Service-Based Architecture

CSSE6400

Richard Thomas

March 21, 2022

Definition 1. Distributed System

A system with multiple components located on different machines that communicate and coordinate actions in order to appear as a single coherent system to the end-user.

Quote

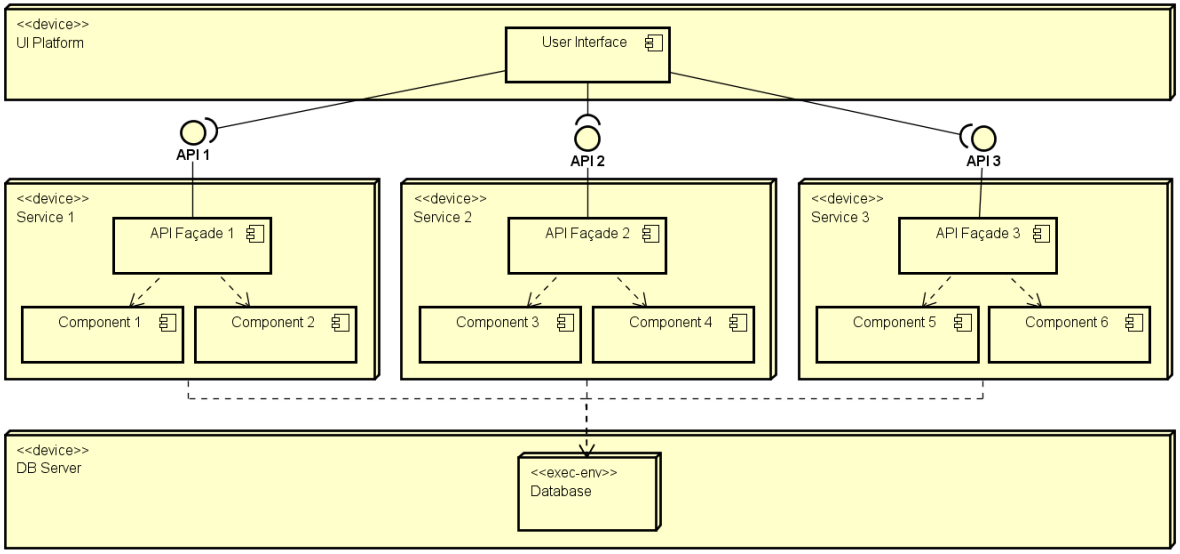
A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.

– Leslie Lamport [Turing Award, 2013]

Definition 2. Service-Based Architecture

System is partitioned into business domains that are deployed as distributed services. Functionality is delivered through a user interface that interacts with the domain services.

Service-Based Architecture



Terminology

User Interface Provides access to system functionality

Services Implement functionality for a single, independent business process

Service APIs Communication mechanism between UI and each service

Database Stores persistent data for the system

Definition 3. API Abstraction Principle

Services should provide an API that hides implementation details.

Definition 4. Façade Design Pattern

Provide a simple, abstract interface to use a service domain's functionality. A component within the service coordinates how to deliver the requested functionality with the service's internal components.

Definition 5. Independent Service Principle

Services should be independent, with no dependencies on other services.

Question

What are the consequences of having a shared database?

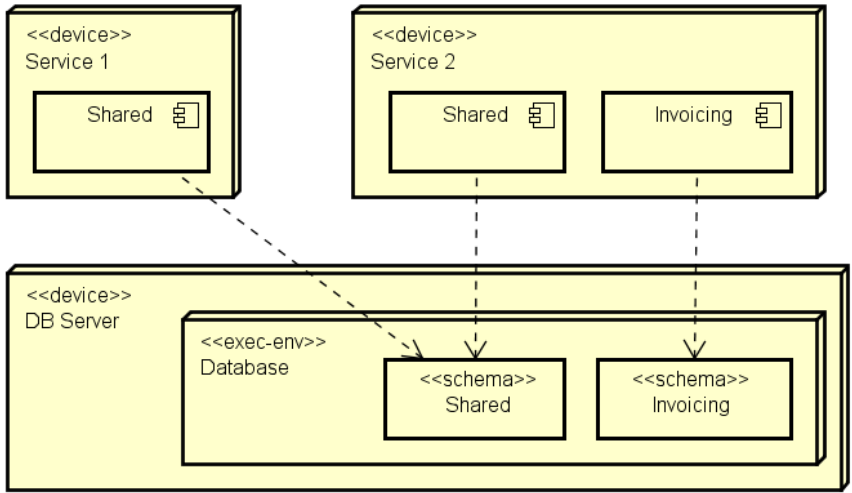
Question

What are the consequences of having a shared database?

