



Microservices pitfalls

Addressing the most frequent pitfalls when
transitioning to Microservices

Magnus Kulke

Engineering Manager

github.com/mkulke

lnkd.in/magnuskulke



Lothar Schulz

Head of Engineering

lotharschulz.info

github.com/lotharschulz

[lnkd.in/lotharschulz](https://lnkd.in/g/lotharschulz)





Contracts

Lawyer up!

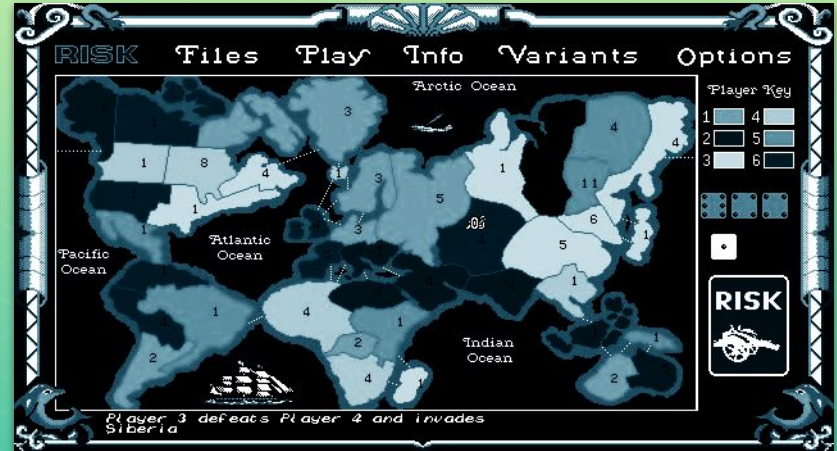
**Ambiguities and Unmet
Expectations**

Microservices are (also/primarily?) a social tool

- There is a relation between architecture and team setup
- **“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization’s communication structure.”**

Conway’s Law

- Enables teams to make autonomous decisions

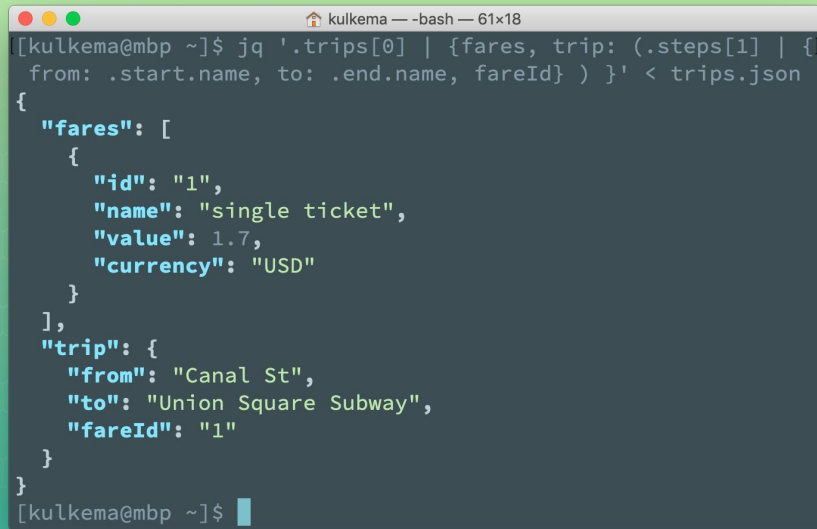


Service Boundaries are Defined by Contracts

- Codify expectations towards an API from the consumer's perspective
 - Behaviour: does not change unexpectedly
 - Availability: when can we retire an API?
- How to express such a contract?
 - Machine readable: Swagger/OpenAPI, JSON Schema, GraphQL
 - API Versions
- Abstain from breaking changes
 - Additional properties?
 - Extending enums?
- Make everything optional: Protobuf3

Problem: A Schema might not be expressive enough

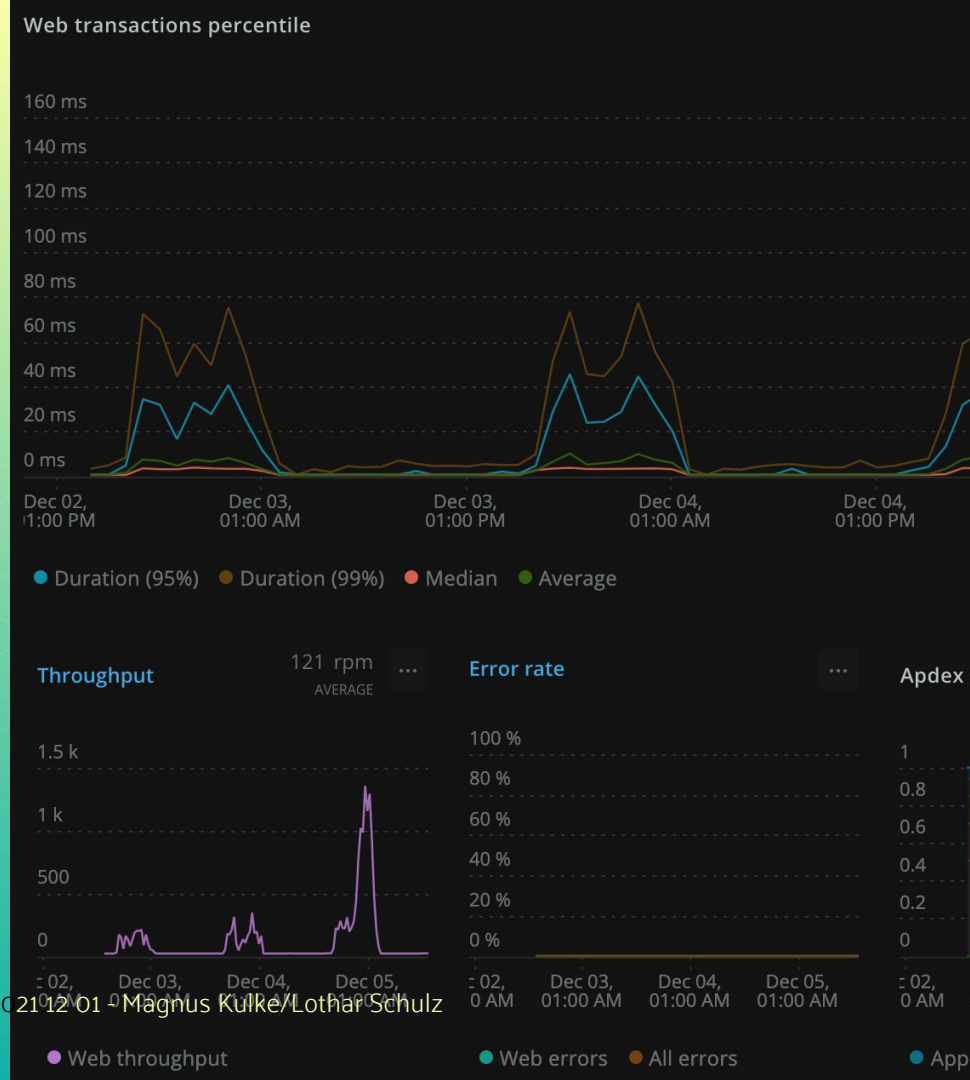
- Documents can be formally correct
- But semantics have changed
 - References in a document
 - Content: New ID for entity
- Pragmatic solution: Contract tests



```
kulkema — -bash — 61x18
[kulkema@mbp ~]$ jq '.trips[0] | {fares, trip: (.steps[1] | {
  from: .start.name, to: .end.name, fareId} ) }' < trips.json
{
  "fares": [
    {
      "id": "1",
      "name": "single ticket",
      "value": 1.7,
      "currency": "USD"
    }
  ],
  "trip": {
    "from": "Canal St",
    "to": "Union Square Subway",
    "fareId": "1"
  }
}
```

