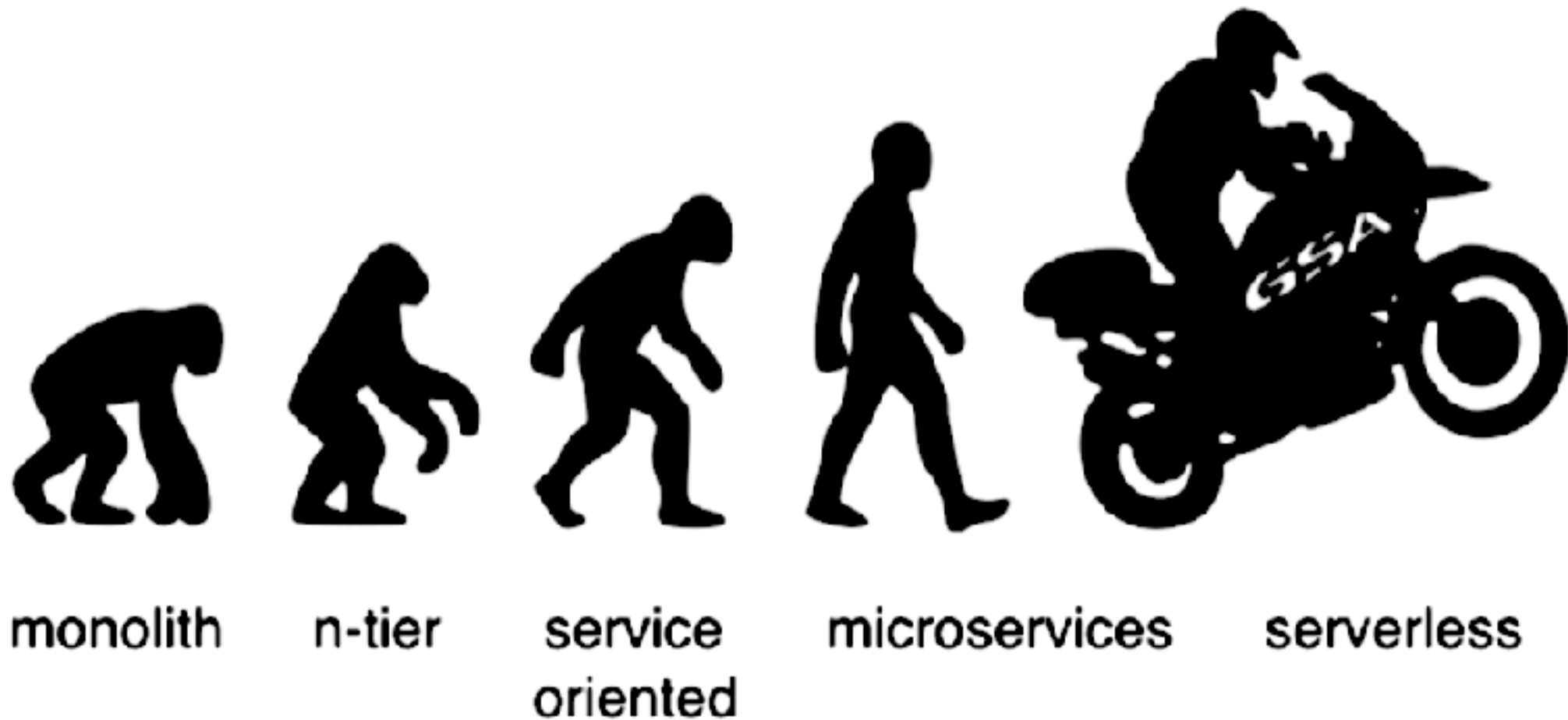


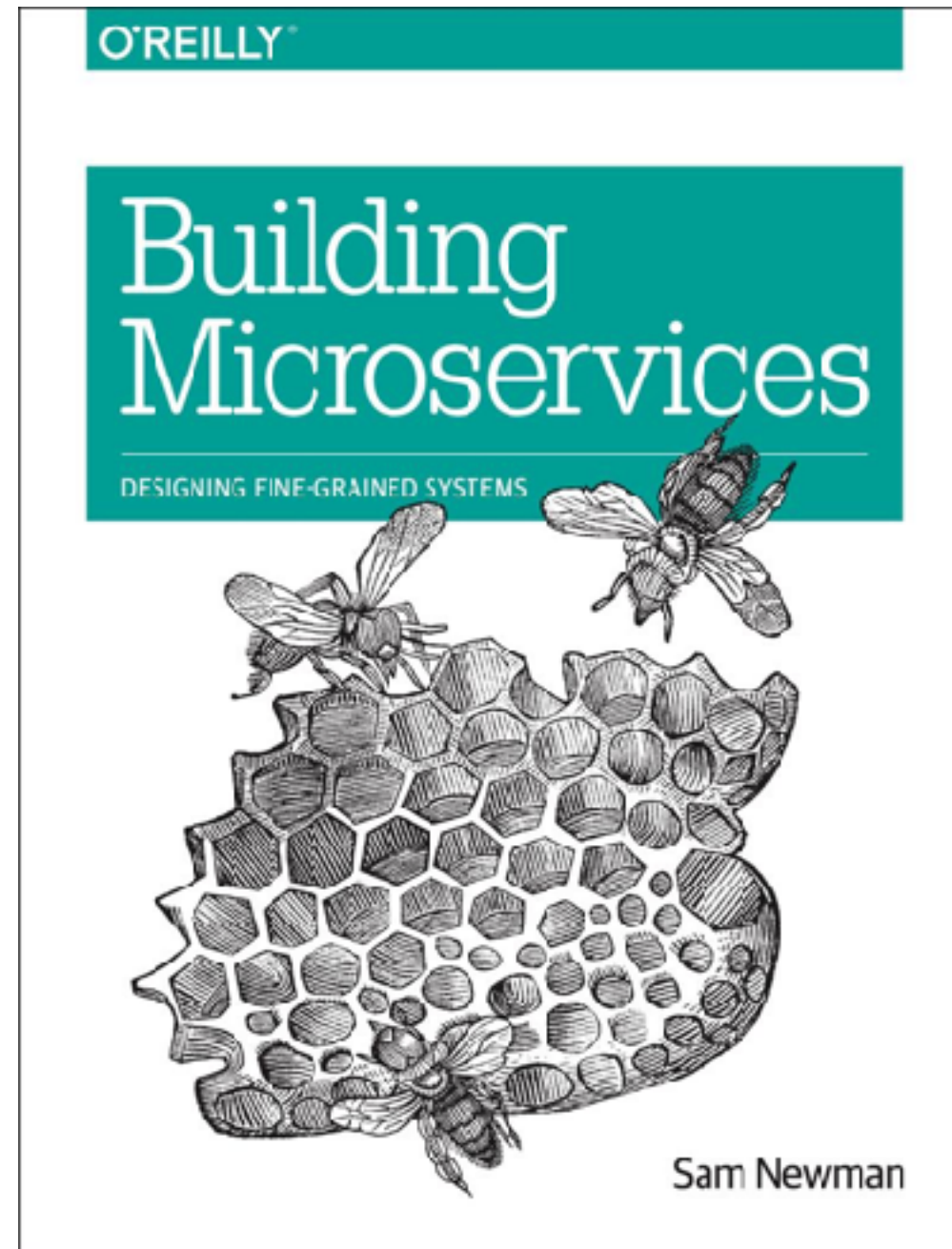
Microservice



@somkiat



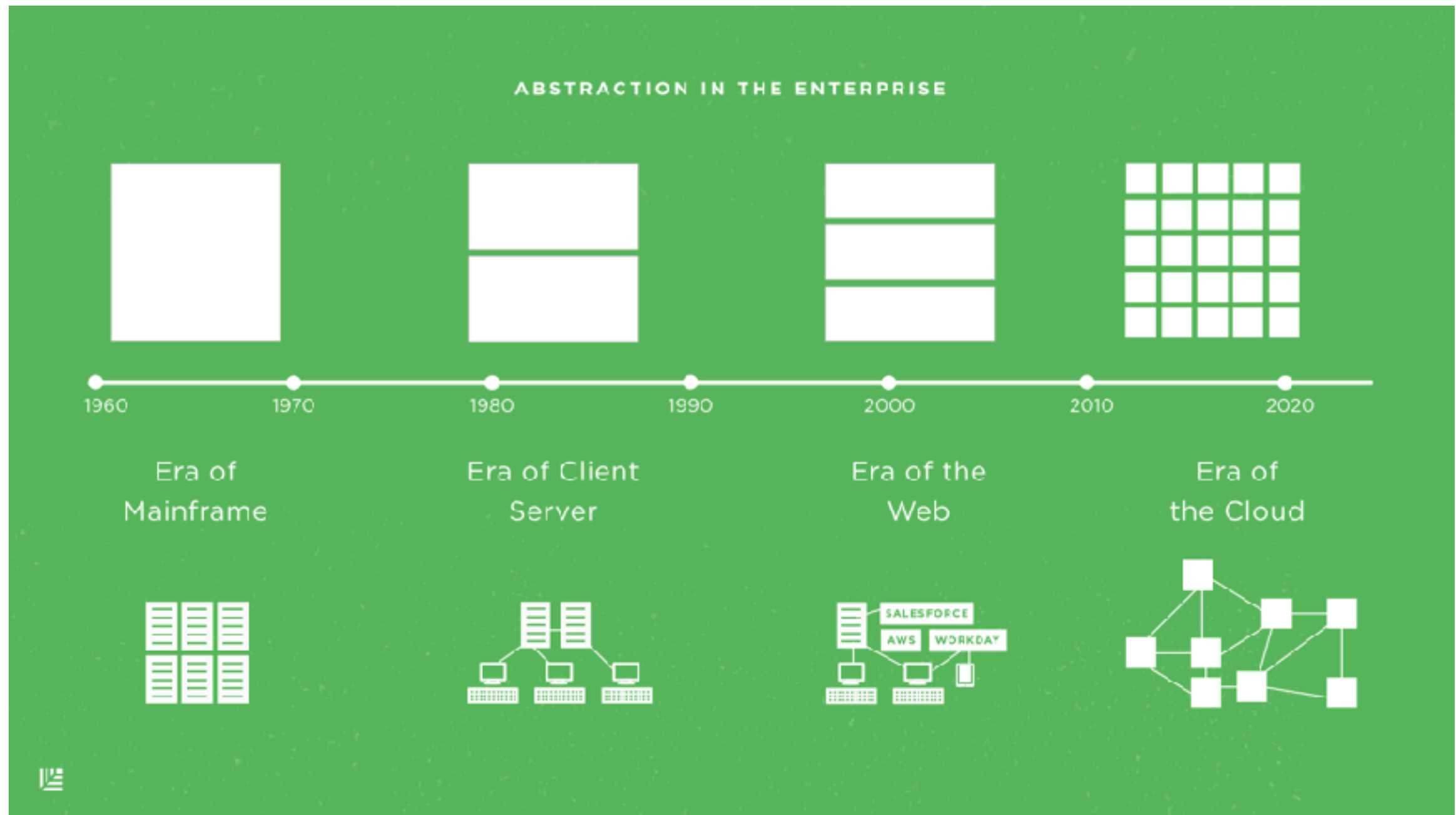
Reference



<http://shop.oreilly.com/product/0636920033158.do>



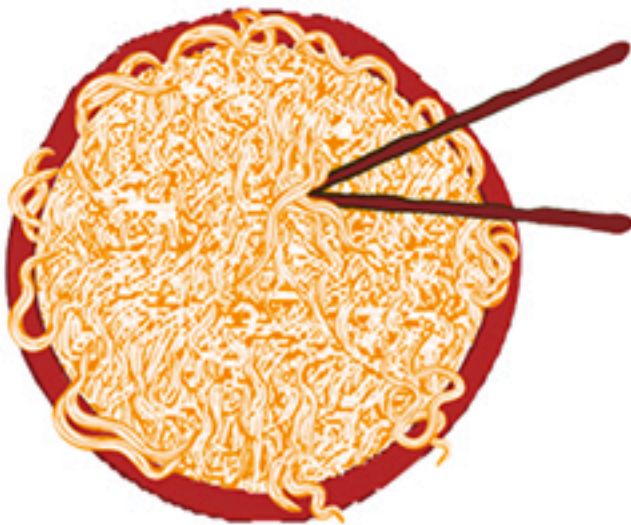
Evolution of Architecture



Developer's perspective

1990s and earlier

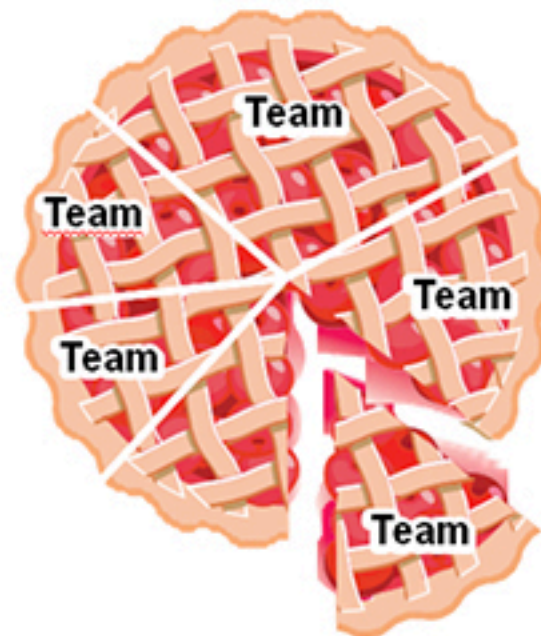
Pre-SOA (monolithic)
Tight coupling



