

15-319 / 15-619

Cloud Computing

Recitation 7

October 9, 2018

Overview

- **Last week's reflection**
 - Project 3.1
 - OLI Unit 3 - Module 10, 11, 12
 - Quiz 5
- **This week's schedule**
 - Project 3.2
 - OLI Unit 3 - Module 13
 - Quiz 6
- **Team Project, Twitter Analytics**
 - Phase 1 is out!
 - Early bird bonus, 10/14

Reminder

- Respond to the DDOS warning emails if you have received them and follow the resource abuse process.

Last Week

- **Unit 3: Virtualizing Resources for the Cloud**
 - Module 10: Resource virtualization (memory)
 - Module 11: Resource virtualization (I/O)
 - Module 12: Case Study
- **Quiz 5**
- **Project 3.1**
 - Files v/s Databases (SQL & NoSQL)
 - Flat files
 - MySQL
 - HBase
 - Read the NoSQL and HBase basics primer
- **Team Programming Exercise on Cloud9**

This Week

- **OLI : Module 13 - Storage and network virtualization**
- **Quiz 6 - Friday, October 12**
- **Project 3.2 - Sunday, October 14**
 - Social Networking Timeline with Heterogenous Backends
 - MySQL
 - Neo4j
 - MongoDB
 - Choosing Databases
 - MongoDB Primer
- **Programming exercise for consistency on Cloud9**
- **Team Project, Phase 1 released**

Conceptual Topics - OLI Content

- **Unit 3 - Module 13: Storage and network virtualization**
 - Software Defined Data Center (SDDC)
 - Software Defined Networking (SDN)
 - Device virtualization
 - Link virtualization
 - Software Defined Storage (SDS)
 - IOFlow
- **Quiz 6**
 - **Remember to hit submit before the deadline!**

Individual Projects

- DONE
 - P3.1: Files v/s Databases - comparison and Usage of flat files, MySQL, Redis, and HBase
 - NoSQL Primer
 - HBase Basics Primer
- **NOW**
 - P3.2: Social networking with heterogeneous backends
 - MongoDB Primer
- Coming Up
 - P3.3: Replication and Consistency models

A Social Network Service



← → ↻ 🏠 🌐 www.facebook.com/zuck?sk=wall

facebook 🔍

Mark Zuckerberg

🏢 Works at Facebook 🎓 Studied Computer Science at Harvard University 🏠 Lives in Palo Alto, California 🗣️ Knows English, Mandarin Chinese 🏠 From Dobbs Ferry, New York 📅 Born on May 14, 1984

Wall

RECENT ACTIVITY

💬 "I like dangerous thoughts." on Samuel W. Lessin's status.

Mark Zuckerberg

Steve, you've done so much good for the world already. I hope you get better soon.

📱 January 17 at 11:43am via iPhone

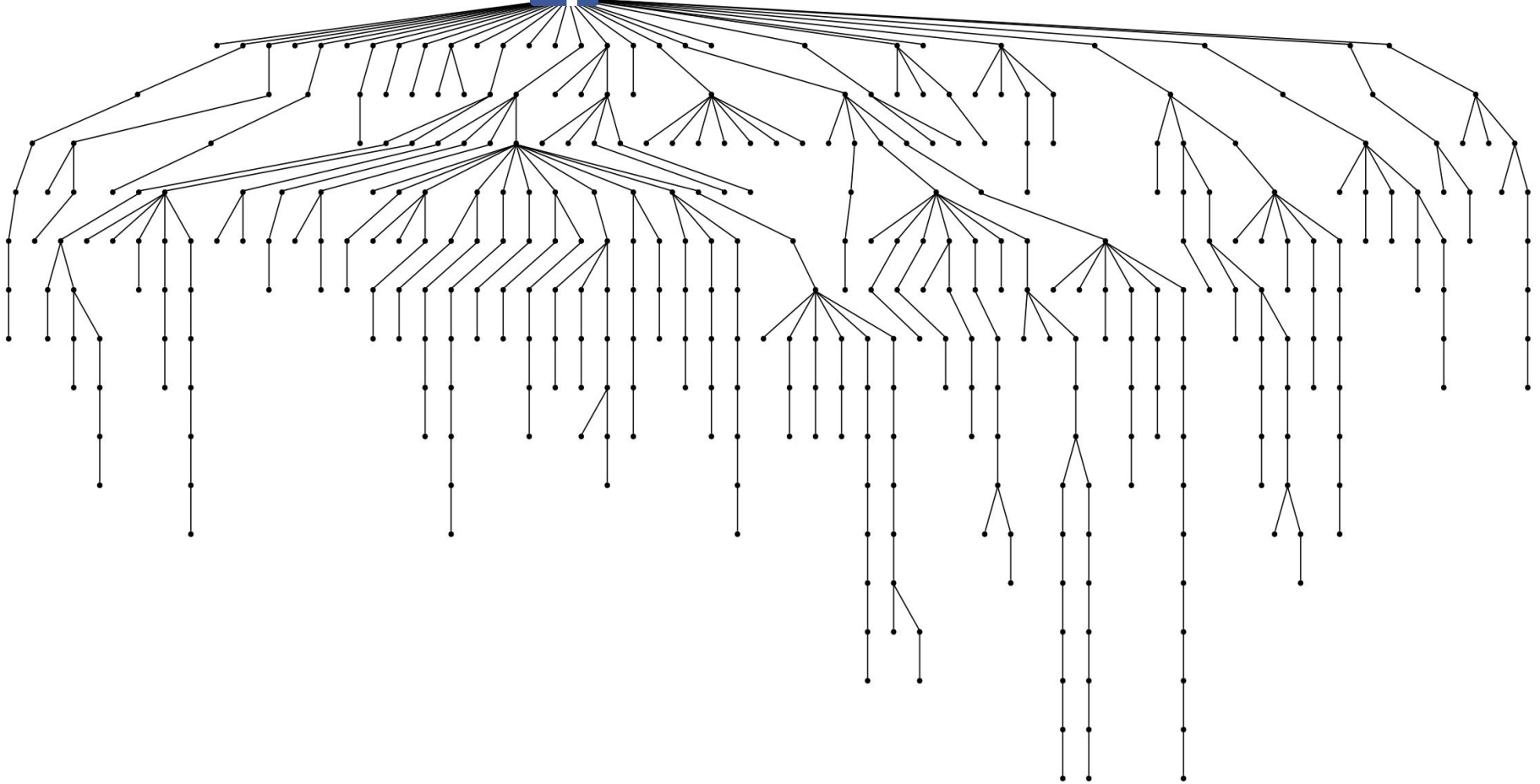
👍 150 people like this.

