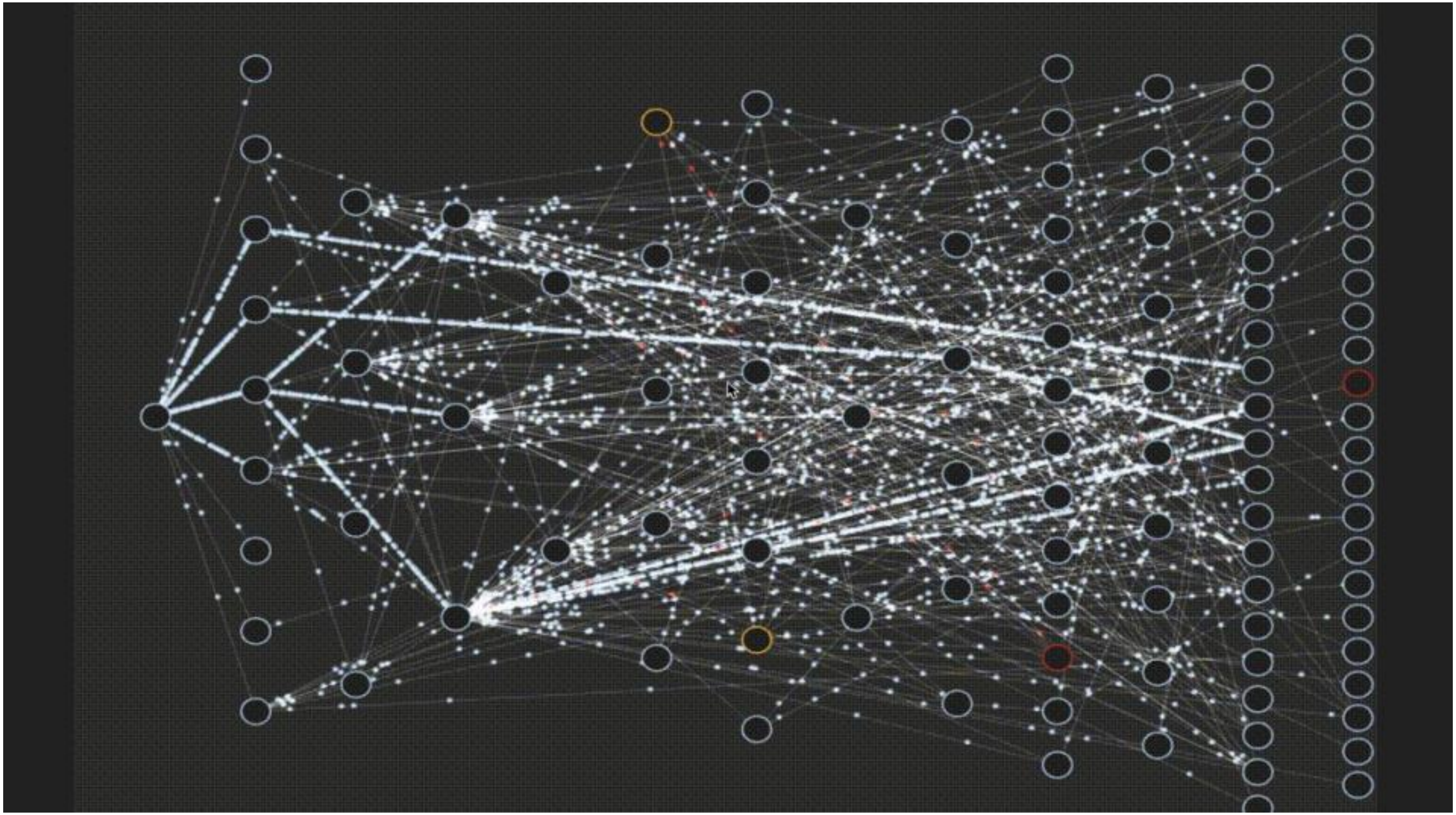
Bookmarks

Recommendations

My List

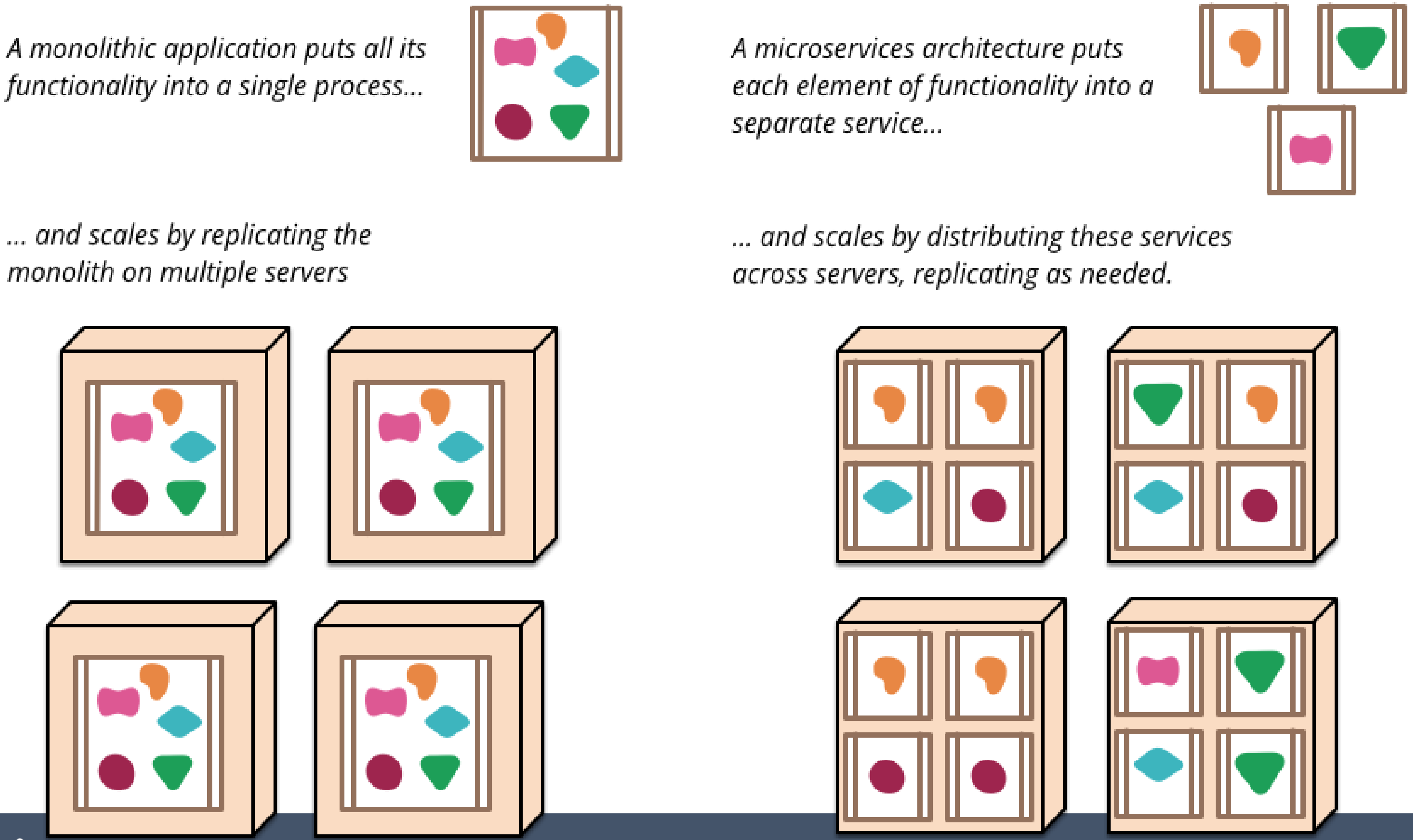
Metrics

(as of 2016)



