

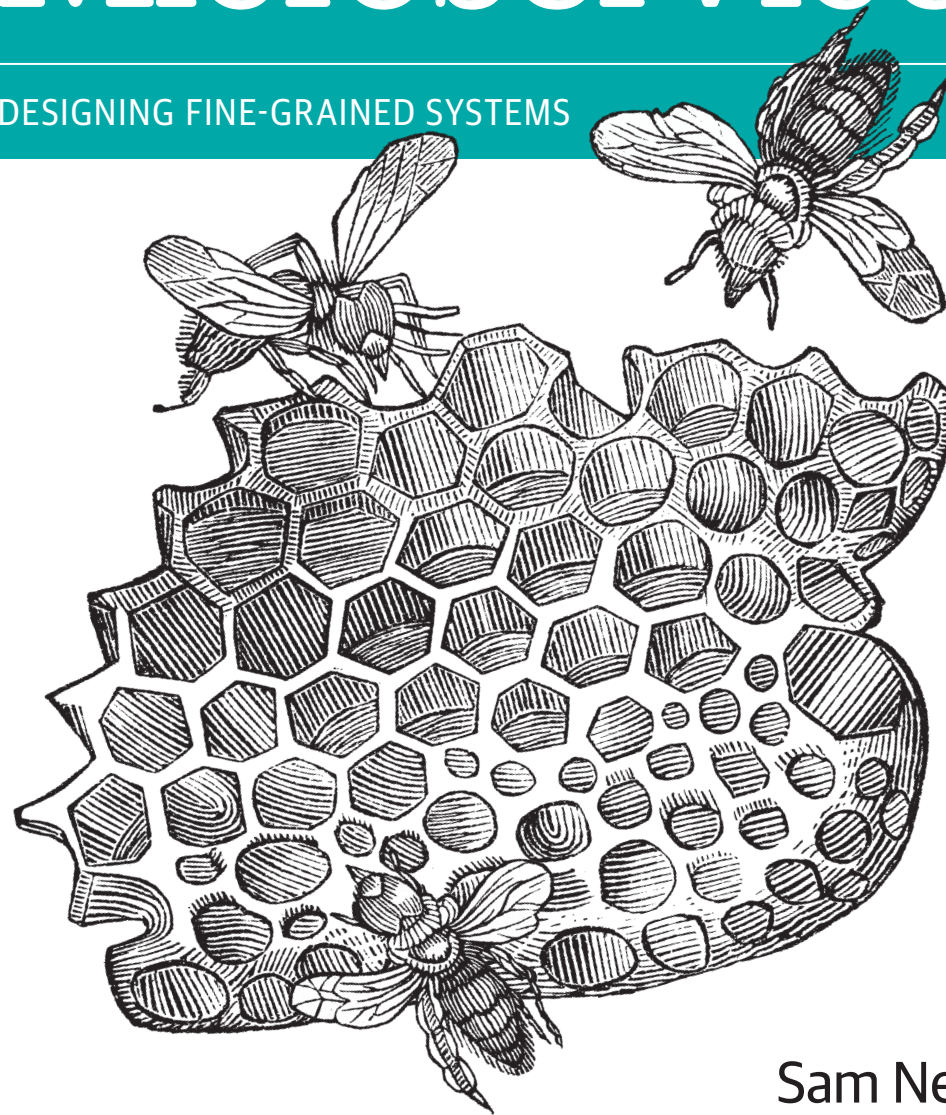
Confusion in the land of the serverless

Sam Newman

O'REILLY®

Building Microservices

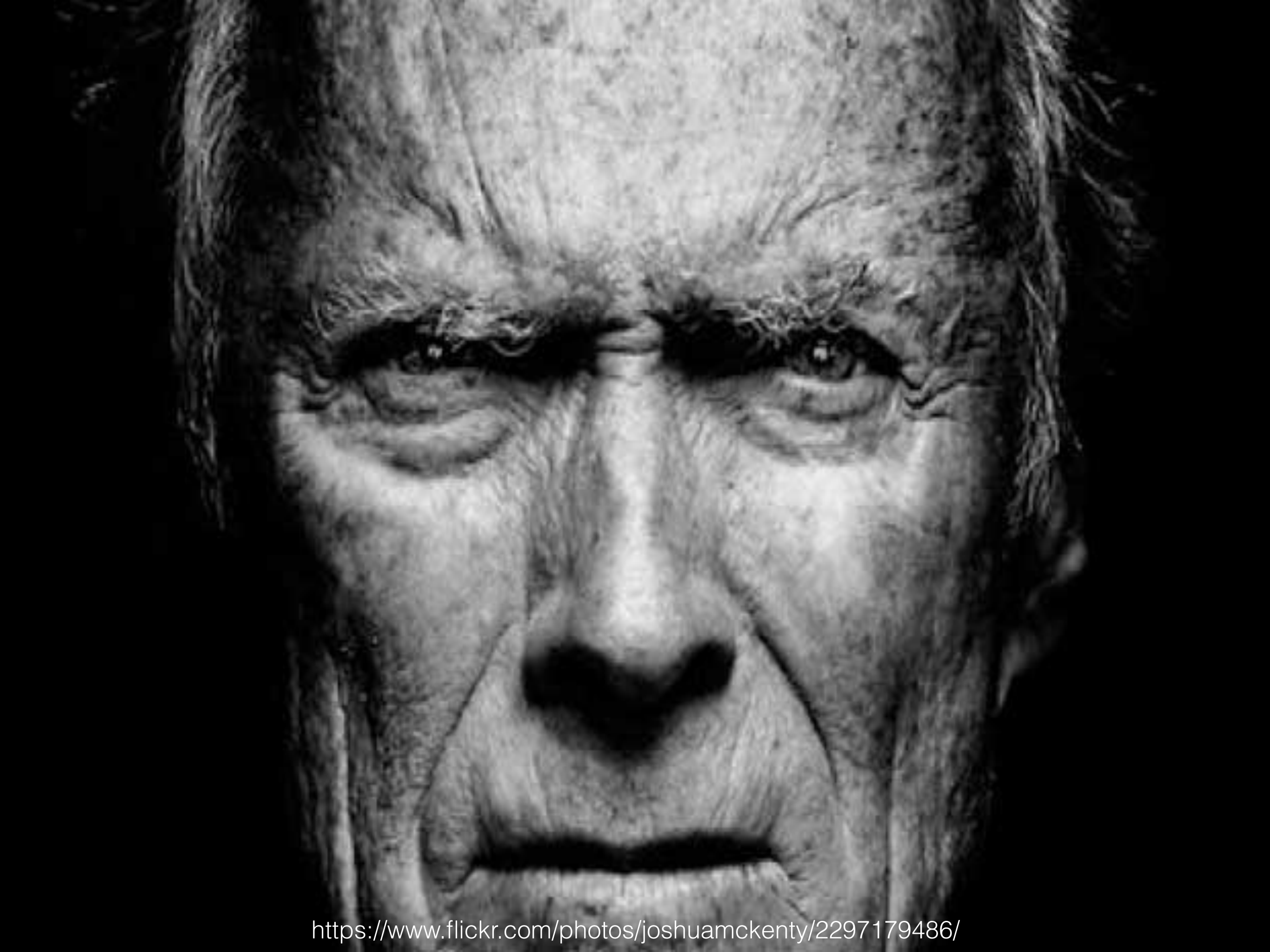
DESIGNING FINE-GRAINED SYSTEMS



Sam Newman

Sam Newman & Associates









[thefacebook]

[login](#) [register](#) [about](#)

Email:

Password:

register

login

Welcome to Thefacebook!

[Welcome to Thefacebook]

Thefacebook is an online directory that connects people through social networks at colleges.

We have opened up Thefacebook for popular consumption at **Harvard University**.

You can use Thefacebook to:

- Search for people at your school
- Find out who are in your classes
- Look up your friends' friends
- See a visualization of your social network

To get started, click below to register. If you have already registered, you can log in.

Register

Login

[about](#) [contact](#) [faq](#) [terms](#) [privacy](#)

a Mark Zuckerberg production

Thefacebook © 2004

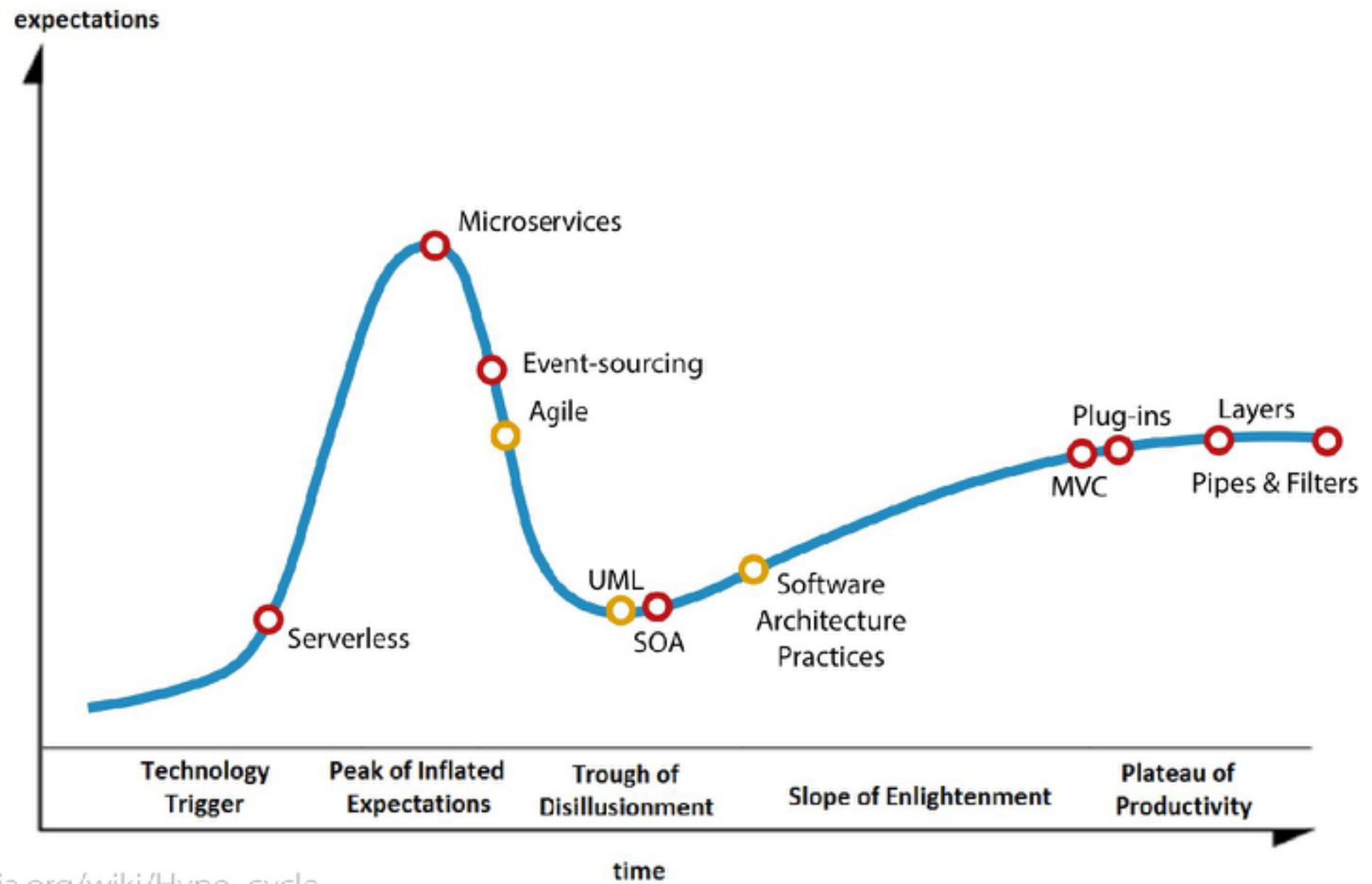






#serverless

Gartner Hype Cycle



https://en.wikipedia.org/wiki/Hype_cycle

6



There is NO CLOUD, just other people's computers

rwLABS

The Internet of Things Accelerator
SAN FRANCISCO | HONG KONG | SHENZHEN

BUILDING AN
IOT STARTUP?

Apply Now

Why The Future Of Software And Apps Is Serverless

Posted on October 15, 2012 in [CLOUD](#)



KEN FROMM

Contributing Writer

207
Shares

f
7

t
17

in
101

<http://readwrite.com/2012/10/15/why-the-future-of-software-and-apps-is-serverless/>

The logo for RW Labs, featuring a stylized 'RW' in red and 'LABS' in white.

The Internet of Things Accelerator
SAN FRANCISCO | HONG KONG | SHENZHEN

BUILDING AN
IOT STARTUP?

Apply Now

The phrase “serverless” doesn’t mean servers are no longer involved. It simply means that developers no longer have to think that much about them. Computing resources get used as services without having to manage around physical capacities or limits.



<http://readwrite.com/2012/10/15/why-the-future-of-software-and-apps-is-serverless/>



lambda launched
in 2014

Platform

WOW!

Platform

Yay, serverless!

WOW!

Platform

Yay, serverless!

WOW!

So abstract

Platform

Yay, serverless!

WOW!

So abstract

Much PAYG

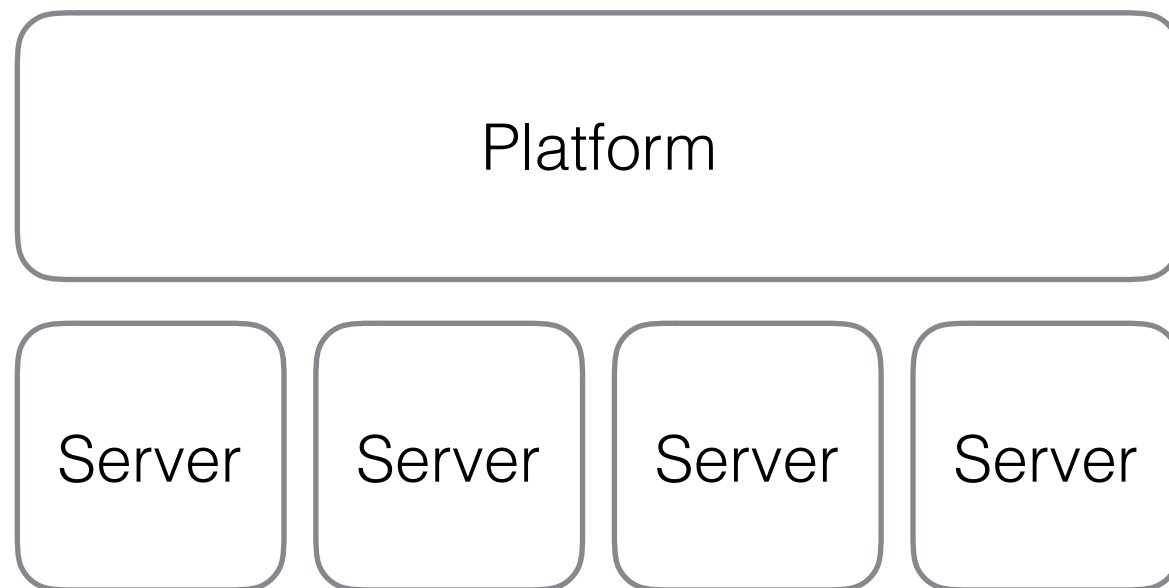
Platform

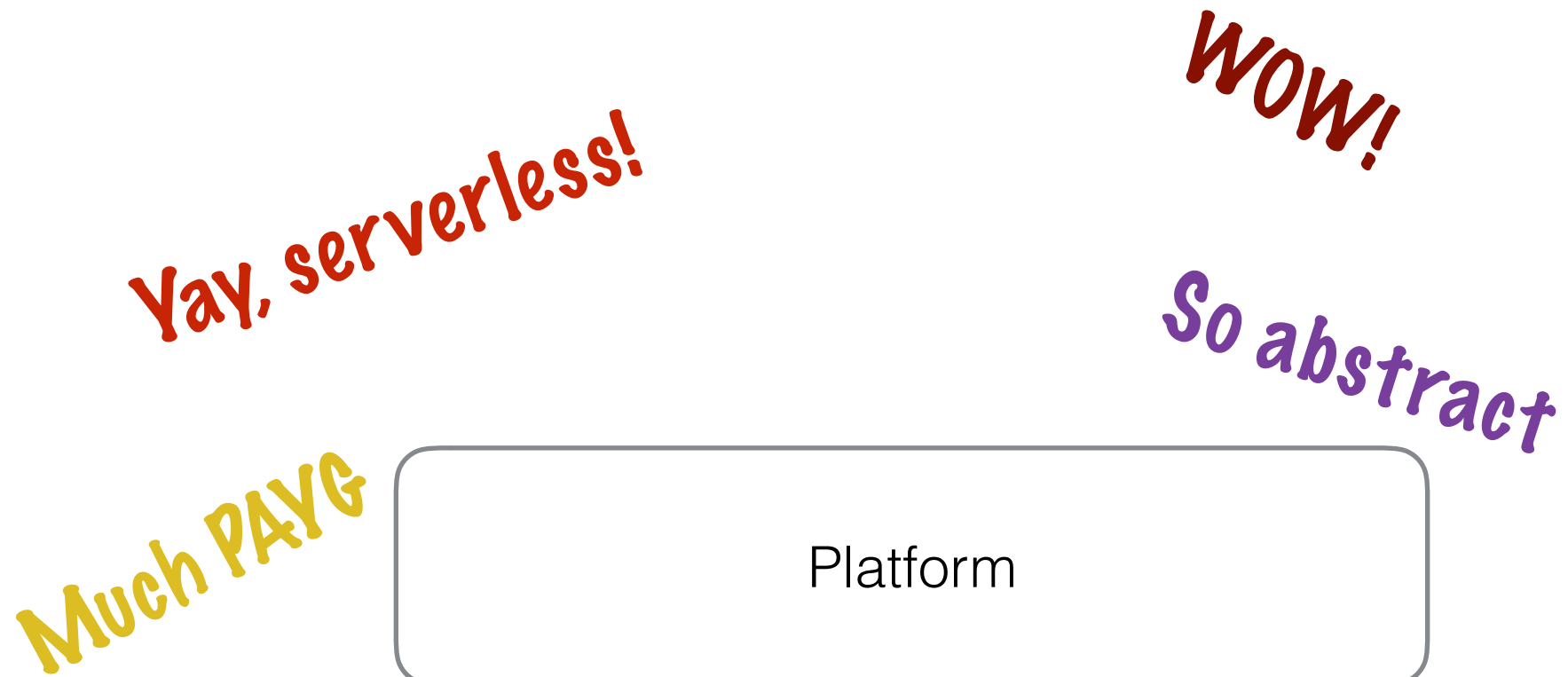
Yay, serverless!

WOW!

So abstract

Much PAYG





**WHAT DO YOU MEAN THE BOAT IS
SINKING?**



MY END IS UP 200 FEET

Serverless Architectures

Serverless architectures refer to applications that significantly depend on third-party services (known as Backend as a Service or "BaaS") or on custom code that's run in ephemeral containers (Function as a Service or "FaaS"), the best known vendor host of which currently is AWS Lambda. By using these ideas, and by moving much behavior to the front end, such architectures remove the need for the traditional 'always on' server system sitting behind an application. Depending on the circumstances, such systems can significantly reduce operational cost and complexity at a cost of vendor dependencies and (at the moment) immaturity of supporting services.

04 August 2016



Mike Roberts

Mike is an engineering leader living in New York City. While spending much of his time these days managing people

and teams he also still gets to code occasionally, especially in Clojure, and has Opinions about software architecture. He is cautiously optimistic that Serverless architectures may be worth some of the hype that they are currently receiving.

Find **similar articles** at the tag:
[application architecture](#)

Contents

expand

What is Serverless?

- A couple of examples
- Unpacking 'Function as a Service'
- What isn't Serverless?

Benefits

- Reduced operational cost
- BaaS - reduced development cost
- FaaS - scaling costs
- Easier Operational Management
- 'Greener' computing?

Drawbacks

- Inherent Drawbacks
- Implementation Drawbacks

The Future of Serverless

- Mitigating the Drawbacks
- The emergence of patterns

<https://martinfowler.com/articles/serverless.html>

Mike Roberts' Definition Of Serverless

Mike Roberts' Definition Of Serverless

1. No management of server hosts or server processes

Mike Roberts' Definition Of Serverless

1. No management of server hosts or server processes
2. Self auto-scale and auto-provision based on load

Mike Roberts' Definition Of Serverless

1. No management of server hosts or server processes
2. Self auto-scale and auto-provision based on load
3. Costs based on precise usage

Mike Roberts' Definition Of Serverless

1. No management of server hosts or server processes
2. Self auto-scale and auto-provision based on load
3. Costs based on precise usage
4. Performance capabilities defined in terms other than host size/count

Mike Roberts' Definition Of Serverless

1. No management of server hosts or server processes
2. Self auto-scale and auto-provision based on load
3. Costs based on precise usage
4. Performance capabilities defined in terms other than host size/count
5. Implicit high availability



FAAS



FAAS



FAAS



BAAS





HEROKU



HEROKU





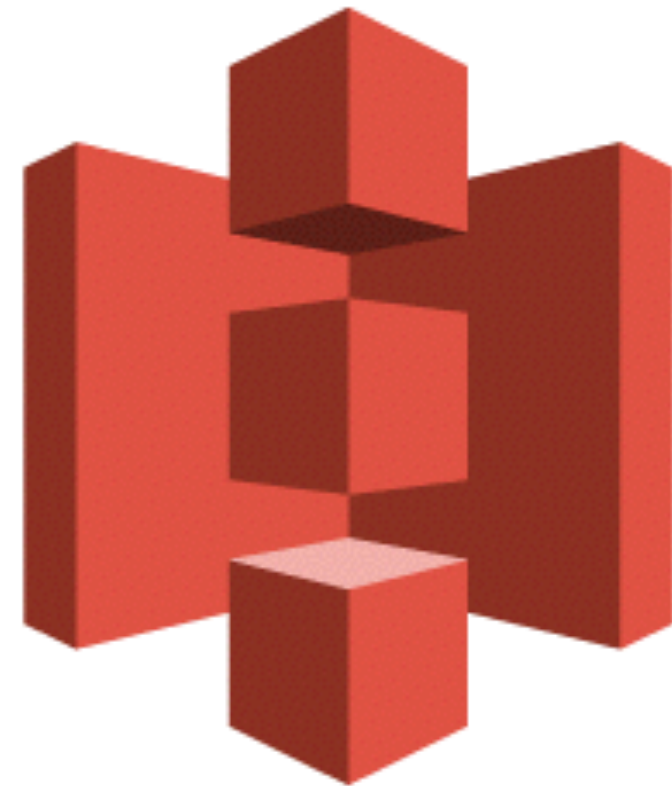


fastly[®]



force.com

fastly®



IAAS

CAAS



IAAS

PAAS

CAAS

IAAS

PAAS

FAAS

BAAAS

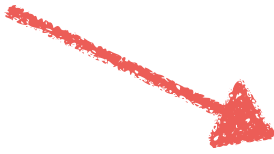
.....

