Components of System Design

1. **Load Balancer**
   - **Function**: Distributes incoming requests across multiple servers to prevent overloading and ensure availability.
   - **Types**:
     - Layer 4: Operates at the network layer using IP and port information.
     - Layer 7: Operates at the application layer using content-based rules.
     - Global Load Balancers: Distribute traffic across geographic regions.
     - Application Load Balancers: Optimized for specific applications or protocols.
2. **Caching**
   - **Function**: Temporarily stores frequently accessed data for faster retrieval.
   - **Benefits**:
     - Reduces database load.
     - Improves response times.
     - Enhances user experience.
3. **Content Delivery Network (CDN)**
   - **Function**: Speeds up content delivery using a distributed network of servers.
   - **How it works**: Delivers cached content from the nearest server to the user, reducing latency.
4. **API Gateways**
   - **Function**: Acts as a single entry point for client requests to backend services.
   - **Features**:
     - Request routing and aggregation.
     - Security (authentication and authorization).
     - Load management and traffic monitoring.
5. **Key-Value Stores**
   - **Function**: Stores data as key-value pairs for quick access.
   - **Types**:
     - Persistent: Stores data on disk for durability.
     - In-memory: Optimized for speed.
6. **Blob Storage and Databases**
   - **Blob Storage**: Manages unstructured data like images, videos, and documents.
   - **Databases**:
     - Relational (RDBMS): For structured data with relationships.
     - NoSQL: For flexible and scalable storage.
     - In-memory Databases: For low-latency access.
7. **Rate Limiters**
   - **Function**: Controls the frequency of requests or operations.
   - **Types**:

    - Request rate limiters.
    - User-specific limiters.
    - Token bucket algorithms.
8. **Monitoring Systems**
    - **Function**: Tracks performance, availability, and metrics.
    - **Types**:
        - Network Monitoring.
        - System Monitoring.
        - Application Monitoring.
9. **Distributed Messaging Queues**
    - **Function**: Facilitates asynchronous communication between system components.
    - **Types**:
        - Point-to-point: Direct message delivery.
        - Publish-subscribe: Broadcasts messages to subscribers.
10. **Distributed Unique ID Generator**
    - **Function**: Creates unique IDs for objects in a distributed system.
    - **Methods**:
        - Centralized services.
        - Distributed consensus algorithms.
        - Timestamp-based generation.
11. **Distributed Search**
    - **Function**: Enables scalable search operations across large datasets.
    - **Implementation**:
        - Distributed search engines (e.g., Elasticsearch).
        - Databases with search capabilities.
        - Cloud-based search services.
12. **Distributed Logging Services**
    - **Function**: Collects and analyzes logs from various components.
    - **Approaches**:
        - Centralized or distributed logging systems.
        - Cloud-based logging solutions.
13. **Distributed Task Scheduler**
    - **Function**: Automates and schedules tasks in a distributed system.
    - **Types**:
        - Standalone schedulers.
        - Built-in system schedulers (e.g., Kubernetes cron jobs).
        - Cloud-based schedulers (e.g., AWS CloudWatch).

Key Considerations for System Design

- Scalability: Ensuring components handle growth efficiently.
- Resilience: Designing for fault tolerance and recovery.
- Security: Protecting data and communication.
- Cost Efficiency: Balancing resources and expenses.