

Інформаційні технології

Humeniuk Andrii

Software Engineer at Glovo, specializing in distributed and high-performance architectures for real-time, mission-critical systems

Terletska Khrystyna

Senior Software Engineer at DraftKings, specializing in high-load distributed systems and real-time data processing, ETLs, data warehousing
(Kyiv, Ukraine)

MULTI-USER ARCHITECTURE AT DRAFTKINGS: ISOLATION OF TRAFFIC, RATES AND RISKS BY REGION

Summary. *The article discusses the theoretical foundations of multi-user architecture and its practical implementation, using the DraftKings platform as an example. It analyzes three basic tenant isolation models and key patterns for organizing data storage in multi-user systems. It pays special attention to the problem of traffic isolation and financial transactions in conditions of high workload and regulatory restrictions. The article shows how the DraftKings architecture combines scalability, resilience, and compliance through the use of microservice approach, AWS infrastructure, proprietary testing, and risk management solutions. It highlights organizational and technical measures to reduce financial, regulatory, and reputational risks. Promising directions for the development of multi-user architectures are also noted.*

Key words: *multi-user architecture, tenant isolation, DraftKings, AWS, Aurora, Route 53, Global Accelerator, CloudFront, compliance, risk management, financial transactions, high-load systems, multi-cloud, Zero-Trust, training.*

Multi-user architecture is a method of organizing software and infrastructure resources in which a single system serves multiple, isolated "tenants" on a shared platform. The key premise of multi-user systems has become the cloud paradigm of the "resource pool": computing, storage and network are aggregated by the provider and dynamically redistributed among consumers within the framework of a common model, which in the literature is formulated as resource pooling with support for multi-user mode. This feature is fixed in the classical definition of cloud computing NIST SP 800-145, where the multiuser model is explicitly called a "multi-consumer" service mechanism with dynamic assignment of physical and virtual resources [3].

In the context of cloud providers and industry standards, multi-tenant architecture refers to the way and level of separation of computing resources, network connections, and data between tenants. The key aspect is isolation: the system must strictly restrict access, preventing any inter-tenant leaks and visibility, while ensuring cost-effectiveness and ease of management. The practical recommendations for SaaS emphasize that choosing an isolation level is a conscious balance between costs, regulatory requirements and the necessary flexibility.

There are three main approaches to tenant isolation in the industry, often referred to as "silo", "pool" and "bridge" in AWS materials. The Silo model provides for complete isolation: each client receives its own stack of computing resources, network, and storage, which reduces the risks of overlap, but increases costs and management complexity. The pool model is based on the sharing of infrastructure with separation at the logical level in a common cluster, which saves resources, but requires strict access control mechanisms and protection from the influence of "noisy neighbors". The Bridge model combines both options, for example, a common web layer with isolated business services and databases for each tenant. (Figure 1).

