



PRINCIPLES OF MICROSERVICES

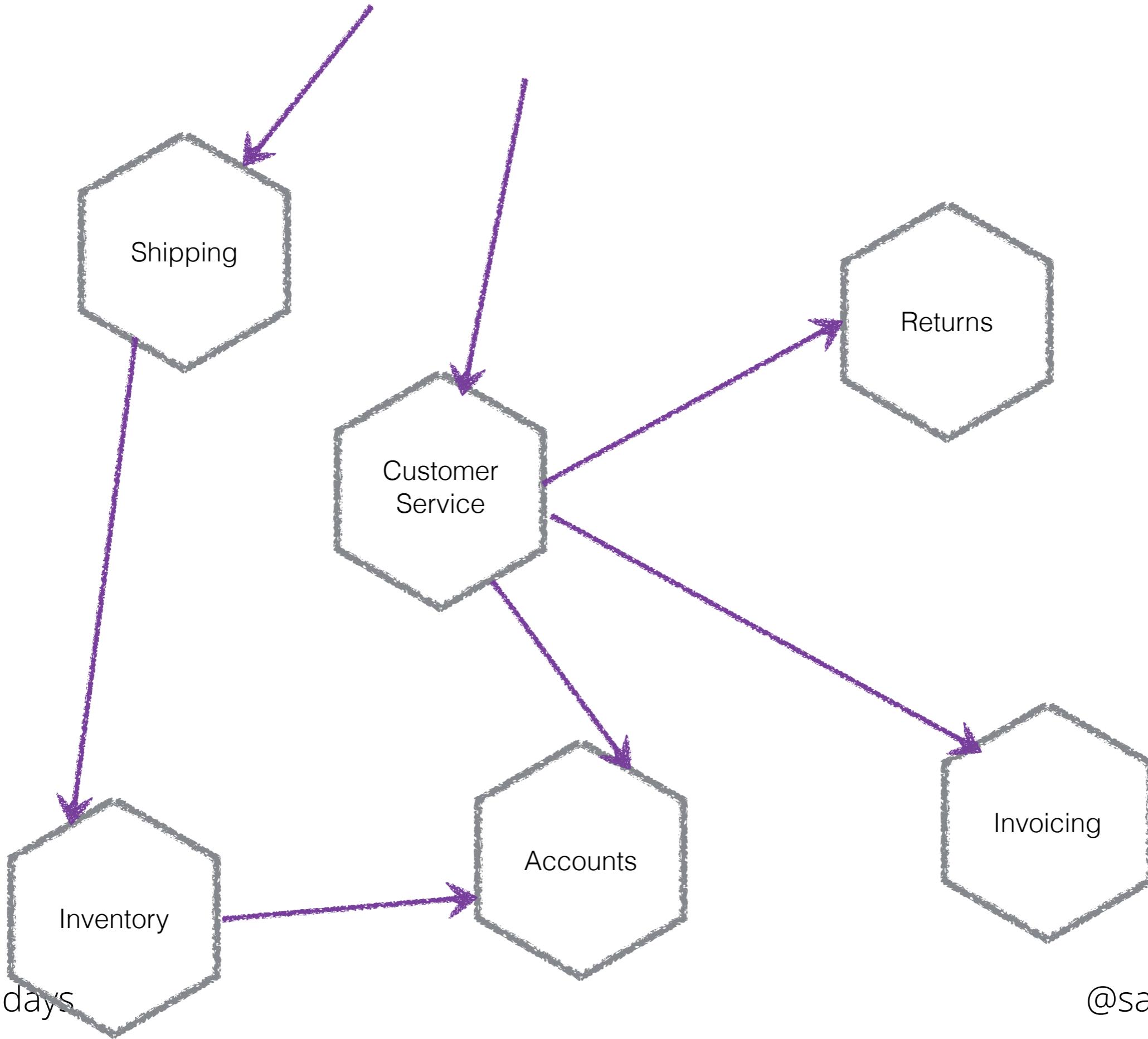
Sam Newman

XP Days Ukraine December 2014

ThoughtWorks®

#xpdays

@samnewman



#xpdays

@samnewman



Small ***Autonomous*** services
that ***work together***

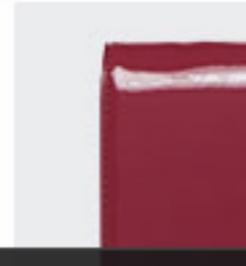
The diagram illustrates a network of six small, autonomous services represented as hexagons. Arrows show the flow of data between them. A central node labeled 'Customer Service' has arrows pointing to 'Inventory', 'Accounts', and 'Returns'. 'Inventory' has an arrow pointing to 'Accounts'. 'Shipping' has an arrow pointing to 'Customer Service'. 'Returns' has an arrow pointing to 'Customer Service'. 'Invoicing' has an arrow pointing to 'Customer Service'.

```
graph TD; CS[Customer Service] --> Inv[Inventory]; CS --> Acc[Accounts]; CS --> Ret[Returns]; Inv --> Acc; Shp[Shipping] --> CS; Ret --> CS; Inv --> CS; Inv --> Invo[Invoicing]; Invo --> CS;
```

Shipping available to United Kingdom Up to 60%

10 Perfect Gifts

Skip the guesswork with our edit of stylish surprises for all on your list

[Shop this Sale](#)

Shipping available to United Kingdom Up to 60%



10 Perfect

Skip the guesswork with our exciting surprises for all on your list.

[Shop this Sale](#)

Our purpose is to empower people by making property simple, efficient and stress-free.

ASX Share Price (REA)

[Home](#)[About REA Group](#)

Shipping available to United Kingdom Up to 60%



10 Perfect

Skip the guesswork with our easy surprises for all on your list.



Our purpose is to empower people by making property simple, efficient and

NETFLIX

Home

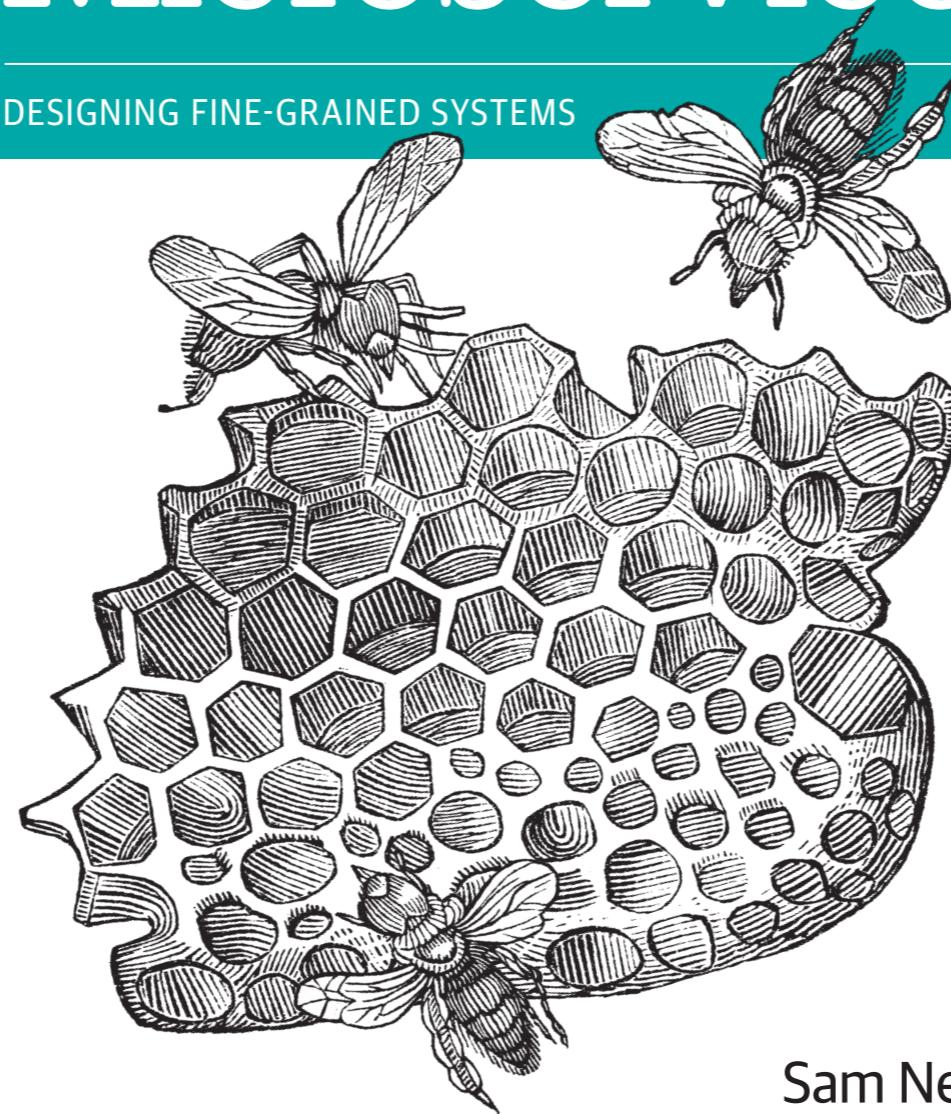
About REA Group



O'REILLY®

Building Microservices

DESIGNING FINE-GRAINED SYSTEMS



Sam Newman

#xpdays

@samnewman



THE TWELVE FACTORS

I. Codebase

One codebase tracked in revision control, many deploys

II. Dependencies

Explicitly declare and isolate dependencies

III. Config

Store config in the environment

IV. Backing Services

Treat backing services as attached resources

V. Build, release, run

Strictly separate build and run stages

VI. Processes

Execute the app as one or more stateless processes

VII. Port binding

Export services via port binding

VIII. Concurrency

Scale out via the process model

IX. Disposability

Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity

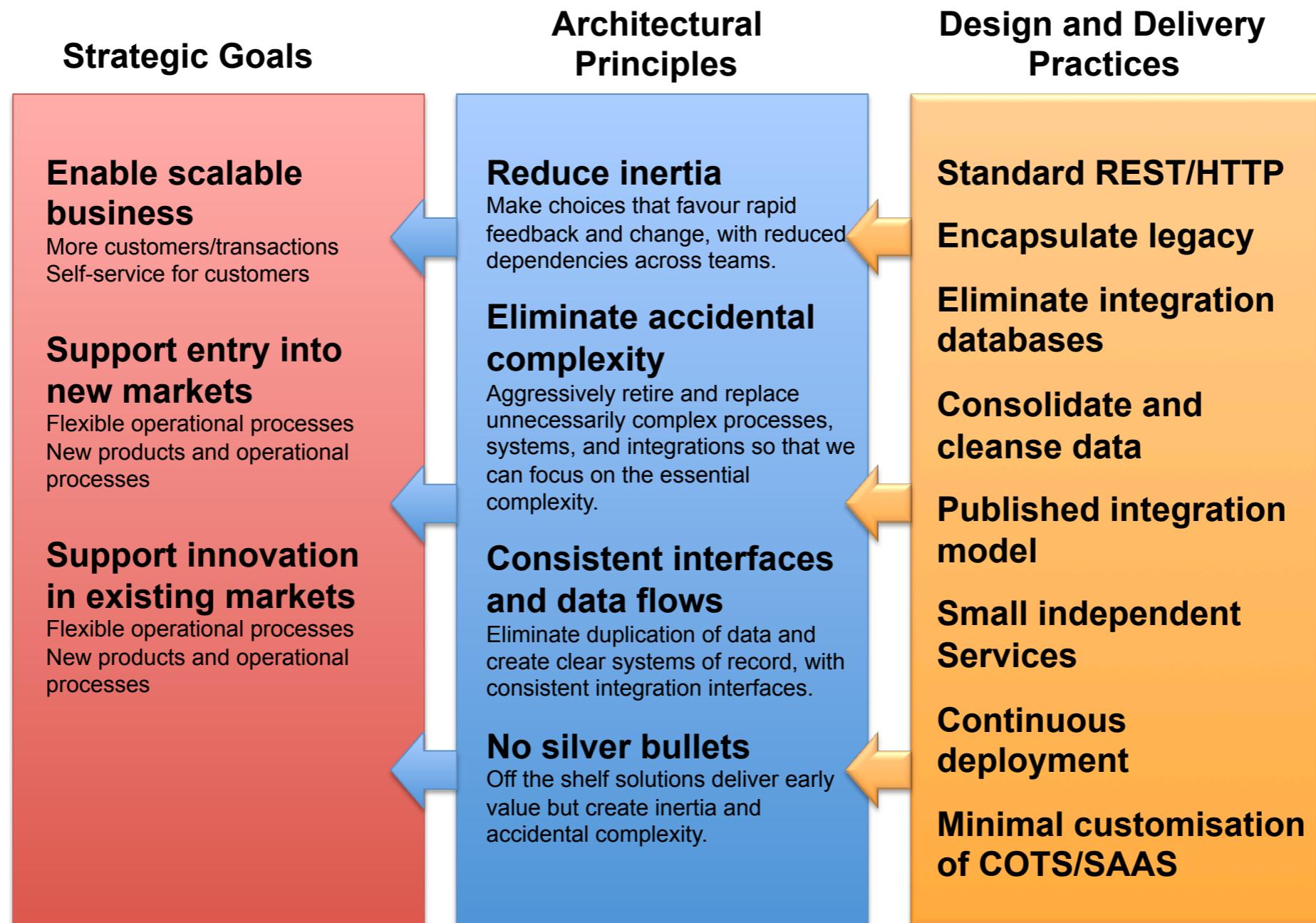
Keep development, staging, and production as similar as possible

XI. Logs

Treat logs as event streams

XII. Admin processes

Run admin/management tasks as one-off processes



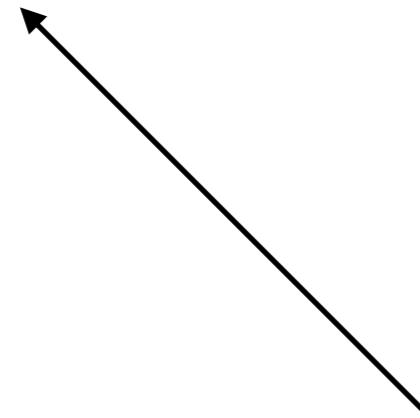
Small ***Autonomous*** services
that ***work together***

