**SOA Week 2**

***Microservice*** -> แต่ละบริการแยกกันอย่างชัดเจน ถ้ามีอย่างได้อย่างนึงพังจะไม่พังทั้งระบบ และง่ายต่อการ Deploy โดยแบ่งเป็น service ย่อยๆ ทำให้แต่ละ service รับภาระน้อยลง

***Microservice***

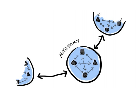
* ต้องเล็ก
* จะ run process เป็นอิสระต่อกัน
* การติดต่อสื่อสารระหว่าง service จะใช้สิ่งที่เรียกว่า REST -> (JSON || XML)
* การ Deploy จะเป็นอิสระต่อกัน

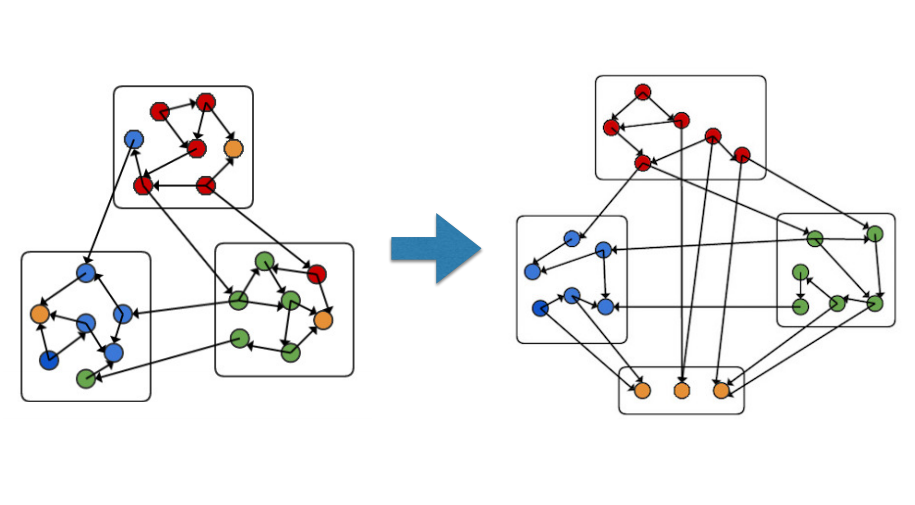
ประโยชน์ ของ Microservice [Scala, Groovy, Clojure, Kotlin]\*\*

* แต่ละ service แยกภาษาที่ใช้เขียน และ database ได้ แต่ต้องใช้ความรู้สูง(Heterogeneity)
* สามารถขยายระบบเพื่อรองรับการใช้งานได้ง่ายขึ้น (Scaling)
* เมื่อระบบได้ระบบหนึ่งตาย จะส่งผลการทบต่อระบบและผู้ใช้น้อยมาก (Resilience)
* การ Deploy
  + 1 เครื่อง หลาย service ทำงานแยก process กัน
  + 1 เครื่อง 1 service –> 1 เครื่องพัง ระบบยังไม่พังทั้หมด
  + 1 เครื่อง 1 service โดย ยัด service ไว้ใน container

การออกแบบ Microservice (ที่ดี)

* Loose coupling ต้องแบ่ง service ให้แยกออกจากกันให้ได้



* High cohesion สิ่งที่ทำงานเหมือนกันควรอยู่ด้วยกัน ถ้าต่างกันควรแยกออกจากกัน