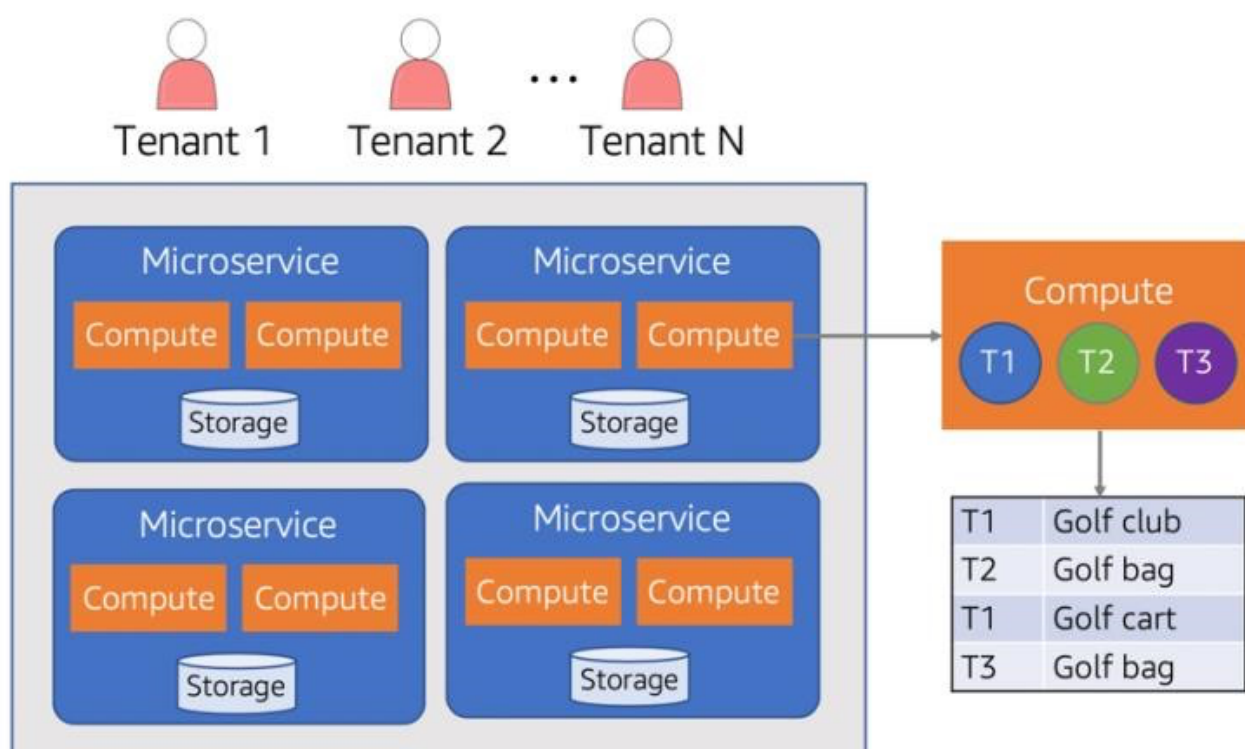


Fig. 1. The "pool" isolation model in a multiuser architecture [4]

At the data level, there are three patterns of multi-user storage: "tenant database", "shared database - separate schemas", and "shared database - common schema (low-level isolation)". The AWS guide on storage strategies for SaaS presents these options as "silo" and "pool" separation with varying degrees of isolation and operational costs.

In RDBMS (Relational Database Management System), each of these patterns correlates differently with regulations and operations. The tenant base simplifies auditing and enforcement, but it multiplies instances. Schemas provide a compromise, while string isolation requires strict access policies (RBAC/ABAC) and coding discipline.

Review articles and practices, including engineering blogs and guides on BigQuery, analyze the advantages and risks in detail and recommend selecting a pattern based on compliance, scalability, and noisy load requirements.

Table 1 compares the basic models based on key criteria.

Table 1

Basic models of multiuser isolation and their properties

Model	Isolation level	Key advantages	Key disadvantages	Typical areas
Silo (full stack per tenant)	Dedicated stack/base/VPC on tenant	Maximum security and compliance; simplicity of the mental isolation model	Highest cost; complexity of scaling and operations with multiple tenants	Strict industry and legal requirements
Pool (shared infrastructure)	Sharing, logical differentiation (IAM/ABAC/RLS)	Economies of scale; fast delivery	Mature policies and telemetry are needed; risk of "noisy neighbor" in case of configuration errors	Massive SaaS products, analytics
Bridge (hybrid)	Shared top layers + "silo" on critical services/databases	Balance isolation and efficiency; segment flexibility	Architectural complexity; option management	Diverse customer/region requirements

After considering the general principles and models of multi-user architecture, it would be beneficial to proceed to the study of their practical implementation through the example of one of the largest platforms in the online gaming and sports betting industry. DraftKings demonstrates how the concepts of isolation and scalability can be implemented in conditions of extreme loads and diverse regional constraints.

In the context of DraftKings, several elements of hybrid isolation have been confirmed. First, the company has undergone a major modernization, with hundreds of applications being transferred from the .NET Framework to .NET Core and deployed as microservices on AWS. At the same time, DraftKings emphasizes a highly regulated environment and strict testing and release procedures, typical of "common" platform services [2].

Secondly, the financial registry is implemented on Amazon Aurora, a MySQL-compatible database, and is segmented into 200 separate databases. The

cluster can process up to 1 million transactions per minute, with typical write delays of around 6 milliseconds and replication times of 10-30 milliseconds, reflecting the company's "silo" approach to critical data. Thirdly, access to products is strictly limited by jurisdiction. DraftKings is licensed by individual states, and user access is ensured through mandatory geolocation verification, including the use of GeoComply, which reinforces regional isolation in terms of access policies and user traffic.

Features of DraftKings as a highly scalable platform:

1) Scalability and load capacity.

DraftKings serves 2 million average monthly active users, with an annual active user base of 14.8 million. During peak periods (such as Super Bowl and March Madness), the platform processes more than 50,000 bets per minute. The platform is designed to handle increased load during major sporting events, such as the Super Bowl and March Madness.

2) Modernization and migration.

DraftKings successfully migrated hundreds of .NET Framework-based applications to .NET Core and Amazon Web Services (AWS) to reduce costs, improve scalability, and enhance developer efficiency. The transition to a microservices architecture was accompanied by the establishment of a DevOps culture, which included the use of tools like New Relic and AWS for monitoring and ensuring reliability under high loads [2].

3) Testing and Sustainability: The CleanRoom.

DraftKings has developed a testing environment called the CleanRoom to ensure the stability of its services when new features are released. The CleanRoom is an isolated testing environment built on Kubernetes.

The CleanRoom uses several key features to ensure stability:

- A new cluster and separate namespace are created for each test environment.

