­VR Audio Visualizer Design Document

*Team name: SIMM*A

*Team members: Shayan* ***S****., Evan* ***I****., Joseph* ***M****., Emma* ***M****., Brant* ***A****.*

1. Introduction

*[Provide an overview of the design document, including its purpose, scope, and intended audience.]*

This document provides a comprehensive\* overview of the architectural design for the Reverb XR application. It aims to outline the system’s architecture, component design, and interaction with various technologies to ensure a seamless user experience in a virtual reality environment. The intended audience includes our developers and any future contributors who may work on extending or maintaining the system.

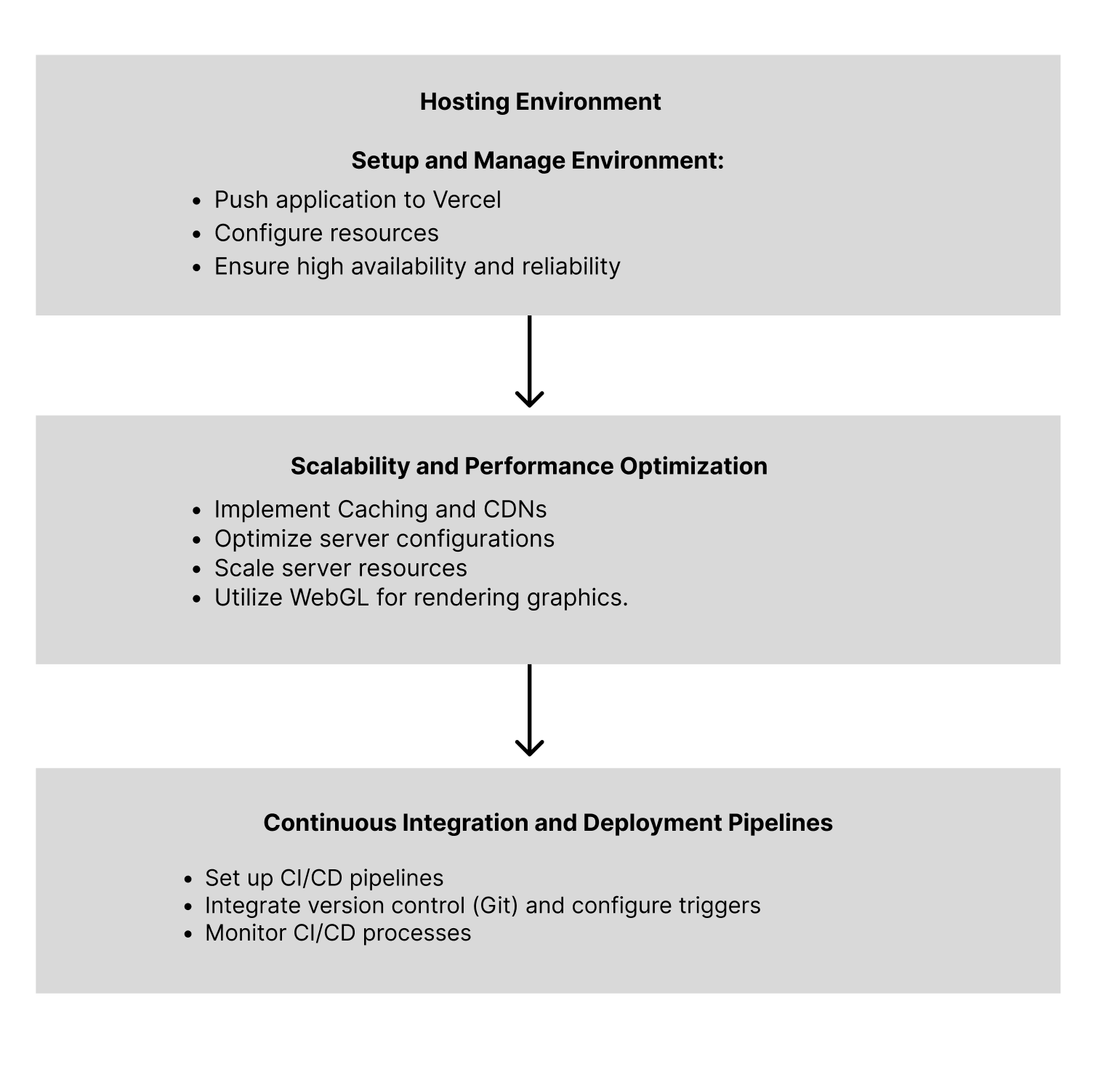
2. System Overview (Architectural Design)

*[Describe the high-level architecture of the system, including its major components, layers, and interactions. Include a system architecture diagram to illustrate the overall structure.]*

The Reverb VR application leverages a combination of Unity for immersive VR content creation, A-Frame for integrating this content into the web, and Vercel for hosting and deploying the application to end users.

