SAV1: SIMPLE VIDEO PLAYBACK

Simply Powerful Video Decoding

VIDEO PLAYBACK IS HARD

Video is perhaps the most important component of modern digital media. Some of the largest websites on the internet are almost entirely devoted to video, and browser environments handle many of the mechanics of playing video. However, for developers of applications running outside of a browser environment, playing video is a difficult task with a high barrier to entry.

WHAT'S INVOLVED IN VIDEO PLAYBACK?

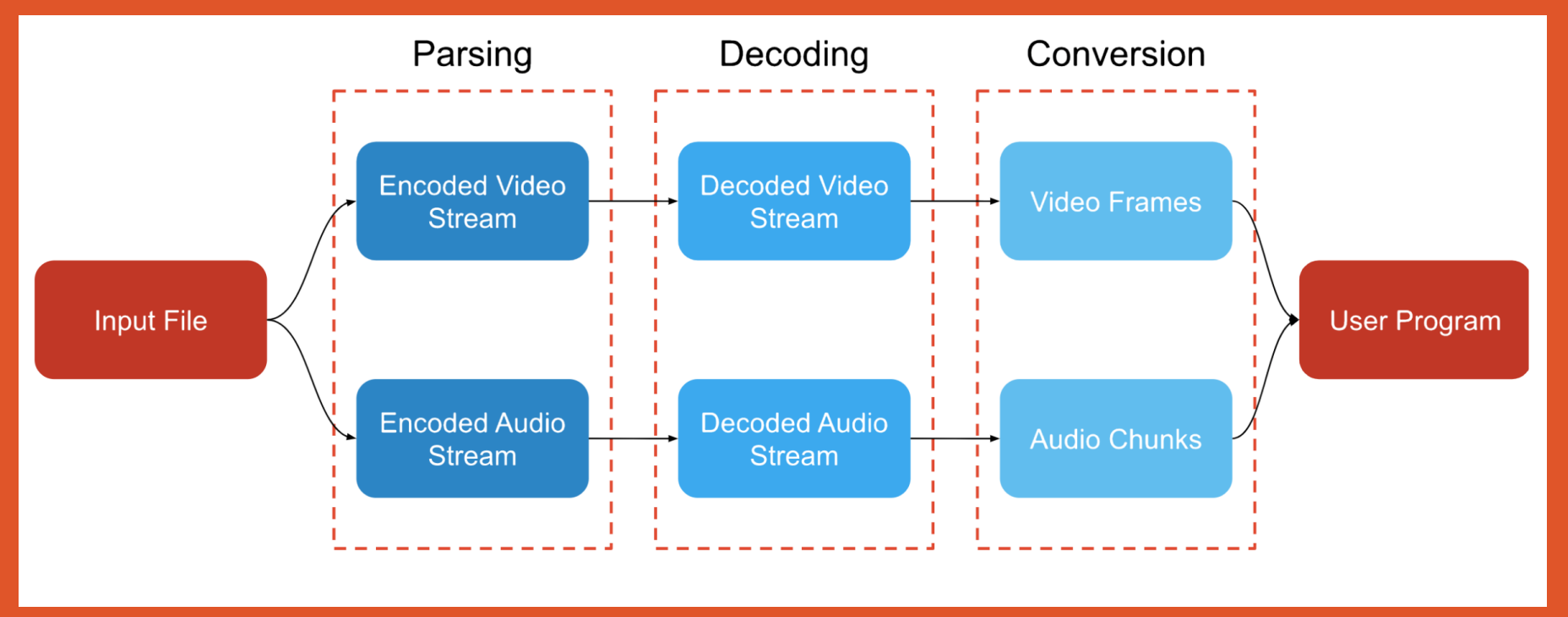


Figure 1: High level overview of preparing a video file for playback.

In the open-source ecosystem, there are many tools for video that will do one specific stage of video playback. To play a video, a developer must wrangle together a bunch of these smaller tools. There are larger projects, like FFmpeg, that aim to handle most of the video playback stages. However, these projects can suffer from other drawbacks for this use case; such as large file sizes or inconvenient APIs.

Ideally, a developer adding video to their project shouldn't need to worry about parsing or decoding their video files, getting the output into a usable format to display, or tracking frame times to know when to display them. Instead, they should just worry about how to put the frames into their existing graphical setup, and about making the best app that they can.

Oregon State University

WHAT IS SAV1?

SAV1 is a C/C++ library designed to facilitate the integration of audio and video playback into any application with ease. SAV1 provides ready-to-use video and audio data for developers in the format of their choice. Within the WEBM file format, SAV1 supports the AV1 video and Opus audio codecs which are both modern and open-source.

