

1

7

Economic Home Security System

Advisor & Client

Goce Trajcevski

Team Members

Andrew Tran
Kamini Saldanha
Lucas Jedlicka
Merin Mundt
Sohum Sawant
Uma Abu

Problem

- Crimes are common
 - Video evidence and alerts can help prevent them
- Security systems have many costs
- Observation: many smartphones are sitting unused

Solution

- Build a security system that allows the use of smartphones

Intended User

- Homeowner seeking a security system with access to smartphones and a computer

Intended Uses

- Single or multiple phone setup
- Get alerted when criteria is met
- View live feed from any camera
- Save clips

Title

Team

Introduction

Intended Users and Uses

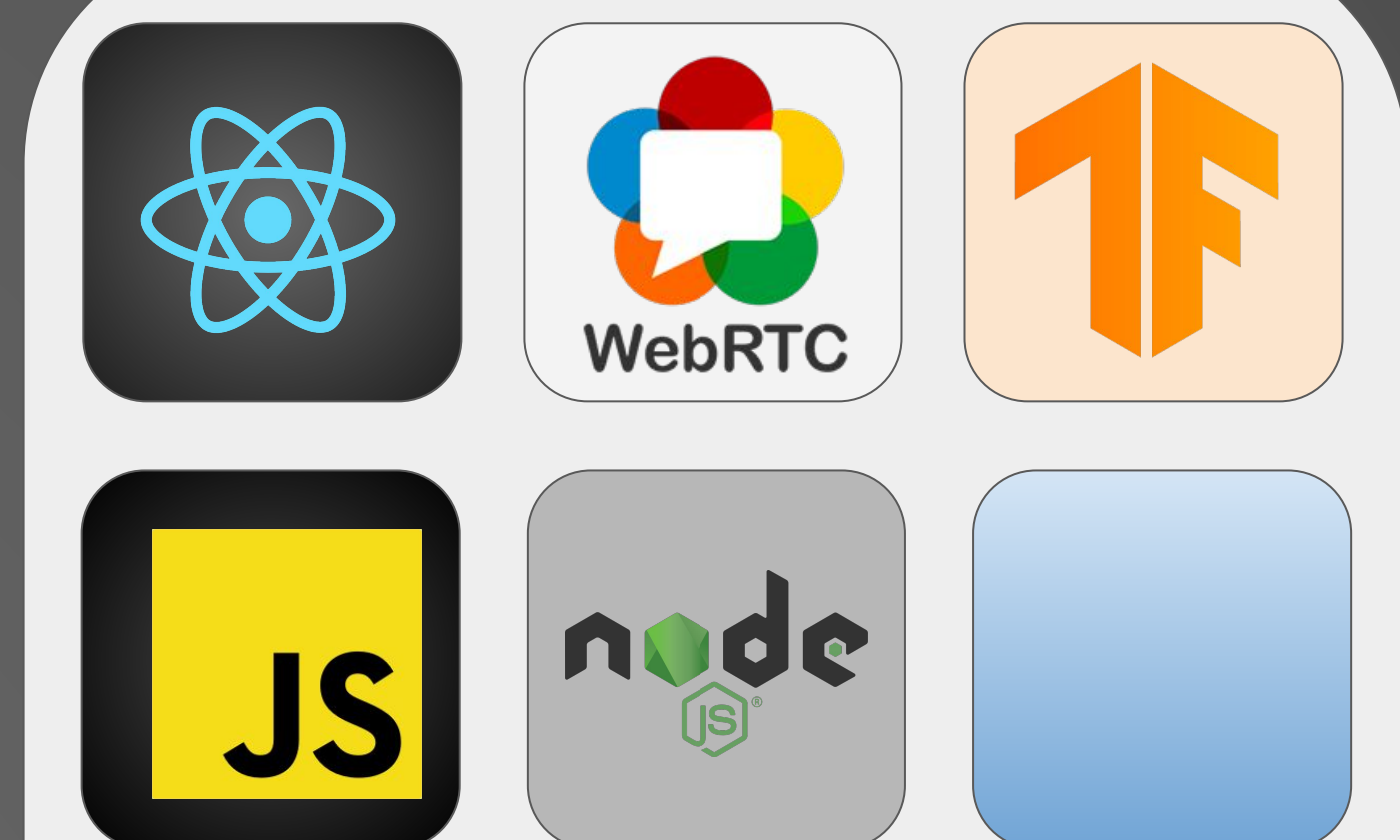
Functional Requirements

- Phones locally detect motion
 - Begin streaming upon detecting motion
 - Continuous streaming until kill signal is sent
 - Object detection
 - Send notifications or alerts
 - Save, view, and delete clips
- #### Non-functional Requirements
- Respond to an event within 5 seconds
 - Accessible from any modern web browser
 - Support at least 3 streams concurrently

- Safe from the elements
- Reliable internet and power
- Android 4.4+
- iOS 11+



