Documentation for Gaming Multiplayer Website

Prepared by : *Jignesh More* Last updated : *May 20th, 2024*

----

1. Objective

To create clear and accessible documentation that helps internal and external stakeholders to understand and effectively utilize the data lake ecosystem, reporting tools, and key performance indicators of the gaming multiplayer esport website. Help them to Identify improvement areas and troubleshoot challenges and optimize the ecosystem.

2. Scope

The scope of this documentation will take you thorough exploration of the Technology used, gaming architecture, data lake ecosystem, json, data schema, reporting layer, and key performance indicators (KPIs) relevant to the gaming multiplayer esport website. It will take you into the architecture of the database infrastructure, providing an overview of each database's structure and purpose, including user profiles, authentication data, preferences, game statistics, and more. Additionally, it will help you to better understand the reporting layer built atop the data lake, detailing the tools and technologies employed for analytics and business intelligence, as well as the types of reports and dashboards available to stakeholders. Furthermore, the documentation will define and classify key performance indicators into categories such as user engagement, game performance, community interaction, and revenue generation, outlining how each KPI is measured, reported, and evaluated. Throughout, clarity and accessibility will be prioritized to ensure comprehension by both technical and non-technical audiences, facilitating effective utilization of the data ecosystem to assess and enhance the website's success and performance.

