**Open source tools for Application monitoring   
  
  
1. OpenTelemetry**:

* **Description**: OpenTelemetry provides a set of APIs, libraries, agents, and instrumentation to capture distributed traces and metrics from applications.
* **Features**: It is a unified standard for tracing and metrics, supporting various languages and backends.

**2. Jaeger**:

* **Description**: Jaeger is an open-source end-to-end distributed tracing tool that helps monitor and troubleshoot micro services-based distributed systems.
* **Features**: Distributed context propagation, transaction monitoring, root cause analysis, and service dependency analysis.

3. **Zipkin**:

* **Description**: Zipkin is a distributed tracing system that helps gather timing data needed to troubleshoot latency problems in service architectures.
* **Features**: Collects timing data, provides trace and dependency views, and supports various storage backends.

4. **Apache SkyWalking**:

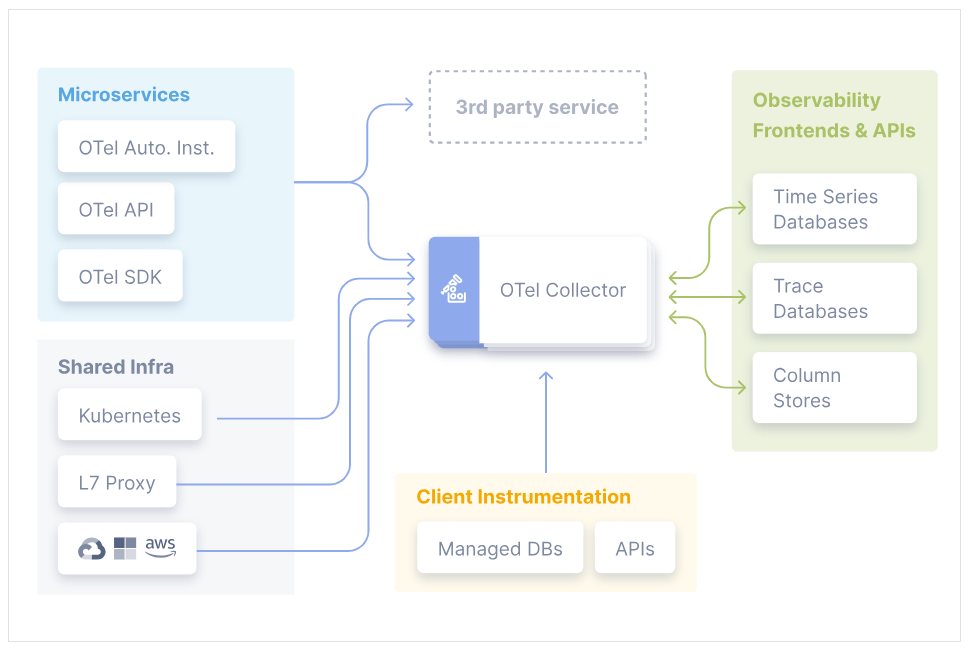
* **Description**: Apache SkyWalking is an open-source APM tool for observing and analyzing distributed systems, especially designed for micro services, cloud native, and container-based architectures.
* **Features**: Distributed tracing, service mesh observability, and performance monitoring.

5. **Elastic APM**:

* **Description**: Part of the Elastic Stack, Elastic APM is an open-source application performance monitoring system built on the Elastic Stack.
* **Features**: Real-time application performance monitoring, error tracking, and detailed transaction breakdowns.

