

# MONOLITHS TO MICROSERVICES

Sam Newman

Day Two

# Overview

- Questions from day 1 & exercises
- Service collaboration options
- Build & Deployment
- Monitoring

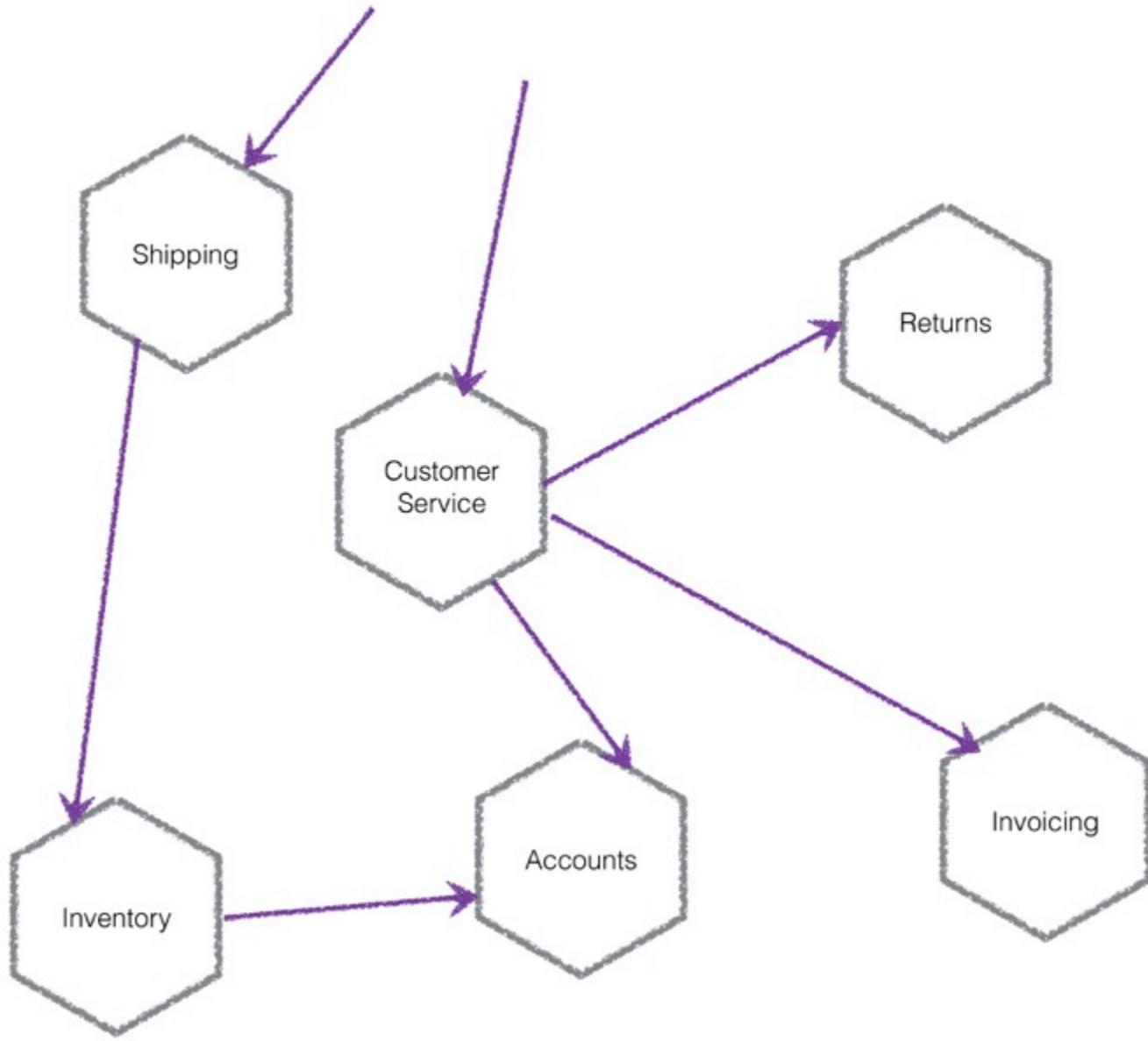
# **MONOLITHS**

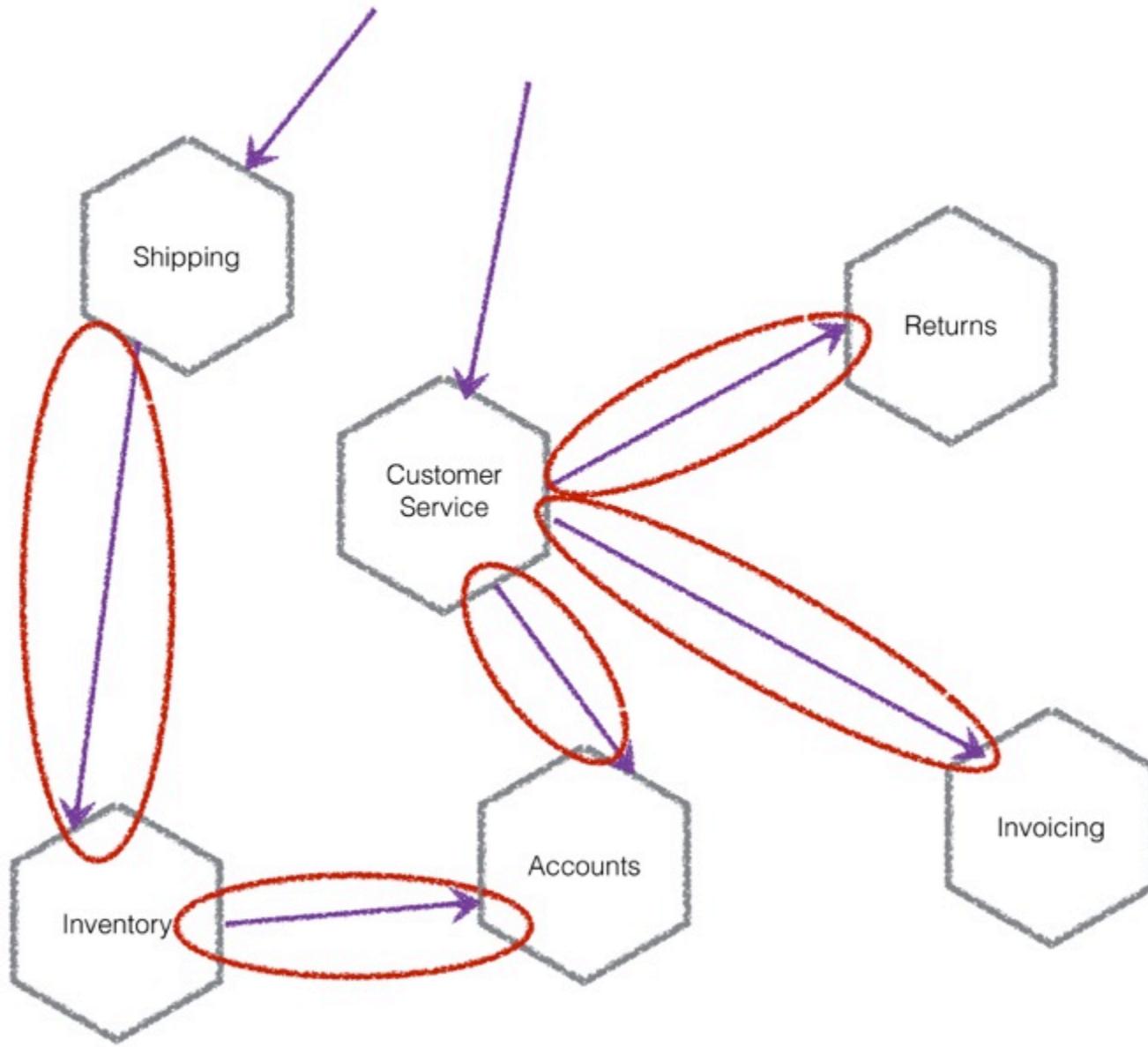
## **TO MICROSERVICES**

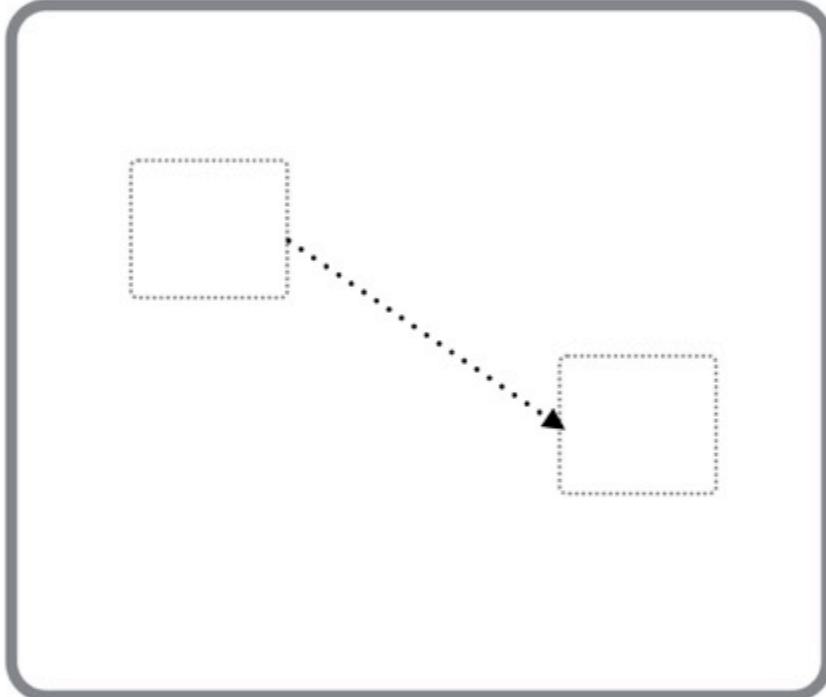
Sam Newman

## Service Interactions



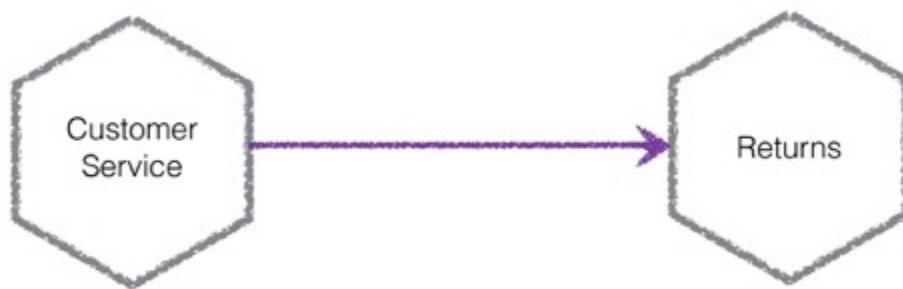




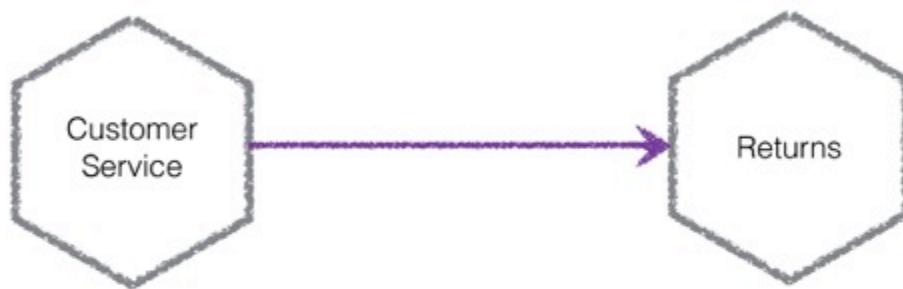


Cost of change  
is low

Easy to reason  
about



Changing a call ?



Changing a call ?  
potential API breakage



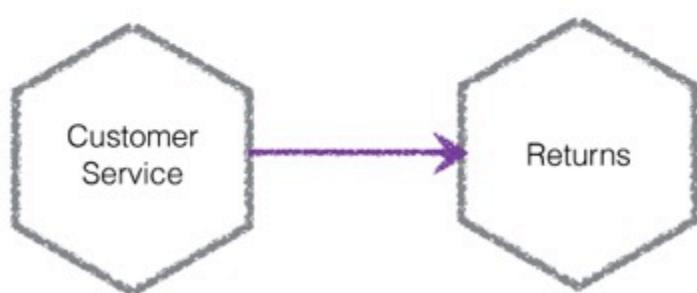
Changing a call ?  
potential API breakage  
two deployments to rollout a change

Are calls between services like  
calls inside a process boundary?

## PERFORMANCE IMPLICATIONS



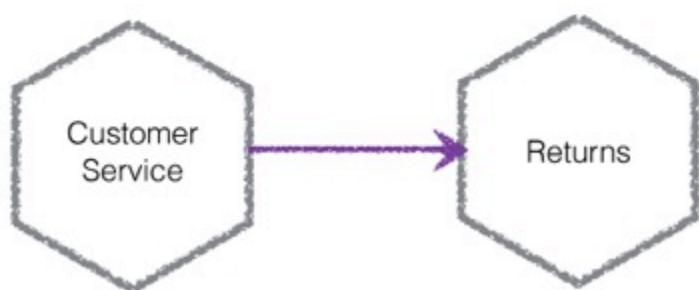
---



## PERFORMANCE IMPLICATIONS



Per-call overhead is very low

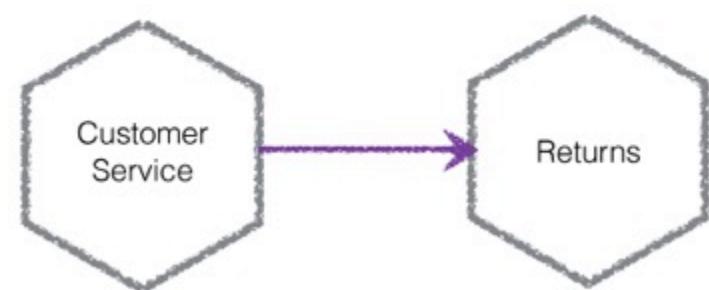


## PERFORMANCE IMPLICATIONS



Per-call overhead is very low

Movement of data by reference

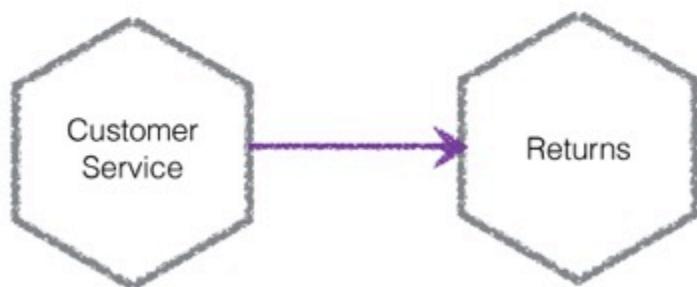


## PERFORMANCE IMPLICATIONS



Per-call overhead is very low

Movement of data by reference



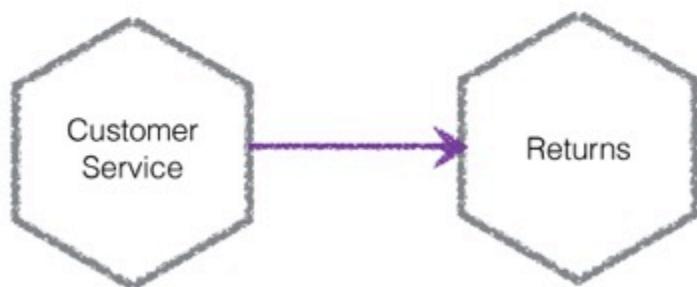
Call overhead can be very high

## PERFORMANCE IMPLICATIONS



Per-call overhead is very low

Movement of data by reference



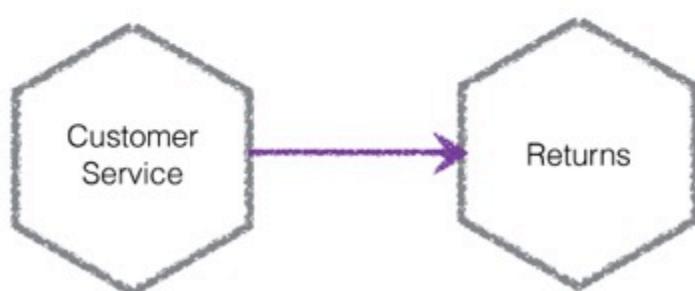
Call overhead can be very high

Data moved by marshalling, or handing off to an external store

## HANDLING ERRORS



---

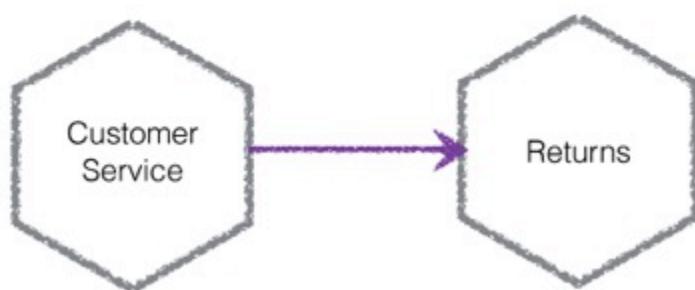


## HANDLING ERRORS



Errors straightforward

---

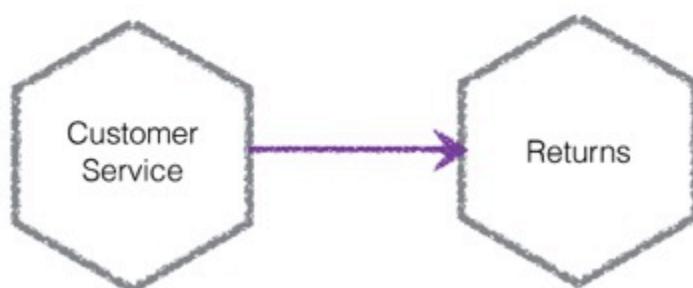


## HANDLING ERRORS



Errors straightforward

---

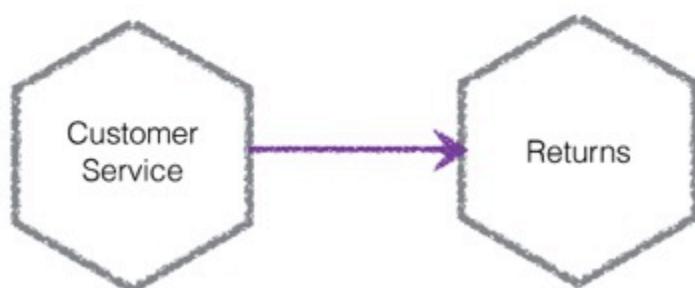


Timeouts

## HANDLING ERRORS



Errors straightforward



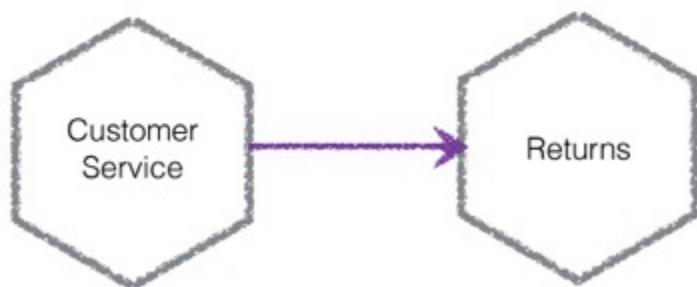
Timeouts

Downstream outage

## HANDLING ERRORS



Errors straightforward

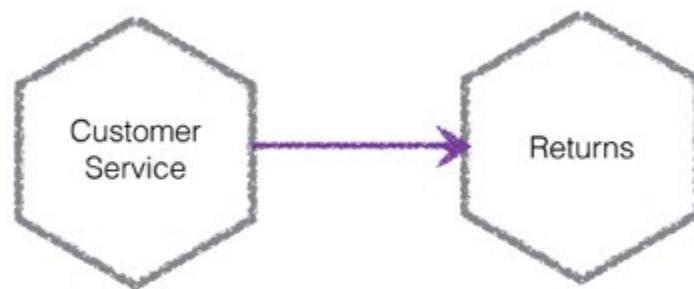


Timeouts

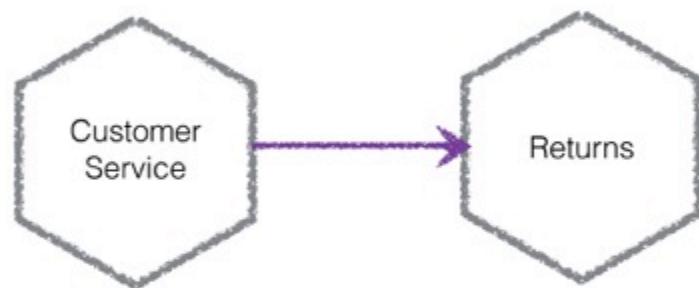
Downstream outage

Difference between client and server errors

## 4XX vs 5XX status codes

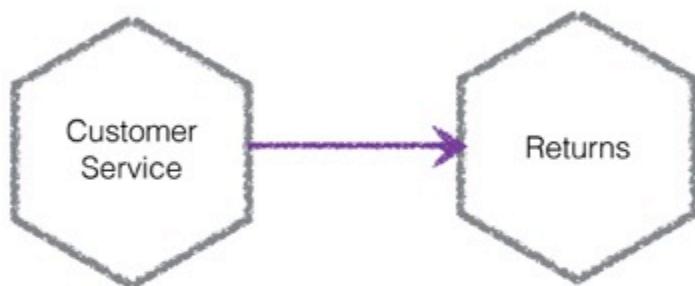


## 4XX vs 5XX status codes



4XX - you did something wrong!

## 4XX vs 5XX status codes



4XX - you did something wrong!

5XX - there is something wrong at my end...

[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)

<b>401 Client Error</b>	[401]
The user does not have a credential for authentication which that client needs to have access. It's used when responding to a <b>401 Unauthorized</b> message.	
<b>401 Bad Request</b>	
The server cannot or will not process the request due to an apparent client error (e.g., malformed request syntax, too large size, invalid request message).	
<b>401 Unauthorized</b>	[HTTP 1.0/1.1]
Similar to 401 Protection, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include the appropriate credentials.	
<b>401 Client Error</b>	[HTTP 401]
The client does not have the right to access the resource because it is unauthorized. This can be caused by supplying insufficient or incorrect credentials. This status code is different from 403 Forbidden, because 403 Forbidden is applied to users who are authenticated but do not have specific permissions to access the resource.	
<b>402 Payment Required</b>	
Reserved for future use. The original intention was that this code might be used as part of some form of digital cash or micropayment scheme, but that has never happened.	
<b>403 Forbidden</b>	
The request was valid, but the server is refusing to respond to it. The user might be logged in but does not have the necessary permissions for.	
<b>403 Not Found</b>	
The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.	
<b>403 Client Error</b>	[HTTP 403]
An request method is not supported for the requested resource, for example, a <b>DELETE</b> request on a form which requires data to be presented via <b>POST</b> , or a <b>PUT</b> .	
<b>403 Not Acceptable</b>	
The requested resource is capable of generating output not acceptable according to the <b>Accept</b> header sent in the request. See Content negotiation.	
<b>403 Proxy Authentication Required</b>	[HTTP 1.0/1.1]
The client must authenticate itself with the proxy.	
<b>403 Requested Range Not Satisfiable</b>	
The server understood neither the request nor the presentation of the range that the client put within the time that the server was p.	
<b>403 Conflict</b>	
The client PUT method has processed more than one of the resources in the request, or there are multiple conflicts between multiple resources in update.	
<b>403 Gone</b>	
Indicates that the resource requested is no longer available and will not be available again. This should be used when a resource has been intentionally or permanently removed.	
<b>403 Length Required</b>	
The request does not specify the length of its contents, or it is a response on one request contains no content-length header.	
<b>403 Precondition Failed</b>	[HTTP 1.0/1.1]
The server does not meet one of the preconditions that the request put on the request.	
<b>403 Request Entity Too Large</b>	[HTTP 1.0/1.1]
The request is larger than what the server is willing or able to process. Previously called <b>Request Entity Too Large</b> .	
<b>403 URI Too Long</b>	[HTTP 1.0/1.1]
The URI provided was too long for the server to process. Often the result of too much data being encoded as a query string of a <b>GET</b> request, instead of a <b>POST</b> .	
<b>403 Unprocessable Entity</b>	
The request which has a media type which the server or resource does not support. For example, the client uploads an image as <b>image/bmp</b> but the server only supports <b>image/png</b> .	
<b>403 Merge-Via Malformed</b>	[HTTP 1.0/1.1]
The client has asked for a portion of the file (byte serving), but the server cannot supply that portion. For example, if the client asked for a part of the file <b>image/bmp</b> but the server only supports <b>image/png</b> .	
<b>403 Invalid Header</b>	
The server cannot meet the requirements of the <b>Expect</b> request header field.	
<b>403 It's a leap!</b>	[HTTP 1.0/1.1]
This code was defined in 1988 as one of the traditional <b>HTTP Agent Features</b> , in RFC 1341 v. Hyper Text Coffee Pot Control Protocol, and is no longer.	
<b>403 Method-Requires</b>	[HTTP 1.0/1.1]
The request was directed at a server that is not able to produce a response (for example because a connection reused).	
<b>403 UnprocessableEntity</b>	[HTTP 1.0/1.1]
The request was well-formed but was unable to be followed due to semantic errors.	
<b>403 Locked</b>	[HTTP 1.0/1.1]
The resource the client is trying to access is locked.	
<b>403 Invalid Header</b>	[HTTP 1.0/1.1]
The request failed due to failure of a previous request (e.g., a <b>PROPFIND</b> ).	
<b>403 Invalid Request</b>	[HTTP 1.0/1.1]
The client should issue a different protocol such as <b>TCP/1.1</b> , given in the <b>Upgrade</b> header field.	
<b>403 Precondition Required</b>	[HTTP 1.0/1.1]
The origin server requires the request to be conditional. Intended to prevent "the lost update" problem, where a client GETs a resource's state, modifies it, and then POSTs it back.	
<b>403 Too Many Requests</b>	[HTTP 1.0/1.1]
The user has sent too many requests in a given amount of time. Intended for use with rate-limiting schemes.	
<b>403 Request Entity Too Large</b>	[HTTP 1.0/1.1]
The server is unwilling to process the request because either an individual field, or all the header fields collectively, are too large.	
<b>403 Unprocessable Entity</b>	[HTTP 1.0/1.1]
A server understood the request but found it to be invalid for other reasons. It is used for a set of resources that includes the requested resource.	

## 28 4XX Error Codes

[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)

4xx Client Error [400]	
The user views of status codes is intended for situations in which the client source in these errors. Except when responding to a <a href="#">401-409 request</a> , the server should	
<b>400 Bad Request</b>	The server cannot or will not process the request due to an apparent client error (e.g., malformed request syntax, too large size, invalid request message).
<b>401 Unauthorized (RFC 7235.1)</b>	
	Similar to 403 Forbidden, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include the appropriate challenge.
	For example, when an IP address is banned from the website (outside the website domain) and that specific address is refused point.
<b>402 Payment Required</b>	Reserved for future use. The original intention was that this code might be used as part of some form of digital cash or micropayment scheme, but that has never happened.
<b>403 Forbidden</b>	The request was valid, but the server is refusing to respond to it. The user might be logged in but does not have the necessary permissions for the requested resource.
<b>404 Not Found</b>	The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.
	For example, when a user enters a URL that has been misspelled.
	A request method is not supported for the requested resource; for example, a GET request on a form which requires data to be presented via POST, or a PUT request on a file.
<b>405 Method Not Allowed</b>	The requested resource is capable of generating content not acceptable according to the Accept headers sent in the request. See Content negotiation.
<b>407 Proxy Authentication Required (RFC 7235.2)</b>	The client must authenticate itself with the proxy.
<b>408 Request Timeout</b>	The server timedout waiting for the request. According to HTTP specifications, "The client did not produce a request within the time that the server was p".
<b>409 Conflict</b>	The client's request would interfere with the processing of another's request. In other words, there is a conflict between multiple simultaneous updates.
<b>410 Gone</b>	Indicates that the resource requested is no longer available and will not be available again. This should be used when a resource has been intentionally or permanently removed.
<b>411 Length Required</b>	The request did not specify the length of its contents, which is required on one-response messages.
<b>412 Precondition Failed (RFC 7232.1)</b>	The request did not meet the requirements of the preconditions that the request put on the request.
<b>413 Payload Too Large (RFC 7231.4)</b>	The request is larger than the server is willing or able to process. Previously called "Request Entity Too Large".
<b>414 URI Too Long (RFC 7231.4)</b>	The request URI was too long for the server to process. Often the result of too much data being encoded as a querystring of a GET request, methods or URLs.
<b>415 Unsupported Media Type</b>	The request entity has a media type which the server/resource does not support. For example, the client uploads an image as image/png but the server only handles text/html.
<b>416 Range Not Satisfiable (RFC 7232.2)</b>	The client has asked for a portion of the file (byte serving), but the server cannot supply that portion. For example, if the client asked for a part of file #1 from byte 100 to 500, but the file only has 400 bytes.
<b>417 Expectation Failed (RFC 7232.2)</b>	The server cannot meet the requirements of the Expect request header field.
<b>418 I'm a teapot (RFC 2324)</b>	This code was defined in 1998 as one of the traditional ASCII April Fools jokes, in RFC 2324's "Hyper Text Coffee Pot Control Protocol", and is no longer used.
<b>421 Misdirected Request (RFC 7240.3)</b>	The request was directed at a server that is not able to produce a response (for example because a connection reused).
<b>422 Unprocessable Entity (RFC 4918)</b>	The request was well-formed but was unable to be followed due to semantic errors.
<b>423 Locked (RFC 4918)</b>	The resource the client is trying to access is locked.
<b>424 Failed Dependency (RFC 4918)</b>	The request failed due to failure of a previous request (e.g., a PROPFIND or a PRELEAVE).
<b>425 Too Many Requests</b>	The client should wait a different protocol such as 10-15 s, given in the <code>Retry-After</code> header field.
<b>426 Upgrade Required (RFC 6585.3)</b>	The origin server requires the request to be conditional. Intended to prevent "the first capital" problem, where a client GETs a resource's state, modifies it, and then POSTs back to the server.
<b>428 Request Header Fields Too Large (RFC 6585.1)</b>	The user has sent too many requests in a given amount of time. Intended for use with rate-limiting schemes.
<b>429 Too Many Requests (RFC 6585.1)</b>	The server is refusing to process the request because either an individual field, or all the header fields collectively, exceed size.
<b>431 Unprocessable Entity</b>	A server responded with a field element for other resource or a field of resource that includes the requested resource.

## 28 4XX Error Codes

5xx Server Error [500]	
The server failed to fulfill an apparently valid request. [500]	
	Response status codes beginning with the digit "5" indicates cases in which the server is aware that it has encountered a user error. These response codes are applicable to any <a href="#">method</a> .
<b>500 Internal Server Error</b>	
	A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.
<b>501 Not Implemented</b>	The server either does not recognize the request method, or it lacks the ability to fulfill the request. Usually this implies that the server is acting as a gateway or proxy and received an invalid request from the upstream server. [501]
<b>502 Bad Gateway</b>	The server was acting as a gateway or proxy and received an invalid response from the upstream server. [502]
<b>503 ServiceUnavailable</b>	The server is currently unavailable (because it is overloaded or down for maintenance). Generally, this is a temporary condition.
<b>504 Gateway Timeout</b>	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server. [504]
<b>505 HTTP Version Not Supported</b>	The server does not support the HTTP protocol version used in the request. [505]
<b>506 Variant Also Negotiates (RFC 2295)</b>	Transparent content negotiation for the request results in a circular reference. [506]
<b>507 Insufficient Storage (WebDAV / RFC 4913)</b>	The server is unable to store the representation needed to complete the request. [507]
<b>508 Loop Detected (WebDAV / RFC 5842)</b>	The server detected an infinite loop while processing the request (sent in lieu of 208 Already Reported).
<b>510 Not Extended (RFC 2781.4)</b>	Further extensions to the request are required for the server to fulfill it. [510]
<b>511 Network Authentication Required (RFC 6585.1)</b>	The client needs to authenticate to gain network access. Intended for use by intercepting proxies used to control access.

## 11 5XX Error Codes

[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)





418 - I'm A Teapot



A close-up, slightly off-center portrait of Donald Trump's face. He has his eyes closed and a faint smile. A solid white rectangular box is placed over the lower half of his face, covering his nose and mouth area. The background is dark and out of focus.

410 - Gone

Keep it simple

Simple = Synchronous, Request  
Response Communication

# **Synchronous**

**Synchronous**

**Asynchronous**

## **Synchronous**

Block and wait

## **Asynchronous**

## **Synchronous**

Block and wait

## **Asynchronous**

Fire and (maybe) forget

## **Synchronous**

Block and wait

## **Asynchronous**

Fire and (maybe) forget

Simple to reason about

## **Synchronous**

Block and wait

## **Asynchronous**

Fire and (maybe) forget

Simple to reason about

Technology more  
straight forward

## **Synchronous**

Block and wait

Simple to reason about

Technology more  
straight forward

## **Asynchronous**

Fire and (maybe) forget

Great for long-  
running jobs

## **Synchronous**

Block and wait

Simple to reason about

Technology more  
straight forward

## **Asynchronous**

Fire and (maybe) forget

Great for long-  
running jobs

And low latency too!

## **Synchronous**

Block and wait

Simple to reason about

Technology more  
straight forward

## **Asynchronous**

Fire and (maybe) forget

Great for long-  
running jobs

And low latency too!

More complex

## *Collaboration Styles*

## *Collaboration Styles*

### **Request/Response**

## *Collaboration Styles*

**Request/Response**

**Event-based**

## *Collaboration Styles*

### **Request/Response**

Initiate a request, expect  
a response

### **Event-based**

## *Collaboration Styles*

### **Request/Response**

Initiate a request, expect  
a response

### **Event-based**

Things happen,  
things react

## **Request/Response**

**Request/Response**

**Event-based**

**Request/Response**

**Event-based**

**Synchronous**

**Request/Response**

**Event-based**

**Synchronous**

**Asynchronous**

**Request/Response**

**Event-based**

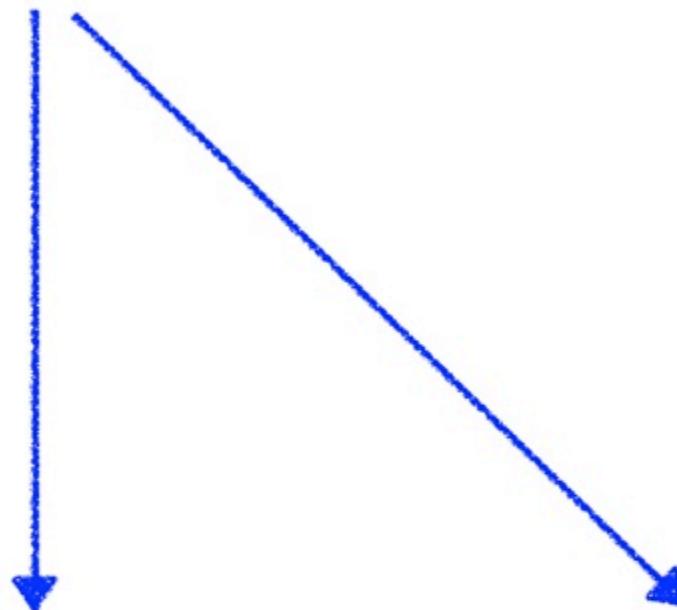


**Synchronous**

**Asynchronous**

**Request/Response**

**Event-based**



**Synchronous**

**Asynchronous**

**Request/Response**

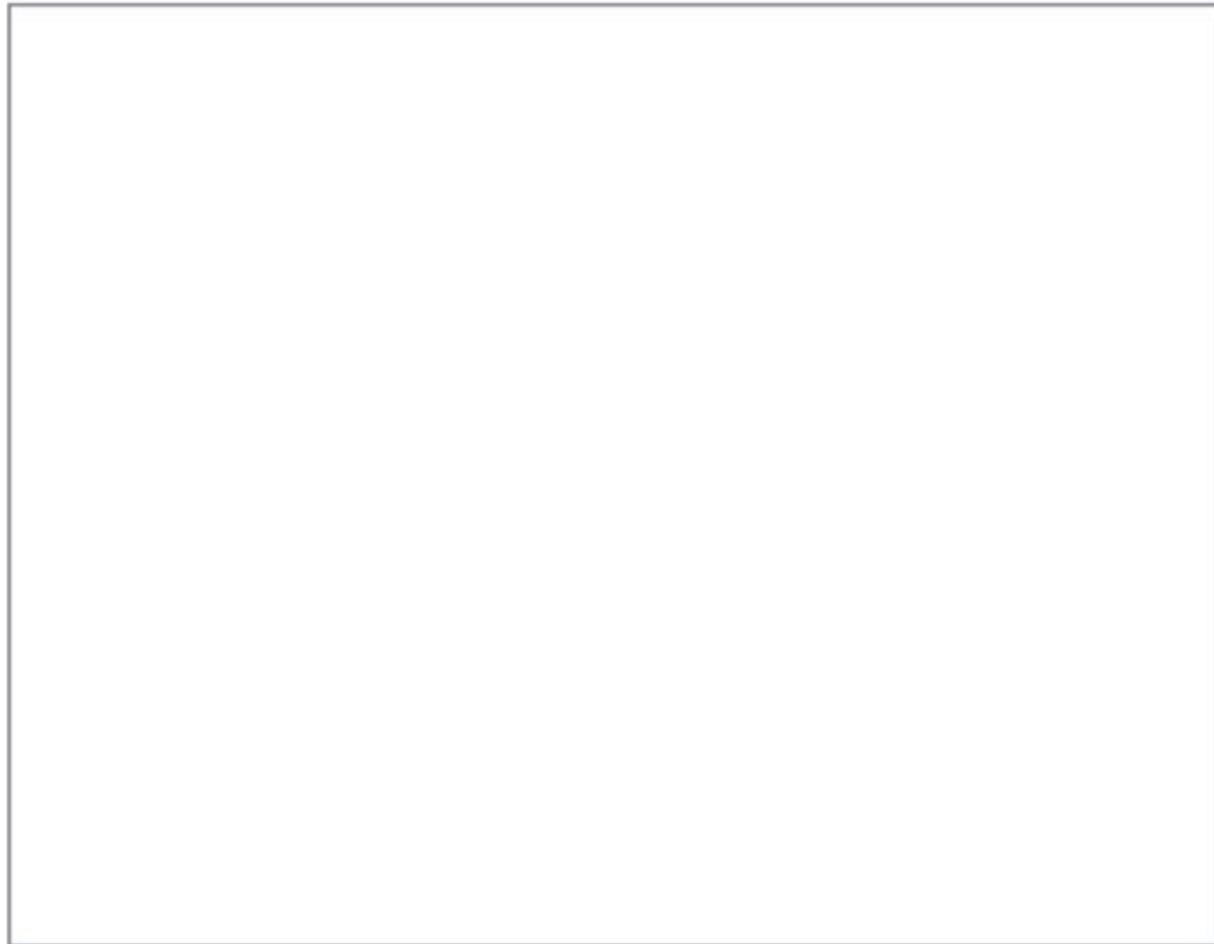


**Synchronous**

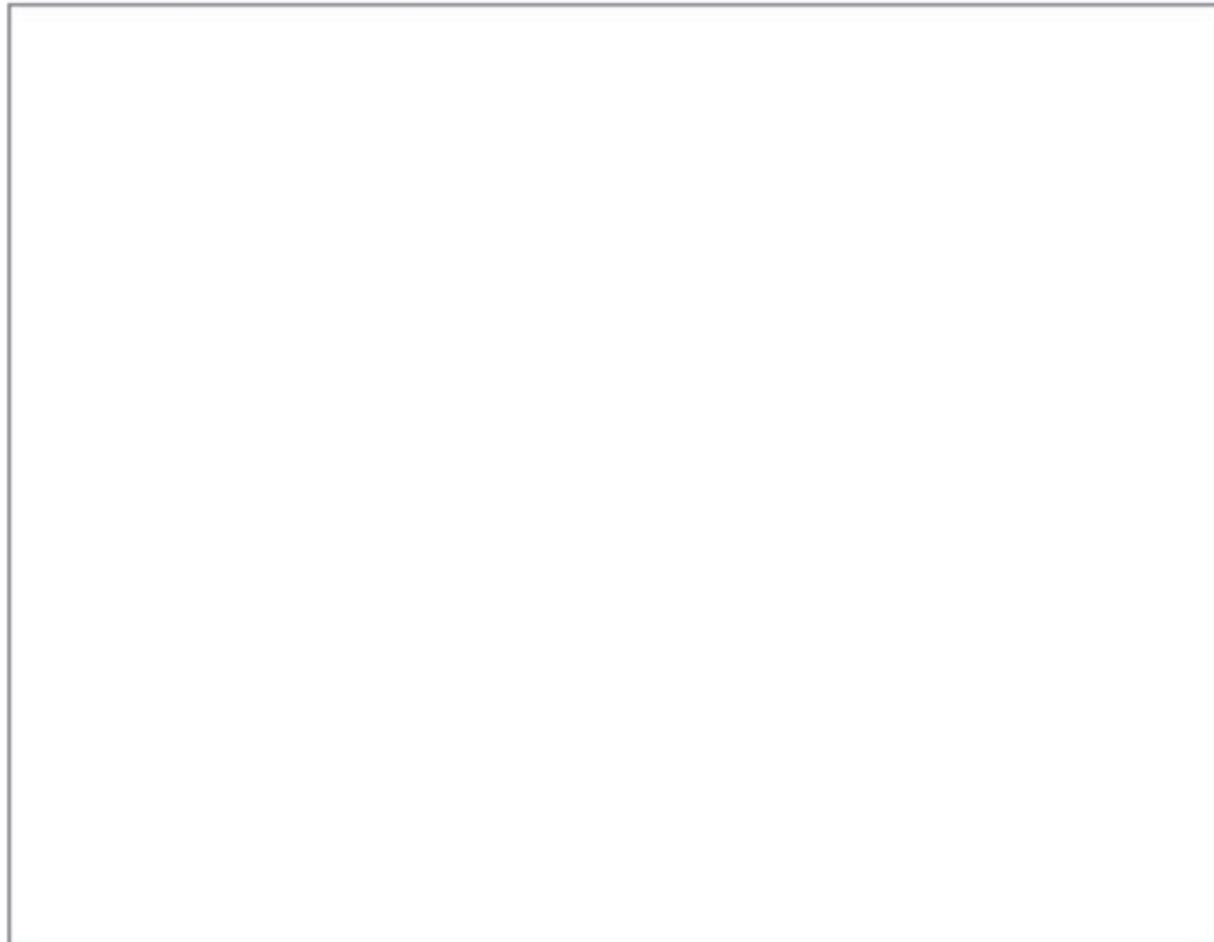
**Event-based**



**Asynchronous**

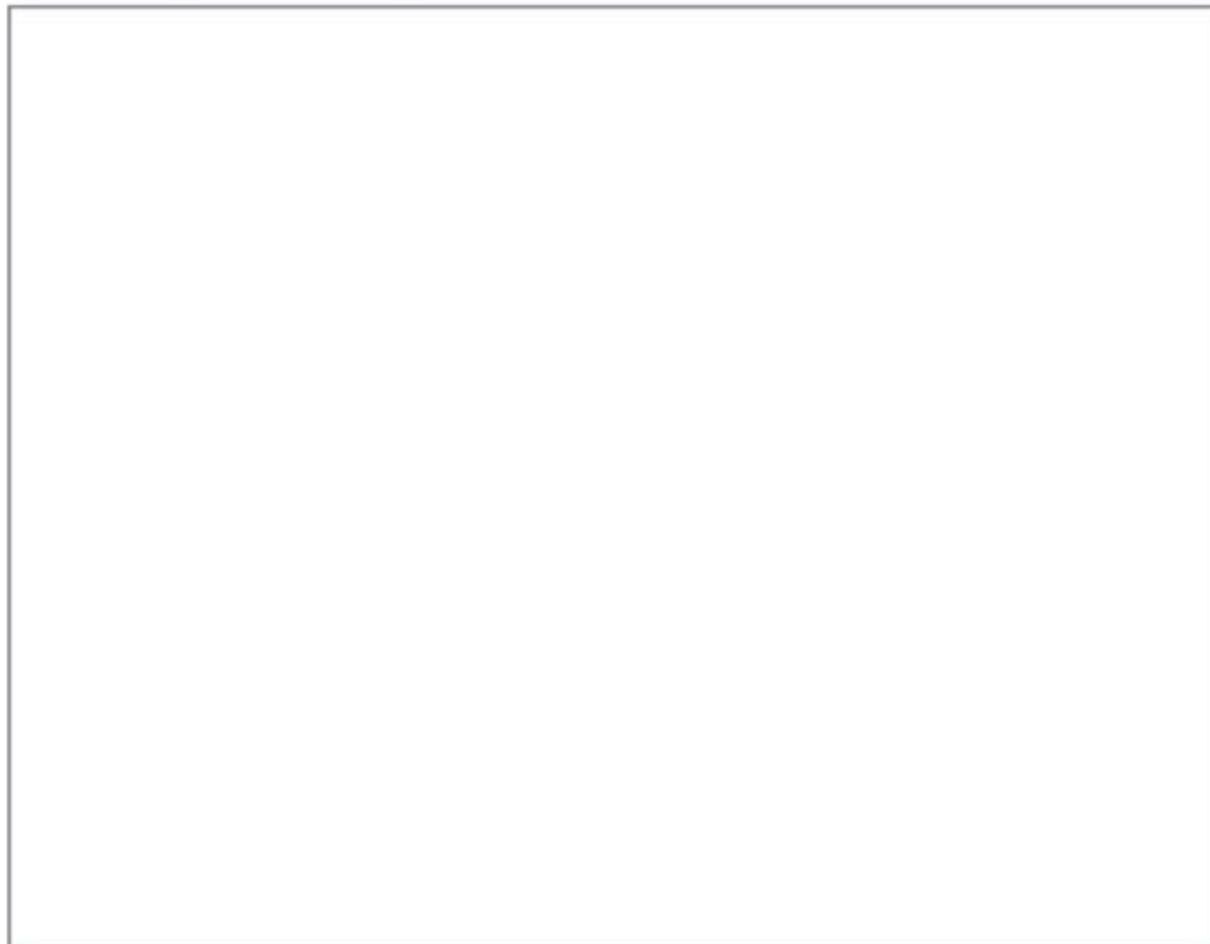


## Request/Response



**Request/Response**

**Event-based**



**Request/Response**

**Event-based**

**Synchronous**

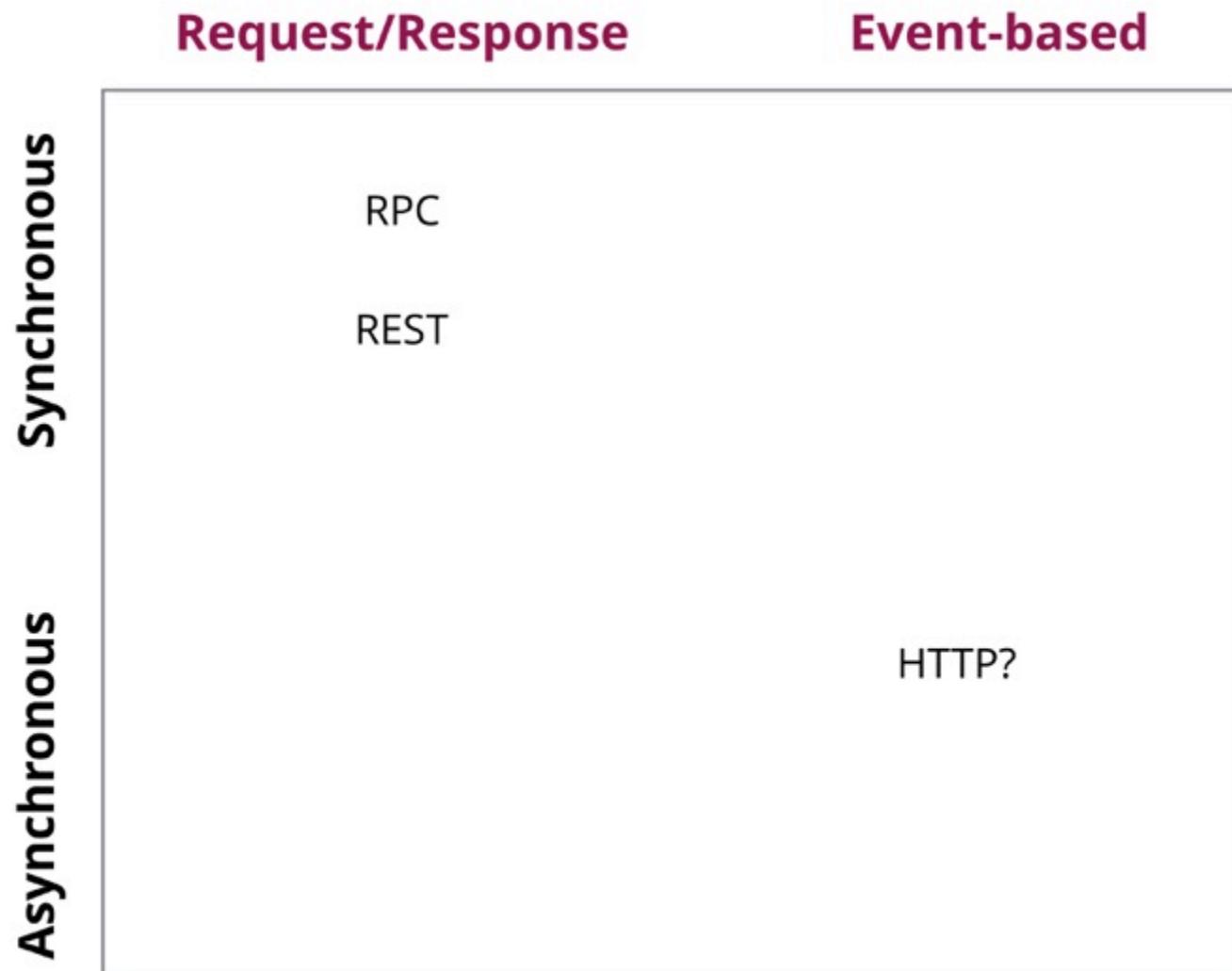
**Request/Response**

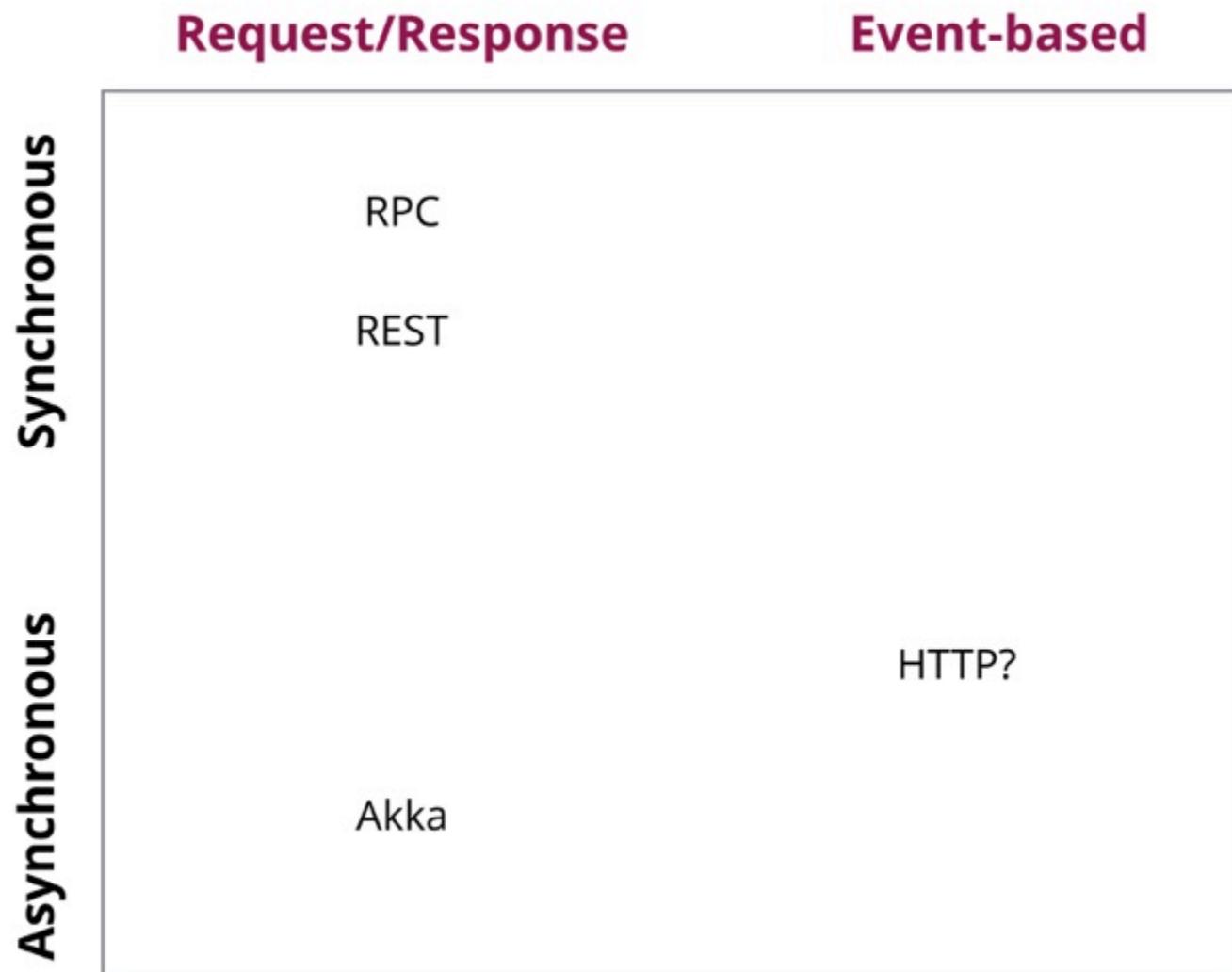
**Event-based**

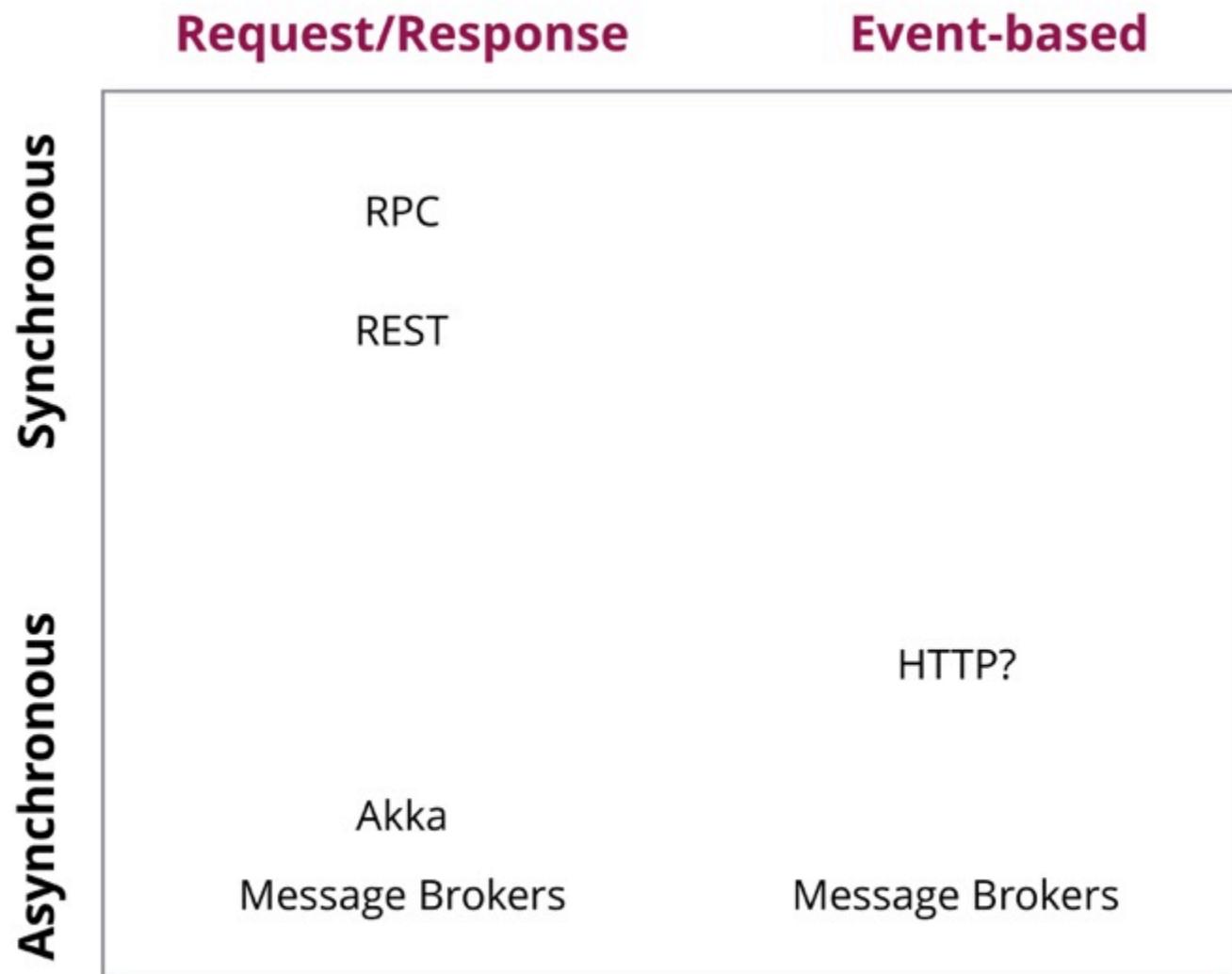
Synchronous

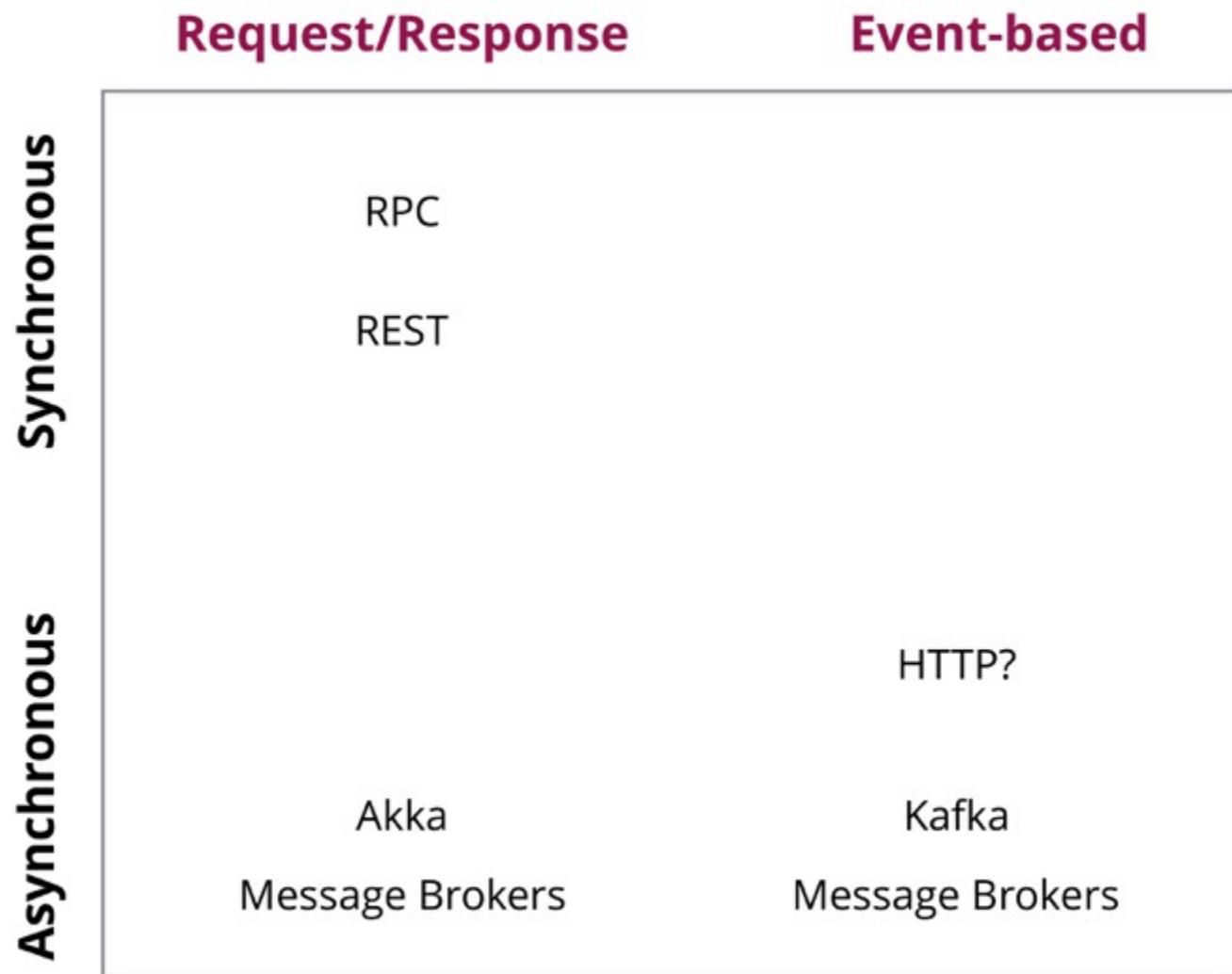
Asynchronous











Simple = Synchronous, Request  
Response Communication

Simple = Synchronous, Request  
Response Communication



HTTP

Very well supported

Very well supported

Good error handling semantics

Very well supported

Good error handling semantics

Cache controls

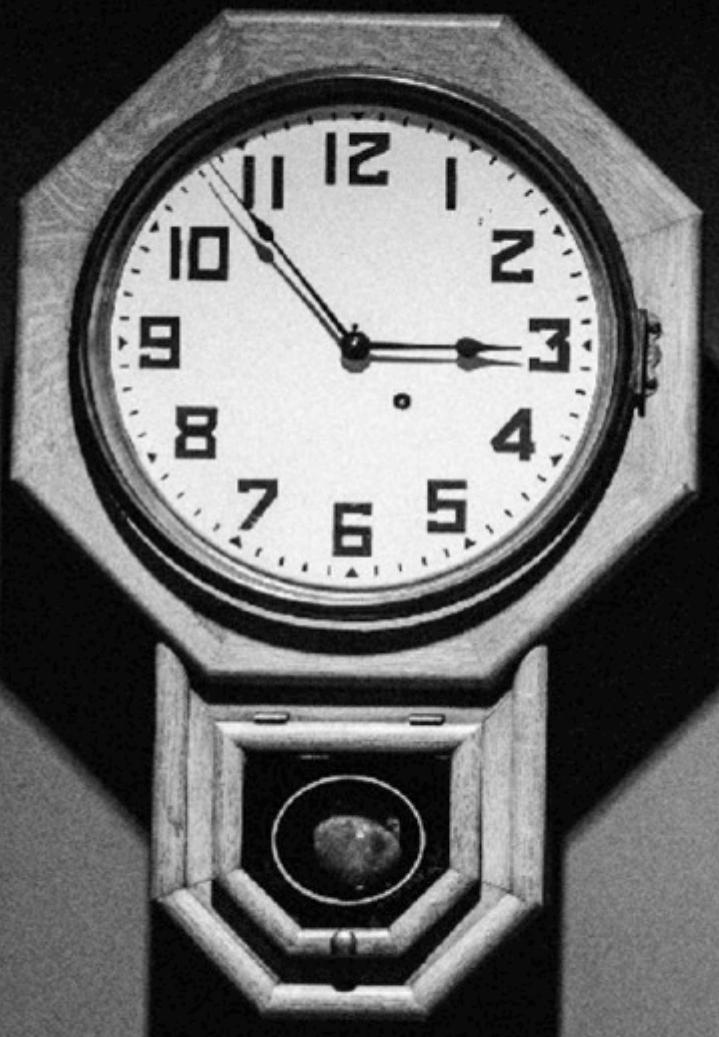
Very well supported

Good error handling semantics

Cache controls

Easy to scale - good, reliable  
(boring?) technology

Where does “simple” stop being  
good enough?

