CSOUP: A System for Census Data Storage, Organization, Usage, and Protection

6.033 Preliminary Report

Vincent Lin, Christopher Wang, Selena Zhang

March 25, 2023

1 Introduction

Fictlandia is modernizing its census system to encourage participation by ensuring each Fictlandian only needs to fill out the census once. To facilitate streamlined censusing, we propose a system for **C**ensus Data **S**torage, **O**rganization, **U**sage, and **P**rotection (CSOUP), which (1) enables reliable and secure data collection during the census period, (2) transfers complete and correct data to the relevant parties, and (3) stores census data long-term.

Our primary design priority is *reliability*, the ability of the system to continue receiving, storing, and distributing correct and complete census data even in the face of unexpected computer or network failures. Loss of census data at any point could prevent hundreds of thousands of people's data from being considered in redistricting, elections, and distribution of public services. We prioritize reliability to ensure that the government can uphold Fictlandians' rights, such as their ability to vote and their access to healthcare and education.

A secondary design priority is *security*, the ability of the system to protect against potential data exposure and deanonymization. Security is critical to respecting the privacy of Fictlandians and retaining their trust in the governments. Since census records contain sensitive address and demographic information, accidental leaks or malicious attacks could expose the locations of vulnerable populations and marginalized groups, including elderly persons living alone, minors, same-sex couples, and racial minorities, who would be disproportionately at risk of physical backlash or discrimination.

Our system achieves *reliability* by transferring data from municipalities' physical machines to the national cloud's virtual machines, which allow for more reliable distribution of data to users. Our system further improves *reliability* by using careful network protocols that ensure correct and complete data transfer. Finally, our system achieves *security* by consistently encrypting data in transit and in storage, at the cost of storage capacity and speed of data distribution.