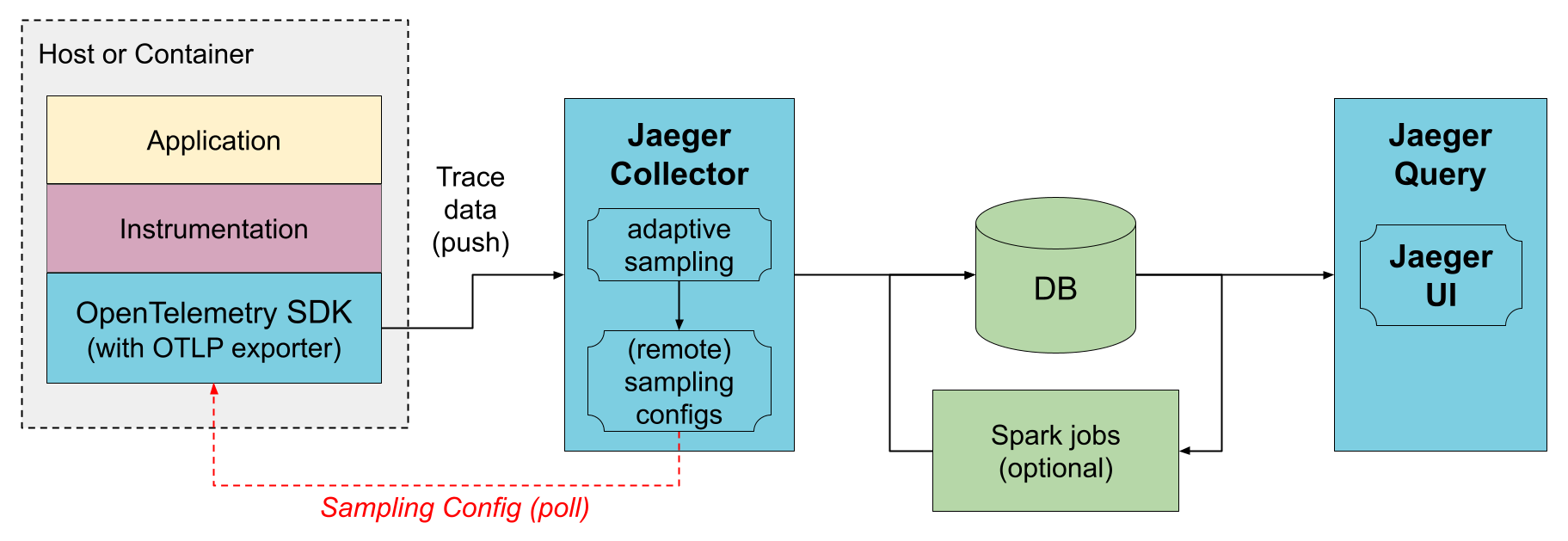
1. **Open Telemetry**:

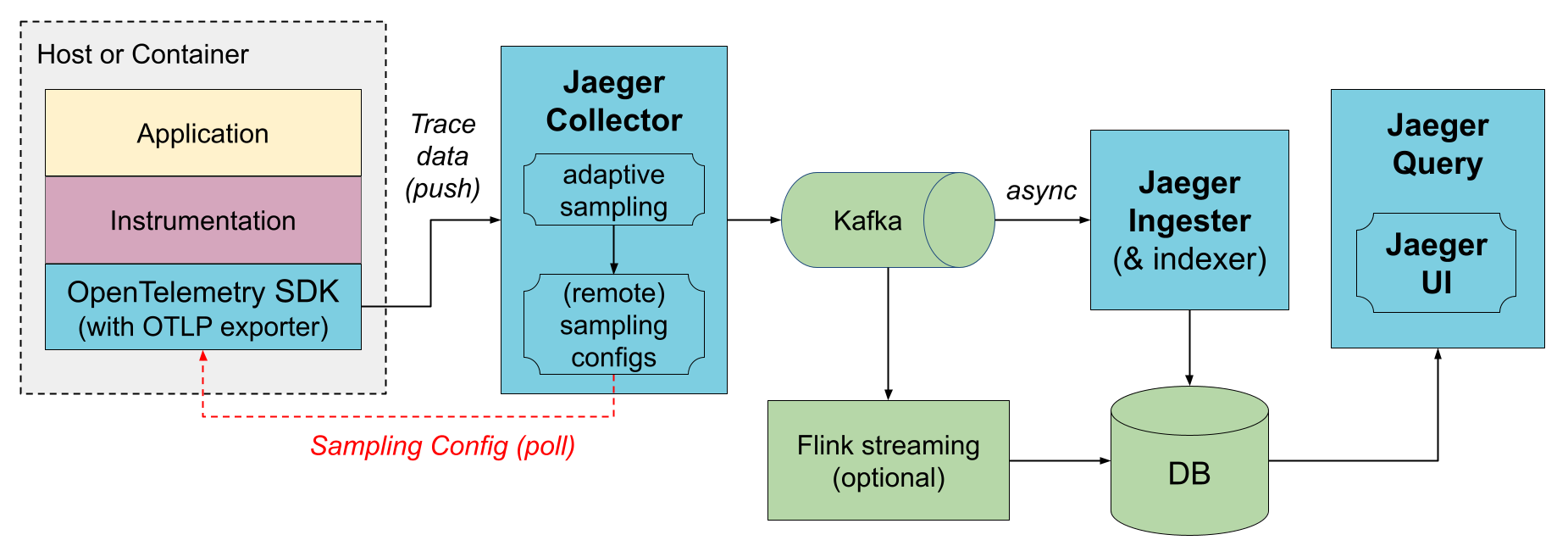
* Open Telemetry, also known as OTel, is a vendor-neutral open source [Observability](https://opentelemetry.io/docs/concepts/observability-primer/#what-is-observability) framework for instrumenting, generating, collecting, and exporting telemetry data such as [traces](https://opentelemetry.io/docs/concepts/signals/traces/), [metrics](https://opentelemetry.io/docs/concepts/signals/metrics/), and [logs](https://opentelemetry.io/docs/concepts/signals/logs/).
* As an industry-standard, OpenTelemetry is [supported by more than 40 observability vendors](https://opentelemetry.io/ecosystem/vendors/), integrated by many [libraries, services, and apps](https://opentelemetry.io/ecosystem/integrations/), and adopted by [numerous end users](https://opentelemetry.io/ecosystem/adopters/).



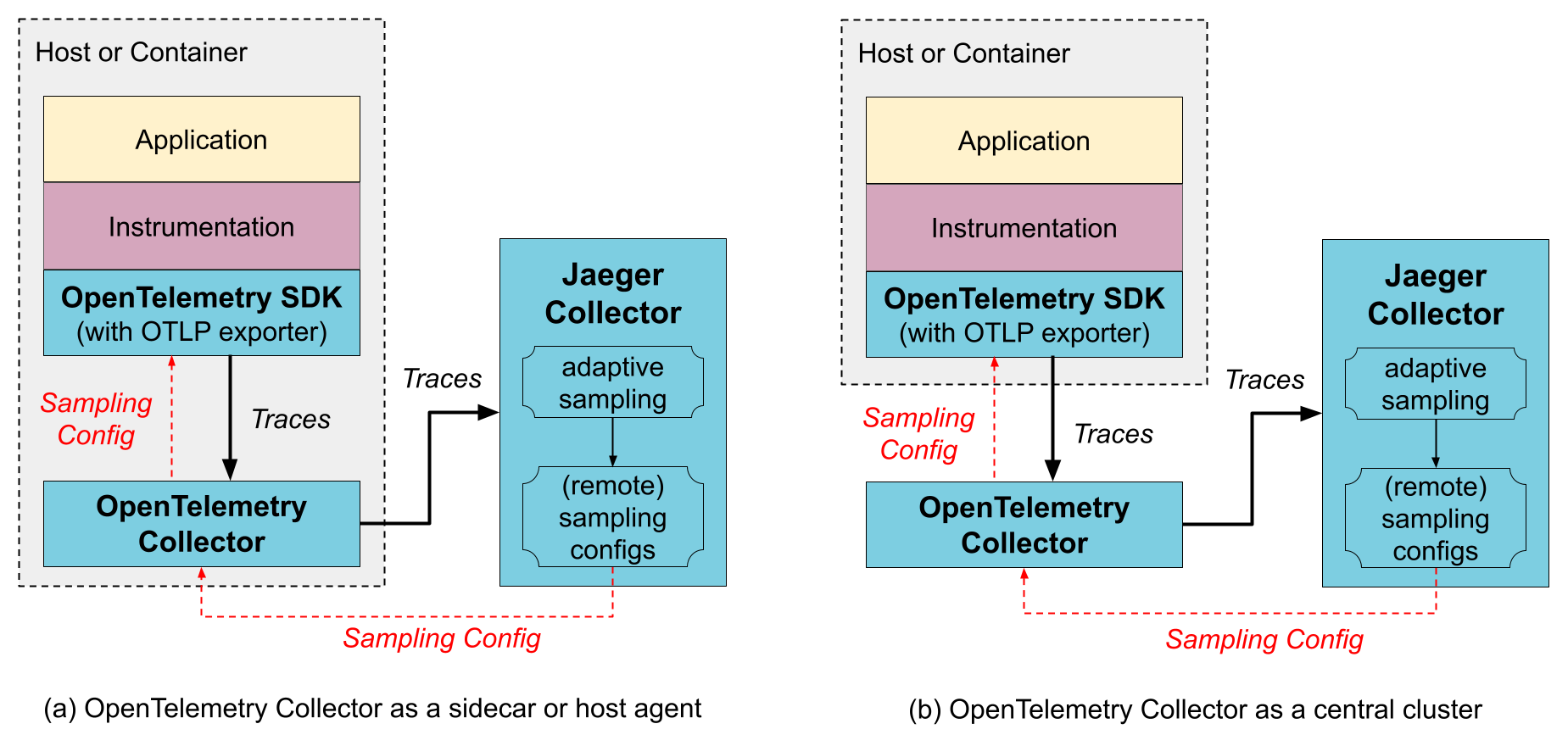
* Reference link: <https://opentelemetry.io/docs/>

