

# Chorus: Coordinating Data Across Multi-Device Data Visualizations

Stephen Wu, Dr. Arnab Nandi // Interactive Data Systems Group, The Ohio State University

## Overview

**Chorus** is a data model and software that presents a novel method of using web sockets to create **chatrooms for data**. Each Chorus server coordinates data across multiple clients and rooms. Any device, projector, or monitor can act as a node in this network of data rooms.

**Problem:** Despite a myriad of browser-based visualization tools, few open-source solutions exist to facilitate real-time data collaboration on the web. Existing solutions are often specialized or expensive, and they fail to solve the problem of intuitive data sharing.

**Solution:** Chorus presents a generic data model and Node.js server that allows for easy integration with any JavaScript-based data visualization. By coordinating shared data and client actions, Chorus adds another layer to turn regular browser-based tools into a collaborative experience.

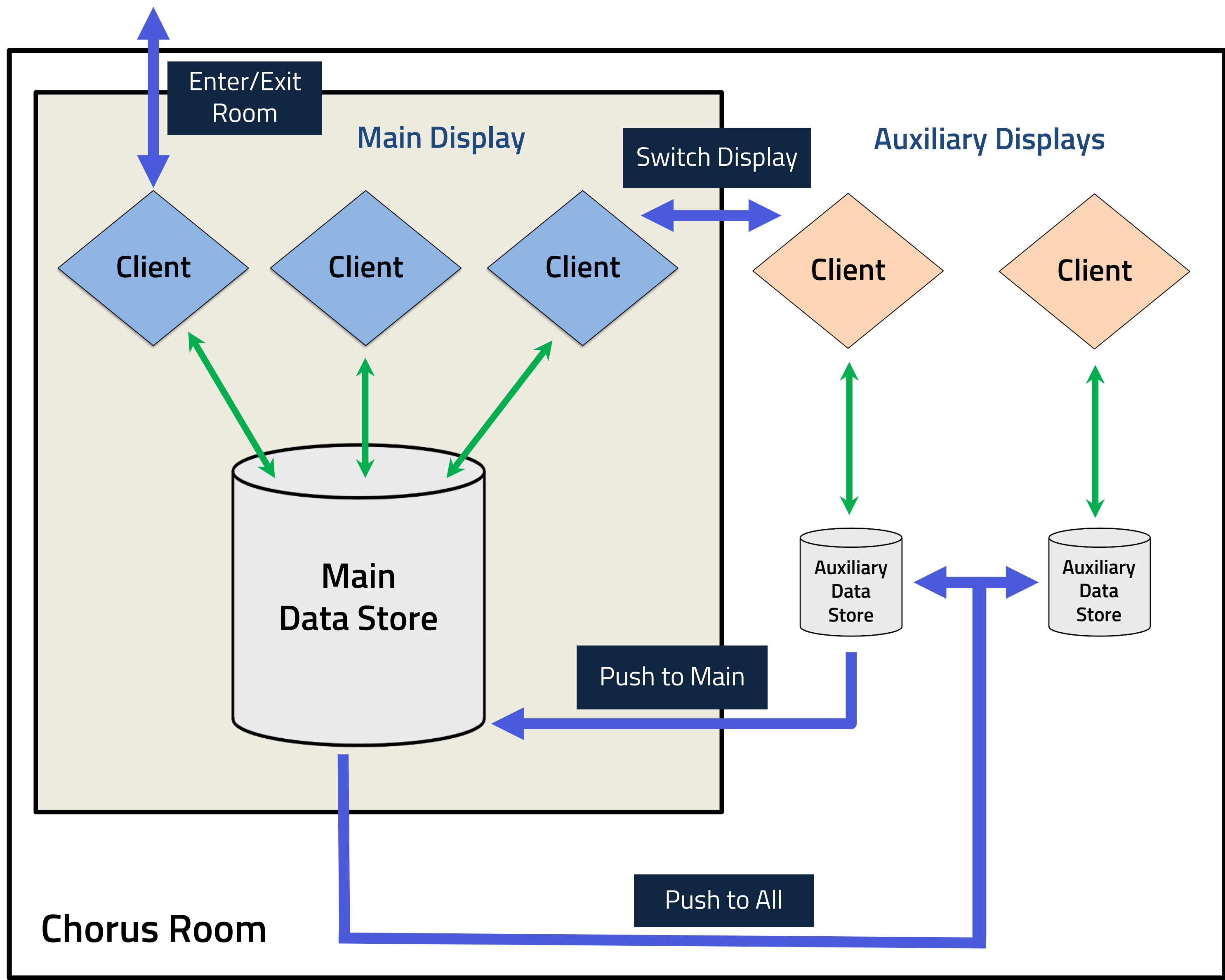
## Features

- Multi-device data coordination for any JavaScript-based data visualization or tool, easily integrated into code
- Server facilitating numerous clients and channels, separating data stores securely with low latency
- Menu to allow users to easily create or follow data rooms and detach if desired
- Built-in Chromecast integration, allowing any monitor or projector to become an synced external display
- Custom backend to add additional socket events or server-sided data processing

