I

Design Document

OepnVidStreamer

by Radoslav Radev

Advanced Software Engineering Semester 6

27/02/2024

ver. 1.0

Document Versioning

Date	**27/02/2024**
Version	V1
State	Initial draft
Author	Radoslav Radev

Contents

1. Introduction

1.1 General Information

This Design Document serves as the technical and architectural blueprint for the creation of OpenVidStreamer The document will cover:

- User stories
- Architectural Design

1.2 Architectural Approach

The architectural approach for the OpenVidStreamer project employs Agile methodology, emphasizing iterative development and delivery through a series of sub-goals including backend architecture, system development, frontend SPA design, scalability integration, and comprehensive testing for functionality, quality, and performance, culminating in cloud deployment.

1.3 Project Methodology

The methodology employed in this project is Agile, and consistent with the approach outlined in the Project Plan.

Agile methodology:

"a software development and project management method that is continuous, and aids teams in delivering value to clients more quickly and with fewer hassles"

1.4 Frameworks & Technologies

Frameworks & Technologies used in this project are as follows:

- **.NET & .NET ASPIRE - Backend Micro Service system**
  - **RabbitMQ - message broker (inter-service communication)**
  - **Ocelot - API Gateway**
  - **Consul - MicroService management and Discovery**
  - **SignalR - Async Communication with the client**
  - **Entity Framework - ORM**
- **Kubernetes - MicroService scalability**
- **React - Front-end**

1.5 Feature Prioritization

Feature prioritization will follow the guidelines below:

- ***MUST – "Feature must be implemented. It has the highest priority."***
- ***SHOULD – "Feature should be implemented if everything goes smoothly and by schedule. It has medium priority."***
- ***COULD – "Feature could be implemented if there is time upon finishing the project. It has lowest priority."***

2. User Stories

A user story is a comprehensive explanation of a software feature written from the viewpoint of the client or end user.

Difficulty of the user stories will be measured on a scale from 1-10

User story I

Title	Priority	Difficulty
Login	**MUST**	**3**
As a customer, **I want to login into the platform so I can watch videos**		
Acceptance Criteria		
Given that the user provided valid login credential information, **And The user has an active subscription to the platform** **When the Log-In button, in the Login component of the website, is pressed** **Then The system Logs in the user and redirects to the home page in case of success** **No/Insufficient information entered:** - error message displaying which data is incorrect.		

User story II

Title	Priority	Difficulty
Register	**MUST**	**3**

As a customer,
I would like to create an account in the platform,
So I can get access to it

Acceptance Criteria

Scenario 1: Displaying the Registration Form

Given that I am a visitor to the platform without an existing account,
And I have navigated to the registration area on the platform,
When I choose to create a new account,
Then the system presents me with a registration form to enter my chosen login information (e.g., email, password).

Scenario 2: Successful Account Creation

Given that I am on the registration page,
And I have entered all required information correctly in the registration form,
When I press the "Register" button,
Then my account is created successfully,
And I am redirected to the account page where I can choose to activate a subscription.

Scenario 3: Failed Account Creation due to No/Insufficient Information

Given I am on the registration page,
When I leave required fields empty or enter insufficient information and press the "Register" button,
Then the system displays an error message indicating which fields are filled incorrectly or are missing.

Scenario 4: Failed Account Creation due to Duplicate Information

Given I am on the registration page,
And I have entered an email address already associated with an existing account,
When I attempt to submit my registration,
Then the system displays an error message indicating that the email address is already in use

User story III

Title	Priority	Difficulty
Activate Subscription	**Must**	**5**

As a customer,
I want to activate(pay) a subscription for the platform

Acceptance Criteria

Scenario 1: Displaying the Payment Portal

Given that I am a registered user without an active subscription,
And I have navigated to the subscription activation area on the platform,
When I choose to activate a subscription,
Then the system presents me with a payment portal where I can enter my debit/credit card information to purchase a subscription.

Scenario 2: Successful Payment Transaction

Given I am on the payment portal page,
And I have entered valid debit/credit card information,
When I submit my payment information,
Then the payment is processed successfully,
And I am redirected to a confirmation page with details of my subscription.

No/Incorrect information entered:

- error message displaying that the transaction has failed, and the reason why

User story IV

Title	Priority	Difficulty
Upload Content	Must	7

As a customer,
I want to contribute to the platform by uploading a video.

