

# Harnessing Domain Driven Design

**Gayathri Thiyagarajan**

Lead Software Engineer Capgemini

**Andrew Harmel**

Law Principal Software Engineer Capgemini



Capgemini  
Week of  
Innovation  
Networks

# Harnessing Domain Driven Design for Distributed Systems



Gayathri Thiyagarajan &  
Andrew Harmel-Law,

Capgemini UK A dark blue spade-shaped logo icon.

<http://capgemini.github.io>



“Who are we?



# Who are we?



# “Microservice”?



*A Microservice Definition*

*Loosely coupled service oriented  
architecture with bounded contexts*



# “Microservice”? (cont.)

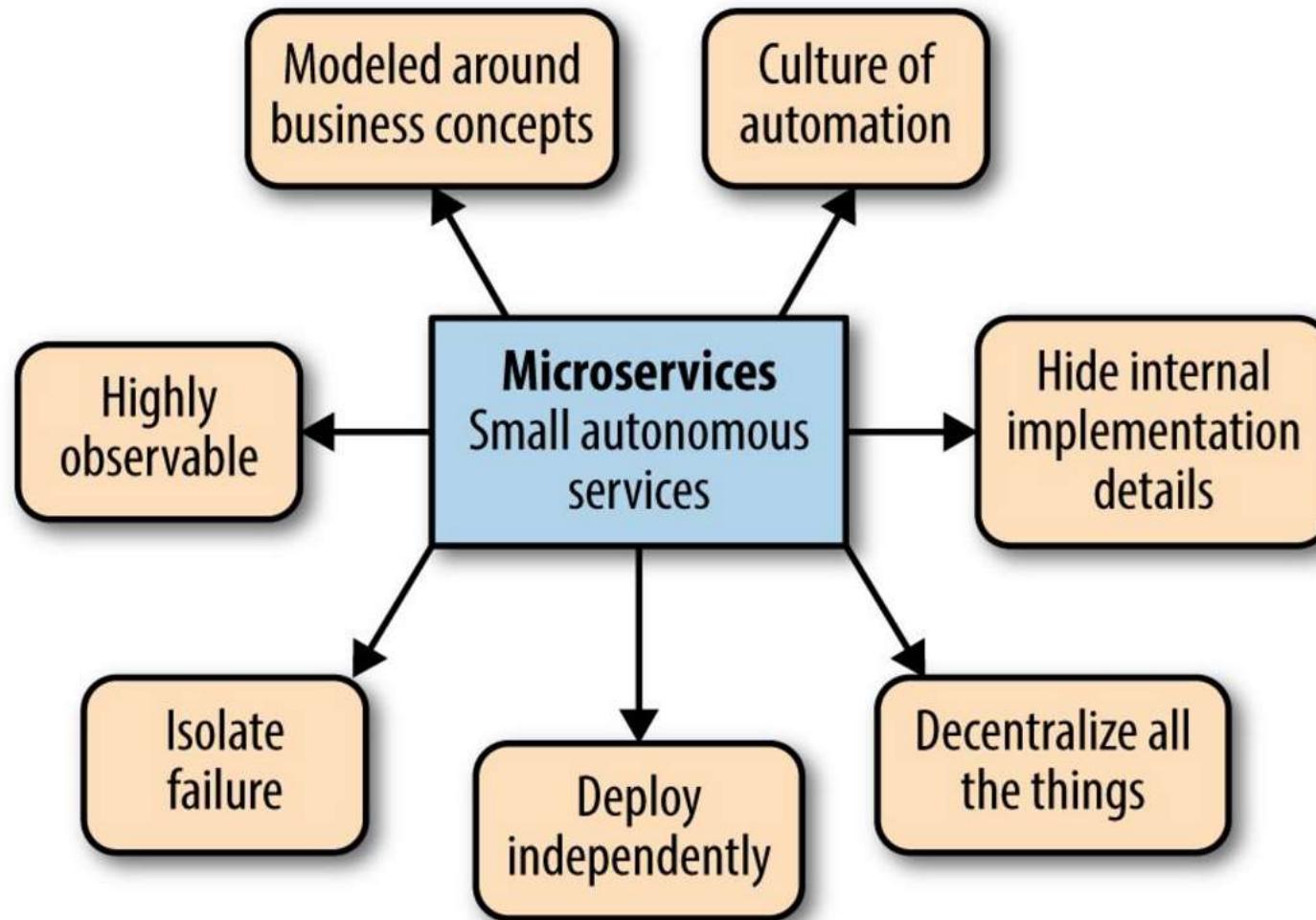


Figure 12-1. Principles of microservices

# “Microservice”? (cont.)

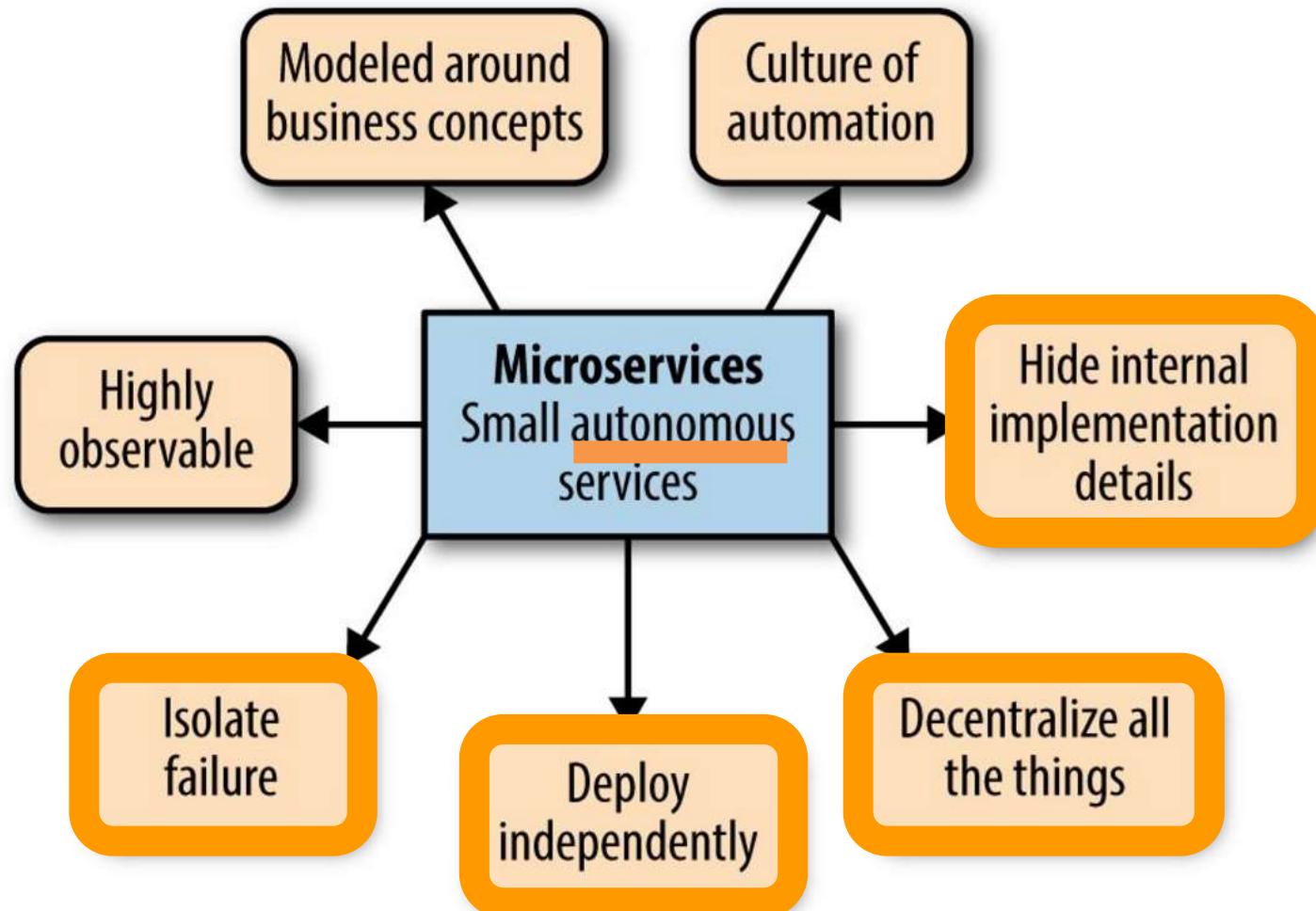


Figure 12-1. Principles of microservices

# “Microservice”? (cont.)

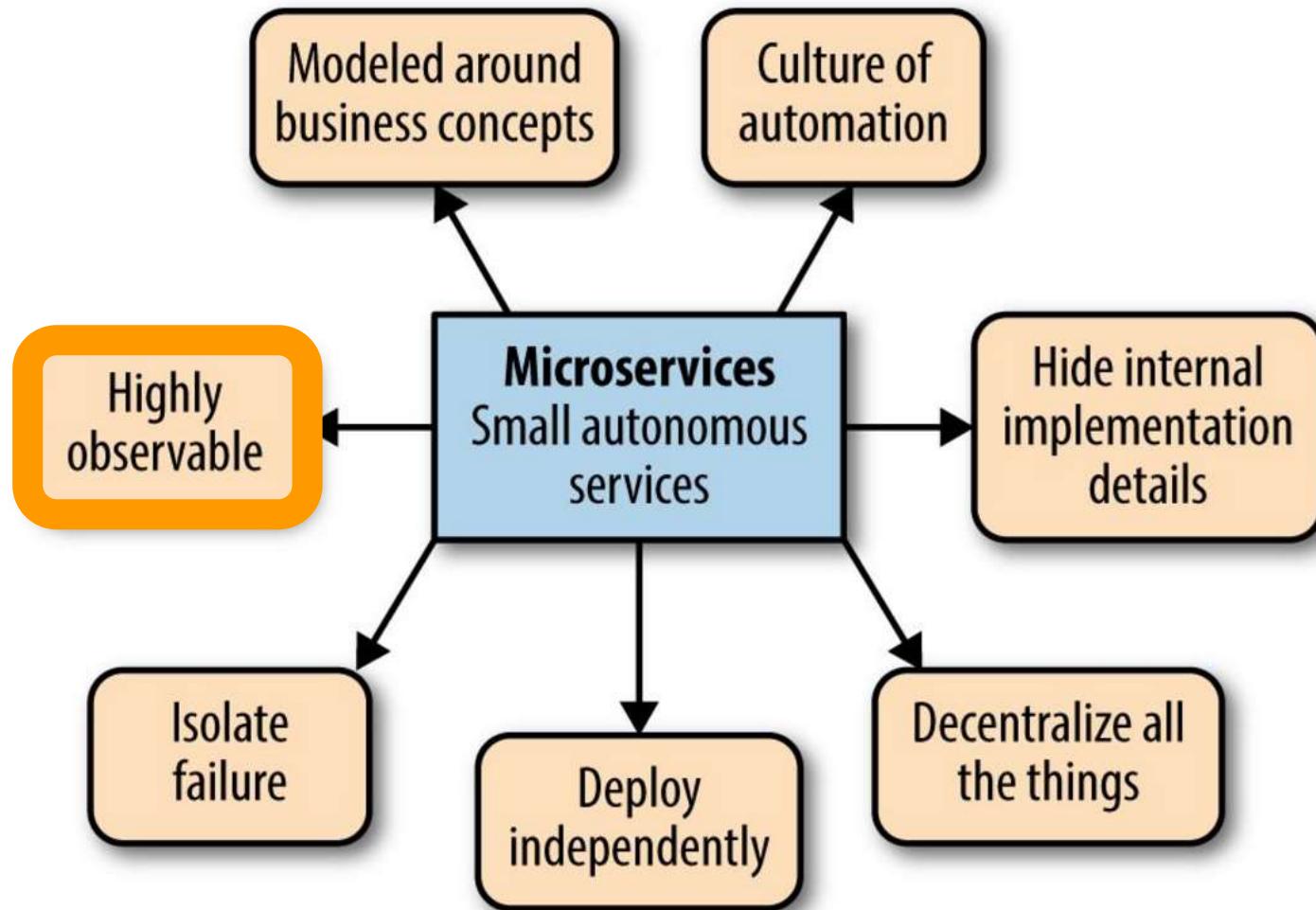


Figure 12-1. Principles of microservices

# “Microservice”? (cont.)

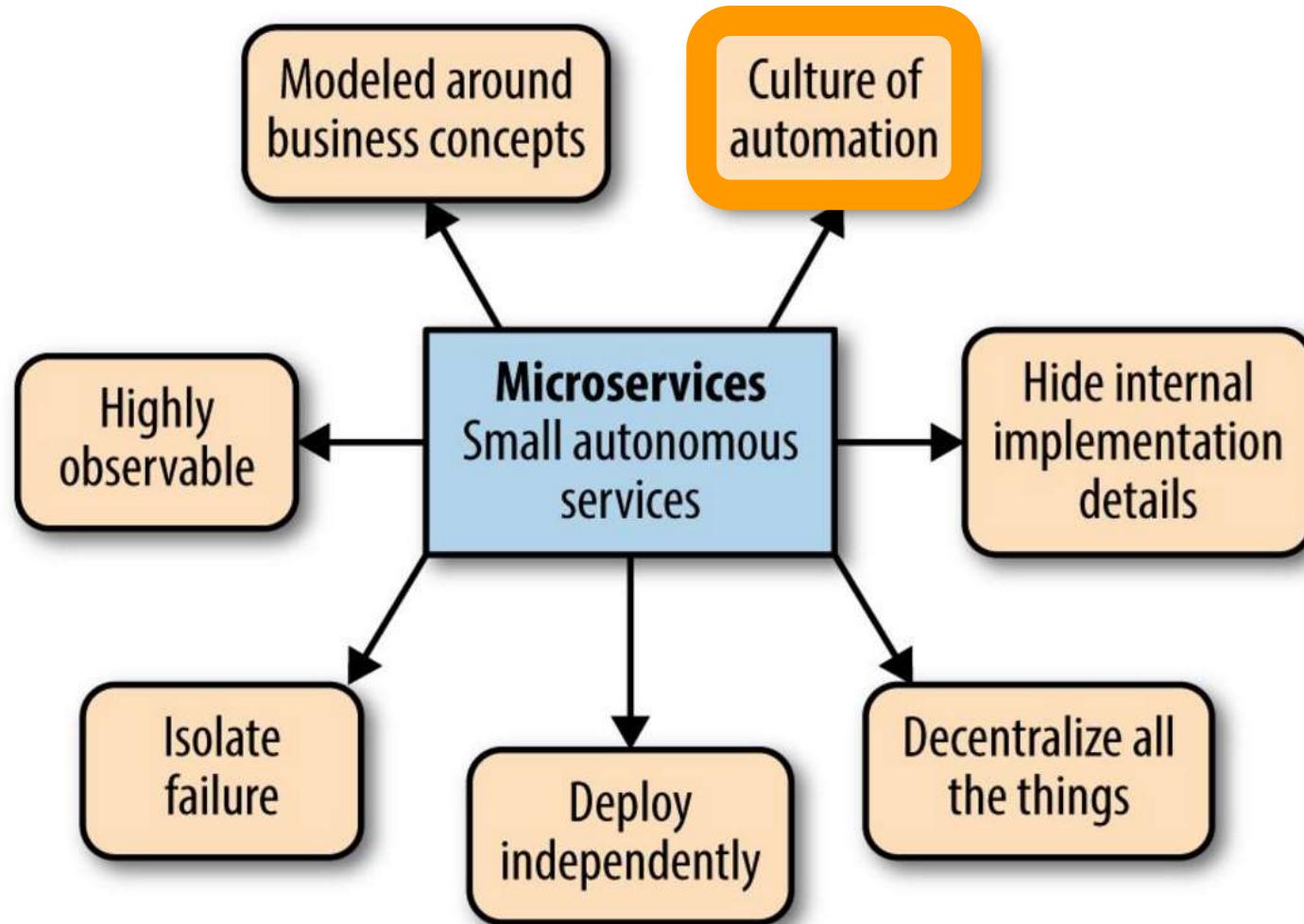


Figure 12-1. Principles of microservices

# “Microservice”? (cont.)

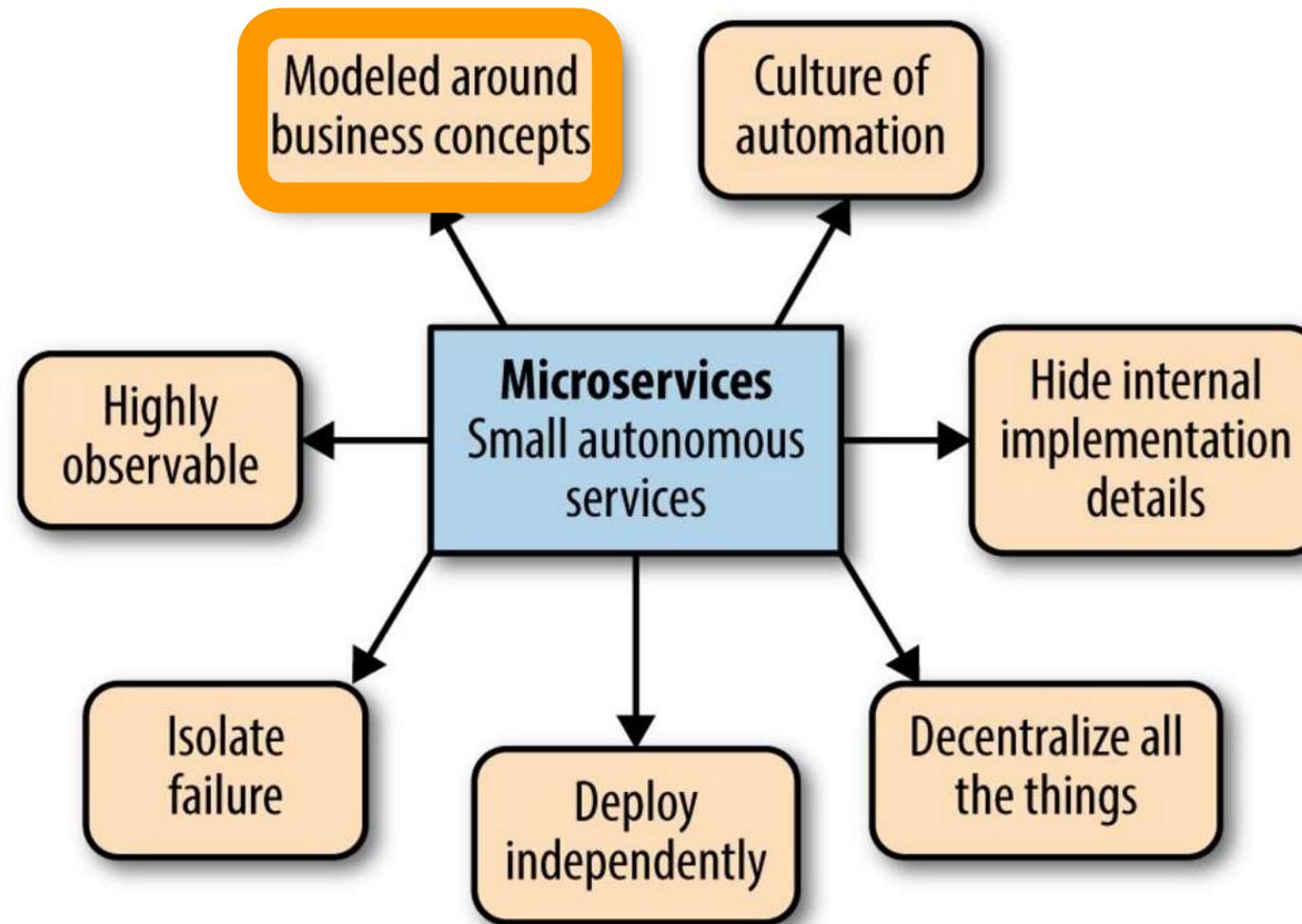


Figure 12-1. Principles of microservices

# Microservices ROCK!

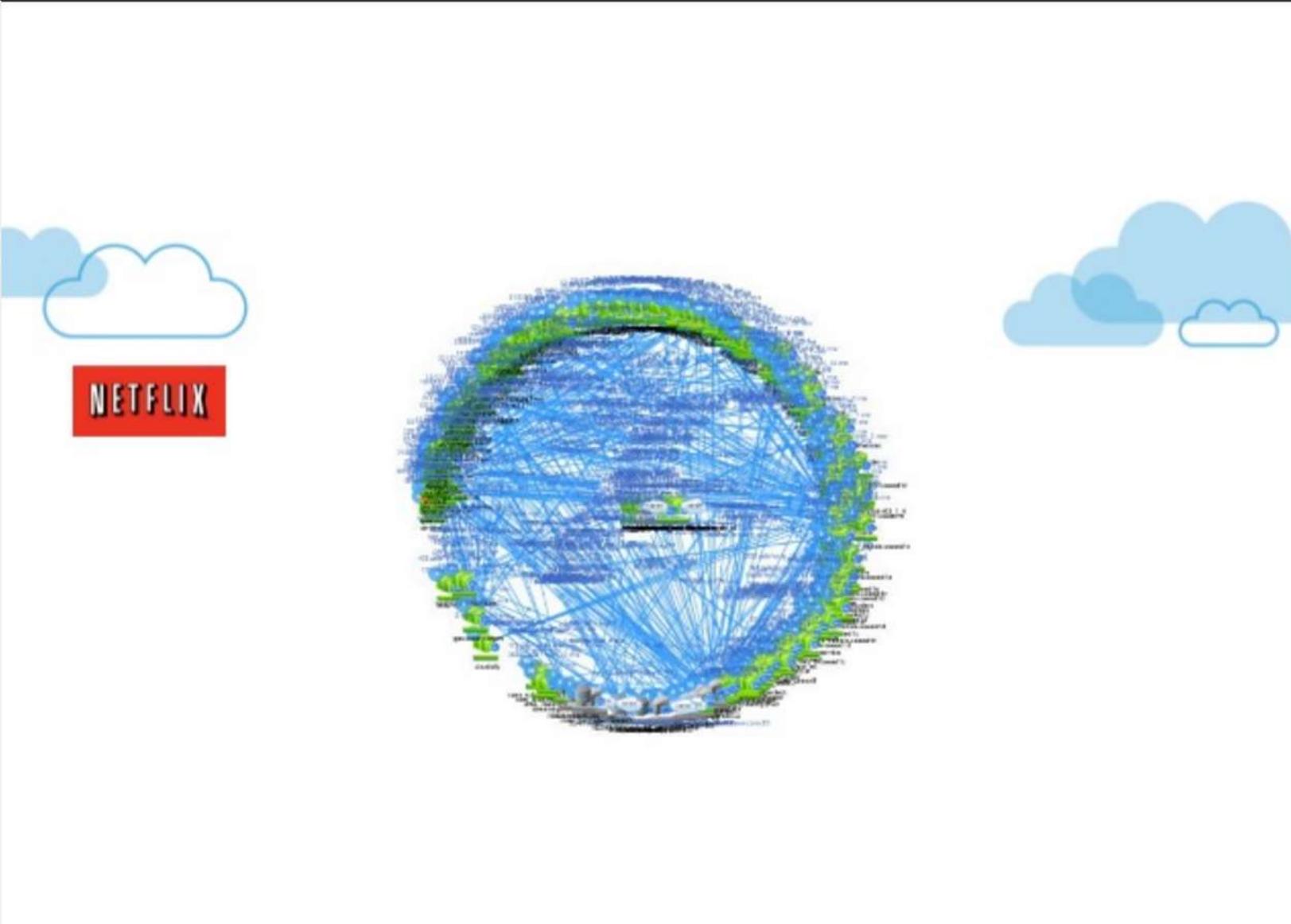


cc: RaeAllen - <https://www.flickr.com/photos/30628871@N00>