Major components include:

* **WebAudio API Component**: Processes audio data for effects rendering
* **Unity WebGL Component**: Manages the creation and rendering of interactive VR scenes.
* **A-Frame Integration Component**: Embeds and enhances the Unity WebGL output within a web-based environment.
* **Vercel Deployment**: Ensures efficient deployment and scalable hosting of the web application.



A diagram of a web design

Description automatically generated

3. Stakeholders (Architectural Design)

*[List the stakeholders involved in the project and their respective concerns or requirements.]*

**Programmers**: Develop and integrate various system interfaces, ensuring functionality across different platforms.

**Users**: Interact with the 3D/VR interface, expecting a responsive and immersive experience.

4. Quality Attributes (Architectural Design)

*[Identify and describe the quality attributes addressed by the architecture, such as performance, scalability, security, and maintainability.]*

**Performance**: Ensuring smooth frame rates (>60fps) and quick load times, critical in VR environments.

**Scalability**: Ability to handle a growing amount of work and concurrent users, facilitated by Vercel’s infrastructure.

**Security**: Secure handling of user data and interactions, implementing standard security measures like HTTPS.

**Maintainability**: Designed with clear documentation and modular components to facilitate future updates and maintenance.

5. Detailed Component Designs (Detailed Design)

5.1 Unity WebGL Component

5.1.1 Overview  
*[Provide an overview of the component, including its purpose, responsibilities, and interfaces.]*

Handles the creation and export of VR content to WebGL format, ensuring compatibility with web browsers. We will use Three.js, A-frame, and WebAudio API to help build the components of this web application. A-frame will build the scene and we will use WebGL to analysis the audio input with webaudio API and render a visualizer with it.

5.1.2 Design Considerations  
*[Discuss the design considerations and constraints for the component.]*

Ensuring high performance and rendering, as well as incorporating/integrating it seamlessly within our web application

5.1.3 Class Diagram  
*[Include a class diagram to illustrate the internal structure of the component, including classes, attributes, methods, and relationships.]*   
A diagram of a program

Description automatically generated

5.1.4 Sequence Diagrams  
*[Include sequence diagrams to depict the interactions between the component and other system components or external entities.]*

A diagram of a computer program

Description automatically generated

5.2 A-Frame Integration Component

5.2.1 Overview  
*[Provide an overview of the component, including its purpose, responsibilities, and interfaces.]*

Integrates the WebGL output into the web environment, adding web-specific enhancements and managing user interactions. A-frame will be used to create the space and visuals of the web browser

* Creation of the 3D scenes using HTML-like markup
* Has built in support for VR, the biggest feature that will help us turn the web application into a VR experience.
* Supports animations through components, allowing to create interactive an interactive menu screen and active buttons.
* User interaction, such as clicking, grabbing, and dragging objects in the scene will allow for the user to be able to navigate and interact within our web browser.

5.2.2 Design Considerations  
*[Discuss the design considerations and constraints for the component.]*

Seamless integration of complex VR scenes into a standard web framework, ensuring cross-platform compatibility.

Being able to tie A-frame, WebGL, and our WebAudio API together

5.2.3 Workflow Diagram

*[Interactive VR elements taking place in A-Frame/Three.js]*

A diagram of a computer

Description automatically generated with medium confidence

5.2.4 Class Diagram  
*[Include a class diagram to illustrate the internal structure of the component, including classes, attributes, methods, and relationships.]*   
A diagram of a structure

Description automatically generated with medium confidence

5.2.5 Sequence Diagrams  
*[Include sequence diagrams to depict the interactions between the component and other system components or external entities.]*   
A diagram of a software development

Description automatically generated

5.1 WebAudio API Component

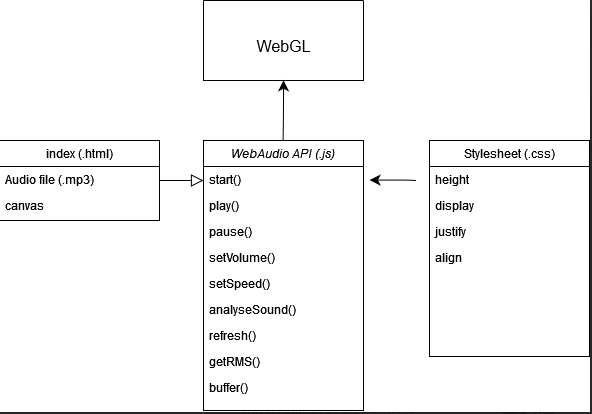
5.1.1 Overview

This component will be responsible for the initialization and playback controls of users’ audio track. It will load file data at each rendered frame and transfer rms values to Web GL. This will hold parameters such as tempo, beat, and frequency.

