

Atención al cliente mediante Telegram

Resumen de desarrollo de aplicación



Arquitectura basada en microservicios

- **Bot de Telegram:** Este microservicio está diseñado para recibir actualizaciones y mensajes enviados por los usuarios a través de Telegram.
- **Mensajería (Pub/Sub):** Permite la distribución de mensajes entre aplicaciones y microservicios, facilitando la comunicación entre ellos y asegurando una arquitectura ágil y resiliente.
- **Suscripción:** Su principal función es recibir mensajes provenientes de un tema (topic) específico en Google Cloud Pub/Sub. Cuando se recibe un mensaje válido del tipo "CLIENT", el microservicio utiliza Socket.IO para emitir el mensaje a todos los clientes conectados en tiempo real.
- **Interfaz web:** El último microservicio es el frontend basado en VueJS. Aquí, el personal de servicio al cliente recibe y visualiza los mensajes distribuidos por el servicio de suscripción.

Solución orientada a eventos

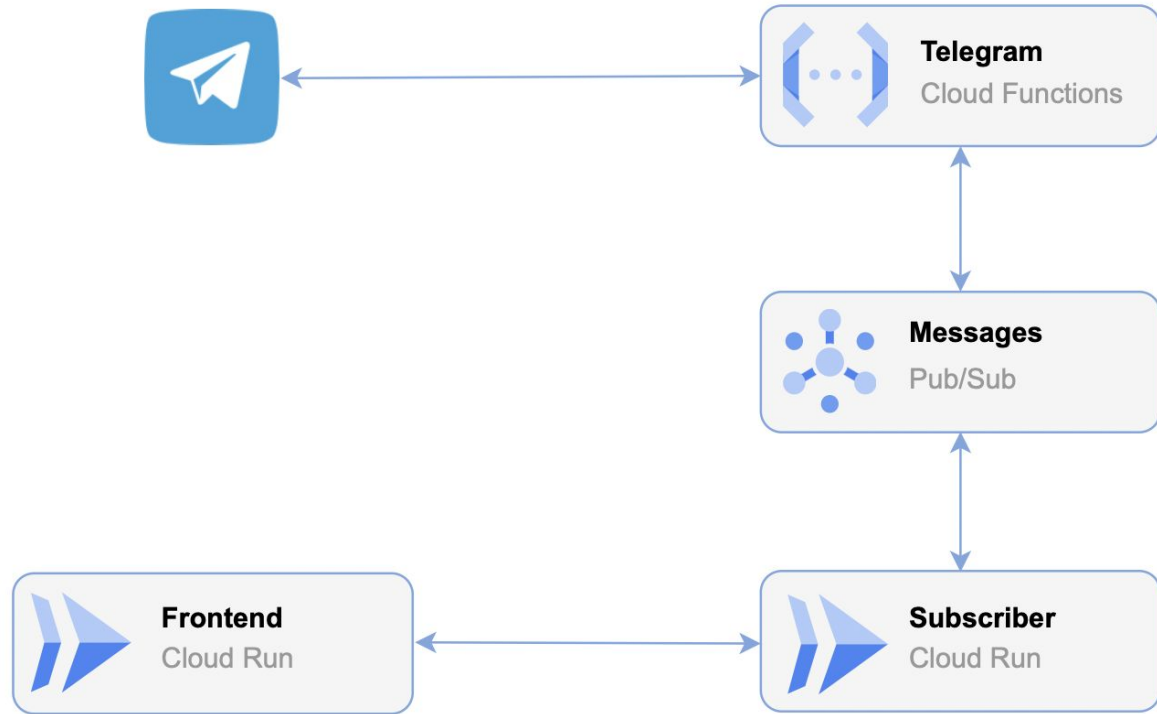
Esta arquitectura orientada a eventos permite una comunicación ágil y escalable entre los distintos microservicios. Cada uno de estos componentes puede interactuar de forma asíncrona gracias a Google Cloud Pub/Sub, un sistema de mensajería distribuido que facilita la transmisión de eventos entre ellos.

Además, para asegurar una experiencia en tiempo real y una interacción fluida con los clientes, se han incorporado websockets en la arquitectura. Estos sockets también funcionan como eventos, lo que permite que el tercer microservicio, el frontend basado en Vue.js, reciba y muestre mensajes distribuidos por el servidor de mensajería en tiempo real de manera instantánea.

El enfoque orientado a eventos no solo mejora la reactividad y la escalabilidad del sistema, sino que también brinda la flexibilidad de agregar nuevos microservicios sin perturbar el funcionamiento existente. Así, se puede adaptar fácilmente la solución a futuros cambios y requerimientos.

Infraestructura GCP

- **Bot de Telegram:** Cloud Function que recibe actualizaciones y mensajes de usuarios en Telegram.
- **Pub/Sub:** GCP Pub/Sub para distribuir mensajes entre microservicios.
- **Suscripción:** Cloud Run que recibe mensajes de un tema de Pub/Sub y utiliza Socket.IO para enviarlos en tiempo real.
- **Interfaz web:** Otro Cloud Run con Vue.js, donde el personal de servicio al cliente ve los mensajes distribuidos.



Frontal web para agentes

Cuando un usuario envía un mensaje solicitando ayuda al bot de Telegram, ese mensaje se distribuye a través de Pub/Sub y llega instantáneamente a la interfaz. Los agentes pueden ver el mensaje de manera inmediata y brindar asistencia al usuario de manera ágil y efectiva.

Está diseñado con una interfaz familiar y similar a Telegram, lo que permite a los agentes de servicio al cliente sentirse cómodos y rápidamente familiarizados con la plataforma. La estética y la disposición del frontend se asemejan a la de Telegram, lo que facilita la navegación y la interacción para los agentes.

Esto garantiza una experiencia intuitiva y sin complicaciones al brindar asistencia a los usuarios, ya que los agentes pueden aprovechar su conocimiento previo de Telegram para responder rápidamente a los mensajes recibidos.

Además, está diseñado utilizando los colores y formas corporativos distintivos, y refleja los valores y la identidad de la organización.

Recolección de datos en tiempo real

La solución tiene un sistema de recolección de datos en tiempo real mediante Google Cloud Pub/Sub con retención de mensajes activada de 7 días. También se pueden configurar métricas en Cloud Functions y Cloud Run para monitorizar el rendimiento. Estos datos nos permiten tomar decisiones informadas y mejorar continuamente los servicios.

Escalabilidad

La solución está diseñada para ser altamente escalable. Gracias a GCP, podemos aumentar o reducir dinámicamente los recursos según la demanda, garantizando un rendimiento óptimo en cualquier situación. La arquitectura de microservicios y el uso de Pub/Sub permite escalar de manera eficiente y brindar una experiencia fluida a medida que crece la base de usuarios y las necesidades de la aplicación evolucionan.