

# Insecure Transit

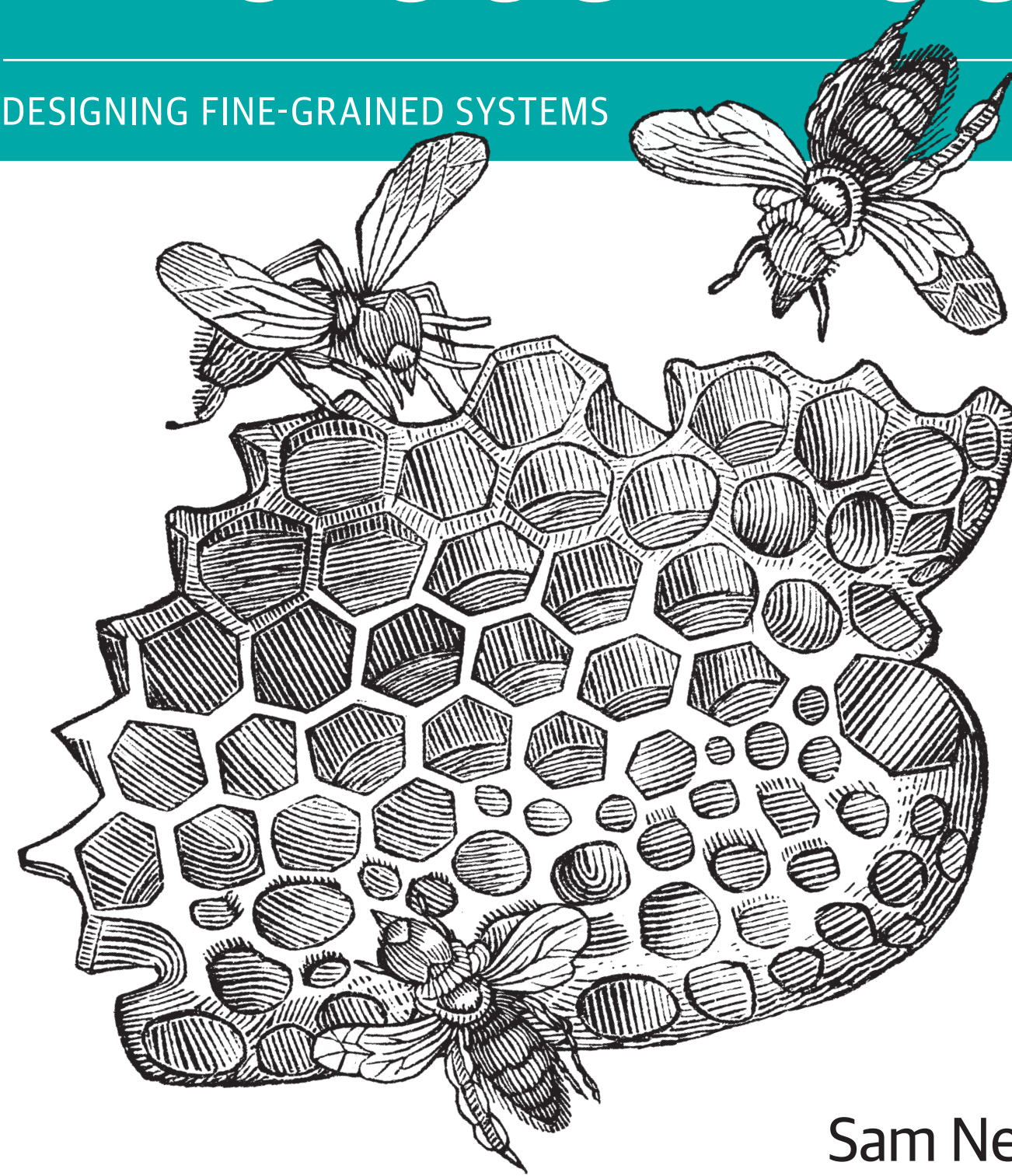
Microservice Security

Sam Newman

O'REILLY®

# Building Microservices

DESIGNING FINE-GRAINED SYSTEMS



Sam Newman

**Sam  
Newman**  
& Associates





# Massive Equifax data breach - what you need to know



By [Callum Mason](#), News Reporter  
12 Sep 2017 | Updated 19 Sep 2017



Credit report heavyweight Equifax has warned that up to 400,000 UK consumers may have had their personal details stolen as part of a massive global data breach. Info on exactly who's been affected and what you can do about it is still somewhat sketchy, but here's what we know.

Equifax revealed on 8 September that 143 million consumers in the US could have been affected by the incident, which saw hackers access data such as names, address and dates of birth, as well as credit card numbers in a smaller number of cases.

Although its UK business – Equifax Ltd – now says systems in this country are not affected, it admits a file which was stored in the US and contained more limited personal information on up to 400,000 UK consumers may have been accessed.

## Related MSE Guides

### [Credit Scores](#)

Bust myths & improve your score

### [30+ Ways to Stop Scams](#)

As scams get clever, we need to too!

### [Check your credit report for free](#)

Grab your file and check your score, or even get PAID to do it



## Get Our Free Money Tips Email!

For all the latest deals, guides and loopholes - join the 12m who get it.

*Don't miss out*

[GET IT!](#)

[FAQs](#) | [Privacy Policy](#) | [Past Emails](#) | [Unsubscribe](#)

## What is Equifax and what data does it have?

Equifax is the second biggest credit reference agency in the UK, after Experian.


<https://www.moneysavingexpert.com/news/protect/2017/09/massive-equifax-data-breach---what-you-need-to-know>

Security

## Meltdown, Spectre: The password theft bugs at the heart of Intel CPUs

AMD, Arm also affected by data-leak design blunders, Chipzilla hit hardest

By Chris Williams, Editor in Chief 4 Jan 2018 at 07:29

252  SHARE ▼



**Summary** The severe design flaw in Intel microprocessors that allows sensitive data, such as passwords and crypto-keys, to be stolen from memory is real – and its details have been revealed.

On Tuesday, we warned that a [blueprint blunder in Intel's CPUs](#) could allow applications, malware, and JavaScript running in web browsers, to obtain information they should not be allowed to access: the contents of the operating system kernel's private memory areas. These zones often contain files cached from disk, a view onto the machine's entire physical memory, and other secrets. This should be invisible to normal programs.

[https://www.theregister.co.uk/2018/01/04/intel\\_amd\\_arm\\_cpu\\_vulnerability/](https://www.theregister.co.uk/2018/01/04/intel_amd_arm_cpu_vulnerability/)

# Design



**Design**

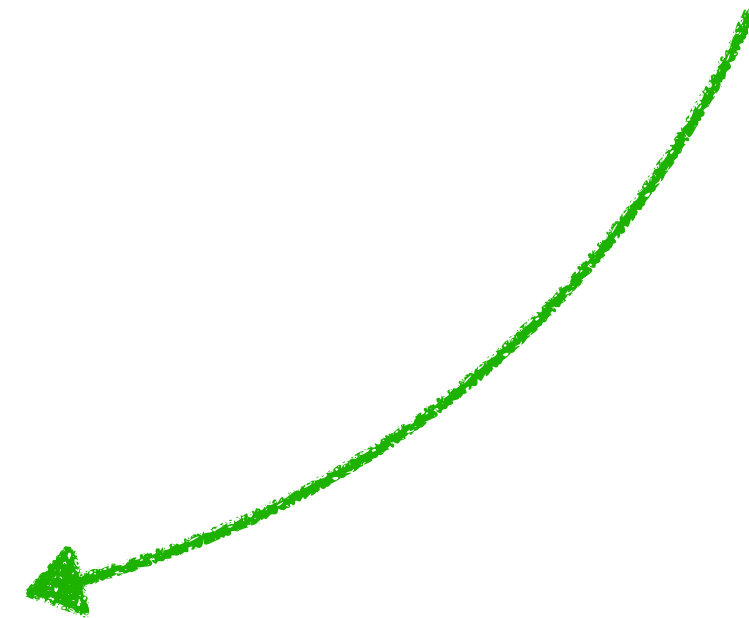
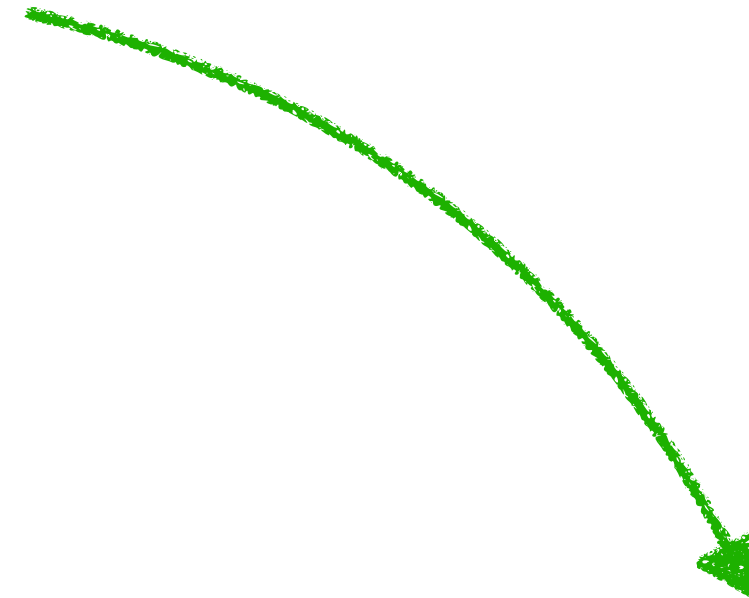


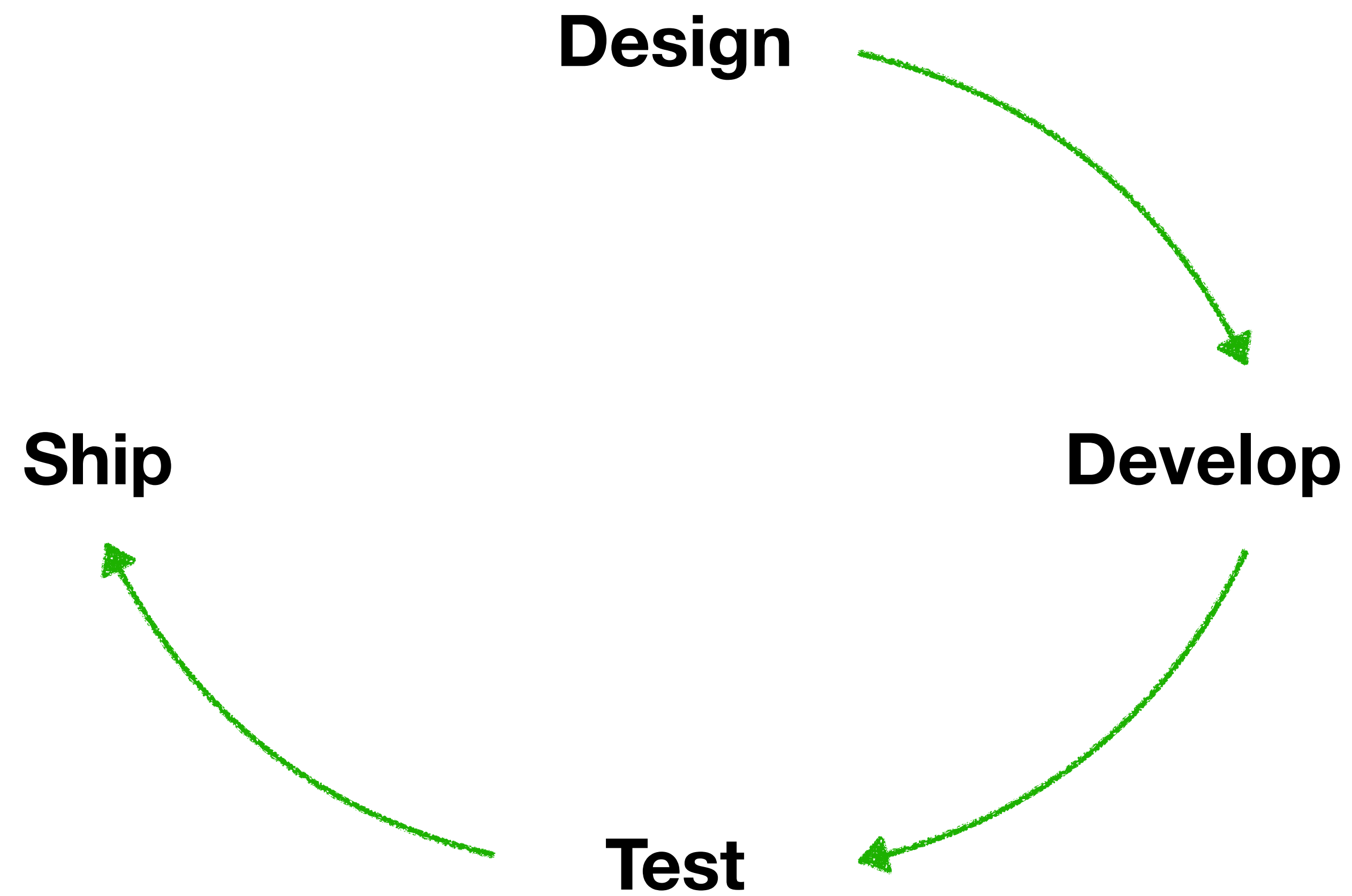
**Develop**

**Design**

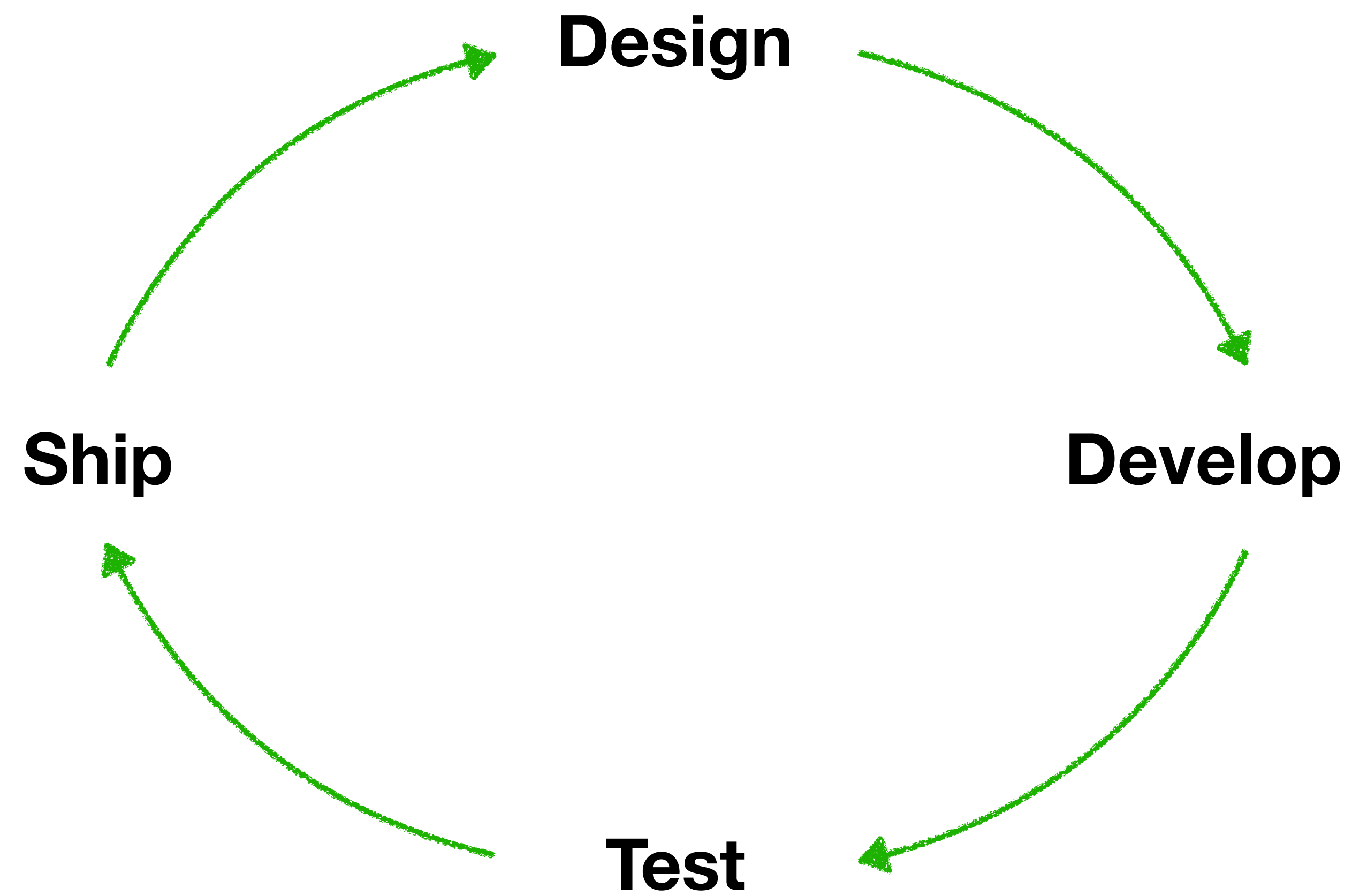
**Develop**

**Test**

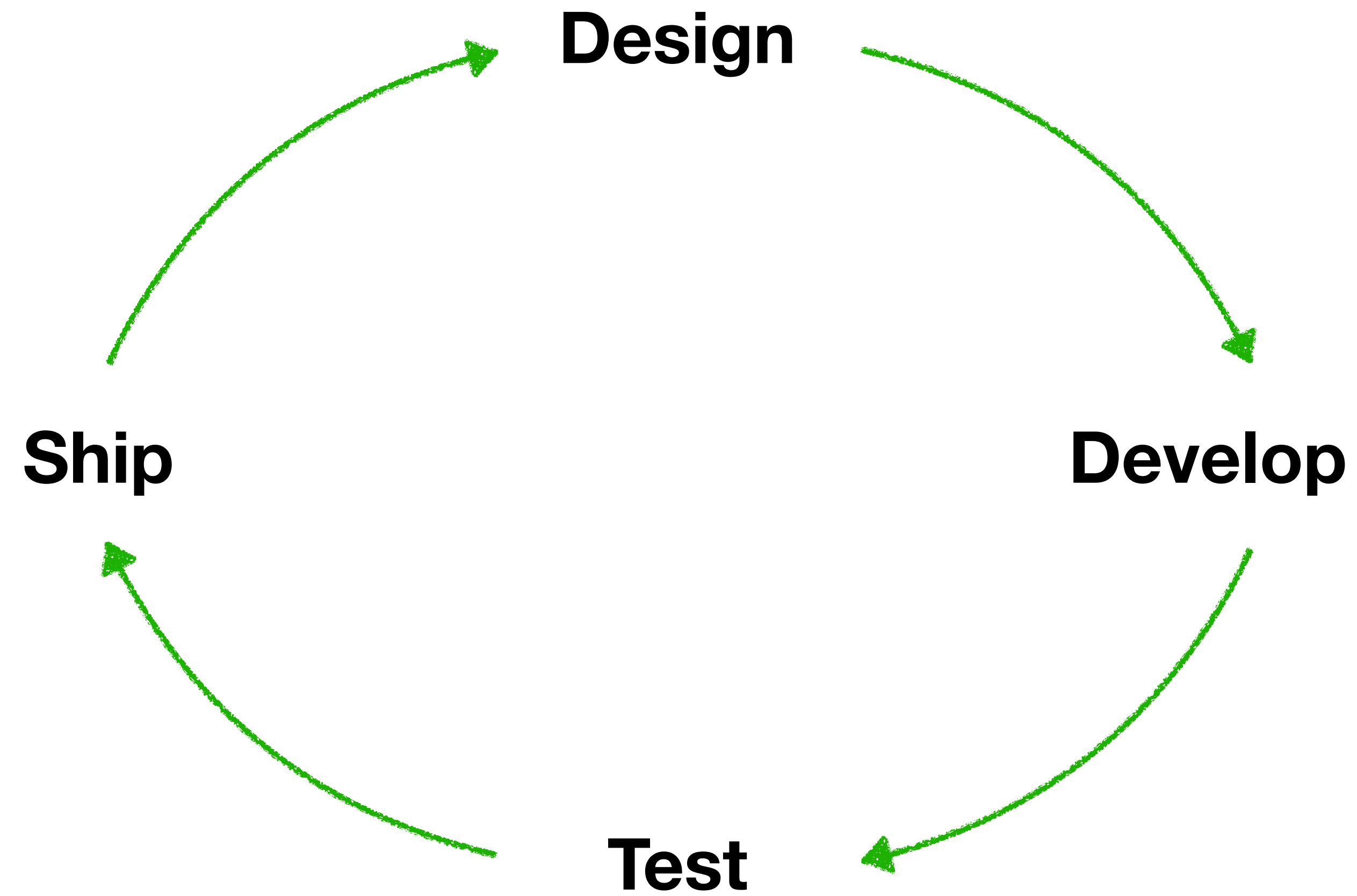








*Err .... security?*

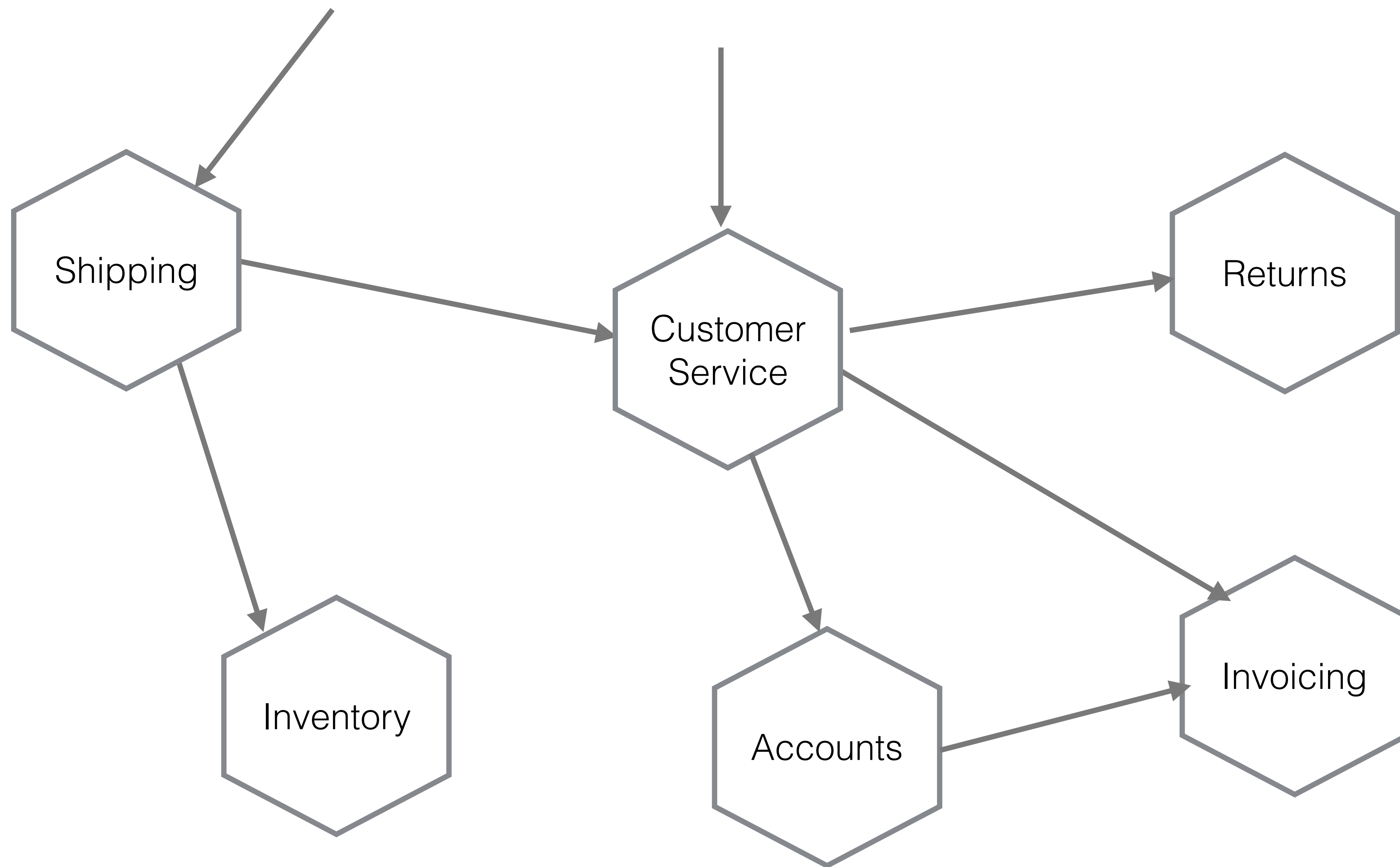


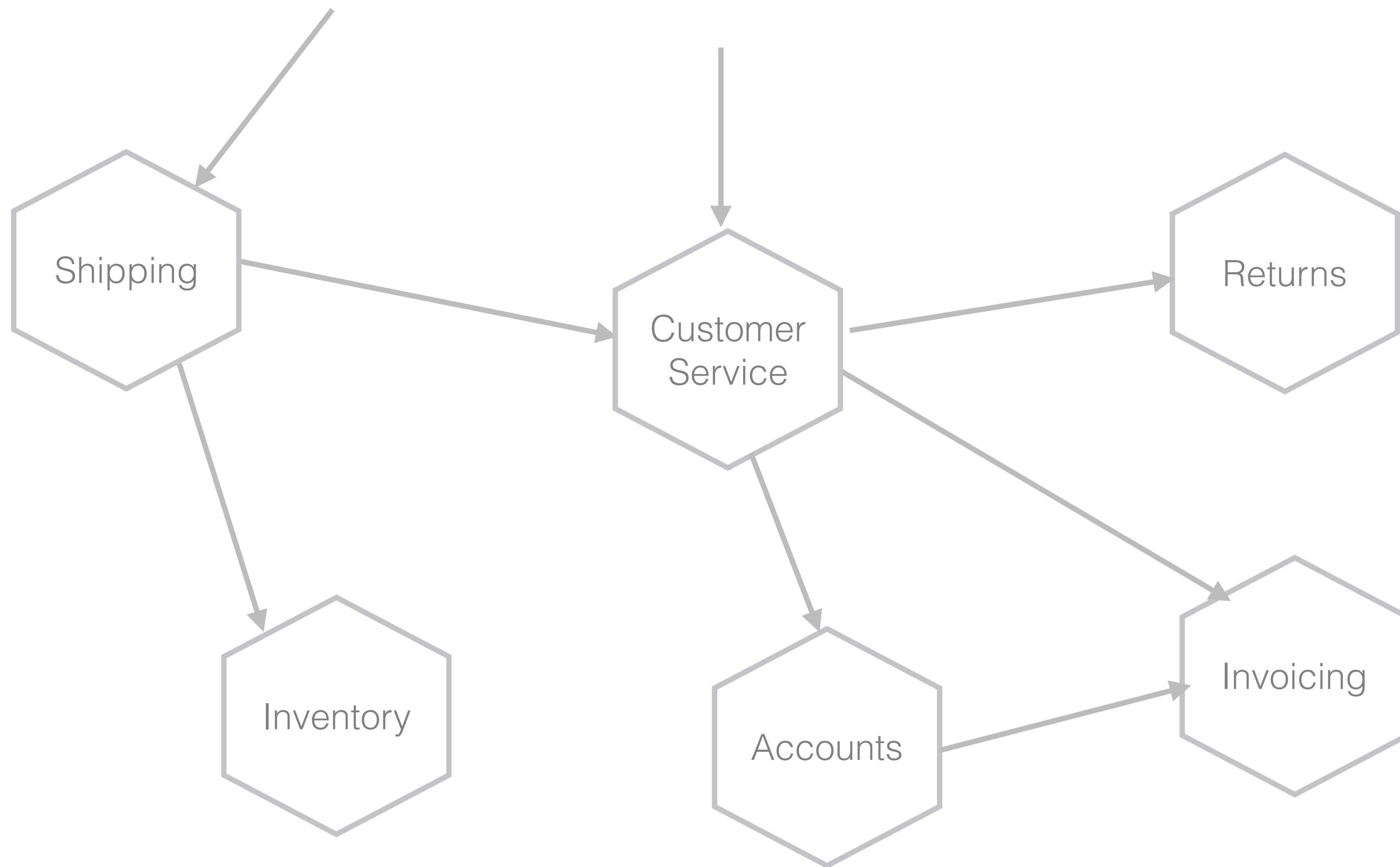






# Just Enough Security

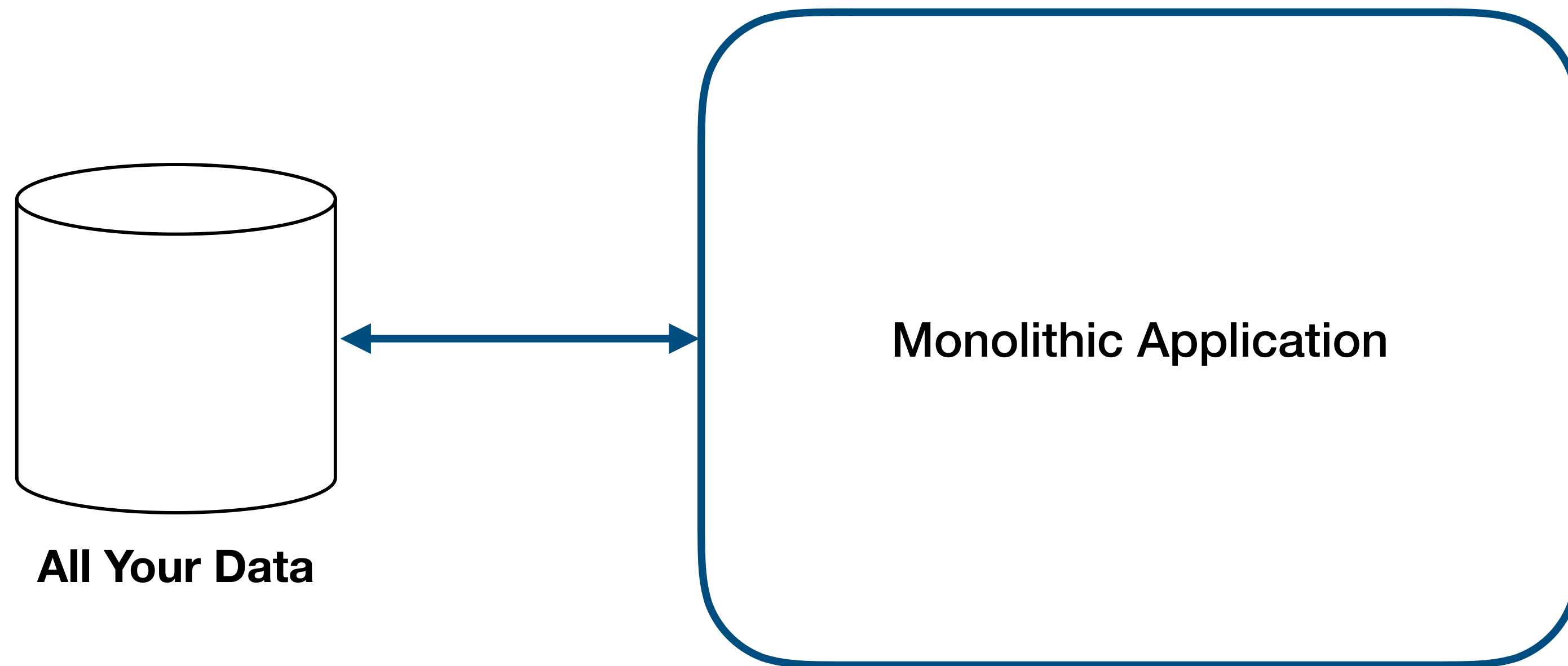








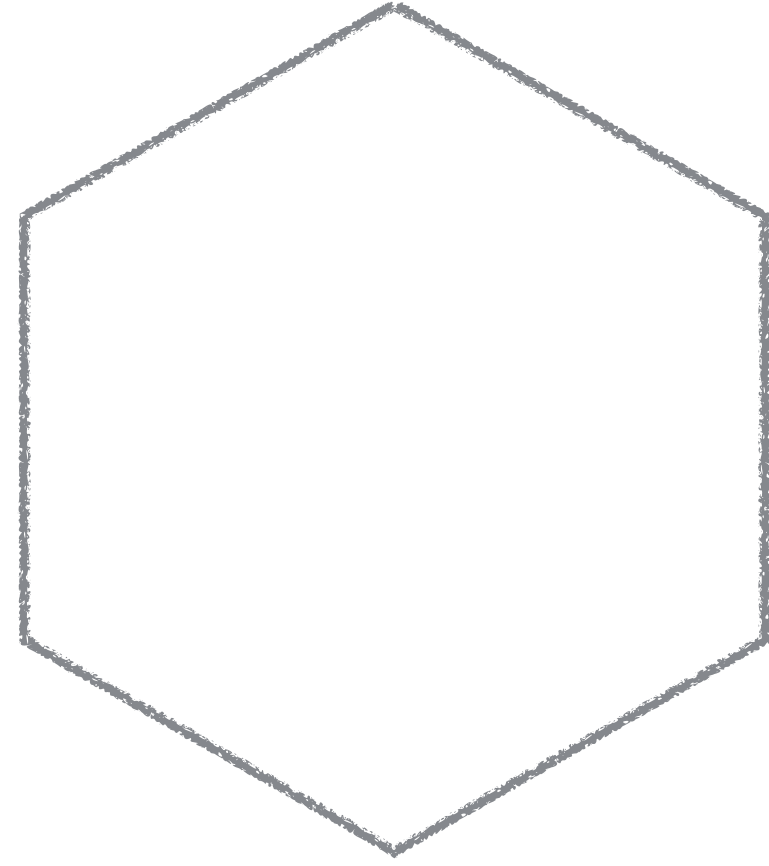
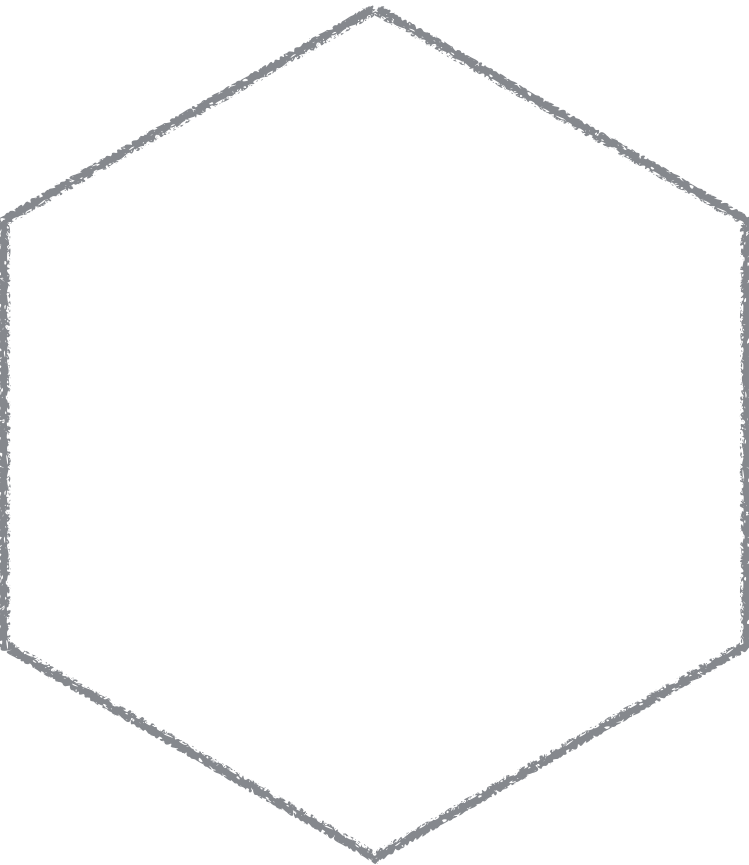
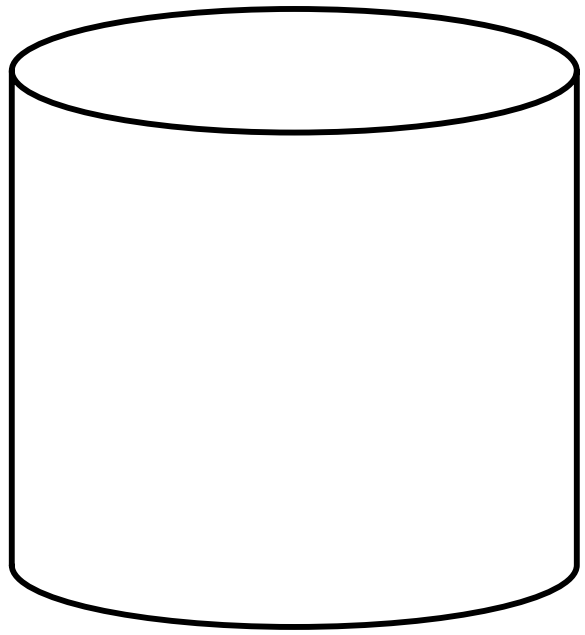
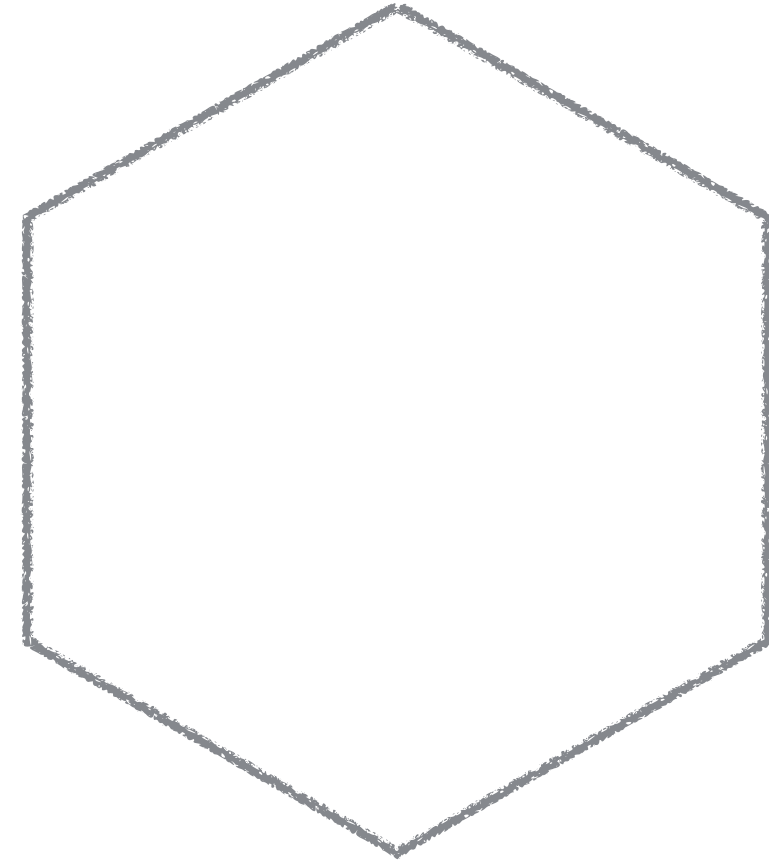
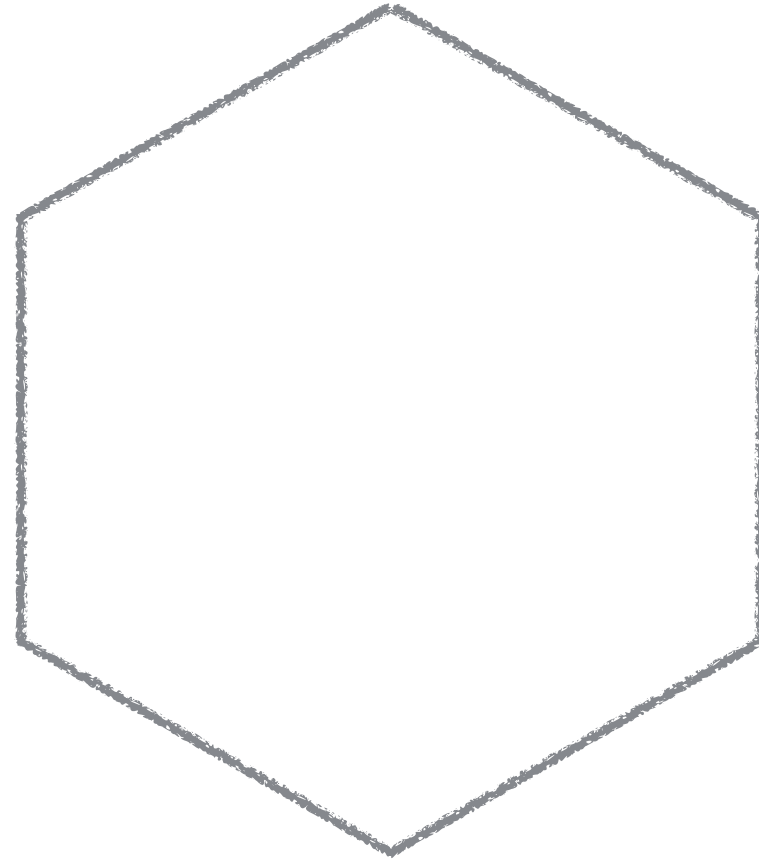
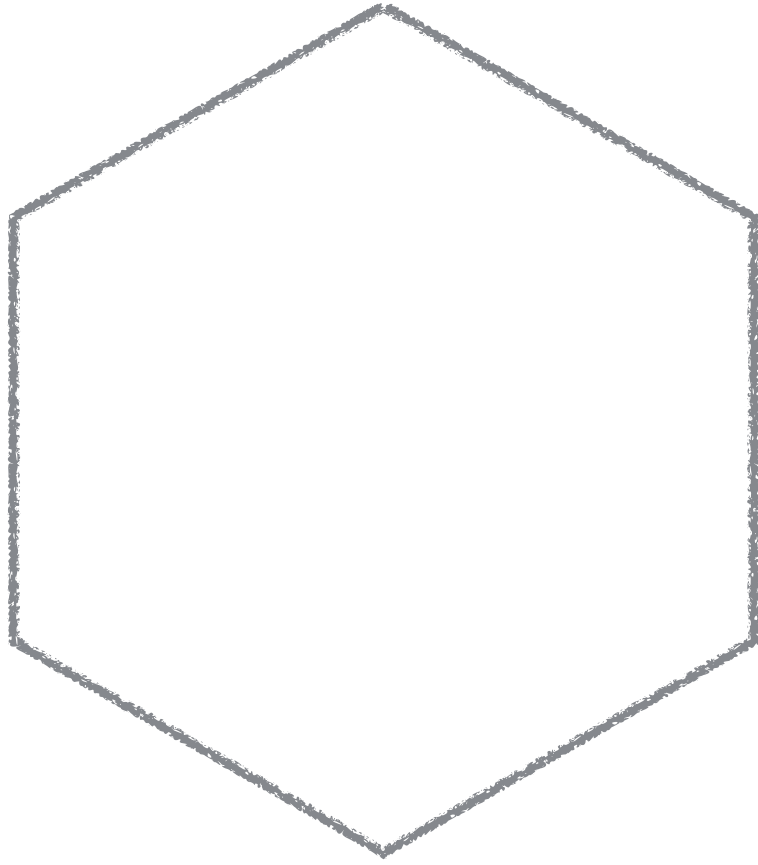


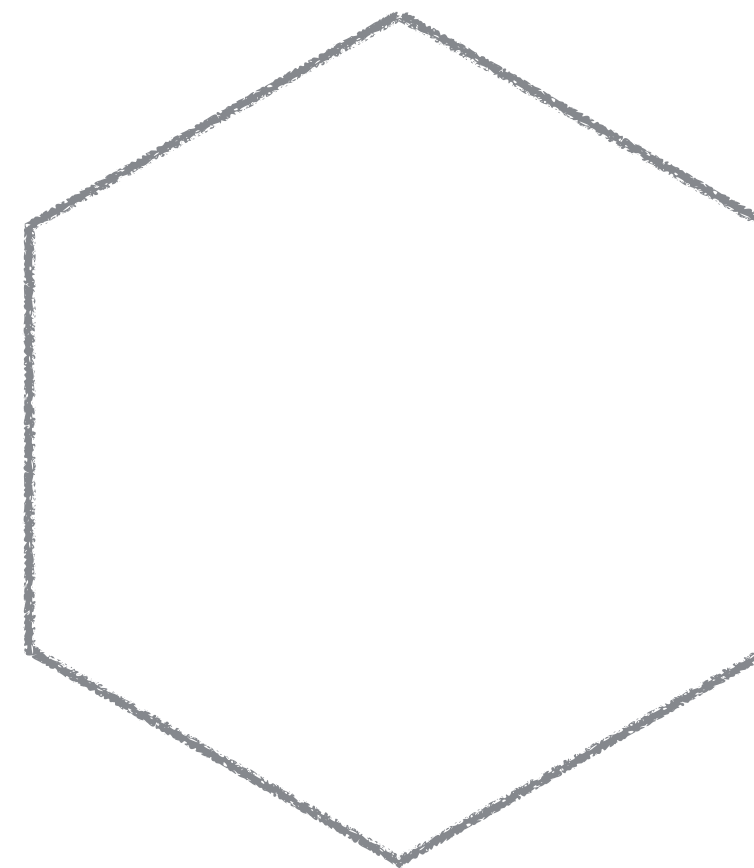
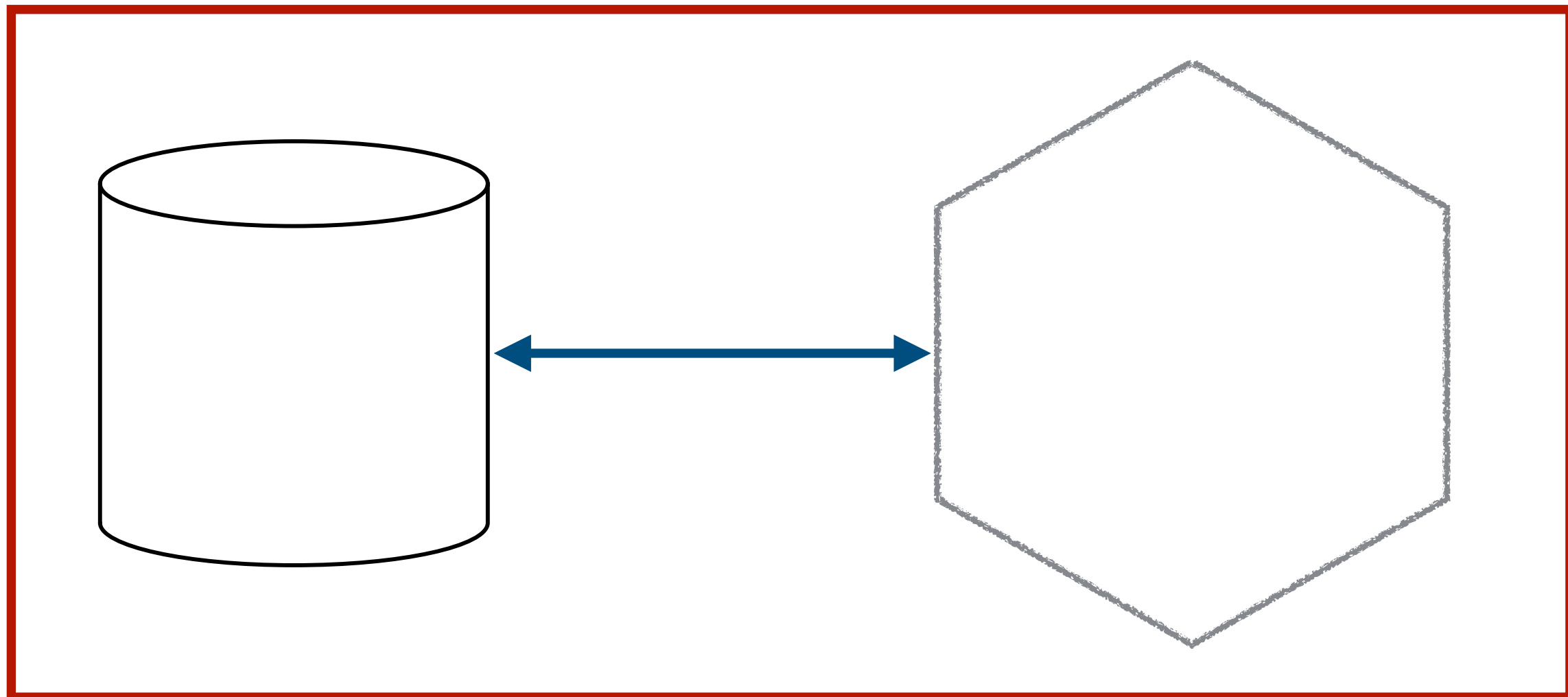
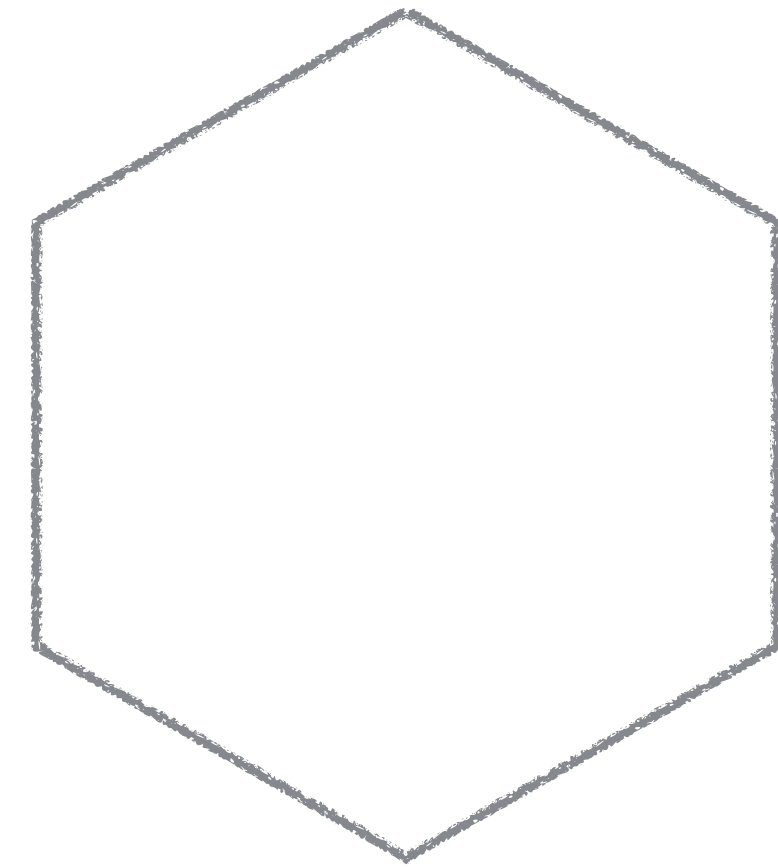
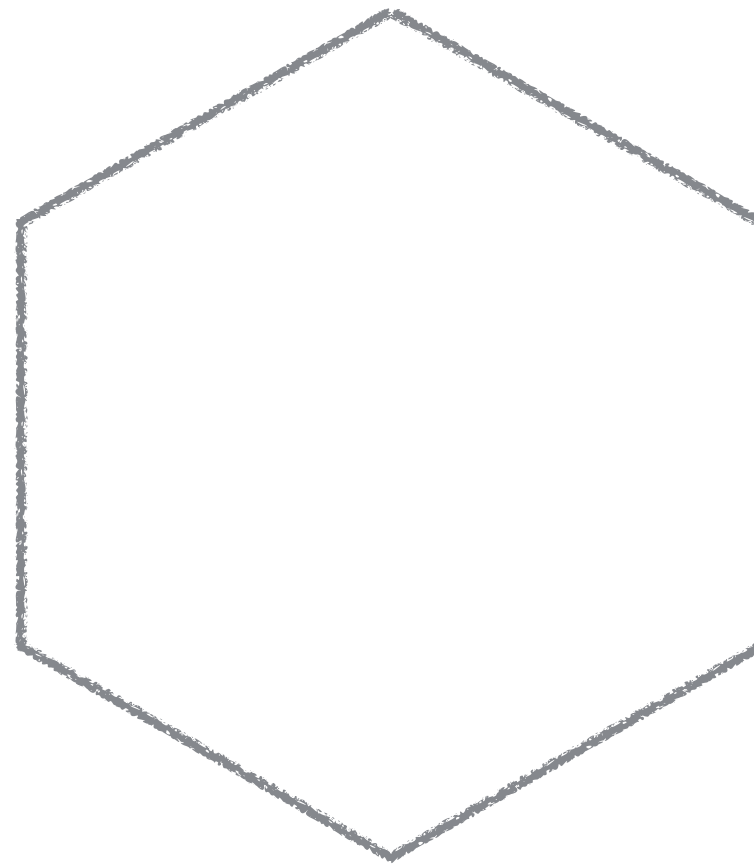
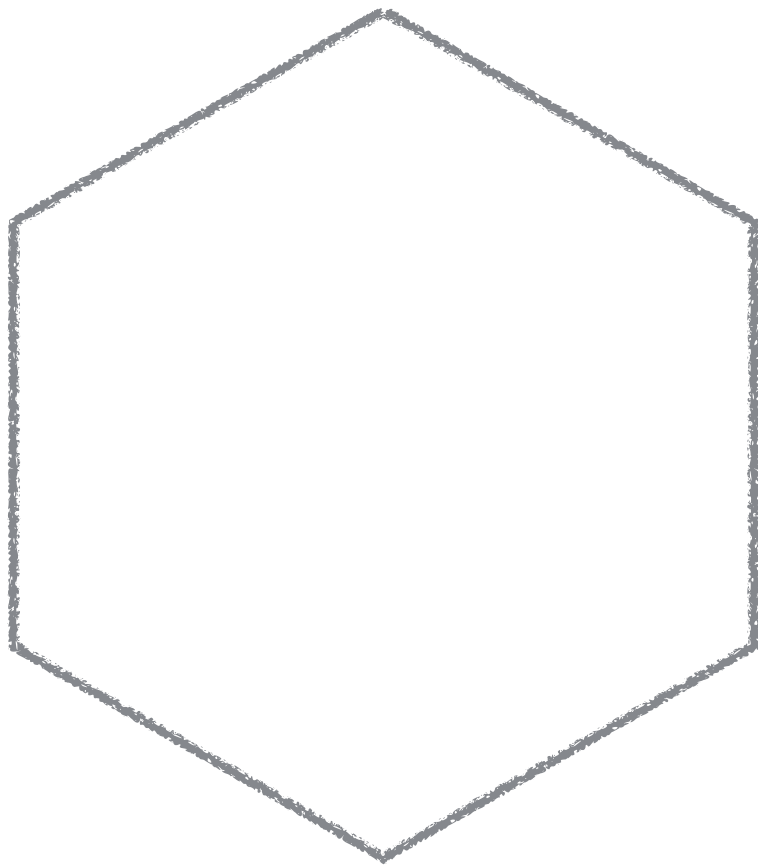













For organisations /

# Guide to the General Data Protection Regulation (GDPR)

Share  Download options 

Search this document 

Introduction

[What's new](#)

[Key definitions](#)

[Principles](#)

[Lawful basis for processing](#)

[Consent](#)

[Legitimate interests](#)

[Special category data](#)

[Criminal offence data](#)

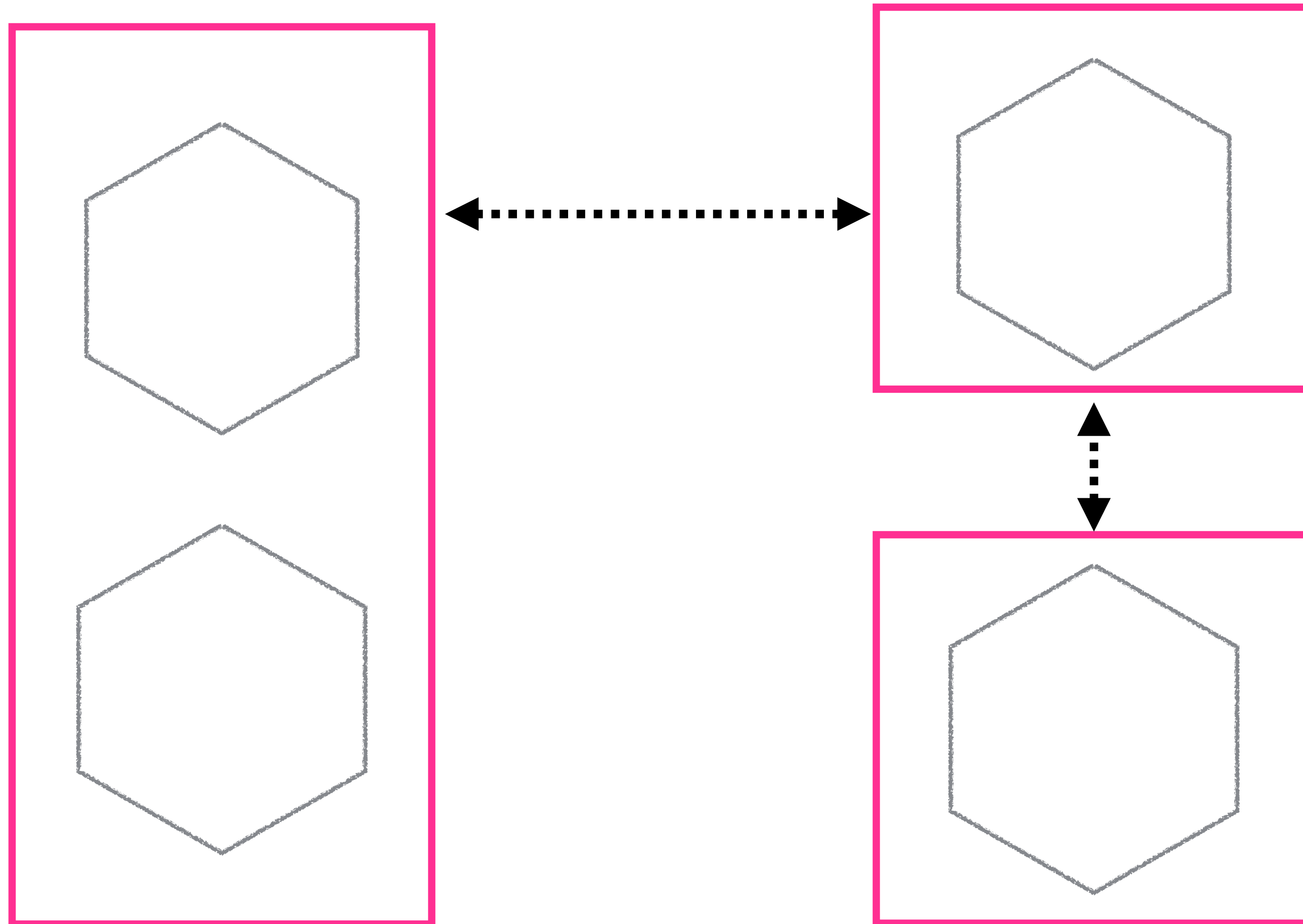
## Introduction

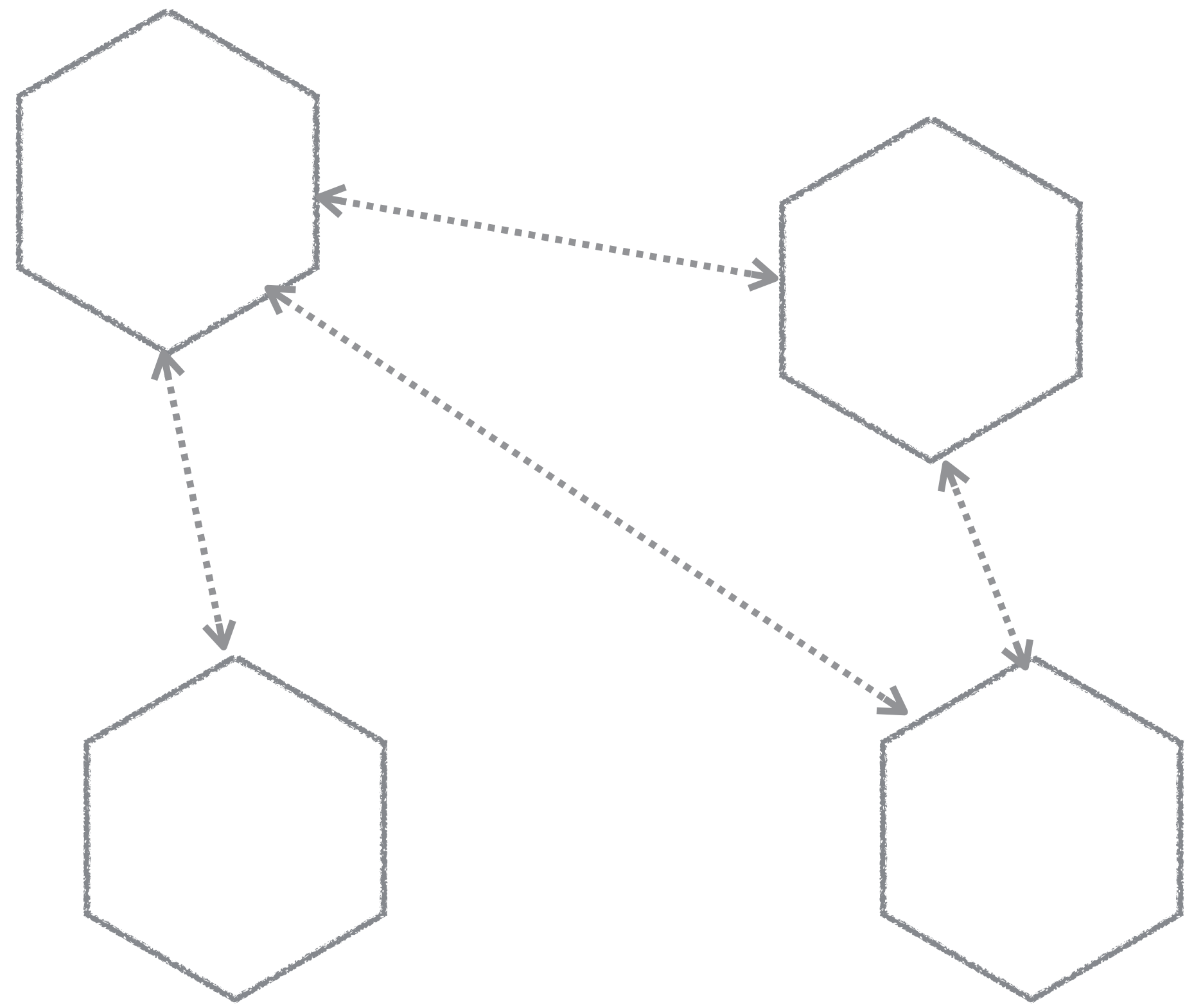
The Guide to the GDPR explains the provisions of the GDPR to help organisations comply with its requirements. It is for those who have day-to-day responsibility for data protection.

This is a living document and we are working to expand it in key areas. It includes links to relevant sections of the GDPR itself, to other ICO guidance and to guidance produced by the EU's Article 29 Working Party. The Working Party includes representatives of the data protection authorities from each EU member state, and the ICO is the UK's representative.

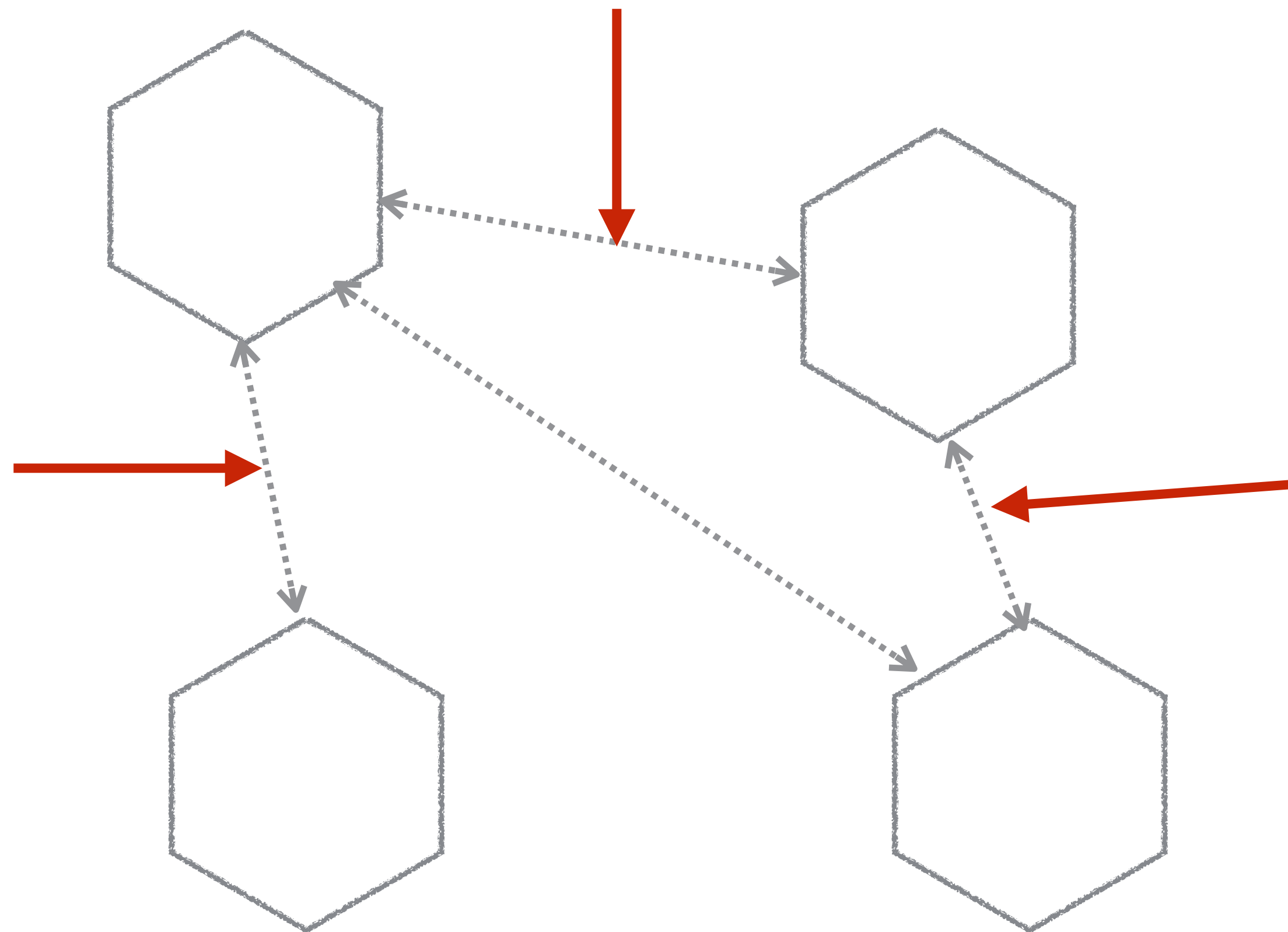
Alongside the Guide to the GDPR, we have produced a number of tools to help organisations to prepare for the GDPR:

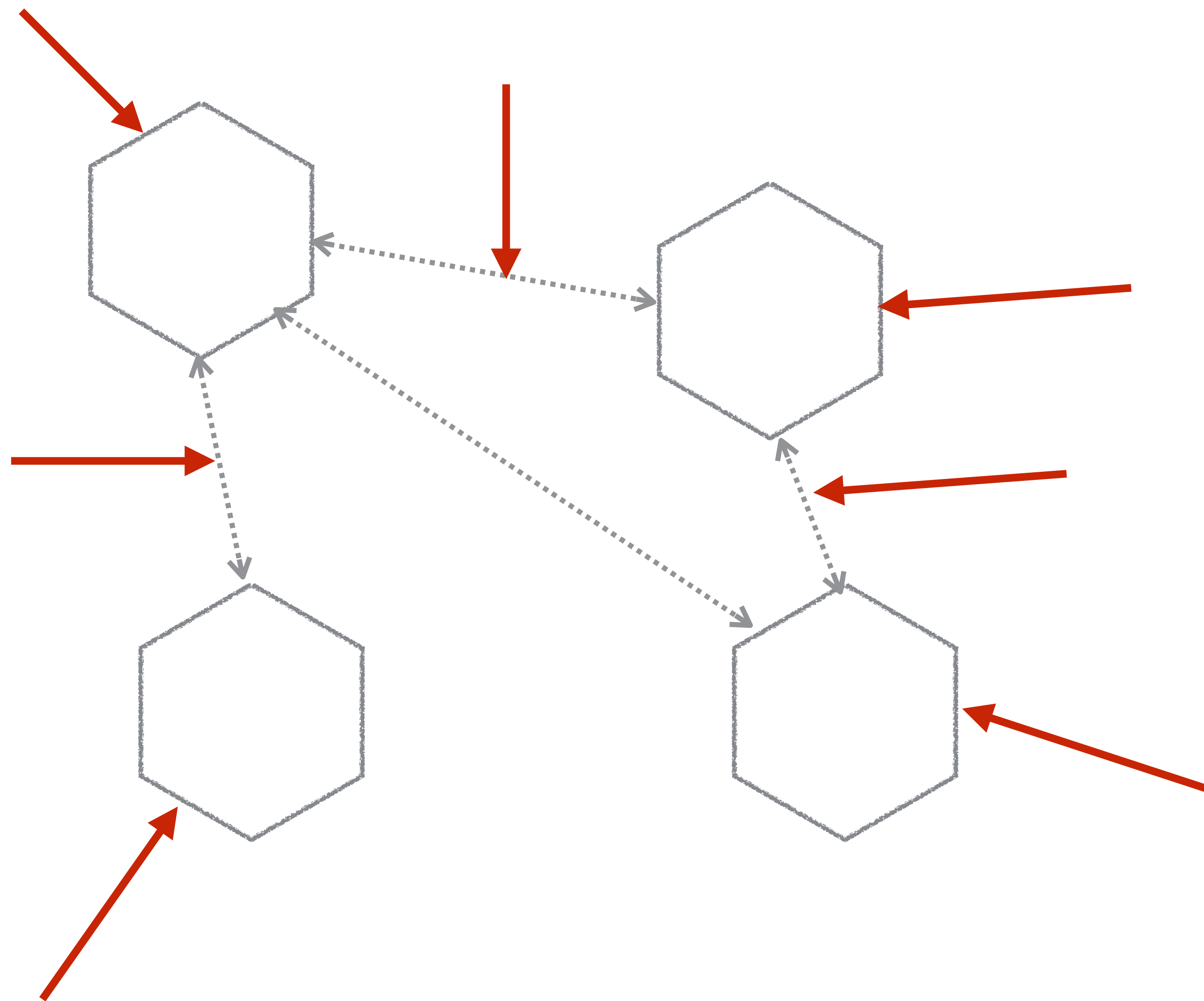
 [GDPR: 12 steps to take now](#) 











# The Basics

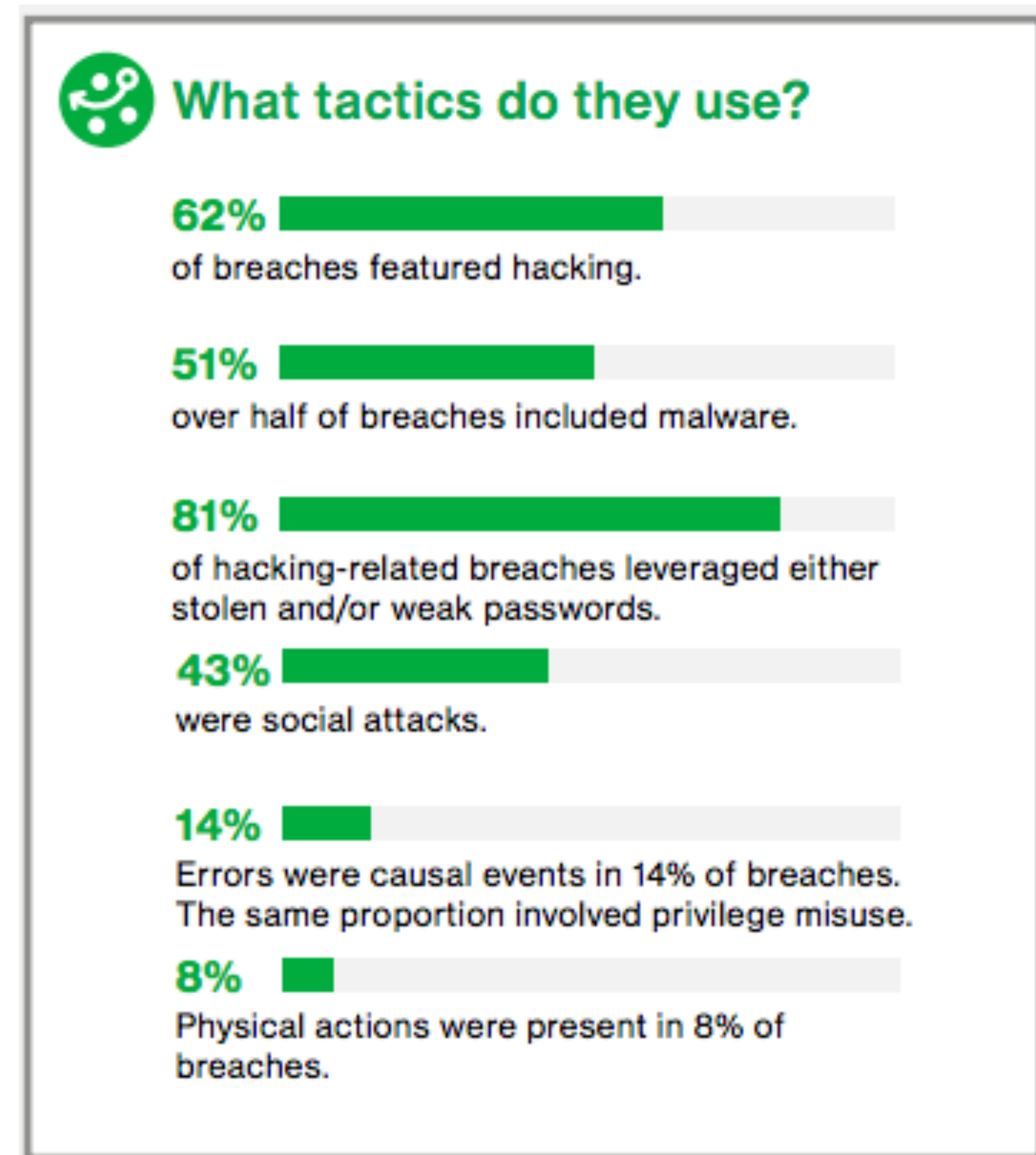
**Who here thinks they can assess risks?**



[http://www.verizonenterprise.com/resources/reports/rp\\_DBIR\\_2017\\_Report\\_en\\_xg.pdf](http://www.verizonenterprise.com/resources/reports/rp_DBIR_2017_Report_en_xg.pdf)

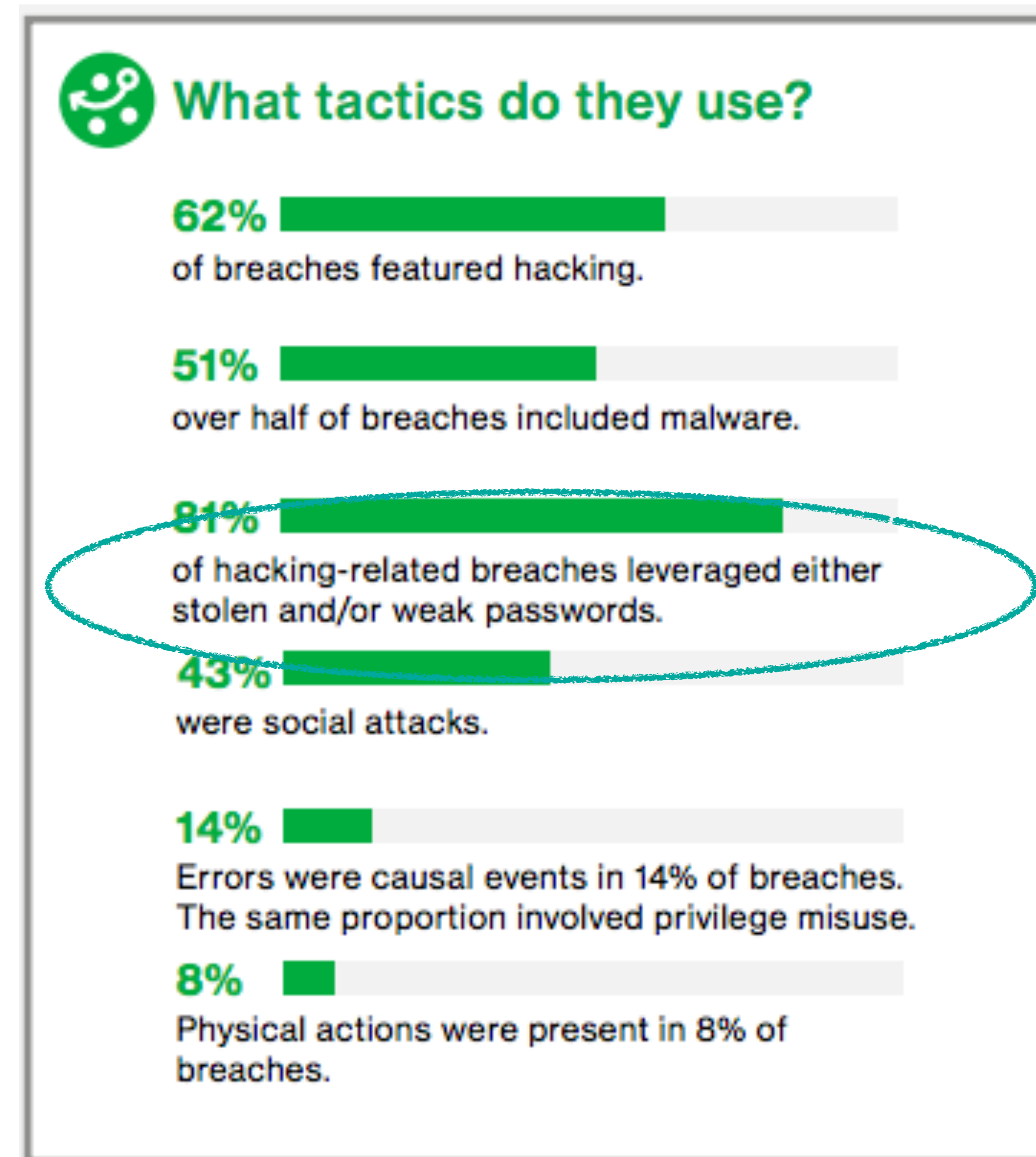


# HOW DO BREACHES OCCUR?



<http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/>

# HOW DO BREACHES OCCUR?



<http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/>

# BETTER PASSWORD RULES?

## Passwords Evolved: Authentication Guidance for the Modern Era



26 JULY 2017

In the beginning, things were simple: you had two strings (a username and a password) and if someone knew both of them, they could log in. Easy.

But the ecosystem in which they were used was simple too, for example in [MIT's Time-Sharing Computer](#), considered to be the first computer system to use passwords:



<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>



# BETTER PASSWORD RULES?

## Passwords Evolved: Authentication Guidance for the Modern Era



26 JULY 2017

In the beginning, things were simple: you had two strings (a username and a password) and if someone knew both of them, they could log in. Easy.

But the ecosystem in which they were used was simple too, for example in [MIT's Time-Sharing Computer](#), considered to be the first computer system to use passwords:



Summarises ideas from NIST and the UK's National Cyber Security Centre

<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>

# BETTER PASSWORD RULES?

## Passwords Evolved: Authentication Guidance for the Modern Era



26 JULY 2017

In the beginning, things were simple: you had two strings (a username and a password) and if someone knew both of them, they could log in. Easy.

But the ecosystem in which they were used was simple too, for example in [MIT's Time-Sharing Computer](#), considered to be the first computer system to use passwords:



Summarises ideas from NIST and the UK's National Cyber Security Centre

Packed with great tips, like...

<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>

# PASSWORDS EVOLVED

<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>

# PASSWORDS EVOLVED

Longer is stronger

<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>

# PASSWORDS EVOLVED

**Longer is stronger**

**Eliminate complex character composition rules**

**<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>**

 Send ...

 Save

**"Sorry that password won't work, you must include: a symbol, a number, a hieroglyph, a gang sign, an inspiring quote, a poem that you just wrote, a picture of your favorite animal made using only characters on your keyboard, and an uppercase letter."**

 Saved from  
**blurbsbybrenda.blogspot.com**

Visit

<https://www.pinterest.dk/pin/566679565591724157/>



# PASSWORDS EVOLVED

**Longer is stronger**

**Eliminate complex character composition rules**

**<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>**

# PASSWORDS EVOLVED

**Longer is stronger**

**Eliminate complex character composition rules**

**Embrace password managers**

**<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>**



# PASSWORDS EVOLVED

**Longer is stronger**

**Eliminate complex character composition rules**

**Embrace password managers**

**Do not mandate password changes**

**<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>**

# PASSWORDS EVOLVED

**Longer is stronger**

**Eliminate complex character composition rules**

**Embrace password managers**

**Do not mandate password changes**

**Block previously breached passwords**

<https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/>



# CHECK FOR BREACHED PASSWORDS!

## Finding Pwned Passwords with 1Password

February 22, 2018 / 68 Comments / in News, Security, Watchtower / by Shiner

Yesterday, Troy Hunt launched [Pwned Passwords](#), a new service that allows you to check if your passwords have been leaked on the Internet. His database now has more than **500 million passwords** collected from various breaches. Checking your own passwords against this list is immensely valuable.

We loved Troy's new service so much that we couldn't help but create a proof of concept that integrates it with 1Password. Here's how it looks:



The screenshot shows a 1Password form for a Gmail account. The username is 'rob@agilebits.com'. The password field contains 'zGi5wD, PP9eU4S] [0zr1c' and is followed by buttons for 'Copy', 'Conceal', 'Large Type', and 'Check Password'. A green progress bar indicates the password strength. A video player overlay is visible in the center of the password field.

<https://blog.agilebits.com/2018/02/22/finding-pwned-passwords-with-1password/>

# THE THREE R'S



Justin Smith

Follow

Identity and Security Geek

Apr 19 · 7 min read

## The Three R's of Enterprise Security: Rotate, Repave, and Repair

<https://medium.com/built-to-adapt/the-three-r-s-of-enterprise-security-rotate-repave-and-repair-f64f6d6ba29d>



## THE ADVANCED PERSISTENT THREAT

**“At or near the top of security concerns in the datacenter is something called an Advanced Persistent Threat (APT). An APT gains unauthorized access to a network and can stay hidden for a long period of time. Its goal is usually to steal, corrupt, or ransom data.”**

*- Justin Smith, Pivotal*

# Dutch intelligence first to alert U.S. about Russian hack of Democratic Party

© DO 25 JANUARI, 21:35 AANGEPAST DO 25 JANUARI, 21:44 BUITENLAND

In the Summer of 2015, Dutch intelligence services were the first to alert their American counterparts about the cyberintrusion of the Democratic National Committee by Cozy Bear, a hacking group believed to be tied to the Russian government. Intelligence hackers from Dutch AIVD (General Intelligence and Security Service) had penetrated the Cozy Bear computer servers as well as a security camera at the entrance of their working space, located in a university building adjacent to the Red Square in Moscow.

Over the course of a few months, they saw how the Russians penetrated several U.S. institutions, including the State Department, the White House, and the DNC. On all these occasions, the Dutch alerted the U.S. intelligence services, Dutch tv programme *Nieuwsuur* and *de Volkskrant*, a prominent newspaper in The Netherlands, jointly report on Thursday. This account is based on interviews with a dozen political, diplomatic and intelligence sources in The Netherlands and the U.S. with direct knowledge of the matter. None of them wanted to speak on the record, given the classified details of the matter.

