

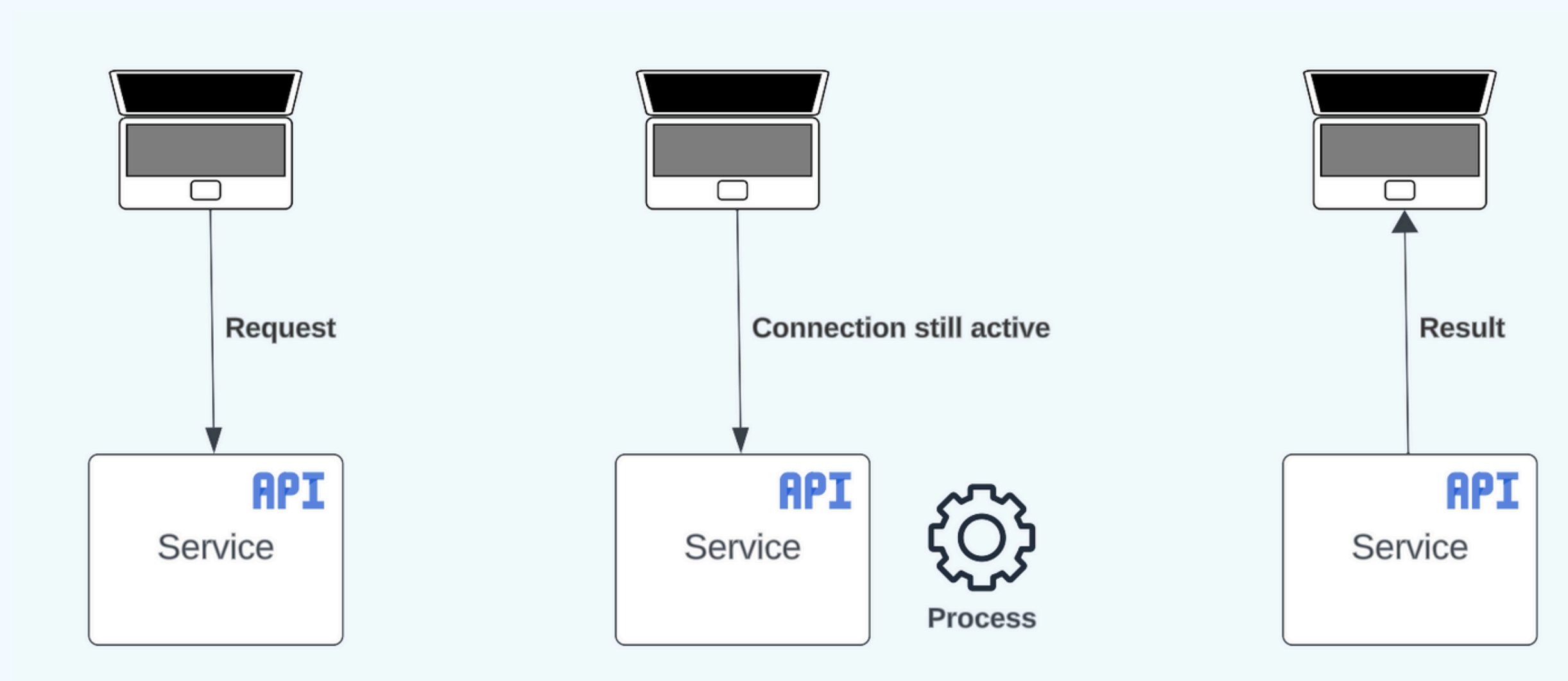
CONCEPTOS CLAVE DISEÑO SISTEMAS A GRAN ESCALA

MESSAGE BROKERS

Daniel Blanco Calviño

COMUNICACIÓN SÍNCRONA

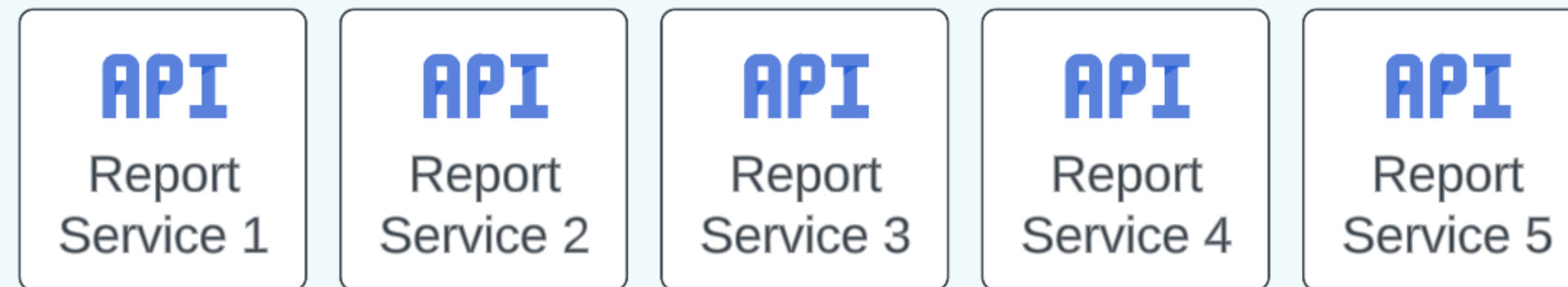
1. El cliente realiza una petición.
2. El servidor procesa la petición, manteniendo la conexión abierta con el cliente en todo momento.
3. Al finalizar el procesamiento, se devuelve el resultado al cliente y se cierra la conexión.