## Technology

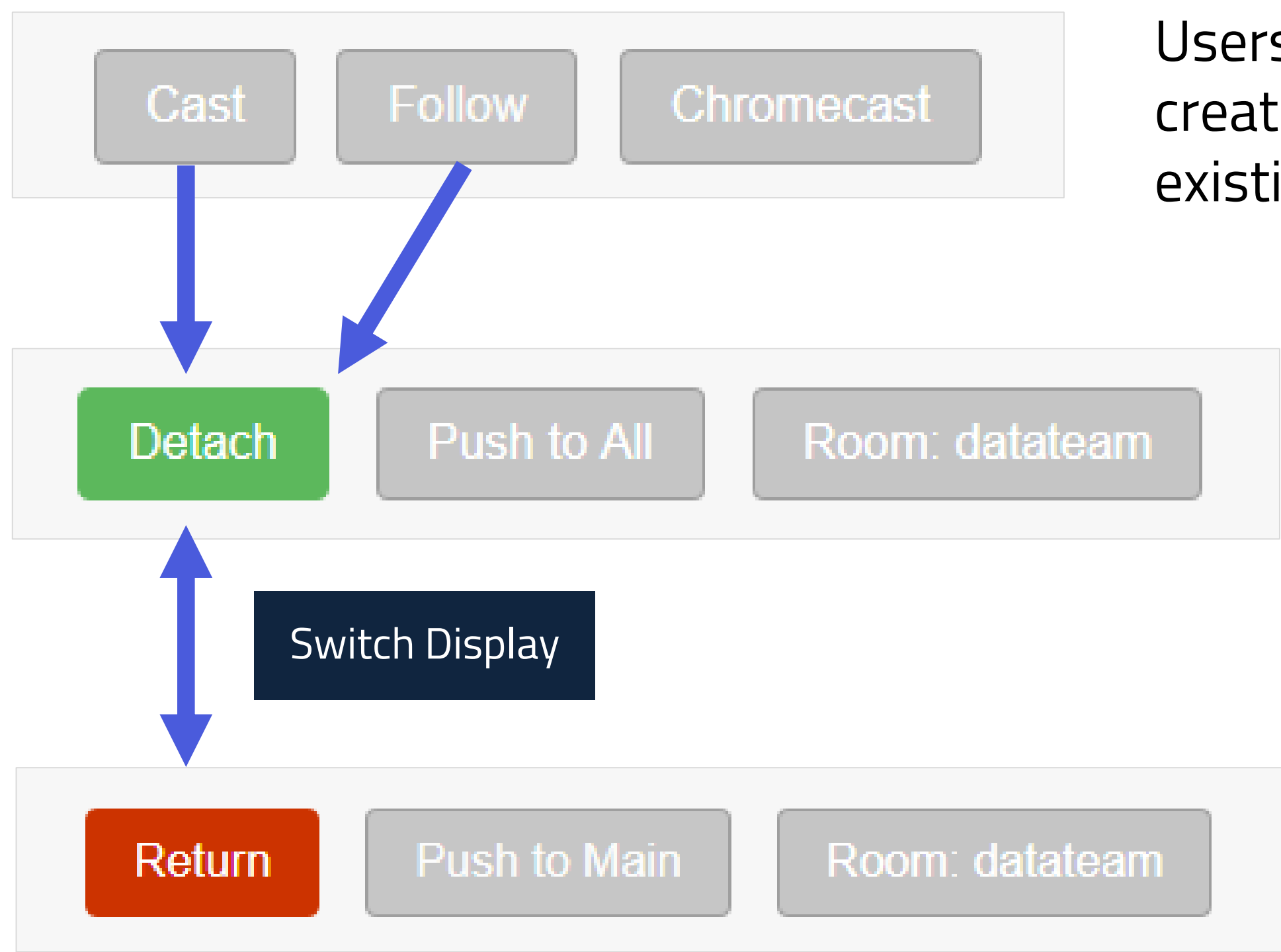


## Model



## Software

After a developer integrates Chorus, they can choose to inject the Chorus menu into the window which they can style as desired. This menu allows the client to access and collaborate on the Chorus server.



Users can **Cast** or **Chromecast** to create new rooms or **Follow** to join existing rooms.

