

Estudiante: Jersson Fabián Buitrago Murcia

4. En el contexto de una empresa que está migrando su sistema monolítico a una arquitectura basada en microservicios, ¿cómo utilizarías el modelo C4 para diseñar y comunicar la nueva arquitectura, asegurando que los equipos de desarrollo, operaciones y negocio comprendan su rol y responsabilidades, y cómo abordarías la identificación y mitigación de riesgos asociados a la transición, como la consistencia de datos y la latencia inter-servicios?

Comunicado a la nueva arquitectura.

Para asegurar que todos los equipos comprendan su rol y responsabilidad en la nueva arquitectura:

- Se debe utilizar diagramas C4 sencillos y bien estructurados para representar la arquitectura.
- Organizar reuniones para explicar los nuevos conceptos claves del modelo C4 y de cómo se va aplicar en la nueva estructura.
- Crear una documentación concisa y actualizada donde se describa cada nivel del modelo C4 y su relación con cada componente.
- Utilizar herramientas como Lucidchart para crear los diagramas colaborativos y facilitar la comunicación entre equipos.

Identificación y mitigación de riesgos:

Como la consistencia de datos y la latencia inter-servicios. Para abordarlos:

Consistencia de datos:

- Identificar datos críticos.
- Estrategia de sincronización.
- Duplicación de datos.
-

Latencia inter-servicios

- Optimizar las comunicaciones.
- Caching.
- Circuit breakers.

Estrategias de migración.

- Big bang.
- Migración incremental.

5. ¿Cómo integrarías de manera efectiva la representación de aspectos no funcionales (como escalabilidad, seguridad y disponibilidad) dentro del modelo C4, asegurando que estos elementos se visibilicen adecuadamente en los diferentes niveles (contexto, contenedores, componentes) sin perder la claridad del diseño?

Integrando Aspectos No Funcionales en el Modelo C4

El modelo C4 es una herramienta excelente para visualizar la arquitectura de software, pero para asegurar una implementación exitosa, es crucial incorporar los aspectos no funcionales desde el inicio del diseño. Estos aspectos, como la escalabilidad, seguridad y disponibilidad, son fundamentales para garantizar el éxito a largo plazo del sistema.

Cómo integrar estos aspectos en cada nivel del modelo C4:

1. Contexto

- Identificar dependencias externas: Definir cómo el sistema interactúa con otros sistemas externos y cómo estos afectan la escalabilidad, seguridad y disponibilidad.
- Establecer límites: Definir los límites del sistema y cómo se manejan los fallos en estos límites.
- Considerar el entorno: Evaluar el entorno de despliegue y cómo afecta a los aspectos no funcionales (por ejemplo, nube vs. on-premises).

2. Contenedores

- Definir responsabilidades claras: Cada microservicio debe tener una responsabilidad clara y bien definida. Esto ayuda a garantizar la escalabilidad y la facilidad de mantenimiento.
- Considerar el tamaño: El tamaño de los microservicios debe ser adecuado para permitir una escalabilidad independiente.
- Identificar puntos de falla: Identificar los puntos de falla potenciales en cada servicio y cómo se pueden mitigar.

3. Componentes

- Seleccionar tecnologías adecuadas: La elección de las tecnologías (bases de datos, frameworks, etc.) debe estar alineada con los requisitos no funcionales.
- Implementar patrones de diseño: Utilizar patrones de diseño como circuit breakers, bulkheads y timeouts para mejorar la resiliencia y la disponibilidad.
- Considerar la seguridad: Implementar medidas de seguridad a nivel de componente, como autenticación, autorización y cifrado.

4. Código

- Escribir código de alta calidad: Seguir buenas prácticas de programación para garantizar la calidad y la mantenibilidad del código.
- Realizar pruebas exhaustivas: Implementar pruebas unitarias, de integración y de extremo a extremo para verificar la funcionalidad y los aspectos no funcionales.
- Monitorear y analizar: Implementar mecanismos de monitoreo para rastrear el rendimiento, la seguridad y la disponibilidad del sistema.

Ejemplos concretos de cómo representar aspectos no funcionales:

- Escalabilidad:
 - Utilizar símbolos para indicar componentes que pueden ser escalados horizontalmente.
 - Incluir información sobre la capacidad de procesamiento y almacenamiento de cada componente.
- Seguridad:
 - Incluir firewalls, sistemas de detección de intrusiones y otras medidas de seguridad en los diagramas.
 - Utilizar colores o símbolos para indicar componentes críticos desde el punto de vista de la seguridad.
- Disponibilidad:
 - Mostrar redundancia de componentes críticos.
 - Incluir información sobre los tiempos de recuperación ante fallos (RTO).