OPTIMIZED PLATFORM FOR ASYNCHRONOUS VIDEO COMMUNICATION (OPAVC)-SAAS PRODUCT

***Project Reference No.:** 48S_BE_6417*

College : Dayananda Sagar Academy of Technology and Management, Bengaluru
Branch : Department of Information Science and Engineering
Guide(s) : Dr. Rajesh L
Ms. Surabhi K R
Student(s) : Mr. Akash Deep Singh
Mr. C Manjunath
Mr. Dhanush L
Mr. Dheeraj M N

Keywords:

Real-Time Recording, Whisper AI, Rust Backend, Video Sharing, Cross-Platform, Media Processing, Transcription, Open Source, Automation

Introduction:

Screen recording and sharing have become essential in remote collaboration, education, and technical demonstrations. However, many traditional tools depend heavily on cloud services, which often raise privacy concerns and restrict offline usability.

To overcome these challenges, OPVAC is developed as an optimized, open-source platform that enables seamless screen and webcam recording, automatic transcription, and instant video sharing. Built using modern technologies like Tauri, Rust, FFmpeg, and OpenAI Whisper, OPVAC offers a secure and scalable solution for video-based asynchronous communication—bridging the gap between usability, speed, and data ownership.

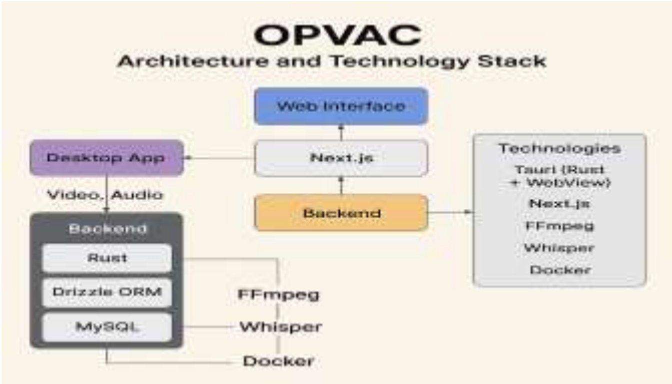

Figure 1: Architecture and Technology Stack

Objectives:

1. Develop a cross-platform desktop application using Tauri and Rust to enable lightweight, high-performance screen and webcam recording.
2. Integrate FFmpeg for real-time video encoding and compression to ensure high-quality recordings with minimal file size.
3. Utilize OpenAI Whisper for automatic speech-to-text transcription to enhance accessibility and content indexing.
4. Design a responsive frontend using Next.js, TailwindCSS, and TypeScript for intuitive video editing and sharing workflows.
5. Use Docker and TurboRepo for scalable deployment and efficient monorepo-based development.

Methodology:

The OPVAC project began with a clear focus on solving the limitations of existing video communication tools by offering a secure, performant, and user-friendly alternative. The initial phase involved analysing the need for asynchronous screen recording and video sharing, particularly in educational, technical, and remote work contexts. Based on this requirement, we selected a modern and efficient tech stack that supports cross-platform compatibility, fast performance, and scalable architecture.

During the design and implementation phase, the system was structured to support modular development and easy integration between components. We adopted a monorepo structure using TurboRepo and pnpm, allowing seamless collaboration and efficient version control across the frontend, backend, and desktop modules. Each component was designed with scalability, reusability, and user experience in mind.

The architecture and design are structured as follows:

- Desktop App: Built using Tauri (Rust + WebView) to enable fast and lightweight screen recording and video capturing.
- Backend: Developed in Rust using Drizzle ORM and MySQL for robust data handling and security.
- Frontend: Created with Next.js, TypeScript, and TailwindCSS for a modern, responsive user interface.
- Video Processing: FFmpeg is used to compress, trim, and optimize video recordings.
- Transcription: OpenAI Whisper is integrated to automatically convert audio from the recordings into text.
- Cloud Storage: AWS S3 is used to securely store and serve video files and transcriptions.
- Deployment: Docker is used to containerize the application for consistent development and deployment across environments.

  Once the system architecture was defined, development followed an iterative model. The desktop client was built to capture screen and webcam input efficiently. Backend APIs were developed to manage video metadata, handle authentication, and control video uploads. FFmpeg and Whisper were connected to allow seamless background processing for compression and transcription.

Throughout development, rigorous testing was conducted. This included unit tests for individual modules, integration tests between services, and end-to-end testing for full user flows such as recording a video, generating a transcription, editing, and sharing via a secure link. The team used version control and CI/CD pipelines to ensure a stable and clean deployment process.

Finally, the application was containerized using Docker and deployed in a staging environment. Recordings are uploaded to AWS S3, while user data and metadata are stored in MySQL. Users can access their recordings via instant, embeddable links, supporting easy integration into chats, documentation, or support portals.