Principles Of Microservices

#xpdays

@samnewman

Modelled Around
Business Domain

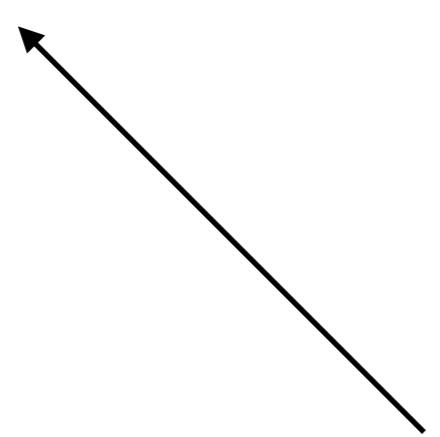


Principles Of Microservices

Modelled Around
Business Domain

Culture Of
Automation

**Principles Of
Microservices**

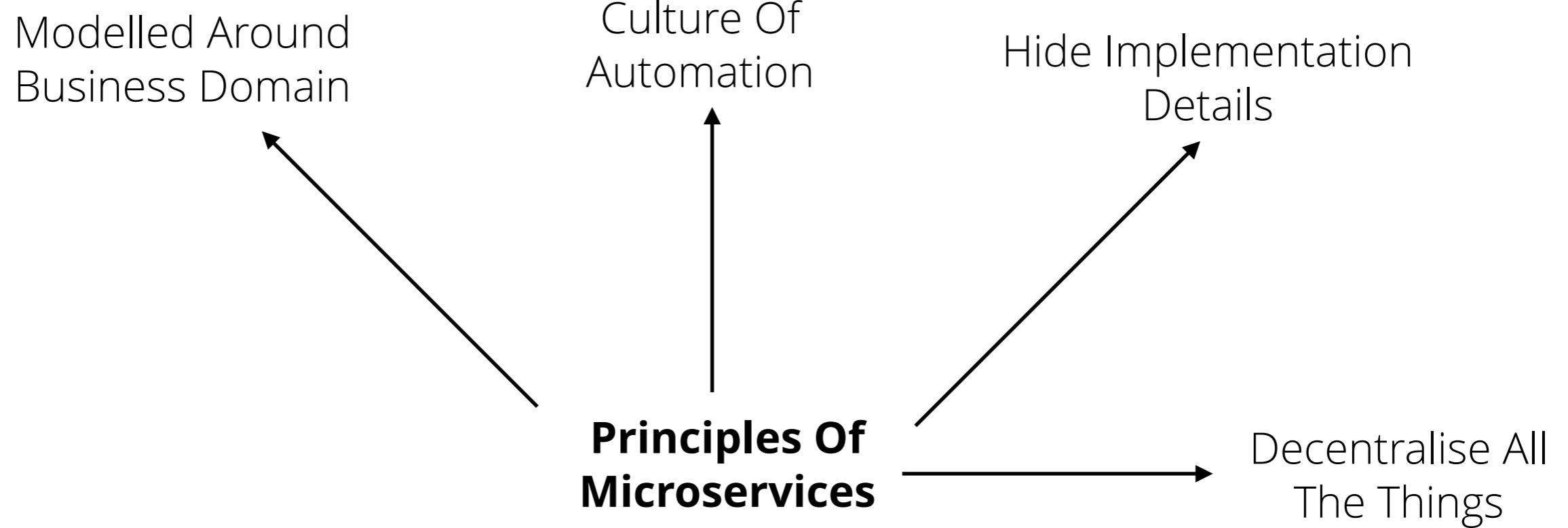


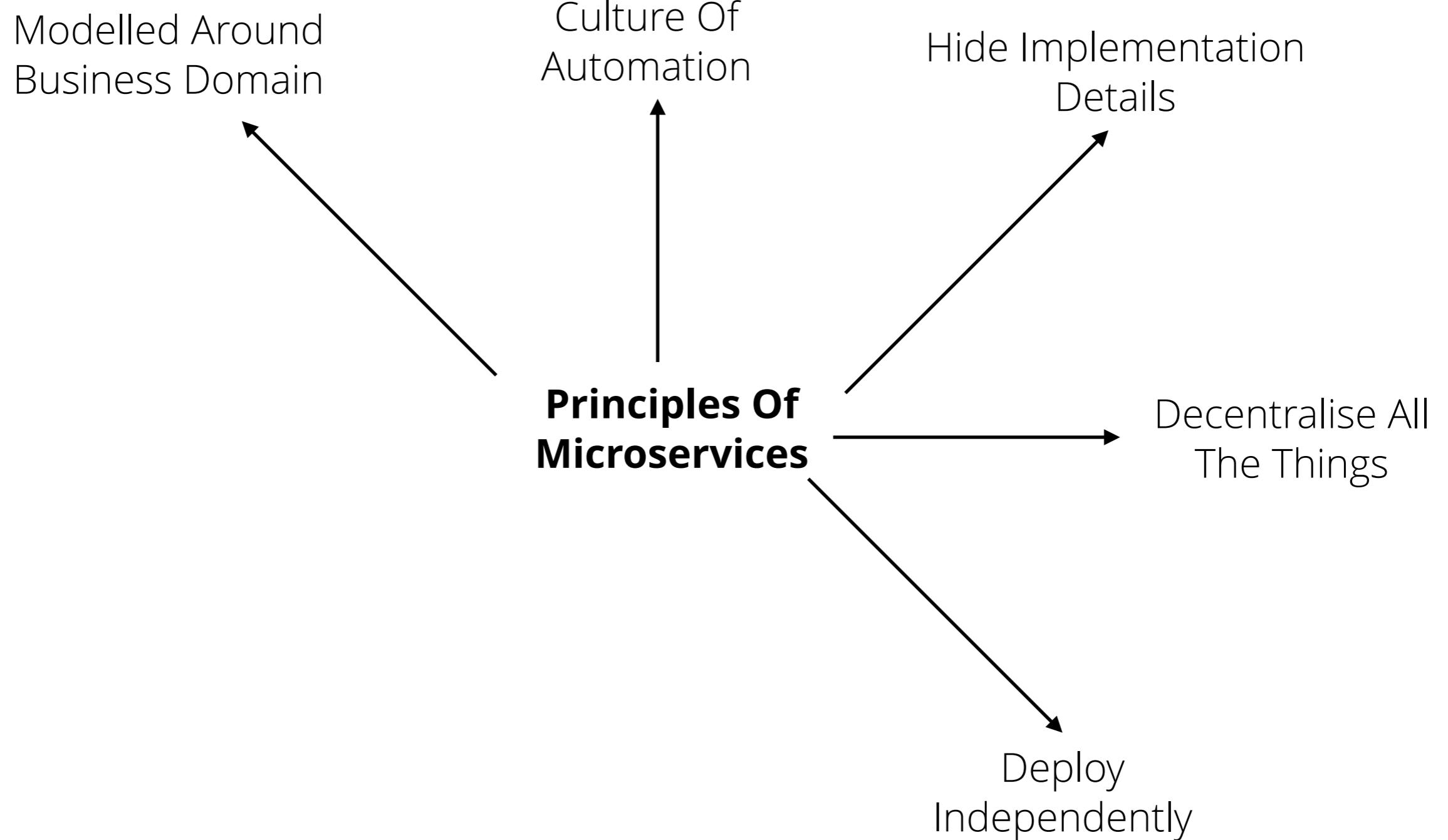
Modelled Around
Business Domain

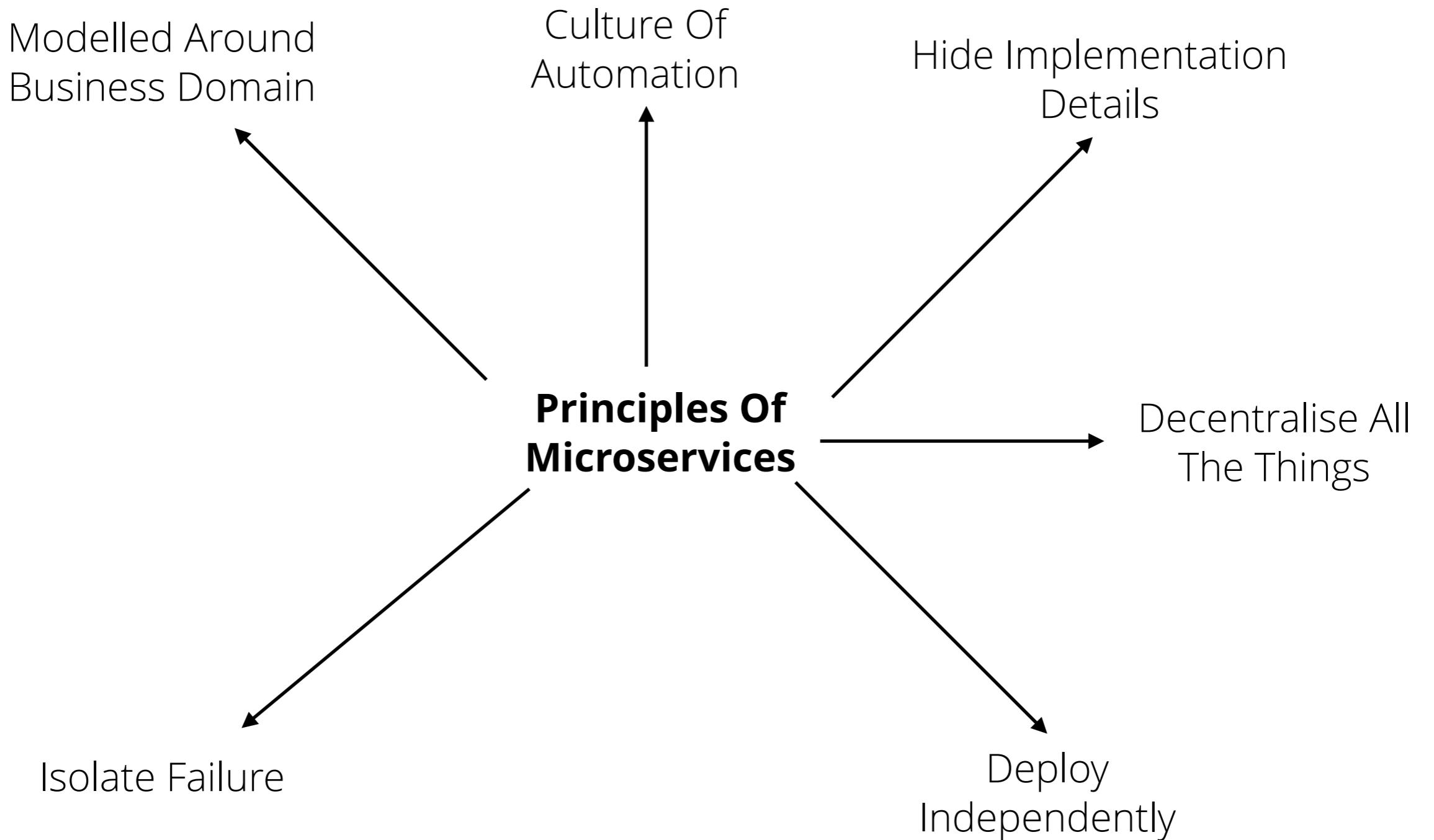
Culture Of
Automation

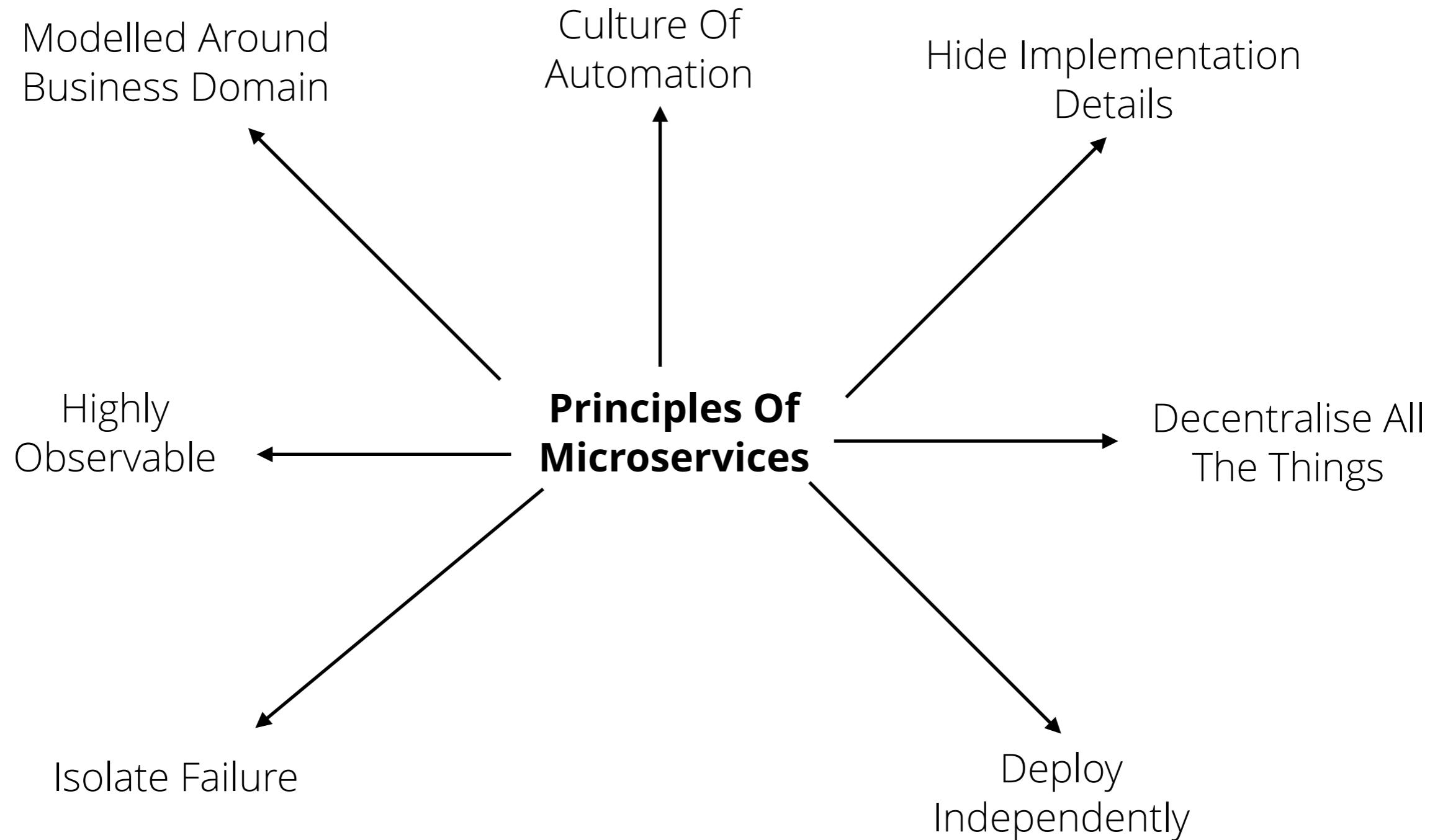
Hide Implementation
Details

Principles Of Microservices









Modelled Around
Business Domain

Culture Of
Automation

Hide Implementation
Details

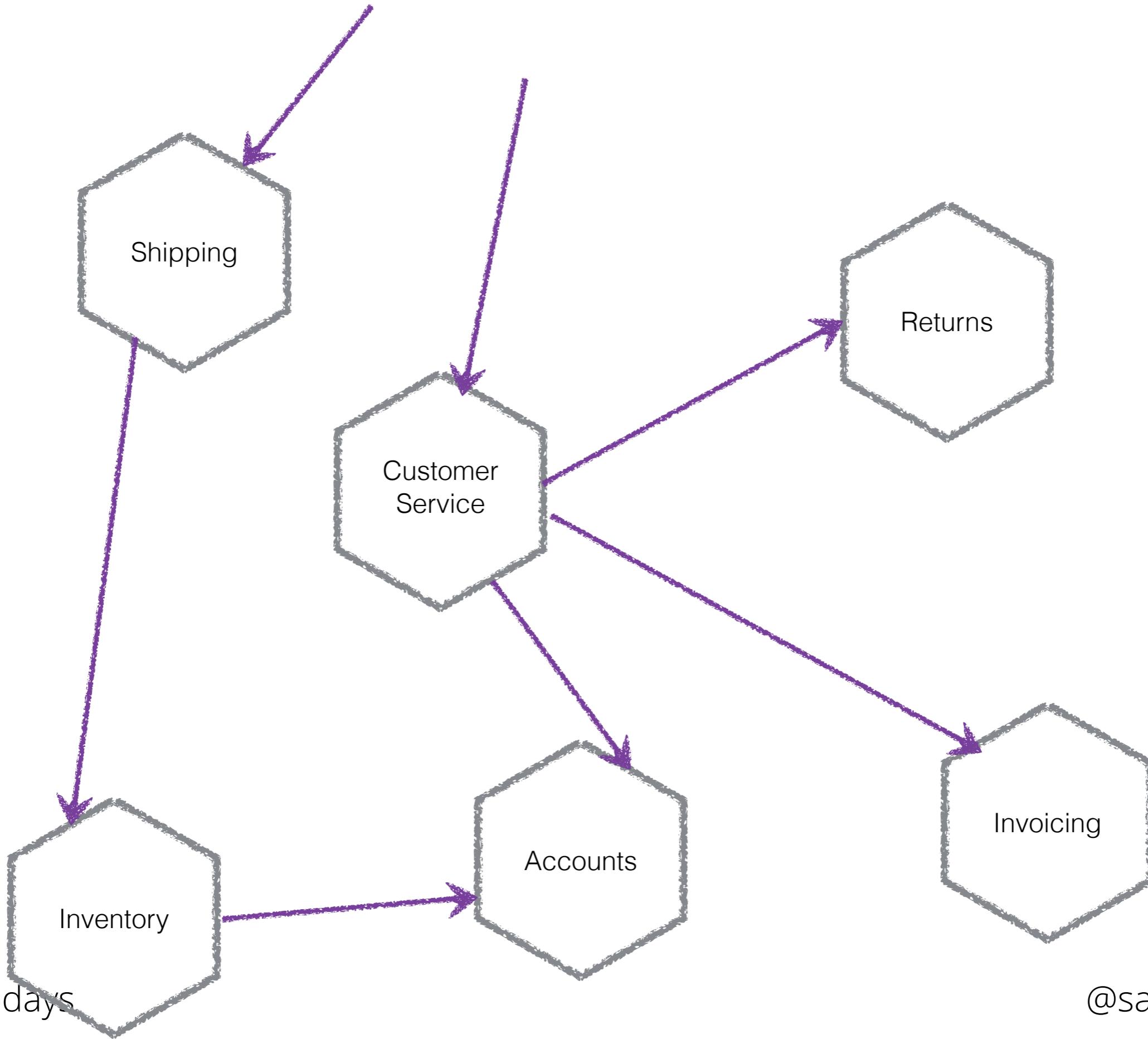
Principles Of Microservices

Highly
Observable

Decentralise All
The Things

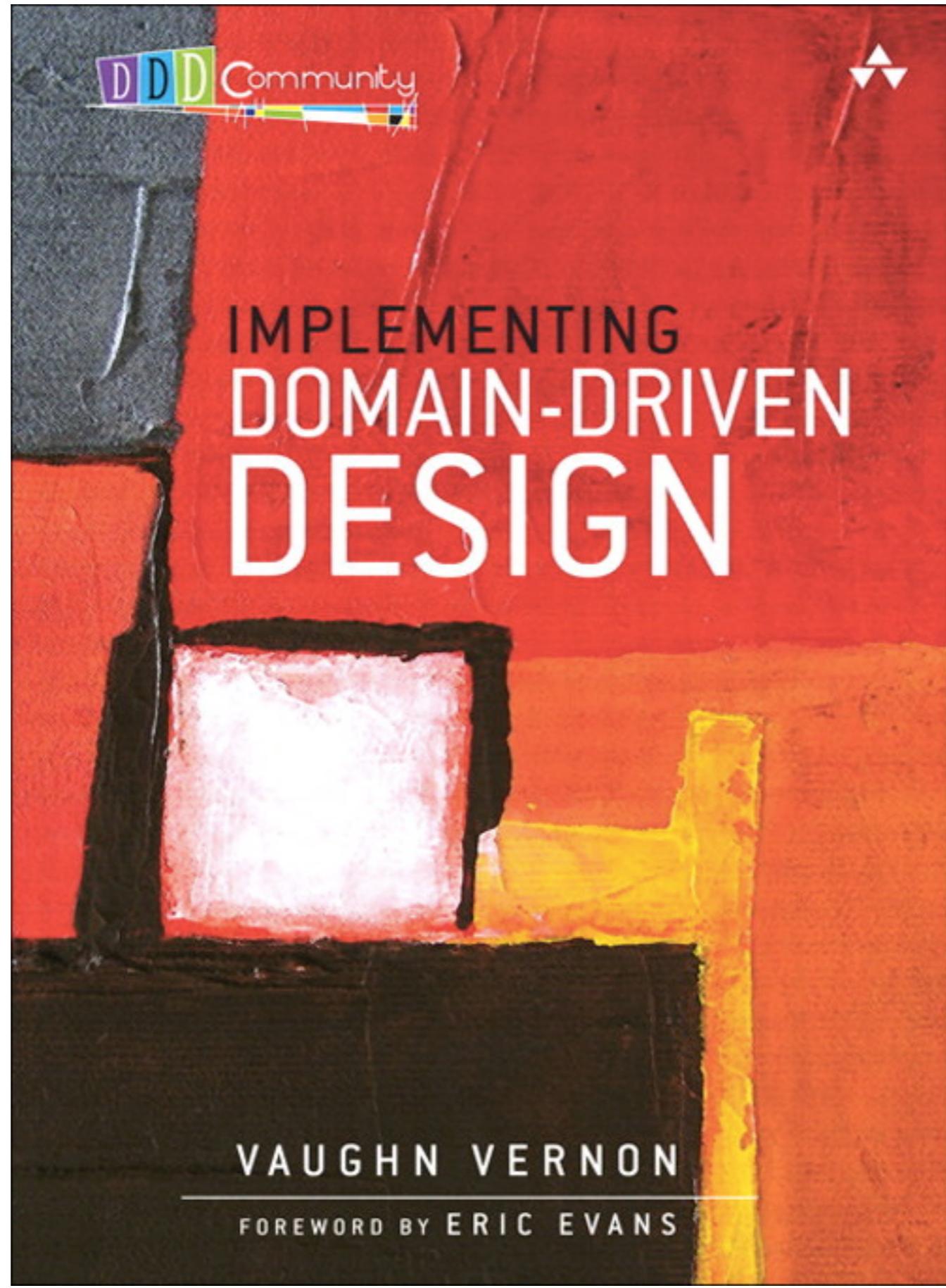
Isolate Failure

Deploy
Independently



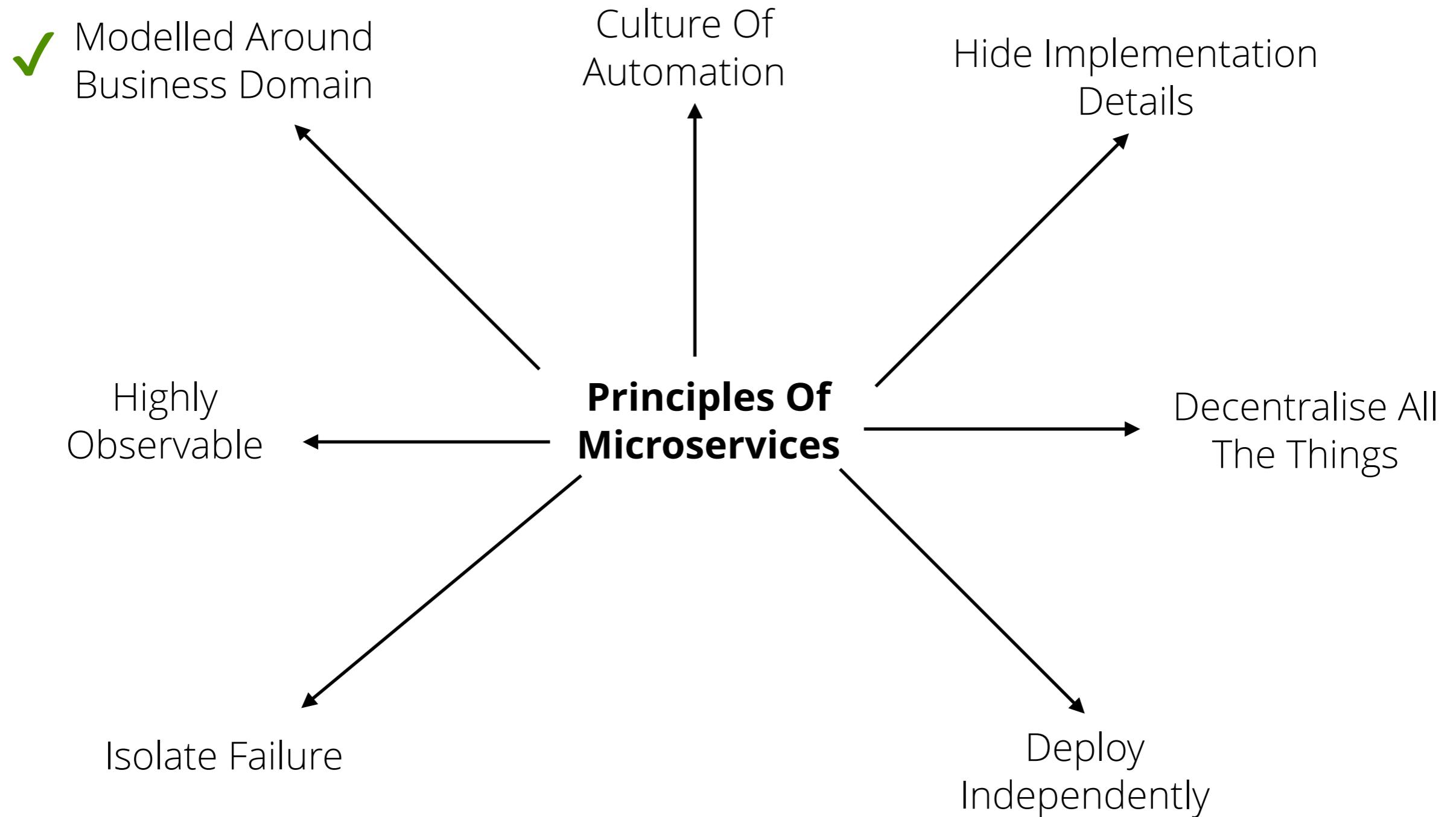
#xpdays

@samnewman



#xpdays

@samnewman





Modelled Around
Business Domain

Culture Of Automation

Principles Of Microservices

Highly
Observable

Isolate Failure

Hide Implementation
Details

Decentralise All
The Things

Deploy
Independently

#xpdays

@samnewman

2 Microservices



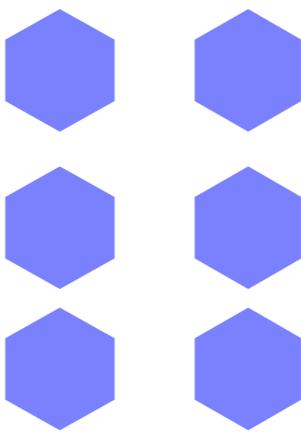
3 Months

2 Microservices



3 Months

10 Microservices



12 Months

2 Microservices



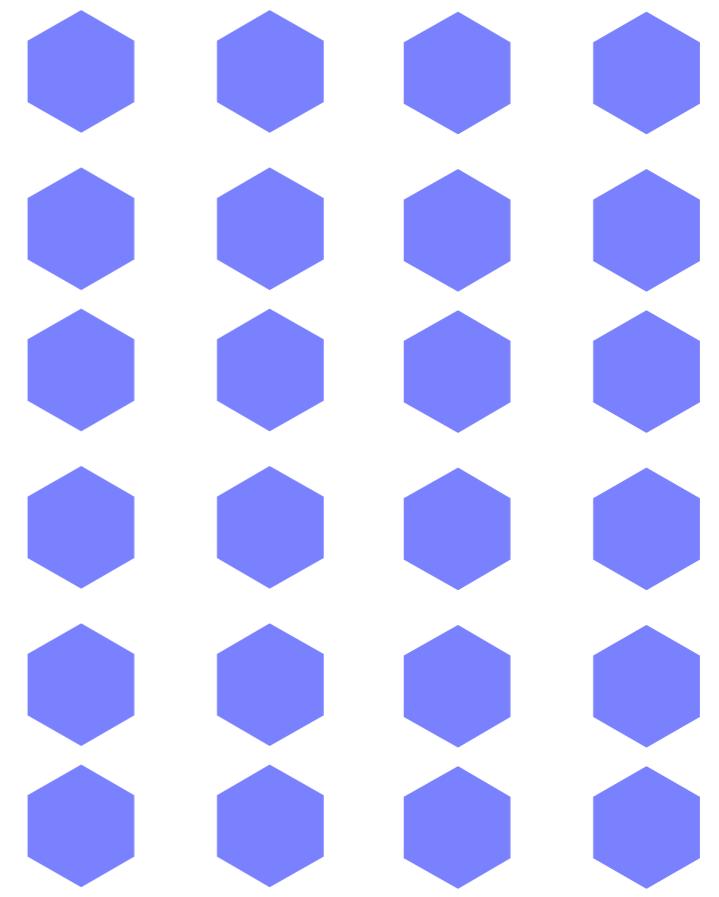
3 Months

10 Microservices



12 Months

60 Microservices



18 Months

Infrastructure Automation

#xpdays

@samnewman

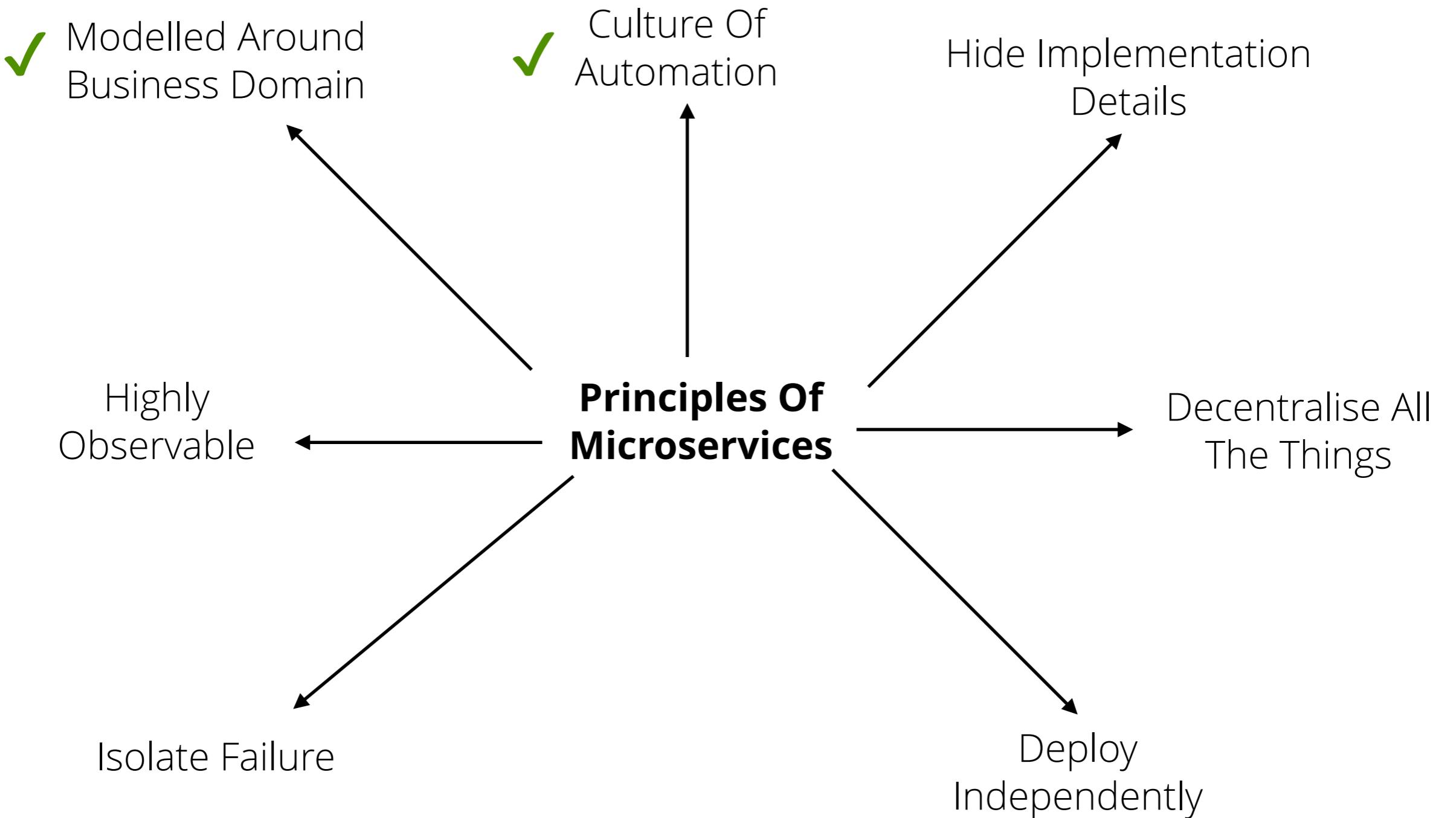
Infrastructure Automation

Automated Testing

Infrastructure Automation

Automated Testing

Continuous Delivery



✓ Modelled Around Business Domain

✓ Culture Of Automation

Hide Implementation Details

Highly Observable

Isolate Failure

Principles Of Microservices

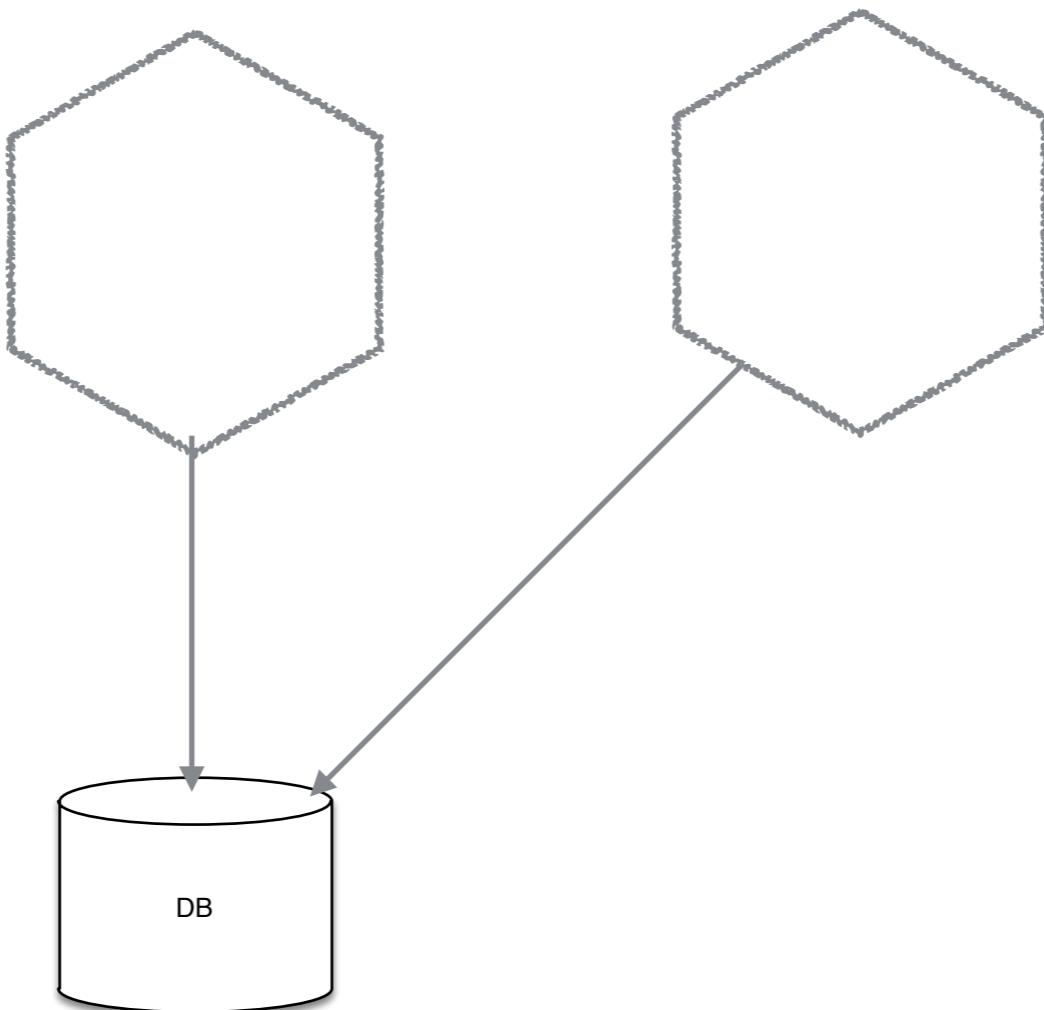
Decentralise All The Things

Deploy Independently



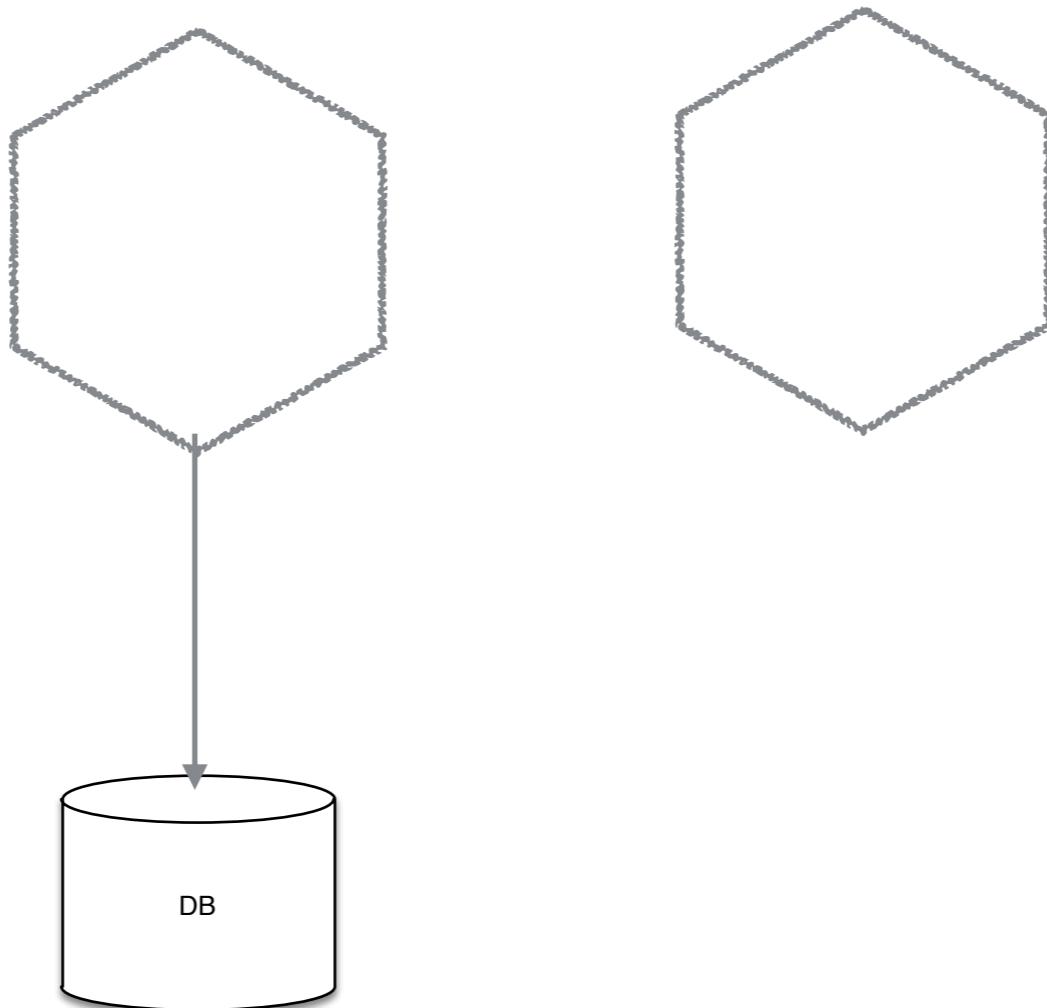
#xpdays

@samnewman



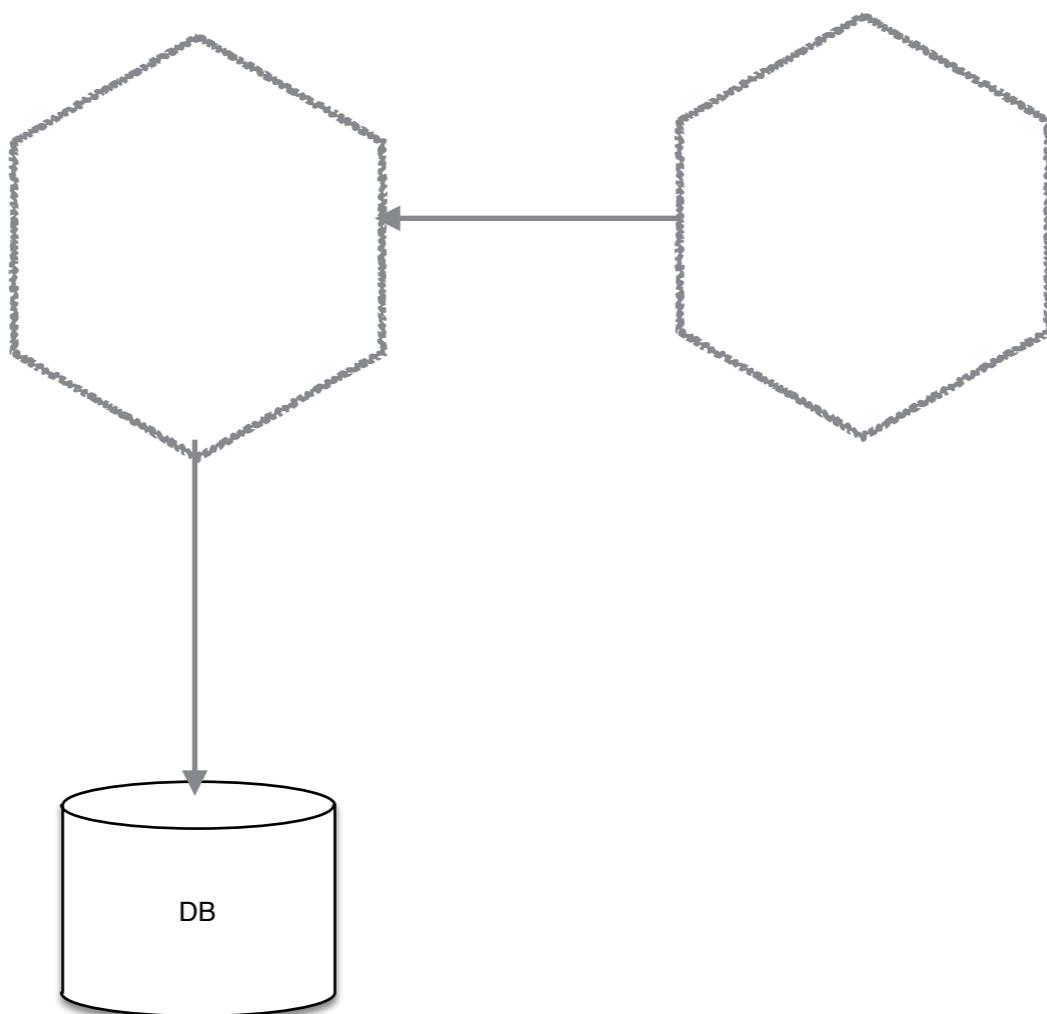
#xpdays

@samnewman



#xpdays

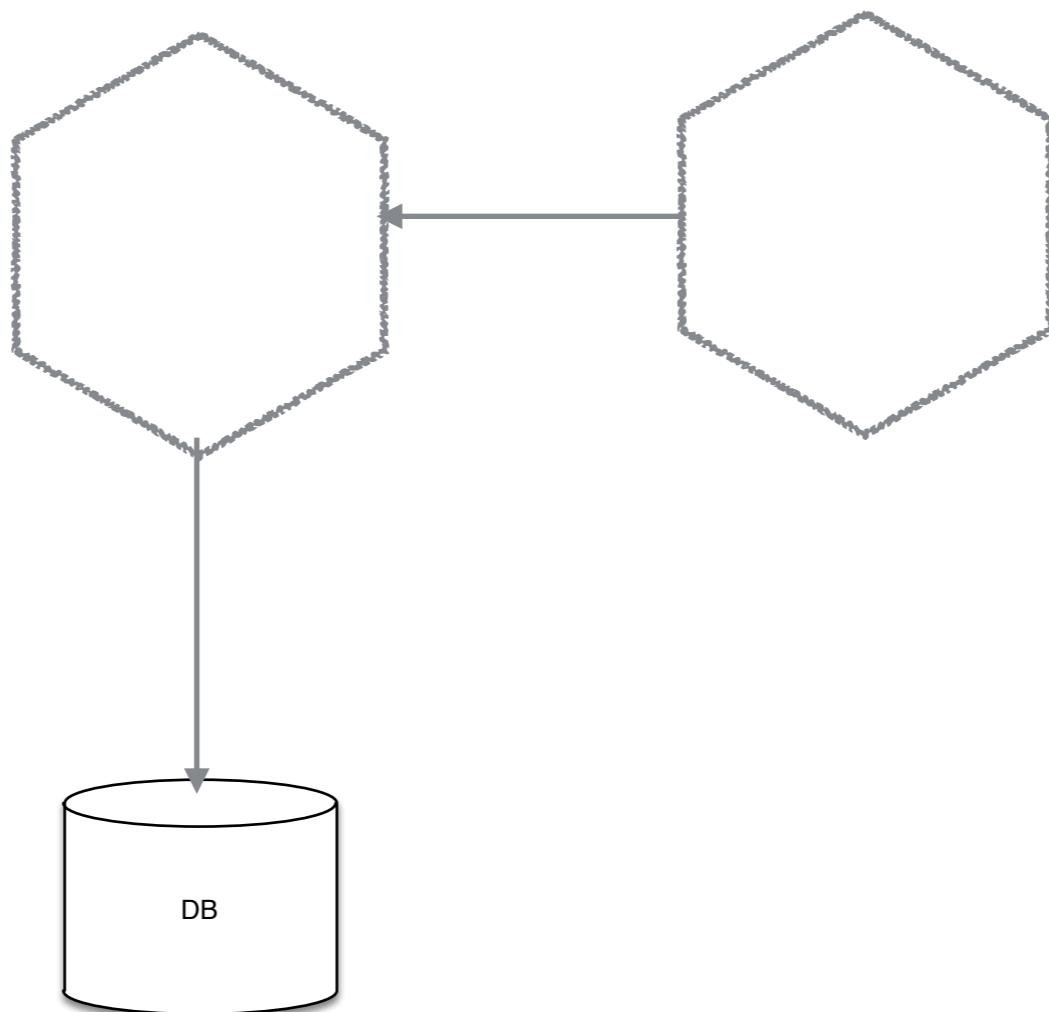
@samnewman

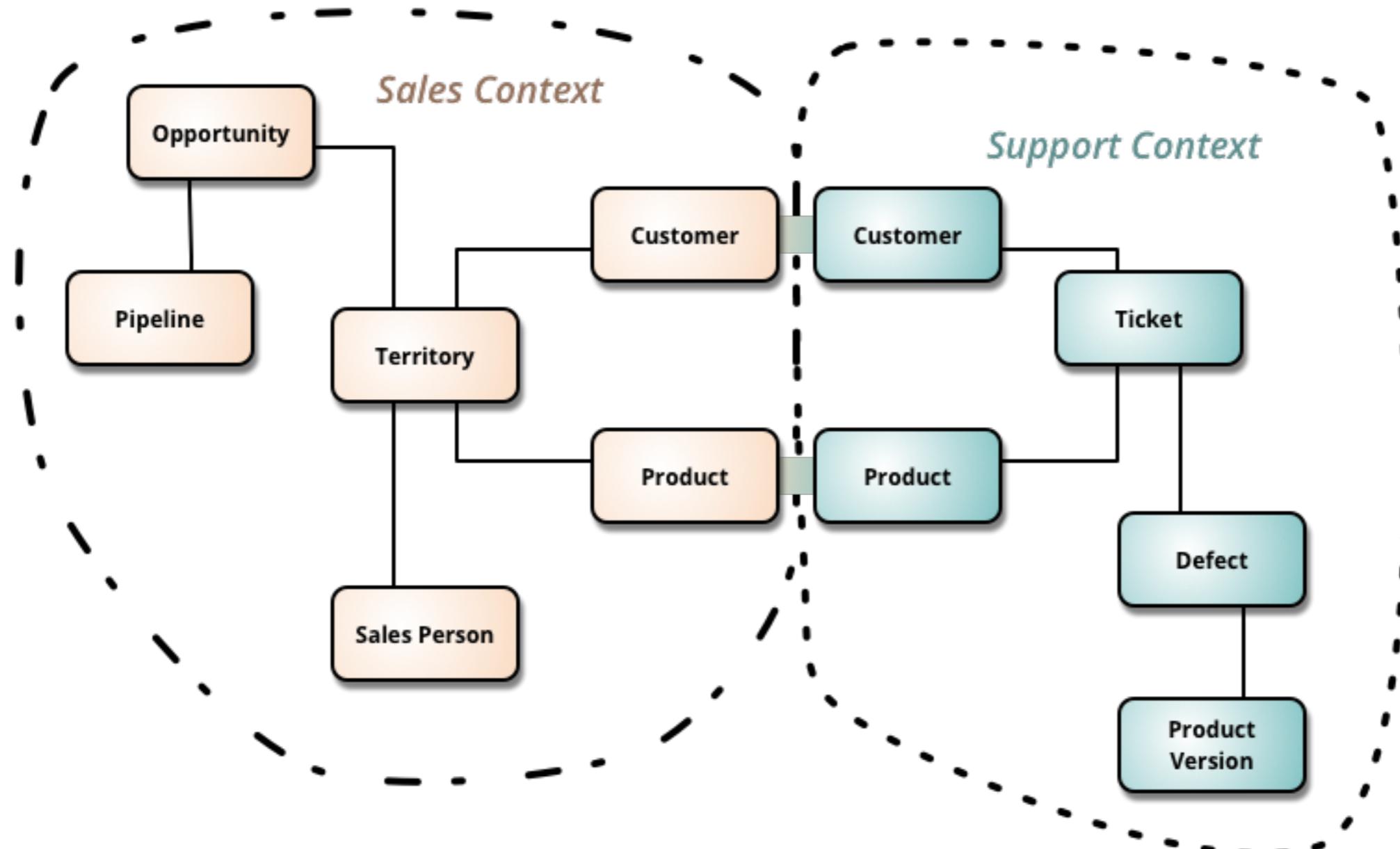


#xpdays

@samnewman

HIDE YOUR DATABASE





✓ Modelled Around Business Domain

✓ Culture Of Automation

✓ Hide Implementation Details

Highly Observable

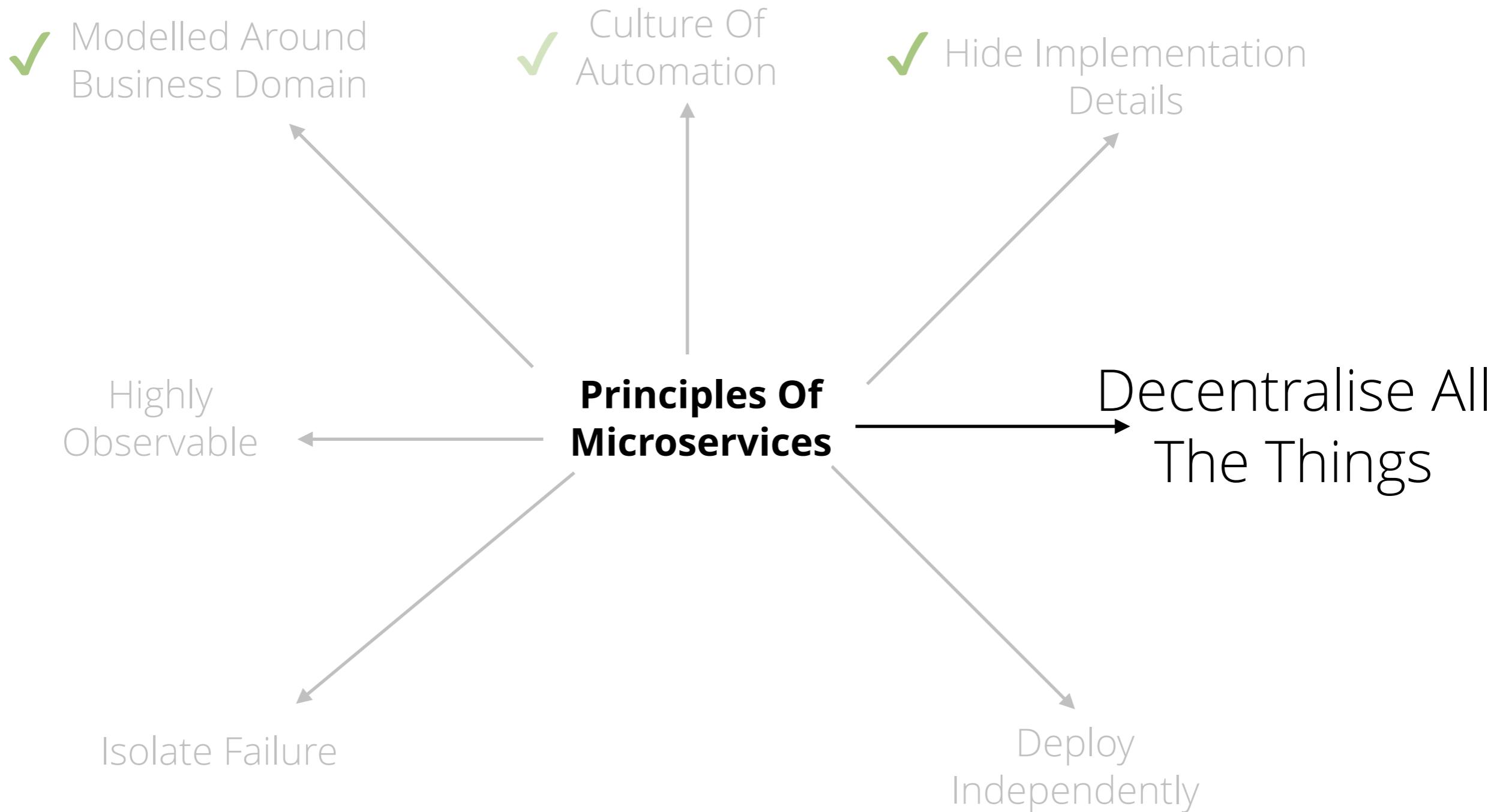