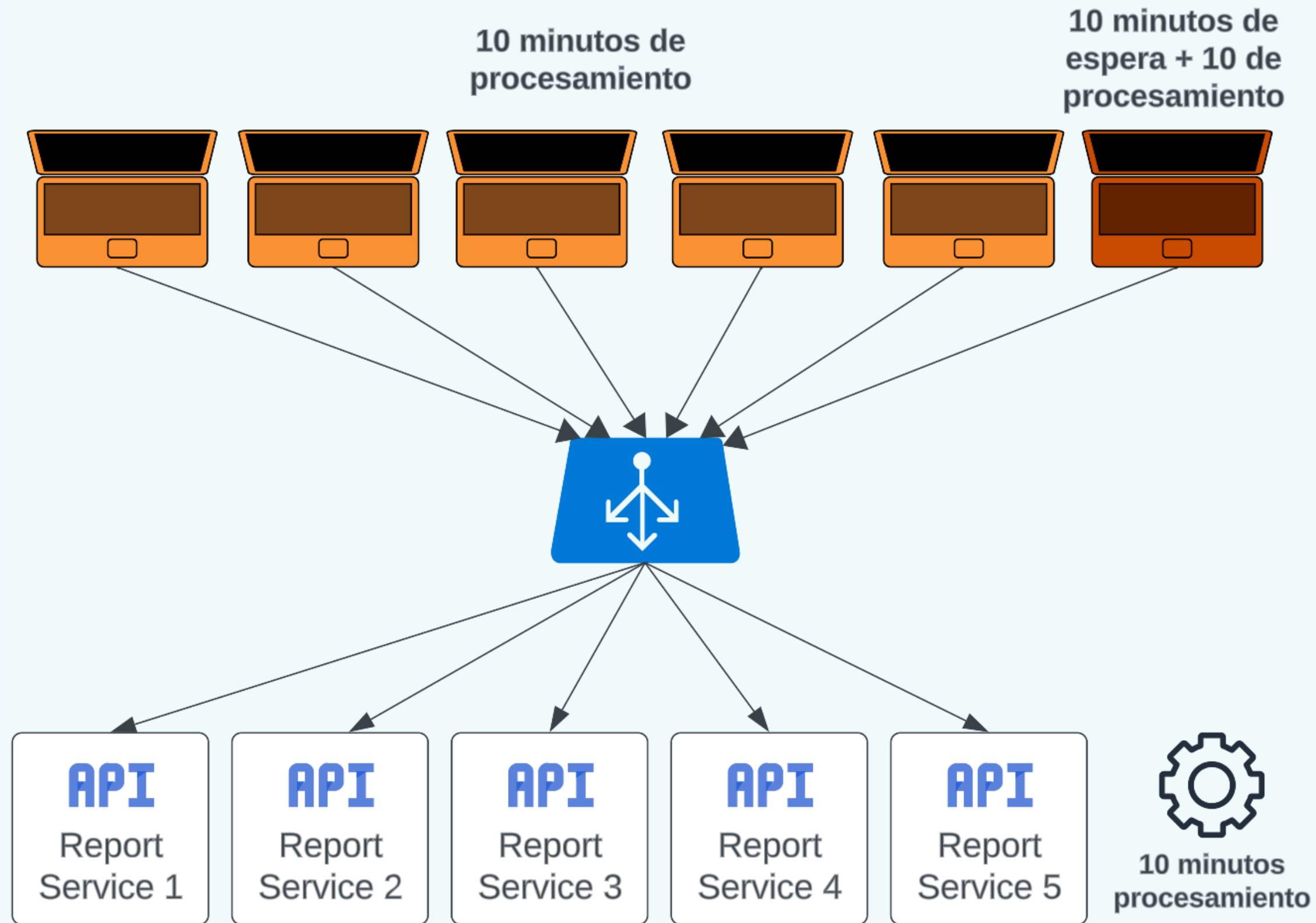
COMUNICACIÓN SÍNCRONA

- Beneficios.
 - Es la más utilizada ya que **es muy simple**.
 - **Óptima** para tareas de **corta duración**.
- Inconvenientes.
 - Se necesita una **conexión saludable en ambos extremos**. Más complejo en **tareas largas**.
 - Cliente y servidor **deben mantenerse a la espera** hasta que el procesamiento termine.
 - **Sufriremos mucho con picos de tráfico** para tareas de larga duración.

