Sesión 02

Command and Query Responsability Separation (CQRS)

ÍNDICE

Command Query Separation (CQS)

CQR

Bertrand Meyer

Libro: “Object Oriented Software Construction”

Principio de DISEÑO

2000

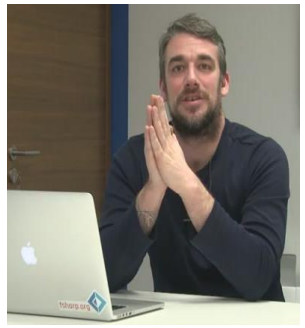
Command and Query RESPONSABILITY Separation (CQRS)

CQRS

Greg Young

Papper: CQRS Documents
(https://cqrs.wordpress.com/)

2010



2024

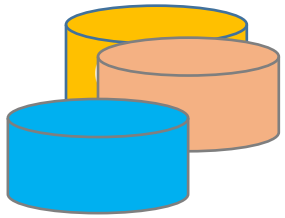
Evolución de CQR -> CQRS

Es un ESTILO arquitectónico, permite cómo organizar los diferentes componentes de una aplicación de manera que sea OPTIMIZADA y segura para lograr objetivos específicos(mantenimiento, RENDIMIENTO, velocidad de desarrollo, calidad, reutilización, etc.).

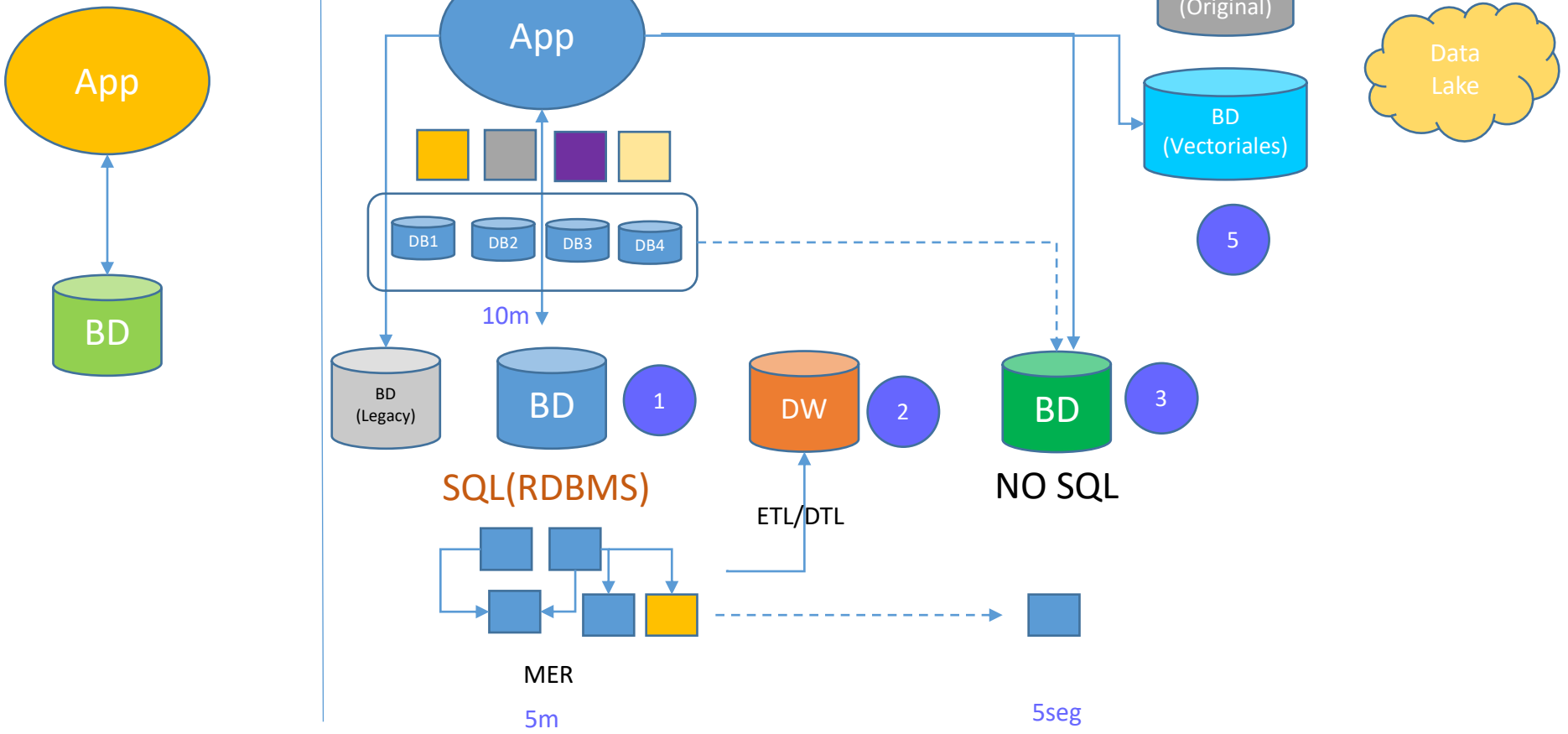
No existe un estilo de arquitectura que OPTIMICE todos los componentes de una aplicación, por lo que es necesario realizar un balance (trade-off).