Performance Characteristics

- Service level objectives
- Rate limits
- Request budgets



The Other Side: Protection from Harmful Workloads

- Unforeseen (ab)use patterns
- How to attribute incoming traffic?
 - Correlation Ids
 - Callers need to tag their requests
- Manage access
 - Service Accounts
 - Declarative: Service Mesh



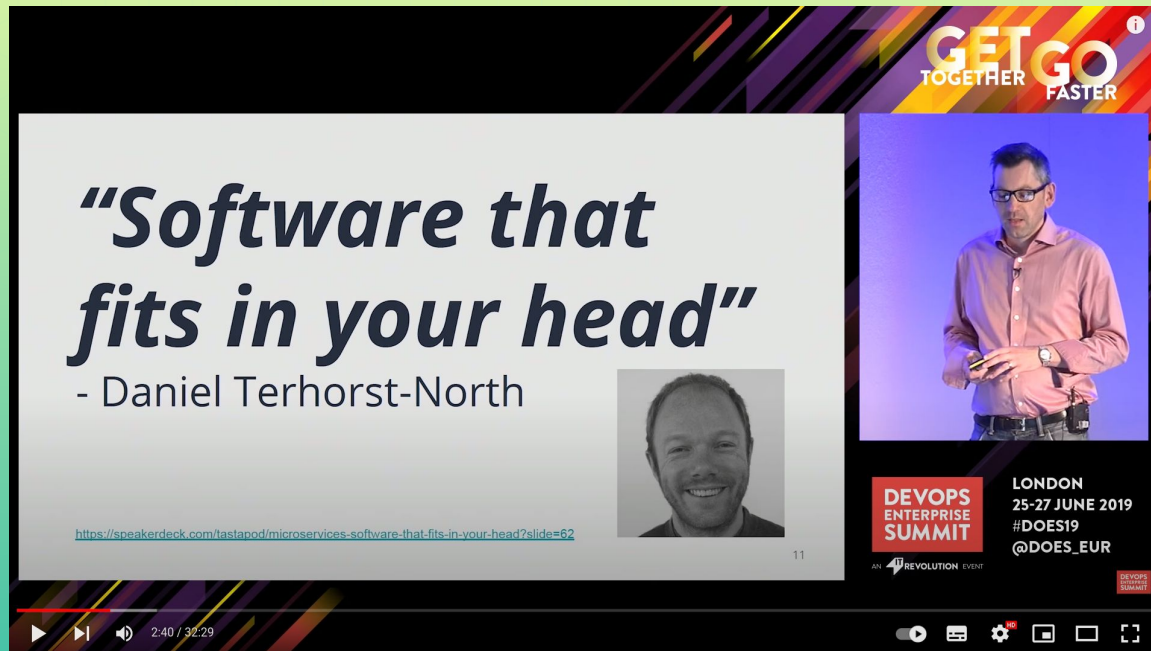
Domains

None of your concern!
Slicing microservices
properly

Database as Microservice



How small is *micro* ?



The screenshot shows a video player interface. The main content area displays a presentation slide with the title **"Software that fits in your head"** in a large, bold, italicized font. Below the title is the name **- Daniel Terhorst-North**. A small circular portrait of the speaker is visible on the right side of the slide. At the bottom of the slide, a URL is provided: <https://speakerdeck.com/tastapod/microservices-software-that-fits-in-your-head?slide=62>. The slide number **11** is in the bottom right corner. To the right of the slide, there is a vertical video frame showing the speaker, Daniel Terhorst-North, a man with glasses wearing a pink shirt, holding a small object. Above this frame is the **GET GO TOGETHER FASTER** logo. Below the frame, text indicates the event: **DEVOPS ENTERPRISE SUMMIT**, **LONDON 25-27 JUNE 2019**, **#DOES19**, and **@DOES_EUR**. The video player controls at the bottom include a play button, a progress bar showing **2:40 / 32:29**, and icons for volume, settings, and full screen.

Monoliths vs Microservices is Missing the Point—Start with Team Cognitive Load - Team Topologies

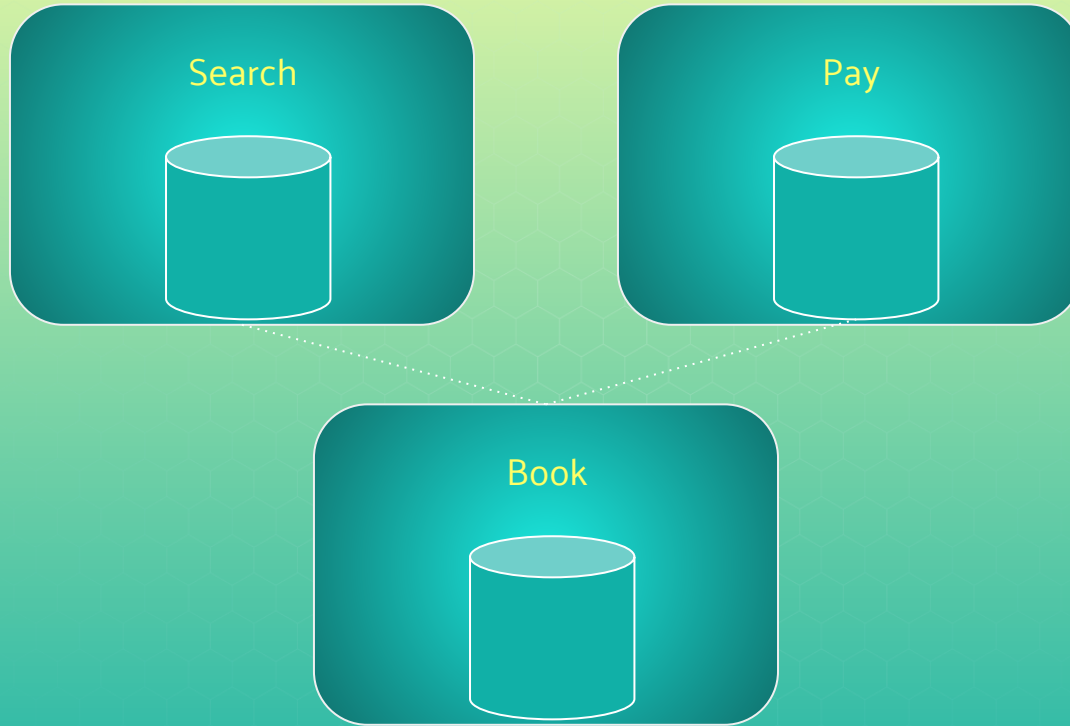
<https://speakerdeck.com/tastapod/microservices-software-that-fits-in-your-head?slide=62>

