Software Requirement System (SRS) for Daylight ES365 Cloud Gaming Platform (Gamer Software)

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the Daylight ES365 Cloud Gaming Platform. It aims to specify the functional and non-functional requirements, system architecture, and key design considerations to ensure successful implementation and deployment.

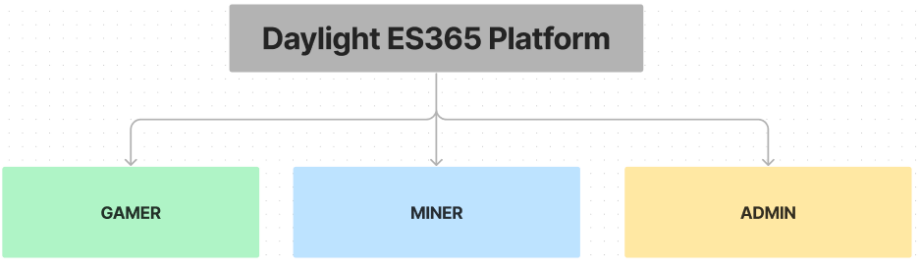
1.2 Scope

Daylight ES365 is a cloud gaming platform that connects gamers with miners. The platform provides the following capabilities for:

- **Gamers:** Gamers can download and install the Daylight ES365 software on their devices. This allows them to play high-end, AAA game titles on their own systems without needing to own high-spec hardware. They can enjoy low-latency game streaming with great picture quality and seamless gameplay.

- **Miners:** Miners can offer their high-quality computer equipment and game titles to gamers. This allows miners to earn extra income by renting out their unused computing resources when they are not in use.

- **Admin:** The admin panel is created for managing users, games, transactions and other platform operations.



2. Overall Description

2.1 System Perspective

The system integrates various components to deliver a seamless gaming experience to users and a profitable avenue for miners. This platform aims to bridge the gap between gamers who lack high-end hardware and miners who possess such resources but have them idle during certain periods.

2.2 System Features

The main functions of the Daylight ES365 platform include:

- **User Registration and Authentication**: Users can register on the platform using their email addresses. Secure authentication mechanisms ensure that only authorized users can access their respective dashboards.
- **Dashboard for Gamers and Miners**:
    - **Gamers**: A personalized dashboard showing account balance, rental history, upcoming bookings, and available game titles, providing alerts before a game is scheduled to be played, authorization using secret code.
    - **Miners**: A dashboard displaying earnings, equipment status, rental history, and availability management.
    - **Match-Making Service**: Connects miners with gamers based on smart contracts.
    - **Cloud Gaming**: Allows gamers to play AAA titles on various devices.
- **Game Rental Management**:
    - **For Miners**: Ability to list their equipment and game titles with details such as location (the precise location of the gamer), price per hour (the cost that a gamer must pay to use a miner's computing resources for one hour.), and availability given. Miners can update or remove listings as needed.
    - **For Gamers**: Search functionality to find and book available games based on location (miner's location), time frame (specific time in which the gamers want to rent the equipment), game genre, and game name.
- **Game Booking and Playing**:
    - **Booking**: Gamers can book a time slot for a game and pay for it using DL365 Coin. The payment is held in an escrow account until the session is completed.
    - **Playing**: Gamers use the Daylight ES365 application to connect to the miner's equipment. Upon successful authorization, they can play the game with streaming video and control signals managed by the platform.
- **Cryptocurrency Transactions**:
    - Payments for game rentals are handled using the platform's cryptocurrency, DL365 Coin.
    - A secure escrow mechanism ensures that funds are held until the gaming session is successfully completed.
    - Miners receive payments after deducting the platform's service fee.
- **Mining Progress Tracking**:
    - The system tracks the usage of miner's equipment and calculates earnings based on the current cryptocurrency value.
    - Detailed reports are provided to miners showing their earnings and equipment usage statistics.
    - No Need for Expensive Hardware
- **Gamer Benefits**:

