MICHIGAN STATE UNIVERSITY

Project Plan Presentation
Command and Control Simulator

The Capstone Experience

Team Vectra

Trevor Davis
Ben Hayes
Nixon Holley
Ben Tuckey
Andrew Vandercar

Department of Computer Science and Engineering
Michigan State University

Fall 2023

From Students…
…to Professionals

Project Sponsor Overview

- Threat detection and response
- Real time detection across variety of networks
- Saas that tracks activity stacks of endpoints and users
- AI models to enrich detection and prioritize attacks
- Detect and identify threats based on network behavior instead of byte signatures

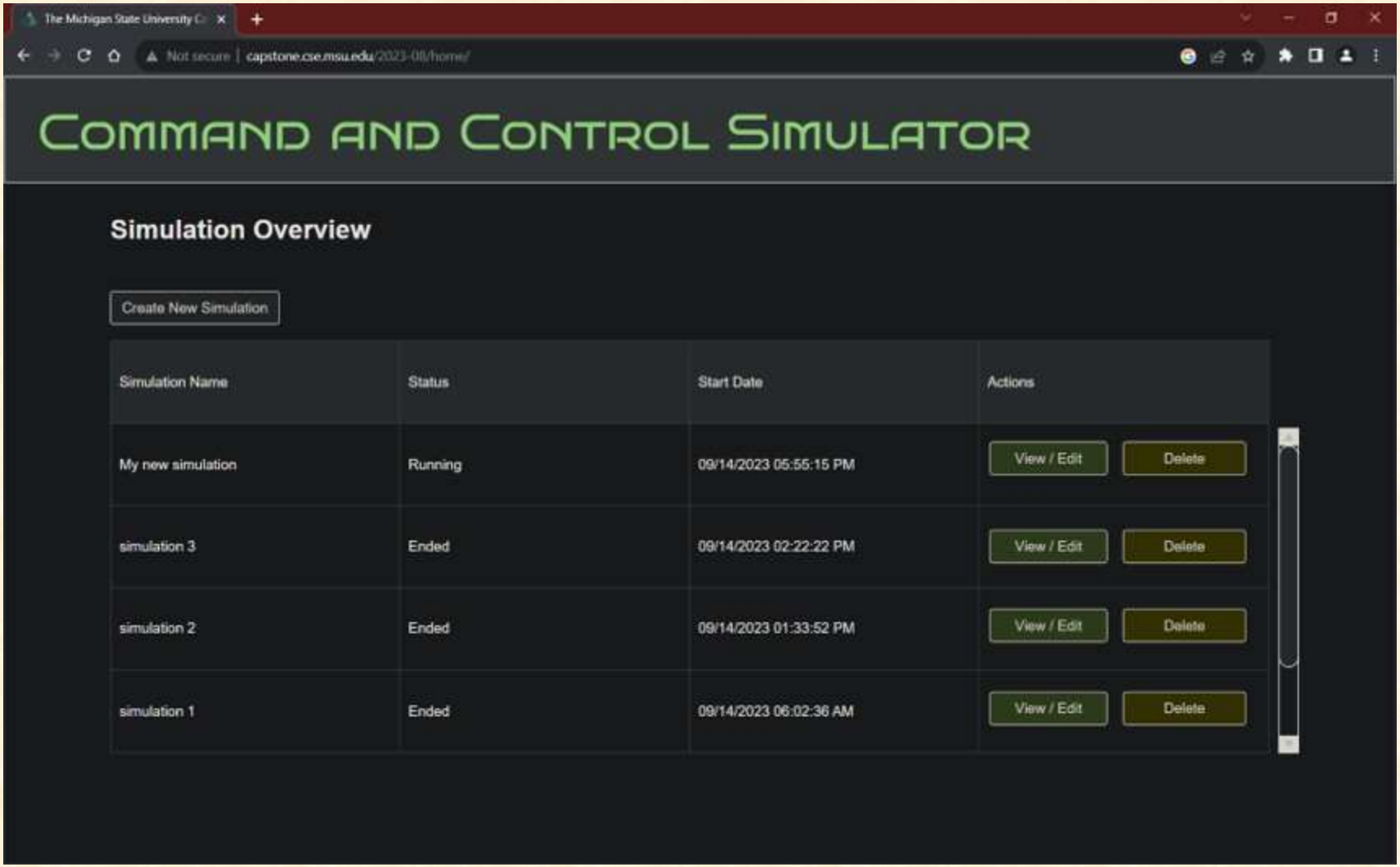
Project Functional Specifications

- C2 channels used by attackers to control devices
- Application to simulate C2 channels
- Configurable parameters and protocols
- Web UI to analyze channel traffic and control channel generation and client activity
- Used as training data for AI model