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



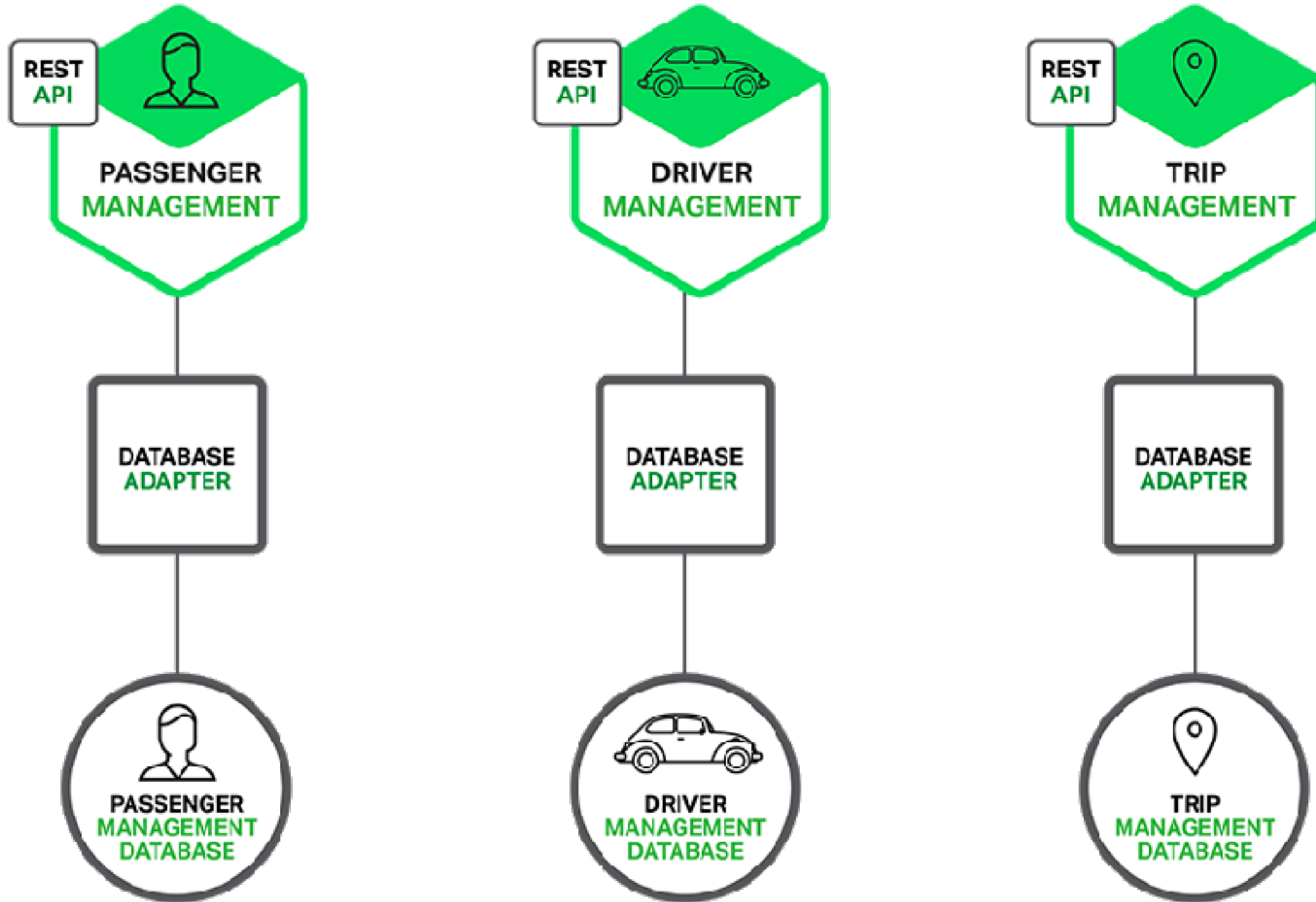
Developers can create and activate new microservices without prior coordination with others. Their adherence to MSA principles makes continuous delivery of new or modified services possible.



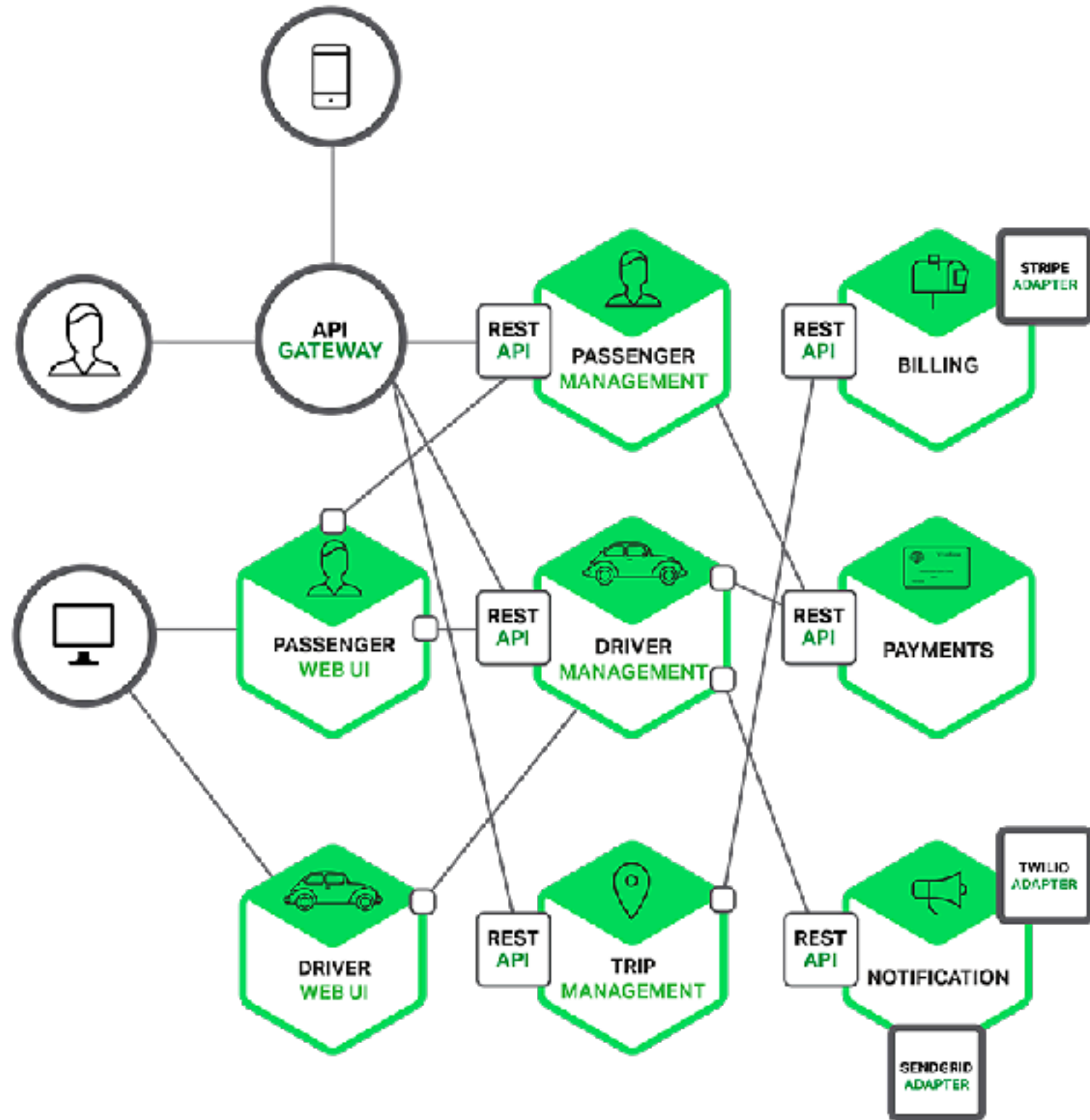
Microservice



Microservice



Microservice



What is microservice ?