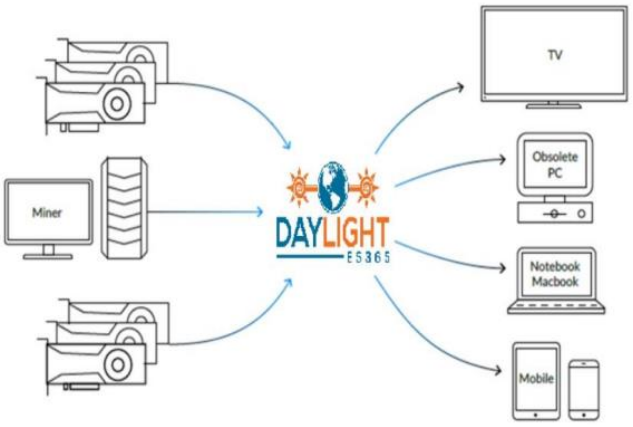
- Wide Device Compatibility
- No Need for Expensive Hardware
- Cost-Effective Gaming
- Seamless Gaming Experience
- Security and Transparency
- Flexible Gaming Options
- Continuous Improvements

2.3 User Classes and Characteristics

- **Gamers**: Individuals who want to play high-end games without investing in expensive gaming hardware. They require a stable internet connection and a compatible device to access the platform.
- **Miners**: Individuals or organizations owning high-performance computer equipment and AAA game titles. They aim to monetize their idle computing resources by renting them out to gamers.
- **Administrators**: Platform administrators who manage user accounts, monitor system performance, handle disputes, and ensure smooth operation of the platform.

2.4 Operating Environment

- **Client-Side**:
  - Compatible with Windows, macOS, and possibly other operating systems (e.g., Linux).
  - Requires installation of the Daylight ES365 application for game streaming.
  - Internet connection with sufficient bandwidth to support low-latency game streaming.



- **Server-Side**:
  - Hosted on cloud infrastructure to ensure scalability and reliability.
  - Integration with blockchain for handling cryptocurrency transactions.
  - Utilizes decentralized storage solutions like Story or Sia for future enhancements.

2.5 Design and Implementation Constraints

- **Scalability**: The system must handle a large number of concurrent users and high volumes of data transfer.
- **Security**: Strong encryption and secure authentication are essential to protect user data and transactions.
- **Latency**: Low latency is crucial for a satisfactory gaming experience.
- **Blockchain Integration**: Implementing smart contracts and secure cryptocurrency transactions.
- **Regulatory Compliance**: Adherence to legal requirements in different regions regarding data privacy and financial transactions.

2.6 Assumptions and Dependencies

- **Internet Connectivity**: It is assumed that users have access to a stable and high-speed internet connection.
- **Blockchain Technology**: The platform's reliability on blockchain for secure transactions assumes the stability and security of the Ethereum network.
- **Gamer Hardware Requirements**: Gamers are expected to have devices that meet the minimum system requirements for the Daylight ES365 application. These are some examples of the Requirements for both gaming High level AAA games and simultaneously mining at the same time:
    - ASUS TUF Gaming NVIDIA GeForce RTX 4090 OC Edition Graphics Card
    - NVIDIA GeForce RTX 3080 Founders Edition
    - MSI NVIDIA GeForce RTX 4070 Graphics Card.
    - NVIDIA GeForce RTX 4080 16GB DDR 6X Graphics Card.

3. Functional Requirements

3.1 User Registration and Authentication

- **Register**
    - **Description**: Users can register with the platform by providing their email address and choosing a password.
    - **Details**:
        - Users enter their email address and create a password.
        - Confirmation email is sent to verify the user's email address.
        - Upon verification, the user's account is activated.
- **Login/Logout**
    - **Description**: Registered users can log in to and log out of their accounts.
    - **Details**:
        - Users enter their registered email and password to log in.
        - Session management ensures users remain logged in until they manually log out or the session expires.

- Logging out terminates the session and returns the user to the login screen.

