Haystack

Observability and Distributed Tracing for Microservices

Jim Schubert
OpenAPI Tools - Core Contributor
https://openapi-generator.tech/

OBSERVABILITY

- We usually talk about three items of observability:
  - Logging
  - Metrics
  - Monitoring (APM)
- It’s mostly (but not entirely) about DevOps

> “Treat your servers like cattle, not pets.”
>
> –Randy Bias (paraphrased)

- Microservices have pushed us from “monitoring” of individual servers toward “observability” of a system as a whole

- Systems have become more distributed, causing issues to become increasingly more complex to verify or understand

- No longer as simple as:

  - “tail the logs”

  - “run tcpdump”

LOGGING/METRICS

- As systems become more distributed, we've moved toward centralized logging (e.g. Splunk) and metrics (e.g. Grafana) solutions

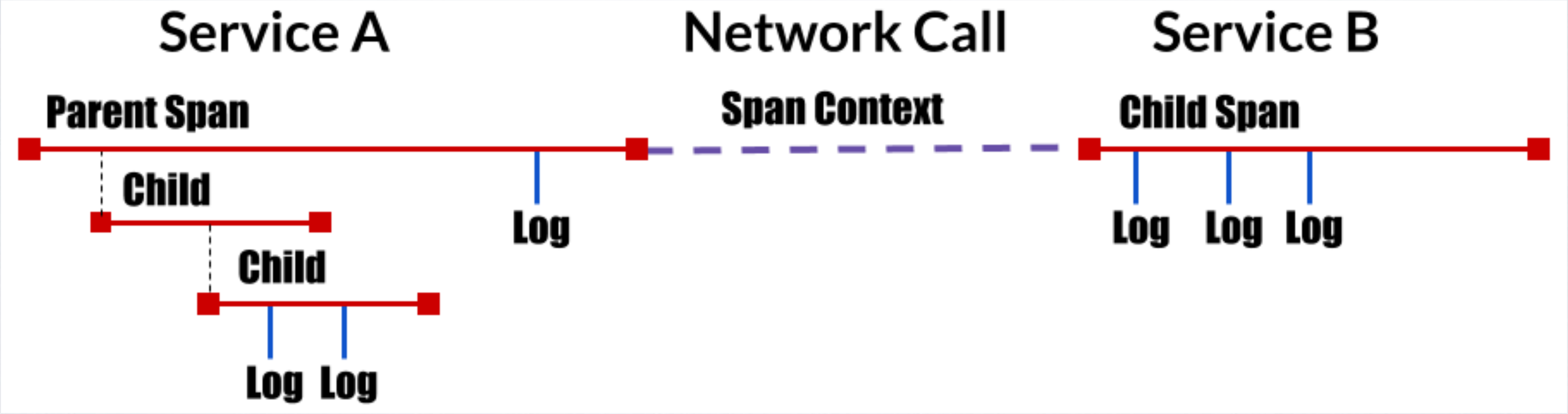
- How do you track overall system health or follow request/response flows when your system has multiple components or dependencies?

- Are all of your services implemented using the same languages and frameworks?

WHAT IS DISTRIBUTED TRACING?