Wall
Info

Share Profile
Report/Block This Person

High Fanout in Data Fetching

A single  page, requires many data fetch operations



Nishtala, R., Fugal, H., Grimm, S., Kwiatkowski, M., Lee, H., Li, H. C., ... & Venkataramani, V. (2013, April). Scaling Memcache at Facebook. In *nsdi* (Vol. 13, pp. 385-398).

Graph Database Neo4j

- Designed to treat the relationships between data as equally important as the data
 - Relationships are very important in social graphs
- Property graph model
 - Nodes
 - Relationships
 - Properties
- Cypher query language
 - Declarative, SQL-inspired language for describing patterns in graphs visually

MongoDB

- Document Database
 - Schema-less model
- Highly Scalable
 - Automatically shards data among multiple servers
 - Does load-balancing
- Allows for Complex Queries
 - MapReduce style filter and aggregations
 - Geo-spatial queries

P3.2 - Overview

- Build a social network about Reddit comments
- Dataset generated from Reddit.com
 - **users.csv, links.csv, posts.json**
- Build a social network timeline on the Reddit.com data
 - **Task 1:** Basic login
 - **Task 2:** Social graph
 - **Task 3:** Rank user comments
 - **Task 4:** Timeline
- **Task 5: Selecting Databases**
 - Choosing the right database for a given scenario

