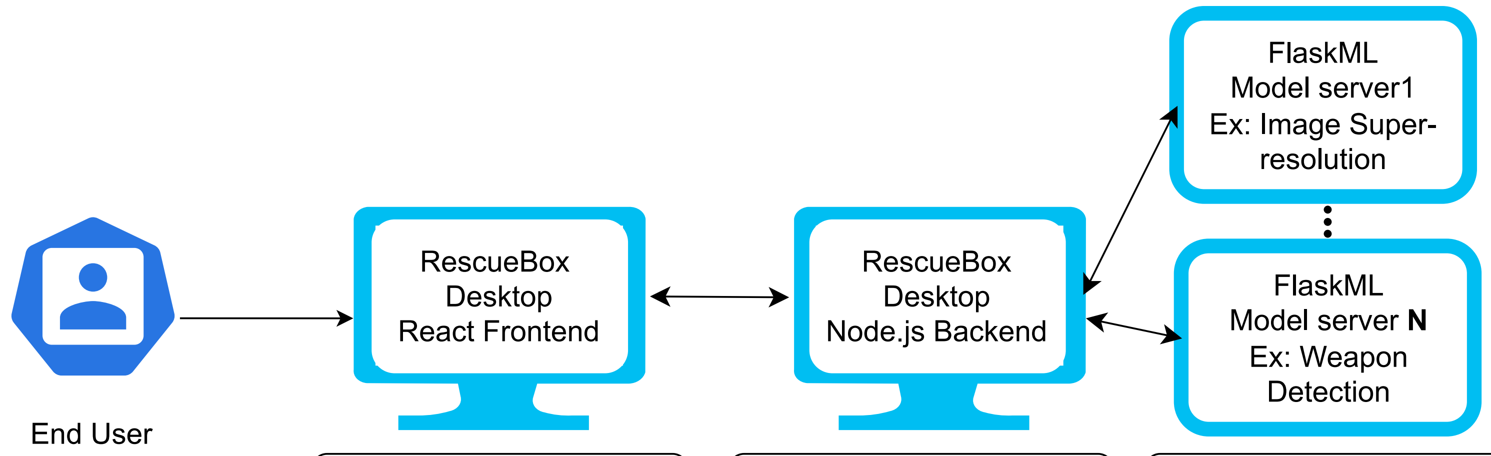
End User
RescueBox Desktop React Frontend
RescueBox Desktop Node.js Backend
FlaskML Model server1 Ex: Image Super-resolution
FlaskML Model server **N** Ex: Weapon Detection

User Experience:

1) Run the flask-ml servers for the desired models

2) Run our frontend application to easily interact with the model, which includes: inserting inputs, sending requests, viewing outputs

Electron-React Frontend

1) Model page: Viewing available models & registering new model through its ip address

2) Specific Model app pages: Viewing model details & running the model (collecting inputs - form)

3) Jobs page: Viewing Active/Past Jobs, cancel button for running jobs. Job details page & outputs tab for each job

Electron-Node.js Backend

1) Data Persistence: Tables to store jobs, models and model registration and creating an API using some ORM to access data easily.

2) Handle model server registration: storing IP addresses and polling server to check if it's running

3) Jobs: exposing a standard API to start jobs which would then interact with the flask ml servers as needed

4) Outputs: Standard API to access job metadata

5) CI: github actions to build executables for all 3 OSes

6) Logging: standard logging

Flask-ml-model apps

1) Provide standard API exposing the model's functionality.