NECTAR: National Electronic Census with Tracking, Analysis, and Reporting

Katie Pelton, Jake Jones, and Nikhil Kakarla

July 2, 2024

Contents

1 Introduction

The accurate and efficient collection of population data is a critical task that allows governments to make informed decisions about resource allocation, policy making, and social services provisions. However in Fictlandia, the outdated census system has required citizens to respond multiple times to national, state, and municipal surveys, all of which seek the same information. Our goal is to increase participation and streamline the census process. The National Electronic Census with Tracking, Analysis, and Reporting (NECTAR) is a modernized census system that unifies the census process and improves data collection and management. NECTAR aims to (1) provide governments and researchers with reliable demographic data, (2) enhance the availability of the census to citizens by unifying the input process, and (3) increase scalability and adaptability by modularizing the system to accommodate future population and organizational changes. The challenges of creating such a system include collecting and storing data given storage and network constraints, ensuring reliable and secure record transfers, and establishing procedures to guarantee timely data distribution.

The design of NECTAR prioritizes *reliability*, *availability*, *privacy*, and *modularity*. The foremost objective of the system is to ensure that it continues to function *reliably* even in the face of unexpected failures. NECTAR also fosters high *availability*, ensuring that all citizens and users have the ability to input and retrieve data promptly and uniformly. NECTAR is designed to respect the *privacy* of the people of Fictlandia by anonymizing citizen data and ensuring data security. Lastly, NECTAR prioritizes *modularity*, breaking down the system into distinct, abstractable modules to allow for scalability and easy adaptation in the future.

NECTAR accomplishes these goals by providing a census storage, management, and distribution system spread across local municipal machines and the national cloud service. By simplifying the census collection process and designing network protocols to ensure the reliable transfer of data, NECTAR provides an efficient and reliable network that provides census data to all tiers of government.

In this paper, Section 2 gives a high level overview of the system and hardware. Section 3 explains the design of NECTAR's major components: collection, storage, transfer, and retrieval. Section 4 contains the use case and extreme case quantitative evaluation. Finally, Section 5 discusses NECTAR's limitations as well as potential long term changes.

2 System Overview

2.1 Functional Modules

NECTAR relies on four functional modules. The three layers (collection, storage, and retrieval) are connected through the transfer module. In the collection layer, *availability* is crucial to ensure the maximum number of Fictlandia citizens can complete their census forms. In the storage module, NECTAR prioritizes *privacy* and *modularity* to allow for change in the system and keep citizen data safe. Finally, the transfer and retrieval modules mainly focus on *reliability* and *privacy* to ensure data is safely transfered throughout the system and to the end users. Figure 1 summarizes this organization.

Collection: Census surveys are submitted in two forms: paper and online. Paper forms that are mailed in are uploaded via scanner to the municipal computers each morning before the online form opens. For each paper form this processing produces both a compressed PDF and a database record. For online forms, citizens can use a website to enter their information. NECTAR implements a simple queue to handle a surplus of concurrent online submissions.

Storage: The storage module is divided into two domains: the physical machines and the virtual machines. The physical machines host a hierarchical file system for the compressed PDFs, as well as a database to maintain the full municipal records and late voter registration records. As records are uploaded, the local machine sends the portion of the database record containing state and national information to the cloud for storage on its corresponding VM.

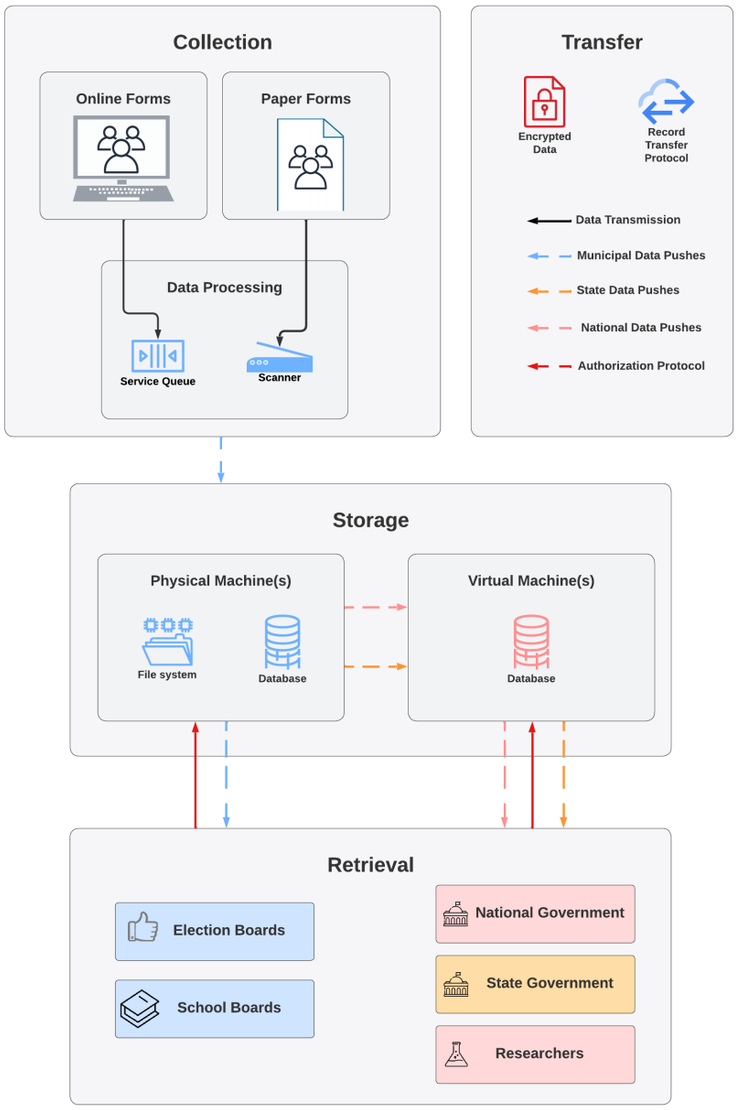

Figure 1: *An overview of the NECTAR system, showing major streams of communication, between the Collection, Storage, and Retrieval modules.*

Retrieval: NECTAR provides methods for governments and researchers to access census records. The authentication protocol limits authorization to specific data for each use case. Election and school boards send signed requests with their database queries and their data is retrieved in real time from the respective municipal physical machine. State and national governments receive their appropriate records directly from the virtual machines in real time as they are uploaded. Researchers send requests to the virtual machines and are given an aggregated, anonymized, and *private* report containing the relevant national statistics.