(as of 2016)

Microservices



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

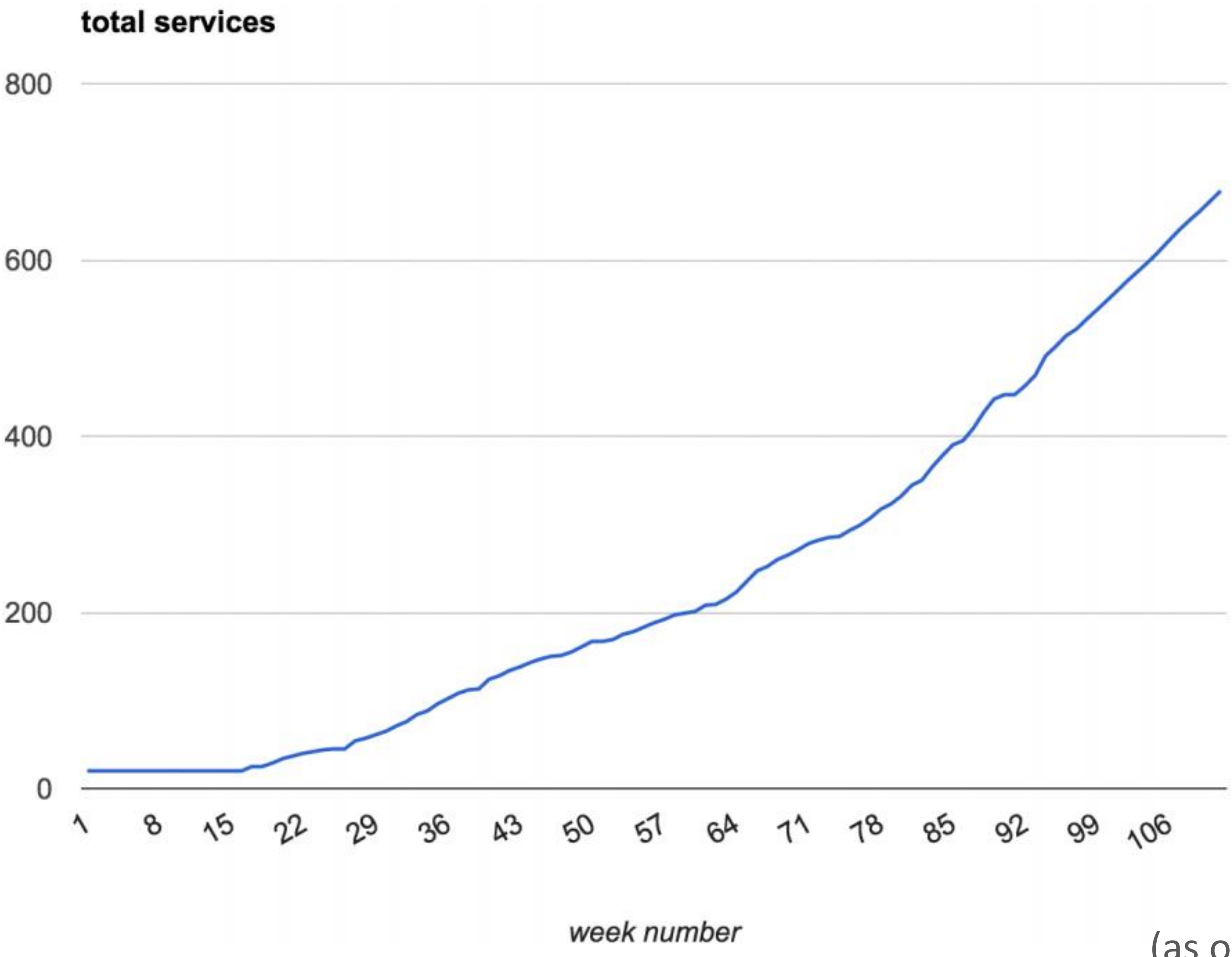
... and scales by replicating the monolith on multiple servers

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

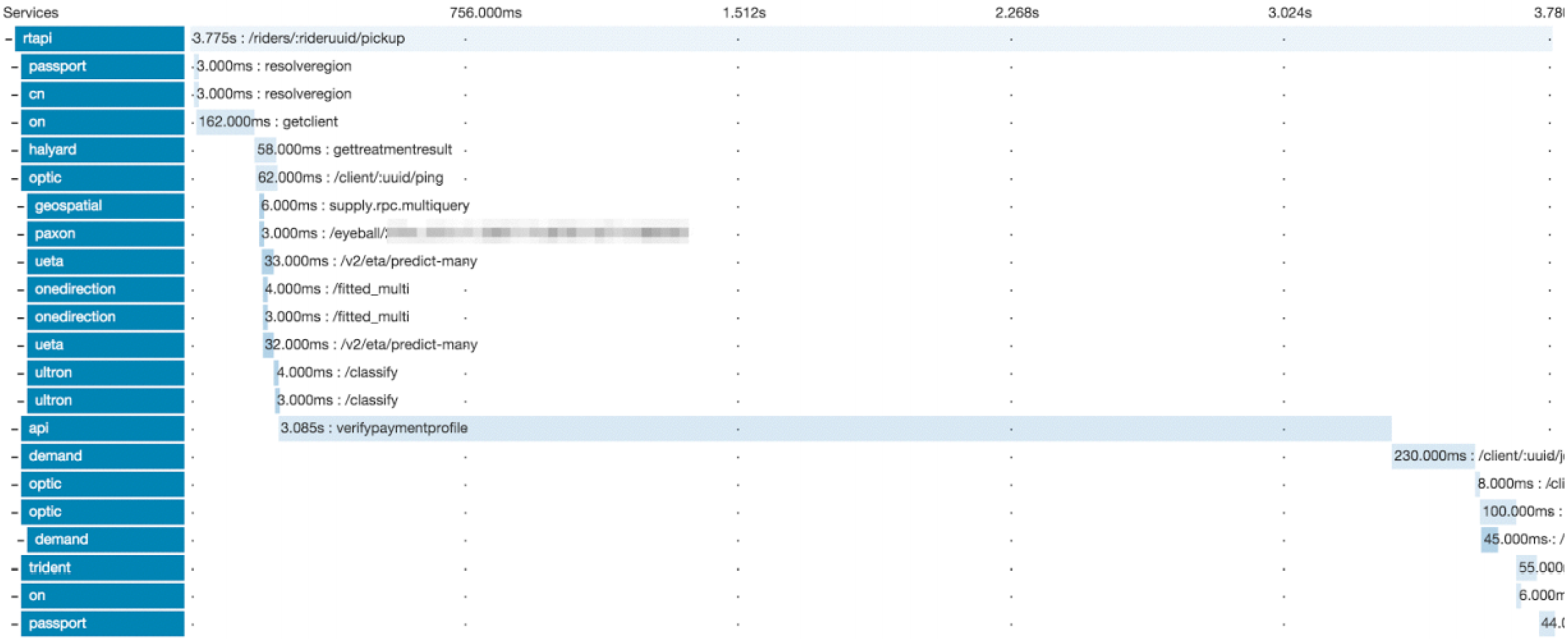
... and scales by distributing these services across servers, replicating as needed.

