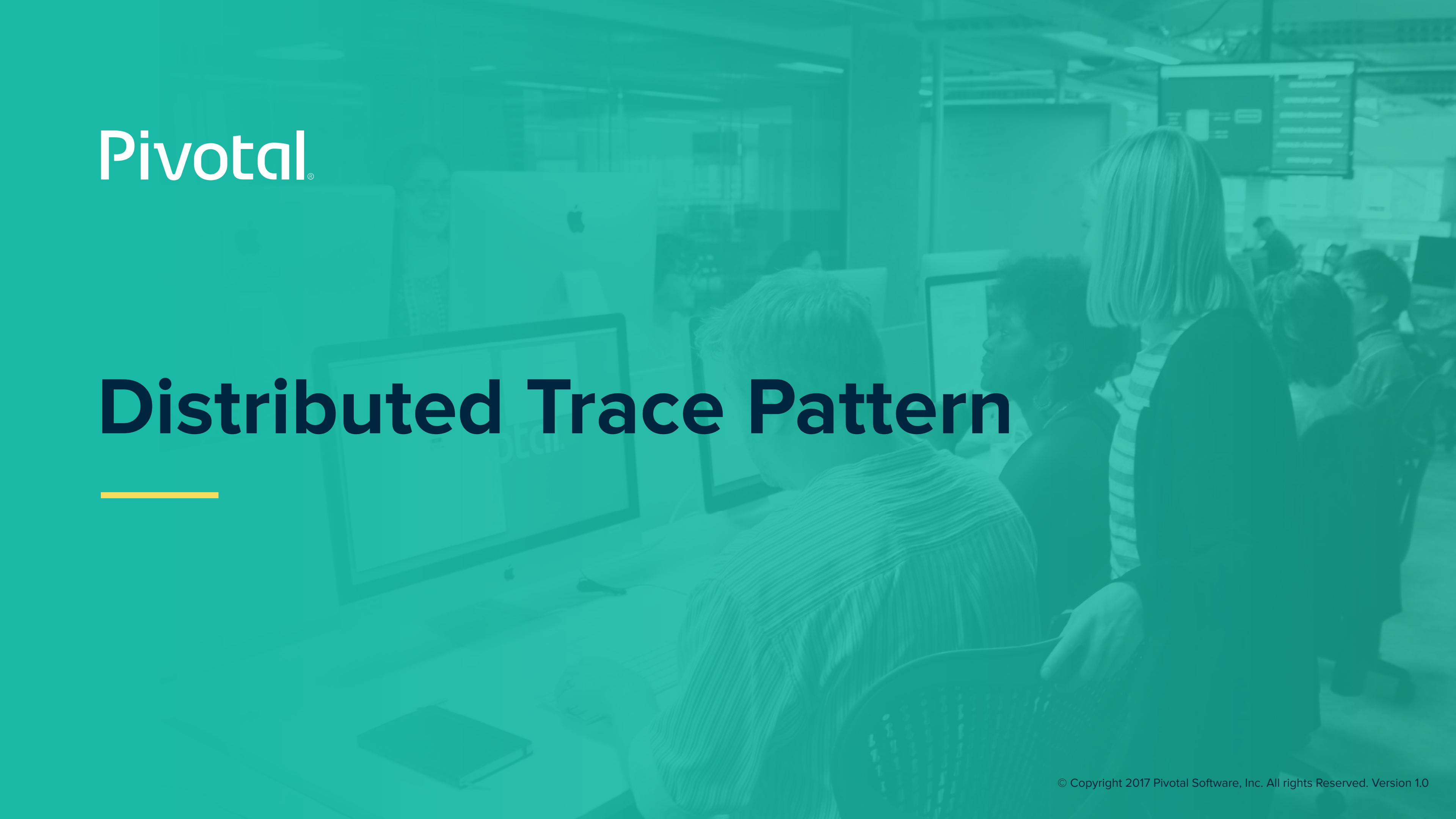


Pivotal

Distributed Trace Pattern

Problems with Distributed Applications

- Distributed applications bring added complexity:
  - Ability to troubleshoot across applications and their downstream dependencies
  - Large number of instances - no longer able to log into individual machines to trace or troubleshoot
- Logs
  - On platforms we have logs as event streams
  - Cloud Foundry has loggregator for log aggregation - solves second problem
  - But what about ability to troubleshoot downstream services?

Telemetry

- Cloud Native architectures scale horizontally
- Many instances
- Problem: How to trace user requests through a service call graph?

