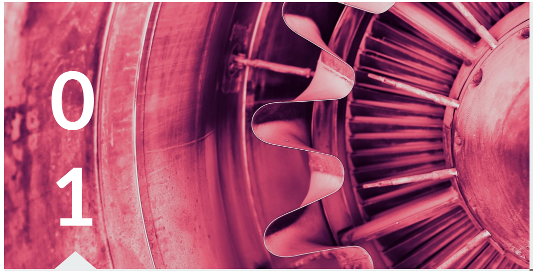
Error Handling in Large Scale Microservice

Wayne Yeh @ 2019-08-23

Outline



01

Recap of Microservice

Microservice at Scale

02

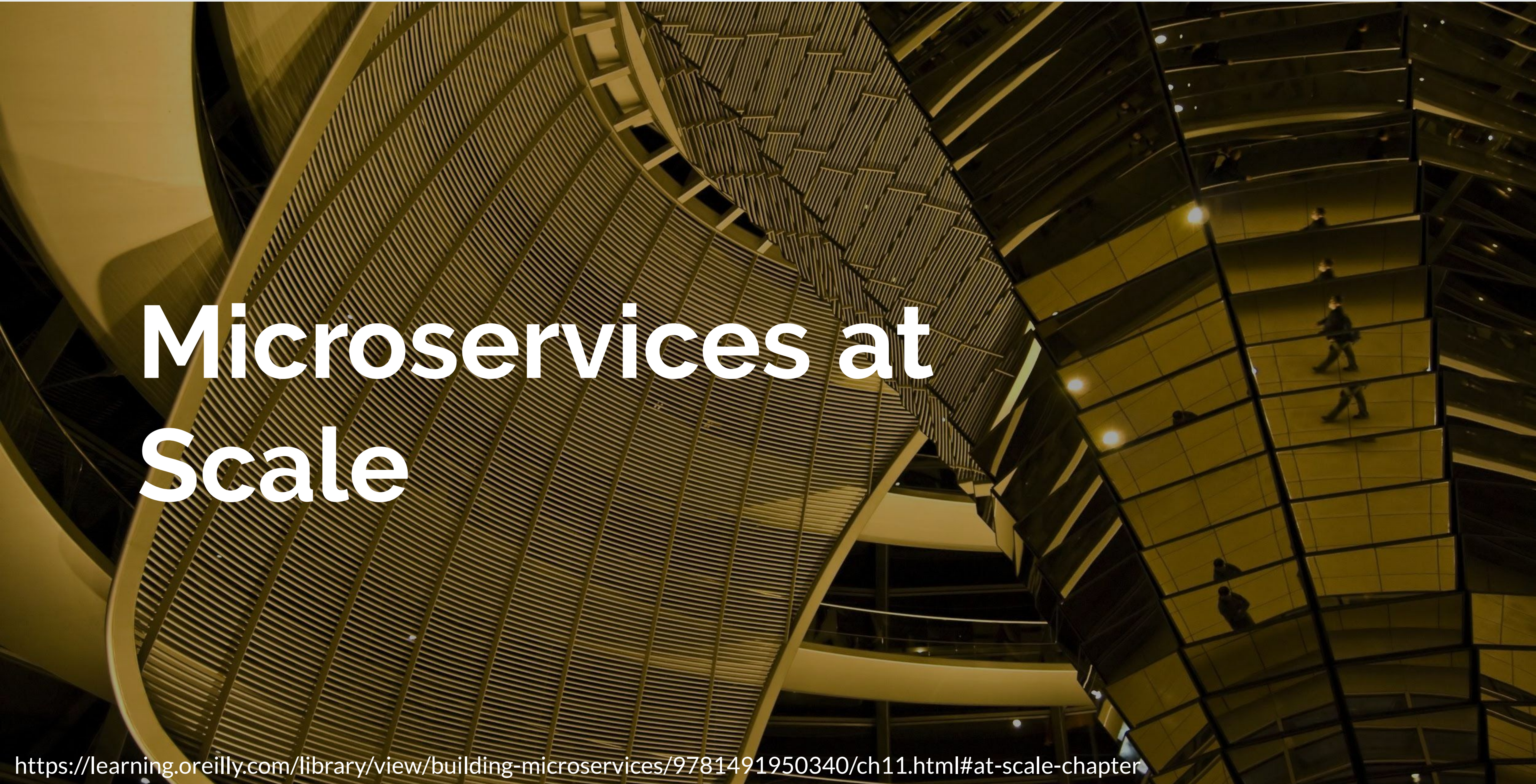
03

Monitoring

Recap of Microservice

Microservices are small, autonomous services that work together.