Docker, Django Rest Framework, Djoser, SimpleJWT, Linux Alpine Containers, NGINX, SSL/HTTPS, Python, Websockets/WSS



Medooze Media Server/WebRTC, NodeJS, React, TensorFlow

Requirements

Operating Environment

Backend Folder

Frontend Folder

Manual Testing

- Postman
- Local server
- Correct render/updates to React components

Integration Testing

- End-to-end testing for each function (App, REST API, Server)
- Automated tests upon code changes
- CI/CD



Web Application

- Access camera using getUserMedia()
- TensorFlow for local motion and object detection
- Media Server/Webrtc to send video stream
- Components
 - Authentication
 - Clip Viewer
 - Notifications
 - Profile
 - Streamer
 - Settings



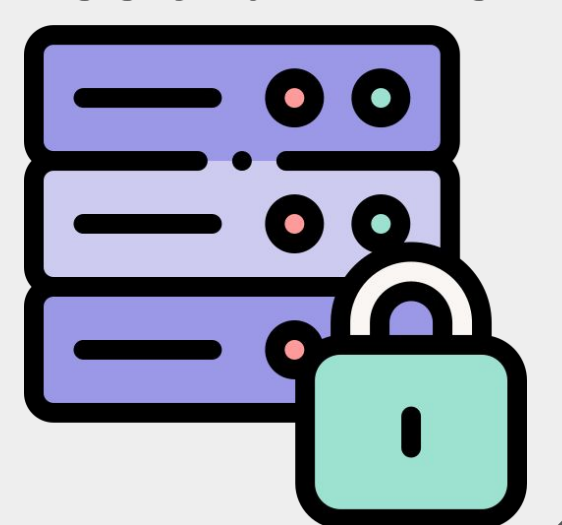
Django Rest Framework

- Built-in user model and Djoser library with web tokens for authentication
- Gmail's SMTP server for notifications
- MySQL database for easy setup and scalability
- Each component separated into apps
 - User
 - Camera
 - Clip
 - Notification



Server

- Docker used for deploying and running
 - Docker networking to limit access
 - Port redirection from URL
- NGINX reverse proxy encryption
 - Multiple applications behind a single domain
- Trusted authority SSL certificate to allow encryption via WSS and HTTPS