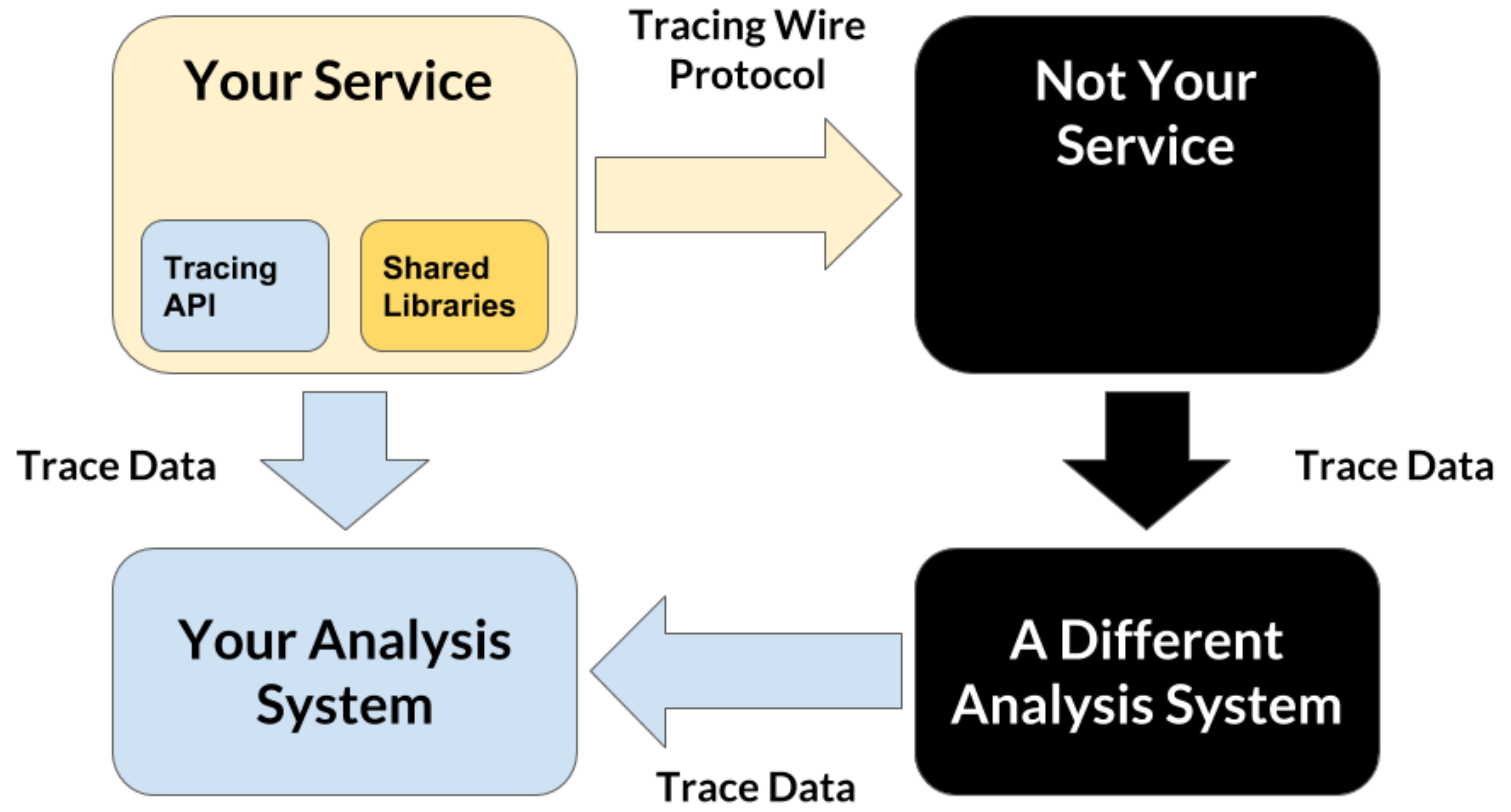
- Collection and presentation of data which allows engineers to:
  - Debug distributed requests
  - Analyze performance characteristics of system internals
  - Find hotspots and/or failures quickly

