Experiments with the Streamlit Components API

asehmi@oxfordeconomics.com

10 February, 2021

Systems architecture

Capabilities

Streamlit App

Main Python application
Port 4009

- Streamlit User Interface
- Component Host Wrapper
- Session State
- Component Received Events Handler
- Component Rerun Management (optional)

Component Front End App

Next.js component implementation
Port 3001

- Component User Interface
- Events Notifier
- Authentication (Auth0)
- Local Storage Management
- Serverless APIs

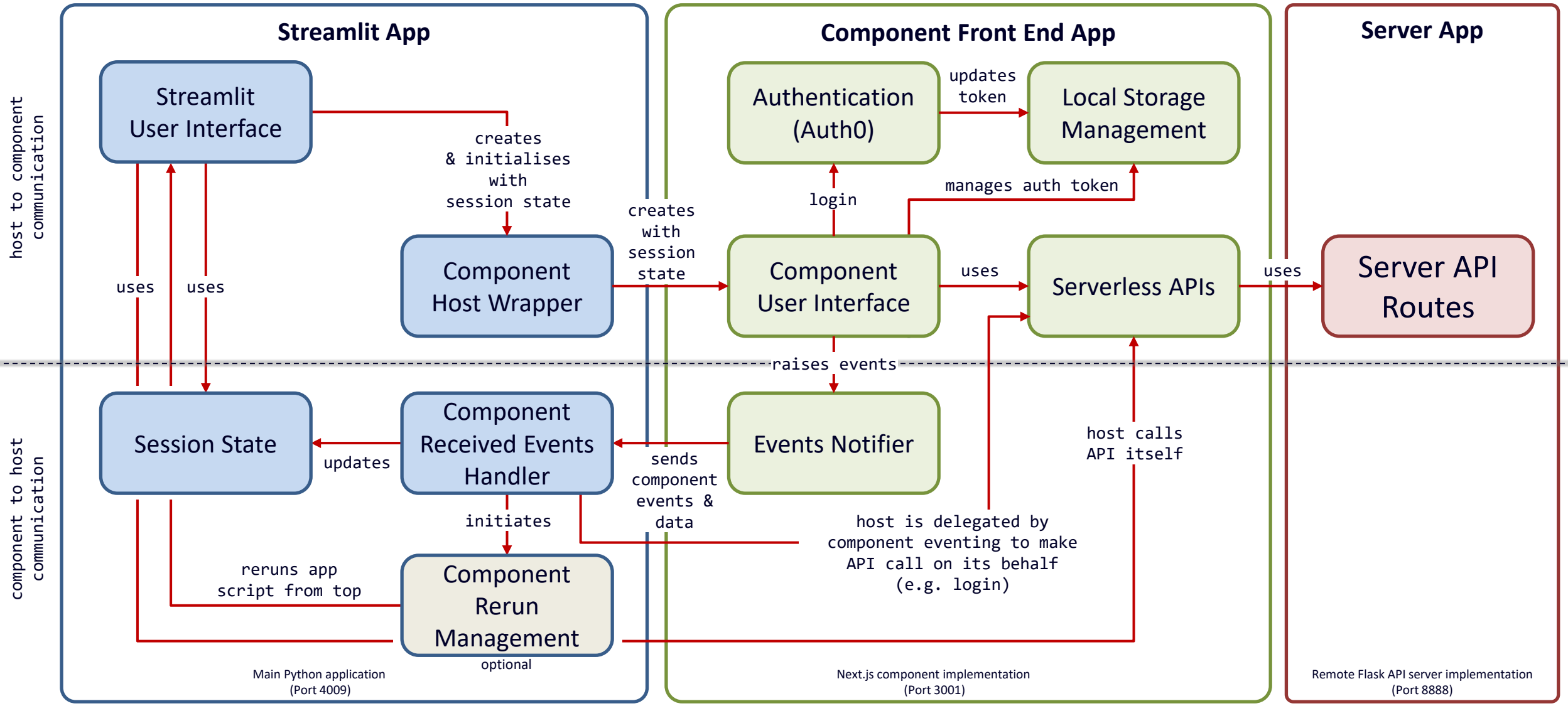
Server App

Remote Flask API server implementation
Port 8888

- Server API Routes

https://github.com/asehmi/Data-Science-Meetup-Oxford/tree/master/StreamlitComponent