Principles Of Microservices

Decentralise All The Things

Isolate Failure

Deploy Independently

Principles Of Microservices



DECENTRALIZE



#xpdays

@samnewman

What is autonomy?

What is autonomy?

Giving people as much freedom as possible
to do the job at hand

What is autonomy?

Giving people as much freedom as possible
to do the job at hand

SELF-SERVICE



<https://www.flickr.com/photos/katsrcool/15184711908/>

GILT TECH

Search the blog

Making Architecture Work in Microservice Organizations



<http://tech.gilt.com/post/102628539834/making-architecture-work-in-microservice>
#xpdays

@samnewman

OWNER-OPERATOR

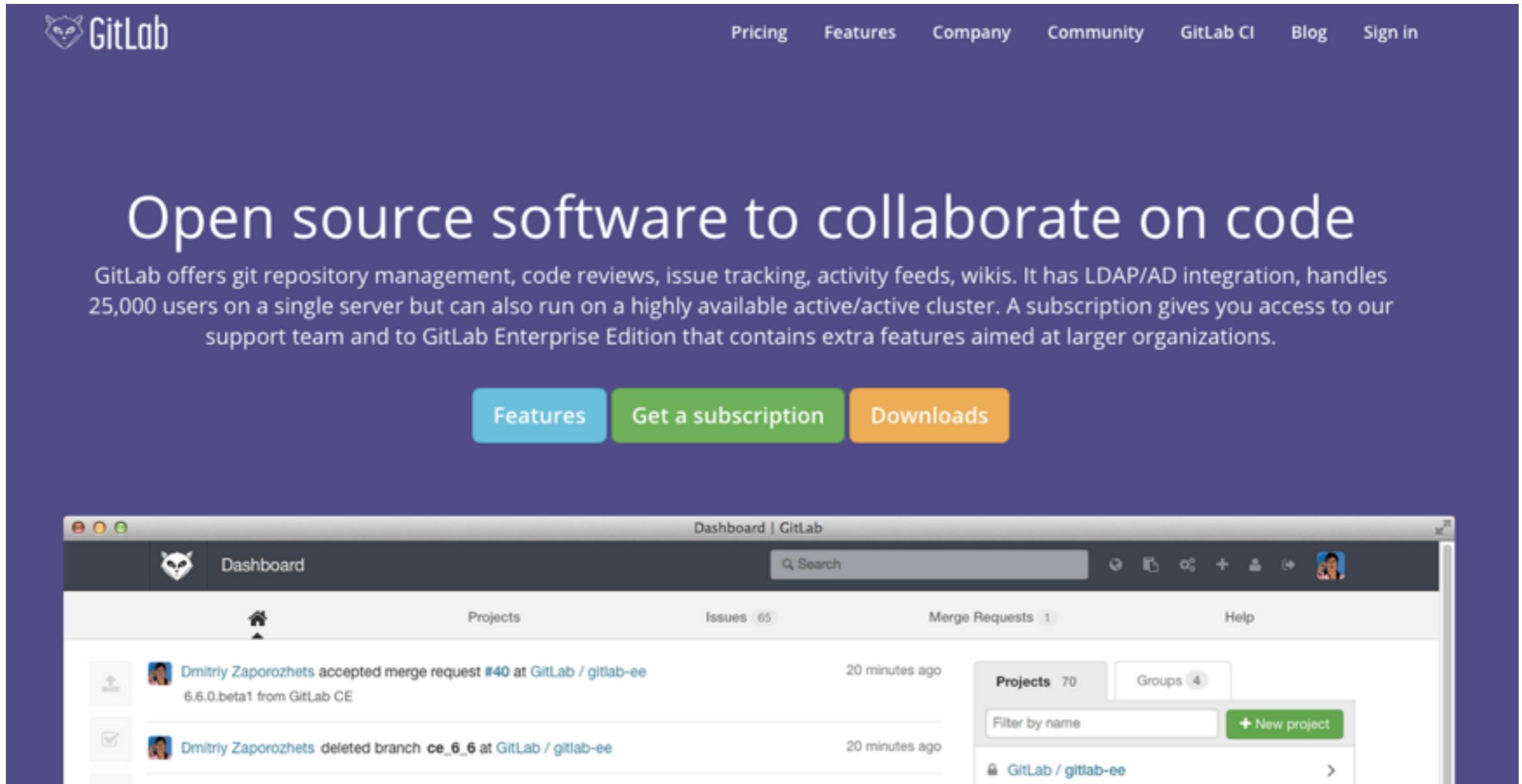
Lemonade
50¢ each



Lemonade
\$50 each

<https://www.flickr.com/photos/stevendepolo/5939055612>

INTERNAL OPEN SOURCE



The screenshot shows the GitLab homepage with a dark purple header. The header includes the GitLab logo, navigation links for Pricing, Features, Company, Community, GitLab CI, Blog, and Sign in, and a search bar. Below the header, a large white title reads "Open source software to collaborate on code". A descriptive paragraph follows, stating: "GitLab offers git repository management, code reviews, issue tracking, activity feeds, wikis. It has LDAP/AD integration, handles 25,000 users on a single server but can also run on a highly available active/active cluster. A subscription gives you access to our support team and to GitLab Enterprise Edition that contains extra features aimed at larger organizations." At the bottom of the page are three buttons: "Features" (blue), "Get a subscription" (green), and "Downloads" (orange). The main content area below the title shows a screenshot of the GitLab dashboard. The dashboard has a dark theme with a light gray header. It displays a "Dashboard" button, a search bar, and various project statistics: Projects (70), Groups (4), Issues (65), Merge Requests (1), and Help. Below these stats is a timeline of recent activity:

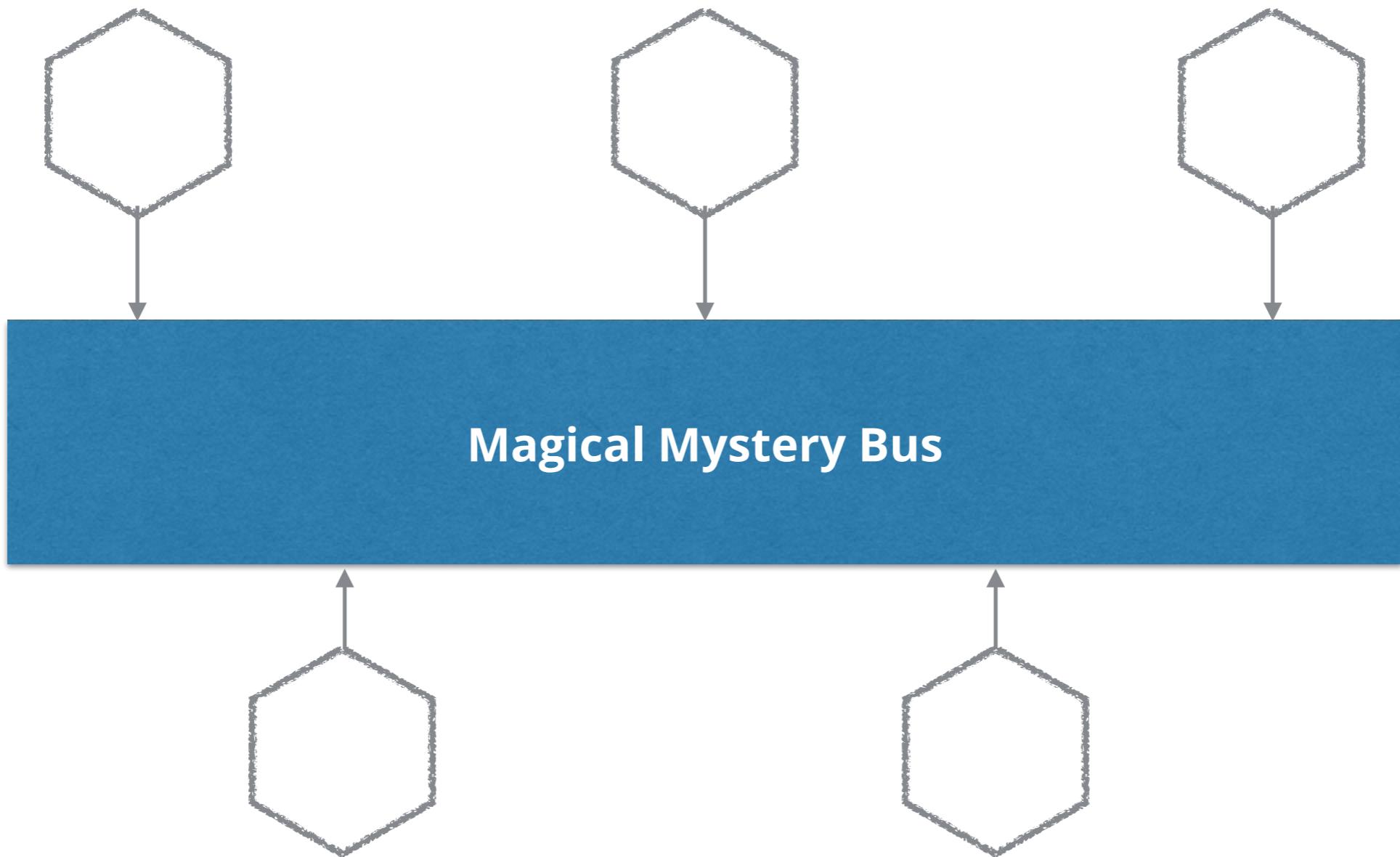
- Dmitriy Zaporozhets accepted merge request #40 at GitLab / gitlab-ee 6.6.0.beta1 from GitLab CE 20 minutes ago
- Dmitriy Zaporozhets deleted branch ce_6_6 at GitLab / gitlab-ee 20 minutes ago

A sidebar on the right shows a list of projects under "GitLab / gitlab-ee".