PARTS OF TRACING



https://opentracing.io/docs/overview/

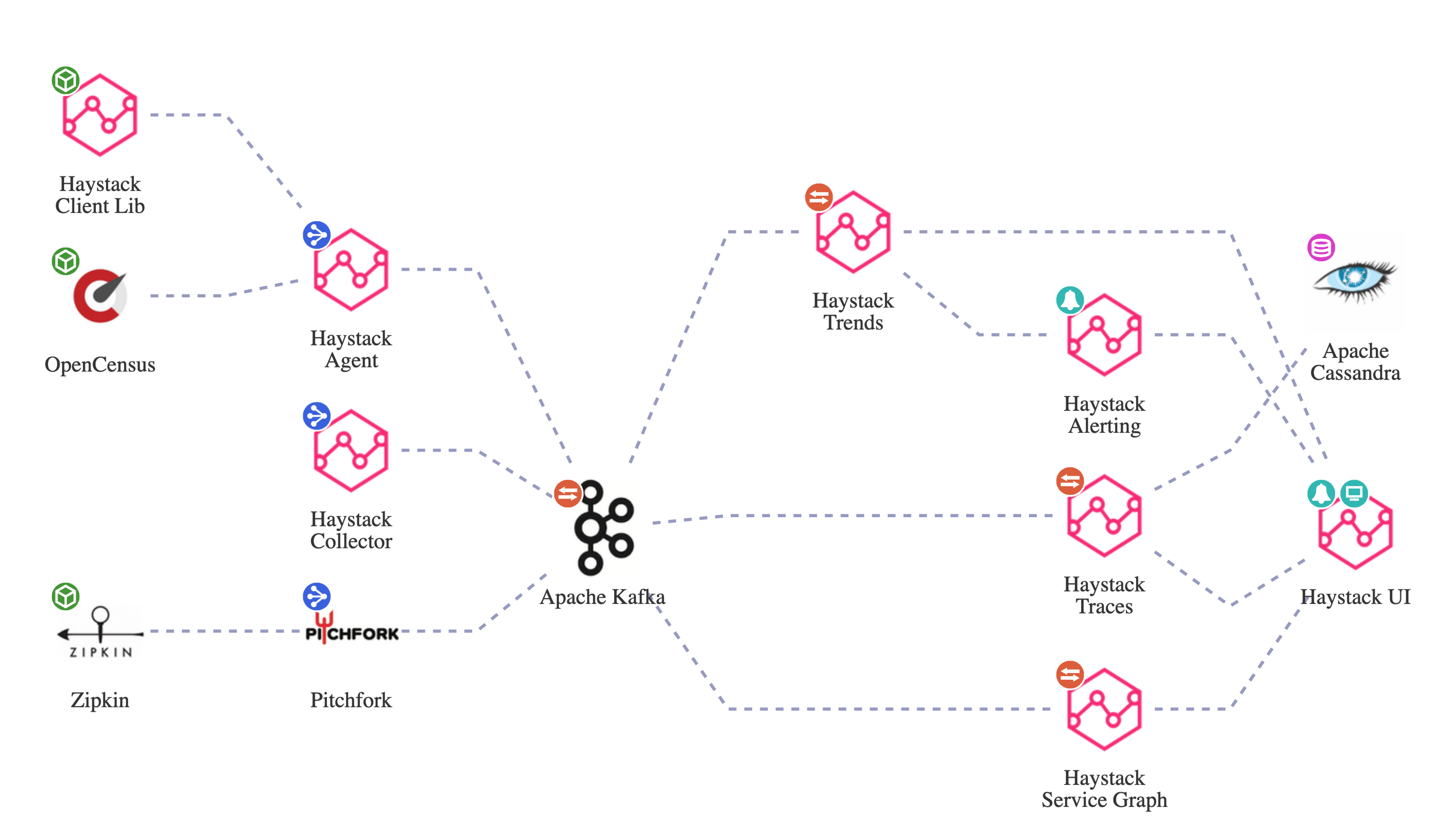
PARTS OF TRACING (2)



https://opentracing.io/docs/overview/

HAYSTACK COMPONENTS

Traces	Based on Dapper, fully OpenTracing compliant. Rebuilds events spanning across multiple micrservices.
Trends	Responsible for reading spans and generating vital service health trends.
Anomaly Detection	Reduce Mean Time To Detect (MTTD) for production incidents; identify anomalies in system health and alert.
Collector	Collects spans from data sources other than those that report to Haystack directly.
Pipes	Packages to send or "pipe" Haystack data to external sinks like AWS Firehose or another Kafka queue.
Service Graph	Visualization of services within a system, showing how requests flow through those services.
UI	Visualized processed data across the various Haystack components.
Agent	Facilitates sending of trace data to Haystack; Spring integration, containerized agent “sidecar”



HOW MIGHT HAYSTACK FIT?

Haystack components are adaptable!

Image created at https://openapm.io

WHAT STANDS OUT?

- Anomaly detection!
  - https://github.com/ExpediaDotCom/adaptive-alerting/wiki
- Pipes!
  - Transactional data warehousing
  - E-commerce long-term reporting and analysis

RUN IT LOCALLY

- https://expediadotcom.github.io/haystack/docs/about/getting_started.html
  - Docker Compose is ready to go!
  - You’ll want to increase Docker daemon memory to 6gb to evaluate all components
  - Also: kubernetes and terraform scripts available

DISTRIBUTED TRACING?

- Haystack
- Zipkin: https://zipkin.io
- Jaeger: https://www.jaegertracing.io/
- New Relic: https://blog.newrelic.com/product-news/distributed-tracing-general-availability/
- OpenCensus/OpenTracing: https://opentelemetry.io/

FURTHER READING

- What is Distributed Tracing?
- Haystack: An Extensible Observability Platform from Expedia
- http://expediadotcom.github.io/haystack
- ExpediaDotCom/opentracing-spring-haystack-starter
- Video: New Relic Complexity in Context (Not Haystack, but great short intro into New Relic's PaaS Distributed Tracing, based upon Opentracing)