

EJERCICIOS PRÁCTICOS

---

# E-COMMERCE

---

Daniel Blanco Calviño

# REQUISITOS

- **Requisitos funcionales.**
  - Ingesta de productos.
  - Búsqueda.
  - Carrito.
  - Confirmación de compra.
  - Consulta de pedidos pasados o en curso.
- **Requisitos no funcionales.**
  - Alta consistencia.
  - Alta disponibilidad.

# REQUISITOS

- **Requisitos funcionales.**
  - Ingesta de productos.
  - Búsqueda.
  - Carrito.
  - Confirmación de compra.
  - Consulta de pedidos pasados o en curso.
- **Requisitos no funcionales.**
  - Alta consistencia.
    - Pagos, stock, etc.
  - Alta disponibilidad.

# REQUISITOS

- **Requisitos funcionales.**

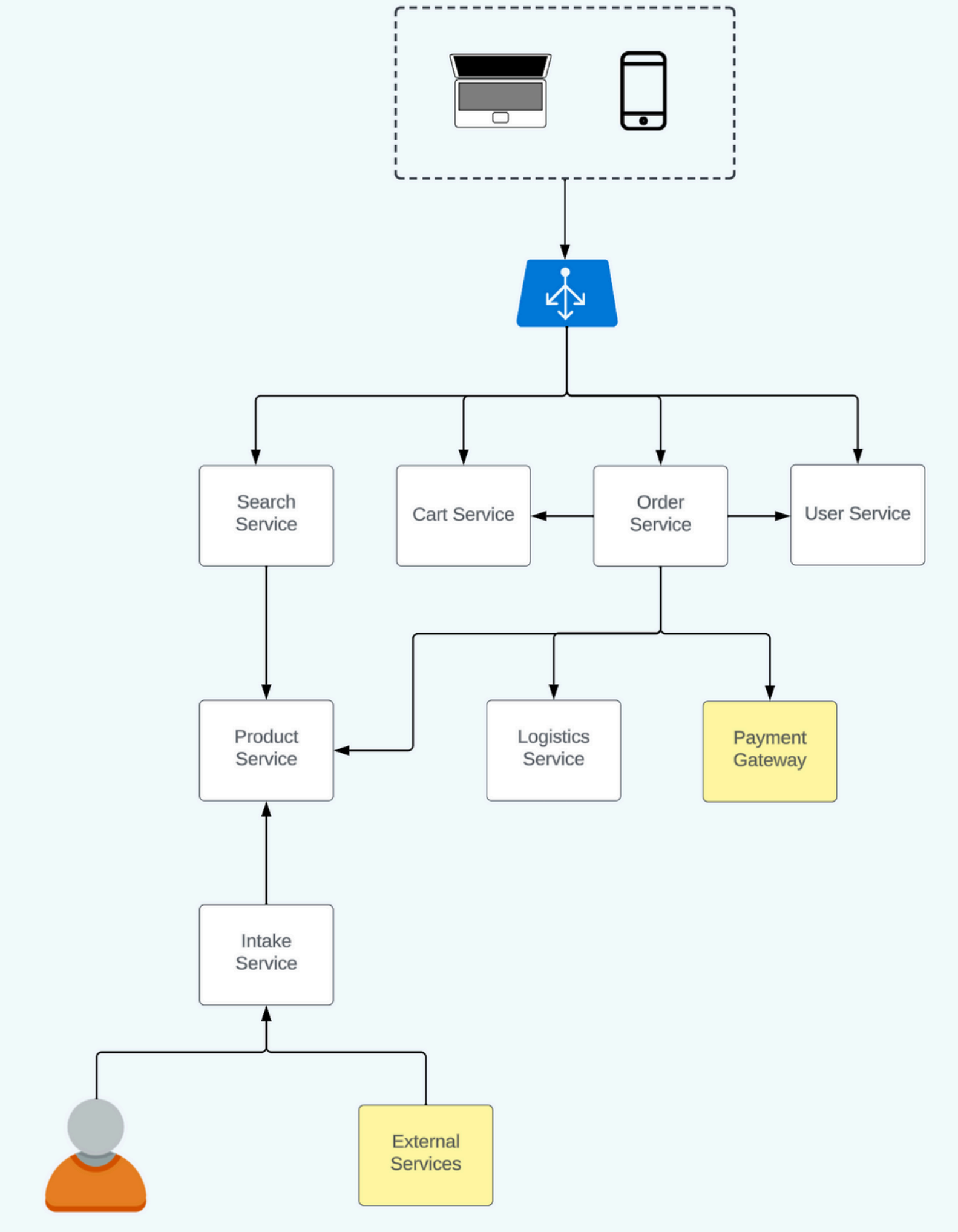
- Ingesta de productos.
- Búsqueda.
- Carrito.
- Confirmación de compra.
- Consulta de pedidos pasados o en curso.