What is microservice ?

Small services



What is microservice ?

Small services

Running in its own process

ไม่ใช่เอา service ไป deploy ไว้ที่เดียวกัน บน server เดียวกัน ถ้า server พัง ก็ตายหมด ไม่ใช่ microservices
service อยู่บนเครื่องเดียวกัน แต่ต้อง isolate ออกจากกัน และ ออกจาก OS ด้วย

1 VM มีได้หลาย container แต่ละ container มี service ที่อยู่คนละ environment แยกกัน



What is microservice ?

Small services

Running in its own process

Communicating with lightweight mechanism

ต้อง lightweight



What is microservice ?

Small services

Running in its own process

Communicating with lightweight mechanism

Independently deployable

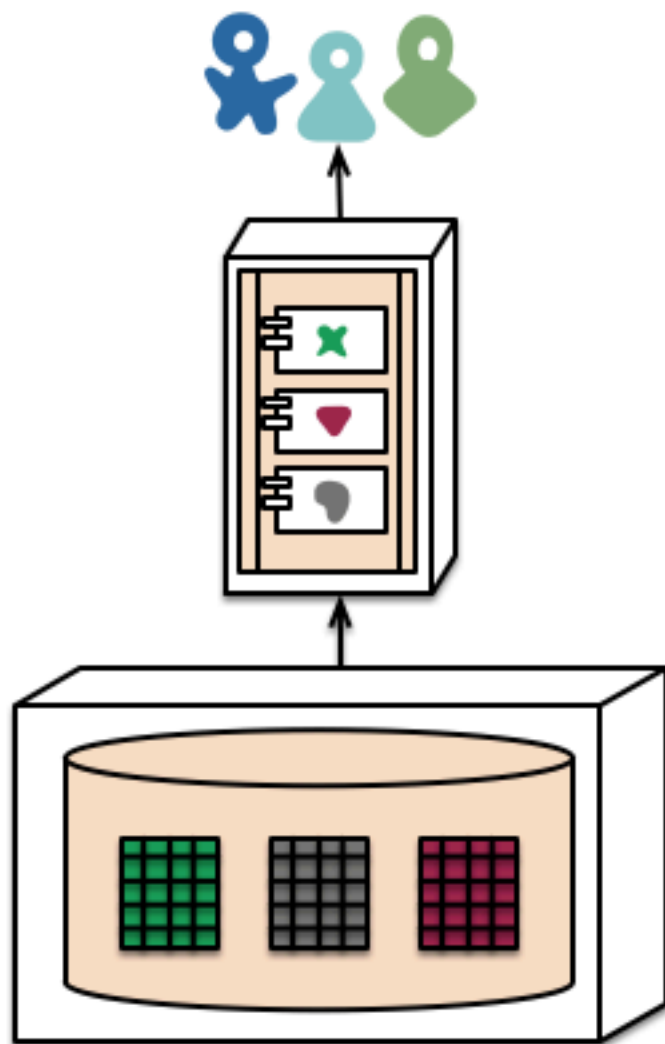
สามารถ deploy แยกกันได้ แต่ในทางปฏิบัติ service มักจะ depend กัน ก็ต้อง deploy ให้ถูก



