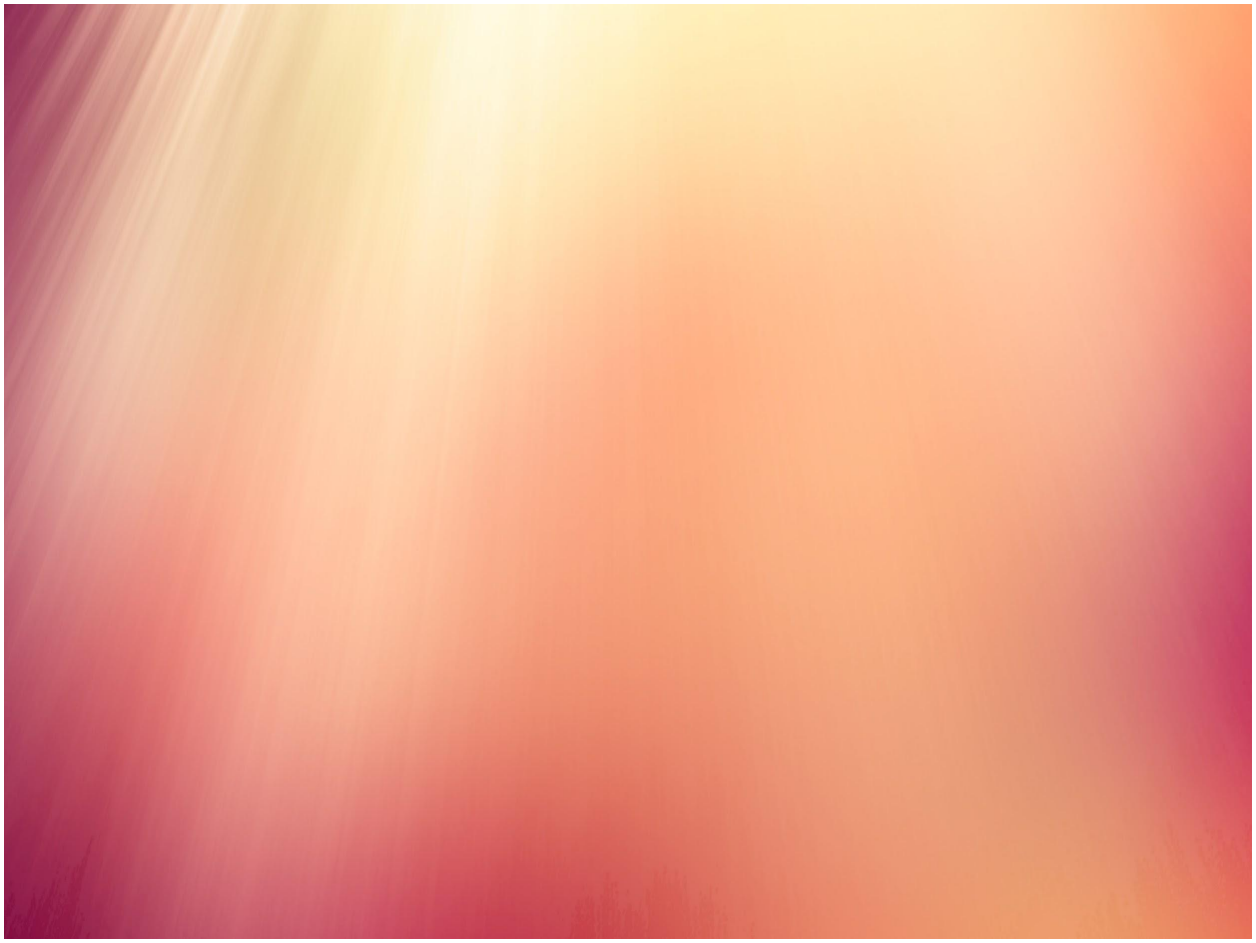
Design document

for the service



Overview

The system is a collection of 3 microservice which facilitate books as content applications. It is expected to communicate with the application through a RESTful API and should follow a Microservice-based design.

Assumptions

1. 1. The application is expected to be under minimal stress and a low number of users given the scope of the project.
2. 2. The system is not supposed to be production-ready and is not rigorously tested hence one can face some unintended behaviours or errors.
3. It is also assumed that the users will be closed group testers.
4. The security of the application is not taken under consideration.

