

# What Tech Leaders Must Know About Microservices



- **Leslie Chapman**
- **Distinguished Engineer - Comcast**
- **I LOVE CATS**



- **Michael Winslow**
- **Director - Comcast**
- **I still code for management tasks**

 @leslieachapman

 /lesliechapman

 @michaelwinslow

 /michaelwinslow



COMCAST  
TECHNOLOGY  
CENTER



NBC 10

TELEMUNDO 62





A global media and technology company with several businesses, including Comcast, NBCUniversal, and Sky.

## COMCAST

Products & Services

Cable Networks

Broadcast

Film

Parks

Products & Services

Channels & Content

Comcast Spectacor



@michaelwinslow | @leslieachapman



# Culture Automation Measurement Sharing

Credit: Damon Edwards, John Willis  
<https://itrevolution.com/devops-culture-part-1>



@michaelwinslow | @leslieachapman



# So, Microservices...



@michaelwinslow | @leslieachapman



# Some background:

1. Started as Principal Developer – Web Services
2. The team decided to pursue Microservices
3. The team grew (fast), I took a leadership role
4. Two unique perspectives ( Cool Tech / Business Value )
5. Things worked out well, **but it was not always sunshine**

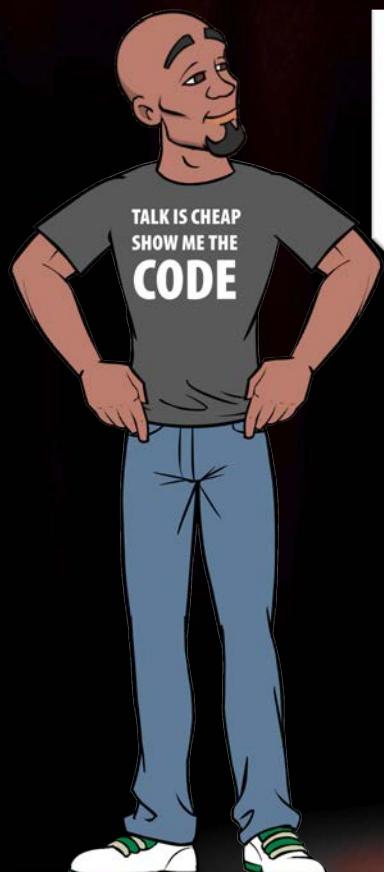
YOUR CHOICE



## Now you have a choice:

- Hear about the lessons your team learnt! (**red**)
- Hear about how great your team is at making microservices! (**blue**)





@michaelwinslow | @leslieachapman

COMCAST  
DevOps

# Ok seriously, Microservices...



@michaelwinslow | @leslieachapman



“A true, seasoned **professional** should be able to speak for five minutes, **intelligently**, about any subject.” – **Duane Winslow,**  
**Mentor / Uncle**



@michaelswinslow | @leslieachapman



# Monolith vs Microservice

**ASK YOUR TEAM**

*“Are we all on the same page as to what MONOLITHS and MICROSERVICES actually are?”*



@michaelwinslow | @leslieachapman

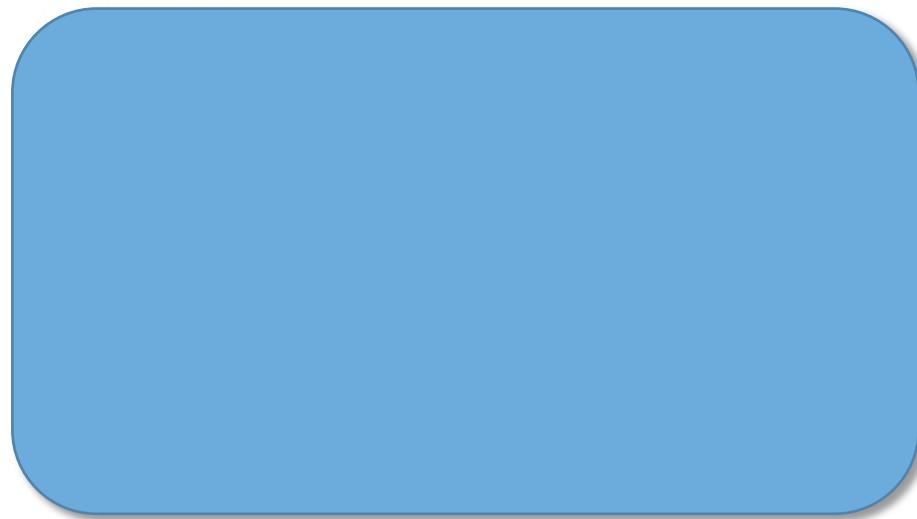
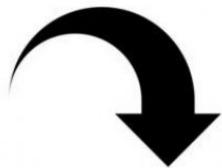