Transfer: For all network communications, NECTAR employs its own Record Transfer Protocol (RTP) to move records across the system. Layered on top of TCP, RTP achieves *reliable*, in-order transport for each data record. To ensure *privacy*, NECTAR also uses public-private key encryption to encrypt each record during the transfer process.

2.2 Hardware Modules

The functional modules described above are built into the physical infrastructure provided by the government of Fictlandia. This integration is depicted in Figure 2. The hardware modules are divided into two major components: the national cloud which hosts virtual machines depicted in blue and the physical machines depicted in orange which are located in each municipality. These machines are all linked by our transfer protocol, which is used whenever data is moved from machine to machine or uploaded from online forms.

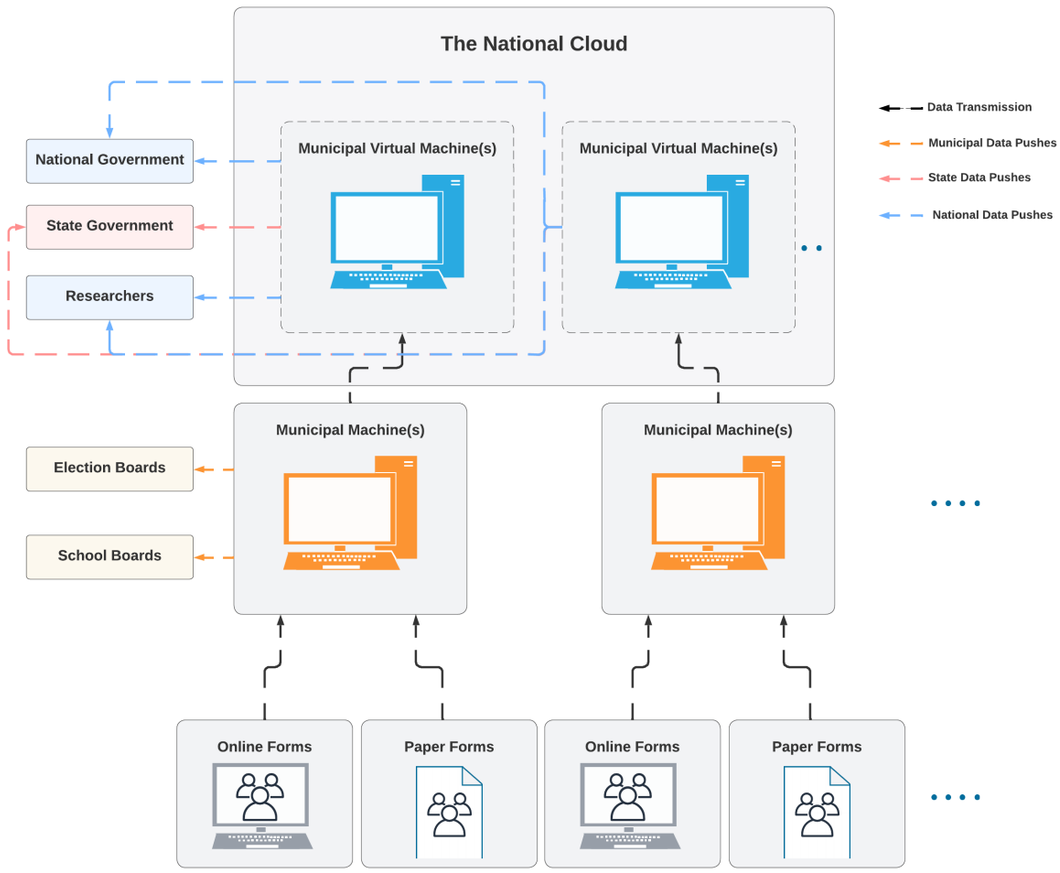

Figure 2: *Illustrates the main hardware components of NECTAR, including storage and network mechanisms.*

Municipal Machine(s)
The physical machines located in each municipality are integral for the collection, storage, and retrieval processes. For collection, these machines take in census forms from households before storing that data in a local database for complete municipal records, a database for newly registered voters, as well as a hierarchical file system that stores compressed PDF files. Finally, the municipal machines are heavily involved in data retrieval as their databases are equipped to be queried by both school and election boards to provide accurate, up-to-date information about the local populations.

Virtual Machines
NECTAR's other segment of machines exist in the national cloud. Each of these virtual machines is paired with a counterpart physical machine and serve as an important hardware component. Specifically, these virtual machines are involved in data storage and retrieval. This is because they receive state and national data from their physical counterparts and store this data in their databases. This data is then used for retrieval, as the VMs send data records to national and state governments and work their coordinators to fulfill queries for researchers.

3 System Design

3.1 Data Collection

NECTAR data collection occurs through two main avenues. Citizens can enter their information either through paper forms which are mailed in or via online submission. This process is depicted in Figure 3.