Addressing the most frequent pitfalls when transitioning to Microservices - 2021 12 01 - Magnus Kulke/Lothar Schulz

Monolith first

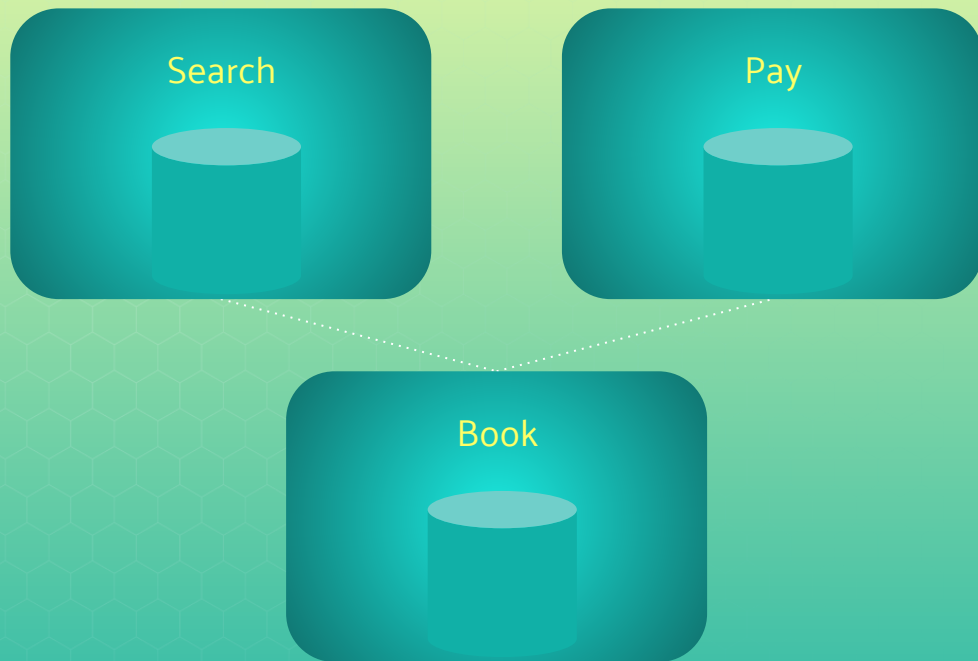


Domains



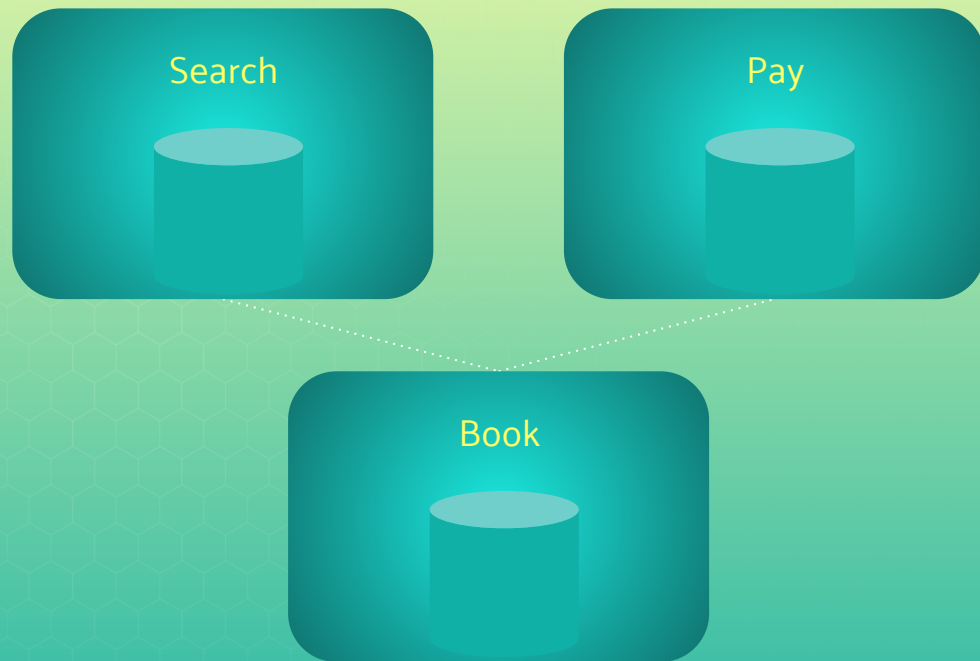
Domains

Scaling



Domains

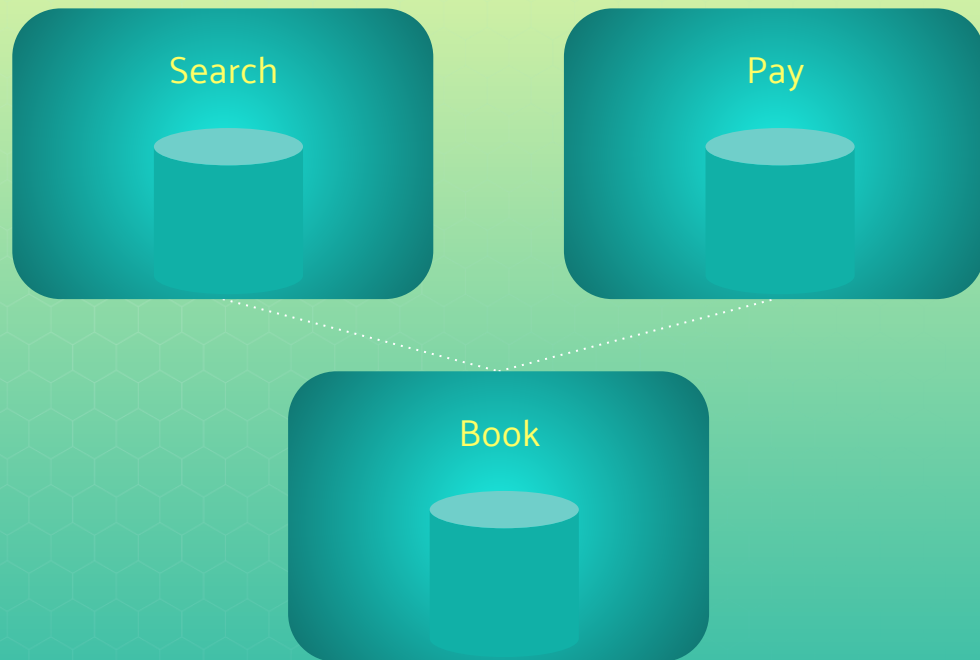
Scaling
- Vertical



Domains

