

Monitoring Protocol Conformance with Multiparty Session Types and OpenTelemetry (Ongoing Work)

Francisco Ferreira, Nobuko Yoshida, Fangyi Zhou

Imperial College
London

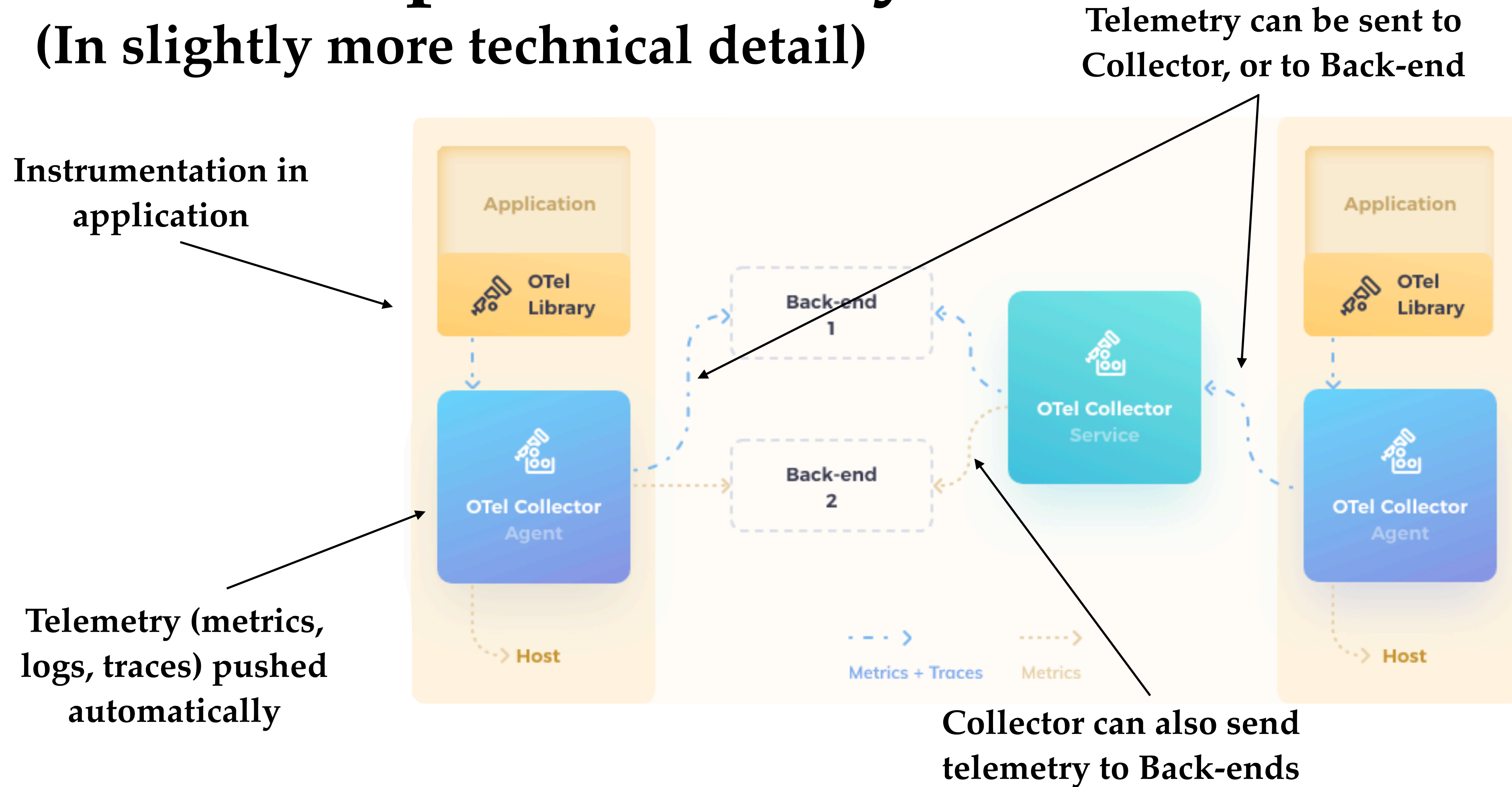
What is OpenTelemetry?

- “An observability framework for cloud-native software”
- Incubating Project of Cloud Native Computing Foundation (CNCF)
- Vendor-agnostic Specification of Telemetry Data
- Supports various languages: Java, Go, JavaScript, Python, Rust, Erlang...
- Supported by Industrial Stakeholders
- Open Source
- <https://opentelemetry.io/>



What is OpenTelemetry?