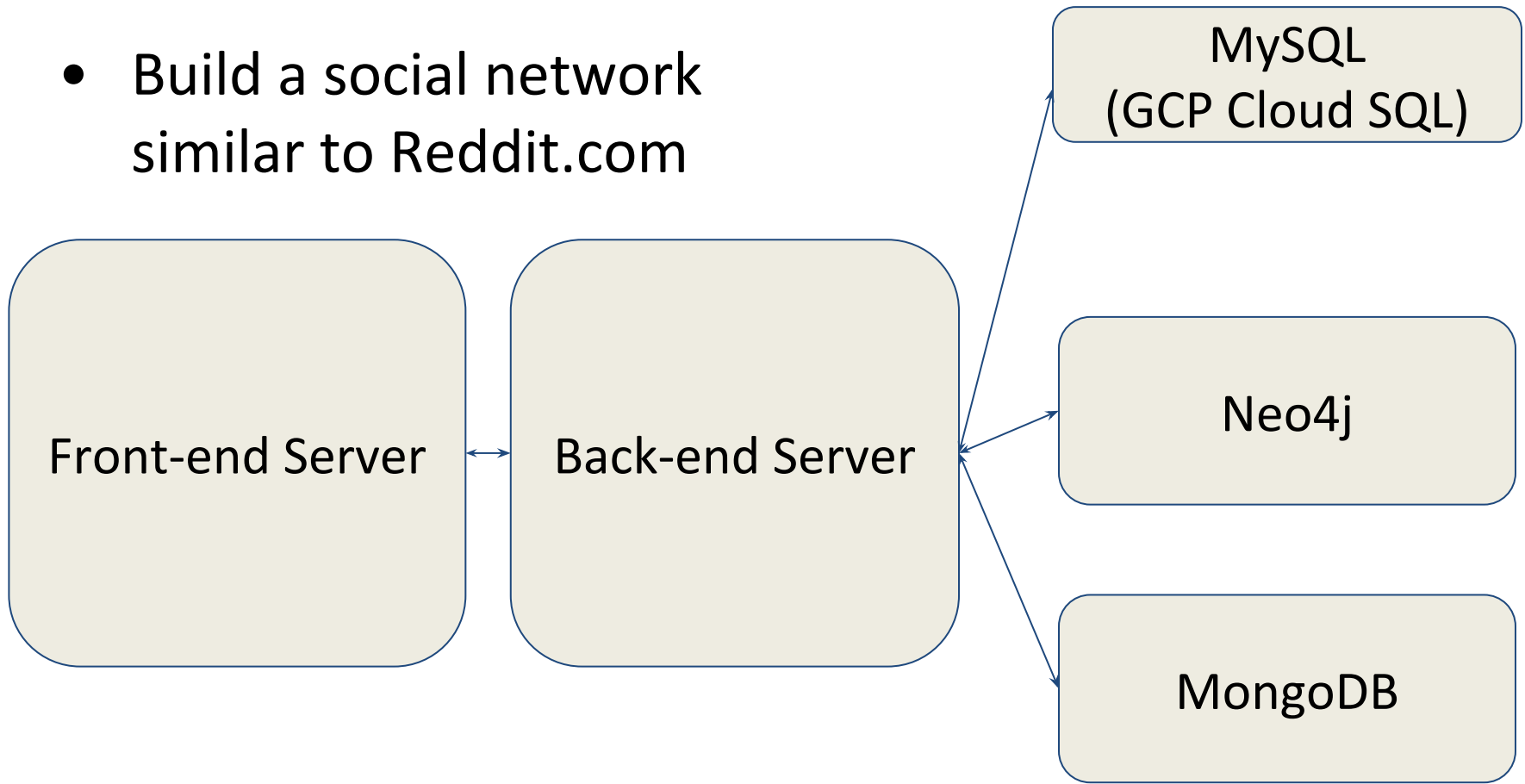
P3.2 - Reddit Dataset

- Task 1: User profiles
 - User authentication system : GCP Cloud SQL(**users.csv**)
 - User info / profile : GCP Cloud SQL
- Task 2: Social graph of the users
 - Follower, followee : Neo4j (**links.csv**)
- Task 3: User activity system
 - All user generated comments : MongoDB (**posts.json**)
- Task 4: User timeline
 - Put everything together



P3.2 - Architecture

- Build a social network similar to Reddit.com



Tasks, Datasets & Storage

Introduction

The Scenario: Build Your Own Social Network Website

Task 1: Implementing Basic Login with SQL

Task 2: Storing Social Graph using Neo4j

Task 3: Build Homepage using MongoDB

Task 4: Put Everything Together

Task 5: Choosing Databases

Dataset Name	Data Store Type
Login Information	RDBMS
Relation	Graph Database
Comments	Document Stores
Profile Images	S3

P3.2 - Task 5

- **Choosing Databases**

- Use your knowledge and experience gained working with the databases in the project to
 - Identify advantages and disadvantages of various DBs
 - Pick suitable DBs for particular application requirements
 - Provide reasons on why a certain DB is suitable under the given constraints
- Instructions provided in **runner.sh**
- Score will be shown **after** the deadline

Terraform

- Required in P3.2
- Required in the team project, get some practice
- Files provided
- Use **'terraform destroy'** to terminate resources
- Apply the tags for you
 - The tag is "3-2" instead of "3.2" (for GCP only)

P3.2 - Reminders and Suggestions

- Set up a budget alarm on GCP
 - Suggested budget: \$15
 - No penalties
- Learn and practice using a standard JSON Library. This will prove to be valuable in the Team Project
 - **Google GSON** - Recommended for Java
- Set up Gcloud in your environment
- No AWS instances on your individual AWS account are allowed
 - Otherwise you will receive warning emails and penalties

P3.2 - Reminders and Suggestions

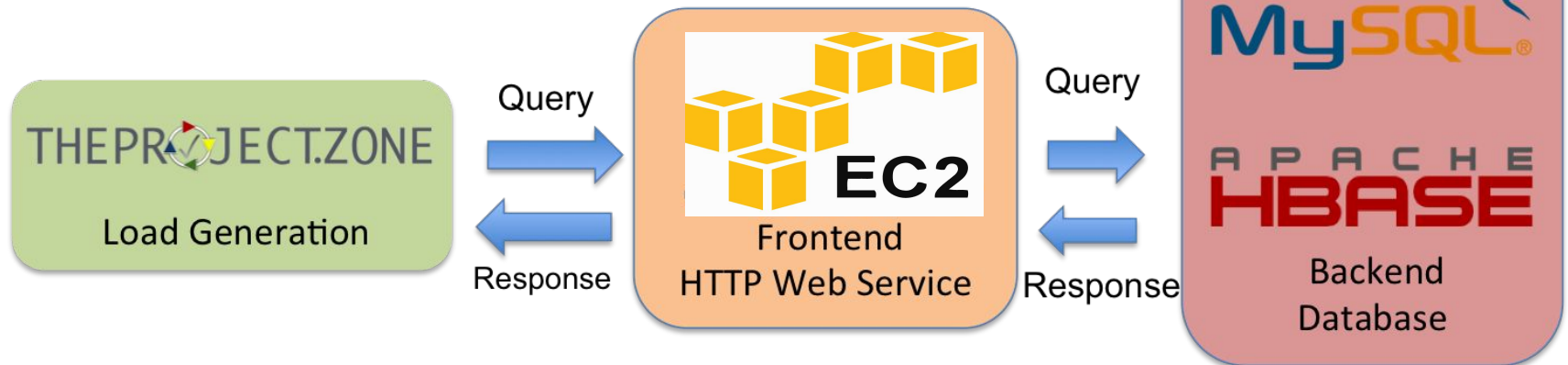
- In Task 4, you will use the databases from all previous tasks. Make sure to have **all** the databases loaded and ready when working on Task 4.
- You can submit one task at a time using the submitter. Remember to have your Back-end Server VM running when submitting.
- Make sure to terminate **all** resources using “terraform destroy” after the final submission. Double check on the GCP console that all resources are terminated.