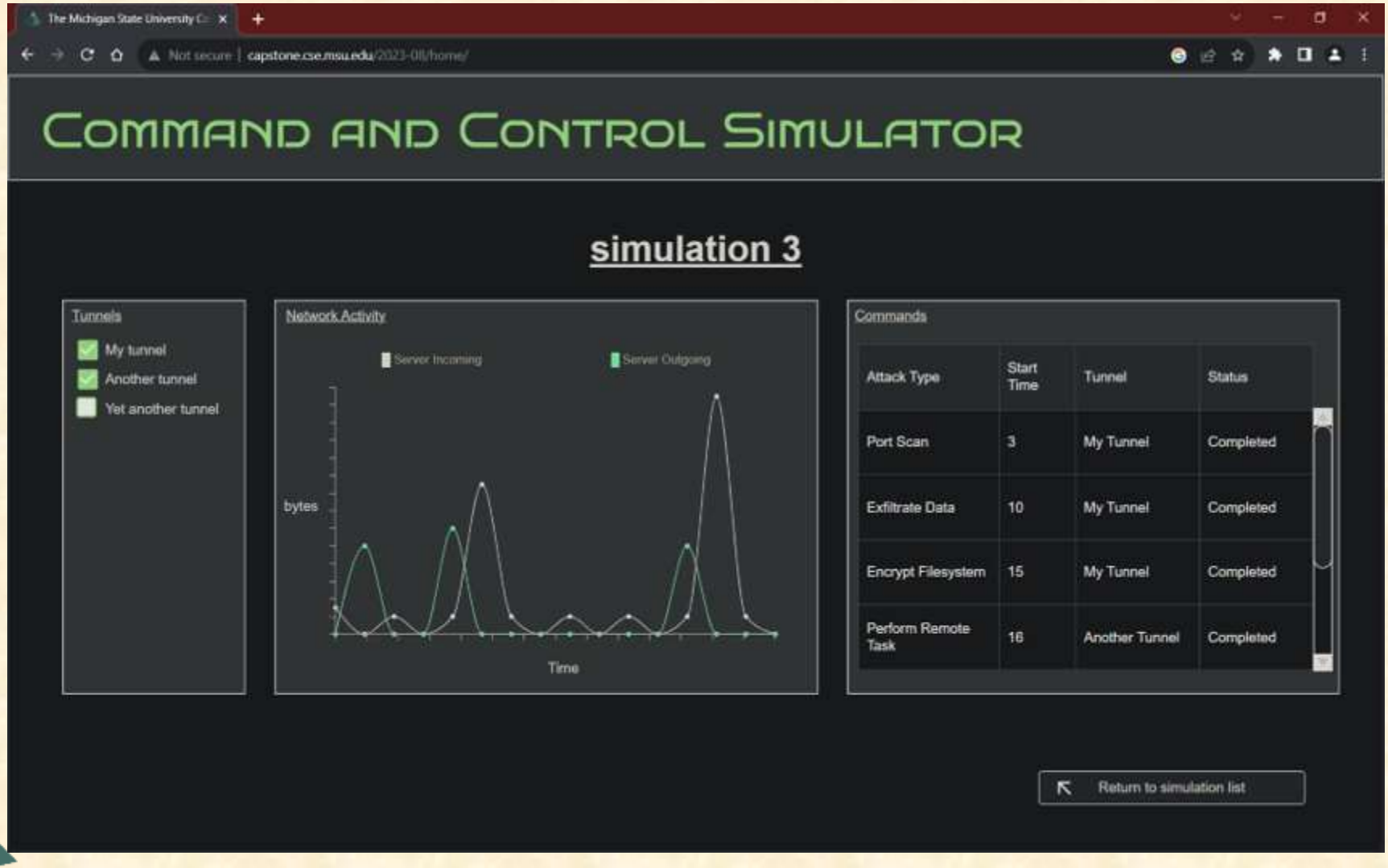
Project Design Specifications

- Server App
  - Process input from API.
  - Spawn tunnel(s) w/ client over multiple protocols and connection profiles.
  - Deliver payloads to client and return client response and meta-data to user.
- Client App
  - Establish initial connectivity with server
  - Customizable behavior received from server (jitter, beacon interval, padding, etc..
  - Handle payloads from server and return simulated data and status.
- Web-Based UI and Rest API
  - Allow user to customize tunnel config and parameters
  - Package and deliver jobs from user input to server
  - Provide user with graphed time series displaying beacon activity.

