# Virtual Stage Academy (VSA) Overview

## 1. OBS as the Hub

OBS (Open Broadcaster Software) serves as the core application, where all overlays, media inputs, and dynamic elements are managed. It streams the final presentation to your chosen platform(s) or back to Zoom.  
  
- Why OBS is Central:  
 - Supports multiple platforms: Zoom, YouTube, Facebook, Twitch, etc.  
 - Integrates web-based overlays via browser sources.  
 - Highly customizable for professional-quality live production.

## 2. Data Sources: Chat and Emoji Data

VSA captures real-time interaction data (chat, emoji reactions) from platforms to enhance audience engagement during presentations.  
  
Supported Platforms:  
- Zoom: Via webhooks (e.g., `meeting.chat\_message\_sent`, `meeting.reaction\_added`).  
- YouTube: Via Live Chat API.  
- Facebook: Via Graph API.  
- Twitch (optional): Via Chat API.

## 3. Backend Infrastructure

The backend processes data from platforms, transforming it into actionable formats for overlays.  
  
a. Webhooks and APIs:  
- Webhooks: Platforms like Zoom push events (e.g., chat messages) to Vercel-hosted endpoints in real time.  
- APIs: For platforms like YouTube and Facebook, data is fetched periodically or via WebSocket connections.  
  
b. Centralized Repository:  
- GitHub: For version control and script collaboration.  
- Vercel: For deploying scalable webhook endpoints and APIs.  
  
c. Processing Layer:  
Backend scripts (Python/Flask) process raw inputs:  
- Parse chat messages, emojis, and reactions.  
- Apply logic for overlays (e.g., word clouds, heatmaps).  
- Send processed data to frontend overlays via WebSocket streams or RESTful endpoints.

## 4. Frontend Infrastructure: Overlays

Dynamic overlays display processed data as engaging visuals. Hosted on Vercel, they integrate seamlessly with OBS as browser sources.  
  
Overlay Types:  
1. Chat Overlay: Displays live chat messages.  
2. Word Cloud Overlay: Converts chat messages into visual collages.  
3. Emoji Overlay: Animates reactions dynamically.  
4. Heatmap Overlay: Visualizes participant locations.  
  
Development Tools:  
- Built with HTML, CSS, and JavaScript (or frameworks like React).  
- Data fetched via WebSocket or REST endpoints.

## 5. Workflow: How It All Fits Together

Step-by-Step Flow:  
1. Platform Activity: A participant sends a chat message or emoji reaction.  
2. Data Capture: Webhooks (Zoom) or APIs (YouTube, Facebook) capture the event.  
3. Data Processing: The backend applies logic to transform data into usable formats.  
4. Data Transfer: Processed data is sent to overlays via WebSocket or REST endpoints.  
5. Overlay Rendering: The overlay dynamically renders visuals in real-time.  
6. OBS Integration: OBS displays overlays as browser sources.  
7. Broadcast: OBS streams the final output to platforms like Zoom, YouTube, or Facebook.

## 6. Deployment and Hosting

Component Locations:  
- Development: Scripts are created locally and stored on GitHub.  
- Hosting:   
 - Vercel: For webhooks, APIs, and overlays.  
 - OBS: Browser sources use Vercel-hosted URLs.  
  
Scaling Across Platforms:  
- Centralized backend logic handles multiple platforms.  
- Overlays remain platform-agnostic, working across Zoom, YouTube, Facebook, and more.

## 7. Tools Overview

- OBS: Central application for live streaming and overlay management.  
- Zoom Webhooks: For real-time chat and emoji data.  
- YouTube/Facebook APIs: For fetching live interaction data.  
- Flask: Backend framework for webhook and API processing.  
- Vercel: For hosting webhooks and overlays.  
- GitHub: Version control and collaboration.  
- Python Libraries: For data parsing and processing (e.g., Flask, requests).

## 8. Future Considerations

1. Unified Data Pipeline: Consolidate backend logic into a single microservice for all platforms.  
2. WebSocket Integration: Enable real-time updates for smoother visuals and overlays.  
3. Customizable Overlays: Pre-configured overlays with easy setup for students.  
4. Scalability: Ensure infrastructure supports high-volume live sessions.