Acceptance Criteria

Scenario 1: Accessing the Upload Portal
Given I am a registered user with permissions to upload content,
And I have navigated to the "Upload" page on the platform,
When I access this page,
Then the system presents me with an upload portal where I can select a video file for upload and enter the accompanying metadata (Title, Description, Category).

Scenario 2: Successful Video Upload and Metadata Submission
Given I am on the upload portal page,
And I have selected a video file and entered all required metadata correctly,
When I press the "Upload" button,
Then the system uploads the video to the Video Library,
And makes it available for consumption by all users,
And redirects me to a confirmation page with details of my upload or to my uploaded video.

Scenario 3: Failed Upload due to No/Incorrect Information
Given I am on the upload portal page,
When I attempt to upload a video without selecting a file or entering

required metadata (Title, Description, Category) correctly,
Then the system displays an error message indicating that the upload has failed,
And specifies which information is missing or incorrect, prompting me for correction.

Scenario 4: Failed Upload due to Unsupported Video Format
Given I am on the upload portal page,
And I have selected a video file in an unsupported format,
When I attempt to submit the upload,
Then the system displays an error message indicating that the video format is not supported,
And suggests compatible formats.
Scenario 5: Upload Restrictions and Guidelines
Given I am on the upload portal page,
When I begin the upload process,
Then the system clearly displays any upload restrictions (e.g., file size, length, content guidelines) and guidelines to help ensure successful uploads.

User story V

Title	Priority	Difficulty
Video Recommendations	COULD	8
As a customer, I want to be presented with recommendations on videos, based on my previous, watch statistics.		
Acceptance Criteria		
Given I am a logged-in user, And I have a history of watched videos on the platform, And I have the home page open, When the page loads, Then the system presents me with a section of personalized video recommendations, And these recommendations are based on my previous watch history, taking		

into account factors like viewed genres, liked videos, and watch time.

User story VI

Title	**Priority**	**Difficulty**
Video Playback & Player Page	**SHOULD**	**9**
As a customer, **I want to watch and iteract with a video I selected.**		
Acceptance Criteria		
Scenario 1: Accessing the Video Player **Given I am a logged-in user on the home page,** **And I have selected a video from the list of available content,** **When I click on the video thumbnail or title,** **Then the system navigates me to a dedicated video playback page,** **And the video player automatically loads the selected video with**		

playback controls (play, pause, volume, full screen, and video quality settings).

Scenario 2: Interacting with the Video (Liking/Disliking)

Given I am on the video playback page watching a video,

When I choose to interact with the video by selecting either the like or dislike button,

Then my selection is registered in the system,

And the like or dislike count updates accordingly,

And this interaction influences my personalized recommendations in the future.

Scenario 3: Consuming Video Details

Given I am watching a video (user is on a video playback page),

When the video is playing or paused,

Then I can view additional video details including the title, description, number of views, upload date.

2.1 RESOURCE MAPPING

Assuming we have 100,000 currently active users on the platform, for some operations we will assume 100,000 hourly active users (spreading the load over 1 hour period)

- *Video Playback (Streamer):*

  my HLS Videos are split into 10 second chunks, assuming the worst scenario that 100,000 currently active users watching videos

We will make the calculations on 1 hour bases (every single user will be watching a 1 hour video)

1. *Chunk Requests Calculation:*
    - *Total chunks per hour for one user =* $\frac{3600 \text{ seconds/hour}}{10 \text{ seconds/chunk}}$ *= 360* chunks for each user per hour.
2. *Total Chunks Per Hour for All Users:*
    - *360×100,000 users=36,000,000 chunks per hour*
3. *Chunks Per Second:*
    - *Dividing the total number of chunks by the number of seconds in an hour gives us the chunk requests per second:*
    - $\frac{36{,}000{,}000 \text{ chunks/hour}}{3600 \text{ seconds/hour}}$ *=10,000 chunks per second*

- *Video Uploads (Upload):*
    - *Assuming 5000 videos are uploaded per day*
    - *For 86,400 seconds per day, the load for video uploads = 5000 / 86400≈0.057 uploads per second.*

- *Login/Account Activities (Account):*
    - *Assume about 5% of active users log in, log out, or manage their account settings per hour (considering that many stay logged in).*
    - *5% of 100,000 = 5,000 account activities per hour.*
    - *For 3600 seconds per hour, the load for account activities = 5,000 / 3600 ≈ 1.39 account activities per second.*