Scaling

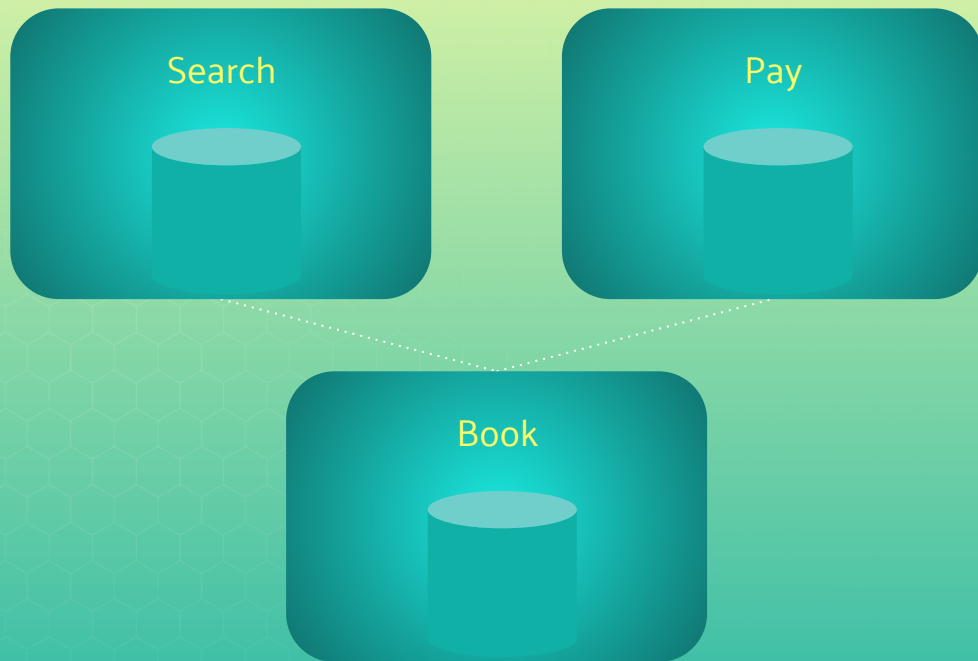
- Vertical
- Horizontal



Domains

Scaling

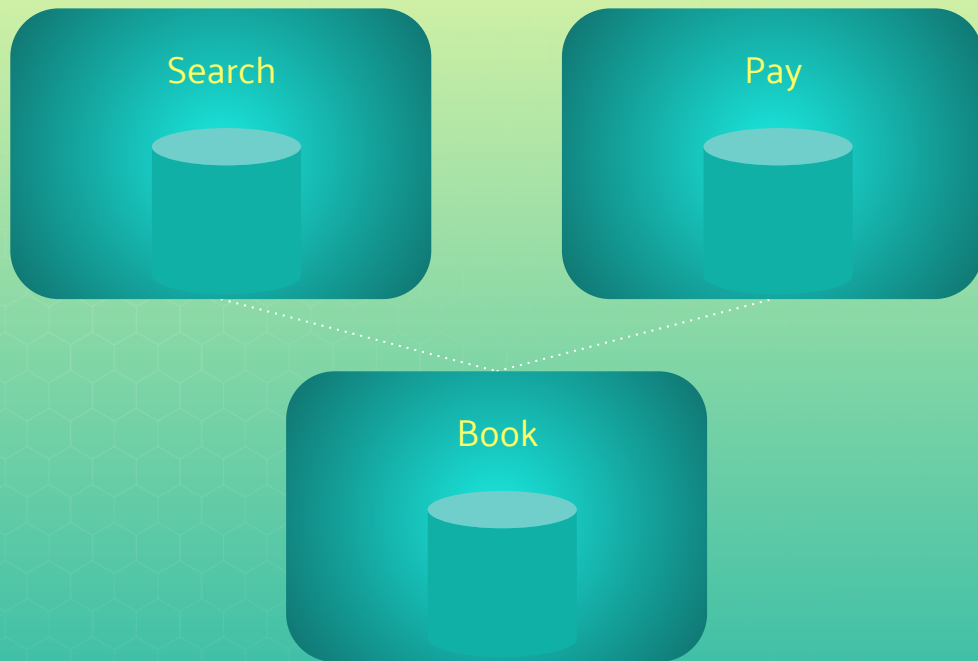
- Vertical
- Horizontal
- Partitioning



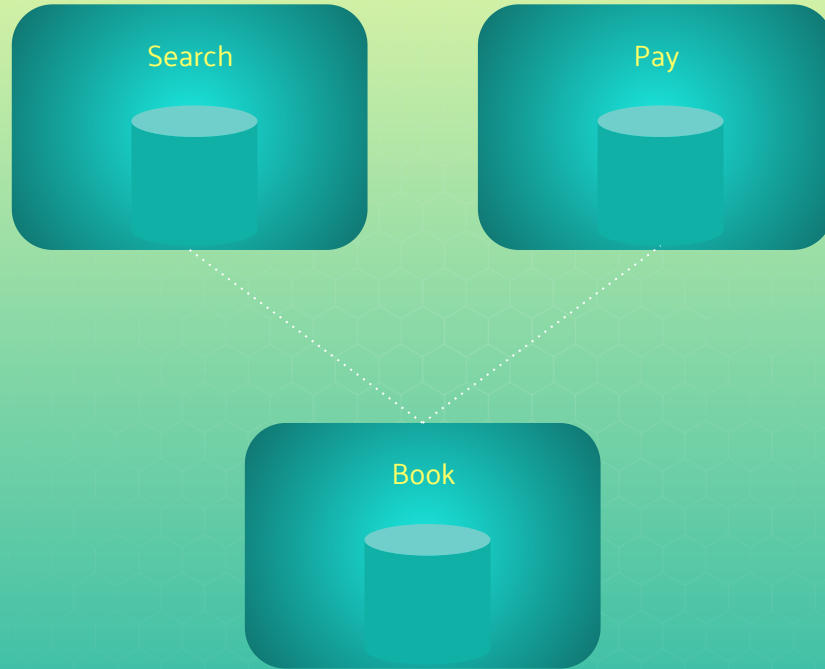
Domains

Scaling

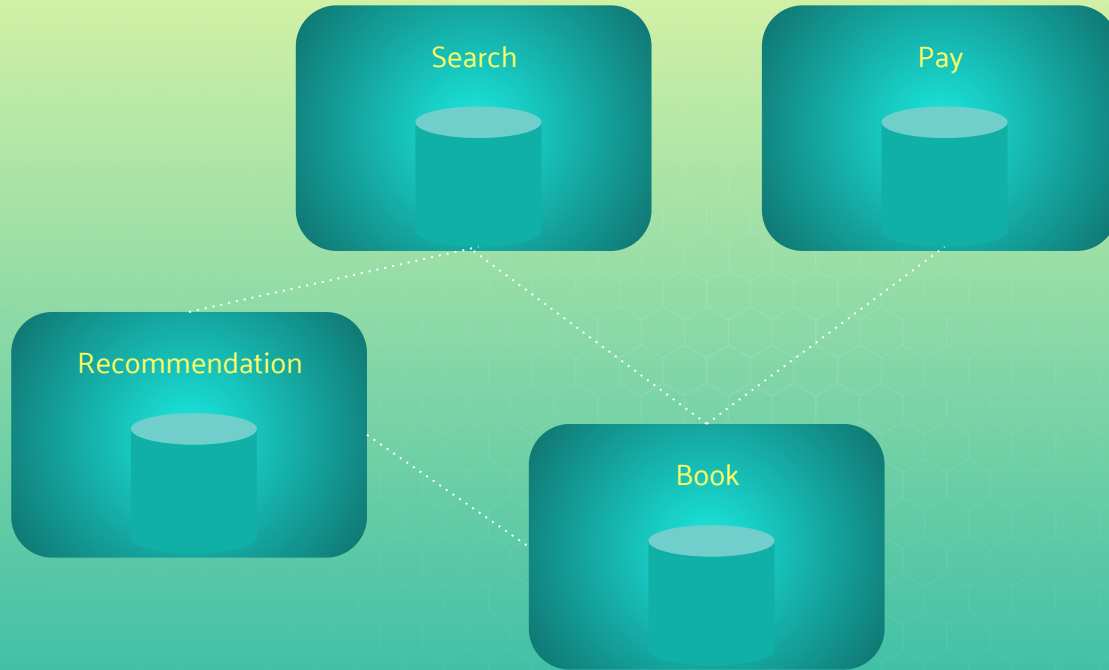
- Vertical
- Horizontal
- Partitioning
 - Sharding