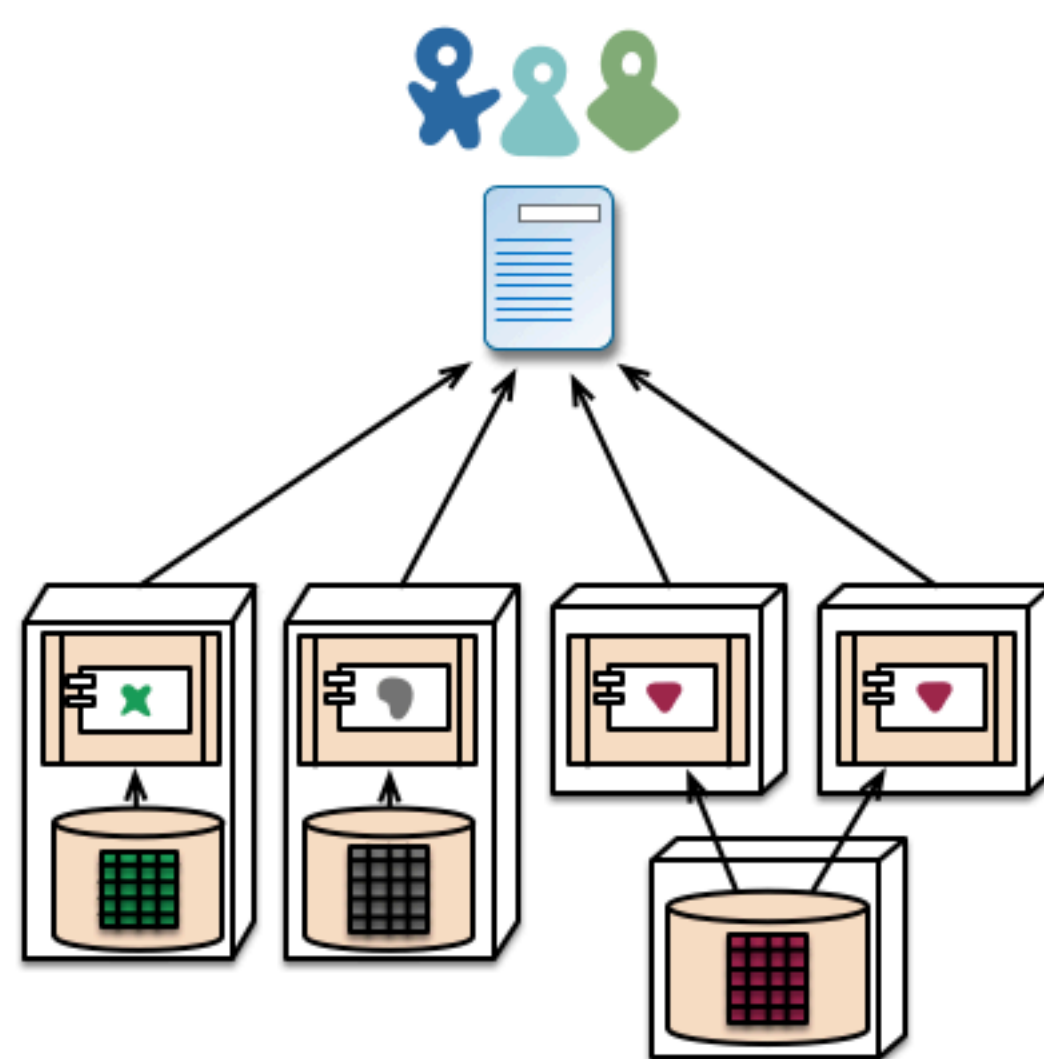
Benefits



Technology Heterogeneity



monolith - single database

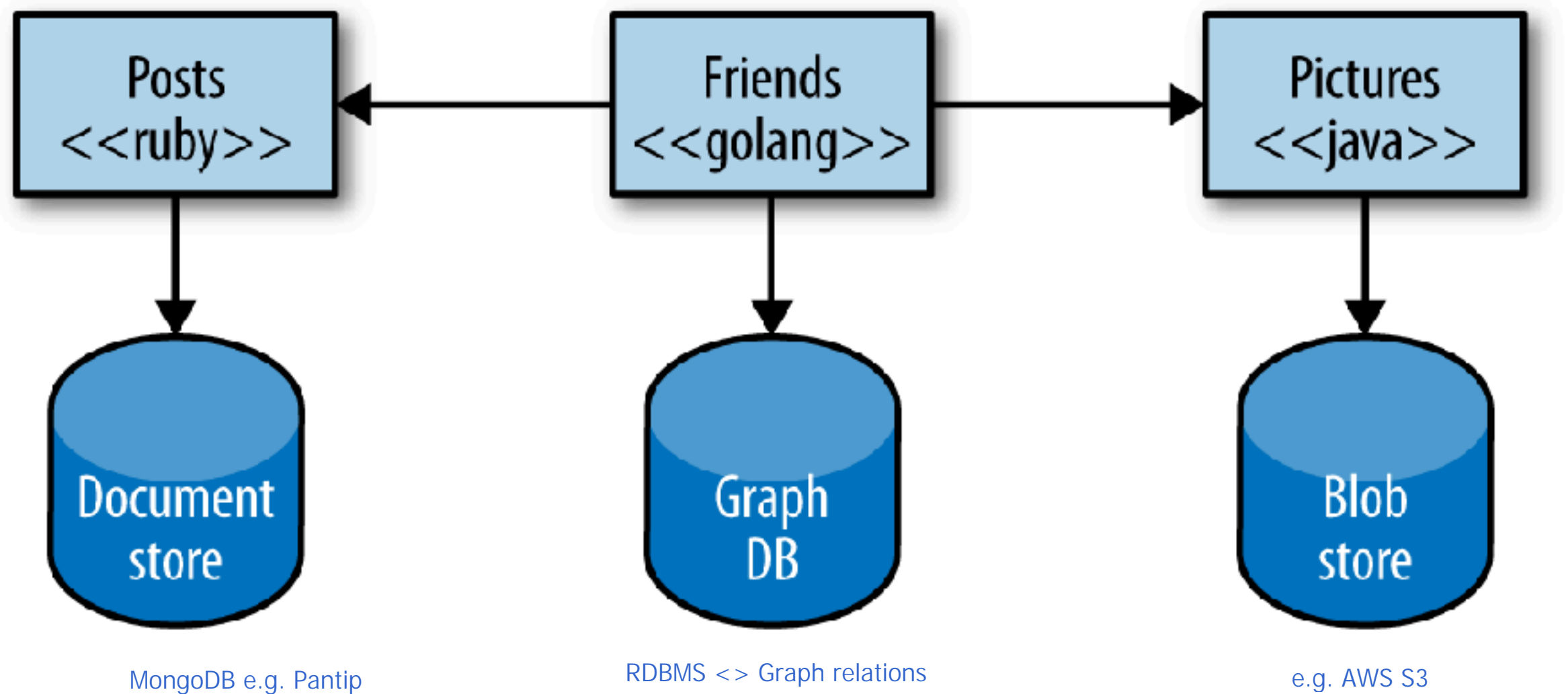


microservices - application databases



Technology Heterogeneity

แต่ละ service คนละภาษา data store ก็ต่างกัน



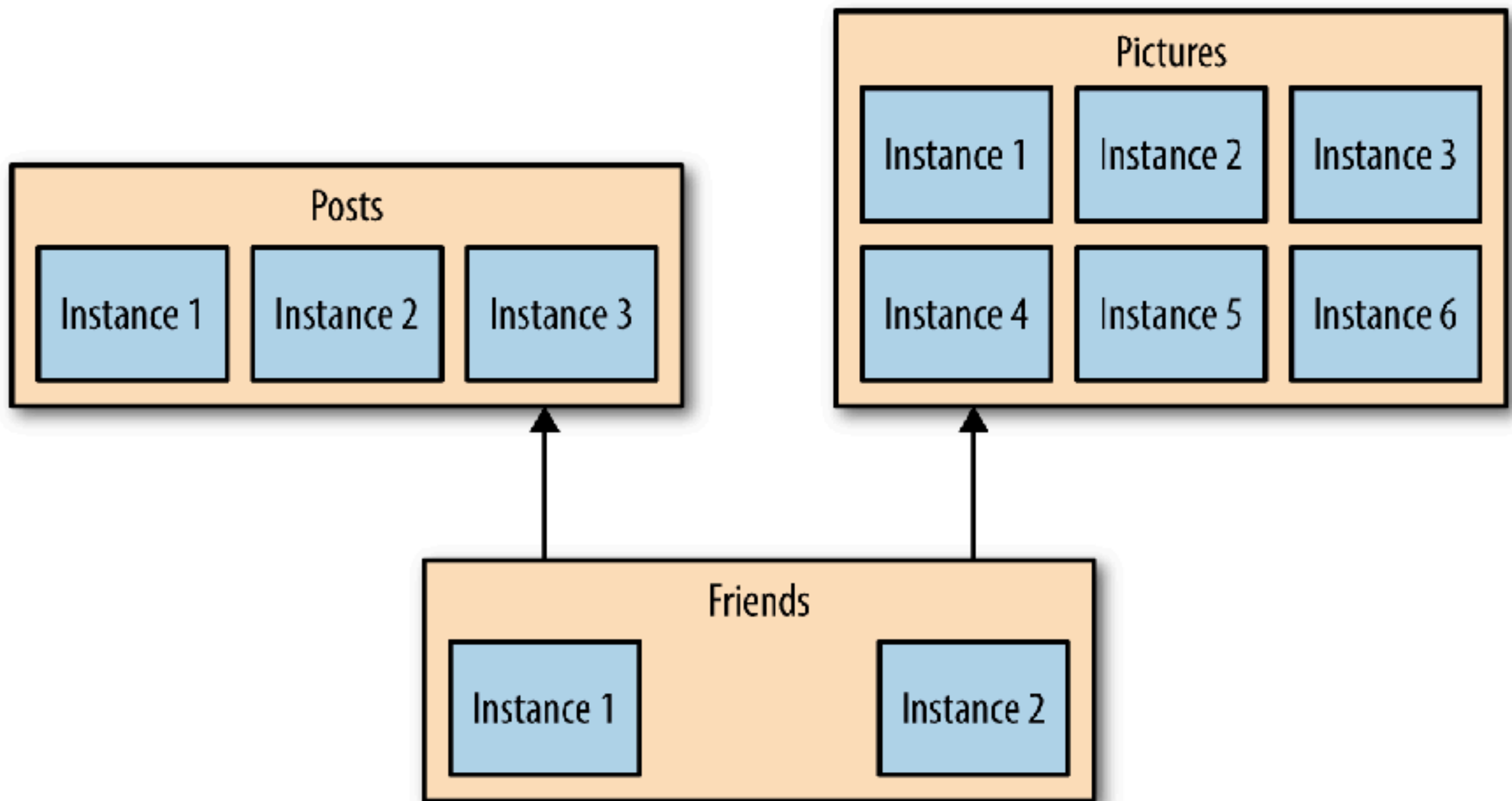
MongoDB e.g. Pantip

RDBMS <> Graph relations

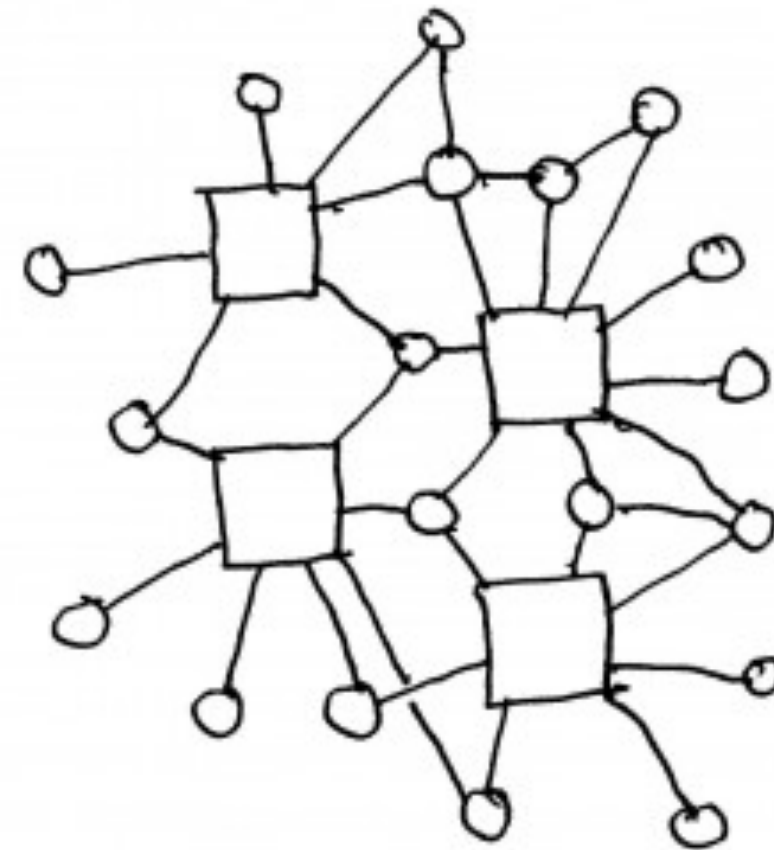
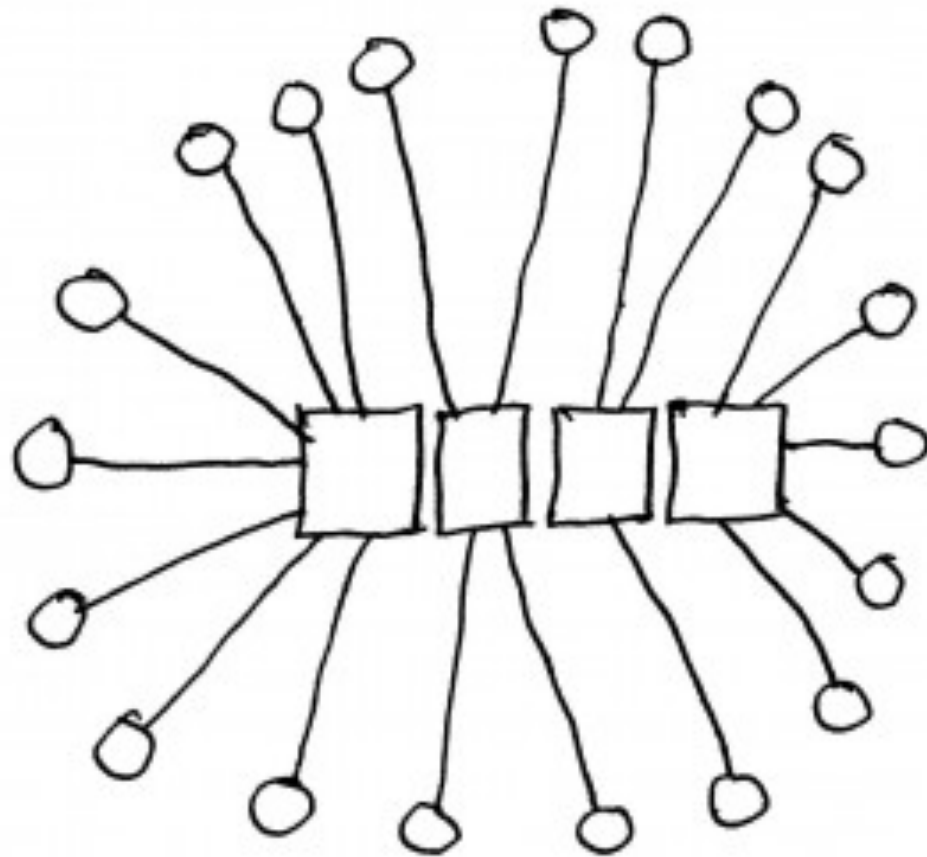
e.g. AWS S3