#xpdays

@samnewman

DUMB-PIPES, SMART ENDPOINTS



Magical Mystery Bus



✓ Modelled Around Business Domain

✓ Culture Of Automation

✓ Hide Implementation Details

Highly Observable

Principles Of Microservices

Decentralise All The Things ✓

Isolate Failure

Deploy Independently

✓ Modelled Around Business Domain

✓ Culture Of Automation

✓ Hide Implementation Details

Highly Observable

Principles Of Microservices

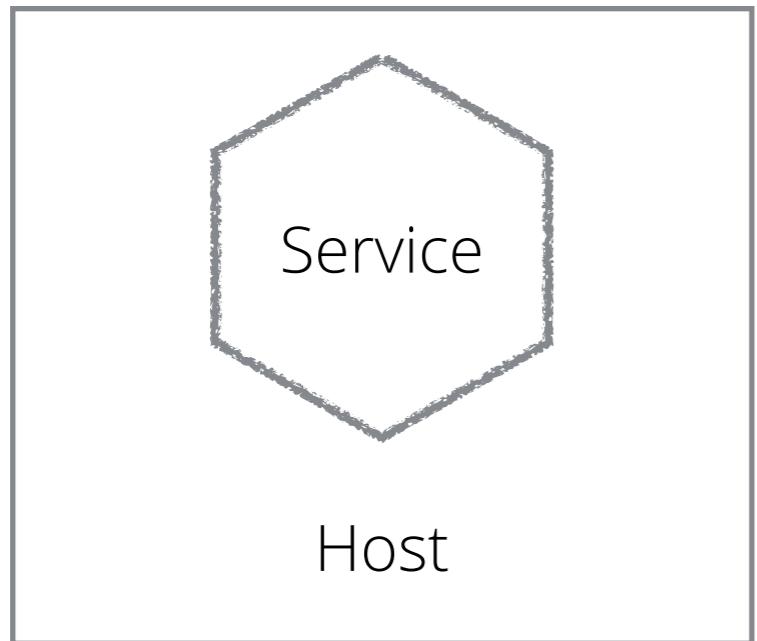
Decentralise All The Things ✓

Isolate Failure

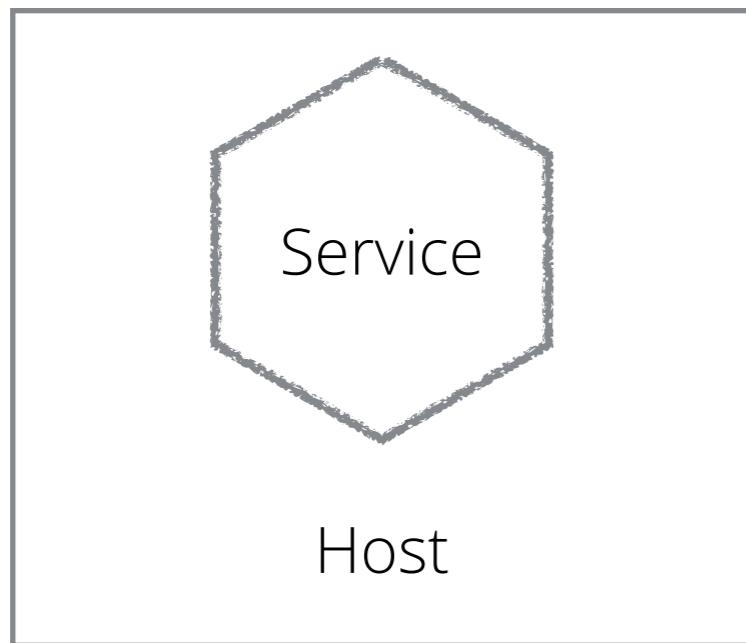
Deploy Independently

After the break!

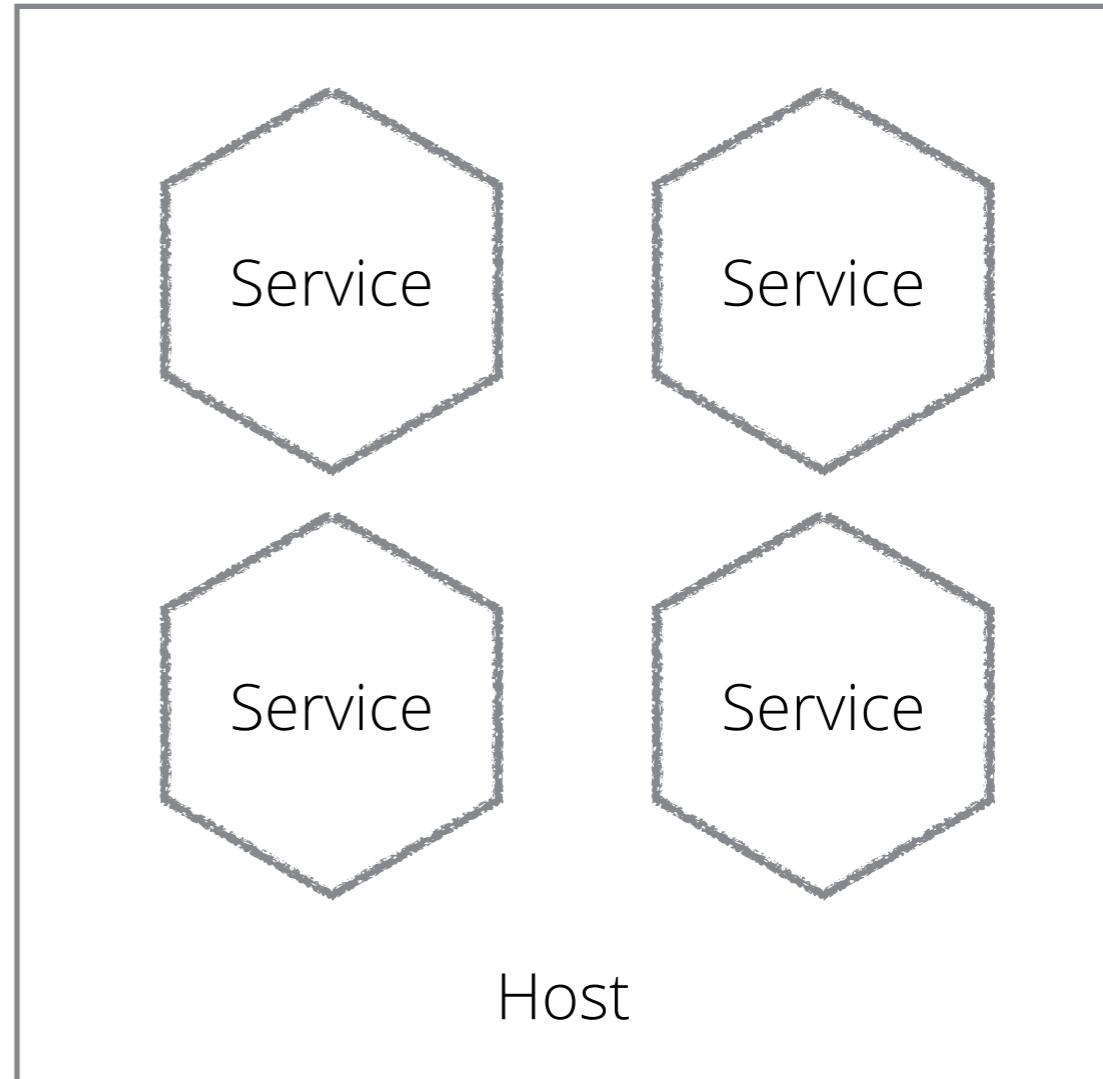
ONE SERVICE PER-HOST



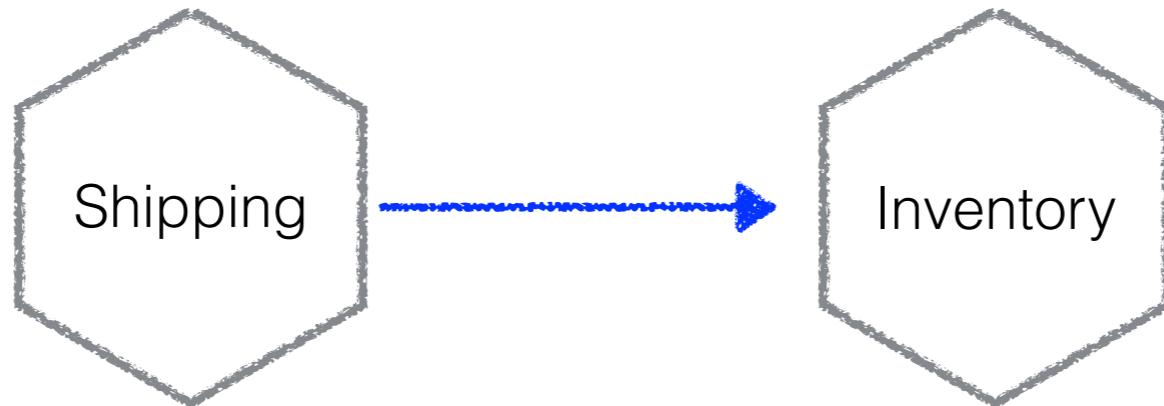
ONE SERVICE PER-HOST



VS



CONSUMER-DRIVEN CONTRACTS



Expectations



```
01 def node(): Node = {
02     ->@attr('http://www.w3.org/2000/svg#') ns1:Lang='en' Lang='en';
03     ->ns2:
04         ->link rel='stylesheet' href='/static/common.css' type='text/css'/>
05         ->link rel='stylesheet' href='/common/style.css' type='text/css'/>
06         ->meta http-equiv='refresh' content='30'/
07     </body>
08     <body>
09         &lt;content buildId=>
10     </body>
11 }
12
13 private def content(buildId: ContentBuildId): Element = {
14     displayType match {
15         case "single" => <div> build.map(build => content(build)) </div>
16         case "short" => {
17             if (build.length == 2) {
18                 <div> build.map(build => content(build)) </div>
19             } else {
20                 <ul class='bulletList'> build.map(build => bulletItem(build)) </ul>
21             }
22         }
23         case _ => <ul class='bulletList'> build.map(build => bulletItem(build)) </ul>
24     }
25 }
26
27 private def bulletItem(buildId: BuildId): Element = {
28     <li class='bullet'> buildId.prettyName.value.asStringCase >
29         <ul style='list-style-type: none; padding-left: 0;'>
30             <li> buildId.buildId </li>
31         </ul>
32     </li>
33 }
```



Pact

Define a pact between service consumers and providers, enabling "consumer driven contract" testing.

Pact provides an RSpec DSL for service consumers to define the HTTP requests they will make to a service provider and the HTTP responses they expect back. These expectations are used in the consumers specs to provide a mock service provider. The interactions are recorded, and played back in the service provider specs to ensure the service provider actually does provide the response the consumer expects.

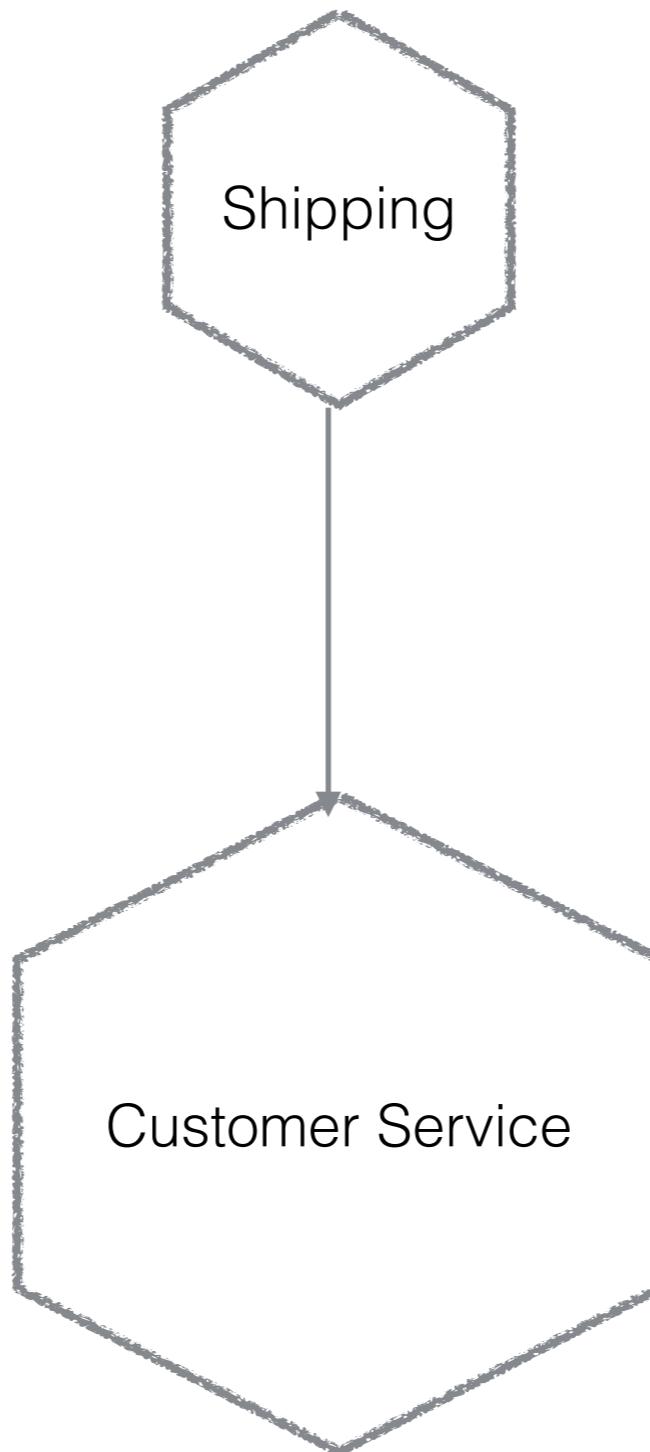
This allows testing of both sides of an integration point using fast unit tests.

This gem is inspired by the concept of "Consumer driven contracts". See
<http://martinfowler.com/articles/consumerDrivenContracts.html> for more information.

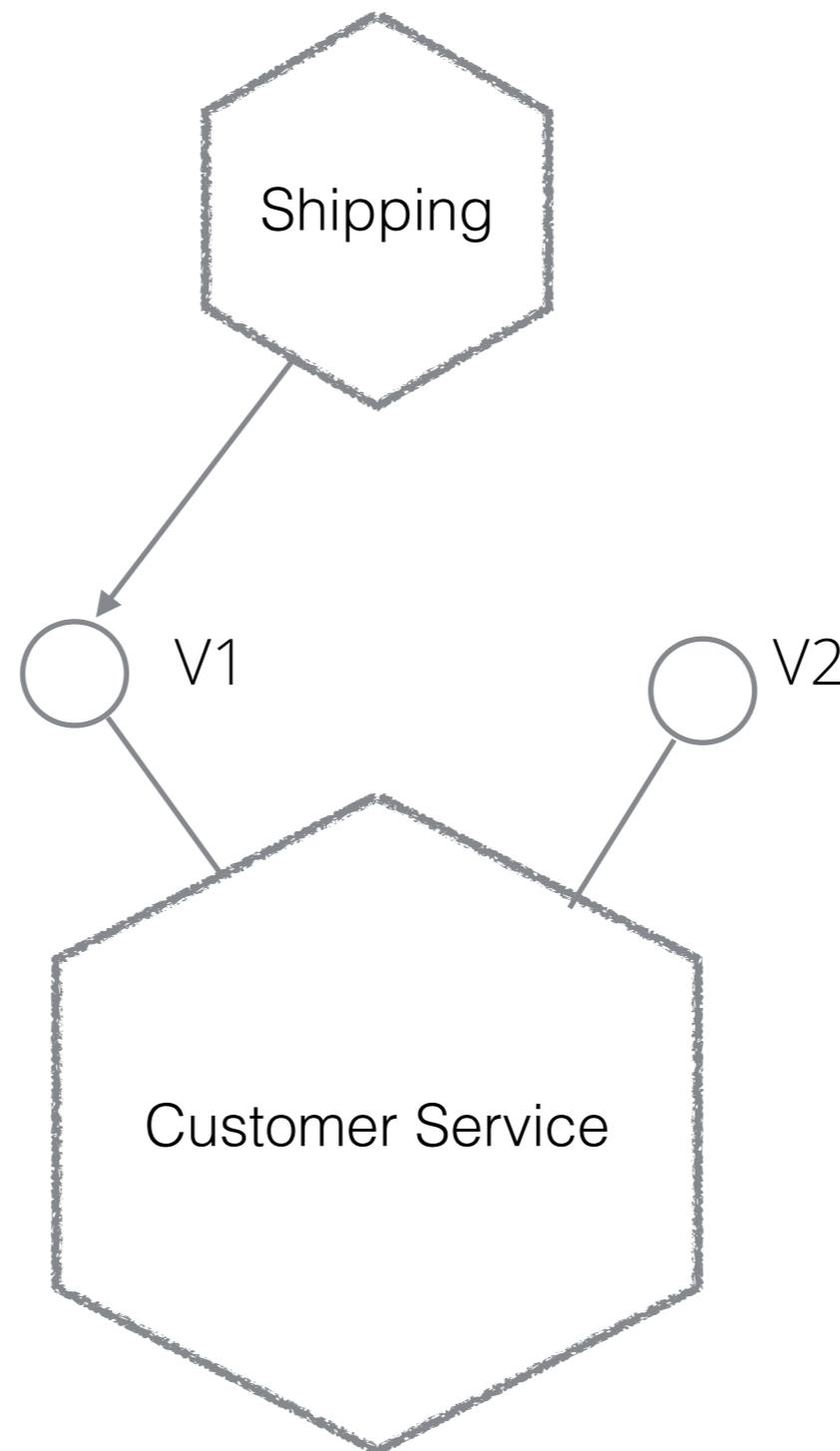
Travis CI Status: 

<https://github.com/realestate-com-au/pact>

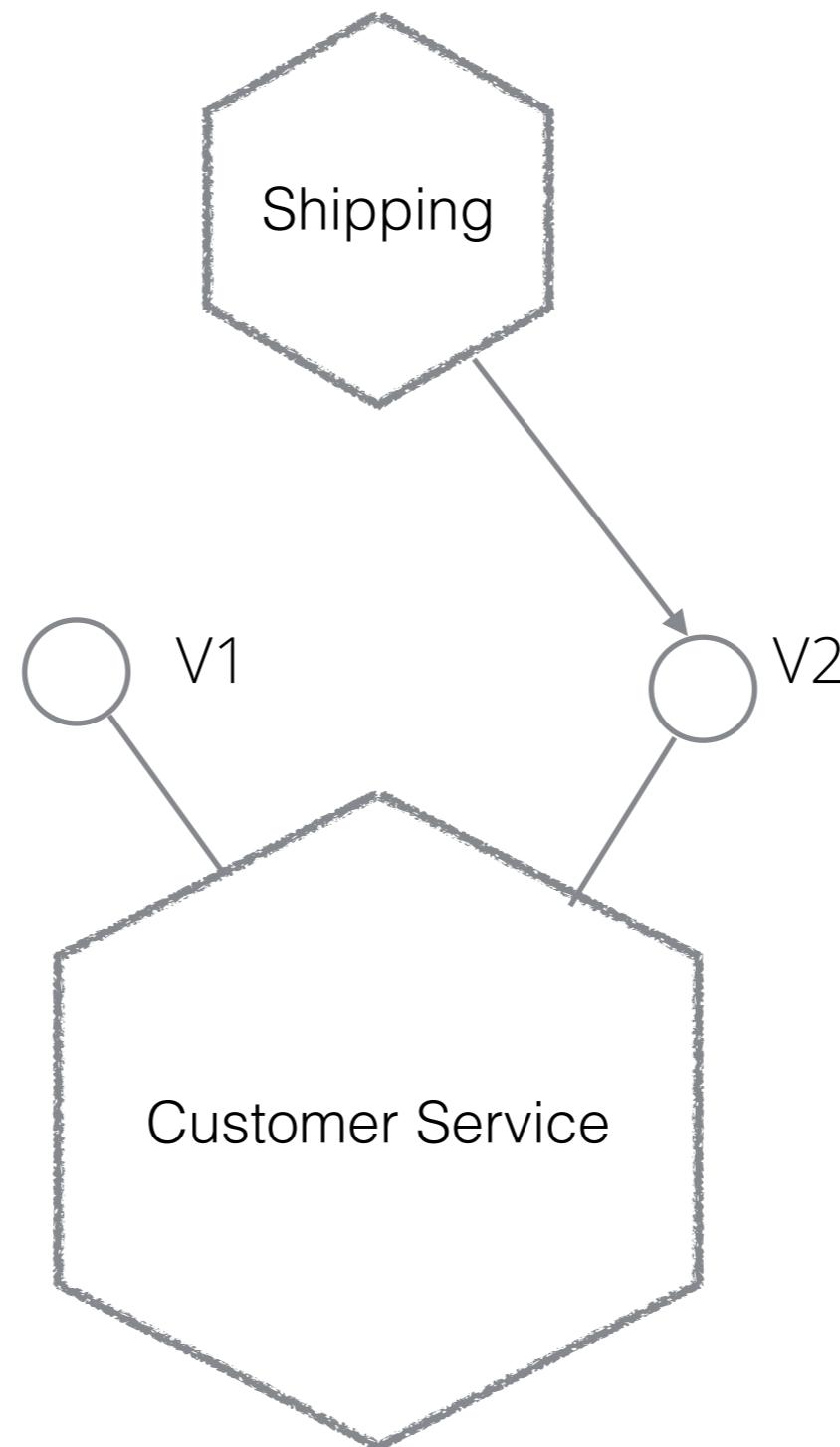
CO-EXIST ENDPOINTS



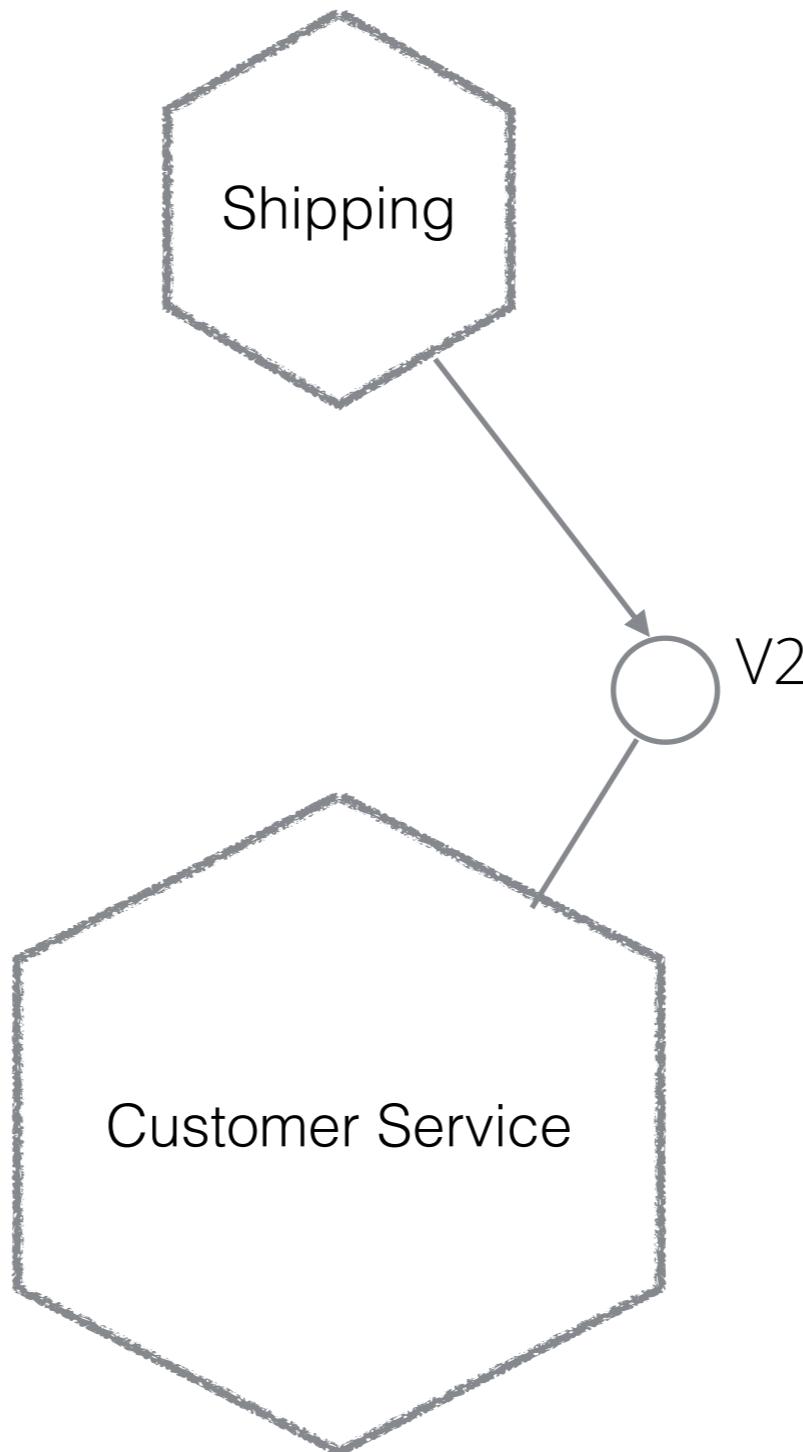
CO-EXIST ENDPOINTS



CO-EXIST ENDPOINTS



CO-EXIST ENDPOINTS



✓ Modelled Around Business Domain

✓ Culture Of Automation

✓ Hide Implementation Details

Highly Observable

Principles Of Microservices

Decentralise All The Things ✓

Isolate Failure

✓ Deploy Independently

✓ Modelled Around Business Domain

✓ Culture Of Automation

✓ Hide Implementation Details

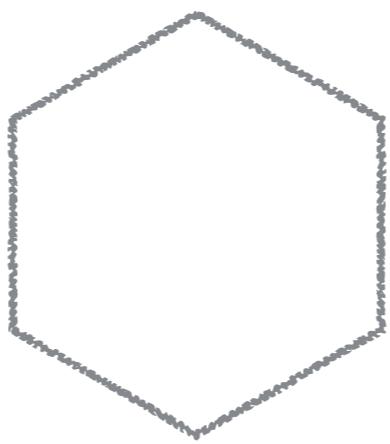
Highly Observable

Principles Of Microservices

Decentralise All The Things ✓

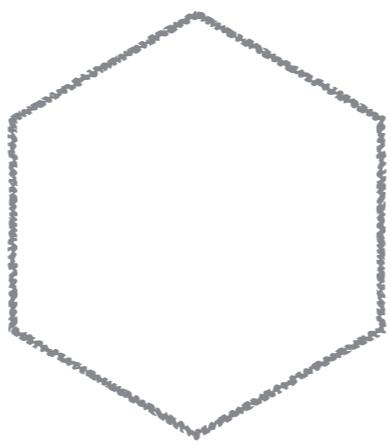
Isolate Failure

✓ Deploy Independently

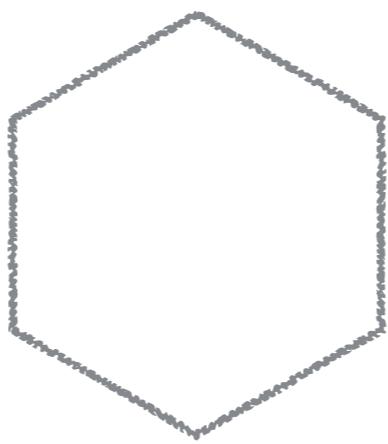


#xpdays

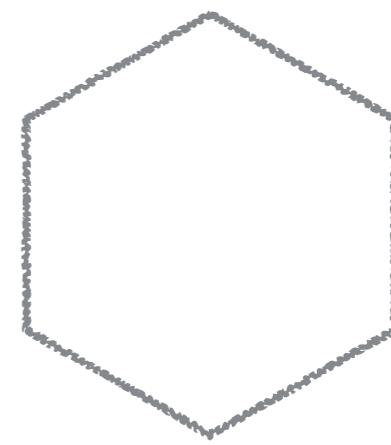
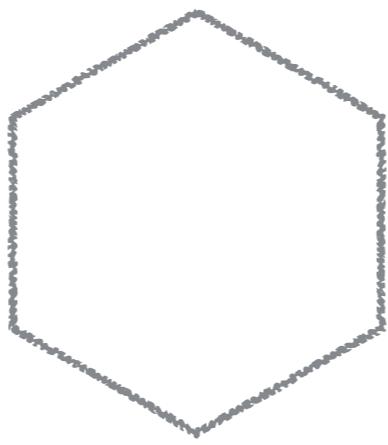
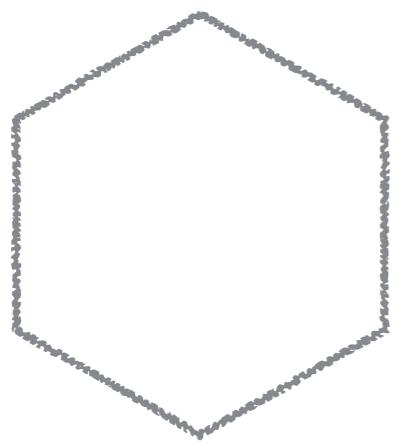
@samnewman

