

# Project 2: WeChat

Emery Berger, COMPSCI 630

Due May 2, 2018

## 1 Overview

For this project, you will be implementing *WeChat*, a totally distributed group chat application written in Golang (a.k.a., Go). This assignment will give you insight into the nature of distributed systems and the problems that arise when you have multiple distributed actors involved.

## 2 Implementation

Your WeChat implementation will allow users with an invite code (provided in advance) to communicate over the network without the need for a central chat server. Just as in ordinary chat, each participant should be able to send and receive small text messages. Unlike ordinary chat, your application should be able to handle a large number of simultaneous chat participants (i.e., it must be multithreaded).

While WeChat should be totally distributed for handling chat functionality, you will still need to provide a central discovery server at a known IP address. Each client should register their location with the service at start up thus allowing other nodes to easily find them.

In addition to supporting multiple clients, your app should have the following properties:

**Consistency** Each WeChat user should see the exact same chain of messages, independent of issues like network delay. In particular, you should implement causal consistency: a user should see messages in strict causal order (implemented via vector (Lamport) clocks).

**Fault-tolerance** WeChat users may periodically get disconnected from the system. Any messages sent to them should eventually appear after they reconnect (i.e. no messages should be lost).

**Encryption** All data messages should be encrypted so that no one who does not know the conference code will be able to read chat messages for the appropriate conference.

### 3 Evaluation

You will be evaluated on the following basis:

1. Correctness of Implementation
2. Directory Service
3. Consistency
4. Fault-Tolerance
5. Encryption
6. Coding Style

### 4 Resources

We have provided a centralized solution in the following repo: [Centralized WeChat](#). Inside this repo is the following:

1. A description of the API that you need to provide.
2. A python reference client that you can use to test your system.
3. A python reference server that provides basic functionality.

You need to implement a decentralized version of this client/server architecture, following the guidelines of the API description [here](#).

**Note:** Since your solution is fully distributed, each node needs to handle things that would be managed by this central server (i.e., chat history, channel management, etc...)

**Remember you MUST implement your code in go!**