E2EE Platform
Literature Review

Roey Fabian - 208277285
Amit Cohen - 316202555
Ron Sharabi - 207209297
Eran Fishbein - 318316635
Supervisor: Dr. Nir Grinberg

Summary

The evolution of surveys has made them an indispensable tool in fields such as marketing, public opinion research, healthcare, education, and more.
Surveys have evolved from door-to-door surveys, to phone surveys, to digital platforms, including online surveys, email surveys, and social media polls.
These advancements have greatly expanded the reach of surveys, allowing researchers to gather data from a much larger and more diverse group of people in a fraction of the time it would have taken with traditional methods, but at the same time, they have posed greater risks for obtaining fake or unreliable data. The ease of access to surveys and the anonymity provided by online platforms can make it easier for respondents to provide inaccurate or false responses, either intentionally or unintentionally.

With the rapid development of technology, everyone nowadays is exposed to news from all over the world and can share their opinions with others from the palm of their hands. The rise of smartphones, social media platforms, and instant messaging has made it easier than ever for individuals to access real-time information, engage in conversations, and participate in global discussions. This data is ready to be used to create up-to-date, more accurate, and more diverse surveys.

To harness this data in the most effective way, we are developing the E2EE (End-to-End Encryption) Platform. With the E2EE Platform, the user could create a system which will generate connections to the participants' instant messaging apps (Whatsapp, Telegram and Signal), manage the connections, feed a database constant data from chats of the particiapte's choosing and anonymize this data.
The E2EE Platform will solve the problems of out-of-date surveys, of misleading responses, and of low participation rates.
Our platform is unique because of its ability to connect the user to apps with different types of protocols, to give the user information about the collected data and about the participants constantly, and it supports hebrew and hebrew anonymization.

We plan to use mainly Matrix protocol and LLMs to achieve our goal and to create a management dashboard for the user in order to make use of our platform as easy and useful as we can.

Current State

The organizations that are expected to use our platform are survey institutes.

Current Workflow

1. **Defining survey objectives** – Determining the topics for which information is needed.
2. **Identifying the target population** – Deciding who are the people to collect data about/from.
3. **Selecting the survey method** – Choosing between by phone, online, or face-to-face surveys.
4. **Designing the survey** – Writing questions relevant to the survey objectives.
5. **Choosing the sampling method and sample size** – Ensuring the sample represents the selected population.
6. **Monitoring response rates** – Tracking participation levels.
7. **Processing and analyzing data** – Cleaning data, weighting responses, and performing statistical analysis.
8. **Presenting findings** – Producing reports and publishing results.

Challenges and Issues

1. **Sample representativeness** – Difficulty in representing all population groups, which may lead to biased results (in cases where the survey does not target a specific population).
2. **Problematic responses** – Participants might misunderstand questions, provide socially desirable answers, or give false responses.
3. **Low participation rates** – People may choose not to participate, leading to unrepresentative samples.
4. **Dependency on technology** – The chosen survey method may affect the results. For example:
    a. Online surveys might not reach all populations and carry risks of receiving fake or misleading responses.
    b. Telephone surveys may face challenges as modern apps flag unknown callers as spam, causing people to ignore calls.
5. **Statistical errors** – A small sample size may result in high errors, and incorrect data analysis can lead to faulty conclusions.
6. **Real-time changes** – Public opinion can shift rapidly, making survey results outdated before they are published.

Similar Platforms

mturk

A platform that can be used to give assignments to different people online in exchange for payment.
It can be used to get private chats from users. [7]

	Mturk	E2EE
Sampling	Advantages: · A platform with a large number of users. · Users from many different locations. Disadvantages: · Users are likely those seeking supplemental income (not all populations). · Limited to users with internet access (not all populations). · Restricted to registered users of the platform. · Not operational in all locations or at the same scale.	Advantages: · **No dependency on an existing user base**, providing flexibility in participant recruitment, unlike Mturk, which relies on its registered users. · The client has **greater control over the sample composition**, tailoring it to specific research needs, whereas Mturk often lacks flexibility in sampling. Disadvantages: · The client is responsible for recruiting participants, which may require more effort than leveraging Mturk's existing user base. · Limited to users with internet access (not all populations). · Participants are often those seeking supplementary income, which may reduce demographic diversity.
Accessibility and Participation Rate	Advantages: · Participants cooperate actively as it serves as a source of income for them. Disadvantages: · Data quality depends on participant availability.	Advantages: · Participants cooperate actively as it serves as a source of income for them. · **Data flows continuously** without requiring participant intervention**,** overcoming the availability issues seen in Mturk.