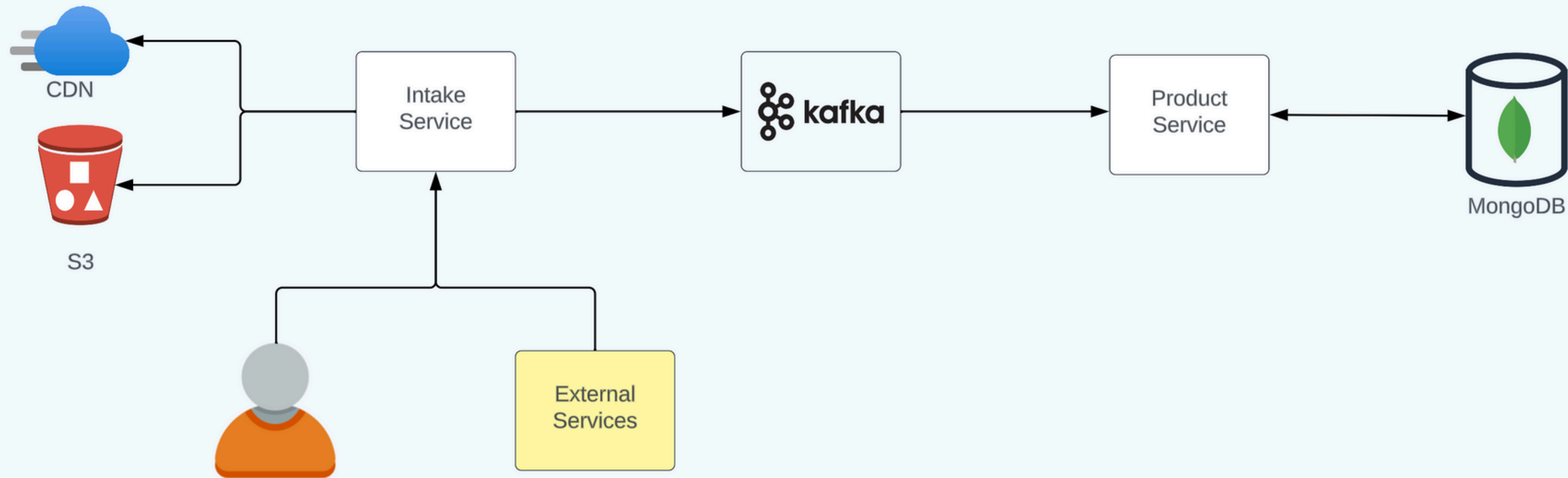
- **Requisitos no funcionales.**

- Alta consistencia.
  - Pagos, stock, etc.
- Alta disponibilidad.
  - Búsqueda, carritos, información de productos, etc.