A graphic featuring the white Twitter bird logo in flight, set against a dark blue background. The background is decorated with several light blue curved lines that suggest data trends or flight paths, and numerous small, faint upward-pointing arrows, symbolizing growth and positive sentiment.

Team Project

Twitter Analytics Web Service

- Given ~1TB of Twitter data
- Build a performant web service to analyze tweets
- Explore front end frameworks
- Explore and optimize storage systems



Team Project

- Phase 1:

- Q1
- Q2 (MySQL **AND** HBase)

Input your team
account ID and GitHub
username on TPZ



- Phase 2

- Q1
- Q2 & Q3 (MySQL **AND** HBase)

- Phase 3

- Q1
- Q2 & Q3 (MySQL **OR** HBase)

Team Project Deadlines

- Writeup and queries were released on Monday, **October 8th, 2018.**
- Phase 1 milestones:
 - Checkpoint 1:
 - Report, due on **Sunday, 10/14**
 - Checkpoint 2:
 - Q1 on scoreboard, due on **Sunday, 10/21**
 - Phase 1 Deadline:
 - Q2 on scoreboard, due on **Sunday, 10/28**
 - Phase 1, code and report:
 - due on **Tuesday, 10/30**

Git workflow

- Commit your code to the private repo we set up
 - Update your GitHub username in TPZ!
- Make changes on a new branch
 - Work on this branch, commit as you wish
 - Open a pull request to merge into the master branch
- Code review
 - Someone else needs to review and accept (or reject) your code changes
 - This process will allow you to capture bugs and remain informed on what others are doing

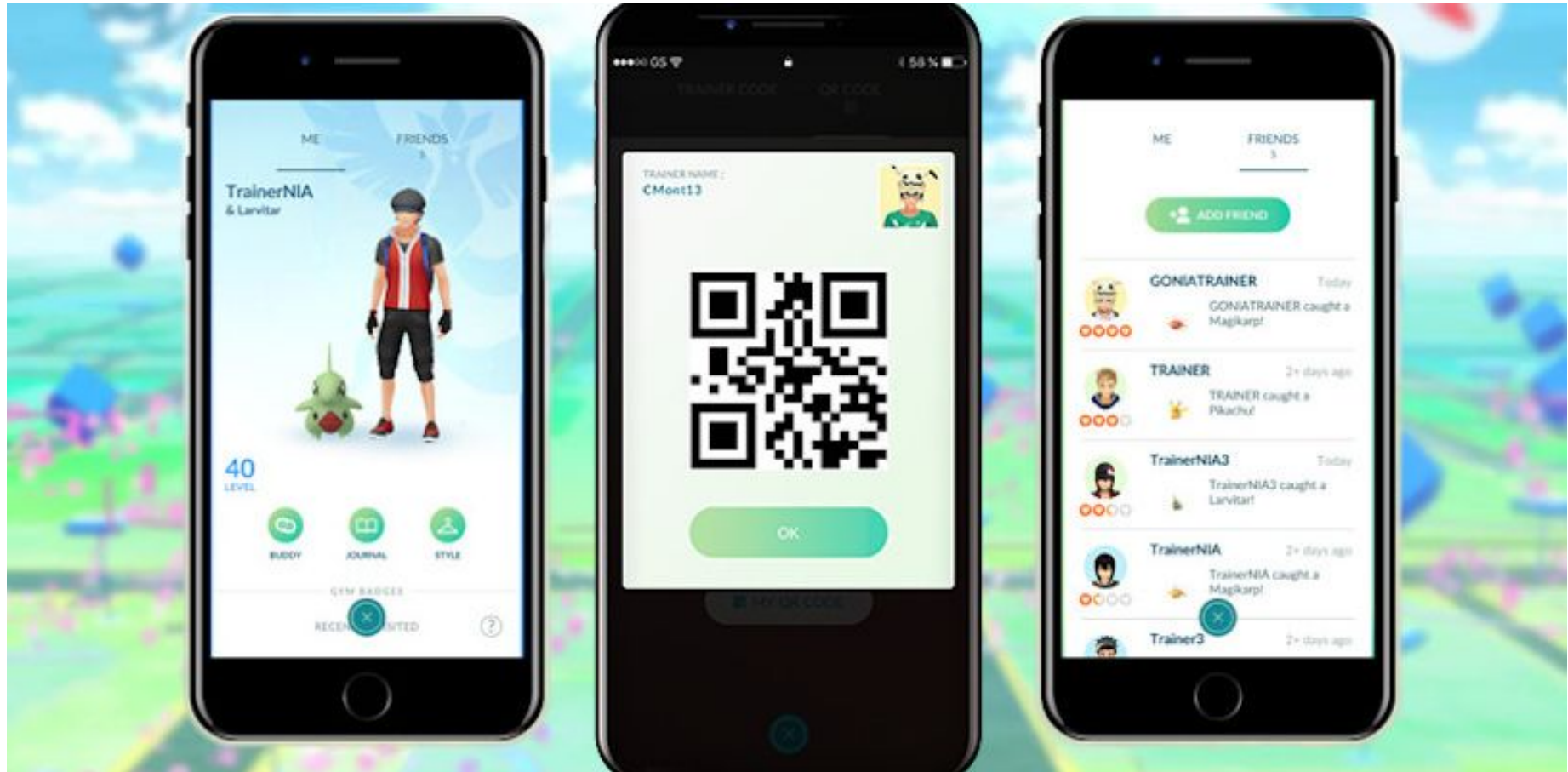
Query 1, QR code

- Query 1 does not require a database
 - Implement encoding and decoding of QR code
 - A simplified version of QR
 - You must explore different web frameworks
 - Get at least 2 different web frameworks working
 - Select the framework with the better performance
 - Provide evidence of your experimentation