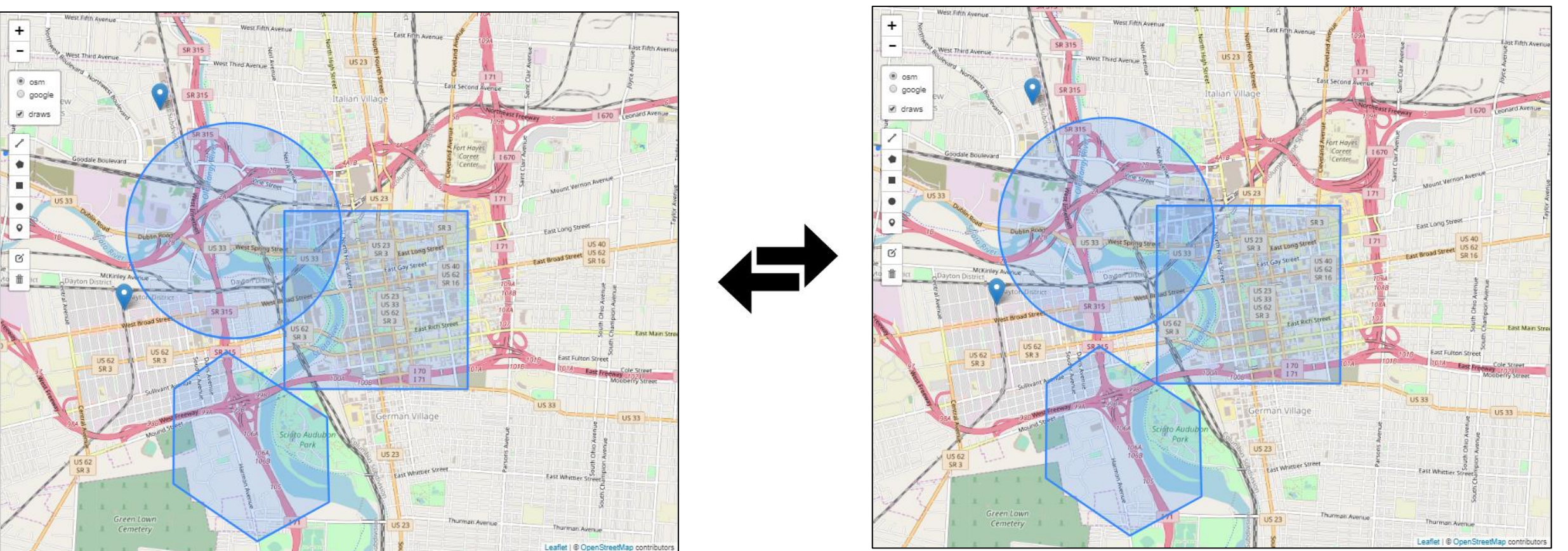
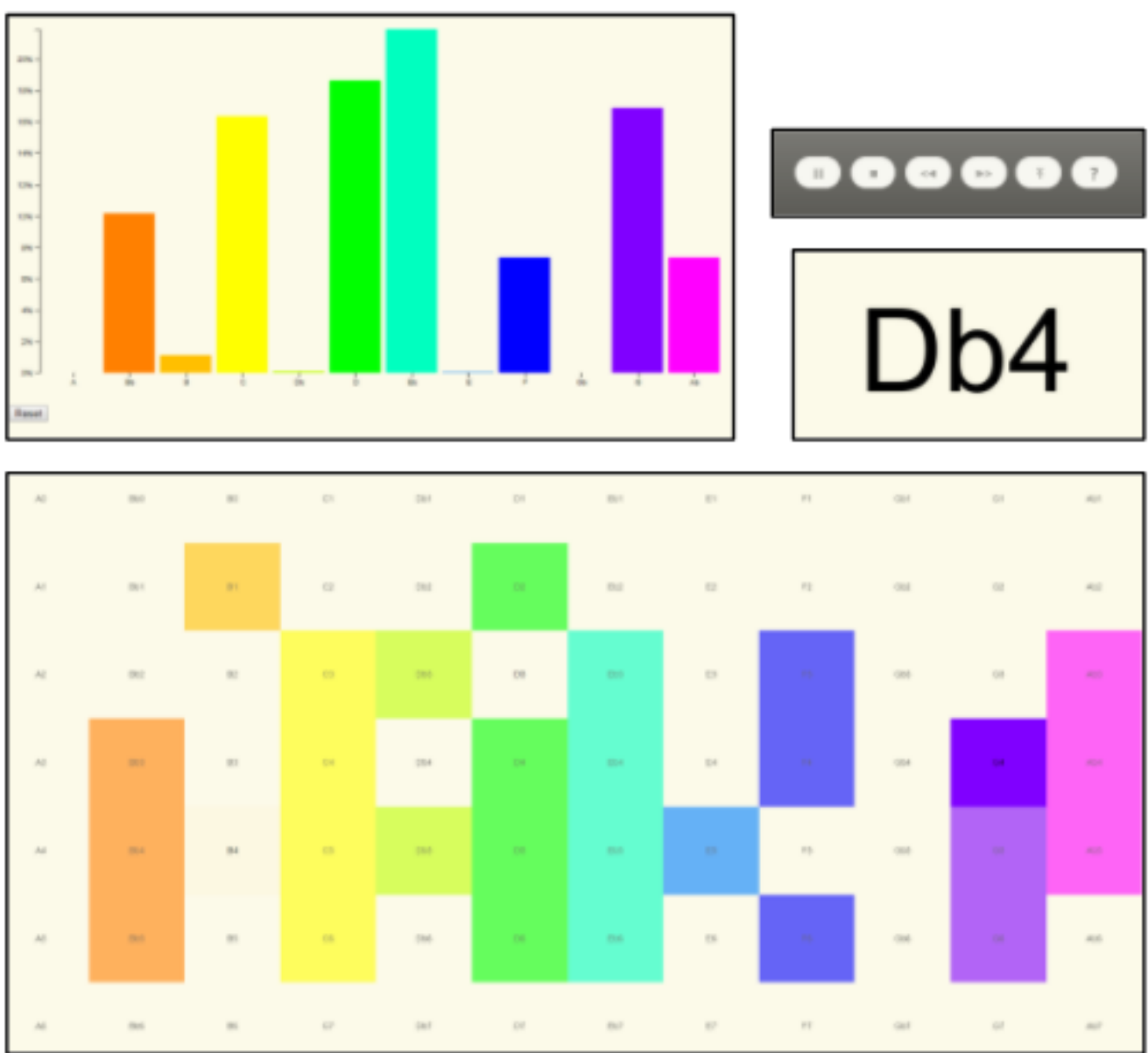
After entering a room, users will be synced to all the changes and actions made in the room in the Main display.

Users can **Detach** to enter an Auxiliary display where data is not synced until they are satisfied and can either **Return** or **Push to Main**.

