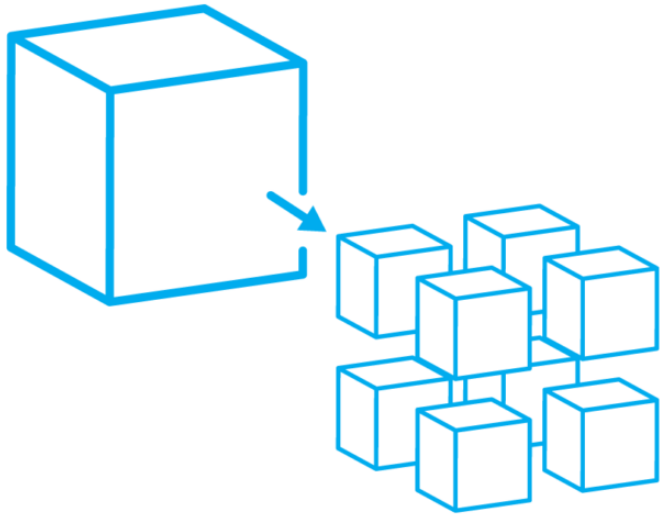


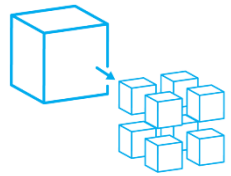
Microservicios – Arquitectura y Desarrollo