@michaelwinslow | @leslieachapman

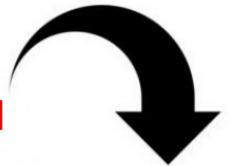


This is a **Service**

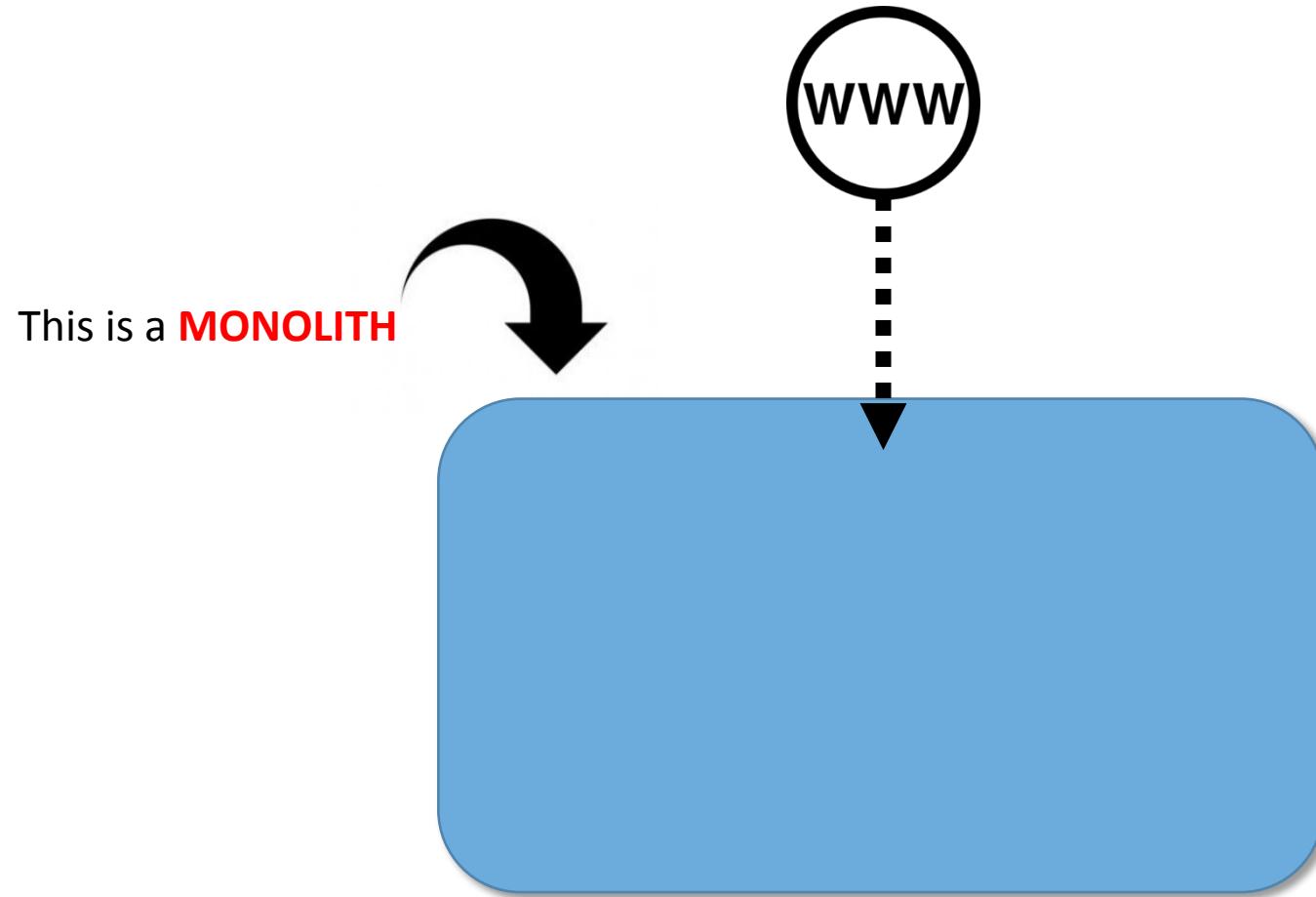


@michaelwinslow | @leslieachapman

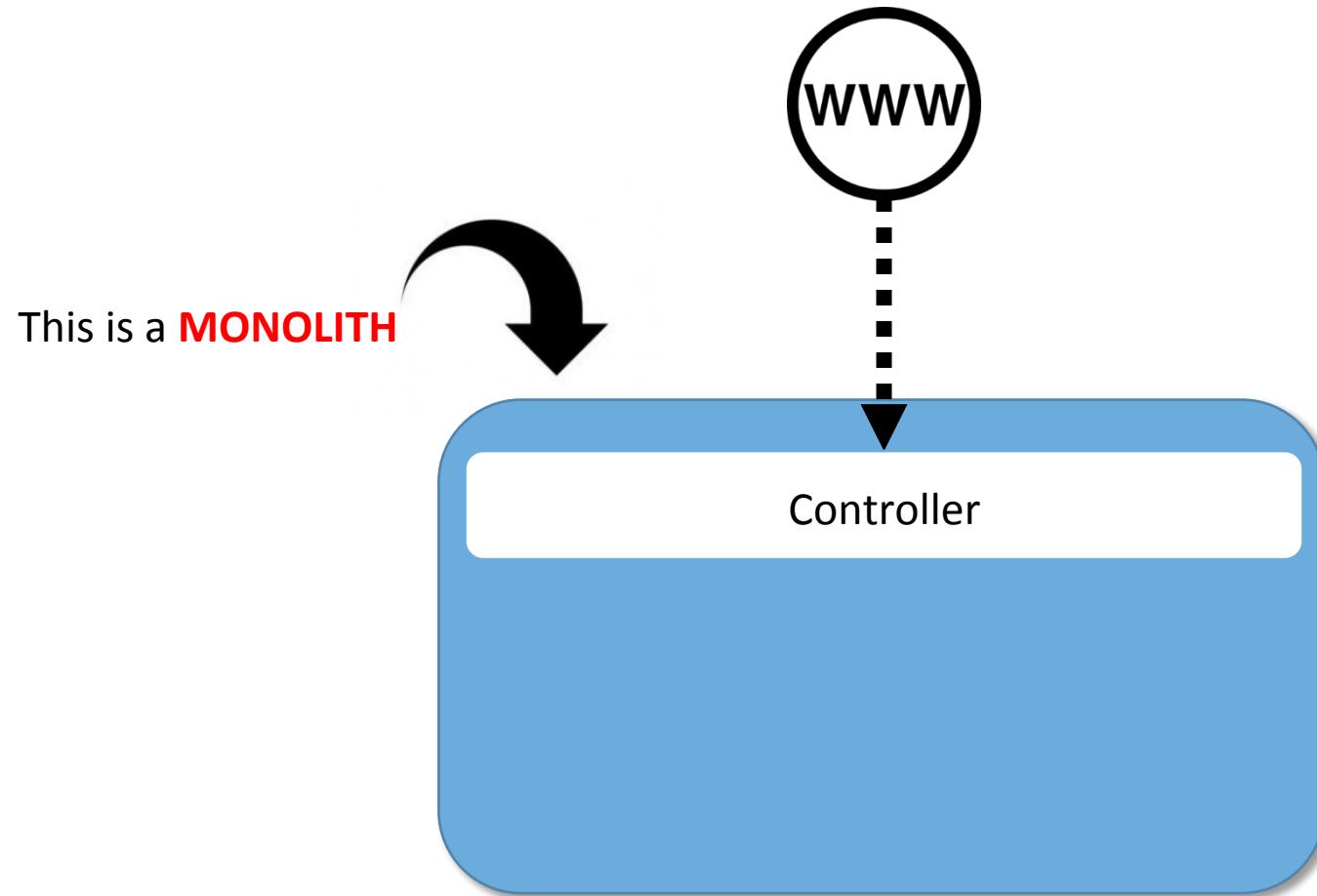
This is a **MONOLITH**



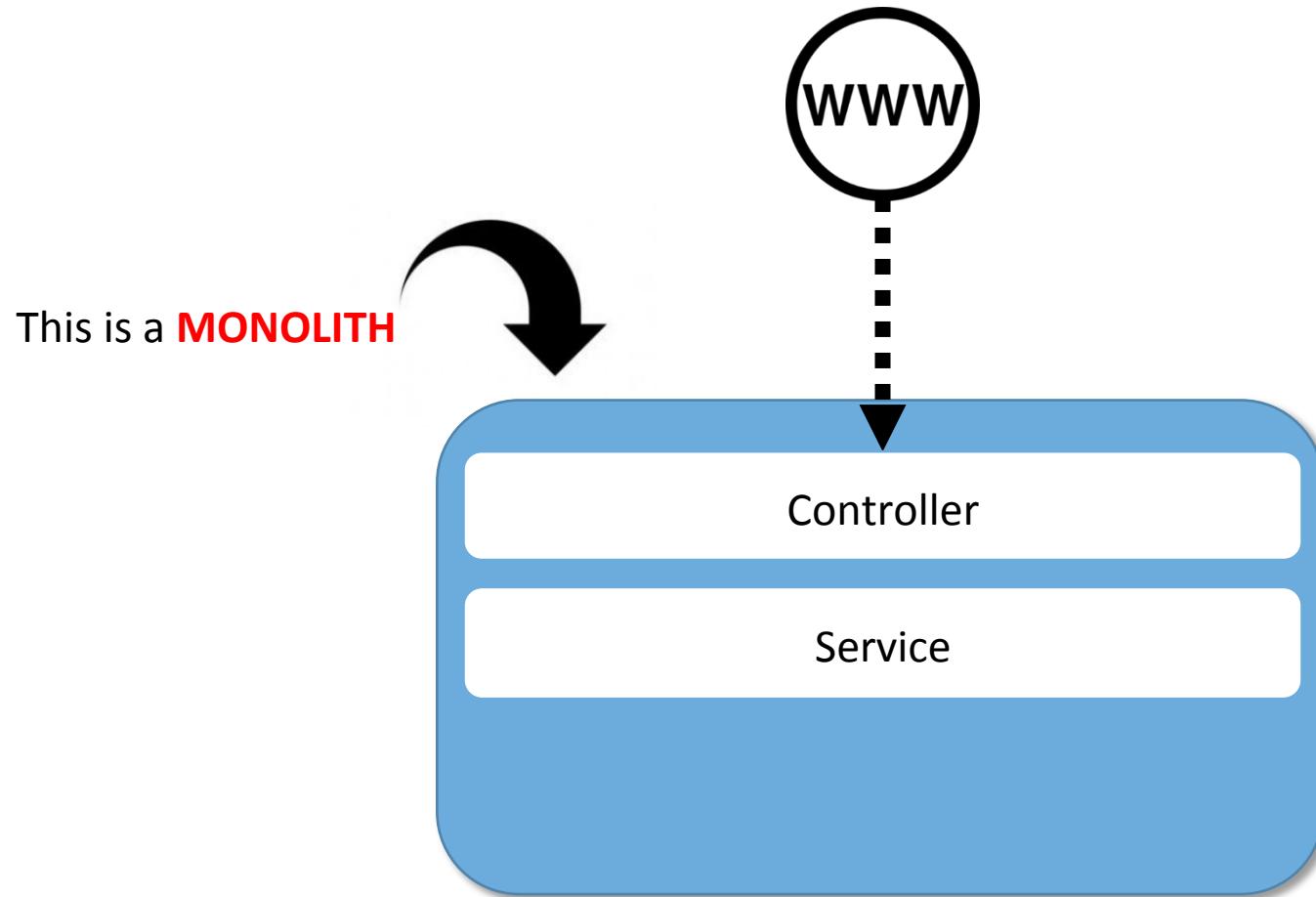
@michaelwinslow | @leslieachapman



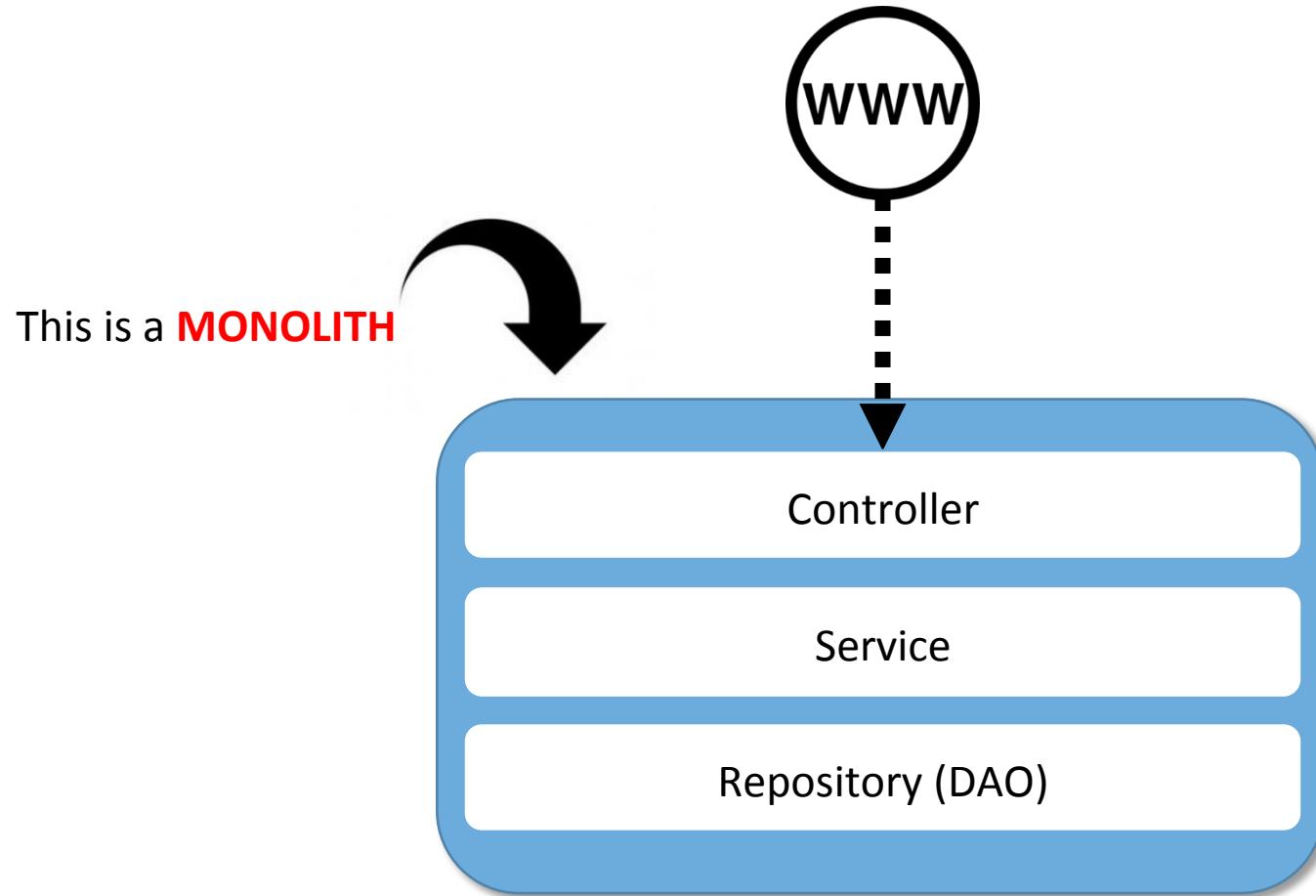
@michaelwinslow | @leslieachapman



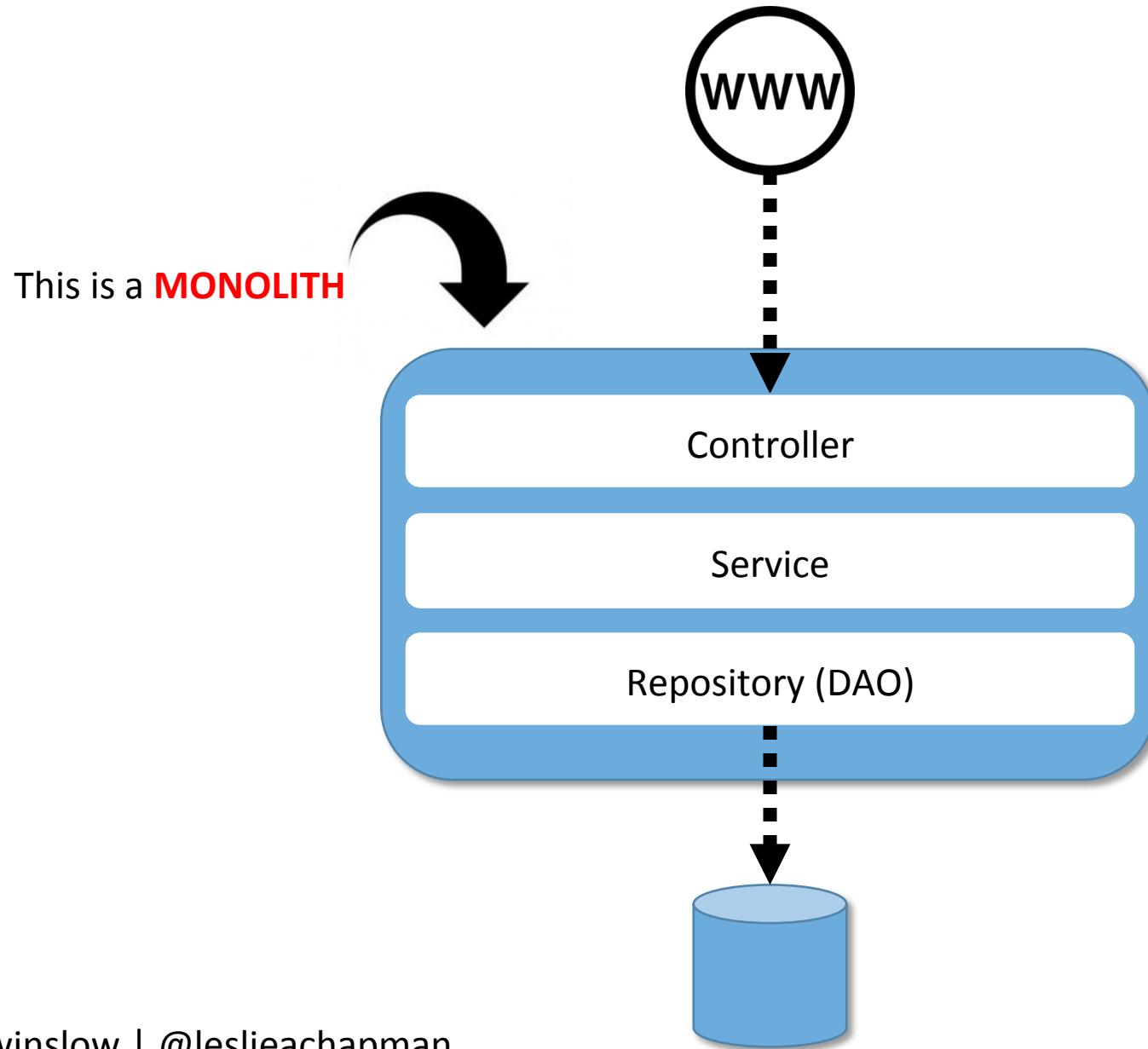
@michaelwinslow | @leslieachapman



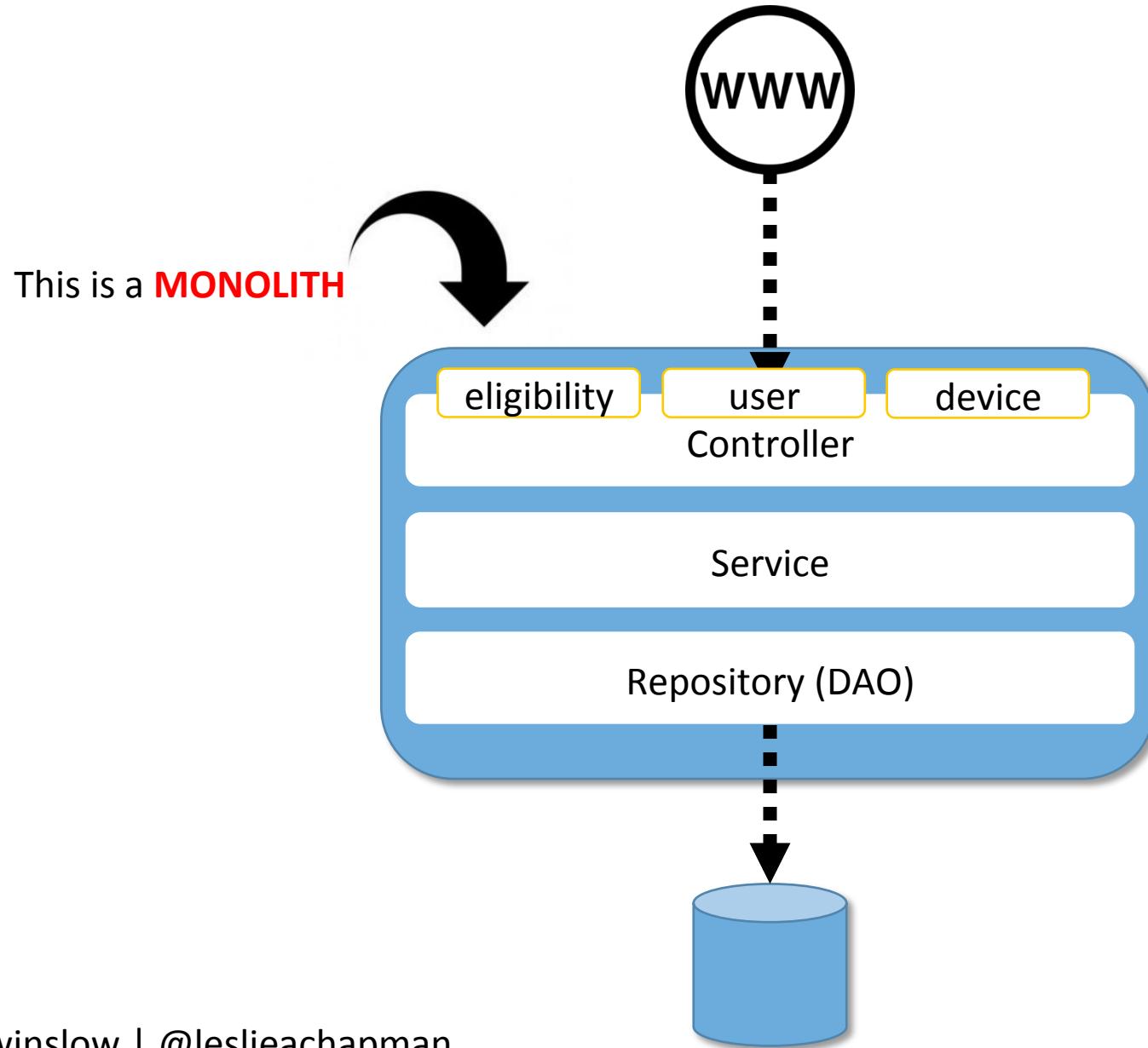
@michaelwinslow | @leslieachapman



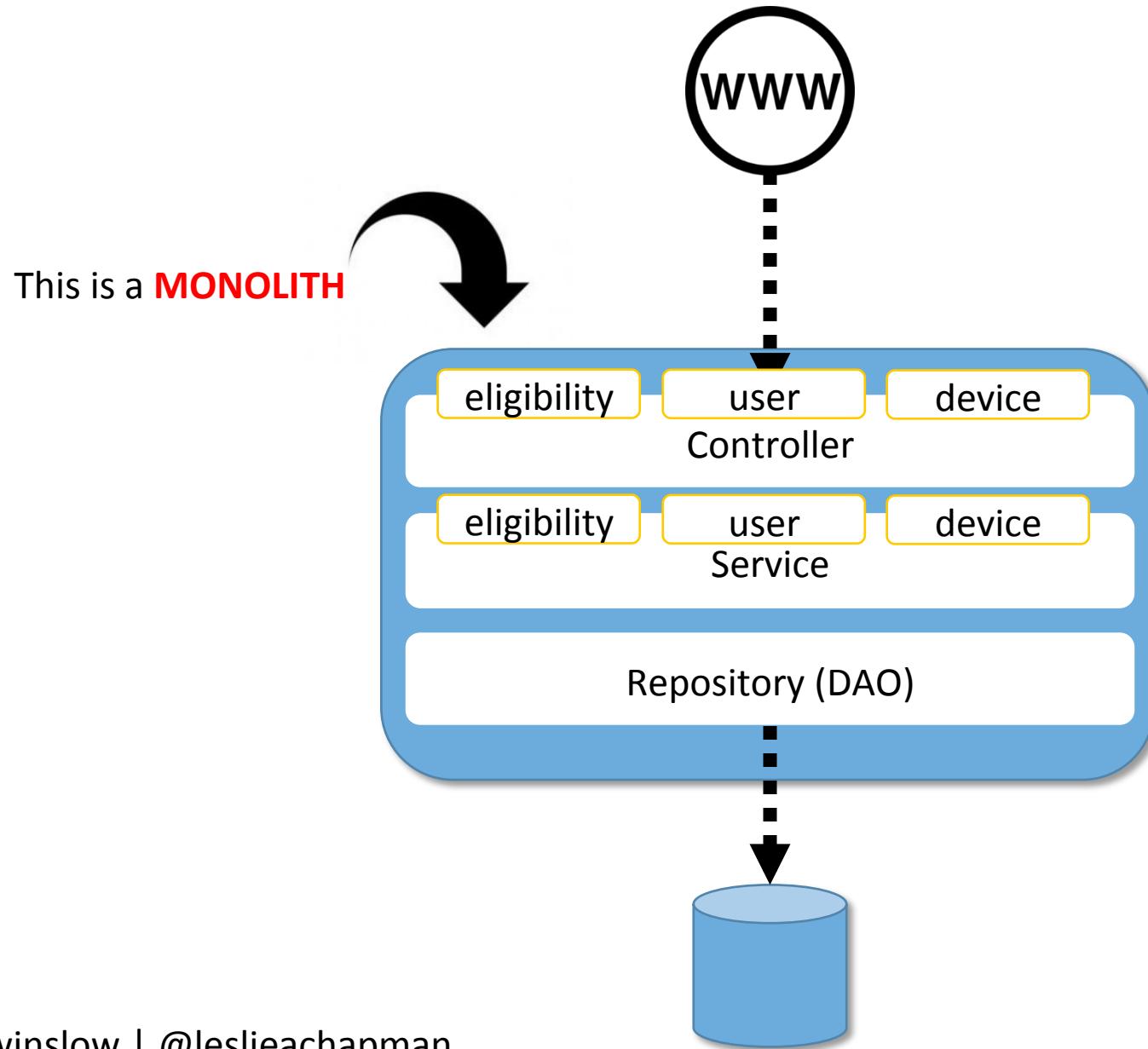
@michaelwinslow | @leslieachapman



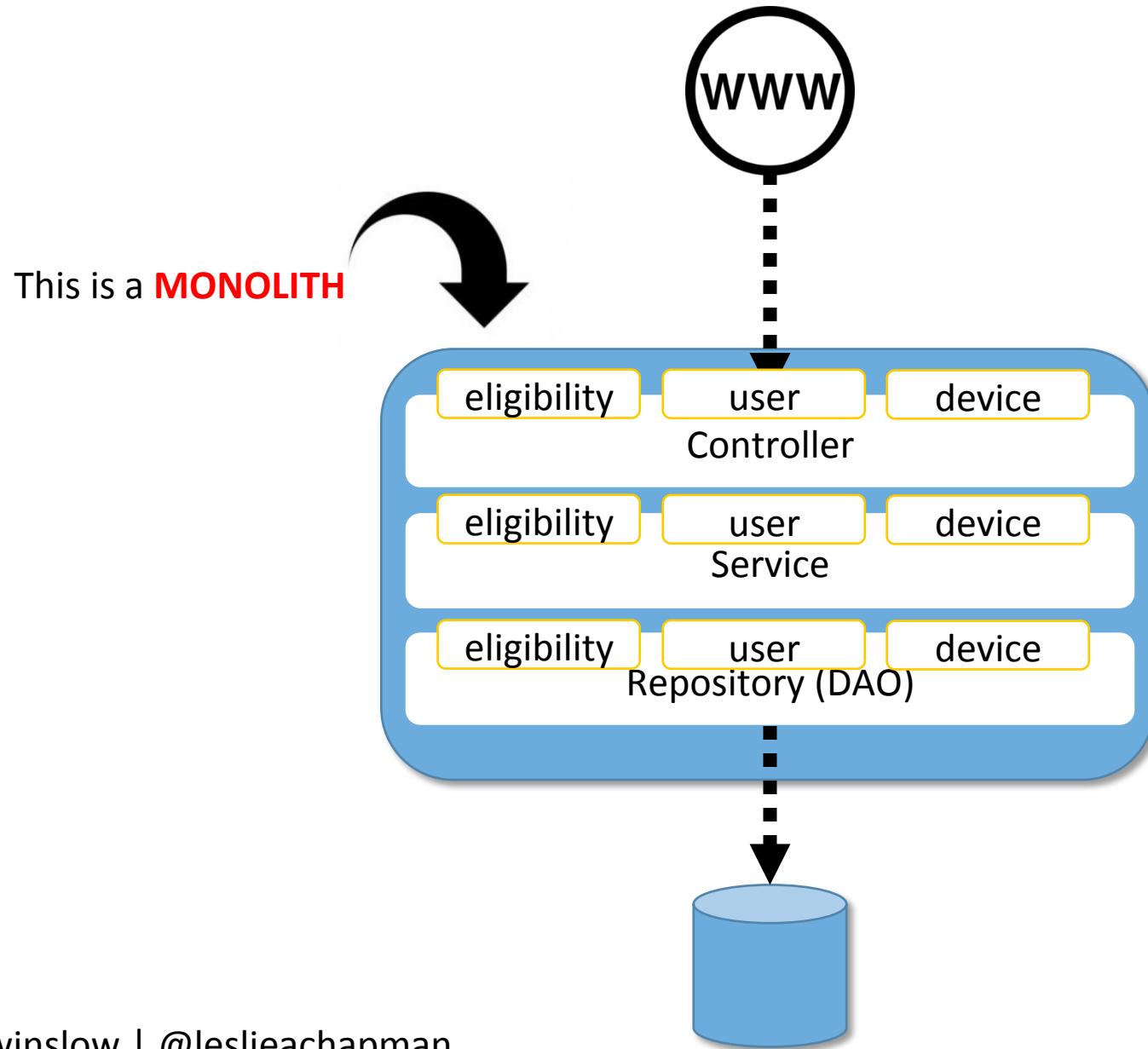
@michaelwinslow | @leslieachapman



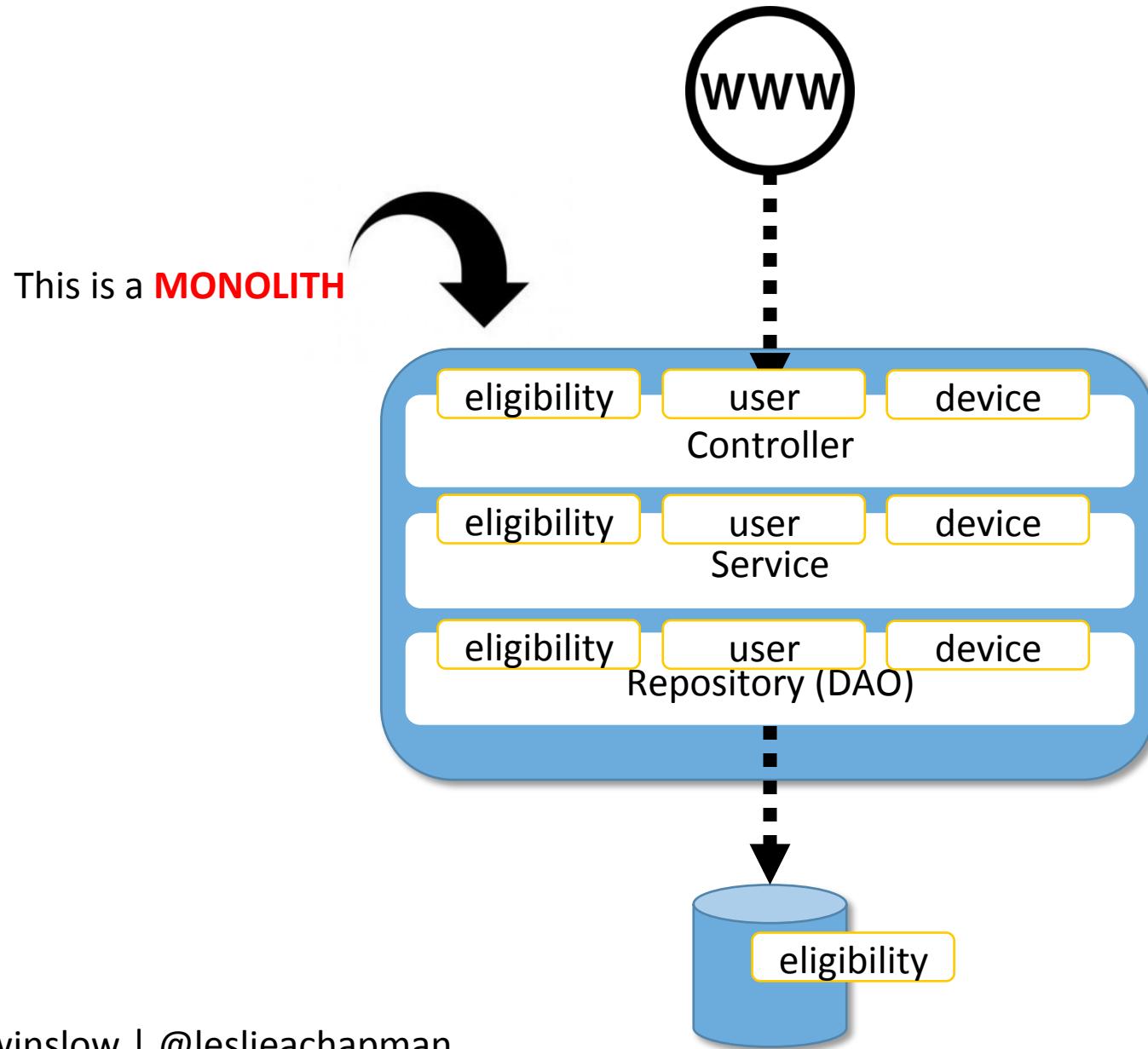
@michaelwinslow | @leslieachapman



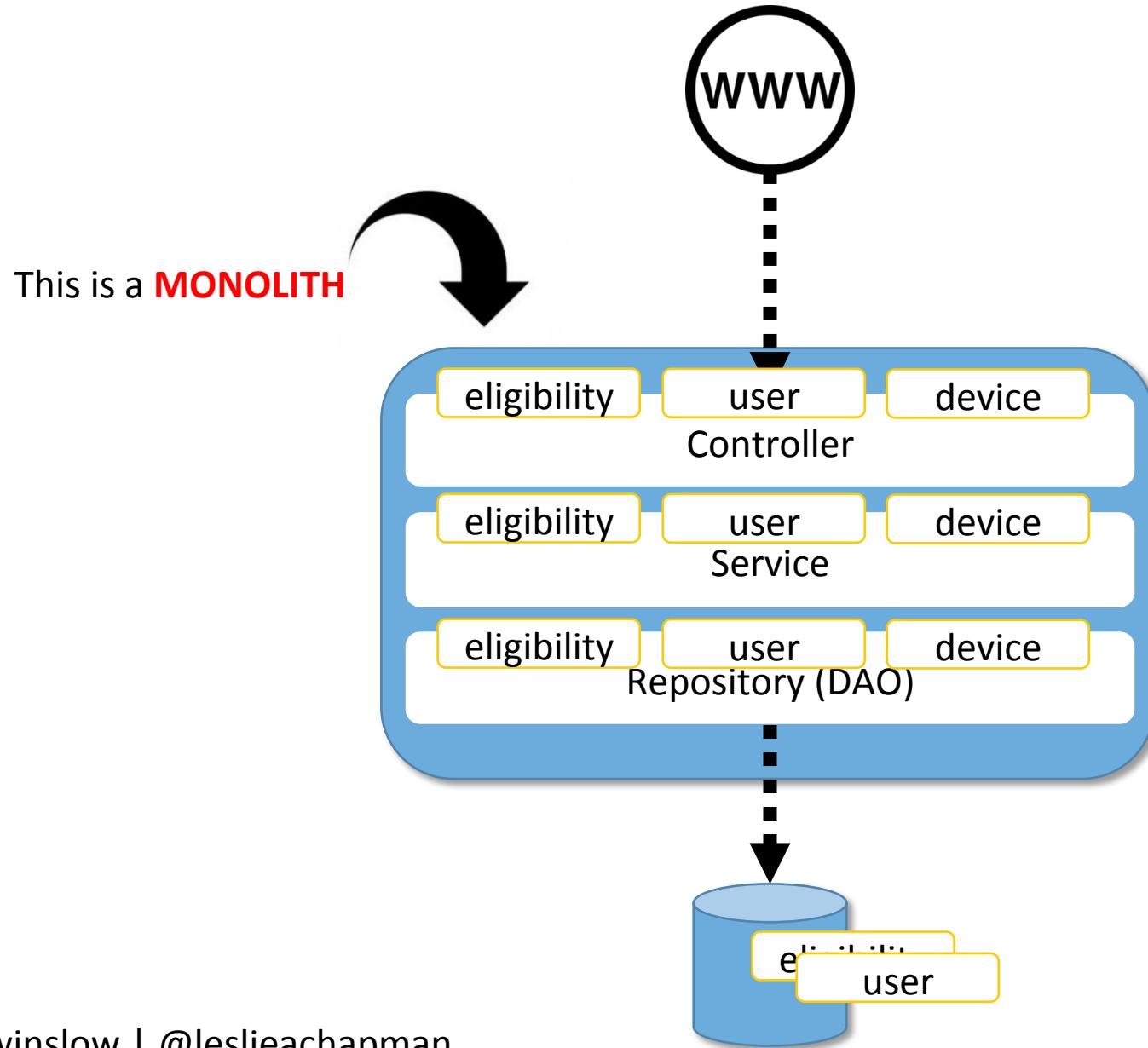
@michaelwinslow | @leslieachapman



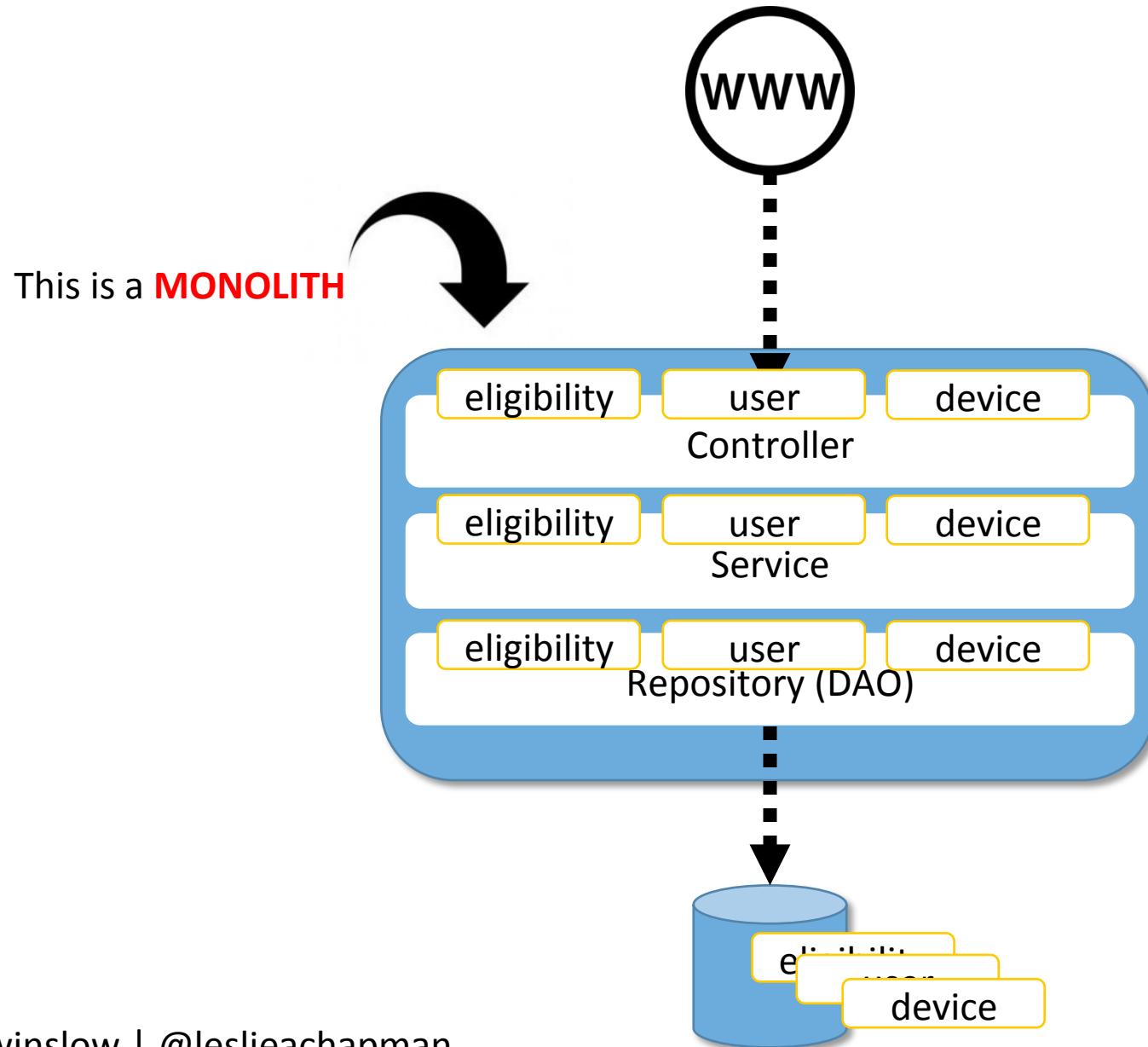
@michaelwinslow | @leslieachapman



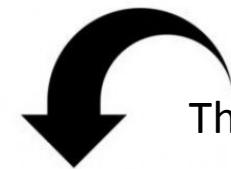
@michaelwinslow | @leslieachapman



@michaelwinslow | @leslieachapman



@michaelwinslow | @leslieachapman

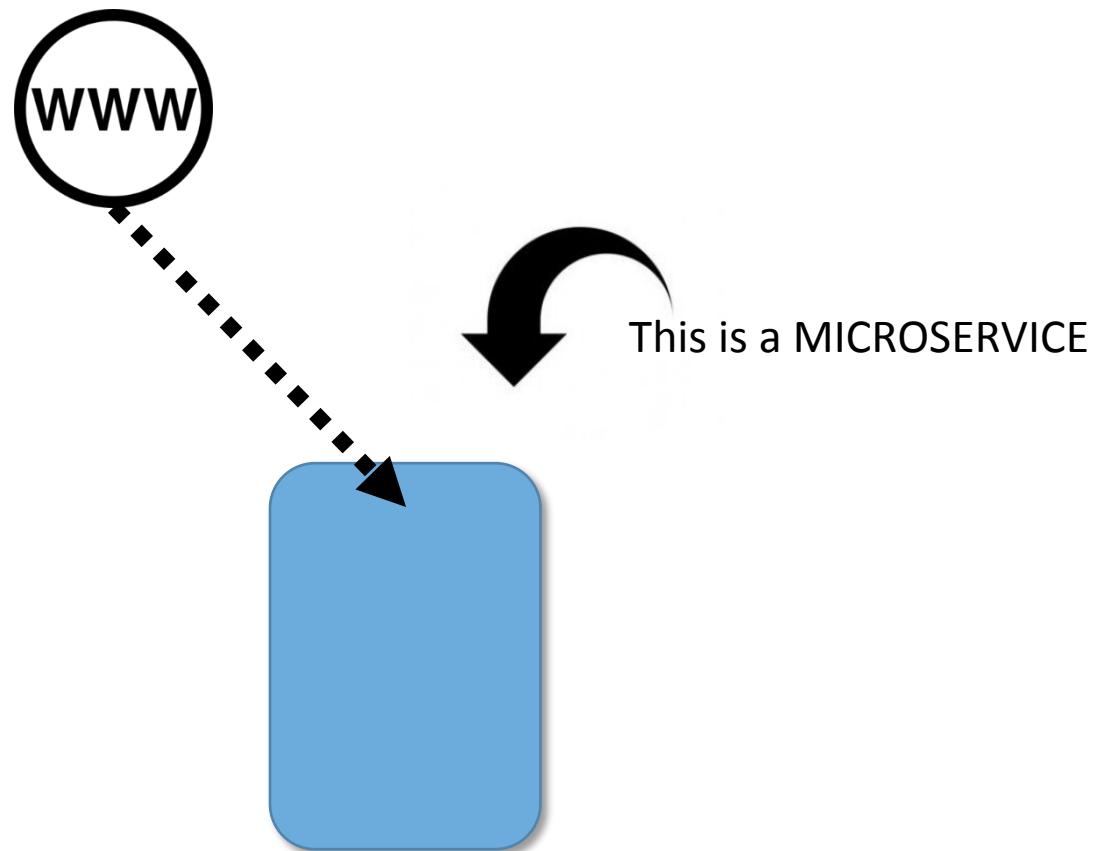


This is a MICROSERVICE

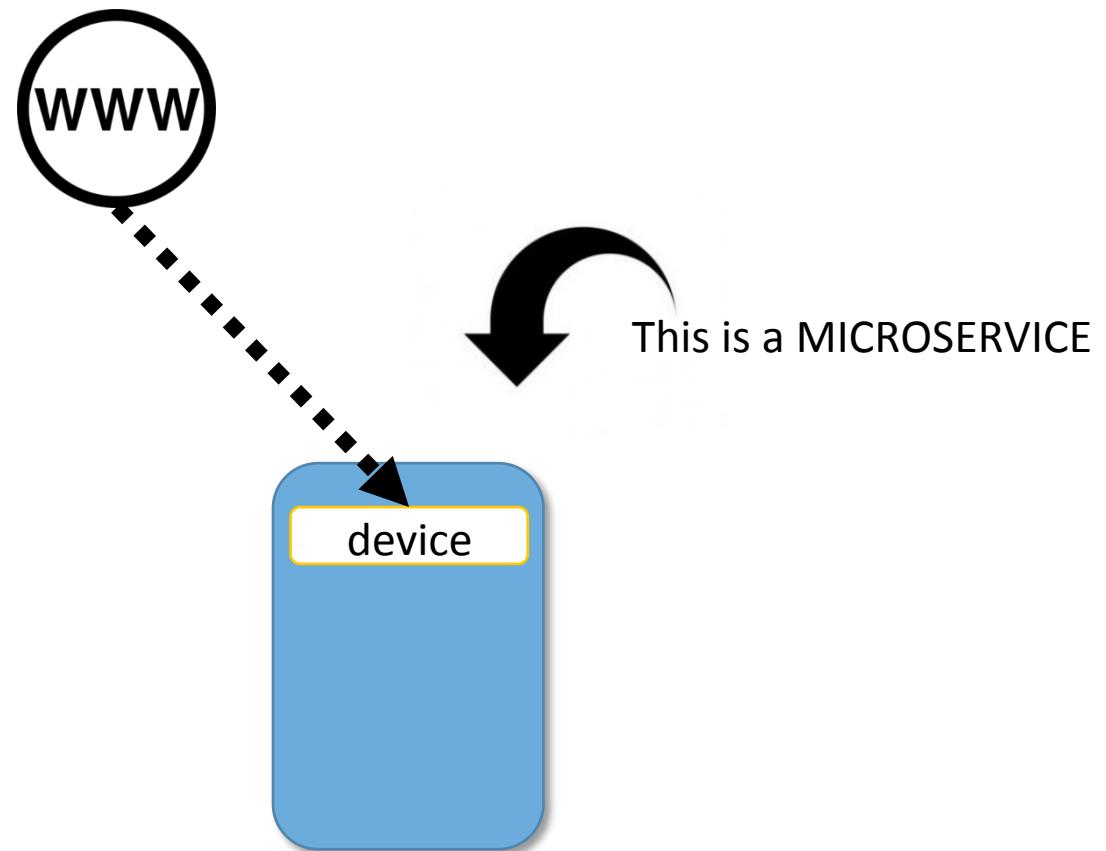


@michaelwinslow | @leslieachapman

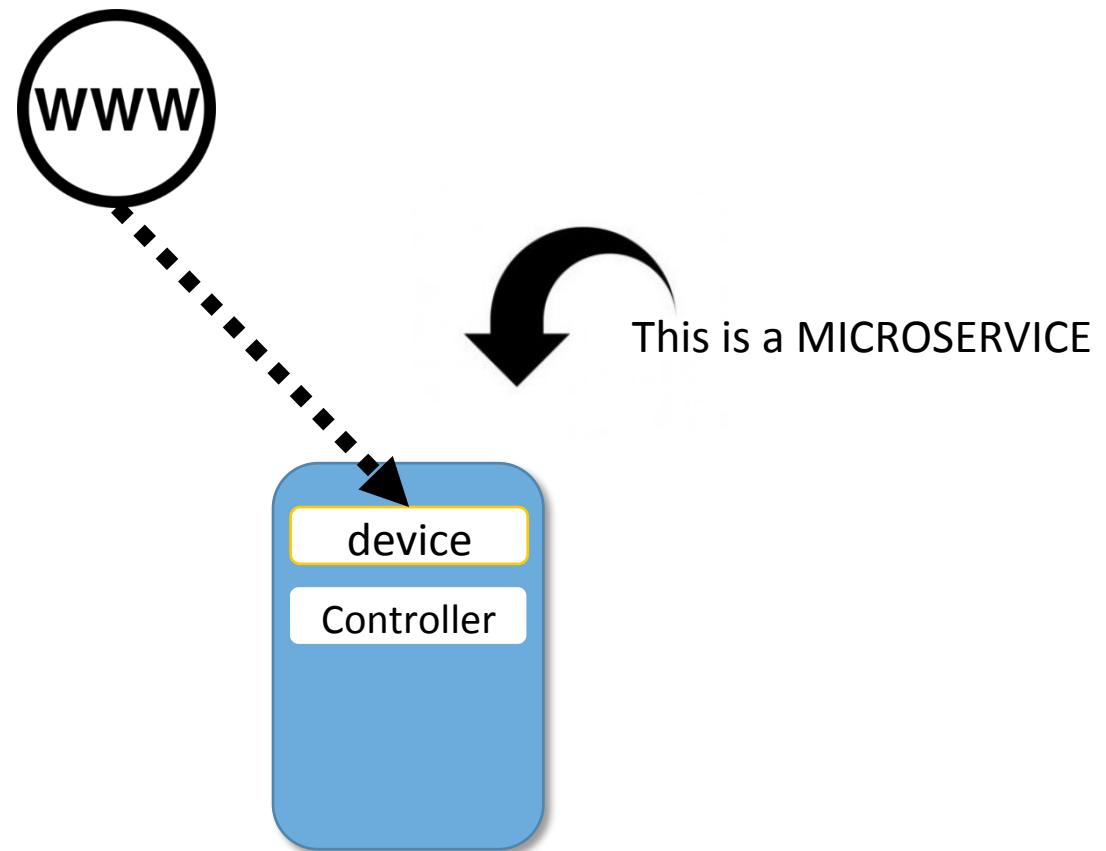




@michaelwinslow | @leslieachapman