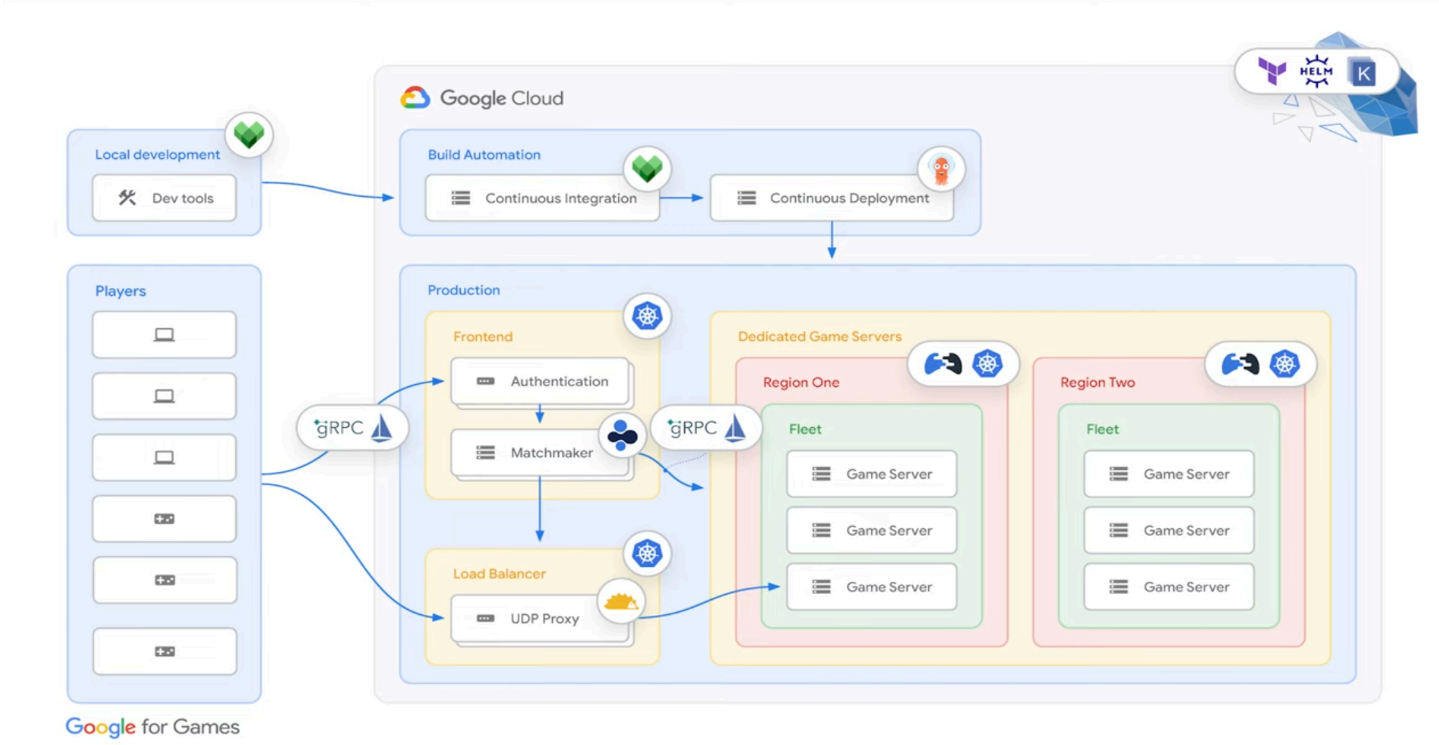
3. Technology used

- Bazel :
  - Purpose: Build system for local development and continuous integration.
  - Why it's used: Supports multiple languages within a monorepo, ensuring consistent builds across platforms.
  - Usage: Used for building and testing back-end code written in languages like C++, Go, Rust, etc.
- Argo CD:
  - Purpose: Continuous deployment tool for Kubernetes.
  - Why it's used: Automates the deployment process, ensuring that changes are rolled out automatically to production.
  - Usage: Monitors the main branch for commits and updates deployment accordingly.

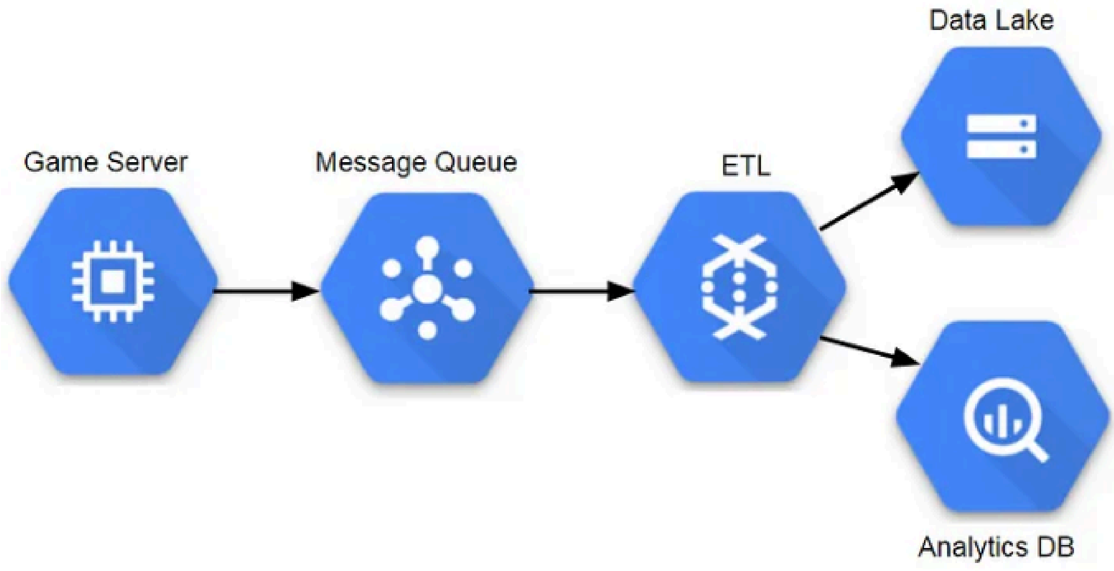
- Kubernetes (GKE):
  - Purpose: Container orchestration platform.
  - Why it's used: Provides scalability, ease of management, and good support from Google Cloud.
  - Usage: Runs 18 clusters, manages memory and CPU resources, and facilitates deployment of various services.
- gRPC:
  - Purpose: High-performance RPC framework.
  - Why it's used: Provides strong typing and compile-time safety for communication between game clients and servers.
  - Usage: Used for efficient communication between game clients and back-end services.
- Istio:
  - Purpose: Service mesh for managing microservices.
  - Why it's used: Enables smart routing and authentication within the cluster.
  - Usage: Handles routing and HTTP requests within production clusters, including developer-specific environments.
- Open Match:
  - Purpose: Matchmaking framework for scalable game matchmaking.
  - Why it's used: Simplifies matchmaking by handling complex state tracking and player grouping.
  - Usage: Utilized for ordering players and managing matchmaking rules efficiently.
- Agones:
  - Purpose: Kubernetes-based game server scaling and orchestration.
  - Why it's used: Teaches Kubernetes about the unique lifecycle requirements of game servers, ensuring uninterrupted gameplay.
  - Usage: Manages game servers, ensuring reliability and scalability.
- Quilkin:
  - Purpose: UDP game server proxy.
  - Why it's used: Controls routing between game clients and servers, while hiding server IPs to mitigate DDoS attacks.
  - Usage: Ensures secure and efficient communication between game clients and servers.
- Go and Rust:
  - Purpose: Programming languages for back-end development.
  - Why they're used: Go offers strong typing, speed, and ease of learning, while Rust provides speed and control over memory management.
  - Usage: Used for writing back-end services, with Rust employed for tasks where Go is not suitable.

