

# Milestone 4: Creatively Extend Your Milestone 2 or Milestone 3

**Due: 9th April 2021**

## Overview

In the marketplace, key-value stores differentiate themselves by offering a large variety of innovative features, aiming for user lock-in, thus, counteracting the initial goals of offering a lean and simple system that scales elastically.

The goal of this milestone is to have you pick an interesting feature based on our below specified guidelines and realize this feature, either as part of your Milestone 2 or your Milestone 3 designs and implementations.

## Learning objectives

With this milestone, we pursue the following learning objectives:

- Deal with a distributed system of enhanced complexity
- Use your imagination and apply what you have learned in class to innovatively extend your storage service

## Guidelines for picking a feature

- The extension you pick should be creative and innovative
- The extension you pick must be non-trivial (e.g., a simple UI for using your storage service would not qualify)
- Your extension should have a “distributed footprint,” i.e., the realization of the extension should involve multiple distributed components; for example, the replication mechanisms of Milestone 3 involve at least three distributed storage servers
- For example, think about the CAP trade-offs and how your service could better promote or guarantee variants of any or all of the three properties

# Sample features

These features are meant as examples that meet the above specified requirements. You are free to ignore the below and pick something of your liking that meets the above requirements.

- Stricter consistency model
  - Data-centric consistency (e.g., sequential, causal consistency)
  - Client-centric consistency (e.g., read-your-write consistency)
- Transactions
  - ACID guarantees for a set of read/write operations
  - API extension for commit operation
- Notification mechanism
  - Subscribe to data mutations (value changes, deletion)
  - API extension for (un)subscribe operation
- Support for tables, queries and derived tables
  - Introduce the definition of tables
  - Support query capabilities, where a client submits a query that selects certain rows from a table, based on a selection predicate (e.g., `SELECT FROM Students Where Mark > 90`; here, we assume there is a table called Students, possibly with Student Name and Mark columns)
  - Materialize the query dynamically as the data the query depends on changes (i.e., support derived tables)
- Dynamic load balancing
  - Servers transfer data amongst each other in order to minimize the overall load-factor of each server
- Fault-tolerance
  - Implement other mechanisms for ensuring data consistency during a failure (e.g. transactional logs, vector timestamps)
- Support for byzantine failures
  - In the scenario of a hijacked server/datacenter, a malicious user can send malformed or spoofed messages from the compromised server to clients and/or other servers
  - Develop a mechanism that helps mitigate an attacker's ability to weaken the system's consistency in various scenarios such as:
    - Attacker attempts to return a wrong value to a client's get request
    - Attacker attempts to update key-value pairs on other servers that were not initiated by a client
- Network partitioning

- In real-life systems, arbitrarily slow or downed links creating partitions are not unexpected
- Design a mechanism for maintaining high availability and consistency despite your storage service being partitioned in two or more partitions
- Once the links have been “restored”, this mechanism should include a recovery phase allowing the system to merge partitions transparently to the clients while maintaining consistency throughout the system

## Detailed milestone description

### Guidelines for implementation

Again, for realizing this milestone, you have the opportunity to make all design decisions on your own. In this milestone, we give you maximal flexibility to be creative. However, you need to follow our above specified guidelines for picking features.

Please note, a portion of the mark will be allocated for creativity, originality, and innovation in this milestone; thus, we recommend you to not share your feature idea outside your group. You may discuss your feature idea with TAs by making a private post on Piazza. A portion of the mark will also be allocated to the perceived difficulty and challenge of realizing the envisioned feature.

Please write at least 10 custom tests to test your implementation.

### Suggested development plan

You should base the design and implementation of your feature on your design for Milestone 2 or Milestone 3.

### Getting the project files

There are no new project files for Milestone 4. Base your project on your Milestone 2 or Milestone 3 implementation and the project files distributed for Milestone 2.

### Deliverables & code submission

By the deadline, you must hand in your software artifacts that implement all the coding requirements and include all necessary libraries and the build scripts. Also, provide a design document of ~3 pages(+performance reports), where you clearly describe your innovative extension, including any design decisions you made in realizing the extension. Also, provide sufficient evidence and explanation in your design document that demonstrates that your extension is working correctly (e.g., you could provide a performance evaluation and/or unit tests.)

To demonstrate your work, also **include a short video walking through an example use case of your extension**, keep the video to between 5-10 minutes. Share the video [here](https://forms.office.com/Pages/ResponsePage.aspx?id=JsKqeAMvTUuQN7RtVsVSEFOgWGfQTVJHs3PgUUV15_JUOFZQUjdJTjkxMk9SWTZINk5PSTNBQ1c1Mi4u_) [\(https://forms.office.com/Pages/ResponsePage.aspx?id=JsKqeAMvTUuQN7RtVsVSEFOgWGfQTVJHs3PgUUV15\\_JUOFZQUjdJTjkxMk9SWTZINk5PSTNBQ1c1Mi4u\\_\)](https://forms.office.com/Pages/ResponsePage.aspx?id=JsKqeAMvTUuQN7RtVsVSEFOgWGfQTVJHs3PgUUV15_JUOFZQUjdJTjkxMk9SWTZINk5PSTNBQ1c1Mi4u_)

## Marking guidelines and marking scheme

All the code you submit must be compatible with the build scripts, interfaces and test cases.

## Document Revisions

Changes to the milestone handout after posting it are tracked here.

### Date Change

None