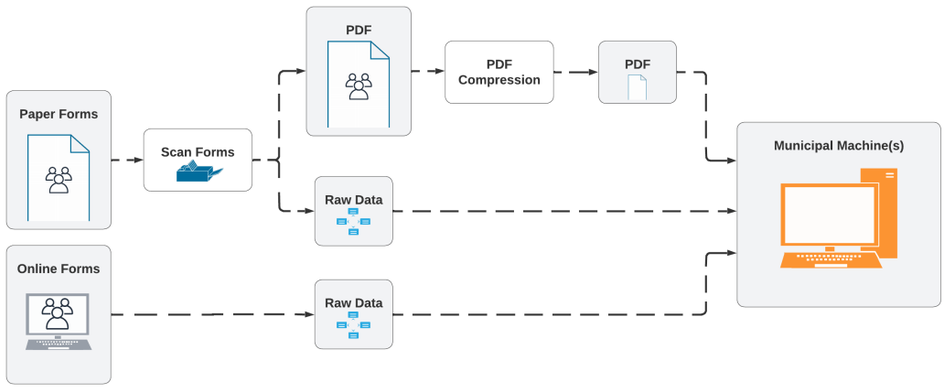

Figure 3: *Illustrates the municipal data collection process from online and paper forms to uploading data into the municipal machine.*

3.1.1 Online Forms

NECTAR's online collection begins with the website form. This piece of the system prioritizes *availability* for as many people as possible, subject to constraints on the number of users that the physical hardware is able to handle. It was important to prioritize *availability* because the form's primary responsibility is to collect large quantities of data; if NECTAR is not available to take in that data, it cannot perform effectively. To that end, NECTAR will track the number of users currently connected to the local machine. Each machine can support up to 125 simultaneous users submitting online forms. If there are already 125 users connected, the machine will place new users into a queue. Users will be removed from the queue once they leave the site or once they move from the front of the queue to the form. Once in the form, users must fill out their information in one session. In the event of a failure, whether a normal crash or catastrophic failure, all users will be kicked off of the site. Entry of users from the queue will be stopped while the machine reboots. Upon rebooting, the local machine will wipe all of the semi-completed records and invite the first 125 users from the queue to begin filling out their forms. Users who had been filling out their form prior to the crash will be asked to rejoin the queue.

To save time and computational space, NECTAR will not implement the ability to save progress on the form. This design makes NECTAR more *reliable* by standardizing failure handling and ensuring that it can continue to receive maximal input following the failure. At the same time, it negatively impacts the user experience by forcing users who have completed much of the form to restart their progress from the back of the queue. Although it would be possible to route user traffic to different municipal machines, it would violate NECTAR's principle of *modularity*. Additionally, failures are rare enough that users will likely tolerate this drawback. As shown below, when forms are submitted at a uniform rate, NECTAR can easily handle all inputs, even during peak hours.

$$\frac{100{,}000 \text{ users}}{1 \text{ machine}} \times \frac{1 \text{ form}}{2.6 \text{ users}} \times \frac{80 \text{ online users}}{100 \text{ total users}} \times \frac{1}{59 \text{ days}} \times \frac{1 \text{ volume 7-8}}{3 \text{ daily volume}} = 172 \text{ forms/machine @ 7-8}$$

$$\frac{125 \text{ forms}}{1 \text{ machine}} \times \frac{1 \text{ form}}{31 \text{ minutes}} \times \frac{60 \text{ minutes}}{1 \text{ hour}} \approx 250 \text{ forms/machine/hour}$$

Max Hourly Input $172 < 250$ **Max Hourly Capacity**

In the event of a 5 minute local machine crash, the website will still be able to handle approximately 230 forms within that hour. In this case, the local machine will still easily be able to handle all of the capacity. If the local machine suffers a catastrophic failure during sysadmin working hours, it will require a 20 minute fix and be able to handle about 167 forms within that hour. In this case, the local machine can handle almost all of the capacity, and the extra capacity can be covered in the extra space provided during all non-failure hours. A discussion of maximum capacity under more realistic, non-uniform conditions and with catastrophic failure can be found in Section 4.2.

Additionally, NECTAR supports late voter registrations. In the case of a voter registration, the record will be manually inputted into the municipal machine database via the

`insertRegistration()` method. This process will happen in real time as new registrations occur.

3.1.2 Paper Forms

In addition to online forms, NECTAR must deal with physical forms. Like the online forms, NECTAR is designed to also have *availability* for all paper forms that arrive. NECTAR's physical collection begins with the delivery of paper forms by mail. Each day, the paper forms will be delivered to the municipal office and compiled. Before the online form is active, the administrative assistant will scan the forms from the day prior. NECTAR will take into account the public's proclivity for procrastination and recognize that there will likely be a significantly higher rate of form submission as the deadline gets closer. Therefore, for the first three-quarters of the census collection period (until the middle of February), the administrative assistant will scan the prior days forms from 7am to 8am. From the middle of February through March 1st, the administrative assistant will scan from 6am to 8am. As shown below, for the first three-quarters of the submission period, NECTAR can easily handle all paper form volume.

$$\frac{1{,}500{,}000 \text{ people}}{1 \text{ municipality}} \times \frac{1 \text{ form}}{2.6 \text{ people}} \times \frac{1 \text{ paper form}}{5 \text{ forms}} \times \frac{1}{59 \text{ days}} = 1{,}956 \text{ paper forms/municipality/day}$$

$$\frac{70 \text{ pages}}{1 \text{ minute}} \times \frac{1 \text{ form}}{1.8 \text{ pages}} \times \frac{60 \text{ minutes}}{1 \text{ hour}} = 2{,}340 \text{ paper forms/municipality/hour}$$

Average Daily Forms $1{,}956 < 2{,}340$ **Max Hourly Capacity**

Paper form collection will also have to *reliably* deal with the possibility of hardware failures. A 5 minute local machine crash has little effect as the assistant can simply wait for the machine to reboot. Each machine can handle 2,145 forms in the remaining 55 minutes, which is more than the projected capacity for the full hour. However, catastrophic failures during paper processing will always occur while the sysadmin is out, since they do not arrive until 9am. Therefore, a machine catastrophically failing during paper processing time will mean those forms will need to be spread out over the following days. If the administrative assistant starts 10 minutes earlier for the 5 days following the catastrophic failure, all of the paper forms from that day can be processed. As with online forms, a discussion of maximum capacity under more realistic, non-uniform conditions can be found in Section 4.2.