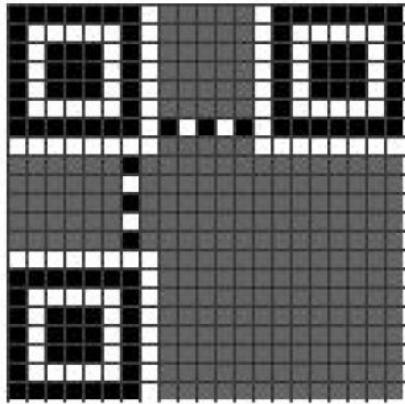
Authentication Service: QR Code



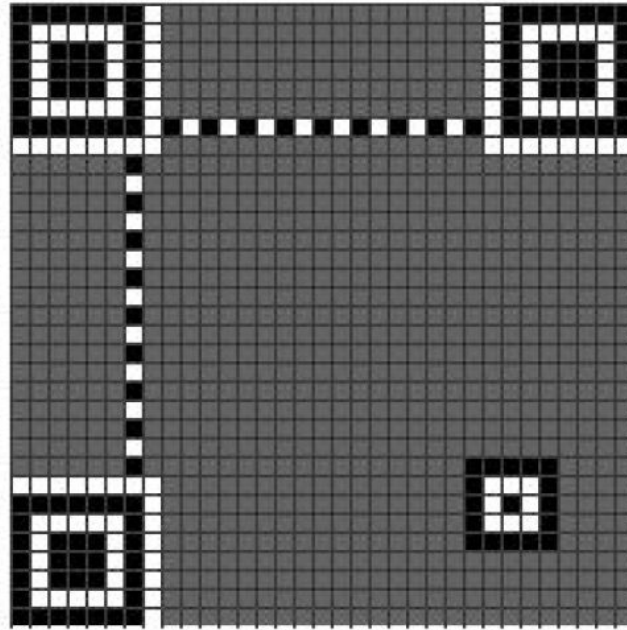
Authentication Service: QR Code



Authentication Service: QR Code

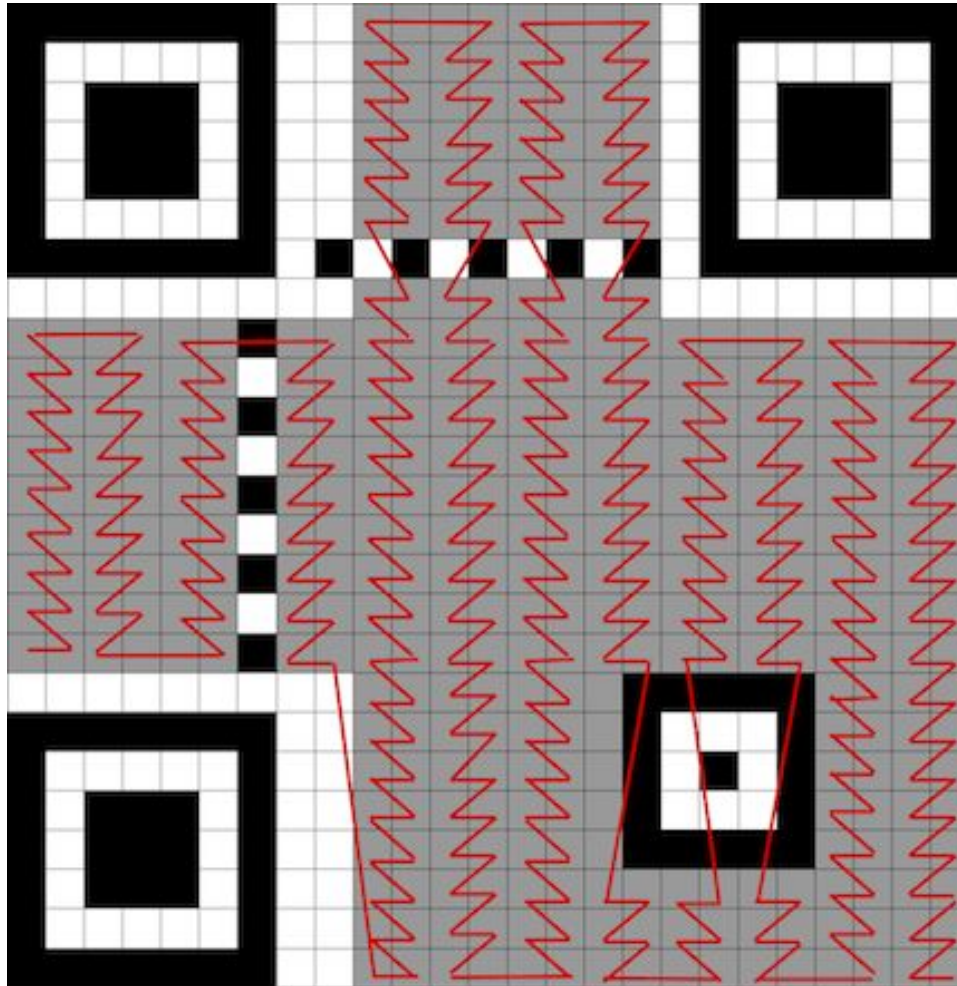


Version 1 (21 * 21)