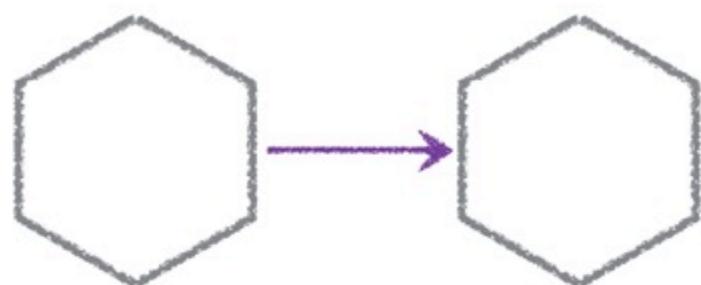


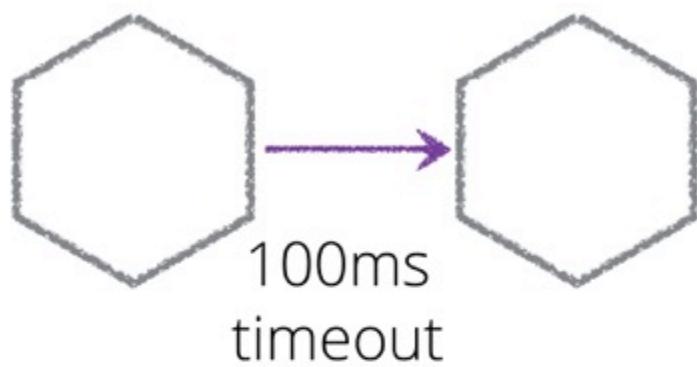


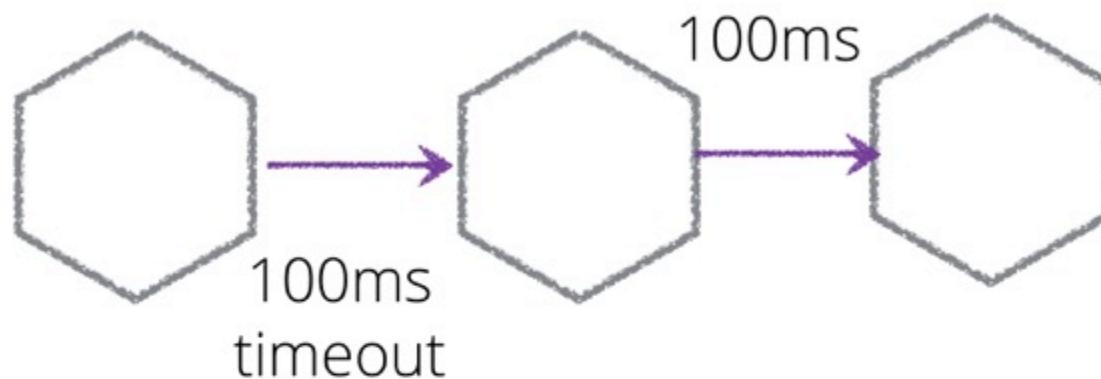
Long-lived processes

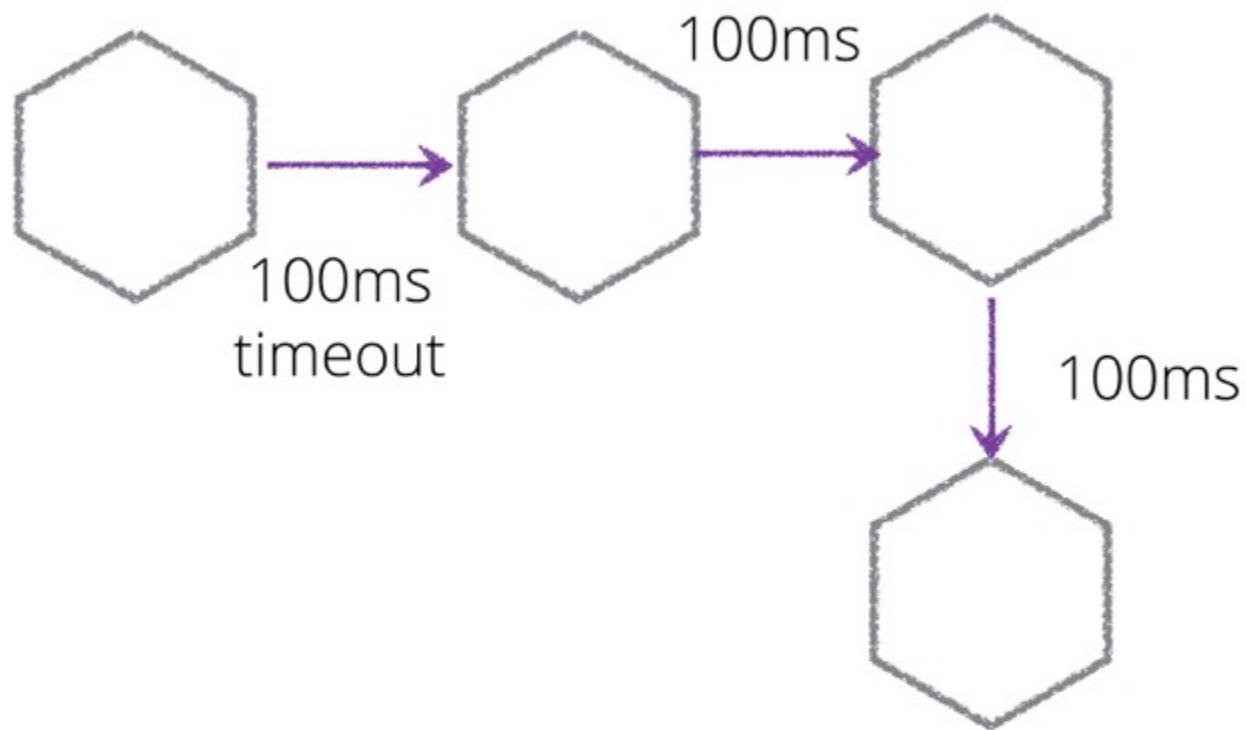


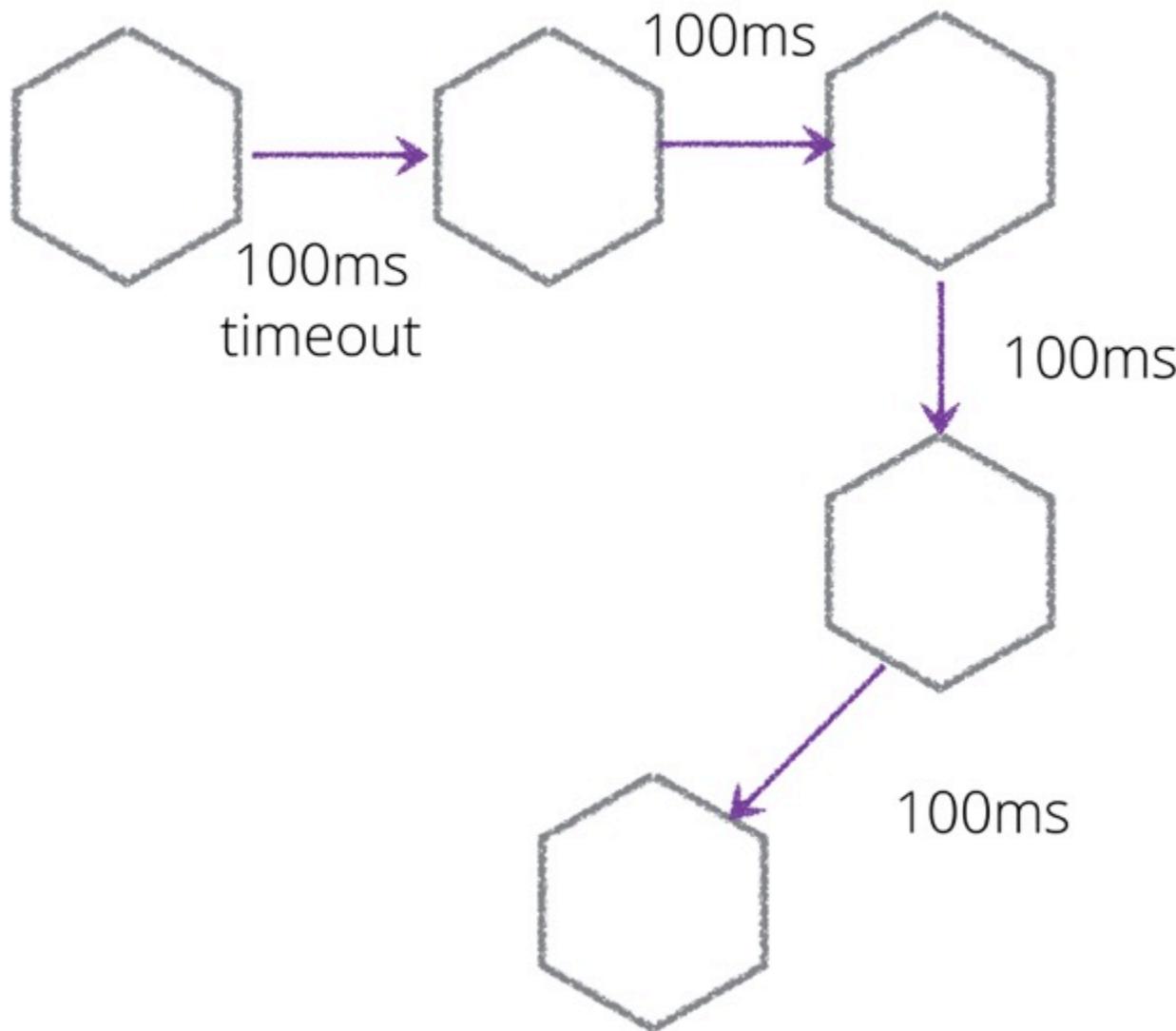
Call overhead

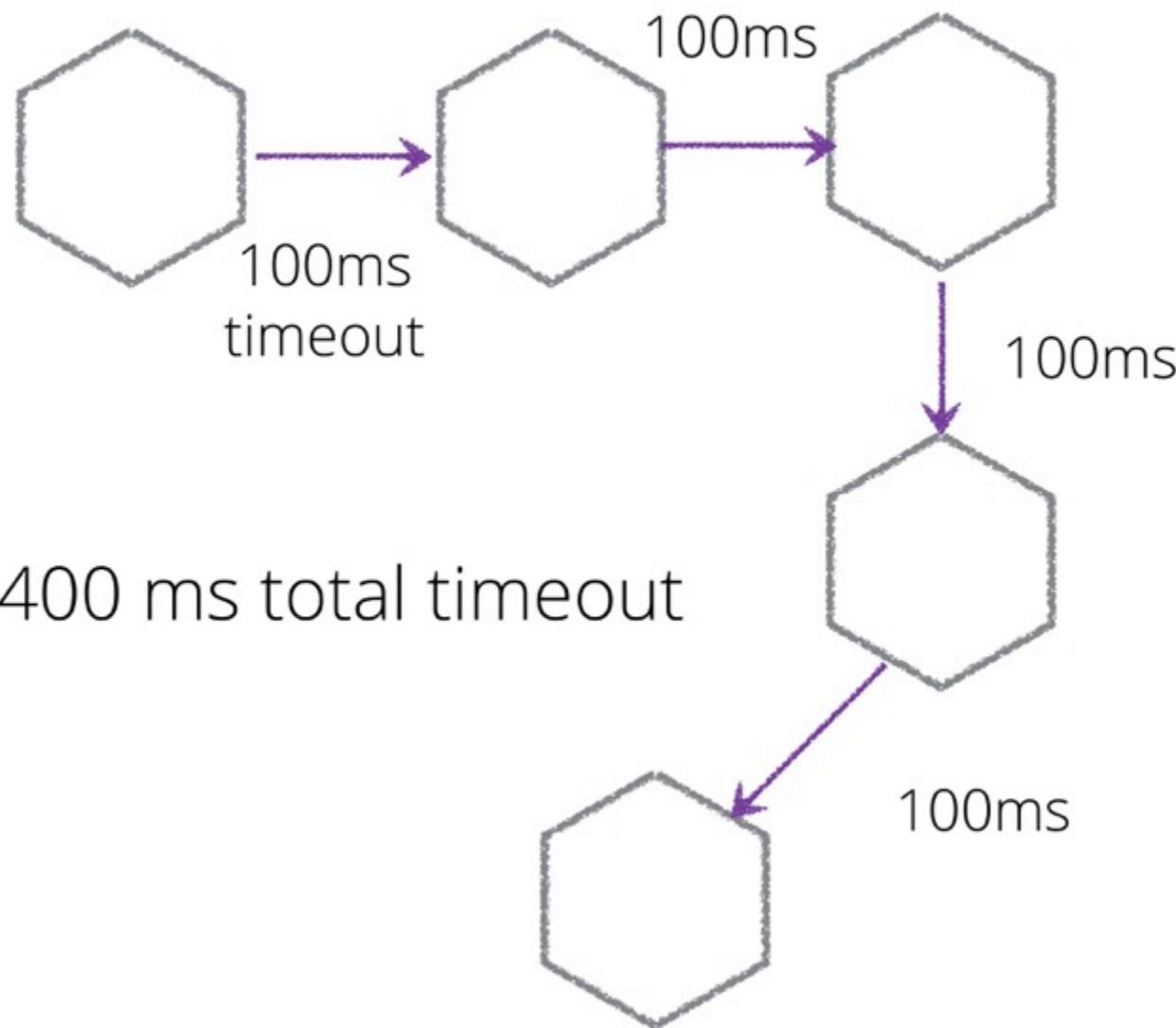








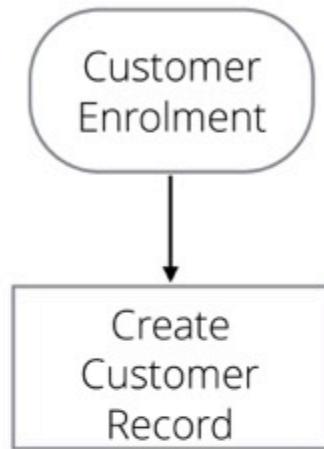


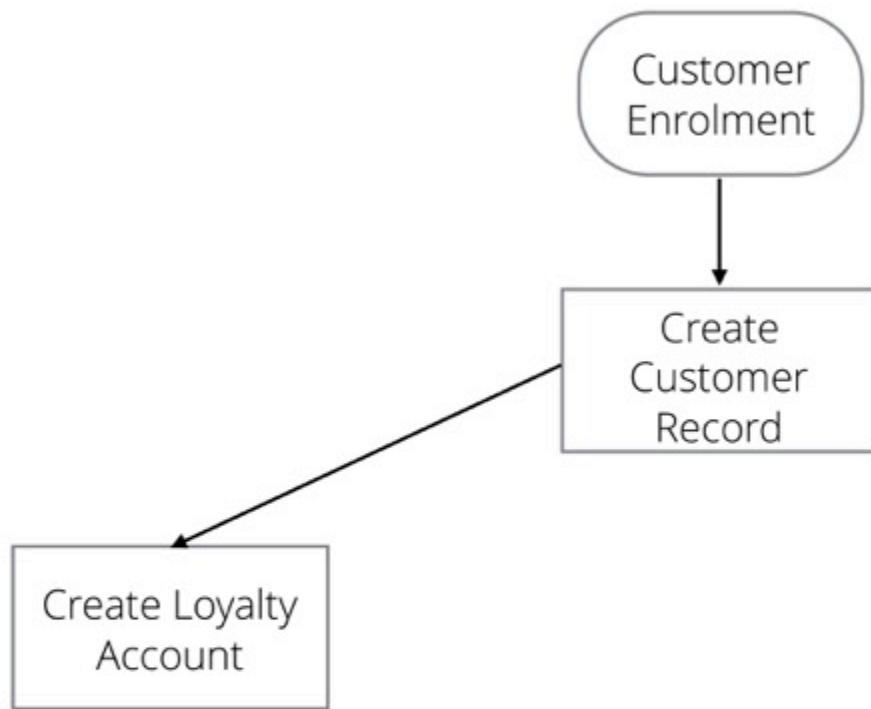


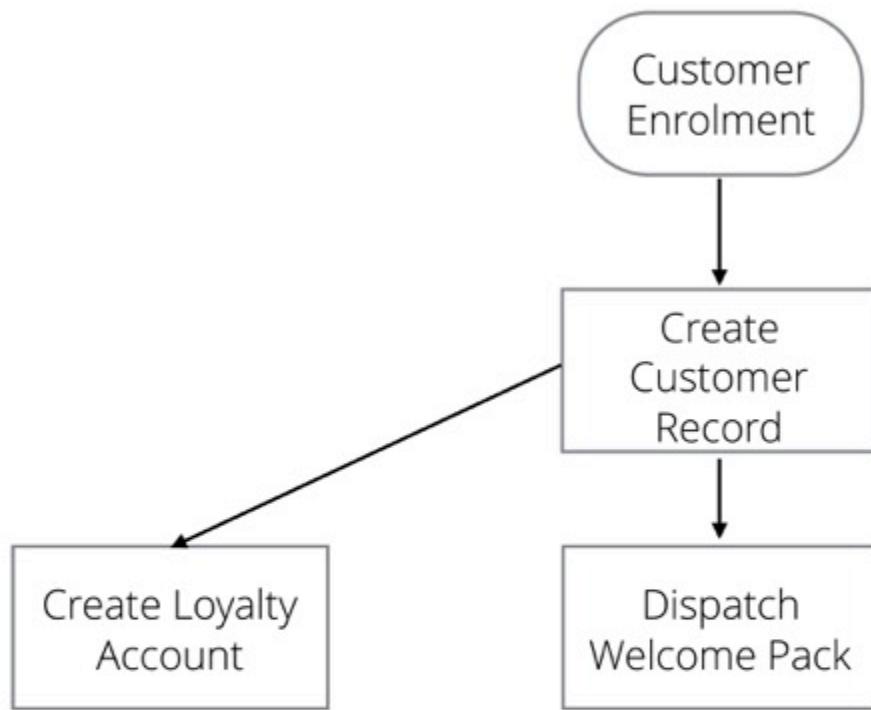
## Choreography vs Orchestration

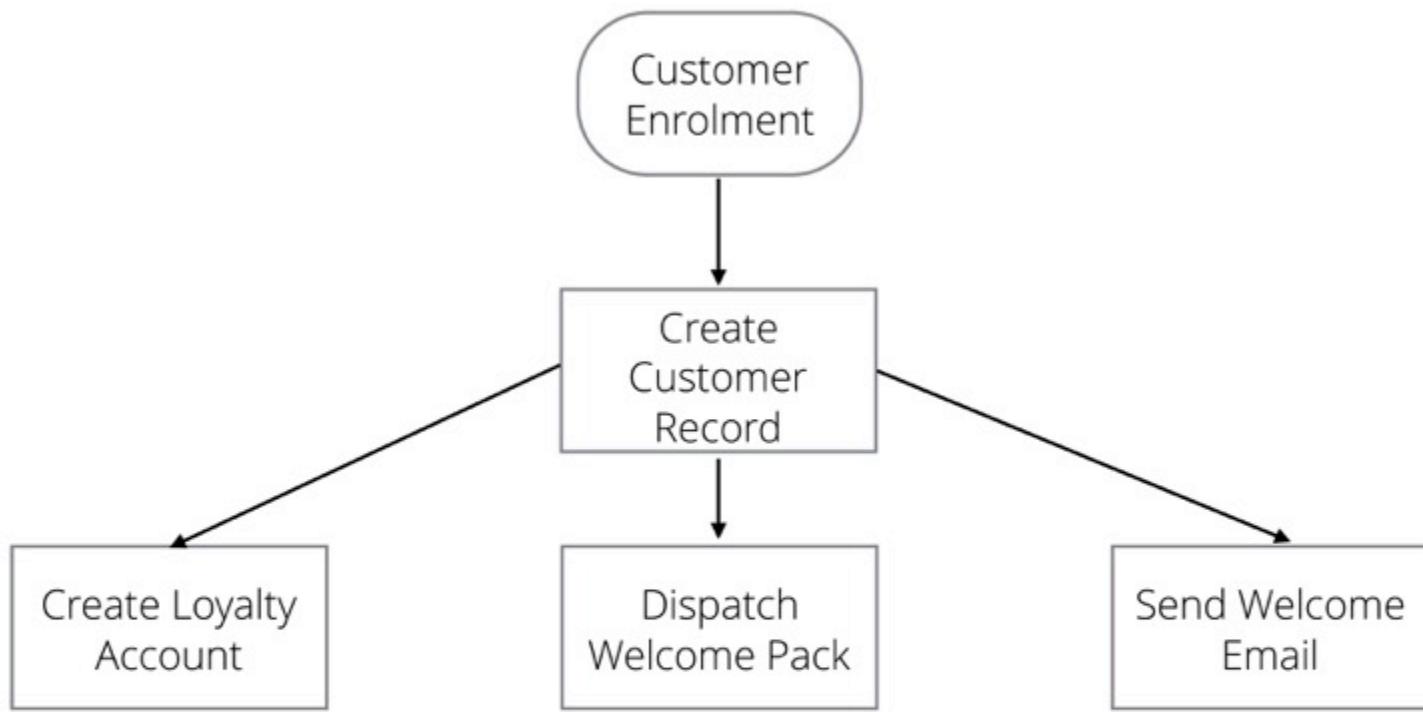
Customer  
Enrolment

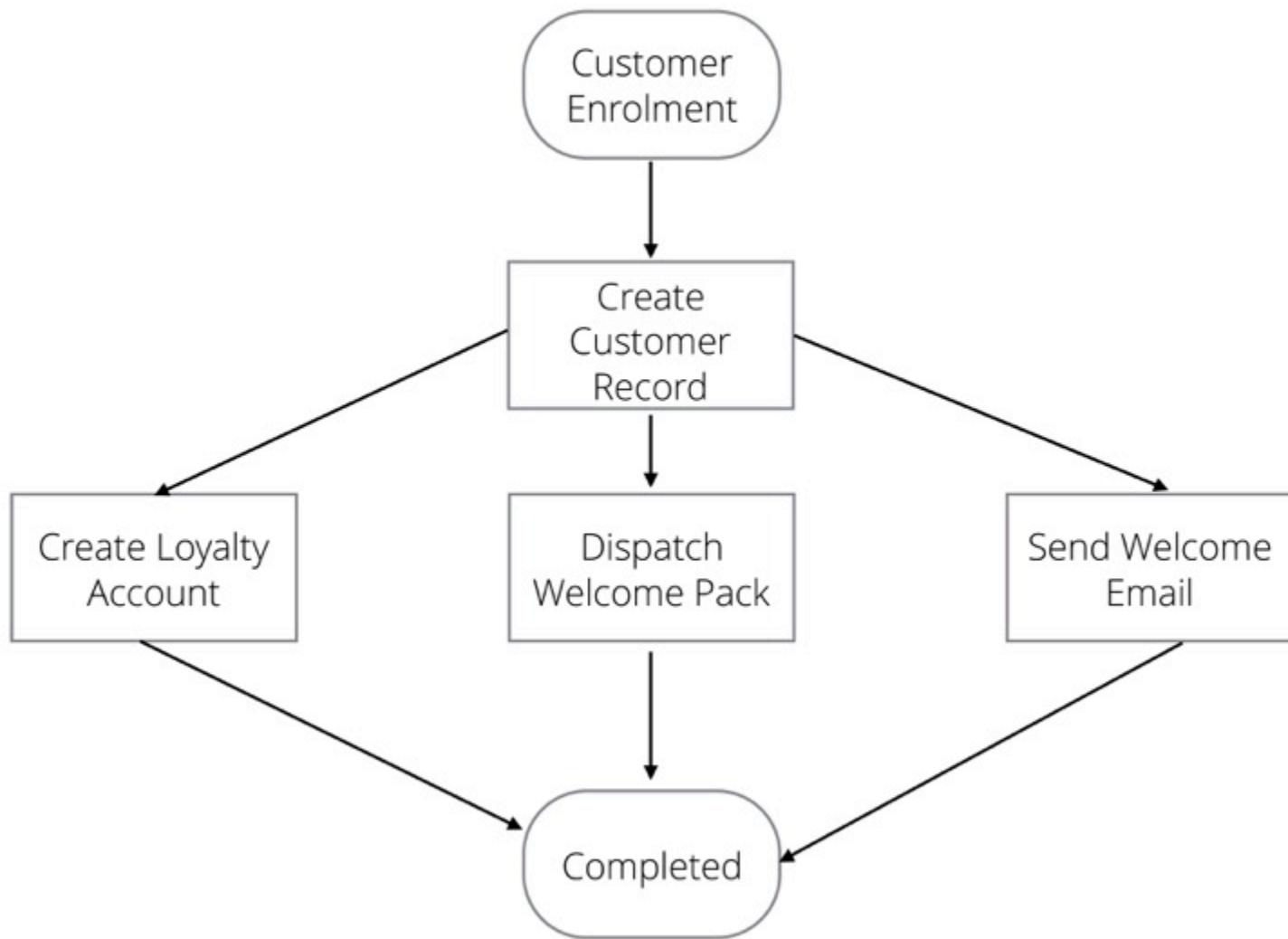




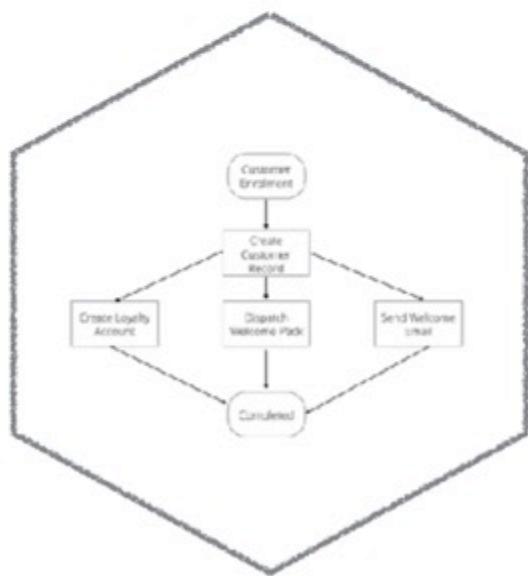




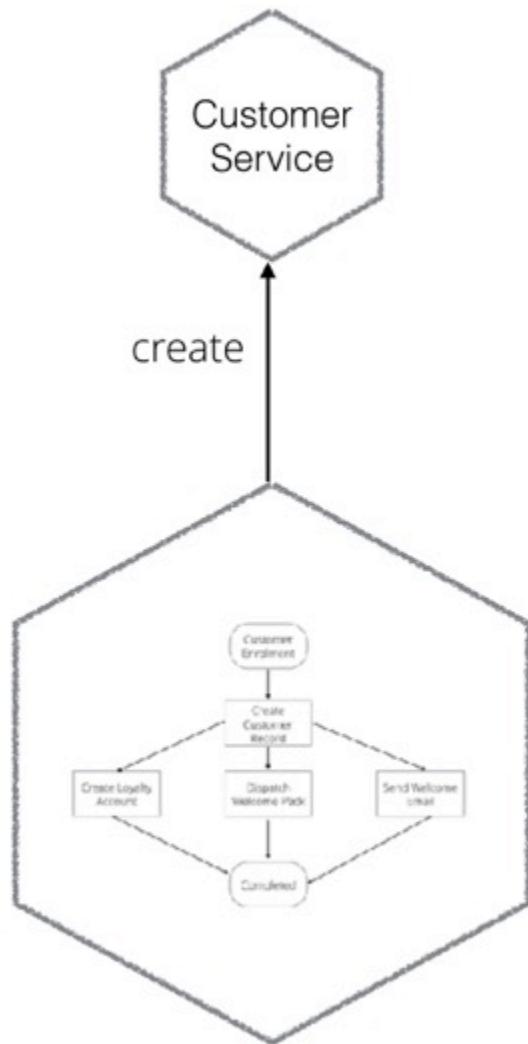




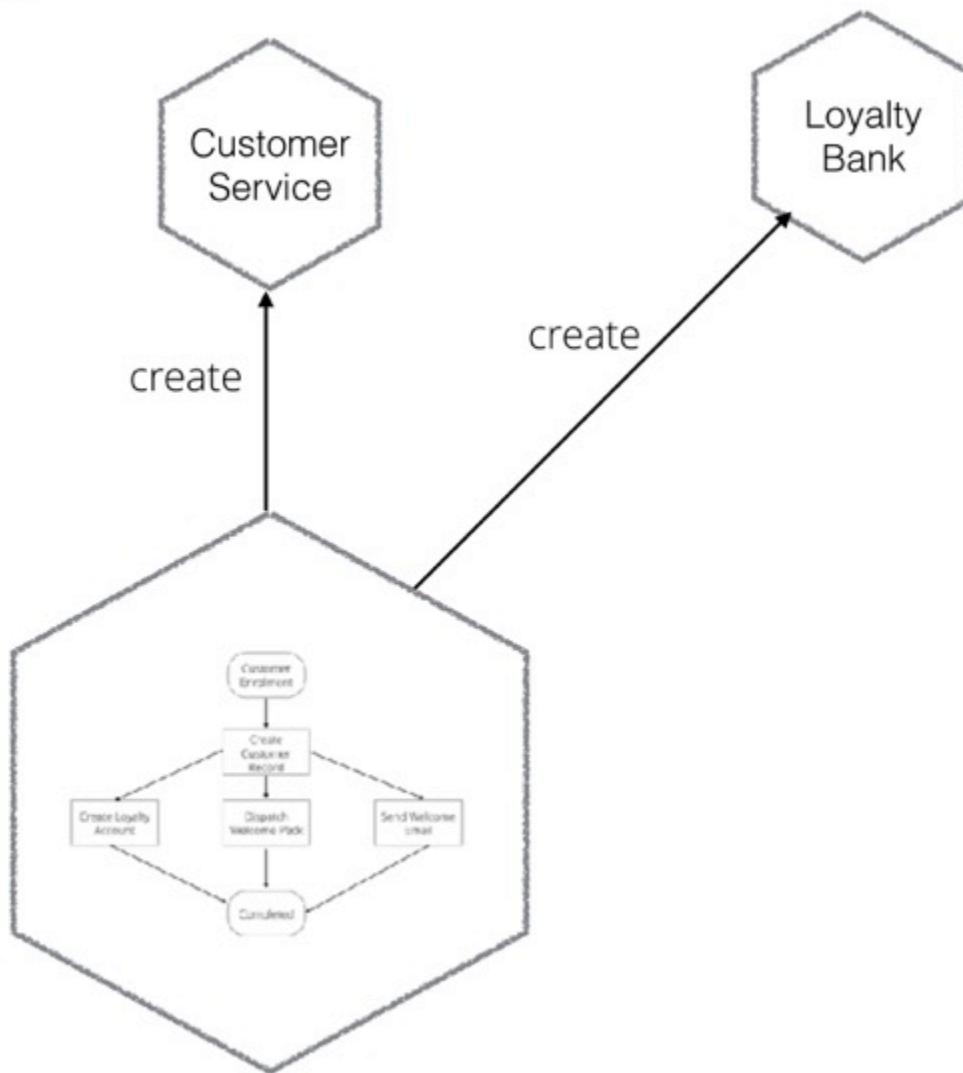
## ORCHESTRATION



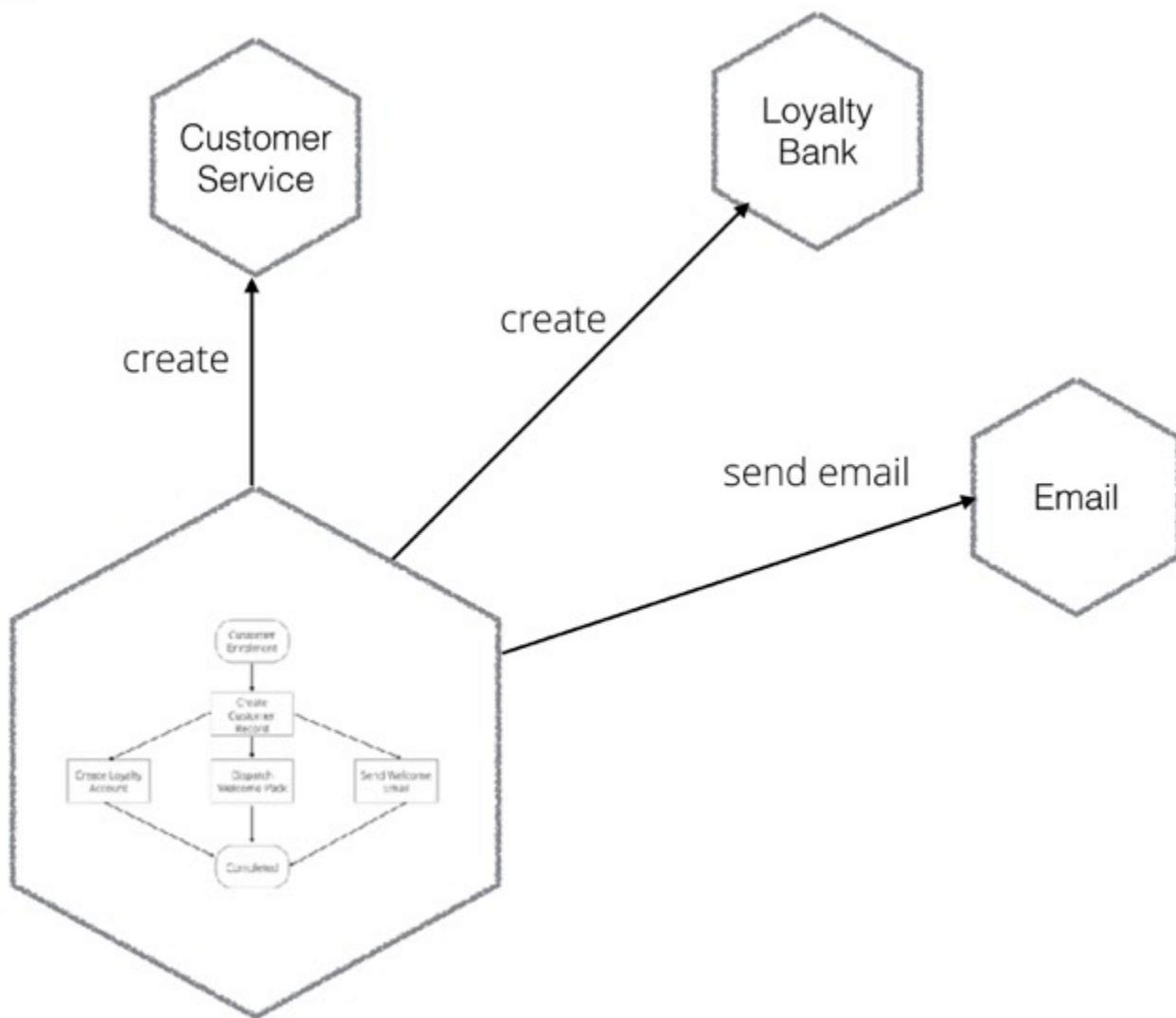
## ORCHESTRATION



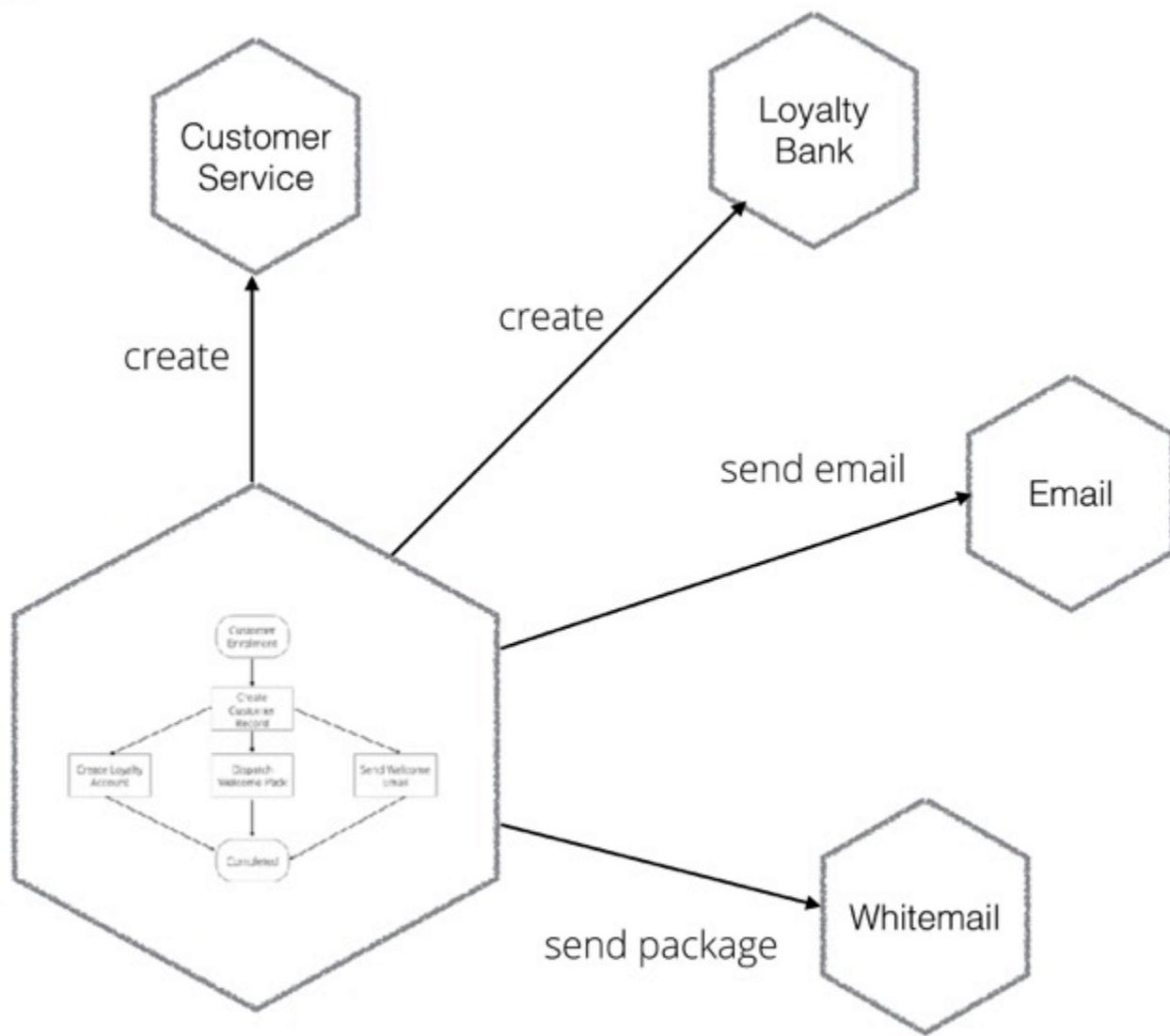
## ORCHESTRATION



## ORCHESTRATION



## ORCHESTRATION



# **Pros**

## **Pros**

Explicit representation of business process

## **Pros**

Explicit representation of business process

Know in-line if there has been a problem

## **Pros**

Explicit representation of business process

Know in-line if there has been a problem

## **Cons**

## **Pros**

Explicit representation of business process

Know in-line if there has been a problem

## **Cons**

Can be fairly coupled

## **Pros**

Explicit representation of business process

Know in-line if there has been a problem

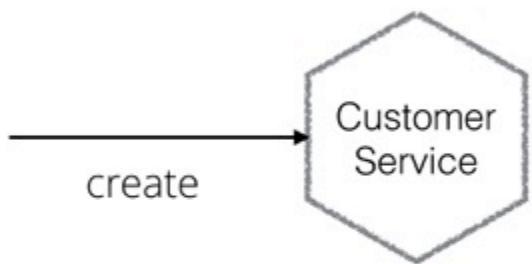
## **Cons**

Can be fairly coupled

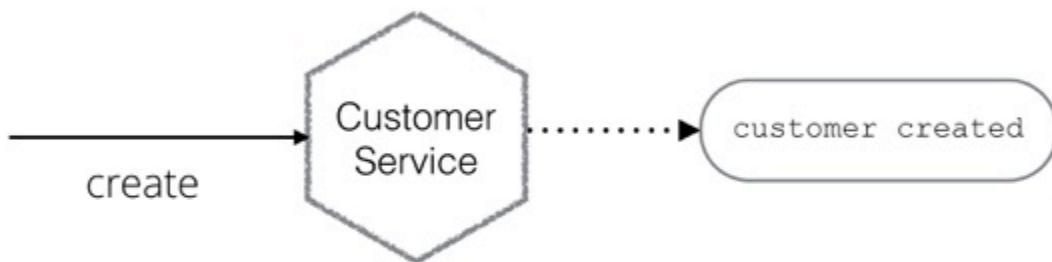
Can lead to overly smart (and dumb  
services)

**CHOREOGRAPHED**

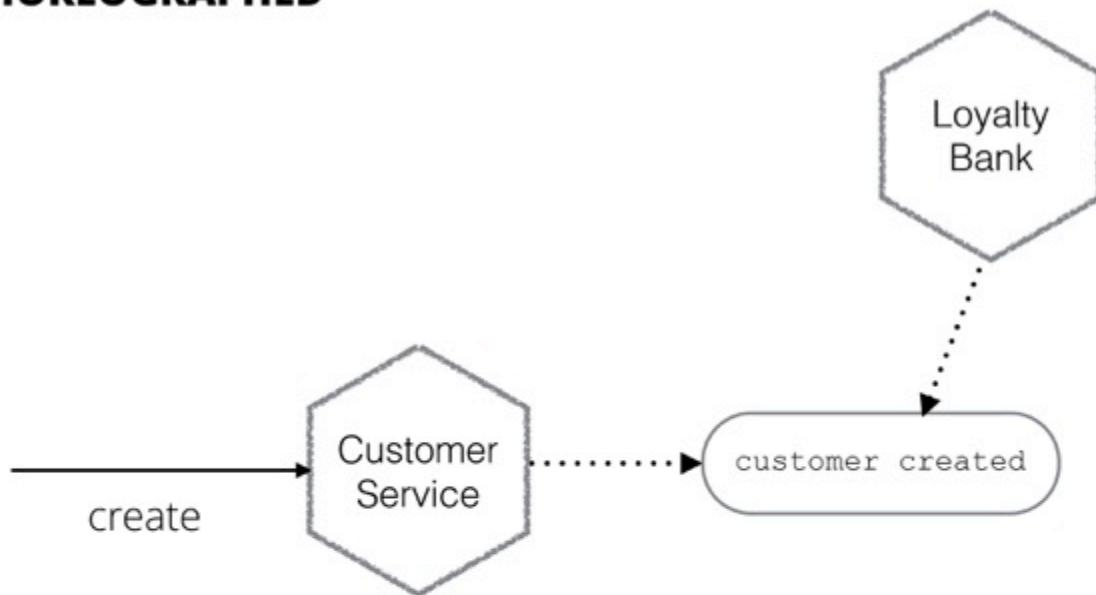
## **CHOREOGRAPHED**



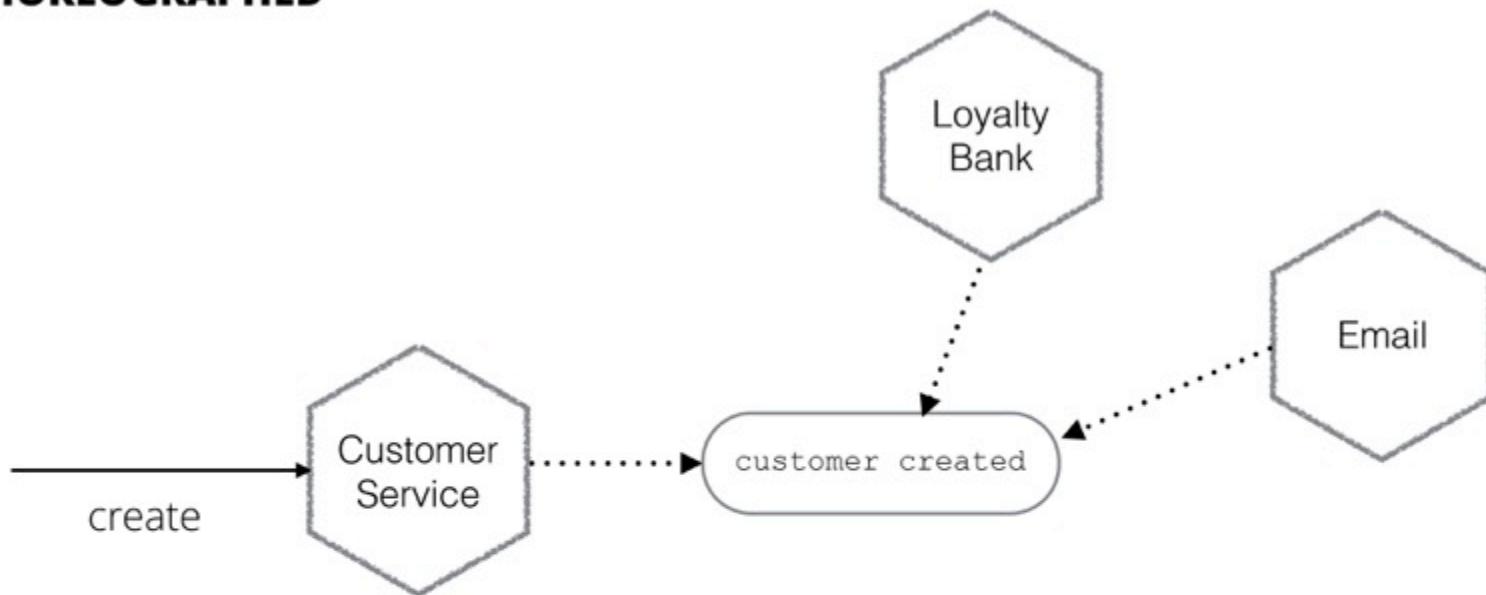
## CHOREOGRAPHED



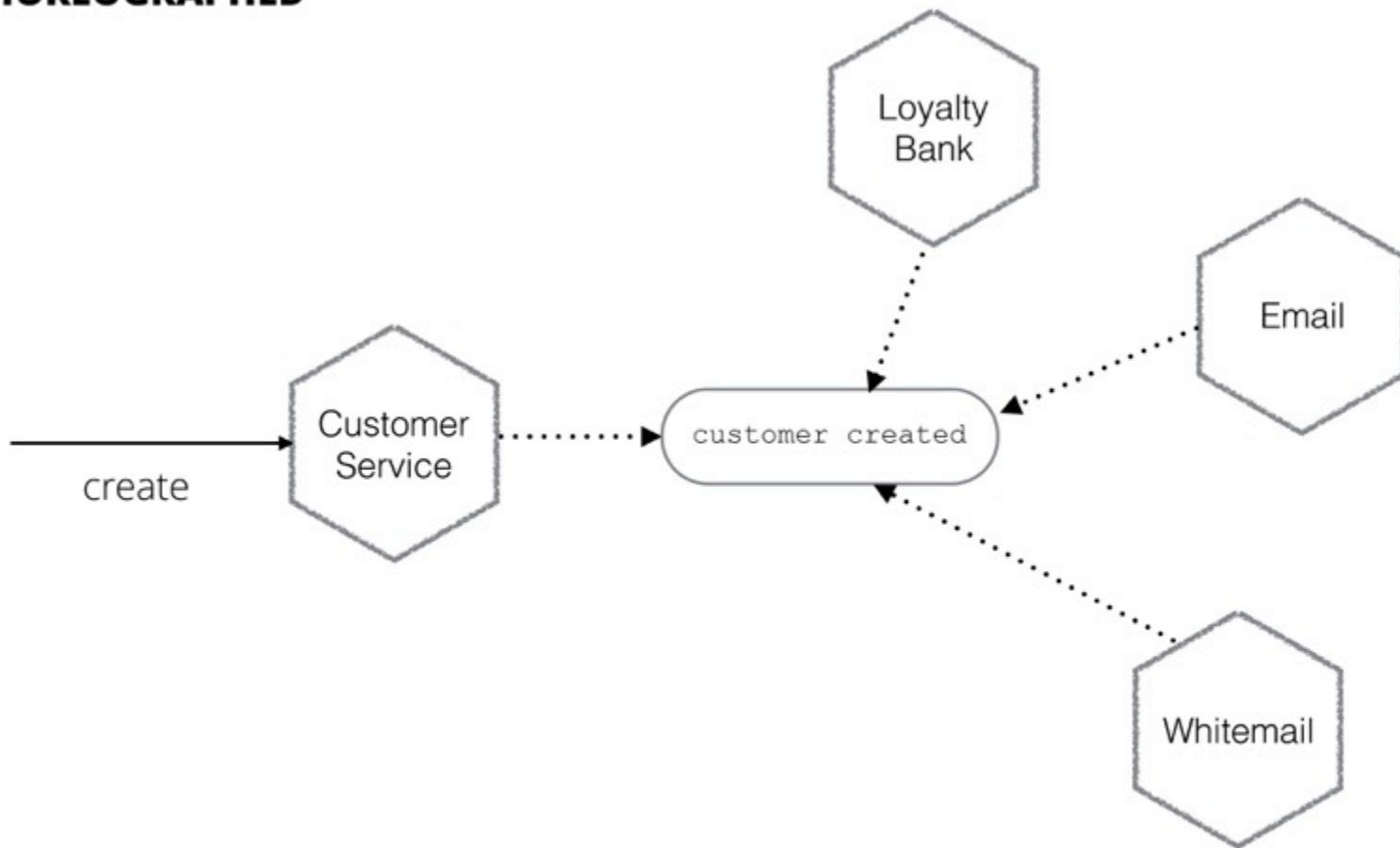
## CHOREOGRAPHED



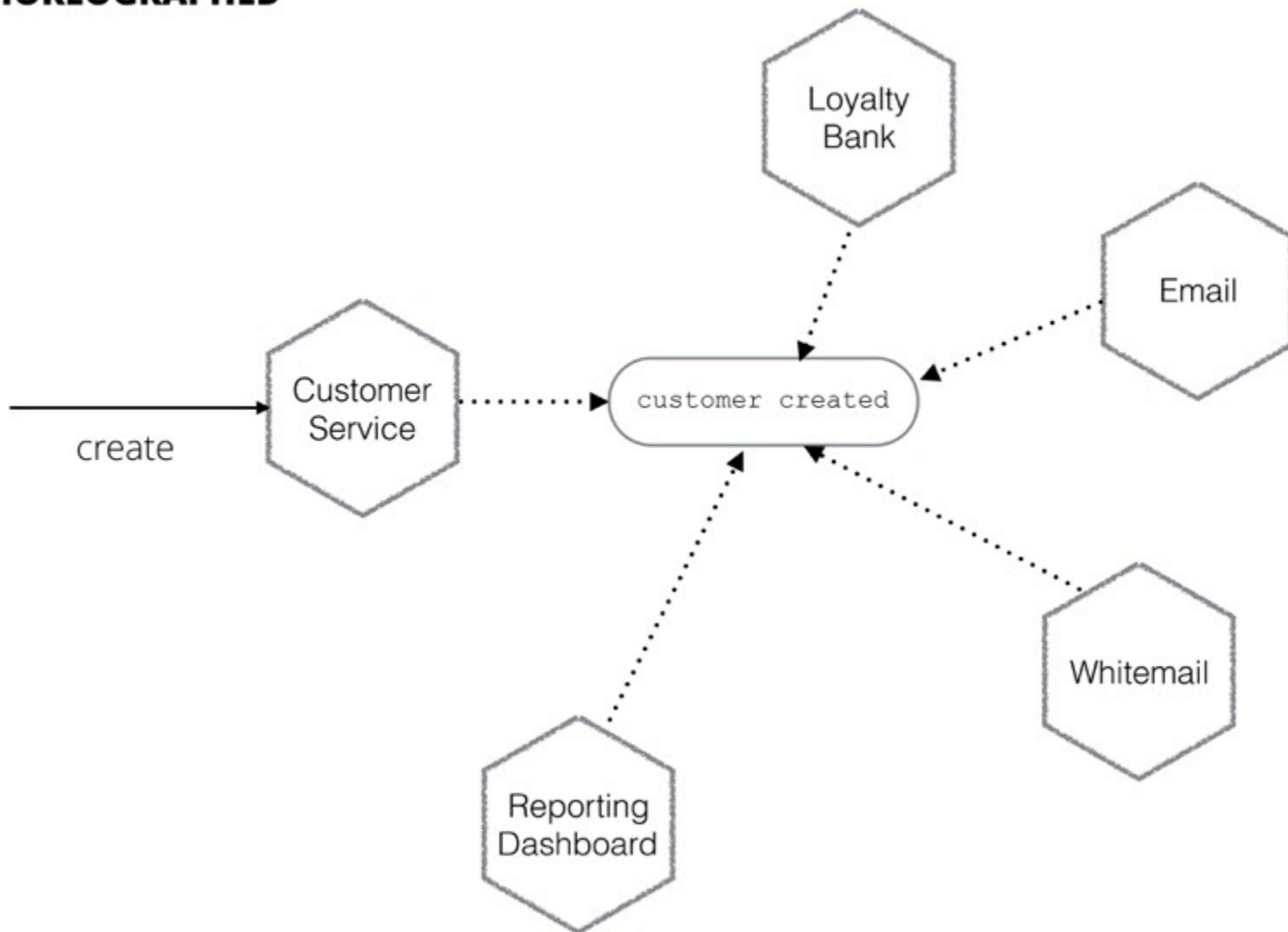
## CHOREOGRAPHED



## CHOREOGRAPHED



## CHOREOGRAPHED



# **Pros**

# **Pros**

Highly decoupled

## **Pros**

Highly decoupled

Evenly distributed smarts

## **Pros**

Highly decoupled

Evenly distributed smarts

## **Cons**

## **Pros**

Highly decoupled

Evenly distributed smarts

## **Cons**

Lost explicit business process mapping

## **Pros**

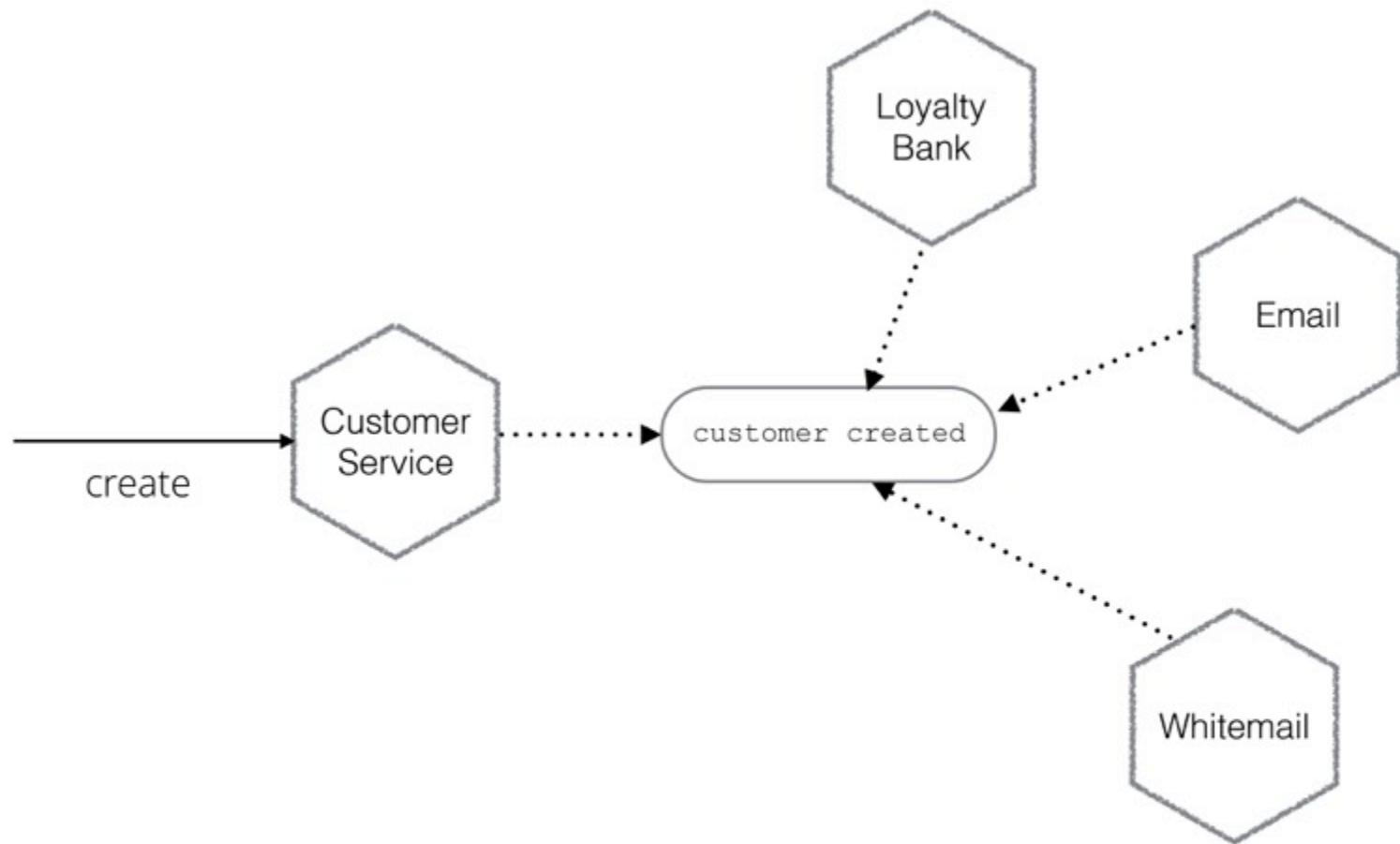
Highly decoupled

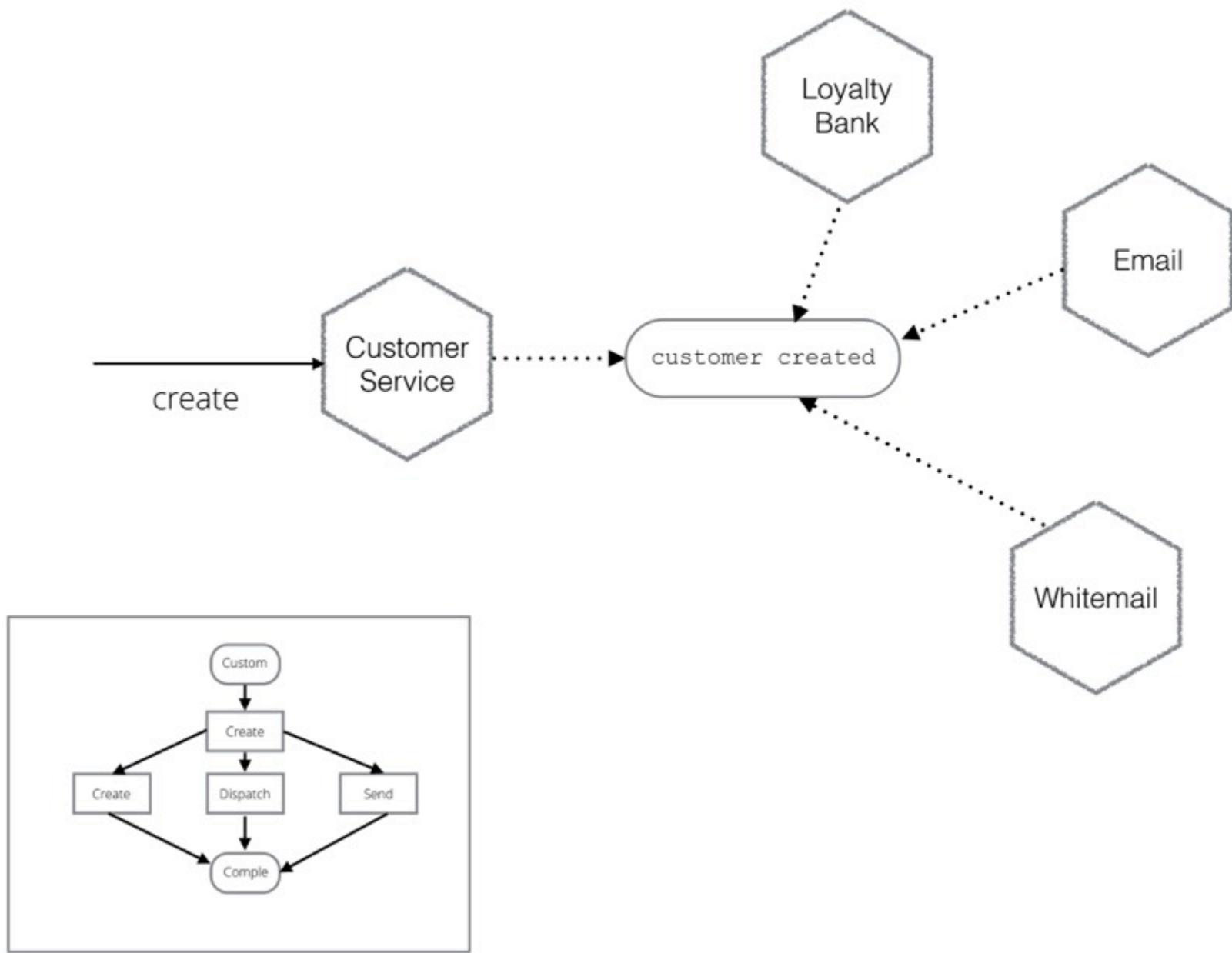
Evenly distributed smarts

## **Cons**

Lost explicit business process mapping

Understanding completion or error states  
is complex







# Summary

Keep it simple

# Summary

Keep it simple

Think about interaction style  
first, tech second

# Summary

Keep it simple

Think about interaction style  
first, tech second

Orchestration vs Choreography

# MONOLITHS TO MICROSERVICES

Sam Newman

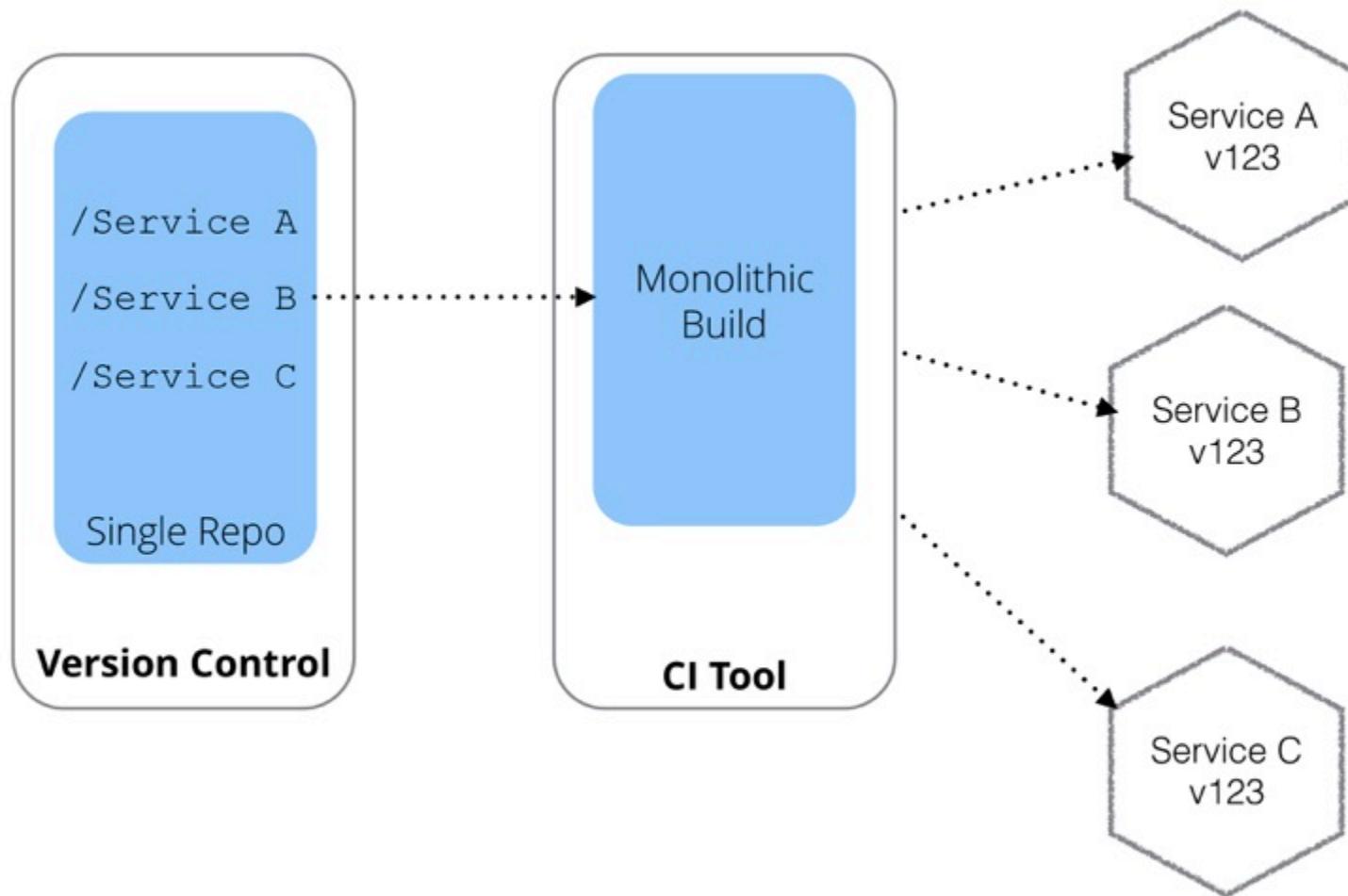
## Build & Deployment

Hands up if you're doing Continuous Integration!