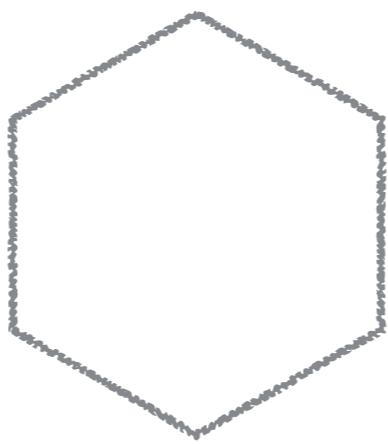


1 in 100

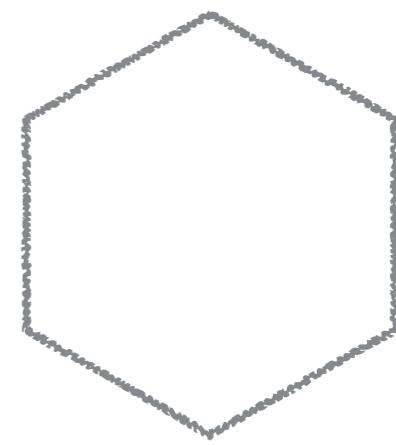
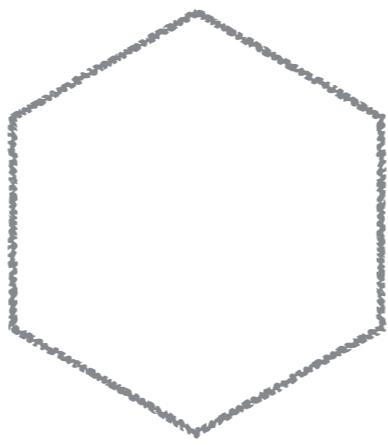
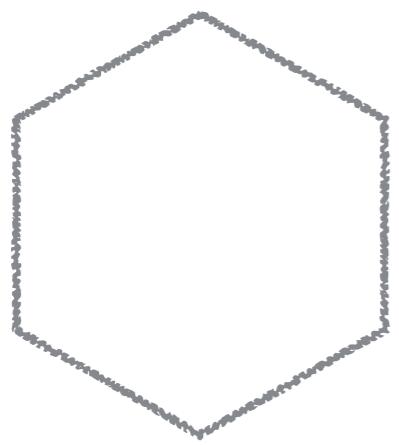