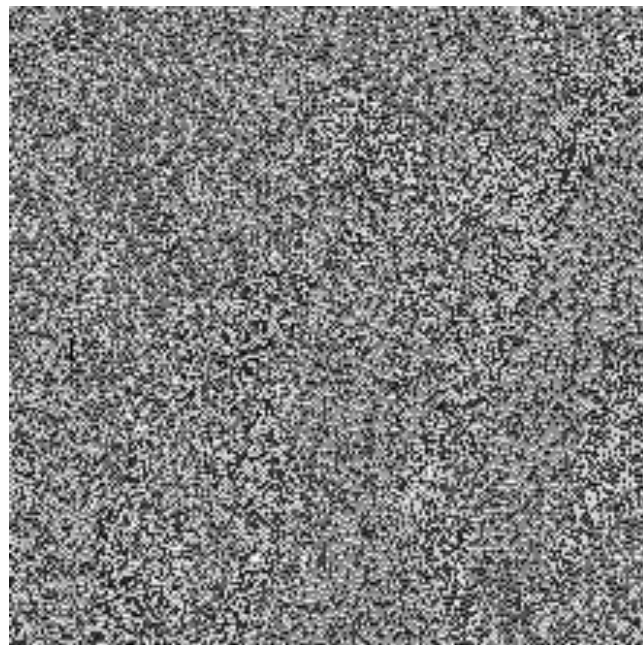


Version 2 (25 * 25)