Por: Carlos Carreño

ccarrenovi@Gmail.com

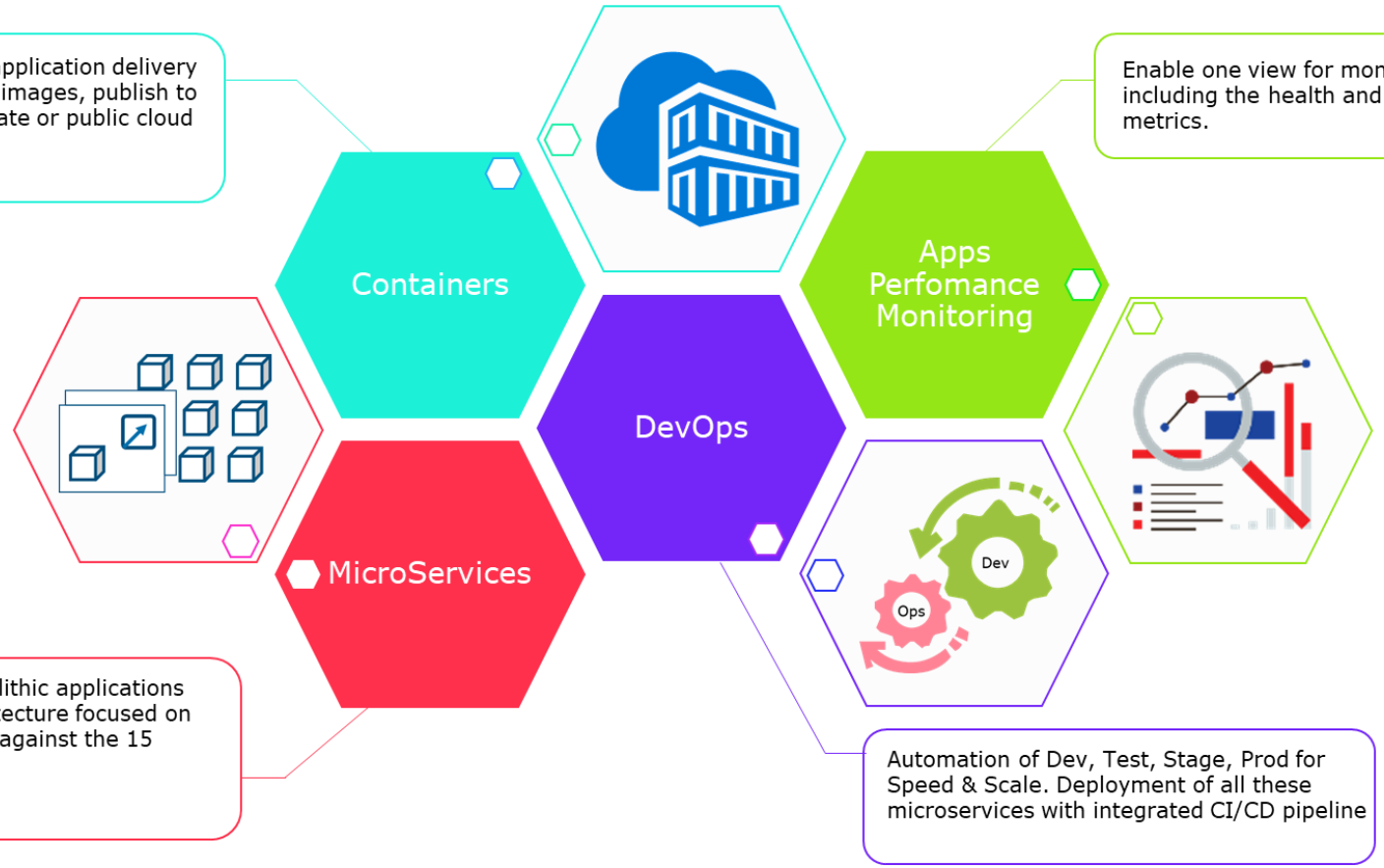
Noviembre, 2020

MÓDULO 9: Monitoreando Microservicios usando OpenZipkin, Spring Cloud Sleuth y Kibana Dashboard

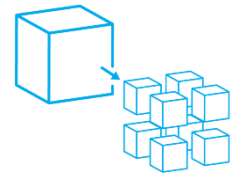


- Métricas y HealthChecks
- Trazas distribuidas
- Logs de aplicaciones y agregación de logs
- Monitoreo usando Dashboards

