

Microservices





Somkiat Puisungnoen

Search

Somkiat | Home

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Intro

Software Craftsmanship

Software Practitioner at สยามชานาญกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Somkiat Puisungnoen 15 mins · Bangkok · ...

Java and Bigdata

When did you work at Opendream? X

... 22 Pending Items

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Intro

Software Craftsmanship

Software Practitioner at สยามชานาญกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Somkiat Puisungnoen 15 mins · Bangkok · ...

Java and Bigdata



Facebook somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help ▾

somkiat.cc
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ... + Add a Button

Help people take action on this Page. X



Agenda

Cloud Native Application
Microservices and DevOps
The architecture of microservices
How to model microservices
Integrating multiple microservices
Design and develop microservices



Agenda

Testing and Developing microservices
Deploying microservices
Maintaining healthy microservices
Monitoring microservices
Scaling up your microservices



<https://github.com/up1/course-microservice>



Customers



“The Business”



Product Teams

Platform Teams

Infrastructure Teams

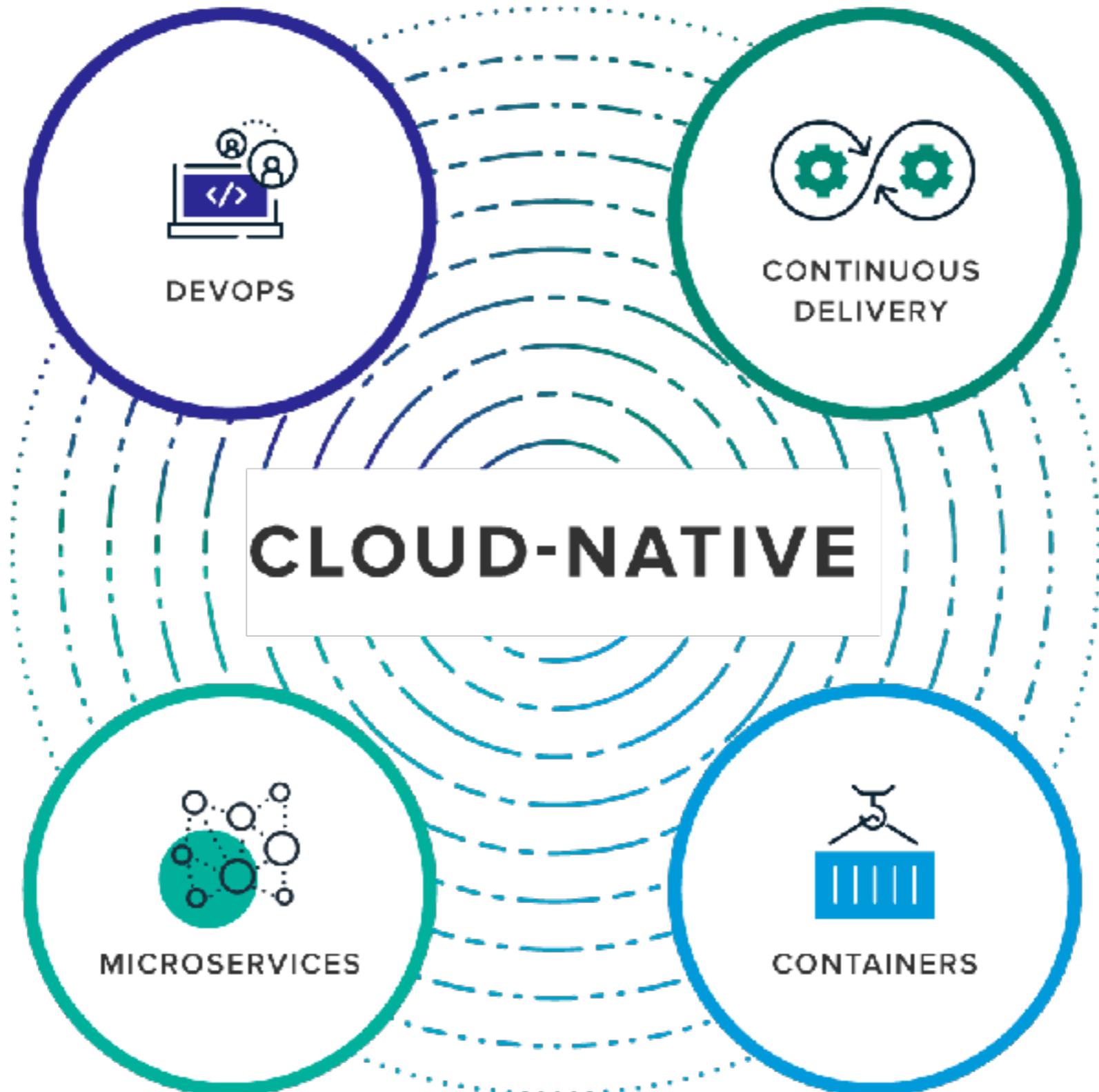
Operations Teams



Google/Amazon

<https://bravenewgeek.com/>





<https://pivotal.io/cloud-native>

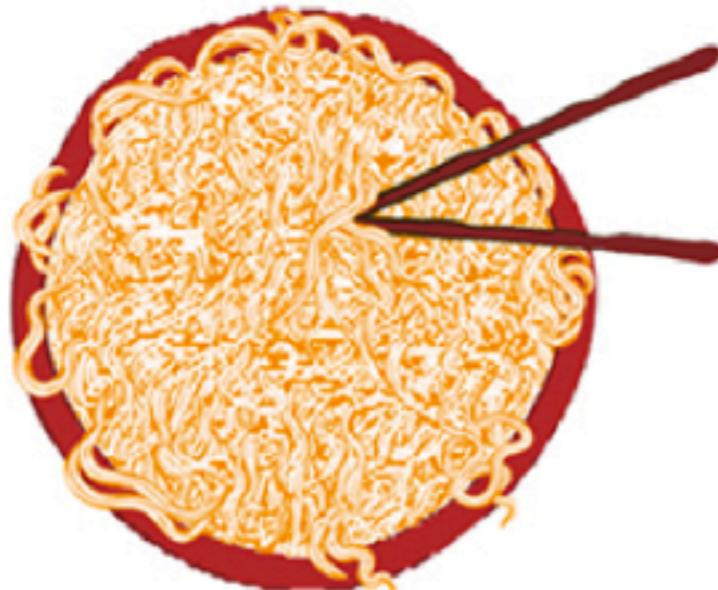


Evolution of Architecture



1990s and earlier

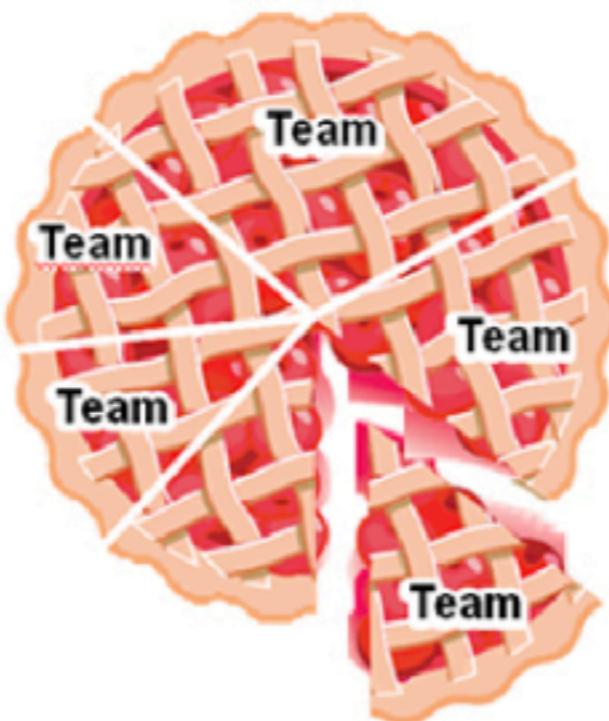
Pre-SOA (monolithic)
Tight coupling



For a monolith to change, all must agree on each change. Each change has unanticipated effects requiring careful testing beforehand.

2000s

Traditional SOA
Looser coupling



Elements in SOA are developed more autonomously but must be coordinated with others to fit into the overall design.

2010s

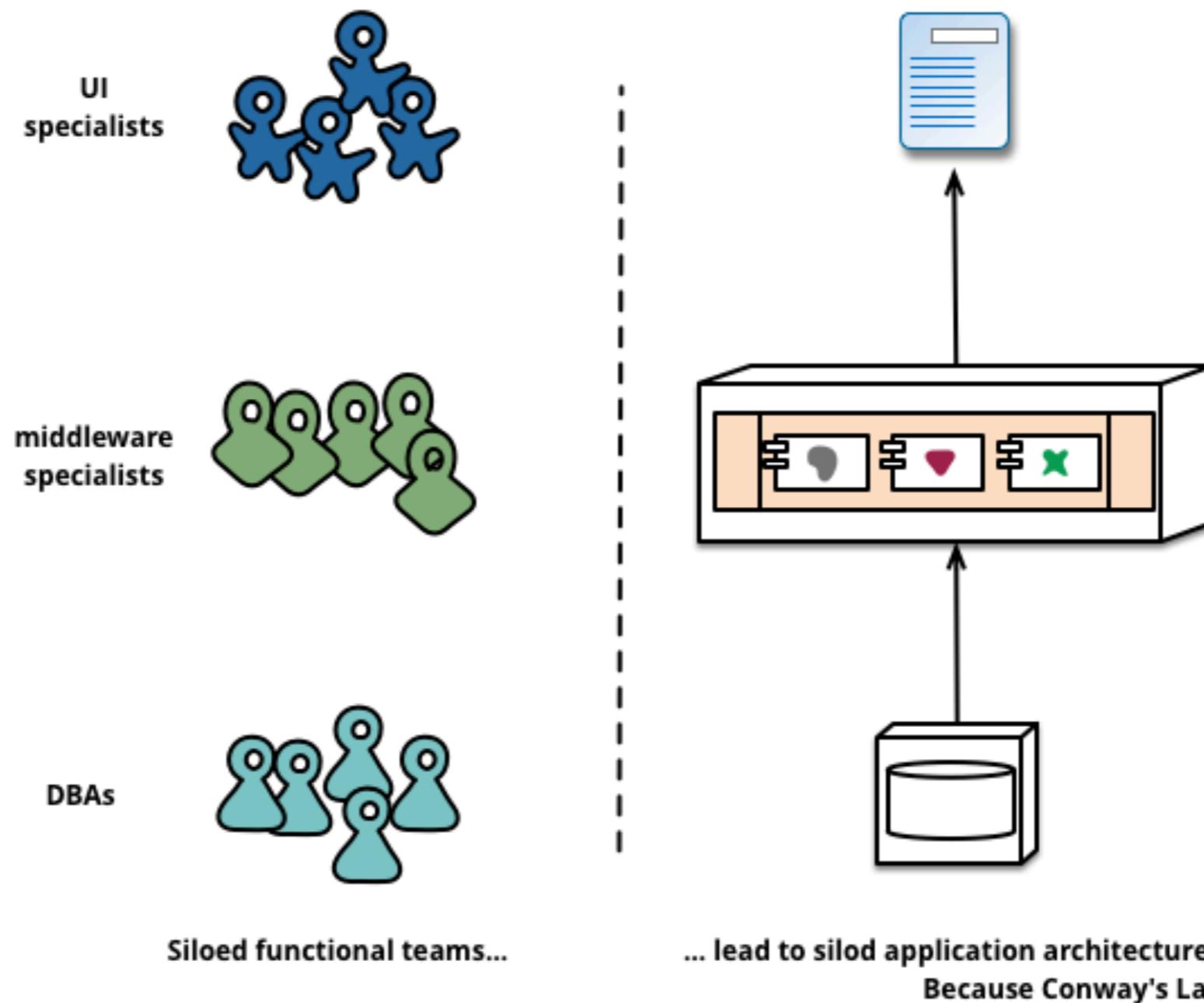
Microservices
Decoupled



Developers can create and activate new microservices without prior coordination with others. Their adherence to MSA principles makes continuous delivery of new or modified services possible.



Conway's Law



Microservices

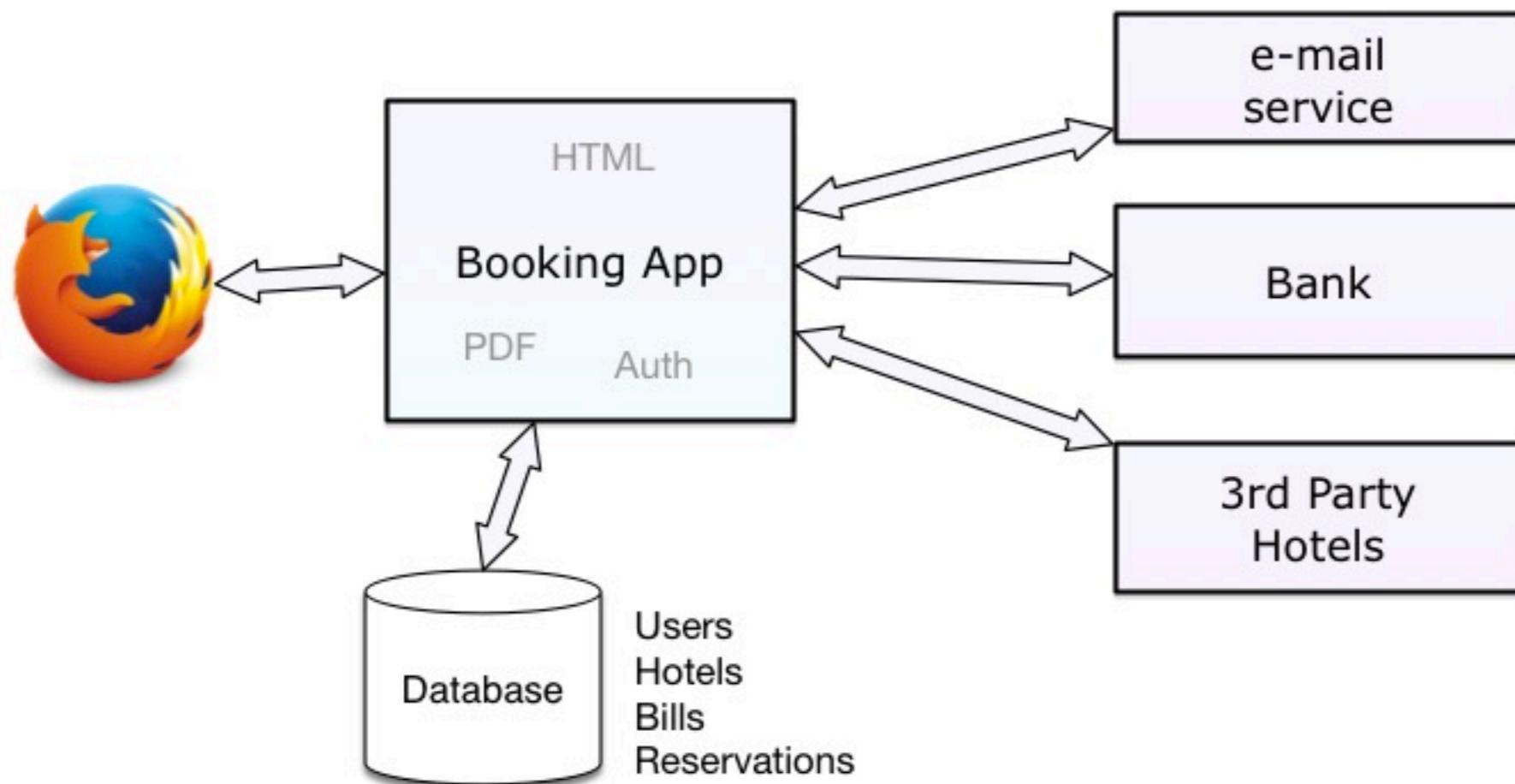


Microservices

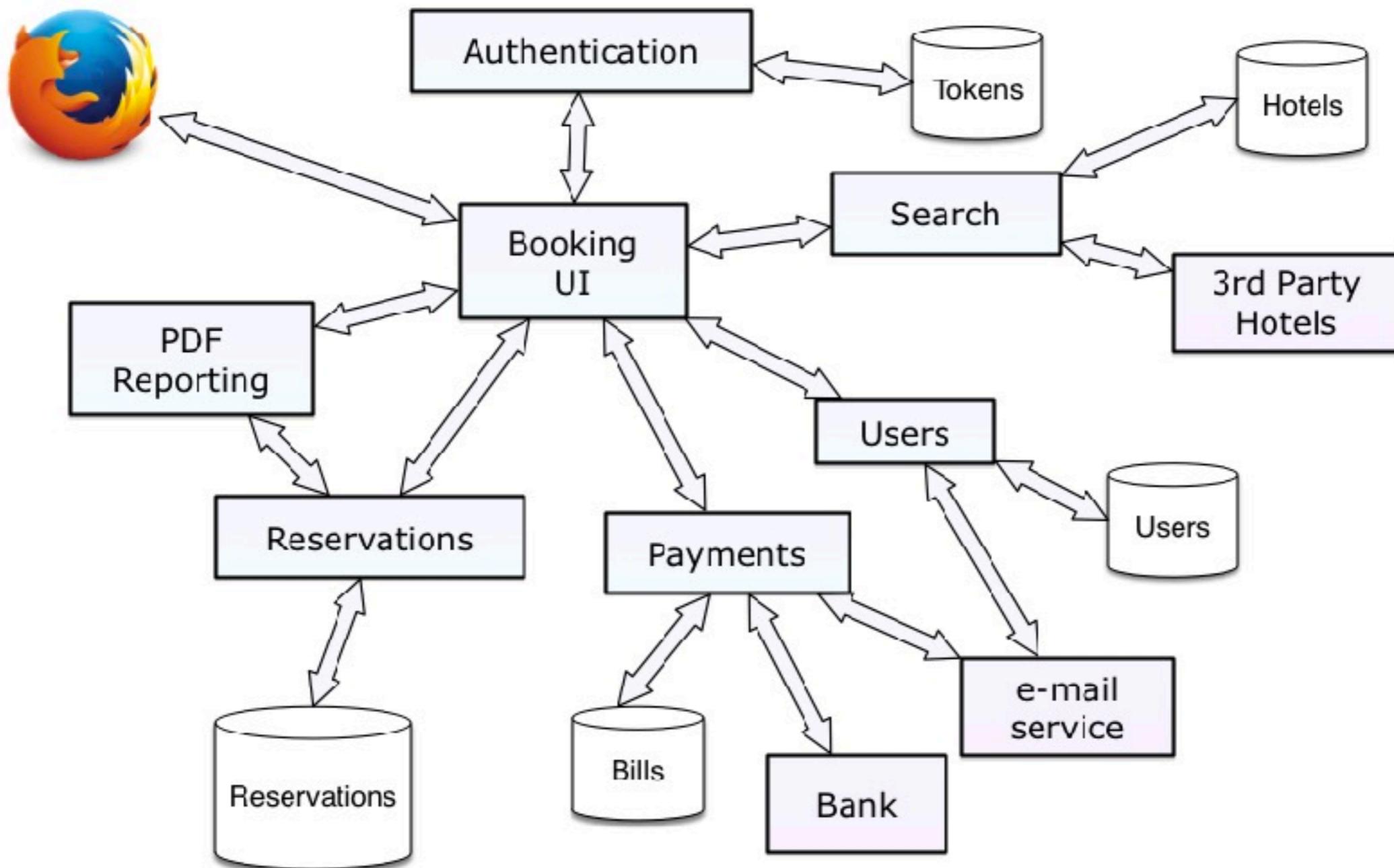
Small, Do one thing
Modular
Easy to deploy
Scale independently



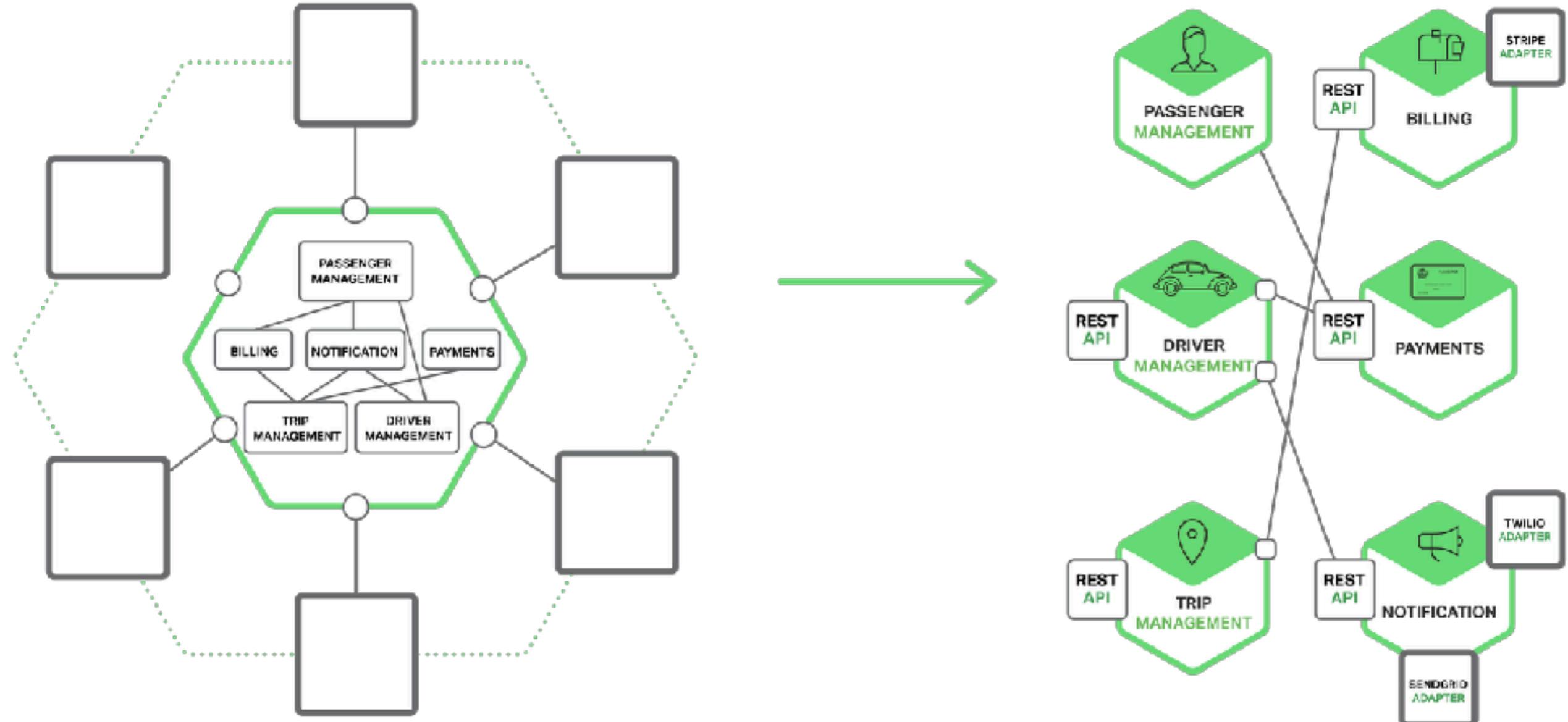
Monolithic



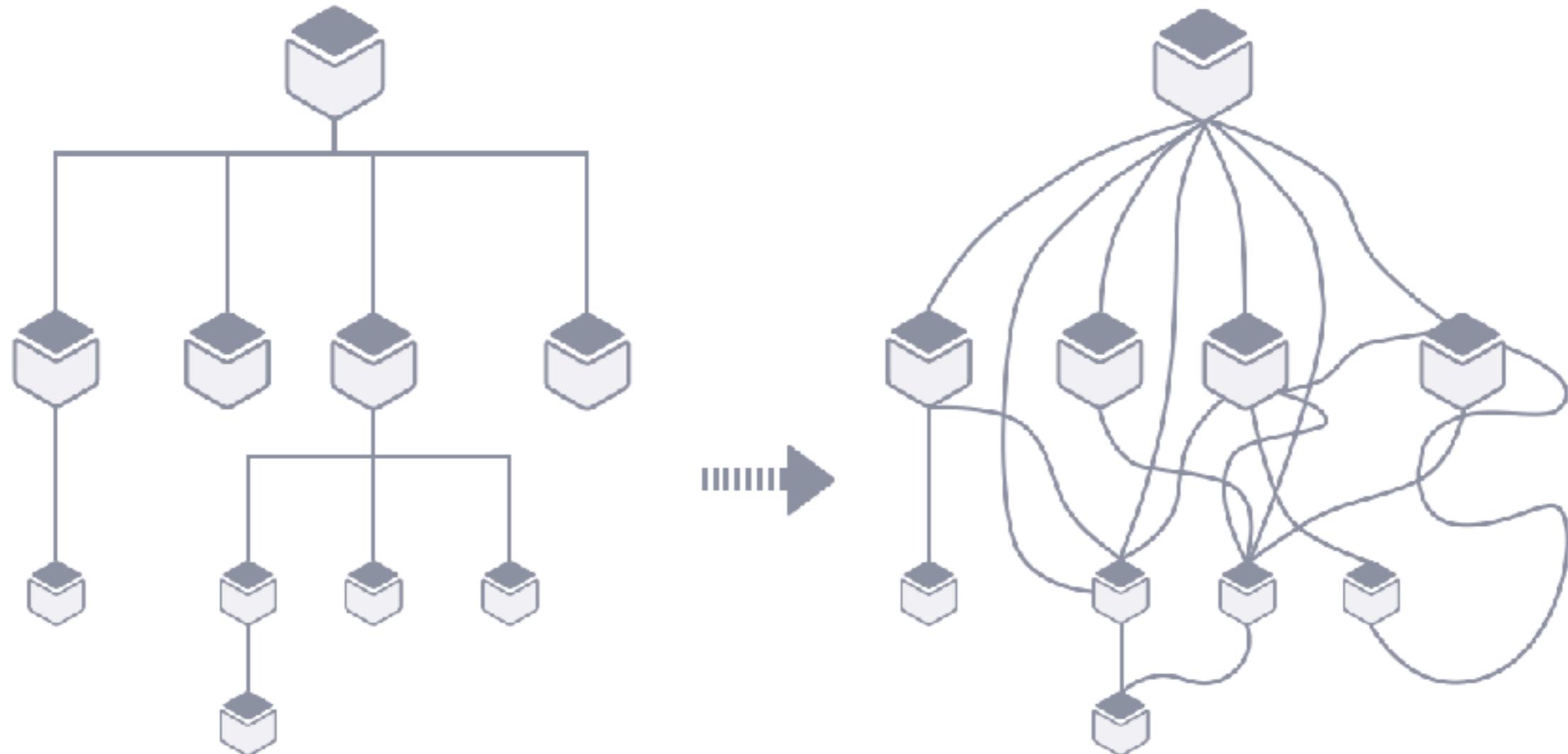
Microservices



Conway's Law

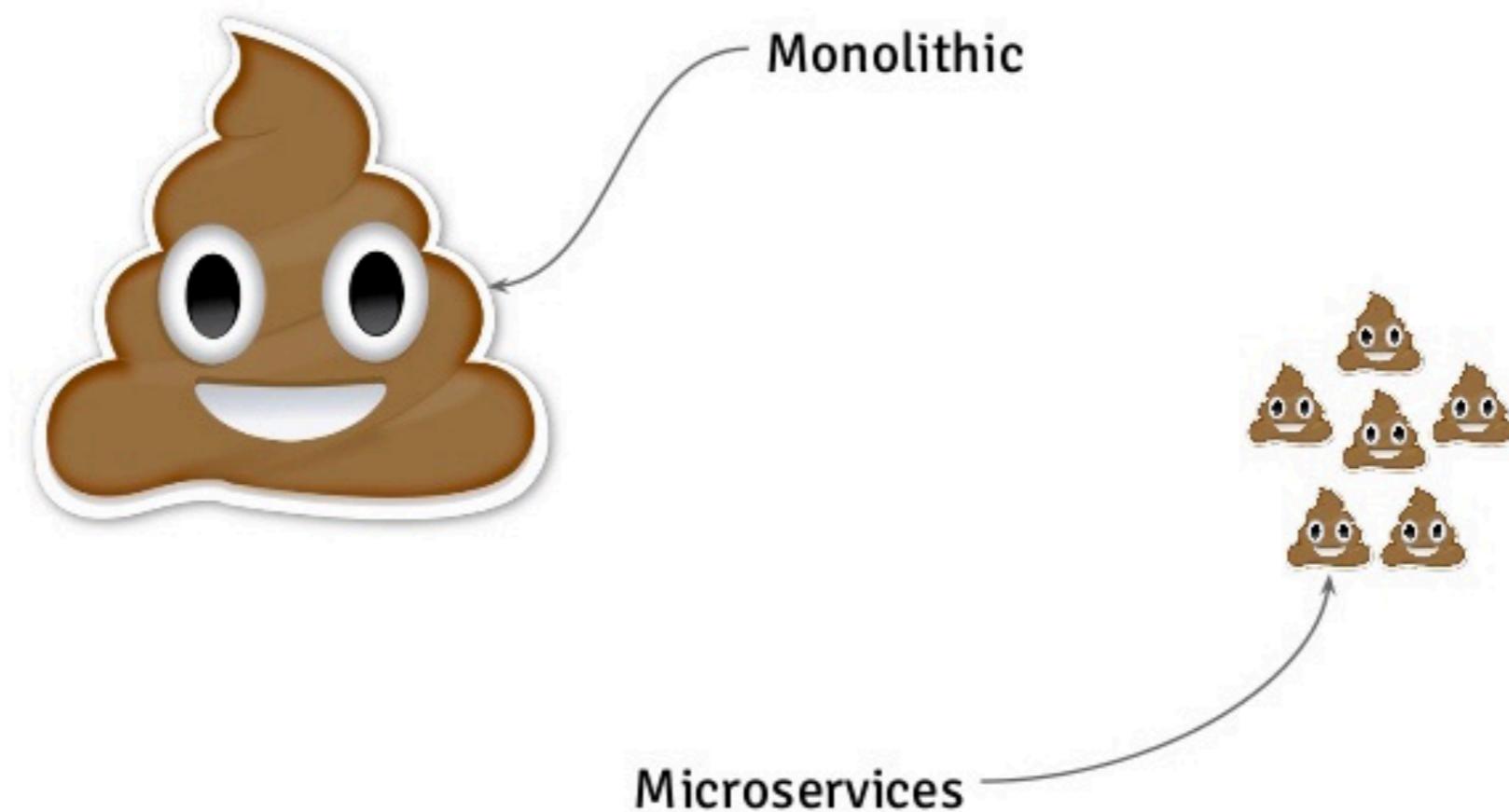


Microservices spaghetti



Microservices spaghetti

Monolithic vs Microservices

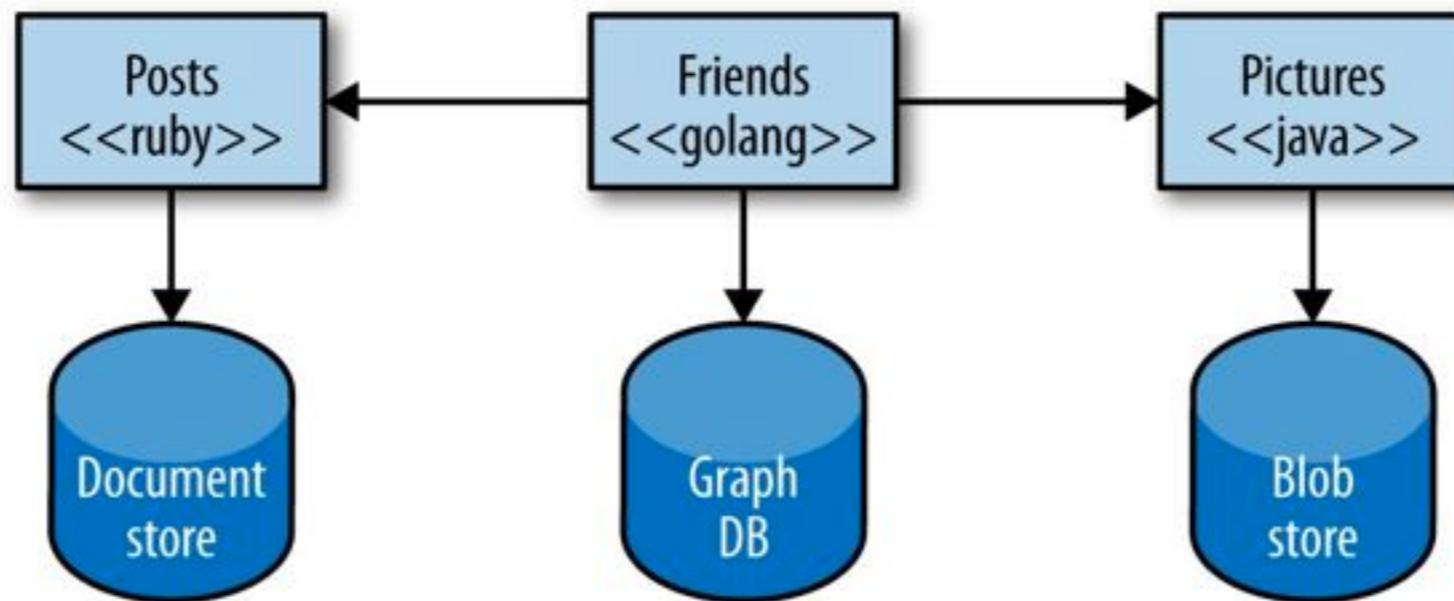


Key Benefits

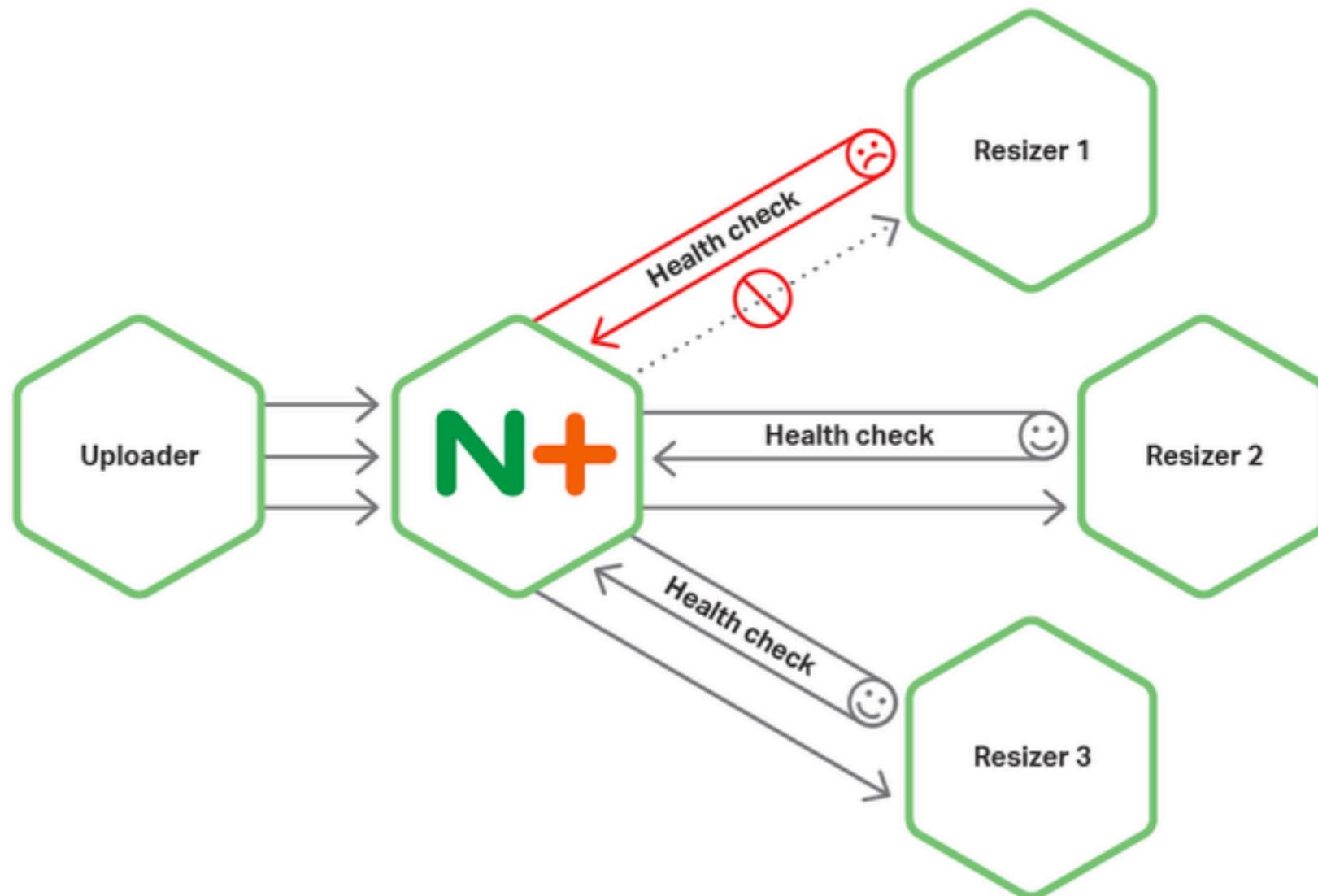


1. Technology heterogeneity

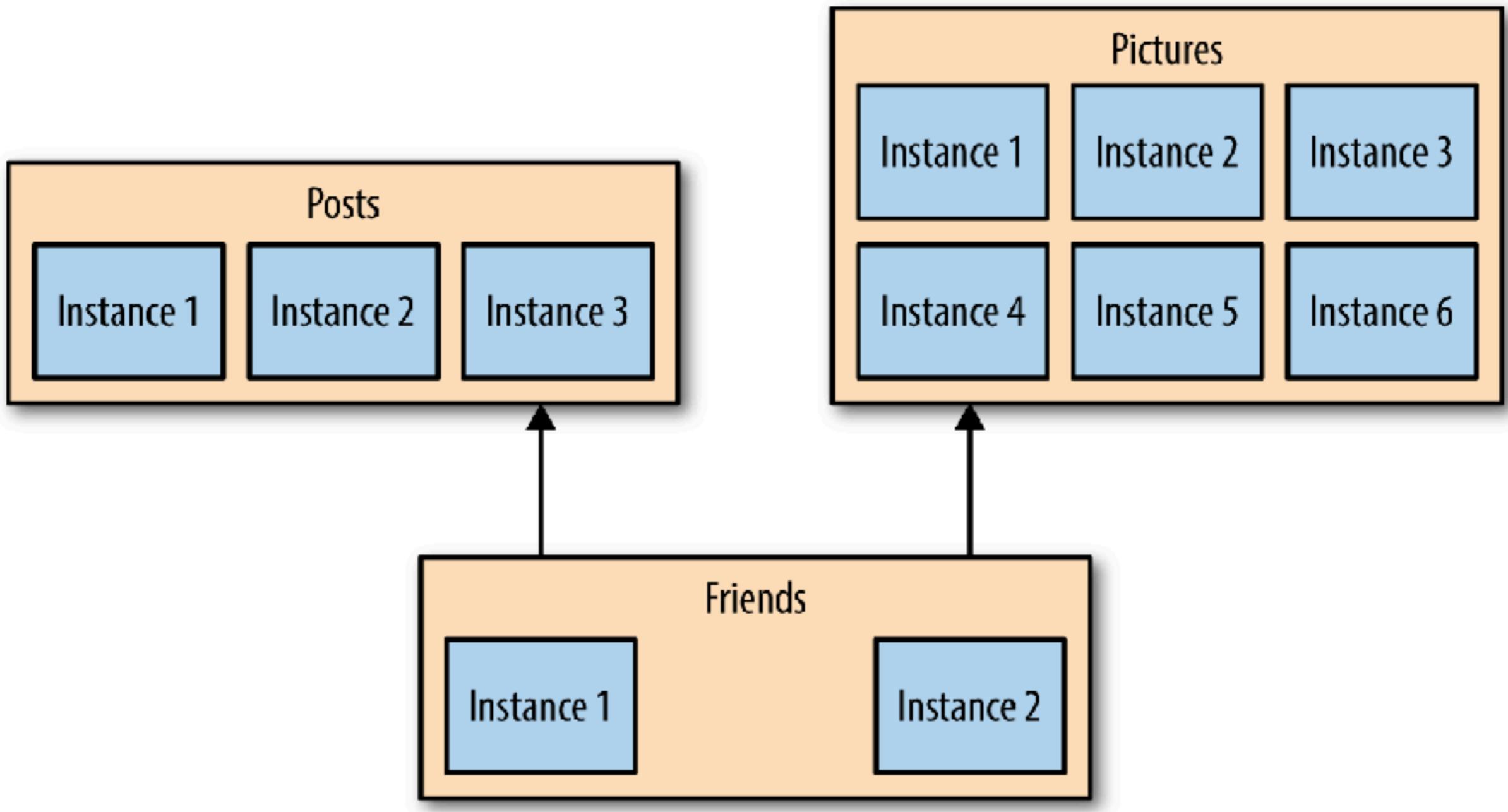
The right tool for each job



2. Resilience

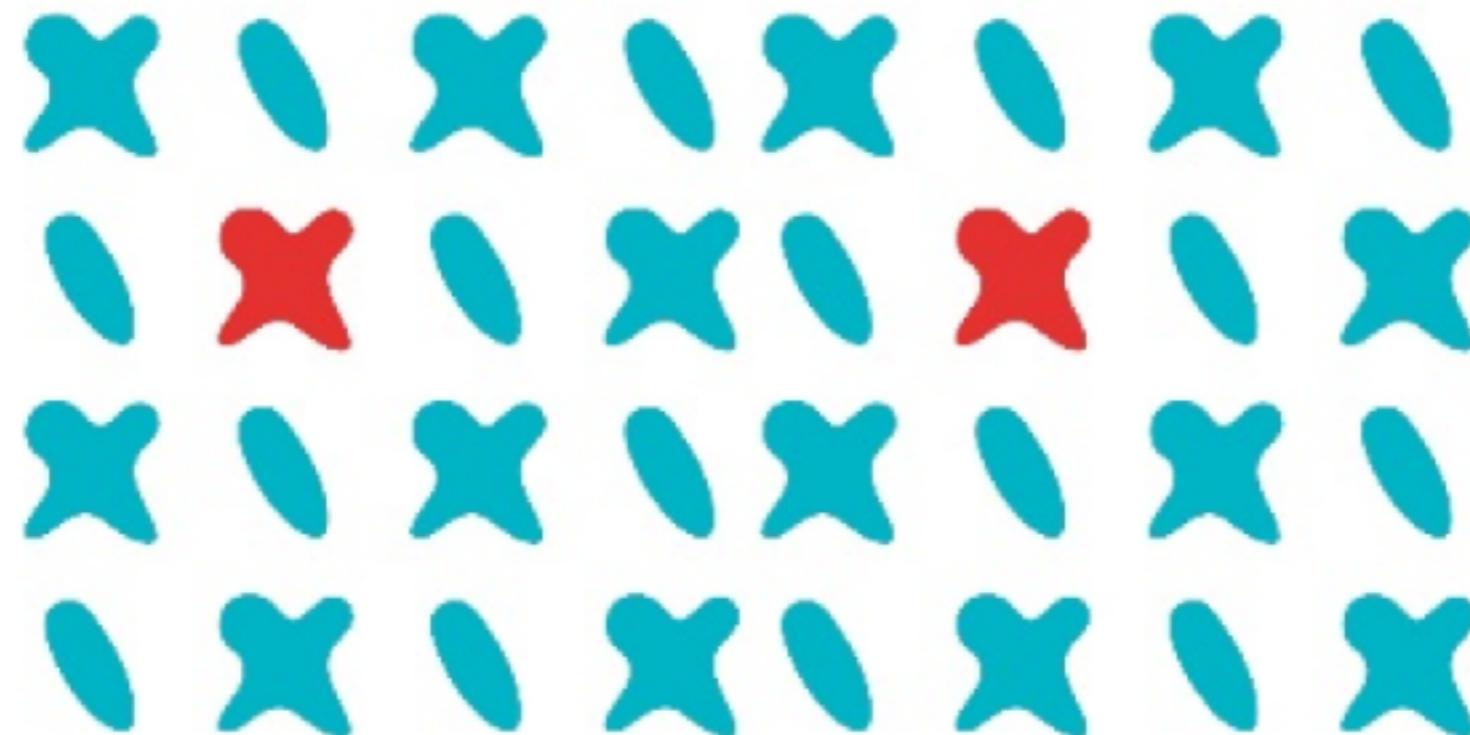


3. Scaling



4. Ease of deployment

Deploys are faster, independent and problems can be isolated more easily

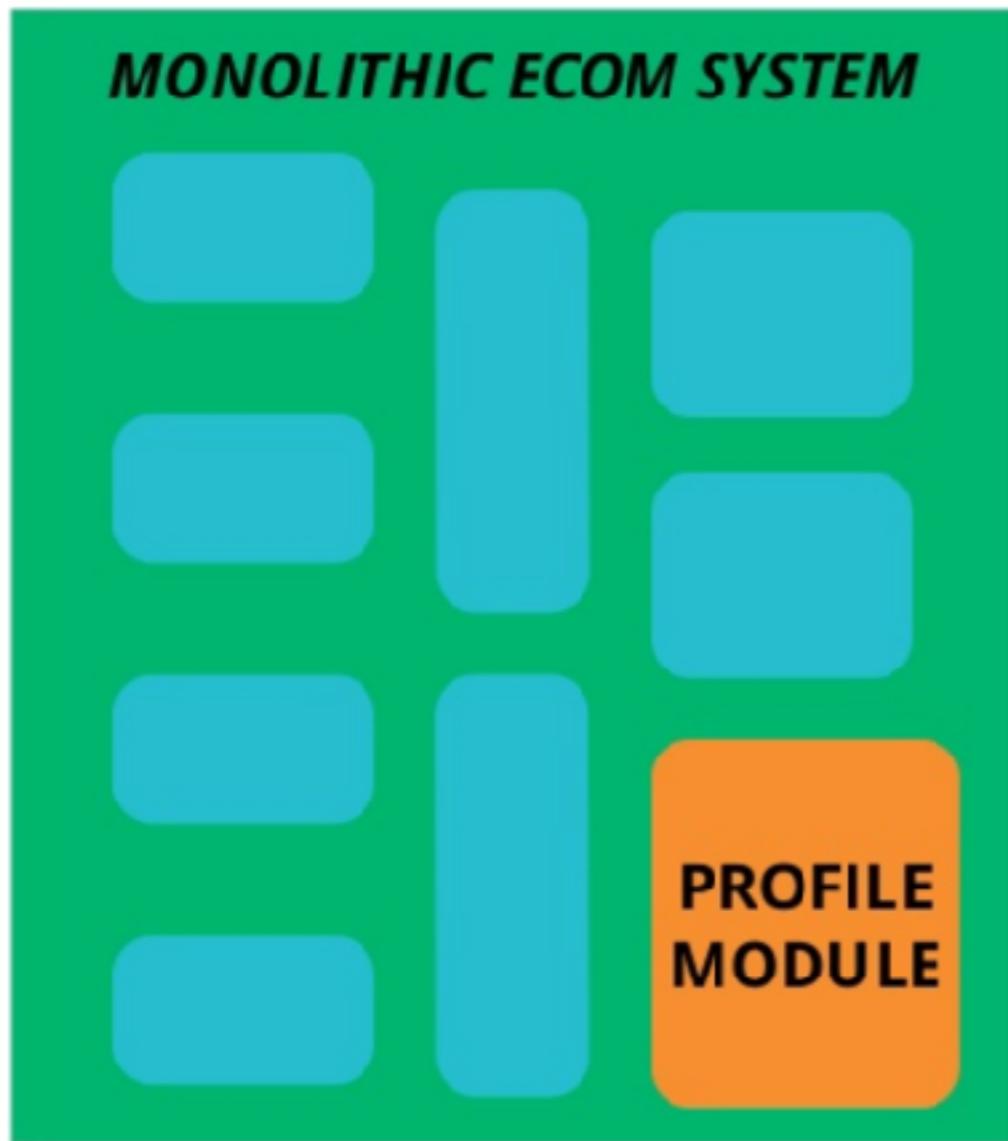


5. Organization alignment

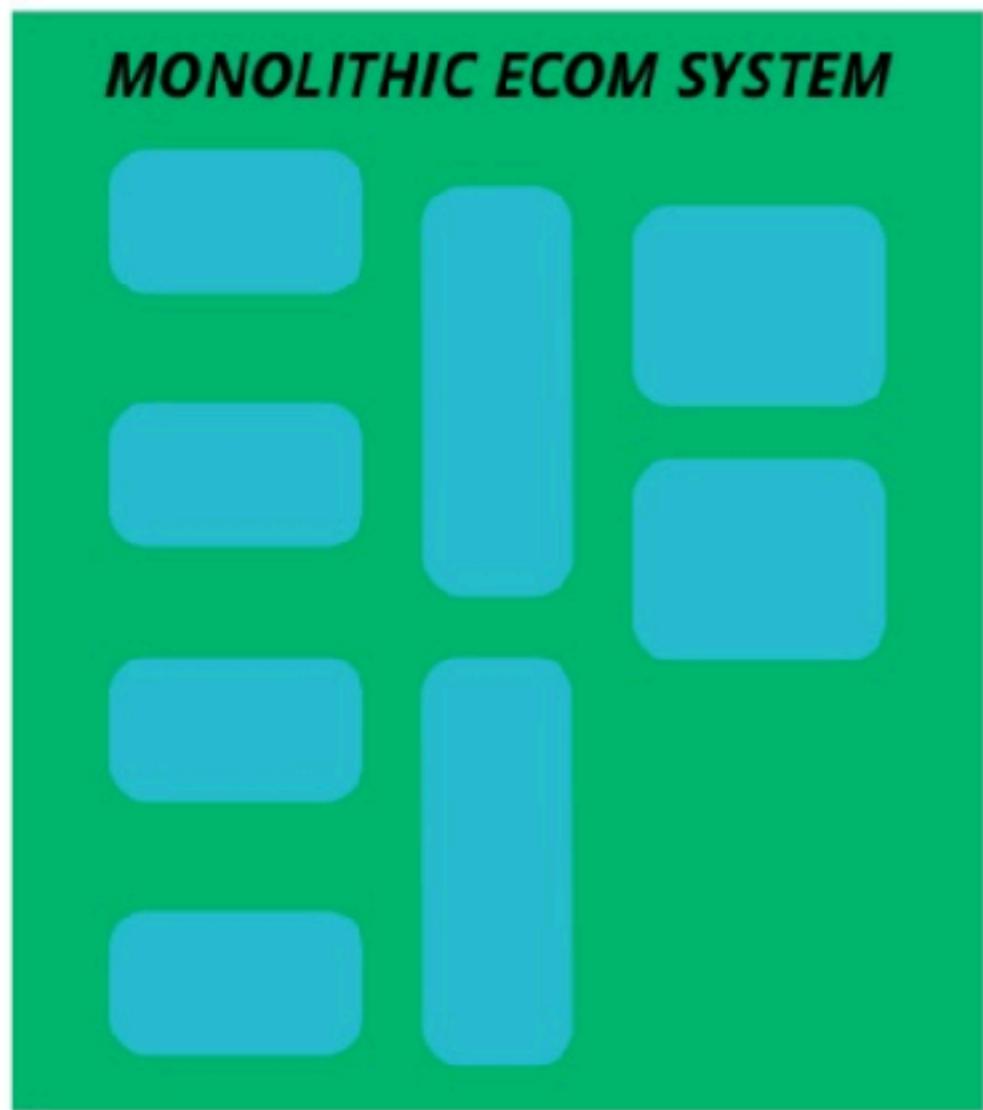
Small teams and smaller codebases



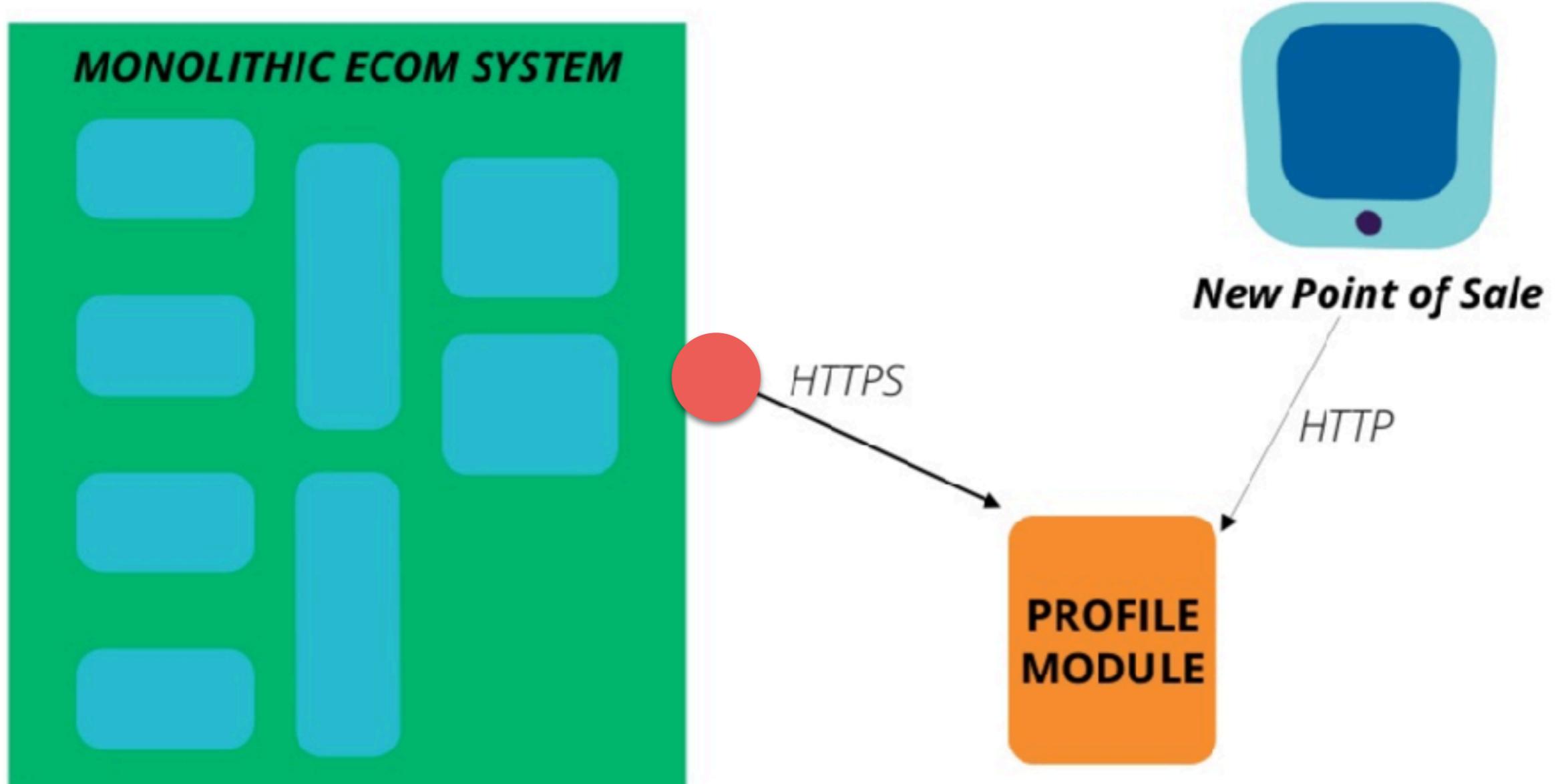
6. Composability and replaceability



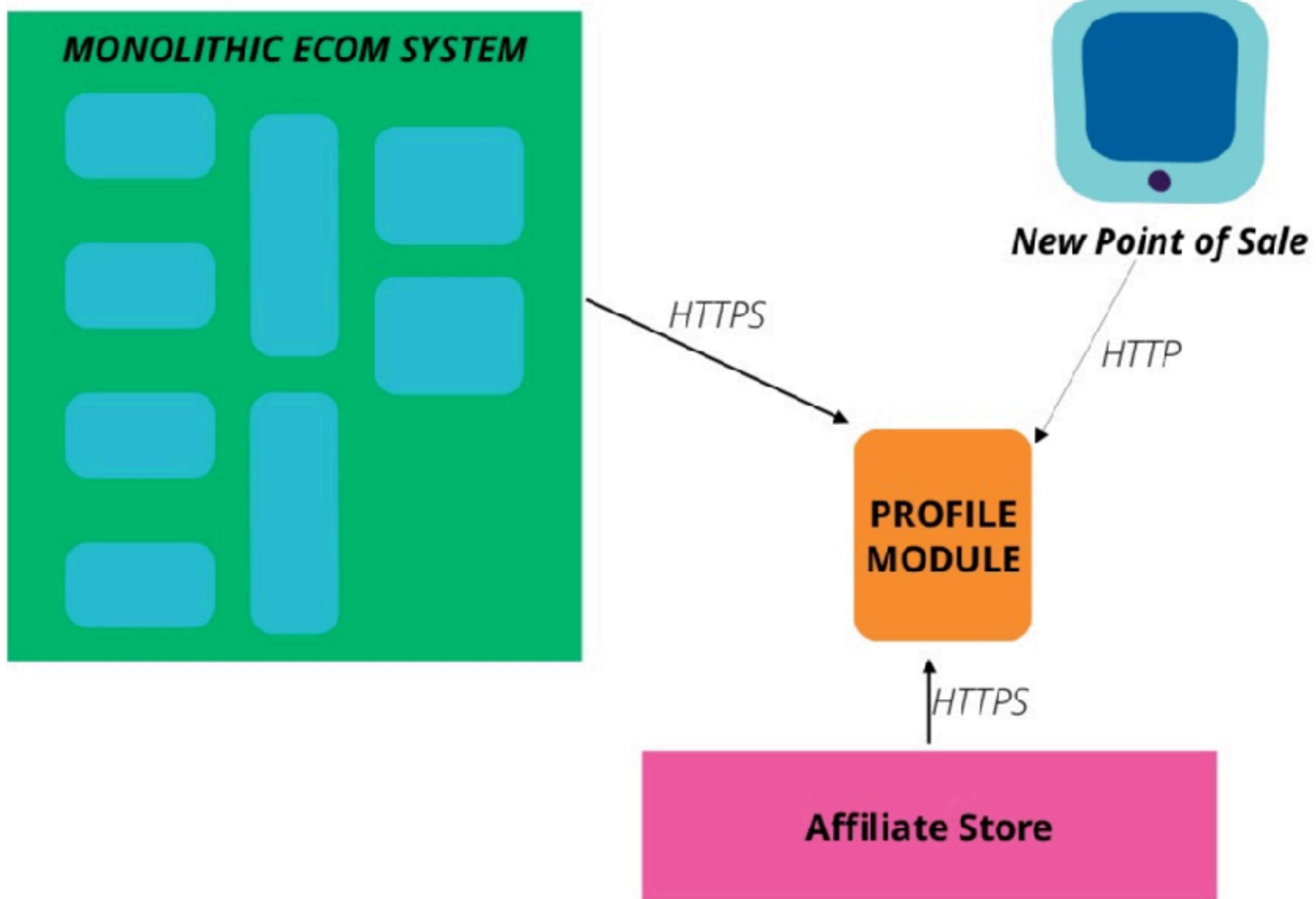
6. Composability and replaceability



6. Composability and replaceability



6. Composability and replaceability



Characteristics



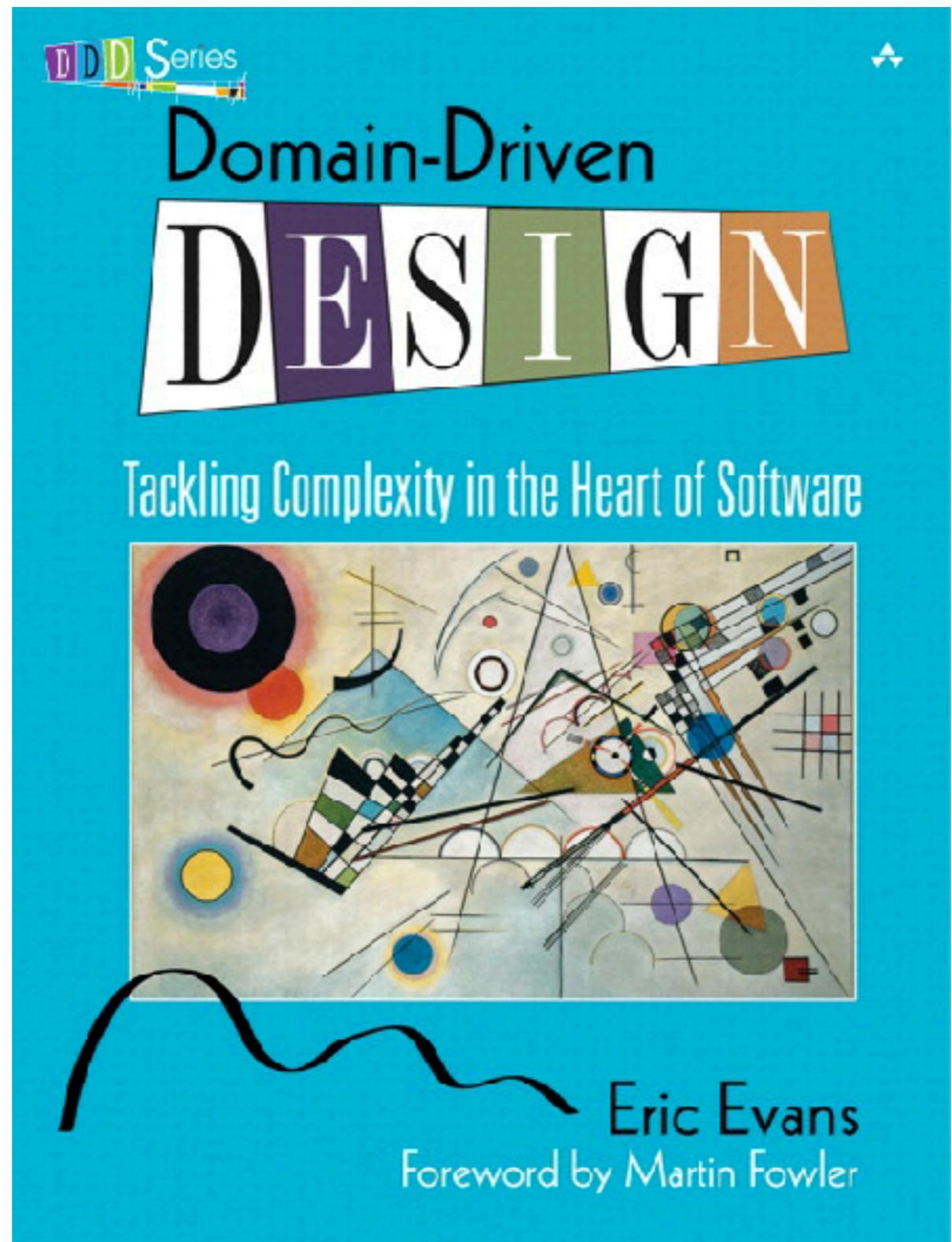
1. Responsible for a single capability



Types of capabilities

Business capability
Technical capability





2. Individually deployable



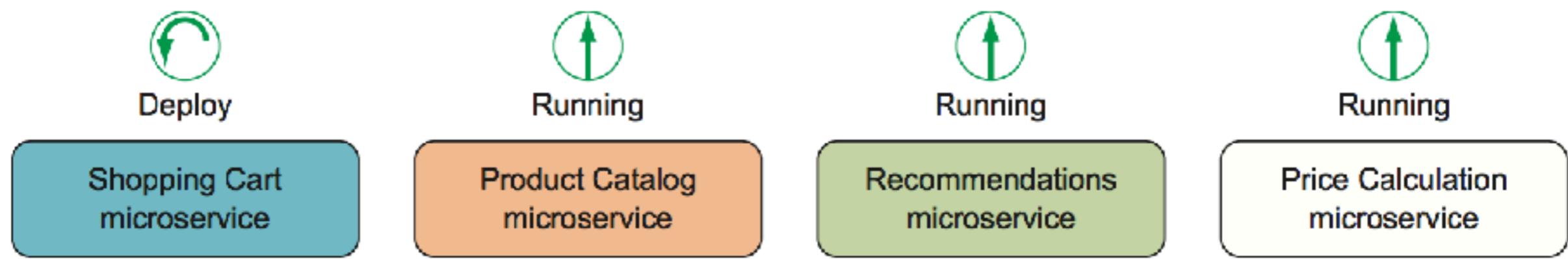


Figure 1.2 Other microservices continue to run while the Shopping Cart microservice is being deployed.