<https://nos.nl/nieuwsuur/artikel/2213767-dutch-intelligence-first-to-alert-u-s-about-russian-hack-of-democratic-party.html>

# **Rotate:** Short-lived Credentials



**Rotate:** Short-lived Credentials

**Repair:** Patch Your Stuff

**Rotate:** Short-lived Credentials

**Repair:** Patch Your Stuff

**Repave:** Burn It Down!

**Rotate:** Short-lived Credentials

**Repair:** Patch Your Stuff

**Repave:** Burn It Down!

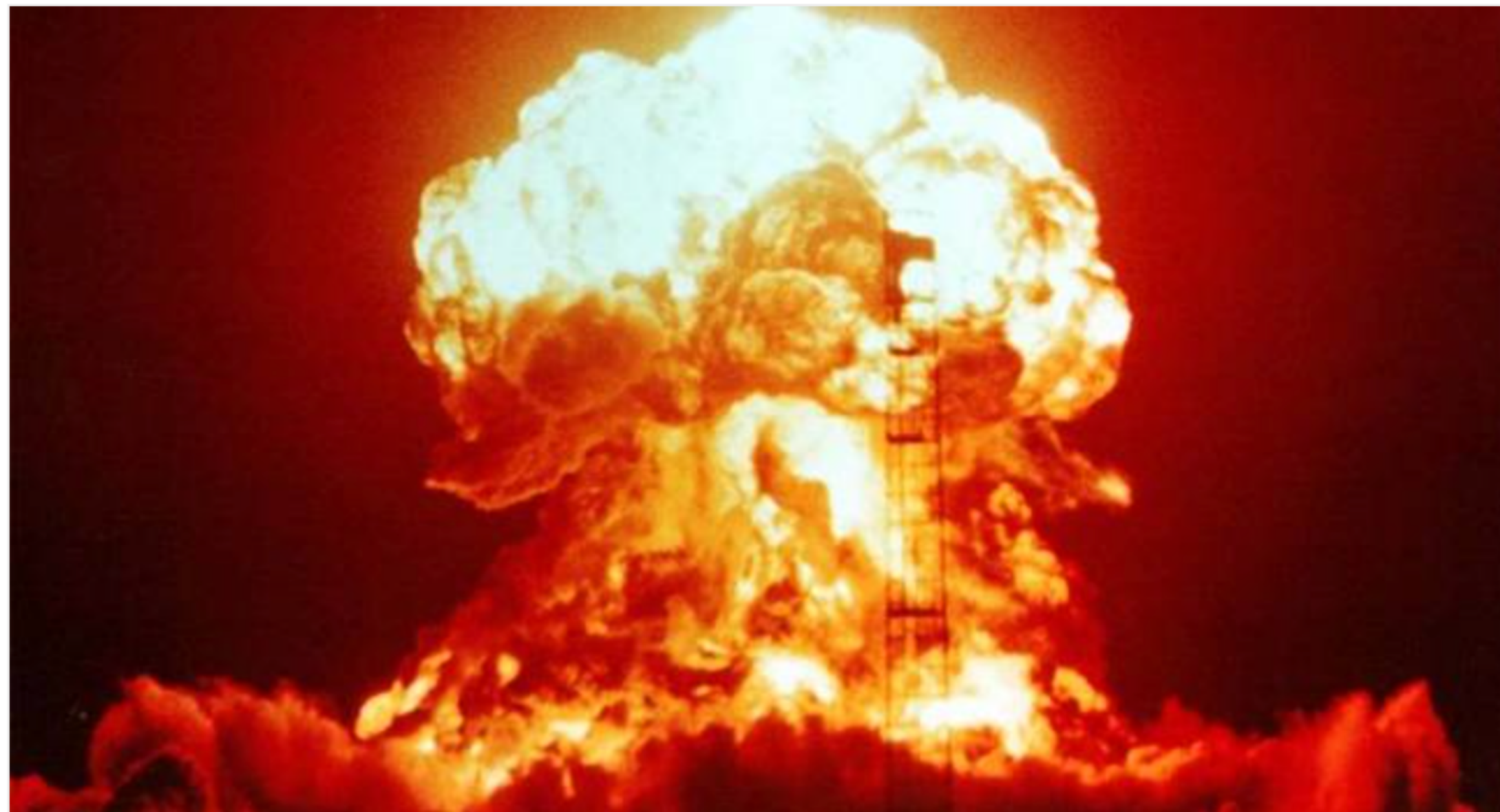


# CODESPACES R.I.P.

Data Center ► **Cloud**

## **Code Spaces goes titsup FOREVER after attacker NUKES its Amazon-hosted data**

Source-sharing site to close following total cloudpocalypse



18 Jun 2014 at 20:54, Neil McAllister



547

[http://www.theregister.co.uk/2014/06/18/code\\_spaces\\_destroyed/](http://www.theregister.co.uk/2014/06/18/code_spaces_destroyed/)

# CHECK FOR LEAKED CREDENTIALS

README.md

## Gitrob: Putting the Open Source in OSINT

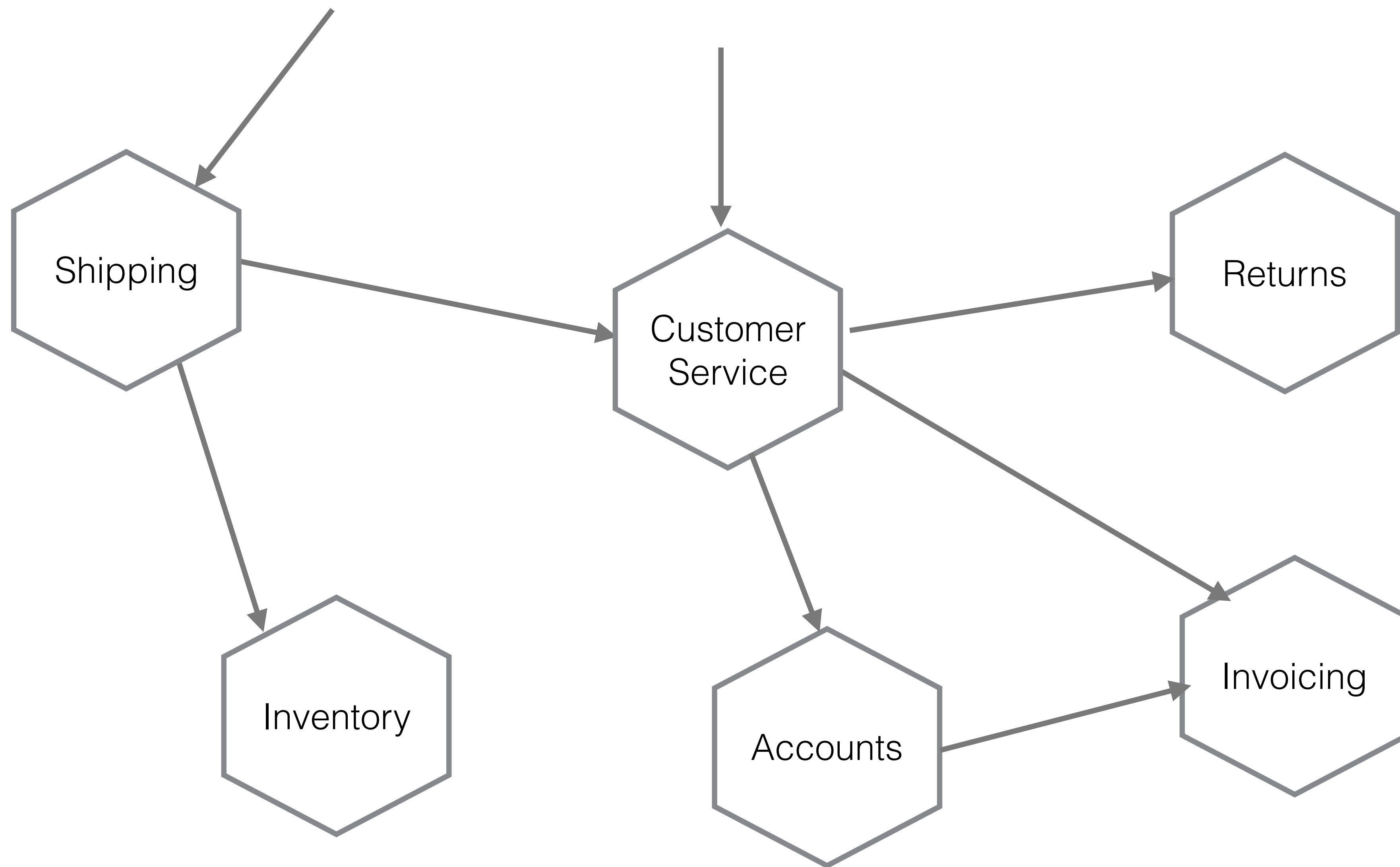
Gitrob is a command line tool which can help organizations and security professionals find sensitive information lingering in publicly available files on GitHub. The tool will iterate over all public organization and member repositories and match filenames against a range of patterns for files that typically contain sensitive or dangerous information.

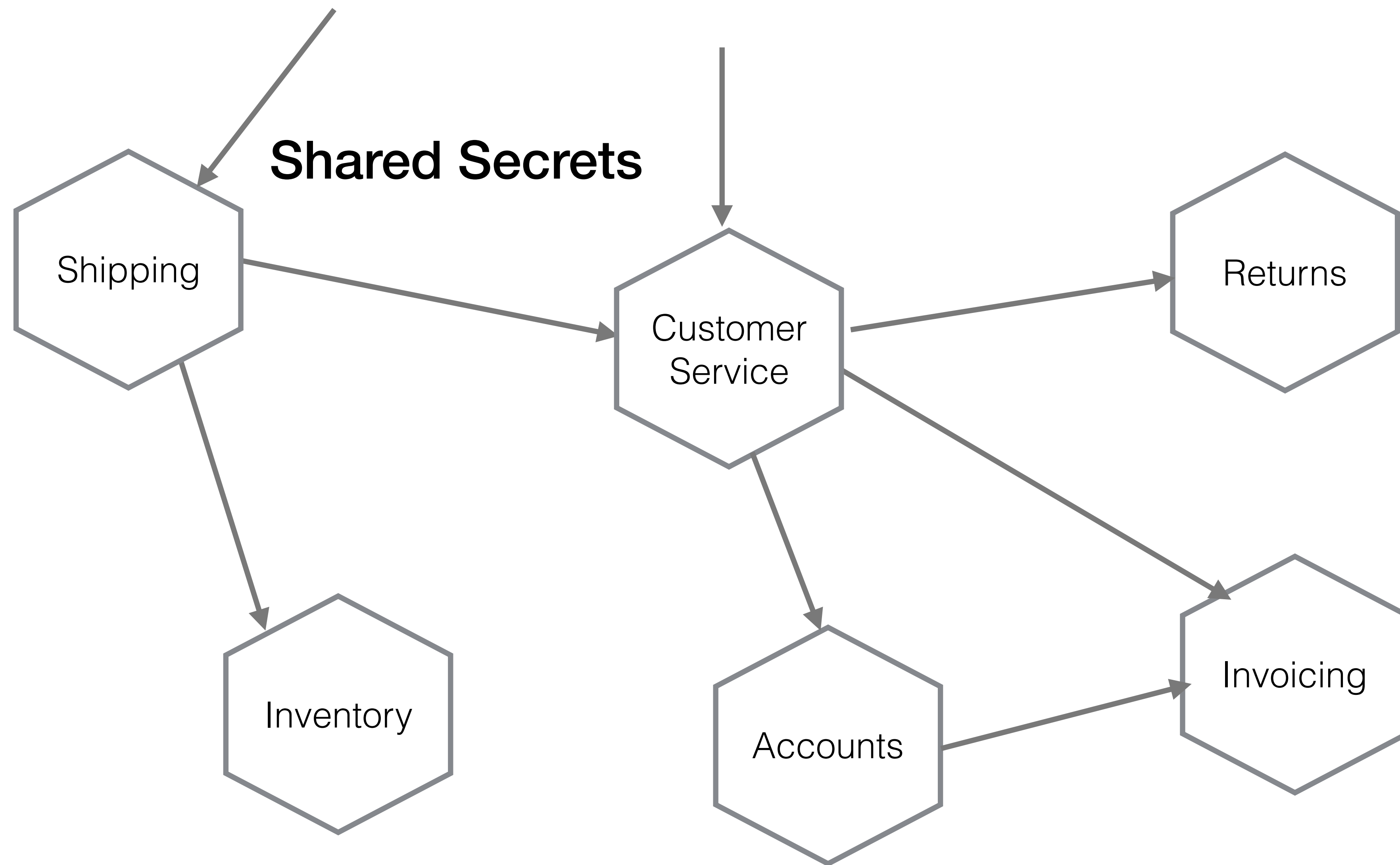
Looking for sensitive information in GitHub repositories is not a new thing, it has been [known for a while](#) that things such as private keys and credentials can be found with GitHub's search functionality, however Gitrob makes it easier to focus the effort on a specific organization.

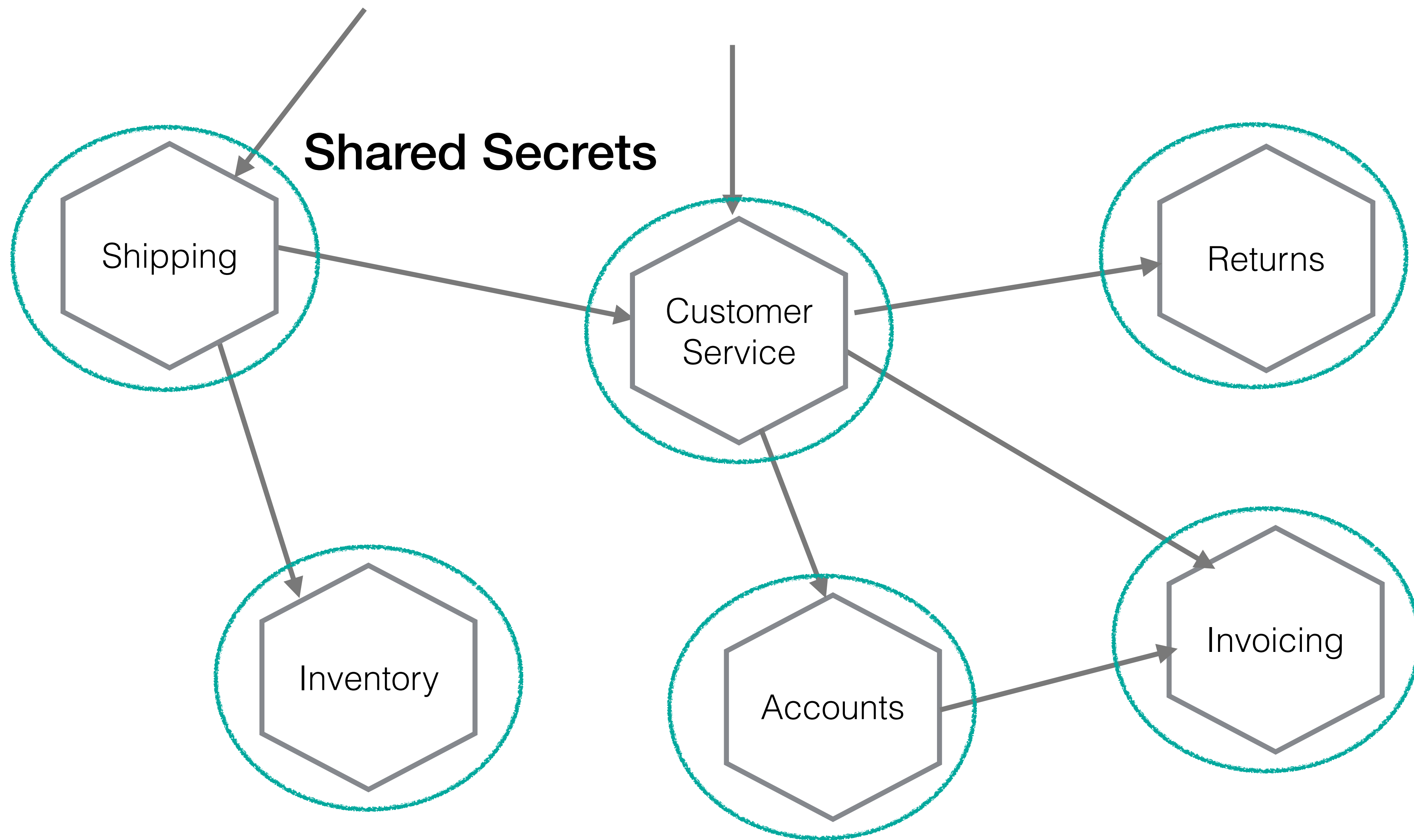
<https://github.com/michenriksen/gitrob>

**Revocation & Rotation Of Credentials**  
**+**  
**Microservices**  
**=**  
**Pain???**

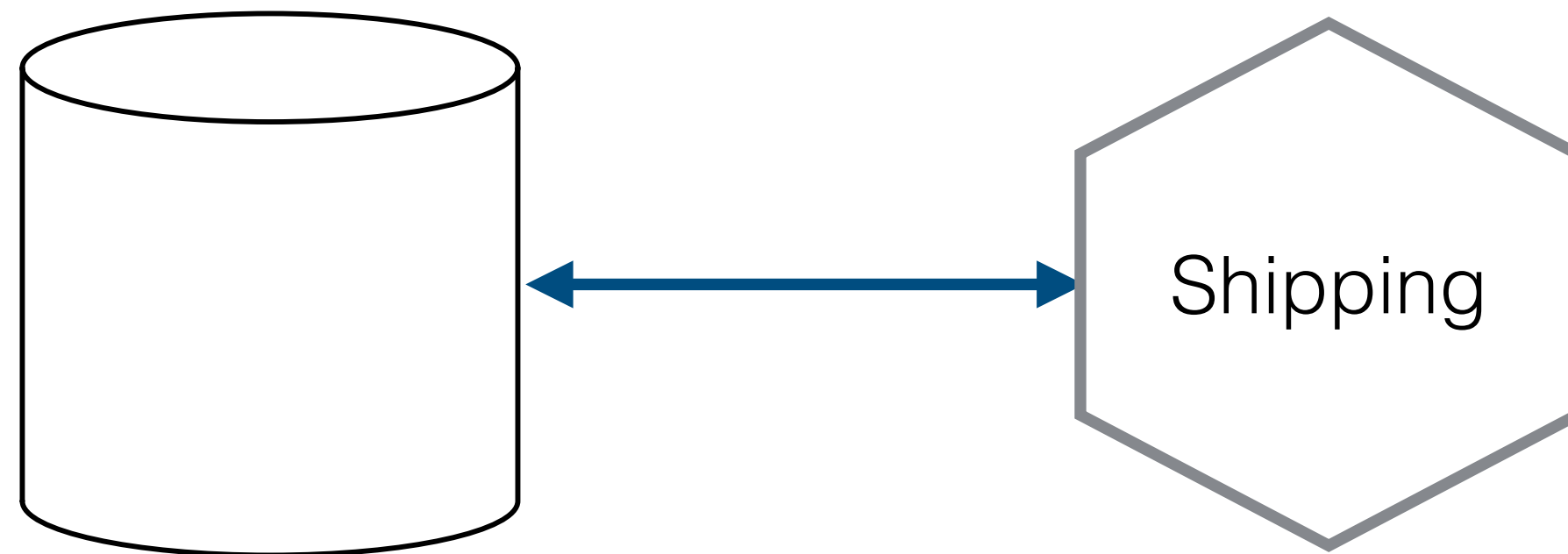


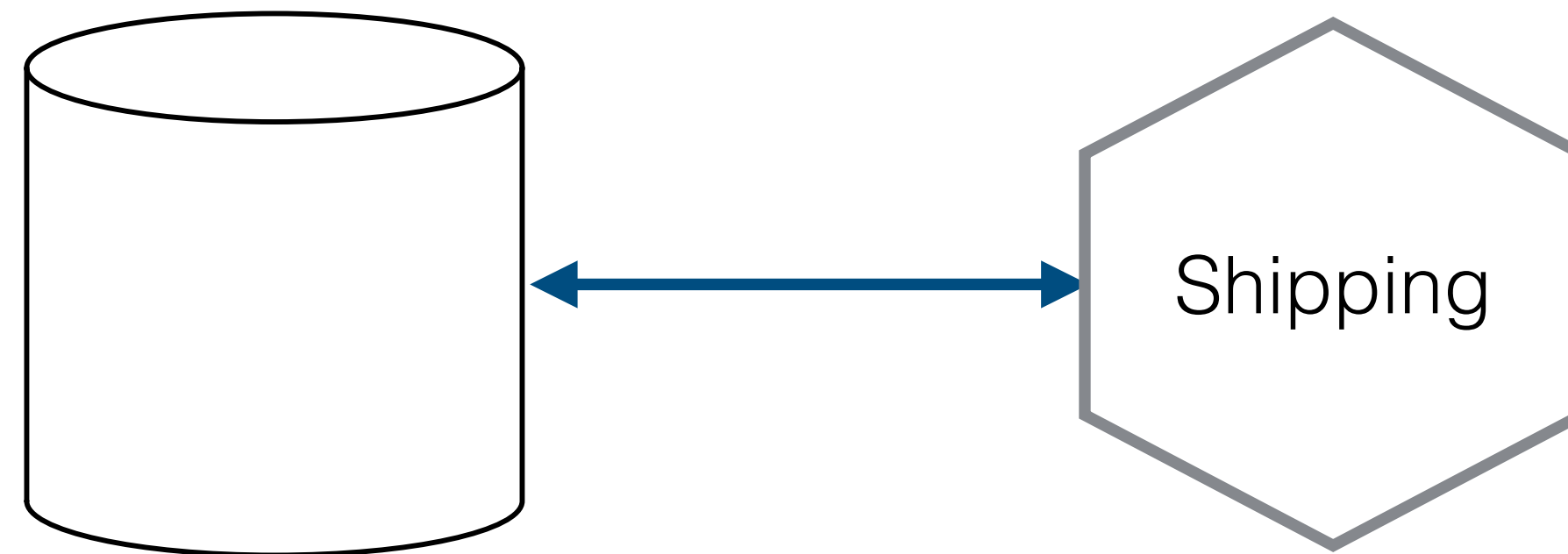




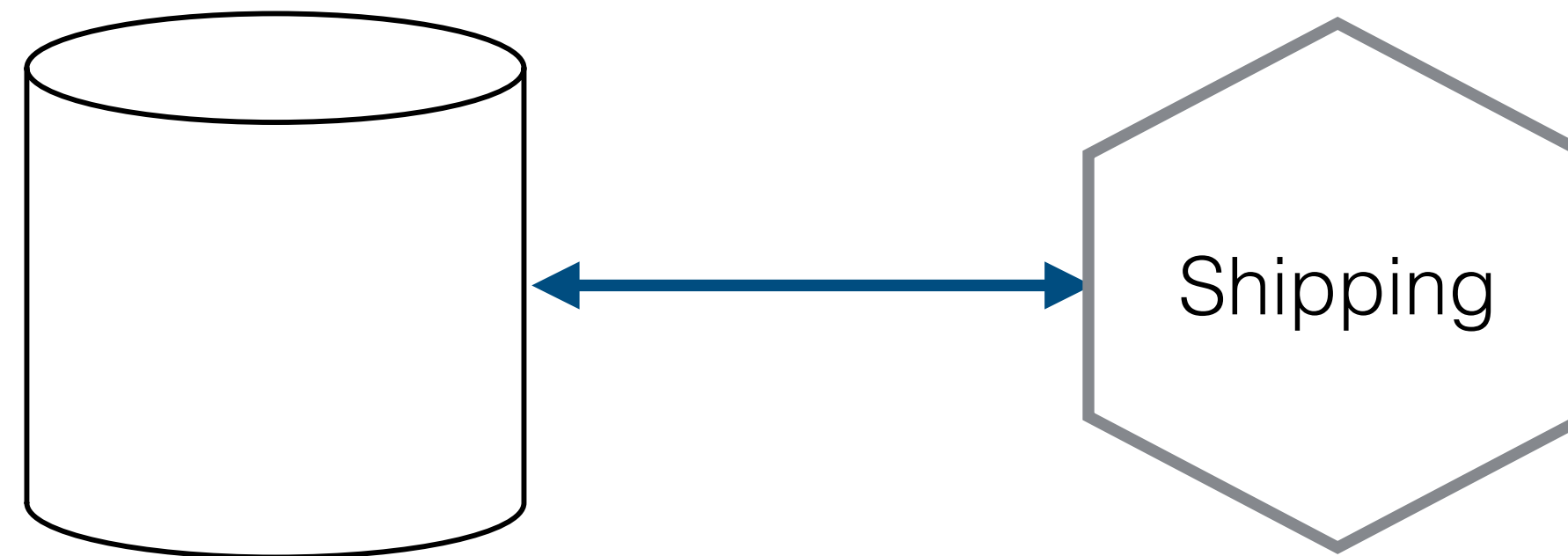








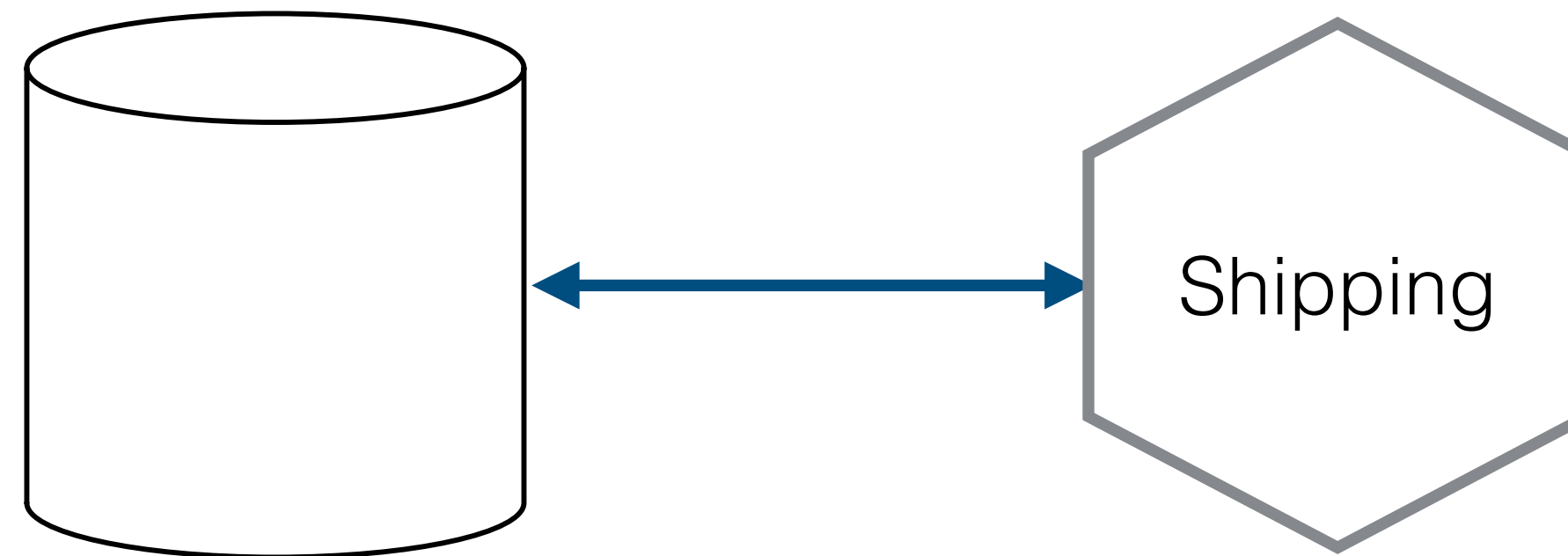
**Auth Credentials**



## Auth Credentials

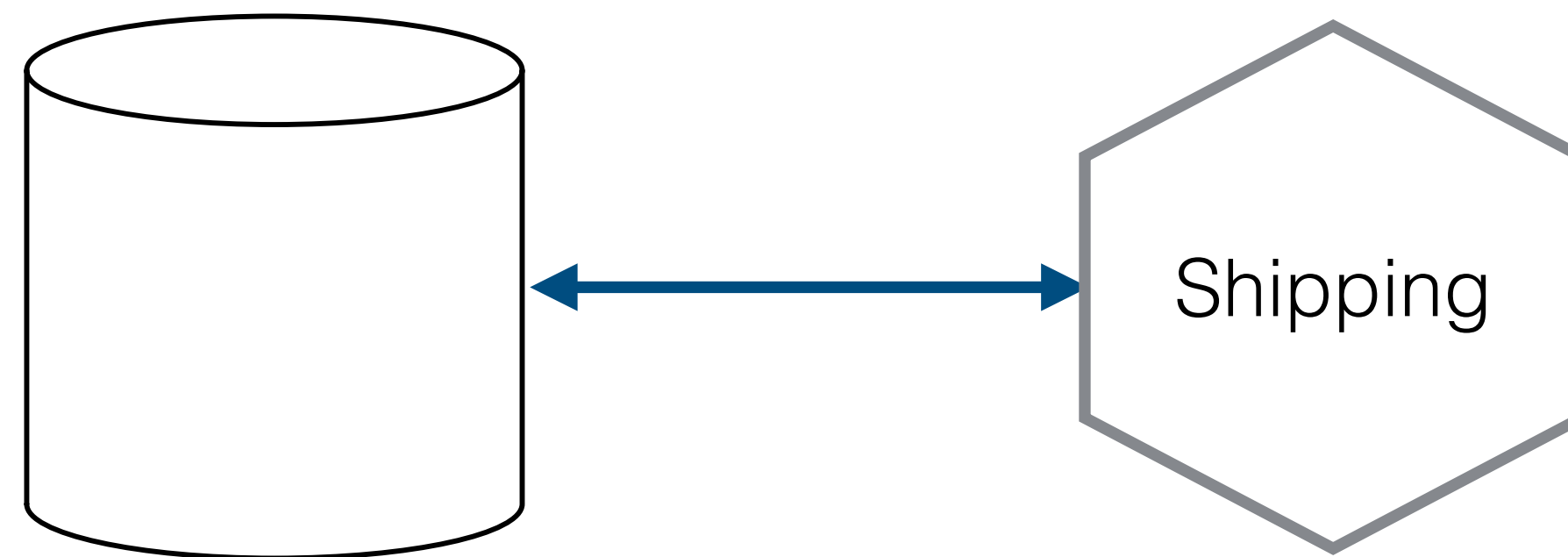
```
DB_USERNAME = admin  
DB_PASSWORD = 123ask48321
```





## Auth Credentials

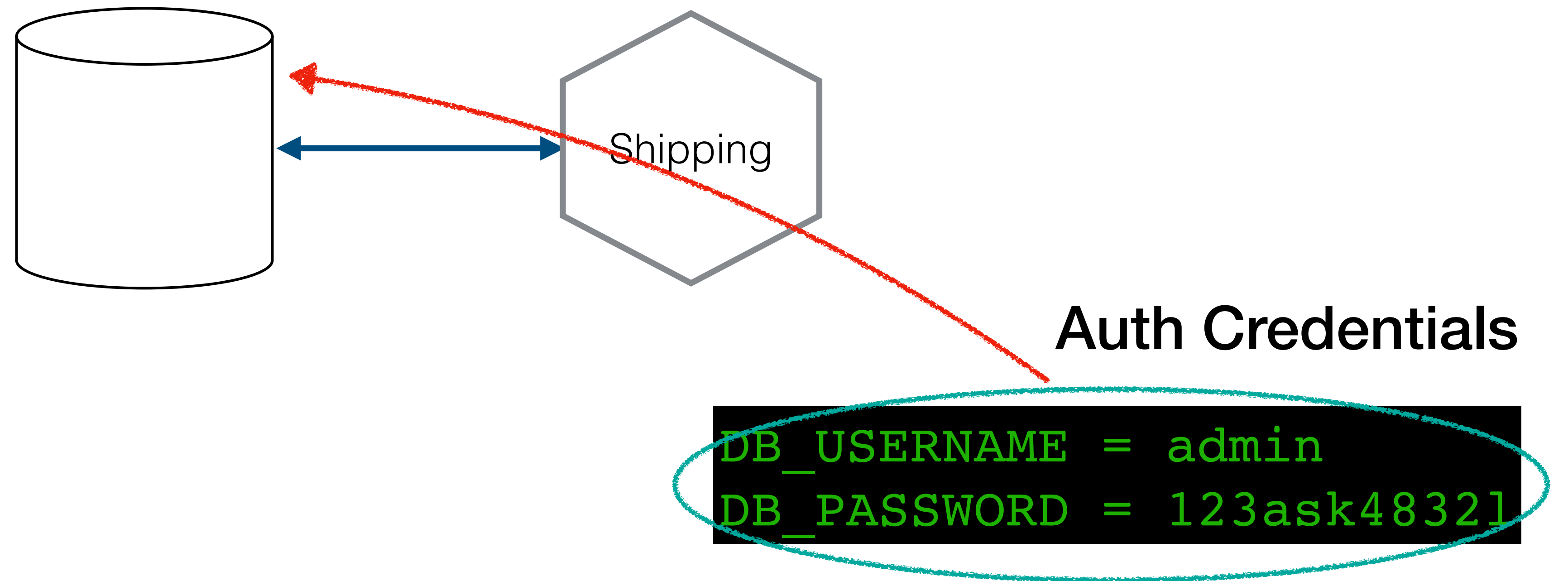
```
DB_USERNAME = admin  
DB_PASSWORD = 123ask48321
```



## Auth Credentials

```
DB_USERNAME = admin  
DB_PASSWORD = 123ask48321
```

Leaving credentials in the open can be bad...



Leaving credentials in the open can be bad...



# Secret stores!



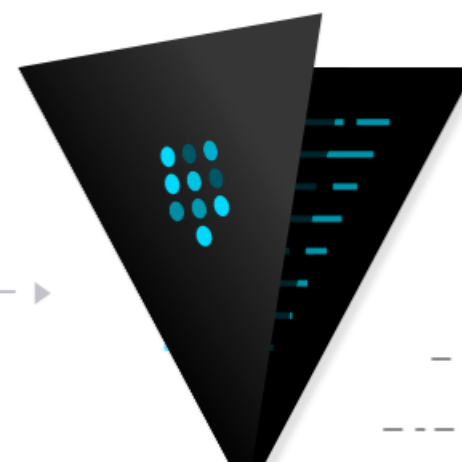
A tool for managing secrets.

[Get Started](#)



[Launch Interactive Tutorial](#)

```
$ vault get api-key  
kk9290jf2mun9m09v20  
ivn20b2vg
```





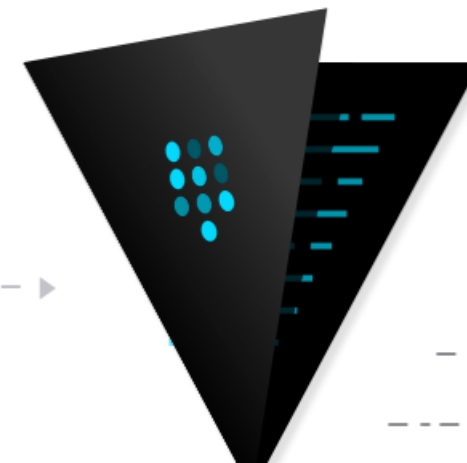
A tool for managing secrets.

[Get Started](#)



[Launch Interactive Tutorial](#)

```
$ vault get api-key  
kk9290jf2mun9m09v20  
ivn20b2vg
```



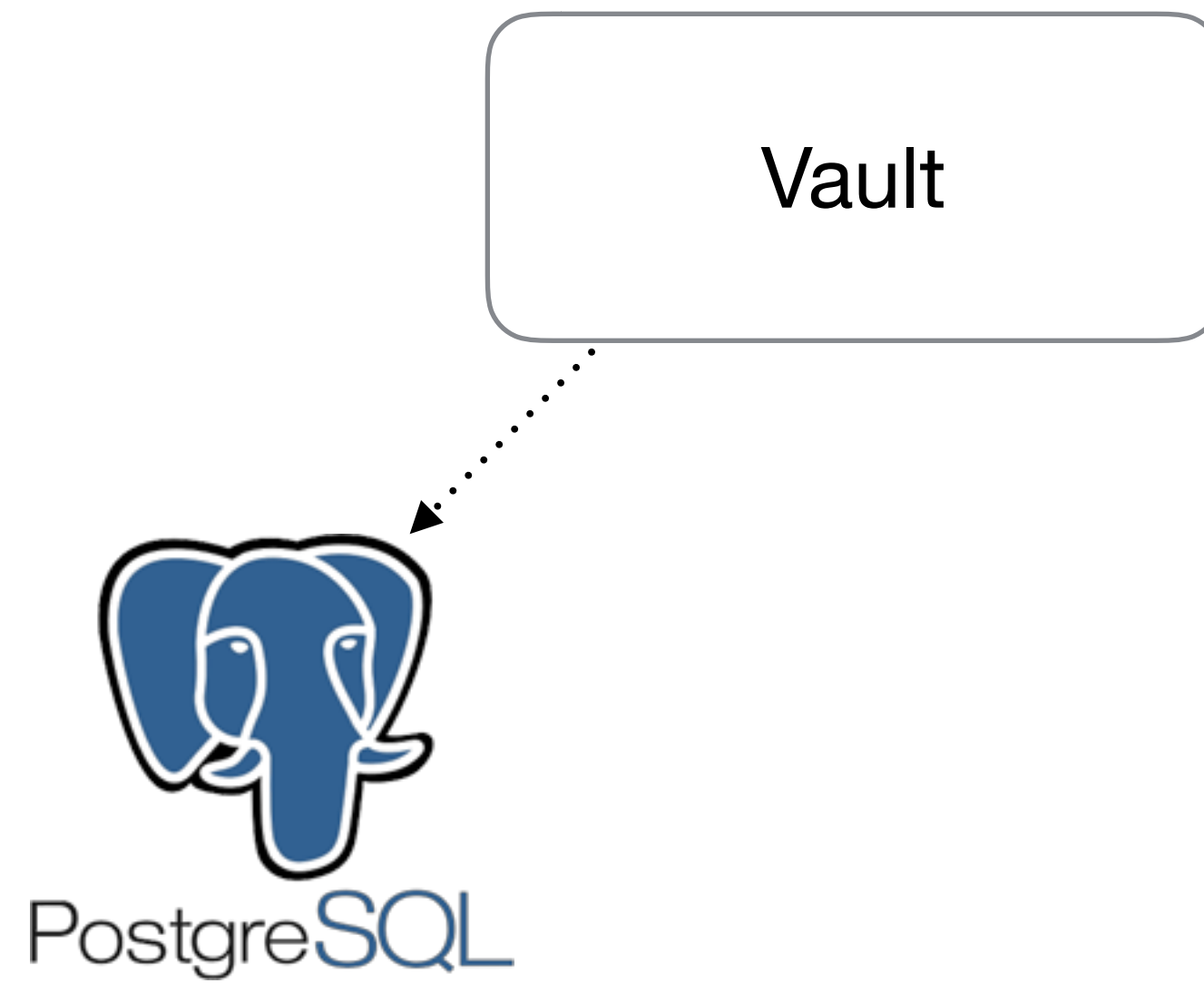
AWS Key Management Service  
(KMS)



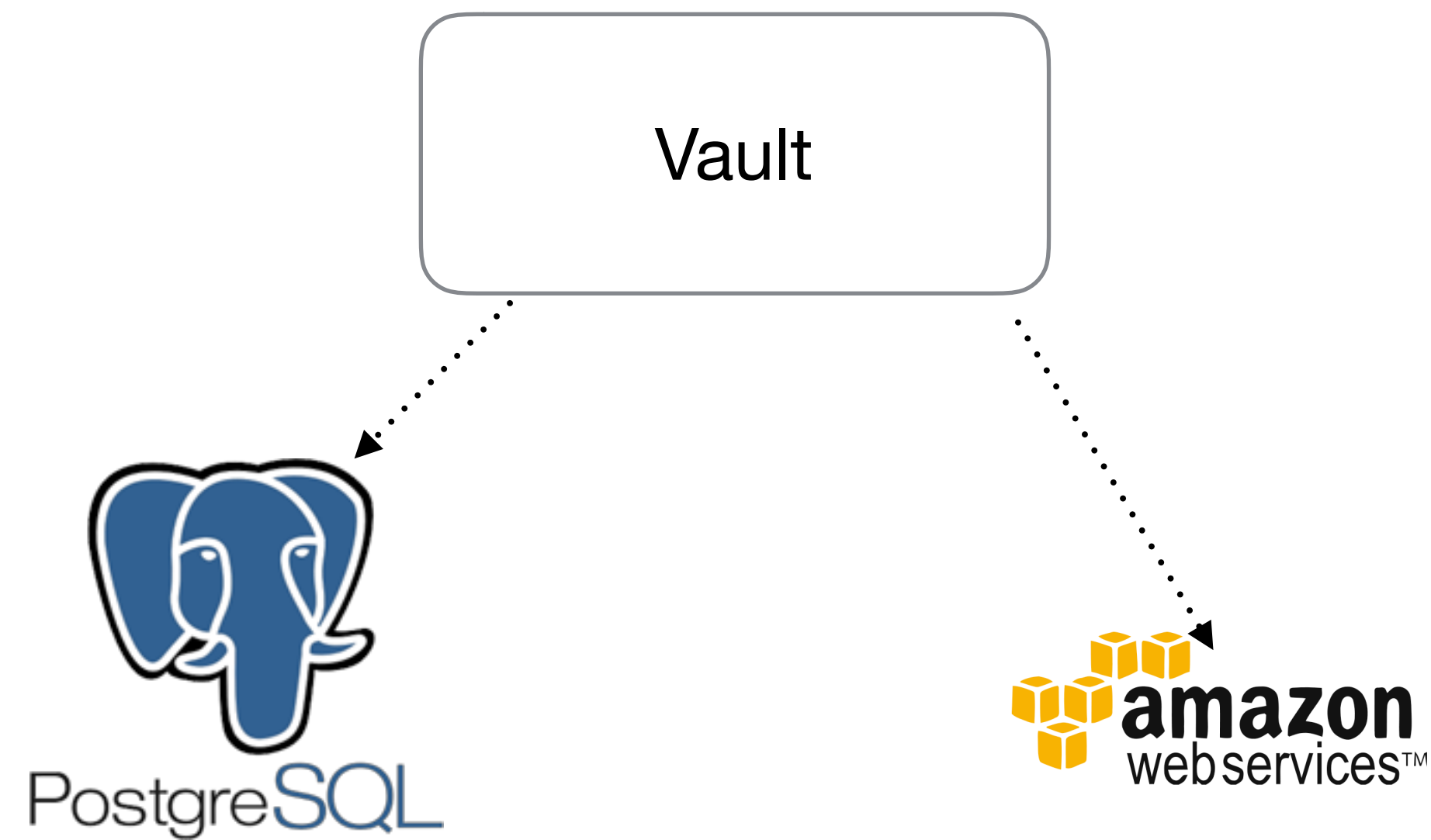
# TIME-LIMITED CREDENTIALS

Vault

# TIME-LIMITED CREDENTIALS

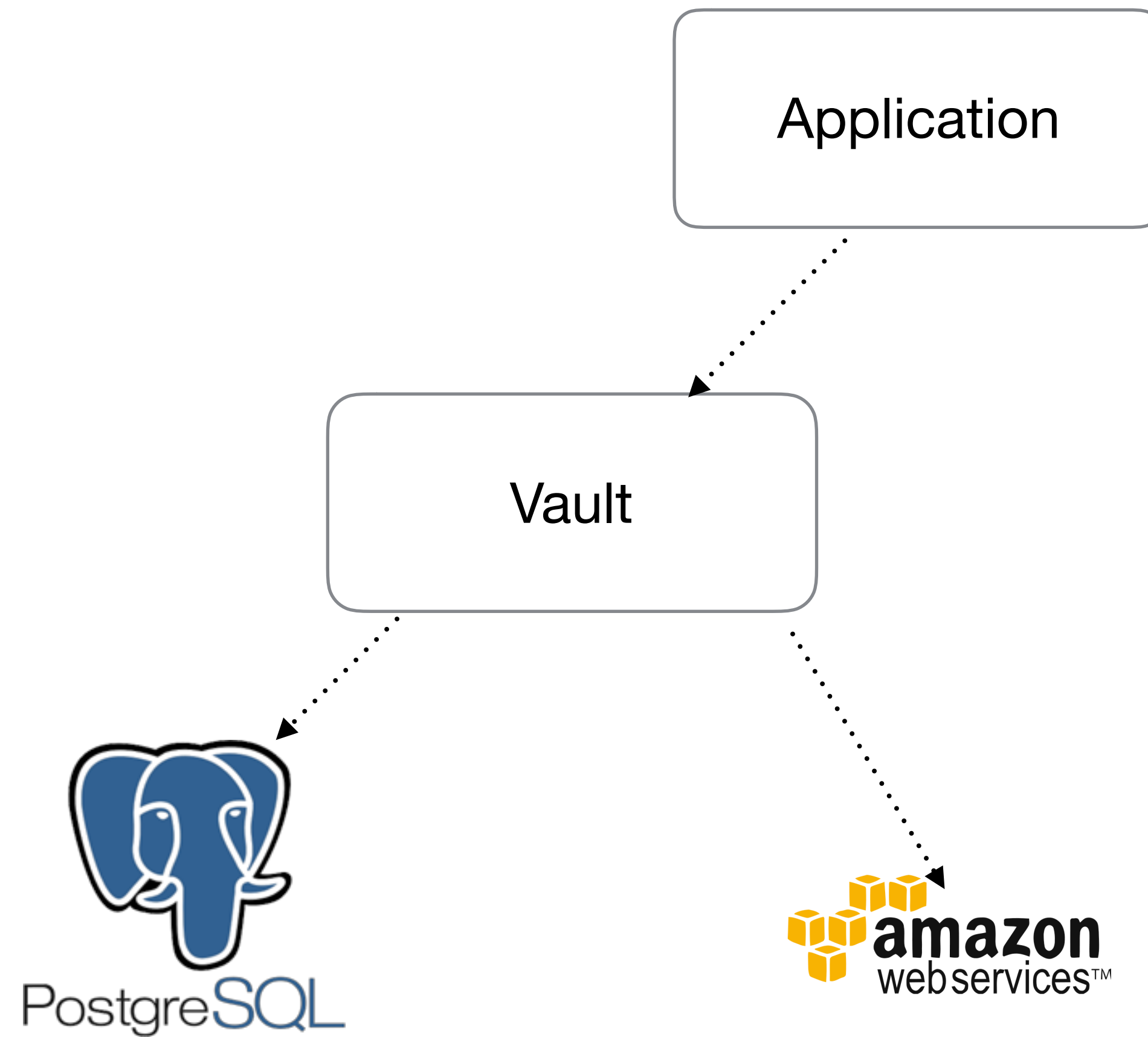


# TIME-LIMITED CREDENTIALS

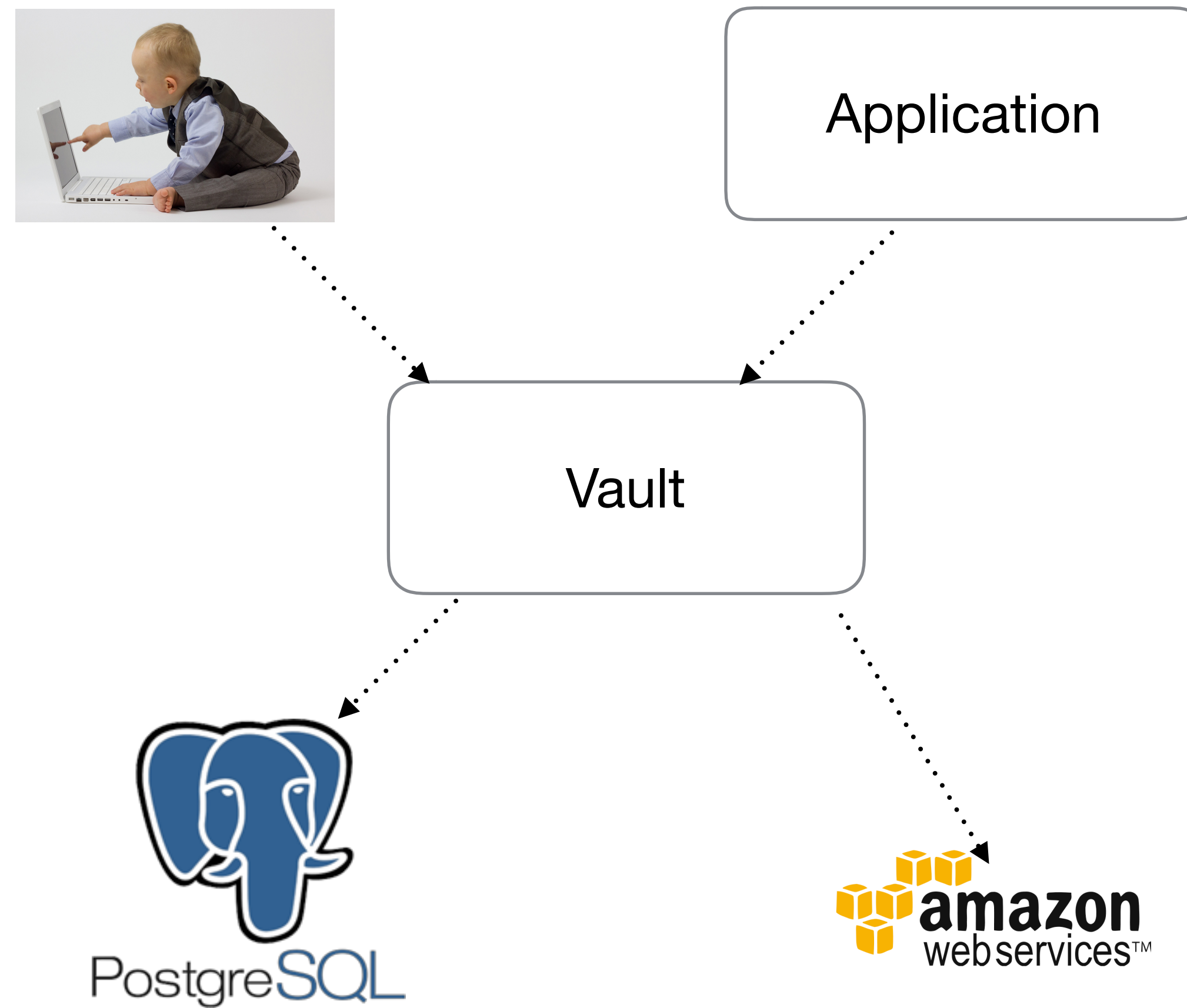




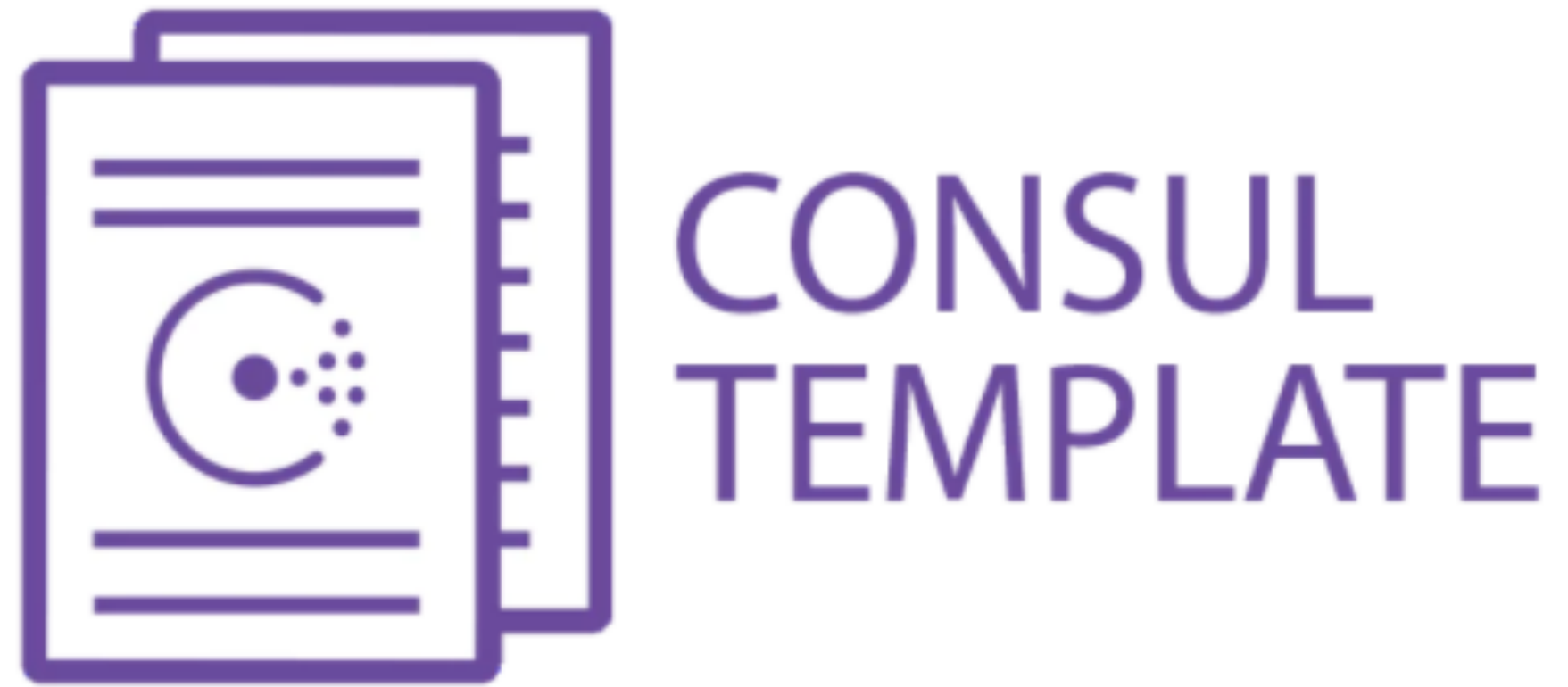
# TIME-LIMITED CREDENTIALS



# TIME-LIMITED CREDENTIALS



# AWESOMENESS



<https://github.com/hashicorp/consul-template>



# AWESOMENESS



```
adapter: postgresql
host: {{key "my-app/production/host"}}
username: {{$secret.Data.username}}
password: {{$secret.Data.password}}
{{end}}
```

<https://github.com/hashicorp/consul-template>

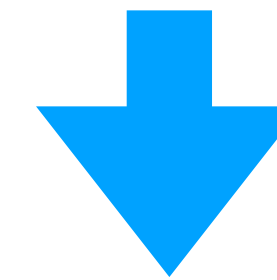
From: <https://www.hashicorp.com/blog/using-vault-with-consul-template>

# AWESOMENESS



<https://github.com/hashicorp/consul-template>

```
adapter: postgresql
host: {{key "my-app/production/host"}}
username: {{$secret.Data.username}}
password: {{$secret.Data.password}}
{{end}}
```



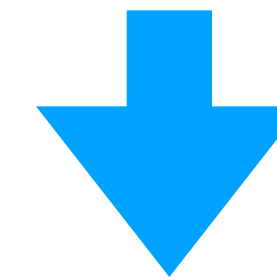
From: <https://www.hashicorp.com/blog/using-vault-with-consul-template>

# AWESOMENESS



<https://github.com/hashicorp/consul-template>

```
adapter: postgresql
host: {{key "my-app/production/host"}}
username: {{$secret.Data.username}}
password: {{$secret.Data.password}}
{{end}}
```



```
adapter: postgresql
host: db-service-183.corp.com
username: as15593kd235423
password: fk1k11492309482
{{end}}
```

From: <https://www.hashicorp.com/blog/using-vault-with-consul-template>



## WHAT ELSE CAUSES BREACHES?

**“44 percent of security breaches occur after vulnerabilities and solutions have been identified. In other words, the problems could have been avoided if found vulnerabilities had been addressed sooner.”**

**- Forbes/BMC, 2016**

# Massive Equifax data breach - what you need to know



By [Callum Mason](#), News Reporter  
12 Sep 2017 | Updated 19 Sep 2017



Credit report heavyweight Equifax has warned that up to 400,000 UK consumers may have had their personal details stolen as part of a massive global data breach. Info on exactly who's been affected and what you can do about it is still somewhat sketchy, but here's what we know.

Equifax revealed on 8 September that 143 million consumers in the US could have been affected by the incident, which saw hackers access data such as names, address and dates of birth, as well as credit card numbers in a smaller number of cases.

Although its UK business – Equifax Ltd – now says systems in this country are not affected, it admits a file which was stored in the US and contained more limited personal information on up to 400,000 UK consumers may have been accessed.

### Related MSE Guides

#### Credit Scores

Bust myths & improve your score

#### 30+ Ways to Stop Scams

As scams get clever, we need to too!

#### Check your credit report for free

Grab your file and check your score, or even get PAID to do it



### Get Our Free Money Tips Email!

For all the latest deals, guides and loopholes - join the 12m who get it.

*Don't miss out*

Enter Email Address

GET IT!

[FAQs](#) | [Privacy Policy](#) | [Past Emails](#) | [Unsubscribe](#)

### What is Equifax and what data does it have?

Equifax is the second biggest credit reference agency in the UK, after Experian.

<https://www.moneysavingexpert.com/news/protect/2017/09/massive-equifax-data-breach---what-you-need-to-know>

# PATCH MUCH?

## Equifax confirms march struts vulnerability behind breach

by Chris Brook for Threat PostEquifax said the culprit behind this summer's massive breach of 143 million Americans was indeed CVE-2017-5638, an Apache Struts vulnerability patched back in March.

September 14, 2017 , 4:00 pm

The bug was widely assumed by experts to be the "U.S. website application vulnerability" implicated by the company last Thursday, especially after an Apache spokeswoman **told Reuters** on Friday that it appeared the consumer credit reporting agency hadn't applied patches for flaws discovered earlier this year.

On Wednesday company specified the flaw in a statement posted to its site and stressed it was continuing to work alongside law enforcement to investigate the incident.

<https://www.pinkconnect.com/equifax-confirms-march-struts-vulnerability-behind-breach/>



# PATCH MUCH?

## Equifax confirms march struts vulnerability behind breach

by Chris Brook for Threat Post  
Equifax said the culprit behind this summer's massive breach of 143 million Americans was indeed CVE-2017-5638, an Apache Struts vulnerability patched back in March.

September 14, 2017 , 4:00 pm

The bug was widely assumed by experts to be the "U.S. website application vulnerability" implicated by the company last Thursday. Reuters on Friday that it appeared the company had patched for flaws discovered earlier this year.

On Wednesday company specified the firm was continuing to work alongside law enforcement.

"Equifax has been intensely investigating the scope of the intrusion with the assistance of a leading, independent cybersecurity firm to determine what information was accessed and who has been impacted. We know that criminals exploited a U.S. website application vulnerability. The vulnerability was Apache Struts CVE-2017-5638. We continue to work with law enforcement as part of our criminal investigation, and have shared indicators of compromise with law enforcement."

<https://www.pinkconnect.com/equifax-confirms-march-struts-vulnerability-behind-breach/>



# PATCH MUCH?

## Equifax confirms march struts vulnerability behind breach

by Chris Brook for Threat Post  
Equifax said the culprit behind this summer's massive breach of 143 million Americans was indeed CVE-2017-5638, an Apache Struts vulnerability patched back in March.

The bug was widely assumed by experts to be the "U.S. website application vulnerability" implicated by the company last Thursday. Reuters on Friday that it appeared the company had patches for flaws discovered earlier this year.

On Wednesday company specified the firm was continuing to work alongside law enforcement.

# CVE-2017-5638

"Equifax has been intensely investigating the scope of the intrusion with the assistance of a leading, independent cybersecurity firm to determine what information was accessed and who has been impacted. We know that criminals exploited a U.S. website application vulnerability. The vulnerability was Apache Struts CVE-2017-5638. We continue to work with law enforcement as part of our criminal investigation, and have shared indicators of compromise with law enforcement."

<https://www.pinkconnect.com/equifax-confirms-march-struts-vulnerability-behind-breach/>

# CVE-2017-5638

## Current Description

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

Source: MITRE    Last Modified: 09/22/2017    [+View Analysis Description](#)

## Impact

### CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical  
Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H (legend)  
Impact Score: 6.0  
Exploitability Score: 3.9

# CVE-2017-5638

## Current Description

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

Source: MITRE   Last Modified: 09/22/2017   [+View Analysis Description](#)

## Impact

### CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical

Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H (legend)

Impact Score: 6.0

Exploitability Score: 3.9

# Impact

## CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical

# CVE-2017-5638

## Current Description

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

Source: MITRE Last Modified: 09/22/2017 [+View Analysis Description](#)

## Impact

### CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical

Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H (legend)

Impact Score: 6.0

Exploitability Score: 3.9

# Impact

## CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical



# CVE-2017-5638

## Current Description

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

Source: MITRE Last Modified: 09/22/2017 [+View Analysis Description](#)

## Impact

### CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical

Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H (legend)

Impact Score: 6.0

Exploitability Score: 3.9

# Impact

## CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical

# Reported March 2017

# CVE-2017-5638

## Current Description

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

Source: MITRE Last Modified: 09/22/2017 [+View Analysis Description](#)

## Impact

### CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical

Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H (legend)

Impact Score: 6.0

Exploitability Score: 3.9

## Impact

### CVSS Severity (version 3.0):

CVSS v3 Base Score: 10.0 Critical

Reported March 2017

Patched in struts 2.3.32 / 2.5.10.1 on 7th March

# EQUIFAX TIMELINE

**sources: <https://krebsonsecurity.com/2017/09/breach-at-equifax-may-impact-143m-americans/>  
<https://www.usatoday.com/story/tech/2017/09/26/timeline-events-surrounding-equifax-data-breach/703691001/>**

## EQUIFAX TIMELINE

**Equifax breach happened between mid-May and July**

**sources: <https://krebsonsecurity.com/2017/09/breach-at-equifax-may-impact-143m-americans/>  
<https://www.usatoday.com/story/tech/2017/09/26/timeline-events-surrounding-equifax-data-breach/703691001/>**



## EQUIFAX TIMELINE

**Equifax breach happened between mid-May and July**

**Equifax spotted it on July 29th**

**sources: <https://krebsonsecurity.com/2017/09/breach-at-equifax-may-impact-143m-americans/>  
<https://www.usatoday.com/story/tech/2017/09/26/timeline-events-surrounding-equifax-data-breach/703691001/>**

## EQUIFAX TIMELINE

**Equifax breach happened between mid-May and July**

**Equifax spotted it on July 29th**

**Reported on September 7th**

**sources: <https://krebsonsecurity.com/2017/09/breach-at-equifax-may-impact-143m-americans/>  
<https://www.usatoday.com/story/tech/2017/09/26/timeline-events-surrounding-equifax-data-breach/703691001/>**

## EQUIFAX TIMELINE

**Equifax breach happened between mid-May and July**

**Equifax spotted it on July 29th**

**Reported on September 7th**

**At the time the breach was discovered, the patch had been out for at least 2 months, and perhaps as long as 4 months**

**sources: <https://krebsonsecurity.com/2017/09/breach-at-equifax-may-impact-143m-americans/>  
<https://www.usatoday.com/story/tech/2017/09/26/timeline-events-surrounding-equifax-data-breach/703691001/>**

**2 to 4 months**



**Hands up if you *\*know\** you patch all your systems every 2-4 months?**







# PATCHING MADNESS!

Underlying Hardware

# PATCHING MADNESS!



Operating System

Underlying Hardware



# PATCHING MADNESS!

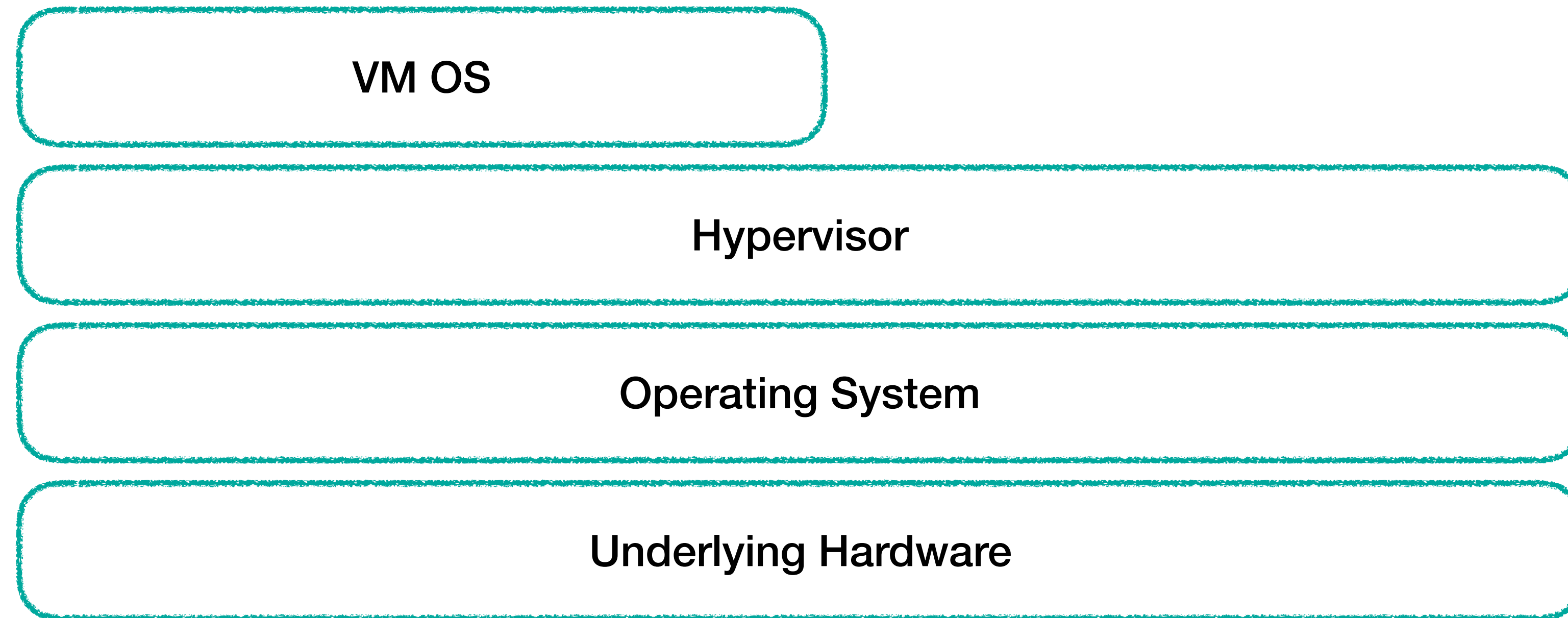


Hypervisor

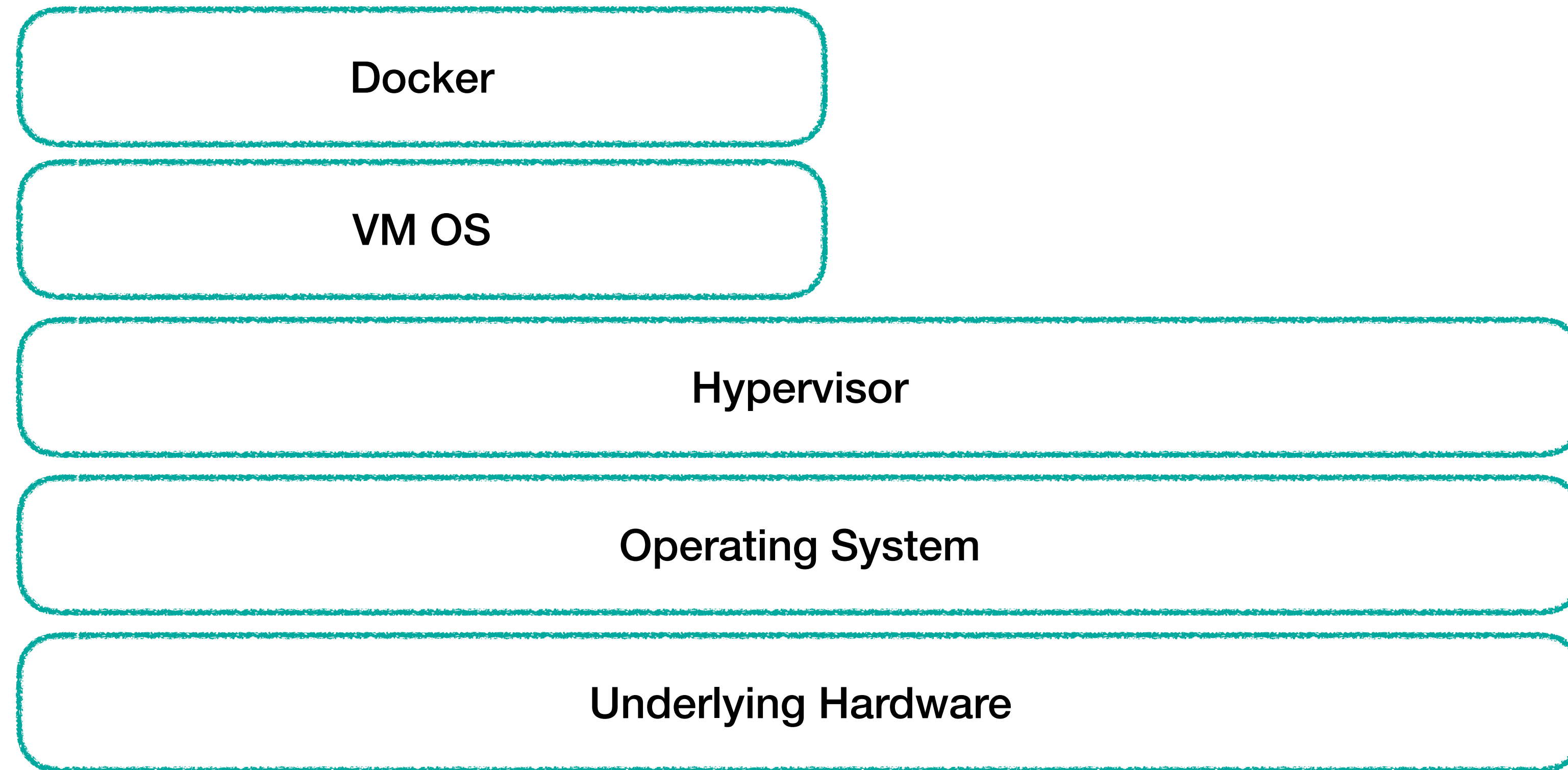
Operating System

Underlying Hardware

# PATCHING MADNESS!

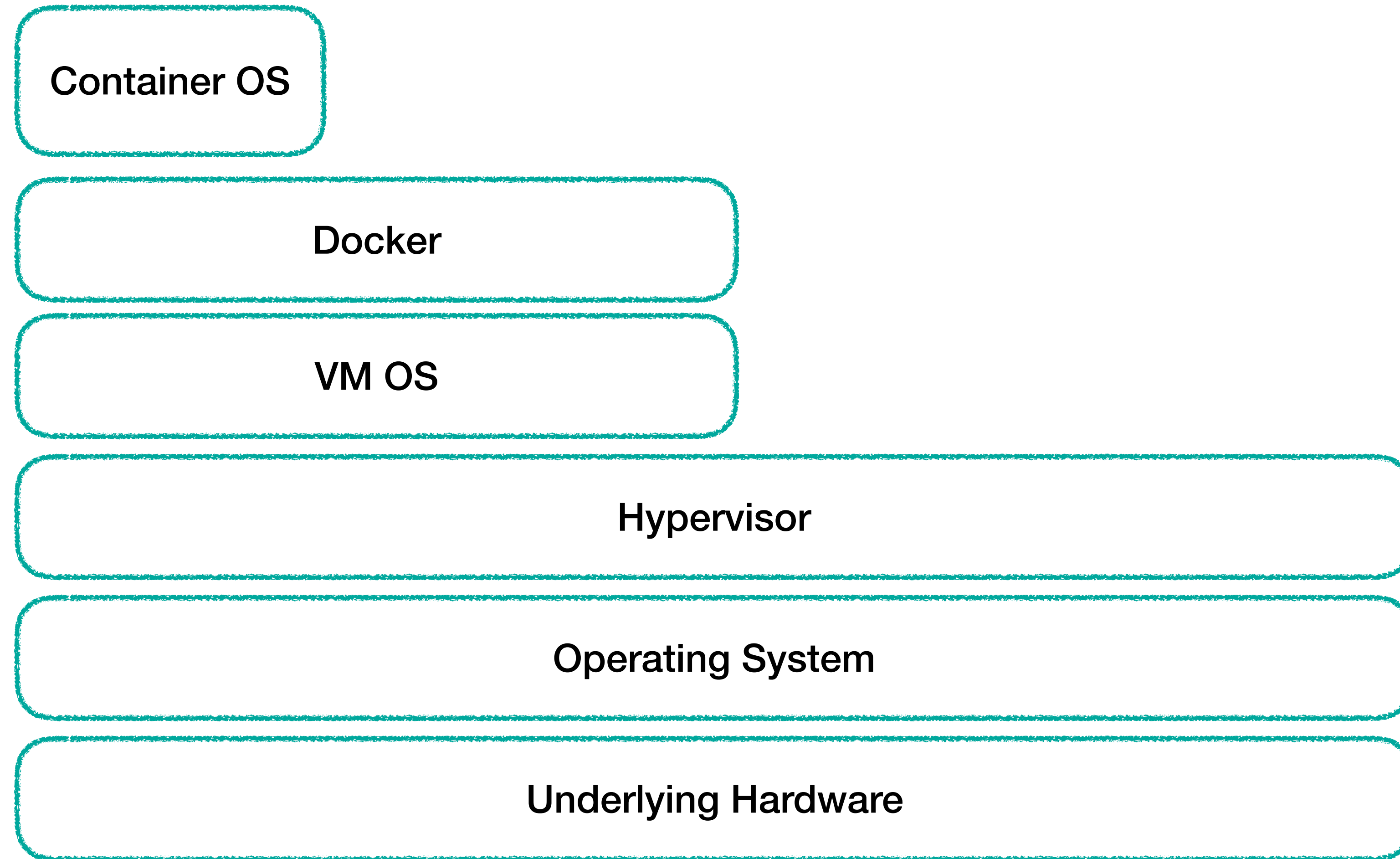


# PATCHING MADNESS!

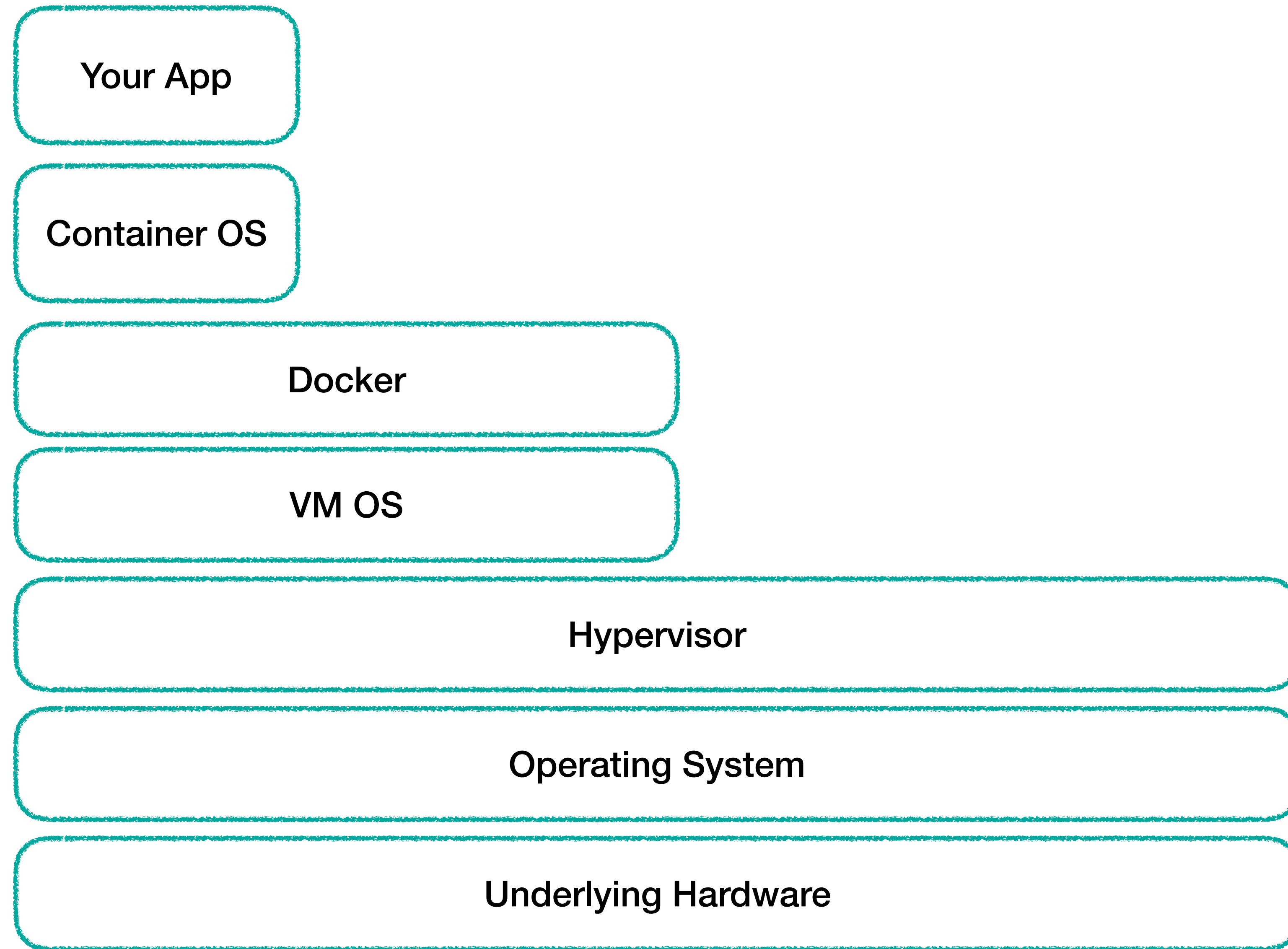




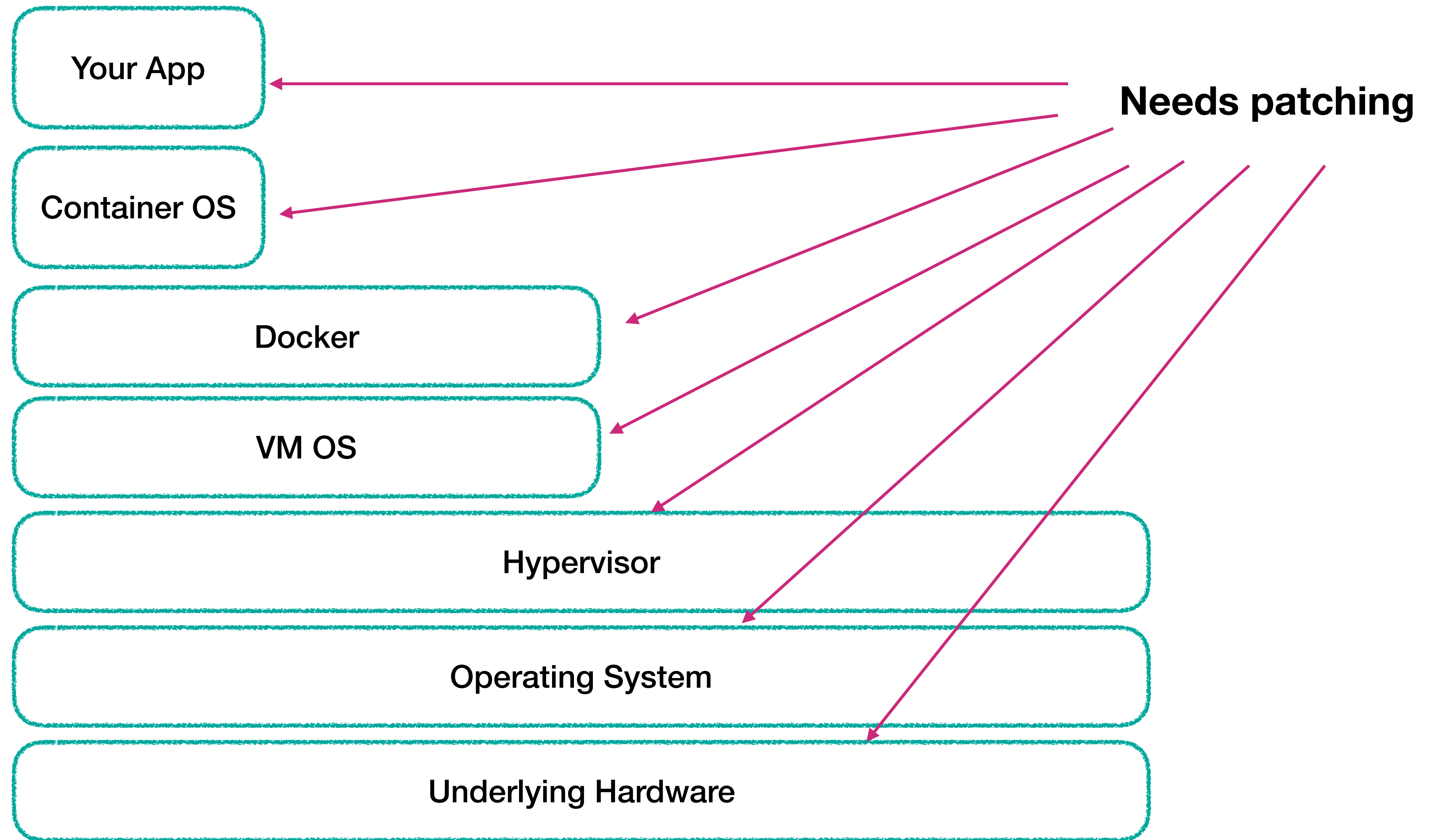
# PATCHING MADNESS!



