C-DOG: Census Data Organization for the Government

6.1800 Preliminary Report

Aryan Kumar, Daniel Xu, Roderick Huang

March 24, 2023

1. INTRODUCTION

Prioritizing the representation of people's voices in government, the fictional country of Fictlandia aims to unify the census process and utilize technology to build a modern census system that will significantly improve the response rate. With the current system, census forms overlap causing disinterest in census participation. Through the availability of computer systems, we propose C-DOG (**C**ensus **D**ata **O**rganization for the **G**overnment), a multi-layered census system that aims to provide a simple census process for individuals, organized data frameworks for public services and elections, and long-term data storage for analysis.

The design of C-DOG impacts the government and the people, so its main focuses are *reliability* and *security*. Our primary objective is *reliability*, as data queried by users must be accurate and received in a timely manner even if failures occur. From national elections to local municipal public services, the data is essential for planning and decision-making purposes for the population. C-DOG not only organizes the data in a hierarchical tree structure but also continuously manages it to keep the data accurate and up-to-date. In addition, we prioritize *security*. While the census system collects massive amounts of information from individuals, it is imperative that people's privacy is protected. Specifically, through the analysis of census data, no individual can be identifiable. In C-DOG, security measures such as public encryption keys are utilized to ensure the confidentiality of the data.

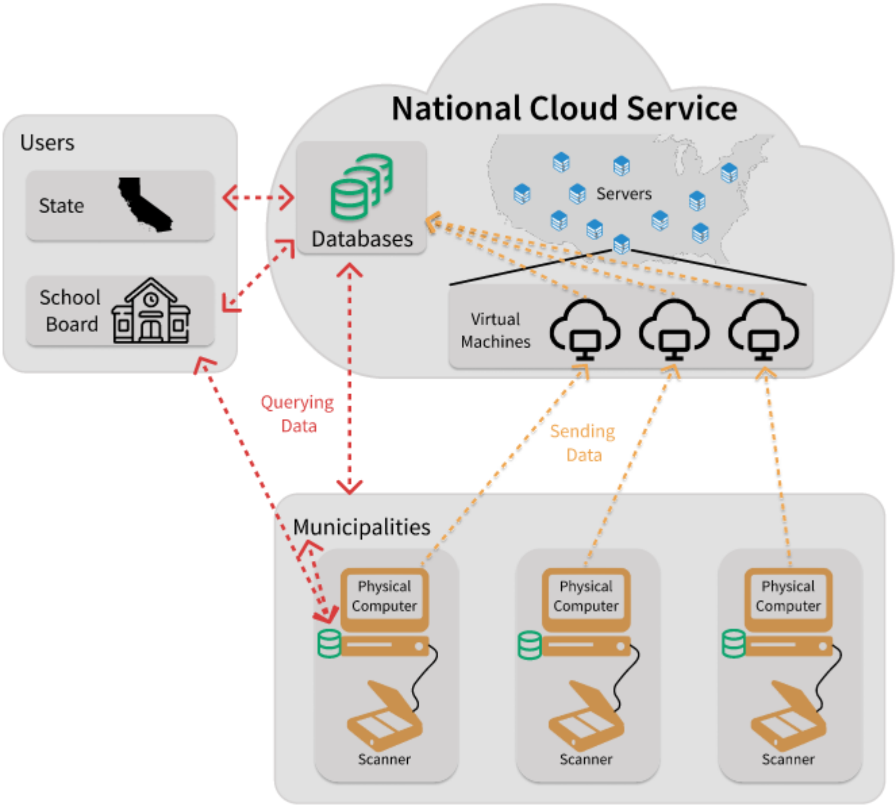
In Section 2, we will introduce the system overview of C-DOG at a high level. In Section 3, we expand into the specific design components highlighted in Section 2. In Section 4, we describe how the system will be utilized in certain use cases. Finally, in section 5, we will discuss the future implications of the system and possible aspects to consider.

2. SYSTEM OVERVIEW

C-DOG implements a centralized database containing all past census records along with provided APIs which are managed by the national servers. It consists of two main layers–the municipal layer and the national layer–along with multiple user classes as shown in Figure 1.