3. Consists of one or more processes



**Problematic process boundary.
Microservices should run in separate
processes to avoid coupling.**

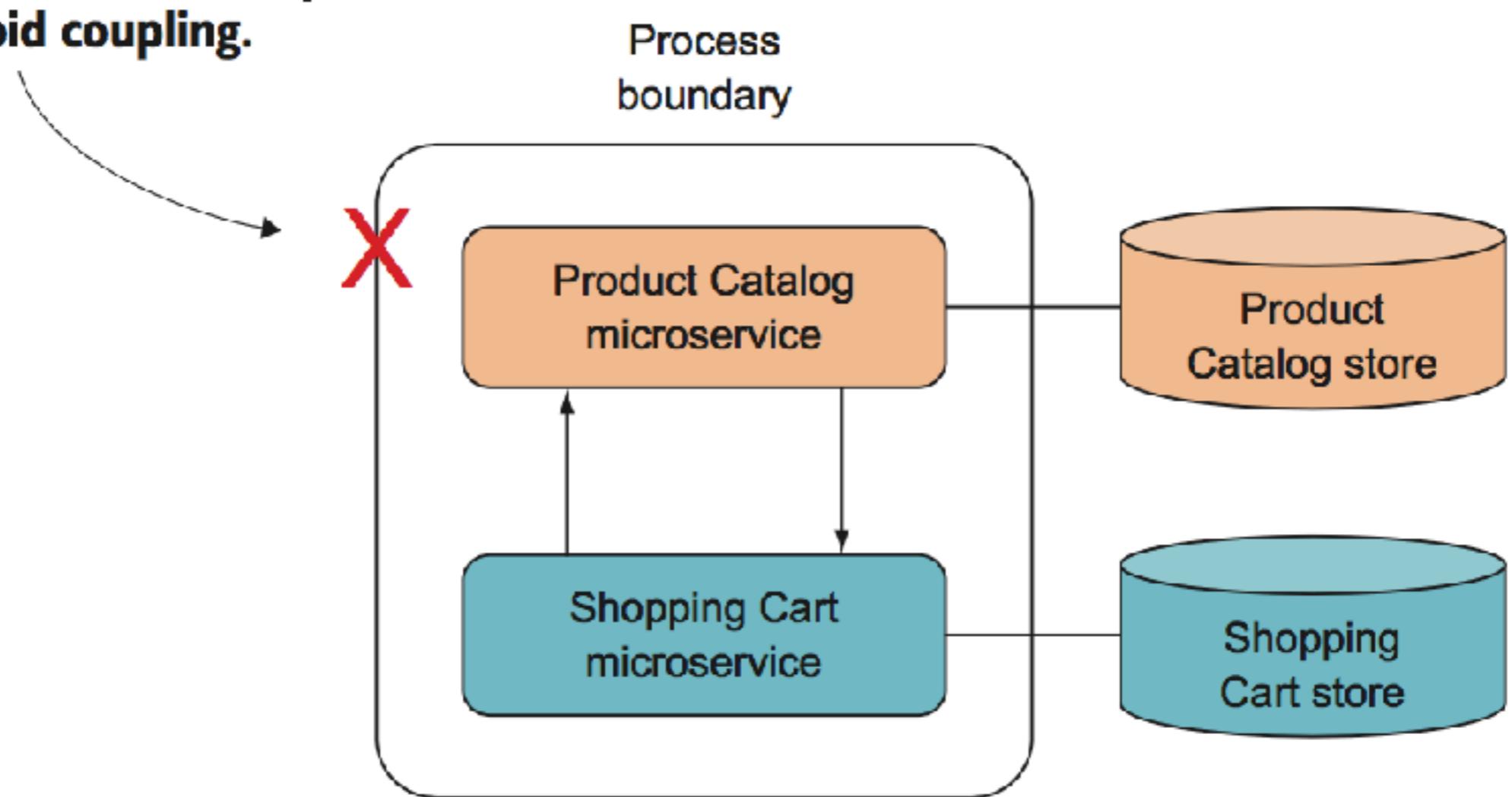


Figure 1.3 Running more than one microservice within a process leads to high coupling.



4. Own data store



All communication with the Product Catalog microservice must go through the public API.

Direct access to the Product Catalog store is not allowed. The Product Catalog microservice owns the Product Catalog store.

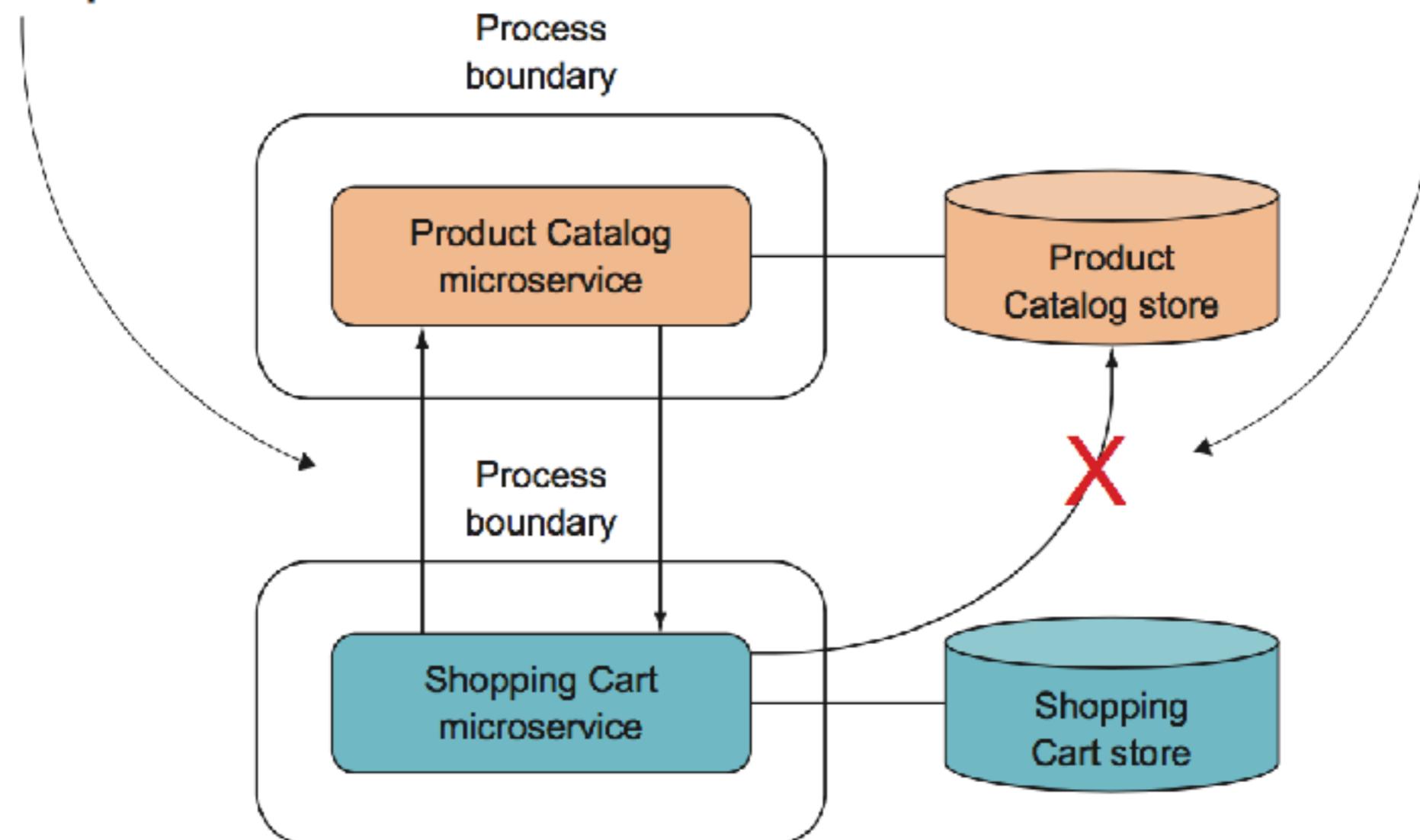
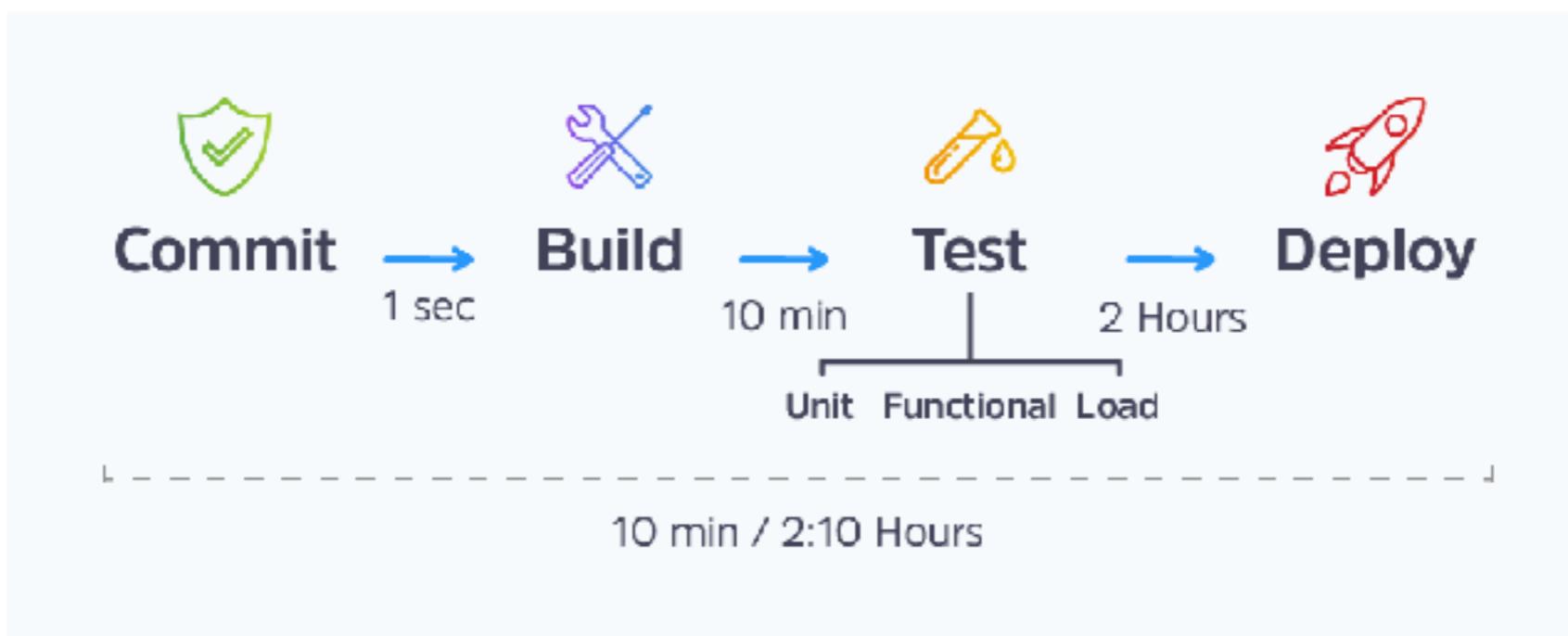


Figure 1.4 One microservice can't access another's data store.



5. Small team can maintain



Challenges with Microservices ?

<https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/architect-microservice-container-applications/distributed-data-management>



Microservices

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

1. How to define the boundaries of each microservices ?



Premature splitting is the root of
all evil.



Every time you make the decision
to split out a new microservice,
there's a **risk** of ending up with a
bloated app.



Why we need to split service ?



Reasons to split ?

Pace of change
Team structure
Security
Technology

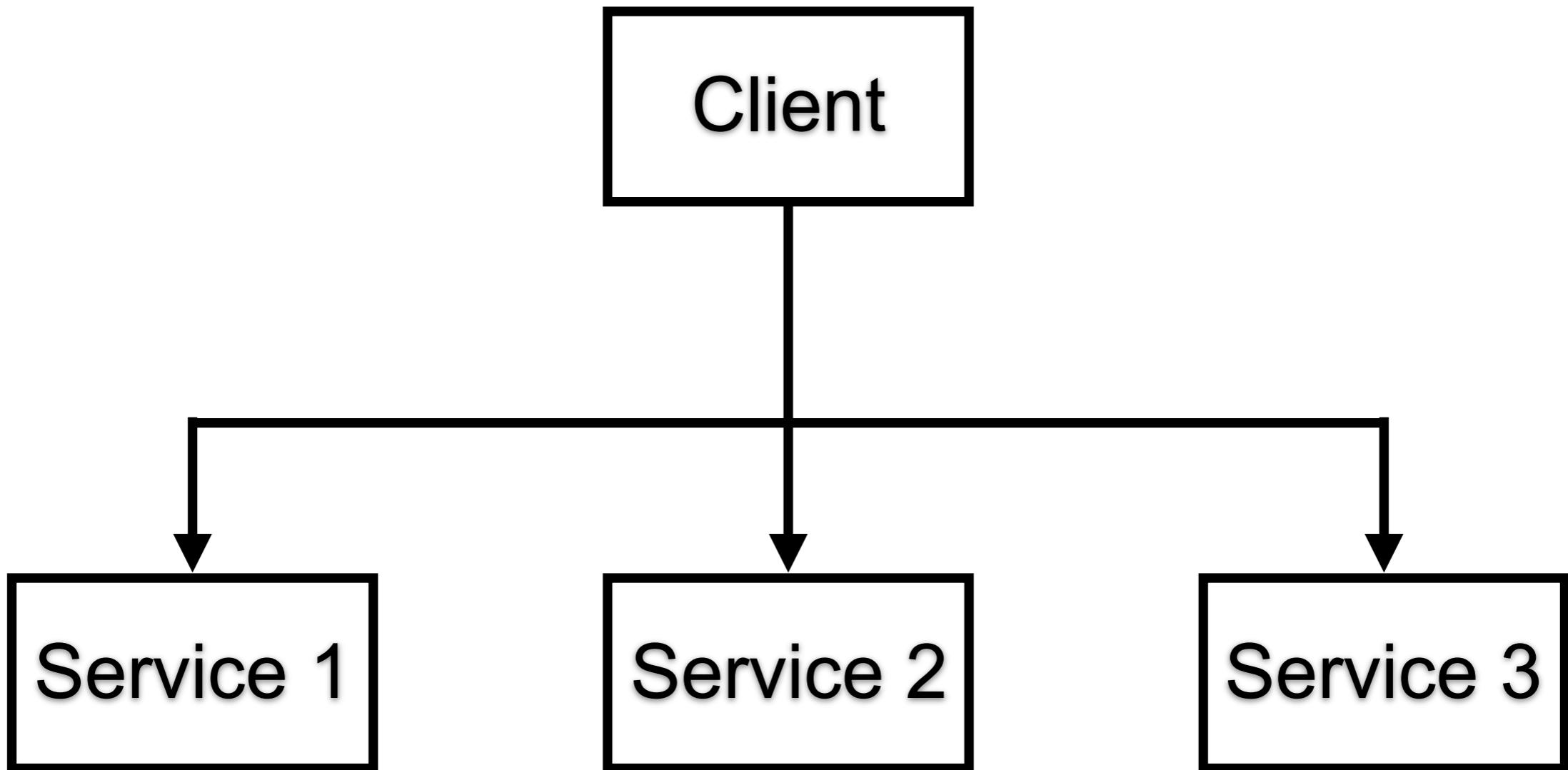


Start with a good service



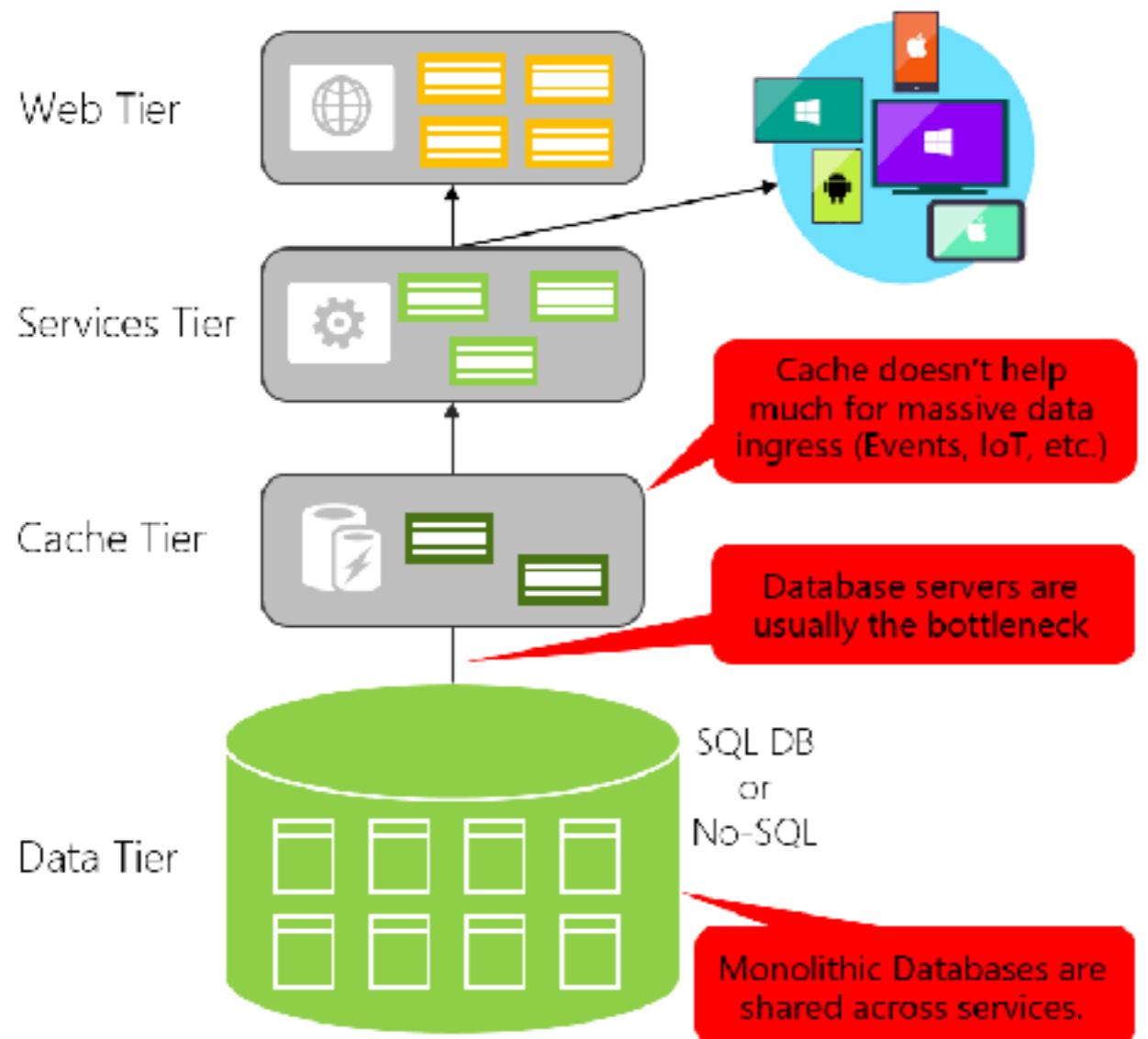
2. How to create queries that retrieve data from several microservices ?





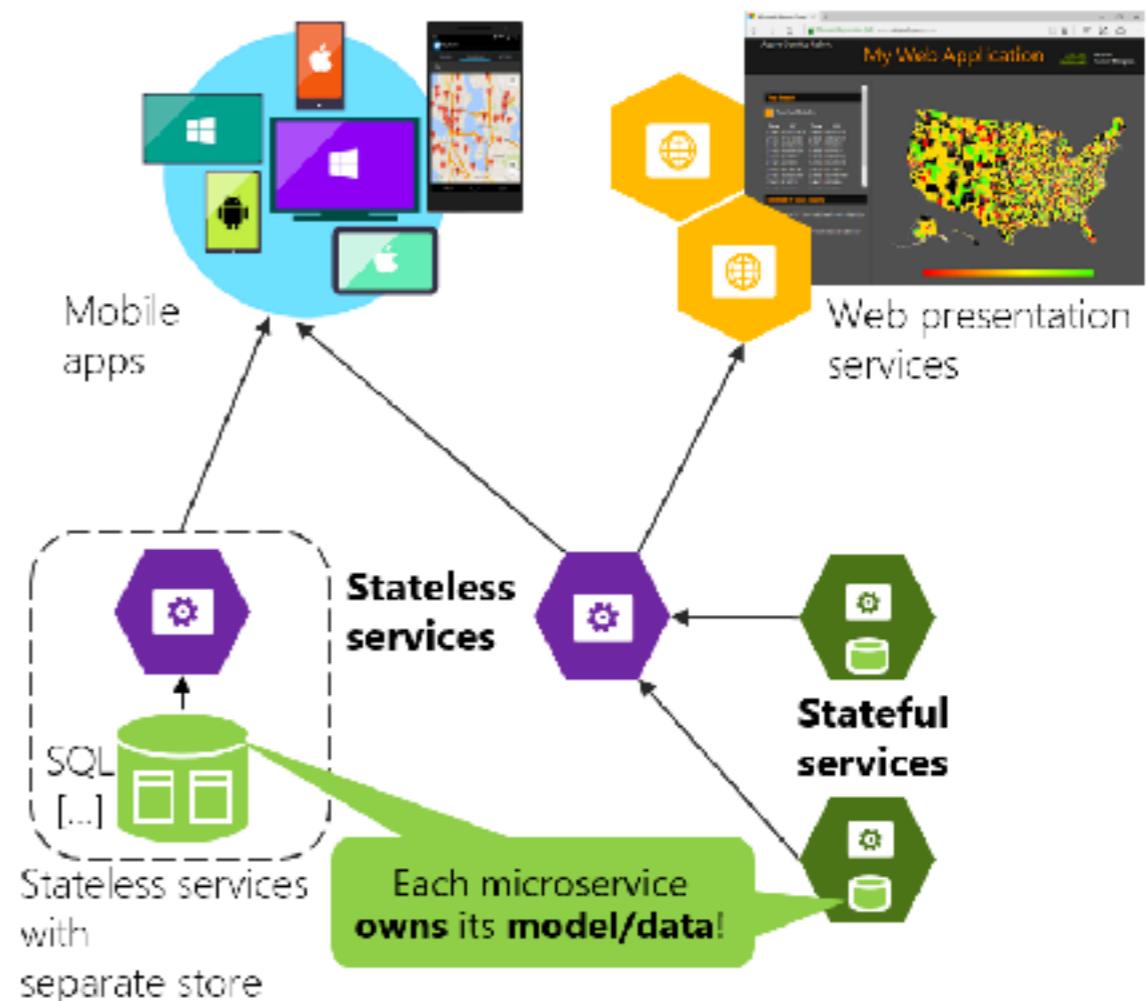
Data in Traditional approach

- Single monolithic database
- Tiers of specific technologies



Data in Microservices approach

- Graph of interconnected microservices
- State typically scoped to the microservice
- Remote Storage for cold data



<https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/architect-microservice-container-applications/data-sovereignty-per-microservice>



Popular solutions

API Gateway

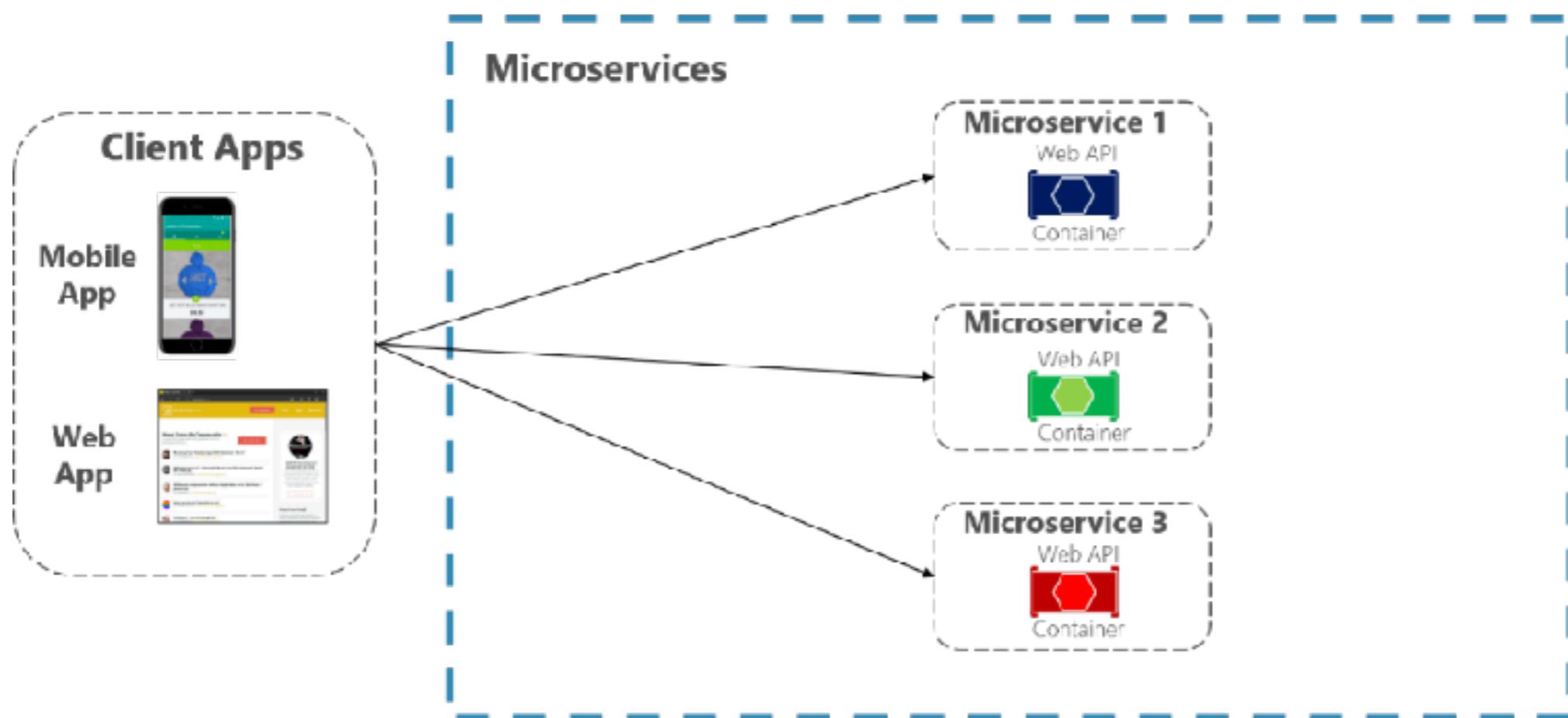
Cold data in central database

CQRS with query/read tables



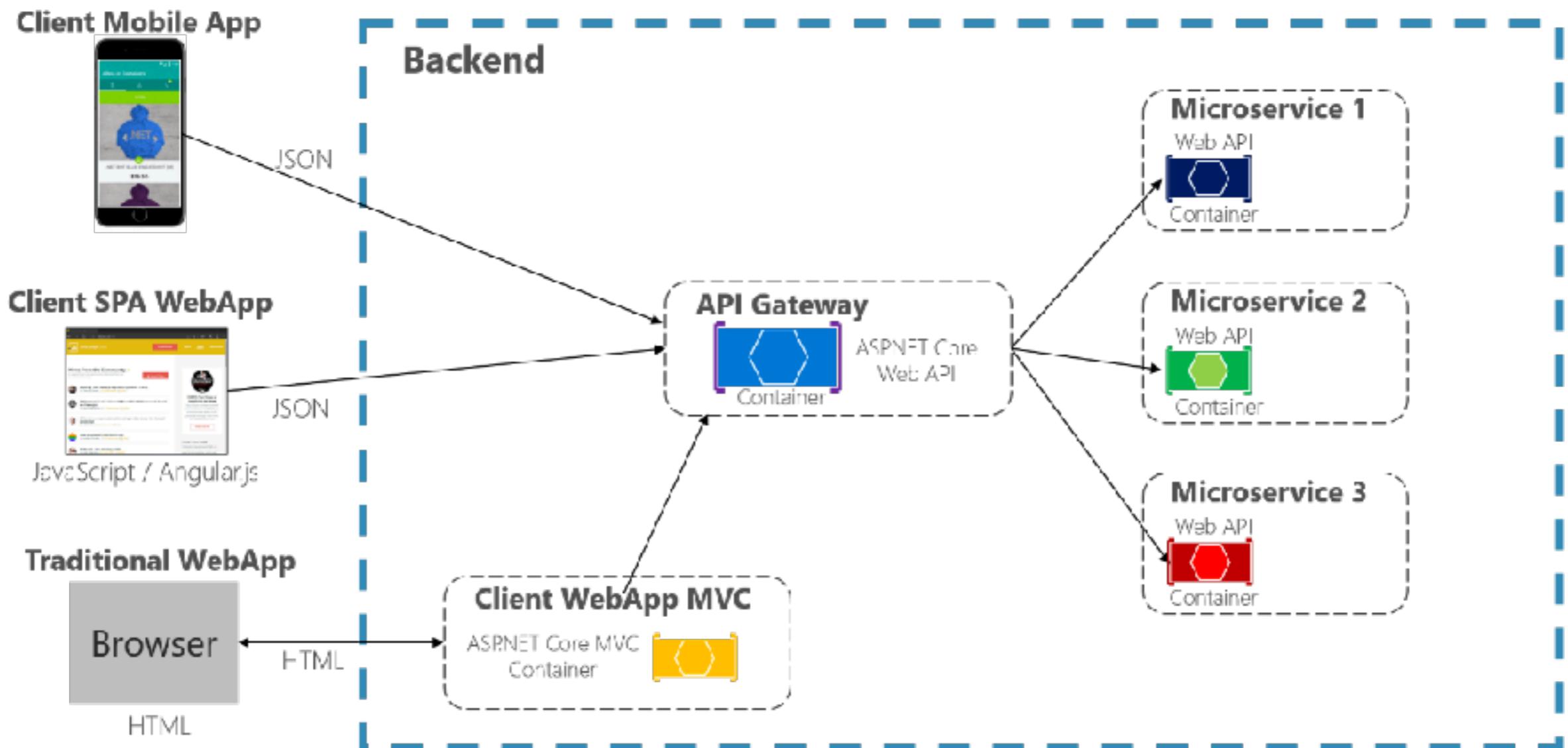
API Gateway

Direct Client-To-Microservice communication Architecture

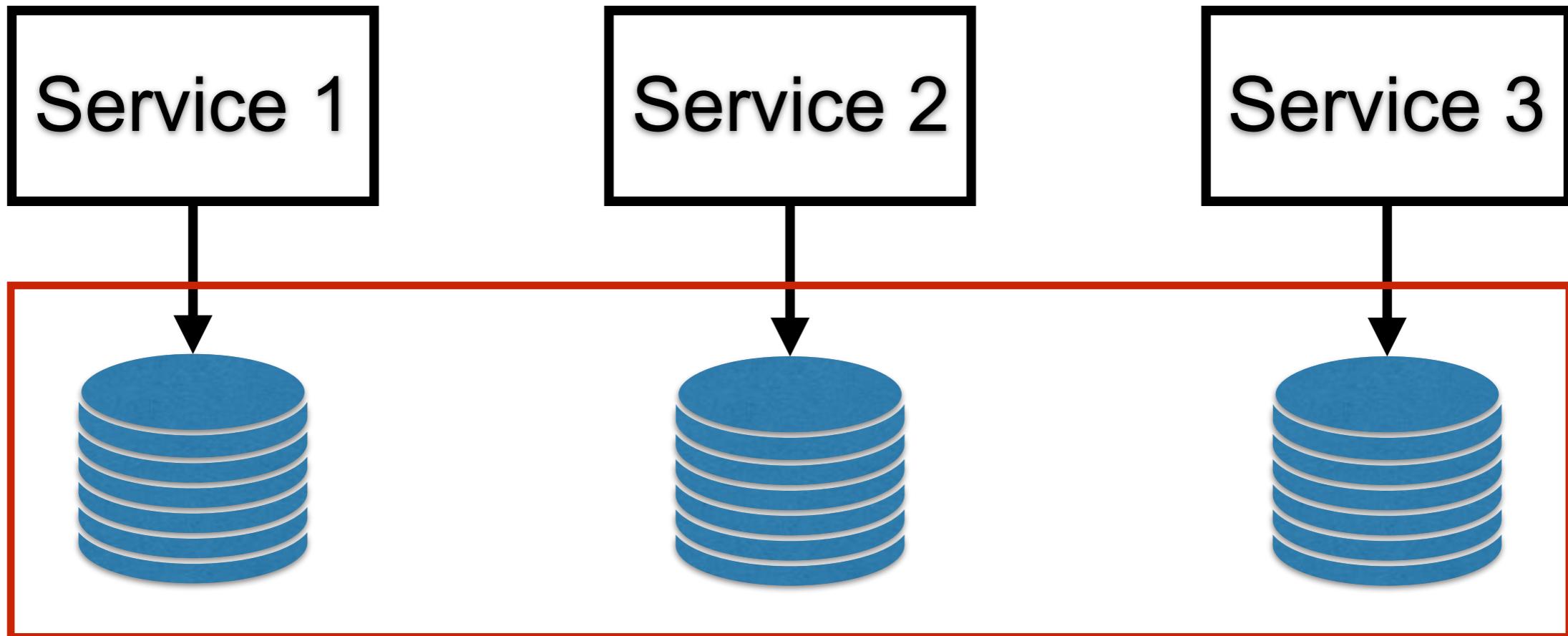


API Gateway

Using the API Gateway Service



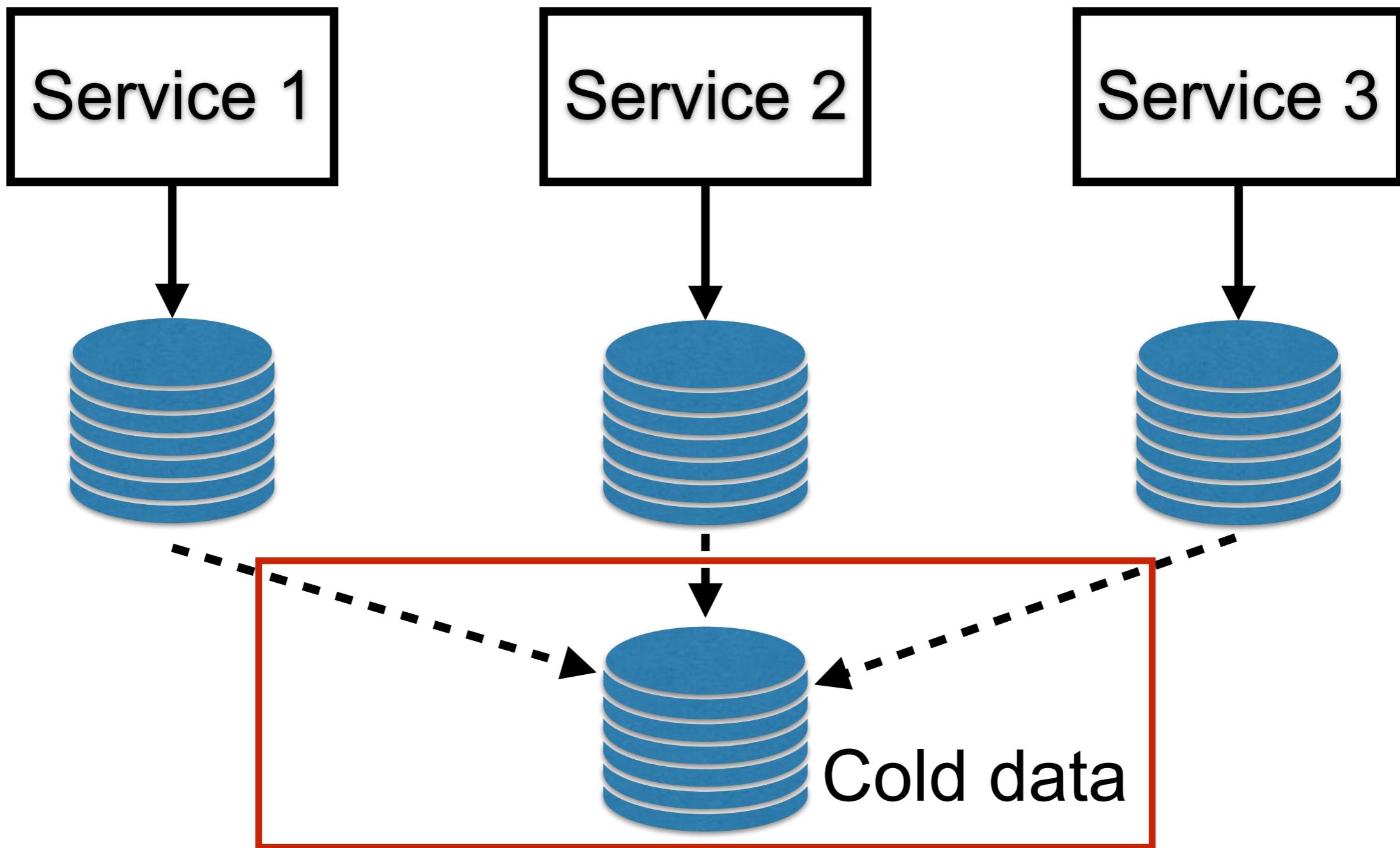
Cold data in central database



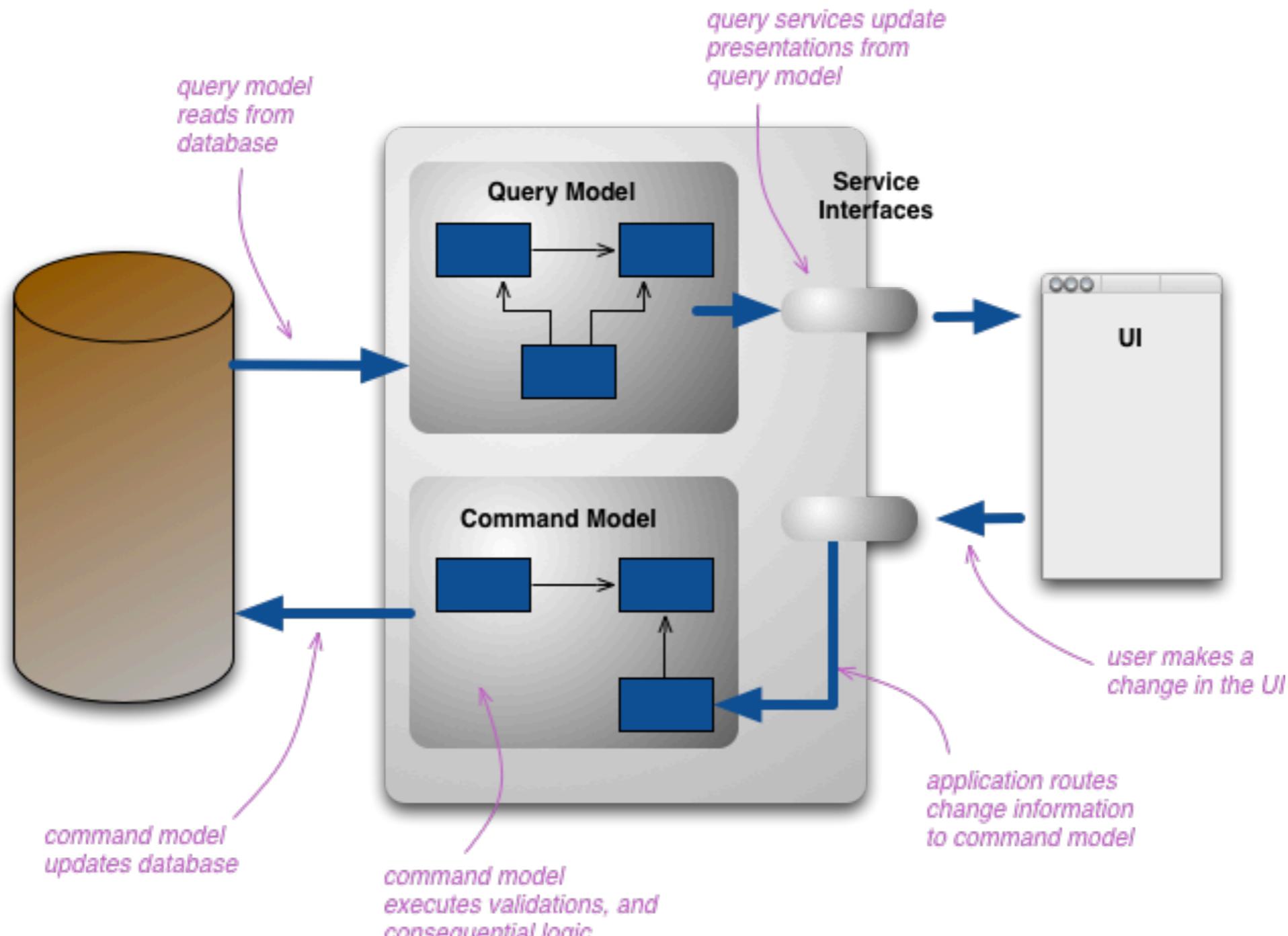
Hot data in each service



Cold data in central database



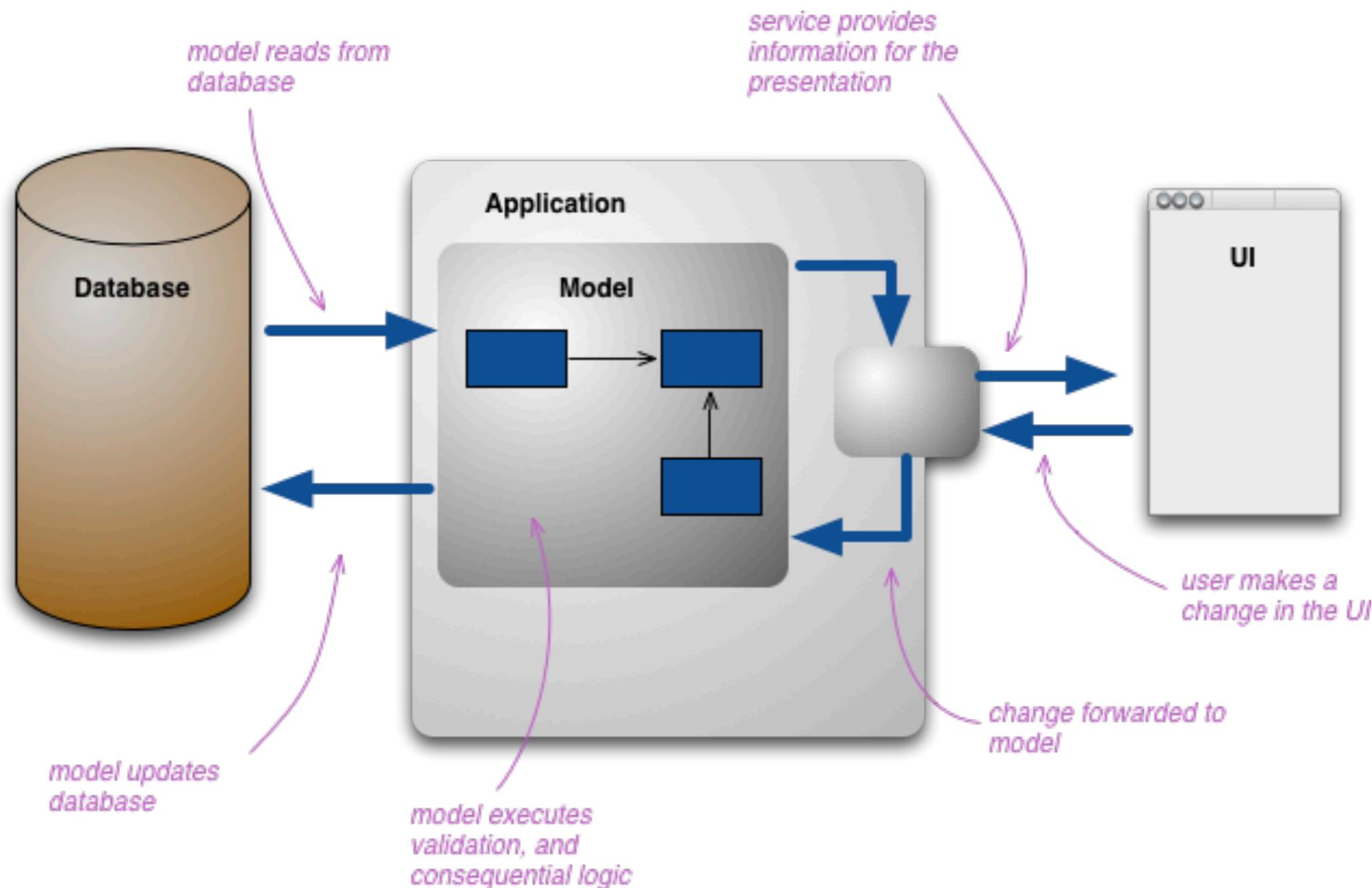
Command Query Responsibility Segregation



<https://martinfowler.com/bliki/CQRS.html>



Command Query Responsibility Segregation



<https://martinfowler.com/bliki/CQRS.html>



3. How to achieve consistency across multiple microservices ?



Ordering microservice

Ordering API



ID	Quantity	ProductID

OrderItems Table

in Ordering-DB
(Remote SQL)

Catalog microservice

Catalog.API



ID	Stock	Name

Products Table

in Catalog-DB
(Remote SQL)

Don't

Databases are private per microservice

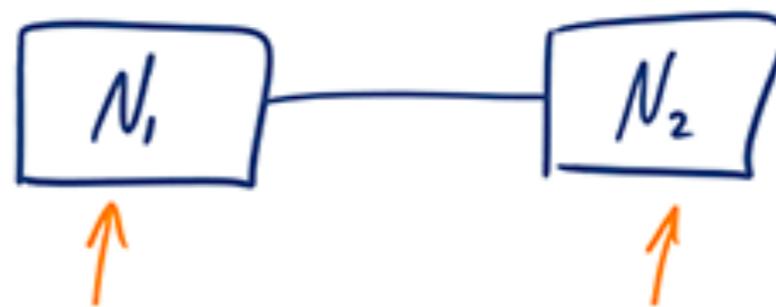


CAP Theorem

Consistency



Availability



Partition Tolerance



<http://robertgreiner.com/2014/08/cap-theorem-revisited/>



Microservices

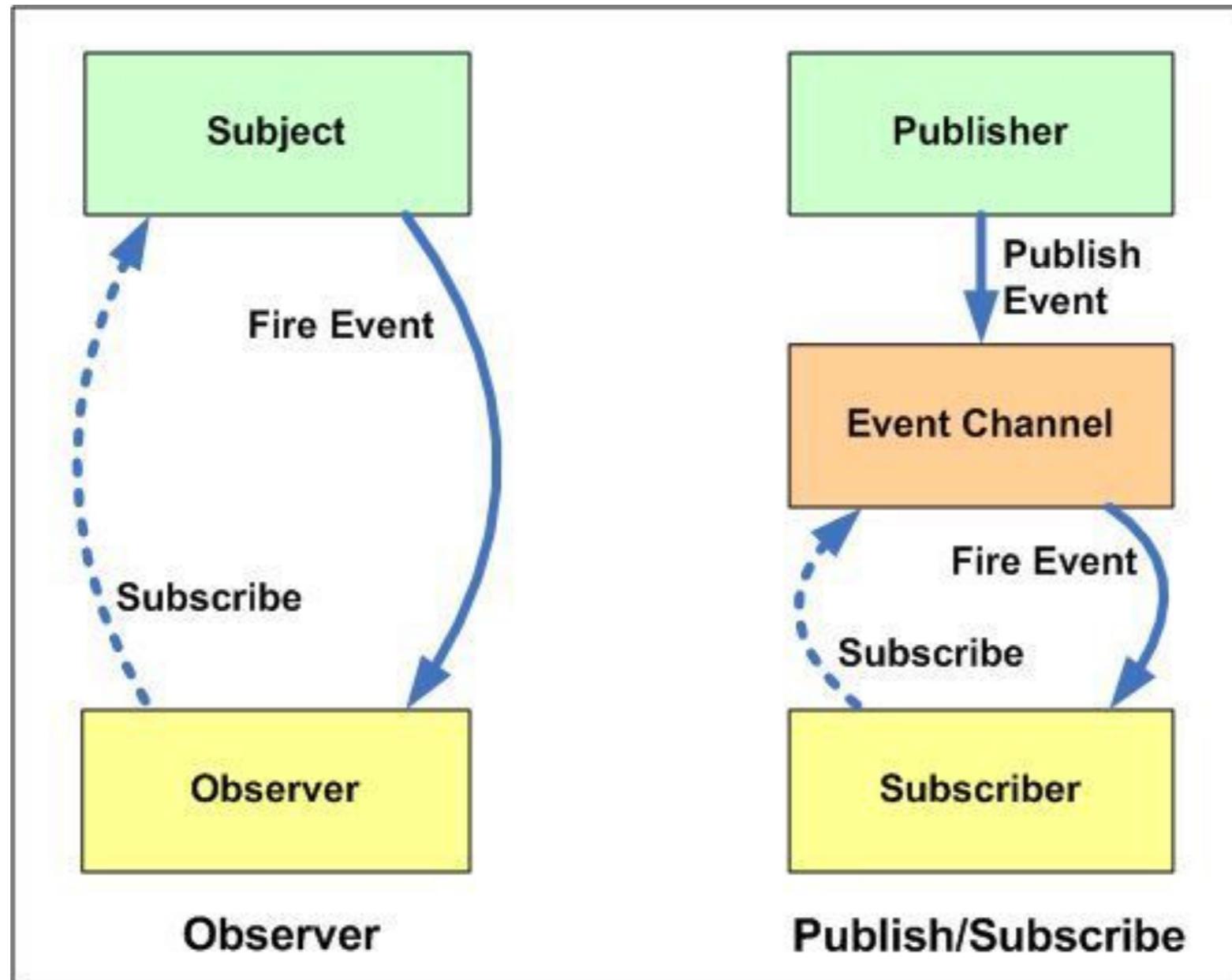
© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

Solutions

Batching process
Event-driven communication
Publish/Subscribe system



Solutions



4. How to design communication across microservices boundaries ?



Protocols ?