3.2 Storage

One of the key functionalities of NECTAR is its ability to store records *reliably*, *privately*, and *modularly*. It is able to ensure that records are not lost under failure, are secure from adversarial attempts to read them, and are easy to move around in case of future system changes.

3.2.1 Municipal Machine(s)

The municipal machines host a shared database to store municipal records, and a separate, small shared database for late voter registrations. The municipal records are hosted solely on these machines because failures will not compromise the storage of the machine. They also contain a file system to store compressed PDFs. The PDFs are stored hierarchically by year, last name, first name, then address. Upon authorization, election boards and school boards can submit querys to access relevant information. The major functions are listed below:

Method	Description
`insert()`	inserts a municipal record into the database.
`insertRegistration()`	inserts a late voter registration into the corresponding database.
`getSchoolBoardData()`	given a school board query, authenticates the user, and returns relevant student identification data from the last completed census.
`getElectionData()`	given a election board query, authenticates the user, and returns relevant voter identification data from the last completed census merged with any information in the recent registration database.
`checkUploaded()`	checks that each stored record has been uploaded to the virtual machine, if not uploads them with RTP.
`clearLateRegistration()`	clears the late voter registration database.

The `getSchoolBoardData()` method returns the school records of all eligible children in the municipality by querying the database and filtering based on the relevant criteria. The `getElectionData()` method returns a compiled list of all voter records, including those in the census database and the late registration database. The late registration database contains only information needed to vote, resets every year, and will be negligibly small when compared to the census database. `checkUploaded()` goes through each census record and checks that it has been marked as uploaded in the database–that is, that RTP sent an acknowledgement back to the local machine. This process is run once every 2 weeks to ensure all records are up to date. Finally, `clearLateRegistration()` is called once a year after the new census is complete to clear the late voter registration database.

3.2.2 Virtual Machine(s)

Each set of virtual machines assigned to a single municipality will run its own database, with a copy of the data sent to it by its corresponding physical machine. These databases will contain the state and national records for each individual that is on the partner local machine. Although this design decision replicates state and national data, it improves performance by dividing the completion of data retrieval between local and virtual machines. Since NECTAR pushes all state and national data in real time to the respective governments, the set of database functions is different that the municipal machines. Thus, the major functions of the machines are listed below.

Method	Description
`insert()`	inserts a state record into the database.
`researcherQuery()`	if the virtual machine is a controller, catches a researcher query, authenticates the user, and returns relevant aggregated statistics for analysis from the last completed census data. If not, then returns its data records in a csv file for combining by the controller.

These functions are different from the municipal storage because virtual machines do not allow state and national governments to query databases for their records. Instead, their methods focus on researcher queries. A discussion of how the researcher queries work can be found in Section 3.4.3.

3.2.3 Database Security

One of the main priorities of NECTAR is *privacy*. Thus, the system guarantees that only authorized agencies can access an individuals' data. To achieve this, NECTAR incorporates MIT's kerberos protocol to authenticate users before granting them access to the database. Users are only authorized to have read capabilities, except for the machines that insert the data themselves. Each user only has authorization for the specific methods required for their use case: for example, a query from the election board will only be authorized to

access `getElectionData()` to prevent access to sensitive data.

To store sensitive username, password, and key information, the database implements hashing with salt as described in lecture [1]. Operating under the threat model of adversarial access to the entire stored table, only hashed passwords are stored and the system uses slow hashing. To authenticate a user, the inputted password is compared to the hash of the concatenated hashed password and its unique salt. This policy protects sensitive authentication data from exploitation.

Each municipal machine also has a unique key that authorizes write access to the national database. When transferring records to the national cloud, NECTAR use a public-private key method to ensure that only the correct virtual machine in the cloud receives the data. This measure guarantees that a municipal record can only be inserted into its corresponding virtual machine, even if an adversary attempts to redirect or disrupt the operation.

3.3 Transfer

One of the integral functionalities of NECTAR is its ability to transfer records *privately* and *reliably* between machines. This is critical to achieving the design principle of *reliability* as data must be complete and consistent across all machines despite failures. Additionally, this functionality must ensure user *privacy*, as unauthorized agents should not have access to sensitive population data.

3.3.1 Record Transfer Protocol (RTP)

In order to facilitate *private* and *reliable* data transfers throughout the system, NECTAR will use its own Record Transfer Protocol (RTP). RTP begins by marking all records in the sender queue as `toSend`. Then, RTP utilizes the *reliable* TCP protocol and secure private-public key encryption to send packets and receive delivery confirmation via acknowledgements. Receivers will only send the last acknowledgement back after inserting the complete record into their databases. Once received, senders will then mark the record as `delivered` and update their database record as such. Additionally, receivers will discard any record that has not completely arrived 5 minutes after the first packet is received. Finally, if the sender has not received a final confirmation from the receiver after 5 minutes, they will timeout and resend the entire record.