- Technology Heterogeneity 技術異質性
- Resilience 彈性
- Scaling 擴展
- Ease of Deployment 容易部署
- Organizational Alignment 組織調校
- Composability 組合性
- Optimizing for Replaceability 最佳化可替換性



Microservices at Scale

https://learning.oreilly.com/library/view/building-microservices/9781491950340/ch11.html#at-scale-chapter

Failure is Everywhere

Everything could be failed. We can do our best to try to limit the causes of failure, but at a certain scale, failure becomes inevitable.

Baking in the assumption that everything can and will fail leads you to think differently about how you solve problems.

Degrading Functionality

An essential part of building a resilient system, especially when your functionality is spread over a number of different microservices that may be up or down, is the ability to safely degrade functionality.

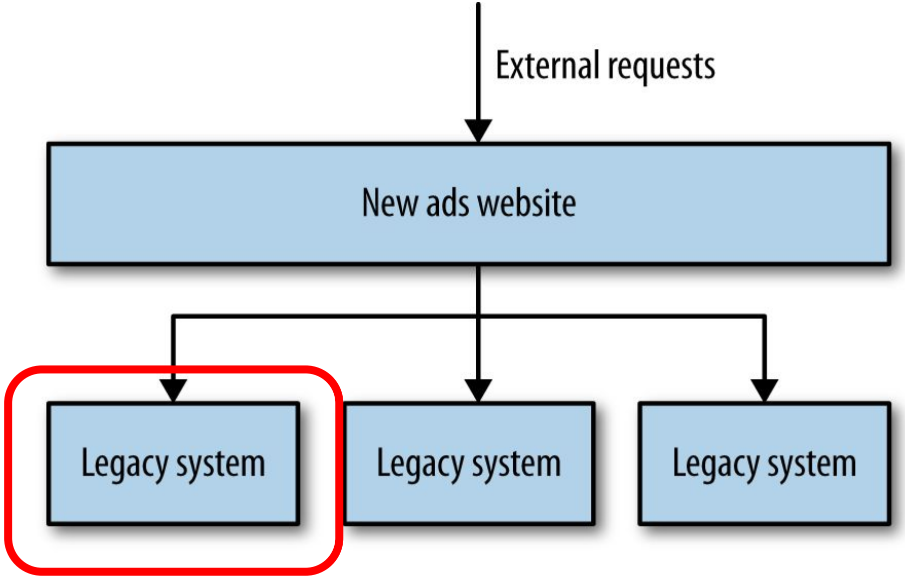
What we need to do is understand the impact of each outage, and work out how to properly degrade functionality.

Type of Failures

- Replacing old system one by one through redirect the requests.
- 6K~7K requests per seconds at peak time.
- But the whole system is crashed one by one.

What’s Happened?

- Slow response to one of the legacy system
- Global HTTP Connection Pool
- Timeout is disabled



Type of Failures (cont.)

What’s the problems?

1. Responding very slowly is one of the worst failure modes you can experience.

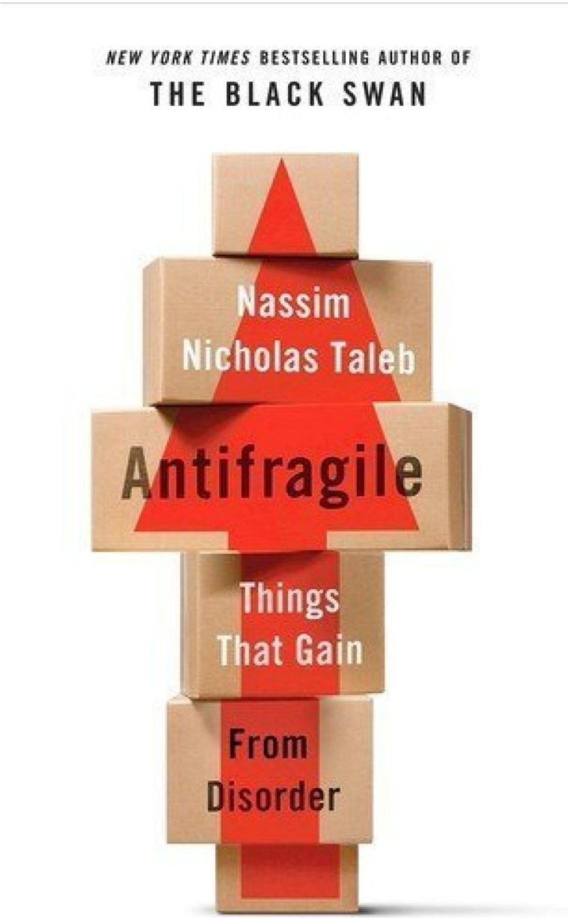
2. Timeout is not enabled.