Scaling



Resilience

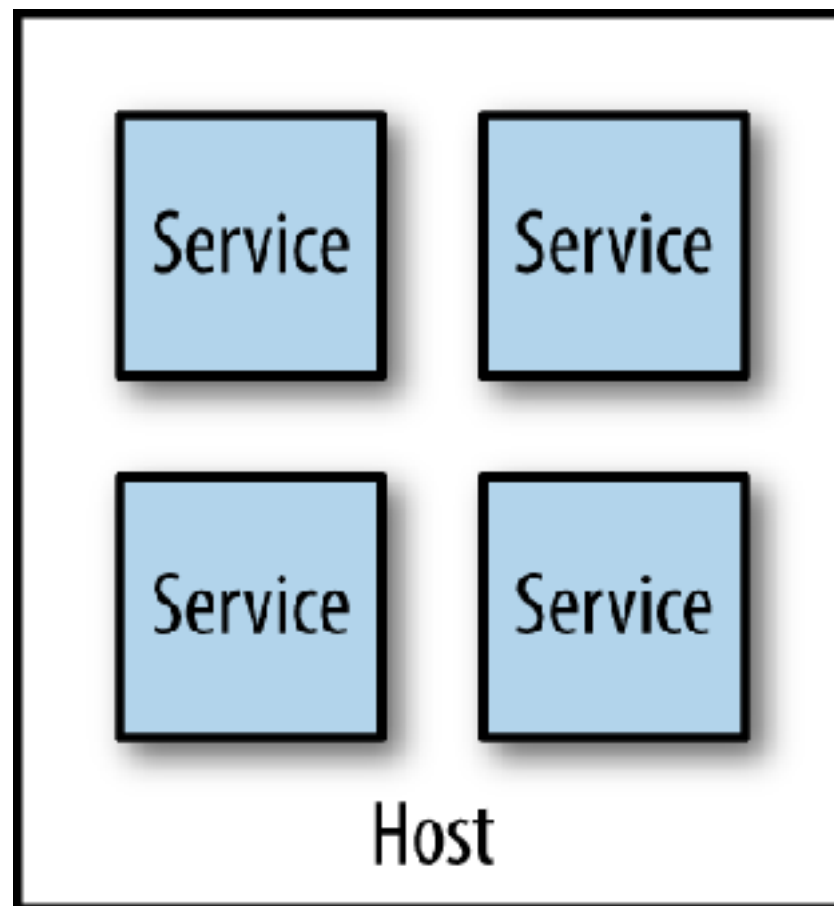


Distributed system



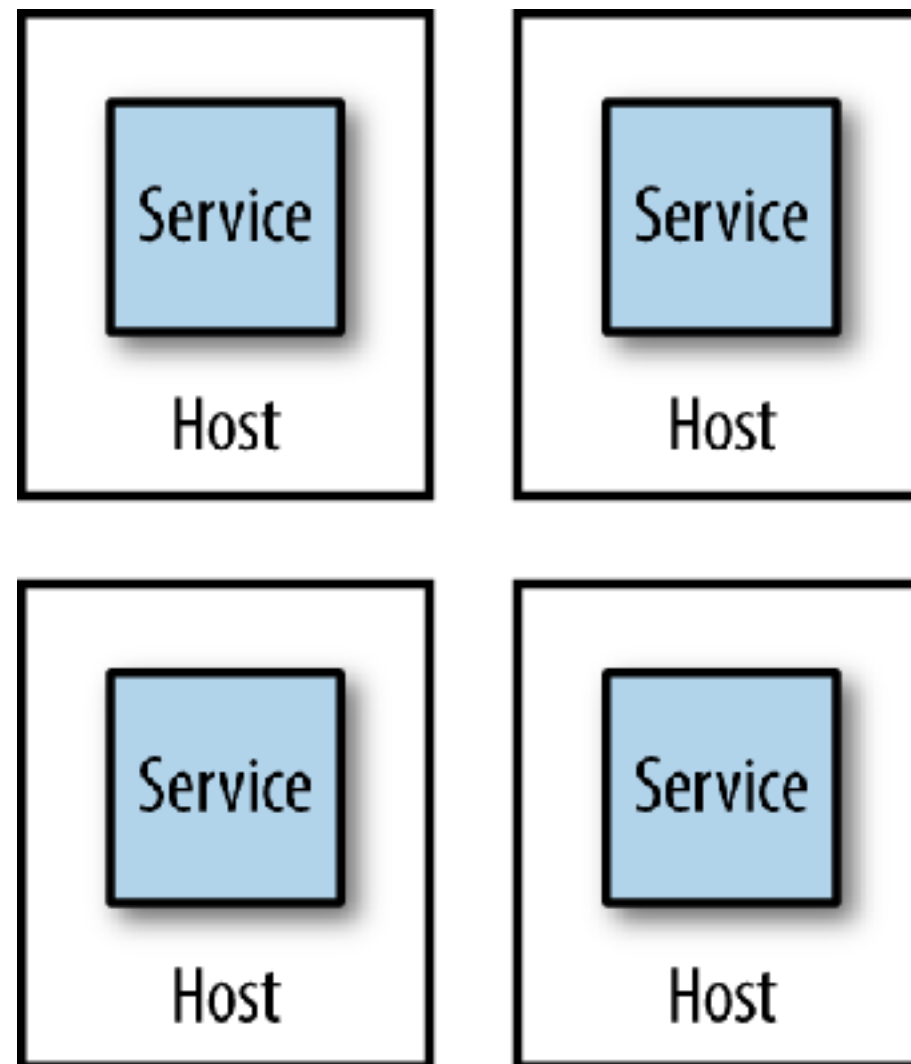
Ease of Deployment

Multiple services per host



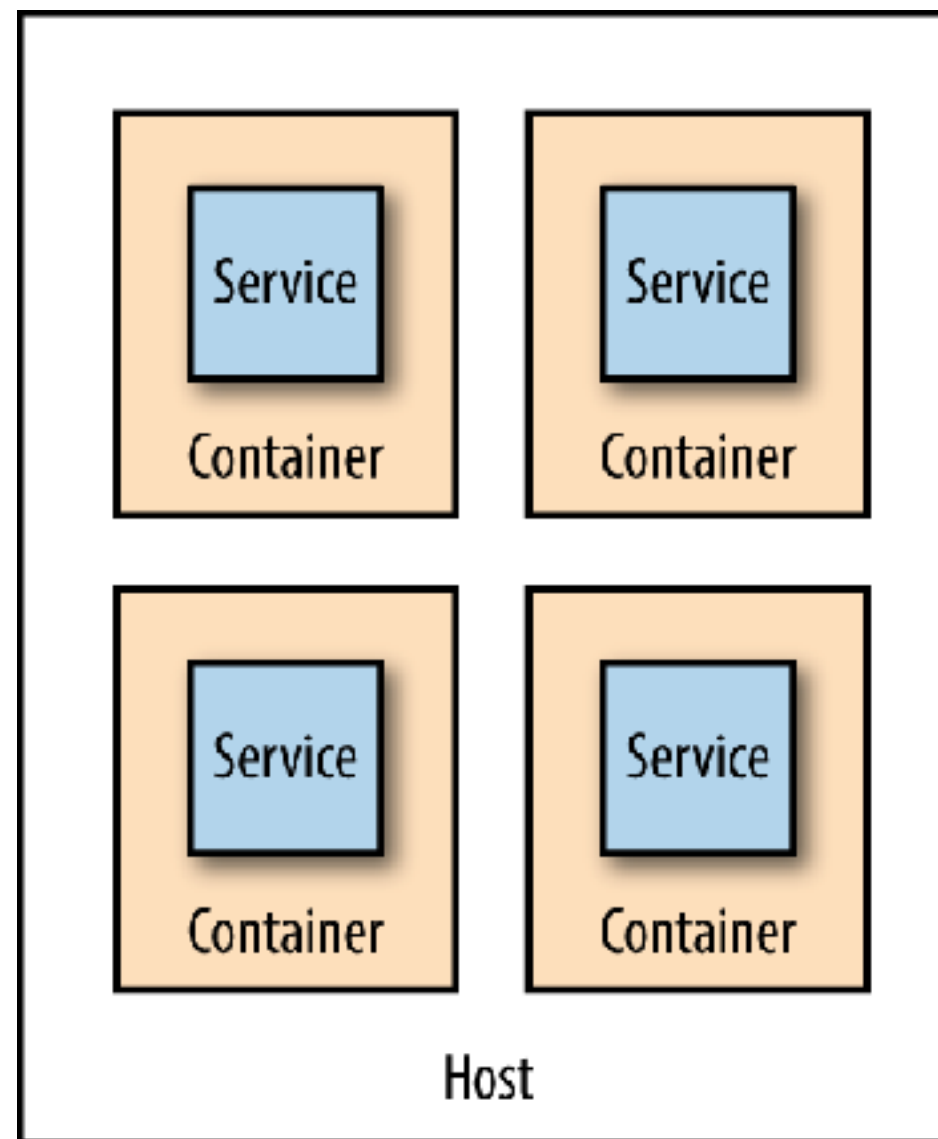
Ease of Deployment

Single services per host



Ease of Deployment

Running services in separate containers



How to model services ?



Good service ?