#harnessing\_ddd

@gaythu\_rajan / @al94781

# Microservices are HARD



# Boundaries are HARDEST

“Getting service boundaries wrong can result in having to make lots of changes in service-service collaboration. An expensive operation.”

from Sam Newman, “Building Microservices”  
(section: When shouldn’t you use Microservices)

# Microservices LOVE DDD

“Despite the hype  
[microservices] probably giv[e]  
us the best environment we  
have ever had for doing  
Domain-Driven Design”

(Interview with Eric Evans at 2015 DDD Exchange)

cc: {AndreaRenee} - <https://www.flickr.com/photos/30282864@N02>

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to be sure you have adequate disk space. If a driver is identified in the Stop message, disable the driver or check with the manufacturer for driver updates. Try changing video adapters.

Check with your hardware vendor for any BIOS updates. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:

\*\*\* STOP: 0x0000008E (0xC0000005,0x00690076,0xA5354B10,0x00000000)

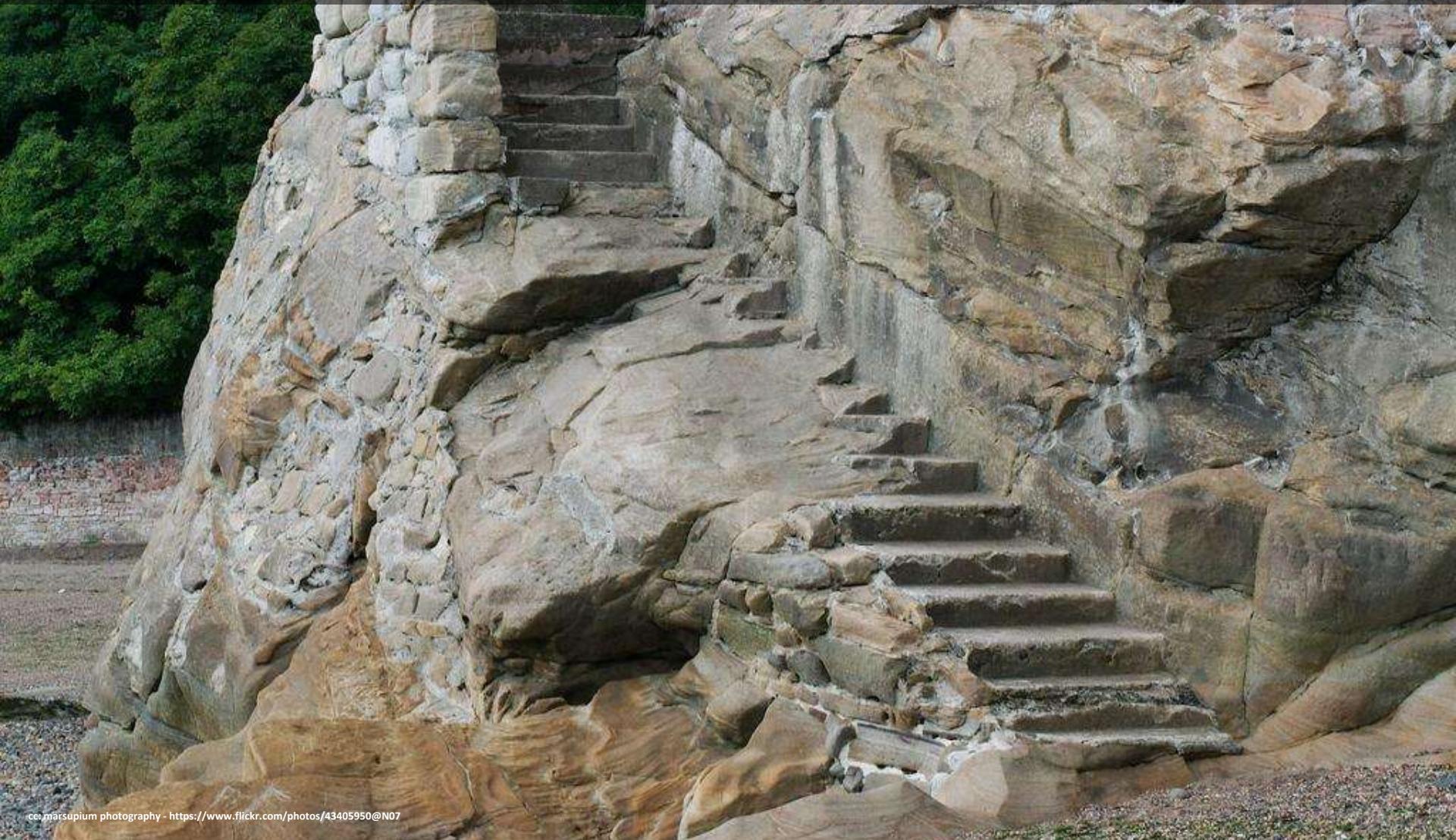
Beginning dump of physical memory

Physical memory dump complete.

Contact your system administrator or technical support group for further assistance.