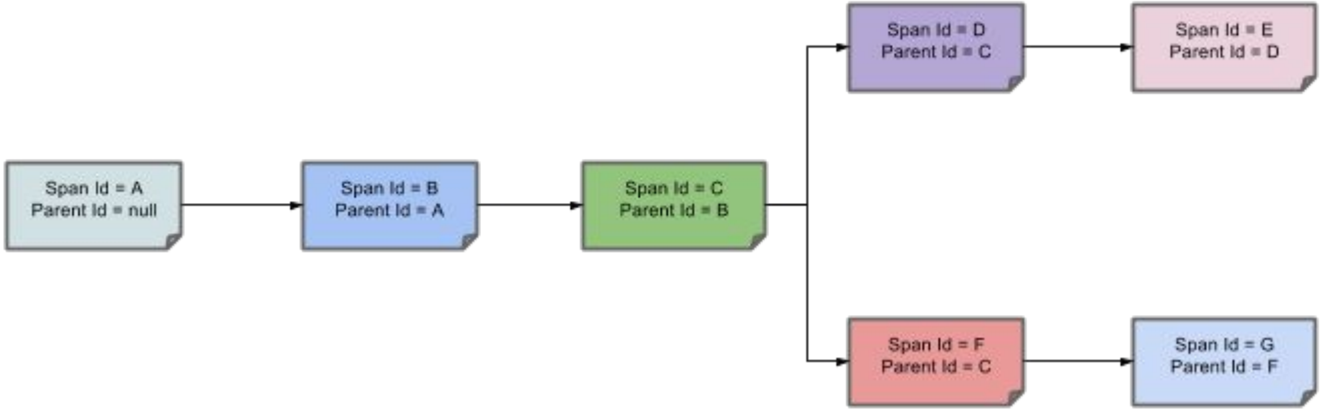
Solutions

- Correlate requests across applications and their runtime dependencies
- Leverage log streams and aggregation
- Visualization Tools
- Effort at standardization - Open Trace API - http://opentracing.io/

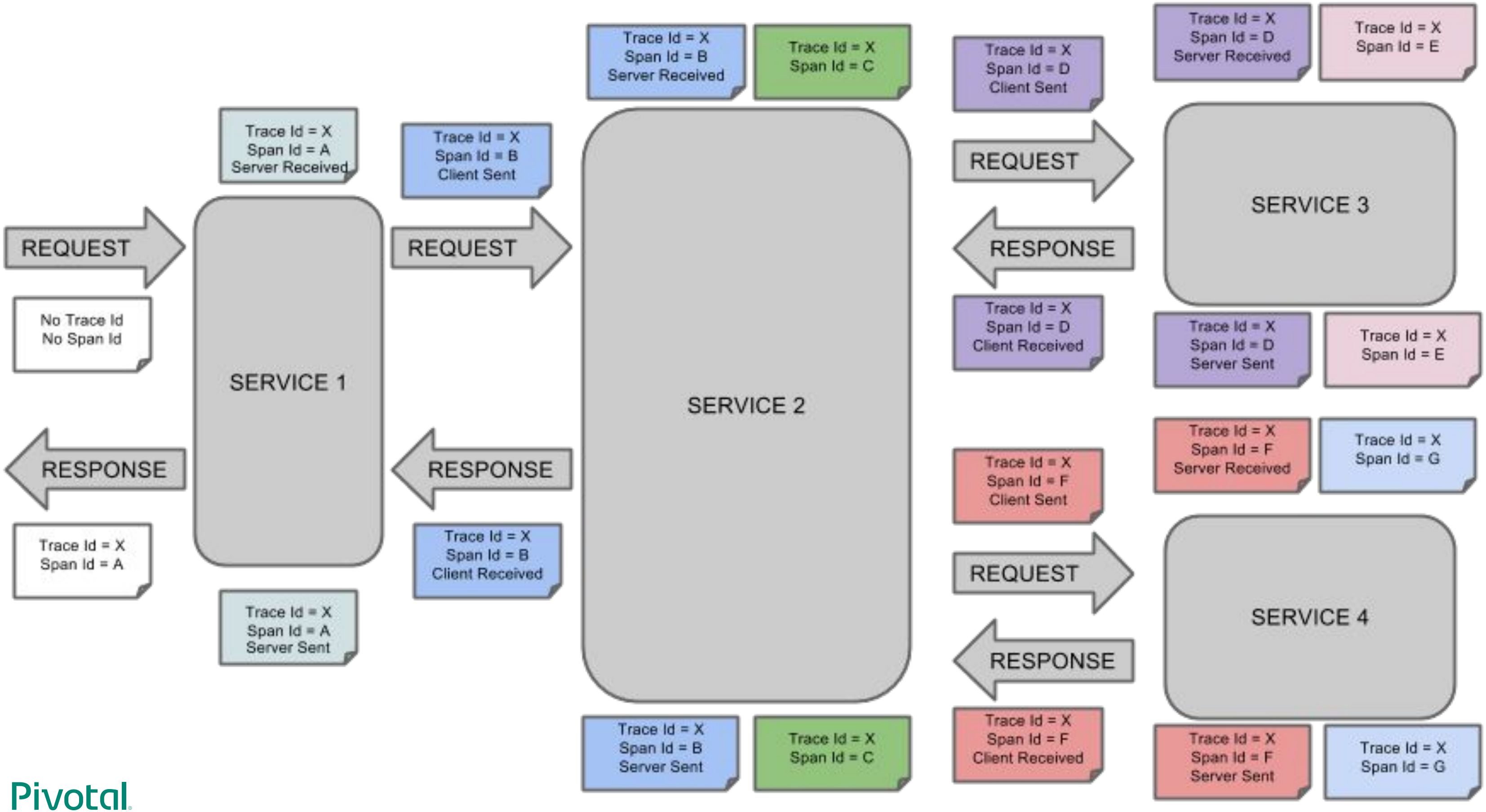
Distribute Trace - Terminology

- **Span**: basic unit of work, in RPC may be correlation of request from initiation to completion
- **Trace**: set of spans comprising a flow, service requests to satisfy a user request

A Trace and its Spans



Distributed Trace - Example



Pivotal