- *Municipal Layer*: Each census record is collected and processed at the municipal level before being sent to the national database for storage. Municipal servers cache a limited amount of data that can be accessed quickly by municipal users and the school board.
- *National Layer:* All data is sent to the national government's data centers, where the data will be aggregated at three varying granularities: municipal, state, and national. The data will be processed to form a hierarchical structure in the database, with important aggregate data at the top, which will improve access times of common requests.
- *User Classes:* Depending on the type of user, different ways of accessing the data are available due to data being stored locally and in the national server.

Figure 1: Illustrates the High-level Overview of C-DOG



3. SYSTEM DESIGN

3.1. Municipal Servers

The municipality and its servers form the lowest but most crucial level in our system. The municipal machines will be responsible for collecting and processing all local census records, and uploading them to the national government's data centers where they will reside for long-term storage and access by other users. They are also responsible for providing the municipalities with local census data.

3.1.1. Municipal Server Compute Allocation

Municipal machines have limited computation so we must devise a scheme to apportion its resources. We will use a tiered priority system to determine which tasks to attempt to perform immediately:

1. Online form submission
2. Queries for municipal census data
3. Uploading paper forms onto the local machines
4. Uploading census data into the national government's data centers and into the local database

We make this choice based on our system's emphasis on reliability. We prioritize being able to accept new census data and being able to handle requests for local census data efficiently above having a completely up-to-date national database. This is because data collection is the core component of the census, and we need to prioritize municipalities' access to the data since they provide services directly to the populous.

Based on this tiered system, we will handle computation as follows. During periods of greater load on municipal machines (from supporting online form submission or from being queried for data by the municipality), they will withhold sending any accumulated census forms to the national data centers or scanning any paper forms until the load dissipates. When the municipal machines are eventually under-utilized, particularly during night hours when no census data will be inputted, they will scan any accumulated paper forms and upload any outstanding census data to the national data centers and their local database.

Despite this priority system where we withhold performing some computation on our municipal machines during periods of greater load, our machines may still become overburdened. For example, during active census periods, municipal machines may become overburdened from processing online forms during peak periods (8-9 am and 7-9 pm); they may also become overburdened from handling requests for local data. Since both types of computation must be handled immediately, we will allocate the paired virtual machine (VM) to support the local machine during this time. The paired VM will support online form submission consistently during peak hours of the census period, and will support querying municipal data from the national database whenever the local machine is overburdened.

3.1.2. Municipal Server Data Upload to the National Servers

Since the municipal machines will be sending over individual census records, it is paramount to send them securely to protect each citizen's privacy. Thus, we will encrypt the files prior to sending them even though this comes at the cost of a larger file transmission, and send them using a TCP protocol. We make these choices to once again prioritize reliability and security, even if it may marginally diminish system efficiency, since we value data being sent completely, accurately, and securely. The municipal machine will send the census data to the paired virtual machine in the national data center, from where it will be uploaded to the national database.

The following API will be provided by the national government's data centers to the municipal machines to query and send data to the national servers.

Table 1: National Government Data Center API for Municipal Machines

Function	Name
`send_census_record`	Sends a census record to the national server
`send_paper_census_record`	Sends a scanned pdf of a census record to the national server
`request_census_record`	Requests a census record from the national server. Server verifies that record is a citizen of the requestor.

3.1.3. Municipal Servers: Data Storage and Access

The municipalities will store data on a shared local database to ensure data is coherent across multiple machines. Due to the limited storage for each municipality, the municipal machines will only store census records for the past five years on the database. The municipal machines will also cache the data on the database once the census is over to improve response times. Any corrupted or lost data on the local level must be recovered by querying the national servers.

The municipal government's first access point for data is the municipal machines. The municipal machines will provide data from their cache or by querying the local database as previously mentioned. If the municipal government queries data at a greater rate than the municipal machines alone can handle (e.g. during voting periods), the paired virtual machines will also respond to requests, providing census data from the national database.

The following API is provided by the municipal machines to the local government, so they may request various types of census data.