cc: taberandrew - <https://www.flickr.com/photos/88442983@N00>

# Some steps...



cc: marsupium photography - <https://www.flickr.com/photos/43405950@N07>

#harnessing\_ddd

@gaythu\_rajan / @al94781

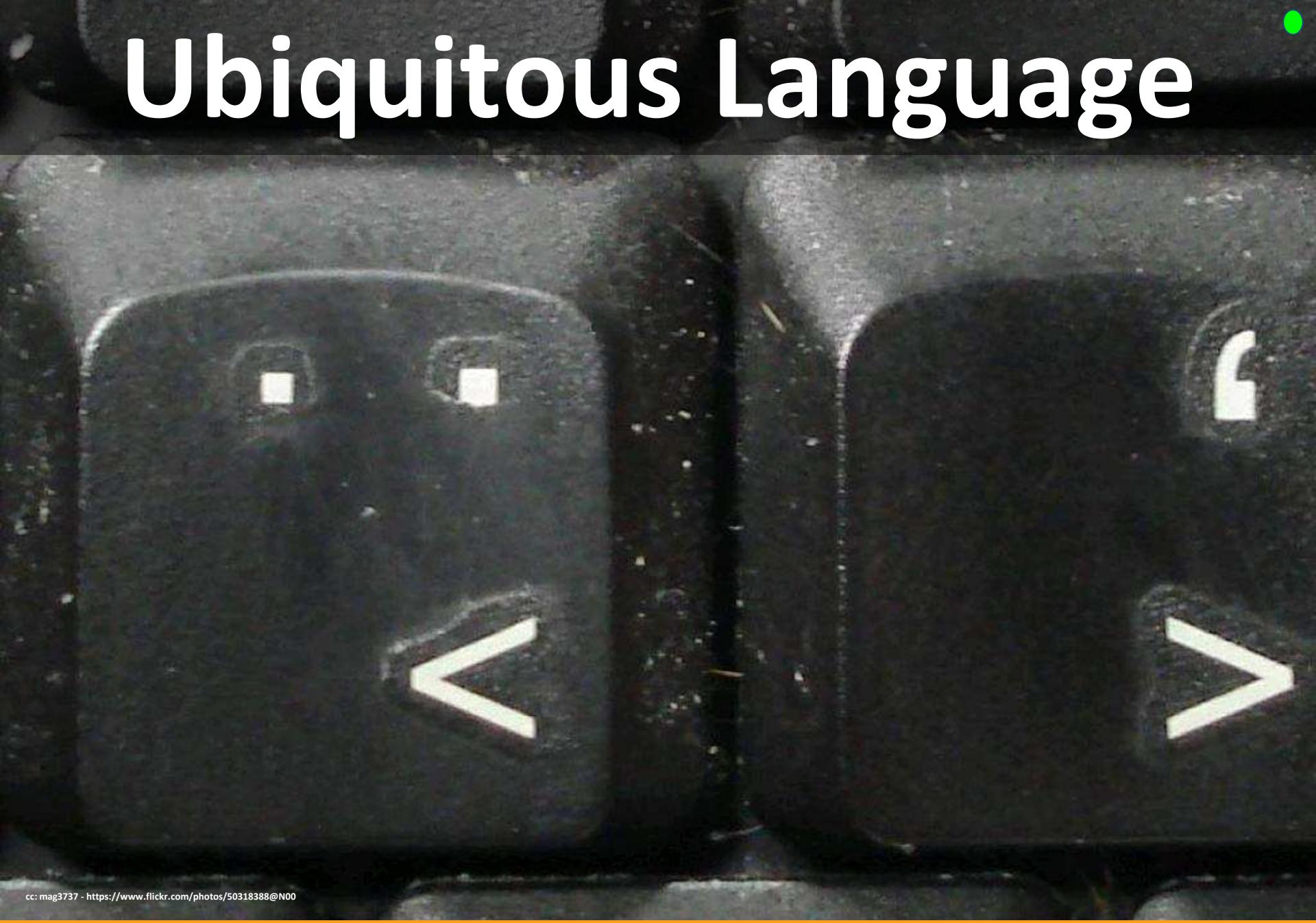


# The Core

How DDD helps our CODE

cc: Laurence Vagner - <https://www.flickr.com/photos/86078191@N00>

# Ubiquitous Language



cc: mag3737 - <https://www.flickr.com/photos/50318388@N00>

# Hands-On Modellers



cc: jovike - <https://www.flickr.com/photos/49503078599@N01>

#harnessing\_ddd

@gaythu\_rajan / @al94781

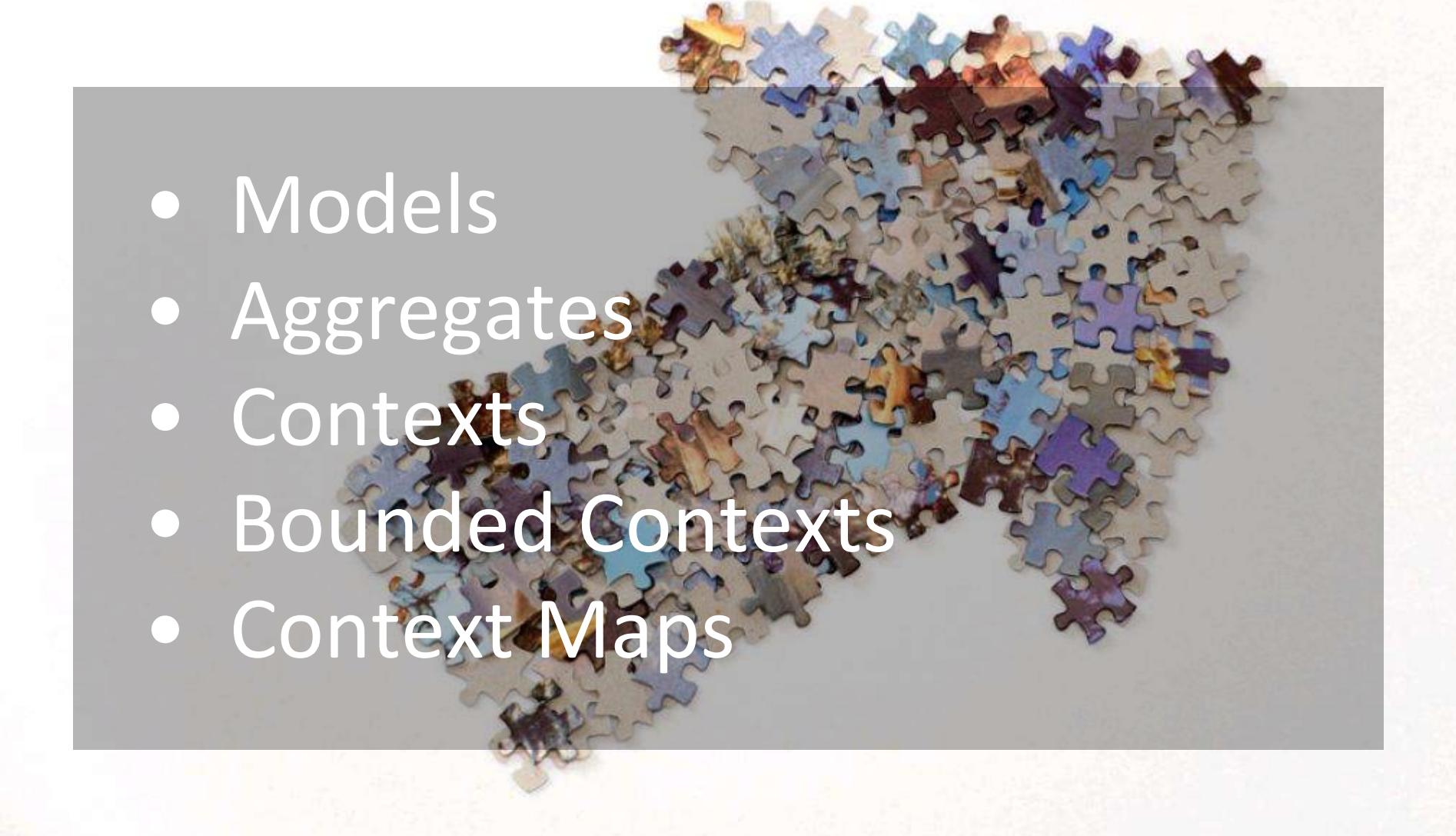
# The Misapplied

How DDD helps our overall Architecture

cc: juhansonin - <https://www.flickr.com/photos/38869431@N00>

# The Important Bits

- Models
- Aggregates
- Contexts
- Bounded Contexts
- Context Maps



# Our Case Study: Shared Payment Services



cc: Khanh Hmoong - <https://www.flickr.com/photos/7997148@N05>



# Shared Payment Services





# Shared Payment Services

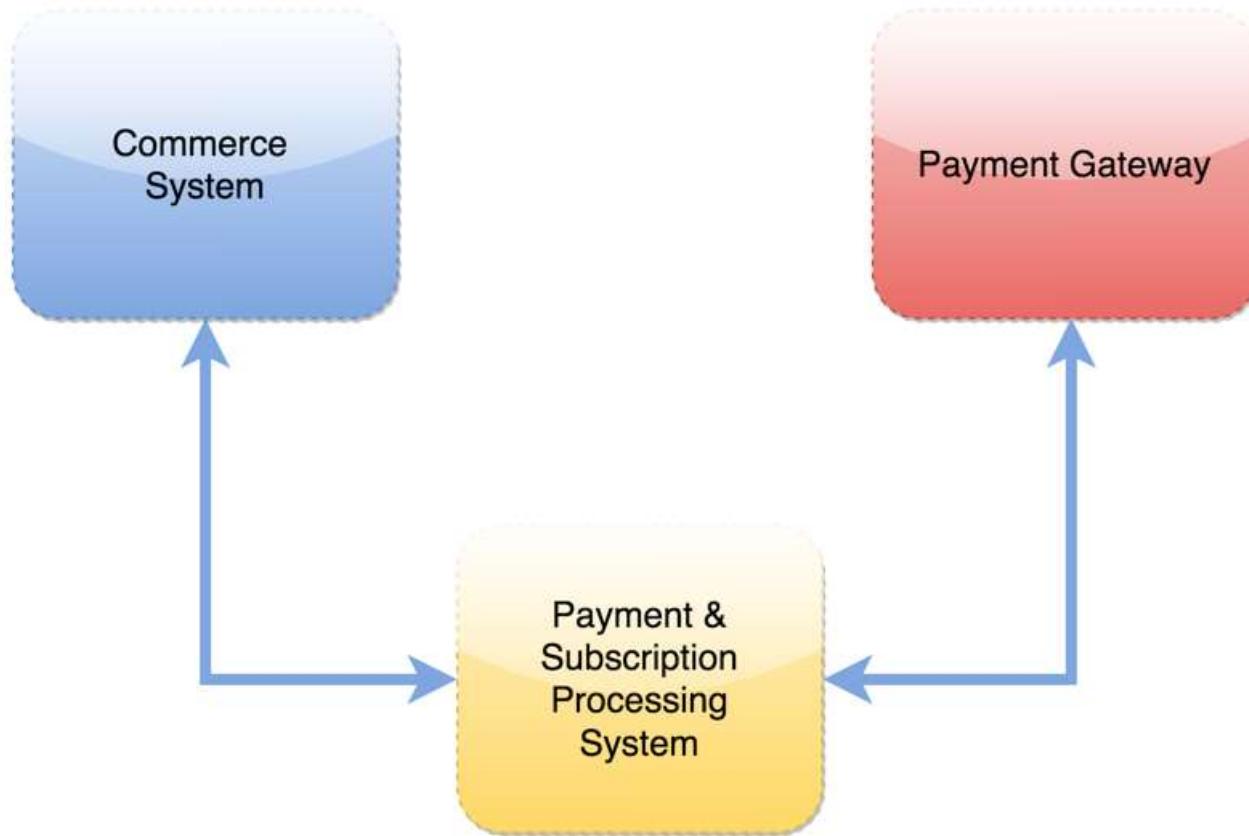




# Shared Payment Services



# Shared Payment Services



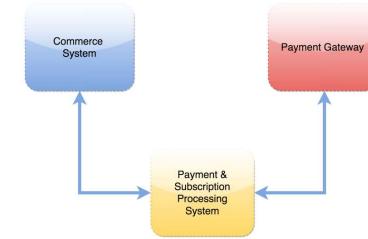
# Models



cc: Rob Swatski - <https://www.flickr.com/photos/50946274@N02>



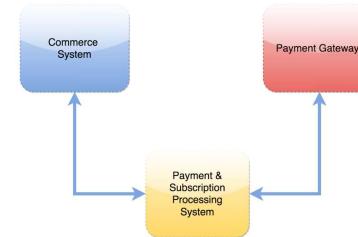
Payment





Payment

Subscription

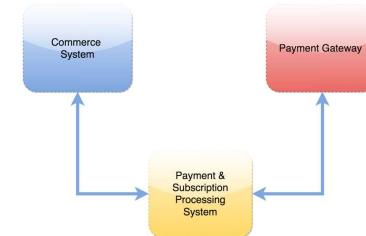




Payment

Subscription

Mandate



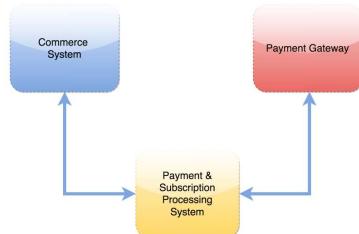


Order

Payment

Subscription

Mandate





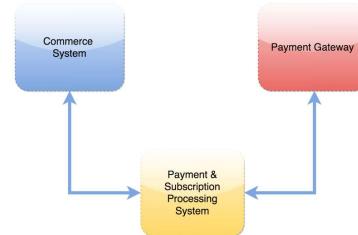
Order

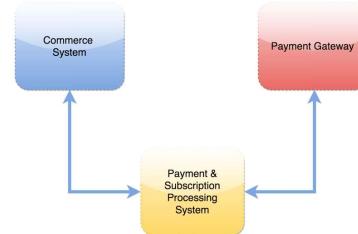
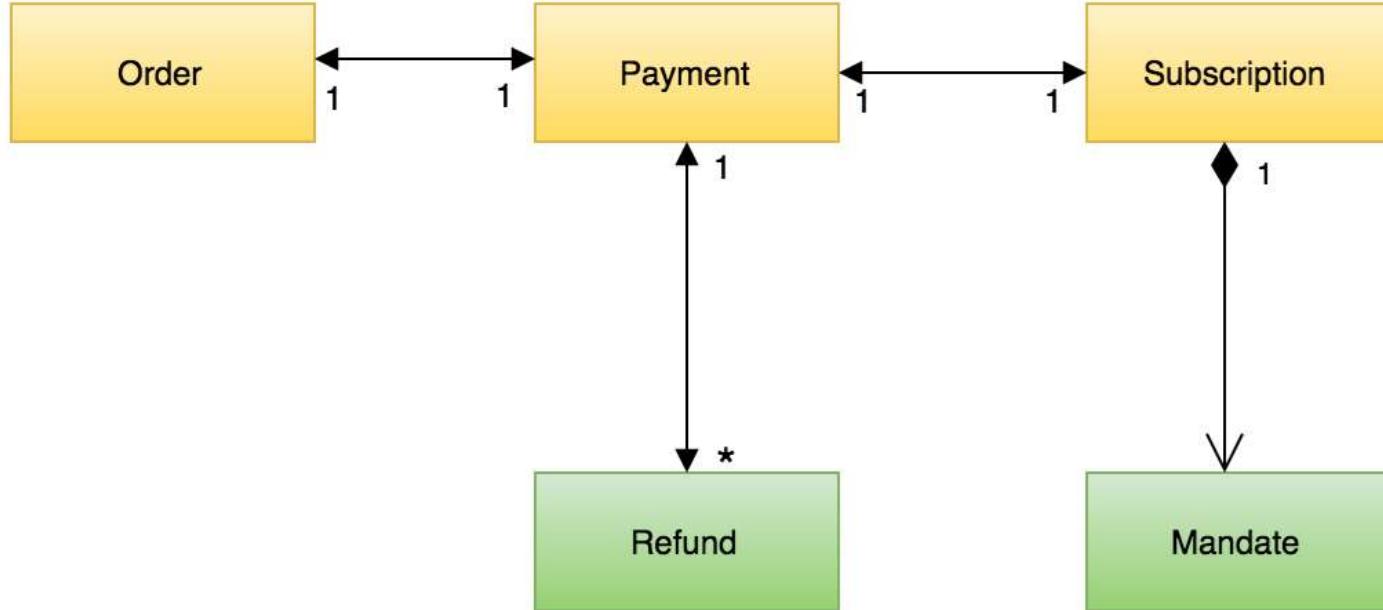
Payment

Subscription

Refund

Mandate

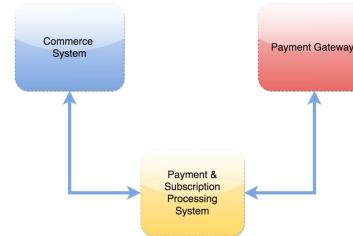
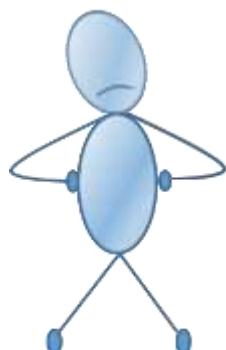
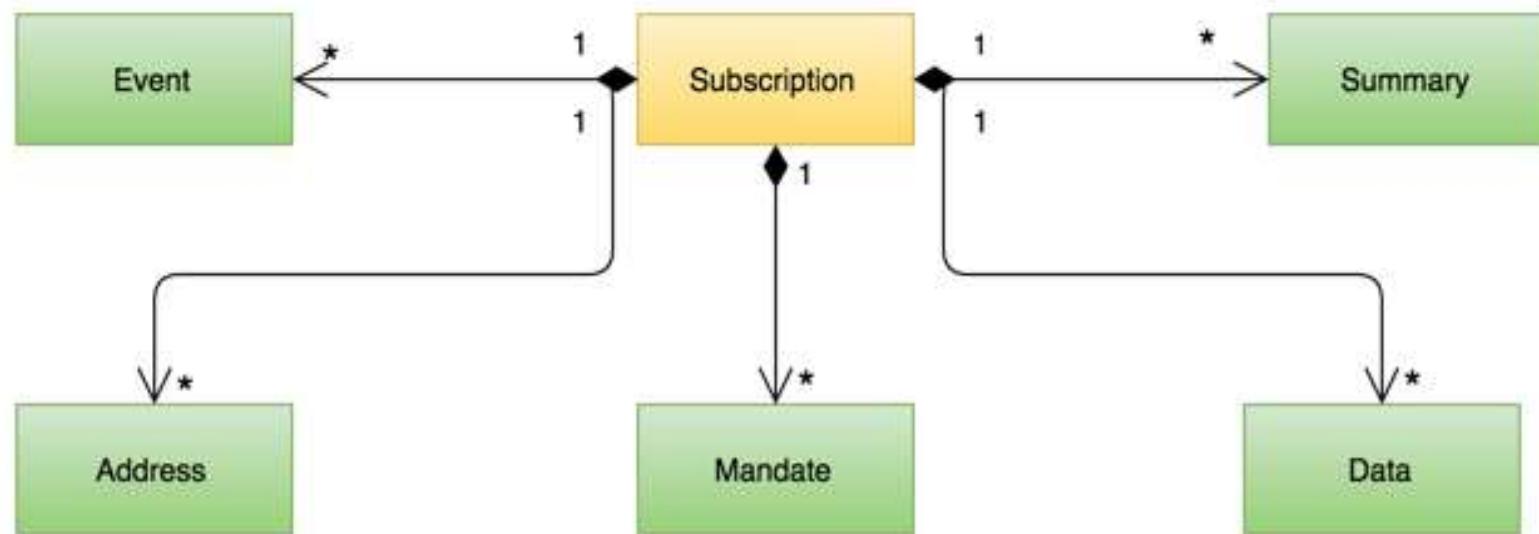




# Aggregates



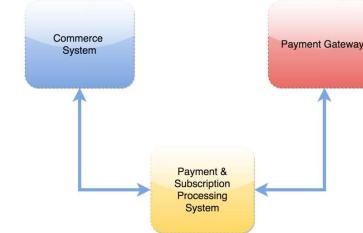
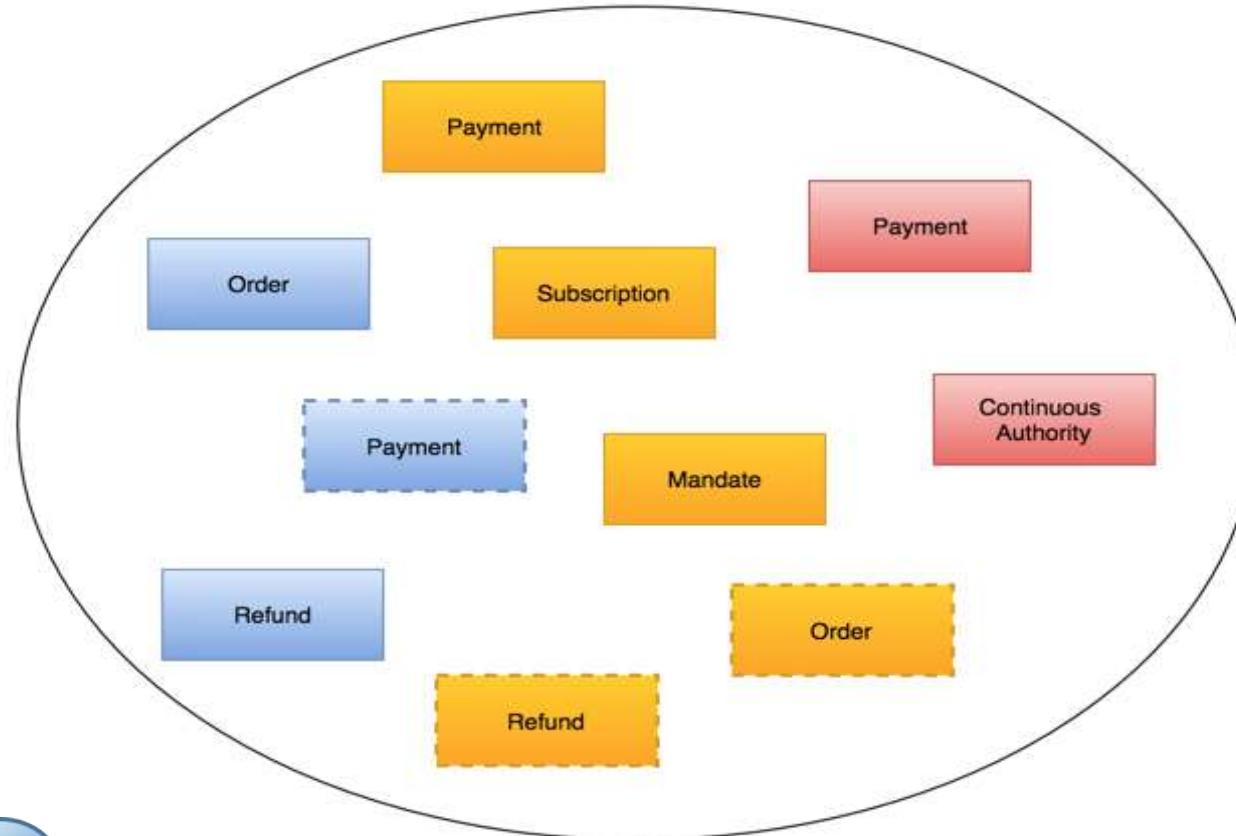
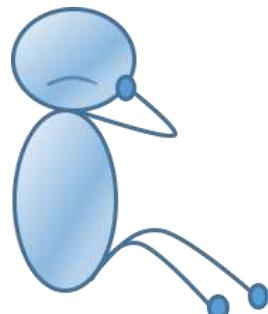
cc: J-W Brown - <https://www.flickr.com/photos/13797487@N00>