Loose coupling
High cohesion

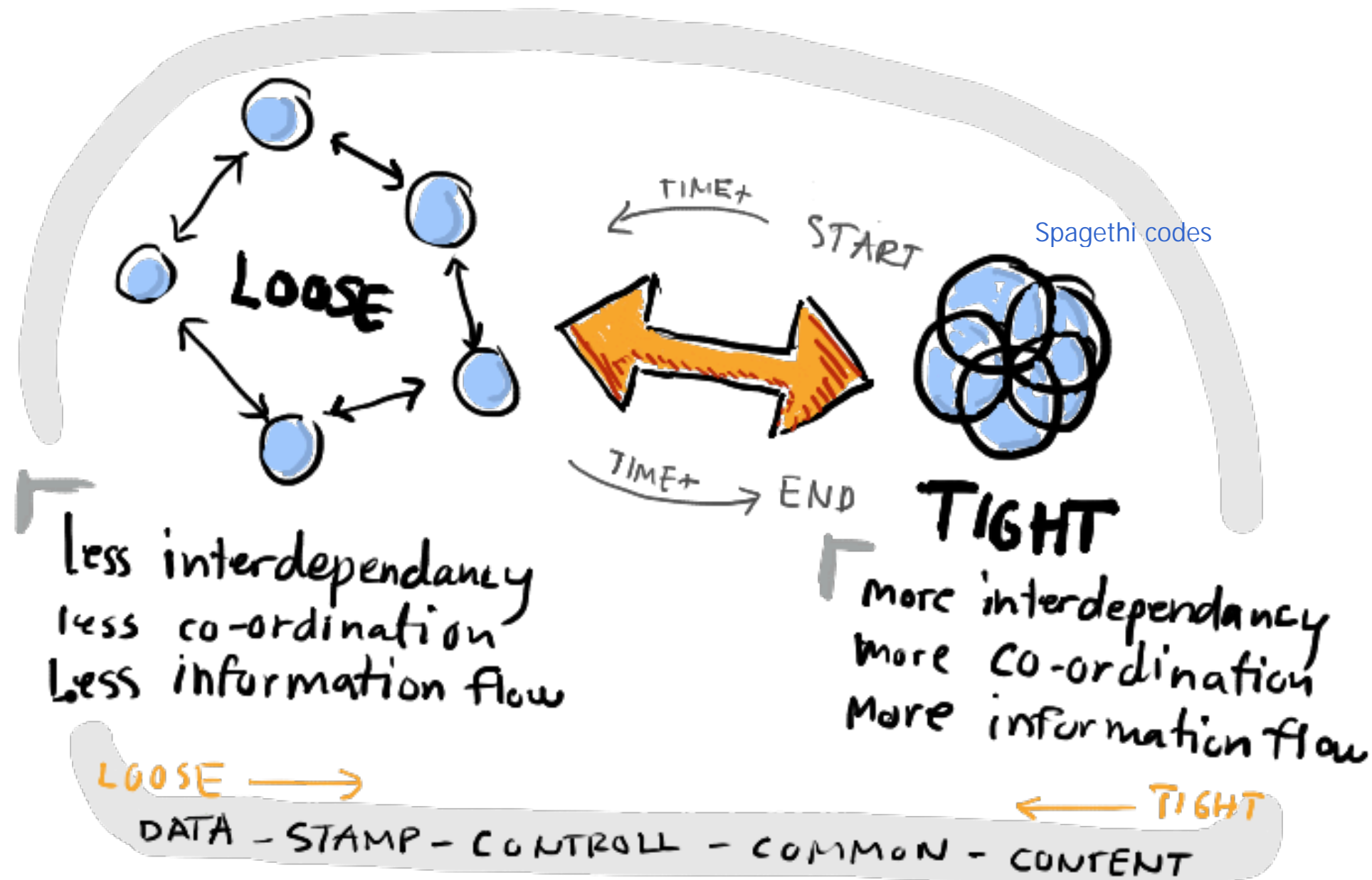
อะไรควรแยก อะไรควรรวม
ดูหน้าที่ความรับผิดชอบ และ ความสัมพันธ์ระหว่าง service



Loose coupling



Loose coupling



<https://infomgmt.wordpress.com/2010/02/18/a-visual-respresentation-of-coupling/>



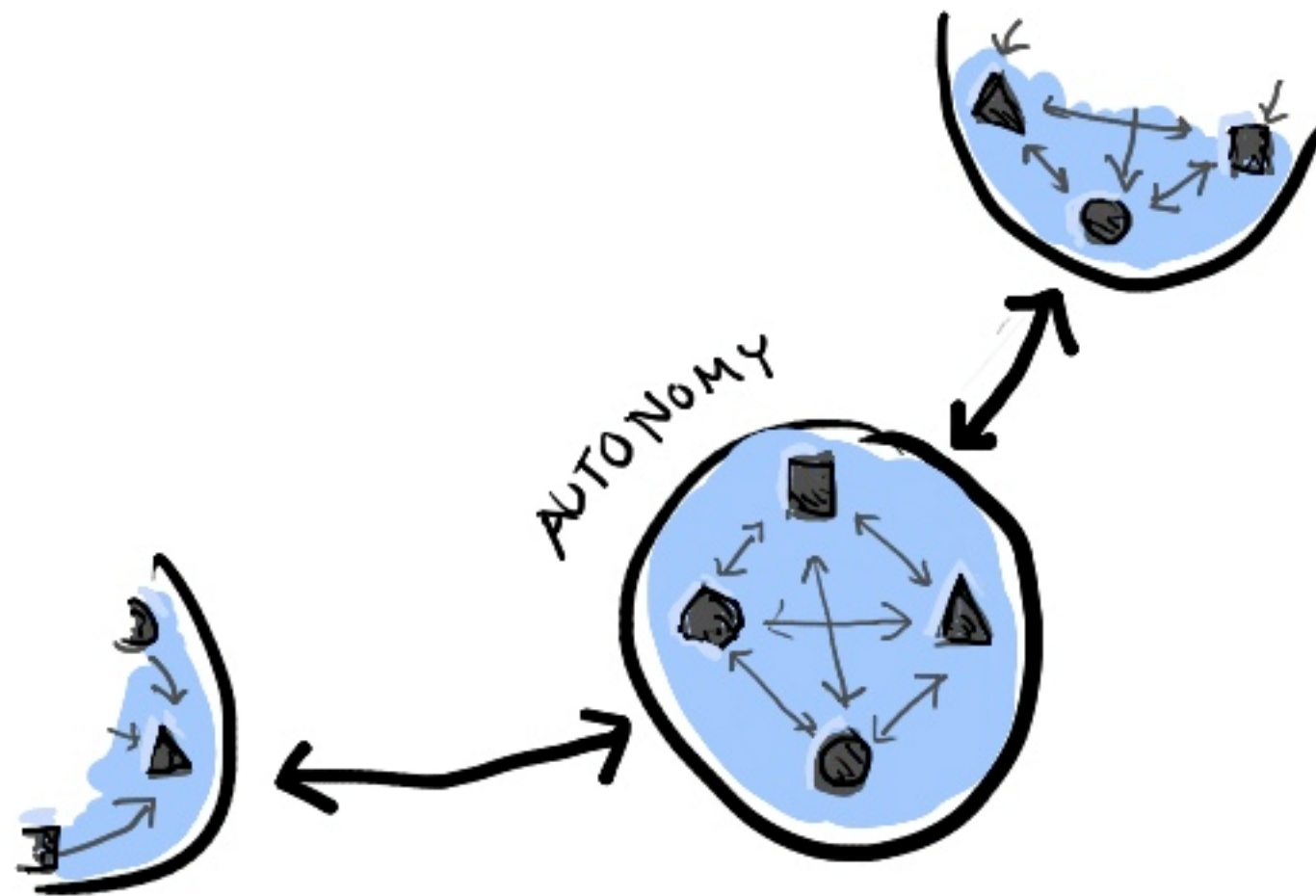
Loose coupling



<https://infomgmt.wordpress.com/2010/02/18/a-visual-respresentation-of-coupling/>



Loose coupling



<https://infomgmt.wordpress.com/2010/02/18/a-visual-respresentation-of-coupling/>