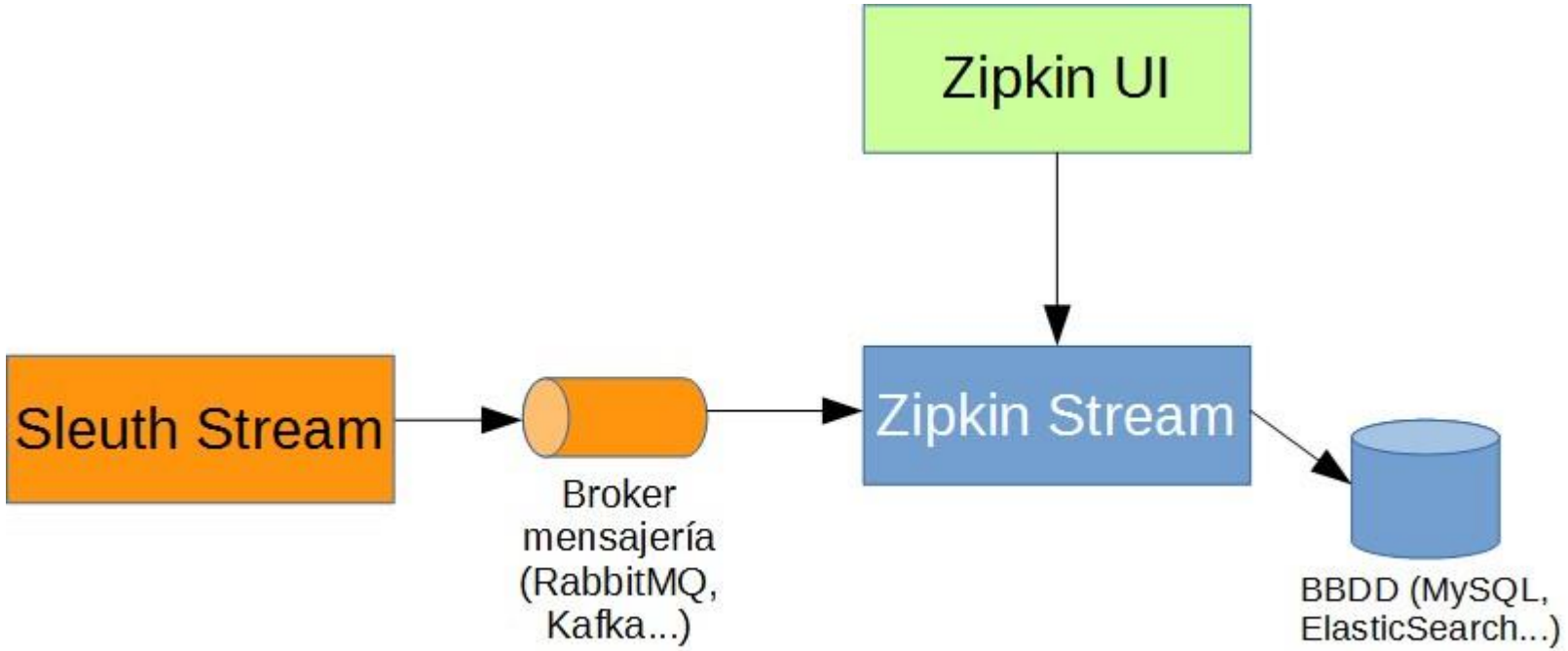
Métricas y HealthChecks



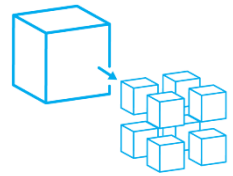
Trazas distribuidas: Arquitectura de la Trazabilidad



- La siguiente arquitectura de implementación se puede aplicar en Spring Cloud



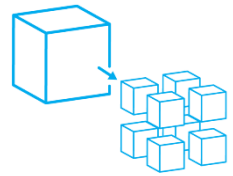
Logs de aplicaciones y agregación de logs: Spring Cloud Sleuth



- Sleuth es una librería que permite el trazado distribuido.
- Mecanismo automático de identificación de peticiones.
- TraceID y SpanID

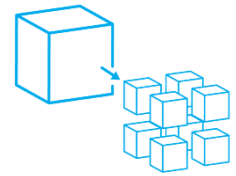
```
2016-11-15 15:12:25.977  INFO [sleuth-client,5501a4bacd8faaa8,44d49402e8b56e98,false] 9544 --- [ni
o-8999-exec-4] c.p.microservices.SleuthApplication      : Handling home
```