Command and Query Responsibility Segregation (CQRS)

Databases

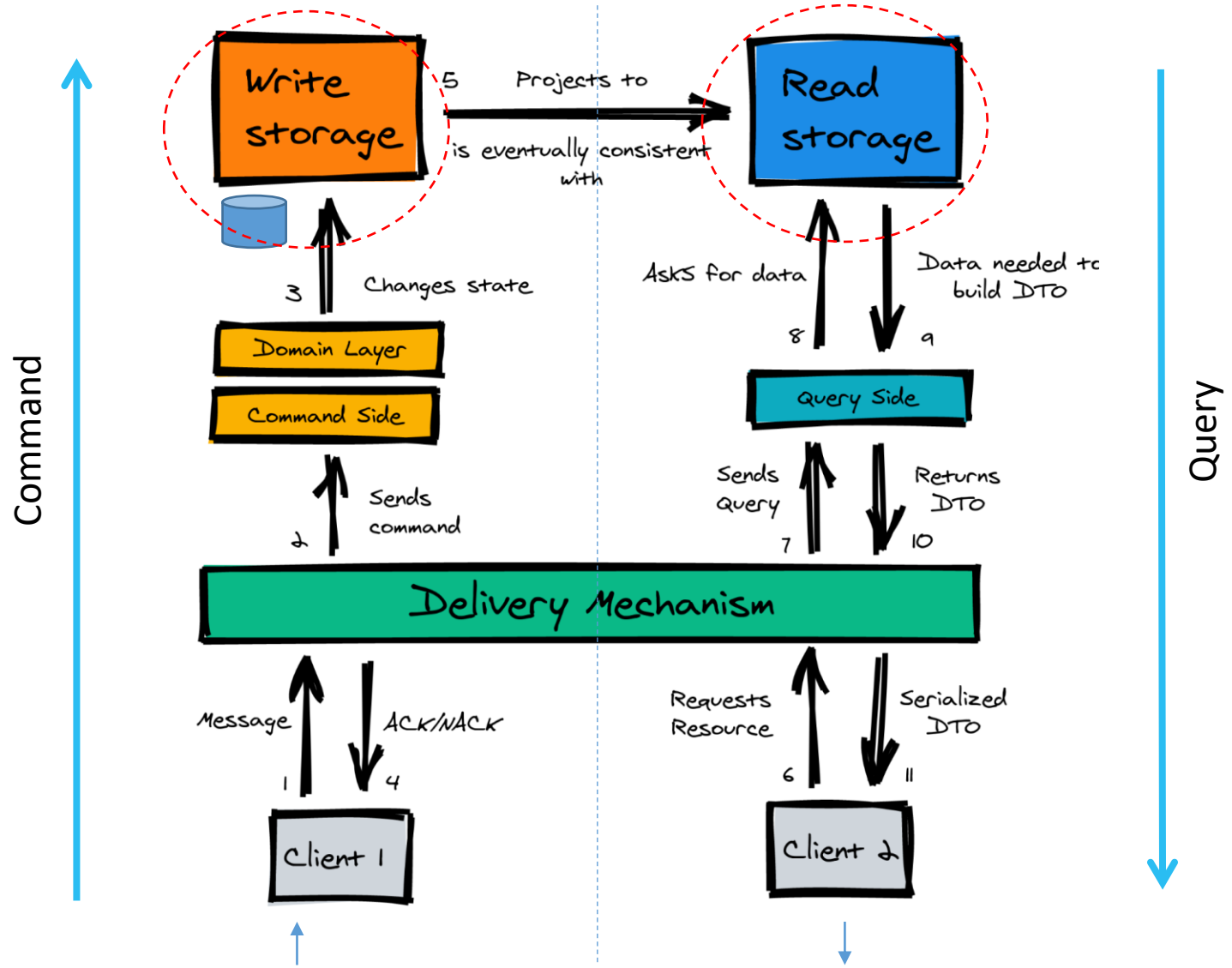


1. BD SQL (RDBMS) – Oracle, MS SQL Server, DB2,… Postgre, MySQL, MariaDB, ..
2. BD NOSQL (MongoDB, Casandra, DynamoDB…)
3. DB Memoria (~~H2~~, Redis, Memcached, …)
4. DB Mesajes (Kafka, RabbitMQ, AWS SQS…)
5. DB Logs (ELK, EFK …)
6. DB Contenidos (Alfresco, Laserfiche, Oracle Document, S3…)
7. DB Grafos
8. DB Raw


App
BD
(N aplicación)
App
4
BD
(Original)
Data
Lake
BD
(Vectoriales)
5
DB1
DB2
DB3
DB4
10m
BD
(Legacy)
BD
1
DW
2
BD
3
SQL(RDBMS)
ETL/DTL
NO SQL
MER
5m
5seg

Commands	Solicitudes que representan la acción que el usuario desea realizar con todos los parámetros requeridos.	Commands Handler	Mecanismo responsable de implementar la lógica de negocio para un comando específico y su persistencia en base de datos – **SQL** (autorización y rendición)
Queries	Solicitudes que representan qué información desea obtener el usuario y los parámetros necesarios para obtenerla.	Query Handler	Mecanismo encargado de acceder a una preparación de la información que el usuario desea obtener para una consulta – **NOSQL-** específica (autorización y rendición)
Domain Events	Mensajes que representan un suceso en el dominio y acciona componentes CQRS. Son activados por comandos y son utilizados para sincronizar modelos de escritura y lectura.	Projections	Modelos de lectura especializados, persistentes y óptimos generados para cada consulta
Projectors	Procesos que generan las proyecciones	Process Managers (Event Handlers)	Procesos que escuchan Domain Events y activan comandos para orquestar procesos de larga ejecución.