- Terraform, Helm, and Customize:
    - Purpose: Infrastructure as code tools for managing cloud resources and Kubernetes configurations.
    - Why they're used: Enable automation and version control of infrastructure setup and deployment.
    - Usage: Terraform for setting up infrastructure, Helm for package management, and Customize for customizing package configurations.

4. Gaming architecture



5. Data Lake Ecosystem (Serverless)



Components in a managed Analytics Architecture (GCP)

Serverless means that instead of managing servers ourselves, we rely on managed services provided by platforms like Google Cloud Platform (GCP). These services handle tasks like data processing, scaling up systems when needed, and keeping things running smoothly in near real-time.

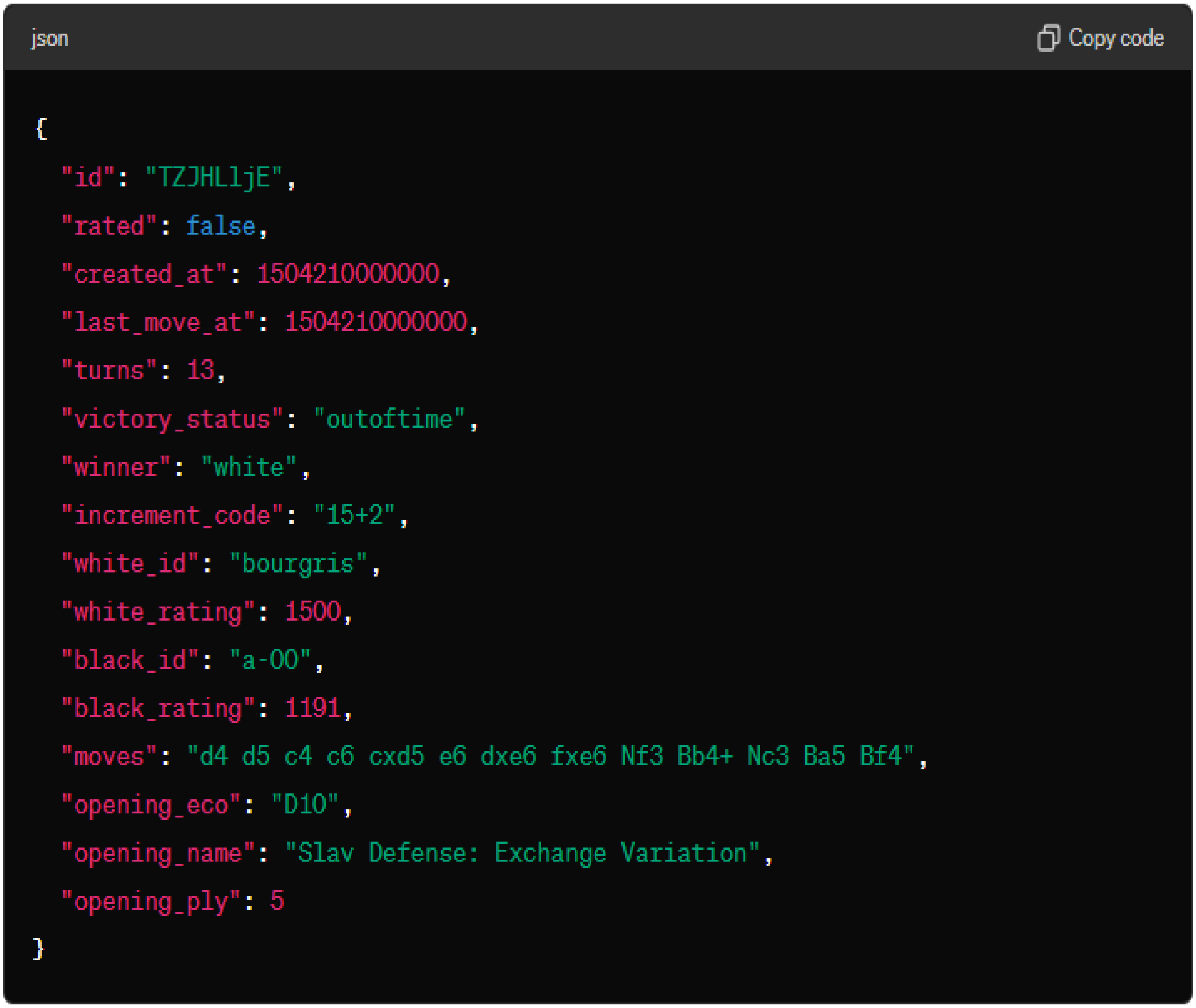
Here's how it works :