1 in 100

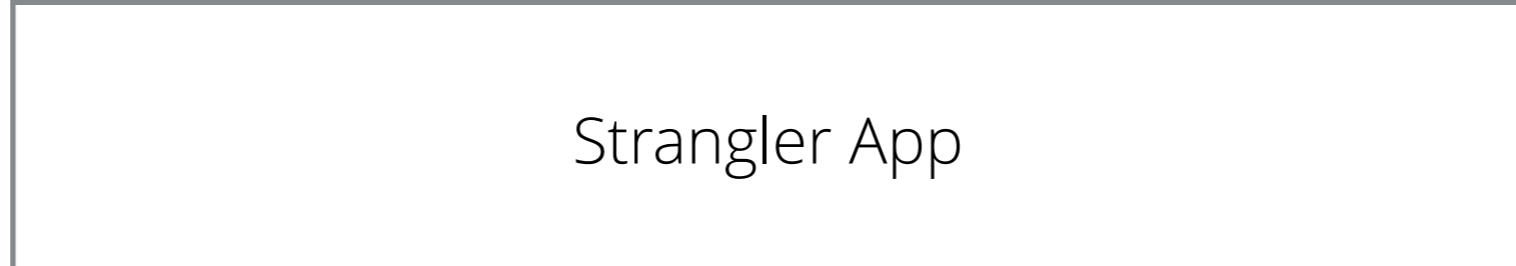




4 in 100



Strangler App

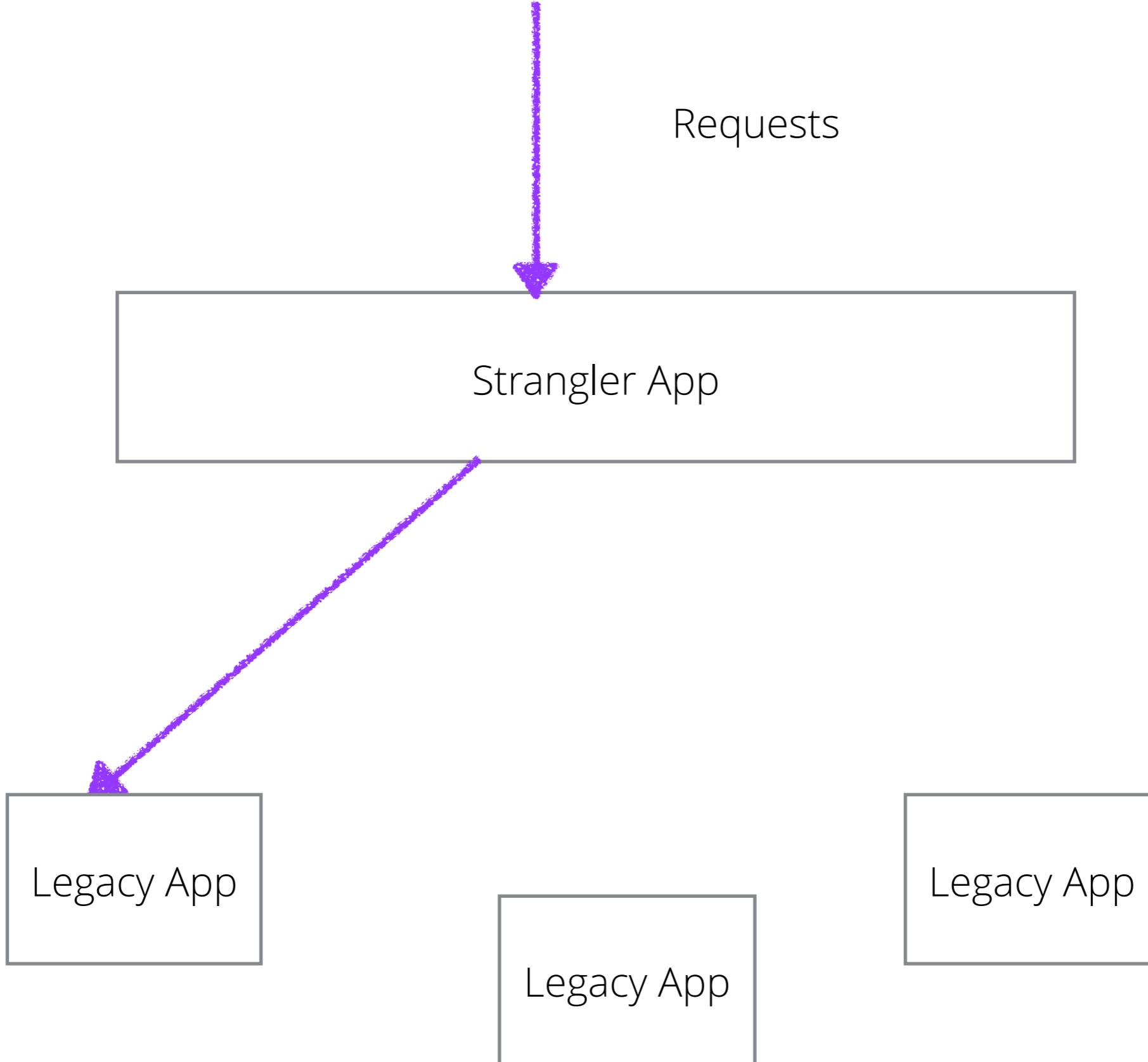


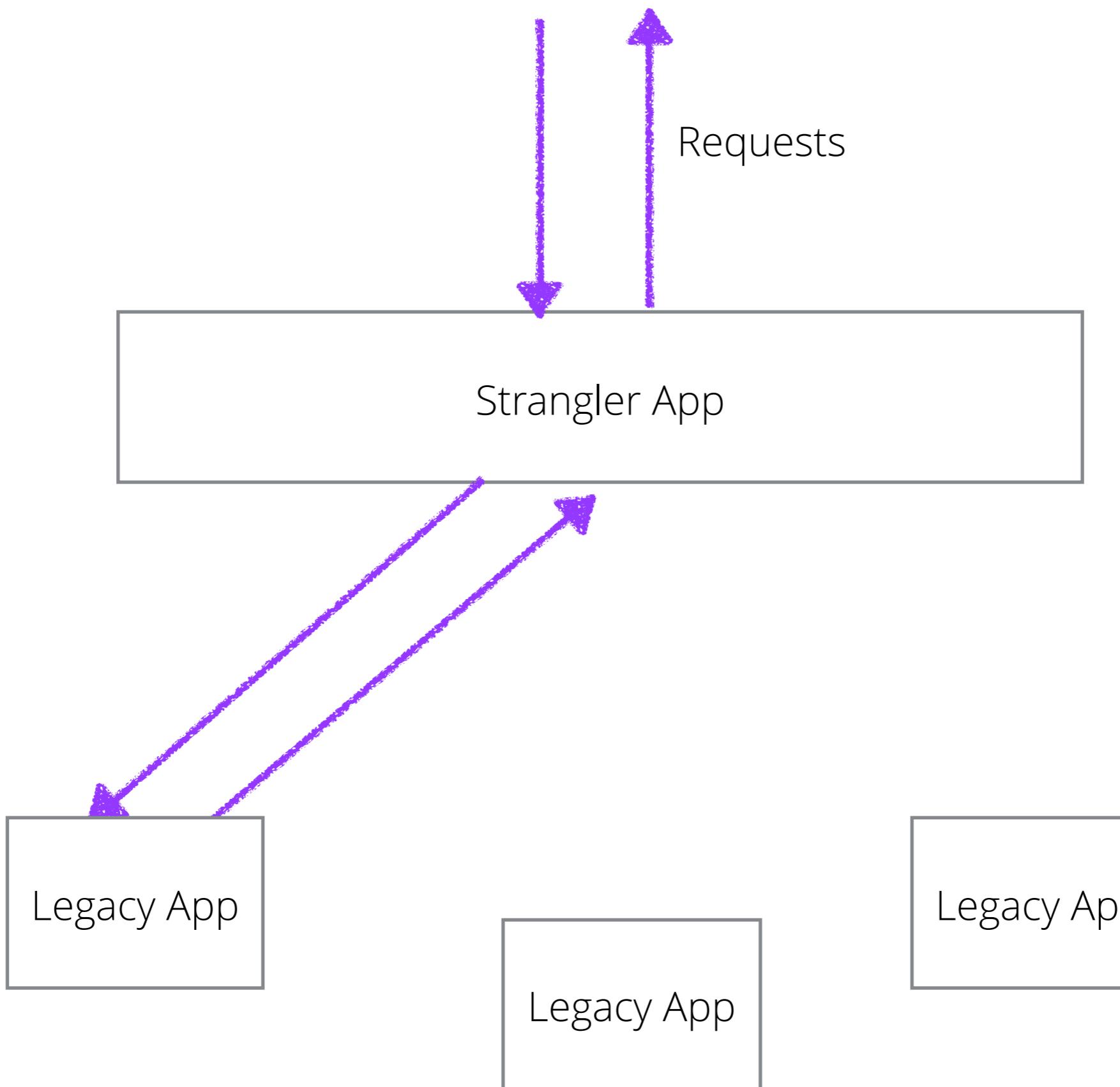
Strangler App

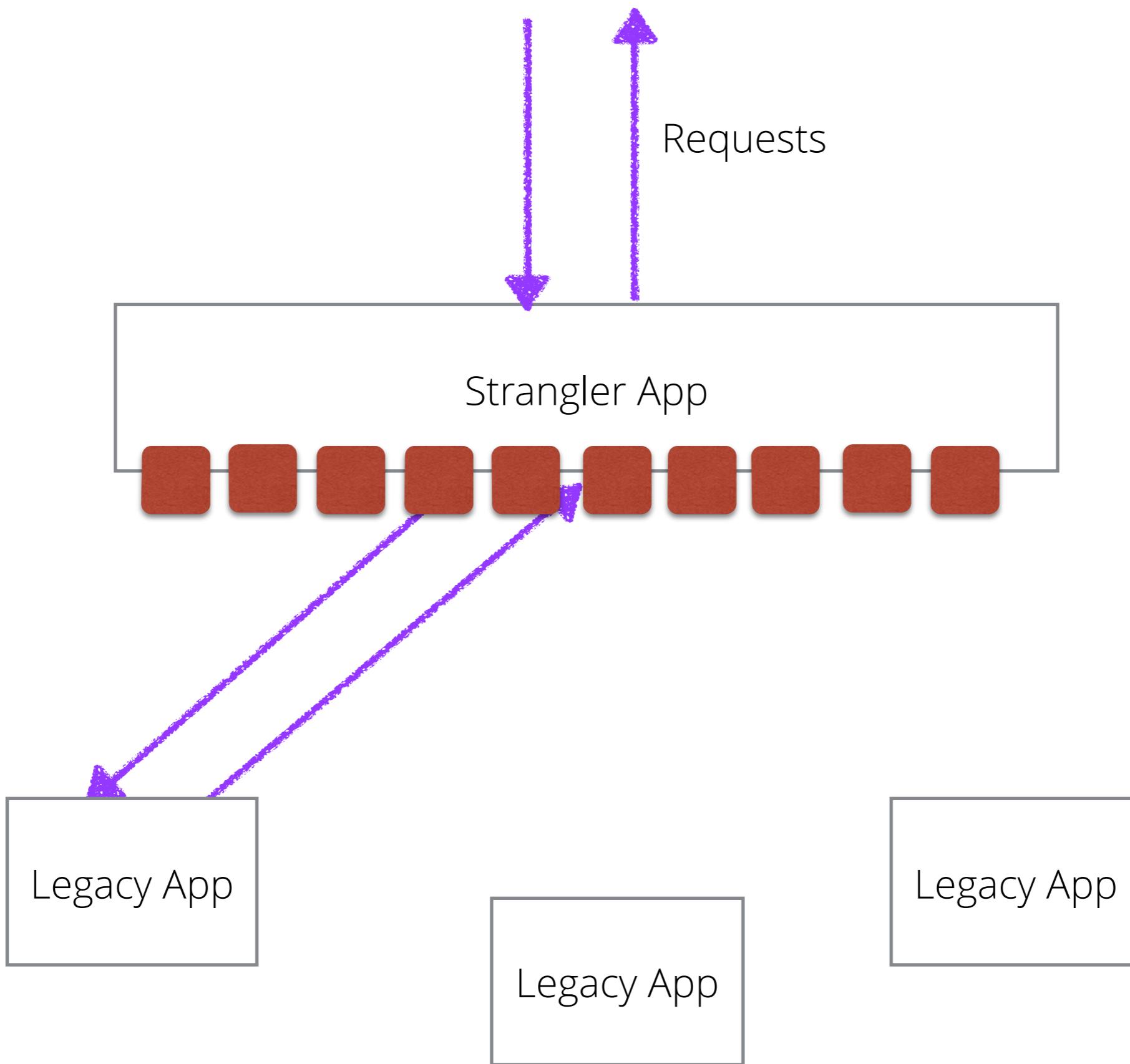
Legacy App

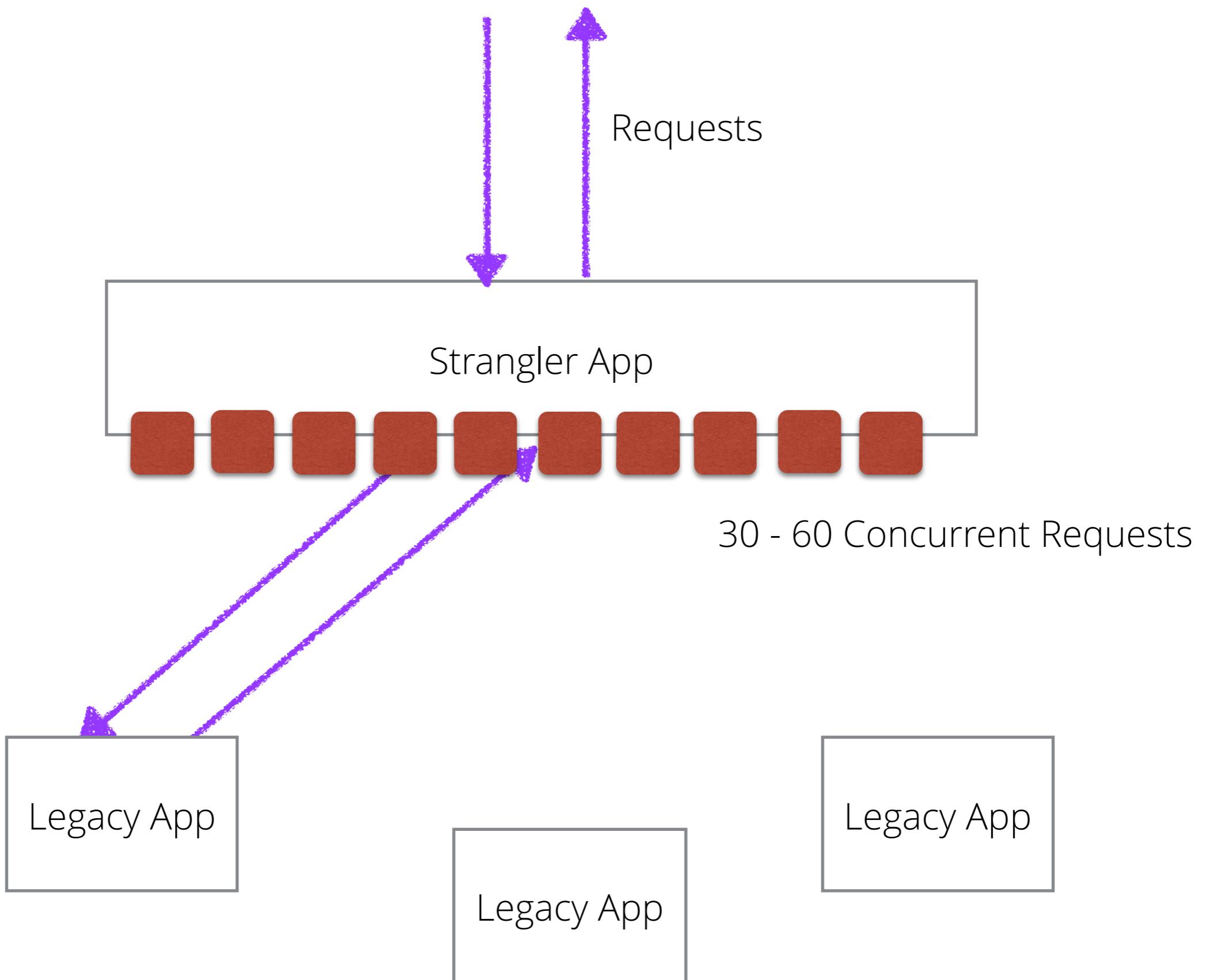
Legacy App

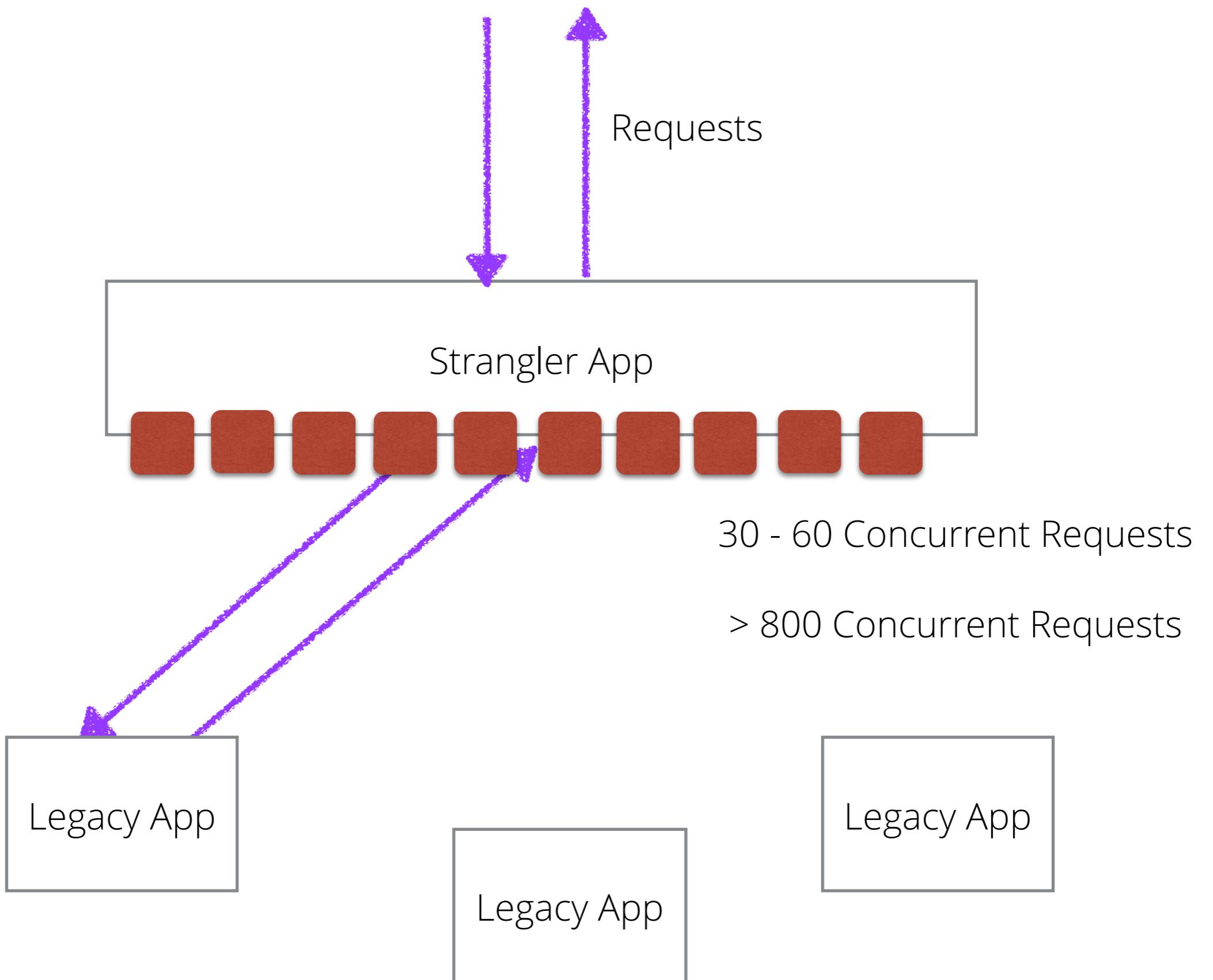
Legacy App

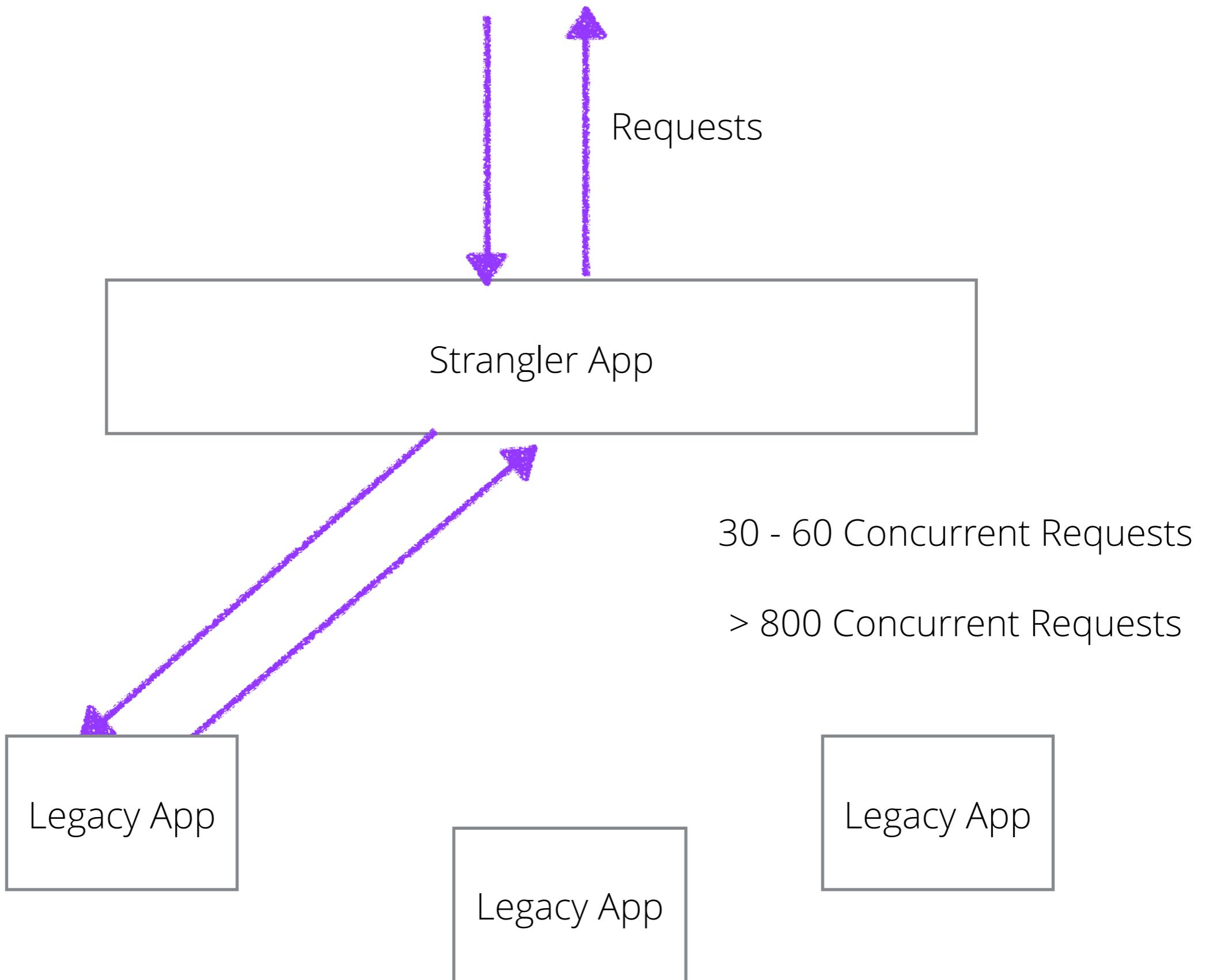






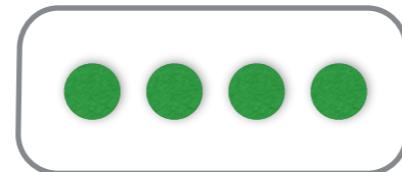






Strangler App

Thread Pool



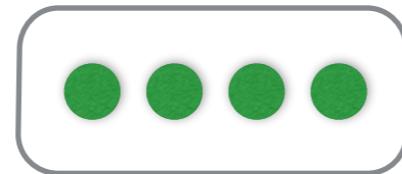
Legacy App

Legacy App

Legacy App

Strangler App

Thread Pool



Failing...slowly!

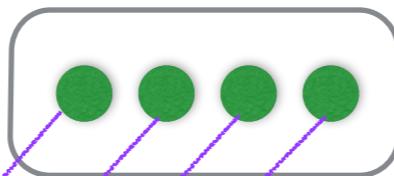
Legacy App

Legacy App

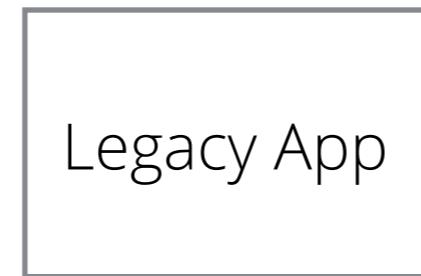
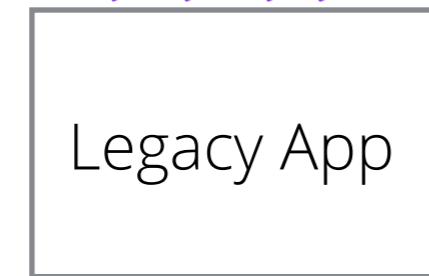
Legacy App

Strangler App

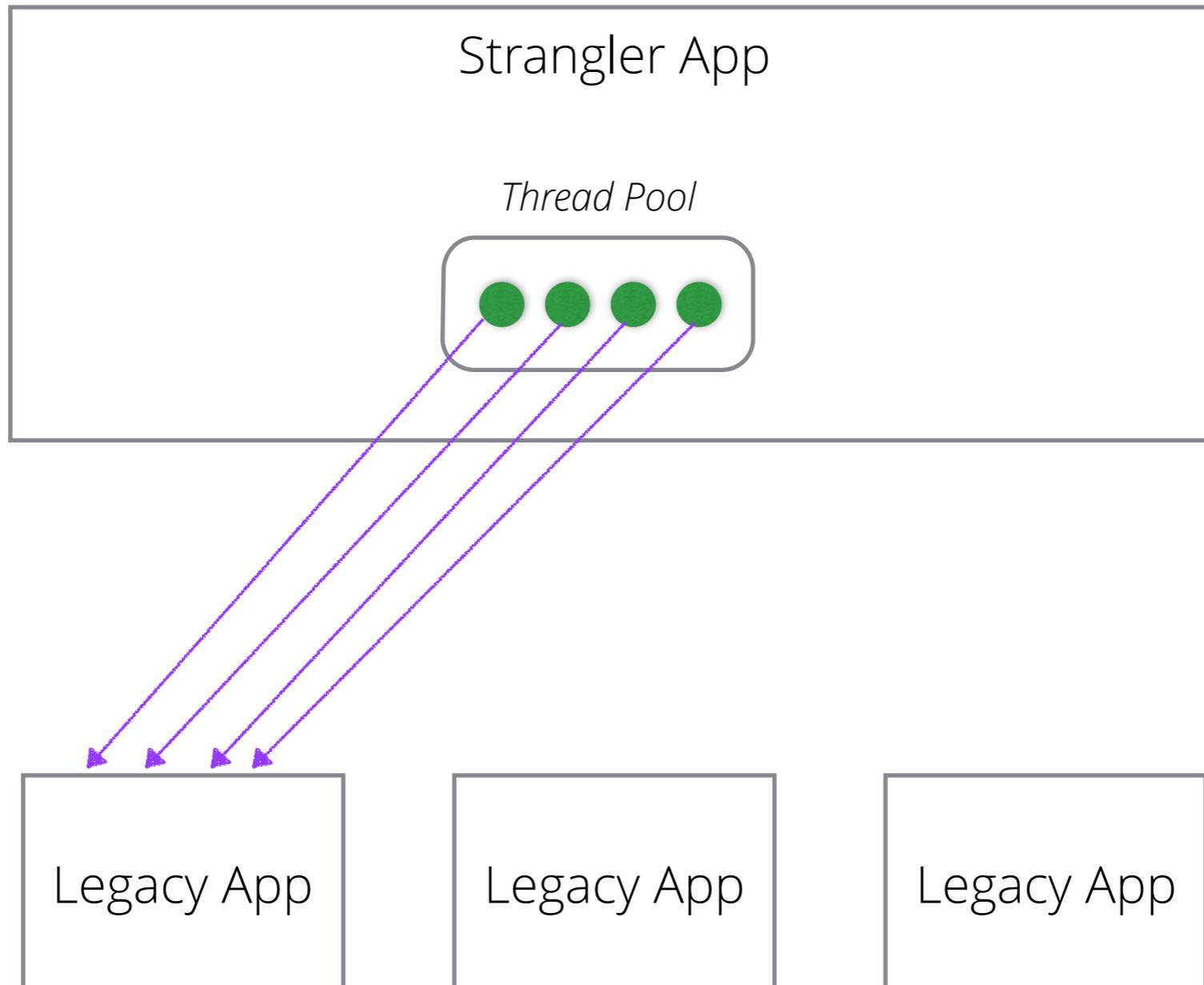
Thread Pool



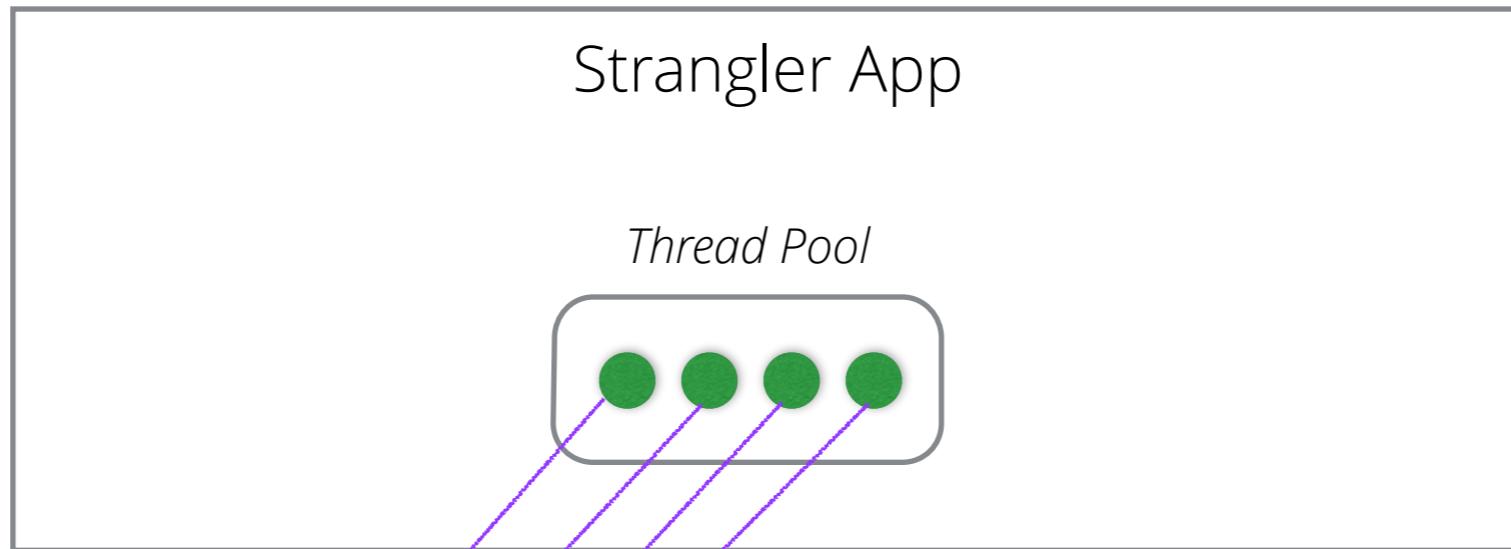
Failing...slowly!



Thread-pool
exhausted



Thread-pool
exhausted



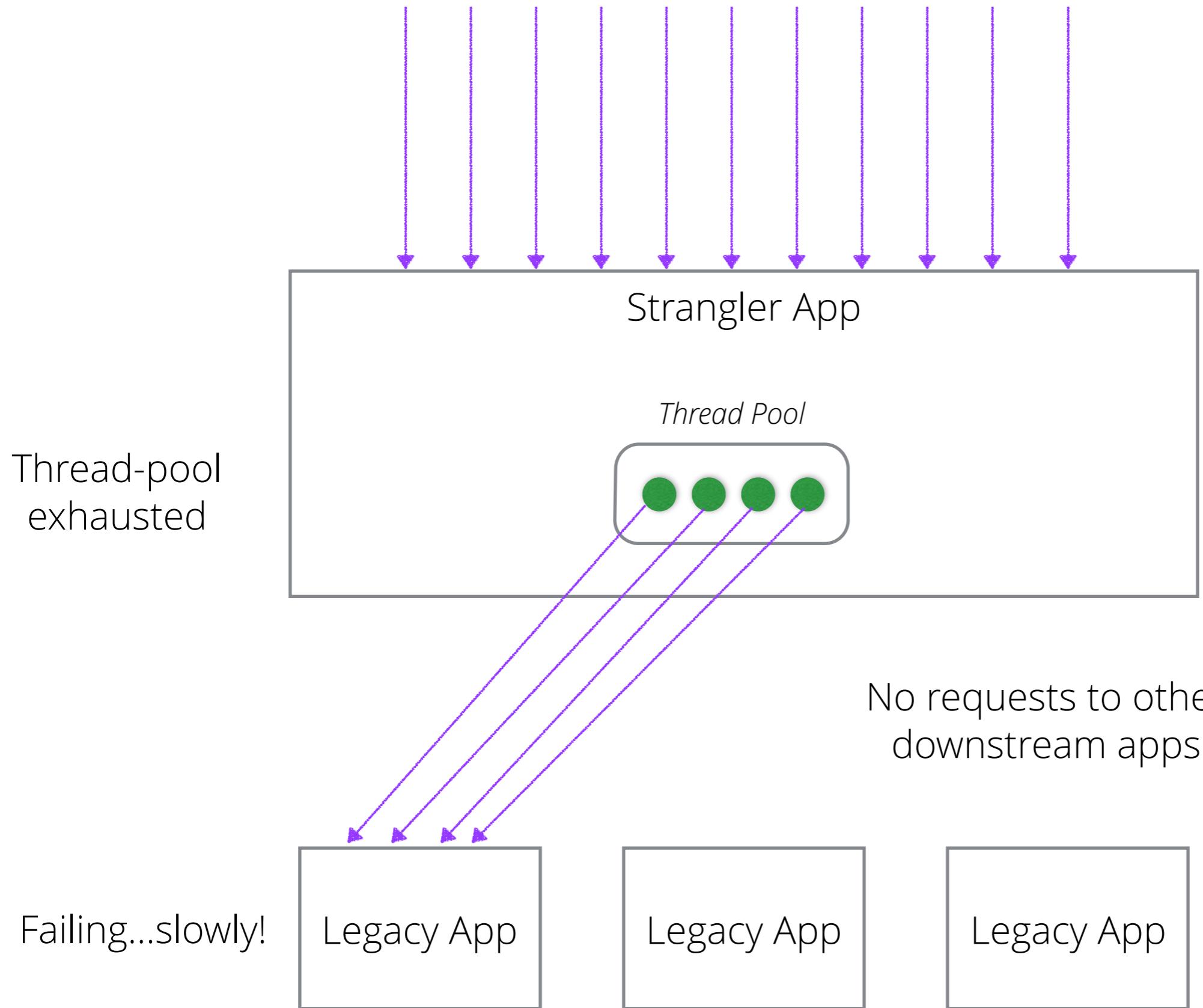
No requests to other
downstream apps

Failing...slowly!

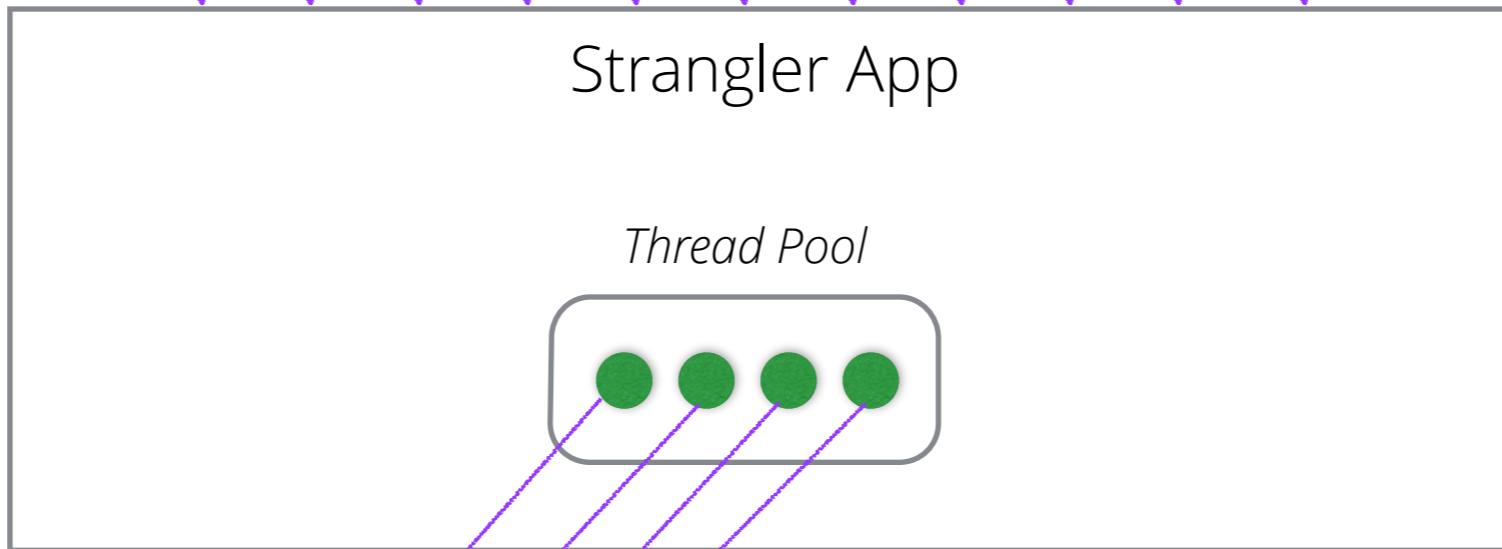
Legacy App

Legacy App

Legacy App



Requests
Building Up



No requests to other
downstream apps

Failing...slowly!