- *Video Recommendations (RecommendationAlgo):*
  - *Assume every user triggers the recommendation algorithm 3 times per hour (once every 20 minutes).*
  - *That's 100,000 * 3 = 300,000 recommendations per hour.*
  - *For 3600 seconds per hour, the load for recommendations = 300,000 / 3600 ≈ 83 recommendations per second.*

- *Pay/Activate Subscription:*
  - *if we have 1, 000,000 monthly paying users and all of them pay their subscription the last 5 days of the month (Subscription is valid for the next 30 days from the date of purchase)*
  - *Total seconds in 5 days = $5\ days \times 24\ hours \times 3600\ seconds = 432{,}000\ seconds$*
  - *$1{,}000{,}000/432{,}000 = 2.31\ activations\ per\ second$*

Load Estimation Table:

Api-Gateway is Hit in Every Request

The FrontEnd Is fetched only on the first page load, with some exceptions

User Story	Estimated Load Per Second	Frontend	Account	Recommendation Algo	Rendeer	Streamer	Upload	Video library
Login	1.39 rps	Partial	Yes	No	No	No	No	No
Register	1.39 rps	Partial	Yes	No	No	No	No	No
Activate Subscription	2.31 rps (last 5 days of the	Partial	Yes	No	No	No	No	No

	month)							
Upload Content	0.057 rps	Partial	No	Yes	Yes	No	Yes	Yes
Video Recommendations	83 rps	Partial	No	Yes	No	No	No	Yes
Video Playback & Player	10,000 cps	Partial	No	No	No	Yes	No	No

3. Technical Architecture

3.1 API URL tables

TO BE ADDED via swagger in later document revisions

3.2 How is SOLID guaranteed?

S– Every class has only one responsibility.
O– Classes are open for extension but closed for modification
L– If class A is a subtype of class B, we should be able to replace B with A
I – Proper use of interfaces and abstract classes
D– Dependency injection and inversion are by the Spring Been system

3.3 Framework selection:

The frameworks used in this project are as followed:

- **.NET & .NET ASPIRE - Backend Micro Service system**
- **Kubernetes - MicroService scalability**
- **React - Front-end**

Why do I use .NET technologies?

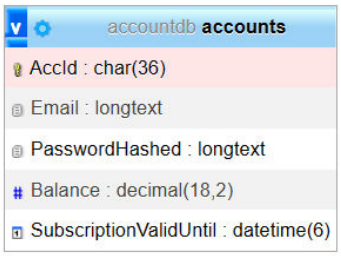
For this project, I chose .NET because it offers a significantly better developer experience compared to Java and Spring, which have lagged in modern features, data structures, and overall framework capabilities for the past decade. .NET's comprehensive library and robust framework provide all the necessary tools and components out of the box, enabling a more streamlined and efficient development process. This is in stark contrast to Java, where developers often face challenges due to missing features and a less cohesive ecosystem.

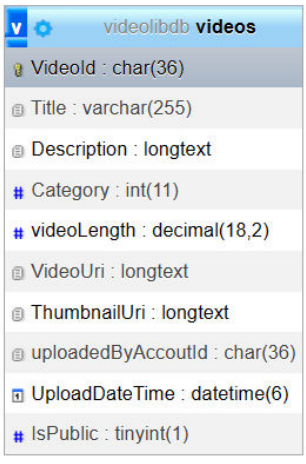
.NET also stands out in terms of performance. It has been consistently proven to handle computational tasks more efficiently, offering superior speed and scalability. This performance edge is critical for my project's requirements, ensuring that we can manage high loads and complex operations without compromise.

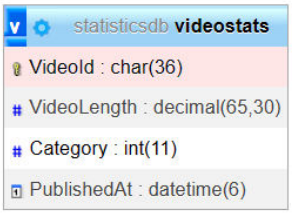
The design and configurability of .NET further cemented my choice. Unlike Spring's complex and error-prone reflection-based system and its convoluted configuration approach, .NET provides clear and direct control over the application's setup. This allows for precise tuning and customization, ensuring our system is both stable and optimized for performance.

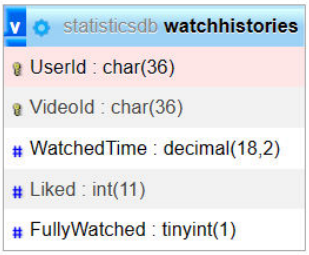
For a deep dive into the different components chosen for this project, please refer to the Research Paper.