# PATCHING MADNESS!



# PATCHING MADNESS!

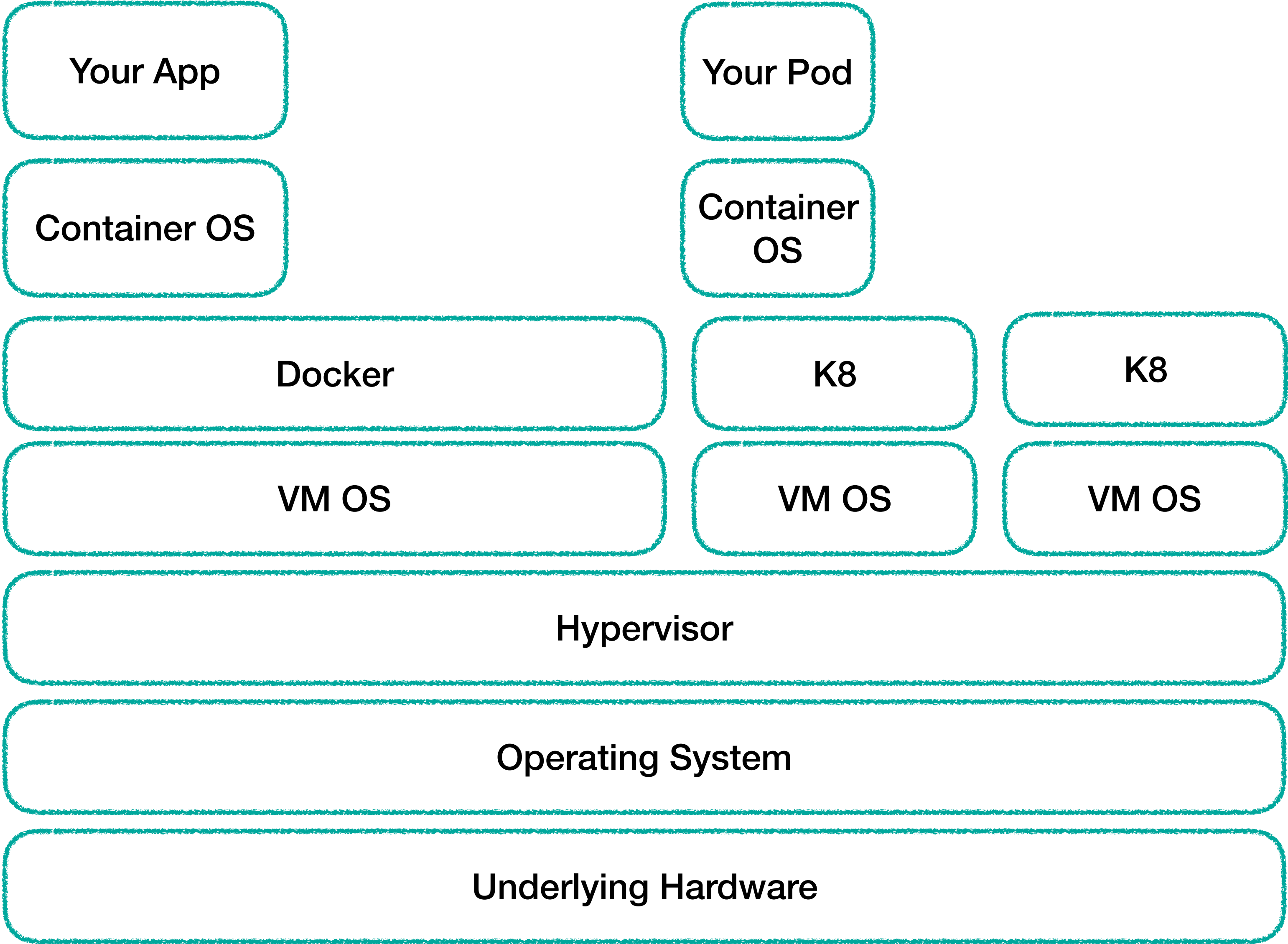




**So, how many of you are still sure you  
apply every patch within 2-4 months?**

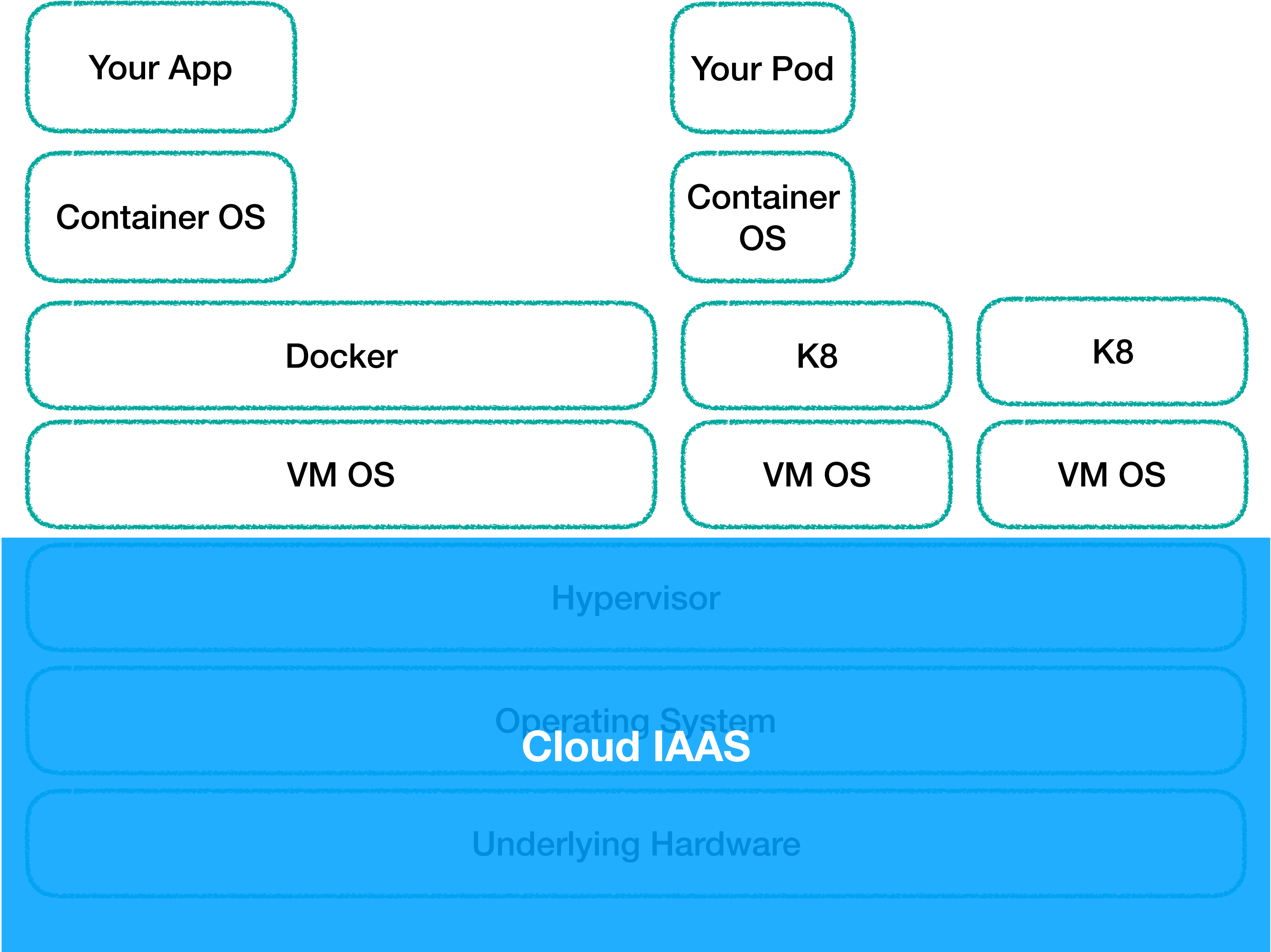
**So what can you do about this?**

# BETTER ON THE CLOUD?

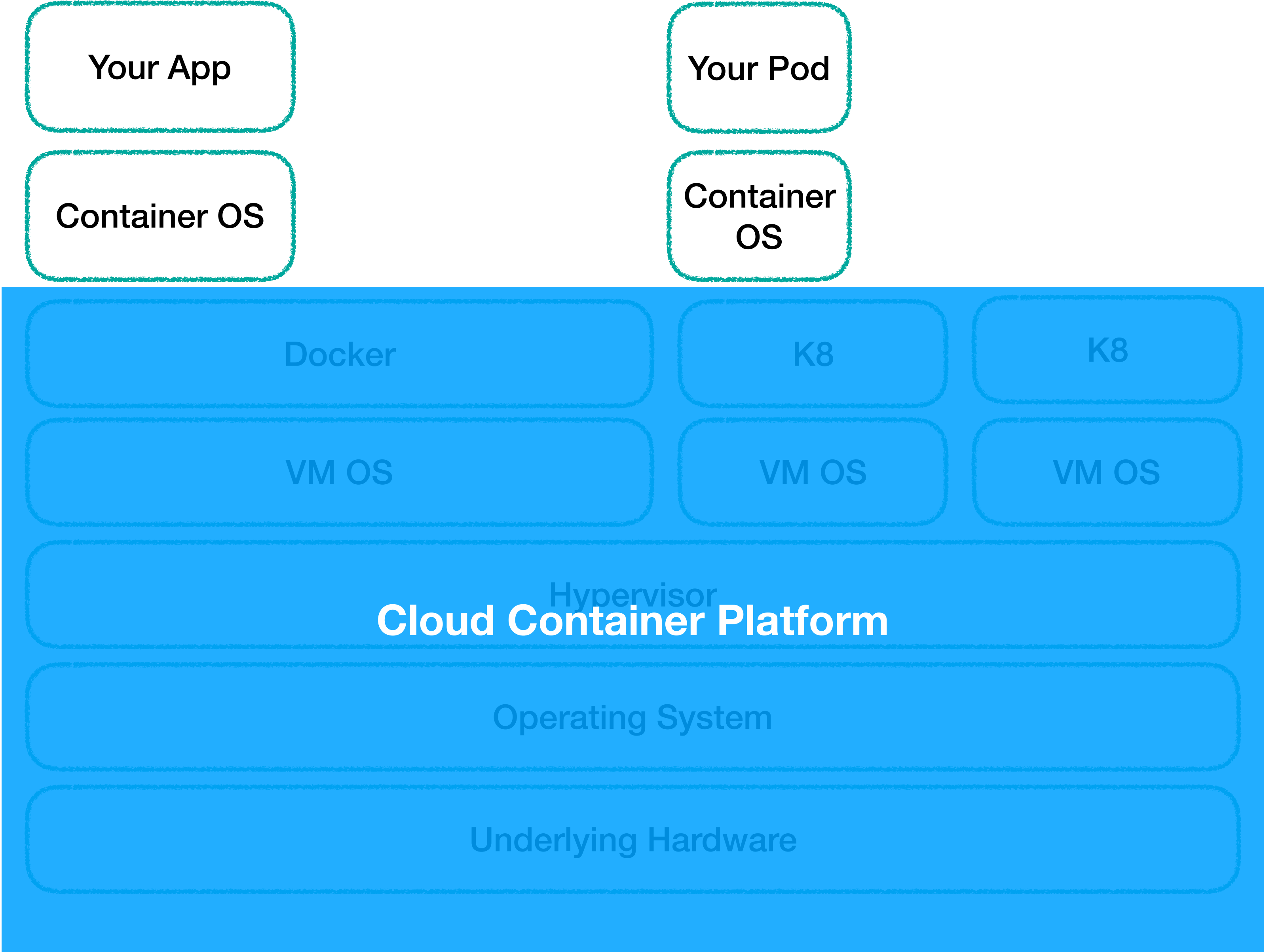




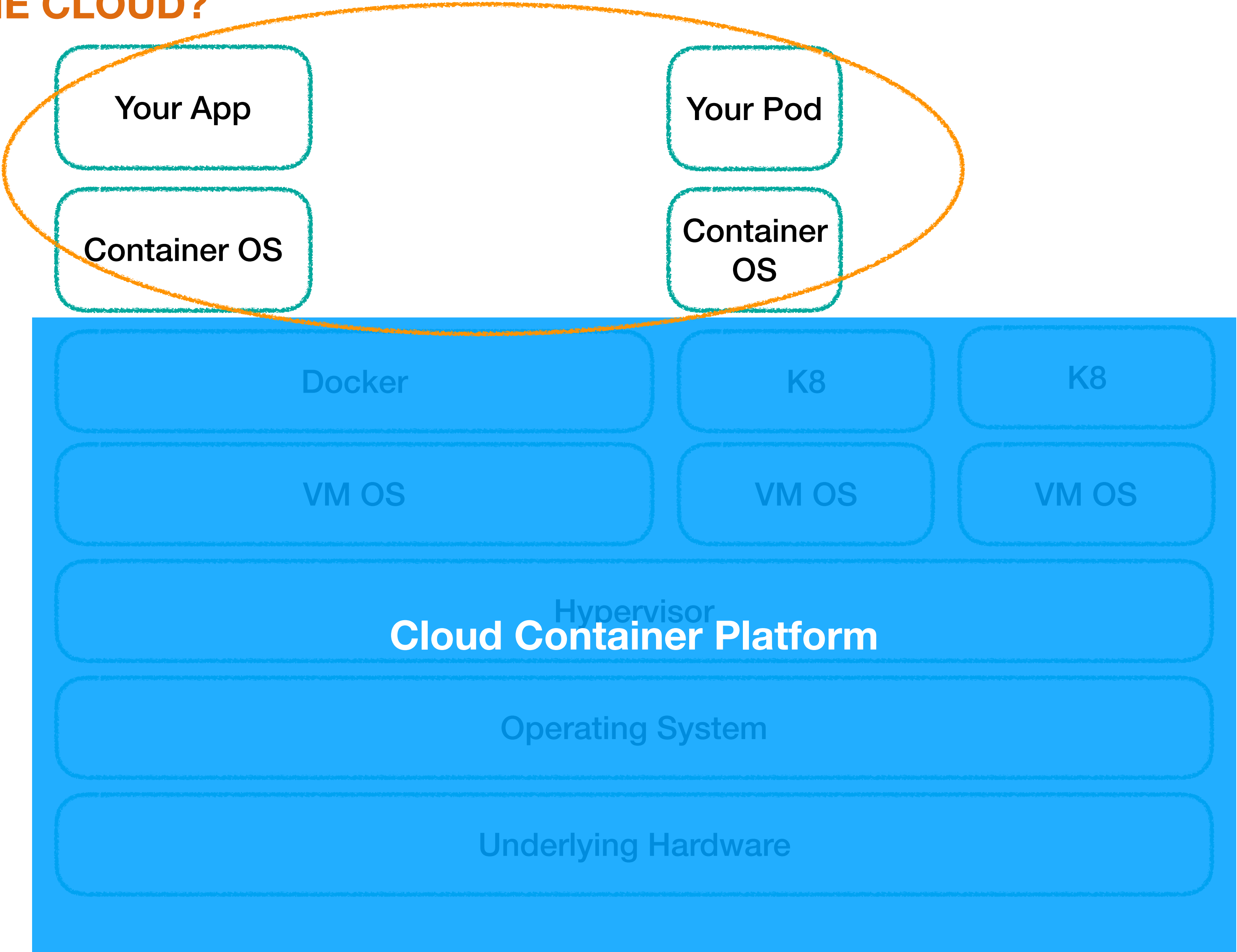
# BETTER ON THE CLOUD?



# BETTER ON THE CLOUD?

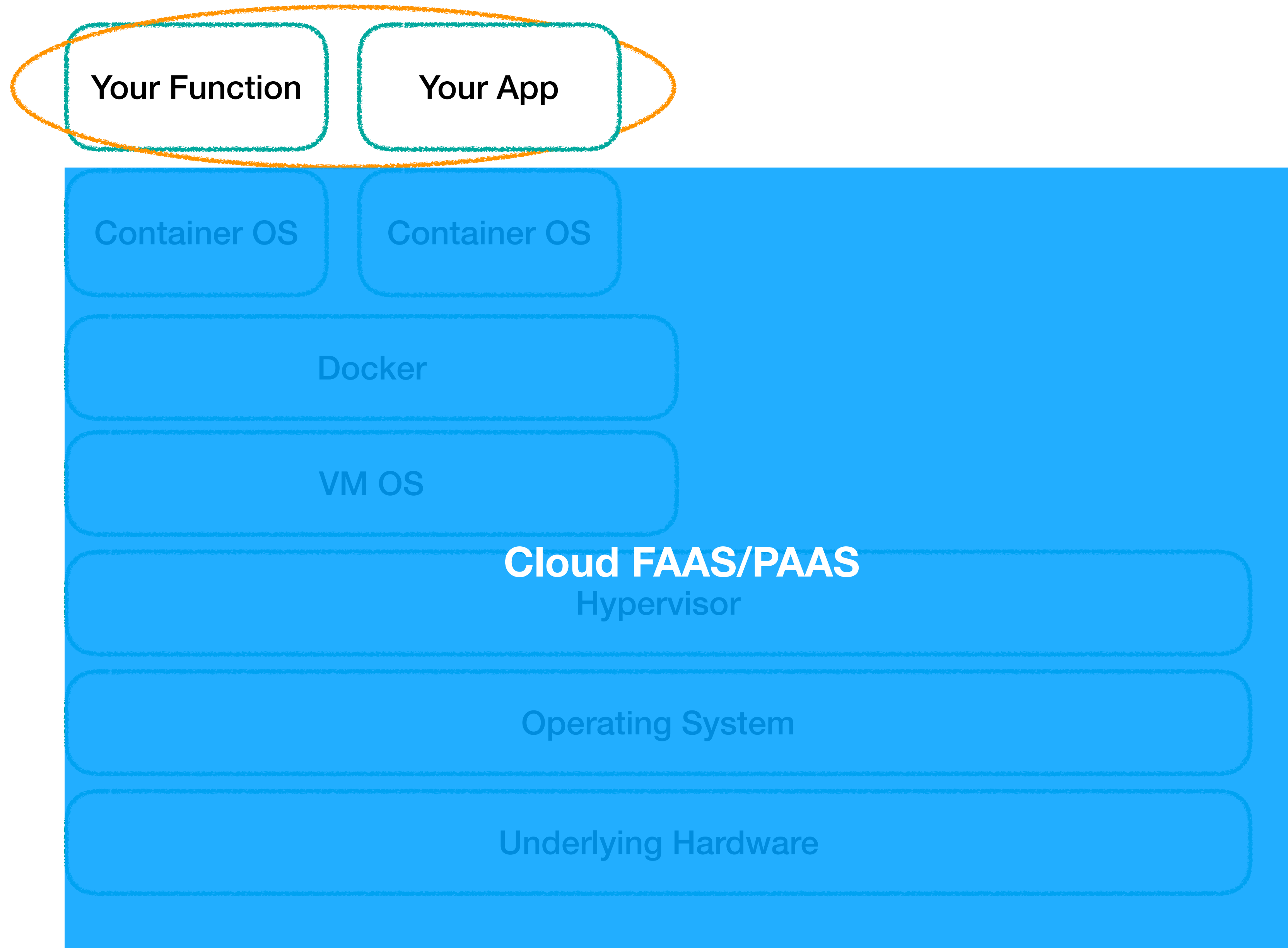


BETTER ON THE CLOUD?





## BETTER WITH FAAS?




# CONTAINER SCANNING

README.md

## Clair

build passing container ready go report A+ godoc reference freenode #clair

**Note:** The `master` branch may be in an *unstable or even broken state* during development. Please use [releases](#) instead of the `master` branch in order to get stable binaries.



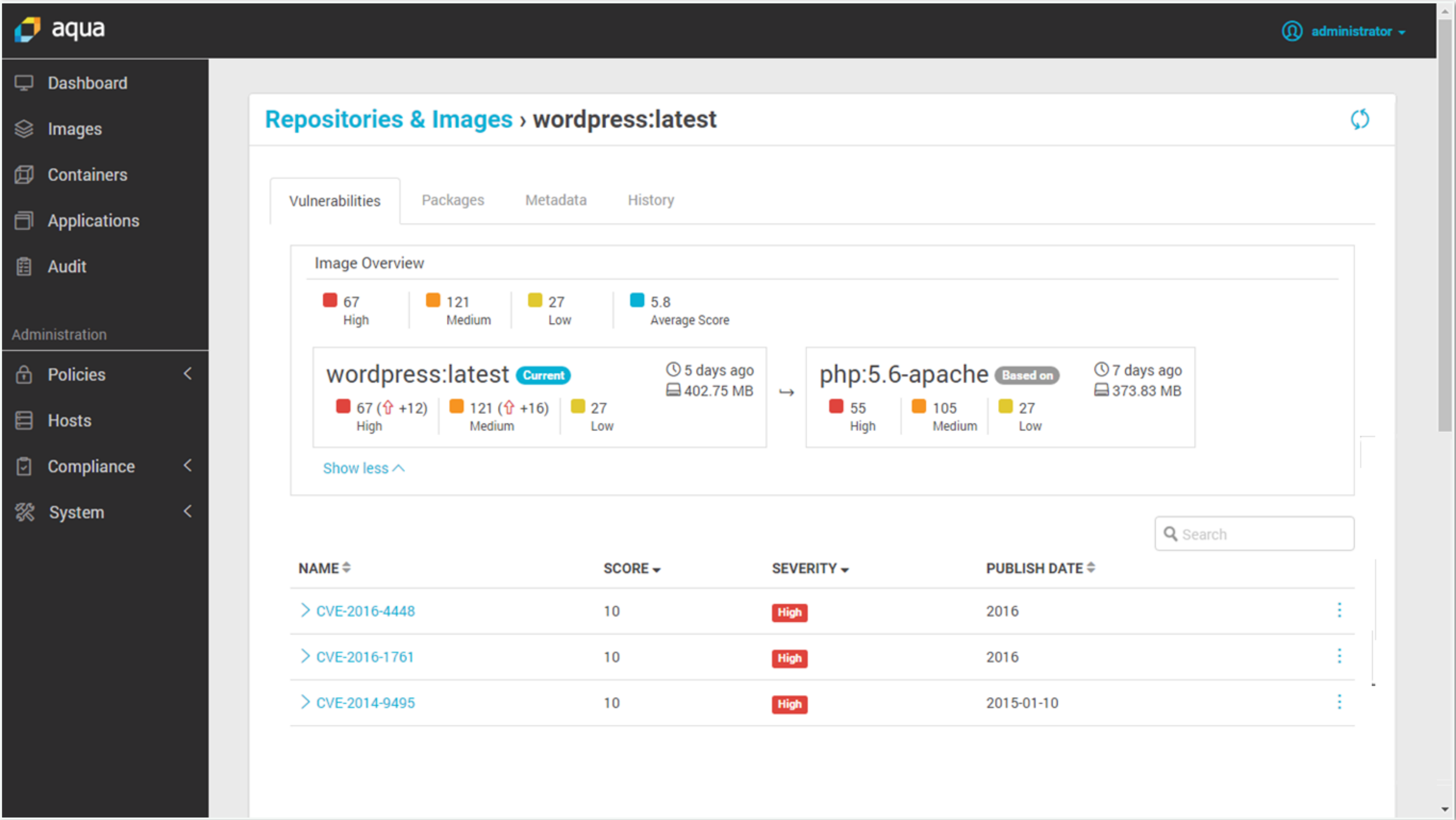
Clair is an open source project for the [static analysis](#) of vulnerabilities in application containers (currently including [appc](#) and [docker](#)).

1. In regular intervals, Clair ingests vulnerability metadata from a configured set of sources and stores it in the database.
2. Clients use the Clair API to index their container images; this creates a list of *features* present in the image and stores them in the database.
3. Clients use the Clair API to query the database for vulnerabilities of a particular image; correlating vulnerabilities and features is done for each request, avoiding the need to rescan images.
4. When updates to vulnerability metadata occur, a notification can be sent to alert systems that a change has occurred.

Our goal is to enable a more transparent view of the security of container-based infrastructure. Thus, the project was named `Clair` after the French term which translates to *clear, bright, transparent*.

<https://github.com/coreos/clair>

# CONTAINER SCANNING (CONT)



<https://www.aquasec.com>



# MONITOR OUTDATED DEPENDENCIES

**snyk** Test Vulnerability DB Docs Blog Features Partners Pricing Log in: [Sign up](#)

**Snyk continuously finds and fixes vulnerabilities in your dependencies.**  
Protect and monitor your JavaScript, Ruby and Java apps

**Source code protection**  
Find vulnerabilities in your **GitHub** repositories and remediate risks with updates and patches. Add Snyk to your CI/CD process with support for **Jenkins**, **Circle CI**, **Travis** and more.

[Quick start with GitHub](#)

**GitHub** **Bitbucket** **Travis CI** **Jenkins**

**New! Serverless & PaaS monitoring**  
Continuously monitor your runtime apps. Get Snyk security alerts and deploy critical updates. Support for **Heroku**, **AWS Lambda** and more.

[Sign up for a free account](#)

**HEROKU** **amazon** web services

**83% of Snyk users found security issues in their dependencies**  
We can help you find, fix, and prevent vulnerabilities:

- **Automatically test** your applications dependencies
- **Fix** security risks with upgrades and patches
- **Prevent** you from adding vulnerable dependencies
- **Stay alert** about new vulnerabilities

[Find out how](#)

**snyk** Canidae Ltd Vulnerability DB Docs My account

Dashboard Projects Settings

Search projects [Add projects](#)

☐ GitHub ☐ CircleCI ☐ Bitbucket Languages Sort

**canidae/pug** 5 14 14 [View report and fix](#) [New](#) [Test weekly](#) Tested 1 hour ago

4 14 14 [View report and fix](#) [Test weekly](#) Tested 1 hour ago

**flat-coated-retriever** 0 0 0 [View report](#) [Test weekly](#) Tested 3 days ago

**canidae/pyrenean-shepherd** 1 14 14 [View report and fix](#) [New](#) [Test weekly](#) Tested 5 days ago


**canidae/anatolian-shepherd** 2 14 14 [View report and fix](#) [Test weekly](#) Tested 1 week ago

**canidae/saint-bernard** 0 [View report](#) [Test weekly](#) Tested 1 week ago


<https://snyk.io/>


# AUTOMATICALLY PATCH APP DEPENDENCIES


## [Snyk Update] New fixes for 25 vulnerable dependency paths #1


 **Open**

 snyk-bot wants to merge 1 commit into `master` from `snyk-fix-836b436e`



 Conversation 0

 Commits 1

 Files changed 2



snyk-bot commented an hour ago

First-time contributor +  

This project has vulnerabilities that could not be fixed, or were patched when no upgrade was available. Good news, new upgrades or patches have now been published! This pull request fixes vulnerable dependencies you couldn't previously address.


The PR includes:

- Changes to `package.json` to upgrade the vulnerable dependencies to a fixed version.

<https://snyk.io/>



# DO SOME THREAT MODELLING




## Life in the Digital Crosshairs

Experience the Untold Story

[Learn more →](#)

### What is the Security Development Lifecycle ?



The Security Development Lifecycle (SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost

Training

Requirements

Design

Implementation

Verification

Release

Response

*Click to select a phase*

### Design Phase

#### SDL Practice #5: Establish Design Requirements

Considering security and privacy concerns early helps minimize the risk of schedule disruptions and reduce a project's expense.

SDL Practice #6: Attack Surface Analysis / Detection

### Assess your security

Discover ways to improve your security practices.

[Get Started](#)

### Tools

[Attack Surface Analyzer 1.0](#)  
Understand your attack surface before & after new apps are deployed.

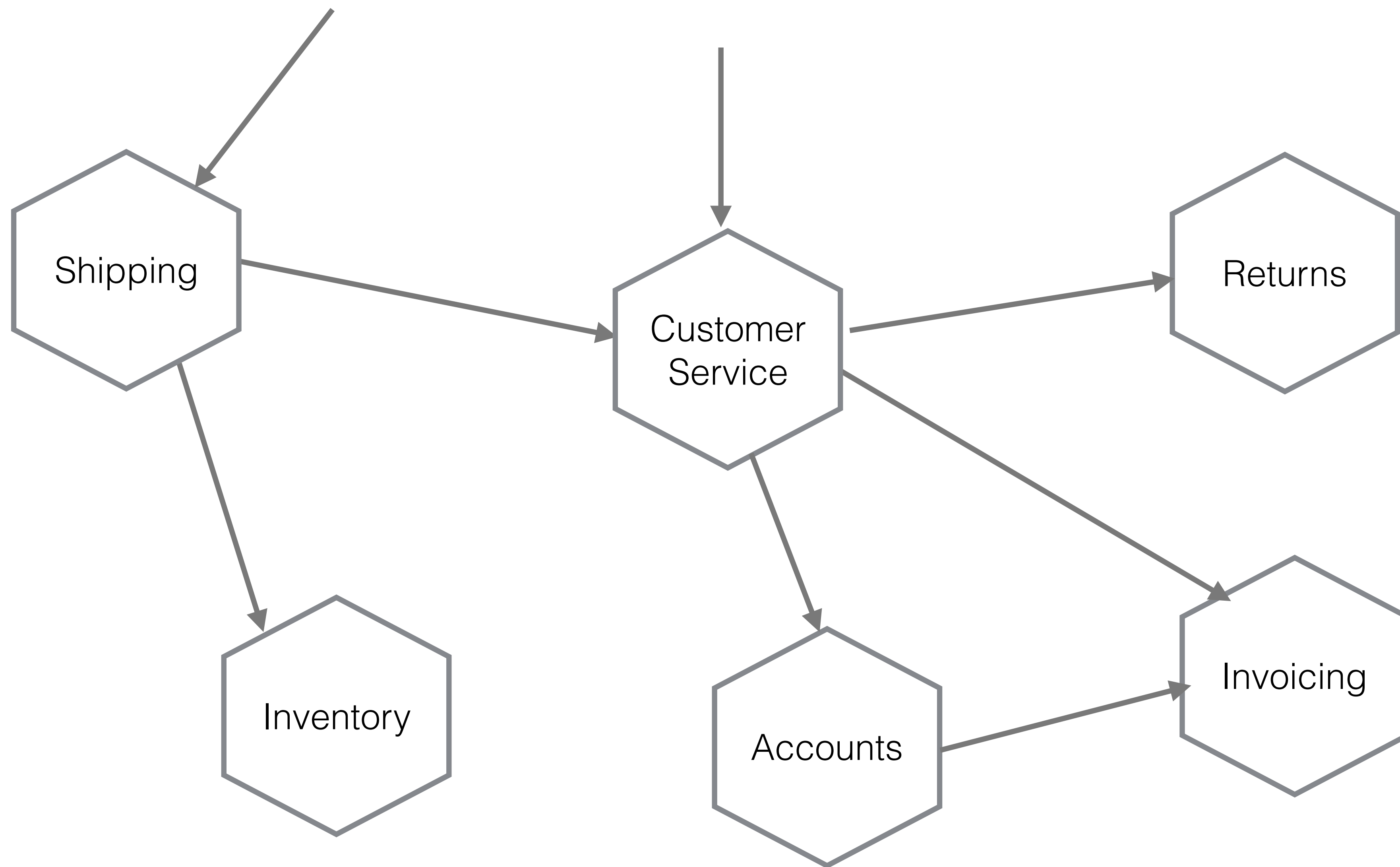
[Microsoft Threat Modeling Tool 2014](#)  
A tool to help engineers find and address system security issues.

[MiniFuzz basic file fuzzing tool](#)  
A simple fuzzer designed to ease adoption of fuzz testing.

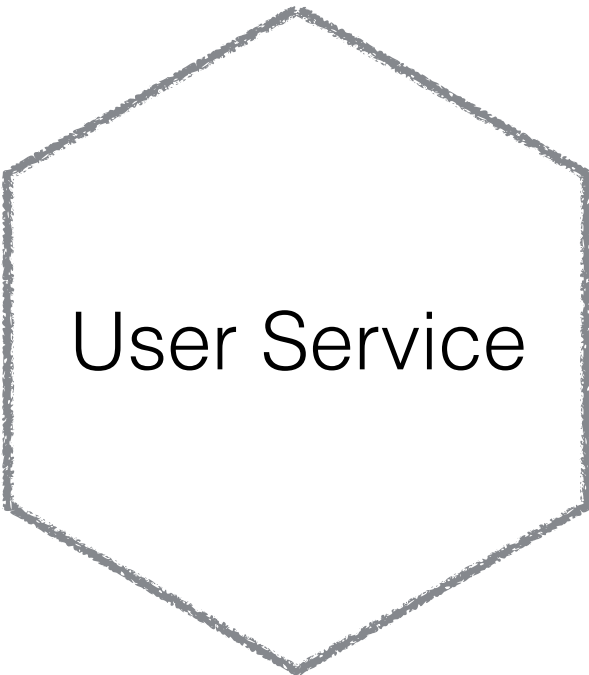
[Regular expression file fuzzing tool](#)  
A tool to test for potential denial of service vulnerabilities.

<https://www.microsoft.com/en-us/sdl/>



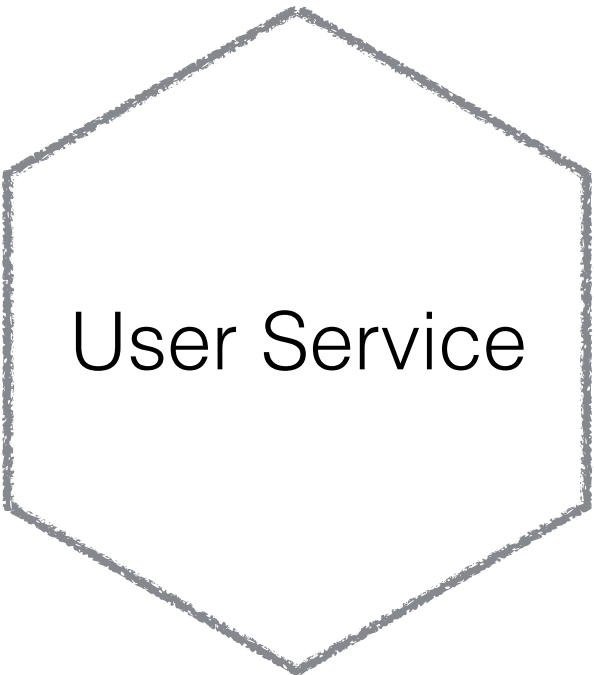


# MUSIC CORP 2018

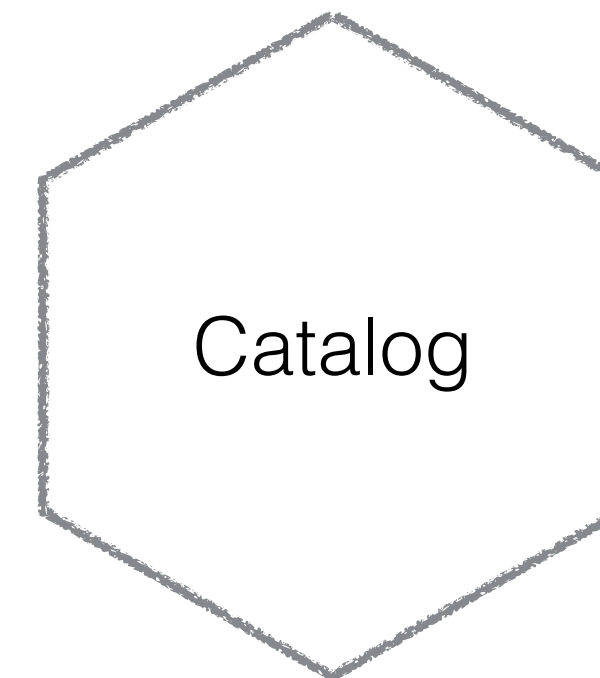
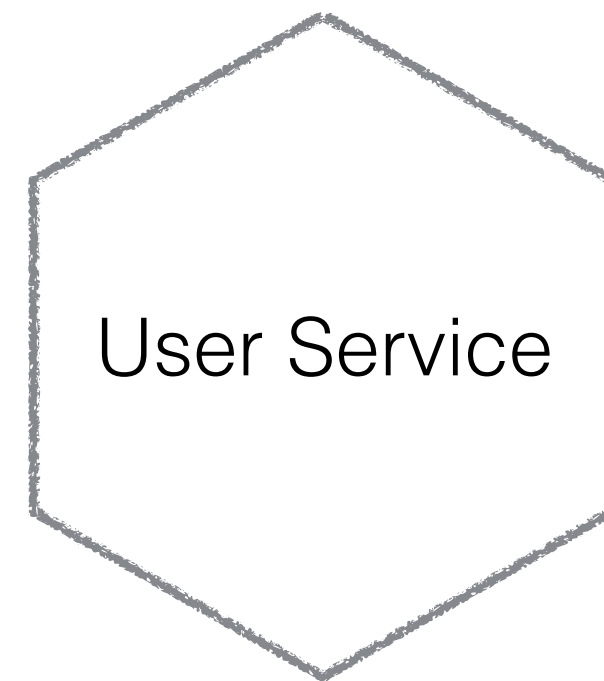




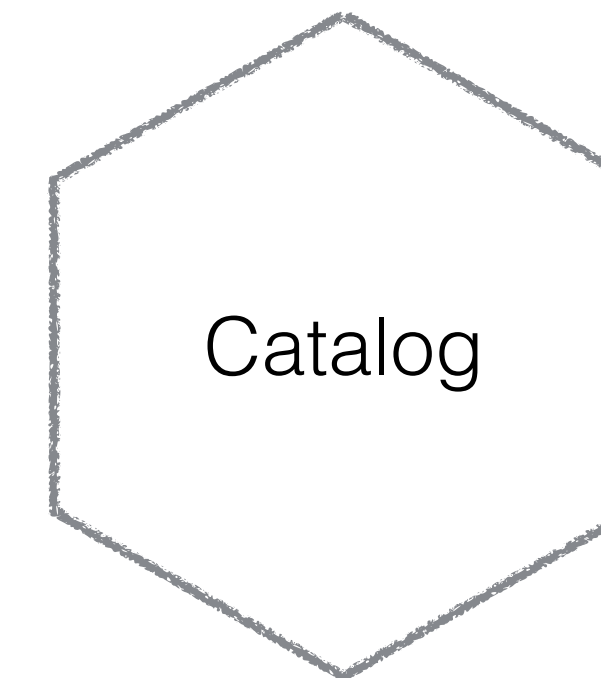
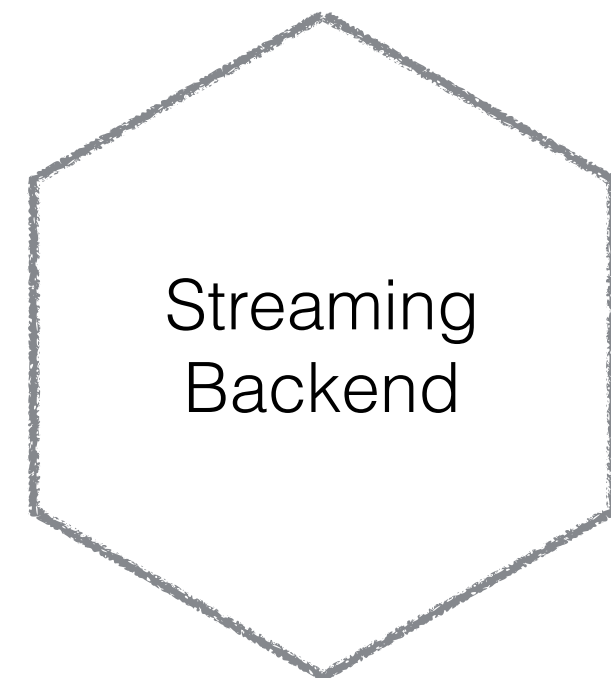
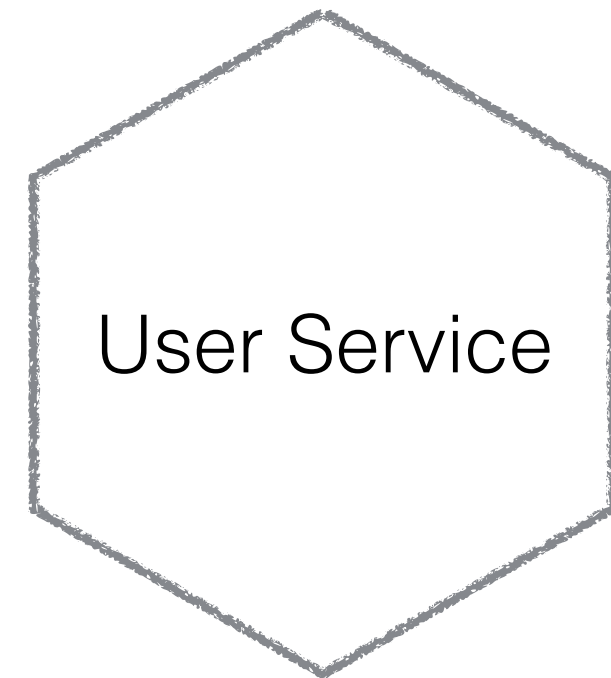
# MUSIC CORP 2018



# MUSIC CORP 2018

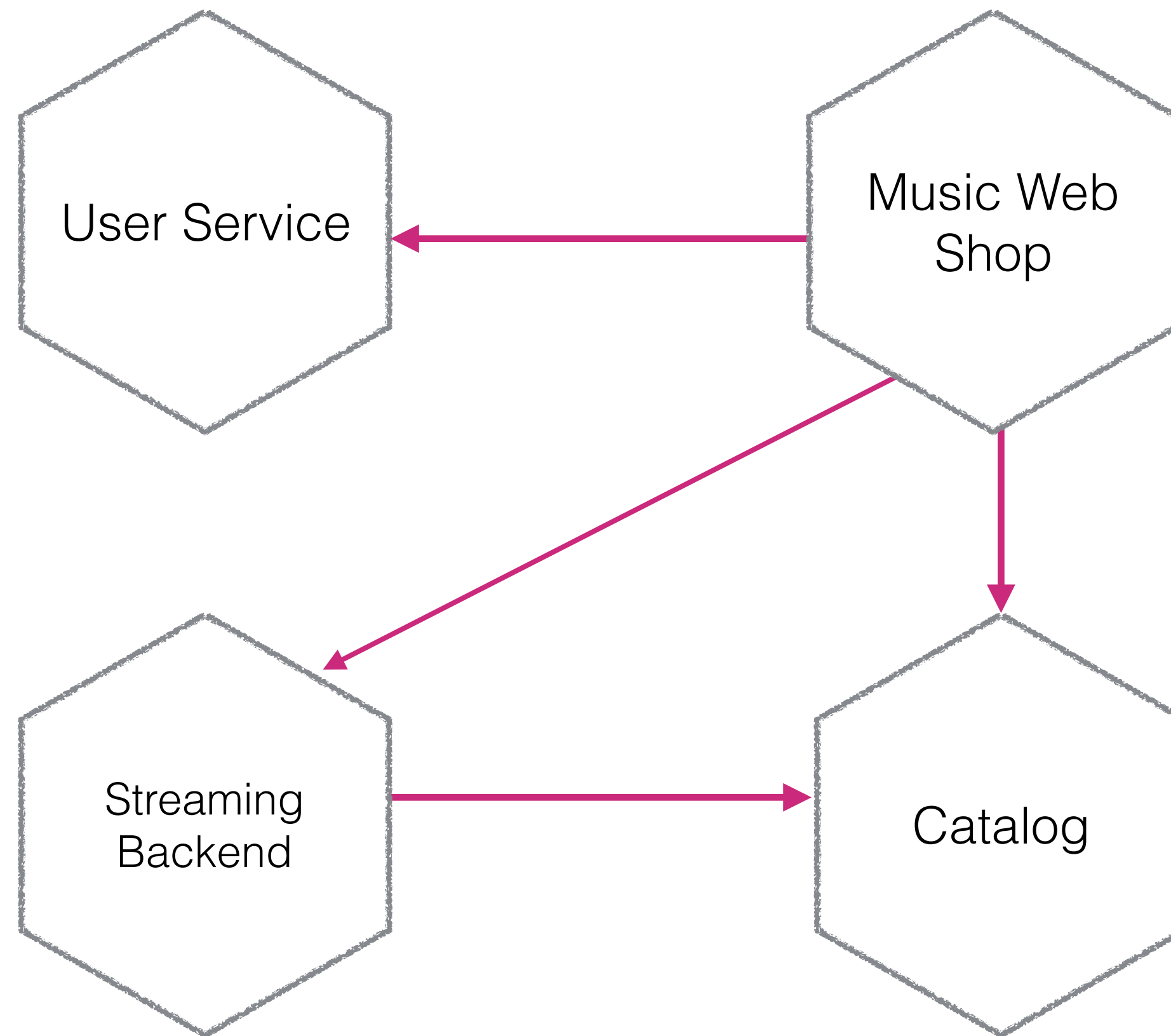


# MUSIC CORP 2018

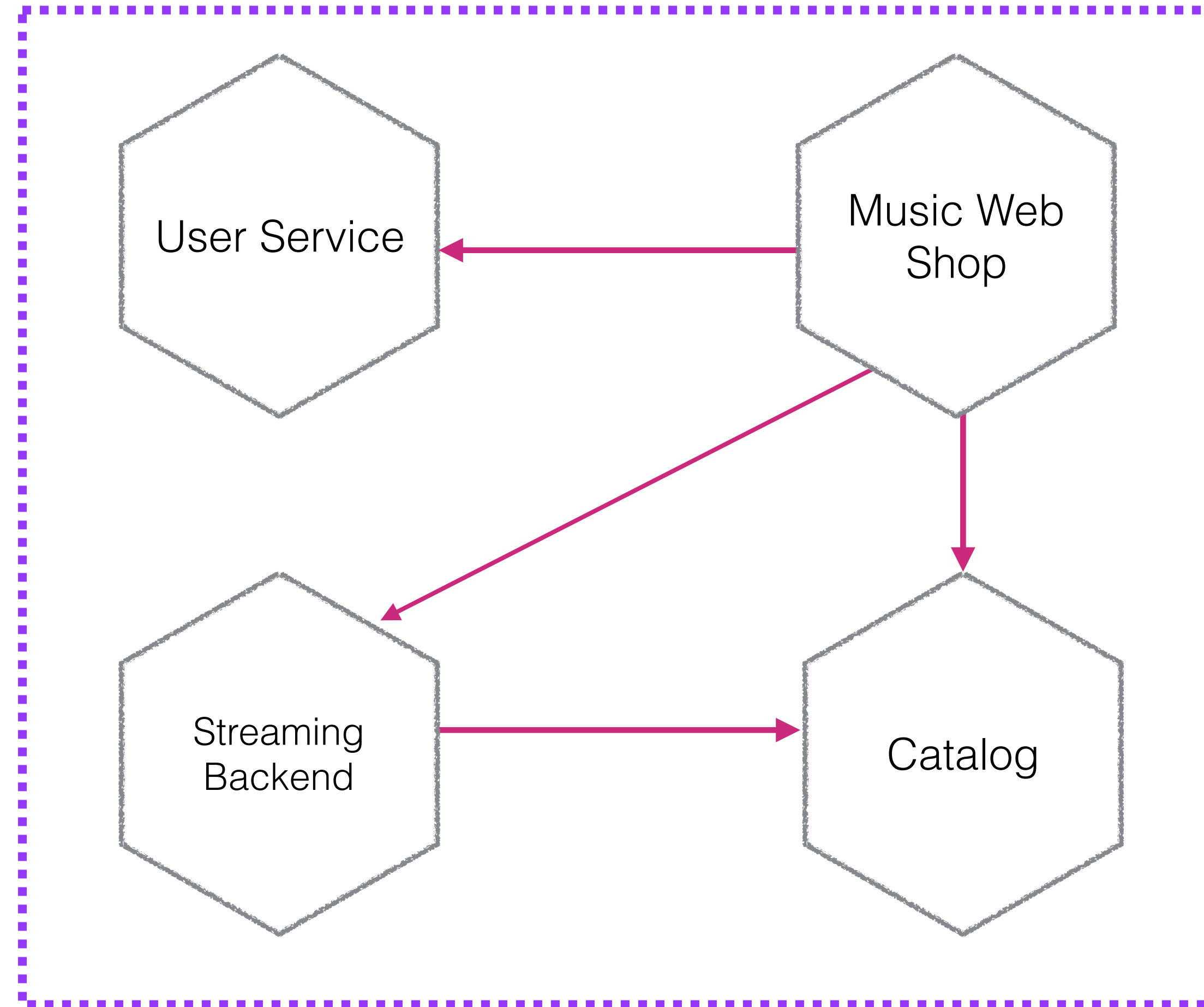




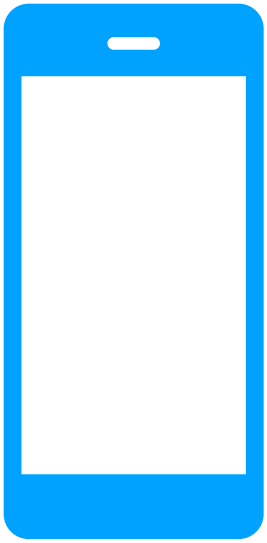
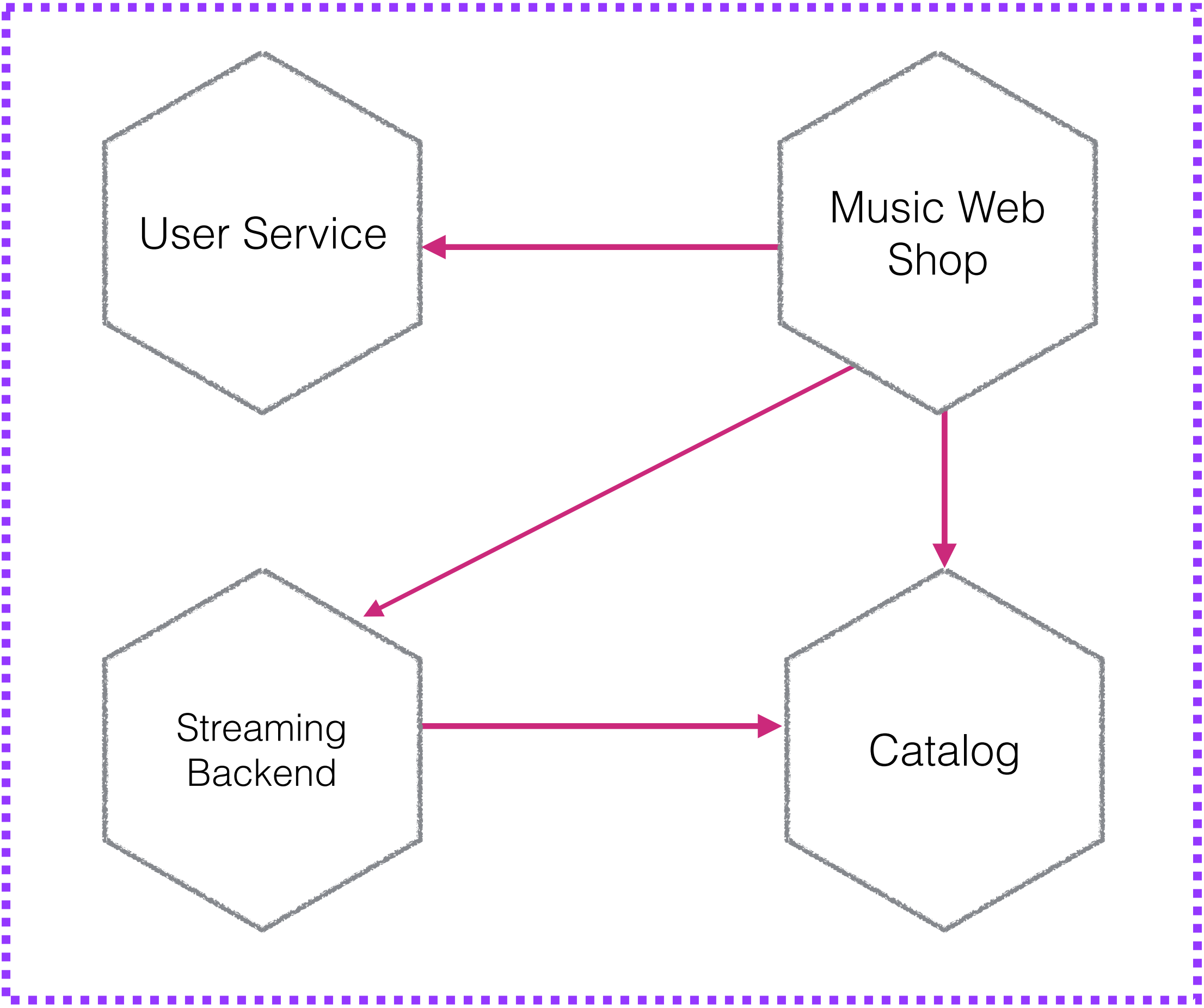
# MUSIC CORP 2018



# MUSIC CORP 2018

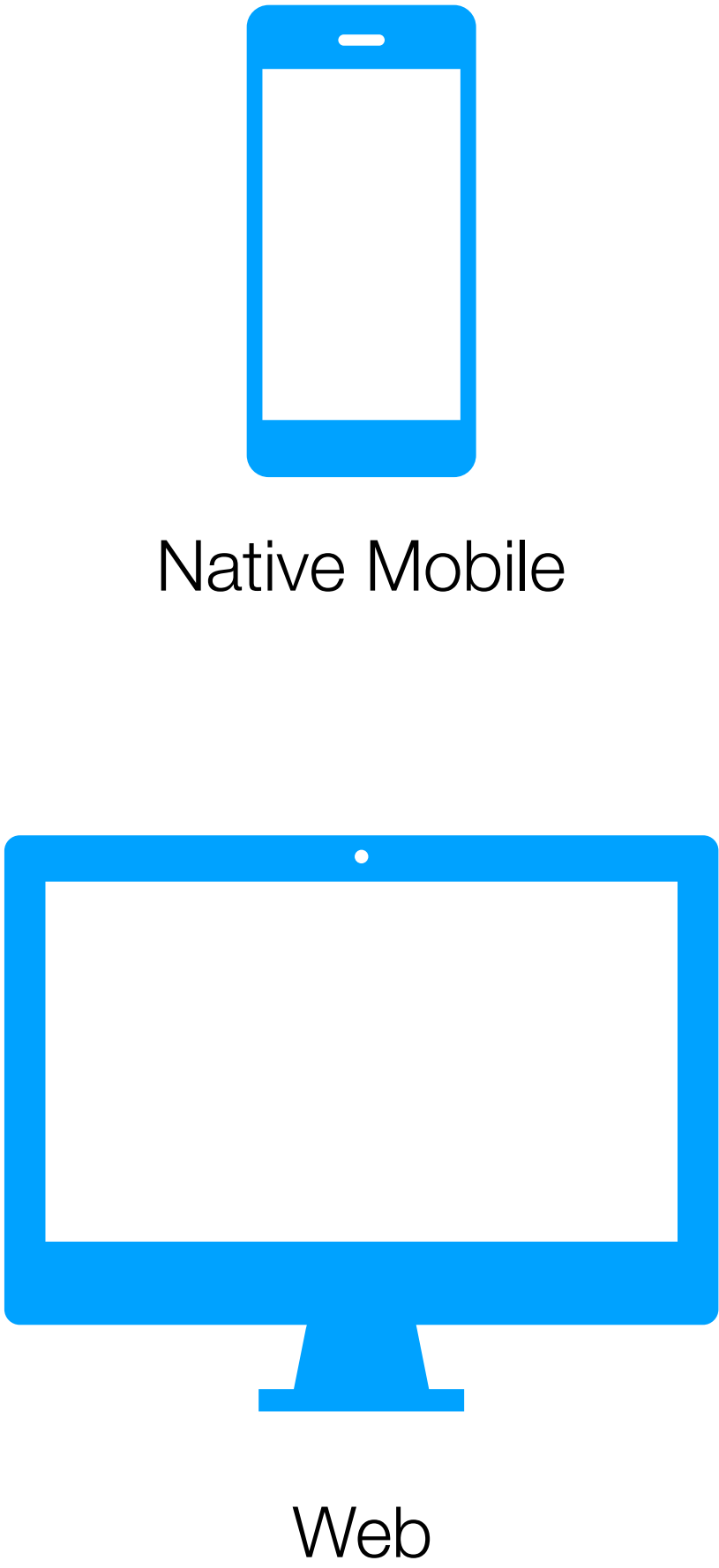
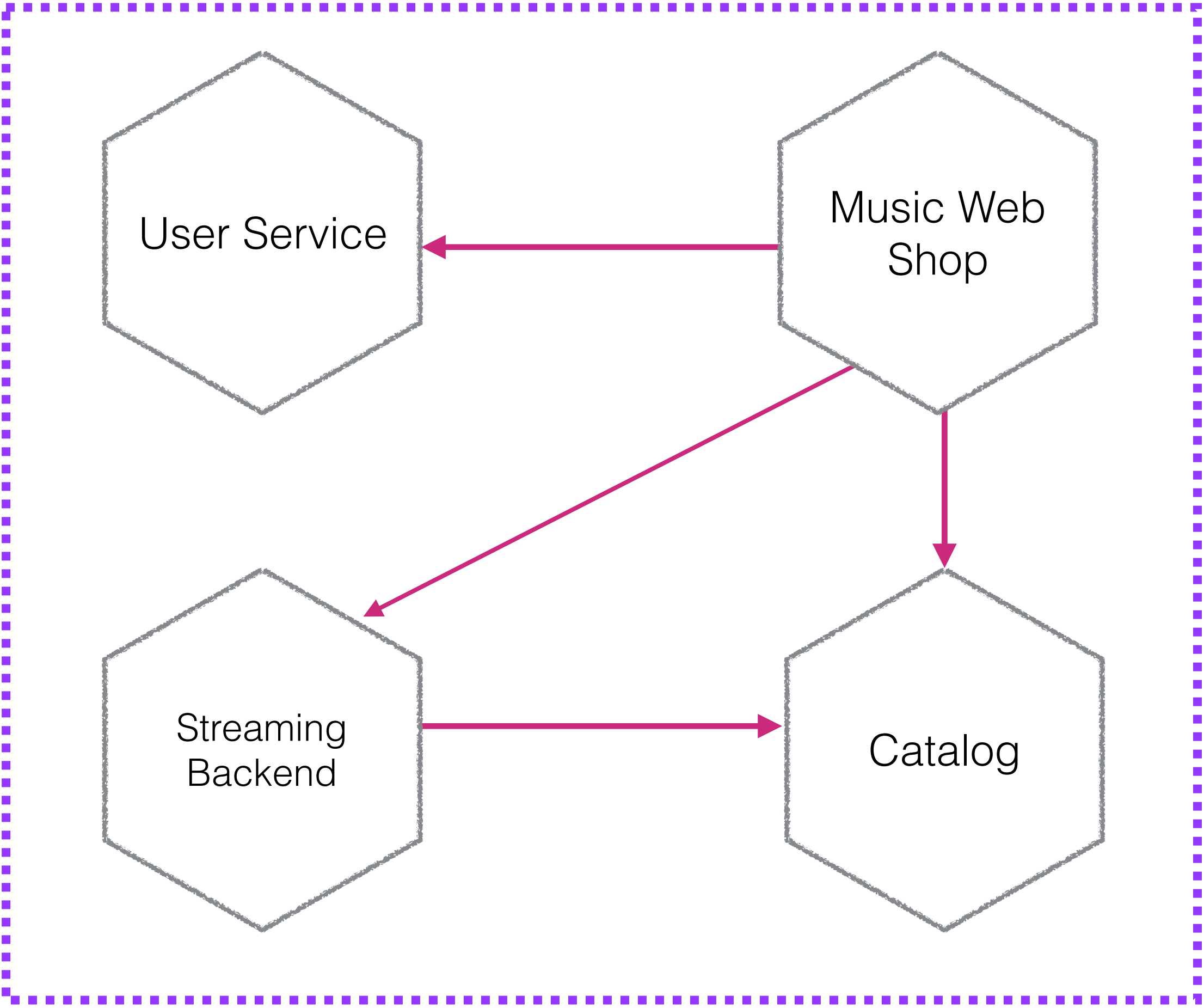


# MUSIC CORP 2018



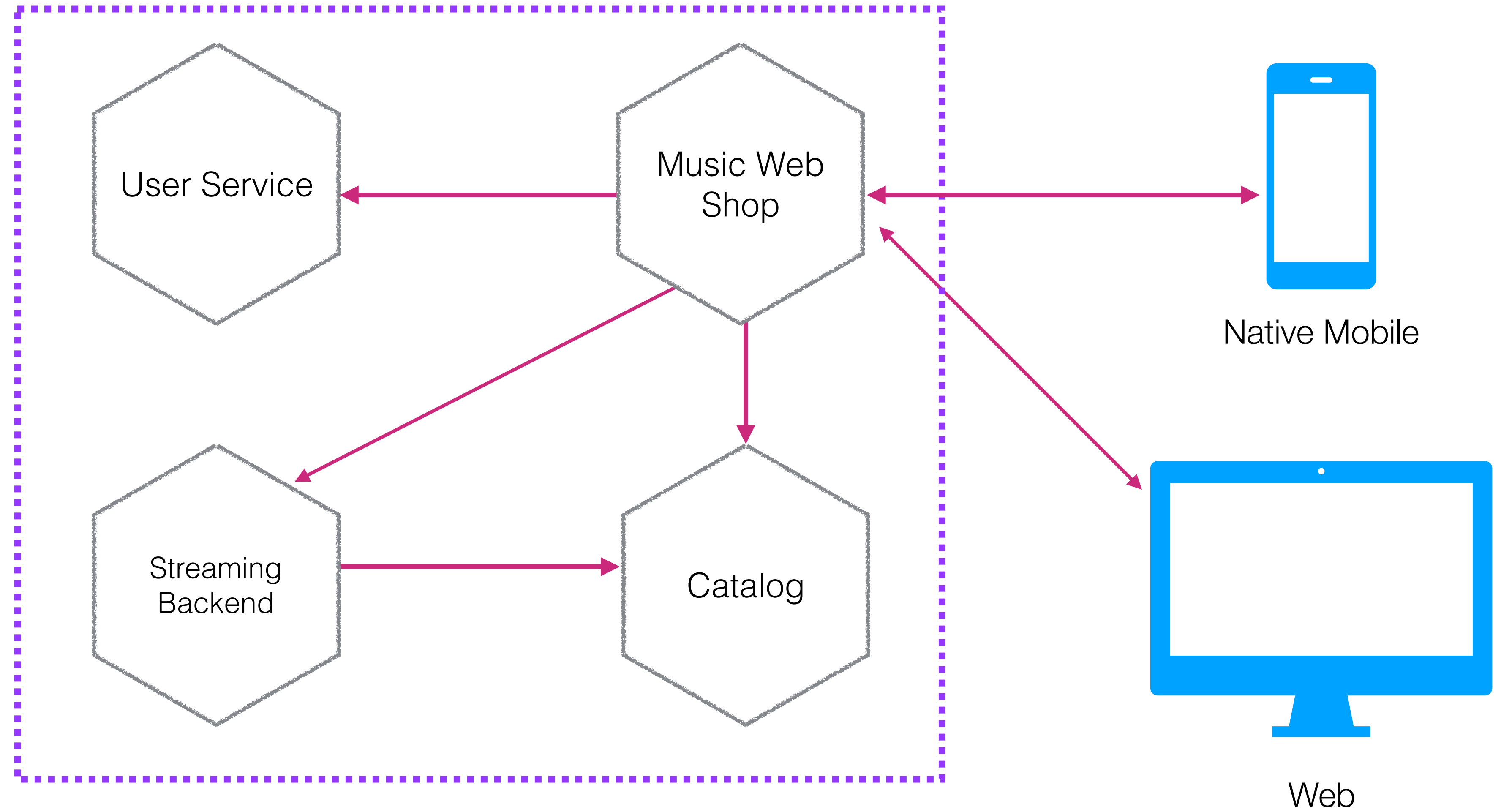
Native Mobile

# MUSIC CORP 2018

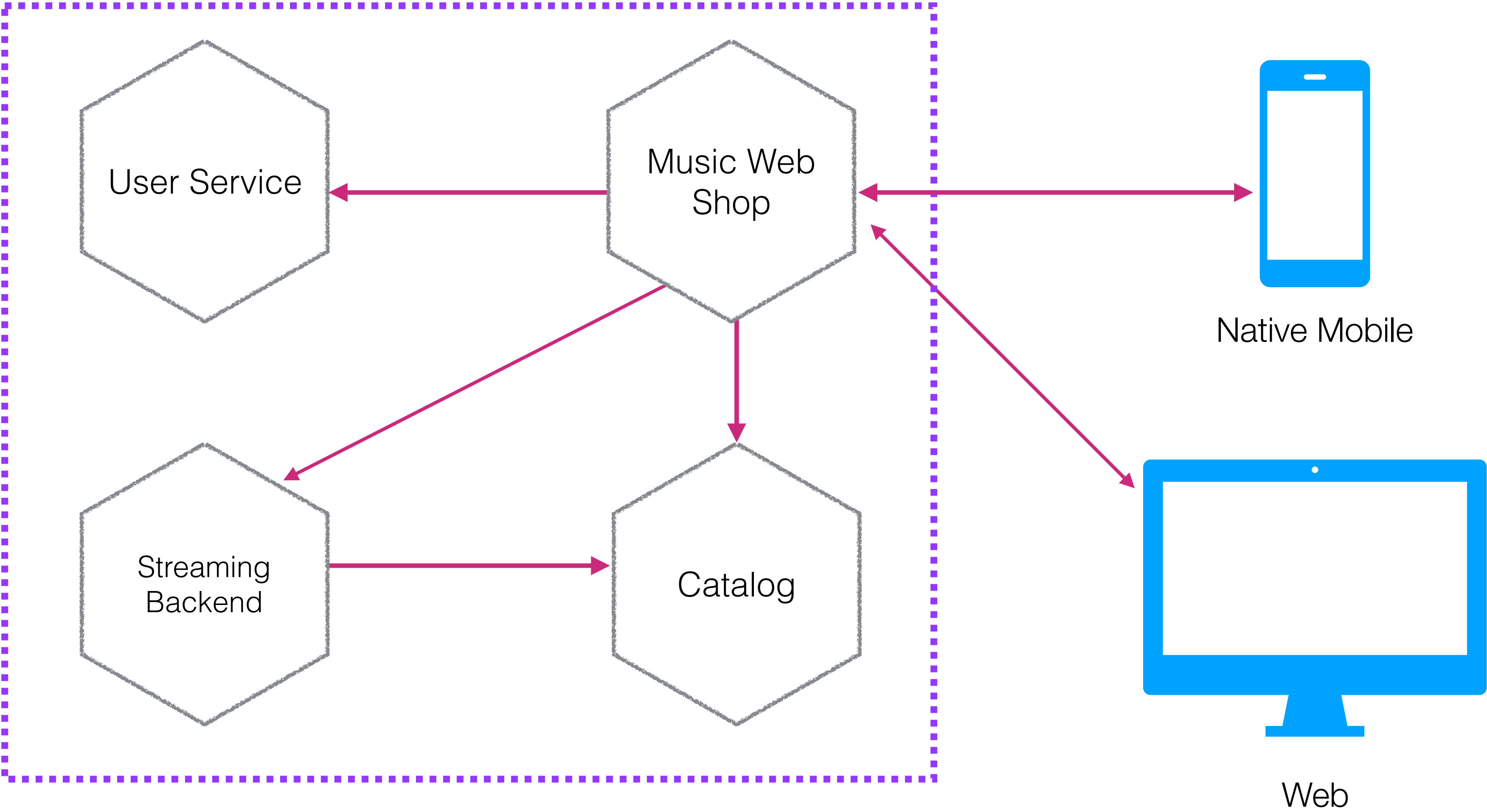




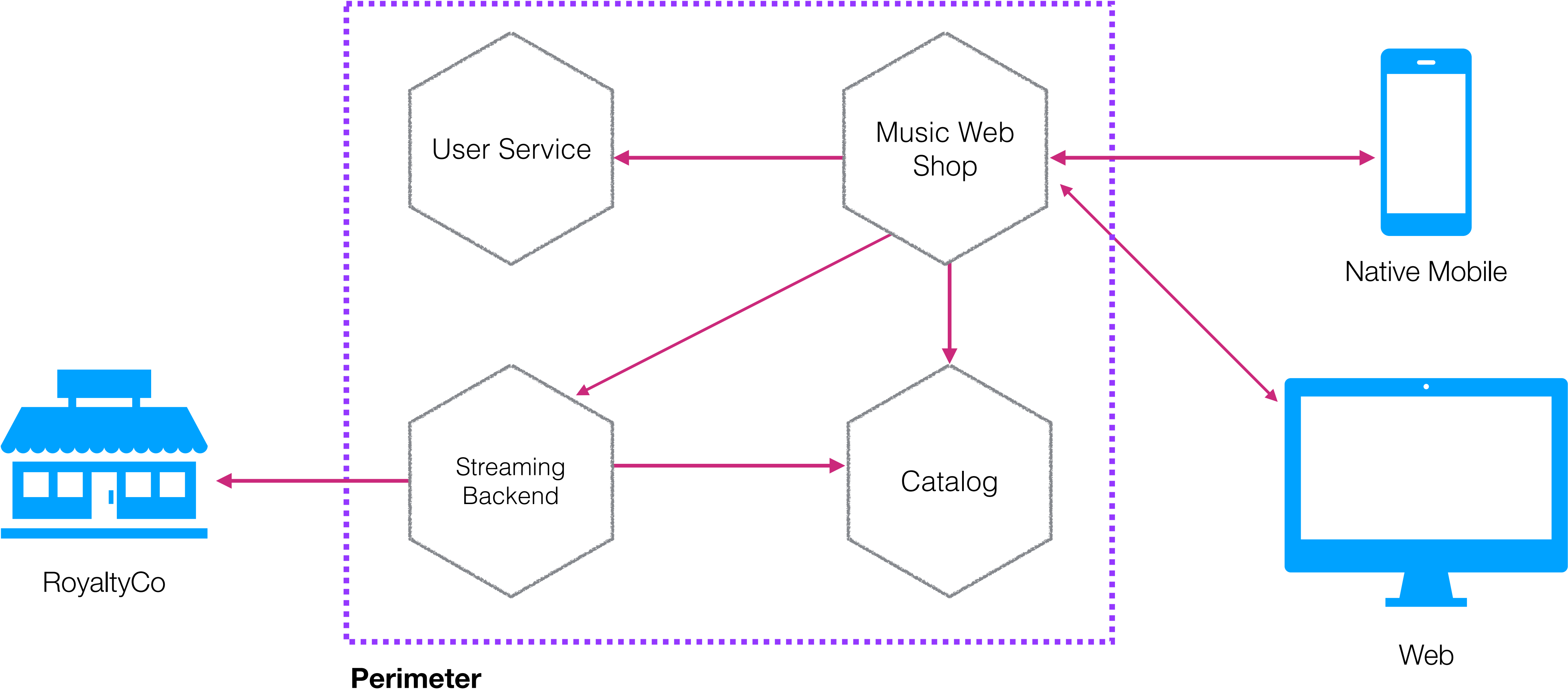
# MUSIC CORP 2018



# MUSIC CORP 2018



# MUSIC CORP 2018



# KEY CONCERNS OF TRANSPORT SECURITY



# KEY CONCERNS OF TRANSPORT SECURITY

Observation of data

# KEY CONCERNS OF TRANSPORT SECURITY

Observation of data

Manipulation of data

# KEY CONCERNS OF TRANSPORT SECURITY

Observation of data

Manipulation of data

Restricting access to endpoints

# KEY CONCERNS OF TRANSPORT SECURITY

Observation of data

Manipulation of data

Restricting access to endpoints

Impersonation of endpoints



# HTTPS Everywhere!

# HTTP + TLS

## Server guarantees!

Server guarantees!

Payload not manipulated



Server guarantees!

Payload not manipulated

Client guarantees?

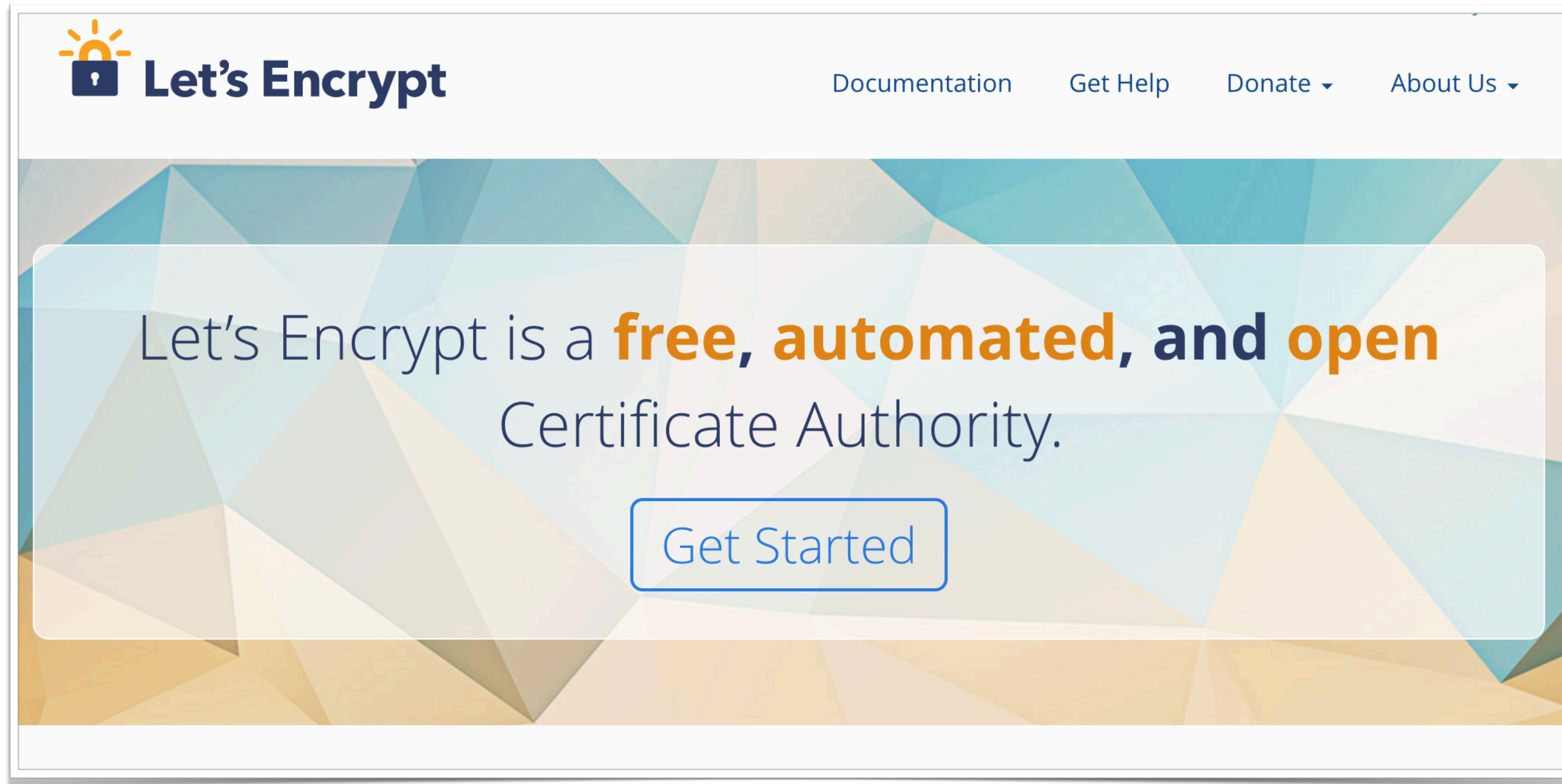
Server guarantees!

Payload not manipulated

Client guarantees?

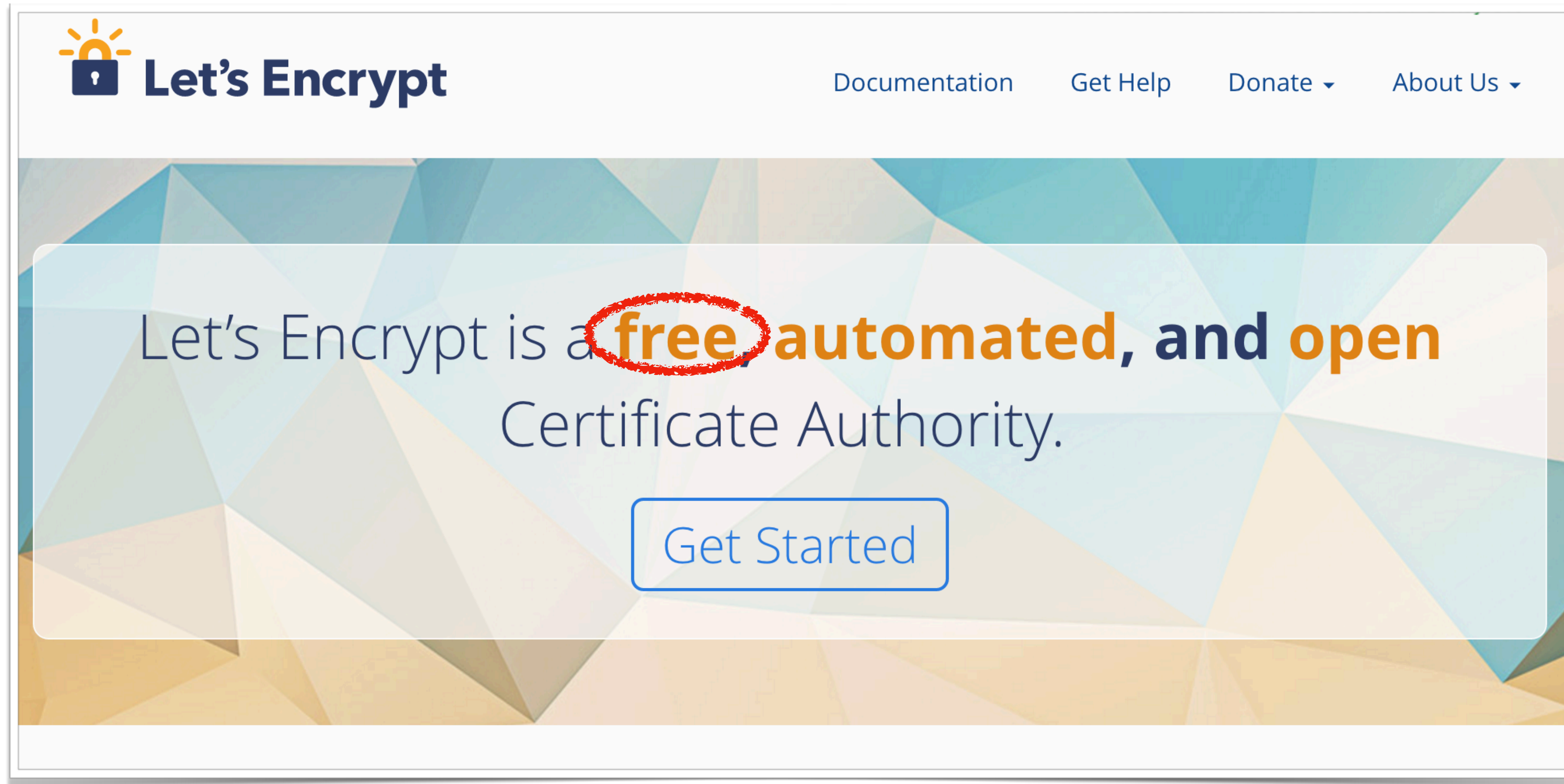
Certificate management can be painful

# LET'S ENCRYPT



<https://letsencrypt.org/>

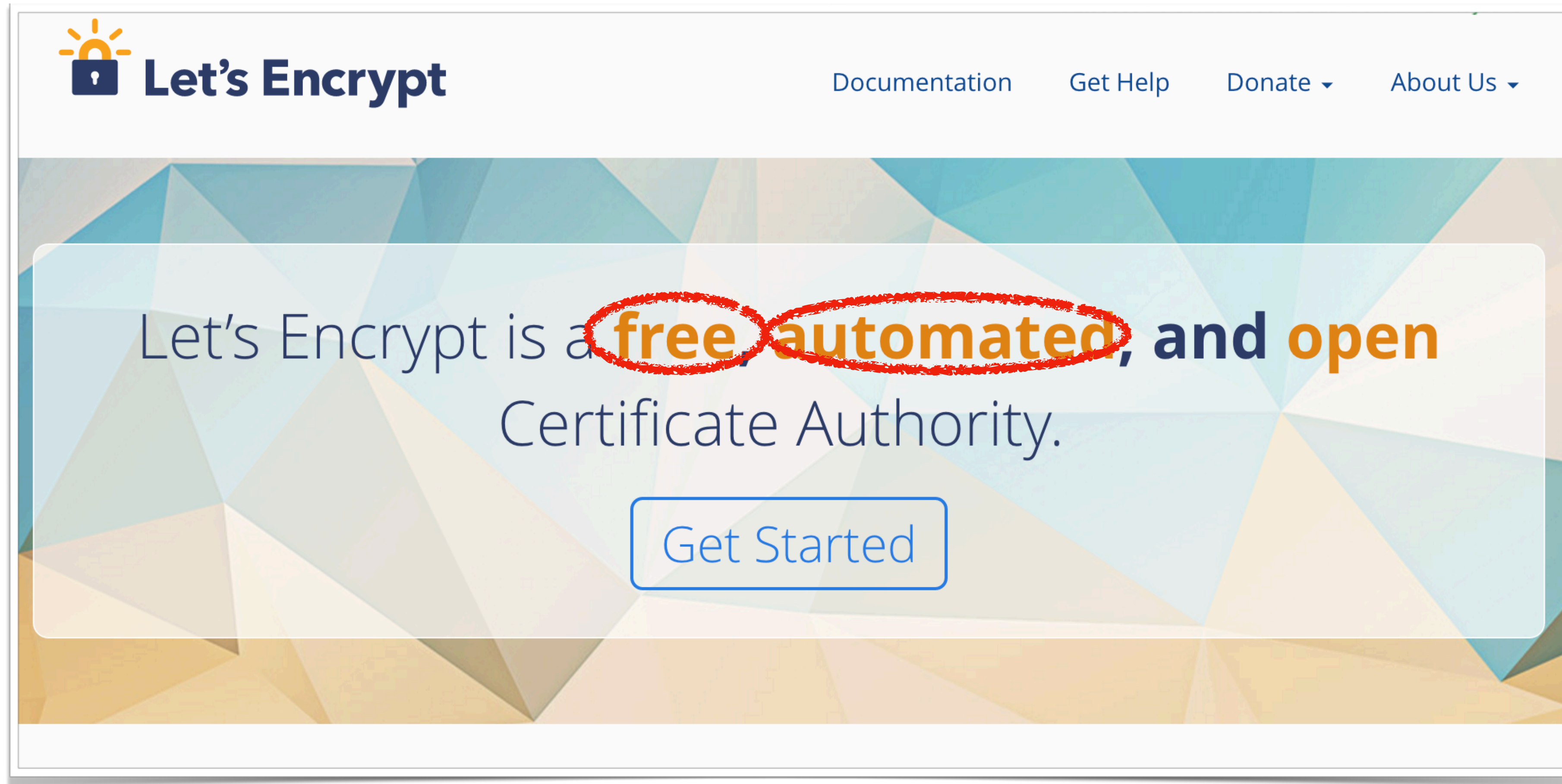
# LET'S ENCRYPT



<https://letsencrypt.org/>



# LET'S ENCRYPT



<https://letsencrypt.org/>

# AWS CERTIFICATE MANAGER

## AWS Certificate Manager

AWS Certificate Manager is a service that lets you easily provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services. SSL/TLS certificates are used to secure network communications and establish the identity of websites over the Internet. AWS Certificate Manager removes the time-consuming manual process of purchasing, uploading, and renewing SSL/TLS certificates. With AWS Certificate Manager, you can quickly request a certificate, deploy it on AWS resources such as Elastic Load Balancers, Amazon CloudFront distributions, and APIs on API Gateway, and let AWS Certificate Manager handle certificate renewals. SSL/TLS certificates provisioned through AWS Certificate Manager are free. You pay only for the AWS resources you create to run your application.

**Manage Your AWS  
Resources**

[Sign in to the Console](#)

**<https://aws.amazon.com/certificate-manager/>**

## HOW DOES THIS STACK UP?

Server guarantees!

Payload not manipulated

Client guarantees?

Certificate management can be  
painful

## HOW DOES THIS STACK UP?

Server guarantees!

Payload not manipulated

Client guarantees?

Certificate management can be painful

Observation of data

Manipulation of data

Restricting access to endpoints

Impersonation of endpoints



## HOW DOES THIS STACK UP?

Server guarantees!

Payload not manipulated

Client guarantees?

Certificate management can be painful



Observation of data

Manipulation of data

Restricting access to endpoints

Impersonation of endpoints

## HOW DOES THIS STACK UP?

Server guarantees!

Payload not manipulated

Client guarantees?

Certificate management can be painful

✓ Observation of data

✓ Manipulation of data

Restricting access to endpoints

Impersonation of endpoints

## HOW DOES THIS STACK UP?

Server guarantees!

Payload not manipulated

Client guarantees?

Certificate management can be painful

✓ Observation of data

✓ Manipulation of data

? Restricting access to endpoints

Impersonation of endpoints

## HOW DOES THIS STACK UP?

Server guarantees!

Payload not manipulated

Client guarantees?