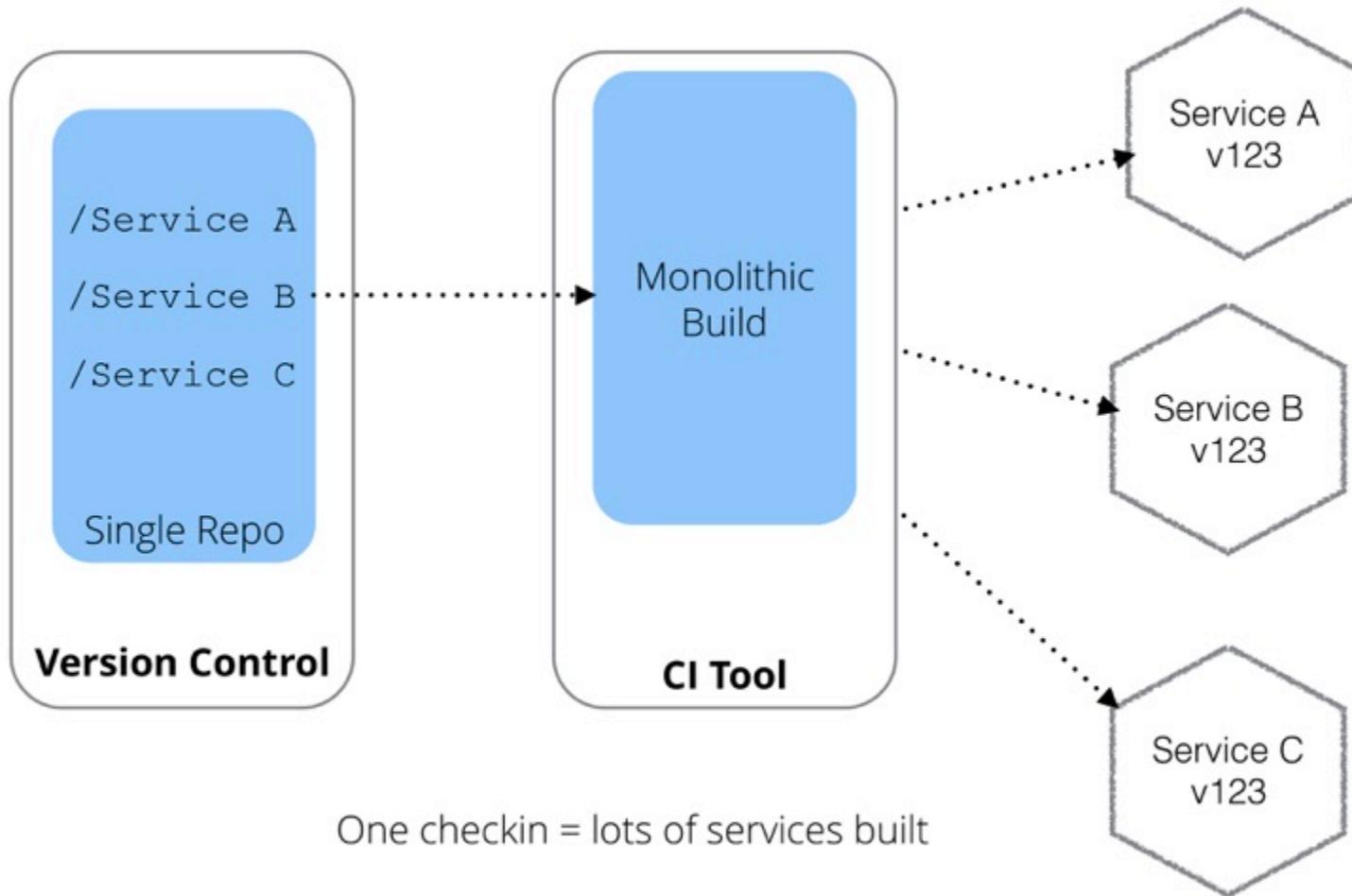
## **RULES OF CONTINUOUS INTEGRATION**

1. Check in to mainline at least once a day
2. Have a suite to validate the integration
3. If the build fails, make it the teams #1 priority to fix it

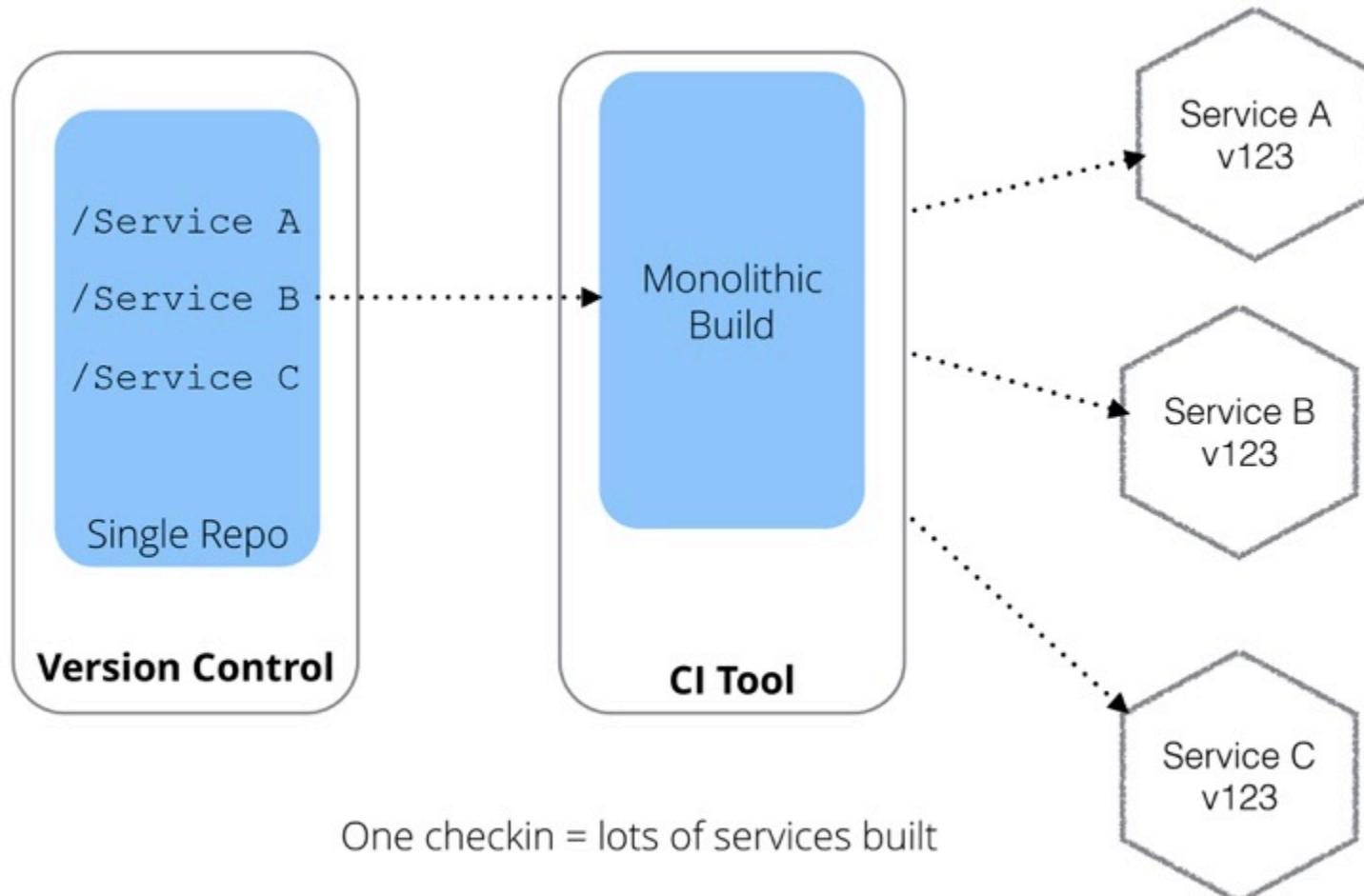
## ONE GIANT BUILD



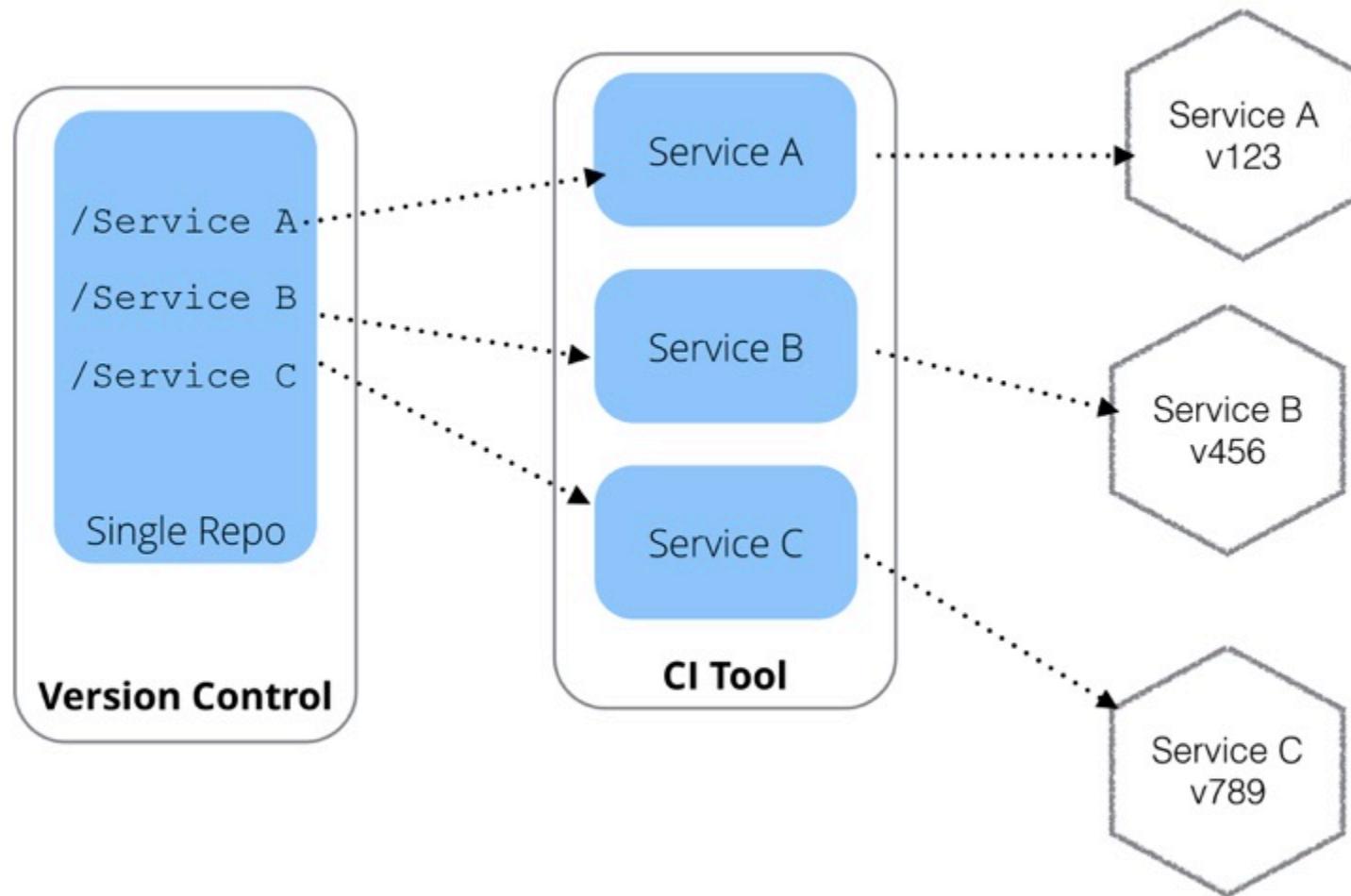
## ONE GIANT BUILD



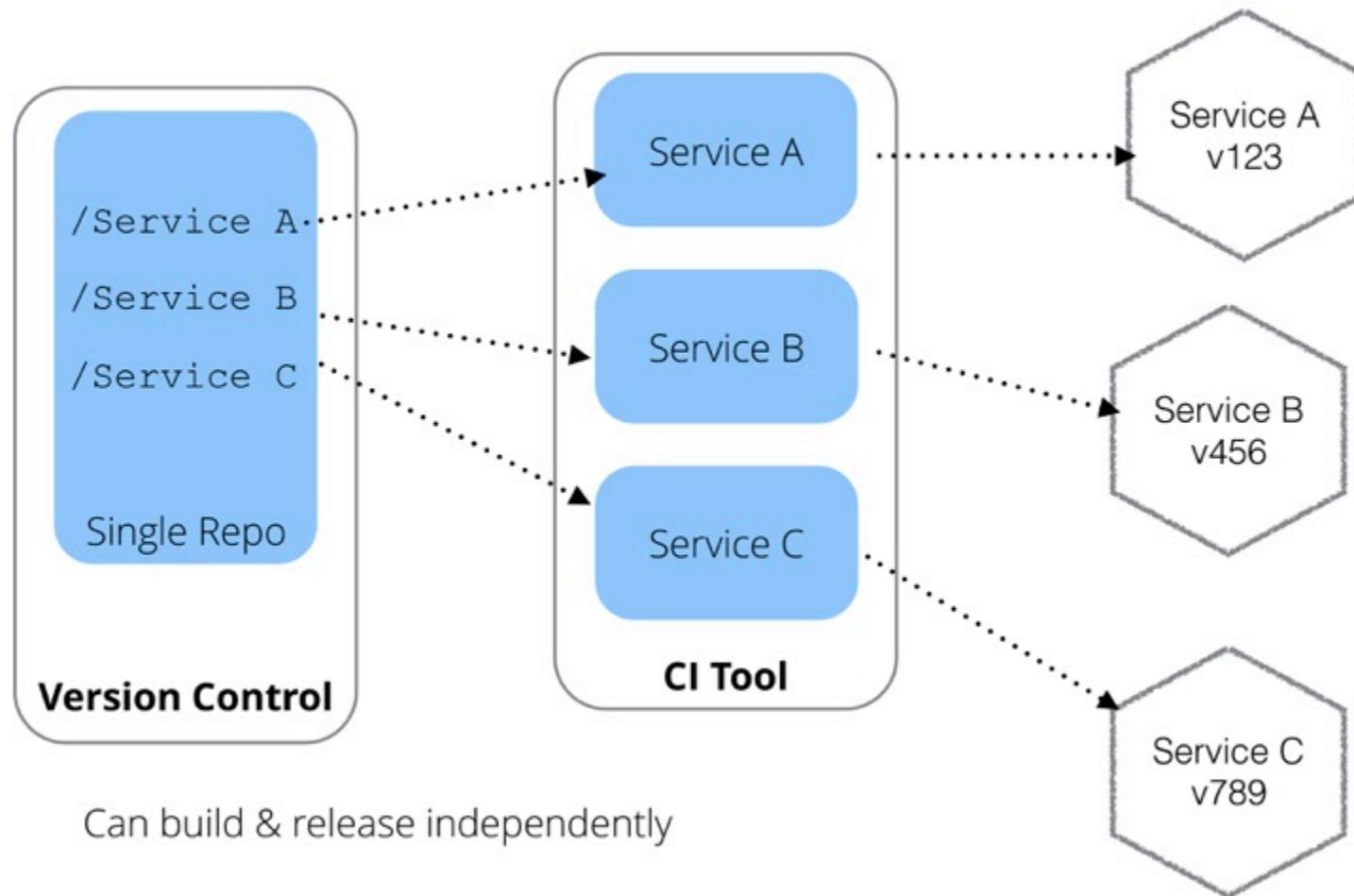
## ONE GIANT BUILD



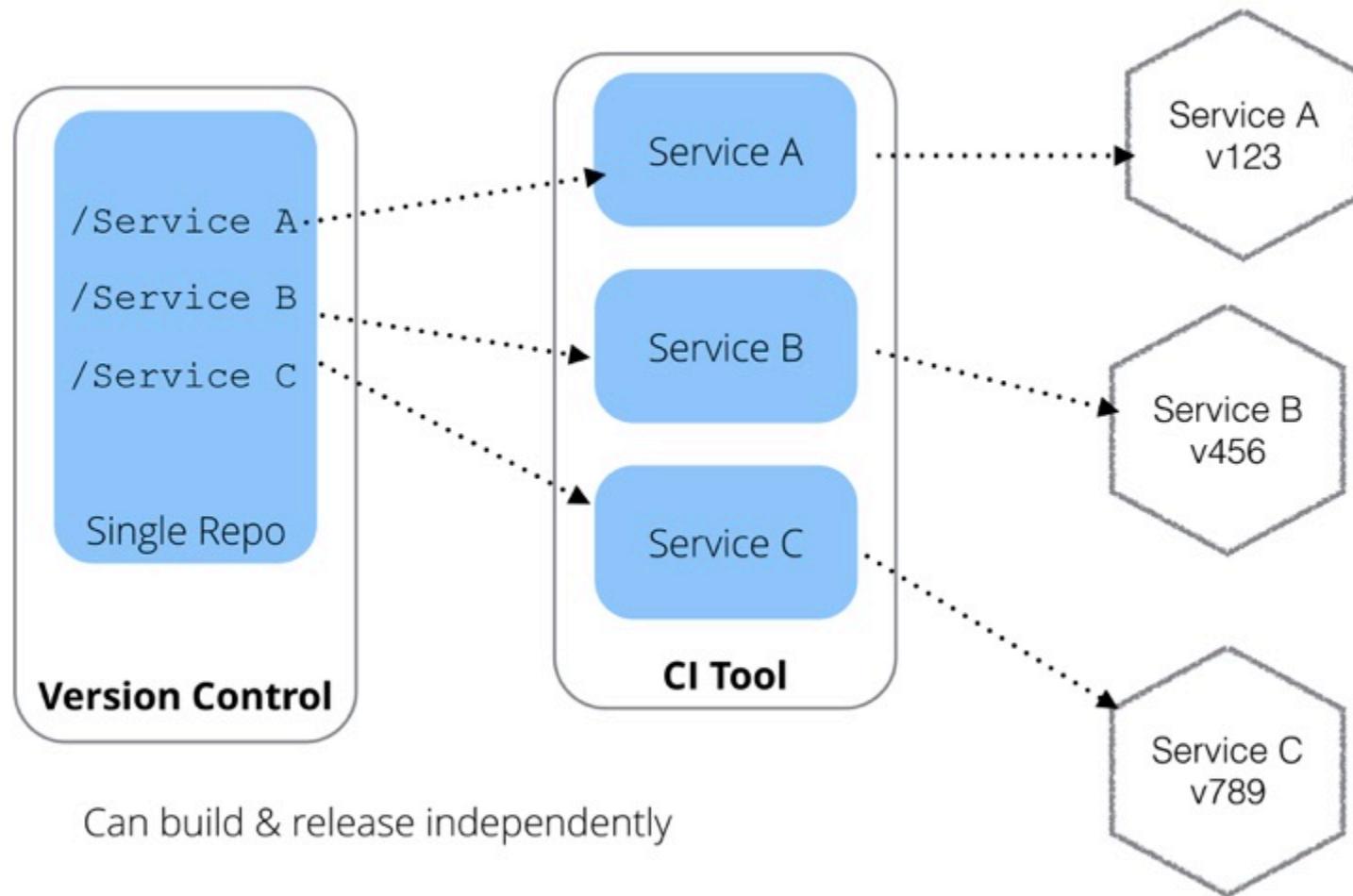
## ONE BUILD PER SERVICE, SINGLE REPO



## ONE BUILD PER SERVICE, SINGLE REPO



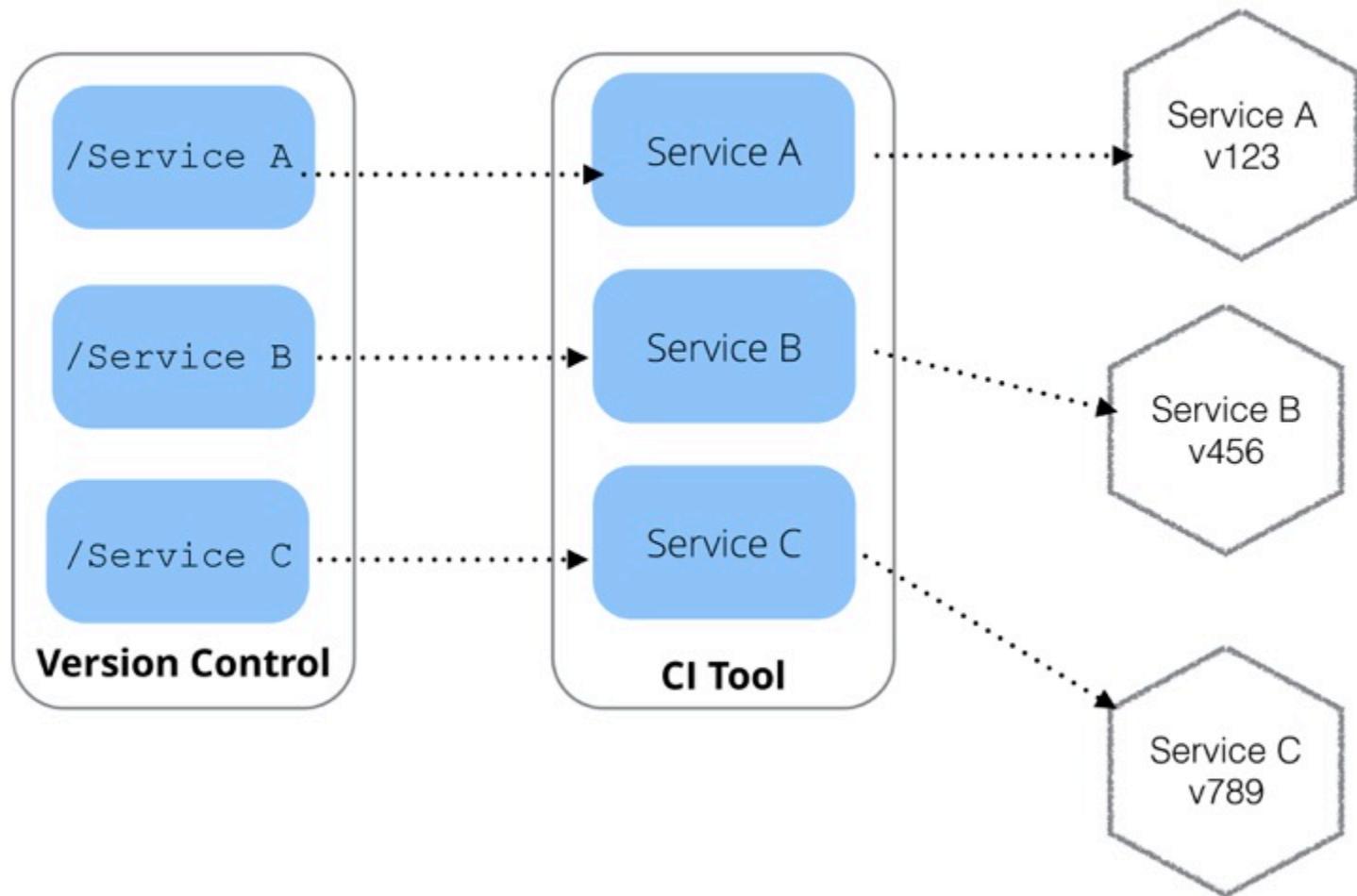
## ONE BUILD PER SERVICE, SINGLE REPO



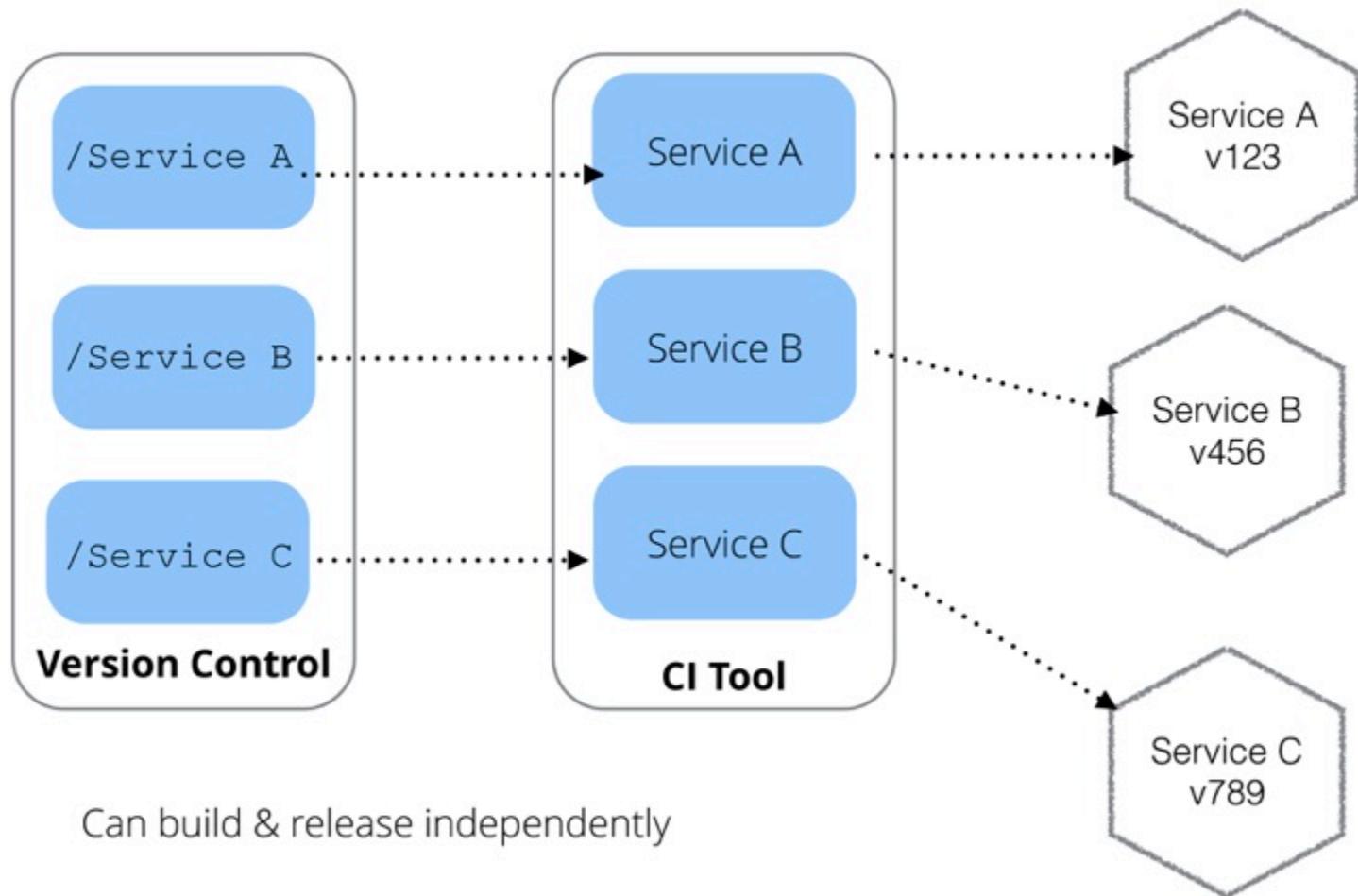
Can build & release independently

Can encourage making cross-repo changes

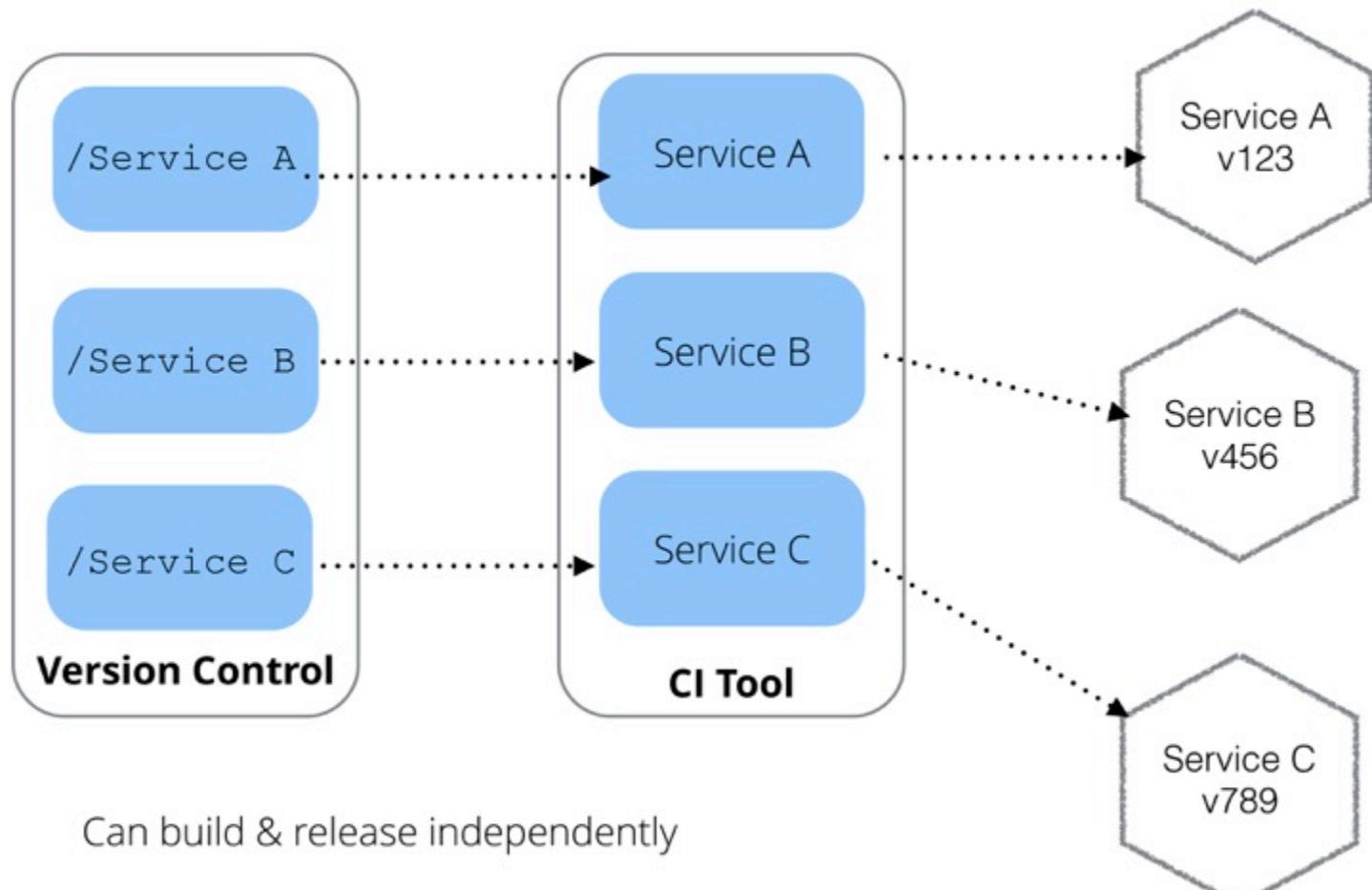
## ONE BUILD & REPO PER SERVICE



## ONE BUILD & REPO PER SERVICE



## ONE BUILD & REPO PER SERVICE



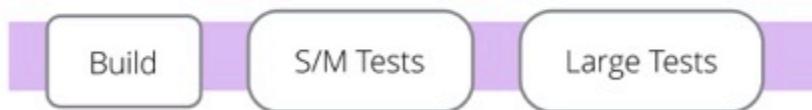
Can build & release independently

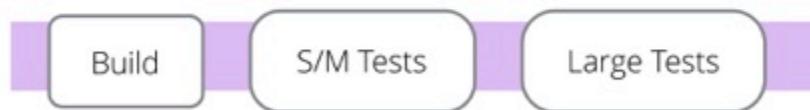
Firmer separation, but developer workflow can suffer

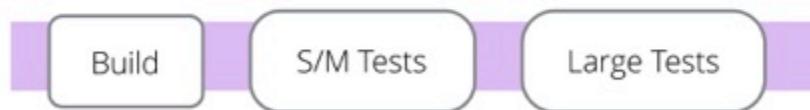




Customer Service





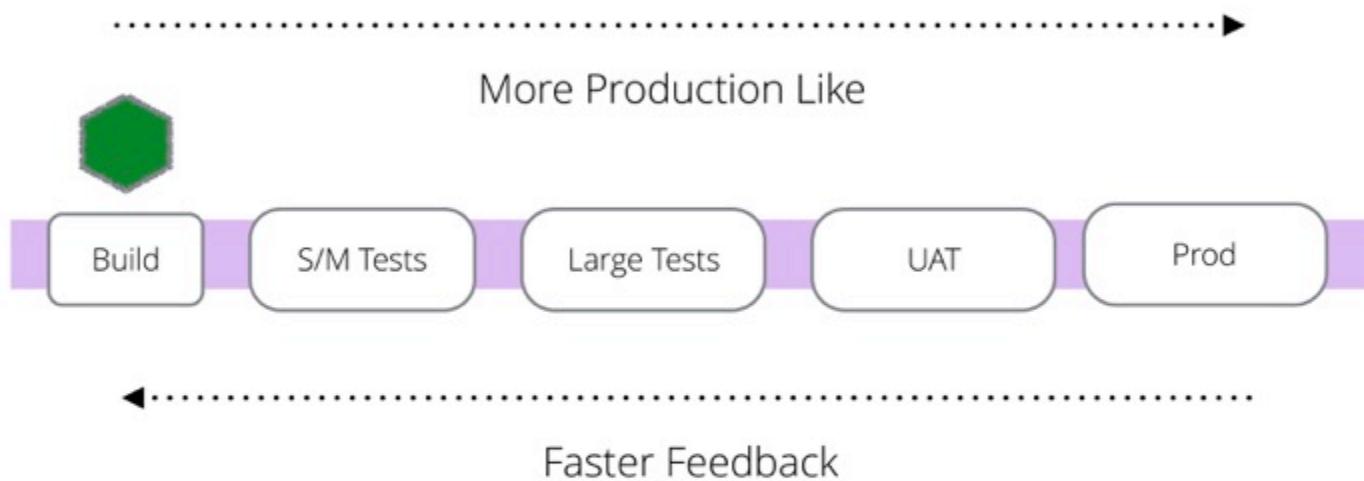


More Production Like

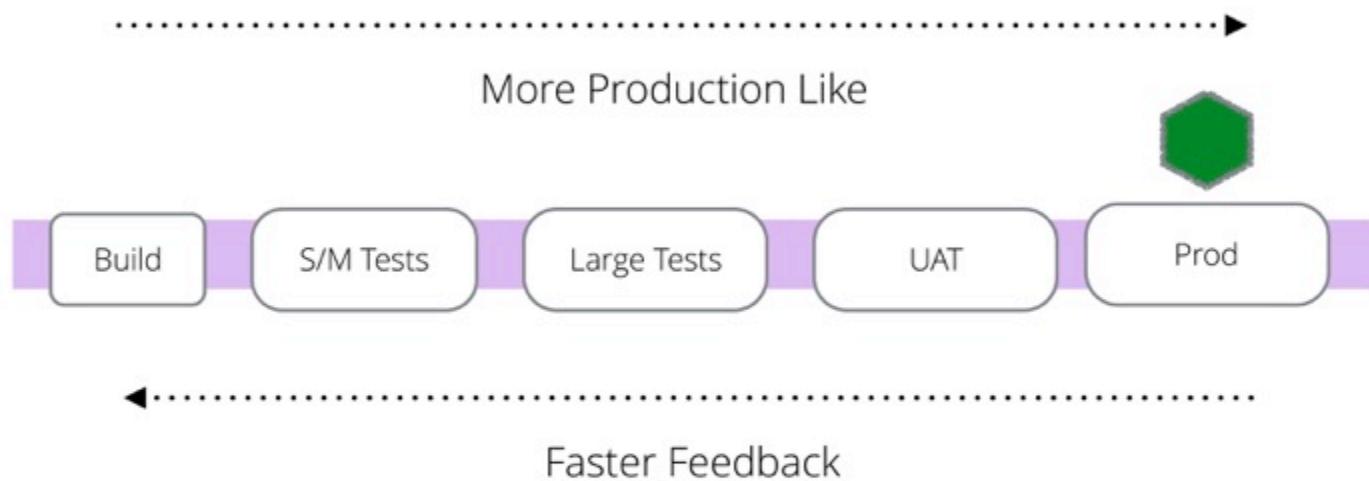


Faster Feedback

Artifacts should be built once, and moved through environments



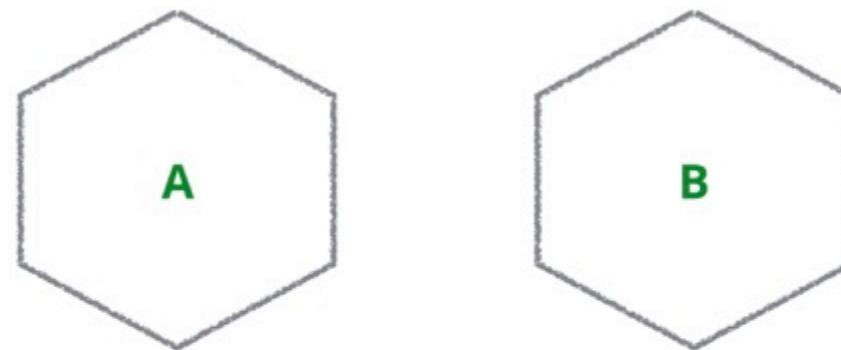
Artifacts should be built once, and moved through environments



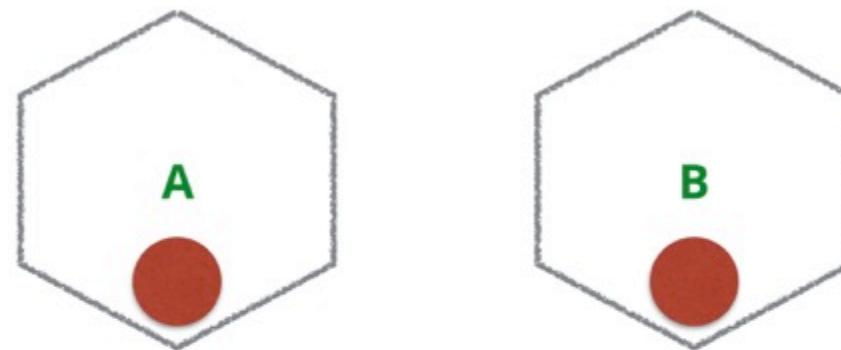
Artifacts should be built once, and moved through environments

# Shared Code?

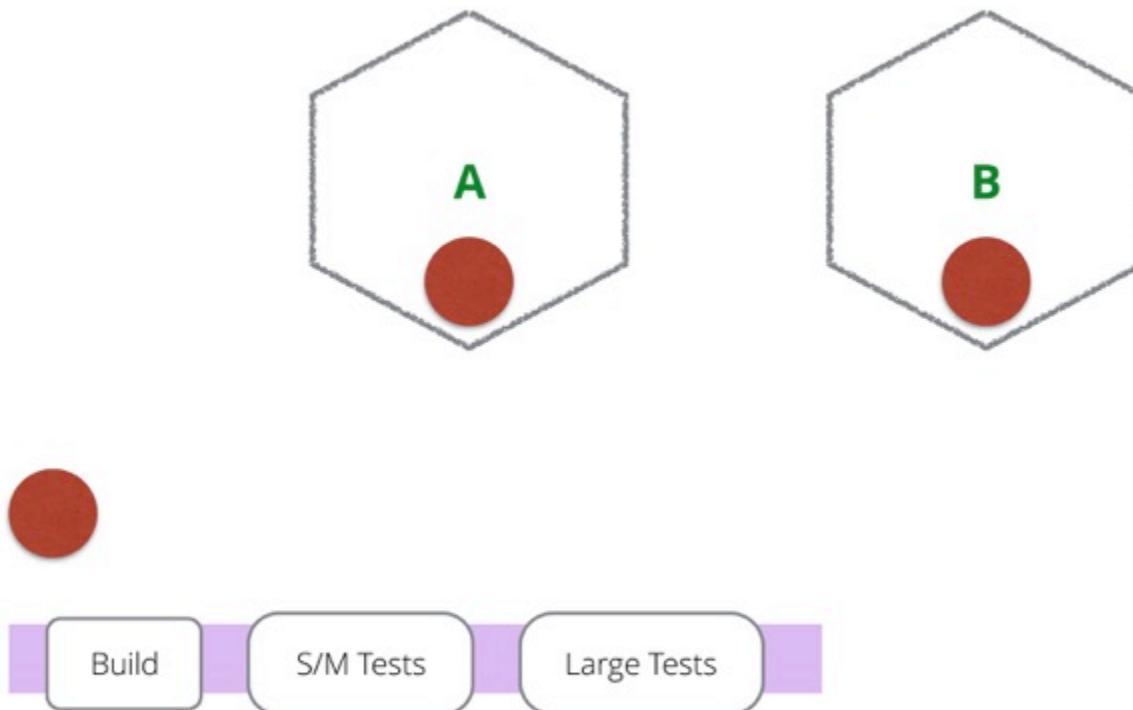
## SHARED DEPENDENCIES



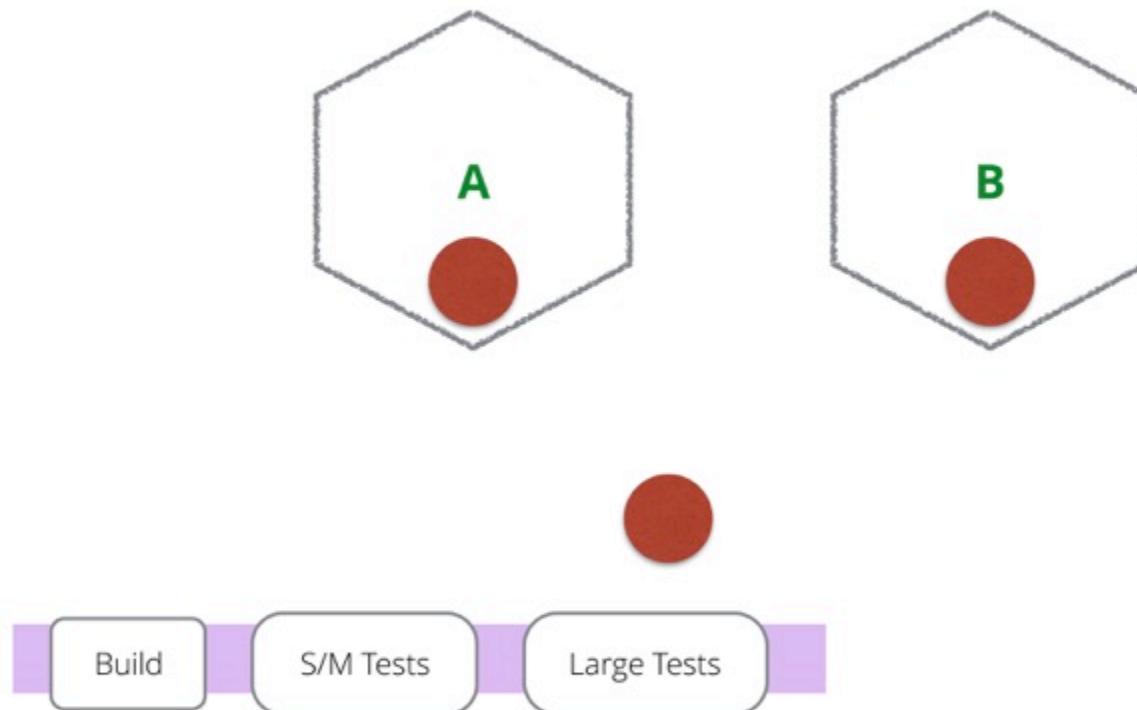
## SHARED DEPENDENCIES



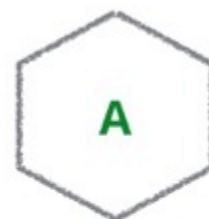
## SHARED DEPENDENCIES



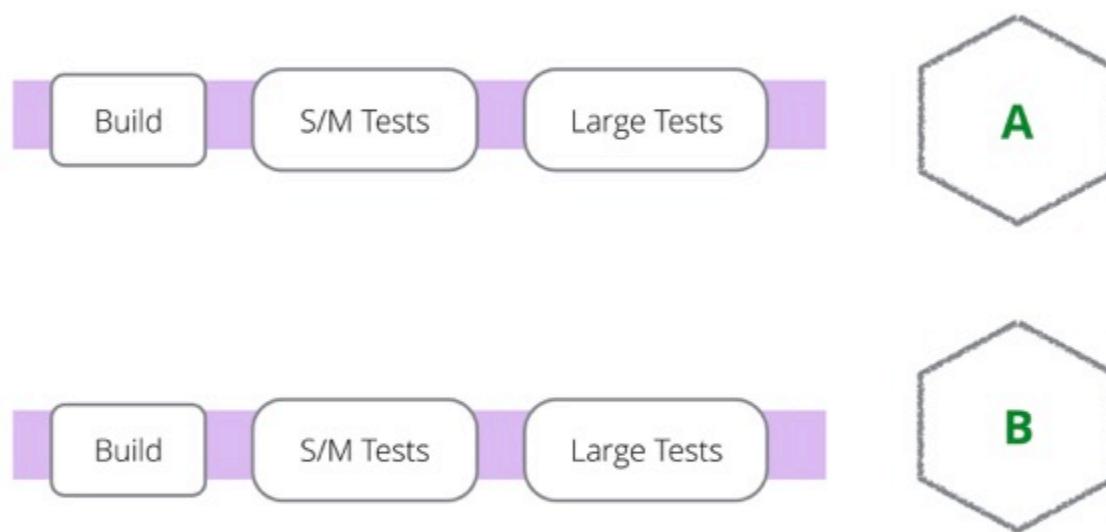
## SHARED DEPENDENCIES



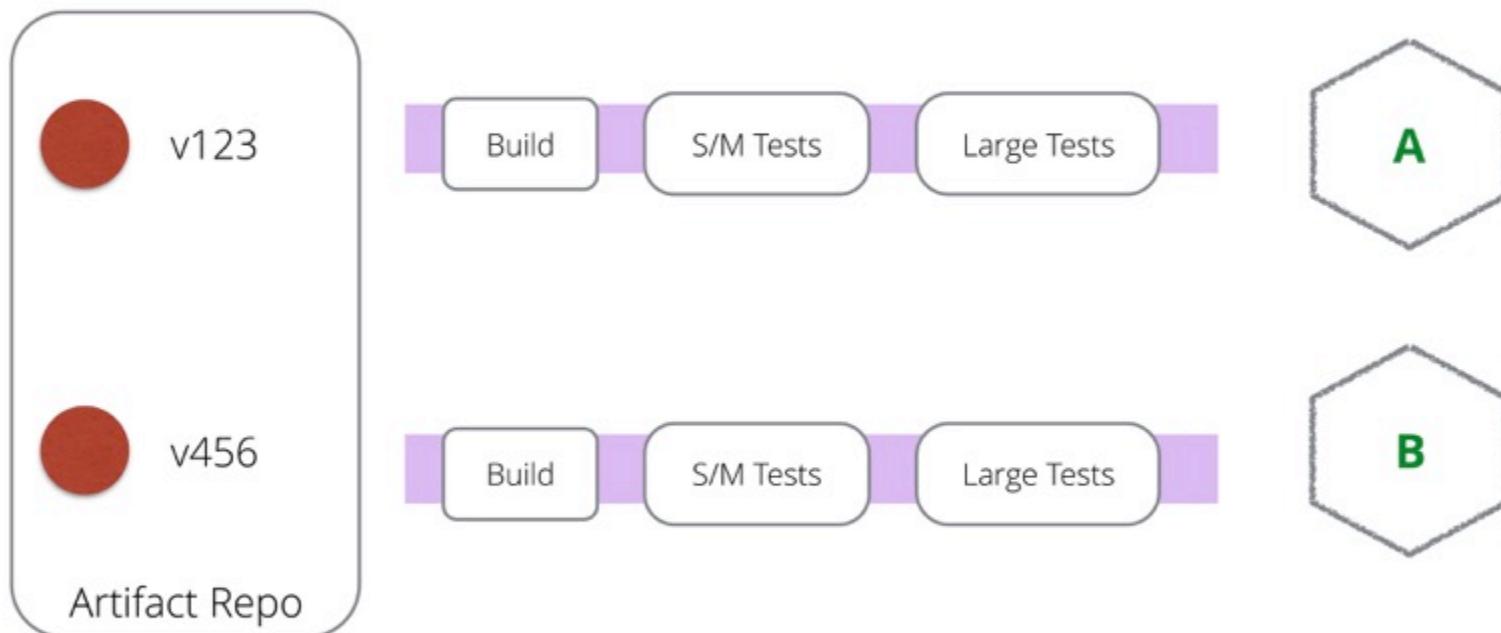
## FIXED DEPENDENCIES



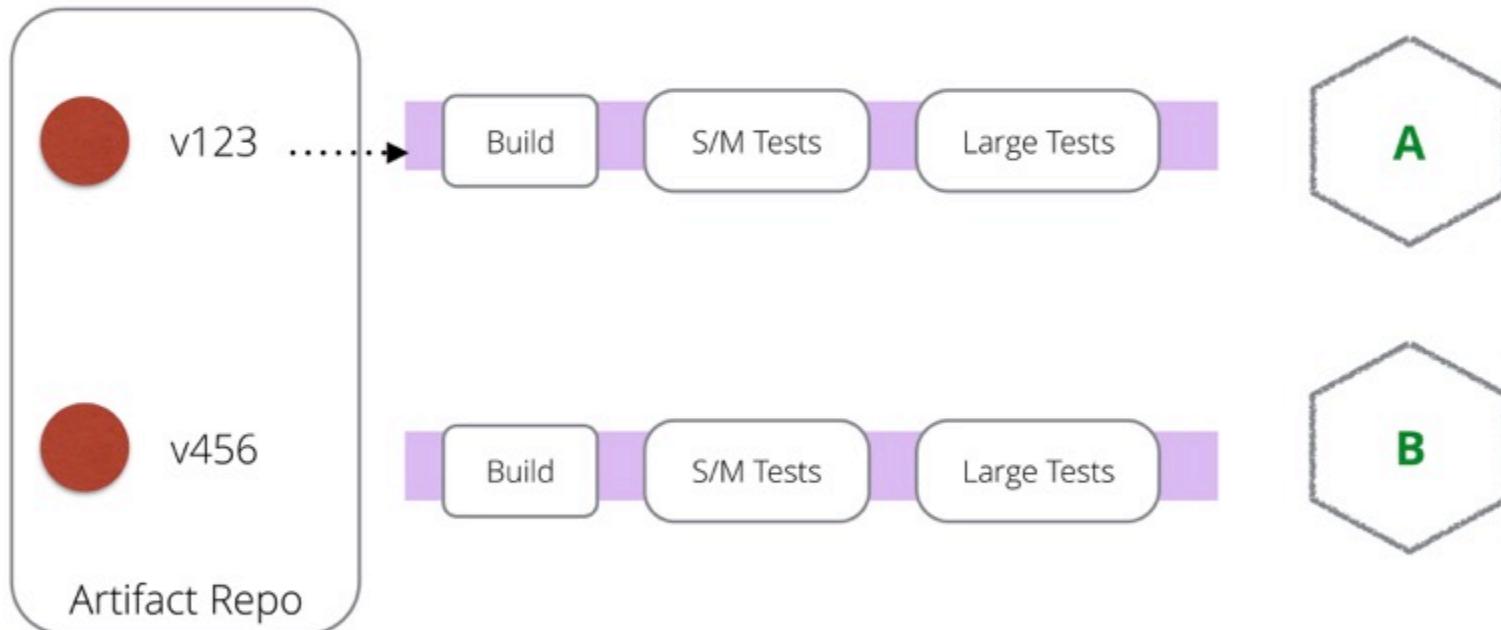
## FIXED DEPENDENCIES



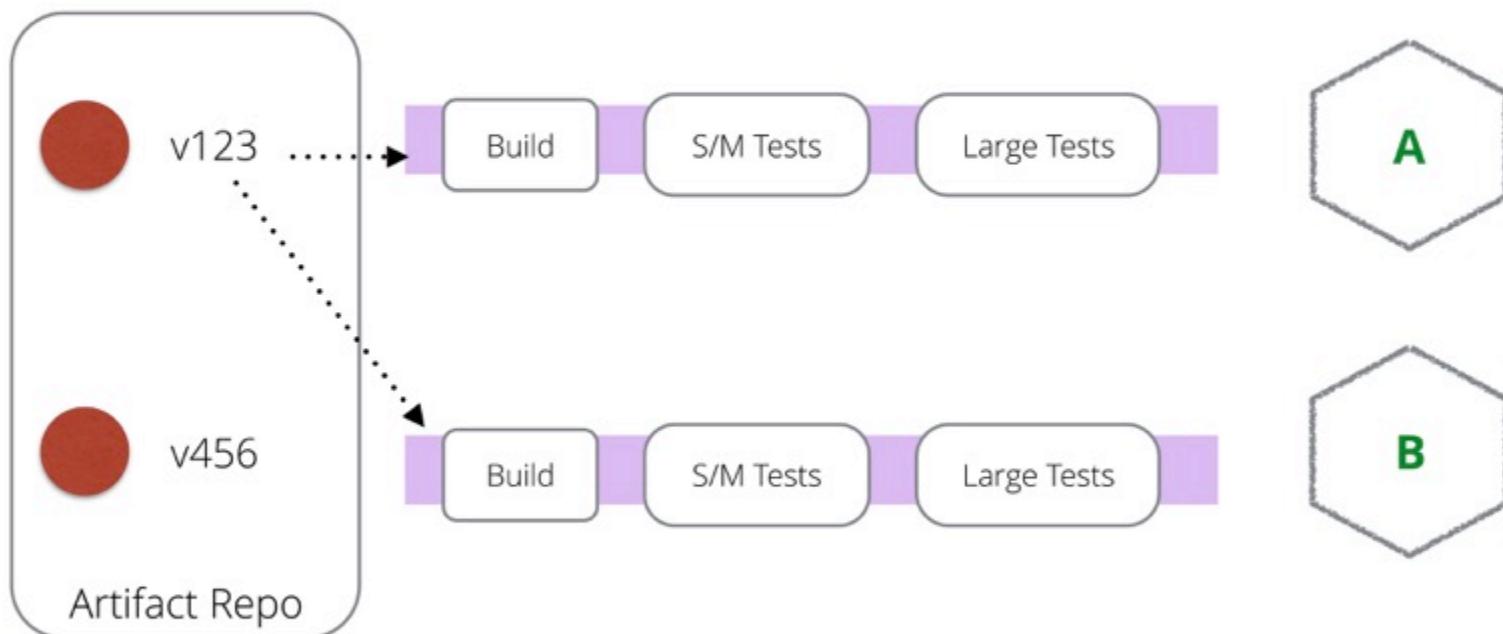
## FIXED DEPENDENCIES



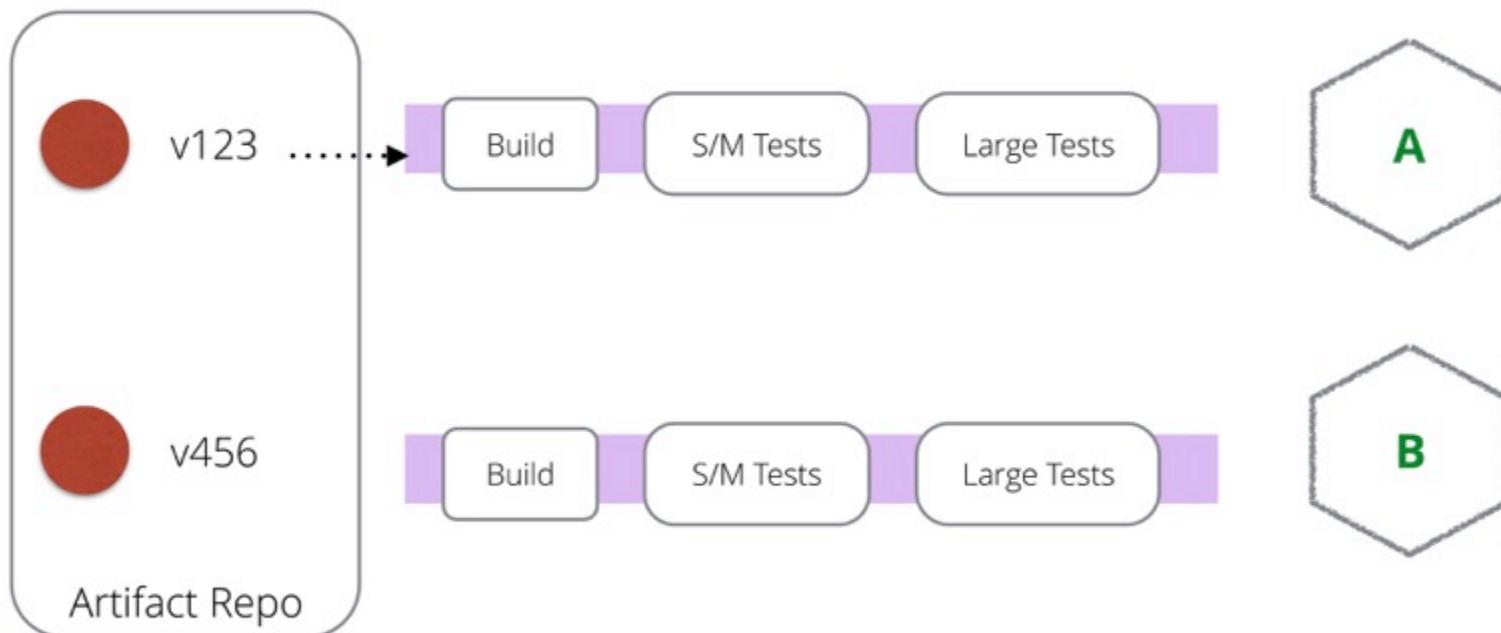
## FIXED DEPENDENCIES



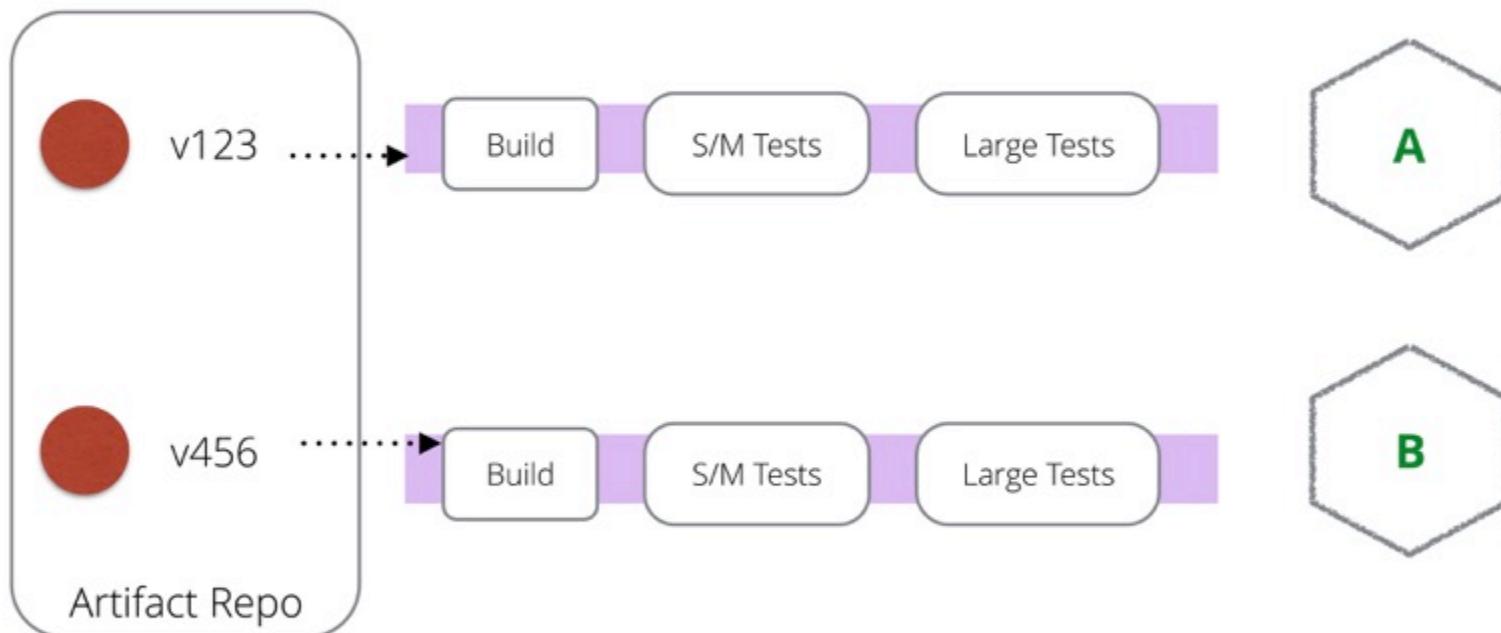
## FIXED DEPENDENCIES



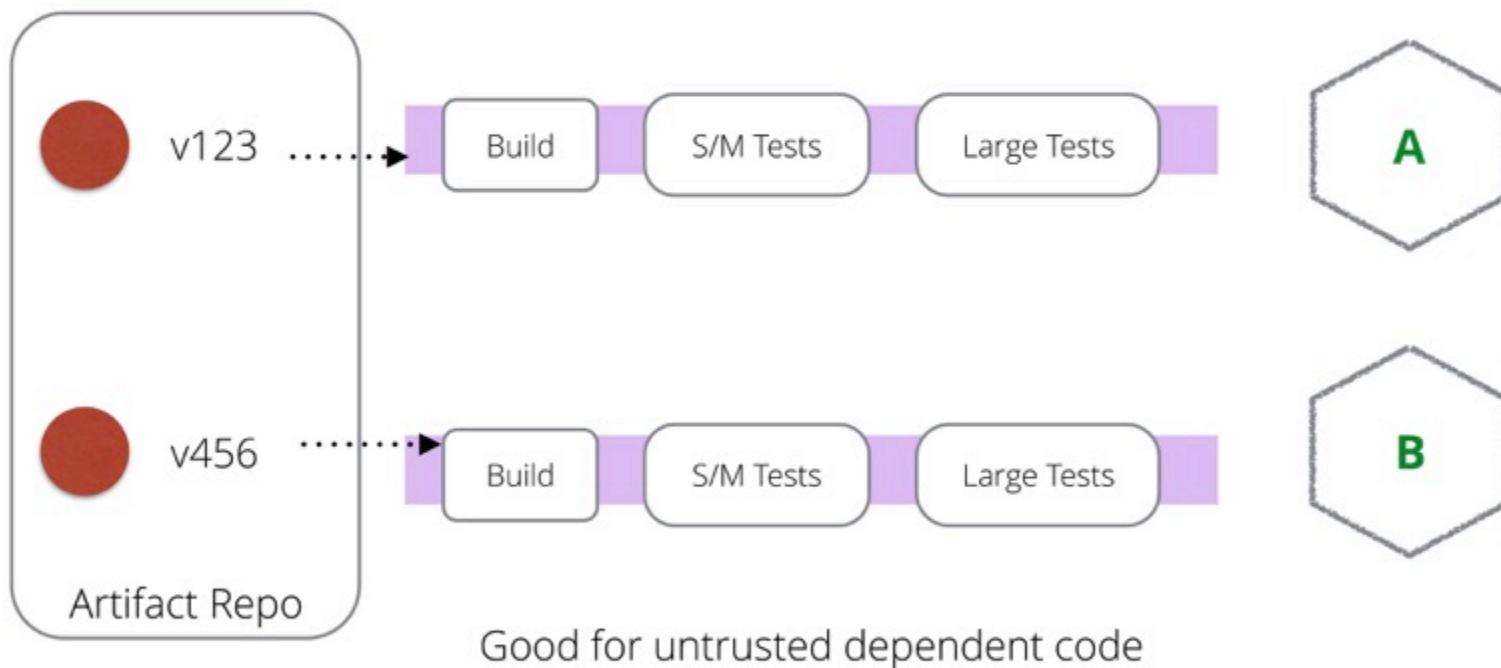
## FIXED DEPENDENCIES



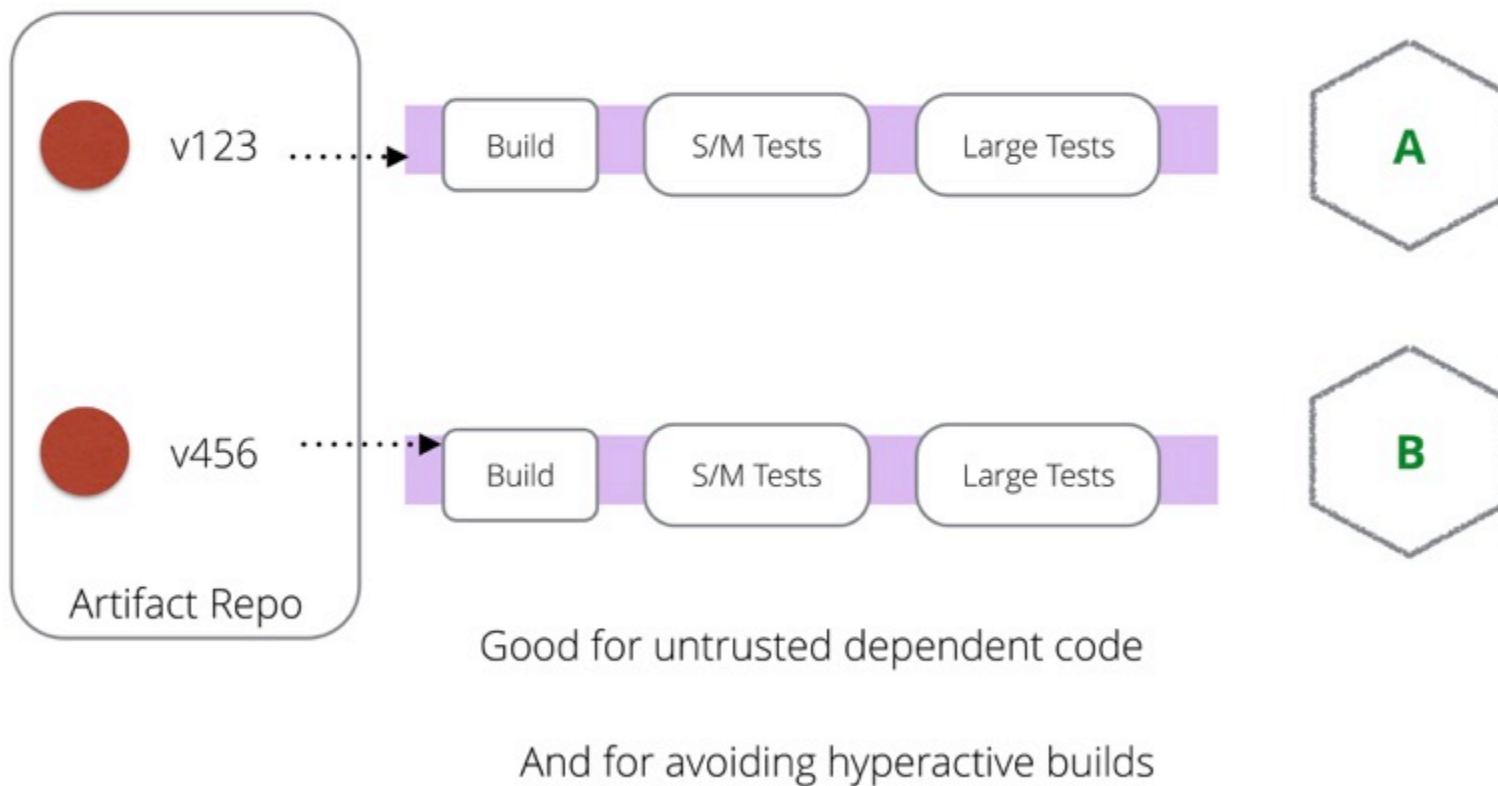
## FIXED DEPENDENCIES



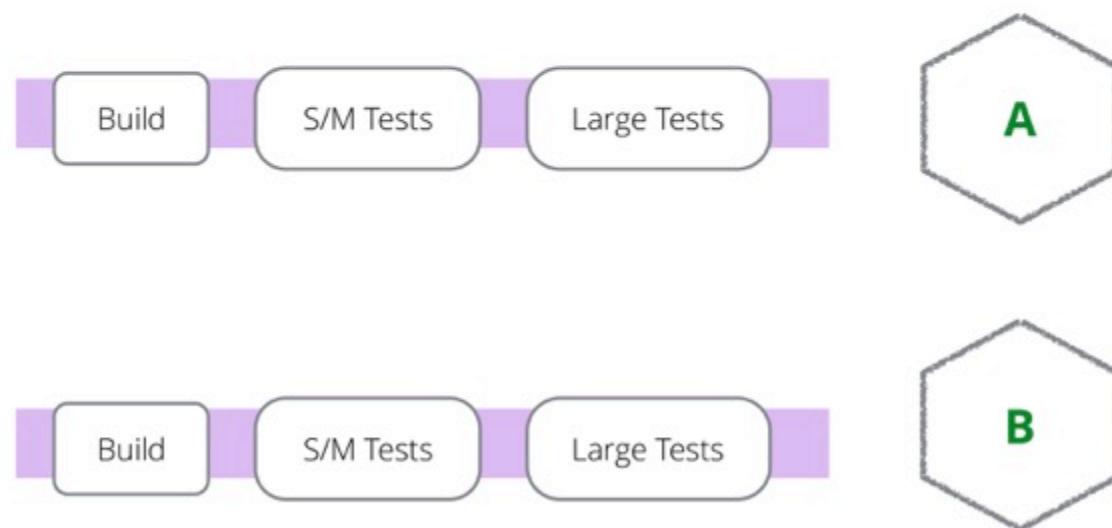
## FIXED DEPENDENCIES



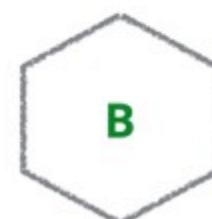
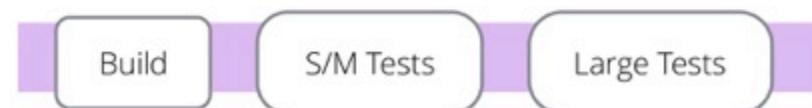
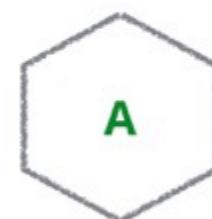
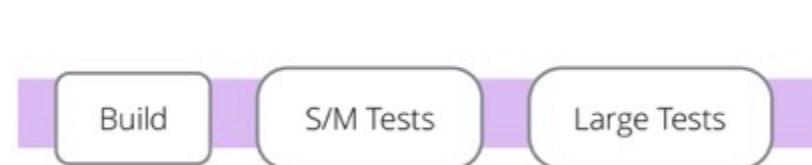
## FIXED DEPENDENCIES