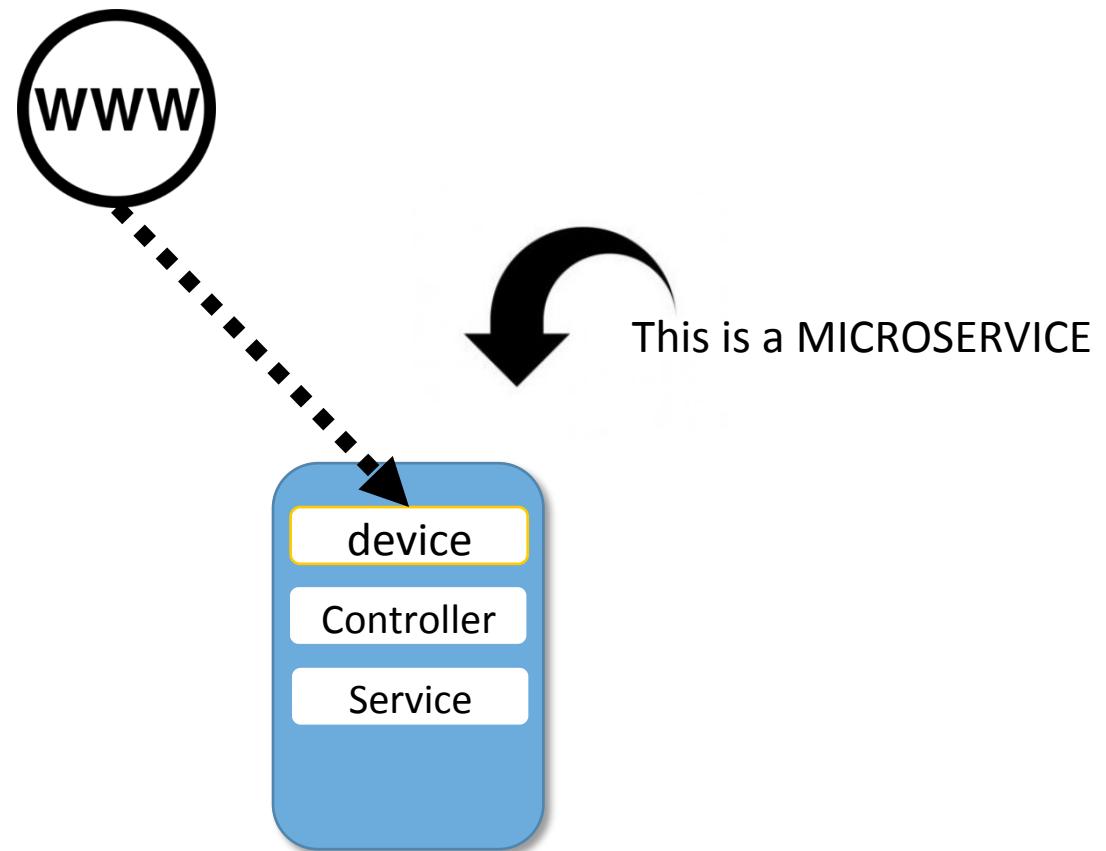


@michaelwinslow | @leslieachapman



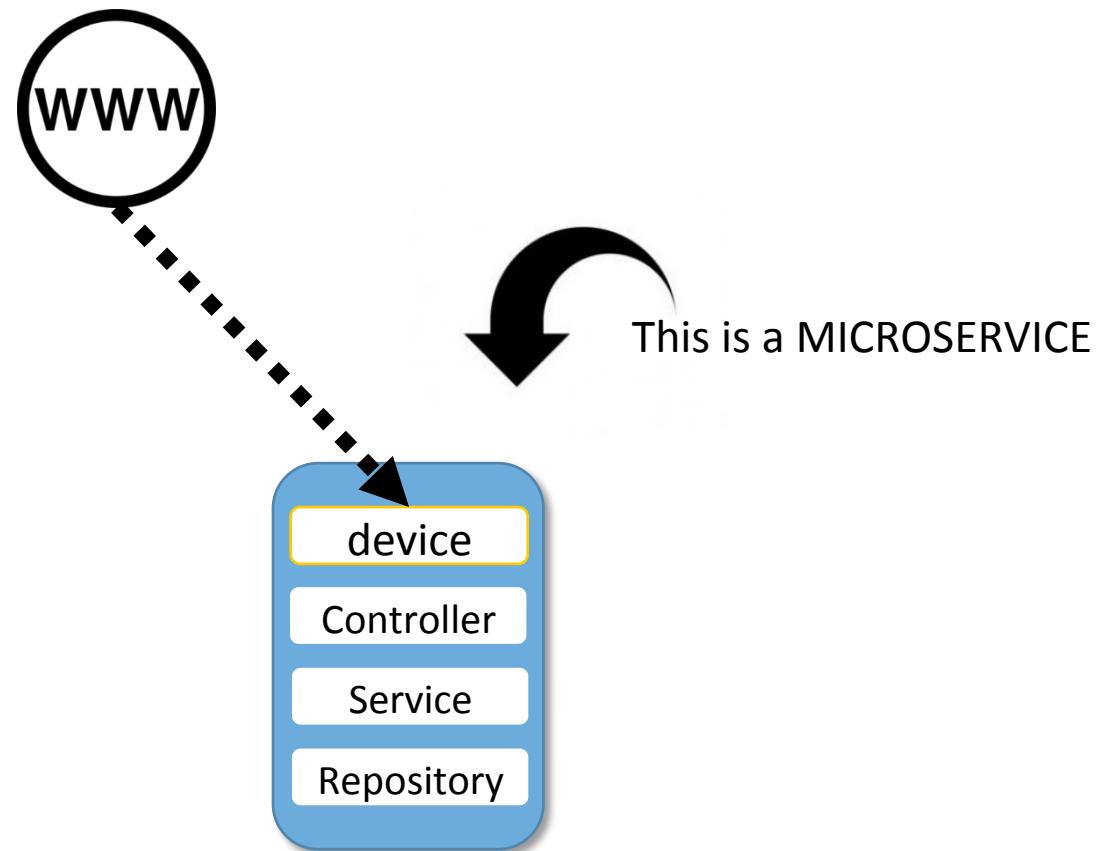
@michaelwinslow | @leslieachapman





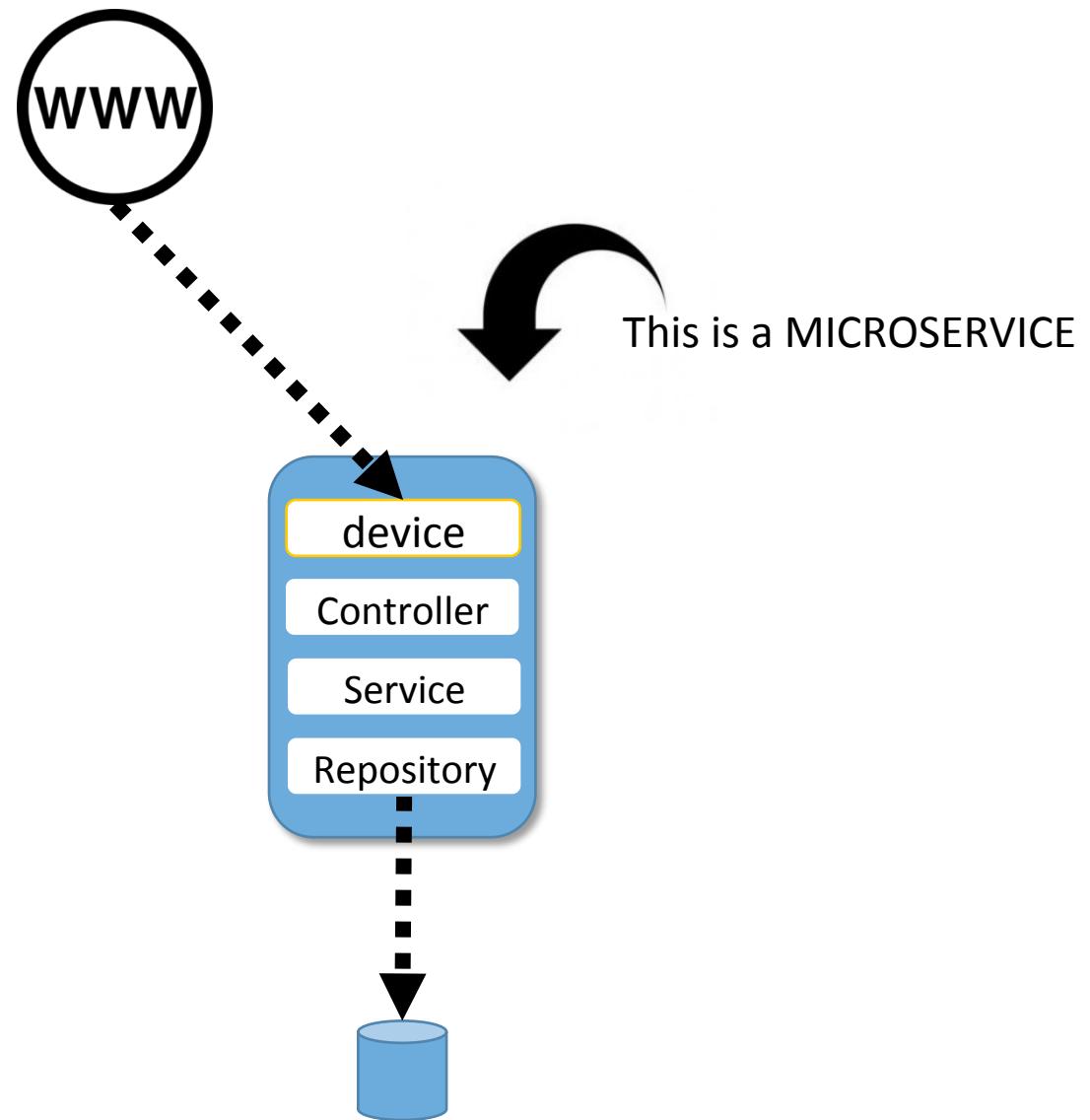
@michaelwinslow | @leslieachapman



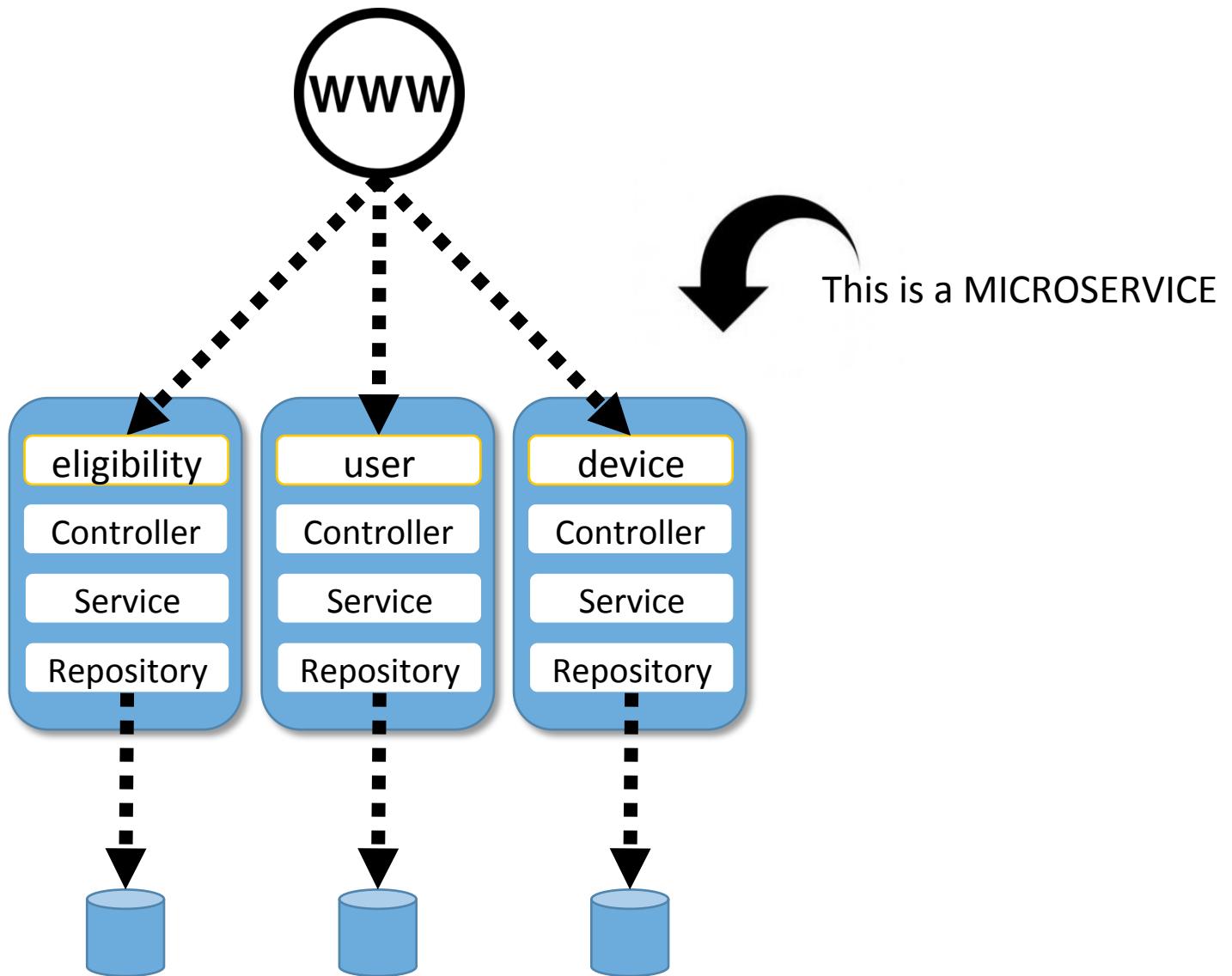


@michaelwinslow | @leslieachapman





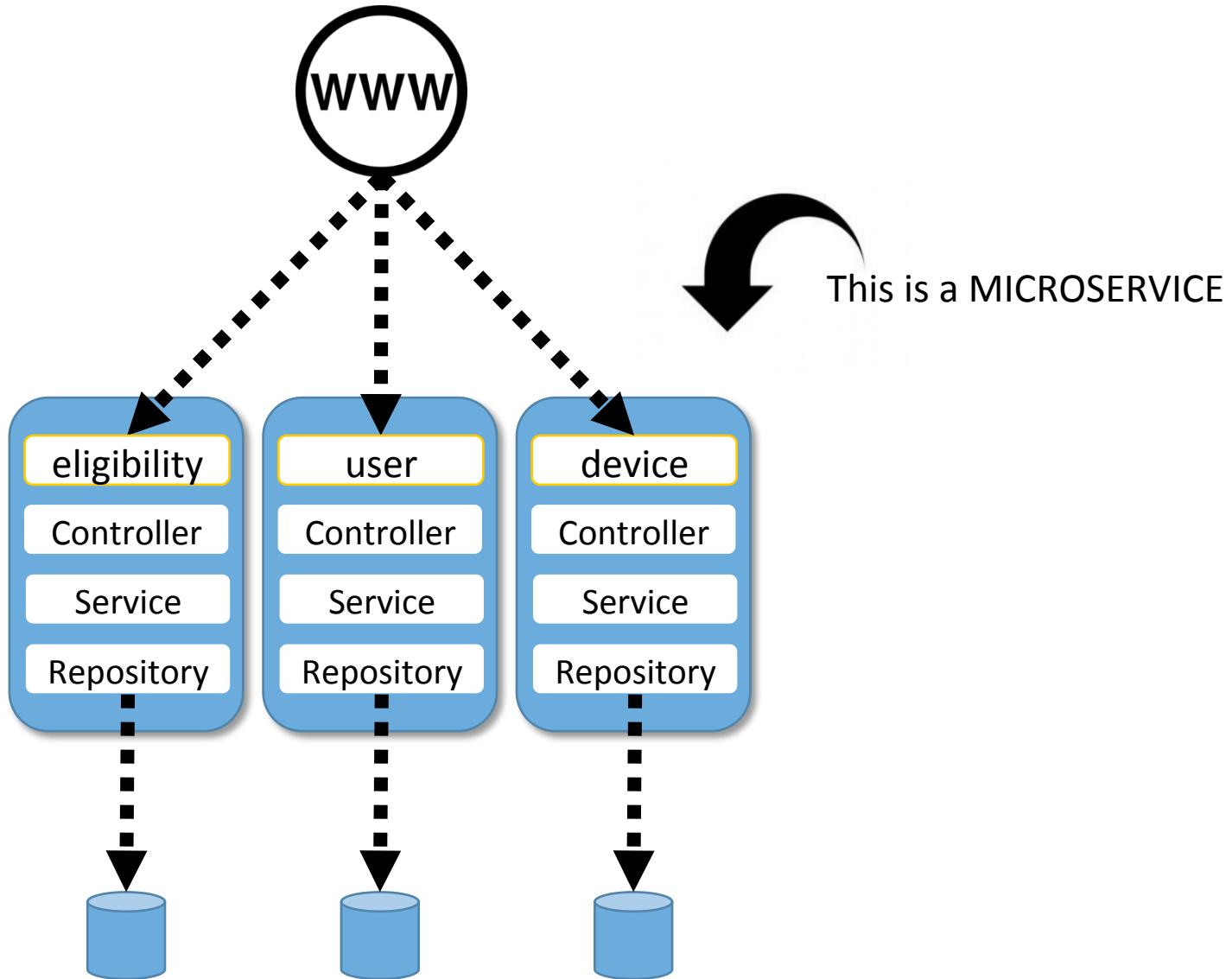
@michaelwinslow | @leslieachapman

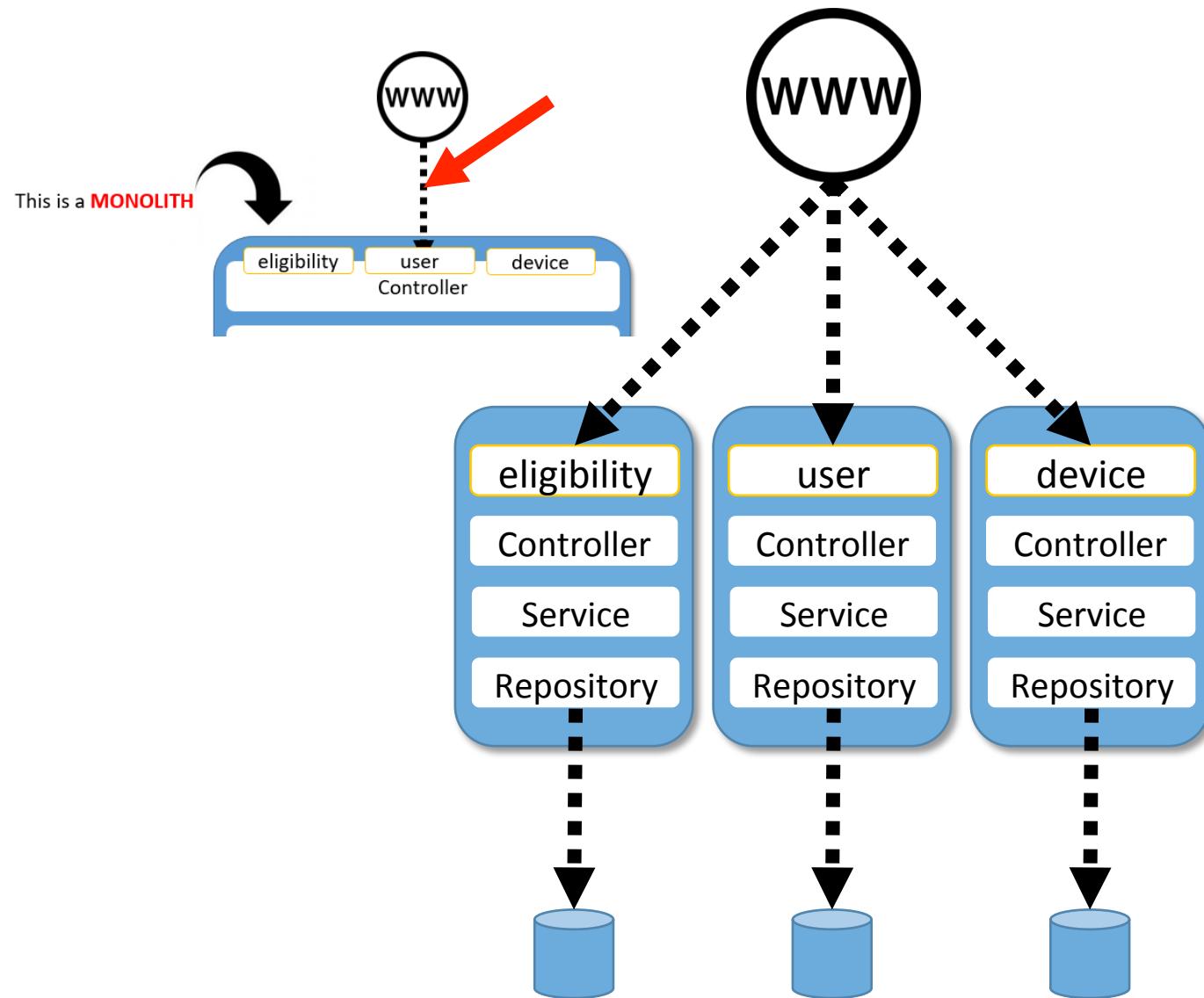


@michaelwinslow | @leslieachapman

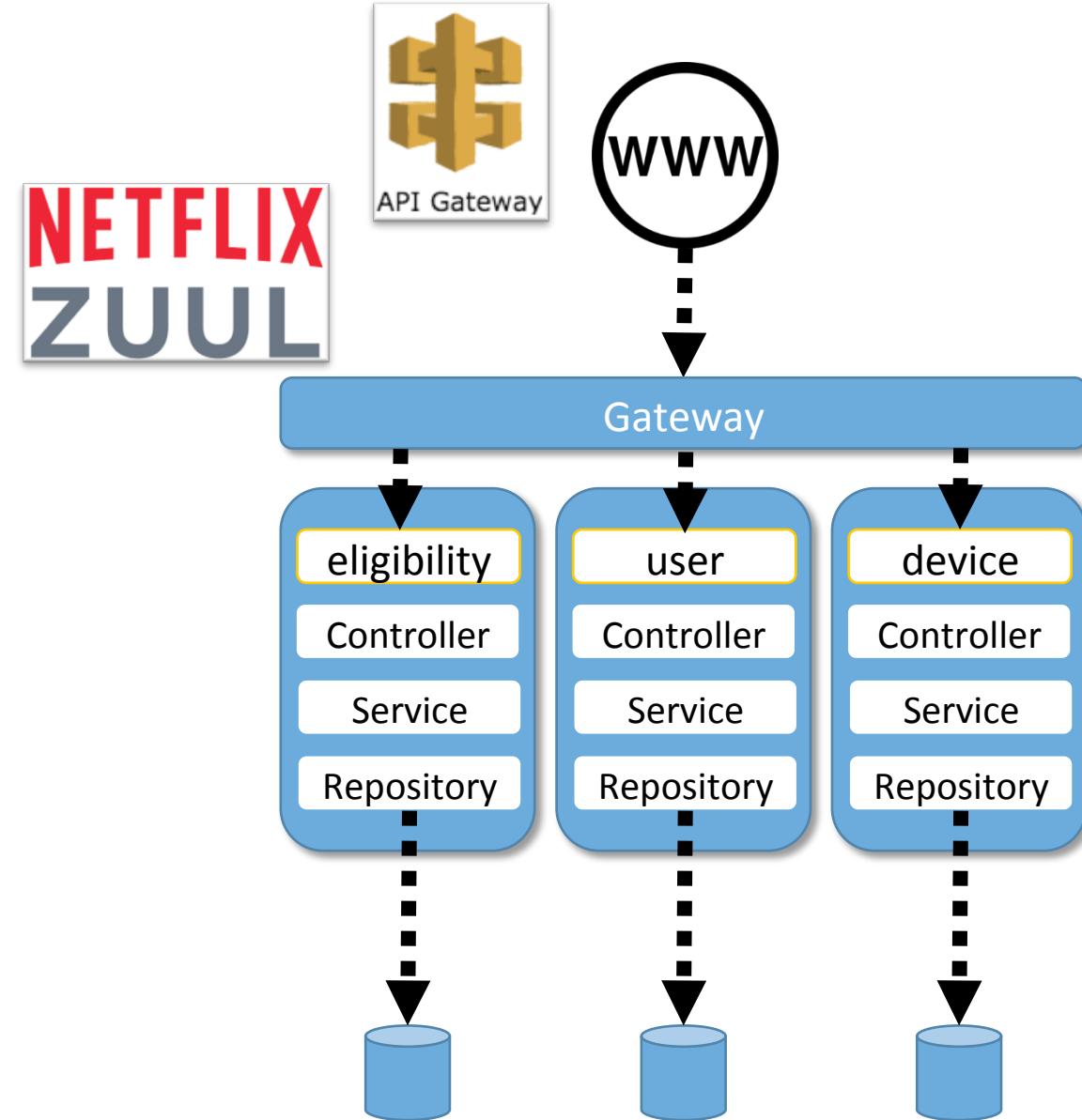
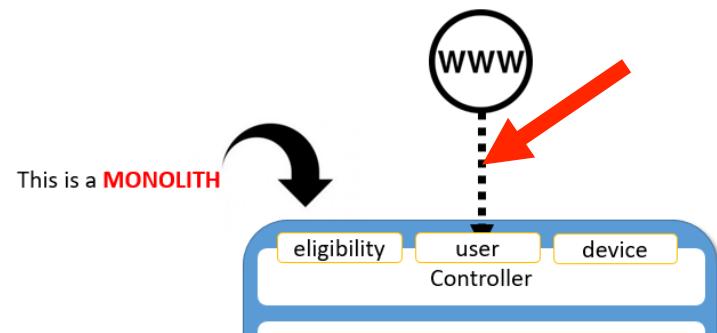
**Note:**

Large organizations may dedicate a small team to each microservice. A team that has a great deal of developers who can work independently may benefit from this model.

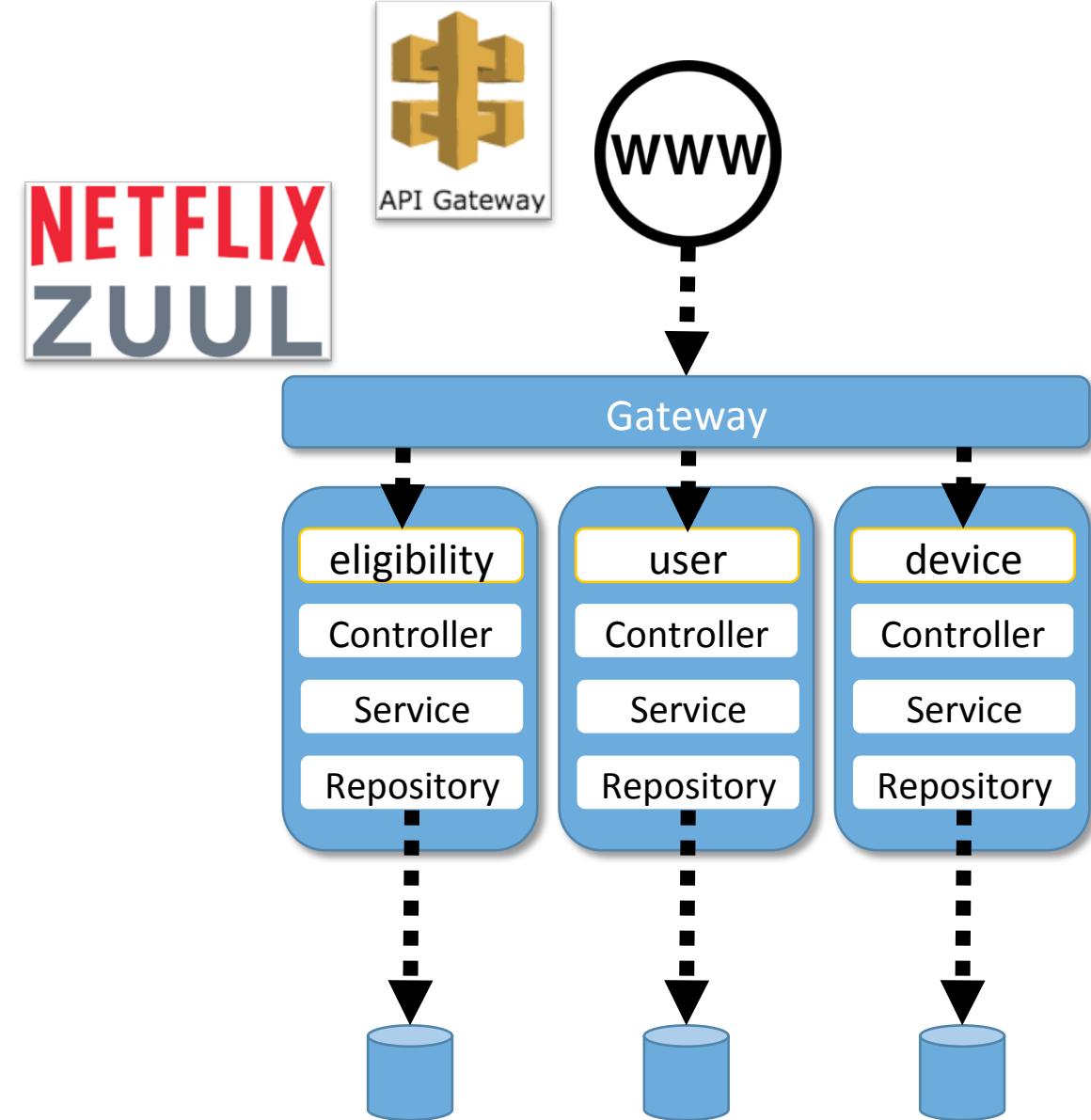




@michaelwinslow | @leslieachapman



@michaelwinslow | @leslieachapman



@michaelwinslow | @leslieachapman

# Scale / Service Discovery

**ASK YOUR TEAM**

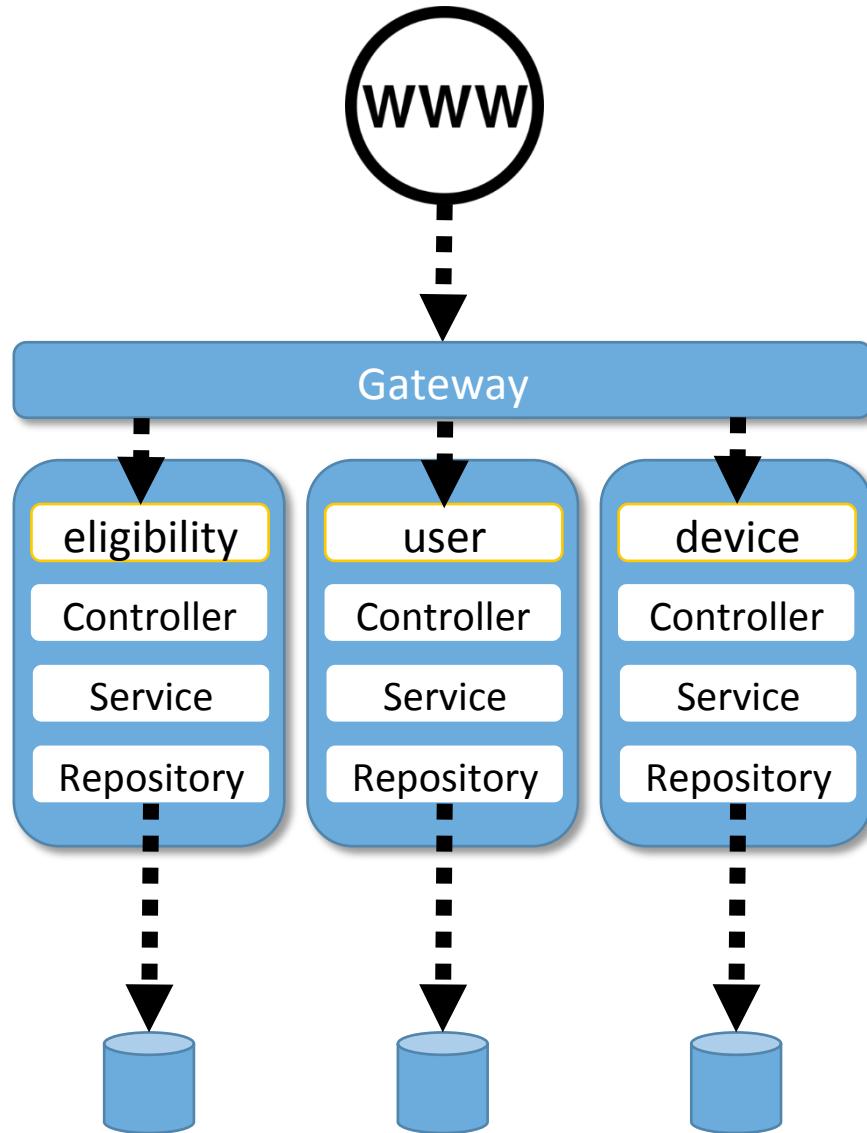
*“How are we handling service discovery and/or auto-scaling?”*

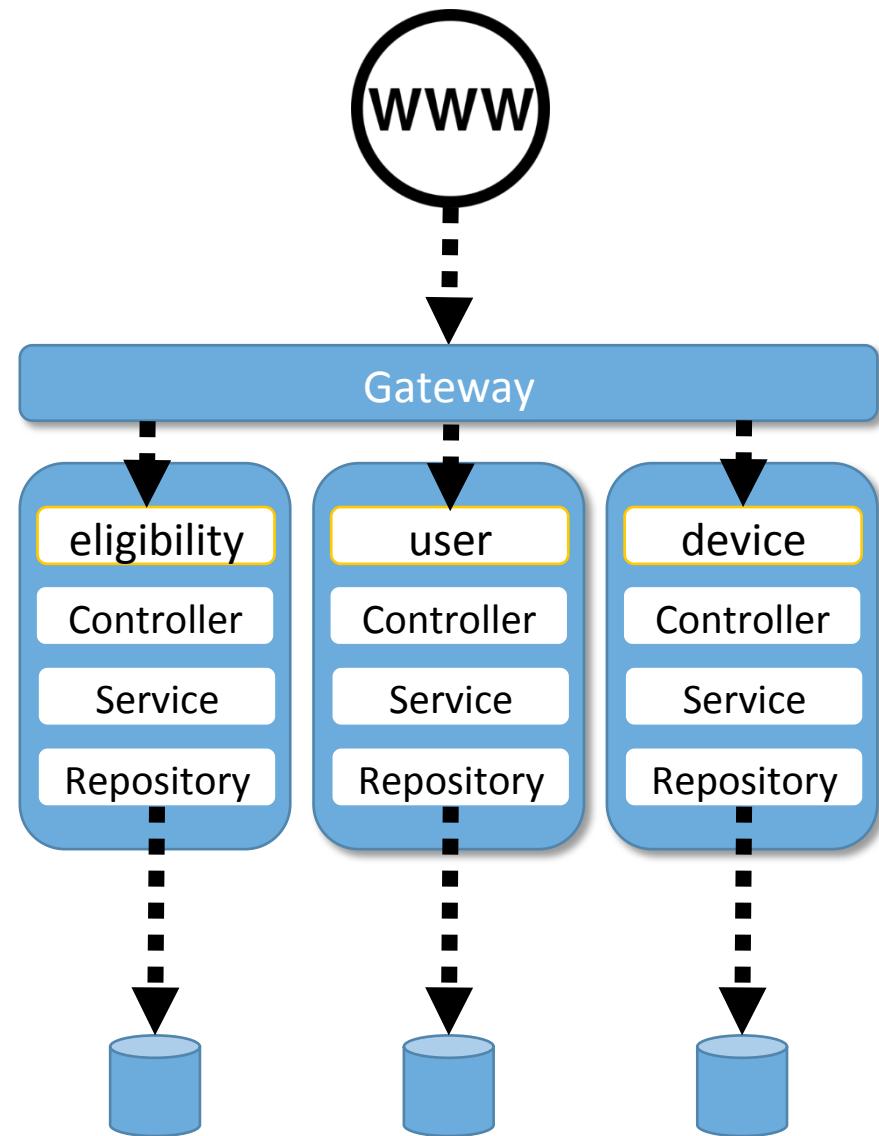


@michaelwinslow | @leslieachapman

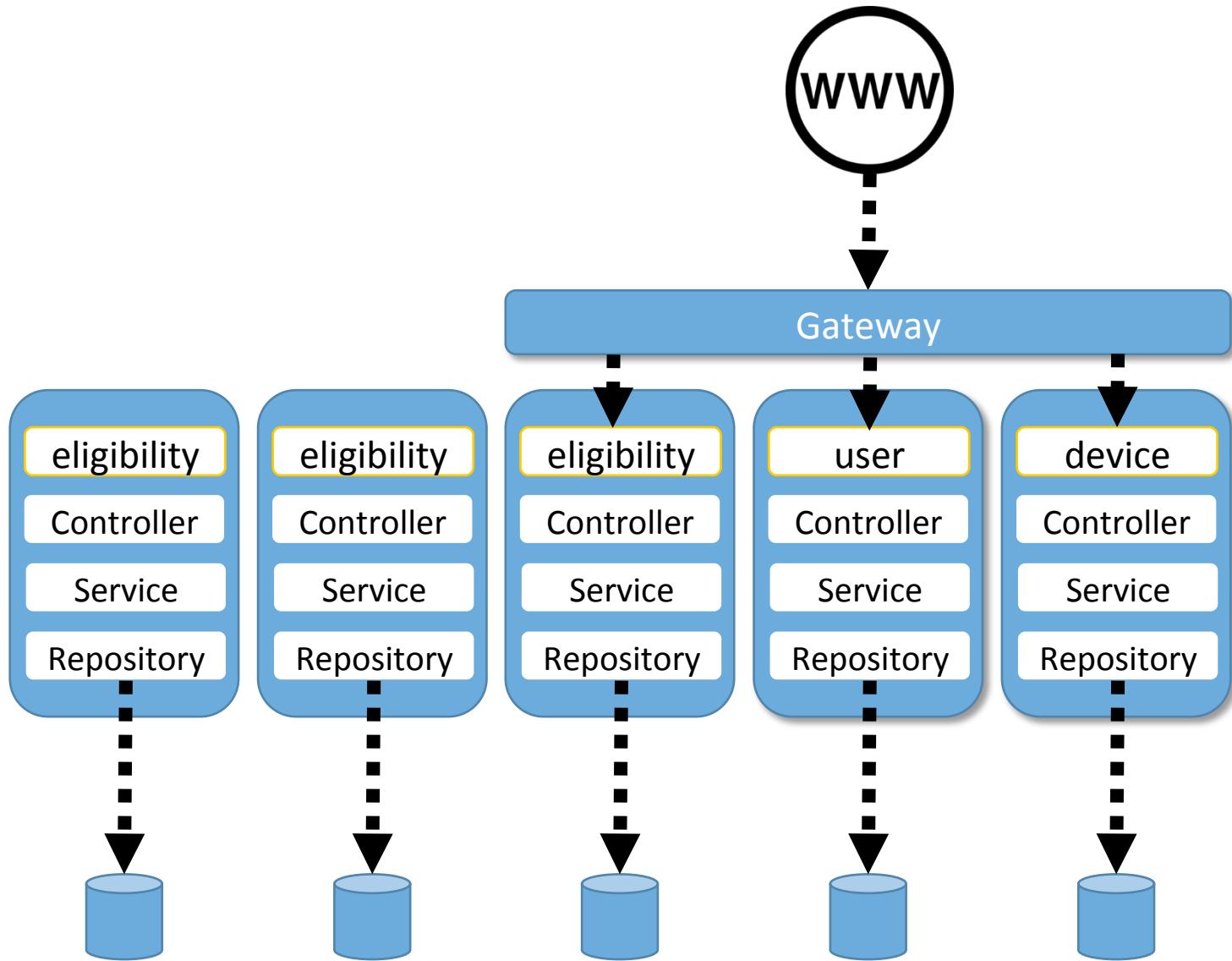


***“Oh no! For some reason requests to the ELIGIBILITY API have gone way up! At least 3x the traffic of any other API!”***