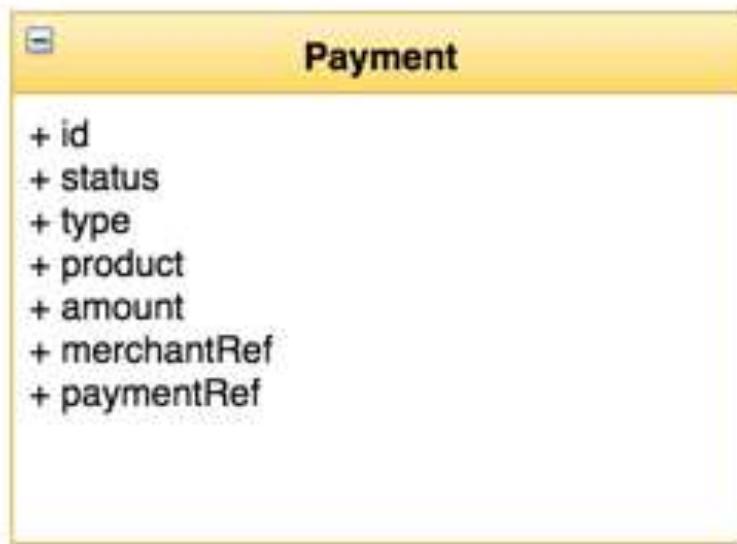
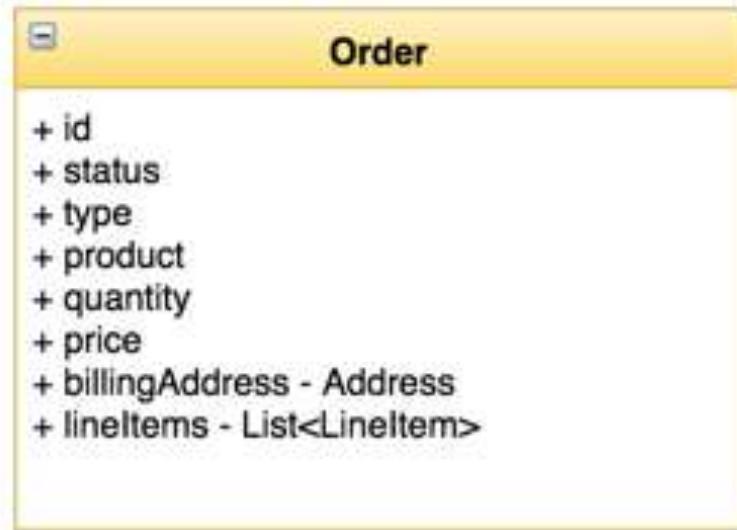


# Contexts



cc: petitshoo - <https://www.flickr.com/photos/29426084@N00>

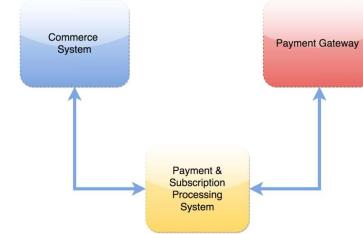
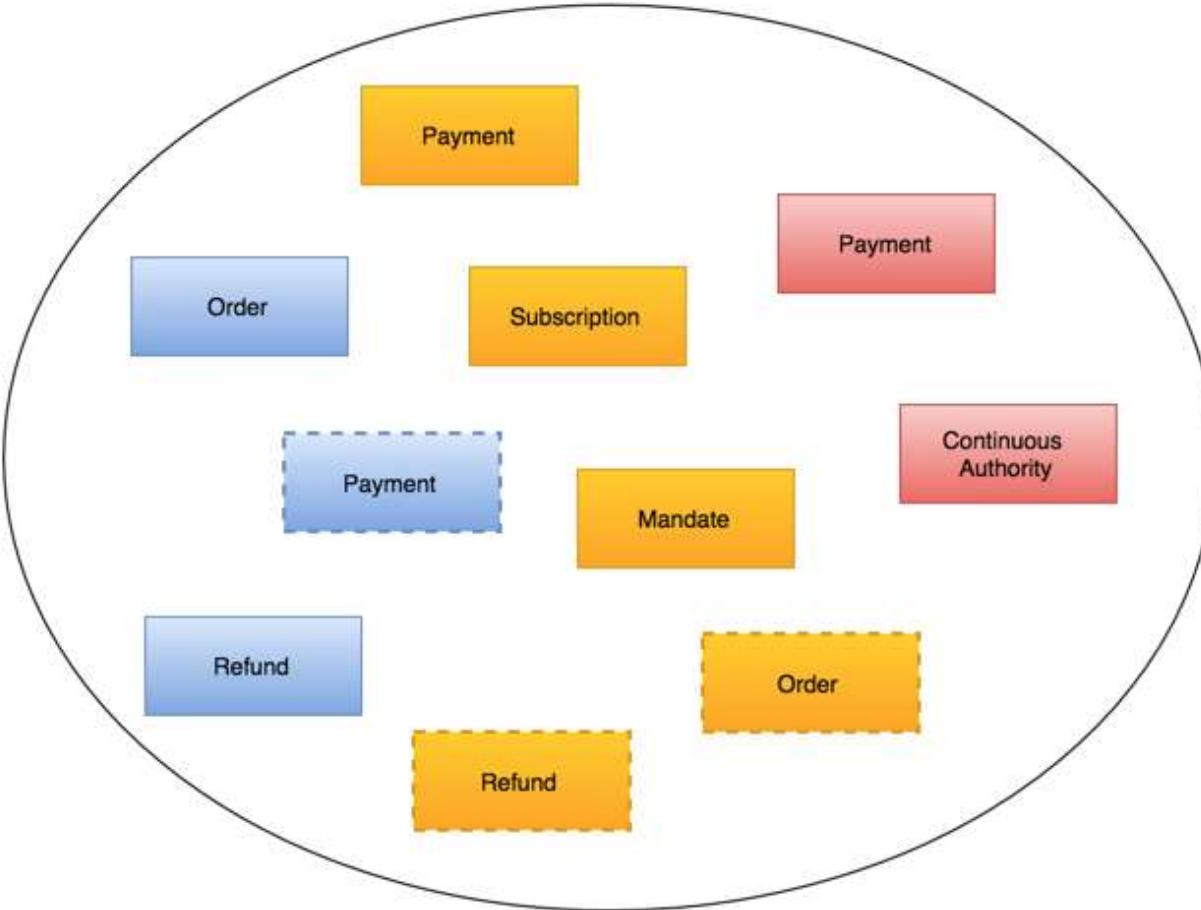


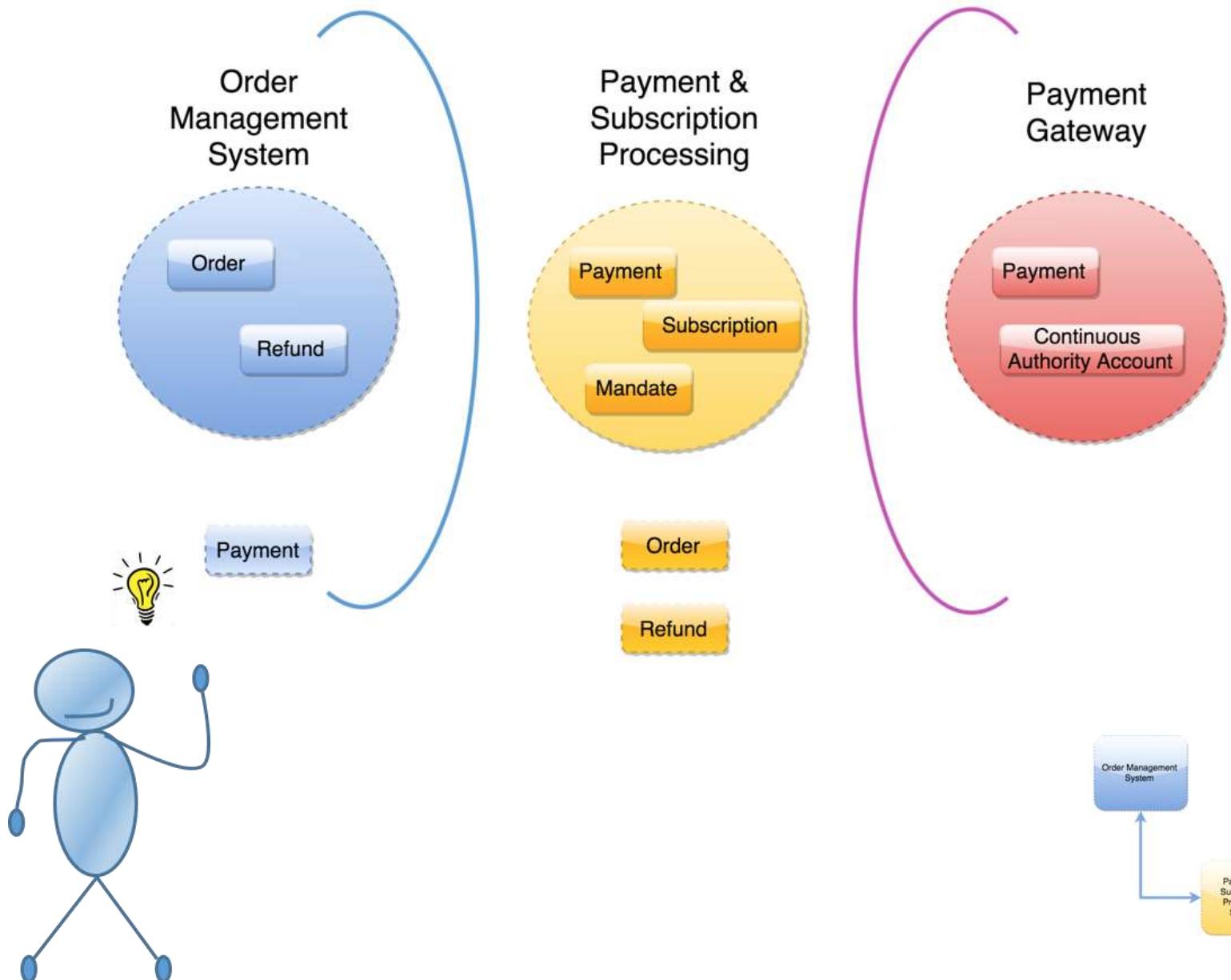


# Bounded Contexts

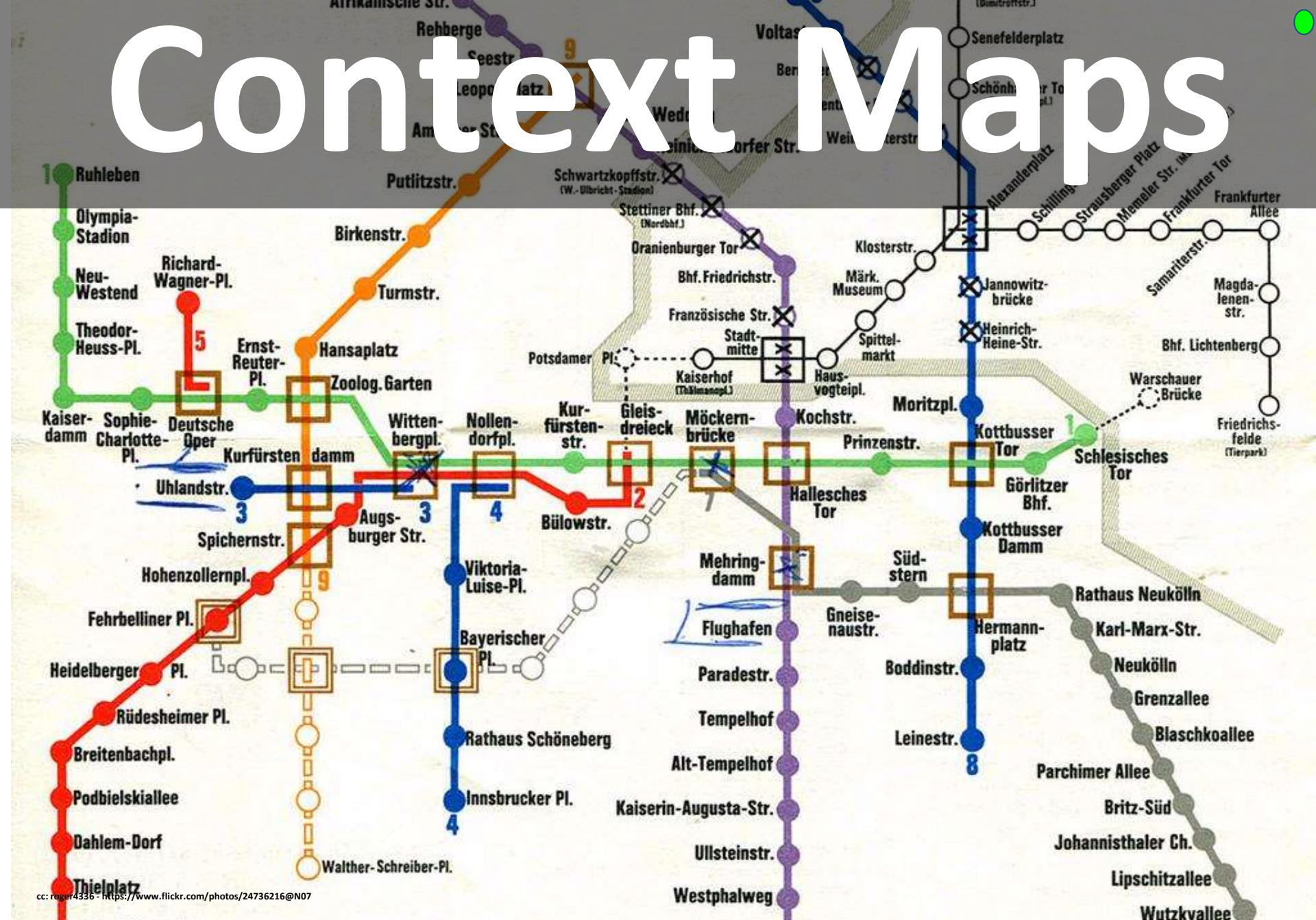


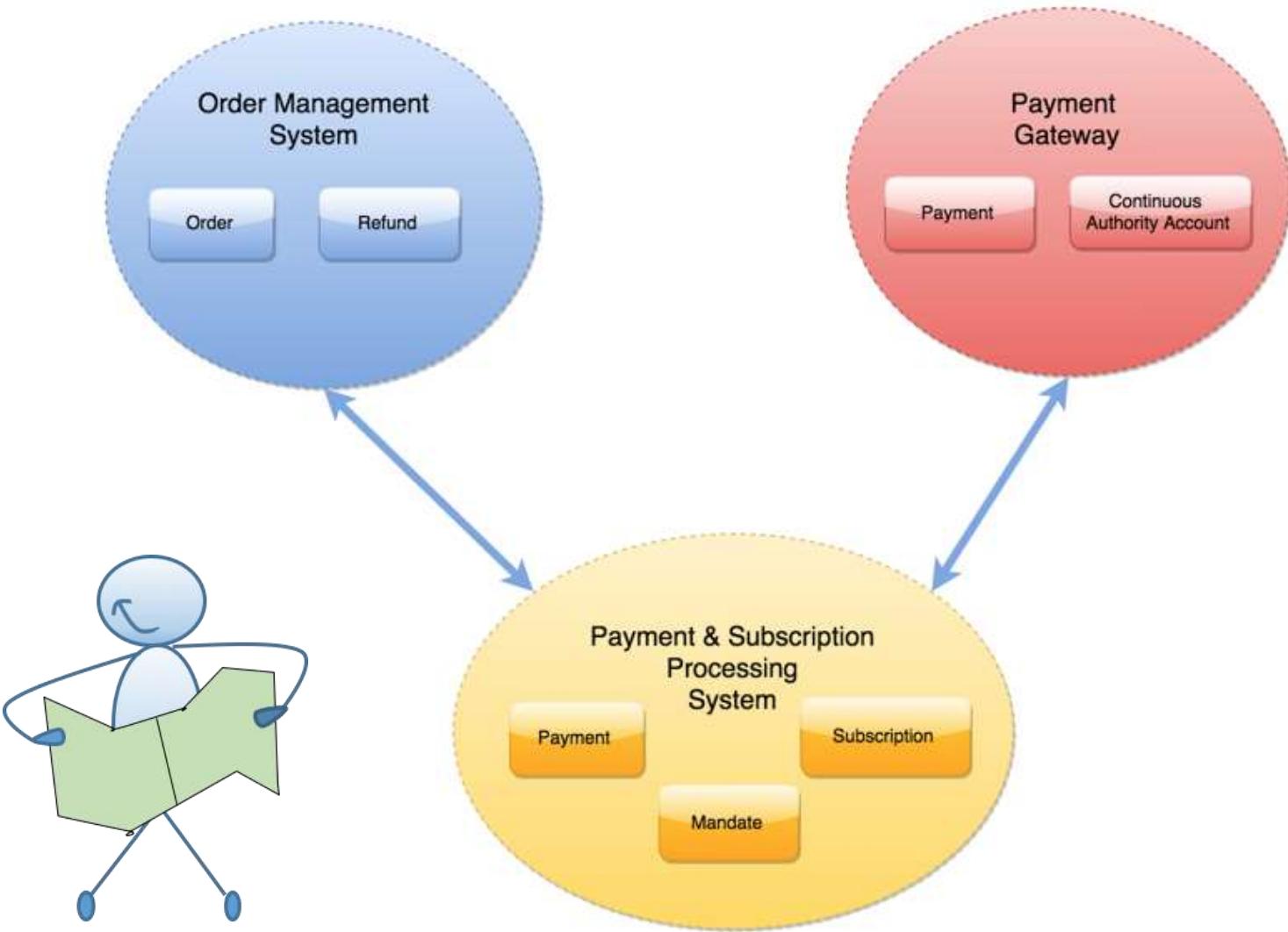
cc: bryce\_edwards - <https://www.flickr.com/photos/98185064@N00>