CAAS

.....

IAAS

Serverless?



PAAS

FAAS

BAAAS

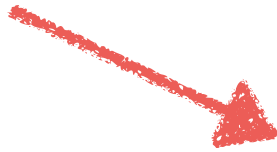


CAAS



IAAS

Serverless?



PAAS

FAAS

BAAAS

.....

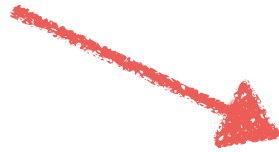
CAAS



.....

IAAS

Serverless?



PAAS

FAAS

BAAAS



CAAS



IAAS



BASS?

***AAS**



Kelsey Hightower ✓

@kelseyhightower

Following



I now understand what all the Serverless fuss is about. When you have a great idea the last thing you want to do is setup infrastructure.

RETWEETS

76

LIKES

157



11:22 PM - 23 Apr 2017



3



76



157

<https://twitter.com/kelseyhightower/status/856272003963039744>

The phrase “serverless” doesn’t mean servers are no longer involved. It simply means that developers no longer have to think that much about them.

<http://readwrite.com/2012/10/15/why-the-future-of-software-and-apps-is-serverless/>



Bilgin Ibryam

@bibryam

Follow



finally, a good explanation for the poorly named serverless concept by @jeffhollan: Functions is a programming model, but serverless is a billing model.

4:32 PM - 14 Jun 2018

37 Retweets 57 Likes



2



37

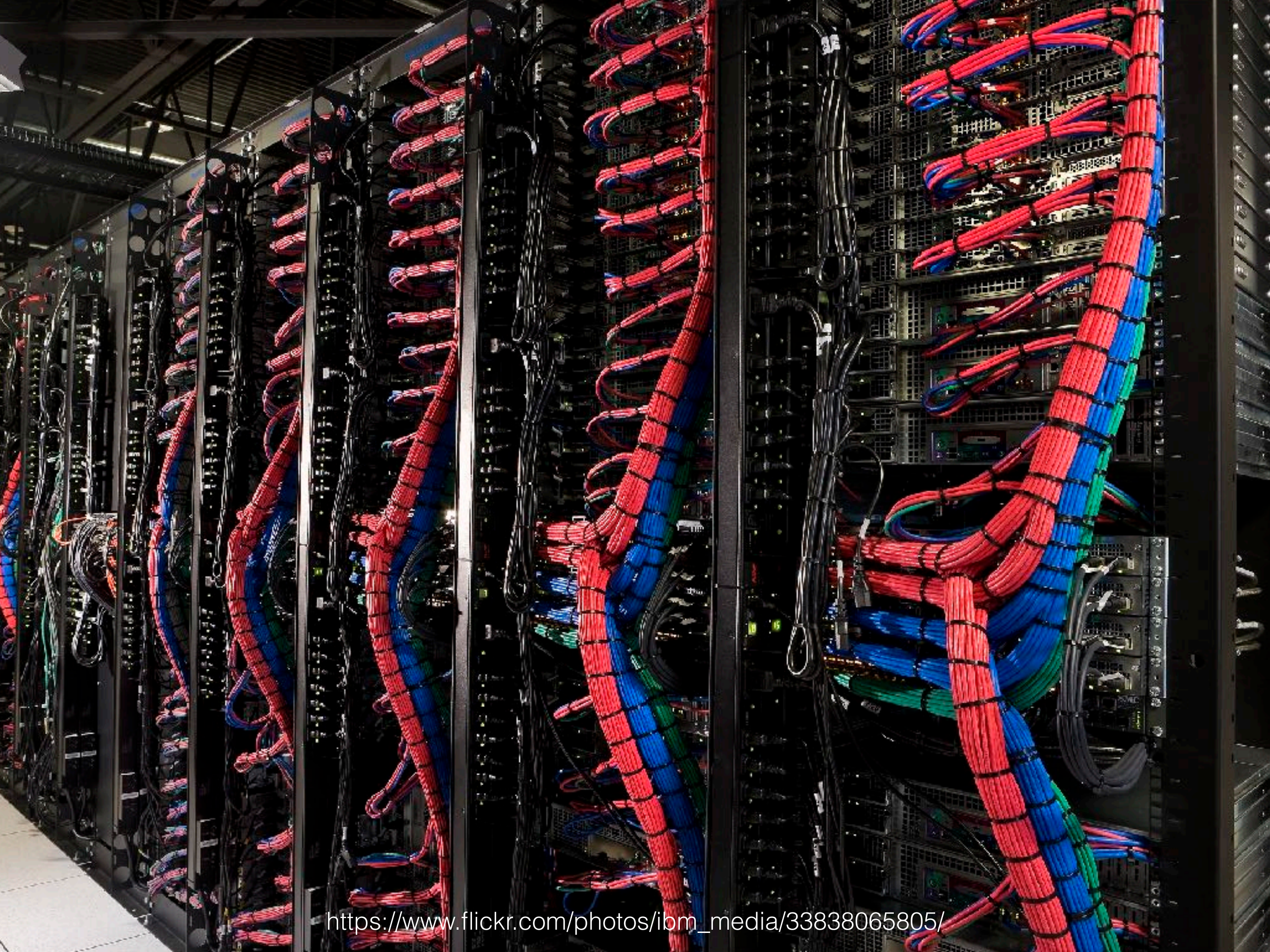


57



<https://twitter.com/bibryam/status/1007284710136000513>

Undifferentiated Heavy Lifting



“Developers turn caffeine into abstractions”