Screen Mockup: Simulation Overview

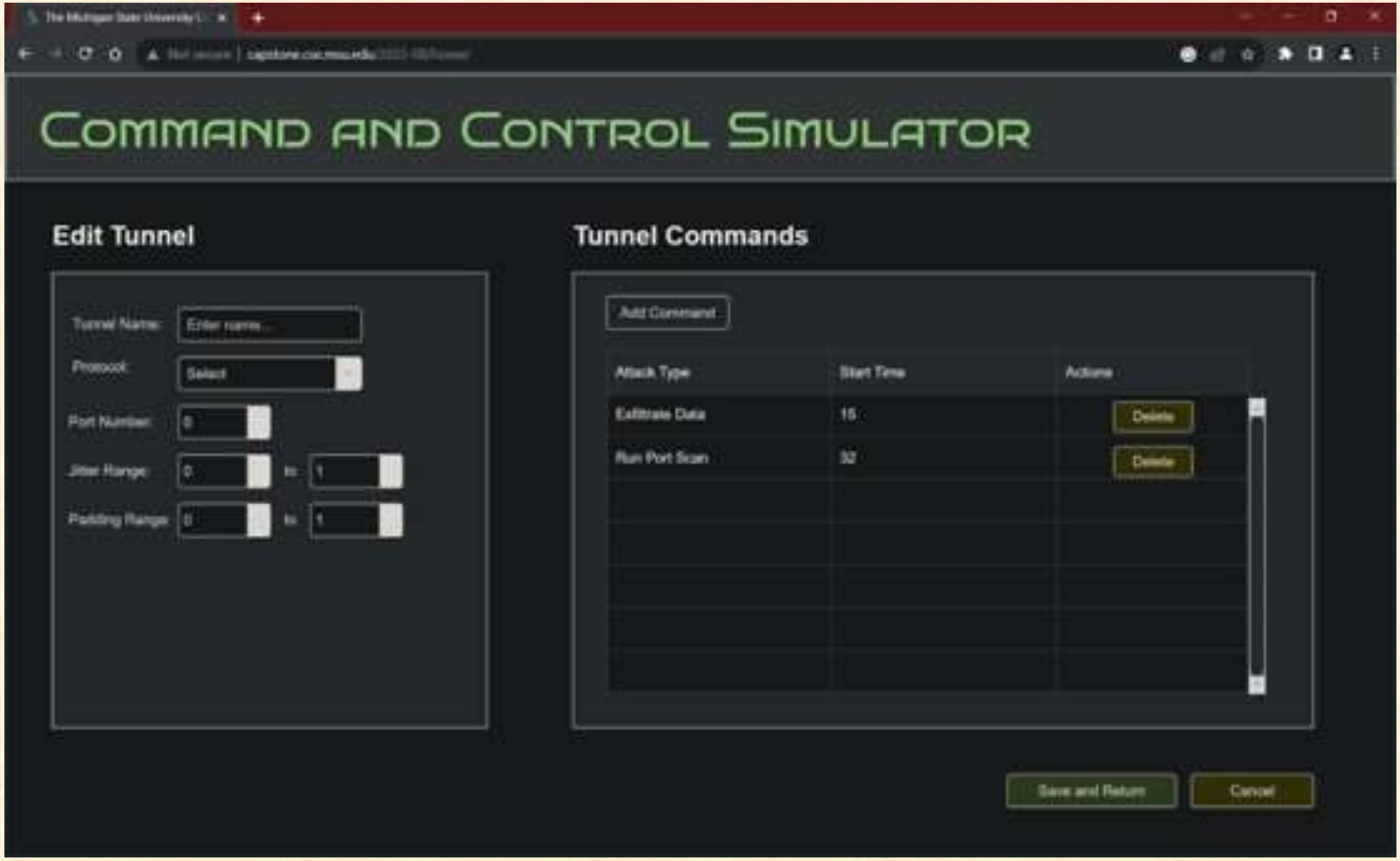


Screen Mockup: View Simulation

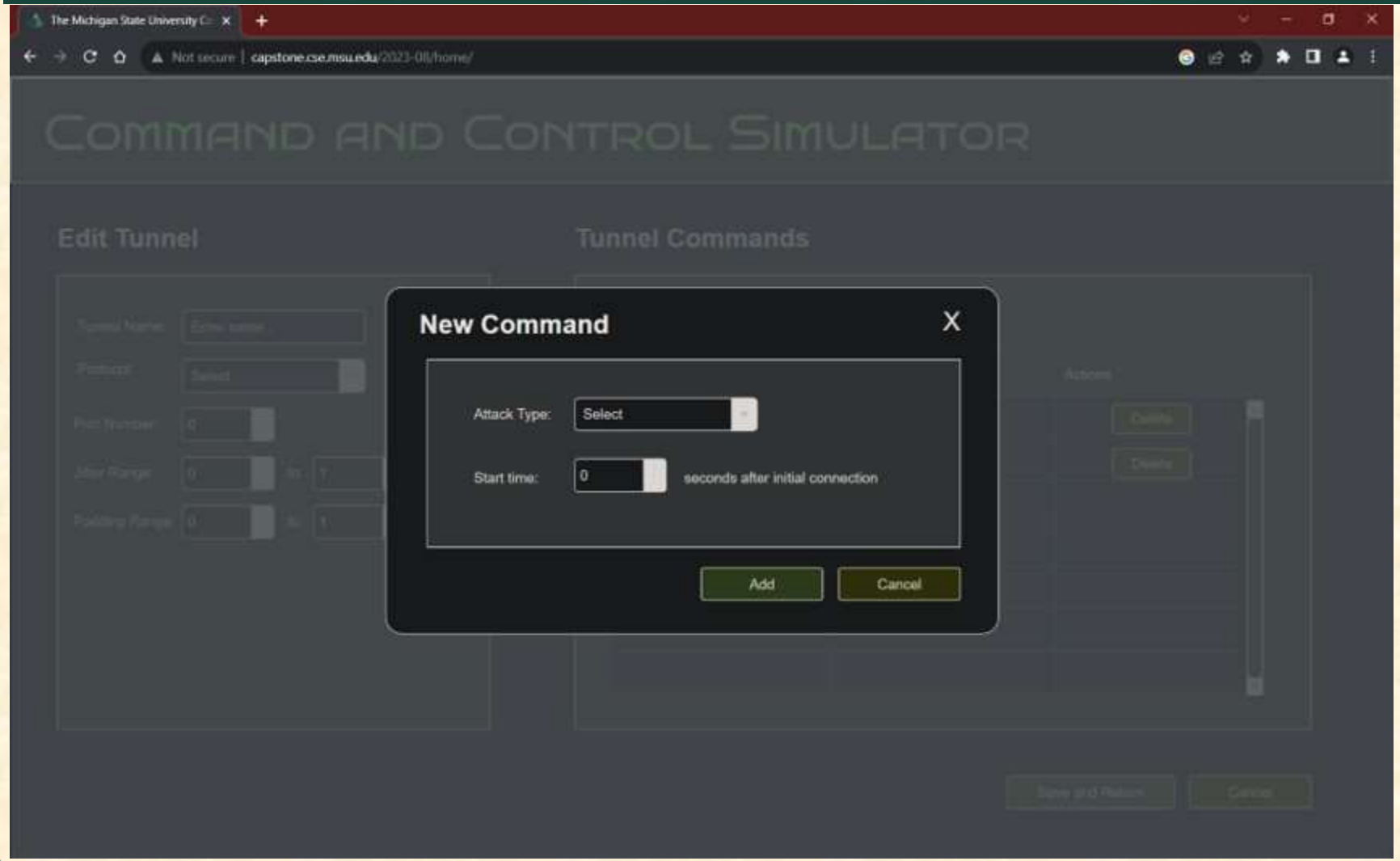


Screen Mockup: New Simulation

Screen Mockup: Edit Tunnel



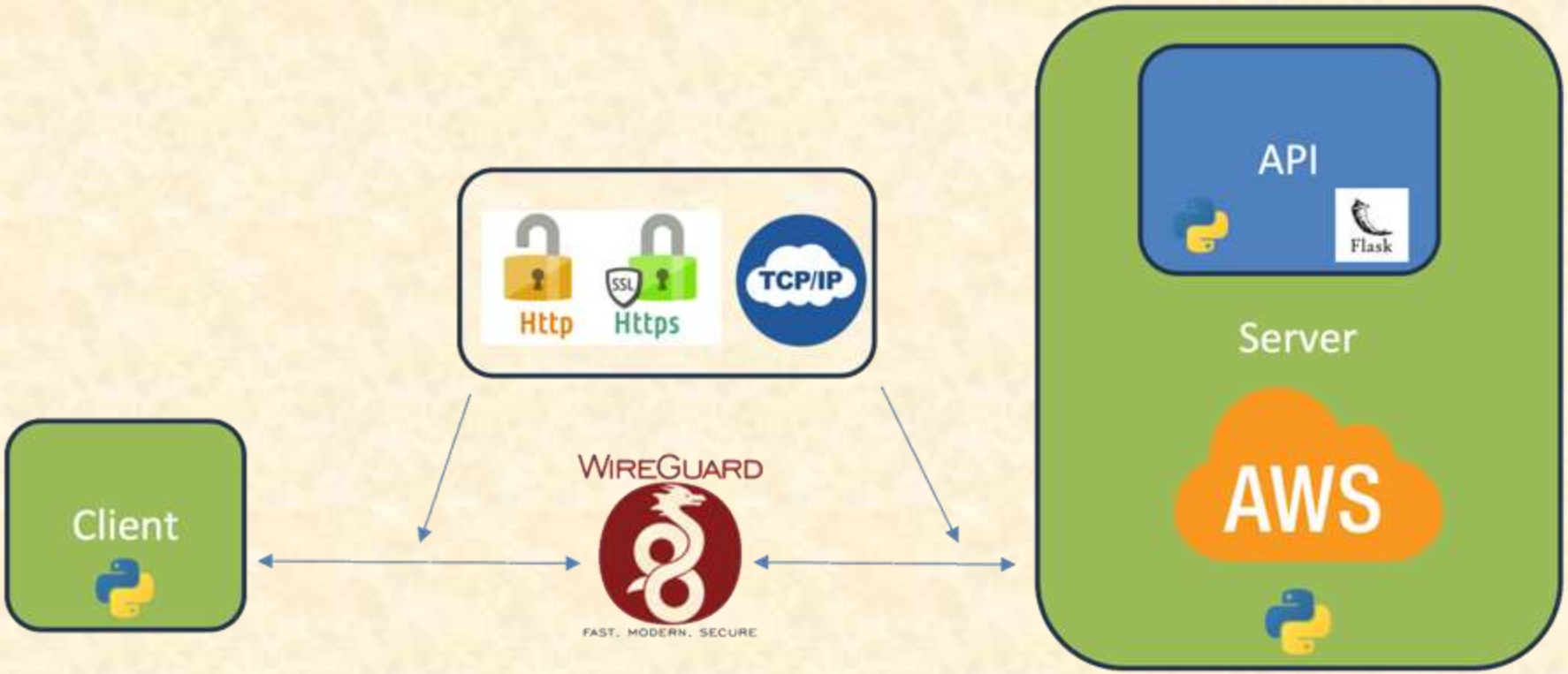
Screen Mockup: Add Command Pop-Up



Project Technical Specifications

- Backend consists of server-client program and API for controlling server
- Backend will be built in Python, using networking libraries
- Client and server will communicate with Wireguard
- Server and API will be hosted on AWS
- API will have a WebUI to view results of connection

Project System Architecture



Project System Components

- Hardware Platforms
  - Rak server used during development
- Software Platforms / Technologies
  - AWS will be used to host server platform
  - Wireguard used as a means of communication between client and server
  - Backend of API, client, and server built in Python

Project Risks

- Handling lost UDP traffic
  - UDP does not guarantee the delivery of packets or provide mechanisms for retransmission in case of packet loss
  - Design our communication protocol to include packet acknowledgment mechanisms.
- Timing Delays after migrating to cloud
  - Timing delays can manifest affecting user experience and potentially causing synchronization problems among our system's components.
  - Choose the AWS availability zones that best align with our project's requirements and we will design our system to be asynchronous and capable of handling timing variations gracefully using asyncio co-routines.
- Process packet capture data received by server to display in Web UI
  - Efficient processing and storage of pcap data, which can impact system performance and the responsiveness of our front-end web UI.
  - Employ a SQL database management system to store pcap data efficiently.
- Memory Management with Asyncio
  - If we don't properly manage/close out resources (sockets and async routines), we will have memory leak which will be a big issue on AWS.
  - Use tracemalloc to test memory usage of our application during run time

Questions?