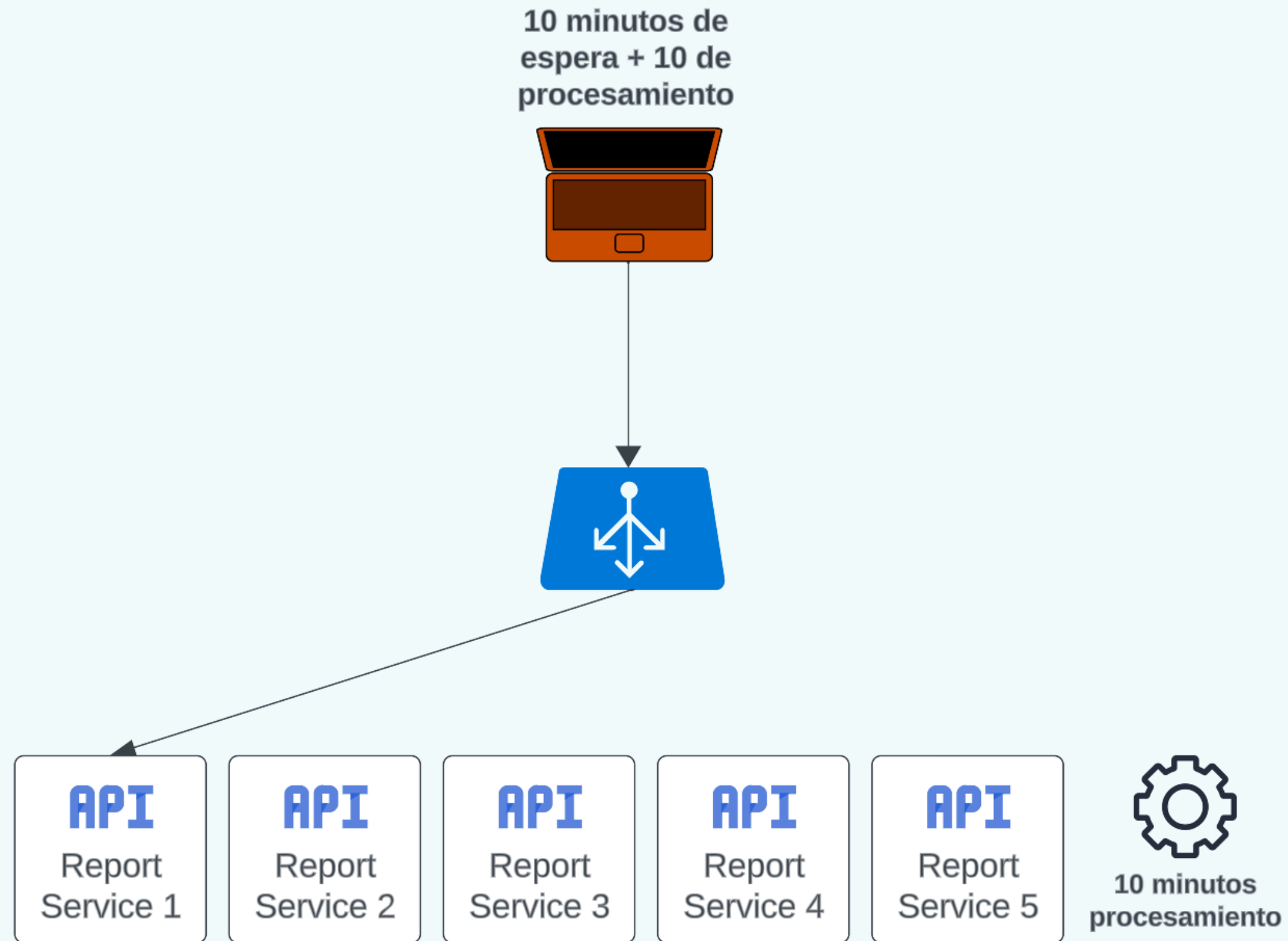
COMUNICACIÓN SÍNCRONA