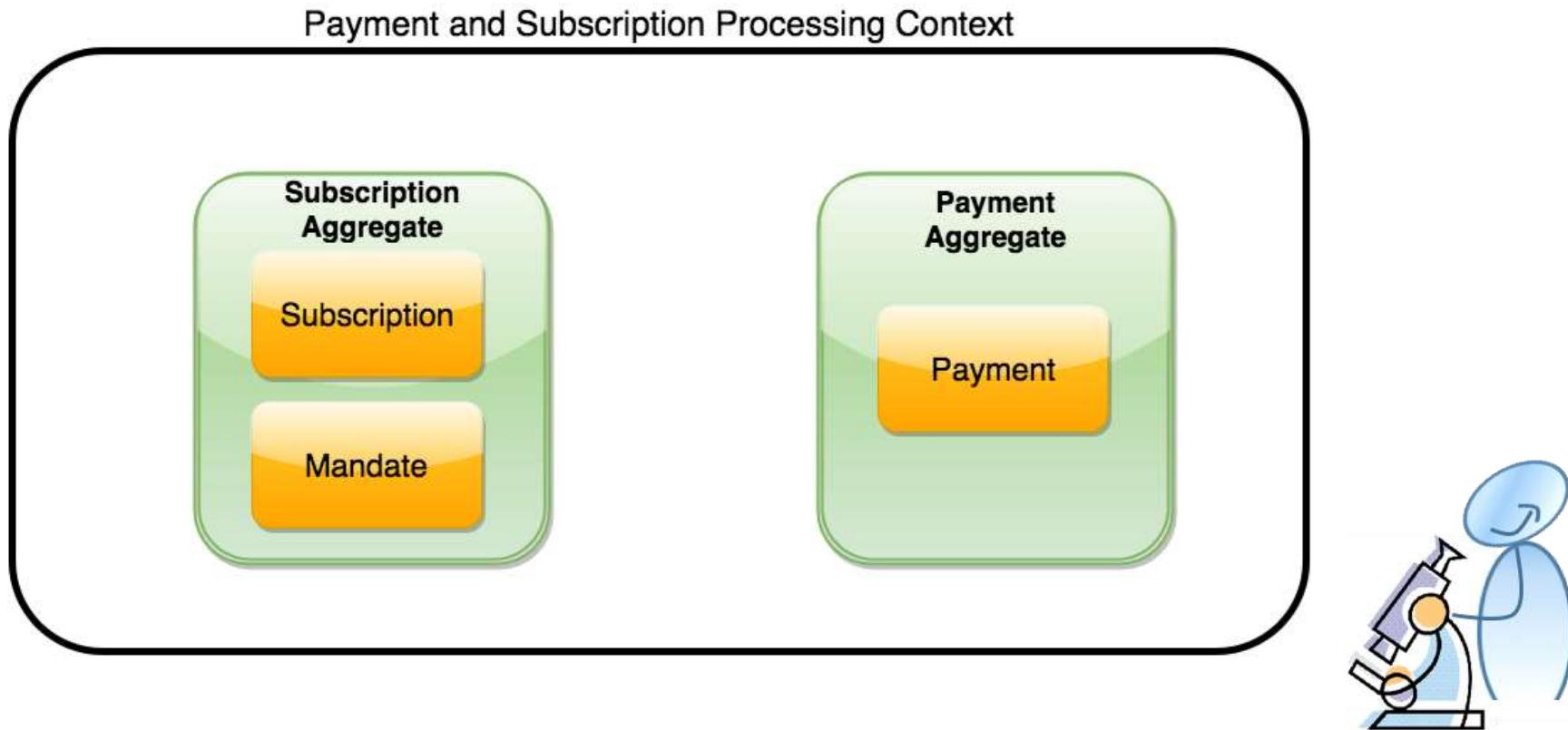


# Context Maps

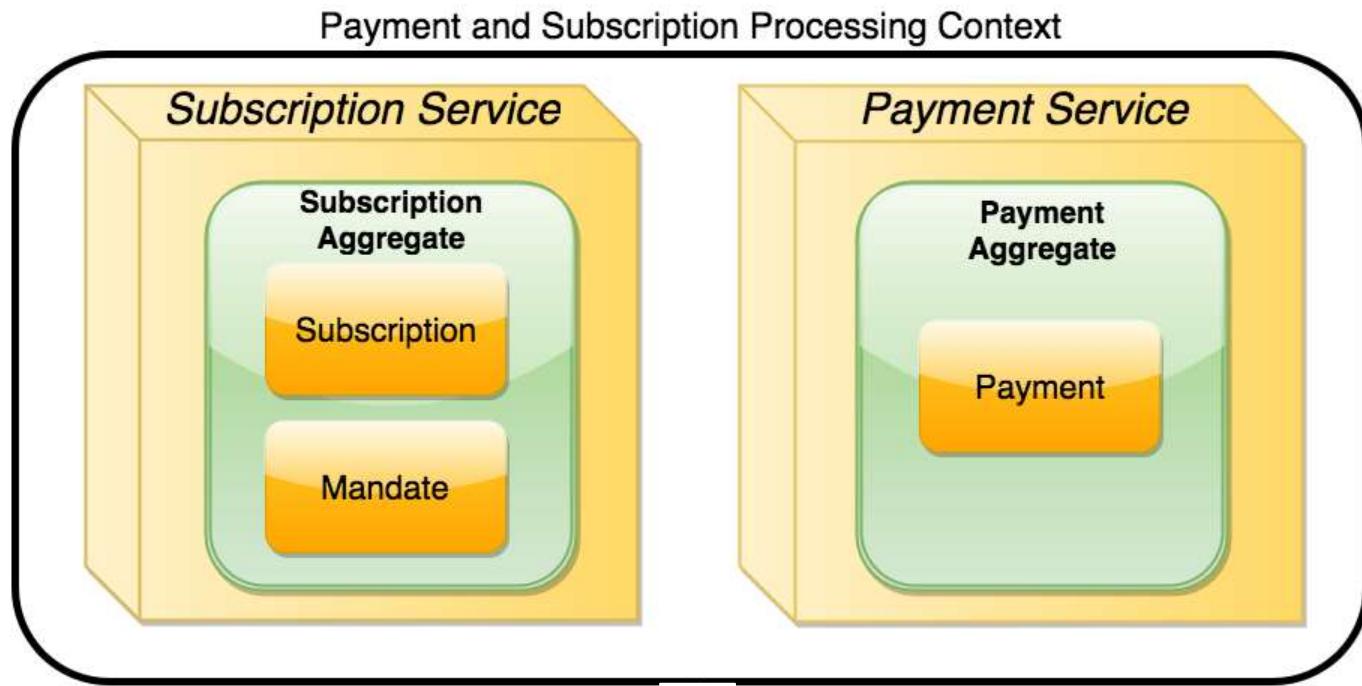




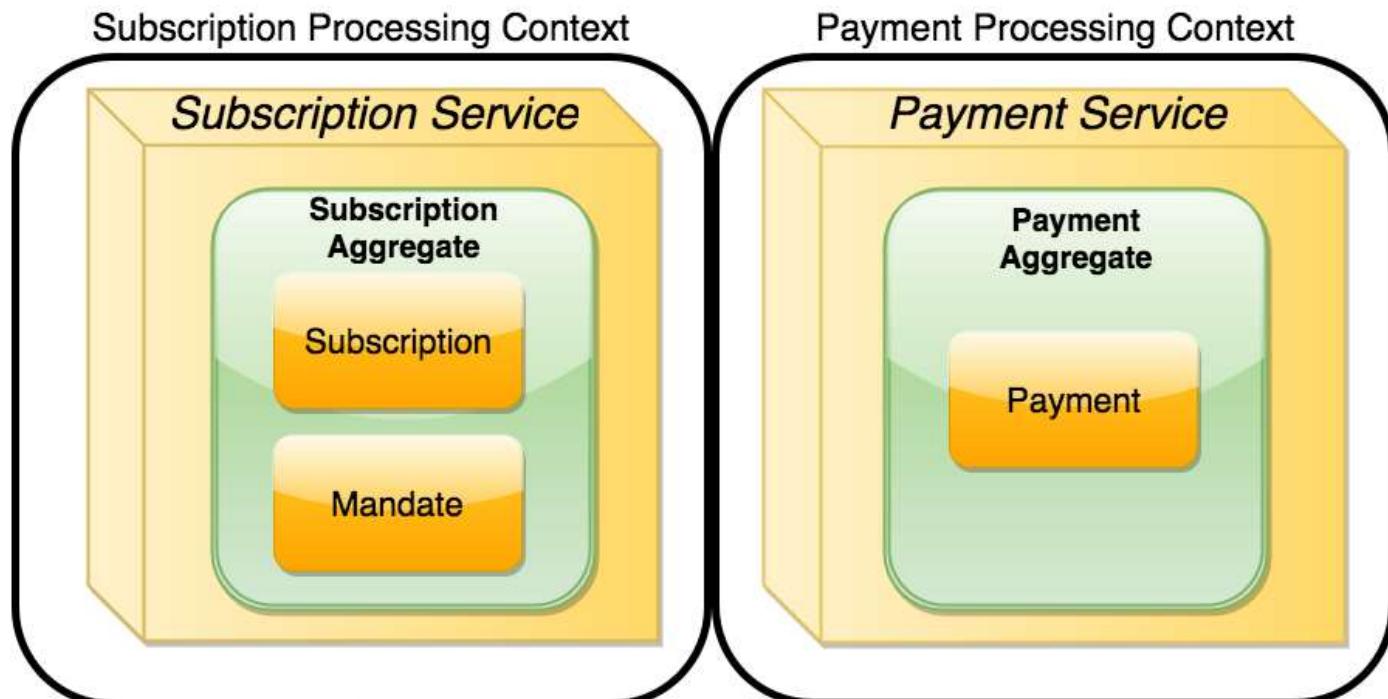
# Why Aggregates?