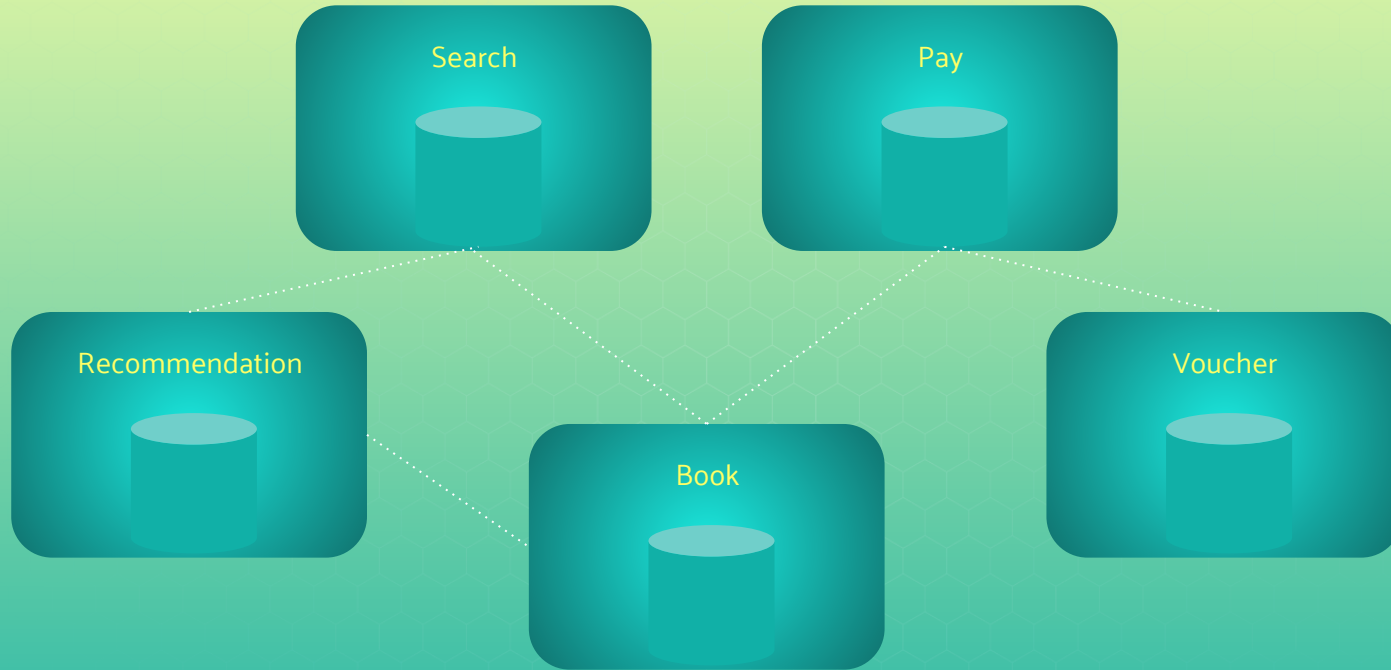
Domains – Bounded Contexts



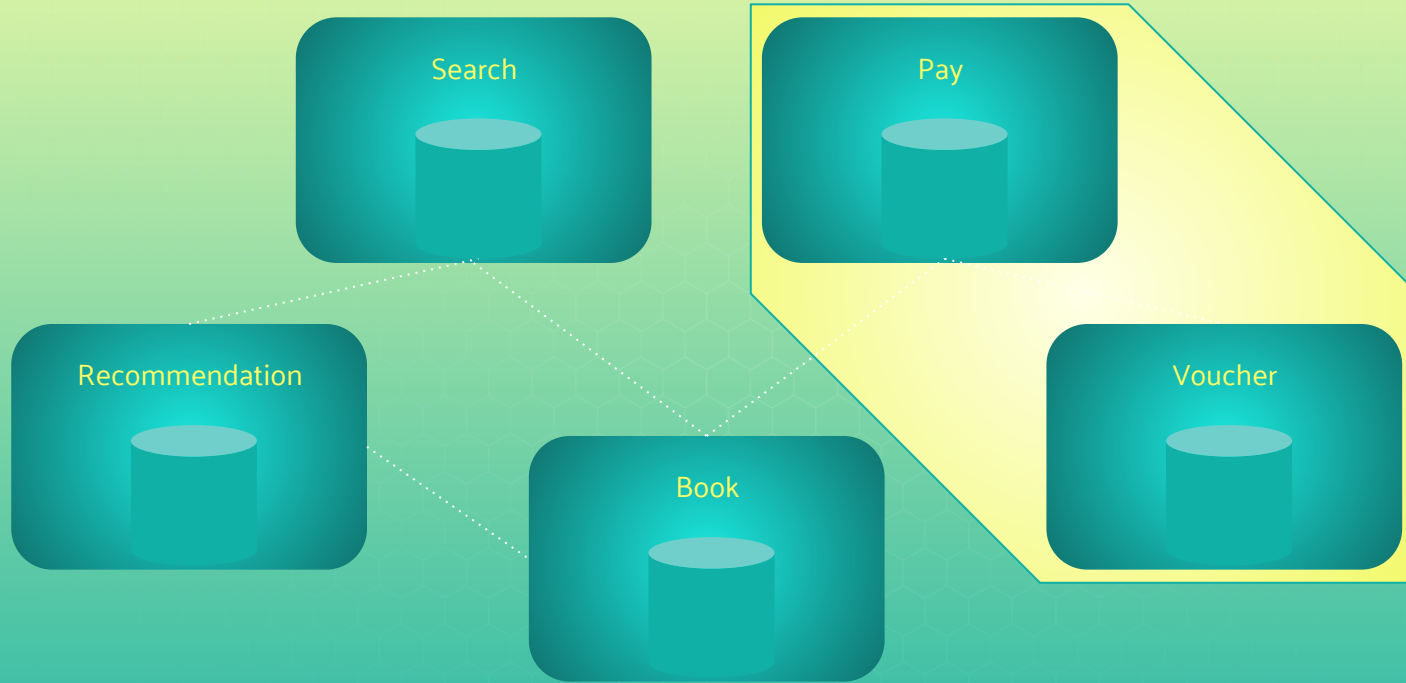
Domains – Bounded Contexts



Domains – Bounded Contexts



Domains – Bounded Contexts



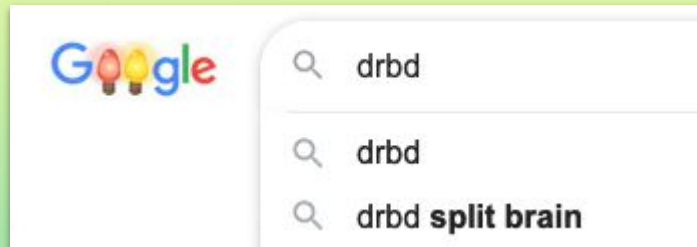


Distributed Systems

**Your Consensus is a
House of Cards**

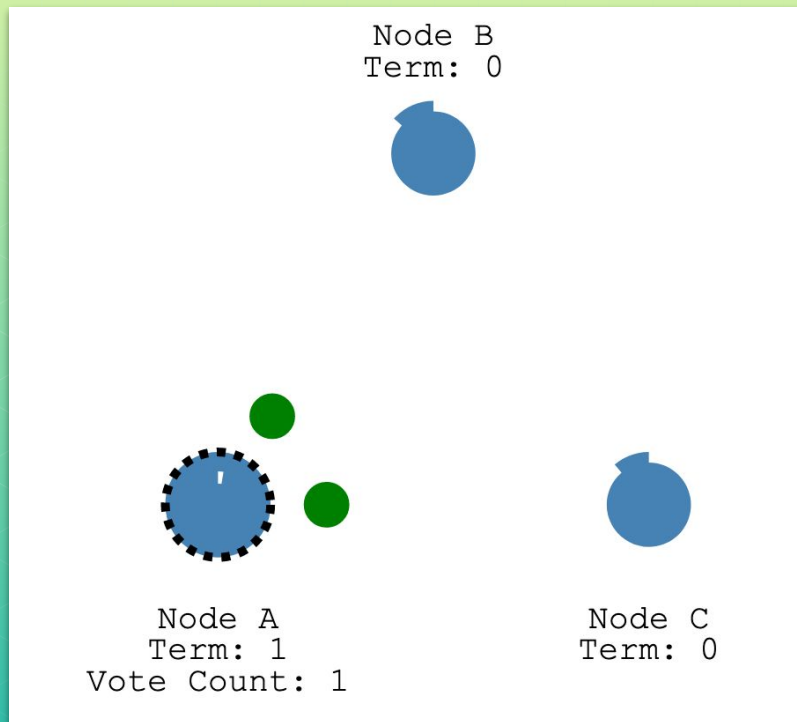
Consensus Systems are Great

- HA/Clustering prior to consensus systems
 - Heartbeats with serial cable
 - DRBD/GFS
 - STONITH Hardware
- Complex HA machinery was often the cause of outages



Safe Coordination in Distributed systems

- Systems need to agree on a single truth
- Consensus Protocols
- L. Lamport: *The Part-Time Parliament*, 1998
- Simple example: Raft (consul, etcd)



However: Murphy's Law

“Anything that can go wrong will [eventually] go wrong”

We take a lot of things for granted + there are unknown unknowns.

Scenario I: DockerHub

- Recently introduced rate limits
 - Urgent rollback, 3am
 - Node cannot pull *redis:latest* 🦊
- DNS Load Balancing
- DNS transport is UDP
- UDP Packages are limited in size
- Per Spec DNS allows ≤ 512 bytes

```
kulkema — -bash — 117x27
[kulkema@mbp ~]$ dig @1.1.1.1 registry.hub.docker.com

; <<>> DiG 9.10.6 <<>> @1.1.1.1 registry.hub.docker.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59050
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;registry.hub.docker.com.      IN      A

;; ANSWER SECTION:
registry.hub.docker.com. 276     IN      CNAME   elb-hub.us-east-1.aws.dc
elb-hub.us-east-1.aws.dcr.io. 876    IN      CNAME   us-east-1-elbhub-1t5fblb
us-east-1-elbhub-1t5fblb53f6sl-411513349.us-east-1.elb.amazonaws.com. 36
us-east-1-elbhub-1t5fblb53f6sl-411513349.us-east-1.elb.amazonaws.com. 36
us-east-1-elbhub-1t5fblb53f6sl-411513349.us-east-1.elb.amazonaws.com. 36

;; Query time: 21 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Mon Dec 07 10:40:29 CET 2020
;; MSG SIZE rcvd: 222

[kulkema@mbp ~]$
```

Scenario I: DockerHub, cont.