- *Brian Marick, possibly*

Machine Code

Assembly Code

Machine Code

Application Code

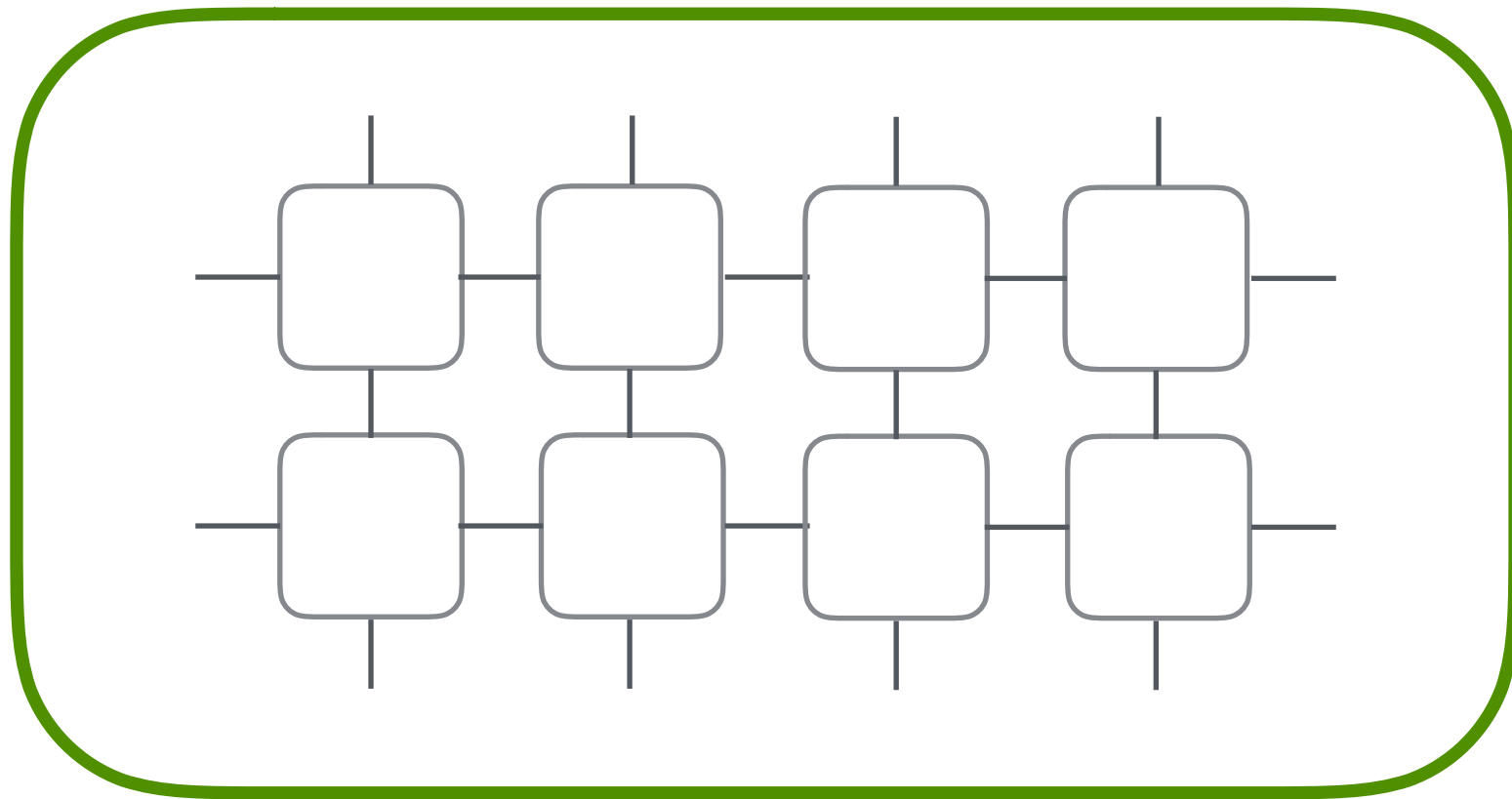
Assembly Code

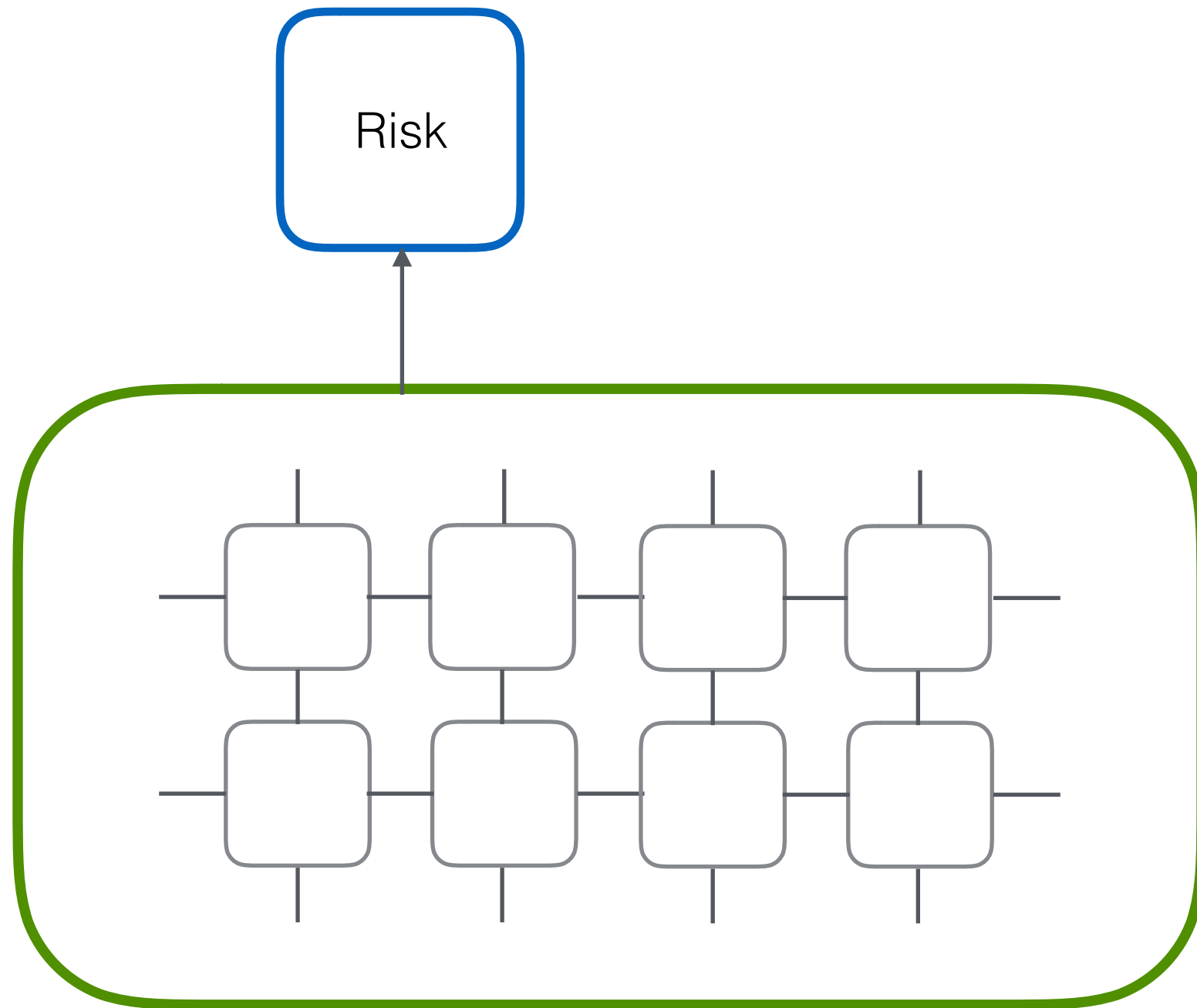
Machine Code

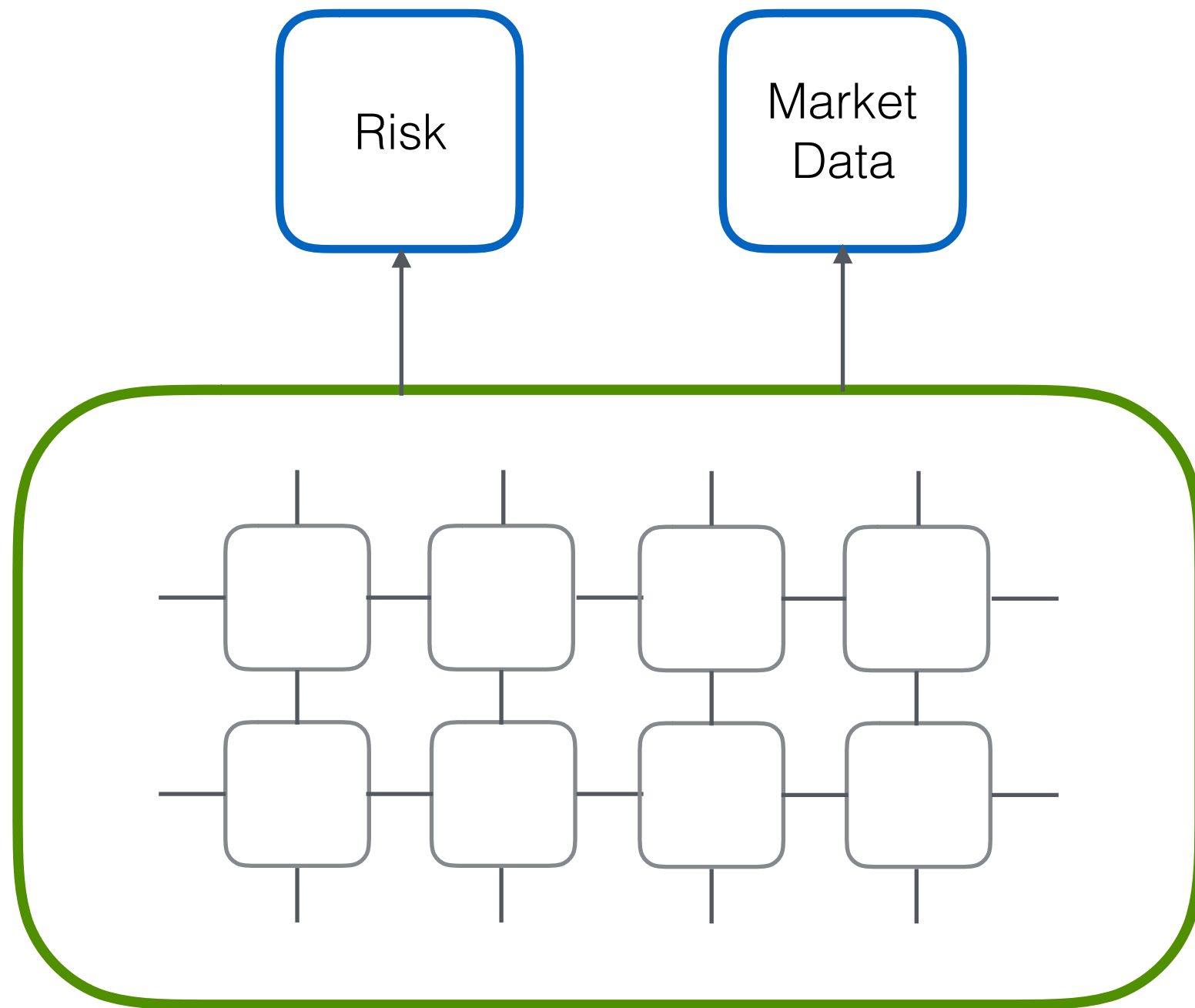
Resiliency

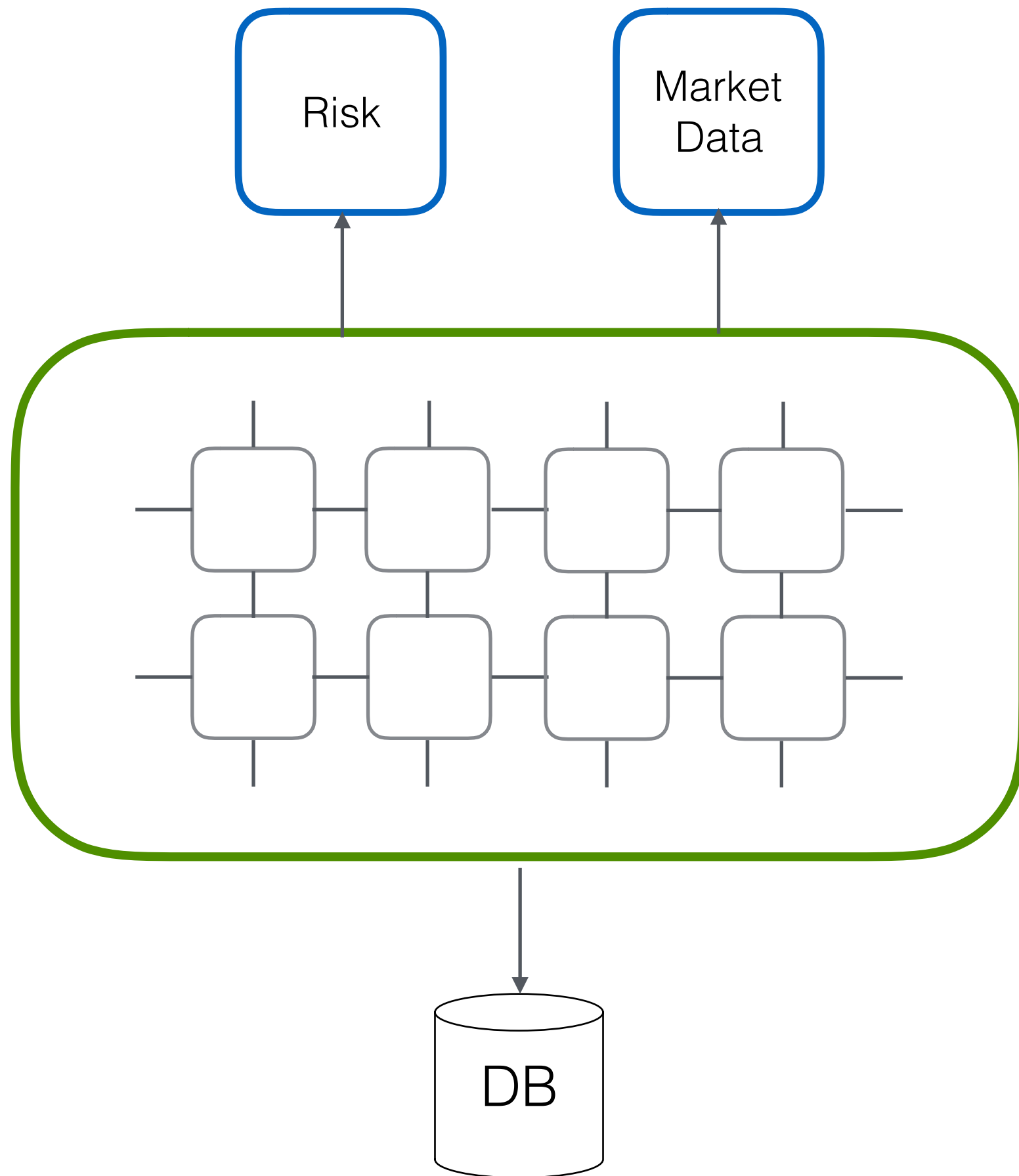




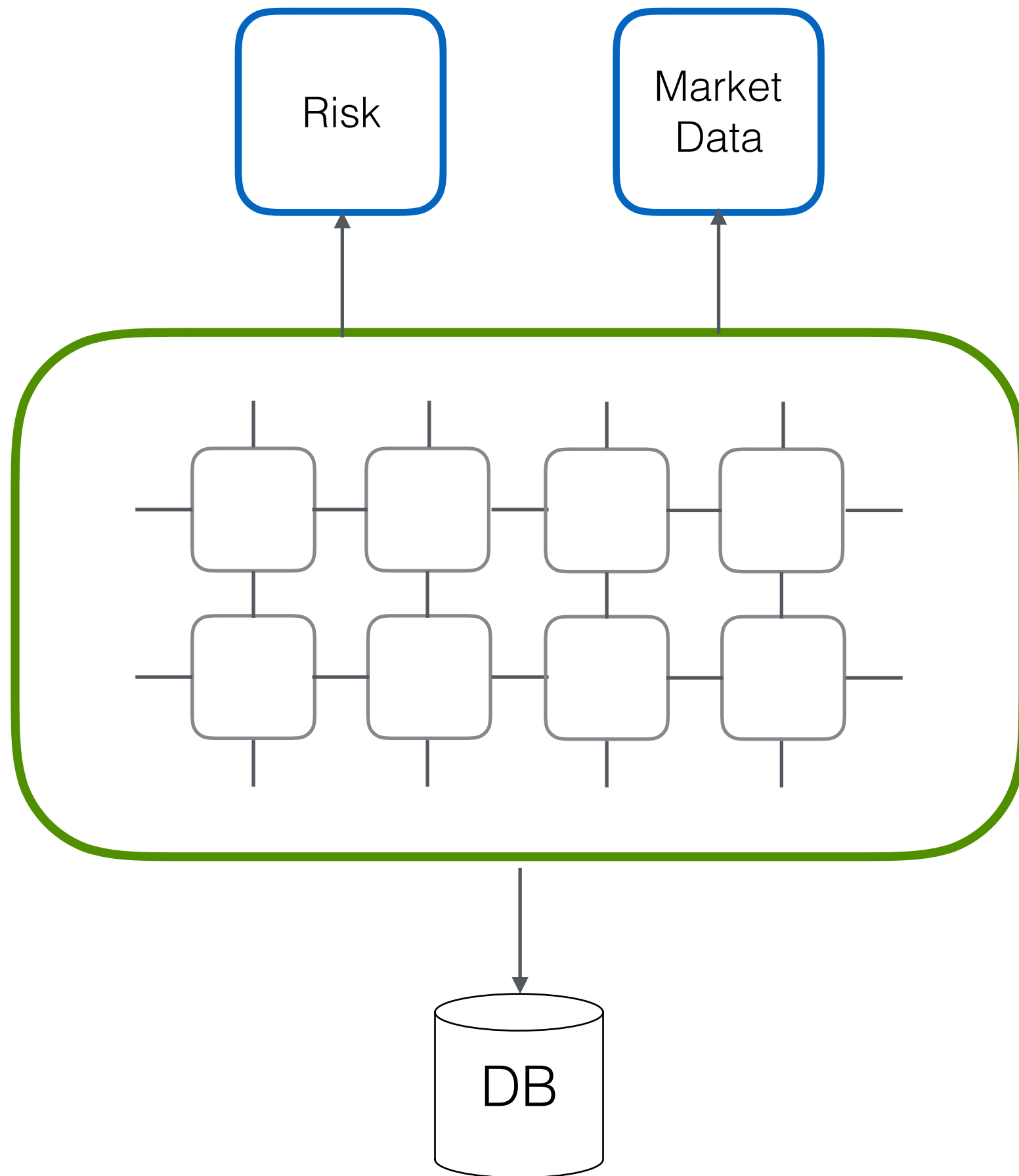