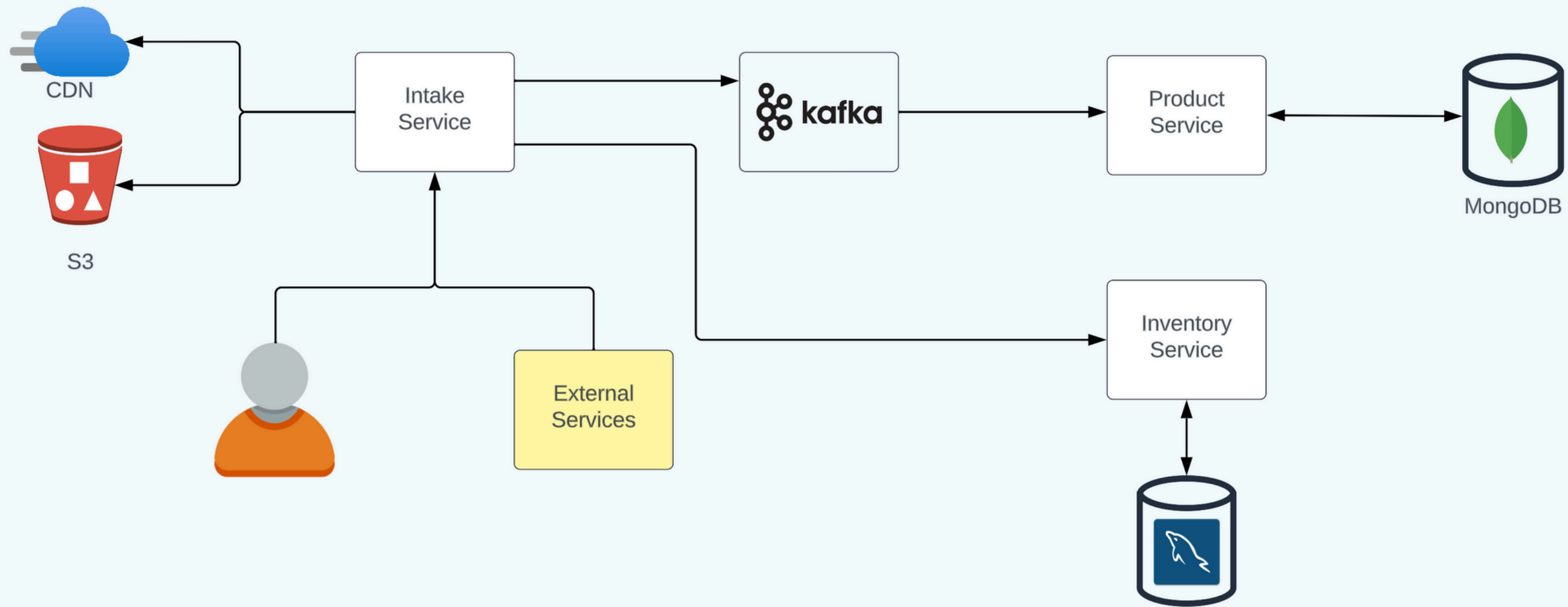
# HIPÓTESIS Y ESTIMACIONES

- 50M usuarios diarios.
  - 1/10 realizan pedidos. **5M pedidos de media al día.**
- Picos máximos de 100M usuarios y 15M pedidos al día.
- Se añaden 5000 productos nuevos al día.
  - 2MB de datos y archivos multimedia. **3.65TB de información de productos al año.**
- Cada usuario realiza una media de **20 consultas al día.**
  - $20 * 100M = 2000M$  de consultas al día.
    - $2000M / 24 / 60 / 60 = \mathbf{23000}$  consultas al segundo (QPS).



