

En **BIAN (Banking Industry Architecture Network)**, el desacoplamiento es un principio fundamental para lograr una arquitectura flexible, reutilizable y alineada con estándares de la industria. A continuación, se describen los **pilares del desacoplamiento en arquitectura de software en BIAN**:

1. Diagrama de Dependencia en BIAN

- BIAN se basa en un **modelo de referencia de dominios de servicio** (*Service Landscape*), donde cada **Service Domain** representa una capacidad de negocio autónoma.
- El diagrama de dependencia en BIAN identifica las **relaciones entre los Service Domains** y define cómo se comunican mediante **Service Operations**.
- Este modelo permite desacoplar servicios al definir **límites claros de responsabilidad**, asegurando que cada dominio opere de manera independiente.
- **Ejemplo en BIAN:**
 - *Card Authorization* depende de *Card Billing*, pero su interacción es mediante una API bien definida, evitando dependencias directas entre sistemas monolíticos.

2. Modelo Ontológico en BIAN

- BIAN utiliza un **modelo ontológico** basado en la representación semántica de datos y procesos bancarios.
- Se fundamenta en:
 - **Business Object Model (BOM):** Define los objetos de negocio clave y sus atributos estándar.
 - **Behaviour Qualifiers (BQs):** Detallan los subcomportamientos dentro de cada Service Domain.
- Este modelo permite que las aplicaciones bancarias utilicen una misma estructura de datos sin estar acopladas a una implementación específica.
- **Ejemplo en BIAN:**
 - Una API de *Loan Management* en un banco A y en un banco B pueden compartir la misma estructura de datos basada en el BOM de BIAN, facilitando la interoperabilidad sin estar acopladas a sistemas específicos.

3. Contrato de Artefactos en BIAN

- BIAN promueve el uso de **contratos de servicio estandarizados**, permitiendo que diferentes sistemas interactúen sin depender de una implementación específica.
- Estos contratos incluyen:
 - **Service Domain Interface:** Define los métodos expuestos por un dominio de servicio.
 - **Service Operations:** Especificaciones estandarizadas para la comunicación

entre dominios.

- **Semantic APIs:** APIs alineadas con la ontología BIAN, evitando acoplamiento a nivel de infraestructura.
- **Ejemplo en BIAN:**
 - Una API de *Customer Offer* define un contrato que puede ser consumido por múltiples aplicaciones sin importar su tecnología subyacente.

Conclusión

El **desacoplamiento en BIAN** se logra mediante:

1. **Diagrama de Dependencia:** Para entender y gestionar relaciones entre dominios de servicio.
2. **Modelo Ontológico:** Para definir estructuras de datos estandarizadas y evitar dependencias tecnológicas.
3. **Contrato de Artefactos:** Para garantizar interacciones bien definidas entre componentes, asegurando interoperabilidad.

Estos pilares permiten a los bancos adoptar una arquitectura más modular, flexible y preparada para la evolución tecnológica. □

Arquitectura BANK



Si te ha interesado este artículo y deseas un apoyo o asesoría en algún requerimiento, envíanos un mensaje a: info@arquitecturabank.com