Sleuth Terminología

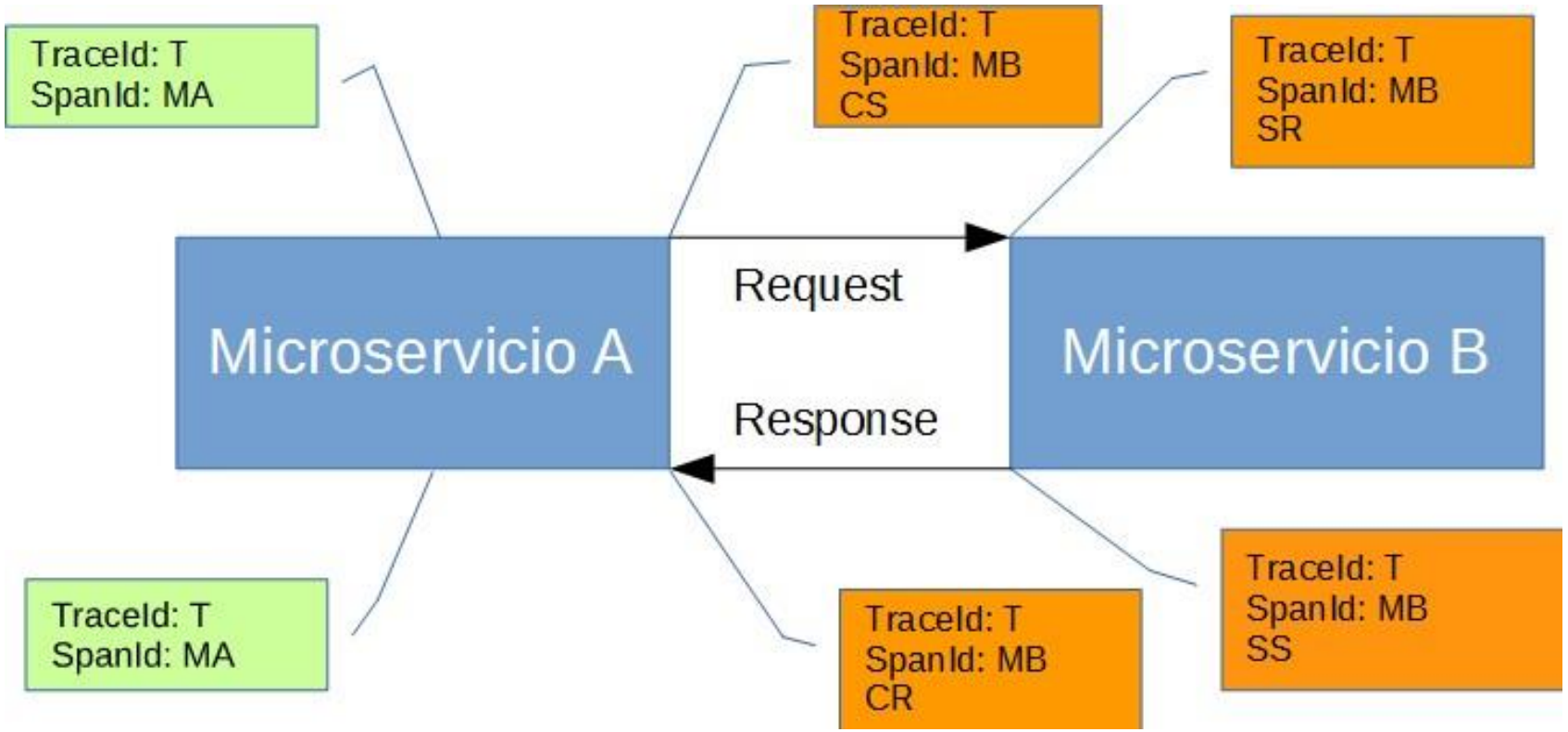


- *Trace*: Un conjunto de spans que forman una estructura de árbol de llamadas, forma la traza de la petición.
- *Span*: Es la unidad básica de trabajo.
  - Ejemplo una llamada a un servicio.
  - Se identifican con un id de span y un id de trace a la que pertenece dicho span.
  - Tienen inicio y fin, y con ello se consigue **trackear** el tiempo de respuesta entre peticiones.