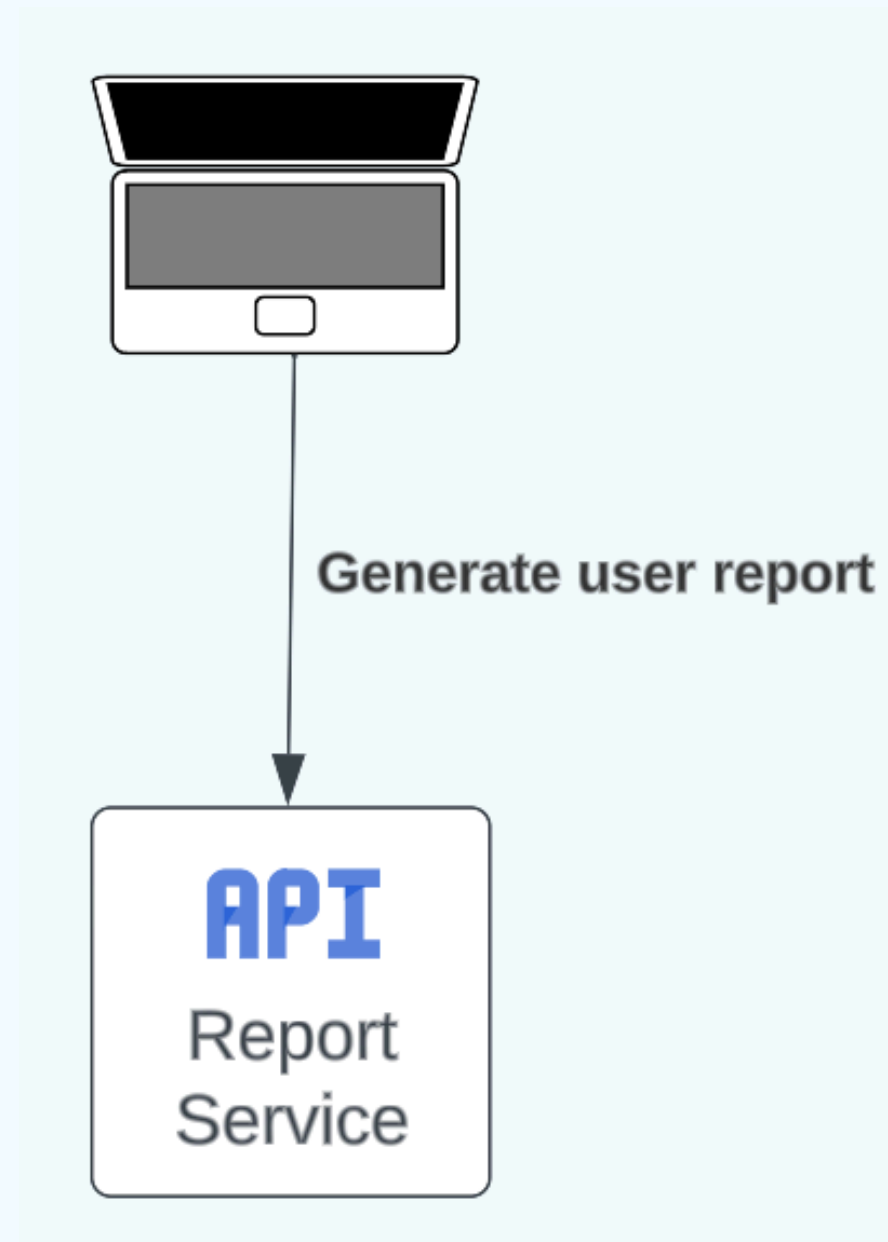
COMUNICACIÓN SÍNCRONA



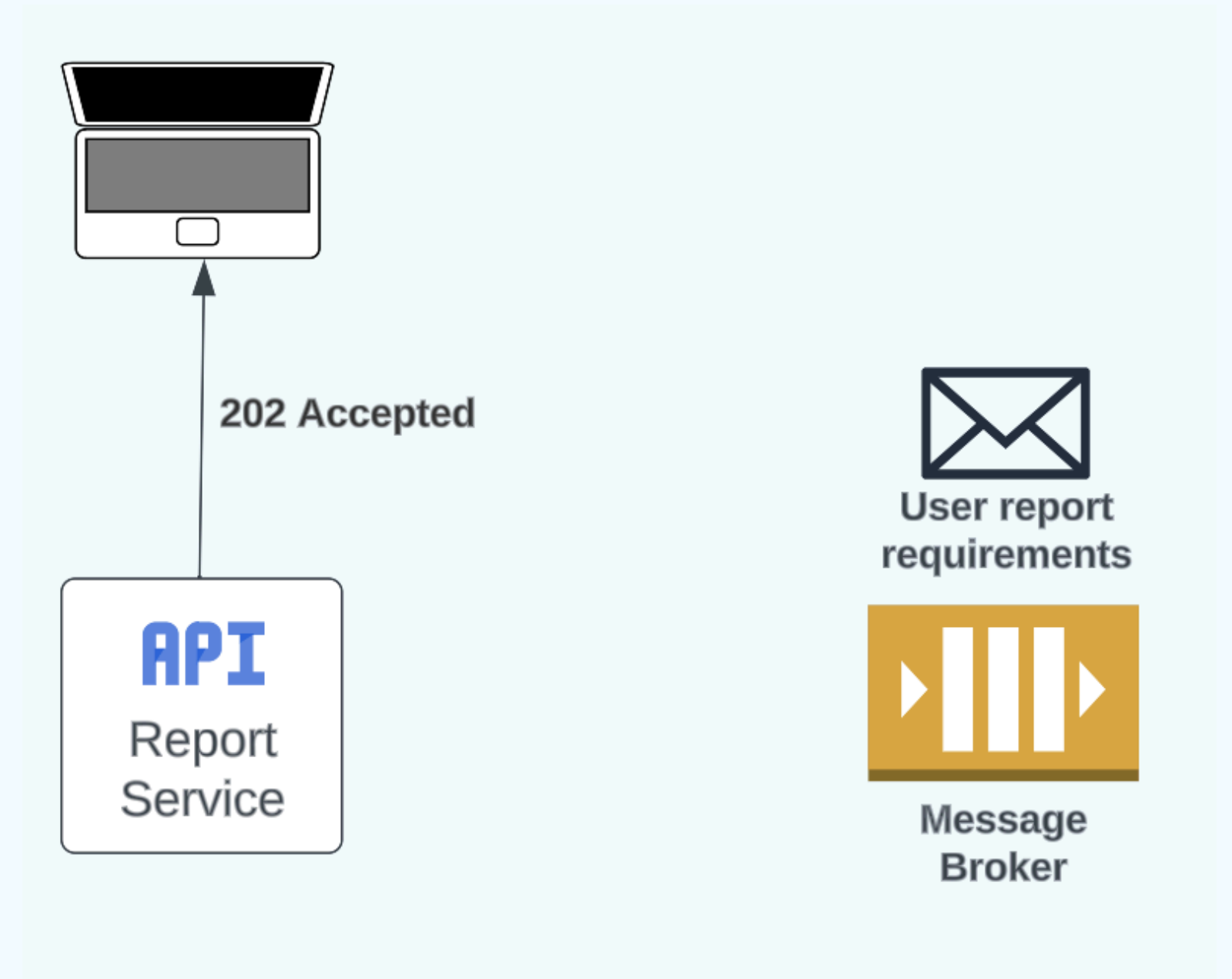
