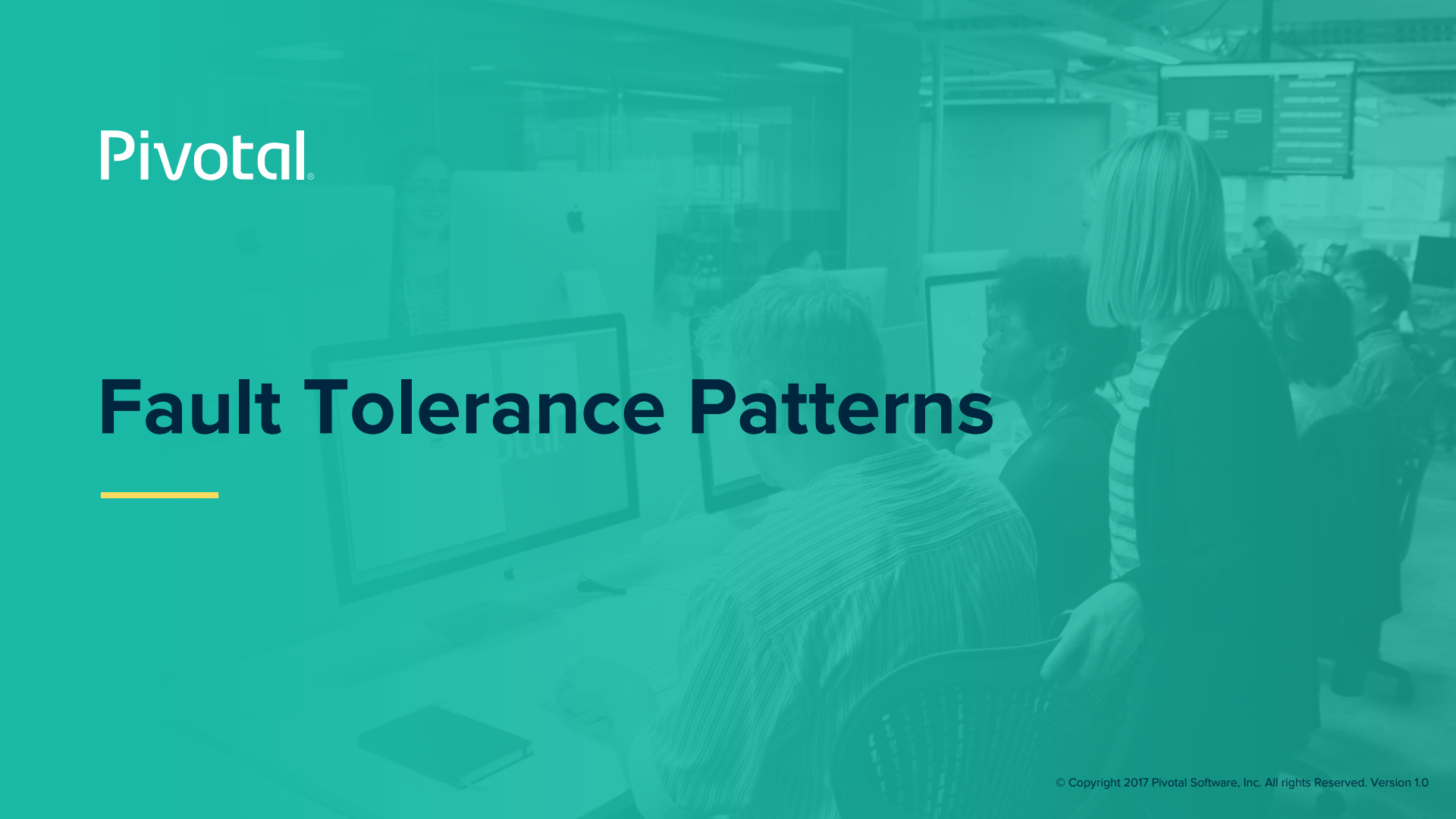
Pivotal

Fault Tolerance Patterns



Motivation

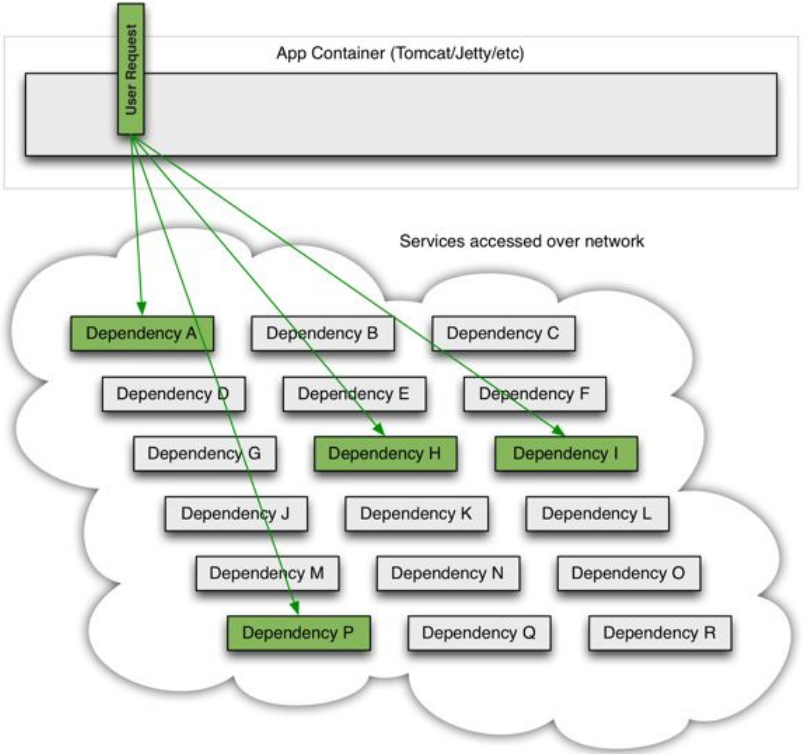
“Applications in complex distributed architectures have dozens of dependencies, each of which will inevitably fail at some point.

If the host application is not isolated from these external failures, it risks being taken down with them.”