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

Uber

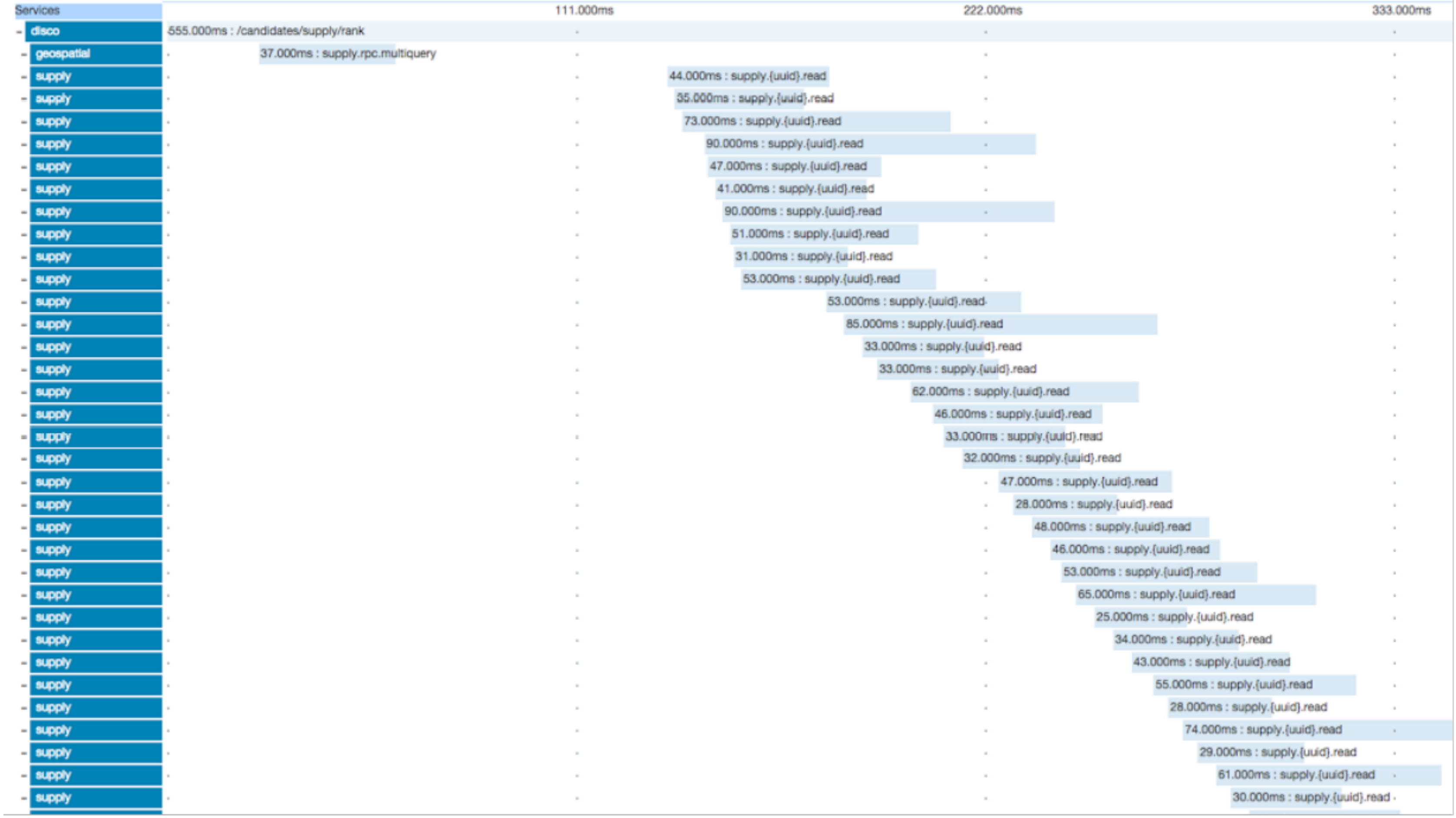

total services
800
600
400
200
0
1
8
15
22
29
36
43
50
57
64
71
78
85
92
99
106
week number

(as of 2016)



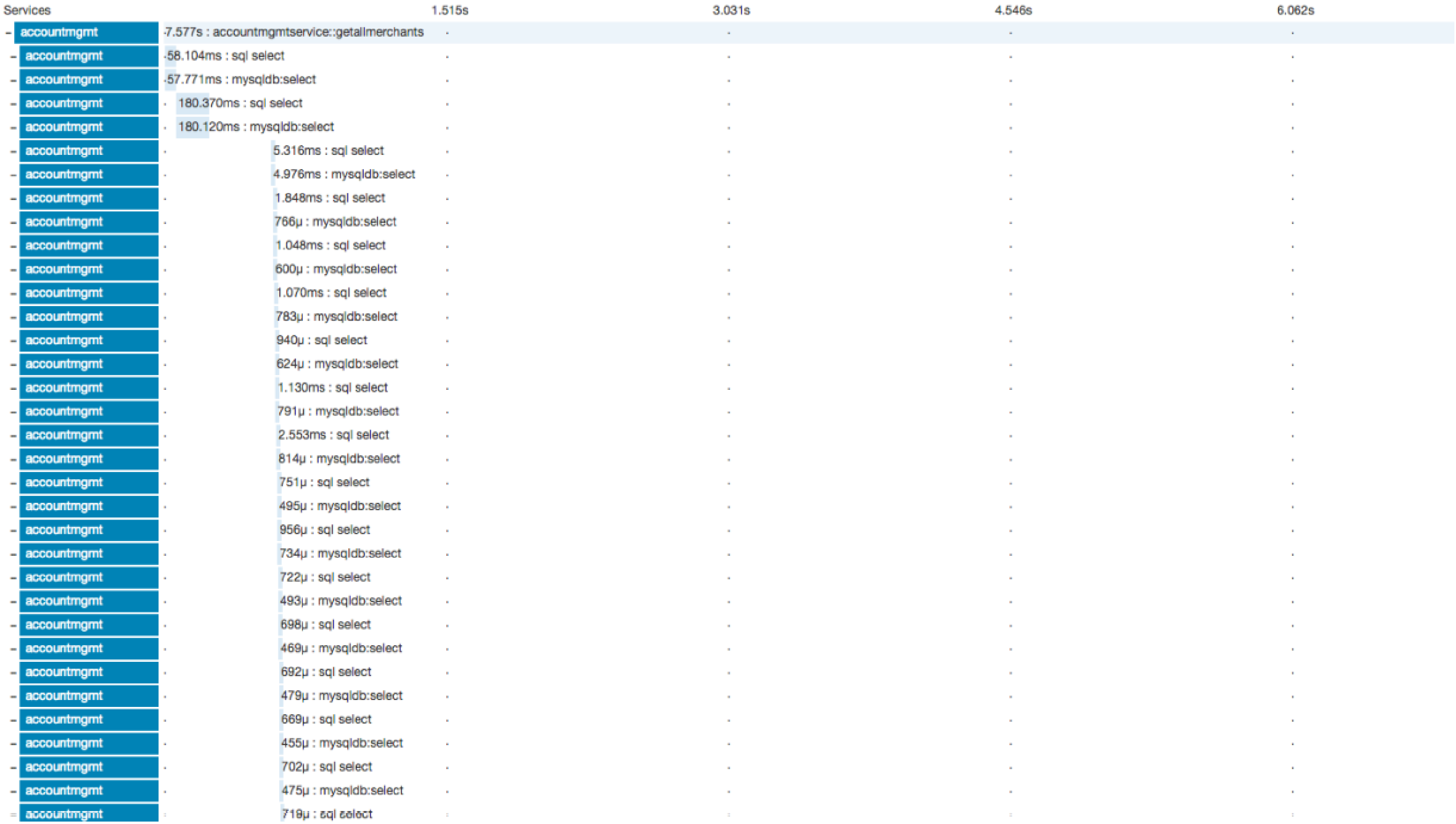
```
Services                                      756.000ms      1.512s      2.268s      3.024s      3.78
- rtapi             3.775s : /riders/:rideruuid/pickup
 - passport         3.000ms : resolveregion
 - cn               3.000ms : resolveregion
 - on               162.000ms : getclient
 - halyard              58.000ms : gettreatmentresult
 - optic                62.000ms : /client/:uuid/ping
  - geospatial           6.000ms : supply.rpc.multiquery
  - paxon                3.000ms : /eyeball/[redacted]
  - ueta                 33.000ms : /v2/eta/predict-many
  - onedirection         4.000ms : /fitted_multi
  - onedirection         3.000ms : /fitted_multi
  - ueta                 32.000ms : /v2/eta/predict-many
  - ultron                4.000ms : /classify
  - ultron                3.000ms : /classify
 - api                     3.085s : verifypaymentprofile
 - demand                                                                            230.000ms : /client/:uuid/j
 - optic                                                                                       8.000ms : /cli
 - optic                                                                                       100.000ms :
  - demand                                                                                     45.000ms : /
 - trident                                                                                         55.000
 - on                                                                                              6.000m
 - passport                                                                                          44.0
```