Testing

App

API

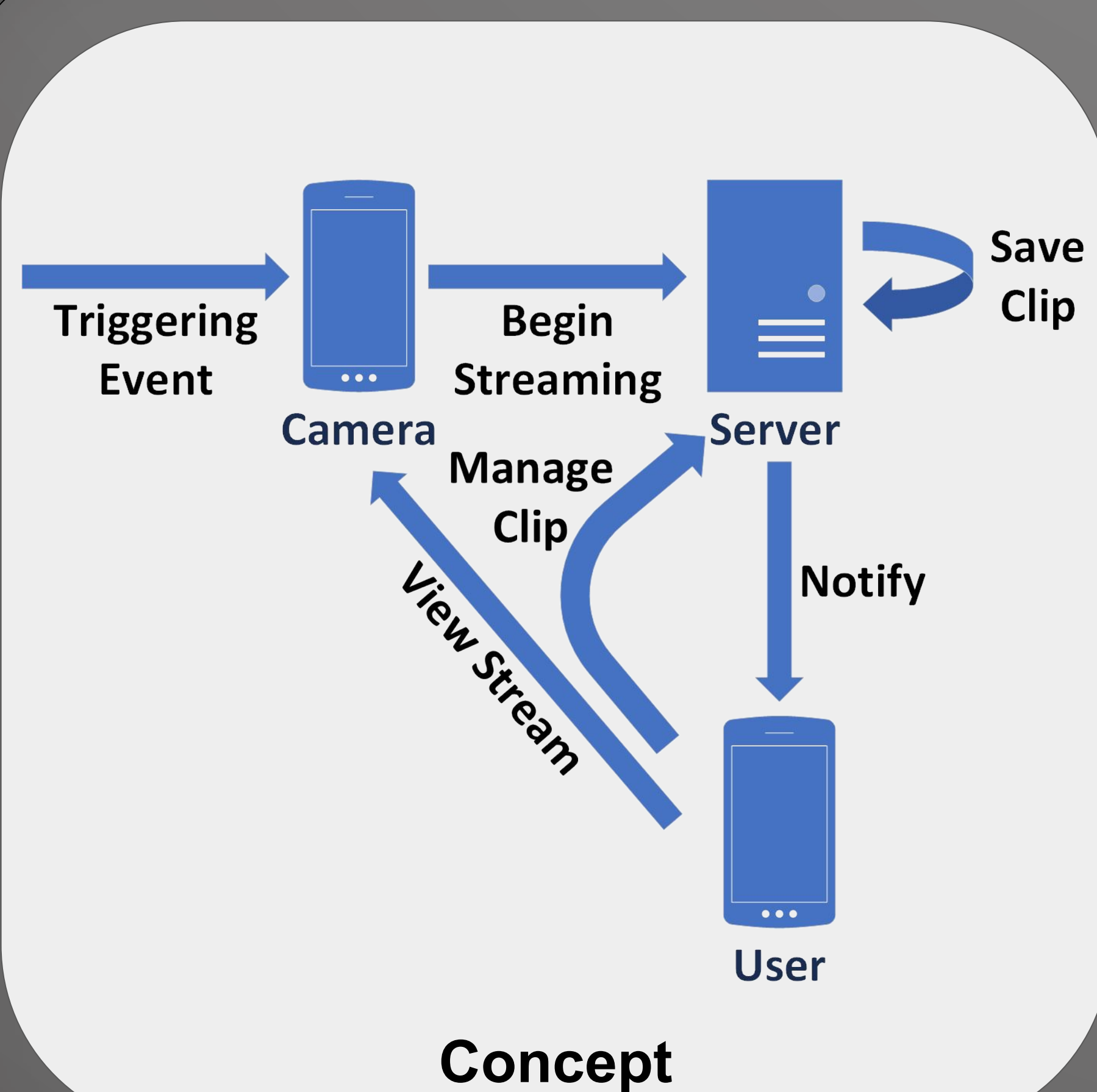
Server

ISO/IEC/IEEE 12207: Software Lifecycle Processes
ISO/IEC/IEEE 29148: Requirements Engineering
ISO/IEC/IEEE 16085: Risk Management
ISO/IEC/IEEE 14764: Maintenance