Reliability	Advantages: ・ Feedback is available about participants. Disadvantages: ・ It is unclear if the data is authentic or has been altered in any way.	Advantages: ・ **Data is reliable and encrypted,** ensuring participants cannot alter it, solving the authenticity issues Mturk might have. ・ The system notifies the client if participants disconnect, ensuring **transparency**, unlike Mturk, where such notifications are unavailable.
Data Processing	Disadvantages ・ No data processing is provided. ・ Requires separate processing after data collection. .	Advantages: ・ Collected data undergoes **anonymization** to ensure privacy compliance.. ・ Integrated into a structured data repository for easy access. ・ Enables quick retrieval of basic statistical data.

The National Internet Observatory

a platform that collects data from computers, tablets, and mobile phones of participants, either on a voluntary basis or for payment. Data is gathered through a browser extension or a mobile application. [8]

	NIO	E2EE
Sampling	Disadvantages: ・ Restricted to American participants aged 18 and older. ・ Dependent on an existing user database, limiting accessibility. ・ Limited to users with internet access (not all populations). ・ Participants are often those seeking supplementary income, which may reduce demographic diversity.	Advantages: ・ No dependency on an existing user base, **providing flexibility in participant recruitment**, and allowing global recruitment without age or location restrictions. ・ The client has greater control over the sample composition, tailoring it to specific research needs. Disadvantages: ・ The client is responsible for recruiting

		participants. ・Limited to users with internet access (not all populations). ・Participants are often those seeking supplementary income, which may reduce demographic diversity.
Accessibility and Participation Rate	<u>Advantages:</u> ・Participants actively engage as it provides a source of income for them. <u>Disadvantages:</u> ・Data availability is dependent on processing by NIO, which may not be immediate.	<u>Advantages:</u> ・ Participants cooperate actively as it serves as a source of income for them. ・Data flows continuously in **real time**, ensuring immediate access to insights, unlike NIO, where processing may face delays.
Reliability	<u>Advantages:</u> ・Data is reliable and encrypted, ensuring participants cannot alter it. <u>Disadvantages:</u> ・ Data integrity cannot always be guaranteed; participants can stop the extension or app at any time and reactivate it later.	<u>Advantages:</u> ・Data is reliable and encrypted, ensuring participants cannot alter it. ・The system ensures data integrity and notifies the client of disconnections, unlike NIO, where participants can stop and resume, causing inconsistencies.
Data Processing	<u>advantages:</u> ・Data processing is handled by NIO, utilizing pre-existing datasets for streamlined analysis. <u>Disadvantages:</u> ・Data collected does not undergo anonymization.	<u>Advantages:</u> ・Collected data undergoes **anonymization**, ensuring compliance with privacy standards, addressing the absence of anonymization in NIO. ・Integrated into a structured data repository for easy access. ・Enables quick retrieval of basic statistical data.

Crimson Hexagon

A distributed system designed for secure and automated data collection from global and personalized users. It operates through background applications on users' mobile devices, ensuring privacy without accessing personal content. [9]

	Crimson	E2EE
Sampling	Advantages: · No dependency on an existing user base, providing flexibility in participant recruitment.. · The client has greater control over the sample composition, tailoring it to specific research needs. · Does not rely on specific platforms or services. Disadvantages: · The client is responsible for recruiting participants. · Limited to users with internet access (not all populations). · Participants are often those seeking supplementary income, which may reduce demographic diversity.	Advantages: · No dependency on an existing user base, providing flexibility in participant recruitment. · The client has greater control over the sample composition, tailoring it to specific research needs. Disadvantages: · The client is responsible for recruiting participants. · Limited to users with internet access (not all populations). · Participants are often those seeking supplementary income, which may reduce demographic diversity.
Accessibility and Participation Rate	Advantages: · Participants cooperate actively as it serves as a source of income for them. Disadvantages: · Temporary interruptions may occur if participants disconnect or are unavailable, potentially leading to gaps in the data collected. · The system cannot access the actual content of messages or calls (e.g., WhatsApp), limiting the types of data collected.	Advantages: · Participants cooperate actively as it serves as a source of income for them. · Data flows continuously without requiring participant intervention, ensuring uninterrupted collection, compared to potential gaps in Crimson.

Reliability	Advantages: ・Data is reliable and encrypted, ensuring participants cannot alter it. ・The system notifies the client if participants disconnect, maintaining transparency. Disadvantages: ・The system cannot access the actual content of messages or calls (e.g., WhatsApp), limiting the types of data collected.	Advantages: ・Data is reliable and encrypted, ensuring participants cannot alter it. ・The system notifies the client if participants disconnect, maintaining transparency. **The platform can analyze more than just metadata**, unlike Crimson, which focuses primarily on metadata processing.
Data Processing	Advantages: ・Collected data undergoes anonymization to ensure privacy compliance.. ・Integrated into a structured data repository for easy access. ・Enables quick retrieval of basic statistical data. Disadvantages: ・Processing is limited to metadata; the system cannot analyze the content of messages, focusing instead on metadata like time, frequency, or IP addresses.	Advantages: ・Collected data undergoes anonymization to ensure privacy compliance.. ・Integrated into a structured data repository for easy access. ・Capable of processing both metadata and certain types of content, **expanding analytical possibilities compared to Crimson**, which is restricted to metadata like time or frequency.