Table 2: Municipal Government Database API for Local Government

Function	Name
`request_individual_data`	Requests data of an individual residing in the municipality for a given year.
`request_municipal_data`	Requests census data for a specific municipality of the nation for a given year. Data provided is aggregate municipal level data

3.2. National Servers

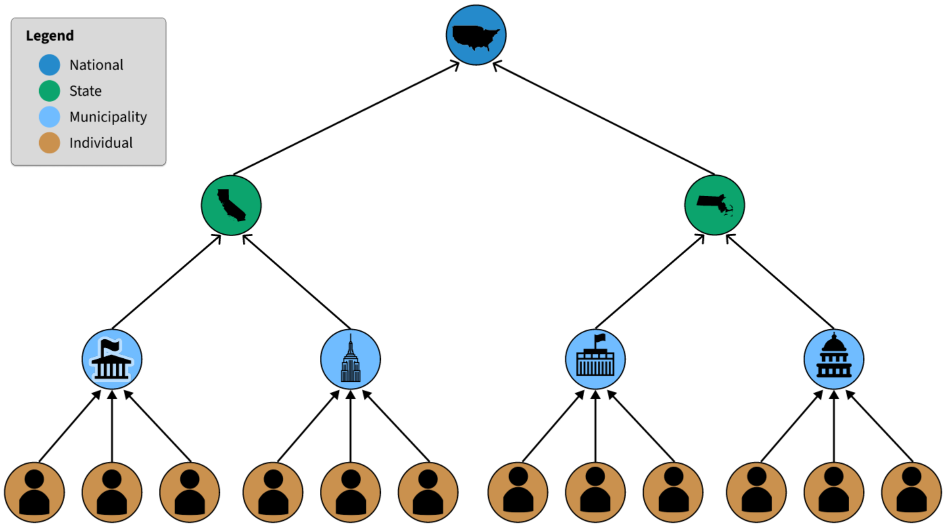
The national servers will provide for long-term storage of the data, and the provisioning of data to various levels of governments and users. Each level of government requires a different subset of the census record, and the national servers will be responsible for storing and serving all parts of the data and maintaining different access privileges for each different user.

The national server architecture consists of five computing clusters of 600 machines each that have an associated data center. 10 machines in each cluster are dedicated to being a proxy for the database. All requests to the database are routed through these machines which have read and write access to the database.

3.2.1. National Database Structure

To improve the efficiency of querying data, the national government's database will be organized in a hierarchical tree structure with each leaf node representing an individual data record, and each level above representing municipal data, state data, and national data. We will use the hierarchy concept to understand the database's organization, but note this is not the literal way data is organized in the database.

Figure 2: Conceptual Hierarchy of Database Organization



As we can see in the figure, each node represents data at a particular granularity (e.g. municipal, state, nation) and will forward this data to its parent node, which represents data of the parent geographical region. We organize the data in this manner to efficiently provide the various levels of government data they may need. Whenever data is needed at a particular granularity, it can be served to the requestor without the need to perform additional computation. Note, each municipal node will also store the census data of all school-aged children within its boundaries.

During periods when census data is actively being collected, local machines will be frequently sending data to the national government's cloud service. We want the census data in the cloud service to be up-to-date, but also not overburden the national data centers with excessive computation. Thus, we will take a meet-in-the-middle approach to this, where the data center's machines tasked with managing the database will periodically (every 30 minutes) process all the data contained in nodes at a given level, and use it to re-aggregate and update data in the parent nodes. During periods when the census is concluded, the data is stagnant, so no such updates will need to be performed.

3.2.2. National Database APIs

We will make the following API available to users (e.g. state governments, school boards), the national government, and researchers to access data from the national government's database.

Table 3: National Government Data Center API for the Government Levels, Researchers, and Users

Function	Name
`request_municipal_data`	Requests census data for a specific municipality of the nation for a given year. Data provided is aggregate municipal level data
`request_state_data`	Requests census data for a specific state of the nation for a given year. Data provided is aggregated state-level data

`request_national_data`	Requests census data for the entire nation for a given year. Data provided is aggregated national-level data
`request_school_children`	Requests data of all school-aged children within a specific municipality in a given year. Server verifies that the requester has credentials of a school board of the municipality for which the data is being requested.
`request_aggregate_data`	Requests an aggregate statistic (like average, standard deviation) about a specific demographic

3.2.3. Accessing Data from the National Database

As mentioned previously, the local government will access census data through the municipal machines. All other users of our system including the state, school boards, national government, and researchers will access data through the national government's cloud service. They will use the API in table 3 to make a request for data, which will be handled by the database proxy. The database proxy verifies that the request is from a valid host, and then obtains the requested data from the database and sends it back.