@michaelwinslow | @leslieachapman

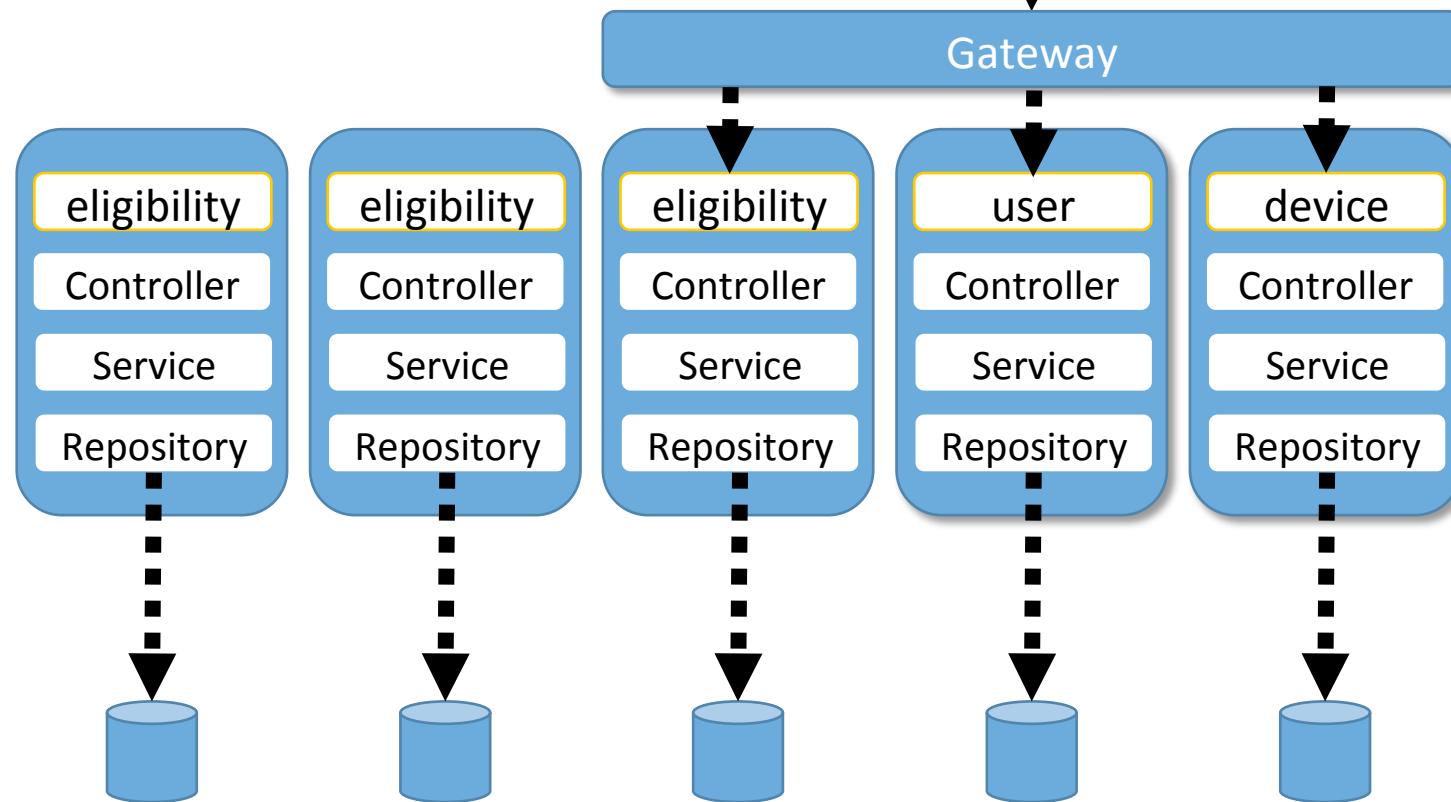
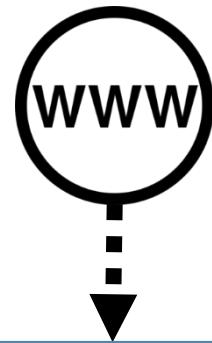


@michaelwinslow | @leslieachapman



**NETFLIX**  
**EUREKA**

Discovery



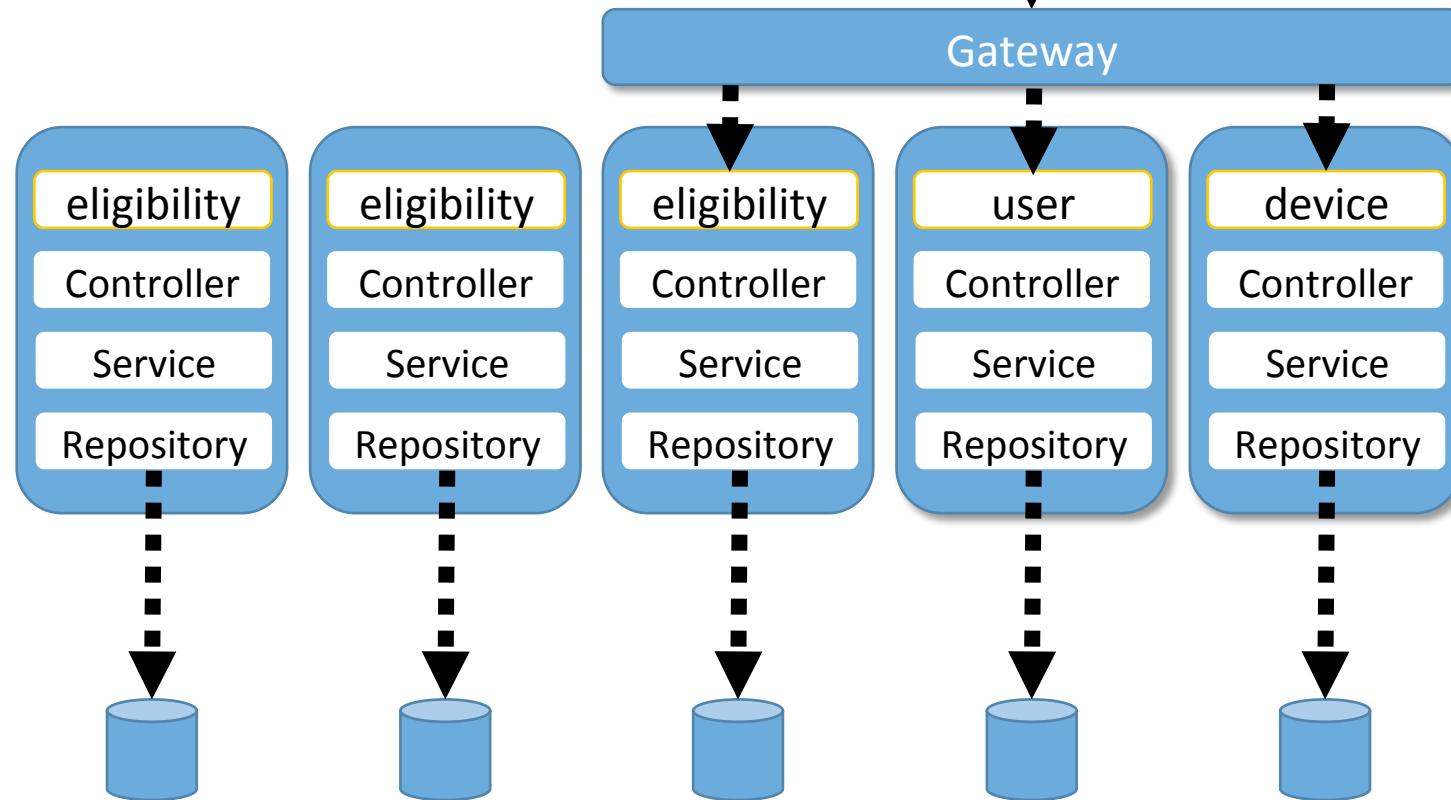
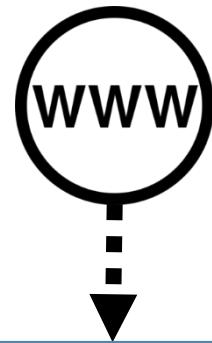
@michaelwinslow | @leslieachapman

COMCAST  
DevOps



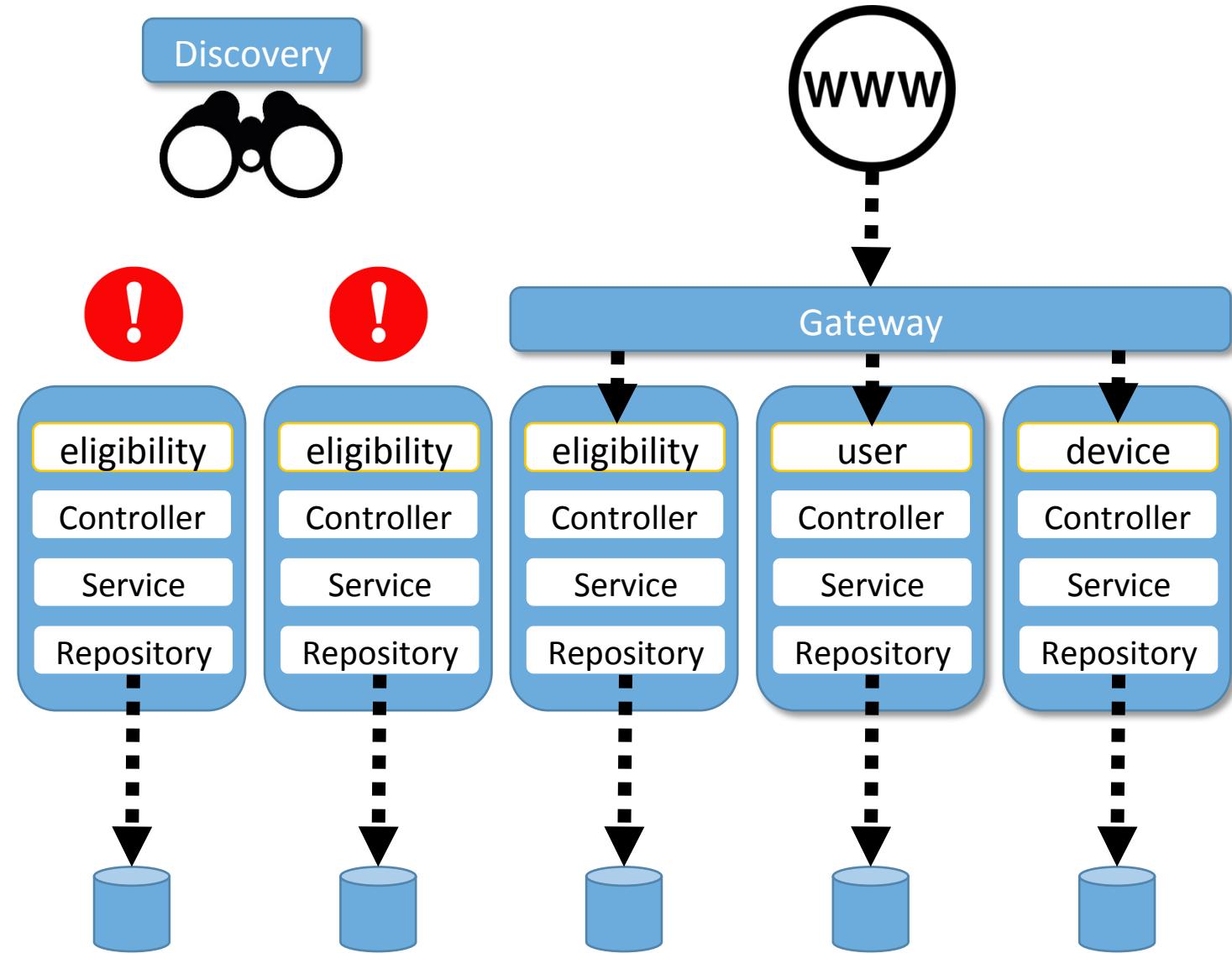
**NETFLIX**  
**EUREKA**

Discovery



@michaelwinslow | @leslieachapman

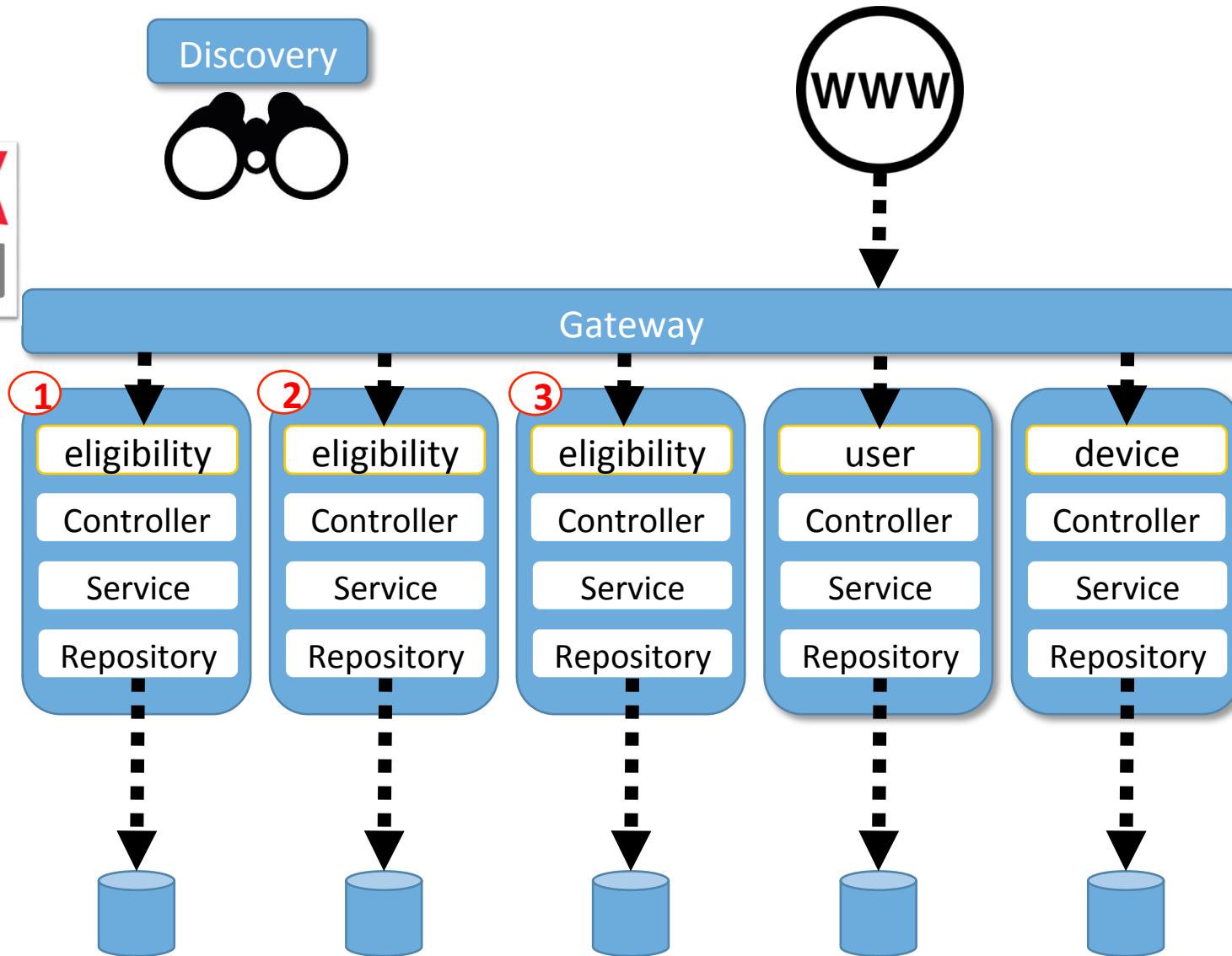
COMCAST  
DevOps



@michaelwinslow | @leslieachapman



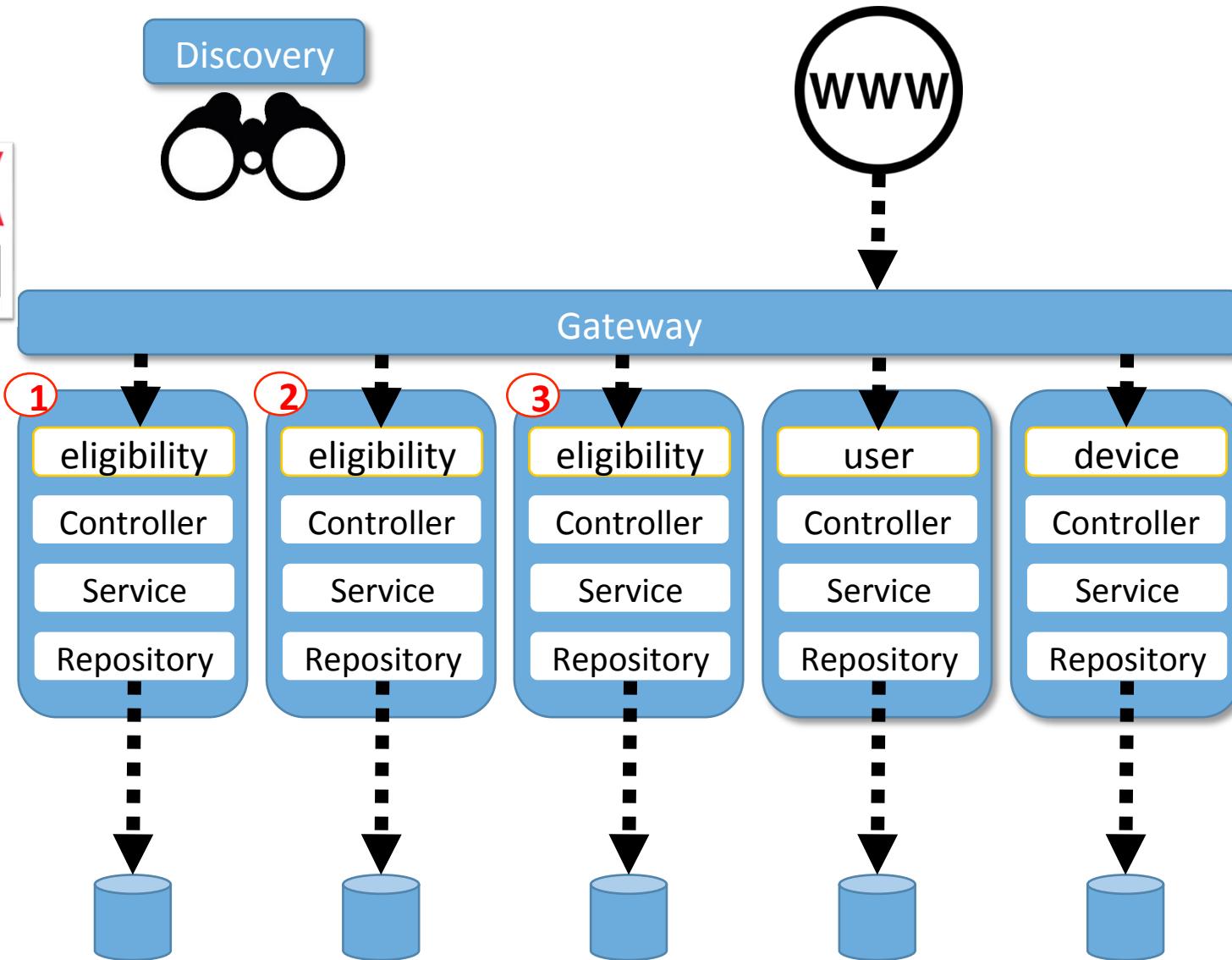
# NETFLIX RIBBON



@michaelwinslow | @leslieachapman

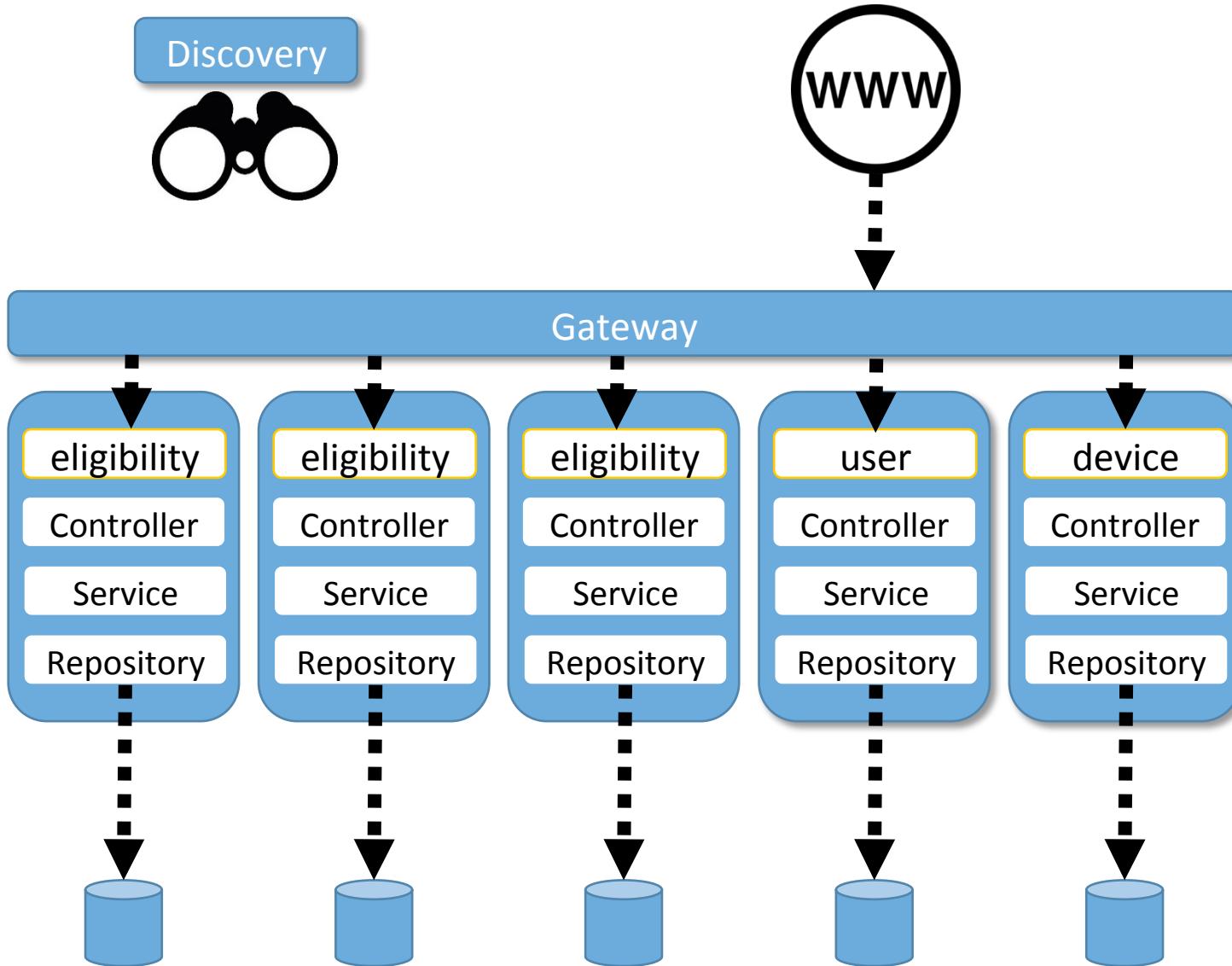


# NETFLIX RIBBON



@michaelwinslow | @leslieachapman

**Note:**  
Elasticity is one of the major reasons to move to microservices. If your traffic tends to increase and decrease dramatically, you may find value. Especially in the cloud.



