

WorldFrame

Executive Feasibility Snapshot

Assessment of offline-first quiz scope, local JSON content architecture, Apple-native competitive features, and the deliberate rejection of a Firebase-backed online game layer



Product concept

WorldFrame is a fully native iOS country-knowledge quiz / puzzle game built around local structured datasets instead of a server-dependent gameplay model. The current SwiftUI codebase manages country attributes, criteria logic, progression, localization, audio feedback, legal information, and save behavior through Apple-native frameworks and local JSON content.

The project reflects a meaningful content base — 195 countries, 81 criteria entries, and a 75-level structure — while remaining focused on offline-first execution. Because WorldFrame was the studio's first shipped app, the first release intentionally avoided custom backend, account, and live-service complexity.

Executive decision

Launch WorldFrame as an offline-first iOS quiz game built around local JSON content, on-device progression, Apple-native architecture, Game Center competition, and lightweight iCloud-backed progress persistence. Keep Firebase-backed online play out of scope.

Reason: even with manageable backend cost, online mode would still have expanded delivery risk through authentication, live sync, security rules, room handling, and operational support. For a first shipped app, gameplay execution and native polish had higher priority.

Local-first app

Local country datasets, criteria logic, level progression, audio feedback, legal content, Game Center competition, and on-device save behavior.

Assessment: right-sized for a first shipped app.

Rejected Firebase online mode

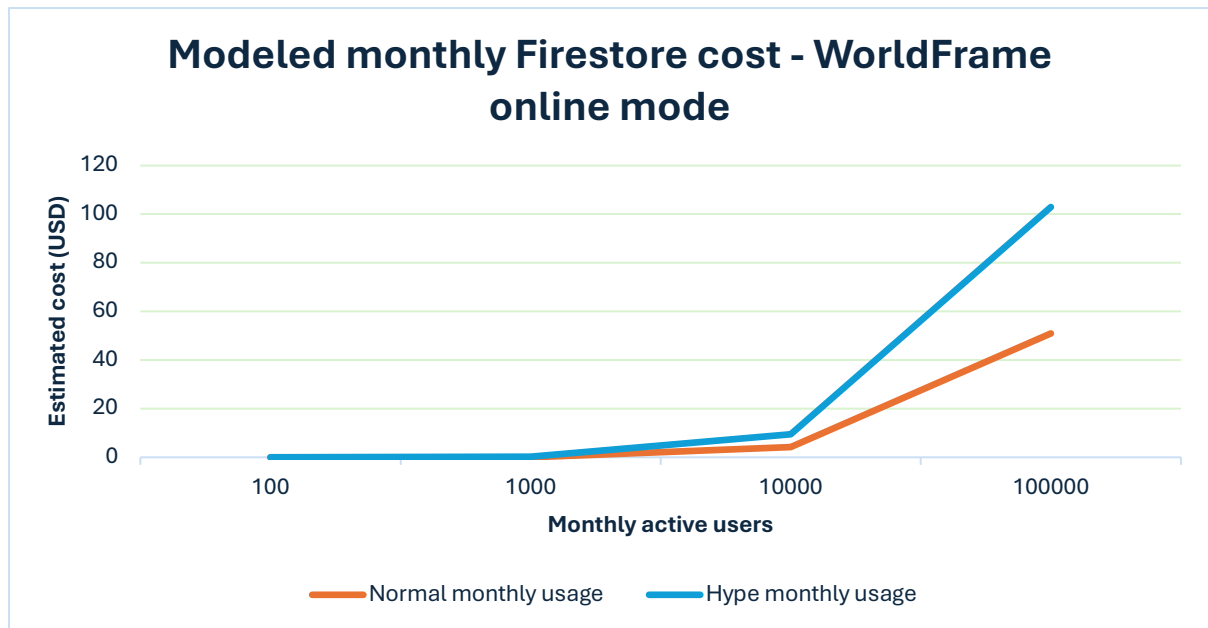
1v1 synchronized country-grid sessions, room state, answer sync, scoring, and cleanup. Assessment: technically feasible, but strategically premature for v1.

Upper stress test

\$102.95 modeled monthly Firestore cost at 100,000 monthly active users in the rejected online-mode hype case. Interpretation: cost was manageable, but backend complexity was not justified.

Cost-growth view

Niche-oriented demand model shown below; the chart uses a compact Firestore benchmark and assumes usage is distributed broadly enough across the month for free-tier quotas to offset smaller traffic levels.



Note: the plotted curve is a management illustration. It uses a simplified Firestore operating model and excludes Firebase Auth edge cases, storage, bandwidth, moderation, tax, and unusual realtime listener spikes. Actual spend would vary with room design, sync frequency, security rules, and traffic concentration.

Benchmark model and management interpretation

Illustrative benchmark assumptions

- Illustrative online concept: 1v1 synchronized country-grid sessions.
- One completed online match modeled as about 500 reads, 120 writes, and 20 deletes across room creation, answer sync, scoring, and cleanup.
- Normal case: 4 online matches per user / month; hype case: 8 online matches per user / month.
- Firestore pricing model uses current public reads / writes / deletes pricing beyond free tier and approximates free usage across a 30-day month.

Why the backend path was premature

For WorldFrame, the online question was not purely a hosting-cost question. A live mode would have introduced account flows, synchronization conflicts, match lifecycle handling, room cleanup, and production debugging into the studio's first shipped app. The chosen architecture instead focused effort on country logic, criteria design, level pacing, localization, audio feedback, polished native UI, and Apple-native competitive features. Game Center preserved lightweight competition value without forcing a custom backend from day one.

Decision framework

Monthly cost benchmark, first-app scope interpretation, and management recommendation.

Monthly cost benchmark (USD)

Normal monthly usage benchmark

Monthly active users	Firestore
100	\$0
1,000	\$0
10,000	\$4.17
100,000	\$50.95

Hype monthly usage benchmark

Monthly active users	Firestore
100	\$0
1,000	\$0.15
10,000	\$9.35
100,000	\$102.95

Annualized view at 100,000 monthly active users: about \$611.40 per year in the normal benchmark and about \$1,235.40 per year in the hype benchmark, before bandwidth and operational add-ons. Apple Developer Program membership remains \$99 per year by comparison.

Risk and management interpretation

Risk area	Management interpretation
First-app backend complexity	Avoid coupling the first release to auth, room sync, conflict resolution, and live incident handling. Result: cleaner execution path and lower delivery risk.
Product scope inflation	Keep WorldFrame centered on level design, country data, progression, and quiz polish. Result: clearer product identity and faster delivery.
Live-service operational overhead	Defer moderation, monitoring, backend analytics, and abuse tooling until broader demand justifies them. Result: lower maintenance burden.
Competitive feature balance	Use Game Center for leaderboard and achievement features instead of building full multiplayer infrastructure. Result: lightweight competitive value without backend lock-in.

Management recommendation

WorldFrame should remain an offline-first iOS quiz game built on Apple-native frameworks, structured local JSON content, Game Center, and lightweight iCloud-backed progress persistence. Firebase-backed online play should remain outside the current product scope because execution maturity, quiz polish, and scope discipline matter more than backend reach for a first shipped app.

The rejected Firebase-backed online direction was later transferred into a broader concept, Project Global Stats, which eventually became the app RivalStats. In that context, global online tracking across board games, online games, and real-life games against friends provided a much stronger reason to invest in cloud identity, shared history, and broader stat infrastructure. What this demonstrates: disciplined feature reduction based on product fit and delivery focus, while preserving a transferable backend concept for a broader future product. Benchmark input sources: official public pricing pages for Google Cloud Firestore and Firebase pricing documentation.