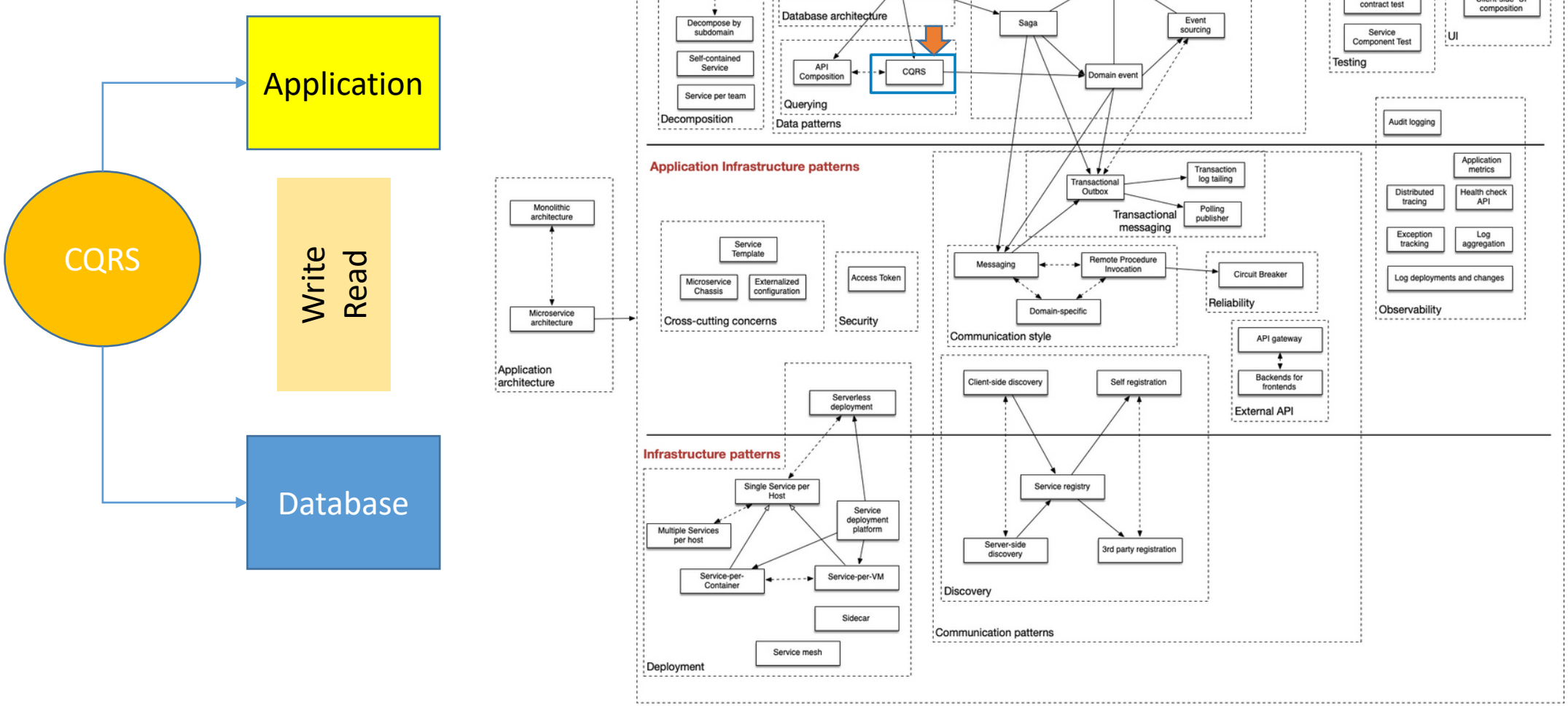
Componentes CQRS



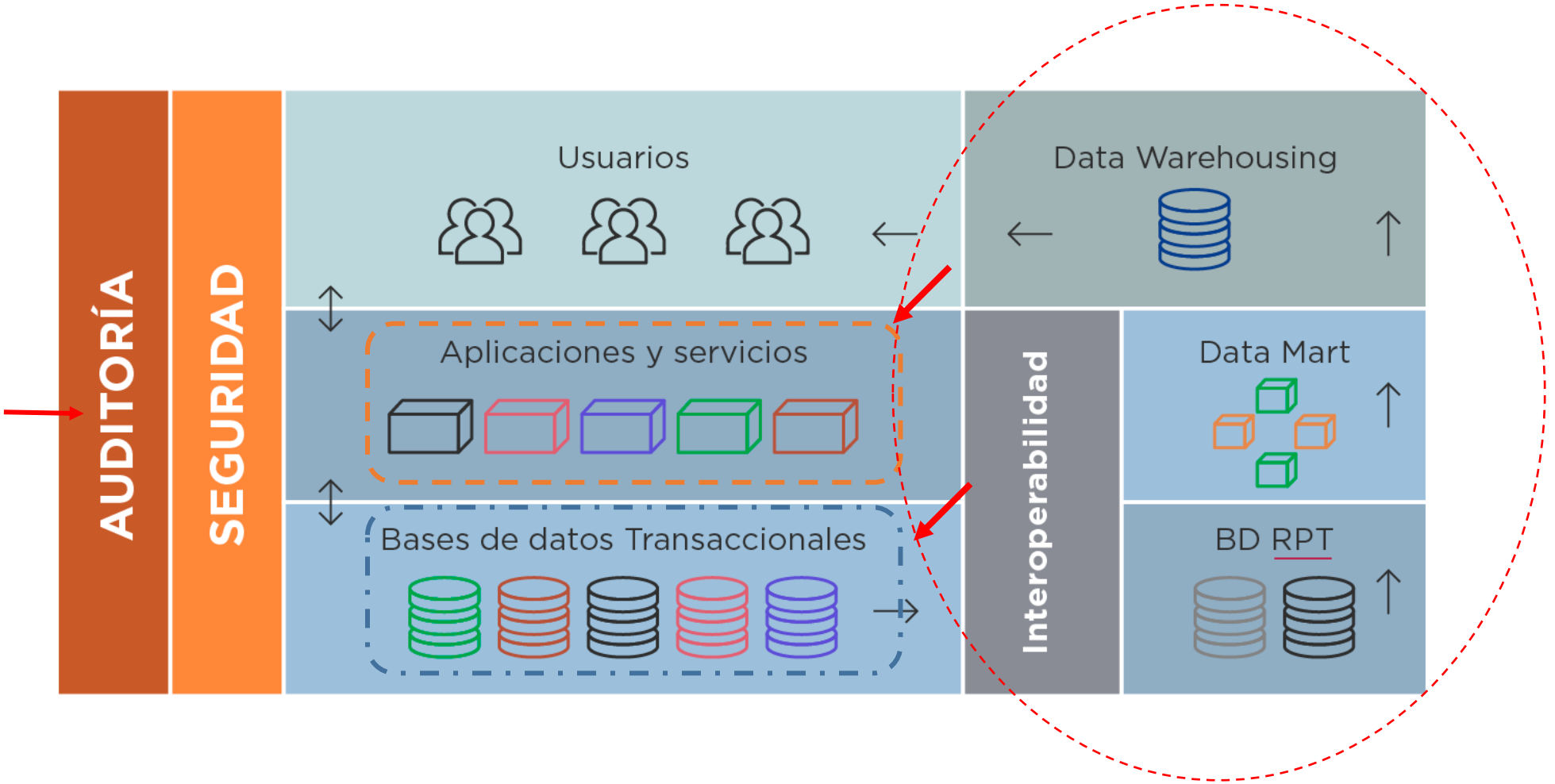
Flujo de CQRS

	Consistencia	Almacenamiento	Escalabilidad
Modelo de Escritura	Consistencia inmediata	Los datos frecuentemente es almacenada en forma normalizada, bases de datos relacionales (SQL)	Número pequeño de transacciones
Modelo de Lectura	Consistencia eventual	Los datos se pueden desnormalizar cuando se registran, para optimar los costos de consulta. Base de datos NO SQL	Gran número de transacciones, la escalabilidad es crítica

