ACE 2.0 Architecture Design Document

Deployment Topology: Distributed Edge vs. Centralization

Ozgur Ural
Project Lead, ACE 2.0

June 25, 2026

Abstract

Objective: Establish the definitive deployment topology for the ACE 2.0 ecosystem across multiple simulation training centers.
Recommendation: I formally recommend adopting the **Hybrid Edge Topology**. This means maintaining a **Centralized Authentication Server** while deploying **Decentralized Edge Backends and Frontends** locally at each simulation center. We must avoid migrating to a centralized server model for the backend, as doing so would critically degrade real-time telemetry performance. Simultaneously, we must avoid decentralizing the Auth server to prevent fragmented identity management.

1 1. Background

The ACE 2.0 platform is scaling to support geographically distributed simulation centers (e.g., Luton, Mumbai, HQ). As documented in the project's `README.md`, ACE ships a Python gRPC backend, a SvelteKit web client, and supporting services including Redis and PostgreSQL. The system currently streams high-frequency telemetry via bidirectional gRPC from simulators to the backend for real-time monitoring and engineering diagnostics. The Authentication service (`avion-authenticator`) operates independently via FastAPI.

2 2. Problem Statement

The core architectural decision facing the engineering team is whether to consolidate components (Frontend, Backend, and/or Auth) into a single centralized server at HQ, or to deploy independent stacks at the edge within each simulation center's Local Area Network. This decision will permanently impact simulator diagnostic fidelity, identity management, and engineering velocity.

3 3. The Recommendation: Hybrid Edge Topology

The recommended approach is a **Hybrid Edge Topology**. This is explicitly defined as:

- **Centralized Authentication:** A single global Auth service (`auth-hq`) handling identities and passwords for all personnel.
- **Decentralized Edge Stacks:** A fully isolated ACE stack (Backend, Frontend, PostgreSQL, Redis, Envoy Proxy) deployed on-premises at the Local Area Network (LAN) of *each* simulation center. This single local stack acts as a centralized hub for that facility, efficiently aggregating real-time telemetry from multiple local simulators (e.g., 8 simulators in Mumbai) over the ultra-fast LAN.
- **Unified Routing:** The SvelteKit frontend runs locally, and dynamically connects the client to the local edge backend based on the "Sim Center" selection made at the login screen.