3.2 Dashboard

- **Gamer Dashboard**
  - **Description**: Gamers have access to a personalized dashboard to manage their gaming activities.
  - **Functionalities**:
    - **View Balance and Transaction History**: Displays current account balance and a history of transactions related to game bookings and payments.
    - **View Upcoming Booked Games and Game Play History**: Lists scheduled game sessions and a history of past game sessions played.
    - **Receive Notifications and Alerts**: Alerts users about upcoming bookings, payment confirmations, or important platform updates.
    - **Add Coins:** All currencies/crypto are converted into DL365 coins for making the payments.
- **Miner Dashboard**
  - **Description**: Miners have access to a dashboard tailored to manage their equipment and earnings.
  - **Functionalities**:
    - **View Balance and Earnings History**: Shows current account balance and mined crypto coins from renting out equipment and from the mining process.
    - **Manage Games Available for Rent**: Allows miners set pricing per hour, and manage availability schedules.
    - **Set Game Availability and Pricing**: Provides tools to adjust the availability of equipment and games and update pricing based on demand or specific criteria.
    - **Hardware Setup:** Detailed information about the computer hardware, including CPU, GPU, RAM, and internet connection speed.
    - **Smart Contract:** Manage smart contracts and get the gamers digitally signed on the contract.
- **Smart Contract Execution**:
  - **Matchmaking Service**: The platform uses smart contracts to connect Miners with Gamers based on the latter's requirements and the former's availability.
  - **Session Initiation**: Upon successful creation of a smart contract with the action "Start," the encrypted profiles from Daylight ES365 are downloaded to the Miner’s computer, and the Gamer is able to start playing.
  - **Session Termination**: After the gaming session ends, the encrypted profiles are uploaded back to Daylight ES365, and a smart contract transaction with the action "Finish" is created.
  - **Payment Processing**: The ESX (Daylight ES365’s cryptocurrency) is transferred from the Gamer’s wallet to the Miner’s wallet upon session completion.

3.3 Game Rental Management

- **Search Game**
  - **Description**: Gamers can search for available games based on specific criteria.
  - **Functionalities**:
    - **Search Parameters**: Search by location, time frame (date and time), game genre, and game name, popular/trending games.
    - **Results Display**: Display search results showing available games matching the criteria specified by the gamer.
- **Book Game**
  - **Description**: Gamers can book a game session and make payment using DL365 Coin.
  - **Functionalities**:
    - **Adjust Availability**: Select a suitable time slot for the game session.
    - **Payment**: Transfer DL365 Coin from the gamer's account to an escrow account managed by the platform.
    - **Secret Code Generation**: Generate a unique secret code for the booked time slot to authorize the game session which is provided by the admin which is unique to every user.
    - **Smart Contract:** Accept the smart contracts created by the Miner to the gamer and the contract is signed digitally by the gamer.

3.4 Game Play Management

- **Start Game Session**
  - **Description**: Gamers initiate the authorized game session using the generated secret code.
  - **Functionalities**:
    - **Authorization**: Enter the secret code to connect to the miner's equipment and start the game session.
    - **Streaming Connection**: Establish a secure connection between the gamer's application and the provider's application for real-time game streaming.
- **End Game Session**
  - **Description**: Terminate the game session upon completion of gameplay.
  - **Functionalities**:
    - **End Session**: Automatically terminate and disconnect the streaming connection.
- **System Compatability Analysis:**

  Implementing a module to check the system compatibility of the user (gamer). In case of non-compatibility, the gamer may request equipment from the miner on lease.

3.5 Cryptocurrency Transactions

- **Payment Handling**
  - **Description**: Manage cryptocurrency transactions for game rentals and earnings.
  - **Functionalities**:
    - **Escrow Management**: Hold payments in an escrow account until the game session is successfully completed.
    - **Transfer Funds**: After successful gameplay, transfer funds from the escrow account to the miner's account, deducting the platform's transaction fee.
- **Mining Earnings**
  - **Description**: Track mining progress and calculate earnings based on the mined cryptocurrency's current value.
  - **Functionalities**:
    - **Real-Time Tracking**: Monitor usage of mining equipment and calculate earnings in real-time.
    - **Account Updates**: Reflect updated earnings in the miner's account balance on the platform dashboard.