## Use Cases

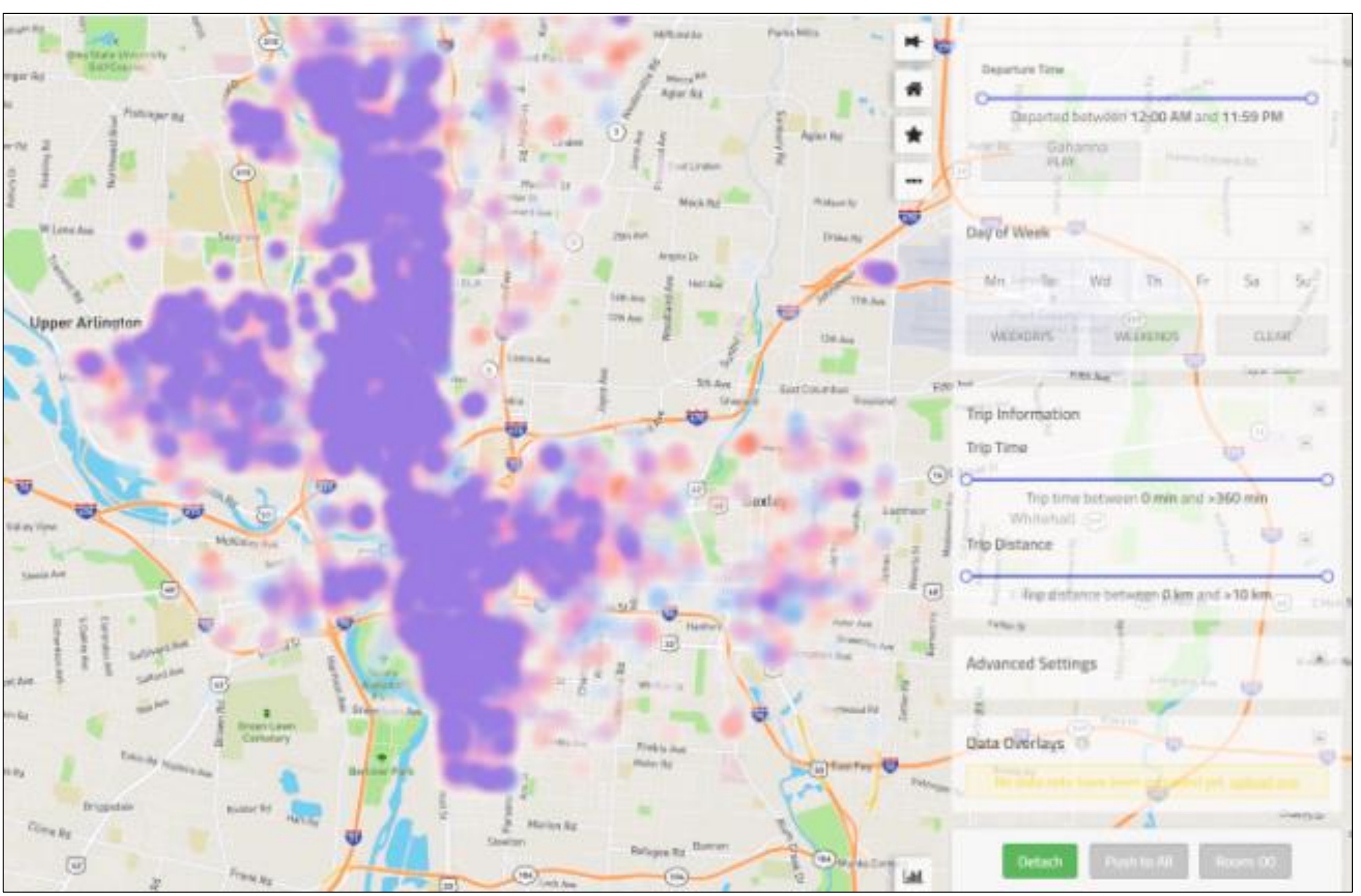
### (1) Collaborative Musical Keyboard (MIDI.js)

This demo shows a real-time piano that reads MIDI input and plays touched notes (e.g. Db4), logging the frequency of the notes played. Multiple users and devices could play on the same grids simultaneously. In this demo, custom server-sided events (noteOn, noteOff) were also demonstrated.



### (2) Map and Annotations (Leaflet.js)

This demo tackles the idea of collaborating with shared map annotations marking locations of interest. Use-cases for synced maps include annotating medical images or real-time events.



### (3) Spatiotemporal Visualization Tool (React, Redux, Crossfilter.js, Leaflet.js)

This tool, named Crossroads, was independently developed by members of the Interactive Data Systems group to visualize large space-time datasets, including car fleet data. Chorus was integrated to allow for real-time collaboration.