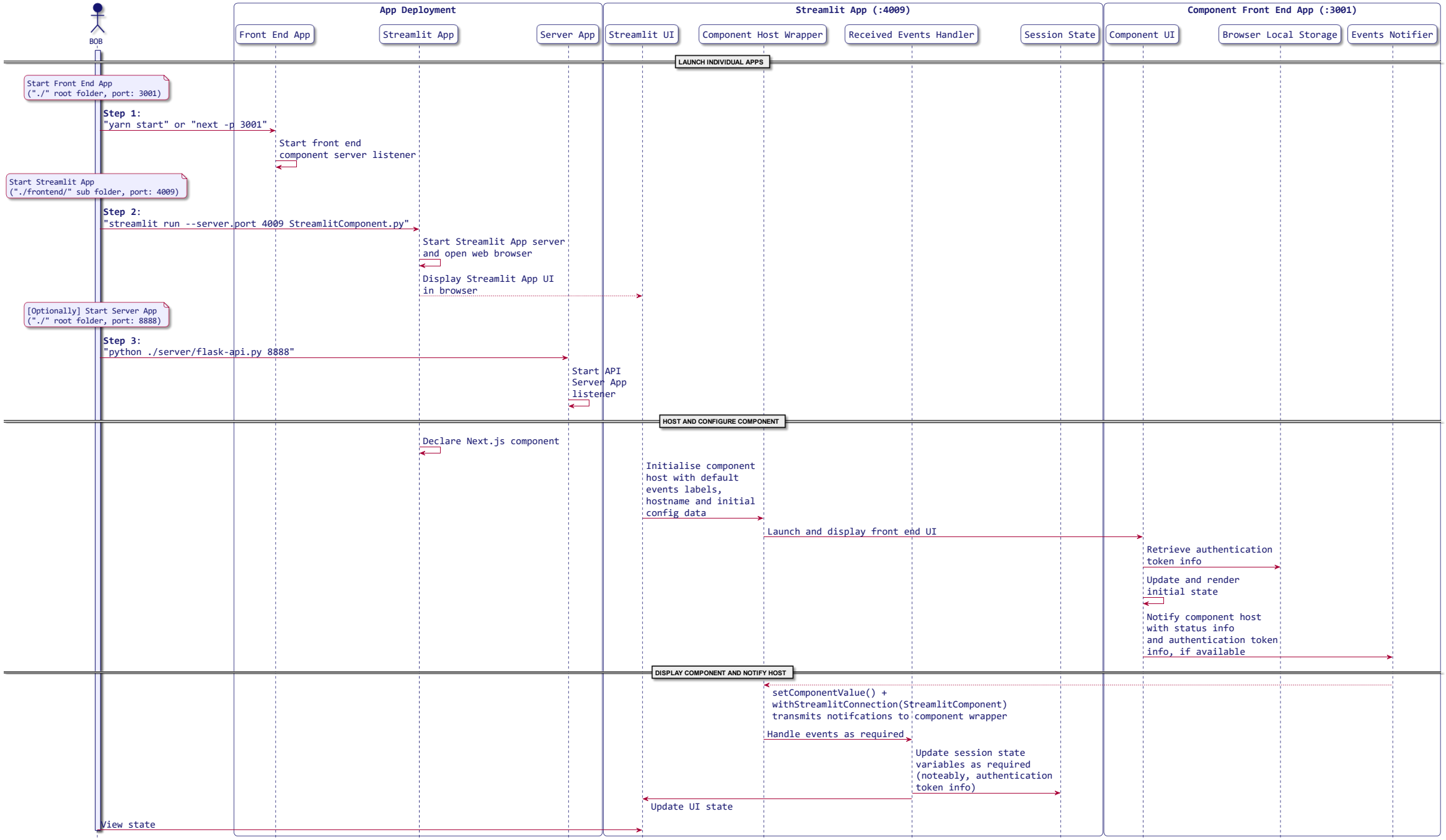
Structure


Streamlit App
Component Front End App
Server App
host to component communication
component to host communication
Streamlit User Interface
creates & initialises with session state
Component Host Wrapper
creates with session state
Authentication (Auth0)
updates token
Local Storage Management
login
manages auth token
Component User Interface
uses
Serverless APIs
uses
Server API Routes
uses
uses
raises events
Session State
updates
Component Received Events Handler
sends component events & data
Events Notifier
host calls API itself
initiates
reruns app script from top
Component Rerun Management
optional
host is delegated by component eventing to make API call on its behalf (e.g. login)
Main Python application (Port 4009)
Next.js component implementation (Port 3001)
Remote Flask API server implementation (Port 8888)