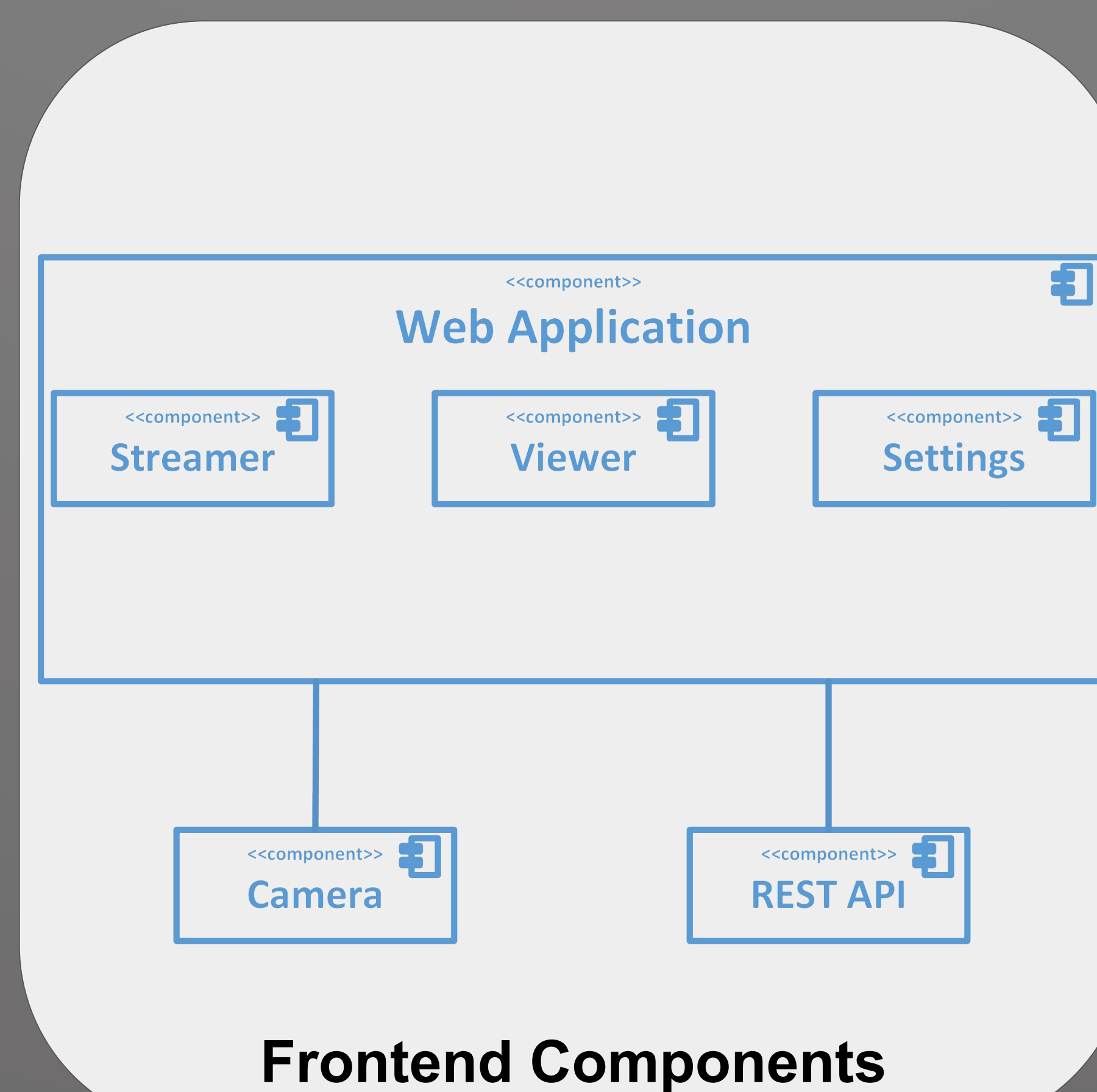
Behavior-Driven Development
Design for Simple Case First
Add Options or Additional Code for Complex
Common Git Practices

HTTPS, WSS, SSL, WebRTC, JSON Web Tokens
App Containerization
Documentation
CI/CD