3.4 Data Distribution







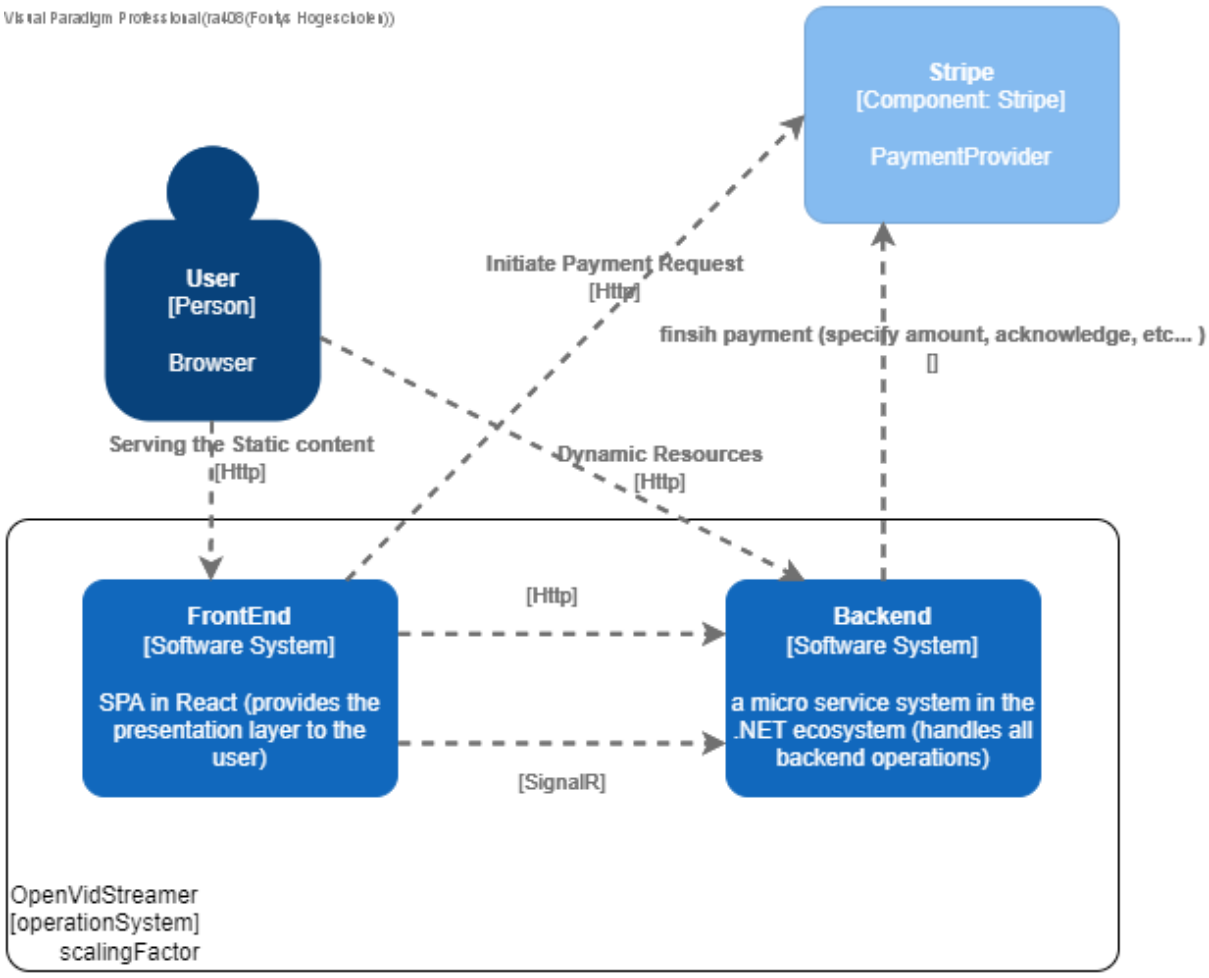


The current Design is "Database per Service" approach, but specifically emphasizing decentralized data management.