- Configuration is managed using Infrastructure as Code (IaC), which is version-controlled.
- After a certain time period, called the time-to-live (TTL), the test environment is automatically deleted.

These features allow DraftKings to test changes in as close-to-production conditions as possible without risking the operation of the platform. Table 2 shows some of the main isolation mechanisms used for distributed architectures and cloud infrastructure.

Table 2

Comparative approaches to traffic isolation in multiuser systems

Method	Description	Advantages	Usage examples
Geolocation DNS (Route 53)	Routing based on the geography of the request	Clear borders, regional isolation	Dividing the API by region
Geospatial bias	Configuring coverage areas with bias	Flexibility in traffic management	Traffic on the "borders" of geofences
CDN (CloudFront)	Caching and delivery via global pops	Latency, load balancing	Static and dynamic content
Global Accelerator	Optimized route within the AWS network	Low latency, fault tolerance	Bids, real-time transactions
Geo-restrictions (CloudFront)	Filtering access by country	Compliance, geo-blocking	Regional access restrictions
DNS Failover (Route 53)	Health check + reservation	Fault tolerance, fault tolerance	Redirecting traffic in case of service failures

The inclusion of these mechanisms in the architecture of a multiuser system allows you to elegantly and reliably isolate traffic across regions, ensuring a balance between performance and compliance with regional requirements.

An AWS reference diagram (Figure 2) is used to illustrate a typical outline of streaming event processing and client actions in betting/gaming scenarios. It demonstrates how low-latency event pipelines (API Gateway/IoT Core →

Kinesis), computing and personalization (Lambda/SageMaker), as well as separate repositories for online access (DynamoDB) and transactional data (Aurora) are organized with end-to-end logging (CloudWatch).

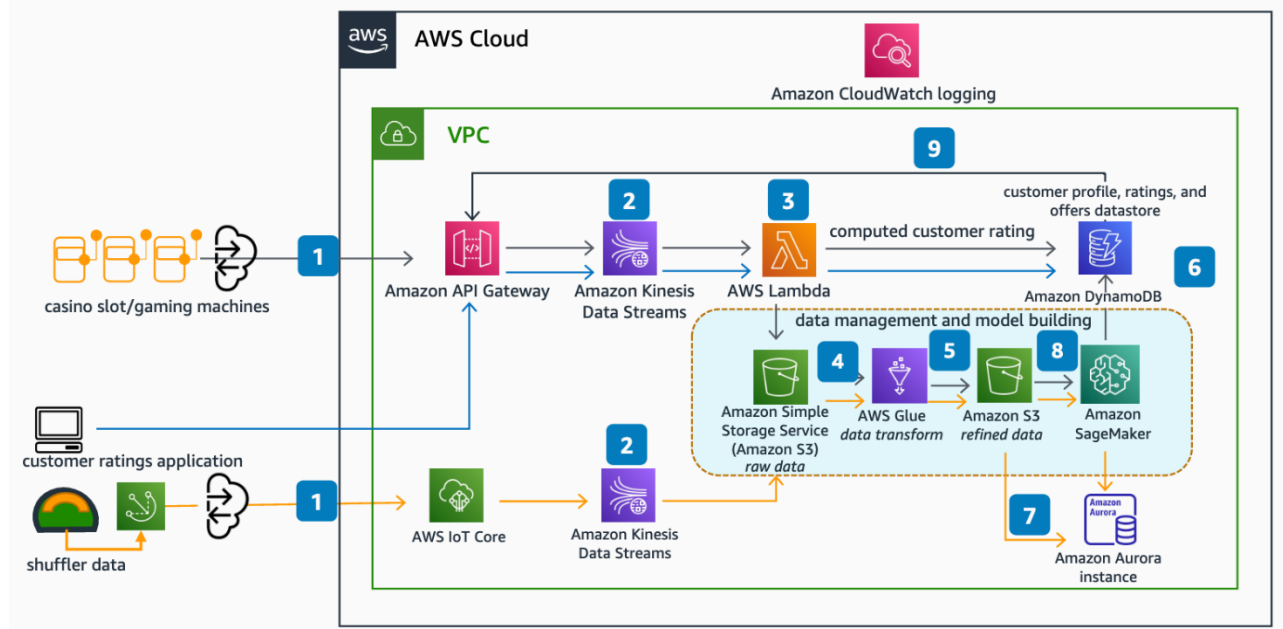


Fig. 2. AWS real-time reference architecture in the Gaming/Casino industry [5]

In the context of platforms like DraftKings, ensuring the isolation of user bets and financial transactions is a crucial aspect that affects security, regulatory compliance, and system sustainability. DraftKings has operated as a vertically integrated platform since April 2020, after the closure of its business combination with SBTech, giving the company full control over key components such as the sportsbook and wallet/payments. The AWS industry guidelines for betting architectures provide for a separate wallet and payment module, but multi-currency primarily applies to scenarios in different countries or jurisdictions. In the United States, DraftKings operates exclusively in US dollars.