BENEFITS

- Simple and convenient user interface
- Frame timing and synchronization built-in
- High performance
- Small file size
- Highly customizable output

EXAMPLE USAGE

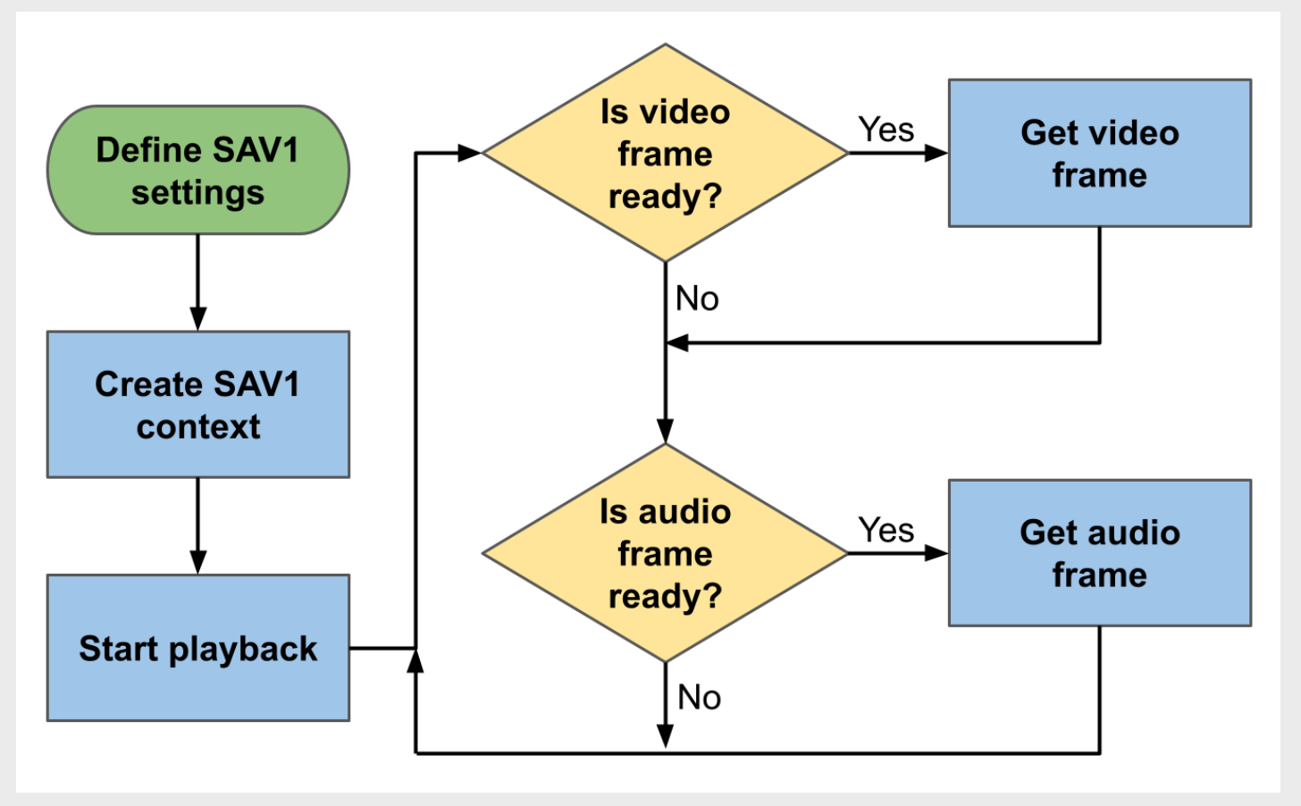


Figure 2: Possible program flow utilizing SAV1 where each block represents one function call.

DATA PIPELINE

SAV1 processes video and audio through a pipeline of modules split across different threads. These modules take in data from a queue, process it in some way, and then output it to another module. Because SAV1 is multithreaded, these modules can operate in parallel which allows SAV1 to be highly performant.