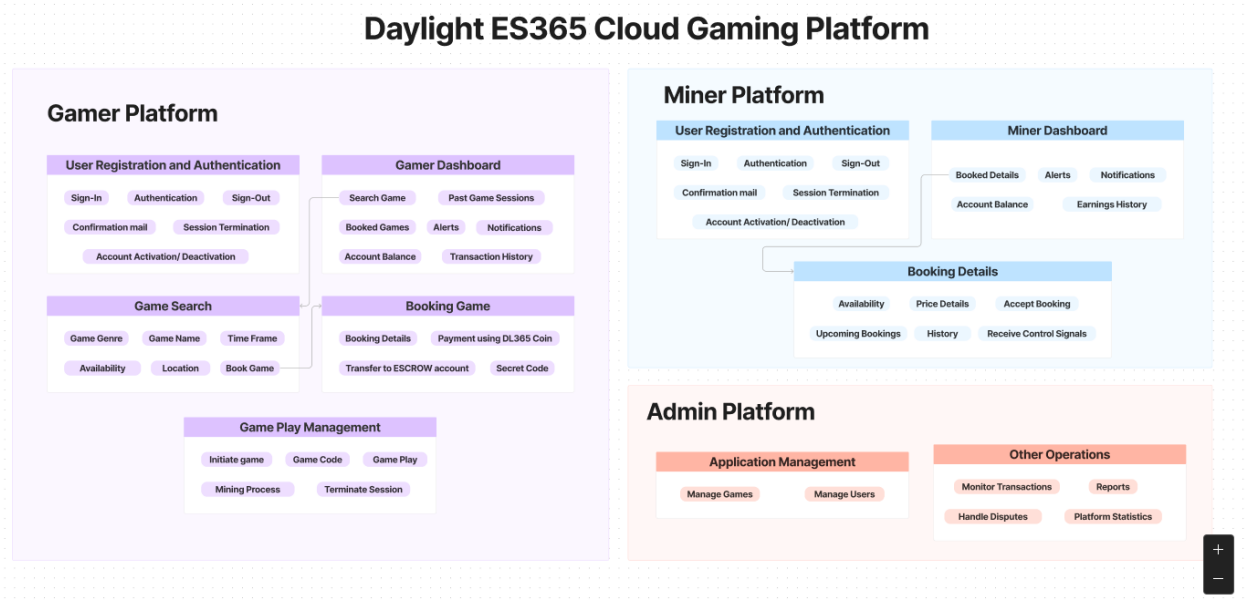
3.6 Admin Interface

The platform ensures continuous support for both Miners and Gamers, providing updates, new features, and improvements based on user feedback and technological advancements.

- **Manage User Accounts:**
  - **Update User Accounts:** Admins can modify user details such as email, password, and account status. This ensures user data is current and accurate.
  - **Delete User Accounts:** Admins can permanently remove user accounts from the platform if needed, such as in cases of fraudulent activity or user requests.
- **Manage Games:**
  - **Add Games:** Admins can manually add new games to the platform. This includes setting game details like name, genre, price per hour, and availability.
  - **Delete Games:** Admins can remove games that are no longer available or that violate platform policies.
  - **Update Games:** Admins can modify game details, such as changing the pricing, updating game descriptions, or adjusting availability schedules.
- **Monitor Transactions and Handle Disputes:**

- **Transaction Monitoring:** Admins can view and track all transactions between gamers and miners. This includes payments, escrow holdings, and final transfers.
- **Dispute Handling:** Admins can resolve disputes between gamers and miners. This may involve reviewing transaction histories, game session logs, and user communications to make informed decisions.

- **View Platform Statistics and Reports:**
  - **Platform Statistics:** Admins can access real-time data on platform usage, including the number of active users, active game sessions, and overall system performance.
  - **Reports:** Admins can create reports based on the statistics based on various metrics such as revenue, user activity, and system uptime. These reports help in making data-driven decisions to improve platform operations and user experience.



4. Non-Functional Requirements

- **Performance:** The system must support concurrent gamers with minimal latency. Game streaming latency should not exceed 100 ms.
- **Security :** All data in transit and at rest must be encrypted. Secure authentication and authorization mechanisms. Regular security audits and compliance with industry standards.
- **Usability:** The user interface should be intuitive and easy to navigate. Comprehensive help and support resources available to users.
- **Scalability:** The system should scale horizontally to handle increased load. Automated load balancing and failover mechanisms.
- **Reliability:** System uptime should be at least 99.9%. Redundant systems and disaster recovery plans in place.