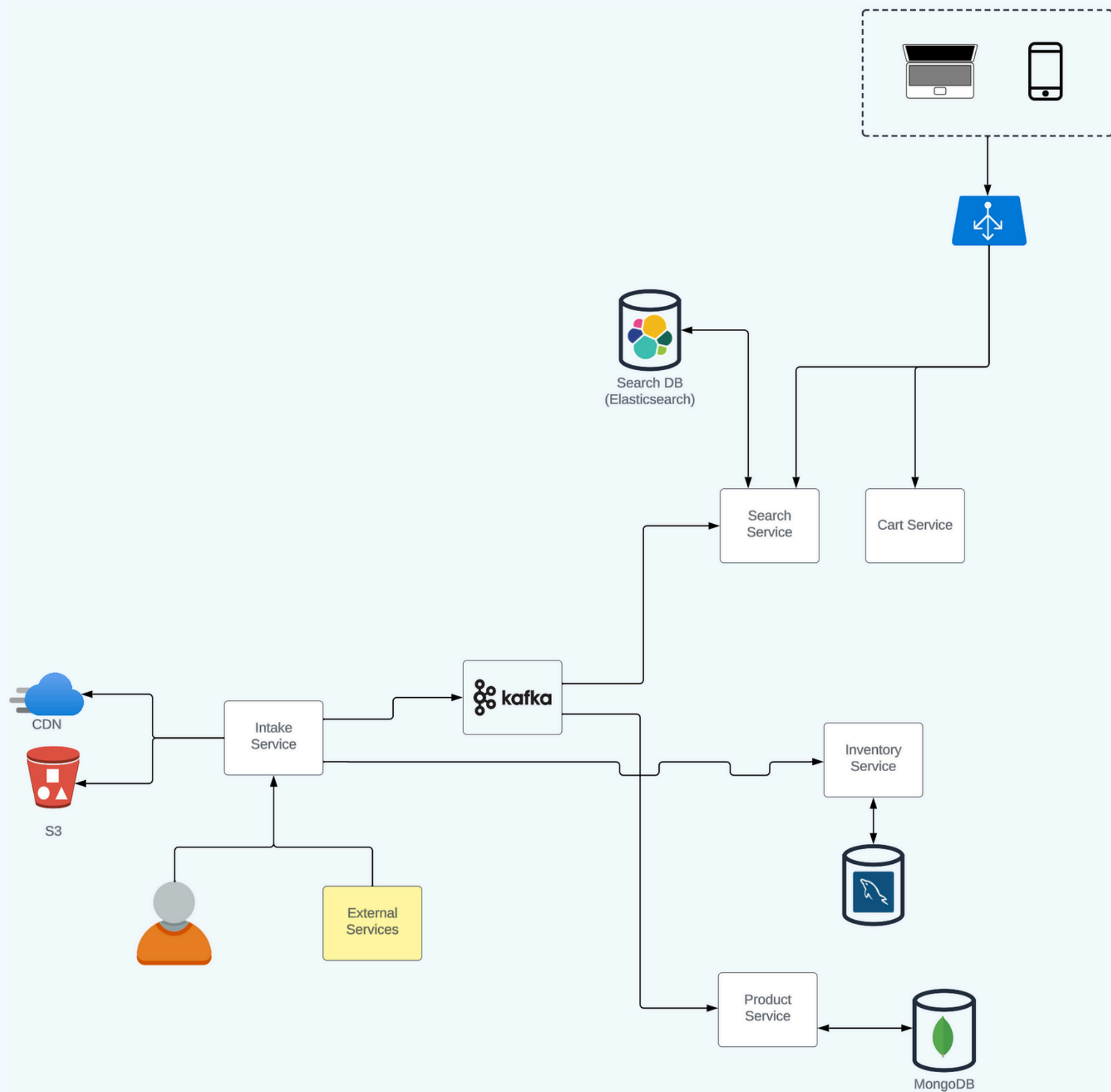


- Intake Service **subirá archivos multimedia** a AWS S3 y al CDN si es necesario.
  - **Enviaré eventos** para añadir, modificar y eliminar productos a Kafka.
- Product Service consumirá los eventos.
  - **Almacenará productos** en BBDD documental MongoDB.
    - Perfecta para la naturaleza no estructurada de la información de productos. Los productos puede tener campos muy dispares.
  - ¿Stock? MongoDB no sería la mejor opción.
    - Sacrifica consistencia para ganar disponibilidad.