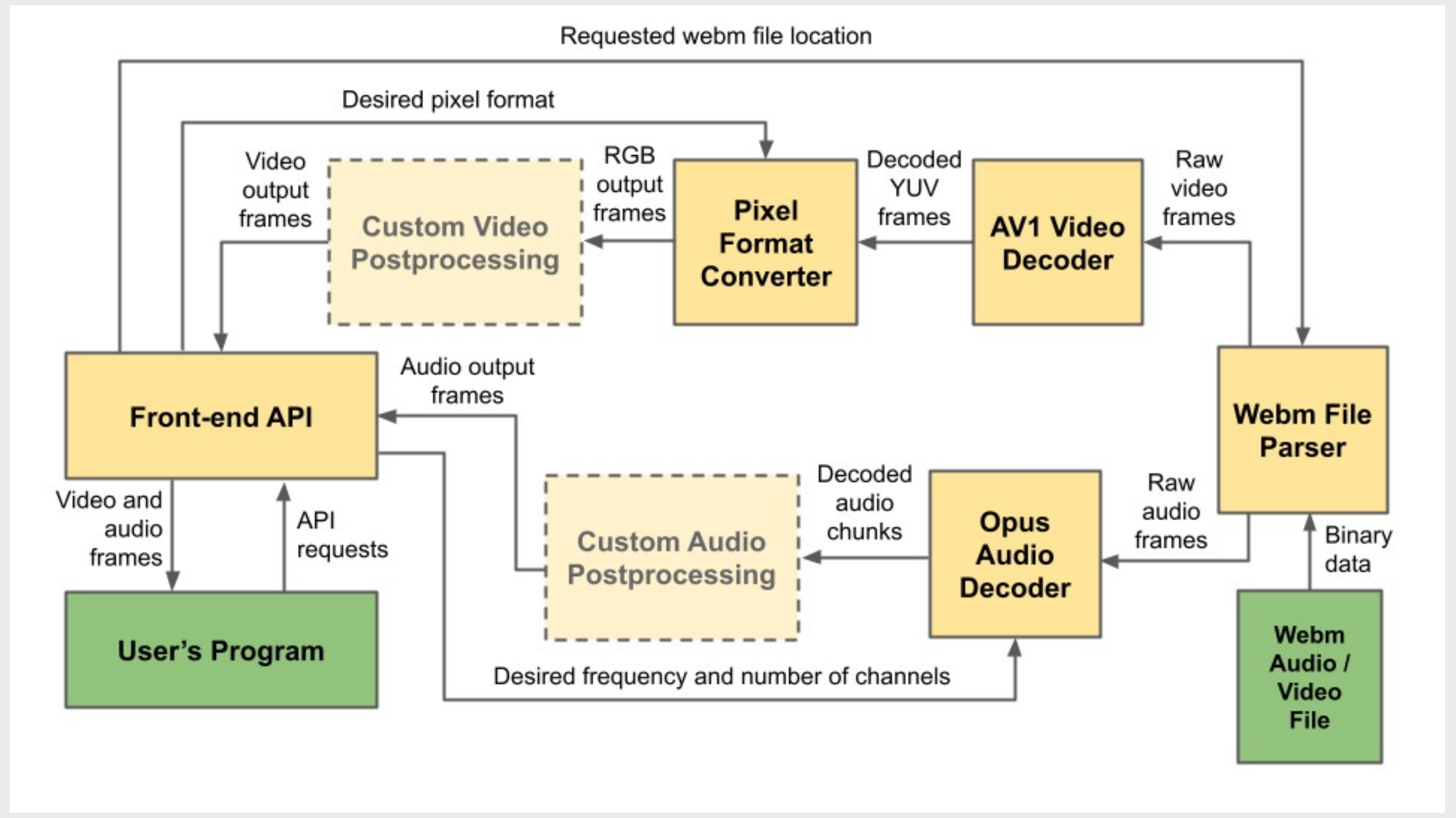


Figure 3: Architecture of the SAV1 processing pipeline.

WebM File Parsing

We use Google's open-source Libwebm library to sequentially parse the webm file into its hierarchical structure of tracks, clusters, blocks, and frames. Encoded video and audio data (and its timing information) is extracted for decoding.

AV1 Video Decoding

SAV1 relies on the Dav1d library to decode AV1 video which is the heavily-optimized, open-source library used by web browsers. Dav1d outputs decoded video frames in the YUV color space, so they must first be converted before they are given to the user.

Opus Audio Decoding

We use the open-source Libopus library which allows us to decode the audio data given by the parsing stage as well as resample it to the user's specified format. This means that no audio conversion module is required in the pipeline.

Pixel Format Converter

Users will usually want their data in some form of RGB rather than YUV, so the images must be converted before they are given to the user. This is a complicated process because there are hundreds of different forms of YUV that could be present. We handled some of these conversions using Google's Libyuv library, while others are implemented directly in SAV1.

Custom Processing

SAV1 will attempt to output video and audio in the format that the user desires, but there are limits to the number of possible formats that can be natively supported. To deal with this, users have the option of inserting their own stages at the end of the audio or video pipelines. These can do whatever conversions the user sets them up to do, and they will do them in a separate thread so that they do not slow down the user's main program.