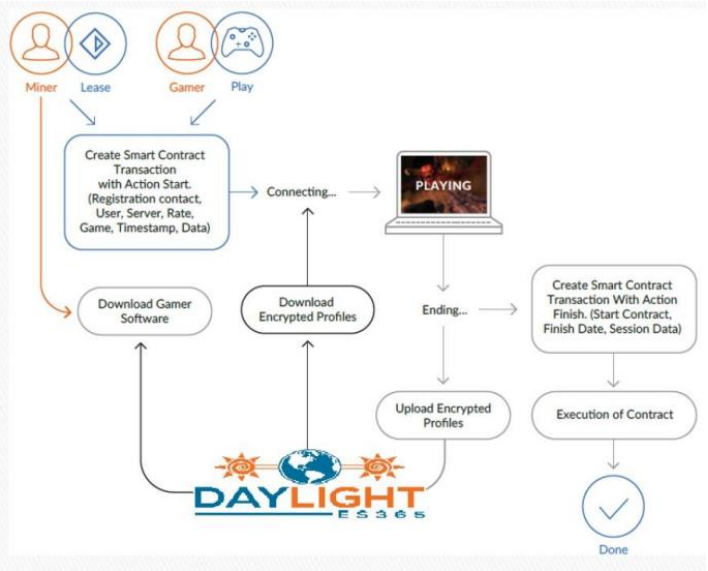
5. System Architecture

5.1 Overview

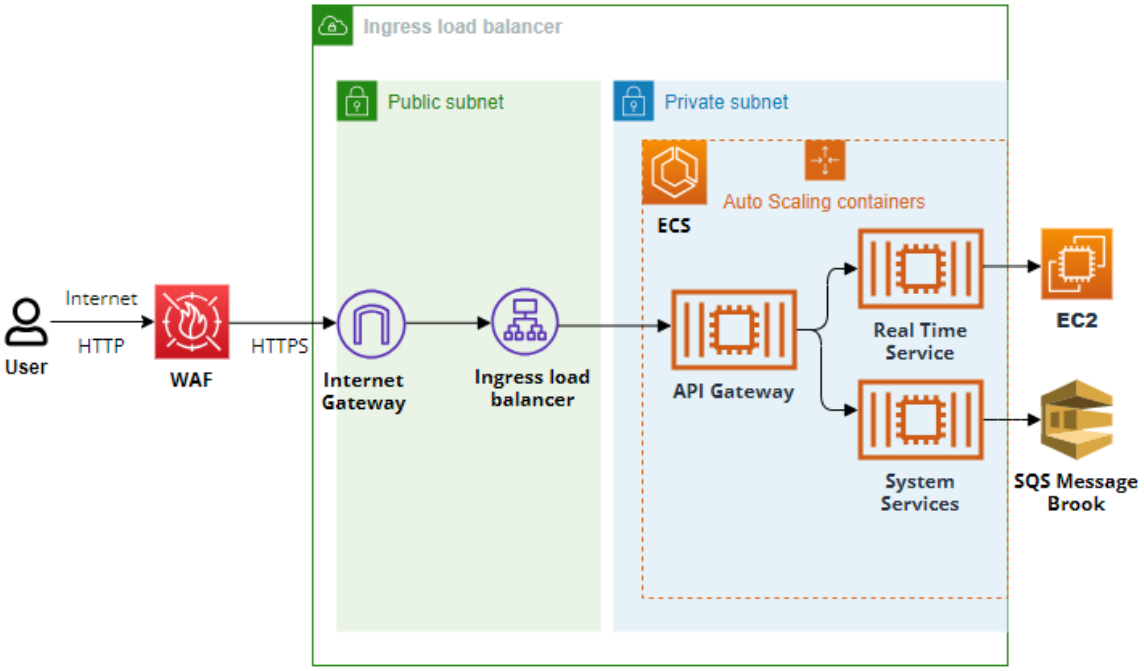
The system architecture is designed around a microservices model, enabling scalability, maintainability, and flexibility. It includes the following components:

- **Frontend**: Web-based dashboard and cross-platform applications.
- **Backend Services**: RESTful APIs for user management, game rental, and cryptocurrency transactions.
- **Blockchain Integration**: Smart contracts on the Ethereum blockchain for secure transactions.
- **Game Streaming**: Low-latency streaming service to deliver high-quality gaming experiences.
- **Database**: Distributed database for user data, game information, and transaction records.