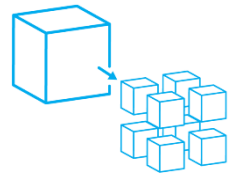
Sleuth Trace y Span



- Comportamiento de una petición HttpRequest



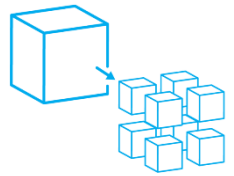
… continua



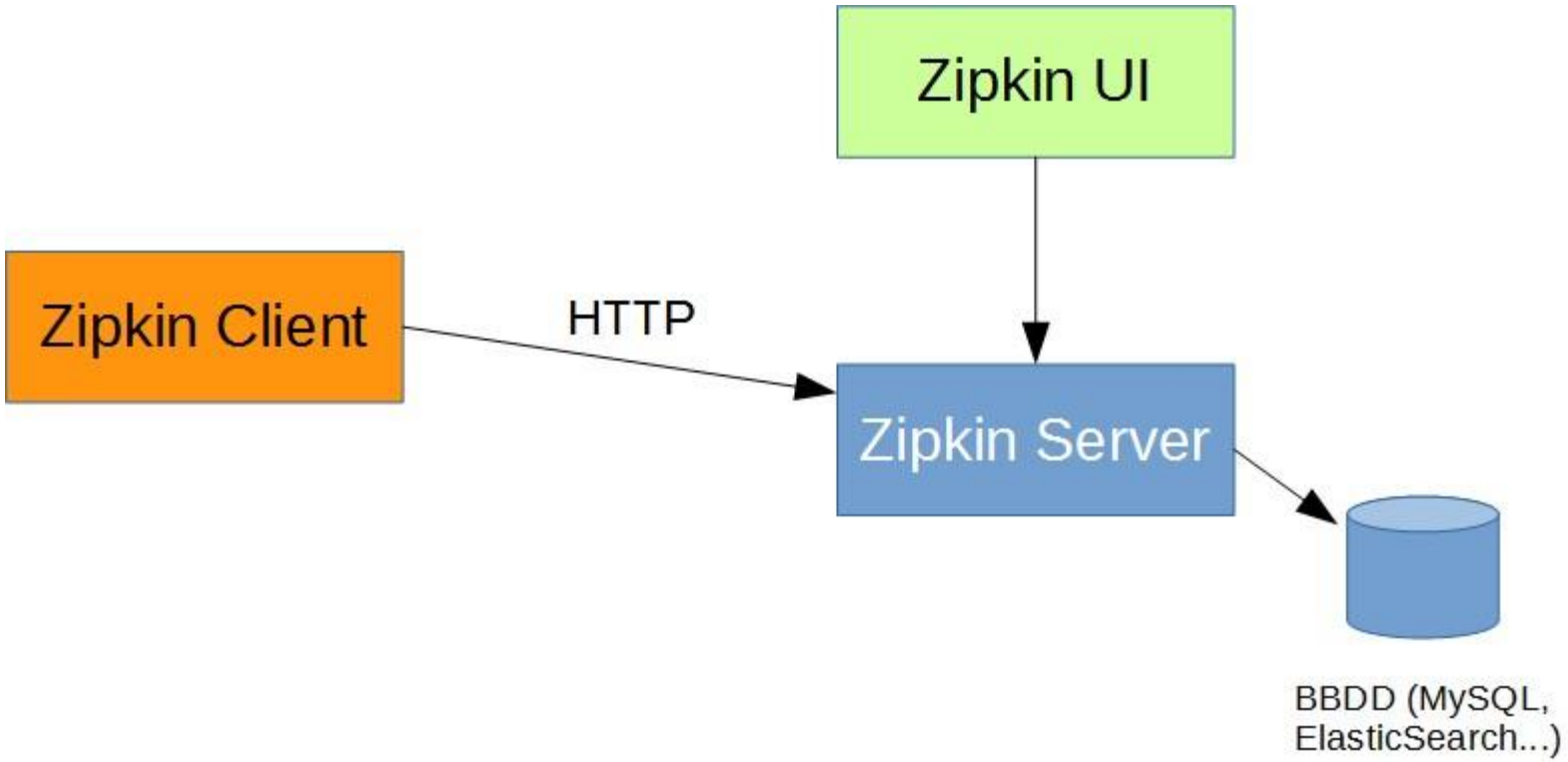
- **Annotation**: Se usa para grabar un evento en el tiempo. Las anotaciones más importantes que usa internamente son:
  - ***cs (Client Sent):*** El cliente ha hecho una petición. Inicio del span.
  - ***sr (Server Received)***: El servidor ha recibido la petición y empieza su procesado. timestampsr - timestampcs = latencia de red.
  - ***ss (Server Sent)***: Envío a cliente desde el servidor de la respuesta. timestampss - timestampsr = tiempo de procesamiento de petición en servidor.
  - ***cr (Client Received)***: Fin del span. El cliente ha recibido correctamente la respuesta del servidor.* timestampcs - timestampcr = tiempo total de la petición*.

Tag: Par clave/valor que identifica cierta información en el span. No contiene timestamps, simplemente identifica