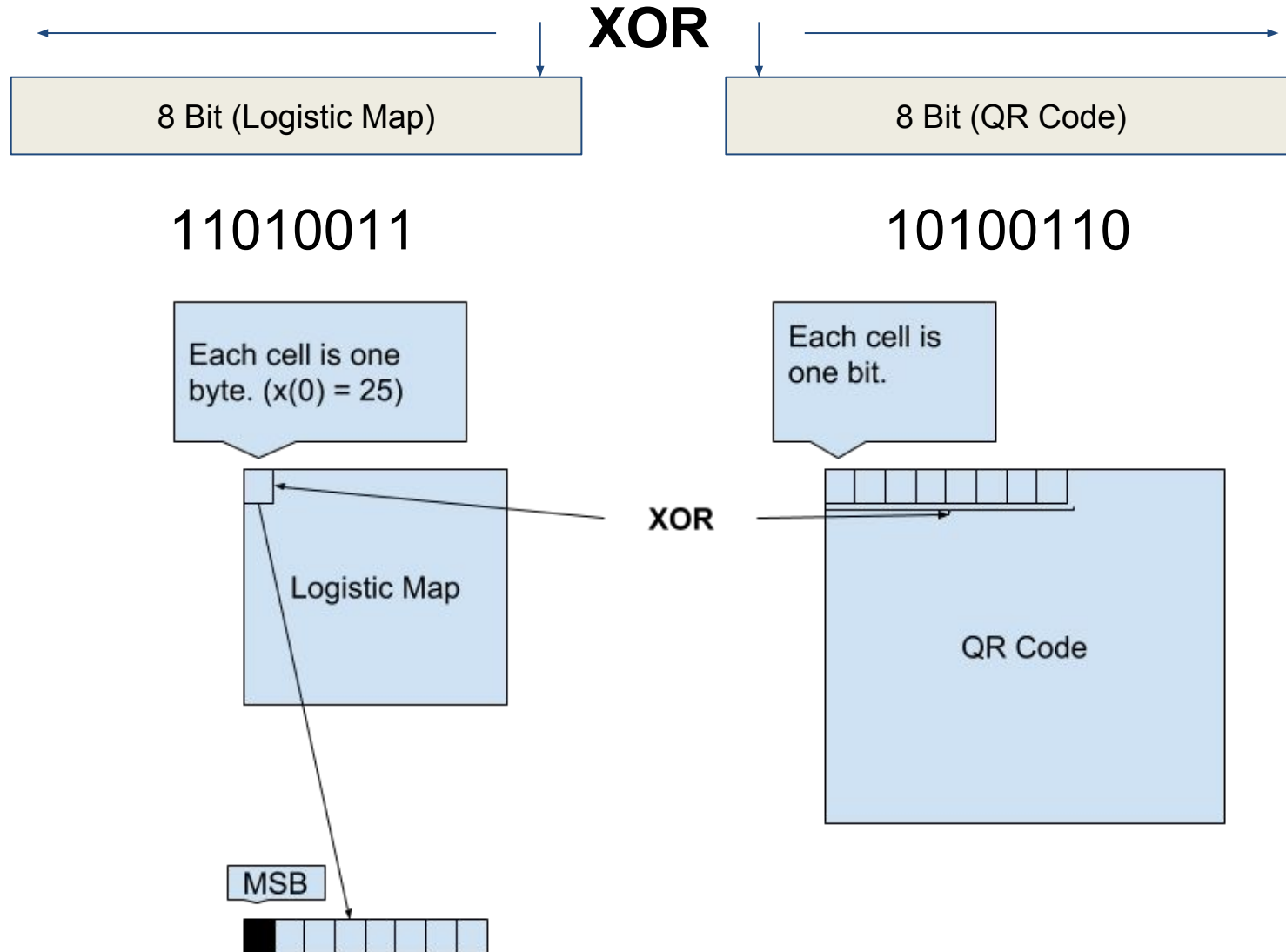
Authentication Service: QR Code



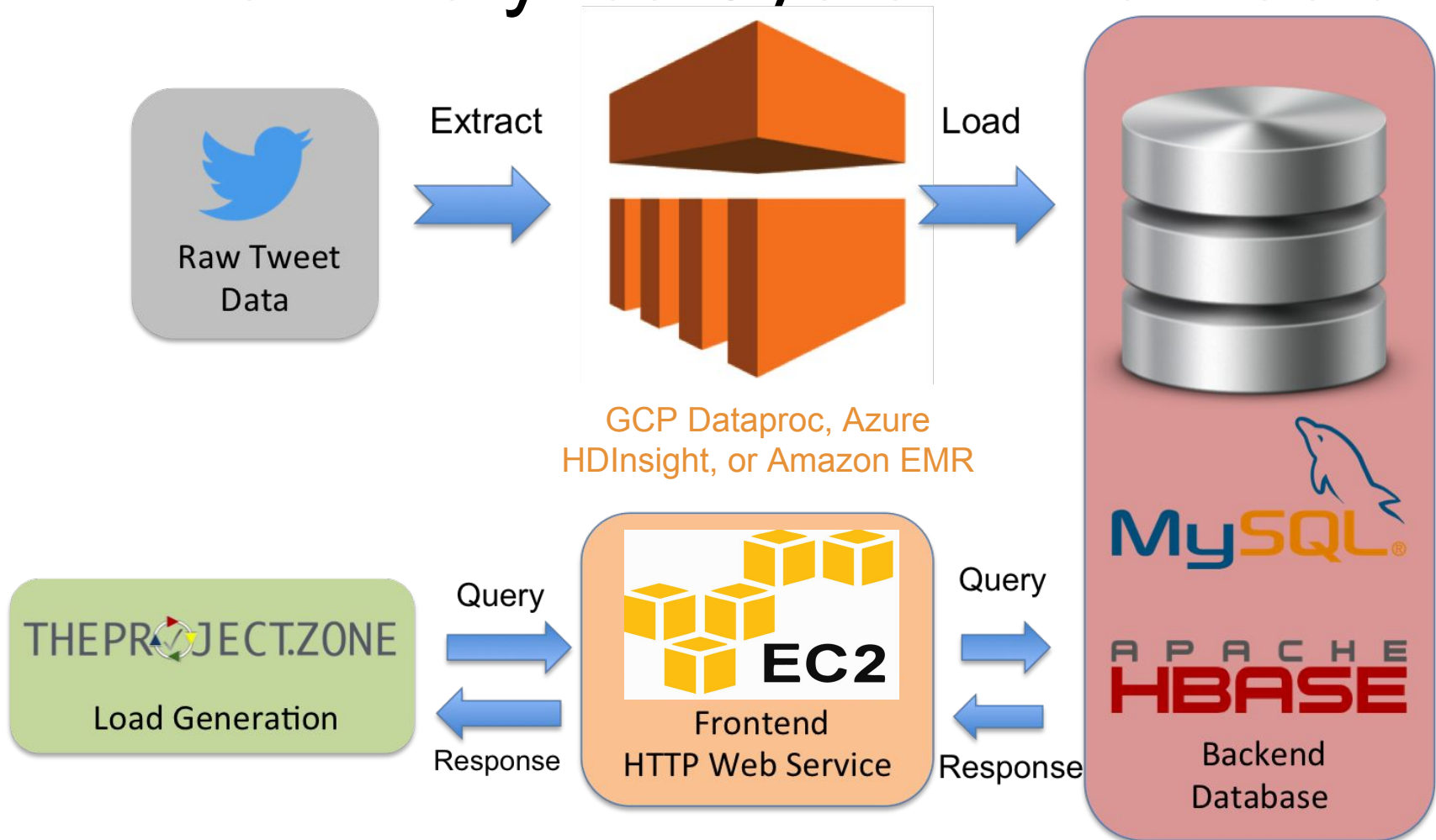
Authentication Service: Logistic Map



Authentication Service: Logistic Map



Twitter Analytics System Architecture



- Web server architectures
- Dealing with large scale real world tweet data
- HBase and MySQL optimization



Twitter Analytics System

Query 2:

Parameters: A User Id, a phrase and a limit 'n'

1. Find all the users who replied or retweeted the given user's tweet or who were replied or retweeted by the given user.
2. Sort them by **intimacy score * (match score + 1)**, break ties by user id.
3. Within the **interacted tweets** between two users, get the **latest tweet** which contains the given phrase. If no match, only get the latest one.
4. Return the **top n records** including the user name, user description and the tweet.