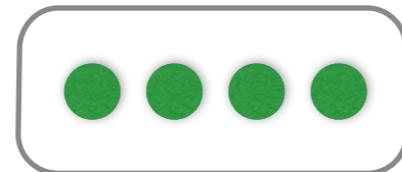
Legacy App

Legacy App

Legacy App

Strangler App

Thread Pool

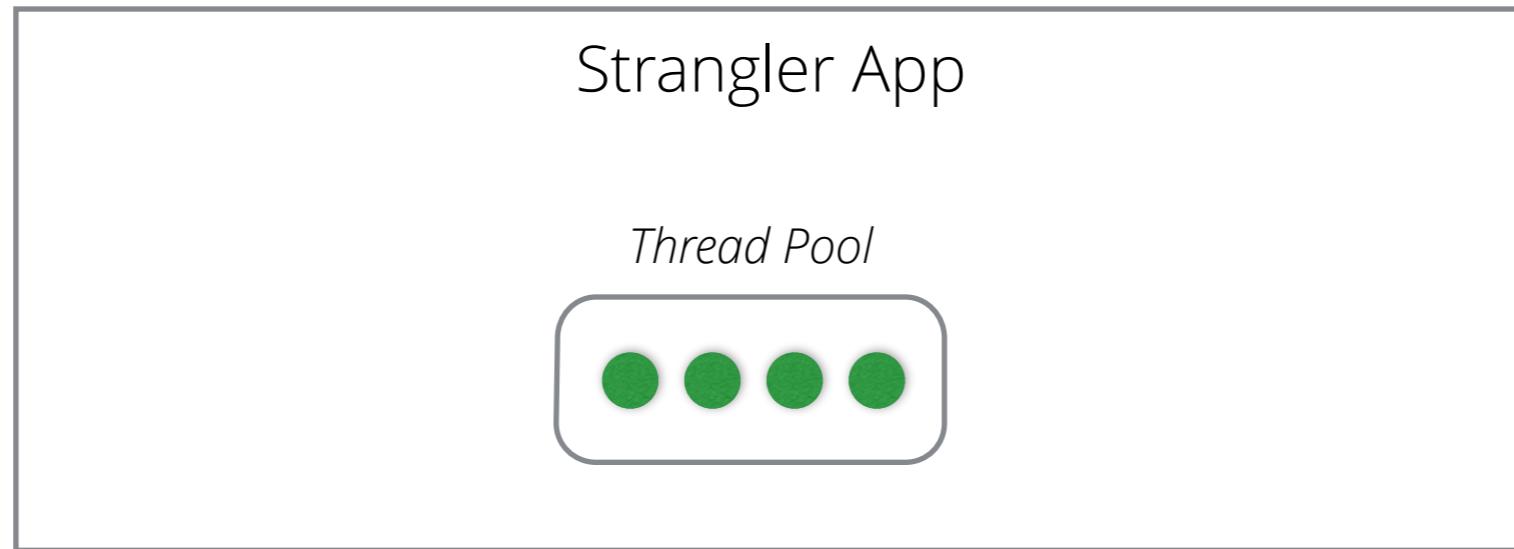


Legacy App

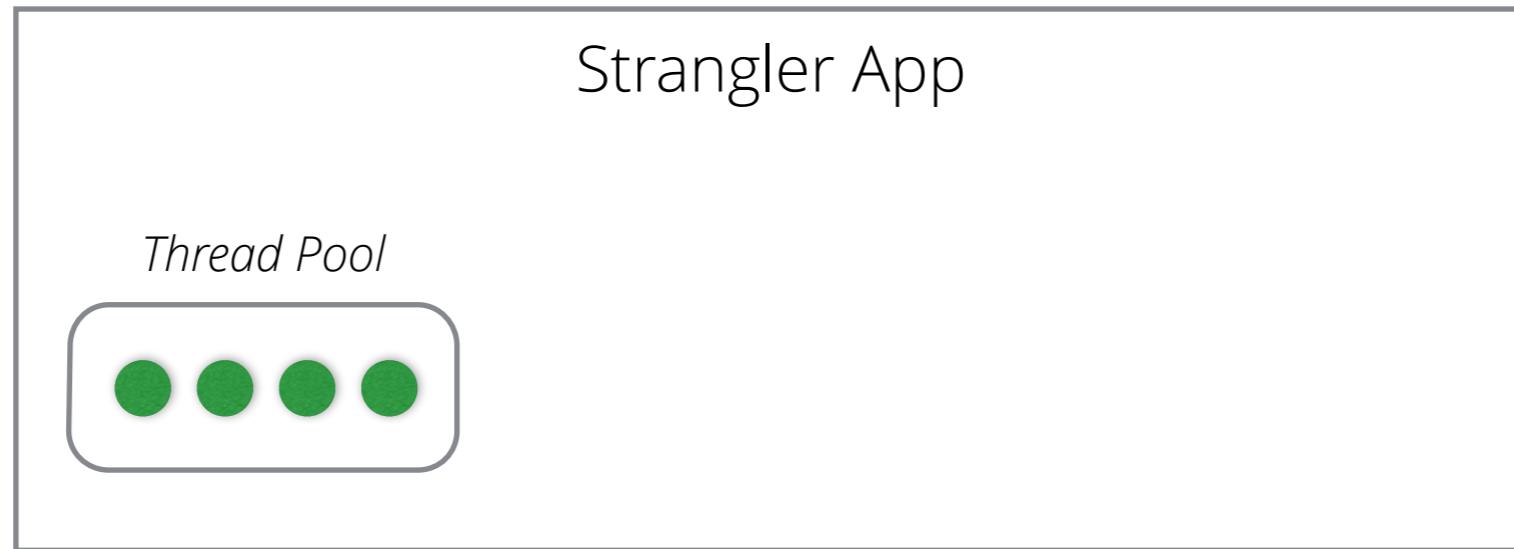
Legacy App

Legacy App

Fix **Timeouts**



Fix **Timeouts**

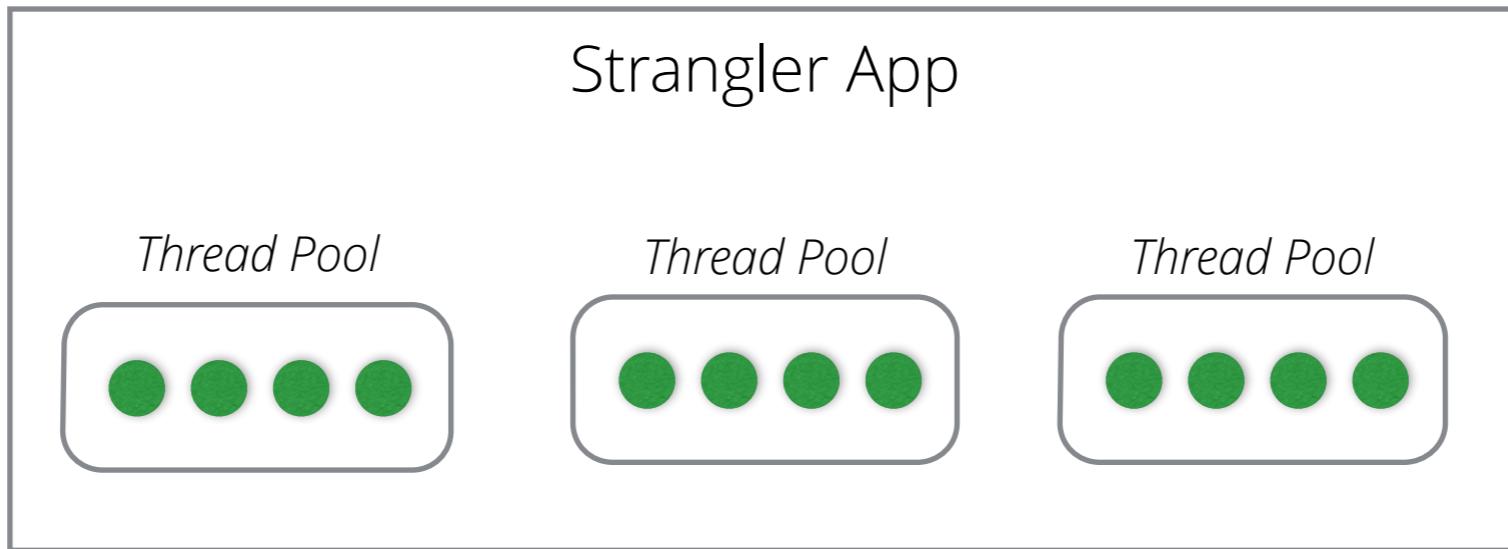


Legacy App

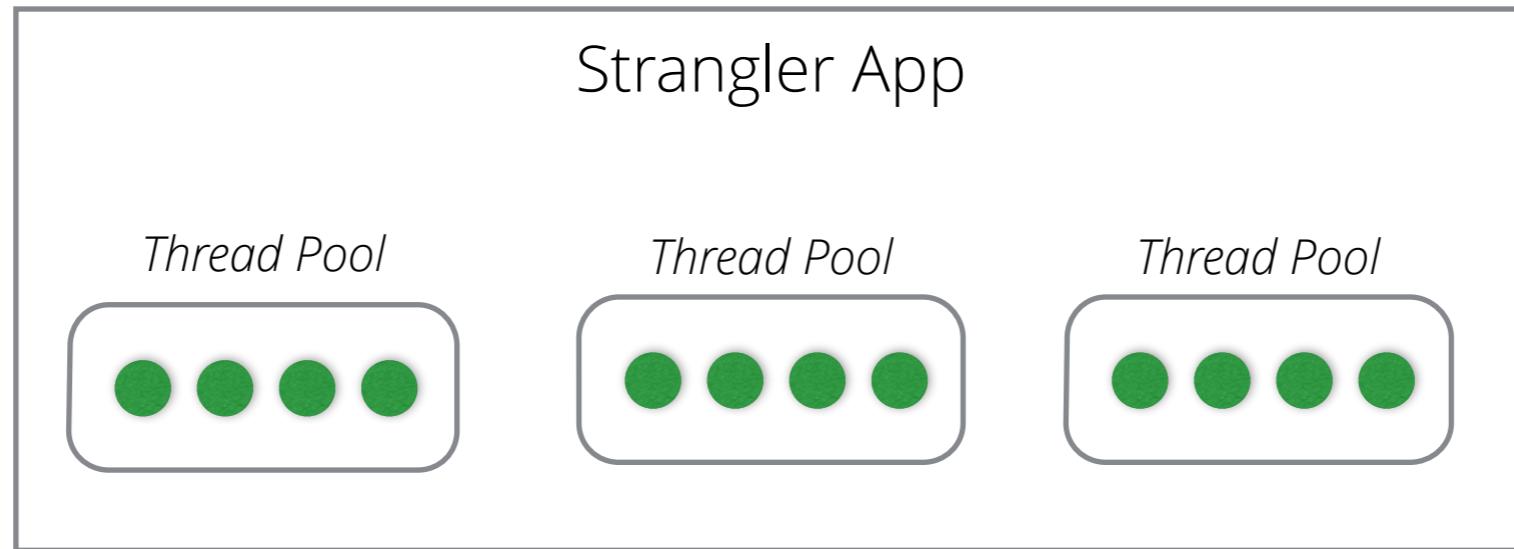
Legacy App

Legacy App

Fix **Timeouts**



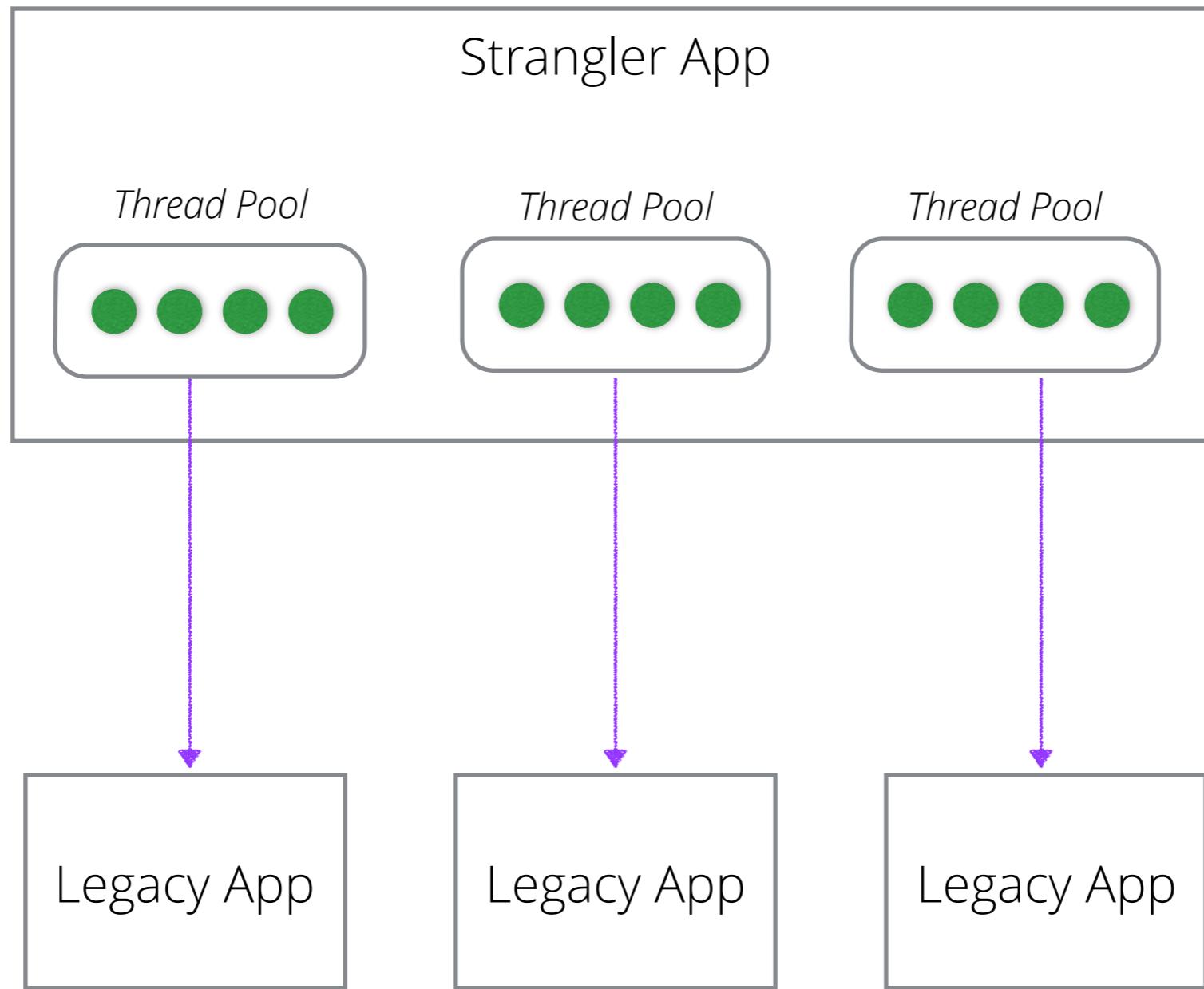
Fix **Timeouts**



Bulkhead
Downstream
Connections

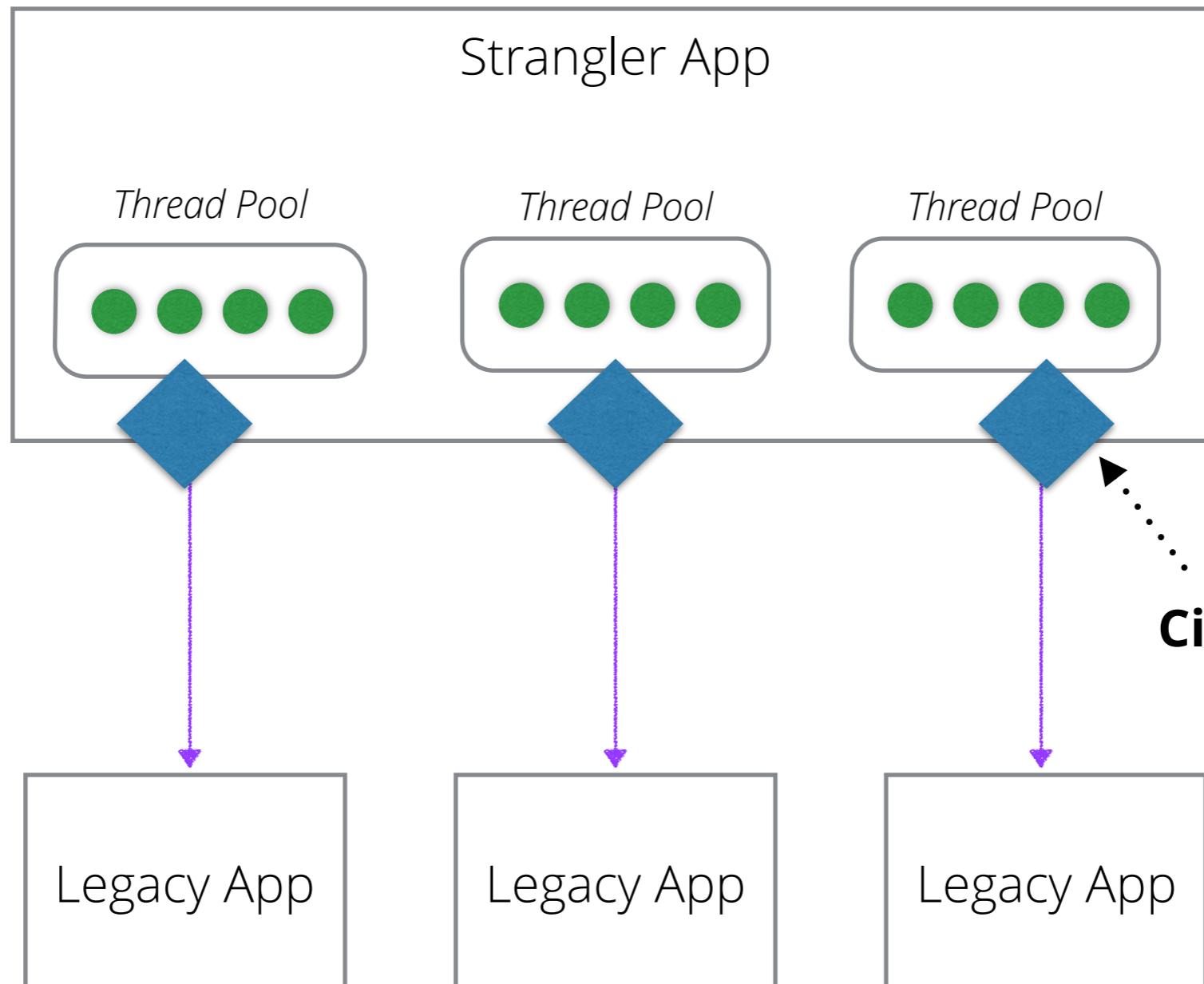


Fix **Timeouts**

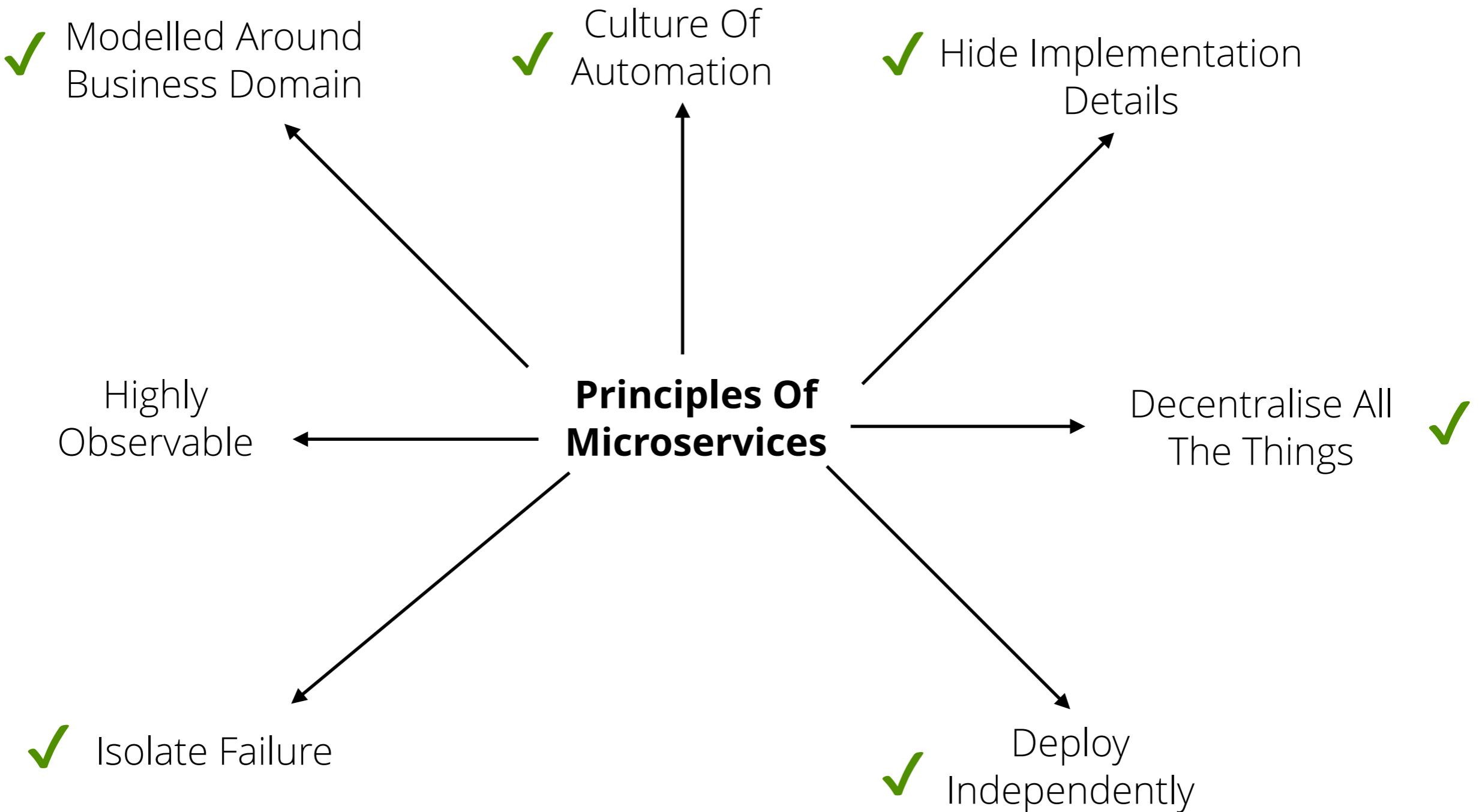


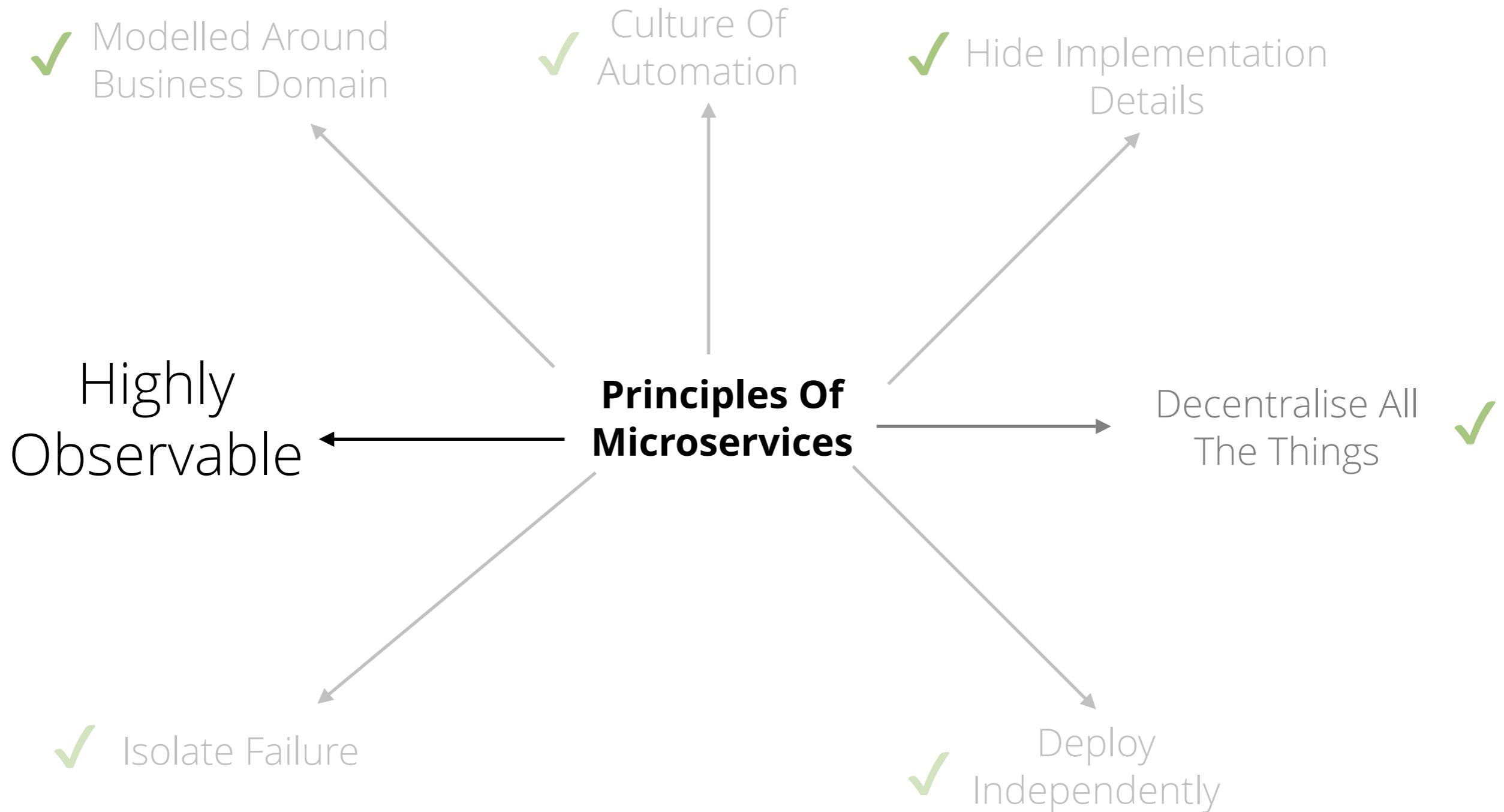
Bulkhead
Downstream
Connections

Fix **Timeouts**



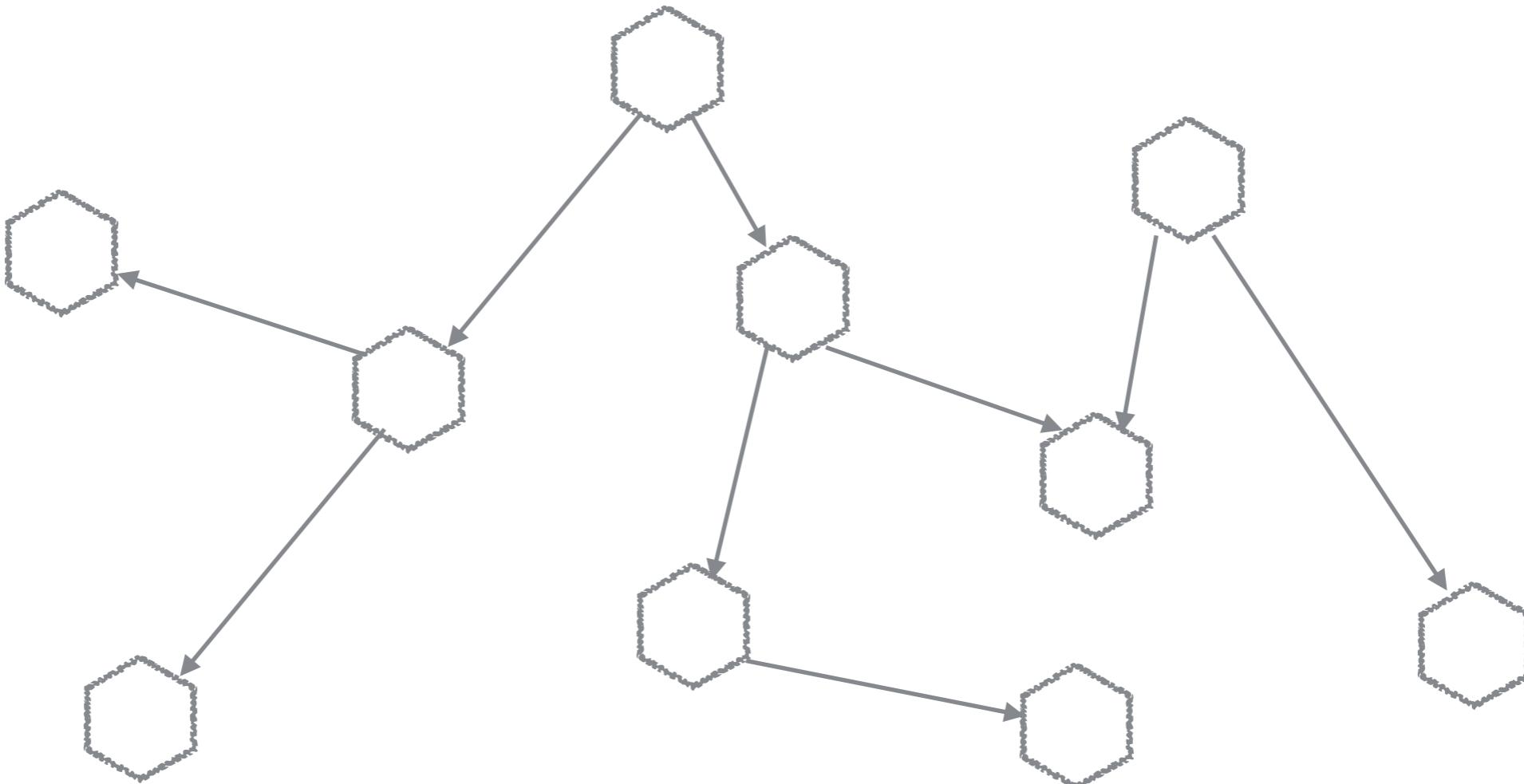
Circuit Breakers







AGGREGATION



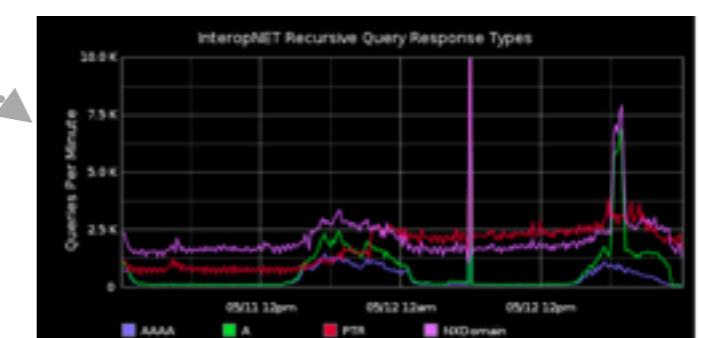
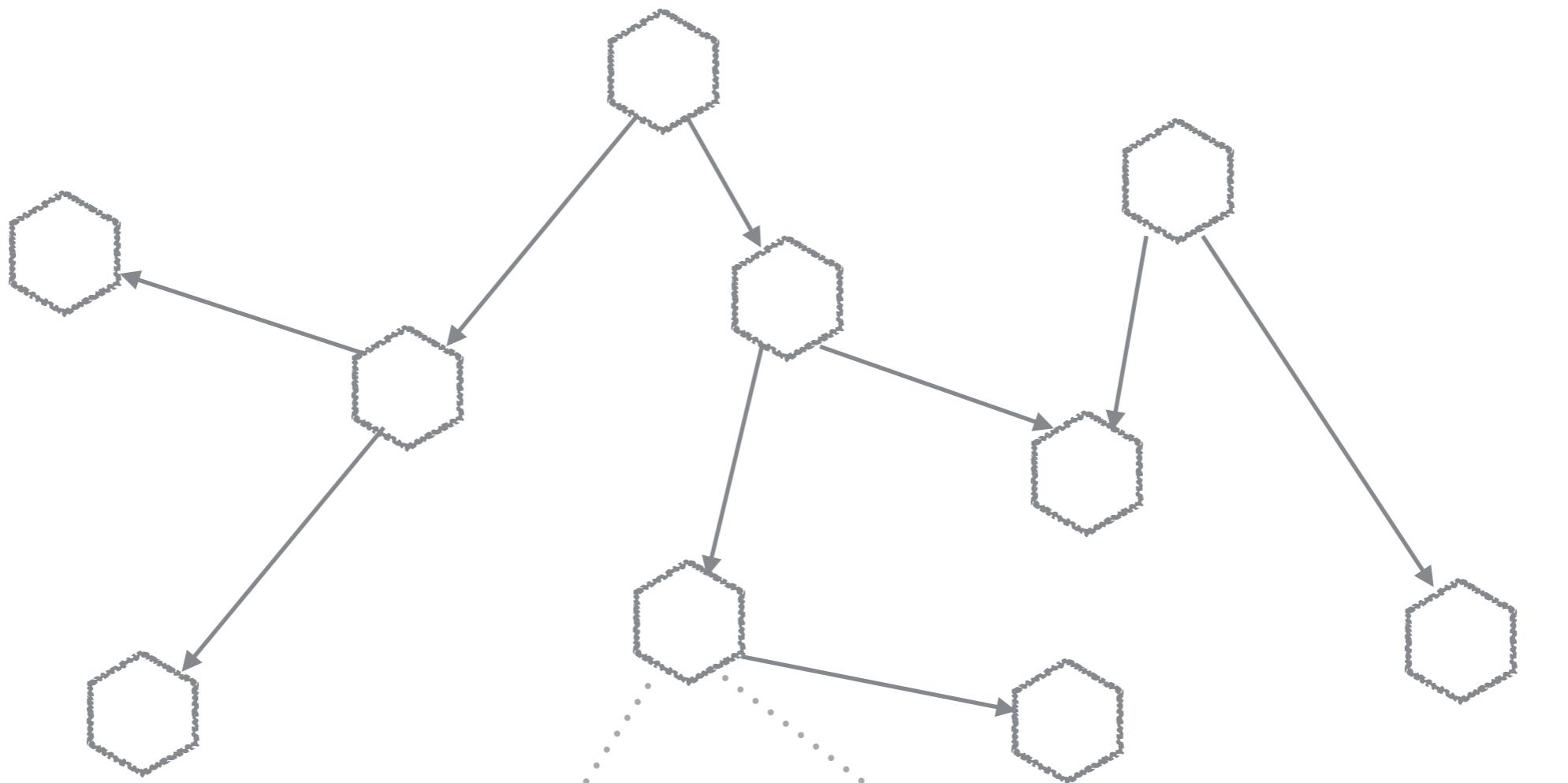
AGGREGATION



