



PRACTICAL CONSIDERATIONS FOR MICRO SERVICES

*Sam Newman
Javazone 2014, September 2014*



14 PRACTICAL TIPS FOR MICROSERVICES

*Sam Newman
Javazone 2014, September 2014*



**YOU WON'T BELIEVE
WHAT THEY DO WITH
MICROSERVICES IN
THIS TALK!**

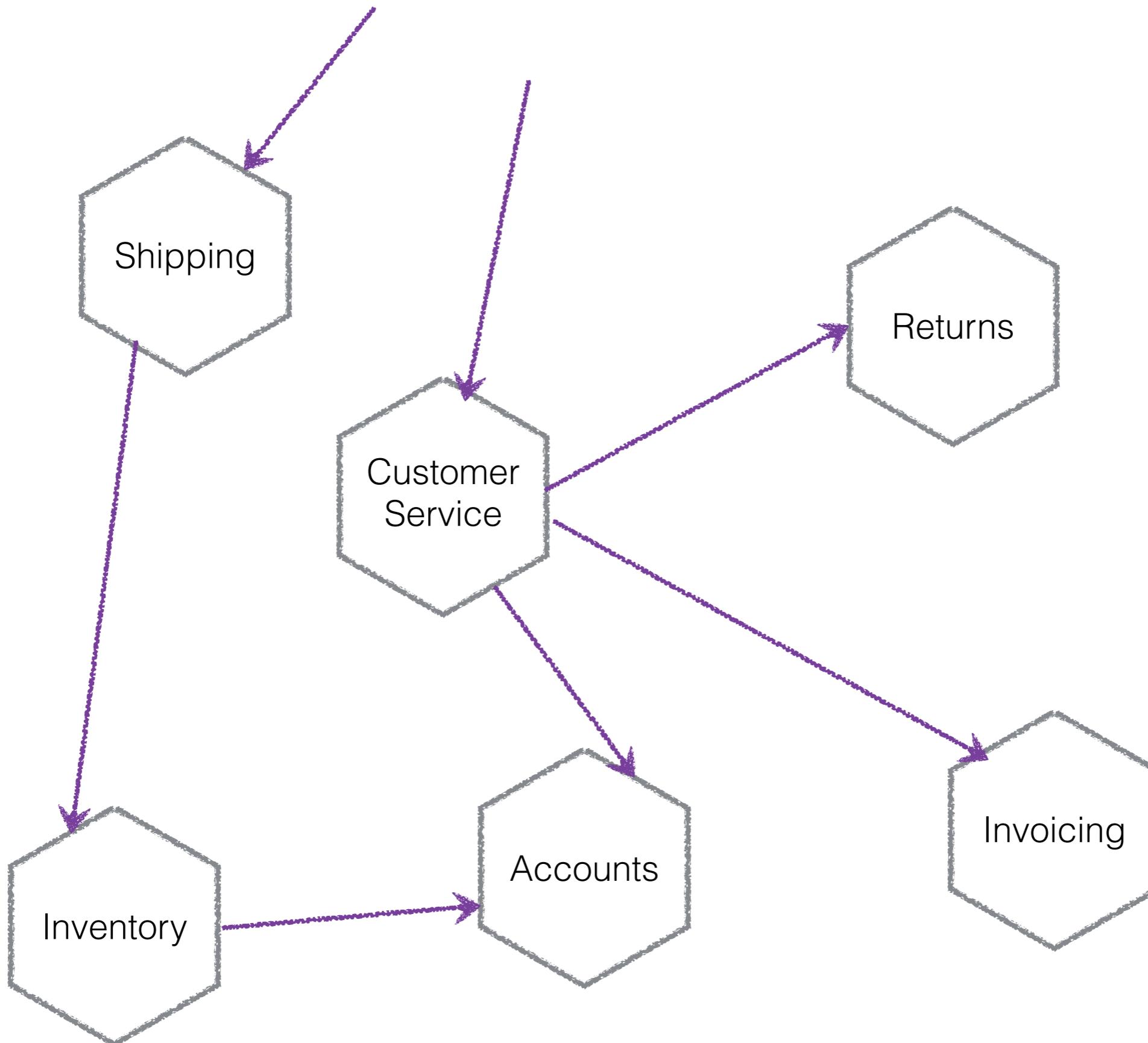
*Sam Newman
Javazone 2014, September 2014*



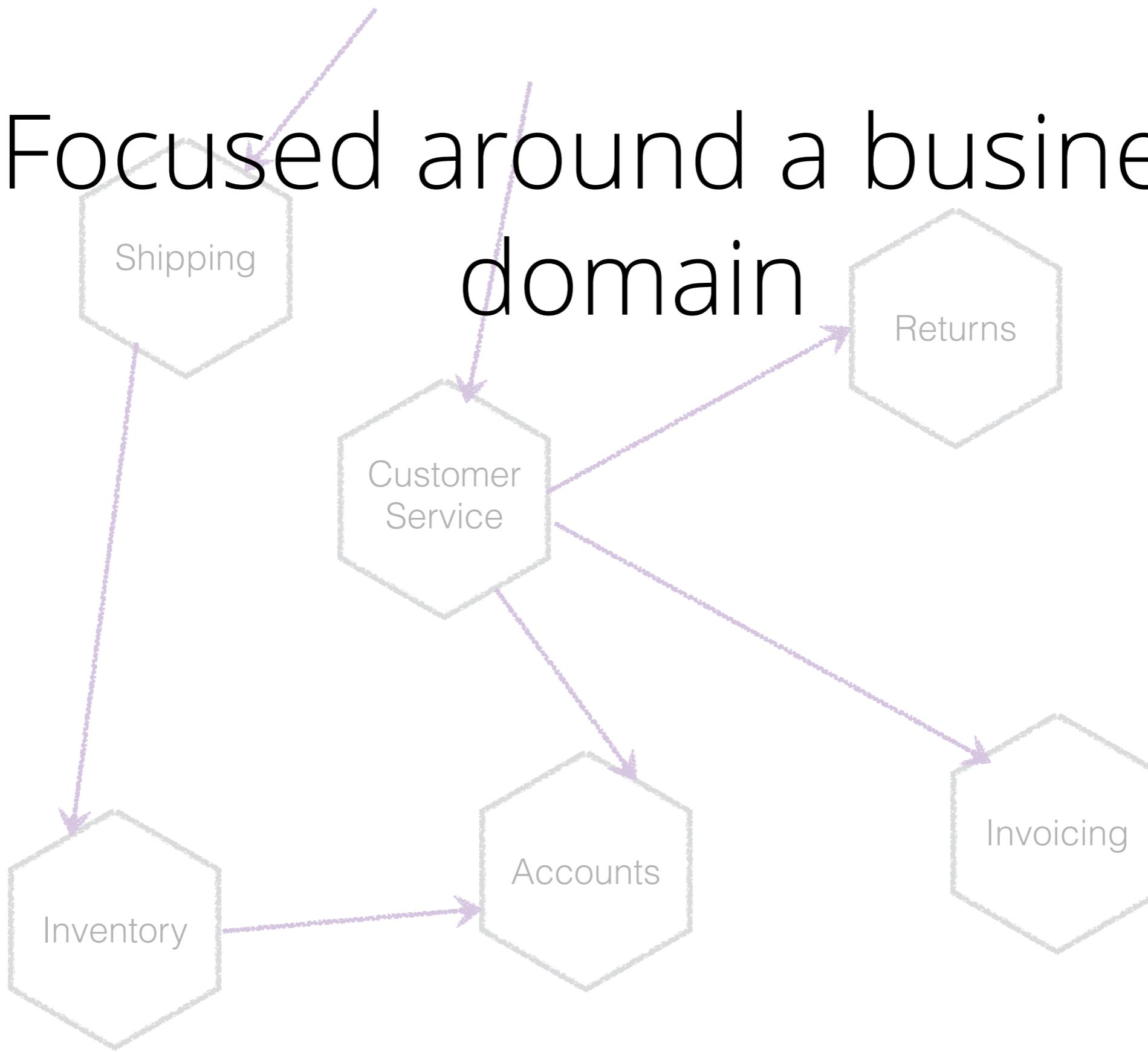
**SEE WHAT JENNIFER
LAWRENCE THINKS
ABOUT DISTRIBUTED
SYSTEMS!**

Sam Newman

Javazone 2014, September 2014

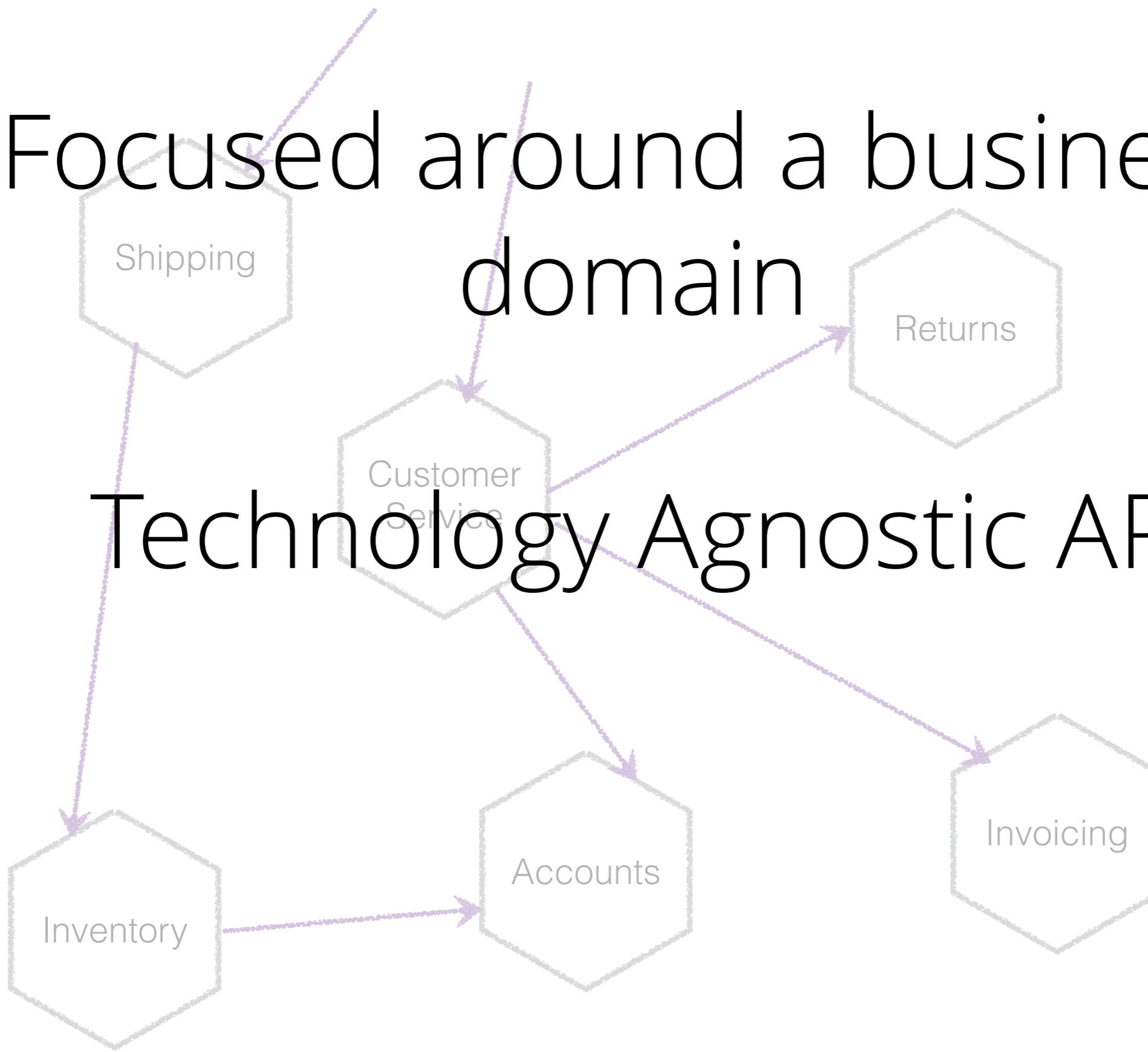


Focused around a business domain



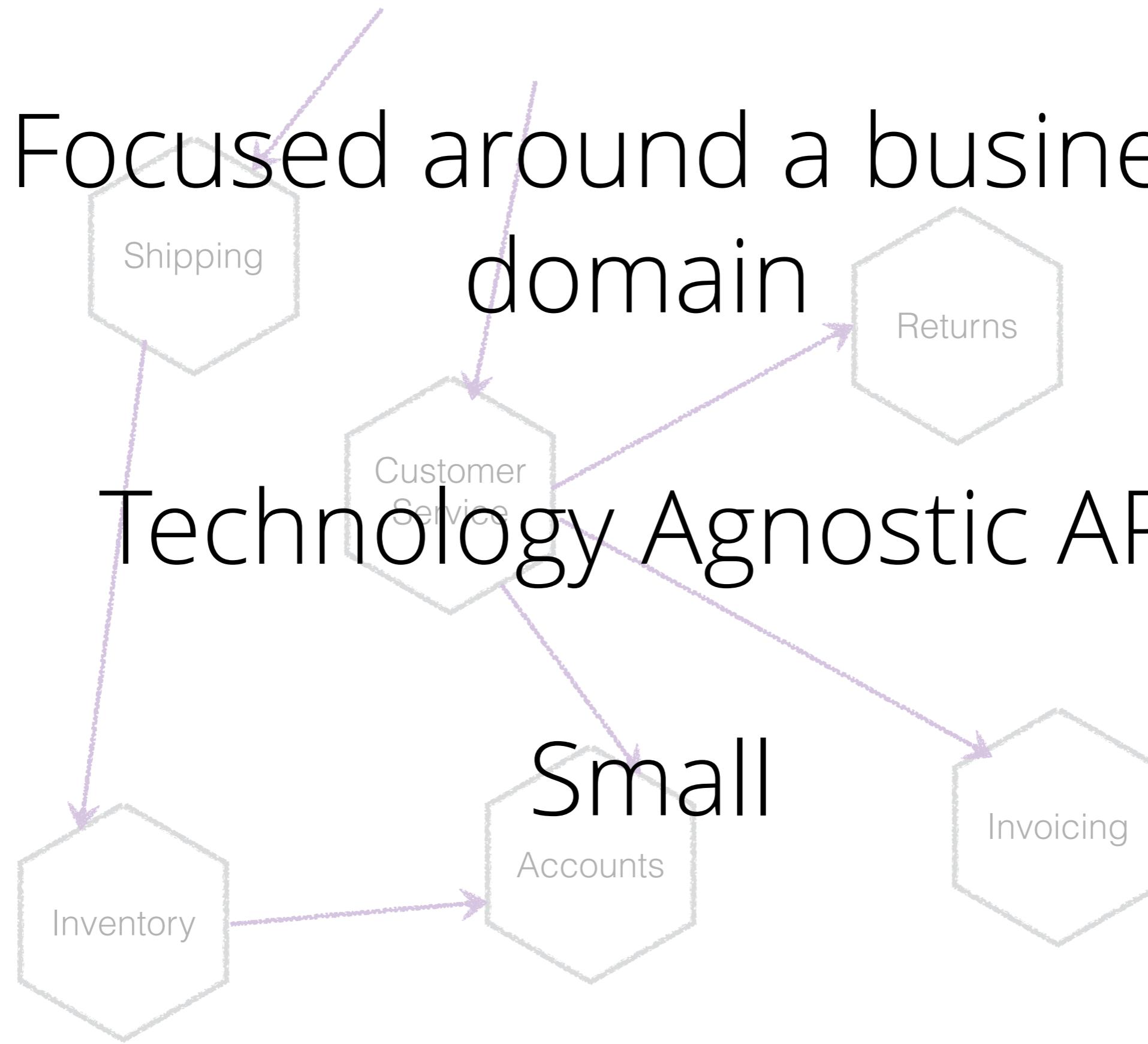
Focused around a business
domain

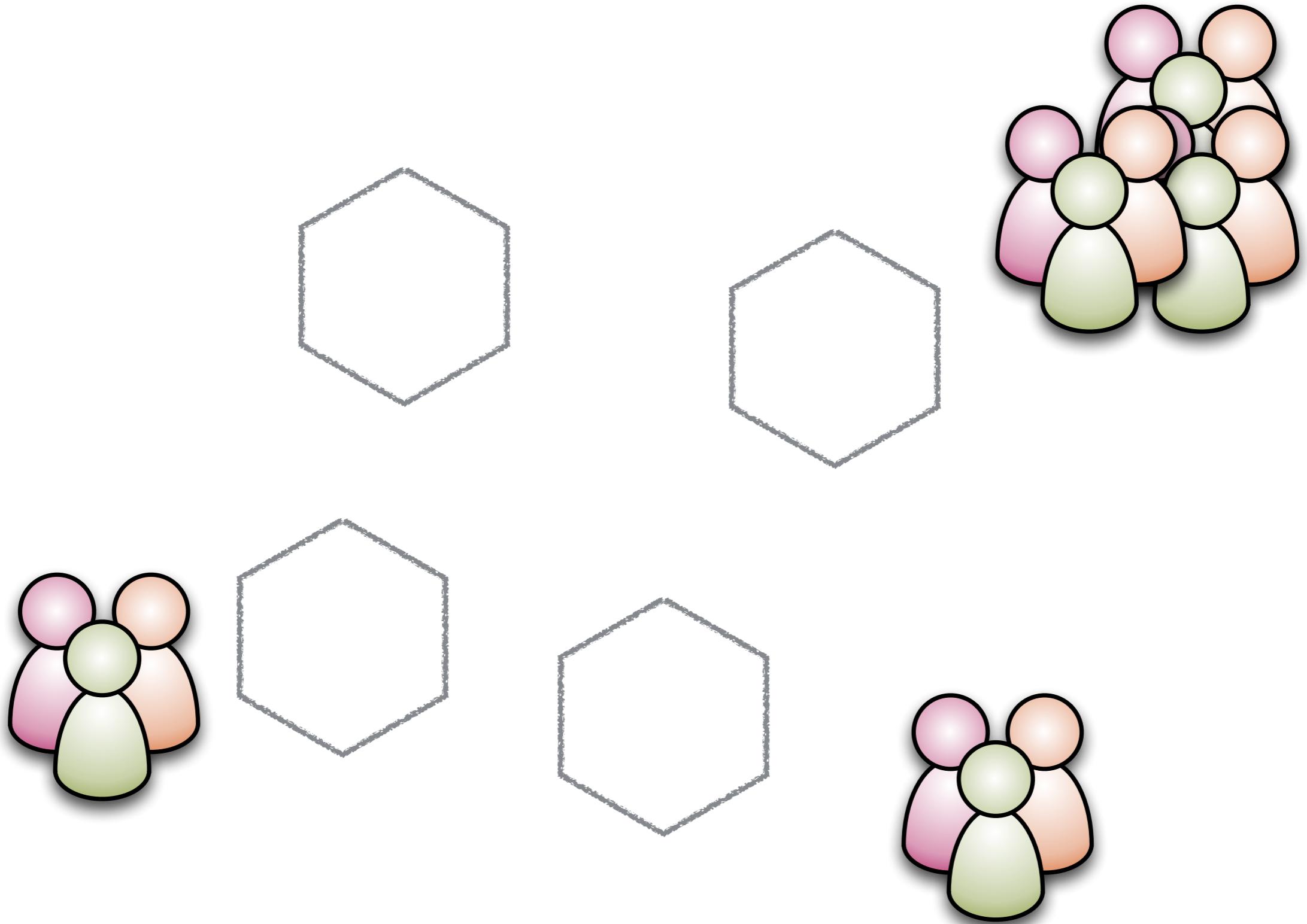
Technology Agnostic API



Focused around a business
domain

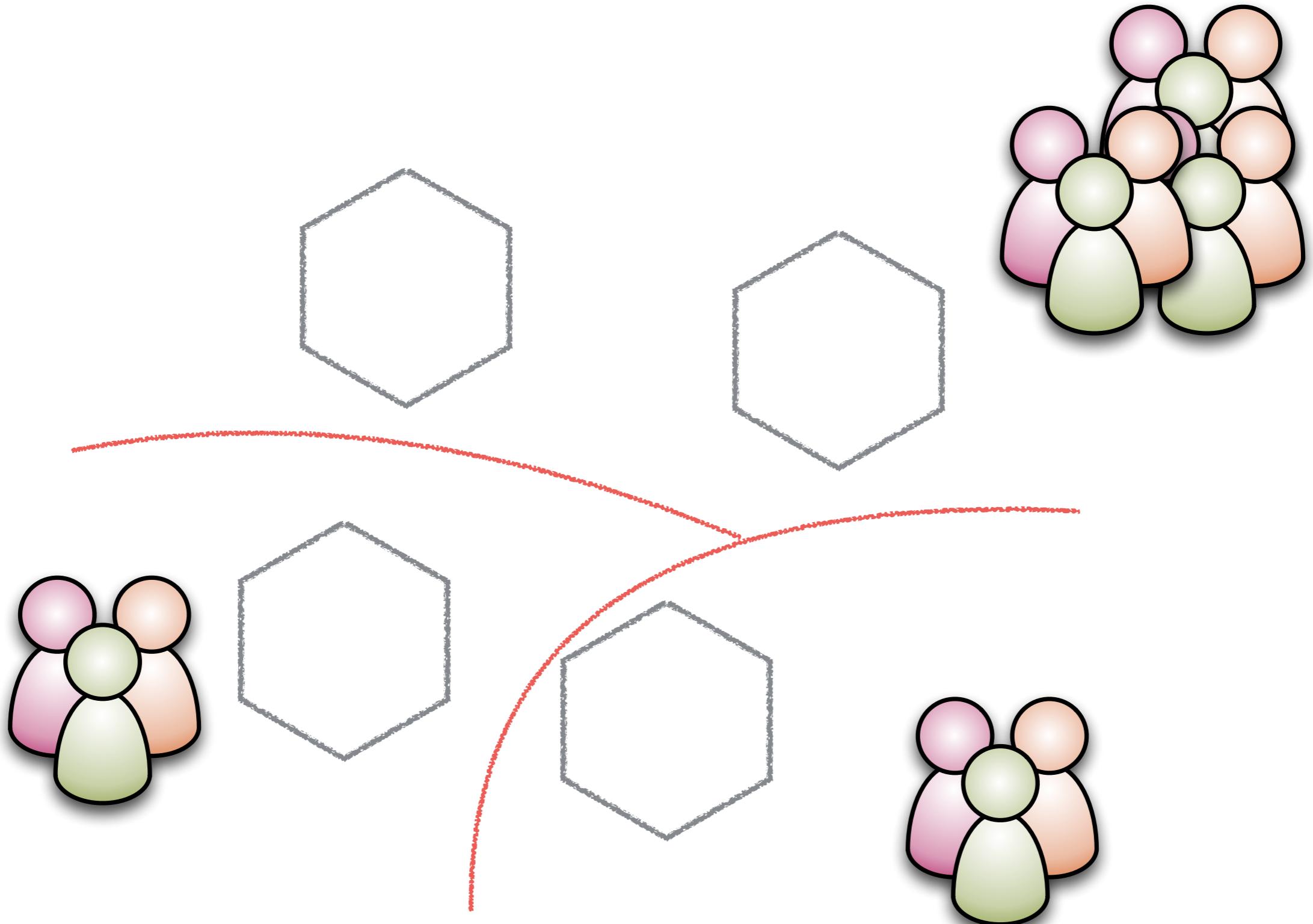
Technology Agnostic API





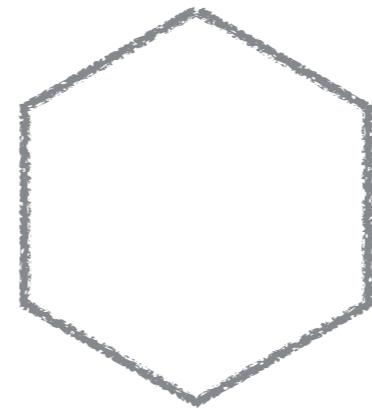
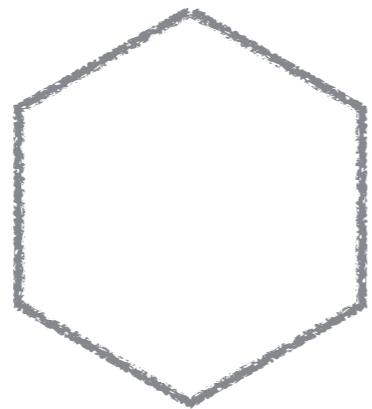
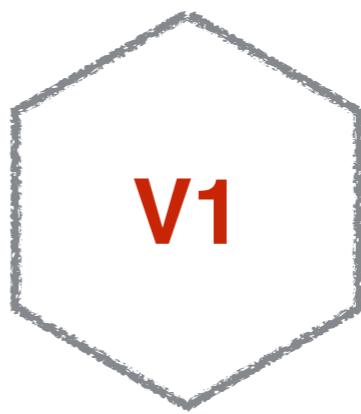
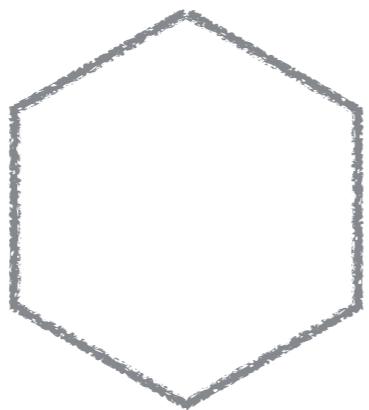
@javazone

@samnewman



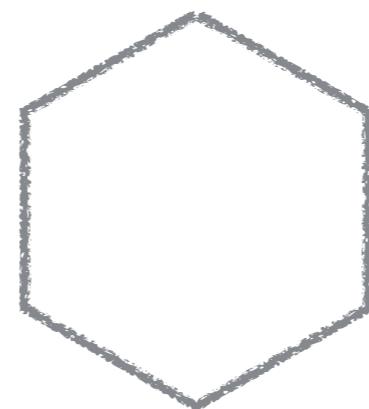
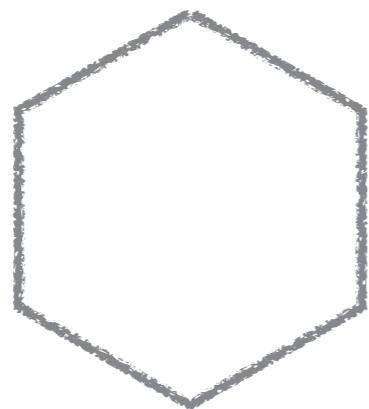
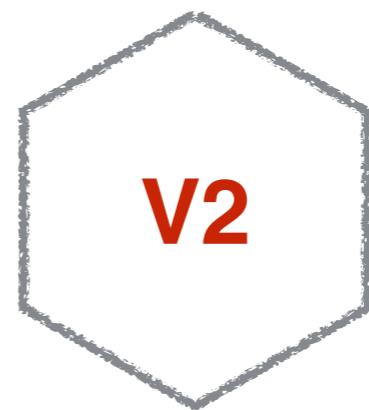
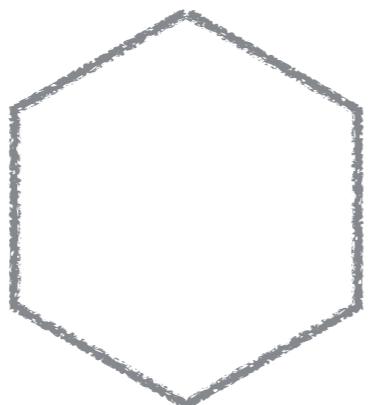
@javazone

@samnewman



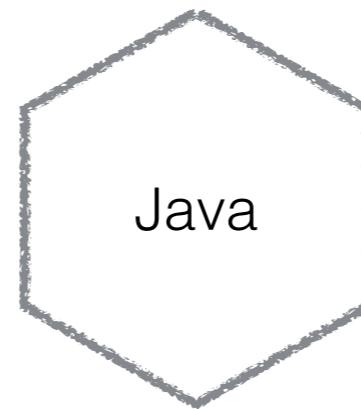
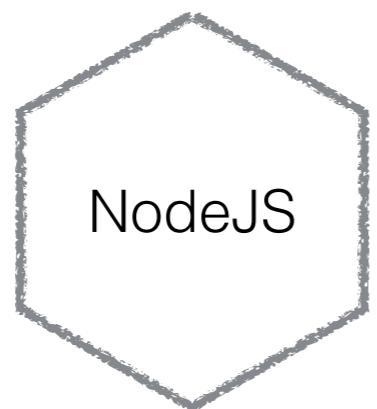
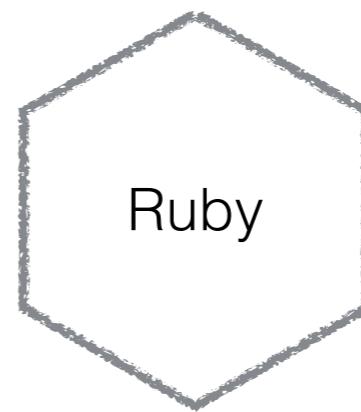
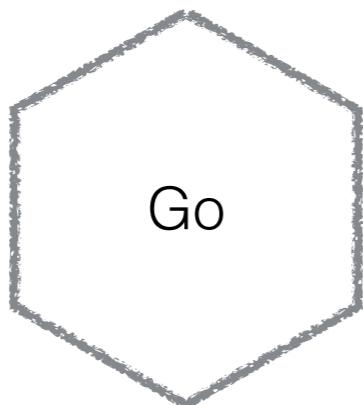
@javazone

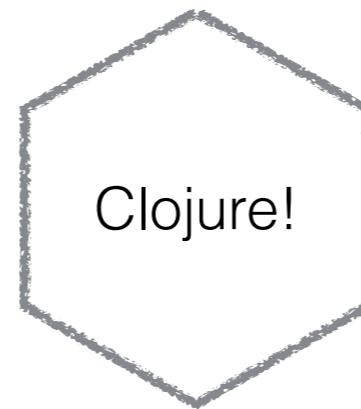
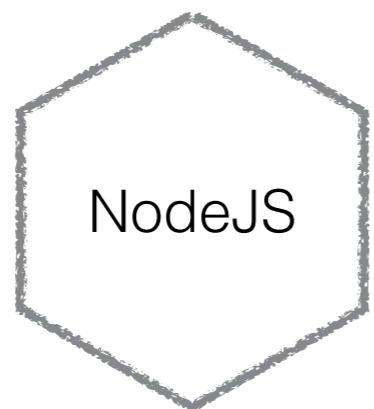
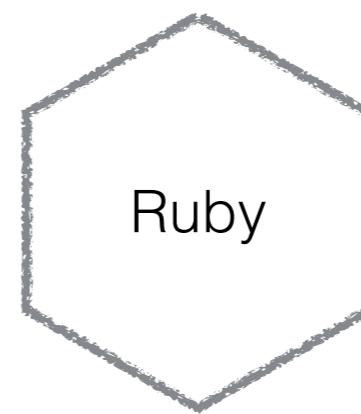
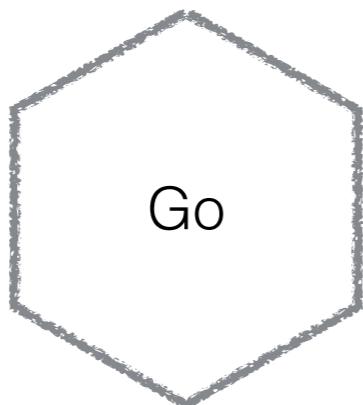
@samnewman

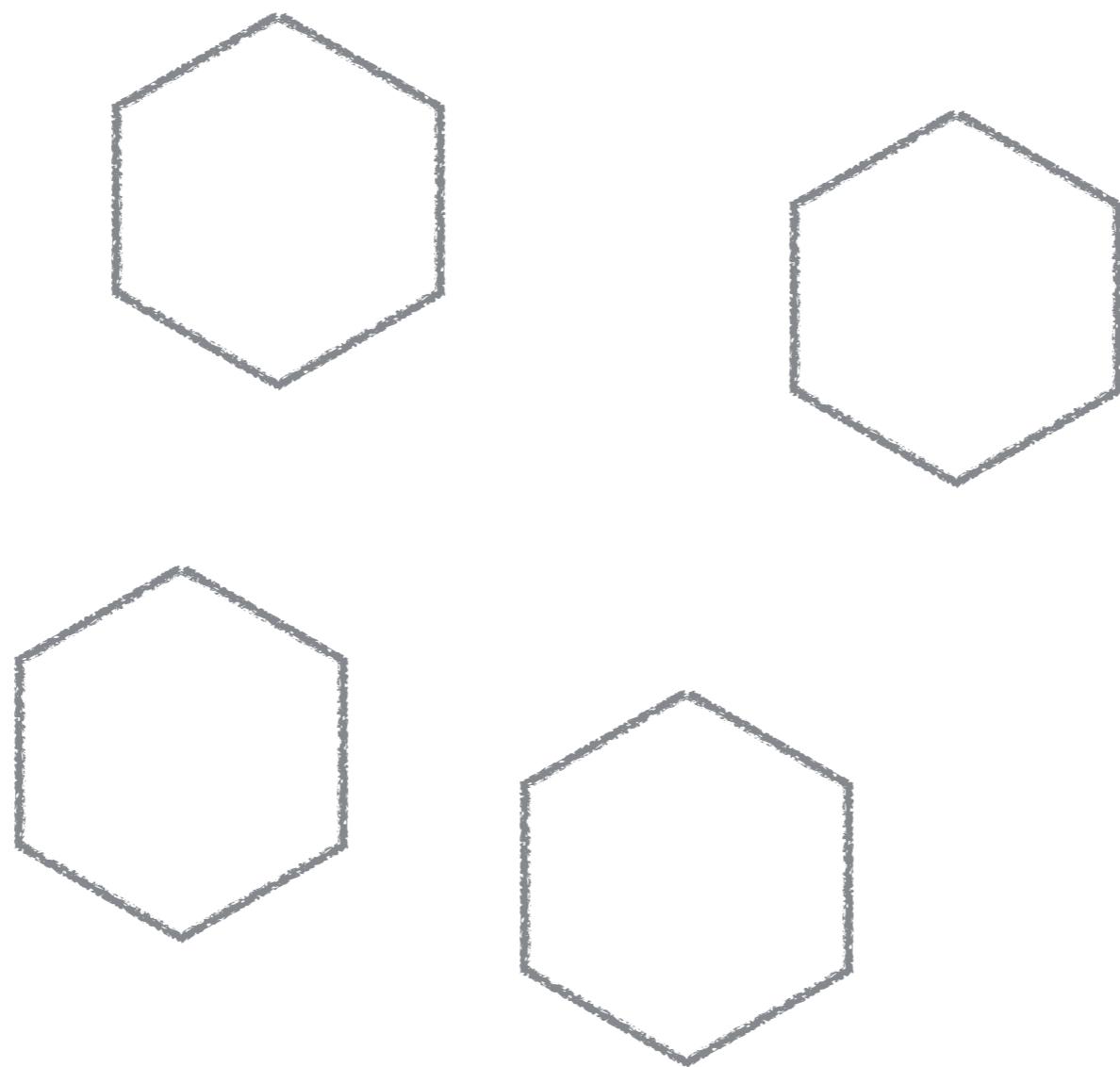


@javazone

@samnewman

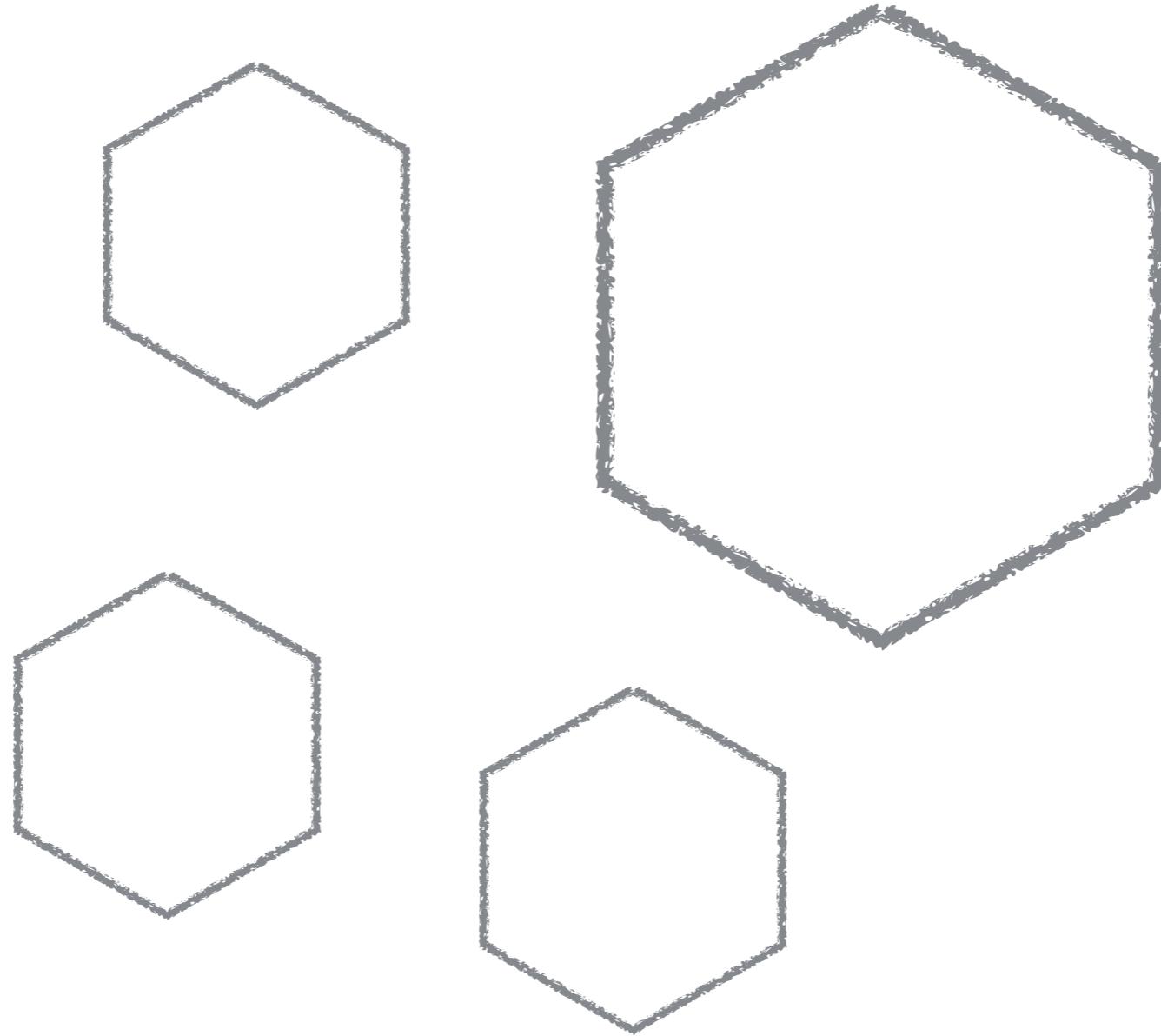






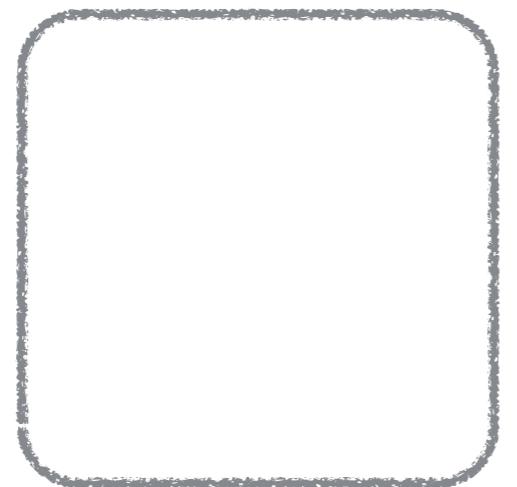
@javazone

@samnewman



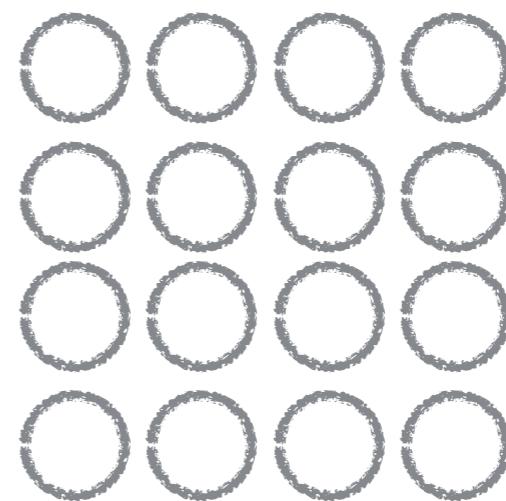
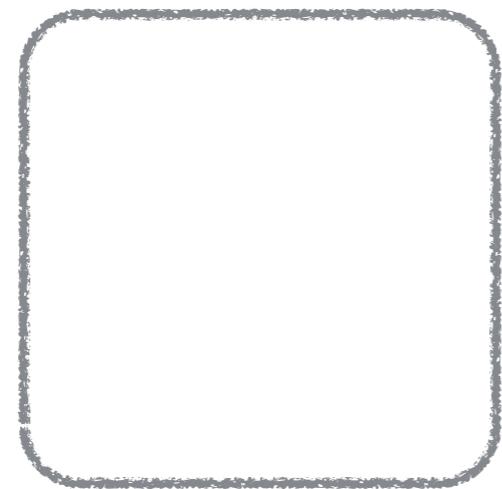
@javazone

@samnewman



@javazone

@samnewman



@javazone

@samnewman

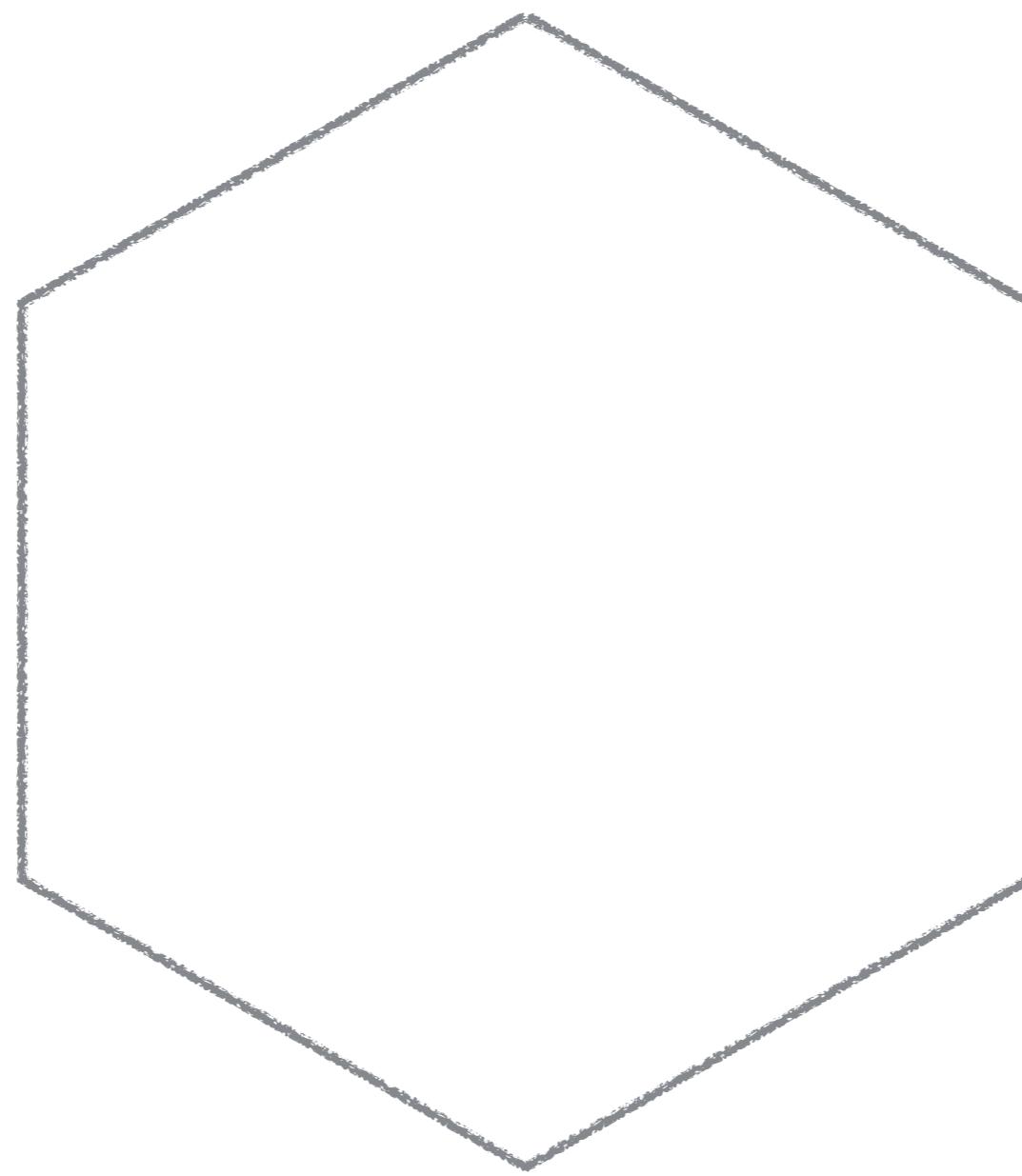
Where To Standardise?

@javazone

@samnewman

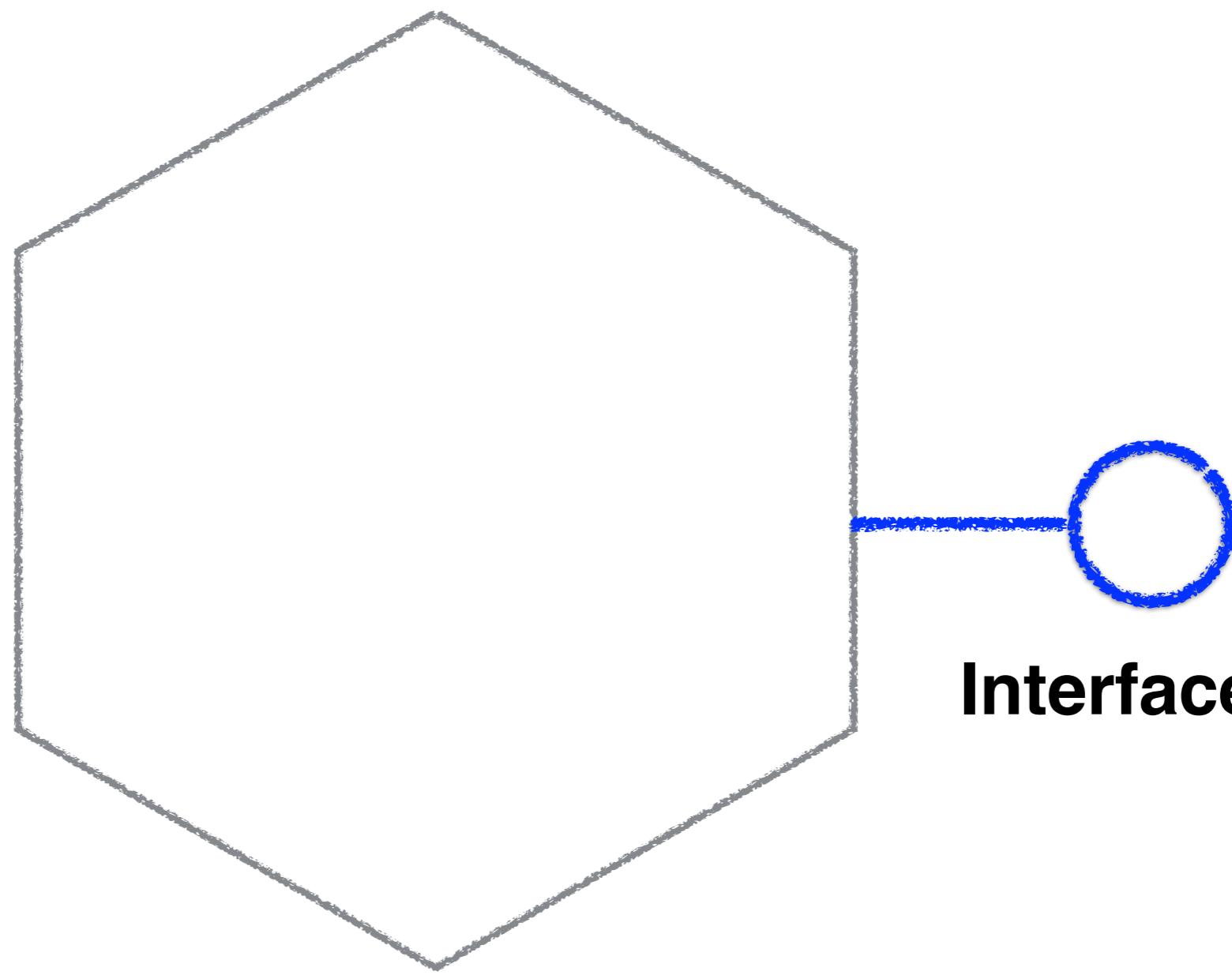


© 2013 Electronic Arts Inc.