CSOUP System Diagram

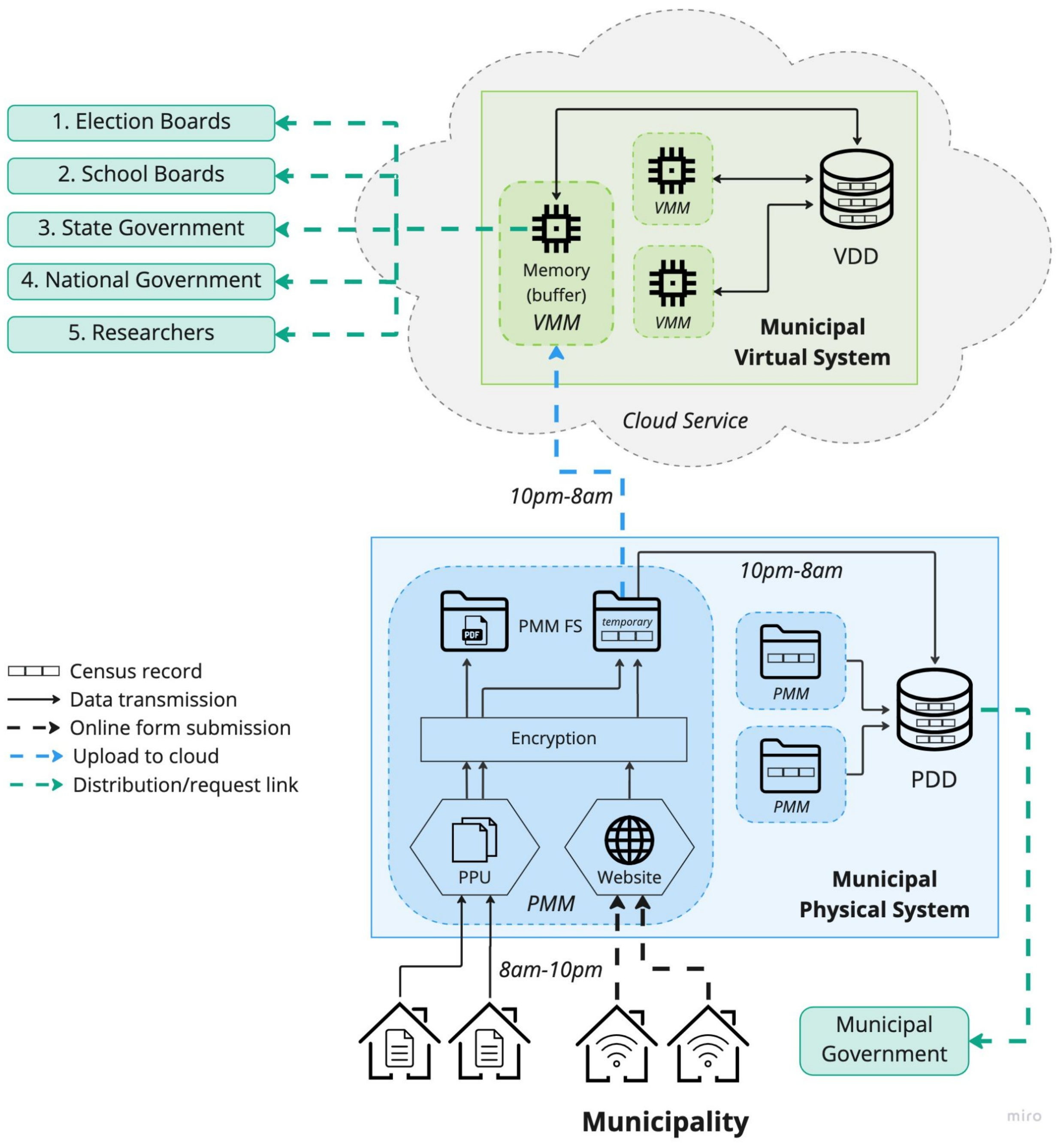

Fig. 1. A diagram of CSOUP's components for a single municipality, outlining its two main layers (PMMs and VMMs), transport and storage of data among layers, and distribution of data to users. Bold, dashed lines indicate data transmission across the network, requiring reliable communication protocols as discussed in following sections. (PDD = physical distributed database, VDD = virtual distributed database, PPU = paper processing unit, FS = file system)

2 System Description

CSOUP is modularized into two main layers, as depicted in Figure 1:

1) *Physical municipal machines (PMMs)*: At the bottom layer, PMMs receive, encrypt and store submissions and send them to virtual municipal machines, discussed further in section 2.1. PMMs within a municipality share a physical distributed database (PDD).
2) *Virtual municipal machines (VMMs)*: At the top layer, each PMM is paired with a corresponding VMM hosted by the national cloud service, discussed further in section 2.2. All VMMs for a particular municipality share a virtual distributed database (VDD).

2.1 Physical Municipal Machines (PMMs)

The PMMs are the first level of the system to receive and handle census data from residents. They provide four key pieces of functionality, as shown in Fig. 1:

1. PMMs host the web server serving the online census form, and receive census submissions through both the online form and scanned paper forms.
2. PMMs encrypt all data upon receipt.
3. PMMs store census records in the PDD (where they can be accessed by municipal governments) and send records to the VMMs.
4. PMMs provide long-term storage for PDFs.

Below, we describe how our system provides these functionalities while prioritizing *reliability* and *security*.

2.1.1 Managing the Web Server and Receiving Submissions

Each PMM can only support 125 users submitting the online form in parallel. If too many users submit at once, the PMM may drop some submissions, resulting in Fictlandians' data being lost. To avoid this issue and provide *reliability* at the input level, the PMMs in our system monitor and limit traffic. Each PMM uses a modified version of Explicit Congestion Notification (ECN) with website monitoring. If the number of active clients exceeds some preset threshold, then the system will notify, uniformly at random, a number of clients (in proportion to the excess) that they cannot submit their form at this time. However, their responses will be saved temporarily on their local machines so that they can submit later. By preventing residents from submitting forms during high traffic, the system proactively reduces the risk of data loss.