Running the sub-systems

Streamlit, Component and Server apps

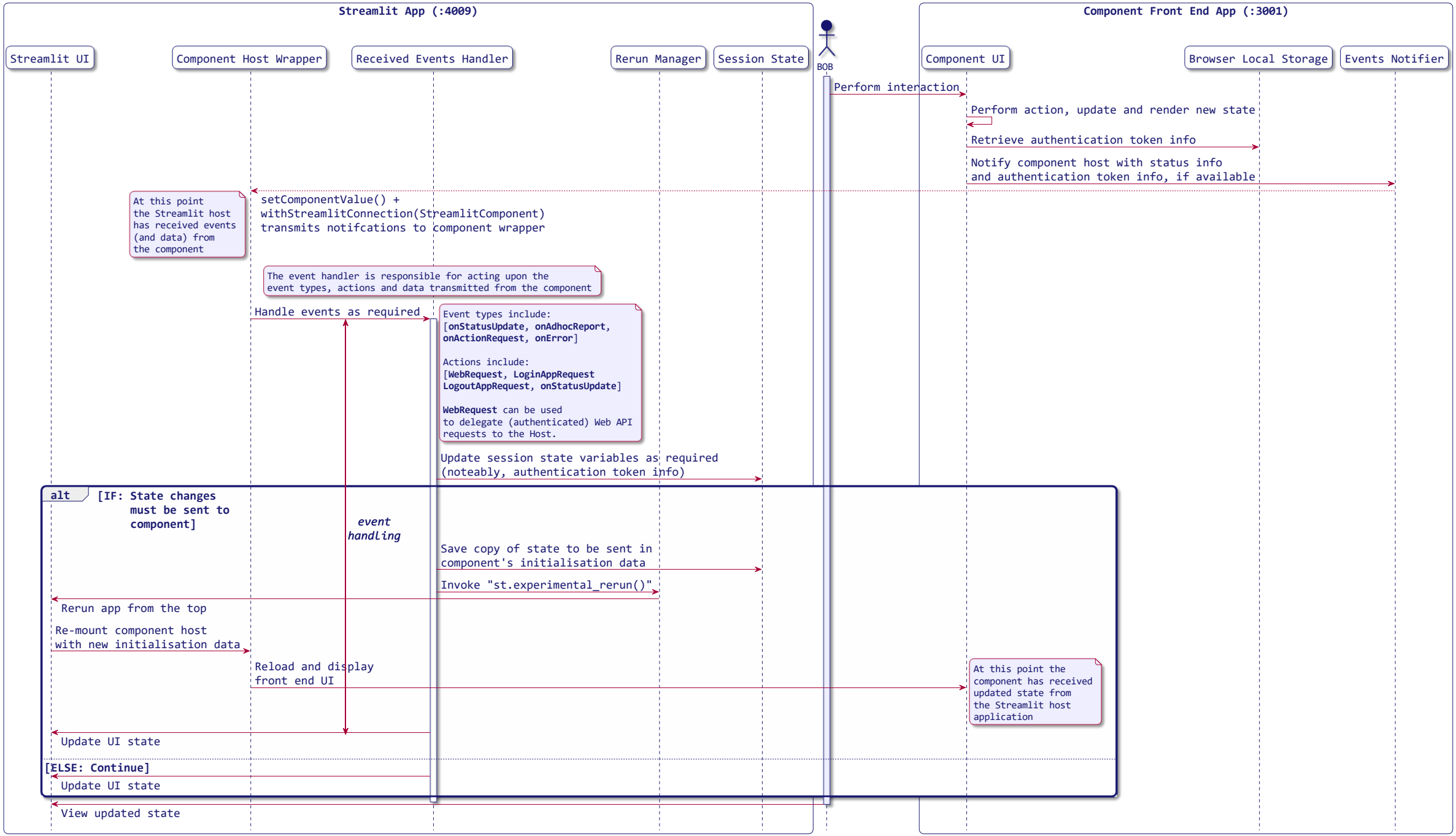
APP START-UP SEQUENCE