Certificate management can be painful

✓ Observation of data

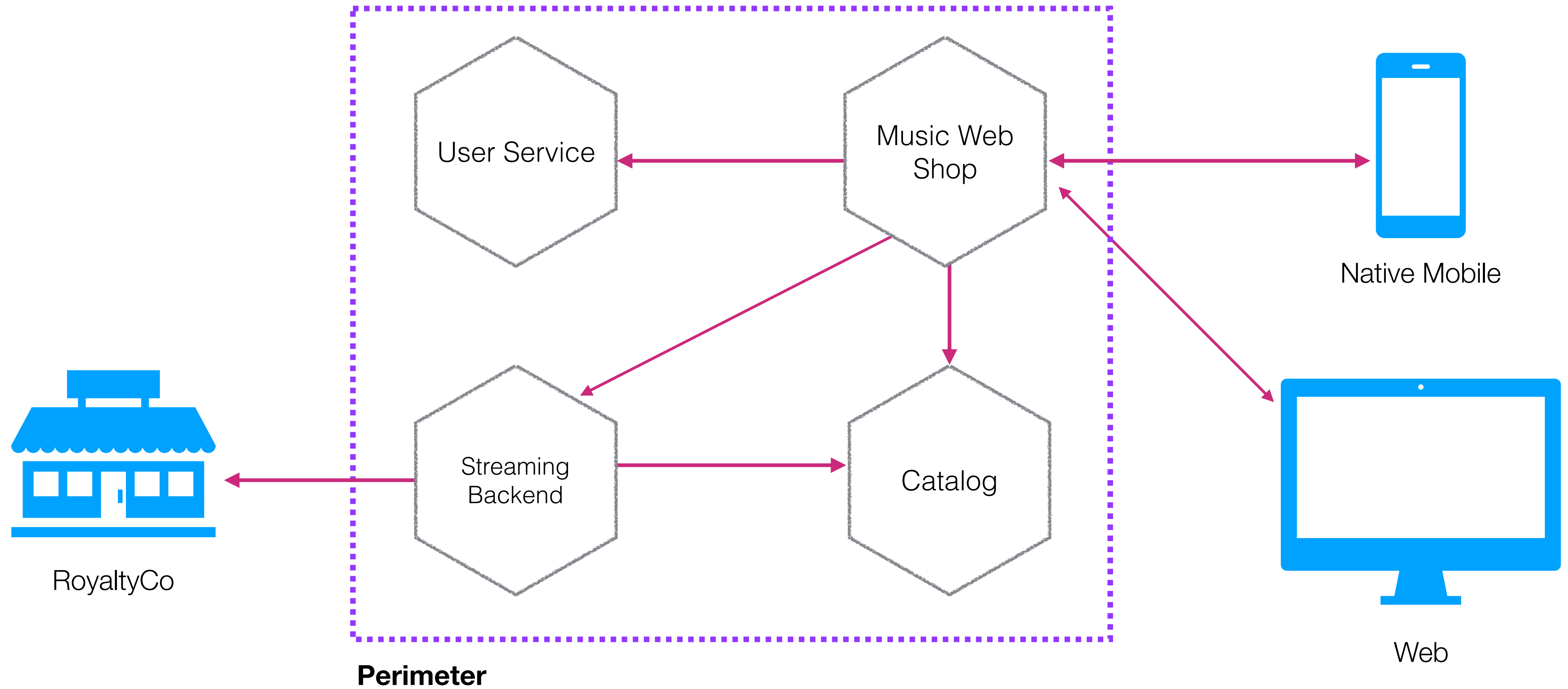
✓ Manipulation of data

? Restricting access to endpoints

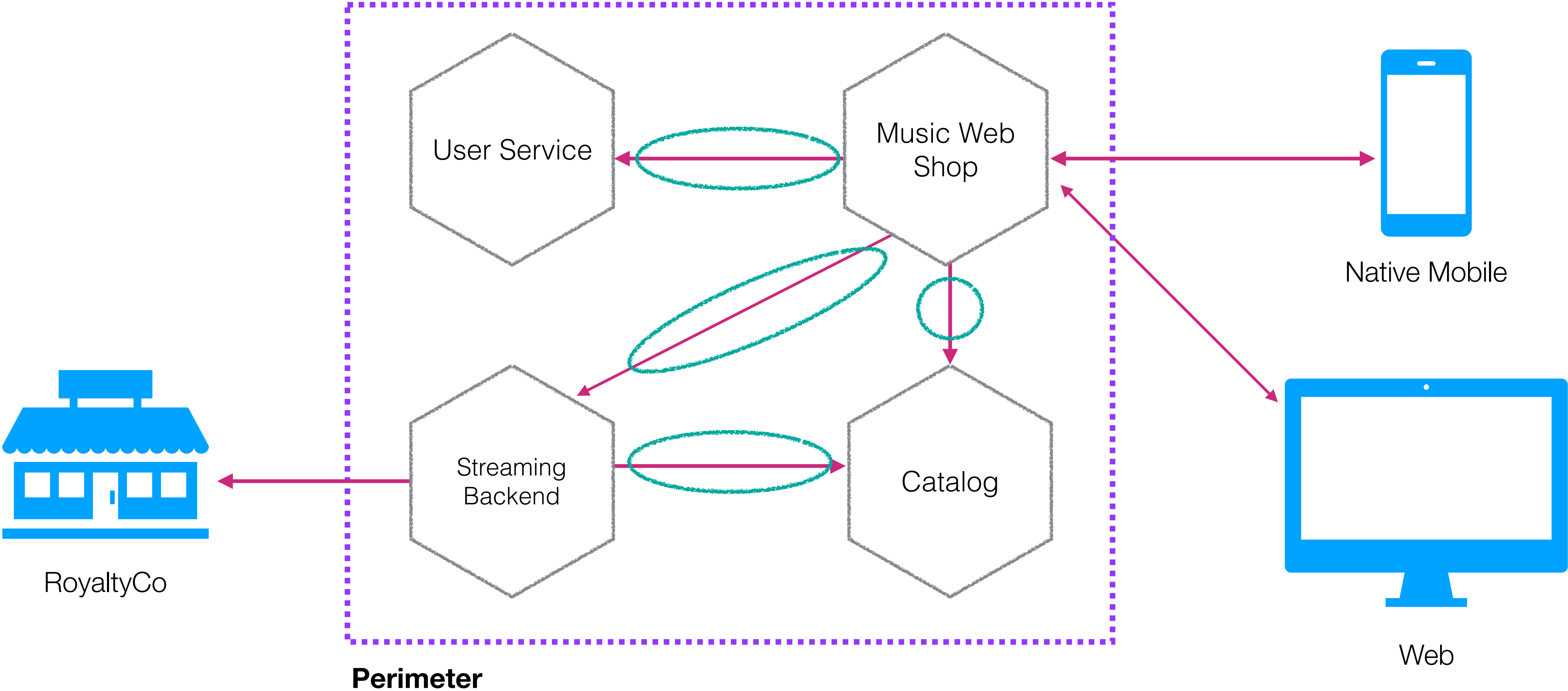
✓ Impersonation of endpoints



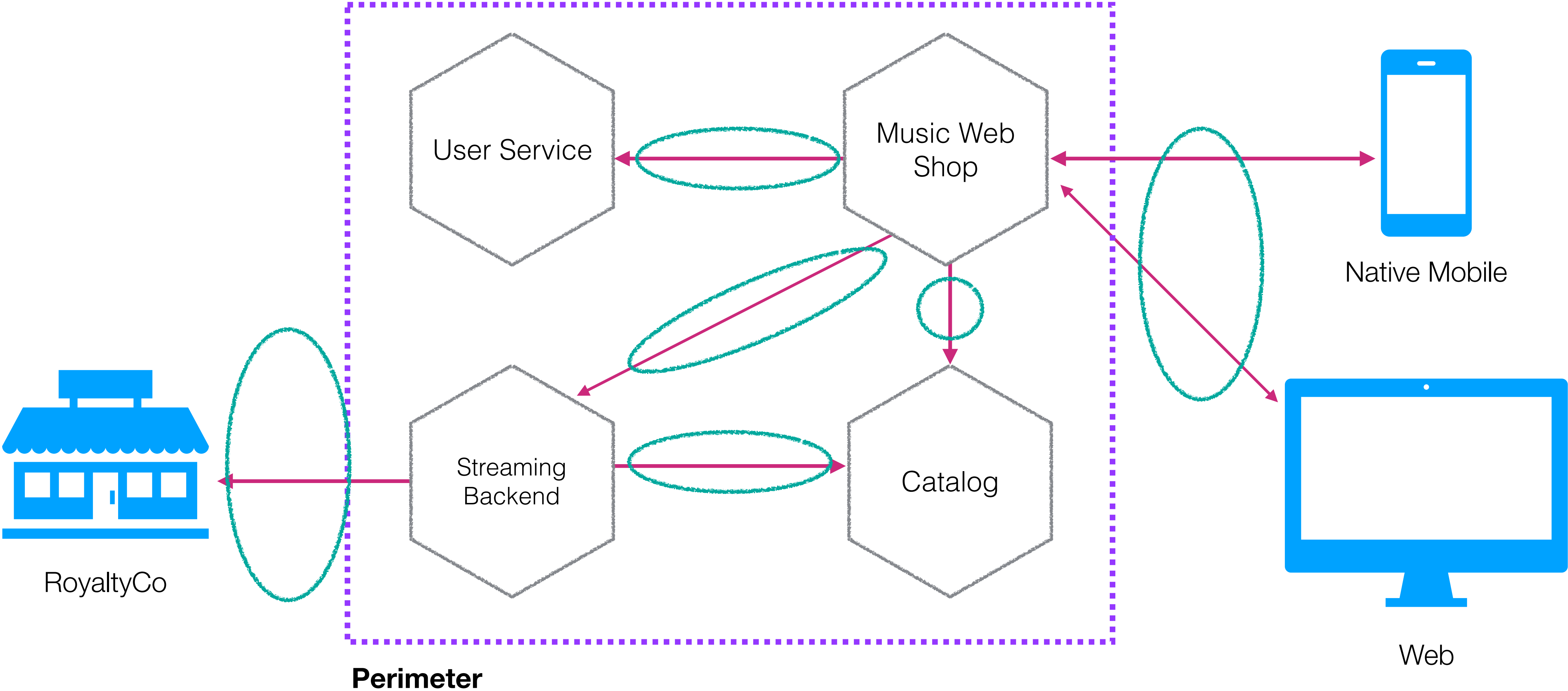
# HTTPS EVERYWHERE!



# HTTPS EVERYWHERE!



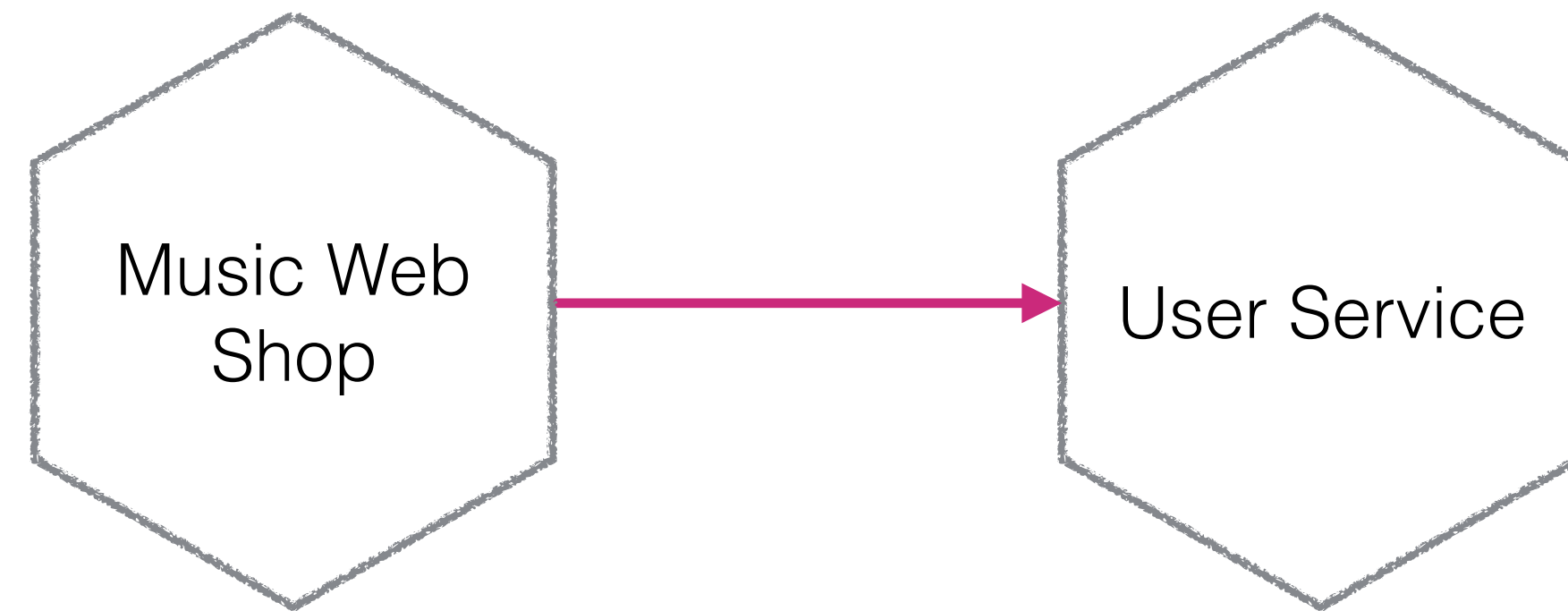
# HTTPS EVERYWHERE!



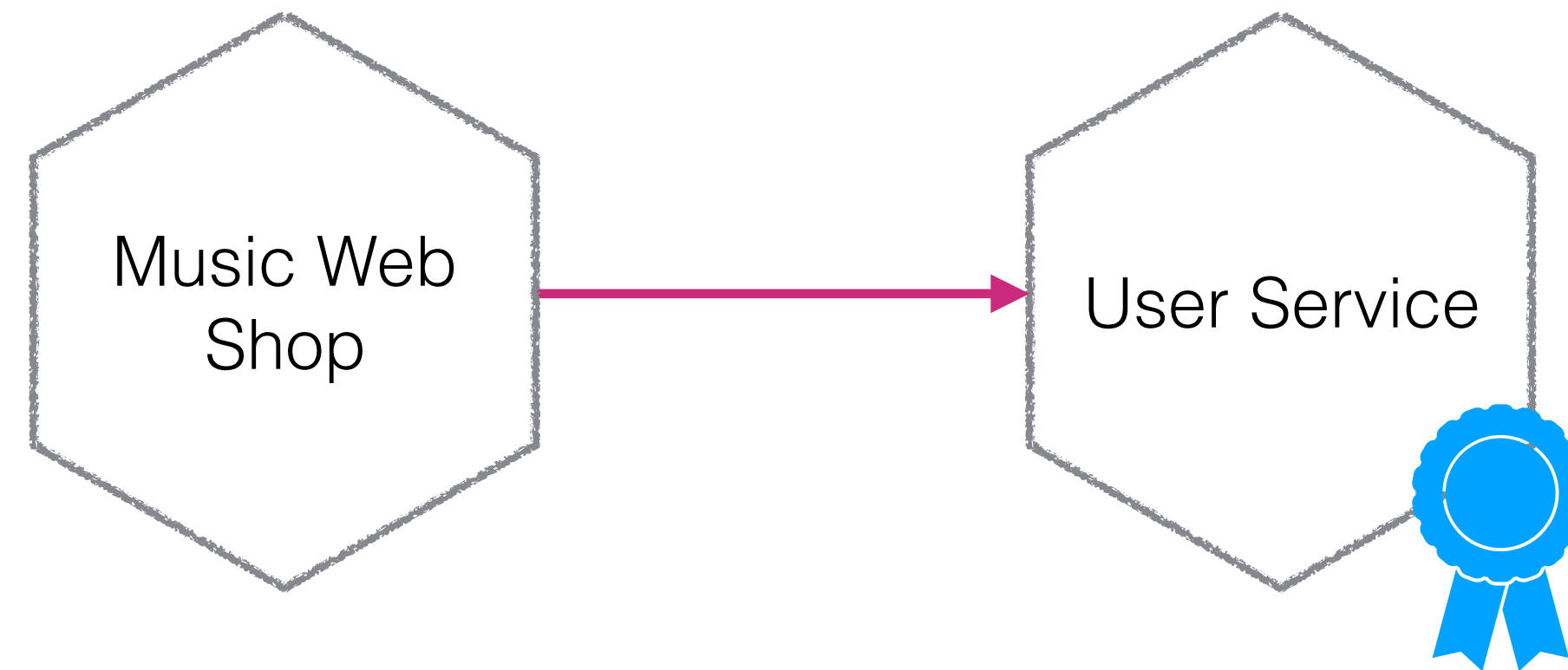
# Mutual TLS



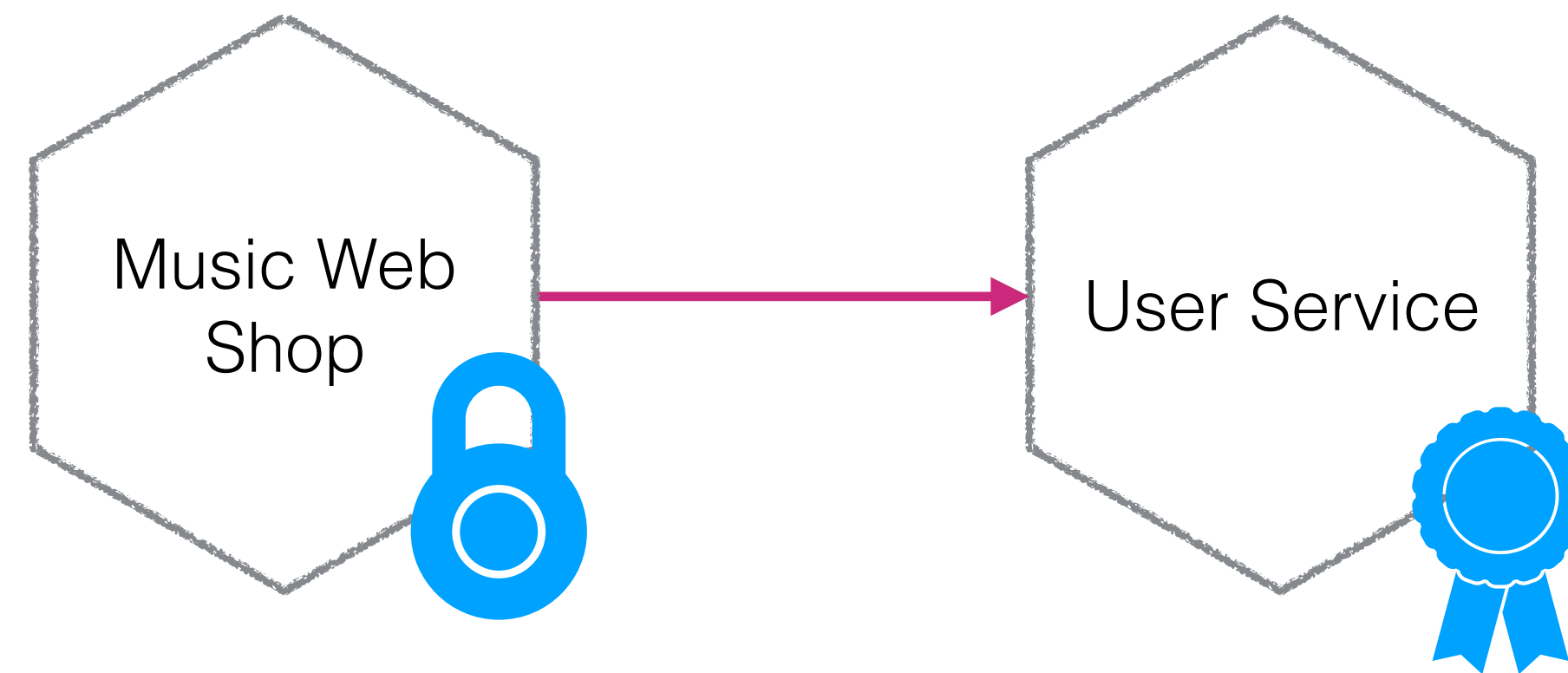
# MUTUAL TLS



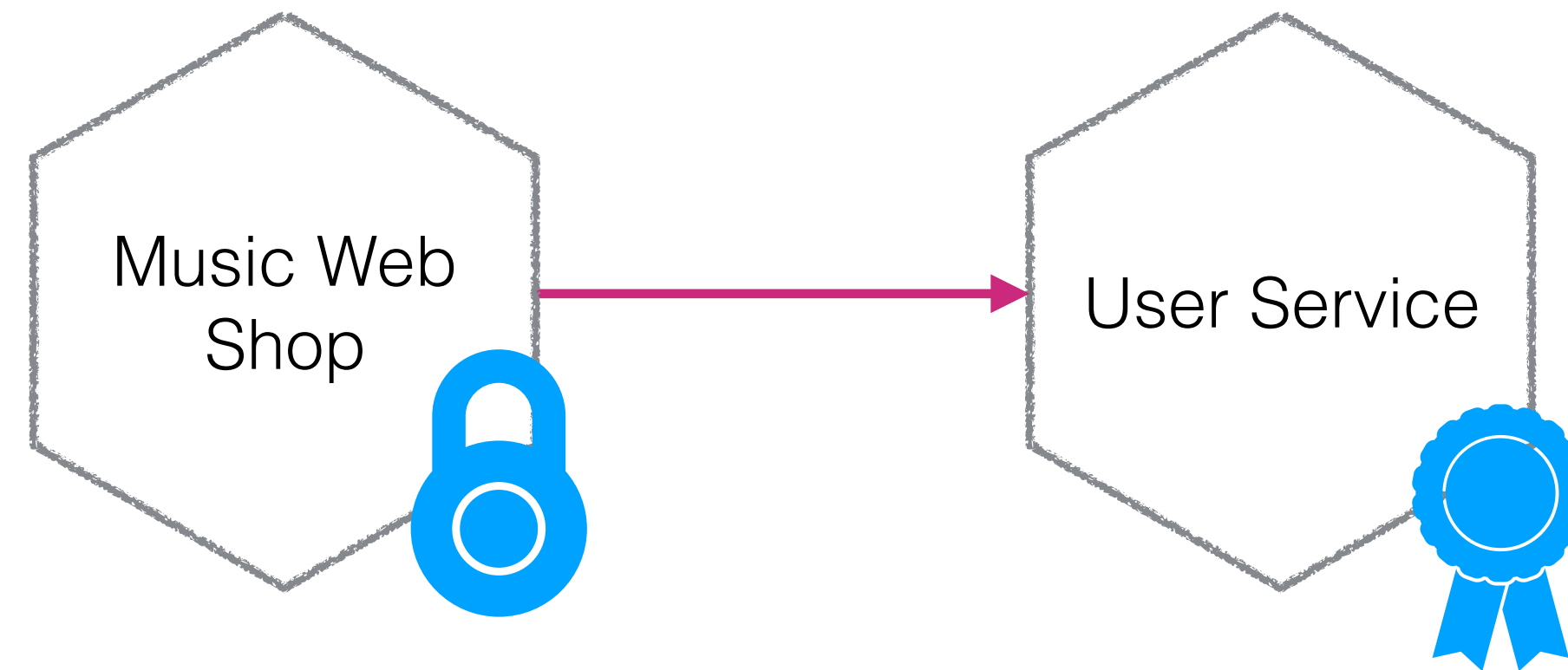
# MUTUAL TLS



# MUTUAL TLS



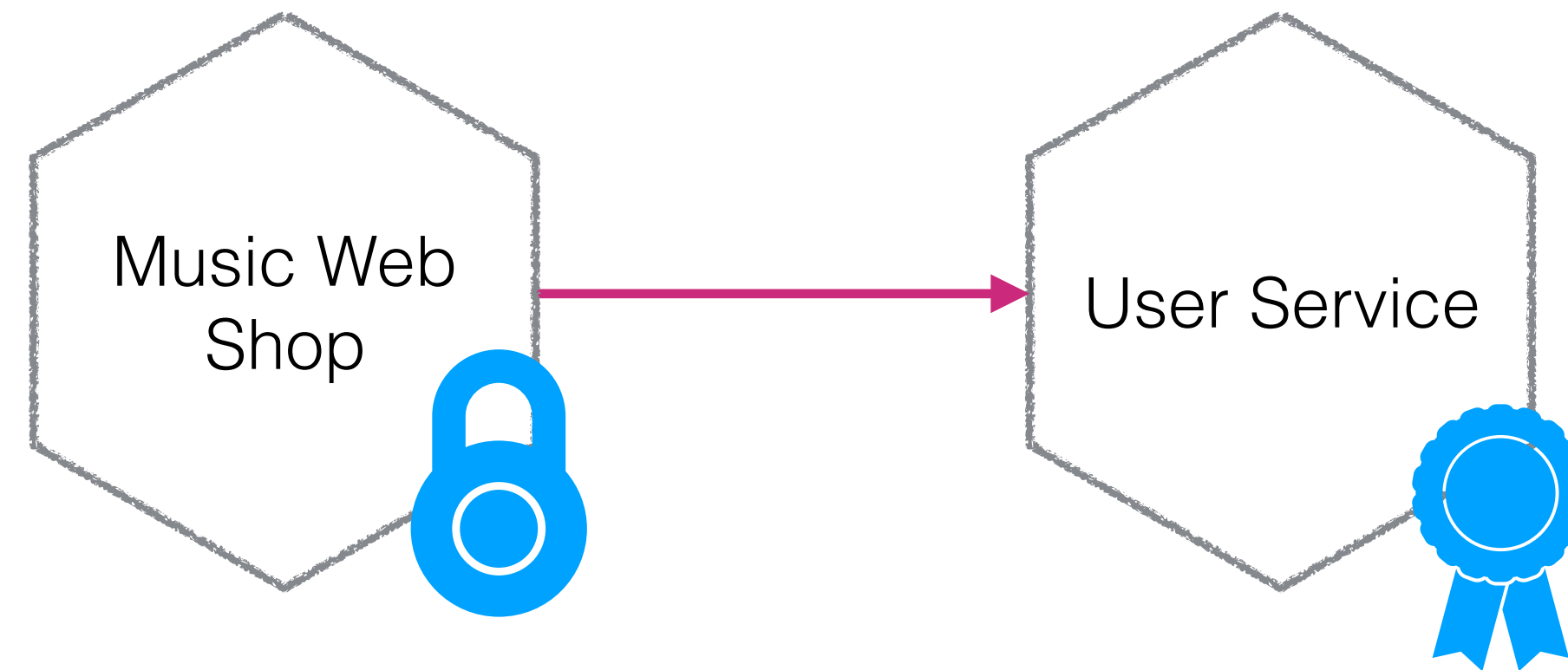
# MUTUAL TLS



Client and server guarantees!



# MUTUAL TLS



Client and server guarantees!

Certificate management is REALLY painful

# AZURE - CLIENT-SIDE CERTIFICATE MANAGEMENT

## How to secure back-end services using client certificate authentication in Azure API Management

📅 10/30/2017 • ⌚ 3 minutes to read • Contributors  all

### In this article

[Prerequisites](#)

[Upload a client certificate](#)

[Delete a client certificate](#)

[Configure an API to use a client certificate for gateway authentication](#)

[Self-signed certificates](#)

[Next steps](#)

API Management provides the capability to secure access to the back-end service of an API using client certificates. This guide shows how to manage certificates in the API publisher portal, and how to configure an API to use a certificate to access its back-end service.

For information about managing certificates using the API Management REST API, see [Azure API Management REST API Certificate entity](#).

<https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-mutual-certificates>

# AWS - CLIENT-SIDE CERTIFICATE MANAGEMENT

[AWS Documentation](#) » [Amazon API Gateway](#) » [Developer Guide](#) » [Controlling Access to an API in API Gateway](#) » [Use Client-Side SSL Certificates for Authentication by the Backend](#)

## Use Client-Side SSL Certificates for Authentication by the Backend

You can use API Gateway to generate an SSL certificate and use its public key in the backend to verify that HTTP requests to your backend system are from API Gateway. This allows your HTTP backend to control and accept only requests originating from Amazon API Gateway, even if the backend is publicly accessible.

### Note

Some backend servers may not support SSL client authentication as API Gateway does and could return an SSL certificate error. For a list of incompatible backend servers, see [Known Issues](#).

The SSL certificates that are generated by API Gateway are self-signed and only the public key of a certificate is visible in the API Gateway console or through the APIs.

### Topics

- [Generate a Client Certificate Using the API Gateway Console](#)
- [Configure an API to Use SSL Certificates](#)
- [Test Invoke](#)
- [Configure Backend to Authenticate API](#)

<https://docs.aws.amazon.com/apigateway/latest/developerguide/getting-started-client-side-ssl-authentication.html>

# MUTUAL TLS

Observation of data

Manipulation of data

Restricting access to endpoints

Impersonation of endpoints



# MUTUAL TLS

✓ Observation of data

Manipulation of data

Restricting access to endpoints

Impersonation of endpoints

# MUTUAL TLS

✓ Observation of data

✓ Manipulation of data

Restricting access to endpoints

Impersonation of endpoints

# MUTUAL TLS

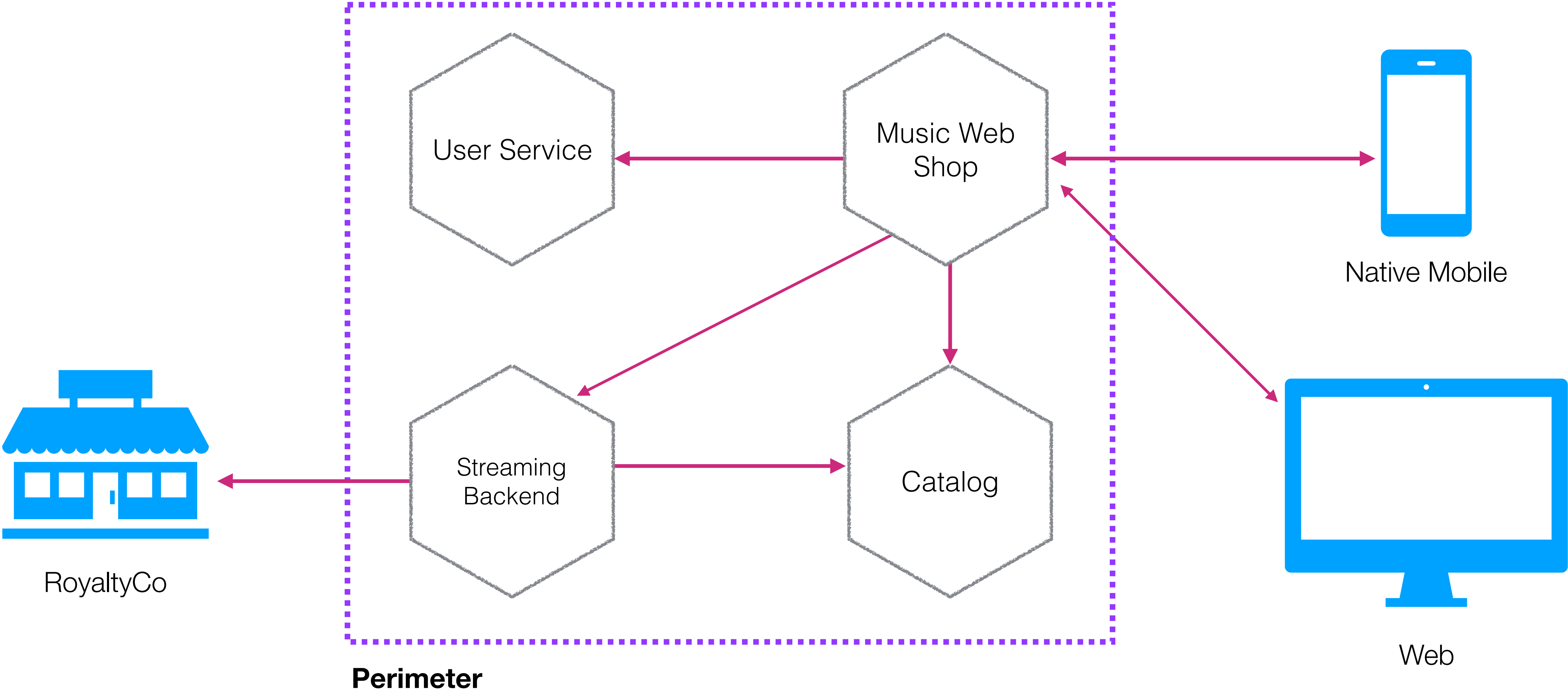
- ✓ Observation of data
- ✓ Manipulation of data
- ✓ Restricting access to endpoints
- Impersonation of endpoints

# MUTUAL TLS

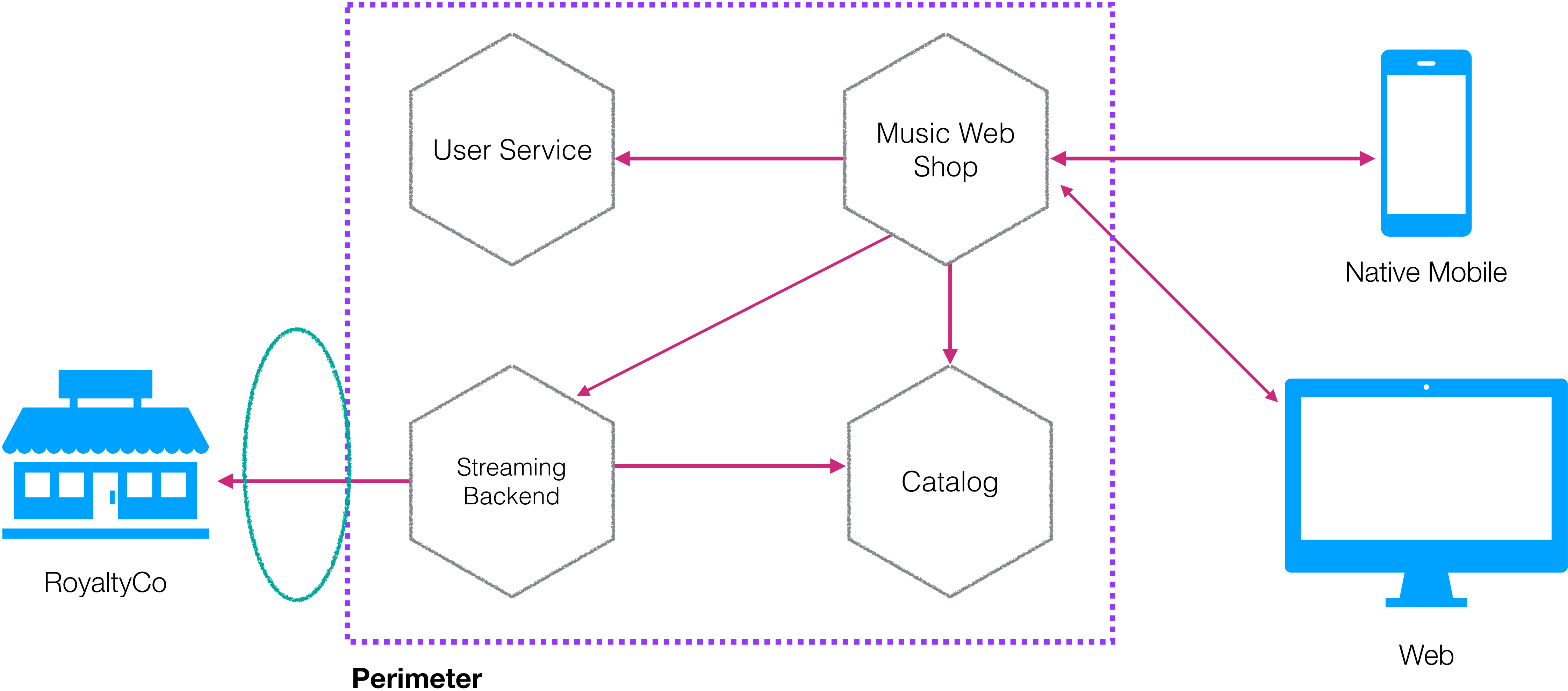
- ✓ Observation of data
- ✓ Manipulation of data
- ✓ Restricting access to endpoints
- ✓ Impersonation of endpoints



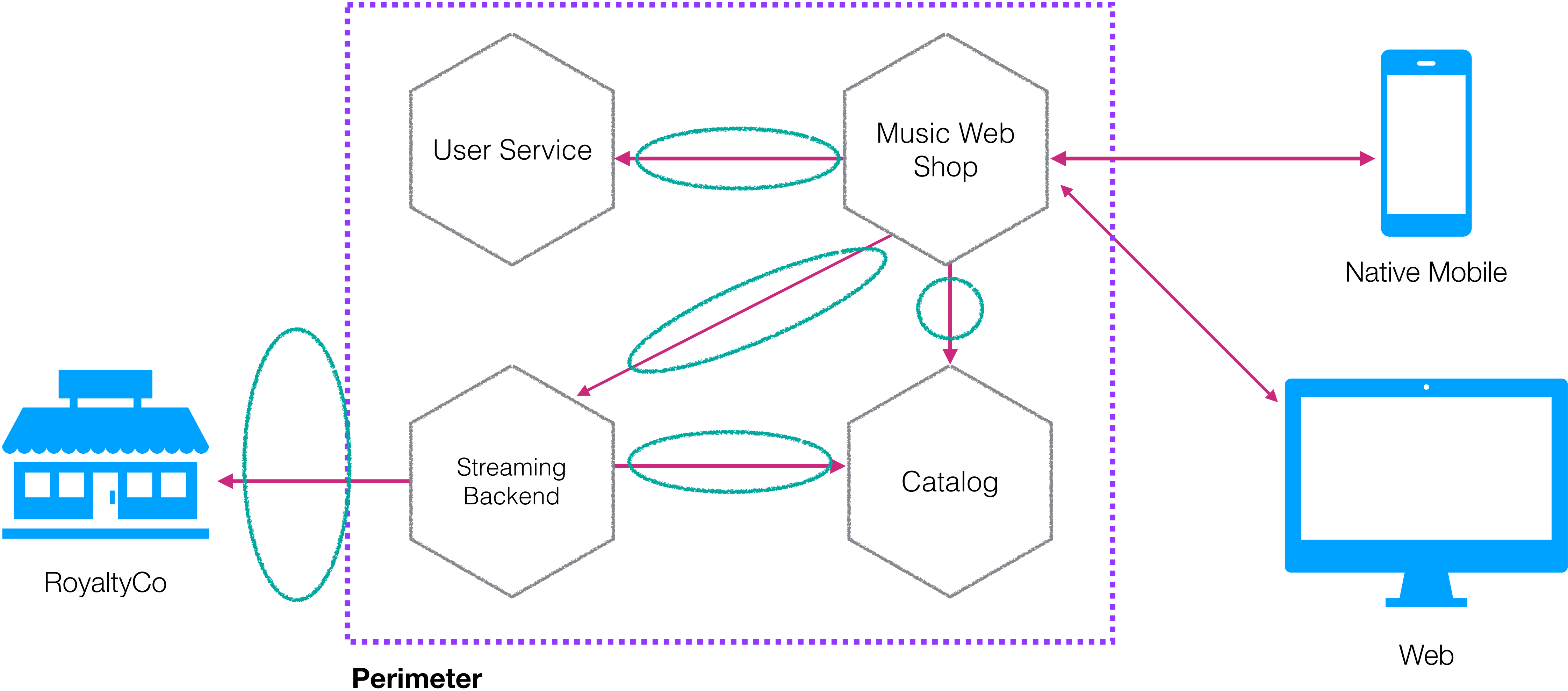
# MUTUAL TLS



# MUTUAL TLS



# MUTUAL TLS



## OTHER PROTOCOLS?





## OTHER PROTOCOLS?



**Synchronous, via HTTP, RPC or other**

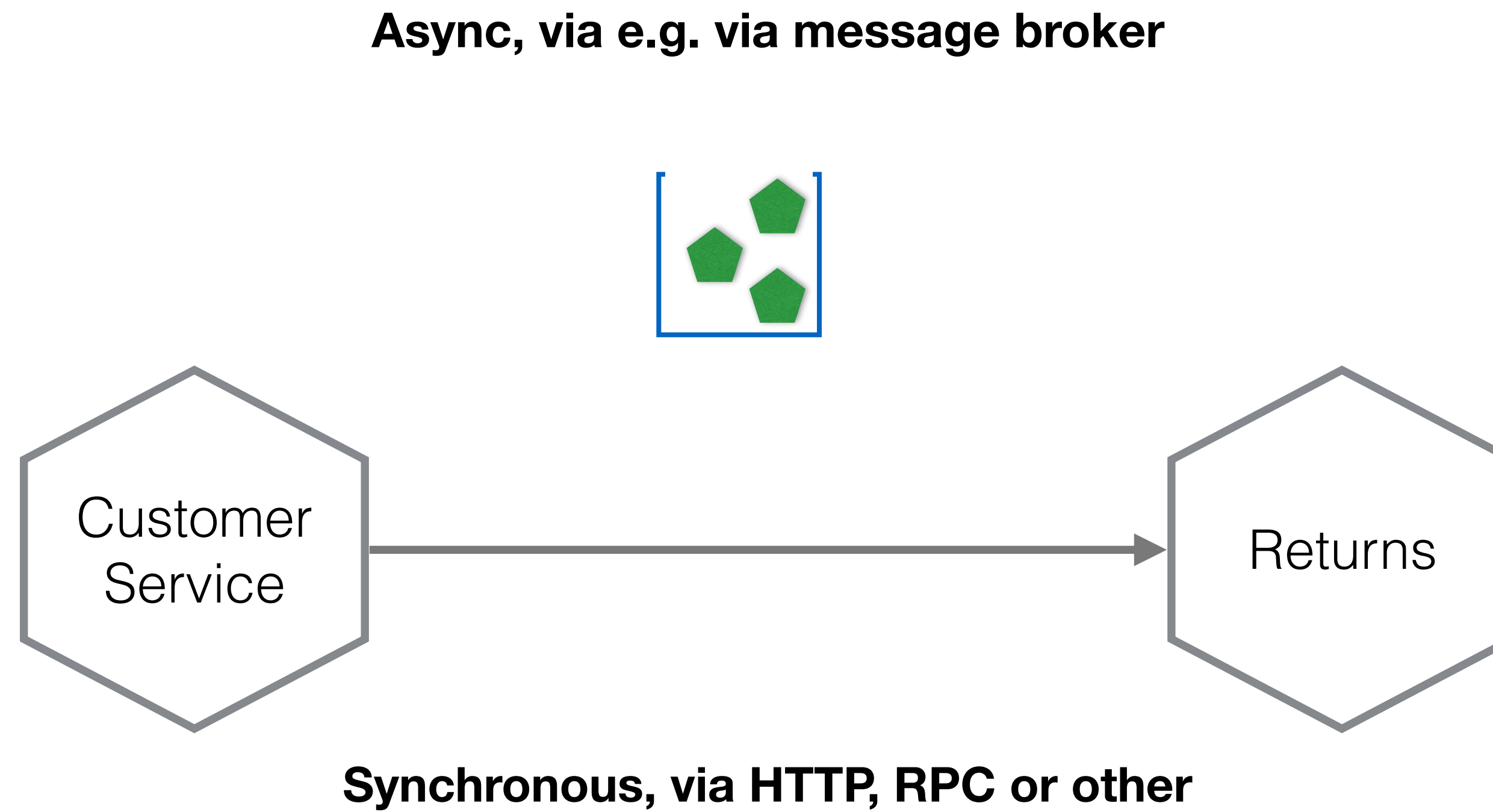
## OTHER PROTOCOLS?

**Async, via e.g. via message broker**

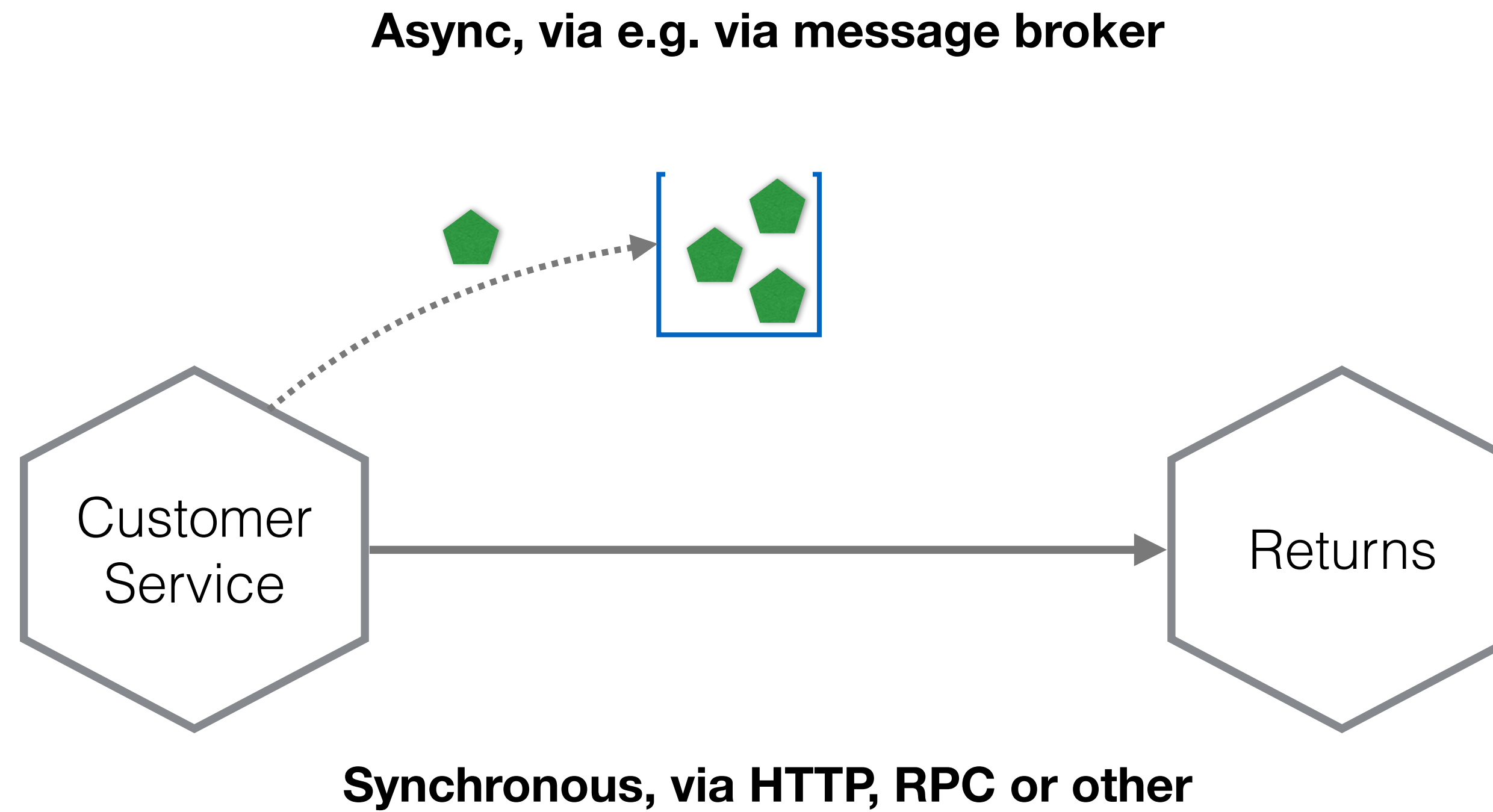


**Synchronous, via HTTP, RPC or other**

## OTHER PROTOCOLS?

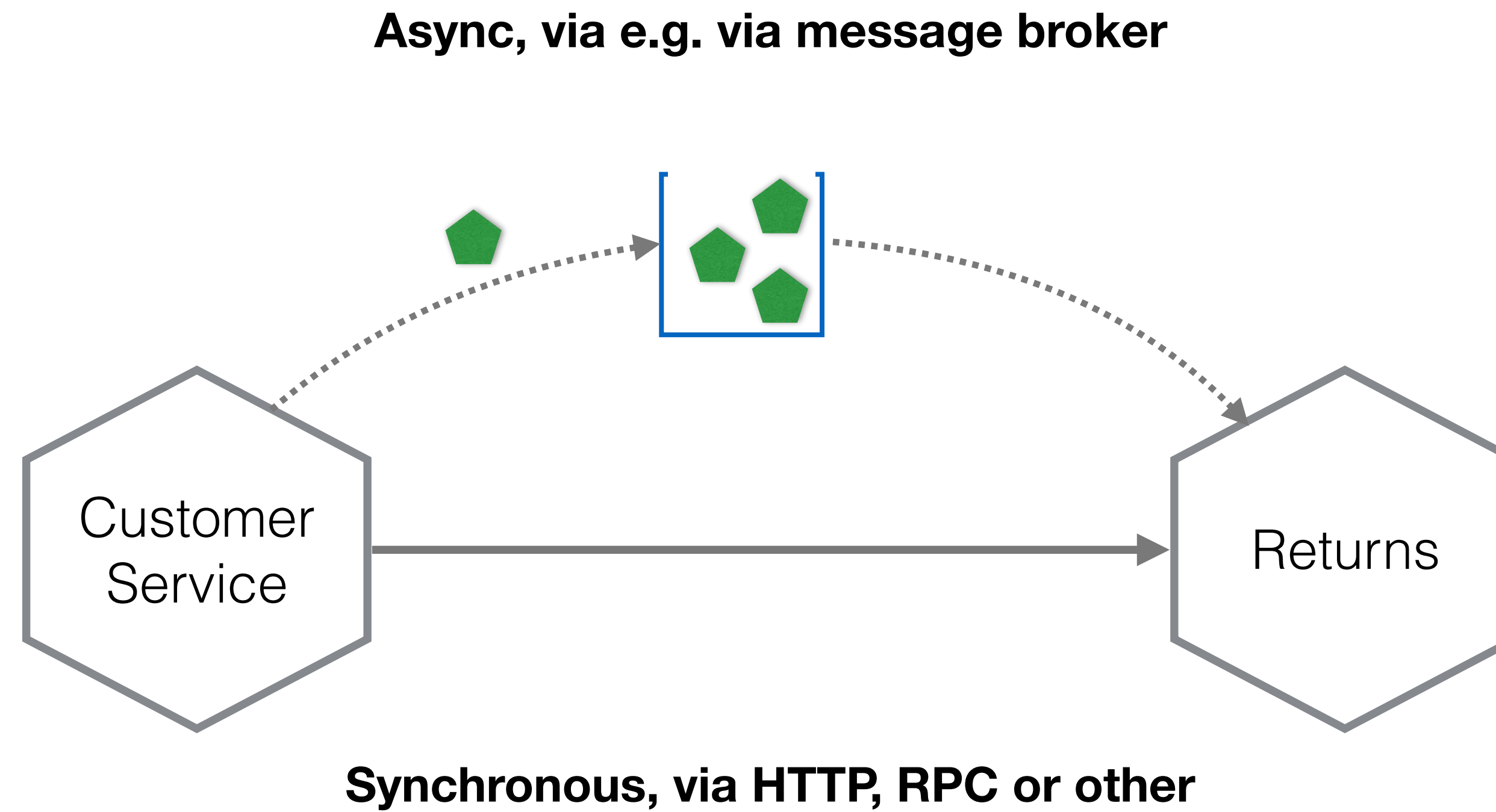


## OTHER PROTOCOLS?

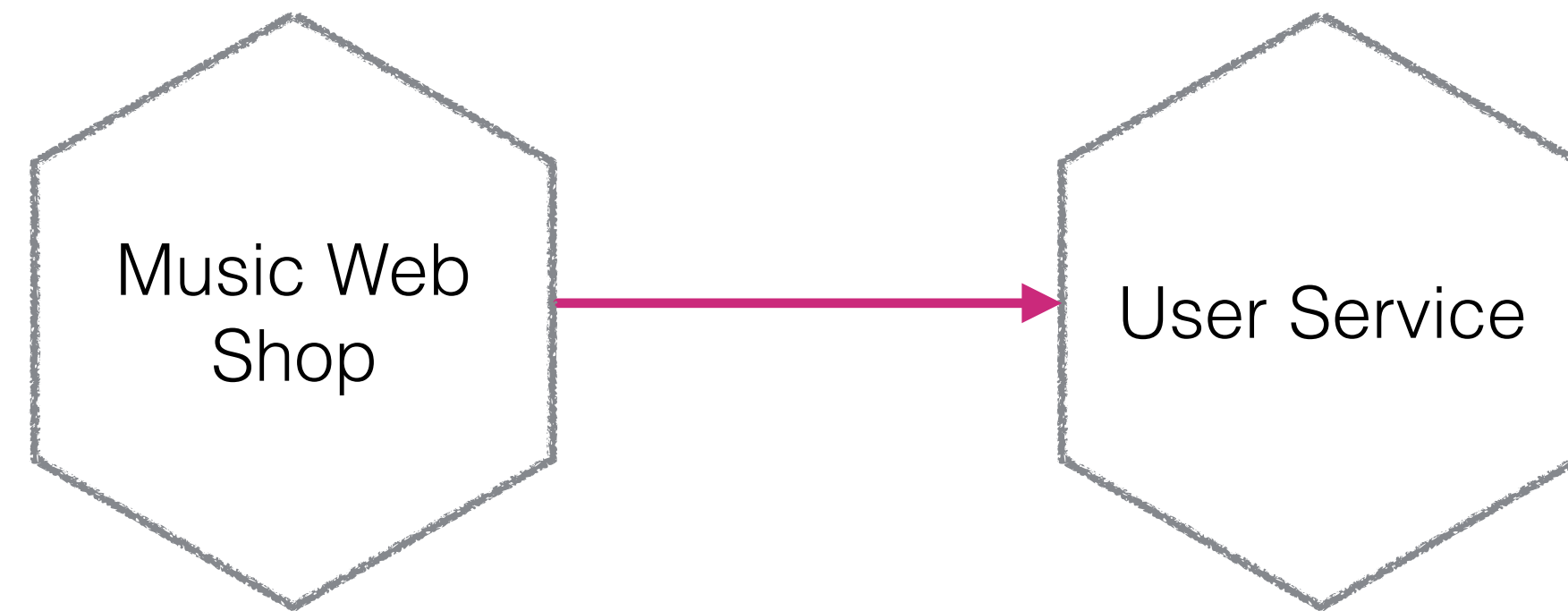




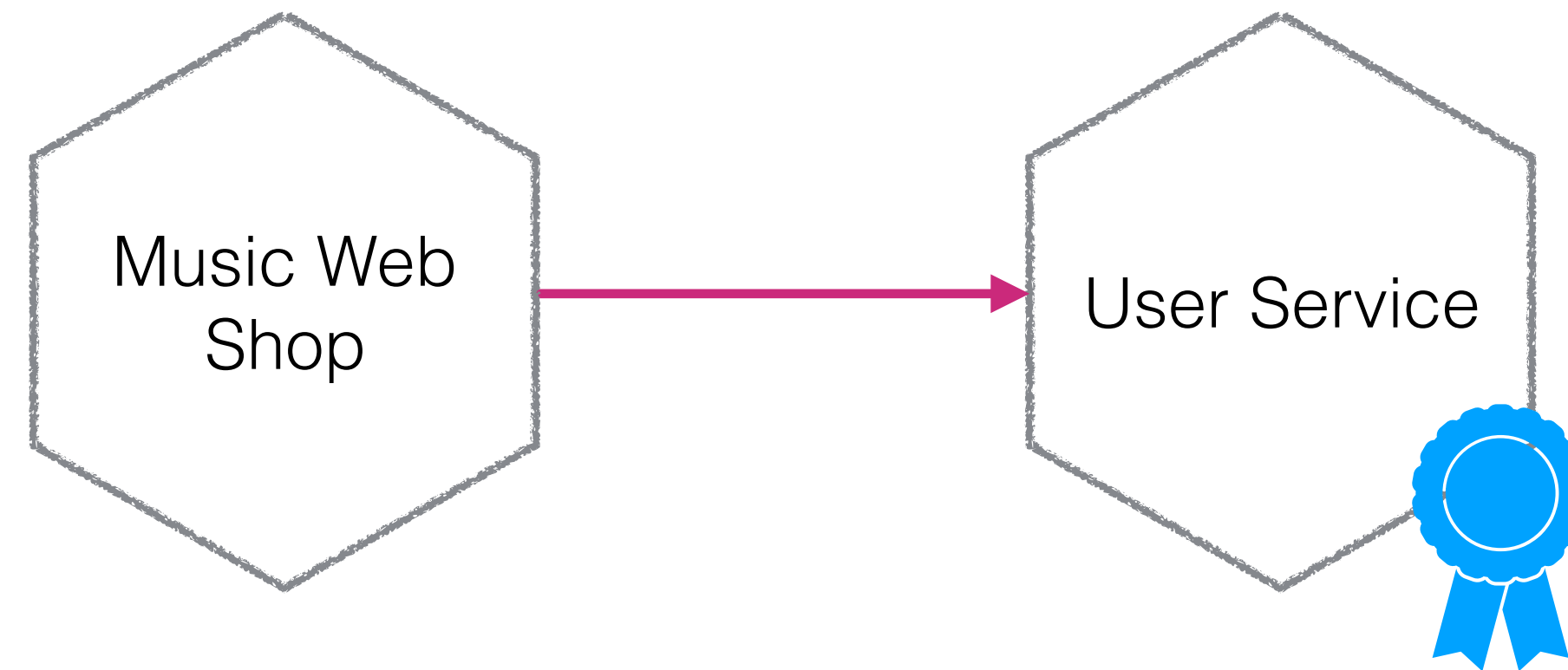
## OTHER PROTOCOLS?



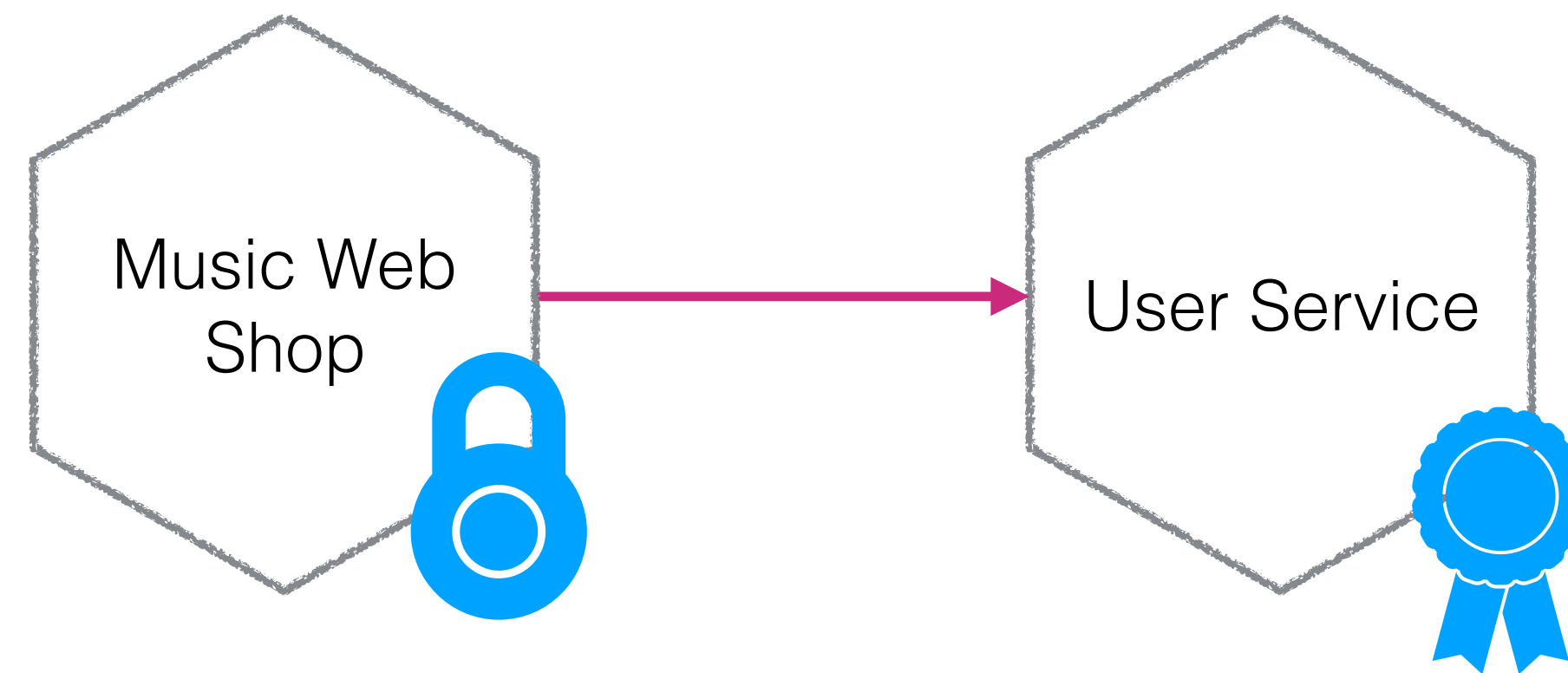
# TRANSPORT AUTHENTICATION



# TRANSPORT AUTHENTICATION

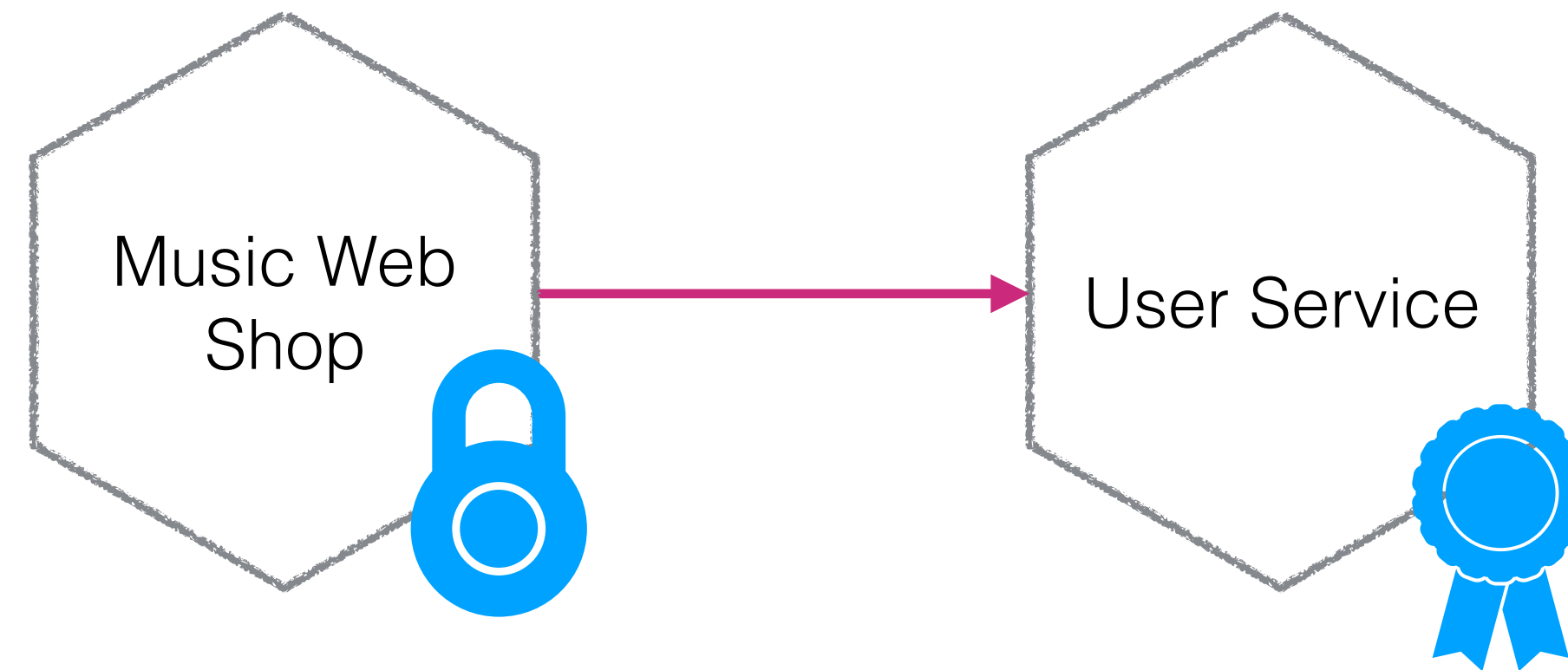


# TRANSPORT AUTHENTICATION



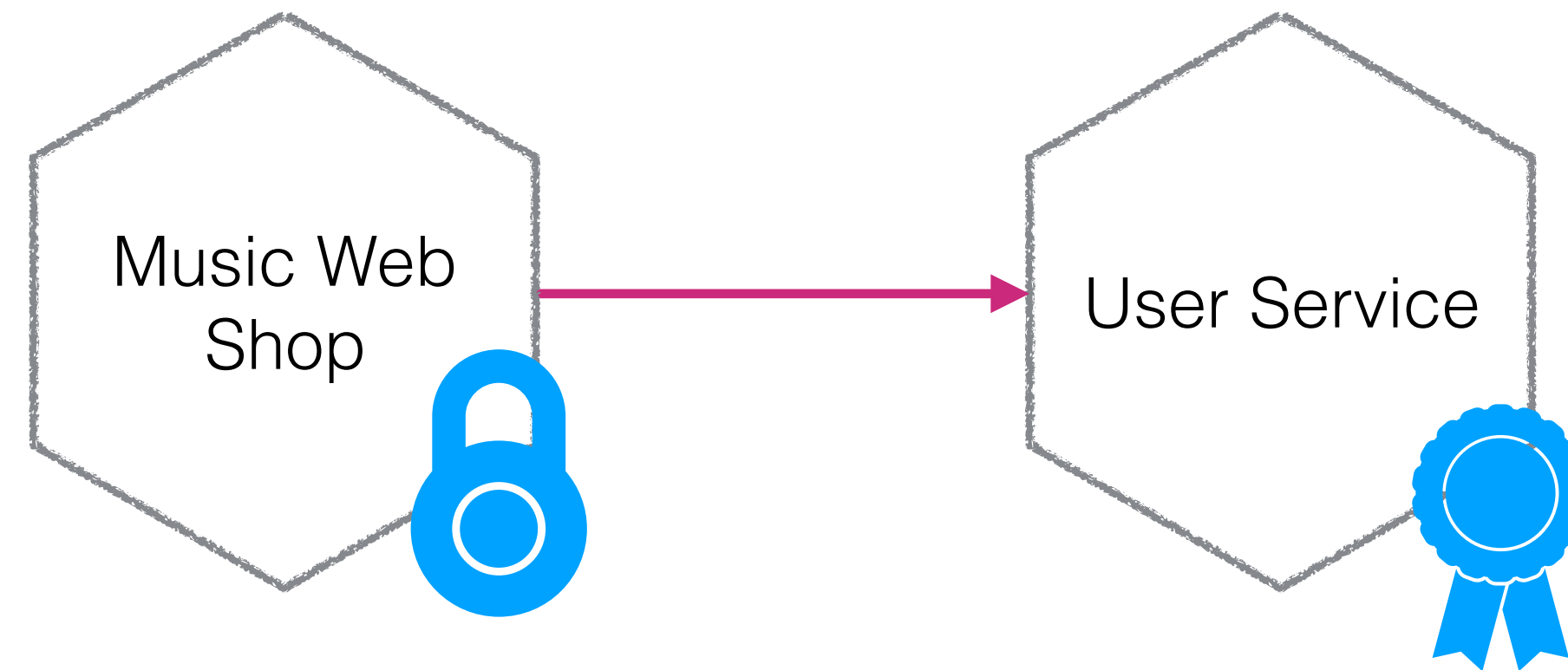


# TRANSPORT AUTHENTICATION



Server-side identity

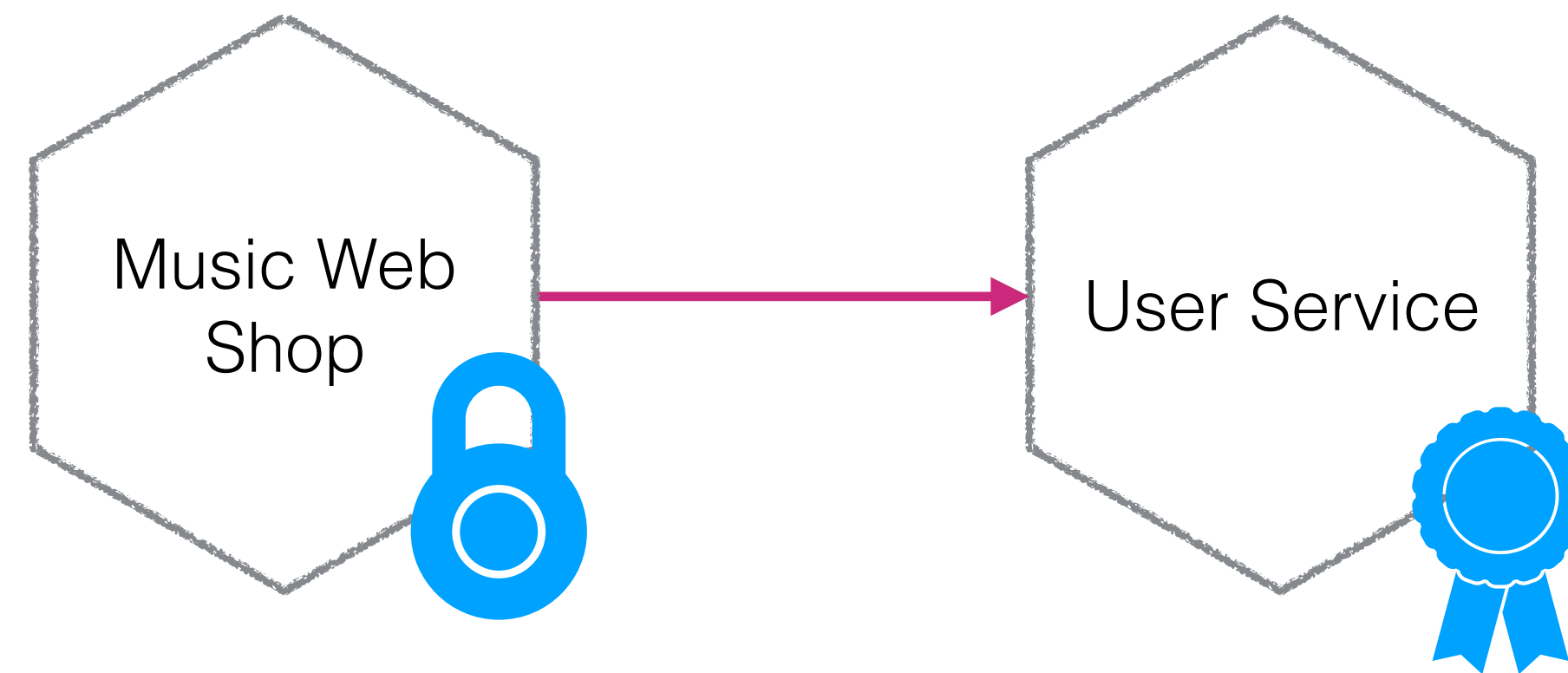
# TRANSPORT AUTHENTICATION



Client-side identity

Server-side identity

# TRANSPORT AUTHENTICATION



Client-side identity

Server-side identity

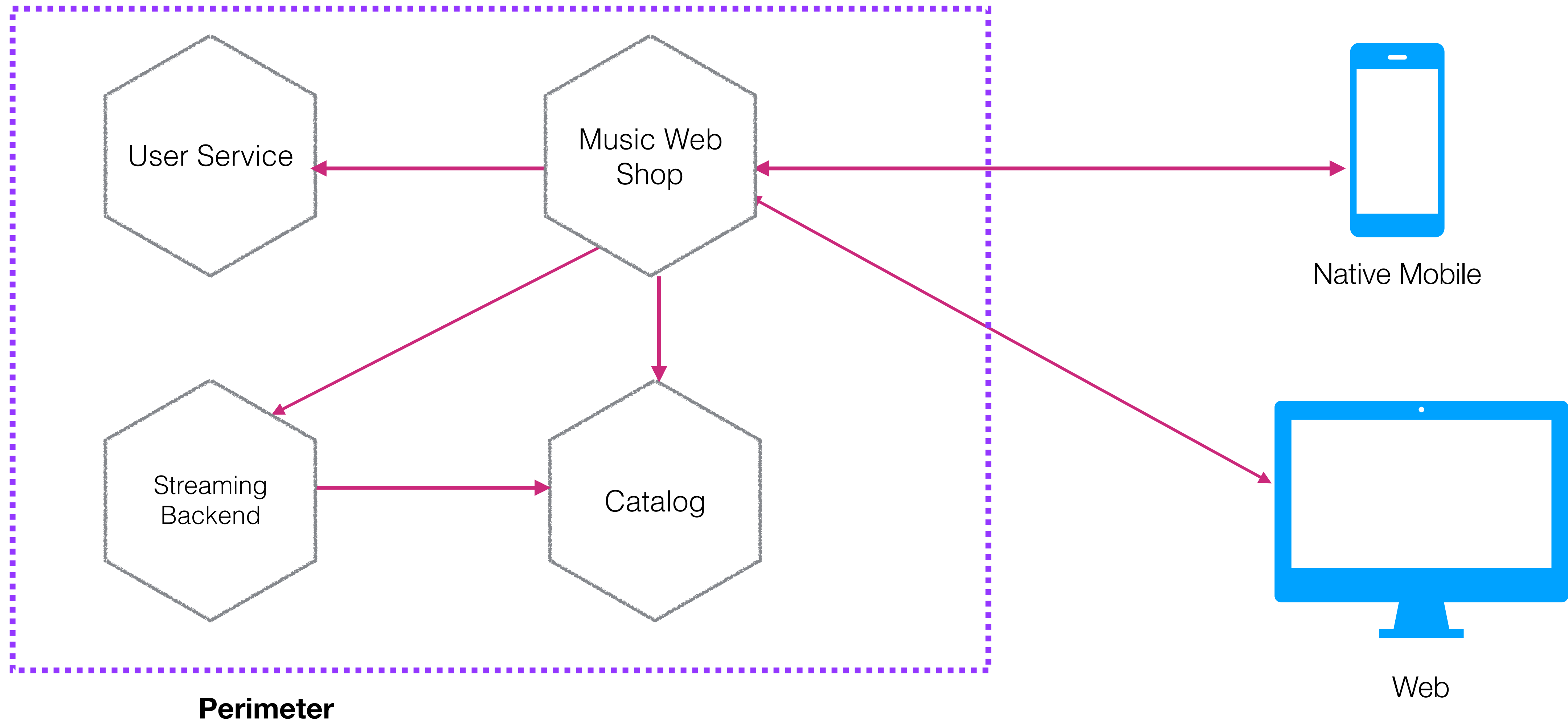
= service-to-service authentication





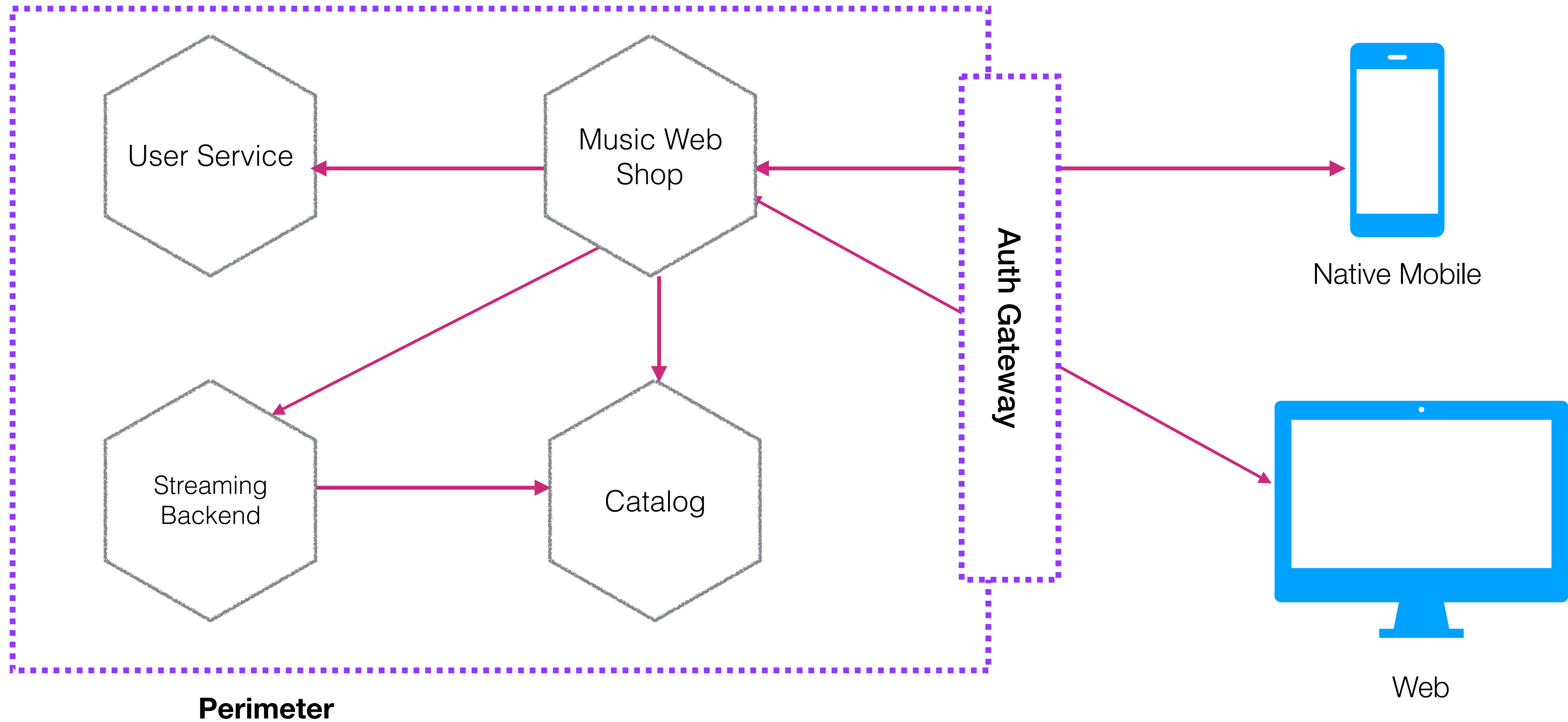


# USER AUTHENTICATION - PROXY-BASED

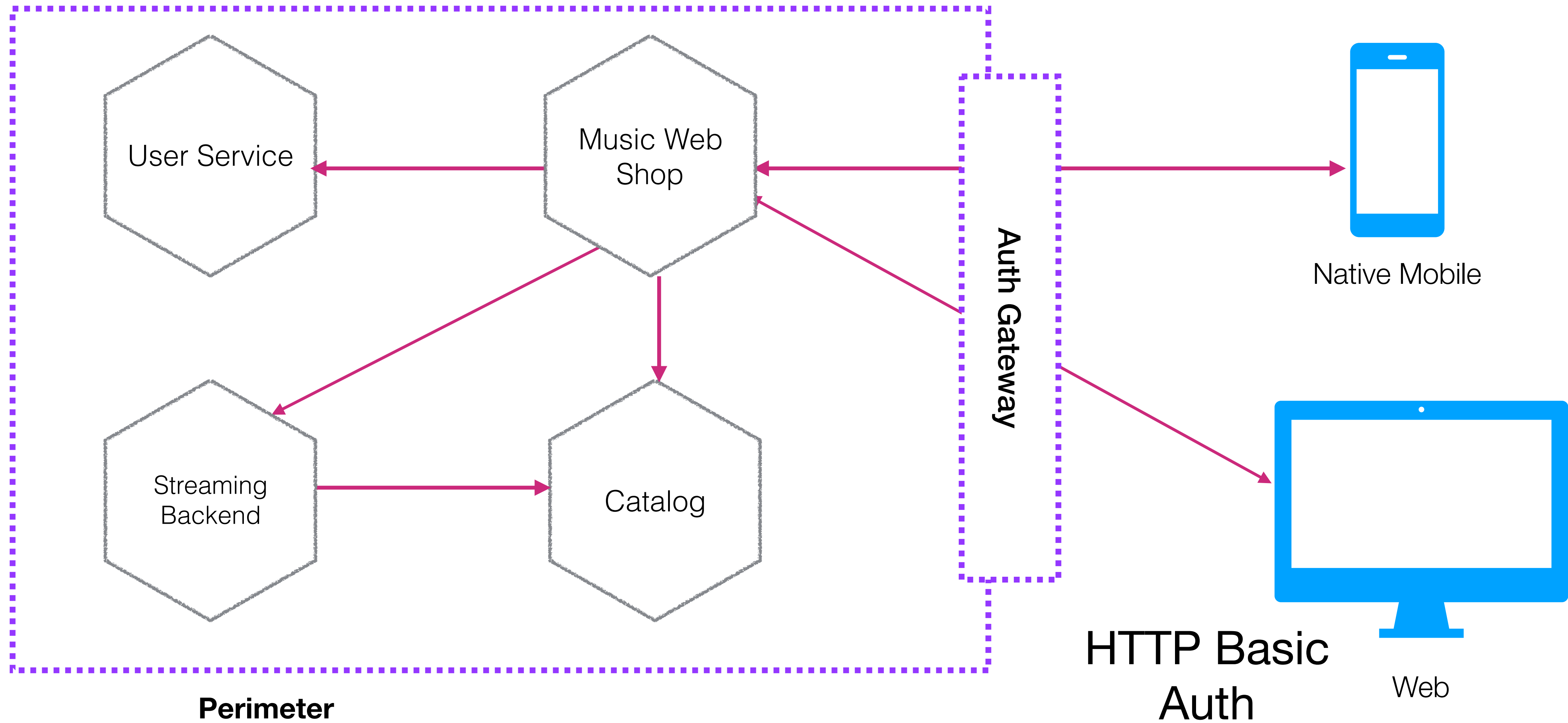




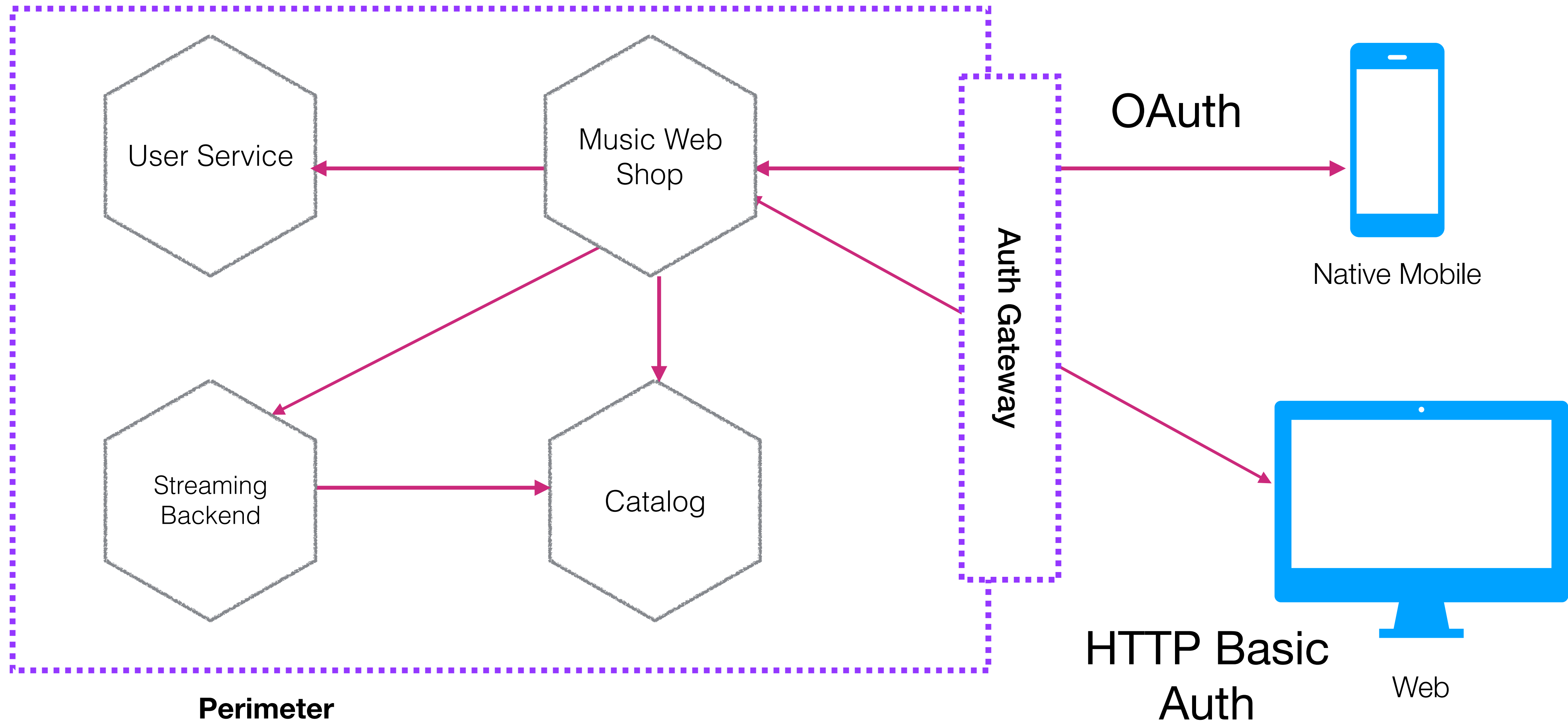
# USER AUTHENTICATION - PROXY-BASED



# USER AUTHENTICATION - PROXY-BASED

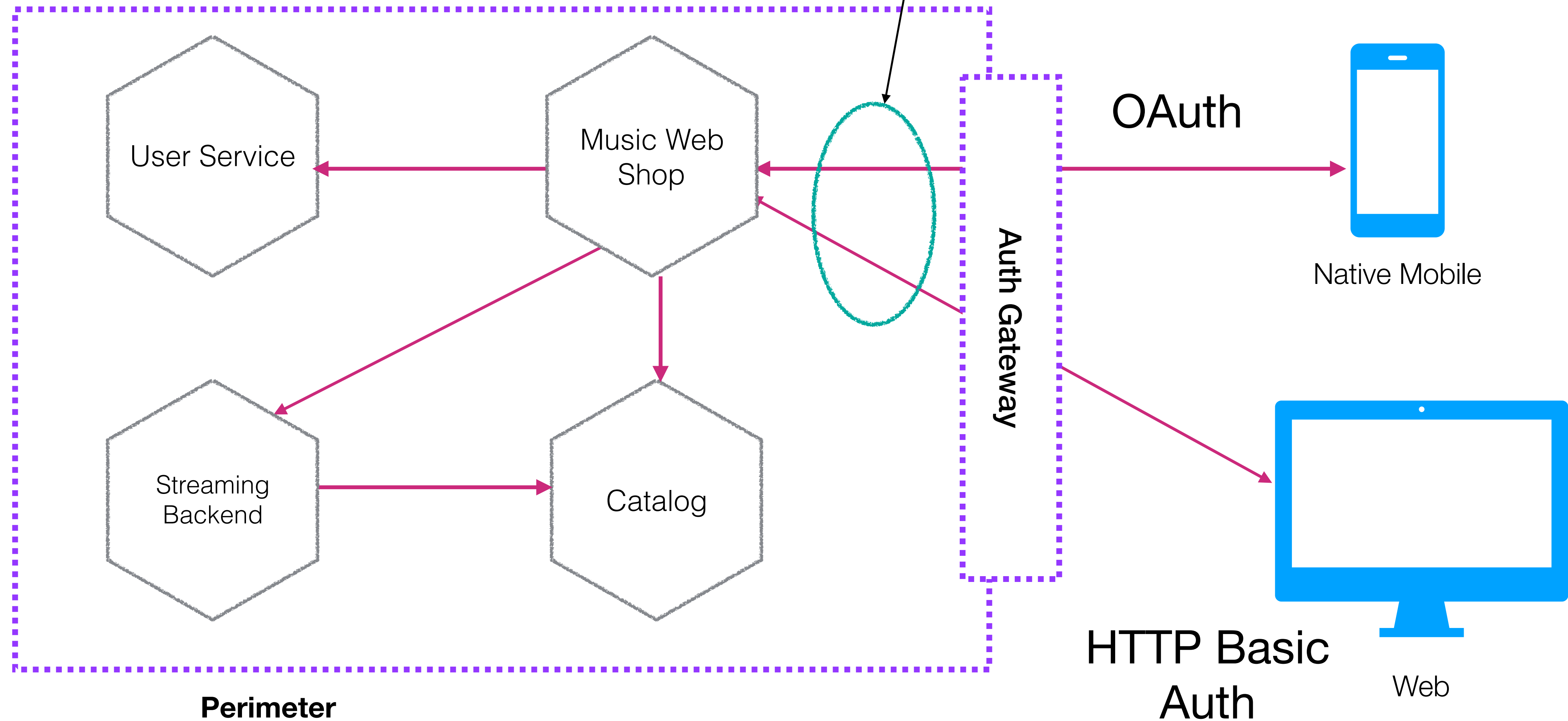


# USER AUTHENTICATION - PROXY-BASED



# USER AUTHENTICATION - PROXY-BASED

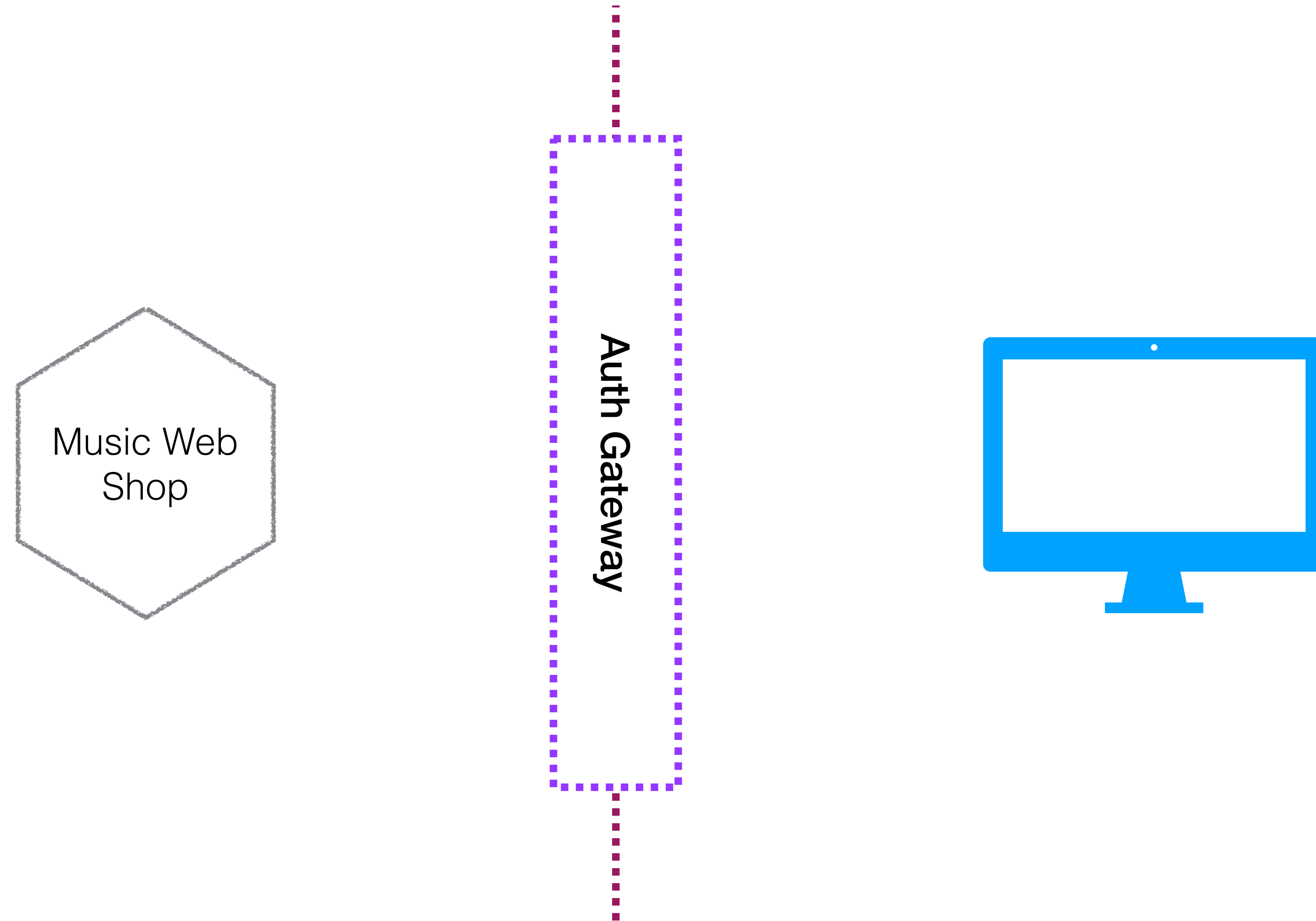
Only authenticated  
users



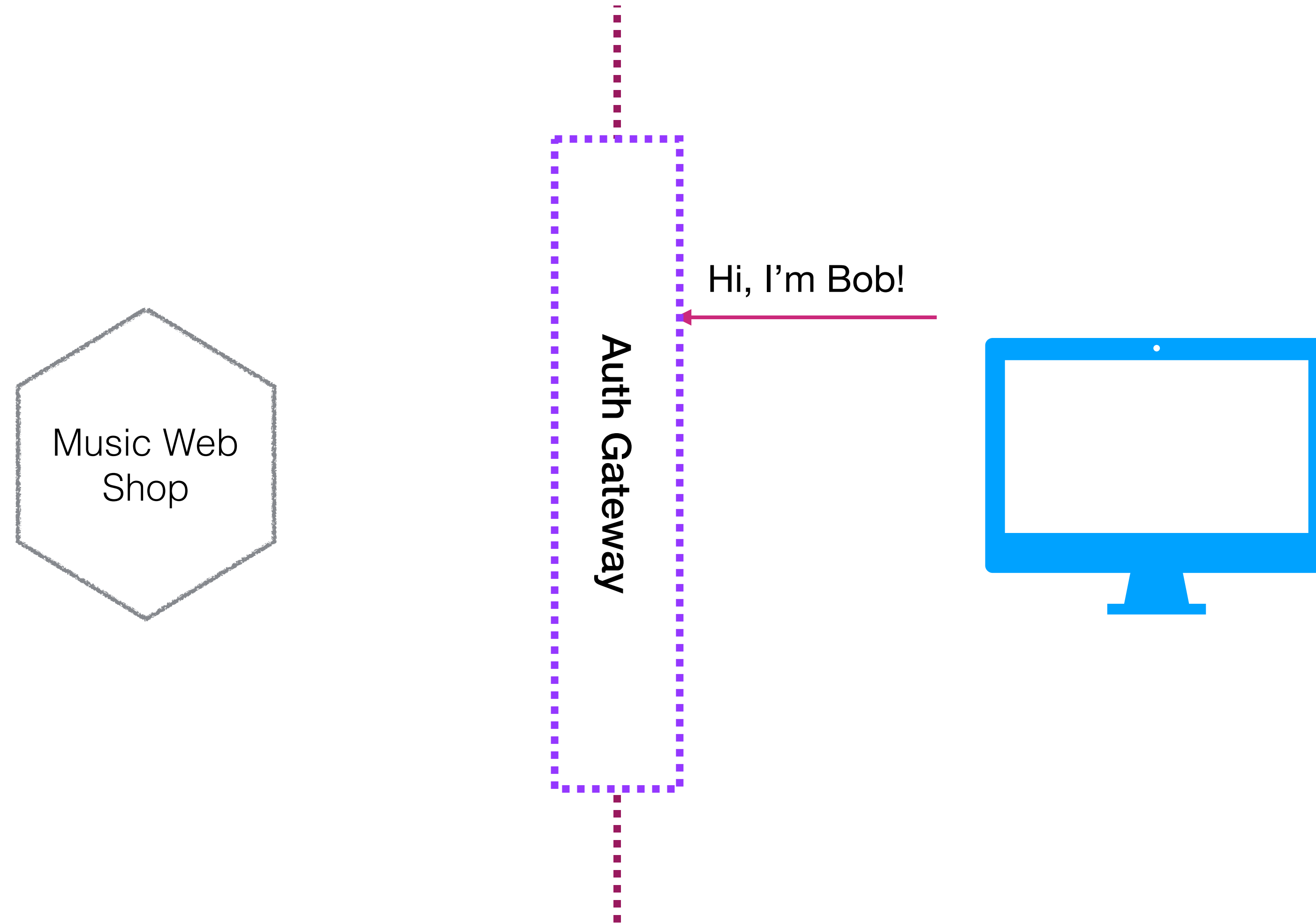
# What about authorisation?