@javazone

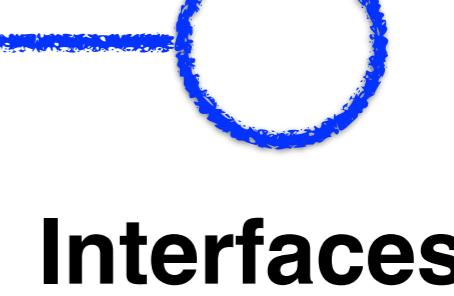
@samnewman



Interfaces



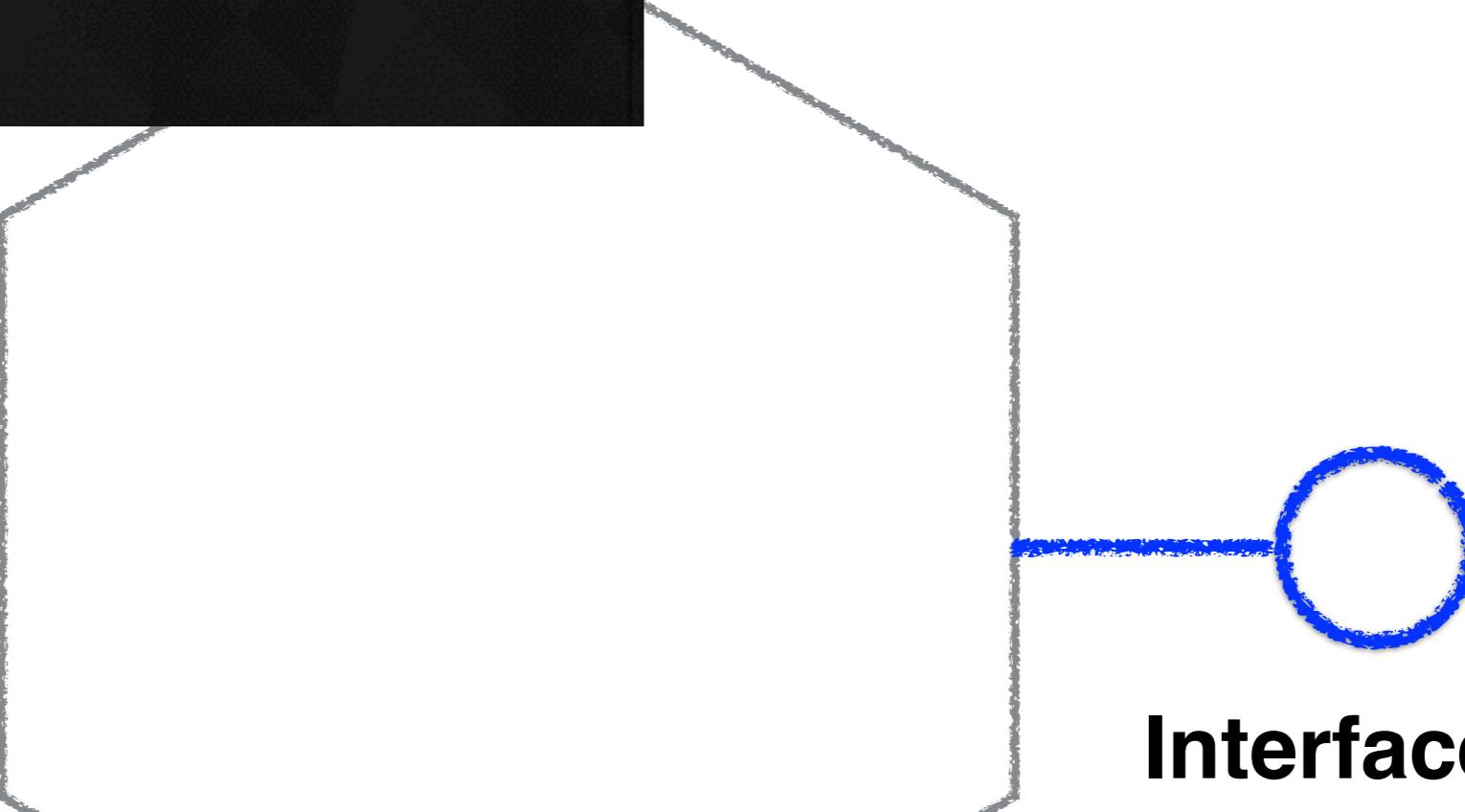
Monitoring



Interfaces



Monitoring

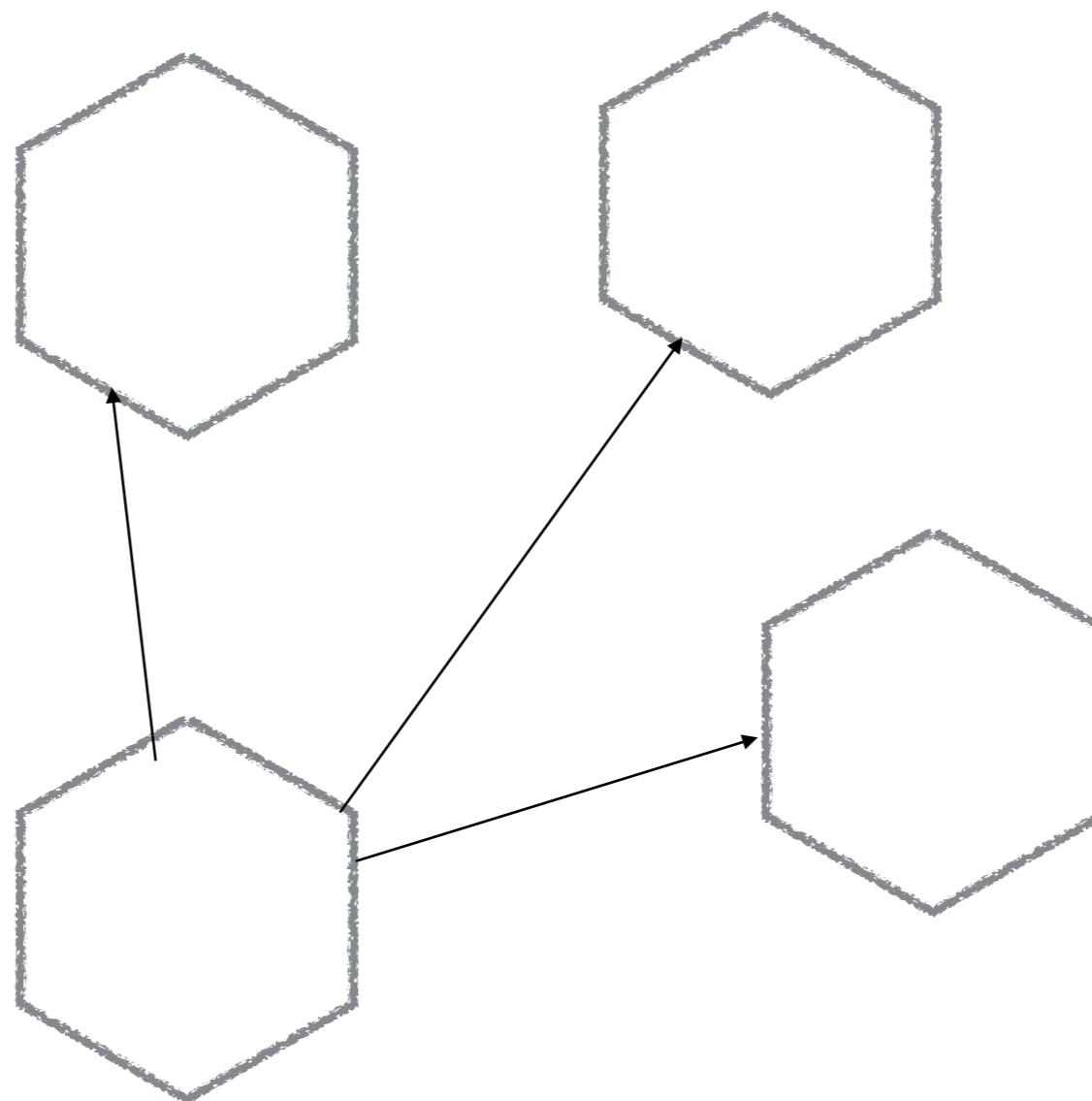


Interfaces

Deployment & Testing

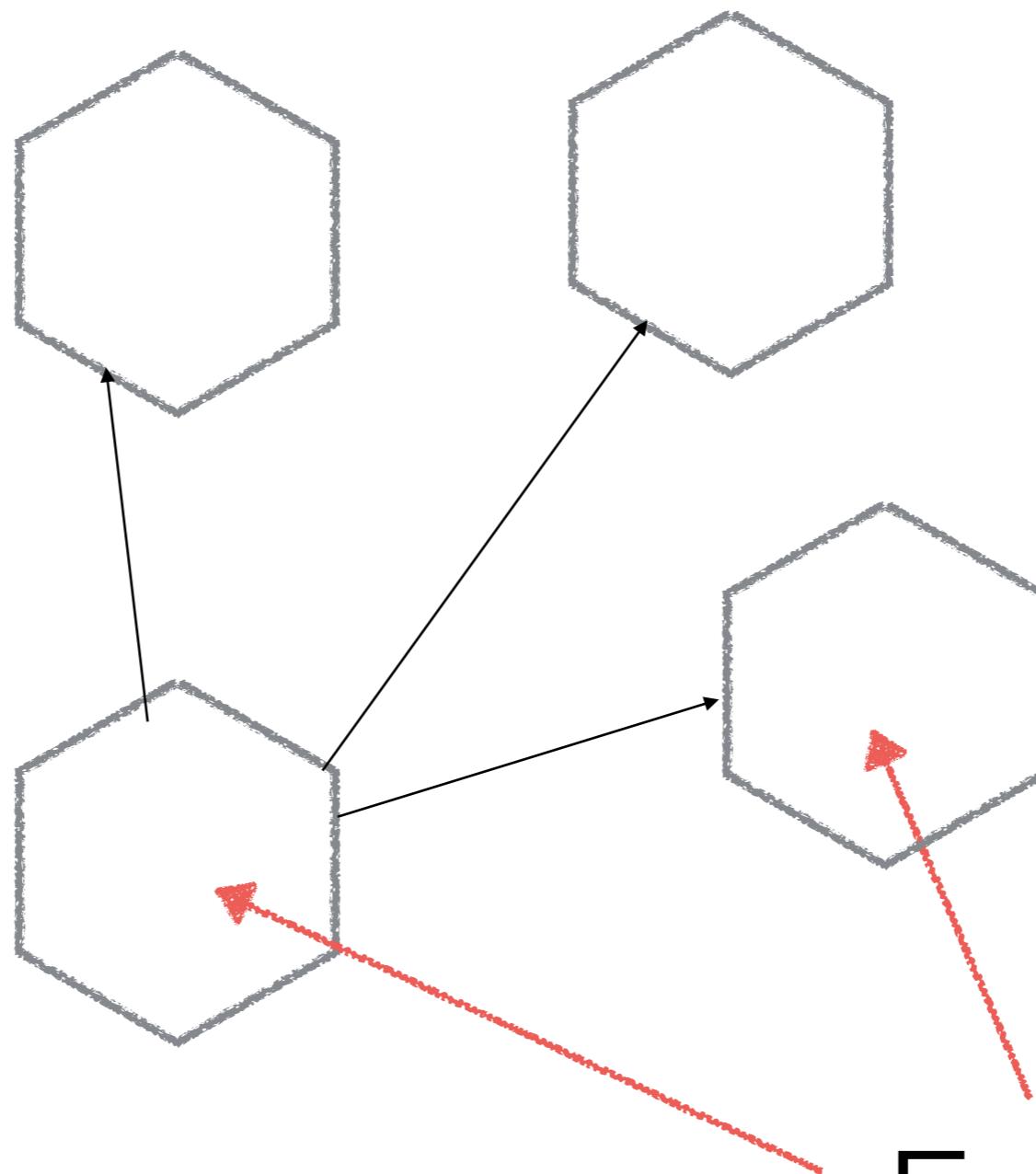
An aerial view of a dense urban landscape featuring numerous skyscrapers of varying heights and architectural styles. A prominent feature is a large, semi-transparent circular watermark centered over the city, which appears to be a stylized dome or a magnifying glass focusing on the central business district. The city is interspersed with green parks, roads with moving vehicles, and smaller buildings.

Architectural Safety



@javazone

@samnewman

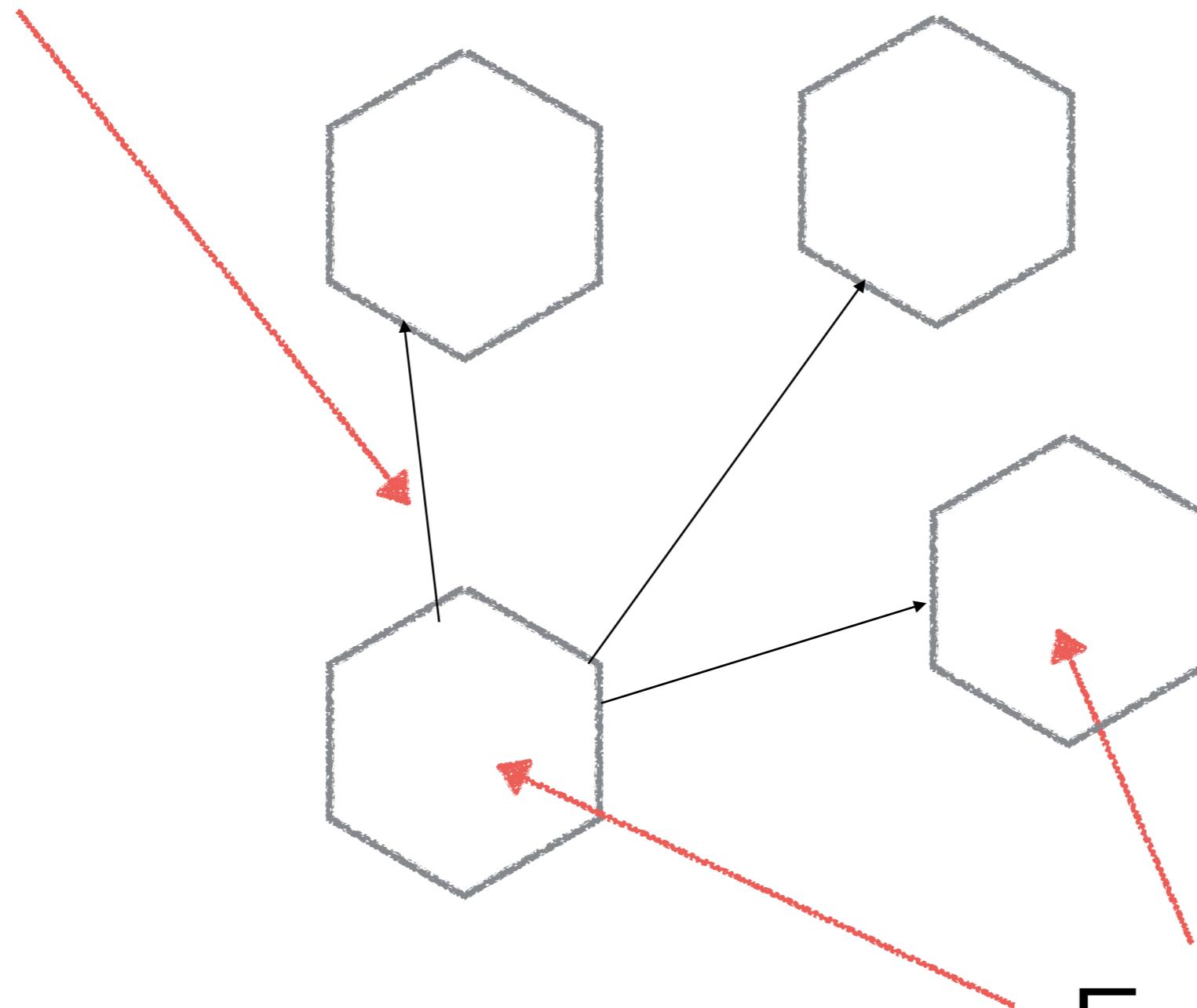


Free For All

@javazone

@samnewman

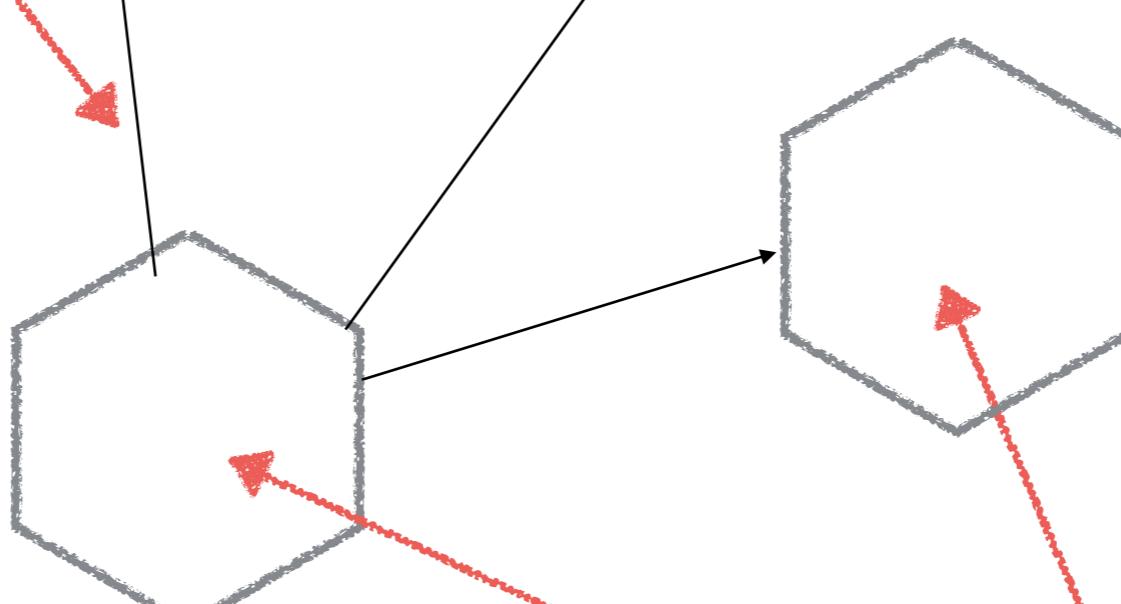
Standardisation



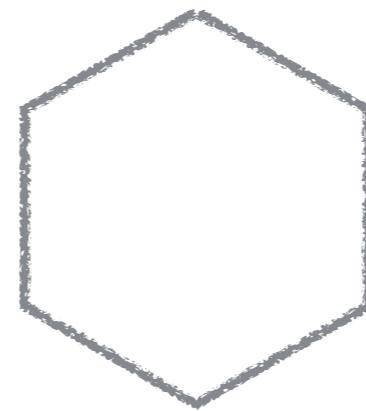
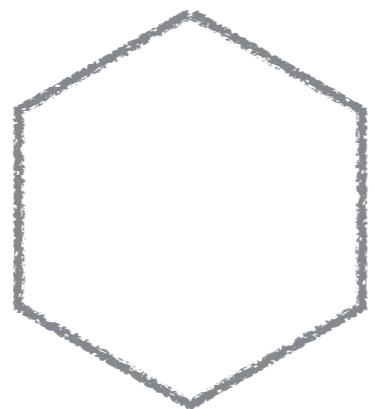
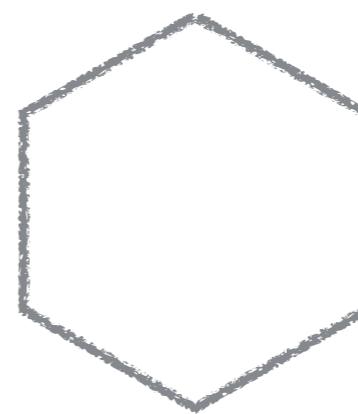
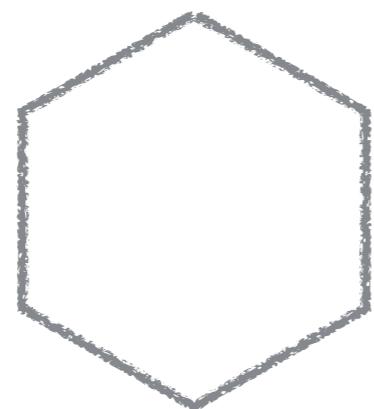
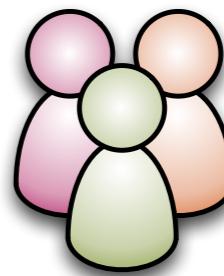
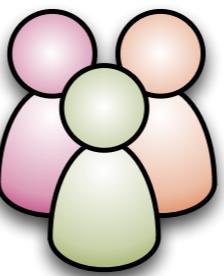
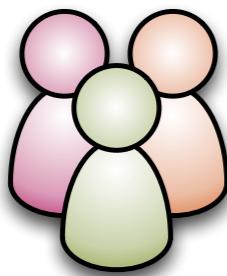
Free For All

Standardisation

TIP 1 / 14: Standardise in the gaps between services - be flexible about what happens inside the boxes

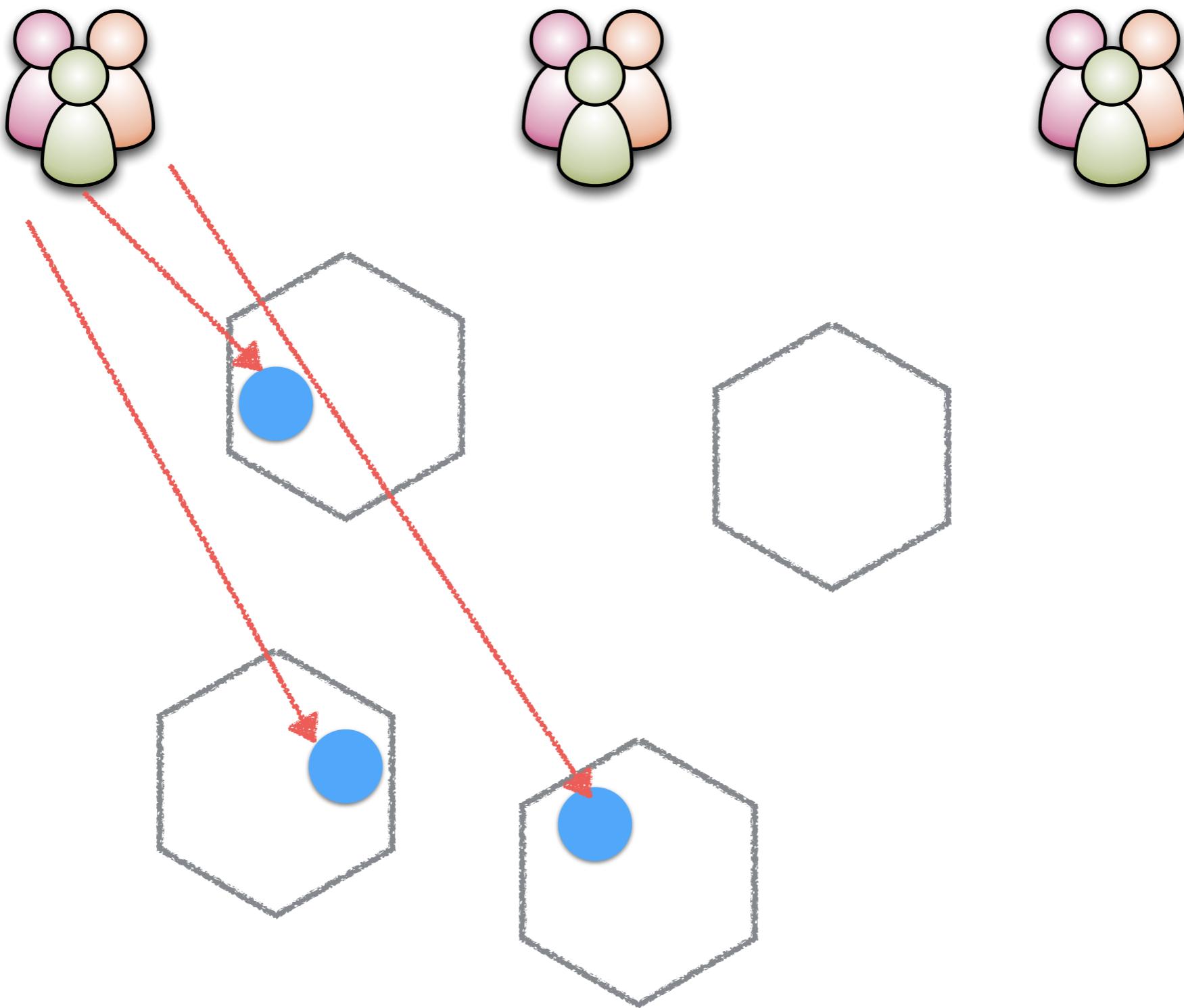


Free For All



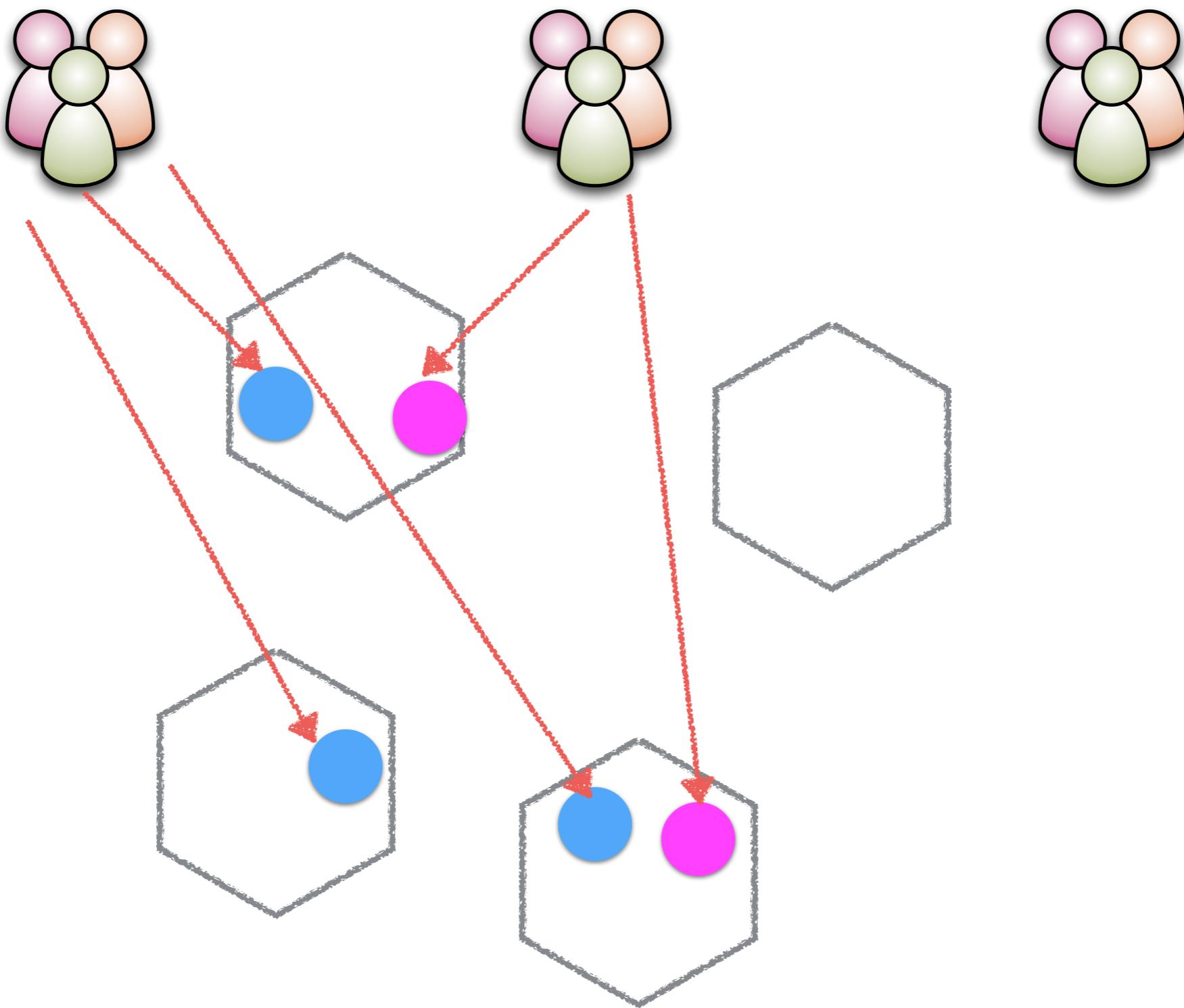
@javazone

@samnewman



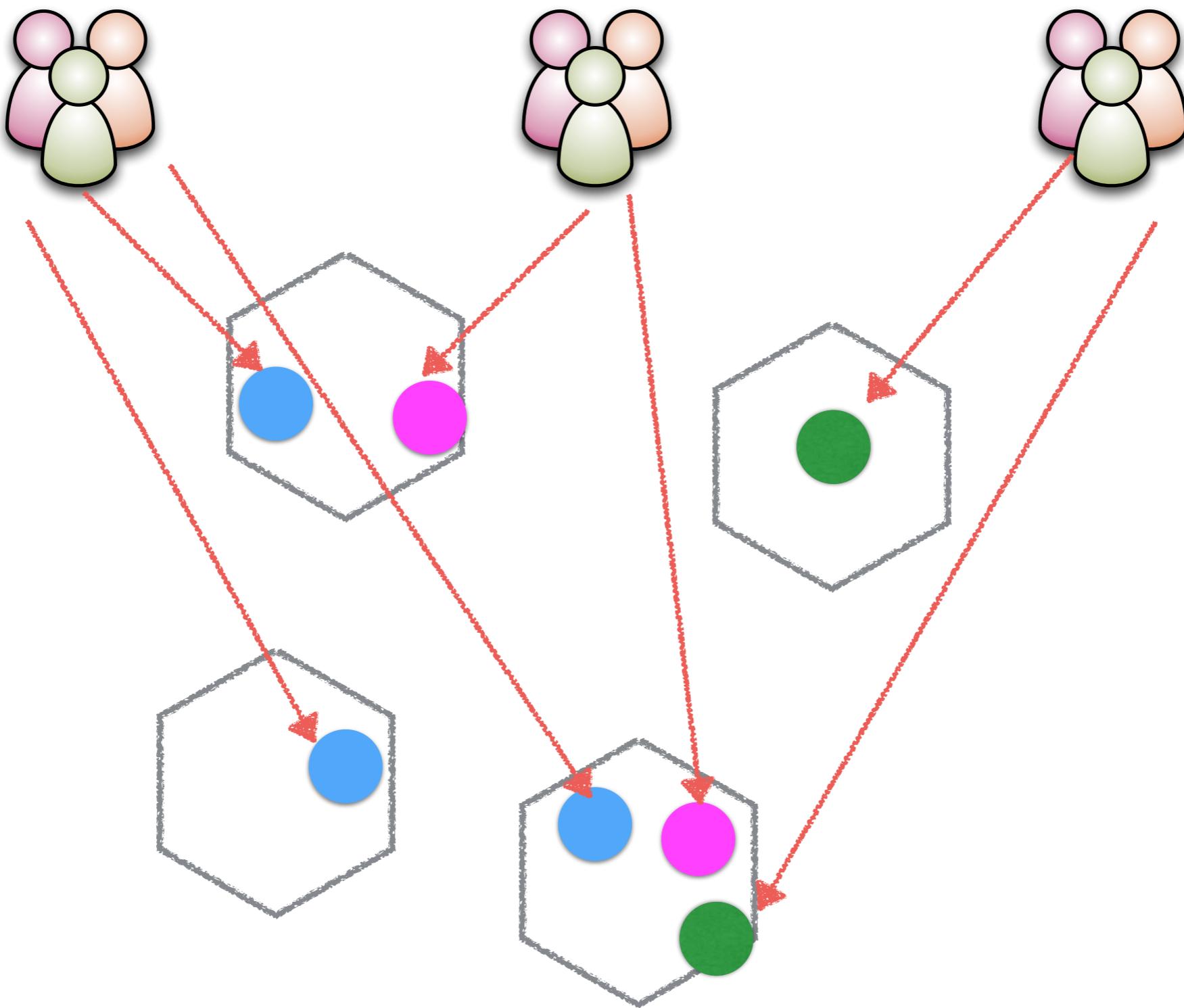
@javazone

@samnewman



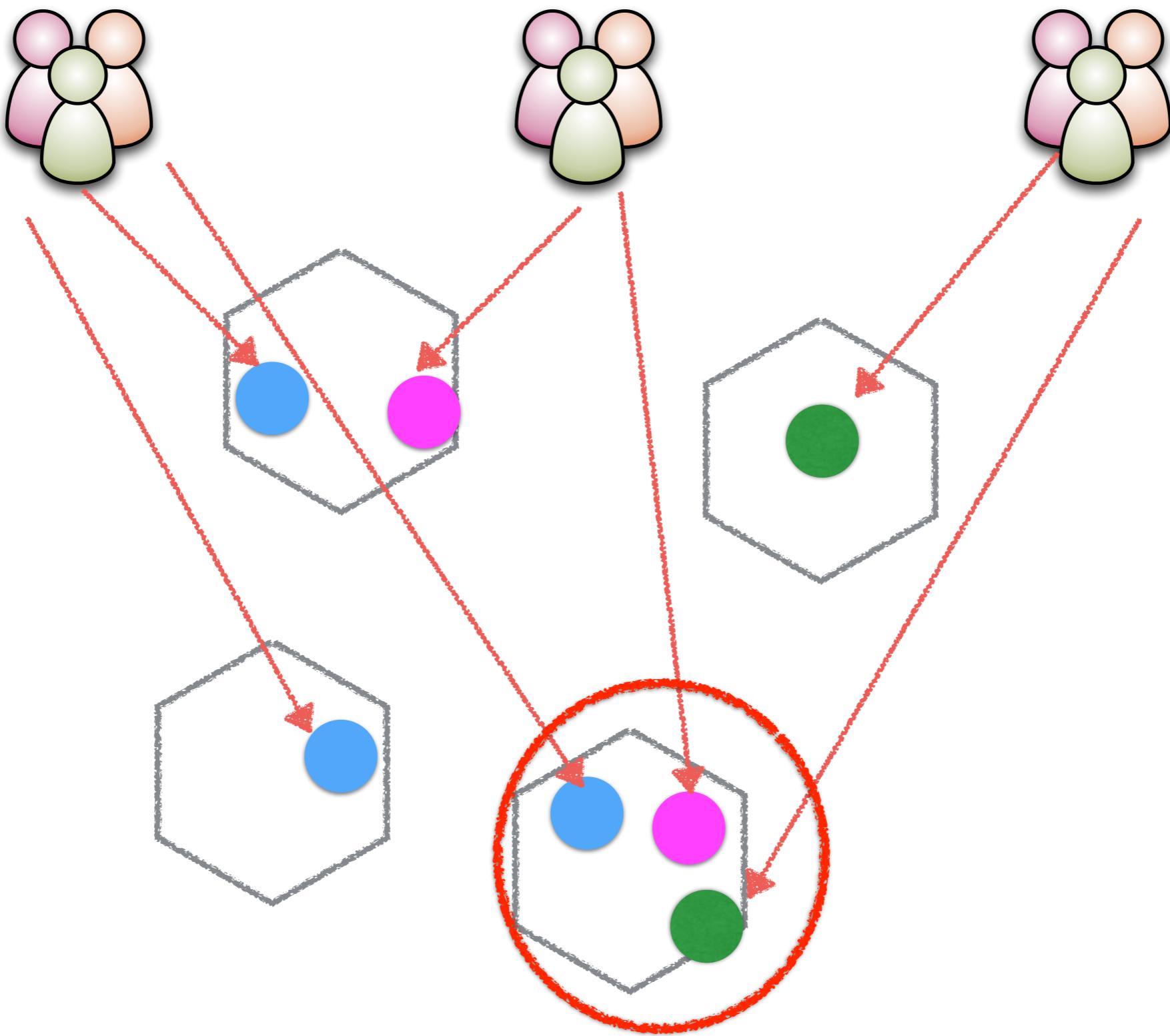
@javazone

@samnewman



@javazone

@samnewman



@javazone

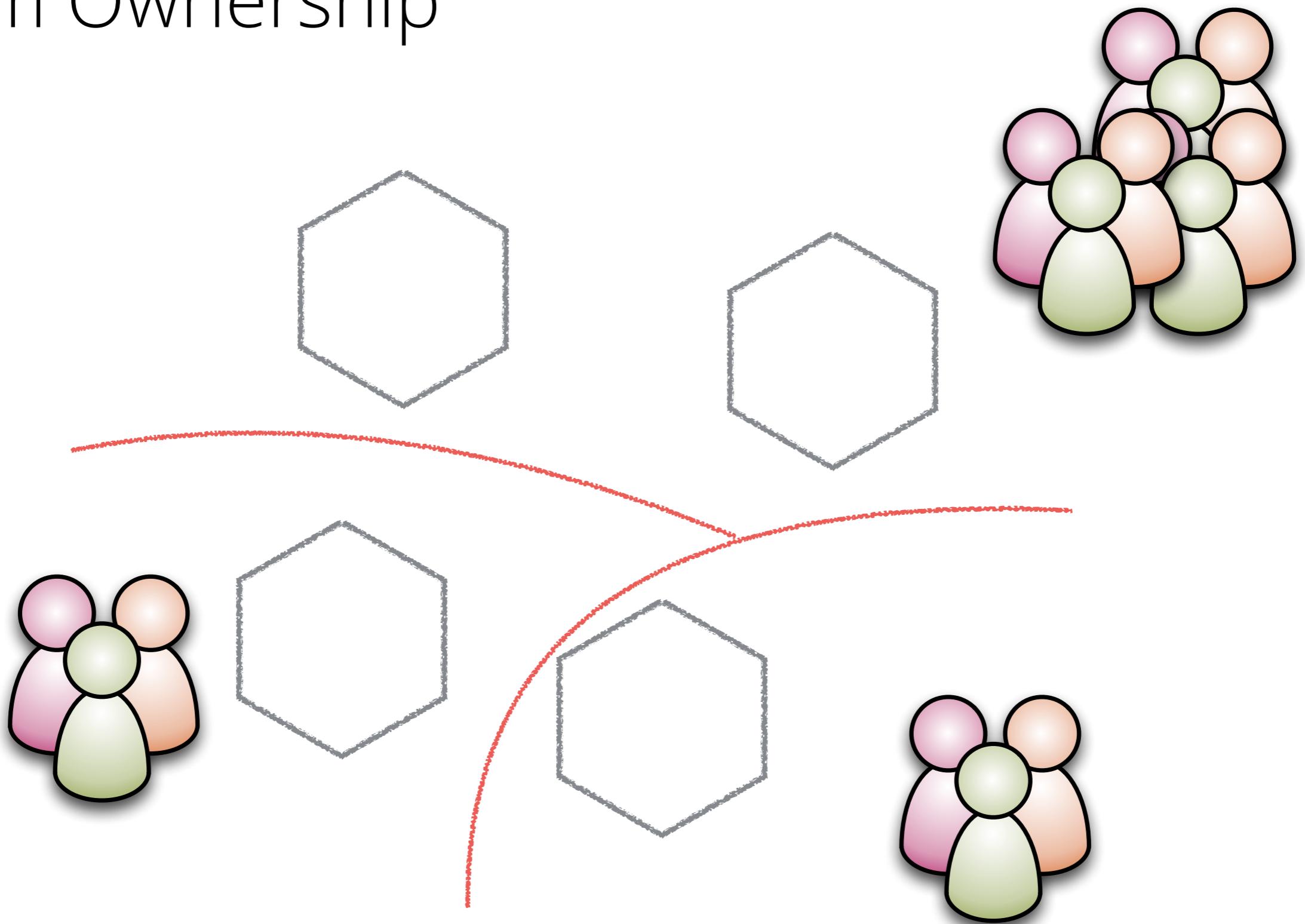
@samnewman



Autonomy

Communication

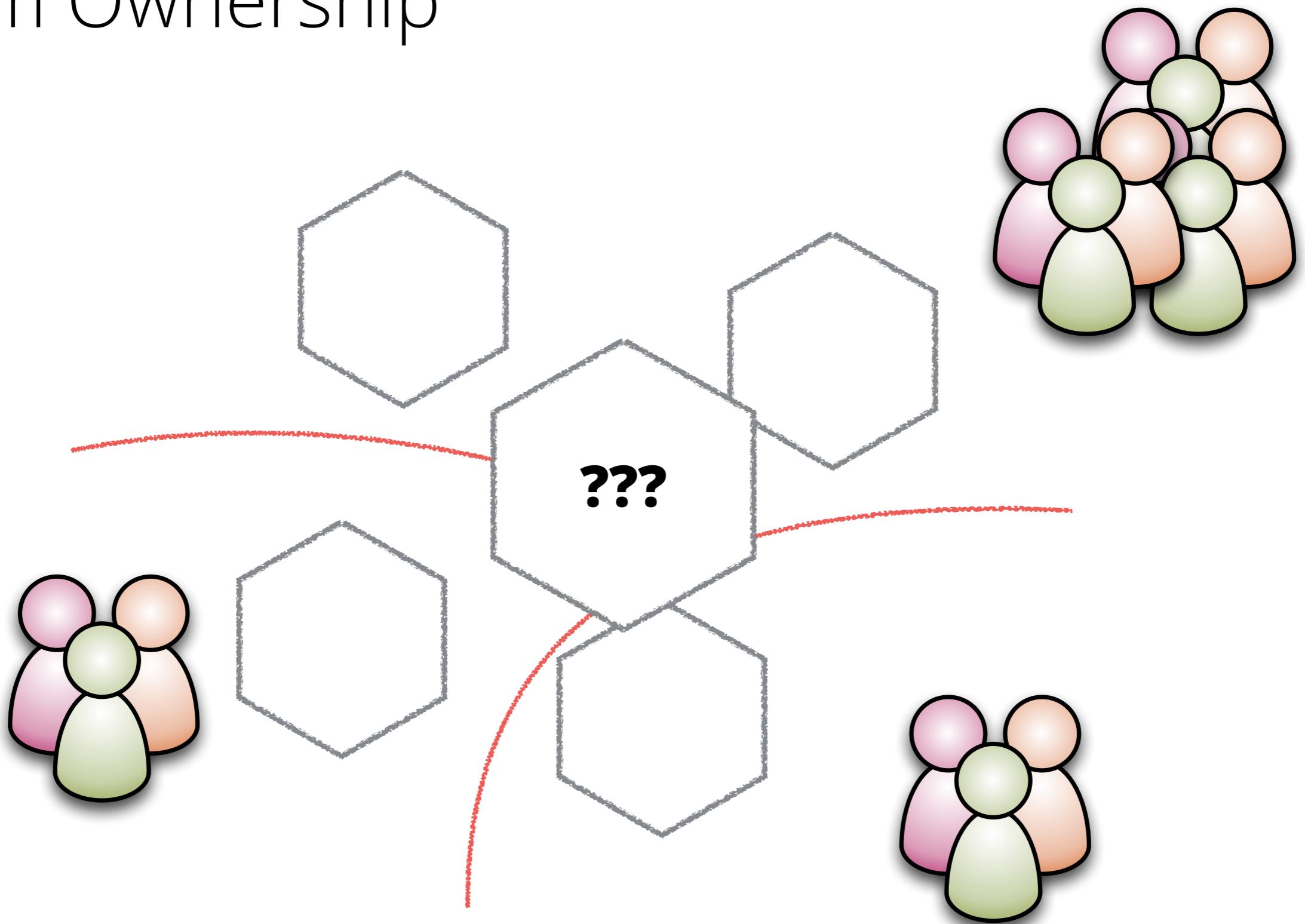
Team Ownership



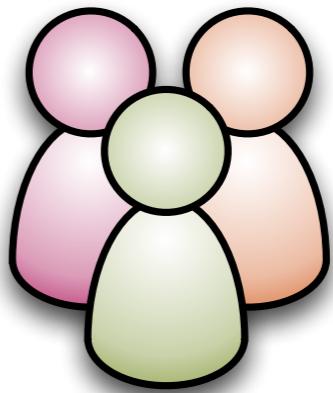
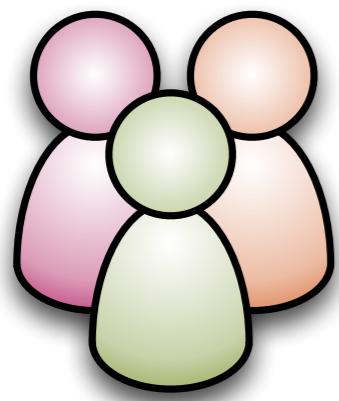
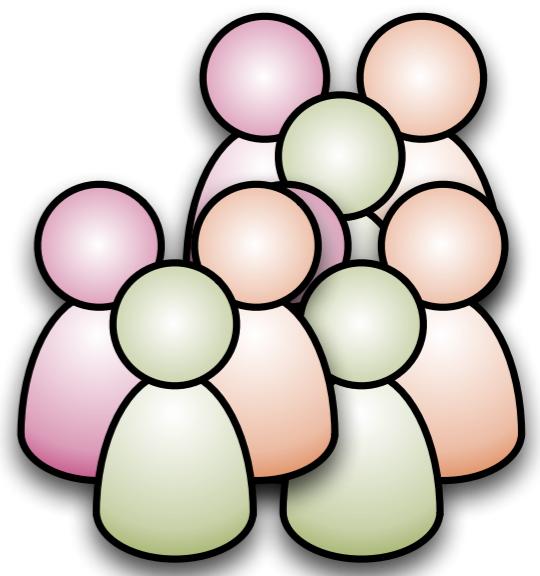
@javazone

@samnewman

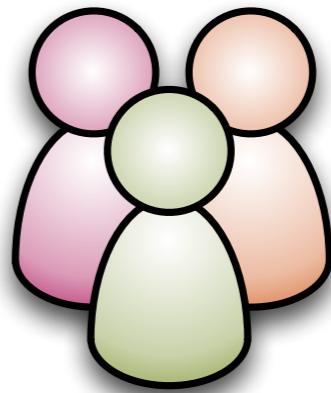
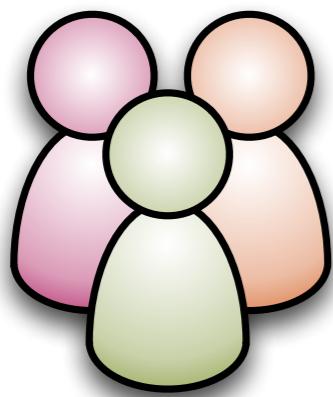
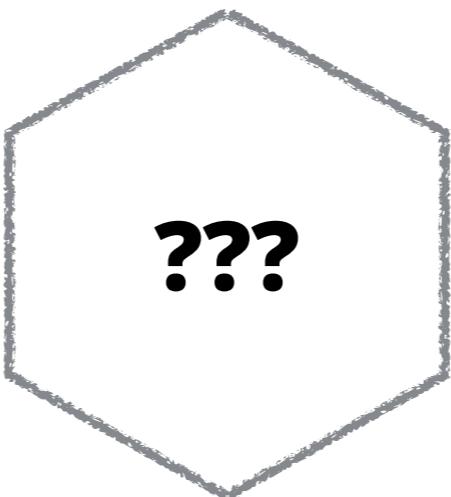
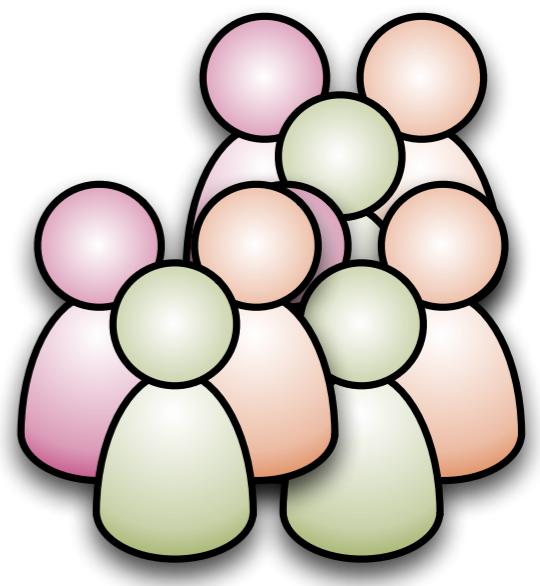
Team Ownership



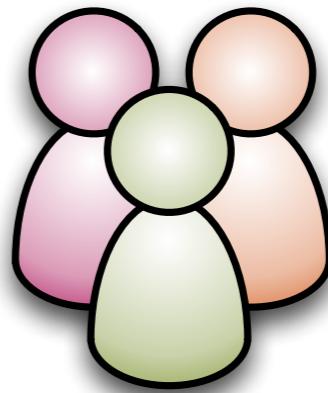
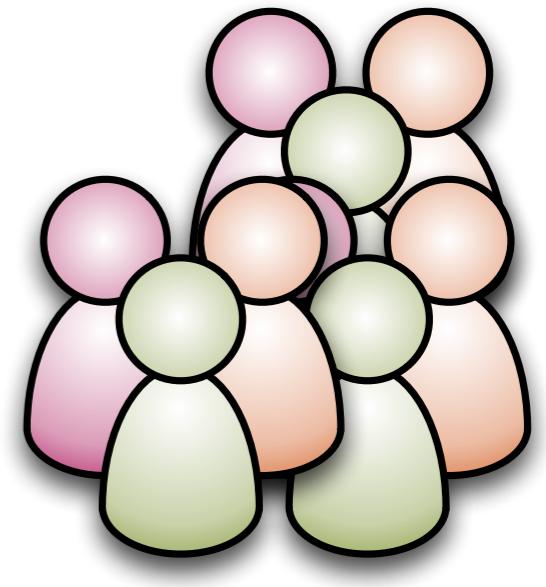
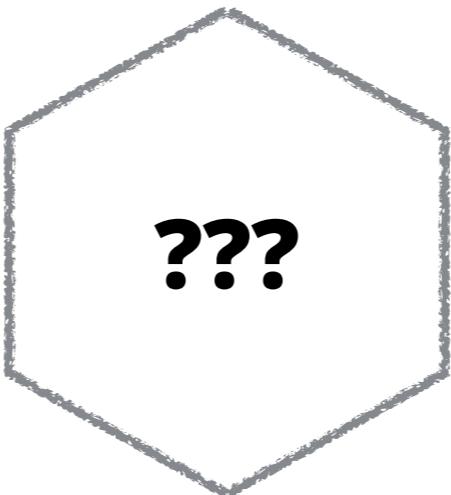
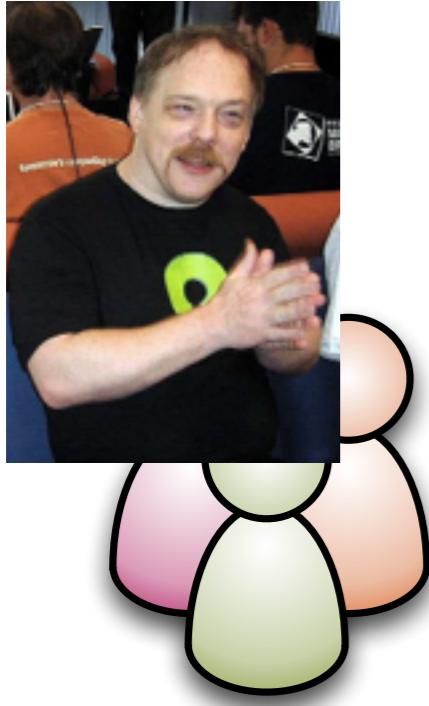
Trusted Committer / Roving Custodian



Trusted Committer / Roving Custodian



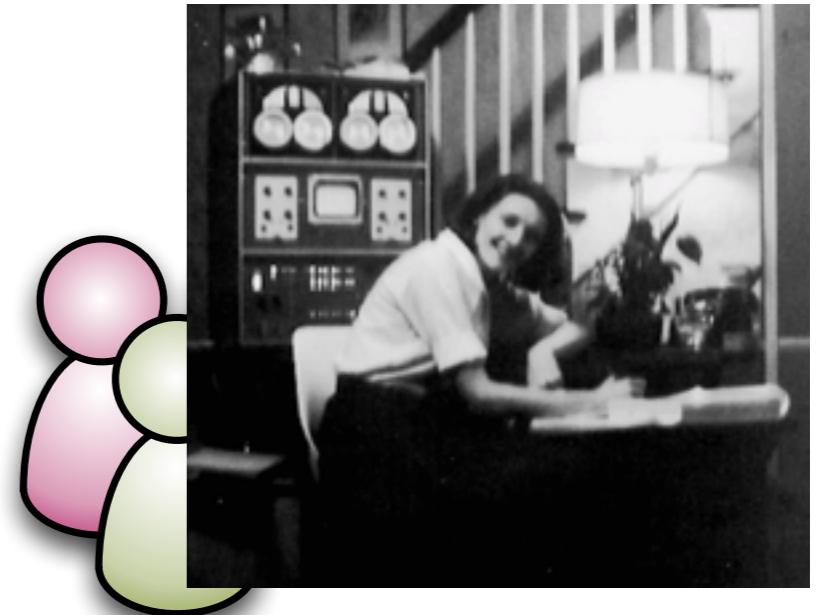
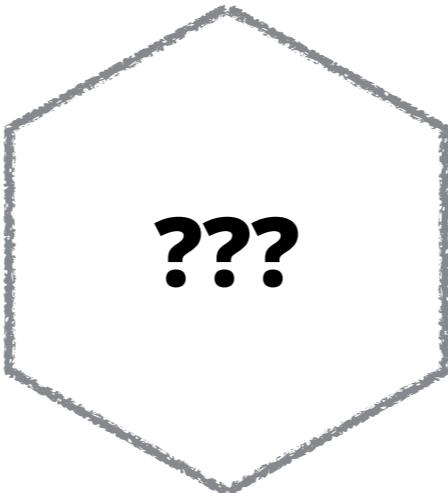
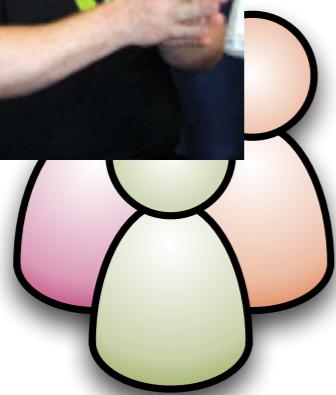
Trusted Committer / Roving Custodian



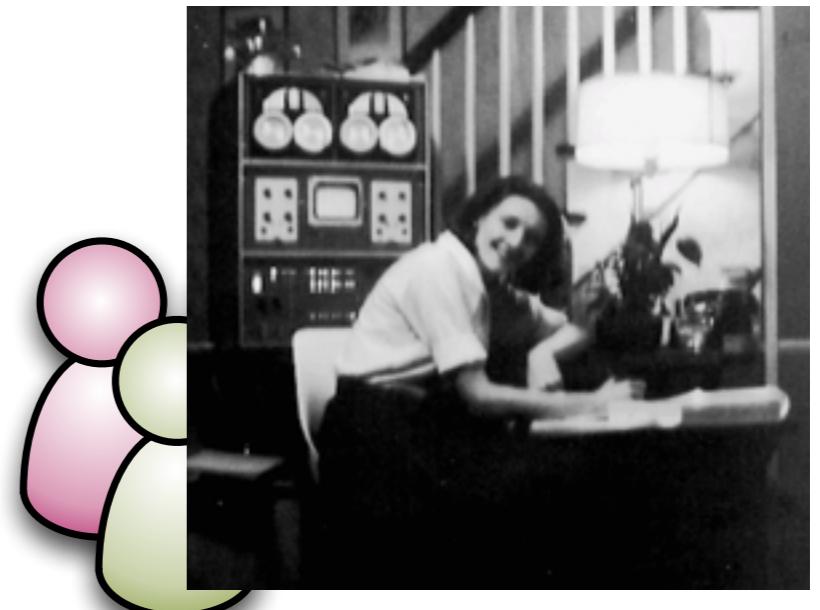
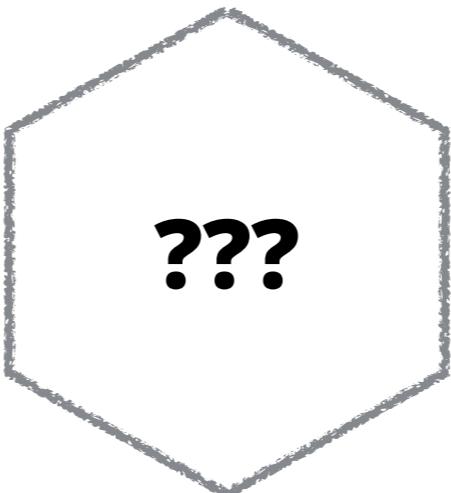
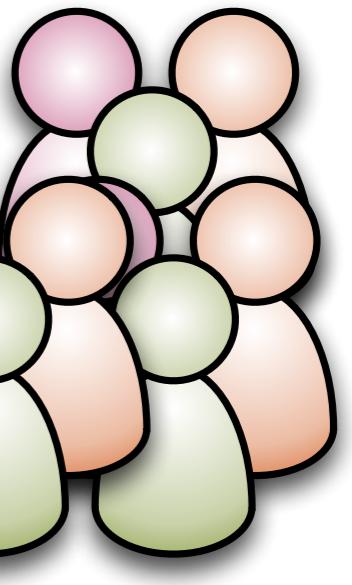
@javazone

@samnewman

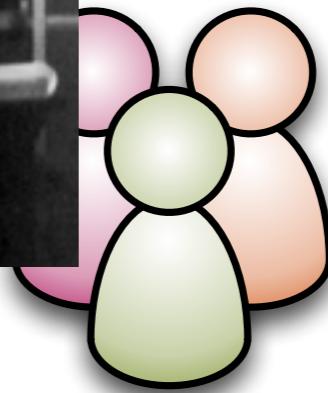
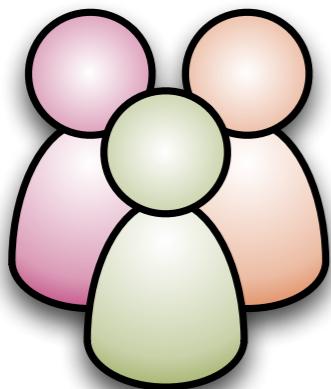
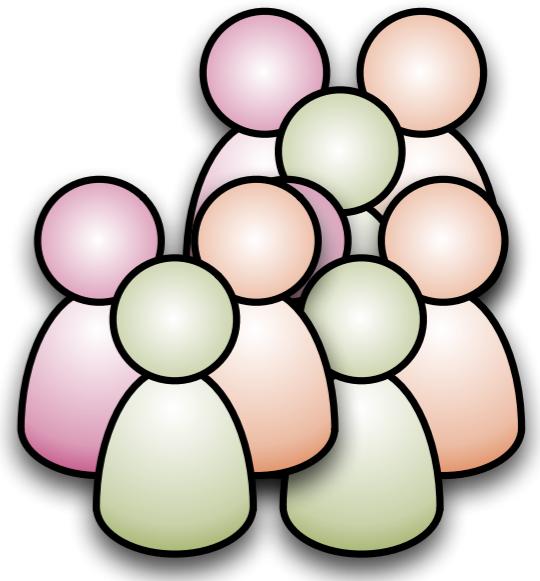
Trusted Committer / Roving Custodian



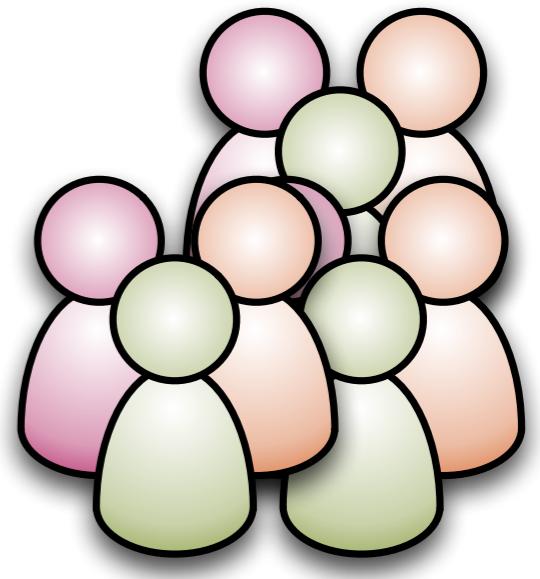
Trusted Committer / Roving Custodian



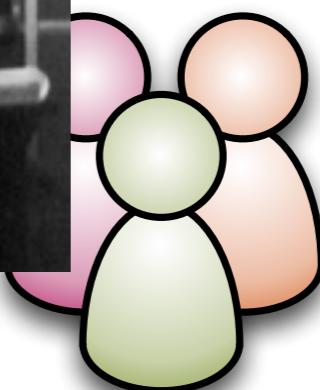
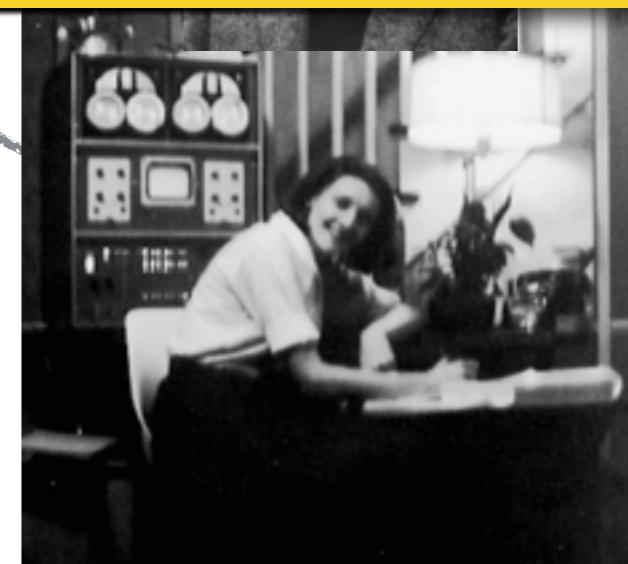
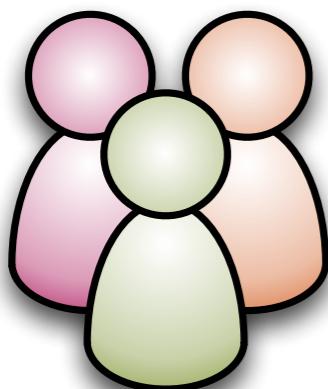
Trusted Committer / Roving Custodian



Trusted Committer / Roving Custodian



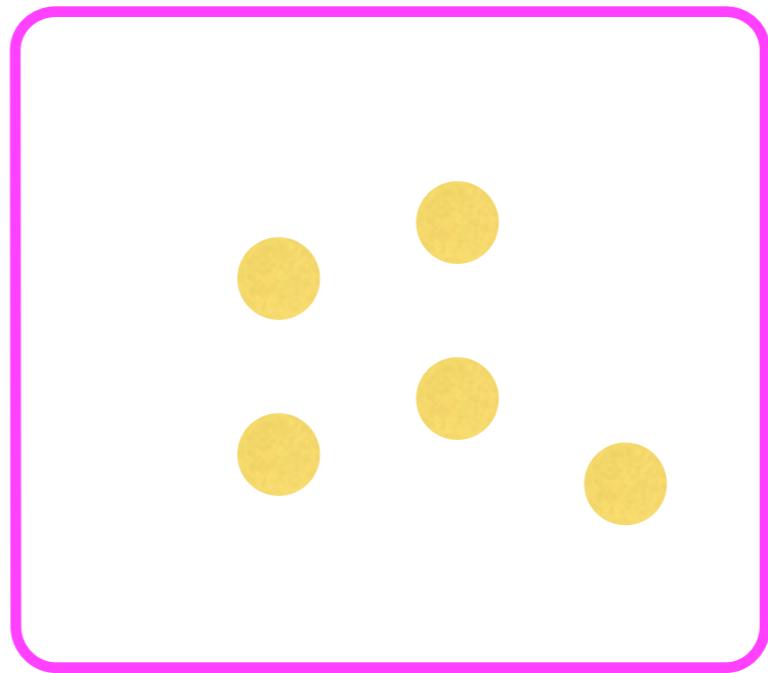
TIP 2 / 14: Have a clear custodian model!