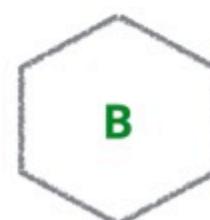
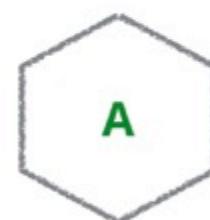
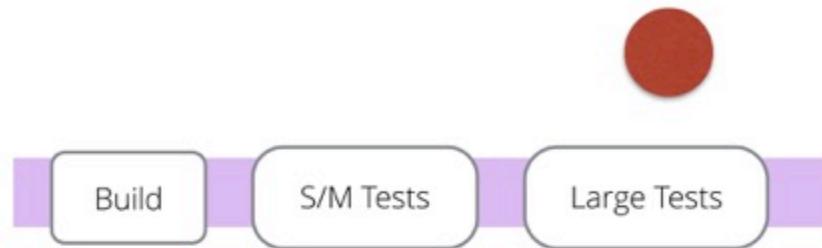
## FLUID DEPENDENCIES



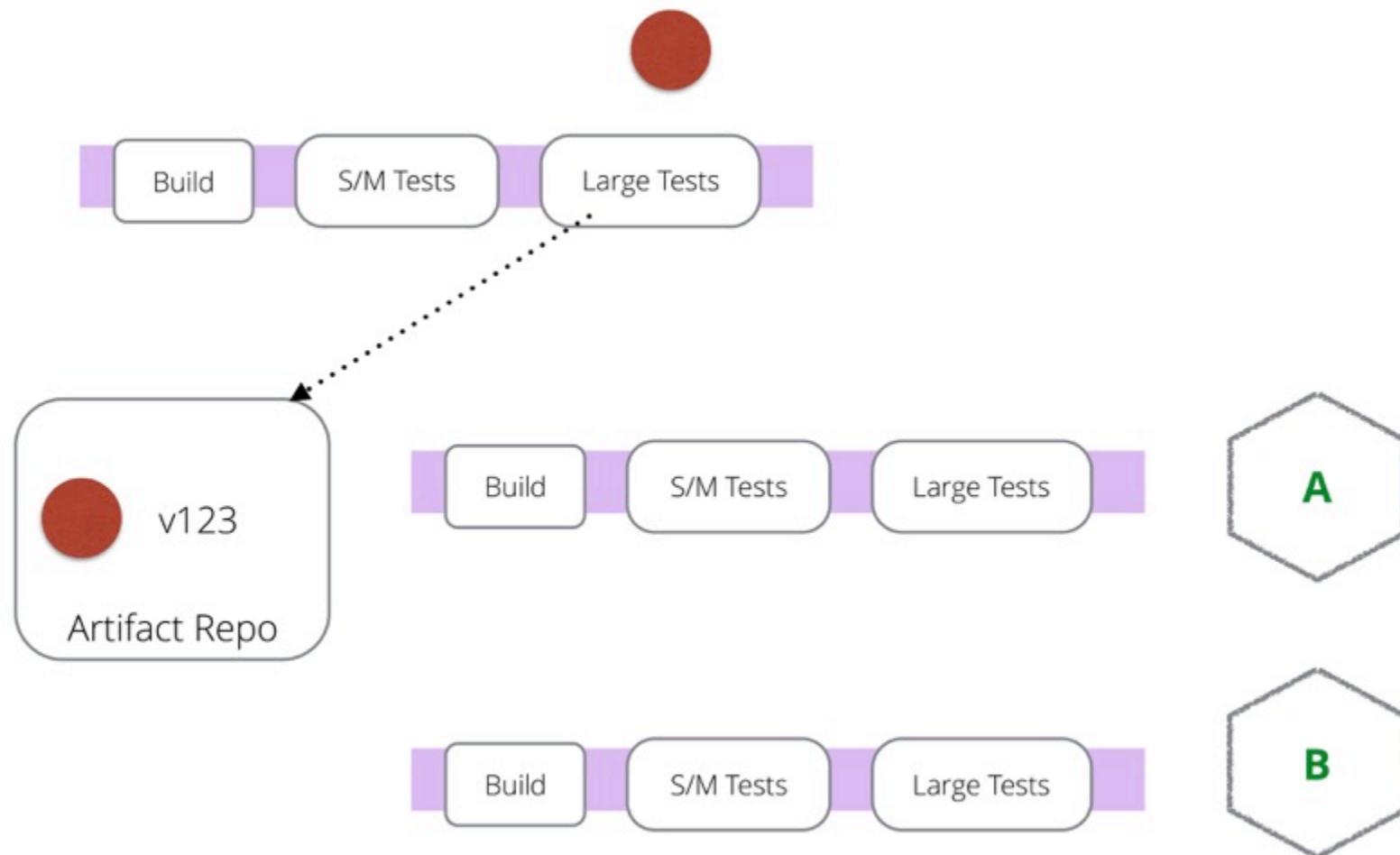
## FLUID DEPENDENCIES



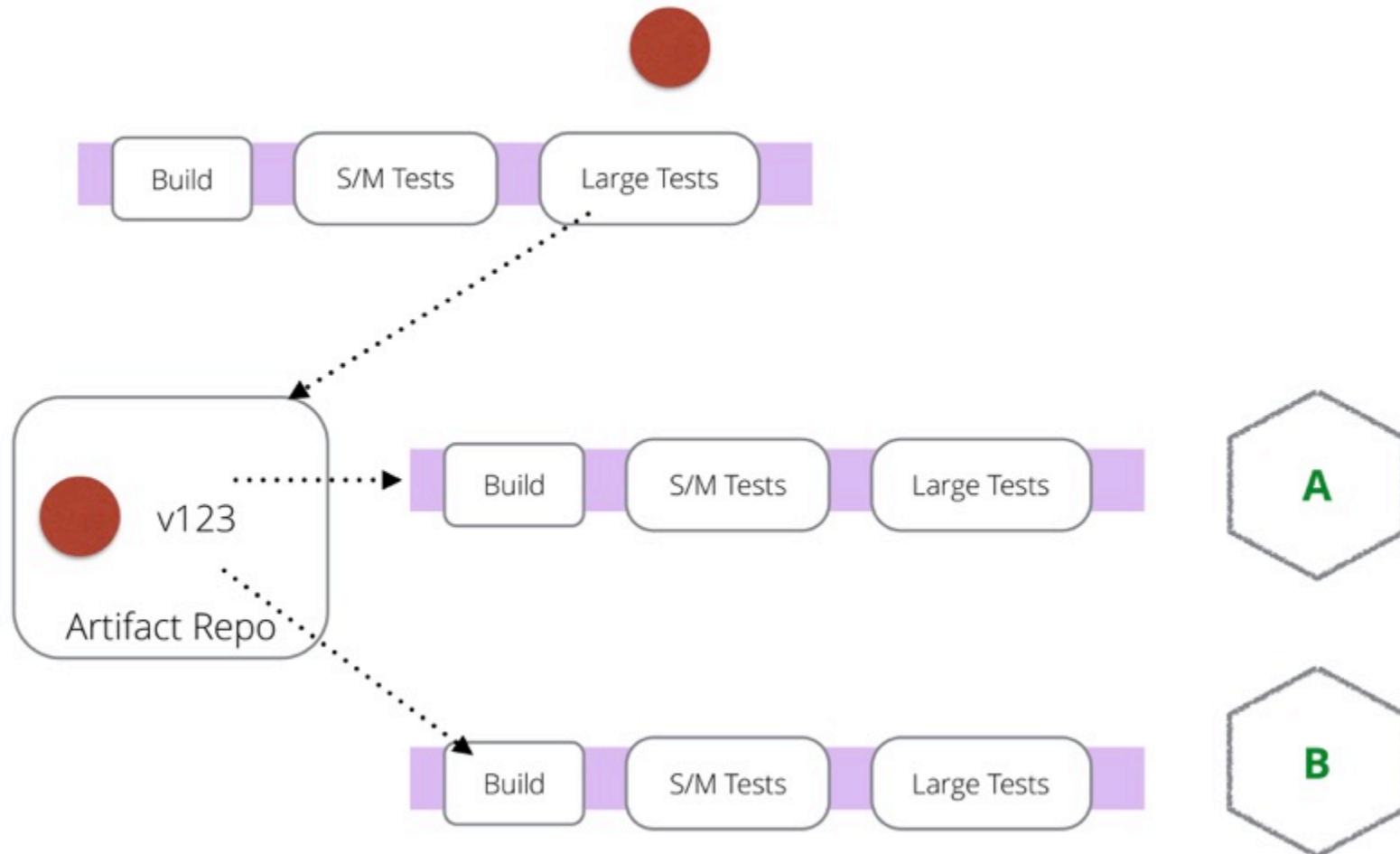
## FLUID DEPENDENCIES



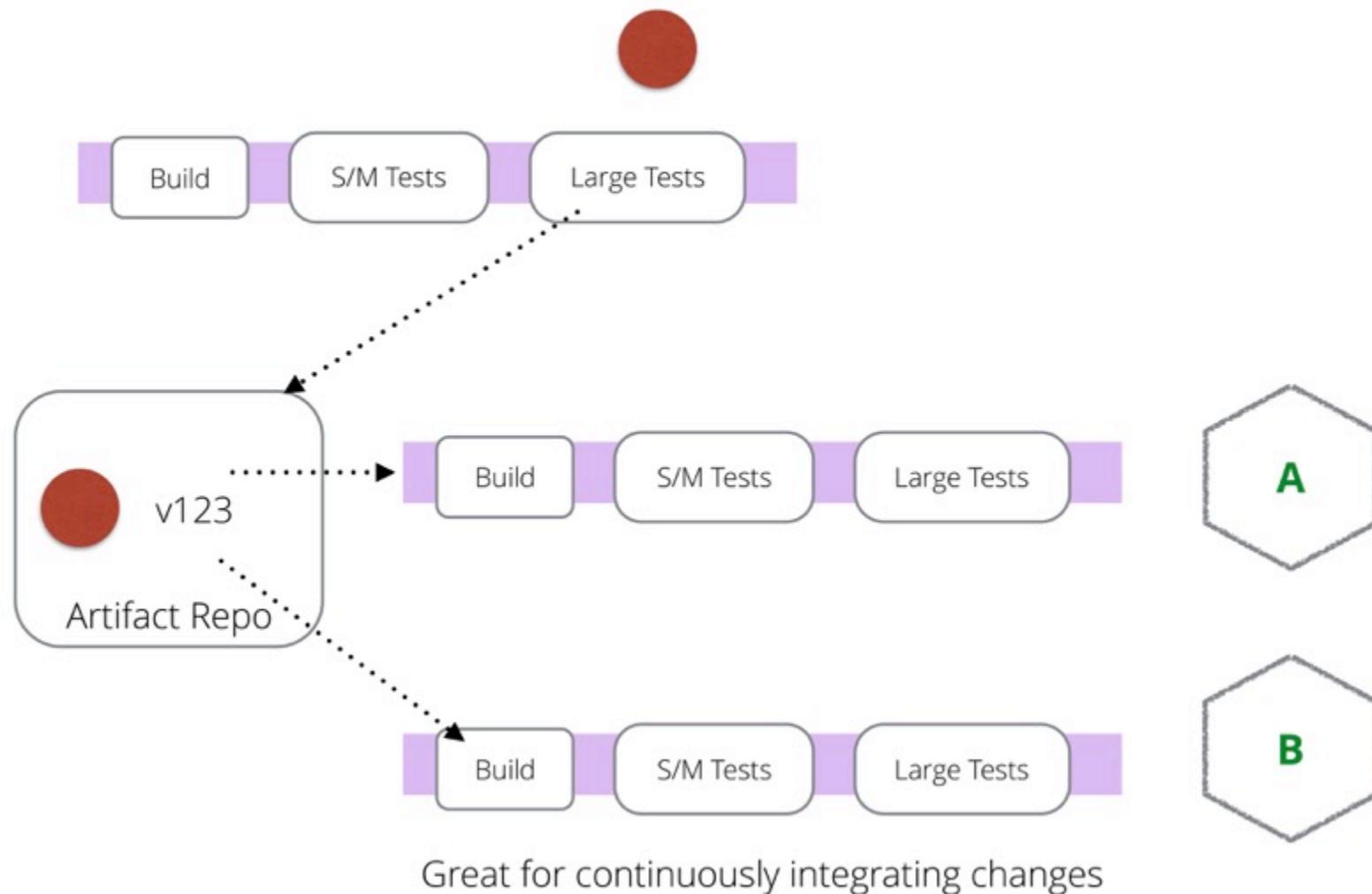
## FLUID DEPENDENCIES



## FLUID DEPENDENCIES



## FLUID DEPENDENCIES













Can we still communicate?

# Independent Deployability

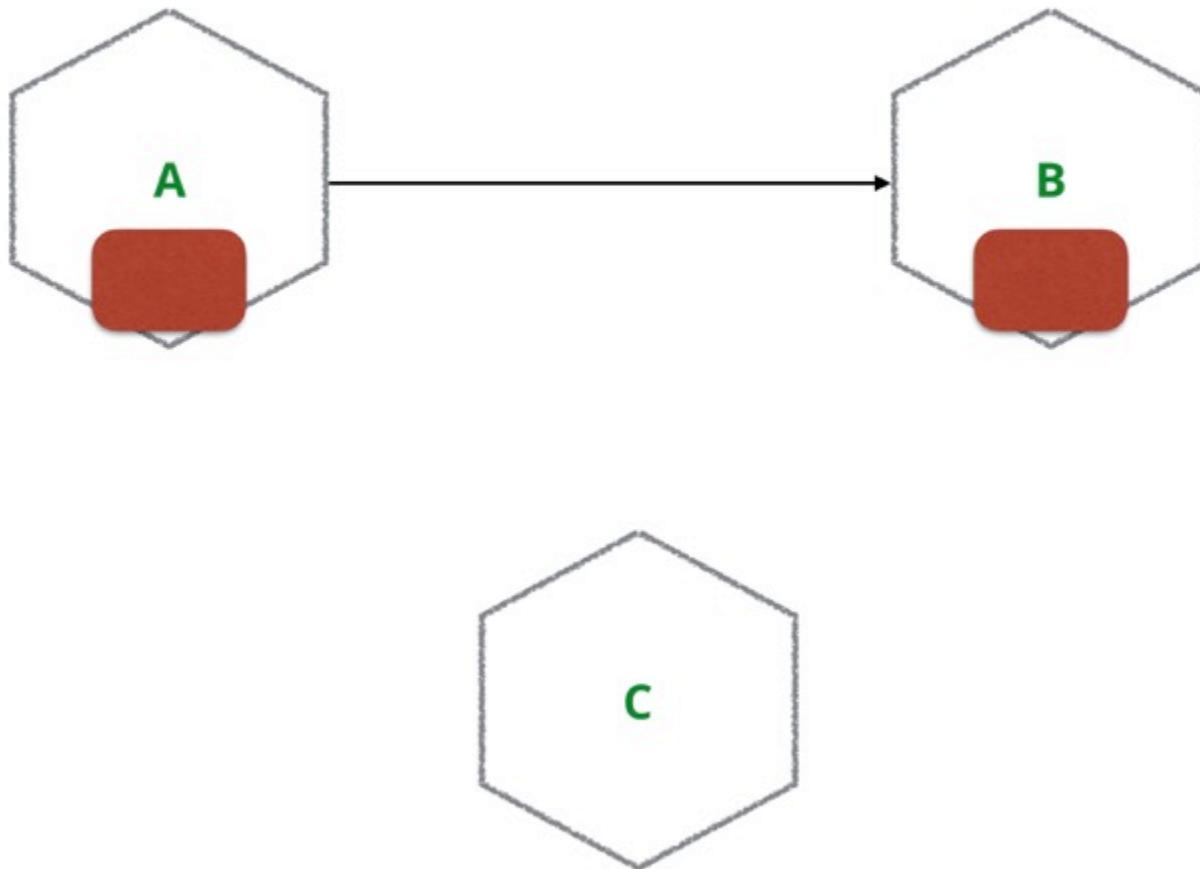
## Independent Deployability

Avoid shared code if it leads to  
the need for lock-step release

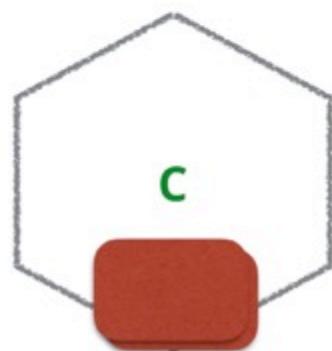
Consider service reuse rather  
than library reuse



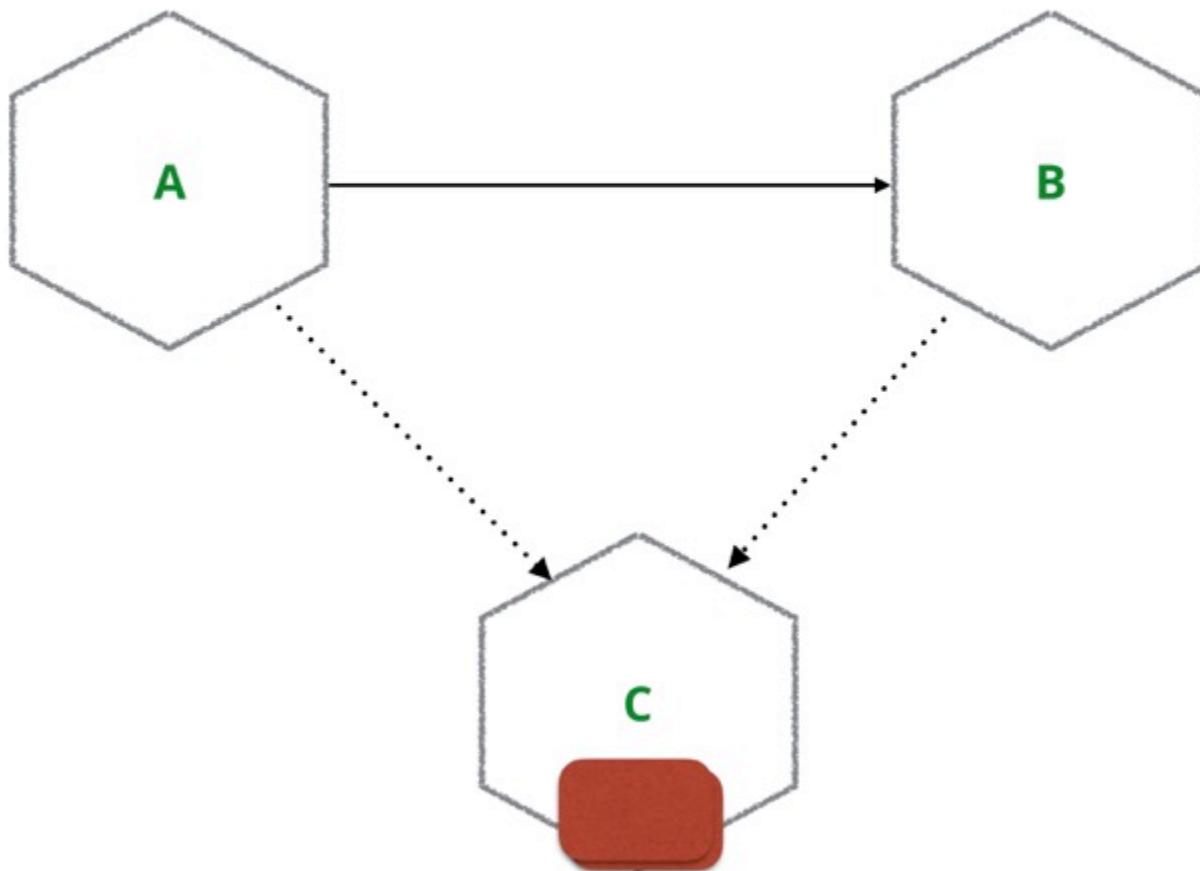
Consider service reuse rather than library reuse



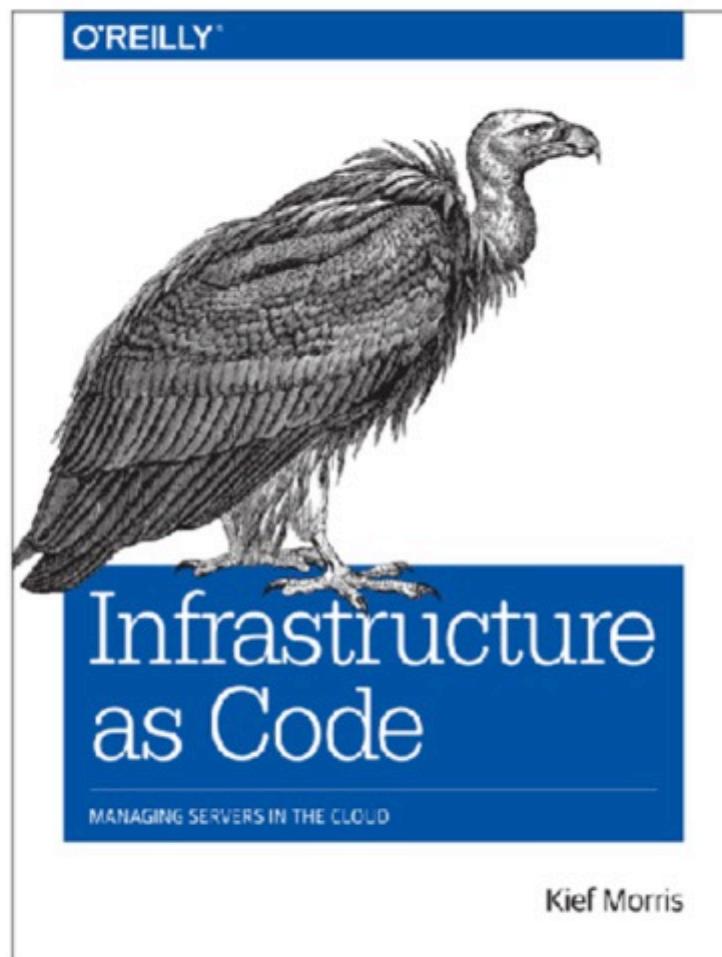
Consider service reuse rather  
than library reuse



Consider service reuse rather than library reuse



# INFRASTRUCTURE AS CODE



<http://infrastructure-as-code.com/>

## DECLARATIVE HOST CONFIGURATION

```
---
- name: Install Java 1.7
  yum: name=java-1.7.0-openjdk state=present

- name: add group "tomcat"
  group: name=tomcat

- name: add user "tomcat"
  user: name=tomcat group=tomcat home=/usr/share/tomcat createhome=no
  sudo: True

- name: Download Tomcat
  get_url: url=http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.61/bin/apache-tomca
  ...

- name: Extract archive
  command: chdir=/usr/share /bin/tar xvf /opt/apache-tomcat-7.0.61.tar.gz -C /opt/ cre
  ...

- name: Symlink install directory
  file: src=/opt/apache-tomcat-7.0.61 path=/usr/share/tomcat state=link

- name: Change ownership of Tomcat installation
  file: path=/usr/share/tomcat/ owner=tomcat group=tomcat state=directory recurse=yes
```



CHEF™



CHEF™



puppet



CHEF™



puppet



ANSIBLE

# DEPLOYMENT

Artifact Store



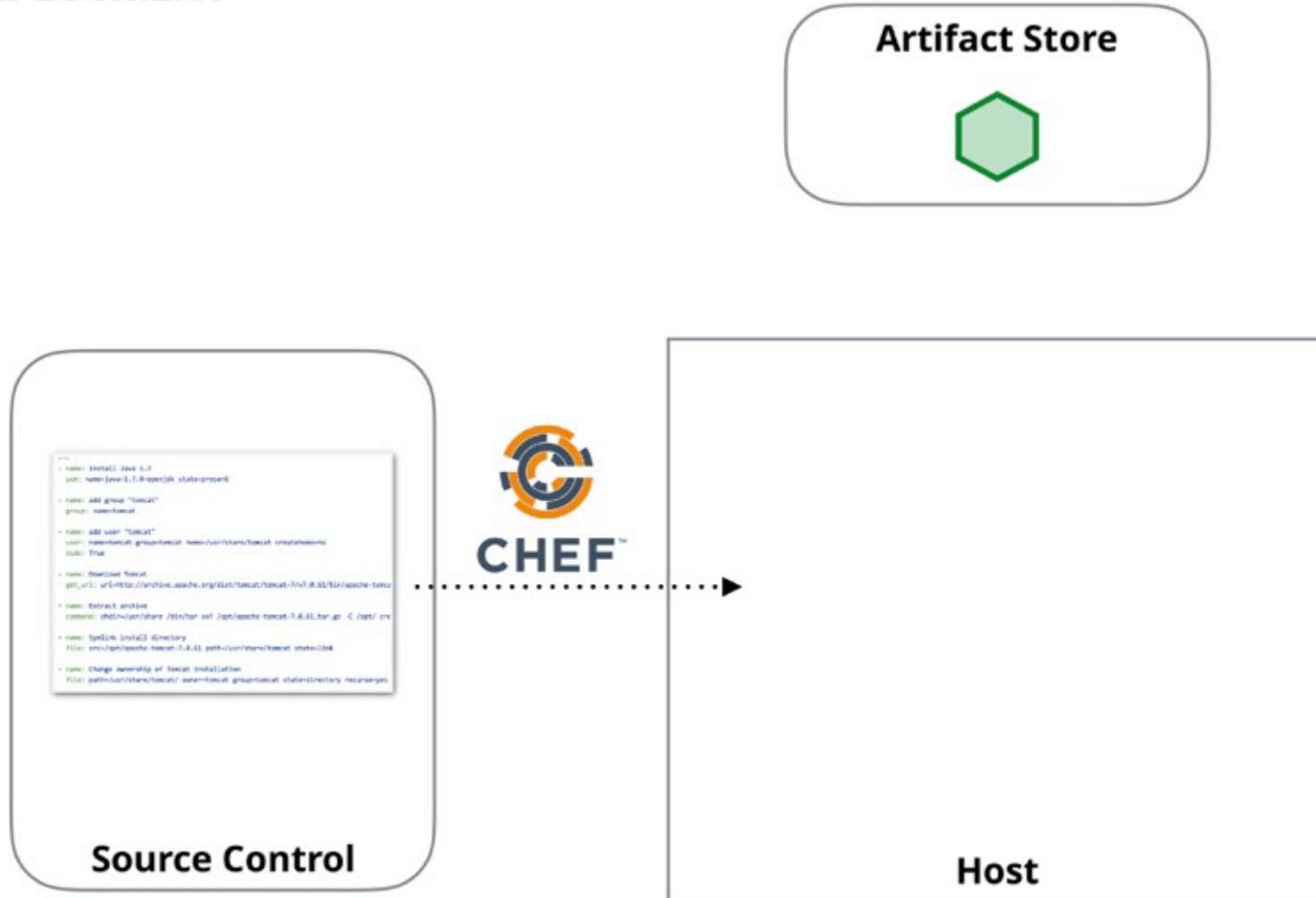
```
...
+ name: install java-1.7
  puts: name[java-1.7]; openjdk statement
+ name: add group "tomcat"
  group: tomcat
+ name: add user "tomcat"
  user: tomcat; grouptomcat; home=/var/share/tomcat; creationmode: 0700; shell: /bin/false
+ name: download tomcat
  get:url: url=https://archive.apache.org/dist/tomcat/tomcat-7/v7.0.82/bin/apache-tomcat-7.0.82.tar.gz
+ name: extract archive
  command: chmod+rx /var/share/tomcat /opt/apache-tomcat-7.0.82; tar -x -C /opt/ apache-tomcat-7.0.82
+ name: update install directory
  title: /var/share/tomcat-7.0.82; puts: /var/share/tomcat; state: 0
+ name: Change ownership of tomcat installation
  title: path=/var/share/tomcat/; owner=tomcat; group=tomcat; state: directory; recursive: yes
```

Source Control

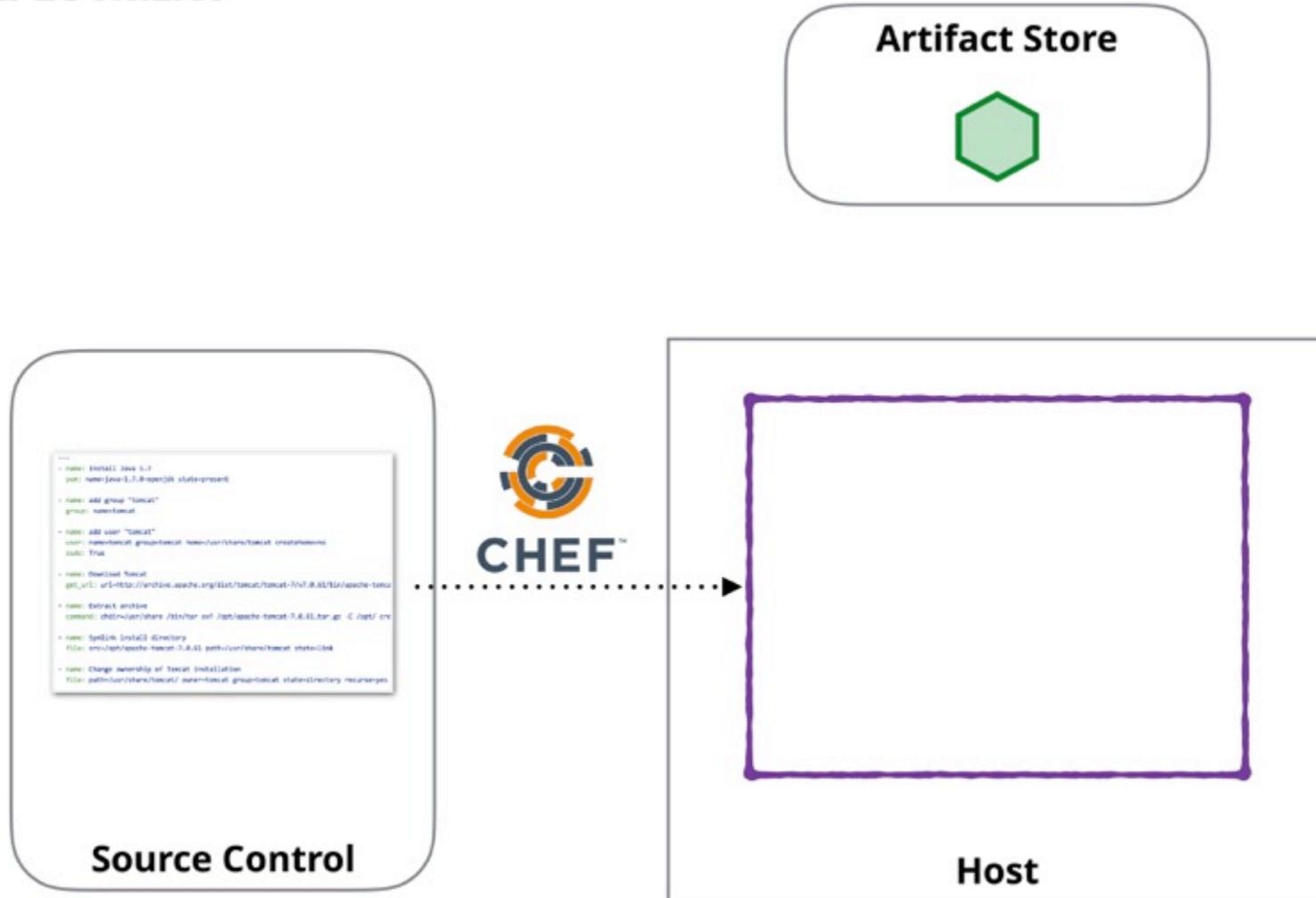


Host

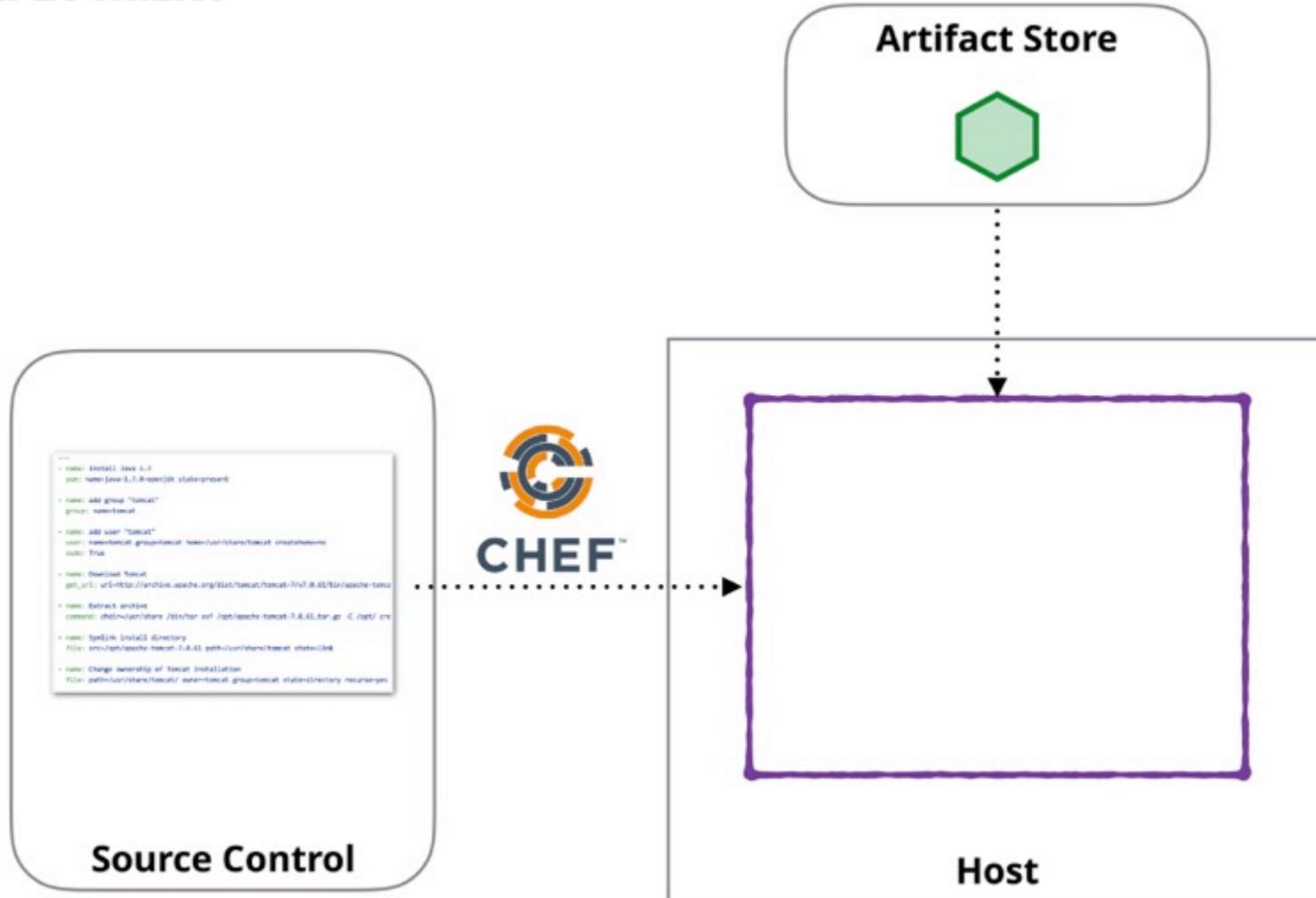
# DEPLOYMENT



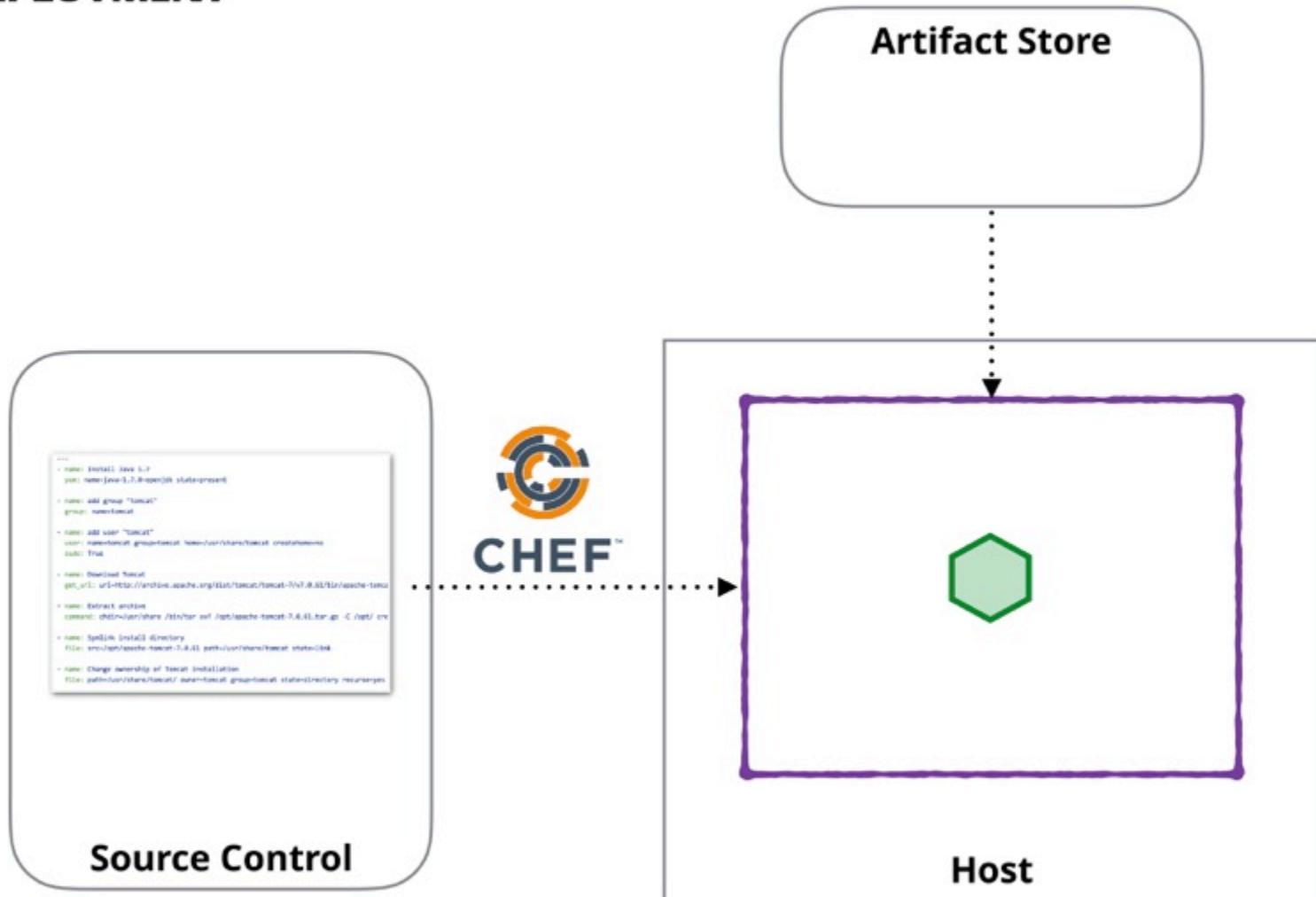
# DEPLOYMENT



# DEPLOYMENT



## **DEPLOYMENT**



## **INFRASTRUCTURE AS CODE**

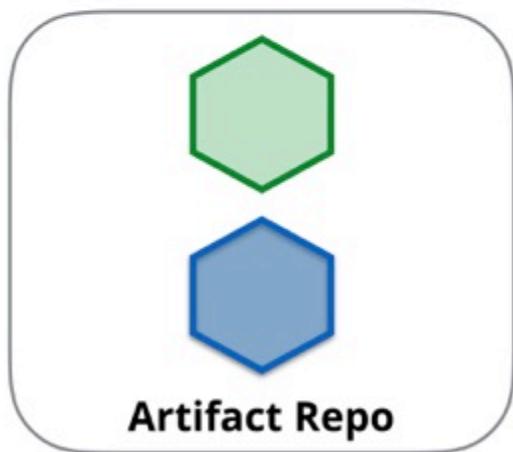
Machine configuration version controlled

Reproduceable

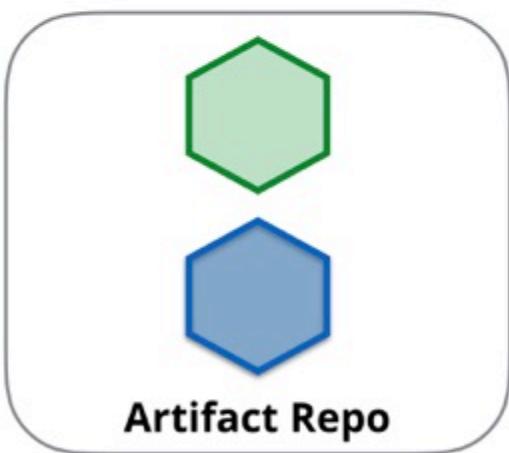
Help rebuild machines

Manage at scale

## CO-EXISTING SERVICES

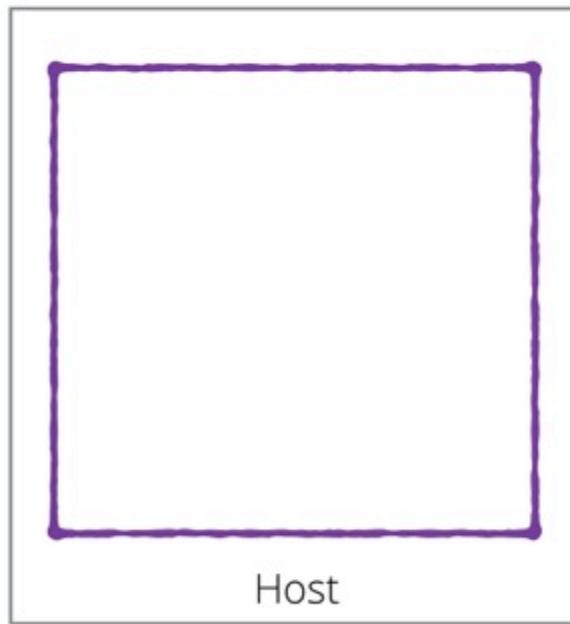
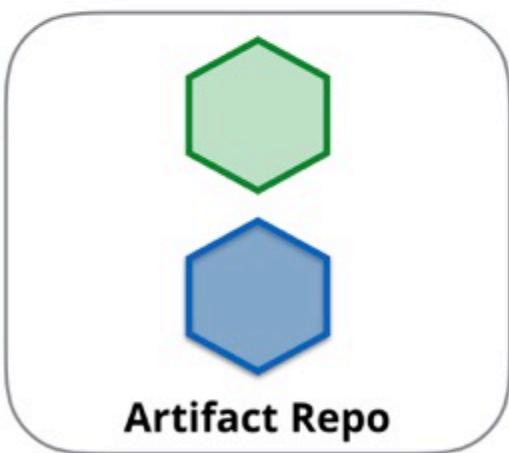


## CO-EXISTING SERVICES

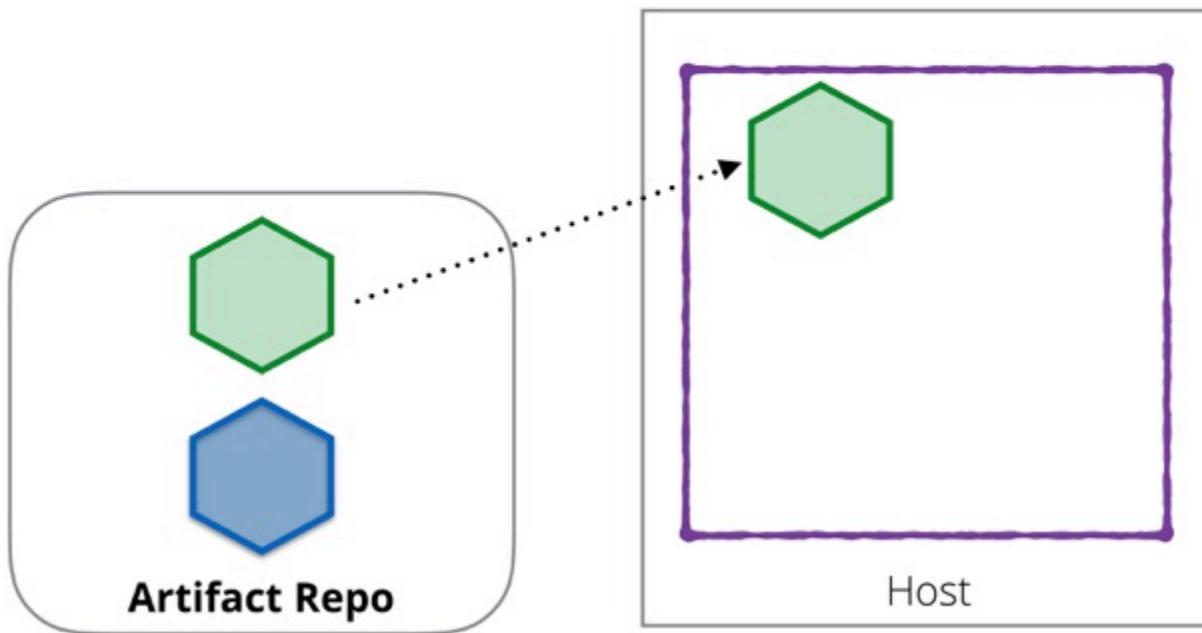


Host

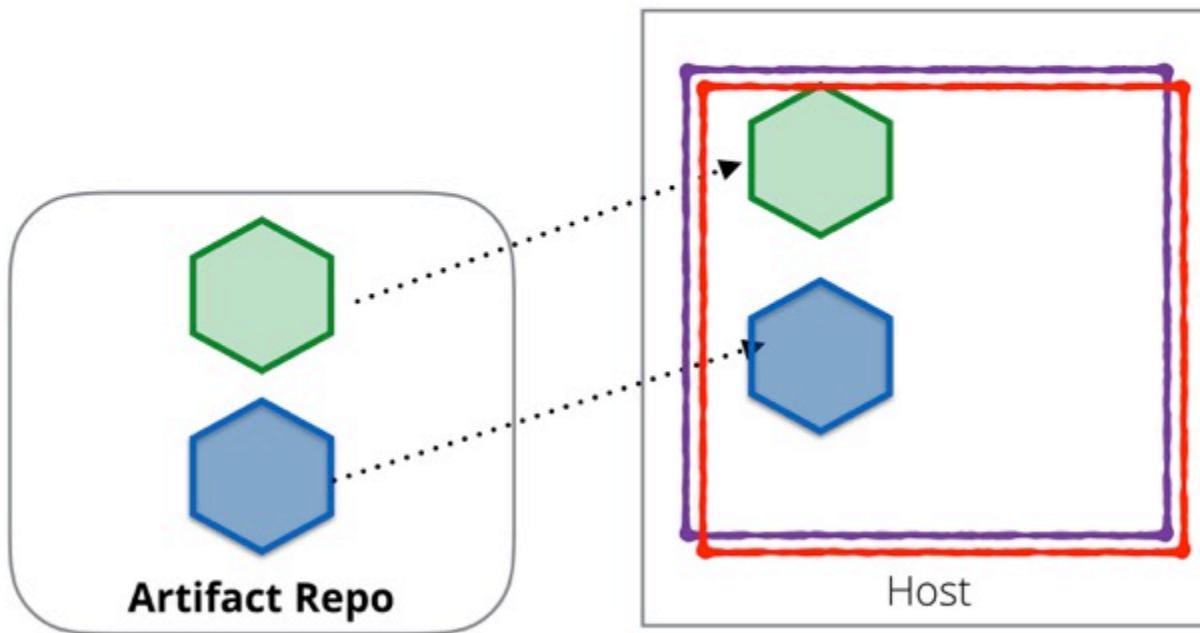
## CO-EXISTING SERVICES



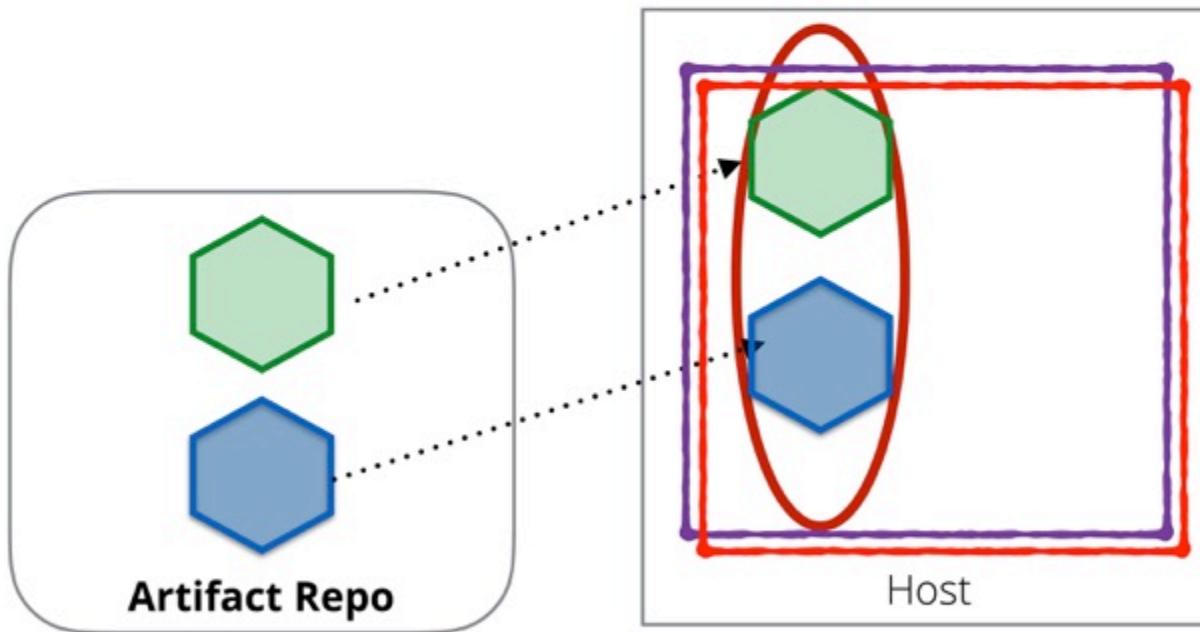
## CO-EXISTING SERVICES

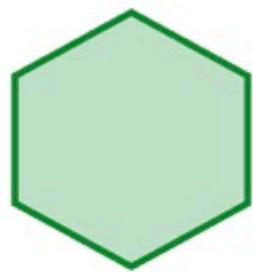


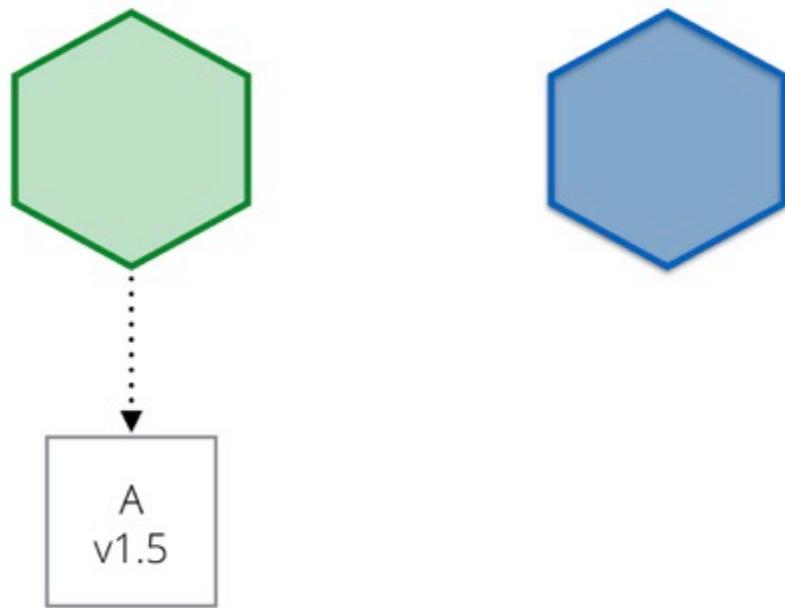
## CO-EXISTING SERVICES

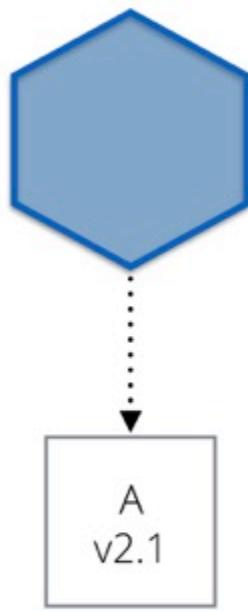
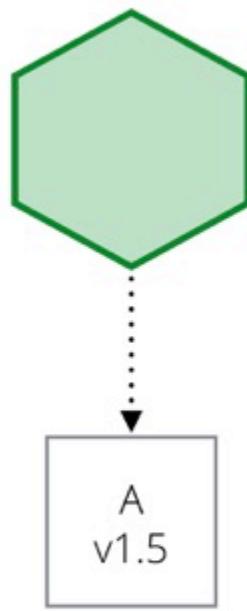


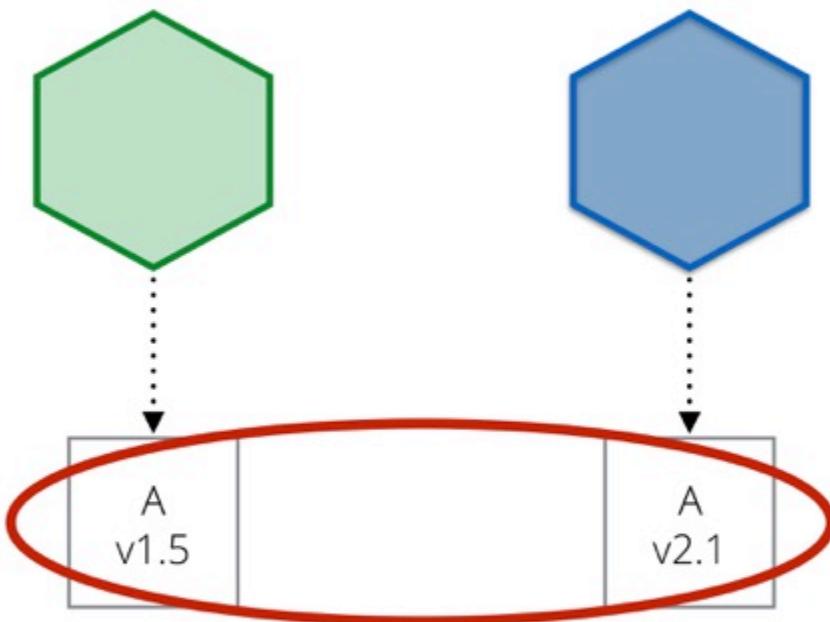
## CO-EXISTING SERVICES

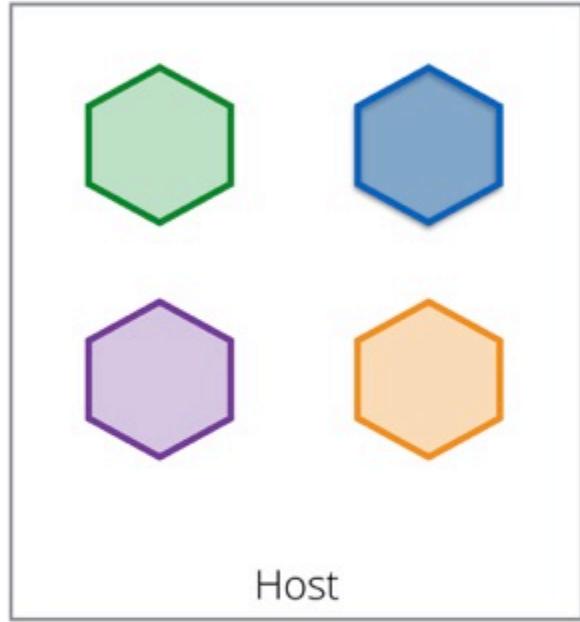


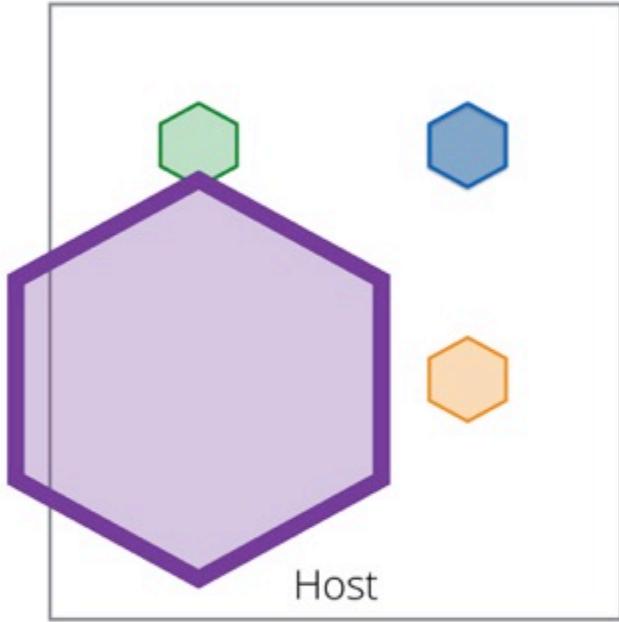


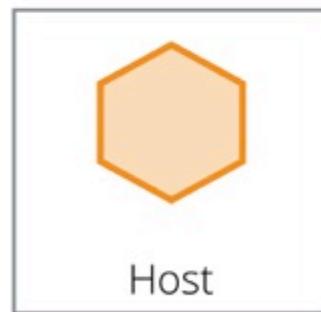
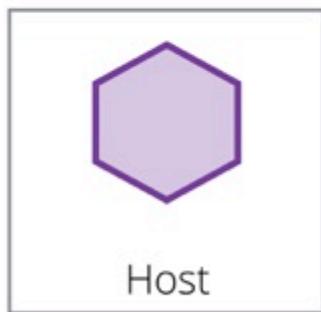
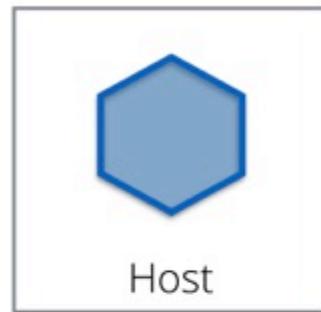
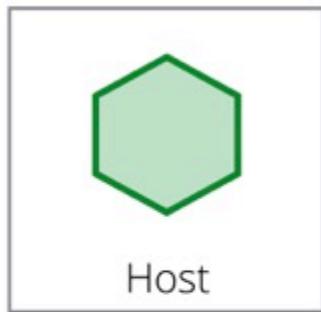


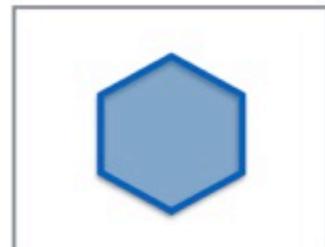
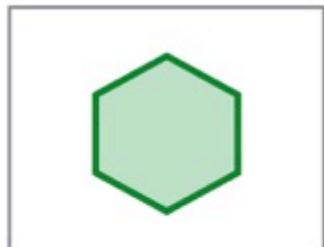








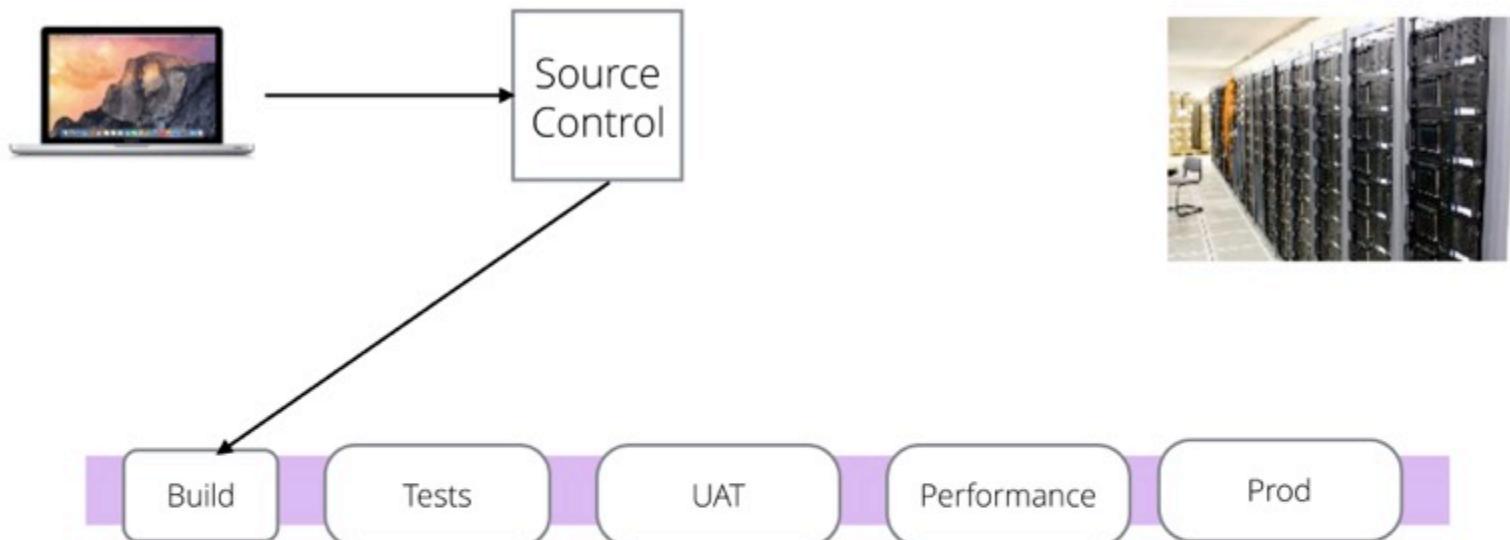


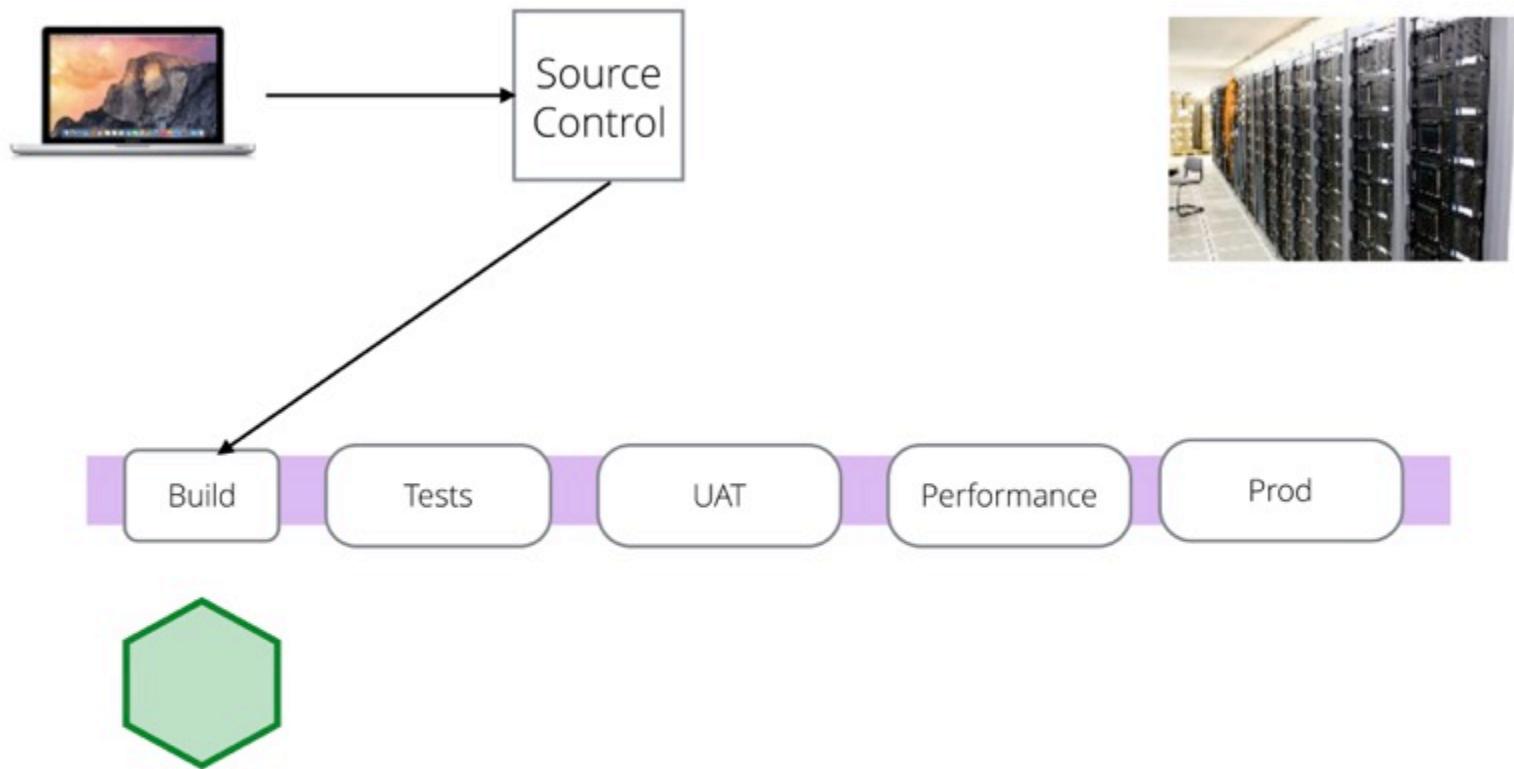


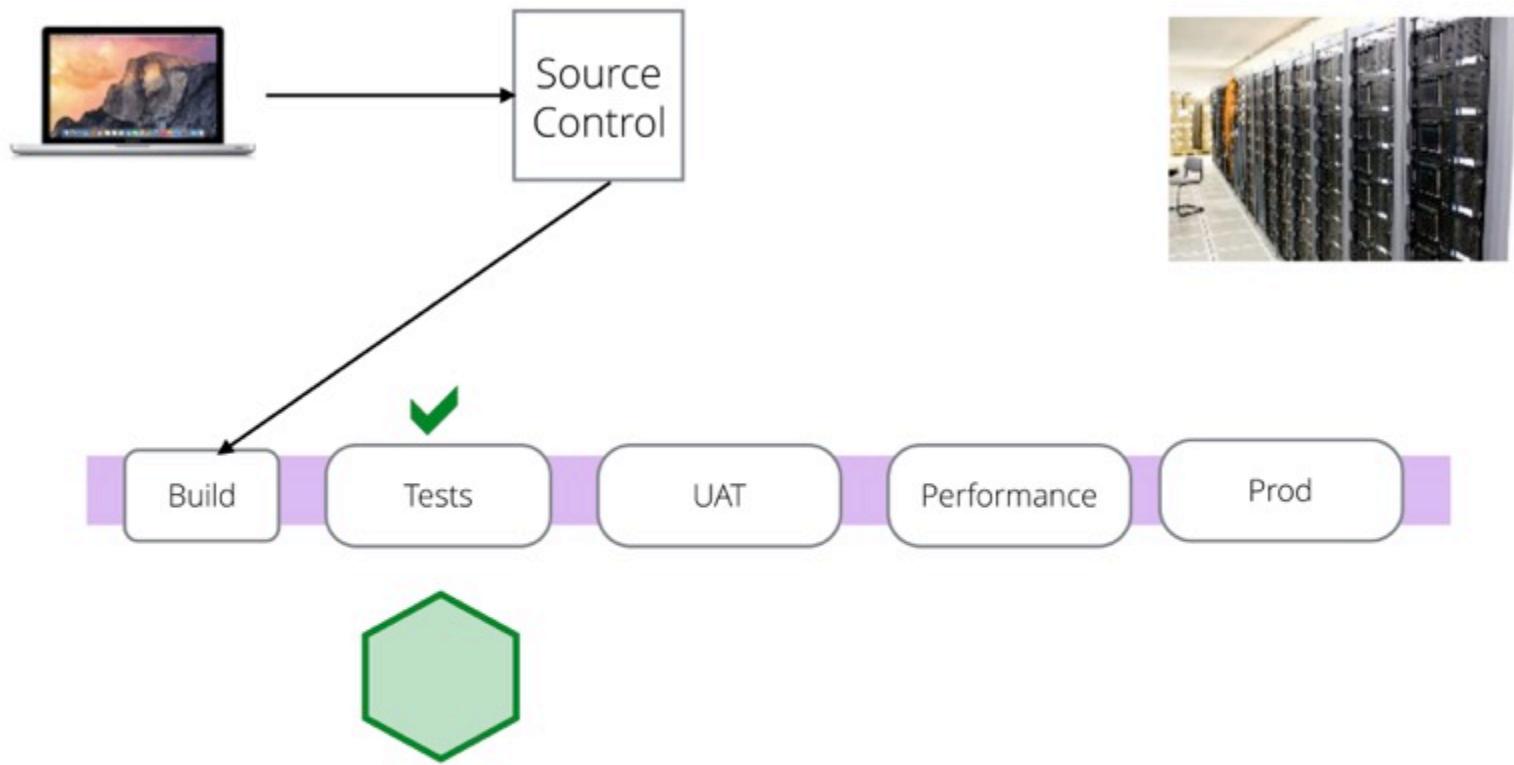
Separate execution environment  
makes independent deployment  
safer