# Fault Tolerance / Circuit Breaker

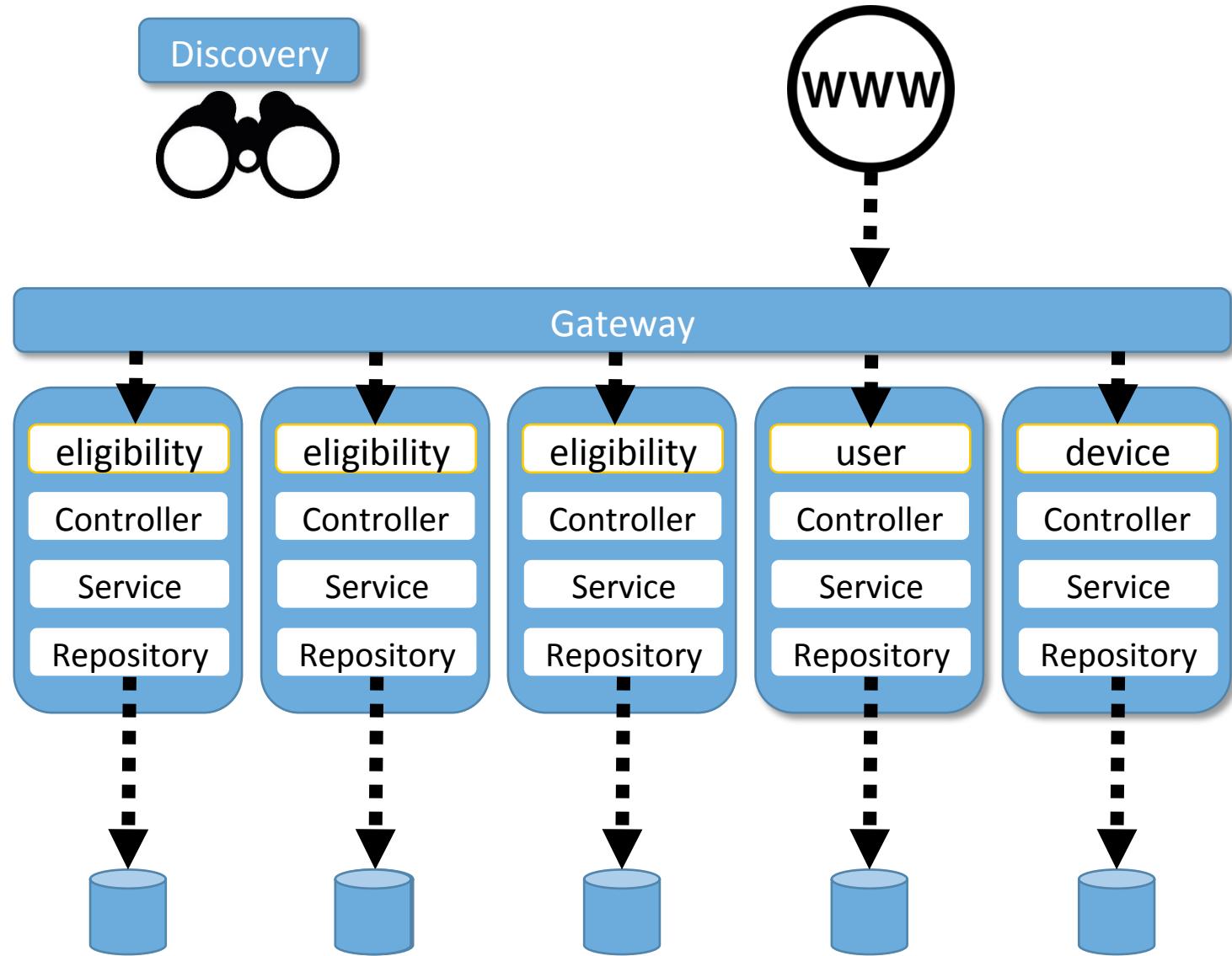
**ASK YOUR TEAM**

*“How do our services handle failure conditions?”*

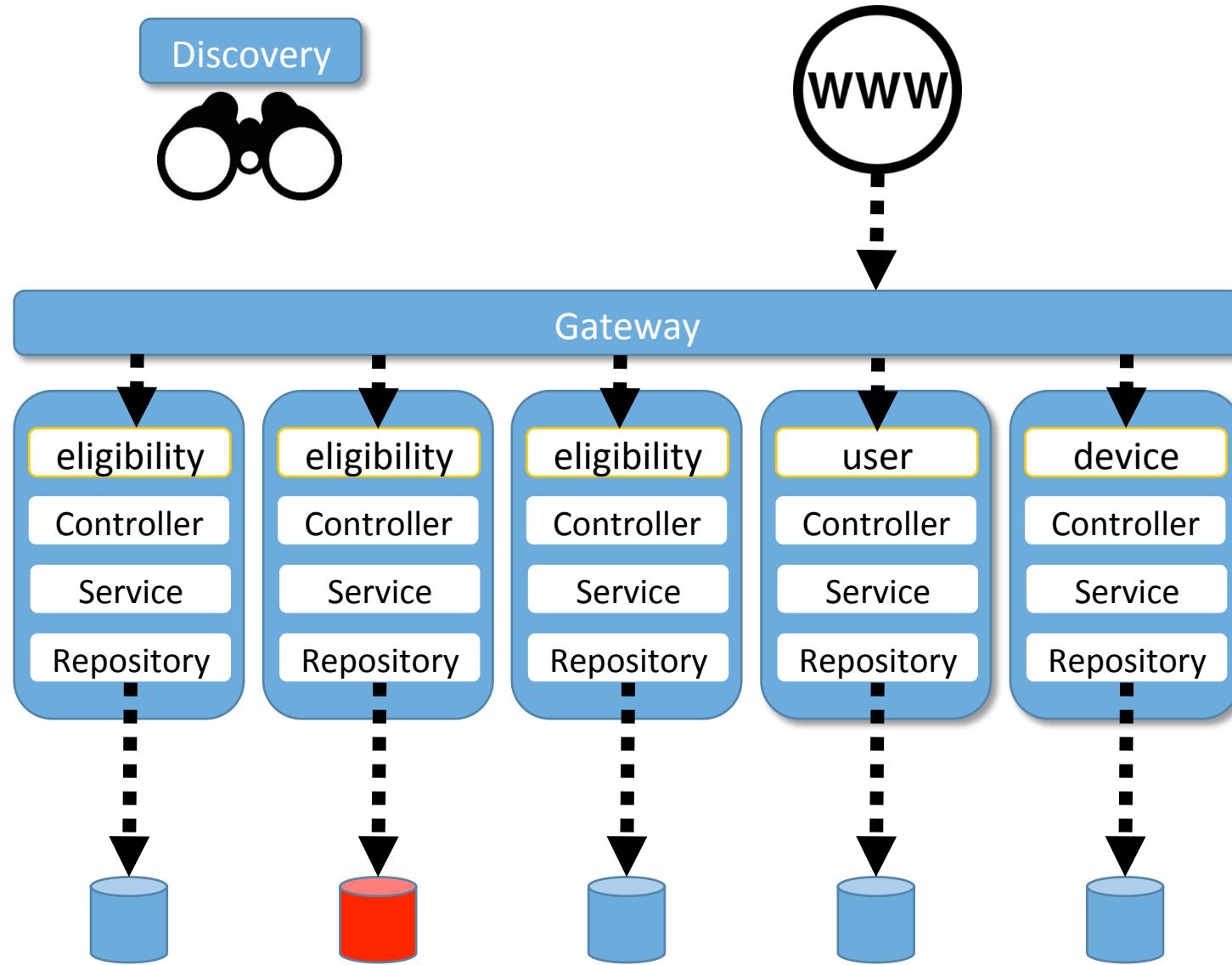


@michaelwinslow | @leslieachapman

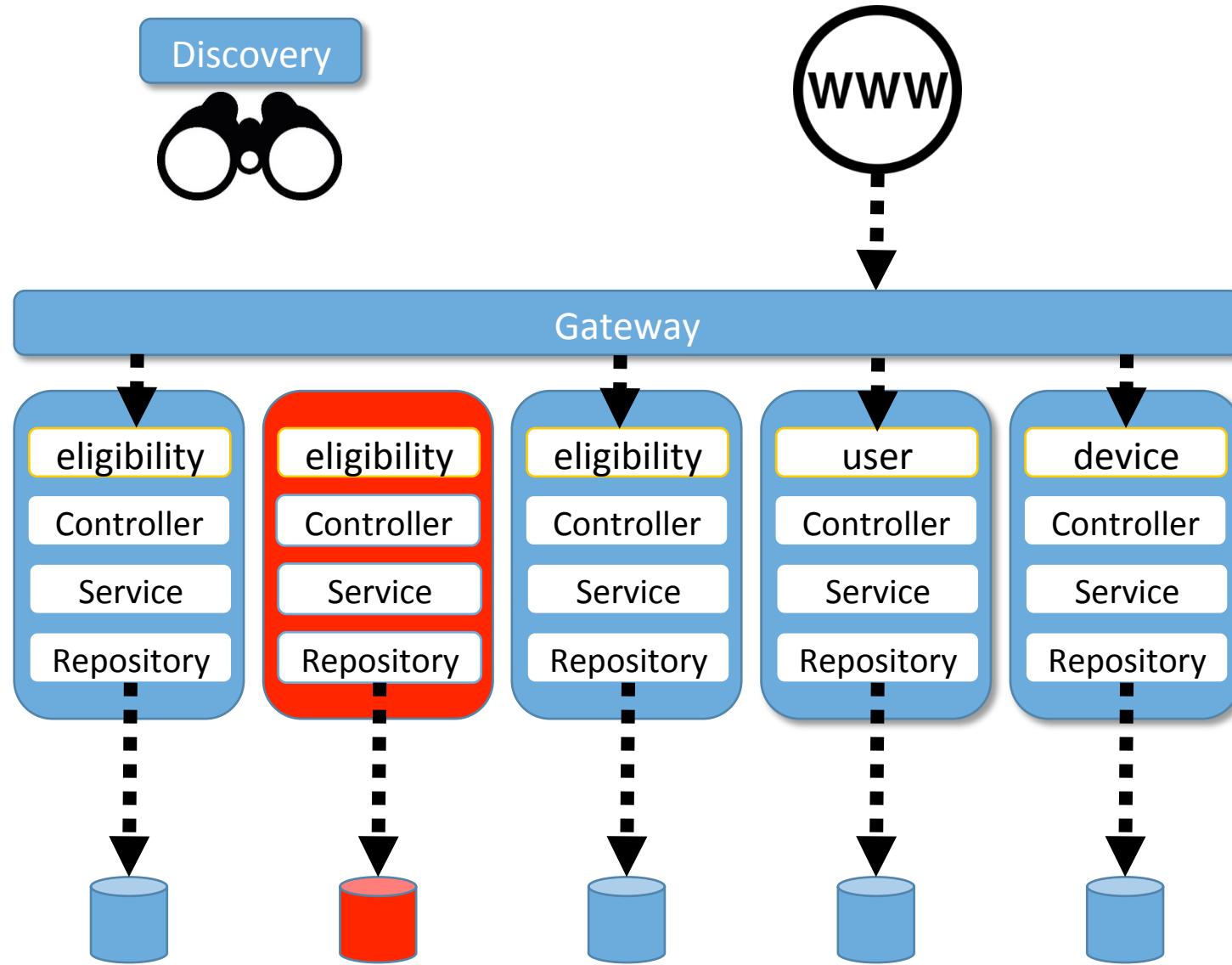




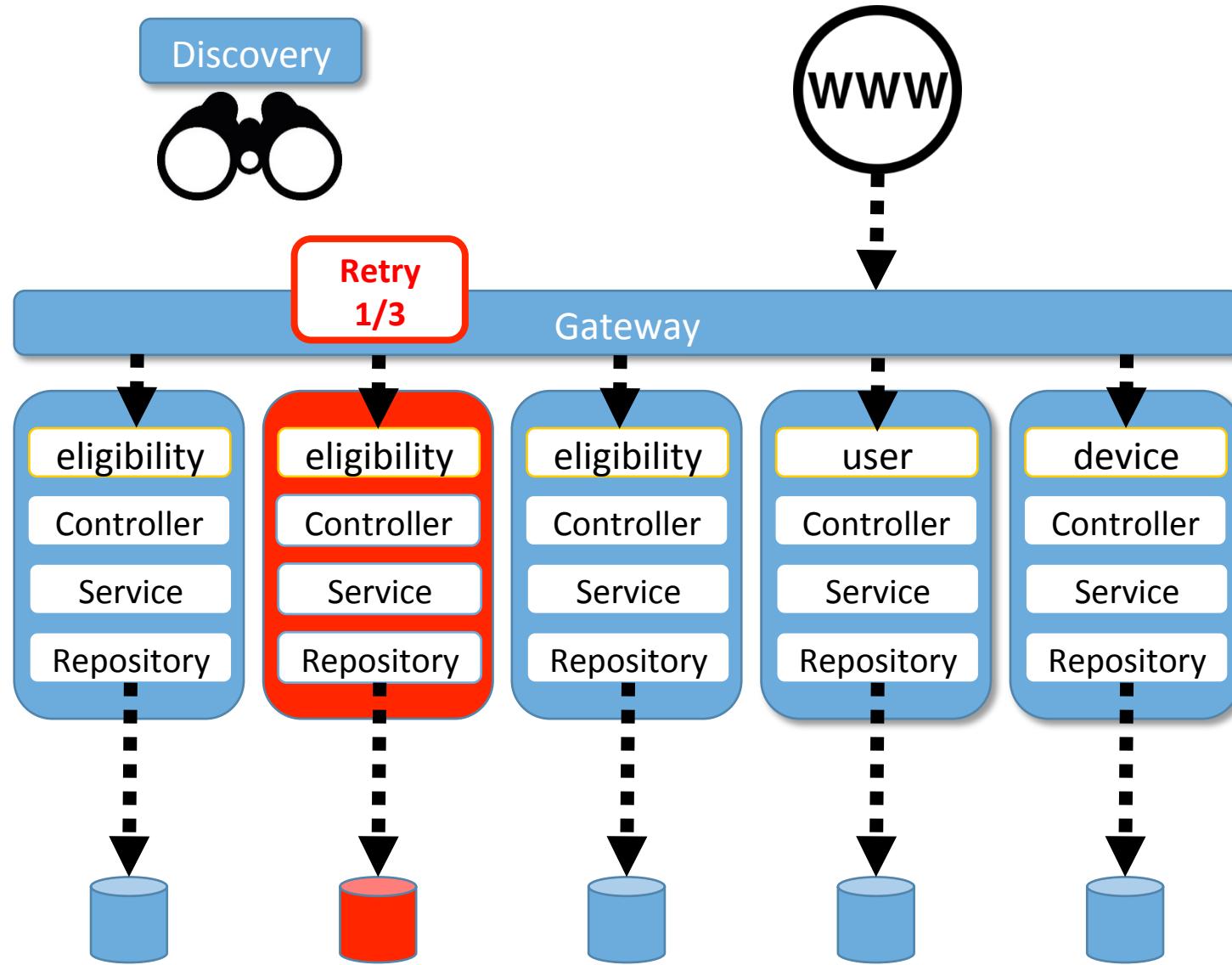
@michaelwinslow | @leslieachapman



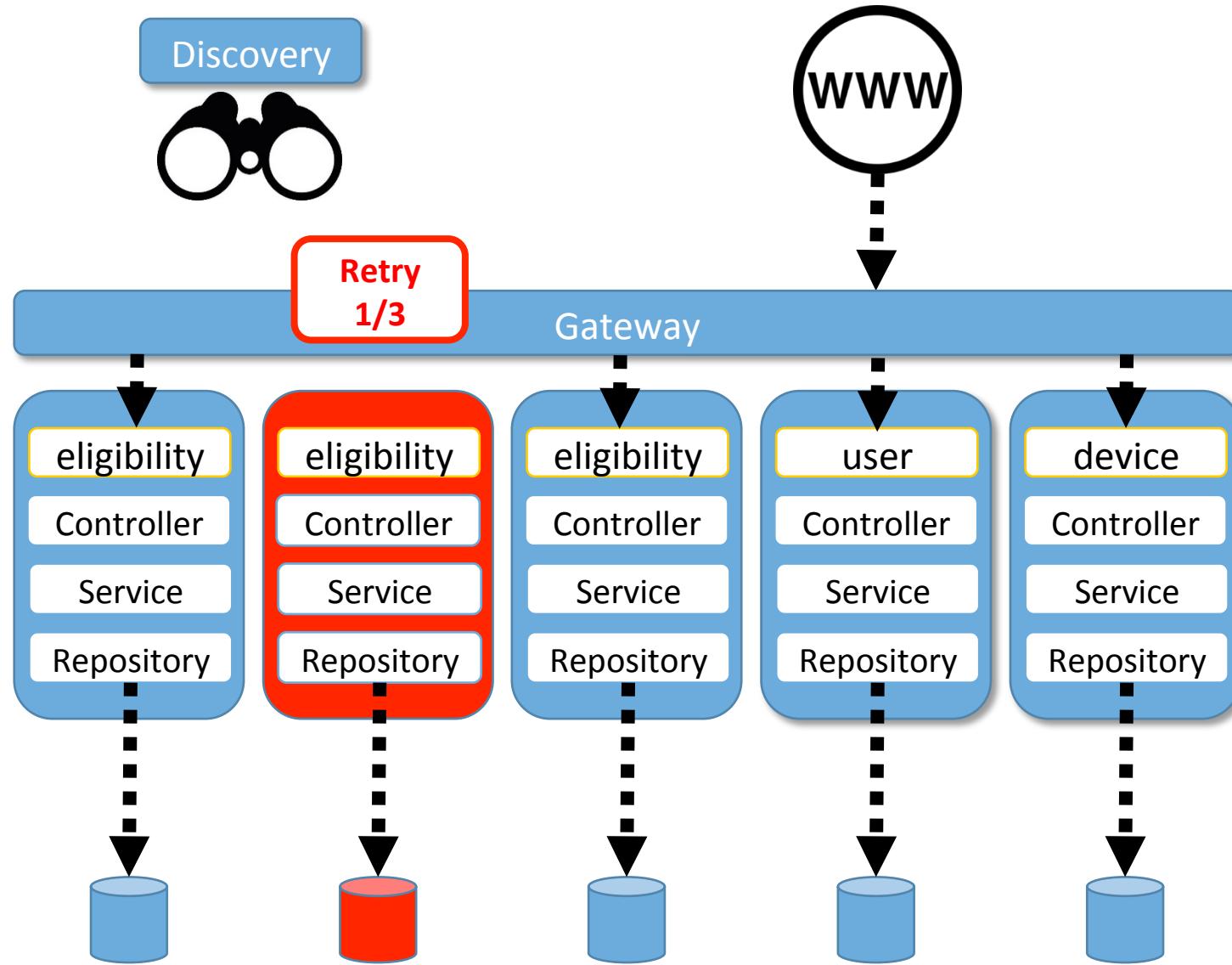
@michaelwinslow | @leslieachapman



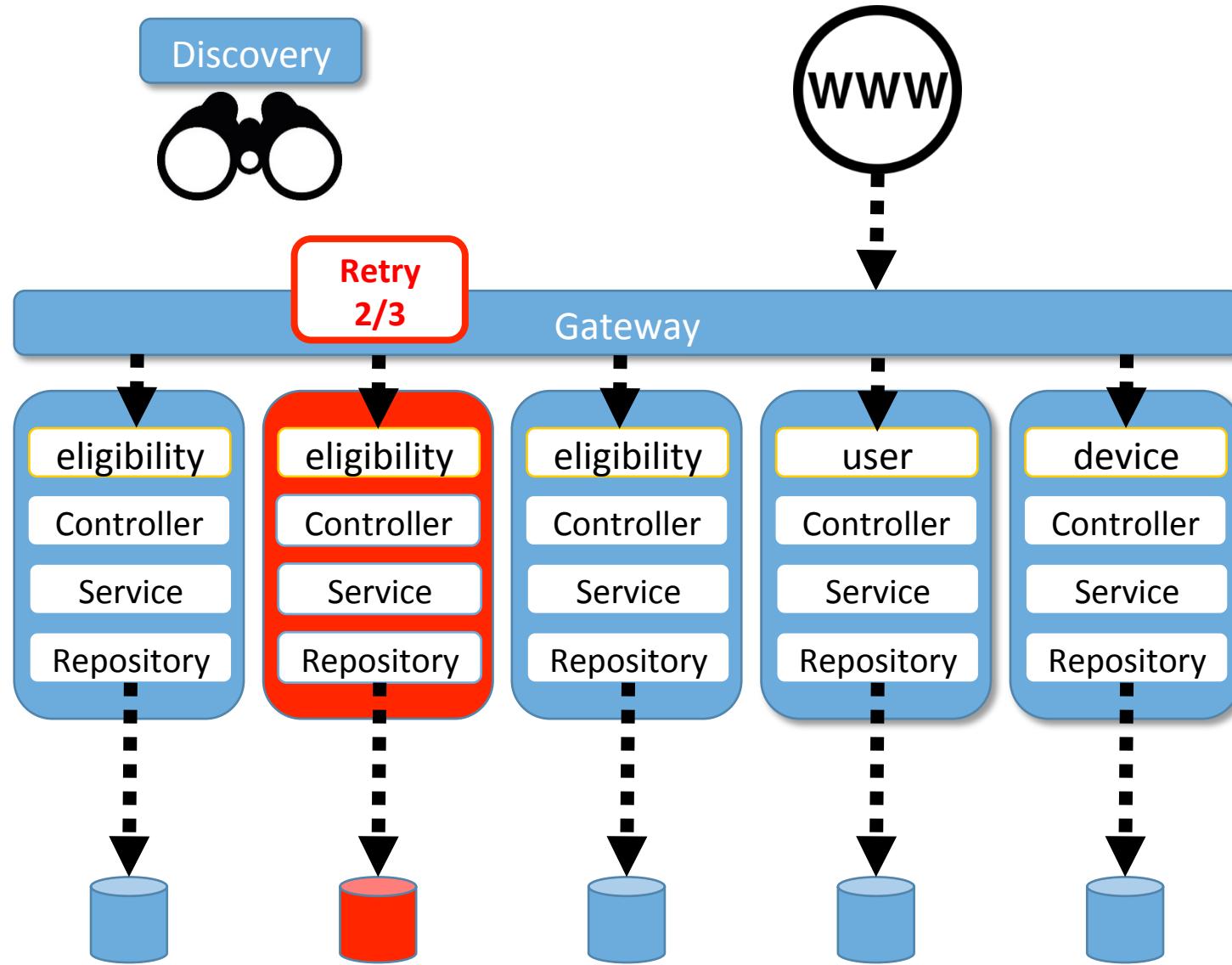
@michaelwinslow | @leslieachapman



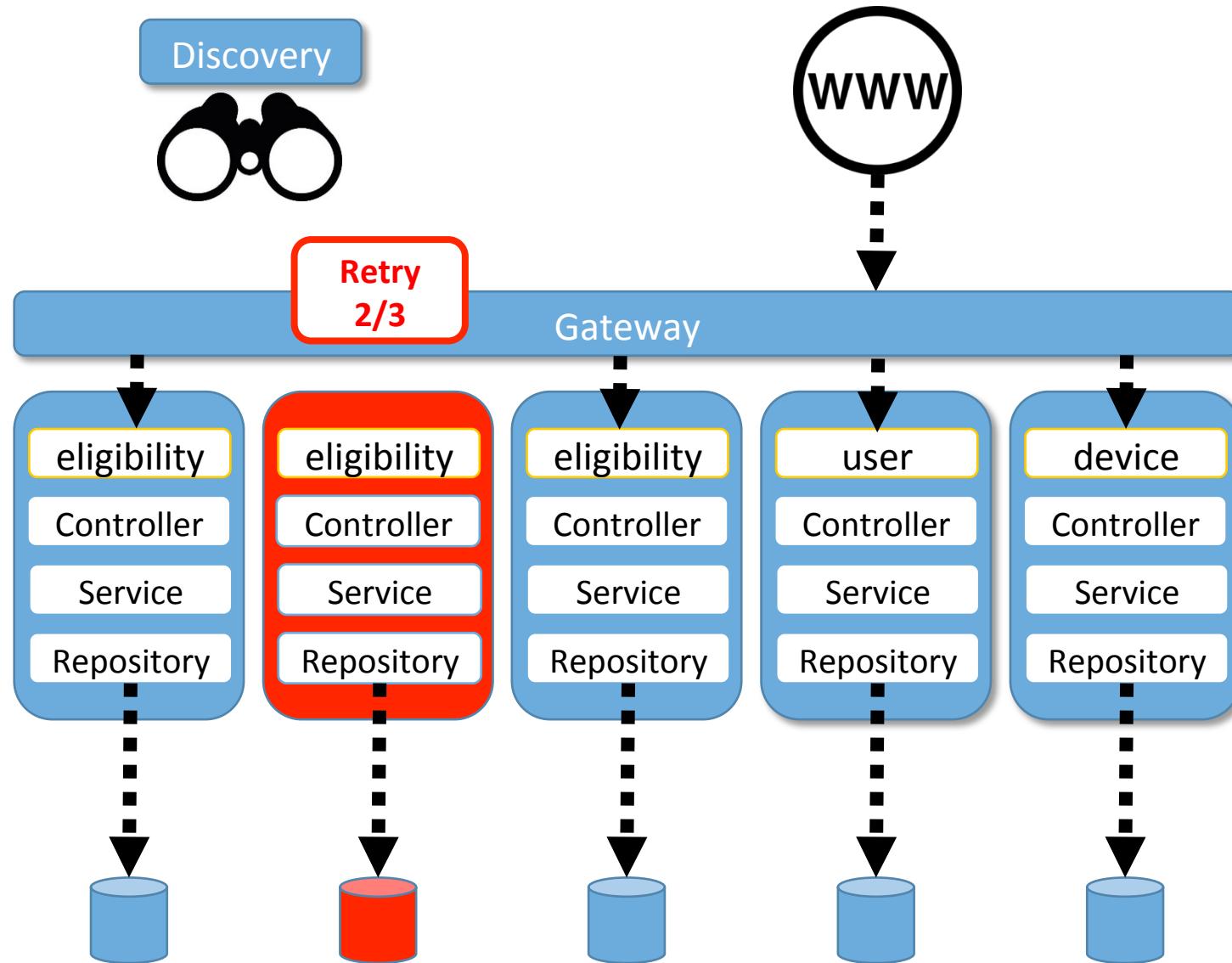
@michaelwinslow | @leslieachapman



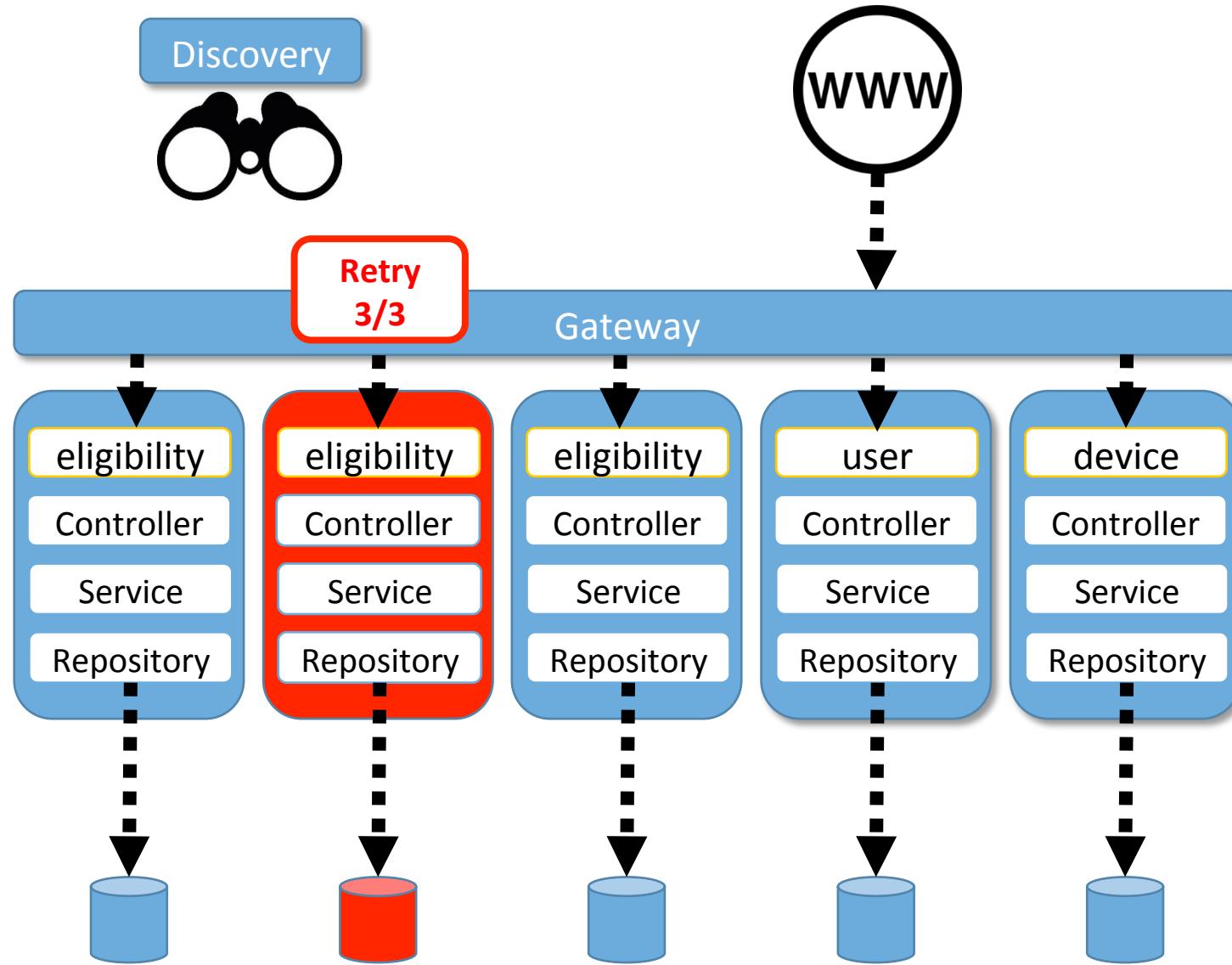
@michaelwinslow | @leslieachapman



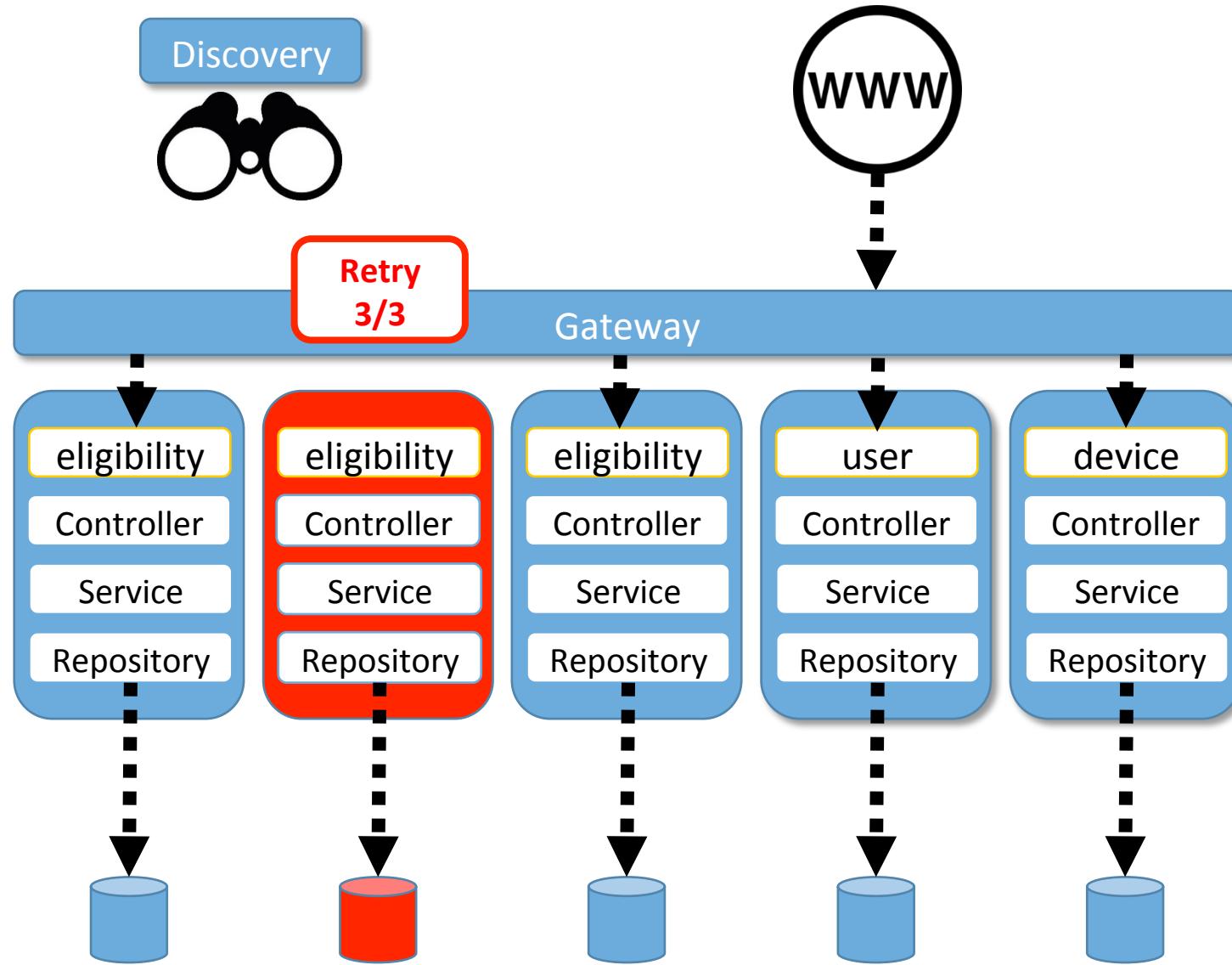
@michaelwinslow | @leslieachapman



@michaelwinslow | @leslieachapman

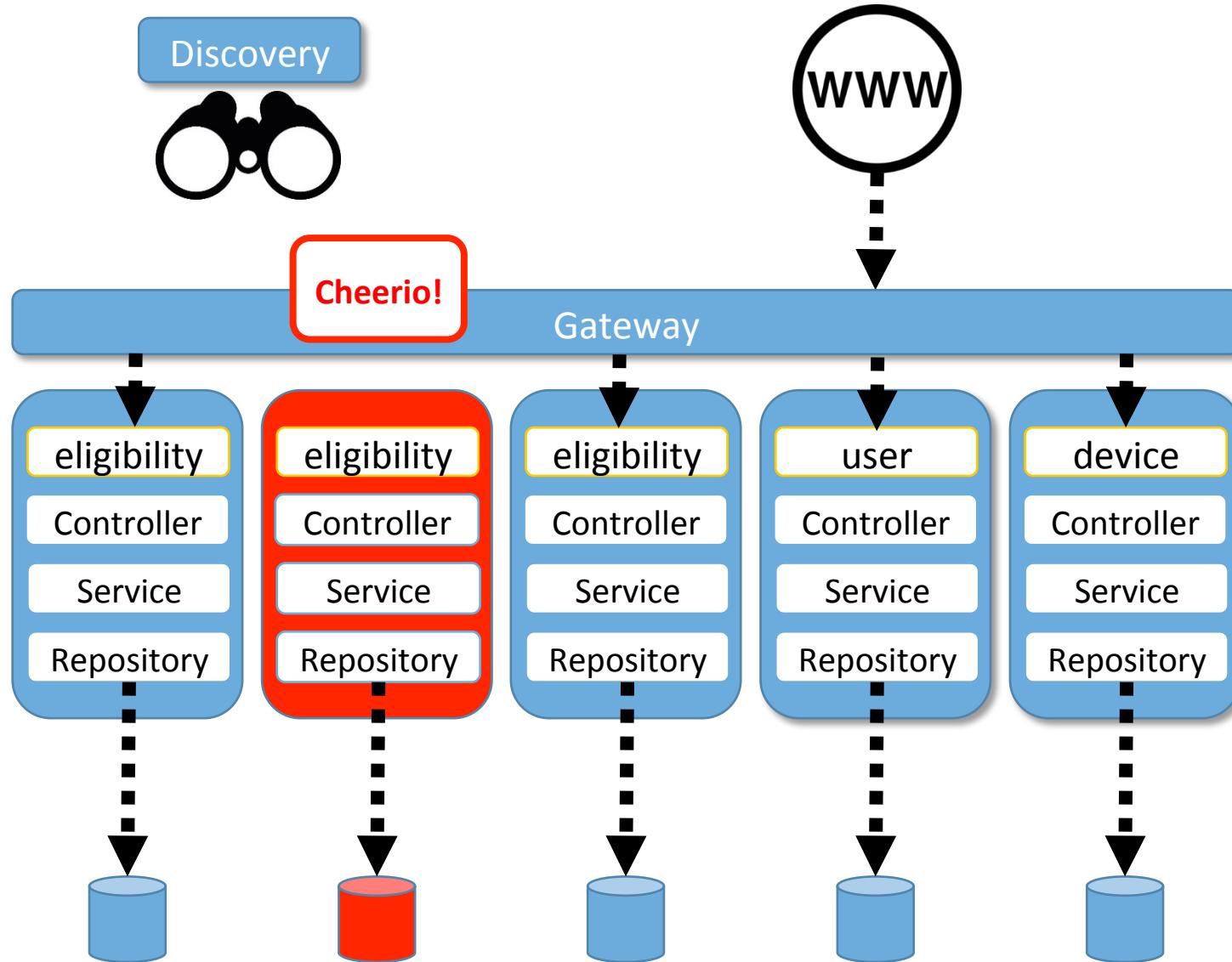


@michaelwinslow | @leslieachapman



@michaelwinslow | @leslieachapman

**Note:**  
Unlike Monoliths, where one failure can affect the entire service, microservices can be designed to limit the impact of failures.