HTTP and REST
AMQP
Messaging



Communication styles

Request-Response model
Observer model

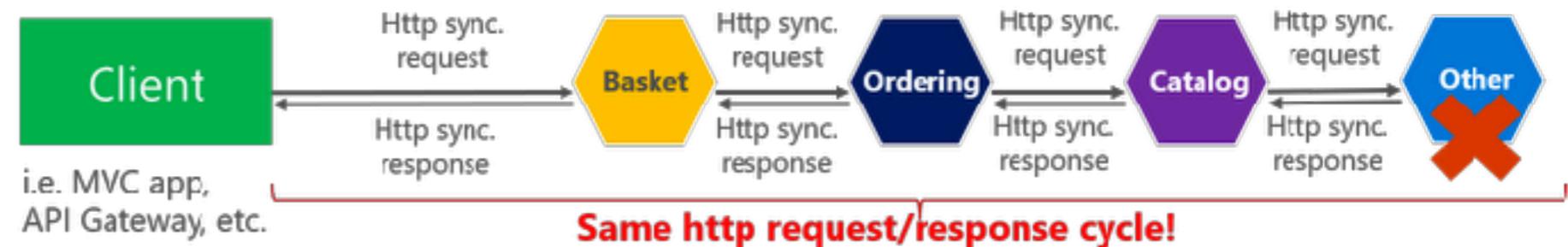


Communication

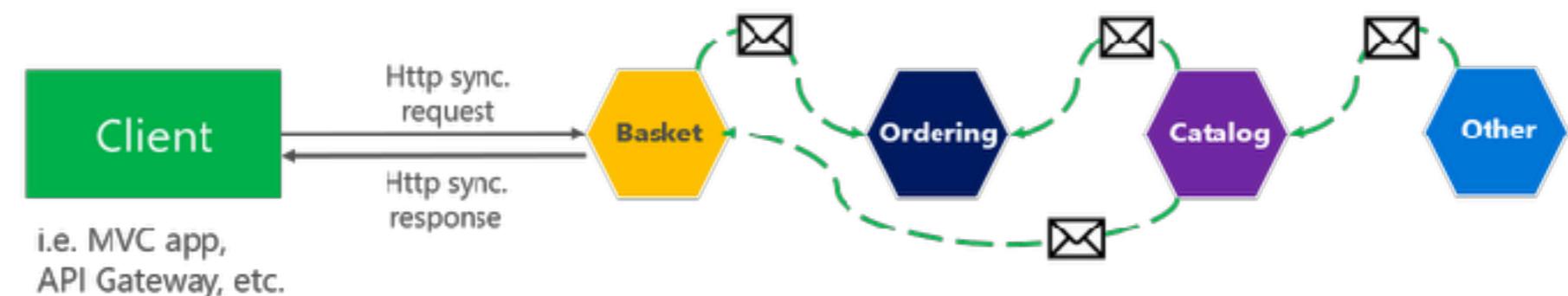
Synchronous vs. async communication across microservices

Anti-pattern

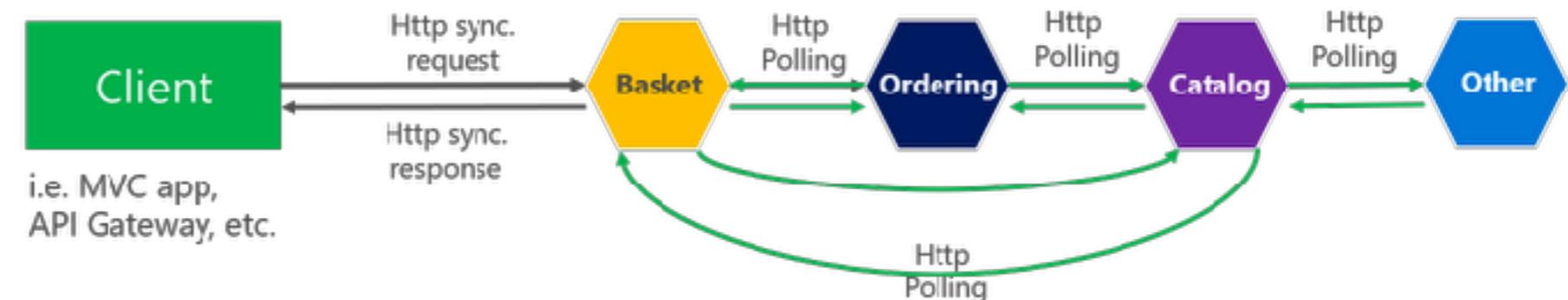
Synchronous
all req./resp. cycle



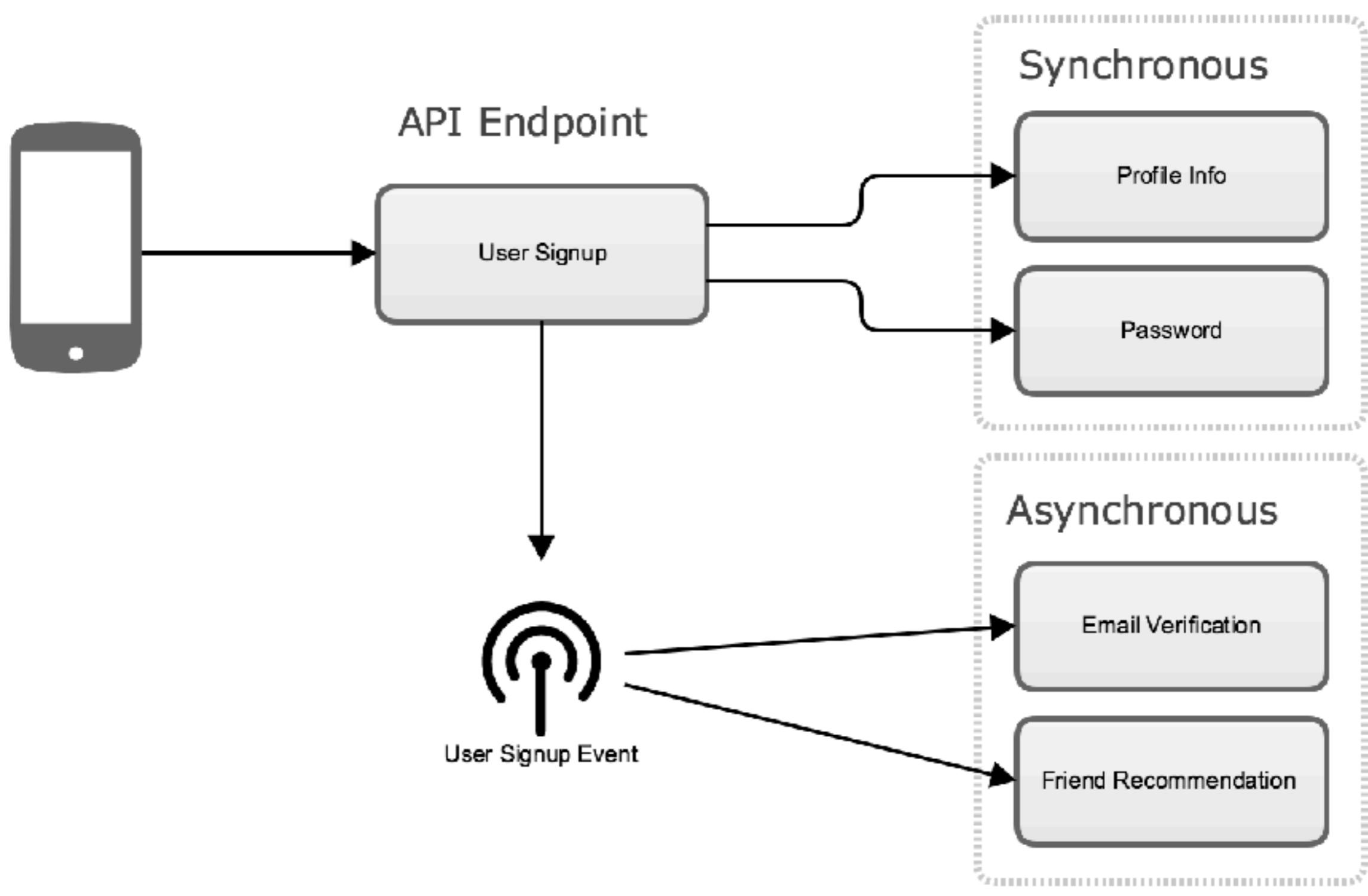
Asynchronous
Comm. across
internal microservices
(EventBus: i.e. **AMQP**)



"Asynchronous"
Comm. across
internal microservices
(Polling: **Http**)

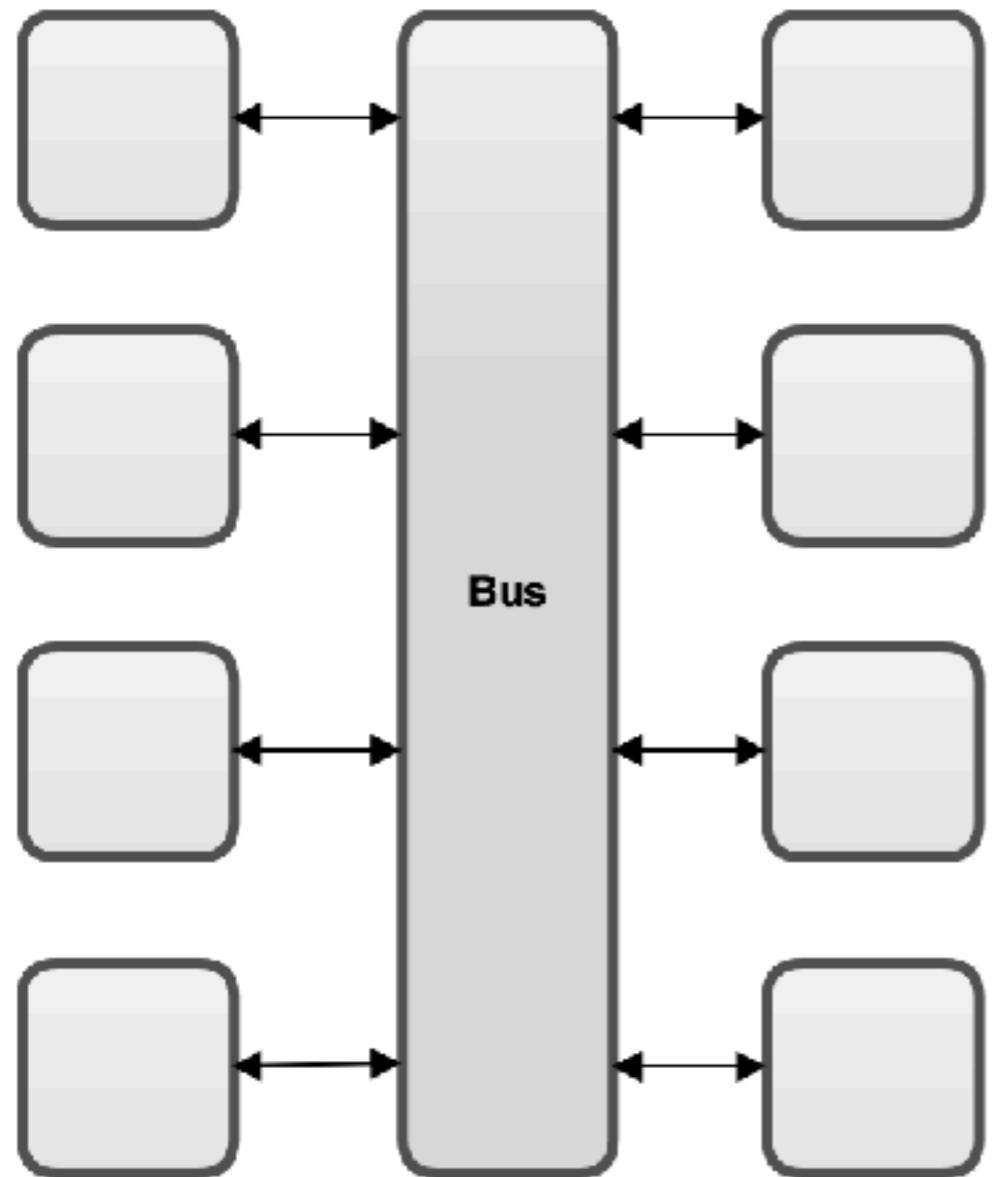


Communication

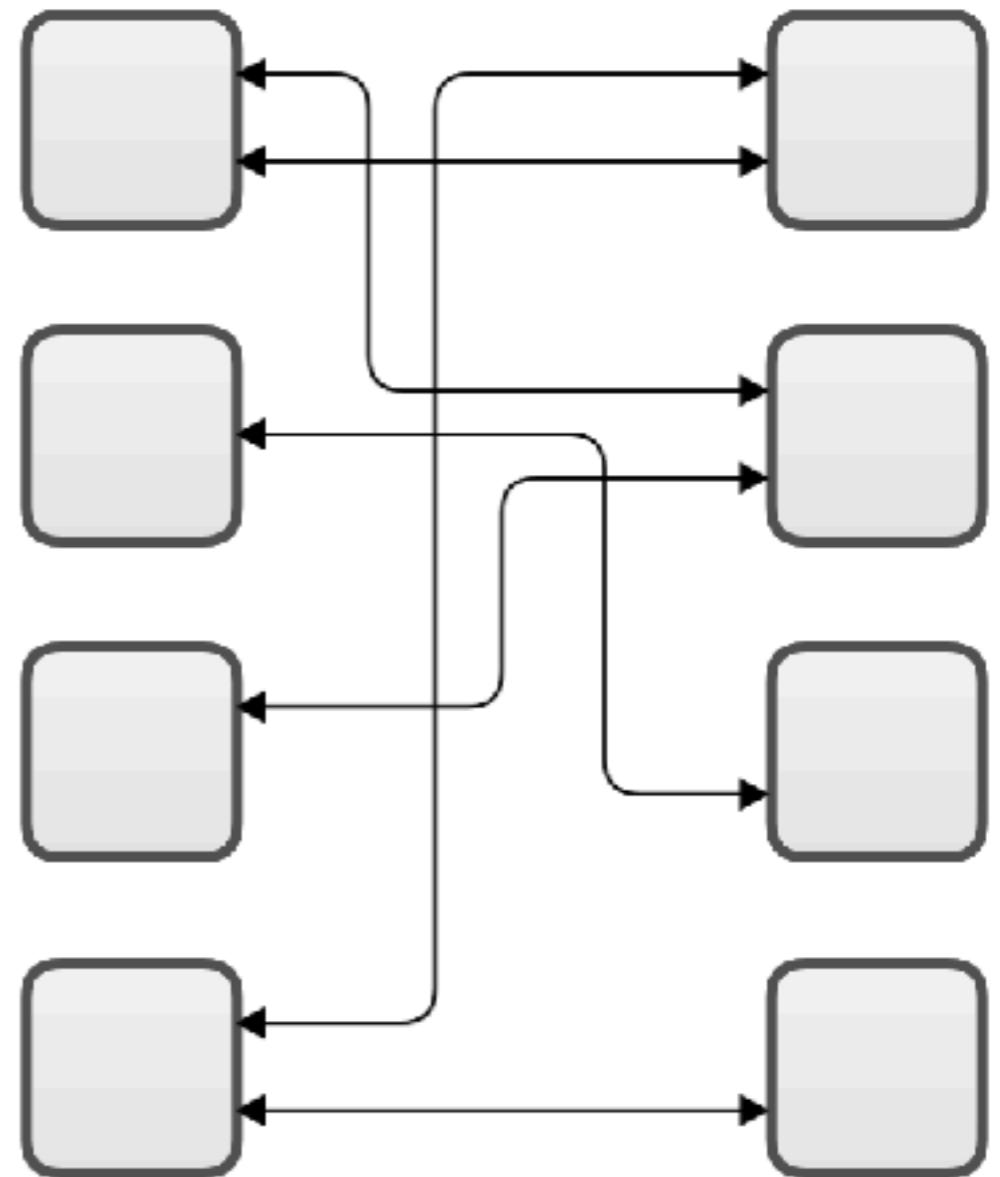


Anti-pattern :: centralize bus service

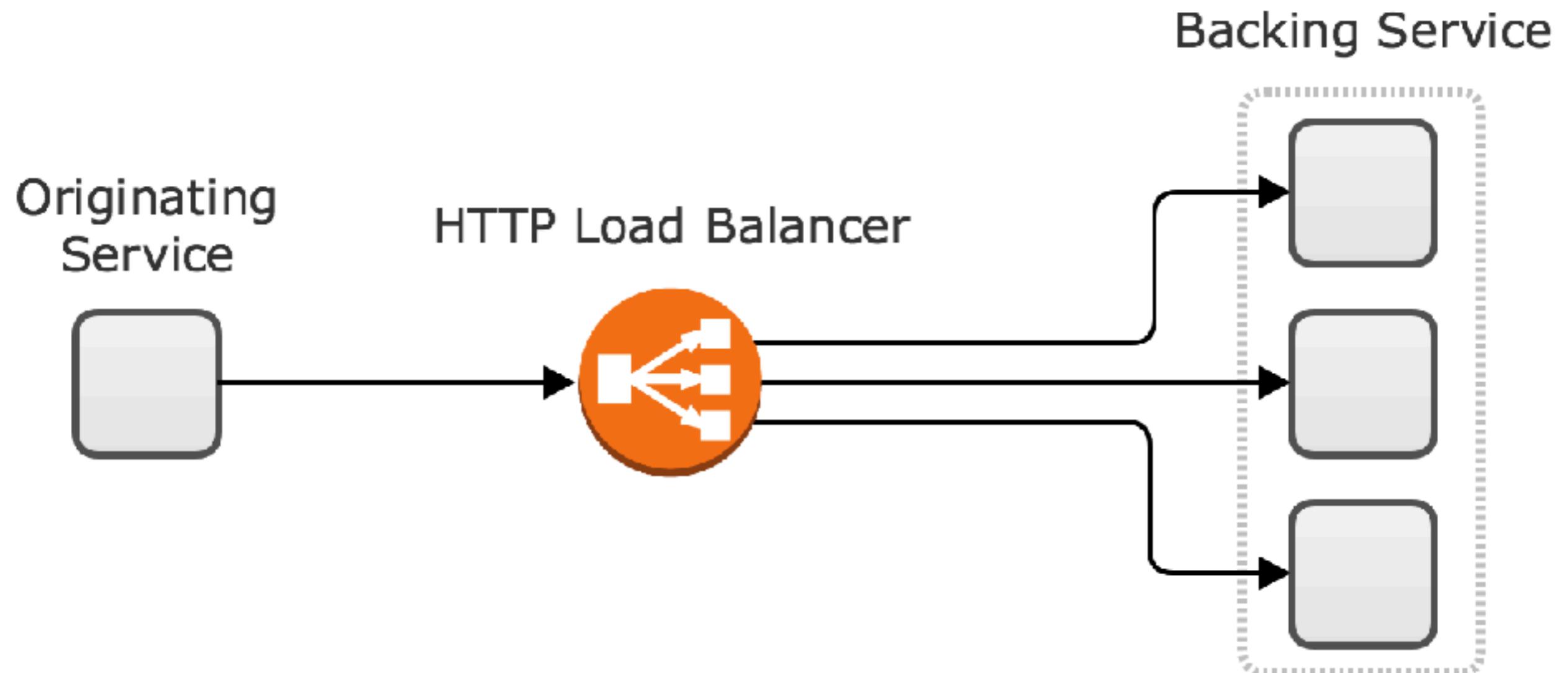
Central Bus



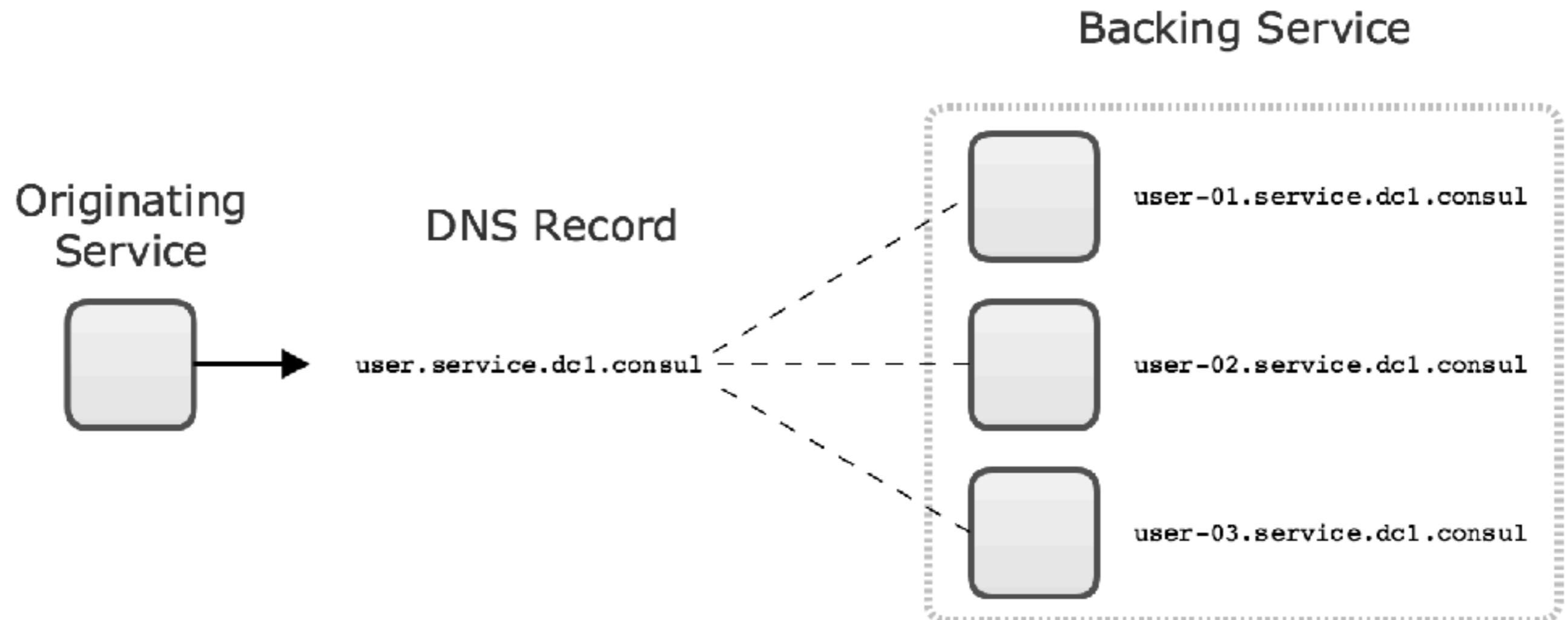
Decentralized



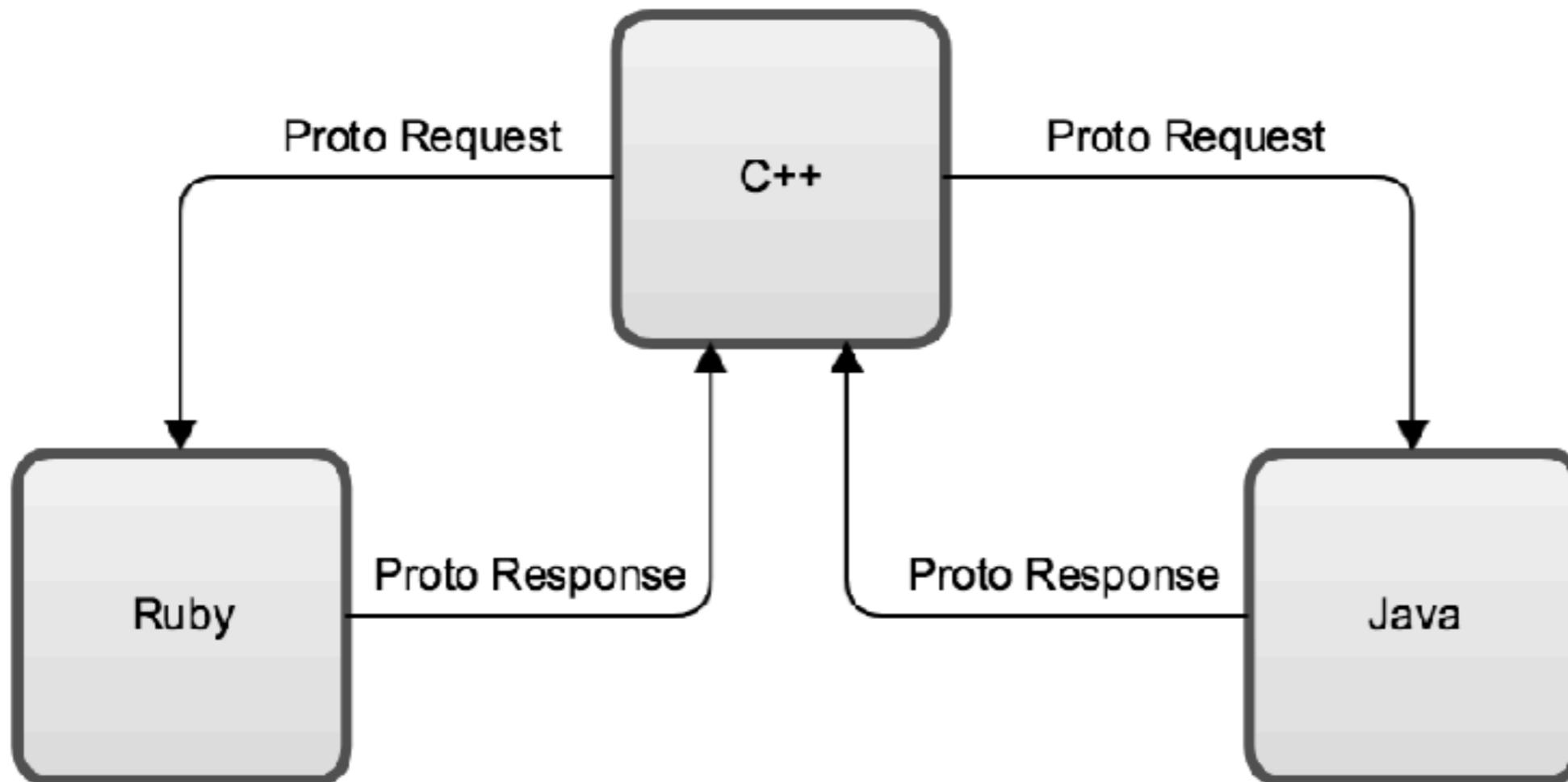
Request-response model



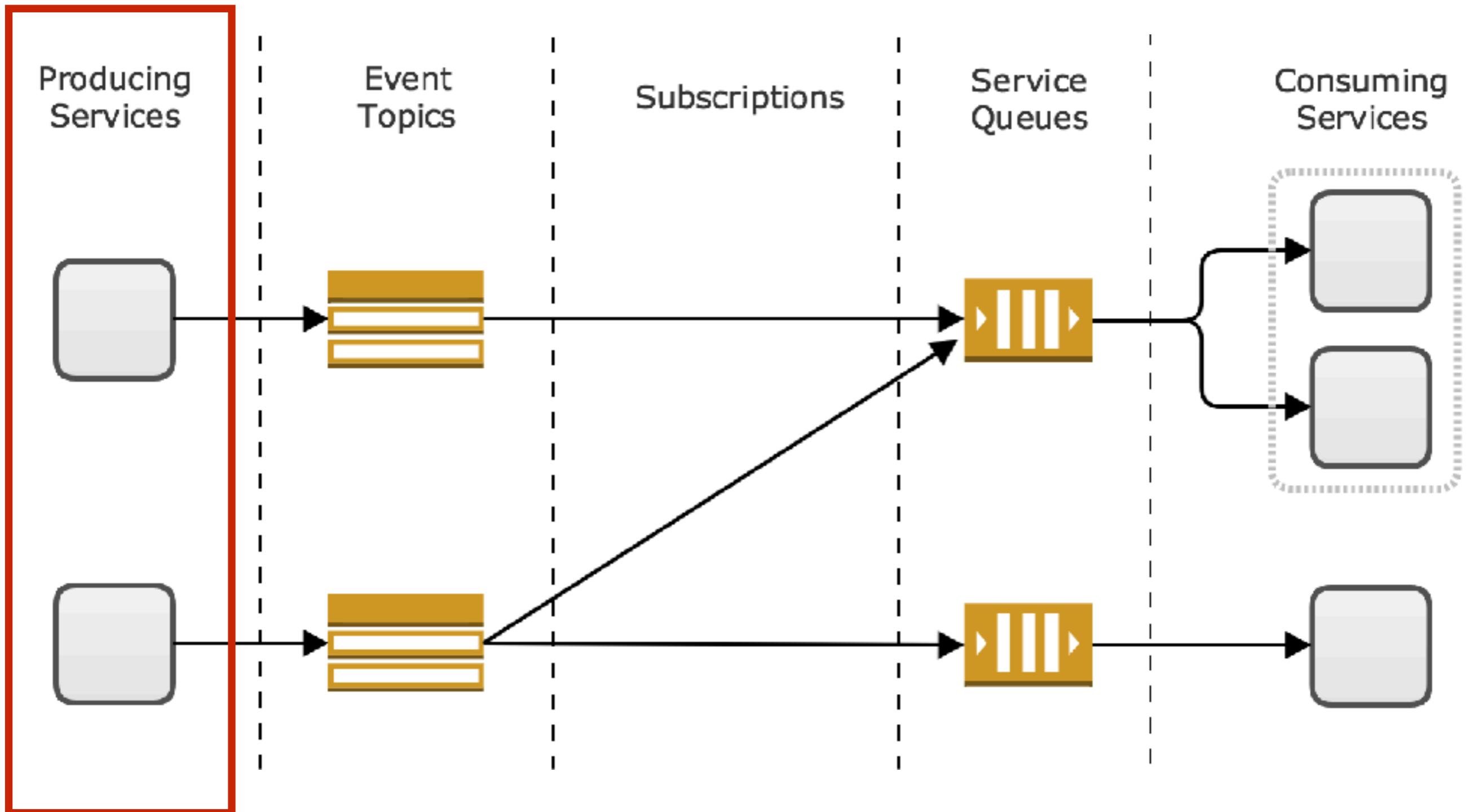
Request-response model



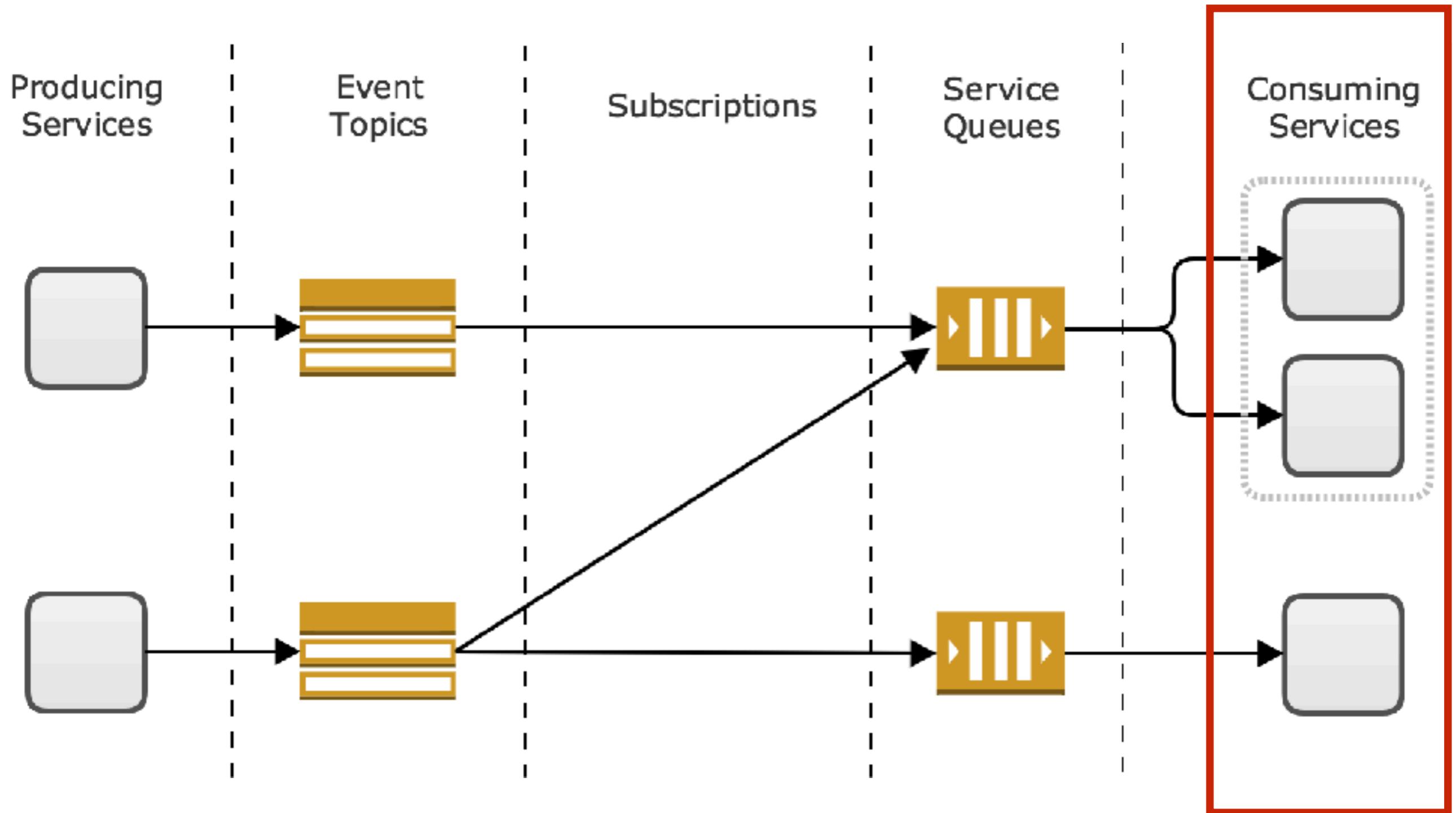
Request-response model



Observer model

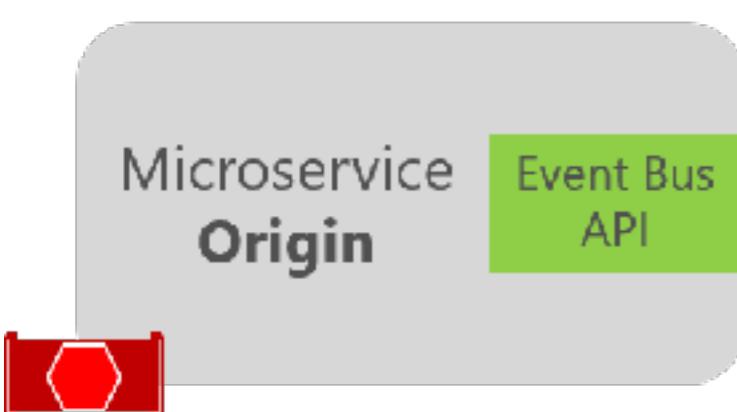


Observer model

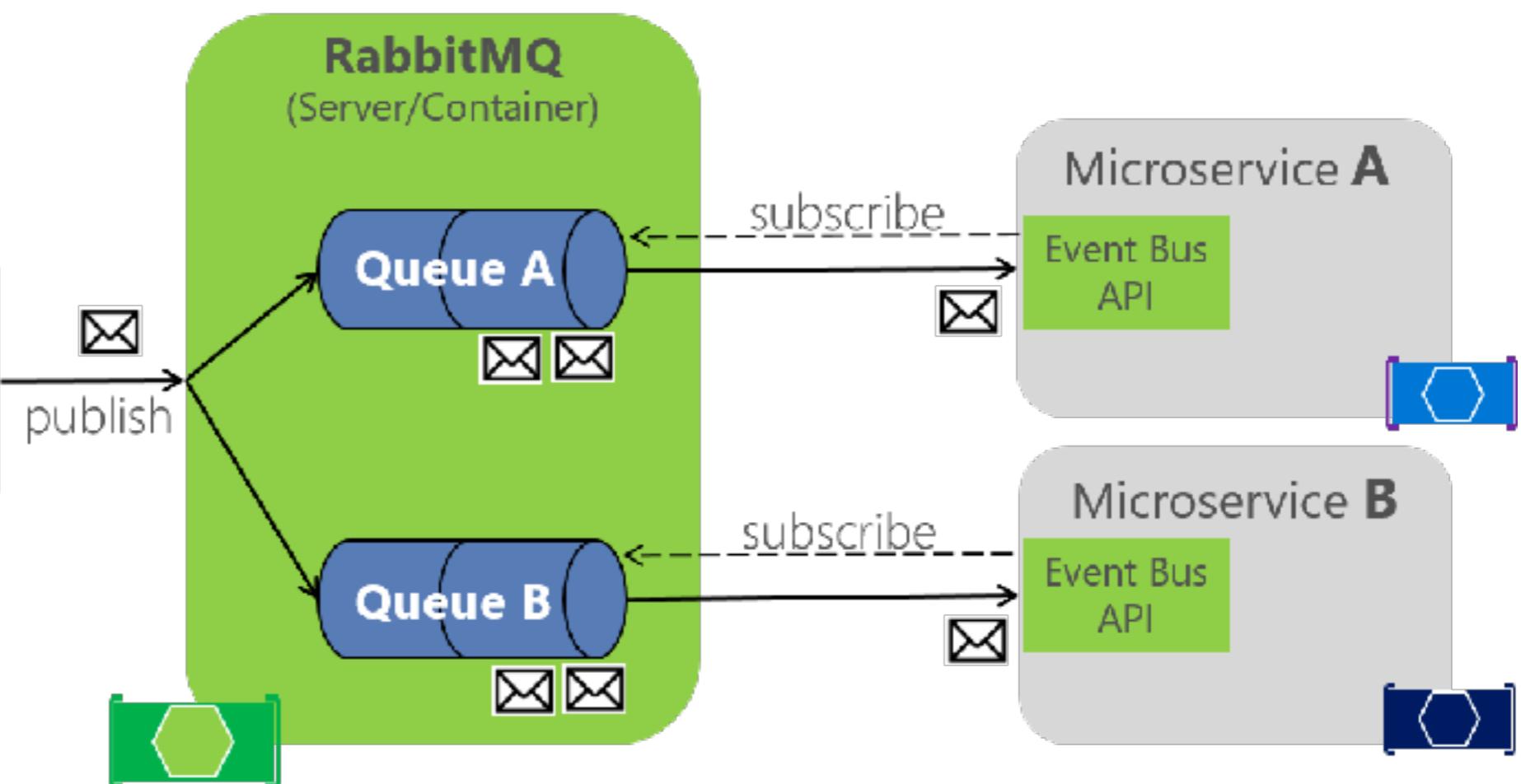
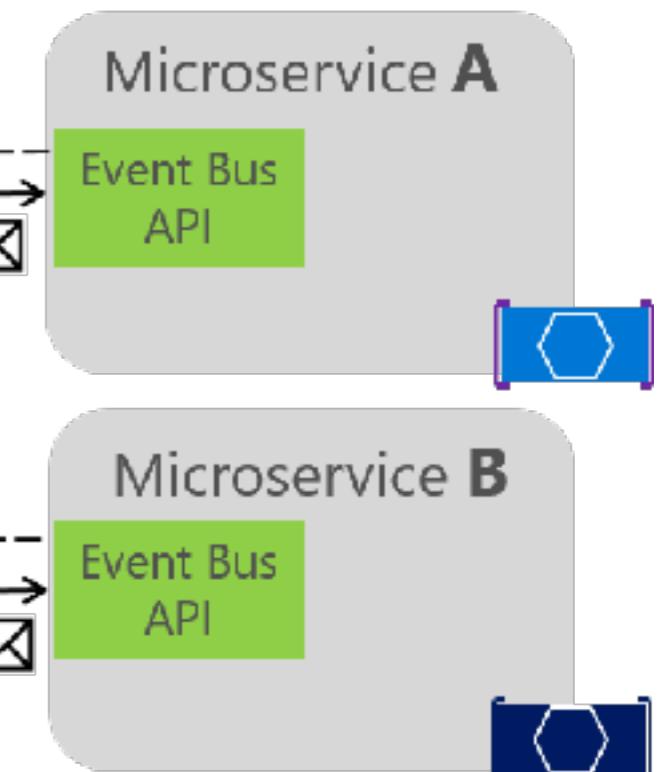


Observer model

**Message
Sender**



**Message
Receivers**



Consideration

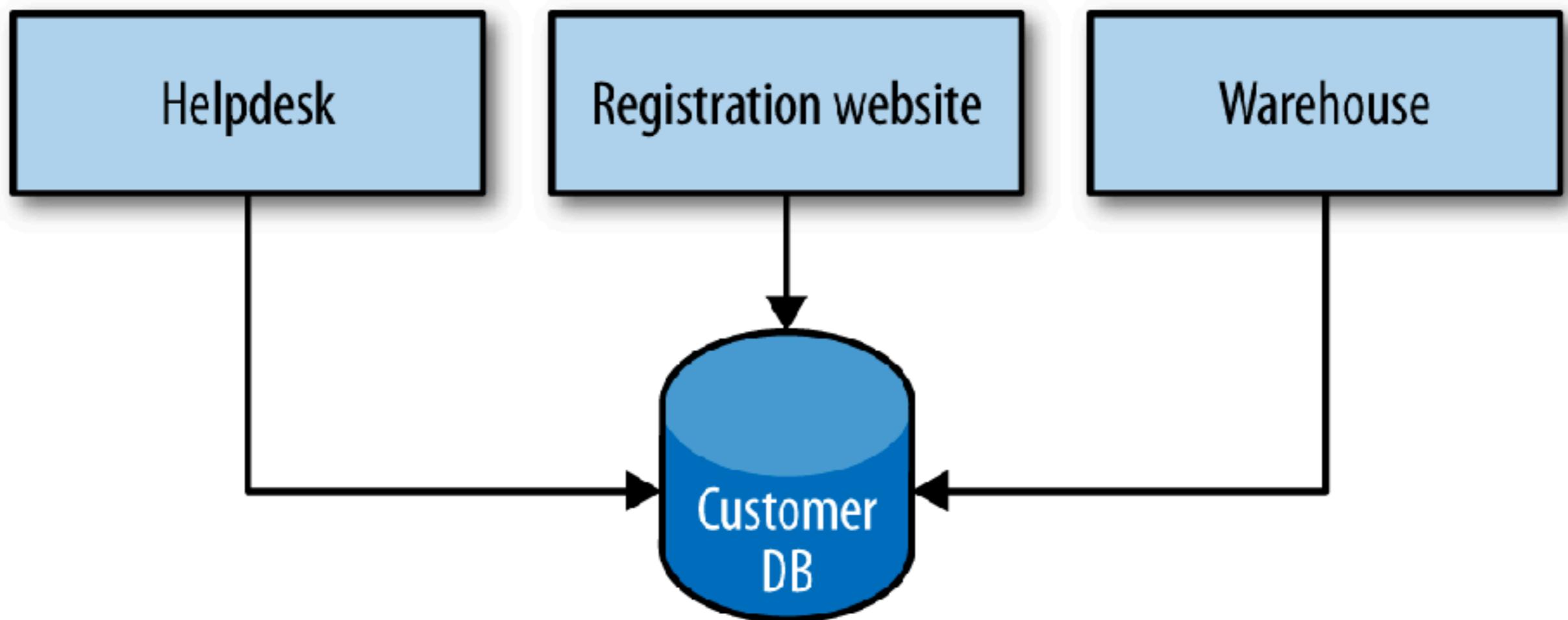
Blocking and low performance
Coupling services
Failure in any one service



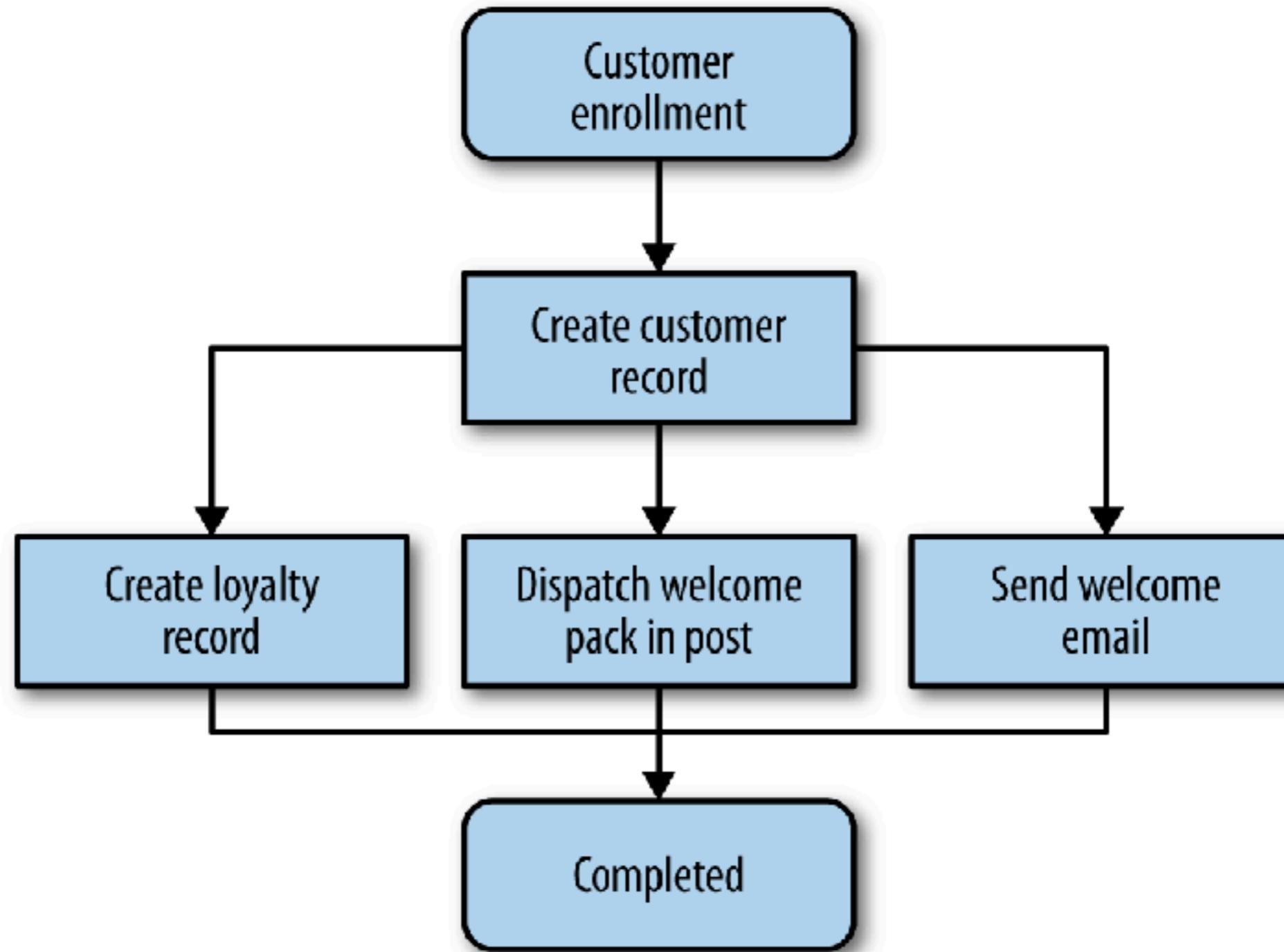
Services Integration



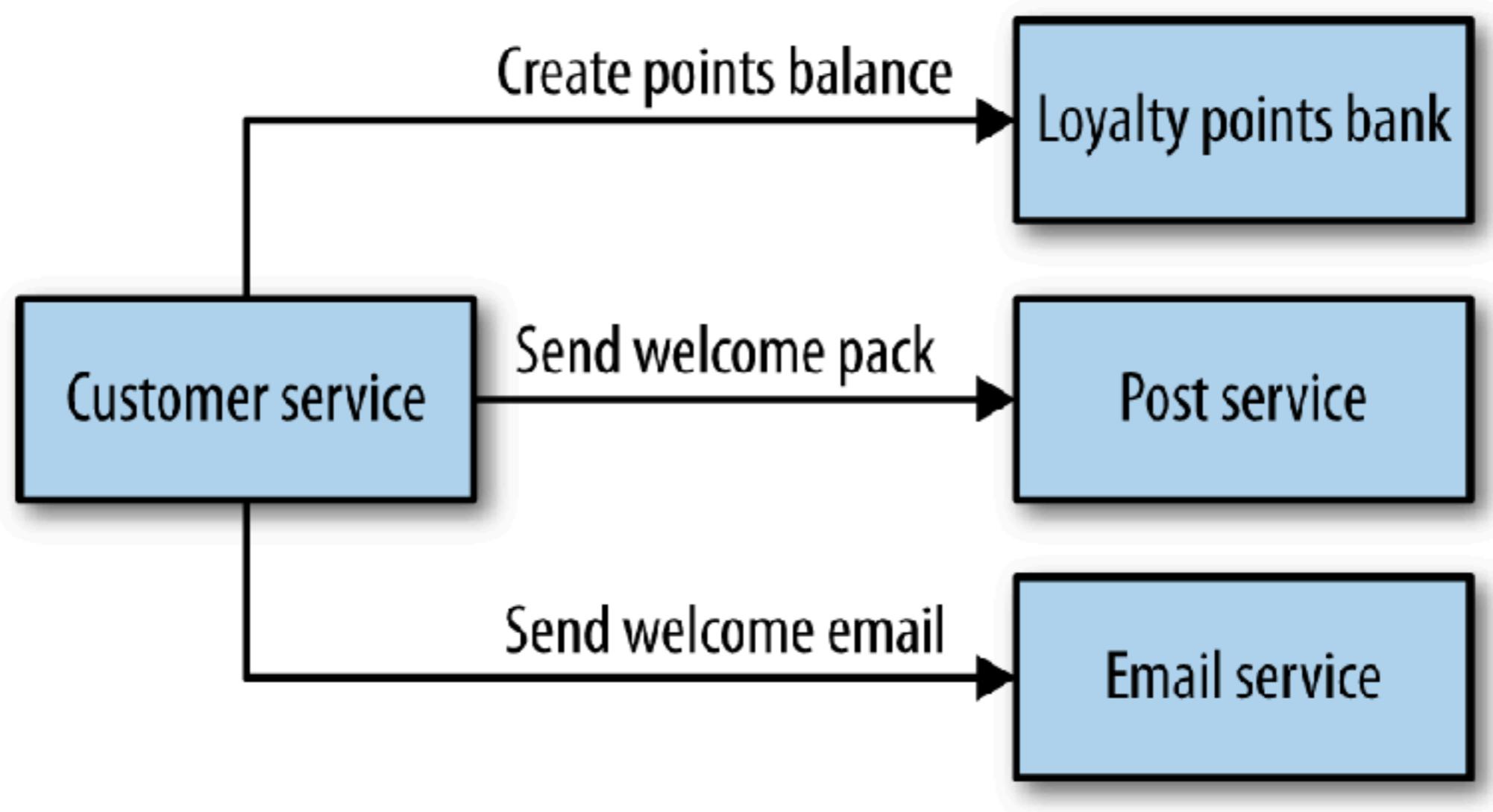
Shared database



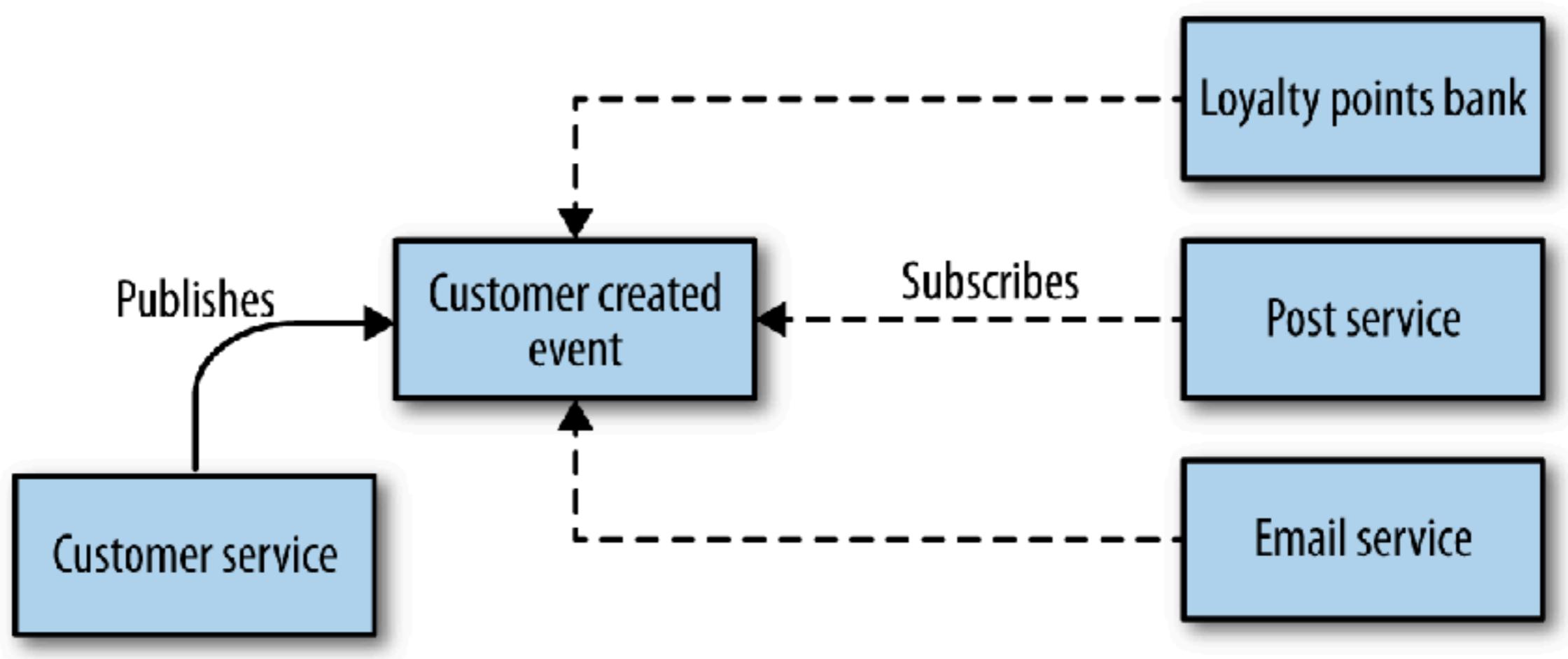
Orchestration vs Choreography



Orchestration



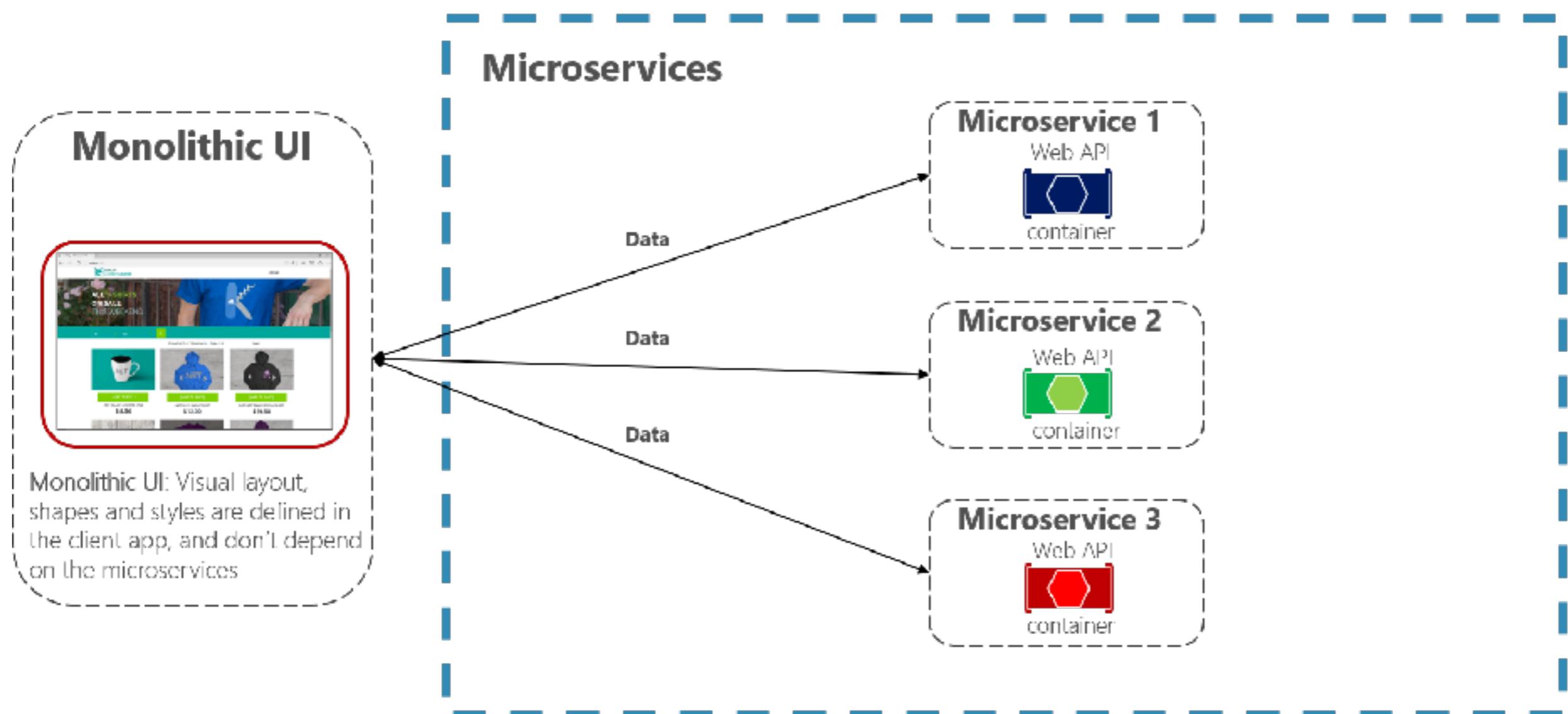
Choreography



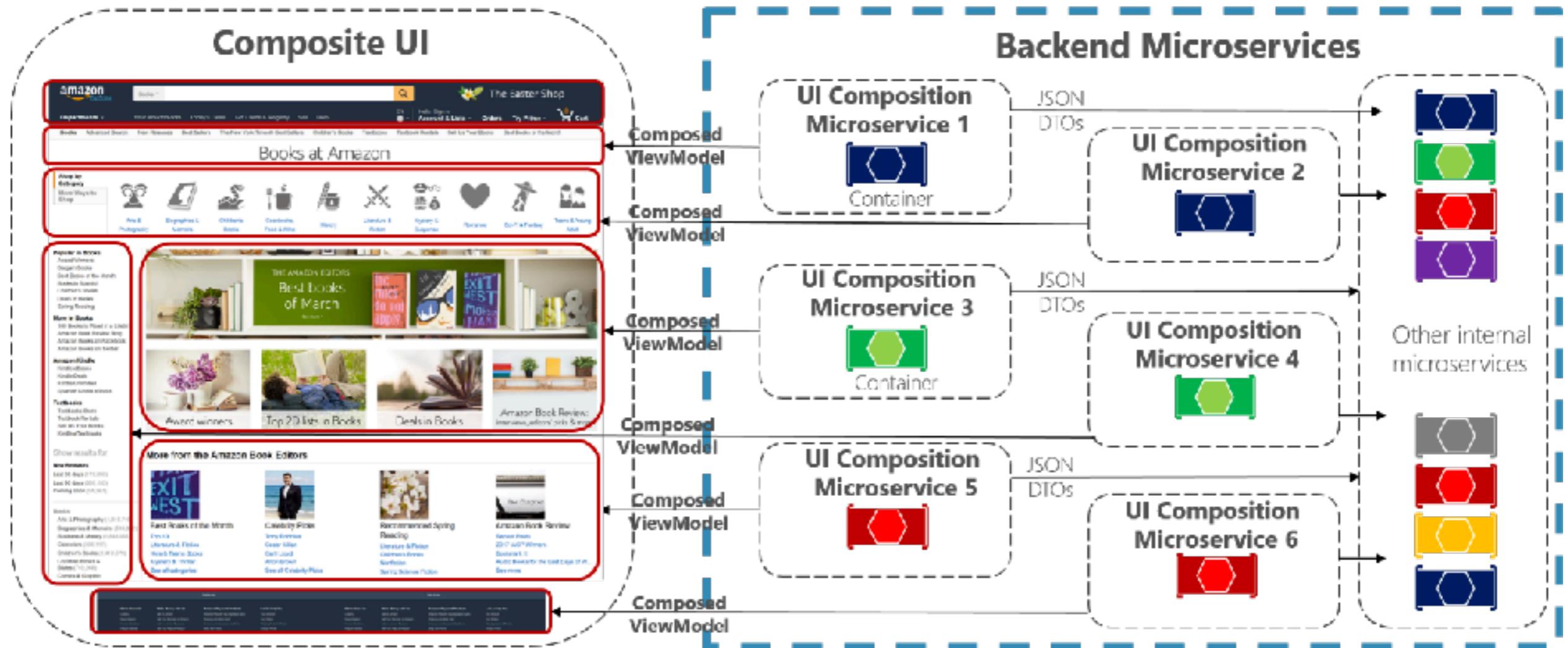
Integrate with User Interface



Monolithic UI consuming microservices



Composite UI generated by microservices



Microservices pitfalls

More/Low splitting
More network interaction
Data storing and sharing
Compatibility issues
Testing
Operation & Monitoring

