

Environmental Monitoring and Prediction System: Non Functional Requirements

SWENG 837

Laina Purachev



Environmental Monitoring and Prediction System:

Create a system that collects environmental data (air quality, water quality, deforestation) to monitor changes and predict future environmental conditions.

Non Functional Requirements

Performance Requirements: Scalability

100 concurrent users or 100 TPS: Vertical scaling can be used in this case. Vertical scaling involves adding more power to existing resources, like upgrading server CPU, RAM, and storage.

1000 concurrent users or 1000 TPS: Optimizing SQL database queries would be a good way to handle this increase in users. Indexing makes columns faster to query by creating pointers to where data is stored within a database. Caching stores data that is frequently accessed, and usually rarely changed, closer to end users. Can help reduce the load on the database and servers. It can also help improve latency which leads to a faster system.

Non Functional Requirements

Performance Requirements: Scalability

10k concurrent users or 3000 TPS: Horizontal scaling is when more hardware nodes are added to the system, like adding more servers. It is used instead of vertical scaling. If one server goes down, other servers can be used, making less downtime and less performance issues. Horizontal scaling is cost effective and helps application resilience.

100k concurrent users: Cloud storage would be the best way to handle this many users. Cloud storage has data in various logical pools that are then distributed over a number of servers. This is perfect for large numbers of users or increased growth because the storage is there, and all that needs to be done is to buy more storage as needed. Load balancing software can also be used to help scalability at this size.

Non Functional Requirements

Performance Requirements: Response Time

100–500 milliseconds is ideal for user experience

Optimize database

Configure caching

Non Functional Requirements

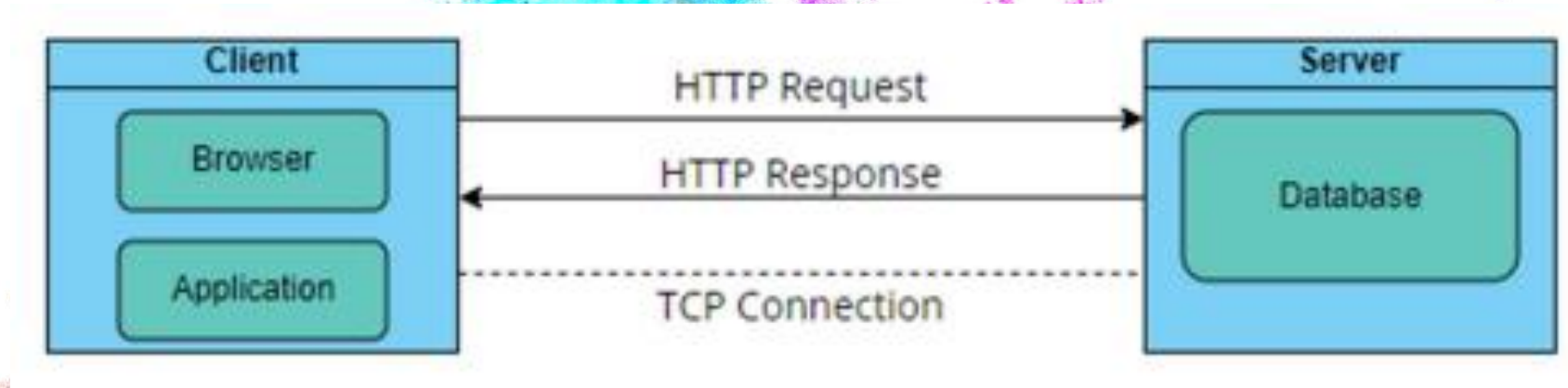
Performance Requirements: Throughput

Throughput: the amount of material or items passing through the system

1. Review Your Existing Workflow.
2. Eliminate Bottlenecks.
3. Reduce Equipment Downtime.
4. Reduce Parts Rejection Rate.
5. Improve Employee Training.
6. Use Factory Automation.

Non Functional Requirements

Security Requirements: Authentication



HTTP is being used. HTTP is a secure application layer protocol that transfers information between devices on the network. It is used for communicating between clients and servers

Non Functional Requirements

Security Requirements: Authorization

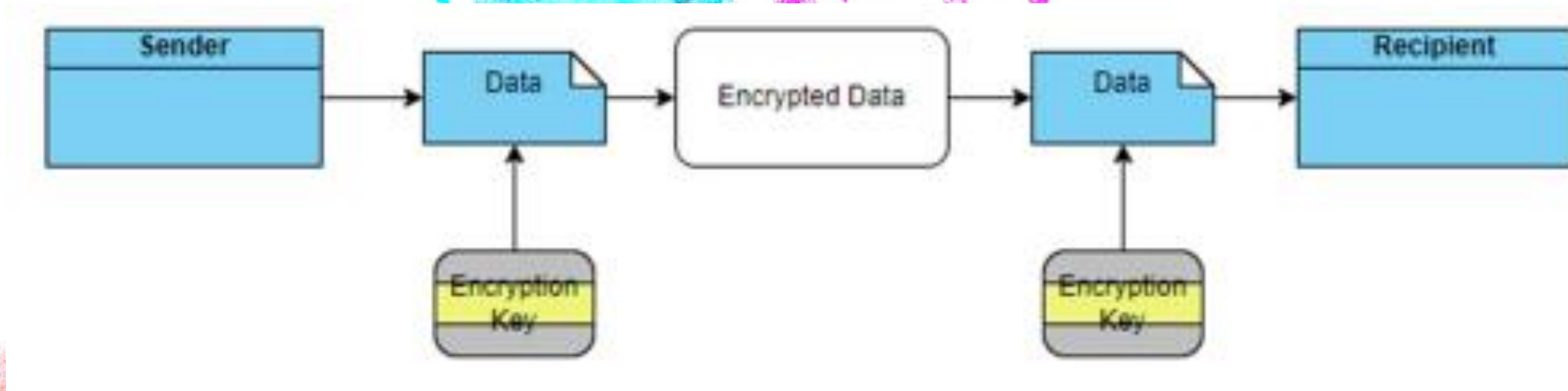
Role-based access control (RBAC): to differentiate between a guest and a registered user

Assigns permissions to users based on their role within an organization

Grants access based only on the role of the user within an organization

Non Functional Requirements

Security Requirements: Data Encryption



End-to-end encryption

Using encryption techniques to ensure data is protected when being sent and at rest

Non Functional Requirements

Maintainability Requirements: Code Modularity

Code modularity involves dividing a system into smaller, independent modules or components, each with a specific function.

This will be done through the use of UML Diagrams

Non Functional Requirements

Maintainability Requirements: Documentation

Documenting code enhances readability, maintainability, and longevity, ultimately saving time, reducing errors, and improving overall software quality.

Determine your audience, define the scope, use a standardized structure, write descriptive titles and explanations, document parameters and return values, document any changes made and when, have a backup.

Non Functional Requirements

Maintainability Requirements: Testing Strategies

Front-End Testing Strategy: making sure everything looks and functions for the user

Static Testing Strategy. A static test evaluates the quality of a system without actually running the system

Cross-platform testing: testing across platforms to ensure application can be used by all