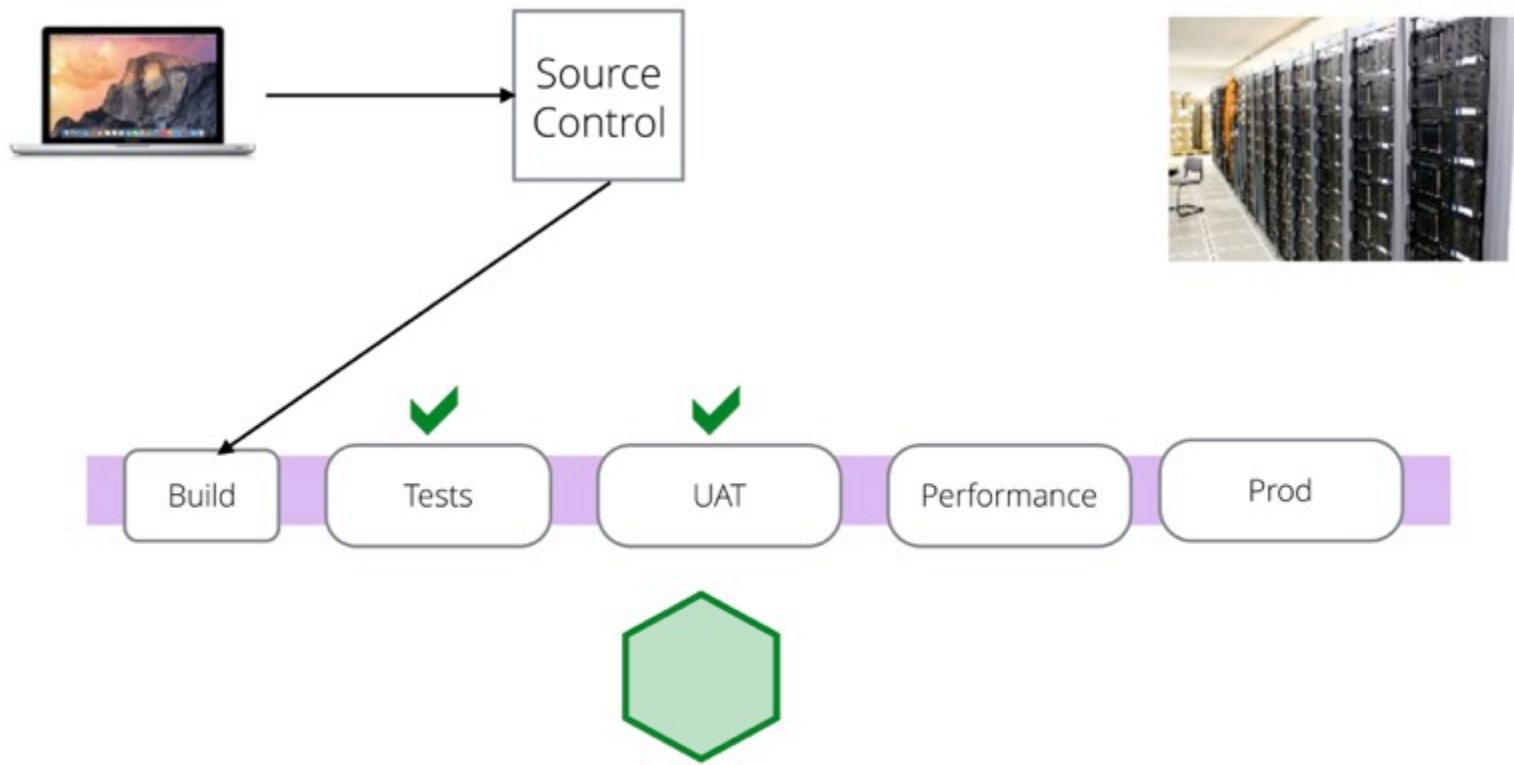


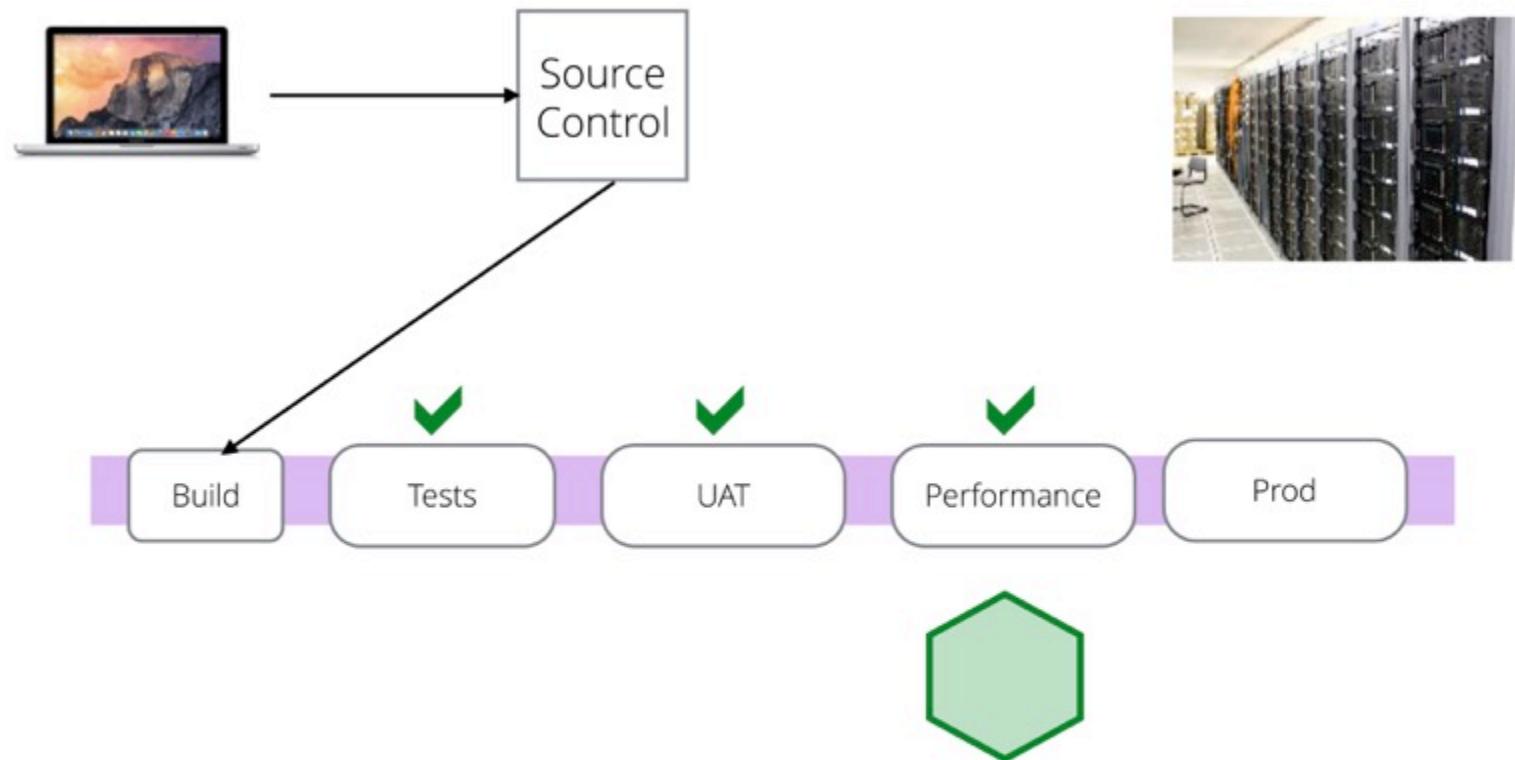


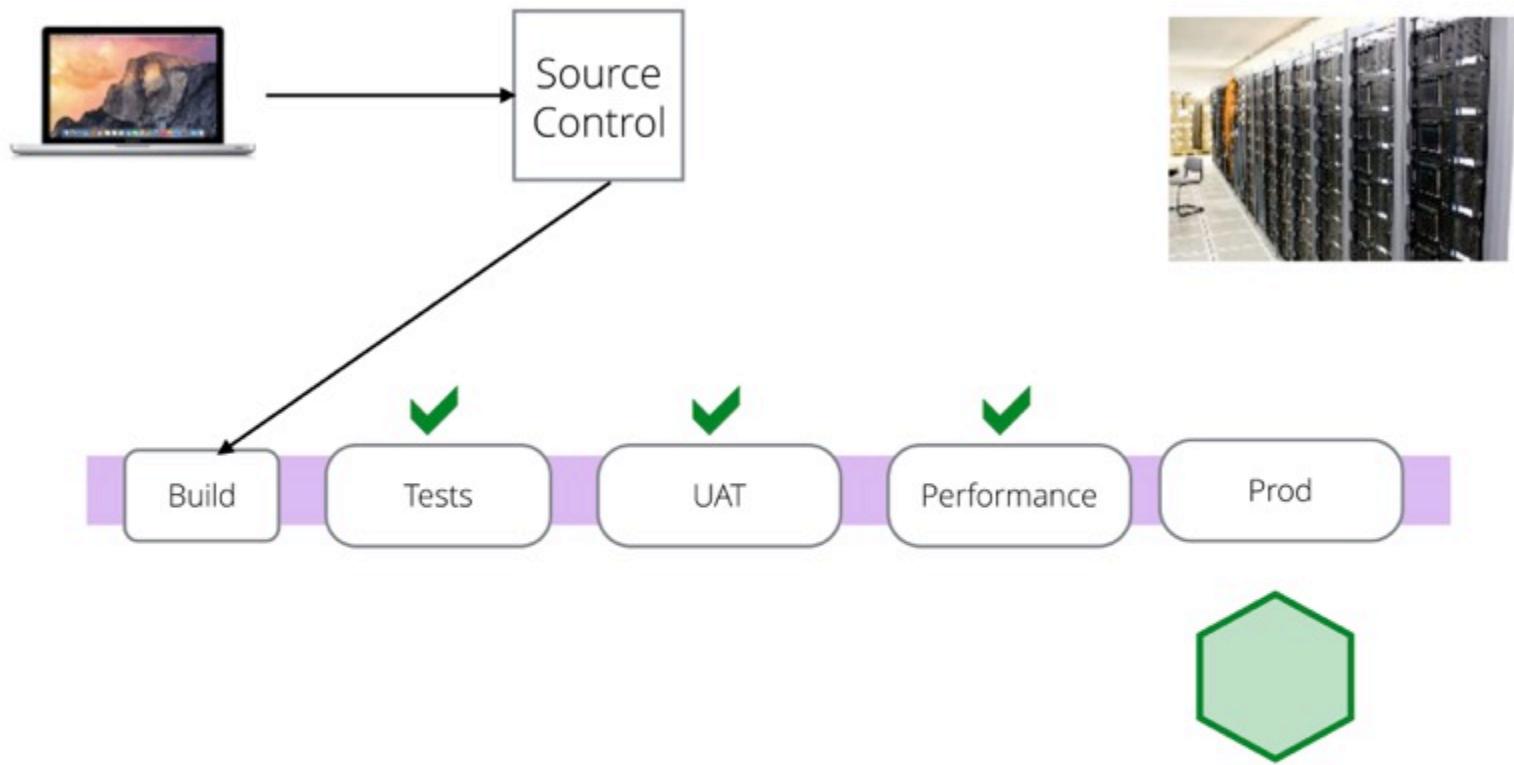


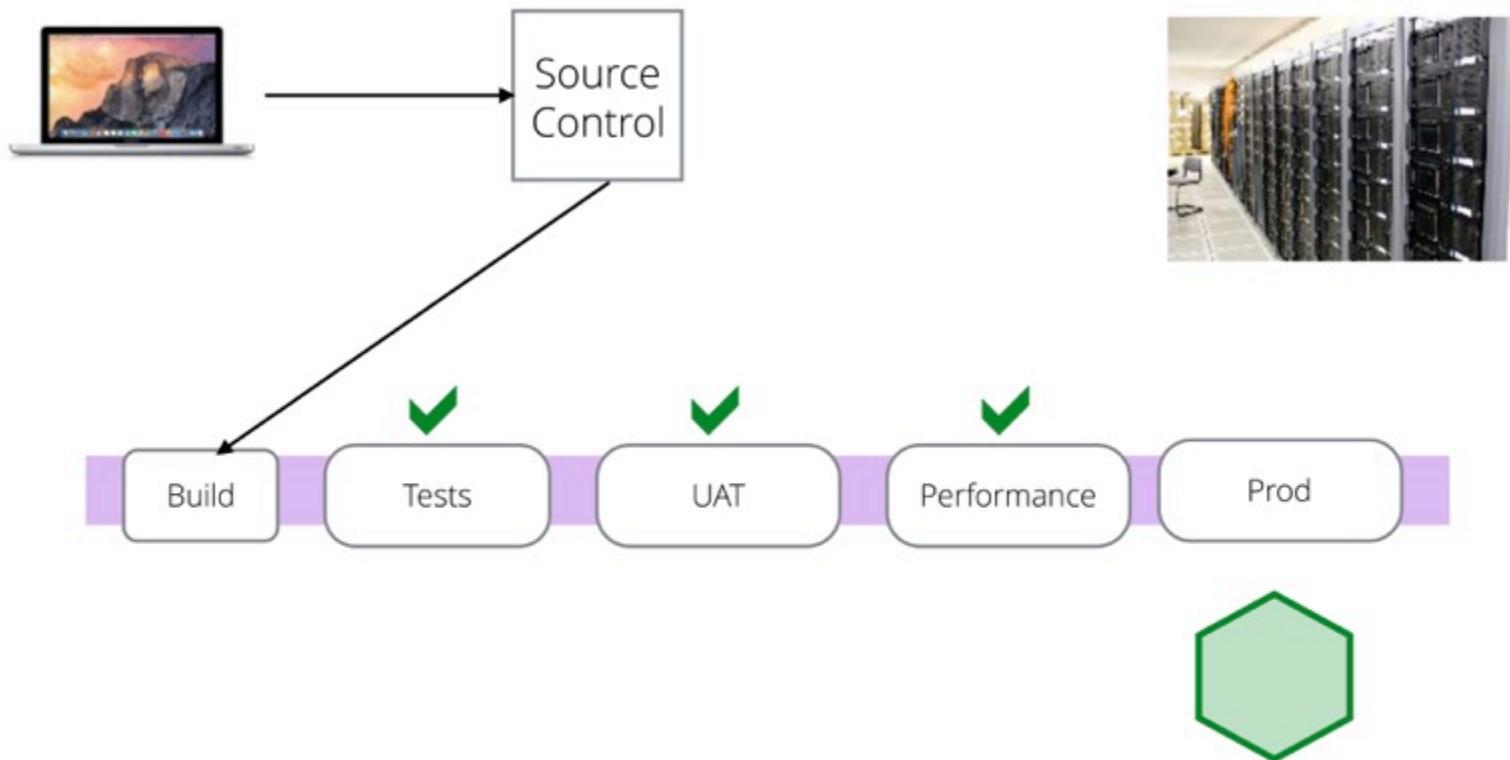




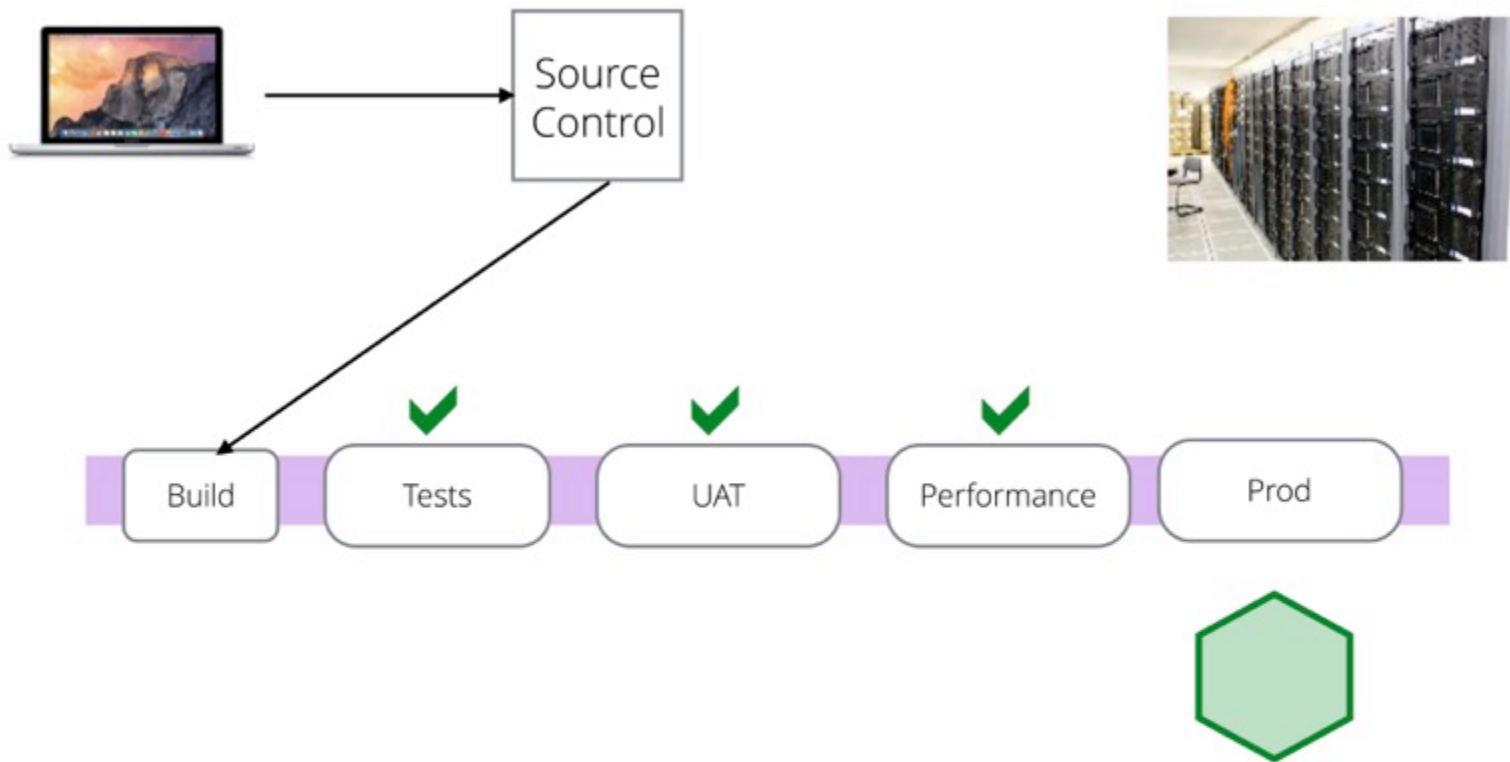








One Artifact For All Environments



One Artifact For All Environments

Same Deployment Process Everywhere

\$ deploy Returns v456 Production

Service Name



```
$ deploy Returns v456 Production
```

Service Name      Version

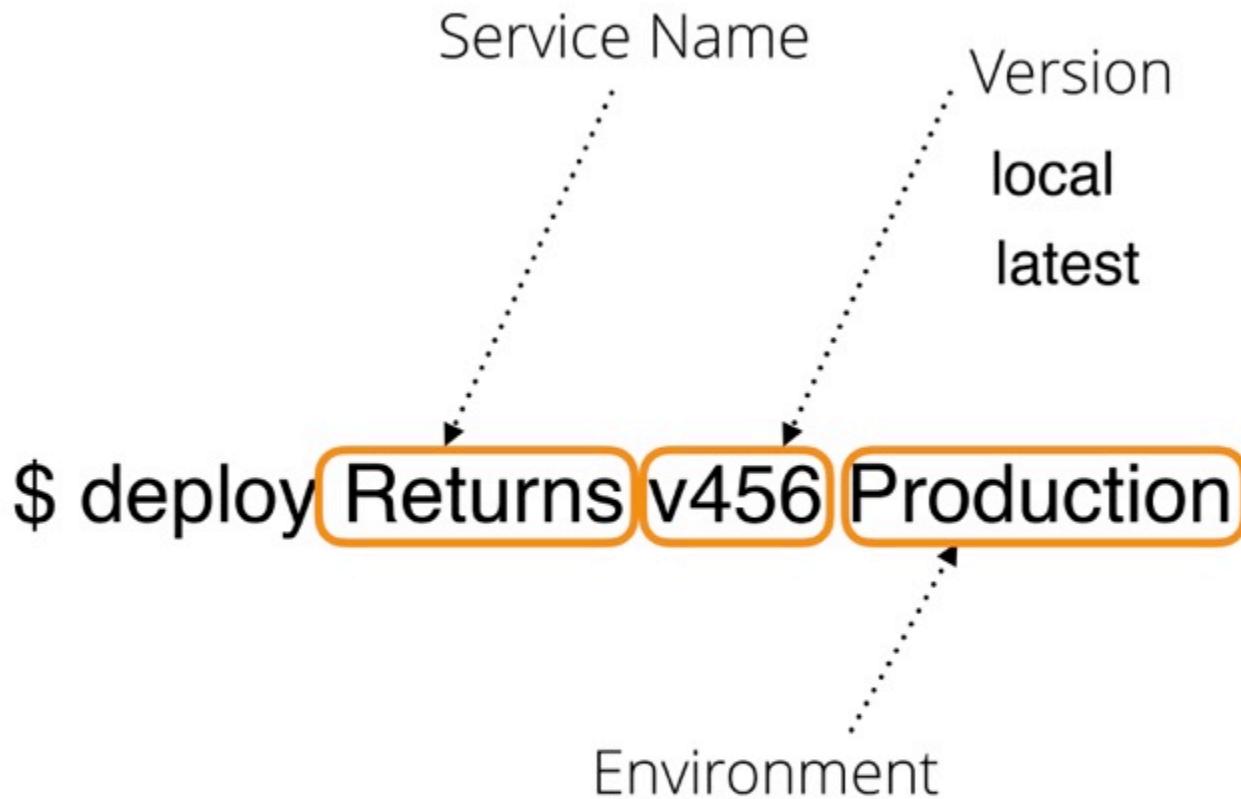
\$ deploy Returns v456 Production

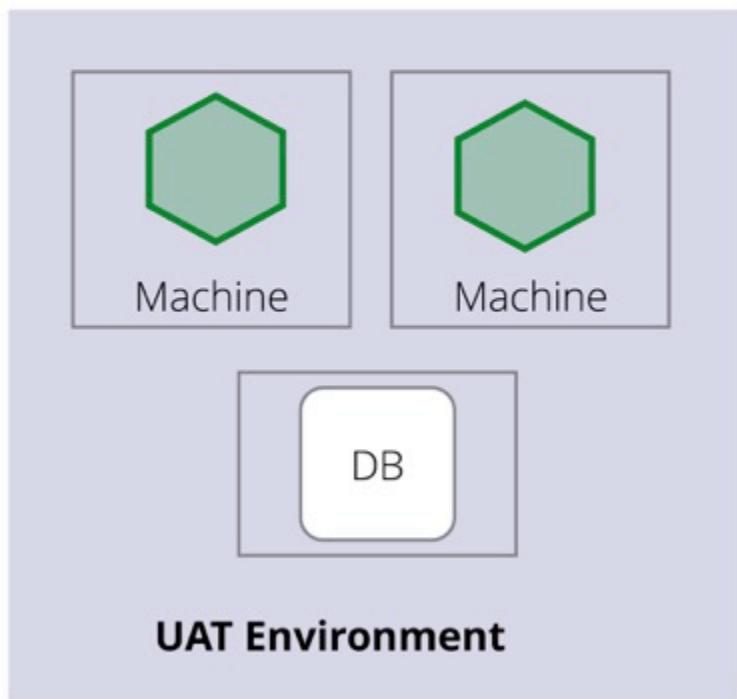
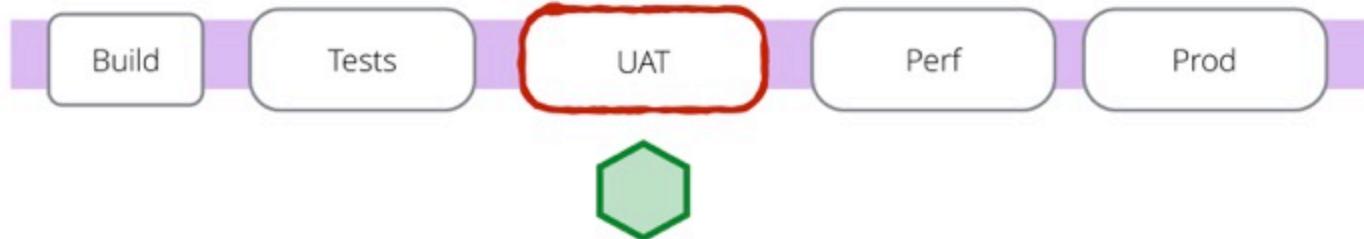
Service Name  
Version  
local

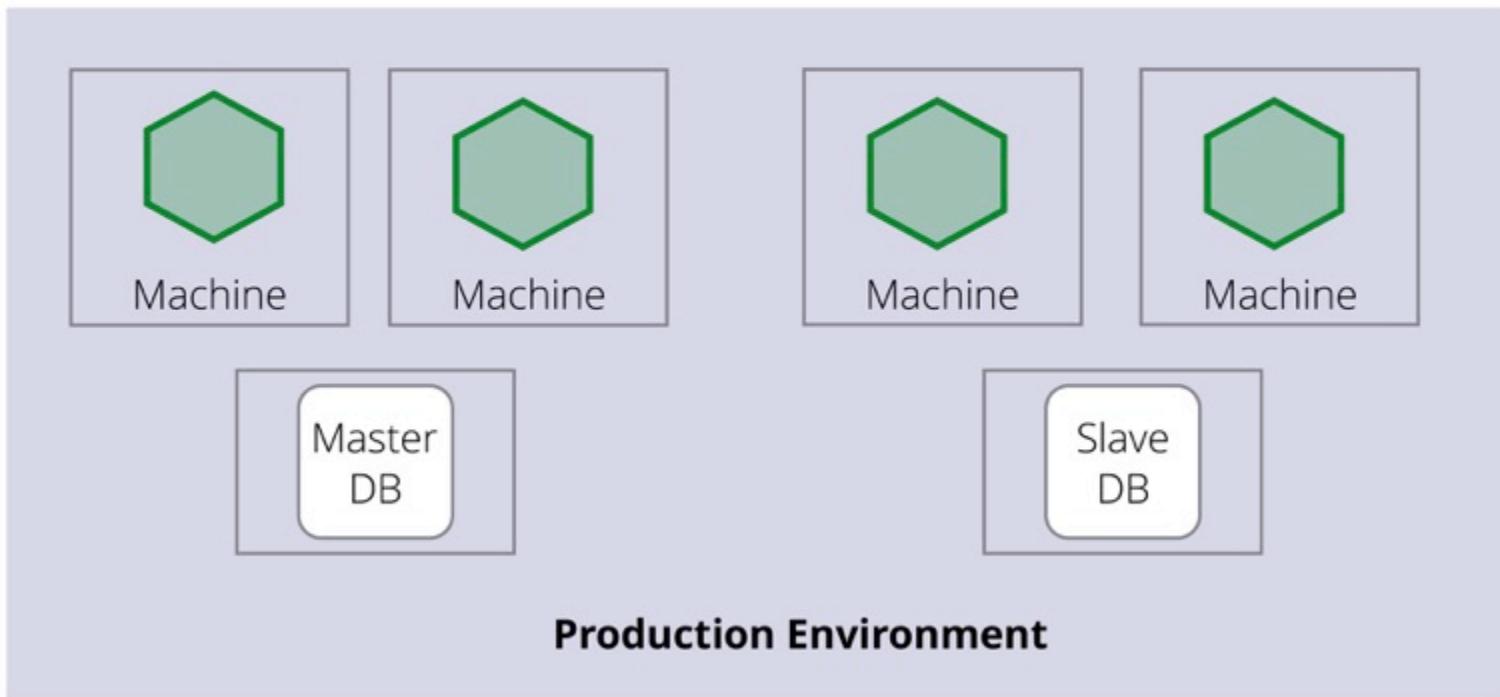
\$ deploy Returns v456 Production

Service Name  
Version  
local  
latest

\$ deploy Returns v456 Production







Same Artifact

Same Artifact

Different Topology

```
# create virtual machine
resource "azurerm_virtual_machine" "helloterraformvm" {
    name = "terraformvm"
    location = "West US"
    resource_group_name = "${azurerm_resource_group.helloterraform.name}"
    network_interface_ids = ["${azurerm_network_interface.helloterraformnic.id}"]
    vm_size = "Standard_A0"

    storage_image_reference {
        publisher = "Canonical"
        offer = "UbuntuServer"
        sku = "14.04.2-LTS"
        version = "latest"
    }

    storage_os_disk {
        name = "myosdisk"
        vhd_uri = "${azurerm_storage_account.helloterraformstorage.primary_blob_endpoints[0].storagecontainer.name}/myosdisk.vhd"
        caching = "ReadWrite"
        create_option = "FromImage"
    }

    os_profile {
        computer_name =
        admin_username =
        admin_password =
    }

    os_profile_linux_config {
        disable_password_authentication =
    }

    tags {
        environment =
    }
}
```



# Version controlled

```
# create virtual machine
resource "azurerm_virtual_machine" "helloterraformvm" {
    name = "terraformvm"
    location = "West US"
    resource_group_name = "${azurerm_resource_group.helloterraform.name}"
    network_interface_ids = ["${azurerm_network_interface.helloterraformnic.id}"]
    vm_size = "Standard_A0"

    storage_image_reference {
        publisher = "Canonical"
        offer = "UbuntuServer"
        sku = "14.04.2-LTS"
        version = "latest"
    }

    storage_os_disk {
        name = "myosdisk"
        vhd_uri = "${azurerm_storage_account.helloterraformstorage.primary_blob_endpoint}${storagecontainer.name}/myosdisk.vhd"
        caching = "ReadWrite"
        create_option = "FromImage"
    }

    os_profile {
        computer_name =
        admin_username =
        admin_password =
    }

    os_profile_linux_config {
        disable_password_authentication =
    }

    tags {
        environment =
    }
}
```

HashiCorp

Terraform



```
# create virtual machine
resource "azurerm_virtual_machine" "helloterraformvm" {
    name = "terraformvm"
    location = "West US"
    resource_group_name = "${azurerm_resource_group.helloterraform.name}"
    network_interface_ids = ["${azurerm_network_interface.helloterraformnic.id}"]
    vm_size = "Standard_A0"

    storage_image_reference {
        publisher = "Canonical"
        offer = "UbuntuServer"
        sku = "14.04.2-LTS"
        version = "latest"
    }

    storage_os_disk {
        name = "myosdisk"
        vhd_uri = "${azurerm_storage_account.helloterraformstorage.primary_blob_endpoint}${storagecontainer.name}/myosdisk.vhd"
        caching = "ReadWrite"
        create_option = "FromImage"
    }

    os_profile {
        computer_name =
        admin_username =
        admin_password =
    }

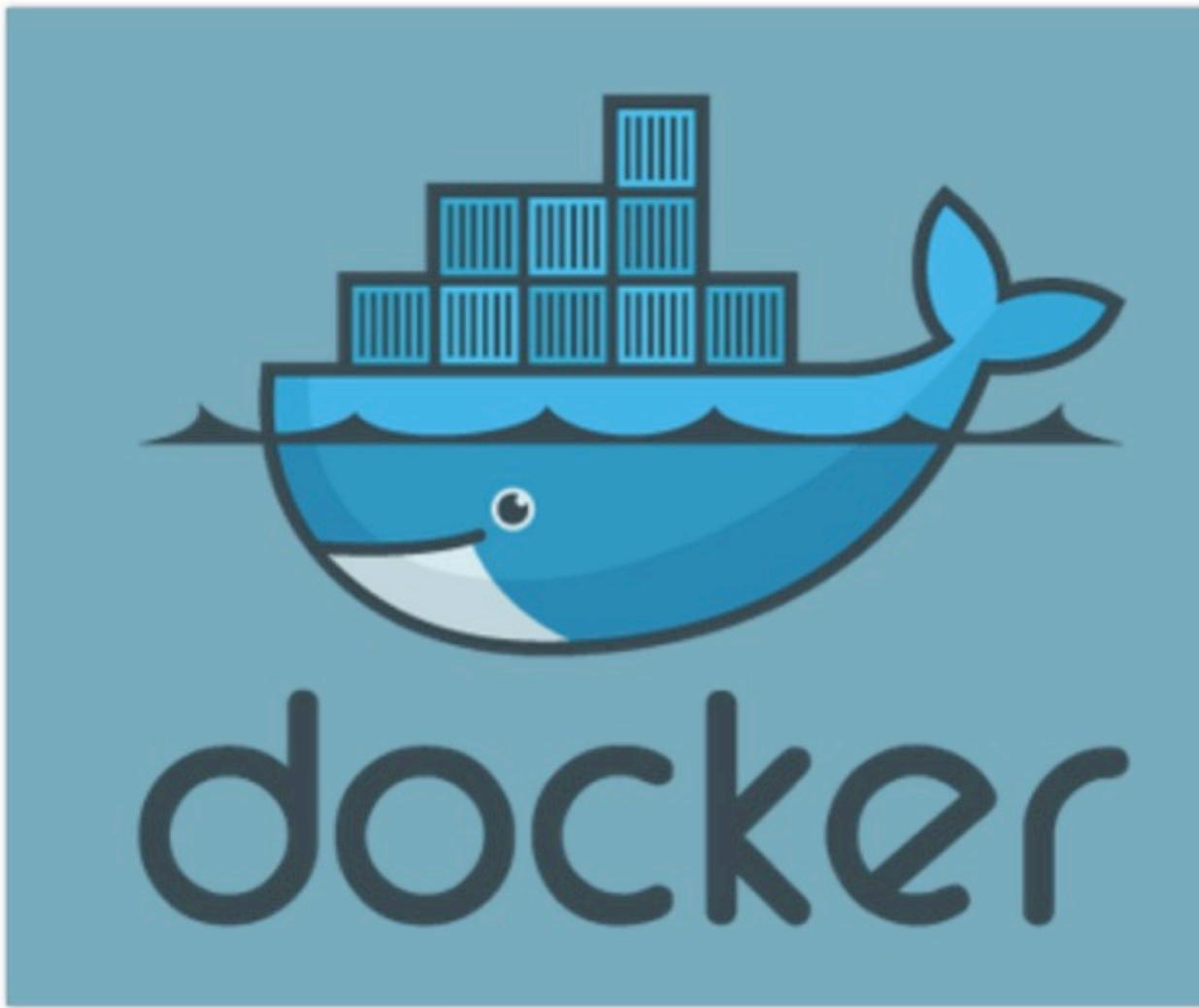
    os_profile_linux_config {
        disable_password_authentication =
    }

    tags {
        environment =
    }
}
```