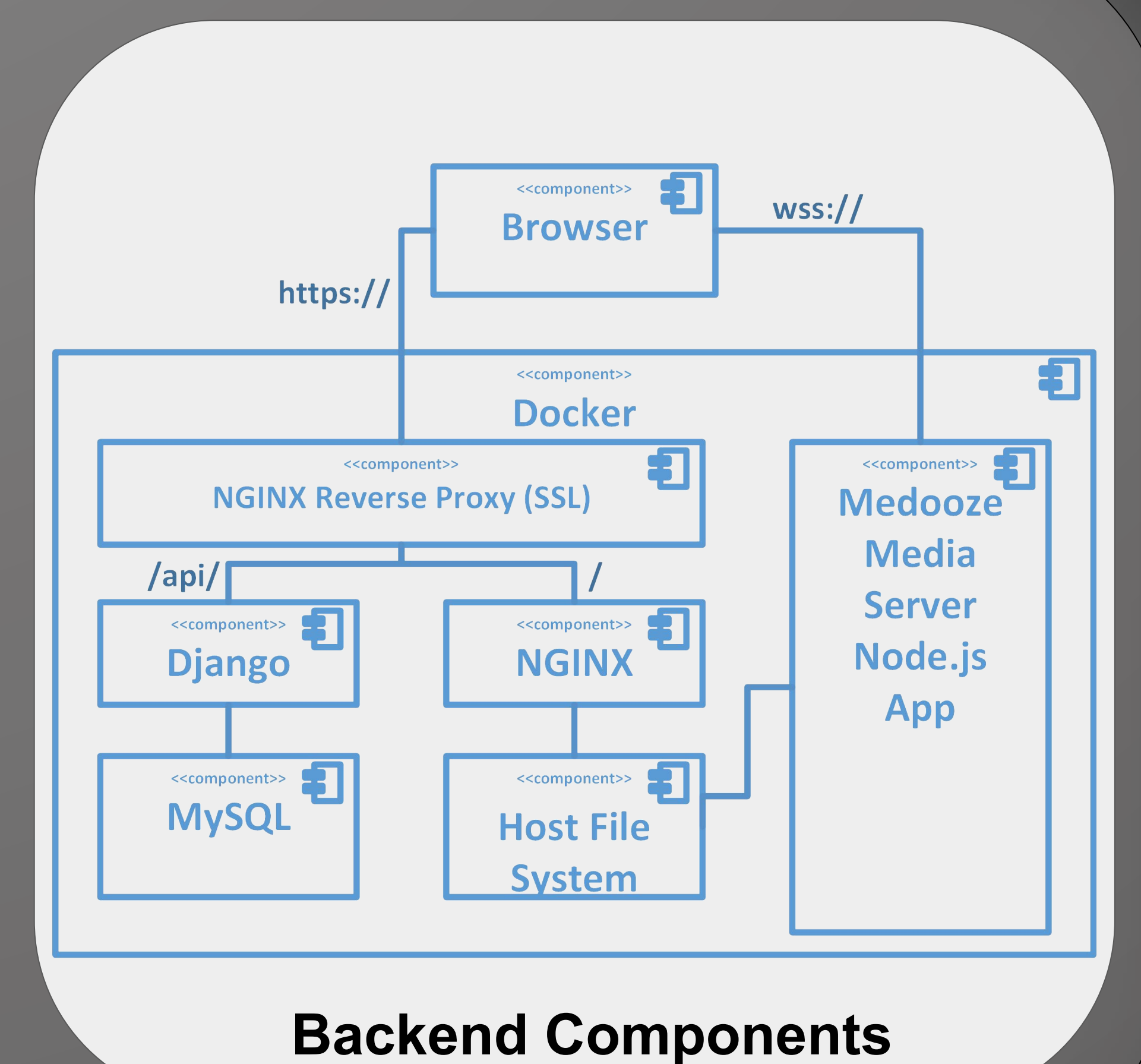
Engineering Standards and Design Practices



Concept



Frontend Components



Backend Components