3. A slow service breaks the whole system by exhausting all of the workers.

4. It cannot be fast-fail.

The Antifragile Organization

- Nassim Nicholas Taleb
  - Fooled by Randomness
  - The Black Swan
  - Antifragile
  - Fragile vs. Robust vs Antifragile.
- Google - Disaster Recovery Test
- Netflix - Chaos Engineering
  - Chaos Monkey


NEW YORK TIMES BESTSELLING AUTHOR OF
THE BLACK SWAN
Nassim Nicholas Taleb
Antifragile
Things That Gain From Disorder

Key concept

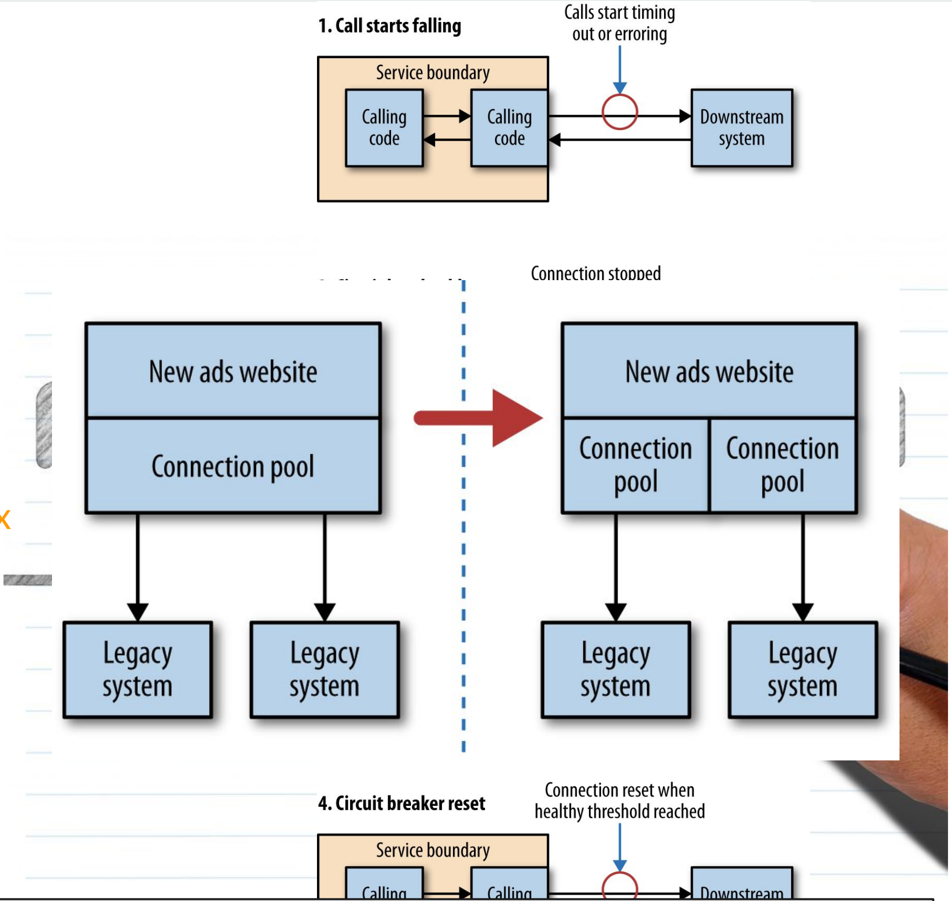
We may not need to do things like Google or Netflix, but we should understand the mindset shift in distributed system. **Things will fail.**