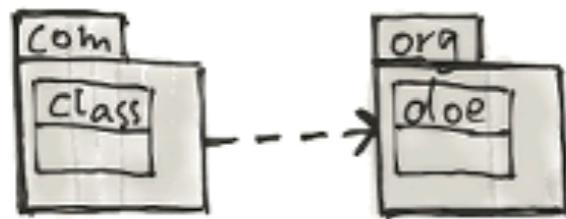


Microservices pitfalls

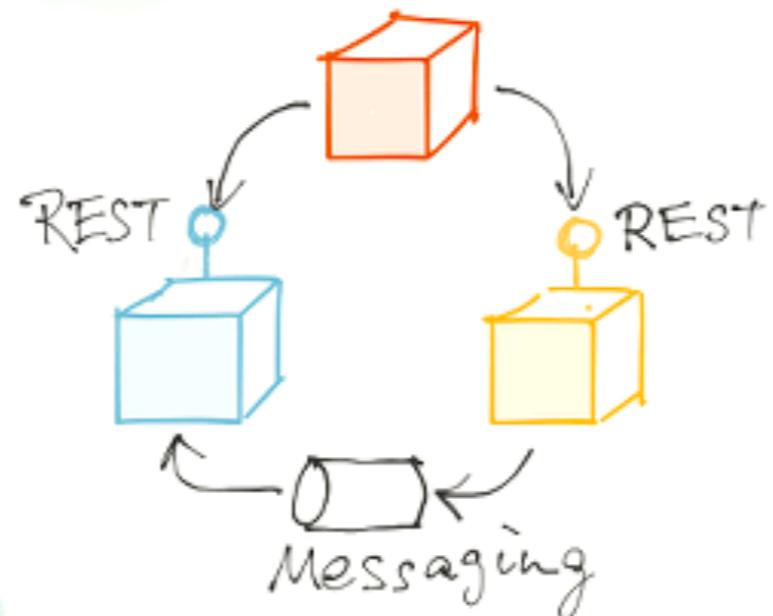
More/Low splitting
More network interaction
Data storing and sharing
Compatibility issues
Testing
Operation & Monitoring



Architecture



Microservices



Deployment

Continuous Delivery

`{ var i=1; }`
Build
Test
Deploy



Infrastructure

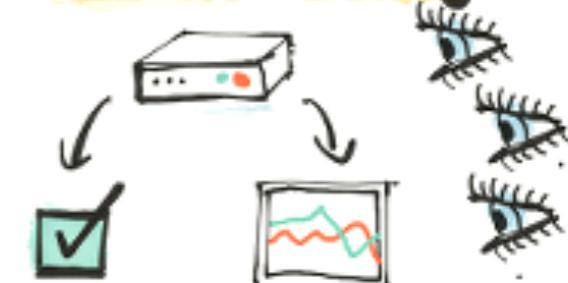


People & Teams



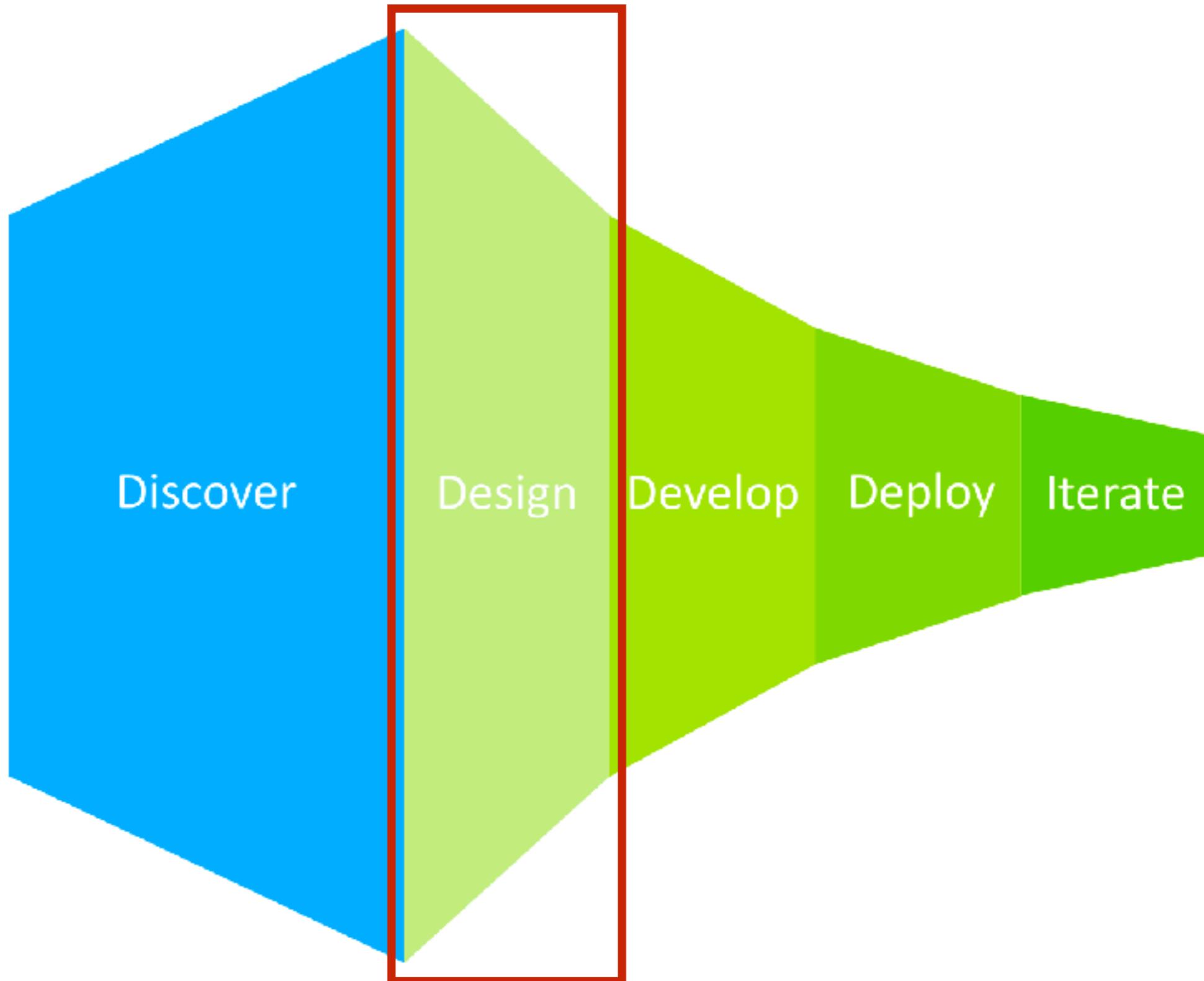
Communication
Collaboration

Monitoring



Features & Technology





Let's workshop with Design



E-commerce system



1. Search product by name

Adidas NMD

350 ค้นพบสินค้าสำหรับ "Adidas NMD"

เรียงตาม: ความเป็นที่นิยม

จำนวนคนดู: [grid icon] [list icon]

Image	Name	Price	Rating	Actions
	Adidas Yeezy Boost 350 V2 Beluga 2.0 (AH2203)	฿28,900.00 ฿30,000.00 -28%	★★★★★ (1)	รายละเอียด
	Adidas NMD R1 Primeknit Core Black / Core Black...	฿9,900.00 ฿15,000.00 -34%	★★★★★ (1)	รายละเอียด
	Adidas NMD R1 PK Japan Triple Black (BZ0220)	฿12,900.00 ฿15,000.00 -14%	★★★★★ (1)	รายละเอียด
	POCA SHOE NMD Sneakers Fashion รองเท้า ลำลอง ผ้าใบ ...	฿399.00 ฿1,000.00 -79%	★★★★★ (1)	รายละเอียด
	Adidas NMD R1 Color Core Black/Icey Blue (BY9951)	฿7,990.00 ฿12,000.00 -33%	★★★★★ (1)	รายละเอียด



2. Choose a product

The screenshot shows a search results page for "Adidas NMD". The search bar at the top contains the query "Adidas NMD". Below the search bar is a dark blue header with the following navigation items: "ร้านค้า ทางการ" (Shop), "Taobao คอลเลคชัน" (Taobao Collection), "ไฟฟ์สไตร์ & เติมเงิน" (Five Styles & Top-up), and "ສະໂຕດ ສົດເທິນ" (Cart). A dropdown menu for sorting is open, showing "เรียงตาม: ຄວາມເປັນທີ່ນິຍົມ" (Sort by: Popularity) with a downward arrow. To the right of the dropdown is a "ຈຳນວນຄົນດູ" (Number of products) button with a grid icon.

350 ຄົນພັນສິນຄ້າສໍາຫລວນ "Adidas NMD"

Image	Product Name	Price	Original Price	Discount (%)	Rating	Reviews	Action
	Adidas Yeezy Boost 350 V2 Beluga 2.0 (AH2203)	฿28,900.00	฿39,000.00	-28%	4.5	(70)	ລັບອອກ
	Adidas NMD R1 Primeknit Core Black / Core Black...	฿9,900.00	฿15,000.00	-34%	4.5	(70)	ລັບອອກ
	Adidas NMD R1 PK Japan Triple Black (BZ0220)	฿12,900.00	฿15,000.00	-14%	4.5	(70)	ລັບອອກ
	POCA SHOE NMD Sneakers Fashion ລາງເກົ່າ ລໍາລອງ ພ້າໃນ ...	฿399.00	฿1,000.00	-79%	4.5	(70)	ລັບອອກ
	Adidas NMD R1 Color Core Black/Icey Blue (BY9951)	฿7,990.00	฿12,000.00	-33%	4.5	(1)	ລັບອອກ



3. Show product detail

POCA SHOE NMD Sneakers Fashion รองเท้า ลำลอง ผ้าใบ ผู้หญิง-ผู้ชาย แฟชั่น
ราคาถูกswyฯ Sport Unisex รุ่น PSN-Black/White

★★★★☆ (70) แสดงความคิดเห็น
ชื่อ Poca Shoes | เพิ่มเติม สุภาพบุรุษ จาก Poca Shoes



2 Weeks Warranty by Seller [เพิ่มเติม](#)

- สวมใส่สบาย
- เพิ่มเติม

เลือก ขนาด

ขนาด [เลือก](#)

ขนาด [เลือก](#)

399 บาท

ราคาปกติ 1,900 บาท,
ประหยัดทันที 79%
ราคาโปรโมชั่นสามารถใช้ได้กับ 25/2/2018

ใส่ตะกร้า

← [วิธีการสั่งซื้อ](#)



4. Add product to basket

POCA SHOE NMD Sneakers Fashion รองเท้า ลำลอง ผ้าใบ ผู้หญิง-ผู้ชาย แฟชั่น
ราคาถูกswyฯ Sport Unisex รุ่น PSN-Black/White

★★★ (70) แสดงความคิดเห็น

ชื่อ Poca Shoes | เพิ่มเติม สุภาพบุรุษ จาก Poca Shoes



2 Weeks Warranty by Seller [เพิ่มเติม](#)

- สวมใส่สบาย
[เพิ่มเติม](#)

เลือก ขนาด

ขนาด [เลือก](#)

399 บาท

ราคาปกติ 1,900 บาท,
ประหยัดทันที 79%
ราคาโปรโมชั่นสามารถใช้ได้กับ 25/2/2018

ใส่ตะกร้า



5. Show data in basket

✓ สินค้า 1 ชิ้น ได้ถูกเพิ่มเข้าไปยังตะกร้าสินค้าของคุณ



POCA SHOE NMD Sneakers
Fashion รองเท้า ลำลอง ผ้าใบ ผู้หญิง-ผู้ชาย แฟชั่น ราคาถูกswy Sport
Unisex รุ่น PSN-Black/White

ไซส์: EU:40

Poca Shoes

399 บาท

1,900 บาท 79% ปิด

ตะกร้าสินค้าของคุณ (1 สินค้า)

มูลค่าสินค้า: **399 บาท**

ยอดสุทธิ รวมภาษีมูลค่าเพิ่ม (จำนวน): **399 บาท**

เลือกชื่อสินค้าต่อ

ชำระค่าสินค้า

People Who Bought This Item Also Bought



◀ กางเกงสแลคขายาว Hopper Progress ผ้ายืด ทรงเข้ารูป

900 บาท

67% ปิด

299 บาท



6. Checkout

✓ สินค้า 1 ชิ้น ได้ถูกเพิ่มเข้าไปยังตะกร้าสินค้าของคุณ



POCA SHOE NMD Sneakers
Fashion รองเท้า ลำลอง ผ้าใบ ผู้หญิง-ผู้ชาย แฟชั่น ราคาถูกswy Sport
Unisex รุ่น PSN-Black/White

ไซส์: EU:40

Poca Shoes

399 บาท

1,900 บาท 79% ลด

ตะกร้าสินค้าของคุณ (1 สินค้า)

มูลค่าสินค้า: **399 บาท**

ยอดสุทธิ รวมภาษีมูลค่าเพิ่ม (ตาม): **399 บาท**

เลือกชื่อสินค้าต่อ

ชำระค่าสินค้า

People Who Bought This Item Also Bought



◀ กางเกงสแลคขายาว Hopper Progress ผ้ายืด ทรงเข้ารูป

900 บาท

67% ลด

299 บาท



Microservices

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

7. Shipping

LAZADA
CO-TH

1. คำสั่งซื้อ

2. ชำระเงิน

ที่อยู่ที่จะจัดส่ง

Login for speedy checkout

ชื่อ	กฤษดา ใจ อิเมล์ ของท่าน	
ชื่อ และ นามสกุล	ชื่อและนามสกุล	
ที่อยู่	ที่อยู่	
รหัสไปรษณีย์	รหัสไปรษณีย์	ทางเราระบุการตรวจสอบเมืองและจังหวัดของคุณ
เมือง	เมือง	
จังหวัด	กรุงเทพมหานคร/ Bangkok	
โทรศัพท์มือถือ	+66 เบอร์โทรศัพท์	เพื่อให้รับไปรษัทการจัดส่งได้

ท่องเที่ยวในประเทศ/ในกำกันภาษี - กรุณาเดือนของการออกข้อมูลเพื่อทำการขอในกำกันภาษี

ข้อมูลการส่งเงินค่า

ชั่วโมงเวลา: พี.วี.
Get it วันอังคาร, 27 ก.พ. - วันจันทร์, 5 มี.ค. 2018

ค่า斐นการต่อ

สูปการสั่งซื้อ (1 items)

สินค้า	จำนวน	ราคาร
POCA SHOE NMD Sneakers Fashion รองเท้า ลั่นลง แนวใหม่ สีฟ้า-สีขาว แฟชั่น ราคาถูกสุดๆ Sport Unisex รุ่น PSN-Black/White ขนาด: EU:40	1	399
รวมค่าสินค้า		399 บาท
ยอดสุทธิ รวมรวมภาระค่าสัมภาระ (ถ้ามี)		399 บาท

 คุณภาพดี 100%





8. Payment

LAZADA
.CO.TH

✓ 1. ค่าซื้อขั้นต่ำ

2. ชำระเงิน

เลือกคัวเลือกสำหรับการชำระเงิน

บัตรเครดิตหรือเดบิต	เก็บเงินปลายทาง	ชำระเงินผ่านเคาน์เตอร์	PayPal/Amex	มอนชาร์	LINE Pay	หักบัญชีธนาคาร/ช่องทางATM

หมายเหตุ:
ชื่อบนบัตร: Somkiat Puisungnoen
วันที่บัตรหมดอายุ: mm yy
CCV / CVV: ?

ข้อมูลใบกำกับภาษีไม่สามารถเปลี่ยนแปลงได้หลังการสั่งซื้อสินค้า

ล็อก สั่งซื้อสินค้า

VERIFIED by VISA MasterCard SecureCode

สมัครรับข่าวสารกับลาซาด้าเพื่อรับส่วนลดและข้อเสนอสุดพิเศษ

โดยการร่วมค้าสั่งซื้อของคุณ, คุณยอมรับข้อกำหนดของทางลาซาด้า [ในการร่วมสินค้าทางช่องทางที่กำหนดไว้ และ ร้ออกกฎหมายและเงื่อนไข](#)

ส่งที่ ไปรษณีย์
Somkiat Puisungnoen
122/64 , Sci Phahonyothin 2, Phahonyothin Road Prom Condo กรุงเทพมหานคร/ Bangkok - พญาไท/ Phaya Thai - 10400 โทรศัพท์: 0868696209

สรุปการสั่งซื้อ (1 items)

สินค้า	จำนวน	ราคา
POCA SHOE NMD Sneakers Fashion รองเท้า ลำลอง ถ้าใบ สีฟ้า-เขียว แพ็ค ราคาปกติ รายๆ ละ 399 รุ่น PSN-Black/White ขนาด: EU:40	1	399 บาท

ส่งแบบธรรมด้า
วันอัจฉริ, 27 ก.พ. - วันเสาร์, 3 มี.ค. 2018

กรอกคุณปองส่วนลดที่นี่ **ขึ้นชั้น**

มูลค่าสินค้า
ค่าซื้อขั้นต่ำ 399 บาท
ยอดสุทธิ 399 บาท

รายการรวมภาษีมูลค่าเพิ่ม (มีภาษี)

คุณภาพดี 100%
 ปลอดภัย 100%



9. Confirm to order

LAZADA
•CC•TH

✓ 1. ค่าใช้จ่ายค่า

2. ชำระเงิน

เลือกคัวเลือกสำหรับการซื้อขาย

บัตรเครดิตหรือ Debit	เก็บเงินปลายทาง	ชำระเงินผ่านเคาน์เตอร์	PayPal/Amex	มอนชาร์	LINE Pay	หักบัญชีธนาคาร/ช่องทางATM

หมายเหตุบัตร

ชื่อบนบัตร

Somkiat Puisungnoen

วันที่บัตรหมดอายุ

CCV / CVV

ข้อมูลใบกำกับภาษีไม่สามารถเปลี่ยนแปลงได้หลังการสั่งซื้อสินค้า

ล็อก สั่งซื้อสินค้า

ตรวจสอบใบกำกับภาษีเพื่อทราบผลตรวจสอบเลขอย่างถูกต้อง

ส่งที่ แยก
Somkiat Puisungnoen
122/64 , Sci Phahonyothin 2, Phahonyothin Road Prom Condo กรุงเทพมหานคร/ Bangkok - พญาไท/ Phaya Thai - 10400 โทรศัพท์: 0868696209

สรุปการสั่งซื้อ (1 items)

รายการ	จำนวน	ราคารวม
POCA SHOE NMD Sneakers Fashion รองเท้า ลำลอง ถ้าไม่สูงสู่-สูงสุด เฟชั่น ราคาถูกสุดๆ Scott Unisex รุ่น PSN-Black/White ขนาด: EU:40	1	399 มากที่สุด

สั่งแบบธรรมด้า
วันอัจฉริยา, 27 ก.พ. - วันเสาร์, 3 มี.ค. 2018

กรอกคุณปวงส่วนลดที่นี่

ขึ้นชั้น

มูลค่าสินค้า
ค่าใช้จ่ายค่า

ยอดสุทธิ
รวมภาษีมูลค่าเพิ่ม (มีภาษี)

399 บาท
หัก

399 บาท

ทุก笔ของคุณค้า 100%

Verified by VISA MasterCard SecureCode JUN-2016 more security - less risk to you

โดยการวางแผนค่าสั่งซื้อของคุณ, คุณจะได้รับข้อเสนอแนะจากเราในการซื้อสินค้าทางช่องทางที่กำหนดให้ และ ร้อยละและเพื่อไป



10. Summary



ใบแจ้งการชำระเงิน(PaySlip)

Counter Service Co., Ltd.

เลขที่ใบแจ้ง สินค้า/Invoice No:	3779254692
ผู้ชำระ เงิน/Payer:	Somkiat Puisungnoen
วันที่รายการ / Transaction Date:	25/02/2018 23:33
กำหนดชำระเงิน / Expired Date:	27/02/2018 23:33
เพื่อเข้าบัญชี / Payee:	www.lazada.co.th Tel: 020180000
รายละเอียด / Detail:	Lazada



806010855864737

จำนวนเงินที่ชำระ / Amount:

399.00 บาท /THB

* ไม่รวมค่าธรรมเนียมของเด่านี้เดอร์เซอร์วิส
(Excluding service fees at Counter Service)

คลิกปุ่ม "Print" พิมพ์ใบแจ้งการชำระเงิน
หรือ



กด "รหัส 15 หลักใต้بارك็อด" เพื่อนำไป
ชำระเงินที่
Press "Print" button or write down
paycode 15 digits for pay in cash at
counter service(7-11)

[Back to merchant](#)

[Print](#)



Microservices

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

Try to design system



A-DAPT Blueprint

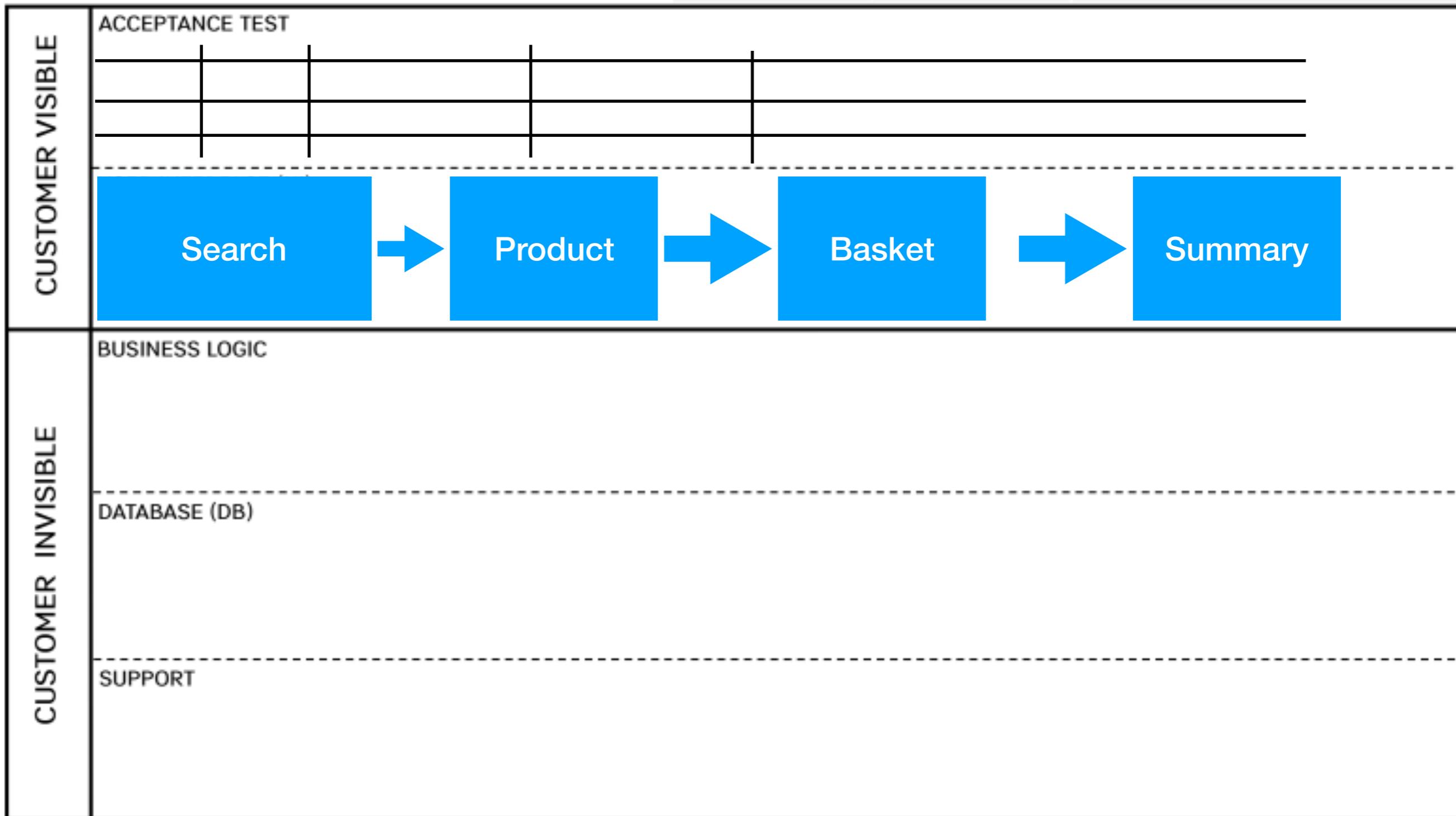
THEME:	EPIC:	FEATURE:	STORY:
DESIGNED BY:	DATE:	NOTE:	



A-DAPT Blueprint

THEME: EPIC: FEATURE: STORY:

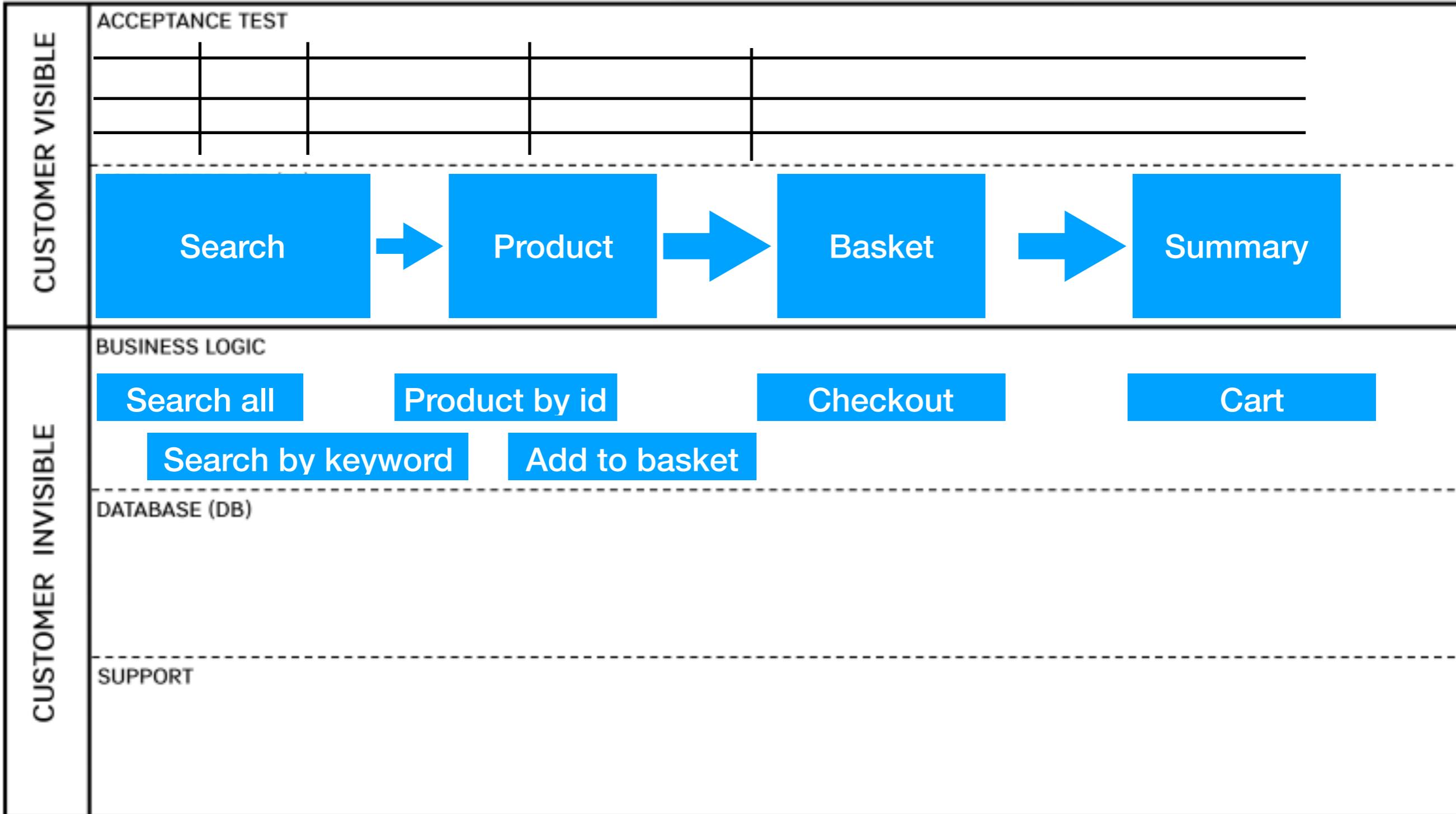
DESIGNED BY: DATE: NOTE:



A-DAPT Blueprint

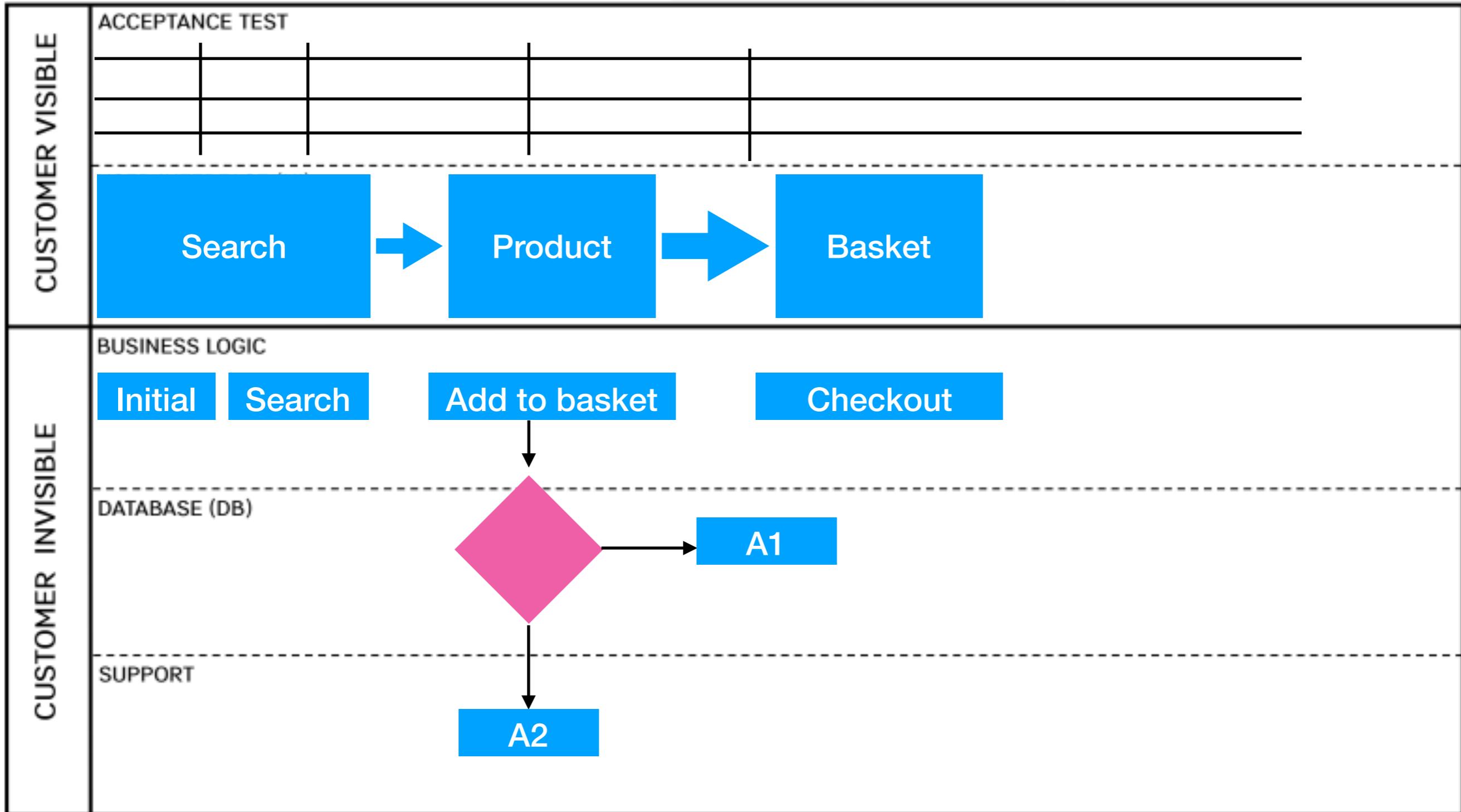
THEME: EPIC: FEATURE: STORY:

DESIGNED BY: DATE: NOTE:



A-DAPT Blueprint

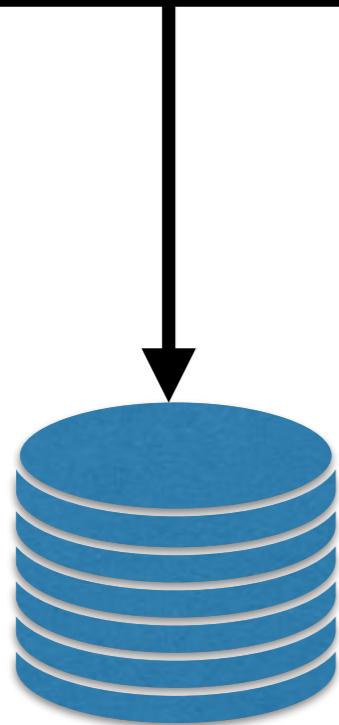
THEME:	EPIC:	FEATURE:	STORY:
DESIGNED BY:	DATE:	NOTE:	



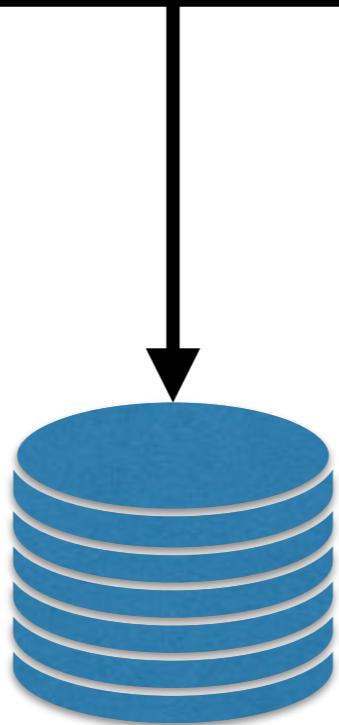
Let's discuss about splitting service



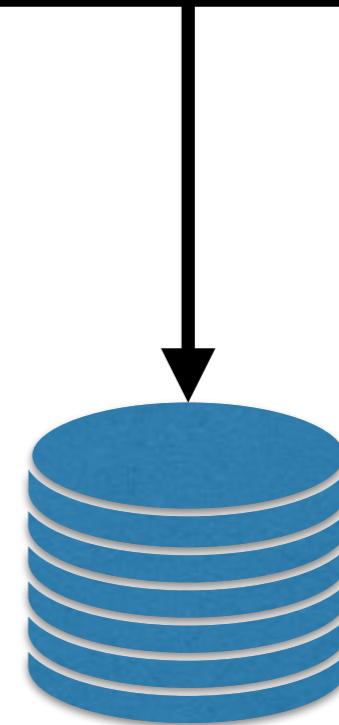
Service 1

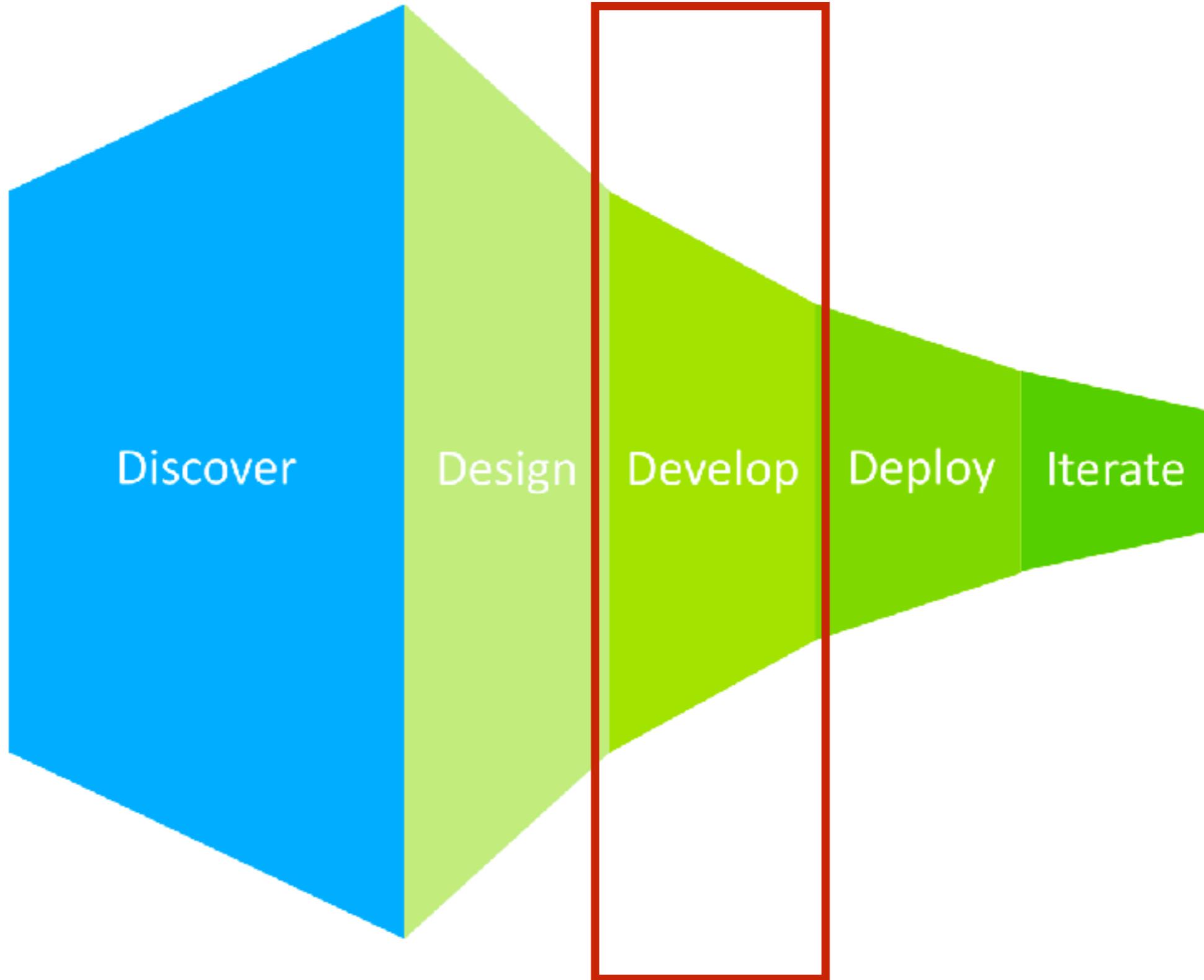


Service 2

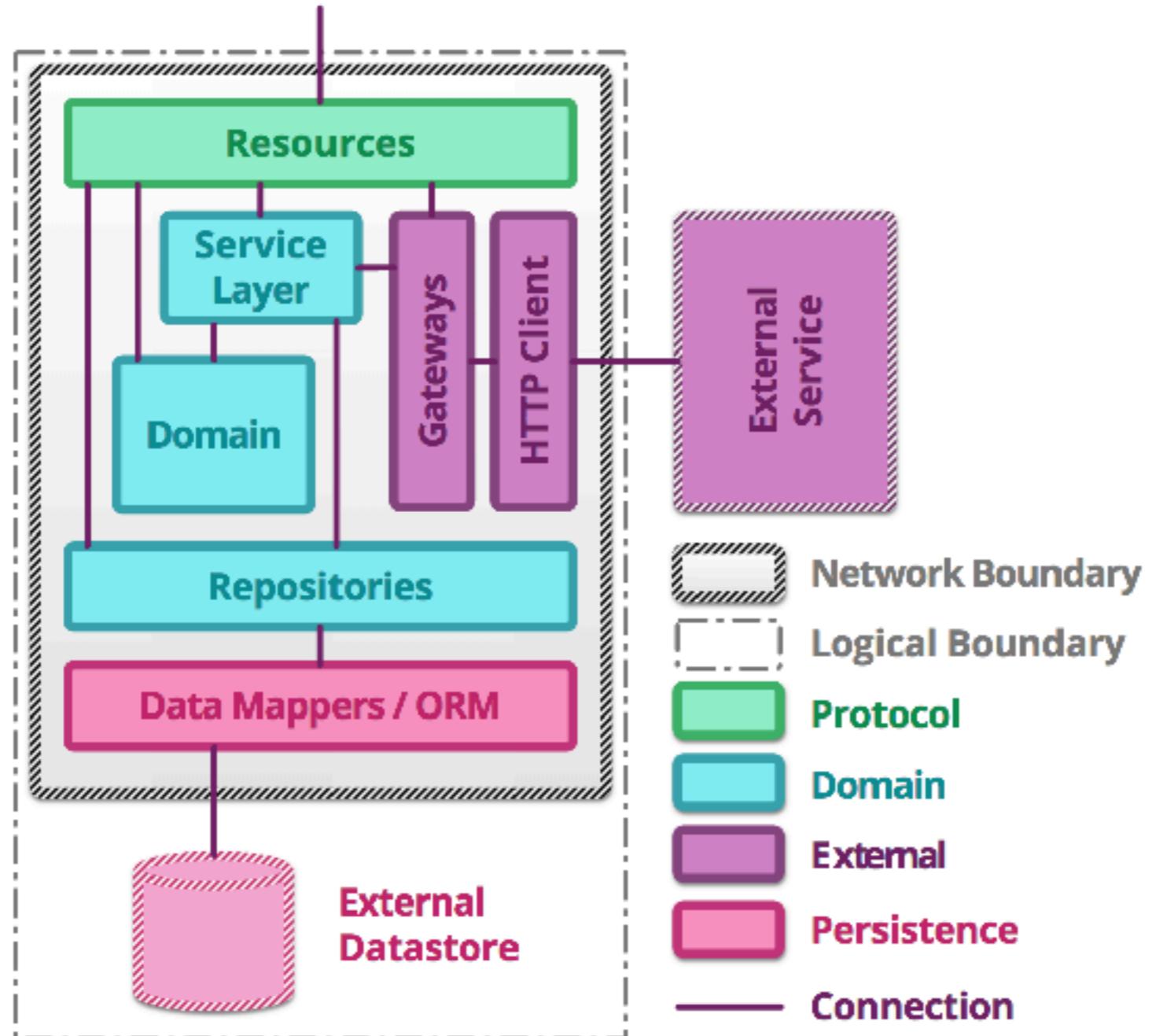


Service 3





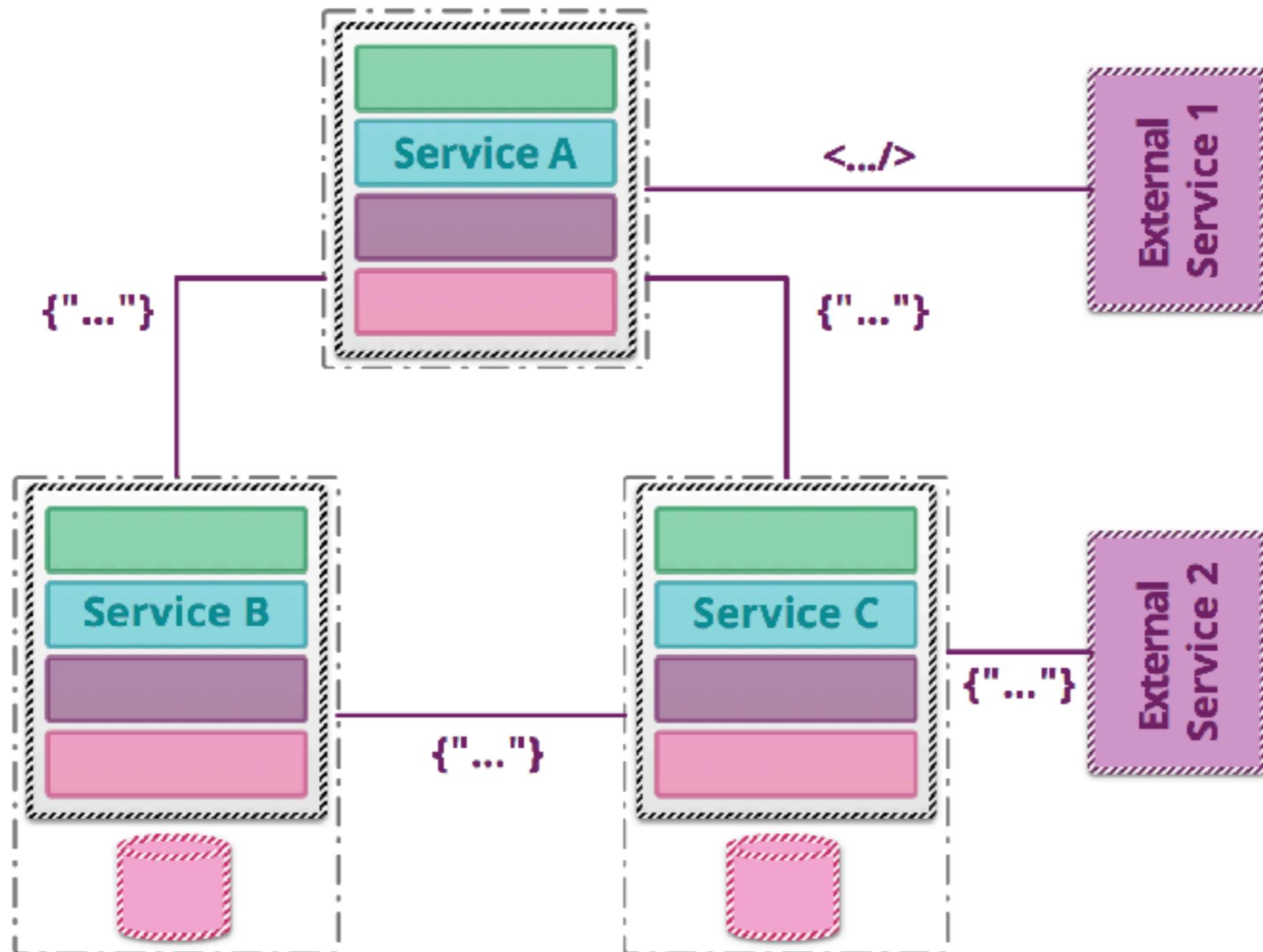
Service structure



<https://martinfowler.com/articles/microservice-testing>



Multiple services

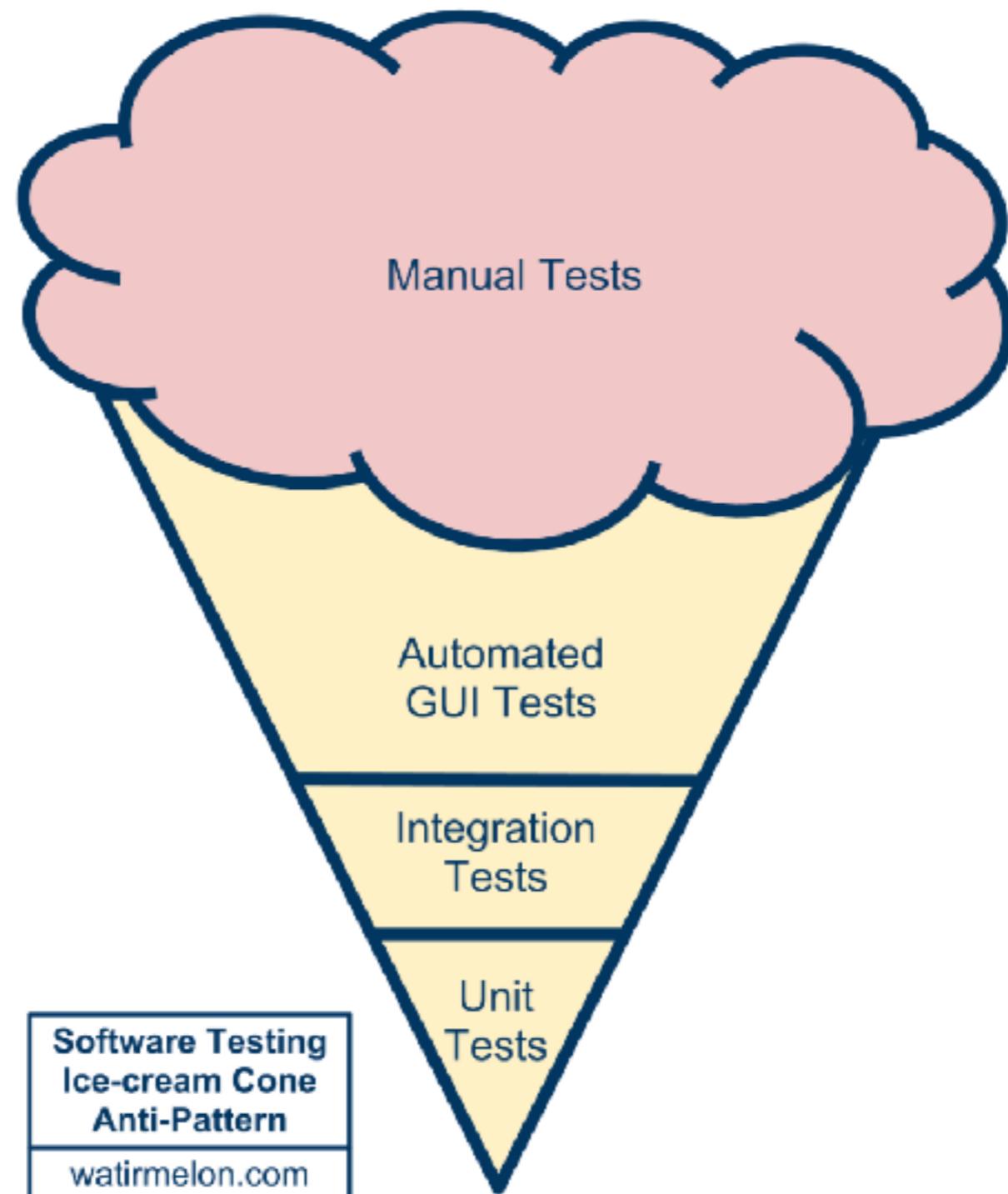


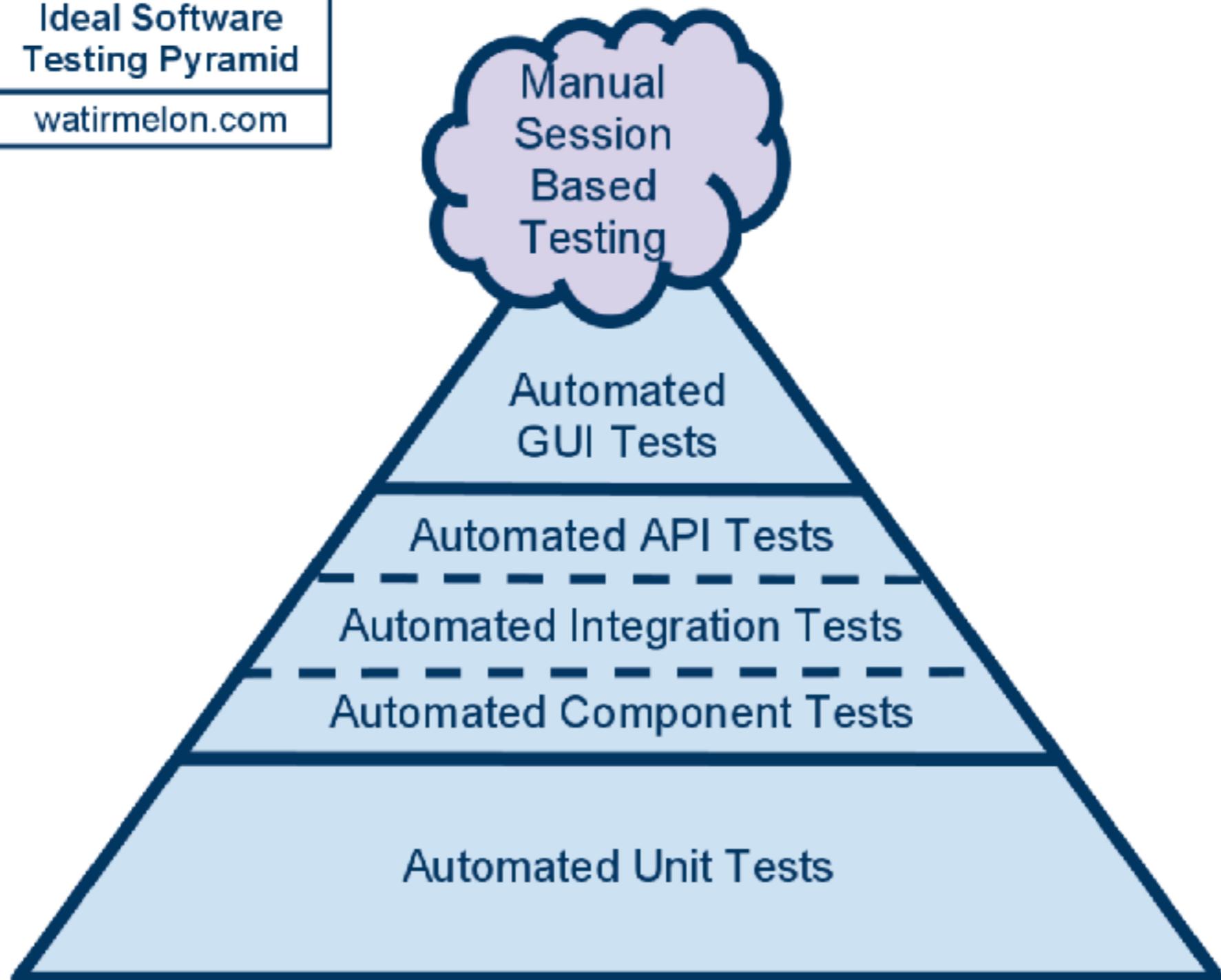
<https://martinfowler.com/articles/microservice-testing>