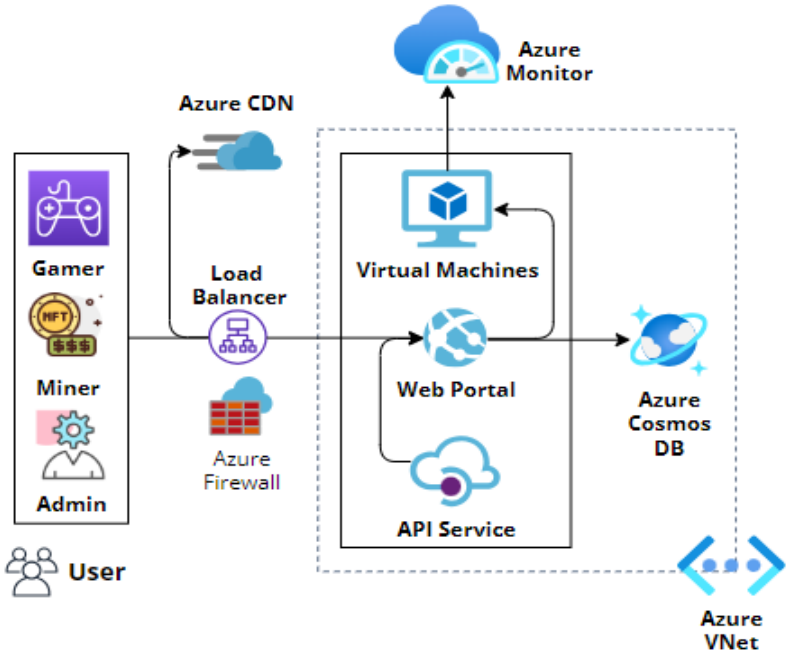
5.2 Component Diagram



5.3 AWS Server Architecture



5.4 Azure Server Architecture



6. Design Considerations

6.1 Assumptions and Dependencies

- Users have access to reliable internet connections.

- The platform will integrate with third-party APIs for cryptocurrency prices and mining pool data.
- The Ethereum blockchain will be used for handling smart contracts and DL365 Coin transactions.

6.2 Risks and Mitigations

- **High Latency**: Implement continuous Delivery Network to reduce latency.
- **Security Breaches**: Regular security audits, encryption, and secure coding practices.
- **Scalability Issues**: Use microservices architecture and auto-scaling features of cloud providers.

6. Conclusion

The Daylight ES365 platform represents a significant advancement in the cloud gaming industry. By leveraging blockchain technology and a decentralized model, it offers a compelling solution for both gamers and miners. Gamers gain access to high-quality gaming experiences without the burden of expensive hardware, while miners can effectively monetize their existing resources.

The platform's thoughtful design considerations and future-oriented development plans position it well for success in the evolving eSports and gaming landscape. Overall, Daylight ES365 is poised to transform the way gaming resources are utilized, making high-end gaming more accessible and enjoyable for a global audience.

2. TECHNICAL PROPOSAL

Proposed Technologies:

For the development of the Daylight ES365 Cloud Gaming Platform, we propose using Tauri for the backend (Rust) and Next.js for the frontend. This section provides an overview of these technologies and the reasons for choosing them over other options.

Technology Overview

- **Next.js (Frontend)**
  Next.js is a popular React framework that enables server-side rendering and static site generation. It is known for its simplicity, speed, and scalability, making it a solid choice for developing robust web applications.
- **Tauri (Backend - Rust)**
  Tauri is a framework for building tiny, fast binaries for all major desktop platforms. It leverages Rust for the backend, known for its performance, safety, and concurrency capabilities. Tauri produces much smaller binaries and offers more control over the application's security and resource usage.

Reasons for Choosing Tauri and Next.js

Tauri (Backend - Rust)

1. **Performance and Efficiency**: Tauri, using Rust, generates significantly smaller binaries compared to other frameworks, resulting in faster load times and reduced memory usage. This is crucial for a gaming platform that demands high performance.

2. **Security**: Tauri offers advanced security features out of the box, including strong isolation between the web and native layers, making it a secure choicefor handling sensitive user data and transactions.

3. **Concurrency and Safety**: Rust is renowned for its memory safety and concurrency model, which helps in developing a robust and crash-free backend.

4. **Cross-Platform Support**: Tauri supports macOS, Windows, and Linux, ensuring broad compatibility for users regardless of their operating system.

5. **Developer Experience**: Tauri integrates seamlessly with modern front-end tools and frameworks, providing a smooth development experience.

Next.js (Frontend)

1. **Server-Side Rendering (SSR)**: Next.js enables SSR, which improves performance and SEO by rendering pages on the server before sending them to the client. This is beneficial for a gaming platform where performance and discoverability are key.

2. **Static Site Generation (SSG)**: Next.js supports SSG, allowing for pre-rendering of pages at build time. This can significantly improve load times and reduce server load.

3. **Scalability**: Next.js is highly scalable, making it suitable for a platform expected to handle a large number of users and high traffic.