Interfaces

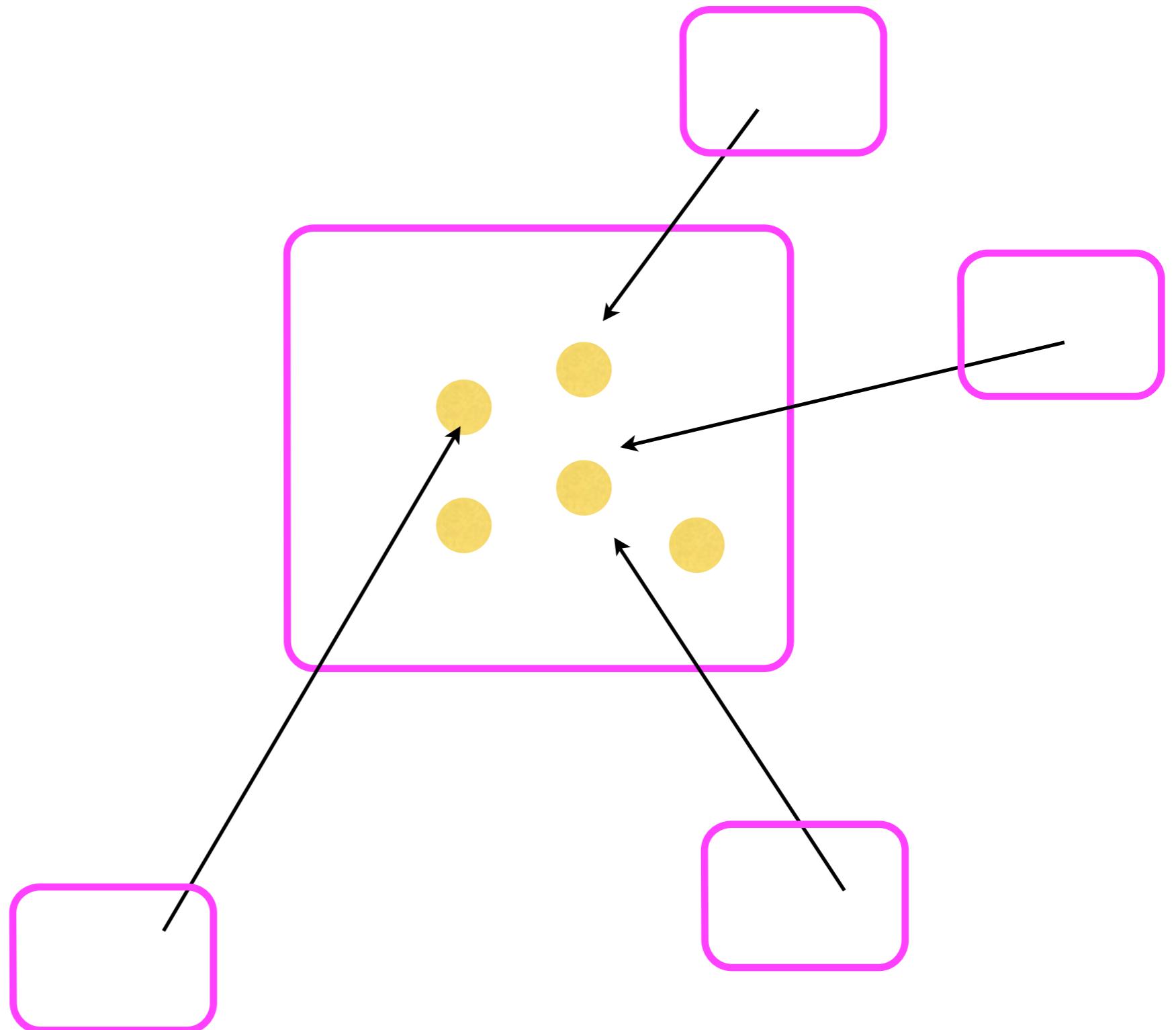
@javazone

@samnewman



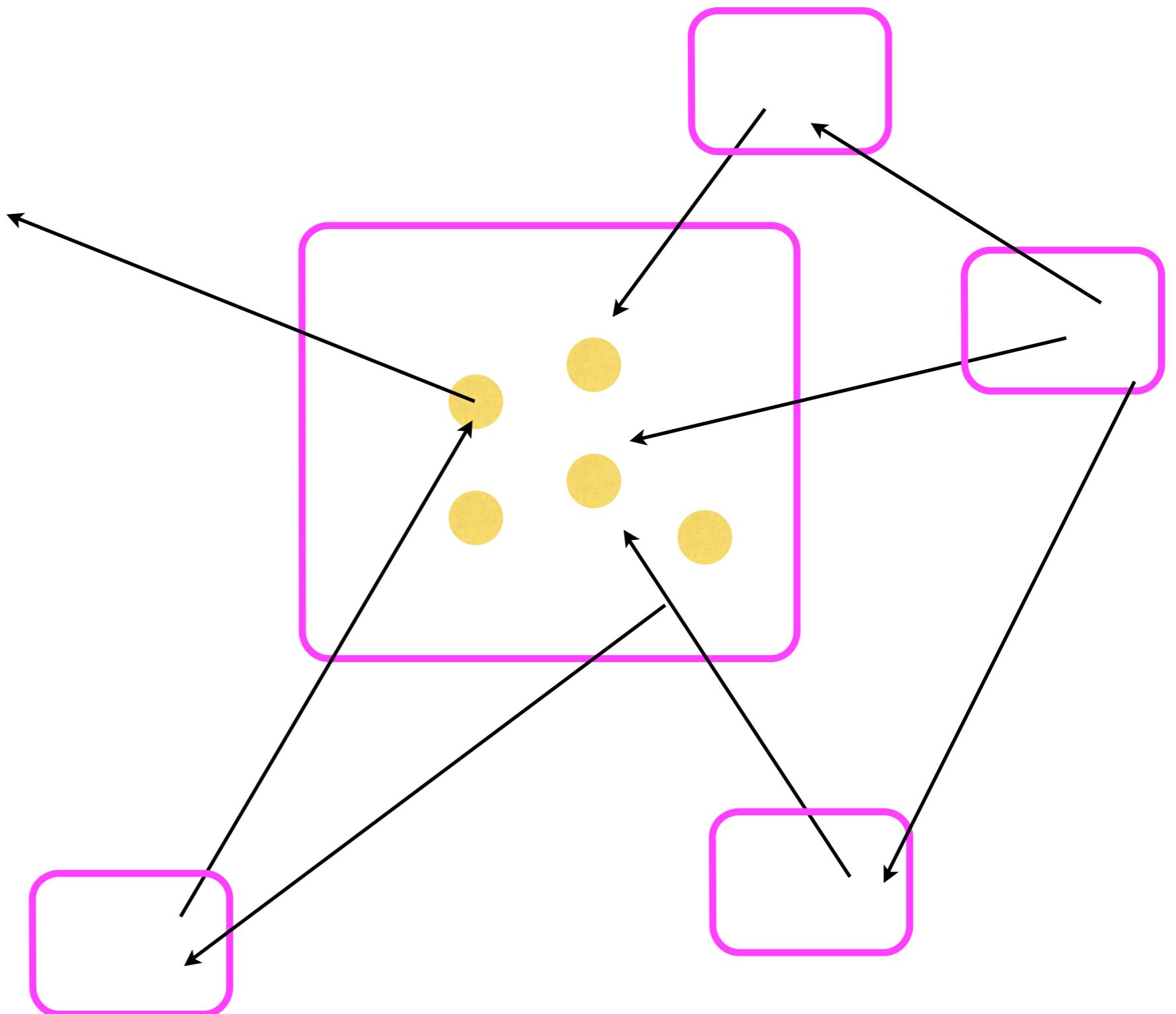
@javazone

@samnewman



@javazone

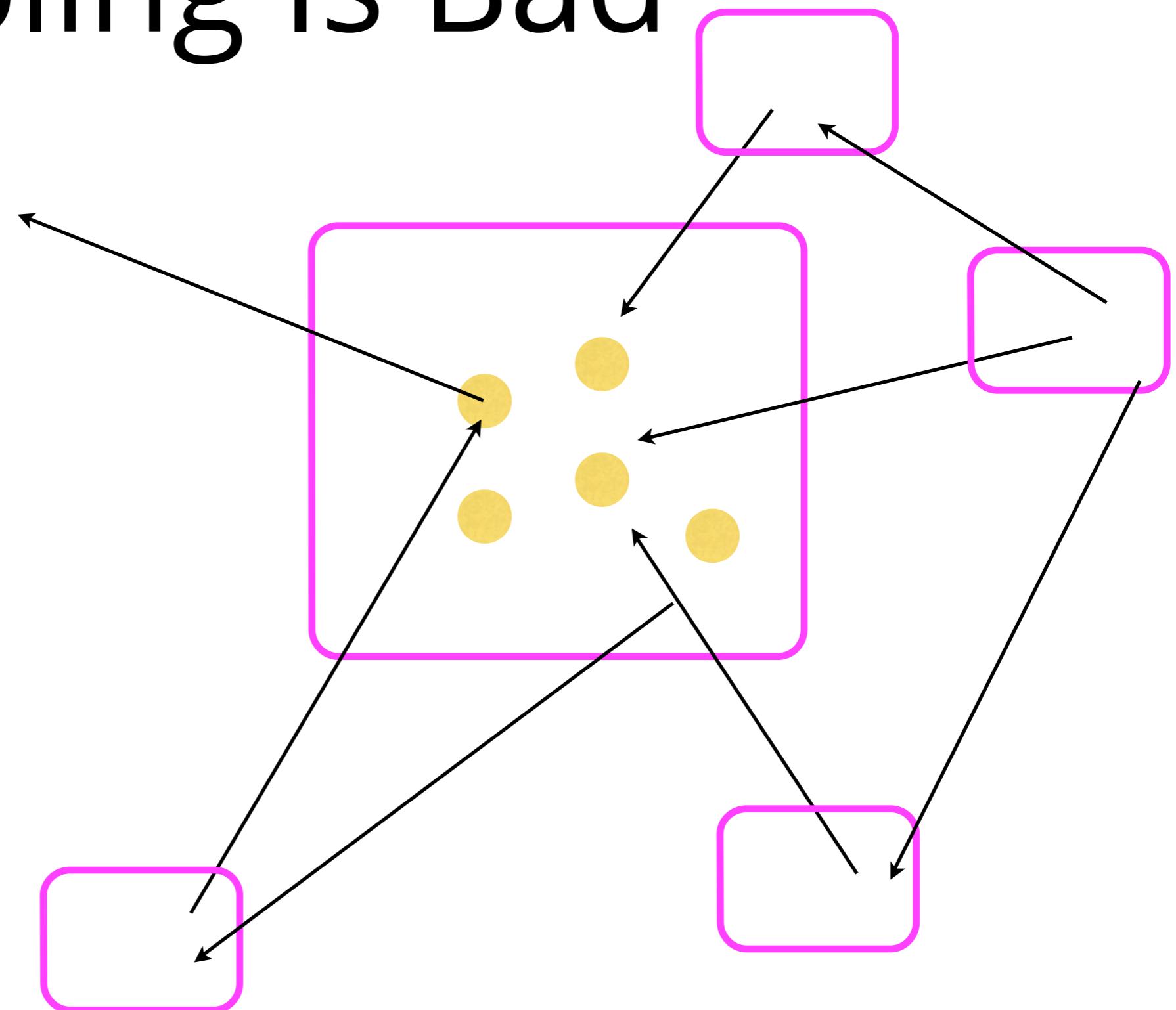
@samnewman



@javazone

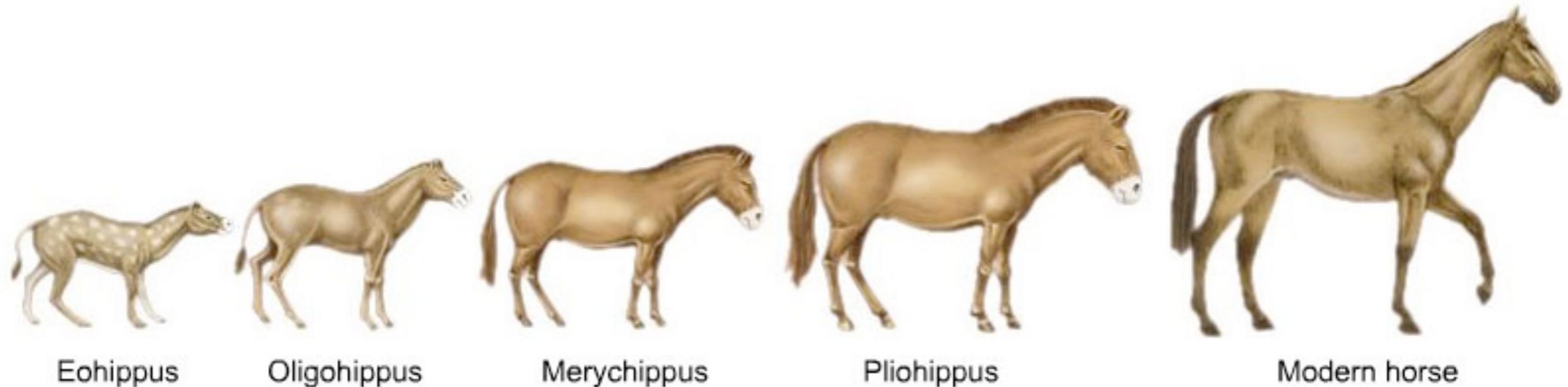
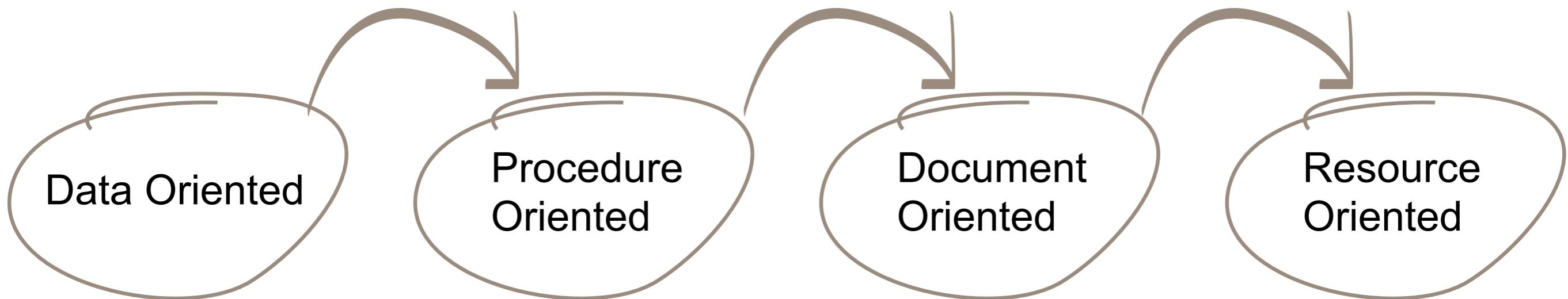
@samnewman

Coupling Is Bad



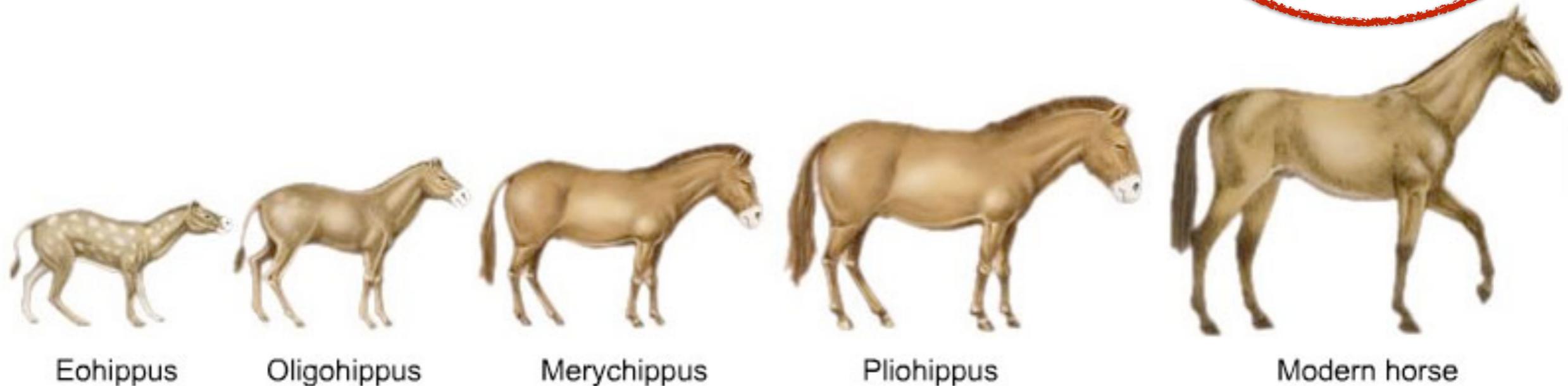
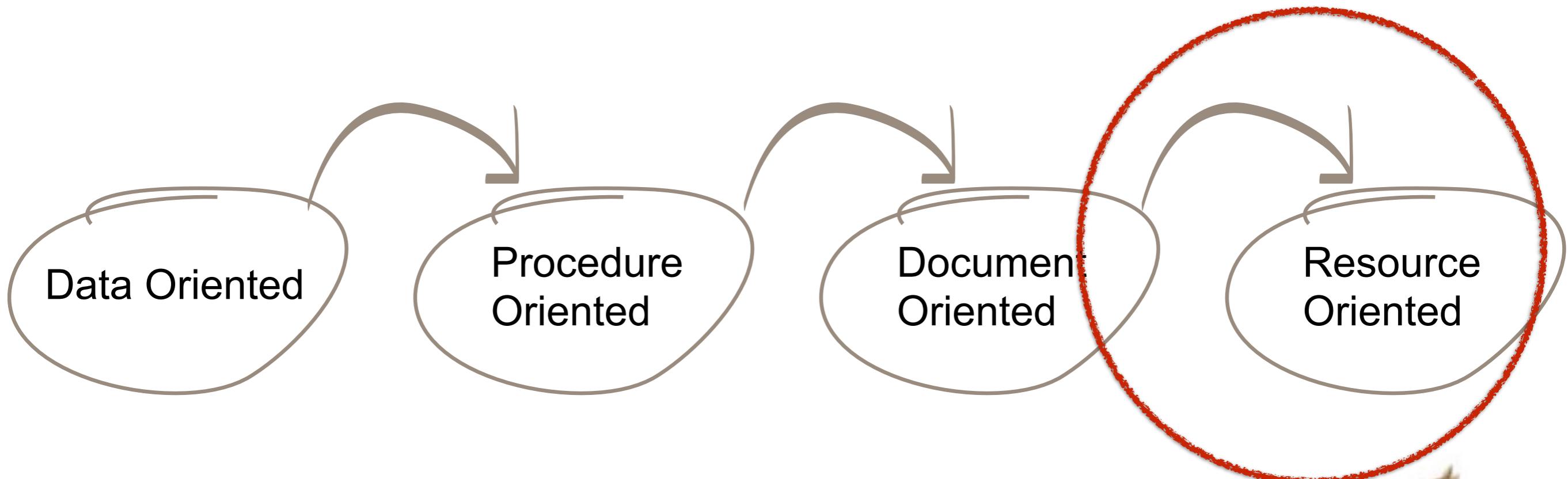
Integration Styles

An Evolutionary View



Integration Styles

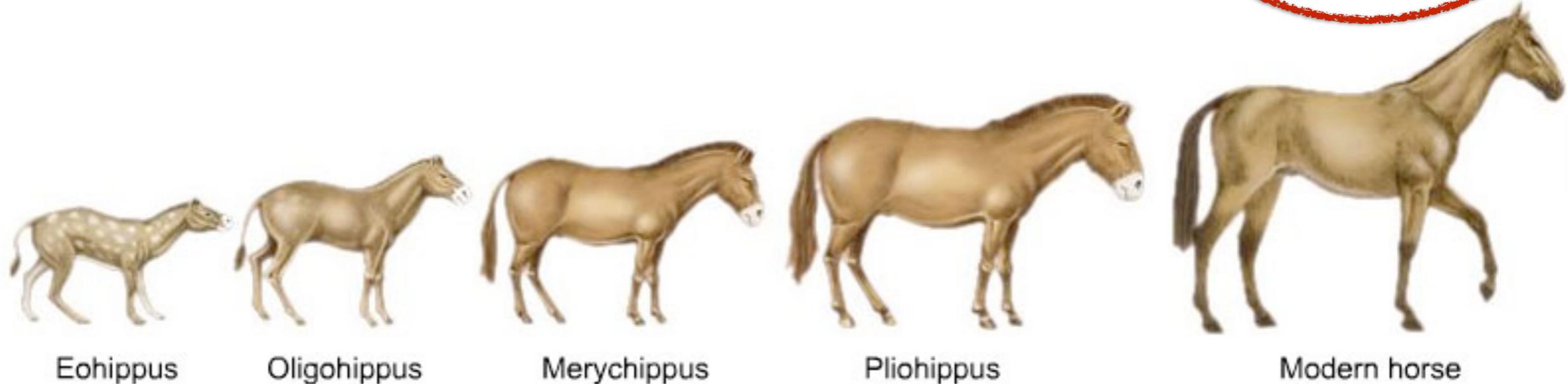
An Evolutionary View

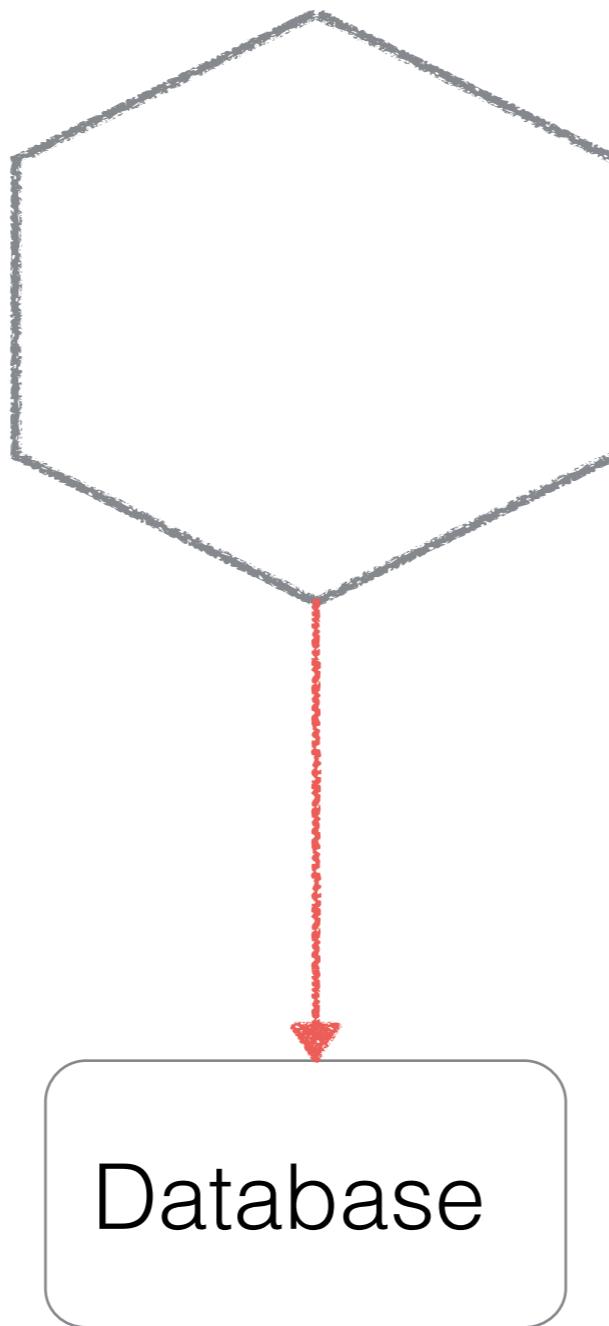


Integration Styles

An Evolutionary View

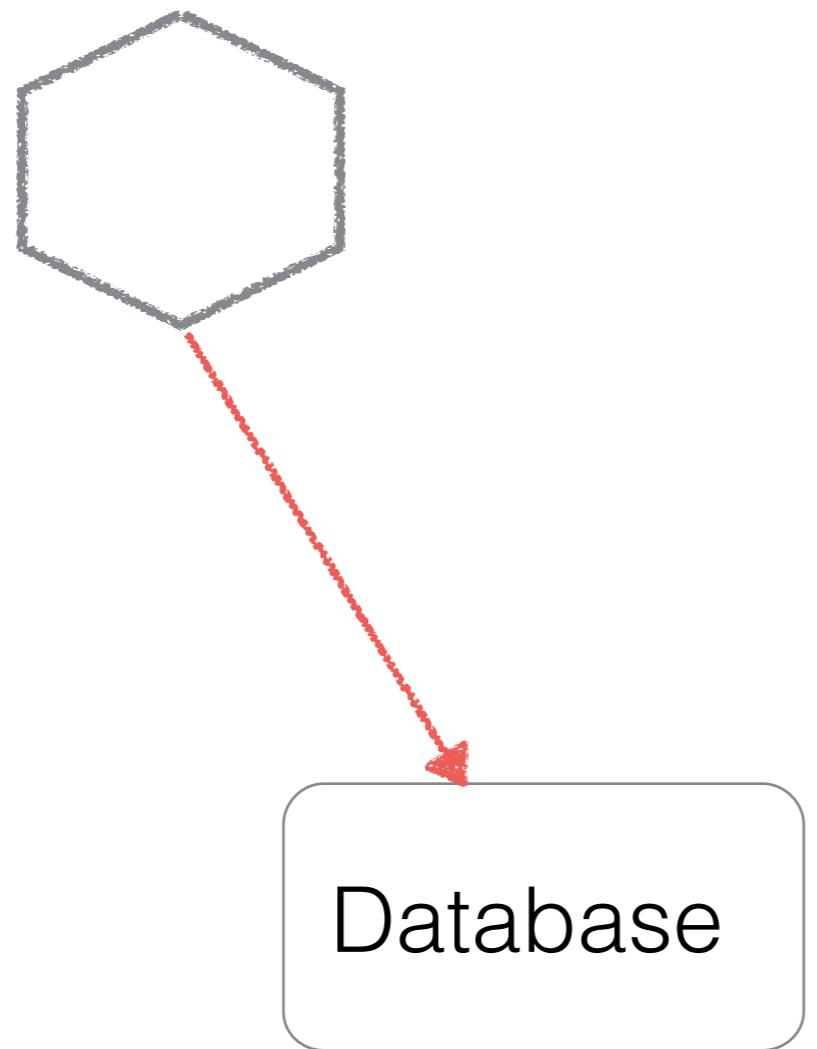
TIP 3 / 14: Avoid RPC-mechanisms or shared
Data (serialisation protocols to avoid coupling)





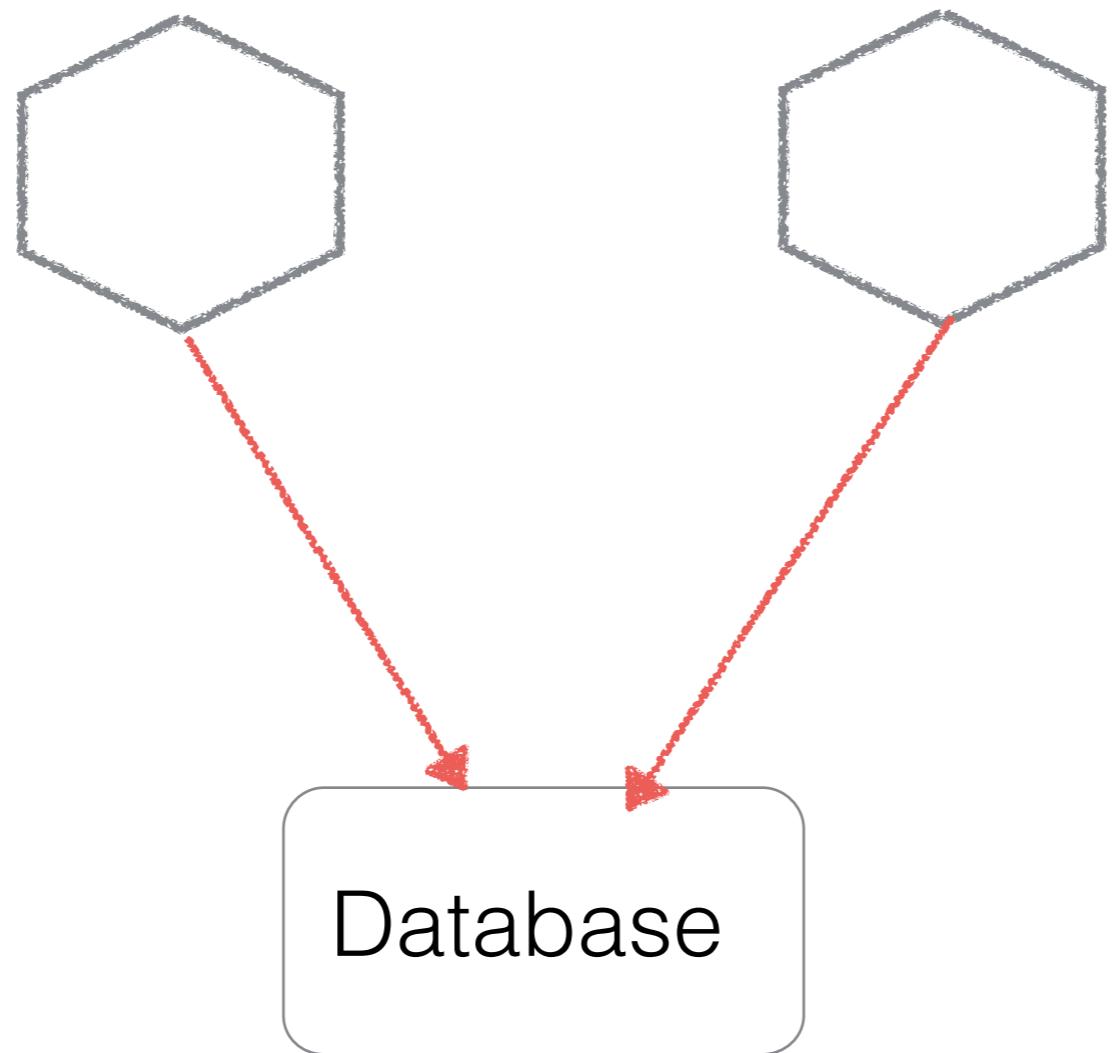
@javazone

@samnewman



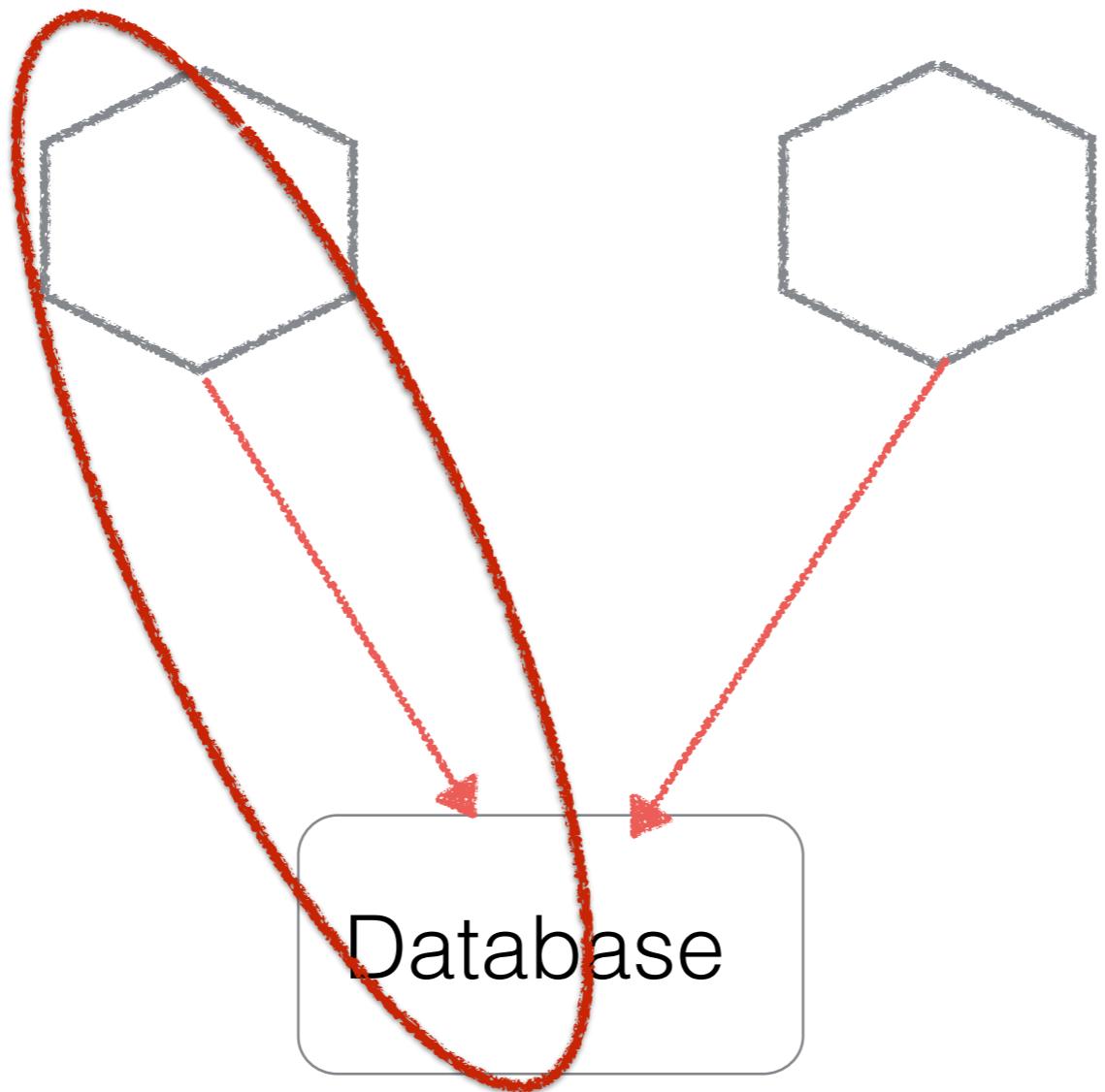
@javazone

@samnewman



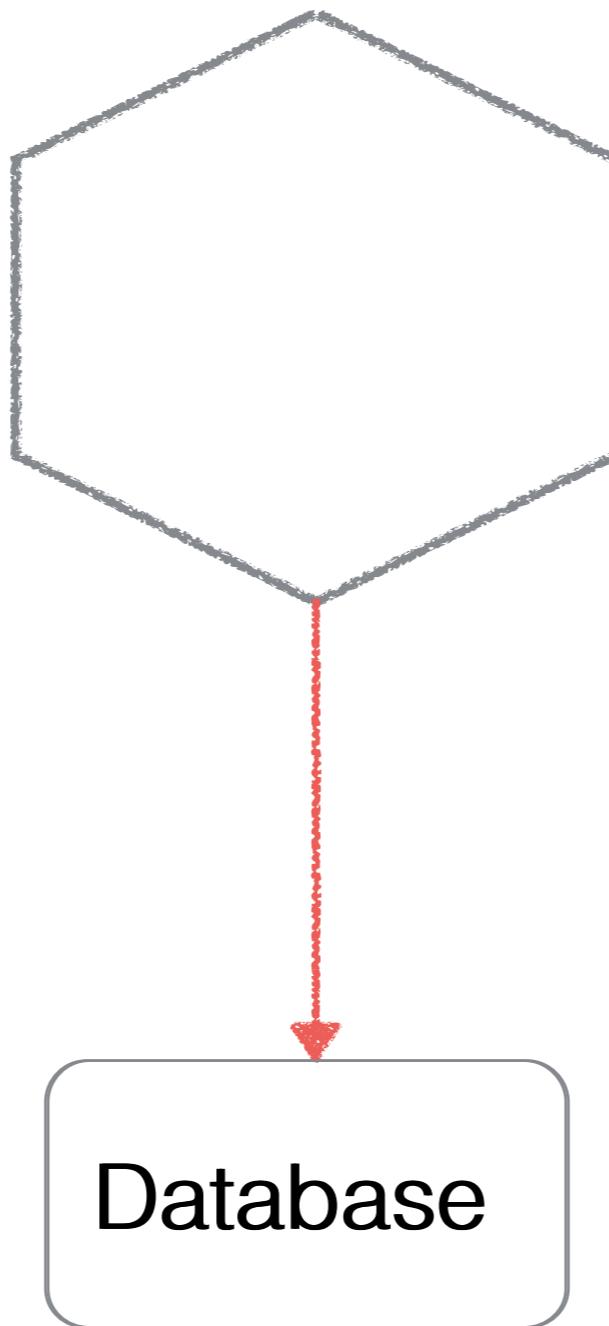
@javazone

@samnewman



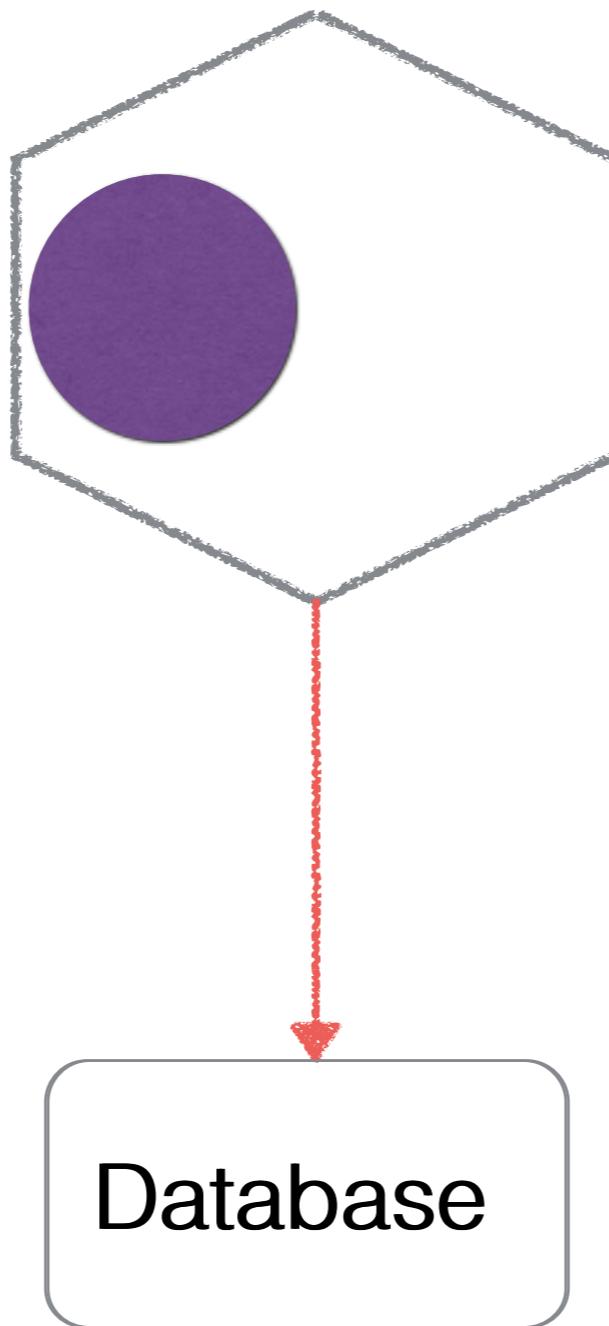
@javazone

@samnewman



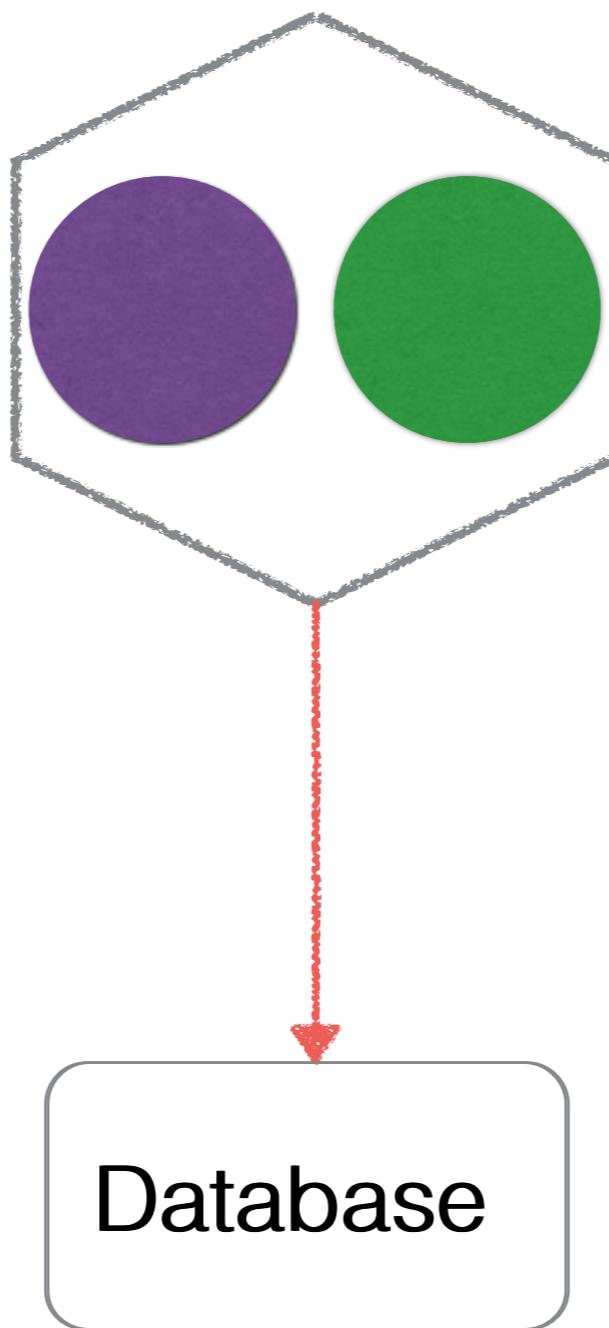
@javazone

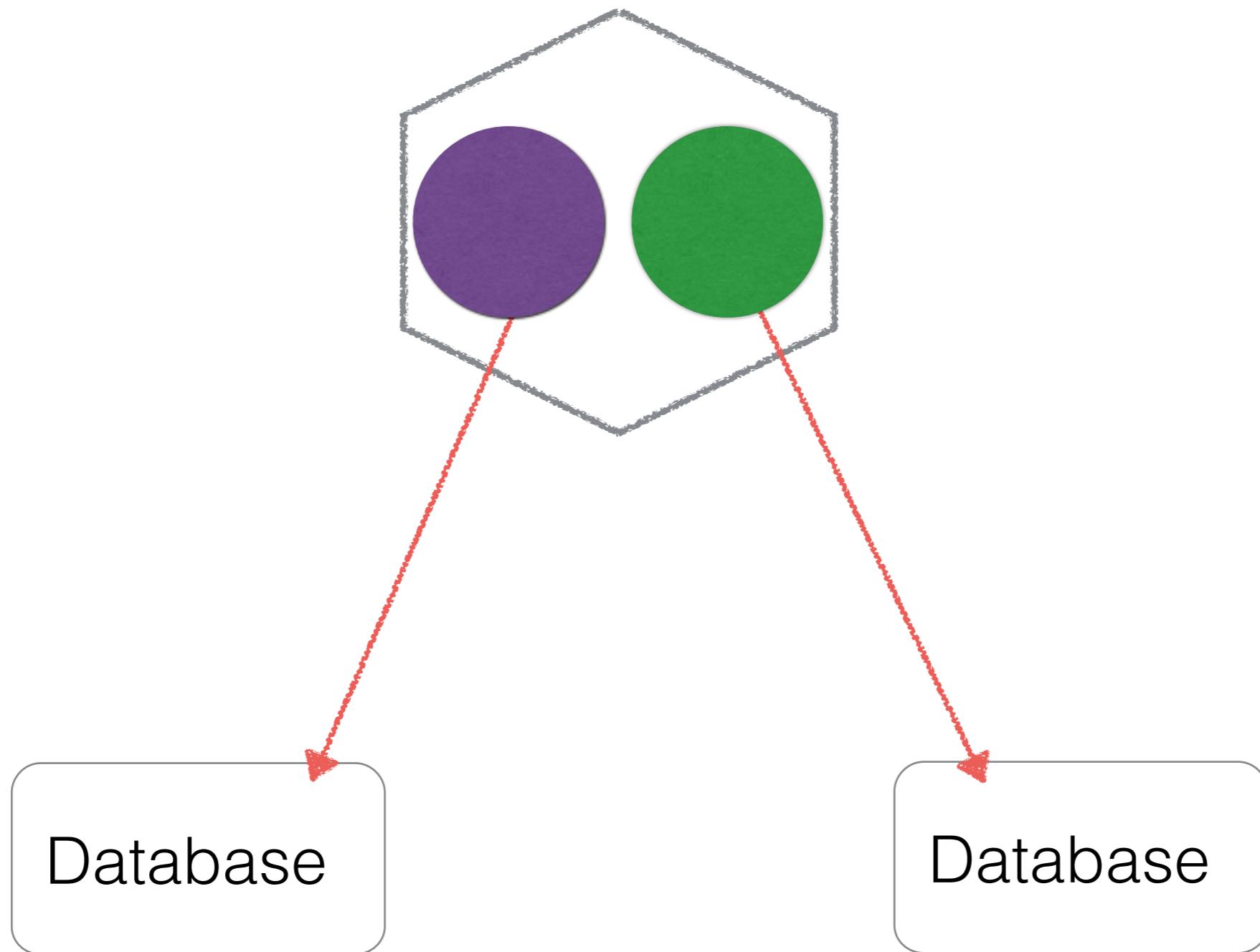
@samnewman



@javazone

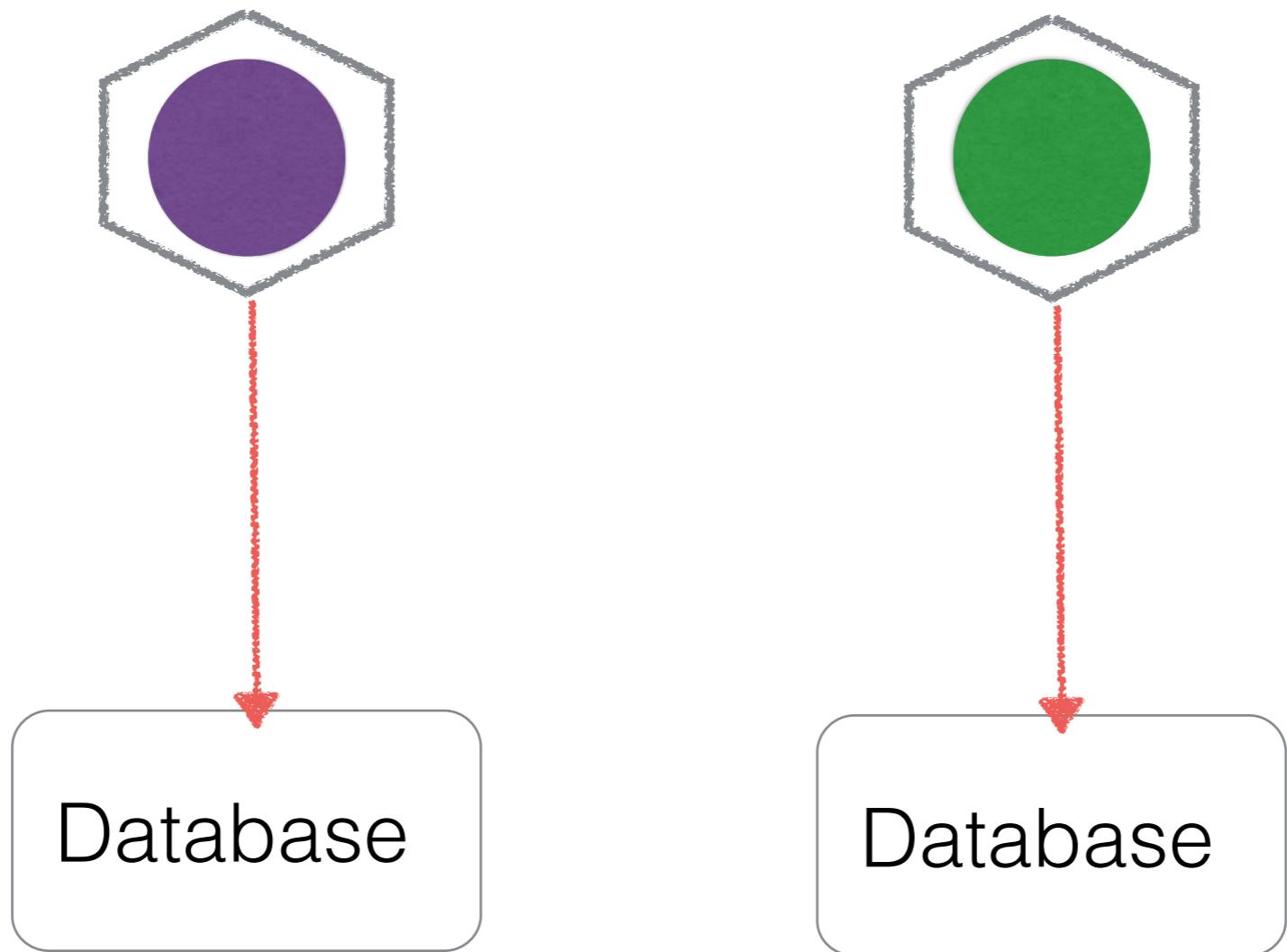
@samnewman





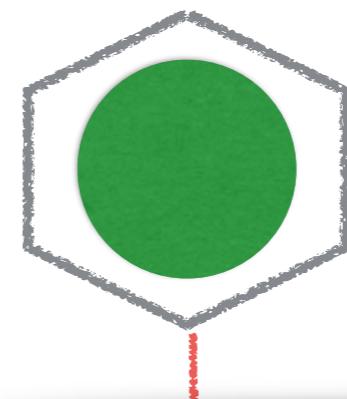
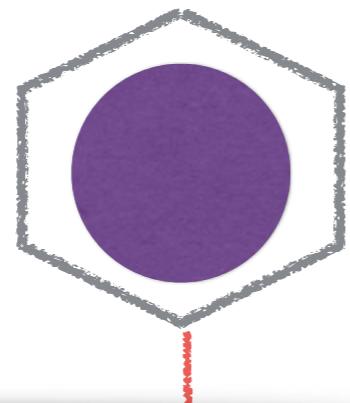
@javazone

@samnewman

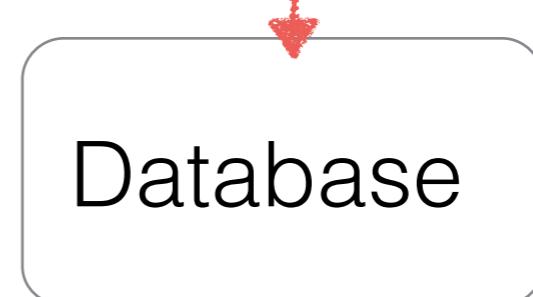
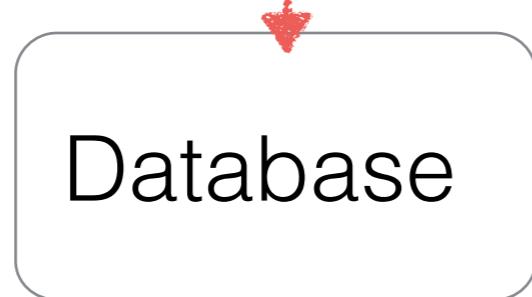


@javazone

@samnewman



TIP 4 / 14: Separate the databases before separating services



The Customer Is Always Right

@javazone

@samnewman

Explorer

Custom Request Headers

Properties

```
{  
  "welcome": "Welcome to a halTalk server.",  
  "hint_1": "You need an account to post stuff..",  
  "hint_2": "Create one by POSTing via the ht:signup link..",  
  "hint_3": "Click the orange buttons on the right to make POST requests..",  
  "hint_4": "Click the green button to follow a link with a GET request..",  
  "hint_5": "Click the book icon to read docs for the link relation."  
}
```

Explorer

/

Custom Request | Swagger™



Properties

```
{  
  "welcome": "Welcome to a halting point.  
  \"hint_1\": \"You need an account  
  \"hint_2\": \"Create one by POSTing to /user/  
  \"hint_3\": \"Click the orange button  
  \"hint_4\": \"Click the green button  
  \"hint_5\": \"Click the book icon  
}
```

Document your API with style

Swagger™ is a specification and complete framework implementation for describing, producing, consuming, and visualizing RESTful web services. The overarching goal of Swagger is to enable client and documentation systems to update at the same pace as the server. The documentation of methods, parameters, and models are tightly integrated into the server code, allowing APIs to always stay in sync. With Swagger, deploying, managing, and using powerful APIs has never been easier.