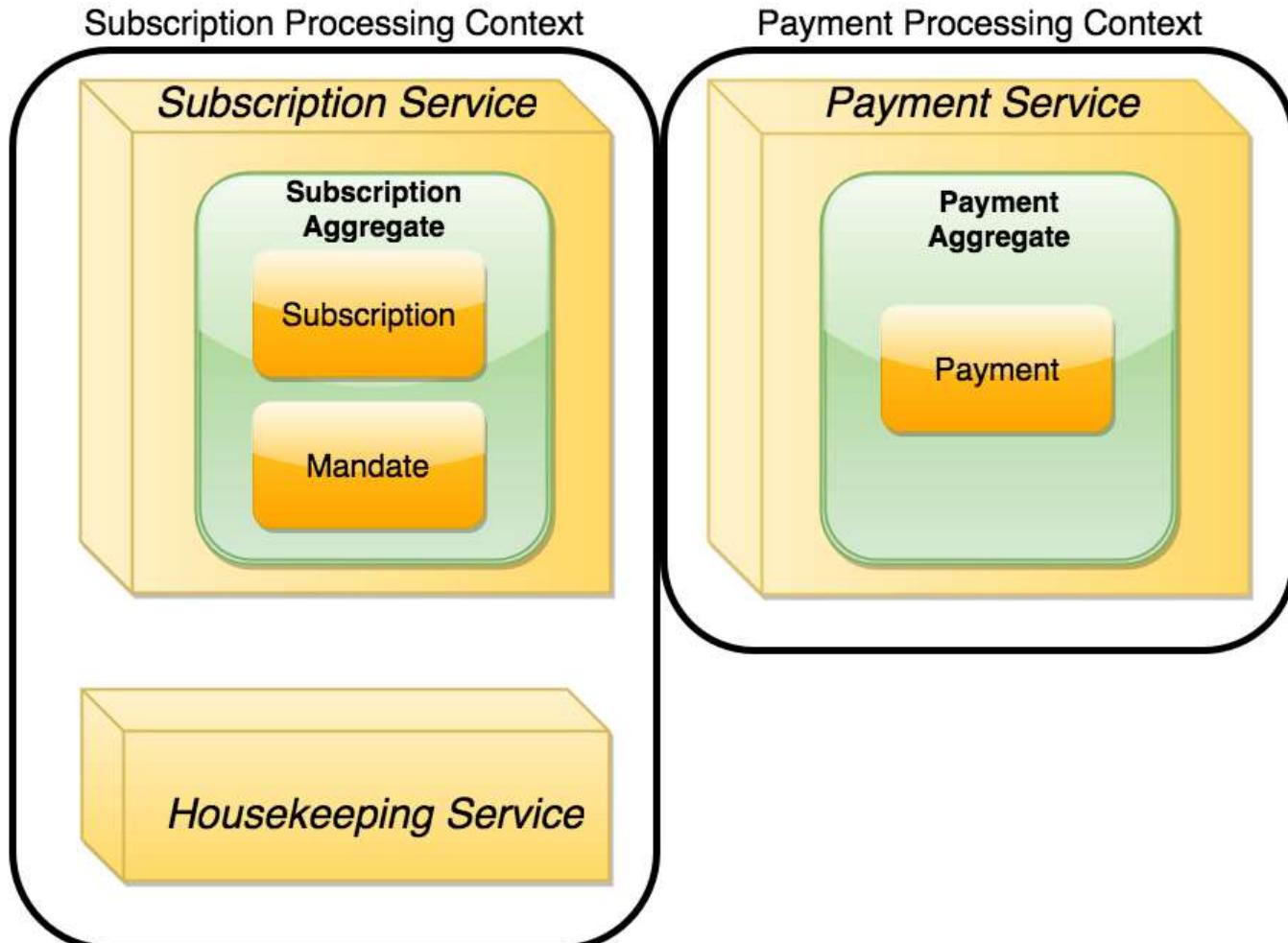
# Aggregate == Microservice



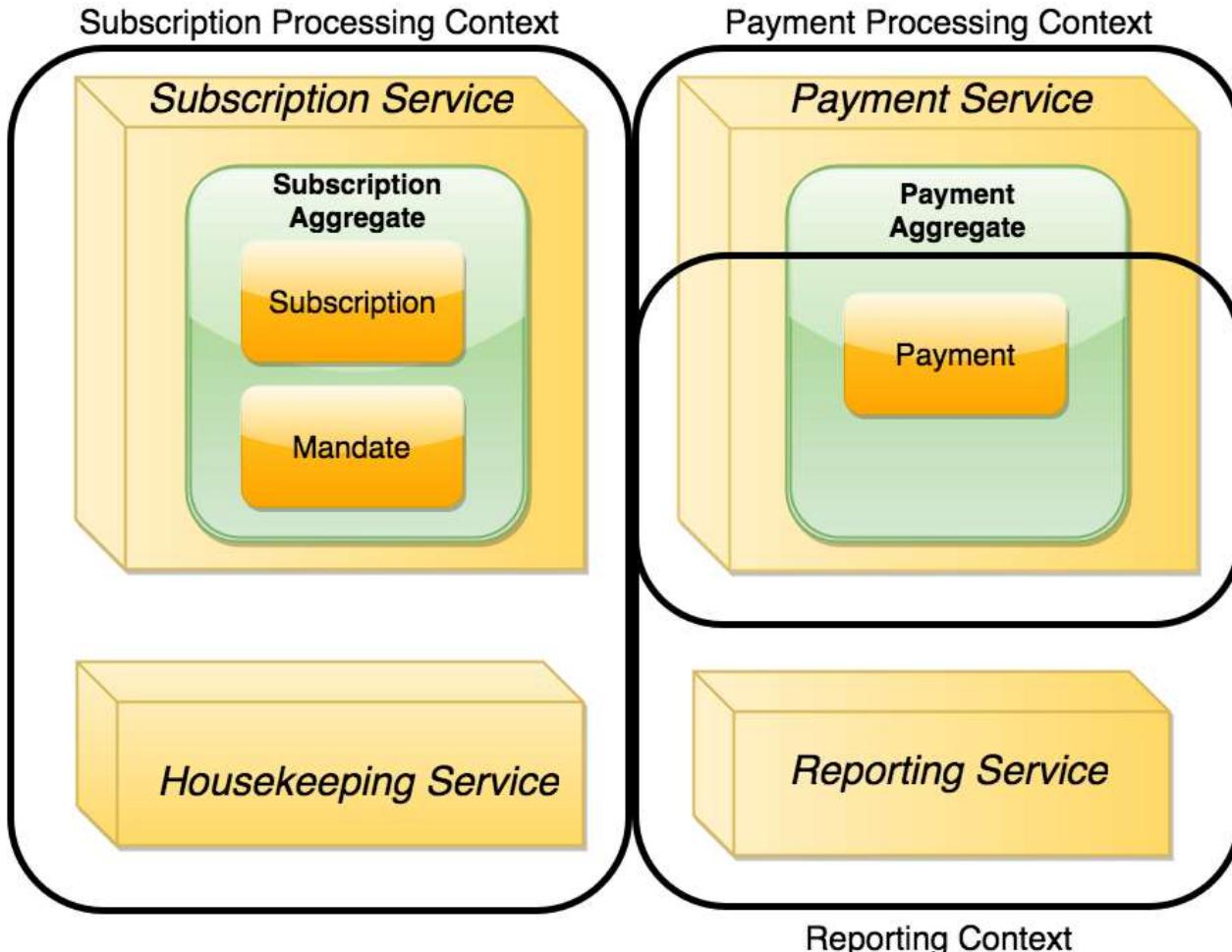
# Aggregate == Microservice

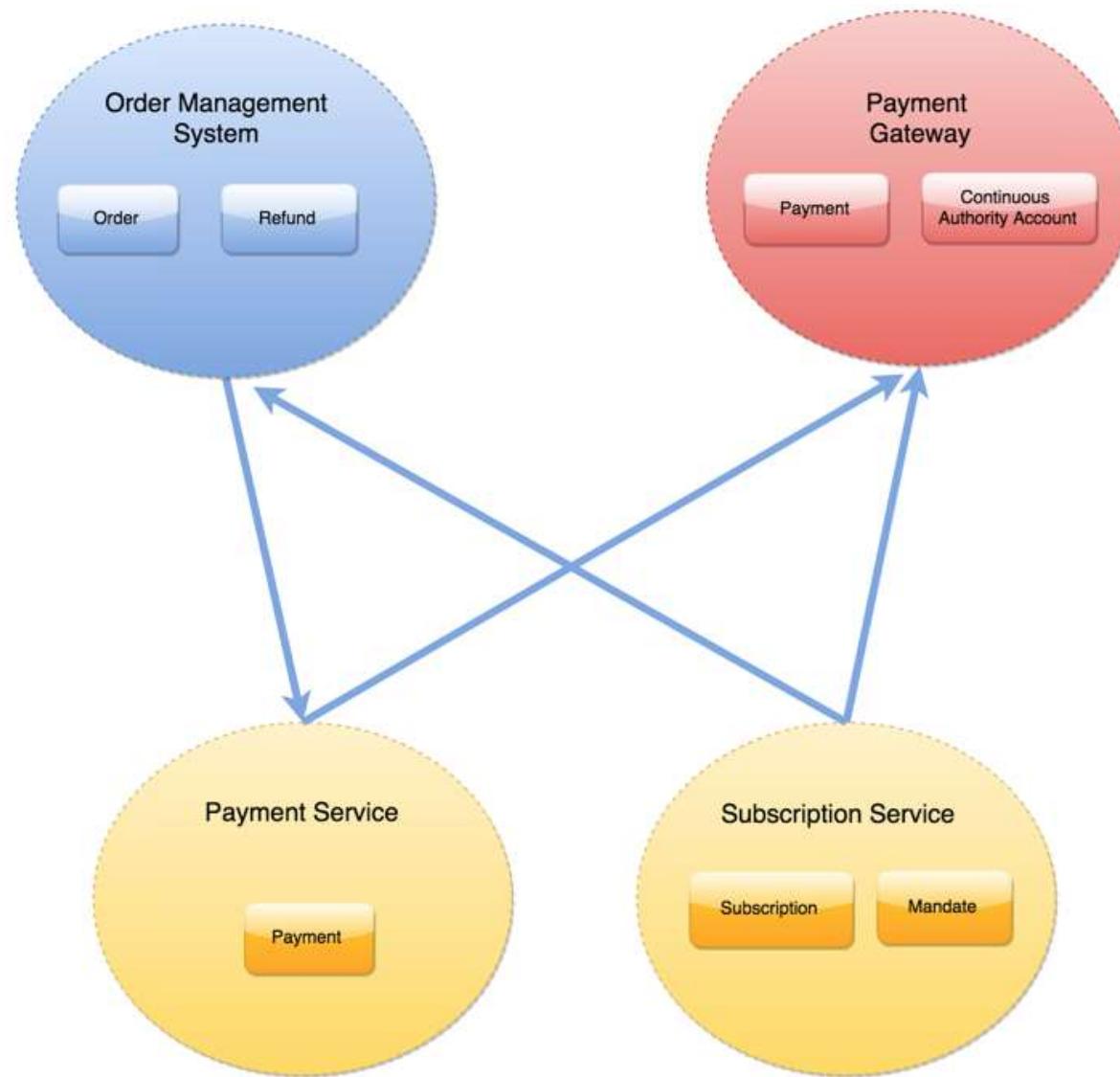
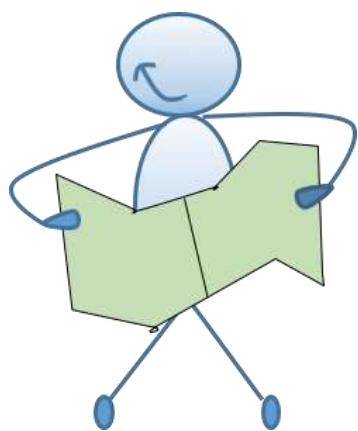


# But, Aggregate != Microservice (not always)



# But, Aggregate != Microservice (not always)





# An Approach

1. Draw a Model and begin to implement it

cc: Muffet - <https://www.flickr.com/photos/53133240@N00>

Purposive Self-Crème.

# An Approach

1. Draw a Model and begin to implement it
2. As you draw it and code it, listen to the Language and cultivate your Model accordingly

cc: Muffet - <https://www.flickr.com/photos/53133240@N00>

# An Approach

1. Draw a Model and begin to implement it
2. As you draw it and code it, listen to the Language and cultivate your Model accordingly
3. Identify your Aggregates - these are your starter Microservices

# An Approach

1. Draw a Model and begin to implement it
2. As you draw it and code it, listen to the Language and cultivate your Model accordingly
3. Identify your Aggregates - these are your starter Microservices
4. When you hit pain points (indicated by the Language) split into different Bounded Contexts to resolve

# An Approach

1. Draw a Model and begin to implement it
2. As you draw it and code it, listen to the Language and cultivate your Model accordingly
3. Identify your Aggregates - these are your starter Microservices
4. When you hit pain points (indicated by the Language) split into different Bounded Contexts to resolve
5. Revisit your Bounded Contexts now you know more about how each Aggregate - you may end up splitting them