(as of 2016)



```
Services          111.000ms          222.000ms          333.000ms
- disco       555.000ms : /candidates/supply/rank
- geospatial  37.000ms : supply.rpc.multiquery
- supply      44.000ms : supply.{uuid}.read
- supply      35.000ms : supply.{uuid}.read
- supply      73.000ms : supply.{uuid}.read
- supply      90.000ms : supply.{uuid}.read
- supply      47.000ms : supply.{uuid}.read
- supply      41.000ms : supply.{uuid}.read
- supply      90.000ms : supply.{uuid}.read
- supply      51.000ms : supply.{uuid}.read
- supply      31.000ms : supply.{uuid}.read
- supply      53.000ms : supply.{uuid}.read
- supply      53.000ms : supply.{uuid}.read
- supply      85.000ms : supply.{uuid}.read
- supply      33.000ms : supply.{uuid}.read
- supply      33.000ms : supply.{uuid}.read
- supply      62.000ms : supply.{uuid}.read
- supply      46.000ms : supply.{uuid}.read
- supply      33.000ms : supply.{uuid}.read
- supply      32.000ms : supply.{uuid}.read
- supply      47.000ms : supply.{uuid}.read
- supply      28.000ms : supply.{uuid}.read
- supply      48.000ms : supply.{uuid}.read
- supply      46.000ms : supply.{uuid}.read
- supply      53.000ms : supply.{uuid}.read
- supply      65.000ms : supply.{uuid}.read
- supply      25.000ms : supply.{uuid}.read
- supply      34.000ms : supply.{uuid}.read
- supply      43.000ms : supply.{uuid}.read
- supply      55.000ms : supply.{uuid}.read
- supply      28.000ms : supply.{uuid}.read
- supply      74.000ms : supply.{uuid}.read
- supply      29.000ms : supply.{uuid}.read
- supply      61.000ms : supply.{uuid}.read
- supply      30.000ms : supply.{uuid}.read
```

(as of 2016)



```
Services                                          1.515s        3.031s        4.546s        6.062s
- accountmgmt   7.577s : accountmgmtservice::getallmerchants
 - accountmgmt  58.104ms : sql select
 - accountmgmt  57.771ms : mysqldb:select
 - accountmgmt  180.370ms : sql select
 - accountmgmt  180.120ms : mysqldb:select
 - accountmgmt  5.316ms : sql select
 - accountmgmt  4.976ms : mysqldb:select
 - accountmgmt  1.848ms : sql select
 - accountmgmt  766µ : mysqldb:select
 - accountmgmt  1.048ms : sql select
 - accountmgmt  600µ : mysqldb:select
 - accountmgmt  1.070ms : sql select
 - accountmgmt  783µ : mysqldb:select
 - accountmgmt  940µ : sql select
 - accountmgmt  624µ : mysqldb:select
 - accountmgmt  1.130ms : sql select
 - accountmgmt  791µ : mysqldb:select
 - accountmgmt  2.553ms : sql select
 - accountmgmt  814µ : mysqldb:select
 - accountmgmt  751µ : sql select
 - accountmgmt  495µ : mysqldb:select
 - accountmgmt  956µ : sql select
 - accountmgmt  734µ : mysqldb:select
 - accountmgmt  722µ : sql select
 - accountmgmt  493µ : mysqldb:select
 - accountmgmt  698µ : sql select
 - accountmgmt  469µ : mysqldb:select
 - accountmgmt  692µ : sql select
 - accountmgmt  479µ : mysqldb:select
 - accountmgmt  669µ : sql select
 - accountmgmt  455µ : mysqldb:select
 - accountmgmt  702µ : sql select
 - accountmgmt  475µ : mysqldb:select
 - accountmgmt  718µ : sql select
```

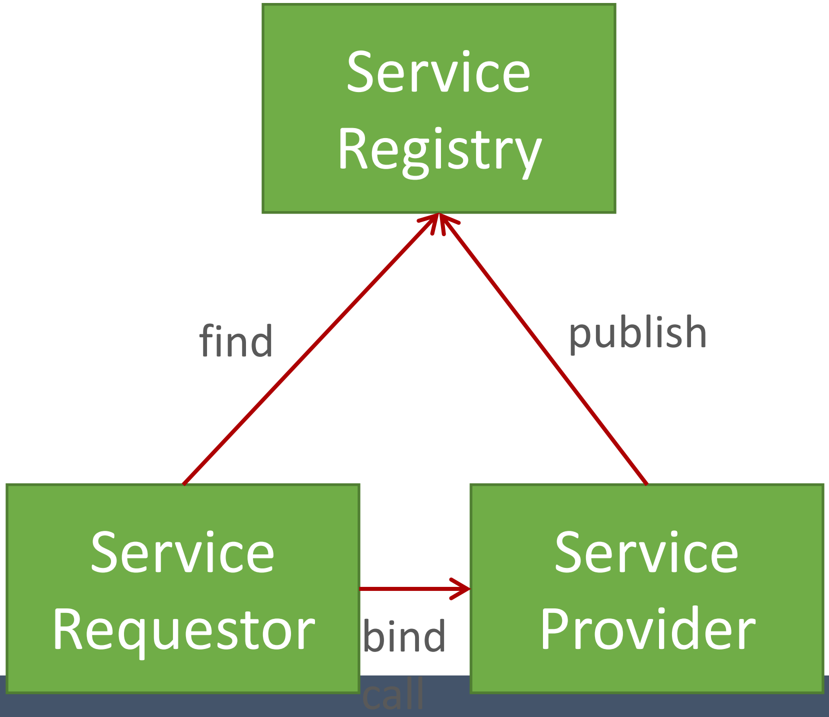
(as of 2016)