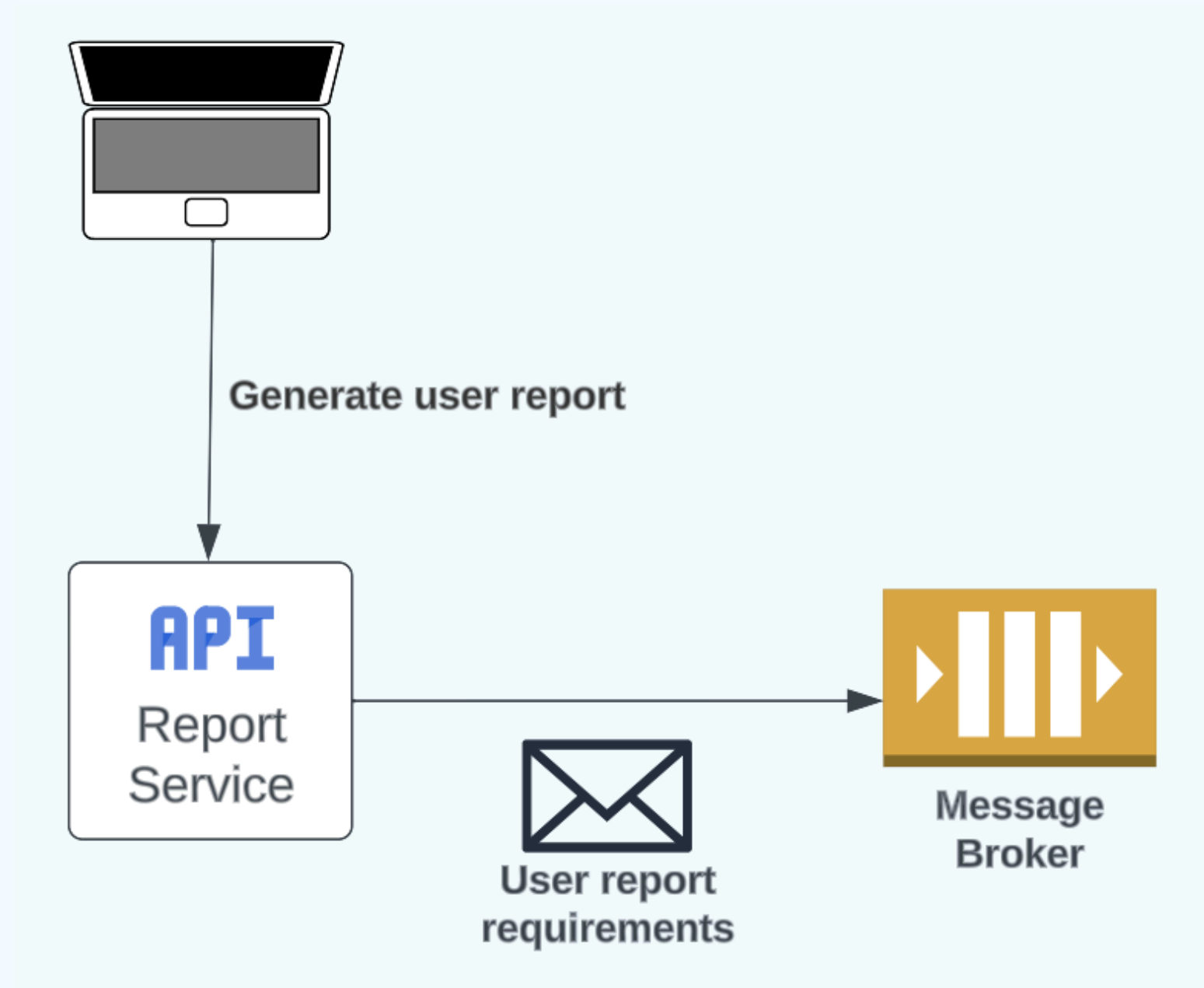
COMUNICACIÓN SÍNCRONA