- DNS responses > 512 bytes fall back to TCP
 - Your sysadmin might not know this
 - Security Group blocks tcp/53
- Not all resolvers are alike / agree on the spec
 - Glibc “salvages” truncated DNS messages
 - Golang DNS resolver (Docker) does not
 - Quick fix: `CGO_ENABLED=1`

Scenario 2: DNS, again (it's always DNS)

- Our J2EE service is stuck in an exception loop
 - Logs a lot of large stack traces (lots of lines)
- Engineers integrate cool .io SaaS for tailing logs in Logstash
 - Every line a request to cool .io data sink
 - Every line a hostname is resolved
- Cloud Providers disapproves, starts rate-limiting DNS for the service's node
- K8S api-server/node comm. is affected.
 - Node is marked as broken
 - Scheduler moved ever-crashing service to fresh, healthy node
- Repeat

Scenario 3: Seemingly unlimited resources

- Nov 2020 Kinesis outage
 - every node connects with every other node
 - After scaling exceeded threads-max
- File Handles
 - Some workloads do not properly close TCP/IP connections
 - Intermediate proxies have to arbitrarily terminate
 - (Old) user-land kube-proxy leaked goroutines & file handles





Observability

How to X-Ray a hairball

```
var bs.alert.noConflict=function(){return  
alert.constructor.d.a.fn.alert.noConflict=function(){jQuery},  
"bs.button",f="object"==typeof b&&b.d.data("bs.button"),  
this.d={this.$element=a(b),this.options=a.extend({},this.defaults,  
"loading..."),c.prototype.setState=function(b){var d="resetText",  
data()&b="Text",null=f.resetText&d.data("resetText"),  
function(){loadingText"==b?(this.isLoading=  
this.isLoading=!1,d.removeClass(c).removeAttr(c)),this.  
"bs.button"=1;if(b.length){var c=this.  
prop("checked")&&this.$element.hasClass("active"),  
c.prop("checked"),!this.$element.hasClass("active"),  
a.fn.button,a.fn.button=b,a.fn.button.Constructor.c,a.fn.button.  
"click.bs.button.data-api",[data-toggle="button"]},function(a){  
c.preventDefault()}})(jQuery),+function(a){"use strict";  
c.DEFAULTS,d.data(),"object"==typeof b&&b.g="string"=  
typeof b?e.to(b):g?e(g):f.interval,e.pause(),  
this.keydown,this),this.$indicators=this.$element,  
this.sliding=this.interval=this.$active=this.$item,  
this.pause,this)).on("mouseleave.bs.carousel",a.proxy(this.  
prototype.keydown,function(a){switch(a.which){case 37:  
c.preventDefault(),c.prototype.cycle=function(b){return b  
this.interval,this.options.interval&&  
this.next,this),this.options.interval)),this.  
return b-this.$items.length-1||0?void 0:this.sliding?  
this.pause()&b?this.pause().cycle():this.slide(b,"next",  
this.cycle(!0)),this.interval=clearInterval(this.interval),  
this.$element.find(".next, .prev"),  
this.$element.find(".item.active"),  
b?first:"last",i=this;if(a.length){if  
this.sliding=!1;var j=0,k=a[j],k.isDefaultPrevented()  
active";var l=a(this.$indicators()){if(this.sliding  
direction:g);return a.support.transition?  
a.addClass(g).d.one("transition",function(){a.removeClass(g)
```

