

Project 50

Executive Feasibility Snapshot

Comparative assessment of backend infrastructure, native voice APIs, and broader RTC room models for a live multiplayer quiz app

Product concept

Project 50 is a moderator-led live quiz app for 4 to 8 participants. Each match contains 50 questions and combines buzzer rounds, estimation questions, live scoring, and synchronized session flow.

The concept phase tested three service categories: Firebase for real-time gameplay infrastructure, voice-only APIs for native in-app audio, and broader managed RTC room models as a benchmark for fully embedded communication.

Executive decision

Proceed with Firebase-backed live gameplay, moderator controls, and score synchronization.

Exclude native in-app voice from the initial scope and rely on third-party platforms where voice is required.

Reason: the backend model remains commercially manageable, while voice and full RTC room services create a materially higher recurring cost base because they scale by connected participant minutes rather than quiz logic.

Firebase backend

Realtime session sync,
moderator state, answer
handling
Assessment: viable for MVP

Voice-only APIs

Native in-app audio between
connected participants
Assessment: technically
feasible, financially high-risk

Managed RTC rooms

Video + voice room model
used as a conservative upper
market benchmark

Assessment: not suitable for
MVP economics

Firebase backend

\$136

100,000 players / hype case

Agora Voice

\$35.6k

100,000 players / hype case

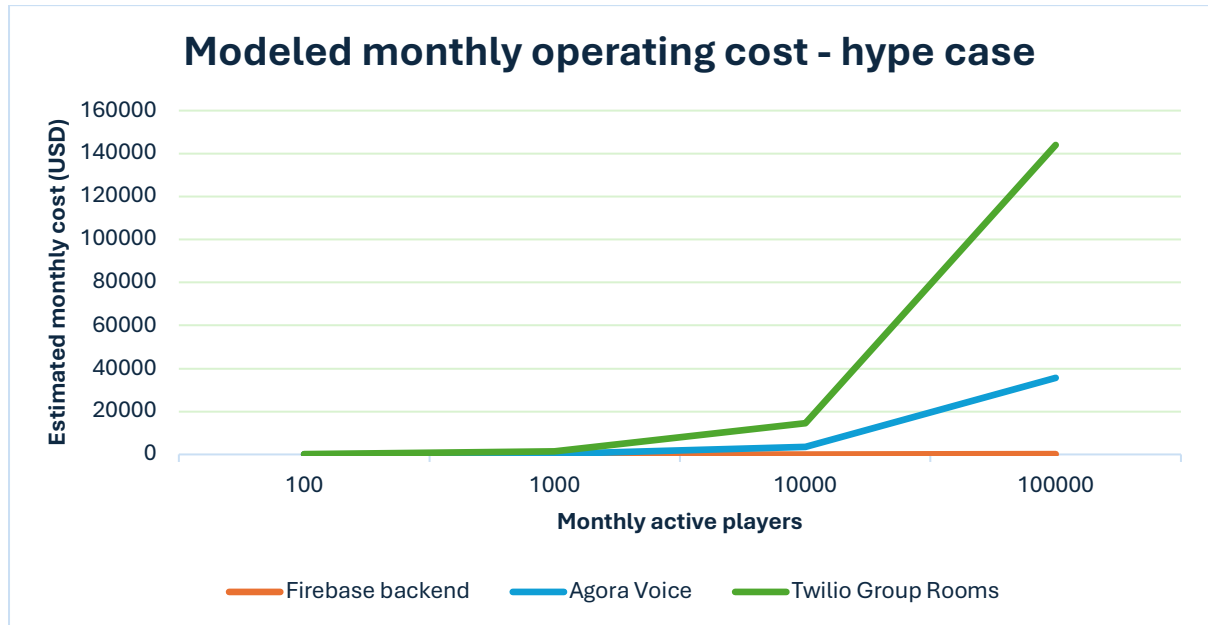
Twilio Group Rooms

\$144k

100,000 players / hype case

Cost-growth view

Hype-case model shown below; stepped player tiers are used to compare cost growth across all three service categories.



Note: the 100-player Firebase point is effectively \$0 because the benchmark remains within Firestore free-tier usage at that scale.

Benchmark model and management interpretation

Illustrative benchmark assumptions

- 4-8 players per match; model uses 6 average participants
- 50 questions per moderated match
- Normal case: 4 matches per user / month at 35 minutes
- Hype case: 8 matches per user / month at 45 minutes
- Firebase benchmark: about 2,500 reads, 300 writes, and 50 deletes per completed match
- Voice and RTC room benchmarks priced per participant minute using public list pricing
- Model excludes tax, negotiated discounts, premium add-ons, and unusual bandwidth spikes

Why the spread is so large

Firebase mainly scales with app transactions such as reads, writes, and deletes. Voice APIs and RTC room services scale with connected participant time, which turns session duration into a recurring cost driver. A successful user spike can therefore remain manageable on the gameplay layer while becoming expensive on the communication layer.

Twilio Group Rooms is included as a broader video + voice room benchmark rather than the intended MVP architecture. It serves as a deliberately conservative upper-cost market reference.

Decision framework

Monthly cost benchmark (USD)

Normal monthly usage benchmark

Monthly active players	Firebase	Agora Voice	Twilio Group Rooms
100	\$0	\$3.96	\$56
1,000	\$0.05	\$128.70	\$560
10,000	\$5.81	\$1,376.10	\$5,600
100,000	\$67.28	\$13,850.10	\$56,000

Hype monthly usage benchmark

Monthly active players	Firebase	Agora Voice	Twilio Group Rooms
100	\$0	\$25.74	\$144
1,000	\$0.55	\$346.50	\$1,440
10,000	\$12.62	\$3,554.10	\$14,400
100,000	\$135.62	\$35,630.10	\$144,000

Management recommendation

Project 50 should launch with Firebase-supported multiplayer gameplay only. Native in-app voice should remain out of scope until monetization, retention, or vendor pricing materially changes. What this document demonstrates: market comparison across multiple service categories, translation of architecture choices into operating-cost exposure, and disciplined scope reduction based on financial risk rather than technical preference.

Benchmark input sources: official public pricing pages for Google Cloud Firestore, Agora Voice Calling, and Twilio Video Group Rooms. Pricing should be revalidated before launch.