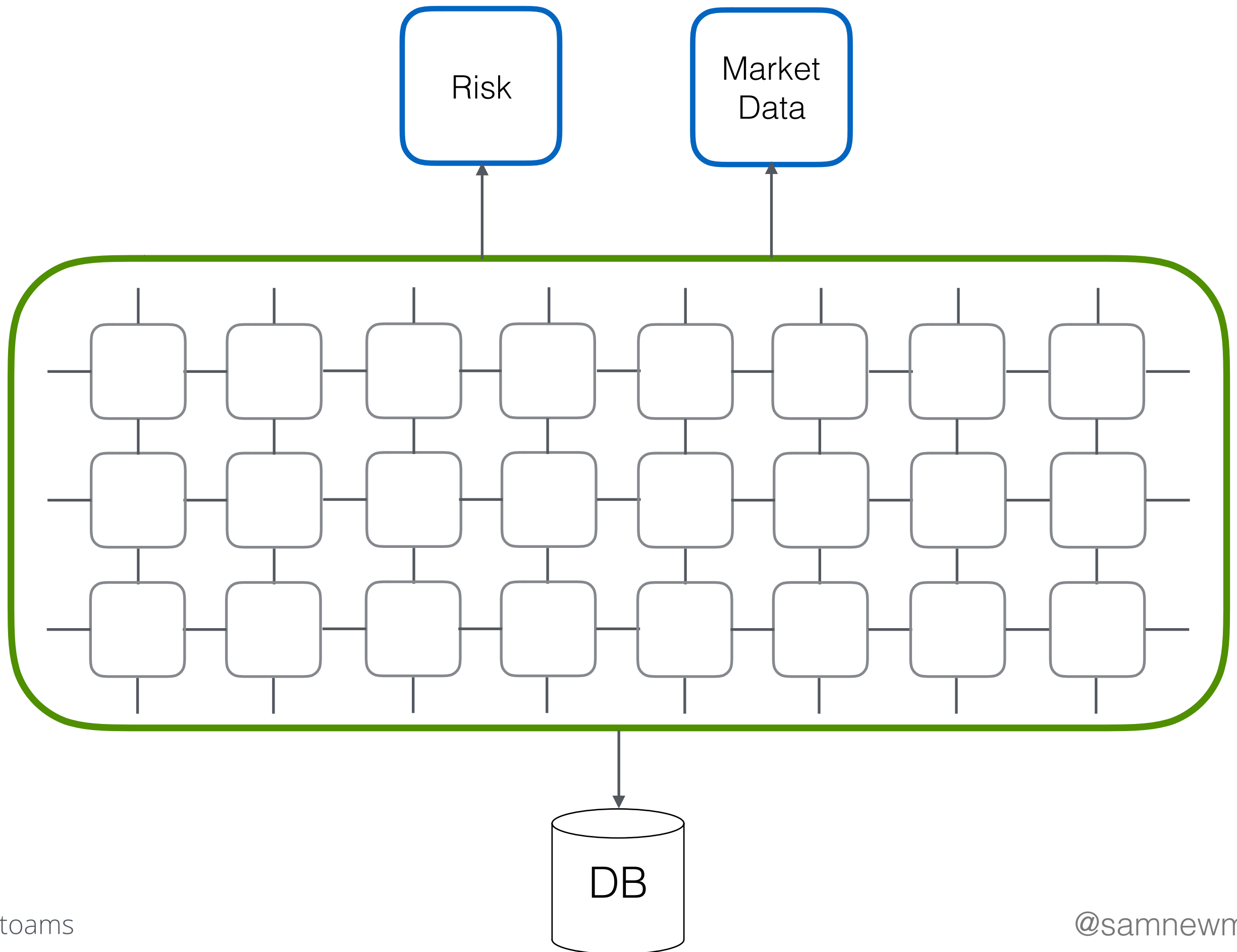


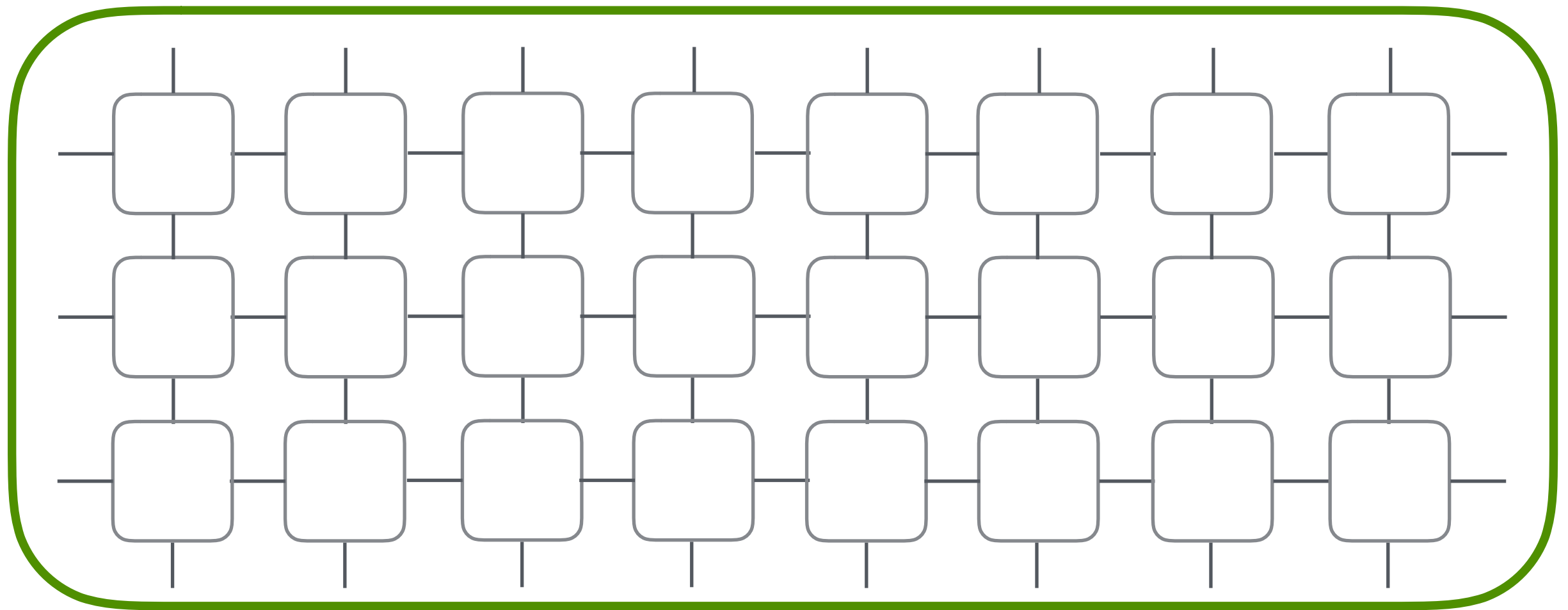


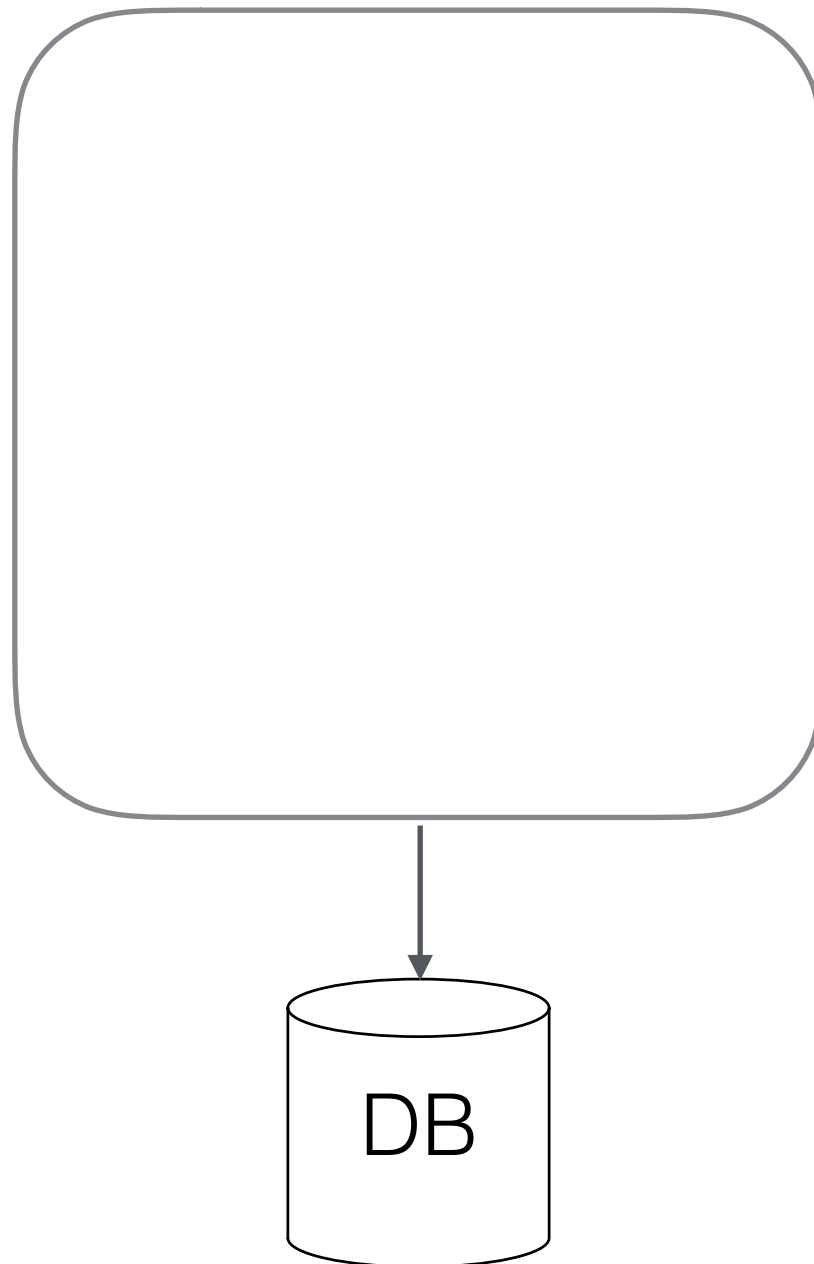


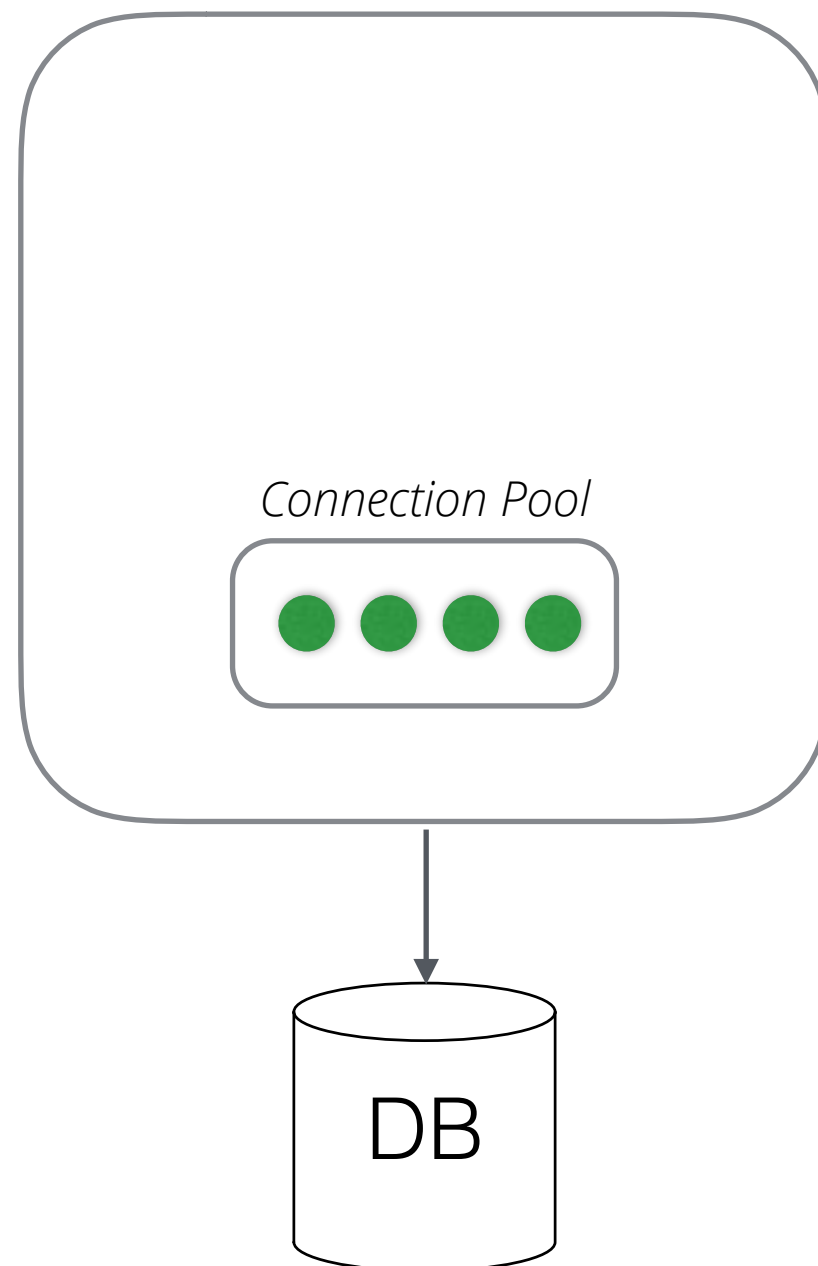


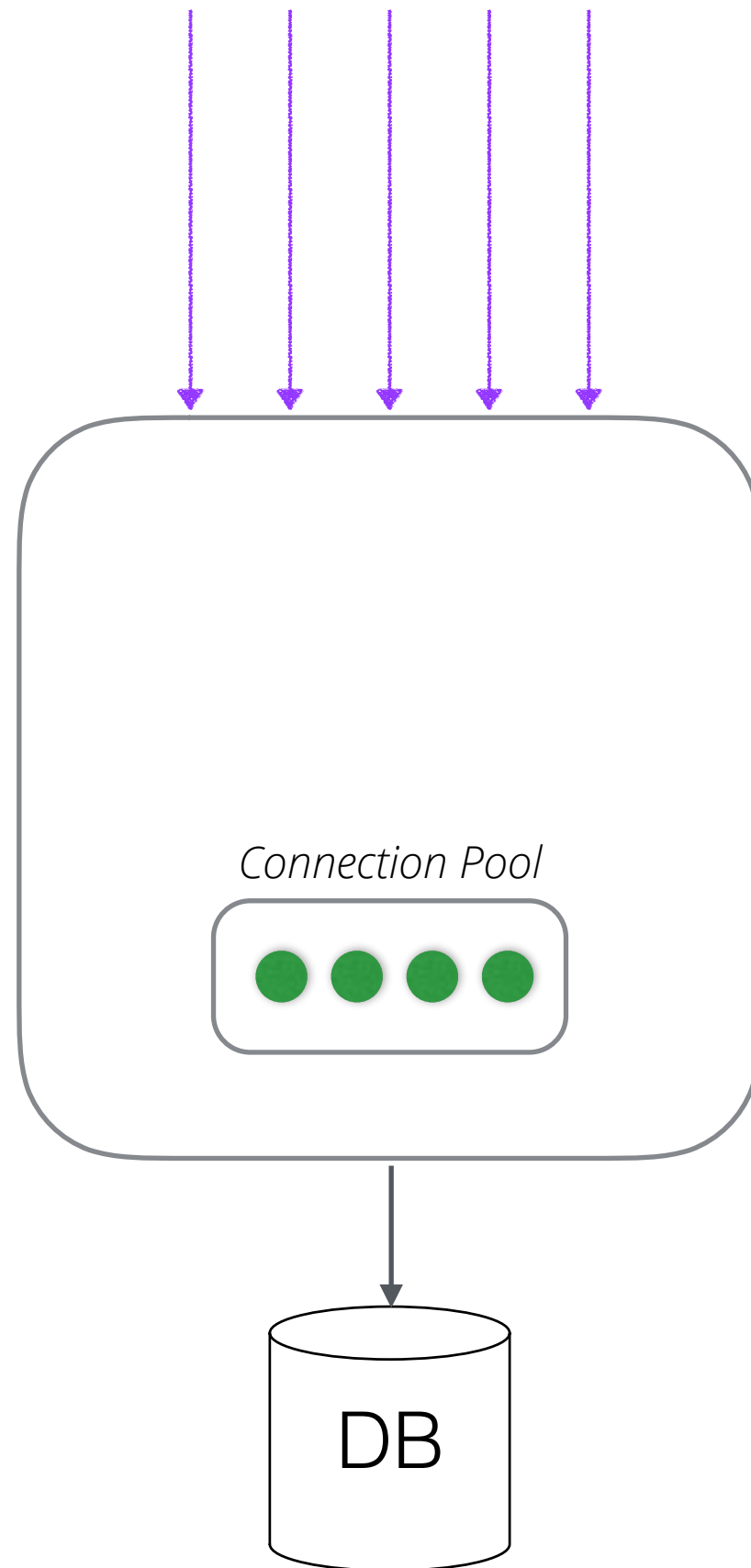


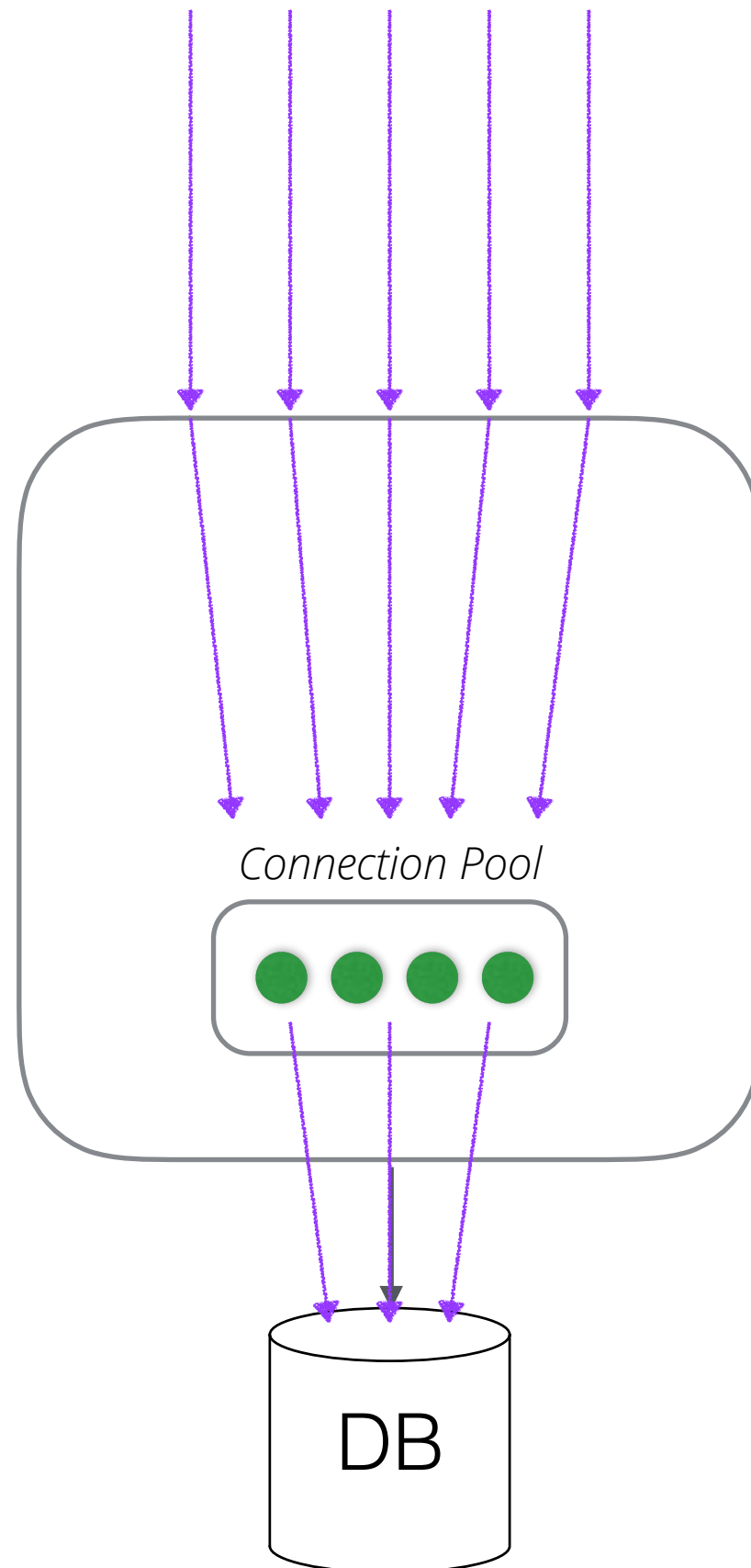




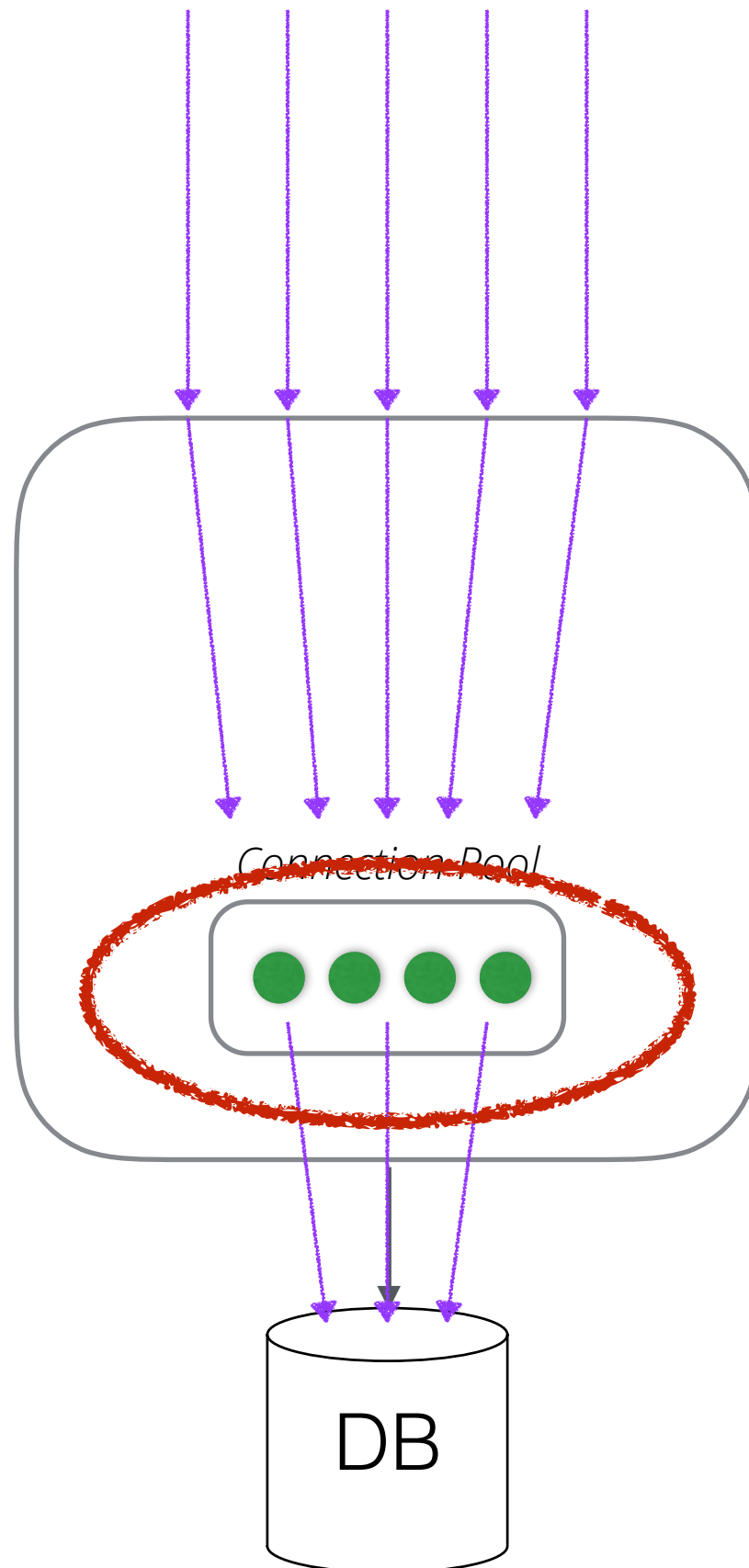