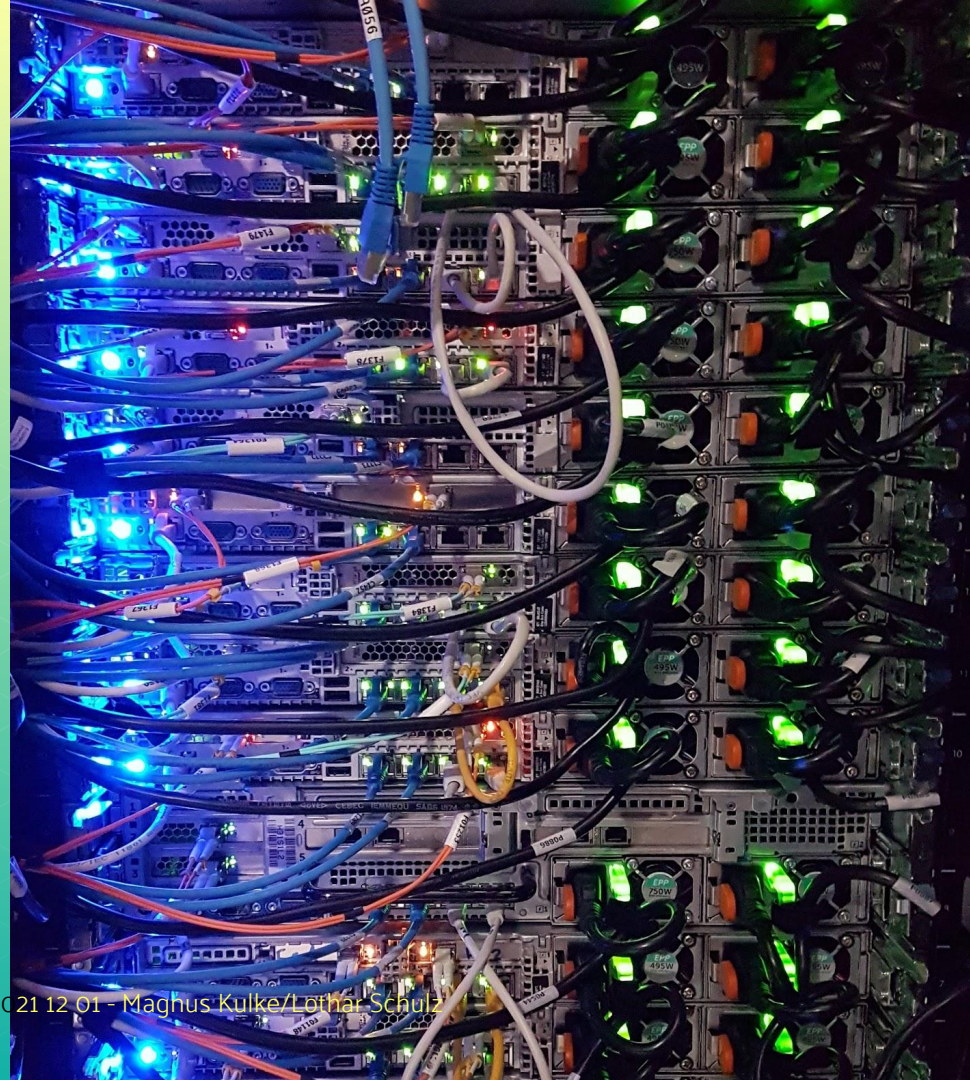
WELCOME TO THE ISILON CLUSTER SUMMARY DASHBOARD

Total Nodes ↗ 3	Nodes Down ↗ 0	Alert Status ↗ Attention	Cluster CPU ↗  14.80%	Cluster Capacity ↗  97%	NFSv3 Throughput ↗ 34 Bps	NFSv3 Op/s ↗ 1 ops	NFSv3 Latency ↗ 0.07 ms	SMB2 Throughput ↗ N/A	SMB2 Op/s ↗ N/A	SMB2 Latency ↗ N/A
Total Nodes ↗ 9	Nodes Down ↗ 0	Alert Status ↗ Healthy	Cluster CPU ↗  14.40%	Cluster Capacity ↗  30%	NFSv3 Throughput ↗ 891 Bps	NFSv3 Op/s ↗ 48 ops	NFSv3 Latency ↗ 0.7 ms	SMB2 Throughput ↗ N/A	SMB2 Op/s ↗ N/A	SMB2 Latency ↗ N/A
Total Nodes ↗ 9	Nodes Down ↗ 0	Alert Status ↗ Attention	Cluster CPU ↗  8.10%	Cluster Capacity ↗  30%	NFSv3 Throughput ↗ 41.7 MBps	NFSv3 Op/s ↗ 25.7K ops	NFSv3 Latency ↗ 3 ms	SMB2 Throughput ↗ 187 kBps	SMB2 Op/s ↗ 83 ops	SMB2 Latency ↗ 1.6 ms
Total Nodes ↗ 3	Nodes Down ↗ 0	Alert Status ↗ Healthy	Cluster CPU ↗  81.6%	Cluster Capacity ↗  7%	NFSv3 Throughput ↗ 119.2 MBps	NFSv3 Op/s ↗ 110.5K ops	NFSv3 Latency ↗ 21 ms	SMB2 Throughput ↗ N/A	SMB2 Op/s ↗ N/A	SMB2 Latency ↗ N/A
Total Nodes ↗ 16	Nodes Down ↗ 0	Alert Status ↗ Healthy	Cluster CPU ↗  24.3%	Cluster Capacity ↗  69%	NFSv3 Throughput ↗ 2.076 MBps	NFSv3 Op/s ↗ 3.00K ops	NFSv3 Latency ↗ 2 ms	SMB2 Throughput ↗ 57.1 kBps	SMB2 Op/s ↗ 713 ops	SMB2 Latency ↗ 0.7 ms
Total Nodes ↗ 15	Nodes Down ↗ 0	Alert Status ↗ Attention	Cluster CPU ↗  18.00%	Cluster Capacity ↗  69%	NFSv3 Throughput ↗ 125.2 MBps	NFSv3 Op/s ↗ 33.8K ops	NFSv3 Latency ↗ 4 ms	SMB2 Throughput ↗ N/A	SMB2 Op/s ↗ N/A	SMB2 Latency ↗ N/A