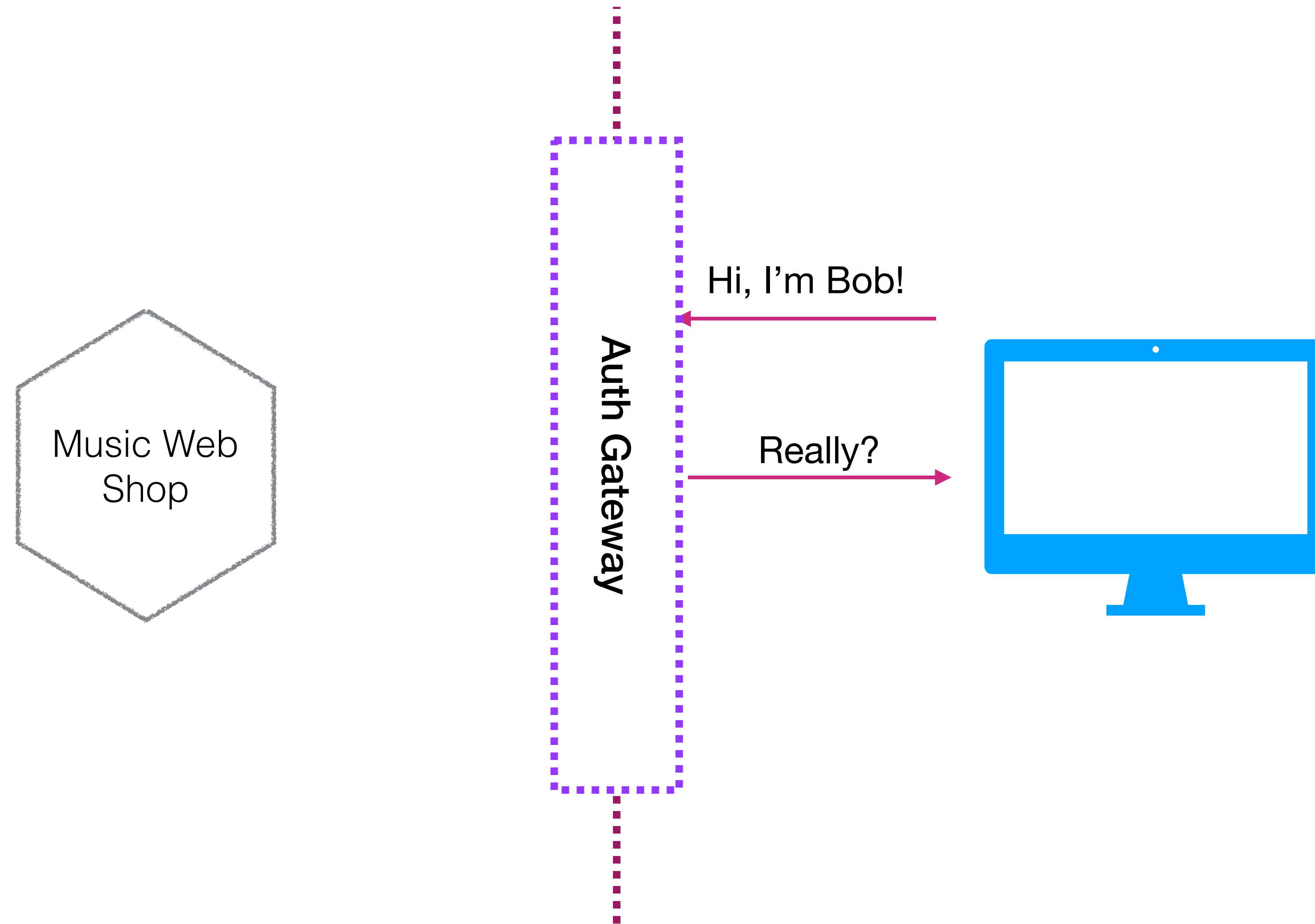
# DO YOU EVEN AUTH?



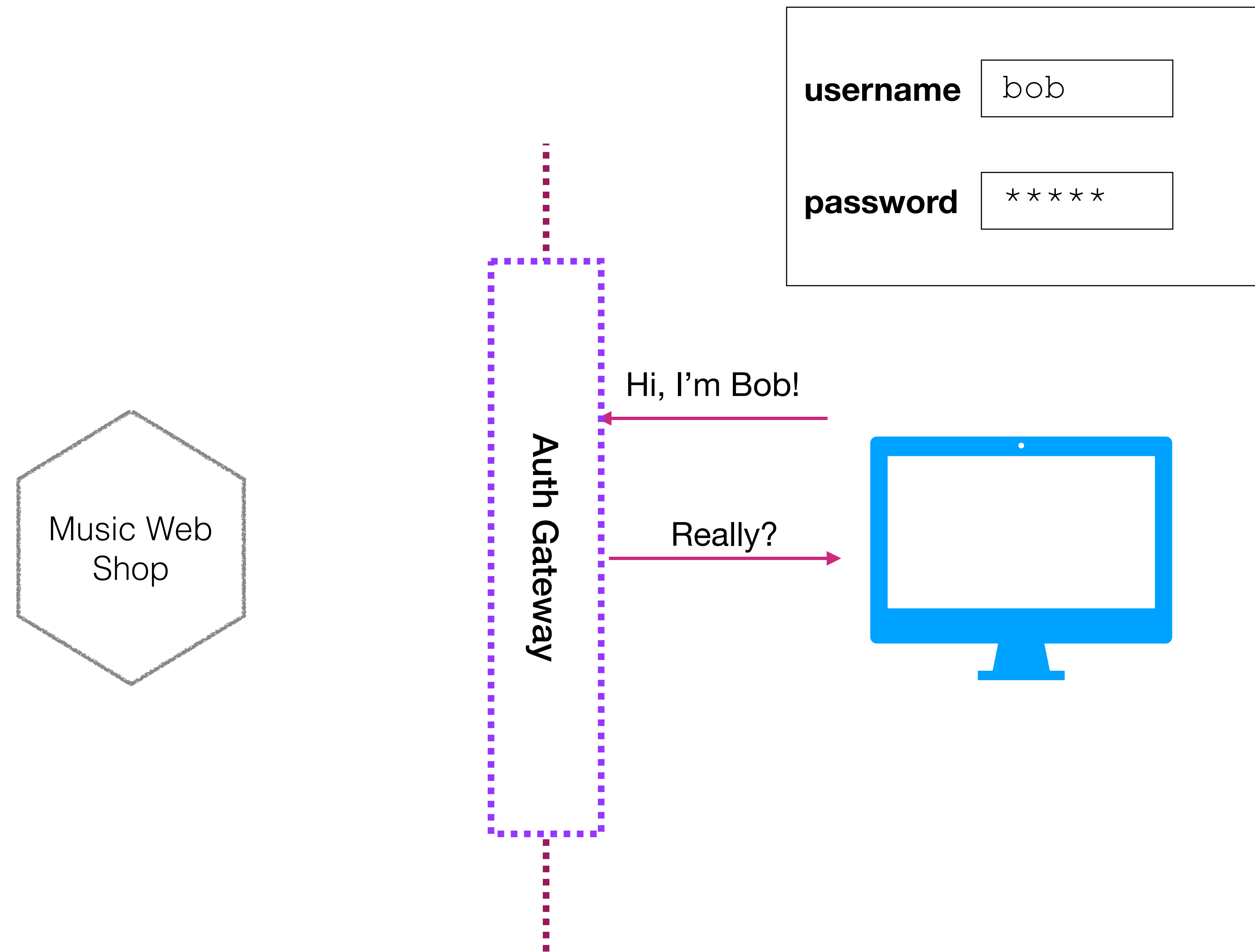
# DO YOU EVEN AUTH?



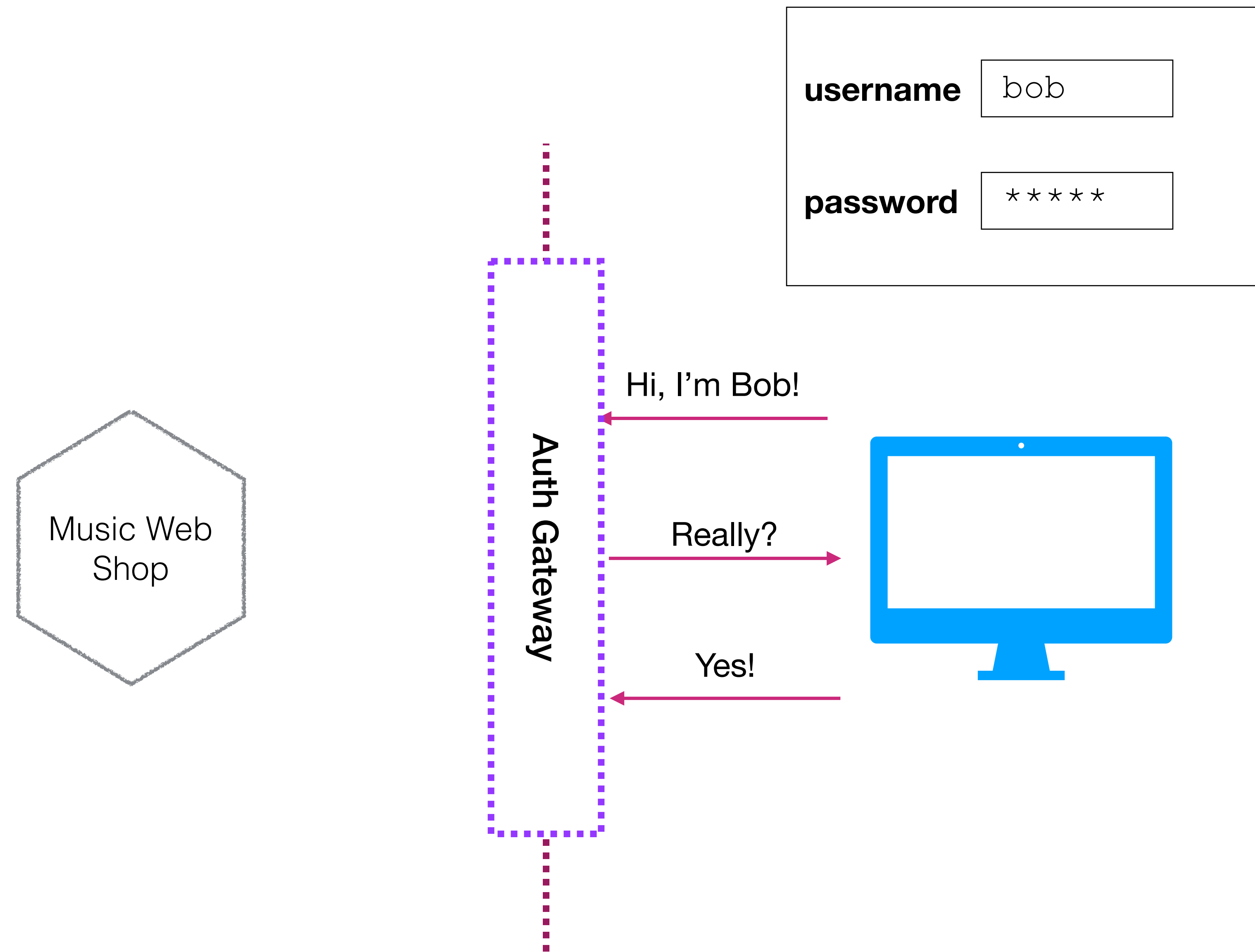
# DO YOU EVEN AUTH?



# DO YOU EVEN AUTH?

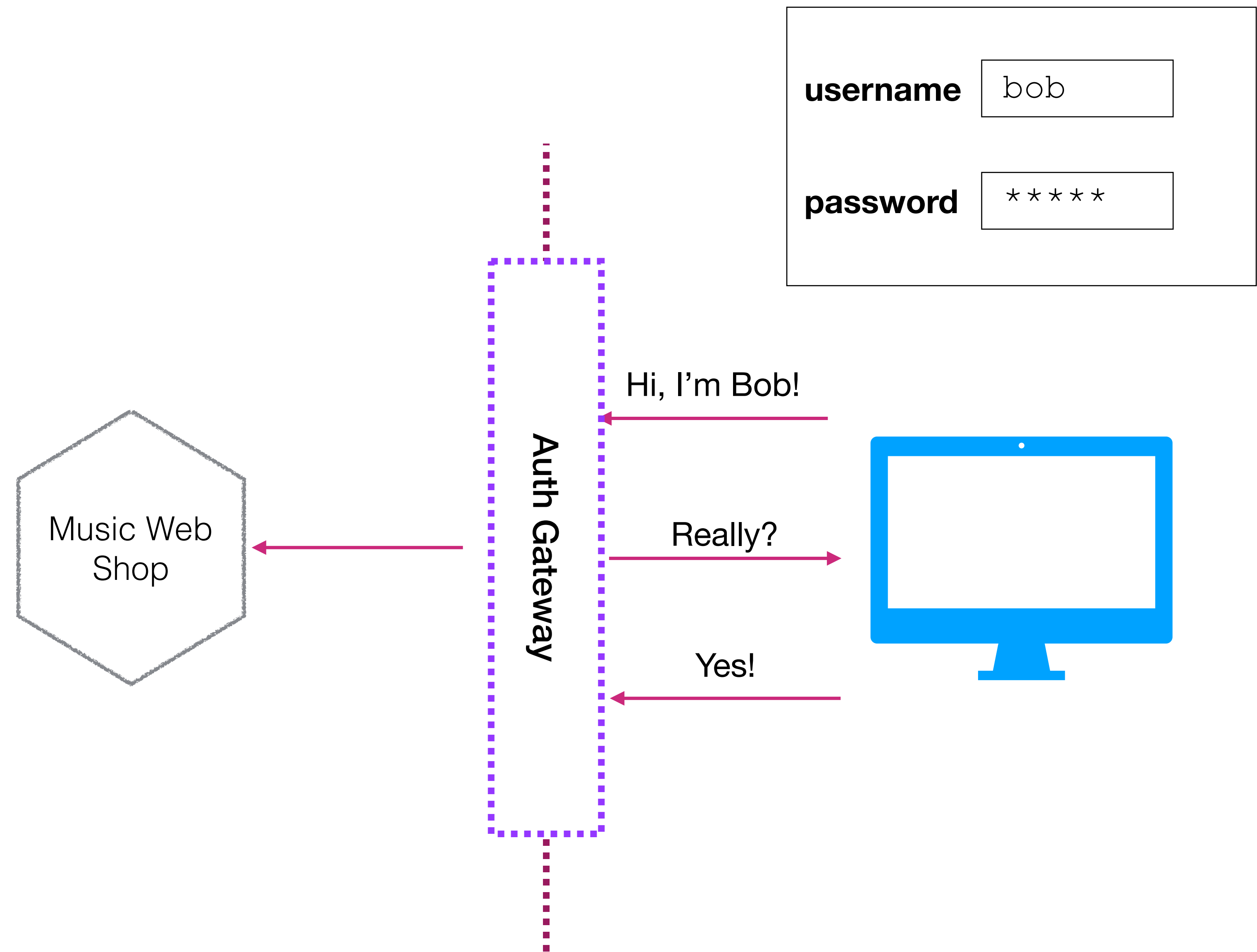


# DO YOU EVEN AUTH?

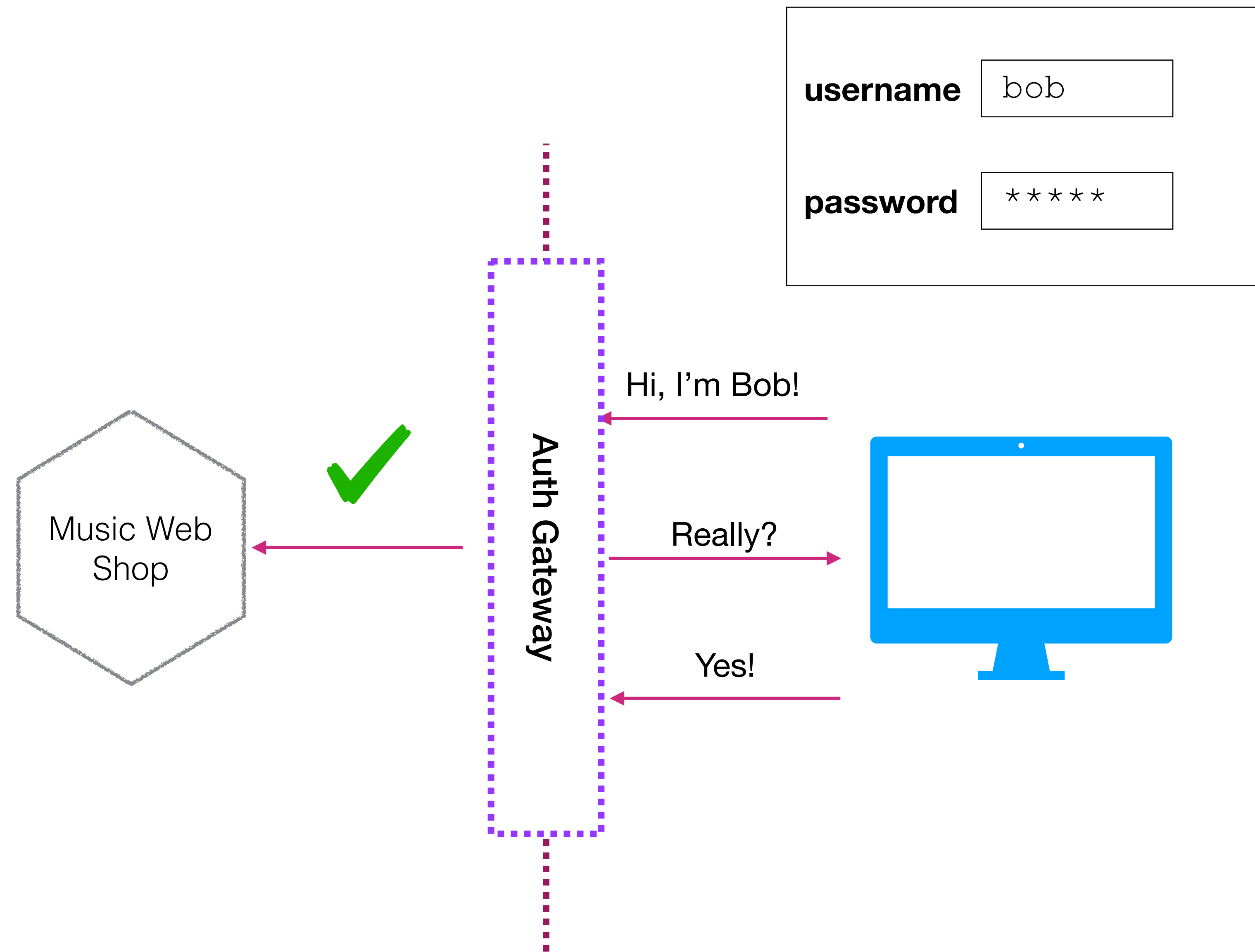




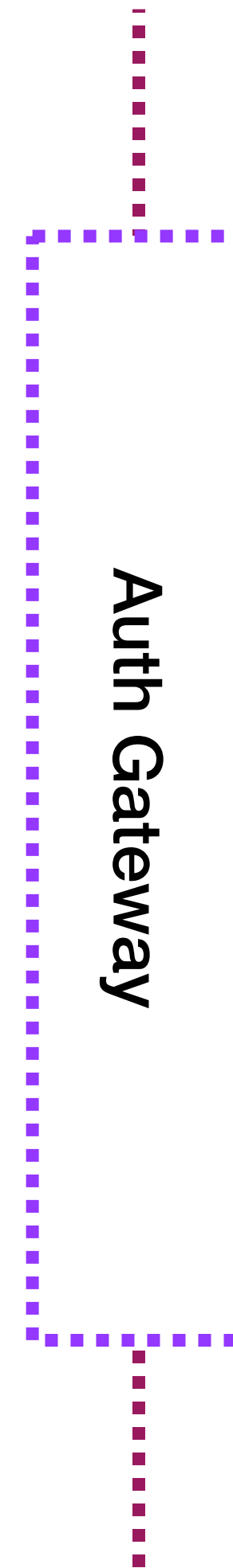
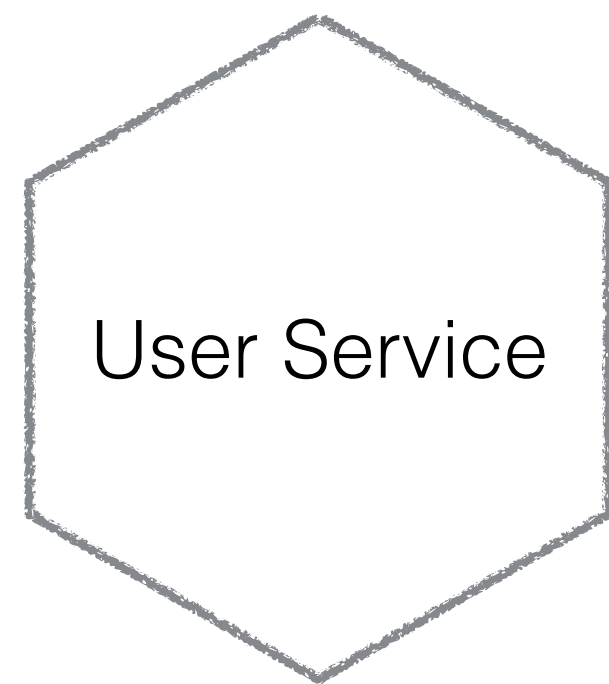
# DO YOU EVEN AUTH?



# DO YOU EVEN AUTH?

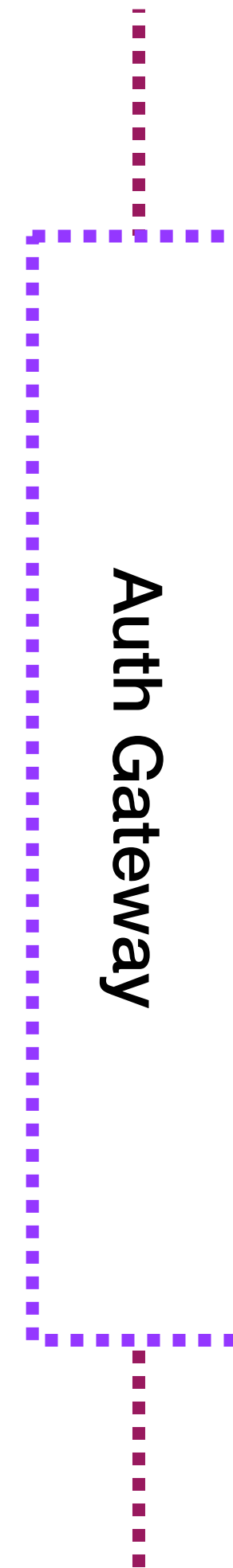
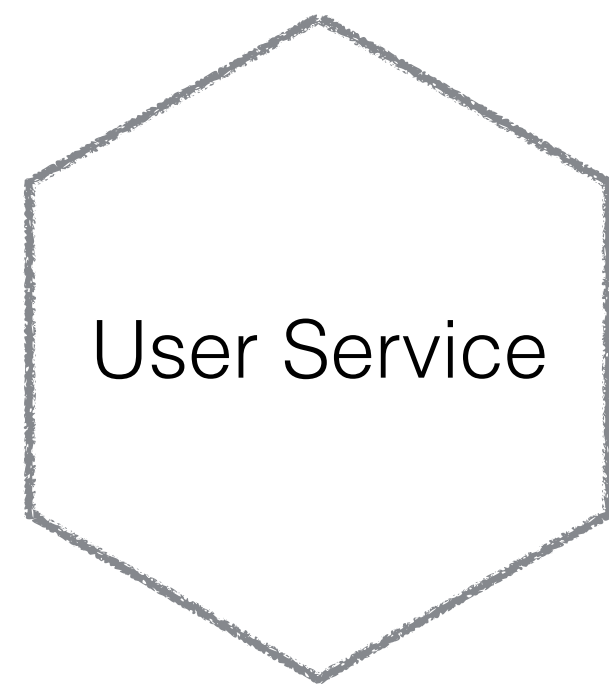


## DOWNSTREAM AUTH - IMPLICIT TRUST?



**Logged in as Bob**

## DOWNSTREAM AUTH - IMPLICIT TRUST?

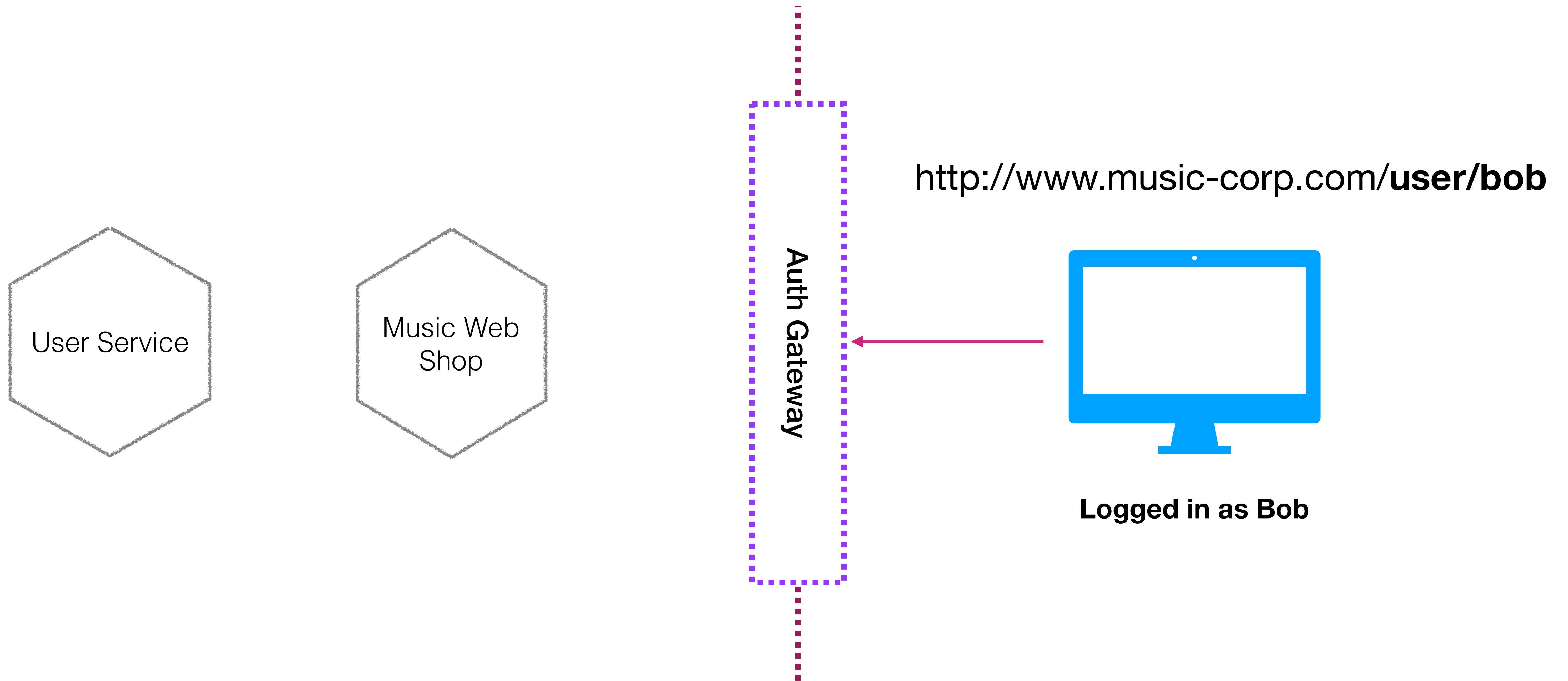


<http://www.music-corp.com/user/bob>



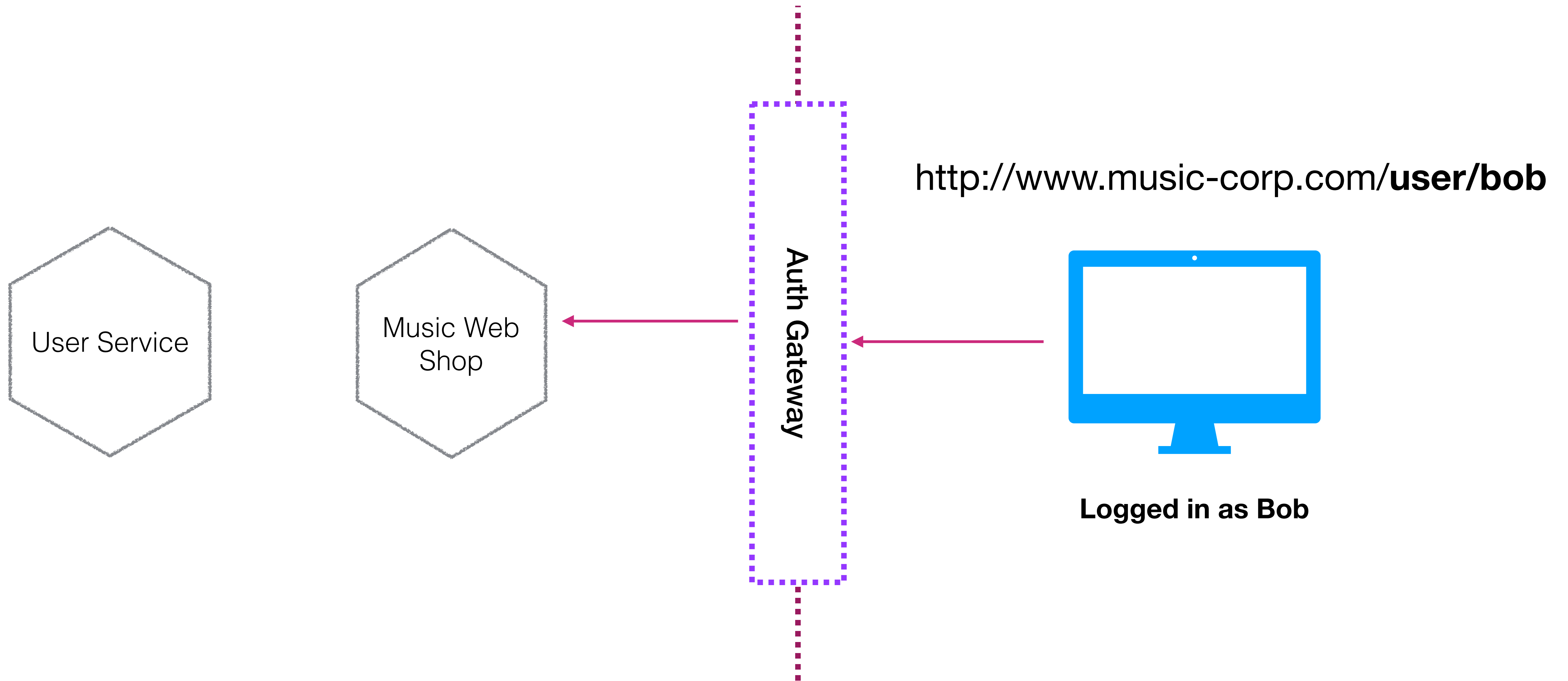
**Logged in as Bob**

## DOWNSTREAM AUTH - IMPLICIT TRUST?

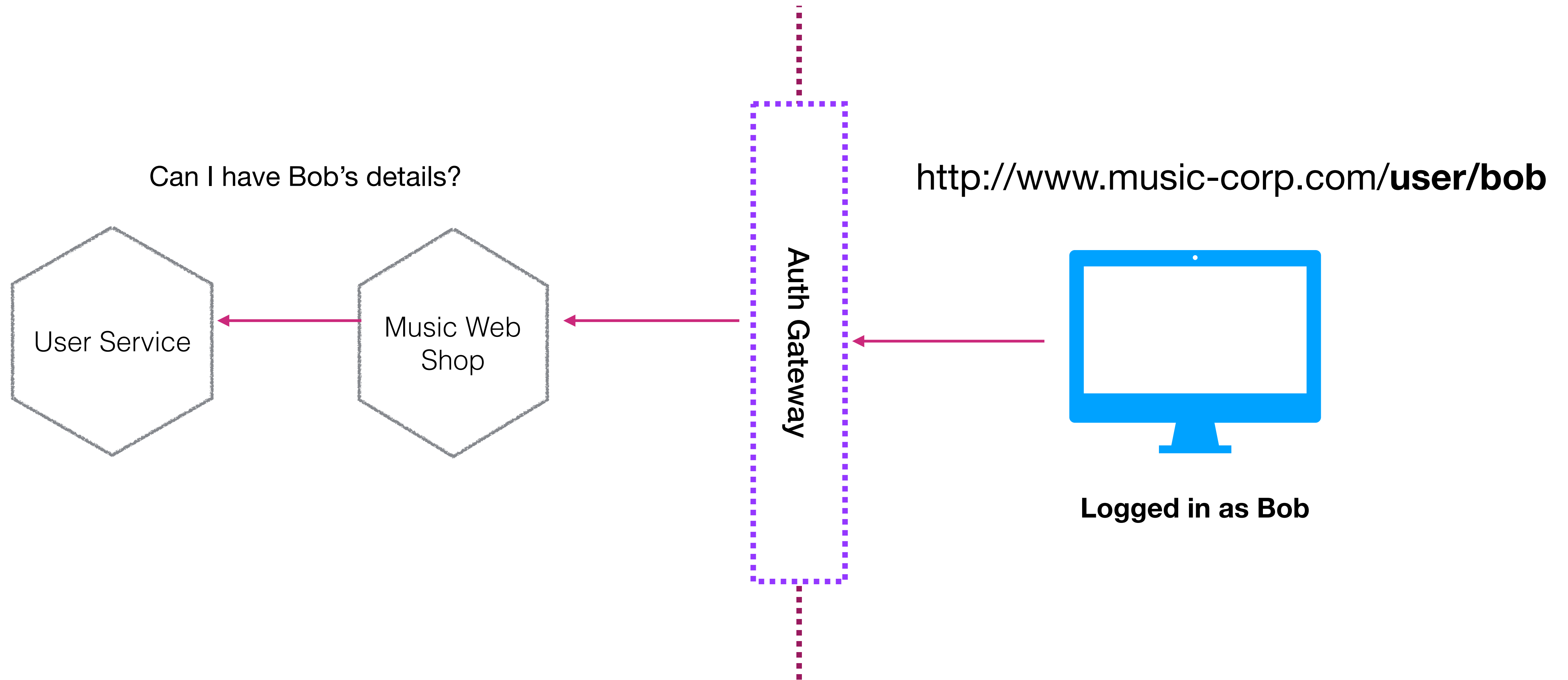




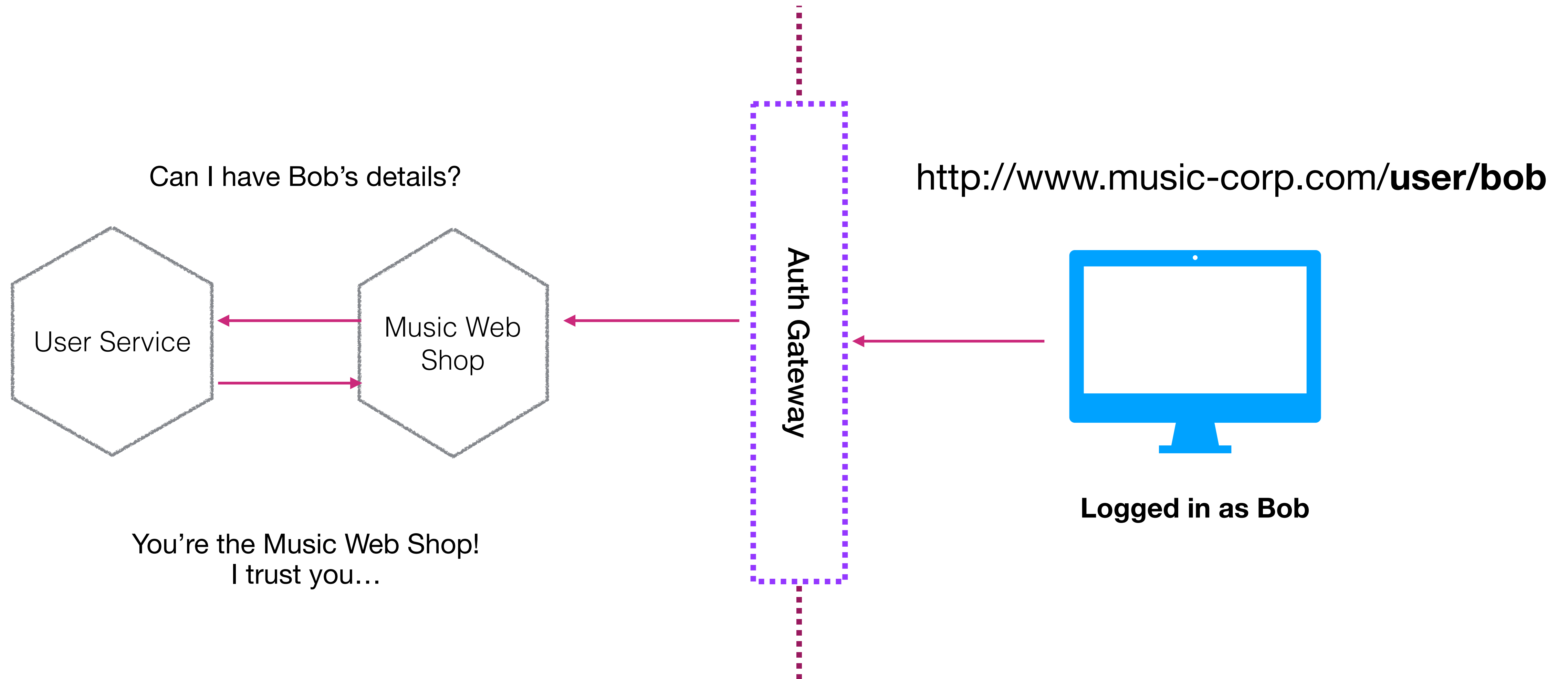
## DOWNSTREAM AUTH - IMPLICIT TRUST?



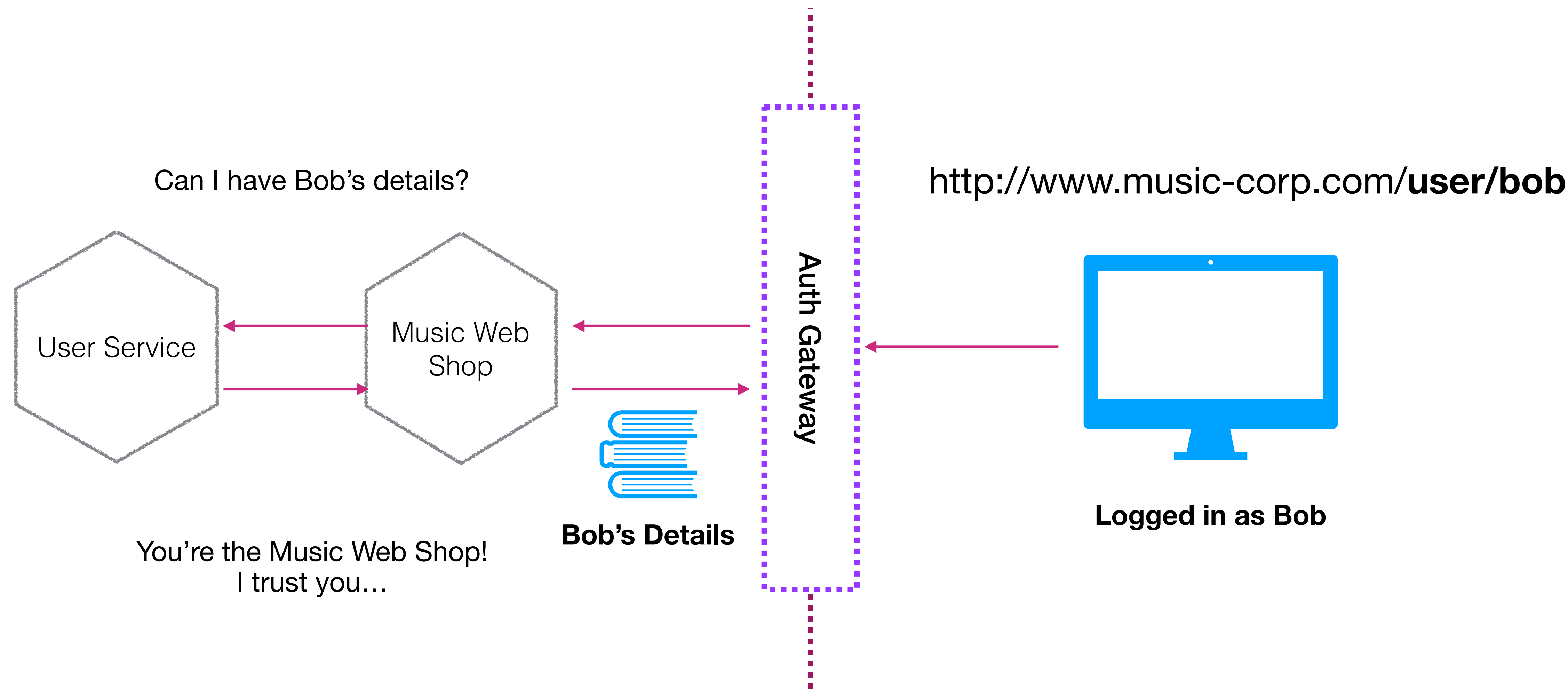
## DOWNSTREAM AUTH - IMPLICIT TRUST?



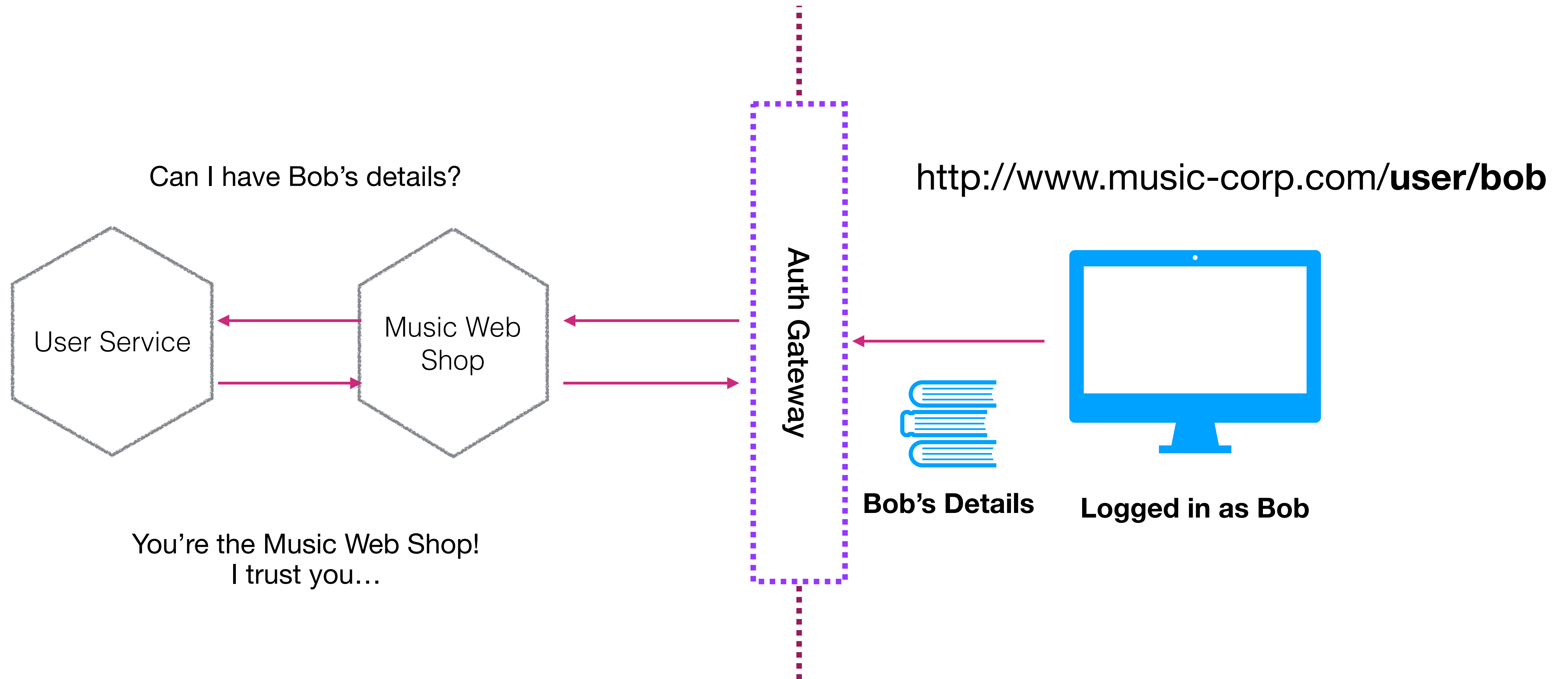
## DOWNSTREAM AUTH - IMPLICIT TRUST?



# DOWNSTREAM AUTH - IMPLICIT TRUST?

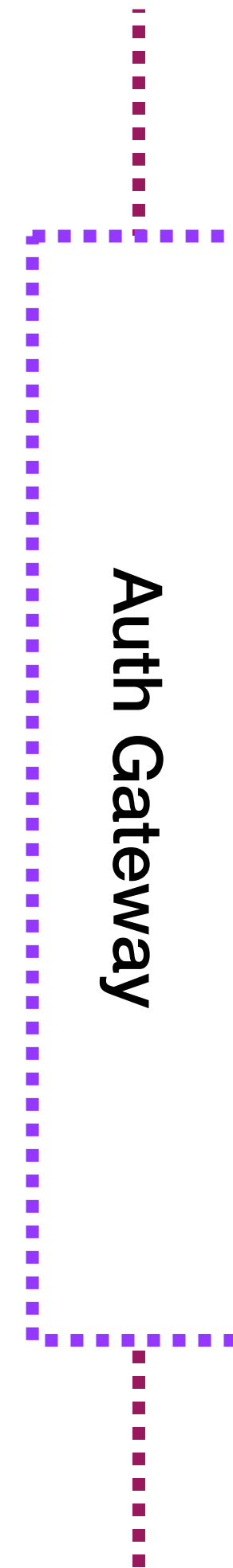
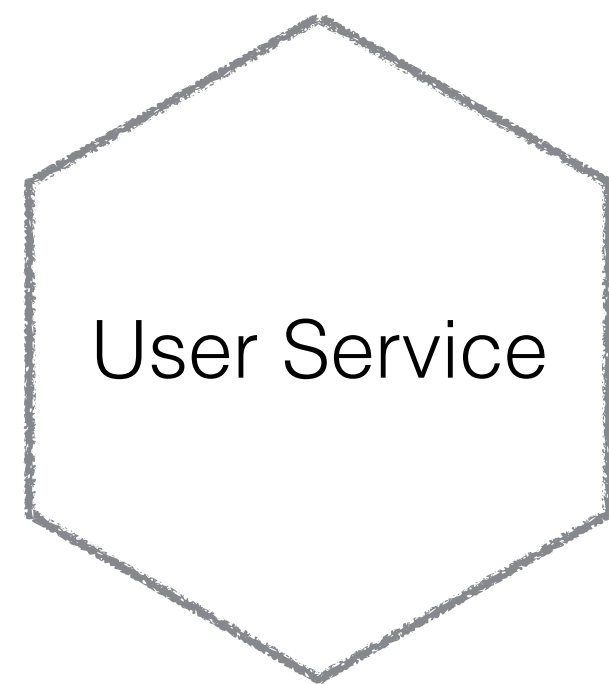


# DOWNSTREAM AUTH - IMPLICIT TRUST?



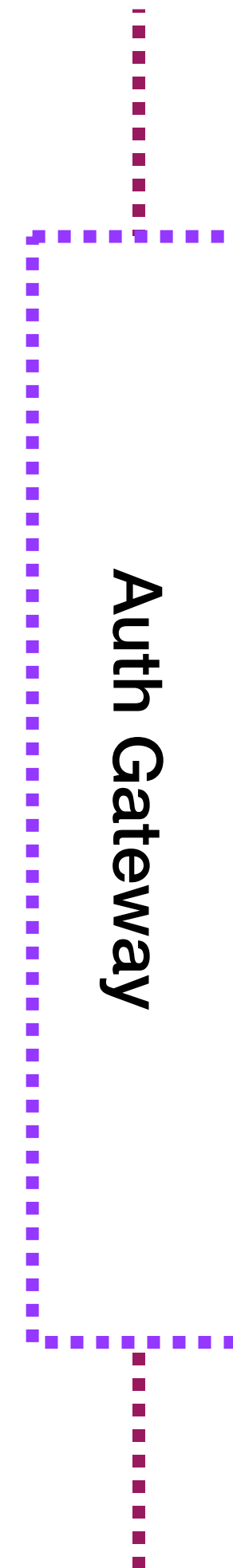
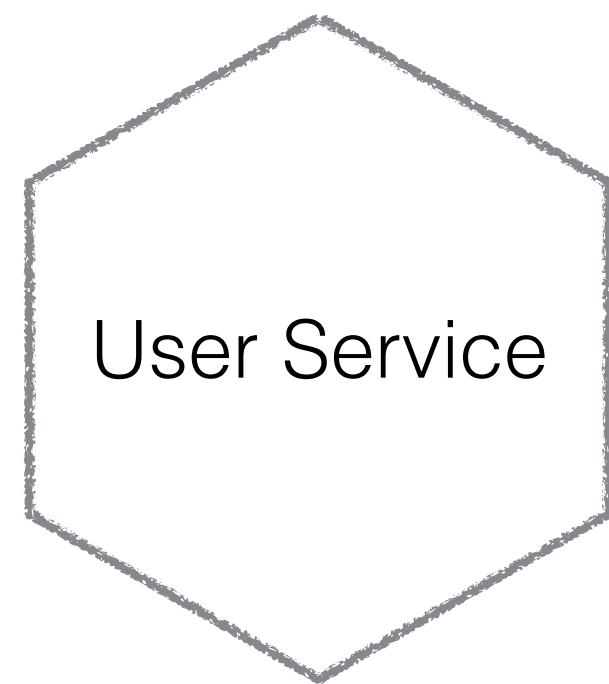


## DOWNSTREAM AUTH - IMPLICIT TRUST?



**Logged in as Bob**

# DOWNSTREAM AUTH - IMPLICIT TRUST?

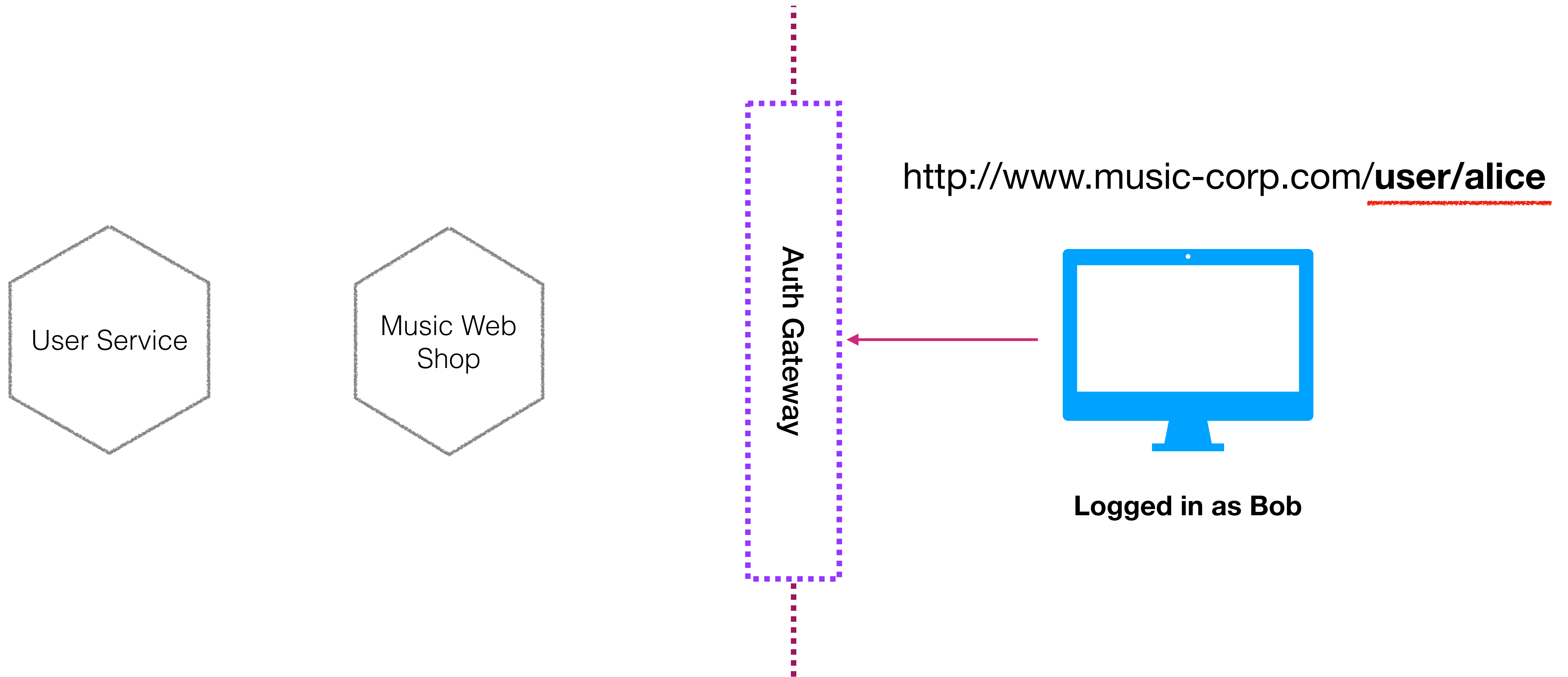


<http://www.music-corp.com/user/alice>

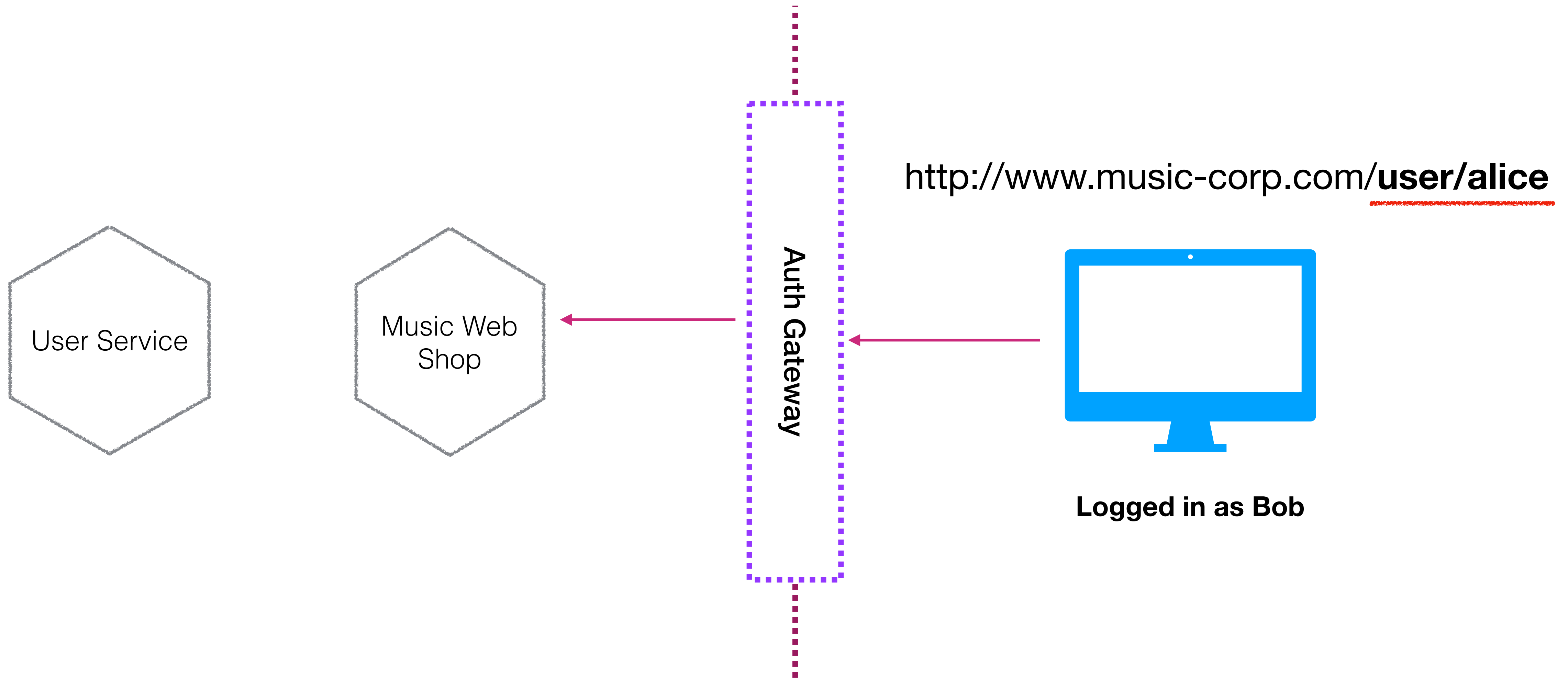


Logged in as Bob

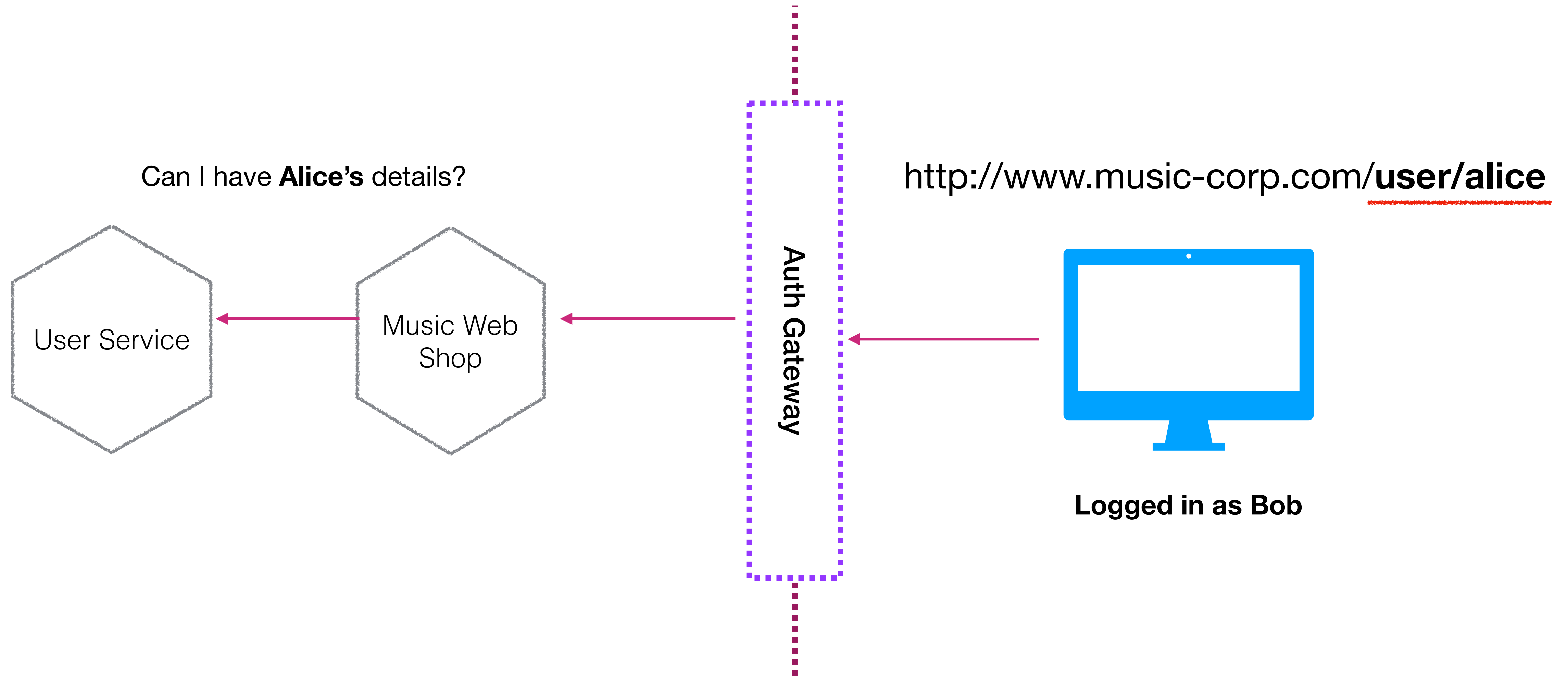
## DOWNSTREAM AUTH - IMPLICIT TRUST?



## DOWNSTREAM AUTH - IMPLICIT TRUST?

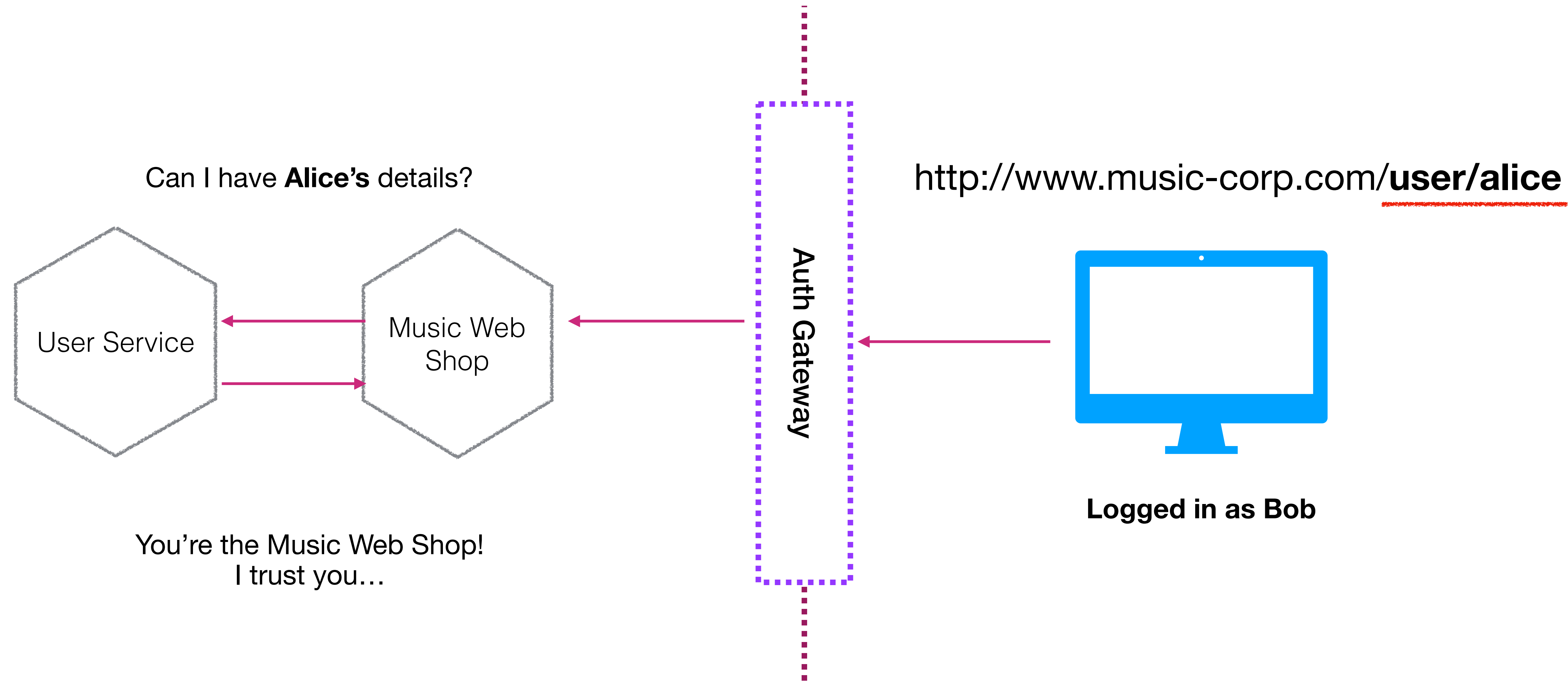


## DOWNSTREAM AUTH - IMPLICIT TRUST?

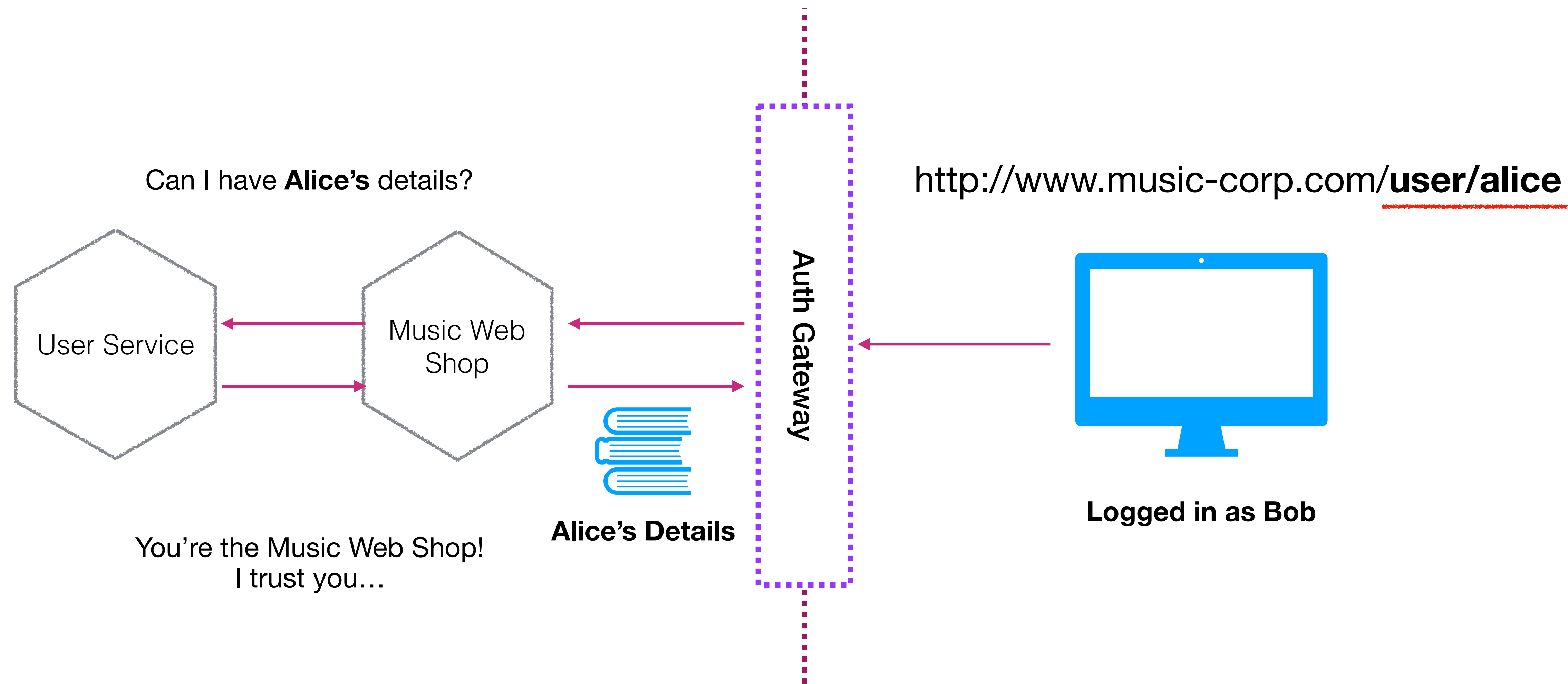




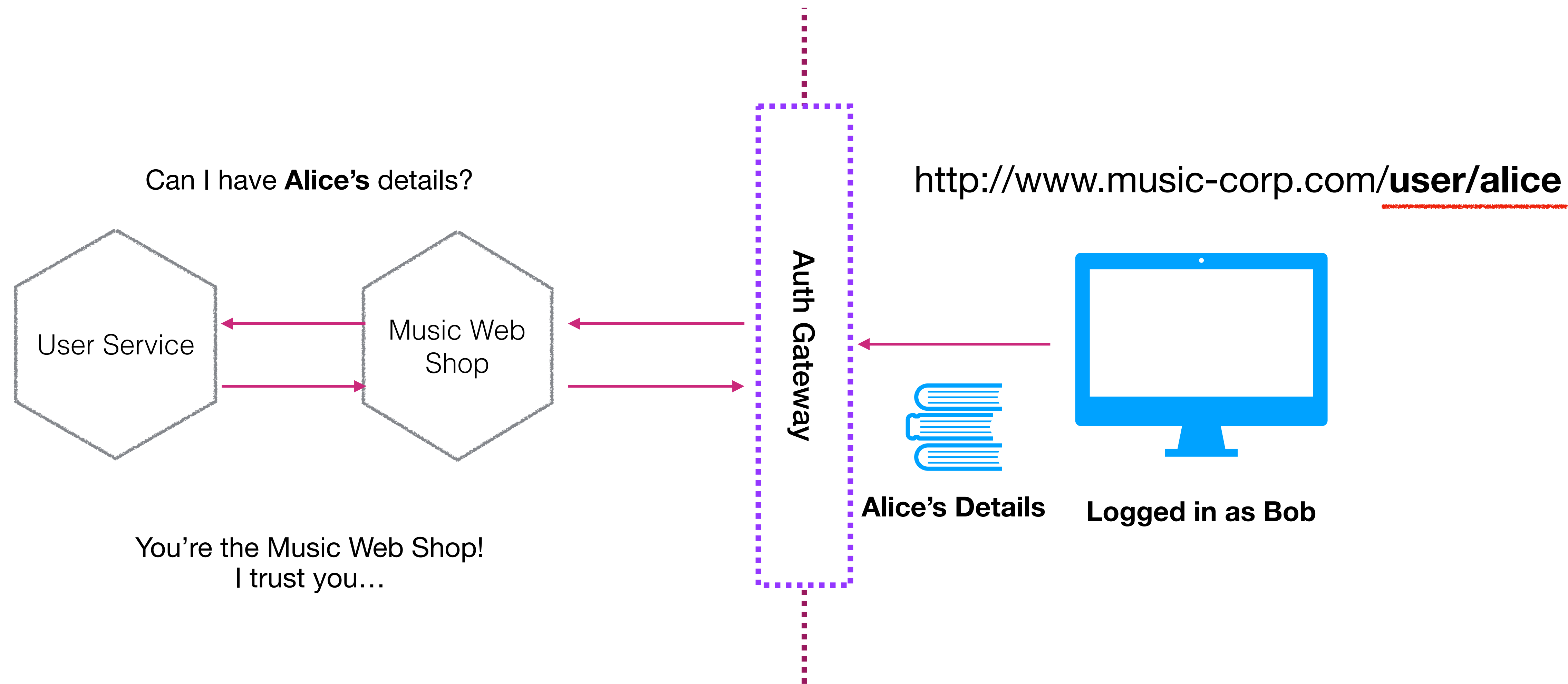
# DOWNSTREAM AUTH - IMPLICIT TRUST?



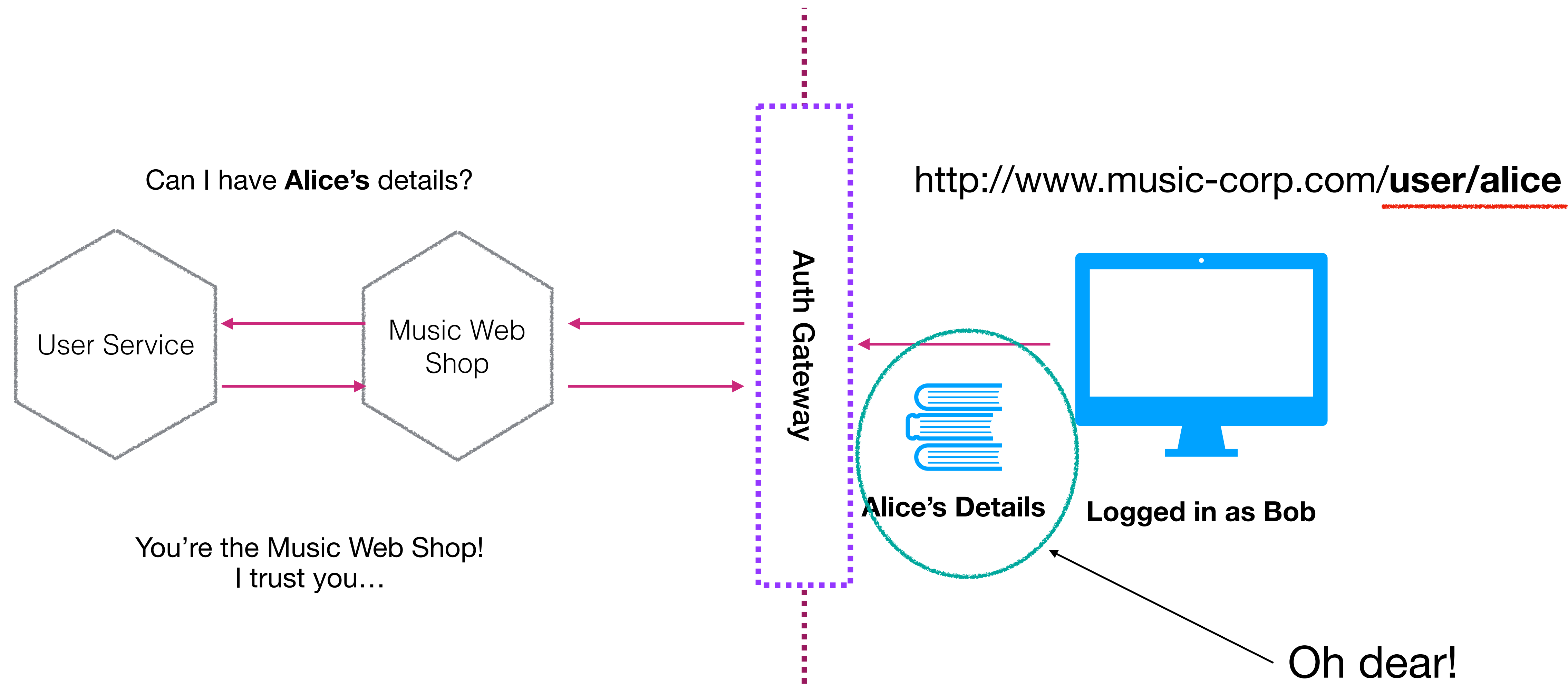
# DOWNSTREAM AUTH - IMPLICIT TRUST?



# DOWNSTREAM AUTH - IMPLICIT TRUST?



# DOWNSTREAM AUTH - IMPLICIT TRUST?





Confused  
Deputy  
Problem!