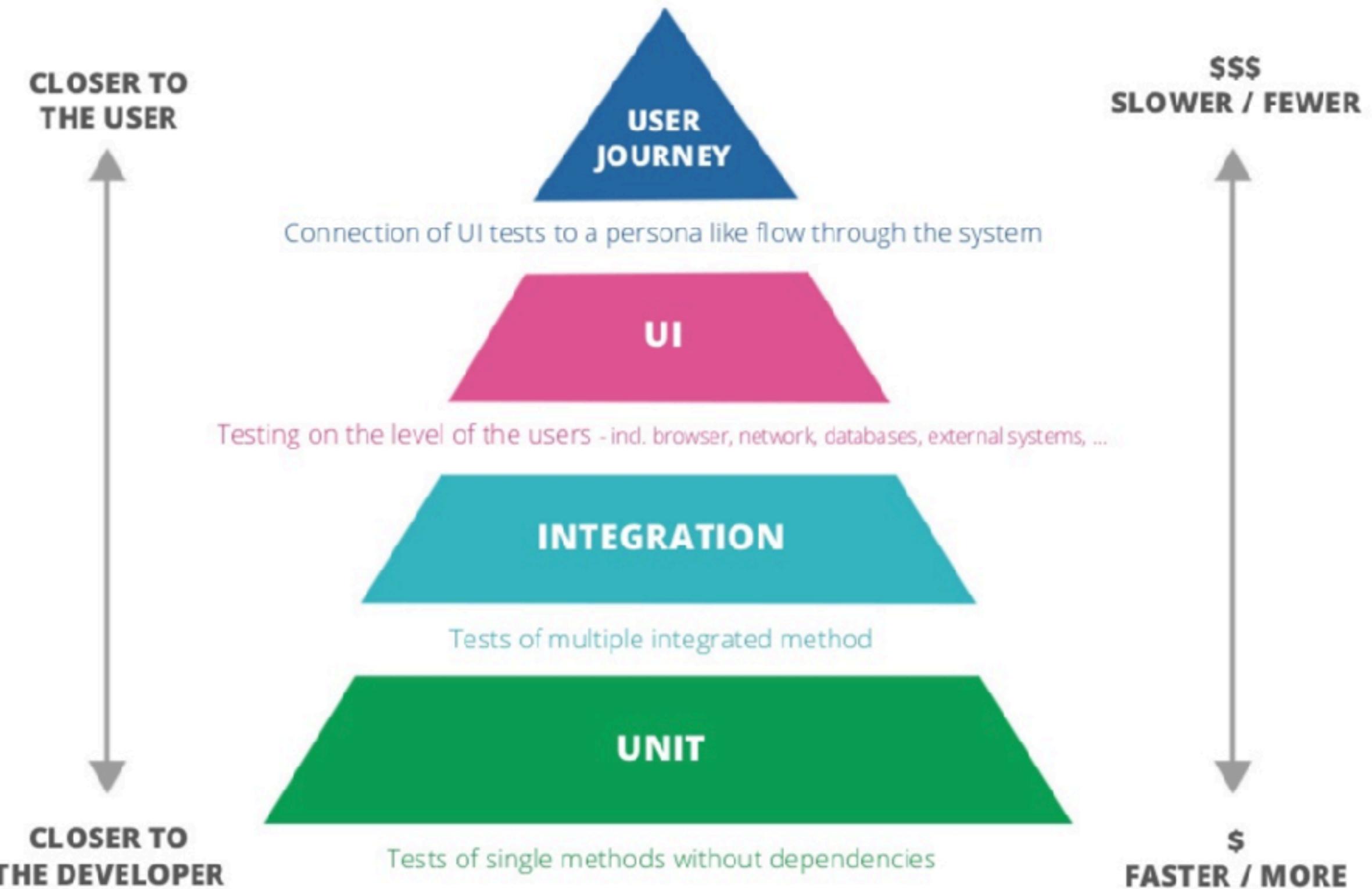


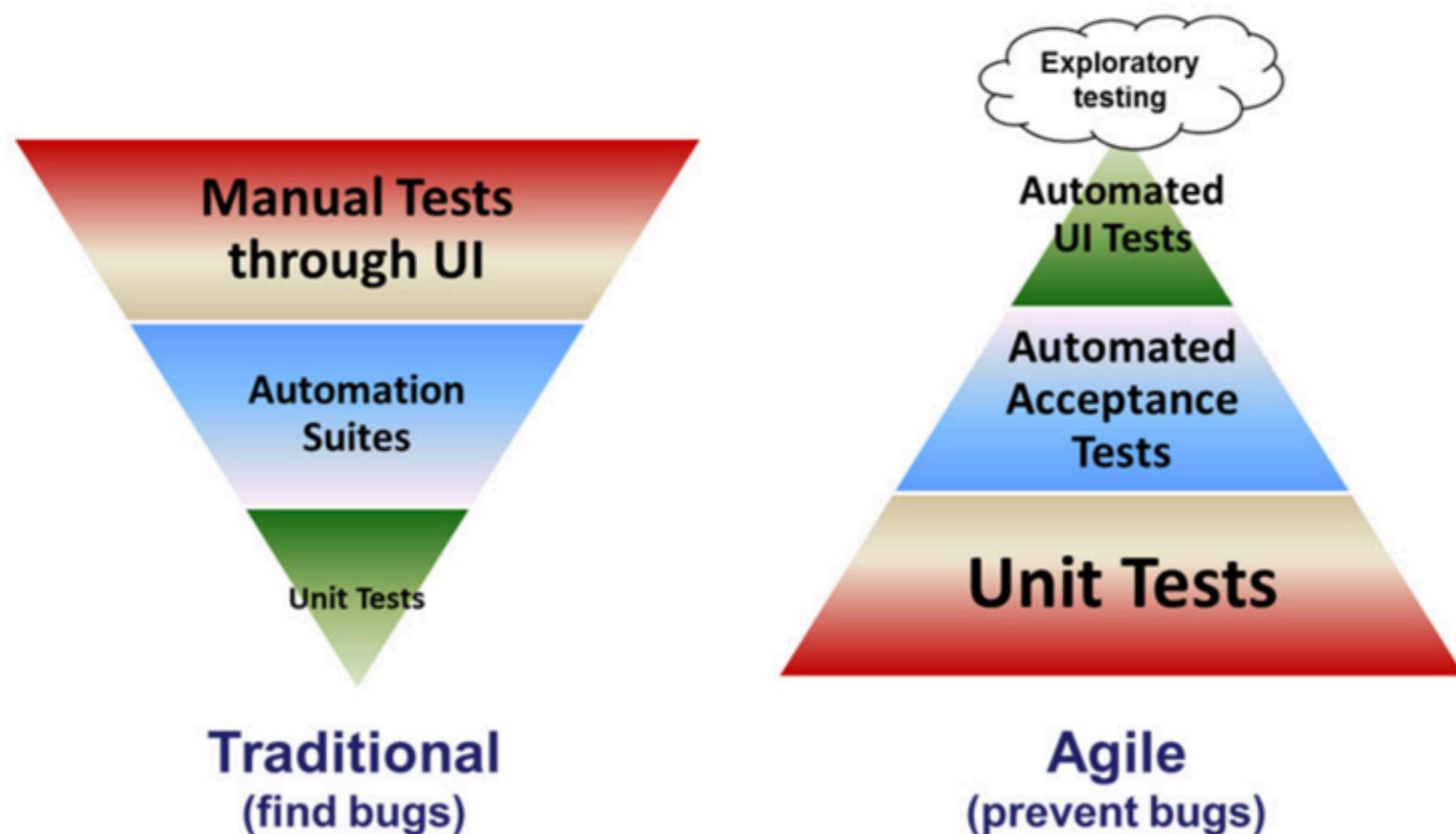
Microservice Testing





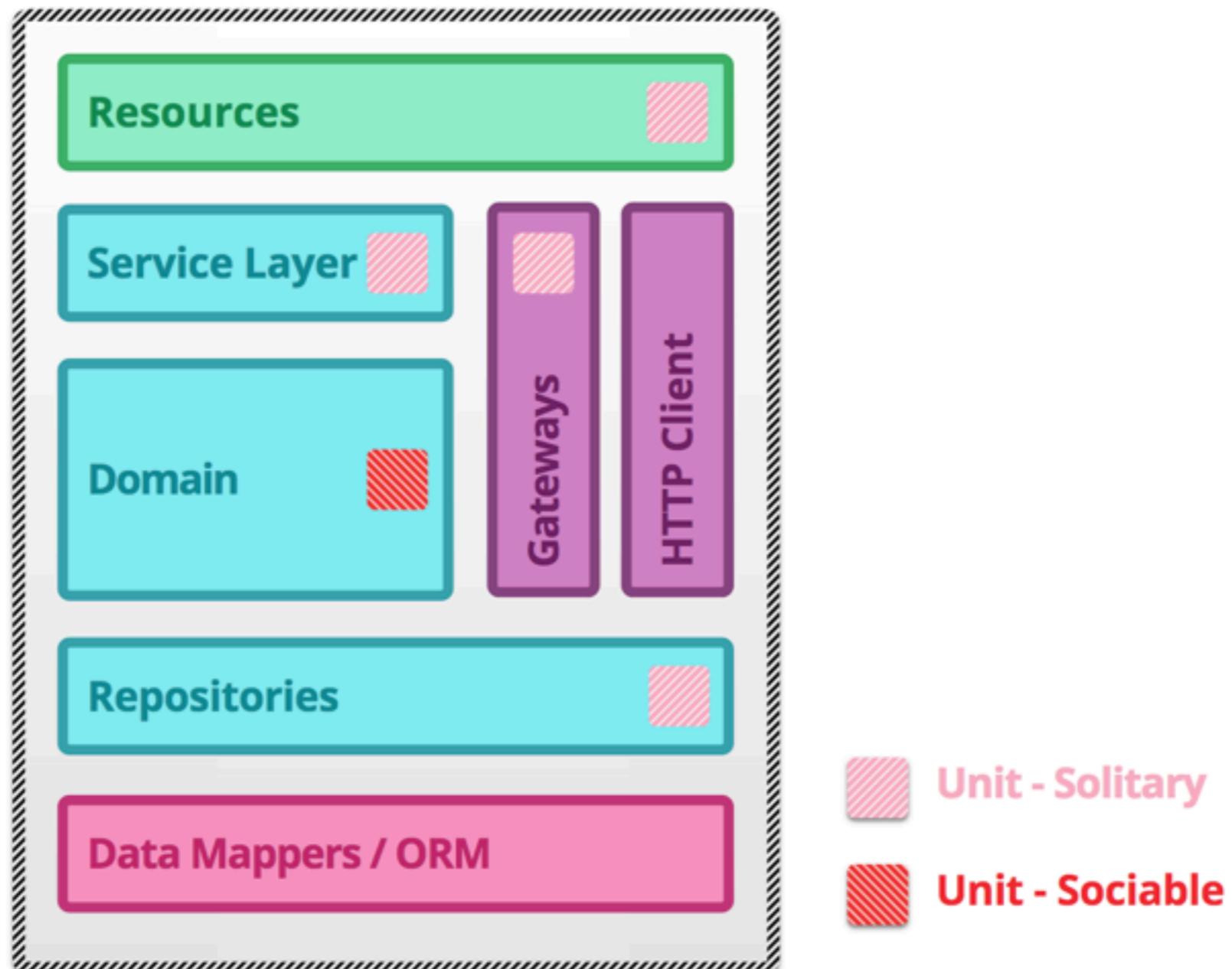




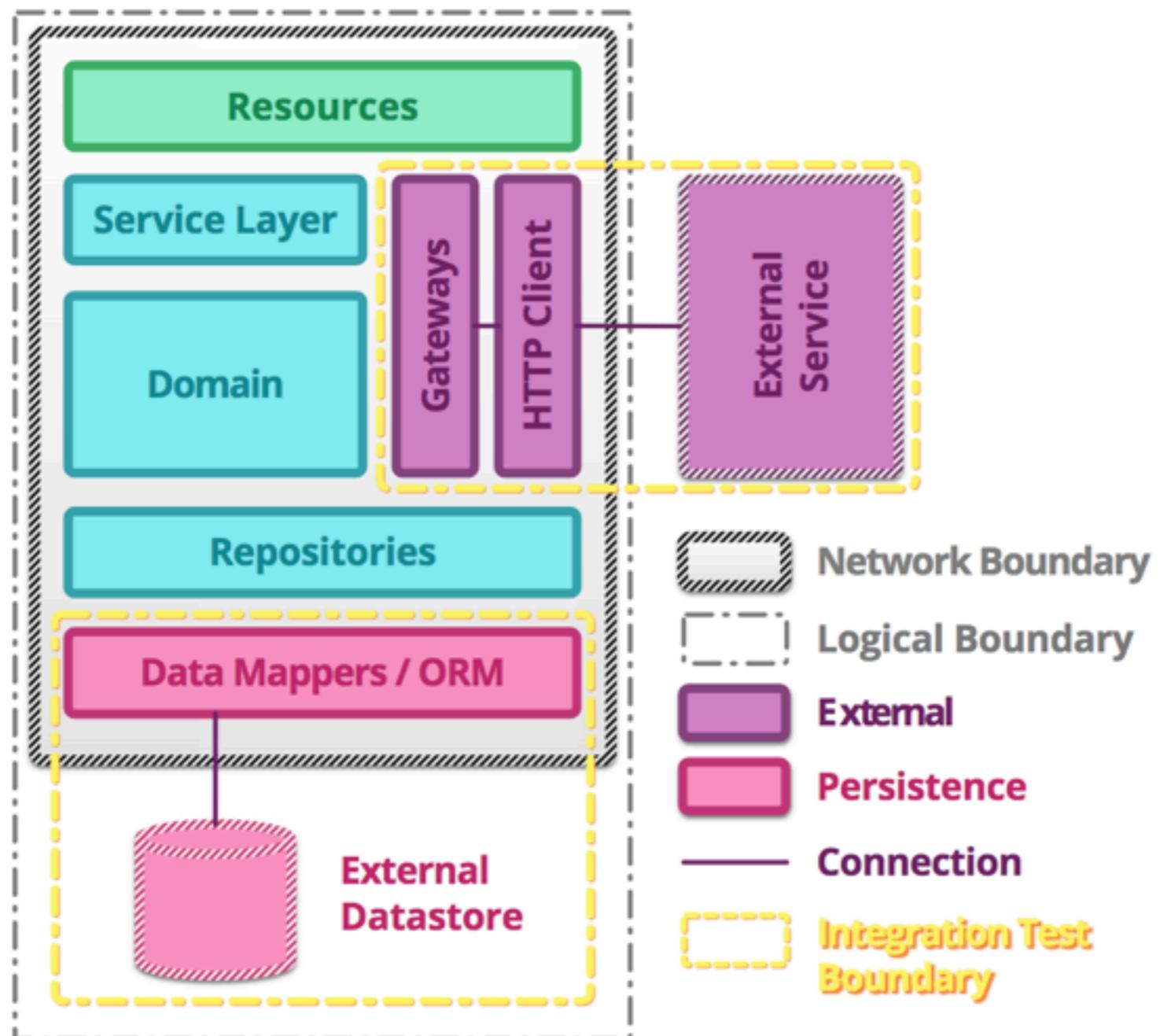




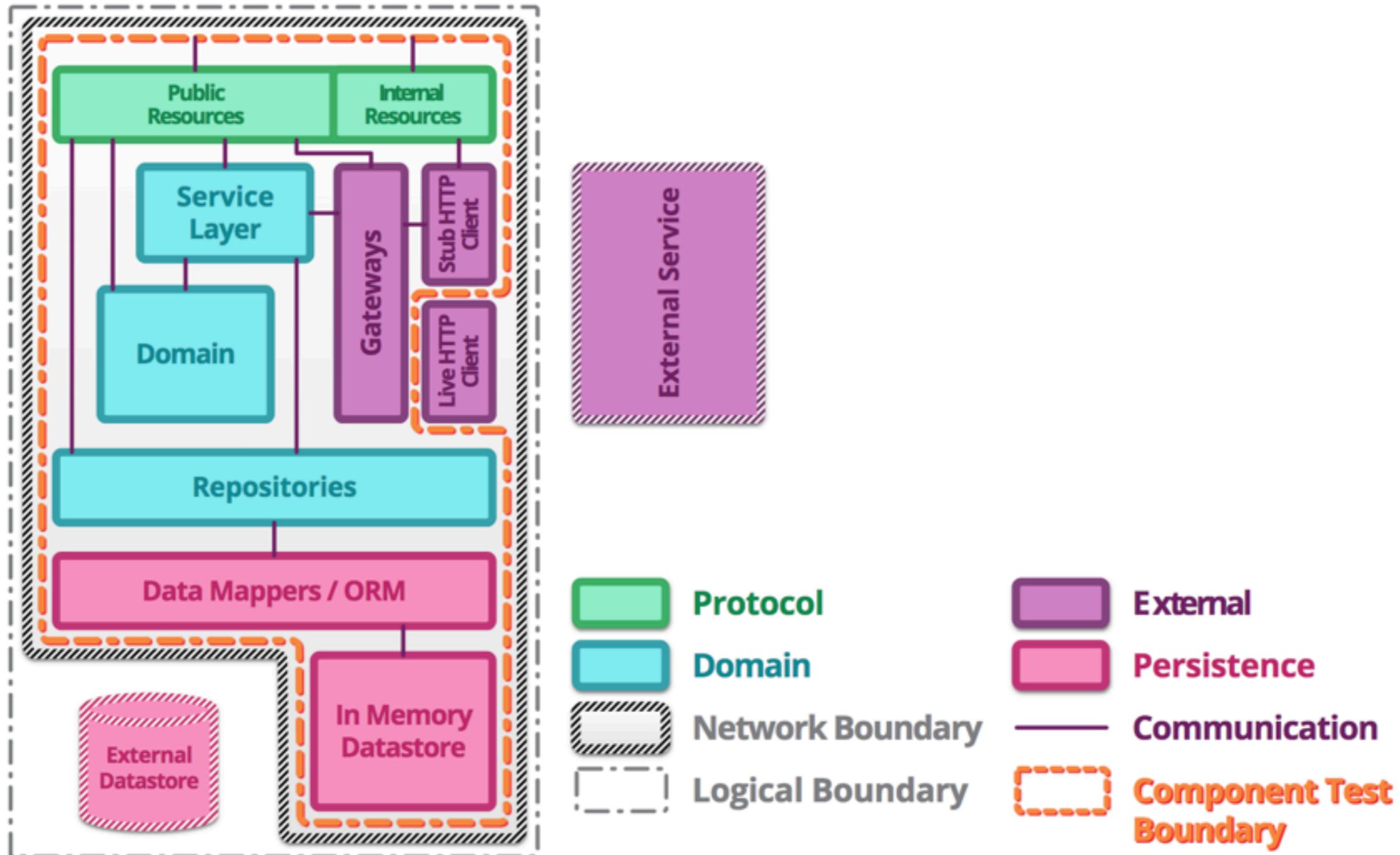
Unit testing



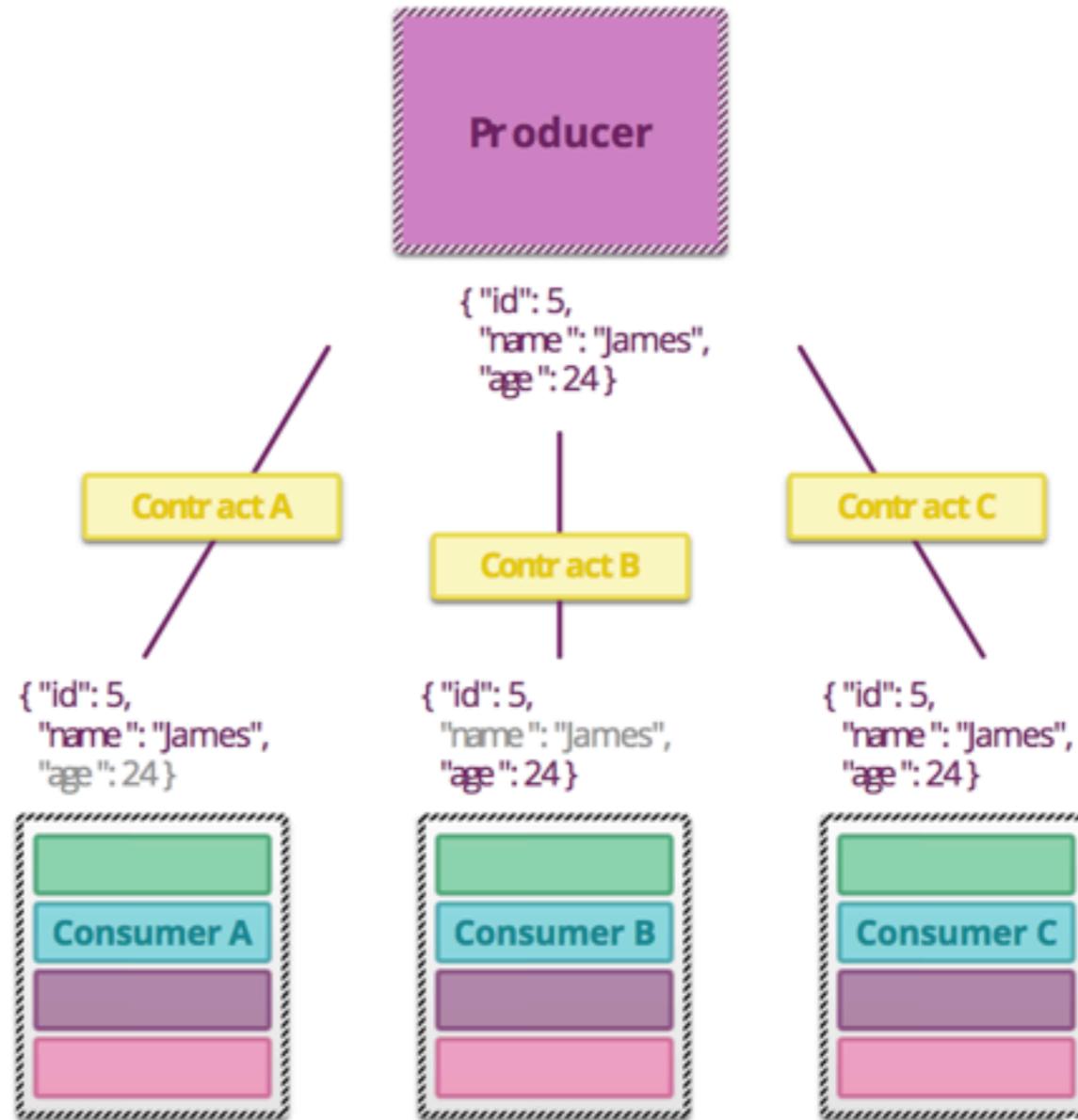
Integration testing



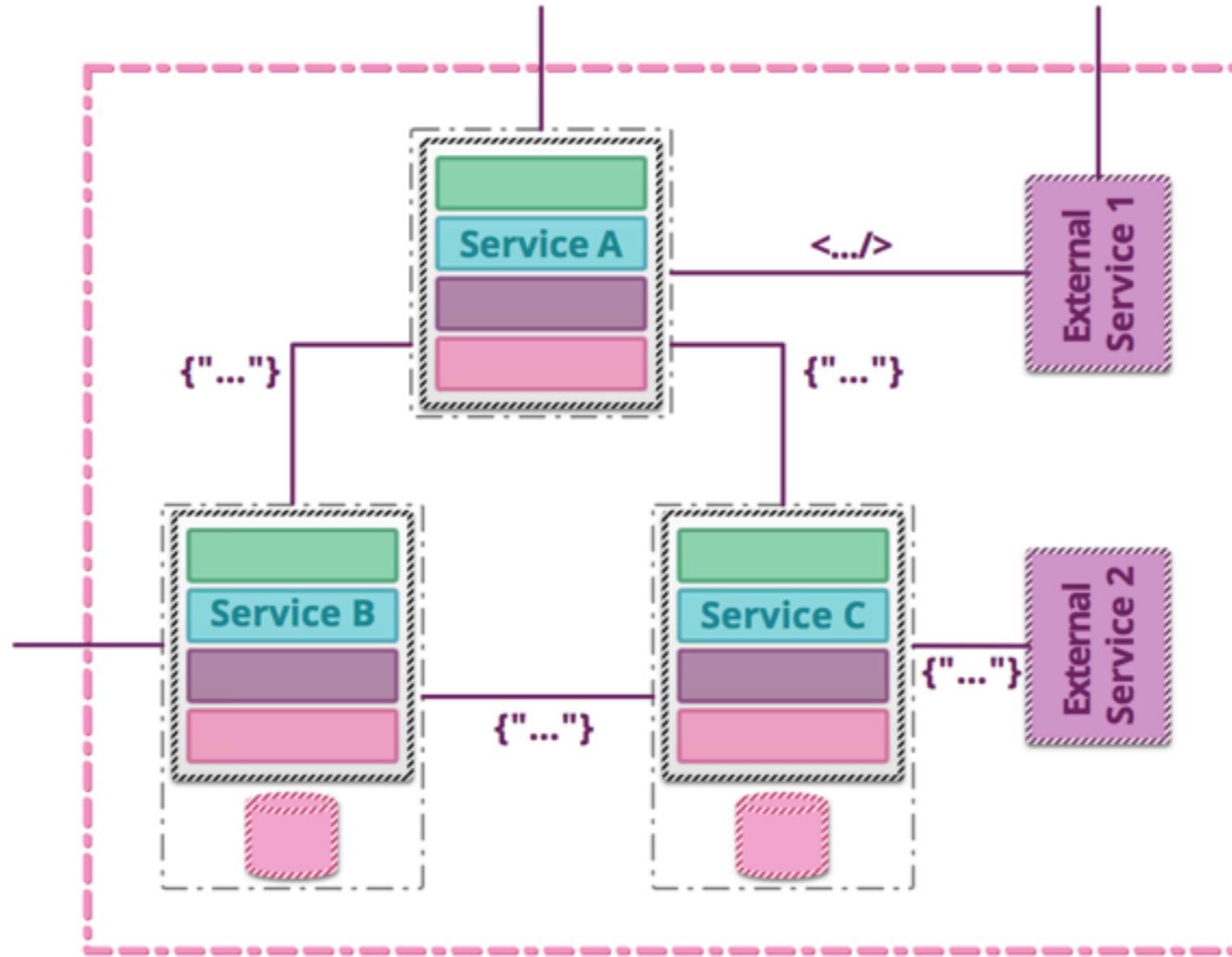
Component testing



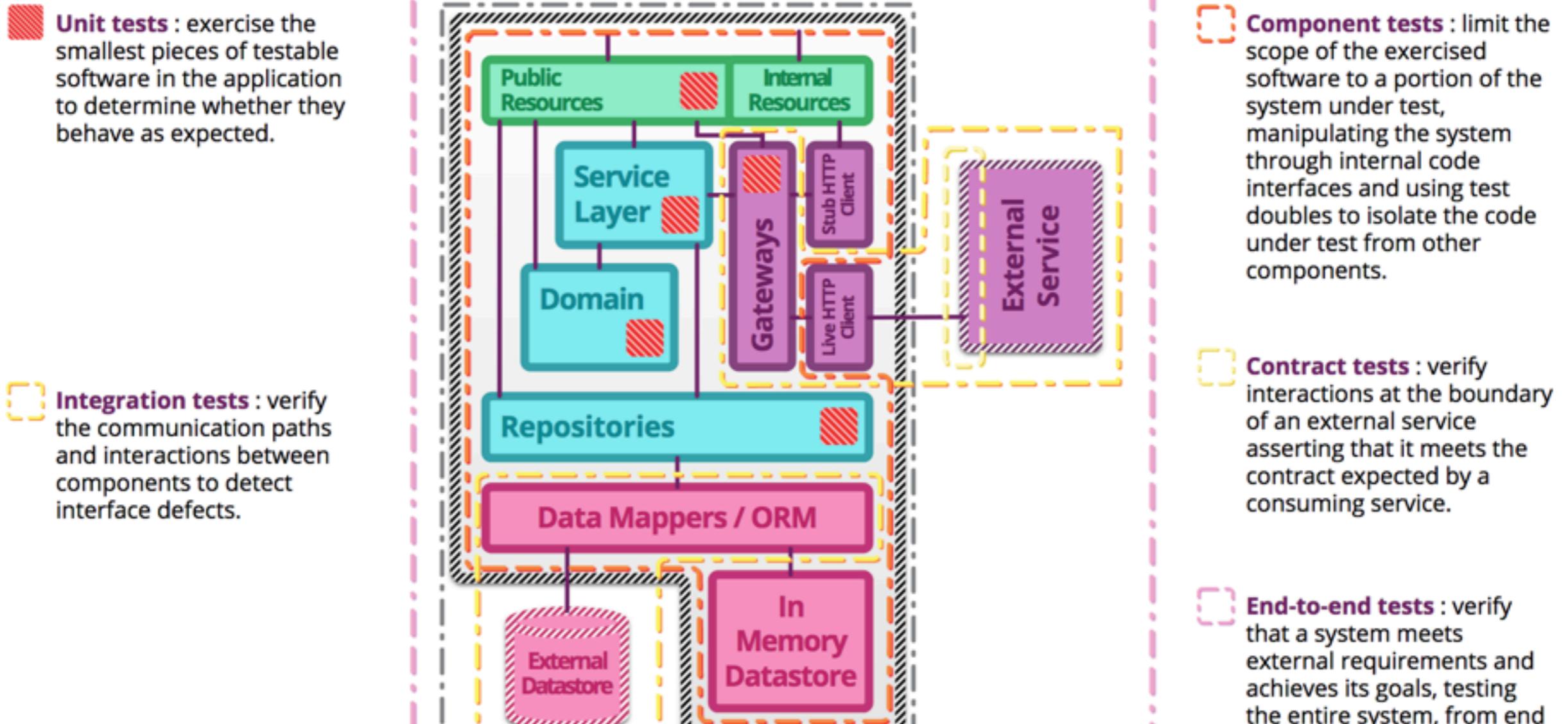
Contract testing



End-to-End testing



Summary



More ...



Performance testing ?



Security testing ?



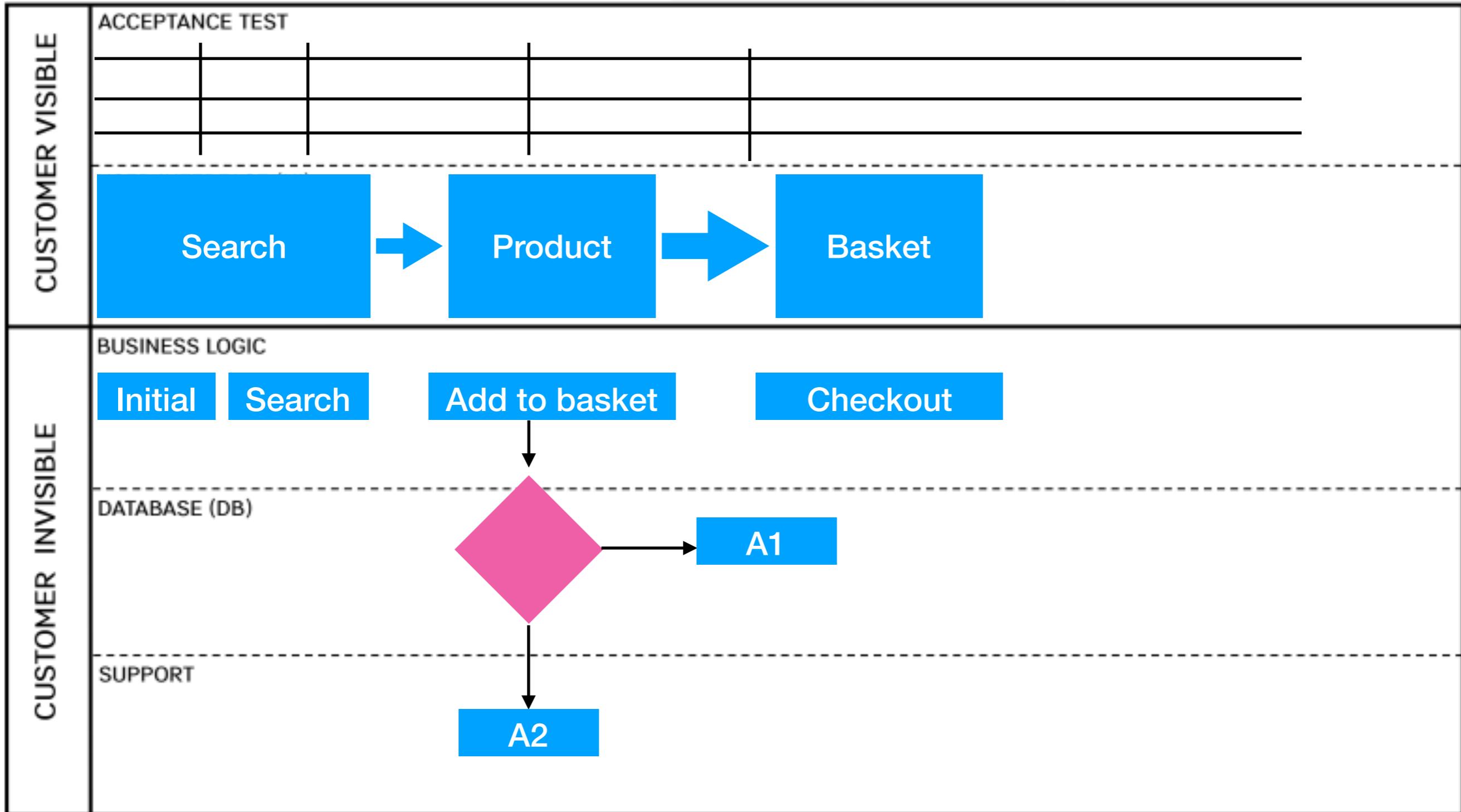
Workshop

How to test your services ?



A-DAPT Blueprint

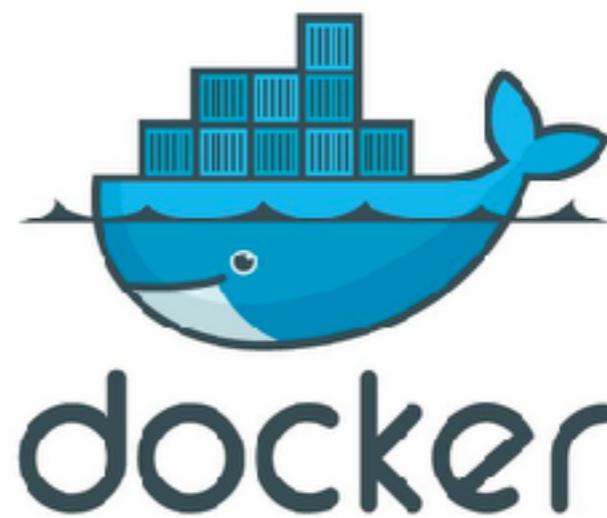
THEME:	EPIC:	FEATURE:	STORY:
DESIGNED BY:	DATE:	NOTE:	



Let's start to develop



Go



Good microservices

Circuit breaking

Load balancing

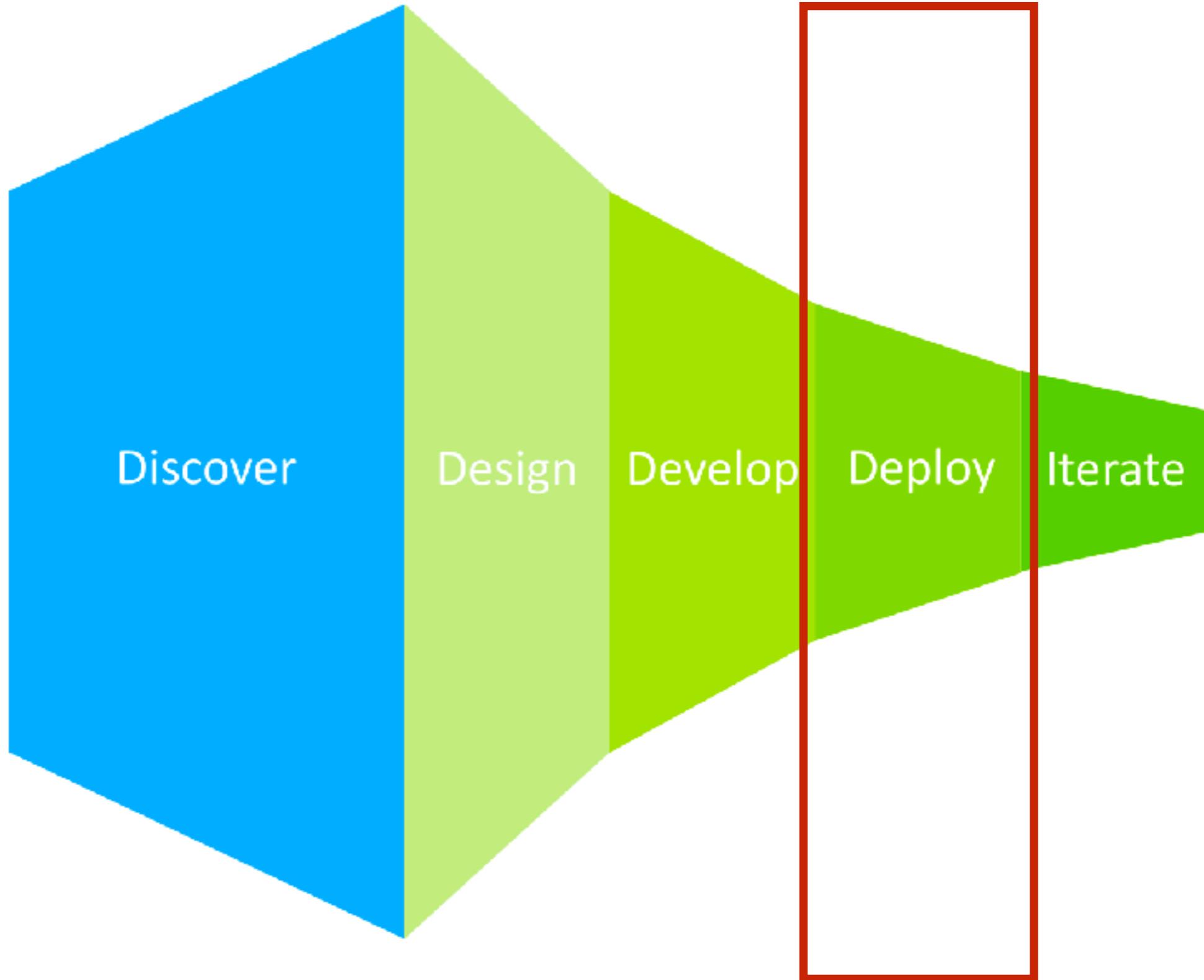
Service registry/discovery

Health check

Application monitoring

Tracing service





How to deploy your services ?



Strategies to deploy

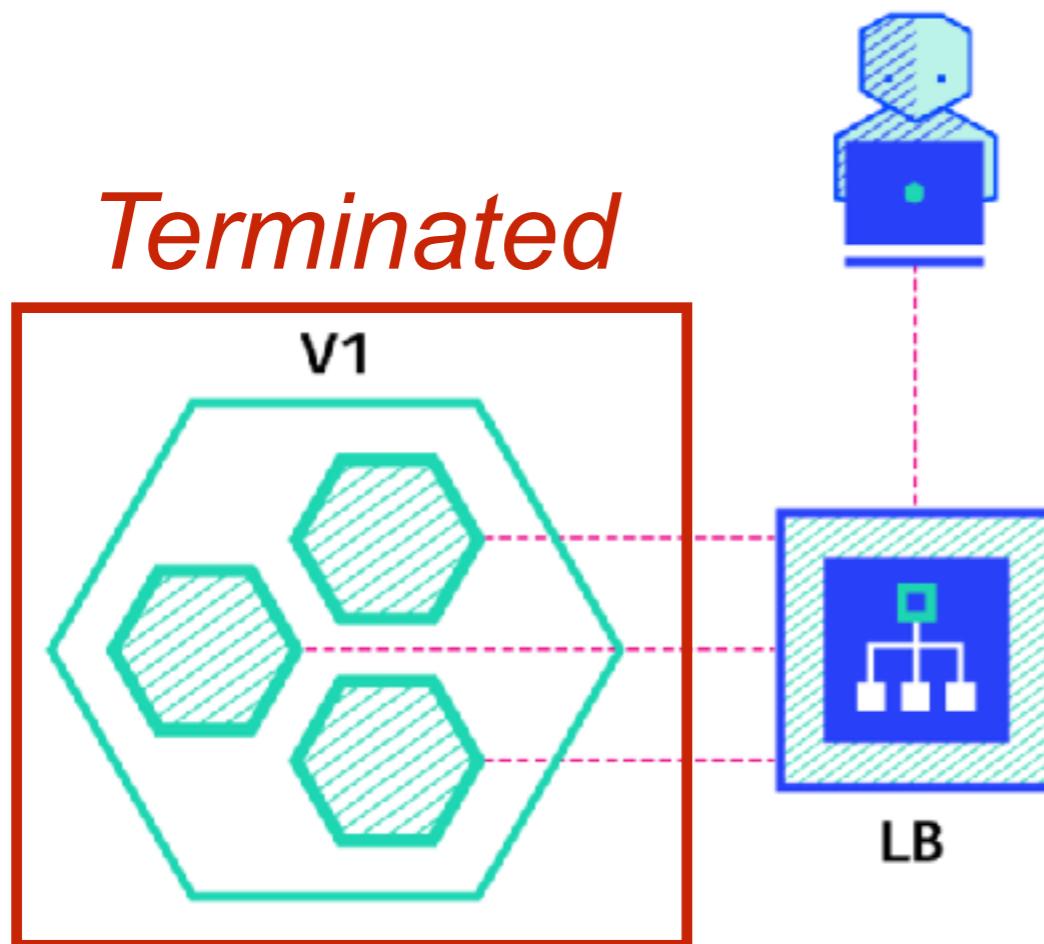
Recreate
Ramped
Blue/Green
Canary
A/B testing
Shadow