(In slightly more technical detail)



REFERENCE ARCHITECTURE

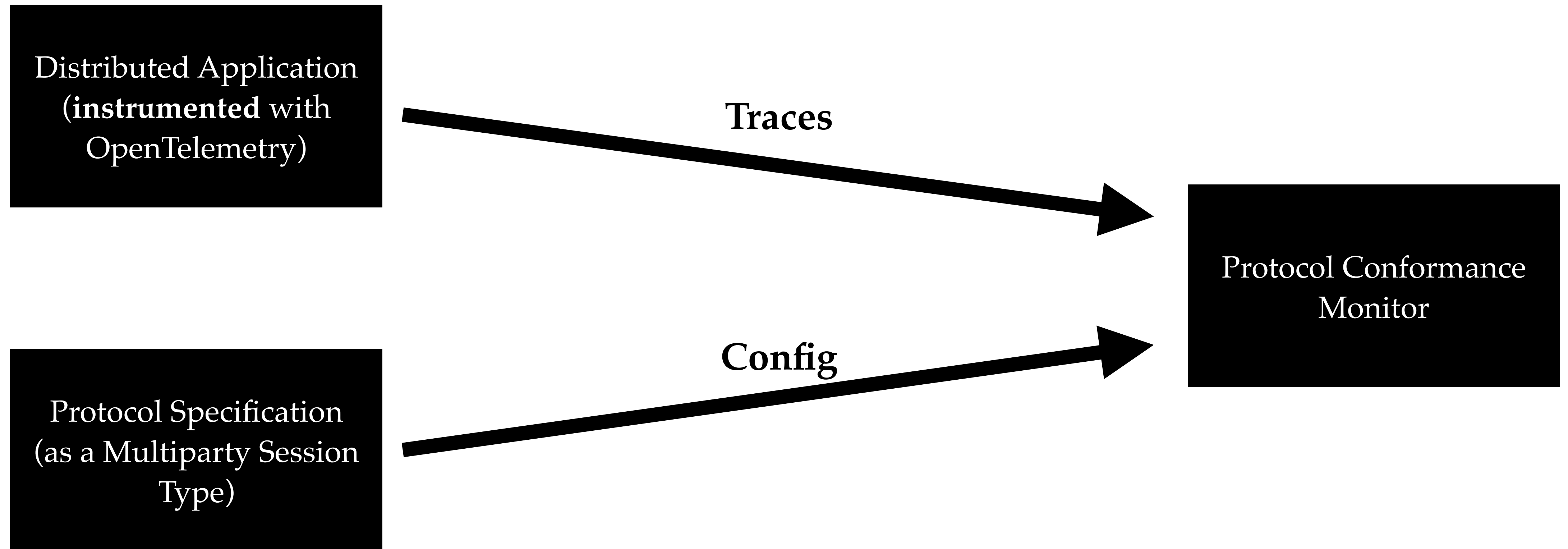
Why are we interested in OpenTelemetry?

- Theory:
 - ‘Central’ monitoring means potentials for more expressive protocols.
- Applicability:
 - OpenTelemetry is a new standard supported by industry.
 - Example use cases include **cloud microservices**.
- Other: Industrial connection with Red Hat.

Goal of Our Work

- Monitor protocol conformance for existing distributed system
 - Monitor process behaviour via **tracing** (using OpenTelemetry)
 - “Traces track the progression of a single request, as it is handled by services that make up an application.”
 - **Instrument** existing applications, instead of asking developers to use generated APIs
 - Possibly via automatic instrumentation provided by OpenTelemetry

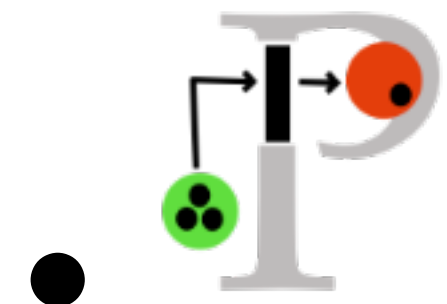
Workflow



A Short Demo

Next Steps...

- Develop a more expressive semantic model for global protocols



- Pedro: Petri-net Semantics for MPST
- Integrate with existing distributed systems
 - First Step: a mini Uber example used as a tutorial for distributed tracing

Thank you!

<https://github.com/fangyi-zhou/mpst-tracing>