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
- Easily deployable and replicable
- Embrace automation, embrace faults
- Highly observable

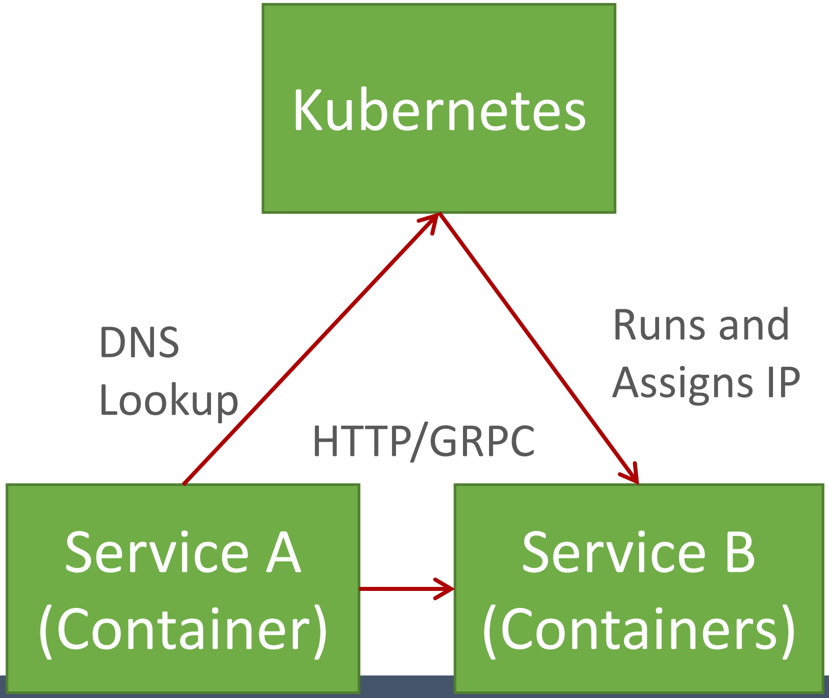
Service Oriented Architectures (SOA)

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



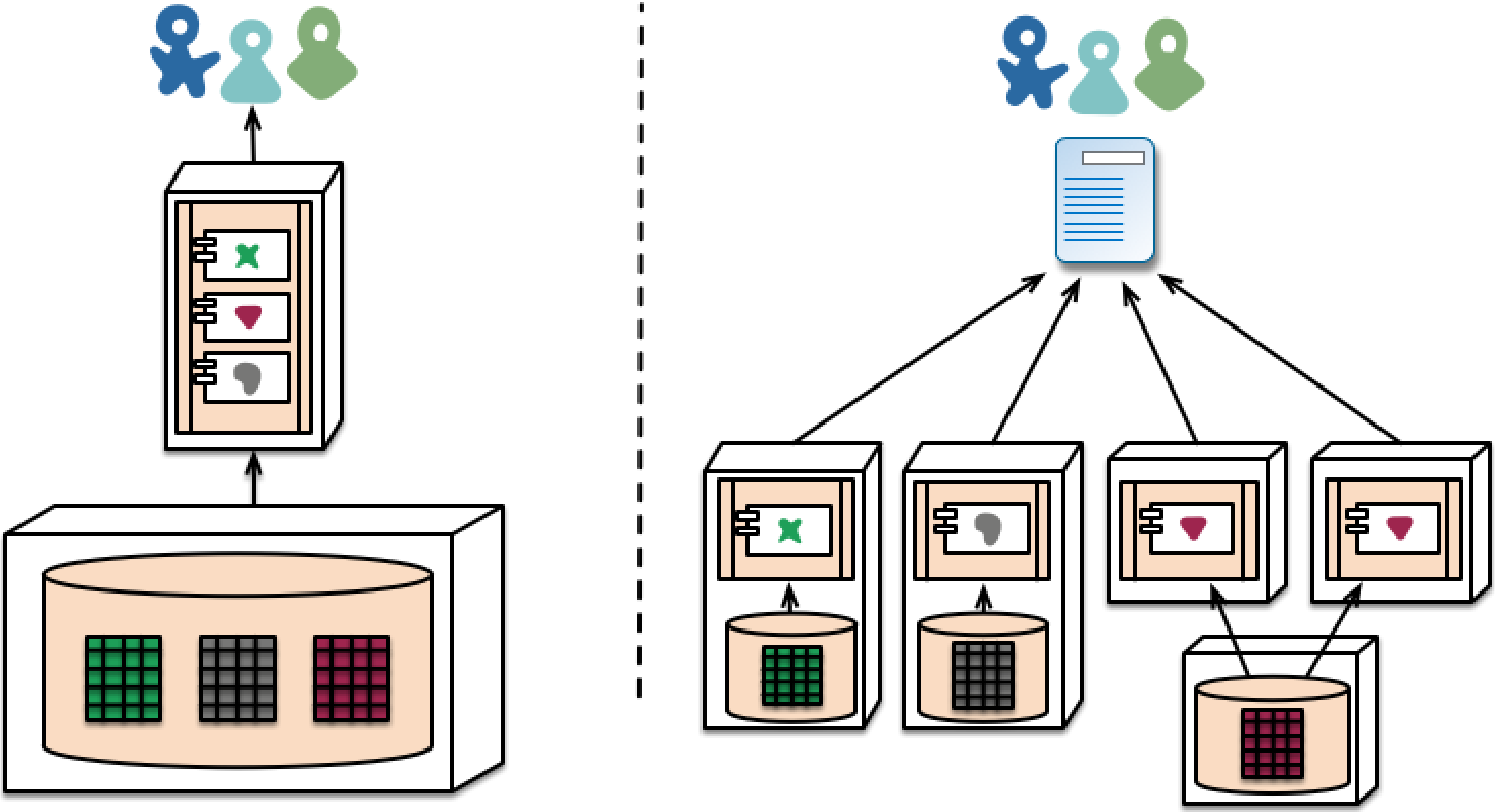
~~Service Oriented Architectures (SOA)~~ Microservice Architecture

- Service: self-contained functionality
- Language-independent interface
- Dynamic lookup