<https://thenewstack.io/deployment-strategies/>



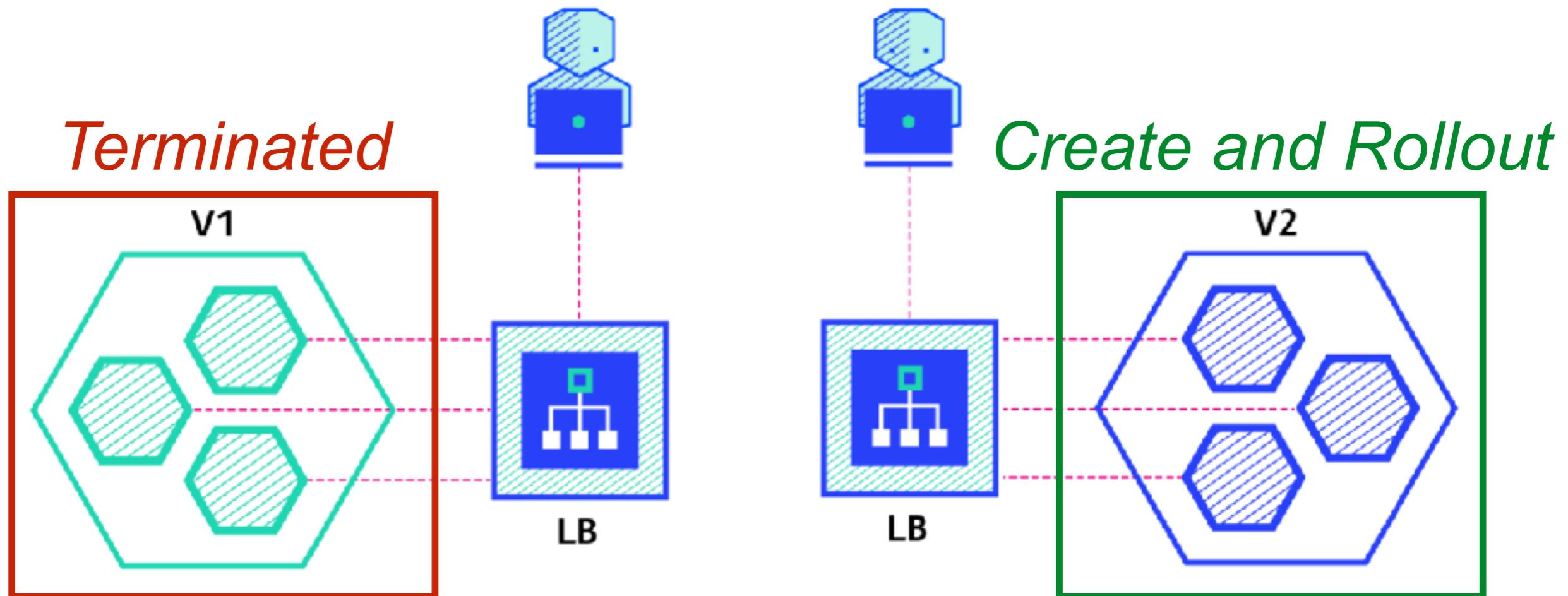
1. Recreate

Version A is terminated then version B is rollout



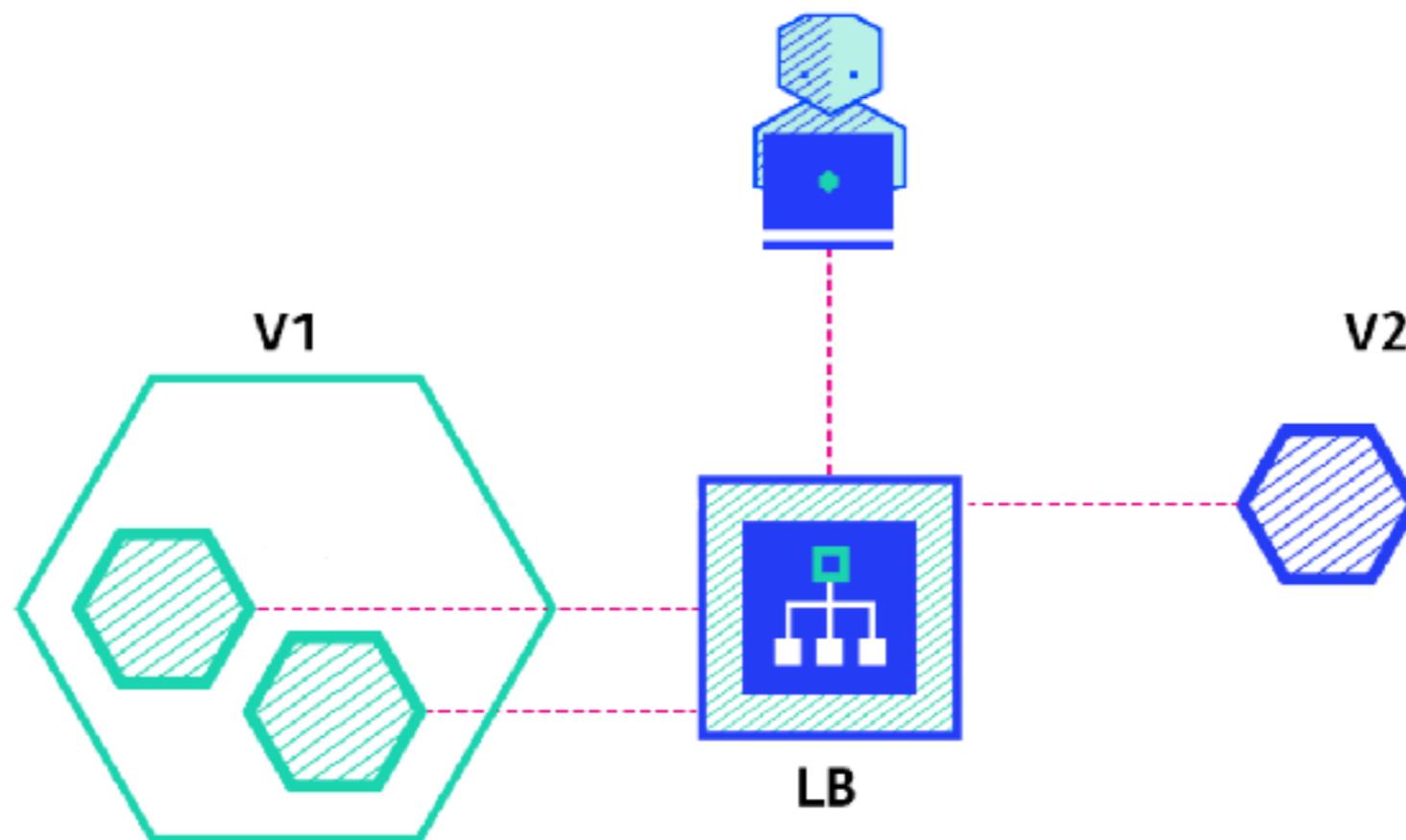
1. Recreate

Version A is terminated then version B is rollout



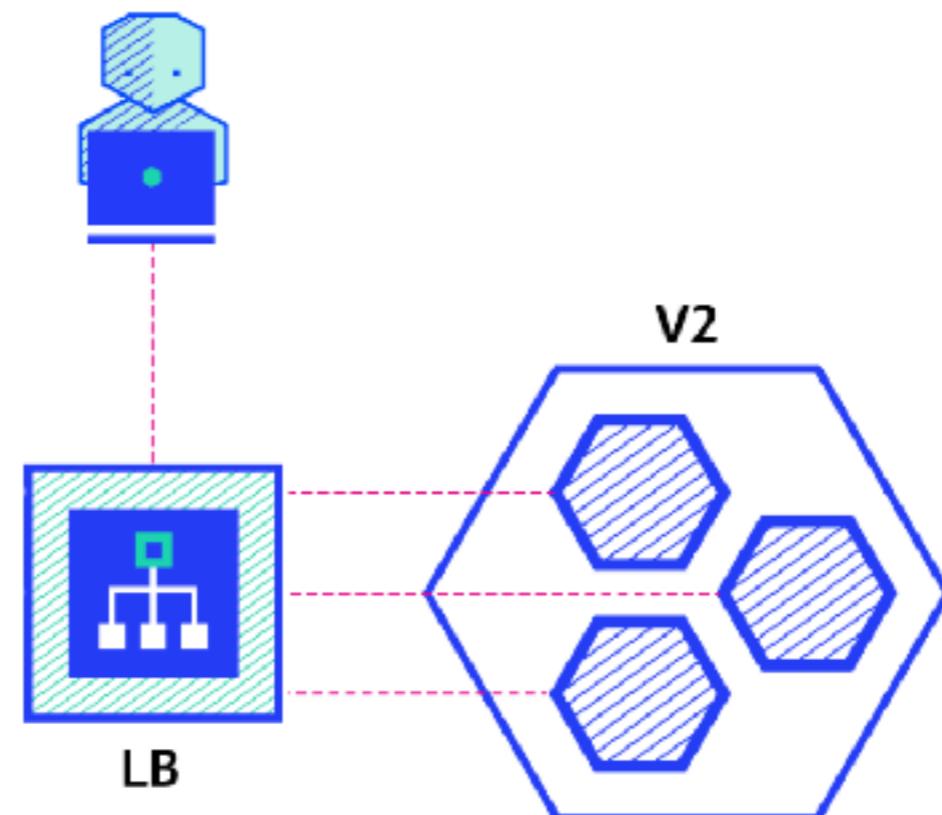
2. Ramped

Slow roll out by replace instance one-by-one



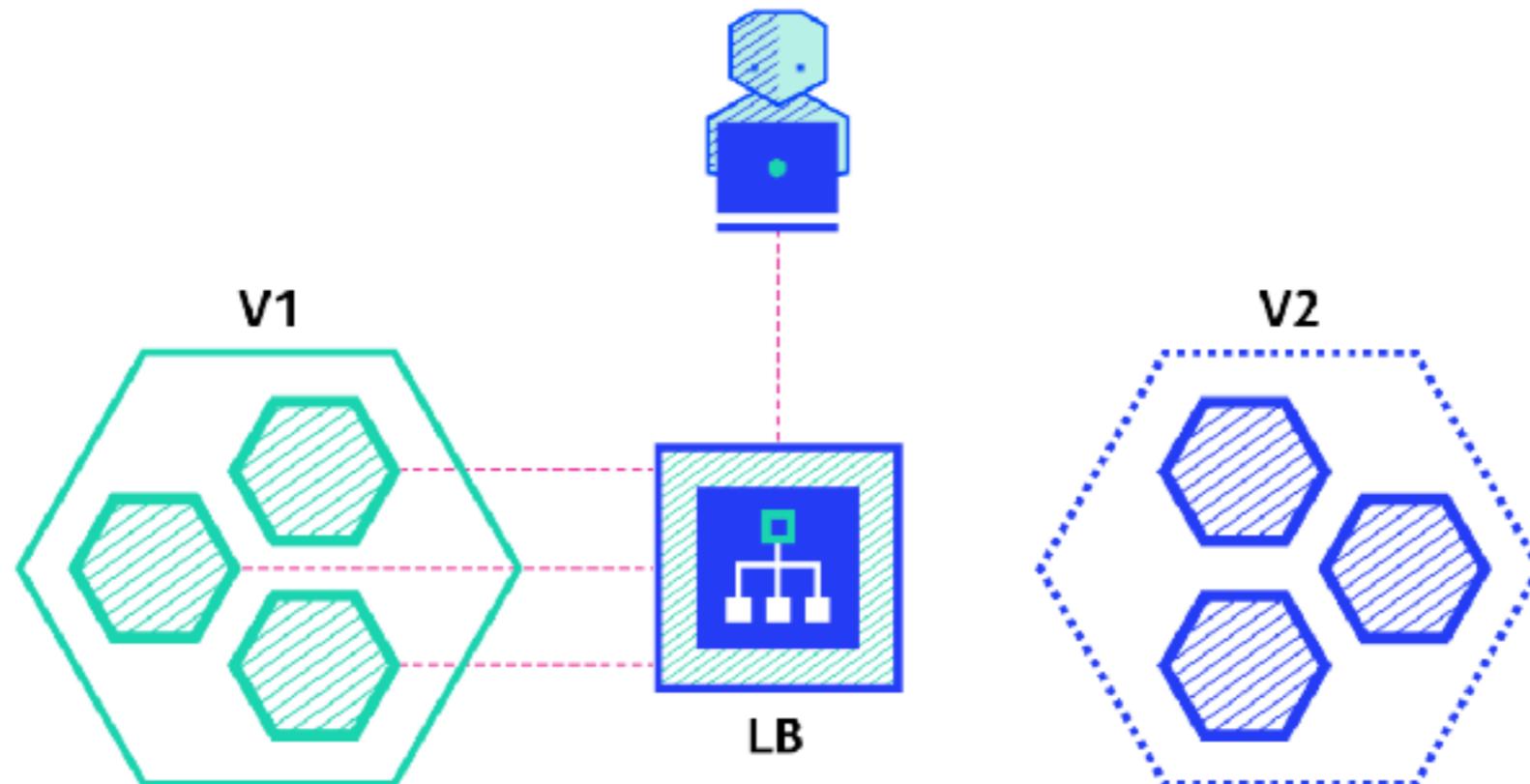
2. Ramped

Slow roll out by replace instance one-by-one



3. Blue/Green

Current version is called **Blue**
New version is called **Green**

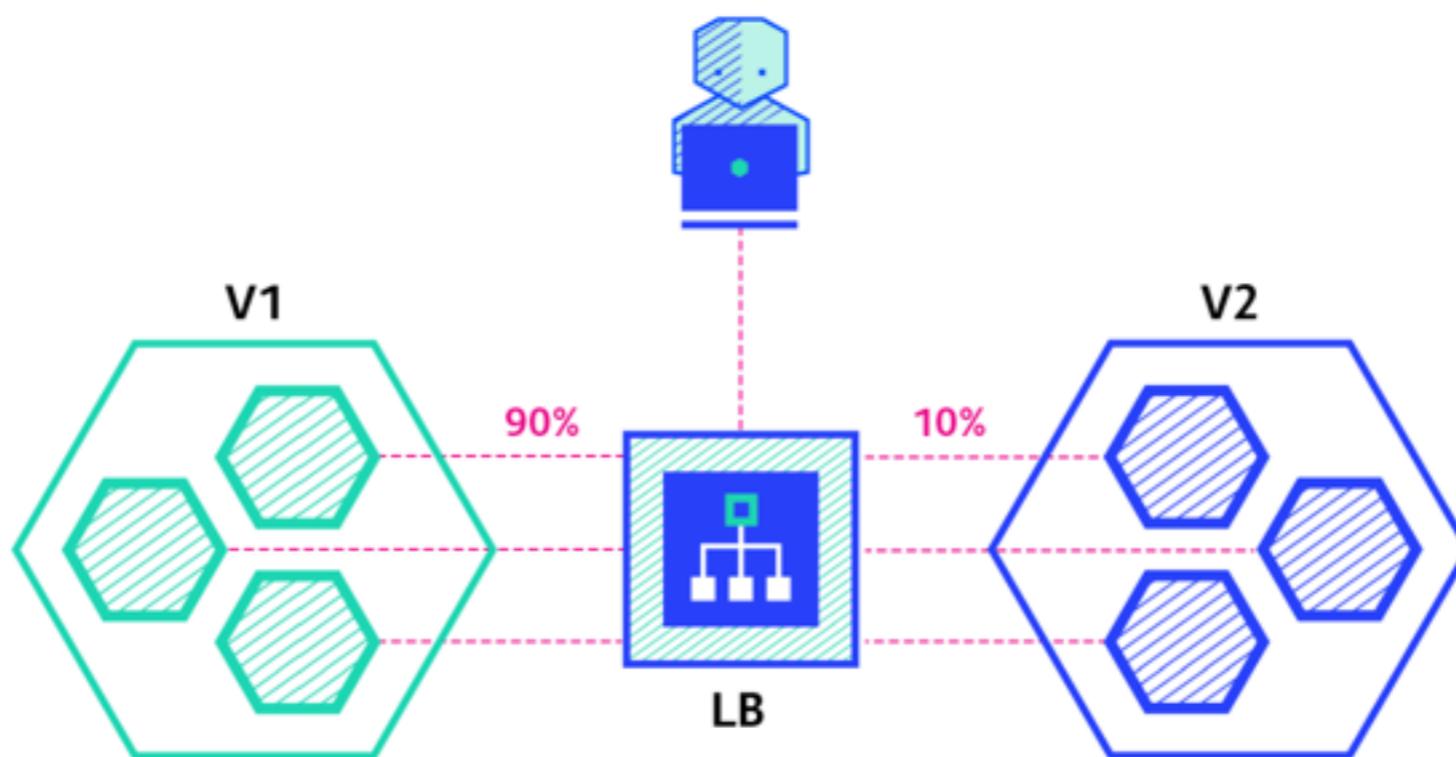


4. Canary

Shift production traffic from version A to B

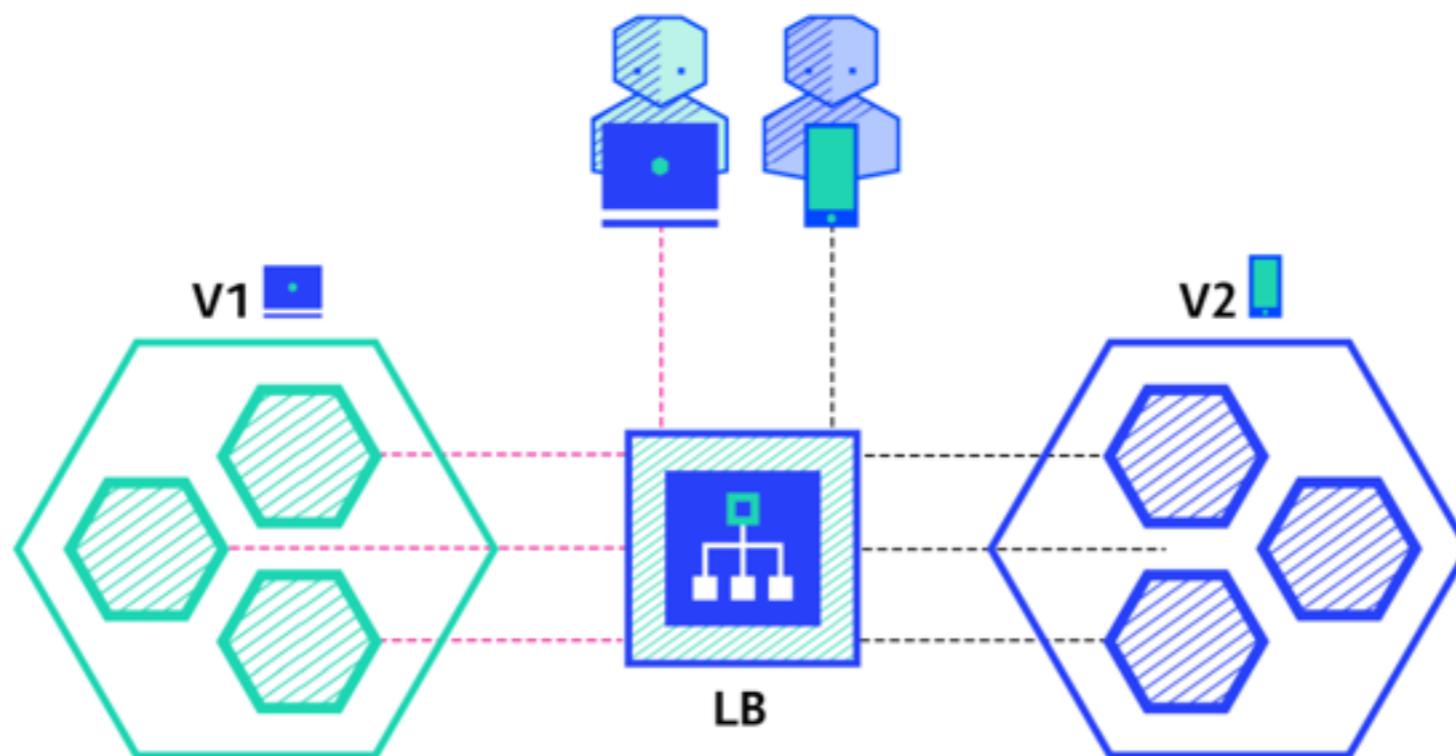
Traffic is split based-on weight

Use when tests are lacking/not reliable and less confident in system



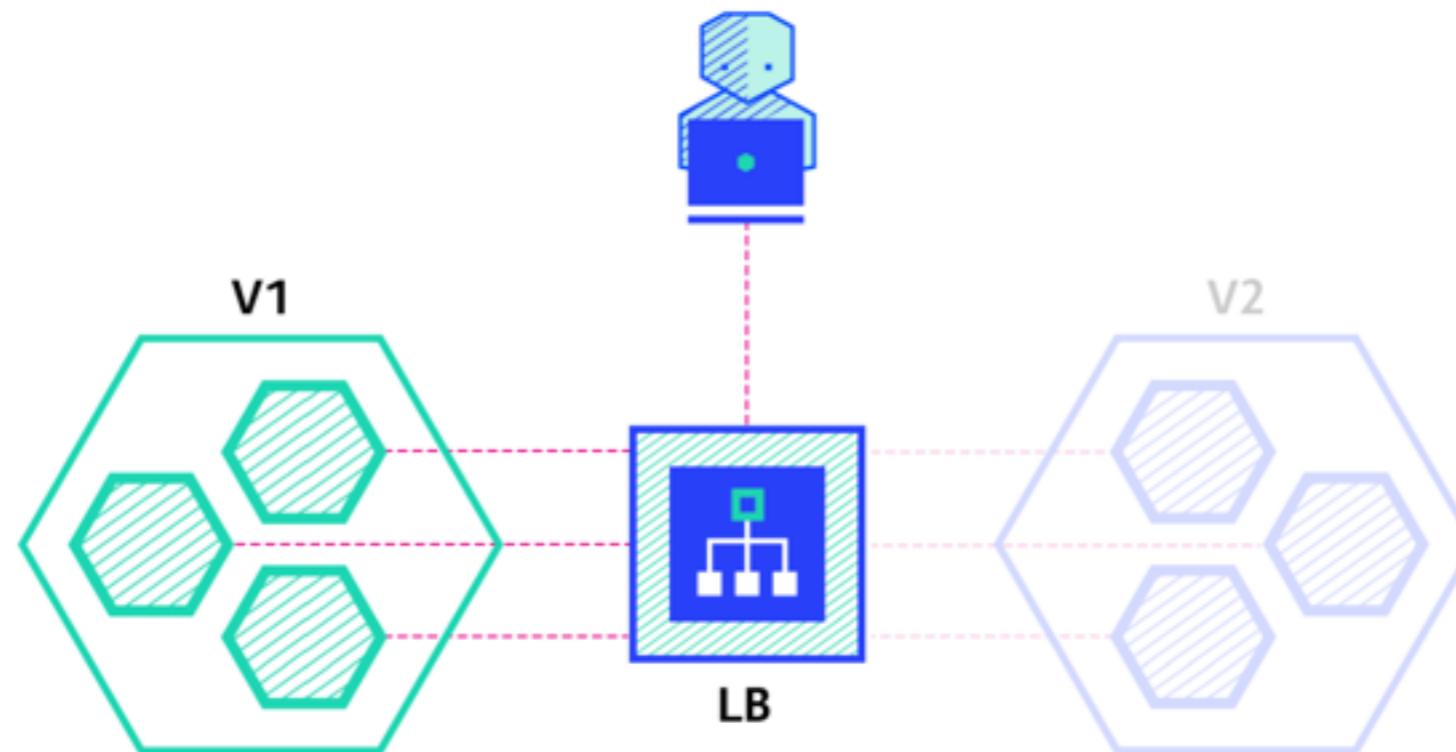
5. A/B testing

Routing the subset of users to new services under the specific condition



6. Shadow

Release version B alongside version A
Send request's A to B without production impact



DEPLOYMENT STRATEGIES

When it comes to production, a ramped or blue/green deployment is usually a good fit, but proper testing of the new platform is necessary.

Blue/green and shadow strategies have more impact on the budget as it requires double resource capacity. If the application lacks in tests or if there is little confidence about the impact/stability of the software, then a canary, a/b testing or shadow release can be used.

If your business requires testing of a new feature amongst a specific pool of users that can be filtered depending on some parameters like geolocation, language, operating system or browser features, then you may want to use the a/b testing technique.



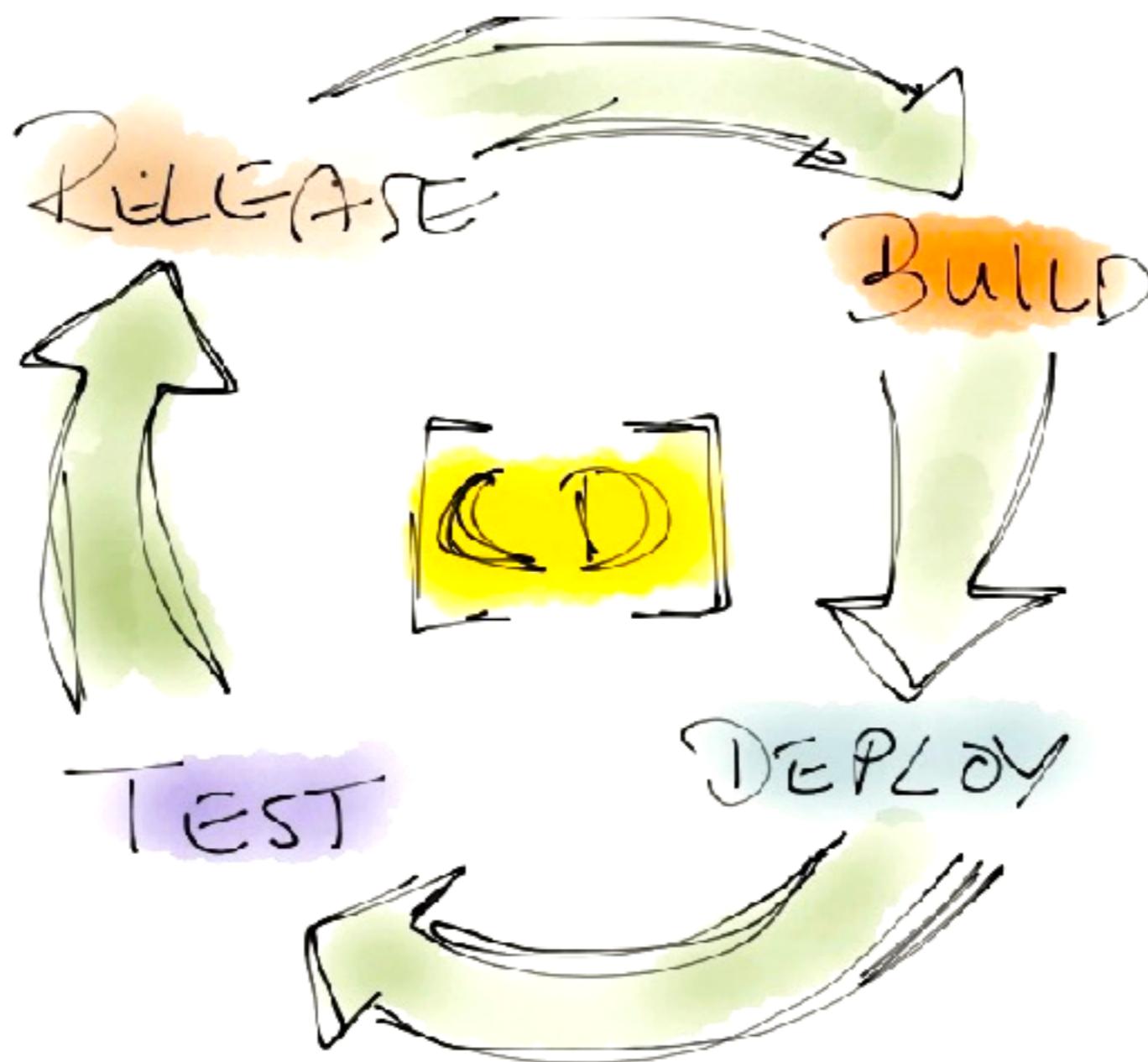
Strategy	ZERO DOWNTIME	REAL TRAFFIC TESTING	TARGETED USERS	CLOUD COST	ROLLBACK DURATION	NEGATIVE IMPACT ON USER	COMPLEXITY OF SETUP
RECREATE version A is terminated then version B is rolled out	✗	✗	✗	■■■	■■■	■■■	□□□
RAMPED version B is slowly rolled out and replacing version A	✓	✗	✗	■■■	■■■	■□□	■□□
BLUE/GREEN version B is released alongside version A, then the traffic is switched to version B	✓	✗	✗	■■■	□□□	■■□	■■□
CANARY version B is released to a subset of users, then proceed to a full rollout	✓	✓	✗	■■■	■□□	■□□	■■□
A/B TESTING version B is released to a subset of users under specific condition	✓	✓	✓	■■■	■□□	■□□	■■■
SHADOW version B receives real world traffic alongside version A and doesn't impact the response	✓	✓	✗	■■■	□□□	□□□	■■■



Deploy vs Release



Deploy vs Release



Mean Time to Recover (MTTR)



Mean Time to Recover (MTTR)

Tests are very important to reduce amount of defects in your systems. However, it's important to acknowledge that bugs will always happen in production.



Mean Time to Recover (MTTR)

How **fast** to recover from them will help determining our success !

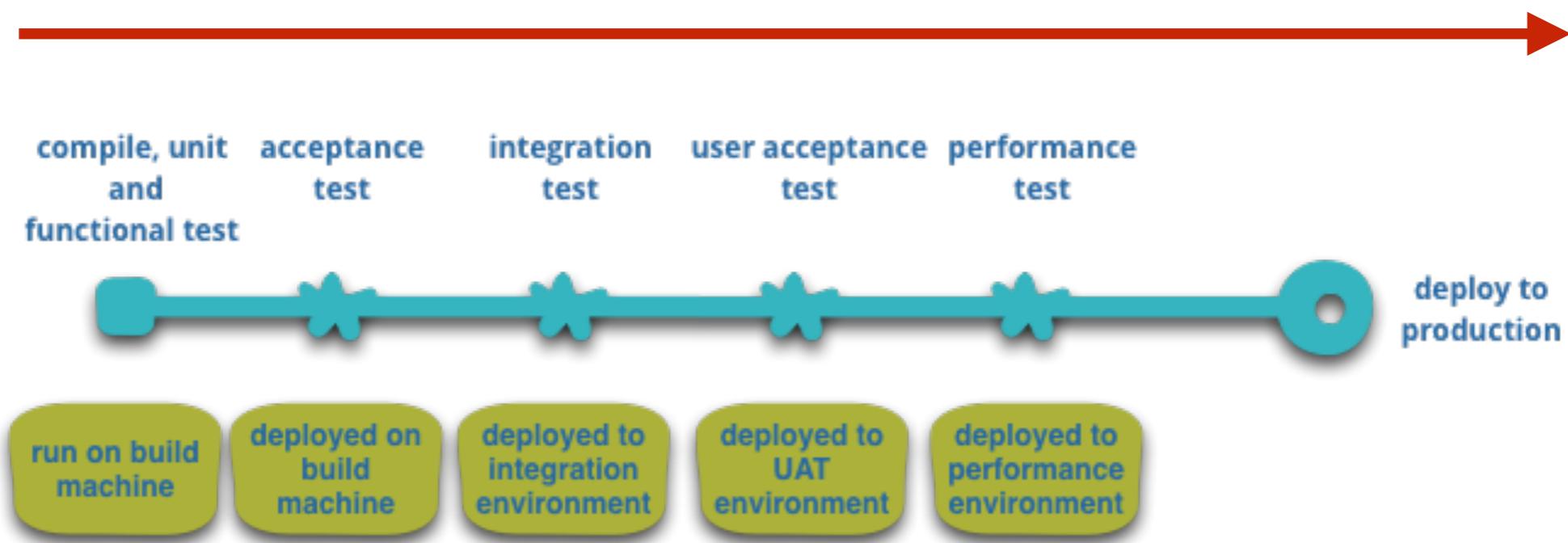


Current situation ?

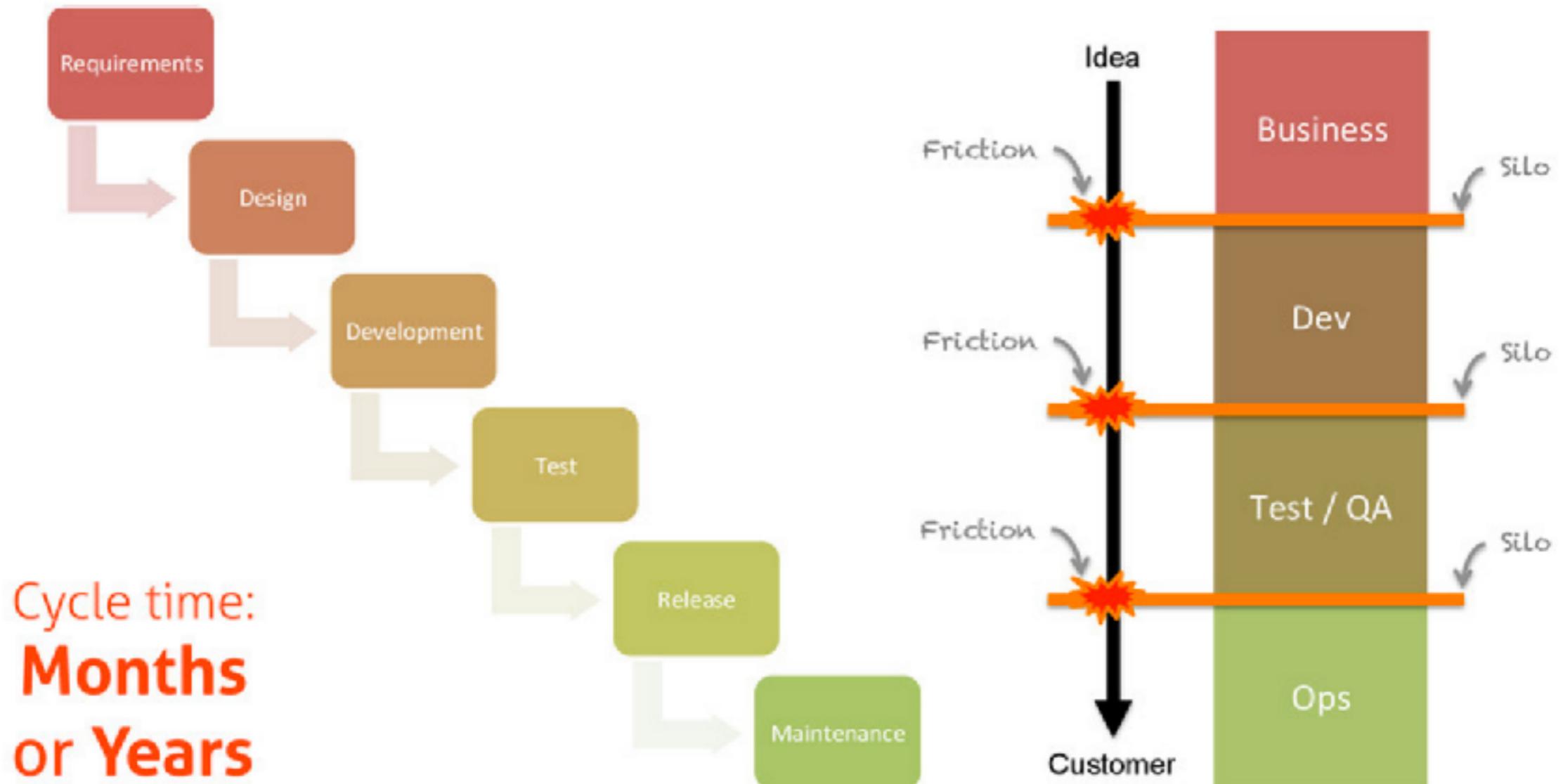


Infrastructure Automation

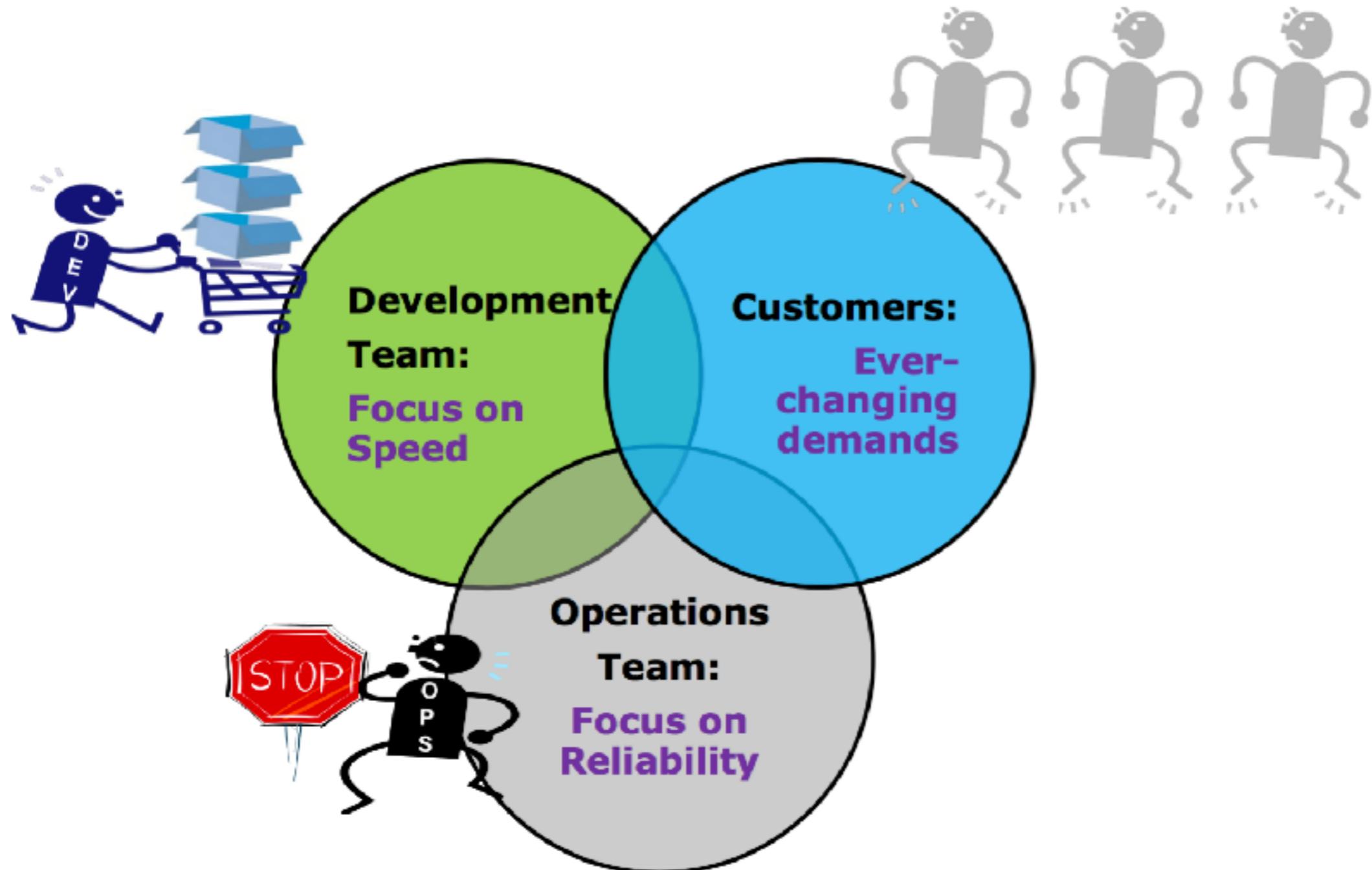
Lead time ?



Traditional development



Conflict of Interest

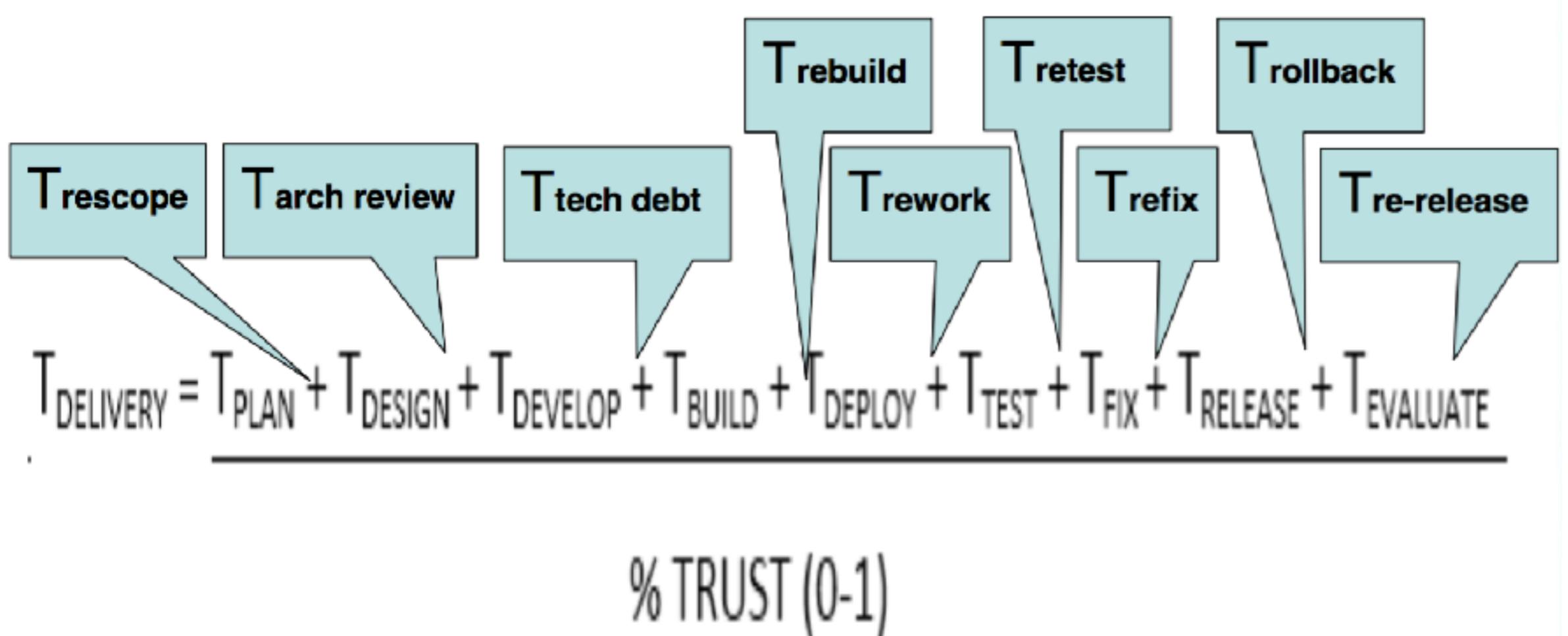


Conflict of Interest

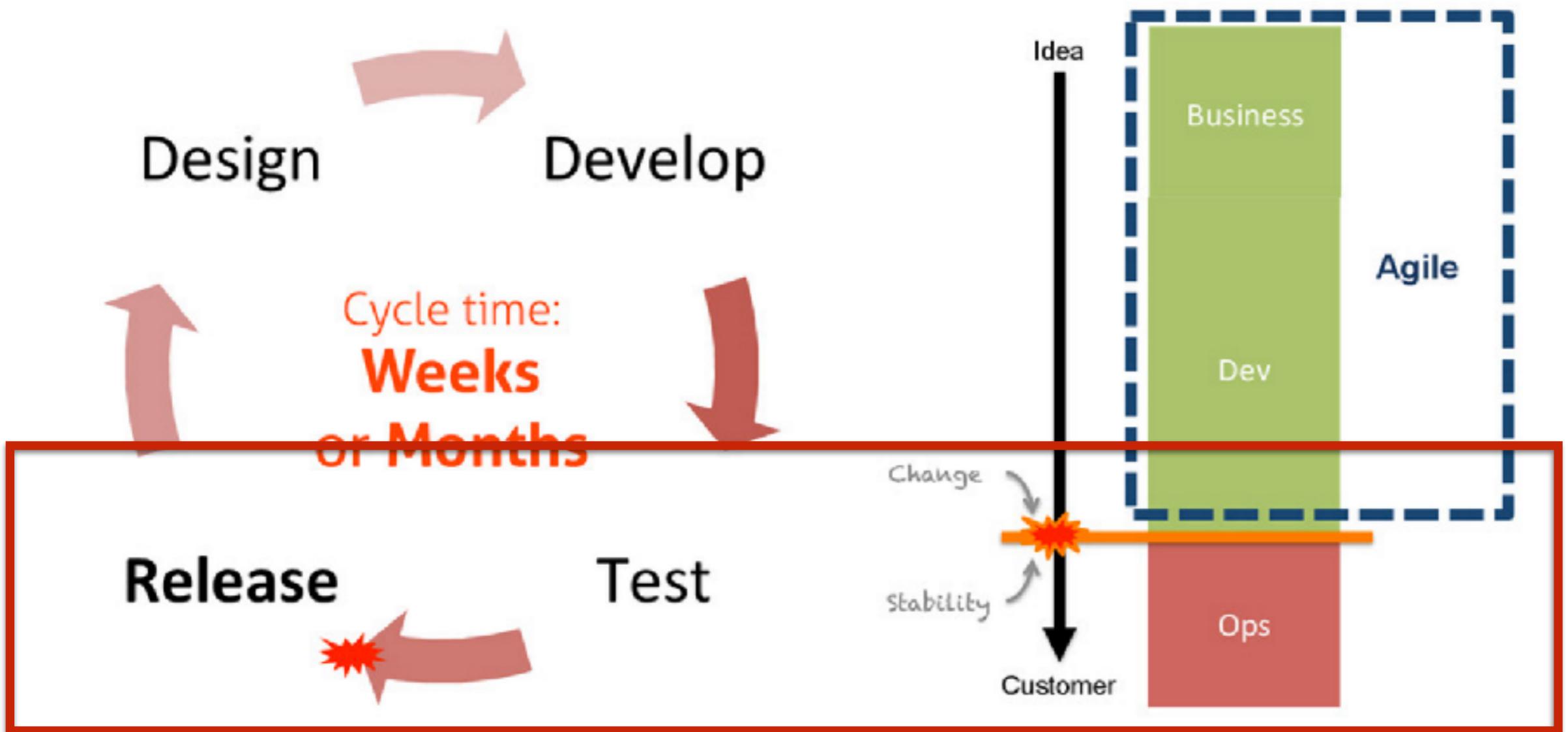




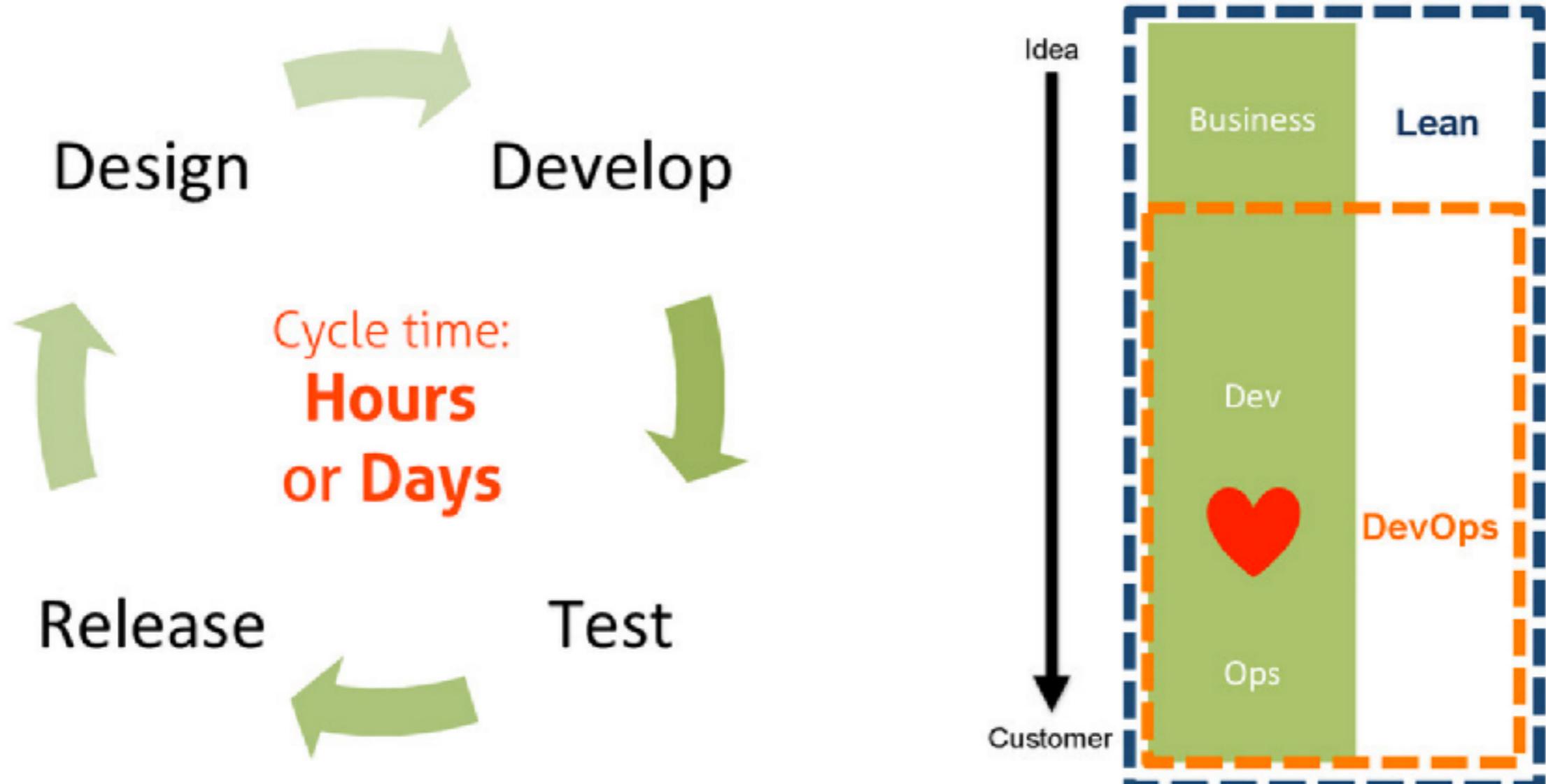
Low trust create extra steps

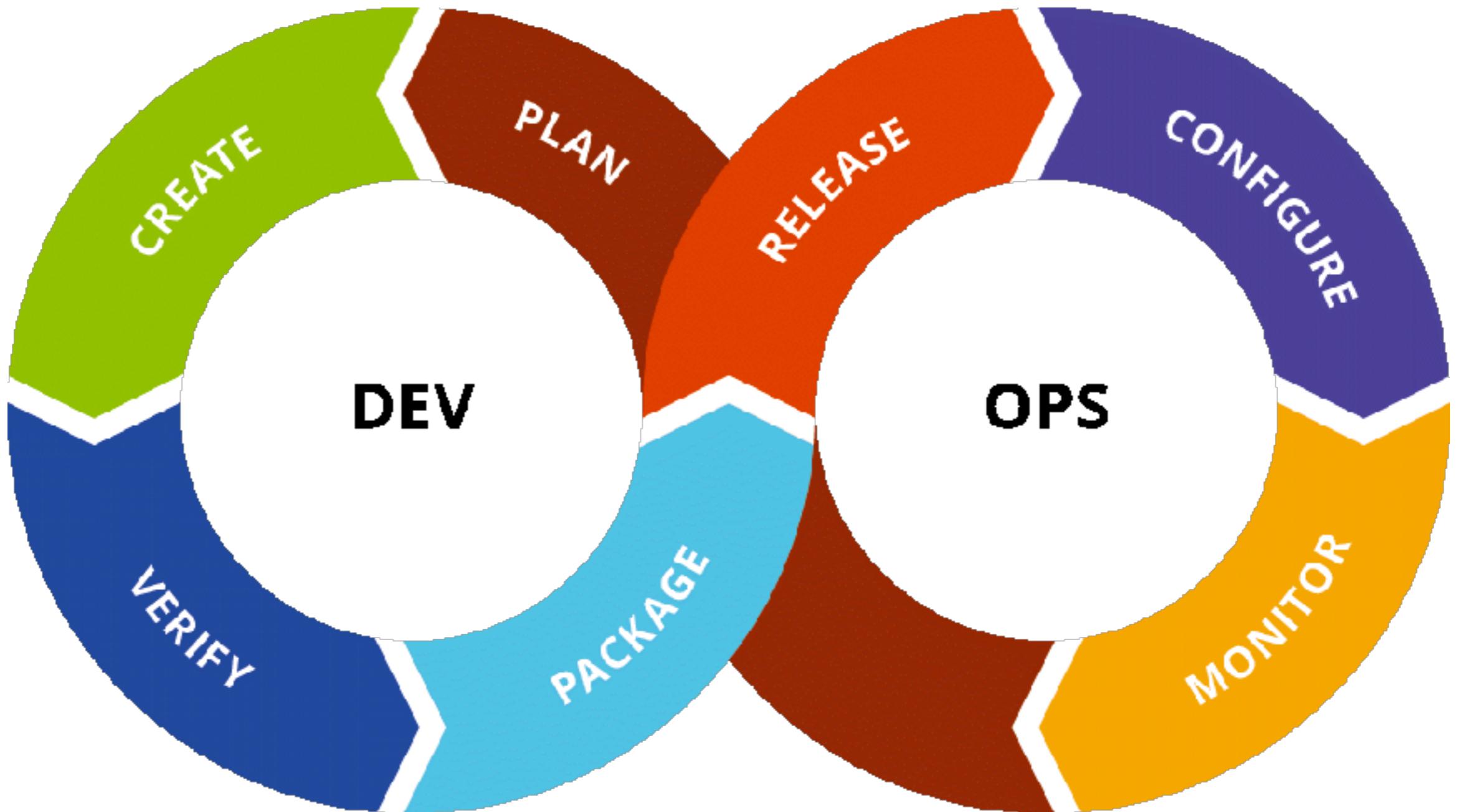


Iterative/Agile development



Rise of DevOps





DevOps ?

"DevOps is
development
and operations
collaboration"

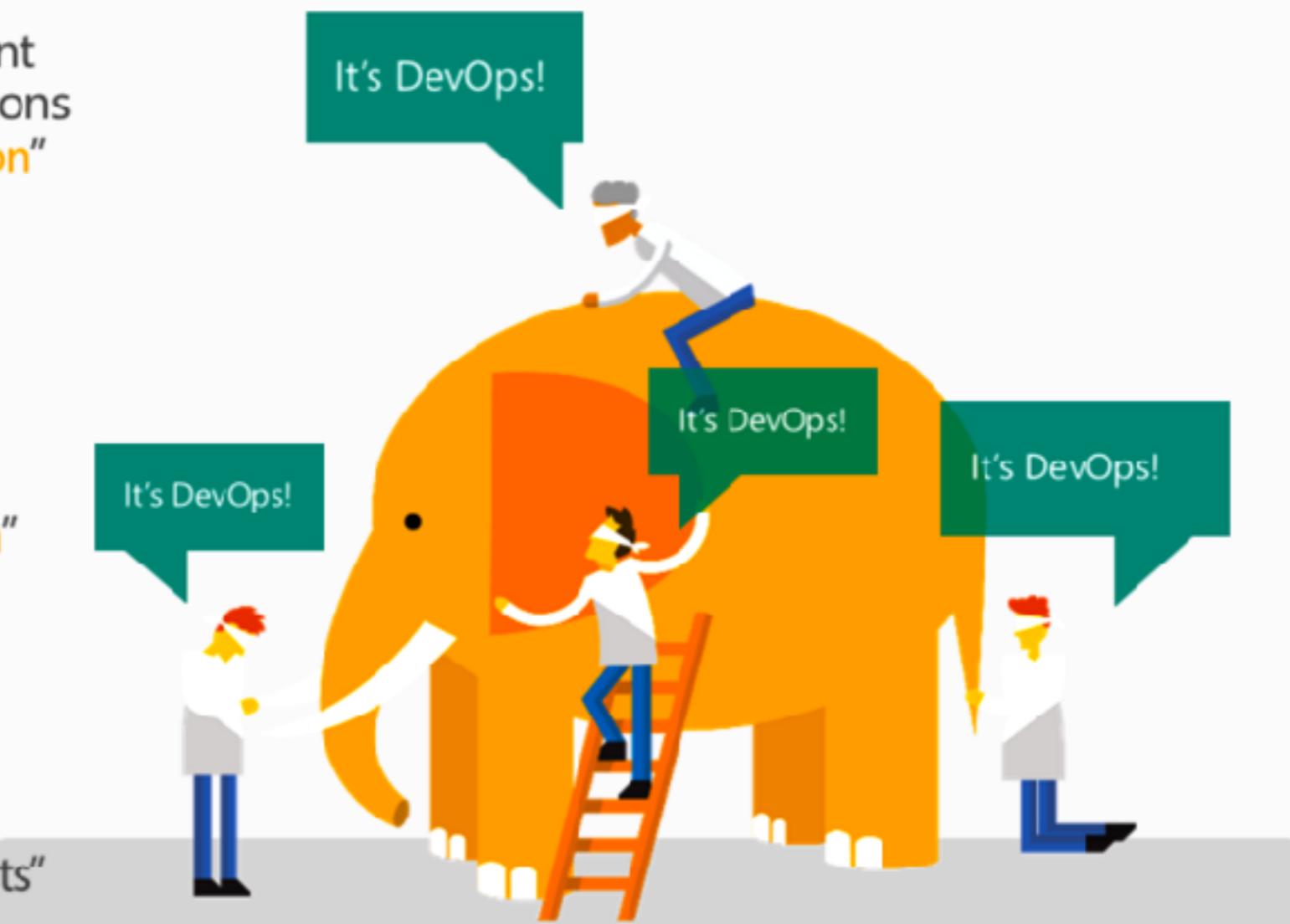
"DevOps
is using
automation"

"DevOps
is **small**
deployments"

"DevOps is
treating your
infrastructure
as code"

"DevOps
is feature
switches"

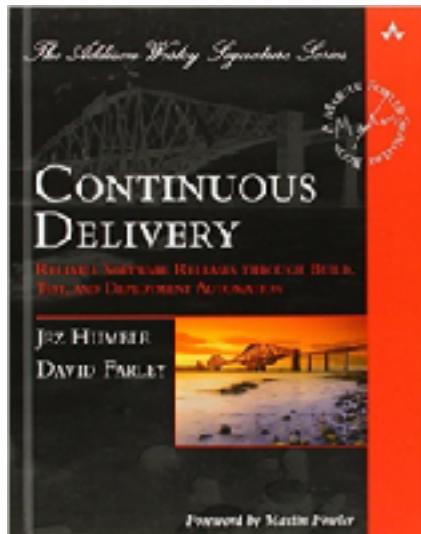
"Kanban
for Ops?"



DevOps ?

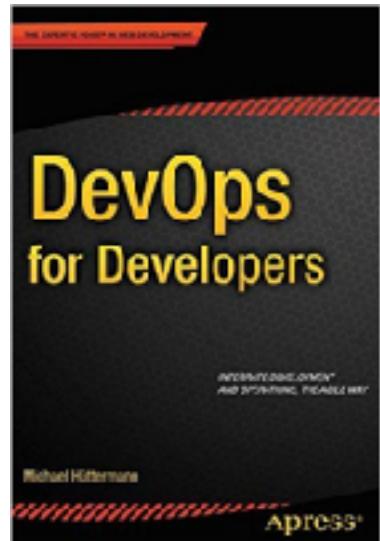
“A movement of people who care about developing and operating reliable, secure, high performance systems at scale.”

- Jez Humble -



DevOps ?

“A mix of patterns intended to **improve collaboration** between development and operations. DevOps addresses **shared goals and incentives** as well as **shared processes and tools.**”

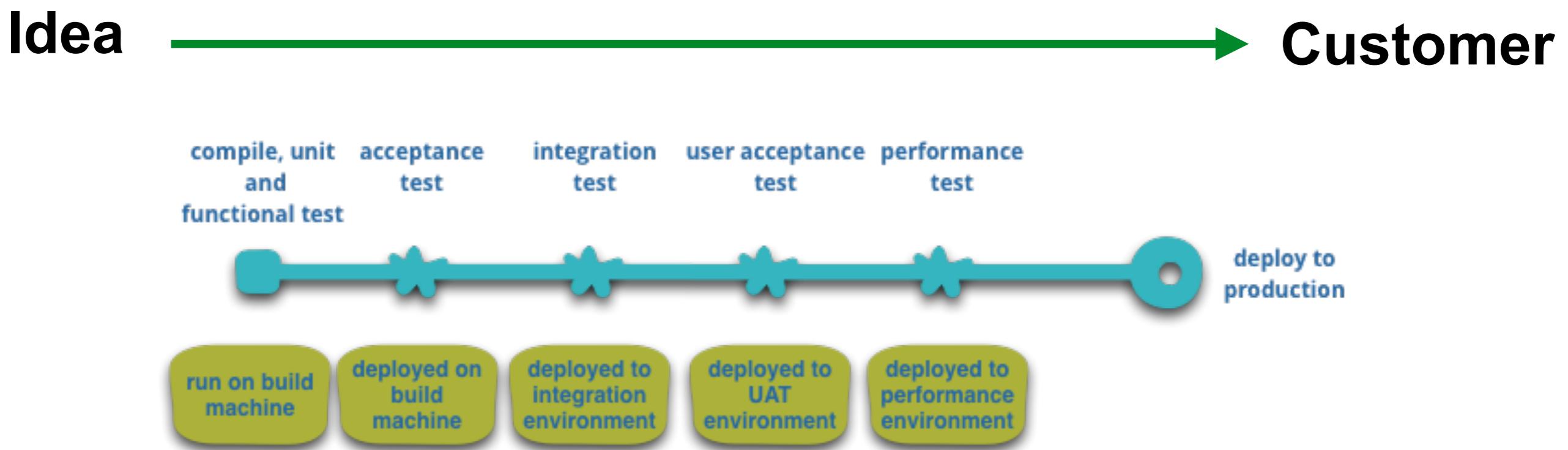


- Michael Huttermann -

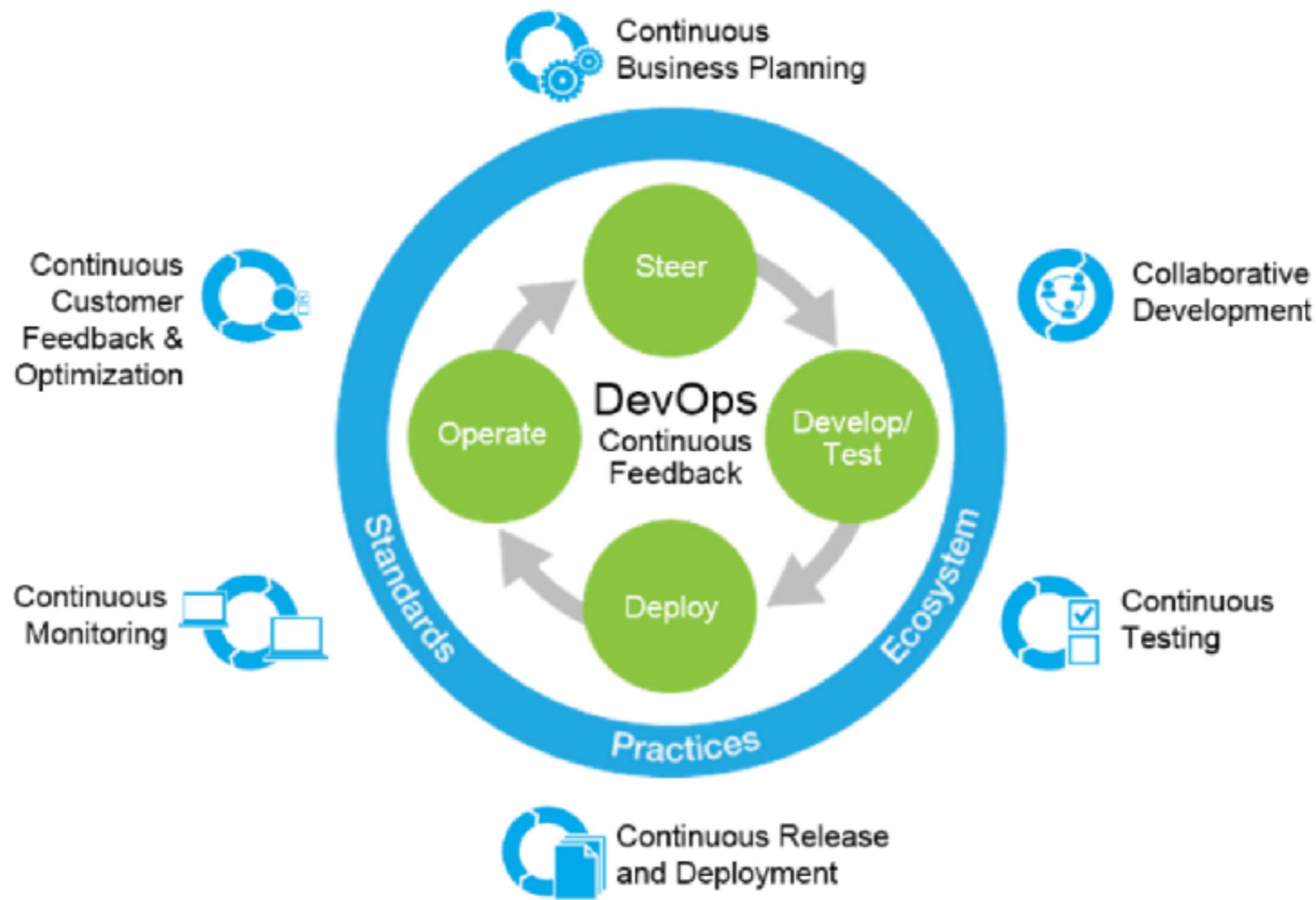


Goal of DevOps

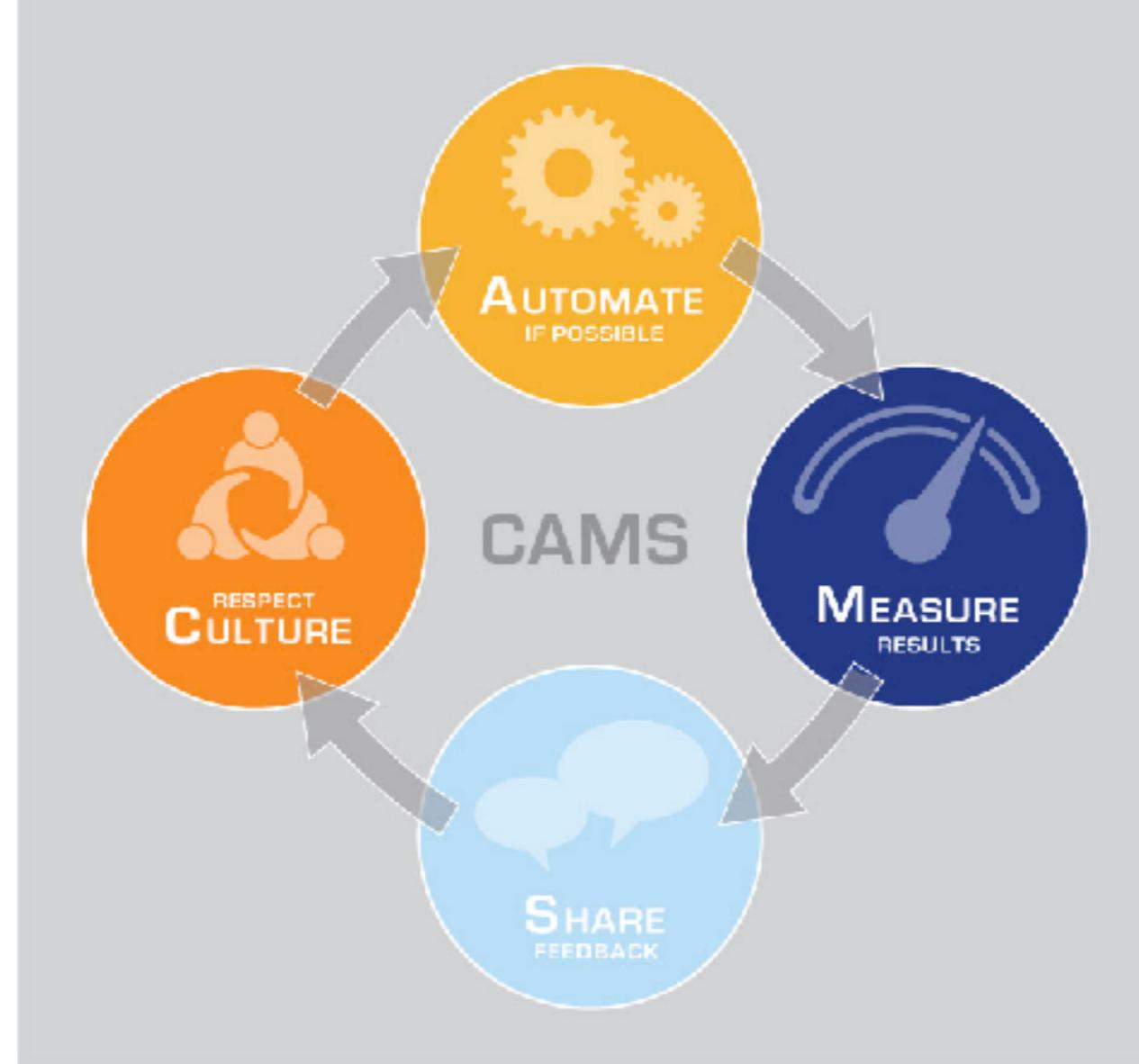
“Improve the delivery of value for Customer and Business”