**2. Jaeger:**

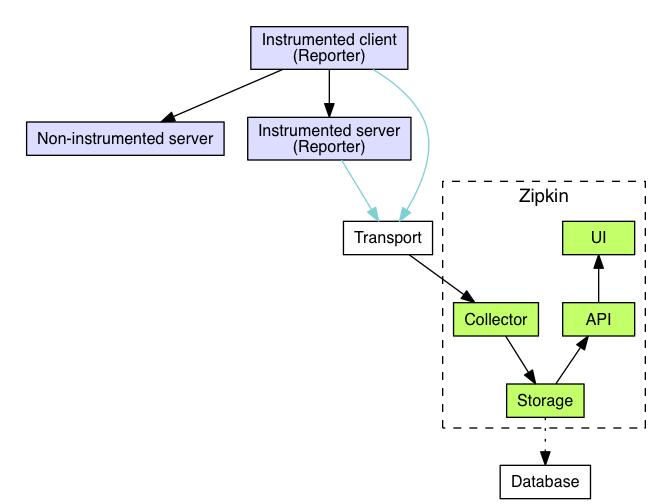
* Jaeger can be deployed either as an **all-in-one** binary, where all Jaeger backend components run in a single process, or as a scalable distributed system. There are two main deployment options discussed below.  
    