1. Game Server: This is where all the action happens - players are making moves, and games are being played. The game server sends data in a structured format called JSON to the next stop in the pipeline.
2. Message Queue: Think of this like a digital post office. It receives the data from the game server in JSON format and queues it up, ready to be processed.
3. ETL (Extract, Transform, Load): This is where the magic happens. The data from the message queue is processed, cleaned, and transformed into a format that's easy to work with. It's like taking raw ingredients and turning them into a delicious meal.
4. Data Lake: Imagine a vast reservoir where all the processed data is stored. This is the data lake, a centralized repository where we keep all our game analytics data safe and sound.

5. Analytics Database: This is where the data gets sliced and diced to extract insights. Analysts and developers can run queries and perform analysis on this database to understand player behavior, game performance, and more.

6. Json (Raw - String sample batch-created)

The server generates JSON strings that encapsulate raw data for batch processing. Here's an example of a sample JSON string:



```json
{
  "id": "TZJHLljE",
  "rated": false,
  "created_at": 1504210000000,
  "last_move_at": 1504210000000,
  "turns": 13,
  "victory_status": "outoftime",
  "winner": "white",
  "increment_code": "15+2",
  "white_id": "bourgris",
  "white_rating": 1500,
  "black_id": "a-00",
  "black_rating": 1191,
  "moves": "d4 d5 c4 c6 cxd5 e6 dxe6 fxe6 Nf3 Bb4+ Nc3 Ba5 Bf4",
  "opening_eco": "D10",
  "opening_name": "Slav Defense: Exchange Variation",
  "opening_ply": 5
}
```

7 a. Data Schema (with Data File - BQ)