Our system also protects against unnoticed data loss with the following network protocol: When a resident submits, the PMM, upon receiving the submission, will send back a matching "ack" packet to the web server. Once the web server receives the "ack," it can inform the resident that their submission has been received. Otherwise, if the web server times out waiting for the "ack," it will inform the user that their submission failed and they should resubmit. (If an "ack" gets dropped, the web server might resend data that the PMM already received. To handle this, the PMM can keep track of recently received submissions and discard any duplicates.)

2.1.2 Encrypting Submissions

Each PMM encrypts all data upon receipt, including records and PDFs from scanned forms. We encrypt before the submission is moved to permanent storage to uphold *security*. This ensures that, if data from the PMM is leaked, Fictlandians' privacy will still be maintained. Although this comes at a tradeoff of storage and speed, this does not impair our system's ability to meet the requirements, as shown ahead (the exception being PDF storage, discussed in 2.1.5).

2.1.3 Writing Records to Disk and to VMMs

To store census records in an organized, consolidated location, the PMMs insert the encrypted records into the PDD on disk. However, since inserting into a database takes 300 ms for encrypted records, a PMM crash could result in records being lost after they are received but before they are written to disk. To avoid this, PMMs temporarily save each census record to the file system immediately after encryption. Writing to the file system is limited by disk write speed (4,100 MBps), which can write a 2,400-byte encrypted record in 0.6 µs. Since this is much faster than writing to a database, it is much less likely for data to be lost in a crash before being saved to disk, increasing *reliability* in the face of potential failures.

For each census record in the file system, the PMM sends a copy of the record to the VMM (communication protocol discussed in 2.2.1) and inserts it into the PDD on disk. Afterward, the PMM deletes the record from the file system to free up space. To avoid compromising the PMM's ability to process incoming submissions, the PMM only writes to the PDD and the VMMs from 10pm to 8am, when there are few or no form submissions. Since it takes 300 ms to insert encrypted records into the PDD, this 10-hour window is sufficient to insert 10 h * 3600 s/h * (1 record/0.3 s) = 120,000 records, far more than a single PMM is likely to receive in a day.

After the data collection period, our system provides an interface that municipal governments can use to query data from the PDD. Since the PMMs belong to the municipality, it is most resource-efficient to let municipal governments access the data in PMMs directly. Unlike the system's other users, municipal governments are permitted to access the entire set of fields for each person.

2.1.4 Storing PDFs

Scanned PDFs are saved to the PMM file system on disk and stored there long-term. Each encrypted PDF is named with a large, randomly generated ID number to provide *security* and anonymity; even if the file is leaked, no individually identifying information is revealed. We organize the PDFs in the PMM file system by year, but not by name, address, or any other information that could be used to identify the individual. The PDF's ID number is appended to the corresponding (encrypted) record so that, when the data is decrypted and de-anonymized in 70 years, the system can match records to PDFs and de-anonymize the PDF filenames.

Unlike records, the system does not send copies of the PDFs to VMMs because the PDFs are much larger and are required by users much less frequently, reducing the need for consistently available access. Sending PDFs to VMMs would incur costs to both *reliability* and *security* by increasing the risk of data loss or breach over the network, with minimal benefit.

2.1.5 Storage Evaluation

The government requires that data is stored in perpetuity. Although our system may not store everything forever due to data accumulating each year, we can estimate whether our system's storage lifetime is reasonable. The size of each encrypted record is:

200 words * (4 bytes/word) * 3 (encryption) = 2,400 bytes

With one PMM per 100,000 residents, the total size of all records per PMM each year is:

(100,000 records/yr) * (2,400 bytes/record) * (1 MB/10^6 bytes) = 240 MB

The government also requires that we store PDFs. Assuming that 20% of submissions are paper forms, the average PDF size is 3 MB per household, and there are 2.6 people per household on average, the estimated size of all PDFs per PMM each year is:

(100,000 ppl/PMM) * 20% * (1 household/2.6 ppl) = 7,692 households
7,692 households * (3 MB/household) * 3 (encryption) = 69,230 MB

Then, the estimated time it would take for the records and PDFs to fill up each PMM's 1 TB disk (our system's estimated storage lifetime) is:

1 TB * (10^6 MB/TB) * 1 yr/(240 + 69,230 MB) = **14.4 years**

Since national censuses occur every 10 years and data is released after 70 years, our system's lifetime should be larger than the order of tens of years. However, our current resources are insufficient to store PDFs across that timeframe. Our system's lifetime is bottlenecked by PDFs, which occupy nearly 300 times more storage per year than records. Even if we distributed PDF storage across both PMMs and VMMs (doubling capacity) and stored data unencrypted (tripling capacity), our system's lifetime would only be 14.4 * 6 = 58 years, still not enough to reach the 70-year mark. Moreover, storing unencrypted data threatens the *security* of our system, since residents' confidentiality could be compromised by a data breach. Thus, we request that the national government provide additional resources for us to satisfy the PDF storage requirement.

2.2 Virtual Municipal Machines (VMMs)

A reliability concern with the PMMs is that they may crash or suffer catastrophic failures, or the municipal networks may suffer outages, compromising the PMMs' ability to provide reliable access to census data. This is especially true in municipalities with only one PMM, in which the PMM becomes a single point of failure.

Consequently, we use the VMMs provided by the national cloud service to (1) store census data long-term in virtual distributed databases (VDDs) and (2) distribute the data to users, as shown in Fig. 1. Since we take the national cloud service to be highly redundant and distributed, we assume that the VMMs rarely or never fail, providing more *reliable* access to data than PMMs.

2.2.1 Communication with the PMM

From 10pm to 8am, each VMM receives census records and PDFs from a PMM, as discussed in section 2.1.3. However, data sent by the PMM may fail to reach the VMM due to network outages or packet drops. To ensure the VMM receives complete data, we employ the same protocol described at the end of 2.1.1 with "ack" packets, ensuring that the VMMs will successfully receive all records and insert them into the VDD.

Since the bandwidth into the cloud service is high (10 GBps), the VMM is bottlenecked by database insertion time (300 ms for encrypted records). Thus, the PMM sends records no faster than one per 300 ms so that the VMM has enough time to store each record before receiving the next one, ensuring that data is not dropped.

2.2.2 Storage

Within each municipality's VDD, each year's data is stored in a separate table. This allows VMMs to distribute the data for the current year without iterating over data from previous years, and it also allows researchers to query data from particular ranges of years (discussed in section 3.4).

2.2.3 Distribution Functions

To meet the needs of users, VMMs distribute the necessary data to them: specifically, the school and election boards and the state and national governments. One challenge is *reliably* distributing the data to all users within the one-month timeframe (March 1 - April 1) without data loss, even if network failures occur. This is especially true for the largest municipalities, which can have up to 1,500,000 records per year.

To meet this requirement, we delegate up to 100,000 records from the VDD to each VMM for distribution, since each municipality has one VMM per 100,000 residents. This parallelization allows the rate of distribution to scale with municipality size. Since we assume the cloud service is reliable, we can parallelize completely without worrying about individual VMMs failing.

The VMMs send data to the election boards first, then to the school boards, then the state governments (every five years), and lastly the national governments (every ten years), as shown in Fig. 1. We prioritize the local boards first because they may require the most time-sensitive access to the data, primarily the election boards (discussed in section 3.2).

For each of those four sets of users, the VMM accesses each census record from the VDD, decrypts it, extracts the subset of record fields required by that user, encrypts it with a protocol shared with the user, and sends it across the network using a *reliable* network protocol as discussed in section 2.1.1. Afterward, it erases the decrypted record from memory to uphold *security*. Although decrypting the record in the first place produces a brief security risk, the VMM must do so to extract only the information needed by that particular user. We believe it would be an even larger security risk to provide users with information they need not (or *should* not) know, which could be leaked or used maliciously.