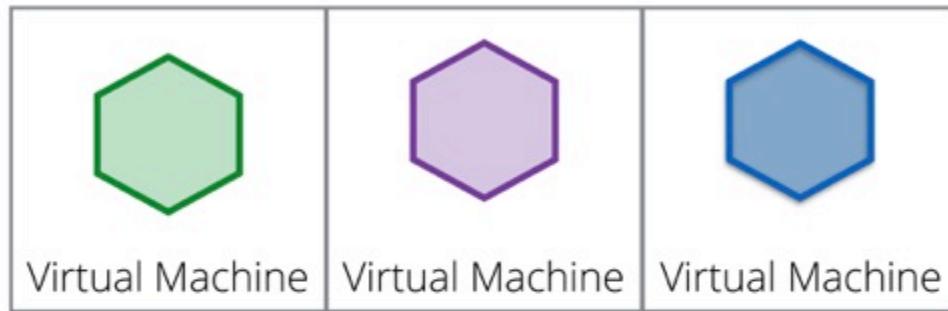
Version controlled

Parameterisable

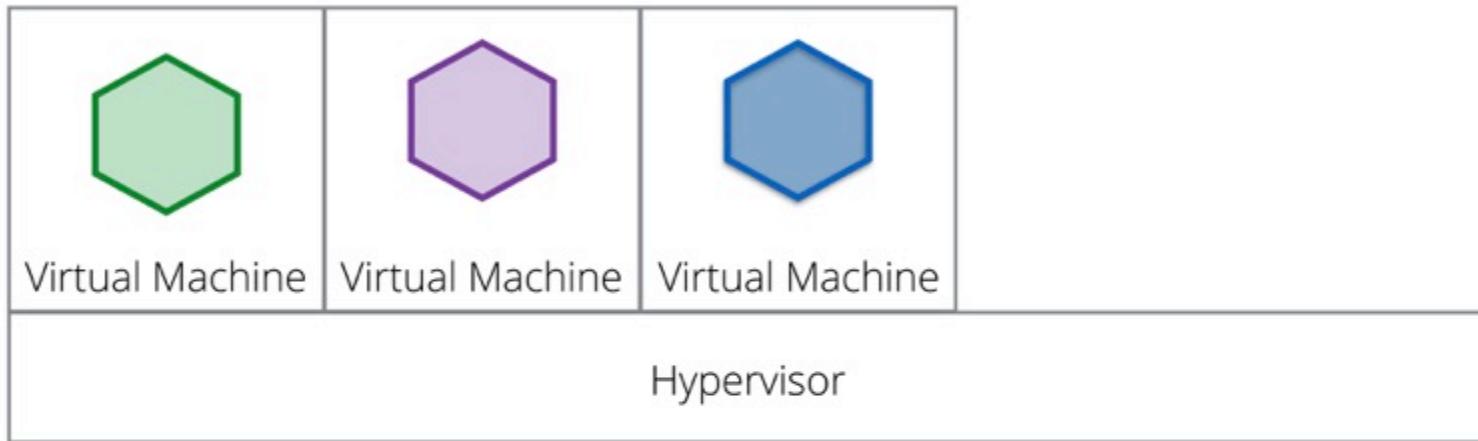
HashiCorp  
**Terraform**



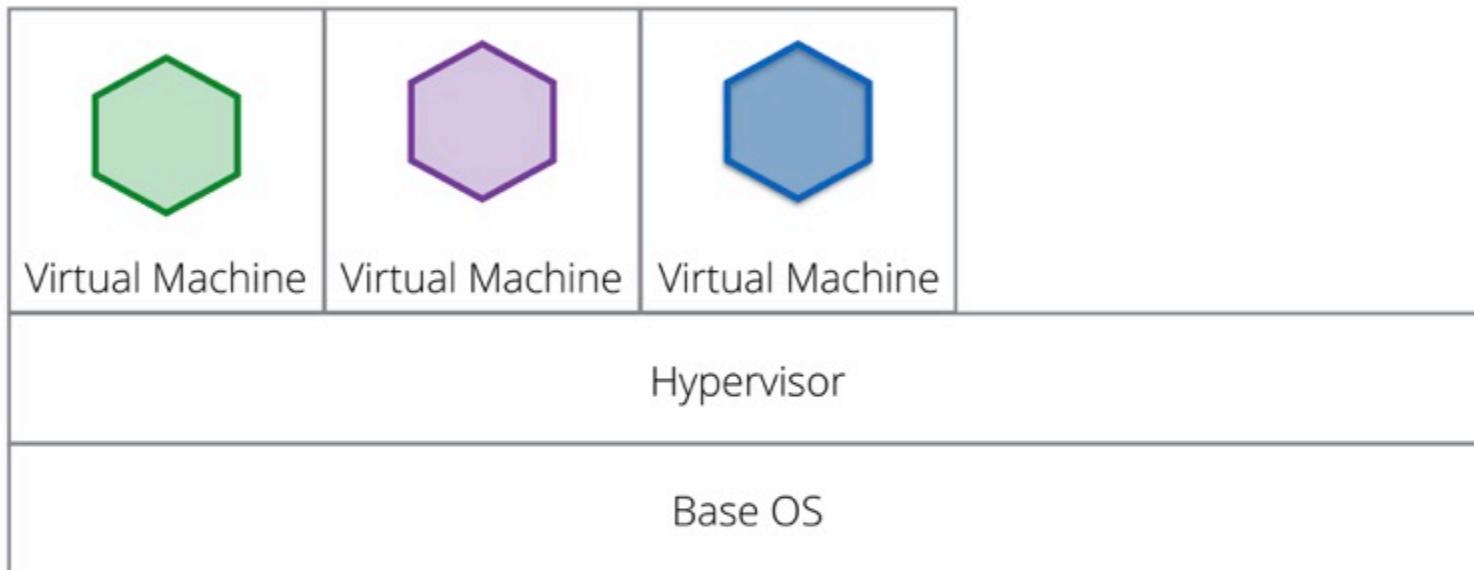
## TYPE 2 VIRTUALISATION



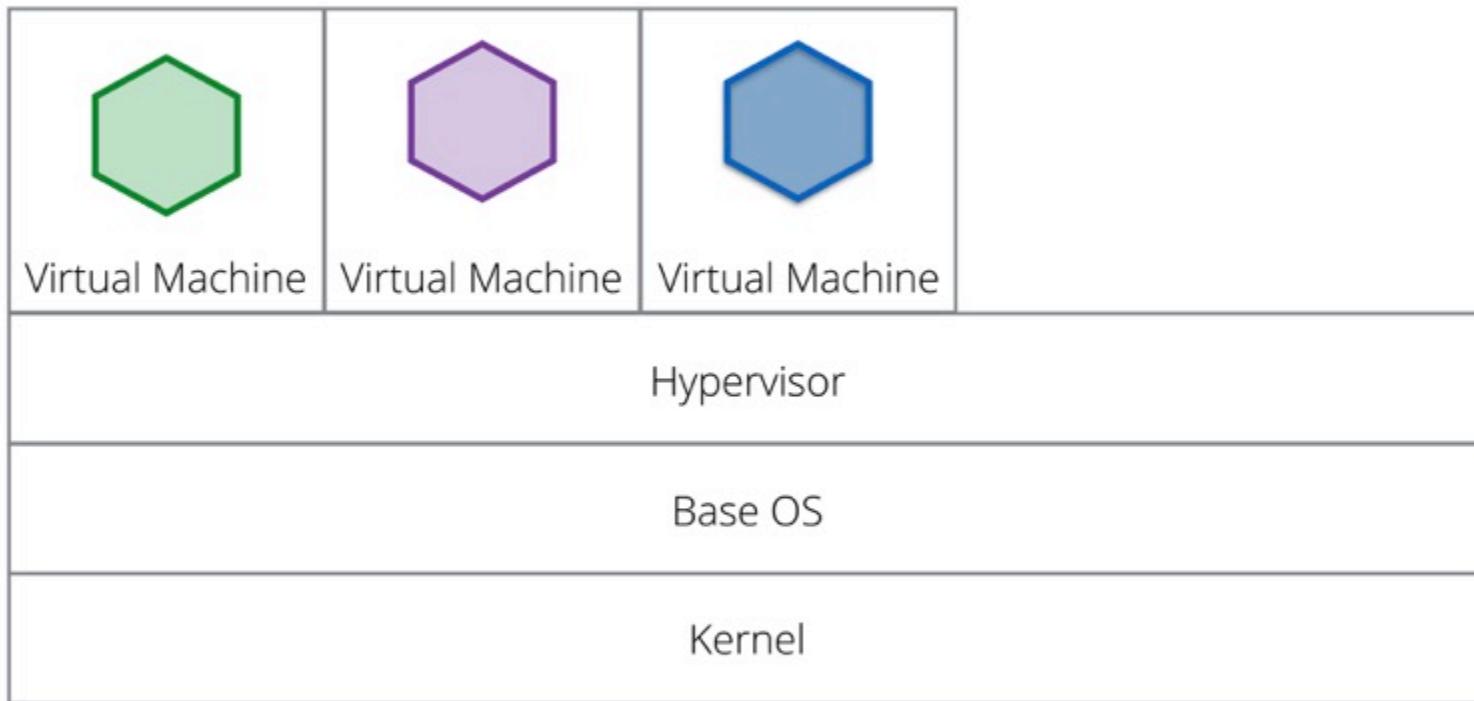
## TYPE 2 VIRTUALISATION



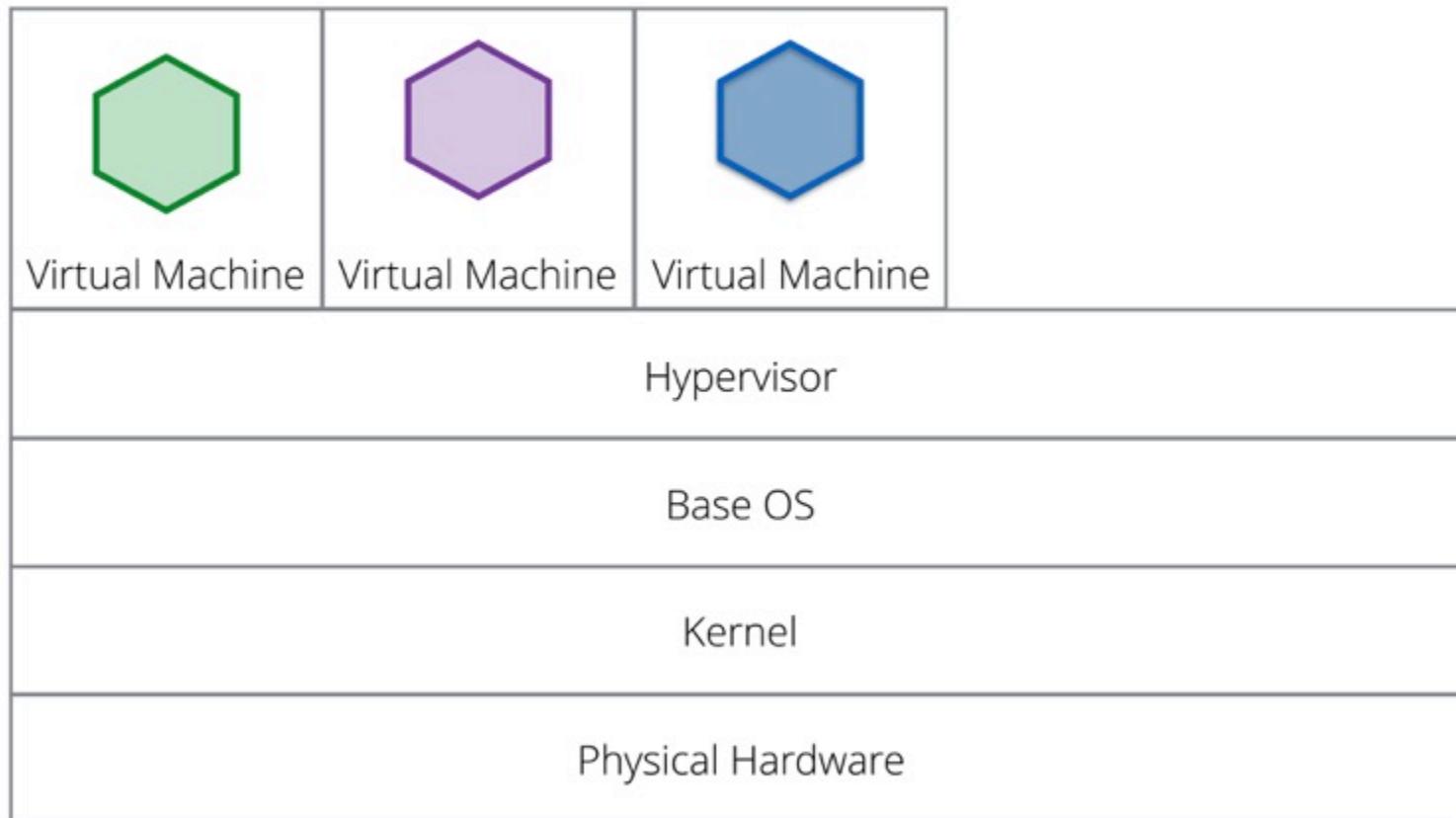
## TYPE 2 VIRTUALISATION



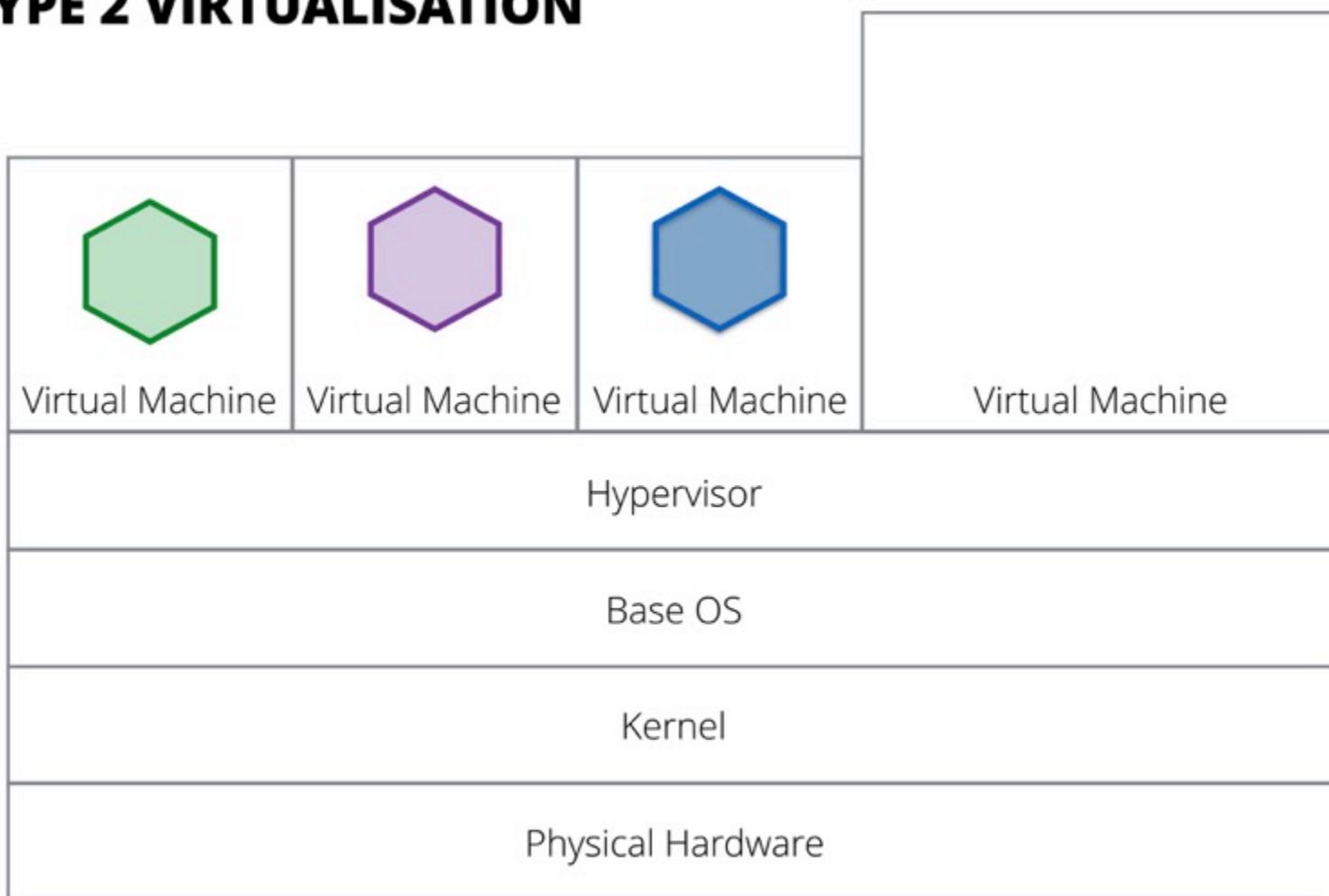
## TYPE 2 VIRTUALISATION



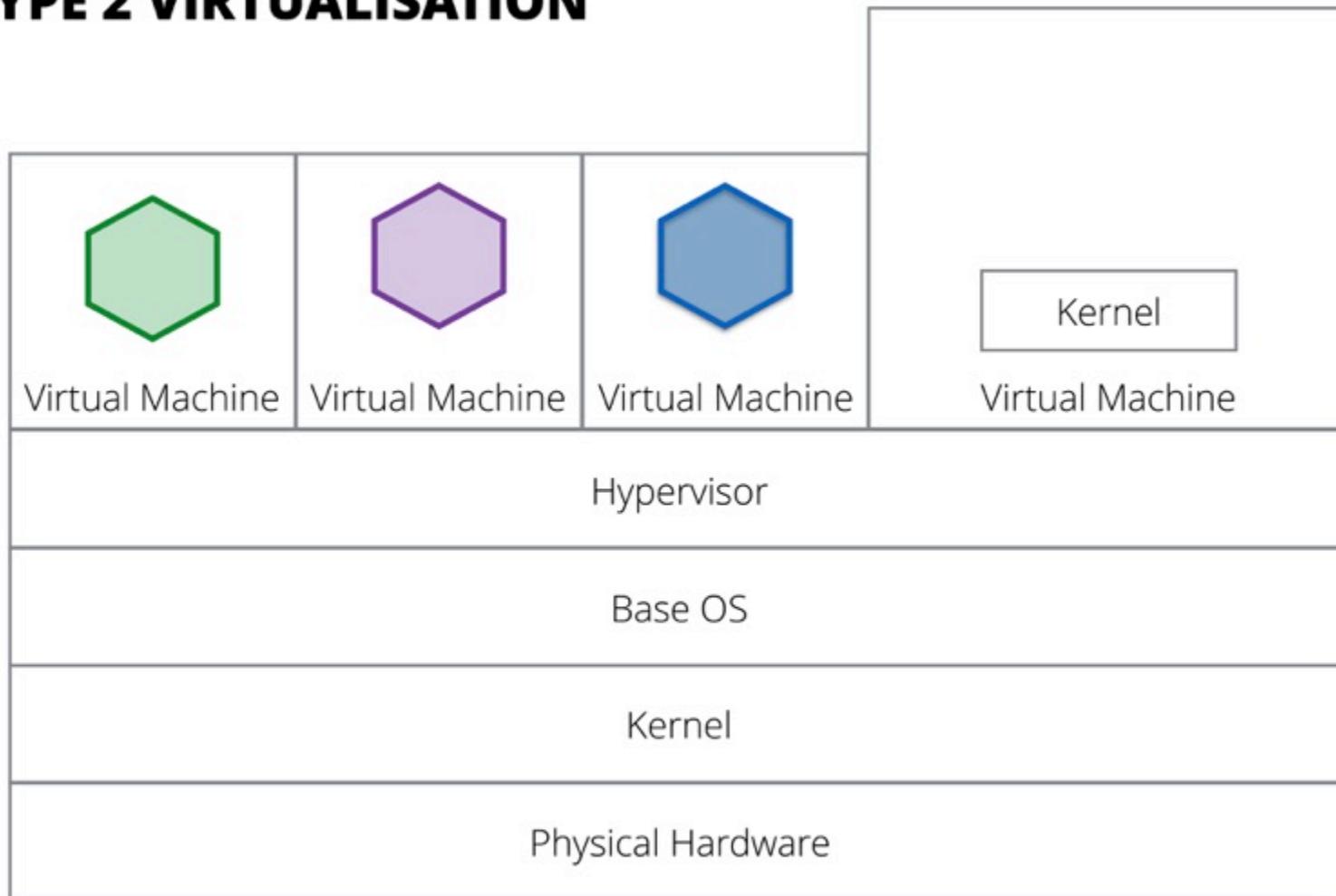
## TYPE 2 VIRTUALISATION



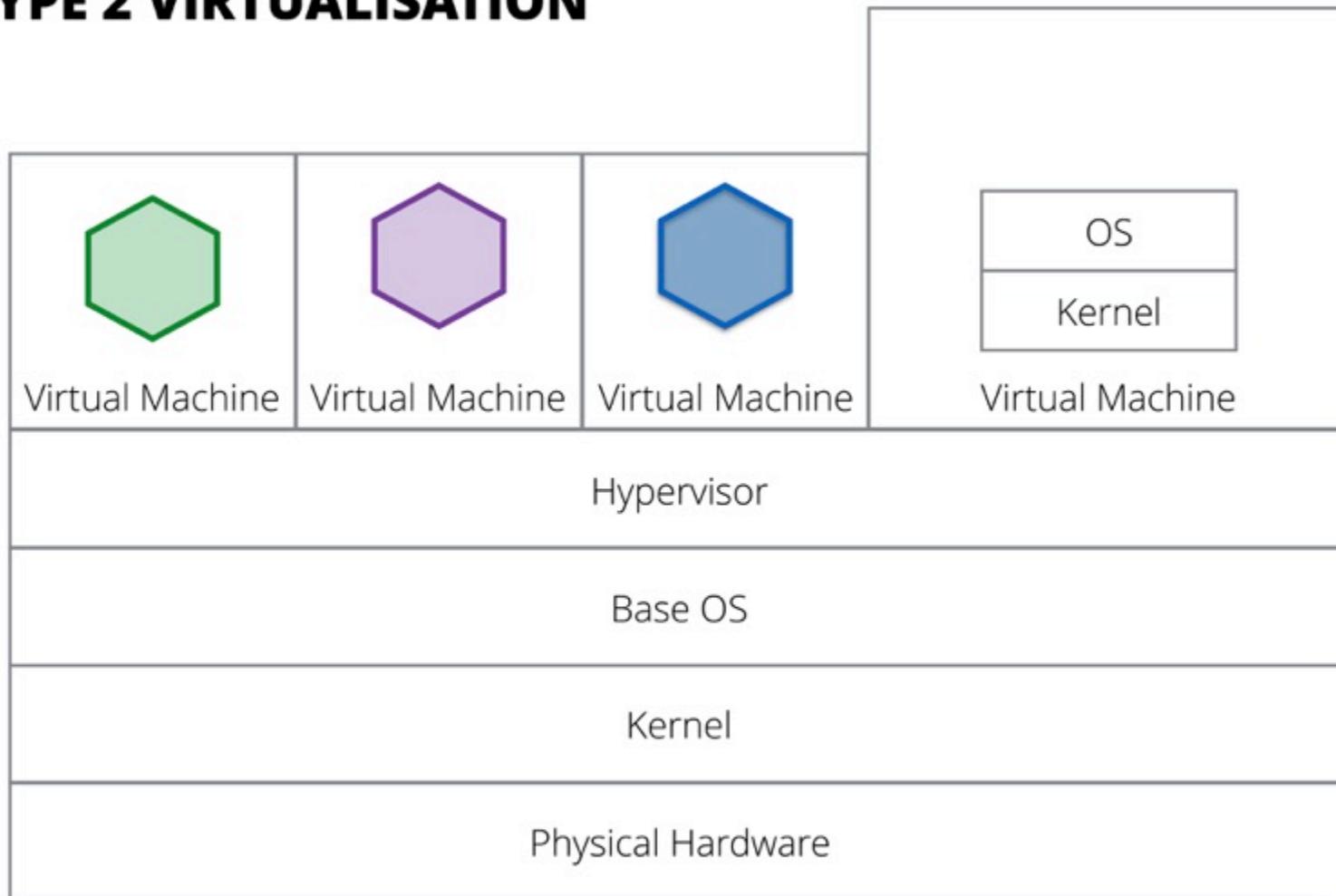
## TYPE 2 VIRTUALISATION



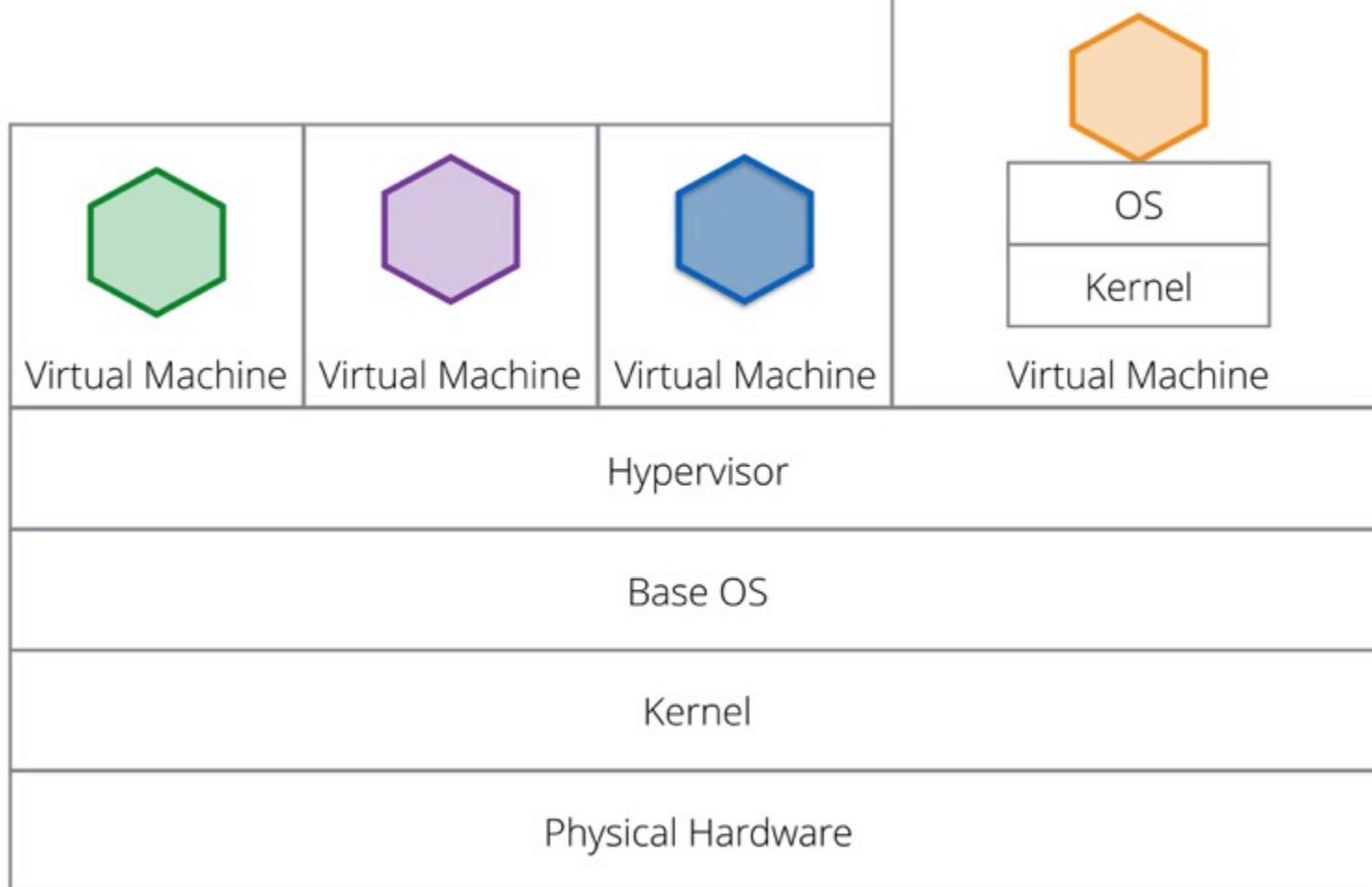
## TYPE 2 VIRTUALISATION



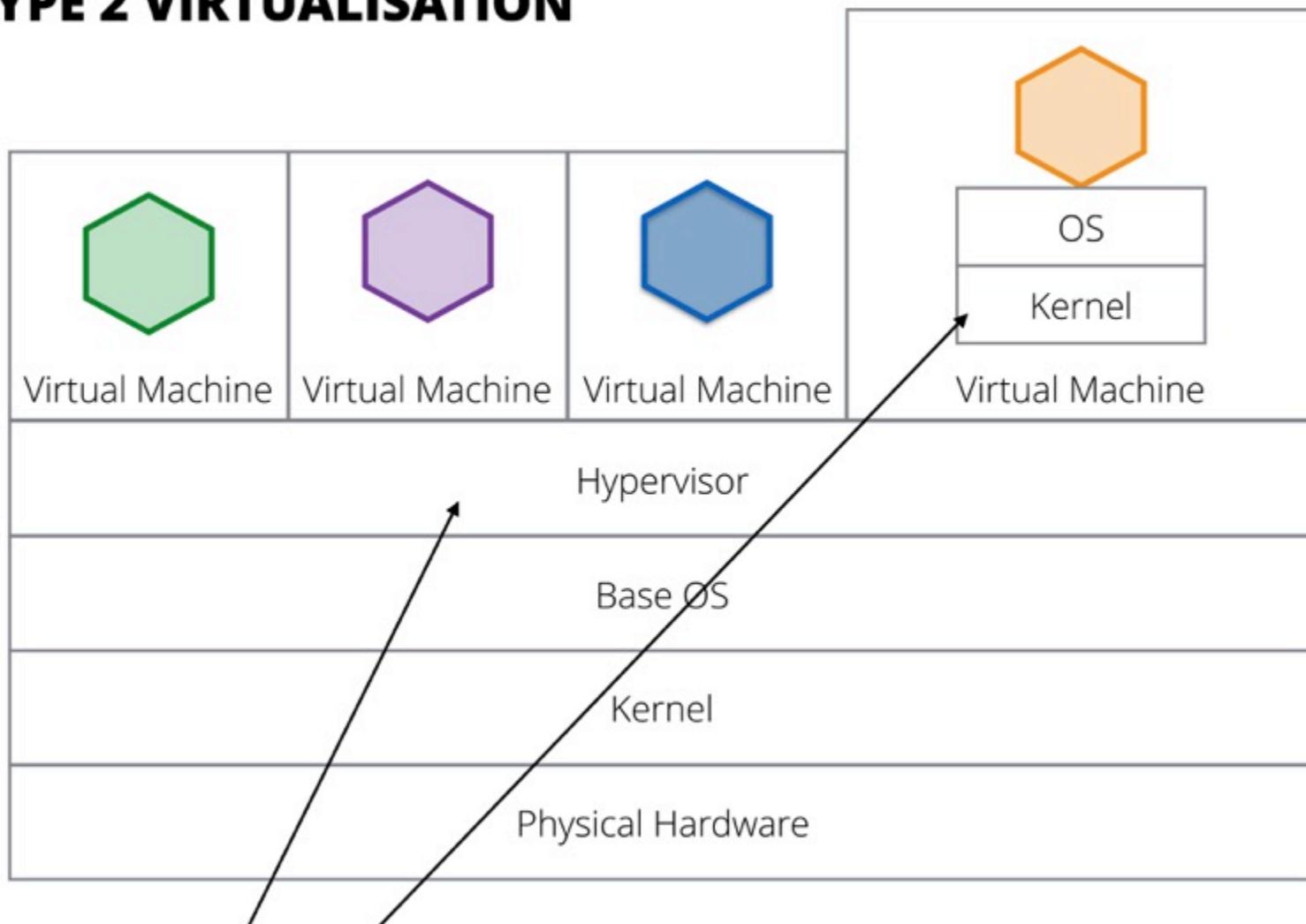
## TYPE 2 VIRTUALISATION



## TYPE 2 VIRTUALISATION

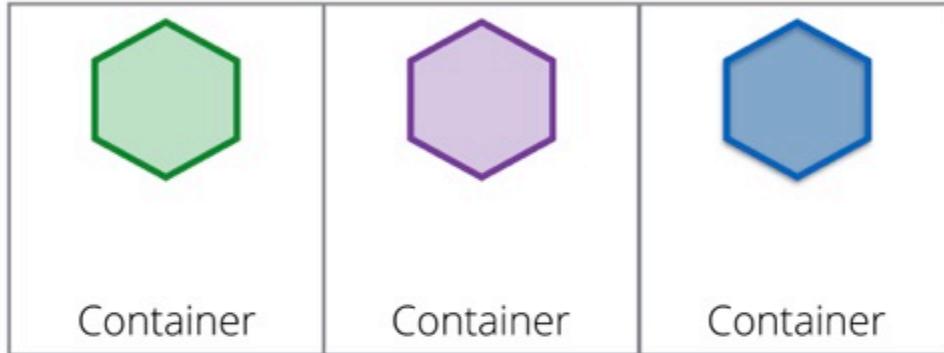


## TYPE 2 VIRTUALISATION

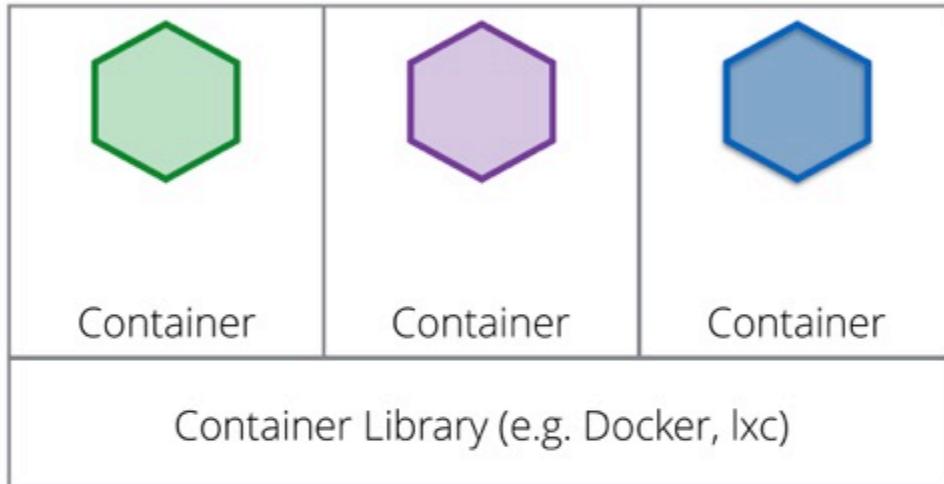


**Expensive!**

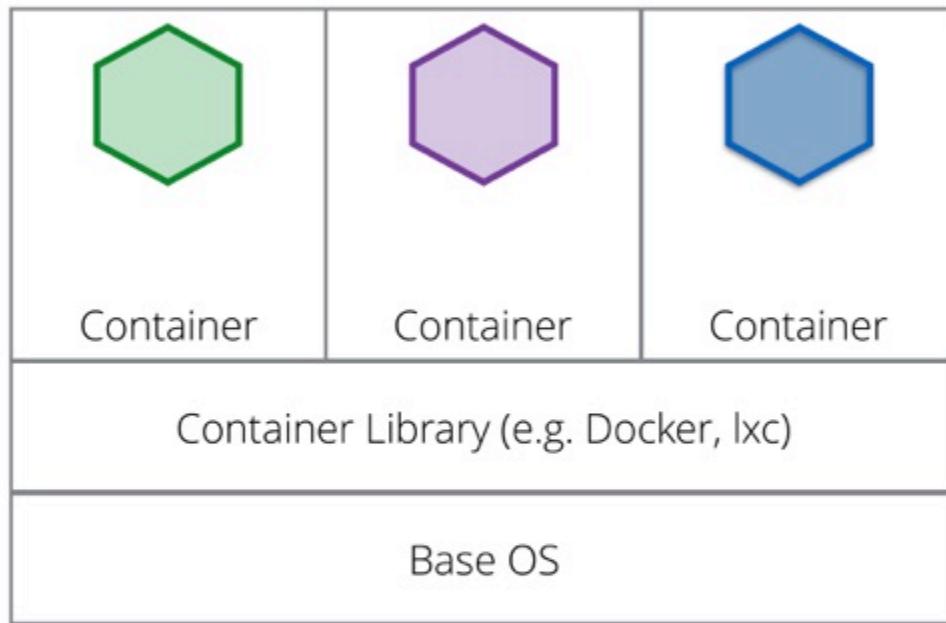
# CONTAINERISATION



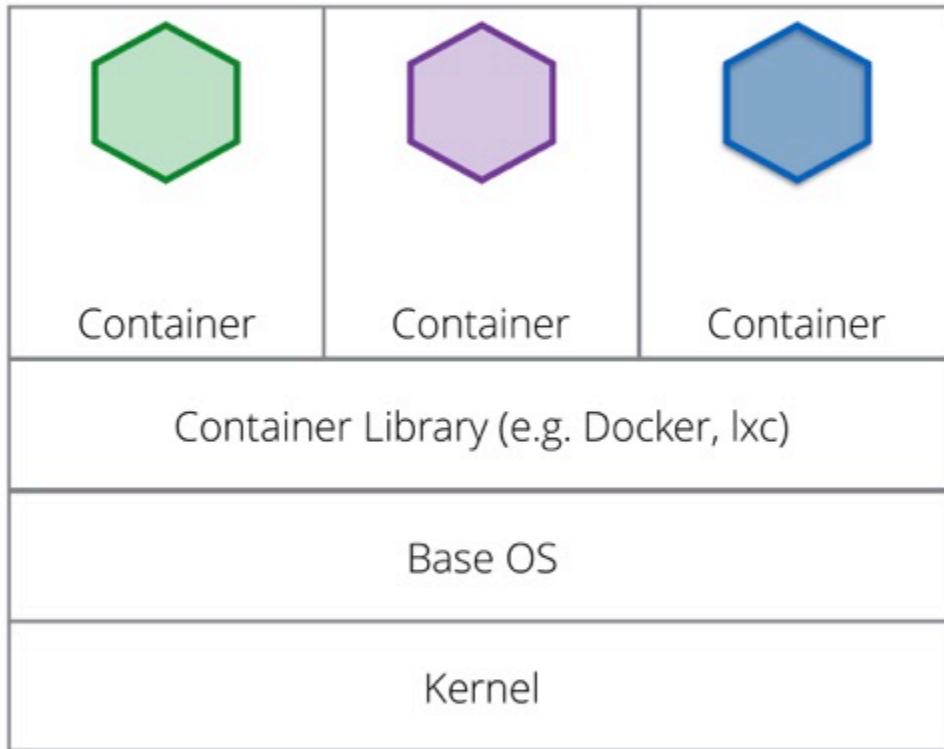
# CONTAINERISATION



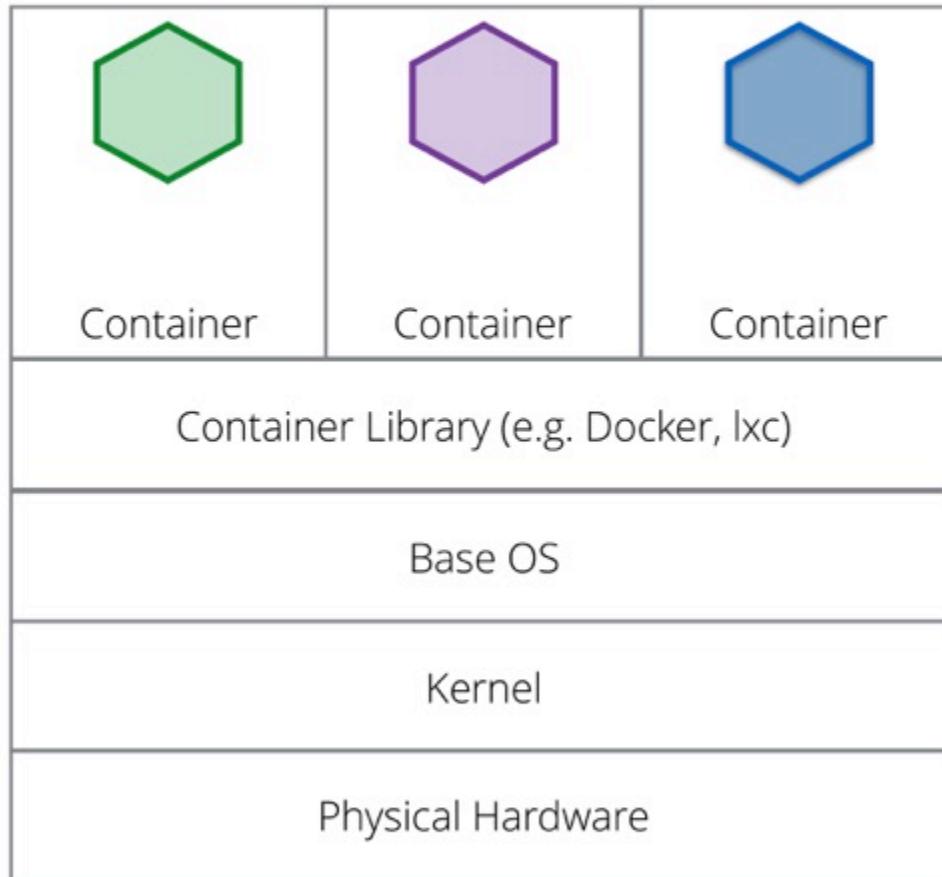
# CONTAINERISATION



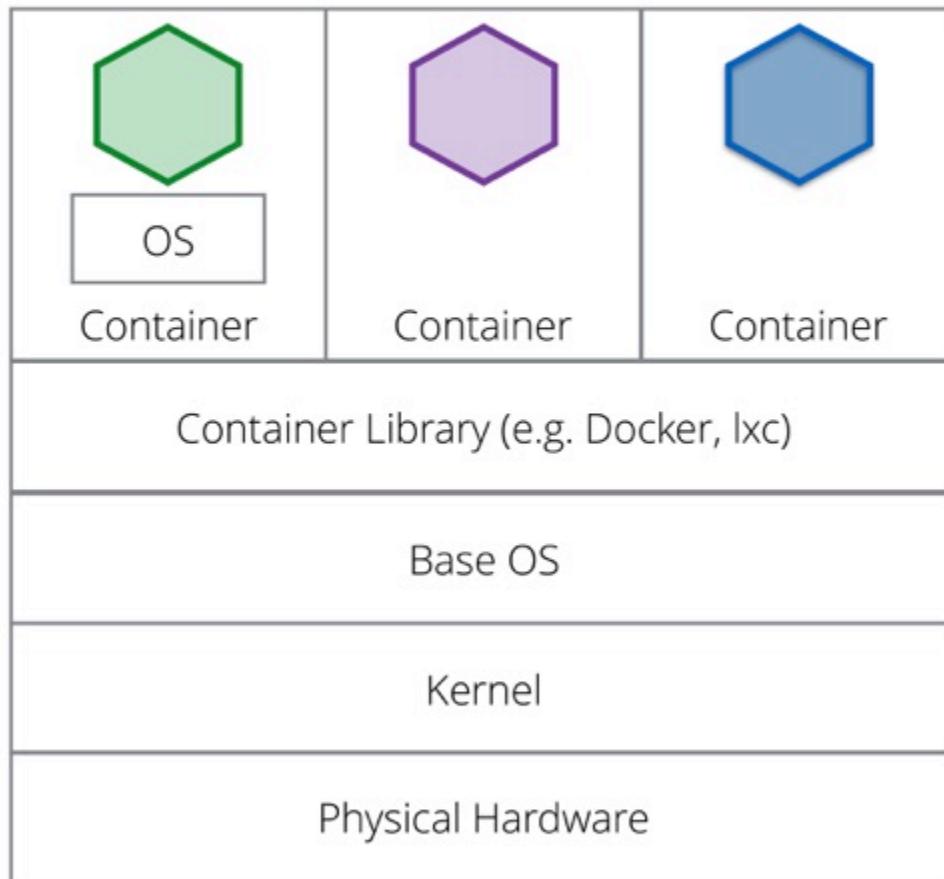
# CONTAINERISATION



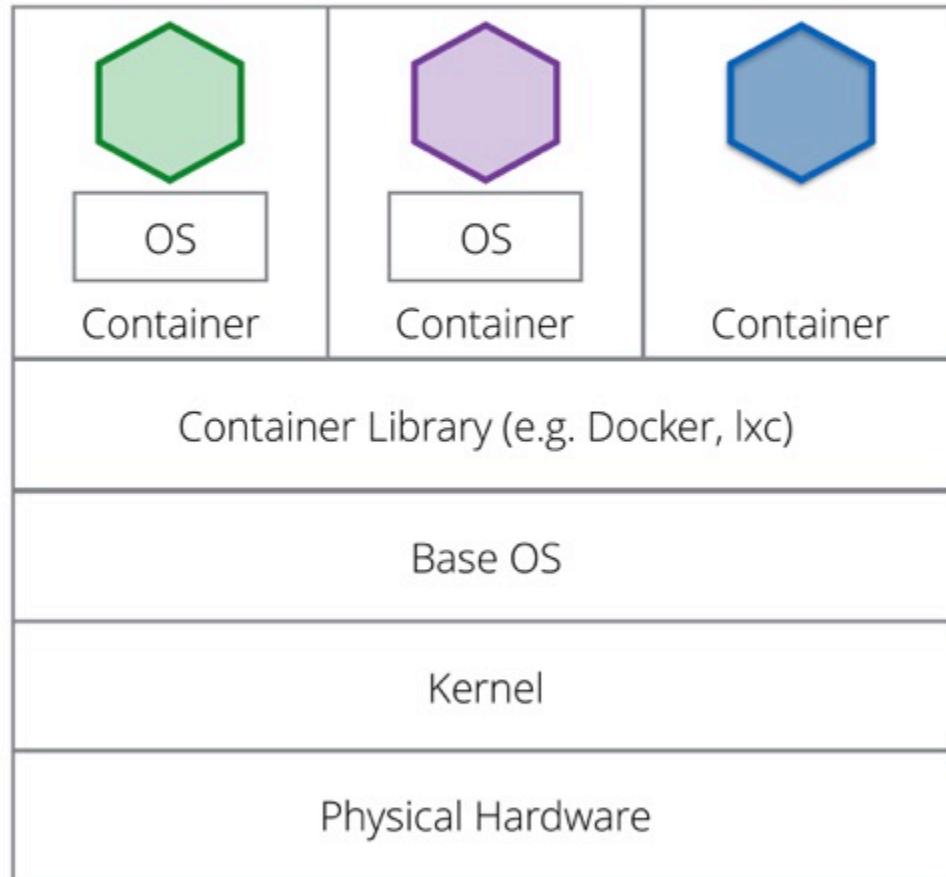
# CONTAINERISATION



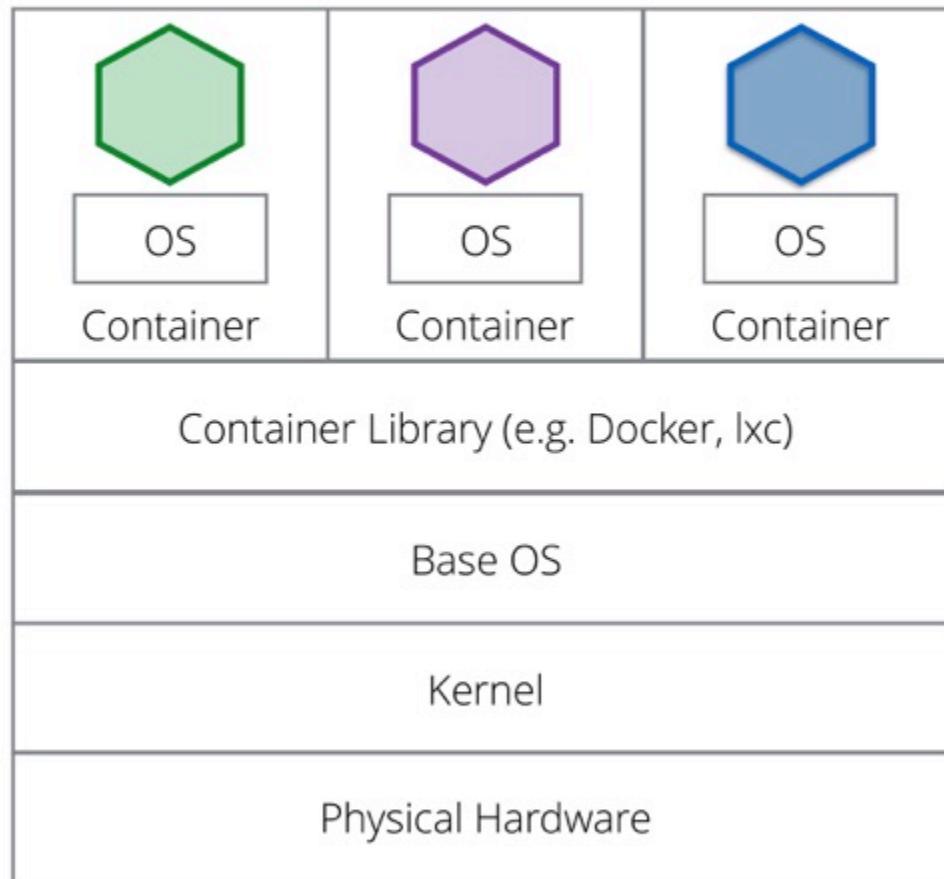
# CONTAINERISATION



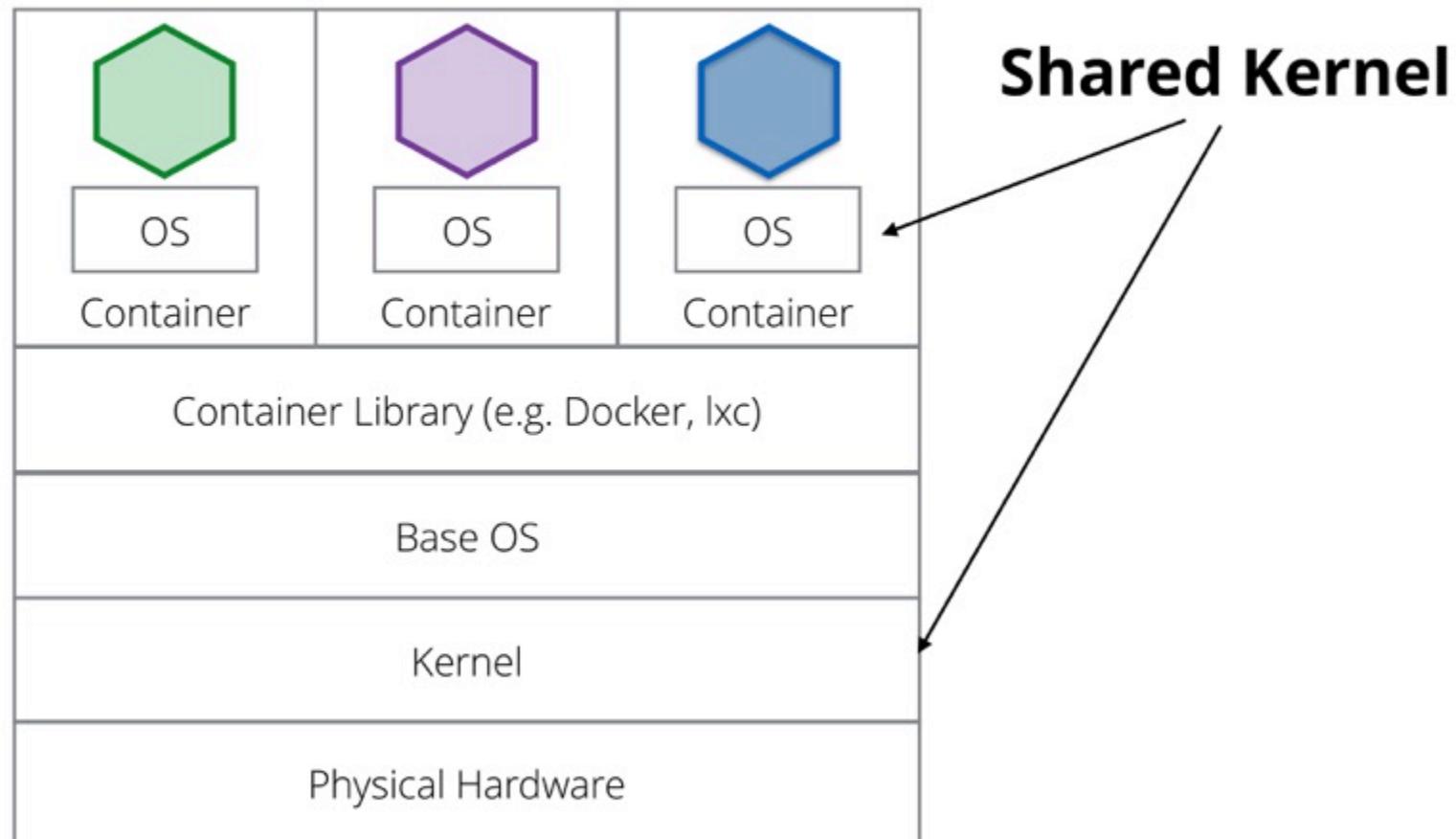
# CONTAINERISATION



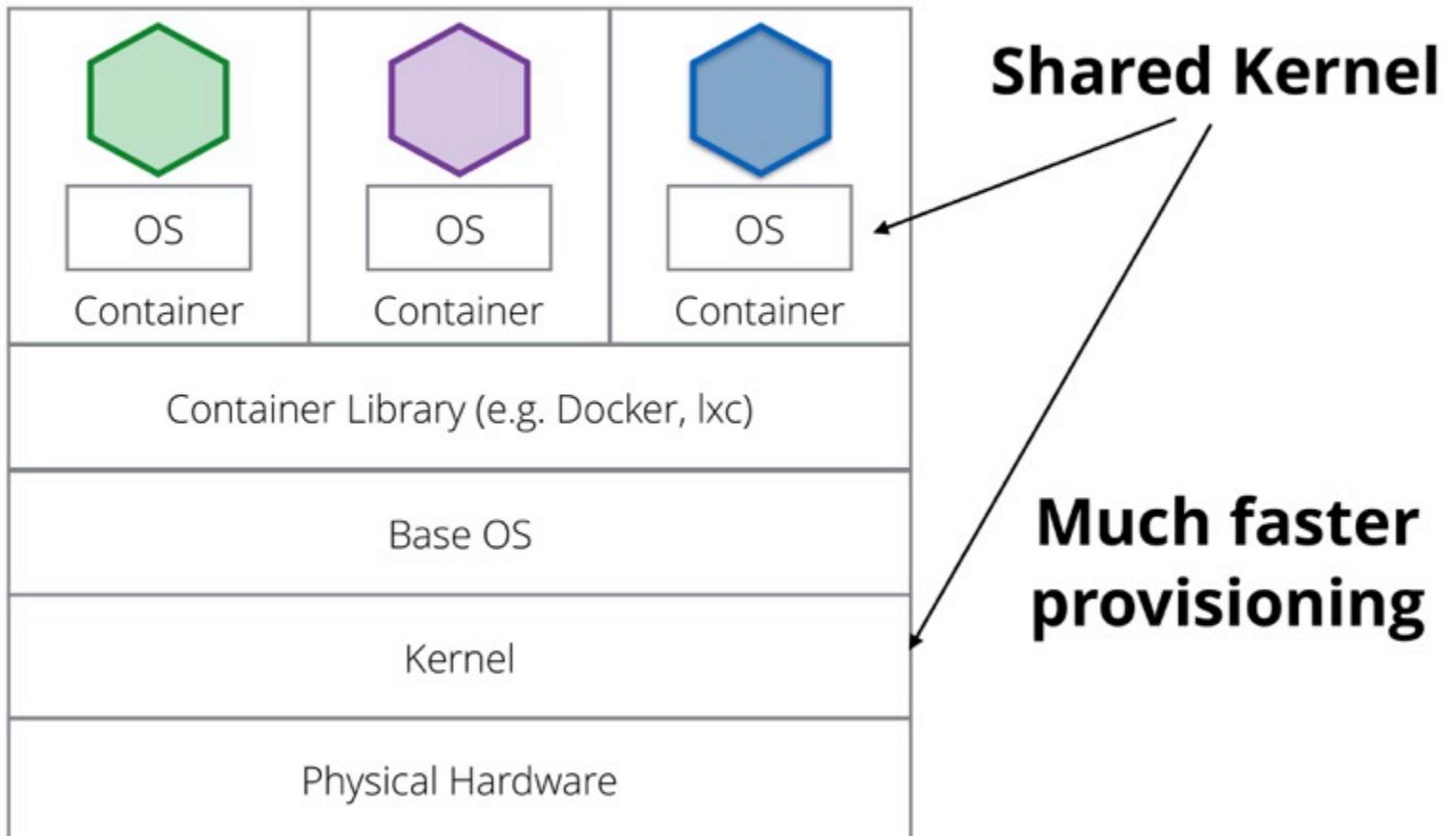
# CONTAINERISATION



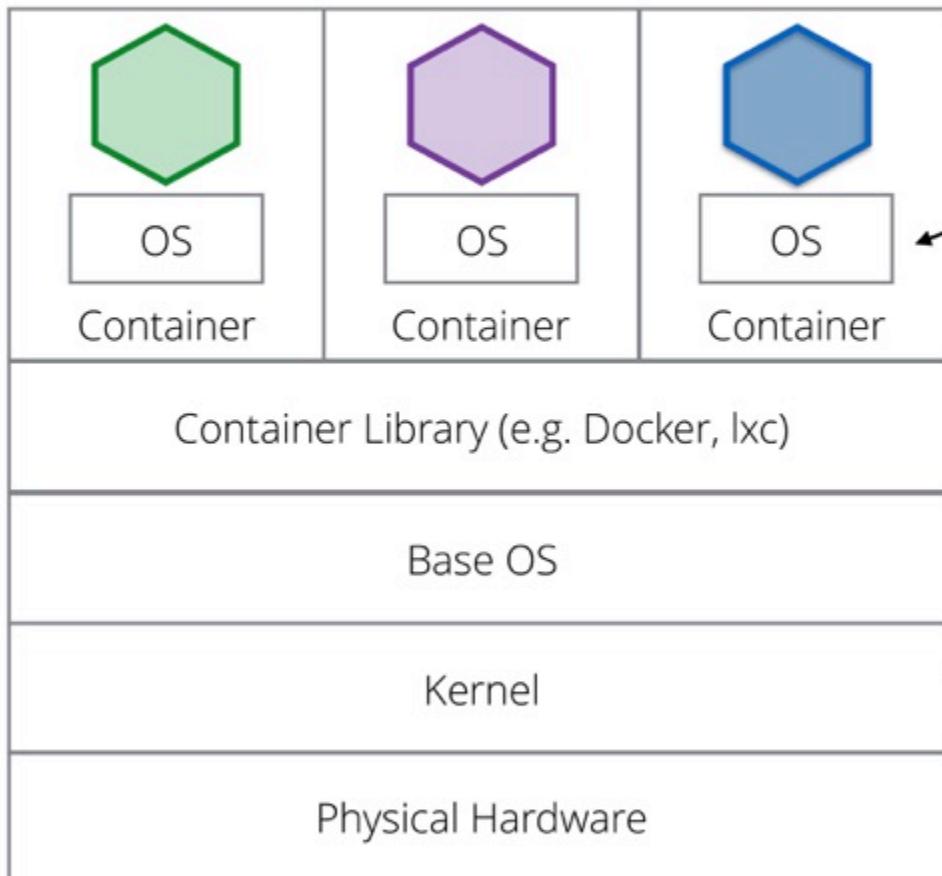
# CONTAINERISATION



# CONTAINERISATION



# CONTAINERISATION



**Shared Kernel**

**Much faster provisioning**

**Poorer Isolation**

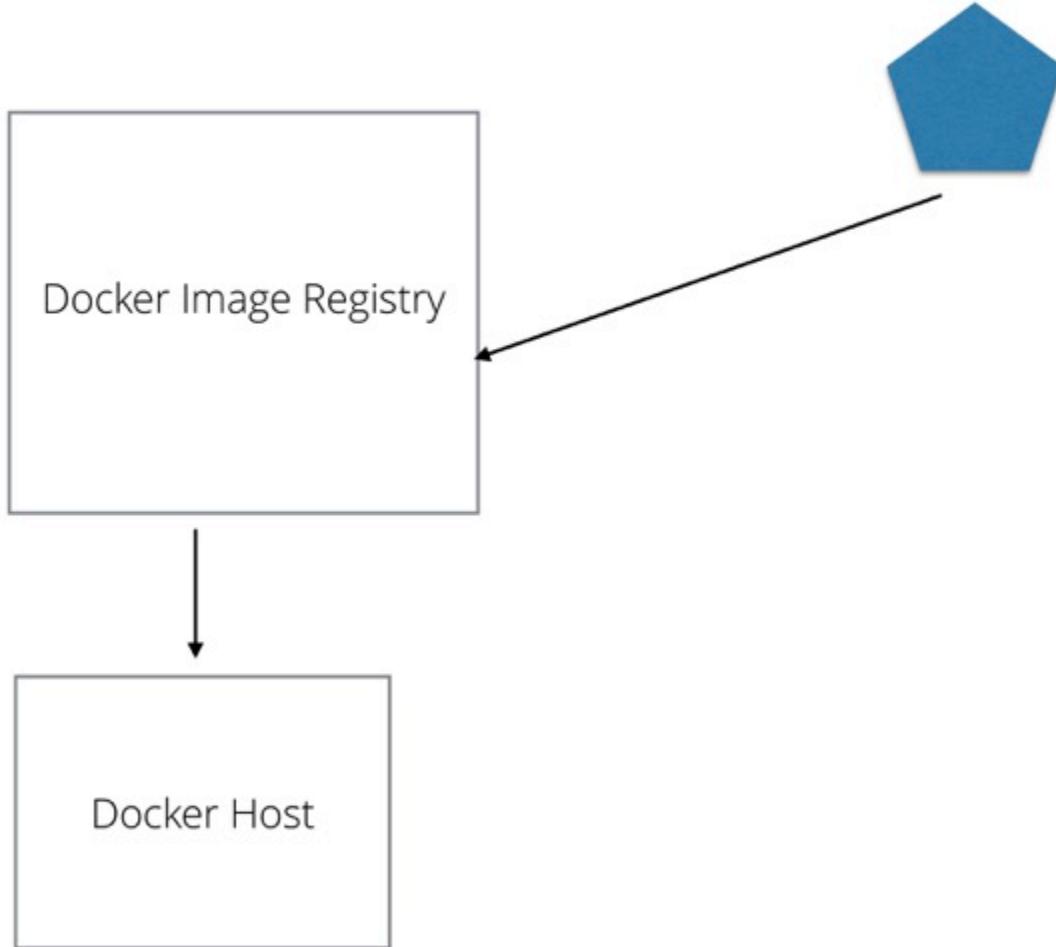
# DOCKER



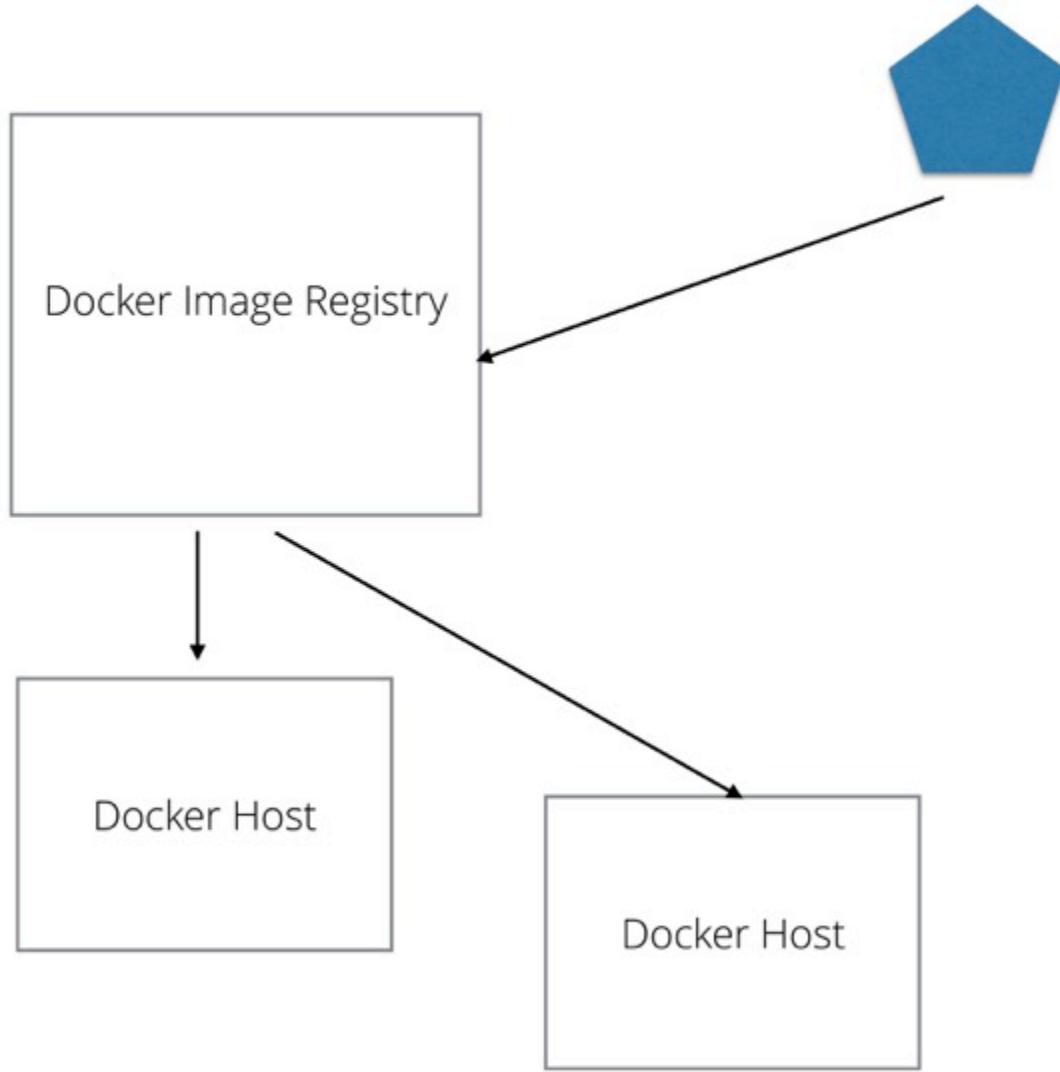
# DOCKER



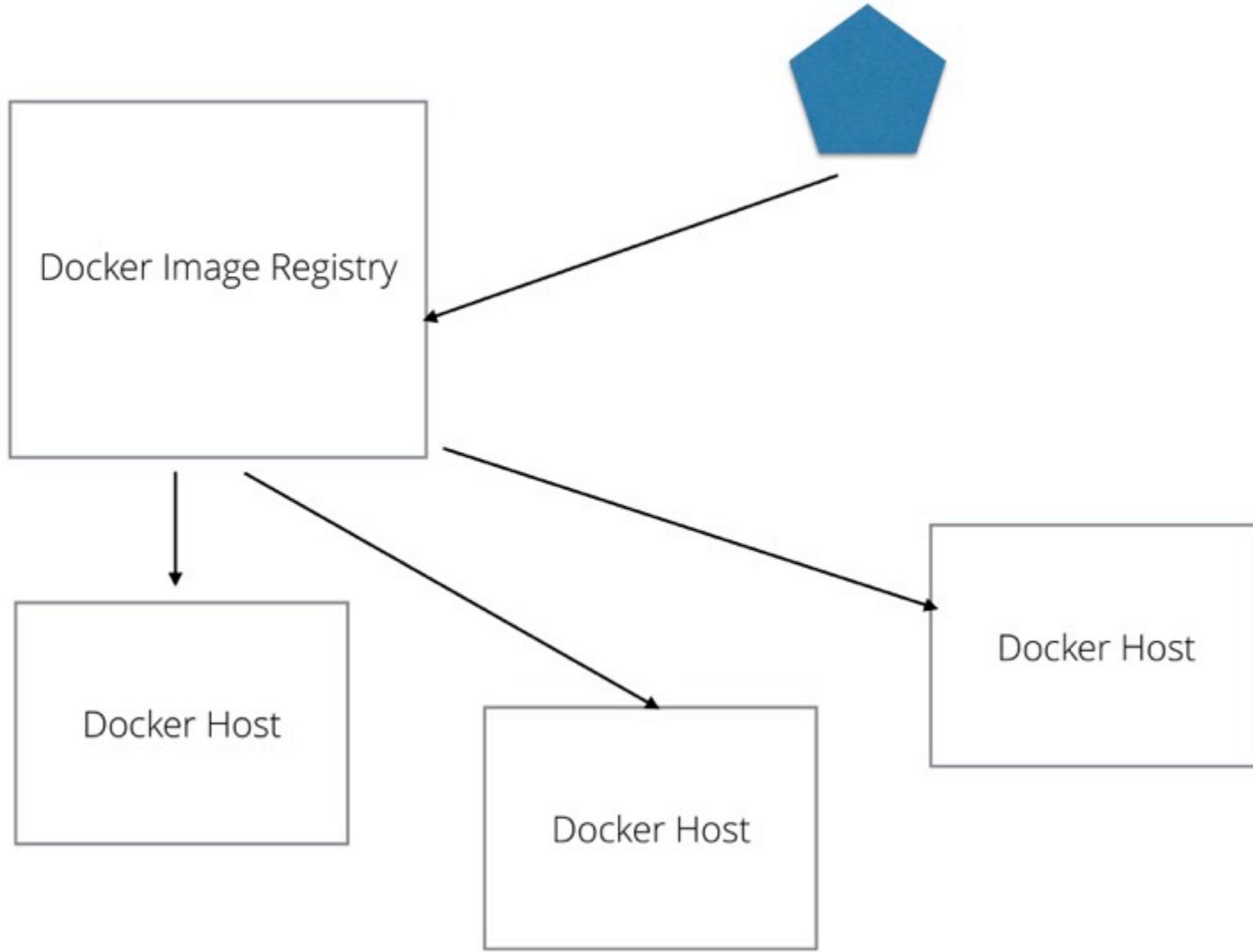
# DOCKER



# DOCKER



# DOCKER







Cost of isolated hosts is reduced...



Cost of isolated hosts is reduced...

...in terms of effort...



Cost of isolated hosts is reduced...

...in terms of effort...

...and computing resources

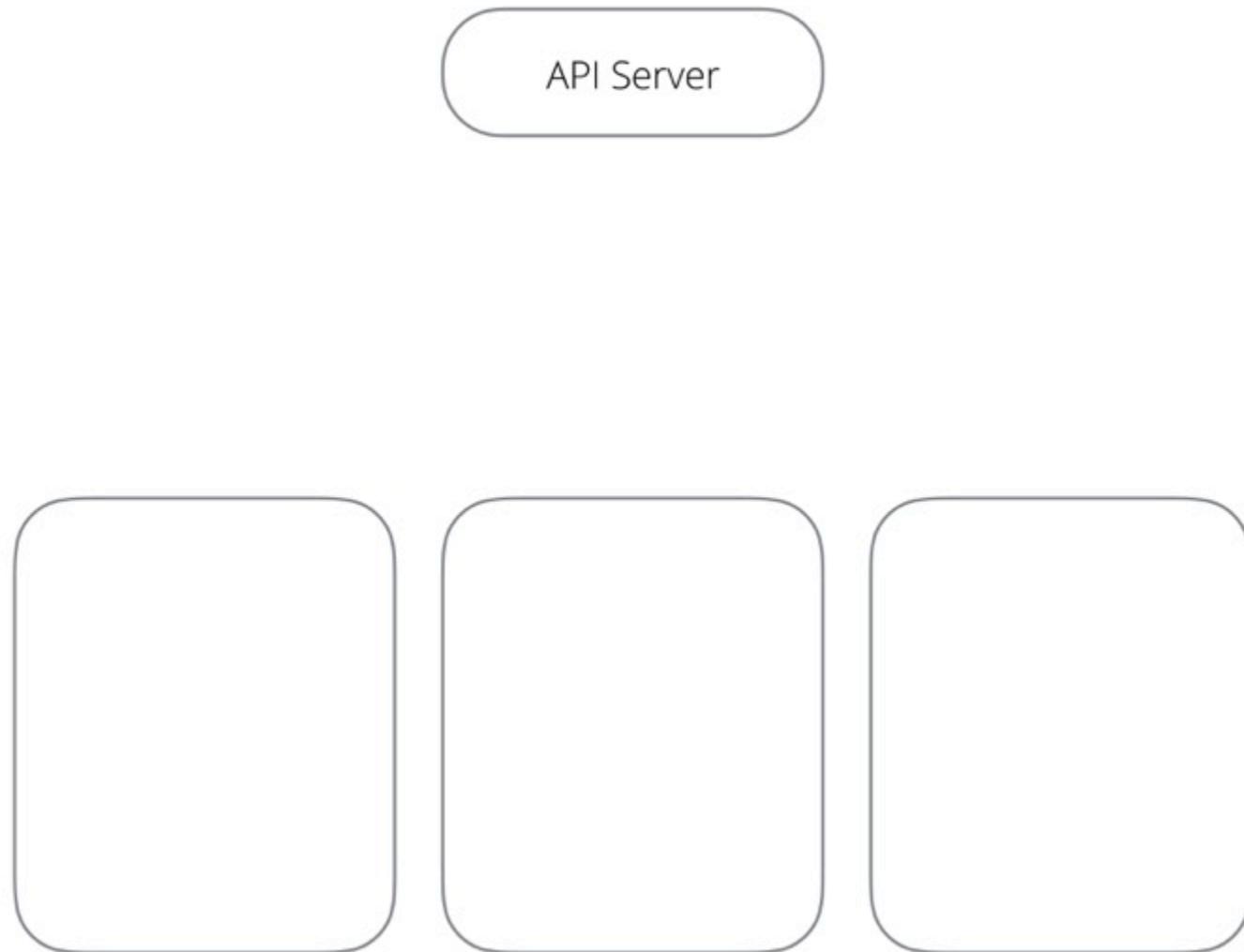


**kubernetes**  
by Google<sup>®</sup>

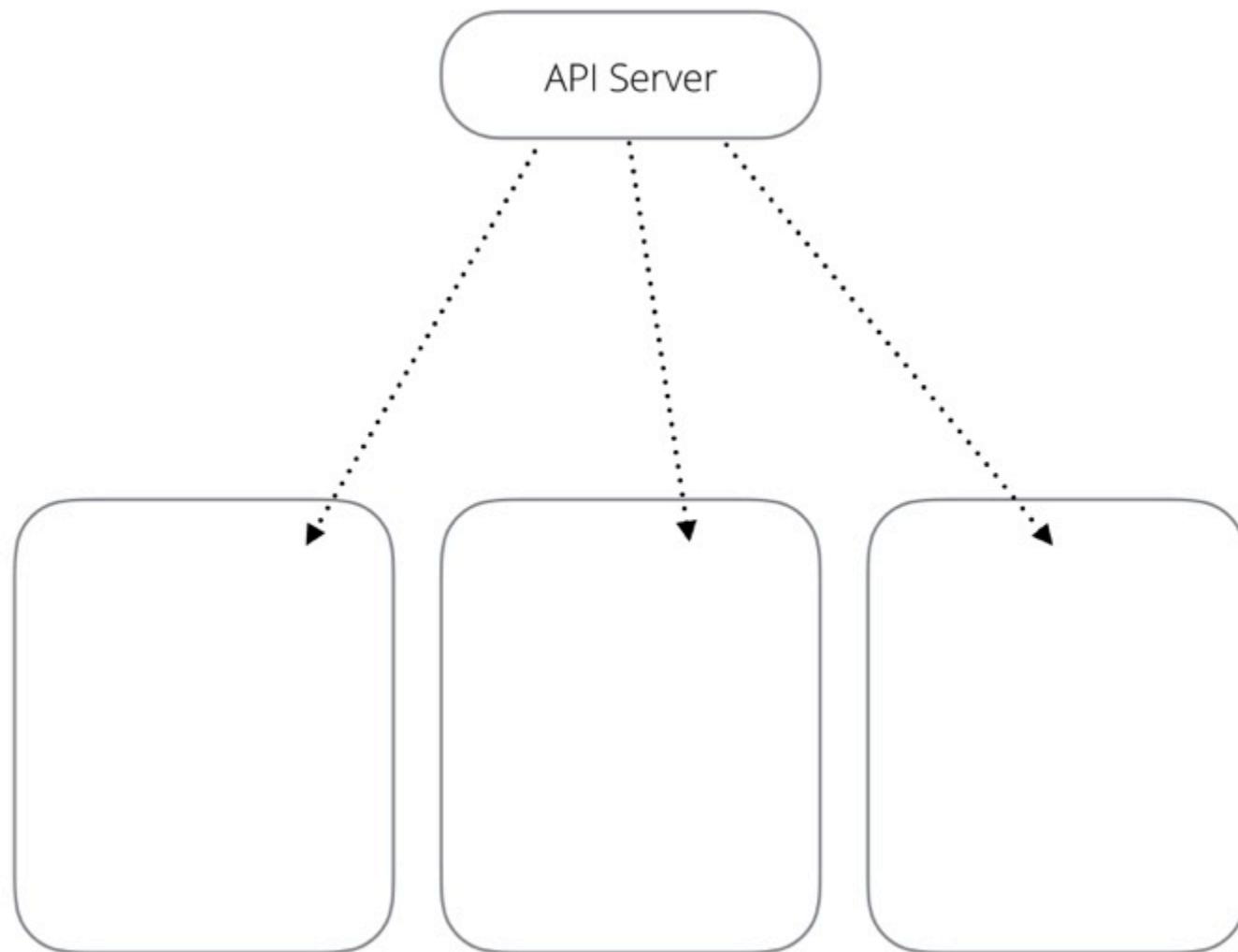
## KUBERNETES ARCHITECTURE...SORT OF



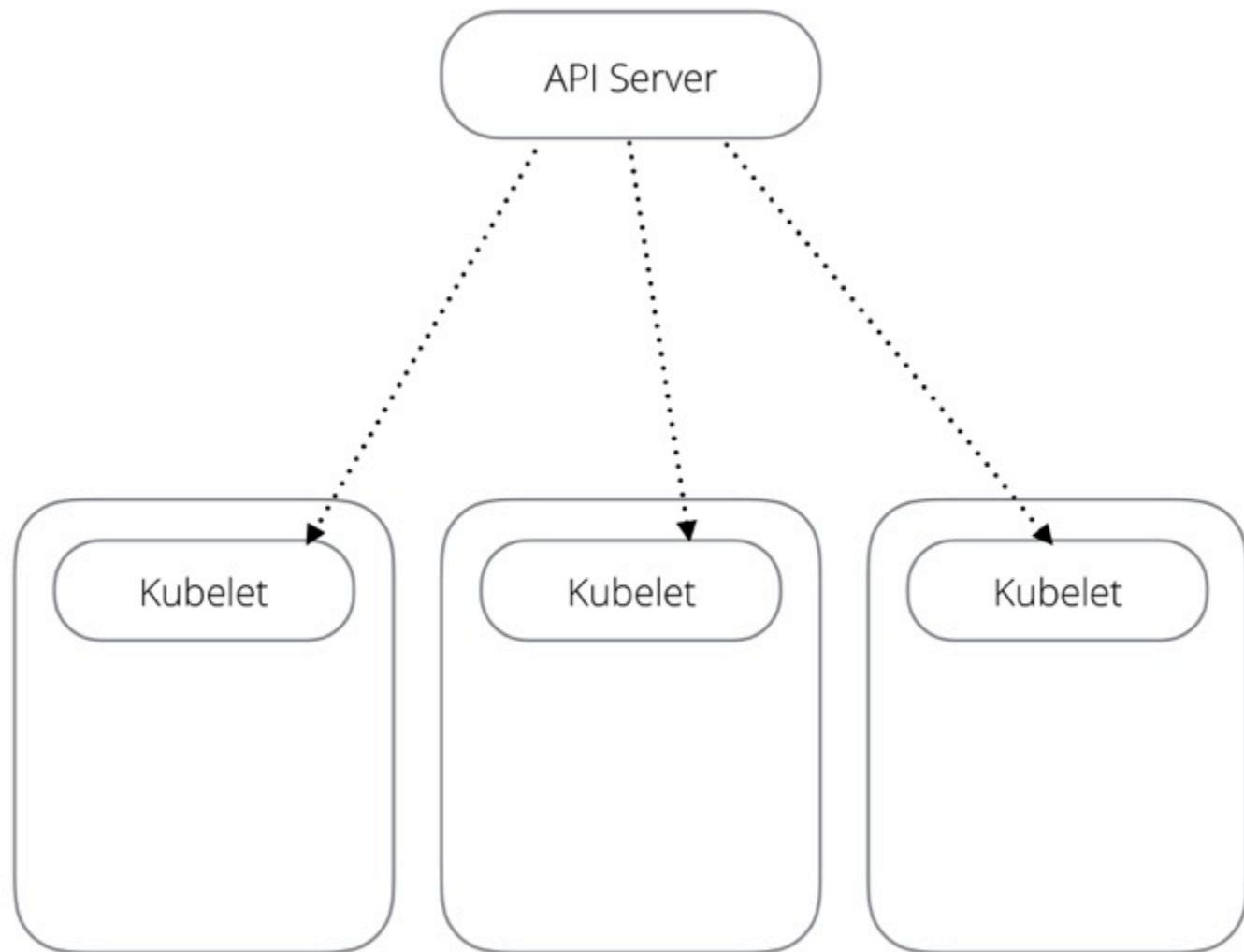
## KUBERNETES ARCHITECTURE...SORT OF



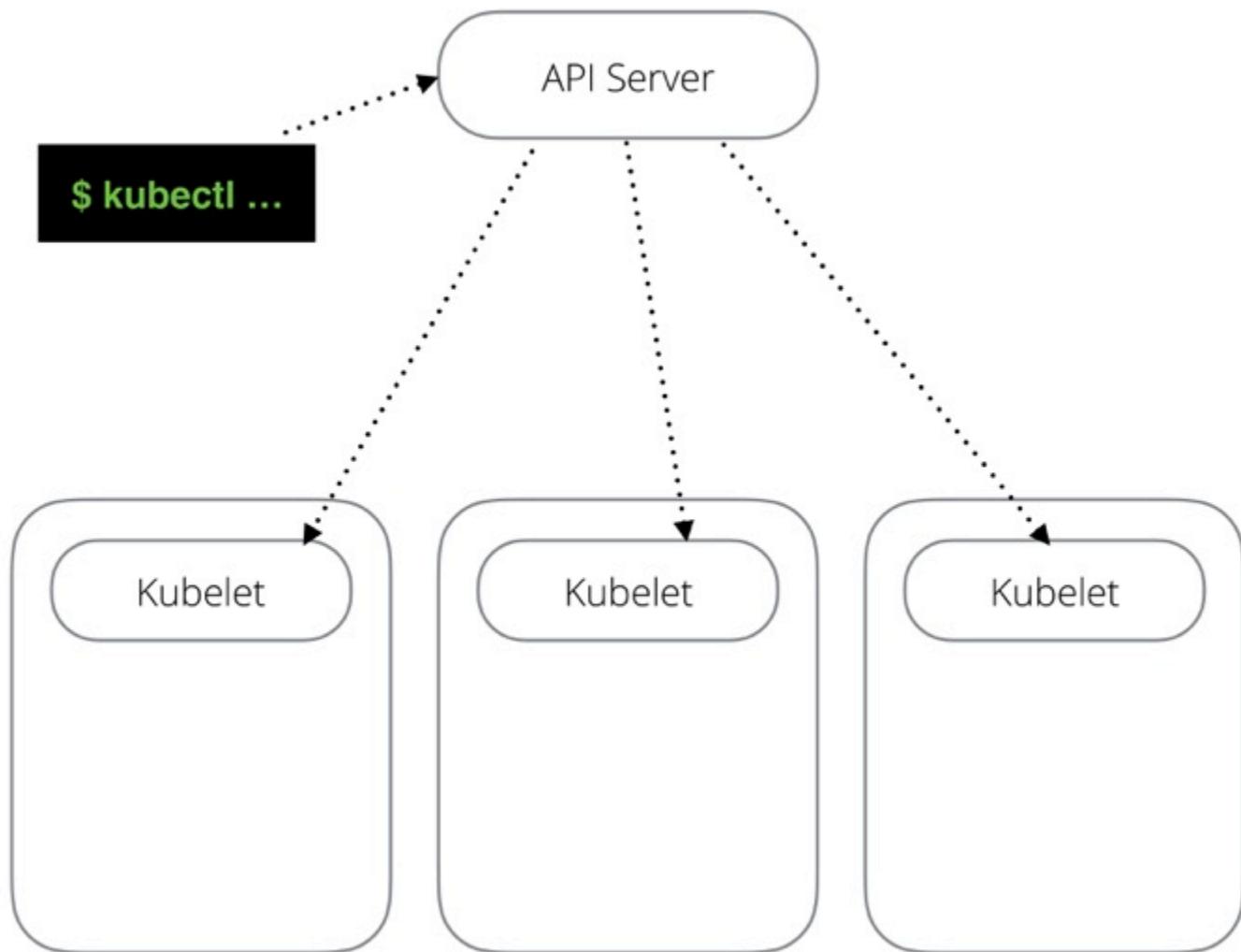
## KUBERNETES ARCHITECTURE...SORT OF



## KUBERNETES ARCHITECTURE...SORT OF

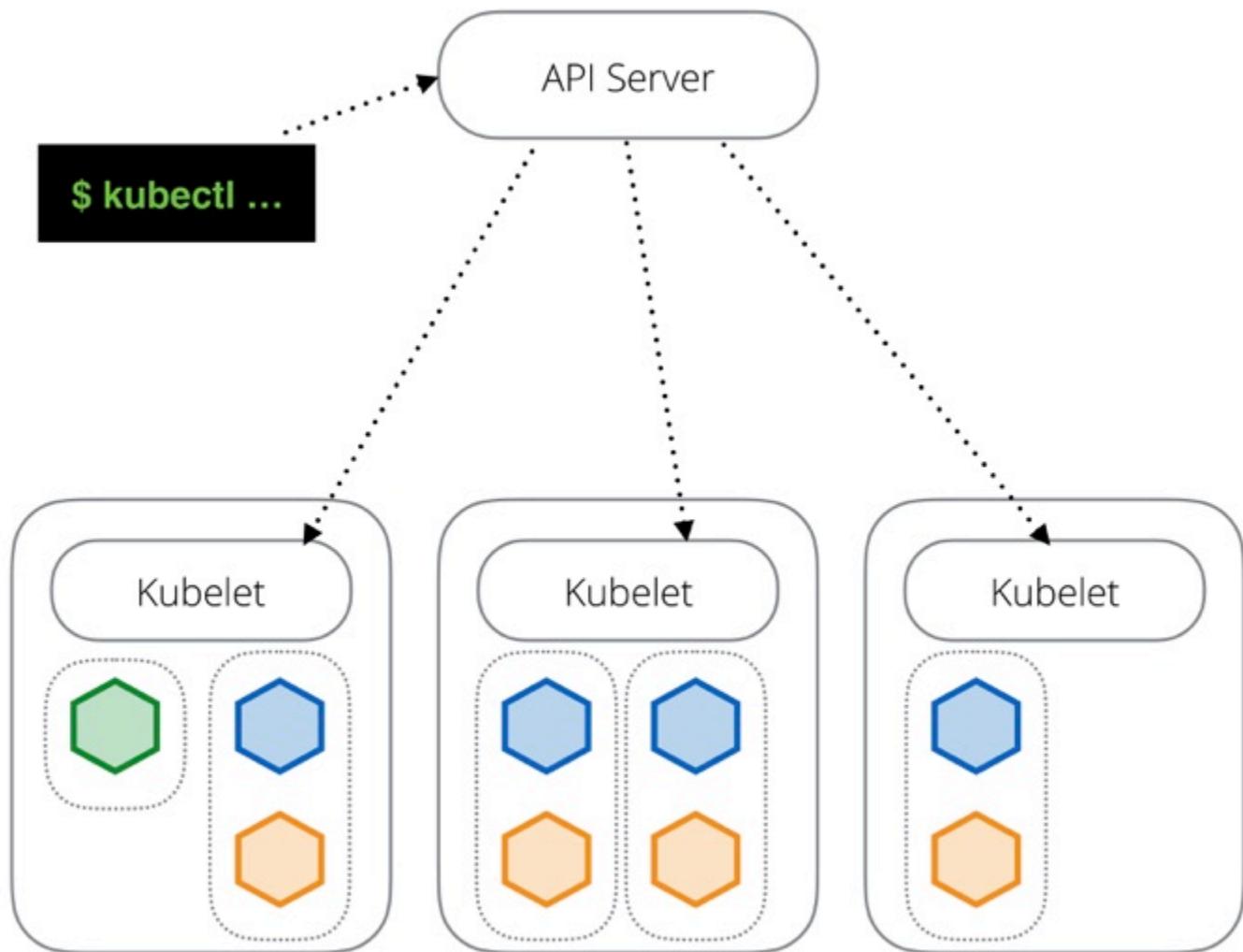


## KUBERNETES ARCHITECTURE...SORT OF

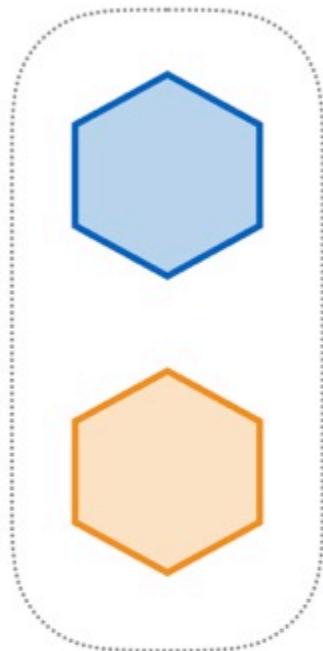


\$ kubectl ...

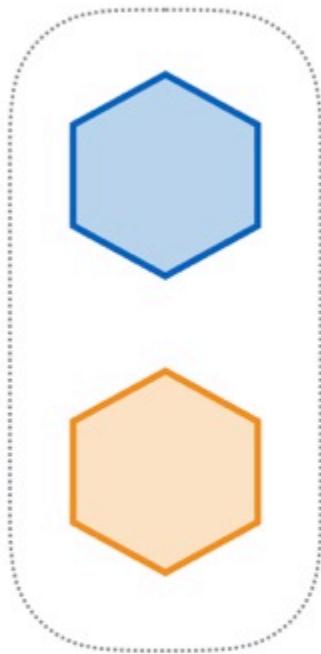
## KUBERNETES ARCHITECTURE...SORT OF



## PODS?

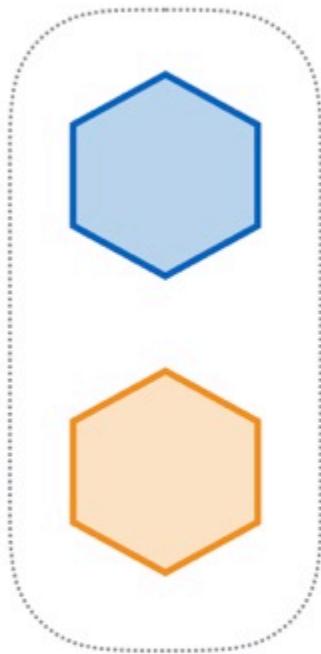


## PODS?



A collection of tightly coupled  
containers, running on one node

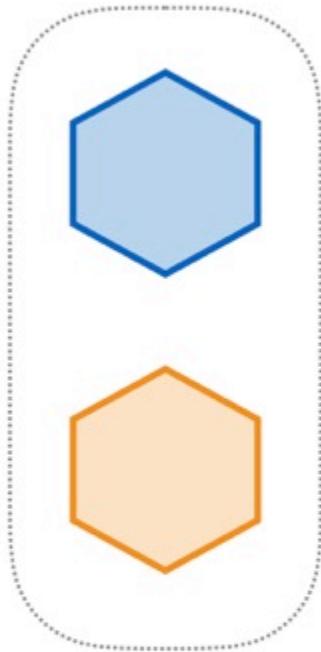
## PODS?



A collection of tightly coupled containers, running on one node

Can have metadata, volumes too

## PODS?

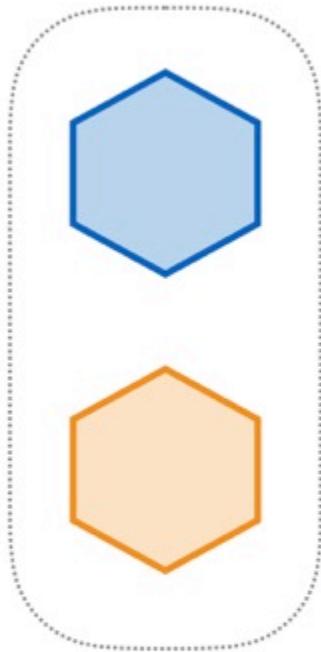


A collection of tightly coupled containers, running on one node

Can have metadata, volumes too

Pods are mortal - not long running!

## PODS?



A collection of tightly coupled containers, running on one node

Can have metadata, volumes too

Pods are mortal - not long running!

A pod = a unit of scheduling

## **SERVICES!**

## SERVICES!

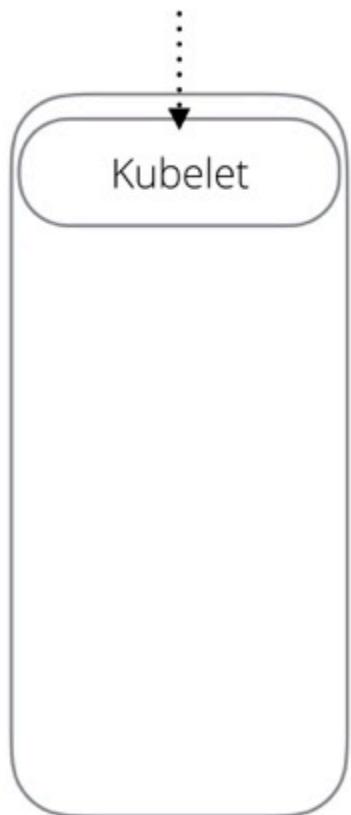
```
{  
  "kind": "Service",  
  "apiVersion": "v1",  
  "metadata": {  
    "name": "my-service"  
  },  
  "spec": {  
    "selector": {  
      "app": "MyApp"  
    },  
    "ports": [  
      {  
        "protocol": "TCP",  
        "port": 80,  
        "targetPort": 9376  
      }  
    ]  
  }  
}
```

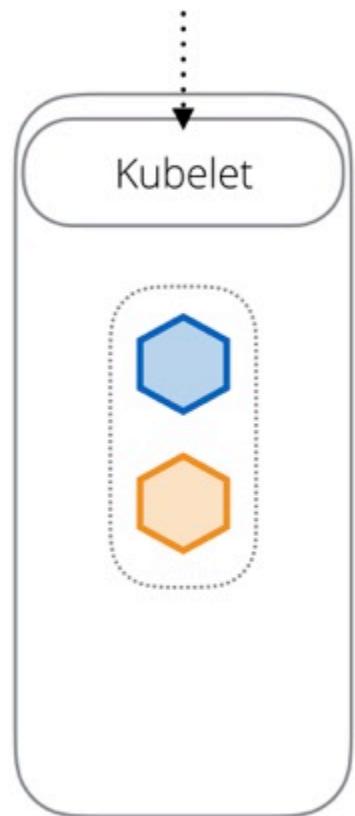
## SERVICES!