DevOps Life Cycle



DevOps Principles



DevOps Principles

Culture => People, Process, Tools

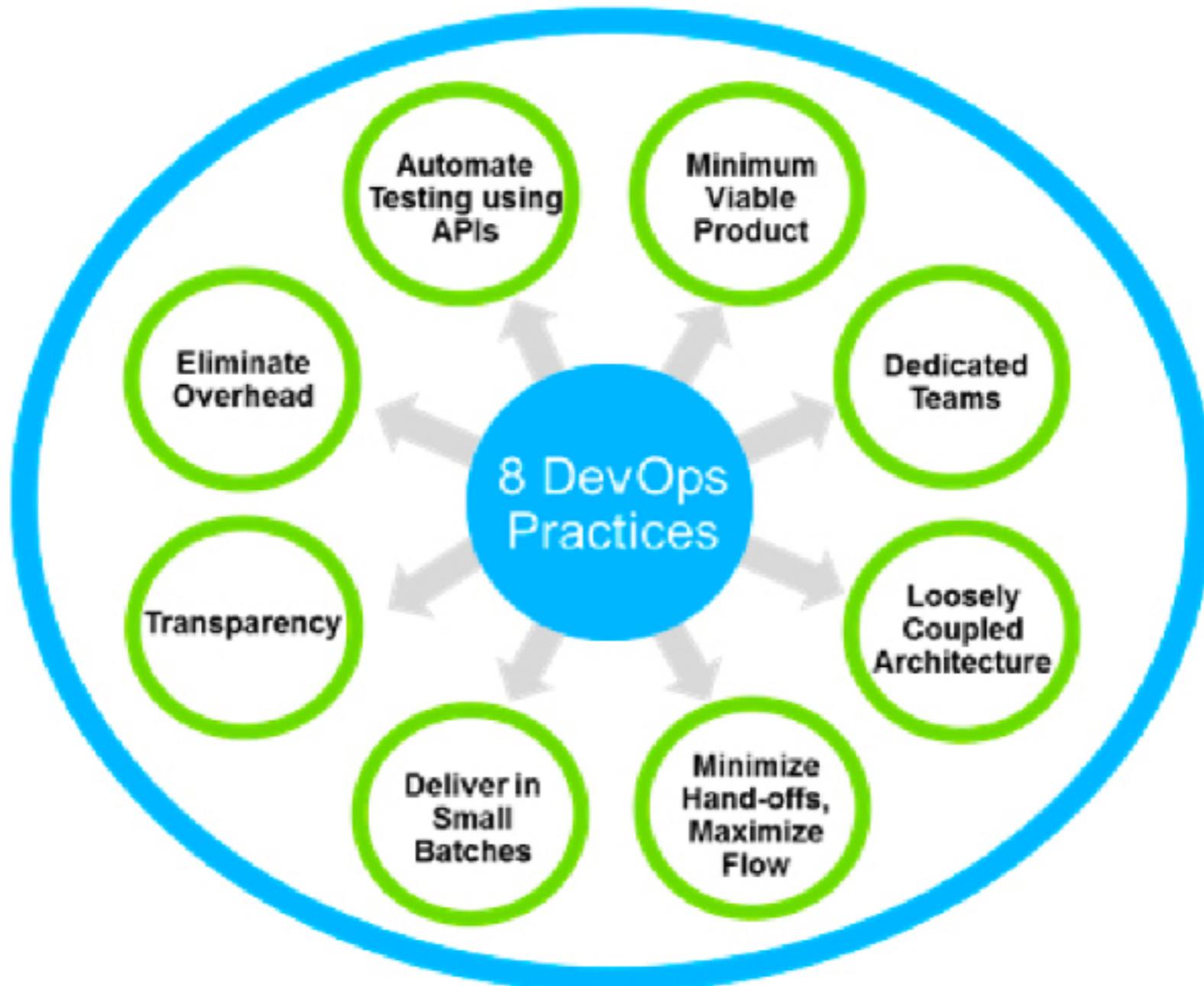
Automation => Infrastructure as Code

Measurement => Measure everything

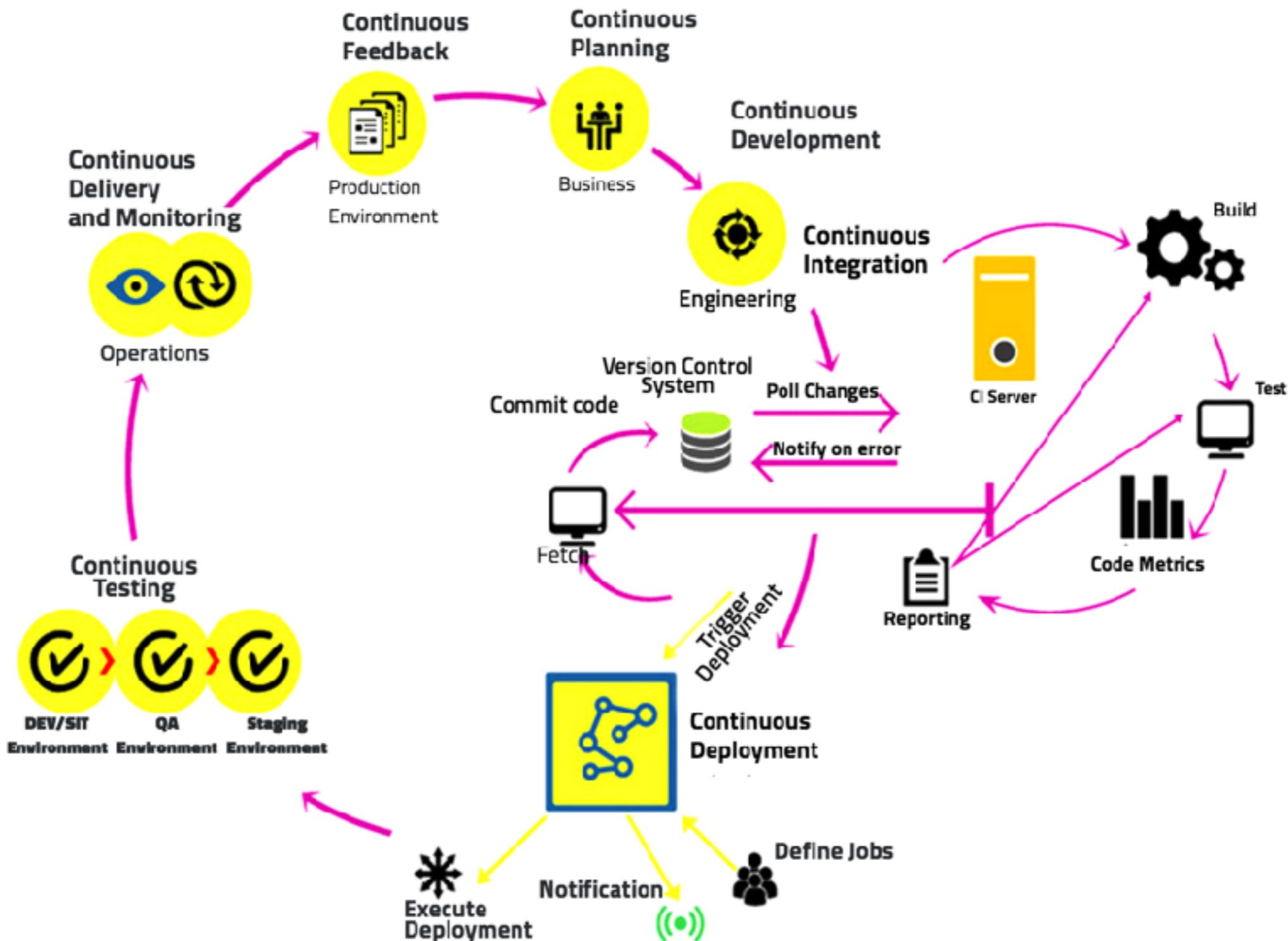
Sharing => Collaboration/Feedback



DevOps Practices



DevOps 7C Happiness



DevOps

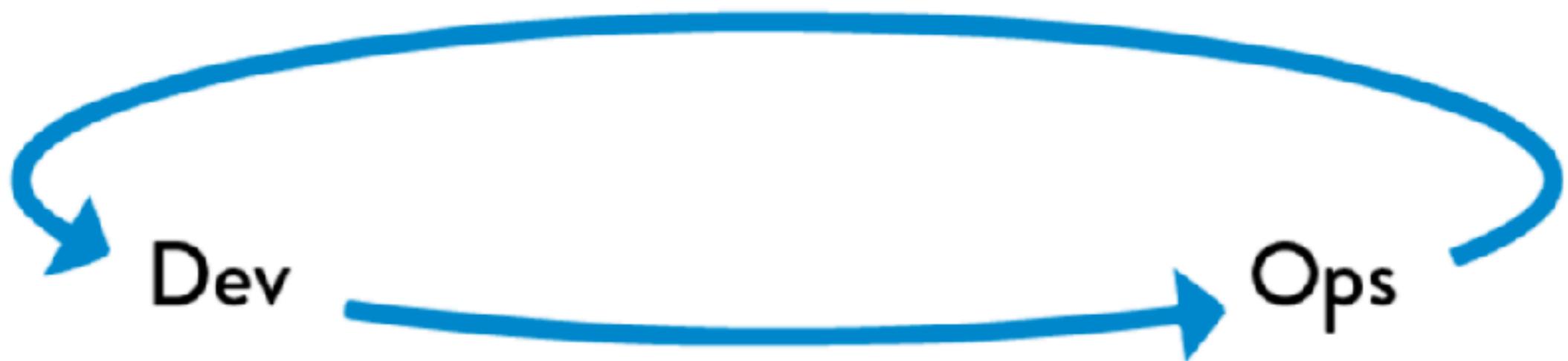
3 ways principle



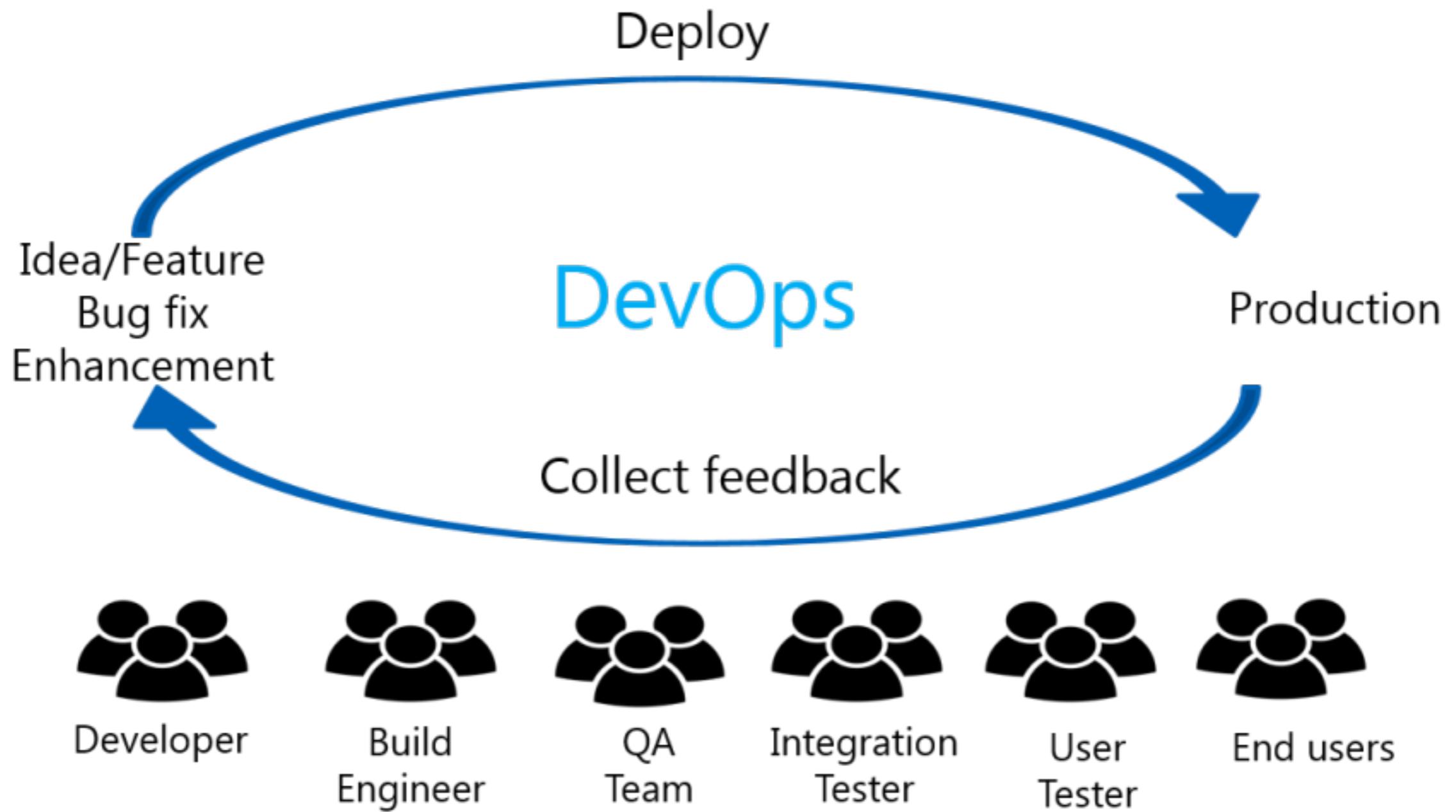
Flow principle



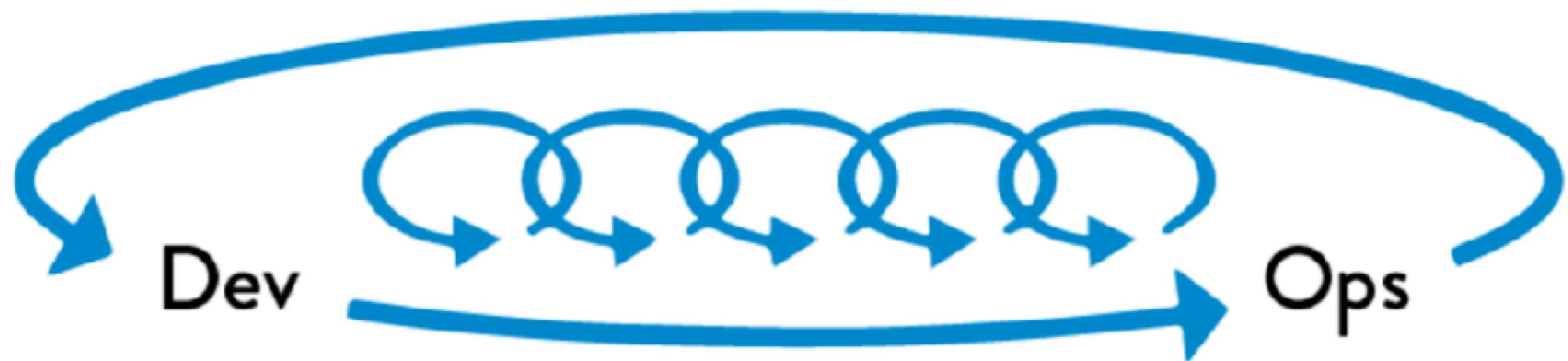
Feedback principle



Feedback principle



Continuous learning principle



All about Organization structure and culture



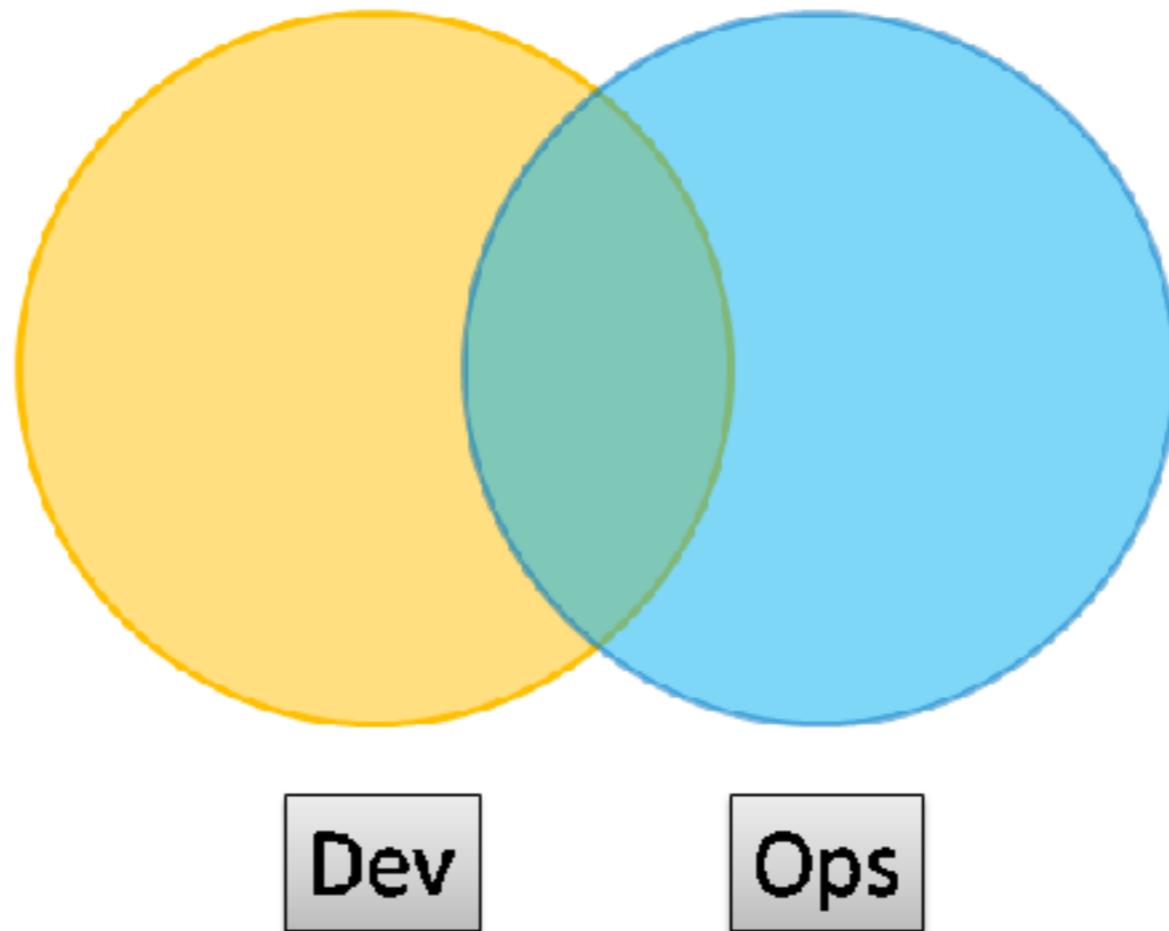
People -> Process -> Tool



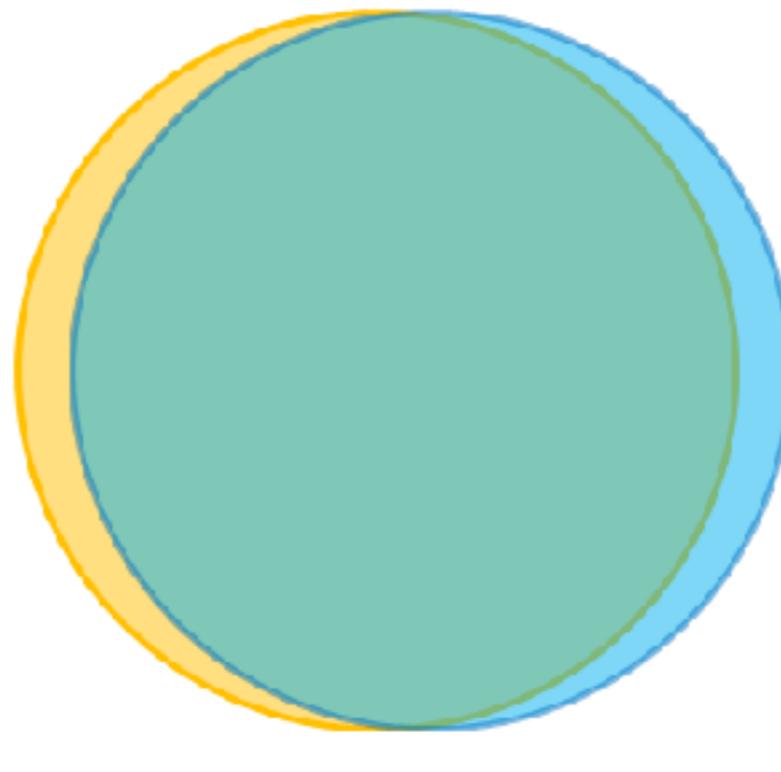
DevOps Topologies



Type 1 – Smooth Collaboration



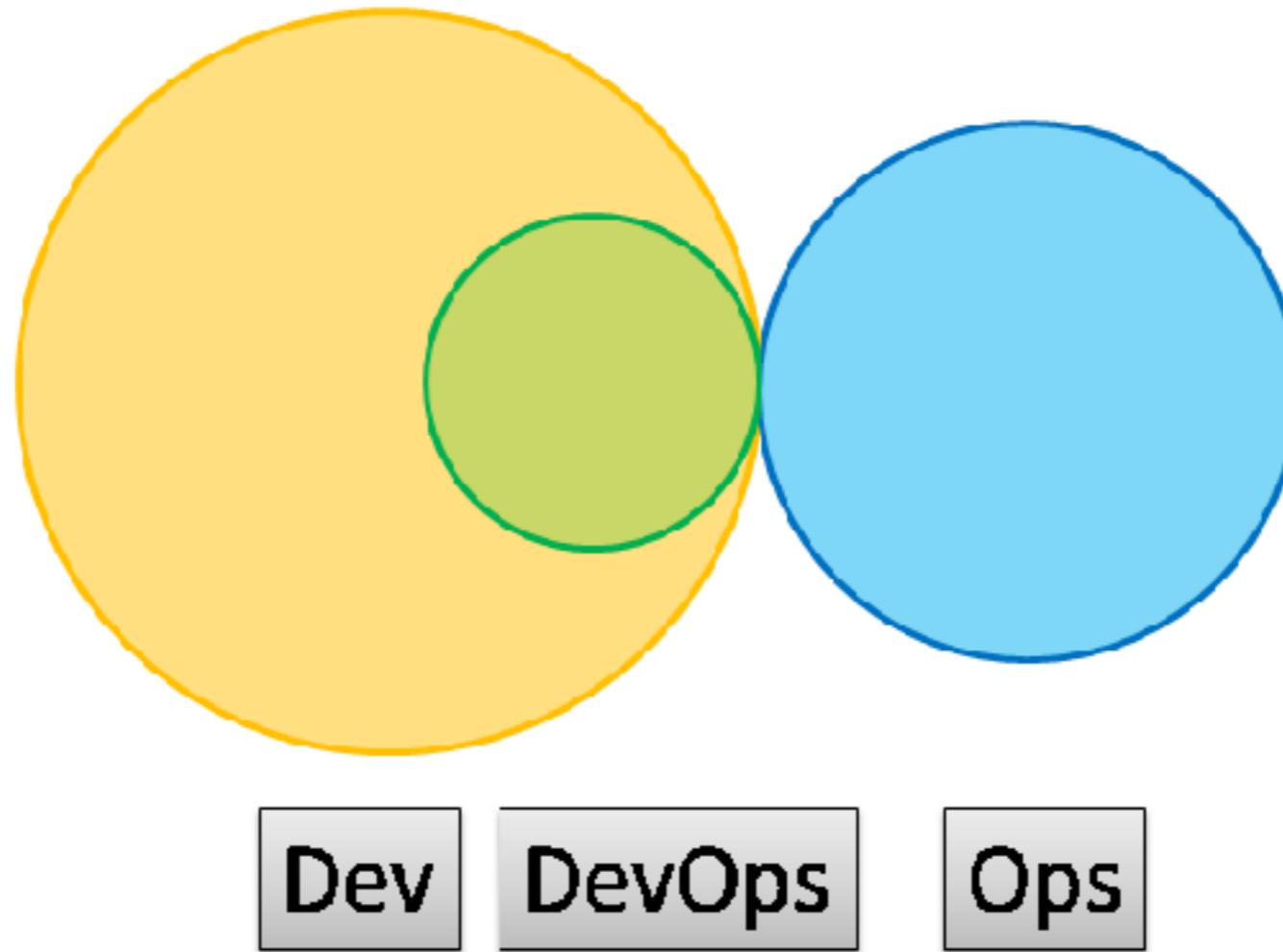
Type 2 – Fully Embedded



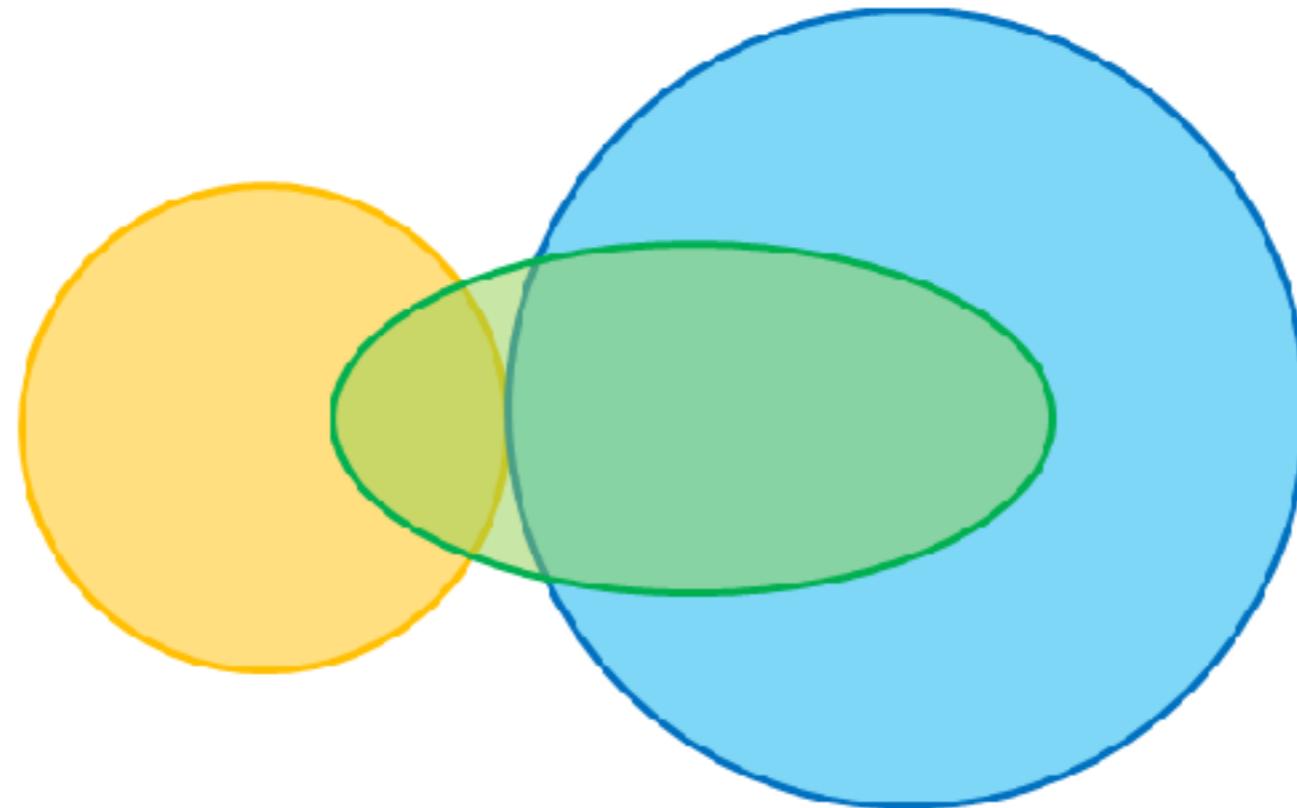
Dev | Ops



Type 3 – Infrastructure-as-a-Service



Type 4 – DevOps-as-a-Service



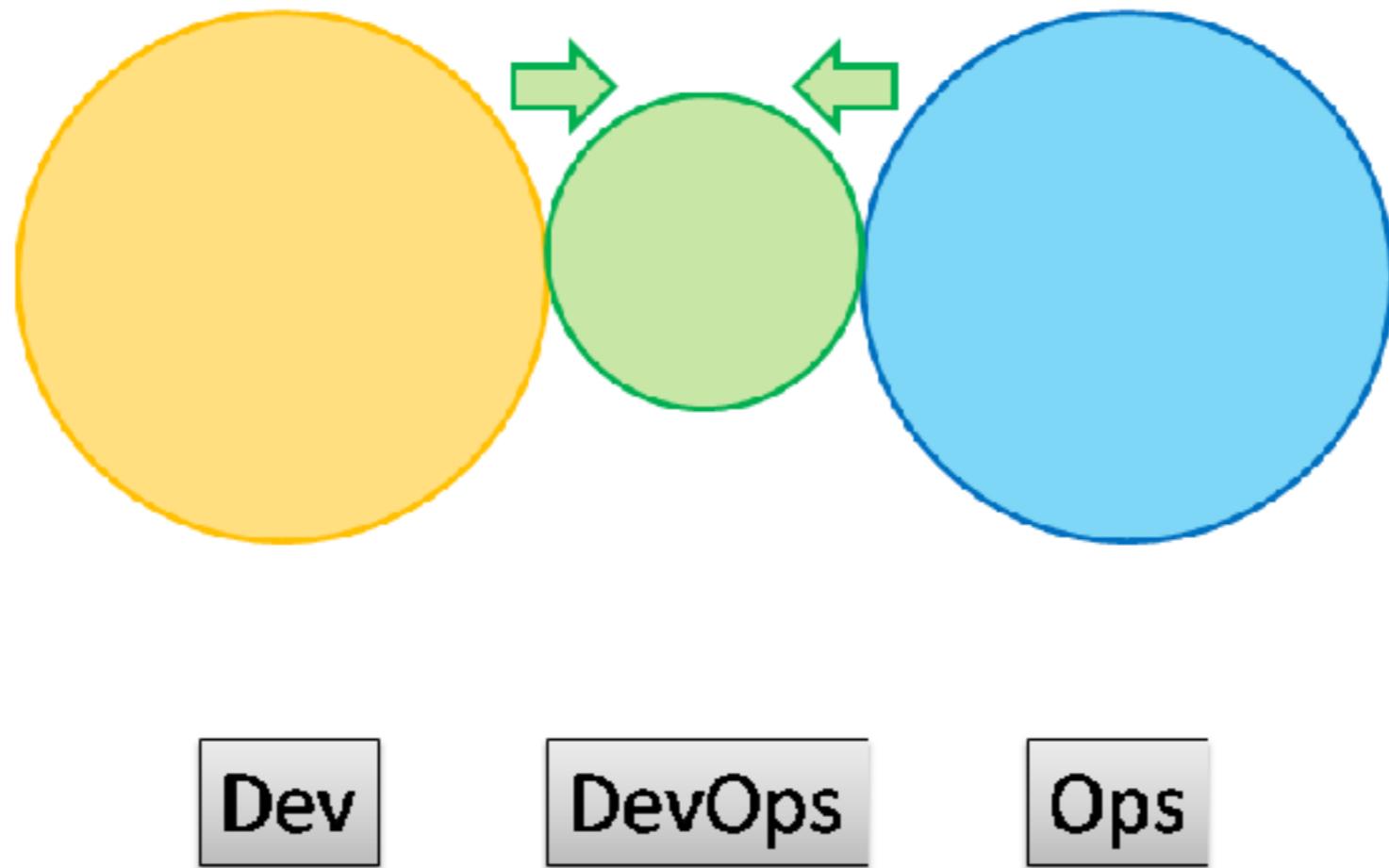
Dev

DevOps

Ops



Type 5 – Temporary DevOps Team



DevOps Tools

Periodic Table of DevOps Tools (v2)																					
		Open Source			SCM			Database Mgmt			Build			Embed	Download	Add					
1	1m	Gh	Github	Os	Frc	Fr	Freemium	Pd	Cloud / IaaS / PaaS	Repo Mgmt	Testing	An	Analise	Ch	Pu	An	Sl	Dk	2	1m	
3	Os	Gt	Git	Dm	CB-maestro	4	En	Fm	Enterprise	Deployment	Config / Provisioning	Containerization	11	En	Pu	Analise	Sl	Dk	Az	Aws	
12	Fm	Bb	Bitbucket	Lb	Liquibase	13	Os	En	Enterprise	Cloud / IaaS / PaaS	Release Mgmt	Collaboration	14	En	An	Salt	Docker	Azure	Google Cloud Platform		
19	Os	Gl	GitLab	Rg	Hedgegate	Mv	Maven	Gr	Gradle	At	ANT	Fn	FitNesse	Se	Selenium	Ga	Dh	Jn	Ch	Pu	
37	Os	Sv	Subversion	Dt	svnkit	Gt	Grunt	Gp	Gulp	Br	Bamboo	Cu	Cucumber	Cj	Cucumber.js	Qu	Npm	Cs	Gd	Sf	
55	Os	Hg	Mercurial	Dp	Dephtix	Sb	rbt	Mk	Makko	Ck	JUnit	Jt	JUnit	Jm	JMeter	Tn	Artifactory	Tc	Sf	Cn	
73	En	Cw	ISPW	Id	Idea	Msb	MSBuild	Rk	Reke	Pk	Pecker	Mc	Mocha	Xltv	XLT TestView	Jm	Jasmine	Nx	Ba	Travis CI	
91	Fn	Xlr	XL Release	Ur	UrbanCode Release	Bm	RMC Release Process	Hp	HP Cedar	Au	Automatic	Pi	Plutora Release	Sr	Serena Release	Tfs	Team Foundation	Tr	Jr	Rf	
106	Os	Ki	Kibana	Nr	New Relic	Ni	Nagios	Zb	Zabbix	Dd	DataDog	Ei	Elasticsearch	St	StackStorm	Sp	Splunk	Le	Logentries	Sl	HiChat
107	Fm			108	Os	109	Os	110	En	111	Os	112	En	113	En	114	Fm	115	Fm	116	Os
117	Os	Gr	Graylog	Sn	Smart	Tr	Tripwire	Ff	Fortify					118	Os	119	Os	120	En	Pv	Rental Tracker
121	Fm																				Sn
122	En																				ServiceNow

XebiaLabs
Deliver Faster

Follow @xebialabs

92	Fn	Ur	UrbanCode Release	93	Fn	Bm	RMC Release Process	94	Fn	Hp	HP Cedar	95	Fn	Au	Automatic	96	Fn	Pi	Plutora Release	97	Fn	Sr	Serena Release	98	Pd	Tfs	Team Foundation	99	Fm	Tr	Logstash	100	Pd	Jr	Jira	101	Fm	Rf	HiChat	102	Fm	Sl	Slack	103	Fm	Fd	Flowdock	104	Pd	Pv	Rental Tracker	105	Fn	Sn	ServiceNow
106	Os	Ki	Kibana	107	Fm	Nr	New Relic	108	Os	Ni	Nagios	109	Os	Zb	Zabbix	110	En	Ei	Elasticsearch	111	Os	St	StackStorm	112	En	Sp	Splunk	113	En	Le	Logentries	114	Fm	Sl	Logstash	115	Fm	ls	Sumsologic	116	Os	Gr	Graylog	117	Os	Sn	Smart	118	Os	Tr	Tripwire	119	Os	Ff	Fortify

<https://xebialabs.com/periodic-table-of-devops-tools/>



Microservices

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

No DevOps Team

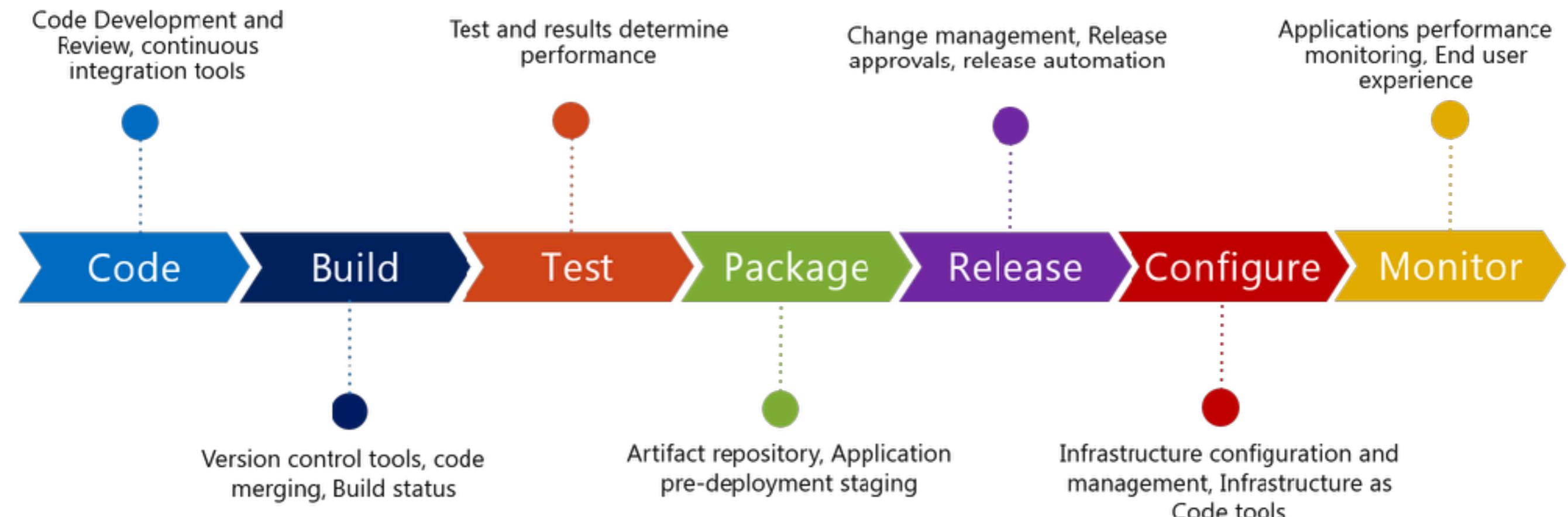
Problem department !!



DevOps != Tools
Tools enable DevOps



DevOps Process & Tools



What can i measure ?

Mean Time to Recover/Repair (MTTR)

Mean Time to Detection (MTTD)

Change Lead Time

Change Failure Rate

Deployment or Change Frequency

Deployment Time

Percentage of successful deployments



What can i measure ?

Application Usage and Traffic

Application Performance

Automated Test Pass (%)

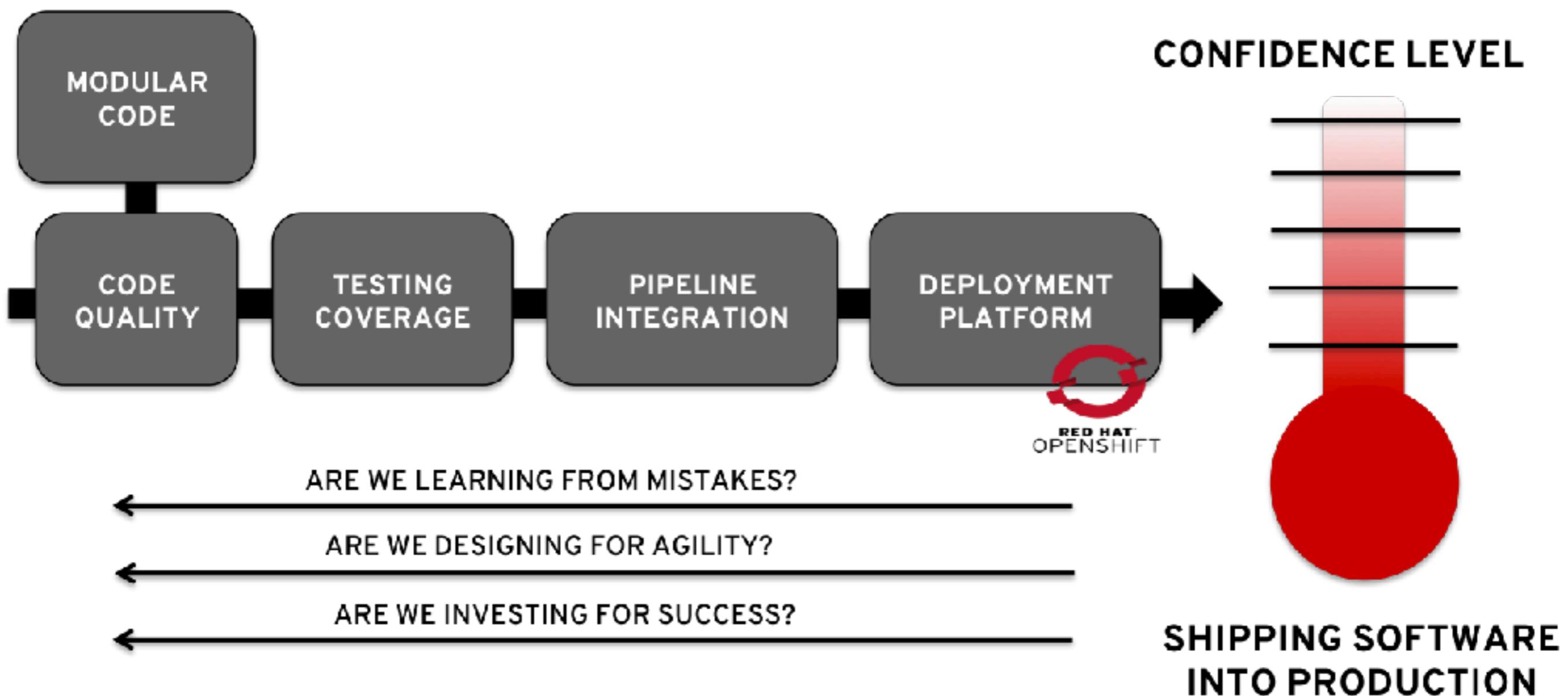
Defect Rate

Failed Deployments

Availability



What can i measure ?

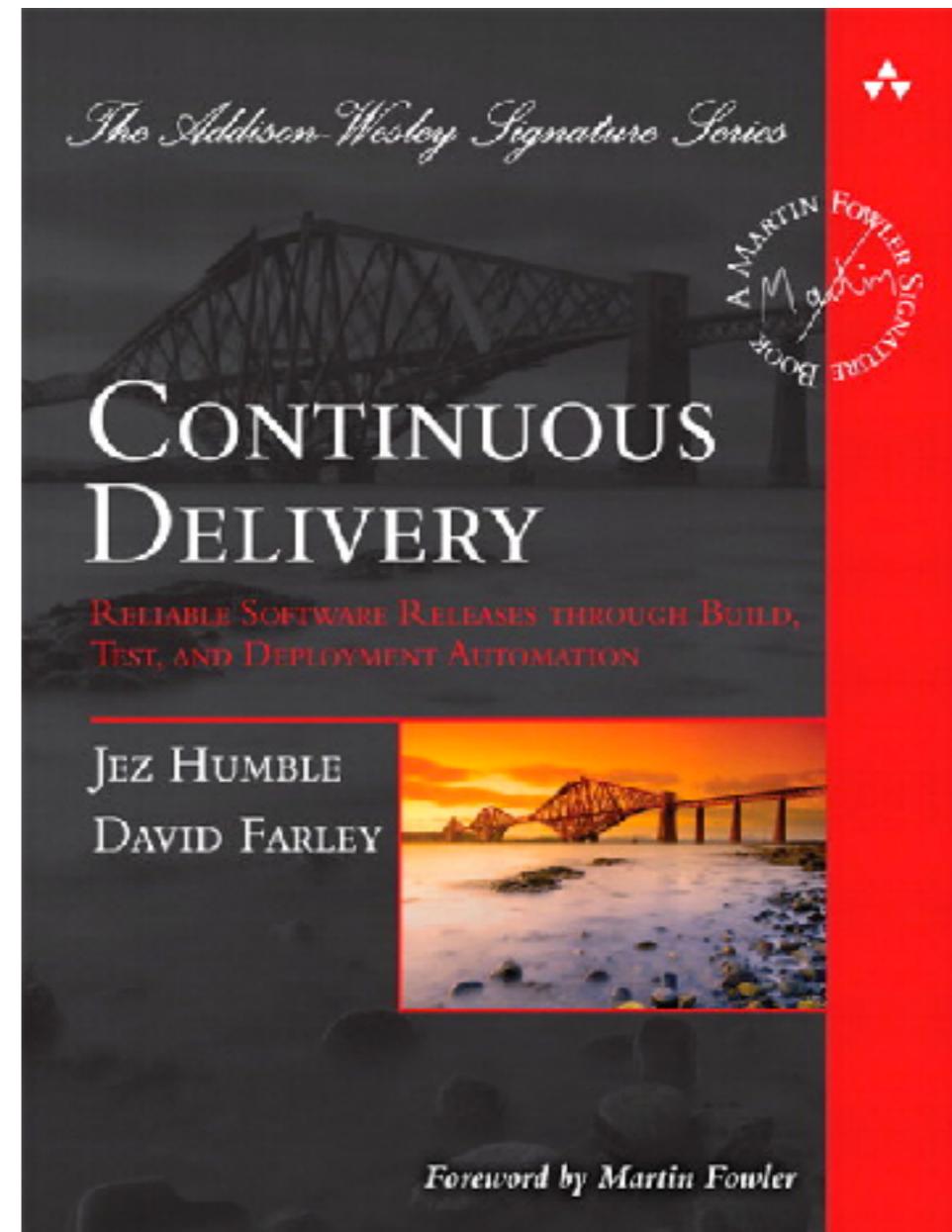
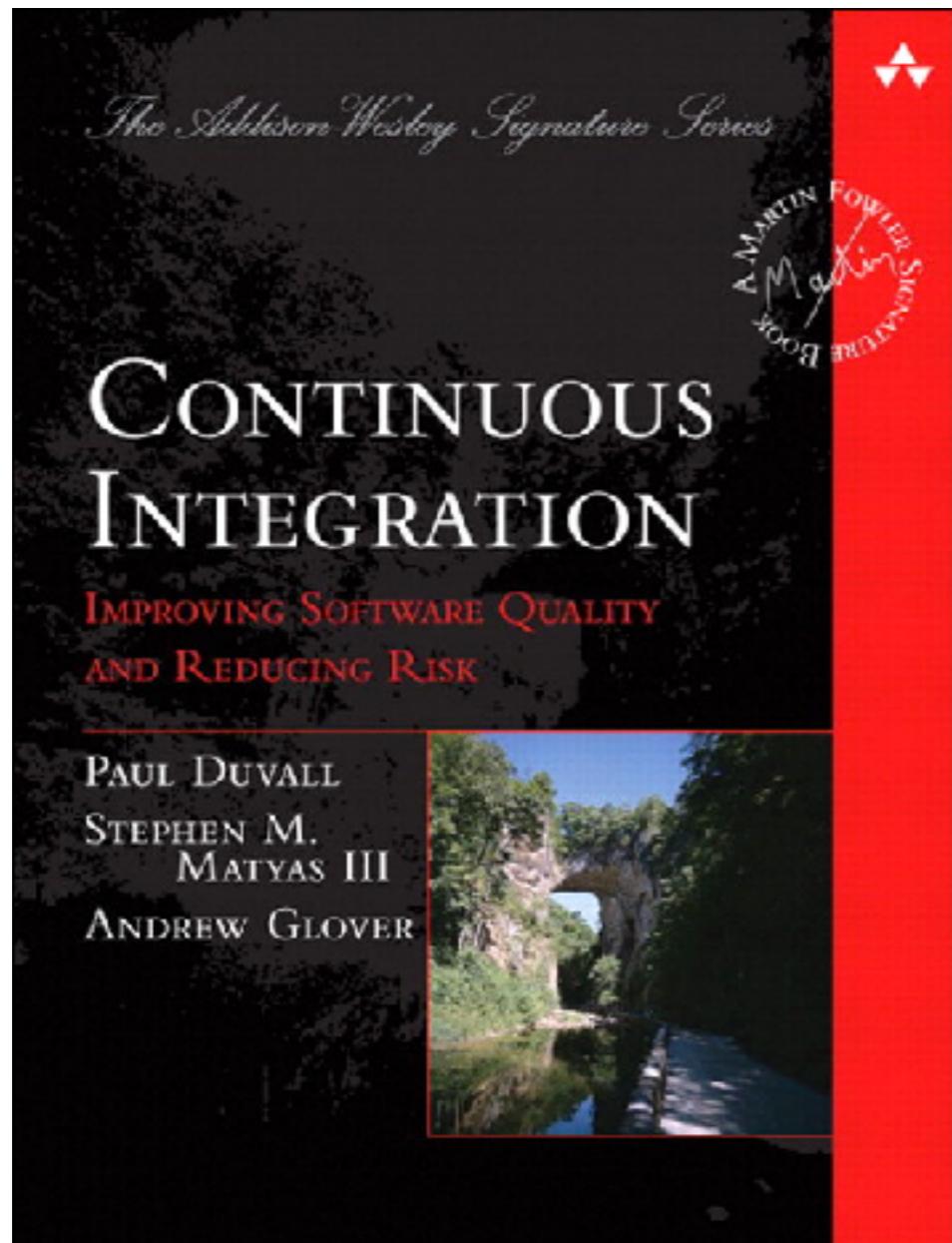




Start with Continuous Integration Continuous Delivery



Improve quality and reduce risk

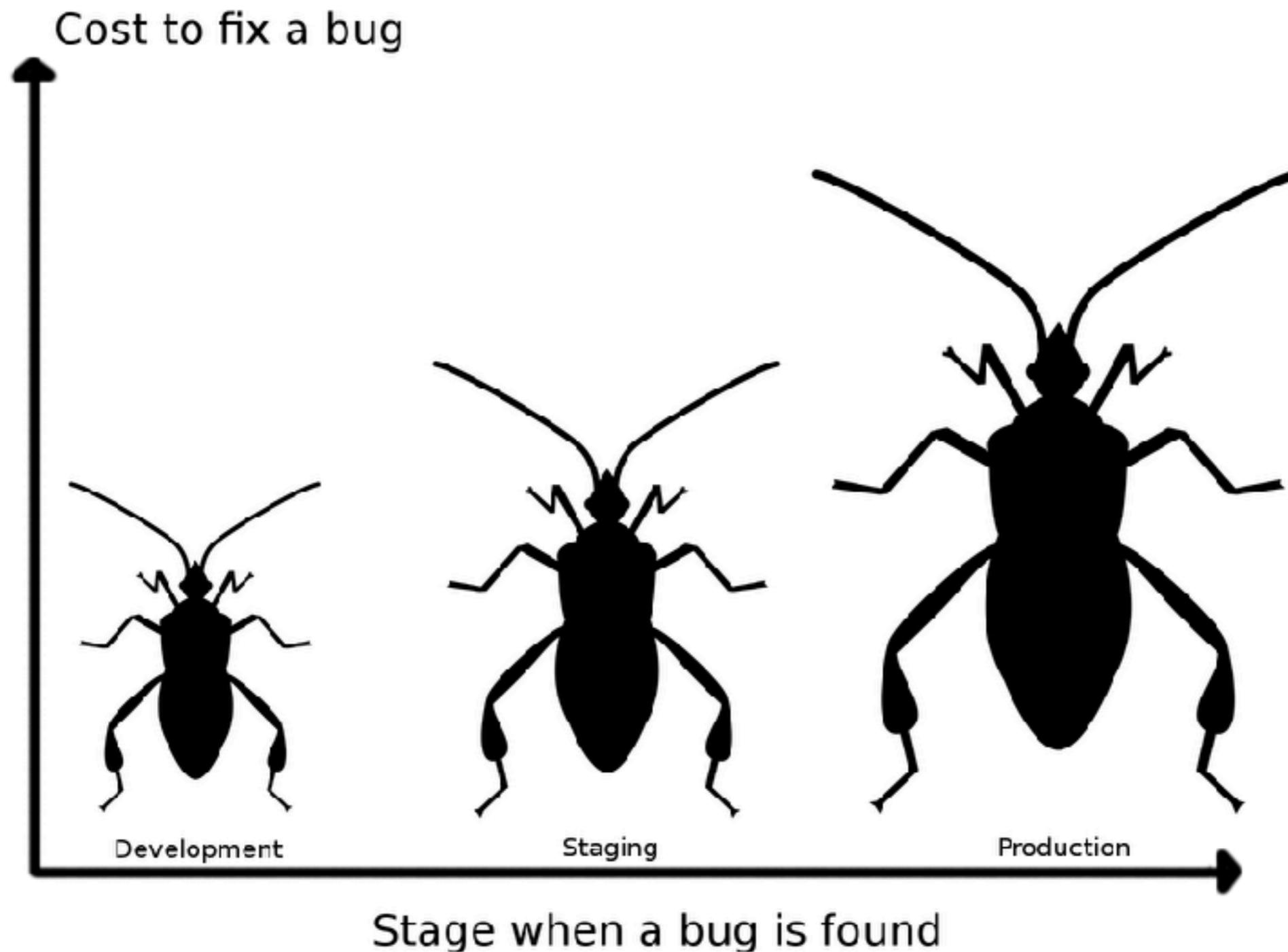


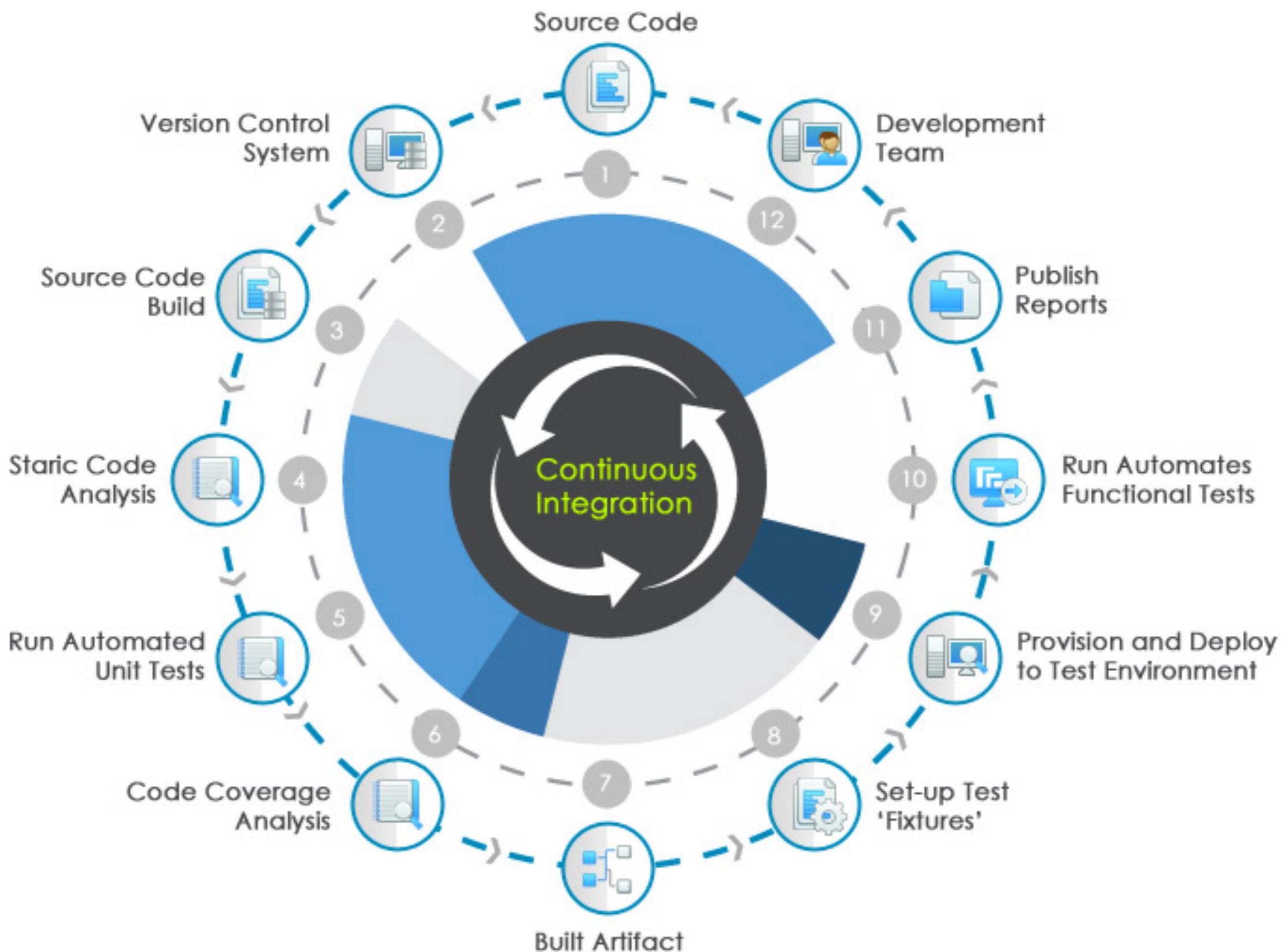
The cost of integration

1. Merging the code
2. Duplicate changes
3. Test again again !!
4. Fixing bugs
5. Impact on stability



The cost of integration







Jenkins

Bamboo



TeamCity

> goTM



Hudson





Jenkins

Bamboo

CI is about what people do
not about what tools they use



Hudson



travis

wercker

circleci



Continuous Integration

Discipline to integrate frequently



Continuous Integration

Strive to make **small change**

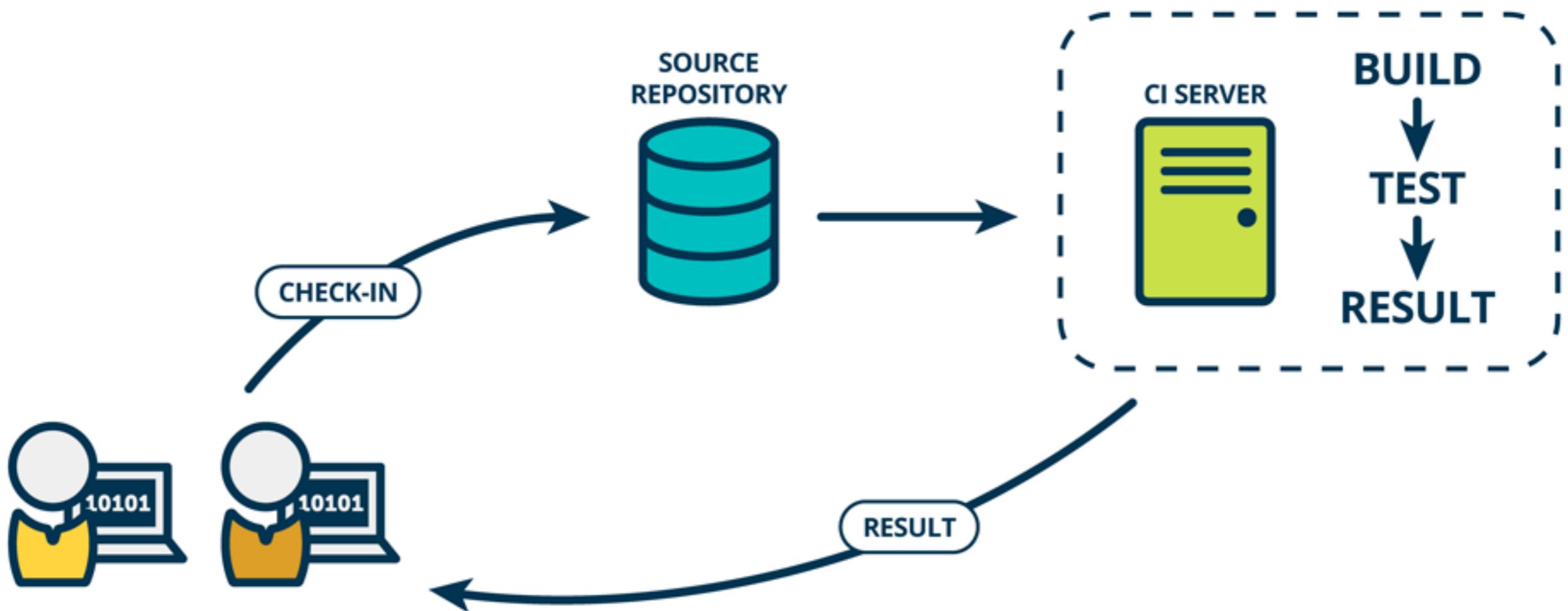


Continuous Integration

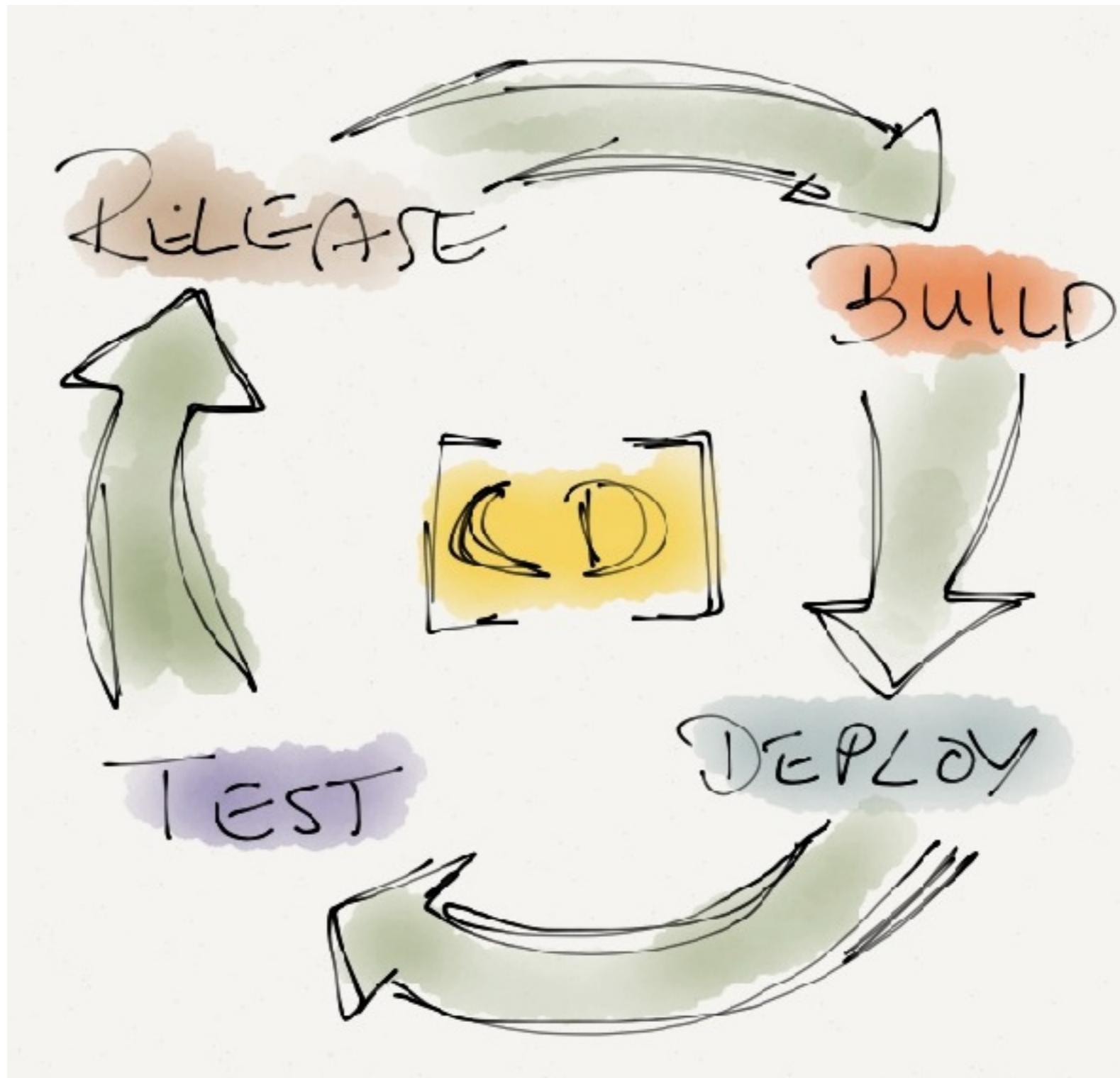
Strive for **fast feedback**



Continuous Integration



CD ?