@michaelwinslow | @leslieachapman

# Summary:

1. Monolith vs Microservice
2. Netflix Libraries: Zuul, Eureka, Ribbon, Hystrix

## Benefits:

1. Spread the Codebase Across Functional Teams
2. Elasticity
3. Limit the Impact of Failures

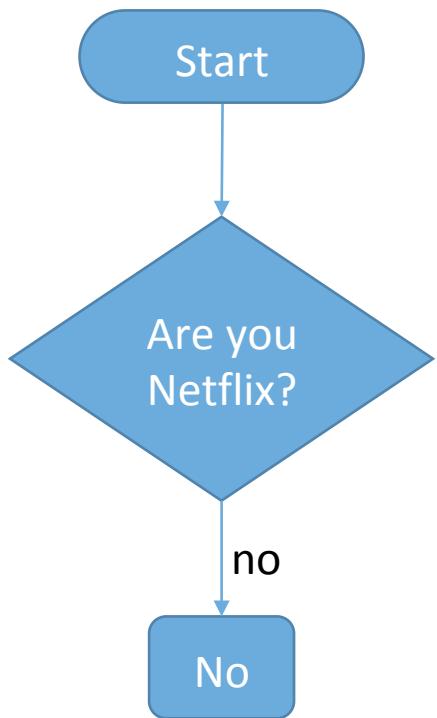
## Now you have a choice:

- That was amazing! Now tell us what your team learnt! (**red**)
- That was amazing! Now tell us what your team learnt! (**red**)



@michaelwinslow | @leslieachapman

# Is There a Need To Move to Microservices?



@michaelwinslow | @leslieachapman



**“Complexity begets complexity.** Soon tools are needed to manage the complexity. Then *tools are needed to manage the tools.*” – Ryan Emerle, Principal Engineer

 @remerle



@michaelwinslow | @leslieachapman

# Impact on System Logs

**ASK YOUR TEAM**

*“Do we know how much our log file storage and network traffic will increase?”*



@michaelwinslow | @leslieachapman



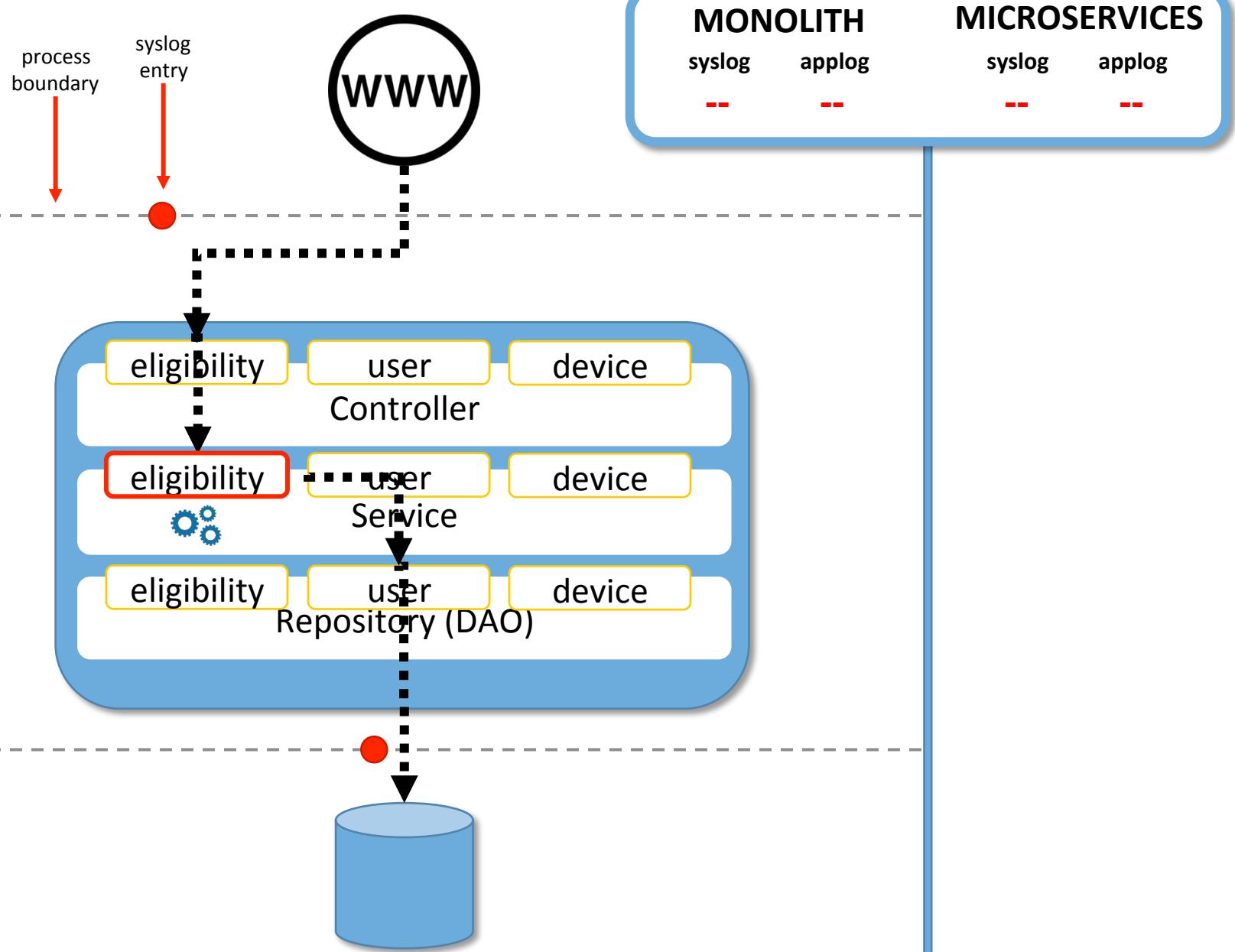
## **Sample:**

We need to call our API to determine  
if a particular **USER** is **ELIGIBLE**  
to purchase a **DEVICE**

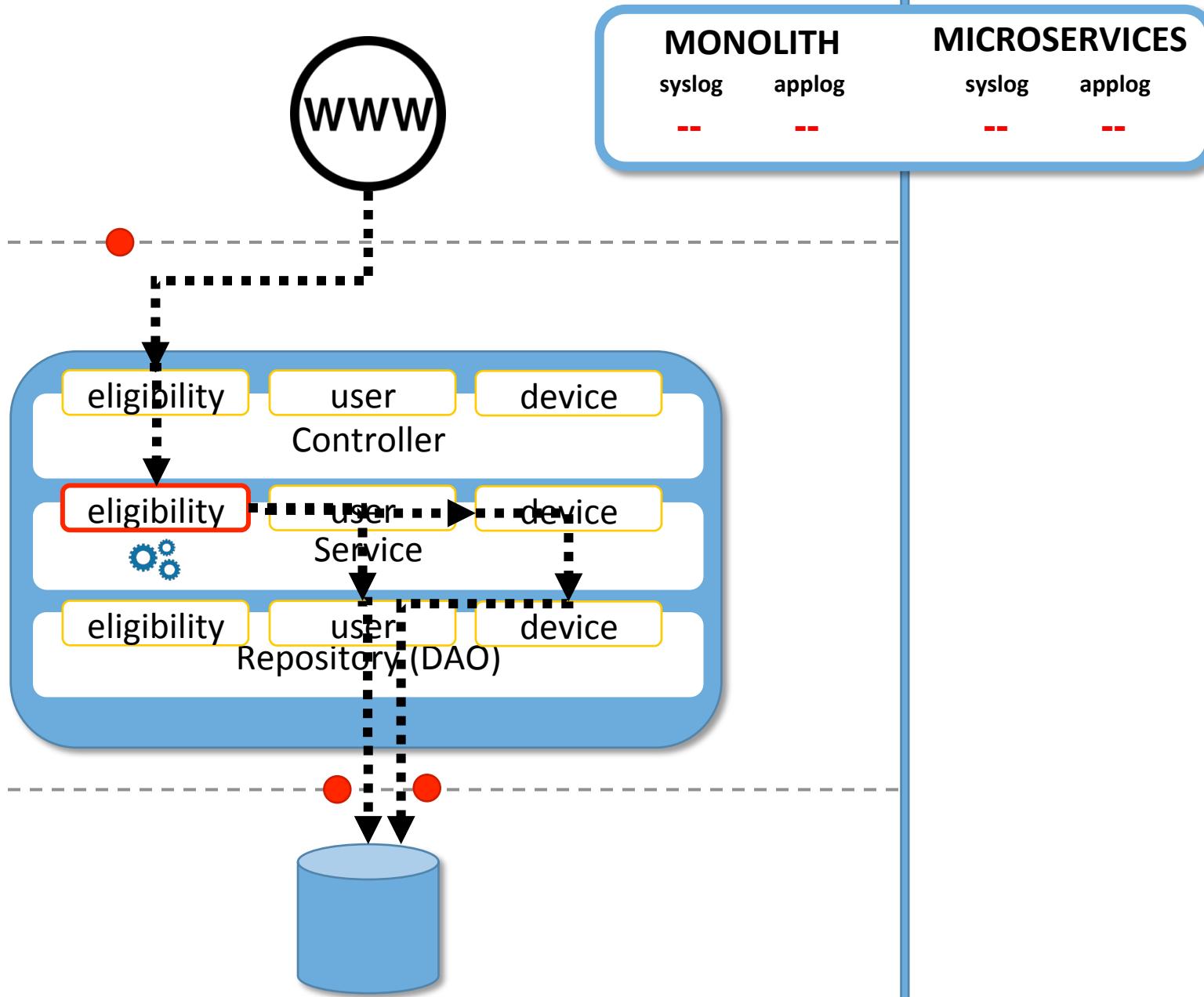


@michaelwinslow | @leslieachapman





@michaelwinslow | @leslieachapman



@michaelwinslow | @leslieachapman

## MONOLITH

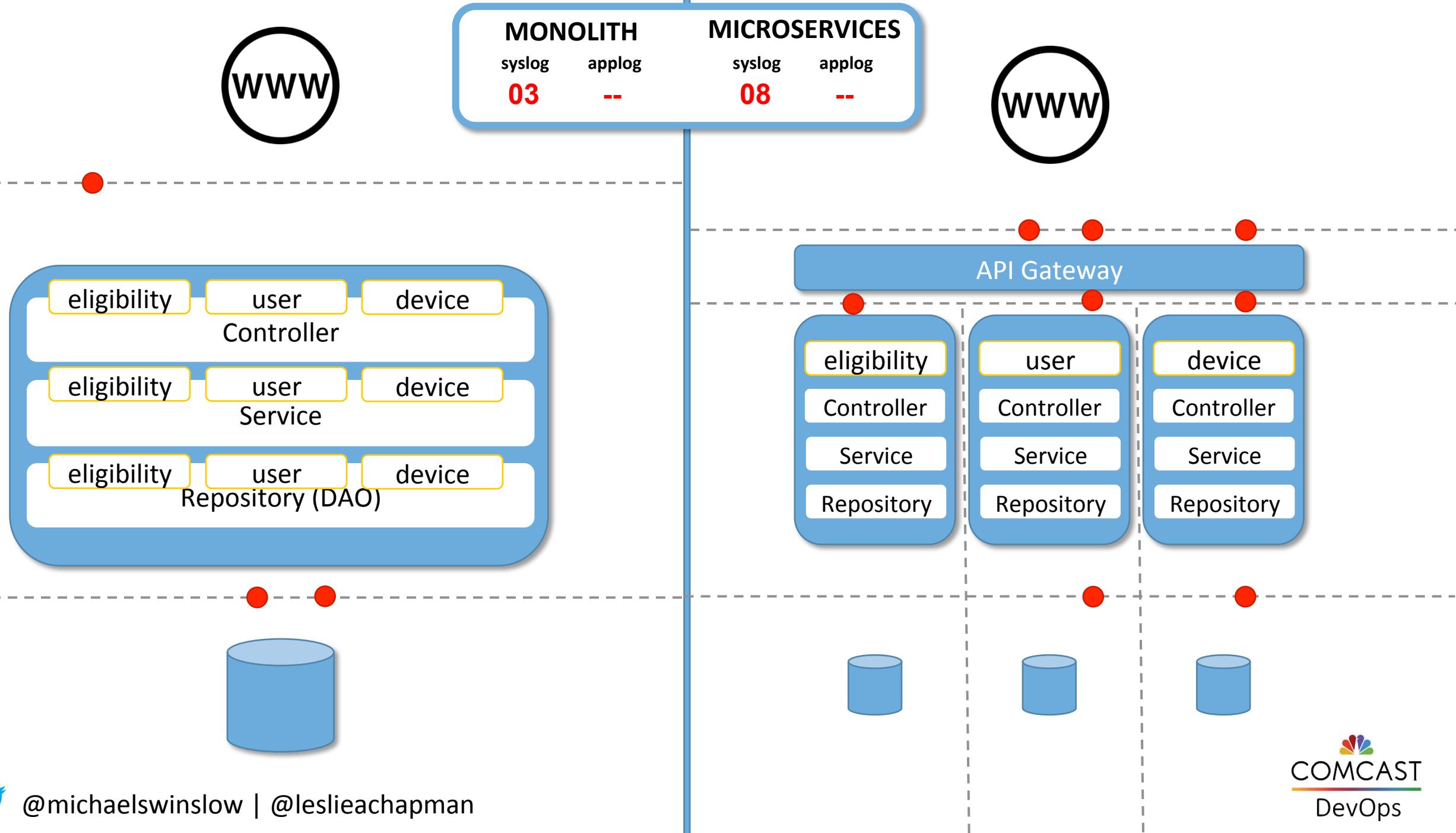
syslog applog

**03**

## MICROSERVICES

syslog applog

-- --





**Note:**

Increased number of process boundaries also creates more surface area for security attacks during runtime.



**So What?**  
**Microservices** inflate the System Logs  
a little (and security risk).

It's totally worth it!



@michaelwinslow | @leslieachapman



**Maybe...**  
but we're not finished yet.



@michaelwinslow | @leslieachapman



# Impact on Application Logs

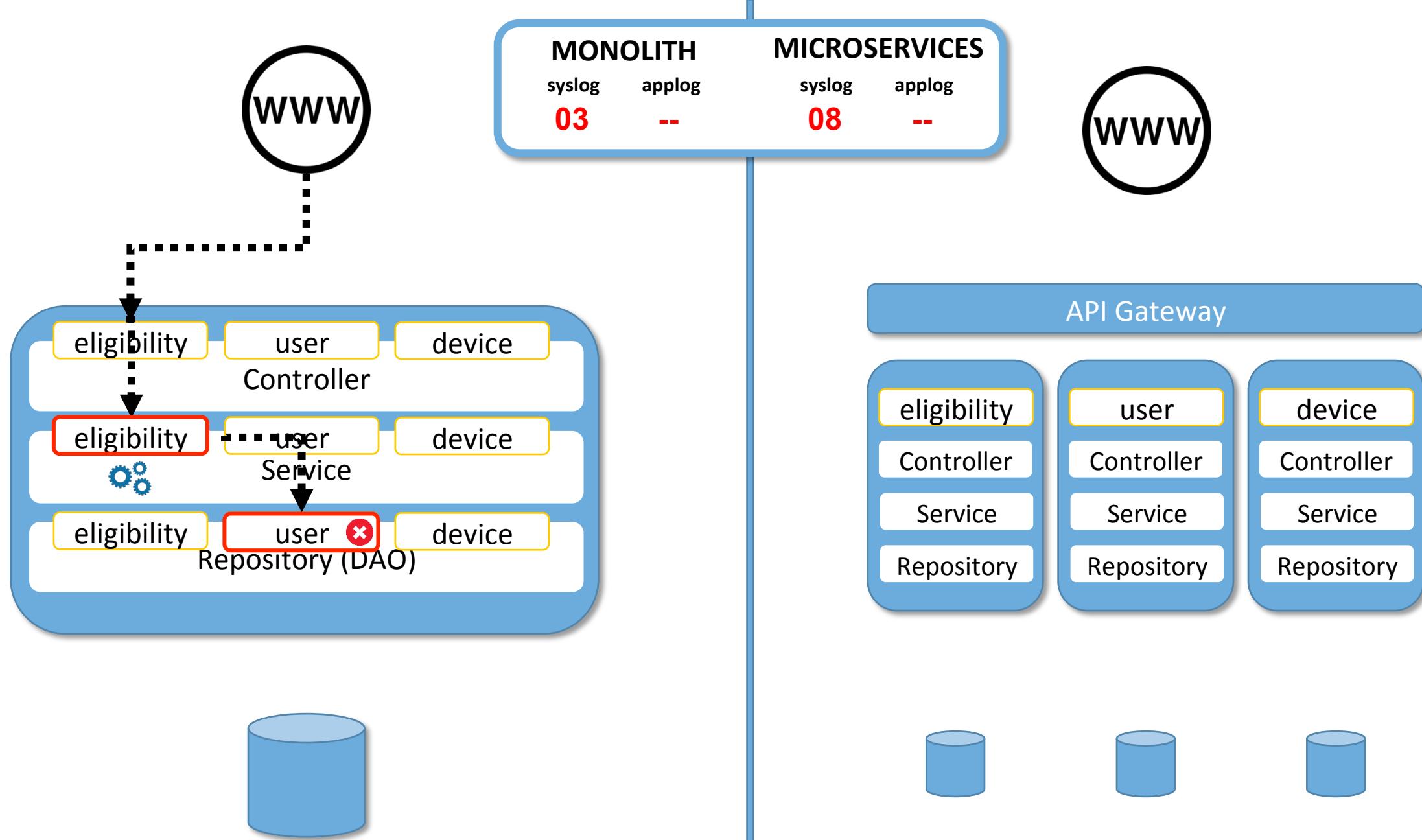
**ASK YOUR TEAM**

*“How do we make sure we can diagnose production issues?”*

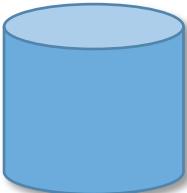
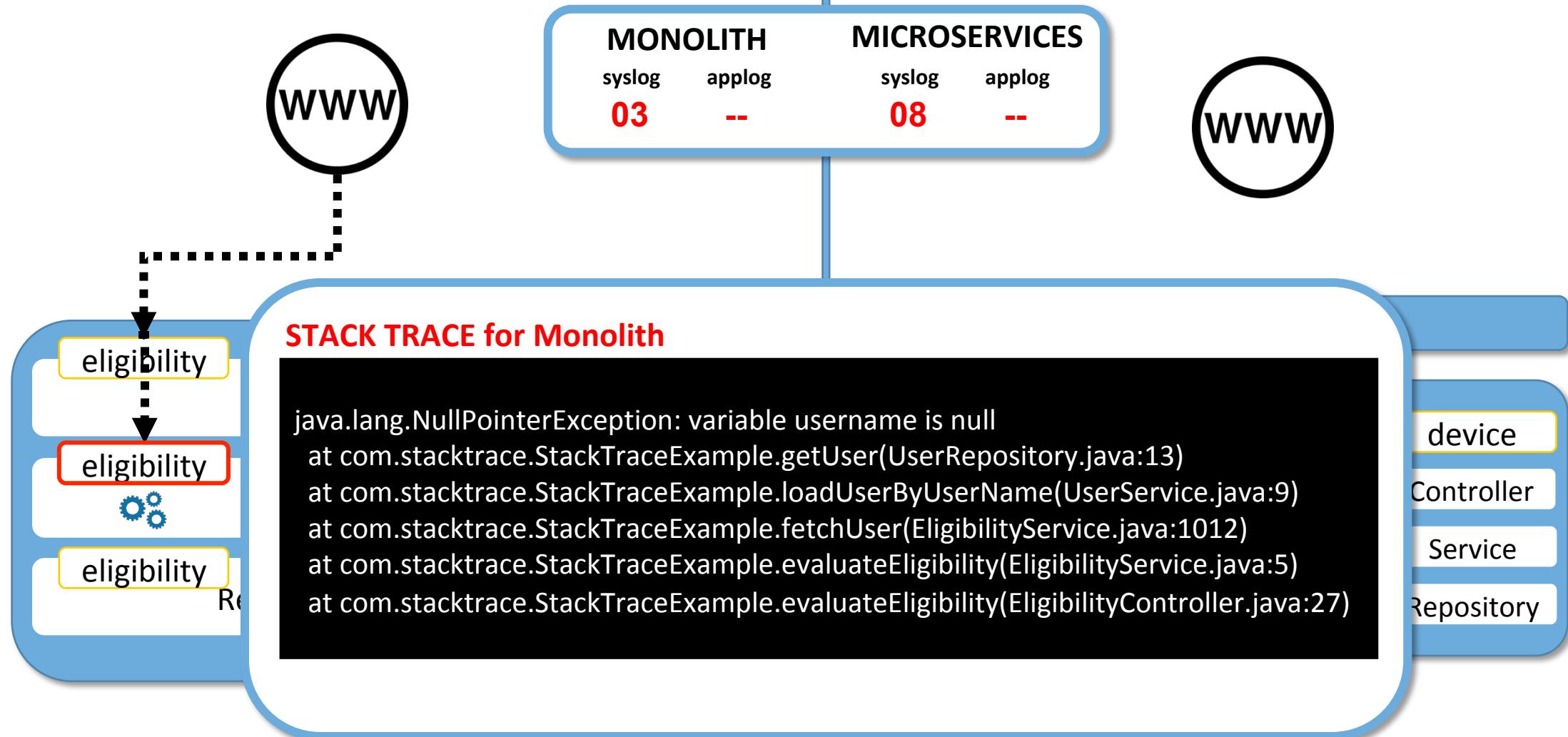


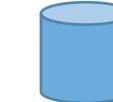
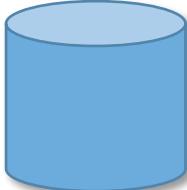
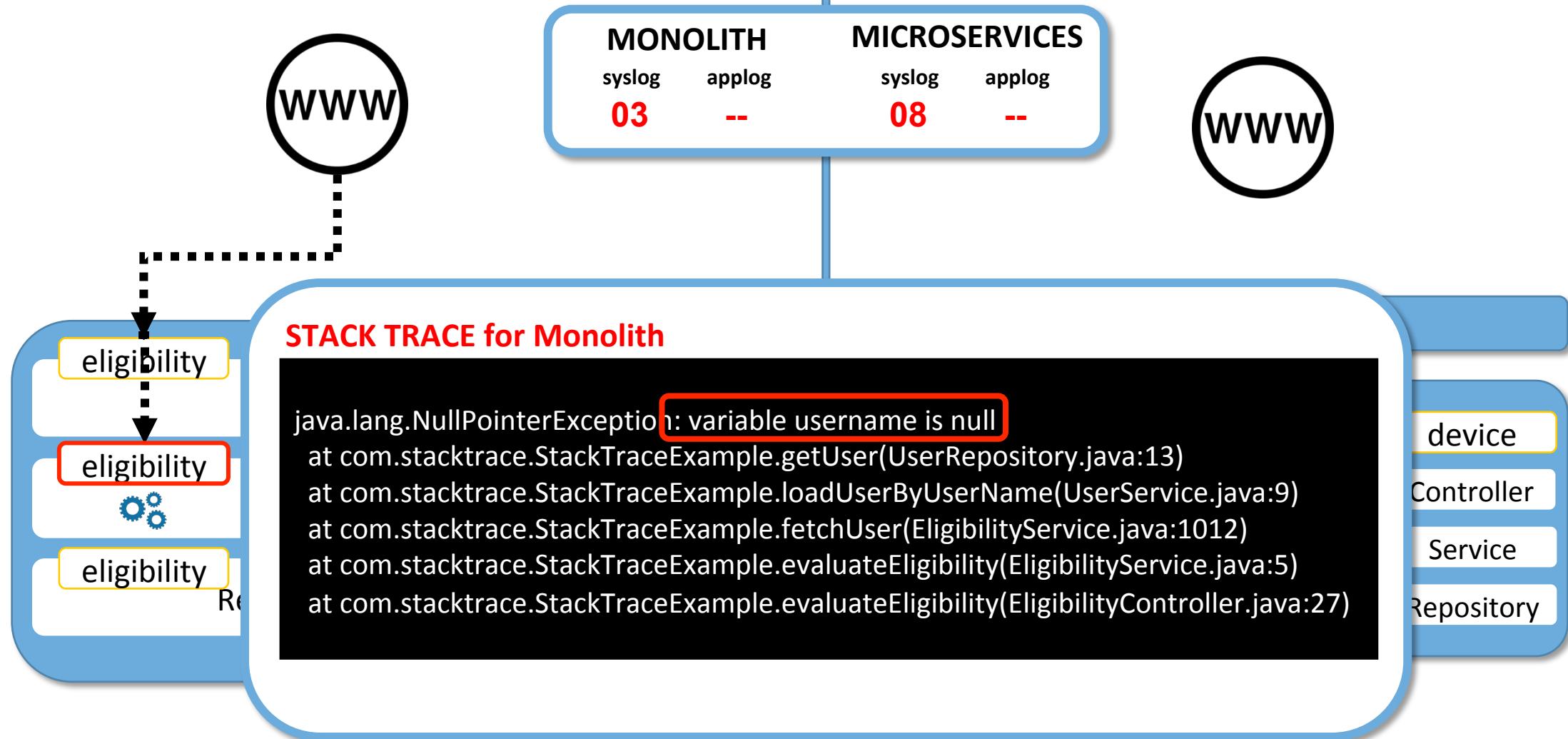
@michaelwinslow | @leslieachapman

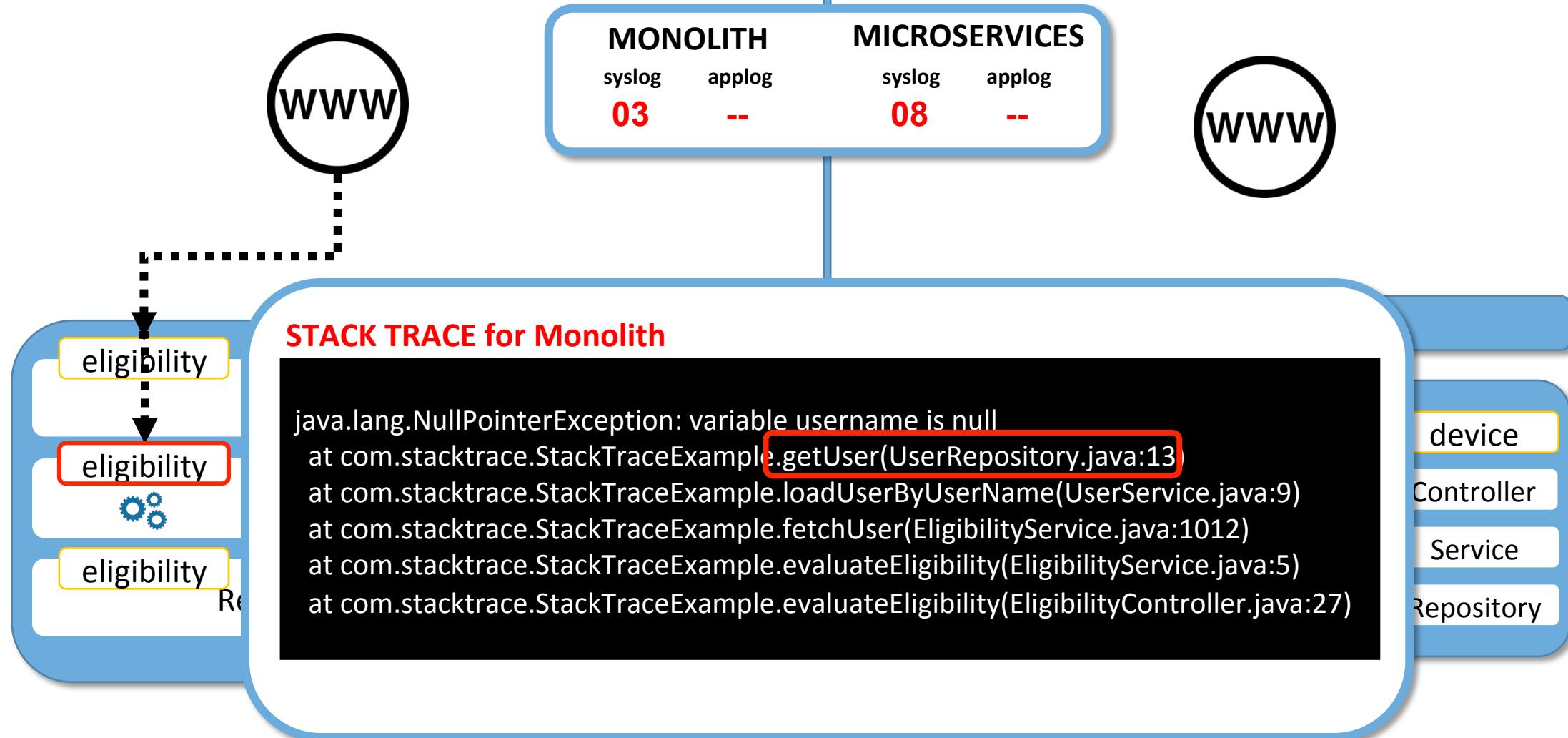




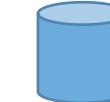
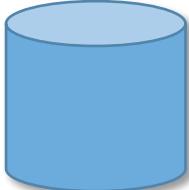
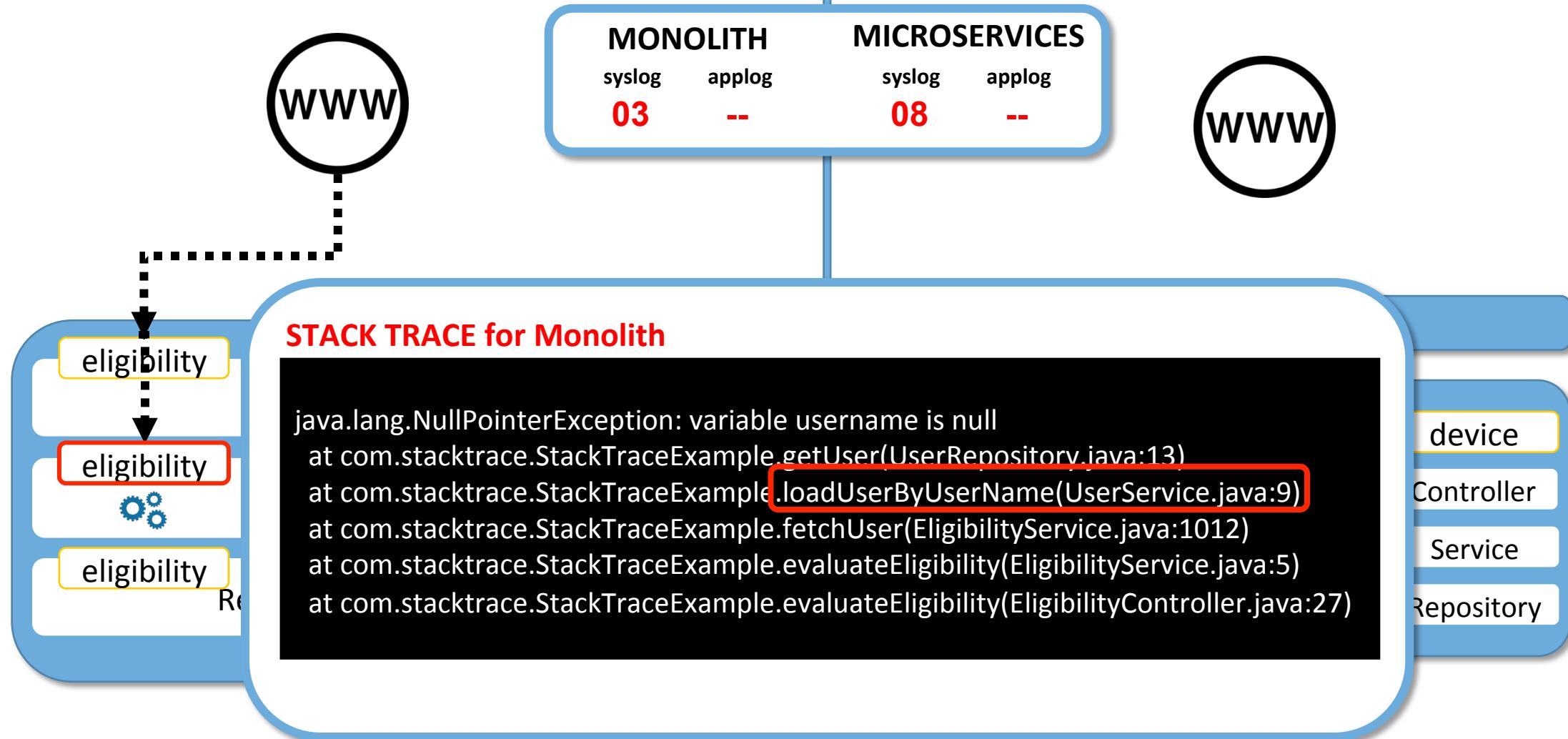
@michaelwinslow | @leslieachapman