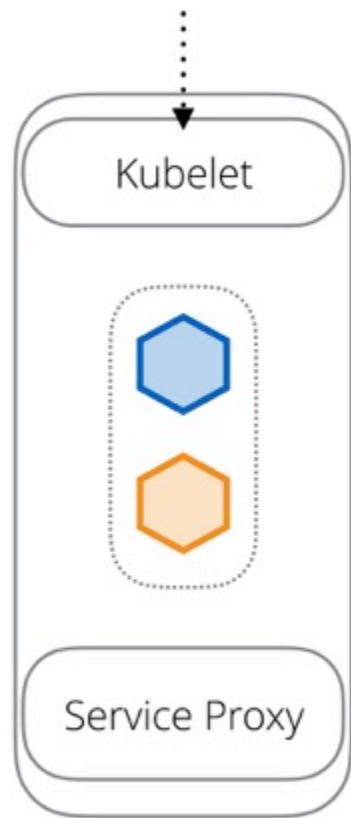
```
{  
  "kind": "Service",  
  "apiVersion": "v1",  
  "metadata": {  
    "name": "my-service"  
  },  
  "spec": {  
    "selector": {  
      "app": "MyApp"  
    },  
    "ports": [  
      {  
        "protocol": "TCP",  
        "port": 80,  
        "targetPort": 9376  
      }  
    ]  
  }  
}
```

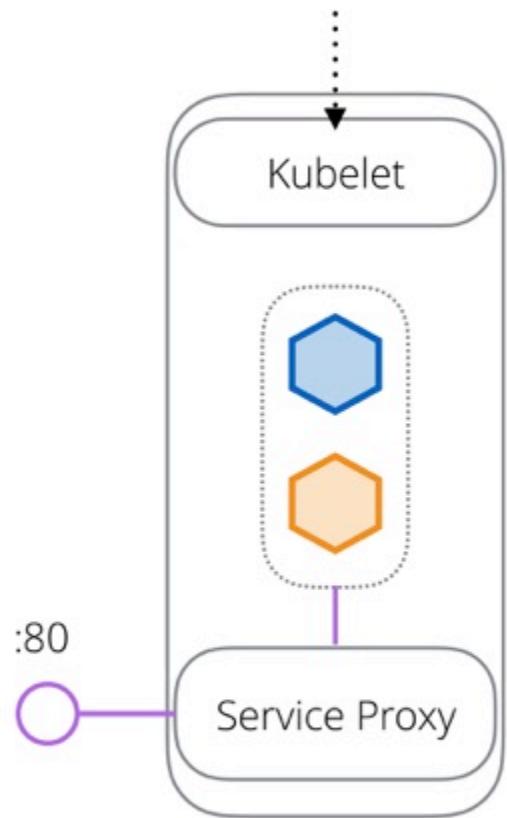
A mapping of metadata and ports to a set of pods

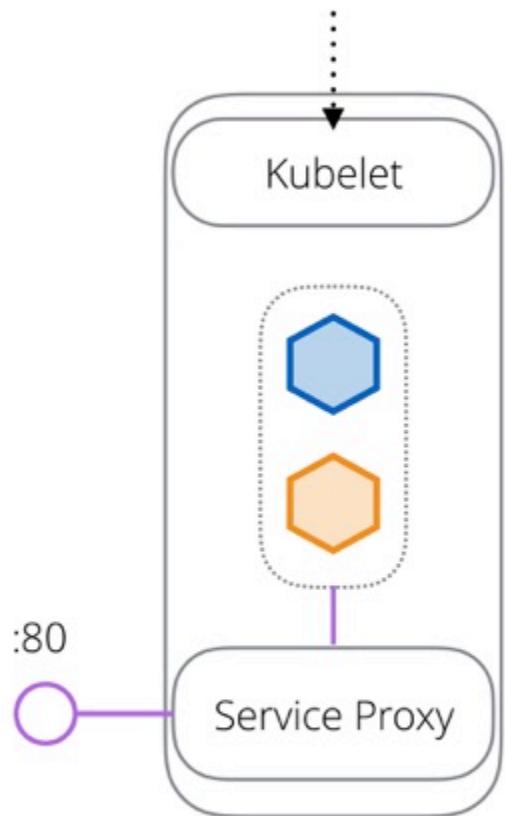




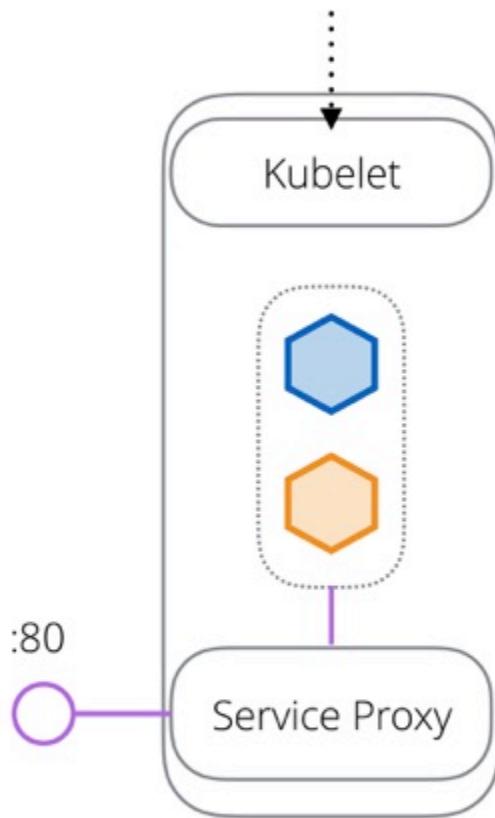








You don't scale a service...



You don't scale a service...

...you scale the pods!



### Windows Server Containers

Kubernetes version 1.5 introduces support for Windows Server Containers. In version 1.5, the Kubernetes control plane (API Server, Scheduler, Controller Manager, etc) continue to run on Linux, while the kubelet and kube-proxy can be run on Windows Server.

**Note:** Windows Server Containers on Kubernetes is an Alpha feature in Kubernetes 1.5.

### Prerequisites

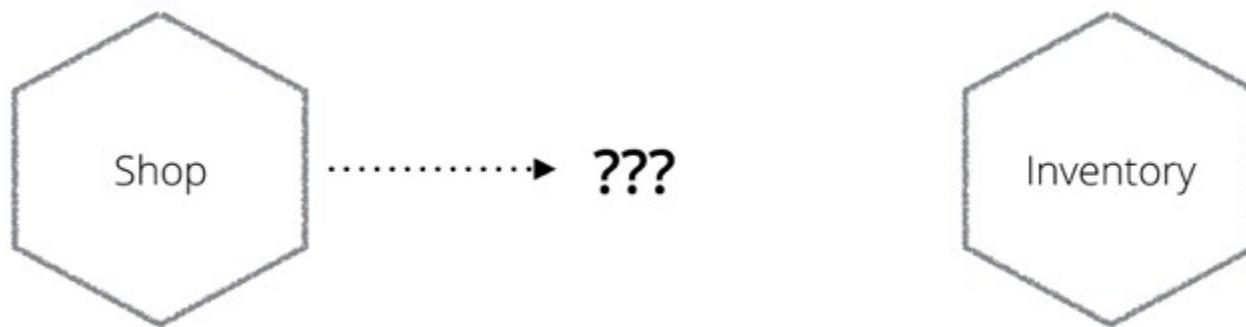
In Kubernetes version 1.5, Windows Server Containers for Kubernetes is supported using the following:

1. Kubernetes control plane running on existing Linux infrastructure (version 1.5 or later)
2. Kubelet network plugin setup on the Linux nodes
3. Windows Server 2016 (RTM version 10.0.14393 or later)
4. Docker Version 1.12.2-cs2-ws-beta or later for Windows Server nodes (Linux nodes and Kubernetes control plane can run any Kubernetes supported Docker Version)

<https://kubernetes.io/docs/getting-started-guides/windows/>

But What About Configuration?

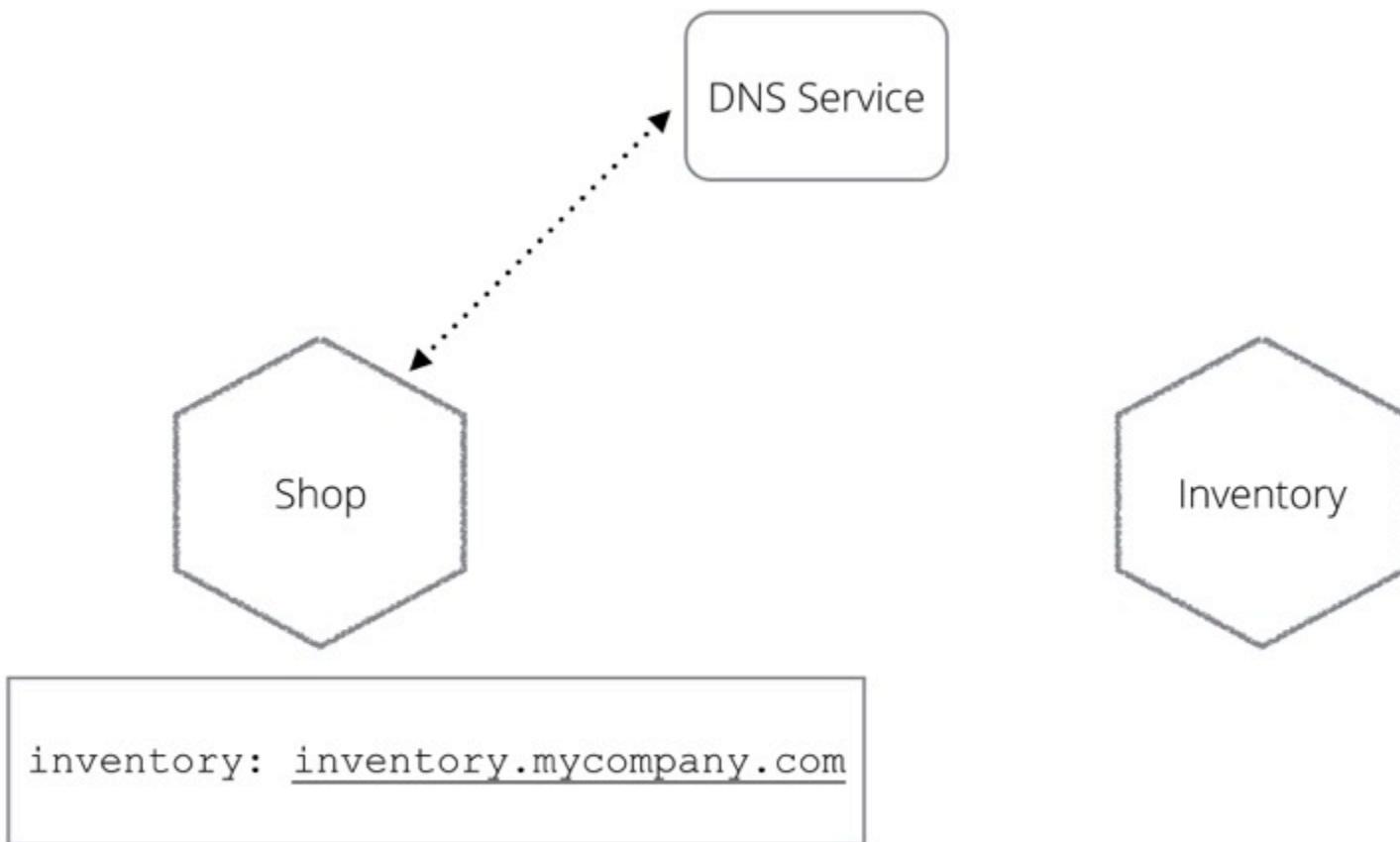
## How Do We Find Things?



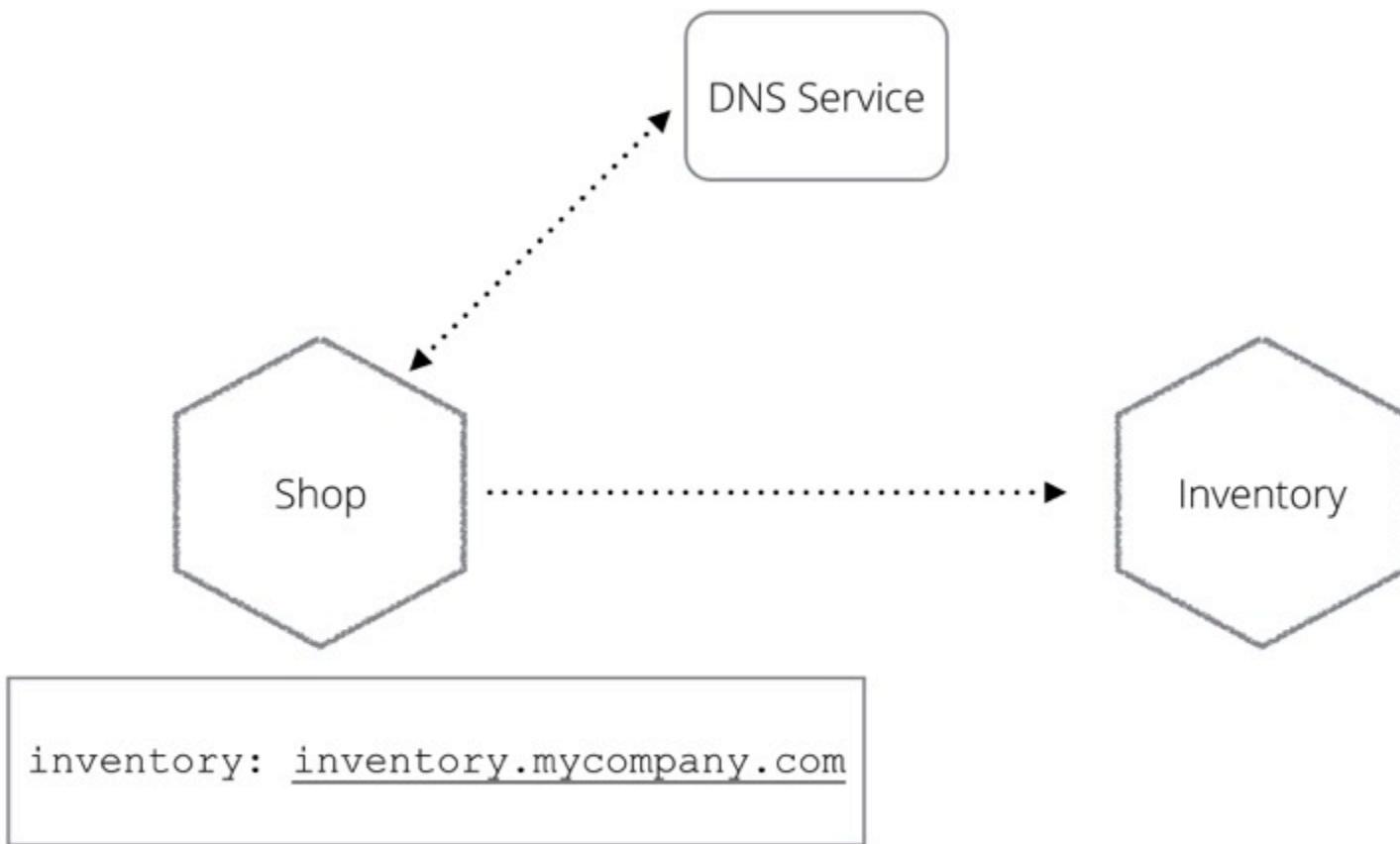
# HARDCODE IP!



# DNS

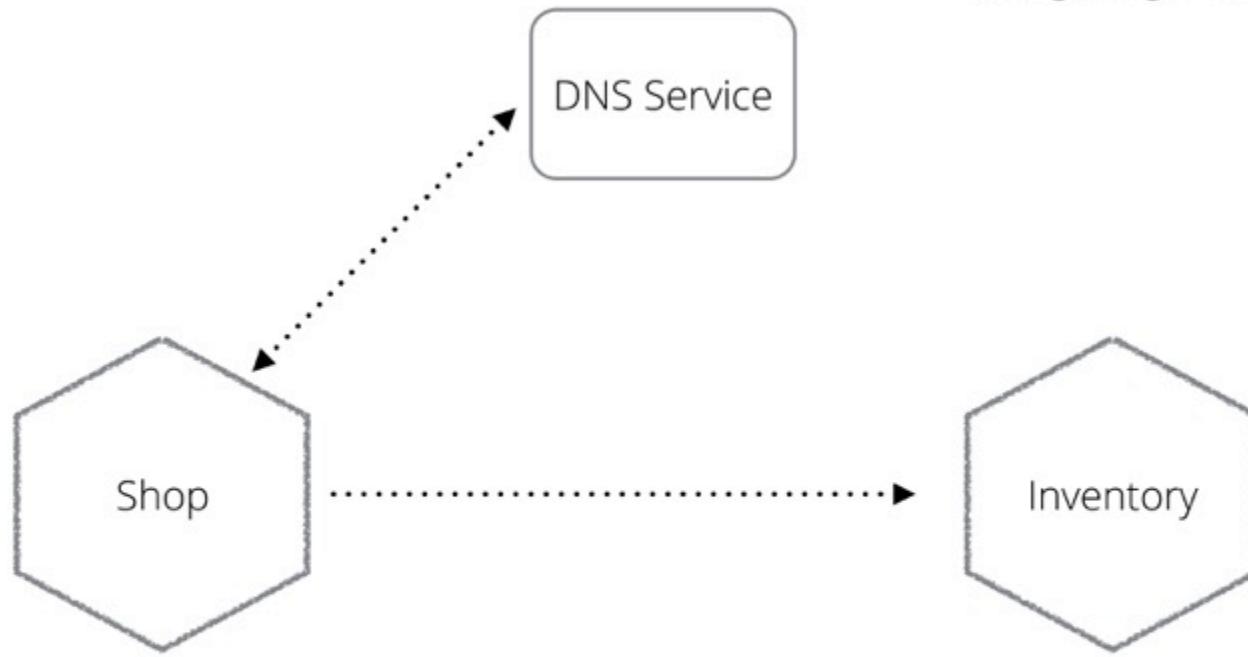


# DNS



# DNS

Configuring DNS?

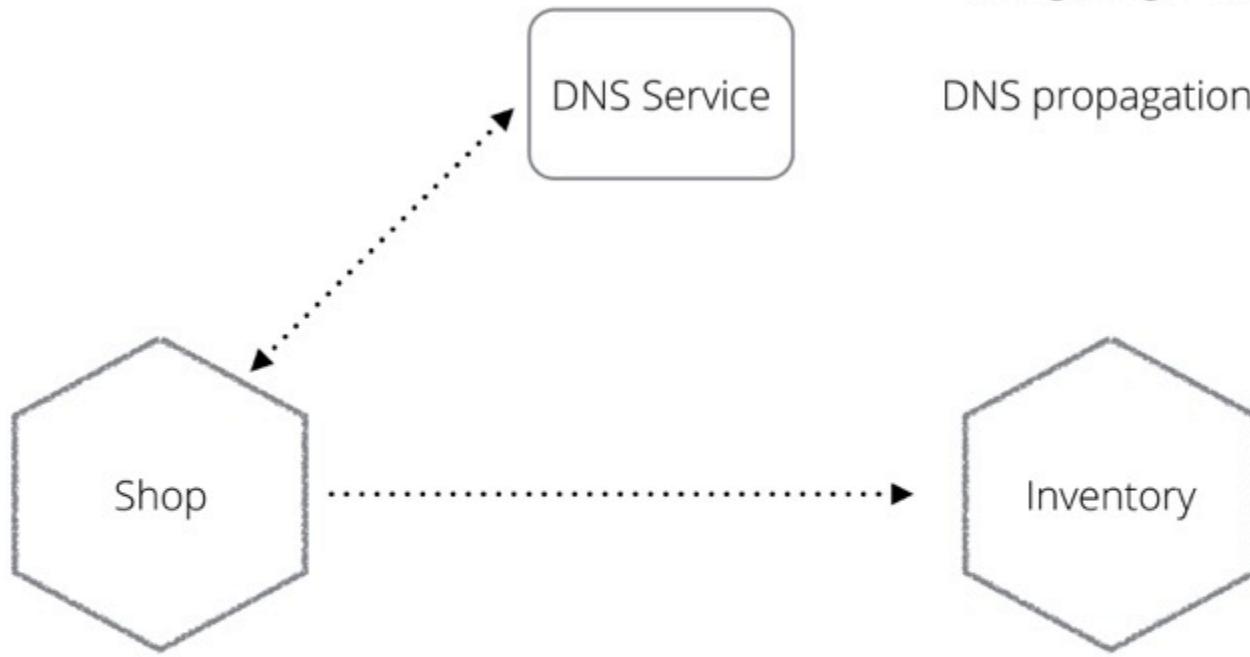


inventory: inventory.mycompany.com

# DNS

Configuring DNS?

DNS propagation time?



inventory: inventory.mycompany.com

# SERVICE DISCOVERY TOOL



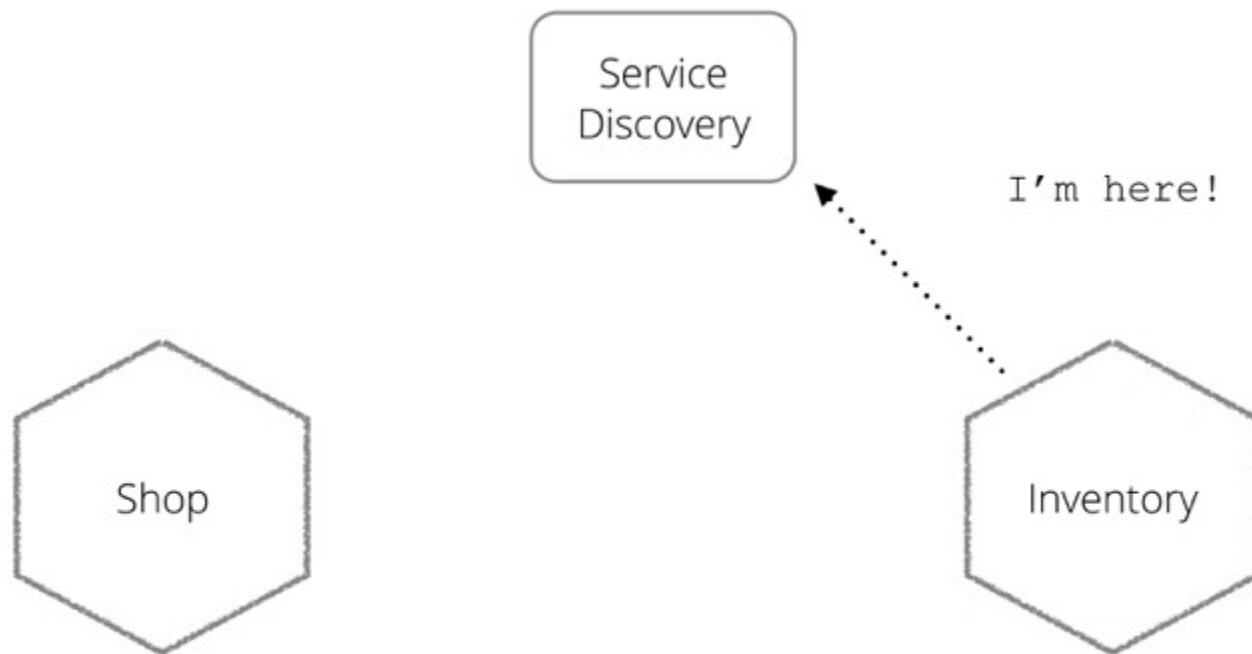
# SERVICE DISCOVERY TOOL

Service  
Discovery

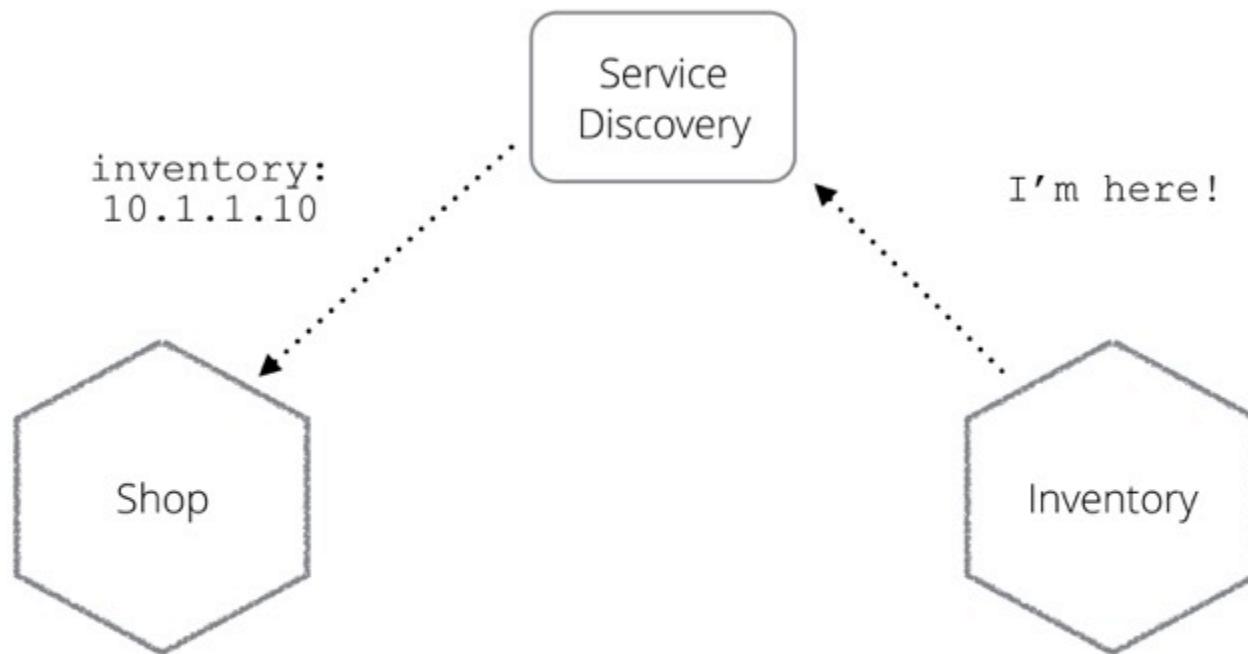
Shop

Inventory

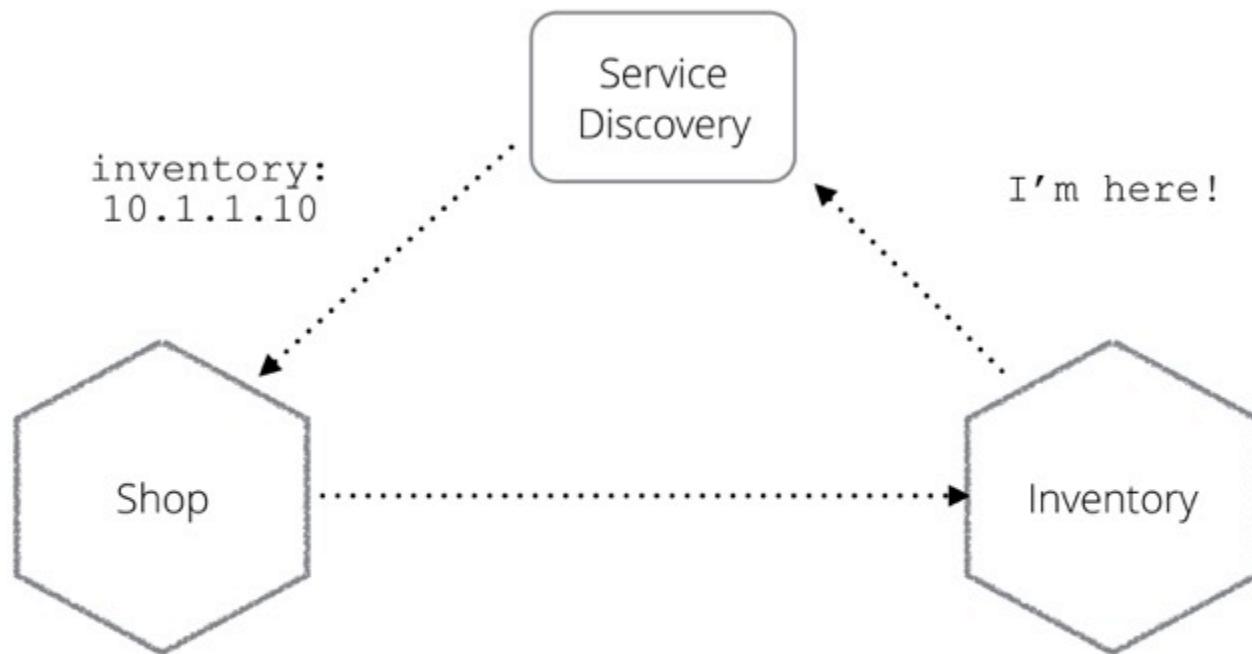
## SERVICE DISCOVERY TOOL



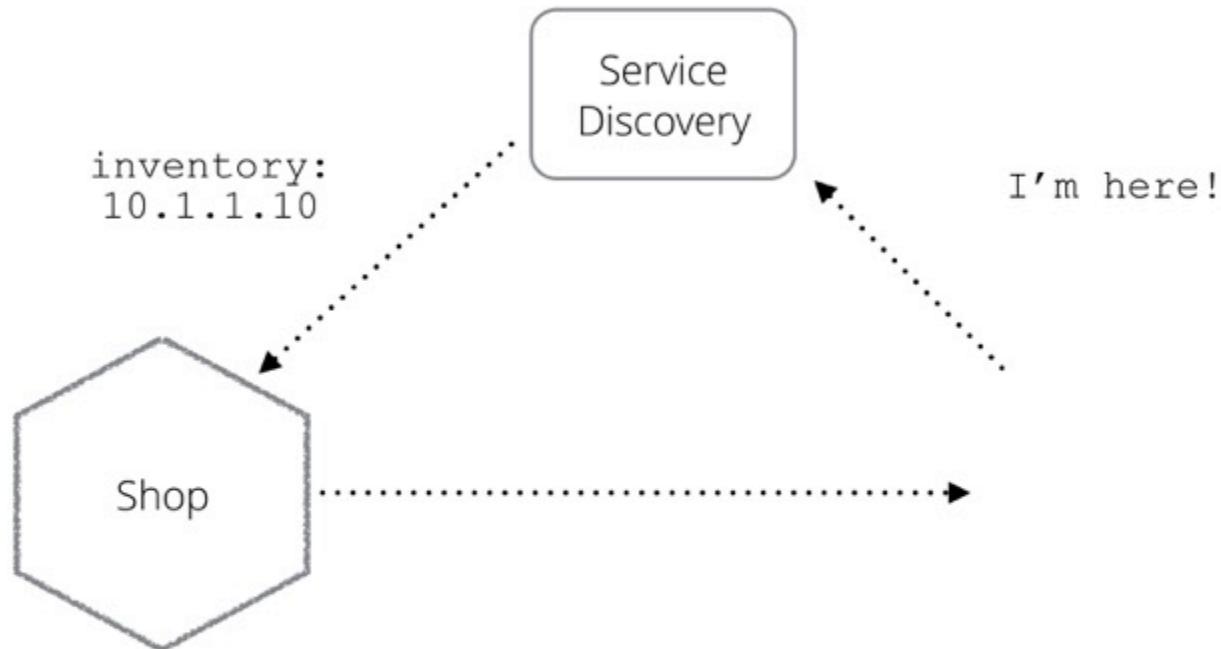
# SERVICE DISCOVERY TOOL



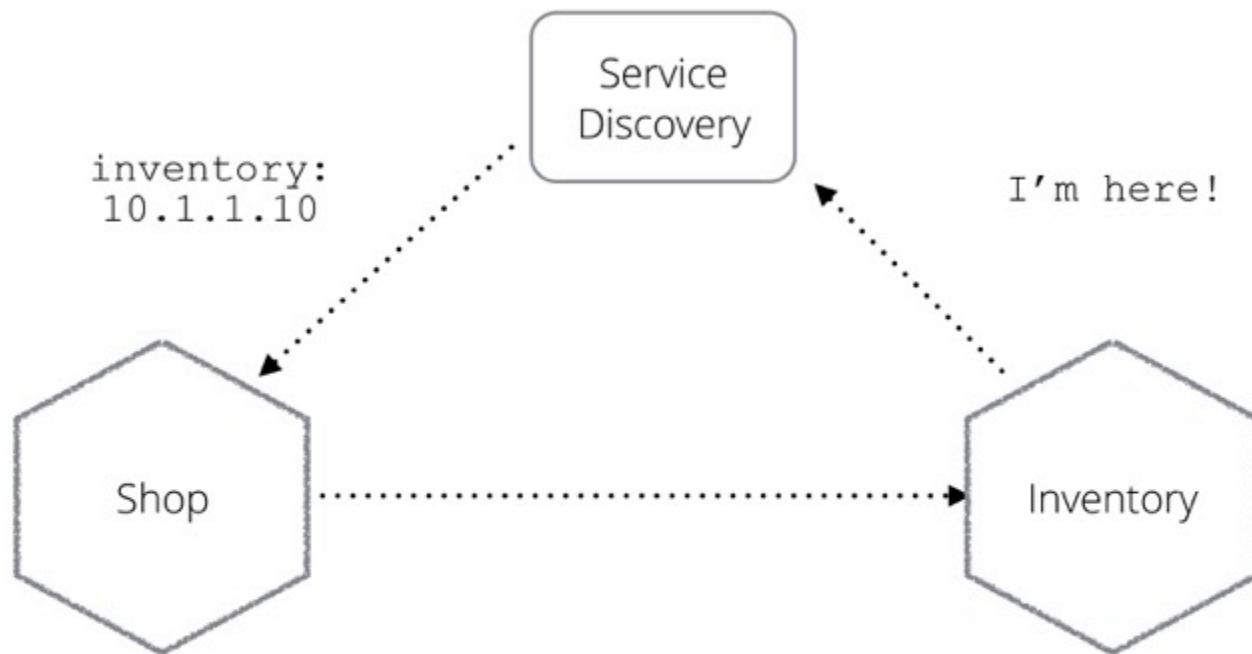
# SERVICE DISCOVERY TOOL



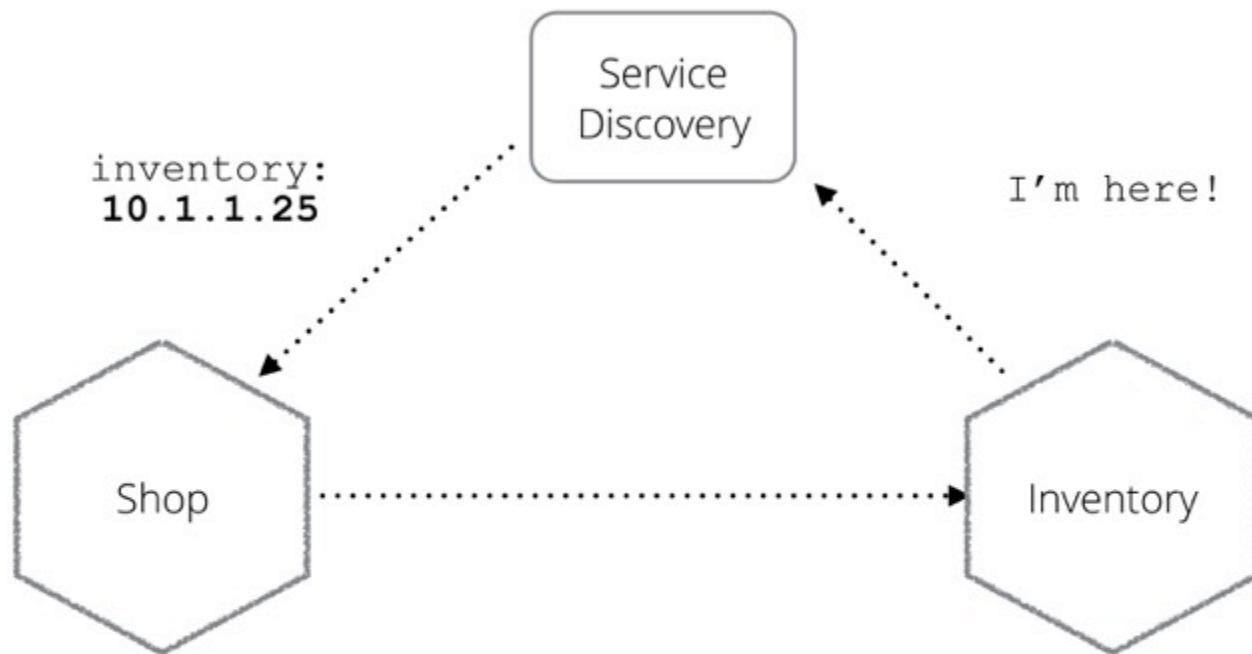
# SERVICE DISCOVERY TOOL



# SERVICE DISCOVERY TOOL



# SERVICE DISCOVERY TOOL





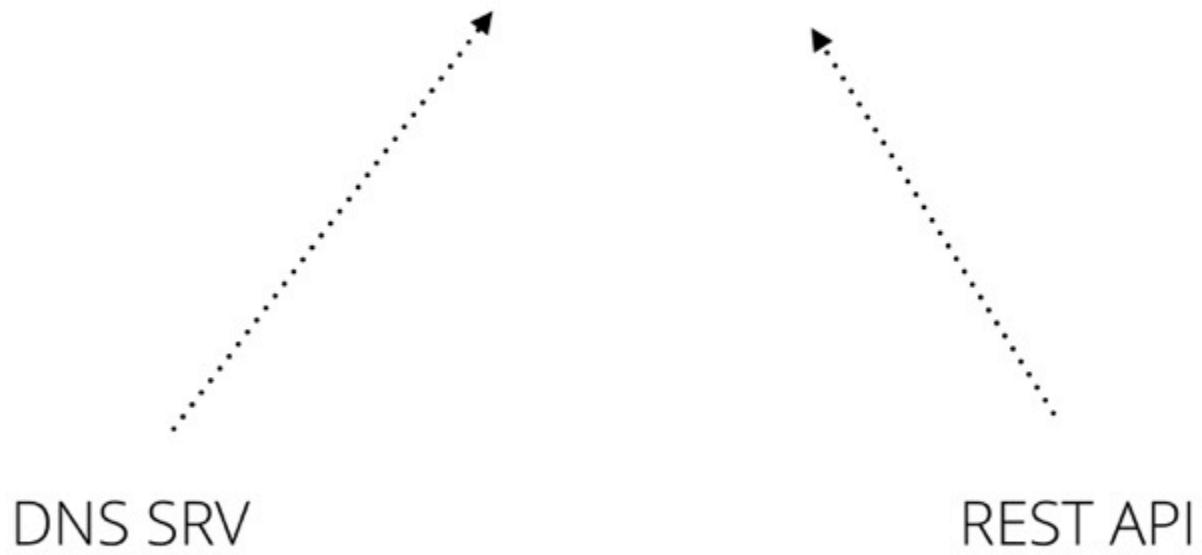








DNS SRV







## Service Discovery



Service Discovery

Configuration



Service Discovery

Configuration

Watches



Service Discovery

Configuration

Watches

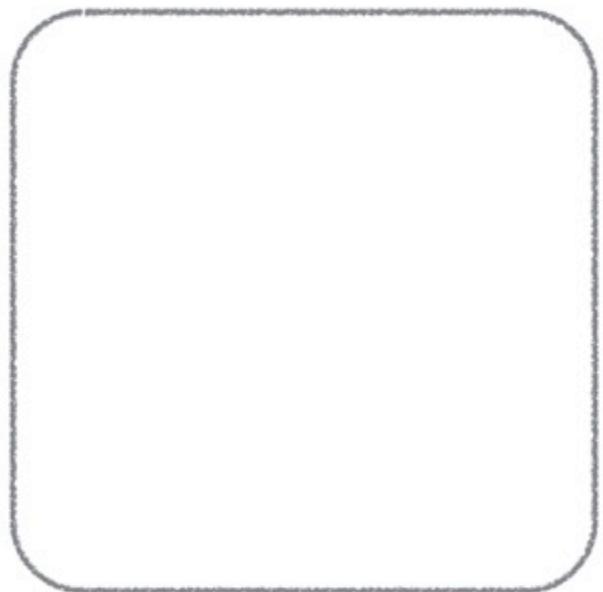
Consul-template

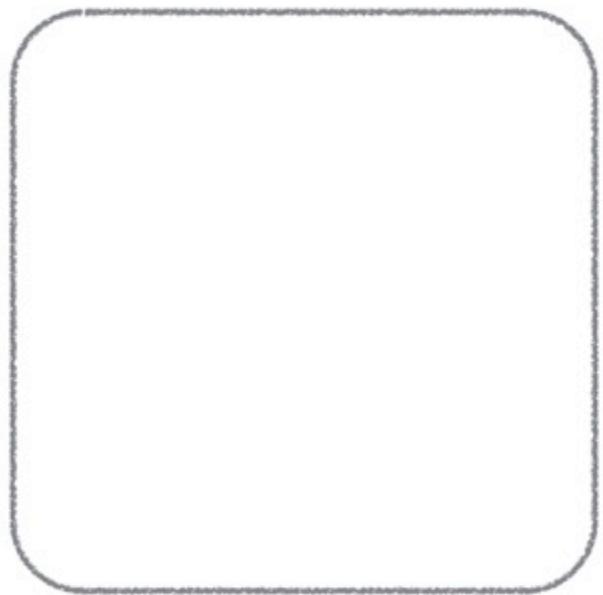
Beware of writing your own service discovery system

# MONOLITHS TO MICROSERVICES

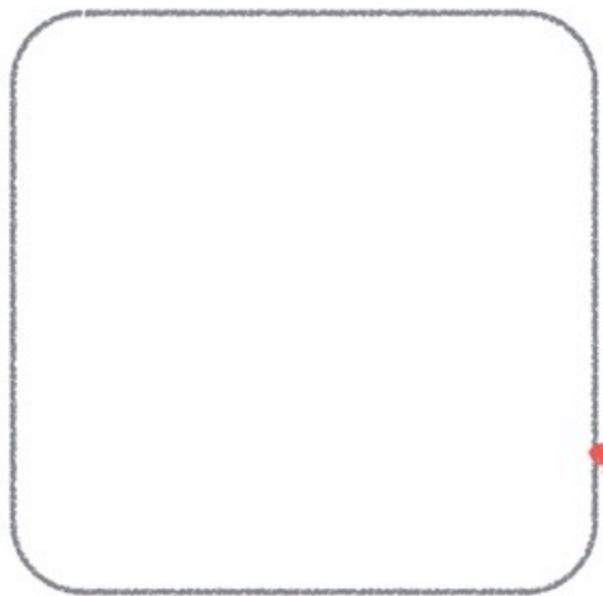
Sam Newman

## Monitoring & Alerting





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



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





**Honest Status Page**

@honest\_update

 Follow

We replaced our monolith with micro services so that every outage could be more like a murder mystery.

RETWEETS

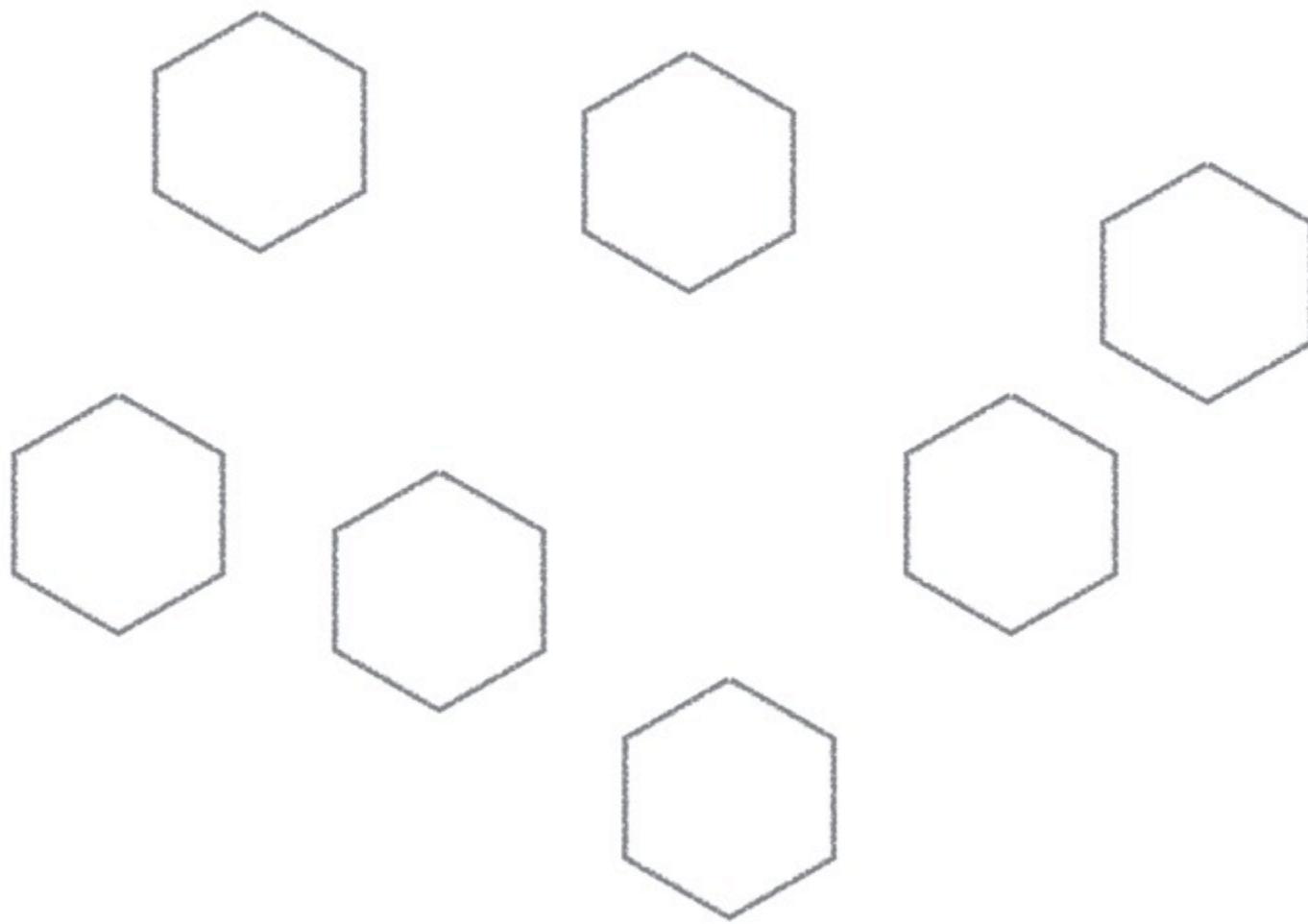
2,880

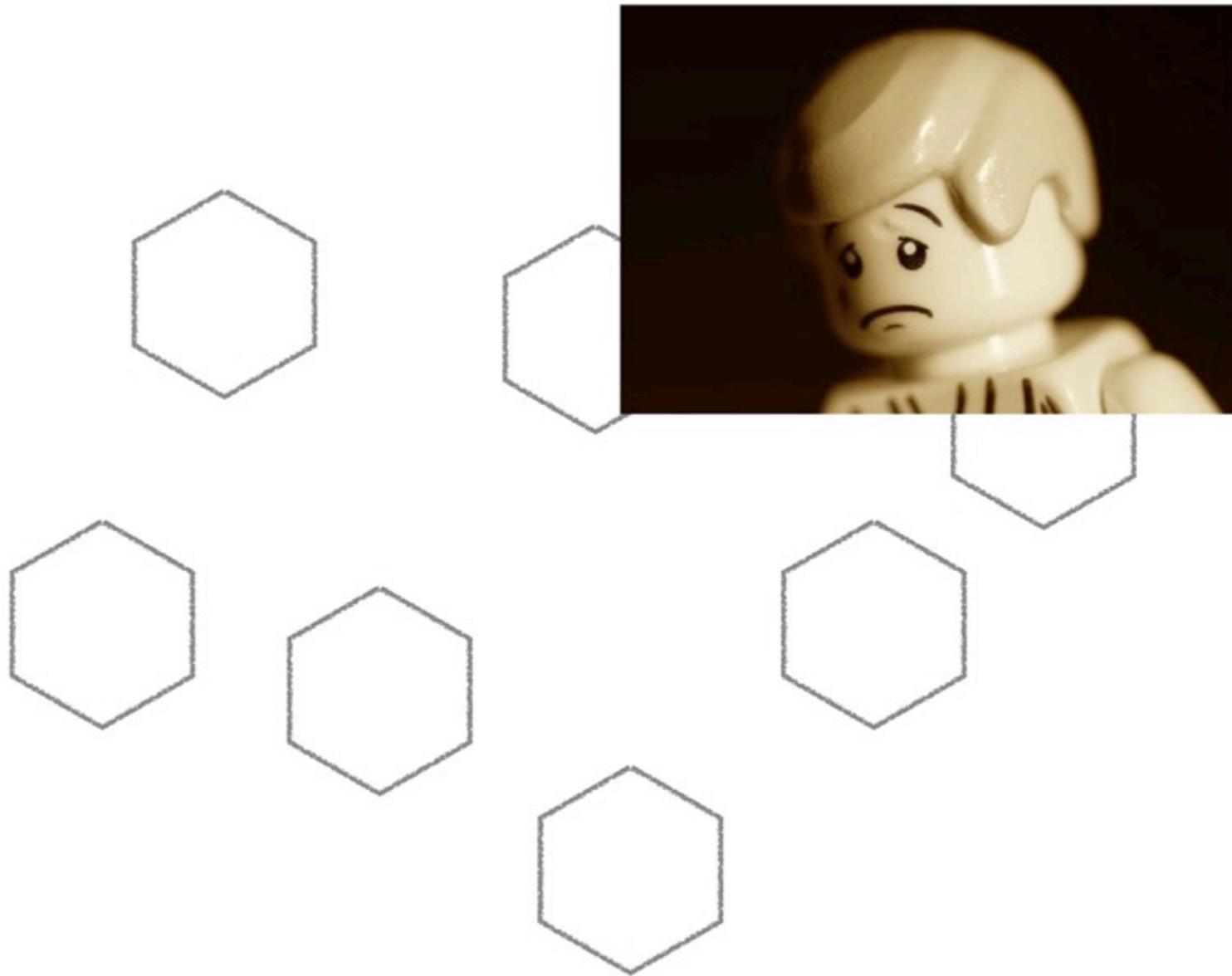
LIKES

2,261



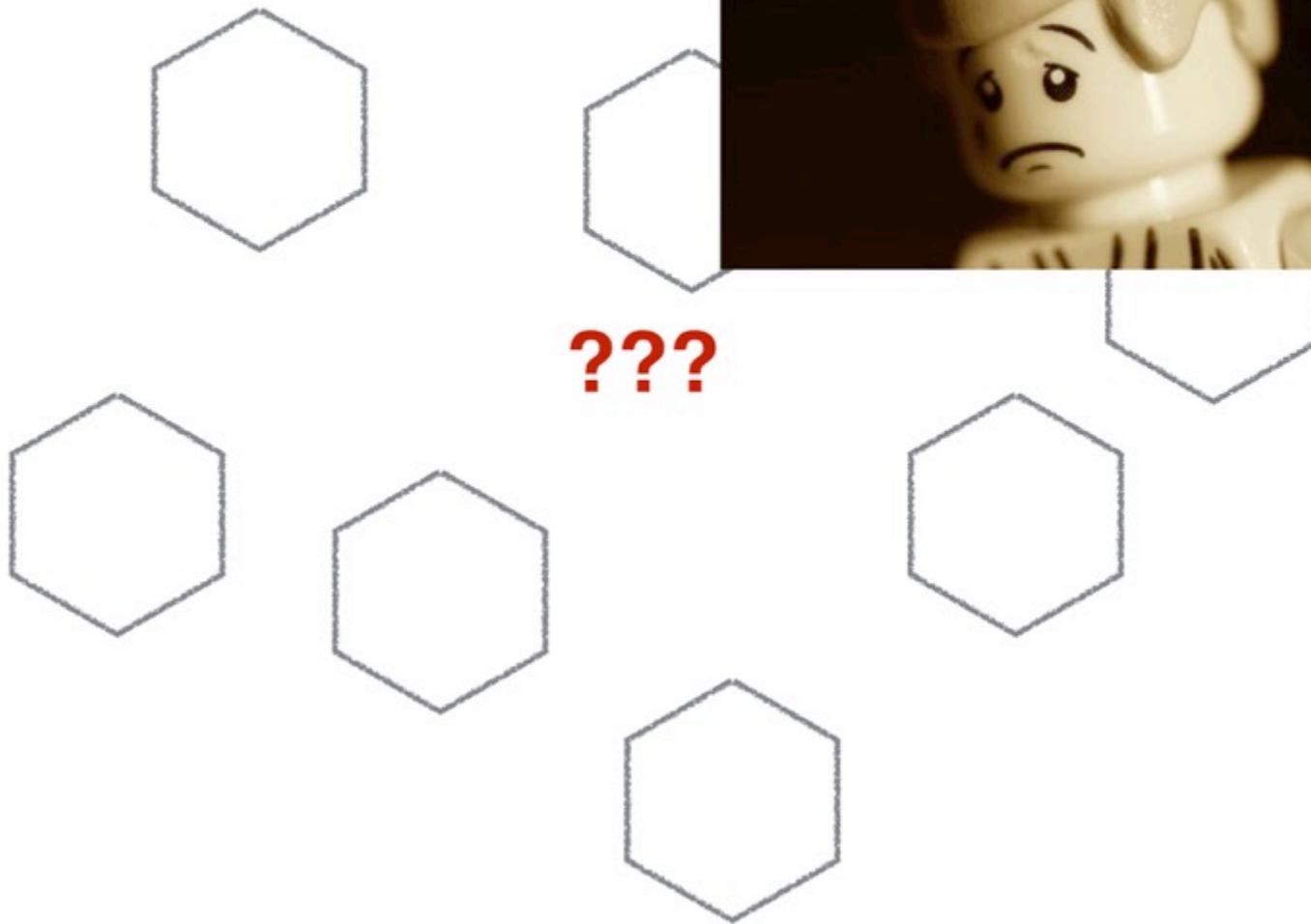
[https://twitter.com/honest\\_update/status/651897353889259520](https://twitter.com/honest_update/status/651897353889259520)



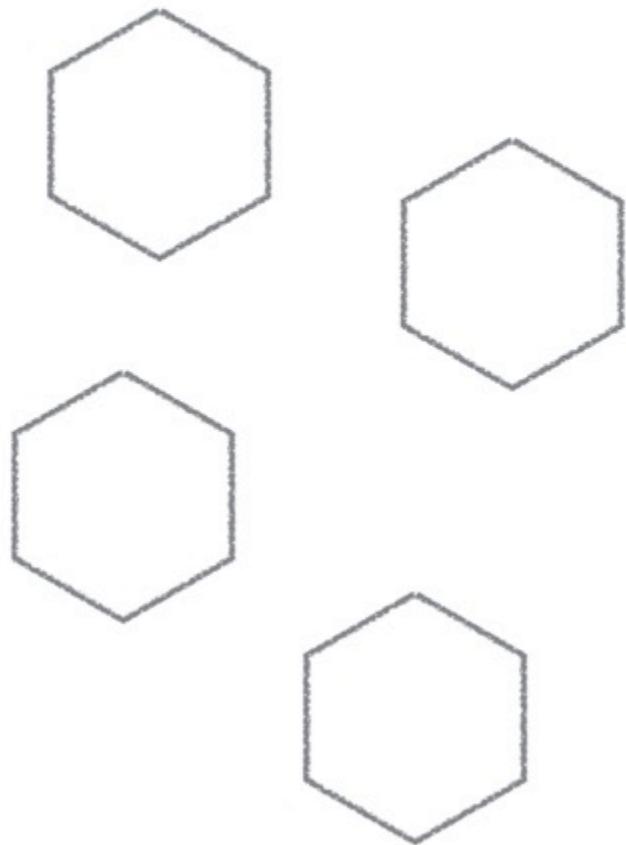


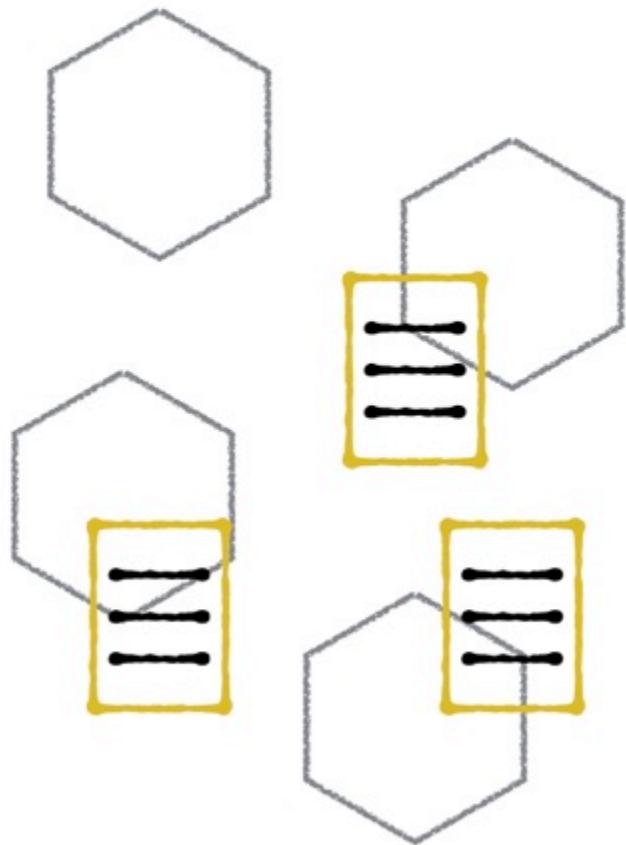


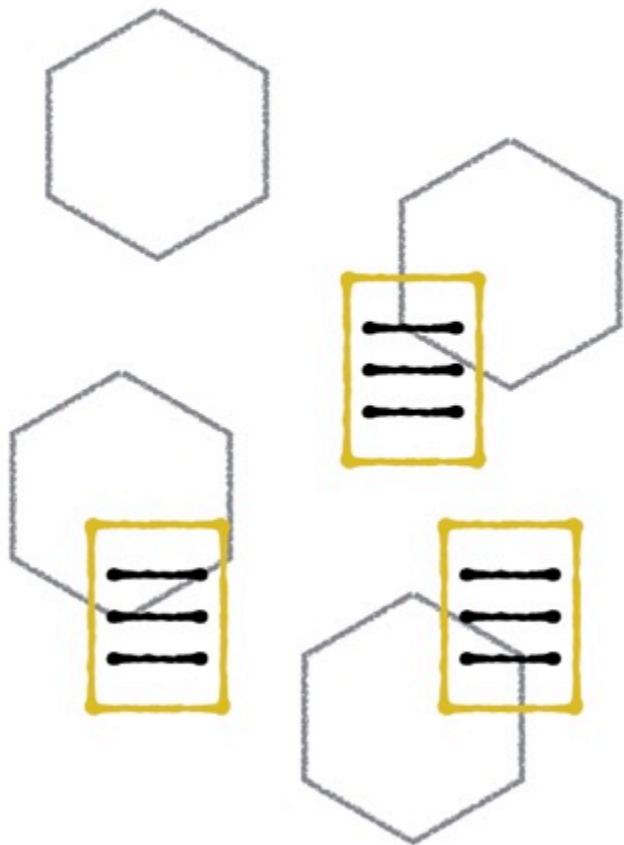
???



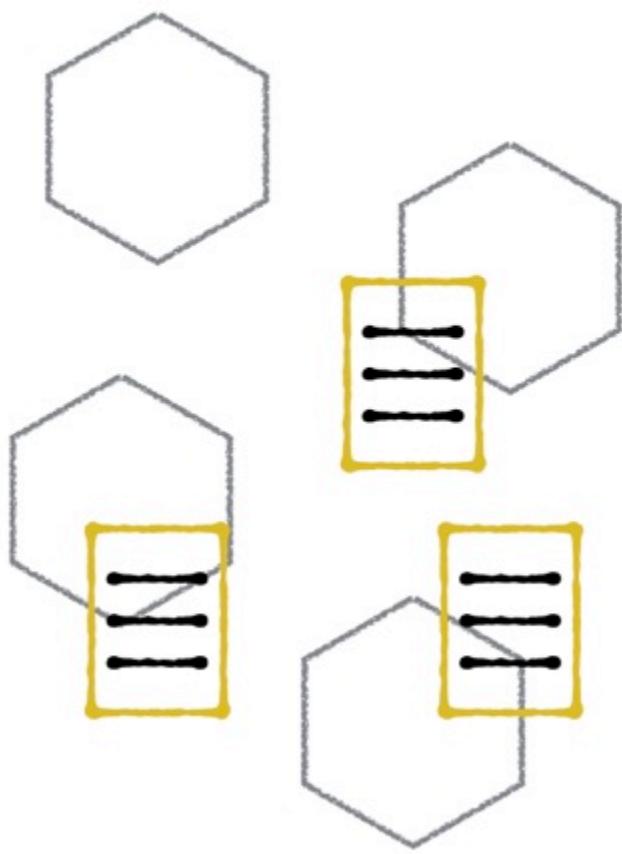
You have to get \*much\*  
better at monitoring







	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100



A screenshot of a terminal window showing multiple command-line sessions. The terminal has a black background with green text. A large red 'X' is drawn across the entire window, indicating that the displayed content is incorrect or irrelevant.

# Too many processes

Too many processes

Machines are ephemeral

Extract information from hosts and services, and aggregate it centrally

The first thing you should do:

## **Log Aggregation**



logstash

logstash is a tool for managing events and log data in real time. It can pull events from multiple sources, transform them on the fly, and store them for later use (like, for search with a web interface for searching and drilling).

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

[Download](#)

[Cookbook](#)



The image shows a composite screenshot of the Kibana landing page. On the left, there's a cartoon illustration of a tree stump with large white eyes and a black mustache, resembling the famous Logstash wood character. To the right of the character is the Kibana logo, which consists of a green square icon followed by the word "Kibana" in a large, bold, white serif font. Below the main title, the tagline "Make sense of a mountain of logs" is displayed, followed by a red button that says "Now in Ruby!". A prominent green "Get Started" button is centered below the tagline. At the bottom of the main section, there are links to "GitHub project", "Logstash", and "ElasticSearch". Below this, there are two small rectangular buttons for "Star" (with 1,198 stars) and "Fork" (with 287 forks). The background of the main section features a blurred image of a forest scene.

**logstash**

logstash is a tool for managing events and log data in real time. It can receive events from many different sources, store them, and then make them available for later use (like, for search with a web interface for searching and drilling).

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 want.

# Kibana

Make sense of a mountain of logs Now in Ruby!