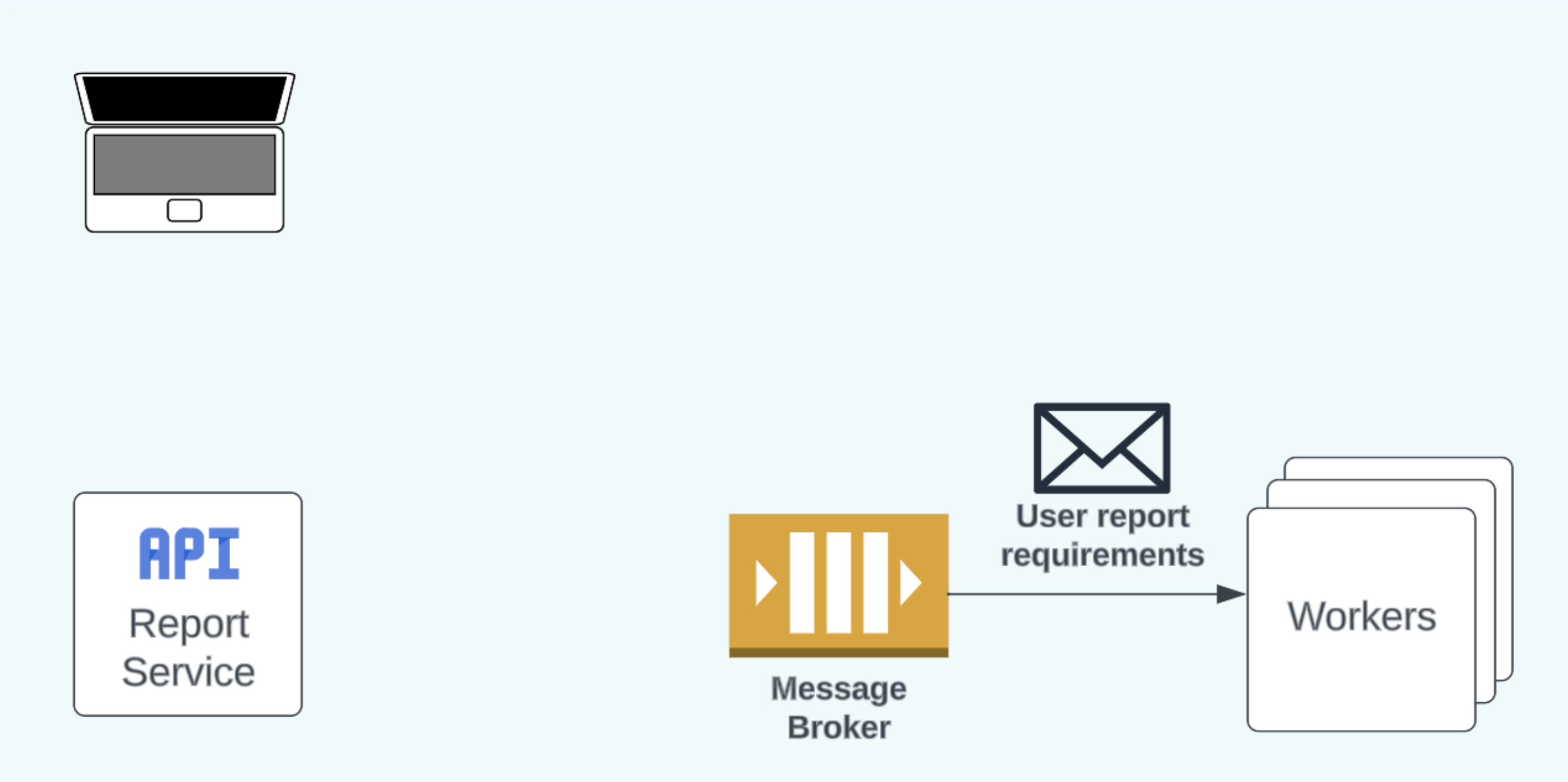
COMUNICACIÓN ASÍNCRONA



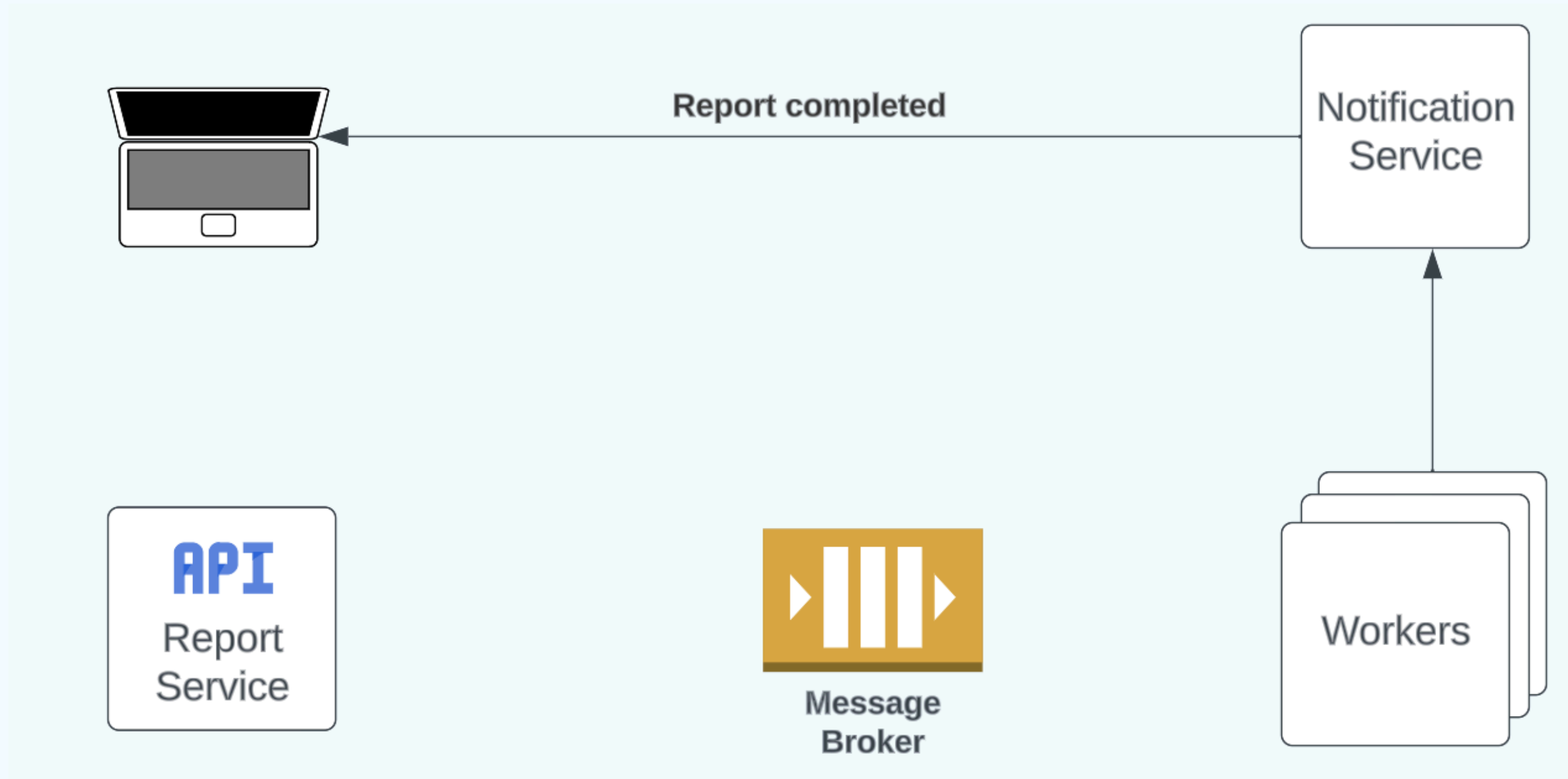
COMUNICACIÓN ASÍNCRONA



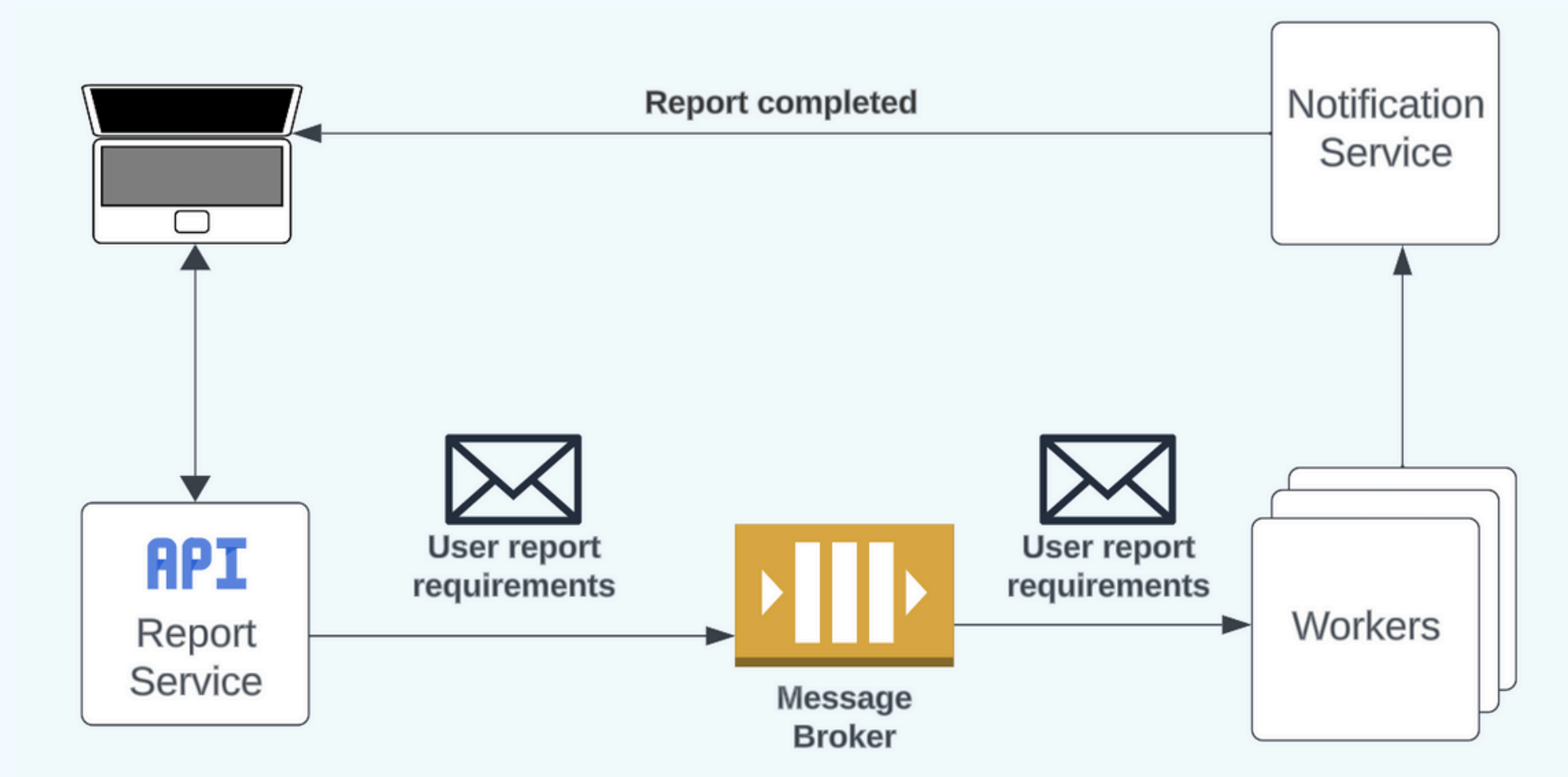
COMUNICACIÓN ASÍNCRONA



COMUNICACIÓN ASÍNCRONA



BENEFICIOS COMUNICACIÓN ASÍNCRONA



- **Menor riesgo de interrupción de conexión** con el cliente.
- Gestión eficiente y fiable de los picos de peticiones.

MESSAGE BROKERS

- **Componente** software que **facilita la comunicación asíncrona** entre dos módulos de nuestro sistema.
 - **Productores.** Envían mensajes al message broker.
 - **Consumidores.** Consumen y procesan los mensajes del message broker.
 - **Colas.** Estructura FIFO en la que se almacenan los mensajes. Se entregan de forma síncrona.
 - **Publicación / suscripción.** Los consumidores se suscriben a un topic. Los productores envían mensajes a un topic, el cual será consumido por los suscriptores.

BENEFICIOS MESSAGE BROKERS

- **Entrega de mensajes garantizada.**
 - Si un consumidor falla antes de finalizar el procesamiento, el mensaje vuelve a la cola.
- **Mayor escalabilidad y fiabilidad.**
 - El rendimiento se ve un poco afectado.
- **Desacoplamos totalmente los productores de los consumidores.**
 - Nos permite escalar independientemente ambos, según la carga de trabajo.
- **Balancean la carga** automáticamente.
 - Se reparte de forma equitativa entre todos los consumidores.

EJEMPLOS MESSAGE BROKERS

- Apache Kafka
- Apache ActiveMQ
- RabbitMQ
- AWS SQS (Simple Queue Service)
- Azure Service Bus