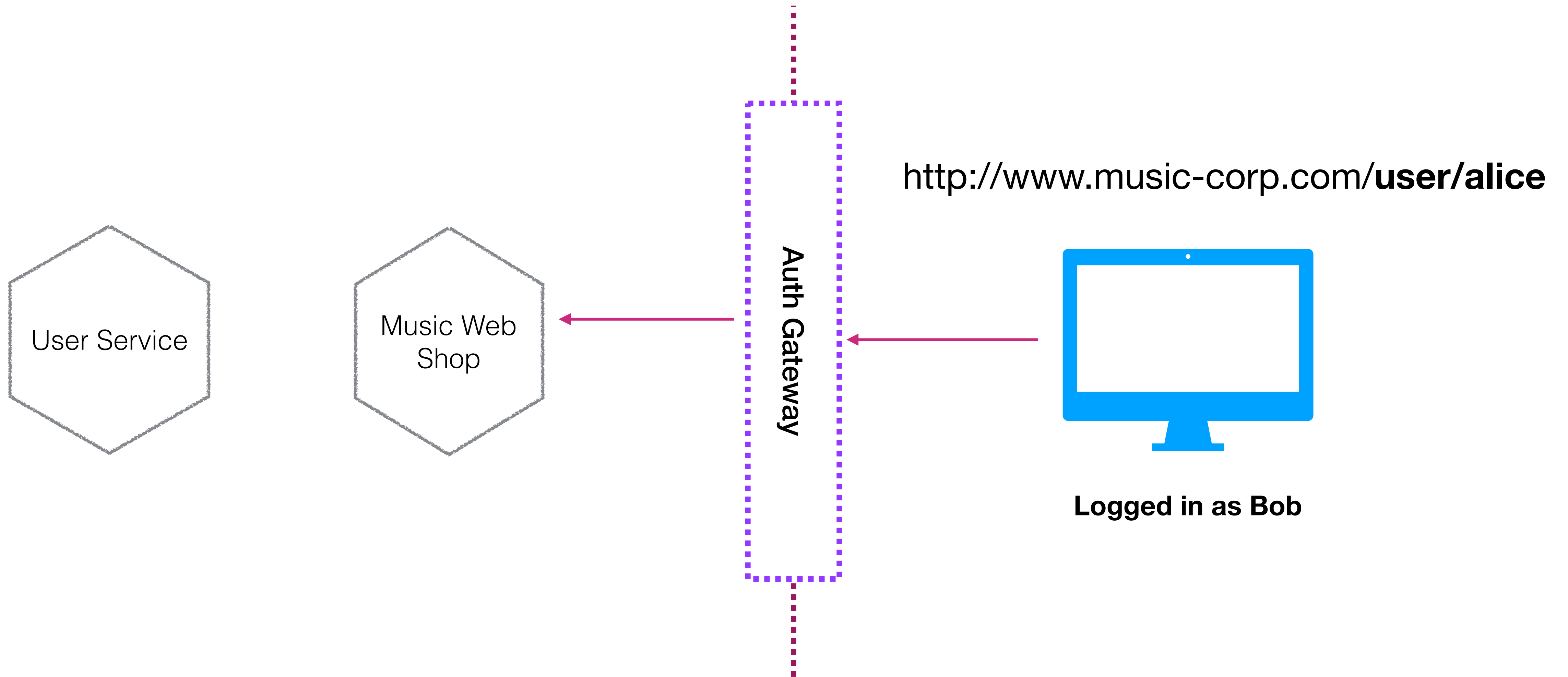


© 2014  
Dave Lundy  
lundyd@dma1.org

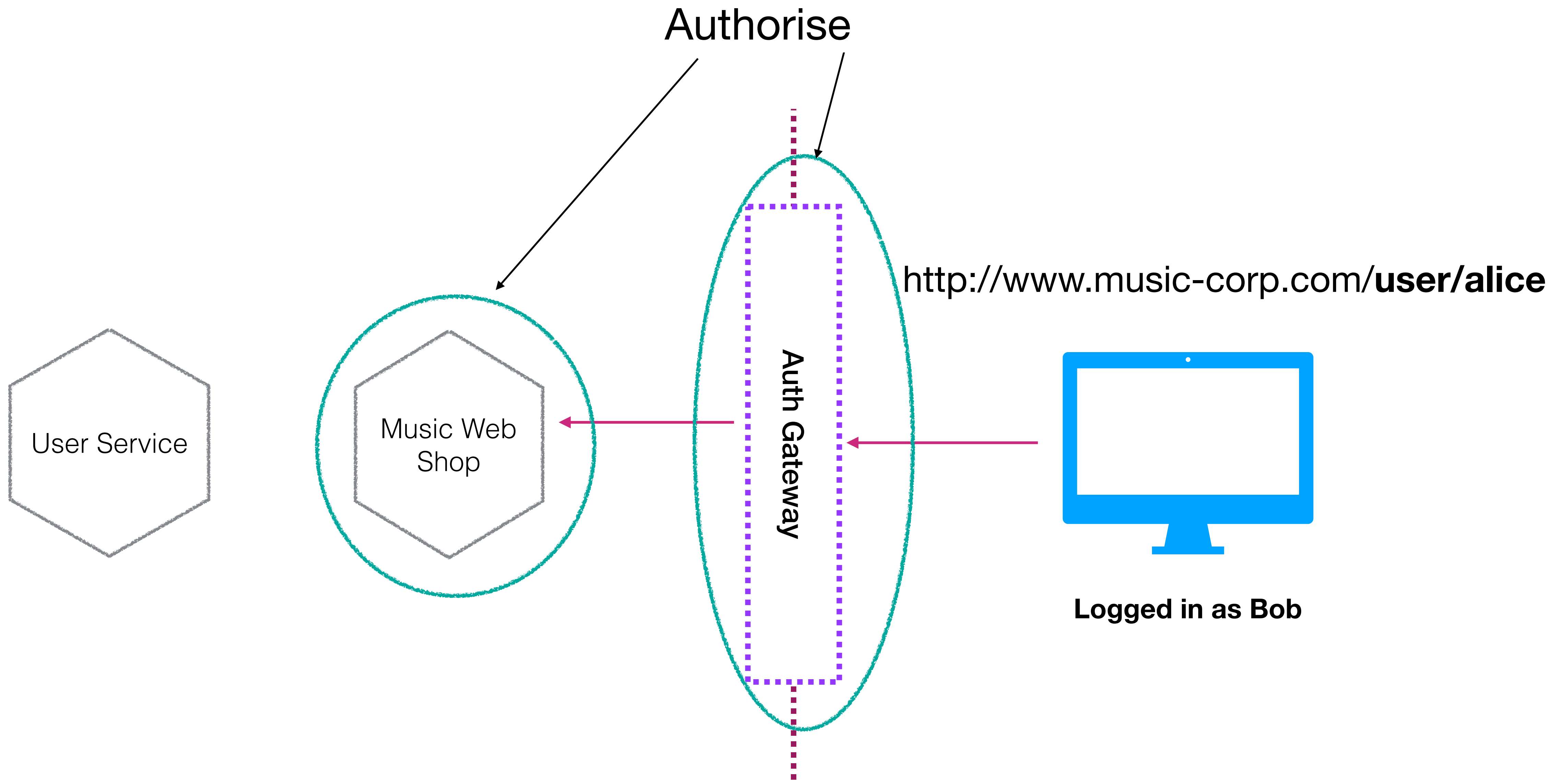
<https://www.flickr.com/photos/lundyd/14481829564>



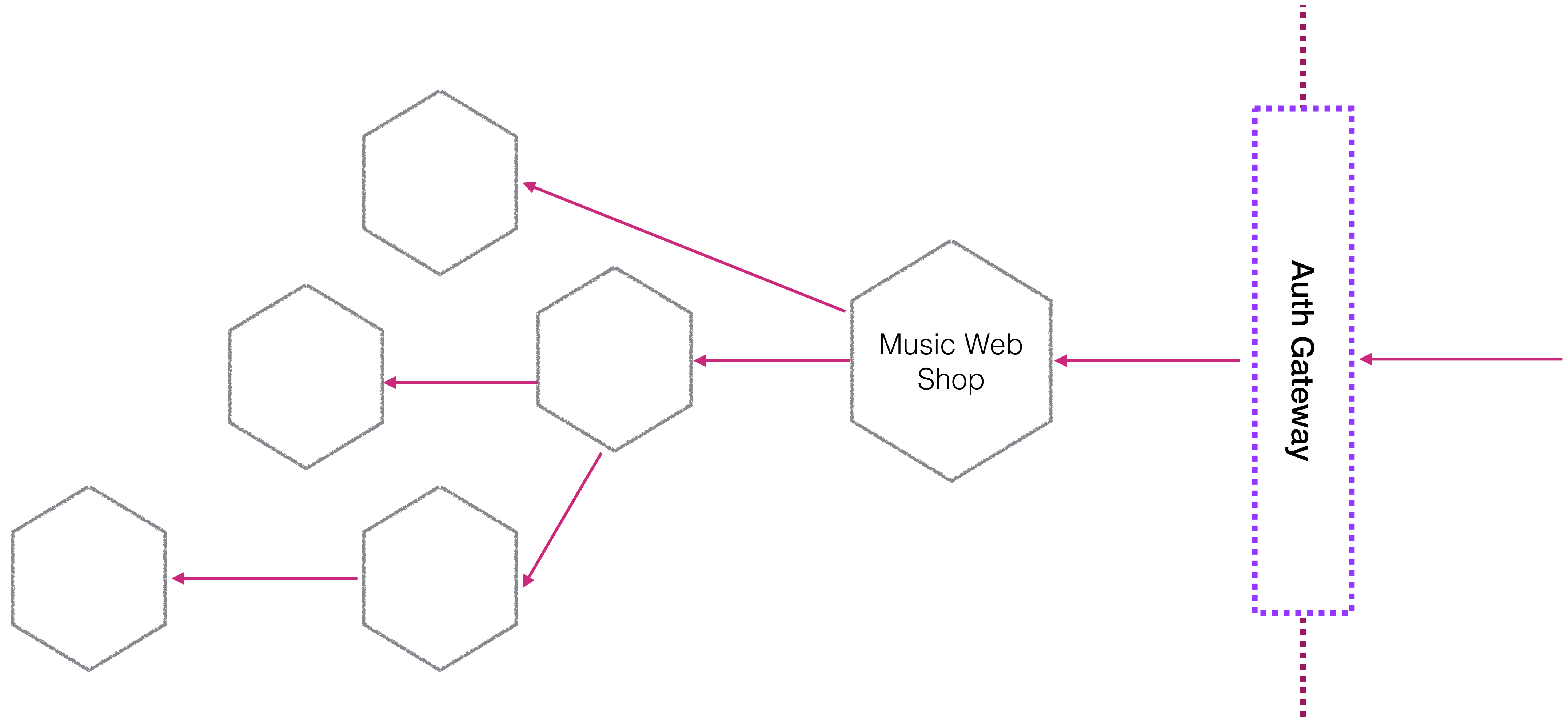
# AUTHORISE UPSTREAM?



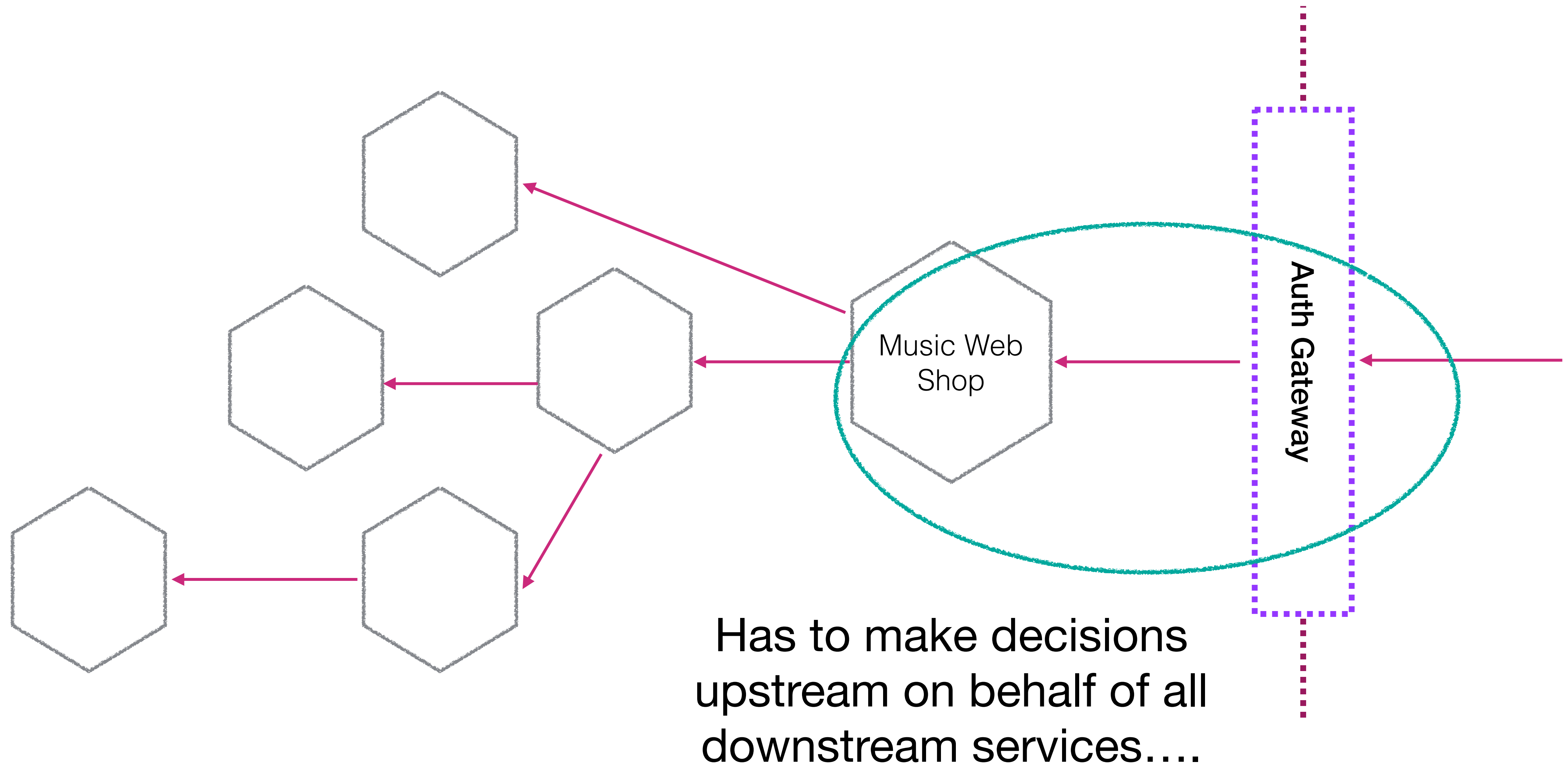
## AUTHORISE UPSTREAM?



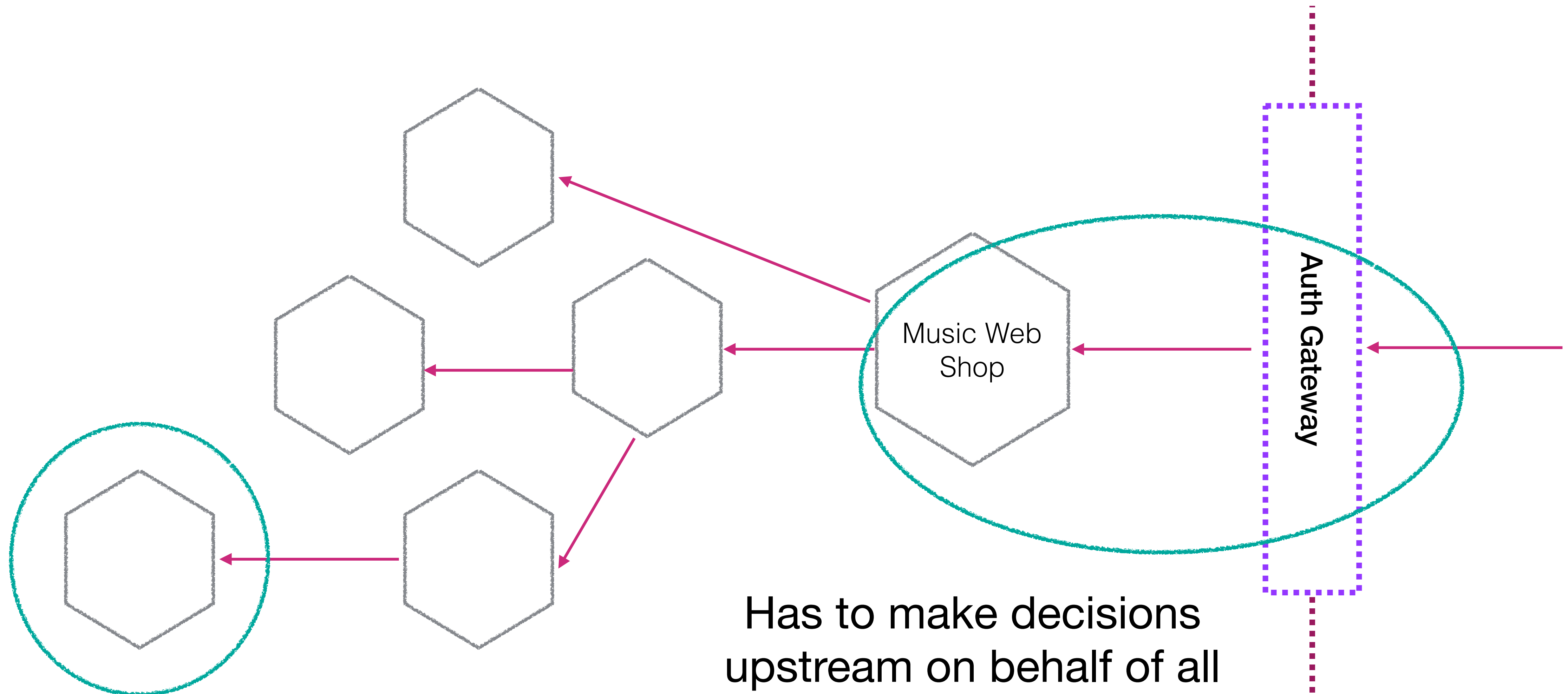
# WHERE DO THE SMARTS LIVE?



## WHERE DO THE SMARTS LIVE?



## WHERE DO THE SMARTS LIVE?

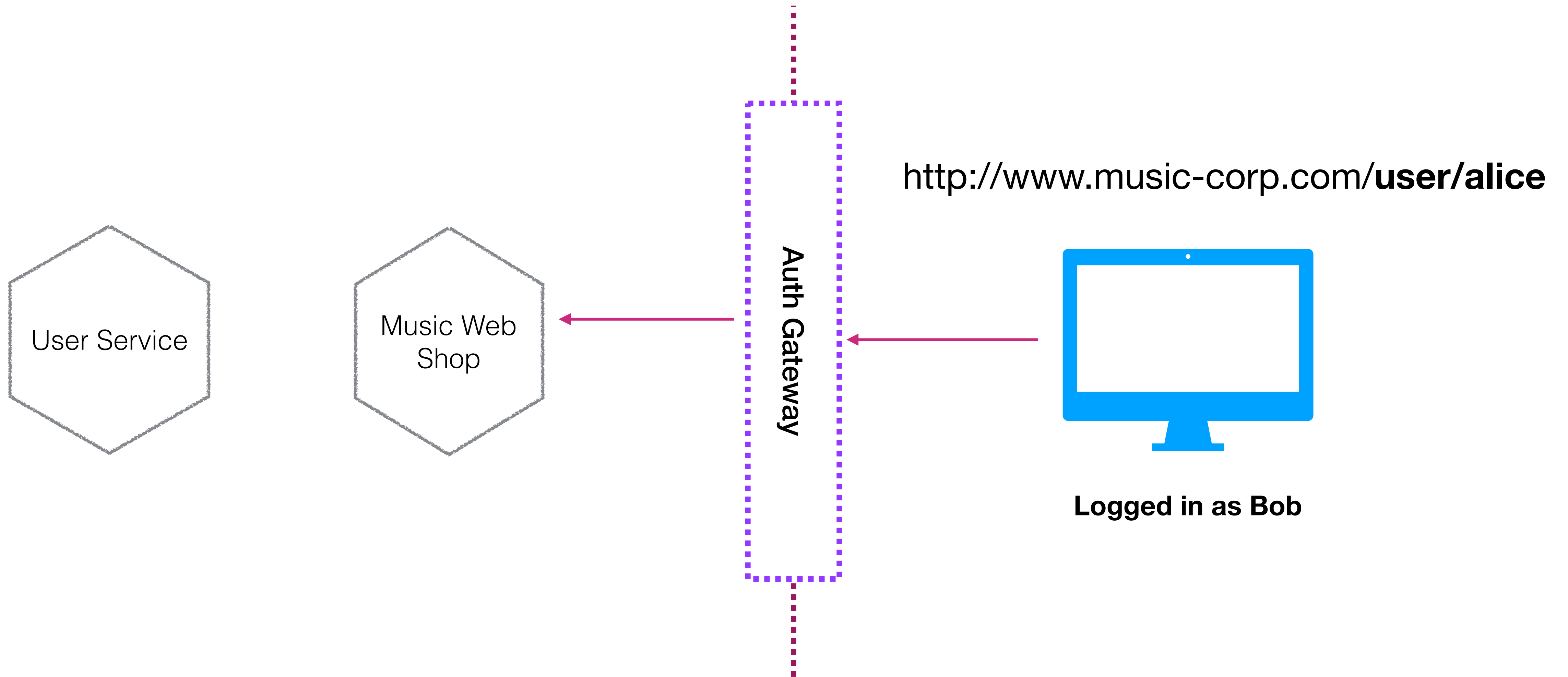


But it can be preferable to push this logic to the service itself

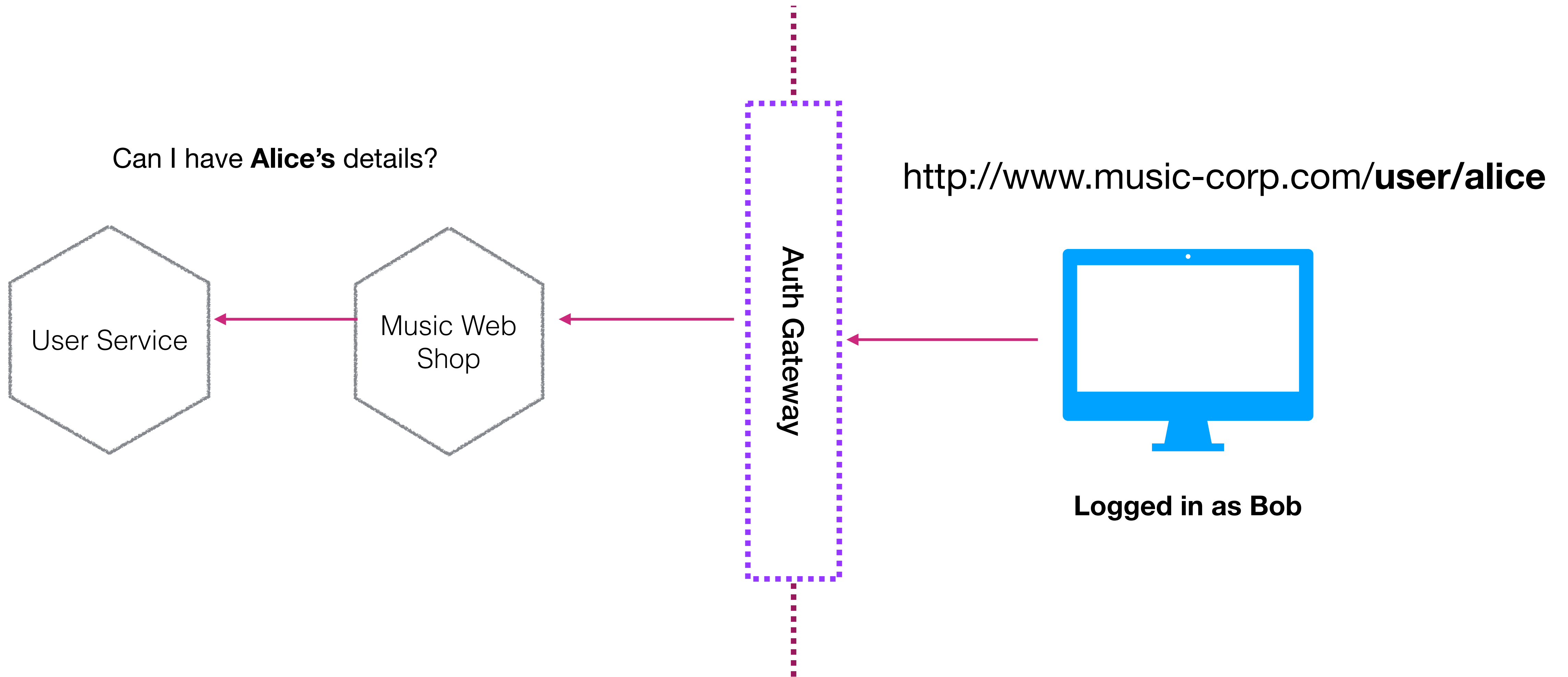
Has to make decisions upstream on behalf of all downstream services....



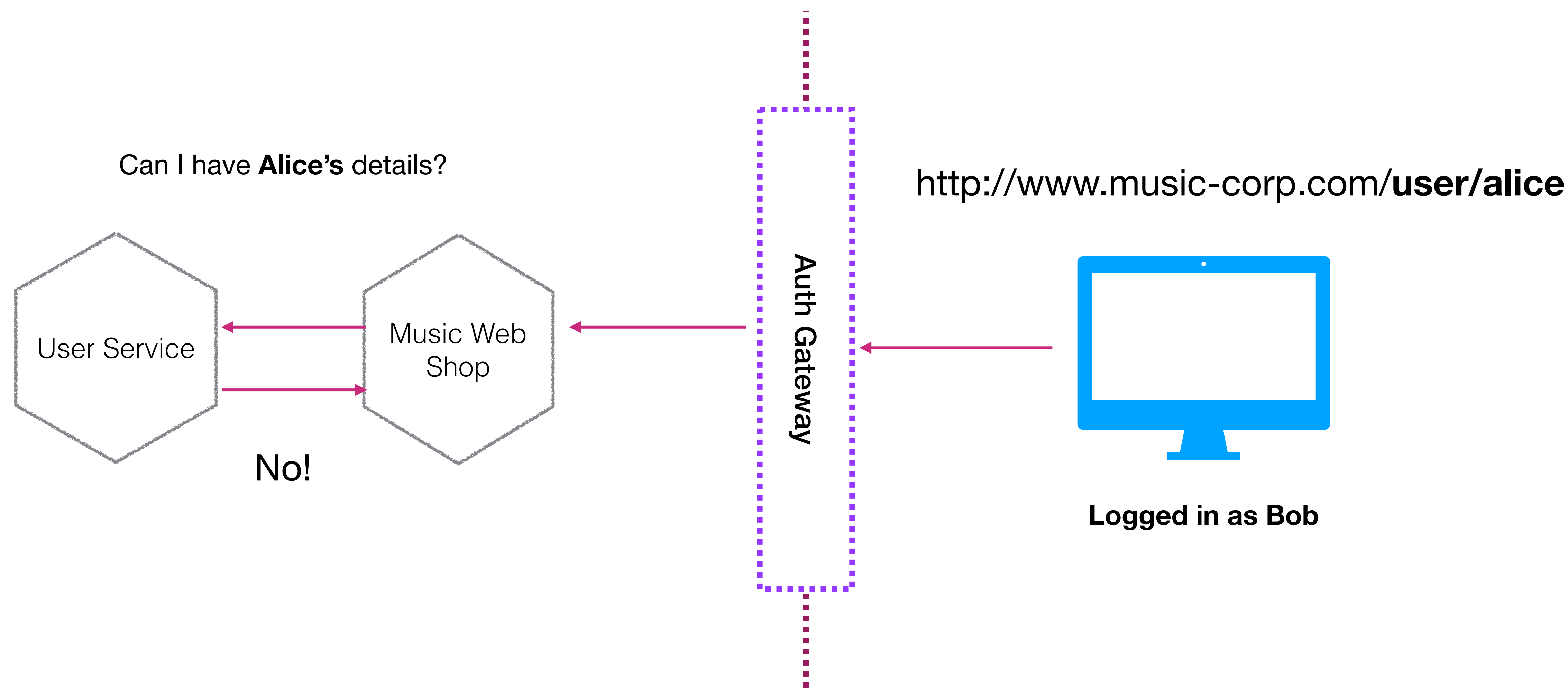
# AUTHORISE DOWNSTREAM



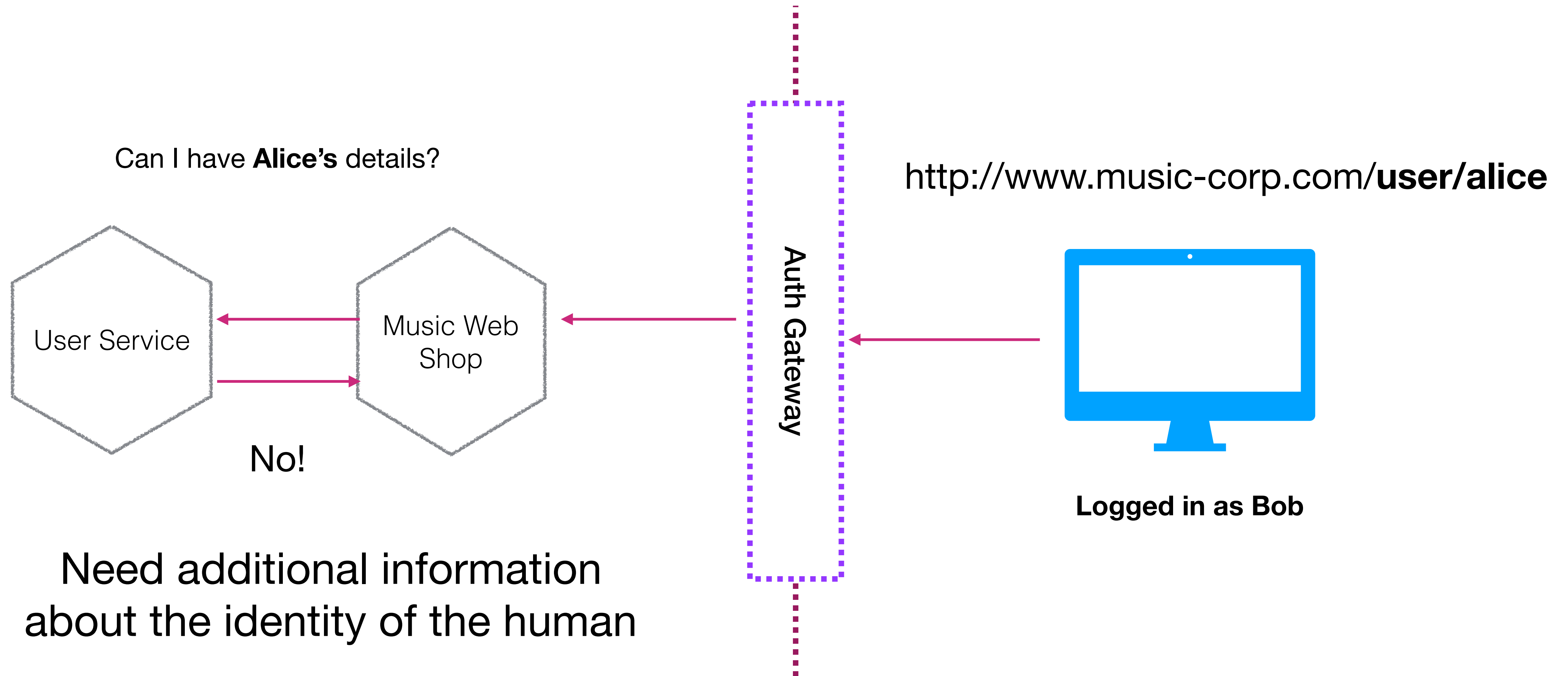
# AUTHORISE DOWNSTREAM



# AUTHORISE DOWNSTREAM



# AUTHORISE DOWNSTREAM



JWT

Debugger

Libraries

Introduction

Ask

Get a T-shirt!

Crafted by  Auth0



JSON Web Tokens are an open, industry standard **RFC 7519** method for representing claims securely between two parties.

JWT.IO allows you to decode, verify and generate JWT.

LEARN MORE ABOUT JWT

<https://jwt.io/>



```
{  
  "id": "402ndj39",  
  "name": "Alice Alison"  
}
```

```
{  
  "id": "402ndj39",  
  "name": "Alice Alison"  
}
```

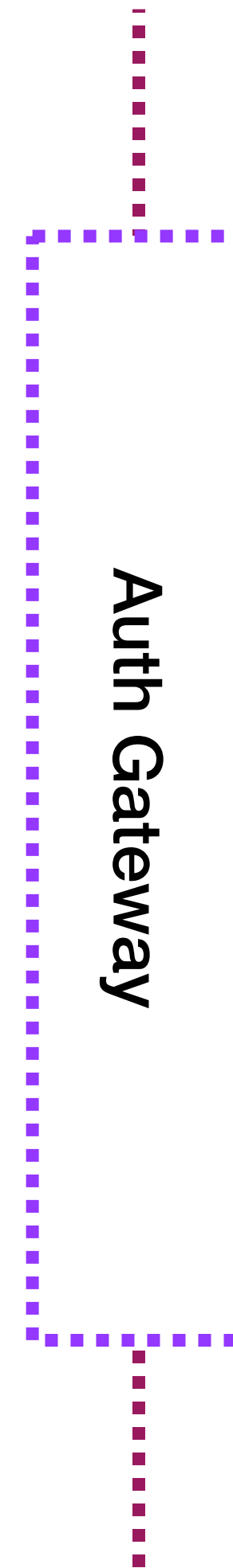
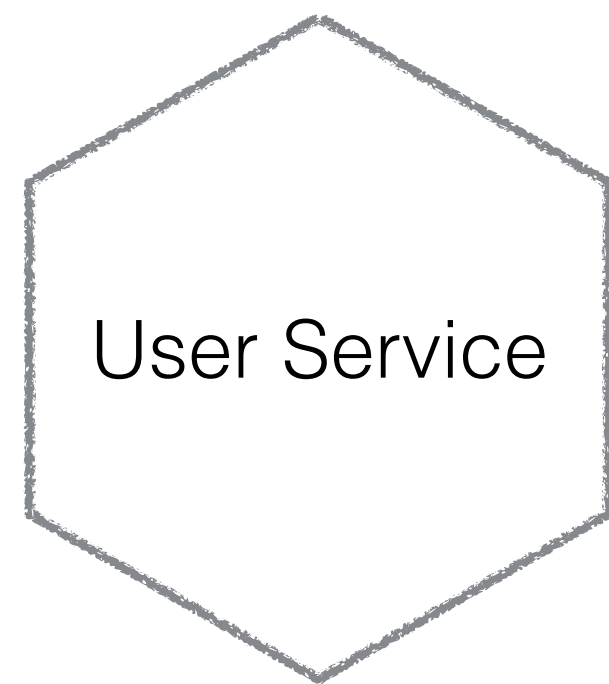
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.

eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4

gRG9lIiwiaXNTb2NpYWwiOnRydWV9.

4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4

# USING JWT TOKENS

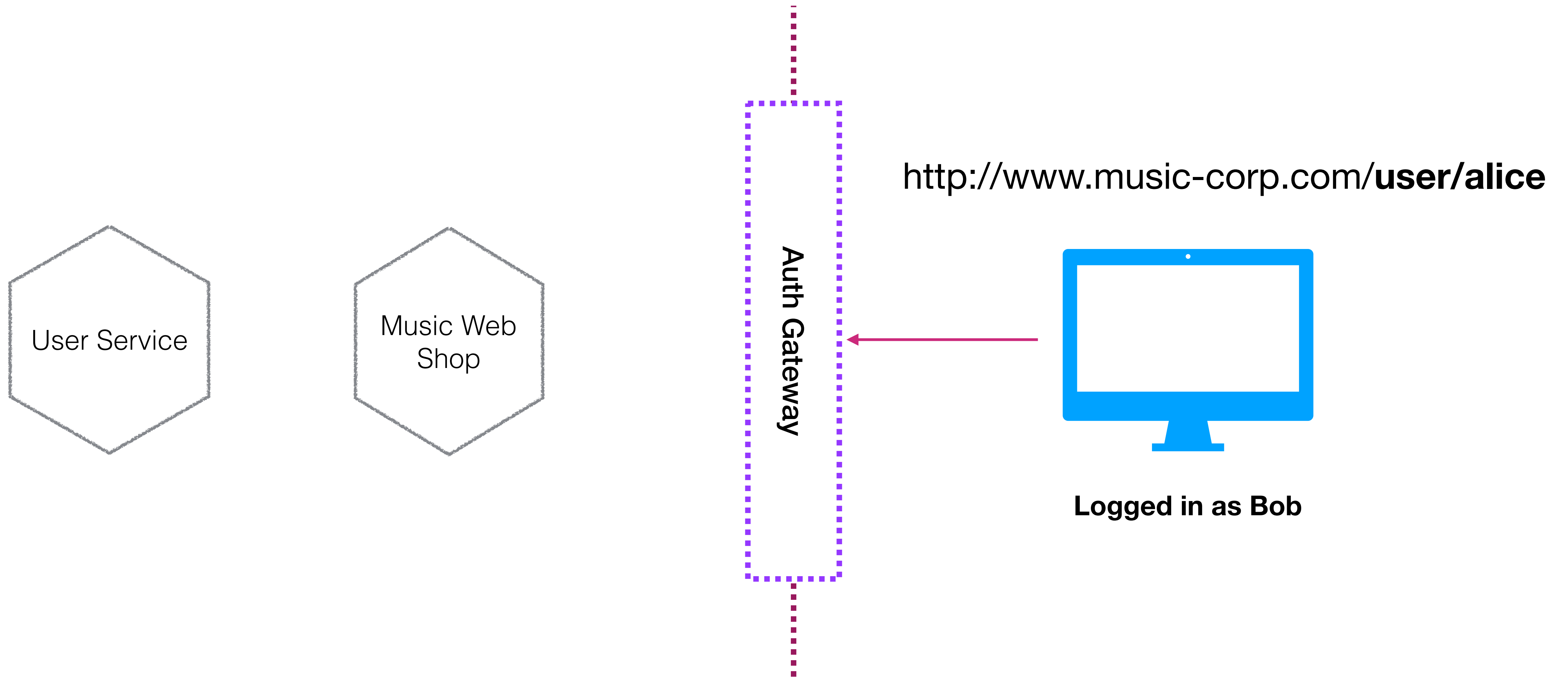


<http://www.music-corp.com/user/alice>

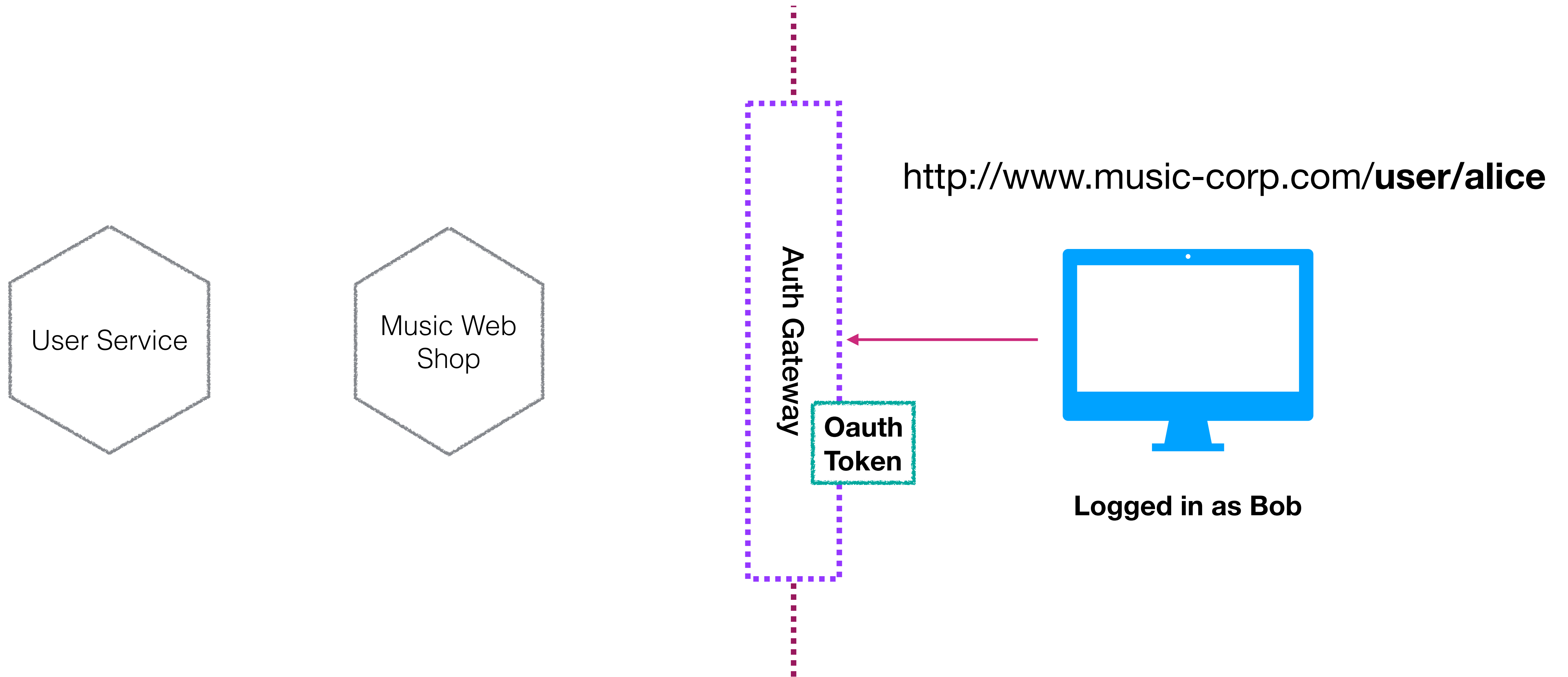


Logged in as Bob

# USING JWT TOKENS

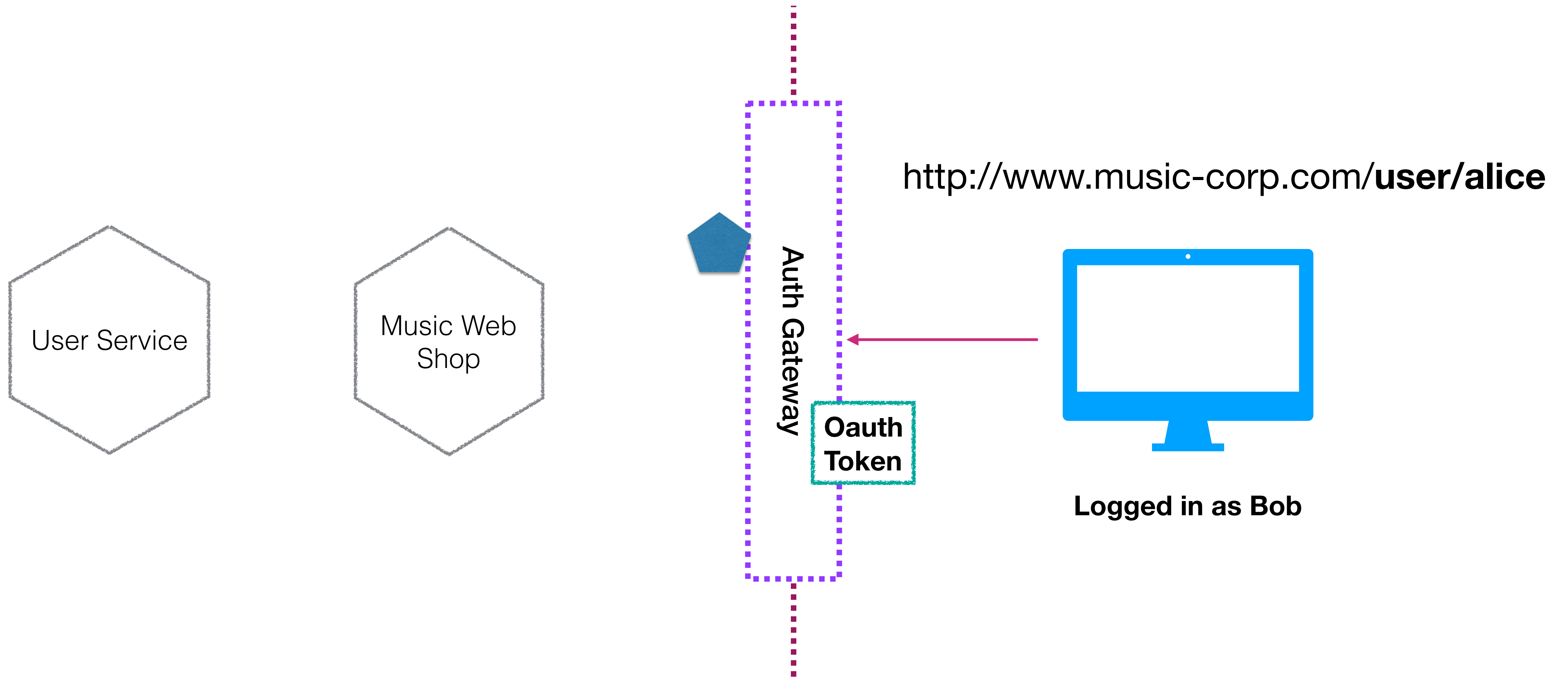


# USING JWT TOKENS

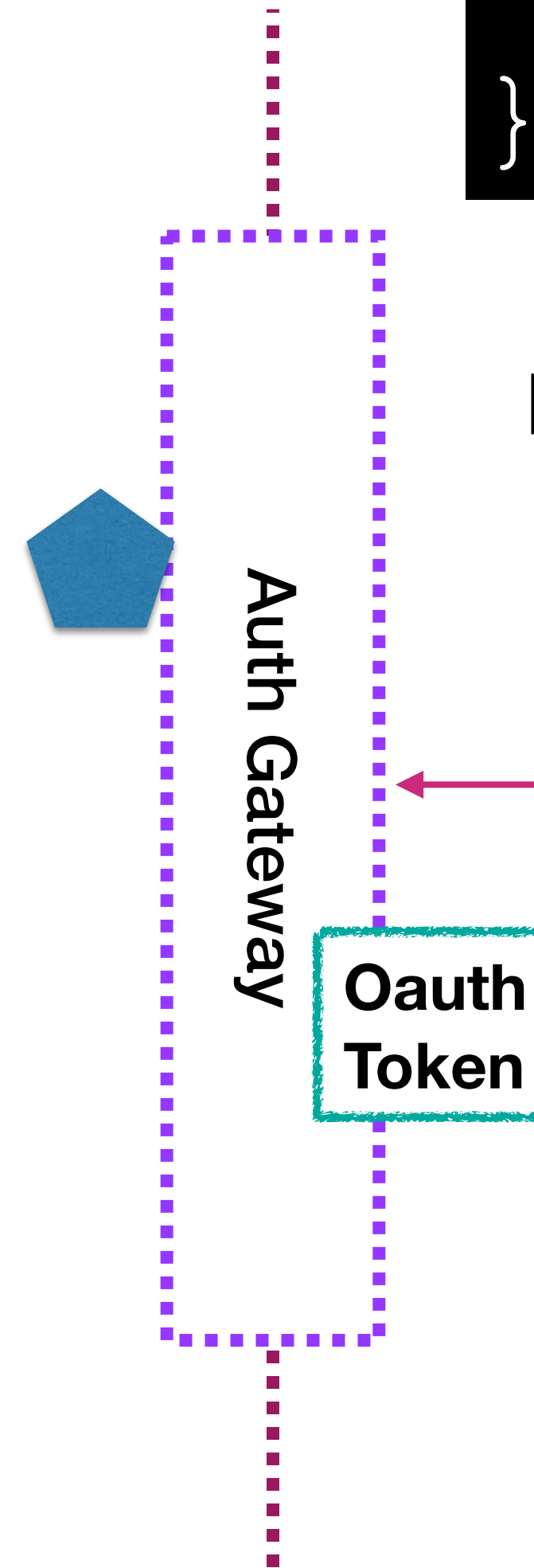
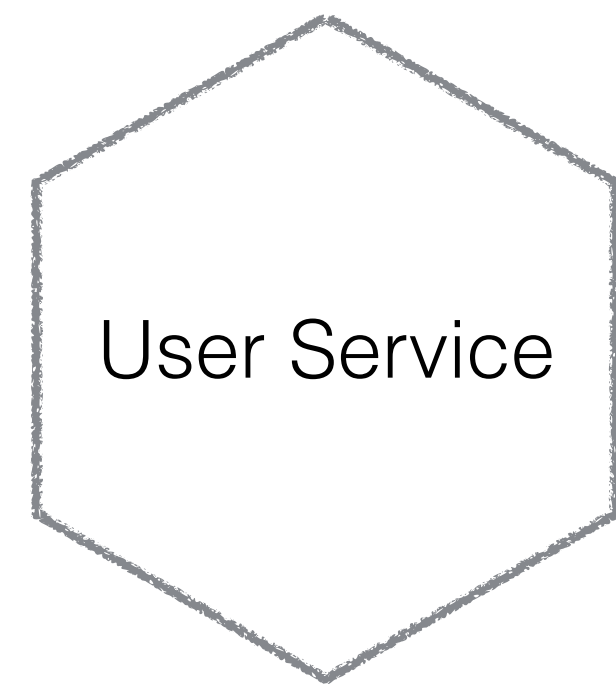




# USING JWT TOKENS



# USING JWT TOKENS



```
{  
  "user": "Bob"  
}
```

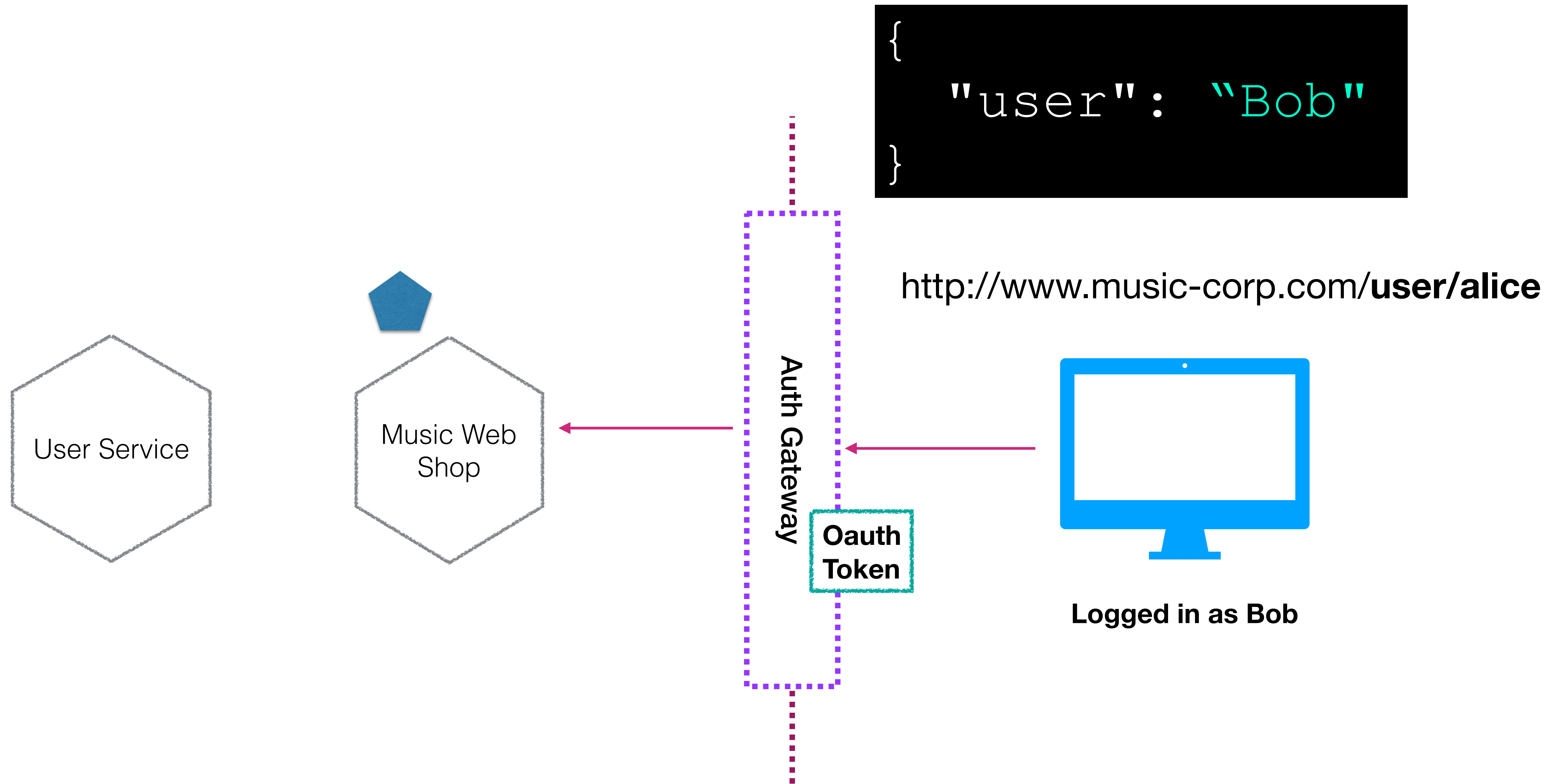
<http://www.music-corp.com/user/alice>



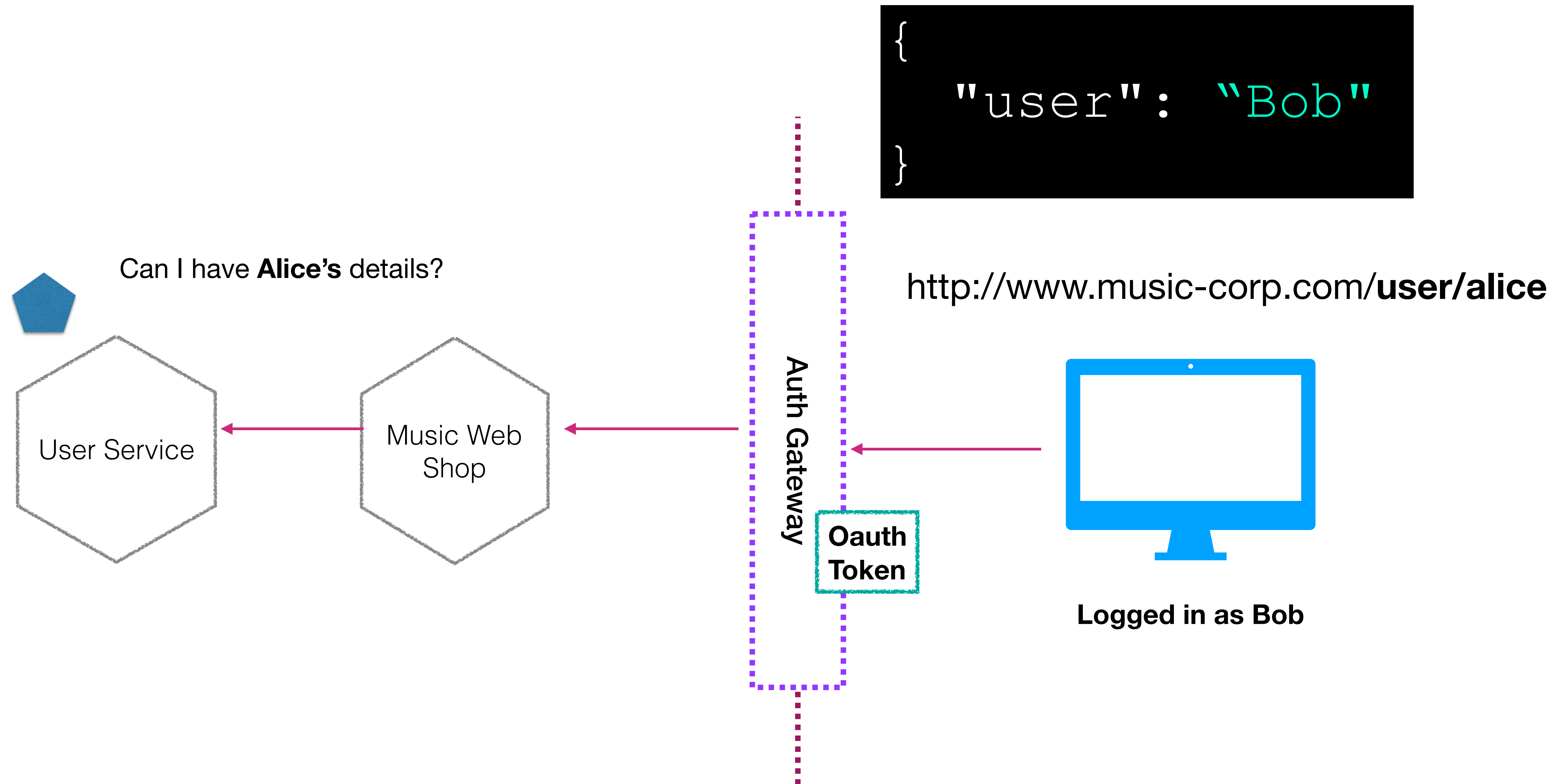
Logged in as Bob

Oauth  
Token

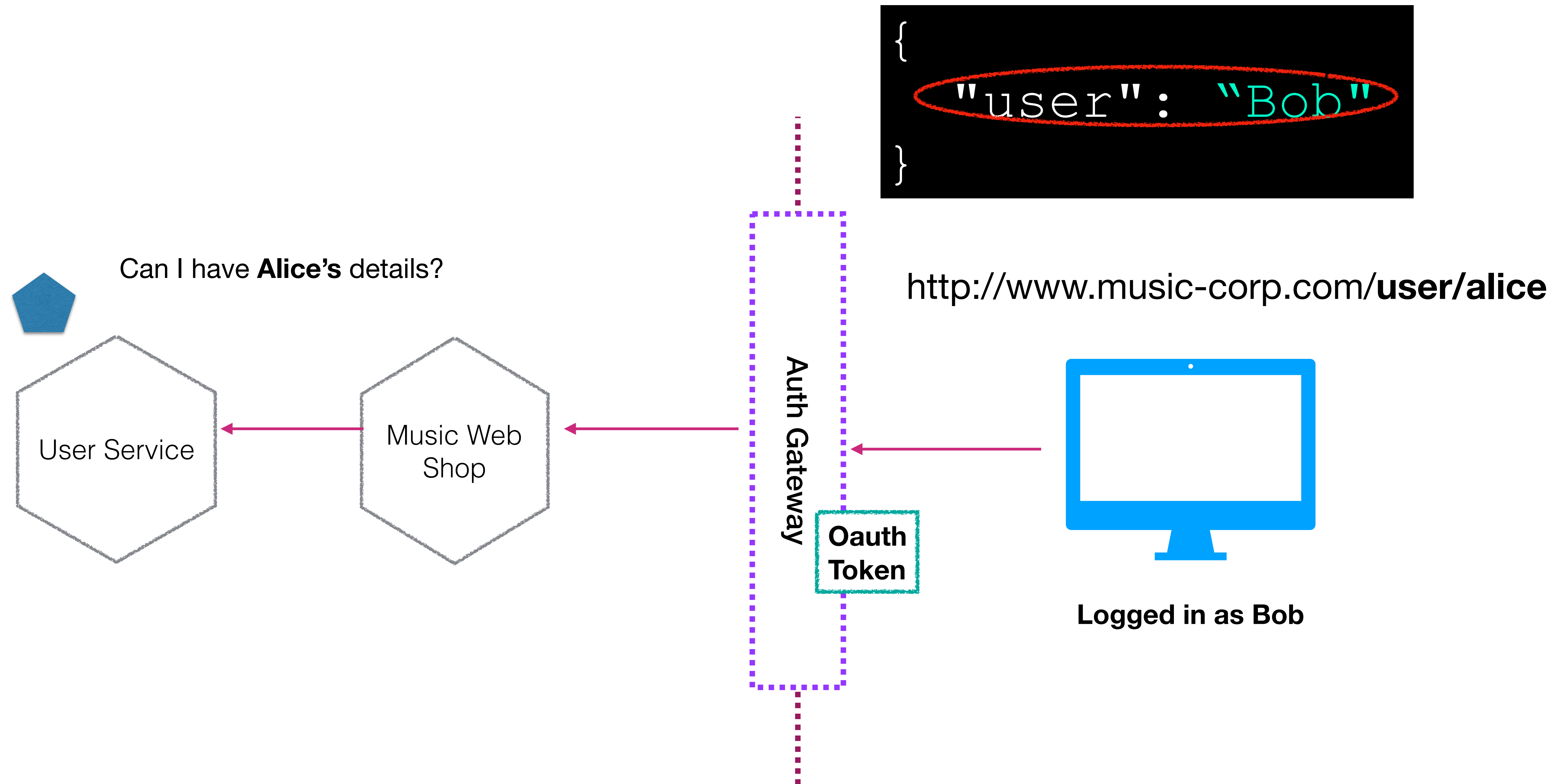
# USING JWT TOKENS



# USING JWT TOKENS

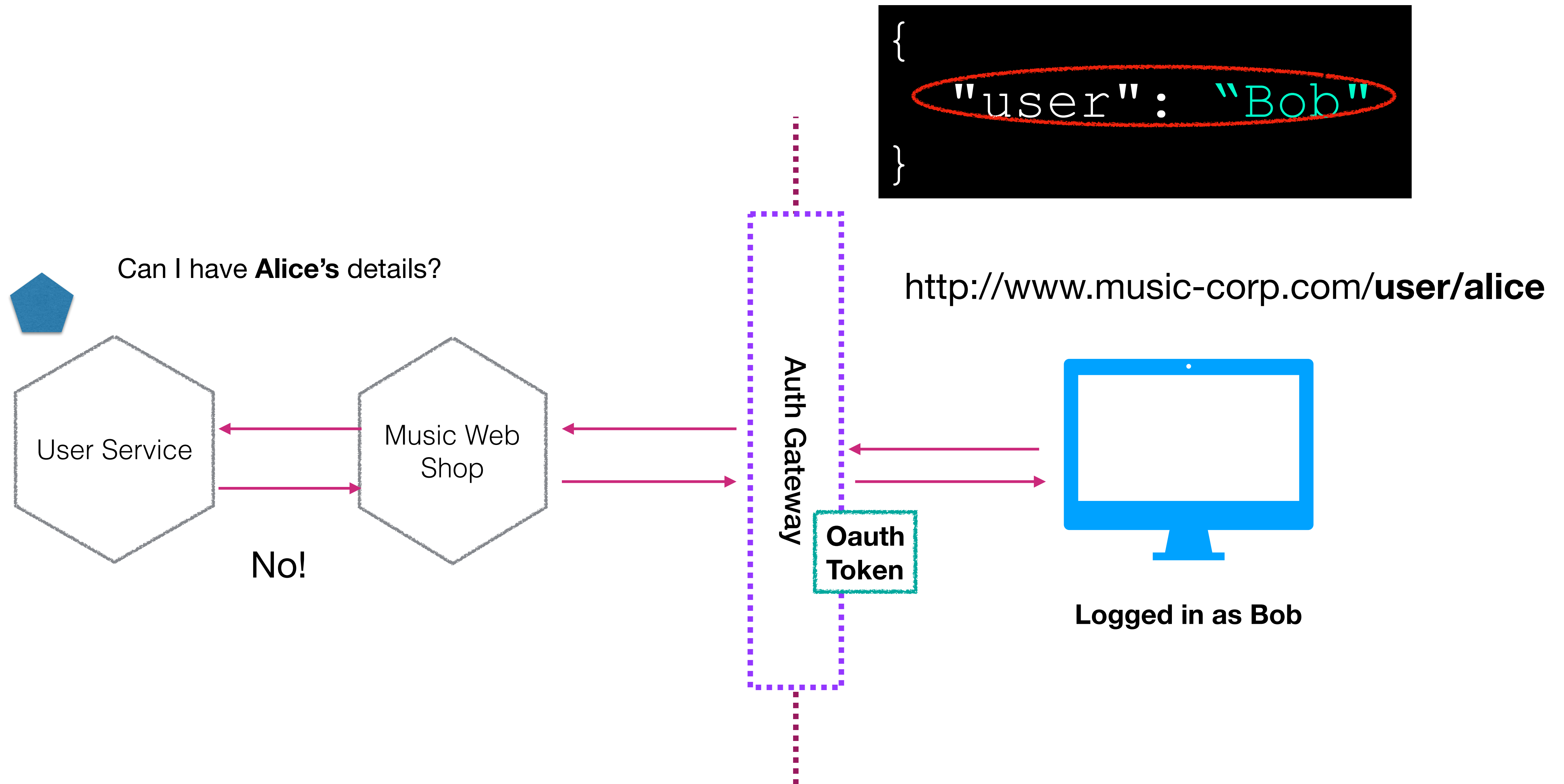


# USING JWT TOKENS



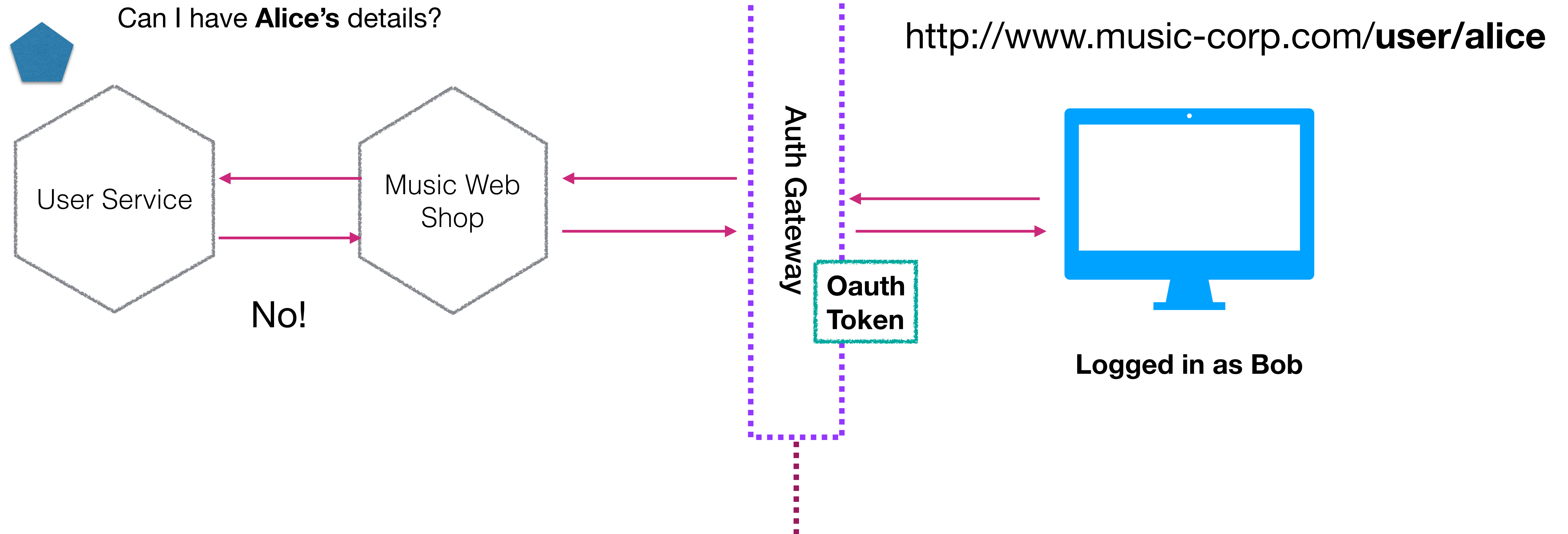


# USING JWT TOKENS



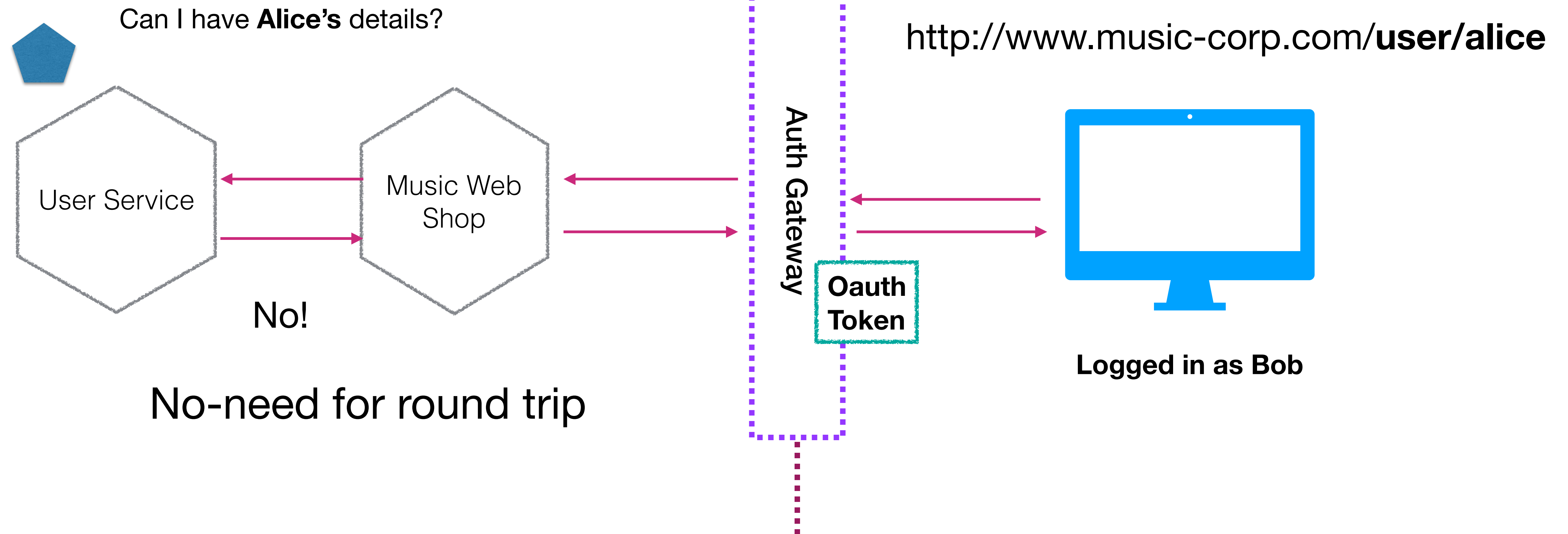
# USING JWT TOKENS

Token can be validated in the user service

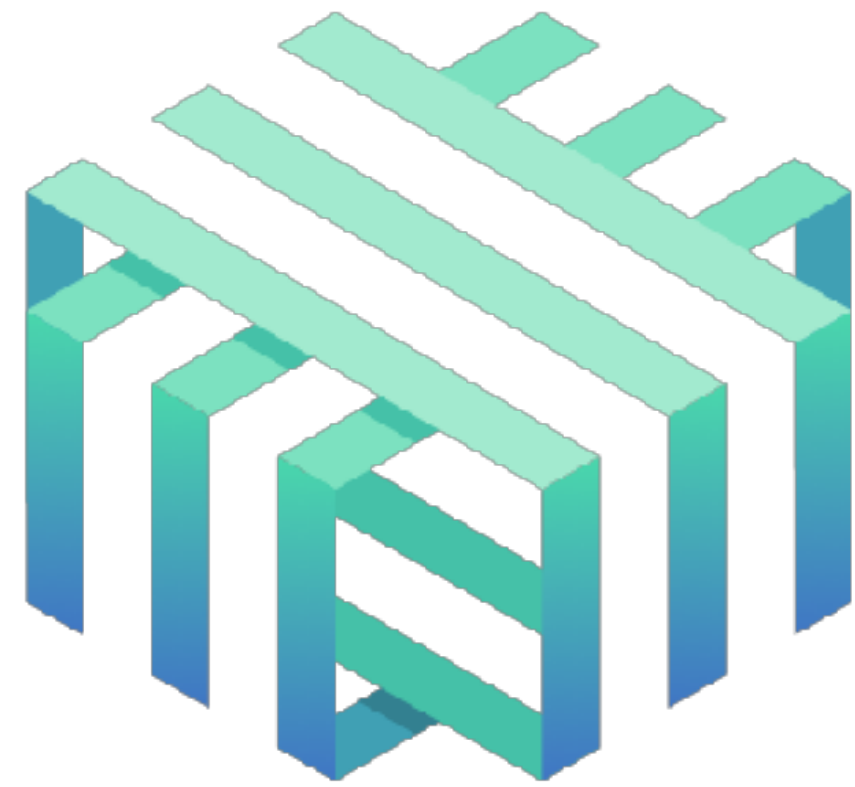


## USING JWT TOKENS

Token can be validated in the user service



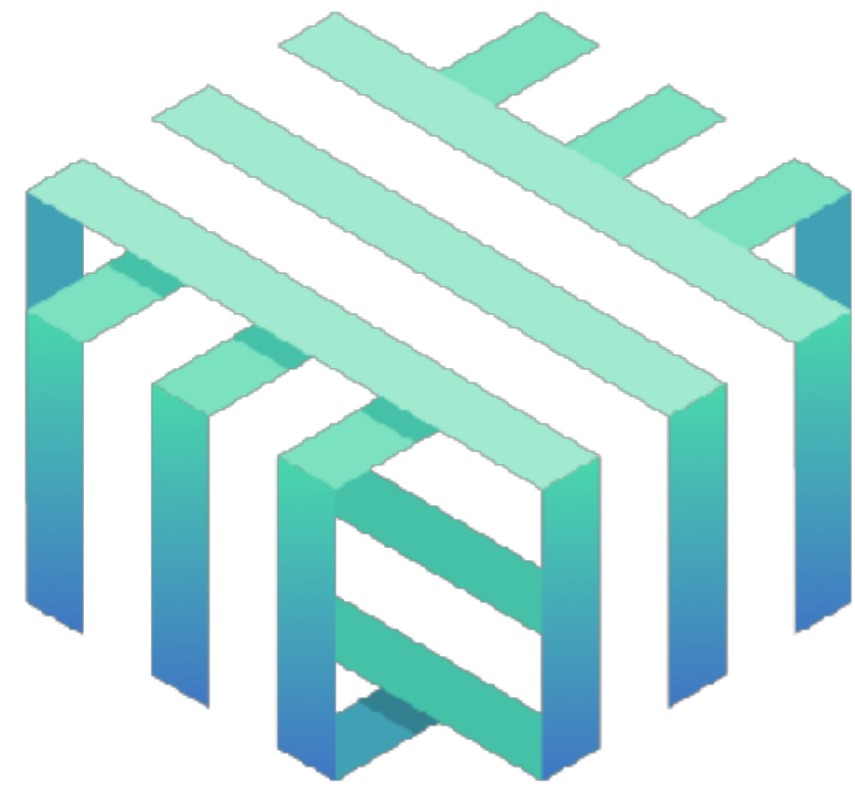
# SERVICE MESHES



**Linkerd**

**<https://linkerd.io>**

# SERVICE MESHES



**Linkerd**

**<https://linkerd.io>**

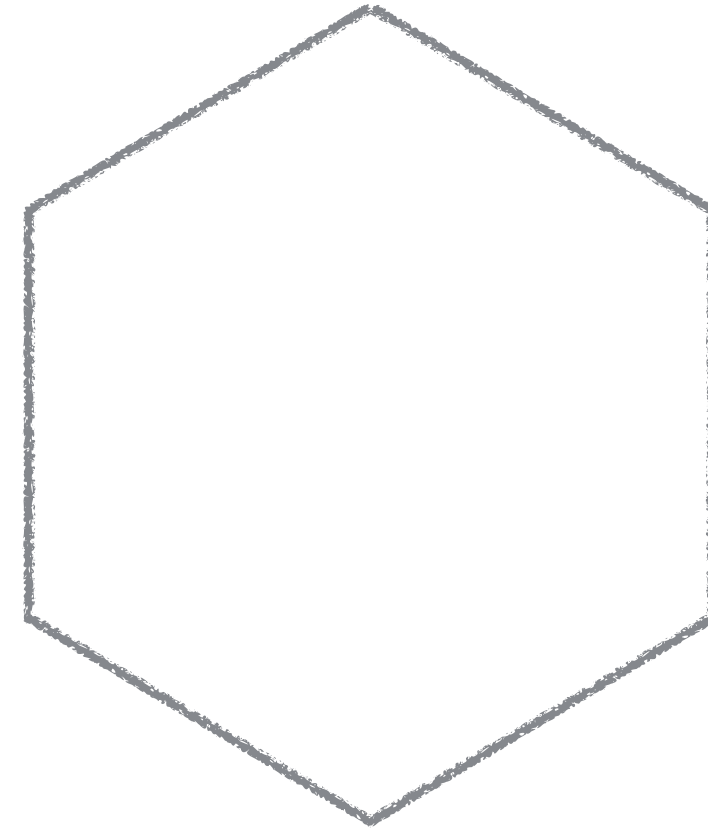
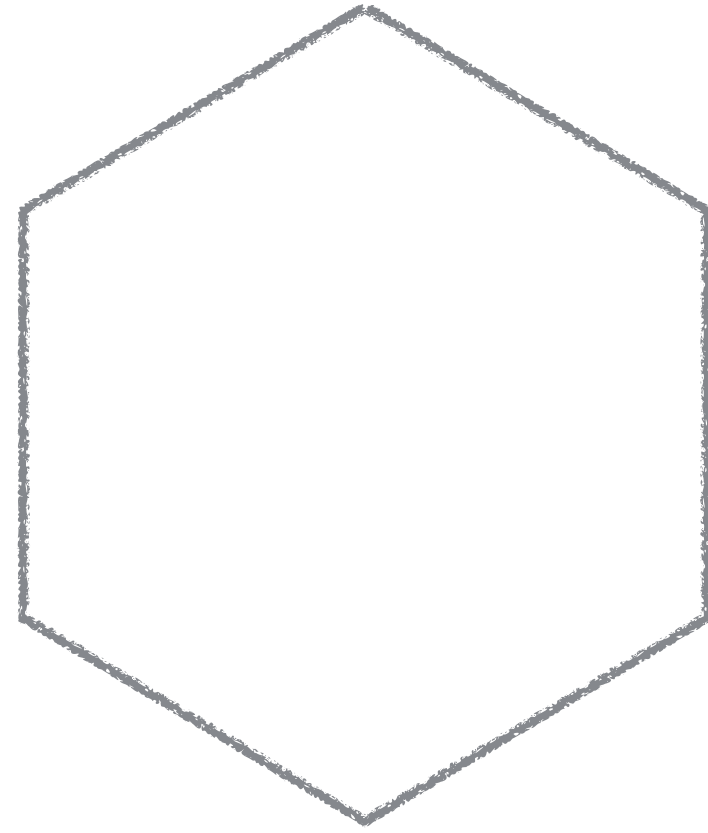