We implement a cache in the file system of all of the national data center's machines to respond to requests faster. The machine receiving the request will immediately return the data if it is cached, and otherwise obtain the requested data from the database and send it back like before. It will also cache the data in its file system in the latter case. If census data is actively being collected, we will set the cached file to expire in 1 hour; otherwise, the data will be cached for a month.

3.3. Reliability and Security

The five computing clusters ensure that requests to the national servers can be load balanced and rerouted given failures. This makes our design both extremely reliable and fault tolerant. The main security issue in this design is the protection of individual citizens' data from being accessed by unprivileged users. We make C-DOG secure by separating the servers that handle API requests and the machines that manage the database. The servers that handle API requests cannot issue direct requests to the database but must be routed through the database proxy machines. The database proxy machines manage the access privileges of each service and validate the identity of each user. Additionally, we make the design more modular by running each separate service on a separate machine. Thus, a compromised service that can access aggregate data cannot make unprivileged access to the database or compromise a service that has access to individual census records. A municipality cannot request a record from a citizen outside of its borders.

As an additional layer of security, the database proxy servers encrypt and decrypt all census records when retrieving and writing to the database. For the sake of storage efficiency, we do not store encrypted records in the data centers, but we make sure to send encrypted copies of the census records to and from the user. Transmission of encrypted data requires a signature from both the requester and the sender.

3.4. Failure Cases

There are three main failure cases that C-DOG must handle: computer crash, computer failure, and network failure. In the case of a computer crash, it takes 5 minutes to reboot. The virtual machine can take over the workload during that time period. In the case of a computer failure, document scanning will not be possible but online submissions can be routed to the virtual machine. The backlog of document scanning can be made up during the system's off-hours. In the case of network failures, the municipal machines can store the documents and records and upload them when the network recovers. Note that it is unlikely that computer and network failures happen at the same time.

4. USE CASES

The modular separation between services and data allows for easy and secure implementation of new services to users. The design of C-DOG supports a wide variety of use cases which are detailed below.

- *National Government (Redistricting)*: In order to properly redistrict, the government needs to query the population of each state. C-DOG's hierarchical data structure (3.2.1) allows this data to be automatically computed during the census uploading process. Thus, querying the state population can be done in one simple API call rather than retrieving all census records. This use case is not affected by the failure of the municipal machines.
- *Municipal Governments (Elections)*: In order to run the election, the municipal government must have access to the most recent years' data. The design of C-DOG ensures that this data is available on the municipal database and can be accessed without needing to query the national databases (3.1). This allows our implementation to bypass network failures, and improves response times. In the case of computer failures, the municipal government can query the national database using C-DOG's API system.
- *School Board (Student Identification and School Assignment)*: The school boards must be able to query census records for children in their school district. Similar to the elections use case, school boards can bypass querying the national servers by directly accessing the data in the municipal servers. In the case of a computer failure, the school board can send requests to the national servers. Both servers encrypt the data prior to sending it to the school boards to protect the information of school-children.
- *External Researchers:* Researchers can send requests for aggregate data such as average age, household size, etc. to the national servers. C-DOG's hierarchical organization of the data (3.2.1) ensures that this data is precomputed and can be retrieved very quickly without needing to iterate through all relevant census records.

5. CONCLUSION

The C-DOG system is designed to provide a centralized, reliable, and secure way for census collection. The impact of this work includes all citizens who can now submit a single census record every year to the centralized system. At all stages, we have prioritized the reliability of the system, ensuring the data we store is accurate and our census system is available for a maximal amount of time. This ensures most people can fill out the census forms on their first attempt, increasing census participation, and that our system is performant and available for data querying in periods that can be critical (e.g. elections). Our system's accuracy enables accurate forecasting of public services demand by the various levels of government.

In addition, we have ensured the security of private data through a modular design and encryption of individual census records. Given that the data collected by the census system is sensitive, it's crucial to build trust in the system. While maintaining accuracy in the data is important, public confidence in the system largely affects the important decisions made in public policy.

Though we have attempted to make the C-DOG census system as robust as possible, there are a few things left to address for its next iteration. C-DOG is intended to last for the foreseeable future, though it currently does not take into account storage constraints. It simply stores all individual records into the national database, but a more efficient scheme may be needed as the size of the data grows. In particular, C-DOG under-utilizes the file system of the datacenter machines, so this may prove helpful. In addition, C-DOG provides a high-level overview of the database organization but it leaves out important lower-level details; it also must provide further details about failure handling and recovery.