

Implement strict backend call state machine (IDLE, FINDING_PARTNER, CALLING, RINGING, CONNECTED, ENDED)
Implement backend enforcement so only IDLE users are callable
Implement user call state validation in every API
Implement FINDING_PARTNER flow with backend controlled state
Implement DB/Redis locking to prevent race conditions
Implement atomic transactional call state updates
Implement hard blocking of parallel incoming calls once user is RINGING
Implement strict rule: one active call per user at any time
Implement ENDED → IDLE automatic reset logic on server
Implement guaranteed reset on reject / missed call cases
Implement backend protection against duplicate, ghost, or concurrent calls
Implement backend validation instead of relying on UI filtering

=====

each point use case

:

To ensure backend is single source of truth and avoid UI-based control failures
To guarantee only eligible users can receive calls and prevent unwanted ringing
To avoid race conditions and double connections in high concurrency environments
To maintain clean and predictable call flow without random behavior
To prevent ghost calls, duplicate calls, and stuck states
To ensure smooth, stable, and professional call experience for users
To block parallel incoming calls and avoid confusion
To make system highly reliable, scalable, and production ready
To automatically recover users to IDLE and avoid permanently locked states
To improve call stability, signaling accuracy, and user trust