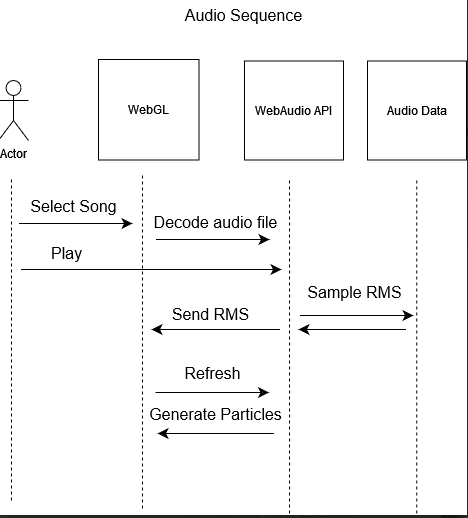
5.1.2 Design Considerations

This component will transfer rms data to WebGL to be rendered and will receive input to control audio playback parameters.

5.1.3 Class Diagram



5.1.4 Sequence Diagrams



6. Data Model (Detailed Design)

*[Describe the data model used by the system, including entities, attributes, relationships, and constraints. Include entity-relationship diagrams (ERDs) to visualize the data model.]*

[TODO ER Diagram] Unity Cloud (Key,Value) for user preferences/flat files

7. User Interface Design (Detailed Design)

A screenshot of a virtual reality game

Description automatically generated*[Describe the user interface design for the system, including wireframes, mockups, or prototypes. Discuss usability considerations and design principles applied.]*

VR and Web interface are parallel, with accessibility in focus for the web controls.

Playback controls subject to change based on implementation.

Detailed settings wrapped within pop-out windows.

A black sound wave in a room

Description automatically generatedA screenshot of a video

Description automatically generatedVR interface heavily influenced by implementation but a focus on not obstructing the vision of the user will be maintained. Floating Menus anchored to a user’s controllers.

8. Error Handling and Exception Management (Detailed Design)

*[Describe how errors and exceptions are handled within the system. Include error handling strategies, exception classes, and error messages.]*

9. Security Design (Detailed Design)

*[Discuss the security mechanisms and measures implemented in the system to protect against unauthorized access, data breaches, and other security threats. Include authentication, authorization, encryption, and auditing mechanisms.]*

* Implement a **Data-minimization policy** and focus on privacy by design. Collect only the minimum necessary data for the functionality of all components.
* Utilize **HTTPS** (HTTP Secure) to encrypt communication between the web browser, VR devices, and backend servers, ensuring data confidentiality and integrity.
* Implement cross-origin resource sharing (**CORS**) policies to restrict access to resources from unauthorized domains and prevent cross-site scripting (XSS) attacks.
* Use secure **WebSocket** connections for real-time communication between the frontend and backend components, ensuring data privacy and protection against eavesdropping.
* User authorization: prompting a message for the user to authorization to use their system audio output, if they agree to the terms, user can continue to use the application, then the application because void

[TODO] Authentication 😊? 🙁

10. Performance Optimization (Detailed Design)

*[Describe the strategies and techniques used to optimize system performance, including caching, indexing, query optimization, and resource management.]*