4. **Rich Ecosystem**: Next.js has a rich ecosystem and community support, providing access to a plethora of plugins, tools, and best practices.

3.2 Alternative Technologies Considered

a) Classification based on Technology:

Technology	Pros	Cons
Qt	Mature framework with extensive cross-platform support, strong performance, rich set of libraries and tools.	Larger binary sizes, steep learning curve, and licensing costs for commercial applications.
Tauri	High performance, lightweight, smaller binary sizes, strong control over security, use Rust for core functionalities, growing ecosystem.	Higher initial learning curve due to Rust, which is relatively new compared to other technologies, and a smaller community.
Electron	Wide adoption, strong community support, & a rich plugin ecosystem.	Larger binary sizes, higher memory usage, and less control over security.
React Native	Excellent for building mobile applications, and cross-platform capabilities.	Primarily targeted at mobile development, it is less suited for desktop applications.
Flutter	Great for building cross-platform applications, and strong performance.	Larger binary sizes are less mature for desktop development compared to Tauri.

b) Classification based on Features:

Feature	**Tauri** (Primary Choice)	**Qt** (Alternate choice)	**Electron**	**React Native**
Performance	High performance, lightweight, smaller binary sizes.	Strong performance, suitable for high-performance applications.	Adequate for many applications, but larger binary sizes and higher memory usage compared to Tauri.	Good performance for mobile applications, but not optimized for desktop applications.
Security	Strong control over security, uses Rust for core functionalities.	Advanced security features, but requires additional effort to implement securely.	Less control over security compared to Tauri.	Decent security features, but primarily targeted at mobile applications.
Cross-Platform Support	Supports Windows, macOS, and Linux with efficient resource usage.	Extensive cross-platform support, including mobile, desktop, and embedded systems.	Supports Windows, macOS, and Linux, but with larger binary sizes.	Excellent for mobile platforms (iOS and Android), limited desktop support.

Ecosystem	Growing ecosystem, strong Rust community support.	Mature ecosystem with extensive libraries and tools, but steeper learning curve.	Wide adoption, strong community support, and rich plugin ecosystem.	Large ecosystem for mobile development, smaller for desktop.
Developer Experience	Easier to secure and optimize due to Rust, but higher initial learning curve.	Requires proficiency in C++ and Qt-specific knowledge, steeper learning curve.	Easy to get started with JavaScript and a large number of available plugins.	Familiar to web developers with JavaScript experience, but primarily for mobile development.
Scalability	Highly scalable with low resource usage, suitable for high-performance applications.	Highly scalable, but development complexity may increase with scale.	Scalable for many applications, but may require more resources due to larger binaries.	Scalable for mobile applications, but less suited for large-scale desktop applications.
Binary Size	Smaller binary sizes, leading to faster load times and lower resource consumption.	Larger binary sizes, but highly optimized for performance.	Larger binary sizes compared to Tauri, which can impact performance and resource usage.	Typically, smaller binary sizes for mobile, but desktop binaries can be larger.
Latency	Optimized for low-latency applications, suitable for high-performance gaming and real-time applications.	Suitable for high-performance applications, but requires careful optimization to achieve low latency.	Adequate for many applications, but may not meet the low-latency requirements for high-performance gaming.	Suitable for mobile applications with decent latency, but not optimized for low-latency desktop applications.
Learning Curve	Steeper due to Rust, but offers long-term benefits in terms of performance and security.	Steeper learning curve due to C++ and Qt-specific knowledge requirements.	Easier for developers familiar with JavaScript, but may require additional effort to optimize for performance and security.	Easier for developers familiar with JavaScript, but additional effort required for optimizing desktop performance.
Licensing Costs	No licensing costs, open-source.	Licensing costs for commercial applications, free for open-source development.	No licensing costs, open-source.	No licensing costs, open-source.

CONCLUSION

Tauri for the backend (using Rust) and Next.js for the frontend provide a robust, secure, and efficient technology stack for the Daylight ES365 Cloud Gaming Platform. Tauri's ability to produce small, high-performance binaries combined with Rust's safety and concurrency benefits, along with Next.js server-side rendering and scalability, make them well-suited for delivering a seamless and responsive gaming experience.

While other technologies like Electron, React Native, Flutter, and Qt were considered, Tauri and Next.js were selected for their optimal balance of performance, security, and developer experience.