Query 2 Example

$\text{Intimacy_Score} = 1 + \log(1 + 2 * \text{reply_count} + \text{retweet_count})$

Example:

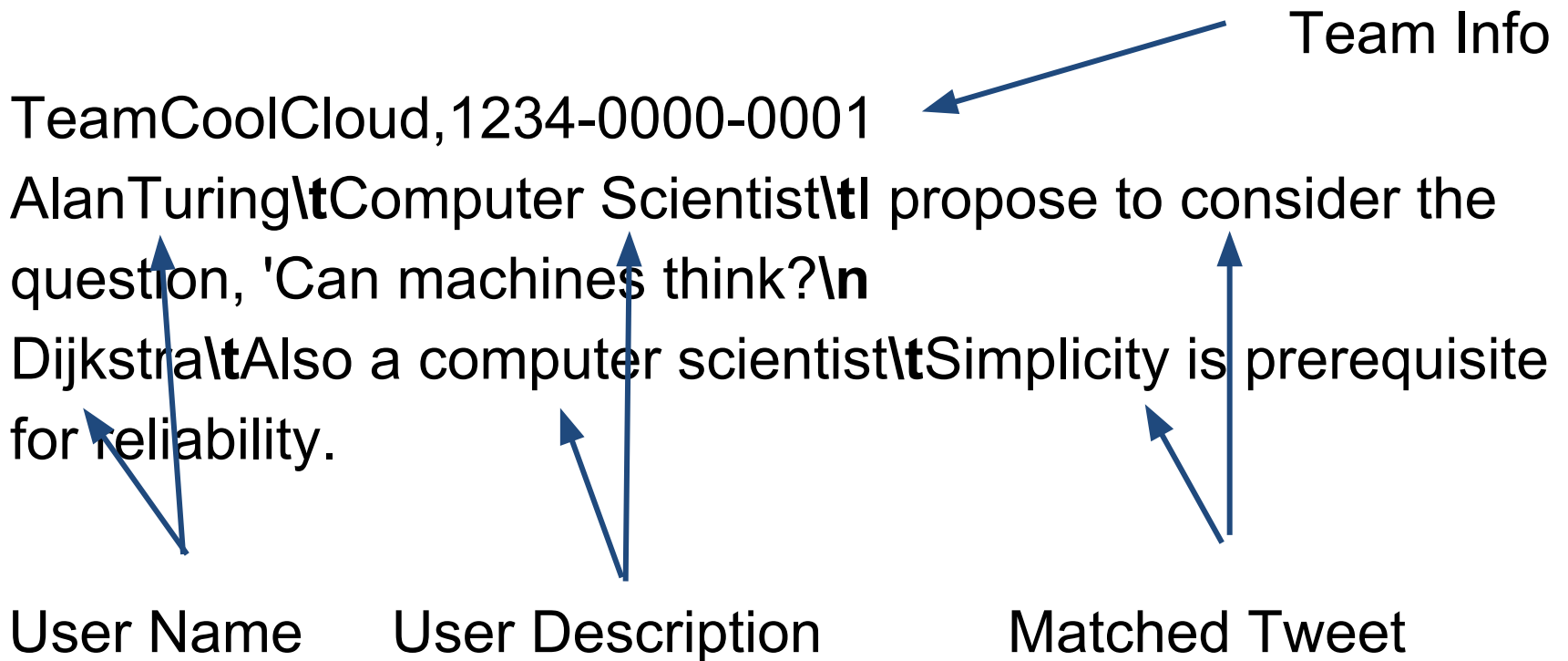
user A replied to user B 2 times, user B replied to user A 1 time
the score is $1 + \log(1 + 2 * (2 + 1))$

* The relationship is **mutual**.

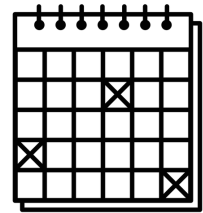
* We only care about these **interaction-tweets (contact tweets)**

Query 2 Example

GET/q2?user_id=12345&phrase=**cloudcomputing%20is%20cool**&n=5



Team Project Time Table



Phase (and query due)	Start	Deadline	Code and Report Due
Phase 1 Q1, Q2	Monday 10/08/2018 00:00:00 ET	Q1: Sunday 10/21/2018 23:59:59 ET Q2: Sunday 10/28/2018 23:59:59 ET	Tuesday 10/30/2018 23:59:59 ET
Phase 2 Q1, Q2, Q3	Monday 10/29/2018 00:00:00 ET	Sunday 11/11/2018 15:59:59 ET	
Phase 2 Live Test (Hbase AND MySQL) Q1, Q2, Q3	Sunday 11/11/2018 18:00:00 ET	Sunday 11/11/2018 23:59:59 ET	Tuesday 11/13/2018 23:59:59 ET
Phase 3 Q1, Q2, Q3	Monday 11/12/2018 00:00:00 ET	Sunday 12/02/2018 15:59:59 ET	
Phase 3 Live Test (Hbase OR MySQL) Q1, Q2, Q3	Sunday 12/02/2018 18:00:00 ET	Sunday 12/02/2018 23:59:59 ET	Tuesday 12/04/2018 23:59:59 ET



Note:

- There will be a report due at the end of each phase, where you are expected to discuss optimizations
- WARNING: Check your AWS instance limits on the new account (should be > 10 instances)**
- Query 1 is due on Oct 21, not Oct 28! We want you to start early!**

Upcoming Deadlines



- Conceptual Topics: OLI (Module 13)
Quiz 6 due: **Friday, 10/12/2018 11:59 PM Pittsburgh**
- P3.2: Social Networking Timeline with Heterogeneous Backends
Due: **Sunday, 10/14/2018 11:59 PM Pittsburgh**
- Team Project: Phase 1
 - Checkpoint 1, (This Sunday, Oct 14!)
Due: **10/14/2018 11:59 PM Pittsburgh**
 - Query 1, (Next Sunday, Oct 21!)
Due: **10/21/2018 11:59 PM Pittsburgh**

Q&A