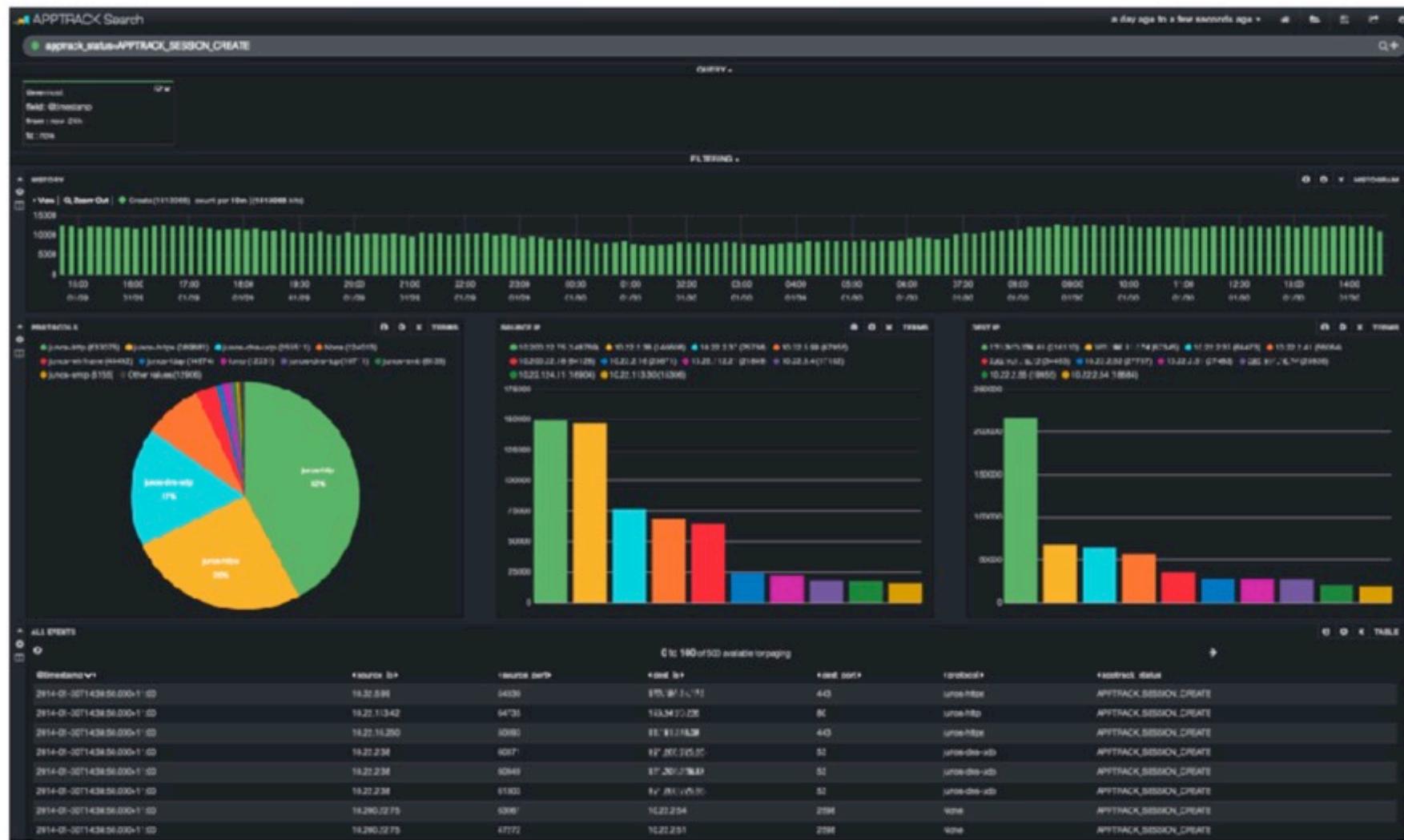
Get Started »

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

Star 1,198 Fork 287

**Every event** under one roof

# KIBANA DASHBOARD





splunk®>



<https://humio.com>

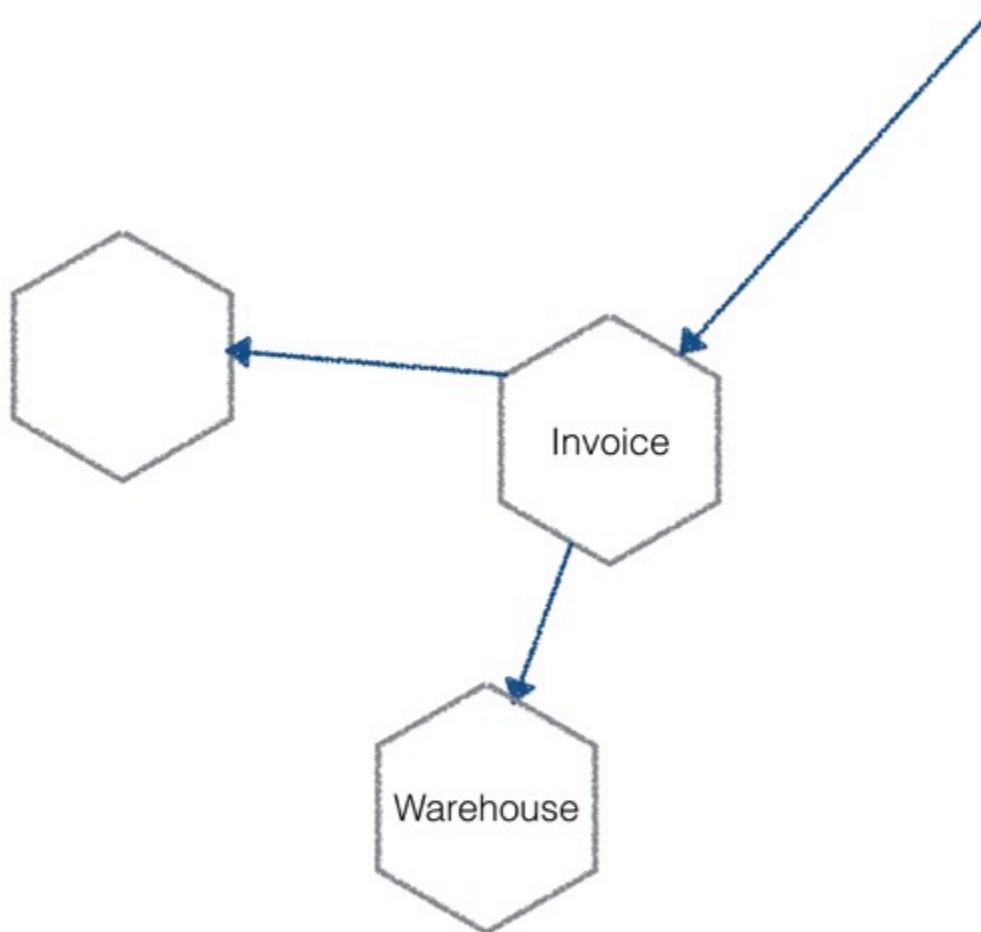
# Consistent Format

# Are your logs important?

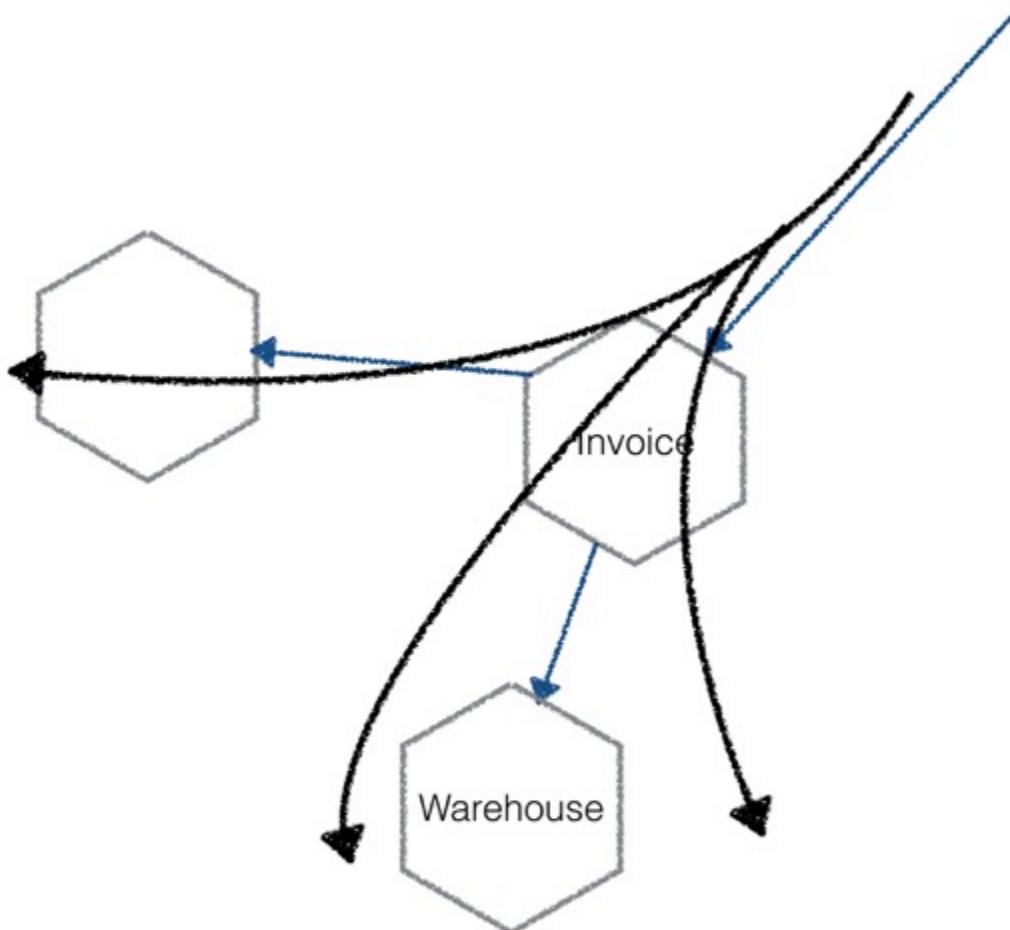
The first thing you should do:

## **Correlation IDs**

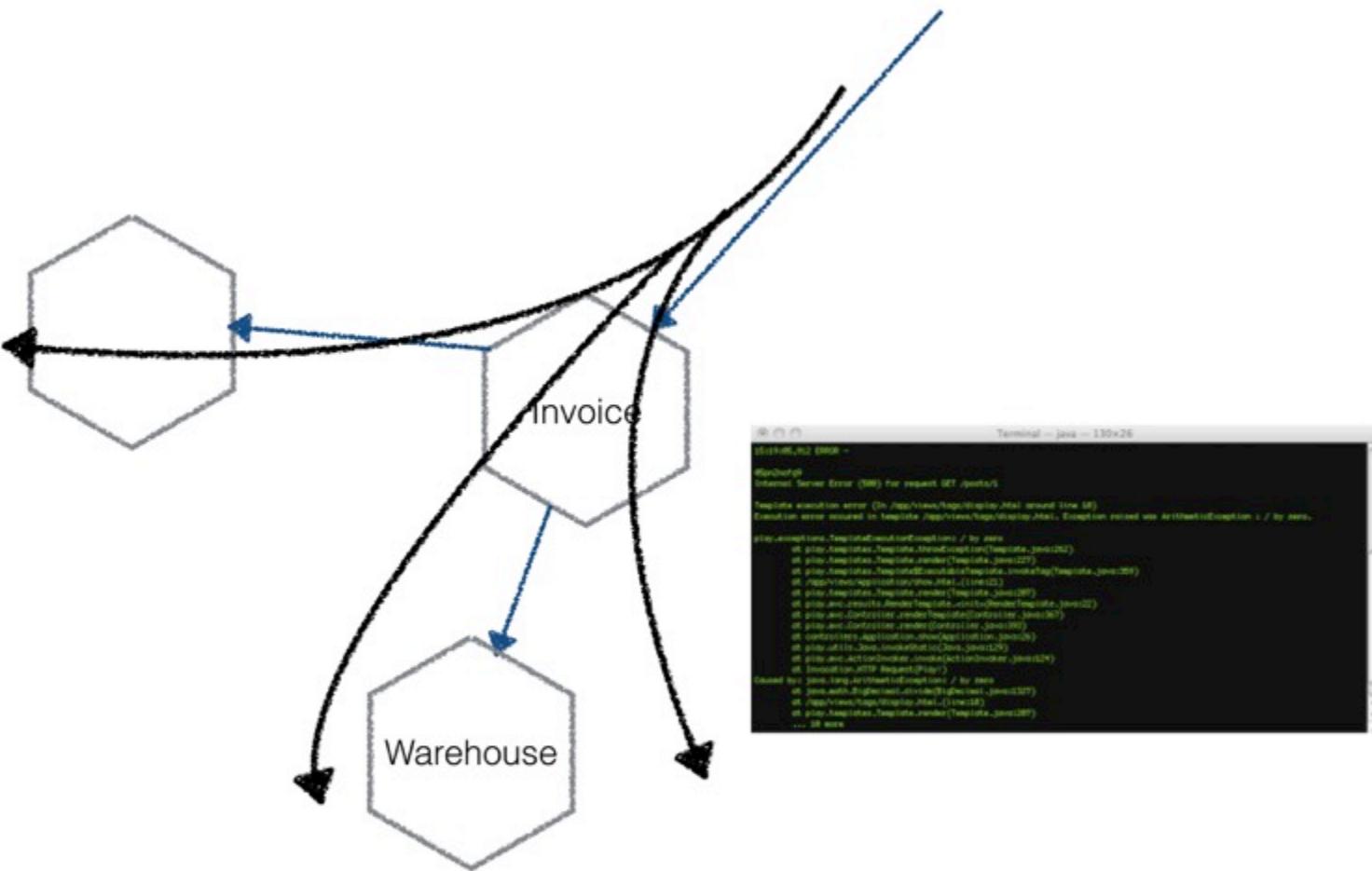
## CORRELATION IDS



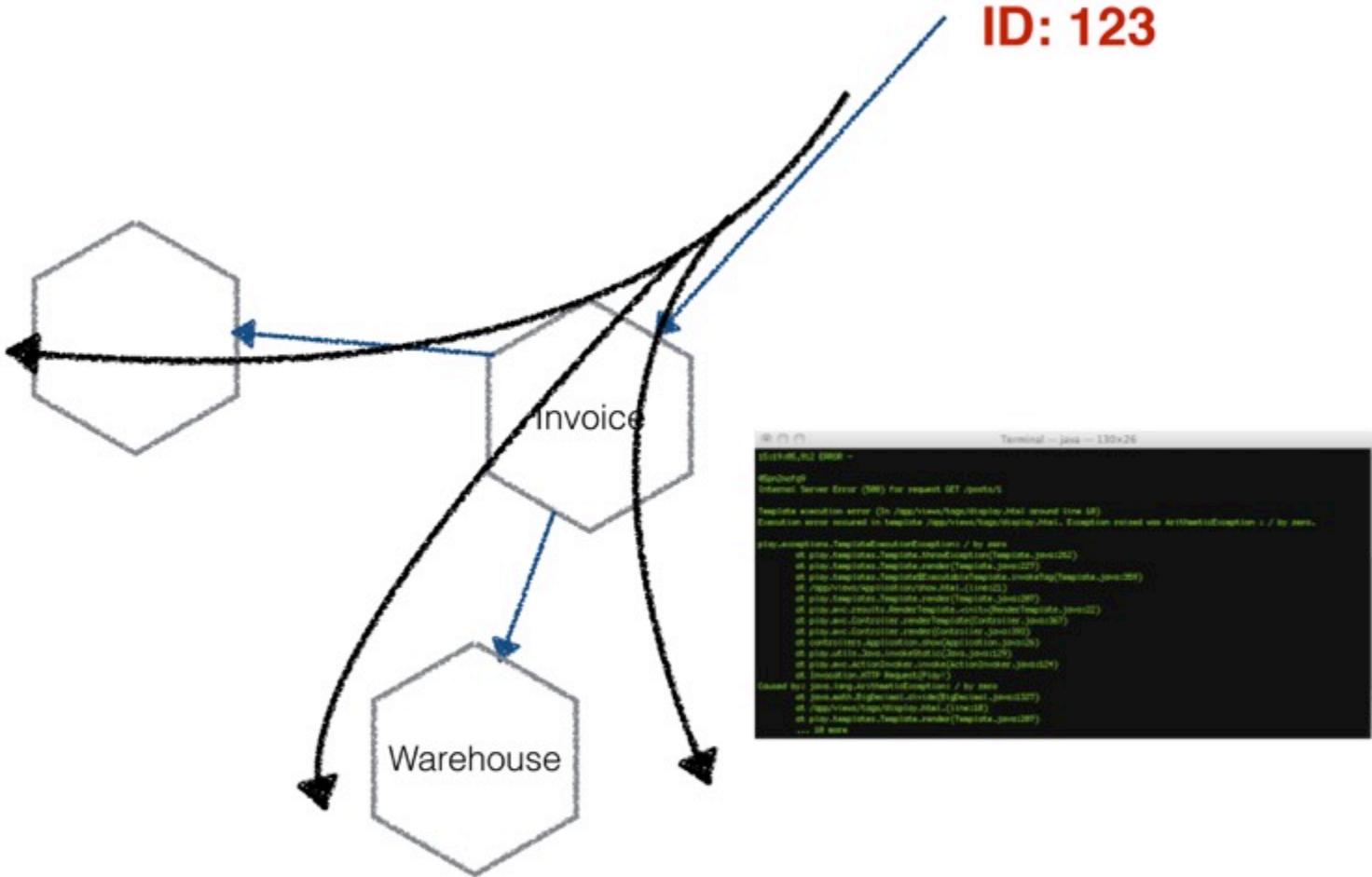
## CORRELATION IDS



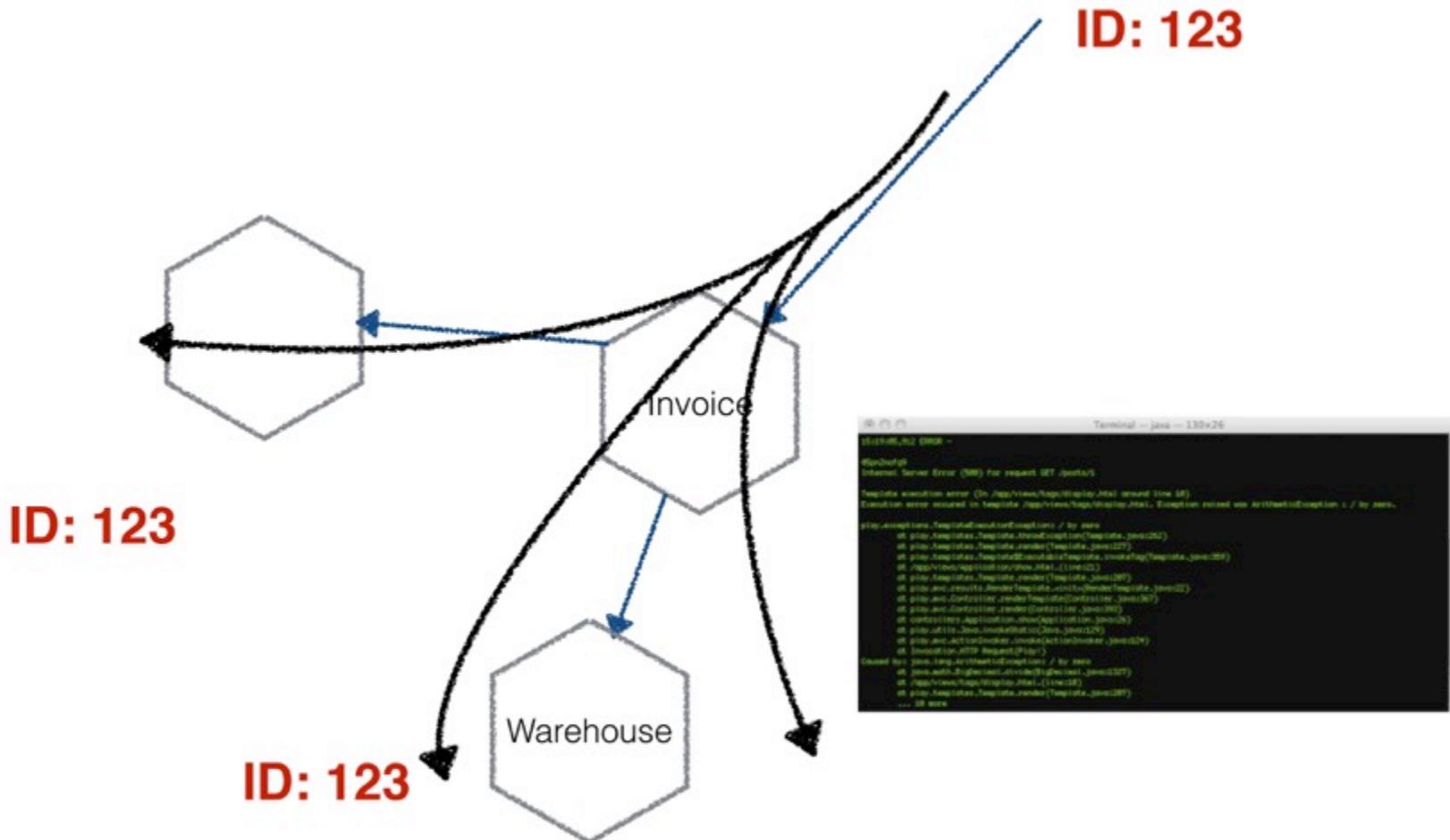
## CORRELATION IDS



## CORRELATION IDS



## CORRELATION IDS



12:00 23/07/2017 [Invoice] WARN [123-gda] Unknown country code

12:01 23/07/2017 [Warehouse] ERROR [] Invalid country code

## Correlation ID



12:00 23/07/2017 [Invoice] WARN [123-gda] Unknown country code

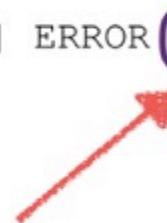
12:01 23/07/2017 [Warehouse] ERROR [] Invalid country code

Correlation ID

12:00 23/07/2017 [Invoice] WARN [123-gda] Unknown country code

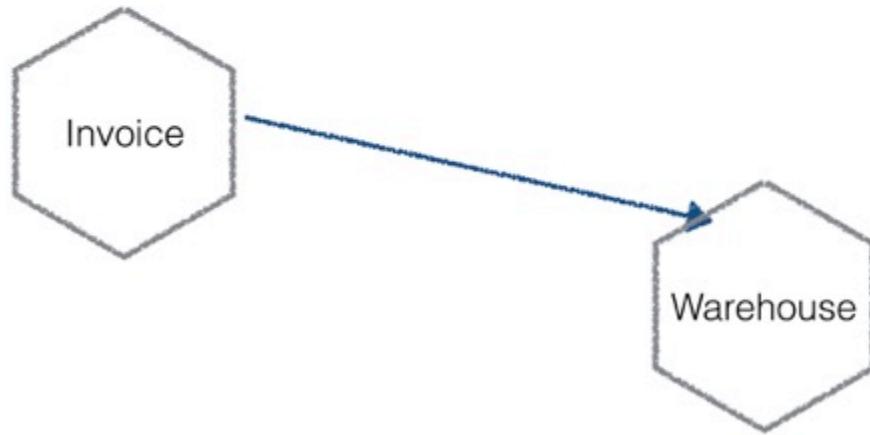
12:01 23/07/2017 [Warehouse] ERROR [] Invalid country code

No correlation ID!



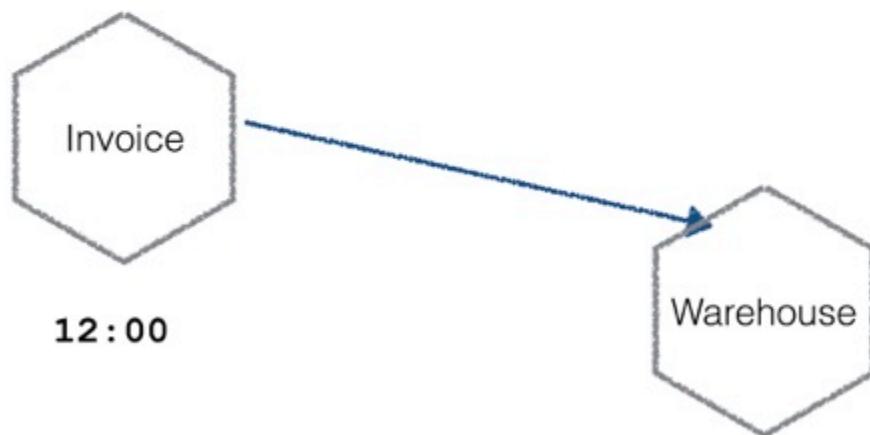
12:00 23/07/2017 [Invoice] WARN [123-gda] Unknown country code

12:01 23/07/2017 [Warehouse] ERROR [123-gda] Invalid country code



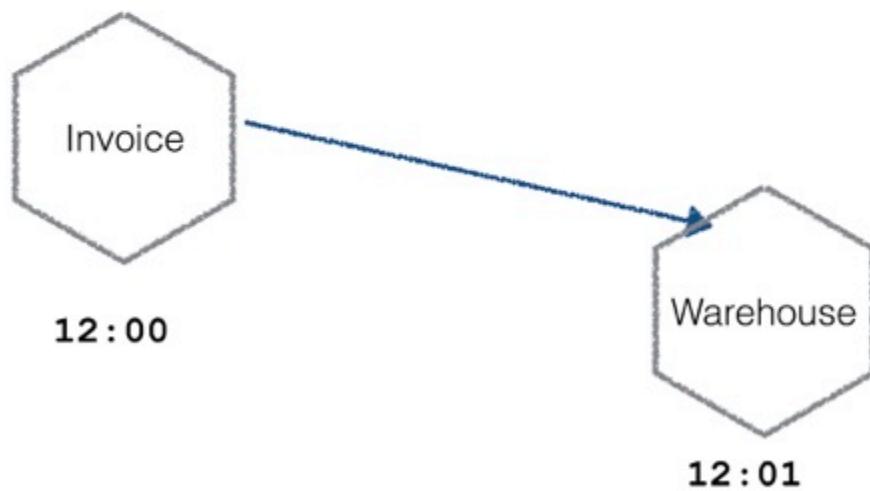
12:00 23/07/2017 [Invoice] WARN [123-gda] Unknown country code

12:01 23/07/2017 [Warehouse] ERROR [123-gda] Invalid country code



12:00 23/07/2017 [Invoice] WARN [123-gda] Unknown country code

12:01 23/07/2017 [Warehouse] ERROR [123-gda] Invalid country code



Time varies between machines!

You can't infer causality from time  
in logs

Operating  
Systems

R. Stockton Gaines  
Editor

# Time, Clocks, and the Ordering of Events in a Distributed System

Leslie Lamport  
Massachusetts Computer Associates, Inc.

---

The concept of one event happening before another in a distributed system is examined, and is shown to define a partial ordering of the events. A distributed algorithm is given for synchronizing a system of logical

A distributed system consists of a collection of distinct processes which are spatially separated, and which communicate with one another by exchanging messages. A network of interconnected computers, such as the ARPA net, is a distributed system. A single computer can also be viewed as a distributed system in which the central control unit, the memory units, and the input-output channels are separate processes. A system is distributed if the message transmission delay is not negligible compared to the time between events in a single process.

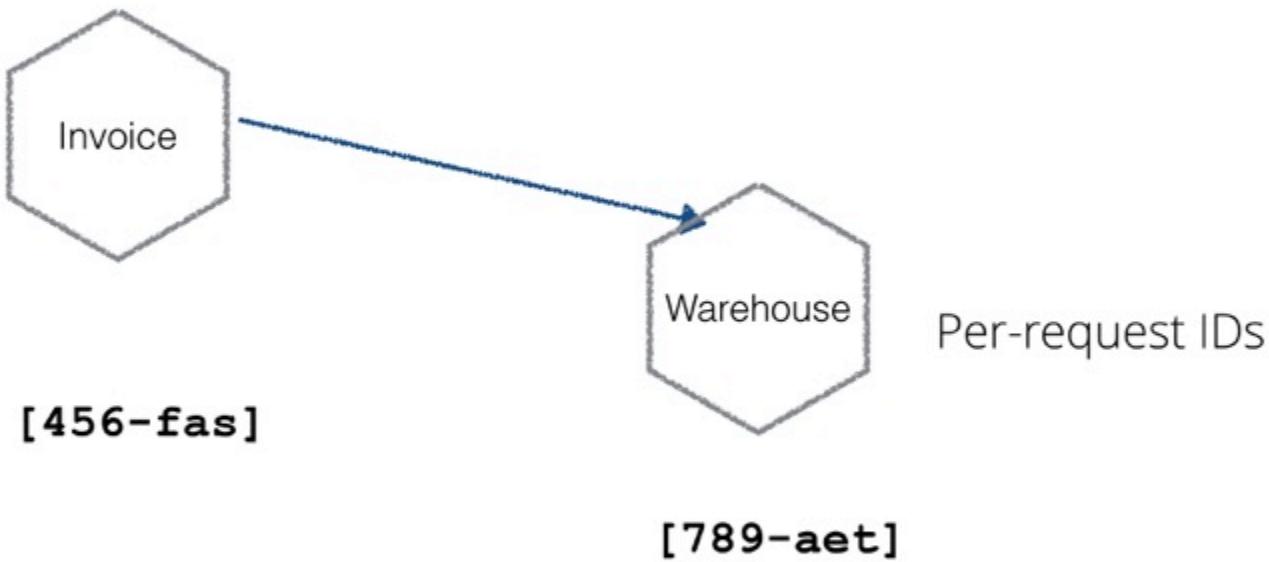
We will concern ourselves primarily with systems of spatially separated computers. However, many of our remarks will apply more generally. In particular, a multiprocesssing system on a single computer involves problems similar to those of a distributed system because of the unpredictable order in which certain events can occur.

In a distributed system, it is sometimes impossible to say that one of two events occurred first. The relation "happened before" is therefore only a partial ordering of the events in the system. We have found that problems often arise because people are not fully aware of the

<http://amturing.acm.org/p558-lamport.pdf>

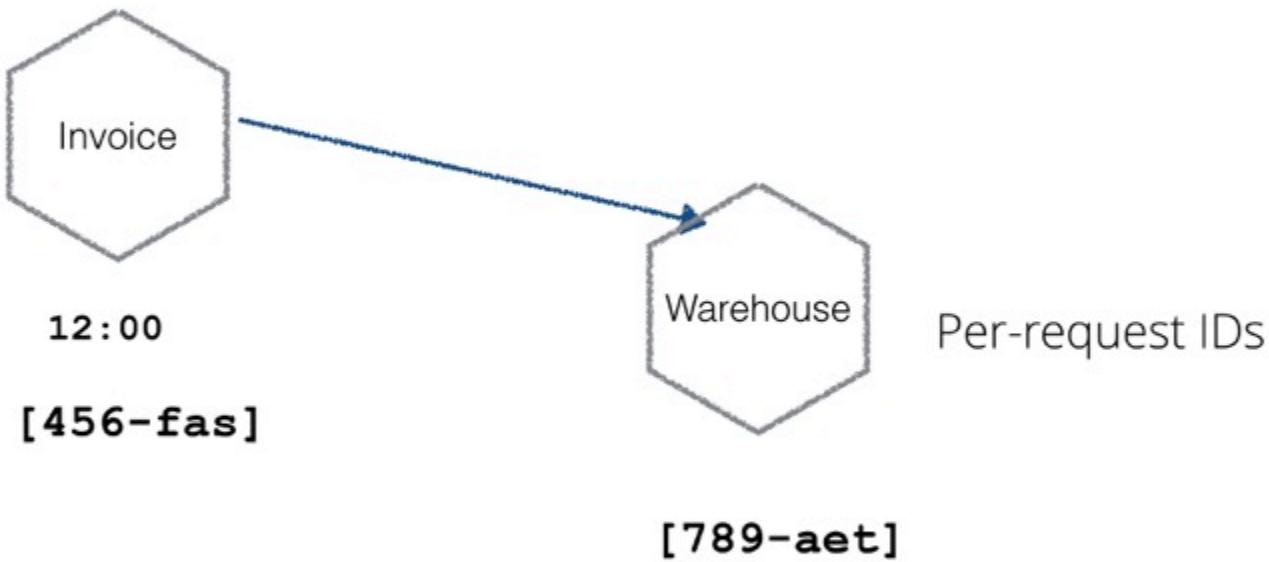
'Global' Correlation ID for whole operation

[123-gda]



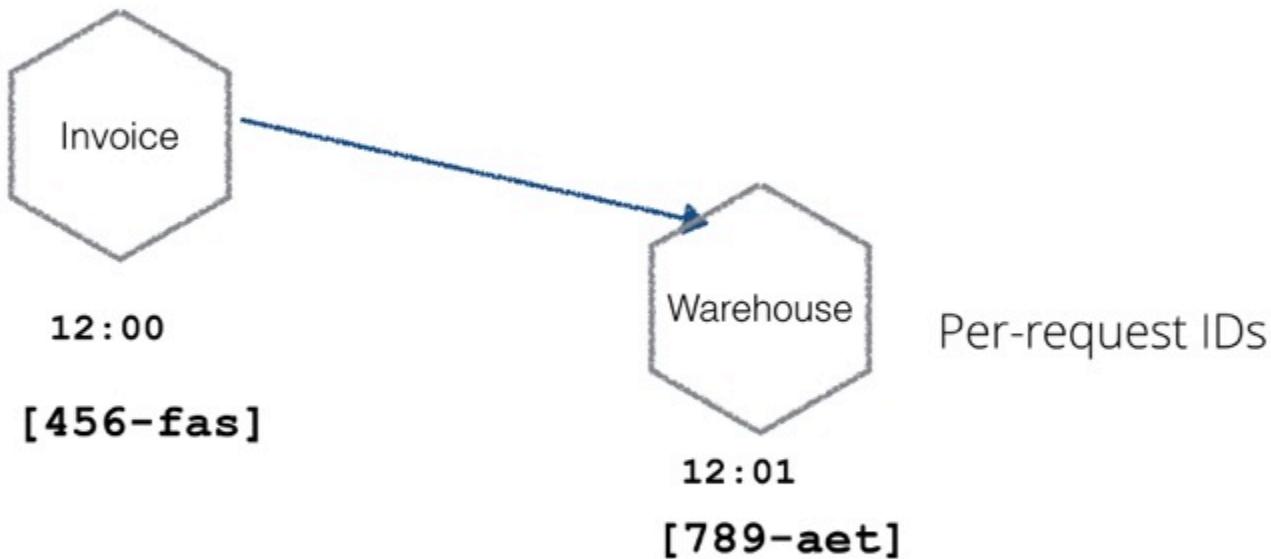
'Global' Correlation ID for whole operation

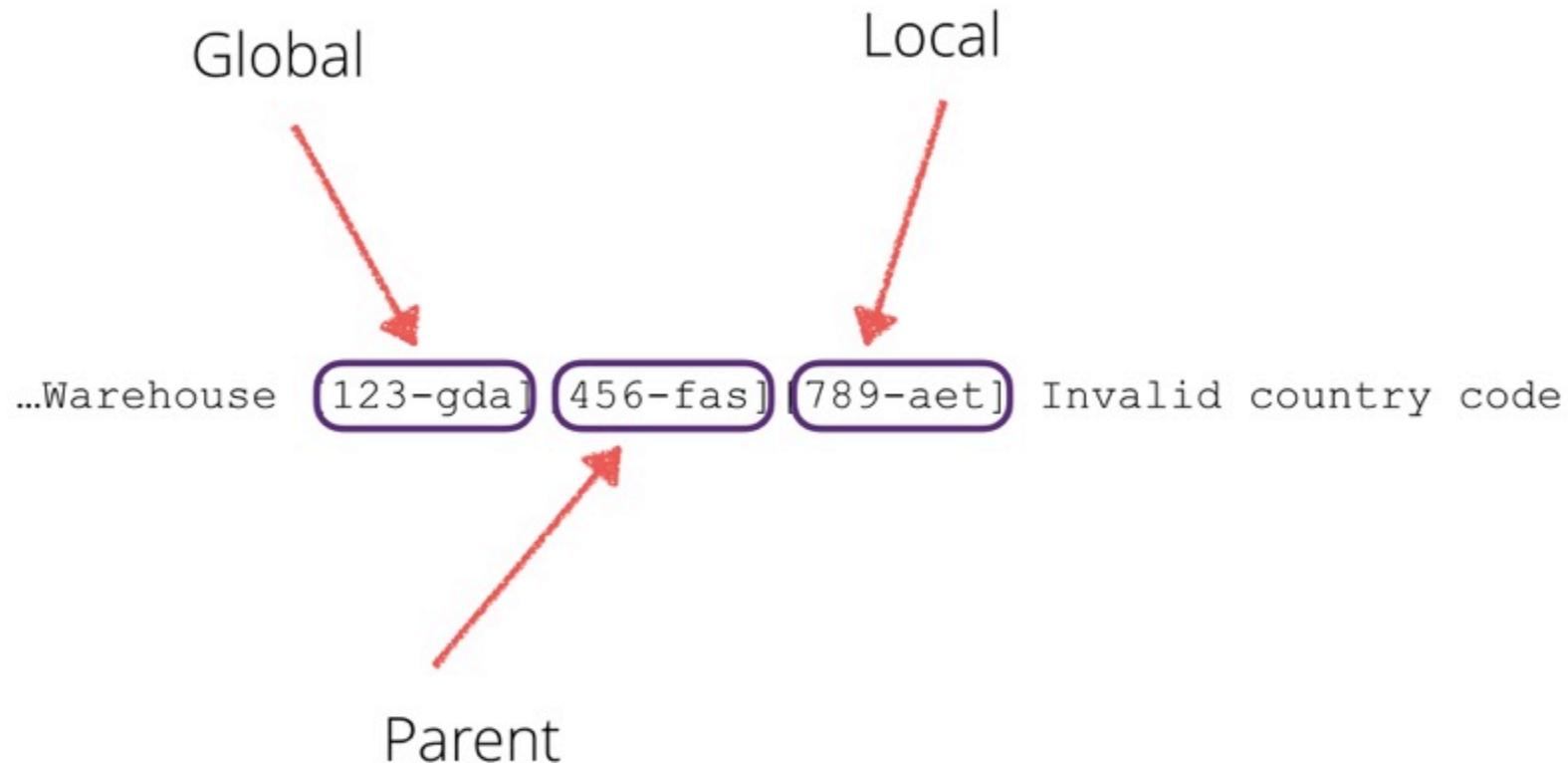
[123-gda]



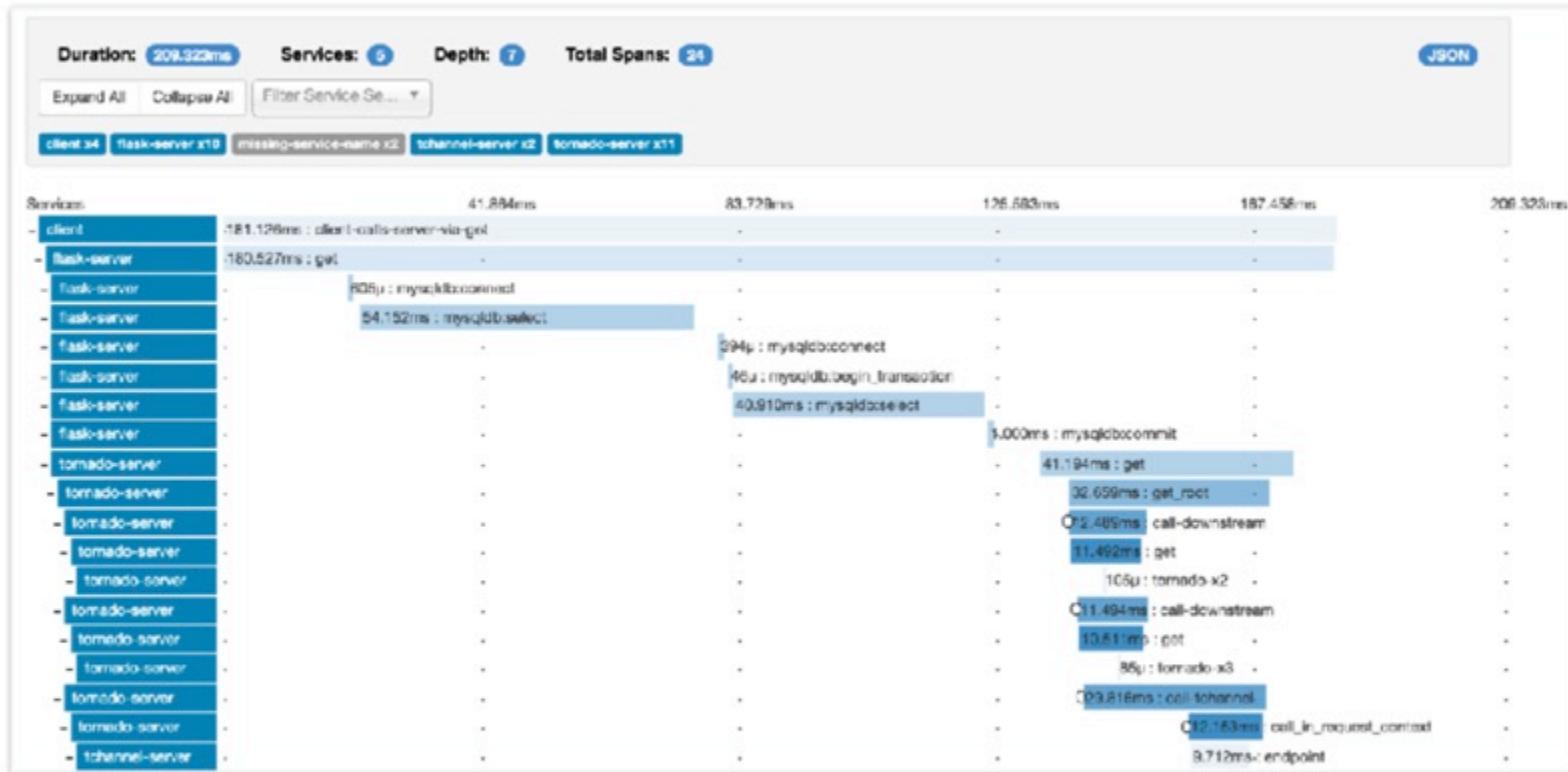
'Global' Correlation ID for whole operation

[123-gda]



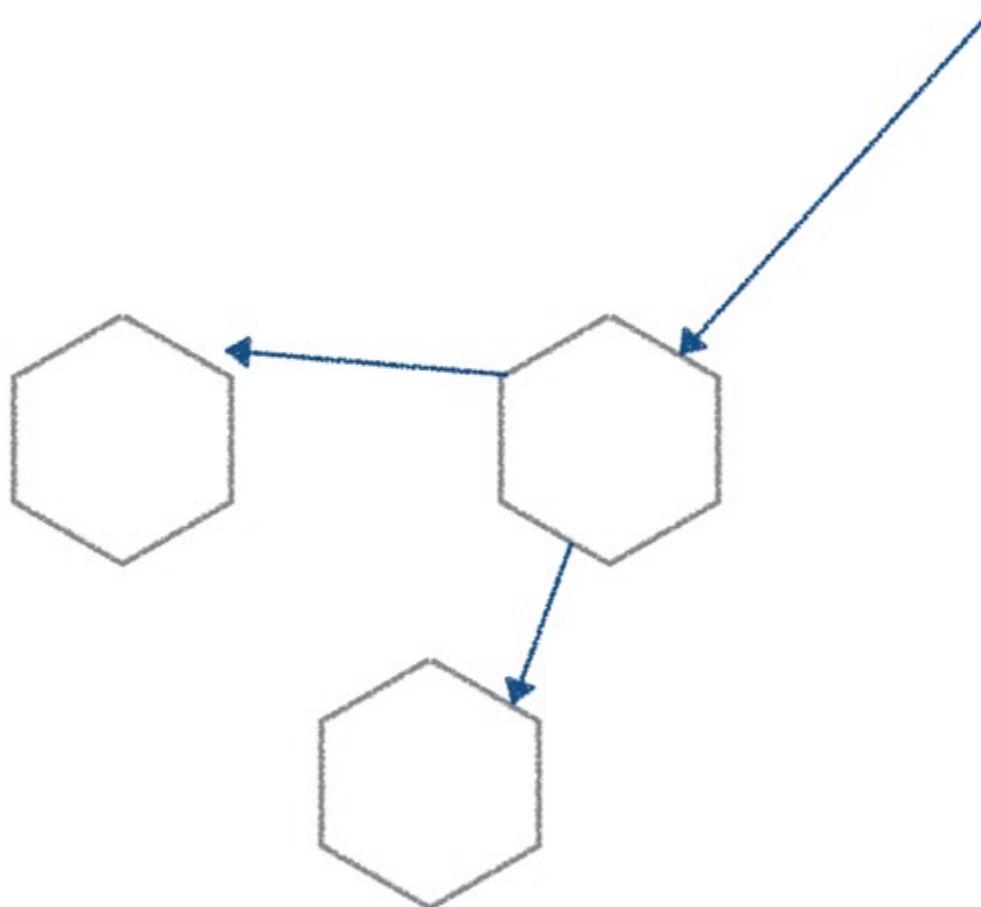


Other mechanisms available

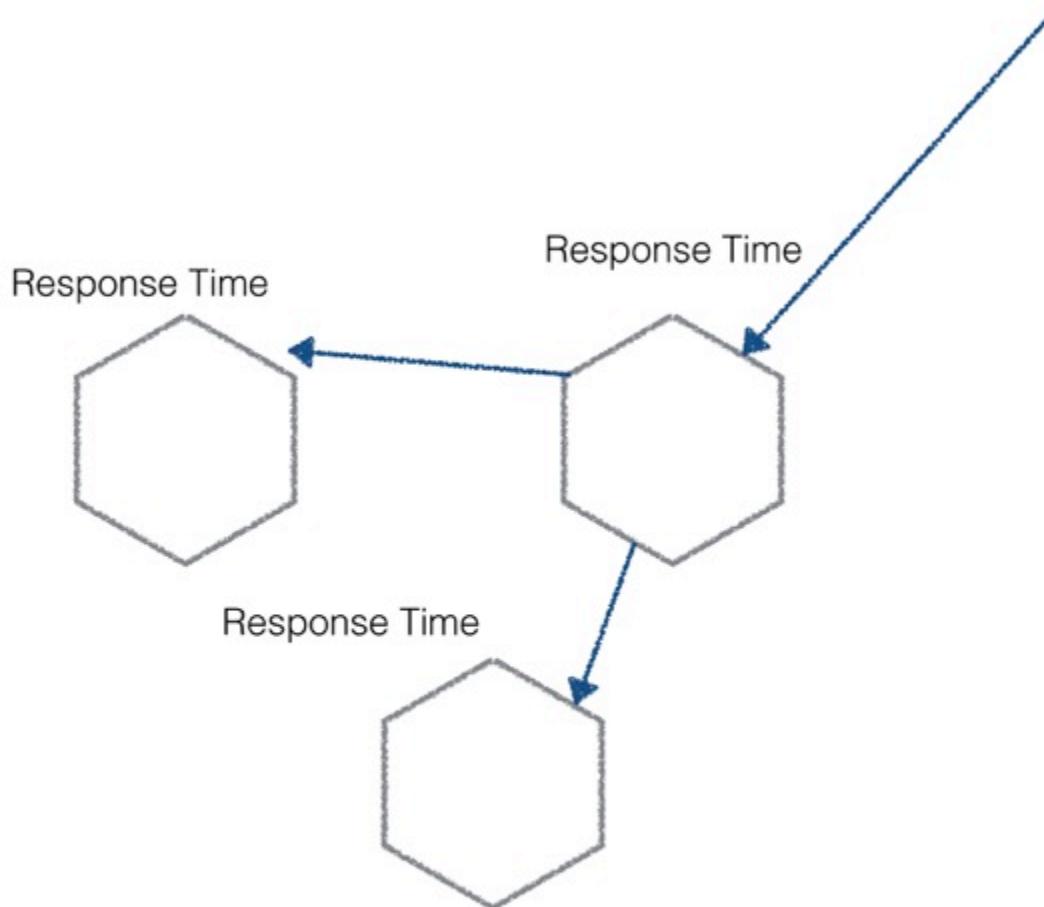


<http://zipkin.io>

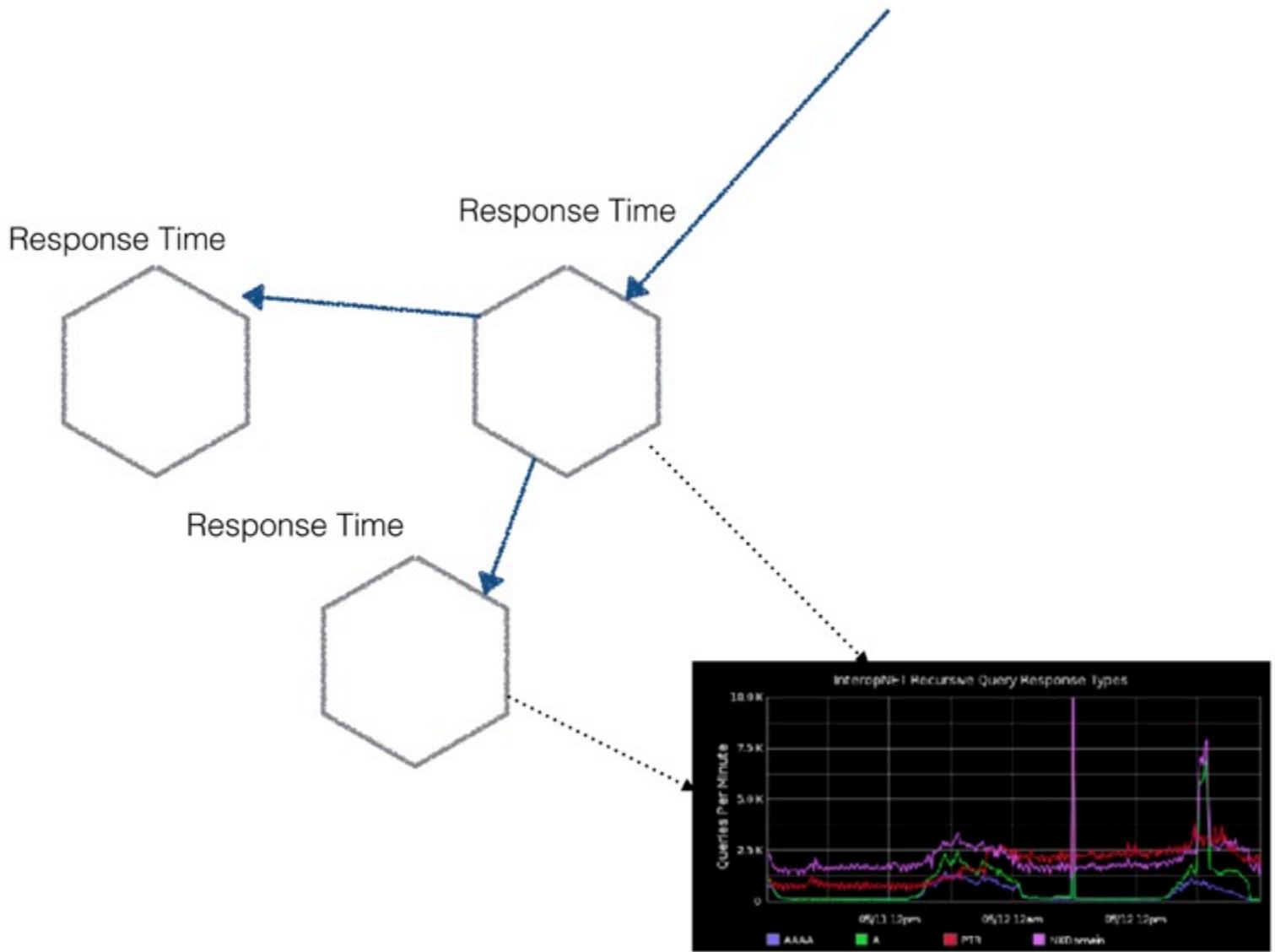
## AGGREGATE METRICS



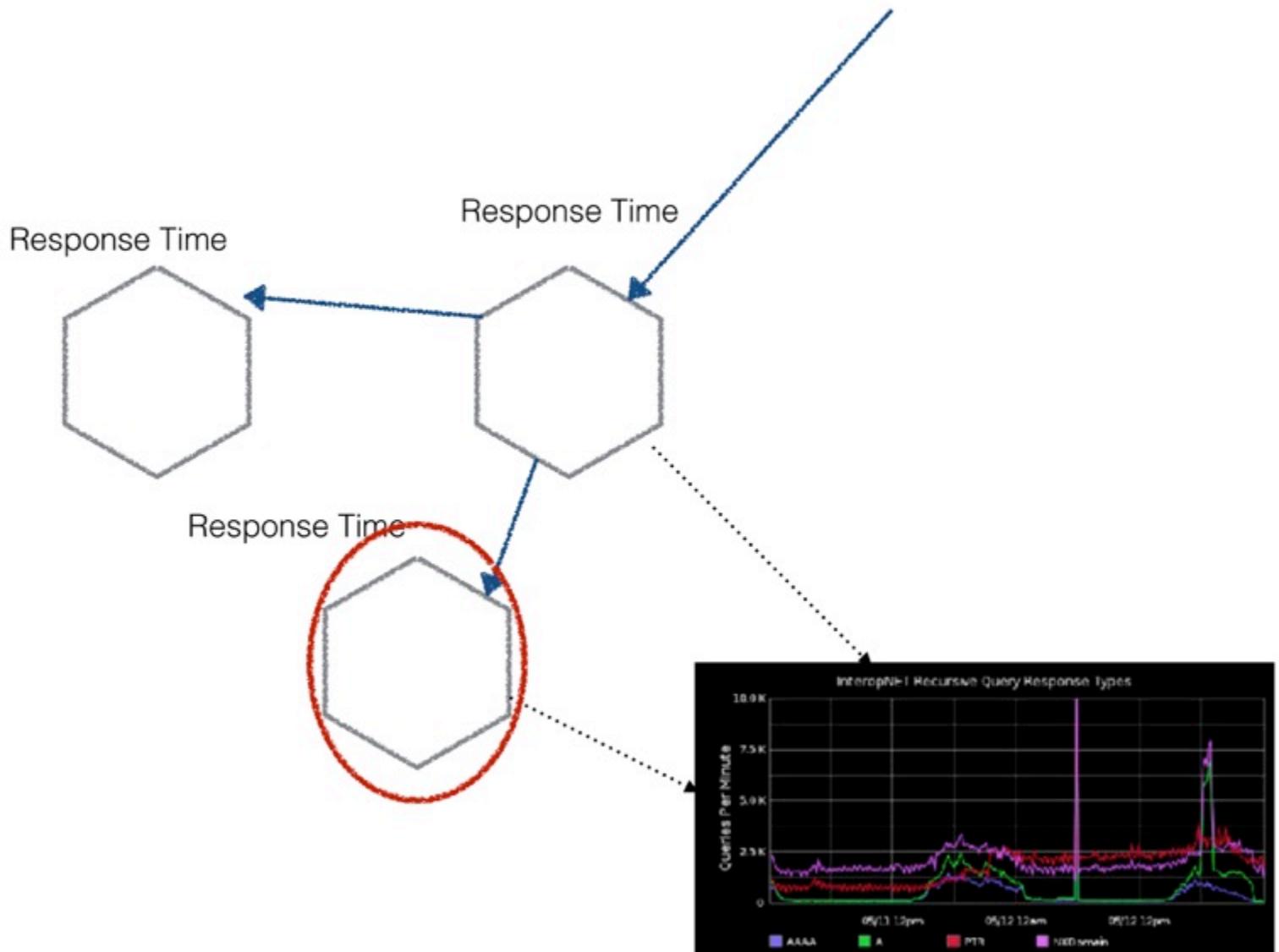
## AGGREGATE METRICS



## AGGREGATE METRICS



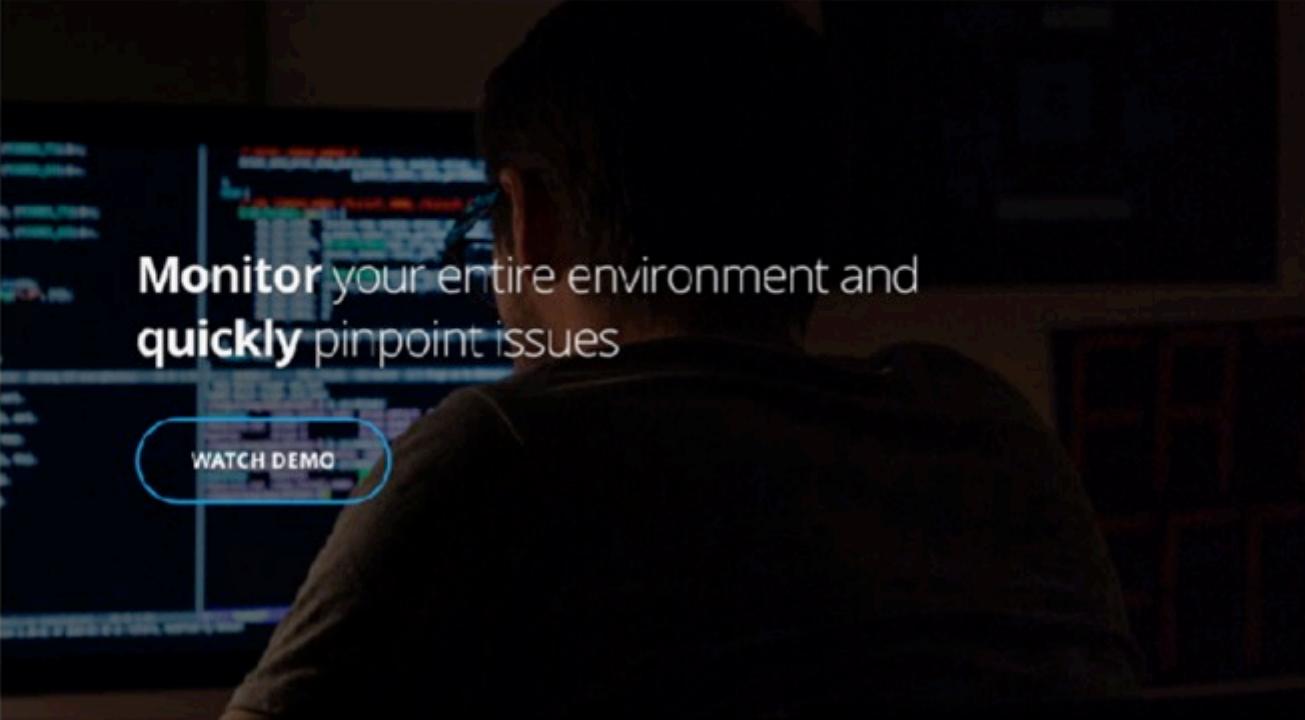
## AGGREGATE METRICS



# Prometheus



<https://prometheus.io>

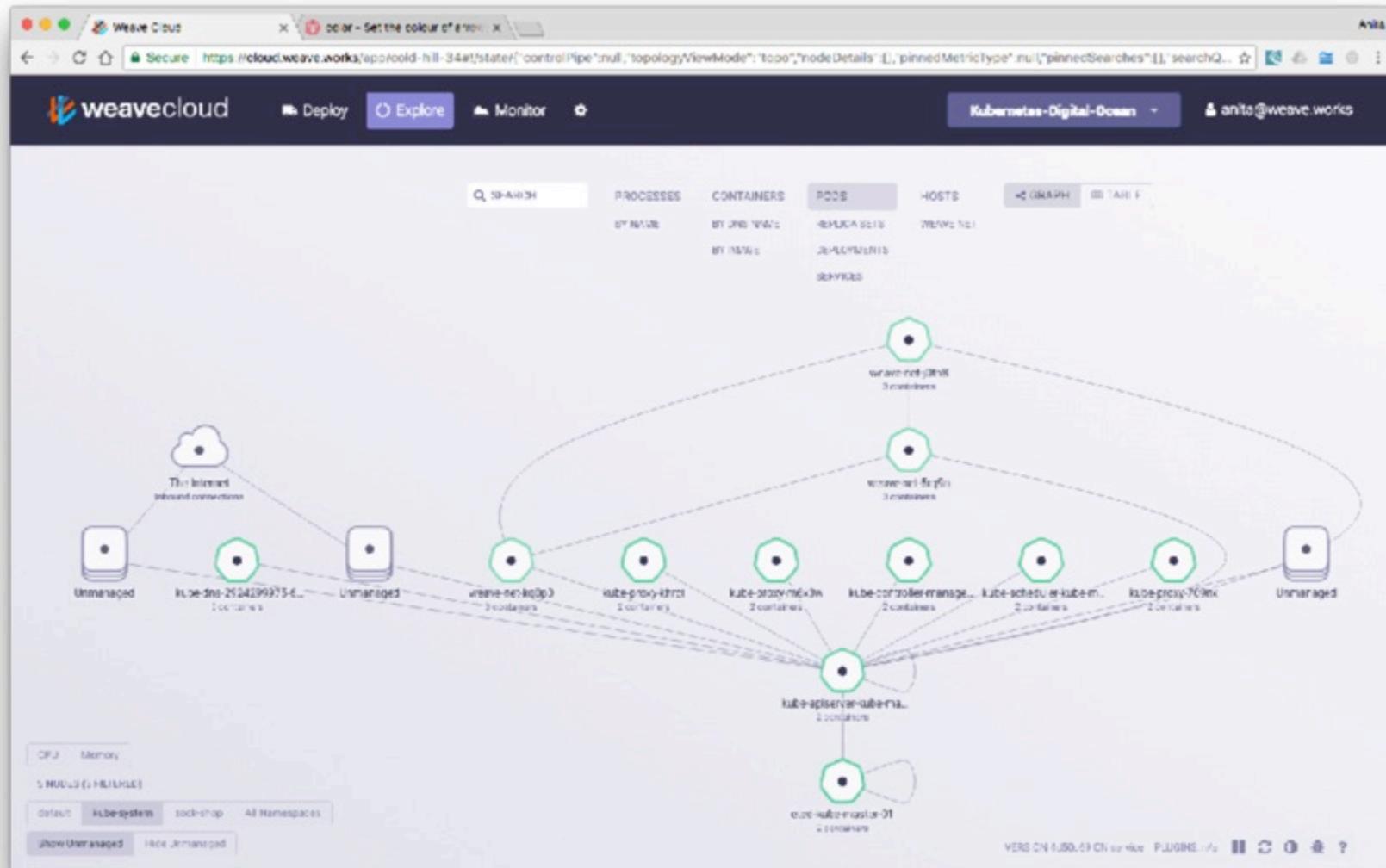


Monitor your entire environment and  
quickly pinpoint issues

WATCH DEMO

No matter how **dynamic** your microservices environment, Weave Cloud's hosted, horizontally **scalable** Prometheus service lets you **quickly** identify issues with your app.

<https://www.weave.works/solution/prometheus-monitoring/>





## **SEMANTIC MONITORING**

Don't look for presence of errors

## **SEMANTIC MONITORING**

Don't look for presence of errors

Define a model for a correctly  
operating system

## **SEMANTIC MONITORING**

Don't look for presence of errors

Define a model for a correctly operating system

Alert if your system doesn't behave correctly

## **SEMANTIC MONITORING - EXAMPLE STATEMENTS**

People can order CDs

## **SEMANTIC MONITORING - EXAMPLE STATEMENTS**

People can order CDs

We are making at least \$30K per hour

## **SEMANTIC MONITORING - EXAMPLE STATEMENTS**

People can order CDs

We are making at least \$30K per hour

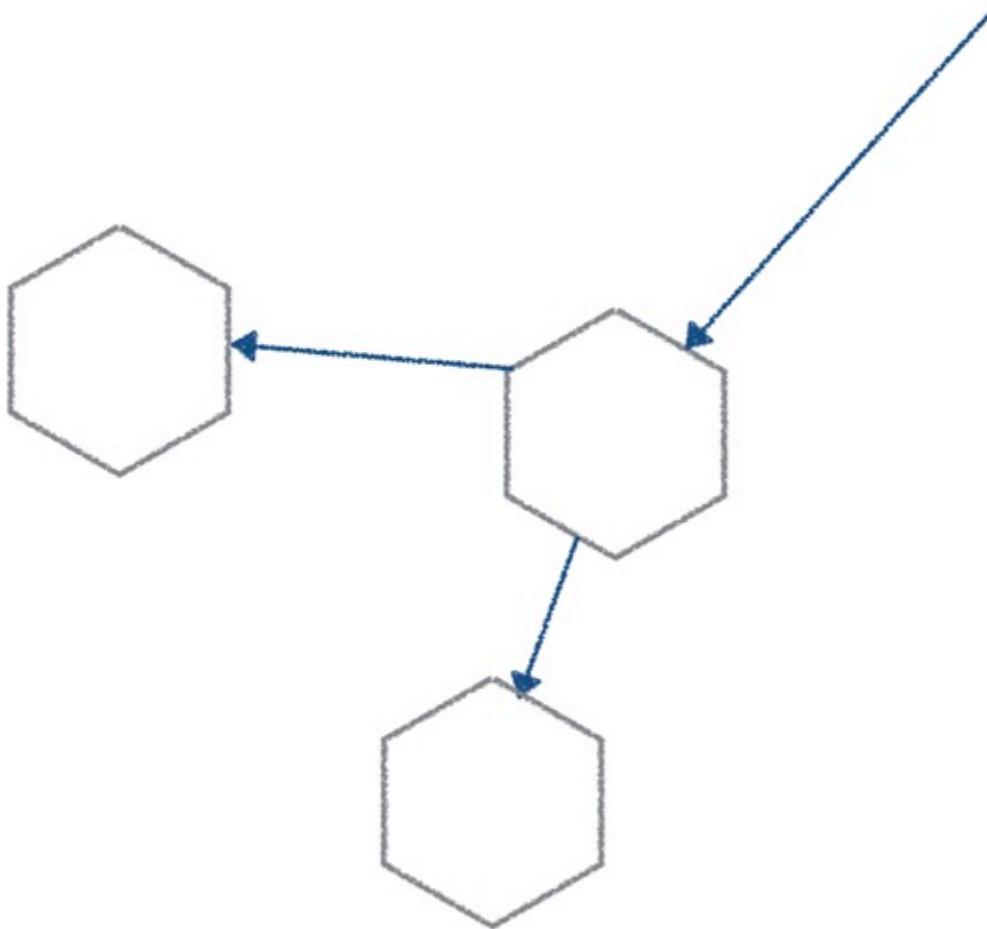
New users can still register

## REAL USER MONITORING

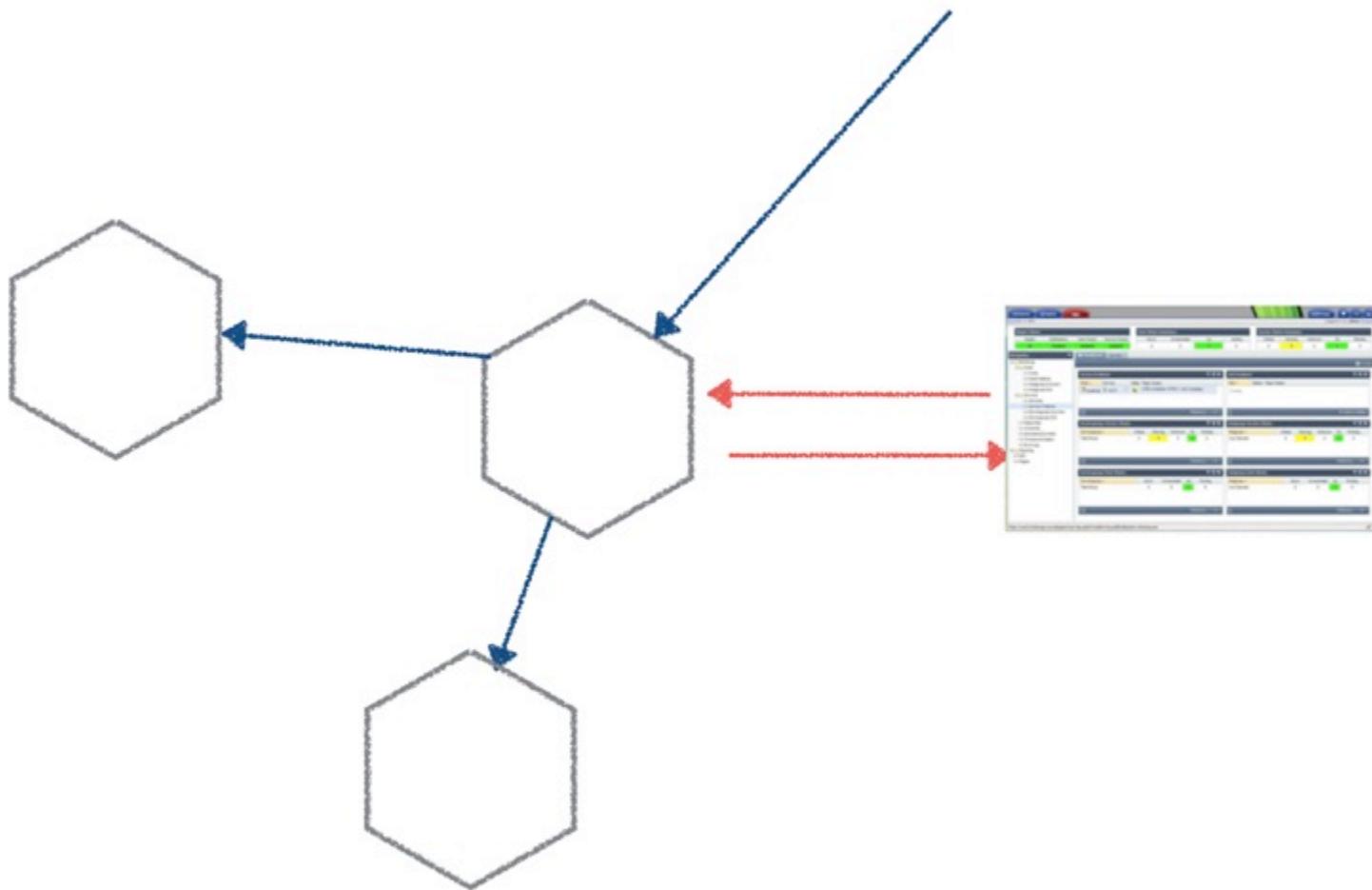


Use metrics to validate the model

## SYNTHETIC TRANSACTIONS



## SYNTHETIC TRANSACTIONS



## SYNTHETIC TRANSACTIONS