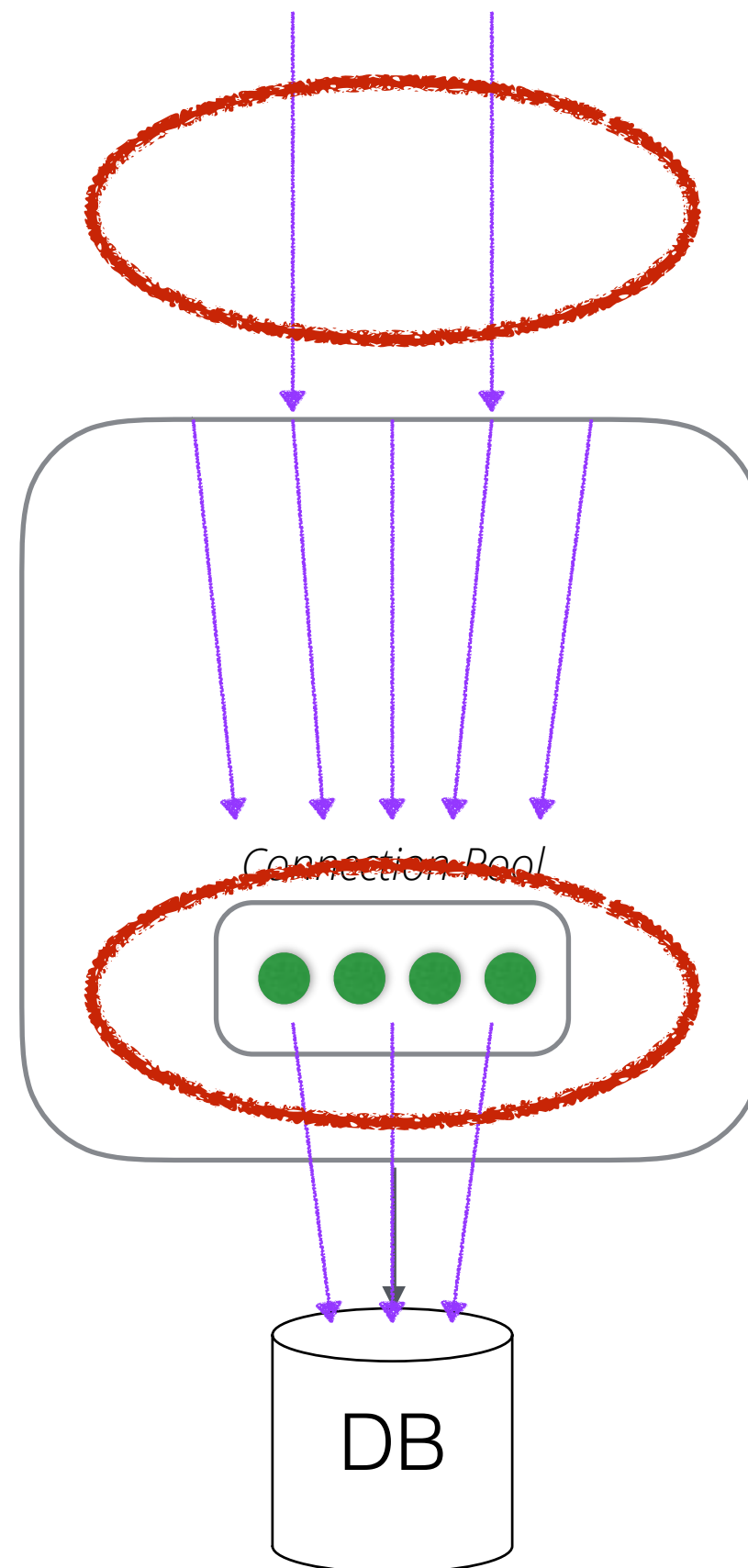






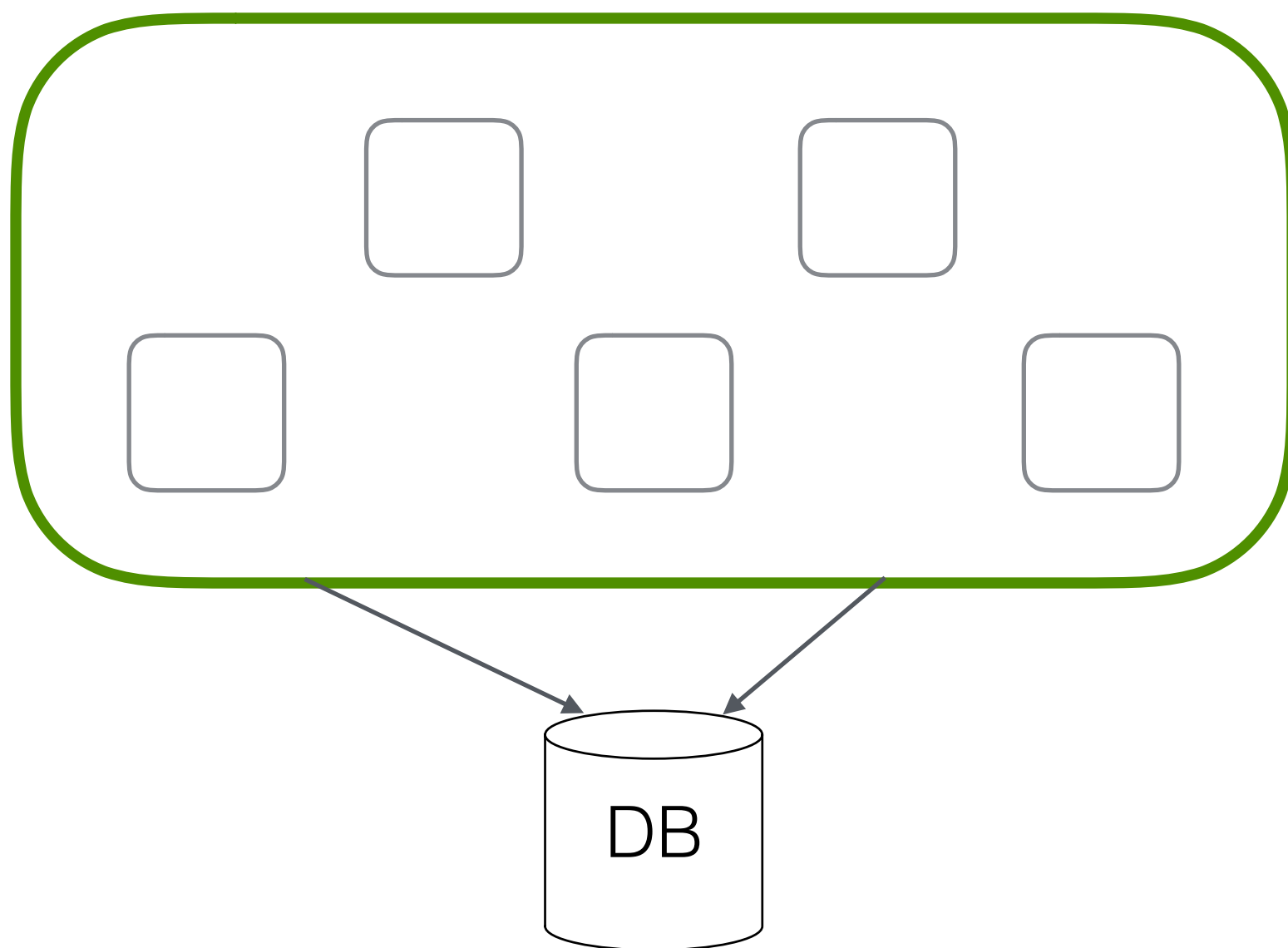
Connection pools
throttle load

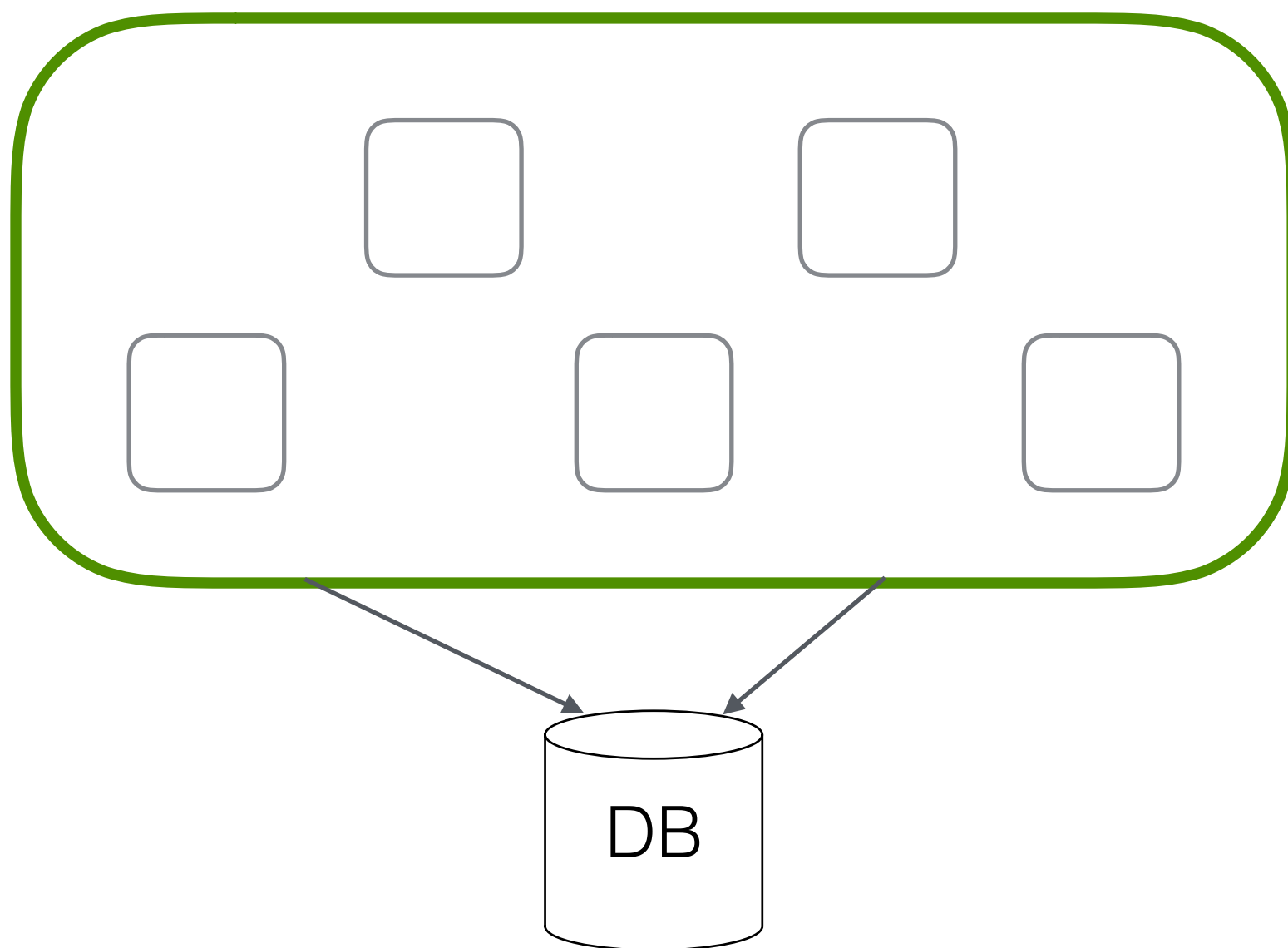


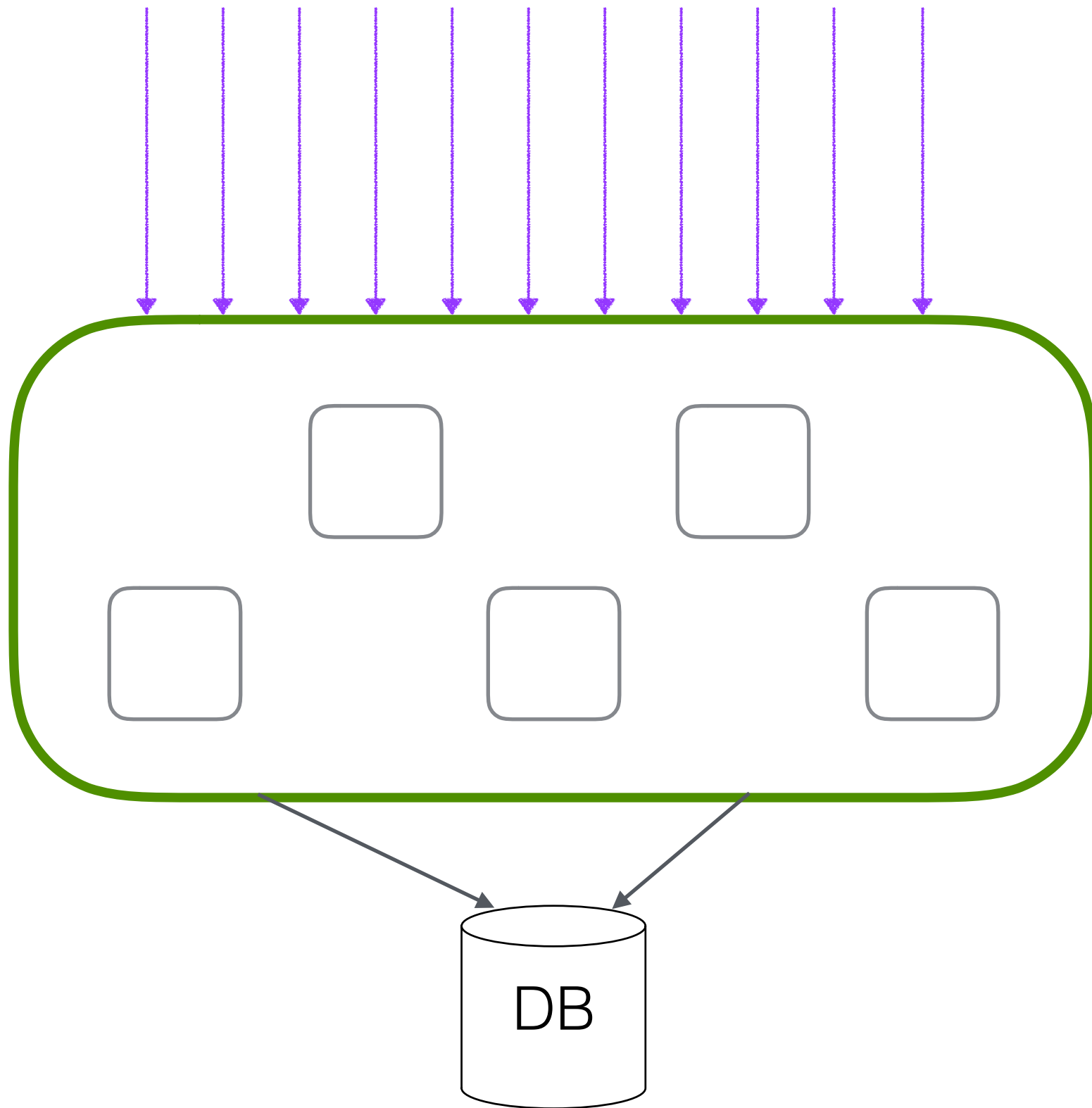


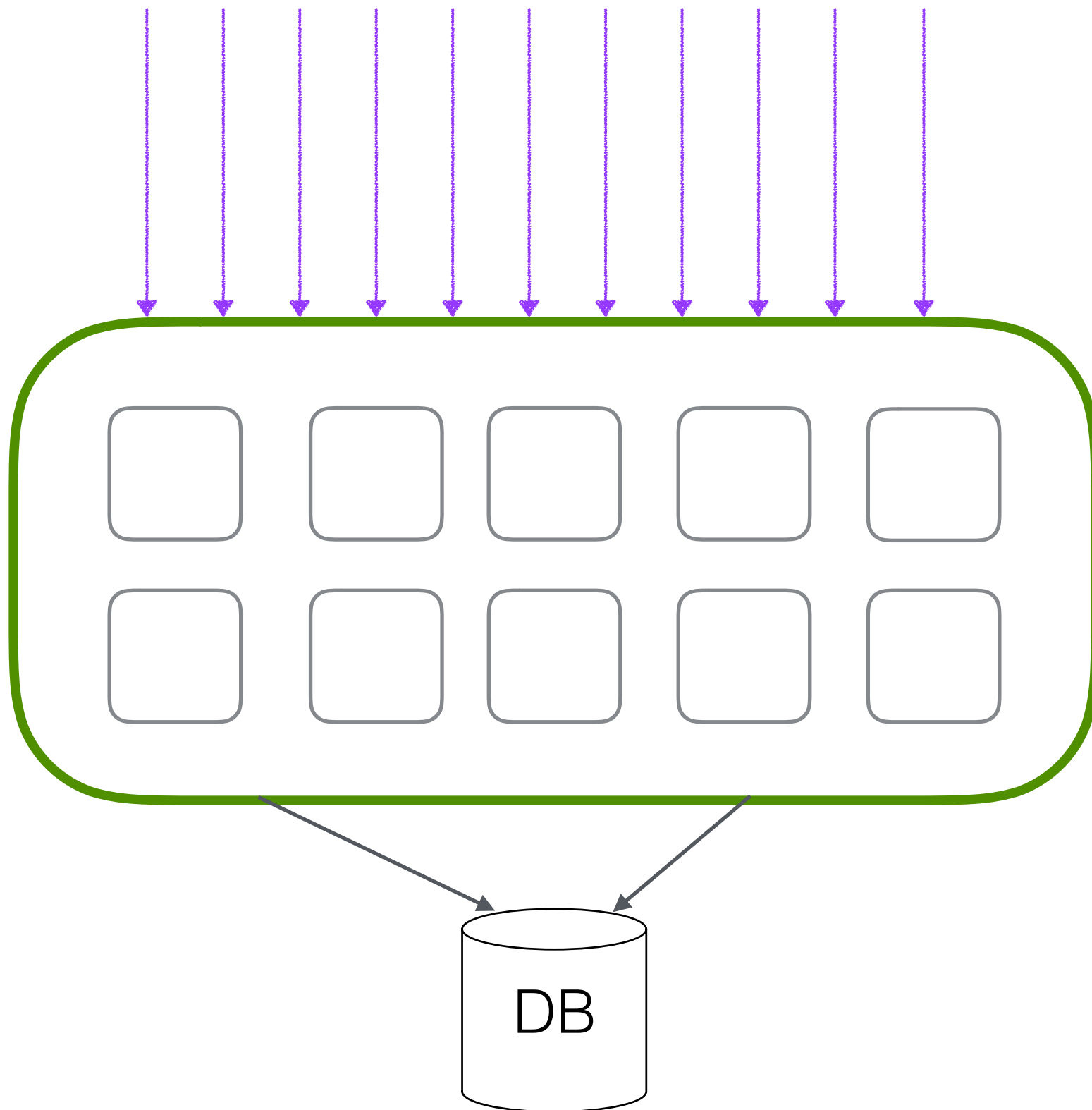
And allow for
load shedding

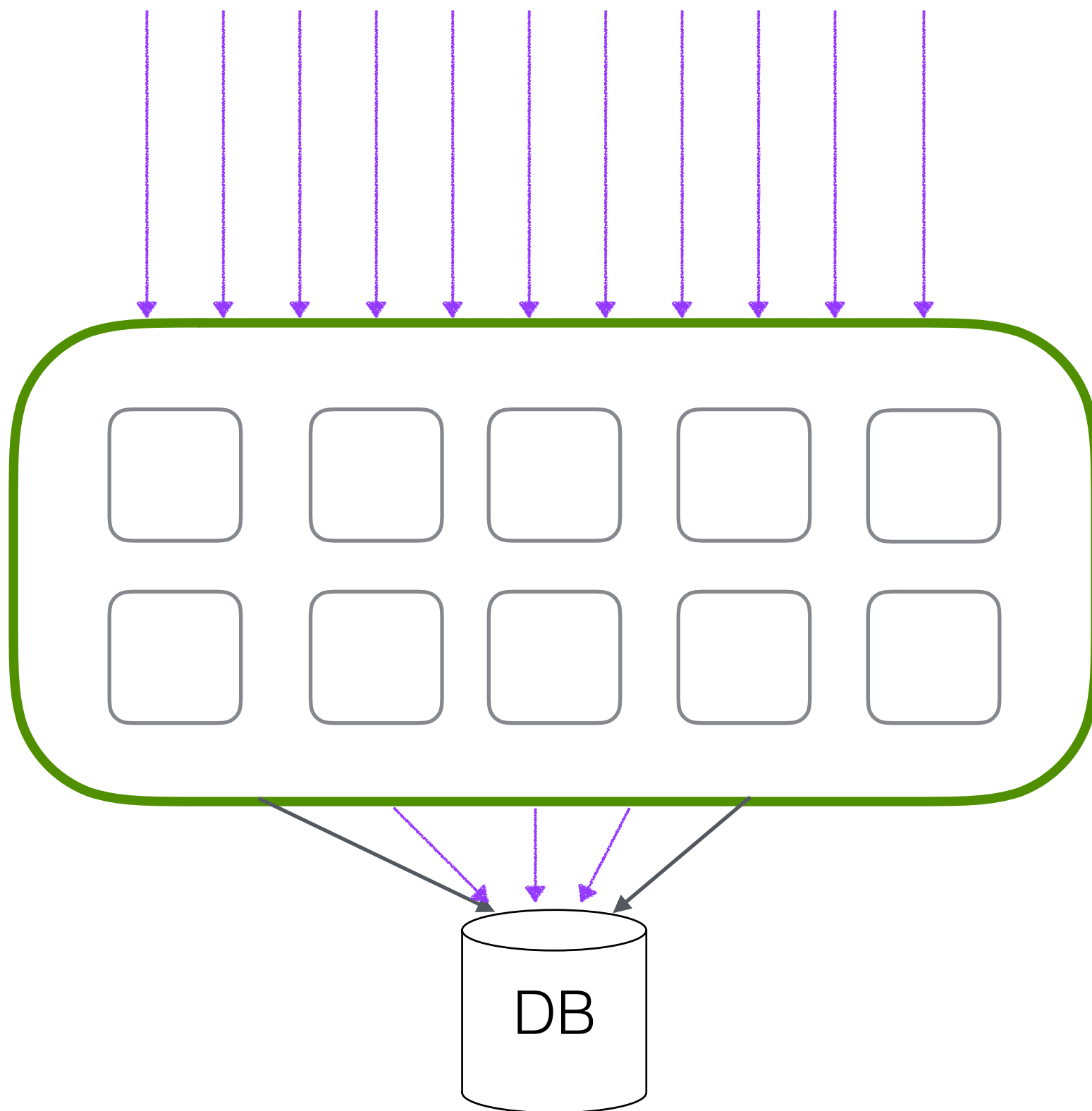
Connection pools