CD ?

CONTINUOUS DELIVERY



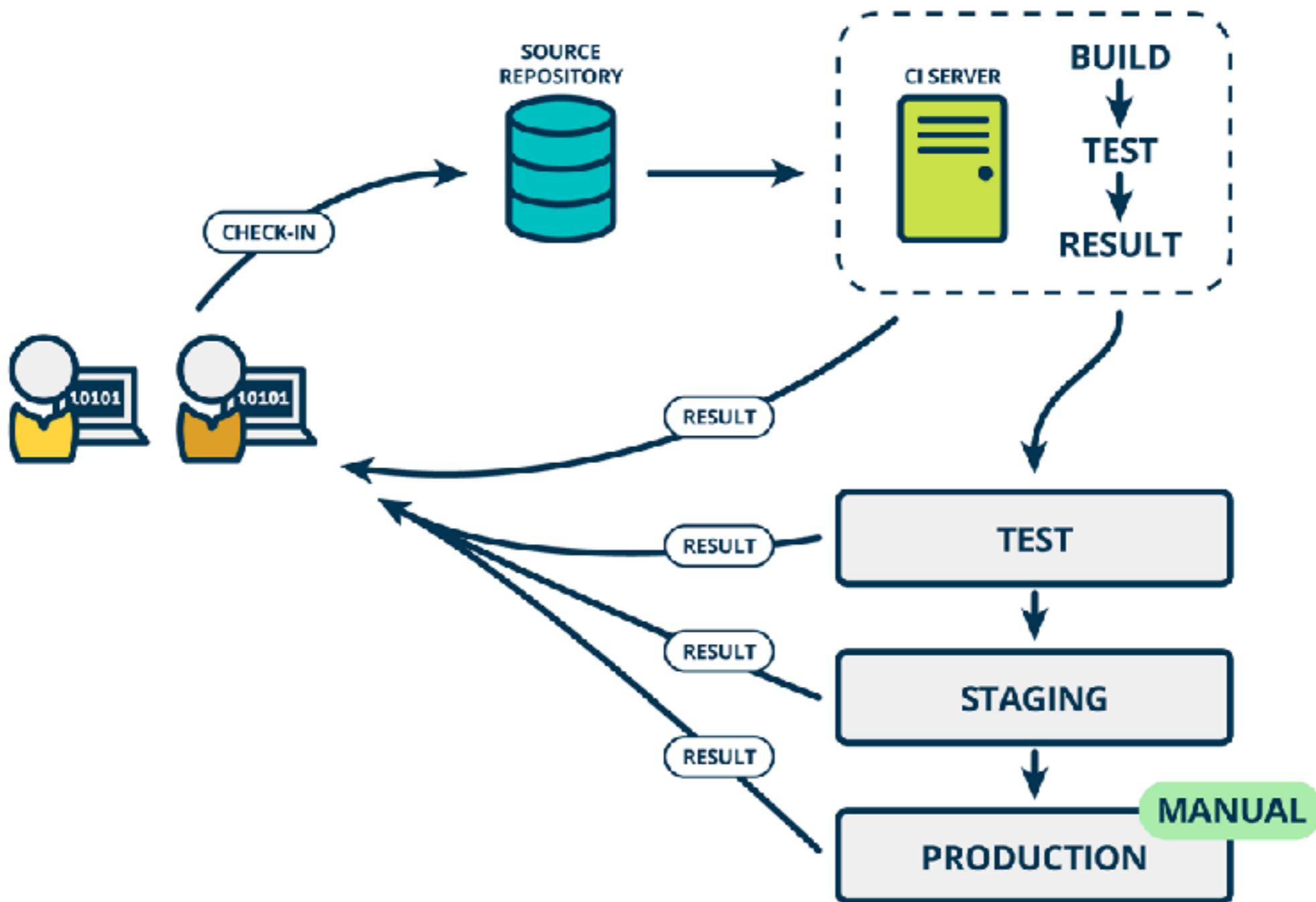
CONTINUOUS DEPLOYMENT



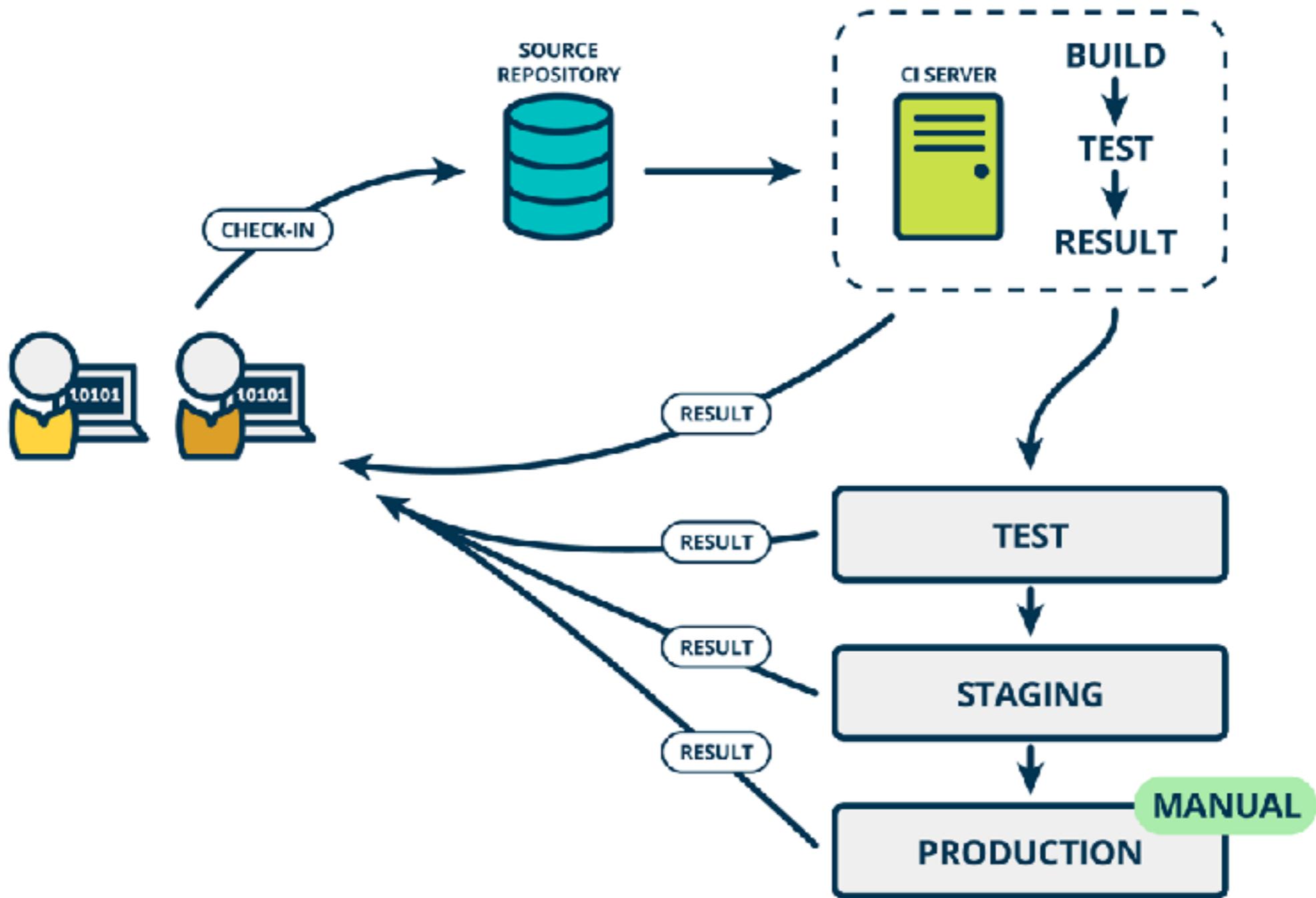
<http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment>



Continuous Delivery



Rise of DevOps



Continuous Integration

is a Software development practices



Practice 1

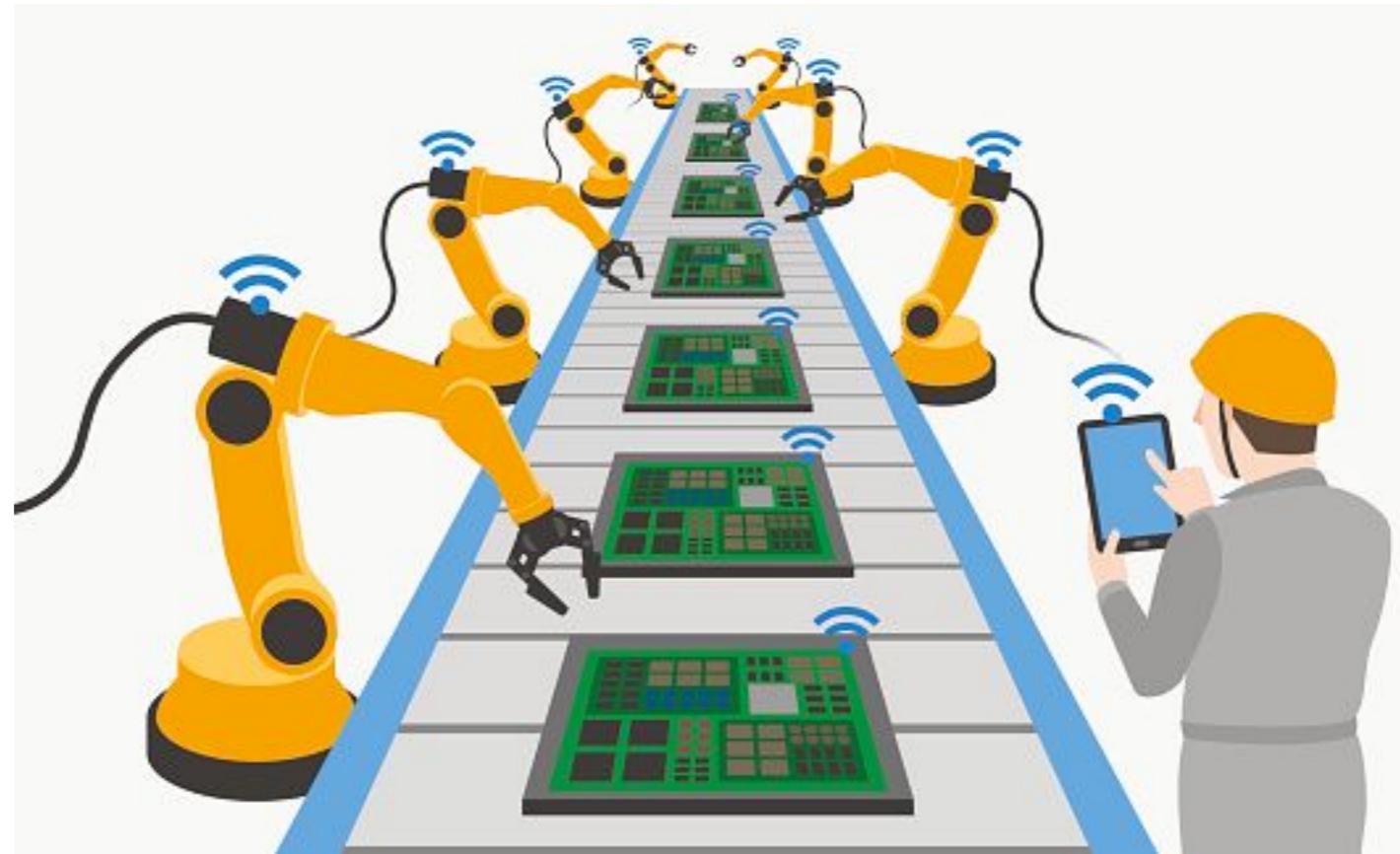
Maintain a single source repository

In general, you should store in source control
everything you need to build anything



Practice 2

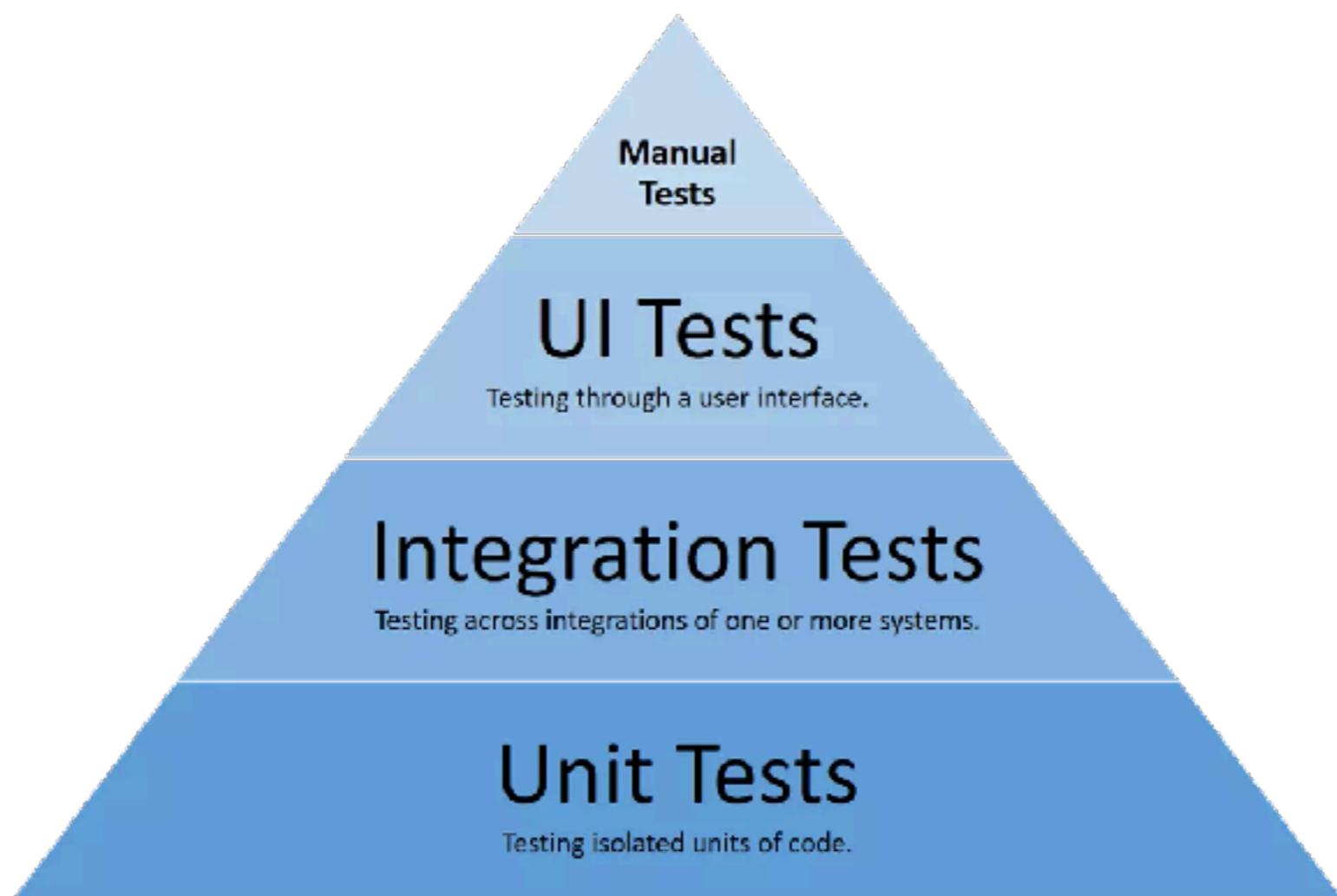
Automated the build
Automated environment for builds



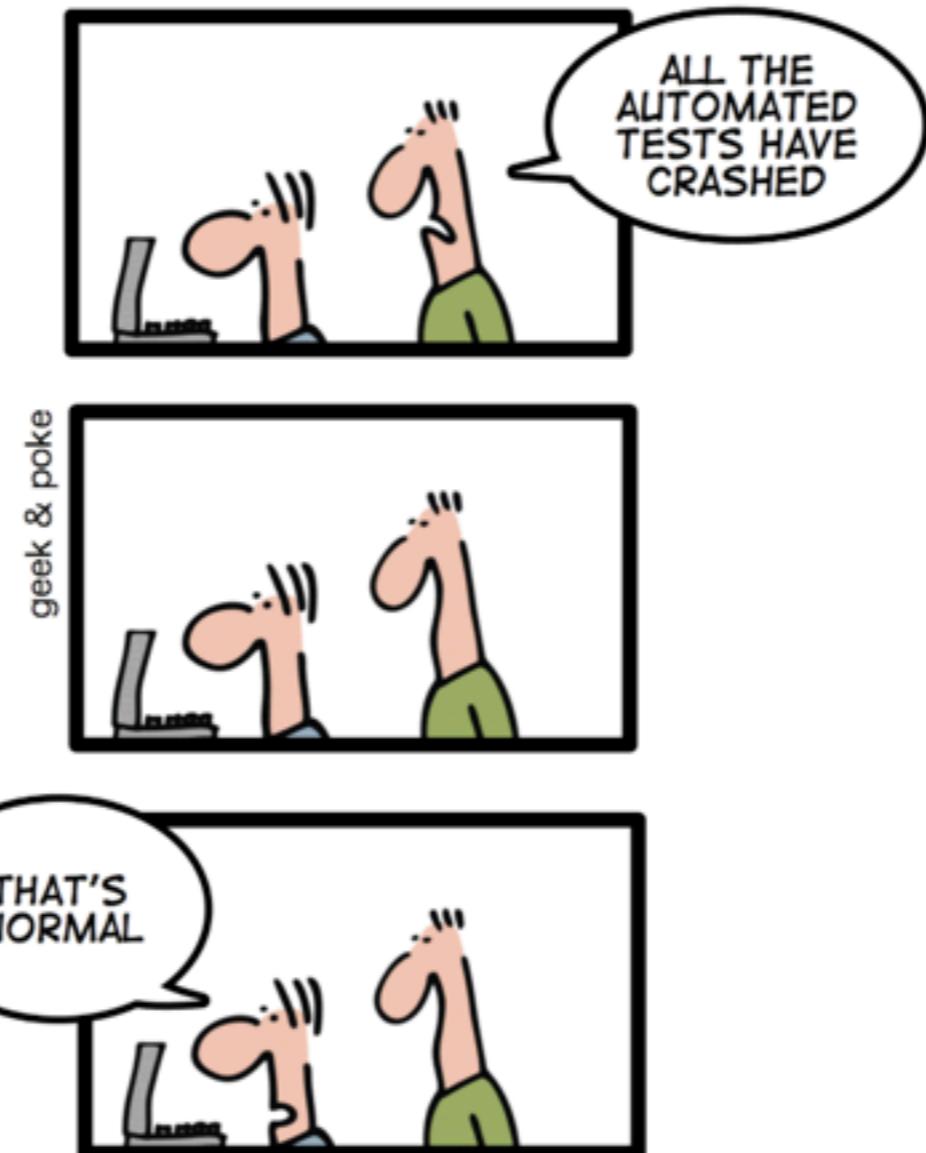
Practice 3

Make your build **self-testing**

Build process => compile, linking and **testing**



*TODAY: CONTINUOUS INTEGRATION
GIVES YOU THE COMFORTING
FEELING TO KNOW THAT
EVERYTHING IS NORMAL*

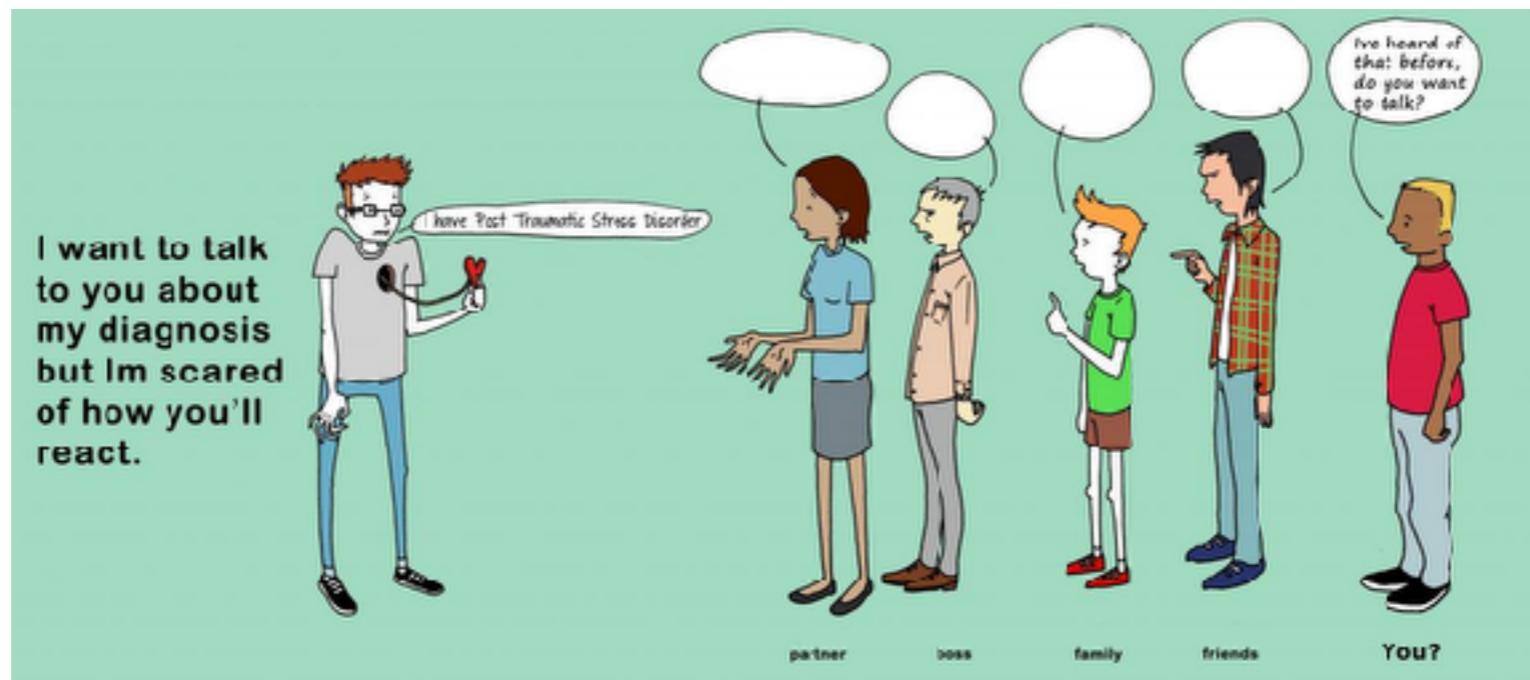


Practice 4

Everyone commits to the mainline everyday

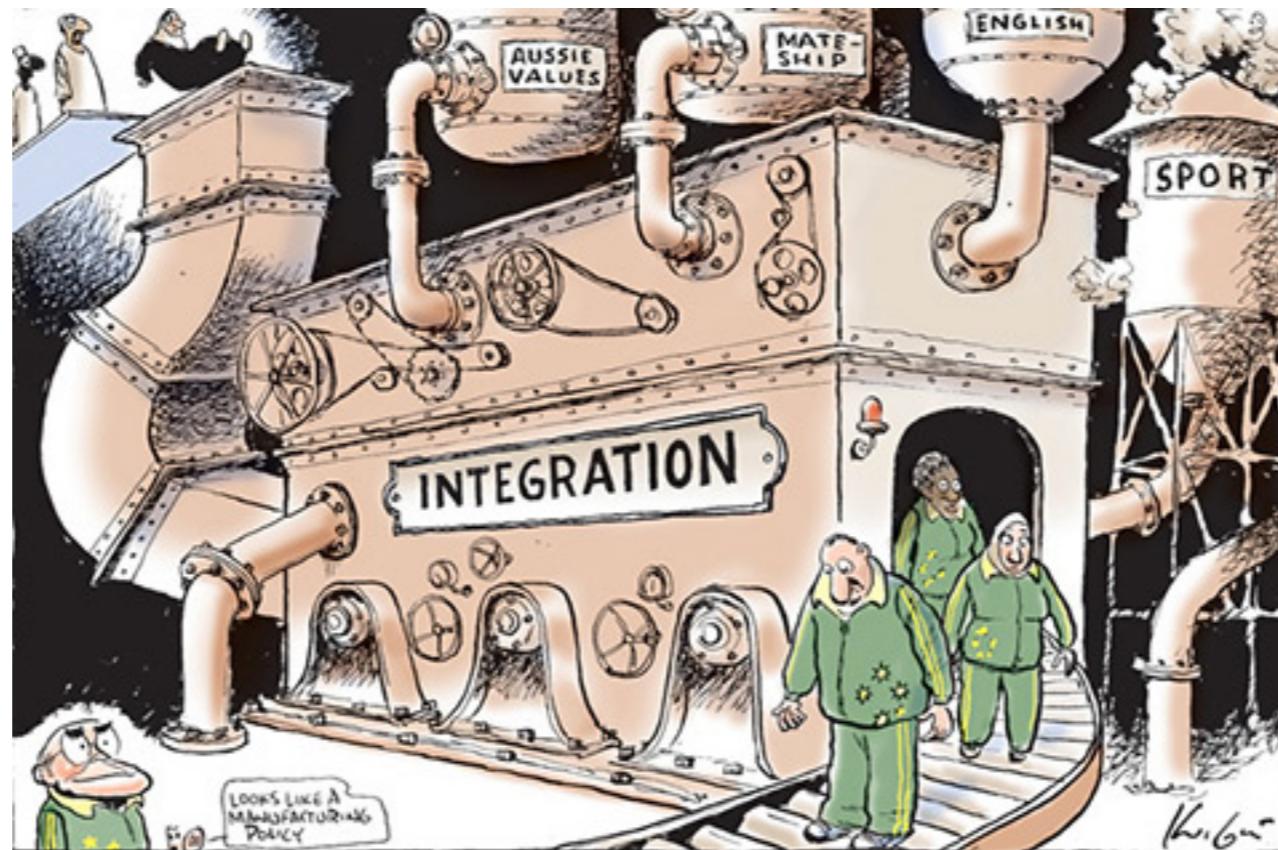
Integration is about communication

Integration allows developers to tell other developers



Practice 5

Every commits should build the mainline on an
Integration machine



Nightly build is not enough for Continuous Integration



Practice 6

Fix broken builds immediately

“Nobody has a higher priority task than fixing the build”



Practice 7

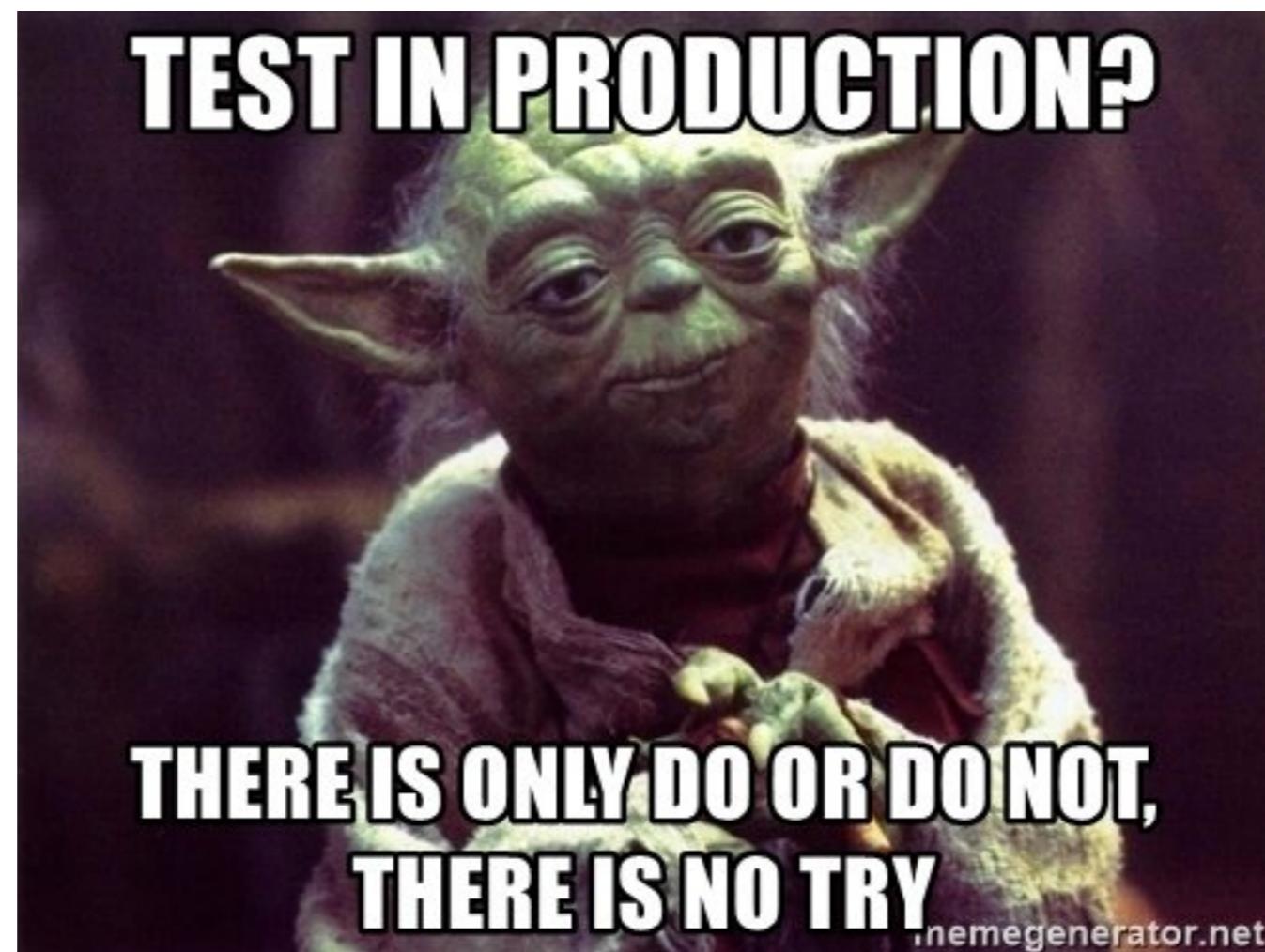
Keep the build **fast**

Continuous Integration is to provide rapid feedback



Practice 8

Test in clone of the **Production** environment



Practice 9

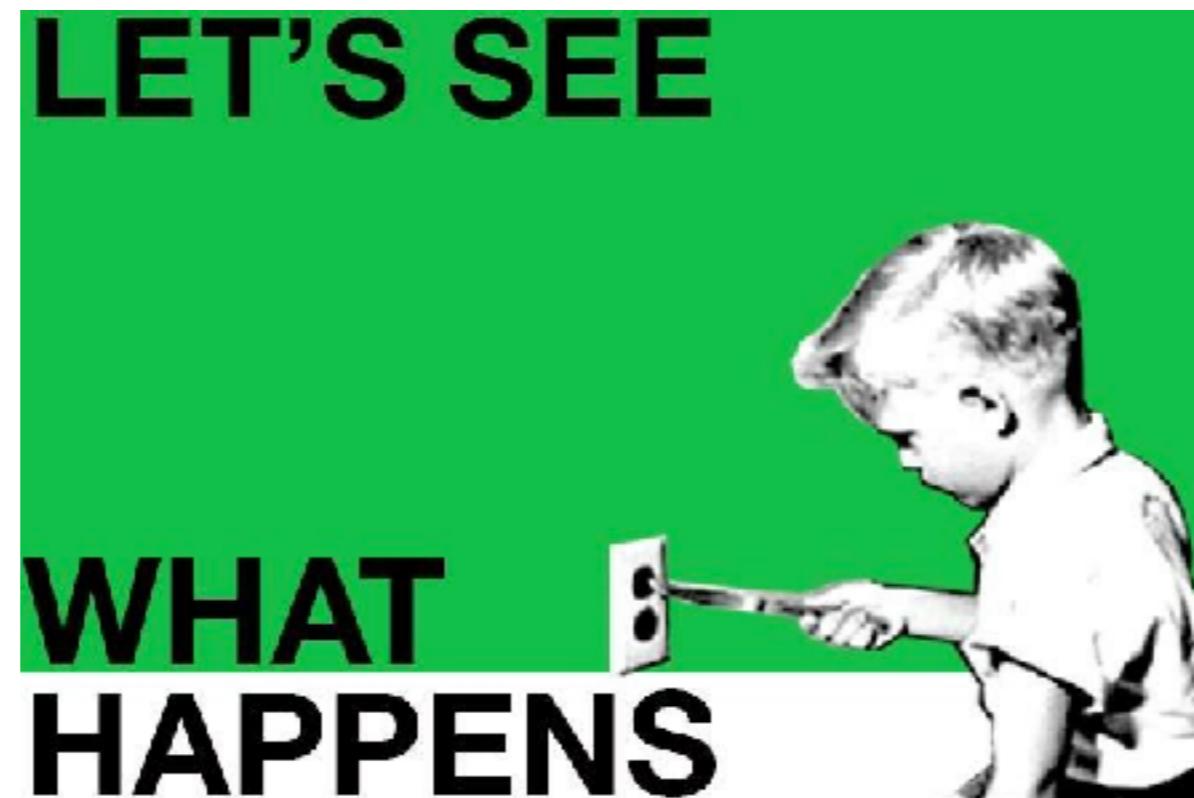
Make it easy for anyone to get
the latest executable

Make sure well known place where people can find



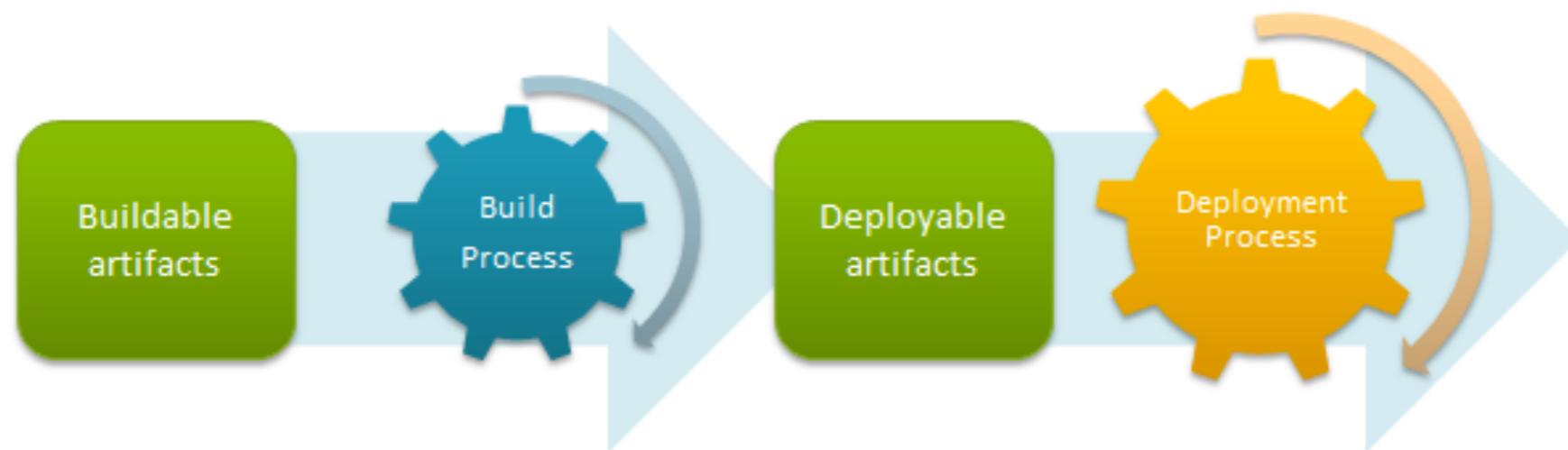
Practice 10

Everyone can see what's happening
Easier to see the state of the system and changes
Show the good information



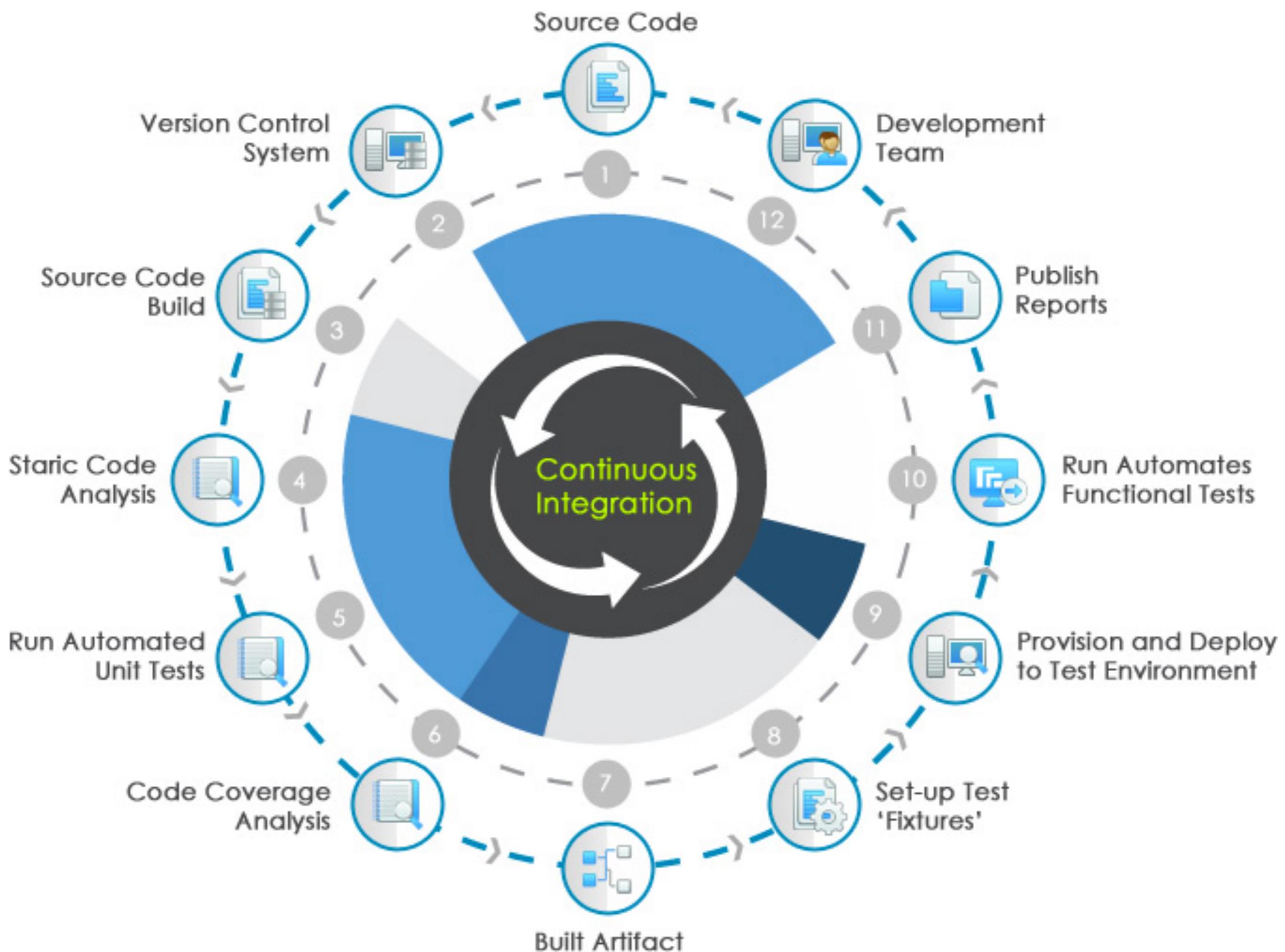
Practice 11

Automated deployment



**“Behind every successful agile
project, there is a
Continuous Integration Server”**





Let's workshop



Jenkins



Summary





Agile manifestos

THE AGILE MANIFESTO

We are uncovering better ways of developing software by doing it and helping others do it.

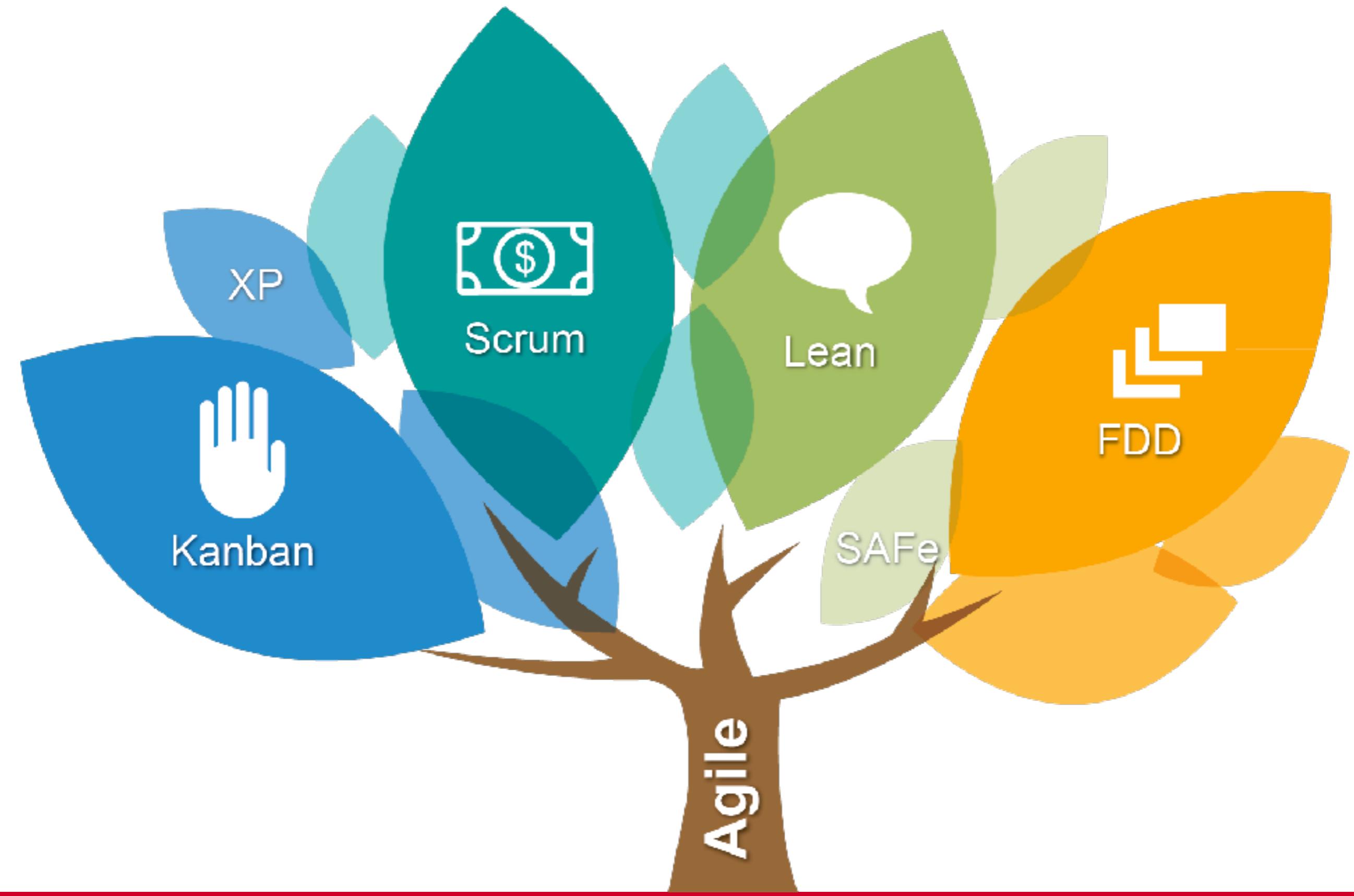
**CUSTOMER
COLLABORATION**
over contract negotiation

**RESPONDING TO
CHANGE**
over following a plan

**INDIVIDUALS AND
INTERACTIONS**
over processes and tools

**WORKING
SOFTWARE**
over full documentation





Agile principles

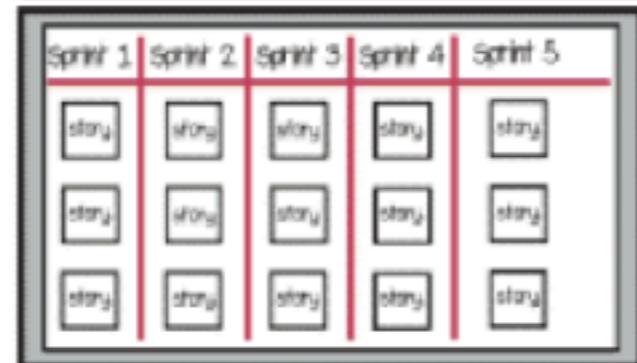
1 Satisfy the **customer**



Welcome **change**



Deliver **frequently**



4 Work **together**



5 Trust and **support**



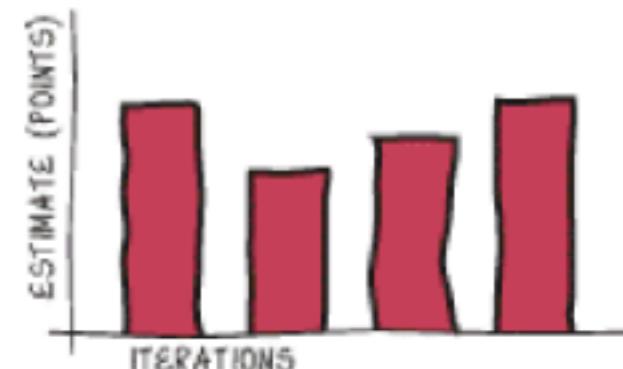
Face-to-face **conversation**



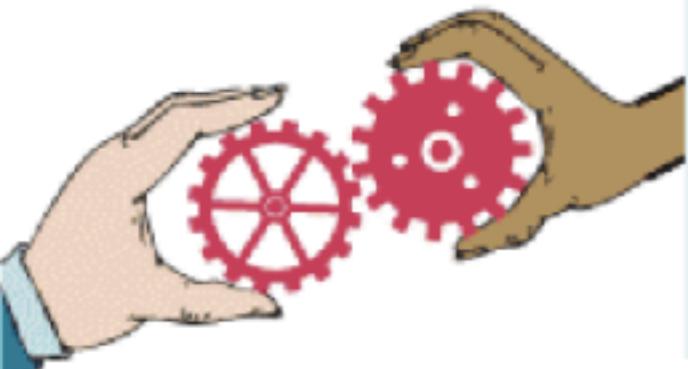
Working **software**



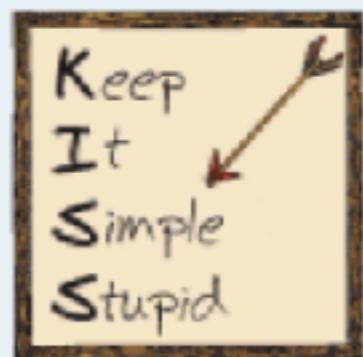
8 Sustainable **development**



9 Technical **Excellence**



10 Maintain **simplicity**



11 Self-organizing **teams**

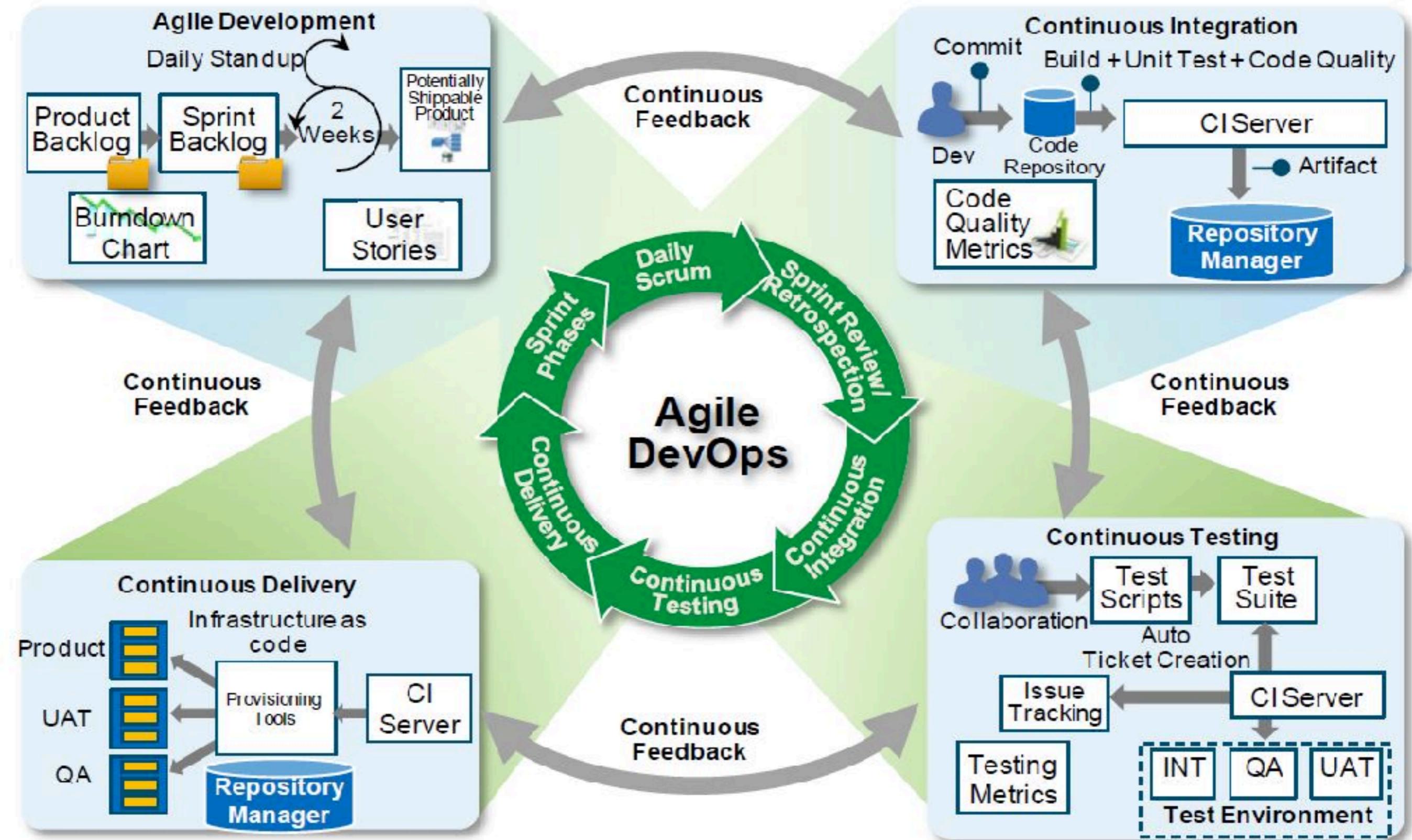


12 Reflect and **adjust**

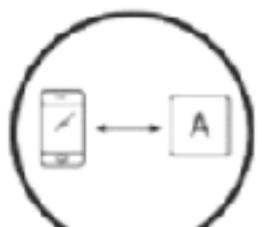


Origin by <https://www.knowledgetrain.co.uk>, modified by Jacky Shen

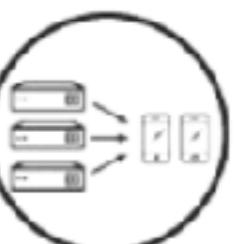




Continuous Improvement



Monolith



N-Tier



Microservices

Applications



Datacenter



Hosted

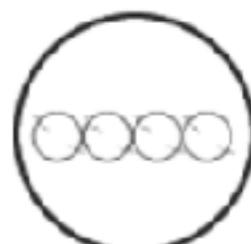


Hybrid

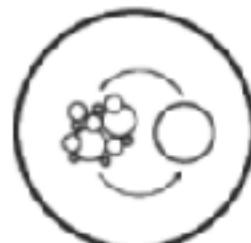
Infrastructure



Waterfall



Agile



DevOps

Process



Closing session