- There are many patterns in areas of Fault Tolerance
- We have seen retry in previous labs, but will see several others…

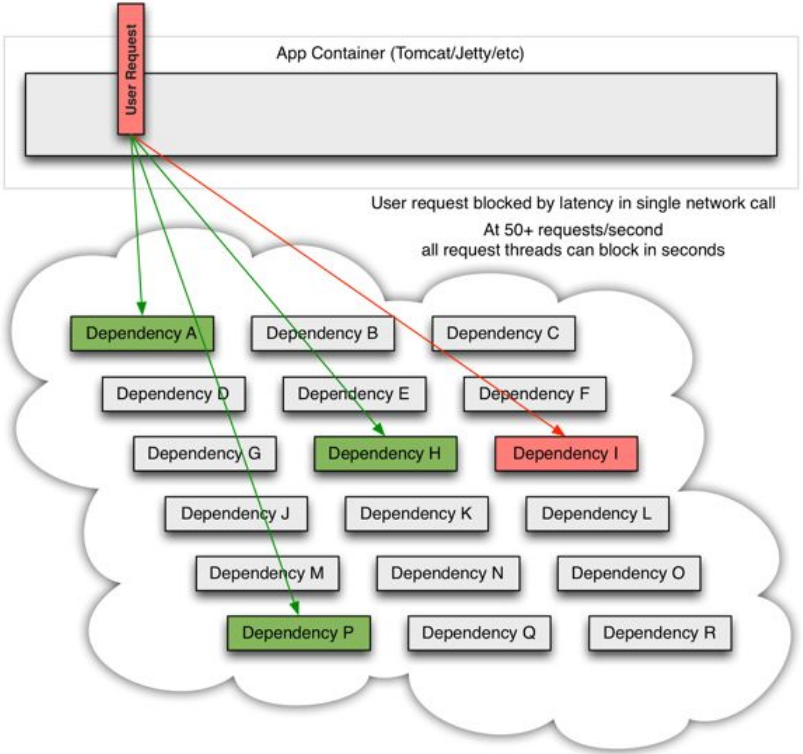
Services Dependency Scenario

- A typical application depending on a number of backing services
- All services are up and behaving normally