throttle load

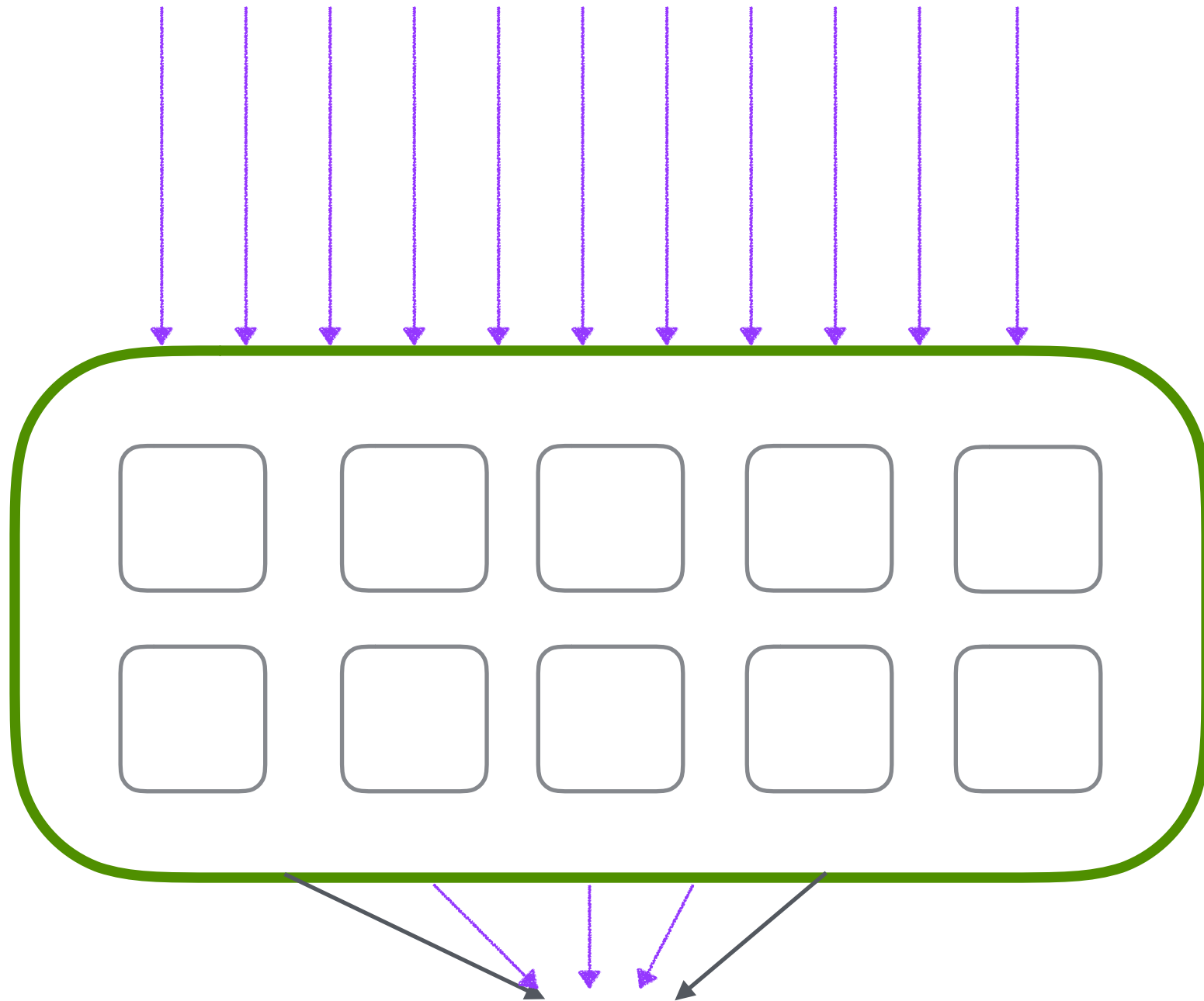






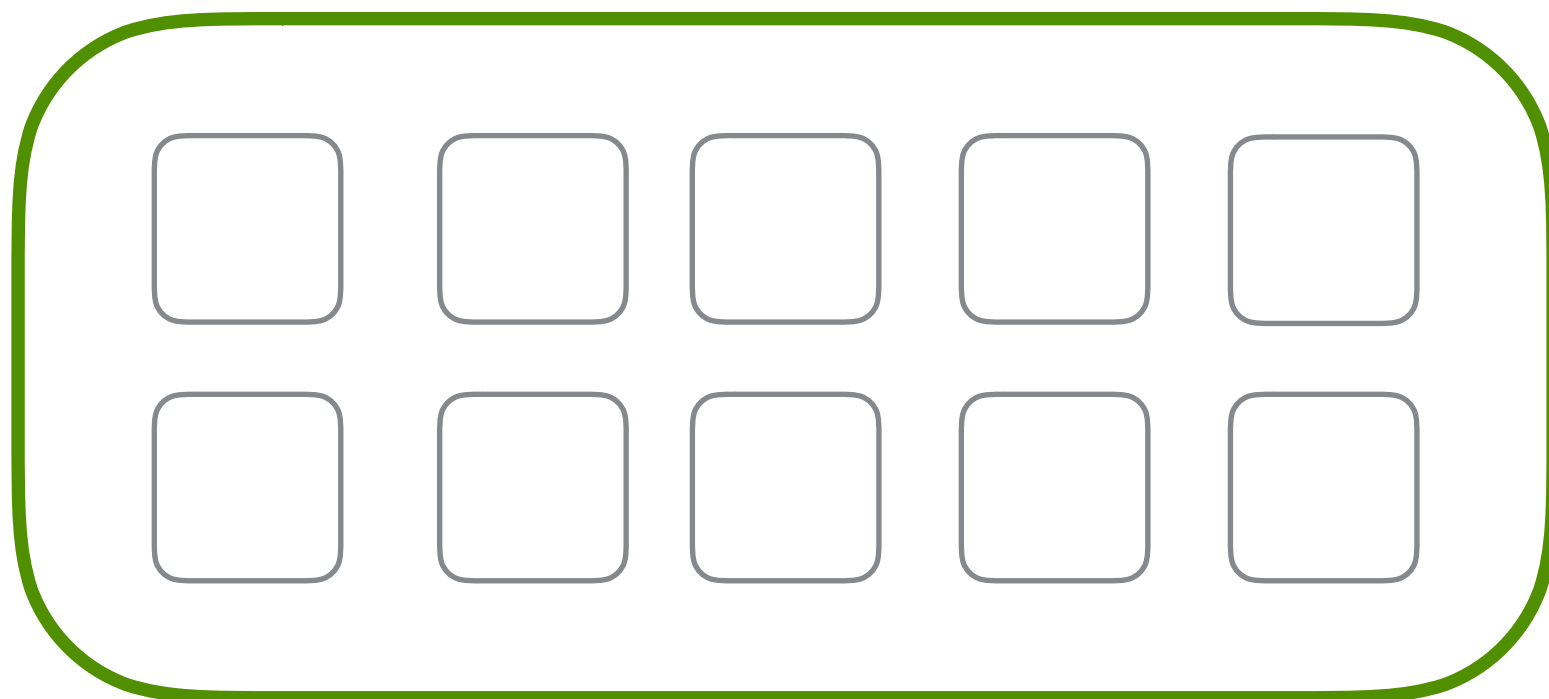


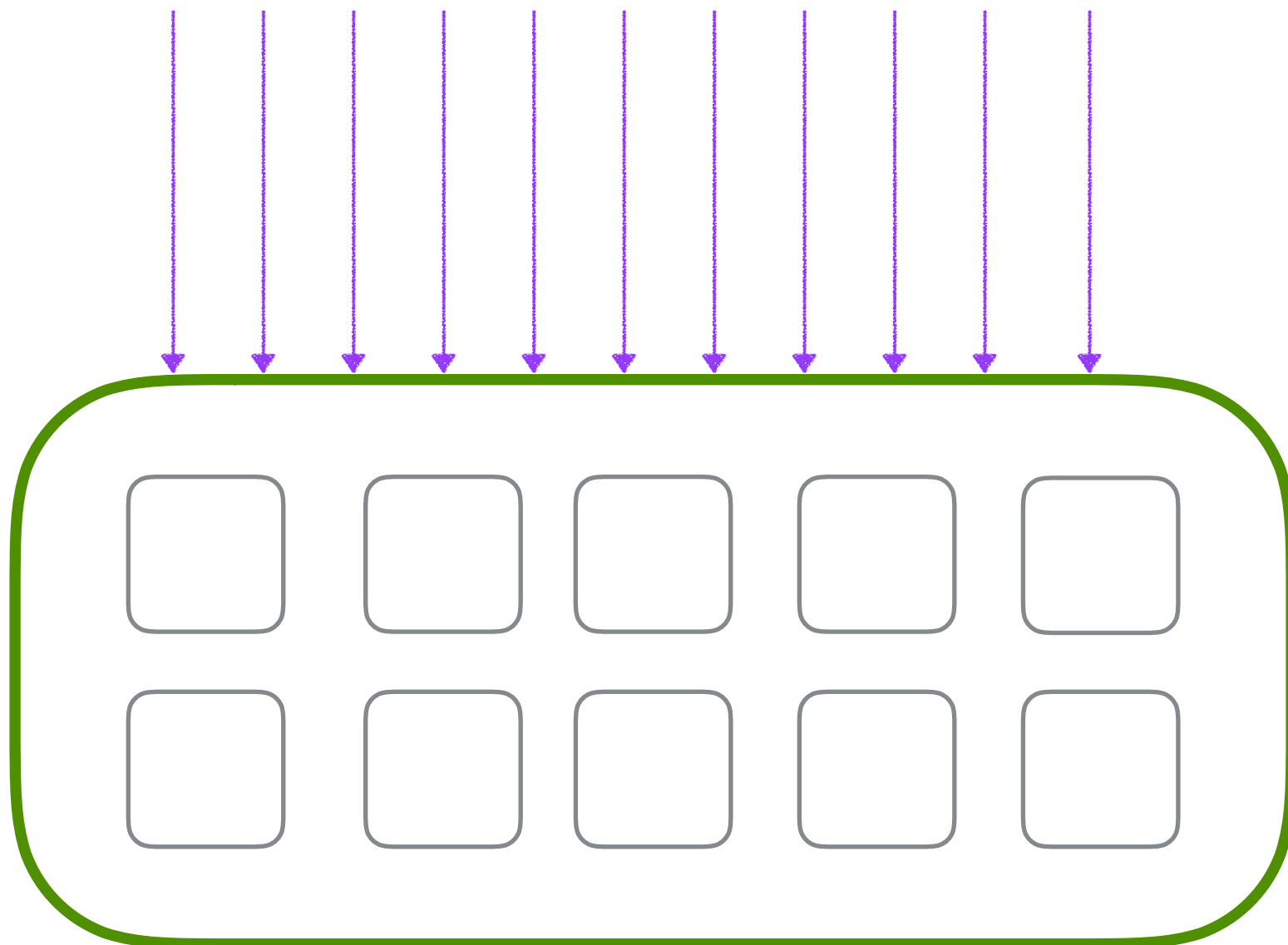






<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



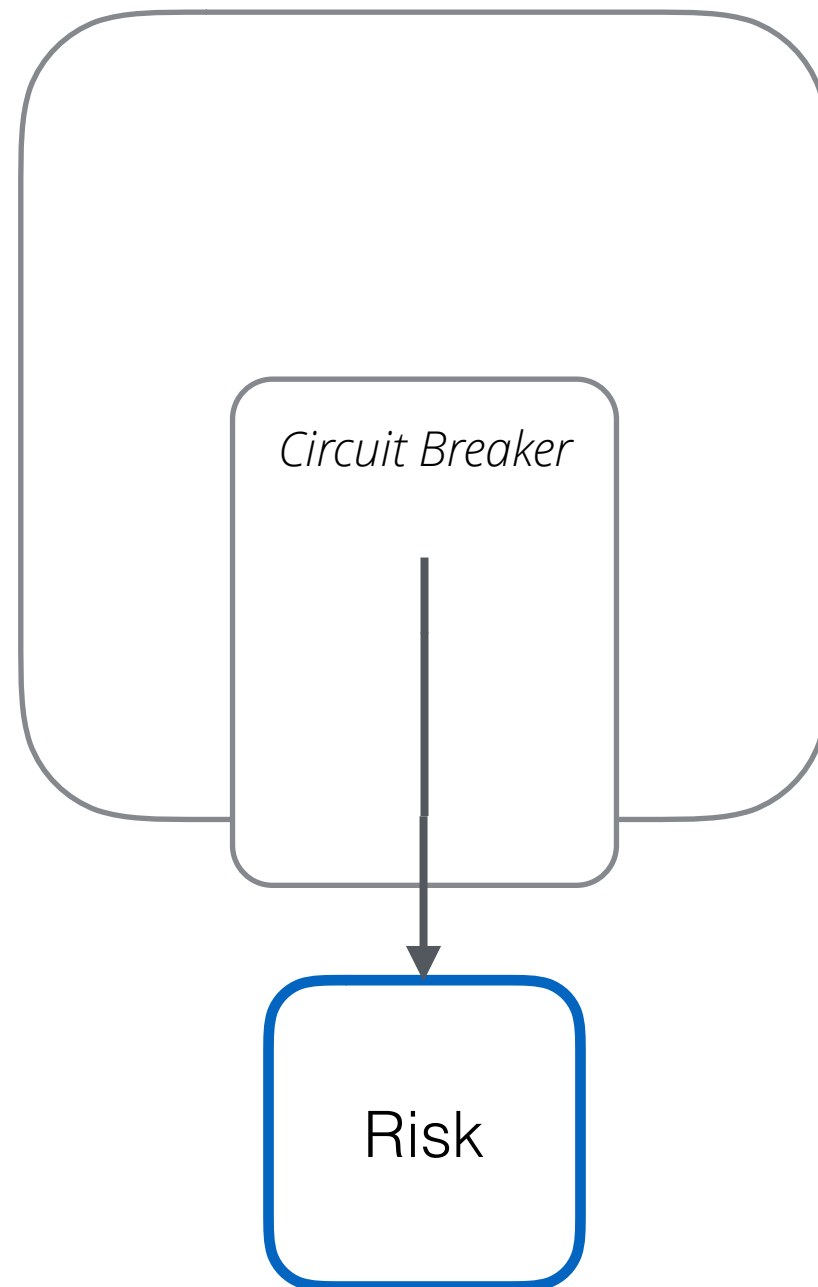


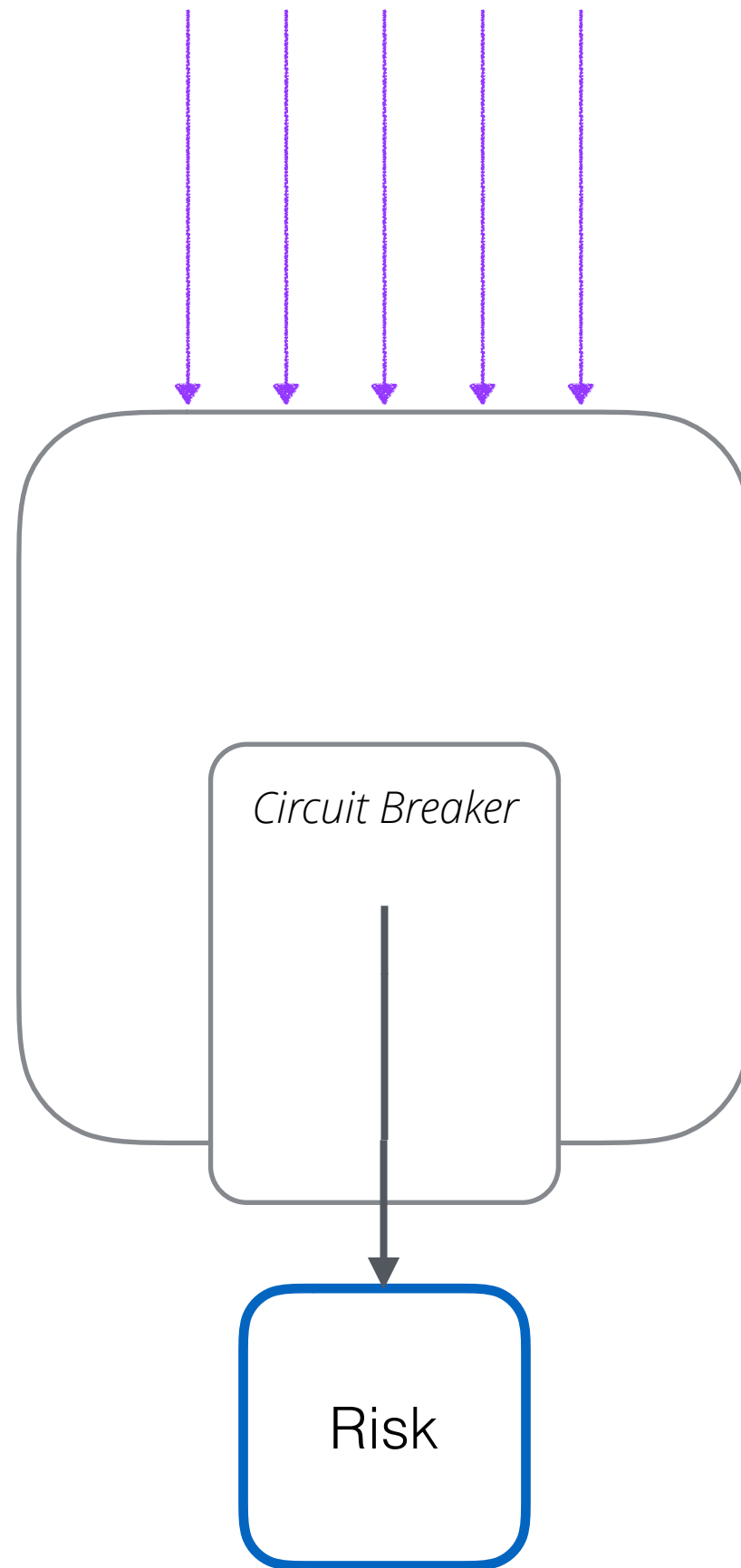
But what about
hybrid apps?