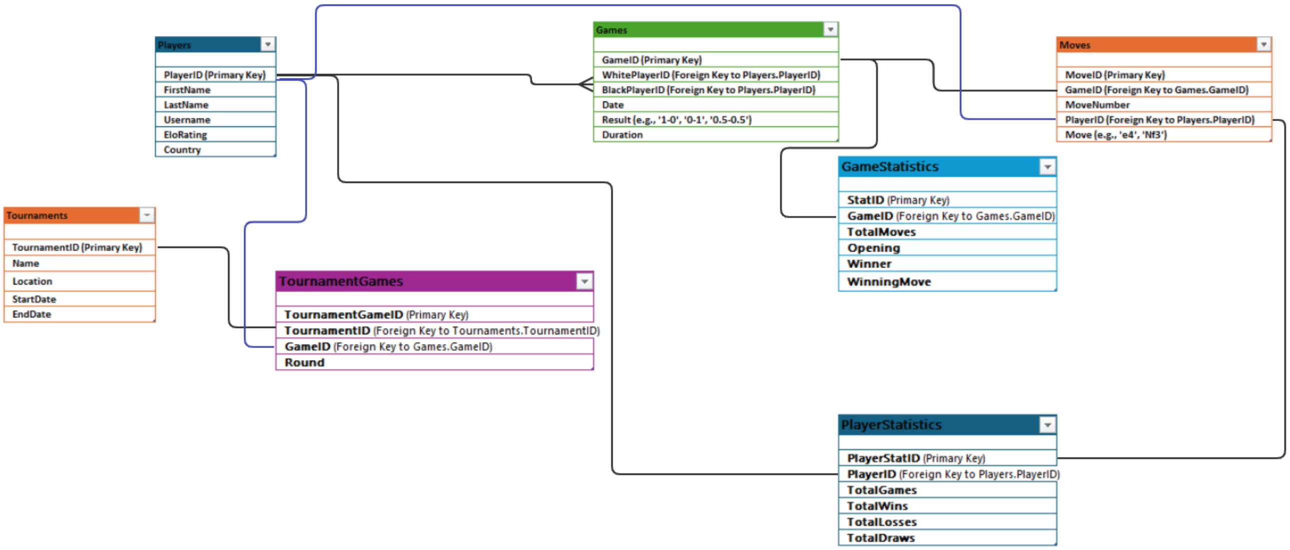
Game Data

These attributes provide a comprehensive view of each chess game, to better understand various aspects of gameplay, strategy, and player performance.

A. Game ID: Unique identifier for each game.
B. Rated: Indicates whether the game is rated (True) or unrated (False).
C. Start Time: Timestamp indicating when the game started.
D. End Time: Timestamp indicating when the game ended.
E. Number of Turns: Total number of turns played in the game.
F. Game Status: Indicates the status of the game (e.g., outoftime, resignation, mate, etc.).
G. Winner: Indicates the winner of the game (either white or black).
H. Time Increment: Time control increment code used in the game.
I. White Player ID: Identifier for the white player.
J. White Player Rating: Rating of the white player.
K. Black Player ID: Identifier for the black player.
L. Black Player Rating: Rating of the black player.
M. All Moves in Standard Chess Notation: Sequence of moves played in the game using standard chess notation.
N. Opening Eco: Standardized code for the opening played, known as the Eco code.
O. Opening Name: Name of the opening played.
P. Opening Ply: Number of moves in the opening phase of the game.

7 b. Data schema (Plex datasite)

This Plex data site schema is designed for the different platforms we use to host games on a third-party website, allowing us to better understand the data collected from the service website.