  1. Direct Storage:  
  

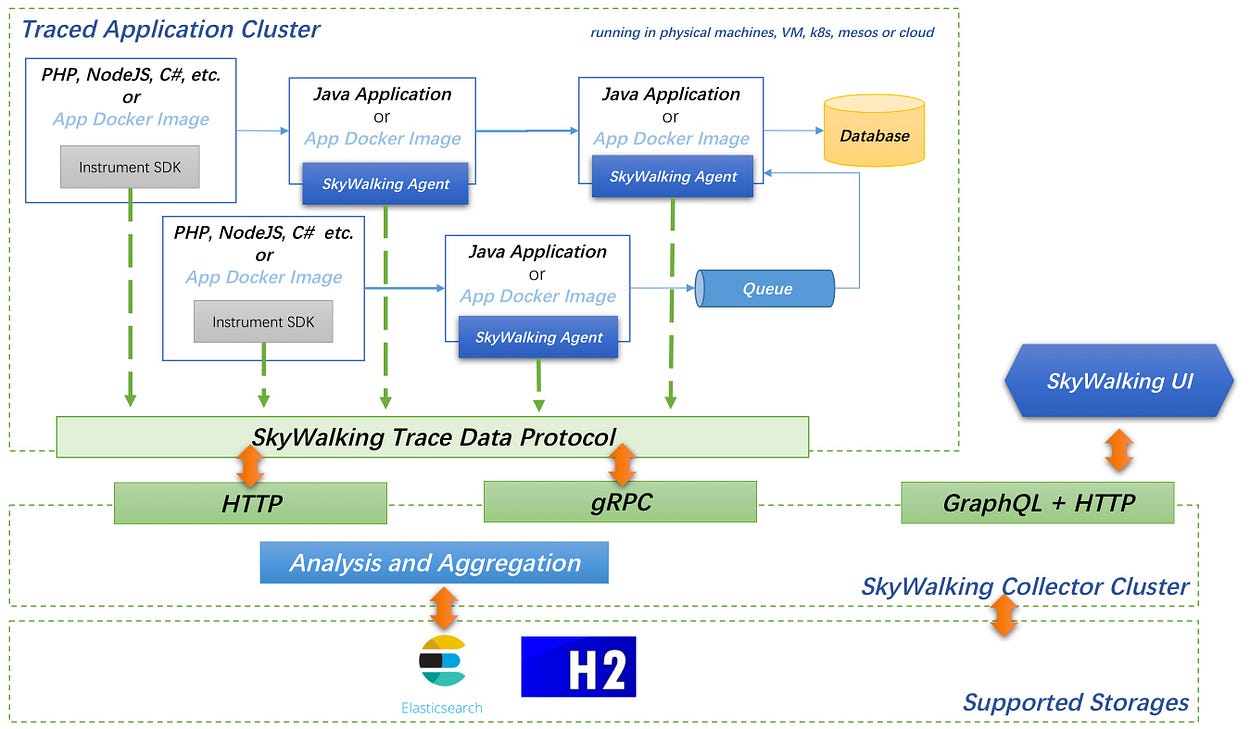
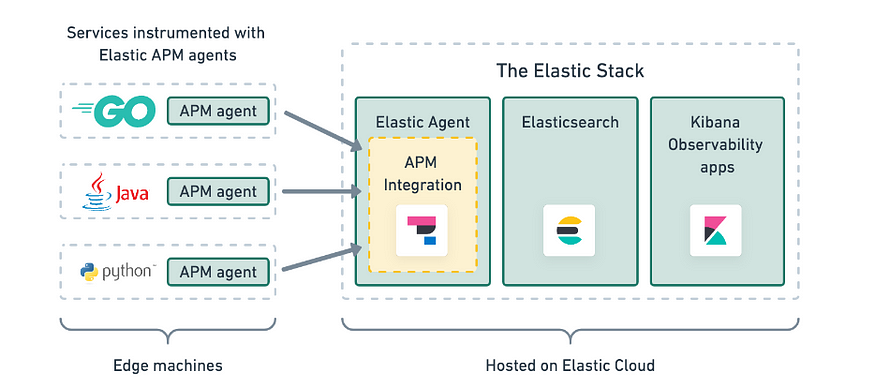
2. via Kafka   


Reference link: <https://www.jaegertracing.io/docs/1.59/architecture/>

**Jaeger with open telemetry collector:**   


**3. Zipkin:**Tracers live in your applications and record timing and metadata about operations that took place. They often instrument libraries, so that their use is transparent to users. For example, an instrumented web server records when it received a request and when it sent a response. The trace data collected is called a Span.



Reference link: <https://zipkin.io/pages/architecture.html>  
  
4. **Apache SkyWalking**  
  
  
Reference Link: <https://medium.com/@AsfSkyWalking/apache-skywalking-architecture-designs-a0d3a19409ff>   
  
5. **Elastic APM**:  


Reference Link: <https://medium.com/@sadhamhussian2020/application-monitoring-in-golang-application-with-elastic-apm-220899261e46>