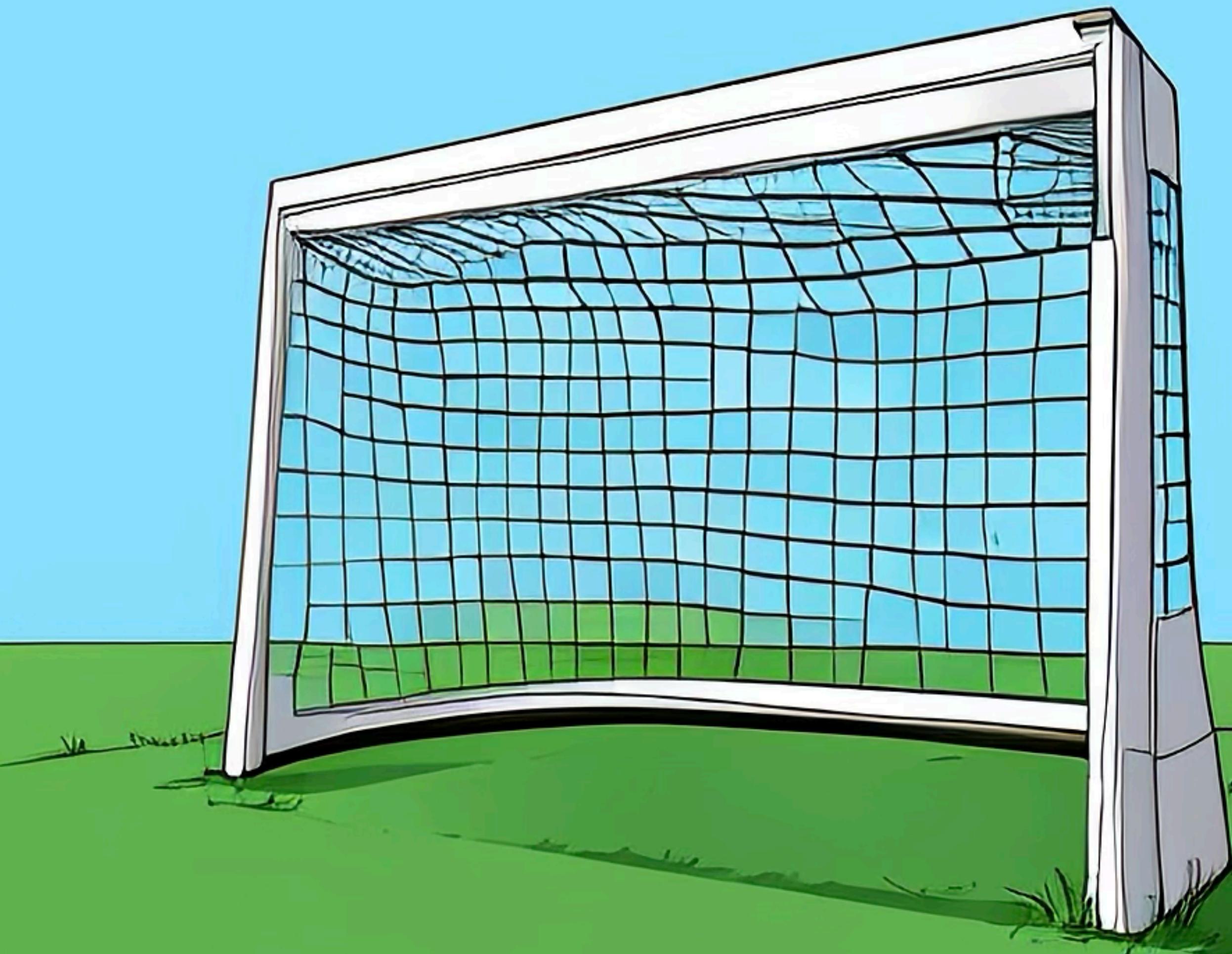


The stats and vote services

Section thirteen

The goal of this
section

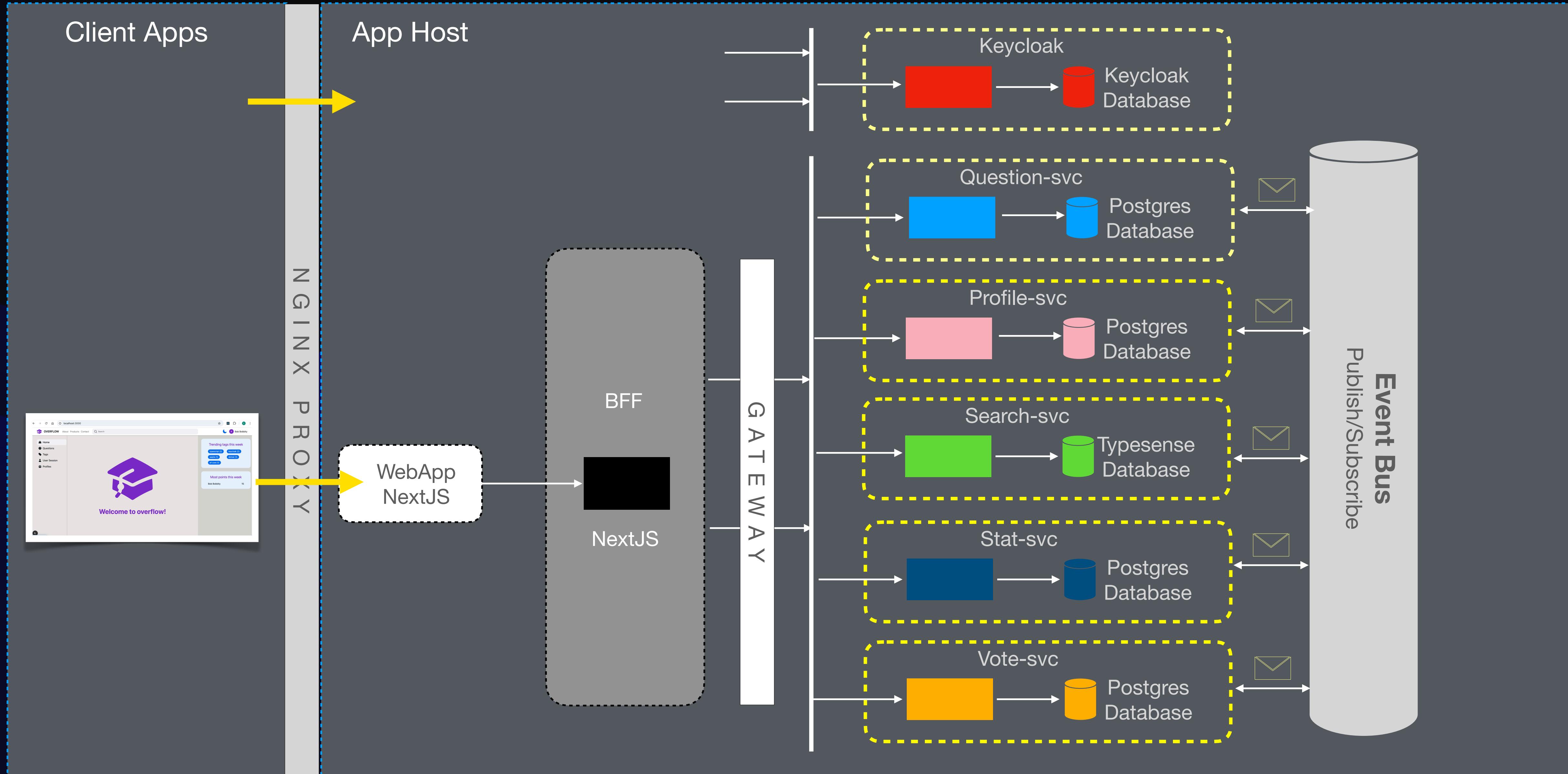


To create 2 new services, one for Stats
and one for Votes

To implement event sourcing for the
Stats service which gives us data over
time e.g top users last 7 days by
reputation

Implement the voting functionality

Overflow Architecture



Demo

Event sourcing

Event sourcing



How many rep points did Bob get on Tuesday?

Event sourcing

Traditional persistence

- Save the current state of an entity
- Perfect for CRUD systems

Event sourcing

- Save series of events that happened
 1. Question asked
 2. Answer added
 3. Vote cast
 4. Etc
- Current state is rebuilt by replaying events in order
- Full audit trail

WolverineFx Marten

- Marten is a .Net document and event store built on PostgreSQL
- Treats Postgres as a native event store
 - Events are appended to a stream (per aggregate, e.g one stream per question)
 - Each event is just JSON in Postgres
- Query events like documents, or rebuild aggregates using projections
- Marten handles:
 - Event storage
 - Projections

Pros and cons

- Full history, audit trail
- Good fit for event driven systems
- Easy to create new “read models” without changing past data