Technical Considerations

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



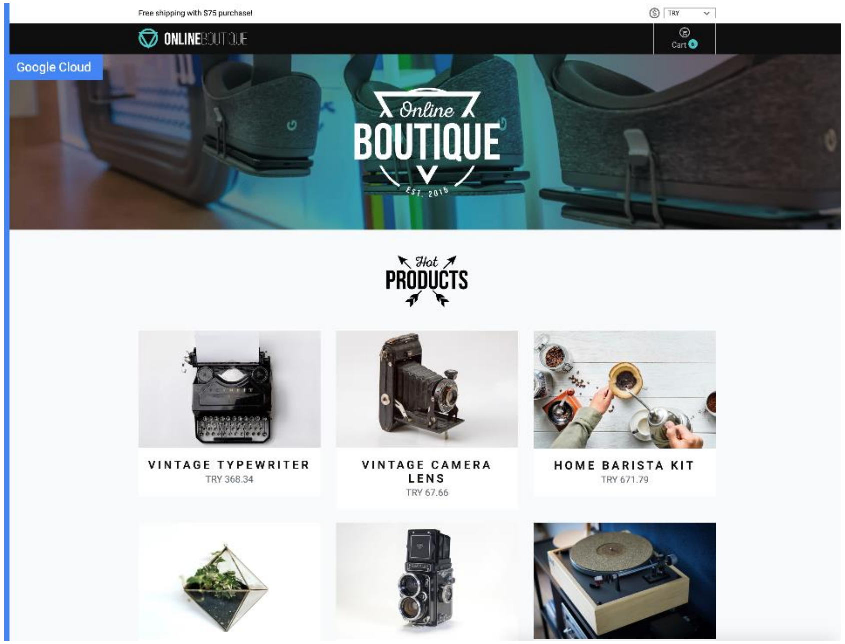
monolith - single database

microservices - application databases

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

Hipster Shop

Hipster Shop User Interface

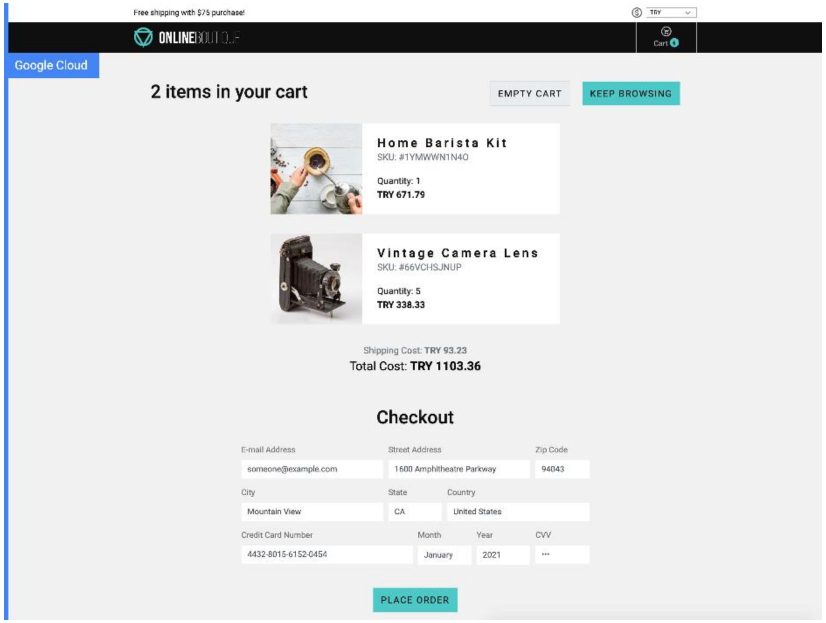


Free shipping with $75 purchase!

VINTAGE TYPEWRITER
TRY 368.34

VINTAGE CAMERA LENS
TRY 67.66

HOME BARISTA KIT
TRY 671.79



Free shipping with $75 purchase!

2 items in your cart

Home Barista Kit
SKU: #1YMWWN1N4O
Quantity: 1
TRY 671.79

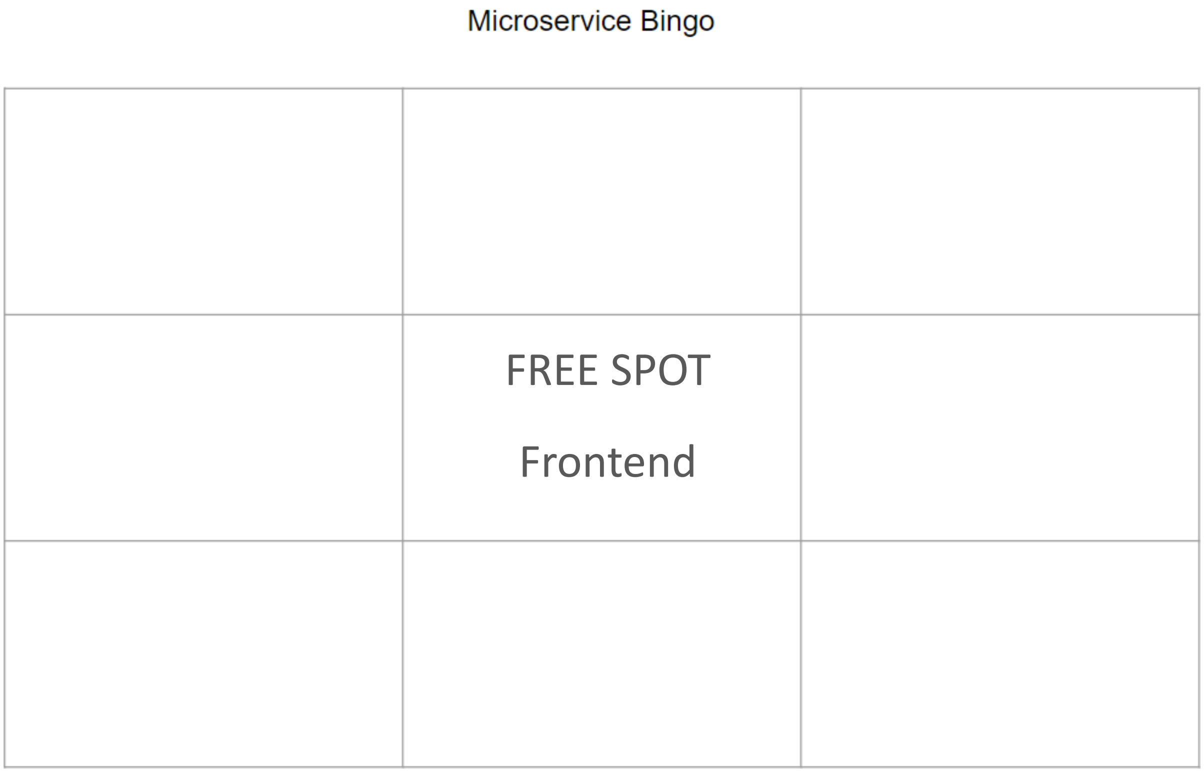
Vintage Camera Lens
SKU: #66VCHSJNUP
Quantity: 5
TRY 338.33

Shipping Cost: TRY 93.23
Total Cost: TRY 1103.36

Checkout

E-mail Address: someone@example.com
Street Address: 1600 Amphitheatre Parkway
Zip Code: 94043
City: Mountain View
State: CA
Country: United States
Credit Card Number: 4432-8015-6152-0454
Month: January
Year: 2021
CVV: •••