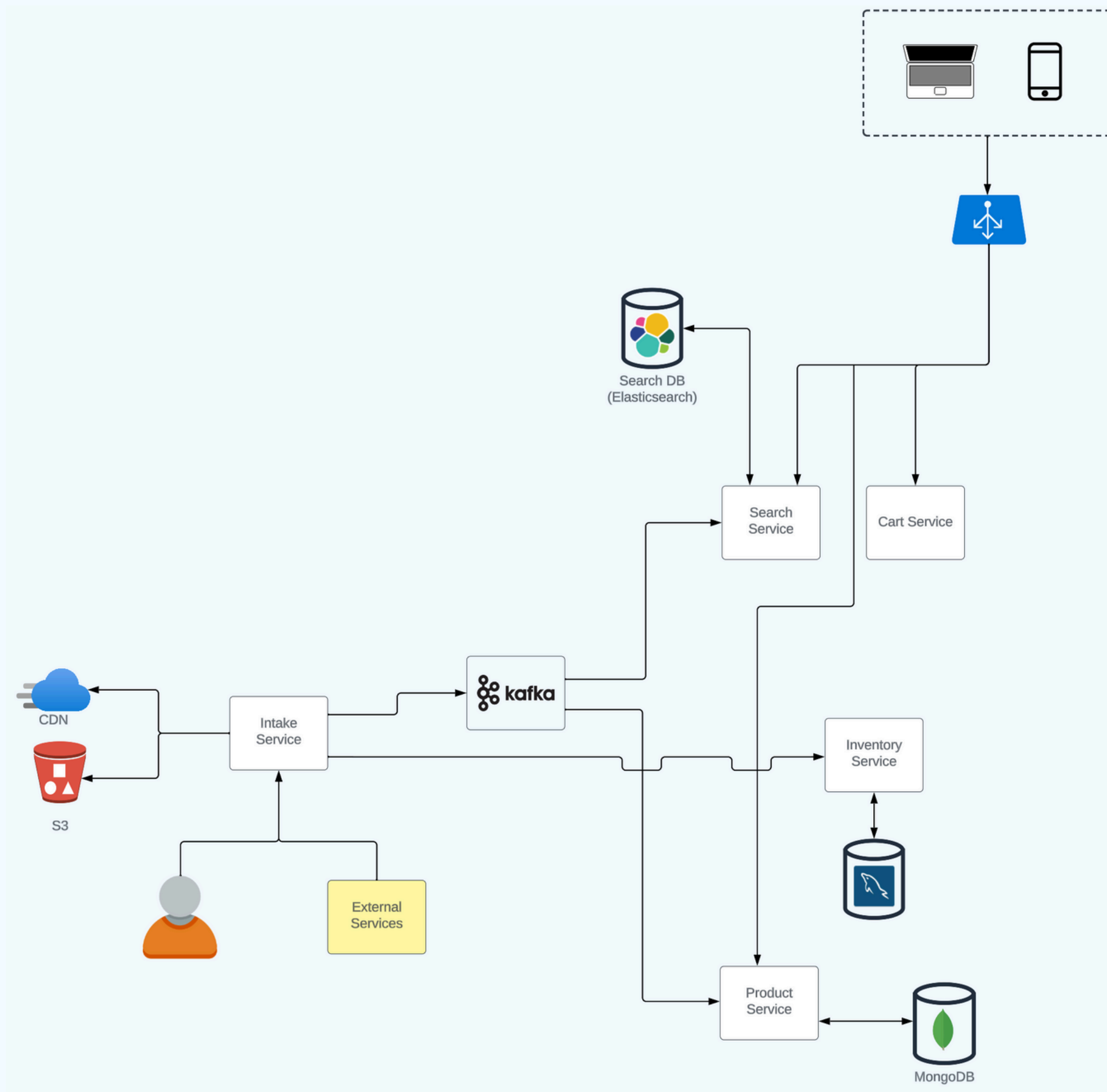


- Con una BBDD relacional y transacciones ACID aseguramos **consistencia en el stock**.
- Gracias a sistema de almacenamiento como MongoDB **aumentamos la disponibilidad de la información de los productos**.