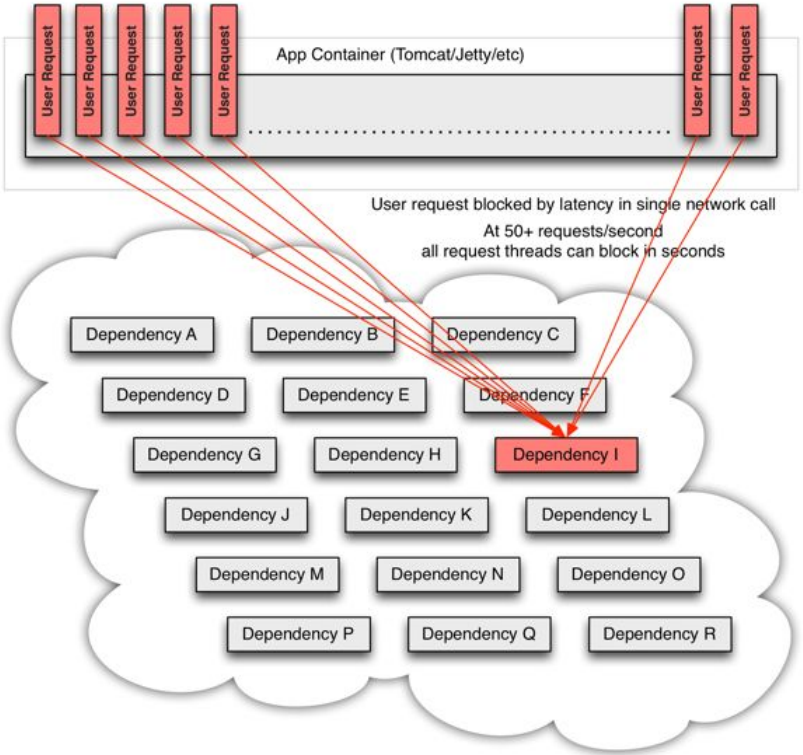
Failing Dependency

- A dependency misbehaves
- Response latency increases, tying up thread in calling application



Failure Cascades to Caller

- Calling application's thread pool is exhausted waiting on misbehaving dependency
- Failure cascades to caller