Zipkin

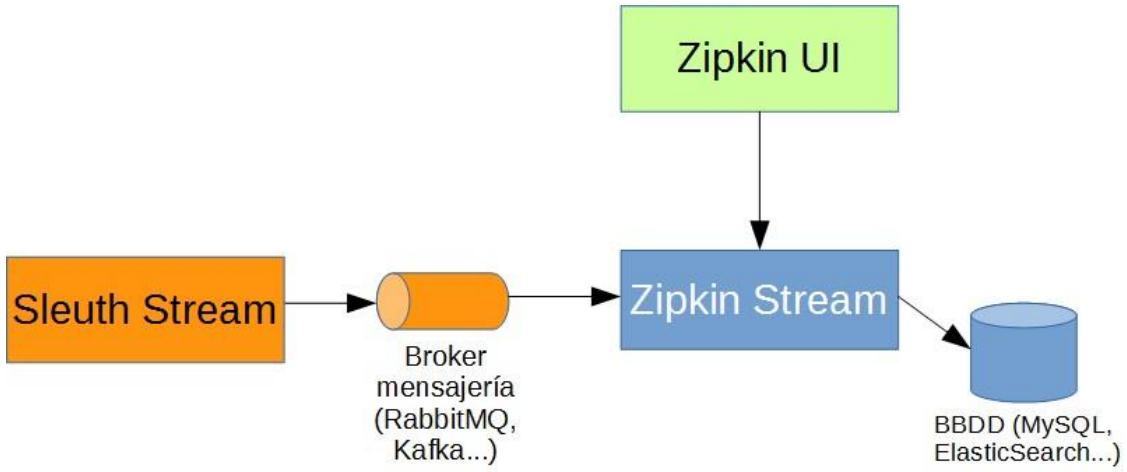


- Zipkin permite el almacenamiento, monitorización y visualización mediante un UI de las trazas de las peticiones.

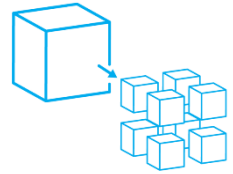


Zipkin Implementaciones

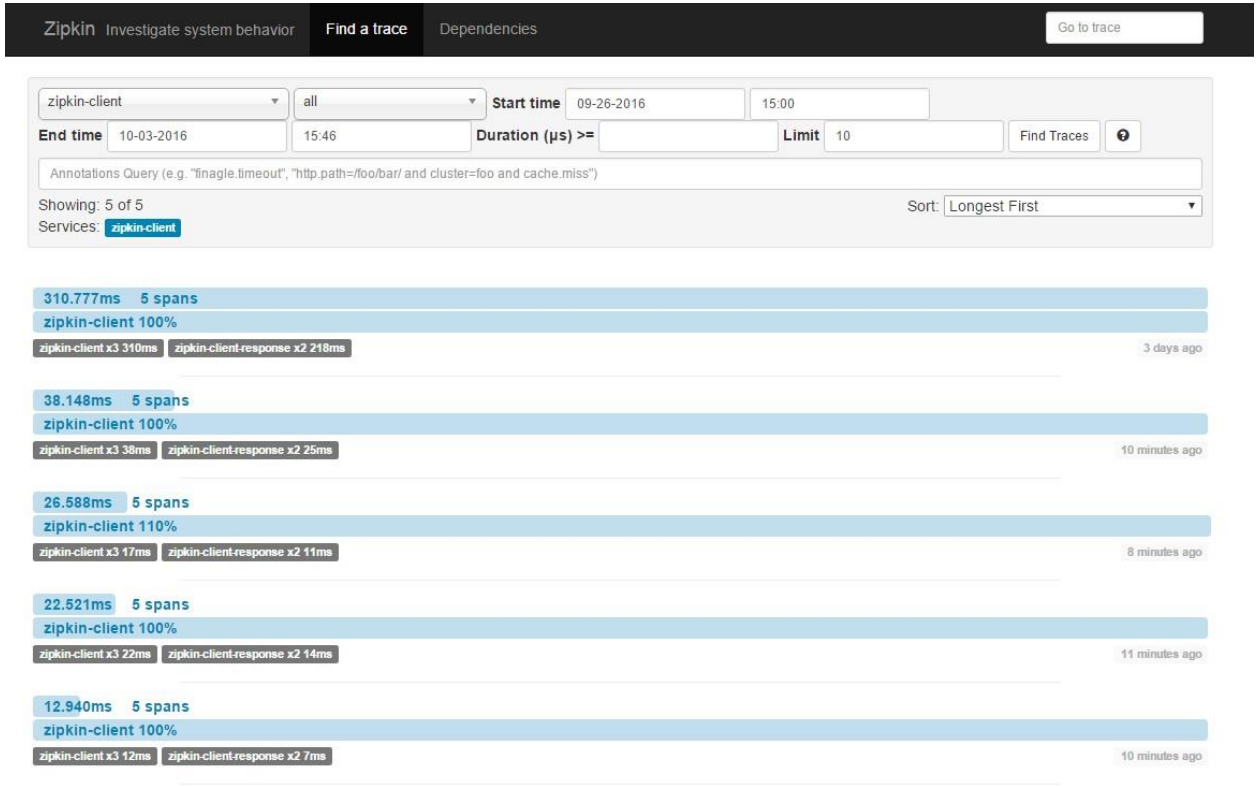
- Mediante Cabeceras Http.
  - Headers
- Usando Brokers de Mensajeria.
  - RabbitMQ, Kafka