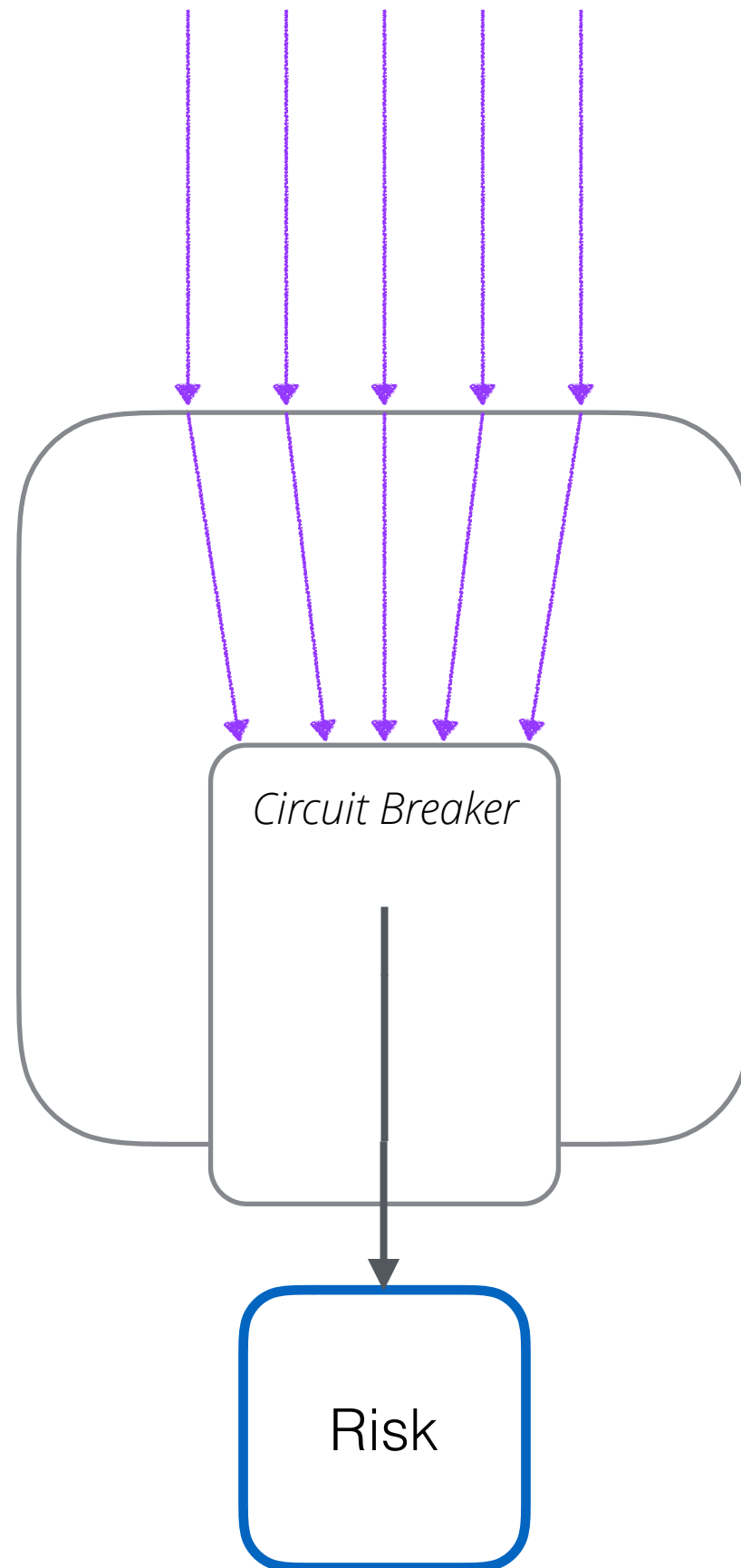
BUSTLE

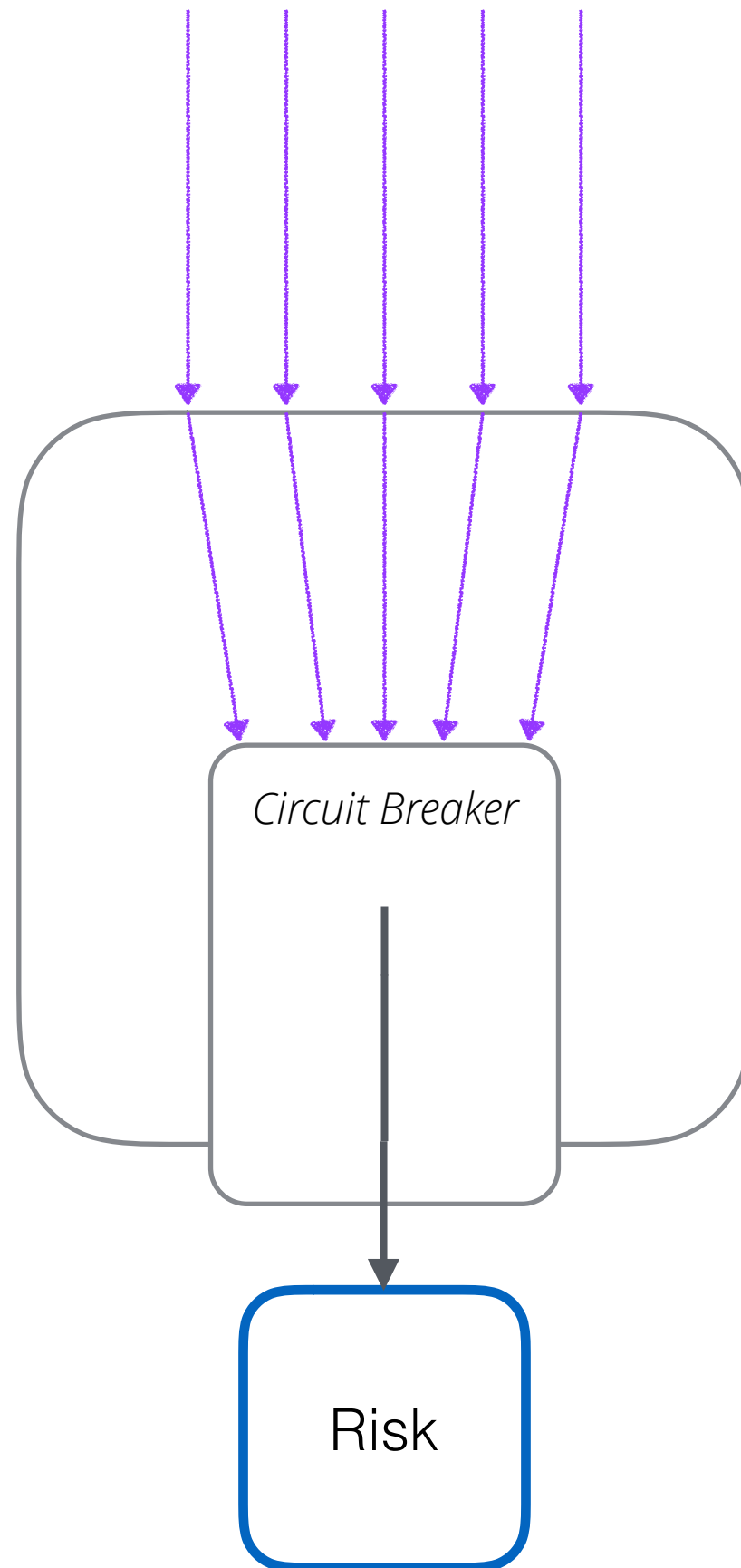


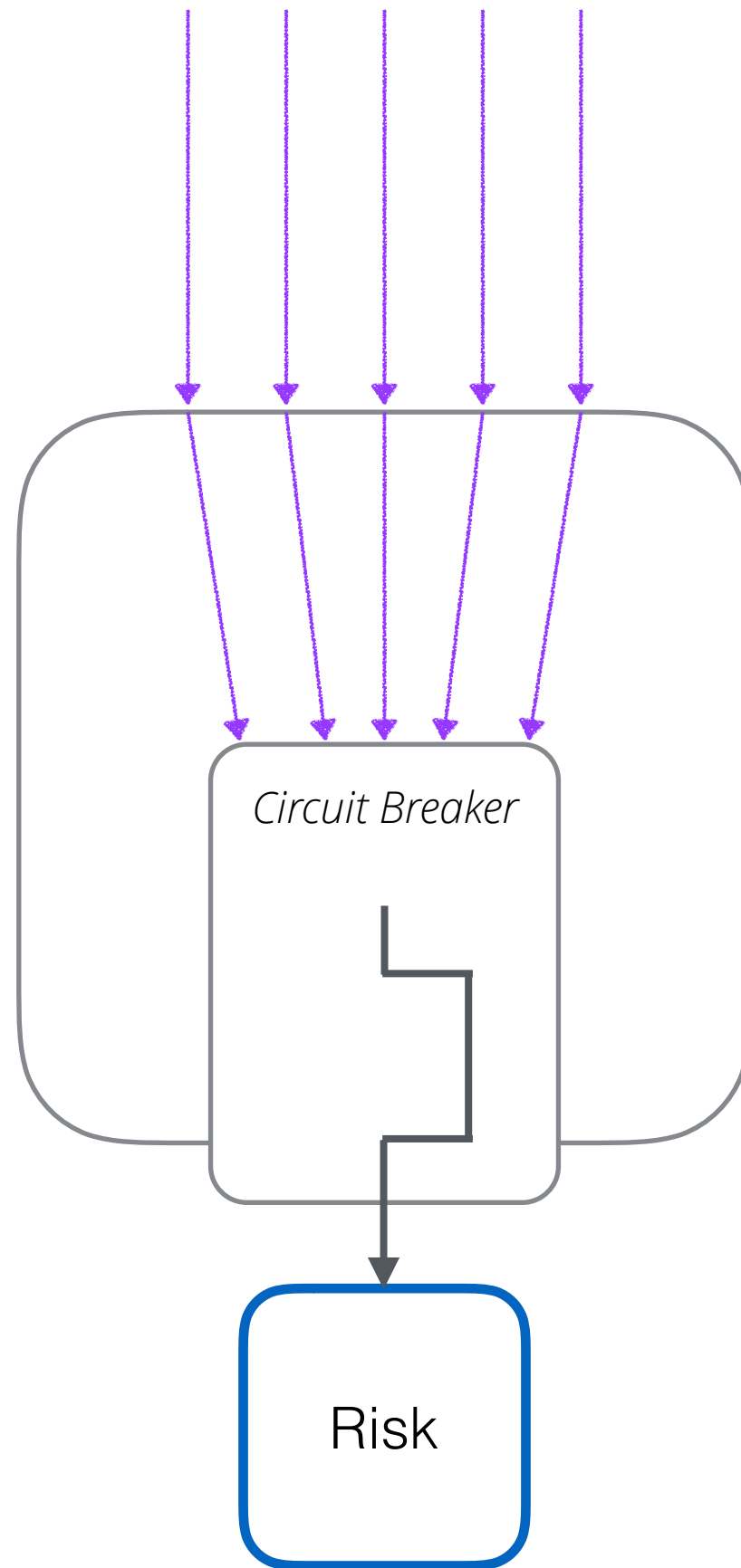


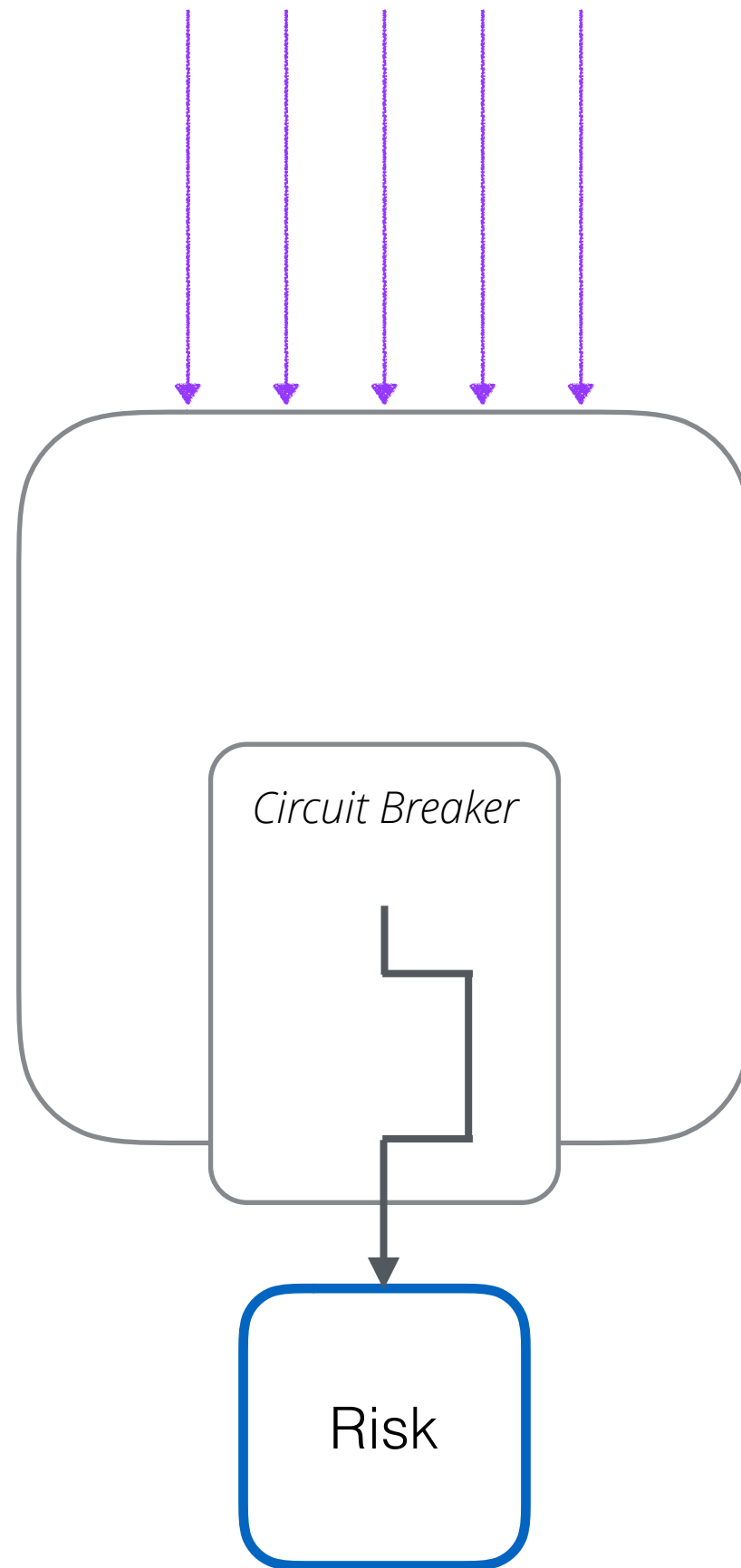


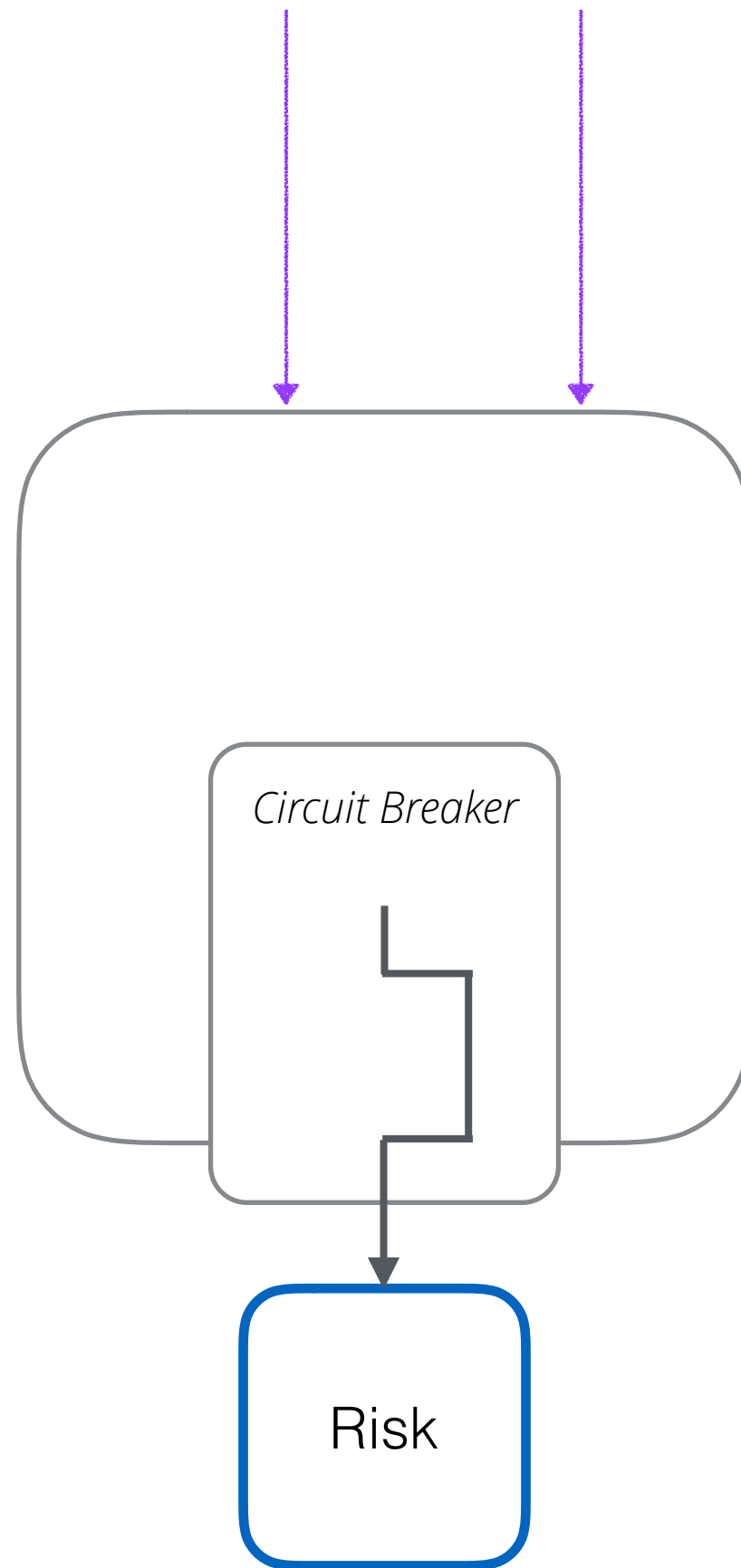




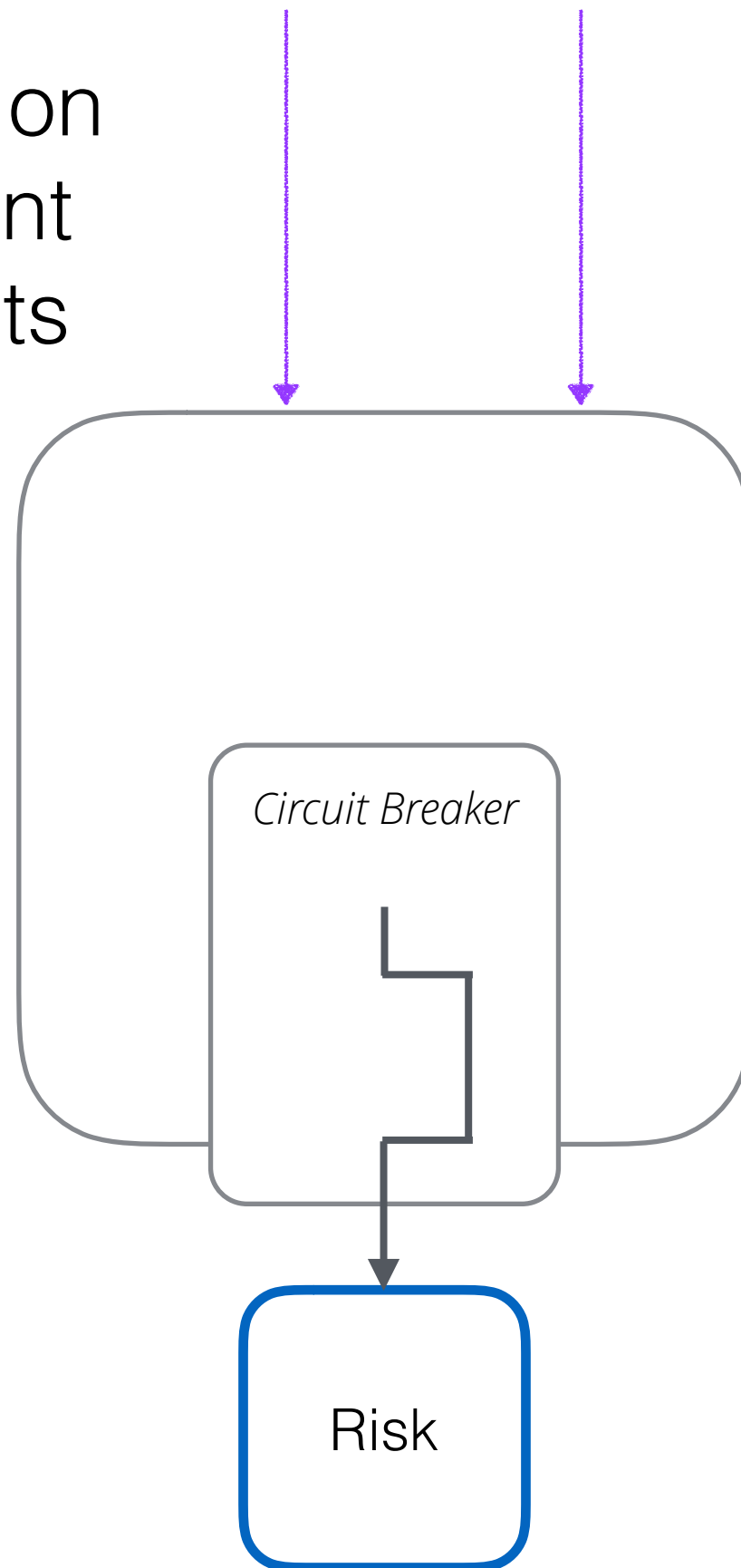


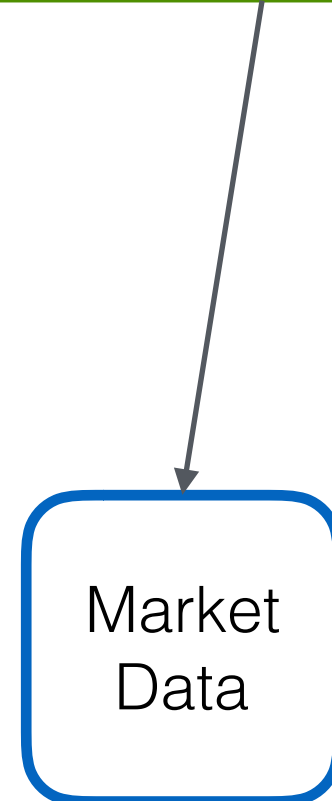
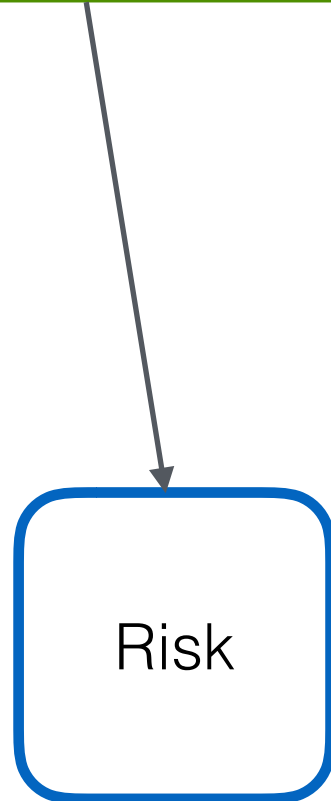
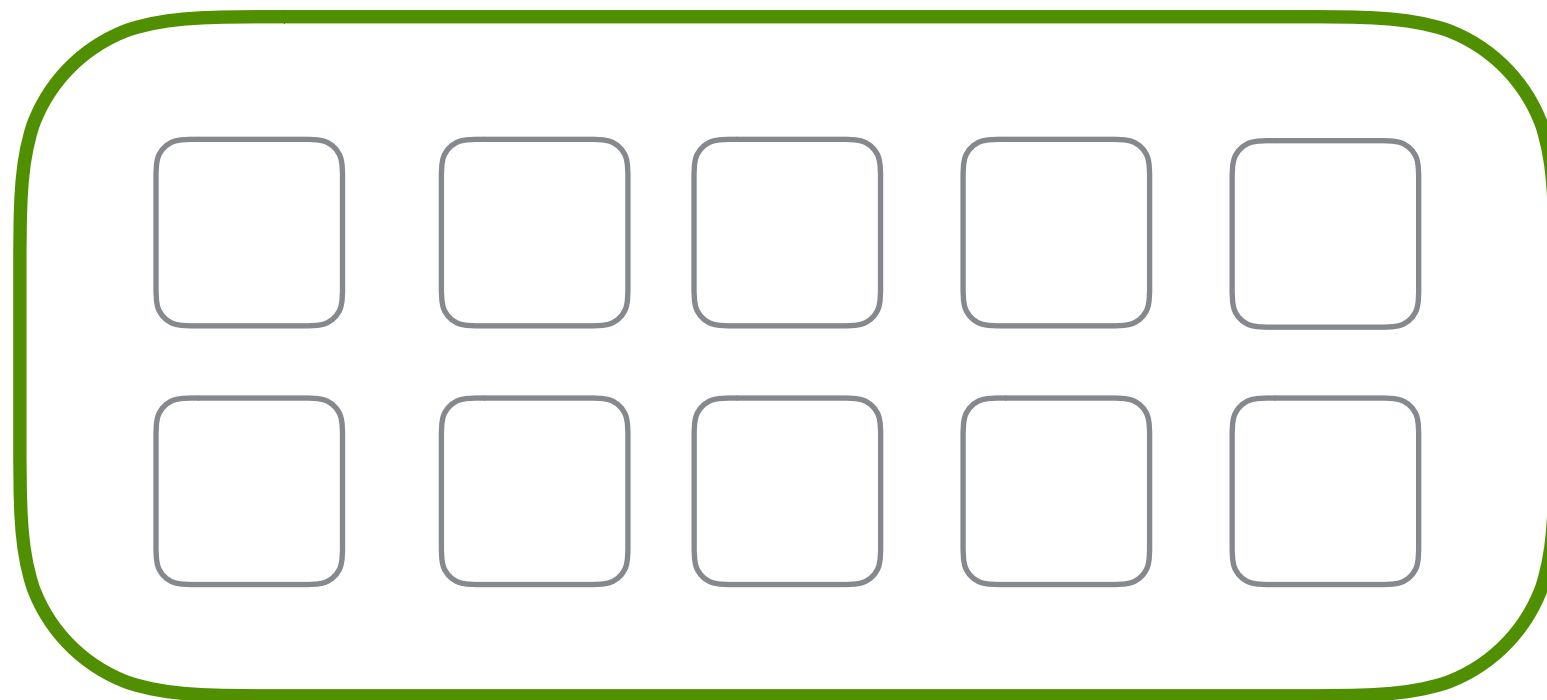


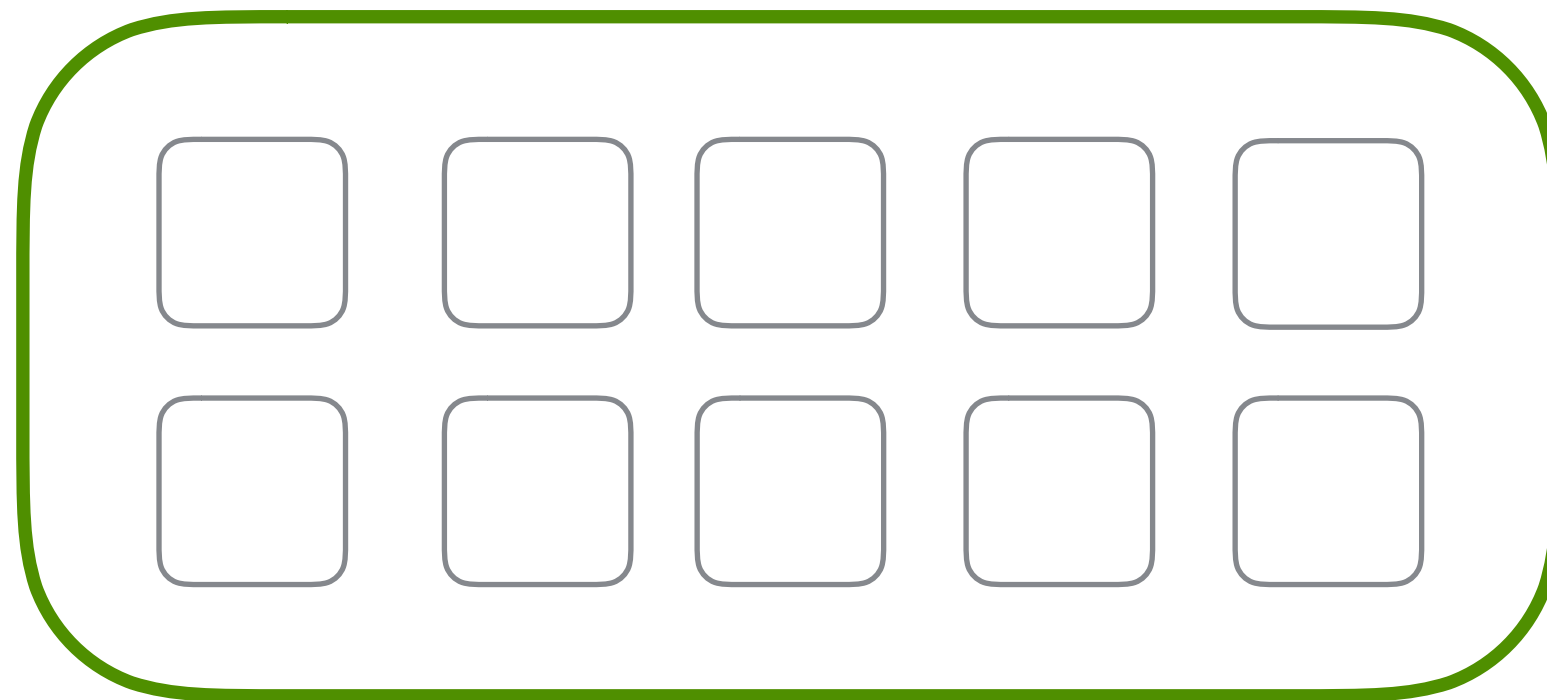




Circuit breakers rely on
maintaining per-client
state across requests



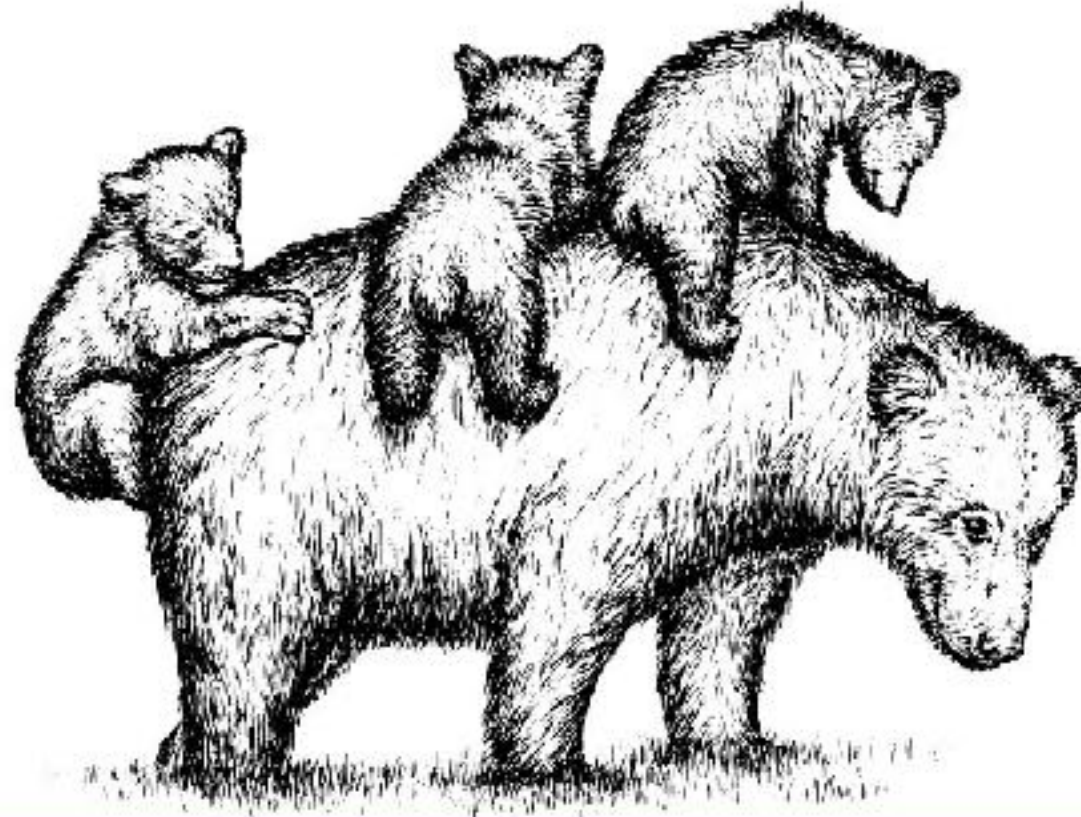




Throttling & Load shedding middleware?



Getting the wrong idea from that conference talk you attended



Solving Imaginary Scaling Issues

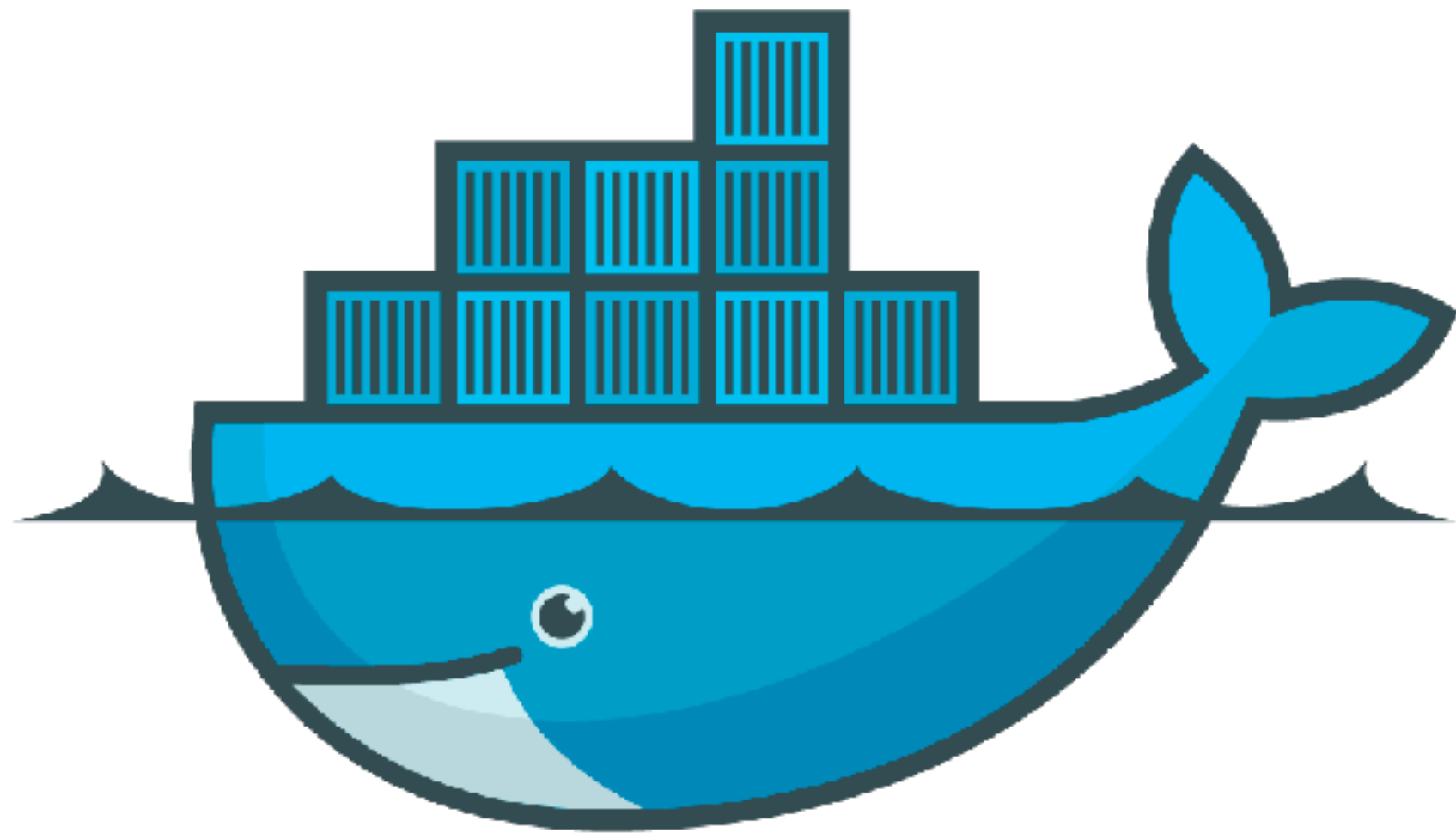
At Scale

O RLY?