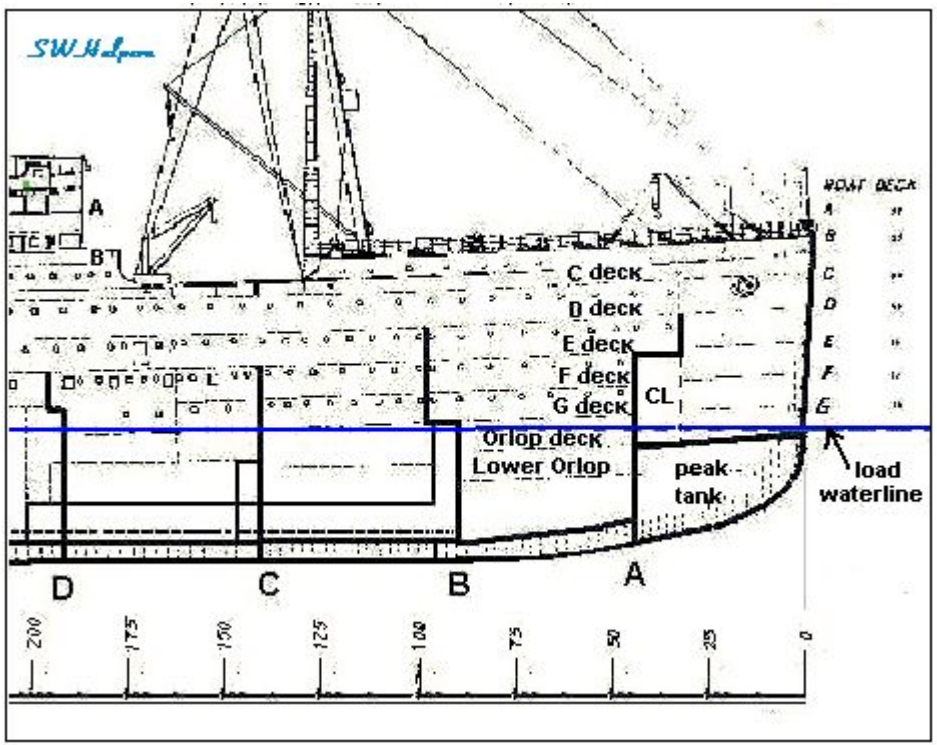
Bulkheads

What?

- Mechanism to isolate parts of a system

Why?

- Isolate points of failure
- Limit scope of failure



Pivotal

Solutions

Hardware, platforms:

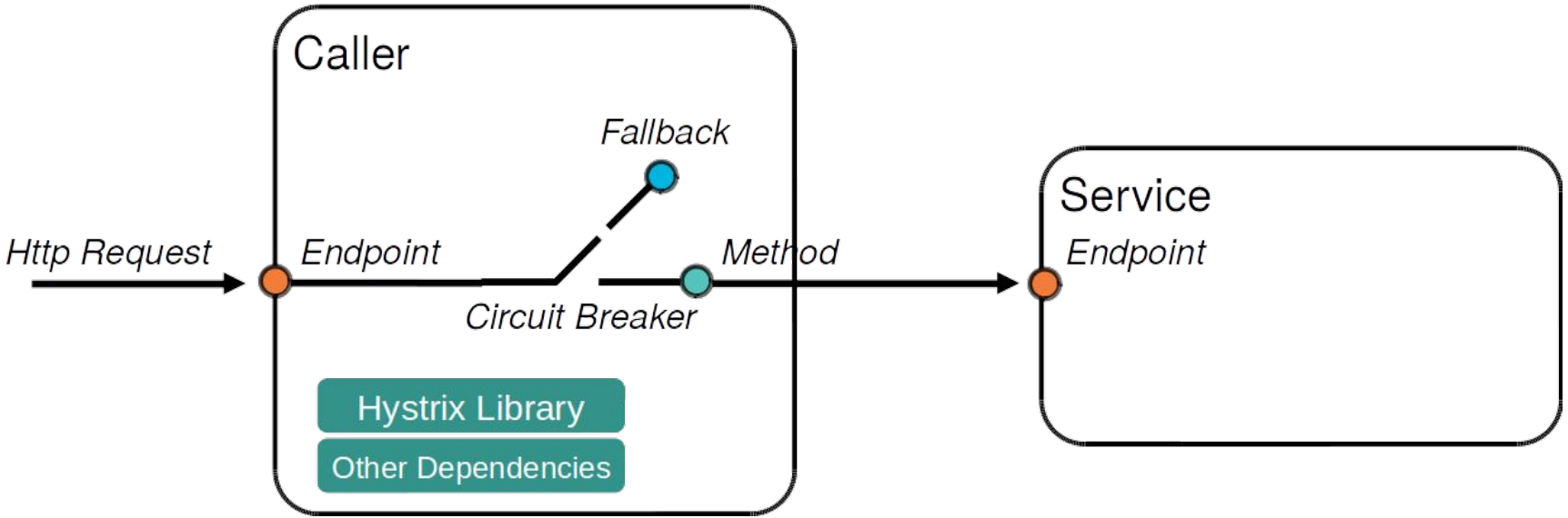
- Resource Pools
  - Geographic: Datacenters
  - Hardware: Racks, Enclosures, Servers, CPU, Core, Threads
  - Virtualization: Hypervisors, Containers
- Network Partitions

Software:

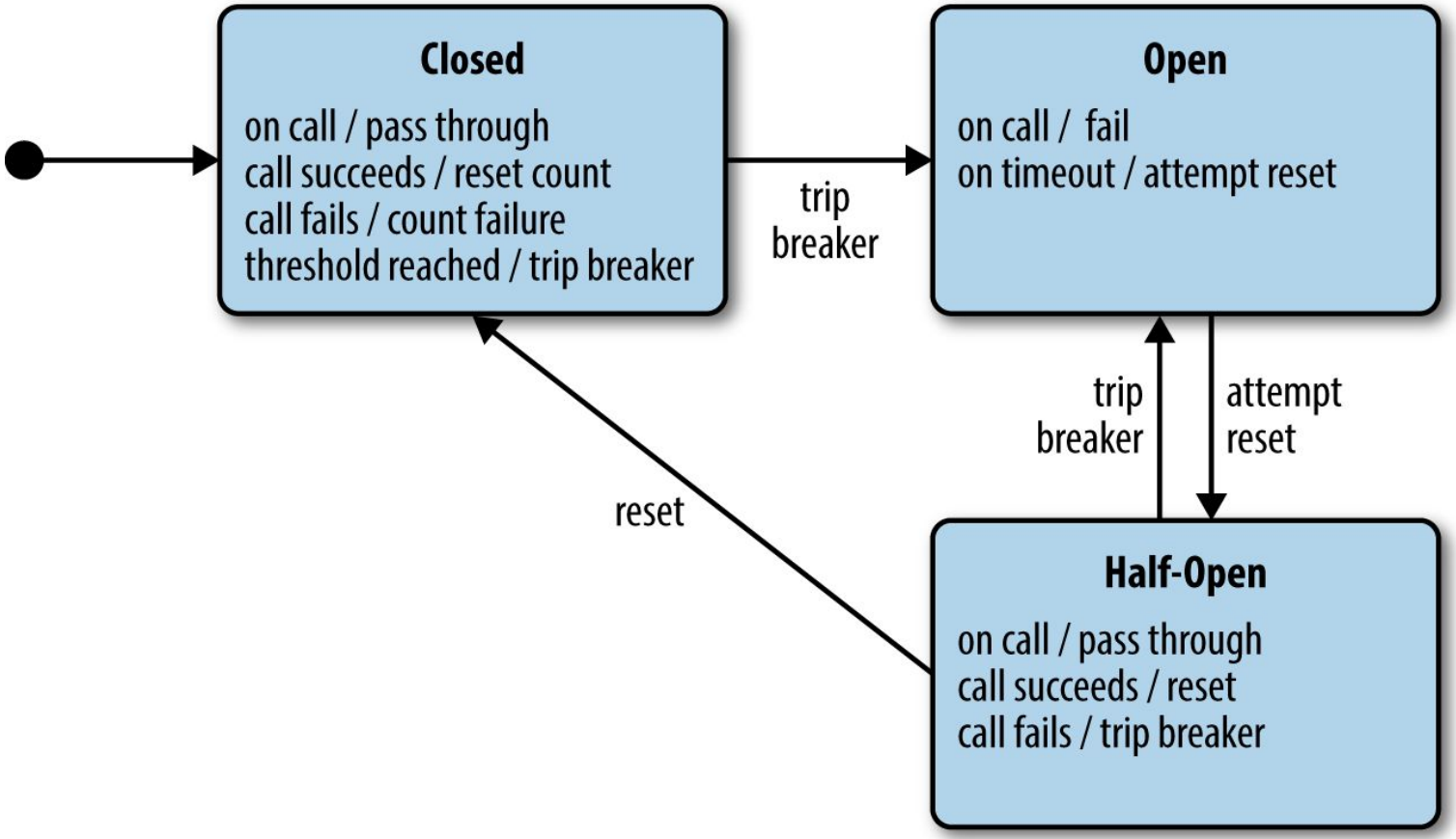
- Timeouts
- Circuit Breakers
- Load Shedding

Circuit Breaker

isolates calls to other services



Circuit Breaker Lifecycle



Resources - Release It!

- V2, 2017
- Michael Nygard
- Covers patterns and case studies backing this unit