We can estimate whether this strategy satisfies our required timeline: Since the VMM bandwidth and read/write rates are high, the distribution pipeline is bottlenecked by the 210 ms required to access encrypted records from the VDD. The estimated time to access 100,000 records is:

$$(210 \text{ ms/record}) * (100{,}000 \text{ records}) * (1 \text{ s}/1000 \text{ ms}) * (1 \text{ h}/3600 \text{ s}) = 5.83 \text{ h}.$$

Thus, it takes about 4 * 5.83 h = 23.3 h, less than a day in total, for each VMM to access and distribute the data to all four sets of users. This is well within the one-month timeframe, with plenty of buffer time to account for unexpected failures like multi-day network outages.

3 System Impact and Use Cases

Governments and school and election boards require census data at regular intervals to run services and make decisions. Additionally, external researchers may also require census data for studies. Below, we describe the impact of each use case and how our system addresses them.

3.1 National Redistricting

The national government requires the national census data by April 1, a timeline that our system easily meets even if network failures occur, as discussed in section 2.2.3. The national government uses the data to determine the number of representatives per state during redistricting. Redistricting plays a key role in national legislature election results, which determine who designs future legislation. Thus, this process impacts all residents across the country, and it is critical that our system provides complete, correct data so that representatives are allocated fairly. This responsibility falls especially to us because the national government is unlikely to notice any instances of data loss, given the high volume of data they are handling. Our system's focus on *reliably* transporting and storing data achieves this goal even if crashes or network failures occur, as discussed throughout section 2.

3.2 Municipal Election Boards

Municipal election boards use census data to determine voter rolls and mail out ballots. This use case directly impacts Fictlandians' right to vote, so it is essential for our system to provide accurate data with *no* data loss so that Fictlandians are guaranteed this right. As mentioned, our system achieves this goal through its focus on *reliable* data transport and storage.

Municipal elections also present the most time-sensitive use case for our system, particularly if updated voter data is needed for a special election shortly after the data collection period. To accommodate this, we send census data to the election boards before other users, which can occur within a single night after the census period, as discussed in 2.2.3.

3.3 School Boards and Student Assignments

In May, school boards use census data to identify school-age children, assign them to schools, and provide language accommodations. School boards are among the system's most data-sensitive users, as the loss of a single child's record could deprive them of receiving appropriate education. This use case, like the others, aligns with our system's focus on *reliability* and protection against data loss. Additionally, our system sends the necessary data to school boards well before May (section 2.2.3), giving them ample time to identify and assign students before the next school year.

3.4 External Researchers

External researchers may request past census data for studies. These studies, particularly those on social or economic issues, can benefit Fictlandians by positively informing policy and government decisions. To serve these requests, our system provides an interface for researchers to query the VMMs for each year's aggregate anonymized data within the last 70 years, or complete de-anonymized data from before then; researchers conducting nationwide studies can query VMMs from all municipalities in parallel. The organization of records by year in VDDs (section 2.2.2) allows VMMs to easily serve requests for particular time ranges.

Since the cloud service is fault-tolerant, the VMMs provide researchers with consistent, reliable access to the data. However, if the census is ongoing, or if the VMMs are currently distributing data to other users, researchers' queries will be deprioritized below the VMMs' other activities to avoid interfering with census data collection and distribution. We consider this acceptable because it maintains our system's *reliability* for the previous use cases and because timelines for research studies are typically less urgent than those of the system's other users.

4 Conclusion

CSOUP is a streamlined census system that provides *reliability* and *security* by verifying that data is not dropped in network transit, creating temporary file system backups, using the national cloud for storage redundancy and reliable distribution, and keeping data encrypted as often as possible. Our system enables school boards, election boards, and all three levels of government to access data by regular deadlines, while still ensuring that data is transferred securely, correctly, and thoroughly. While our system meets most of the requirements posed by the national government, some questions that remain unaddressed include:

1. how to store PDFs long-term, as discussed in section 2.1.5. We request that the national government consider providing additional storage resources to accommodate the PDFs.
2. how to balance the storage load across PMMs/VMMs within a municipality. If the storage load is unevenly balanced, some machines may run out of storage earlier than anticipated. We will consider designing a process that evenly redistributes data across machines.