This system allows NECTAR to *reliably* deal with minor and catastrophic computer failures as well as problems with the network. For example, if the sending machine crashes midway through a file transfer, the receiving machine will throw out the partially filled data. Once the sender reboots, the record will still be marked as `toSend` and it will resend the entire record. Similarly, if a receiver crashes during a record transfer, the sender will not receive the last acknowledgement packet and will continue timing out and trying to resend the data until the receiver is back online. Finally, if the network fails during a record transfer, the last TCP acknowledgement will never be received by the sender and the process will continue restarting until the network is back online. Thus, RTP provides *reliable* data transfer.

Another core feature of RTP is the *privacy* and security with which it transfers records. For this paper, NECTAR's threat model is any adversary with the ability to sniff data across the network, transfer fake packets, and insert packets into the data stream. The desired policy is to provide both confidentiality (adversaries can not read data flowing across the network) as well as integrity (adversaries can not tamper with packets and go undetected). In order to achieve these goals, NECTAR uses public-private key encryption to encrypt each record before it is transferred over the network. On May 1st each year, unique public and private keys are distributed to each user and machine to be used for the encryption and decryption process. This methodology, as designed in lecture, achieves our policy [2].

Therefore, RTP provides *reliable* and *private* data transfer across the network. Although this protocol adds overhead and decreases performance by slowing the data transfers, it is an intentional design choice to guarantee the safe and accurate transfer of all data throughout the NECTAR system.

3.3.2 Immediate Uploads

In order to facilitate the distribution of the data from the municipal physical machines to stakeholders, NECTAR will immediately upload any new records on the local machines to the corresponding virtual machine. However, NECTAR will not upload the full municipal record, but rather will only upload the information required on the state and national level. This allows for smaller uploads and decreased use of virtual machine storage while still maintaining near real time transfer. The decision to make real time transfers requires more parallel processes to be run on each machine, but achieves the design principle of *privacy* by ensuring that citizen's municipal-only data is not stored on the national cloud machines. It also achieves the principle of *modularity* by separating the purpose of the physical and virtual machines based on the data they possess. As shown below, NECTAR has the network capacity for this real time transfer in the average case:

$$\frac{125 \text{ online entries} + 70 \text{ paper entries}}{1 \text{ local machine}} \times \frac{150 \text{ words}}{1 \text{ entry}} \times \frac{3 \text{ encrypted words}}{1 \text{ word}} \times \frac{32 \text{ bits}}{1 \text{ word}} \times \frac{1 \text{ Mb}}{1{,}000{,}000 \text{ bit}} = 2.8 \text{ Megabits}$$

At any given point, the maximum amount of incoming data, 125 simultaneous form entries and 70 scanned paper forms, will cause 2.8 Megabits of upload traffic. Each machine has up to 1000 Mbps of upload speed and will easily be able to handle instantaneous uploads from the physical machines. An analysis of performance under the worst case scenarios can be found in Section 4.2.

3.4 Retrieval

NECTAR's data retrieval is split into three modules based on specific user needs. Each user has a different set of priorities and NECTAR tailors these retrieval methods to meet those varying needs.

3.4.1 School and Election Boards

Retrieval for the school and election boards is handled locally on the municipal level to achieve the design principle of *modularity*. When the database is created on the municipal machines, it has methods for data retrieval that are specific to both types of boards. `getSchoolBoardInformation()` can be used by the school boards to retrieve their data. The function returns the count of the current number of students, their names, and language accommodation status based on the most recently completed census. Similarly, `getElectionInformation()` can be used by election boards to receive voter information. This information is based on the most recent census but also includes any updates that were inputted as late registrations. This way, election boards can receive the most up-to-date and accurate voter information.

This query model is an important design decision within NECTAR. Due to the year round nature of local elections and the changing demands of school systems, it is important that the information always be *available* for local government use. However, due to the potential for local machine or network failures, the database system may take up to several days to return full results in the worst case. Therefore, NECTAR guarantees delivery of school or election board data 3 days following a request and these requests can be made at any time throughout the year. While this design decision introduces a 3 day buffer into the system, in the vast majority of cases the data will be returned almost immediately and it facilitates the major needs of the local boards while following the principle of *modularity*.

3.4.2 National and State Governments

Separate from the local governments, retrieval for the national and state governments will be handled by the virtual machines. When a record is received from a physical machine, the virtual machine will immediately send it on to the correct state computer. It will also partition the record into a national data record and send that to the national cloud. Unlike the municipal governments and researchers, there are no database methods associated with retrieval– the virtual machines simply send the records that they receive in near real-time to the governments. Following retrieval, it will store the full record in the virtual machine

database using `insert()`, as covered in Section 3.2.2.

Since the virtual machines only store state and national information, they are ideal candidates to function as retrieval sources for that data. As discussed above, the virtual machines immediately send their state and national records out to the users of those records. This is because it offers the opportunity to fix a weakness inherent in the school and election boards–delayed guarantees on data retrieval. In the municipal case, failures may make the retrieval of data take much longer than expected, so guarantees on that data must correspondingly be longer. Because national and state data is sent as it is received to its final source throughout January and February, there is very little processing that needs to be done during March. It is then feasible to ensure that all of the data necessary is directly on the state and national systems, regardless of failure, by April 1st. This makes NECTAR more *reliable* and *available* and meets the specific needs of the state and national governments.

3.4.3 Researchers

NECTAR data acquisition for researchers will look significantly different than for national, state, or municipal governments. This is because researchers have different needs and priorities than the government bodies. They want access to national data over time, which allows them to perform national and trend studies. Their needs are also less time-sensitive than, for instance, the need for the municipalities to run spontaneous local elections. At the same time, the researchers must not have access to individual records to ensure citizen *privacy*. This is because, even if anonymized, these records could be used to reveal sensitive information about an individual. With that in mind, NECTAR prioritizes the *privacy* of user data over rapid data access in researcher retrieval. This design choice ensures the needs of all users are met.