Hipster Shop Bingo Game



Microservice Bingo

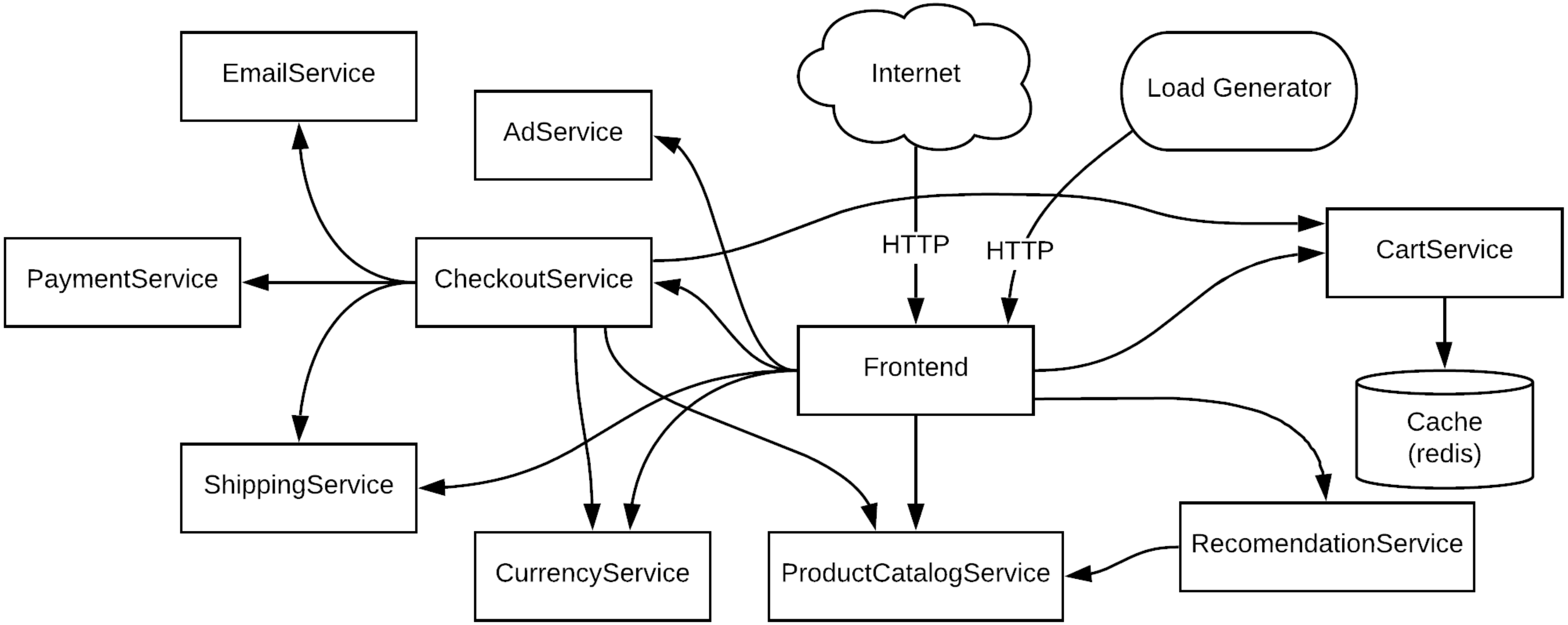
	FREE SPOT Frontend	

Make a copy of the first slide for your group.

Add your Andrew IDs to the slide.

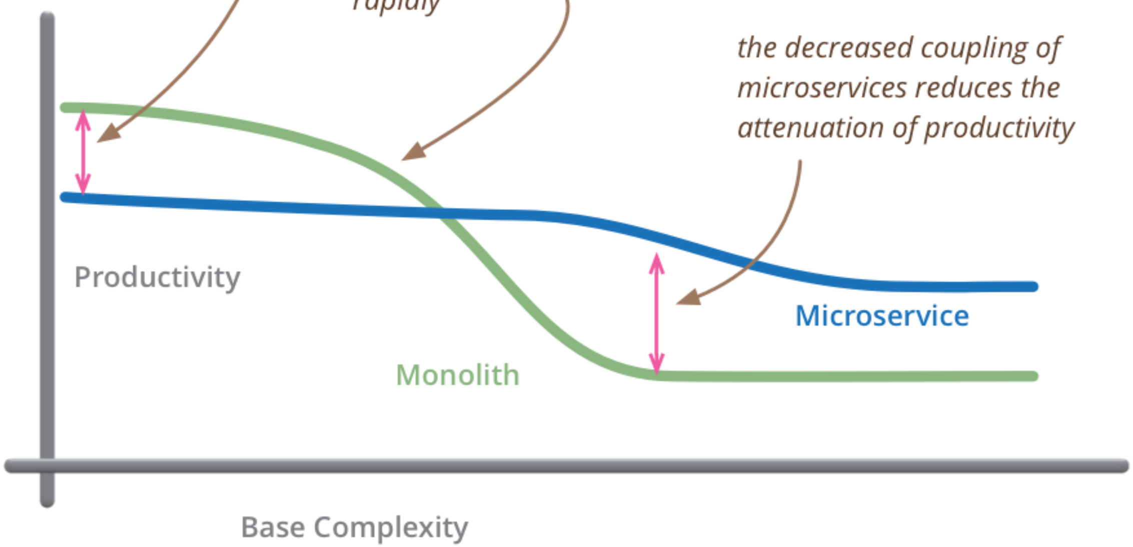
https://docs.google.com/presentation/d/1P7X7nFMIWAQW12kOk6pW66jtk34S_j_RnCt2c3BwMvM/edit?usp=sharing

Hipster Shop Microservice Architecture



https://github.com/GoogleCloudPlatform/microservices-demo

Microservices overhead



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

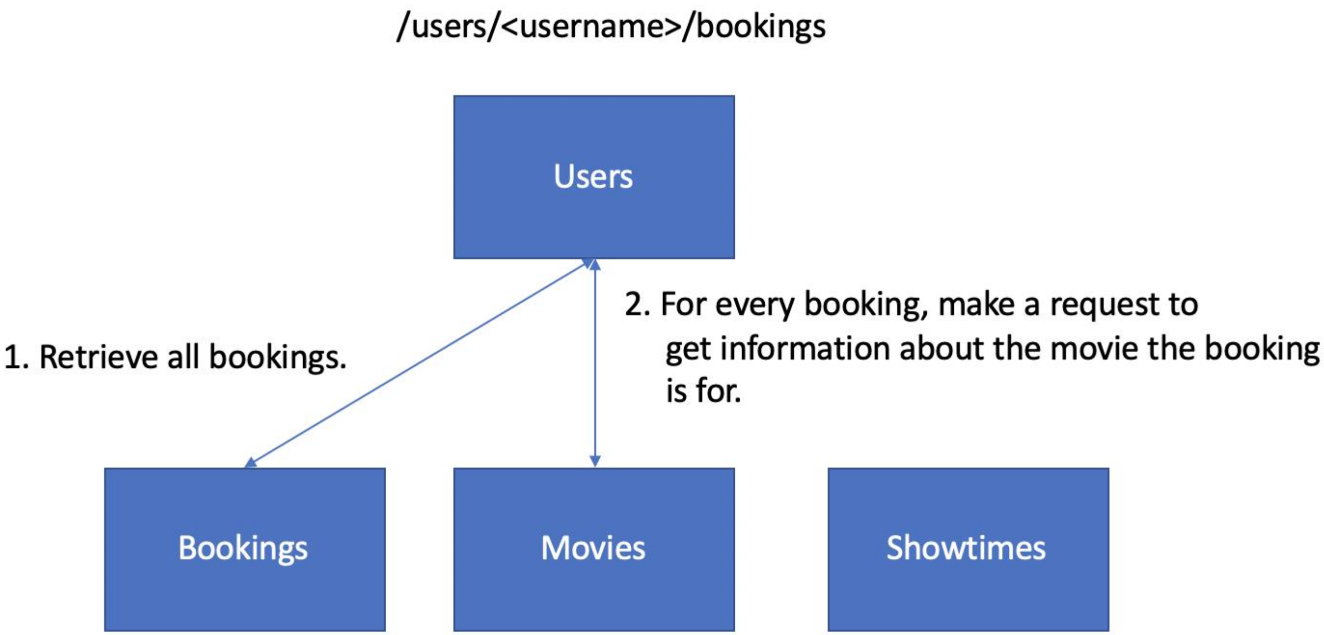
as complexity kicks in, productivity starts falling rapidly