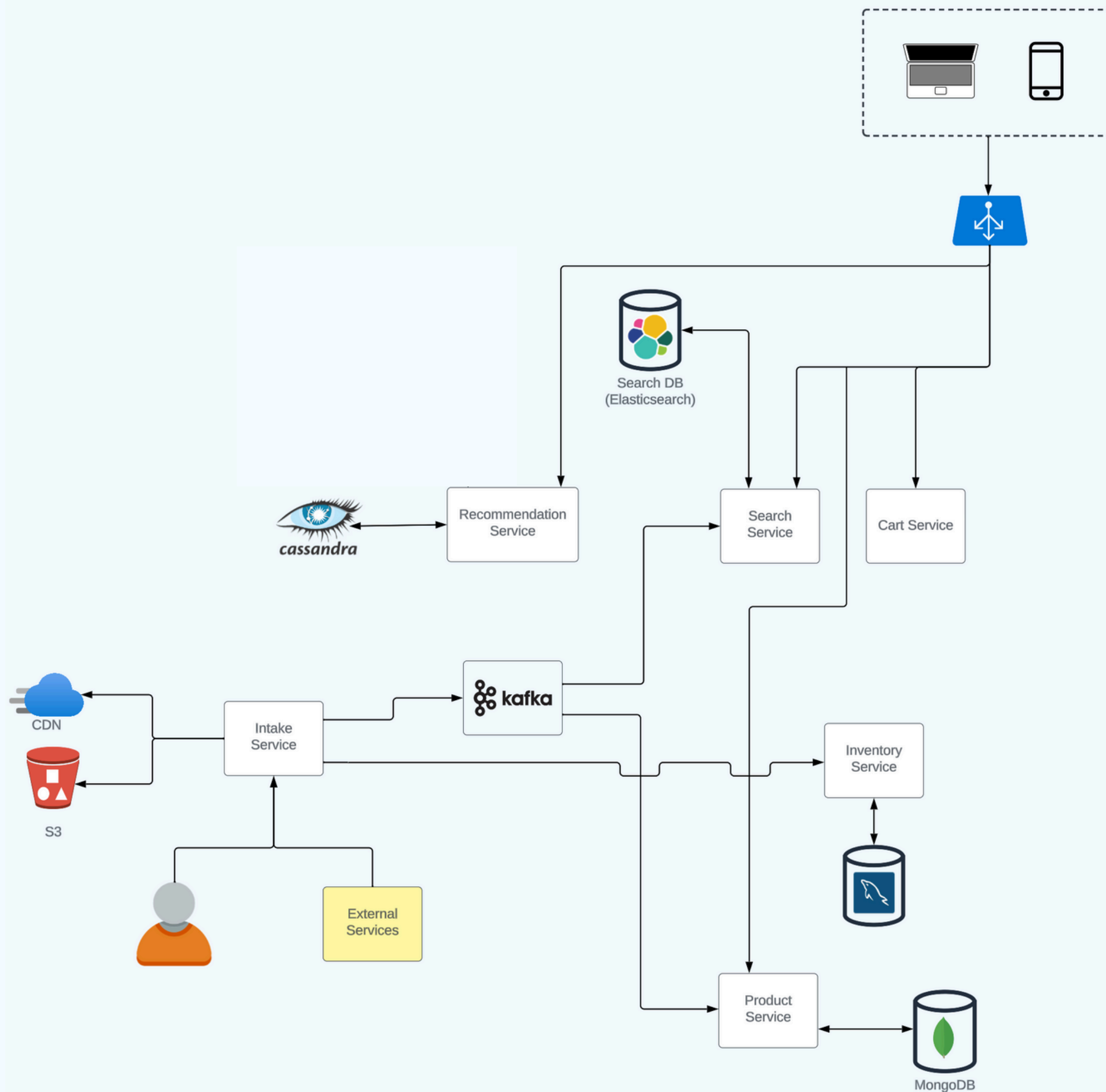




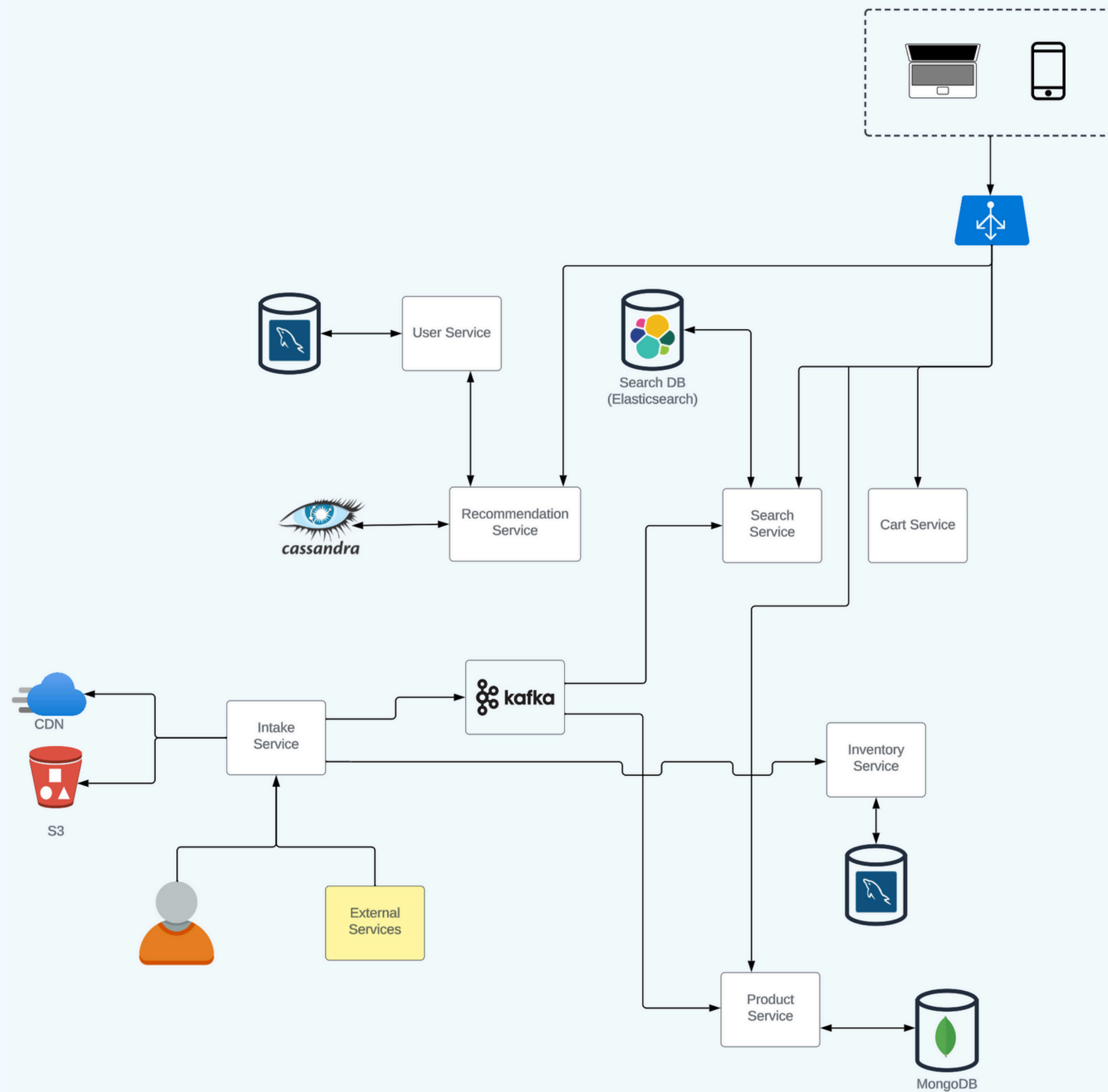
- MongoDB es ineficiente para consultas complejas.
- Añadimos **BBDD optimizada para búsquedas de texto**. Elasticsearch.
- Search Service consumirá eventos del Intake Service.