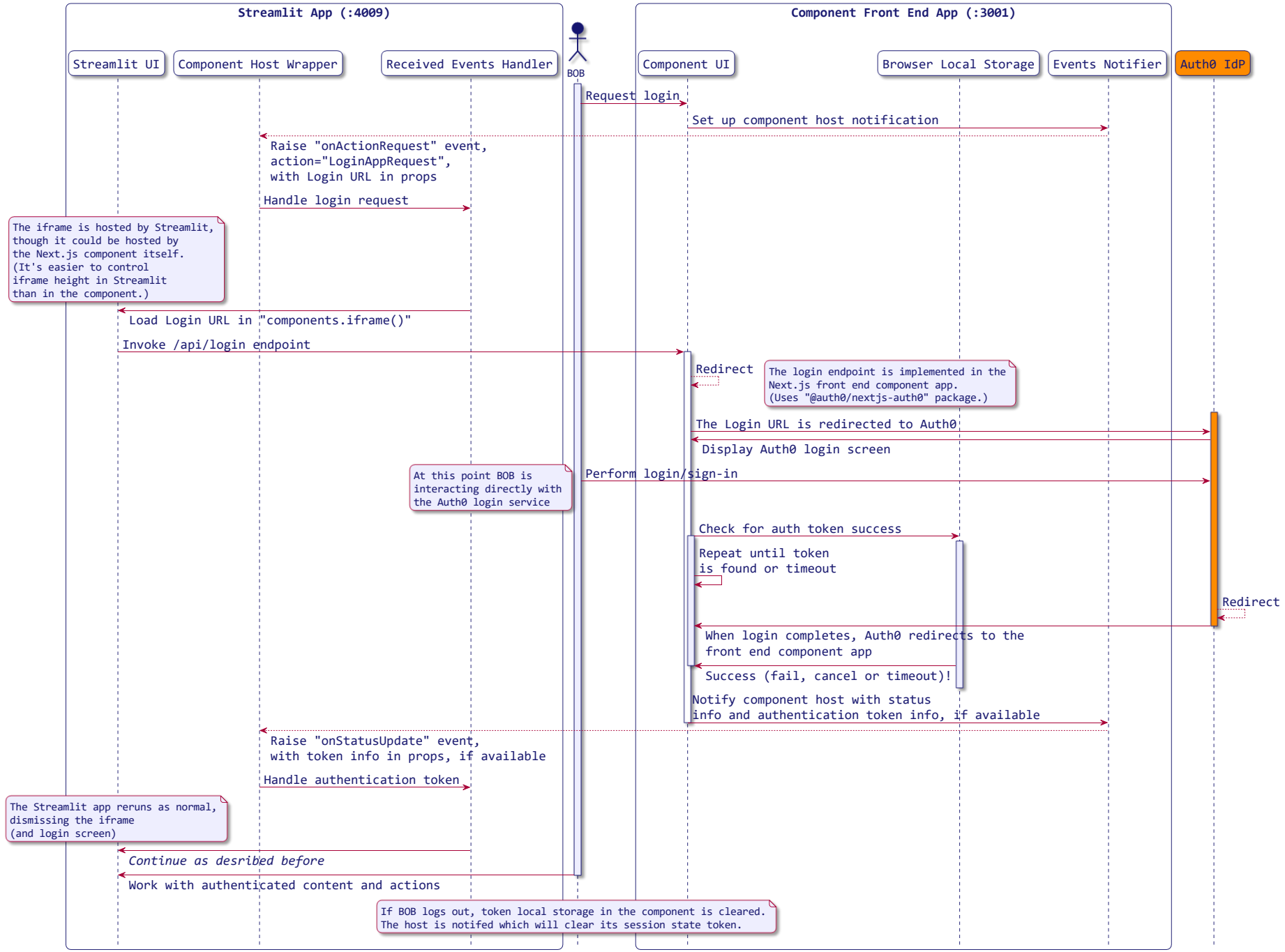
Communication patterns

Bi-directional messaging, Web APIs and Authentication

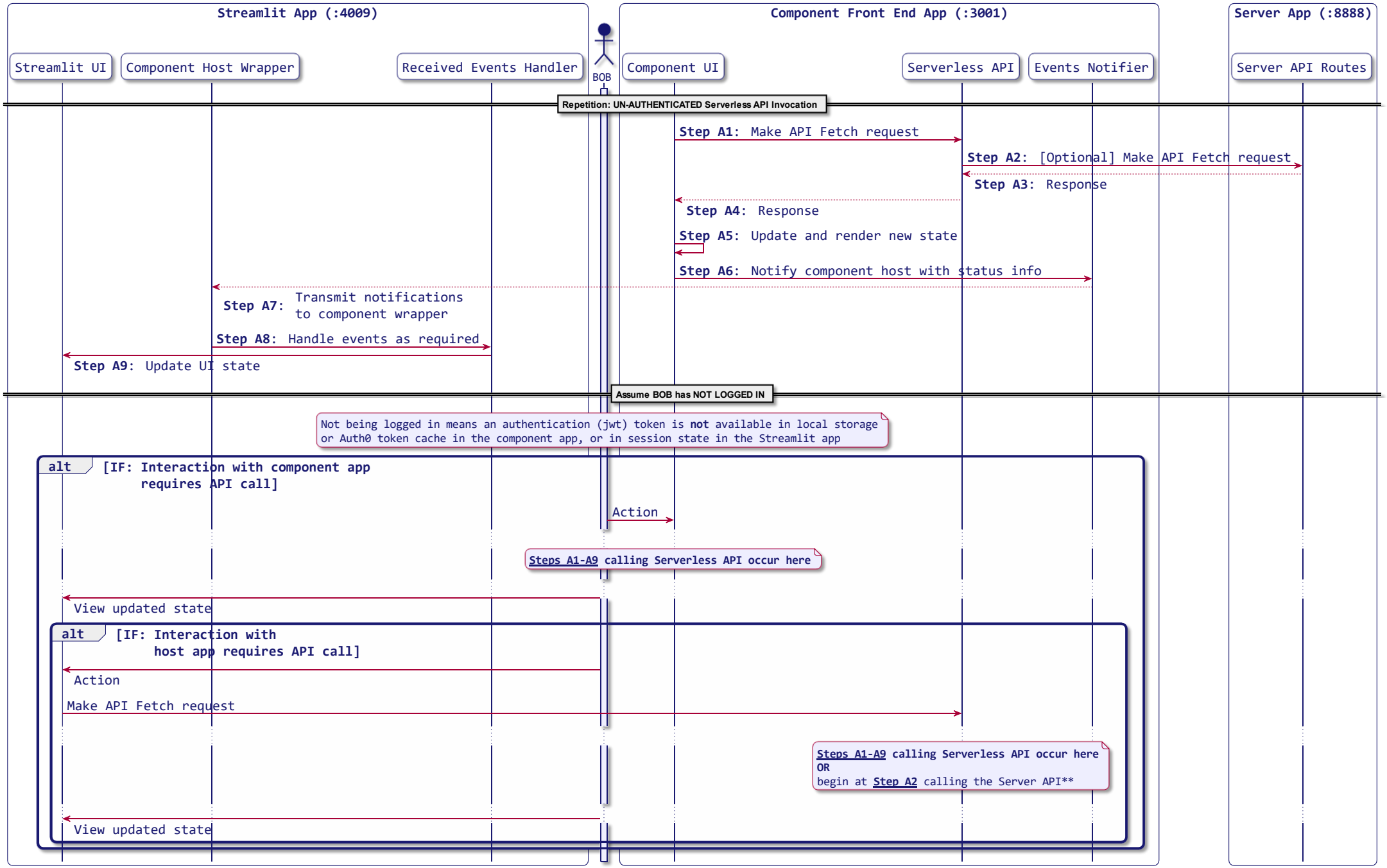