To accomplish these design goals, each of the five national data centers will have one virtual machine that acts as a "controller". A researcher makes a specific query to NECTAR asking for a set of aggregated statistics (i.e. how many adults are there in Fictlandia). The nearest data center to that researcher will catch this `query`. The controller virtual machine at that data center then becomes the "leader" of the entire query. It will contact the controllers in each of the other data centers and relay the `query`. Each controller, including the leader, will then query all of its virtual machines using the `researcherQuery()` method and aggregate their data into anonymous statistics. When the "controller" receives each of the five aggregated reports, it will combine them and parse the data. At the end of the process, the leader will return one report to the researcher. This report will detail the aggregated statistics on a national level without revealing sensitive personal information.

This design decision to aggregate the researcher statistics certainly decreases the speed of the system by making researchers wait for responses. However, it meets the design goals of NECTAR by prioritizing citizen *privacy* and system *modularity* via a decentralized retrieval process.

4 Evaluation

4.1 Use Case Evaluation

The primary means of evaluating NECTAR's performance is to ensure that the system meets the information needs of each of its users while protecting the *privacy* of Fictlandia's citizens. The following sections analyze how NECTAR is able to fulfill these requirements.

4.1.1 National Government Redistricting

The first major use case is providing population data to the national government for redistricting. NECTAR ensures maximum accuracy in population count throughout its design by providing reliable transmission of population data to the national government. When handling inputs, it is *available* to handle all of the load from citizens submitting their information in almost all conditions, as will be discussed in Section 4.2. Additionally, there are multiple features of NECTAR that ensure *reliable* transfer of data by April 1st. First, immediate uploads to the virtual machines, then to the state and national databases through

January and February ensure that the vast majority of data is uploaded by the beginning of March. Second, 10 days before April 1st, NECTAR will query all of the national records and ensure that all of them were sent to the national cloud. Third, two copies of national records exist, on both municipal and virtual machines, to ensure that system failures will not prevent the system fulfilling its responsibilities by April 1st. However, one major impact to consider for population count is the continued under counting of individuals without a permanent address or location to live, which might lead to inaccuracies in redistricting and negative effects on the communities in question.

4.1.2 Municipal Government Elections

Running efficient elections throughout the year is an essential use case of NECTAR. Under NECTAR, every municipality is provided a parallel database, from which they can use the `getElectionInformation()` method to get all relevant records for the current election. This is possible because the system is *modularized*, such that each municipality only handles its own data. It has the benefit of allowing up-to-date information to be accessed throughout the year. However, this decision places a higher load on the municipal computers during the high volume months and computer failures could result in the data being inaccessible. If NECTAR does suffer from a catastrophic local machine or network failure, it will take less than 3 hours to fix on average but could run to several days in the worst case. To be guaranteed recent results, election boards should query NECTAR for voter rolls several days before they need them. If they need to update the rolls or query more results in NECTAR during the election, there is an approximately 1.4% chance that the system is down due to a catastrophic or network failure. In this case, the election boards may have to register new voters by hand. Overwhelmingly though, NECTAR will be able to return voter rolls as needed.

4.1.3 School Board Identification and Assignment

Similarly, before the start of the school year, school boards require accurate data about children to track attendance and make school assignments. The parallelized databases on the physical machines provide a method to query all relevant information:
`getSchoolBoardInformation()`. Since it is essential that all children enter school on time, NECTAR ensures that data is always query-able. In the ideal case, school board records are small and would be rapidly query-able, accessing the most recent year's census data for the school. Every April 1st, the new year's census election data becomes available to query. However, one flaw of NECTAR is that the municipal physical machines contain the sole copies of municipal data. If the local machines fail during a critical time, the municipalities cannot access the databases to get their information. When the school boards require this information in May, they will be guaranteed to receive this information within several days. Most of the time it will return results rapidly, but in the unlikely case of a catastrophic or network failure, it could take several days to restart the system and be able to send information. As a result, school boards should act similarly to election boards, in that they should query NECTAR several days before they need data to be guaranteed that the system can *reliably* send that data by the final due date despite potential network failures.

4.1.4 External Research

The final use case to consider in the NECTAR design was for external research. External research entails studying national data over longer periods of time. Because NECTAR stores a copy of all historical national data on the virtual machines, researchers are able to submit a request for a statistical report from the virtual machine controllers. The Record Transfer Protocol can *reliably* provide this information over time. One caveat to this design is that the researchers priority is placed below that of the national and state data. Since the researchers do not have immediate, essential use for their data and are likely more tolerant of delay, it may take more time to aggregate and send a report. The sending of national and state records will always have higher priority than research aggregation.

4.2 Extreme Case Quantitative Evaluation

In addition to ensuring that NECTAR can meet the design needs of its many users, it is important to understand the limitations of the system. Different constraints and their

threshold values are explored below.

4.2.1 Data Collection Bandwidth

One of the most important limitations of NECTAR is the number of users that can enter their information into the system near the end of the census cycle. This limitation is important because census form submission likely won't be uniform. Rather, many citizens may hold off until the deadline to fill out their forms. Therefore, it is important to understand precisely what percentage of the population could enter their information in the last week of the census cycle without exceeding system capacity. In this scenario, it is assumed that all machines do not crash in the last week. The results are shown below:

Paper Form Constraint:

$$1 \text{ week} \times \frac{7 \text{ days x } 2 \text{ hours}}{1 \text{ week}} \times \frac{60 \text{ minutes}}{1 \text{ hour}} \times \frac{70 \text{ pages}}{1 \text{ minute}} \times \frac{2.6 \text{ people}}{1.8 \text{ pages}} = \frac{84{,}933 \text{ people}}{1.5 \text{ mil ppl * } 20\% \text{ paper users}} = \textbf{28\% capacity}$$