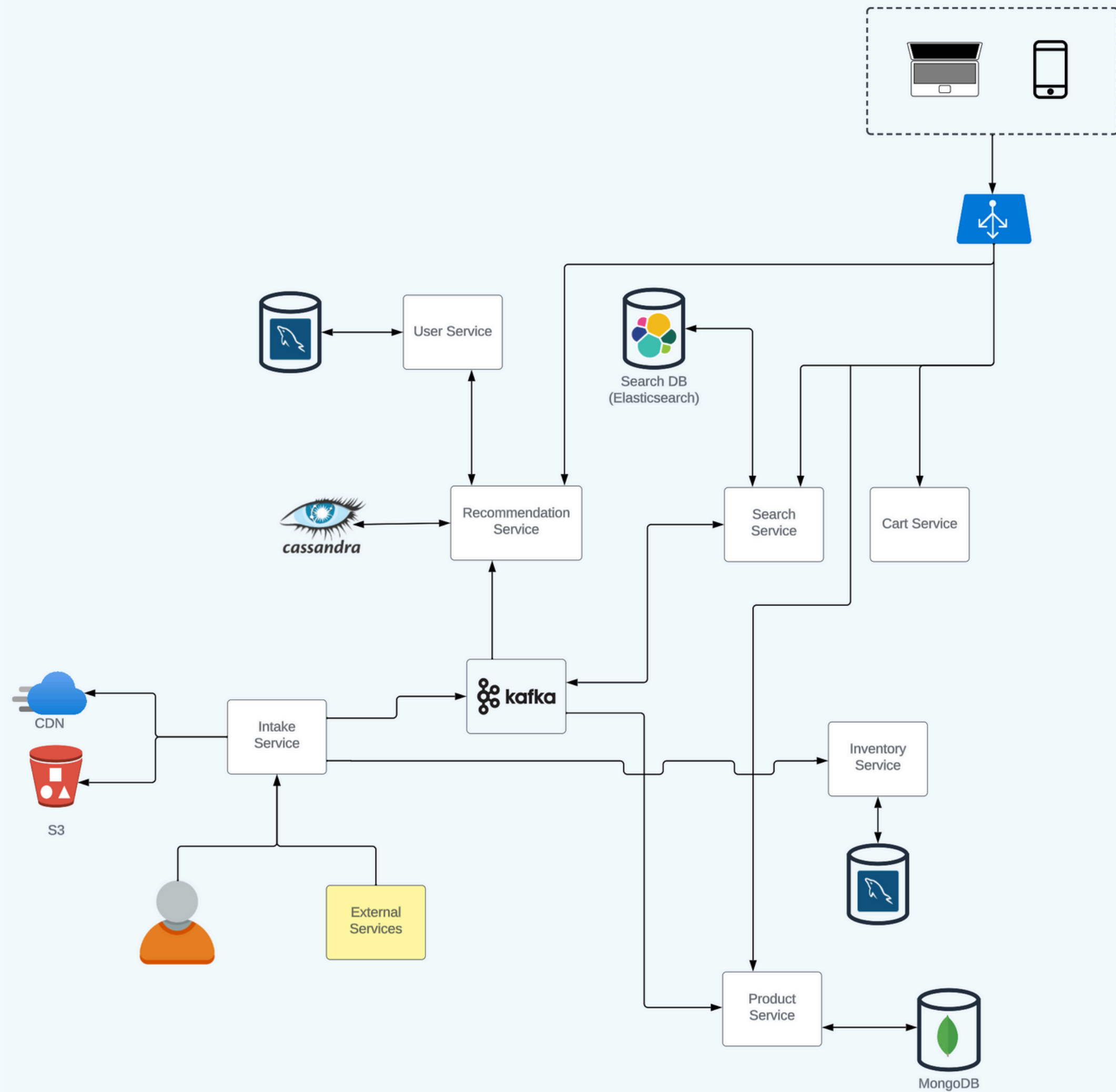
- Search DB no tiene por qué guardar toda la información.
- Consultamos a Product Service para los detalles de un producto.
- Priorizamos disponibilidad sobre consistencia en los productos y la búsqueda.
- Es posible tener inconsistencias entre ambas BBDD.



