Leaf

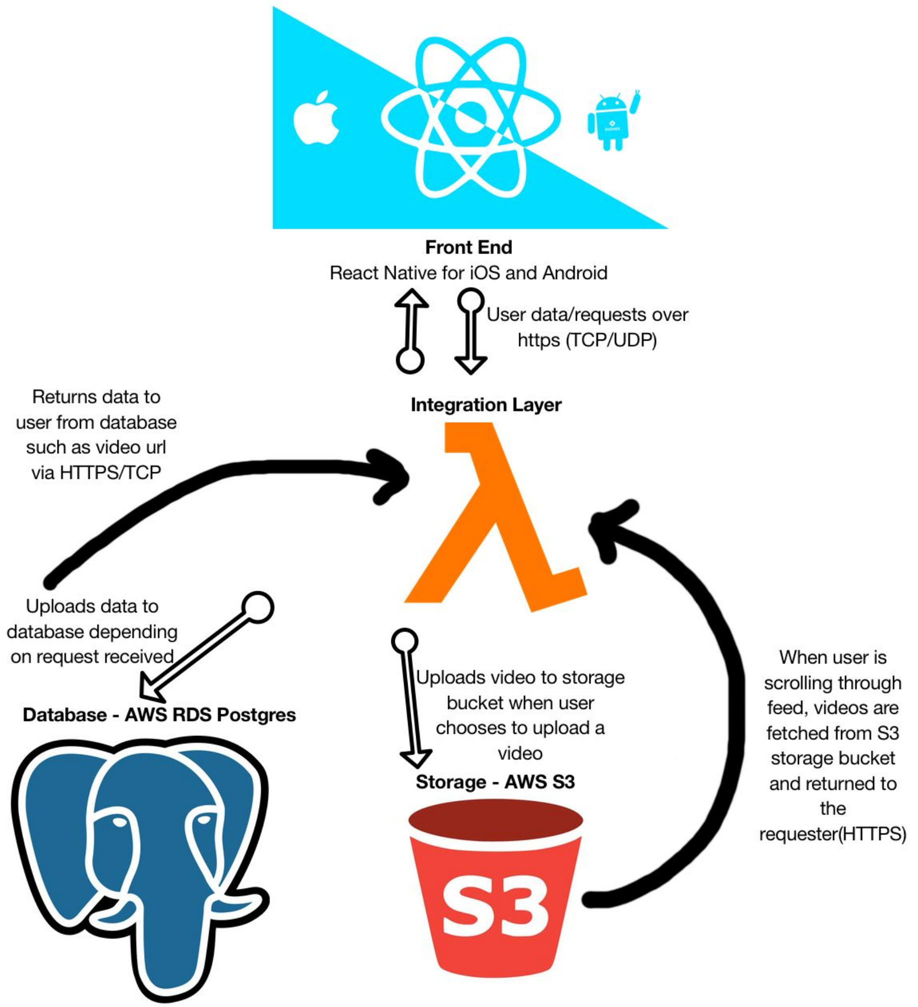
Neil, Connor, Andrew, Akhil

Features

- User Registration/Login
- Forgot Password
- Video Recording
- Video Uploading
- Video Streaming
- Following
- Profiles
- Settings (change username, change password, change name, change profile pic)
- Video “Life”

Tools Used

- **Frontend:** React Native 5/5
- **Backend:** Node.js 5/5
- **Database:** PostgreSQL 5/5
- **Infrastructure:** AWS - API Gateway, CloudFront, IAM, Lambda, RDS, S3 4.5/5
- Other
  - **VCS:** Git, GitHub 5/5
  - **Planning:** Agantty 2/5 and DBDesigner.net 4/5
  - **Testing:** Expo for frontend/integration/deployment 5/5, Postman for backend 5/5
  - **IDE's:** VSCode 5/5, WebStorm 5/5
  - Agile methodology 3/5


Front End
React Native for iOS and Android
User data/requests over https (TCP/UDP)
Integration Layer
Returns data to user from database such as video url via HTTPS/TCP
Uploads data to database depending on request received
Database - AWS RDS Postgres
Uploads video to storage bucket when user chooses to upload a video
Storage - AWS S3
S3
When user is scrolling through feed, videos are fetched from S3 storage bucket and returned to the requester(HTTPS)

Challenge - Streaming Video

Challenge: Needed to be able to upload recorded videos and then have the capability to stream them to many devices.

- Needed secure & scalable solution, Hard to work with AWS security

Solution: API request -> Lambda Function -> IAM Authentication -> Video upload to S3 Bucket -> Create record in DB -> Deliver through CloudFront

Effect: Expected & planned for, but still took longer than expected.

Challenge - .mov vs. .mp4

Challenge: iOS videos are by default in the .mov format, which was taking up quite a bit more space than the .mp4's from Android. We needed to figure out a workaround for this.

Solution: We thought we could use AWS ElementalMedia Convert to transcode the videos to .mp4, but it was too expensive. So instead, we capped video resolution at 720p to avoid large file sizes.

Challenge - Forgot Password SMS

Challenge: Had to be able to send the forgot password message to users who signed up using a phone number

- Traditional methods costed money per message sent, but we wanted a free alternative

Solution: Each carrier has an email where you can put the phone number and their ending to send an email as a text message

- Email to 1234567890@vtext.com will send a text message for verizon number
- Had to use an external API to get the carrier for each phone number

Effect: Made a function that was originally thought of as very easy take a lot more time than anticipated