Modelos CQRS

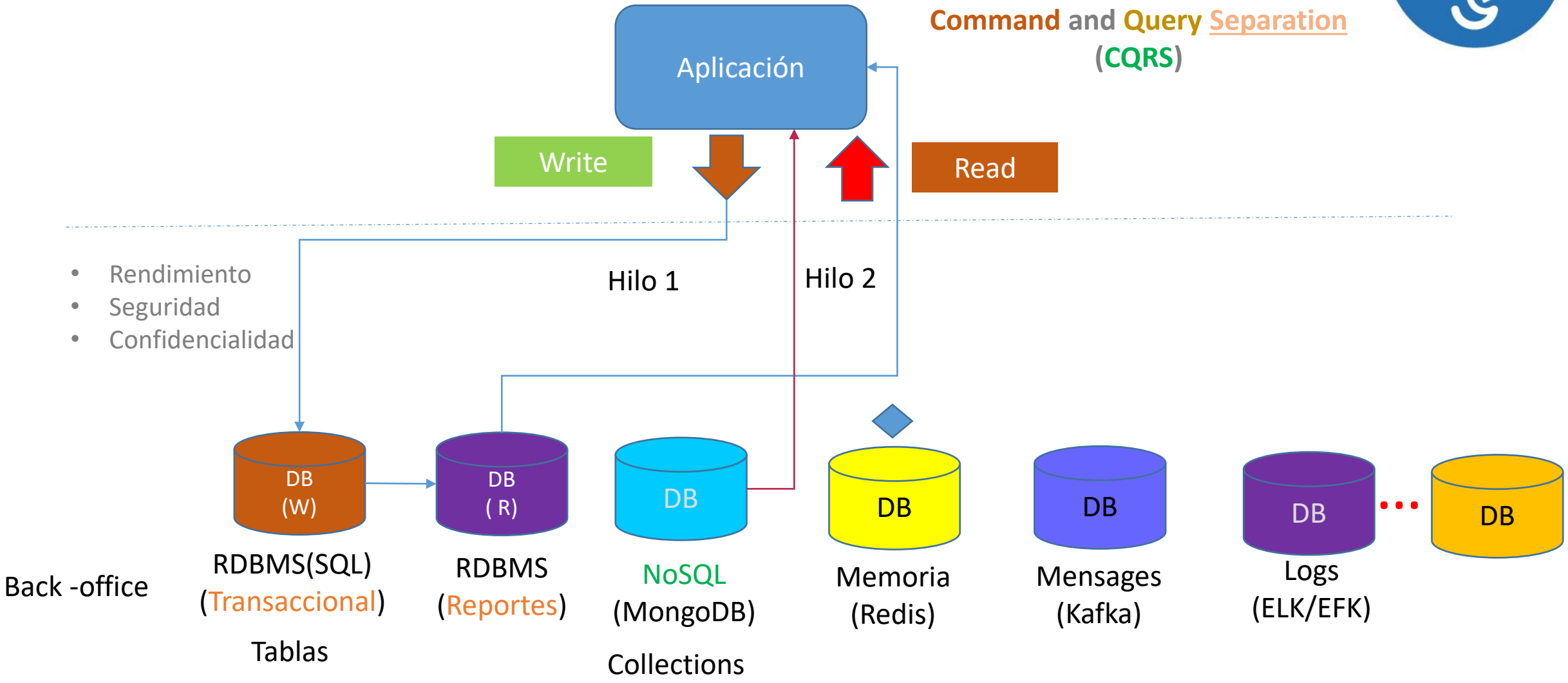