BI-DIRECTIONAL STREAMLIT TO COMPONENT MESSAGE PASSING



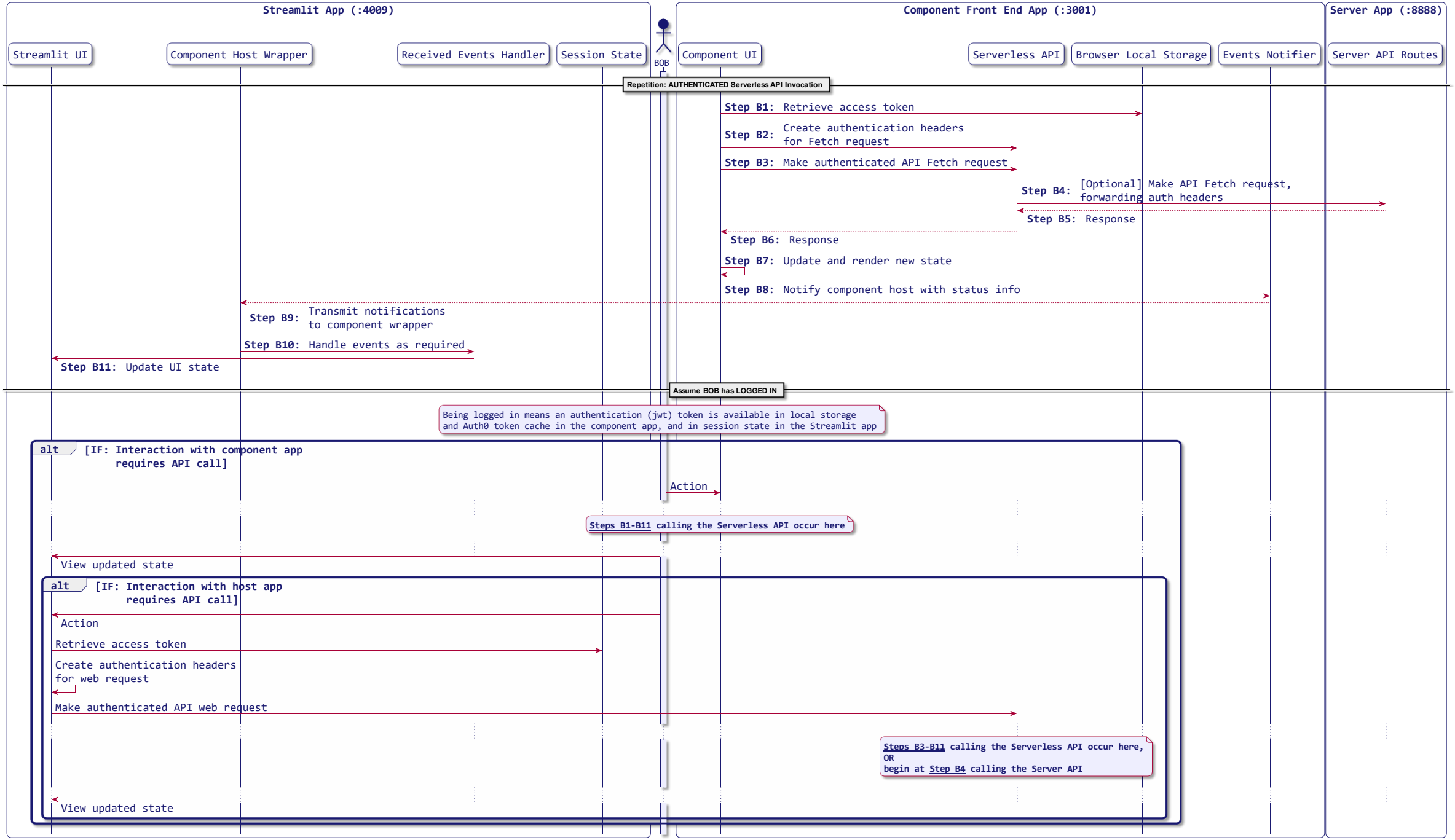
AUTHENTICATION WITH AUTH0 IDENTITY PROVIDER



INVOKING APIs (WITHOUT AUTHENTICATION)



INVOKING APIs (WITH AUTHENTICATION)



Thank You

Arvindra Sehmi

asehmi@oxfordeconomics.com

www.linkedin.com/in/asehmi/