Tech stack

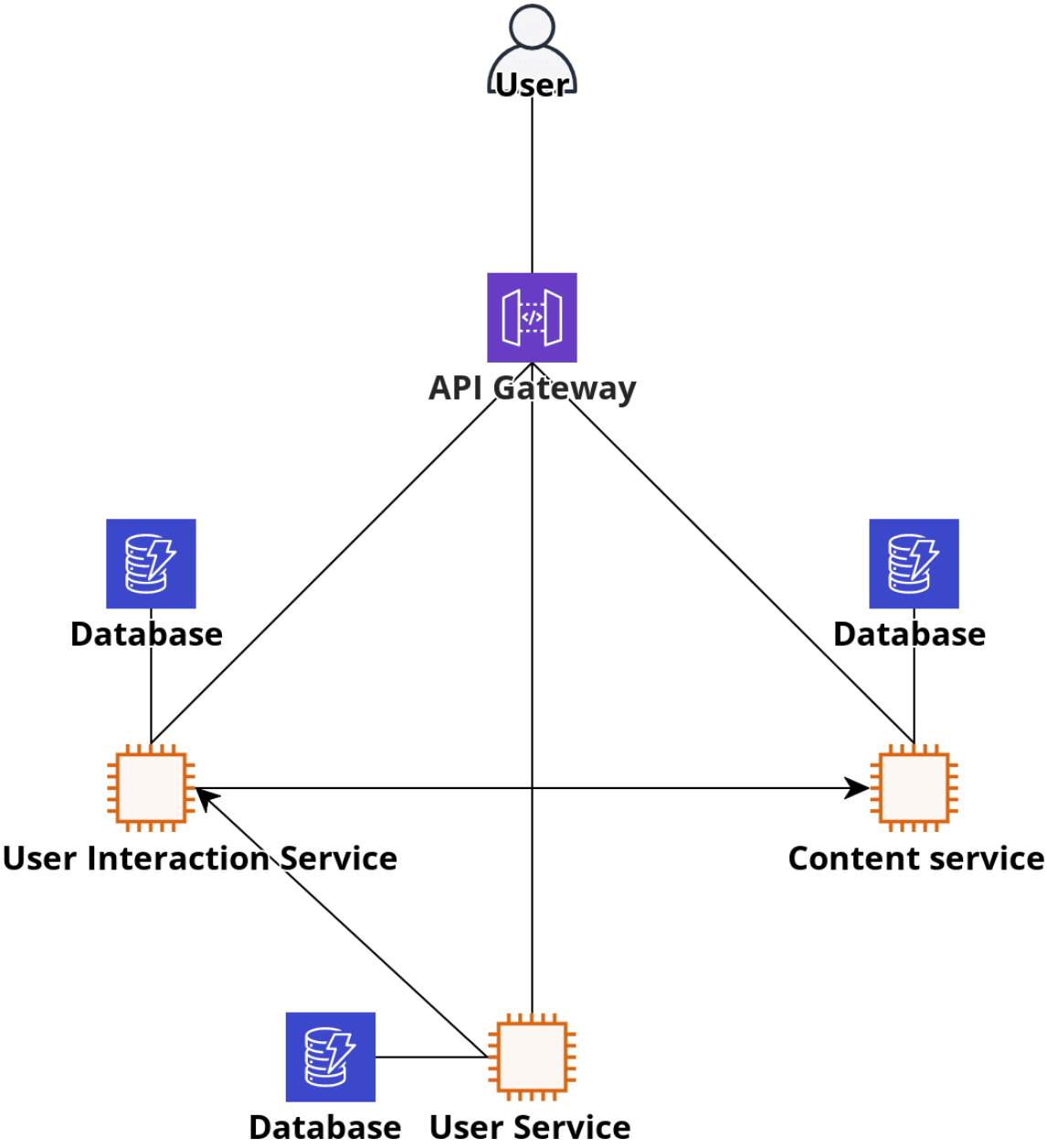
1. Programming language - Node js https://nodejs.org/
2. Database - DiskDb
   Given the scope of the project DiskDB, a simple JSON based database was chosen to reduce overhead and development time https://github.com/arvindr21/diskDB

System Design

Services

1. Content Service - Serves books as content. Has API endpoints for Performing CRUD operations, Book data ingestion through CSV files and Sorting books on the basis of interaction and date.
2. User Service - Handles users. Has API endpoints for CRUD operation on user data.
3. User Interaction Service - Handles user interactions like ‘read’ and ‘like’ while performing User Validation.

API Gateway - exposes unified API to the application.


User
API Gateway
Database
Database
User Interaction Service
Content service
Database
User Service

Database Schema

CONTENT SERVICE DB	USER INTERACTION SERVICE DB	USER SERVICE DB

{
"title": string,
"story": string,
"last_modified":datetime,
"_id": multichar
}
	{
	"story_id": multichar,
	"like": int,
	"read": int,
	"_id": multichar
	}
		{
		"fname": string,
		"lname": string,
		"email": string,
		"phno": int,
		"_id": multichar
		}

Class diagram