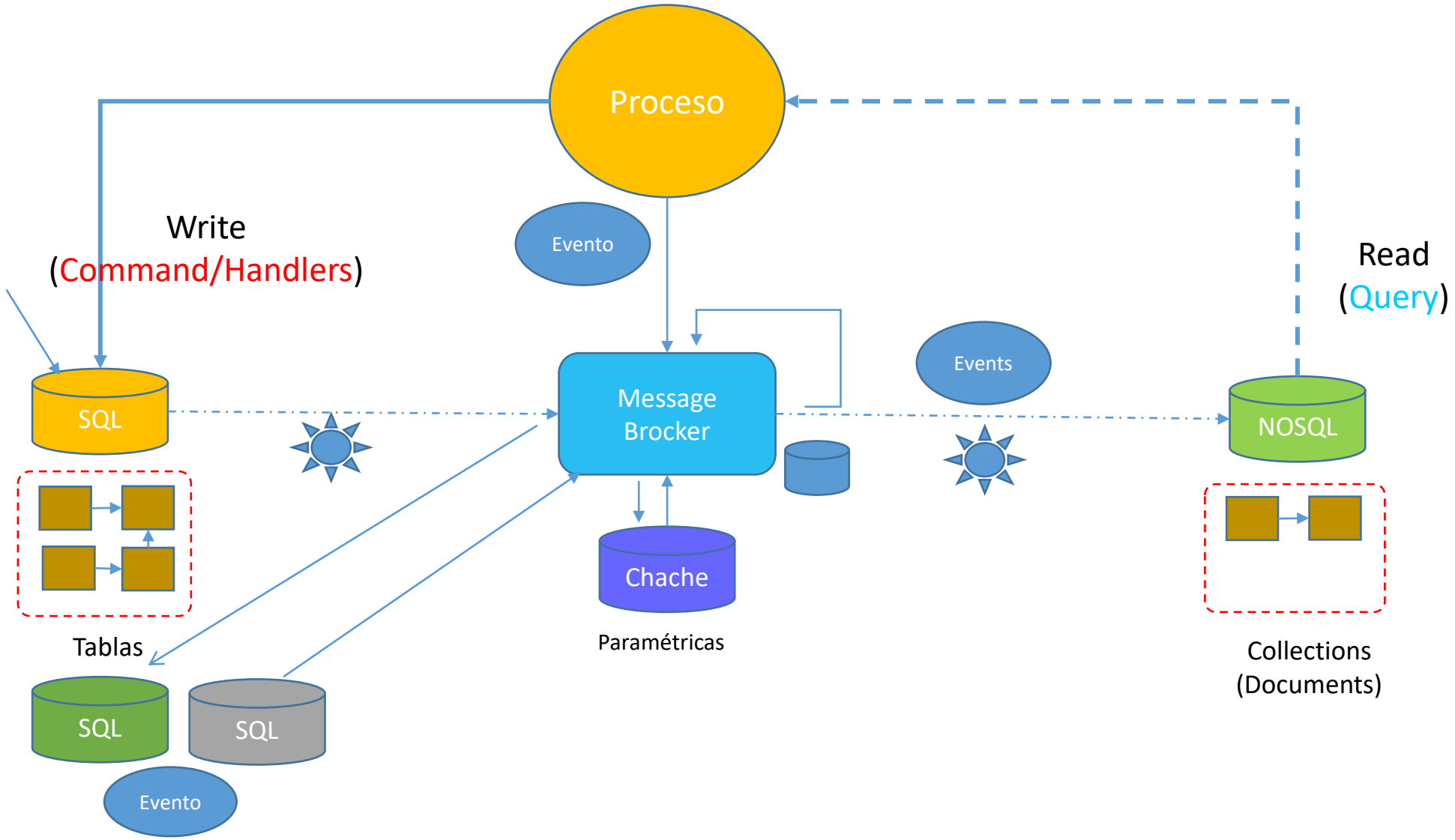
CQRS vs Microservices Architecture Patterns