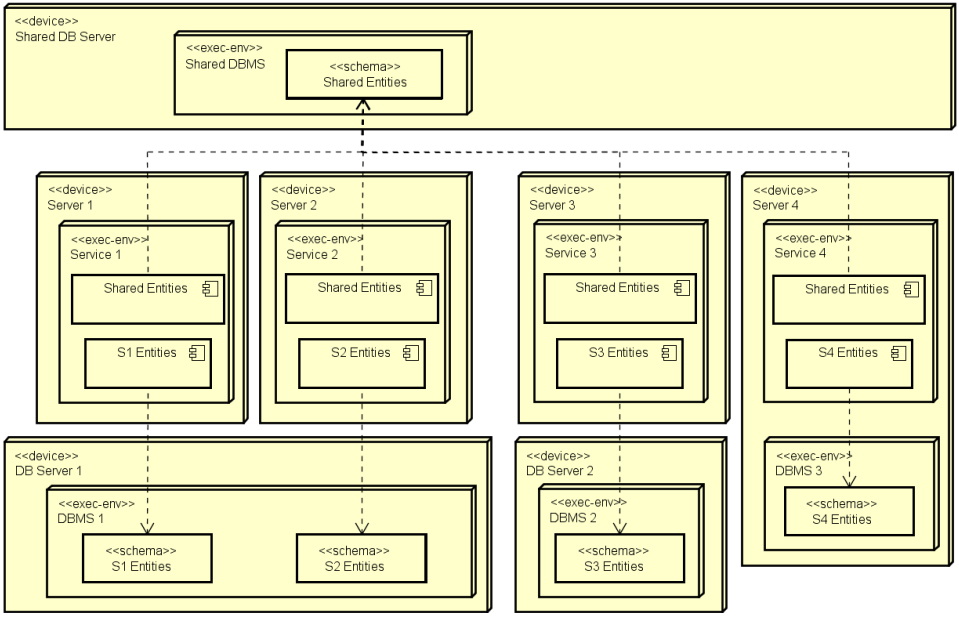
Answer

Increased *data coupling*.

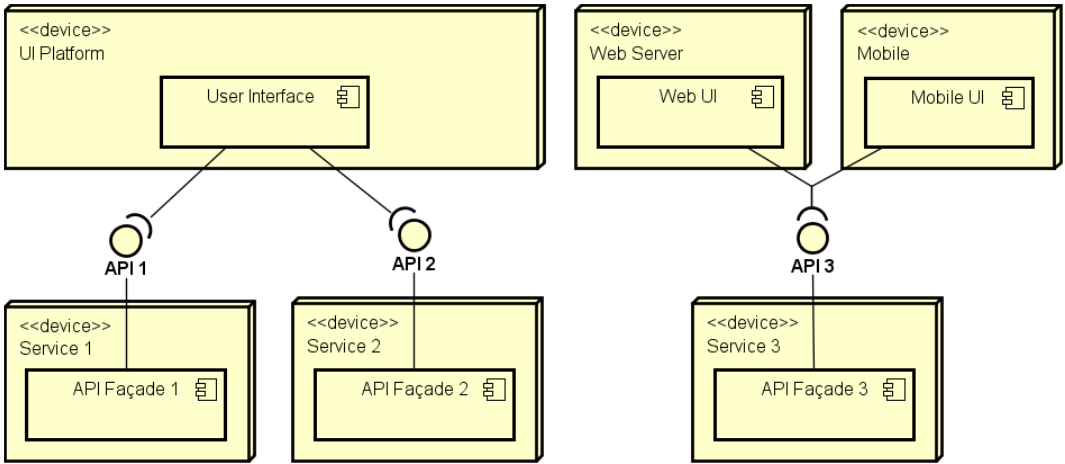
Logical Partitioning of Persistent Data



Separate Databases



Separate UIs



Oi!

Example REST API?
Issue of failures in distributed systems?
Show service architecture for Sahara and summarise parallel flows through services.
Aside about stateless services and possible multi-threaded web pages (e.g. JSF).

Pros & Cons

Simplicity Core system & Plug-in interface 

Extensibility Plug-ins 

Interoperability Plug-ins 

Scalability 

Reliability