Monitoreo usando Dashboards: Zipkin UI

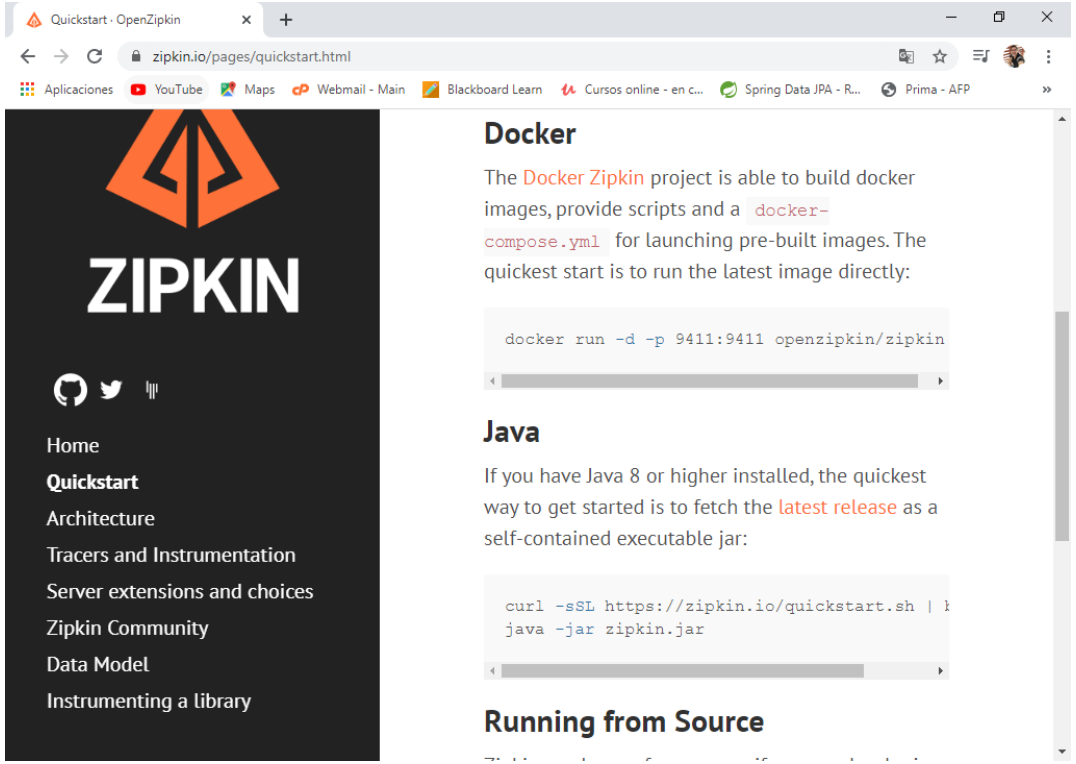


- **Zipkin** provee de una interfaz gráfica que agiliza las consultas a la BD del Zipkin Server y muestra la información recolectada.
- Zipkin UI por defecto usa el puerto: 9411



Obtención de Zipkin

- Java
- Docker Image
- Proyecto Maven del Código Fuente



Preguntas

- Alguna pregunta?