**Istio**

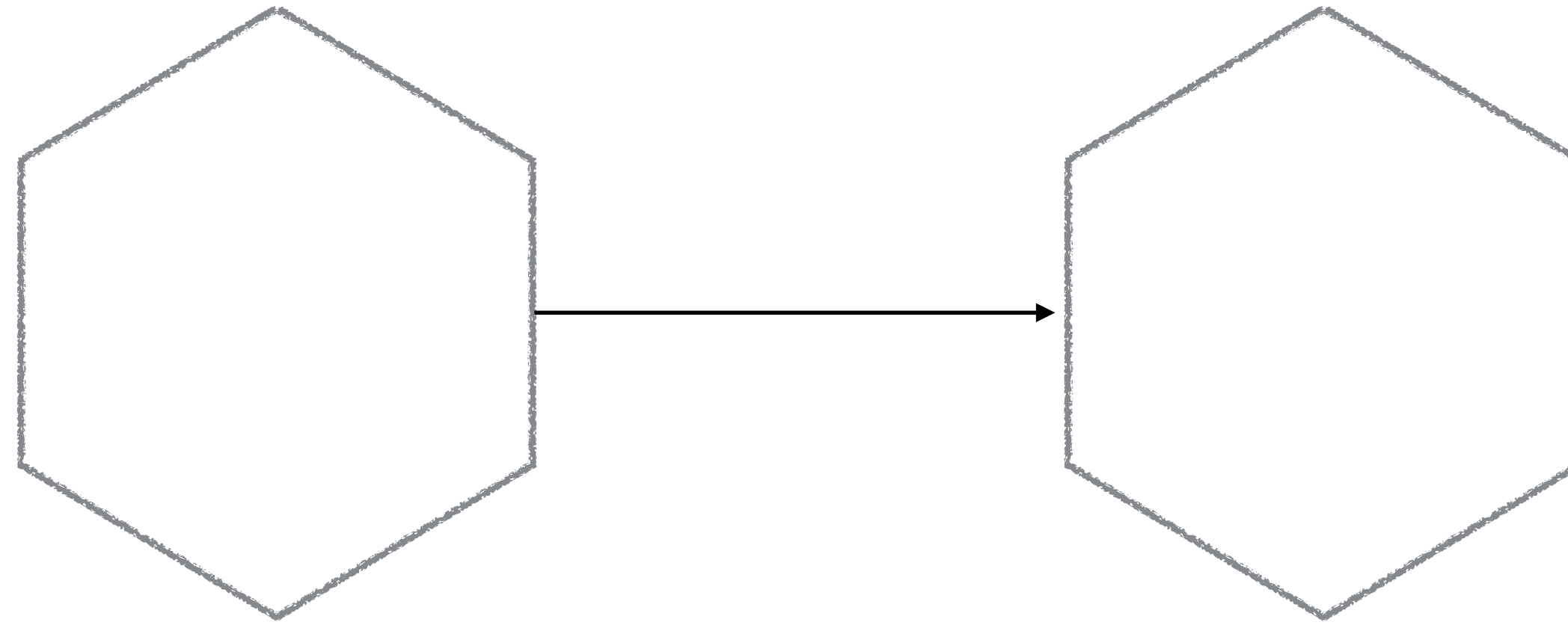
**<https://istio.io>**



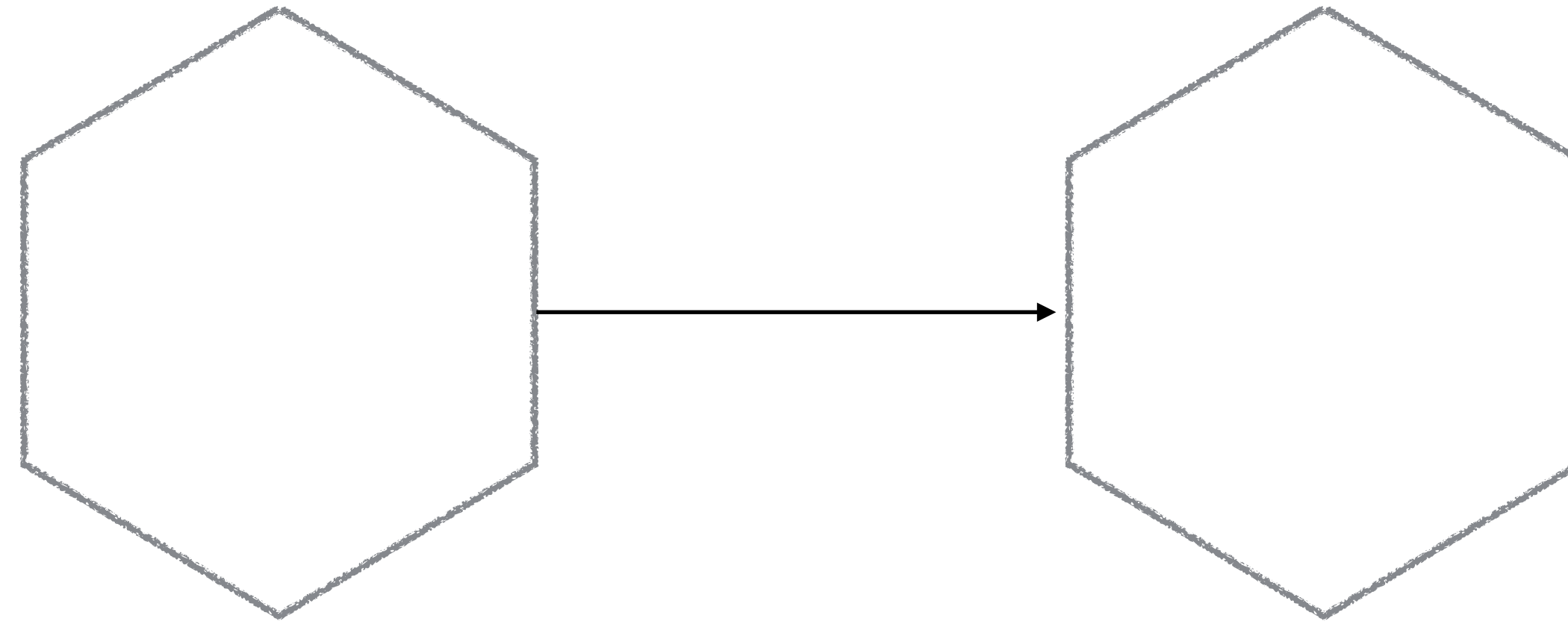
# COMMON CONNECTION CONCERNS



# COMMON CONNECTION CONCERNS

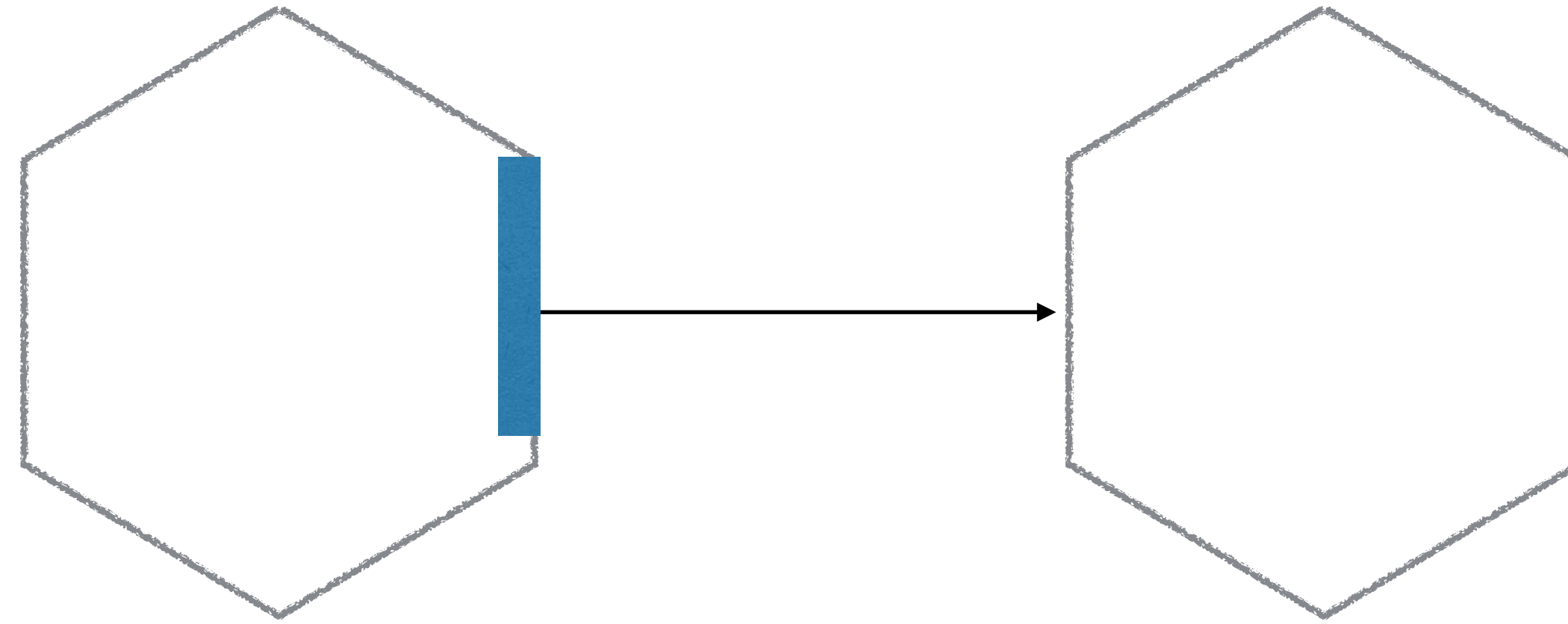


# COMMON CONNECTION CONCERNS



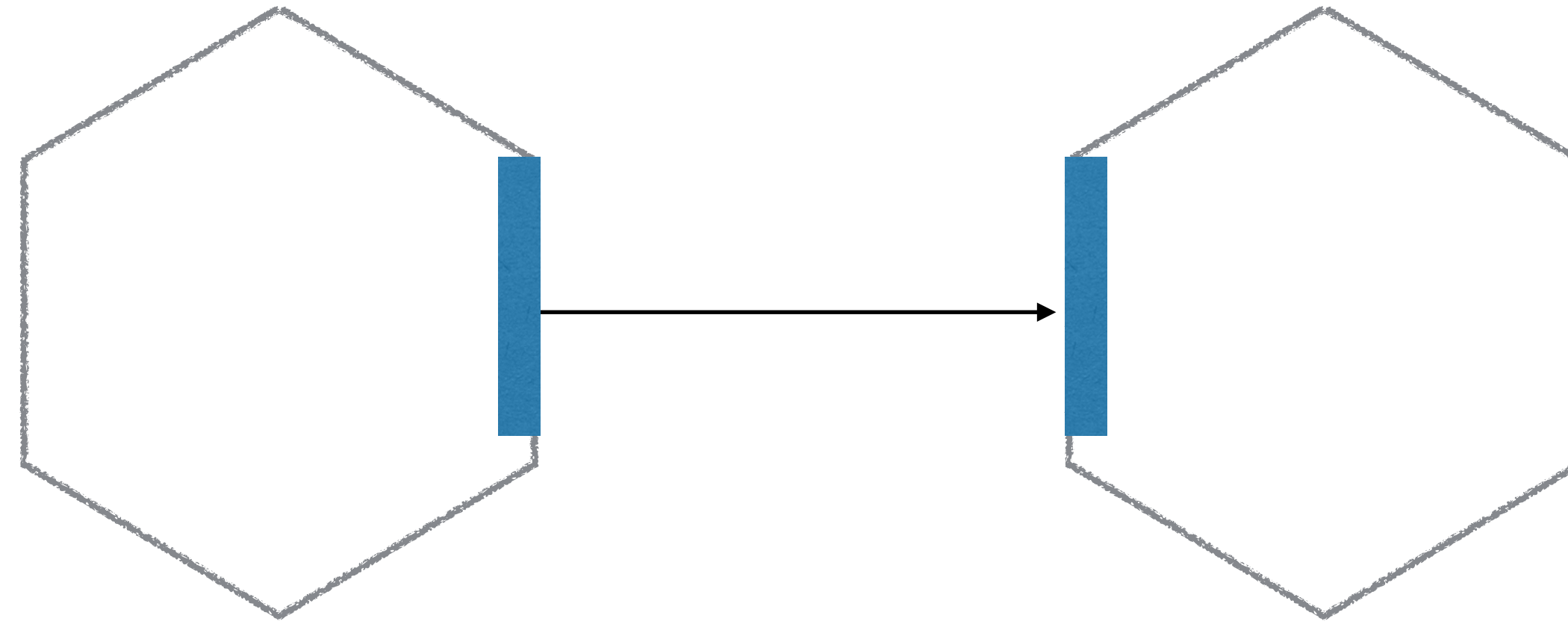
Tracing

# COMMON CONNECTION CONCERNS



Tracing

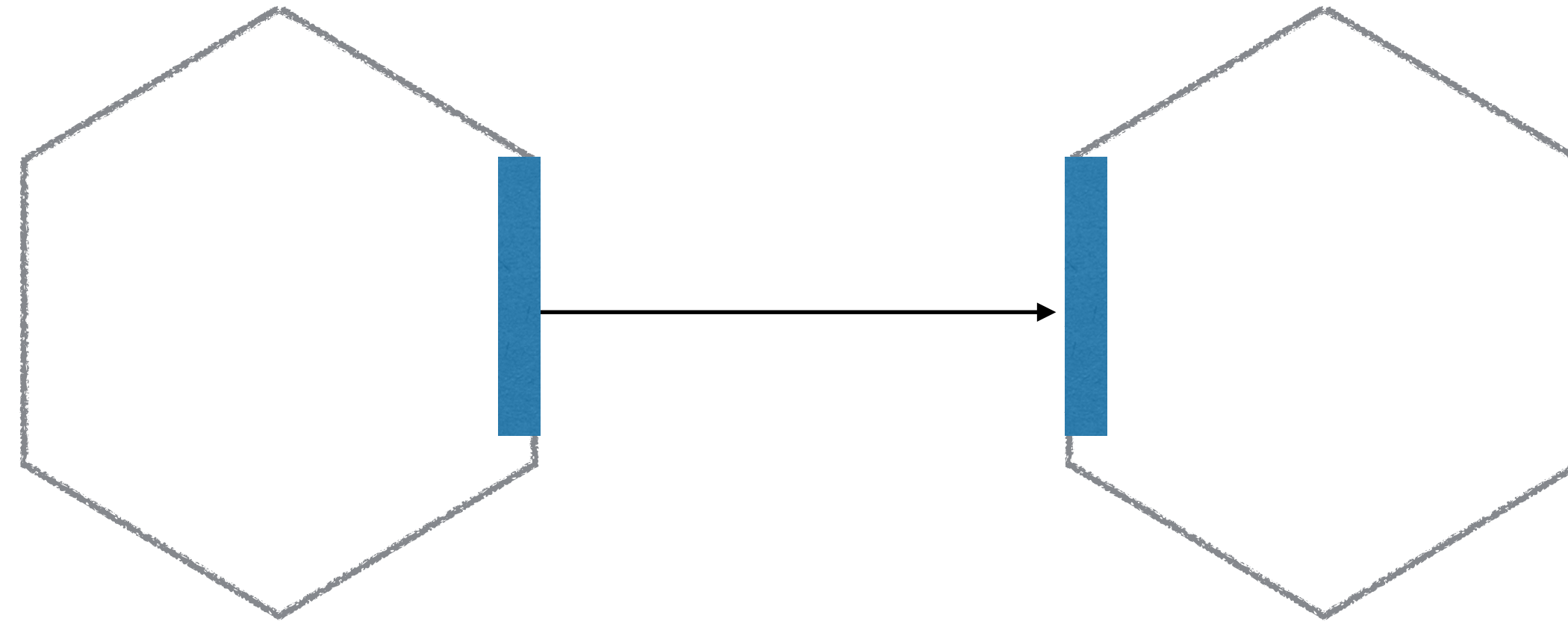
# COMMON CONNECTION CONCERNS



Tracing



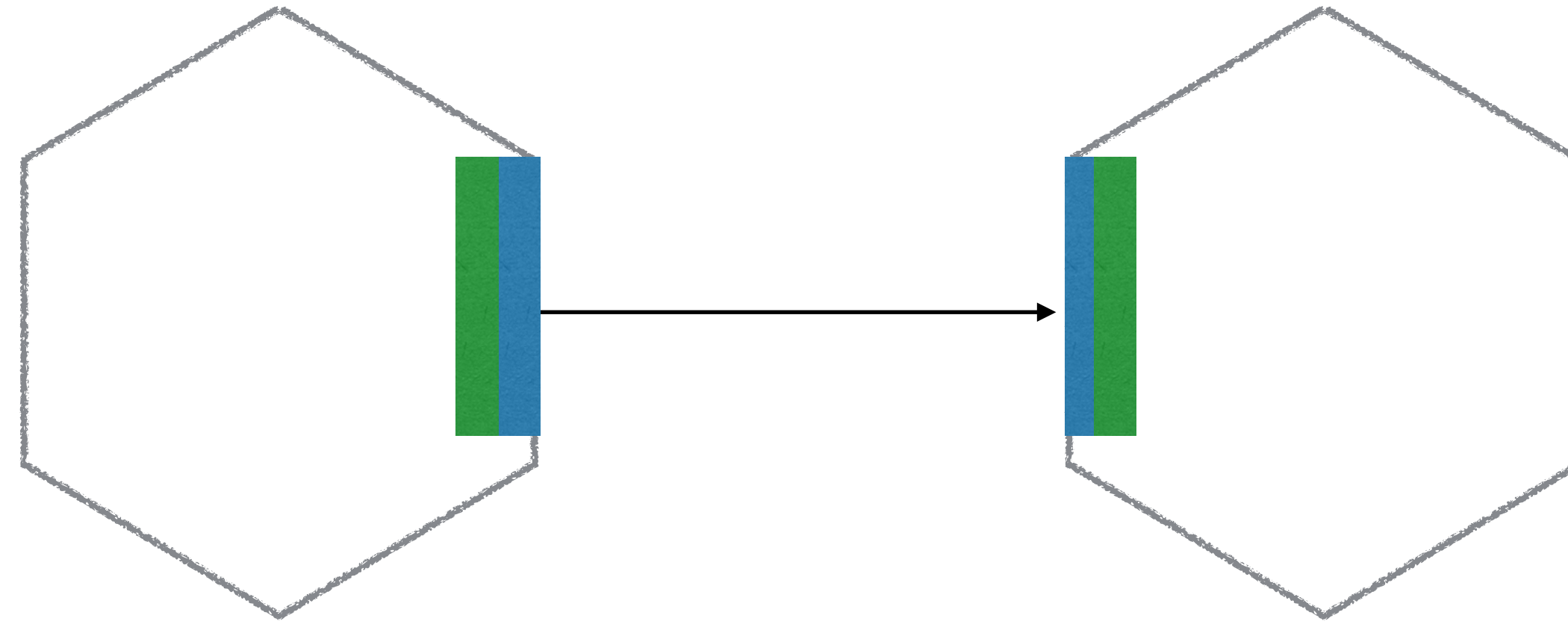
# COMMON CONNECTION CONCERNS



Tracing

Load Balancing & Service Discovery

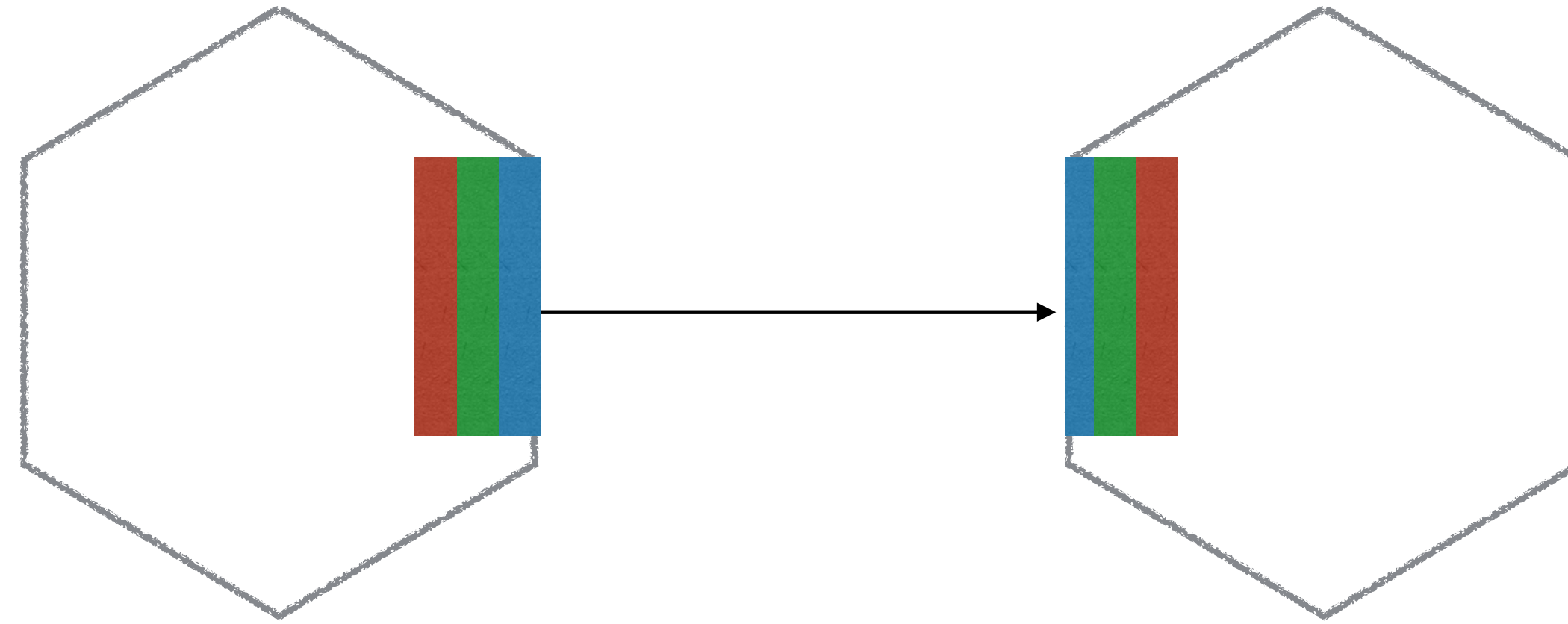
# COMMON CONNECTION CONCERNS



Tracing

Load Balancing & Service Discovery

# COMMON CONNECTION CONCERNS

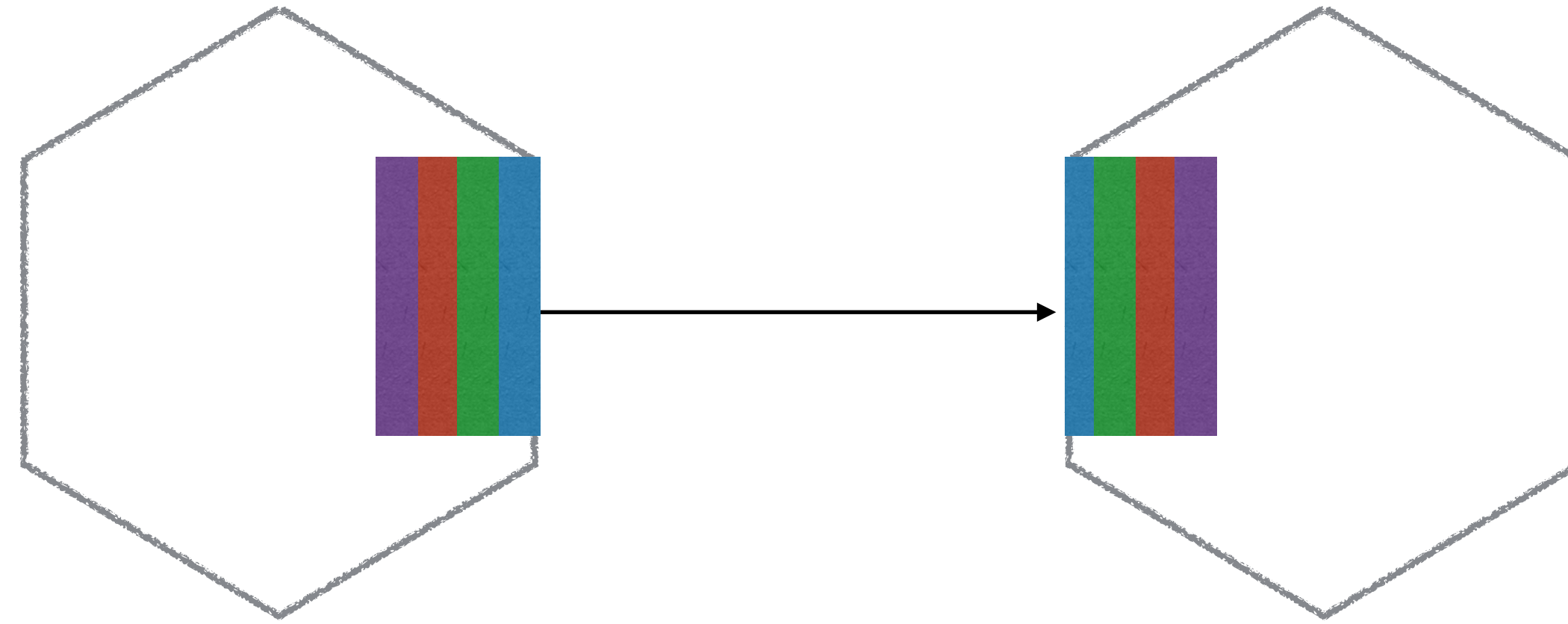


Tracing

Load Balancing & Service Discovery

Authorisation & Authentication

# COMMON CONNECTION CONCERNS



Tracing

Load Balancing & Service Discovery

Authorisation & Authentication

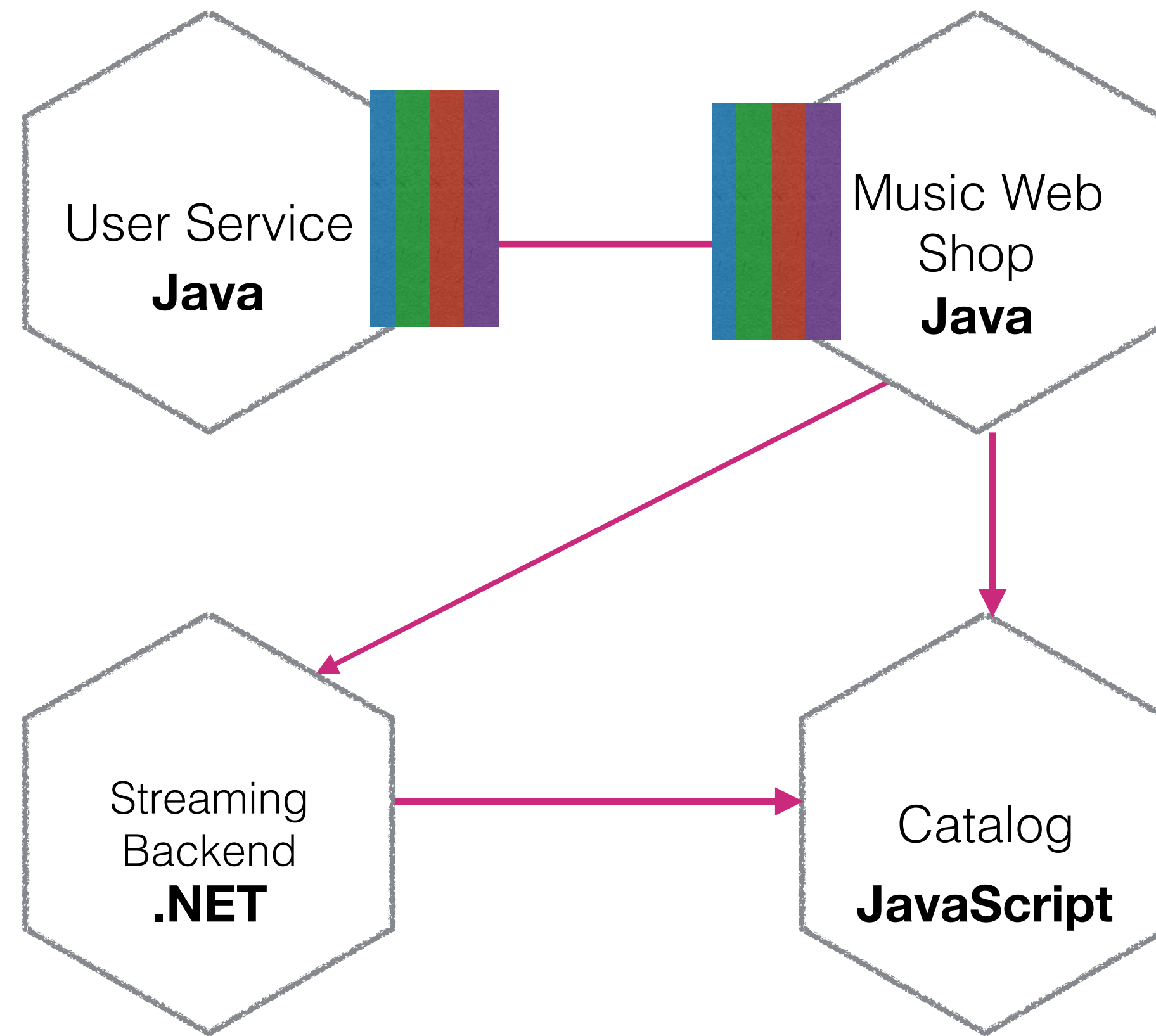
Connection Resilience & Retry

## COMMON MICROSERVICE FRAMEWORKS

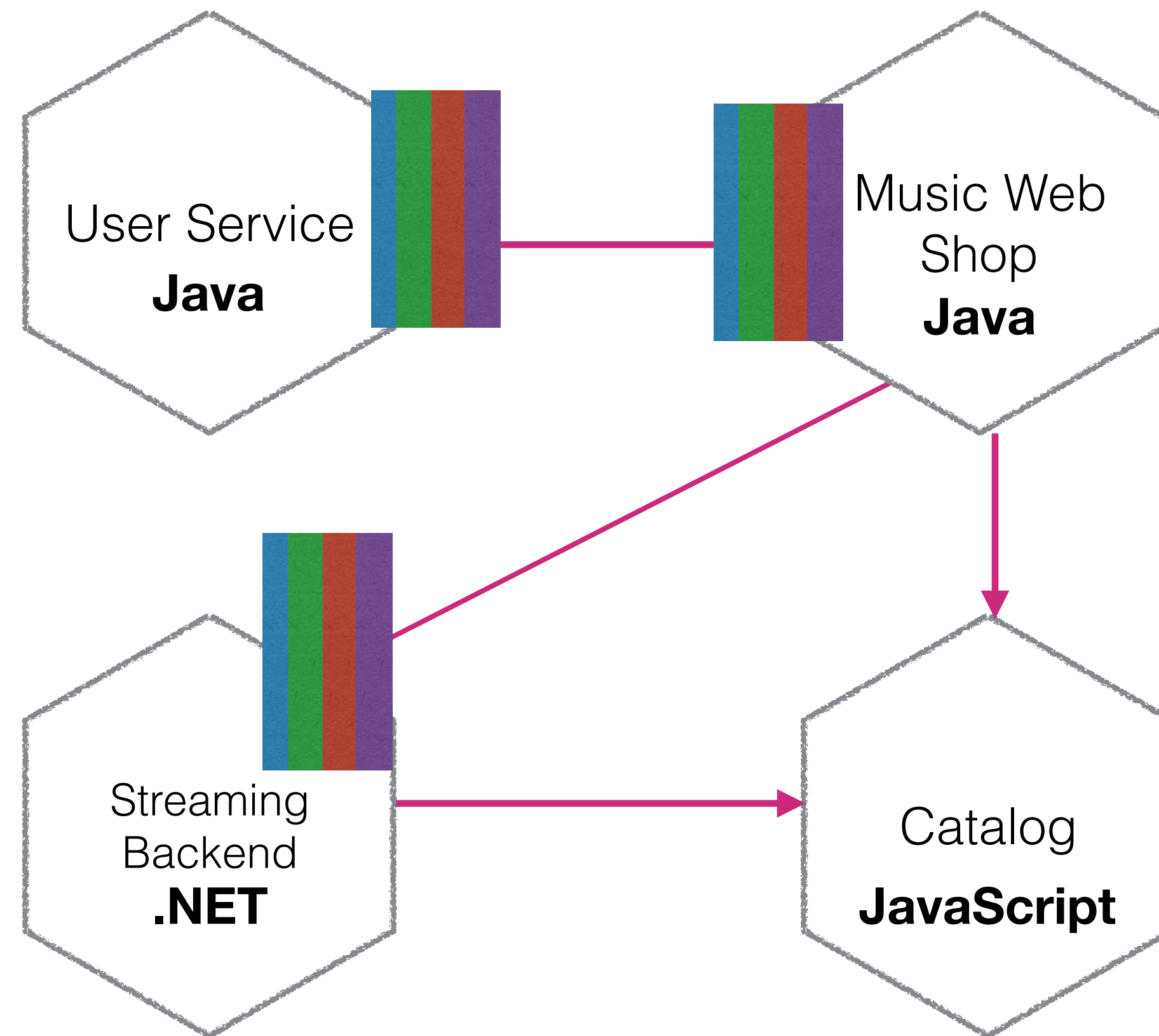




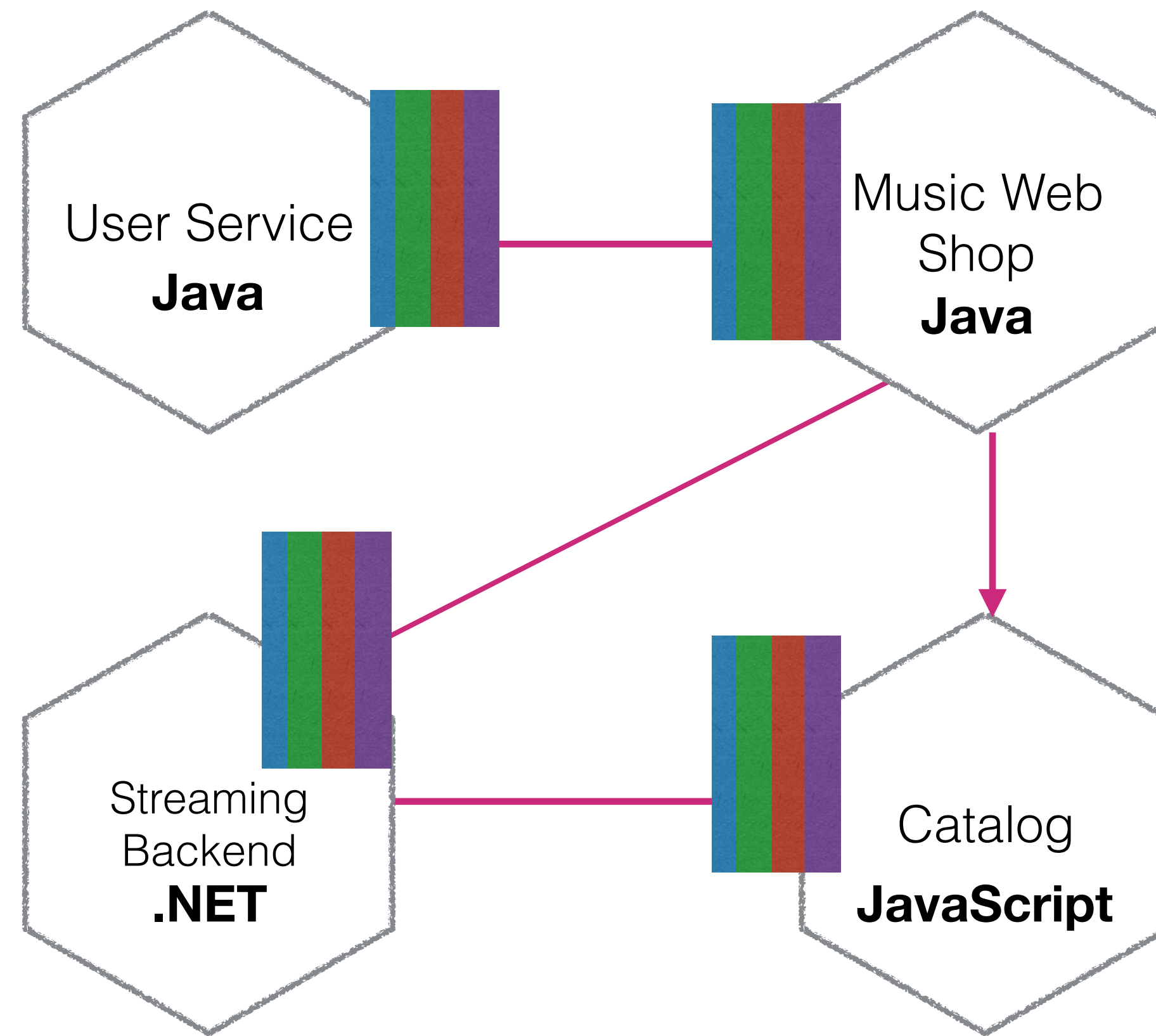
# POLYGLOT?



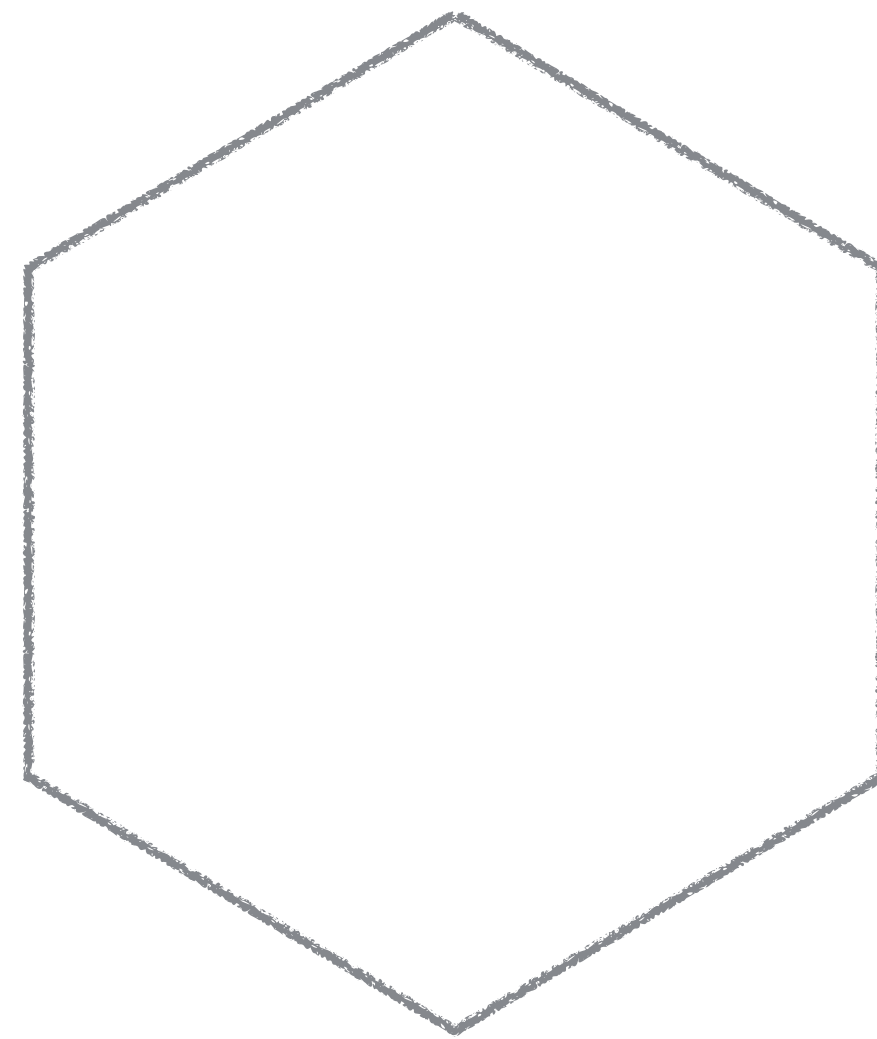
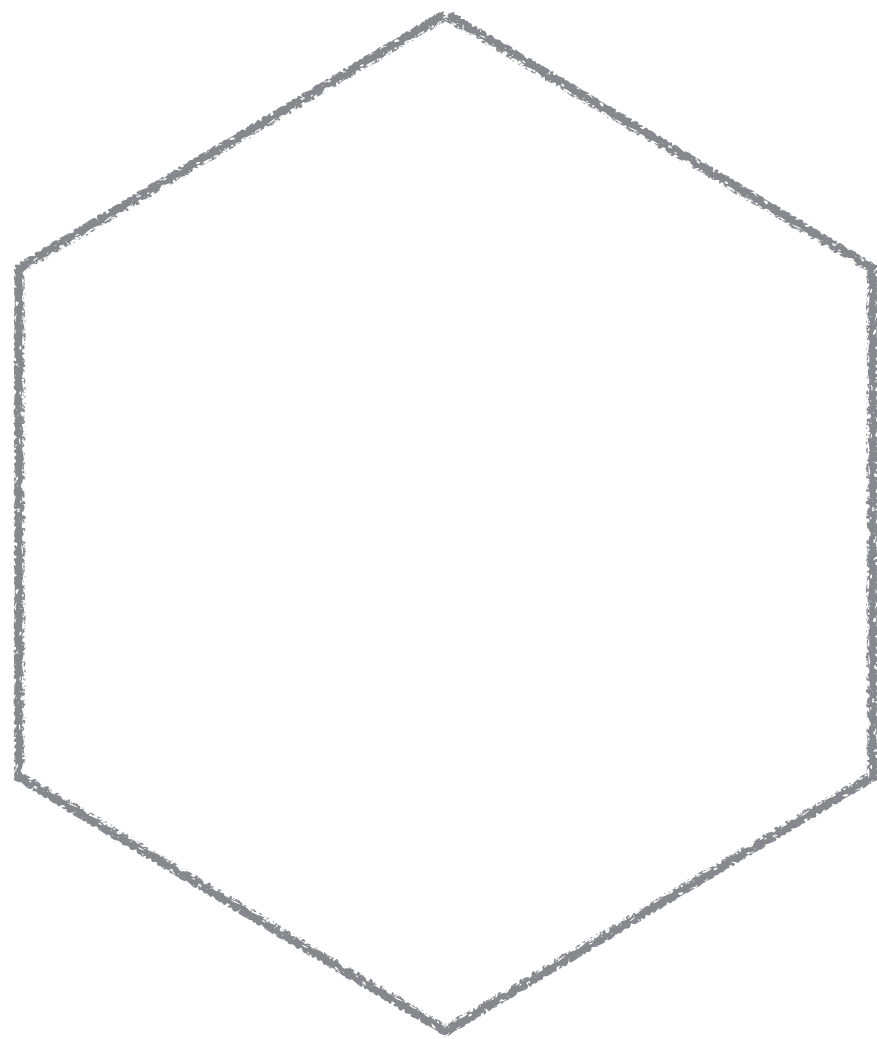
# POLYGLOT?



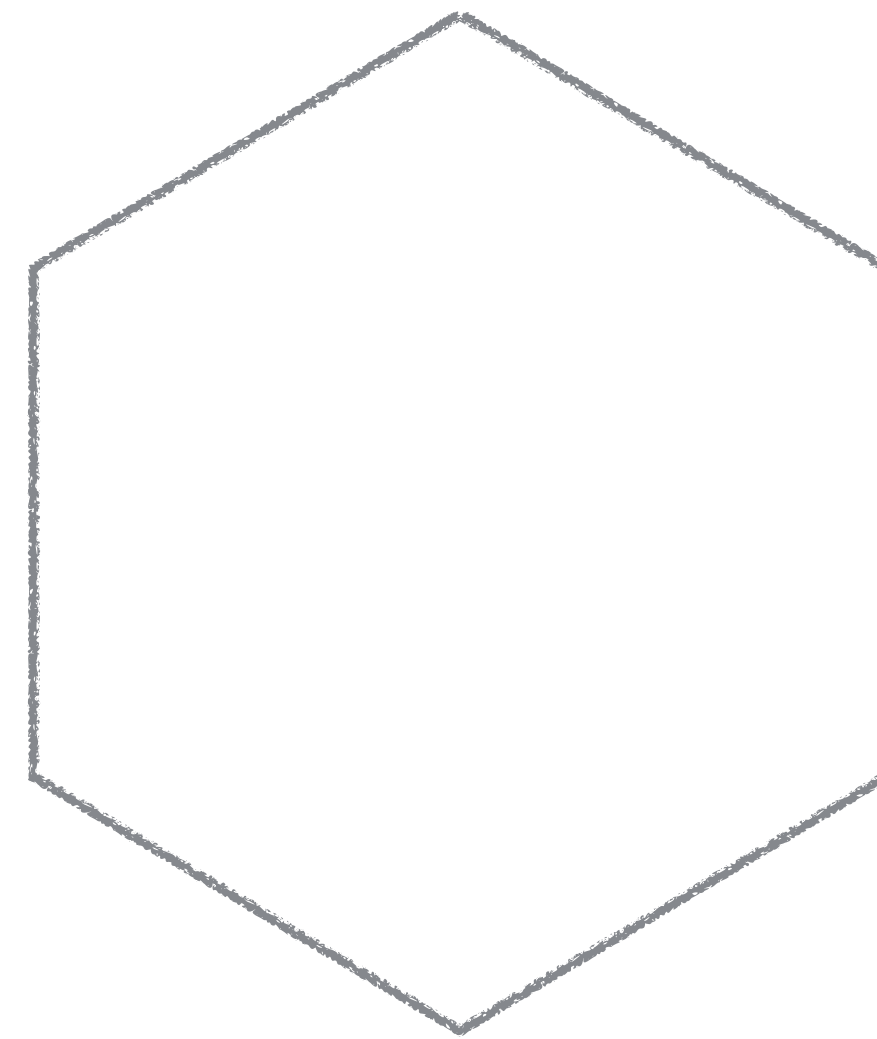
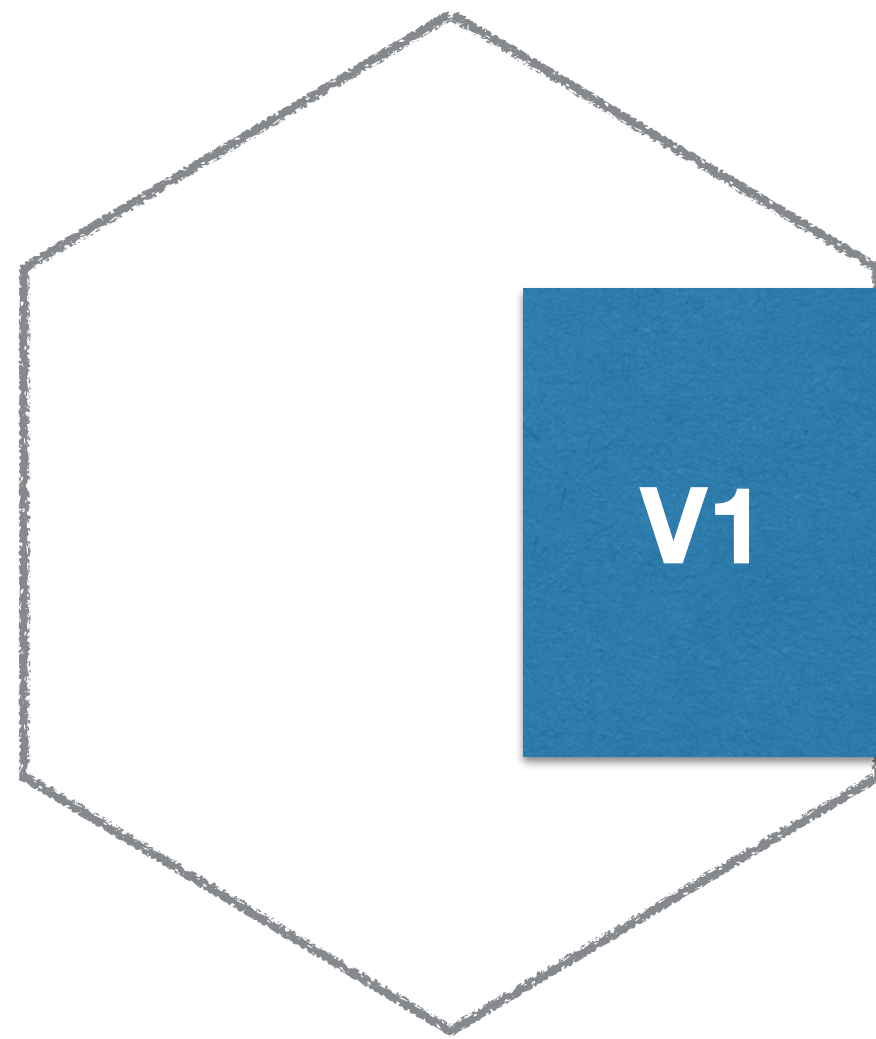
# POLYGLOT?



# VERSION DRIFT

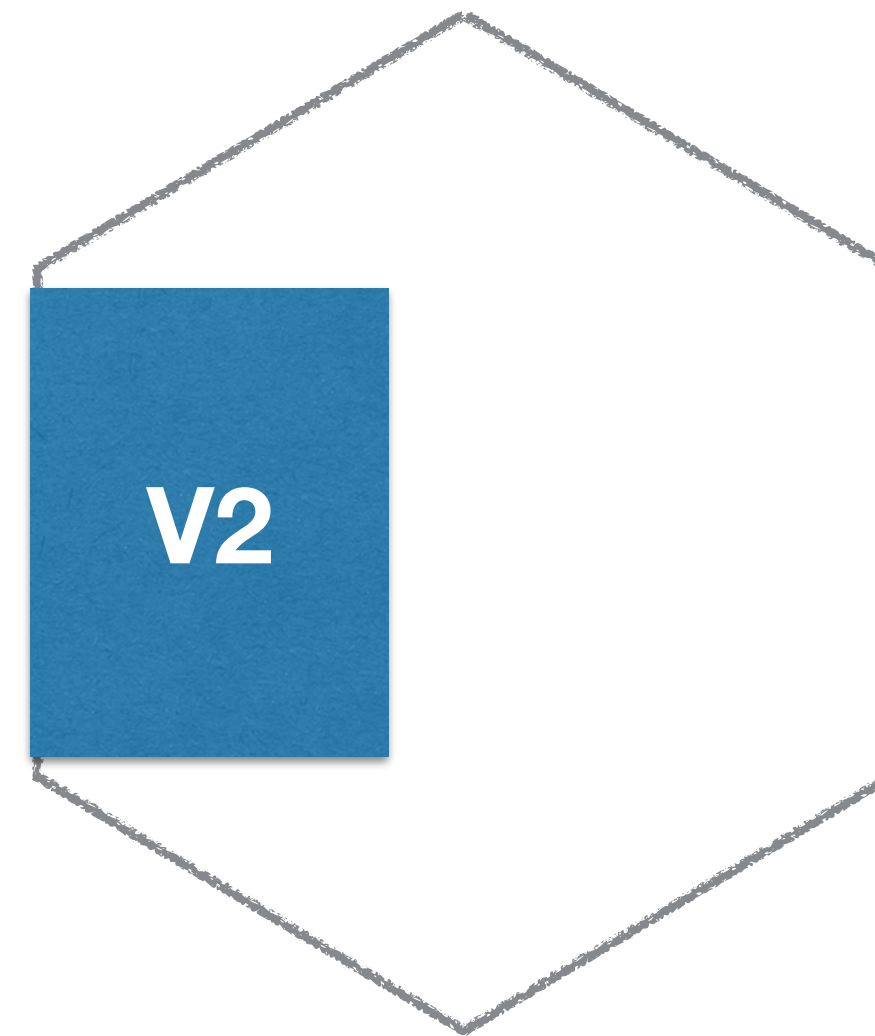
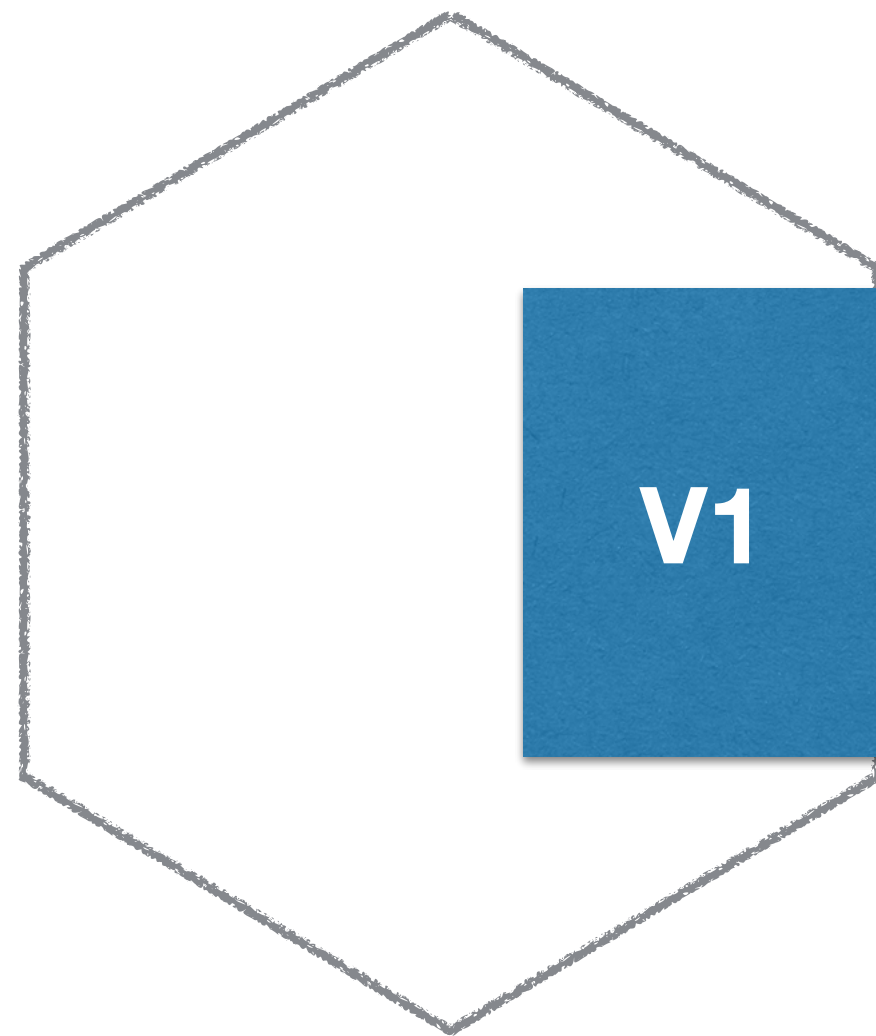


# VERSION DRIFT





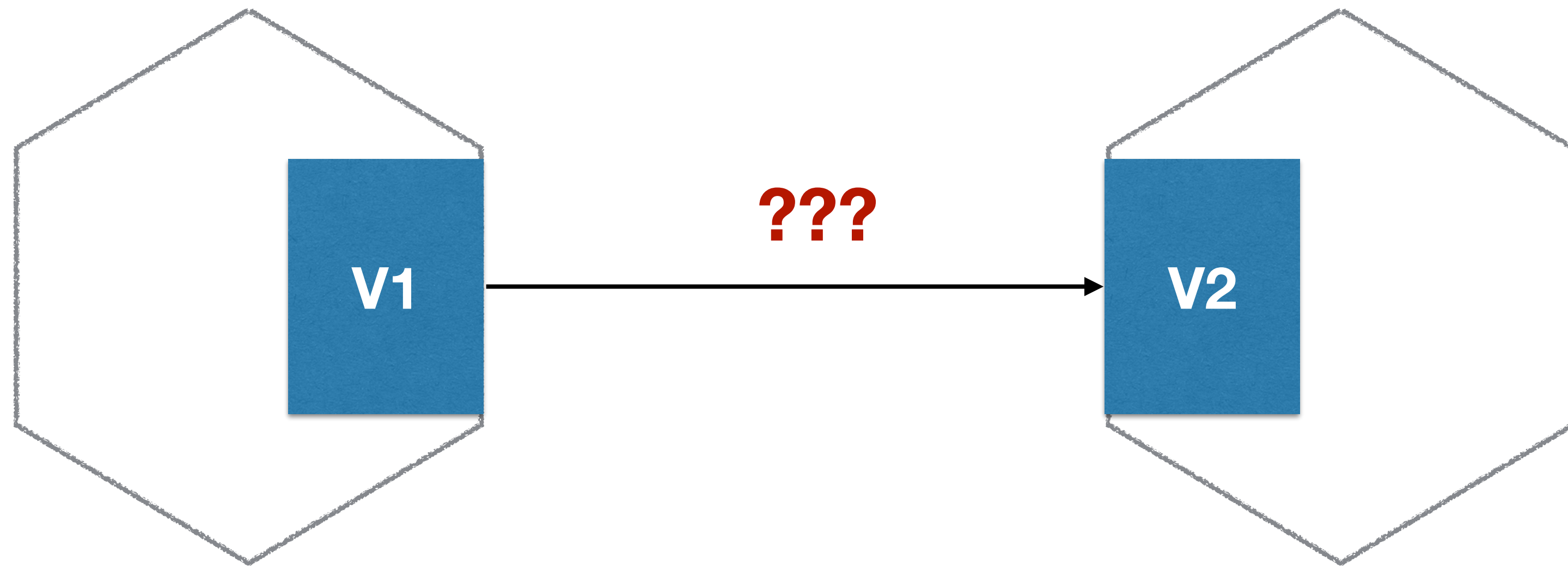
# VERSION DRIFT



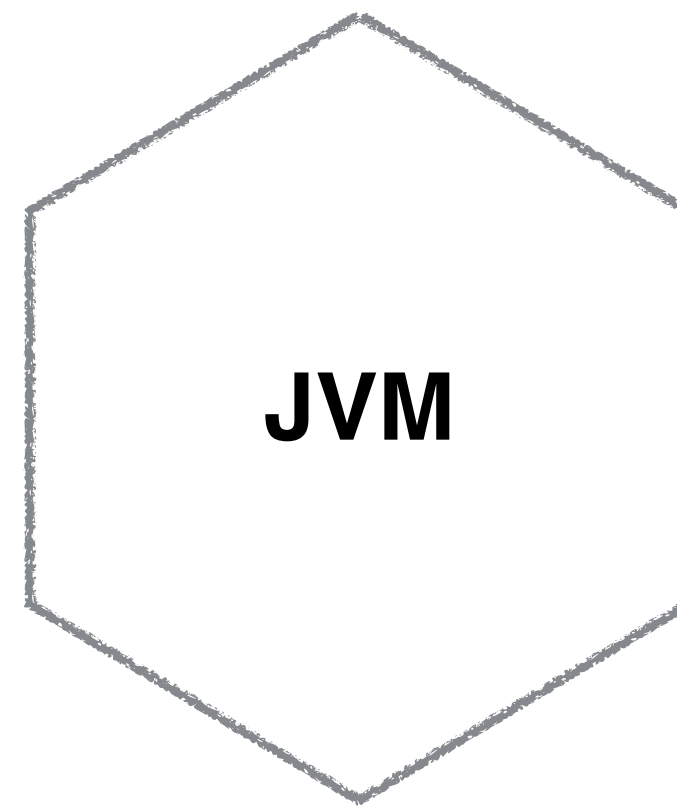
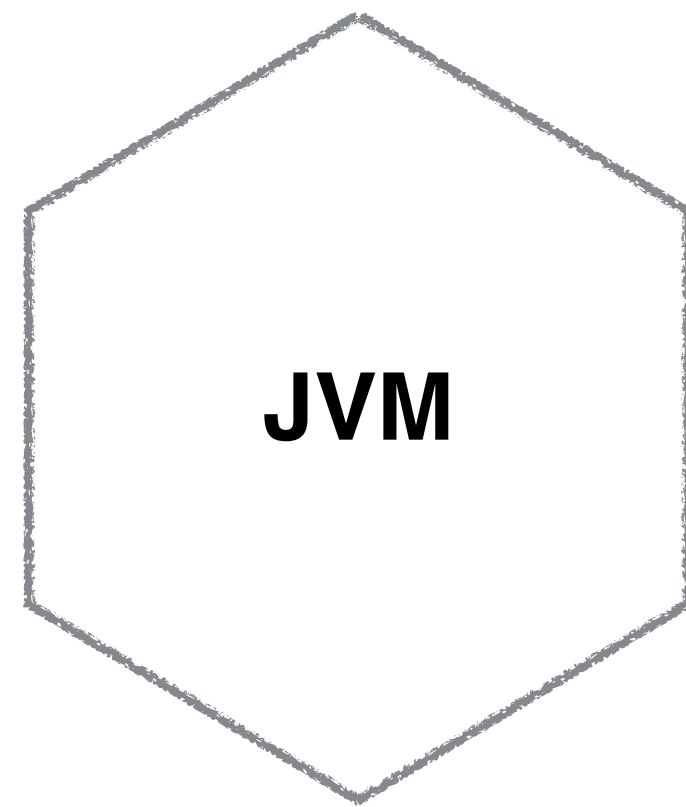
# VERSION DRIFT



# VERSION DRIFT

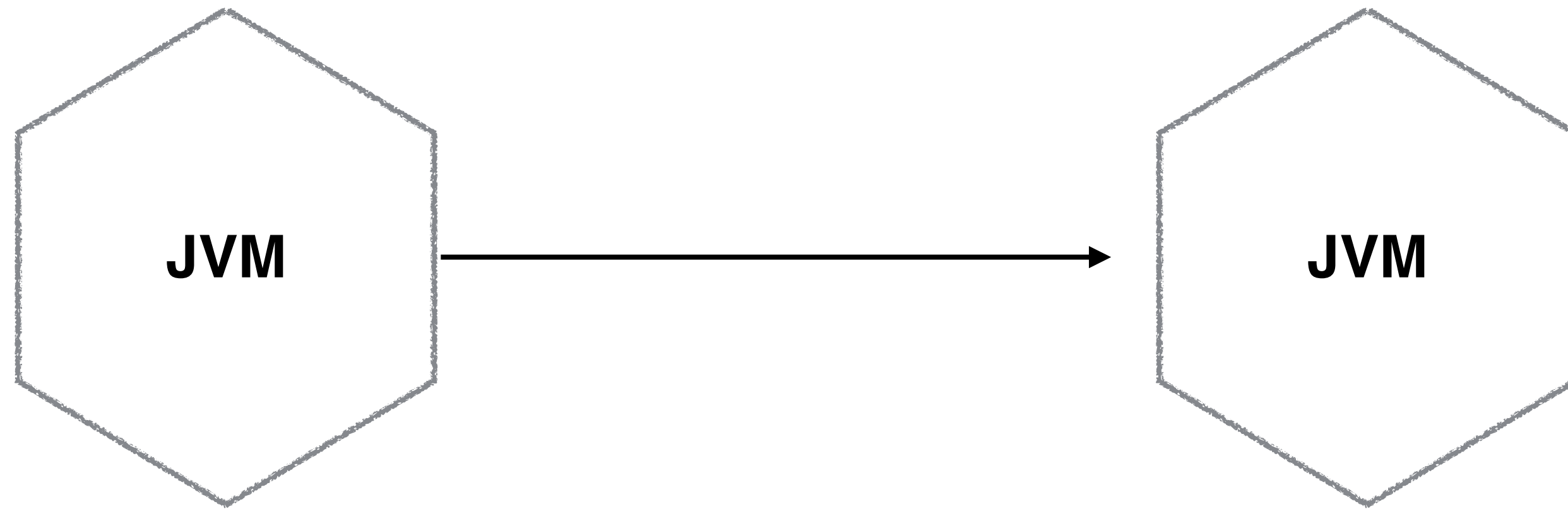


## NETFLIX - ENFORCEMENT OF REUSE



**NETFLIX**

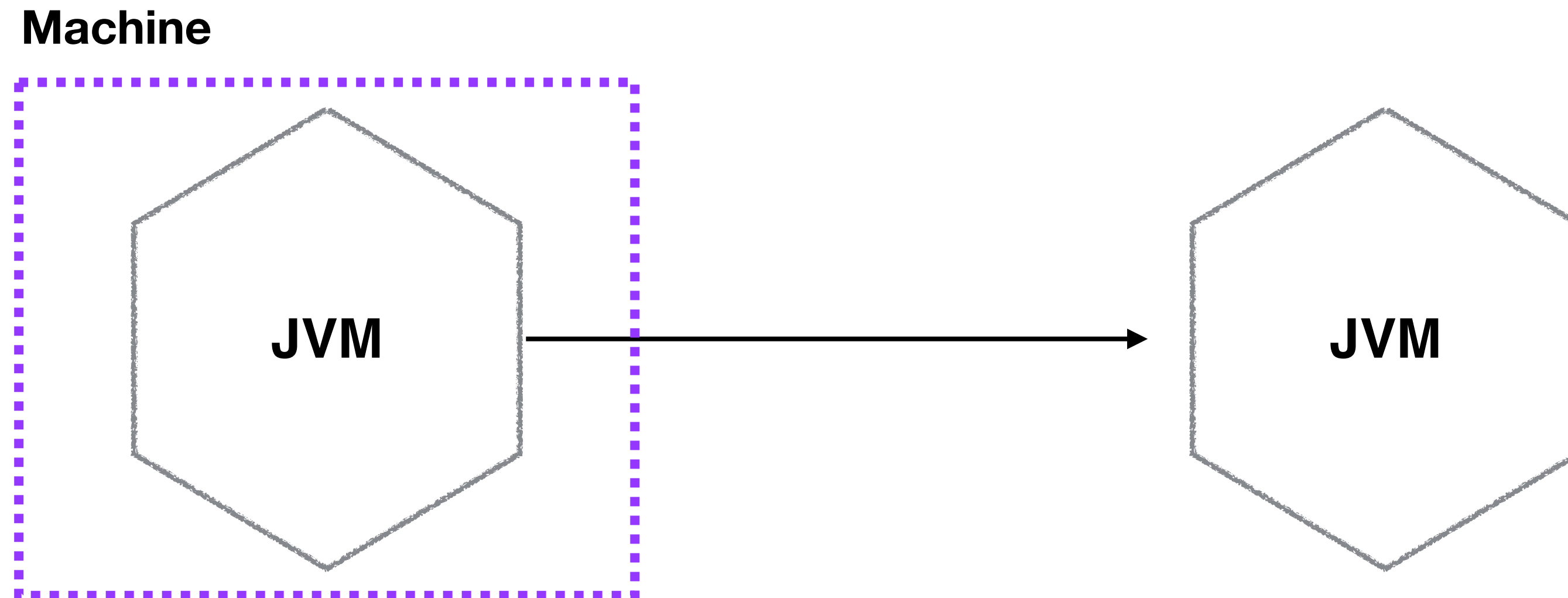
## NETFLIX - ENFORCEMENT OF REUSE



**NETFLIX**

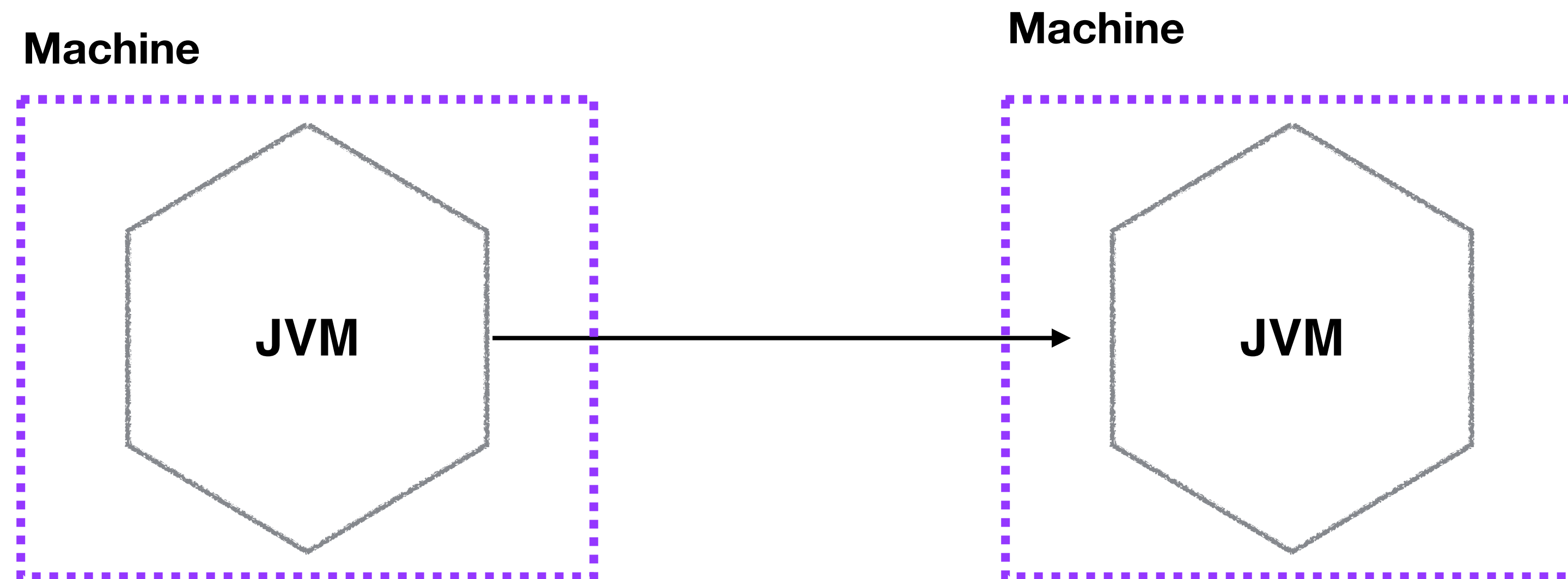


## NETFLIX - ENFORCEMENT OF REUSE



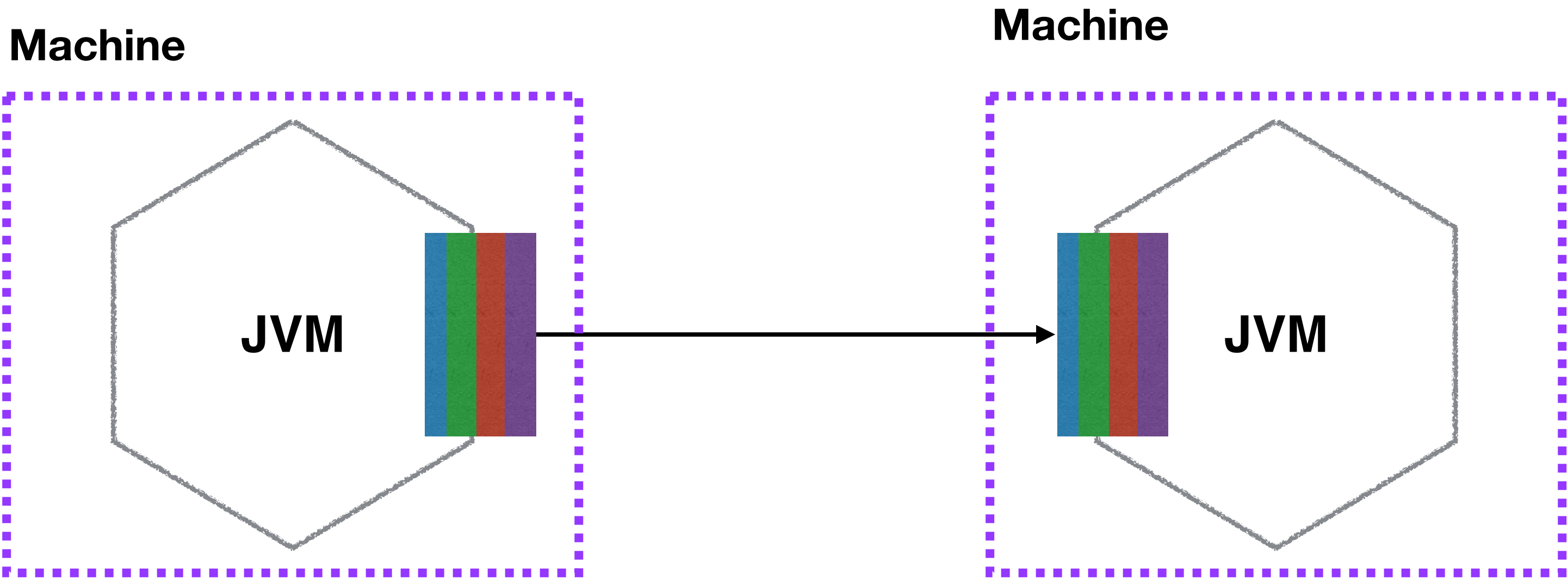
**NETFLIX**

## NETFLIX - ENFORCEMENT OF REUSE



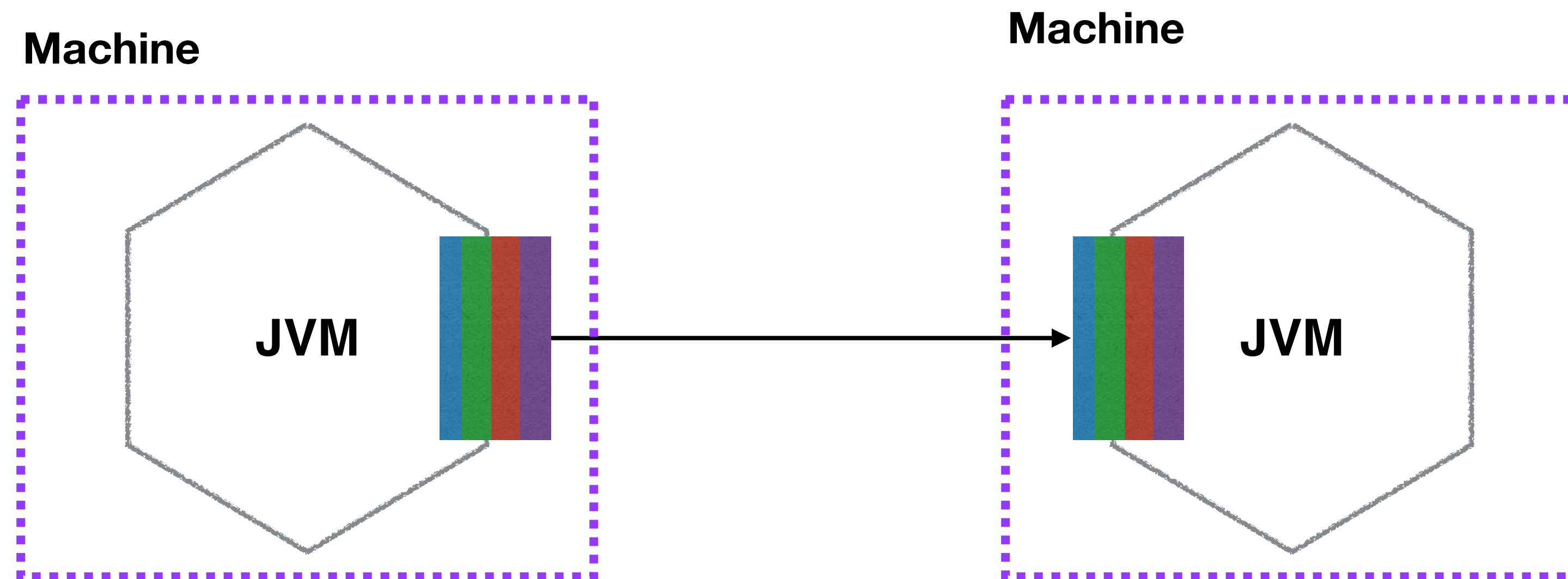
**NETFLIX**

NETFLIX - ENFORCEMENT OF REUSE



NETFLIX

## NETFLIX - ENFORCEMENT OF REUSE

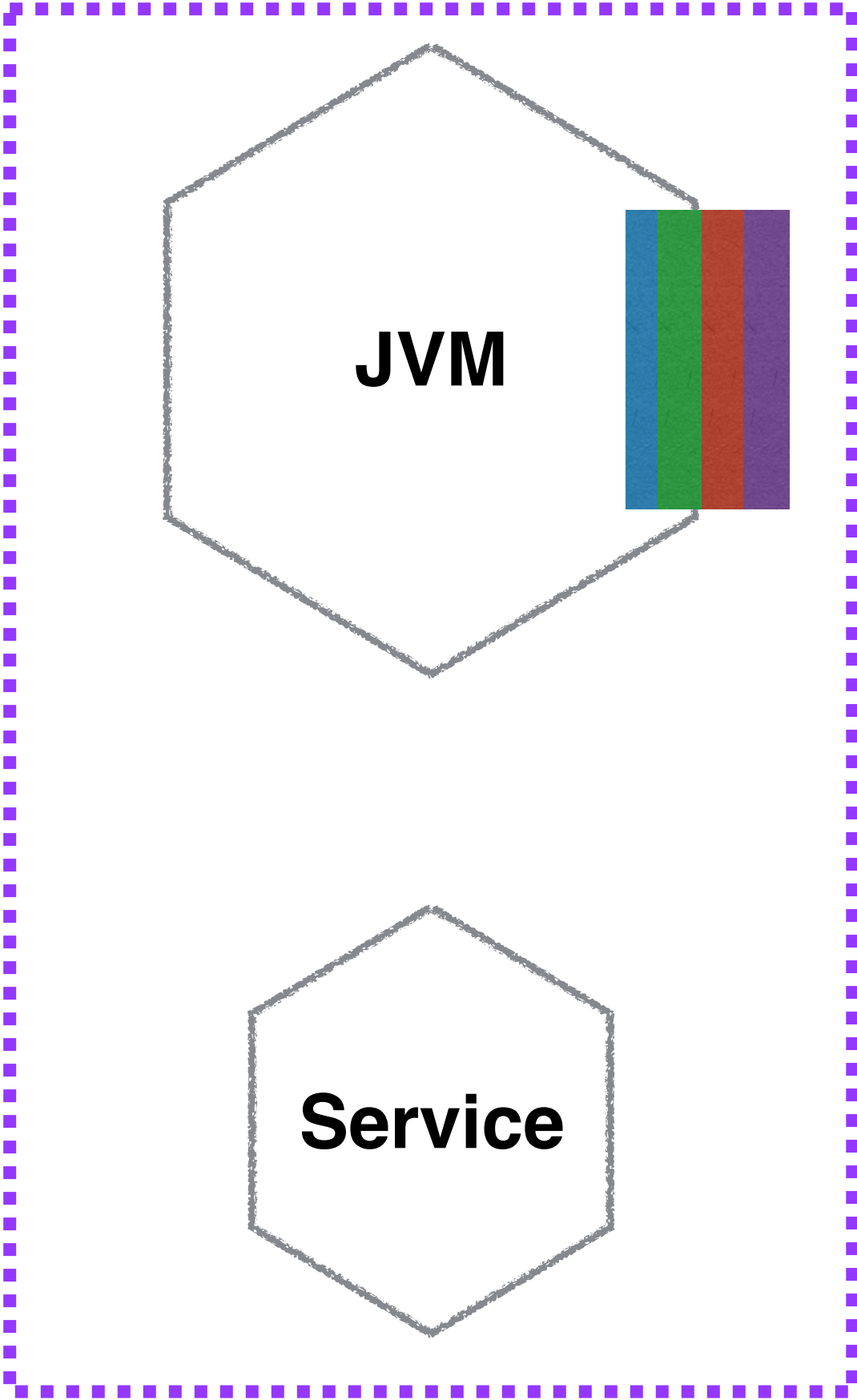


# NETFLIX

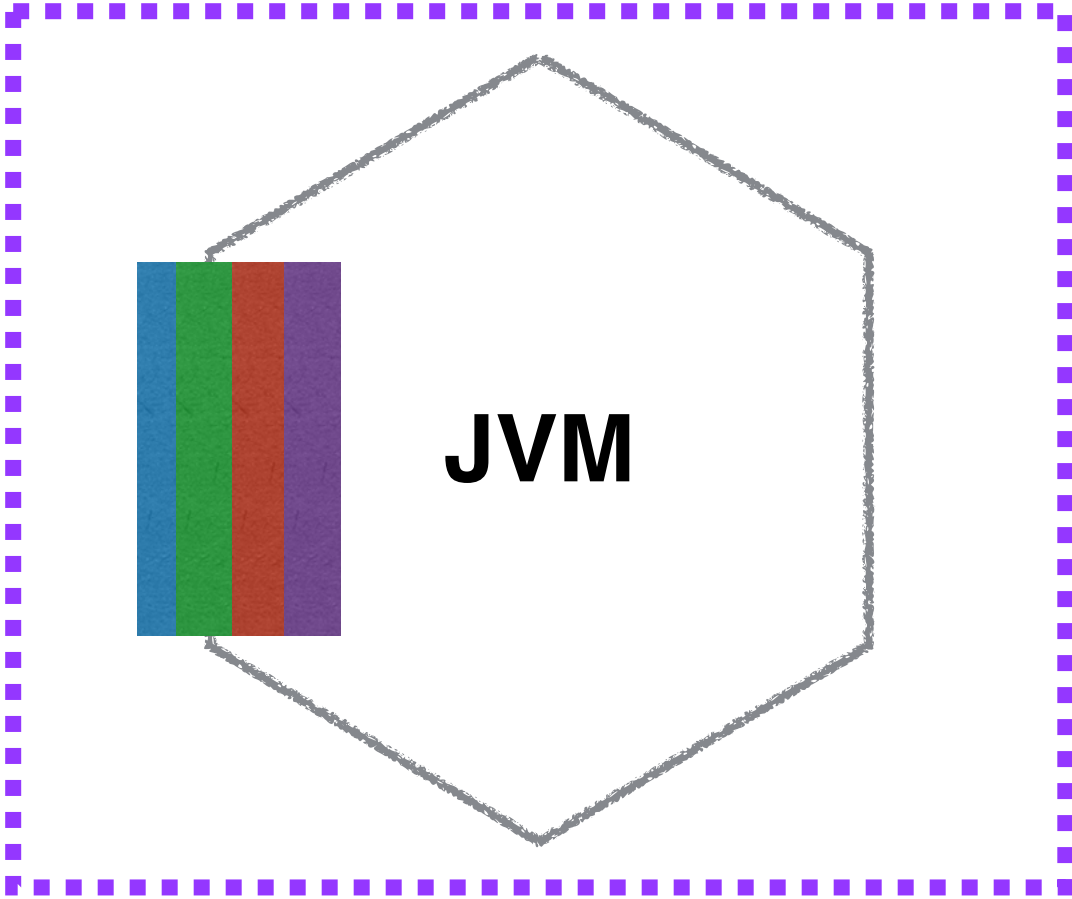
What about non-JVM  
languages?

