System Design Assignment

Requirments

Given

- assume users registration is already implemented

Functional requirements

- a user can subscribe to other users
- a user can publish a new message to their subscribers (5k request/s)
- a user can view messages from the users they follow (300k request/s)

Technical requirements

- ensure consistency on write; If a user write a message and then read the messages he wrote, he should always see the message he just wrote
- consistency on read is not required; If a user write a message, it is ok if its subscribers reads it with some delay
- ensure communication between client and server can't be eavesdroped
- ensure communication between client and server can't be impersonated

Hints

The product attract two different kinds of users

- quidam
  - most of you user base
  - they have few subscribers but subscribe to many users
  - they often check for new messages from their subscriptions
- influencers
  - they have many subscribers
  - they publish often

The system may use

- replication

- partitioning
- caching
- etc...

in any case, justify your decision and their impact on availability, consistency, and tolerence to network/infrastructure failure

Report

You are to produce a report that documents the architecture for your proposed system. Your target audience for reports and documents is a developer who is somewhat but not intimately familiar with your project.

See below the topics you want to consider in your report.

It is ok if you don't have enough time or knowledge to cover everything.

Summary

half page over viewing the key points in your design, targeted to a developer.

Architecture

Give the overall structure of the system that you are proposing to implement, with descriptions of each major component and of the interactions among them.

These components are to be primarily services, files, and databases. In your descriptions, concentrate on requirements, scalability, evolvability, observability, testability, etc., rather than on lower level concepts such as classes, variables and control flow.

However, you may wish to discuss important abstractions, patterns, classes, data structures or algorithms that are critical to the successful implementation of your architecture.

Your system's architecture should be easy to understand, with simple interfaces, and modest interactions among its components.

Diagrams

You should use diagrams that clearly illustrate the structure of your system. Do not give diagrams that are too low level, eg, do not give diagrams of details within objects, nor within modules.

Data

List information transmitted to/from the system. Concentrate on information content. For information to be stored in a database, describe the componemt interactions involved from the issuance of the data

to its persistence in the database. In your descriptions, concentrate on concurrent read/write and consistency guaranties the system provides.

Test Plan

Briefly describe how you plan on testing your system.

Performance

Discuss any parts of the system that are likely to be performance critical, i.e., which might not run fast enough.

Evolvability

Discuss how your architecture is designed to support future changes in the your system.