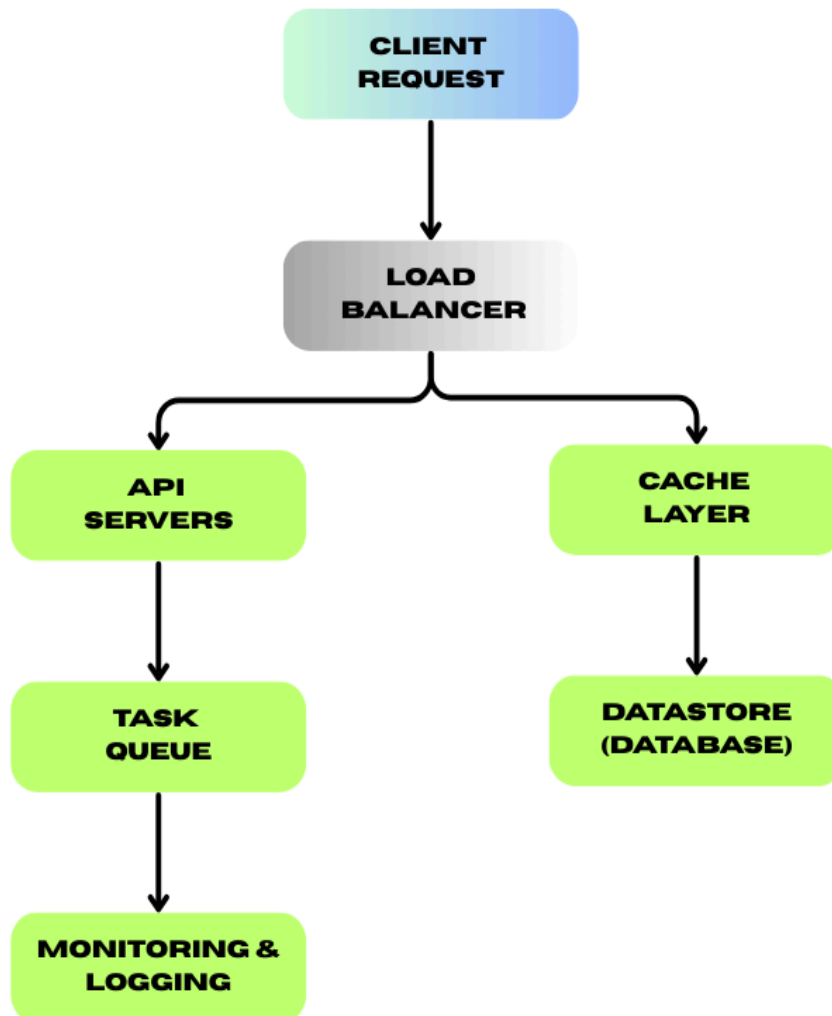


# High Level System Design



## Key Components:

- **Load Balancer:** This helps to evenly distribute incoming traffic across several servers, ensuring no single server gets overwhelmed. It's crucial for handling high traffic.
- **API Servers:** These are responsible for processing all incoming optimization requests. They assign ships to available berths based on constraints like arrival rates and priorities.
- **Cache Layer:** Frequently accessed data like berth availability is stored here for quick access. This helps reduce the load on the database and speeds up response times.
- **Task Queue:** Some tasks, like complex optimization calculations, take time to process. The task queue ensures these long-running jobs don't block other requests.
- **Datastore (Database):** The central repository of data where ship schedules, berth availability, and other details are stored.
- **Monitoring & Logging:** These tools track the system's health, log errors, and ensure everything is running smoothly. They help identify issues before they impact users.

## Scaling & Failure Handling:

- **Scaling:** The system is designed to grow as needed. If traffic spikes, more API servers and cache nodes can be added. If the database grows, replicas can be added for better performance.
- **Failure Handling:** If a server fails, the load balancer redirects requests to other servers. If the cache goes down, the system continues to work, but responses might be slower since it'll fetch data from the database. In case of a database failure, replication ensures there's always a backup available to keep the system running.