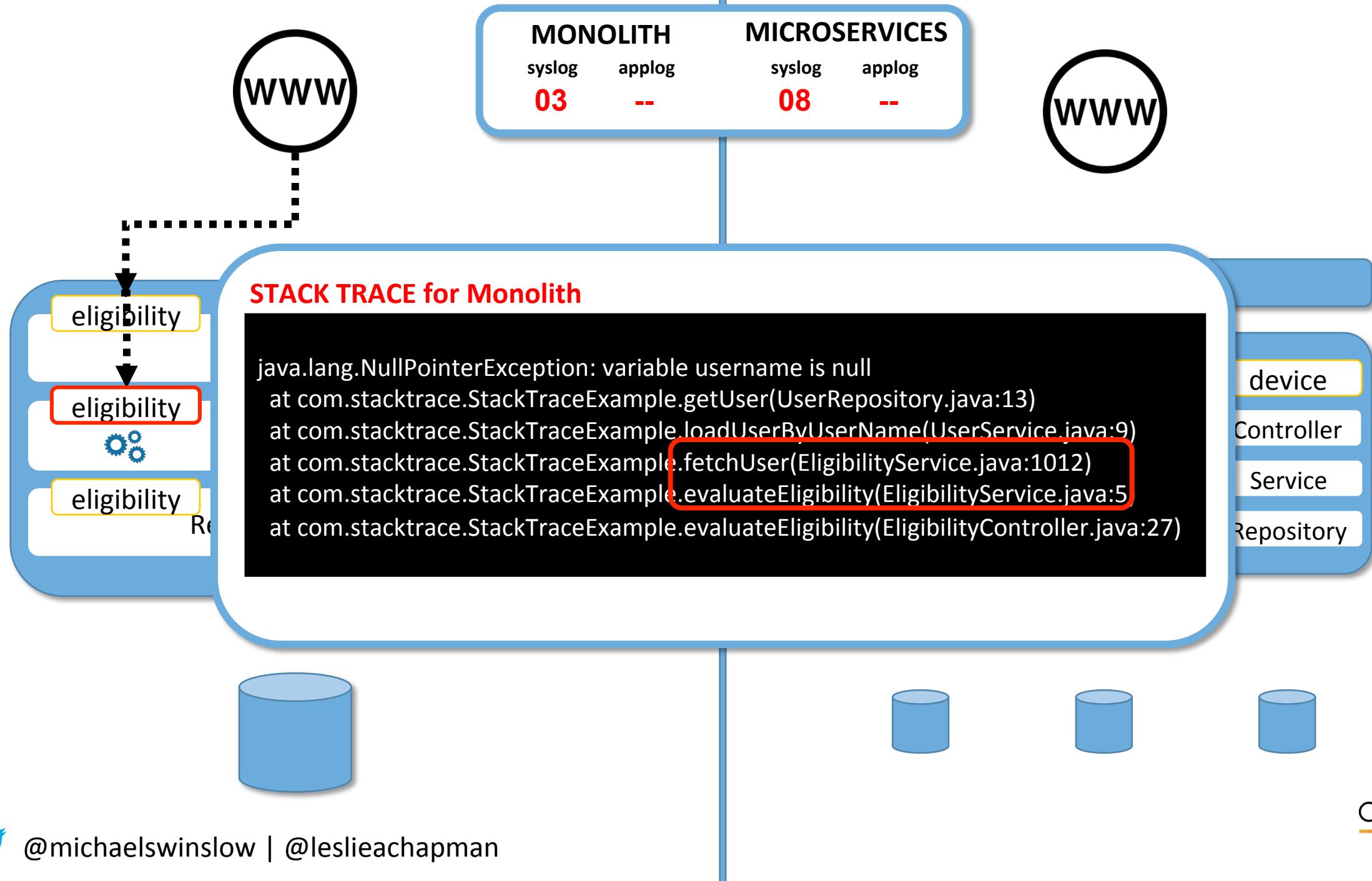


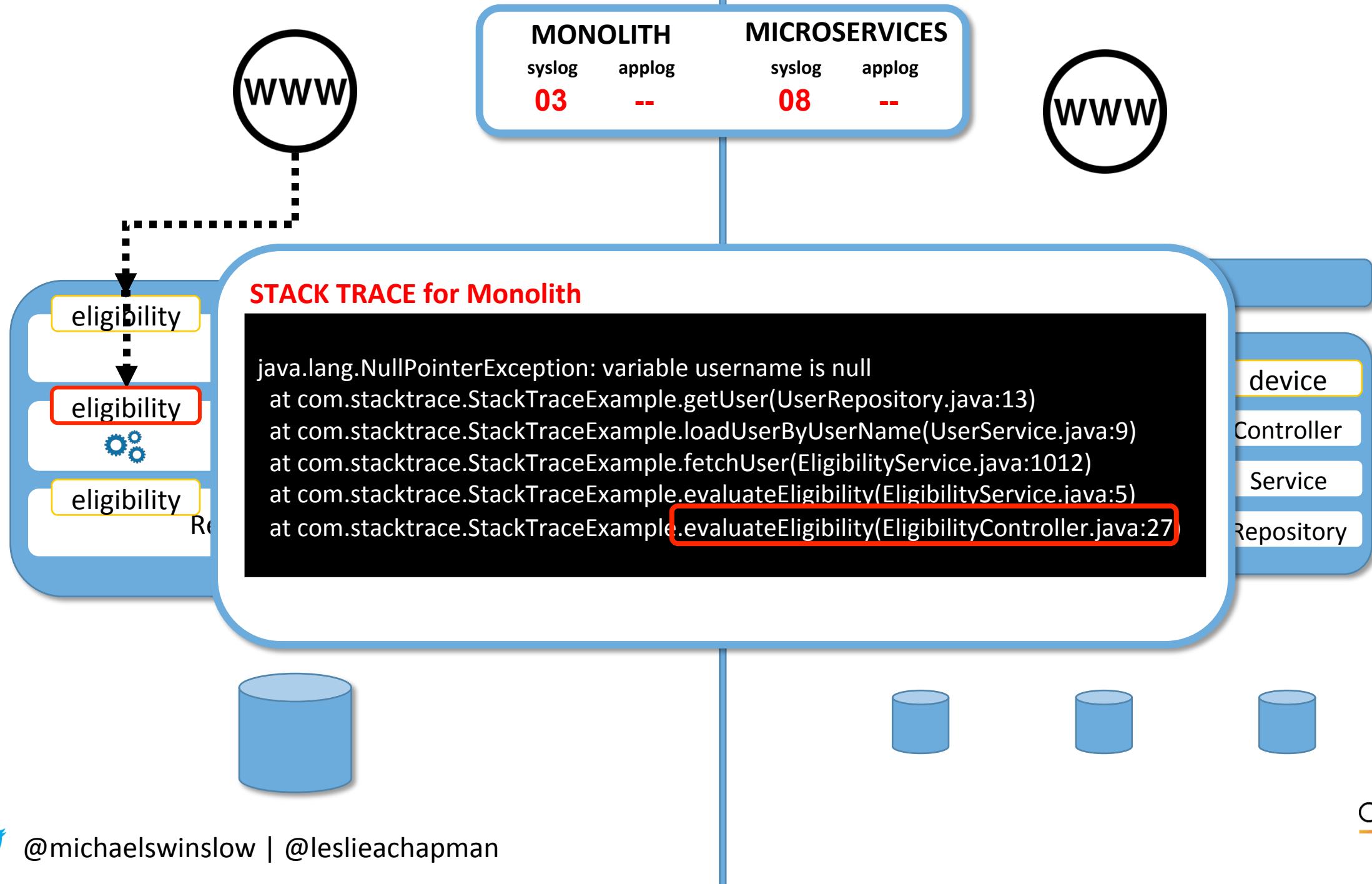




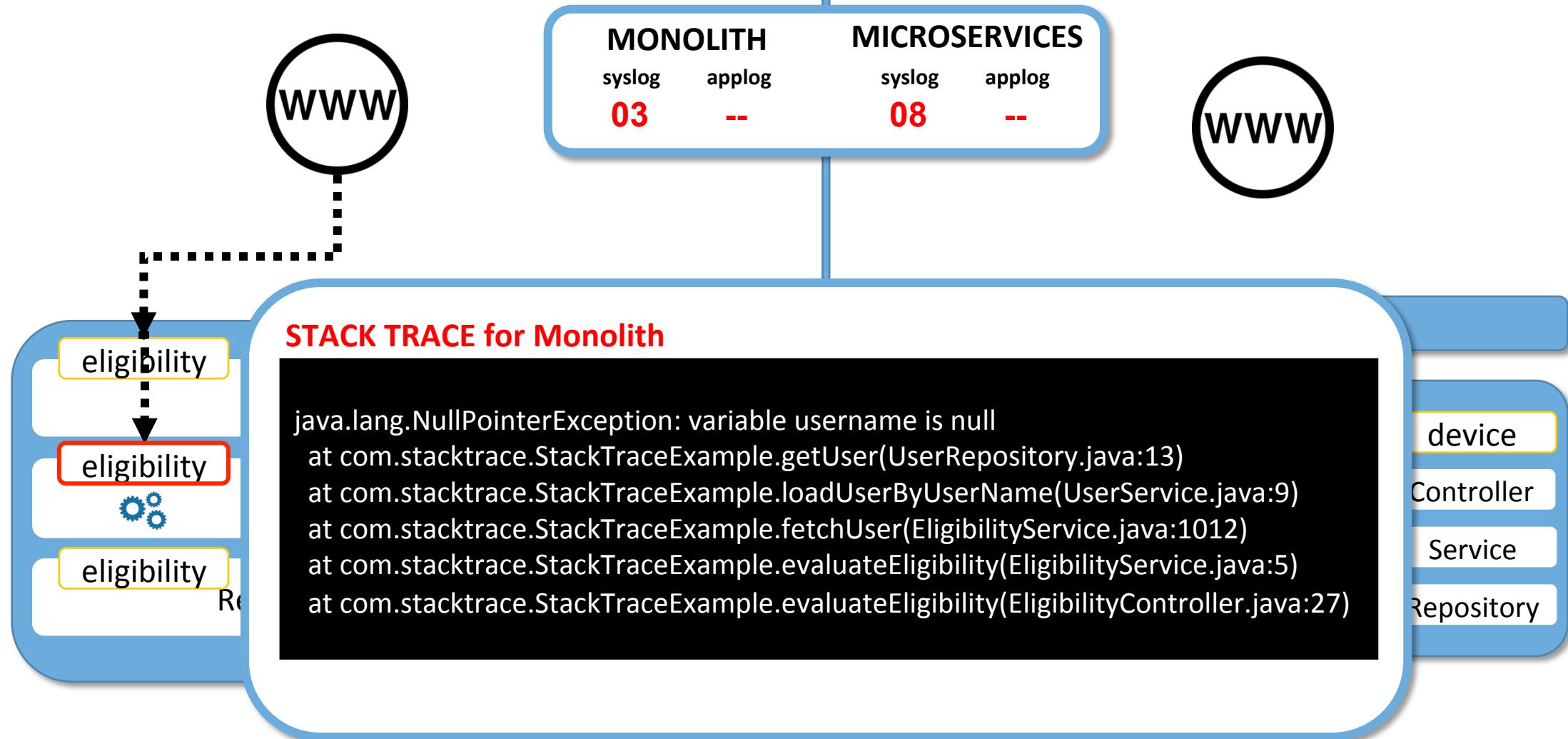
@michaelwinslow | @leslieachapman



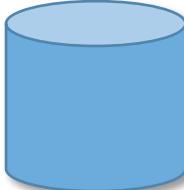
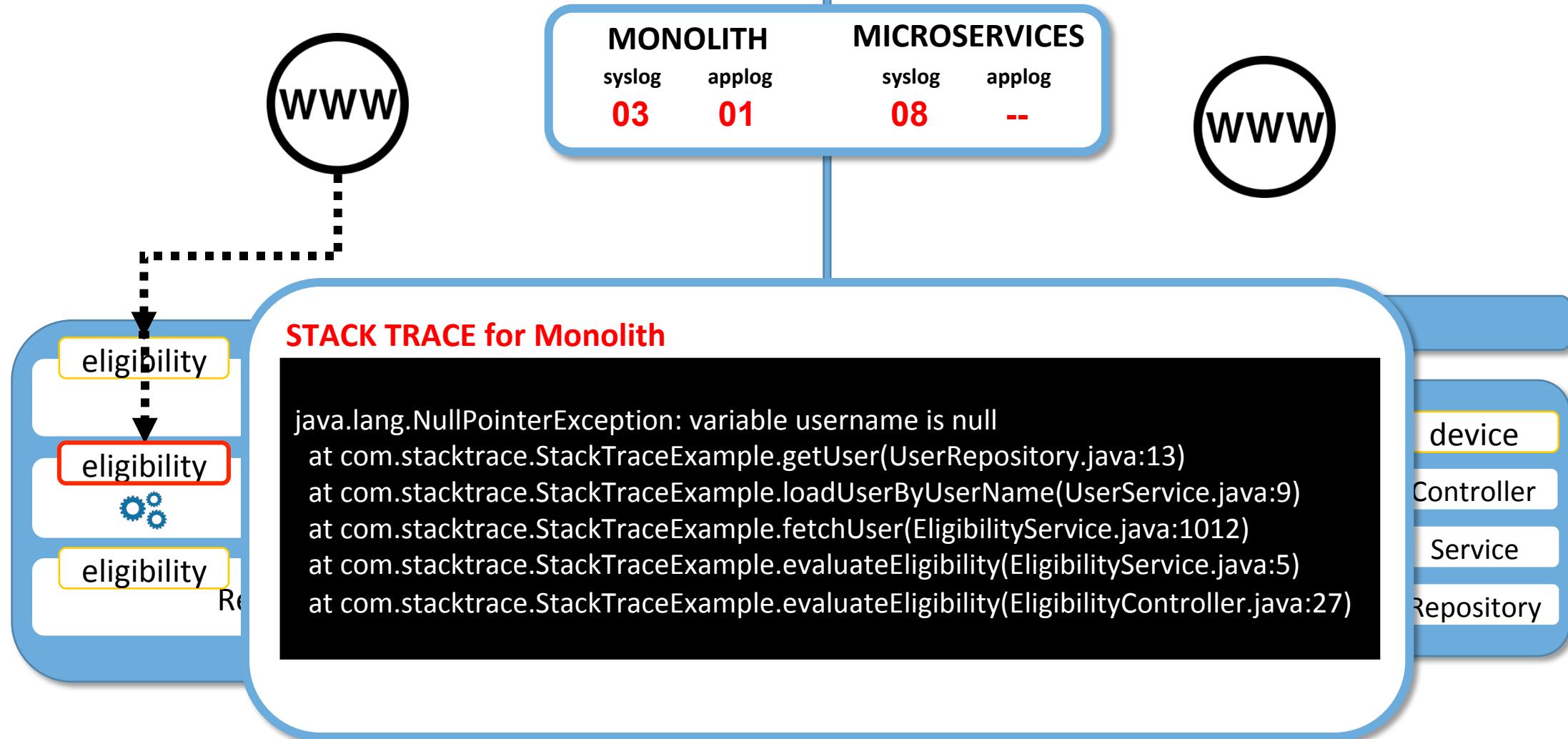


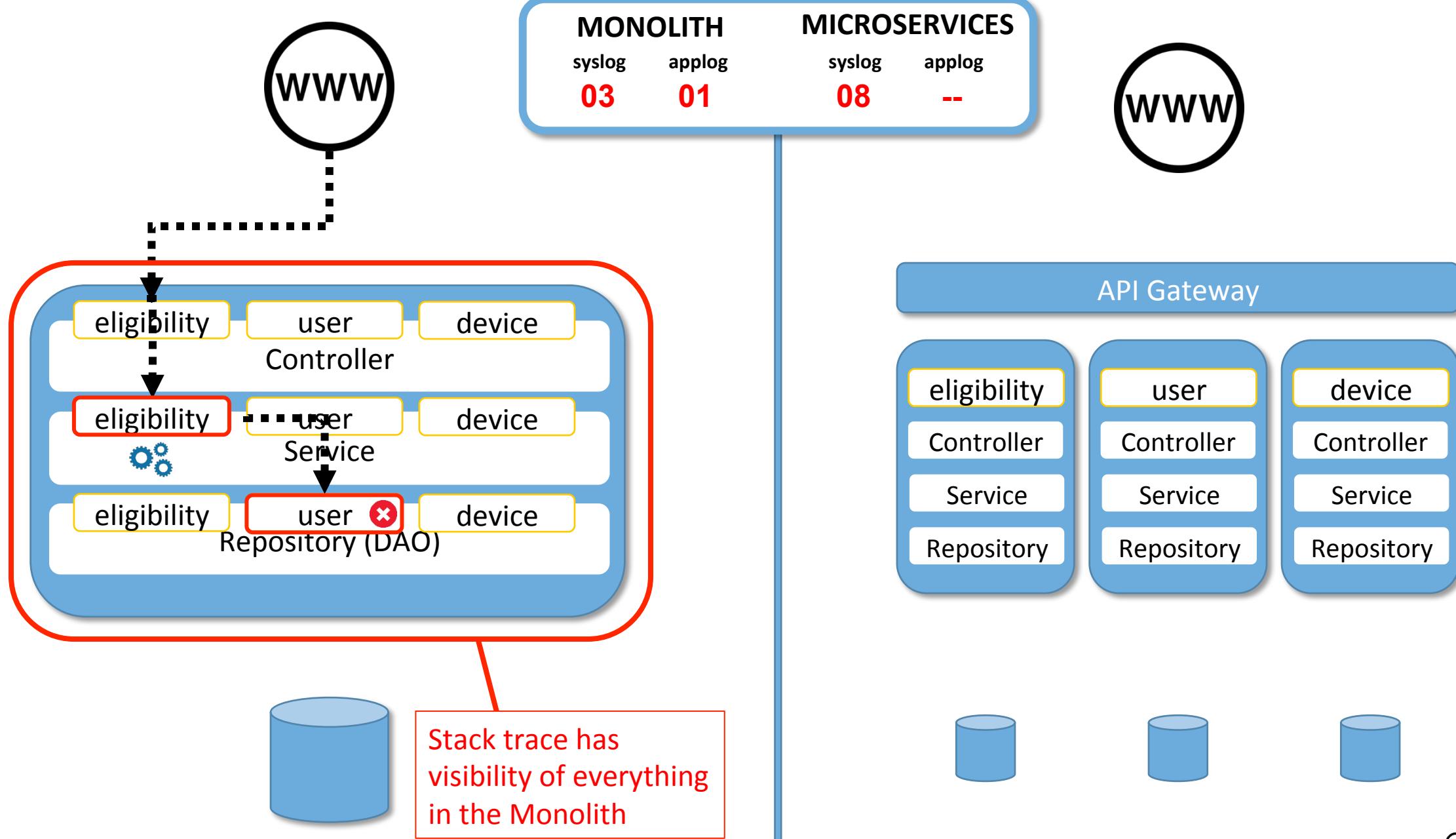


@michaelwinslow | @leslieachapman

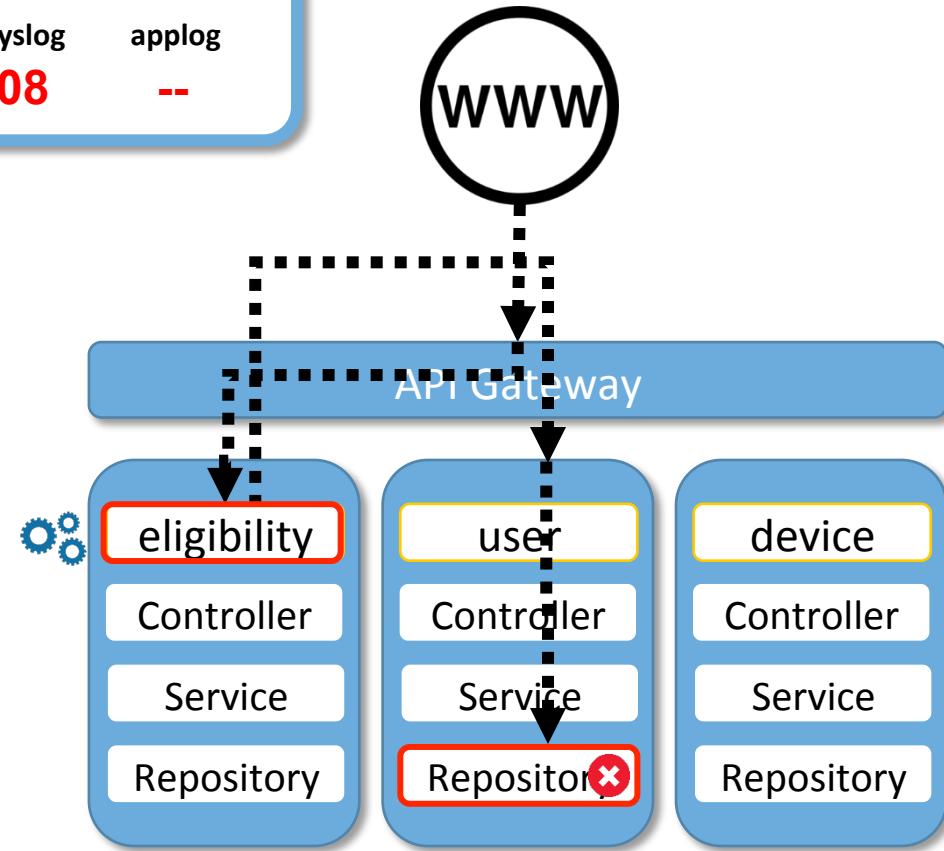
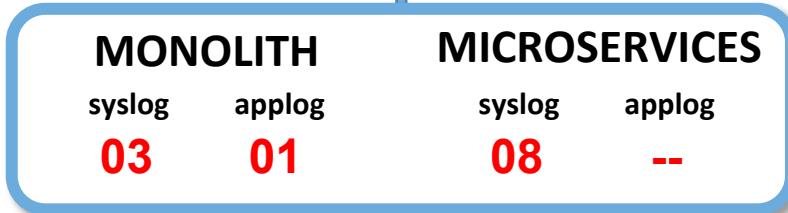
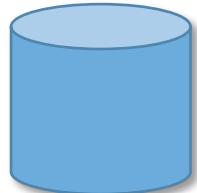
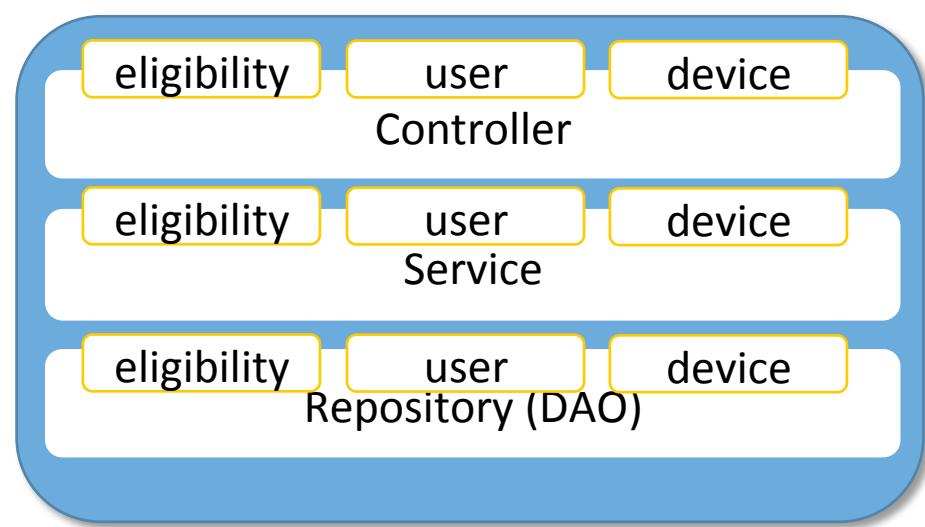


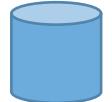
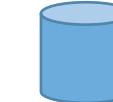
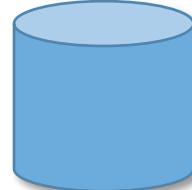
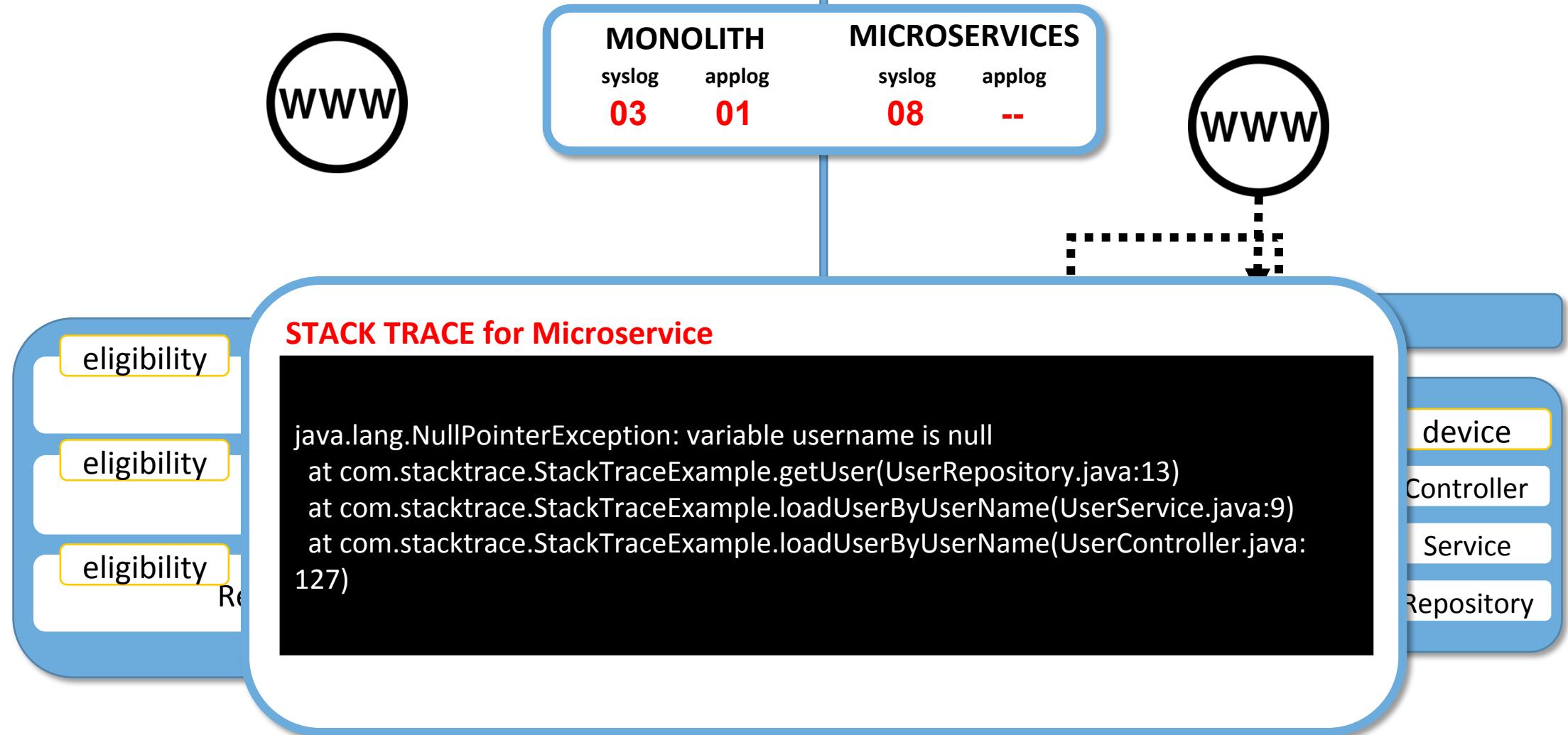
@michaelwinslow | @leslieachapman

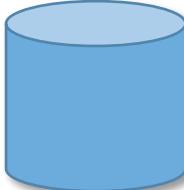
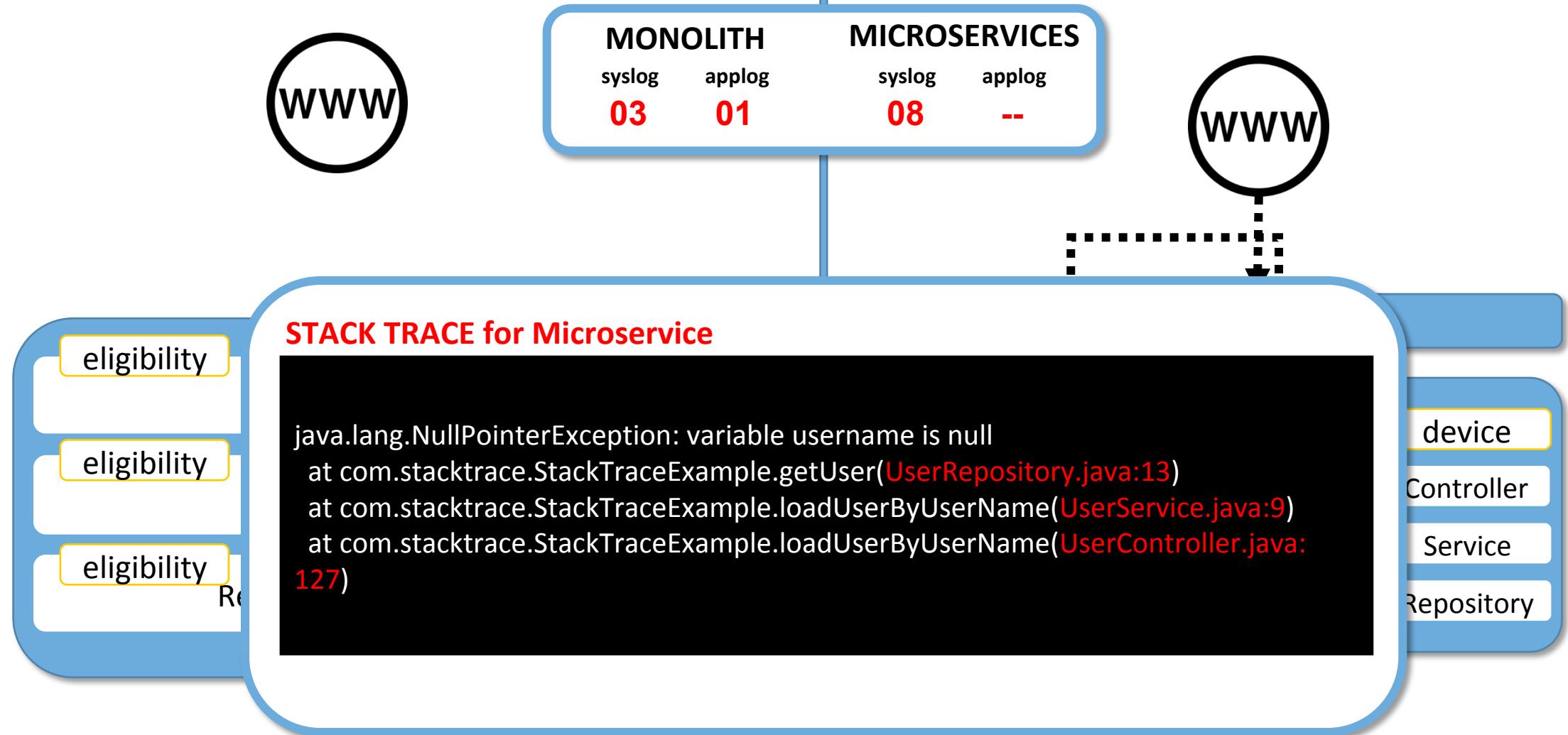