The target architecture for DraftKings is based on the following principles:

- A reliable transactional registry where all user operations are recorded. This ensures transparency and accountability for all transactions.

- Horizontal scaling of reading operations through the use of Aurora Replicas, which helps offload the load on the database during times of high traffic.
- Strict segmentation of users by jurisdiction and geolocation verification to comply with state regulations. This is necessary to ensure compliance with local laws and regulations.
- The use of a single settlement currency, USD, in the United States. DraftKings' instructions for deposits and withdrawals, as well as any limits, are given in USD. Checks and non-cash limits are also indicated in this currency. There is no provision for account currency conversion in the user documentation, so a multicurrency converter is not necessary for operations in the US [1].
- Regulatory differences between states are reflected in the rules for the movement of funds: In certain jurisdictions (e.g., MD, MA, TN), deposits can only be used within the product they were made in (no inter-product transfers) [1].

Effective risk management and compliance with regulatory requirements are crucial components of the sustainability of a sports betting platform. These include several interconnected aspects:

1. Types of risks:

- Financial risks, such as trading, pricing, and payment responsibilities, are critical in a highly volatile market.
- Regulatory and compliance risks, including licensing, anti-money laundering (AML) measures, customer identification (KYC), and player protection.
- Reputational risks, such as negative user experiences, public backlash, and fines from regulatory bodies.

2. Corporate structures and tools:

- DraftKings has established the position of Chief Responsible Gaming Officer to promote a culture of responsible gambling, manage player limits, and communicate with regulators.
- The company utilizes the Contract Lifecycle Management (CLM) platform to automatically manage risks and ensure compliance with regulatory agreements.

3. Technical and Operational Practices

- Governance, Risk, and Compliance (GRC) frameworks help to centralize risk management, compliance controls, and internal oversight.
- Product-level risk controls: betting limits, self-exclusion systems, and in-app messaging as elements of responsible gaming.
- Compliance with regulatory requirements in the infrastructure, such as the AWS Reference Architecture, which shows how additional jurisdictions require the localization of wallets and accounting systems and the delineation of data and infrastructure within a regulated area.

In the future, multi-user betting systems will likely move towards multi-cloud architectures to reduce their dependence on a single provider and improve fault tolerance. One important trend will be the adoption of zero-trust security models, which ensure continuous access verification and minimize the risk of data leaks.

DraftKings continues to invest in key areas of its platform development, with a priority on reducing latency when placing bets. This allows transactions to be processed in under a second, improving the user experience. The company pays close attention to personalizing services using machine-learning models, providing recommendations that are more accurate and allowing for flexible interface customization based on player preferences. At the same time, DraftKings is expanding its presence into new US states to increase its audience reach and strengthen its market position. The company has also strengthened its integration with media content, including the acquisition of sports analytics

network VSiN in 2021. This allows the betting platform to be combined with its own media resource, attracting and retaining users.

Thus, multi-user architecture has become the fundamental basis for creating modern high-load platforms. Scalability, fault tolerance, and compliance with strict regulatory requirements are essential for these platforms. Using DraftKings as an example, we can see that the effective implementation of this architecture requires an integrated approach. This includes choosing the optimal isolation model, utilizing cloud technologies for traffic distribution and transaction processing, as well as integrating risk management and compliance mechanisms across all levels of the system.

Future developments involve enhanced security through Zero-Trust models and expanded multi-cloud strategies. These approaches allow platforms to not only comply with legal requirements but also ensure a high level of trust and service quality for millions of users worldwide.

References

1. DraftKings – Depositing on DraftKings – Overview (US). URL: https://support.draftkings.com/dk/en-us/depositing-on-draftkings-overview-us?id=kb_article_view&sysparm_article=KB0010414&.
2. Modernizing legacy .NET applications: DraftKings' principles for success | Microsoft Workloads on AWS. URL: <https://aws.amazon.com/ru/blogs/modernizing-with-aws/modernizing-legacy-net-applications-draftkings-principles-for-success/>.
3. NIST SP 800-145, The NIST Definition of Cloud Computing. URL: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>.
4. Pool isolation – SaaS Tenant Isolation Strategies: Isolating Resources in a Multi-Tenant Environment. URL: <https://docs.aws.amazon.com/whitepapers/latest/saas-tenant-isolation-strategies/pool-isolation.html>.

5. Real-Time Casino Player Analytics – Real-Time Casino Player Analytics. URL: https://docs.aws.amazon.com/zh_cn/architecture-diagrams/latest/real-time-player-casino-analytics/real-time-player-casino-analytics.html?utm_medium=organic&.