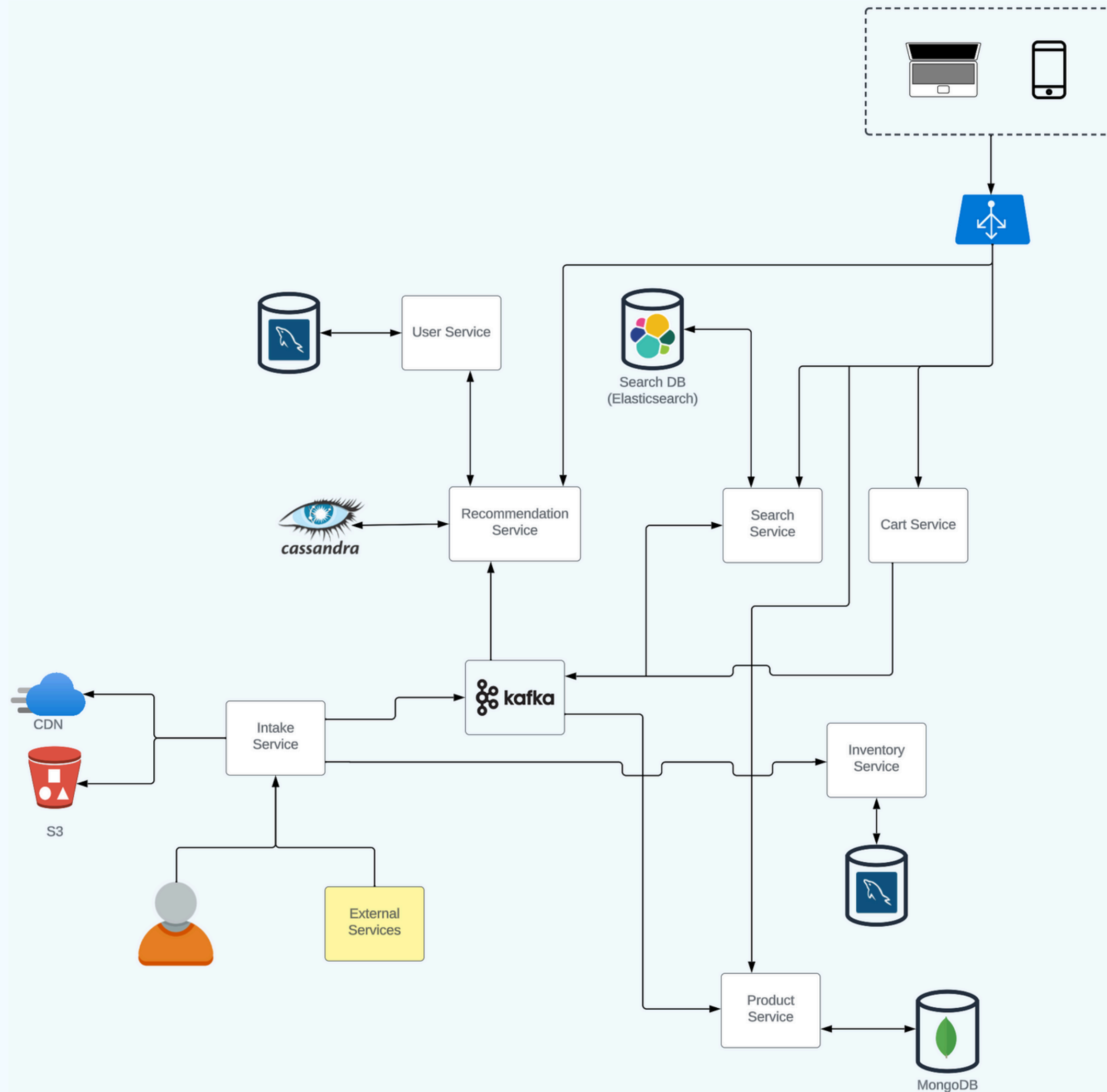
- Recommendation Service mostrará **productos recomendados** en función de los usuarios.
- Necesita obtener los datos de algún punto.



- Obtendrá datos de:
  - User Service



- Obtendrá datos de:
  - User Service
  - Search Service.
    - A través de Kafka.



- Obtendrá datos de:
  - User Service
  - Search Service.
    - A través de Kafka.
  - Cart Service.
    - A través de Kafka.