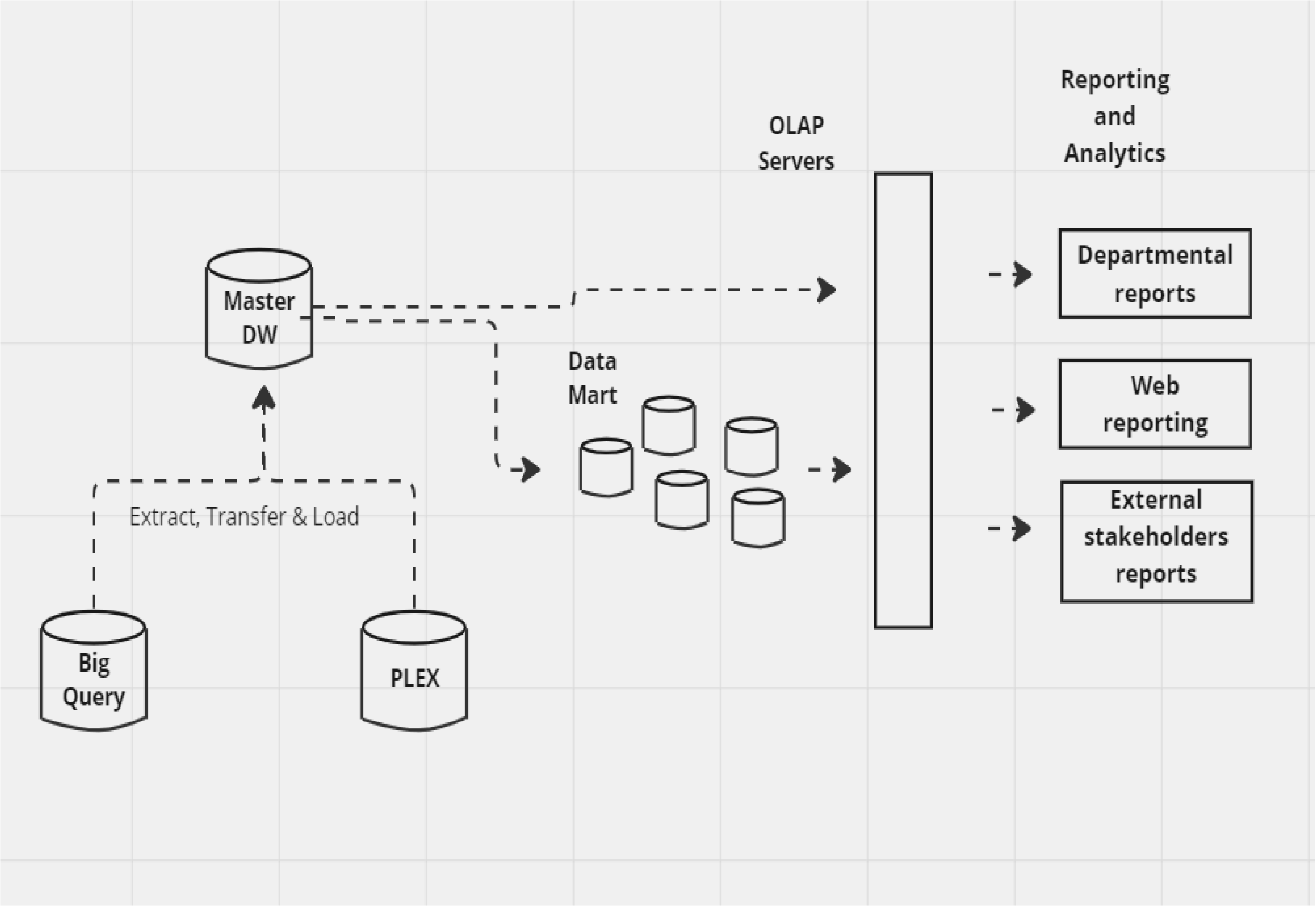

Players
PlayerID (Primary Key)
FirstName
LastName
Username
EloRating
Country
Games
GameID (Primary Key)
WhitePlayerID (Foreign Key to Players.PlayerID)
BlackPlayerID (Foreign Key to Players.PlayerID)
Date
Result (e.g., '1-0', '0-1', '0.5-0.5')
Duration
Moves
MoveID (Primary Key)
GameID (Foreign Key to Games.GameID)
MoveNumber
PlayerID (Foreign Key to Players.PlayerID)
Move (e.g., 'e4', 'Nf3')
GameStatistics
StatID (Primary Key)
GameID (Foreign Key to Games.GameID)
TotalMoves
Opening
Winner
WinningMove
Tournaments
TournamentID (Primary Key)
Name
Location
StartDate
EndDate
TournamentGames
TournamentGameID (Primary Key)
TournamentID (Foreign Key to Tournaments.TournamentID)
GameID (Foreign Key to Games.GameID)
Round
PlayerStatistics
PlayerStatID (Primary Key)
PlayerID (Foreign Key to Players.PlayerID)
TotalGames
TotalWins
TotalLosses
TotalDraws

8. Comprehensive Reporting Layer Documentation

1. Data Storage Architecture

Our gaming data is stored across two primary servers: BigQuery and PLEX.

- BigQuery:
  - Role: Serves as the primary storage for gaming data.
  - Characteristics: Managed data warehouse service provided by Google Cloud, optimized for analytical queries.
- PLEX:
  - Role: Functions as an additional storage solution for gaming data.
  - Characteristics: Custom storage platform tailored to our specific gaming data needs.



2. Data Processing Workflow

To transform raw data into actionable insights, we utilize an ETL (Extract, Transform, Load) process:

- Extract: Data is extracted from BigQuery and PLEX.
- Transform: Data undergoes cleaning, normalization, and aggregation to ensure consistency and relevance.
- Load: The transformed data is loaded into the master data warehouse.