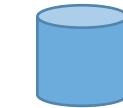
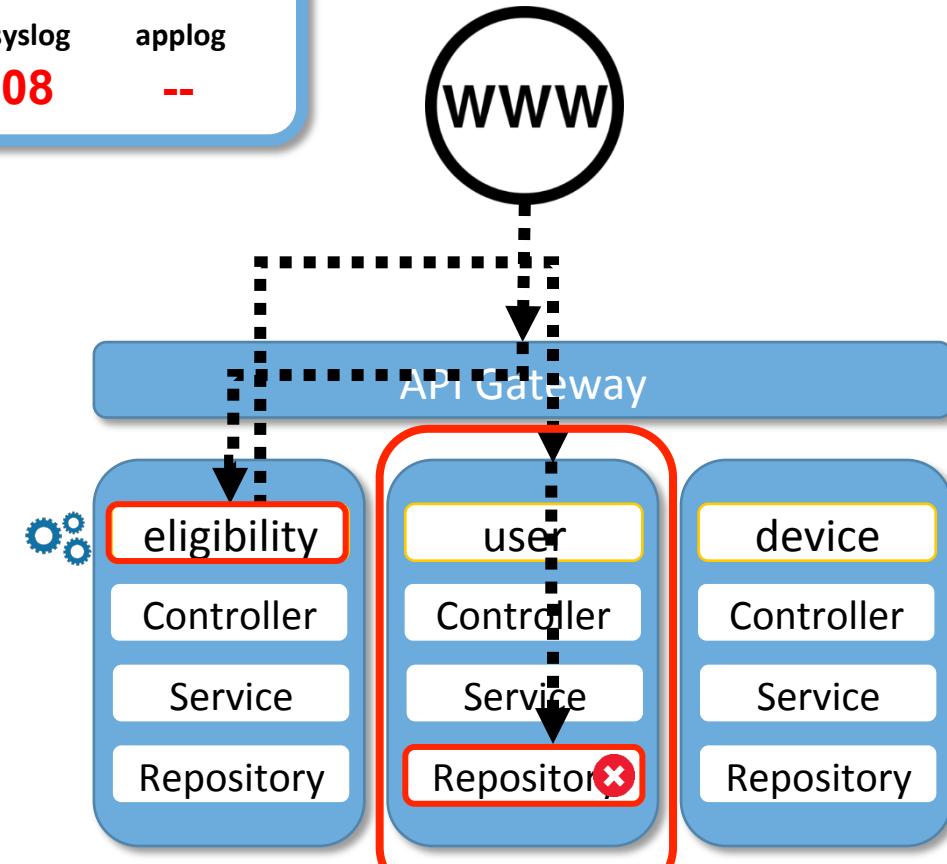
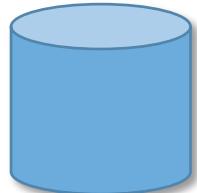
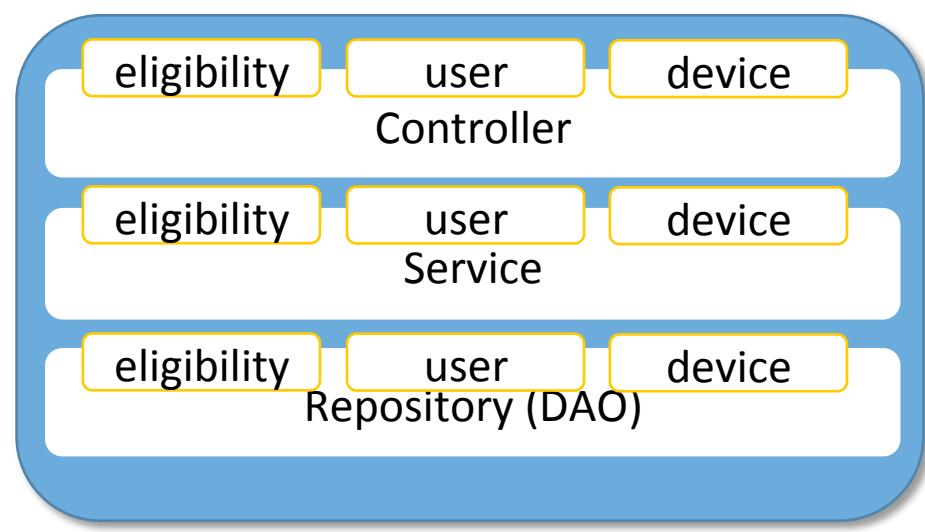


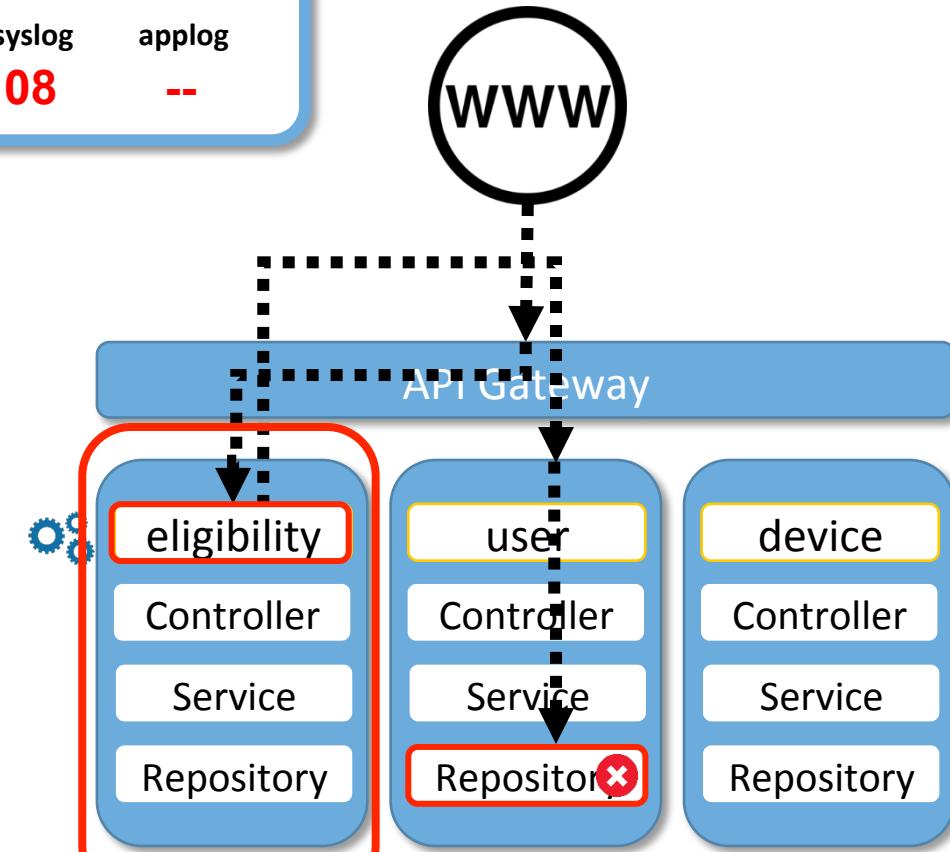
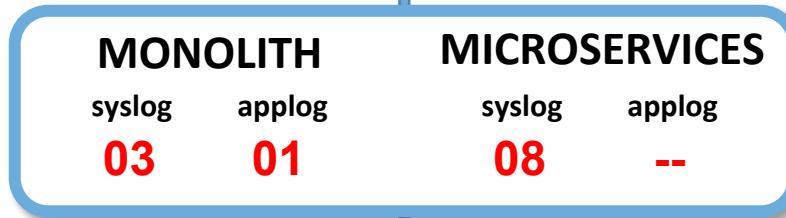
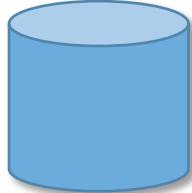
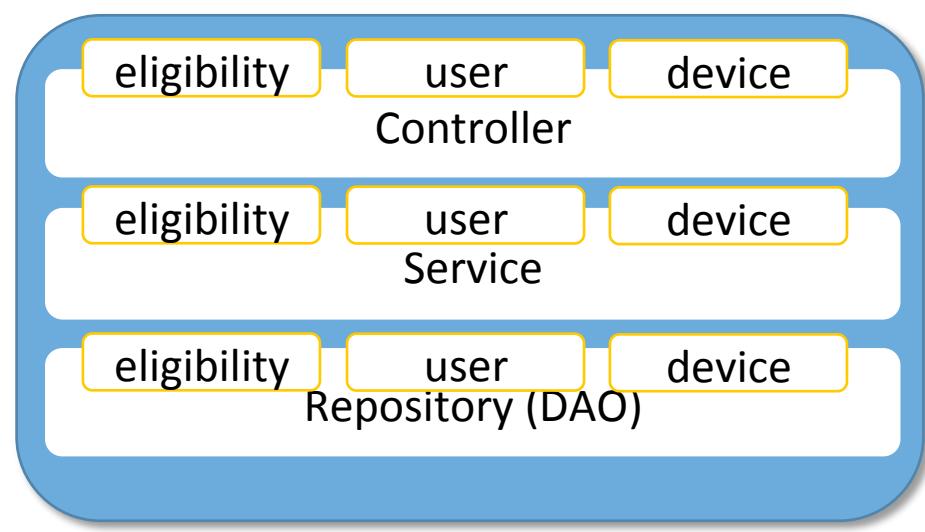
@michaelwinslow | @leslieachapman



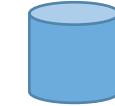








All context of  
where the request  
started is lost



# Distributed Tracing and the logs that come with it

Google  
[Dapper]

twitter  
[Zipkin]

COMCAST  
[Money]

[OpenTelemetry]



@michaelwinslow | @leslieachapman

COMCAST  
DevOps

*“We used to use **distributed tracing** to track transactions **across applications**. Now we need it to track transactions **within a single application.**”* – Catherine Rodriguez,  
Engineering Leader

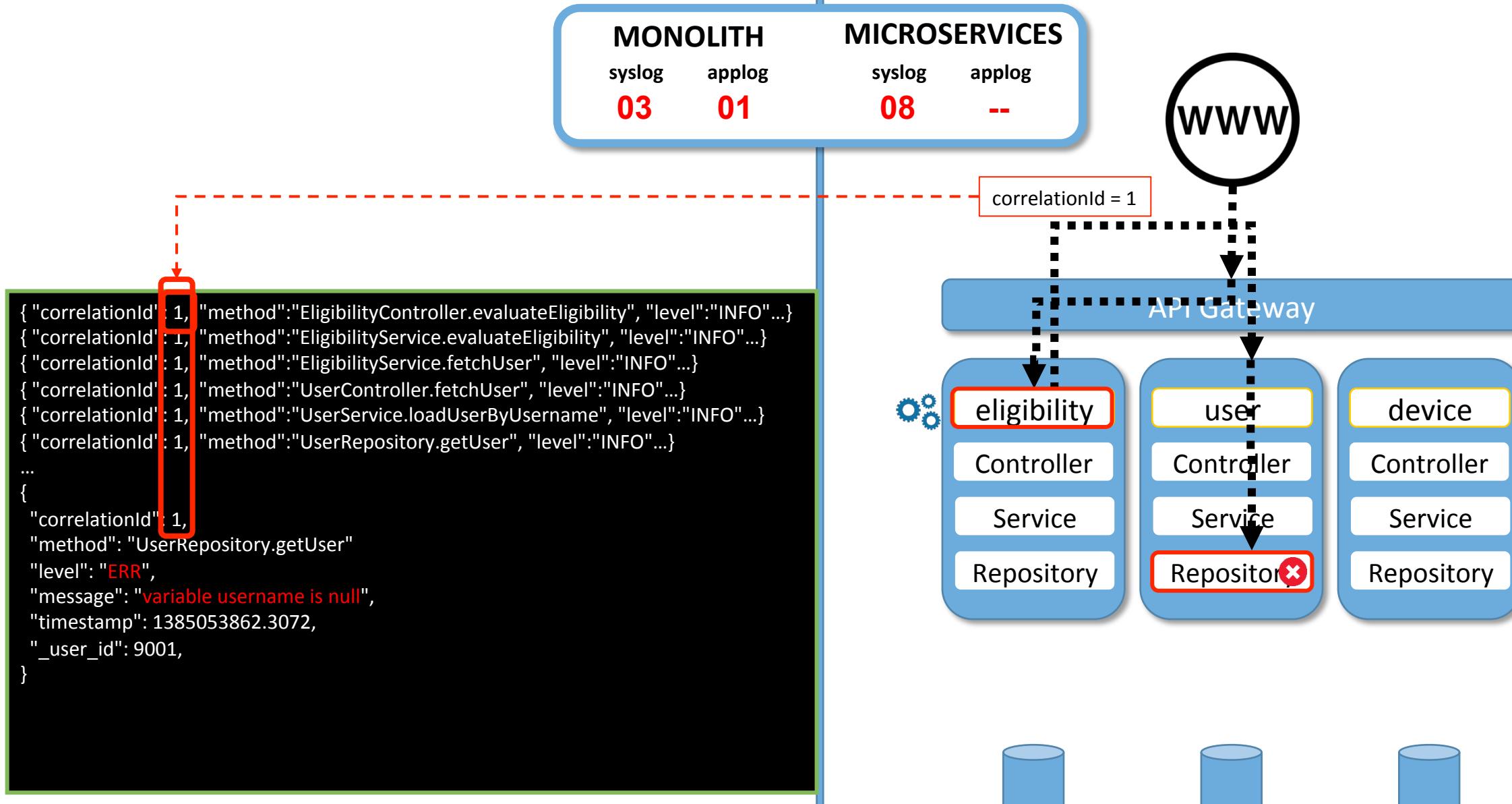


 @catherinerod27



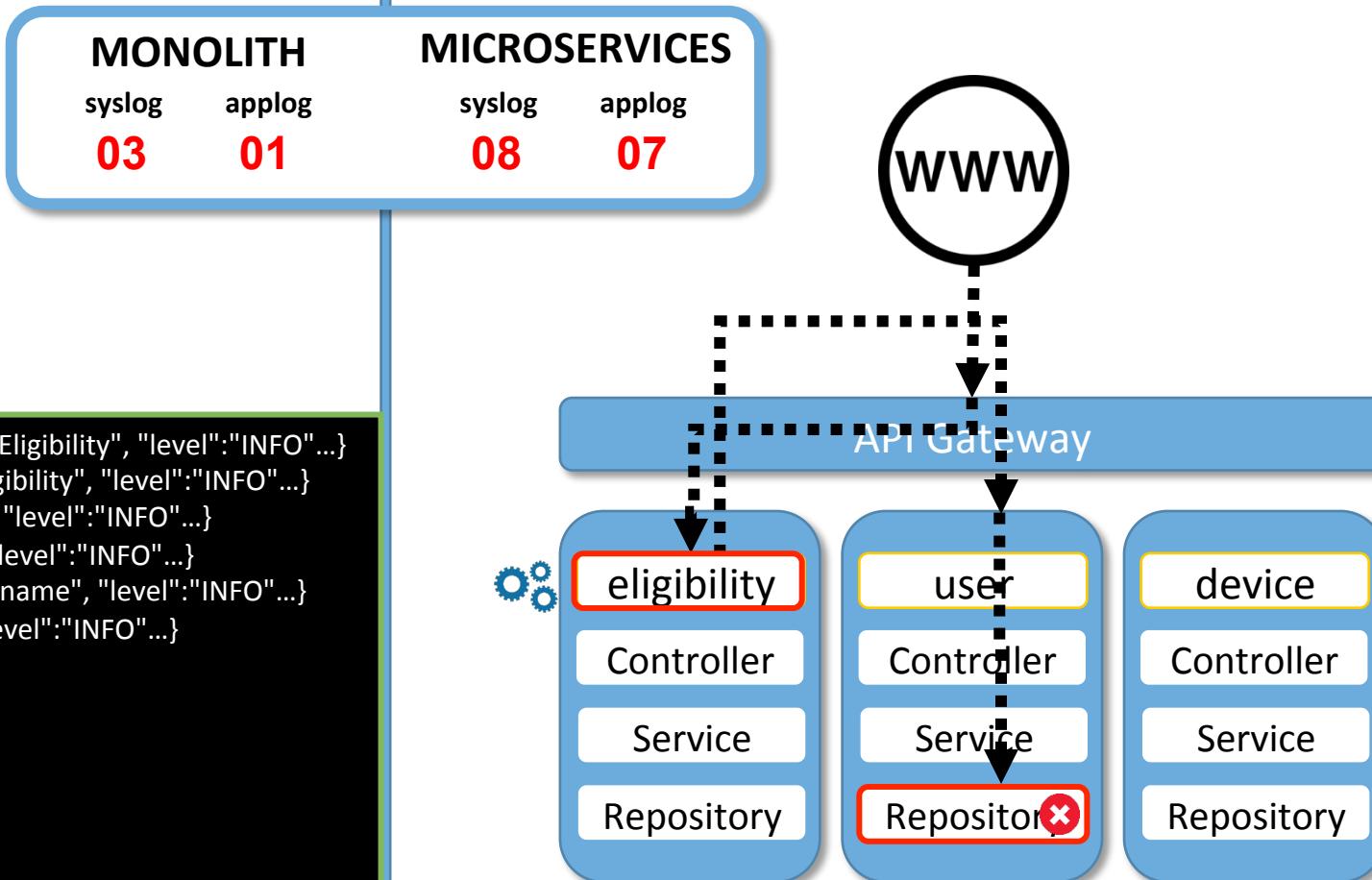
@michaelwinslow | @leslieachapman





@michaelwinslow | @leslieachapman

```
{ "correlationId": 1, "method":"EligibilityController.evaluateEligibility", "level":"INFO"...
{ "correlationId": 1, "method":"EligibilityService.evaluateEligibility", "level":"INFO"...
{ "correlationId": 1, "method":"EligibilityService.fetchUser", "level":"INFO"...
{ "correlationId": 1, "method":"UserController.fetchUser", "level":"INFO"...
{ "correlationId": 1, "method":"UserService.loadUserByUsername", "level":"INFO"...
{ "correlationId": 1, "method":"UserRepository.getUser", "level":"INFO"...
...
{
  "correlationId": 1,
  "method": "UserRepository.getUser"
  "level": "ERR",
  "message": "variable username is null",
  "timestamp": 1385053862.3072,
  "_user_id": 9001,
}
```



@michaelwinslow | @leslieachapman

```
{ "correlationId": 1, "metho  
...  
{  
  "correlationId": 1,  
  "method": "UserRepository  
  "level": "ERR",  
  "message": "variable usern  
  "timestamp": 1385053862.3  
  "_user_id": 9001,  
}  
}
```

## MONOLITH

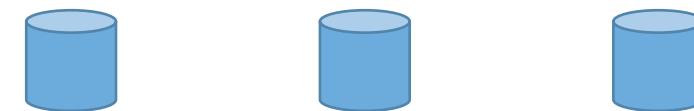
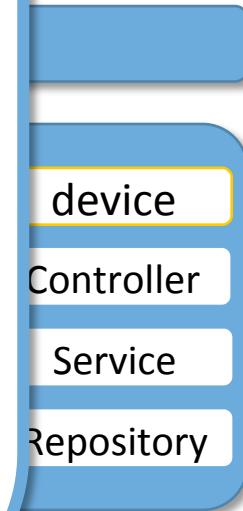
syslog applog

**03** **01**

## MICROSERVICES

syslog applog

**08** **07**



**What you did for ONE  
you now have to do for MANY**

**ASK YOUR TEAM**

*“How do we plan on rolling out these changes? Incrementally or all at once?”*



@michaelwinslow | @leslieachapman



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Microservice Readiness Checklist																								
Github Project	Artifact Type	Gerrit Setup	Jenkins Verify	Jenkins Build	Jenkins Auto-Deploy	Environments				Push Button Deploy				CodeBuild URL				Firewall Issues Resolved				Security Scan		
						INT	QA	STG	PROD	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	
Original-Fat-War	WAR	YES	YES	YES	YES	YES	YES	YES(!)	YES(!)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-1	WAR	YES	ISSUES	ISSUES	ISSUES	YES	YES	YES(!)	NO(!)	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-2	WAR	YES	NO	NO	NO	YES	NO	NO(!)	NO(!)	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-3	JAR	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-4	JAR	YES	N/A	N/A	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	NO	
Microservice-5	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-6	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-7	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-8	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-9	WAR	YES	NO	NO	NO	YES	NO	NO(!)	NO(!)	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-10	WAR	YES	NO	NO	NO	YES	NO	NO(!)	NO(!)	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	



@michaelwinslow | @leslieachapman

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
Microservice Readiness Checklist		Environments												Push Button Deploy				CodeBig2 URL				Firewall Issues Resolved			
Project	Artifact Type	Dev/Test Status	Builds Ready	Builds Built	Builds Auto Deployed	INT	QA	STG	PROD	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	Security Comp.	
Original-Fat-War	WAR	YES	YES	YES	YES	YES	YES	YES(!!)	YES(!!)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-1	WAR	YES	ISSUES	ISSUES	ISSUES	YES	YES	YES(!!)	NO(!!)	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-2	WAR	YES	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-3	JAR	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-4	JAR	YES	N/A	N/A	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	NO	
Microservice-5	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-6	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-7	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-8	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-9	WAR	YES	NO	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-10	WAR	YES	NO	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	NO	



@michaelwinslow | @leslieachapman

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Microservice Readiness Checklist																								
Github Project	Artifact Type	Gerrit Setup	Jenkins Verify	Jenkins Build	Jenkins Auto-Deploy	Environments				Push Button Deploy				CodeBump URL				Firewall Issues Resolved				Security Scan		
						INT	QA	STG	PROD	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	
Original-Fat-War	WAR	YES	YES	YES	YES	YES	YES	YES(!)	YES(!)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-1	WAR	YES	ISSUES	ISSUES	ISSUES	YES	YES	YES(!)	NO(!)	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-2	WAR	YES	NO	NO	NO	YES	NO	NO(!)	NO(!)	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-3	JAR	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-4	JAR	YES	N/A	N/A	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-5	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-6	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-7	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-8	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-9	WAR	YES	NO	NO	NO	YES	NO	NO(!)	NO(!)	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-10	WAR	YES	NO	NO	NO	YES	NO	NO(!)	NO(!)	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	



@michaelwinslow | @leslieachapman



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
Microservice Readiness Checklist		Environments												Push Button Deploy				CodeBump URL				Firewall Issues Resolved			
Github Project	Artifact Type	Gerrit Setup	Jenkins Verify	Jenkins Build	Jenkins Auto-Deploy	INT	QA	STG	PROD	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	Security Scan	
Original-Fat-War	WAR	YES	YES	YES	YES	YES	YES	YES(!!)	YES(!!)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-1	WAR	YES	ISSUES	ISSUES	ISSUES	YES	YES	YES(!!)	NO(!!)	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-2	WAR	YES	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-3	JAR	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-4	JAR	YES	N/A	N/A	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-5	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-6	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-7	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-8	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-9	WAR	YES	NO	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-10	WAR	YES	NO	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	NO	

code review + CI  
(with automated testing)

CD to all environments  
(with automated testing)

firew

security scans



@michaelwinslow | @leslieachapman

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Microservice Readiness Checklist																								
Github Project	Artifact Type	Gerrit Setup	Jenkins Verify	Jenkins Build	Jenkins Auto-Deploy	Environments				Push Button Deploy				CodeBig2 URL				Firewall Issues Resolved				Security Scan		
						INT	QA	STG	PROD	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	DEV	INT	QA	STG	PROD	
Original-Fat-War	WAR	YES	YES	YES	YES	YES	YES	YES(!!)	YES(!!)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-1	WAR	YES	ISSUES	ISSUES	ISSUES	YES	YES	YES(!!)	NO(!!)	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	
Microservice-2	WAR	YES	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-3	JAR	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-4	JAR	YES	N/A	N/A	YES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Microservice-5	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-6	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-7	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-8	WAR	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-9	WAR	YES	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	
Microservice-10	WAR	YES	NO	NO	NO	YES	NO	NO(!!)	NO(!!)	NO	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	NO	

## Suggestions:

1. Do not create all of your microservices at once ( look up the Strangler Pattern )
2. Monitor the usage to find your most utilized endpoints and use that to prioritize your migration
3. Have a template for creating new microservices **and USE IT!**



@michaelwinslow | @leslieachapman

# Change Management Team wants their release notes

**ASK YOUR TEAM**

*“How drastically will this change our communication structure?”*



@michaelwinslow | @leslieachapman





#stc19

## Automated Release Notes

- Michael Winslow
- Core Application - Comcast
- I've automated myself out of 2 jobs before



 @michaelwinslow

 michaelwinslow



<https://speakerdeck.com/michaelwinslow/automated-release-notes-stc19>



@michaelwinslow | @leslieachapman



## Projects:

Many of the links in this report require an active Jenkins session. Click  to login

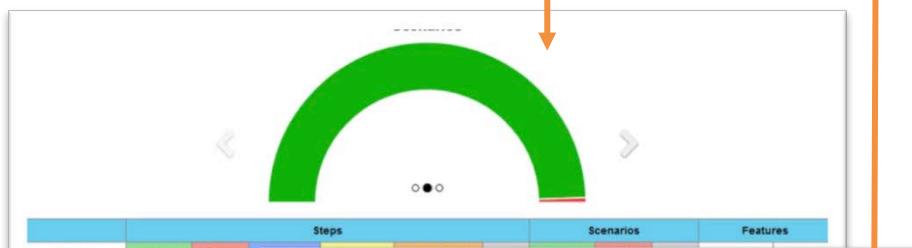
MBO Line Service	Tests 	#2.3.25.1 	VERACODE 	
MBO Prospect Service	Tests 	#2.3.37 	97 	#2.3.25 

6.5.1-autogen

anview 

anview ID is the user Java on Order Fulfill (MSP-0005) [Administrator]  
Primary User's email is part of Jenkins (MSP-0005) [Administrator]  
Same, do not share the same global Jenkins' administrator (MSP-0005) [Owner]  
Confirmation (MSP-ACK-01) to all account email addresses (MSP-0005) [Owner]

Project	Results	Time	Owner
MBO Line Service	Results	1.0.08	JSP
MBO Prospect Service	Results	1.0.08	JSP
VERACODE	Results	1.0.08	JSP
ch2	Results	1.0.08	JSP



Steps      Scenarios      Features

Feature	Passed	Failed
MSP Services	1271	1
1	1271	1
	99.53%	0.08%

**Jenkins**

jenkins > MBO - Microservices > MBO 2d Deploy to Production

- Back to Dashboard
- Status
- Changes
- Workspace
- Build with Parameters

**Project MBO 2d Deploy to Production**

This build requires parameters:

APPLICATION_NAME	lineservice
MAJOR_VERSION_NUMBER	2.3
APP_DC	ch2
BUILD_NUMBER	25.1
CUSTOM_CONFIG_LABEL	NO

**Build History**

find
#45 April 8, 2017 4:13 AM
#44 April 8, 2017 4:08 AM
#43 April 3, 2017 7:35 PM

**Build**

**VERACODE**

Veracode Detailed Report  
Application Security Report  
As of 22 May 2017

Prepared for: Comcast Corporation  
Prepared on: May 22, 2017  
Application: MBO - Microservices - MbosLine  
Business Criticality: Not Specified (BCI: Medium)  
Required Analysis: Static  
Tools of Analysis Conducted: Scope of Static Scan  
1 of 1 Modules Analyzed

**Jenkins**

jenkins > MBO - Microservices > MBO 2d Deploy to Production

- Back to Dashboard
- Status
- Changes
- Workspace
- Build with Parameters

**Project MBO 2d Deploy to Production**

This build requires parameters:

APPLICATION_NAME	lineservice
MAJOR_VERSION_NUMBER	2.3
APP_DC	ch2
BUILD_NUMBER	25
CUSTOM_CONFIG_LABEL	NO

**Build History**

find
#45 April 8, 2017 4:13 AM
#44 April 8, 2017 4:08 AM
#43 April 3, 2017 7:35 PM

**Method of Procedure (MOP)**

Staging Deployment (Date: Unknown)

Task	Assigned
Pre-Deployment	
<input type="checkbox"/> Approved in CAR meeting?	Owner: Michael Winslow
<input type="checkbox"/> Date and time set for deployment?	Owner: Michael Winslow
<input type="checkbox"/> Member of QA team assigned for pre-deployment verification?	Verifier: Michael Winslow
<input type="checkbox"/> Member(s) from DevOps team assigned to execute deployment?	Reviewer: Michael Winslow

**Project MBO 2d Deploy to Production**

This build requires parameters:

APPLICATION_NAME	lineservice
MAJOR_VERSION_NUMBER	2.3
APP_DC	ch2
BUILD_NUMBER	25
CUSTOM_CONFIG_LABEL	NO

**Build**



@michaelwinslow | @leslieachapman

# **Version Control and Compatibility**

can be a nightmare

**ASK YOUR TEAM**

*“How do we identify complete shippable versions?”*



@michaelwinslow | @leslieachapman



## Projects: Release 1.5.7

IMPORTANT: Many of the links in this report require an active Jenkins session. Click  to

				
MBO Account Service	Tests 	#2.3.17 		
MBO Audit Service	Tests 	#2.3.4 		
MBO CallAgent Service	Tests 	#2.3.10 		
MBO Device Diagnostics	Tests 	#2.3.7 		
MBO Device Service	Tests 	#2.3.57 		
MBO Device Warranty	Tests 	#2.3.10 		
MBO Gateway Service	N/A (3rd Party Library)	#2.3.10 	N/A (3rd Party Library)	#2.3.5 
MBO Line Service	Tests 	#2.3.46 	96 	#2.3.25.1 
MBO Prospect Service	Tests 	#2.3.52 	98 	#2.3.37 
MBO Registry Service	N/A (3rd Party Library)	#2.3.8 	N/A (3rd Party Library)	#2.3.3 
MBO Config	N/A (Config File)	#1.5.13	N/A (Config File)	#1.5.11

A RELEASE version now consists of a configuration of many microservices. Each microservice version increments independently from the RELEASE and other microservices.



@michaelwinslow | @leslieachapman

# Burnout is real

**ASK YOUR TEAM**

*“Is our team prepared to take on this task, given our existing workload?”*



@michaelwinslow | @leslieachapman



Note that the **World Health Organization** now classifies burn-out as an official medical disorder.

**World Health Organization (2019).** QD85 – Burn-Out - <https://icd.who.int/browse11/l-m/en#/http://id.who.int/icd/entity/129180281>



@michaelwinslow | @leslieachapman

# Recap with DOs and DON'Ts:

**DO** have a reason to move to microservices (Resiliency, Cost-Optimization, etc.)

**DON'T** practice RDD (Resume-Driven Development)

**DO** have a plan for monitoring, telemetry and distributed tracing

**DON'T** count on traditional logging techniques (Stack Trace)

**DO** peel off one microservice at a time (Strangler Application pattern)

**DON'T** refactor everything at once

**DO** understand that all teams are impacted (QA, Release Management, Security, etc.)

**DON'T** say “we're doing microservices” **#YOLO**

**DO** staff appropriately and monitor employee health

**DON'T** ignore the human impact of your decision

# Recap with DOs and DON'Ts:

**DO** encourage DevOps teams to pursue microservices if there are benefits  
**DON'T** underestimate the time, effort and money involved



@michaelwinslow | @leslieachapman



# Thank You!



- **Leslie Chapman**
- **Distinguished Engineer - Comcast**
- **I LOVE CATS**



- **Michael Winslow**
- **Director - Comcast**
- **I still code for management tasks**

 @leslieachapman

 /lesliechapman

 @michaelwinslow

 /michaelwinslow

## Helpful Links

1. Netflix Zuul - <https://github.com/Netflix/zuul>
2. Netflix Eureka - <https://github.com/Netflix/eureka>
3. Netflix Ribbon - <https://github.com/Netflix/ribbon>
4. Netflix Hystrix - <https://github.com/Netflix/hystrix>
5. Netflix Vizceral - <https://github.com/Netflix/vizceral>
6. Google Dapper – <https://ai.google/research/pubs/pub36356>
7. Comcast Money - <https://github.com/Comcast/money>
8. Twitter Zipkin - <https://zipkin.apache.org/>



@michaelwinslow | @leslieachapman

COMCAST  
DevOps