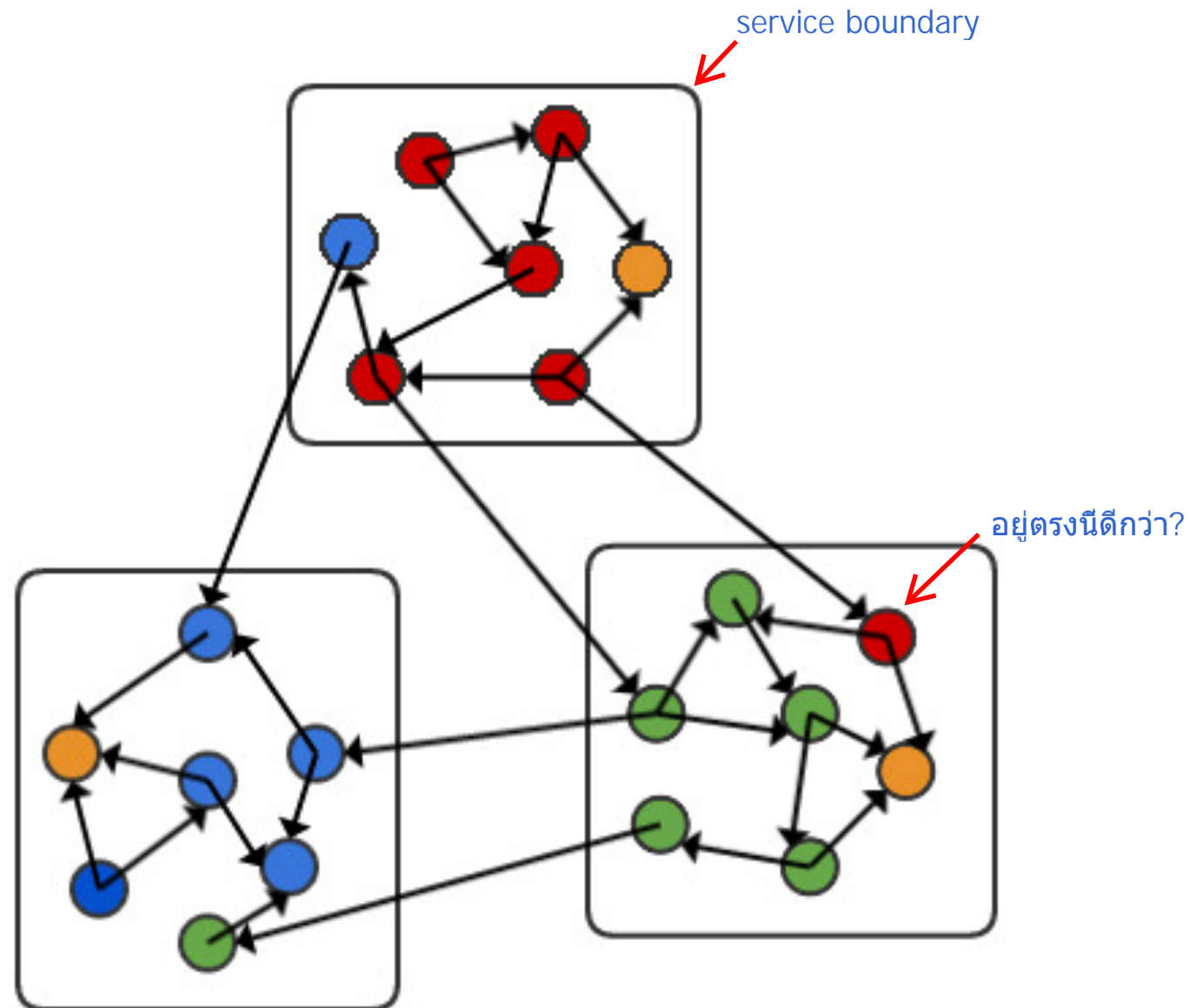


High cohesion



High cohesion

ถ้ามองไม่ออก ก็วาดออกมา

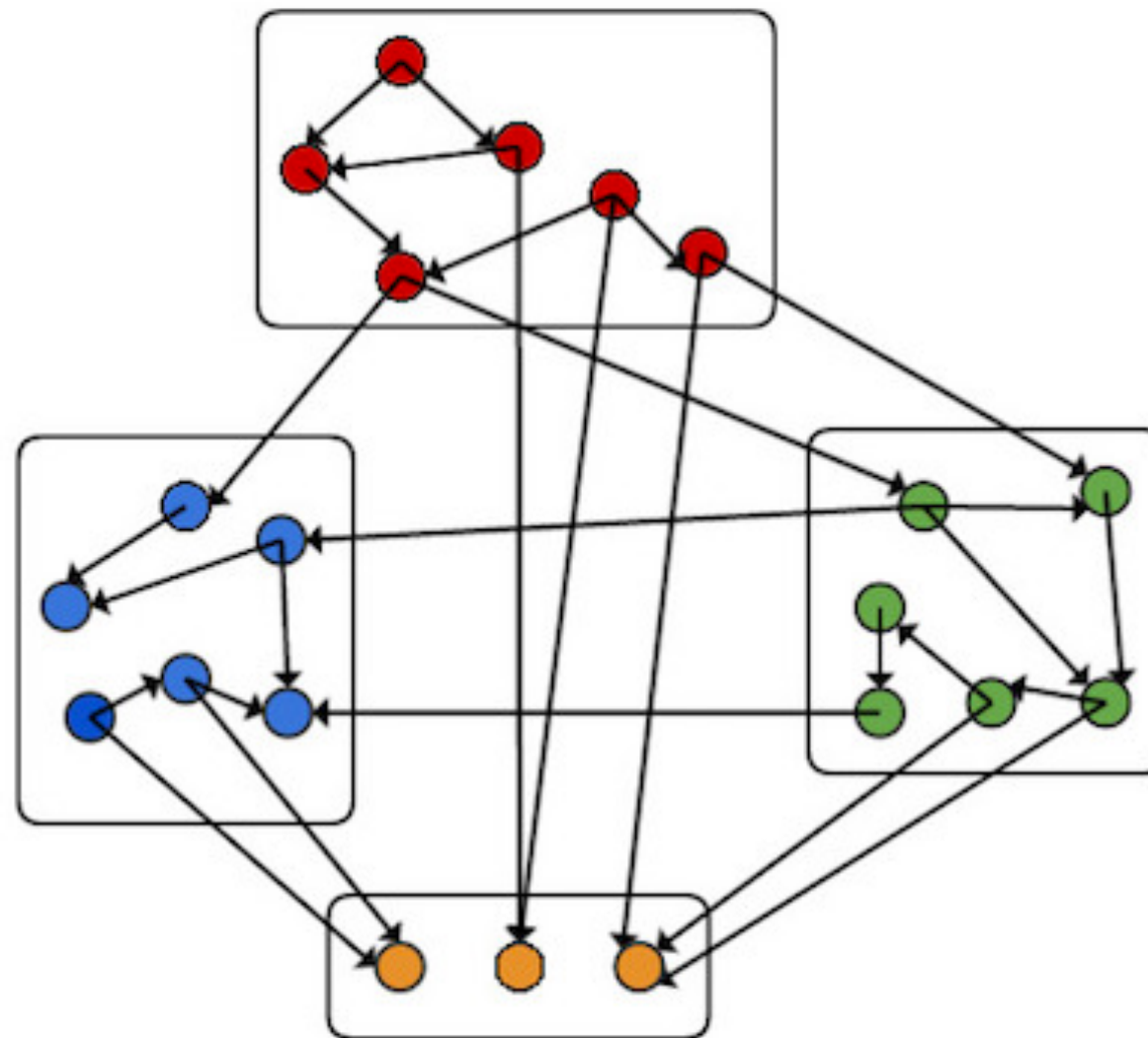


ถ้าลูกศรเข้า-ออกจากกล่องเยอะ ก็อาจจะจัดกลุ่มใหม่

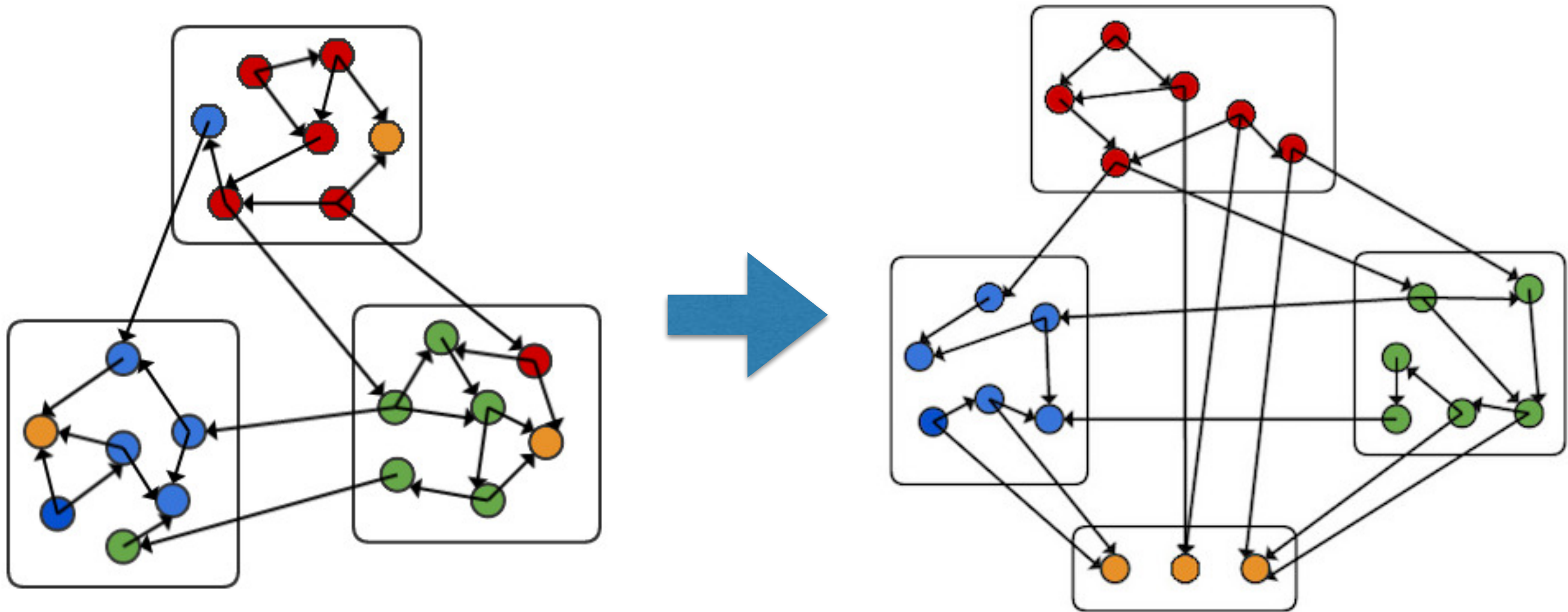
ใครให้เหตุผลดีกว่า ก็มักจะเป็นไปตามนั้น



High cohesion



High cohesion

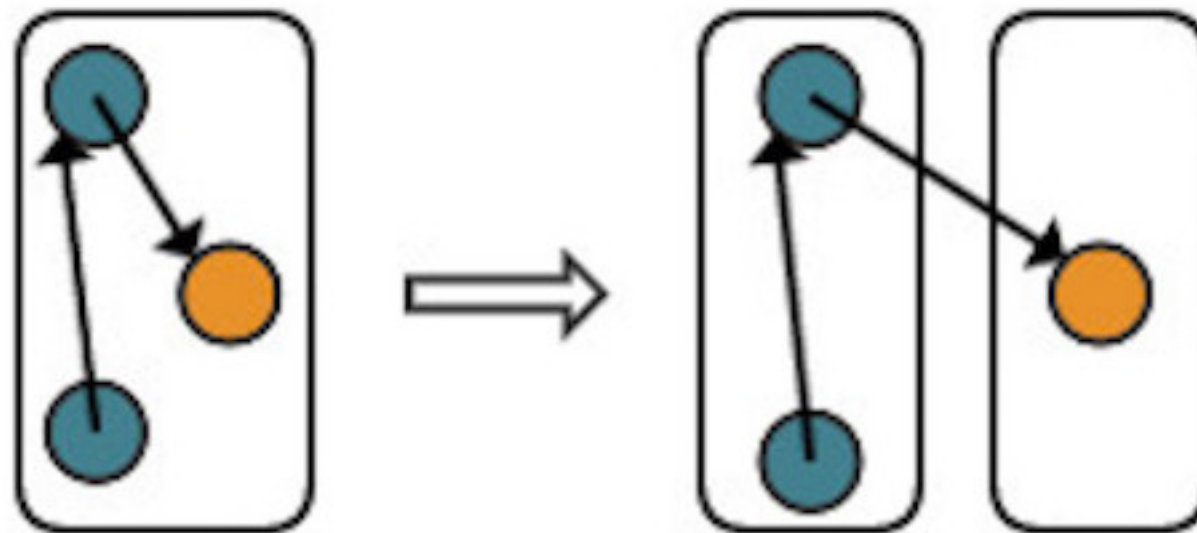


เส้นเยอะกว่าเก่า ก็ไม่น่าจะดี?