3. Master Data Warehouse

- Role: Acts as the central repository for all processed gaming data.
- Platform: Implemented on a high-performance database system capable of handling large volumes of data.
- Features:

    - High availability
    - Scalability
    - Robust security measures

4. Data Distribution Pipelines

From the master data warehouse, data is distributed through several pipelines:

- Departmental Data Marts:
    - Customized data subsets tailored to the specific analytical needs of different departments.
    - Example Departments: Marketing, Sales, Operations, Customer Support.
    - Characteristics: Each data mart is optimized for the respective department's reporting and analysis requirements.
- OLAP Server:
    - Primary Data Feed: Directly from the master data warehouse.
    - Backup Data Feed: Additional feed from the departmental data marts to ensure data redundancy and reliability.
    - Role: Facilitates complex analytical queries and multi-dimensional data analysis.
    - Characteristics: High-performance analytical engine designed for fast query processing.

5. Reporting and Analytics

The OLAP server supports various reporting and analytics functions, including:

- Departmental Reporting:
    - Provides tailored reports and dashboards for internal departmental use.
    - Examples: Sales performance reports, marketing campaign analytics, operational efficiency metrics.
- Web Reporting:
    - Powers web-based dashboards and interactive reports.
    - Characteristics: User-friendly interfaces, real-time data updates.
- External Stakeholder Reporting:
    - Ensures relevant data is accessible to external stakeholders such as partners, investors, and regulatory bodies.

- Examples: Compliance reports, investor presentations, partnership performance metrics.

Detailed Workflow Description

- Data Extraction and Loading:
  - Data is periodically extracted from the BigQuery and PLEX servers.
  - ETL processes are scheduled to run at regular intervals to ensure data is up-to-date.
- Transformation Process:
  - Data cleaning: Removing duplicates, correcting errors, and handling missing values.
  - Normalization: Ensuring data is in a consistent format and structure.
  - Aggregation: Summarizing data to generate higher-level insights.
- Loading into Master Data Warehouse:
  - Transformed data is loaded into the master data warehouse.
  - Data integrity checks are performed to ensure accuracy and completeness.
- Data Distribution:
  - Custom pipelines distribute data from the master data warehouse to departmental data marts and the OLAP server.
  - Each pipeline is monitored for performance and data consistency.
- Reporting and Analytics:
  - The OLAP server processes data for various reporting needs.
  - Data is made available to internal and external users through secure access controls.
  - Regular updates and maintenance ensure the reporting layer remains efficient and reliable.

Security and Compliance

- Data Security:
  - Encryption at rest and in transit.
  - Access controls and authentication mechanisms.
  - Regular security audits and vulnerability assessments.
- Compliance:
  - Adherence to data protection regulations (e.g., GDPR, CCPA).

9. Key Performance Indicators (KPIs) Documentation

This section defines the key performance indicators (KPIs) relevant to our gaming multiplayer website. KPIs are classified into categories such as user engagement, game performance, community interaction, and revenue generation. Each KPI includes measurement details, data sources, calculation methods, and reporting frequency. Thresholds and benchmarks are also highlighted to evaluate KPIs and trigger necessary actions or interventions.

Categories of KPIs

1. User Engagement
2. Game Performance
3. Community Interaction
4. Revenue Generation

User Engagement KPIs

1. Daily Active Users (DAU)

Definition: The number of unique users who engage with the game on a daily basis.
Measurement:

- ❖ Data Sources: Server logs, user sessions.
- ❖ Calculation Method: Count of unique user IDs logged within 24 hours.
- ❖ Thresholds/Benchmarks: Target is a 1% week-over-week increase.

2. Average Session Duration

Definition: The average length of time users spend in a single gaming session.
Measurement:

- ❖ Data Sources: User session logs.
- ❖ Calculation Method: Total session time divided by the number of sessions.
- ❖ Thresholds/Benchmarks: Benchmark of 13 minutes per session.

Game Performance KPIs

3. Match Completion Rate

Definition: The percentage of matches that are completed without players disconnecting or abandoning.
Measurement:

- Data Sources: Game server logs.
- Calculation Method: (Number of completed matches / Total matches started) * 100.
- Thresholds/Benchmarks: Target is above 90%.

4. Lag/Latency Reports

Definition: The number of user-reported lag or latency issues per 1,000 matches.

Measurement:

- Data Sources: User feedback forms, in-game reporting tools.
- Calculation Method: Count of lag/latency reports per 1,000 matches.
- Thresholds/Benchmarks: Maximum of 10 reports per 1,000 matches.