cc: Muffet - <https://www.flickr.com/photos/53133240@N00>

# Done?



cc: WarzauWynn - <https://www.flickr.com/photos/94246031@N00>

#harnessing\_ddd

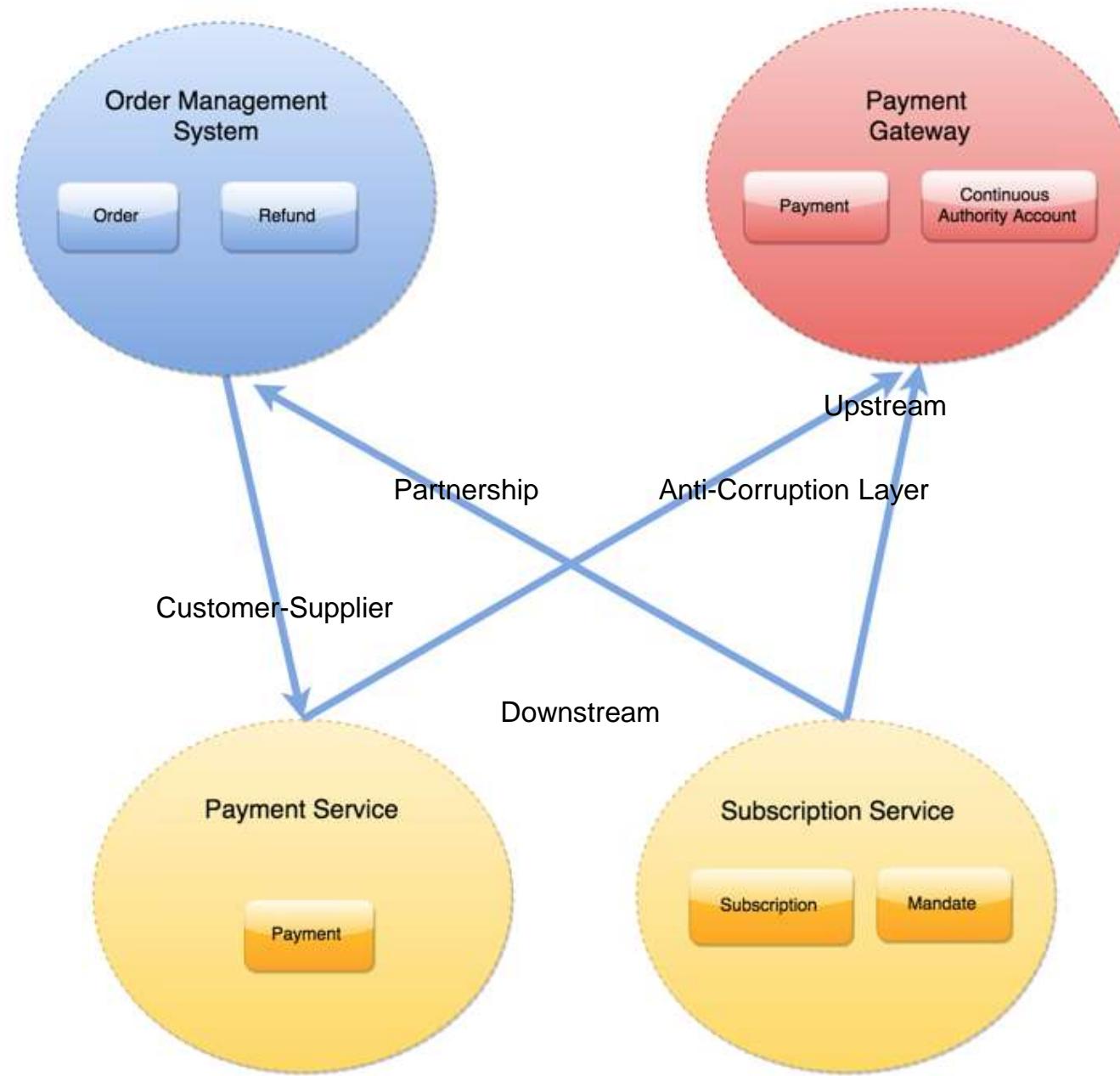
@gaythu\_rajan / @al94781

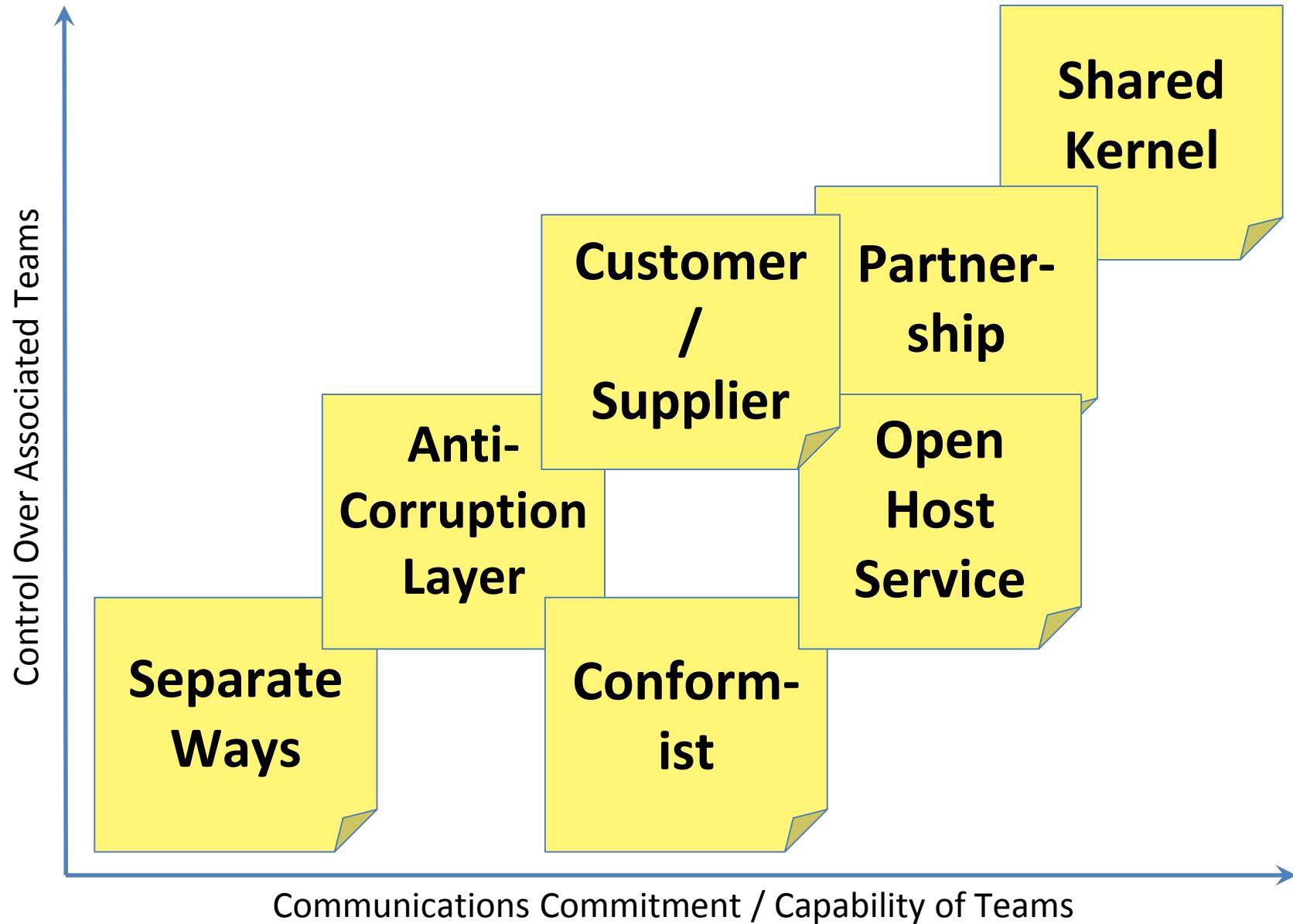


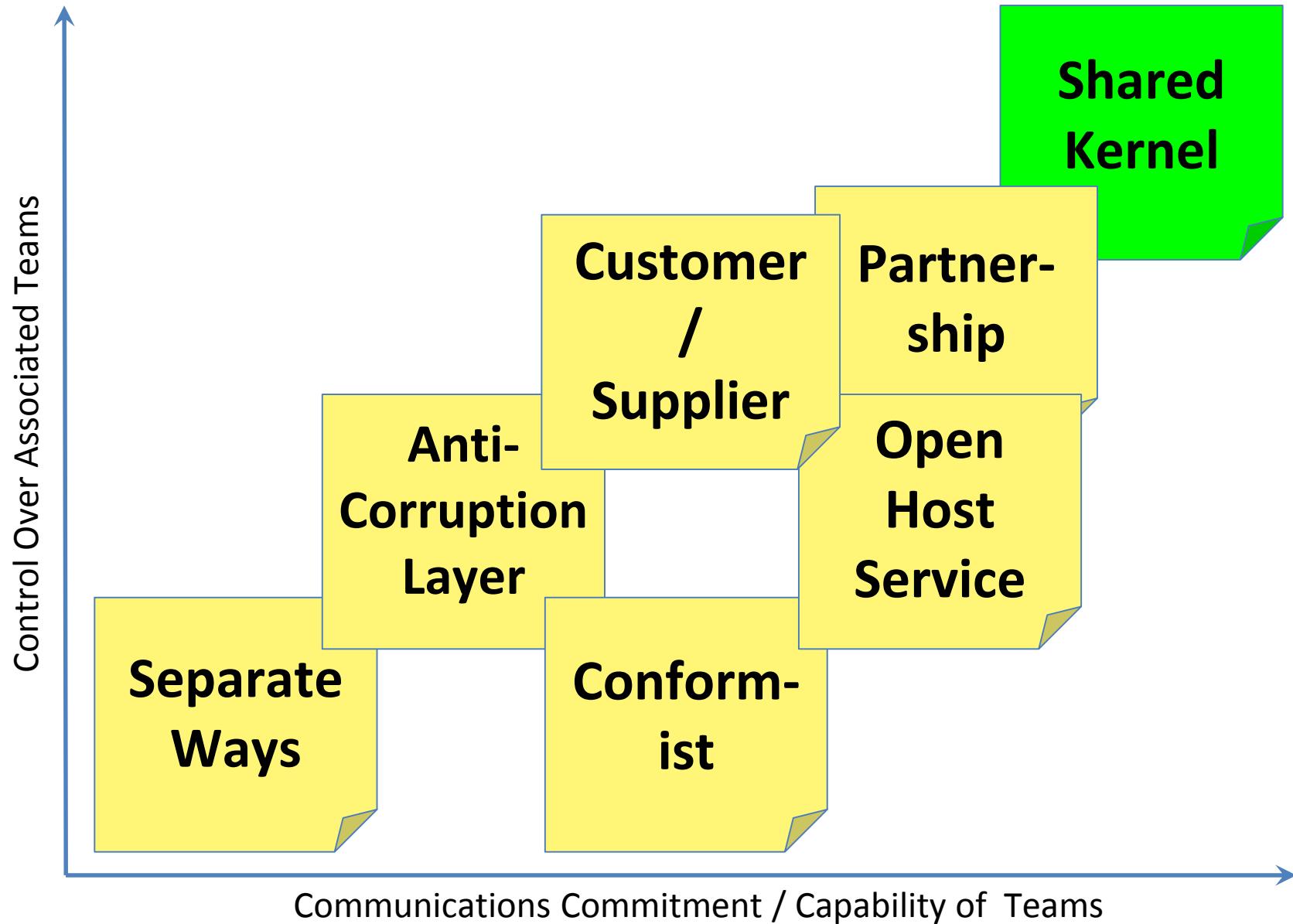
# The Ignored

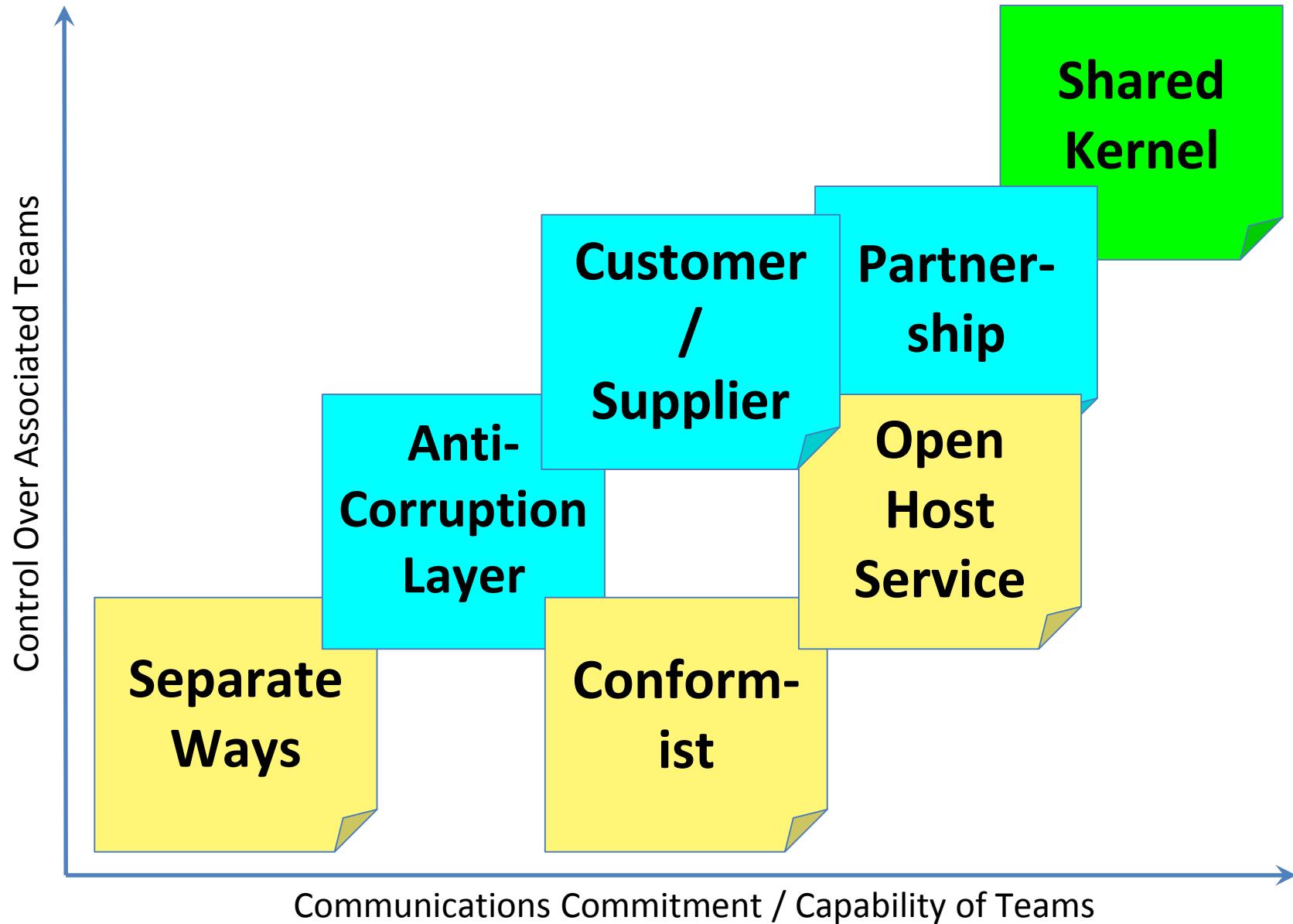
How DDD helps our Teams Organise

cc: ahmed\_tehaha - <https://www.flickr.com/photos/21549162@N05>









# Upstream/Downstream



cc: archer10 (Dennis) (73M Views) · <https://www.flickr.com/photos/2249017@N02>

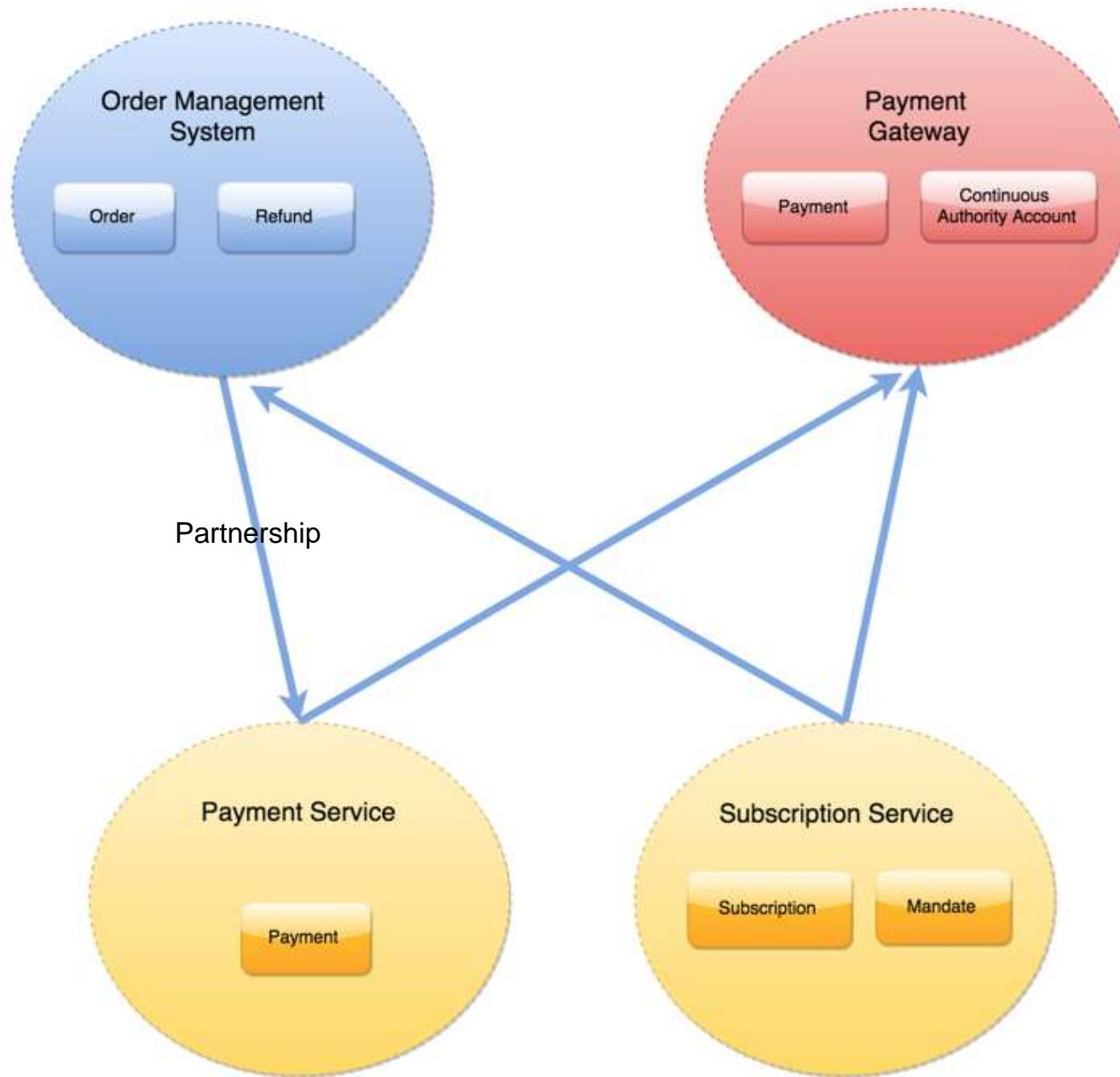
# Partnership

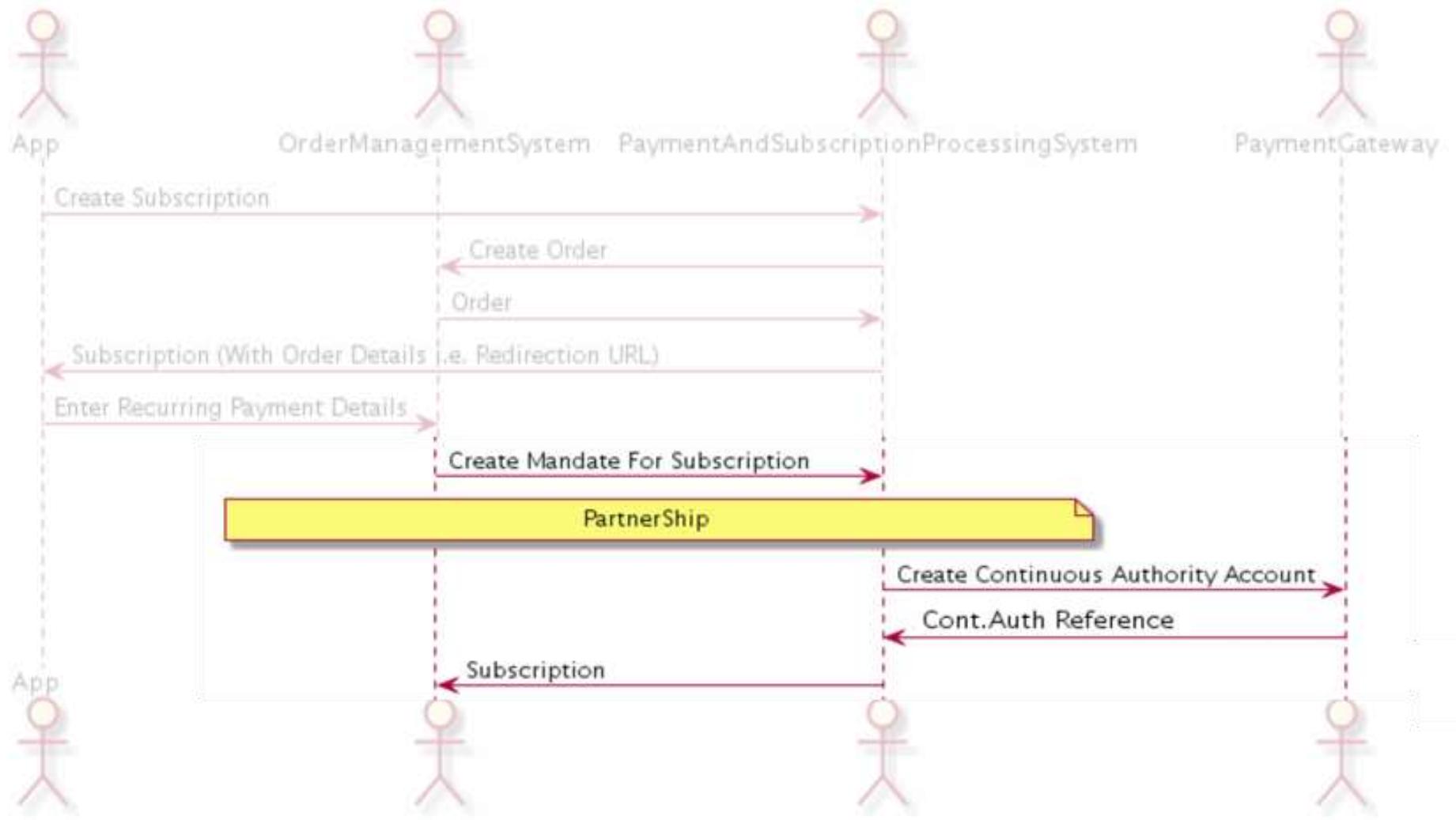


cc: practicalowl - <https://www.flickr.com/photos/55046645@N00>

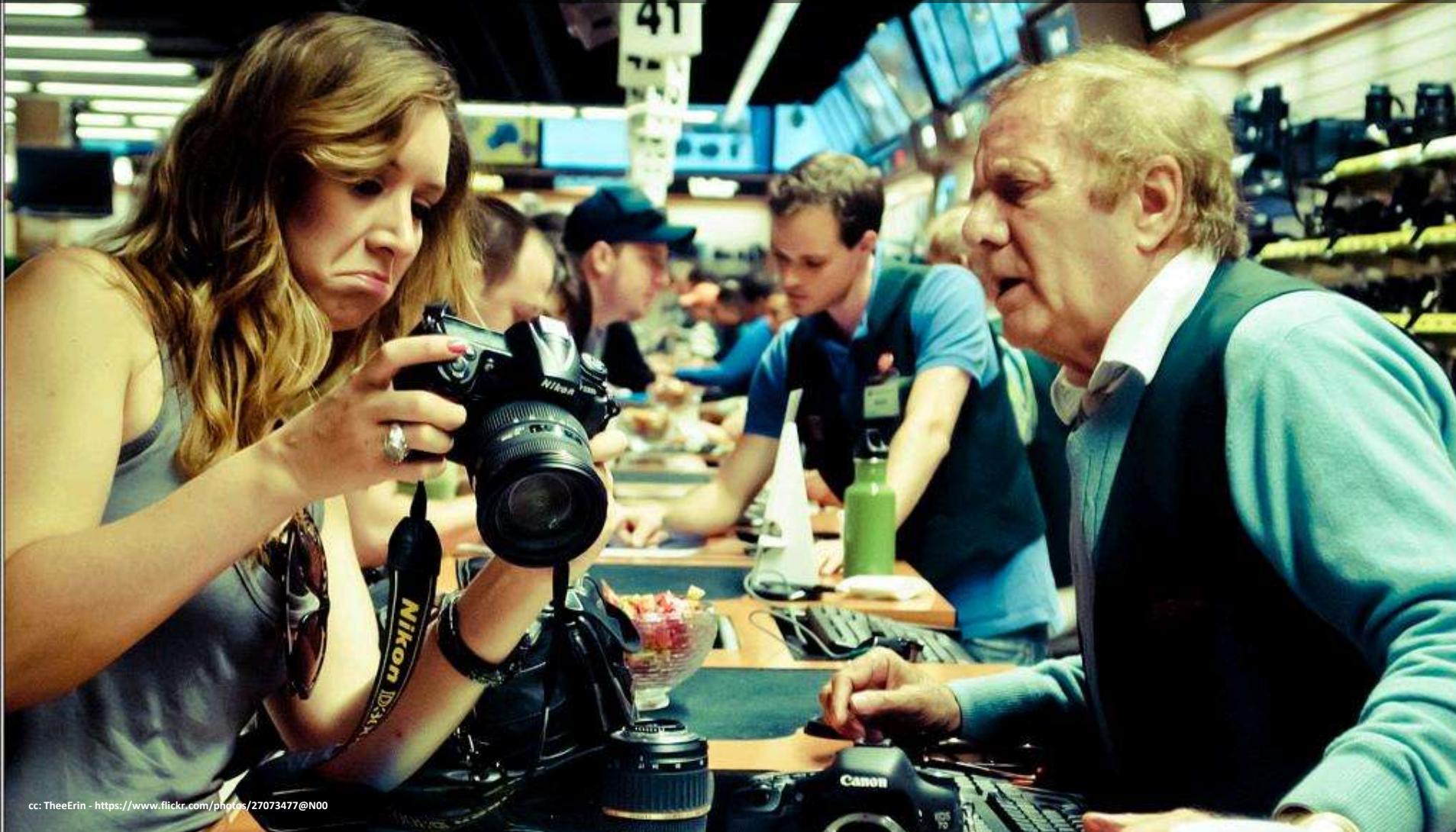
#harnessing\_ddd

@gaythu\_rajan / @al94781





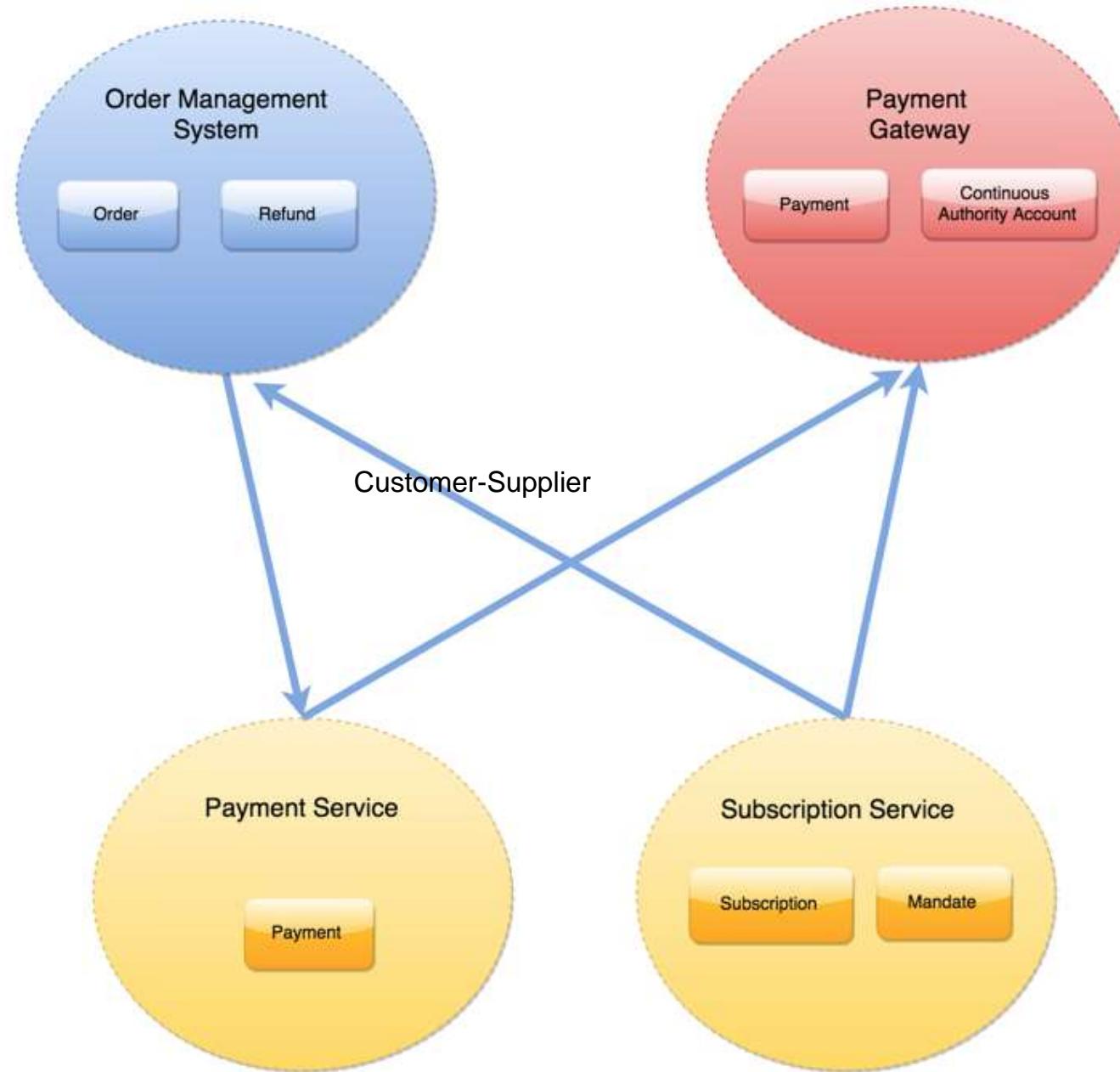
# Customer-Supplier

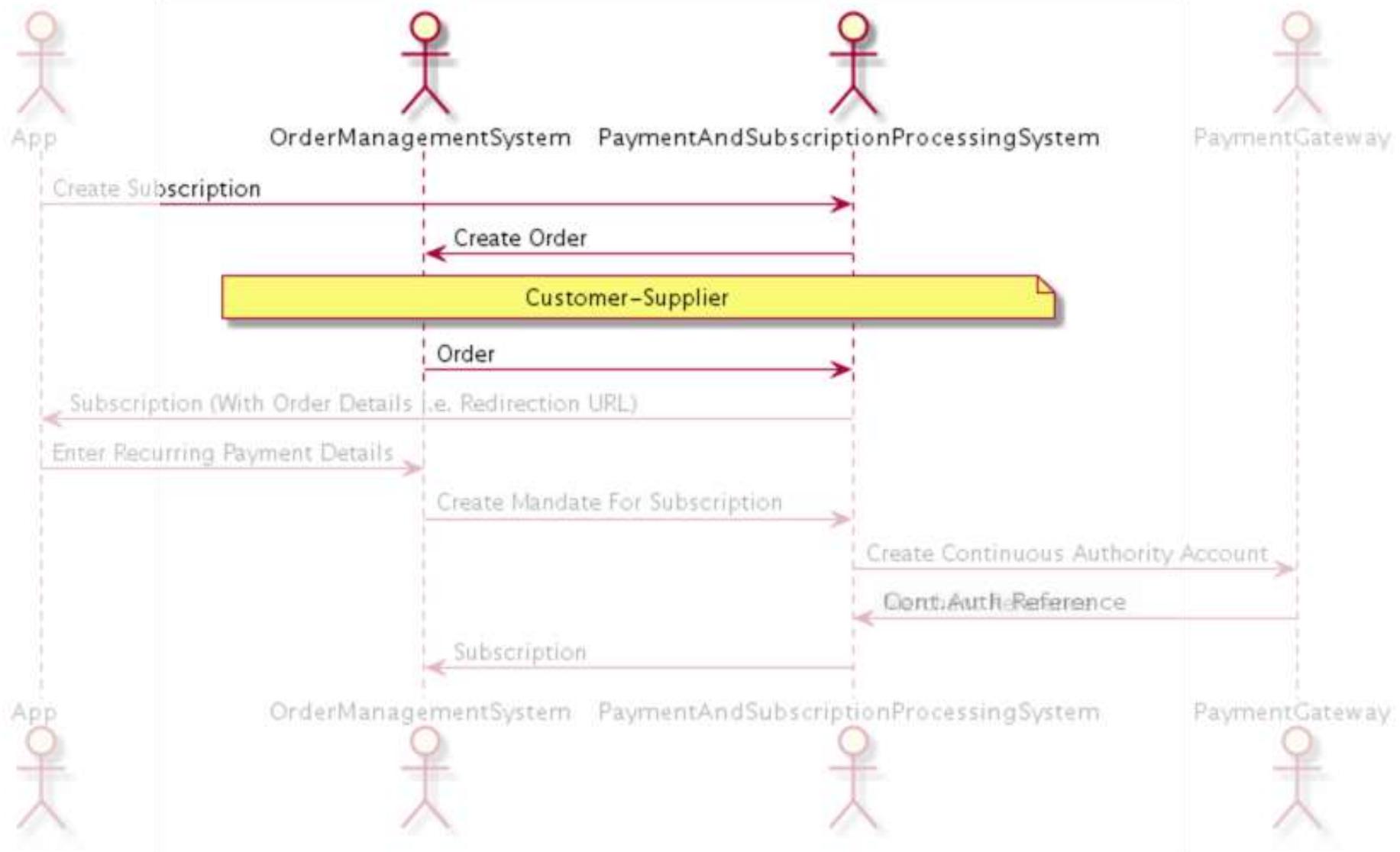


cc: TheeErin - <https://www.flickr.com/photos/27073477@N00>

#harnessing\_ddd

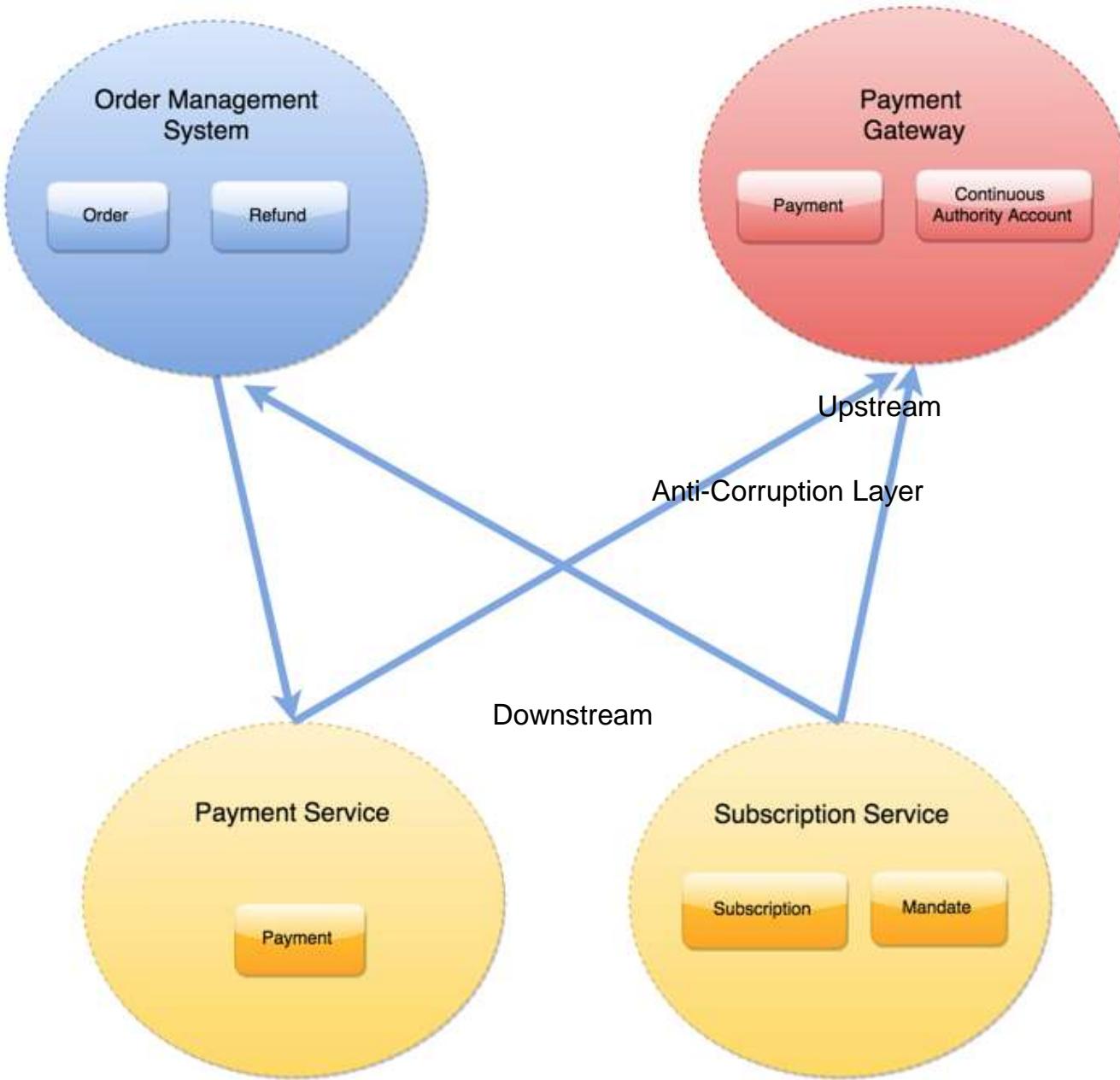
@gaythu\_rajan / @al94781

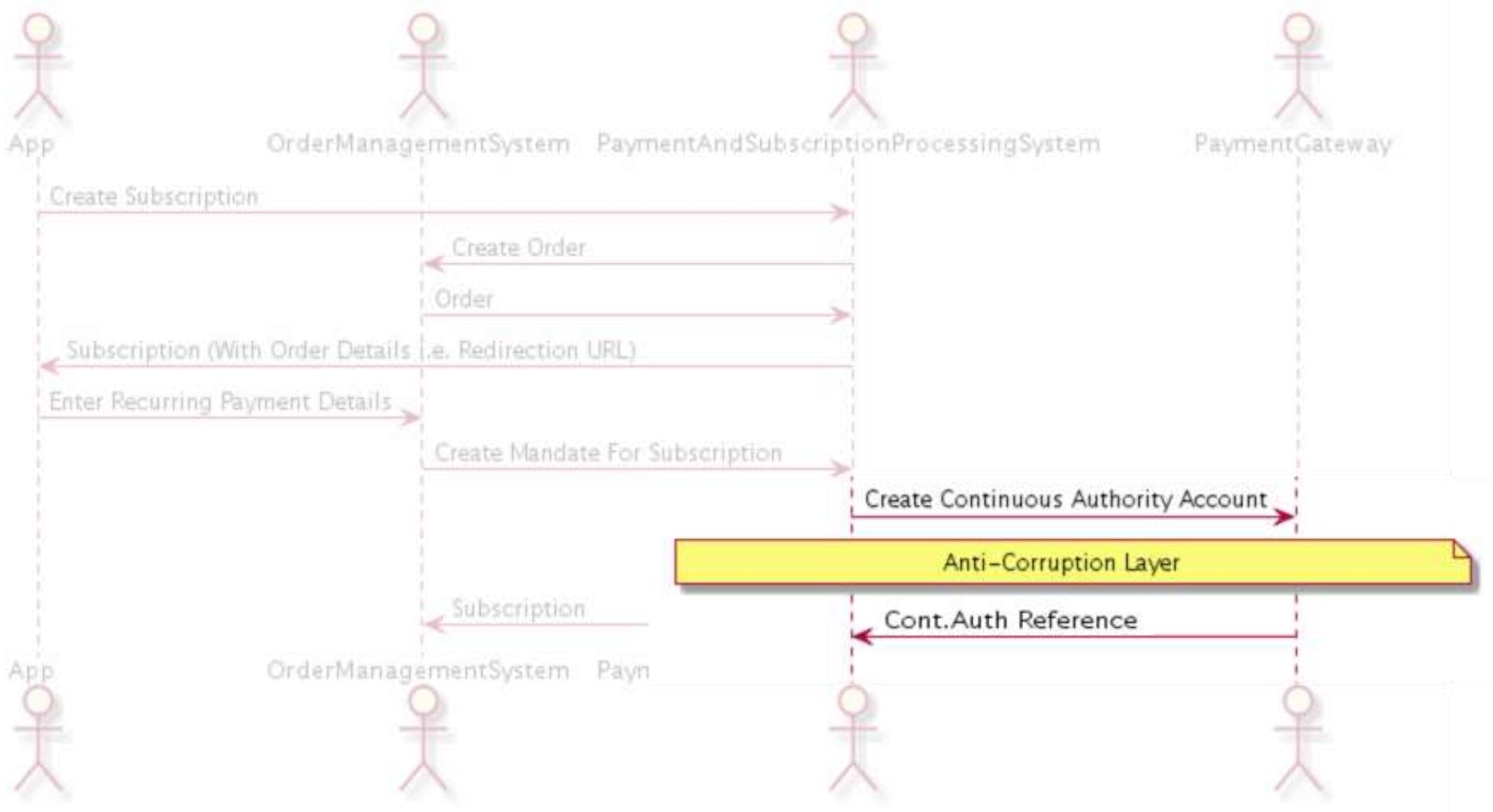




# Anti-Corruption Layer







# Done Done



cc: jayneandd - <https://www.flickr.com/photos/8180853@N07>

#harnessing\_ddd

@gaythu\_rajan / @al94781

# Take Home Messages

- Find the right Microservices - or you'll pine for the Monolith

cc: Manoj Kengudelu - <https://www.flickr.com/photos/11767573@N02>

# Take Home Messages

- Find the right Microservices - or you'll pine for the Monolith
