

# Mission Critical Data: Accurate, Available, Secure

As multi-player gaming goes mainstream, the industry is changing rapidly. Demographics are becoming global and more inclusive, rising production values are driving a wave of consolidation, and an explosion of content is making discoverability into a key competitive challenge. At worldwide scale even niche games have to be prepared for an onslaught of traffic, but building out new infrastructure for every new game is prohibitively expensive. Gamers are especially sensitive to performance, so reducing lag times by distributing data geographically is necessary to compete in the worldwide market. Promoting player engagement beyond the game in the form of experiences and social is a huge contributor in reducing churn and increasing virality of each game through sharing mechanisms and product tie-ins.

All of this innovation is accelerated by the move to shared services, containerization and patterns like serverless compute. Of all components in the stack, the database layer has been last to move to robust, distributed, commodity hardware. Studios are looking to save costs with better results by utilizing shared game backends that are already proven to scale.

Some key database related challenges include:

- Enforcing data consistency: Ensuring strong consistency of data in realtime so players never experience stale data
- Scalability on-demand: Delivering high throughput, low latency experiences to global audiences no matter what device they're on
- Availability and reliability: Making critical data available to interactive and analytic workloads 24x7 without

sacrificing reliability or performance, while running on commodity infrastructure

- Data flexibility: Evolve your data model as you learn which aspects of your game users engage with most
- Security and governance: Data must be secure to prevent breaches from becoming leaks

Thus far, games companies have been forced to give up on consistency to achieve the speed and scale needed to meet user demand – and still can't guarantee the ability to handle opening weekend traffic. Adding metrics and transactional features to an eventually consistent backend has required patching together solutions from traditional databases along with queuing middleware to integrate analytic systems. But that limits innovation.

## INTRODUCING FAUNADB

### TRANSACTIONAL MISSION CRITICAL NOSQL

FaunaDB is a NoSQL database that combines the transactional correctness of a relational database with the flexibility and scalability of NoSQL systems. It is designed to simplify development of distributed applications while making database operations dramatically easier.

The core capabilities of FaunaDB include:

- Transactionality: ACID transactions commit across all replicas, enabling gaming application servers to run nearby your players, without lag or inconsistency
- Interface flexibility: Document-relational model makes it

easy to store and query data in the format best suited to its use, and evolve as requirements change

- Row-level security: Object level access control so that your database can model your application rules in order to simplify your stack or provide defense in-depth
- Horizontal scalability: Add and remove nodes without interrupting application performance within the same site or across global data centers
- Fault tolerance / reliability: Designed for the webscale era, FaunaDB continues processing transactions even as nodes fail
- Temporality: A snapshot based storage engine retains historical data for a configurable period and permits correction of data errors in snapshots, making undo and rollback easy to implement
- Multi-tenancy: Secure and QoS-based workload isolation ensures data and connections can't interfere with one another
- Operational simplicity: Start in the cloud, or run anywhere you can run a JVM, with built-in cluster management and replication to eliminate the pain of traditional database operations

The combination of these capabilities make FaunaDB the best platform to for modern high-performance gaming data. We explore some of the specific use cases in the following section.

## FAUNA TRANSFORMS GAME DEVELOPMENT

FaunaDB is used by game developers and publishers to address some of the most critical business problems. Some of these applications include:

### IDENTITY AND PERSONALIZATION

Gamers increasingly expect their preferences and identities to flow seamlessly across games and platforms. Identity

systems are critical because they play a role in every API request your system receives, so any issues can create widespread inconsistencies or performance bottlenecks. A shared identity and access management service can become the foundation for delighting your users. FaunaDB's distributed architecture means you can keep user data geographically near the user, without sacrificing consistency. A single FaunaDB installation can support multiple games via its built-in multi-tenancy. QoS-based resource scheduling ensures that all your games can interact with a common customer identity repository without impacting each other.

### GAME CONTENT AND SAVED STATE

Game dynamics, computer player strategies, level maps and loot systems all need to operate at the speed of gaming. Incorrect and unexpected results can discourage players. Saved games and characters are very important to players, so any sense of instability will increase churn and lower your profitability. Game data has variety, as new features and special offers invariably add texture to the data model. FaunaDB's flexible documents can store any semi-structured data which can then be adapter over time. Transactional replication with global redundant failover mean your important data is safe and correct, even in the face of natural disasters. Your app content and user updates will never be stale, allowing for updates to roll out.

### IN-APP PURCHASES

Newzoo reports that there are "2.2 billion active gamers in the world, of which 47%, or 1.0 billion gamers, spend money while playing." Validating, synchronizing, and normalizing in-app payments across app stores requires transactional integrity at global scale, and can be greatly simplified with NoSQL documents. FaunaDB's powerful query language is suited to express transactional logic across complex and varying data, taking the pain out of working with multiple payment methods and systems to give your users a consistent experience across platforms. The entire payments pipeline, from application code to process funds via upstream APIs to backend accounting ledgers, benefits from FaunaDB's ACID transactions. Whether you are writing an in-game trading system or tracking monetization across a stable of games, FaunaDB's ability to execute application logic within

ACID transactions takes the fuss out of working with the variety of real world data.

## SOCIAL FEATURES

GameSparks identifies the top five social features that add value to your game as: Facebook integration, points and rewards for inviting friends, gifting virtual items, social leaderboards, and tournaments and other events. FaunaDB's built-in authorization and authentication features are easy to integrate with Facebook login and other third-party authentication providers. Object-level access control means that once authenticated, devices and browsers can safely query the database directly, simplifying your application stack. FaunaDB's transactional integrity is designed for buy, selling, and tracking points in a high-volume environment, so you can add in-game economics with the same capabilities used by the financial services industry. Social activity feeds are easily supported by the FaunaDB query language with its temporal and graph features.

## METRICS AND ANALYTICS

Advances in metrics are driving innovation in content development and business models. Customer and community metrics can help you target the right demographics and sources of new players. Performance metrics can identify technical problems before they impact player experience. Gameplay metrics can help you iteratively improve retention

and engagement, and can include in-game behaviors, interface usage heat maps and system metrics such as which simulated agents are the toughest to beat. FaunaDB is easy to integrate with best-of-breed analytics systems by using its event-driven interfaces to keep data pipelines up-to-date with changes as they happen, allowing your operational database to seamlessly feed the rest of your data ecosystem. FaunaDB's quality-of-service connection management system can prioritize interactive workloads over analytics jobs, allowing both to rely on FaunaDB as the source-of-truth.

## CONCLUSION

FaunaDB is built from ground up to address the database challenges that hamper game development and adoption. It blends the transactional benefits of big-iron SQL systems with the scalability of NoSQL systems, while introducing new innovations that dramatically simplify the lives of both developers and operations. Available as a managed cloud as well as on-premises deployment, FaunaDB is infrastructure agnostic, multi-cloud, and easy to get started with. Funded by leaders like GV (formerly Google Ventures), Point72 Ventures, Capital One Growth Ventures and others, Fauna is a foundation for your digital transformation.

Please reach us at [priority@fauna.com](mailto:priority@fauna.com) for details or get started today at [www.fauna.com](http://www.fauna.com). Expect more from your database.