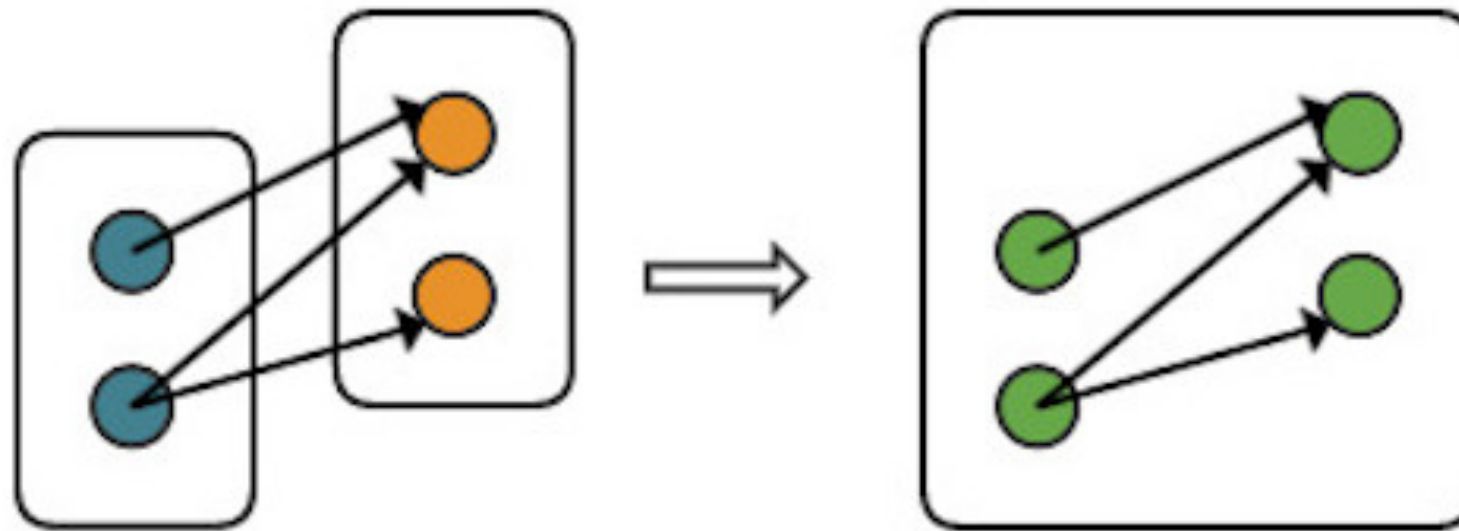
มี Tool ของ twitter เป็น Java เขียนด้วย Scalar ในการทำ visualization ของ service graph



Same responsibility together



Grouping

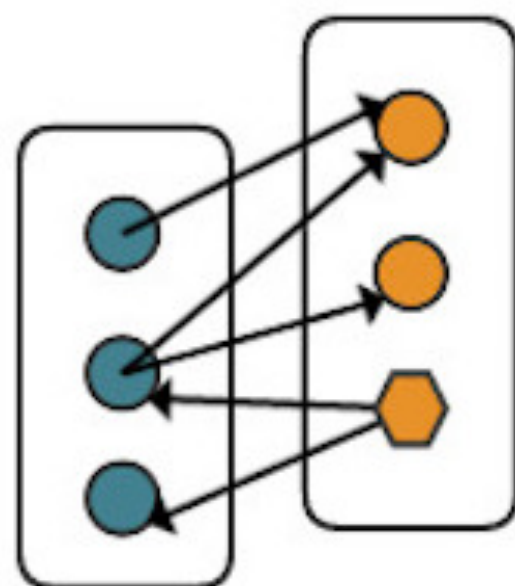


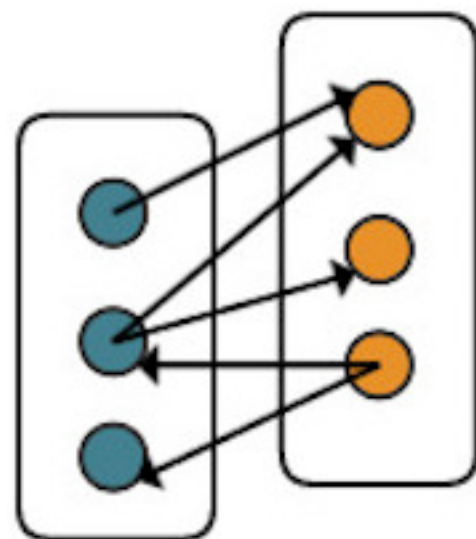
แยกได้ ก็ยุบได้
แยกผิด ก็แยกใหม่



Quiz



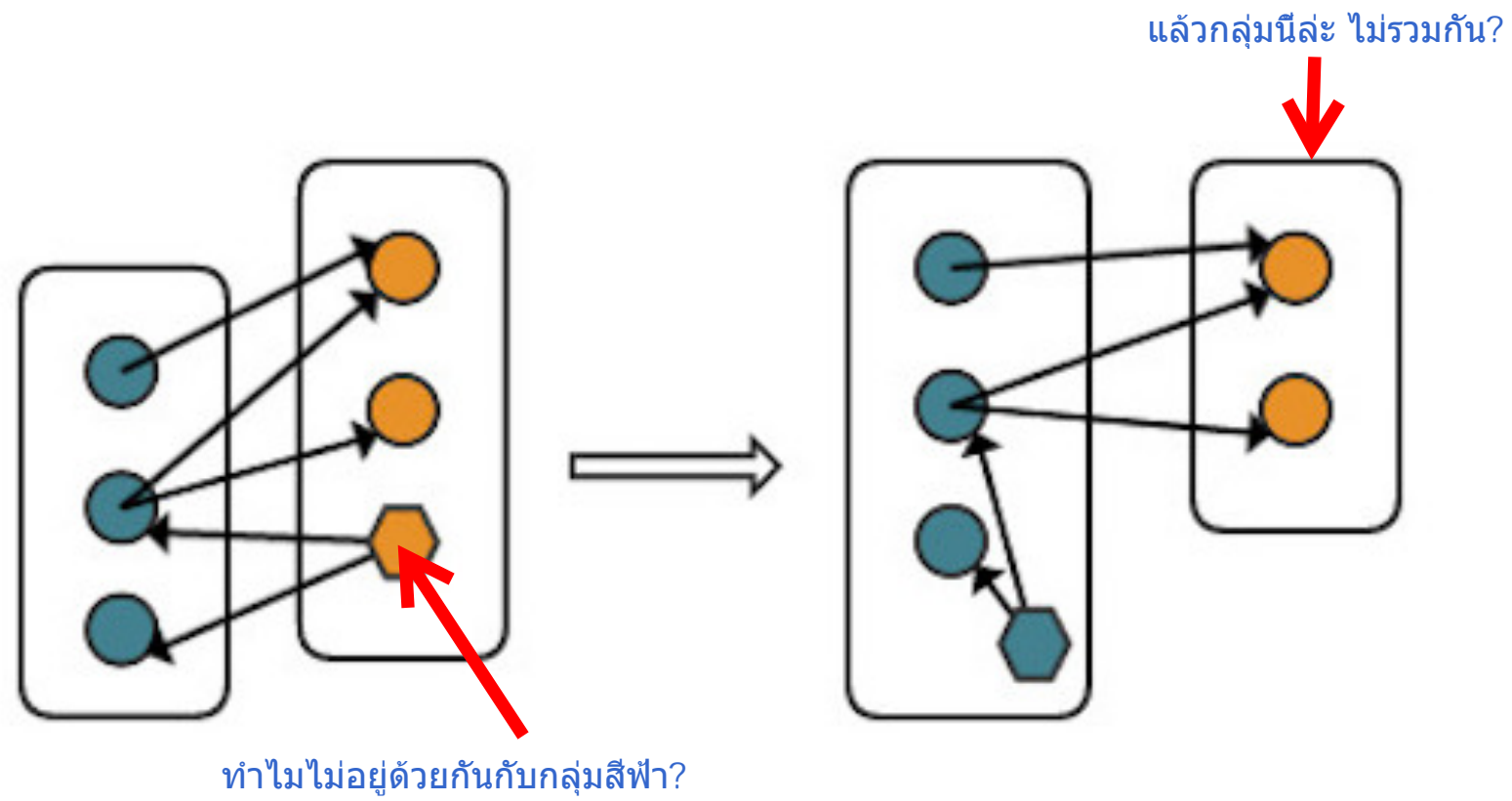


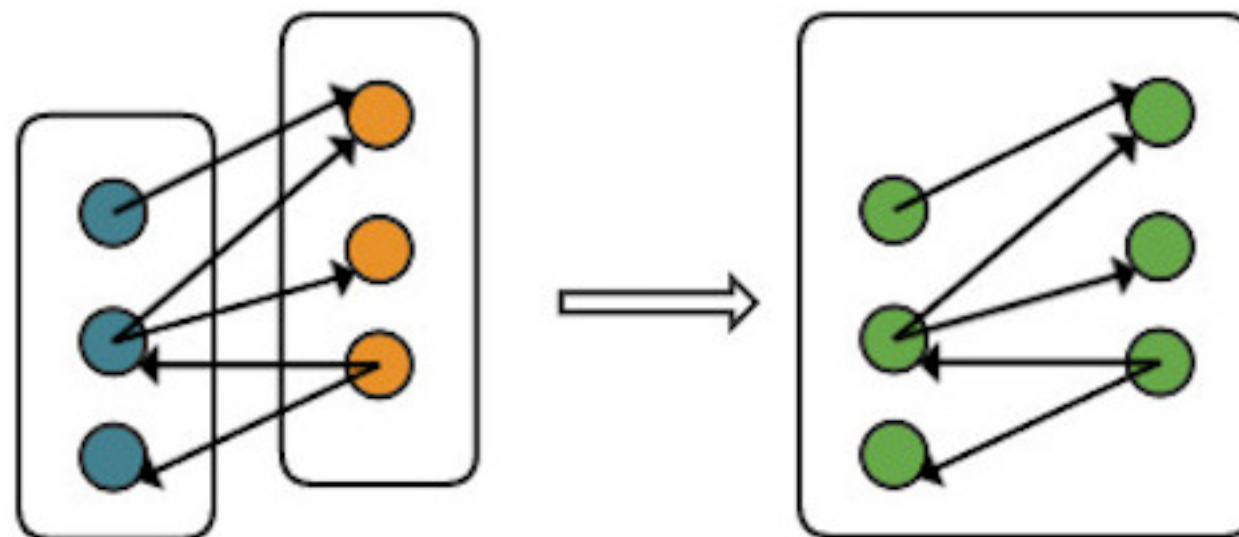


Coupling ?

Cohesion ?







Let's workshop