Who is responsible for Swagger™?

Both the specification and framework implementation are initiatives from Wordnik. Swagger was developed for Wordnik's own use during the development of developer.wordnik.com and the underlying system. Swagger development began in early 2010—the framework being released is currently used by Wordnik's APIs, which power both internal and external API clients.

Why is Swagger™ useful?

The Swagger framework simultaneously solves server, client, and documentation/sandbox needs. As a specification, it is language-agnostic. It also

The screenshot shows the Swagger API Explorer interface. At the top, there's a search bar with 'my-awesome-api.com'. Below it, there are two sections: 'USER' and 'word'. The 'USER' section has a single entry: 'GET /user'. The 'word' section has several entries:

Method	Path	Description
GET	/word.json/{word}/entries	Return entries for a word
GET	/word.json/{word}/examples	Returns examples for a word
POST	/word.json/{word}/examples	Fetches examples for a word
POST	/word.json/{word}/wordForms	Adds a Relationship Map to a word
GET	/word.json/{word}/wordForms	Returns other forms of a word
DELETE	/word.json/{word}/wordForms	Deletes a relationship from a word
GET	/word.json/{word}	Given a word as a string, returns the WordObject that represents it

Below the table, there's a 'Parameters' section with a table:

Parameter	Value	Description
word	(required)	String value of WordObject to return

Explorer

/

Custom Request | Swagger™



Properties

```
{  
  "welcome": "Welcome to the Wordnik API.",  
  "hint_1": "You need to log in to use this API.",  
  "hint_2": "Create one by POSTing to /register.",  
  "hint_3": "Click the orange button to log in.",  
  "hint_4": "Click the green button to log out.",  
  "hint_5": "Click the book icon to view your profile."}
```

Document your API with style

Swagger™ is a specification and complete framework implementation for describing, producing, consuming, and visualizing RESTful web services. The overarching goal of Swagger is to enable client and documentation systems to update at the same pace as the server. The documentation of methods, parameters, and models are tightly integrated into the server code, allowing APIs to always stay in sync. With Swagger, deploying, managing, and using powerful APIs has never been easier.

TIP 5 / 14: Adopt a consumer first approach

Swagger was developed for Wordnik's own use during the development of developer.wordnik.com and the underlying system. Swagger development began in early 2010—the framework being released is currently used by Wordnik's APIs, which power both internal and external API clients.

Why is Swagger™ useful?

The Swagger framework simultaneously solves server, client, and documentation/sandbox needs. As a specification, it is language-agnostic. It also

word		Show/Hide	List Operations	Expand Operations	Raw
GET	/word.json/{word}/entries				Return entries for a word
GET	/word.json/{word}/examples				Returns examples for a word
POST	/word.json/{word}/examples				Fetches examples for a word
POST	/word.json/{word}/wordForms				Adds a Relationship Map to a word
GET	/word.json/{word}/wordForms				Returns other forms of a word
DELETE	/word.json/{word}/wordForms				Deletes a relationship from a word
GET	/word.json/{word}				Given a word as a string, returns the WordObject that represents it

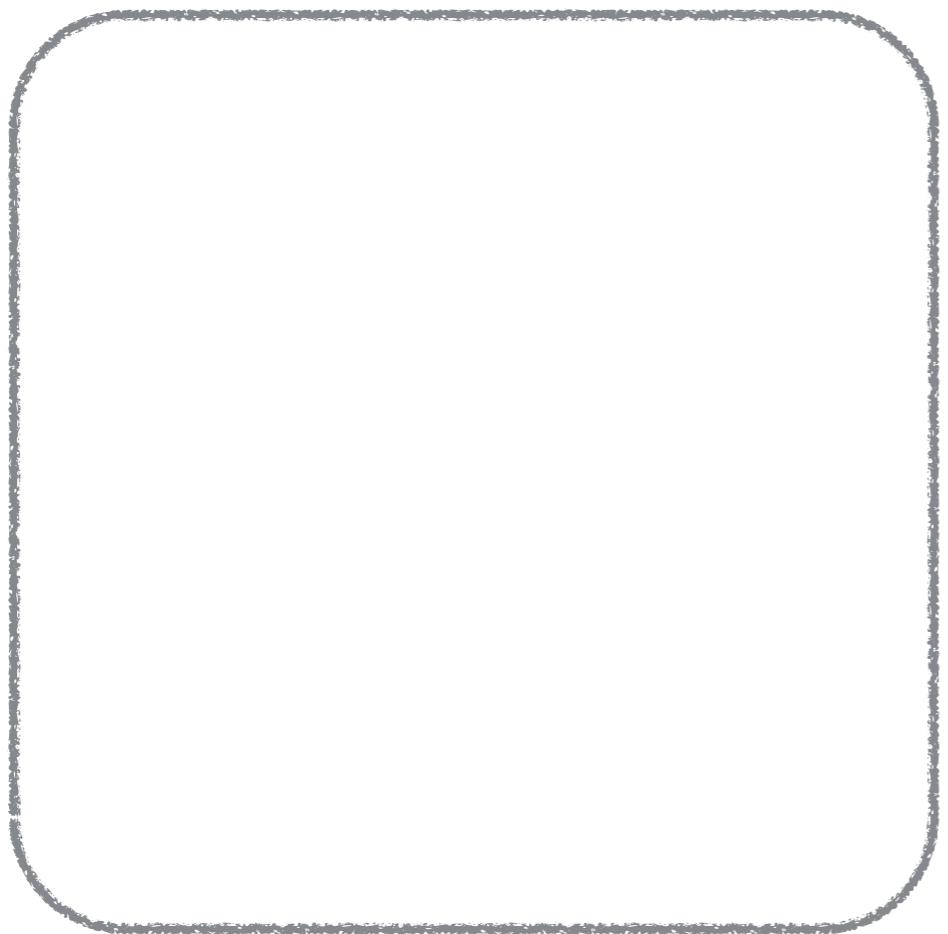
Parameters

Parameter	Value	Description
word	(required)	String value of WordObject to return

Monitoring

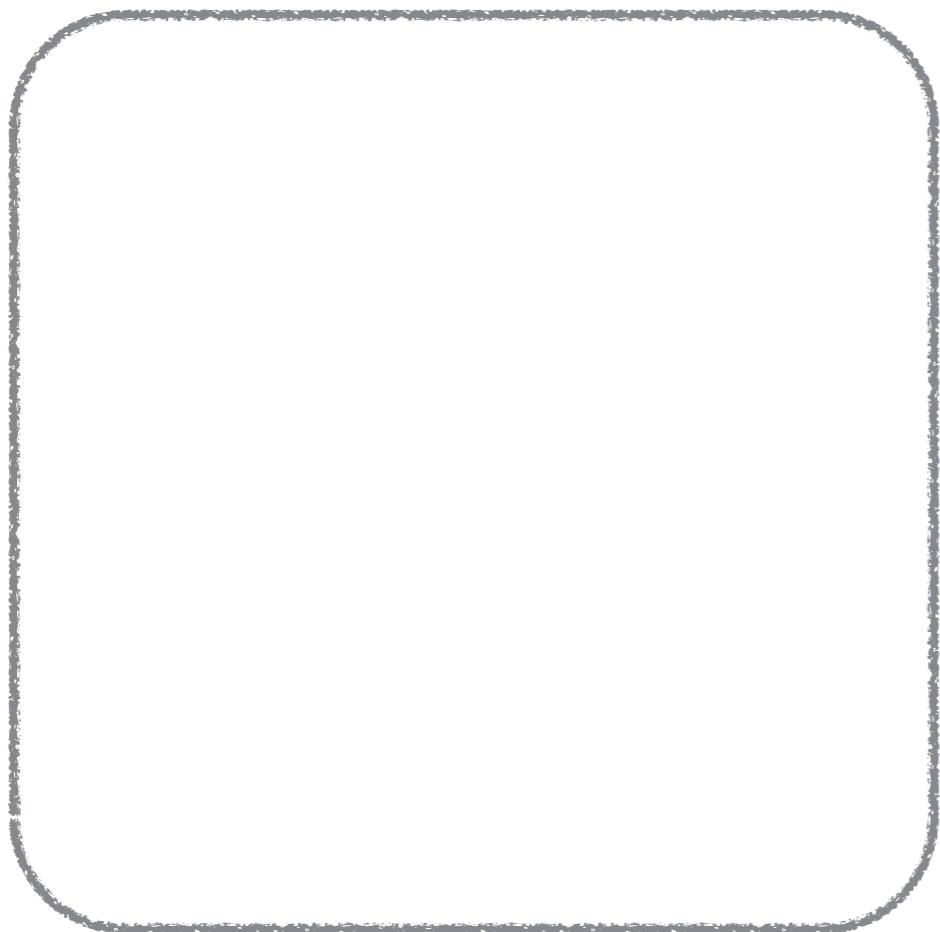
@javazone

@samnewman



@javazone

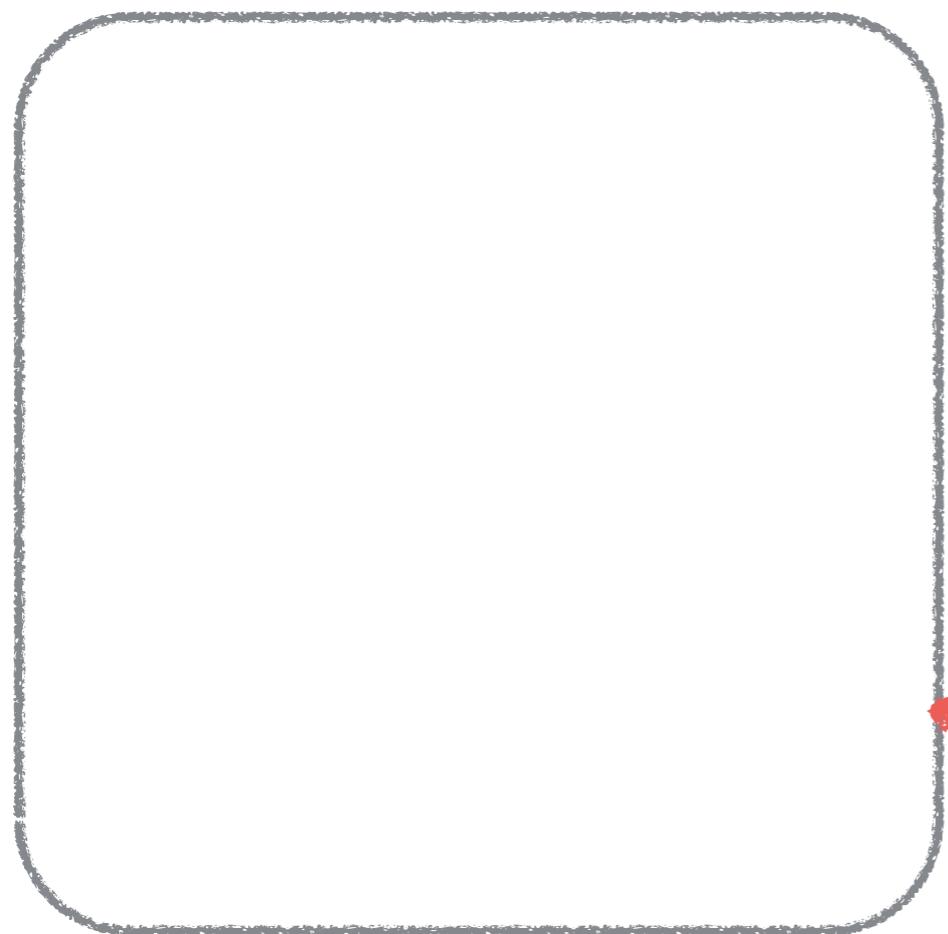
@samnewman



<http://www.flickr.com/photos/kalexanderson/5421517469/>

@javazone

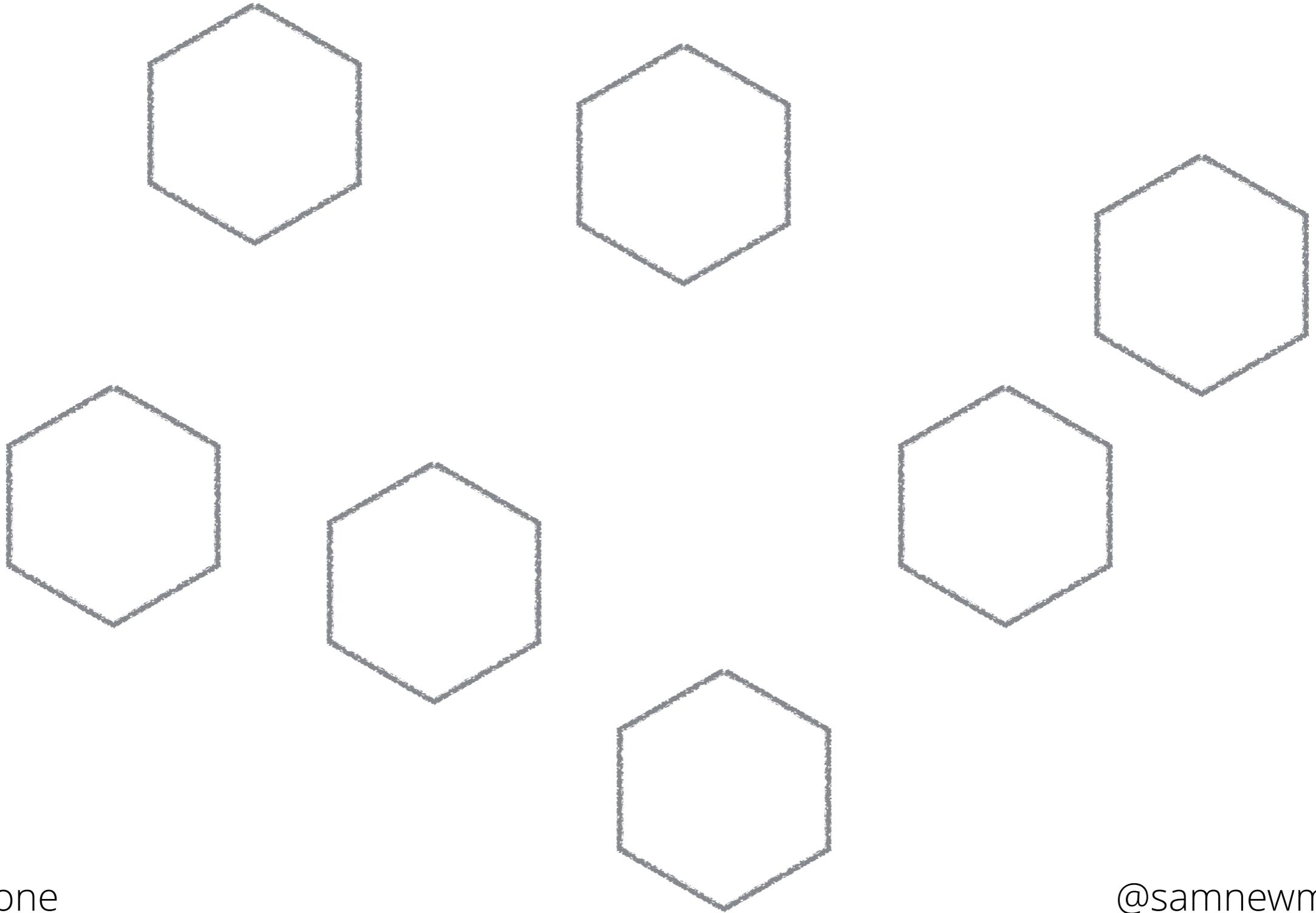
@samnewman



<http://www.flickr.com/photos/kalexanderson/5421517469/>

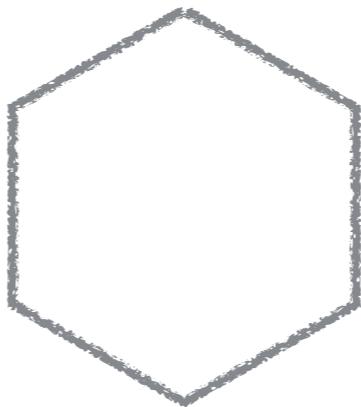
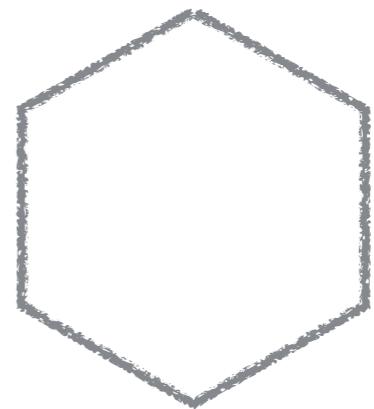
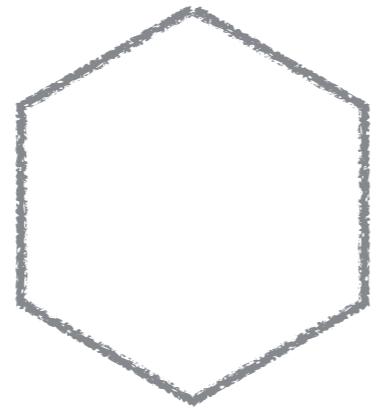
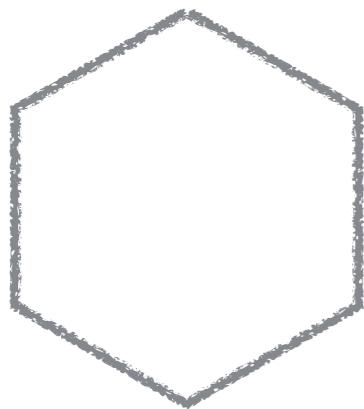
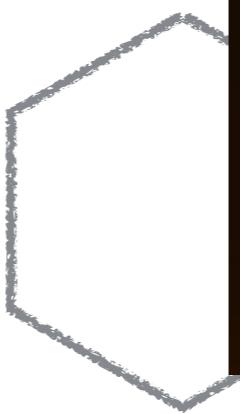
@javazone

@samnewman



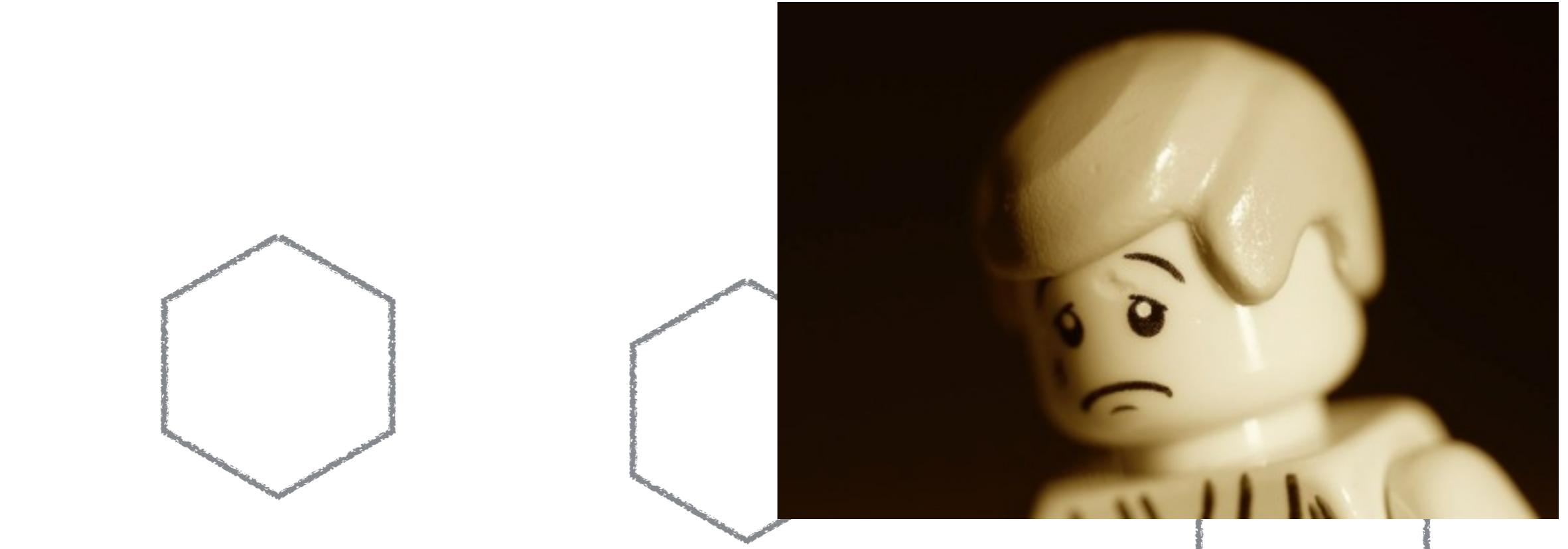
@javazone

@samnewman

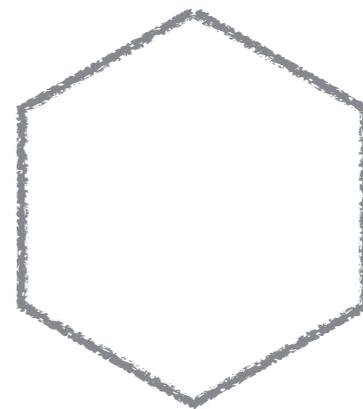
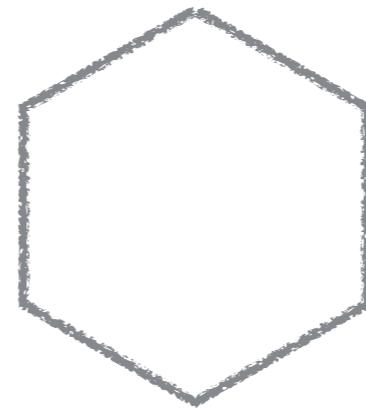
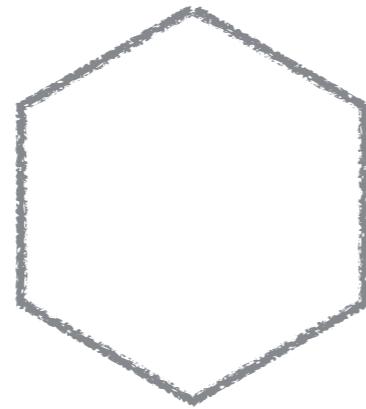
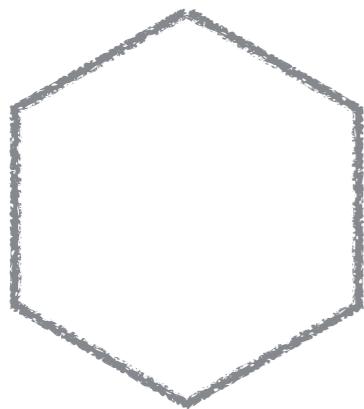


@javazone

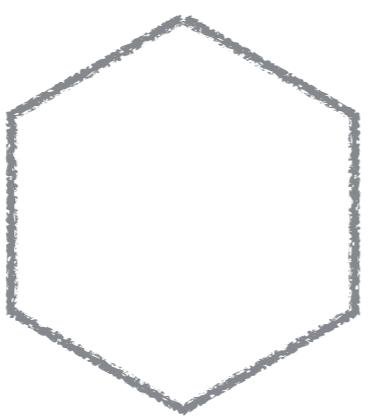
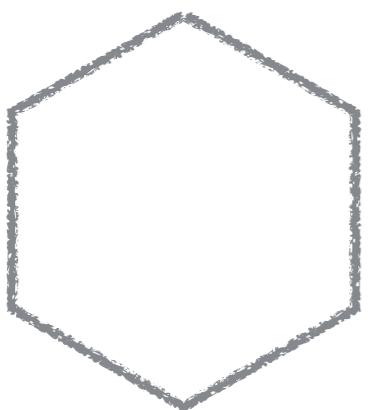
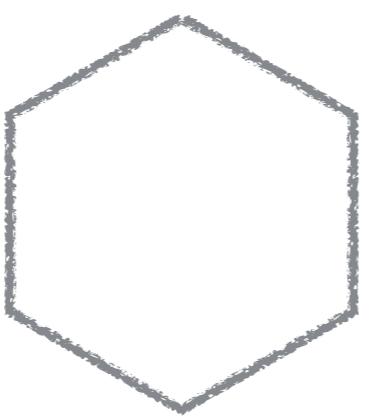
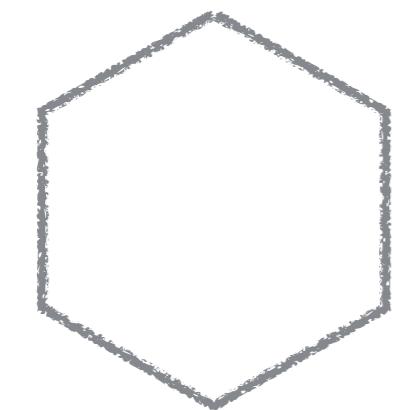
@samnewman



???

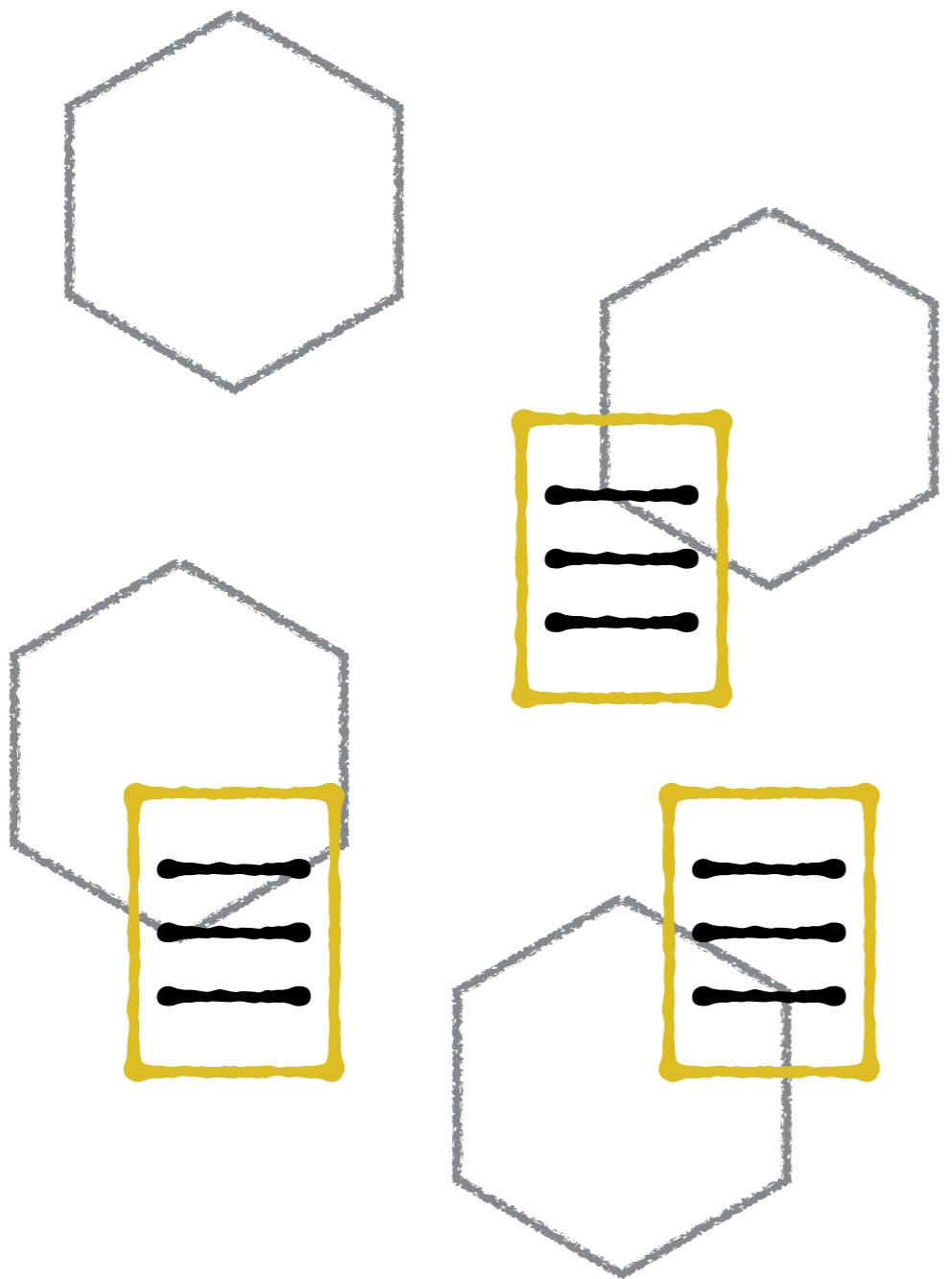


You have to get ***much***
better at monitoring



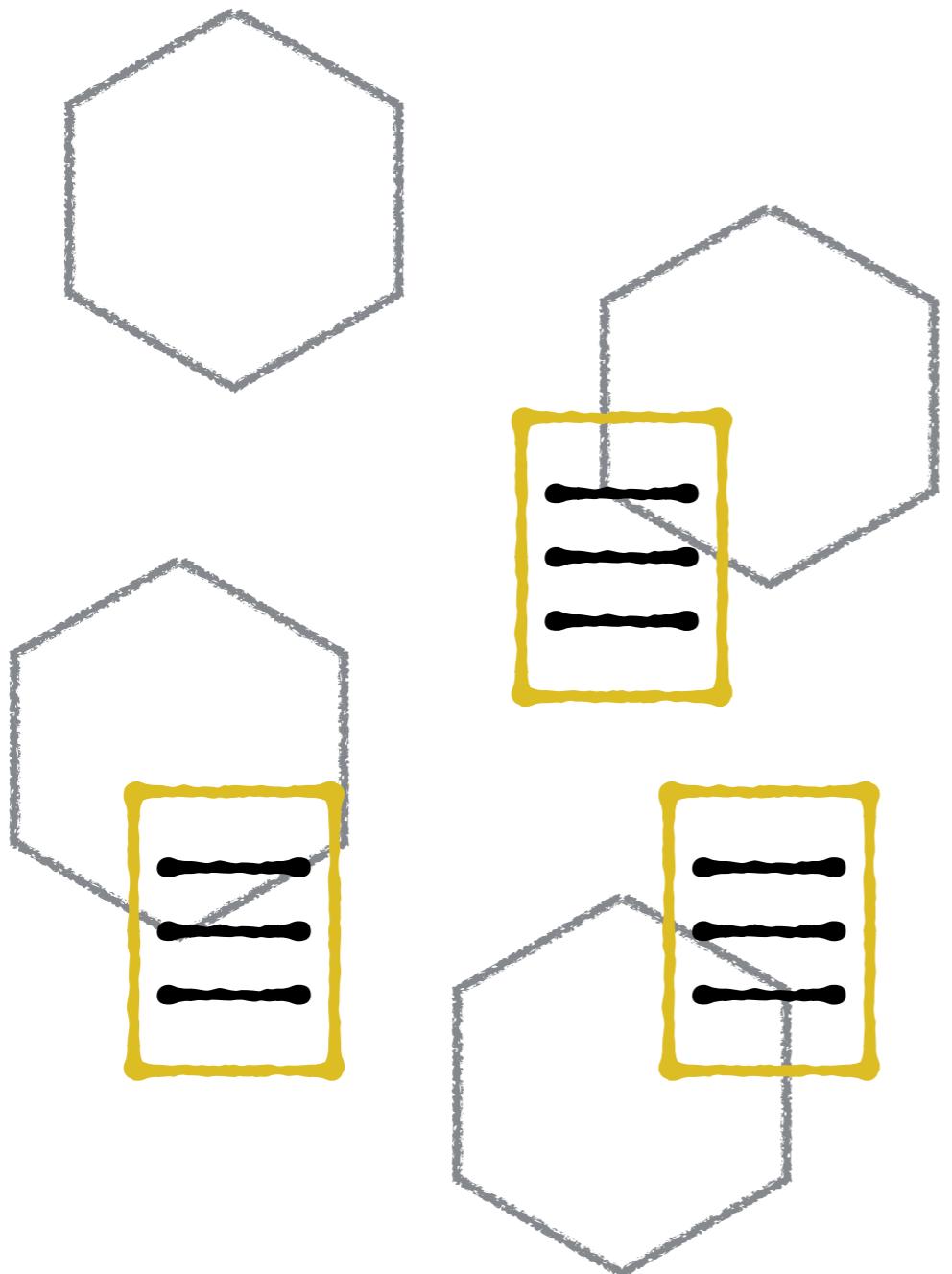
@javazone

@samnewman



@javazone

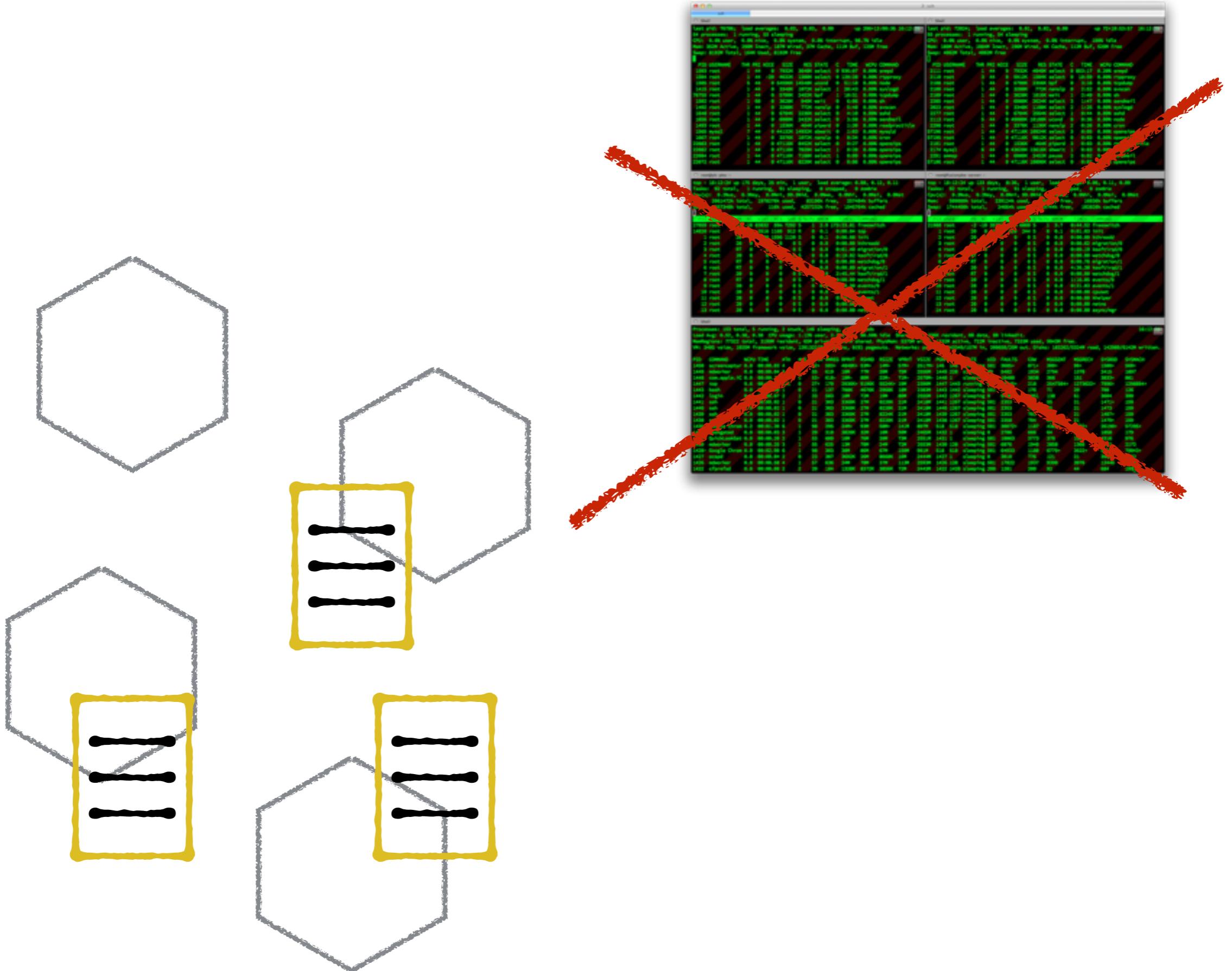
@samnewman



@javazone

A screenshot of a terminal window showing multiple processes running on a Linux system. The processes are listed in a table-like format with columns for PID, PPID, TTY, and various system metrics like CPU usage and memory usage. The terminal has a red and black striped background.

@samnewman



@javazone

@samnewman

You are not a badass if you
use an SSH Multiplexer



logstash

logstash is a tool for managing events and log data. It can collect, parse, transform, and store them for later use (like, for search engines). It also includes a web interface for searching and drilling down.

It is fully free and fully open source. The license is MIT, so you are much free to use it however you want in whatever way you like.

[Download](#)

[Cookbook](#)



logstash

logstash is a tool for managing events and log data. It can collect, transform, and store them for later use (like, for searching). It comes with a web interface for searching and drilling down into your data.

It is fully free and fully open source. The license is MIT, so you are much free to use it however you want in whatever way you like.

Kibana

Make sense of a mountain of logs Now in Ruby!

[Get Started »](#)

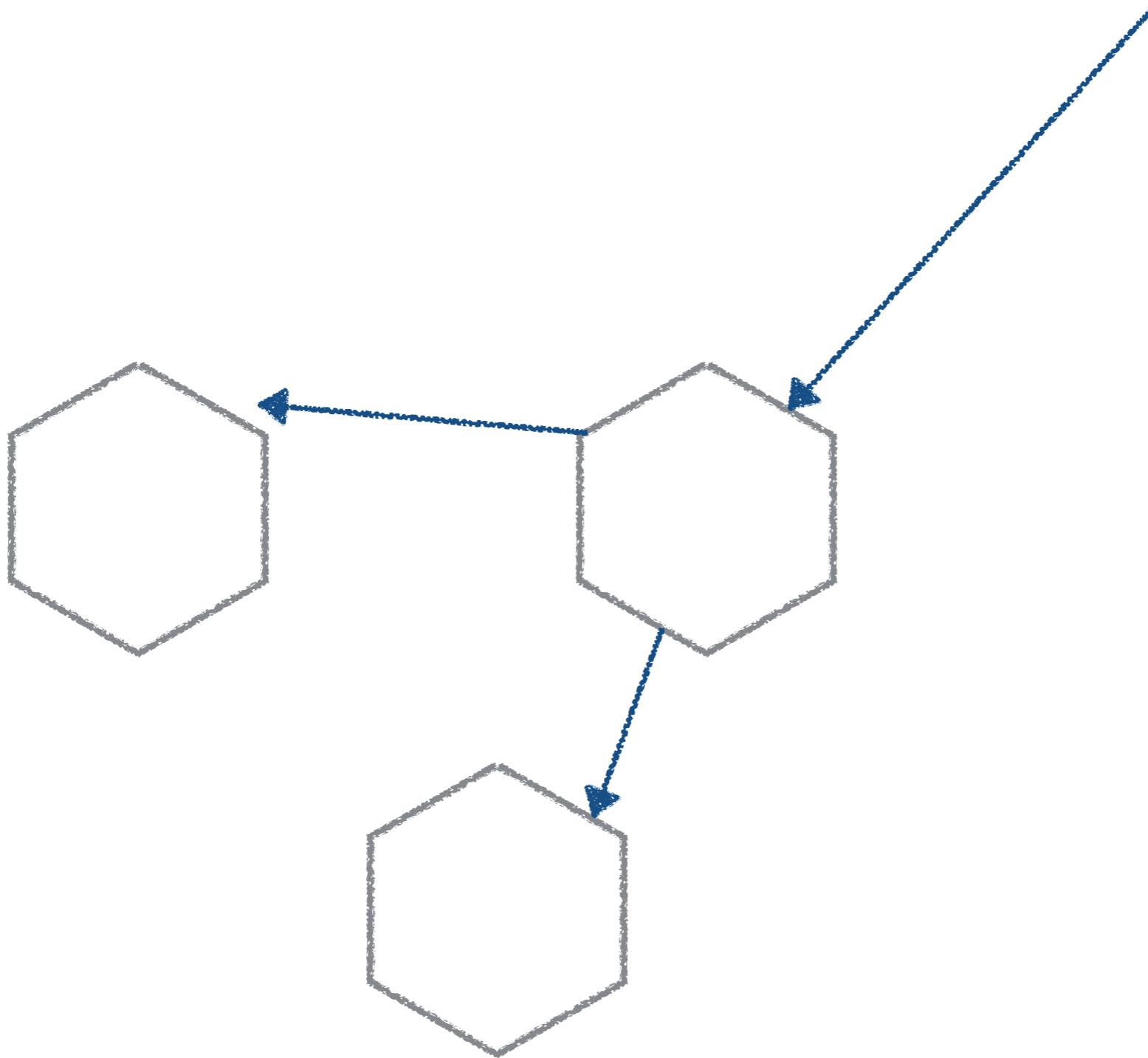
[GitHub project](#) [Logstash](#) [ElasticSearch](#)

Star 1,198 Fork 287

Every event under one roof

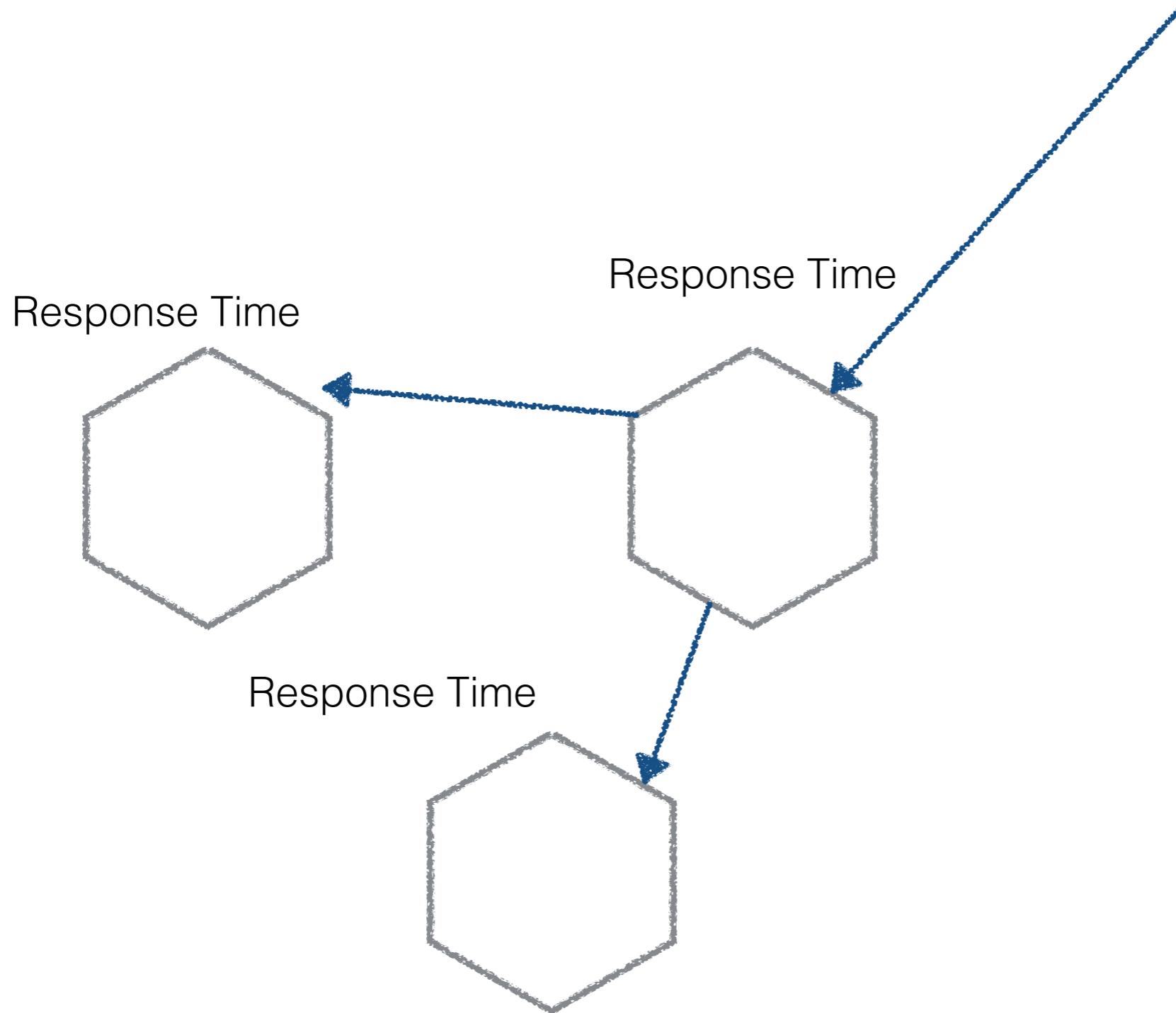
@javazone

@samnewman



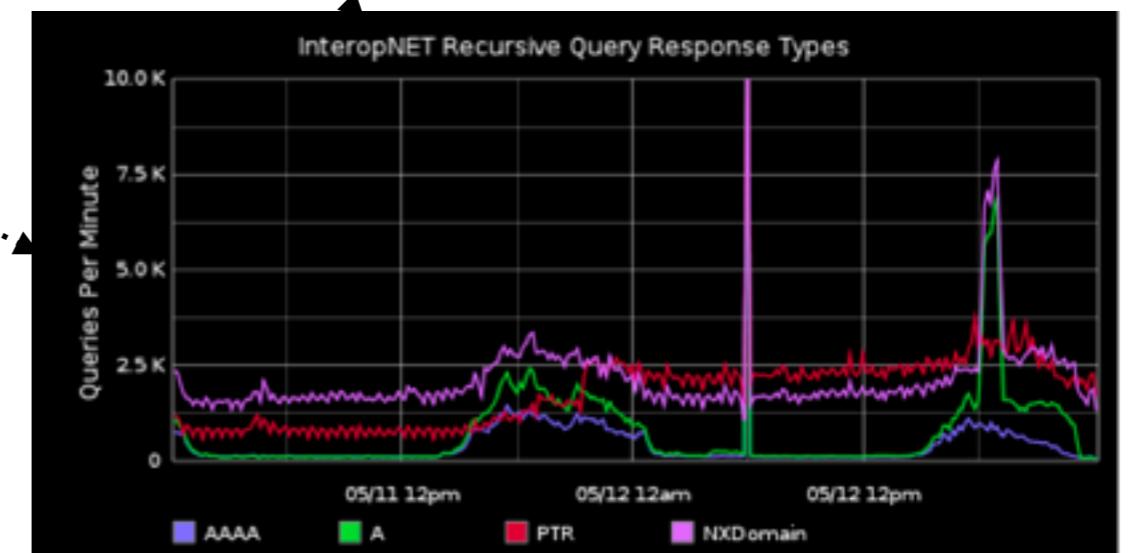
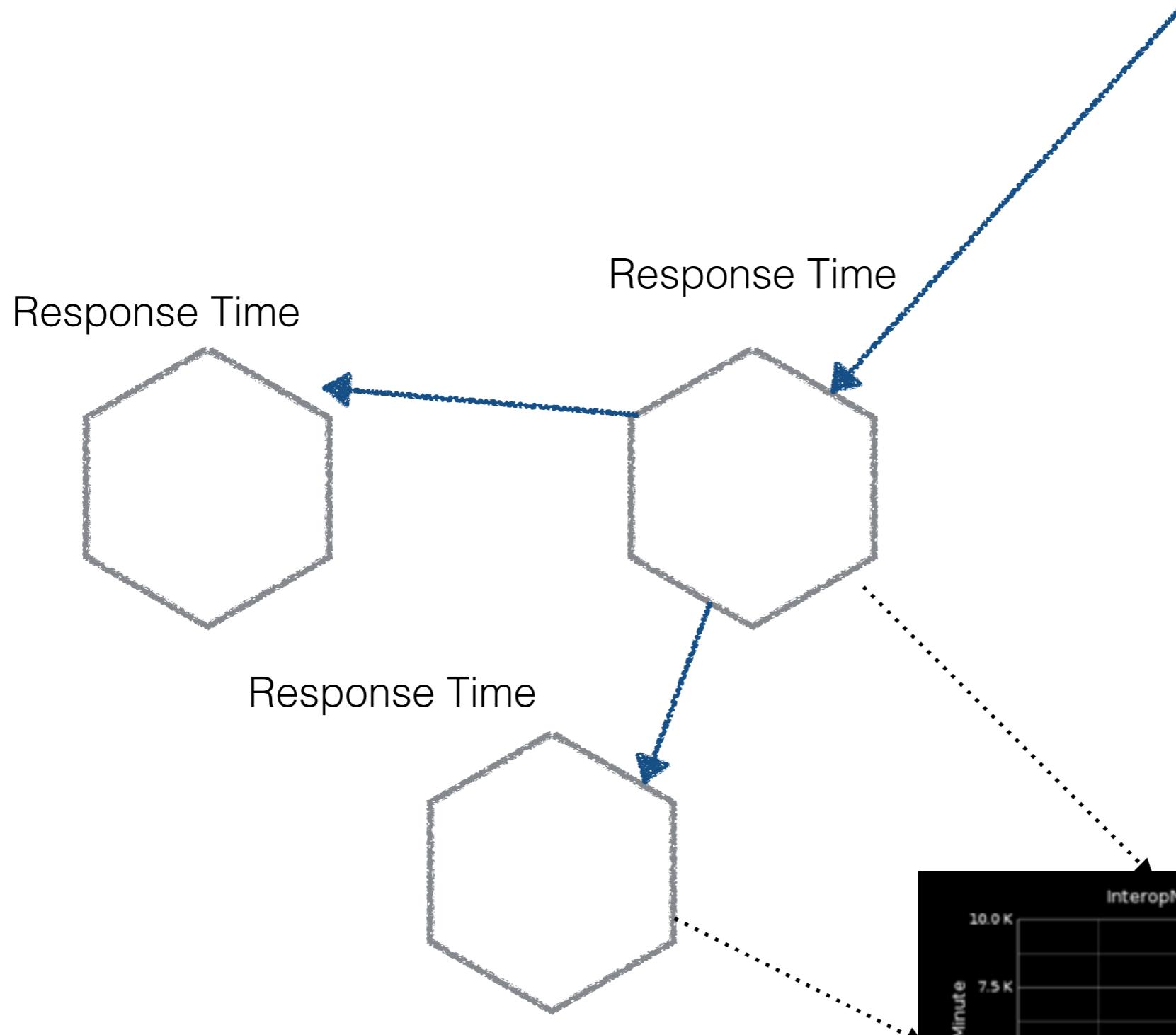
@javazone