- More moving parts (streams, projections, replaying)
- State reconstruction slow without snapshots
- Overkill for CRUD where history not important

Expectations

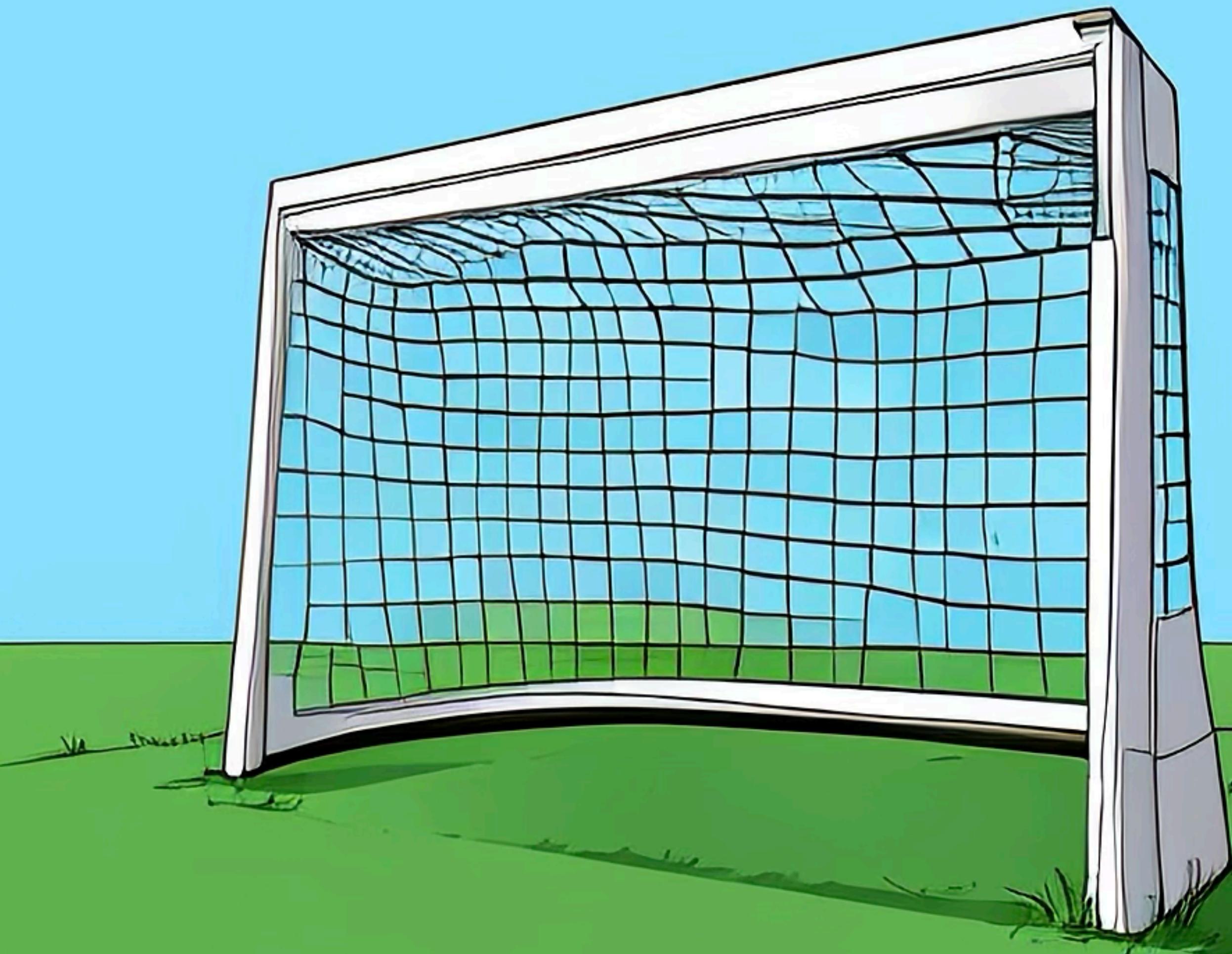
A photograph of two rabbits, one white and one cream-colored, sitting side-by-side in a dark, circular opening of a tunnel made of green artificial grass. They are facing towards the camera. The background is a bright green lawn.

What are the trending
tags last 7 days?

What are the top users
last 7 days?

Summary of section thirteen

The goal of this
section



To create 2 new services, one for Stats
and one for Votes

To implement event sourcing for the
Stats service which gives us data over
time e.g top users last 7 days by
reputation

Implement the voting functionality

Up next...