Online Form Constraint:

$$1 \text{ week} \times \frac{7 \text{ x } 14 \text{ hours}}{1 \text{ week}} \times \frac{60 \text{ minutes}}{1 \text{ hour}} \times \frac{125 \text{ forms}}{31 \text{ minutes}} \times \frac{2.6 \text{ people}}{1 \text{ form}} = \frac{61{,}645 \text{ people}}{100\text{k ppl x } 80\% \text{ online users}} = \textbf{77\% capacity}$$

These calculations attempt to capture what percentage of the population could submit their forms in a given week of the election cycle while still allowing our system to fully process the data. For the paper forms, NECTAR only has one scanner per municipality that is operated two hours per day. Therefore, the rate limiting factor is the number of pages that can be scanned in the 7×2 hours each week. Thus, NECTAR can process up to 28% of the population's paper forms provided that the assumption that 20% of the citizens submit paper forms holds. To increase the paper forms percentage, the administrative assistant could spend more than two hours scanning in the mornings. Similarly, the main bottleneck for the online forms is the parallel processing of the municipality physical machines. Since each machine can only handle 125 parallel users and each form takes roughly 31 minutes to complete, this process allows NECTAR to process up to 77% of the population in a given week.

When considering the most important use case of ensuring that all data is processed before the deadline of April 1st, it becomes evident that the paper form constraint does not apply. This is because there is a month in between the collection deadline and when the data needs to be ready for the governments. Therefore, this time could be used to process any late paper forms. However, the 77% constraint on the online forms is a important limitation. If less than 23% of the population have not submitted their online forms by the last week, it is likely that some citizens will not be able to submit their forms before the deadline even if all citizens line up throughout the day to submit their forms. NECTAR therefore can handle all reasonable load, but in the worst cases will fail to handle all submissions.

4.2.2 Upload Bandwidth

Another important worst-case constraint of the NECTAR system is the upload and download bandwidth. Since NECTAR employs an immediate upload system from its physical machines to its virtual machines, it is important to analyze if the system has proper bandwidth. The calculations below ensure that NECTAR has adequate bandwidth by exploring the number of paper and online forms that can be submitted at the same time. If all paper and online forms were submitted in the same second, would NECTAR have the required bandwidth to immediately upload them?

$$\left(70 \text{ paper forms} \times \frac{6{,}000{,}000 + (200 \text{ x } 32) \text{ bits}}{6 \text{ pages per form}}\right) + \left(125 \text{ online forms} \times \frac{200 \text{ words}}{1 \text{ form}} \times \frac{32 \text{ bits}}{1 \text{ word}}\right) = 70 \text{ MB}$$

This calculation demonstrates that even at max capacity, with 70 paper PDFs, their corresponding records, and 125 online forms all being submitted at once, NECTAR will only need to transfer 70 MB on its network. Since each physical machines has upload and download bandwidths of 1,000 MBps, this calculation shows that even in the most intensive submission scenario, NECTAR has adequate bandwidth to transfer all relevant data.

4.2.3 Long-term Storage Capacity

Another important extreme case scenario is the long term storage constraints of the system. NECTAR retains both the PDF version of paper forms as well as data records stored on the local machines. With 70 year life cycles for this data, it is inevitable that NECTAR will run out of storage. Given that each PDF is 1 MB post-compression and election databases are negligible in size given they reset every year, NECTAR's storage life calculations are shown below:

$$\text{Percent of storage for PDFs} = \frac{\text{1,000,000 bits}}{1,000,000 + (200\ words \times 32\ bits/word \times 13\ records/pdf)} = 92\%$$

$$92\%\ \text{storage} \times \text{1 TB storage/machine} \times \frac{\text{1 year}}{\text{20,000 PDFs} \times \text{1 MB/PDF}} = \text{46 years}$$

The way that NECTAR deals with PDFs will cause the storage of the local machines to fill up in about 46 years. It is important to consider options to mitigate this system drawback. Either more hardware will need to be requisitioned after 46 years, a better compression algorithm will need to be developed, or PDFs should not be stored on the local machine at all. Currently, NECTAR stores the PDFs, which increases *reliability*, since they provide a check on the veracity of each citizen's record to resolve any processing errors.

4.2.4 Computer Failures

In addition, an important extreme case failure for NECTAR is the potential for any individual machine to crash. NECTAR is resilient in the face of a minor crash, where a computer goes down for 5 minutes. Although end users will have to restart their data entry process, the rest of the system, including RTP and immediate uploads, can handle computer failures and guarantee the safe, timely transfer of data.

However, catastrophic computer failure represents a different challenge, since a computer can go down for up to 3 days (Friday at 5PM to Monday at 9AM). In this case, NECTAR will not be able to collect any data from the users that would connect to this municipal machine and will therefore lose significant input capacity. If this happened in the last week of the census, where there is a projected increase in submissions, it would affect the maximum number of final week online forms that could be handled. Following the logic from Section 4.2.1:

$$\text{1 week} \times \frac{\text{(7 days x 14 hrs) - 34 hrs}}{\text{1 week}} \times \frac{\text{60 mins}}{\text{1 hr}} \times \frac{\text{125 forms}}{\text{31 minutes}} \times \frac{\text{2.6 people}}{\text{1 form}} = \frac{\text{40,285 people}}{\text{100k ppl x 80\% online}} = \mathbf{50\%\ capacity}$$

If the machine suffers a catastrophic failure right after the sysadmin leaves on Friday, the local machine cannot input data from 5pm to 10pm on Friday, 8am-10pm on Saturday and Sunday, and 8am-9am on Monday. This is thiry-four hours of lost upload time. In this case, the municipal computer could only handle 50% of its expected two month load during the final week. This is still a reasonable quantity, though a fail point all the same. Also, paper forms do not effect this bottleneck since they can be retroactively scanned after the computer is rebooted and it is almost impossible for a machine to crash more than once in the last week of the census cycle.