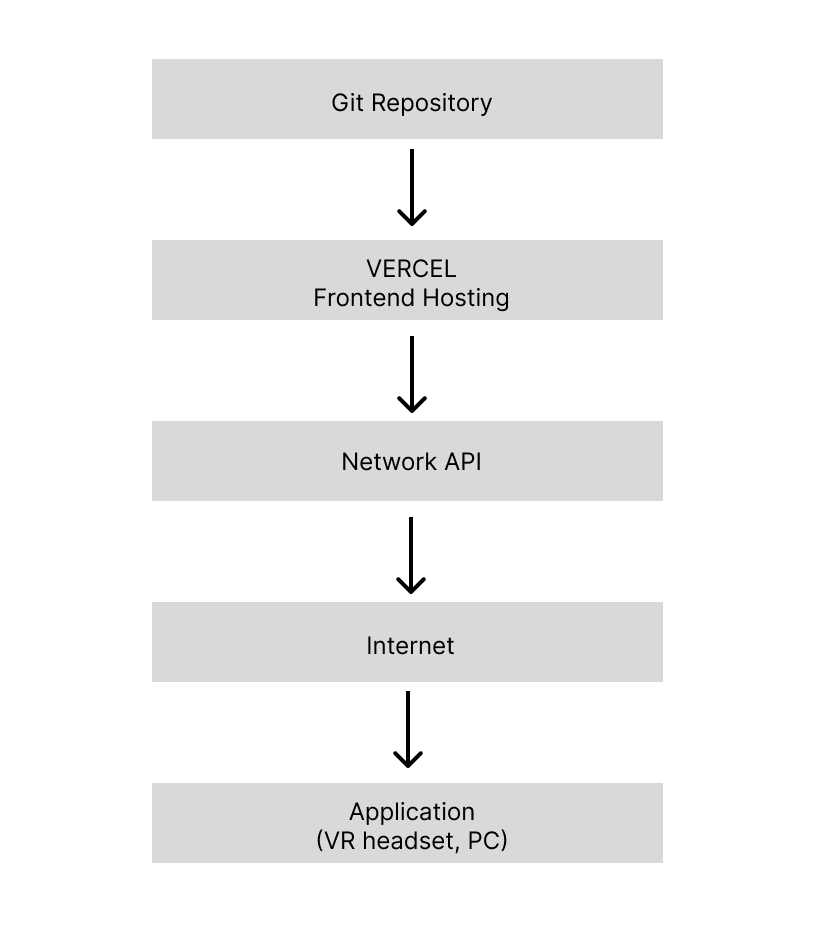
TODO

Asset Compression.

Caching to reduce latency.

11. Deployment Architecture (Detailed Design)

*[Describe the deployment architecture of the system, including hardware infrastructure, network topology, and deployment configurations. Include deployment diagrams to visualize the deployment architecture.]*



12. Integration and Interfaces (Detailed Design)

*[Describe how the system integrates with external systems or services. Discuss communication protocols, APIs, and data formats used for integration.]*

* **Unity APIs**
  + **WebGL API**  
    *Purpose*: Allows Unity games and applications to be run in a web browser using WebGL.  
    *Component Interaction*: Essential for the Unity WebGL Component to export the VR content for web use.
  + **Audio API***Purpose*: Provides access to audio functionality in Unity, allowing for audio capture, playback, and manipulation directly within Unity.*Component Interaction*: Critical for integrating audio-driven interactions and visualizations in the VR scenes.
  + **Cloud Services API***Purpose*: Data handling*Component Interaction*:
* **Web Audio API**  
  *Purpose*: Facilitates audio operations in web applications, allowing for the creation, processing, and control of audio content directly in the browser.  
  *Component* *Interaction*: Used within the A-Frame component to manage audio inputs and outputs, especially if additional web-based audio processing or enhancements are required.
* **A-Frame Component System API**  
  *Purpose*: Enables the creation of reusable components in A-Frame that can encapsulate both appearance and behavior for 3D objects in the scene.  
  *Component Interaction*: Used extensively in the A-Frame Integration Component to create custom behaviors and interactions within the VR environment.
* **Three.js Graphics API**  
  *Purpose*: Provides lower-level access to WebGL capabilities, offering detailed control over 3D scenes and graphics.  
  *Component Interaction*: Integrated within the A-Frame components for advanced visual effects and optimizations that are not directly supported by A-Frame.
* **OAuth / OpenID Connect API**  
  *Purpose*: Standard protocols for secure authorization and authentication over the Internet.  
  *Component Interaction*: Implemented within the Security Design component to manage secure user logins and access controls.

13. Testing Strategy (Detailed Design)

*[Describe the testing strategy for the system, including unit testing, integration testing, system testing, and acceptance testing. Discuss testing tools, frameworks, and methodologies employed.]*

* Unit testing
* Integration testing
* System testing
* Continuous integration
* Analysis

14. Maintenance and Support (Detailed Design)

*[Discuss the maintenance and support considerations for the system, including version control, documentation, bug tracking, and release management processes.]*

Any bug fixes and/or updates can be handled with Unity’s cloud content delivery. Version control can be resolved with the content buckets of the cloud content delivery.

15. Glossary

*[Provide a glossary of terms and acronyms used throughout the document for clarity and consistency.]*

**VR**: Virtual Reality

**API**: Application Programming Interface

**UX**: User Experience

**WebGL**: Web Graphics Library

**JSON**: JavaScript Object Notation

**CORS**: Cross-origin resource sharing

16. References

*[Include any references or resources consulted during the design process.]*

[Web Audio API](https://www.w3.org/TR/webaudio/)

[Three.js](https://threejs.org/)  
[Unity Cloud Services API](https://services.docs.unity.com/cloud-save/v1/index.html)[Vercel](https://vercel.com)[A-Frame](https://aframe.io)  
[UML Quick Reference](https://holub.com/uml/)