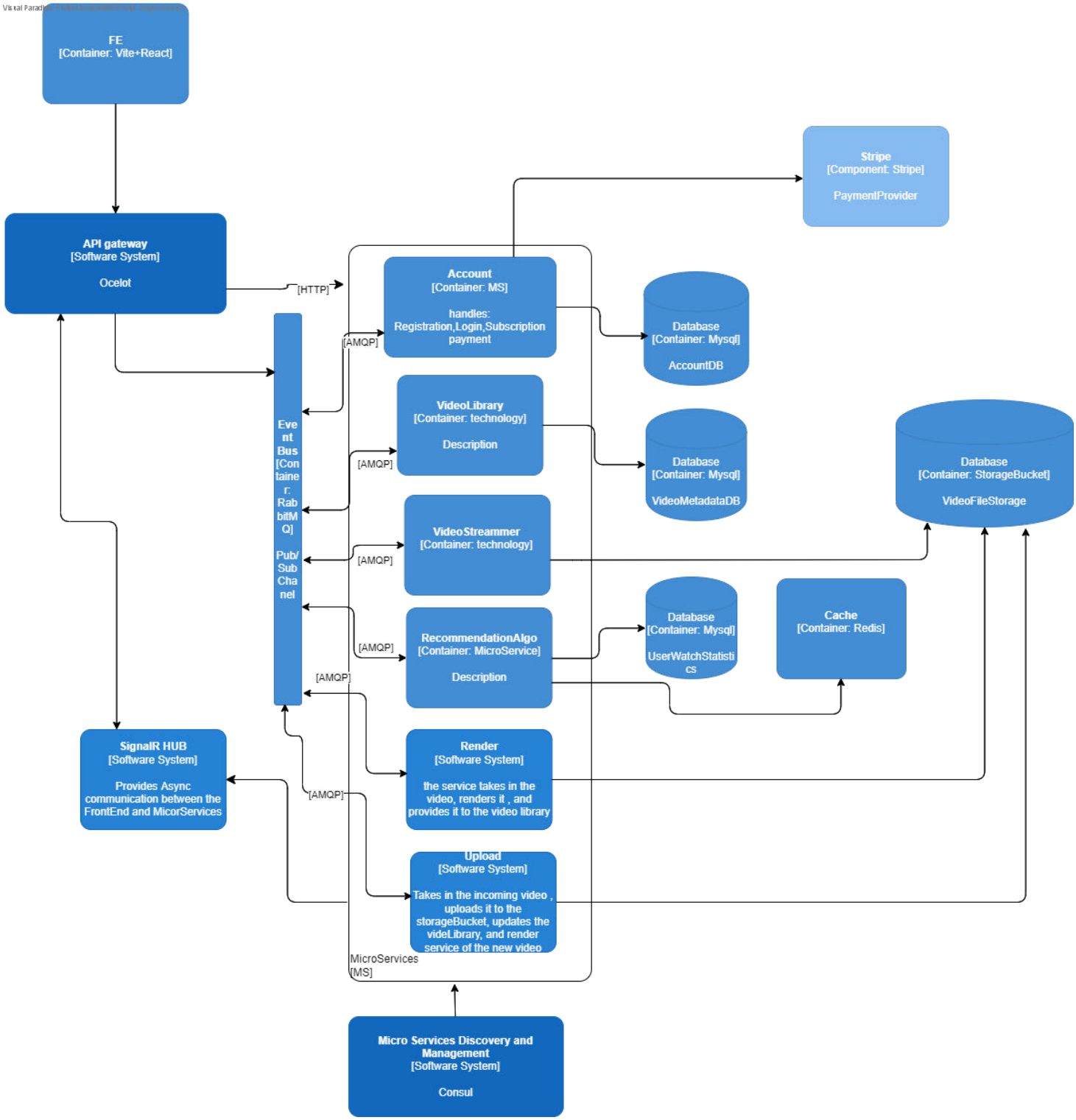
This decentralized approach means that each service manages its own database, and any relationships that would traditionally be enforced through foreign keys are now managed at the application level through messaging. This is part of a broader strategy to ensure that services are decoupled and can be developed, deployed, and scaled independently.

3.5 C4 Diagrams

3.5.1 C1-Context


Visual Paradigm Professional(ra408(Fontys Hogescholen))
Stripe
[Component: Stripe]
PaymentProvider
User
[Person]
Browser
Initiate Payment Request
[Http]
finsih payment (specify amount, acknowledge, etc...)
[]
Serving the Static content
[Http]
Dynamic Resources
[Http]
FrontEnd
[Software System]
SPA in React (provides the presentation layer to the user)
[Http]
[SignalR]
Backend
[Software System]
a micro service system in the .NET ecosystem (handles all backend operations)
OpenVidStreamer
[operationSystem]
scalingFactor

3.5.2 C2-Backend Architecture



3.4.3 React SPA Component Tree

To BE added *in later document revisions*