4.2.5 Network Failure

The final extreme case scenario to consider is that of a inter-network collapse. Specifically, if the link between a municipality and its virtual machines go down, a network failure would create a queue of records that need to be uploaded. Therefore, it is important to understand how long it will take for NECTAR to catch up after an extreme network failure. These calculations assume an extreme network failure of 3 days.

$$\text{3 days} \times \frac{\text{24 hours}}{\text{1 day}} \times \frac{\text{60 minutes}}{\text{1 hour}} \times \frac{\text{125 records}}{\text{31 minutes}} \times \frac{\text{200 * 32 bits}}{\text{1 record}} \times \frac{\text{1 sec}}{\text{1,000 MB}} = \text{0.11 seconds}$$

Even in the case of an extreme network failure on the output side and maximum form inputs during that time, NECTAR will be able to catch up on uploads in less than half a second. Therefore, there is no real limitation on upload speed even in the network fails for a prolonged amount of time.

Additionally, a network failure could occur between a municipal machine and its online users. This failure could also last for up to 3 days. This means that in the worst case, a municipality could lose the ability to collect online forms for 3×14 hours. Therefore, the new maximum percentage of users that can submit their forms is shown below:

$$1 \text{ week} \times \frac{\text{(7 days x 14 hrs) - 42 hrs}}{\text{1 week}} \times \frac{\text{60 mins}}{\text{1 hr}} \times \frac{\text{125 forms}}{\text{31 minutes}} \times \frac{\text{2.6 people}}{\text{1 form}} = \frac{\text{40,285 people}}{\text{100k ppl x 80\% online}} = \textbf{44\% capacity}$$

Therefore, in the event of both worst case network failure and high final week submission, NECTAR could still handle **44**% of the entire submitting population within the last week.

5 Limitations and Long-Term Changes

NECTAR is not without areas of weakness. In collection, local machine crashes kick all users off the form submission page and send them to the end of the queue. Though this allows for a simpler model, it has serious drawbacks in user experience. Also, this functionality detracts from the ability of the nation to maximize responses to the census as some users may not return to the form if its crashes. In the future, the users might be allowed to continue where they left off following a crash or shift to another local machine to continue the form. Another weakness concerns machine failure in the last week of census collection before the March 1st deadline. If many machines go down and the procrastination model holds, there is little NECTAR can do to handle the increased system load. Governments are encouraged to ensure at least 56% of their population have filled their forms before the final week. In the future, this final week capacity bottleneck could be improved. Next, NECTAR's retrieval protocol on the municipal level requires database queries, which means network or machine failures around April 1st could prevent data delivery. Accordingly, NECTAR's protocol recommends all municipal users submit requests proactively to ensure the complete delivery of relevant data, but this still represents an area of weakness. Finally, NECTAR also requires administrative assistants across the country to work early mornings and on the weekends, which is not ideal for government workers or volunteers.

Over the long term, NECTAR will have to adapt to new and changing use cases. On the local level, redistricting changes may alter the size of municipalities. In that case, machines may be moved between municipalities. By ensuring that VMs are *modularized* and not connected with each other, it is much easier to transfer local machines between municipalities, by simply moving storage and transferring database connections. Furthermore, municipalities may require different information at different times as time goes on. NECTAR provides the ability to add methods to local databases, and is always available for real time access. Finally, NECTAR will not be able to handle the long term storage demands of the system– it is predicted to fill all of its storage within 46 years and will require more storage. At that time, NECTAR' *modularity* will again allow it to adapt: more storage can be added to the local machines, which will run out first because they store PDFs. Nothing else in the system will have to change since PDFs are not stored virtually, although advances in technology might mean hardware upgrades would be a generally useful investment.

6 Conclusion

NECTAR meets our goal of performance by ensuring the timely delivery of census data to all stakeholders in all use cases while also allowing as many Fictlandia citizens as possible to confidentially fill out the census forms when and how they want. The core design decisions of NECTAR facilitate this objective. The overall architecture of using virtual machines as intermediaries between the local machines and users allows the offloading of distribution from local machines to the cloud, and provides for immediate use of data on April 1st, regardless of computer failure. The modular design of each municipality allows the overall network to scale and change as Fictlandia evolves. The design decision to upload data in real time to each stakeholder ensures the timely distribution of data, and the compression of PDFs allows for increased storage. Further, aggregating and anonymizing national statistics for researchers ensures the privacy of citizen data while still allowing them to carry out widespread studies. Finally, the careful design of the Record Transfer Protocol ensures that each data record is securely and reliably transferred to each user. These critical system behaviors enable NECTAR to meet its core design principles of *reliability*, *availability*,

privacy, and *modularity*.

7 Author Contributions

The authors collaborated on all portions of the NECTAR system. In writing our final report, Nikhil focused on the quantitative evaluation and record transfer protocol, Katie focused on the introduction, overview, storage, and database security, while Jake focused on use case evaluation, data collection, retrieval, and limitations.

8 Acknowledgements

We would like to thank Karen Sollins for her technical feedback and guidance on NECTAR, as well as Michael Trice for his communications feedback on our paper structure. We are also grateful to Katrina LaCurts, Deb Torres, and the 6.033 staff for their work in preparing and teaching this course.

9 References

[1] 6.033 Lecture 21, Spring 2023
[2] 6.033 Lecture 23, Spring 2023
[3] "How to Compress a PDF and Reduce Its File Size." PDF Blog — Investintech PDF Solutions, 20 Feb. 2023, www.investintech.com/resources/blog/archives/6602-how-to-compress-pdf.html.