Community Interaction KPIs

5. Forum Activity

Definition: The level of activity on game-related forums and discussion boards.

Measurement:

- Data Sources: Forum software logs, social media interaction counts.
- Calculation Method: Number of posts, comments, and likes.
- Thresholds/Benchmarks: Minimum of 5000 posts and 10,000 comments weekly.

6. User-Generated ads

Definition: The amount of ads (videos, streams, guides) created by users.

Measurement:

- Data Sources: Ads sharing platforms (Amazon, Petstore).
- Calculation Method: Count of user-generated content items clicked.
- Thresholds/Benchmarks: Target of 200 items per month.

Revenue Generation KPIs

7. Monthly Recurring Revenue (MRR)

Definition: The total revenue generated from subscriptions and recurring payments.

Measurement:

- Data Sources: Billing systems, financial records.
- Calculation Method: Sum of all subscription and recurring payments.
- Thresholds/Benchmarks: Monthly target of $500,000.

8. In-Game Purchases

Definition: The total value of all in-game purchases made by users.

Measurement:

- Data Sources: Transaction logs, financial systems.

- Calculation Method: Sum of all in-game purchase values.
- Thresholds/Benchmarks: Monthly target of $100,000.

9. Average Revenue Per User (ARPU)

Definition: The average revenue generated per active user.
Measurement:

- Data Sources: Financial records, user activity logs.
- Calculation Method: Total revenue divided by the number of active users.
- Thresholds/Benchmarks: Benchmark of $10 per user per month.

10. Customer Lifetime Value (CLTV)

Definition: The projected revenue a user will generate during their lifetime with the game.
Measurement:

- Data Sources: Historical financial data, user retention rates.
- Calculation Method: Average revenue per user multiplied by the average user lifespan.
- Thresholds/Benchmarks: Target CLTV of $150.

10. Citation

1. Gheorghiu, R. (2023, December 5). *Top 6 tools for documenting your database*. Vertabelo Data Modeler. https://vertabelo.com/blog/top-database-documentation-tool/
2. Be A Better Dev. (2021, May 28). *Real Life AWS Architecture Examples - Gaming app* [Video]. YouTube. https://www.youtube.com/watch?v=q-0yzA3WgXc
3. Howcroft, J., & Mandel, M. (2023, April 4). Making online, containerized games with managed services. *Google Cloud Blog*. https://cloud.google.com/blog/products/containers-kubernetes/making-online-containerized-games-with-managed-services
4. Giri, B. (2023, July 28). 20 Google Cloud Reference Architecture to Start your GCP Architect journey. *Medium*. https://medium.com/google-cloud/20-google-cloud-reference-architecture-to-start-your-gcp-architect-journey-2a85e8728507
5. *Overview of cloud game infrastructure*. (2022, January 28). Google Cloud. https://cloud.google.com/architecture/cloud-game-infrastructure
6. Google for Developers. (2013, May 22). *GDC 2013 - Create amazingly scalable games on Google Cloud Platform* [Video]. YouTube. https://www.youtube.com/watch?v=gc-kBK_BcJg
7. Mulders, M. (2023, December 15). *When to use Rust and when to use Go - LogRocket Blog*. LogRocket Blog. https://blog.logrocket.com/when-to-use-rust-when-to-use-golang/
8. *Introduction - Quilkin book*. (n.d.). https://googleforgames.github.io/quilkin/main/book/introduction.html
9. Google Cloud Events. (2022, March 15). *How Embark Studios utilizes open source backend tech for games* [Video]. YouTube. https://www.youtube.com/watch?v=YjeusLeC5RI
10. *Chess Game Dataset (Lichess)*. (2017, September 4). Kaggle. https://www.kaggle.com/datasets/datasnaek/chess
11. Frid, I. (2023, August 28). *17 KPIs Gaming app marketers should measure*. AppsFlyer. https://www.appsflyer.com/blog/measurement-analytics/gaming-app-marketing-metrics/
12. Zenn, J. (2024, February 25). *50+ KPIs to measure your mobile game or app*. GameAnalytics. https://gameanalytics.com/blog/50-kpi-measure-mobile-game-app/