@ThePracticalDev

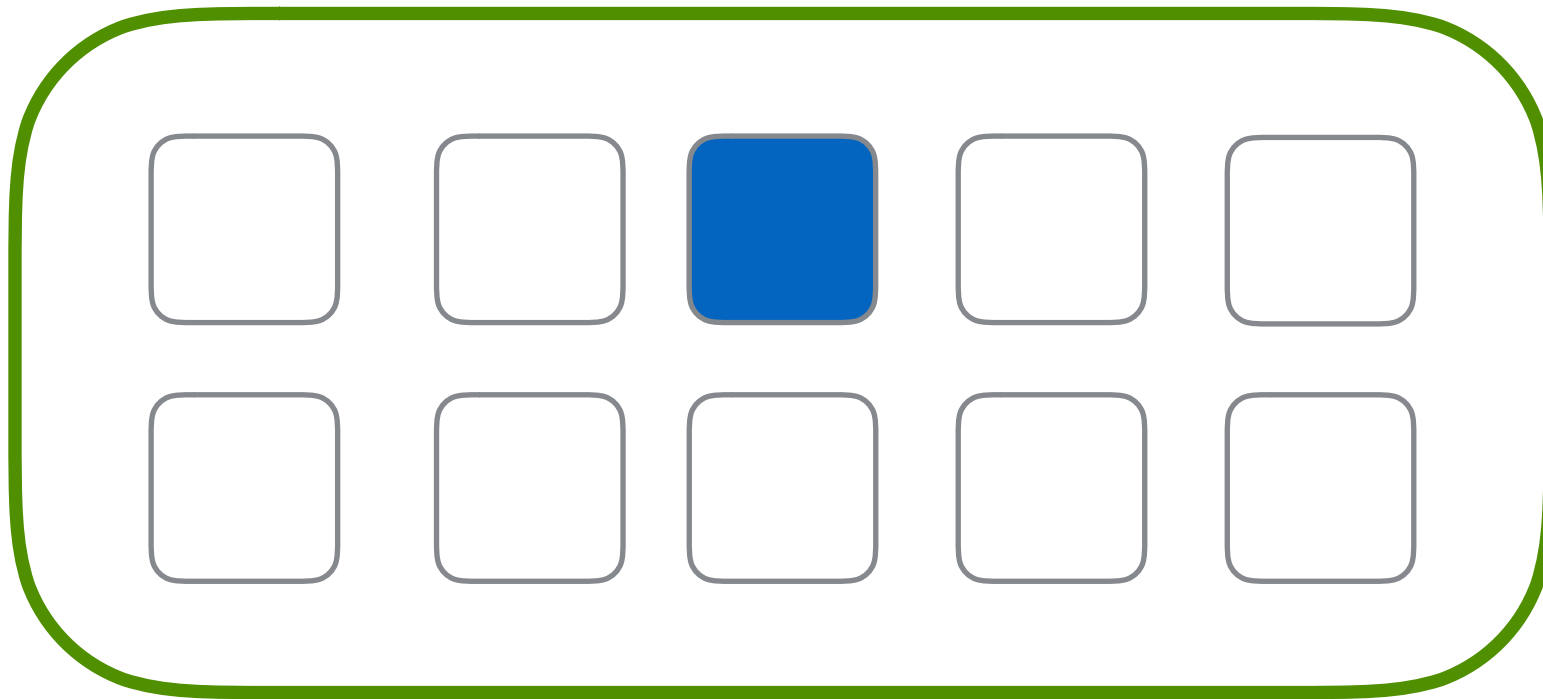


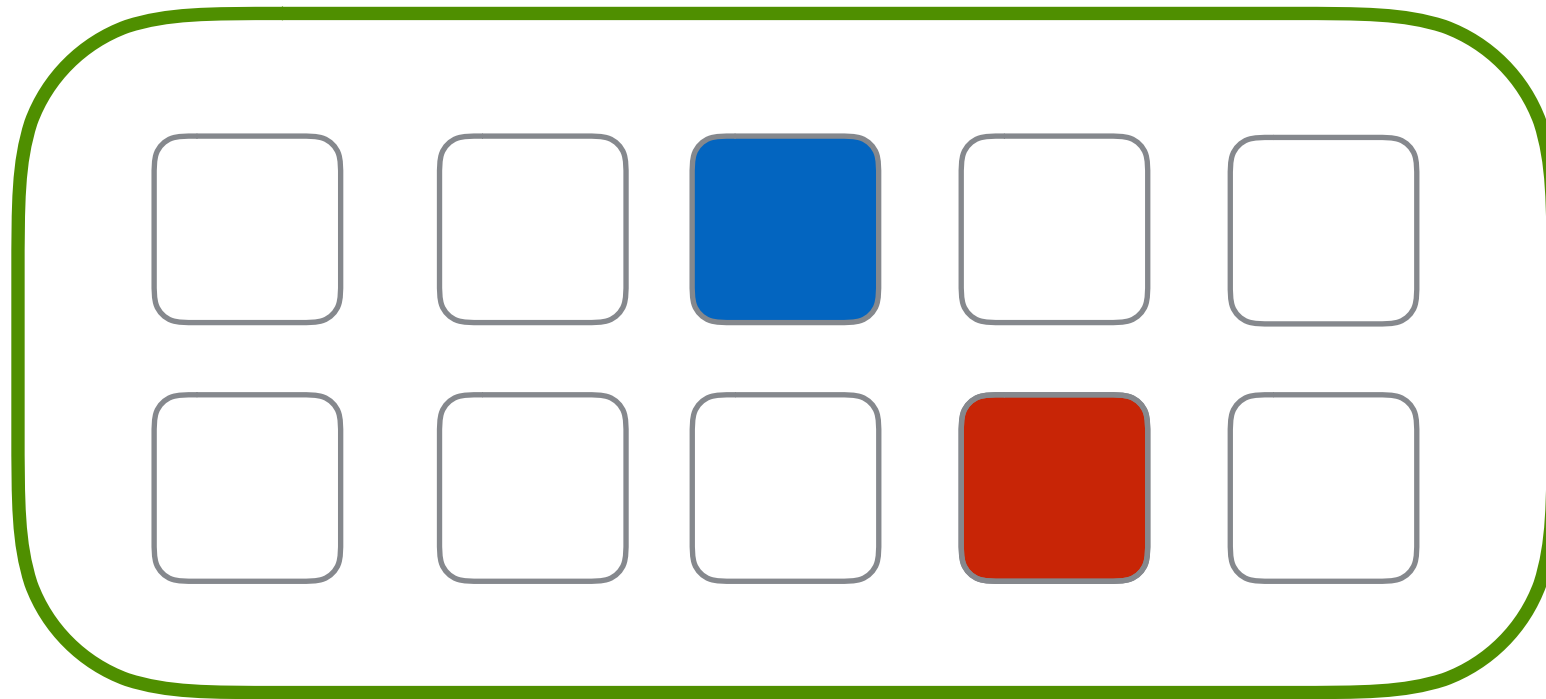
Security

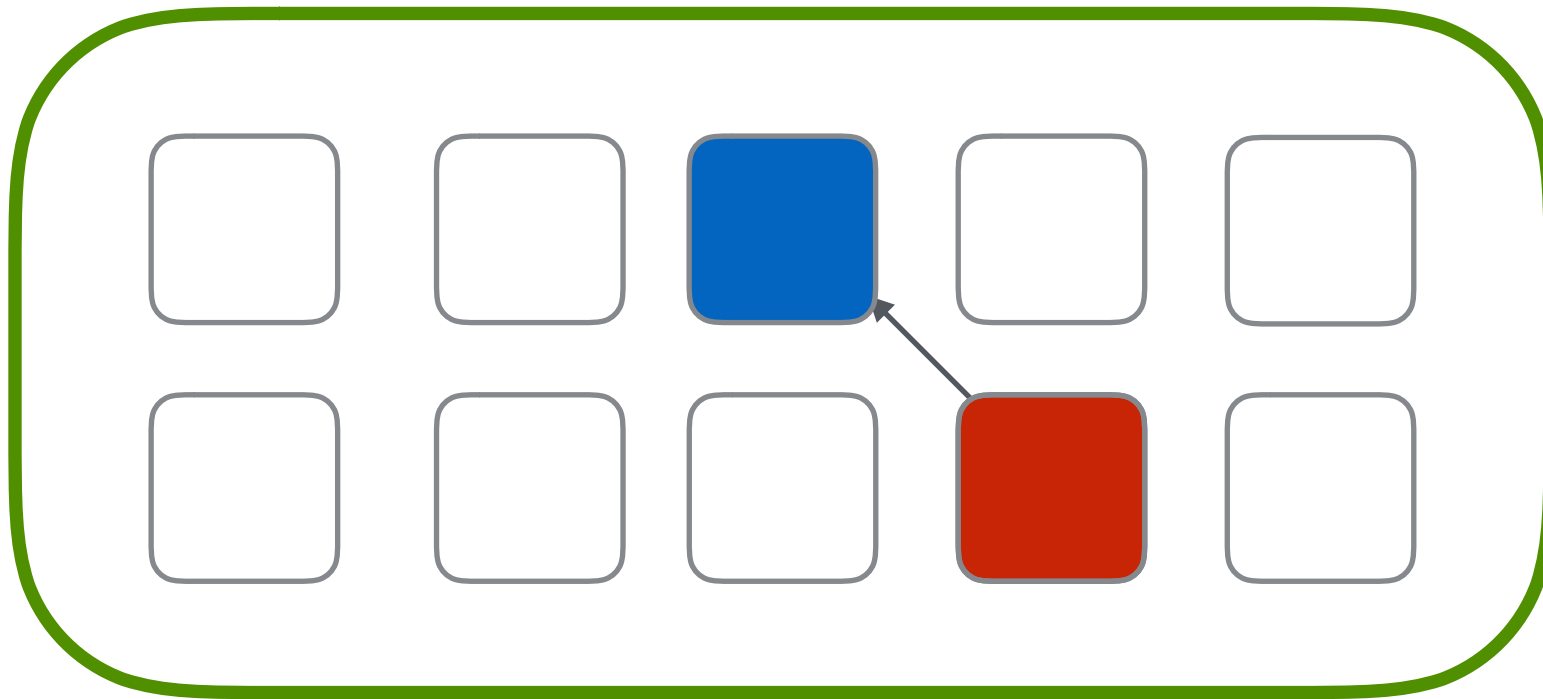


docker

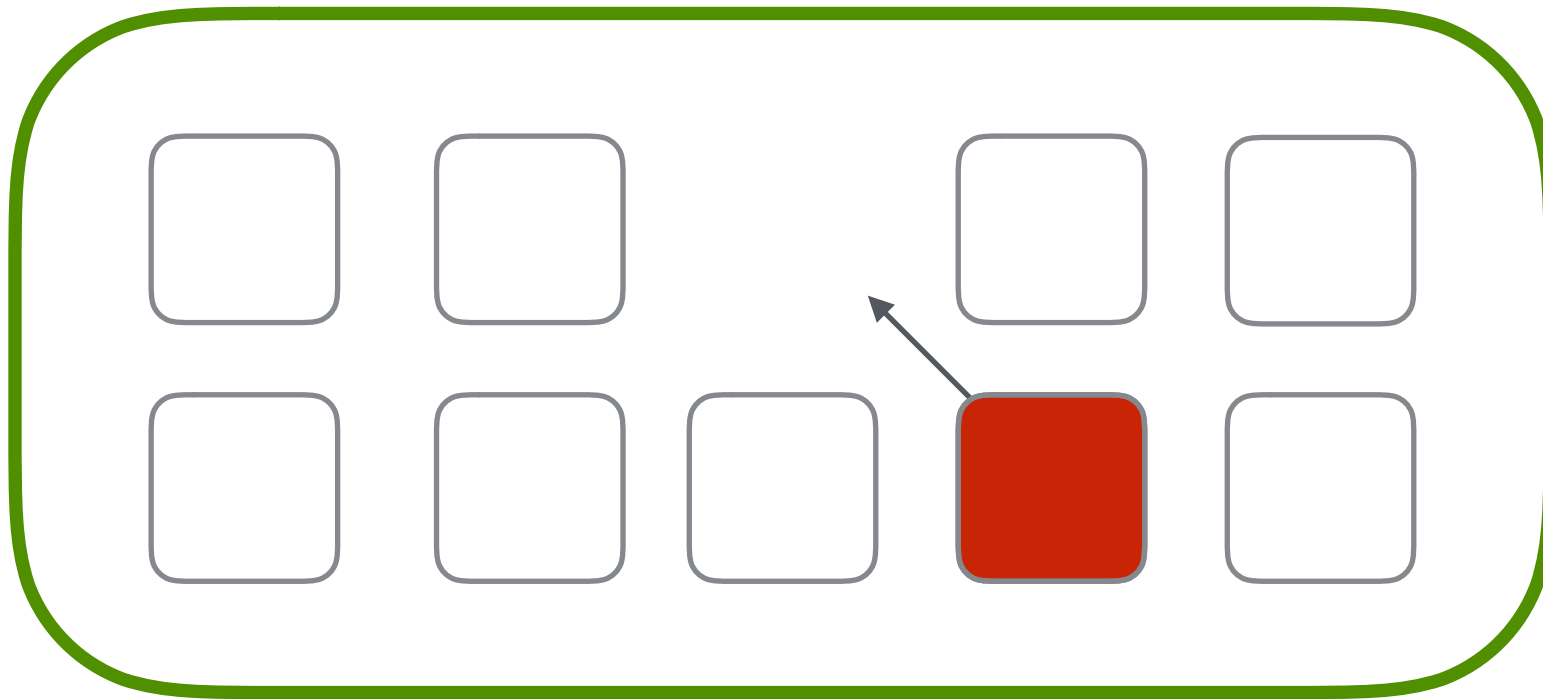
Friends don't let
friends run
untrusted code in
containers



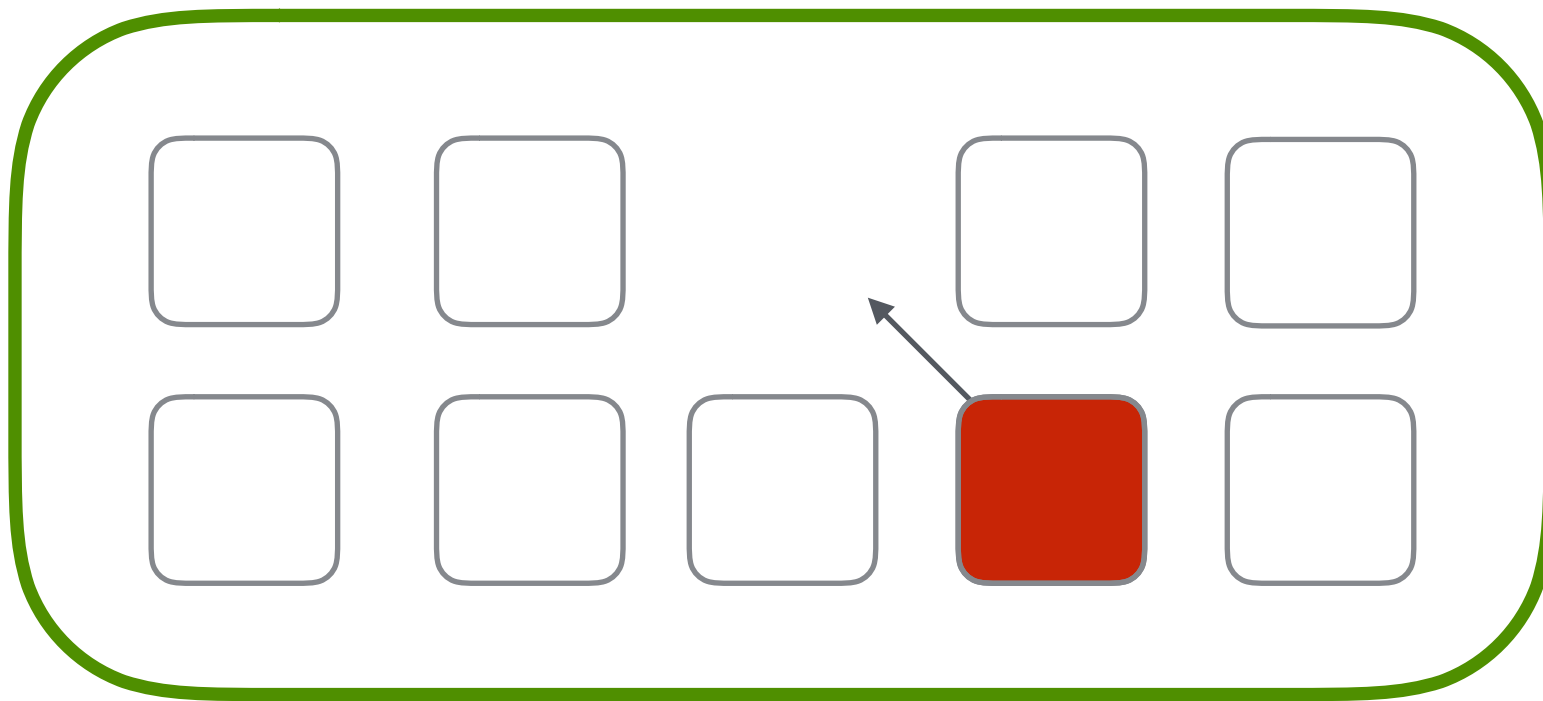




If your function isn't
running, it's not there...

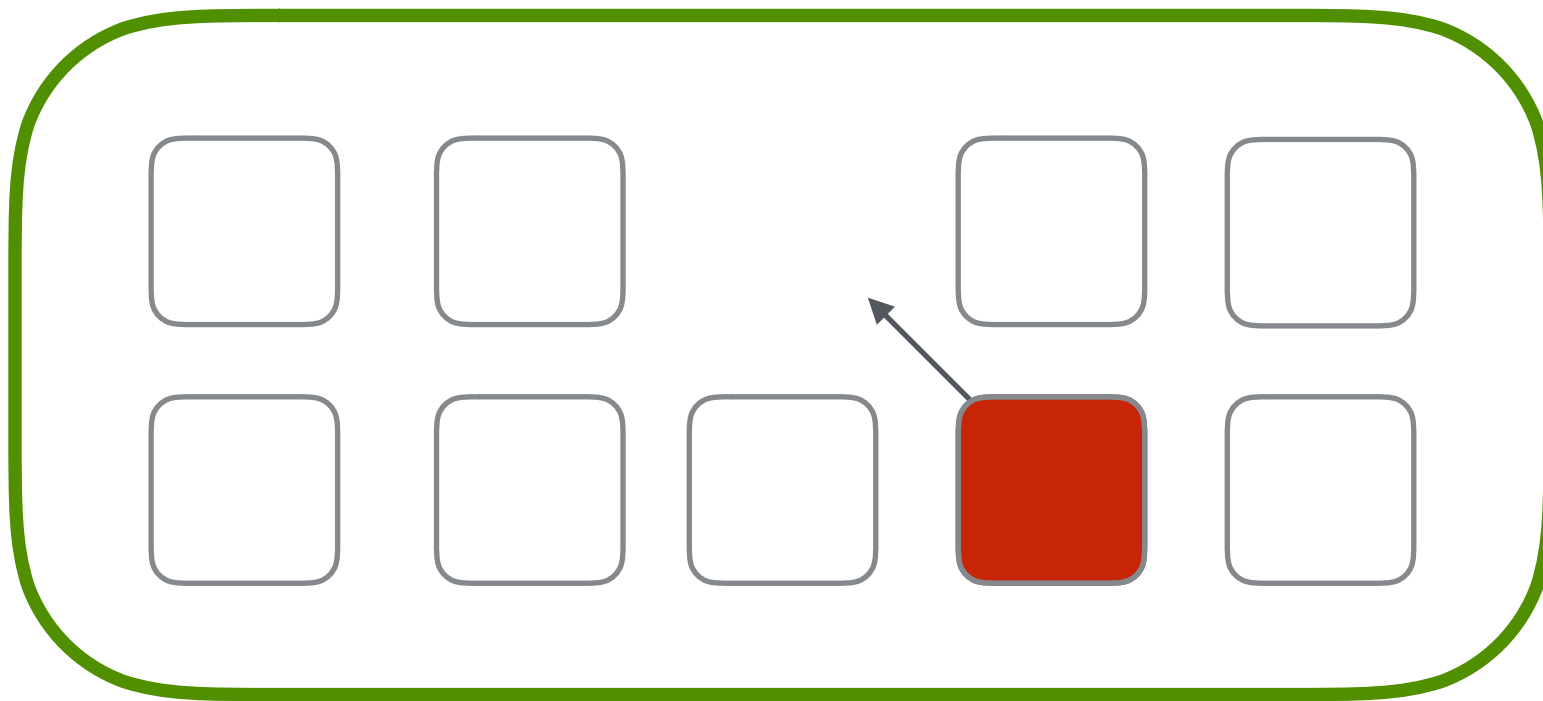


If your function isn't
running, it's not there...



...and you're running in
a sandbox anyway...

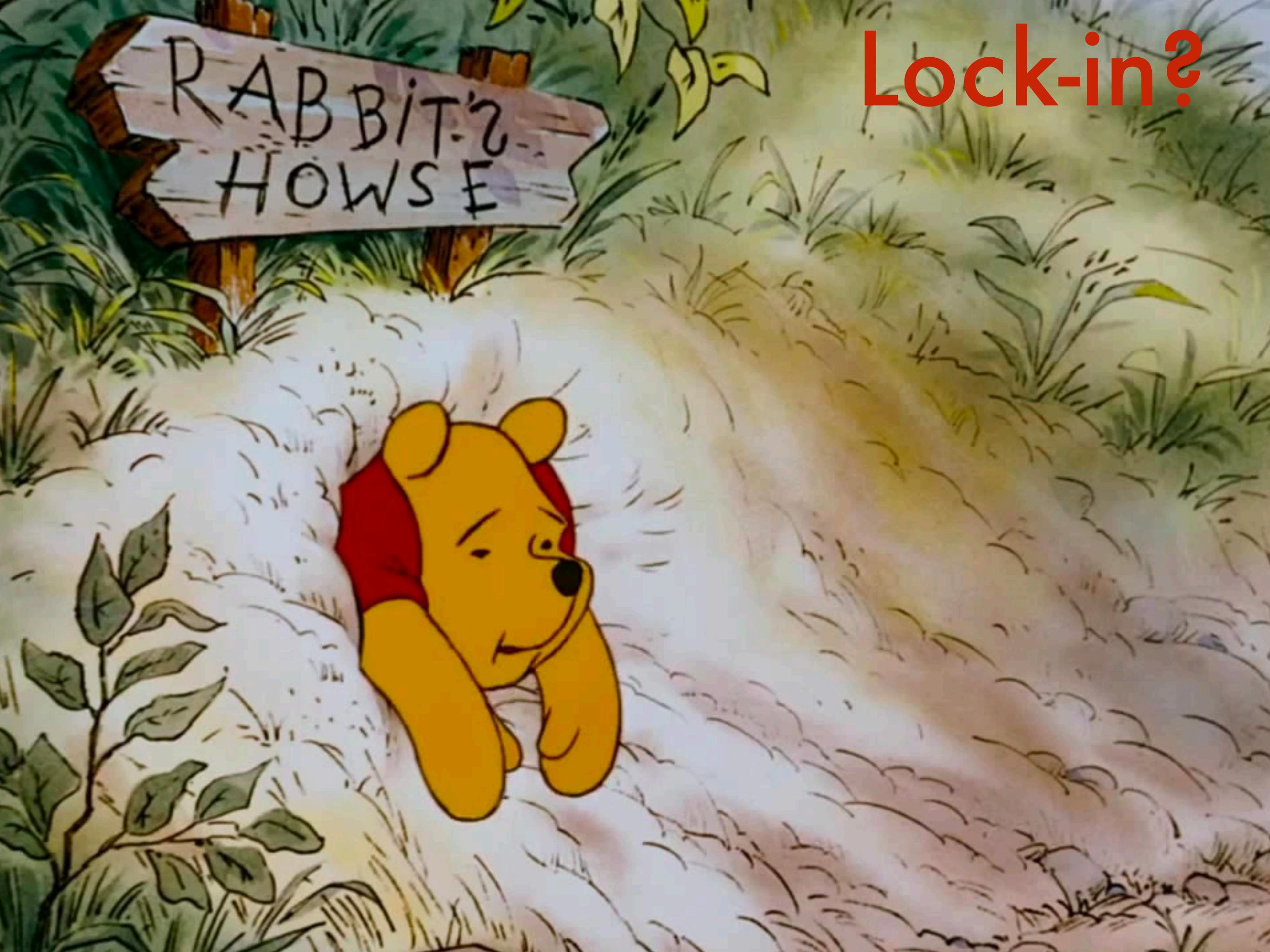
If your function isn't
running, it's not there...



Kinda

...and you're running in
a sandbox anyway...

Lock-in?





Switch now to beat the energy price rises

Seven major suppliers are increasing prices up to 10% - fix your rates today!

 ? ?

Compare energy prices now

Save up to
£618/yr*

Don't think lock-in,
think migration cost

Cost of migration



Cost of migration

Blob storage

Cost of migration

Blob storage

Compute

Cost of migration

Blob storage Load
balancers

Compute

Cost of migration

Blob storage Load
balancers

Compute FAAS

Cost of migration

Blob storage Load balancers BAAS

Compute FAAS

Cost of migration

Pay now or pay later

Mixing vendors?



#gotoams

@samnewman


OpenFaaS - Serverless Functions Made Simple

go report A+ build passing godoc reference License MIT openfaas serverless



OPENFAAS

OpenFaaS (Functions as a Service) is a framework for building serverless functions with Docker and Kubernetes which has first class support for metrics. Any process can be packaged as a function enabling you to consume a range of web events without repetitive boiler-plate coding.

 Follow @openfaas

Highlights

- Ease of use through UI portal and *one-click* install

<https://github.com/openfaas/faas>

faas-netes

go report **A+** build **passing** License **MIT** openfaas **serverless**

This is a plugin to enable Kubernetes as an **OpenFaaS** backend. The existing CLI and UI are fully compatible. It also opens up the possibility for other plugins to be built for orchestration frameworks such as Nomad, Mesos/Marathon or even a cloud-managed back-end such as Hyper.sh or Azure ACI.

Update: [Watch the demo and intro to the CNCF Serverless Workgroup](#)

OpenFaaS is an event-driven serverless framework for containers. Any container for Windows or Linux can be leveraged as a serverless function. OpenFaaS is quick and easy to deploy (less than 60 secs) and lets you avoid writing boiler-plate code.

Functions as a Service

API Gateway

Function Watchdog



Prometheus



Swarm



Kubernetes

<https://github.com/openfaas/faas-netes>

Fission: Serverless Functions for Kubernetes

build passing go report A+ slack 18/351

fission.io [@fissionio](https://twitter.com/fissionio)

Fission is a fast serverless framework for Kubernetes with a focus on developer productivity and high performance.

Fission operates on *just the code*: Docker and Kubernetes are abstracted away under normal operation, though you can use both to extend Fission if you want to.

Fission is extensible to any language; the core is written in Go, and language-specific parts are isolated in something called *environments* (more below). Fission currently supports NodeJS, Python, Ruby, Go, PHP, Bash, and any Linux executable, with more languages coming soon.


Performance: 100msec cold start

Fission maintains a pool of "warm" containers that each contain a small dynamic loader. When a function is first called, i.e. "cold-started", a running container is chosen and the function is loaded. This pool is what makes Fission fast: cold-start latencies are typically about 100msec.

Kubernetes is the right place for Serverless

We're built on Kubernetes because we think any non-trivial app will use a combination of serverless functions and more conventional microservices, and Kubernetes is a great framework to bring these together seamlessly.

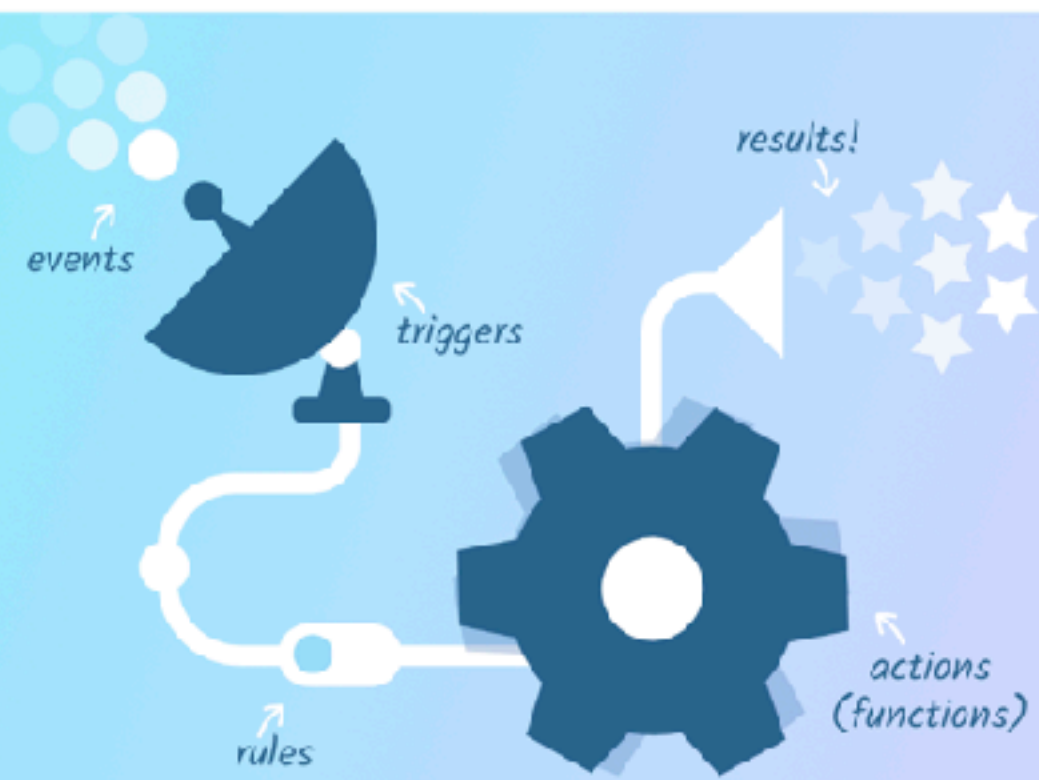
<https://github.com/fission/fission>



DocsGet StartedCommunityResources

Apache OpenWhisk (Incubating) is a serverless, **open source** cloud platform that executes functions in **response to events** at any scale.

Get Started



```
OK: created action hello
$ wsk action invoke --wait --verbose
ok: invoked hello
{
  "message": "hello"
}
```

Where to start.

With Apache OpenWhisk you can easily create actions, test, connect to other actions or debug them. Use the [Apache OpenWhisk CLI](#) target your Apache OpenWhisk instance, and run your first action in seconds.

<http://openwhisk.incubator.apache.org>

OpenLambda

OpenLambda is an Apache-licensed serverless computing project, written in Go and based on Linux containers. One of the goals of OpenLambda is to enable exploration of new approaches to serverless computing. Our research agenda is described in more detail in a [HotCloud '16 paper](#)

Getting Started

These instructions assume you are the root user.

Install Docker and the Go compiler, then build OpenLambda:

```
make
```

Now you can use the admin tool to create a local OpenLambda cluster:

```
./bin/admin new -cluster my-cluster
```

<https://github.com/open-lambda/open-lambda>



Agile

Agile

DevOps

Agile

DevOps

Microservices

I can't use that, it's
not #serverless!

“There is No Future with Fewer Servers”

- Luke Kanies

It's just
abstractions...

It's just
abstractions...

...all the way down

Thank You!

<http://samnewman.io>

@samnewman