Result and Conclusion:

The OPVAC project has successfully delivered a high-performance, cross-platform solution for asynchronous video communication, tailored for developers, educators, and remote teams. By integrating modern technologies like Tauri, Rust, FFmpeg, and OpenAI Whisper, the platform ensures a smooth, privacy-first, and efficient video recording experience. Users can record both screen and webcam inputs with low system resource usage and automatically generate transcriptions, enhancing accessibility and clarity.

The web interface offers a seamless user experience, enabling easy playback, trimming, and sharing through clean, embeddable links. Videos and metadata are stored securely using MySQL and AWS S3, ensuring privacy, scalability, and reliability. The deployment process is streamlined using Docker and TurboRepo for maintainability and speed.

In conclusion, OPVAC is a robust and modern communication tool that empowers users to collaborate asynchronously without sacrificing privacy, performance, or usability. It bridges the gap between functionality and user experience, establishing a solid foundation for future expansion and real-world applicability.

Project Outcomes & Industry Relevance

1. Delivered: A Functional Prototype: OPVAC, a cross-platform desktop program for screen and webcam recording with real-time transcription and video sharing, has a functional prototype that has been successfully created.
2. Including Cutting-Edge Instruments: Implemented cutting-edge and effective techniques like as Whisper AI, FFmpeg, Rust, and Tauri to produce a solution that prioritizes privacy and performance.

3. Smooth Transcription and Distribution: Improved accessibility and asynchronous communication by enabling instant video sharing with embeddable links and automatic speech-to-text transcription through Whisper.
4. Effective Modular Structure: TurboRepo and pnpm were used to design a scalable monorepo that simplifies development and deployment across several modules.
5. Safe Cloud Backend and Storage: Using AWS S3 in conjunction with Rust, Drizzle ORM, and MySQL, a safe backend was created for dependable video and transcription storage.

Industry Relevance

1. EdTech & Remote Learning: With integrated transcription and lightweight operation, OPVAC can improve asynchronous video communication in online training modules and classrooms.
2. Developer and Support Communities: Helpful for technical explanations, problem reports, and demo videos where sharing clarity and convenience are essential.
3. Enterprise & Corporate Tools: With privacy-focused implementation, they can be incorporated into asynchronous team communication tools, internal documents, or onboarding.
4. Open-Source Collaboration: Encourages innovation in video-based technologies by offering an extensible framework for community contributions.

Working Model vs. Simulation/Study:

OPAVC project was simulated in a local development environment using Visual Studio Code, Tauri, and Next.js. Instead of full-scale deployment, tools like FFmpeg, MySQL, and OpenAI Whisper were executed locally to replicate the real-world behavior of the system.

Process Followed:

Requirement Analysis – Identified limitations in existing tools and defined goals.

Technology Selection – Choose Rust, Tauri, FFmpeg, Whisper, MySQL for their performance and modularity.

System Design – Architected a modular monorepo with TurboRepo and pnpm.

Implementation – Developed features like recording, transcription, and sharing.

Testing – Validated all modules through unit, integration, and flow simulations locally.

Study Approach: The study aimed to realistically evaluate the platform's performance, modular architecture, transcription accuracy, and cross-platform functionality through hands-on, iterative simulations—carefully replicating real-world scenarios without relying on live cloud deployment.

Project Outcomes and Learnings

1. Techniques for Real-time Media Processing obtained practical experience with FFmpeg and Rust bindings for real-time video encoding, audio extraction, and media optimization.
2. Desktop Cross-Platform Development learned how to use Tauri to create powerful desktop apps by fusing the capabilities of Rust with a cutting-edge frontend stack (Next.js + TailwindCSS).
3. Integration of Automatic Speech Recognition recognized the subtleties of effectively transcribing speech under various audio settings utilizing AI-based techniques such as OpenAI Whisper.
4. CI/CD Best Practices and Monorepo embraced contemporary DevOps techniques for modular, scalable development, including as CI/CD pipelines, Docker containerization, and monorepo tactics.
5. UX and User-Centric Design prioritized performance, responsiveness, and accessibility while highlighting simple user workflows for video recording, editing, and sharing.

Future Scope:

The future development of OPVAC presents numerous exciting possibilities to further enhance its usability, intelligence, and accessibility. These include:

1. Seamless integration with major cloud collaboration platforms (e.g., Google Drive, Dropbox, AWS, Microsofts Azure or any private clouds) to enhance storage, sharing, and team workflows.

2. Incorporation of AI-powered video summarization, keyword extraction, and smart search capabilities for quicker content discovery.

3. Development of in-depth analytics dashboards to monitor user engagement, view trends, and optimize content strategies.

4. Expansion of platform compatibility to include mobile applications for iOS and Android, enabling anytime, anywhere video creation and sharing.