aws_logo.png

The PieChecker back-end is hosted on Amazon Web Services, and makes extensive use of the Twitch.tv API. Real-time communication between the Pi, server and front-end is handled by SocketIO.

**Authentication**

In order to authenticate the user we made use of the oAuth 2.0 authentication standard. The TwitchTV Javascript SDK redirects the user to twicth.tv and after succesful authentication it redirects the user back to PieChecker.com. This redirect contains a access key in the location hash, which can be appended to requests made by the PieChecker back-end,in order to make authenticated requests on behalf of the user.

To get the access key from the front-end to the back-end, we use SocketIO. It sends a login message with the access key inside of it. As a response the back-end fetches the users channel data, including streaming key and passed broadcasts.  
  
OAuth requires the back-end to run on an SSL enabled server. However, this caused issues since the TwitchTV embedded player was loaded in insecurely (over HTTP instead of HTTPS), and therefore the browser blocked it. I had to write a proxy endpoint, where you could request the embedded object, and the PieChecker back-end would make the request for you, piping the response to it’s own response object.  
  
**File Server**

Besides an API, the back-end also serves up the front-end. We use a simple file server for this, with a limited amount of possible MIME types. All it does is look for a file in the same path written in the URL.

**SocketIO**

We use SocketIO to provide the communication between our three components: Back-End server, Front-End and the Raspberry Pi. SocketIO uses web sockets, but degrades gracefully to other solutions when they are not available (Internet Explorer 9 or lower).