ACKNOWLEDGEMENTS

OUR TEAM

Elijah Cirioli
eli.cirioli@gmail.com
WebM Parsing, Seeking, and Color Conversion

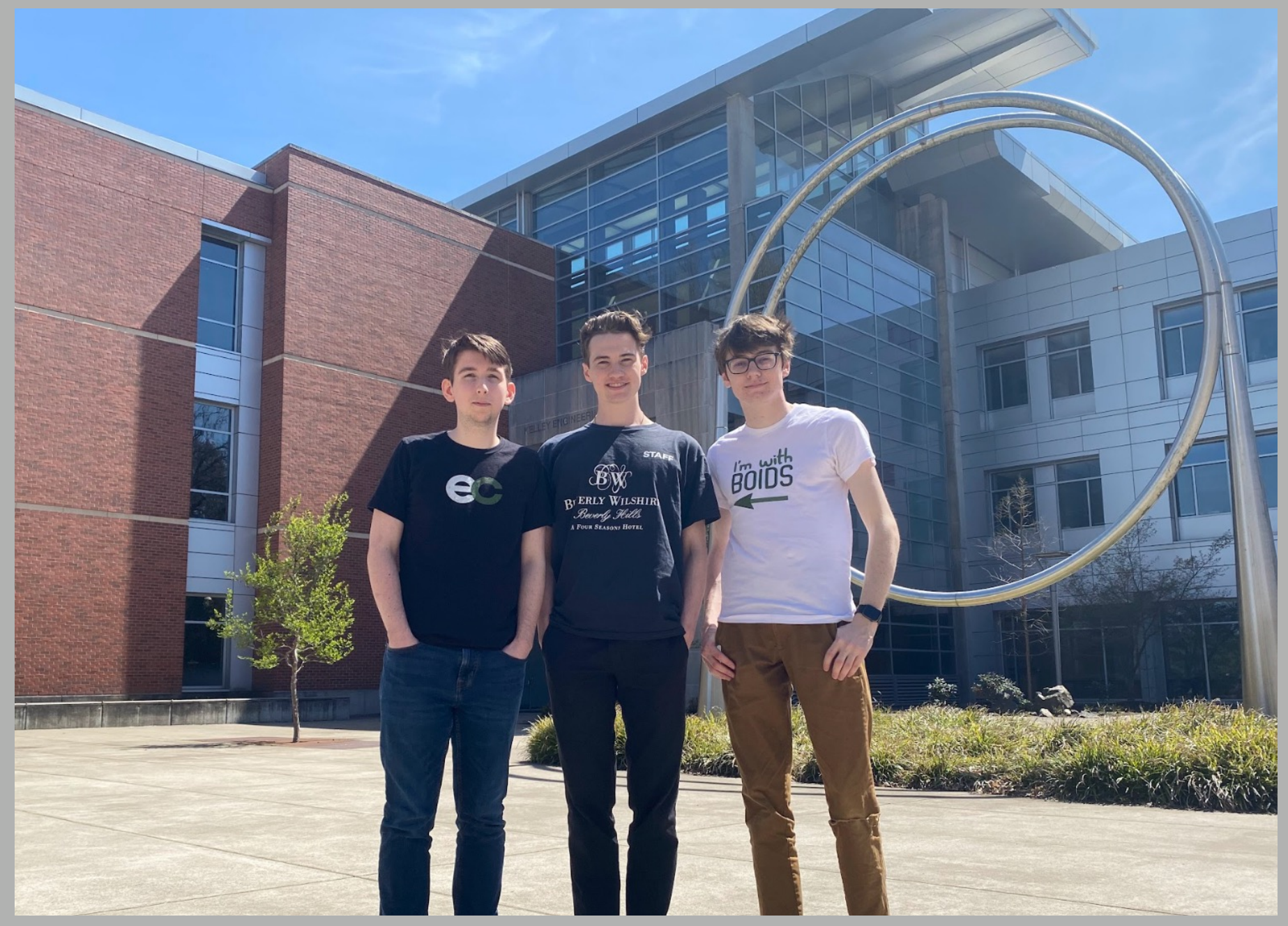
Charles Hayden
charlie.c.hayden@gmail.com
AV1 Decoding, Build Tooling, and Example Programs

Daniel Wolnick
greywolnick@gmail.com
Opus Decoding, Error handling, and API Design

DEPENDENCIES

- DAV1D
- LIBWEBM
- LIBOPUS
- LIBYUV

Find us on GitHub
https://beav.es/SjG



Left to right: Charles Hayden, Daniel Wolnick, Elijah Cirioli