#xpdays

@samnewman

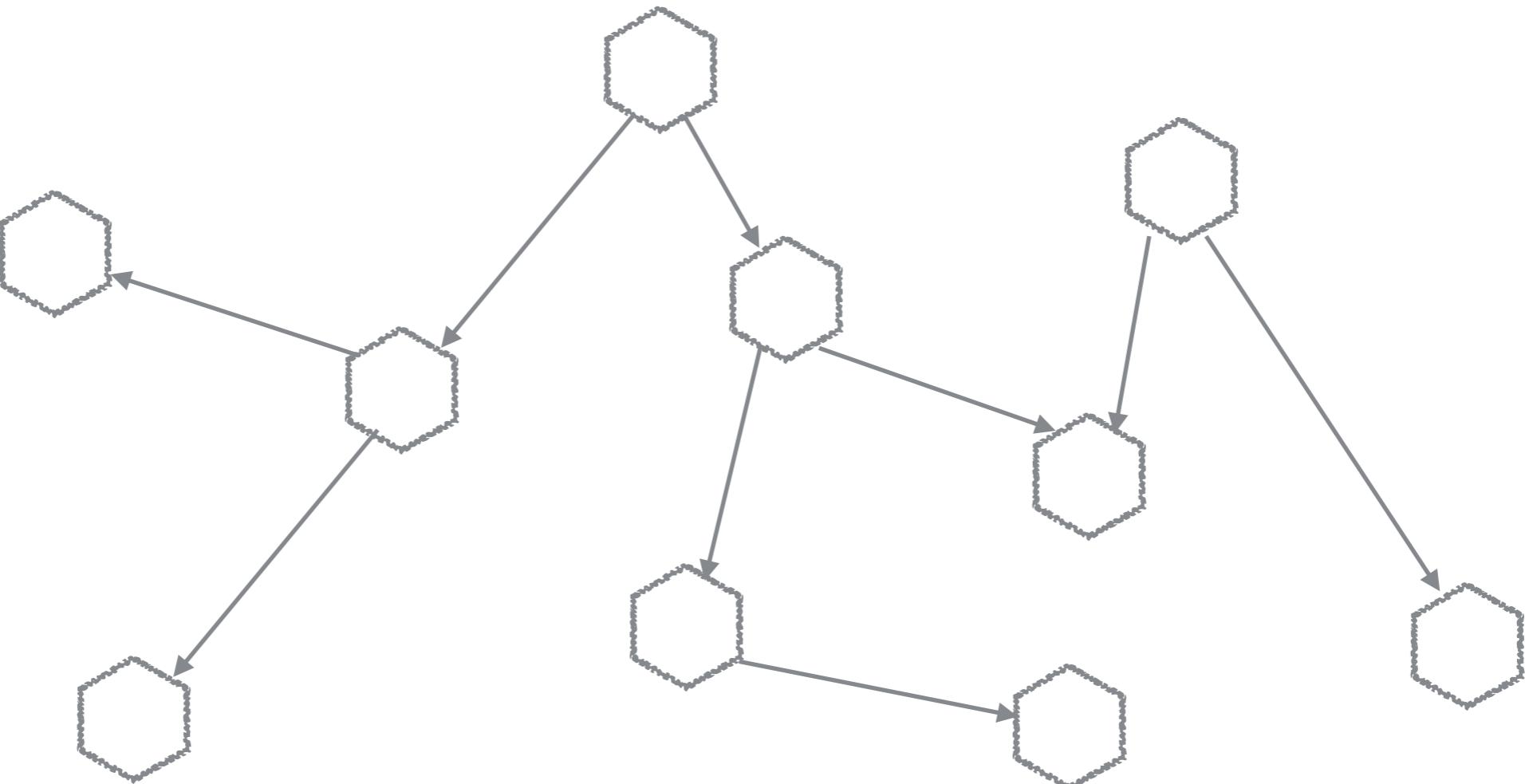
AGGREGATION



#xpdays

@samnewman

CORRELATION IDS

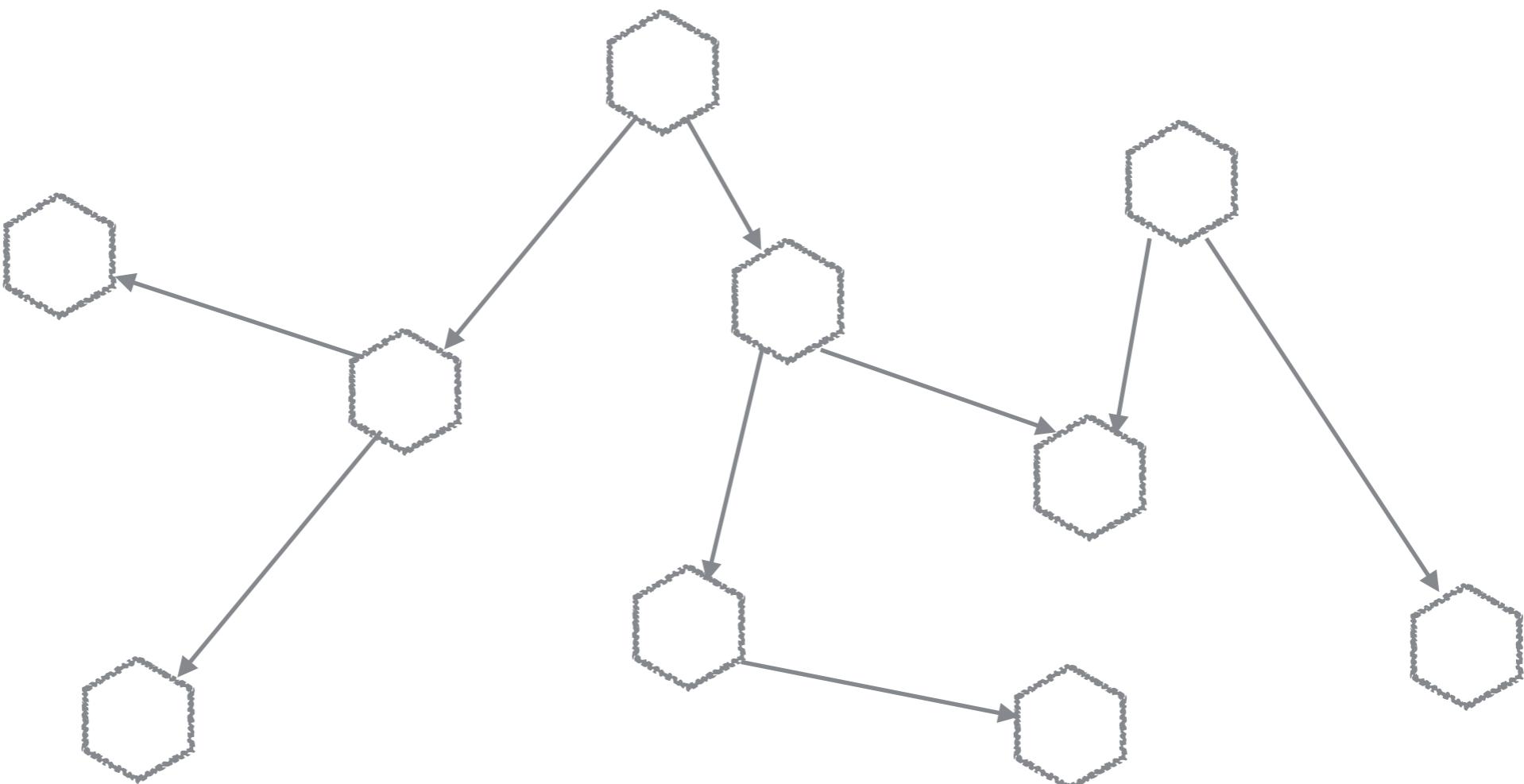


#xpdays

@samnewman

CORRELATION IDS

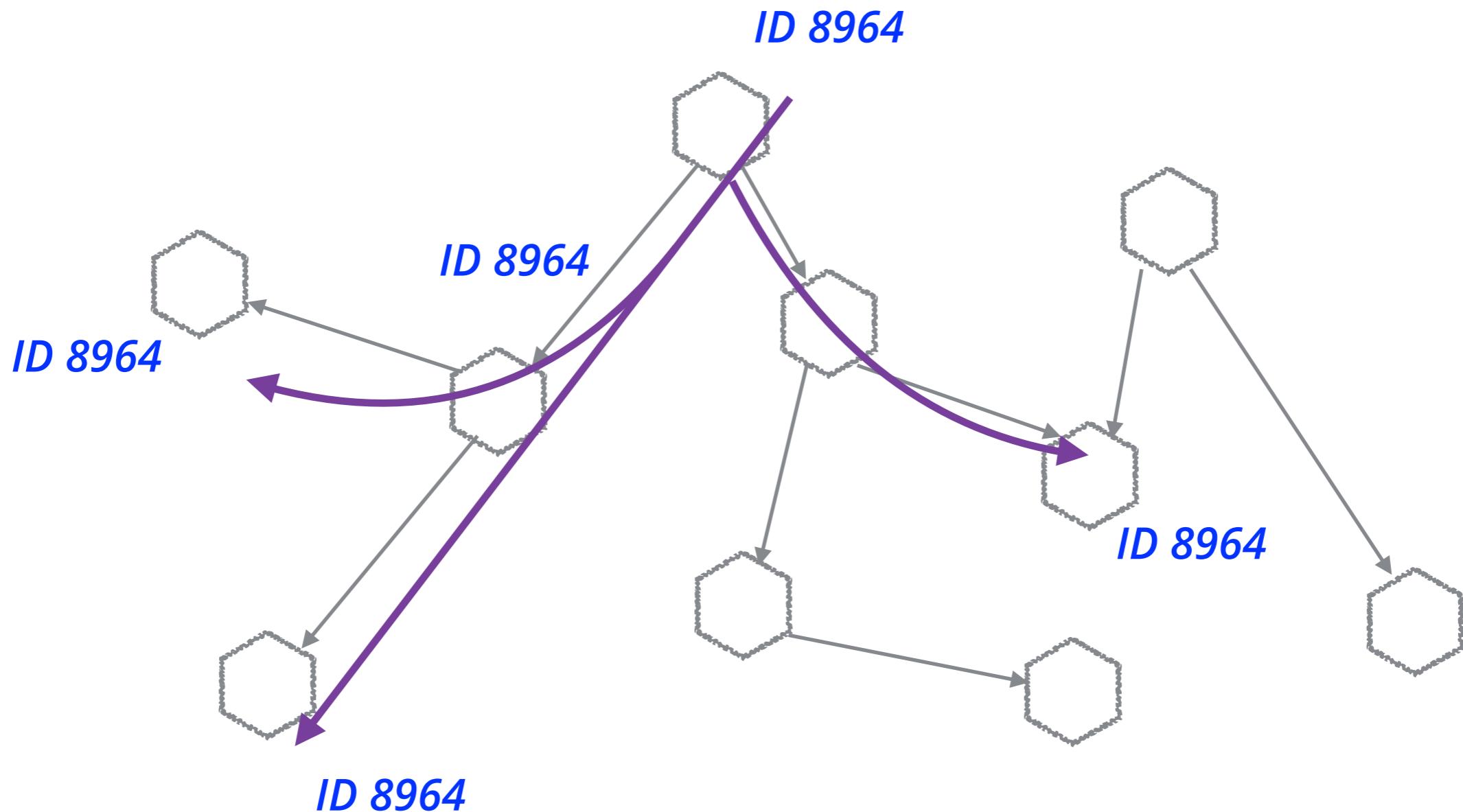
ID 8964



#xpdays

@samnewman

CORRELATION IDS

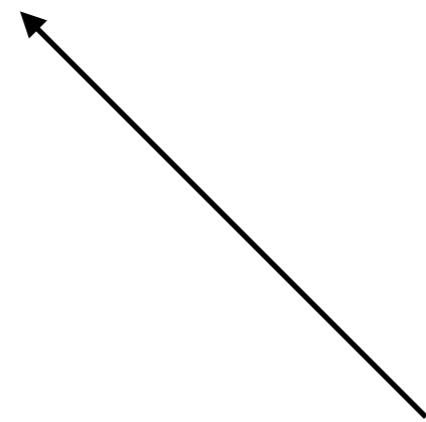


Principles Of Microservices

#xpdays

@samnewman

Modelled Around
Business Domain

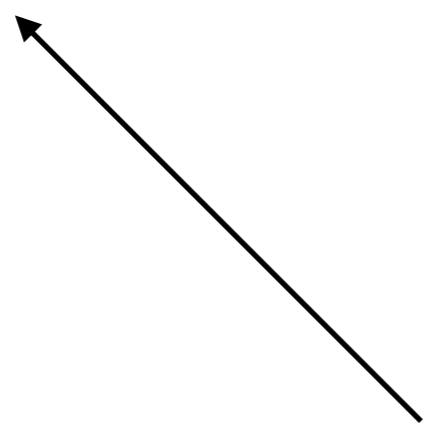


Principles Of Microservices

Modelled Around
Business Domain

Culture Of
Automation

**Principles Of
Microservices**



Modelled Around
Business Domain

Culture Of
Automation

Hide Implementation
Details

**Principles Of
Microservices**

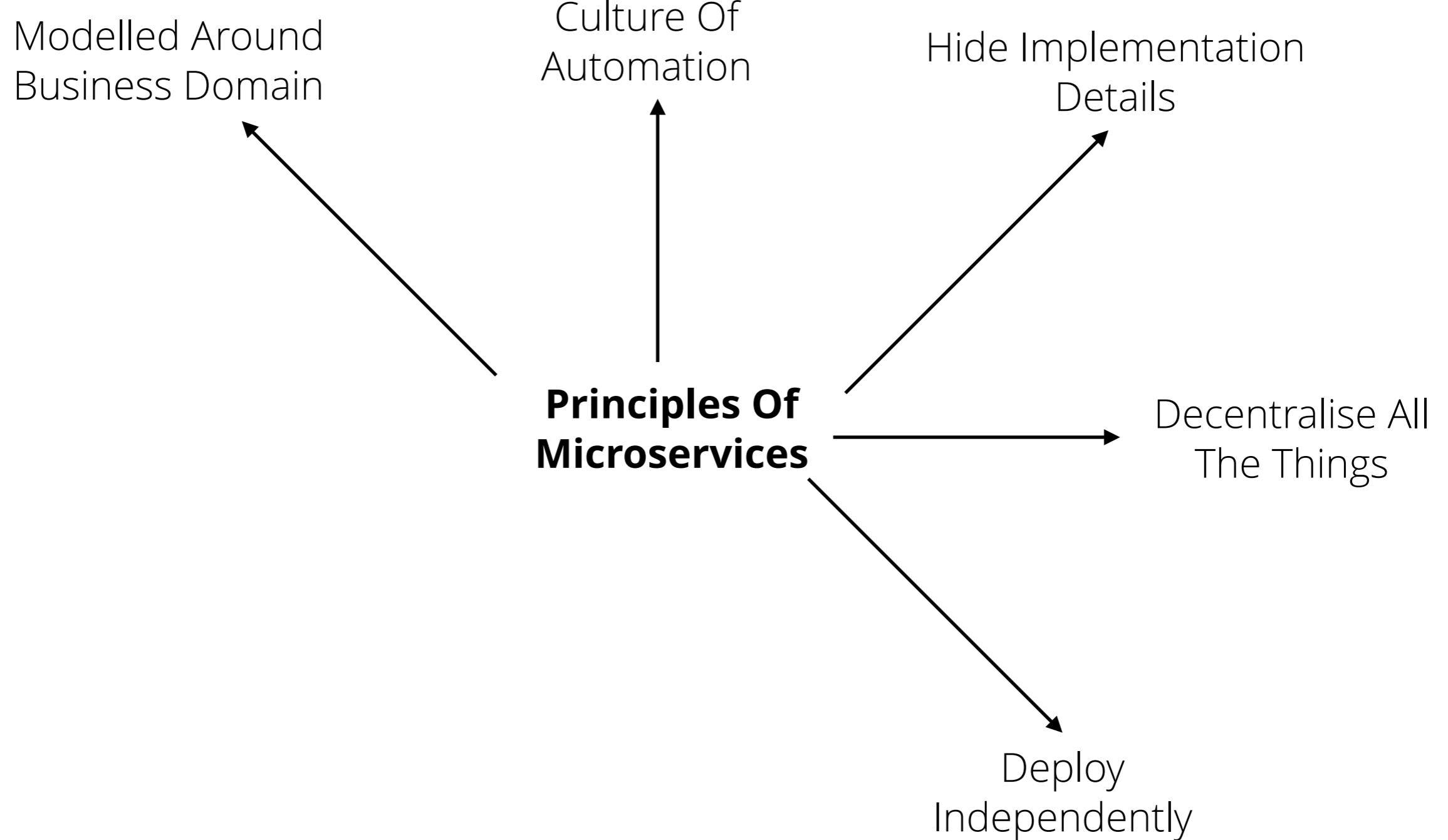
Modelled Around
Business Domain

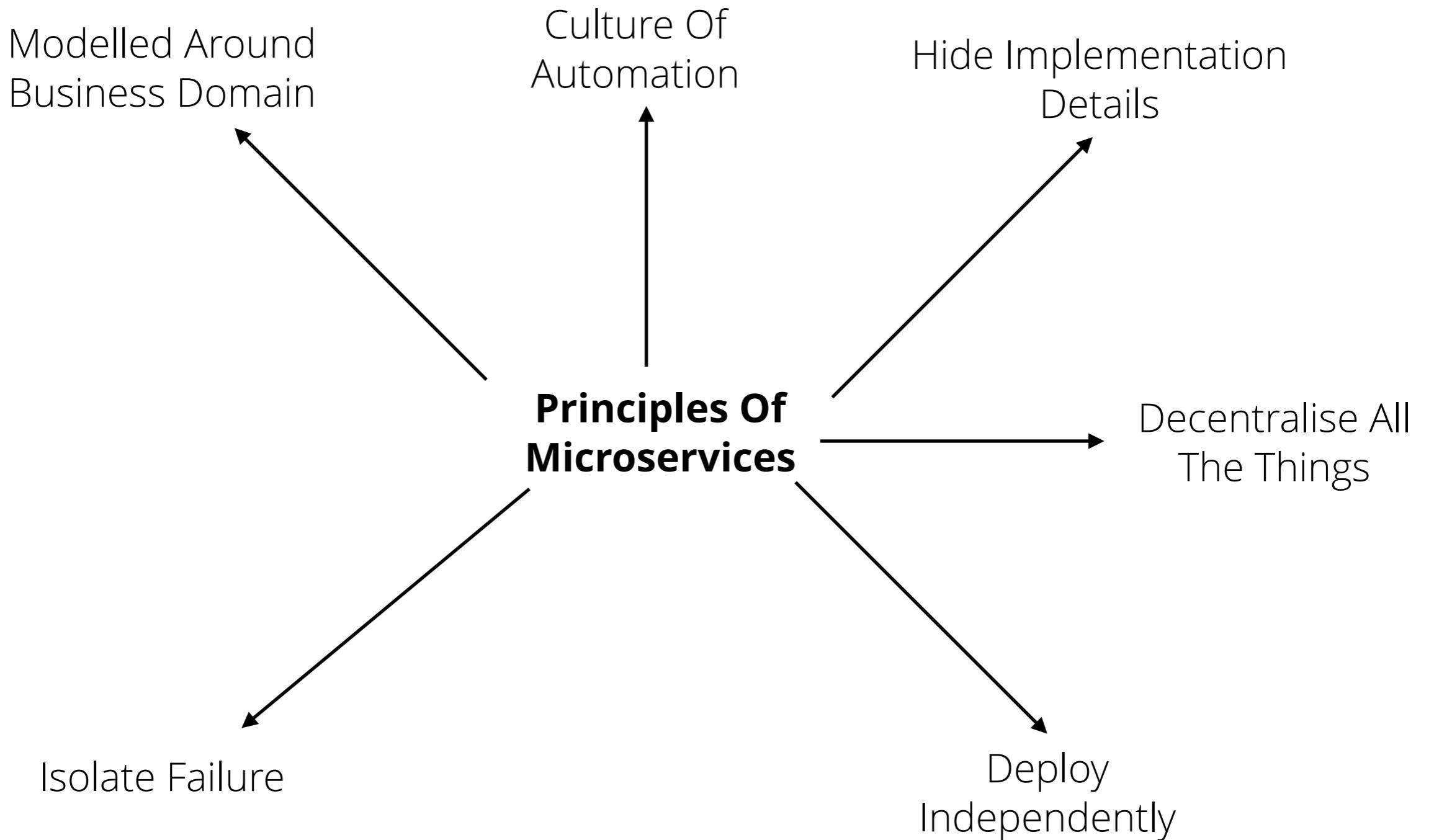
Culture Of
Automation

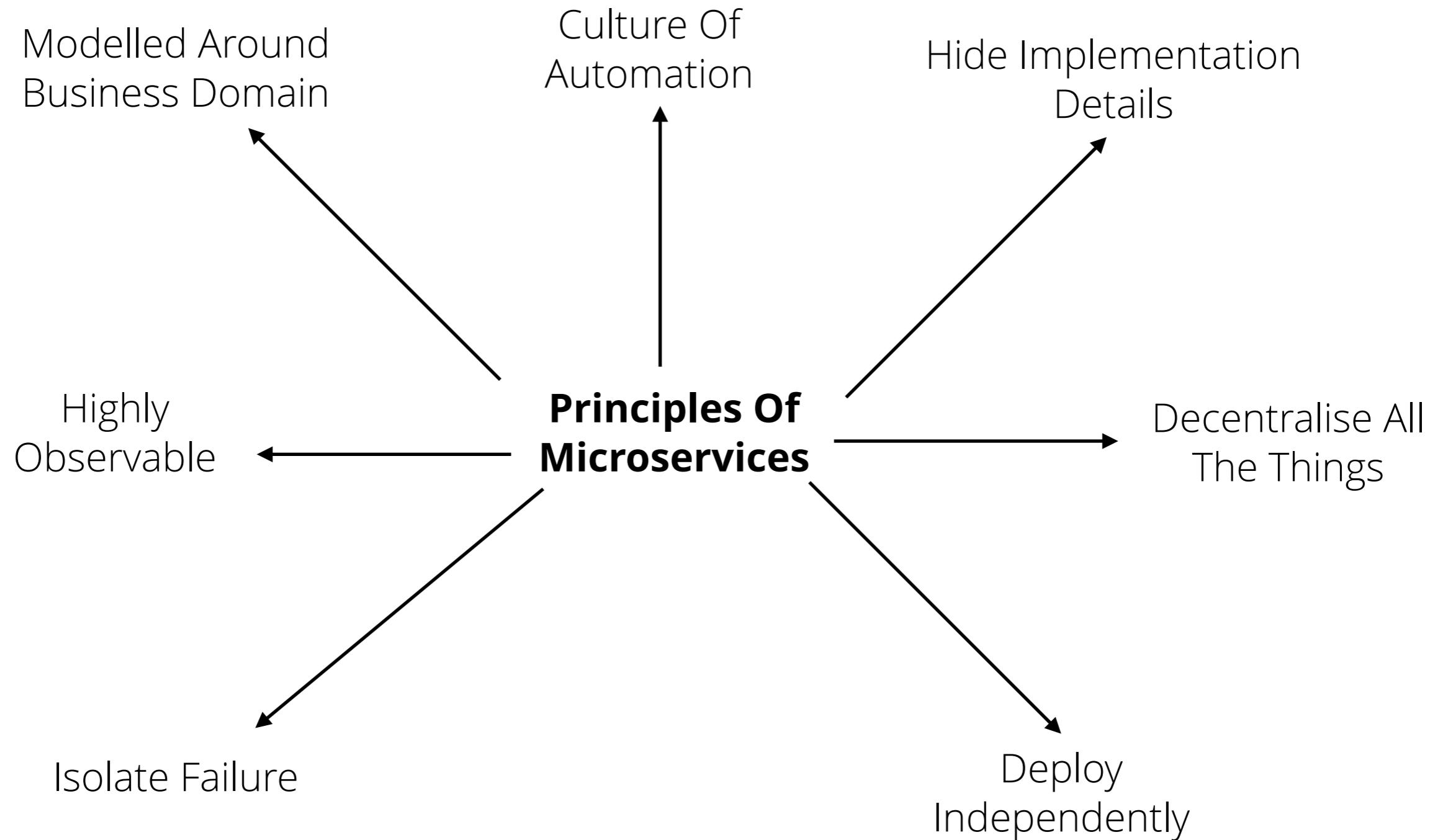
Hide Implementation
Details

Principles Of Microservices

Decentralise All
The Things







THANKS!

Sam Newman
@samnewman

ThoughtWorks®