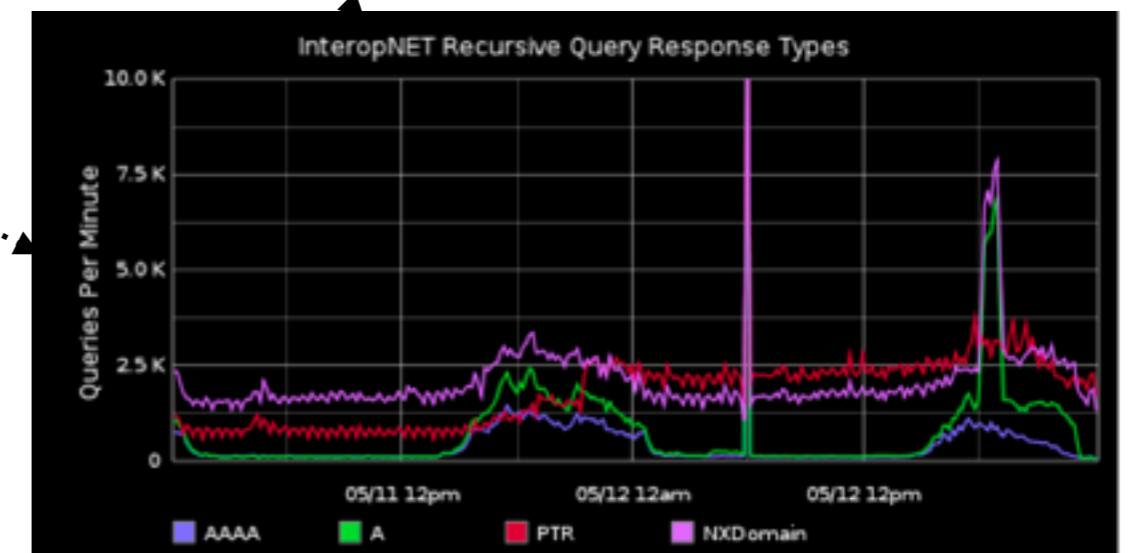
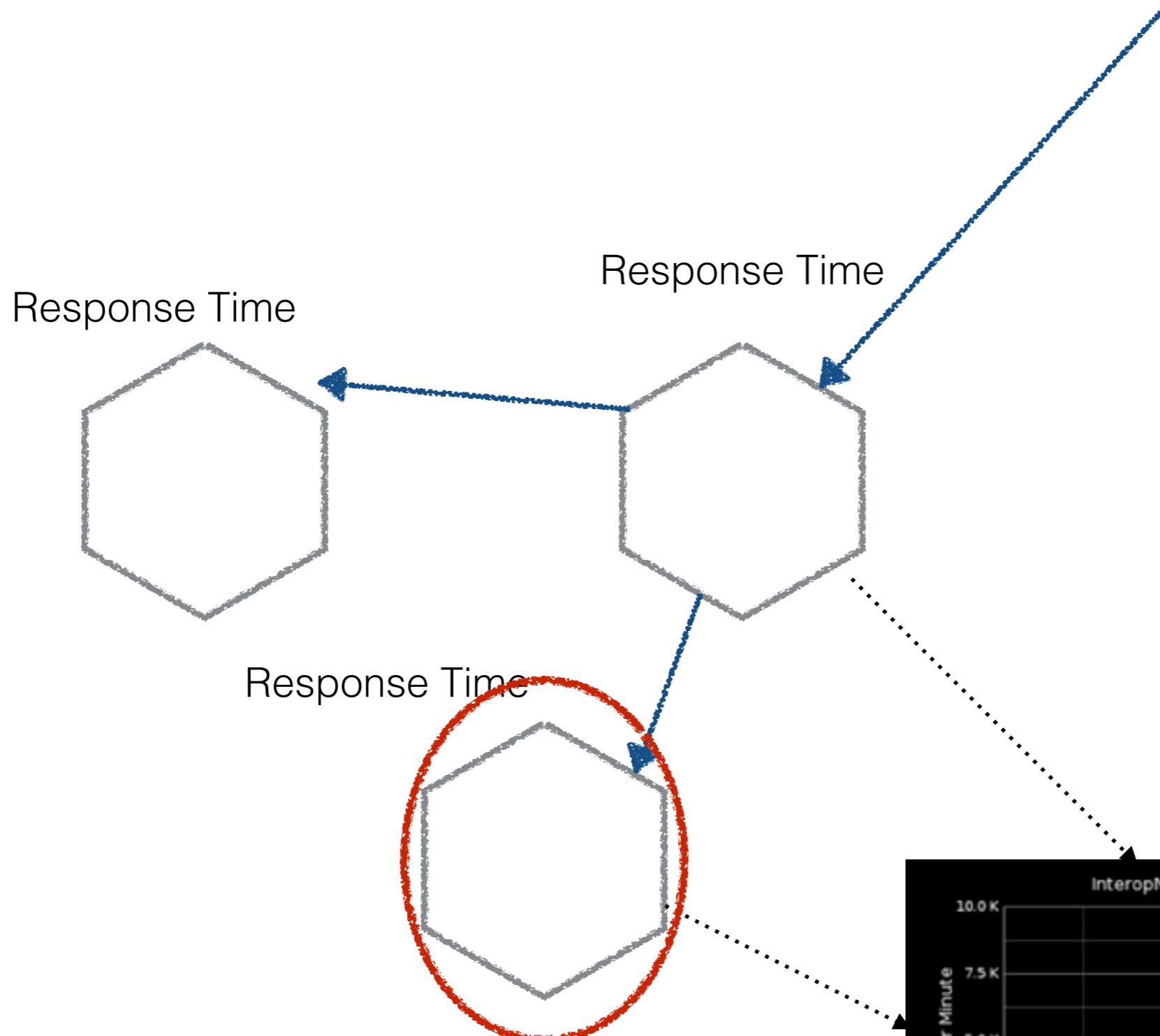
@samnewman

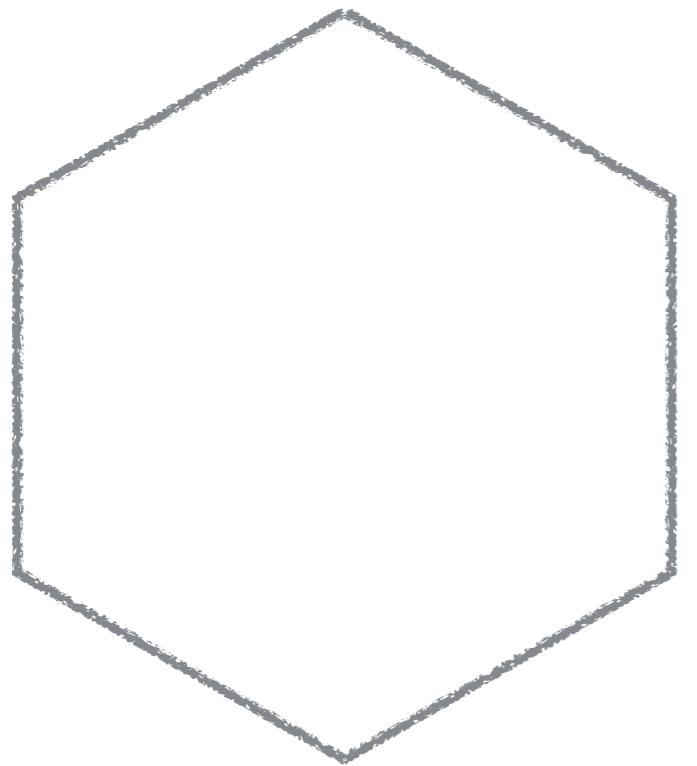


@javazone

@samnewman

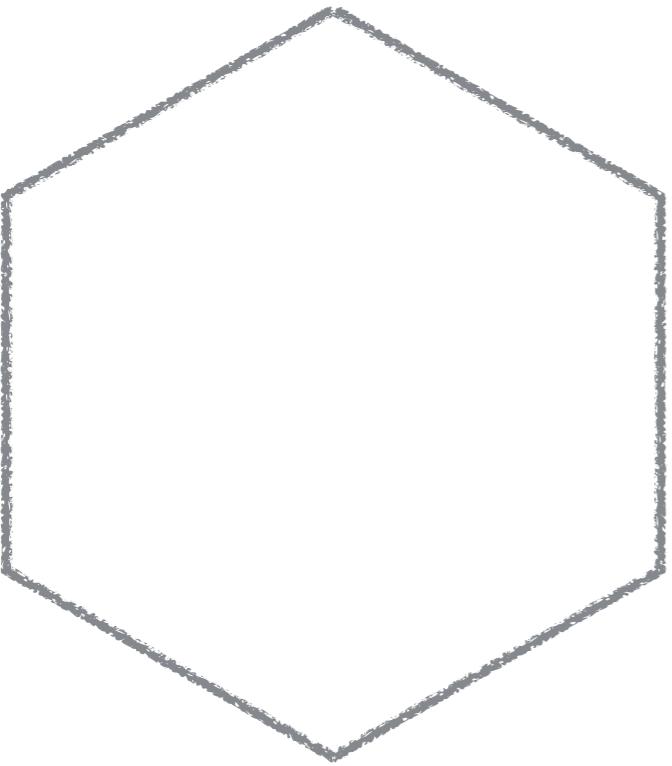




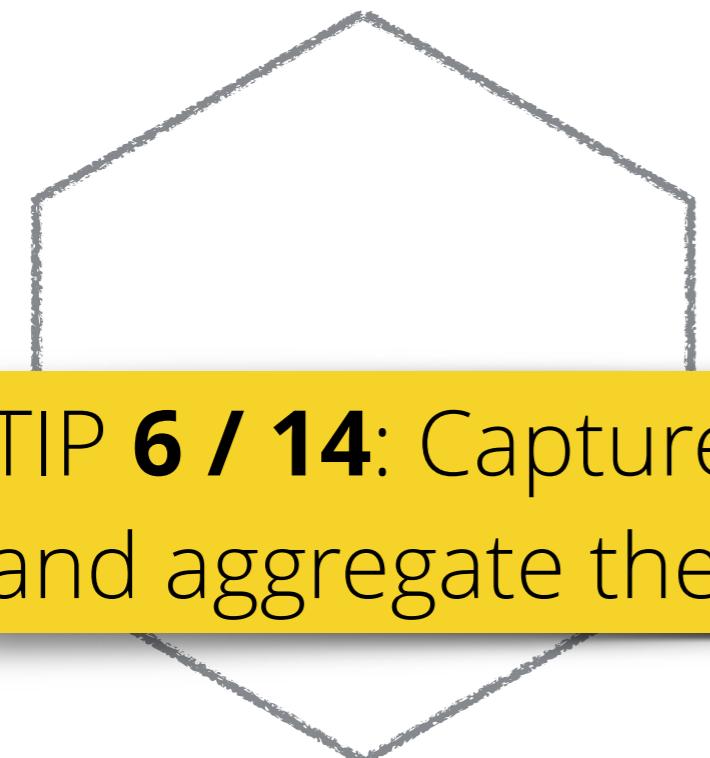


@javazone

@samnewman

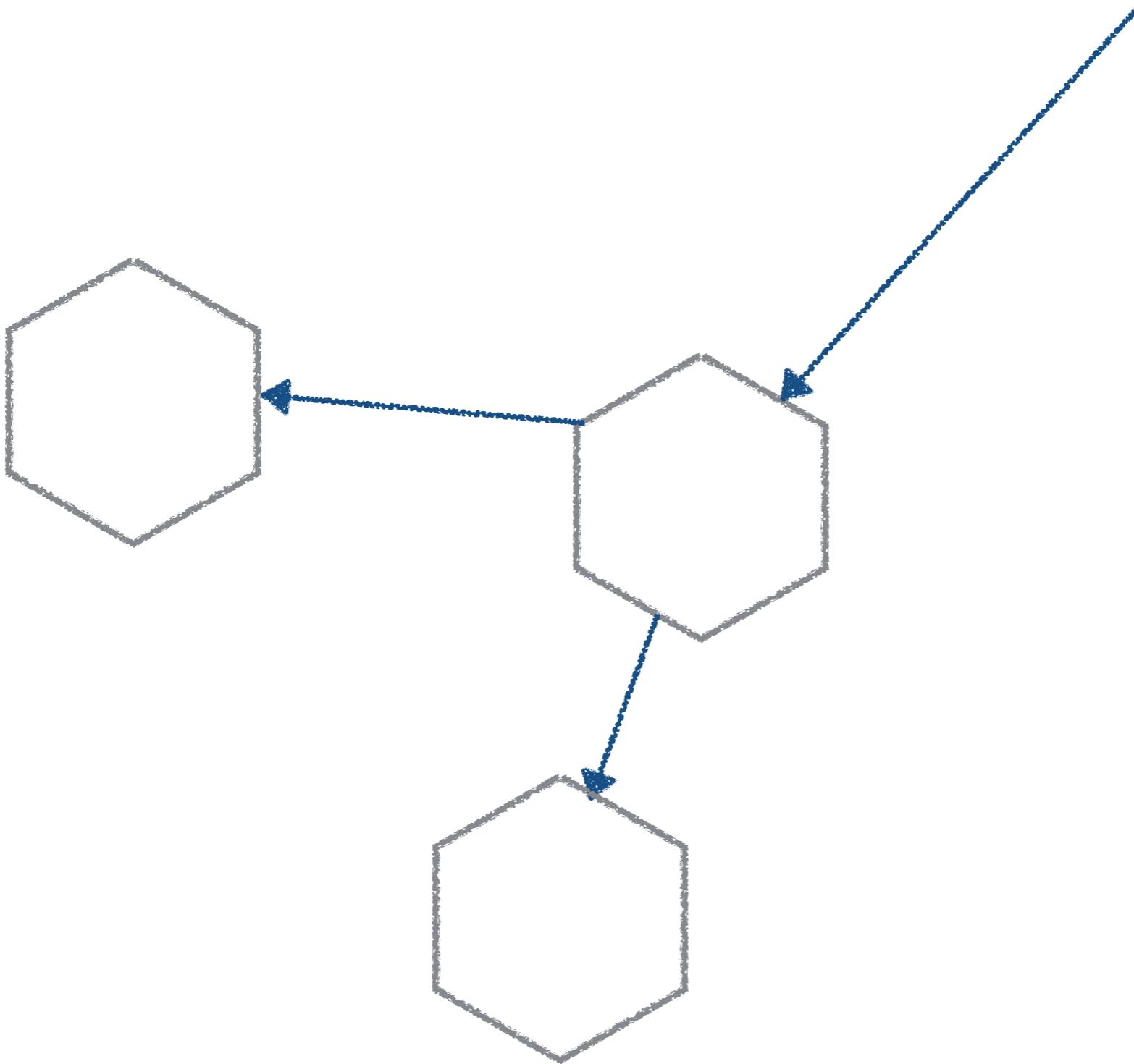


numberOfApplicationErrors	57
numberOfServicedRequestsWithResponse200	136711
numberOfServicedRequestsWithResponse304	27782
numberOfServicedRequestsWithResponse404	303
numberOfServicedRequestsWithResponse500	141
totalNumberOfServicedRequests	172383



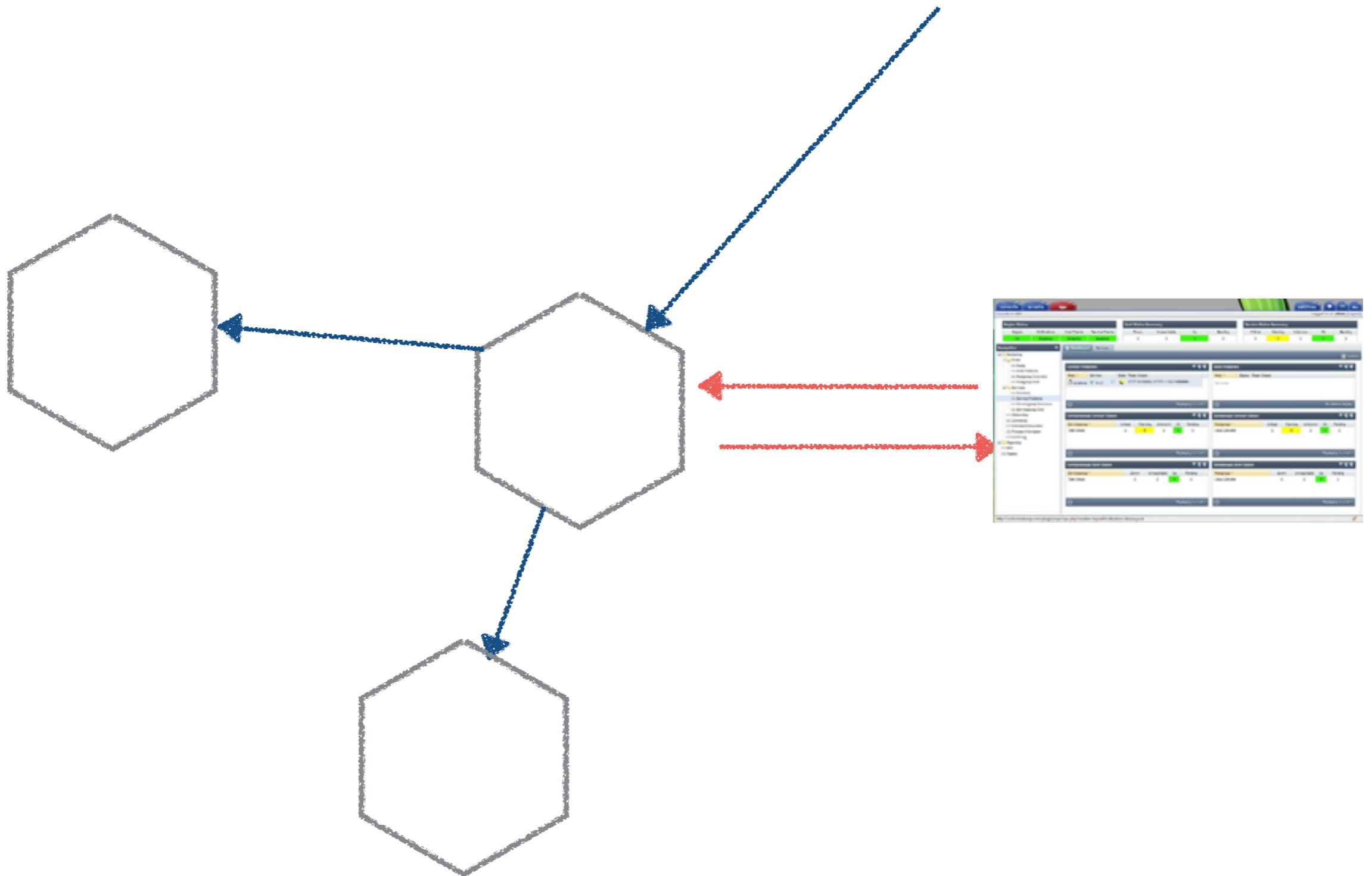
TIP 6 / 14: Capture metrics, and logs, for each node, and aggregate them to get a rolled up picture

numberOfApplicationErrors	57
numberOfServicedRequestsWithResponse200	136711
numberOfServicedRequestsWithResponse304	303
numberOfServicedRequestsWithResponse500	141
totalNumberOfServicedRequests	172383



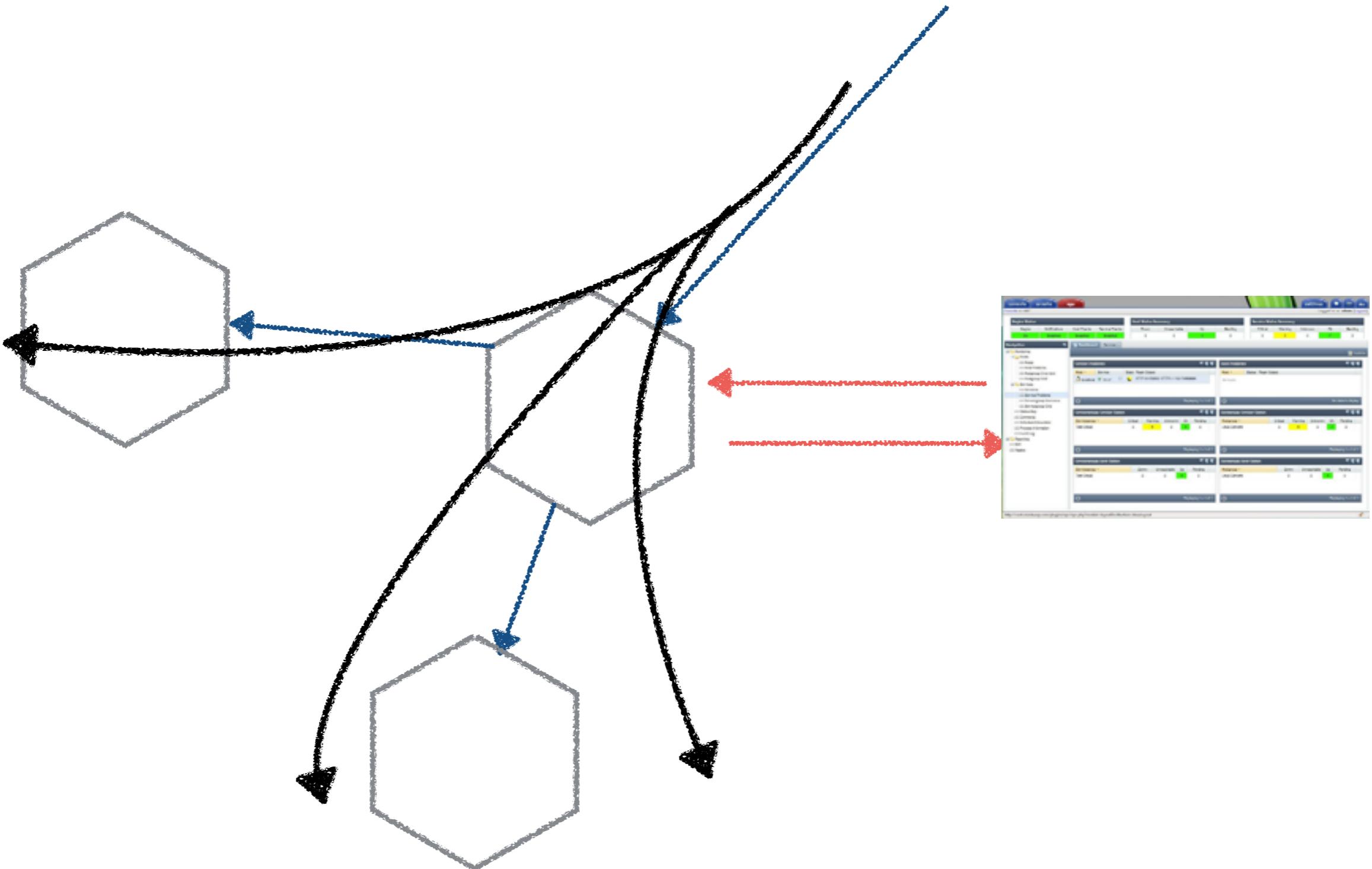
@javazone

@samnewman



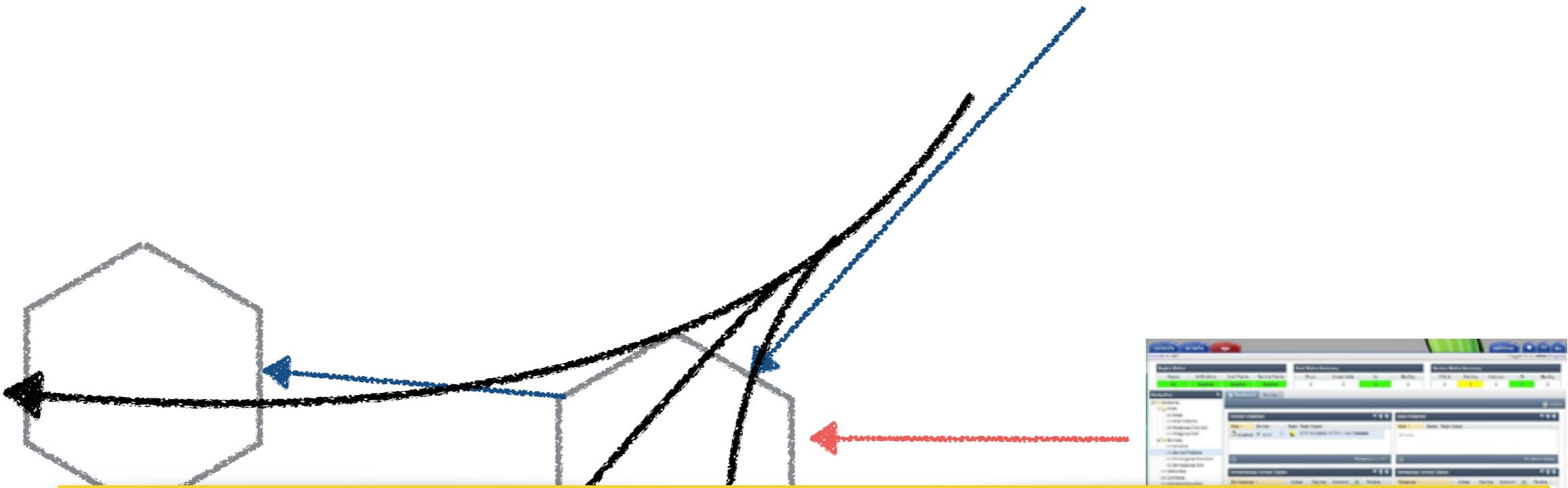
@javazone

@samnewman

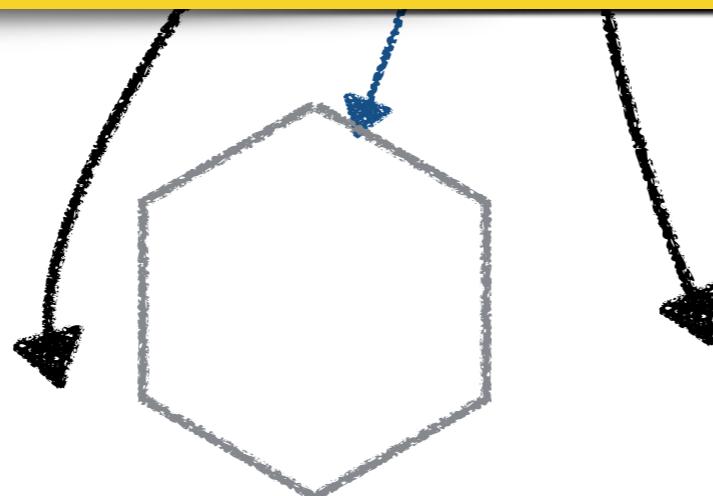


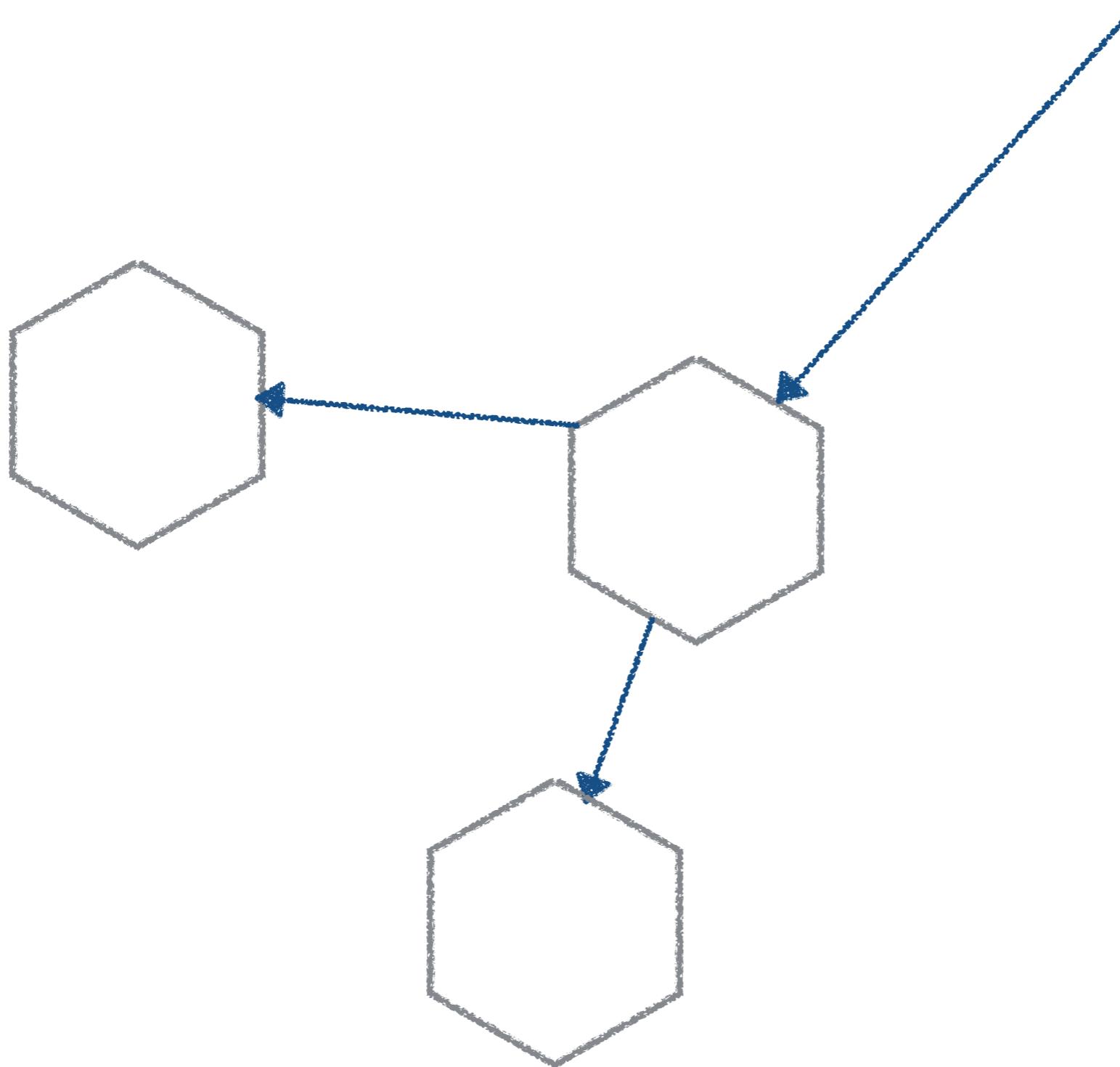
@javazone

@samnewman



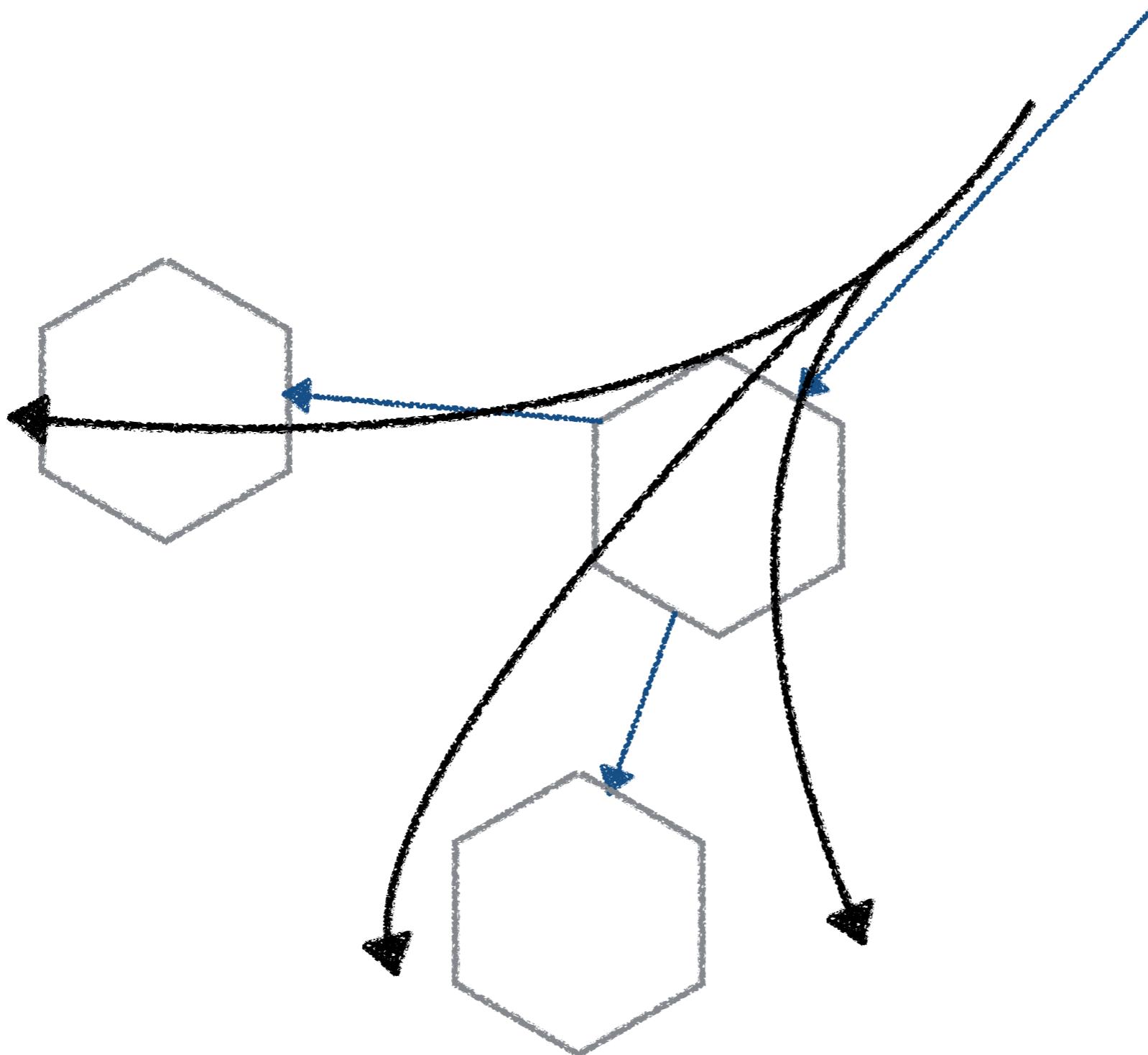
TIP 7 / 14: Use synthetic transactions to test production systems





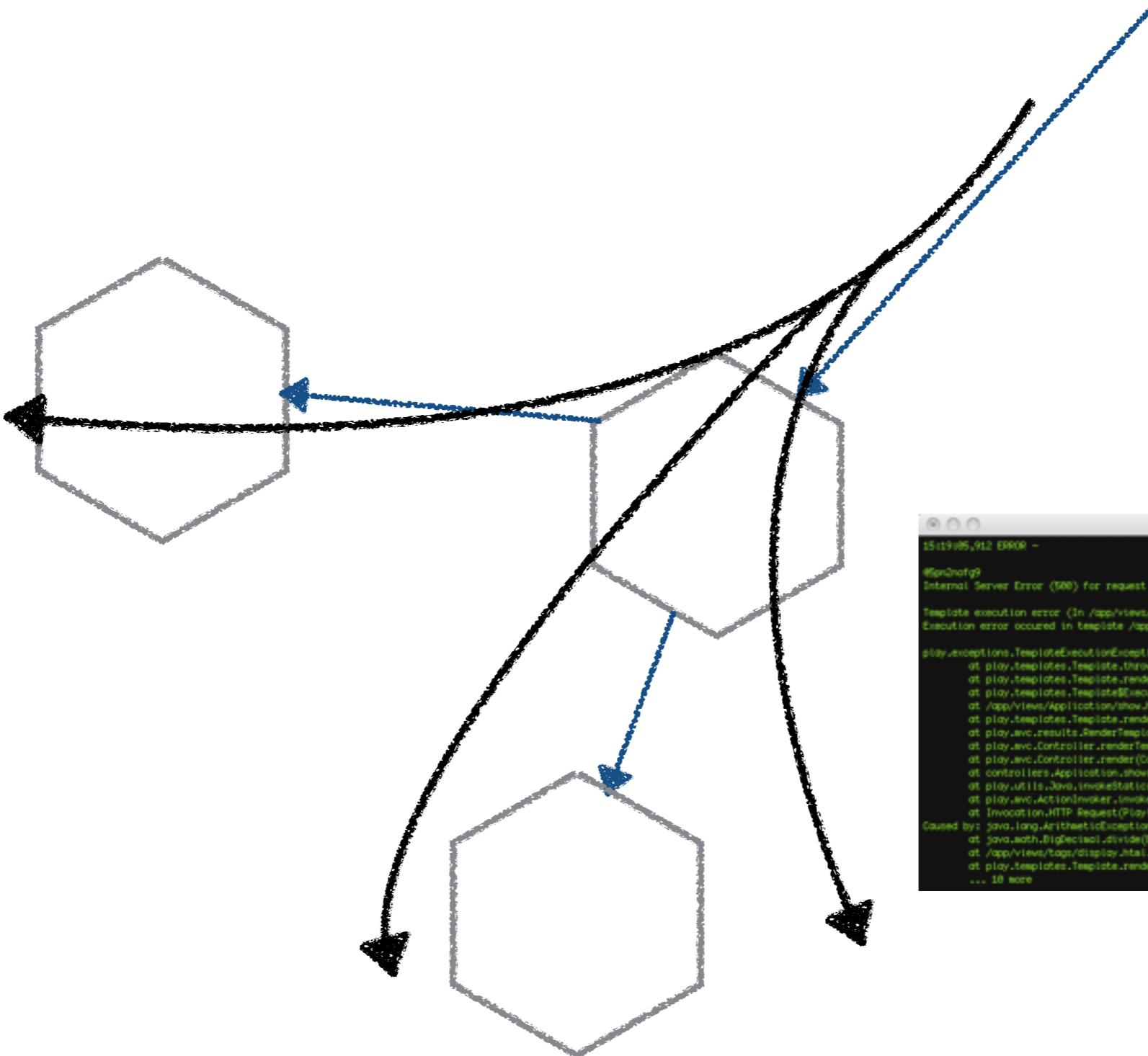
@javazone

@samnewman



@javazone

@samnewman



```

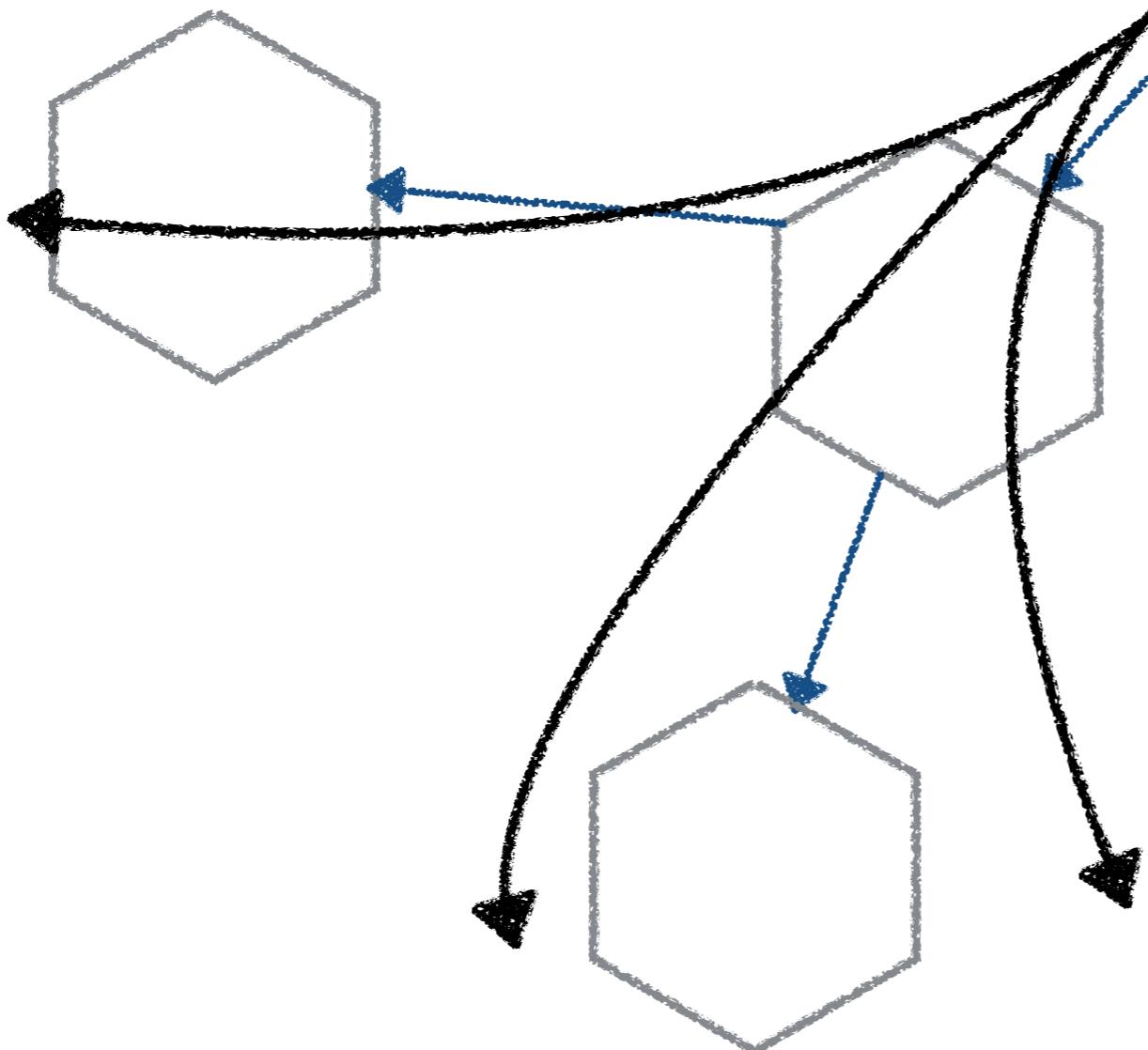
Terminal — java — 130x26
15:19:05,912 ERROR —
@Kon2nfg9
Internal Server Error (500) for request GET /posts/1

Template execution error (In /app/views/tags/display.html around line 18)
Execution error occurred in template /app/views/tags/display.html. Exception raised was ArithmeticException : / by zero.

play.exceptions.TemplateExecutionException: / by zero
    at play.templates.Template.throwException(Template.java:252)
    at play.templates.Template.render(Template.java:227)
    at play.templates.Template$TemplateExecutionTemplate.invokeTop(Template.java:369)
    at /app/views/Application/show.html-(line:21)
    at play.templates.Template.render(Template.java:287)
    at play.mvc.results.RenderTemplate$1.run(RenderTemplate.java:22)
    at play.mvc.Controller.renderTemplate(Controller.java:367)
    at play.mvc.Controller.render(Controller.java:393)
    at controllers.Application.show(Application.java:26)
    at play.utils.Java.invokeStatic(Java.java:129)
    at play.mvc.ActionInvoker.invoke(ActionInvoker.java:24)
    at Invocation.HTTP Request(Play!)
Caused by: java.lang.ArithmaticException: / by zero
    at java.math.BigDecimal.divide(BigDecimal.java:1327)
    at /app/views/tags/display.html-(line:18)
    at play.templates.Template.render(Template.java:287)
    ... 10 more

```

ID: 123



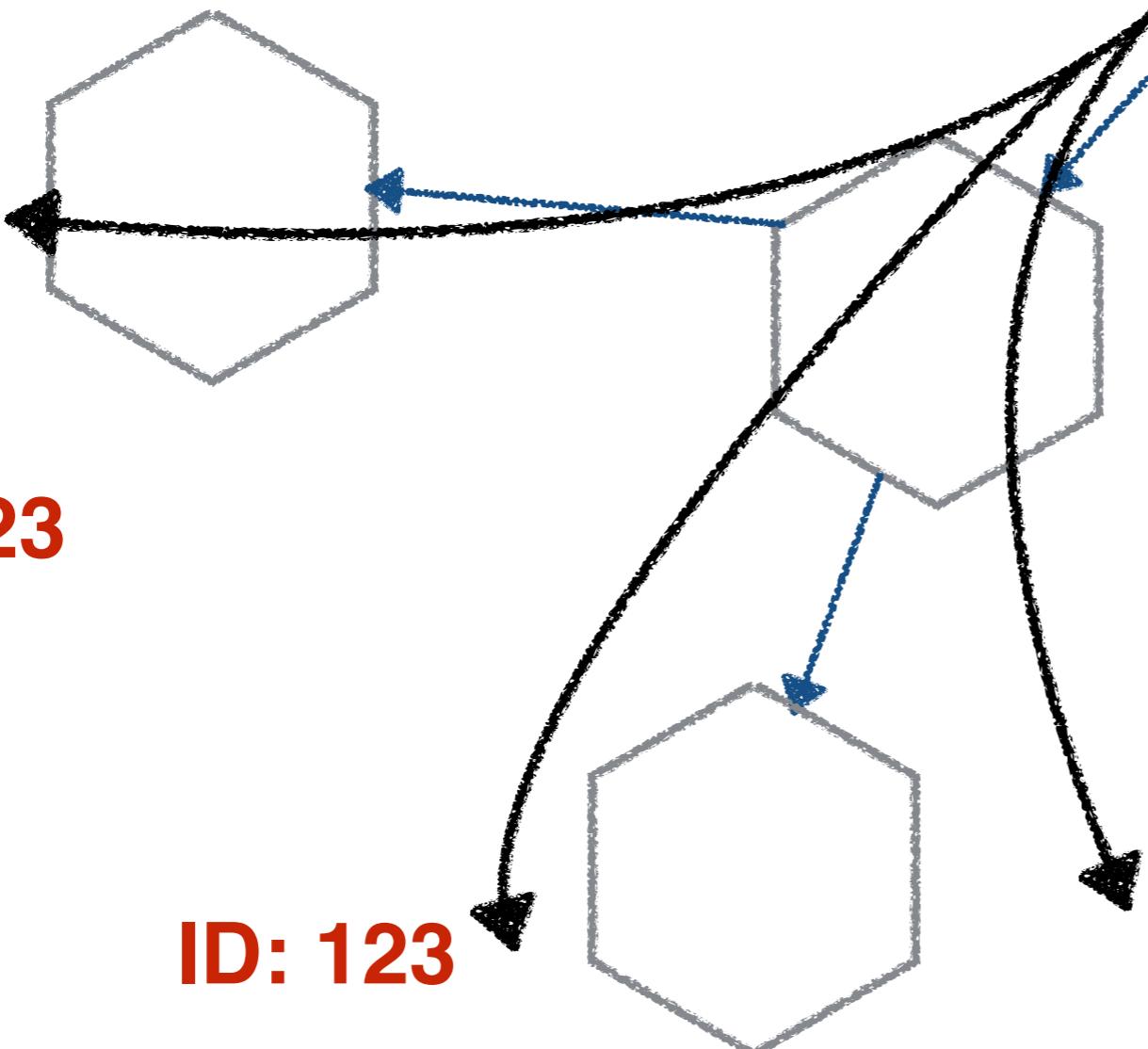
Terminal — java — 130x26

```
15:19:05,912 ERROR ~
@Konfig9
Internal Server Error (500) for request GET /posts/1

Template execution error (In /app/views/tags/display.html around line 18)
Execution error occurred in template /app/views/tags/display.html. Exception raised was ArithmeticException : / by zero.

play.exceptions.TemplateExecutionException: / by zero
    at play.templates.Template.throwException(Template.java:252)
    at play.templates.Template.render(Template.java:227)
    at play.templates.Template$TemplateExecutionTemplate.invokeTop(Template.java:369)
    at /app/views/Application/show.html.(line:21)
    at play.templates.Template.render(Template.java:287)
    at play.mvc.results.RenderTemplate$1.run(RenderTemplate.java:22)
    at play.mvc.Controller.renderTemplate(Controller.java:367)
    at play.mvc.Controller.render(Controller.java:393)
    at controllers.Application.show(Application.java:26)
    at play.utils.Java.invokeStatic(Java.java:129)
    at play.mvc.ActionInvoker.invoke(ActionInvoker.java:124)
    at Invocation.HTTP Request(Play!)
Caused by: java.lang.ArithmaticException: / by zero
    at java.math.BigDecimal.divide(BigDecimal.java:1327)
    at /app/views/tags/display.html.(line:18)
    at play.templates.Template.render(Template.java:287)
    ... 10 more
```

ID: 123



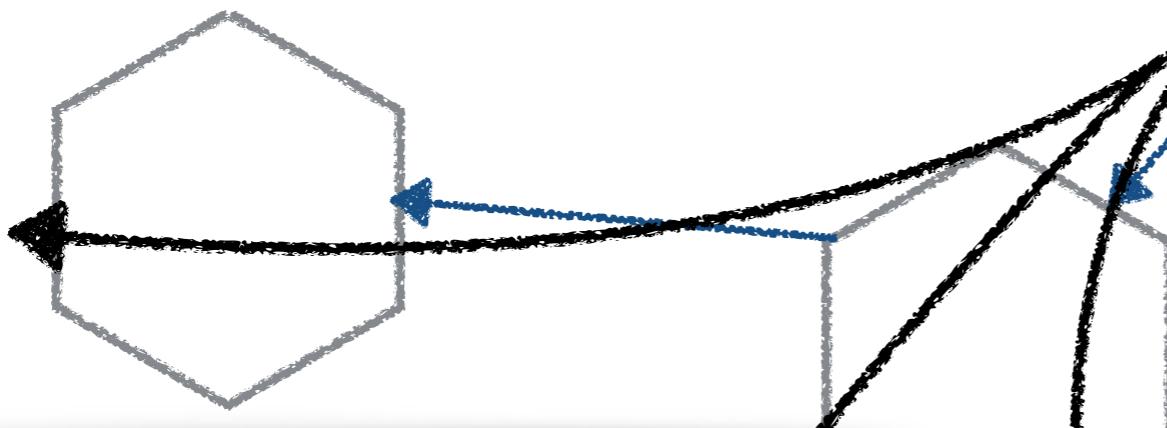
ID: 123

```
Terminal — java — 130x26
15:19:05,912 ERROR —
@Kor2nfg9
Internal Server Error (500) for request GET /posts/1

Template execution error (In /app/views/tags/display.html around line 18)
Execution error occurred in template /app/views/tags/display.html. Exception raised was ArithmeticException : / by zero.

play.exceptions.TemplateExecutionException: / by zero
    at play.templates.Template.throwException(Template.java:252)
    at play.templates.Template.render(Template.java:227)
    at play.templates.Template$TemplateExecutionTemplate.invokeTop(Template.java:369)
    at /app/views/Application/show.html.(line:21)
    at play.templates.Template.render(Template.java:287)
    at play.mvc.results.RenderTemplate$1.run(RenderTemplate.java:22)
    at play.mvc.Controller.renderTemplate(Controller.java:367)
    at play.mvc.Controller.render(Controller.java:393)
    at controllers.Application.show(Application.java:26)
    at play.utils.Java.invokeStatic(Java.java:129)
    at play.mvc.ActionInvoker.invoke(ActionInvoker.java:124)
    at Invocation.HTTP Request(Play!)
Caused by: java.lang.ArithmaticException: / by zero
    at java.math.BigDecimal.divide(BigDecimal.java:1327)
    at /app/views/tags/display.html.(line:18)
    at play.templates.Template.render(Template.java:287)
    ... 10 more
```

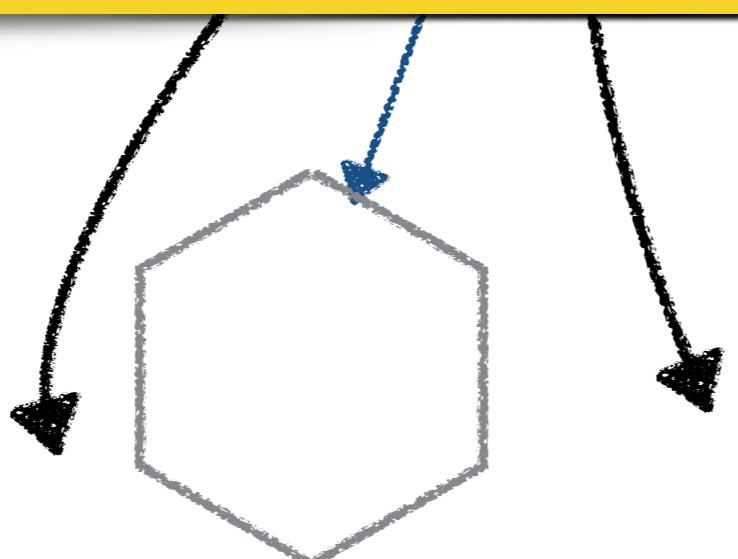
ID: 123



Terminal — java — 130x26
15:19:05,912 ERROR —

TIP 8 / 14: Use correlation IDs to track down nasty bugs

ID: 123



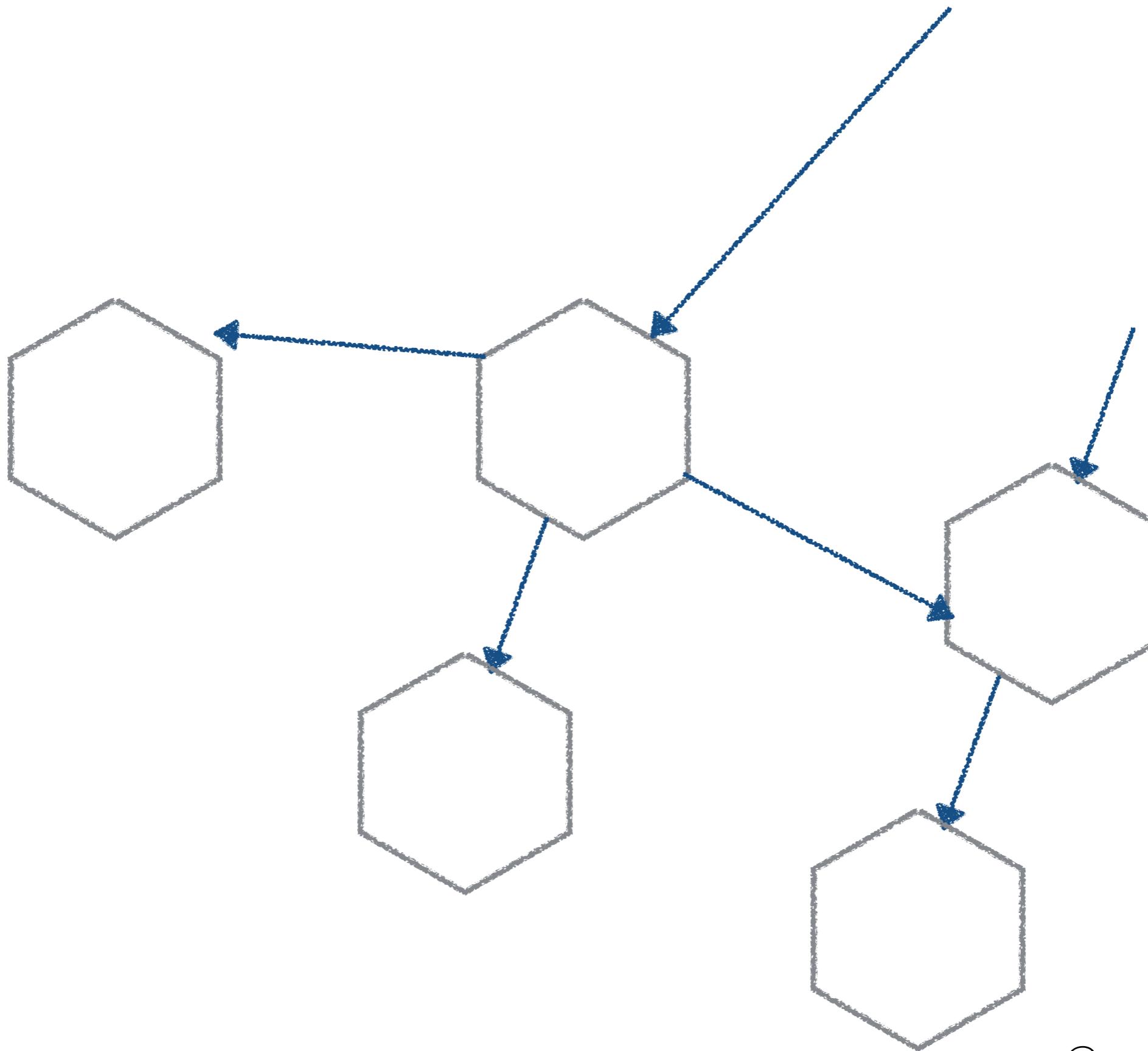
```
at play.templates.Template.throwException(Template.java:125)
at play.templates.Template.render(Template.java:227)
at play.templates.Template$TemplateExecutionTemplate.invokeTop(Template.java:369)
at /app/views/Application/show.html.(line:21)
at play.templates.Template.render(Template.java:287)
at play.mvc.results.RenderTemplate.<init>(RenderTemplate.java:22)
at play.mvc.Controller.renderTemplate(Controller.java:367)
at play.mvc.Controller.render(Controller.java:393)
at controllers.Application.show(Application.java:26)
at play.utils.Java.invokeStatic(Java.java:129)
at play.mvc.ActionInvoker.invoke(ActionInvoker.java:24)
at Invocation.HTTP Request(Play!)
```