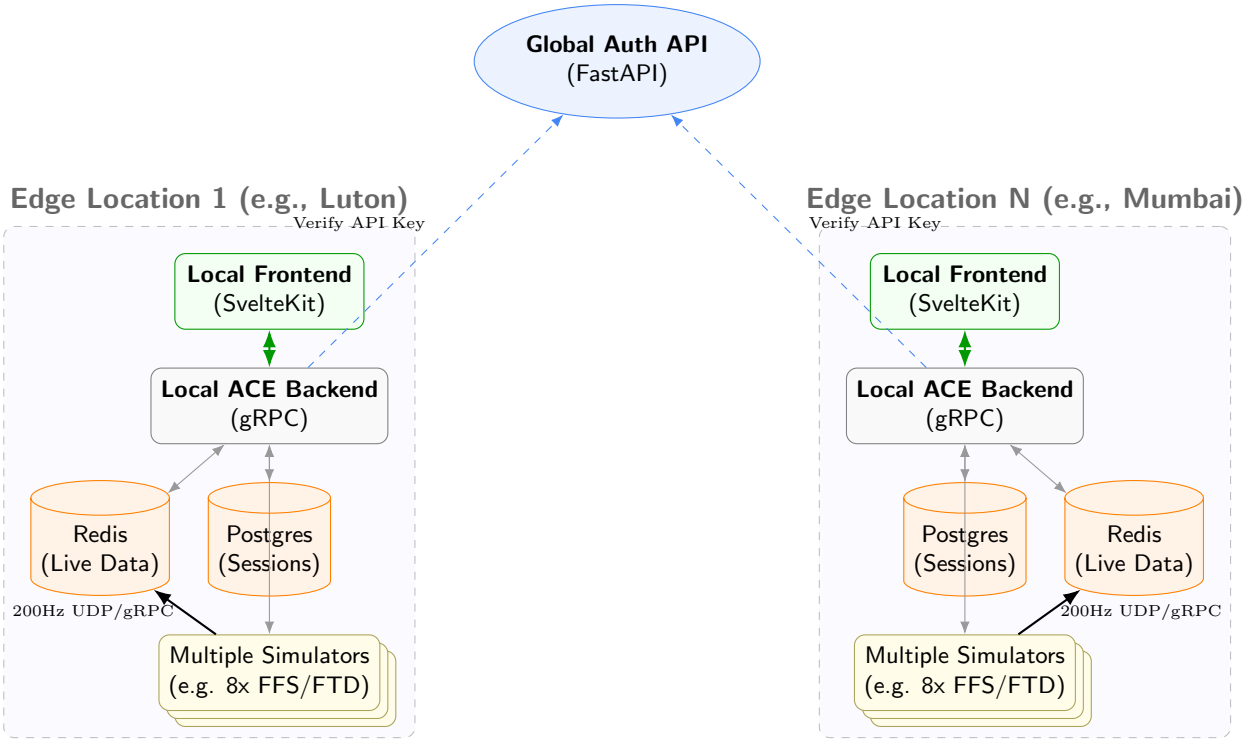

Figure 1: System Context: Hybrid Edge Topology with Localized Stacks

4 4. Analysis of Alternative Topologies

During our evaluations, we considered several architectural permutations, all of which introduce critical failure points or operational inefficiencies:

4.1 4.1. Fully Centralized (Central Auth + Central BE + Central FE)

This approach routes all components and telemetry to a single central server. While Identity Management is unified, it fails on two critical fronts: **Severe Latency** (transporting 200Hz telemetry over WAN breaks real-time diagnostics) and **Multi-Tenancy Debt** (requires rewriting the entire database and Redis schemas to avoid data collisions).

4.2 4.2. Centralized Backend + Edge Frontend

In this model, the frontend runs locally, but the backend is centralized. The telemetry still must traverse the WAN to reach the central backend. This model suffers from the exact same **Latency** and **Multi-Tenancy** blockers as the fully centralized model.

4.3 4.3. Centralized Frontend + Edge Backend

In this model, the backend remains at the edge (solving latency and multi-tenancy), but the SvelteKit frontend is hosted on a single central server. While data stays local, this destroys **High Availability**. If the WAN link drops, local technicians cannot load the web interface to monitor their simulators, rendering the operational local backend useless because the UI is inaccessible.

4.4 4.4. Fully Decentralized (Edge Auth + Edge BE + Edge FE)

In this model, even the Authentication Server is pushed to the edge, meaning every Sim Center has its own completely isolated Auth database. This approach fails due to **Fragmented Identity Management**. Technicians or engineers traveling between HQ, Luton, and Mumbai would require separate accounts and credentials for each location. Centralizing the Auth Server is critical to maintain Single Sign-On (SSO) and unified global access control.

4.5 4.5. Summary of Constraints

To summarize the engineering constraints that invalidate the alternatives above:

- **Latency:** ACE requires $<$ 1ms LAN transmission for 200Hz gRPC streams.
- **Engineering Debt:** The codebase is strictly single-tenant. Attempting to host multiple centers on one backend risks catastrophic namespace collisions (e.g., a bus designated `FTD1:FAVT` in Luton overwriting the telemetry of `FTD1:FAVT` in HQ).
- **Fault Tolerance:** Sim Centers must operate autonomously. Technicians must be able to load the frontend and connect to the backend even during complete internet outages.
- **Identity Management:** Passwords and roles must be synchronized globally.

5 5. Anticipated Q&A

Q: If the Auth Server is centralized, how do technicians continue working if the WAN/internet goes down?
A: While the initial login handshake happens centrally, the active Session UUIDs are generated and enforced by the *local* Edge PostgreSQL database. If the WAN drops, existing active sessions survive uninterrupted because the local Backend validates them locally. For new logins during an outage, the edge stack can fall back to an emergency local cache or offline admin credentials.

Q: Doesn't deploying multiple Edge stacks increase IT and maintenance complexity compared to one central cloud backend?
A: While it increases the physical footprint, it completely isolates failure domains. Managing multiple stacks is a solved infrastructure problem addressed by CI/CD automation and containerization (Docker). Conversely, bypassing the speed of light to solve 200Hz WAN latency, or risking catastrophic data collisions via a rushed multi-tenancy rewrite, are nearly unsolvable architectural blockers.

6 6. Conclusion

To ensure the high-fidelity performance required for aviation simulation diagnostics, prevent incurring massive engineering debt, and maintain unified security policies, we must retain our localized deployment strategy with central identity. The **Hybrid Edge Topology** leverages a Centralized Auth service for unified access control, while deploying Edge Backends and Frontends to respect the physics of real-time data transmission and ensure local fault tolerance.