# NETFLIX - SIDECAR PATTERN

Machine



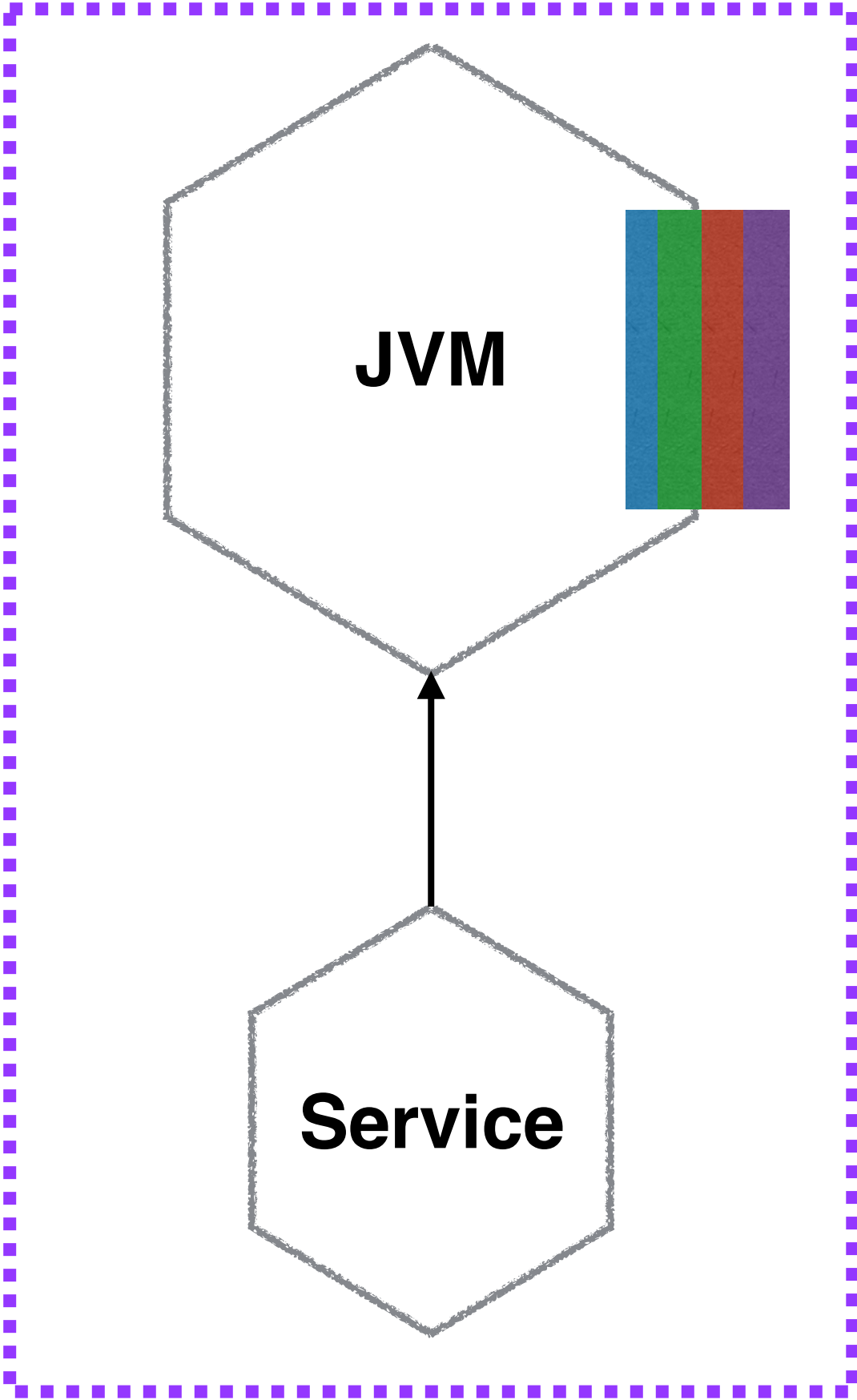
Machine



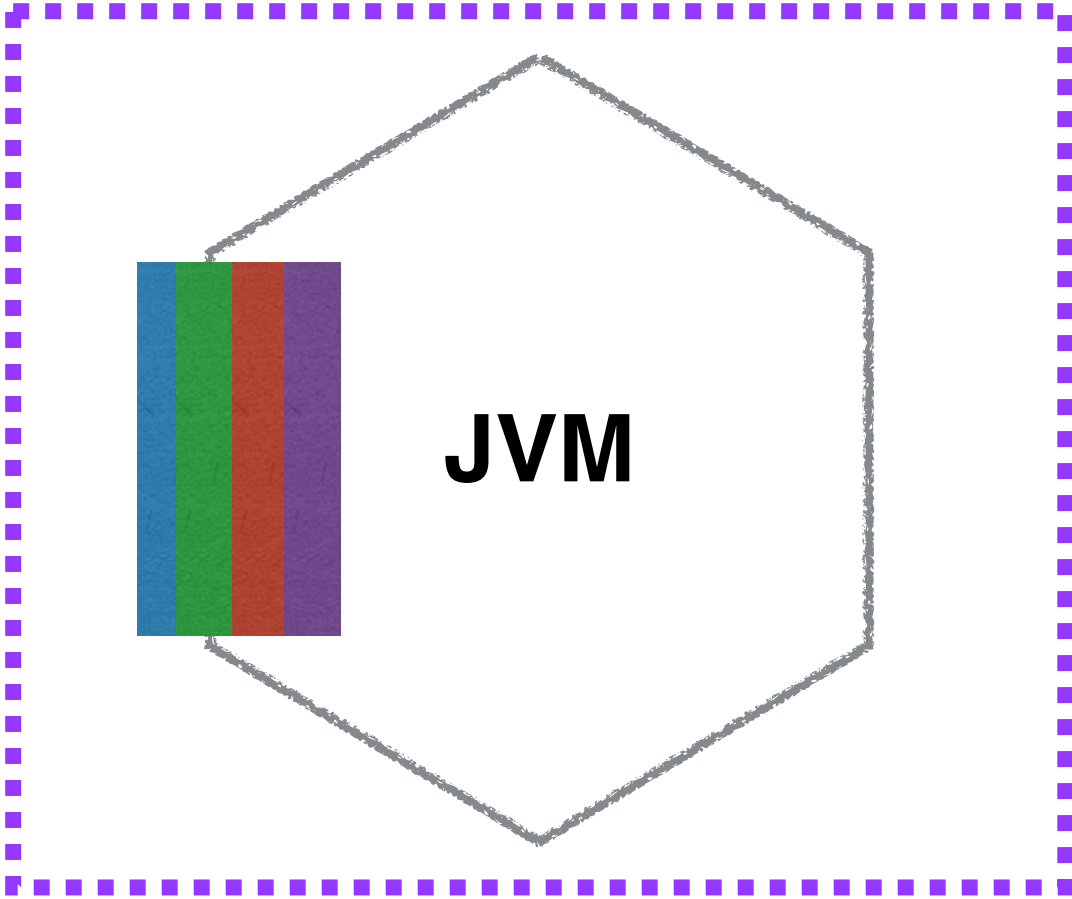


# NETFLIX - SIDECAR PATTERN

Machine

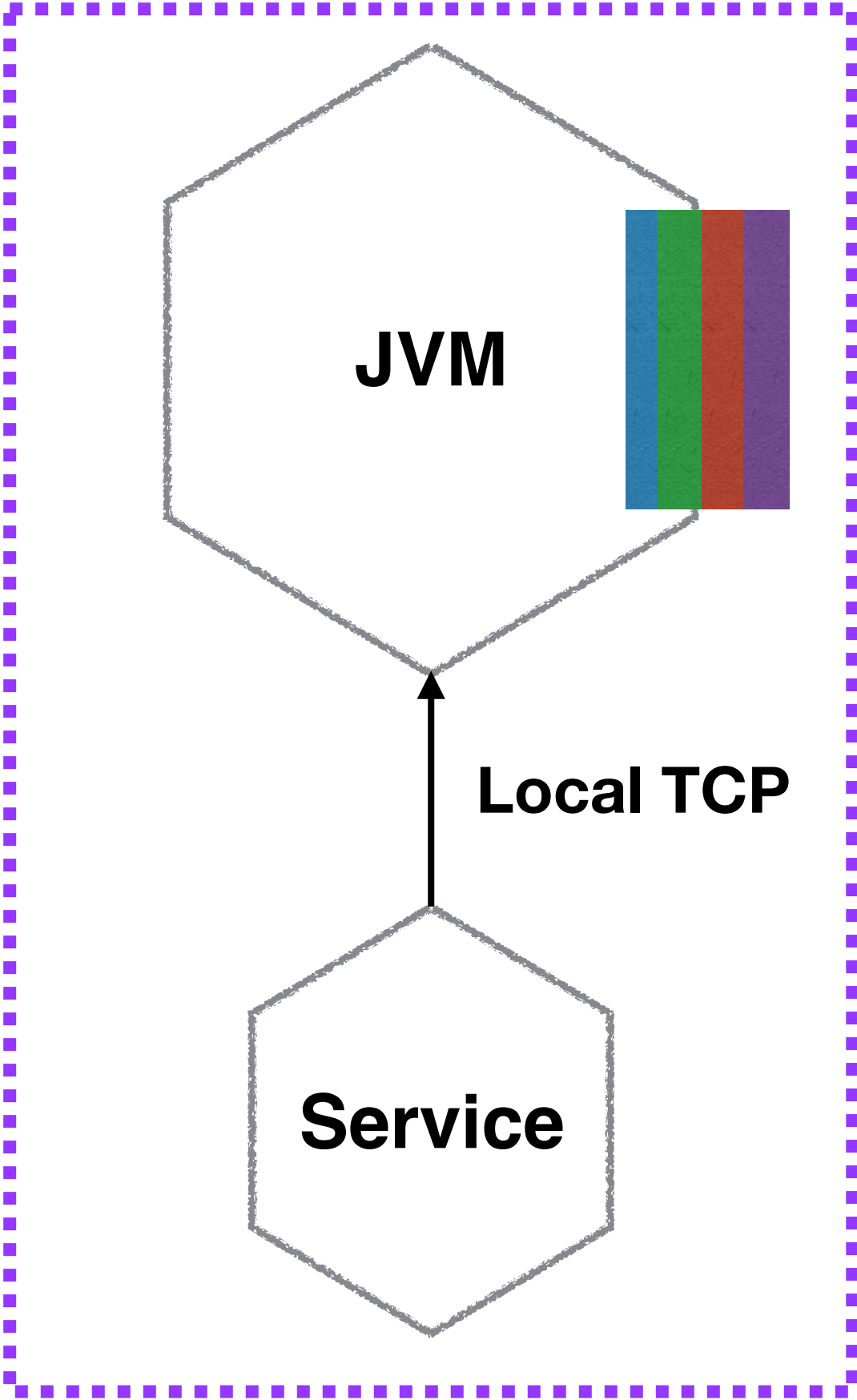


Machine

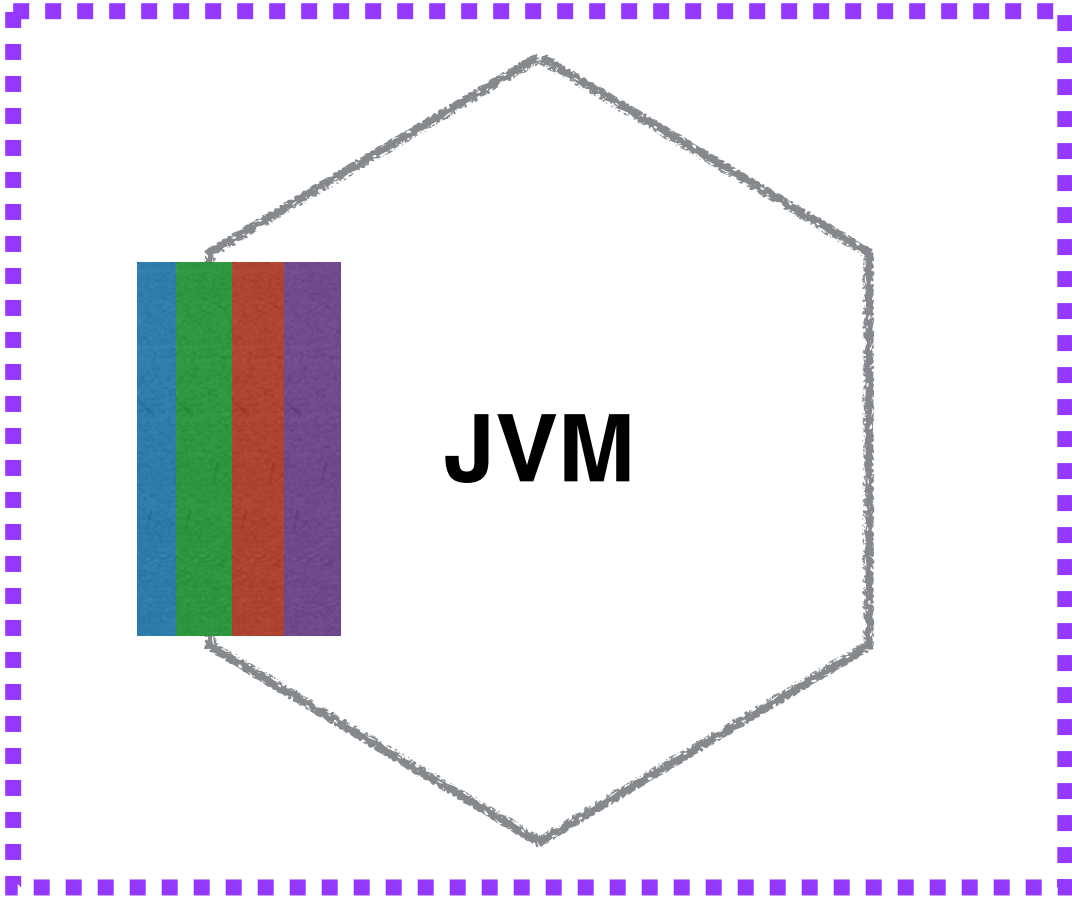


# NETFLIX - SIDECAR PATTERN

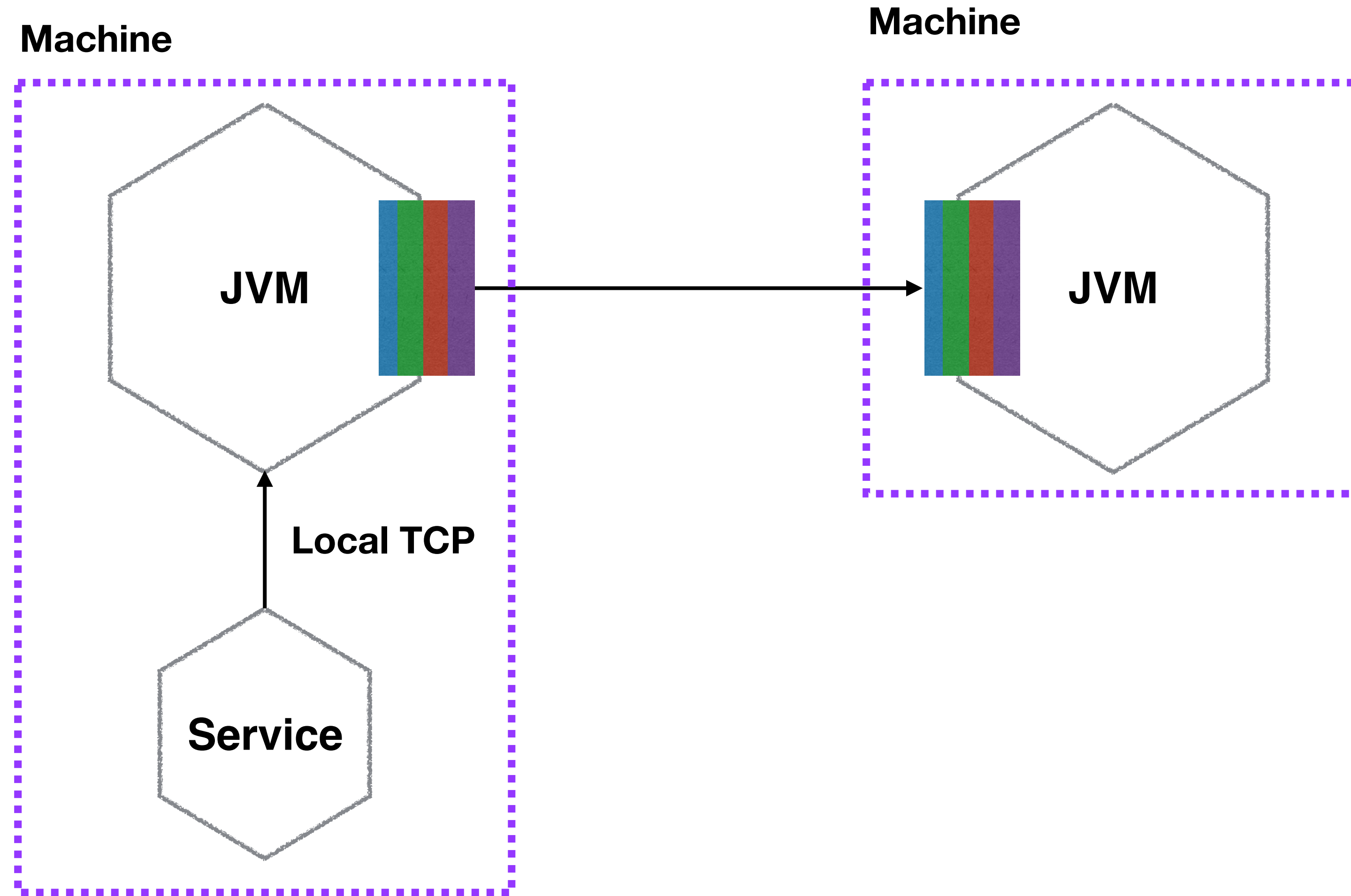
Machine



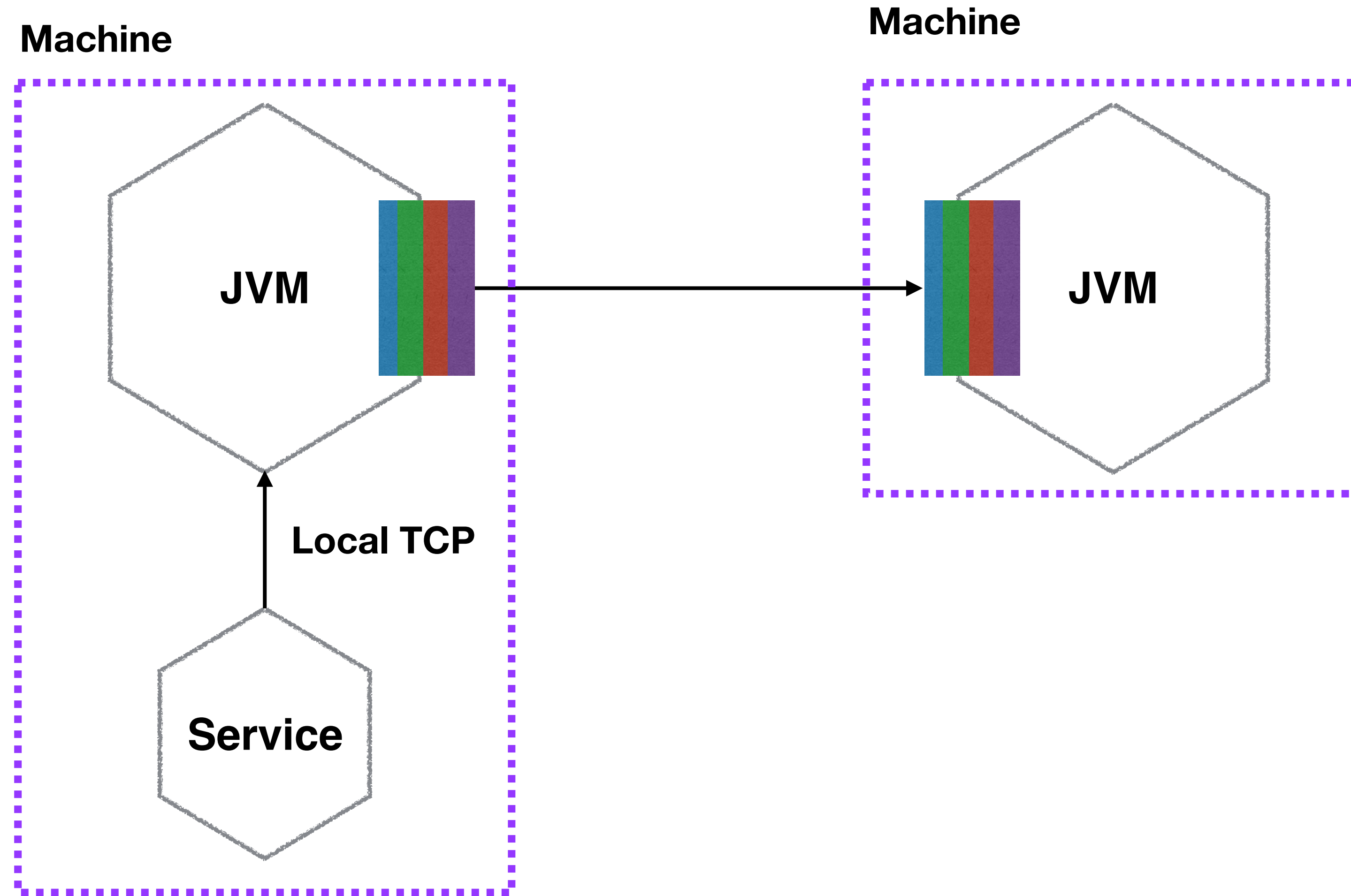
Machine



# NETFLIX - SIDECAR PATTERN

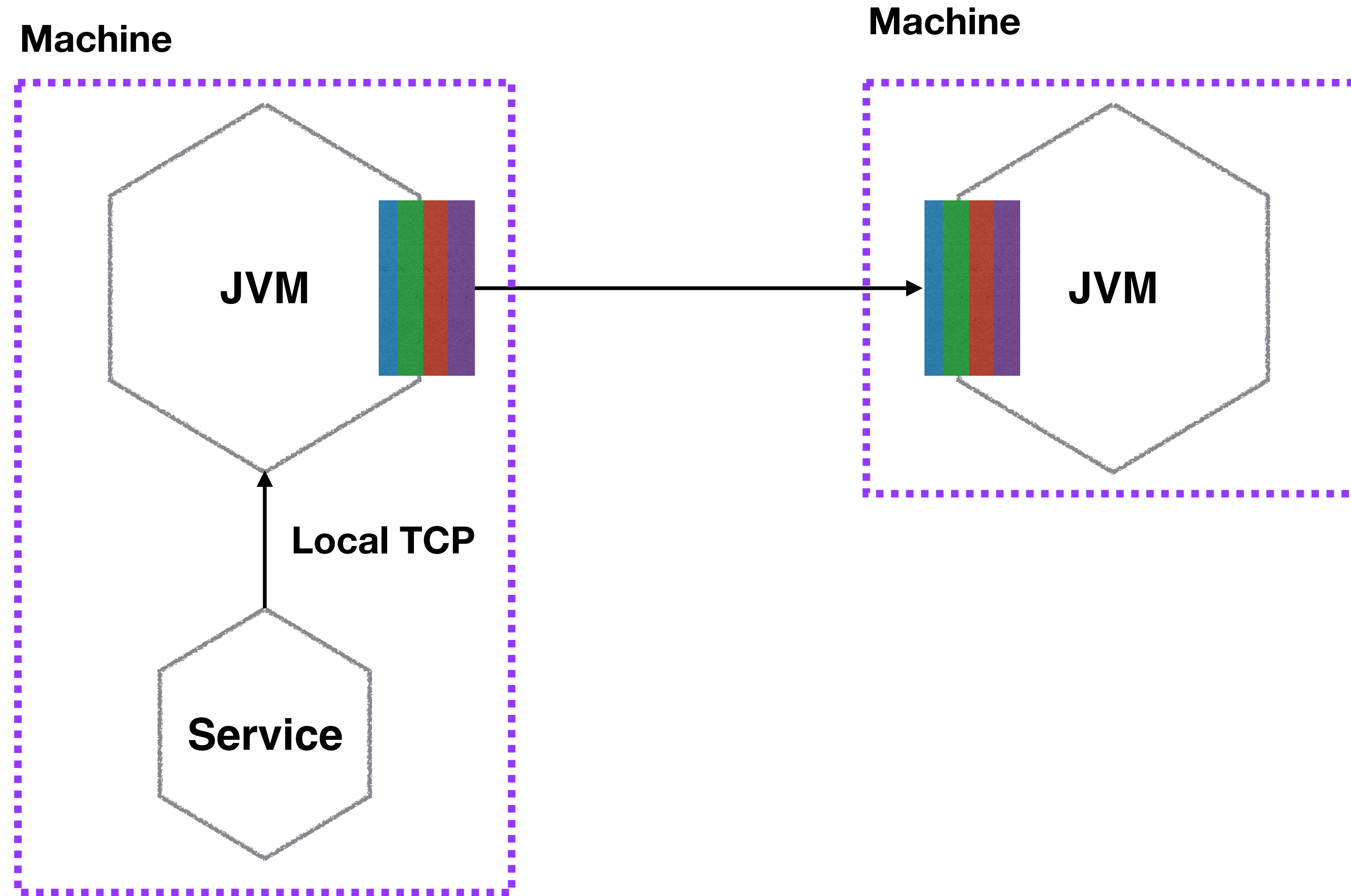


# NETFLIX - SIDECAR PATTERN



Re-use code across tech stacks

# NETFLIX - SIDECAR PATTERN

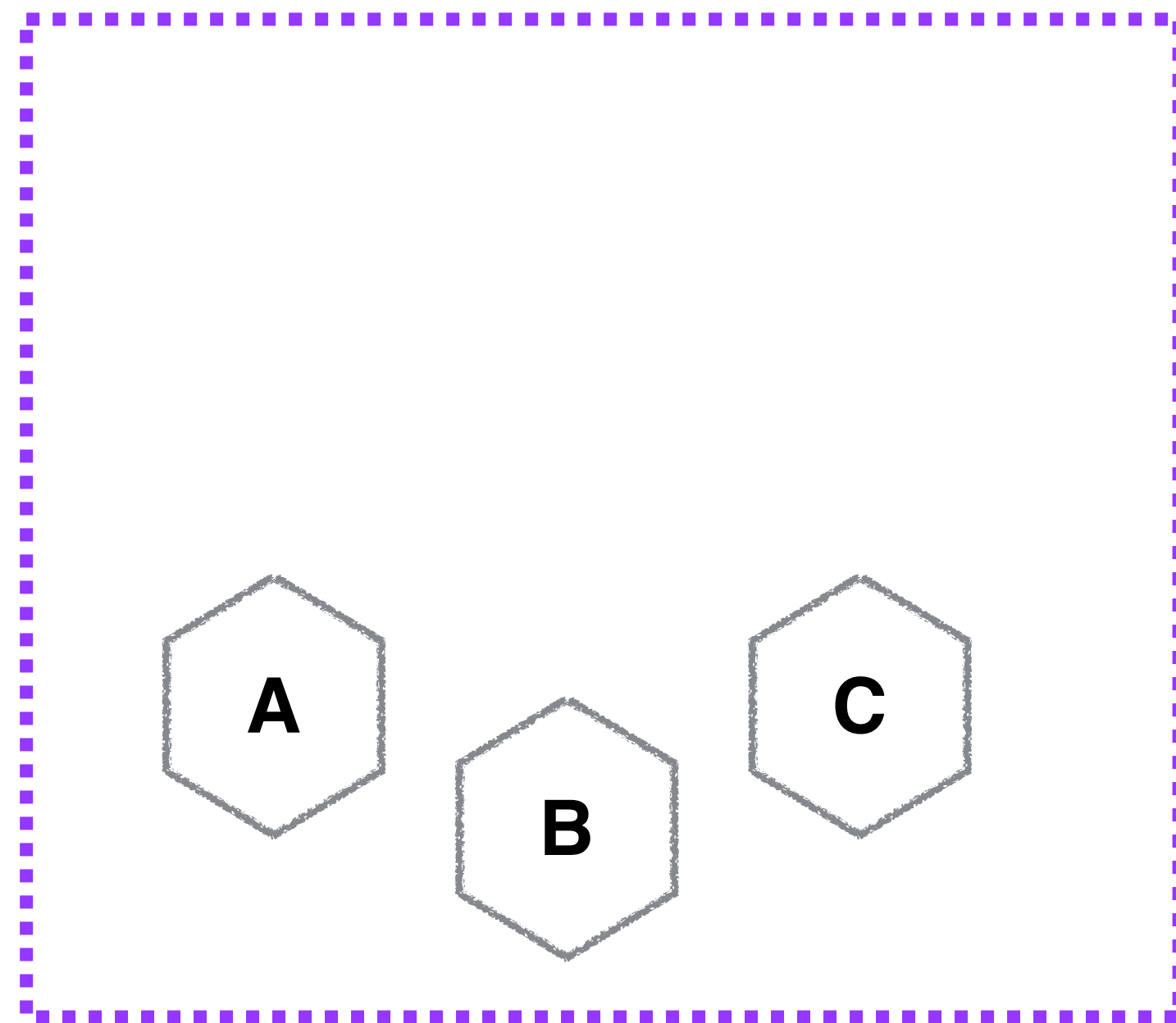


Re-use code across tech stacks

Reduce impact of version drift

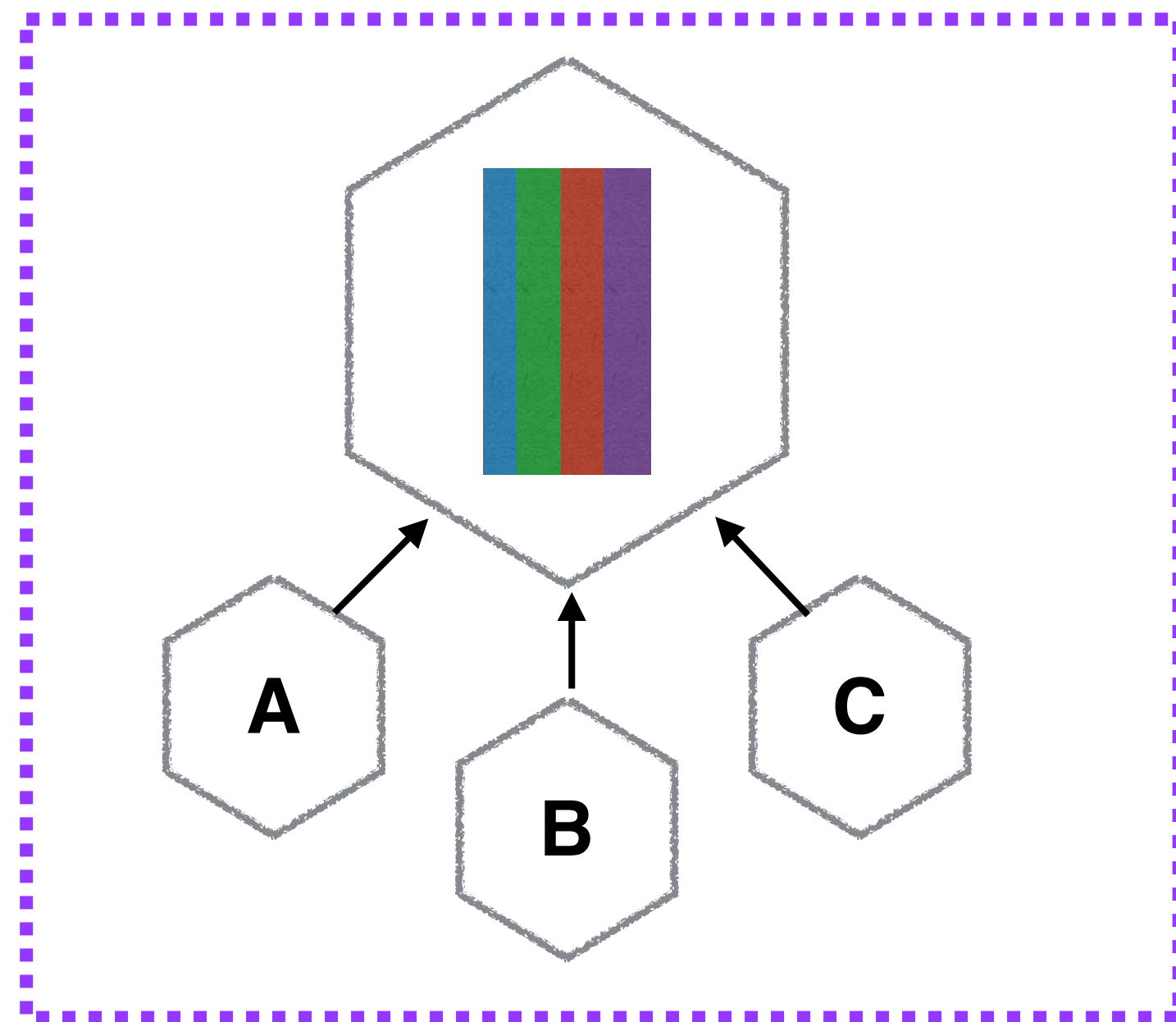


# FROM PROXIES TO SERVICE MESHES



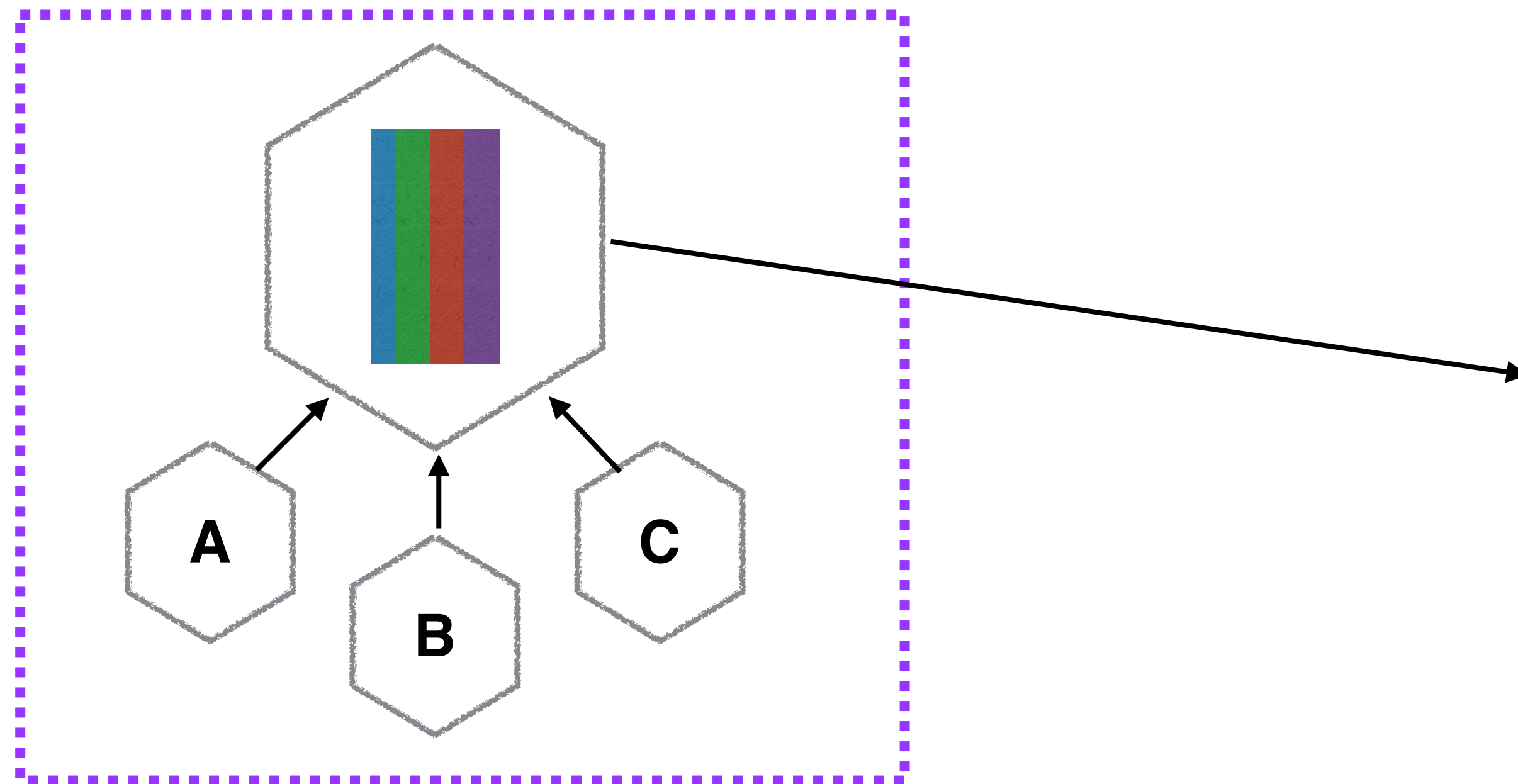
**Machine**

# FROM PROXIES TO SERVICE MESHES



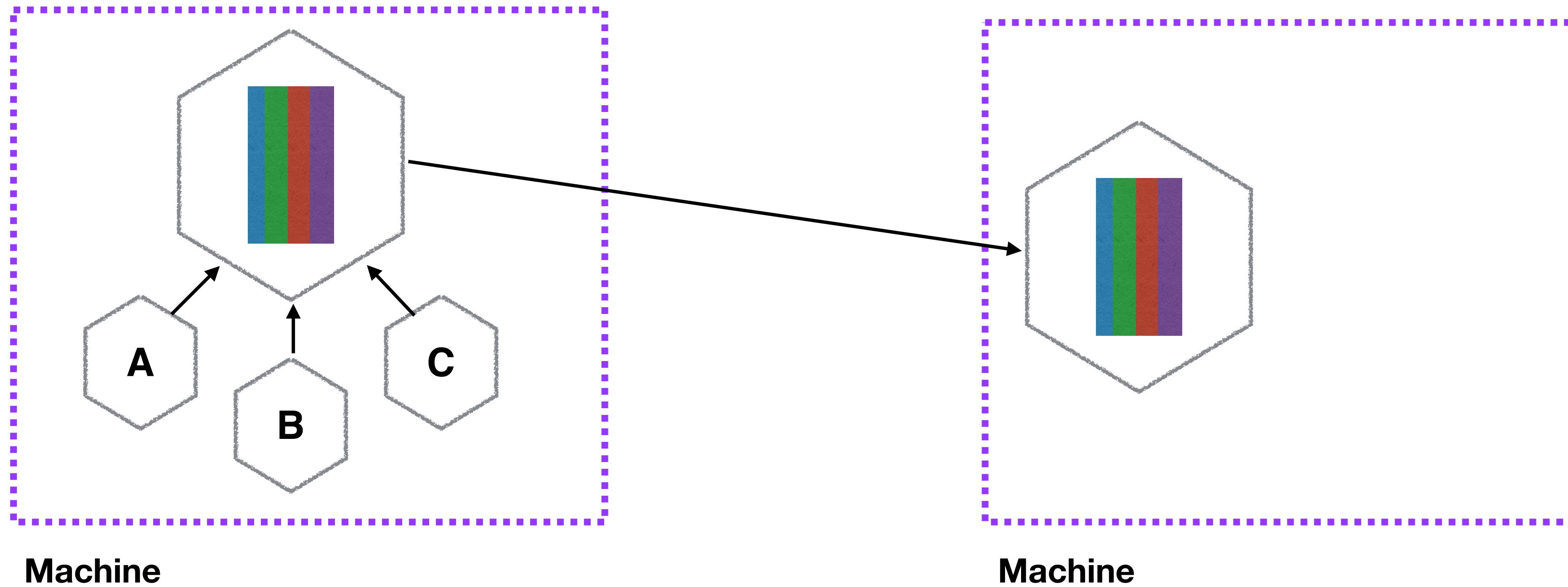
**Machine**

# FROM PROXIES TO SERVICE MESHES

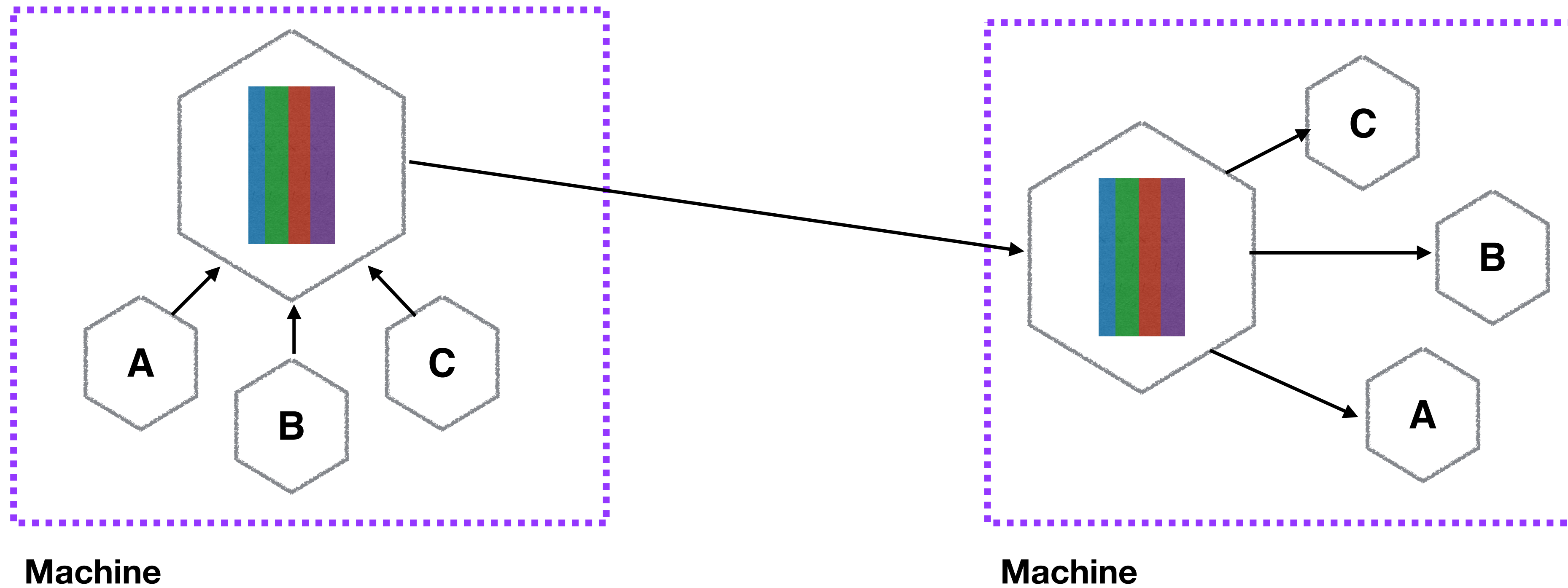


**Machine**

# FROM PROXIES TO SERVICE MESHES

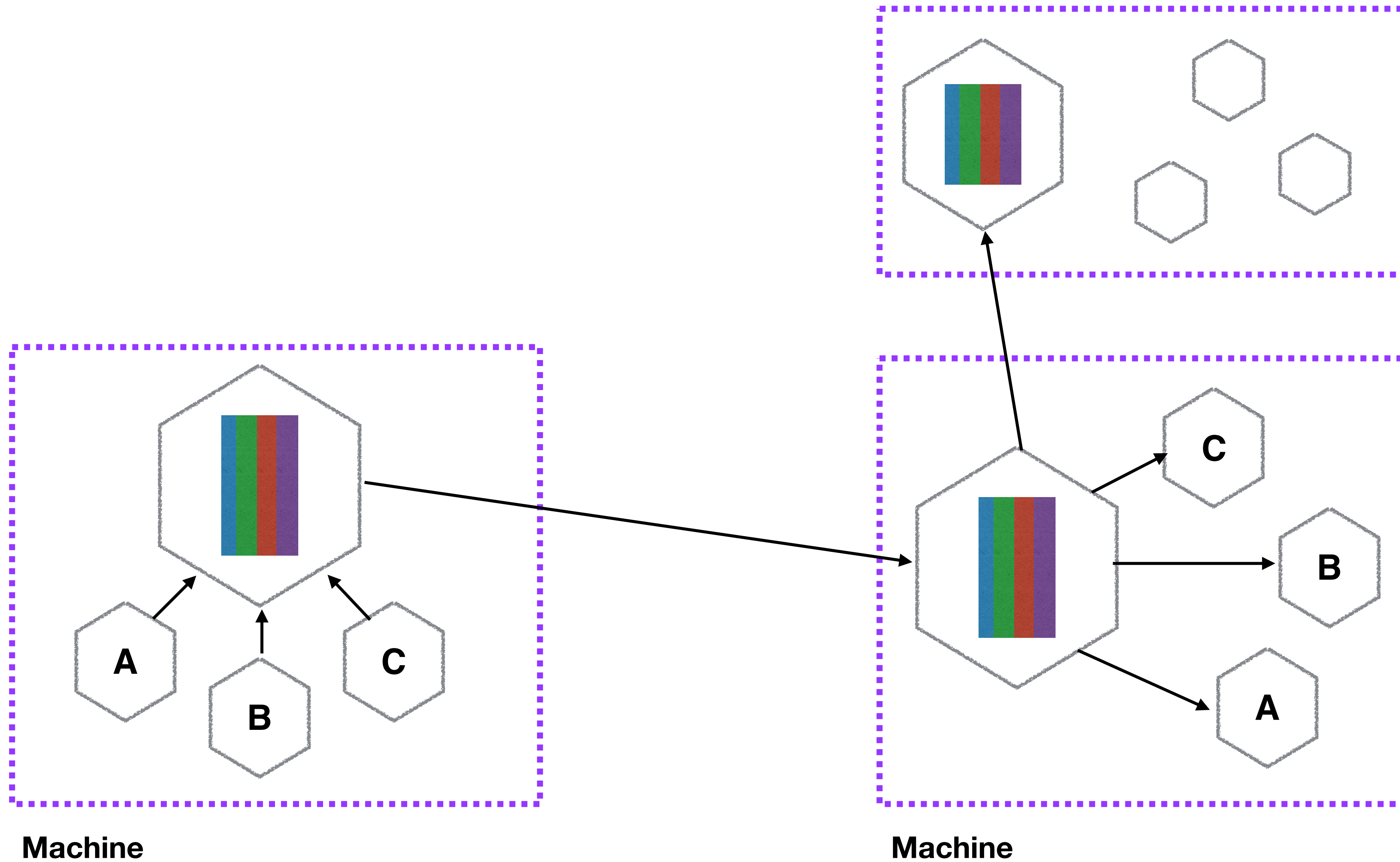


# FROM PROXIES TO SERVICE MESHES

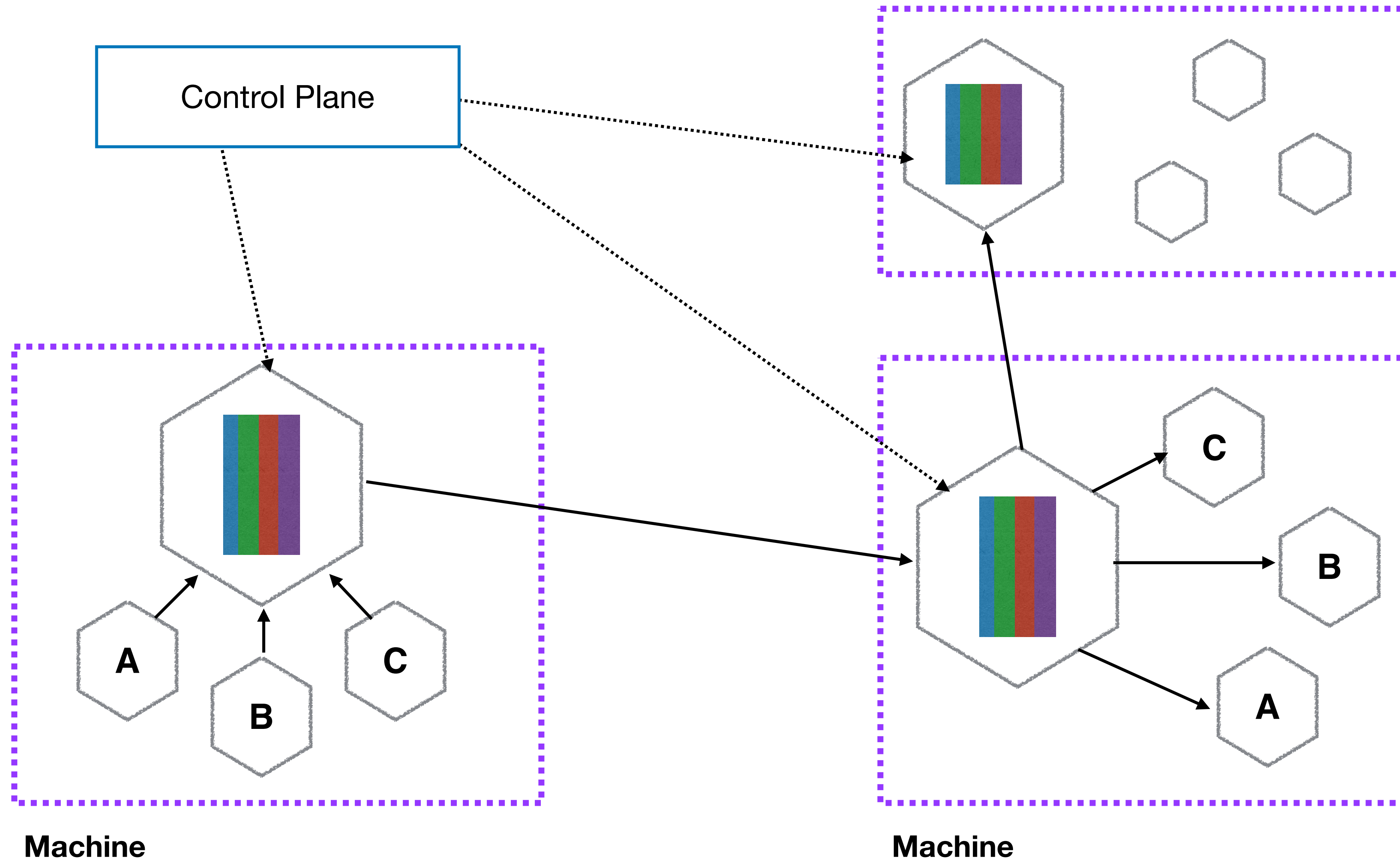




# FROM PROXIES TO SERVICE MESHES



# FROM PROXIES TO SERVICE MESHES



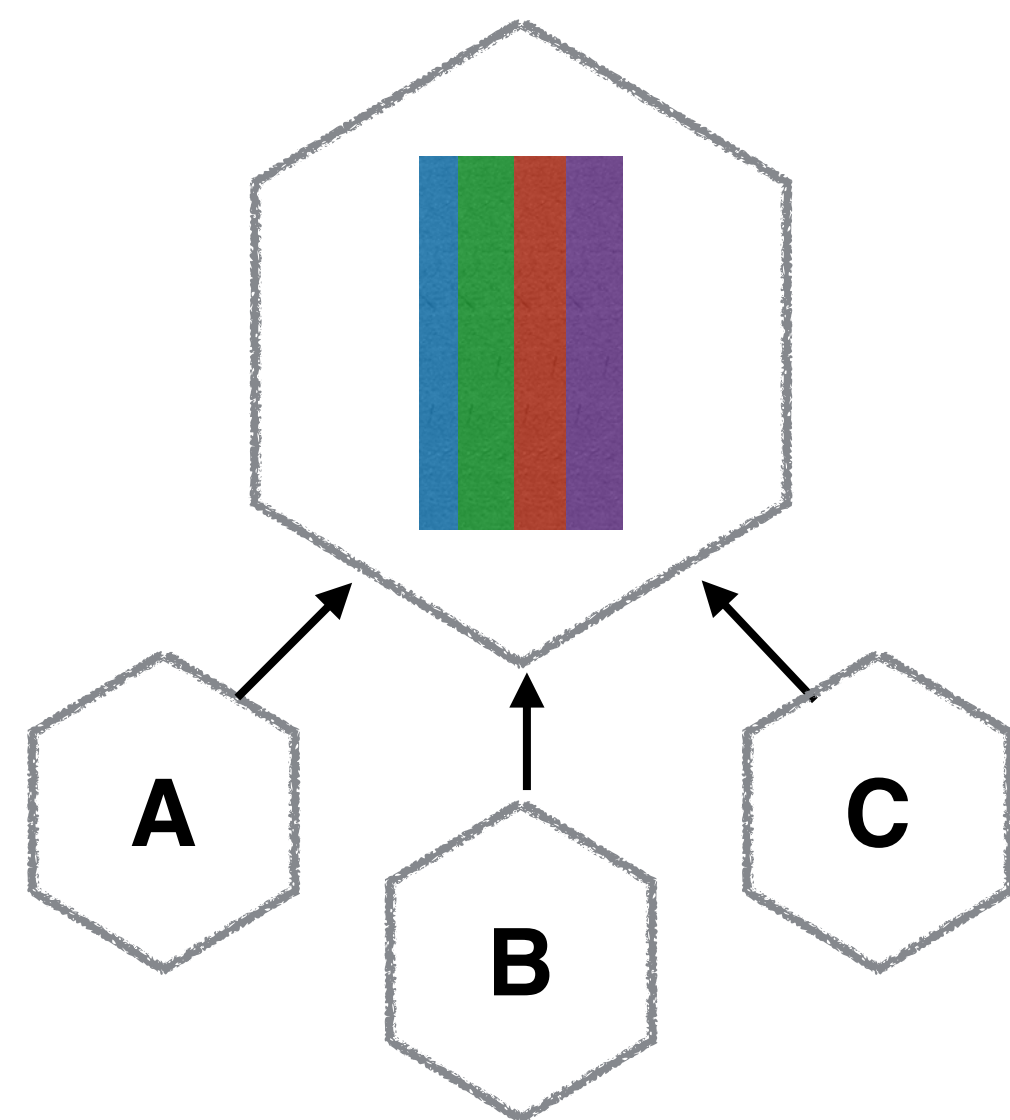
# SIDECARS VS PROXIES

Local Proxy

Sidecar

# SIDECARS VS PROXIES

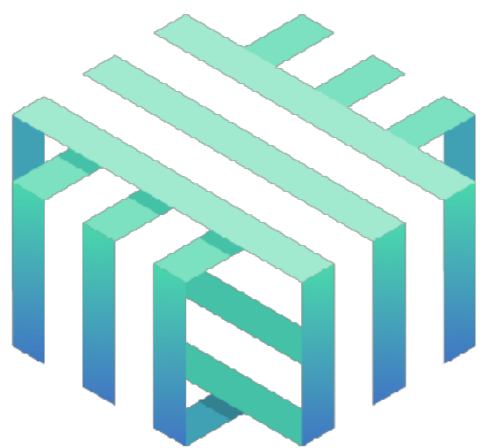
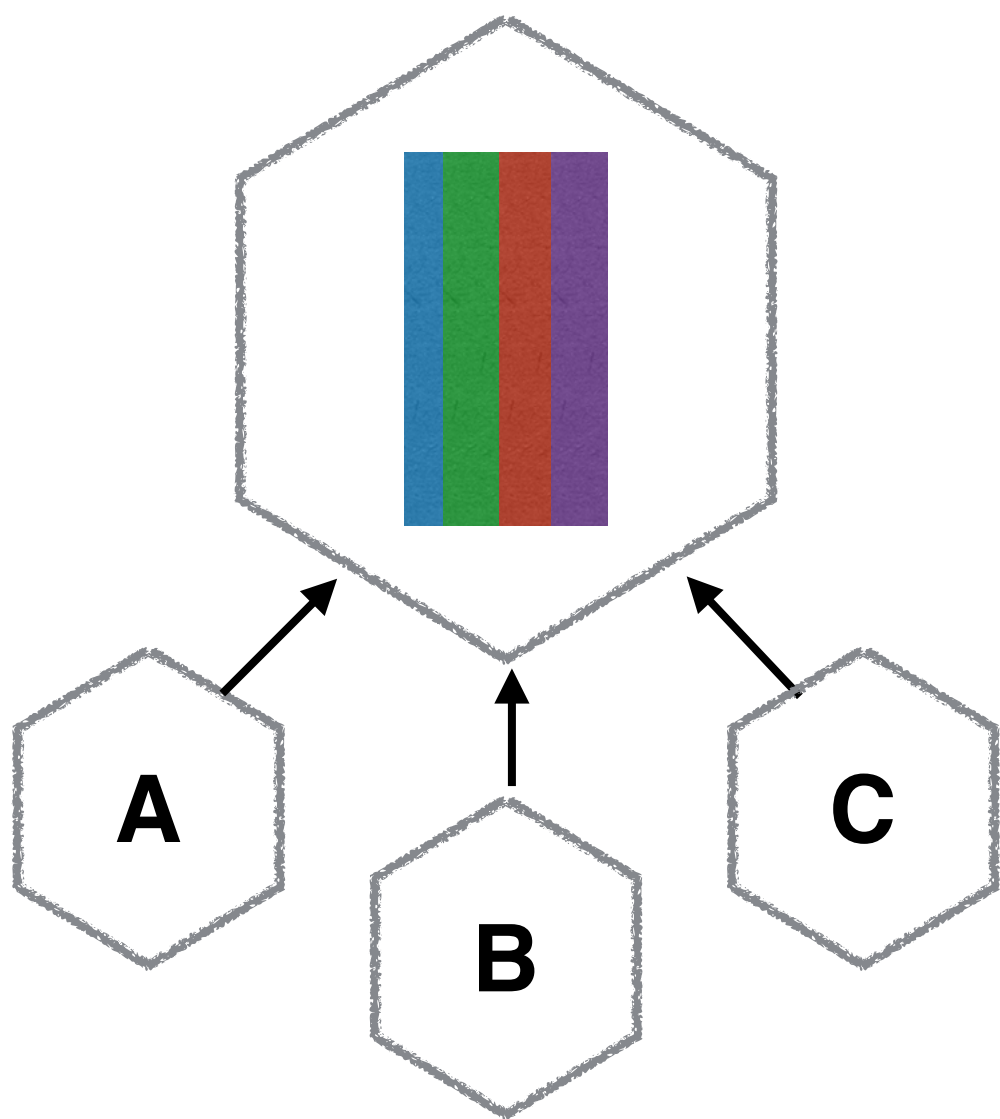
## Local Proxy



## Sidecar

# SIDECARS VS PROXIES

## Local Proxy



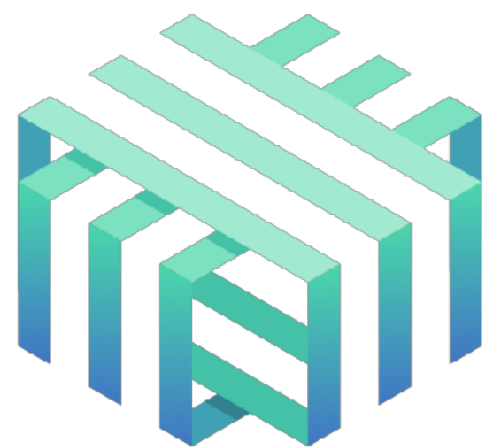
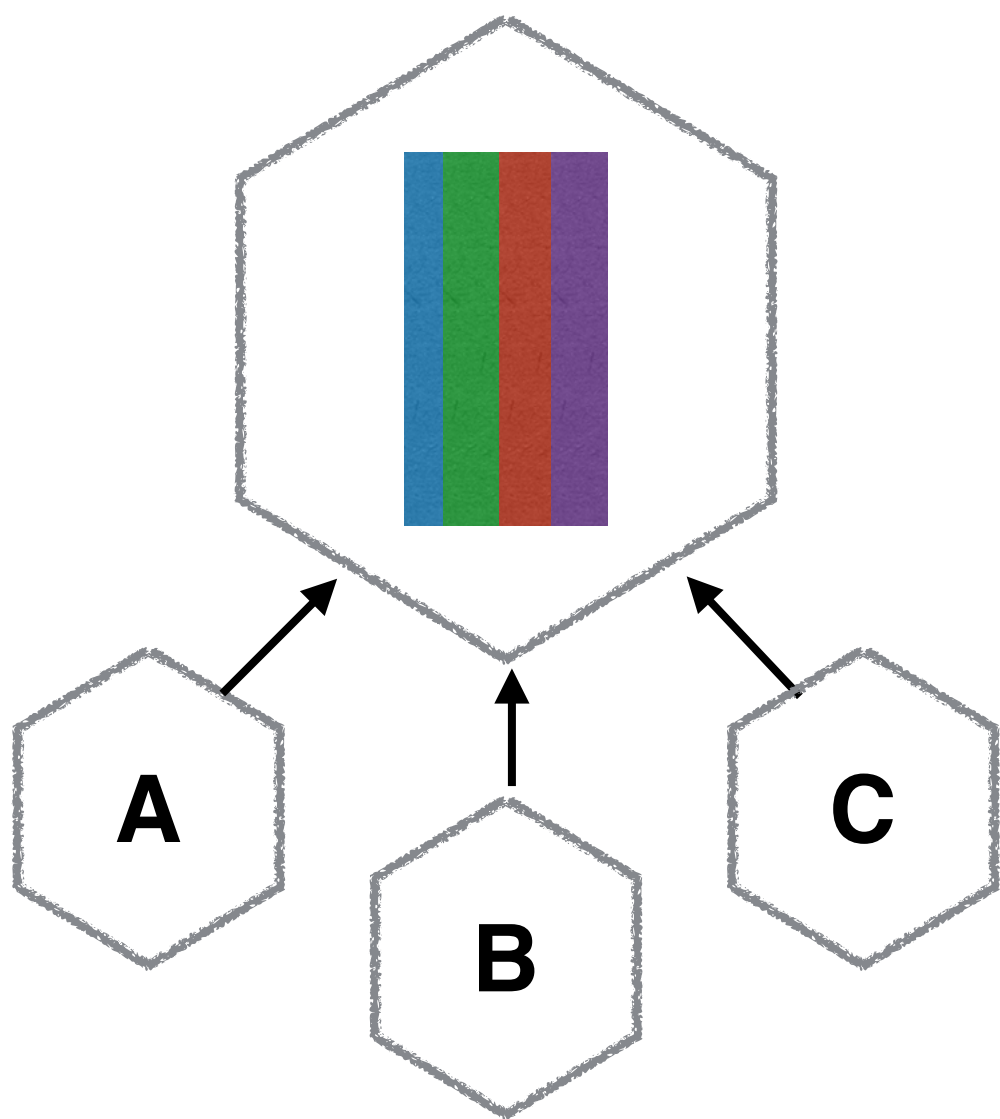
Linkerd

## Sidecar



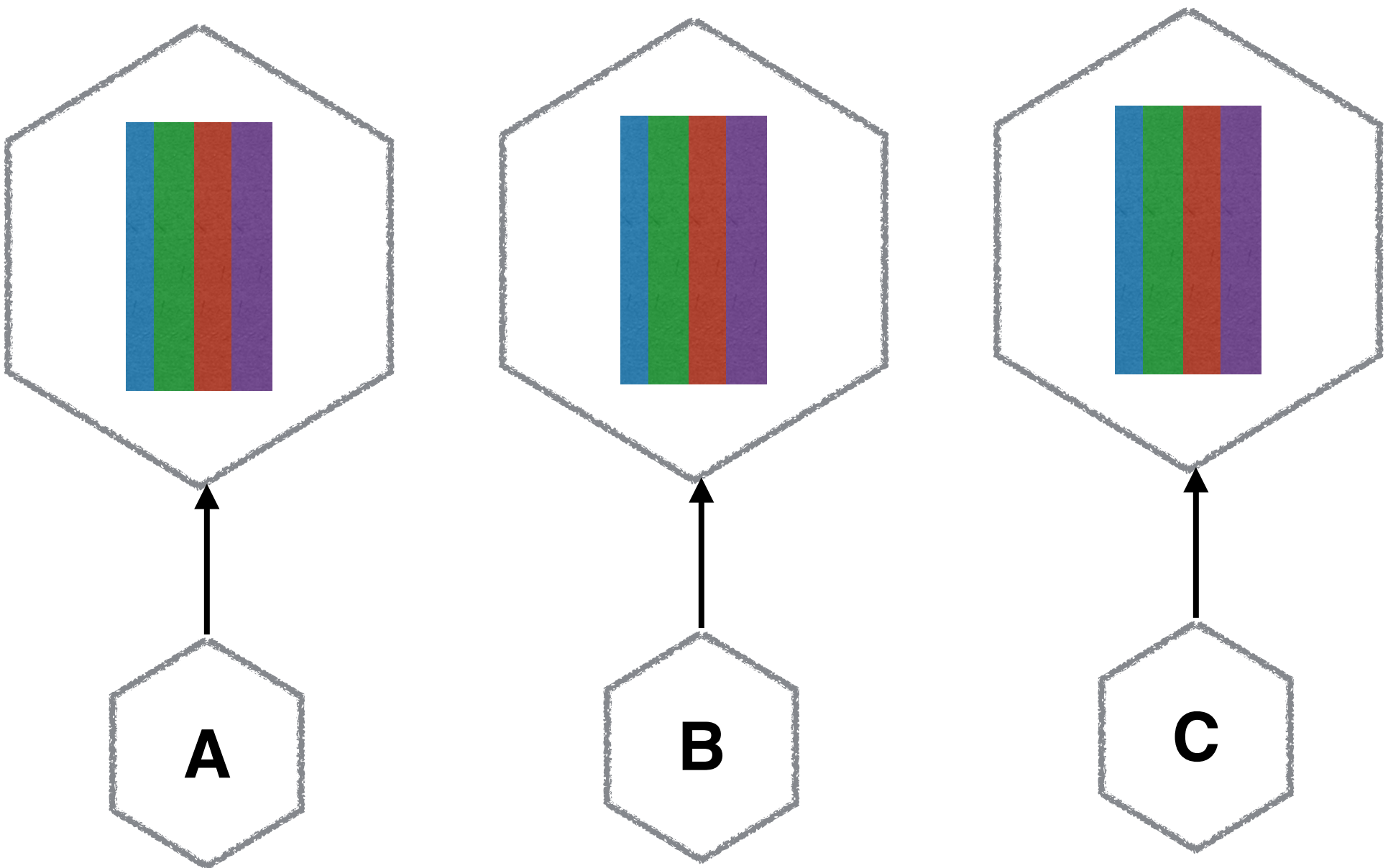
# SIDECARS VS PROXIES

## Local Proxy



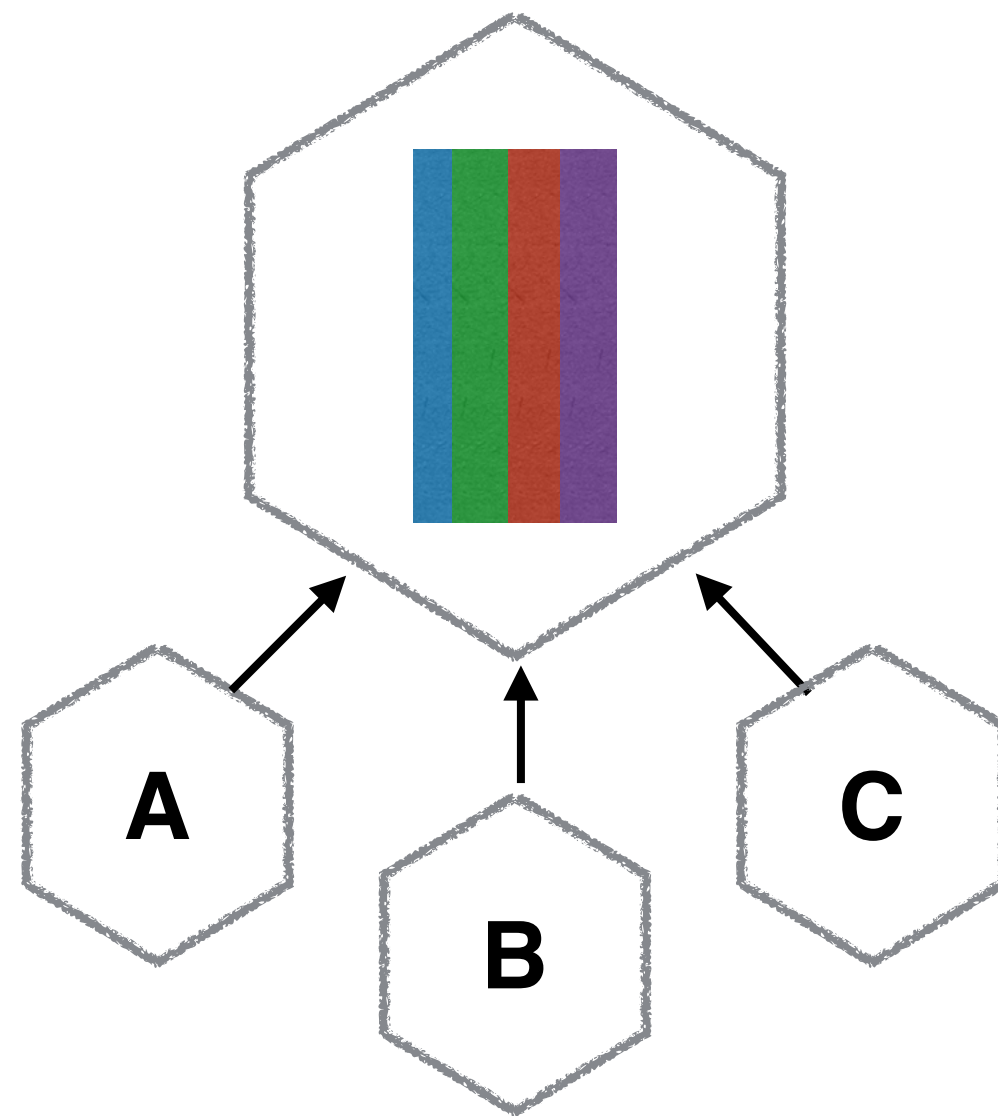
Linkerd

## Sidecar



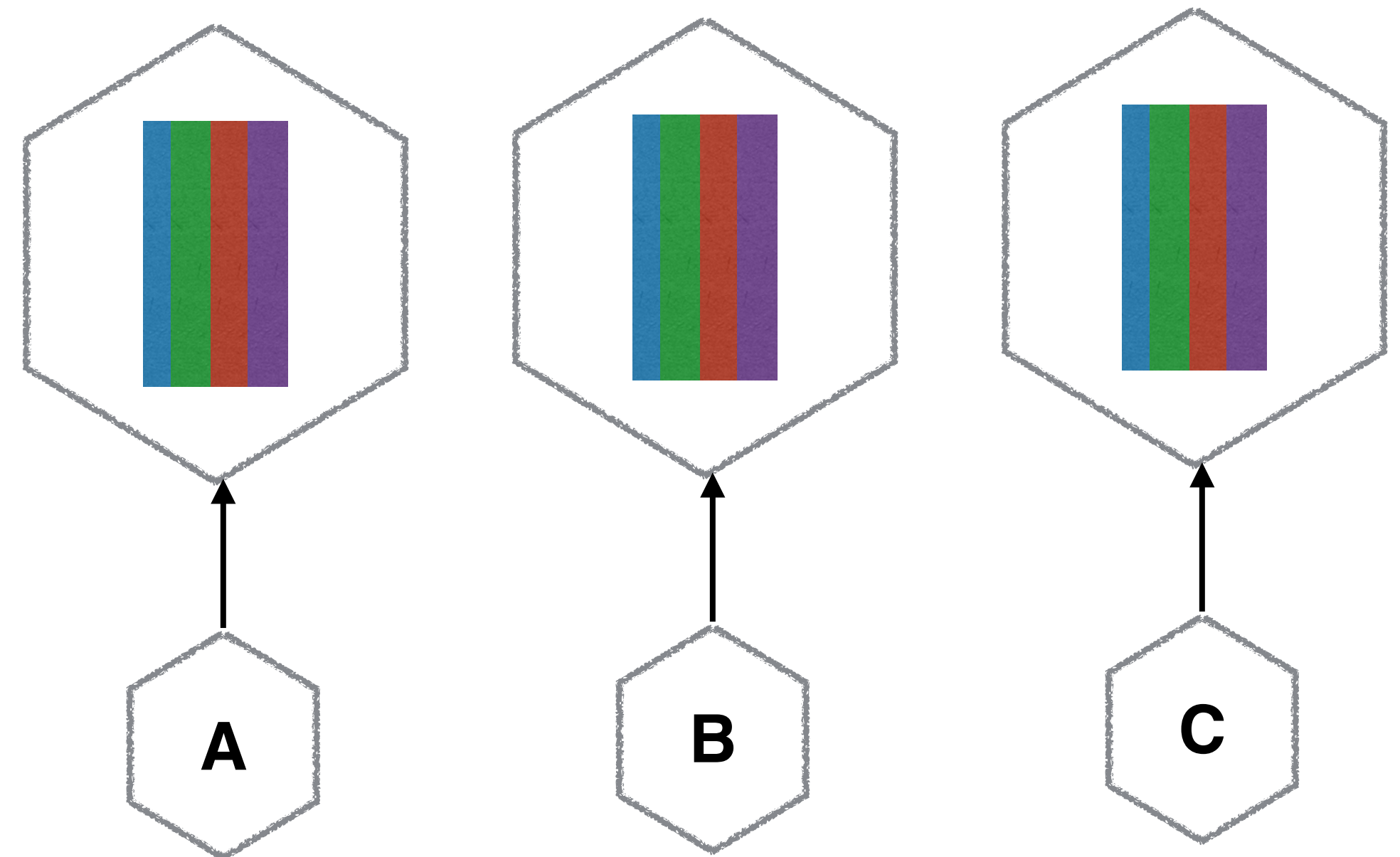
# SIDECARS VS PROXIES

## Local Proxy



Linkerd

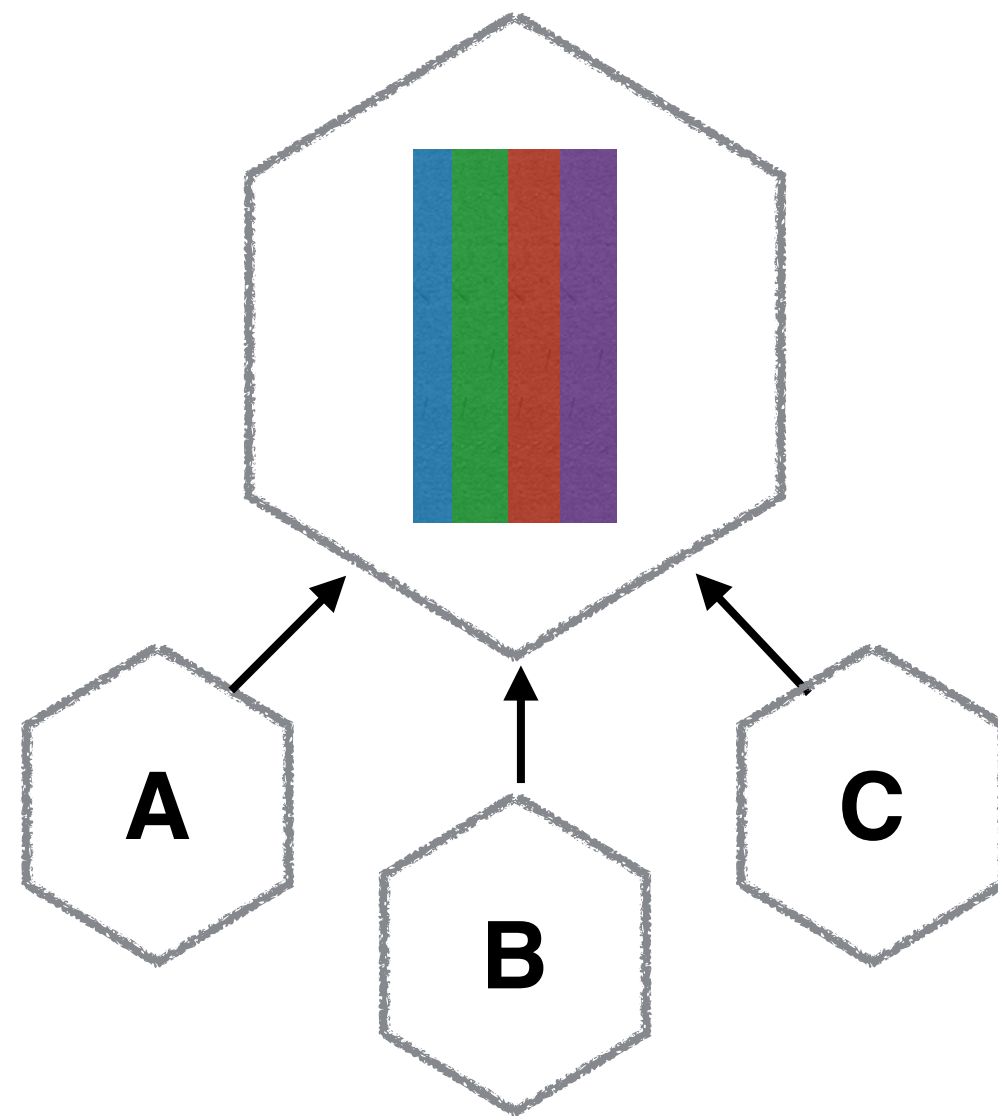
## Sidecar



Istio

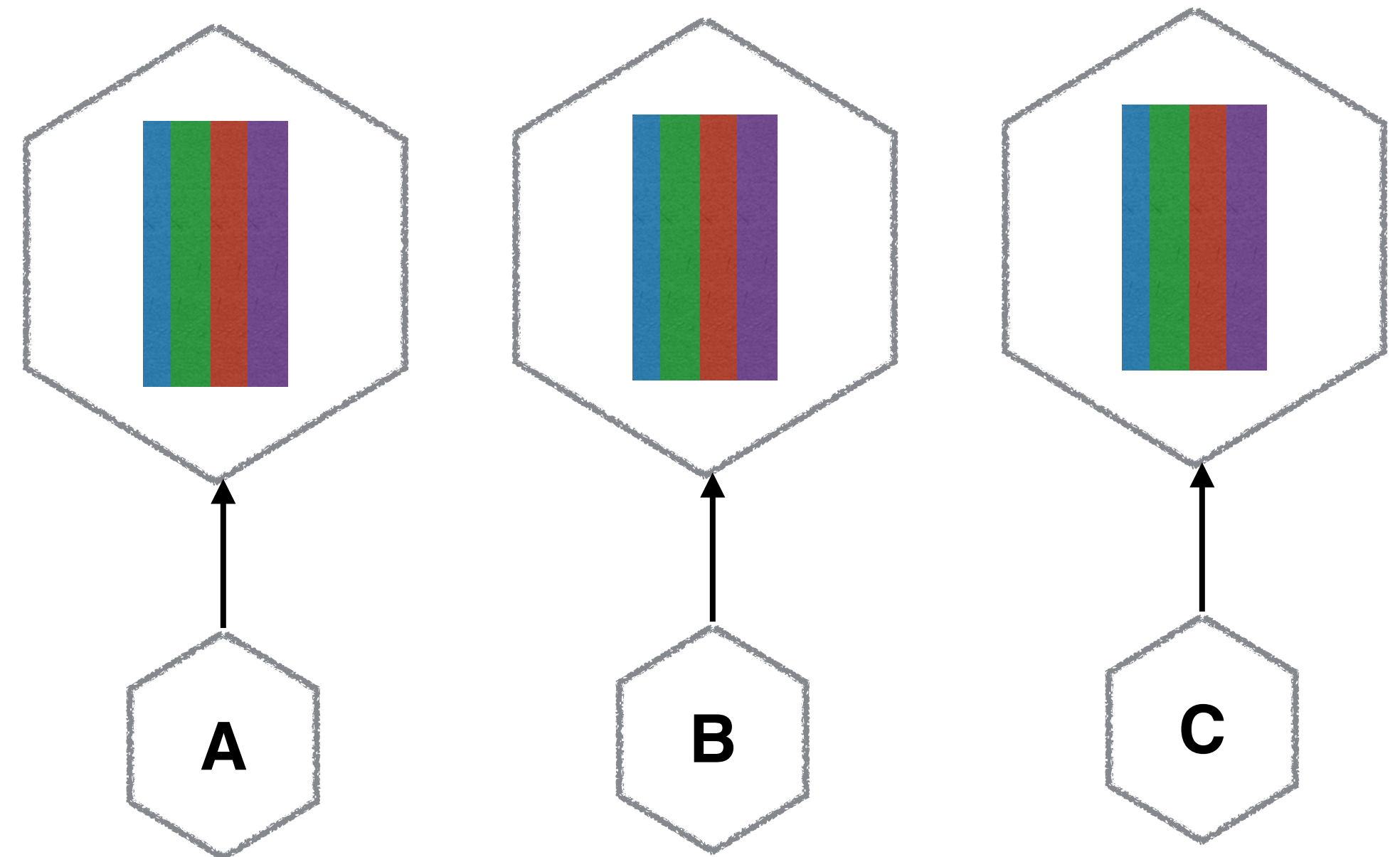
# SIDECARS VS PROXIES

## Local Proxy



Linkerd

## Sidecar



Istio



# SERVICE MESH CAPABILITIES

# SERVICE MESH CAPABILITIES

Load balancing

# SERVICE MESH CAPABILITIES

Load balancing

Traffic Routing (blue/green deploys, canaries)



# SERVICE MESH CAPABILITIES

Load balancing

Traffic Routing (blue/green deploys, canaries)

Service discovery

# SERVICE MESH CAPABILITIES

Load balancing

Traffic Routing (blue/green deploys, canaries)

Service discovery

Tracing

# SERVICE MESH CAPABILITIES

Load balancing

Traffic Routing (blue/green deploys, canaries)

Service discovery

Tracing

Security!

# MUTUAL TLS

## Mutual TLS Authentication

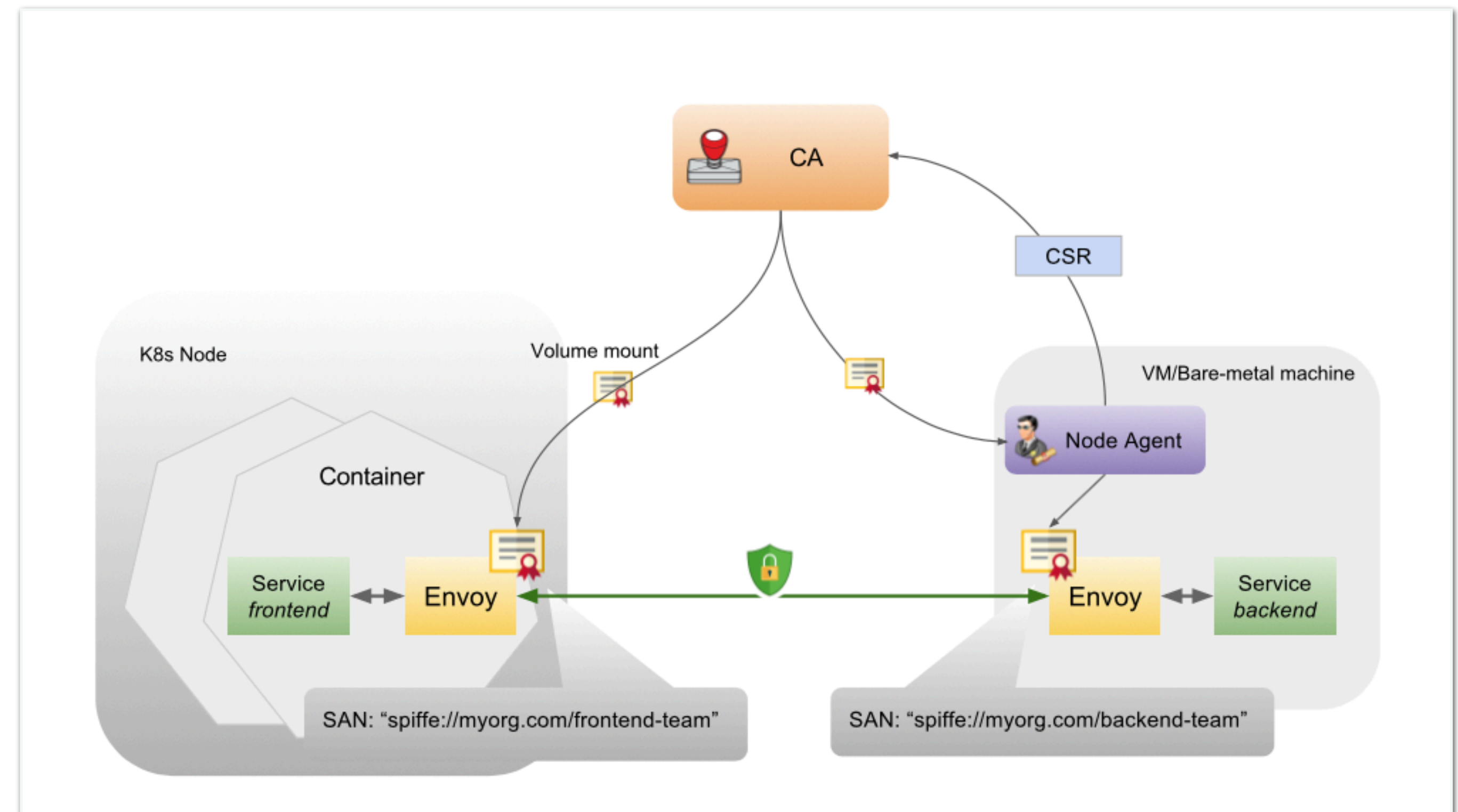
### Overview

Istio Auth's aim is to enhance the security of microservices and their communication without requiring service code changes. It is responsible for:

- Providing each service with a strong identity that represents its role to enable interoperability across clusters and clouds
- Securing service to service communication and end-user to service communication
- Providing a key management system to automate key and certificate generation, distribution, rotation, and revocation

### Architecture [🔗](https://istio.io/docs/concepts/security/mutual-tls.html)

The diagram below shows Istio Auth's architecture, which includes three primary components: identity, key management, and communication security. This diagram describes how Istio Auth is used to secure the service-to-service communication between service 'frontend' running as the service account 'frontend-team' and service 'backend' running as the service account 'backend-team'. Istio supports services running on both Kubernetes containers and VM/bare-metal machines.



<https://istio.io/docs/concepts/security/mutual-tls.html>

# Caution warranted?

# SUMMARY



## SUMMARY

# Patching & Passwords

## SUMMARY

**Patching & Passwords**

**Storing Secrets**

# SUMMARY

**Patching & Passwords**

**Storing Secrets**

**Transport Security**

# SUMMARY

**Patching & Passwords**

**Storing Secrets**

**Transport Security**

**Authorisation**

# SUMMARY

**Patching & Passwords**


**Storing Secrets**

**Transport Security**

**Authorisation**

**Service Meshes**

## MORE DETAILS...

HomeAboutOfferingsEv


# Securing Microservices: Protect Sensitive Data in Transit and at Rest.

3 Hour Online Video

*A 3 hour course looking at how to secure your microservice architecture*

Watch At Safari Books Online

Microservice architectures offer a lot of advantages: making it easier to scale your application and team, use different technology, and ship features more quickly. However, they also create serious challenges, such as how to properly ensure that the systems you create are secure. If done well, microservice architectures can be significantly more secure than other types of software, but if approached in a naive fashion, you can end

HomeAboutOfferings

# Serverless Fundamentals For Microservices.

3 Hour Online Video

*A 3 hour course introducing Serverless, and looking at how it relates to  
Microservice architectures*


Watch At Safari Books Online

Serverless technology offers an attractive proposition: it frees us from much of the administration work we've worried about in the past, giving us more time to focus on building great software. But there's a lot of hype around the technology too. In this video series, microservices expert Sam Newman explains what serverless

<http://samnewman.io/offerings/videos>



THANKS!

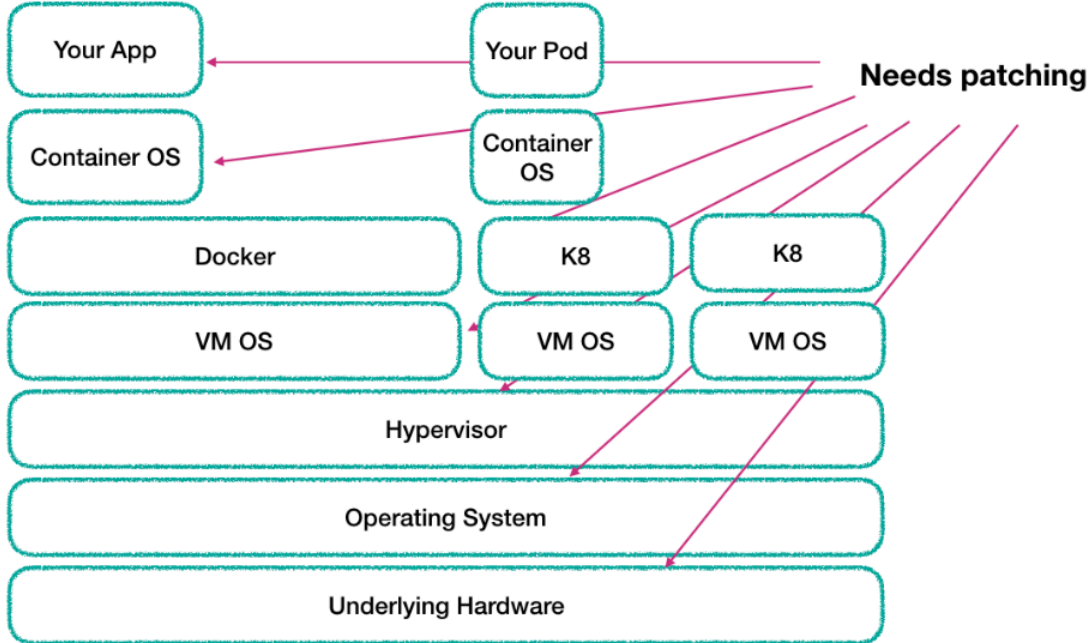


HomeAboutOfferingsEventsWritingContact

## Insecure Transit - Microservice Security.

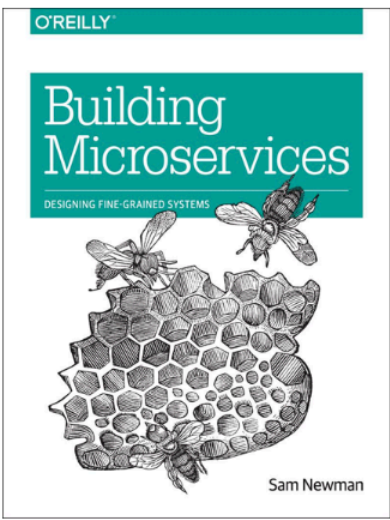
60min Presentation

**PATCHING MADNESS!**




*A deep dive into some of the technical challenges and solutions to securing a microservice architecture.*

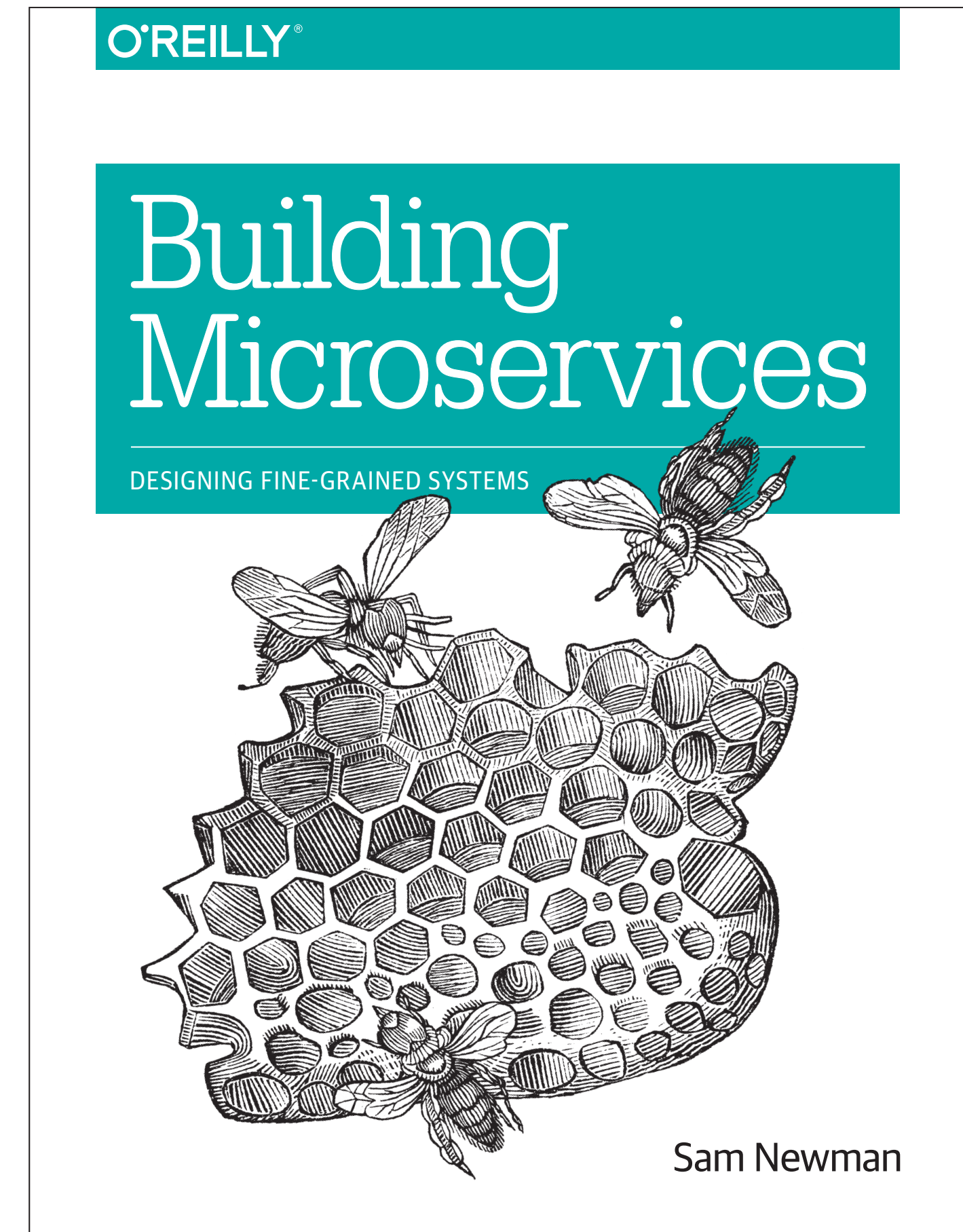
### Book!



I have written a book called "Building Microservices", which is available now. Want to know more? → [Read on...](#)

### Video!





<http://samnewman.io/>

@samnewman