Caused by: java.lang.ArithmaticException: / by zero
at java.math.BigDecimal.divide(BigDecimal.java:1327)
at /app/views/tags/display.html.(line:18)
at play.templates.Template.render(Template.java:287)
... 10 more

Deployment

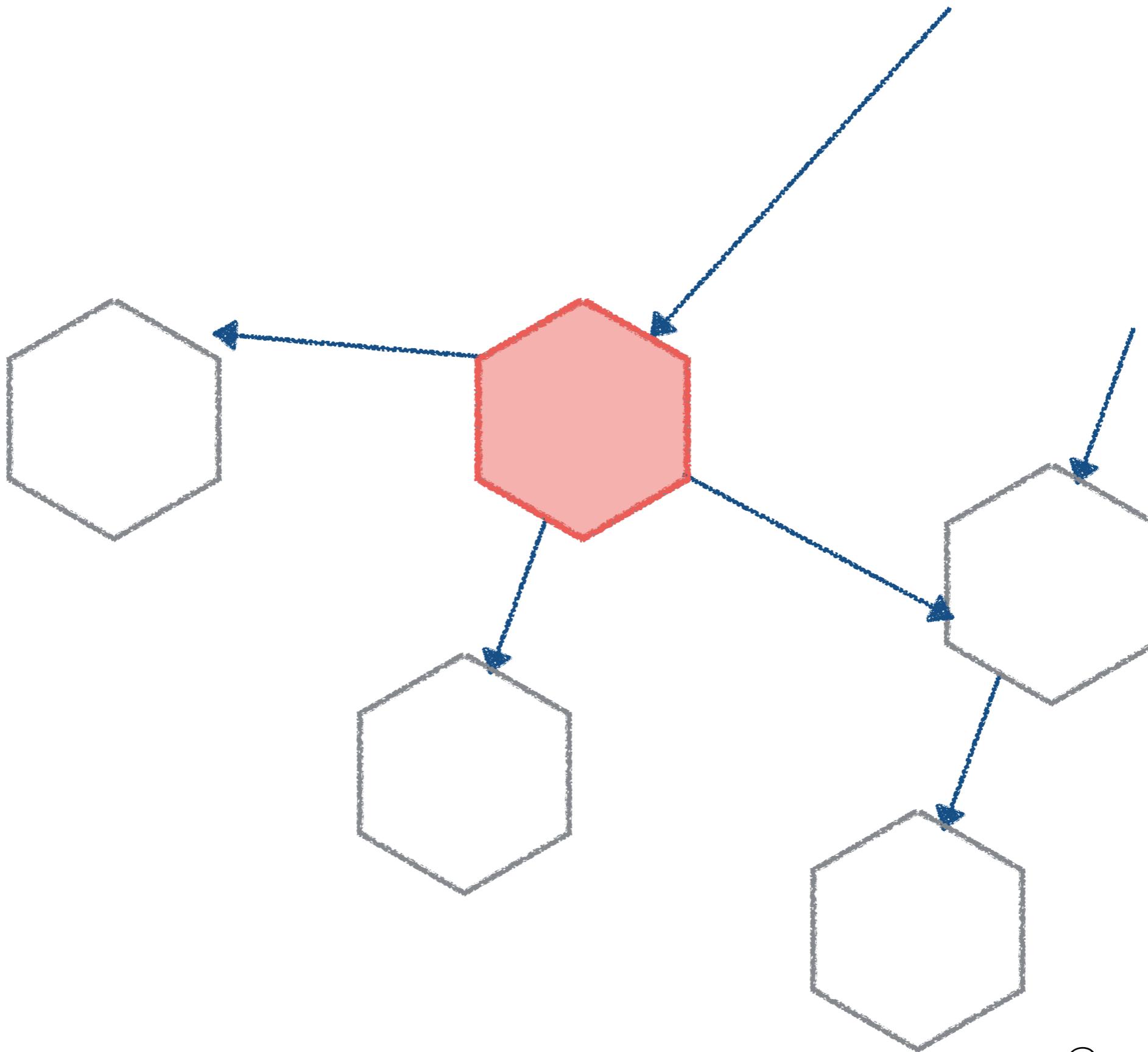
@javazone

@samnewman



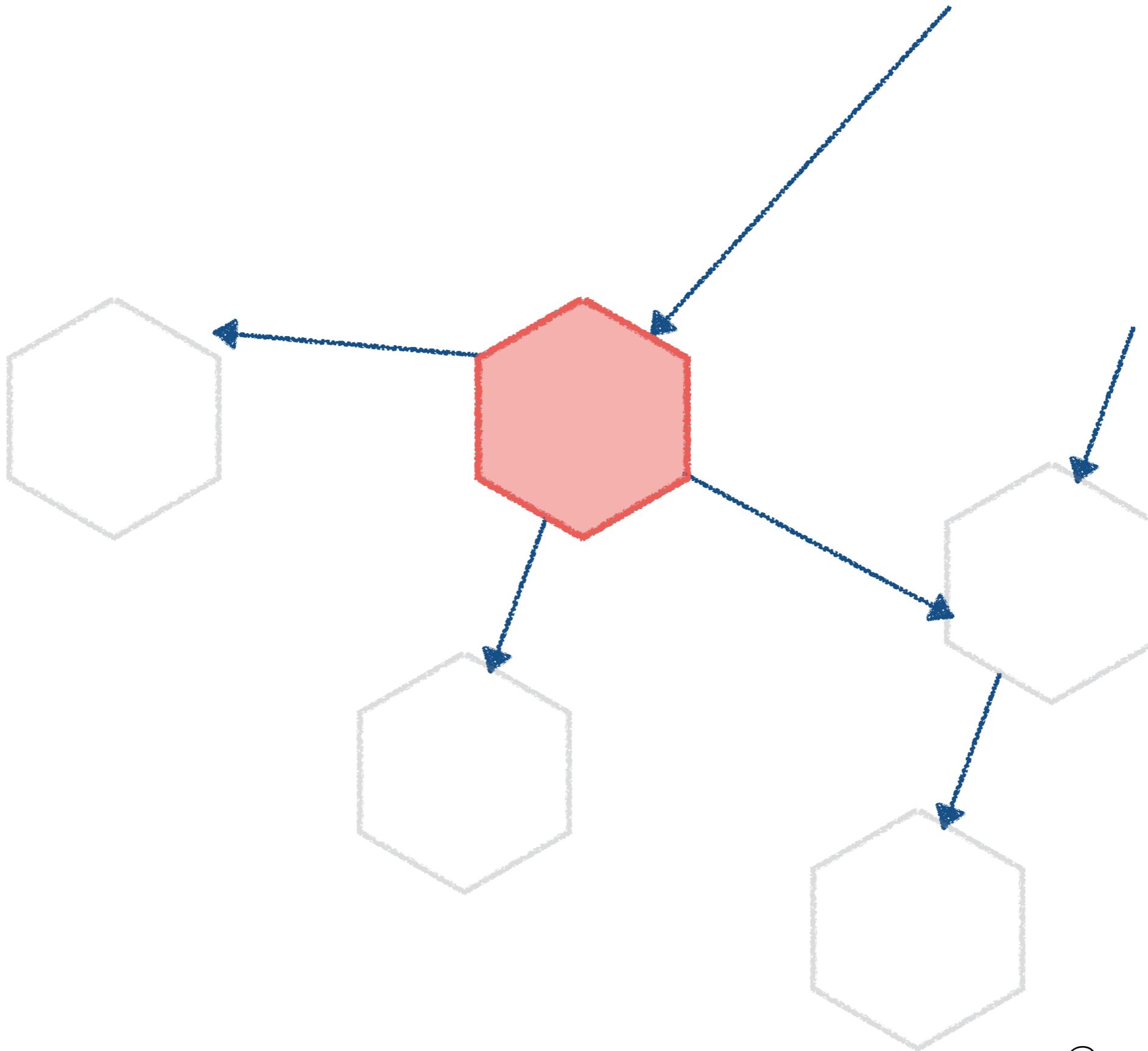
@javazone

@samnewman



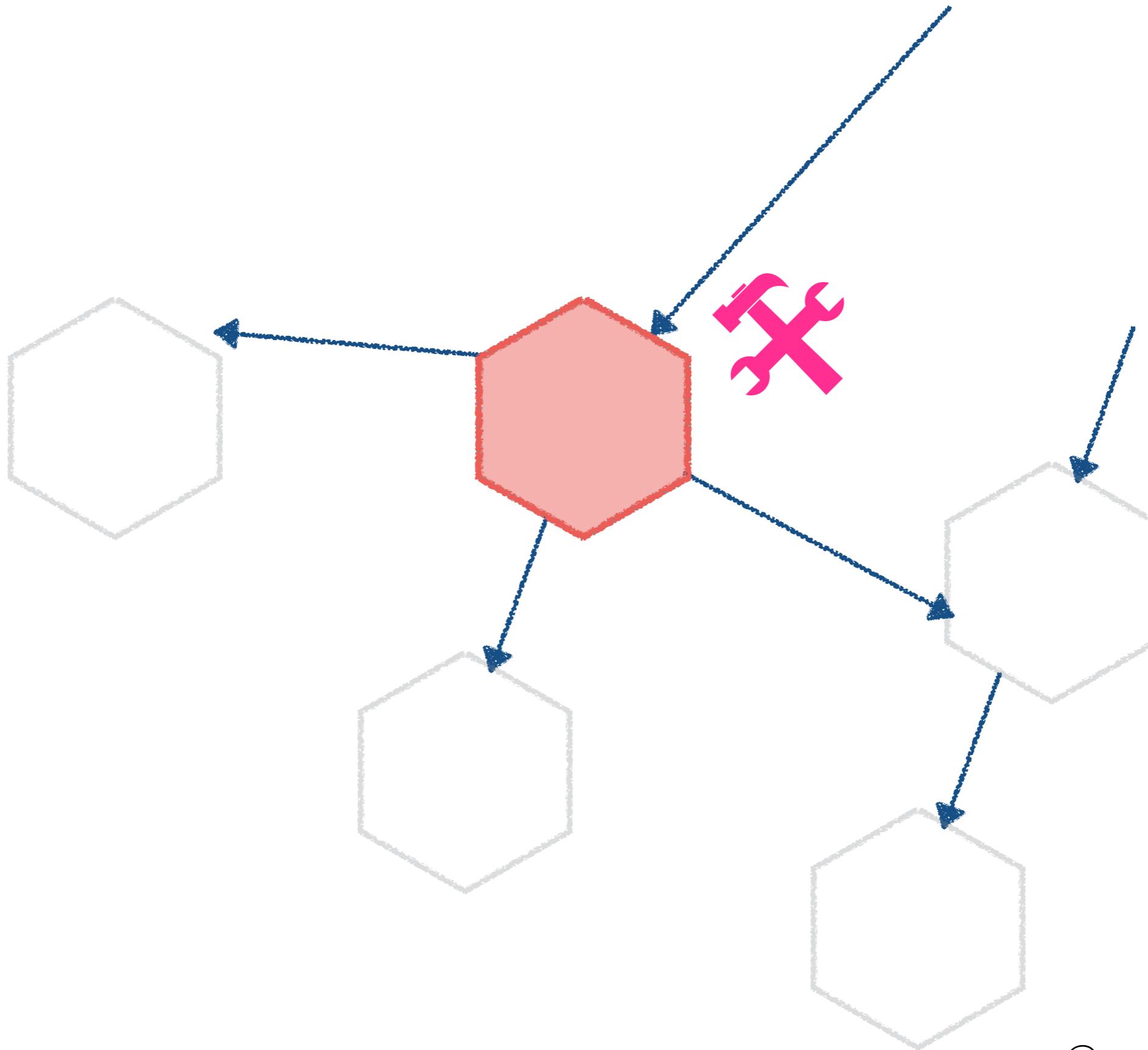
@javazone

@samnewman



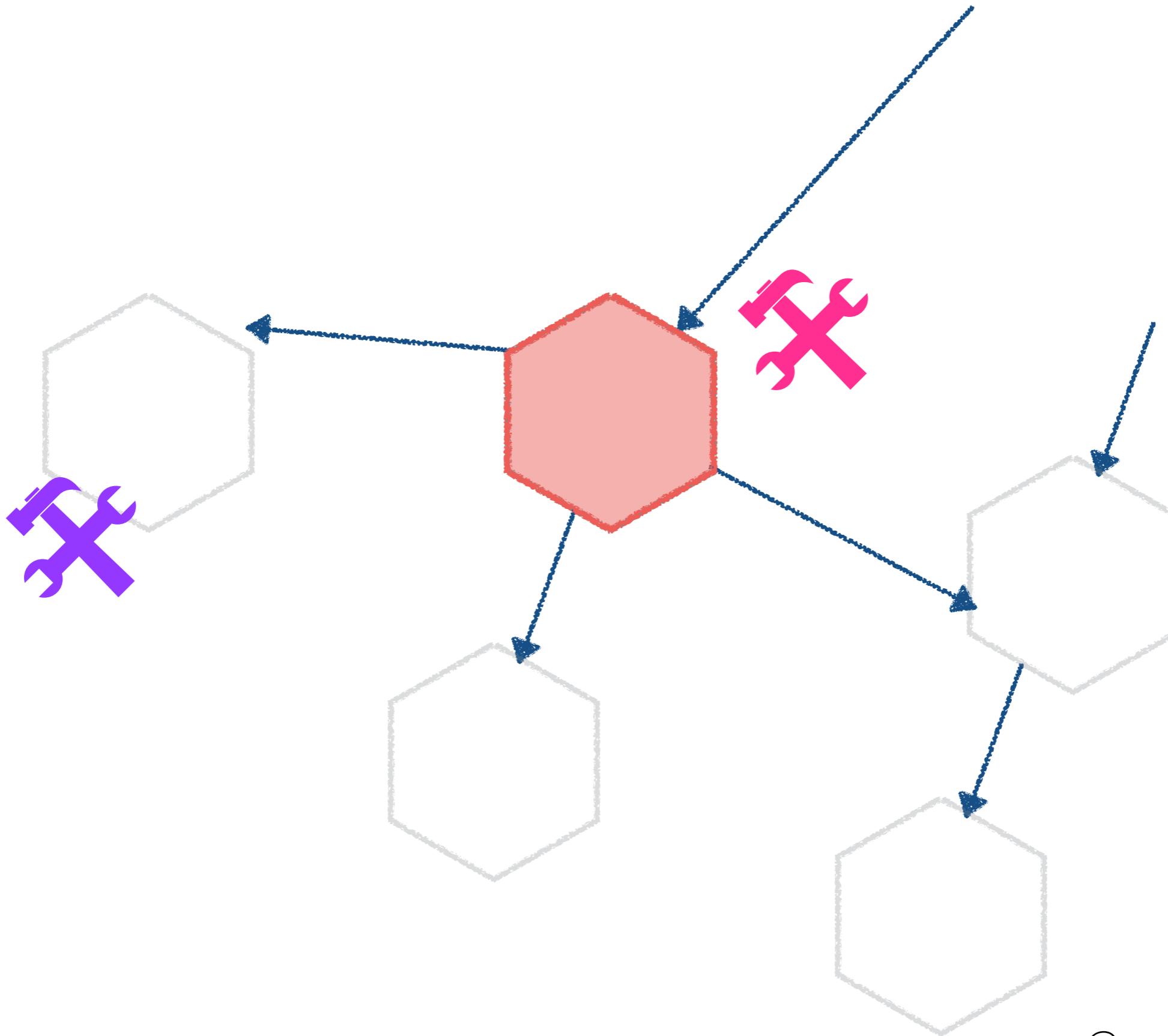
@javazone

@samnewman



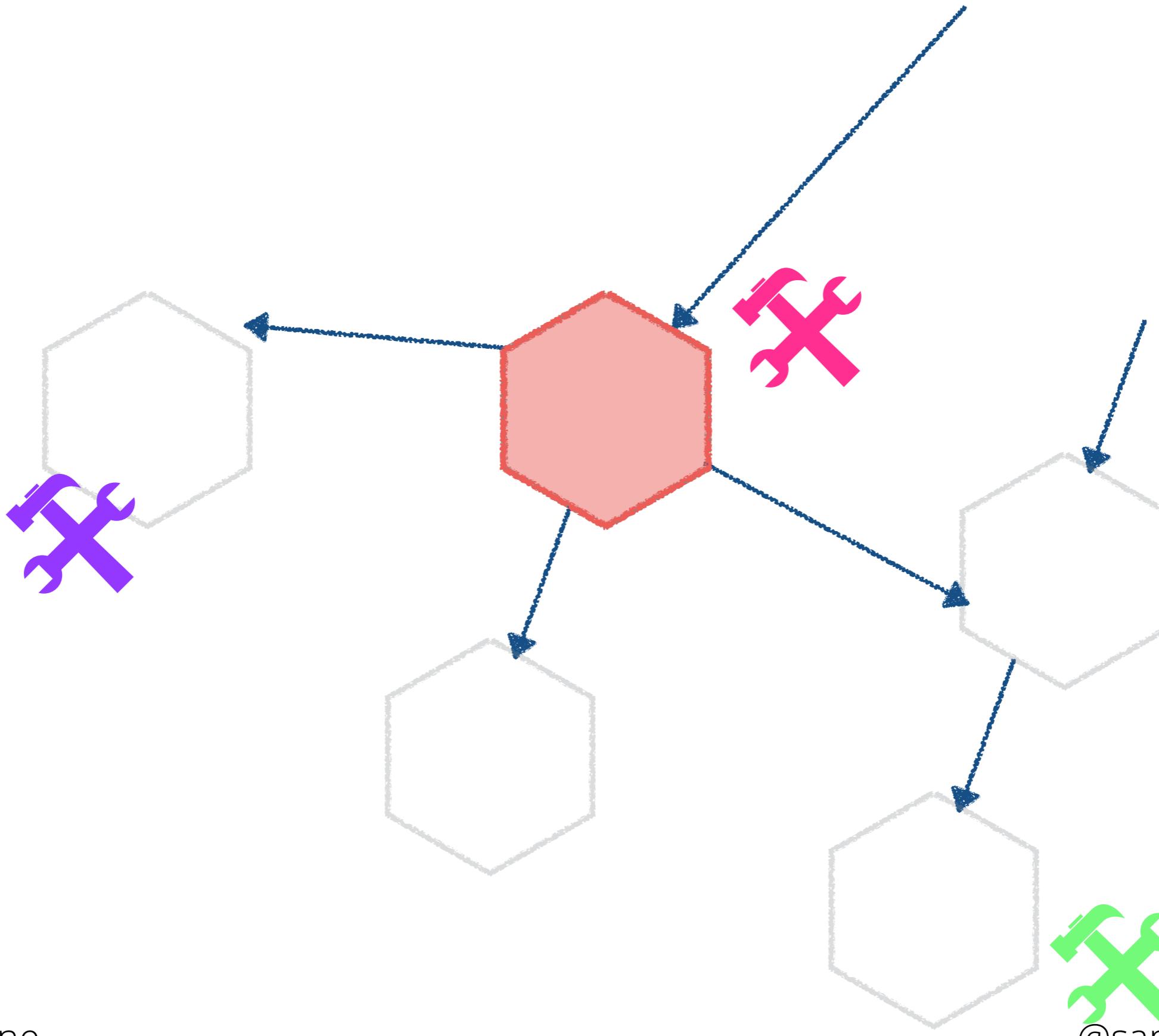
@javazone

@samnewman



@javazone

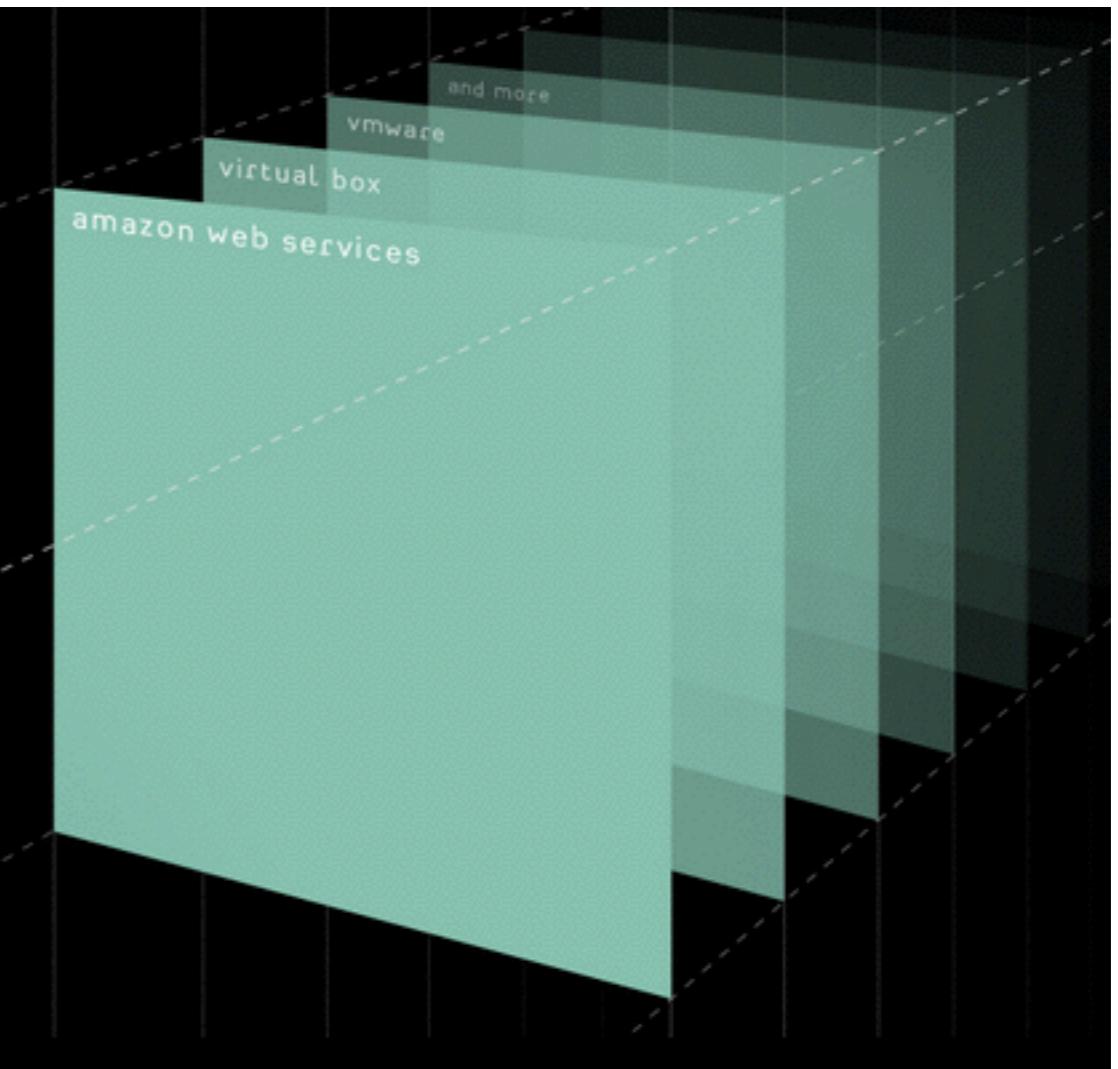
@samnewman



@javazone

@samnewman

Packer is a tool for creating identical machine images for multiple platforms from a single source configuration.





TERRAFORM

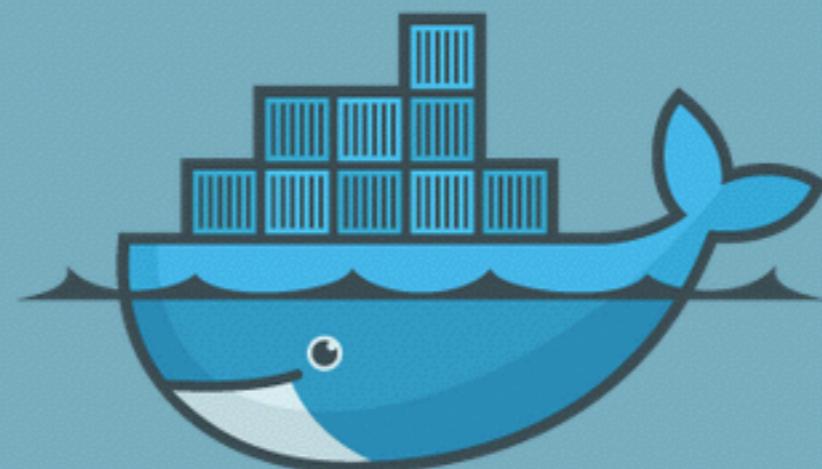
@javazone

@samnewman

[Home](#)[Learn More](#)[Getting started](#)[Community](#)[Documentation](#)[Blog](#)[INDEX](#)[sign up](#)

|

an open source project to pack, ship and run any application as a lightweight container



docker

@javazone

@samnewman

[Home](#)[Learn More](#)[Getting started](#)[Community](#)[Documentation](#)[Blog](#)[INDEX](#)[sign up](#)

|

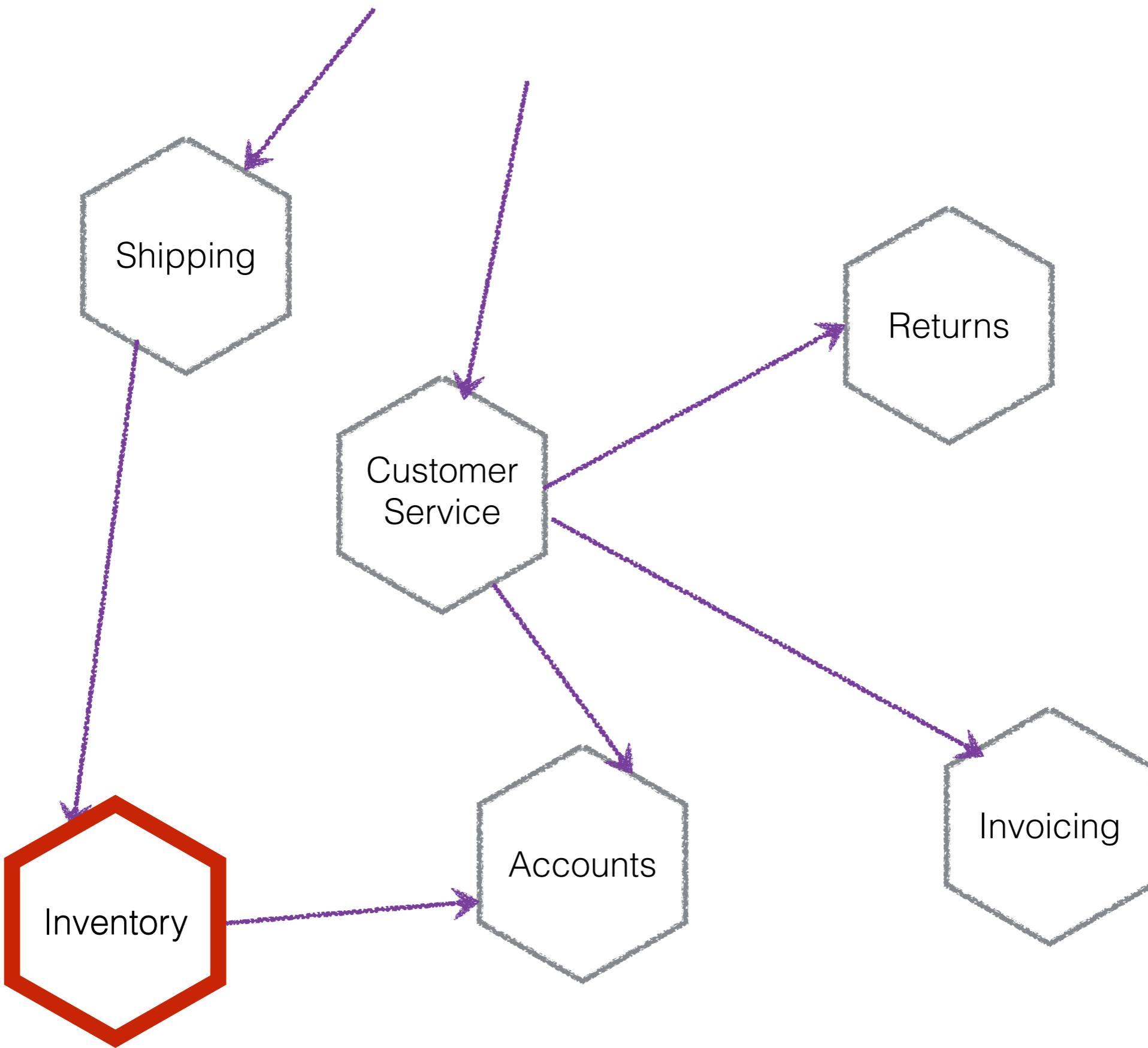
an open source project to pack, ship and run any application as a lightweight container

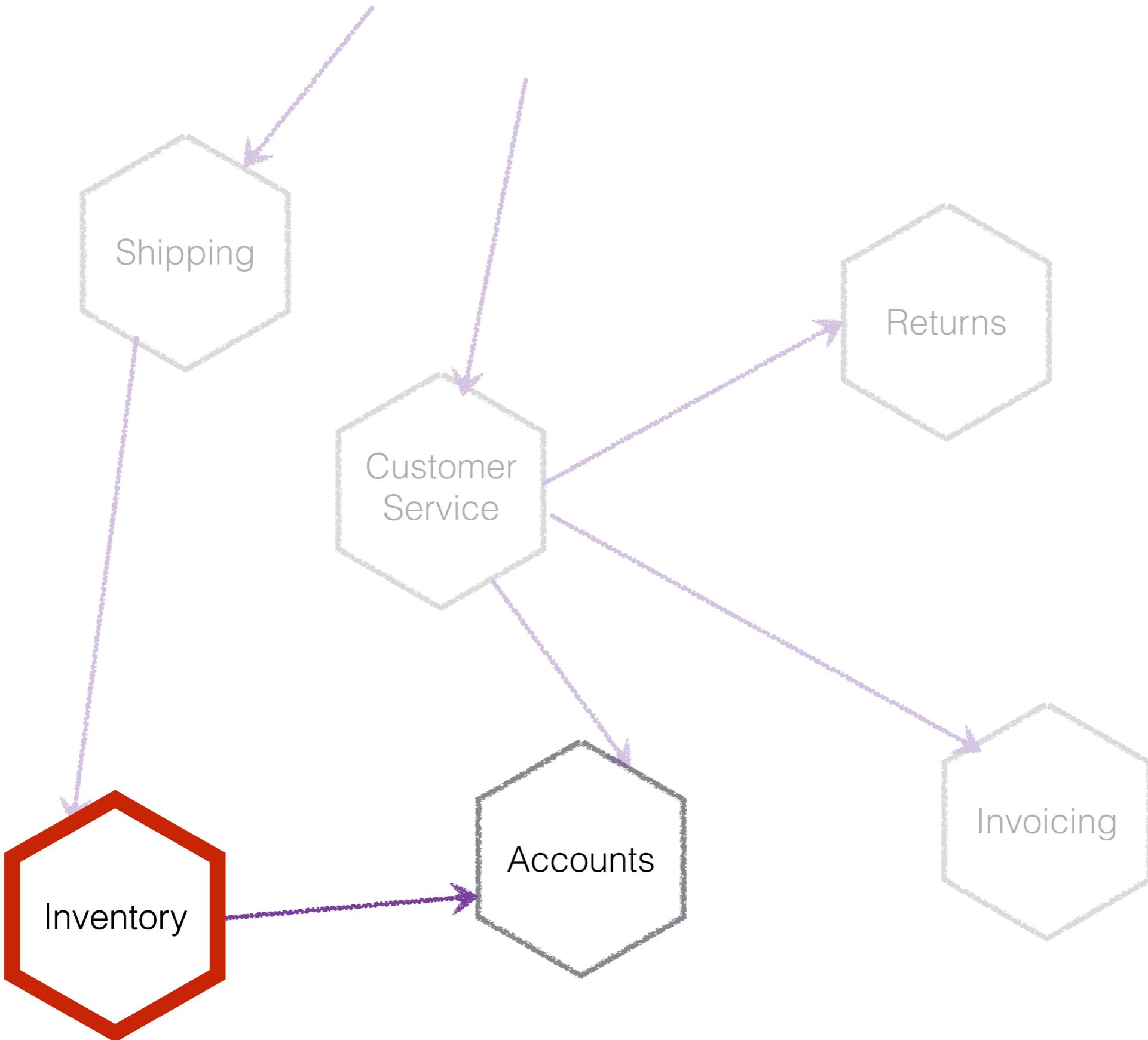
TIP 9 / 14: Abstract out underlying platform differences to provide a uniform deployment mechanism

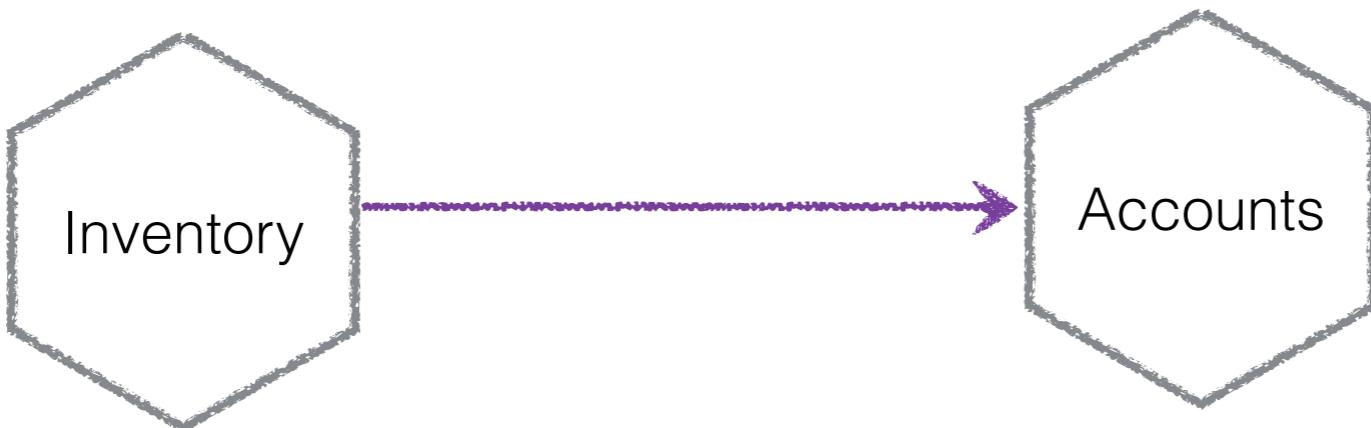


docker









 mountebank - over the wire test doubles

[home](#) [imposters](#) [logs](#) [config](#)

[Fork me on GitHub](#)

the apothecary

- [getting started](#)
- [install options](#)
- [command line](#)
- [faq](#)
- [support](#)
- [glossary](#)

api:

- [overview](#)
- [mock verification](#)
- [stubs](#)
- [stub predicates](#)
- [stub proxies](#)
- [stub injection](#)
- [errors](#)

protocols:

Welcome, friend

mountebank is the first tool to provide multi-protocol, multi-language test doubles over the wire. Simply point your application under test to mountebank instead of the real dependency, and test like you would with traditional stubs and mocks.

mountebank will cure what ails you!

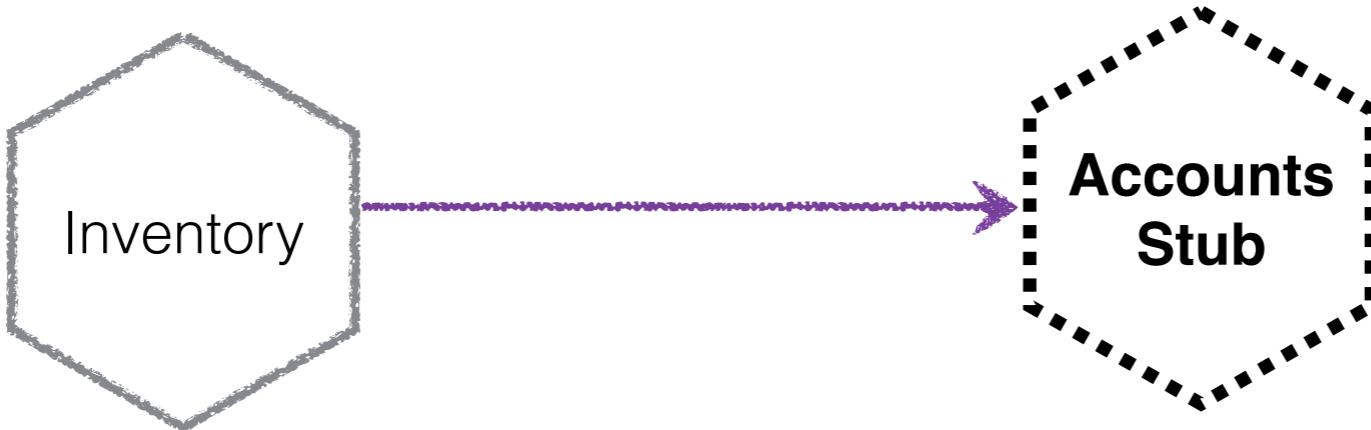
How it works

mountebank employs a legion of imps to serve as test doubles over the wire using the protocol of your choice. In the typical use case, each test will have its own imposter which handles requests and performs a specific type of teardown.

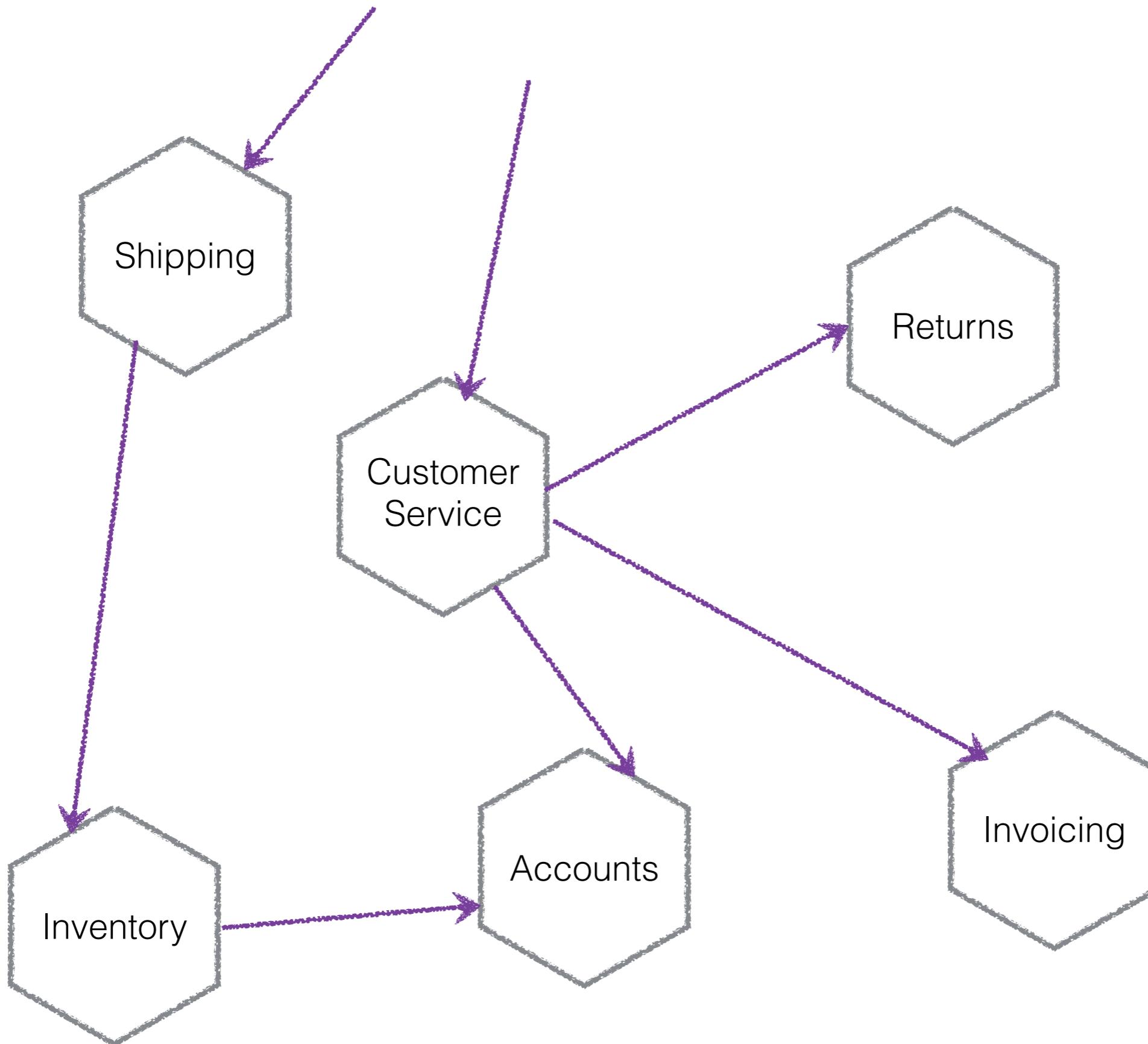
mountebank employs several types of imposta

Mountebank

<http://www.mbttest.org>

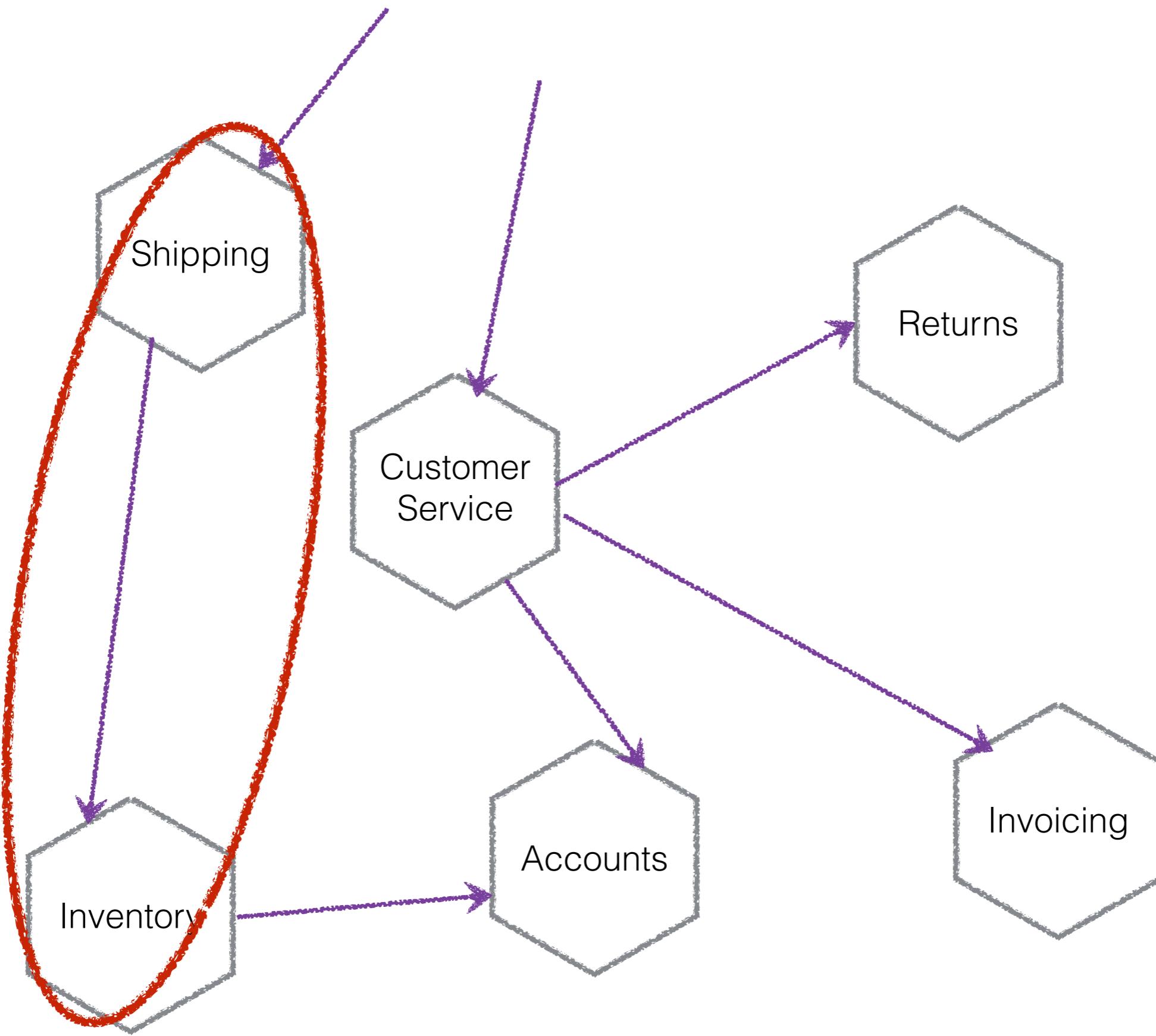


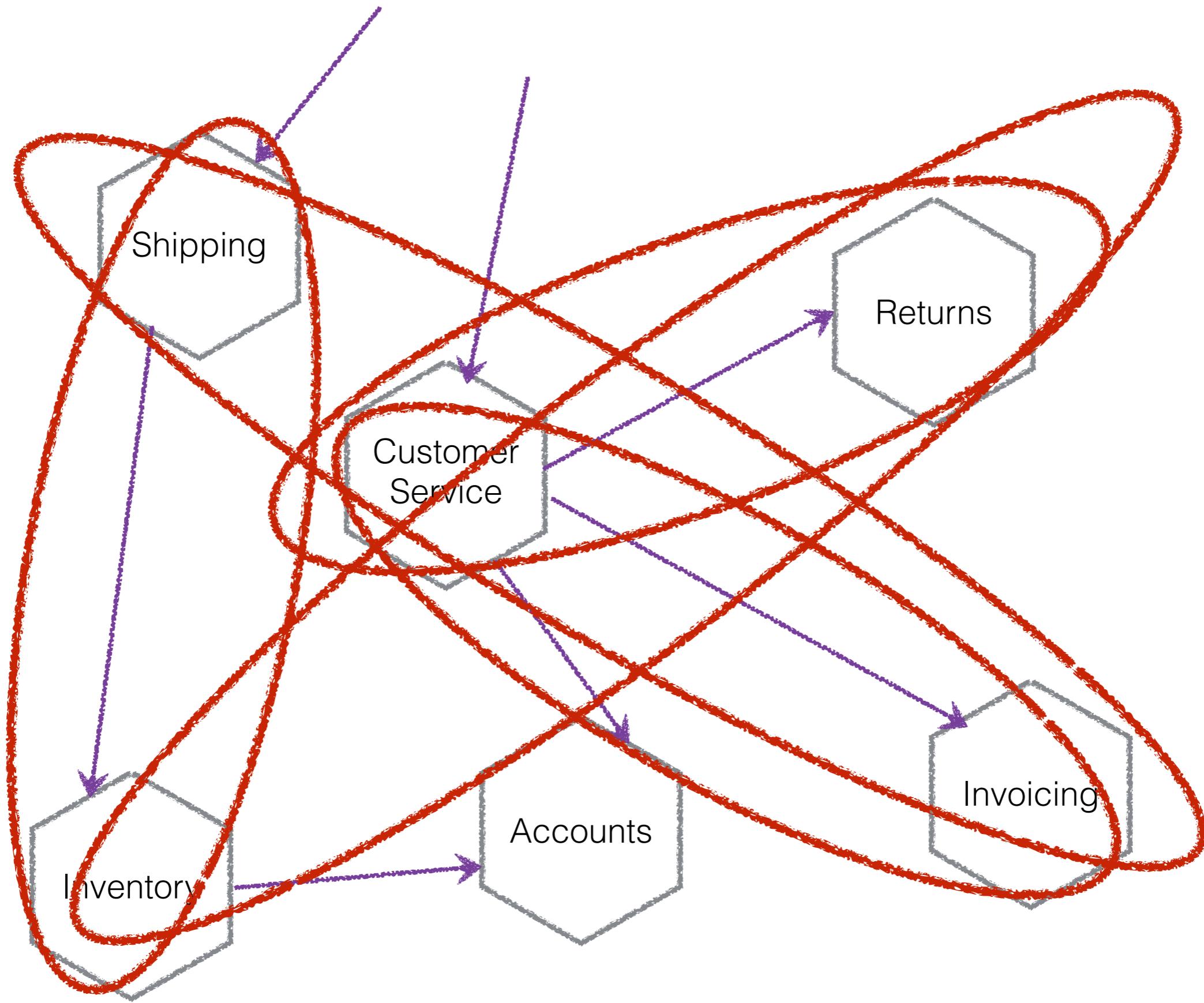
The screenshot shows the homepage of the Mountebank website (<http://www.mbtest.org>). The page features a central hero section with the text "Welcome, friend" and a large "Mountebank" logo. To the left, there is a sidebar with navigation links for "the apothecary" (getting started, install options, command line, faq, support, glossary) and "api:" (overview, mock verification, stubs, stub predicates, stub proxies, stub injection, errors). Below these are "protocols:" links. The main content area includes a "Fork me on GitHub" button and sections for "How it works" and "mountebank is the first tool to provide multi-protocol, multi-language test doubles over the wire. Simply point your application under test to mountebank instead of the real dependency, and test like you would with traditional stubs and mocks."