the decreased coupling of microservices reduces the attenuation of productivity

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

Cinema

Cinema Diagram


/users/<username>/bookings
Users
1. Retrieve all bookings.
2. For every booking, make a request to get information about the movie the booking is for.
Bookings
Movies
Showtimes

https://codeahoy.com/2016/07/10/writing-microservices-in-python-using-flask/
https://github.com/umermansoor/microservices/

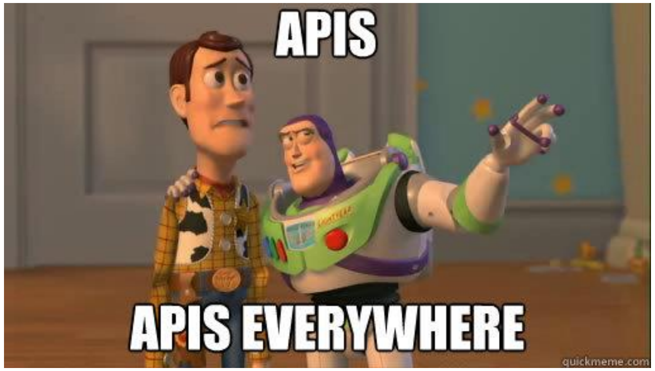
Cinema Code Walkthrough

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
  - Schemas?

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



Serverless

SERVERLESS
SERVERLESS
Everywhere

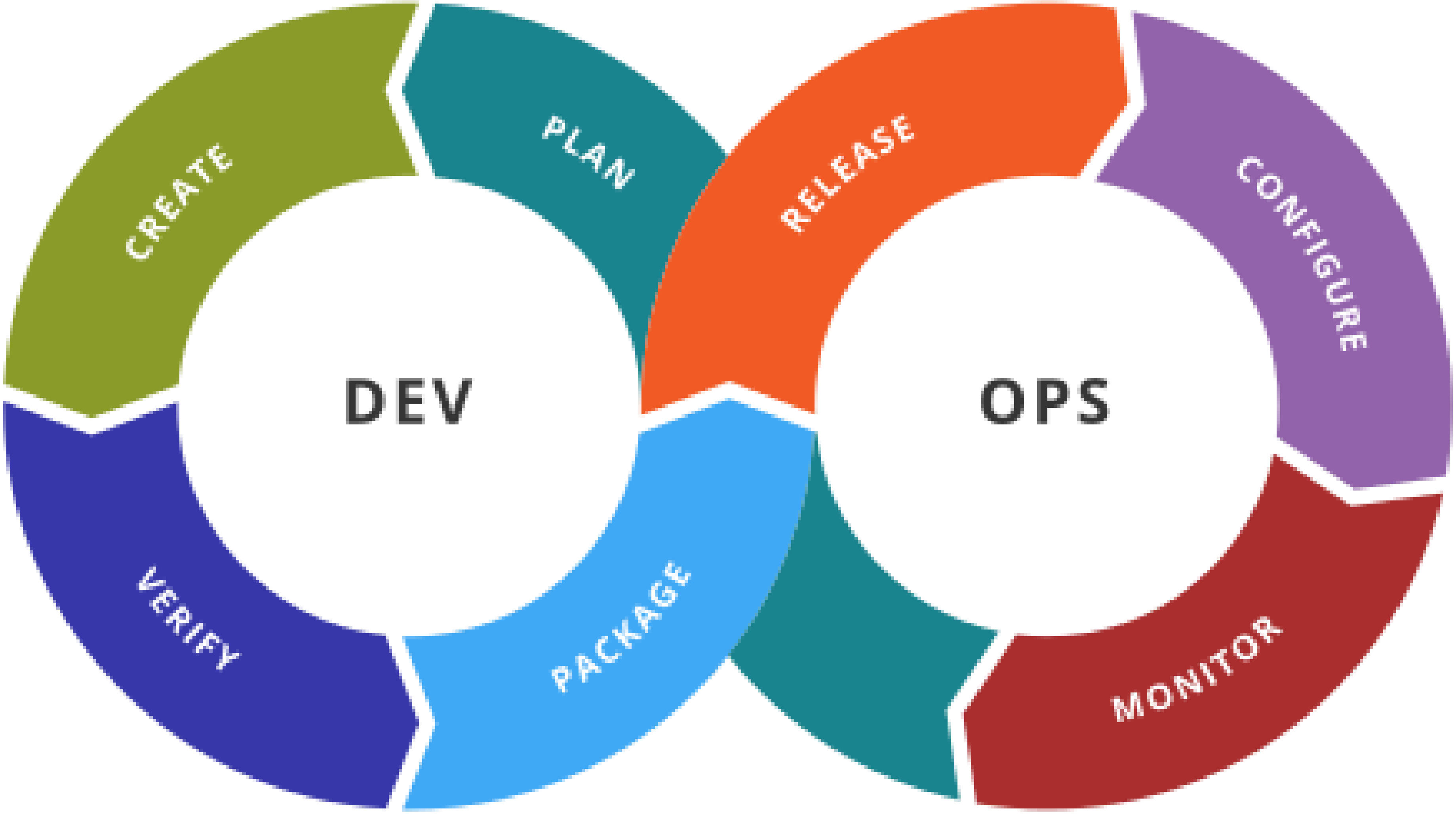
WHAT ARE
SERVERLESS
SERVICES FOR ANTS???!
makeameme.org

SERVERLESS
...SO HOT RIGHT NOW
memegenerator.net

Serverless (Functions-as-a-Service)

- "extreme" use of microservices
- Instead of writing minimal services, write just functions
- No state, rely completely on cloud storage or other cloud services
- Pay-per-invocation billing with elastic scalability
- Drawback: more ways things can fail, state is expensive
- Examples:
  AWS lambda, CloudFlare workers, Azure Functions
- What might this be good for?

- (New in 2019/20) Stateful Functions:
  Azure Durable Entities, CloudFlare Durable Objects


CREATE
PLAN
VERIFY
PACKAGE
DEV
RELEASE
CONFIGURE
MONITOR
OPS