Ways to Handle Failures

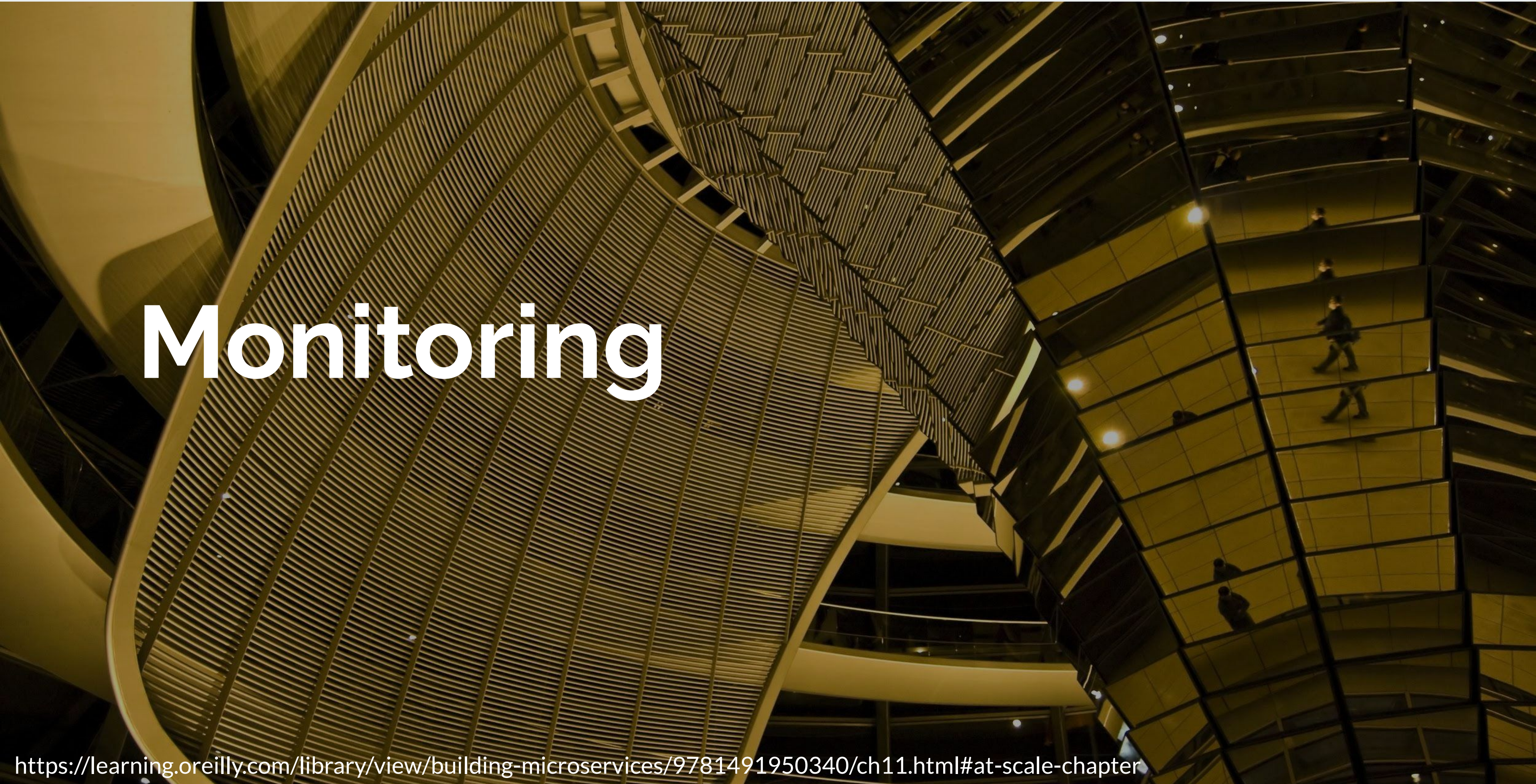
01 | timeout 逾時
02 | circuit breaker 斷路器
03 | bulkhead 隔艙
04 | isolation 隔離

Netflix Hystrix

01 | idempotency 等冪
02 | scale-up/out 擴展
03 | load balance 負載平衡
04 | re-architect 重新架構



you should “design for ~10× growth, but plan to rewrite before ~100×.”



Monitoring

https://learning.oreilly.com/library/view/building-microservices/9781491950340/ch11.html#at-scale-chapter

What should we log?

1. Log everything that may help you to identify what’s going on.
2. Create metrics for statistics.
    - Correlation ID
    - Integration point

Thank you.