@javazone

@samnewman

Browsers

@javazone

@samnewman

Timing

Browsers

@javazone

@samnewman

Provisioning of Environments

Timing

Browsers

Networks

Provisioning of Environments

Timing

Browsers

Deployment

Networks

Provisioning of Environments

Timing

Browsers

Deployment

Networks

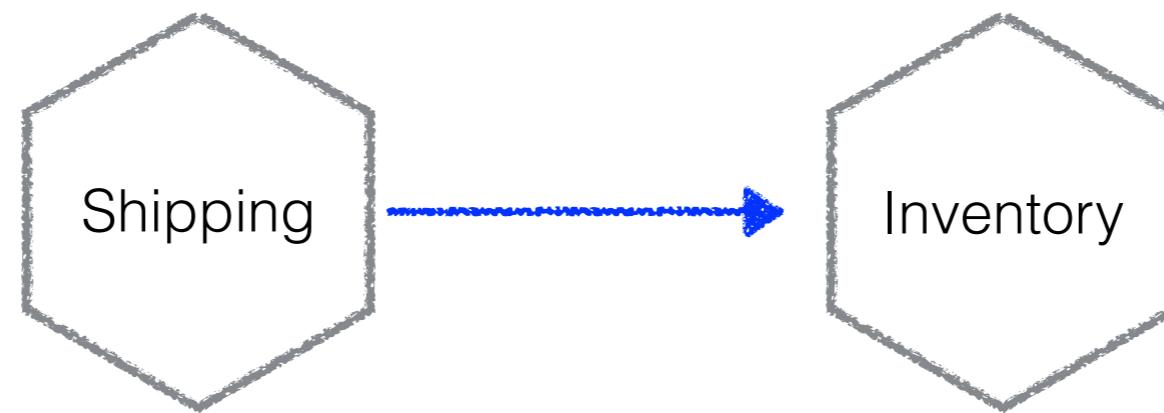
Provisioning of Environments

Timing

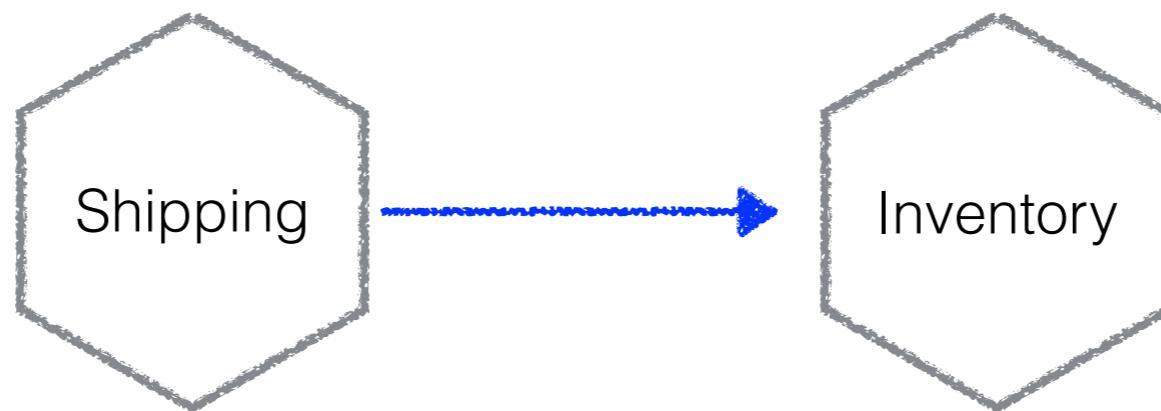
Browsers

Diagnosis

CONSUMER-DRIVEN CONTRACTS

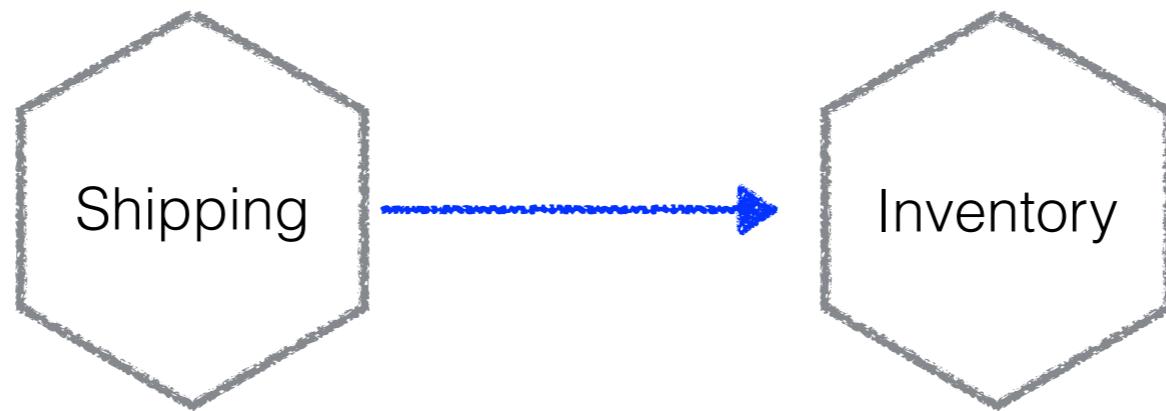


CONSUMER-DRIVEN CONTRACTS



Expectations

CONSUMER-DRIVEN CONTRACTS



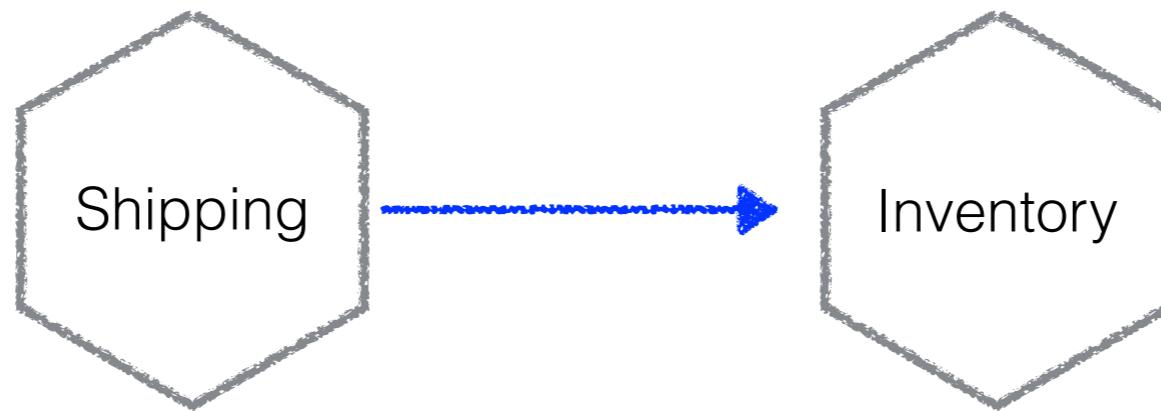
Expectations



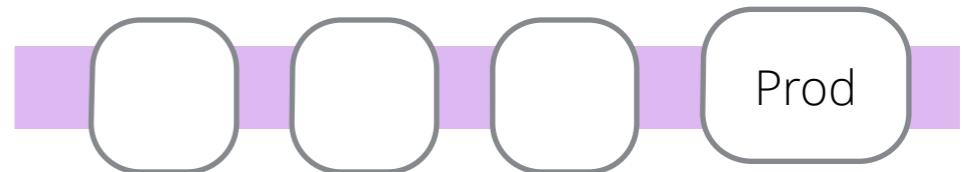
```
26 def animal(): Node = {
27     attr: "http://www.w3.org/1999/xhtml" nsLang="en" lang="en"
28     children:
29         click ref="styleSheet" href="/static/css/common.css" type="text/css"/>
30         click ref="styleSheet" href="/static/css/bootstrap.css" type="text/css"/>
31         meta http-equiv="refresh" content="30"/>
32     head:
33         style
34             { content(buildId) }
35         script
36     html
37 }
```

```
38
39 private def content(buildId: List[Build]): Element = {
40     displayType match {
41         case "single" => divs { buildId.map(build => article(build)) } .divs
42         case "series" => {
43             if (builds.length == 1) {
44                 divs { buildId.map(build => article(build)) } .divs
45             } else {
46                 val classs="builds"&{ buildId.map(build => build.id.substring(buildId)) }.divs
47             }
48         }
49         case _ => val classs="builds"&{ buildId.map(build => build.id.substring(buildId)) }.divs
50     }
51 }
52
53 private def article(buildId: Build): Element = {
54     stable classs("Build") & build.getPreviousName.followedBy
55         str("single","stable") .attr("center")
56             .divs { link(buildId) } .divs
57         .divs
58     .divs
59 }
```

CONSUMER-DRIVEN CONTRACTS

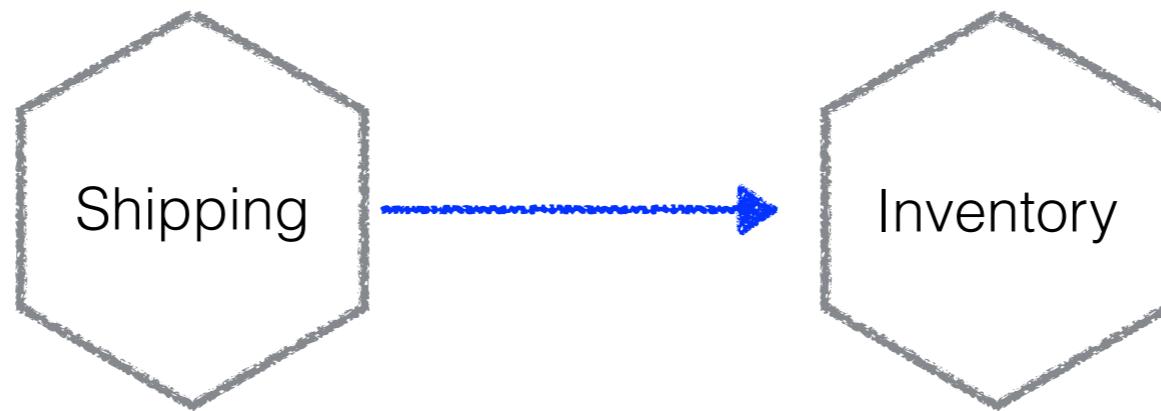


Expectations



```
26 def animal(): Node = {
27     attr: "http://www.w3.org/1999/xhtml" nsLang="en" lang="en"
28     head:
29         elem: "styleSheet", href="/static/css/common.css", type="text/css"
30         elem: "styleSheet", href="/static/css/app.css", type="text/css"
31         meta: "http-equiv", content="refresh", value="0"
32     head:
33     body:
34         content: "Builds"
35     body:
36     attr: "xmlns"
37 }
38
39 private def content(builds: List[Build]): Element = {
40     displayType match {
41         case "single" => ul(id = "builds", map(build => available(build)))
42         case "series" => if (builds.length == 1) {
43             ul(id = "builds", map(build => available(build)))
44         } else {
45             val class: "builds" = build.map(build => build.id == buildId).reduce(
46                 _ + _)
47             case _ => val class: "builds" = build.map(build => build.isFailed(build))
48         }
49     }
50 }
51
52 private def available(buildId: BuildId): Element = {
53     stable class: "Build" = build.getOrThrow.name.toLowerCase
54     stable val type: "Build" = build.type.name.toLowerCase
55     stable val id: "BuildId" = buildId.id
56     stable val status: "BuildStatus" = build.state
57 }
```

CONSUMER-DRIVEN CONTRACTS



Expectations



```
26 def animal(): Node = {
27     attr: "http://www.w3.org/1999/xhtml" nsLang="en" lang="en"
28     children:
29         elem ref="styleSheet" href="/static/css/common.css" type="text/css">
30             elem ref="styleSheet" href="/css/bootstrap.css" type="text/css">
31             meta http-equiv="refresh" content="0"/>
32         head:
33             elem
34                 [ content(buildId) ]
35             endp
36         endp
37 }
38
39 private def content(buildId: List[build]): Elem = {
40     displayType match {
41         case "single" => elem { build.map(build => available(build)) } .with
42             case "multi" => if
43                 build.length == 1 {
44                     elem { build.map(build => available(build)) } .with
45                 } else {
46                     val classs="builds"<{ build.map(build => build.id == buildId) }>.with
47                     classs
48                 }
49         case _ => val classs="builds"<{ build.map(build => build.id == buildId) }>.with
50             classs
51     }
52 }
53
54 private def available(buildId: build): Elem = {
55     stable class("build") <{ build.getUniqueId.name.toLowerCase }>
56         .with("width" allign="center")
57             .with( buildId == buildId ) .with
58             .with
59         endp
60 }
```



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: 

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>

Pact

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

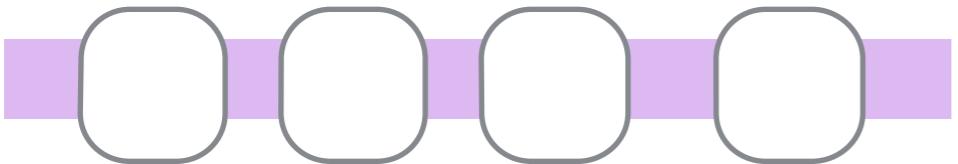
Pact provides an RSpec DSL for service consumers to define the HTTP requests they will make to a service provider and the HTTP responses they expect back. These expectations are used in the consumers specs to provide a mock service provider. The interactions are recorded, and played back in the service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

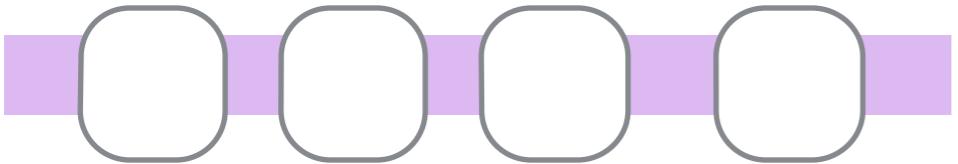
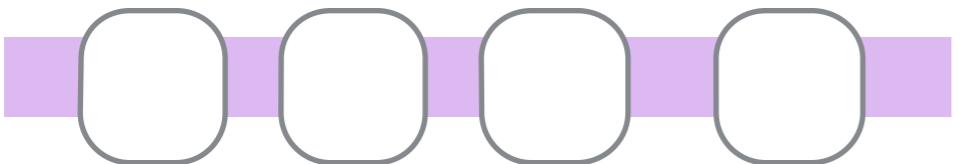
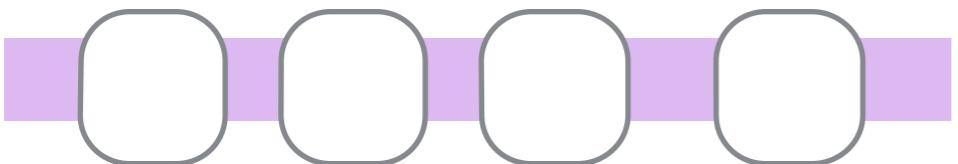
This gem is inspired by the concept of "Consumer driven contracts". See
<http://martinfowler.com/articles/consumerDrivenContracts.html> for more information.

TIP 10 / 14: Consumer Driven Tests to catch breaking changes

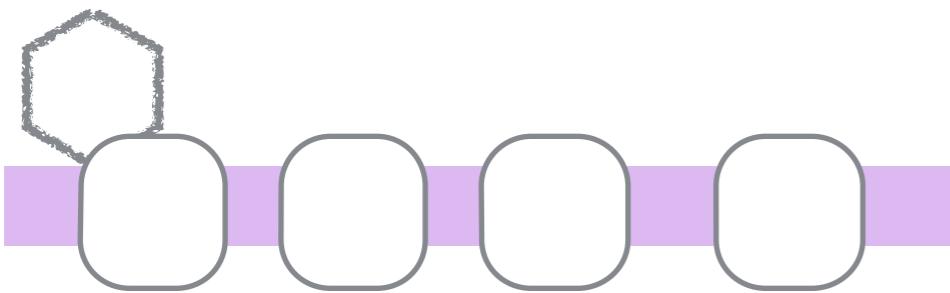
<https://github.com/realestate-com-au/pact>



Pending...

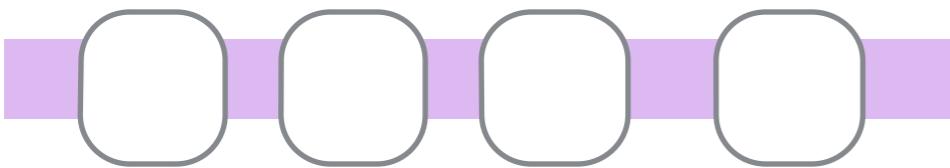
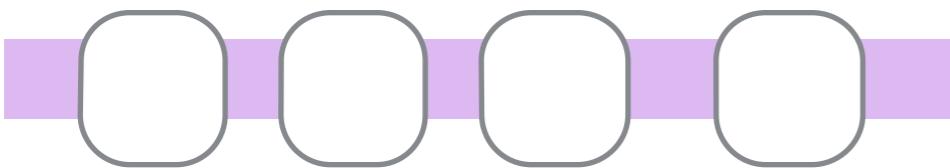
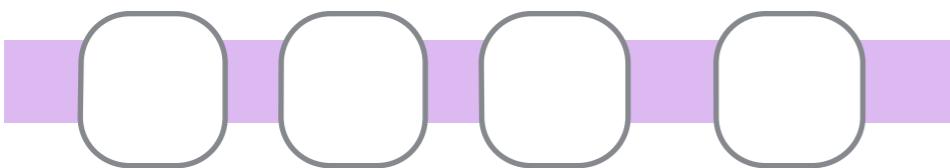


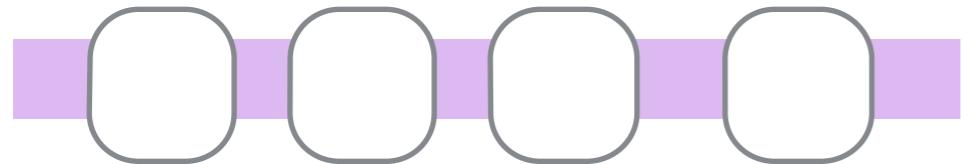
Prod



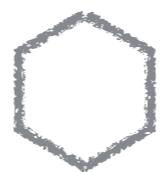
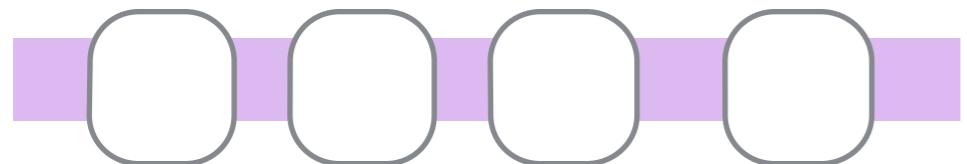
Pending...

Prod

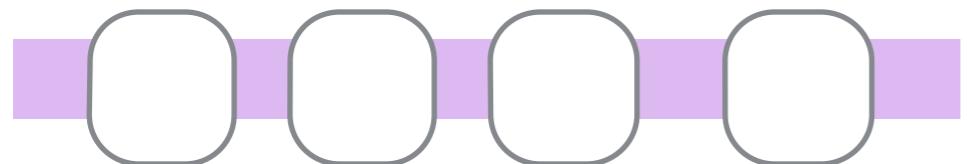
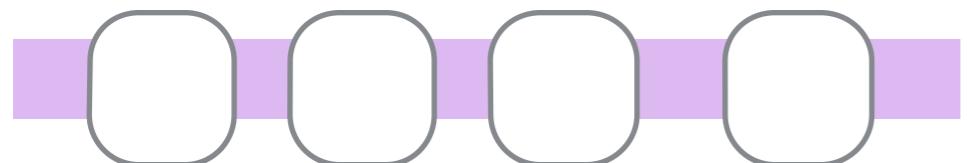


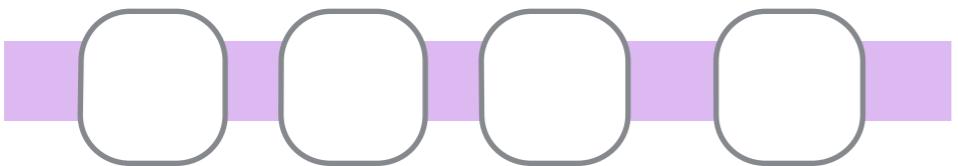


Pending...

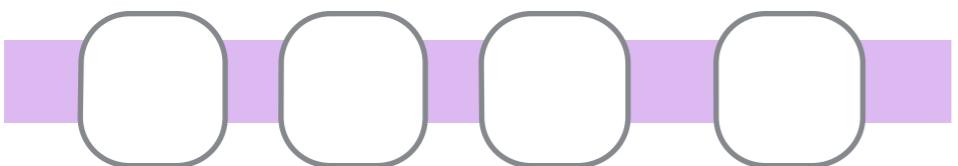


Prod

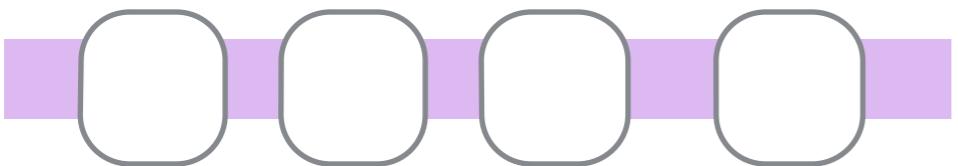
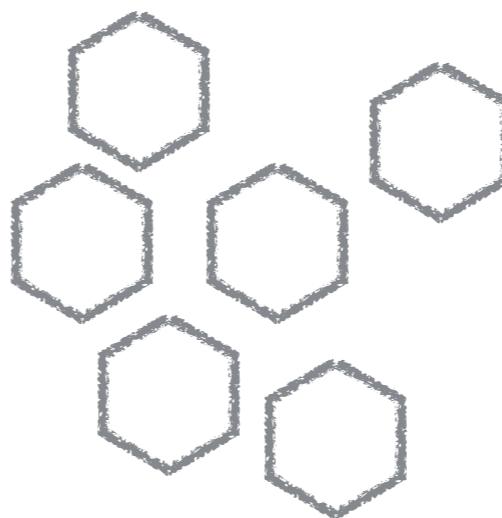
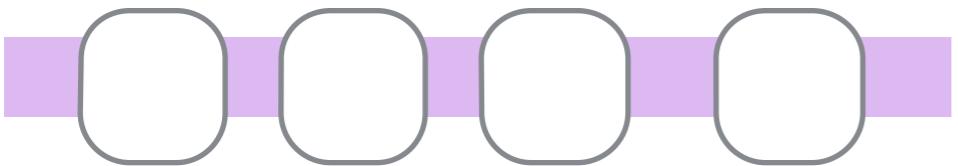


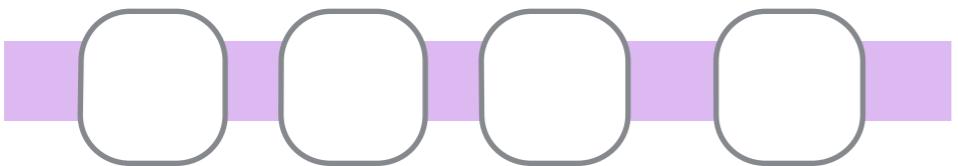


Pending...

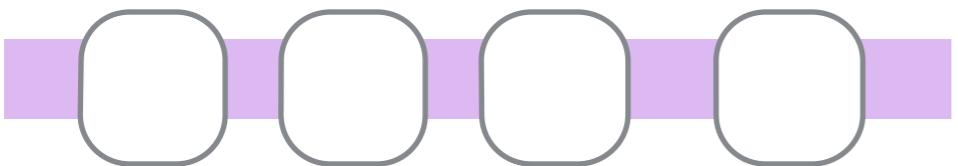
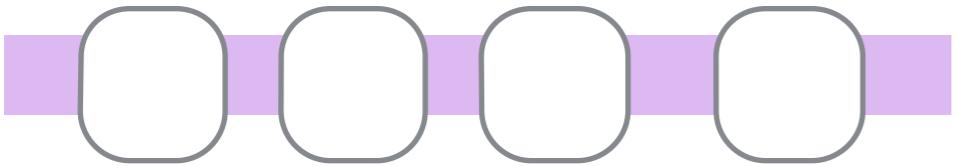
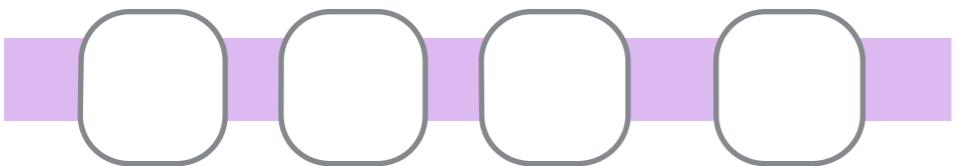


Prod

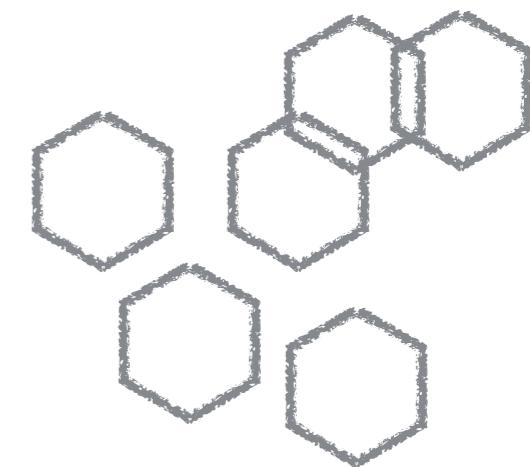


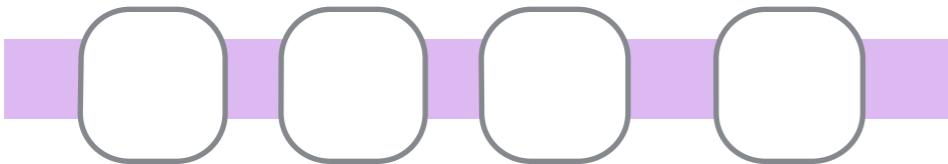


Pending...

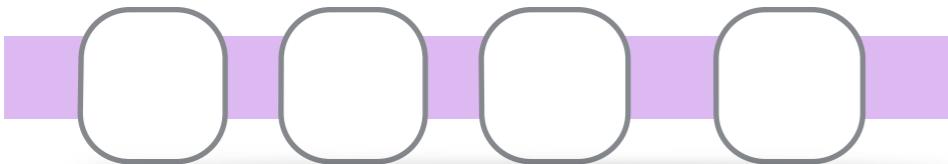


Prod



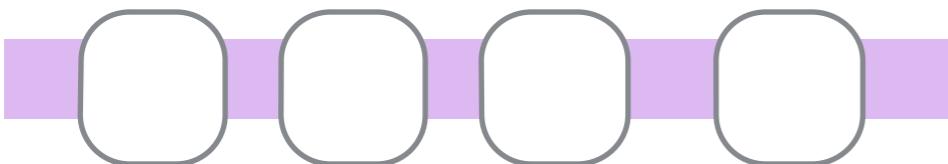


Pending...



Prod

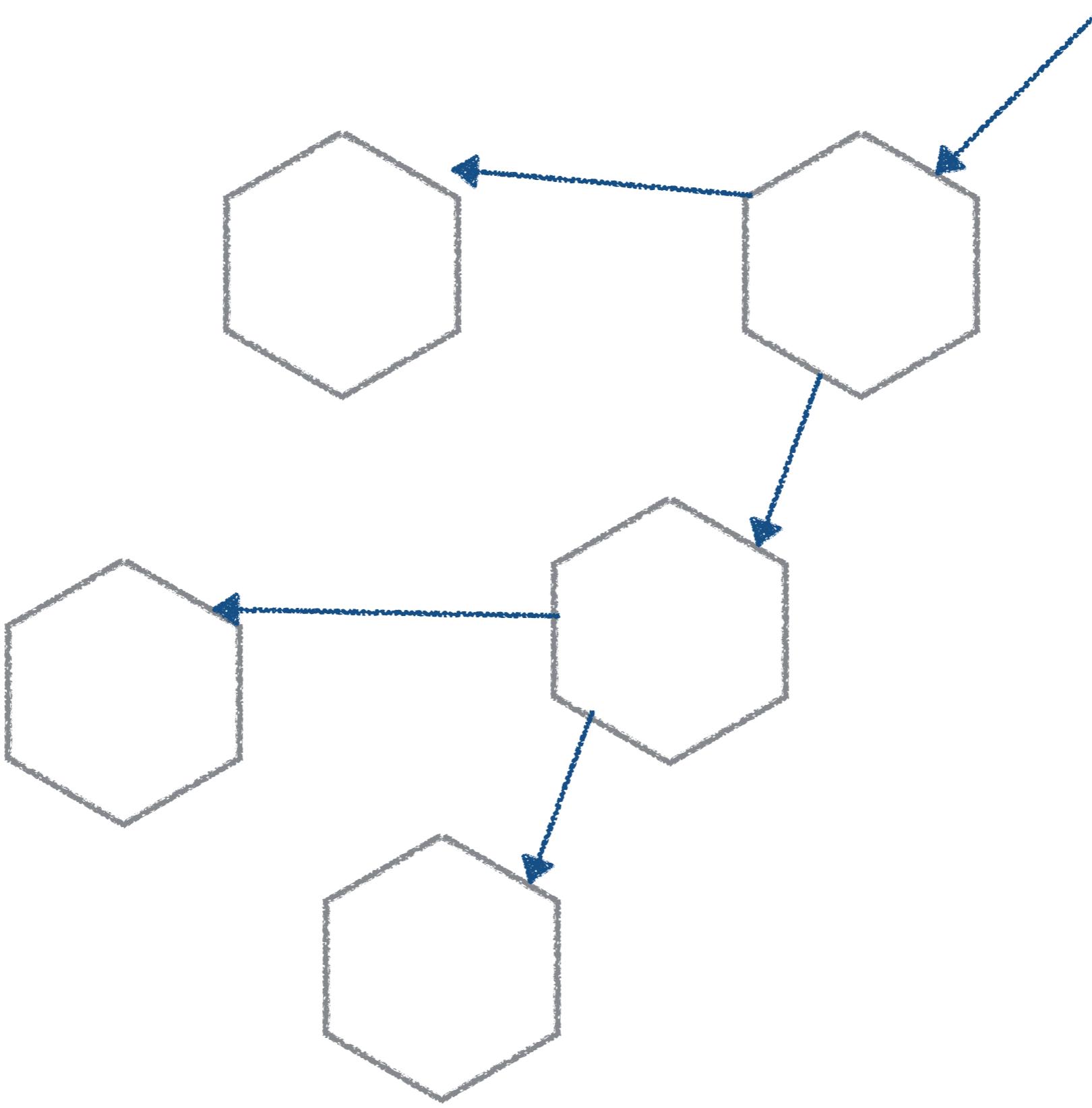
TIP 11 / 14: Don't let changes build up - release as soon as you can, and preferably one at a time!