- Don't get confused about BCs - they will fall out if you listen to the Language

cc: Manoj Kengudelu - <https://www.flickr.com/photos/11767573@N02>

# Take Home Messages

- Find the right Microservices - or you'll pine for the Monolith
- Don't get confused about BCs - they will fall out if you listen to the Language
- Organise around your services better - apply the Strategic Patterns

cc: Manoj Kengudelu - <https://www.flickr.com/photos/11767573@N02>

# Take Home Messages

- Find the right Microservices - or you'll pine for the Monolith
- Don't get confused about BCs - they will fall out if you listen to the Language
- Organise around your services better - apply the Strategic Patterns
- Be inspired to do more Domain Driven Design

cc: Manoj Kengudelu - <https://www.flickr.com/photos/11767573@N02>

# And Finally...



cc: oatsy40 - <https://www.flickr.com/photos/08089229@N06>

#harnessing\_ddd

@gaythu\_rajan / @al94781

# Questions Please



cc: Derek Bridges - <https://www.flickr.com/photos/84949728@N00>

Oh, and we (Capgemini ) are hiring!  
(visit [bit.ly/cg-jvm-jobs-ddd](https://bit.ly/cg-jvm-jobs-ddd))



cc: ptrlx - <https://www.flickr.com/photos/58615912@N05>

# Read More...

- <https://capgemini.github.io/architecture/microservices-reality-check/>
- <https://capgemini.github.io/architecture/why-microservices-are-right-for-us-pt1/>
- <https://capgemini.github.io/architecture/microservices-like-change/>
- <https://capgemini.github.io/architecture/microservices-gotchas/>

cc: Manoj Kengudelu - <https://www.flickr.com/photos/11767573@N02>