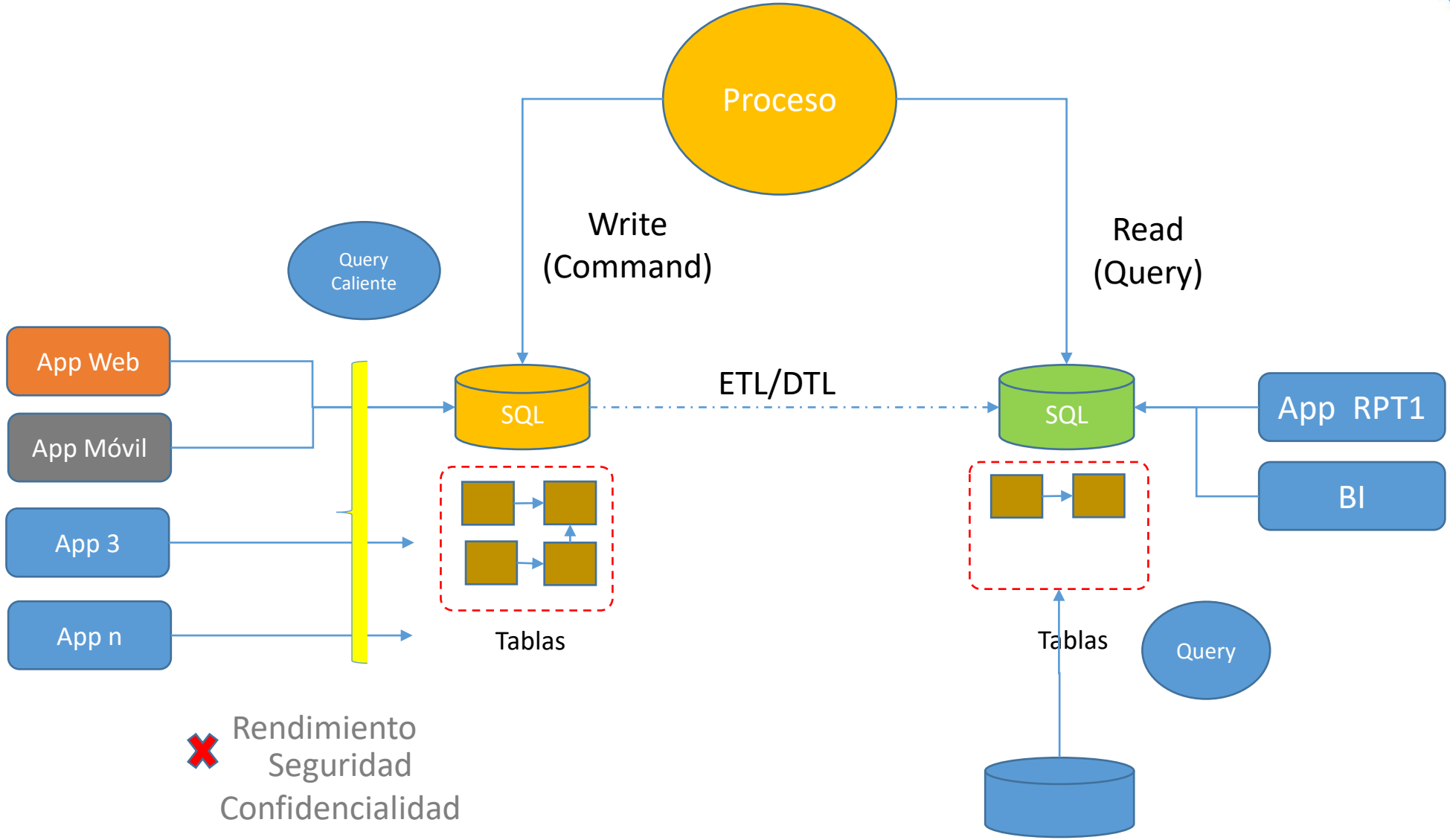
Arquitectura Referencial

Command and Query Separation (CQRS)



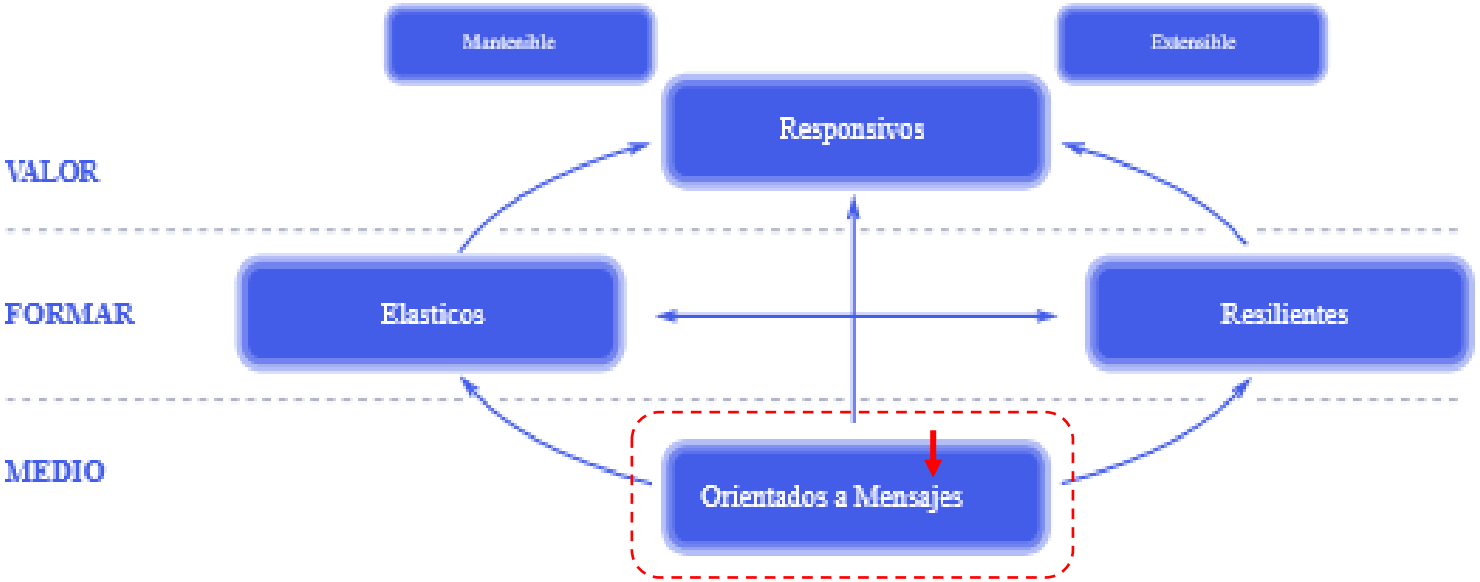


CQRS Arquitecture: SQL, NoSQL, Cache & Message Brocker



CQRS Arquitecture: SQL Clasic

https://www.reactivemanifesto.org/es



Reactivos

Reactive Arquitecture

GALAXY

TRAINING