Architectural Safety

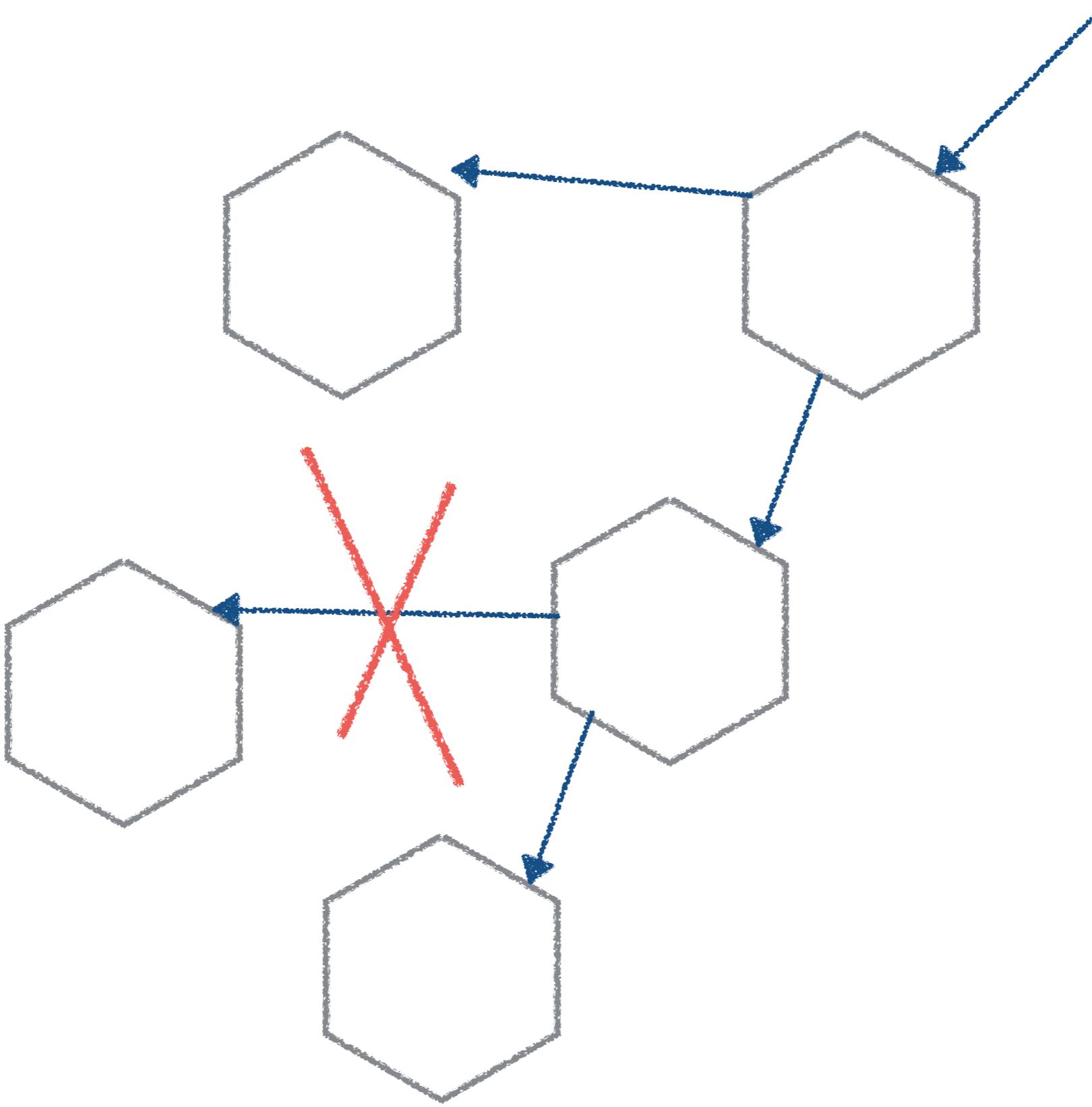
@javazone

@samnewman



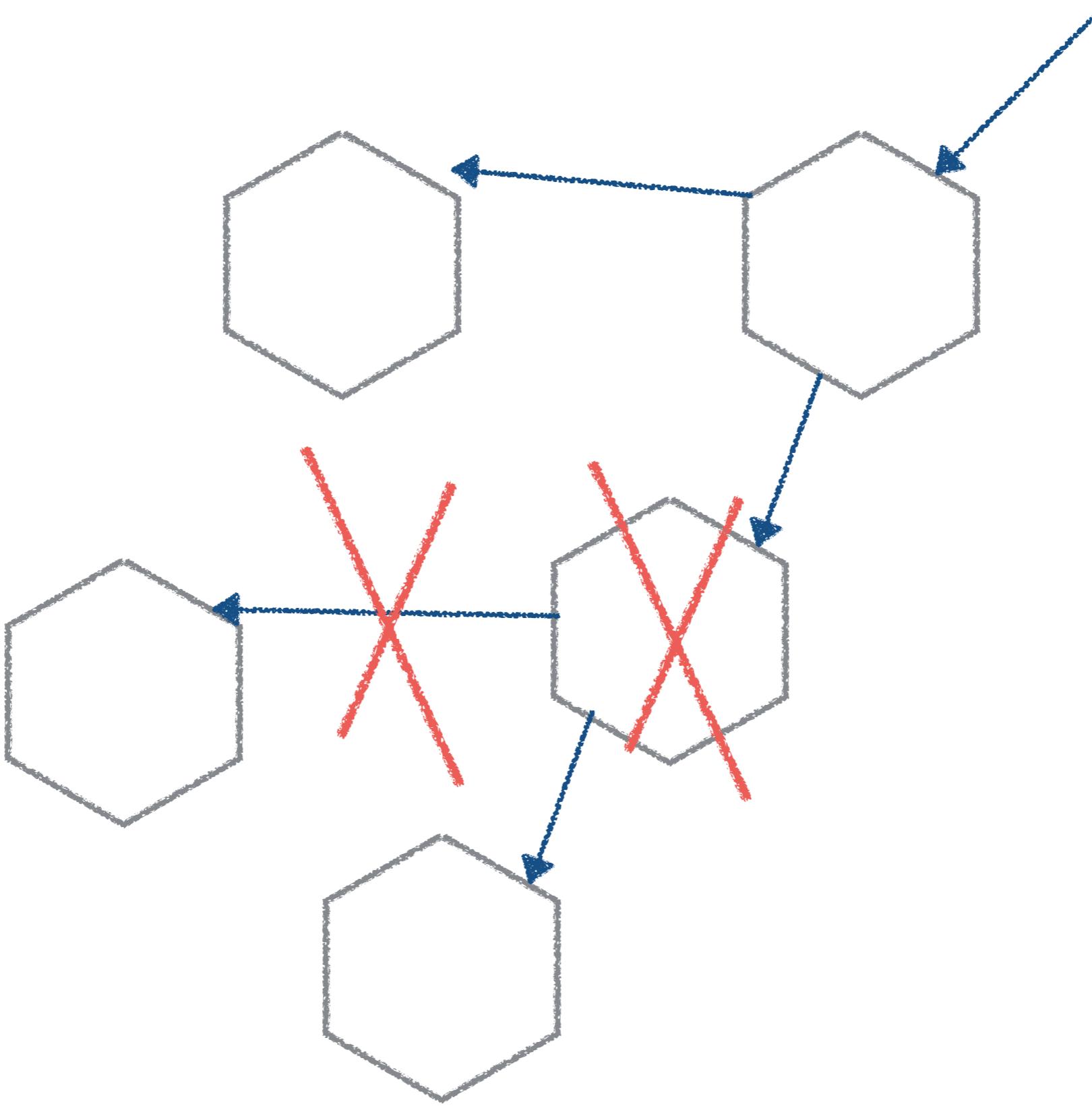
@javazone

@samnewman



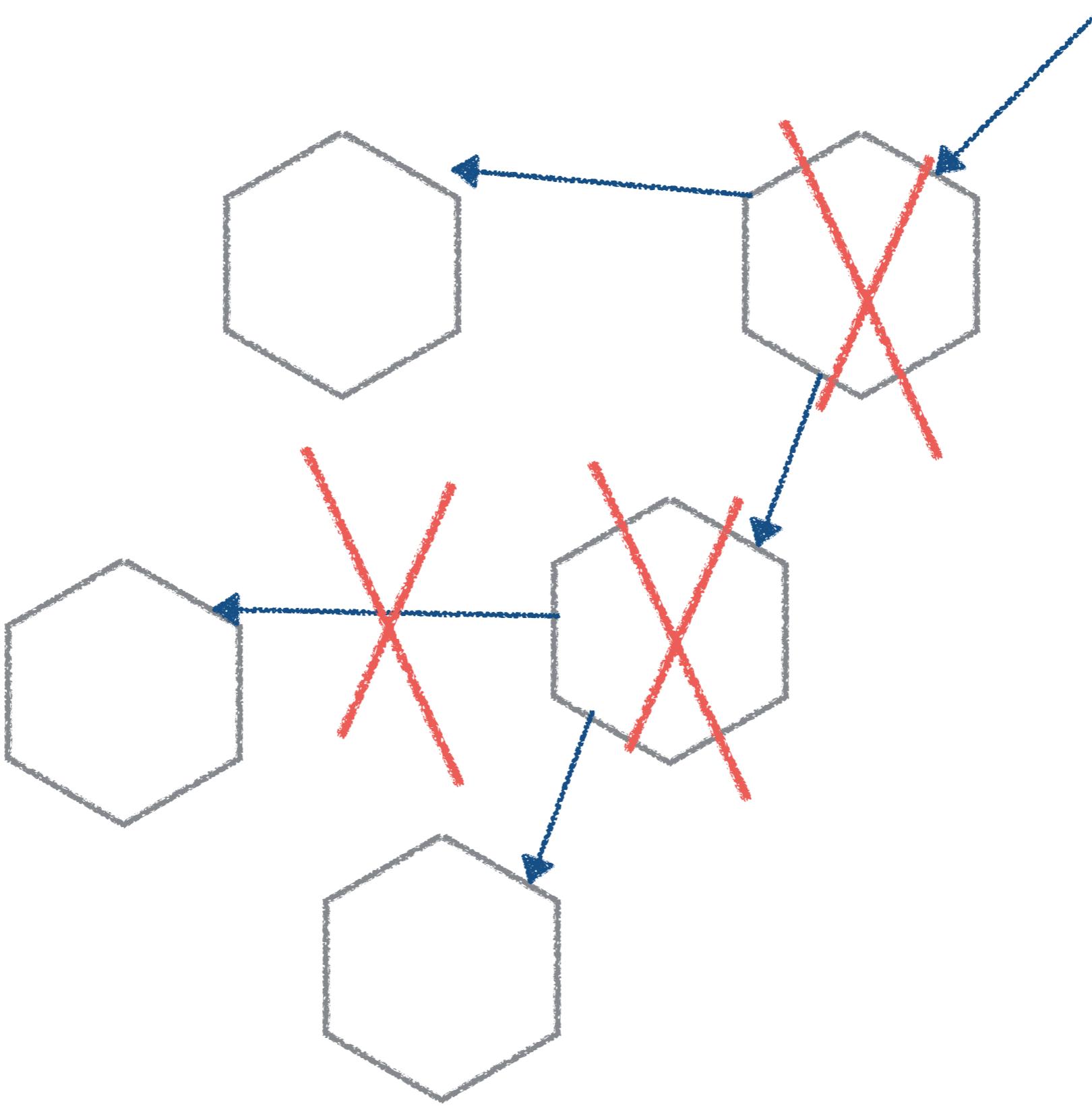
@javazone

@samnewman



@javazone

@samnewman

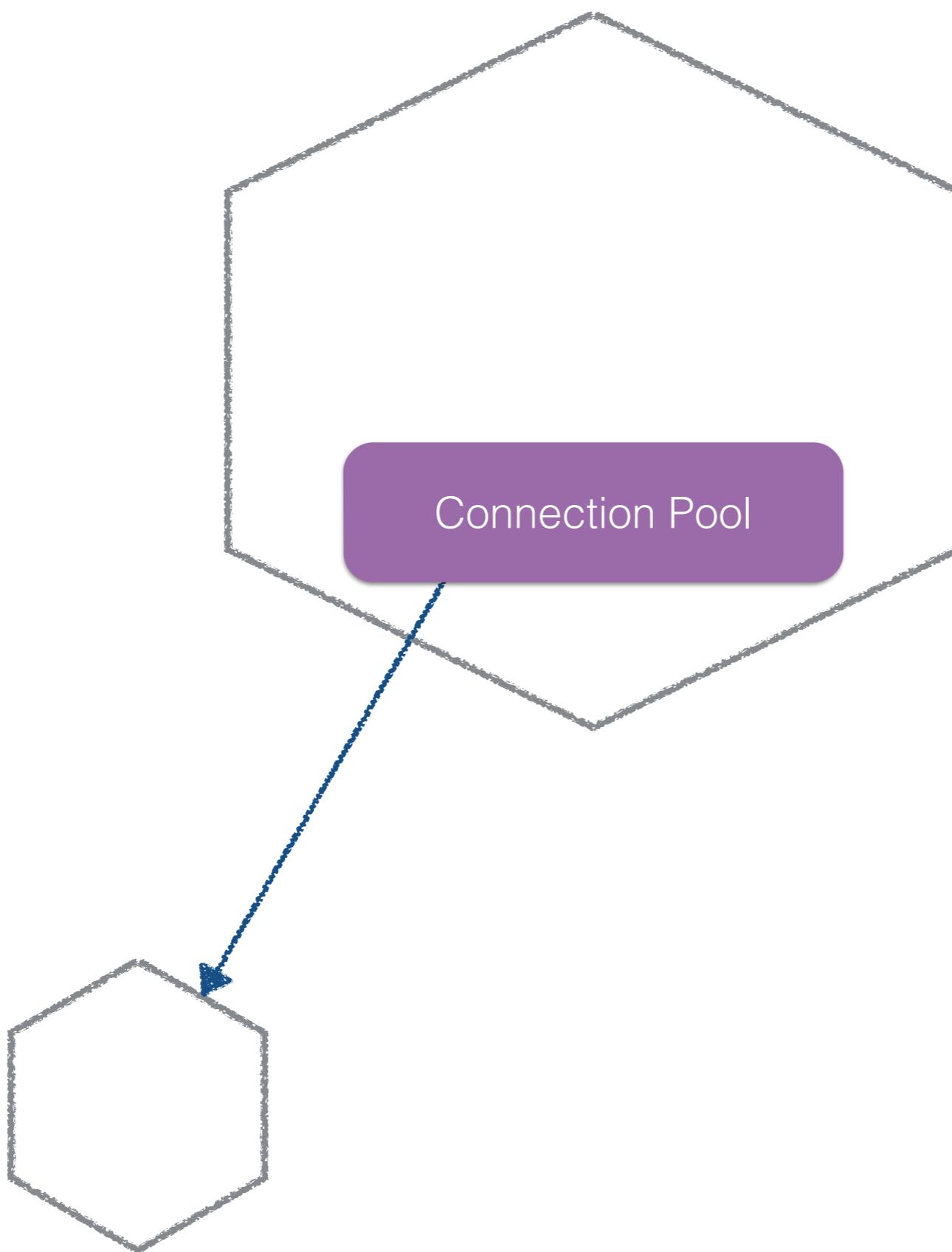


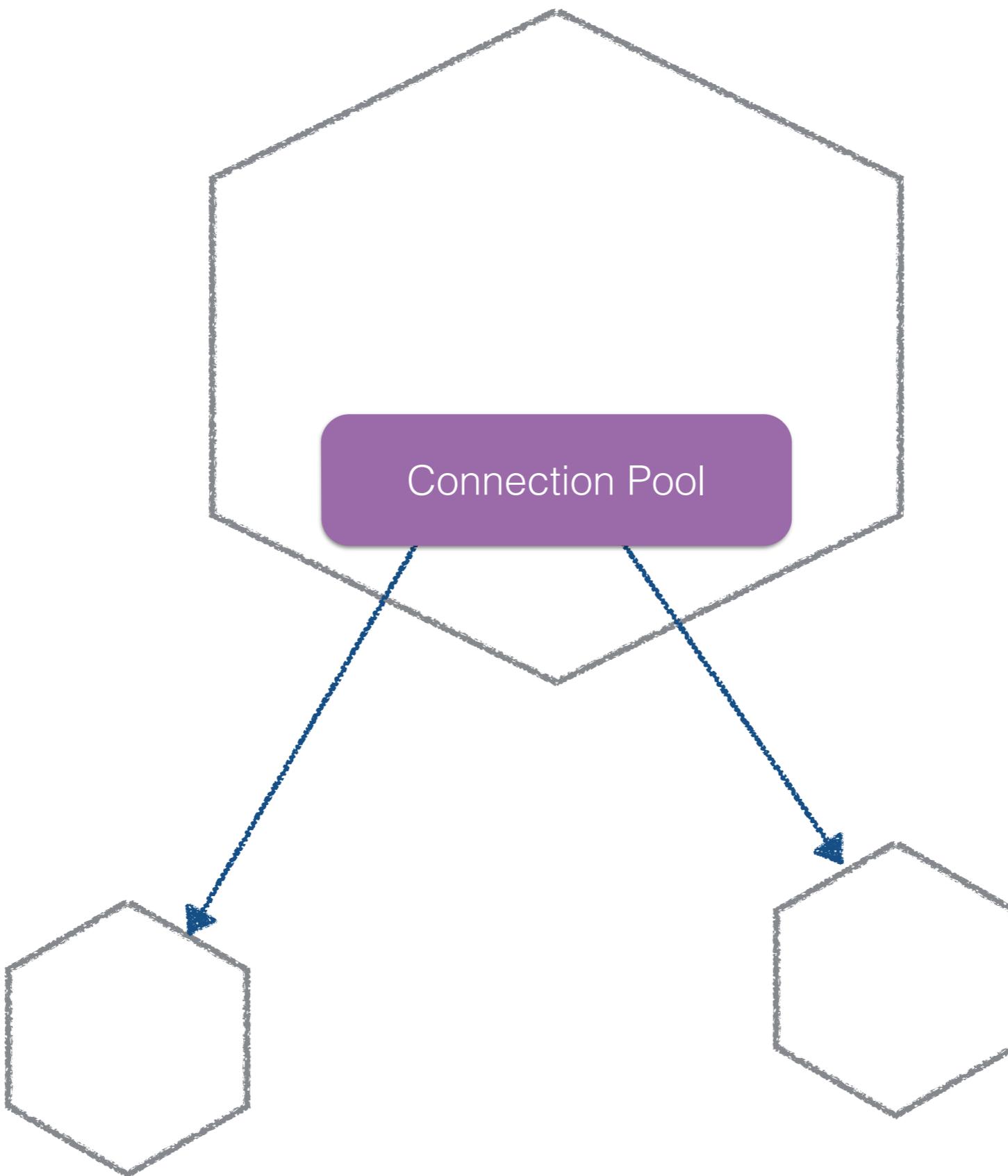
@javazone

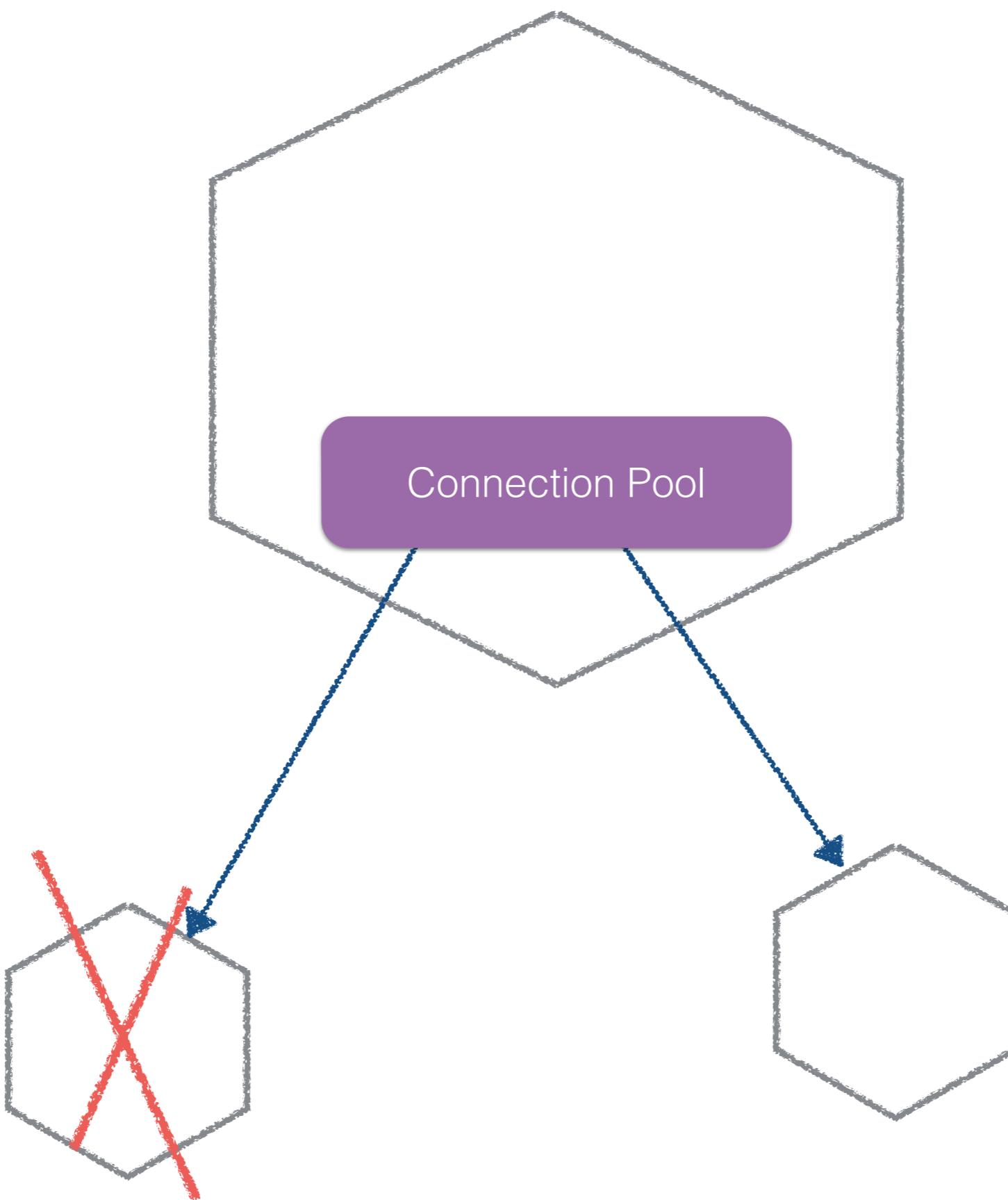
@samnewman

<http://www.flickr.com/photos/louish/5611657857/>



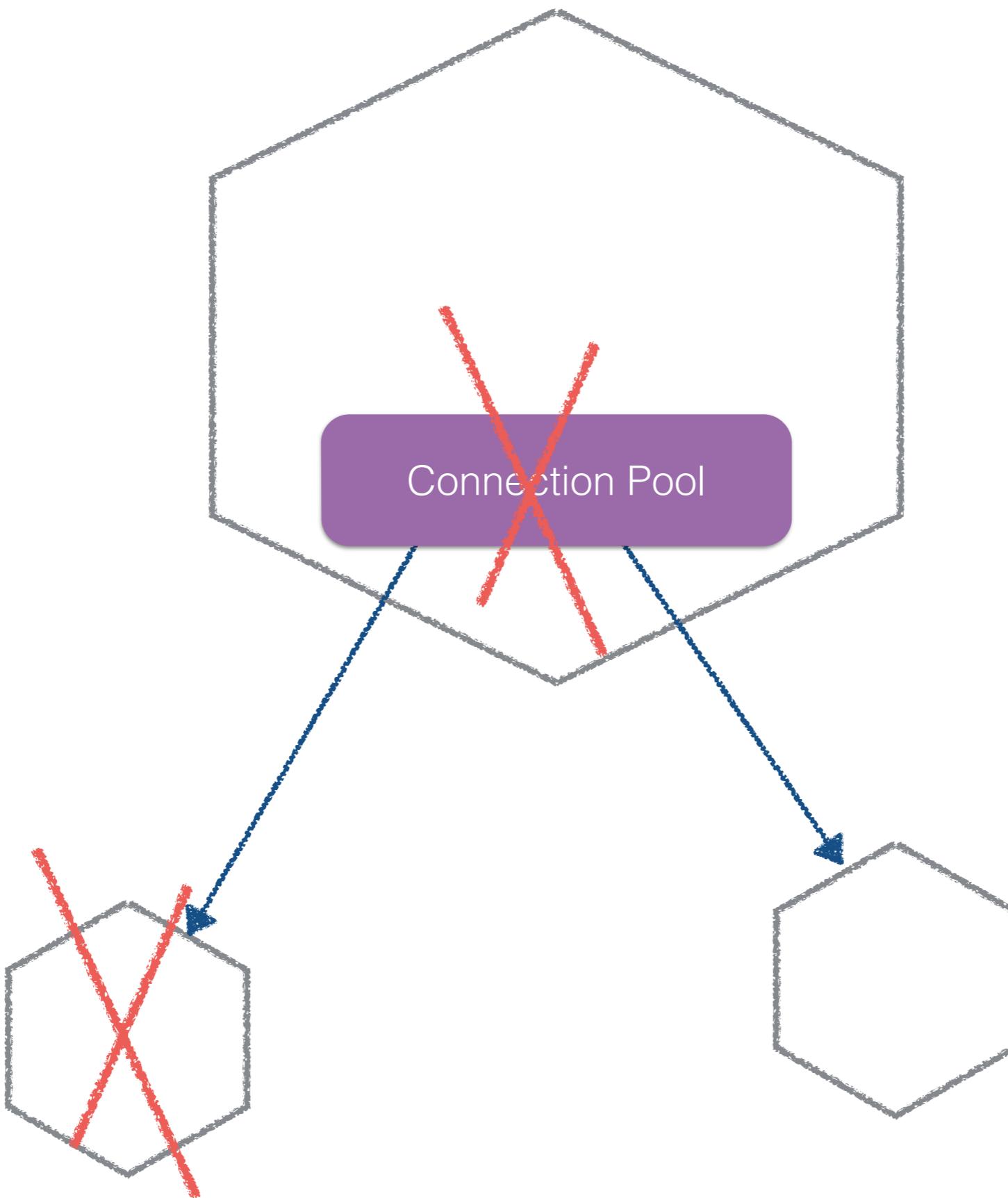






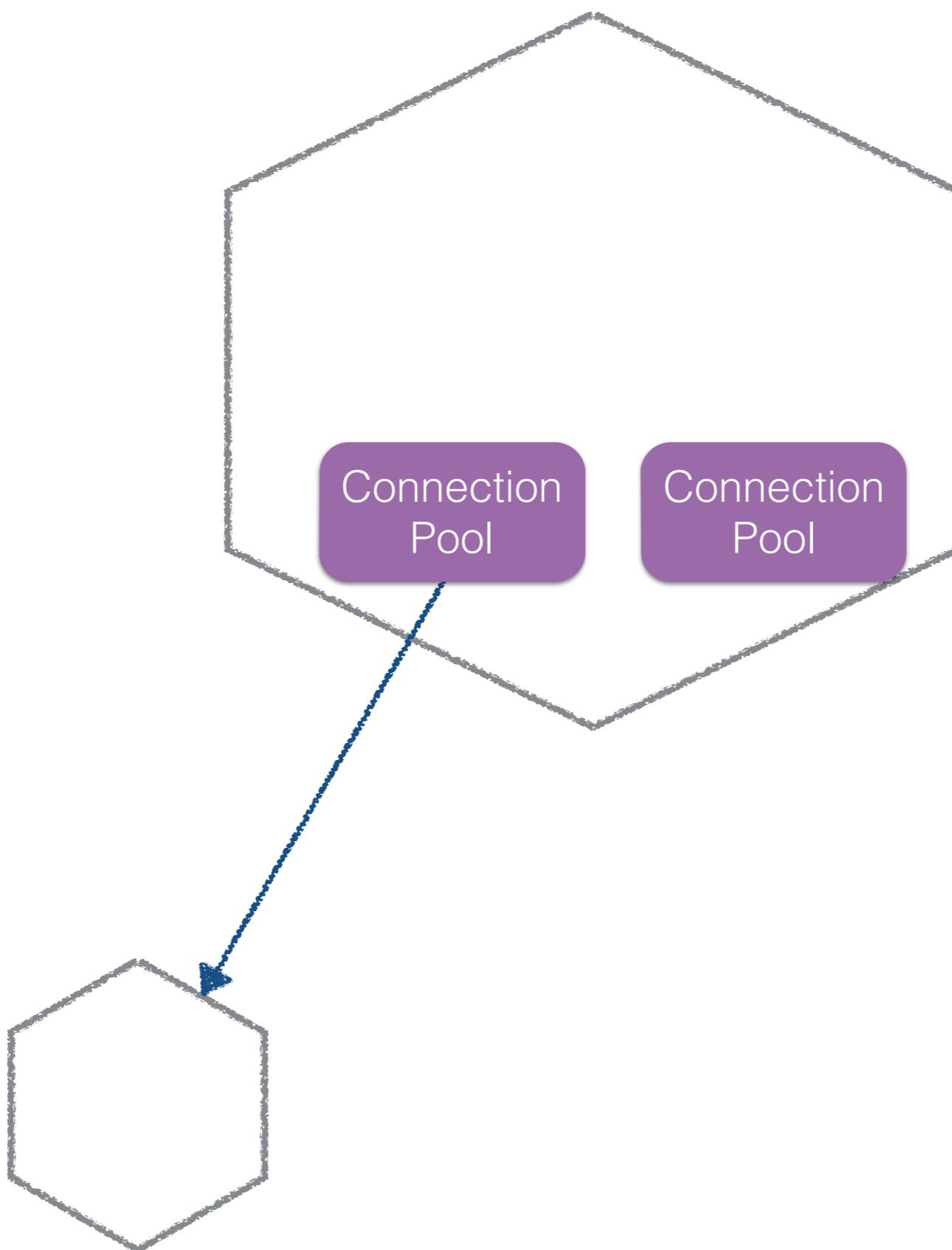
@javazone

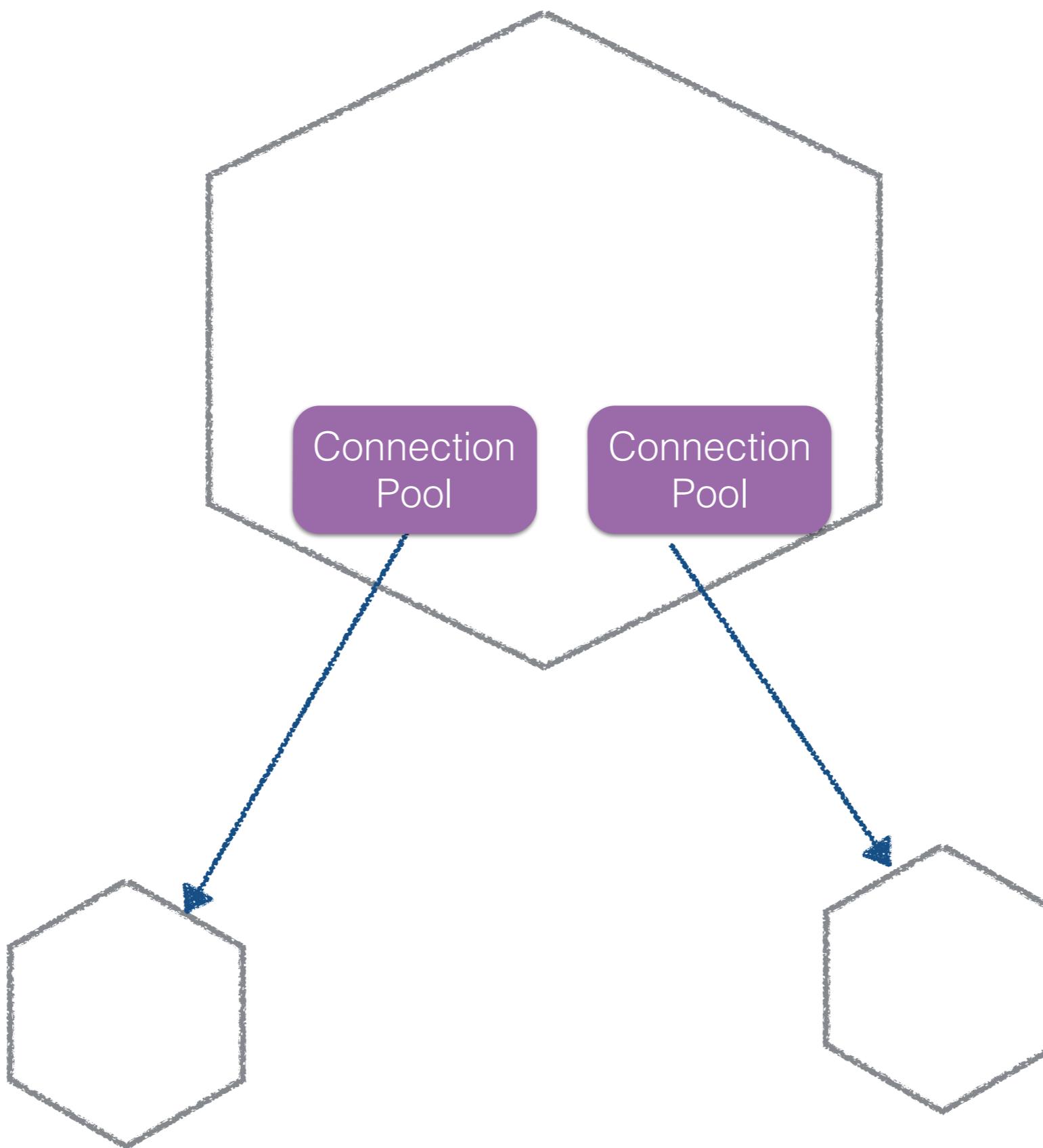
@samnewman

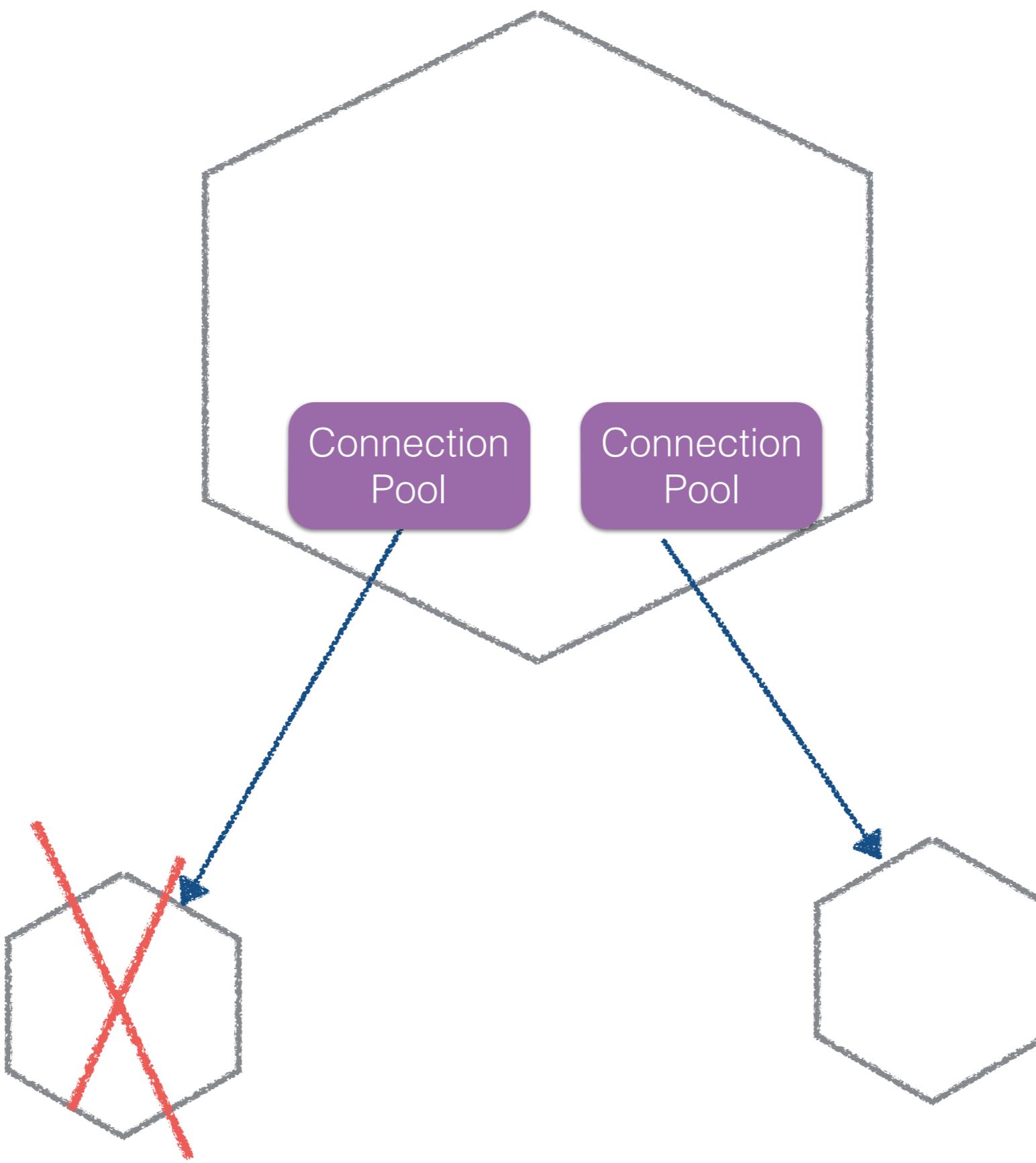


@javazone

@samnewman

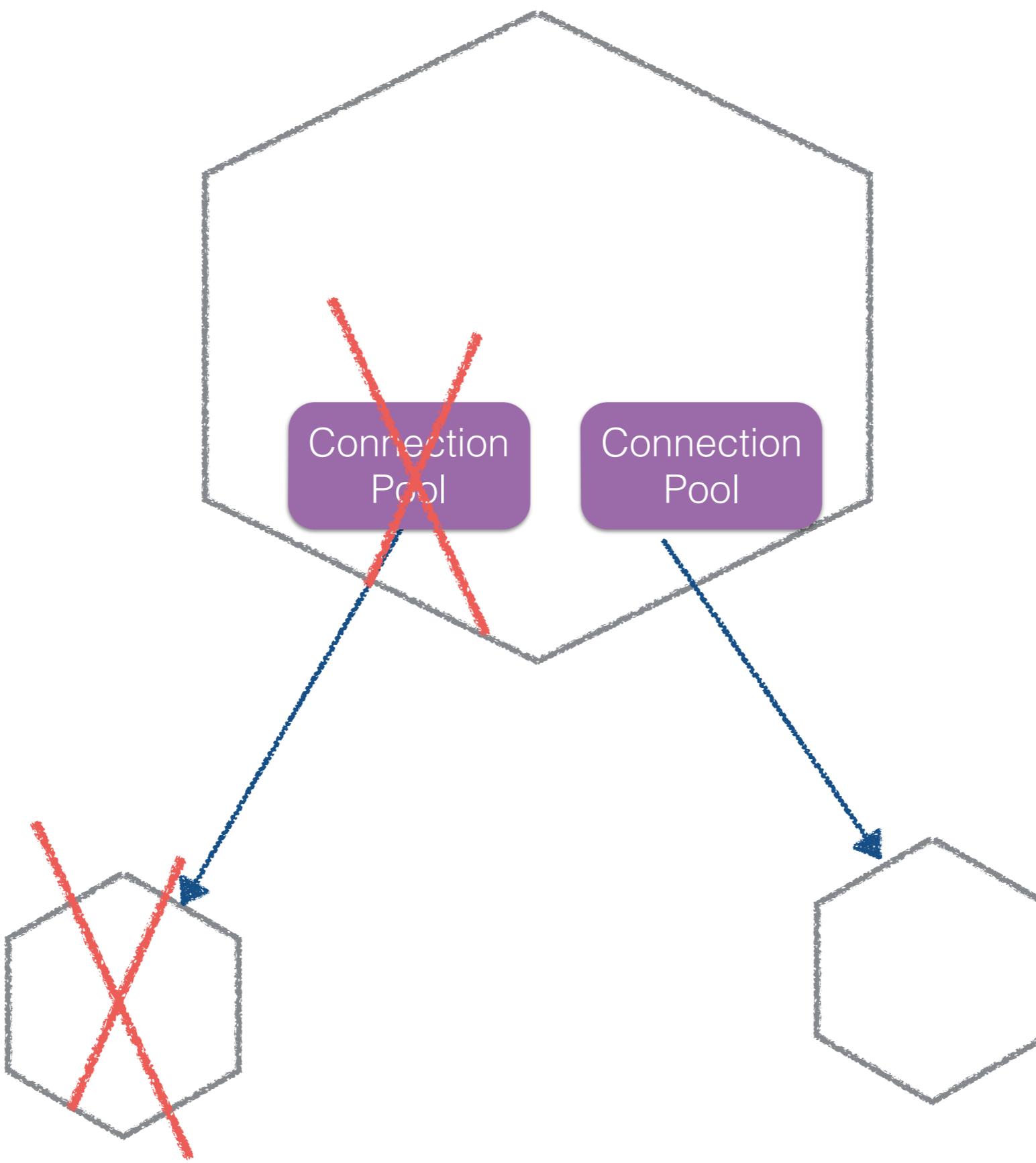






@javazone

@samnewman



@javazone

@samnewman



HYSTRIX
DEFEND YOUR APP



HYSTRIX

DEFEND YOUR APP



2,911
Downloads

1,542
Downloads of v 2.0.0

2014-03-13
Last published

Polly 2.0.0

Polly is a .NET 3.5 / 4.0 / 4.5 / PCL library library that allows developers to express transient exception handling policies such as Retry, Retry Forever, Wait and Retry or Circuit Breaker in a fluent manner.

To install Polly, run the following command in the [Package Manager Console](#)

```
PM> Install-Package Polly
```

[Tweet](#)

[Release Notes](#)

The
Pragmatic
Programmers

Release It!

Design and Deploy
Production-Ready Software



Michael T. Nygard

@javazone

@samnewman

The
Pragmatic
Programmers

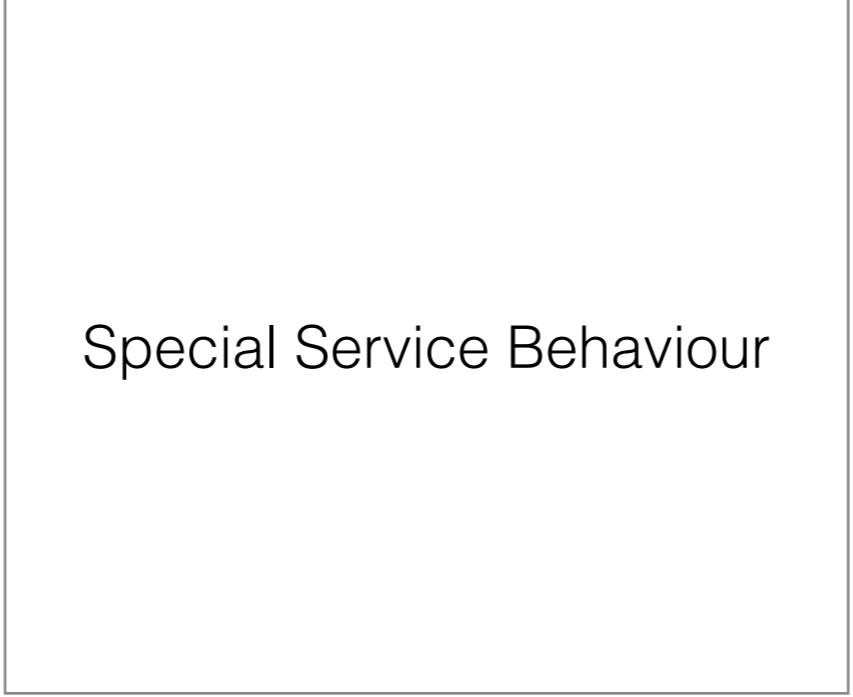
Release It!

Design and Deploy
Production-Ready Software

TIP 12 / 14: Use timeouts, circuit breakers and bulkheads to avoid cascading failure



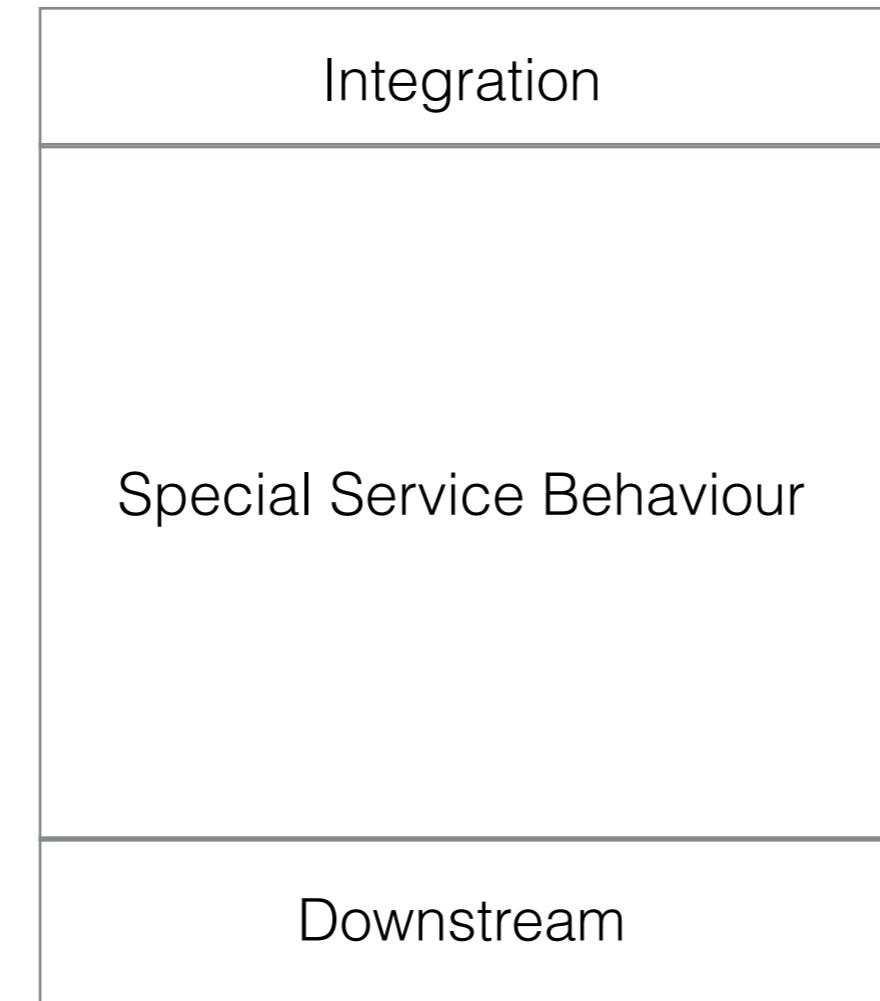
Michael T. Nygard

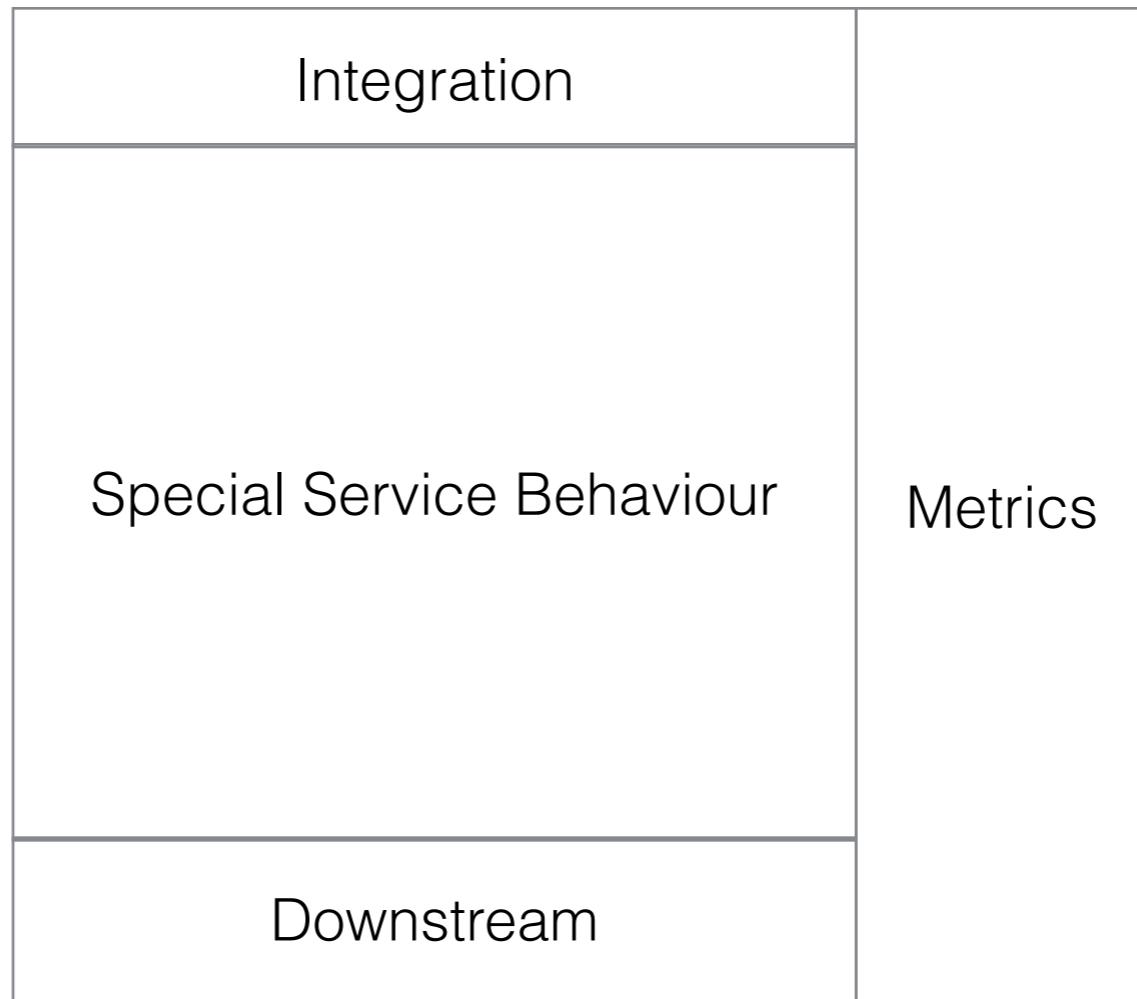


Special Service Behaviour

Integration

Special Service Behaviour





	Integration	
	Special Service Behaviour	Metrics
	Downstream	

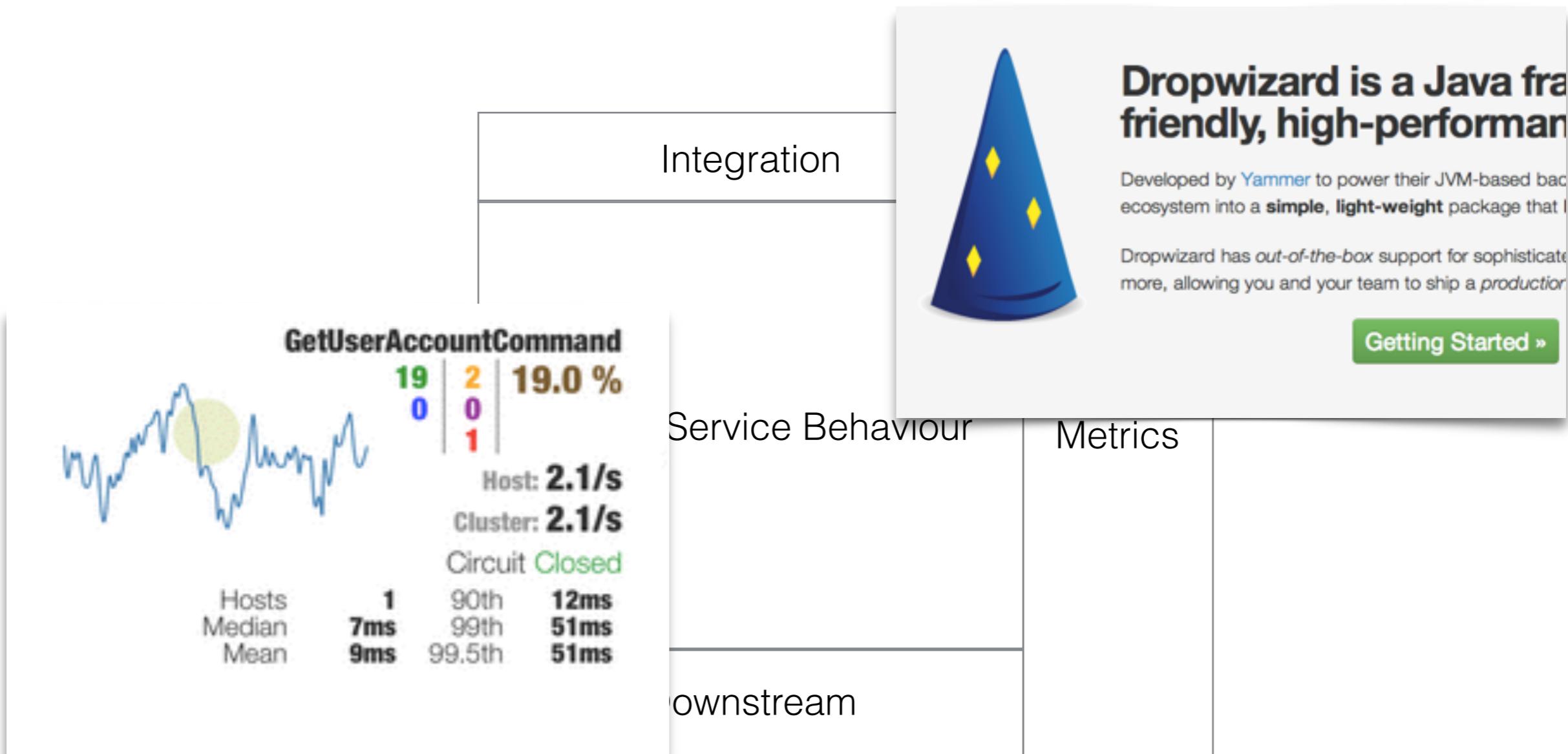


Dropwizard is a Java framework for building robust, friendly, high-performance web services.

Developed by [Yammer](#) to power their JVM-based back-end ecosystem into a **simple, light-weight** package that is easy to use and extend.

Dropwizard has *out-of-the-box* support for sophisticated features like metrics, configuration, and health checks, allowing you and your team to ship a production-ready service quickly.

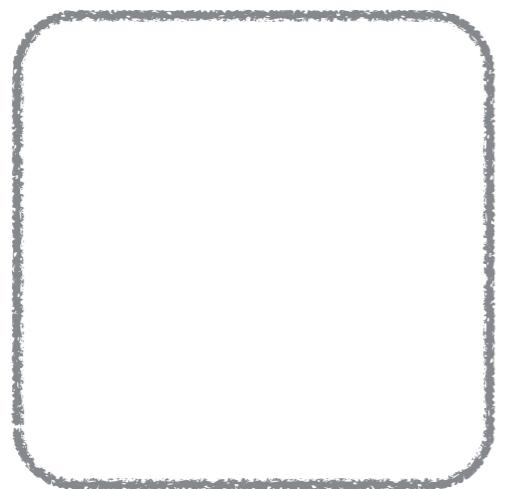
[Getting Started »](#)



The screenshot shows a service monitoring interface. At the top, there's a navigation bar with tabs for 'Integration' (selected), 'Deployment', 'Logs', and 'Metrics'. Below the navigation, a section titled 'GetUserAccountCommand' displays performance metrics: 19 (green), 2 (orange), and 19.0 % (brown). A blue progress bar indicates 19.0%. To the right, there's a large blue wizard hat icon with three yellow stars. The main area contains a yellow box with the text: 'TIP 13 / 14: Consider Service Templates to make it easy to do the right thing!'. Below this, there's a table with performance data:

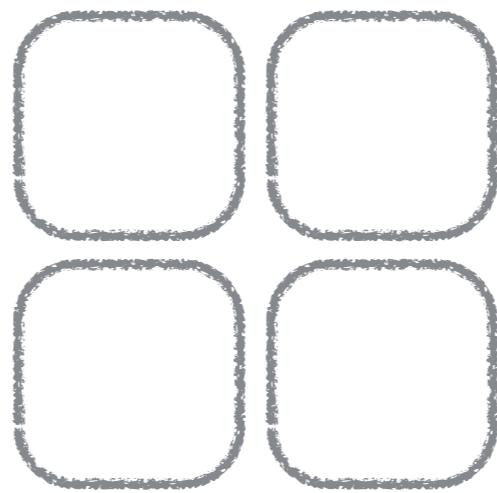
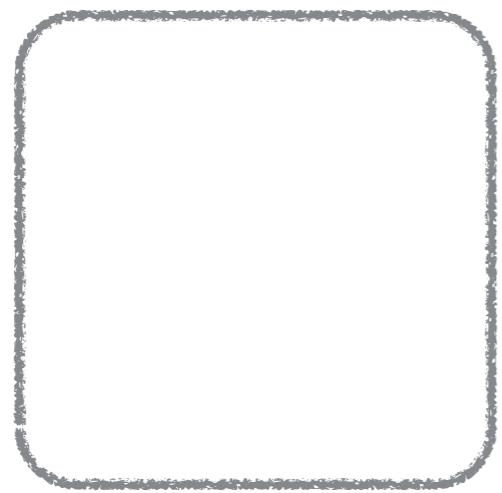
	Hosts	1	90th	12ms
Median	7ms		99th	51ms
Mean	9ms		99.5th	51ms

On the right side of the dashboard, there's a sidebar with sections for 'Service Behaviour', 'Metrics', and 'Logs'. A green button labeled 'Getting Started »' is located in the top right corner of the main content area.



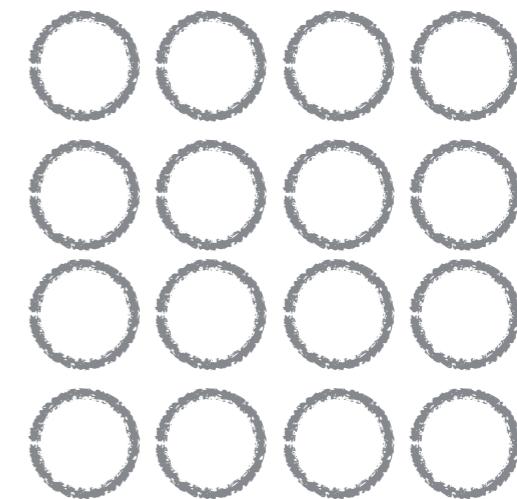
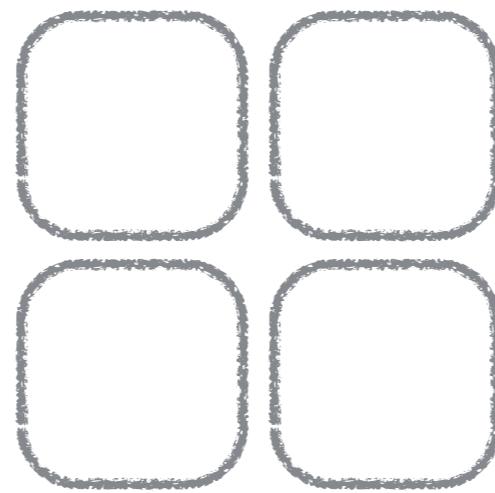
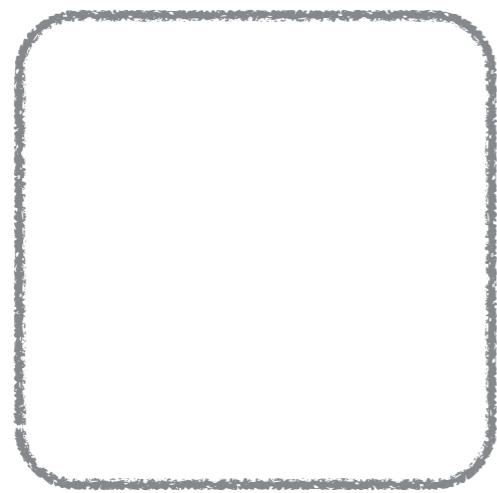
@javazone

@samnewman



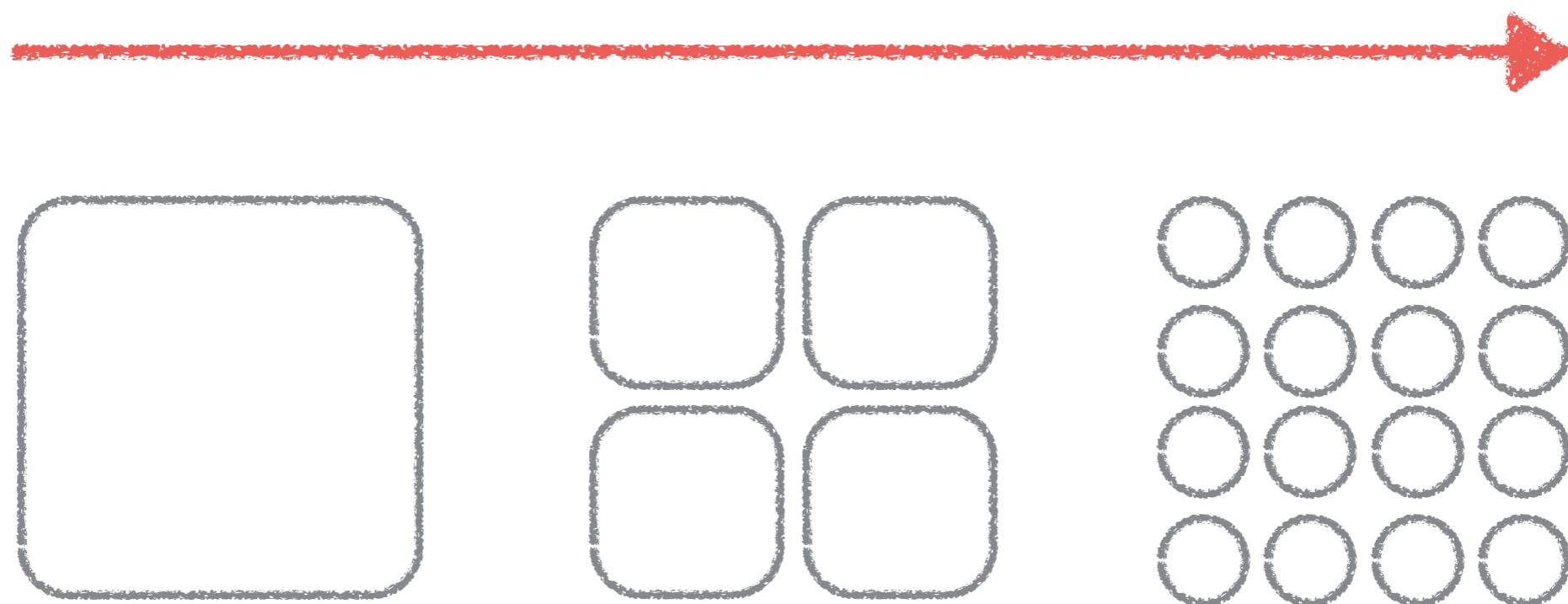
@javazone

@samnewman



@javazone

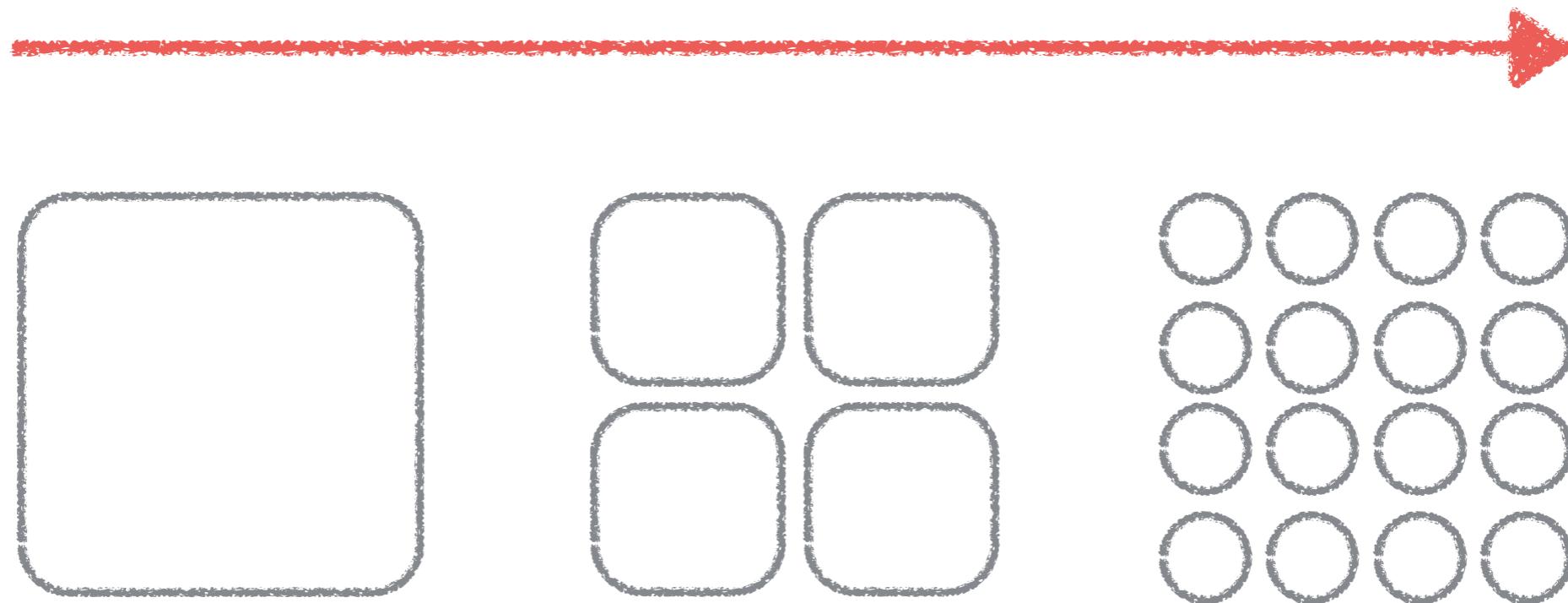
@samnewman



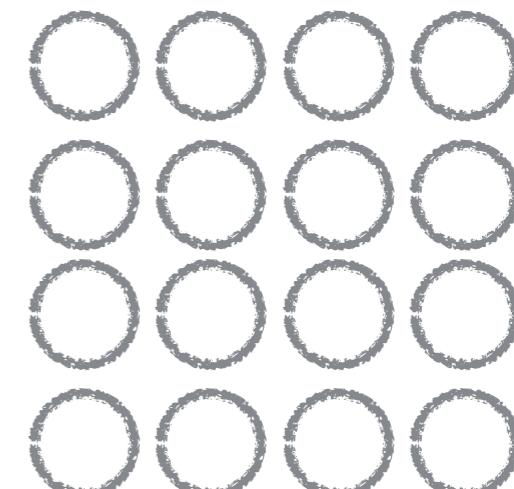
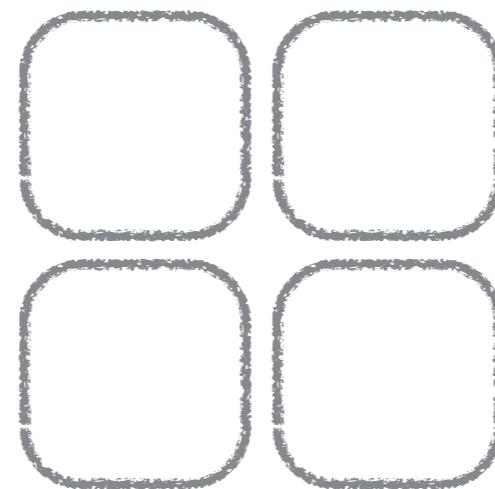
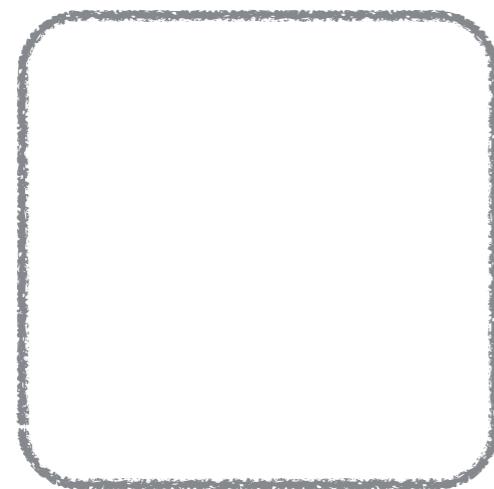
@javazone

@samnewman

Choice Opportunity

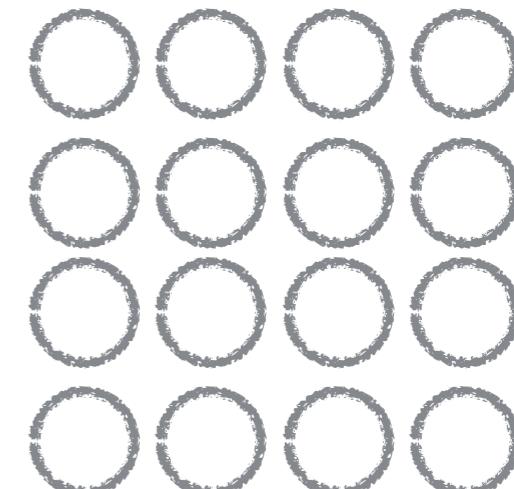
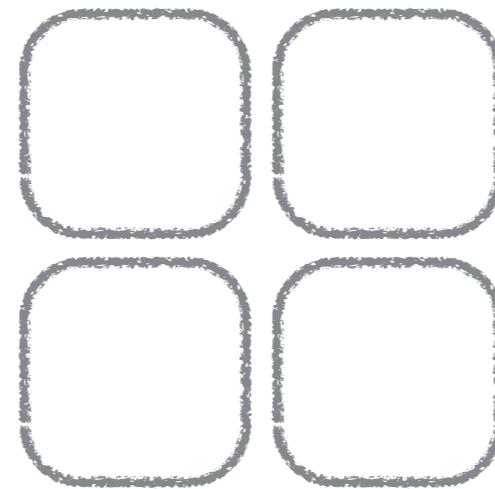
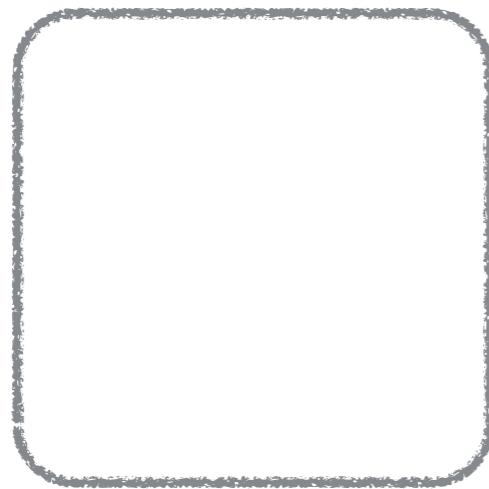


Choice Opportunity



Complexity

Choice Opportunity



Monitoring

Complexity

Safety

Deployment

Integration

Choice Opportunity



Monitoring

Complexity

Safety

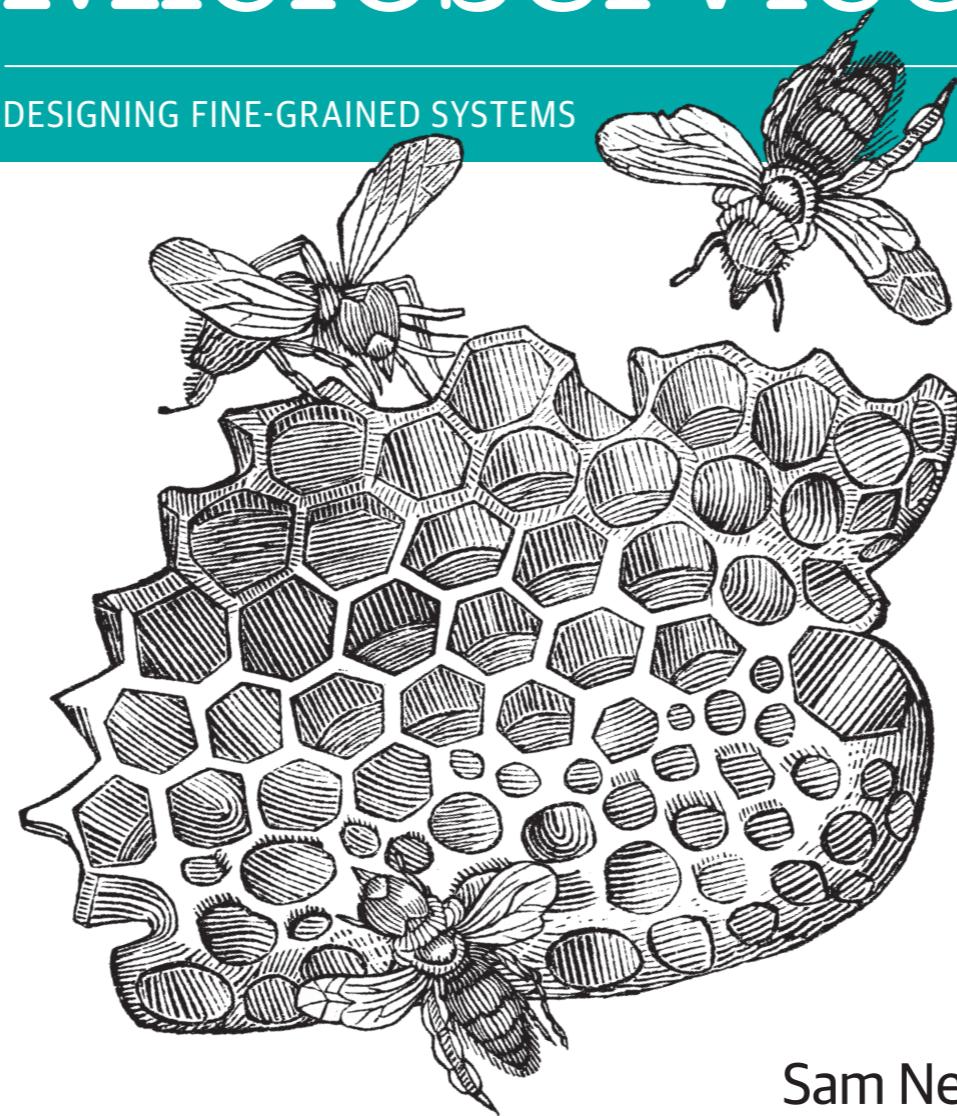
Deployment

Integration

O'REILLY®

Building Microservices

DESIGNING FINE-GRAINED SYSTEMS



Sam Newman

Designing For Rapid Release

Coverage of this session

VIDEO [Designing For Rapid Release - Sam Newman @ Goto Aarhus 2012 \(youtube.com\)](#)

★ Favourite this! Added 11 minutes ago

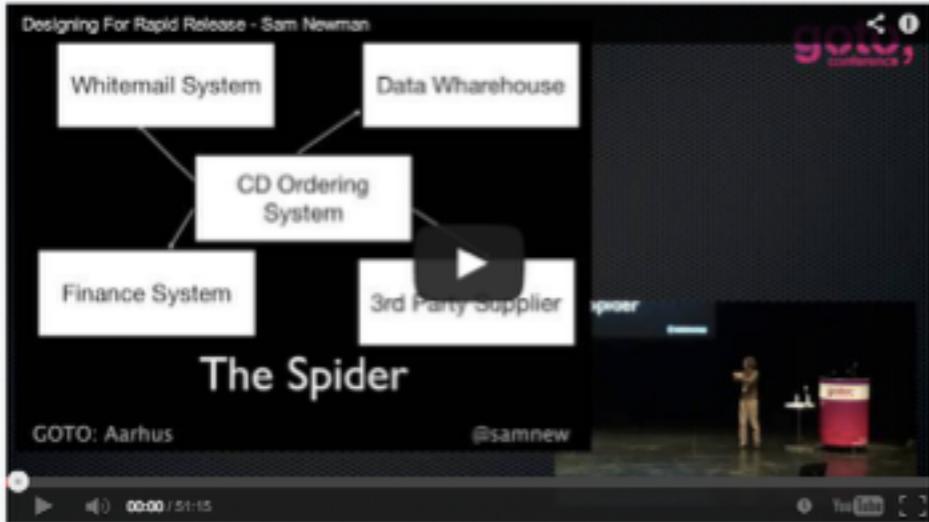
The screenshot shows a video player interface. At the top, it says 'Coverage of this session' and 'VIDEO'. Below that is a link to a YouTube video titled 'Designing For Rapid Release - Sam Newman @ Goto Aarhus 2012 (youtube.com)'. Underneath the link is a small note: '★ Favourite this! Added 11 minutes ago'. The main part of the screenshot is a diagram titled 'The Spider'. It consists of several boxes connected by arrows: 'Whitemail System' and 'Data Warehouse' both point to 'CD Ordering System'; 'Finance System' and '3rd Party Supplier' also point to 'CD Ordering System'. The 'CD Ordering System' box is labeled 'The Spider' at the bottom. To the right of the diagram is a video thumbnail showing a man standing on a stage with a red podium. The thumbnail has a play button in the center. The video player interface includes a progress bar at the bottom left showing '00:00 / 51:15' and a YouTube logo at the bottom right.

500 West Madison
500 West Madison

Designing For Rapid Release

Coverage of this session

VIDEO
[Designing For Rapid Release - Sam Newman @ Goto Aarhus 2012 \(youtube.com\)](#)
★ Favourite this! Added 11 minutes ago



The Spider

GOTO: Aarhus

00:00 / 51:15

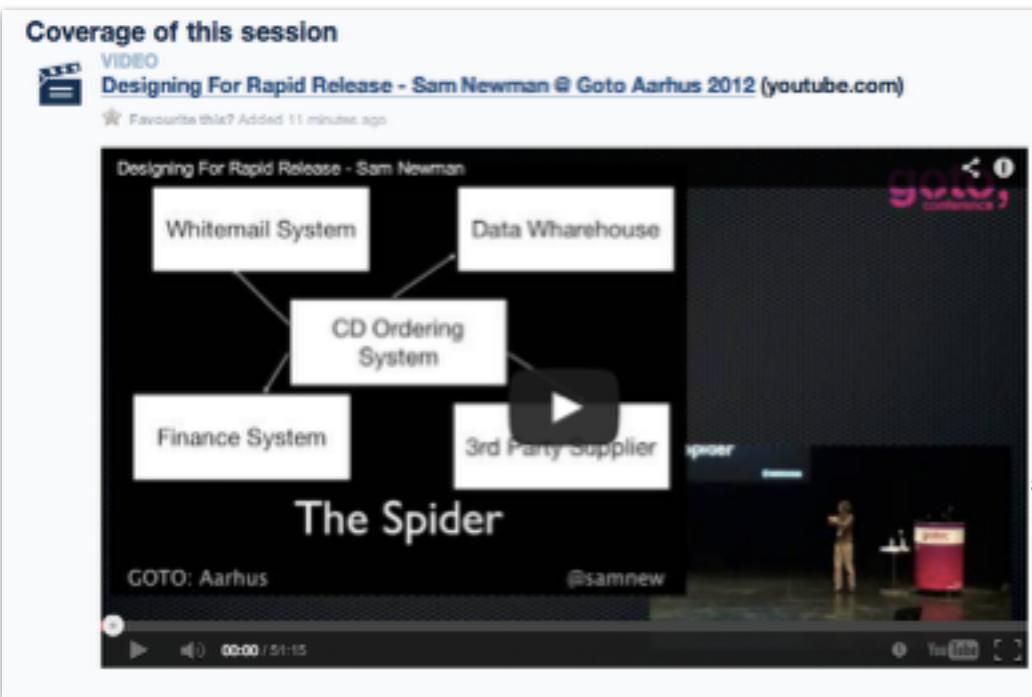


From Macro To Micro

@javazone

@samnewman

Designing For Rapid Release



From Macro To Micro

<http://lanyrd.com/profile/samnewman/>

@javazone

@samnewman

THANKS!

*Any questions:
@samnewman
snewman@thoughtworks.com*

ThoughtWorks®