Total Nodes ↗	Nodes Down ↗	Alert Status ↗	Cluster CPU ↗ 	Cluster Capacity ↗ 	NFSv3 Throughput ↗ 146.7	NFSv3 Op/s ↗ 18.0K	NFSv3 Latency ↗	SMB2 Throughput ↗	SMB2 Op/s ↗	SMB2 Latency ↗

Tailor towards audience

Example:

- 24x7
- the engineering teams
- Management
- End customers



Service Level Objectives

Intuition, experience, and an **understanding** of what engineers know about the services they serve is used to define

- service level indicators (SLIs),
- objectives (SLOs),
- and agreements (SLAs).

SRE Book - Service Level Objectives

Guidance – The Four Golden Signals

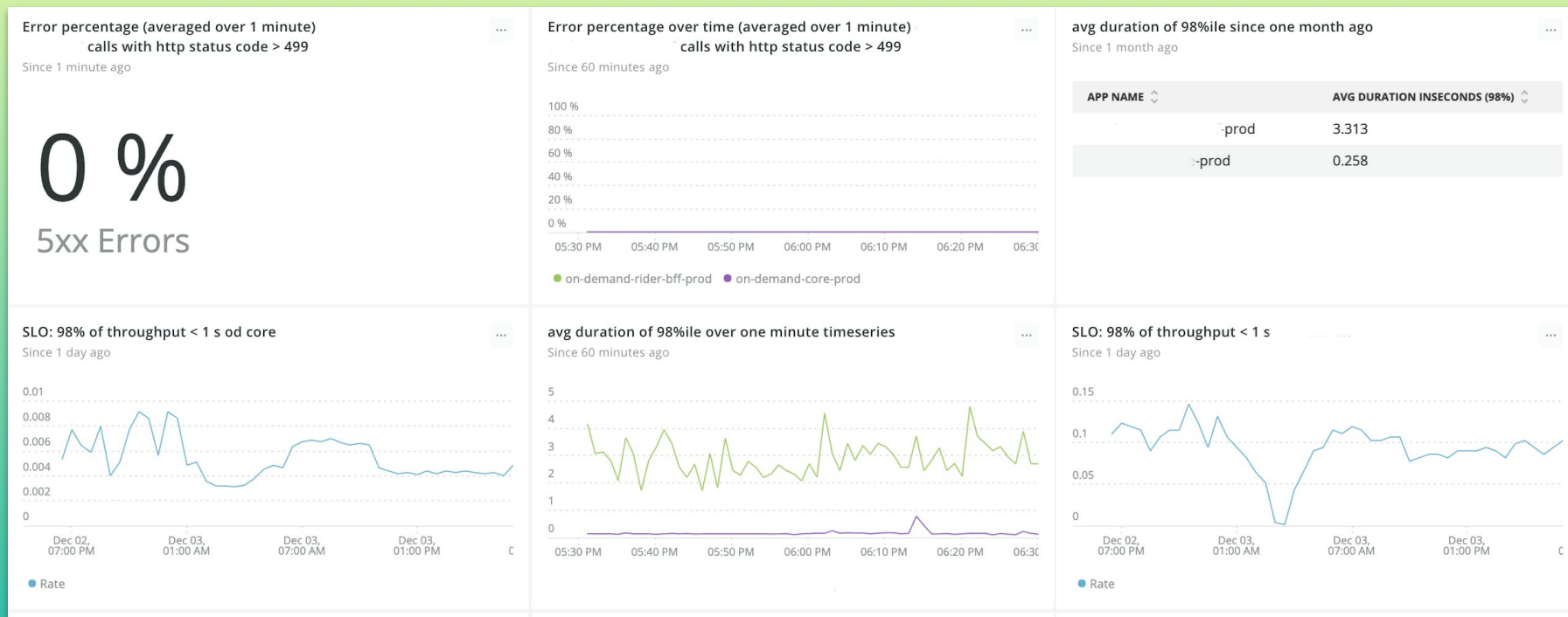
- **request latency** - request response time and/or timeout rate
- **traffic / system throughput** - demand placed on the system - http requests, static & dynamic
- **error rate** - proportion of service errors
- **saturation** - measures the system fraction, emphasizing the resources that are most constrained (e.g., in a memory-constrained system, show memory; in an I/O-constrained system, show I/O).
- **availability** - what's the uptime of a service

SRE Book - The Four Golden Signals

Results



Results





Your Questions Please

Addressing the most frequent pitfalls when transitioning to Microservices - 2021 12 01 - Magnus Kulke/Lothar Schulz