Comparison of Features Across These Platforms:

	E2EE	mturk	NIO	Crismon Hexagon
Access to private chat content	**Yes**	Yes	Yes (only for non E2EE chats)	No
Scalability	Yes	Yes	Yes	Yes
Advanced technology operation - Client side	Yes	No	No	Yes
Advanced technology operation - User side	No	No	No	No
Support for Hebrew	**Yes**	Yes (but participant quantity limited)	No	No
Automated data pulling	Yes	No	Yes	Yes
Data reliability and integrity	Yes	Depends on participant reliability	Yes	Yes
Seamless Data	**Yes**	No	No	Yes
Anonymization	**Yes**	No	No	Yes

Relevant Technologies

Matrix (Dendrite and Mautrix bridges)

To get the data from our contributors on WhatsApp, Signal, and Telegram, we're going with Matrix [3][4]. It's open-source and decentralized, which is great for security and privacy, the Matrix eco-system also has community maintained bridges between Matrix and other platforms, such as WhatsApp, Signal, Telegram and more. Using Matrix we ensure the confidentiality and integrity of the data we collect, and streamline our development process by abstracting the different APIs for the various platforms with one API for the bridges and the Matrix protocol.

Matrix will help us build a data collection tool that's flexible and can handle a lot of data. We'll use Dendrite, which is a Matrix compliant server originally developed by Matrix, as our Matrix server because it's lightweight, efficient, and kept up-to-date by the folks at Element. Some of Dendrite's alternatives are Synapse and Conduit. Both Dendrite and Synapse are being developed and maintained by Element, a company founded by Matrix. For that reason both of these servers are compatible with Matrix and its federation however synapse is the most widely-spread server [3]. While Synapse is written in python, Conduit and Dendrite are written in Rust and Go respectively, this in theory should make their resource footprint smaller while also making them more scalable and performant. As scalability is important for us, as the system will become more and more loaded with each new user, we were motivated to not choose the Synapse server, that left us with Dendrite or Conduit, and we chose Dendrite as we are completely unfamiliar with rust, but some of our team is somewhat familiar with Go.

Matrix's bridging capabilities allow us to connect with other messaging platforms. We'll use bridges to integrate WhatsApp and other services, making it easier for our pollsters and researchers to collect data.

Kubernetes and Docker

We'll use Kubernetes (K8s) and Docker to manage and deploy our applications. K8s is a powerful tool for automating deployments and scaling our platform as needed, while Docker makes it easy to package our applications into portable containers.

Since some of the technologies we're using already have Docker images and resources, we'll be able to leverage them and streamline our development process. This will save us time and effort, and allow us to focus on building the core functionality of our project.

These technologies offer several advantages. Kubernetes excels at scaling applications, allowing for efficient resource utilization, while Docker containers provide portability, enabling consistent execution across different environments [10] [11]

Docker containers are faster to build and deploy compared to traditional virtual machines, offering improved density and performance. They also provide better scalability, allowing for easy adjustment according to needs. [11]

By leveraging Kubernetes and Docker, we can create a efficient, robust, scalable infrastructure that supports our application's growth and maintainability

PostgreSQL

We're going to use PostgreSQL as our database due to its reliability, data integrity, and ability to handle large datasets. As a mature, open-source database with a large and active community, PostgreSQL is well-suited for our project. It integrates seamlessly with Dendrite, our Matrix server of choice, as it is pre-configured in the official Docker Compose setup.

SQLite, another pre-configured option, is not suitable for our platform because it stores data in a single file, supports only one write operation at a time, and lacks the scalability needed for high-volume traffic. Additionally, SQLite does not support remote database access, which is essential for our web application. These limitations make SQLite unsuitable for applications requiring concurrent write operations, high scalability, and remote access capabilities, all of which are crucial for our multi-user messaging platform.

Recent research has highlighted several advantages of PostgreSQL that further support our choice. PostgreSQL outperforms MongoDB in most spatio-temporal queries, with an average speedup of 2.1 across all tested queries [12]. This performance edge is particularly notable in complex business scenarios and queries involving large datasets. The study also revealed PostgreSQL's superiority in handling real-world business scenarios, especially those involving complex spatio-temporal data such as AIS (Automatic Identification System) maritime information [12]. Furthermore, PostgreSQL supports deployment across multiple nodes, ensuring redundancy and scalability for high-traffic applications [12].

While the research identified a minor limitation—PostgreSQL performs slightly worse than MongoDB in specific polygon intersection queries [12] - this drawback is outweighed by PostgreSQL's overall strengths. Its reliability, multi-user support, efficient query handling, and robust indexing capabilities make it an ideal choice for our high-traffic application that requires effective data management and scalability.

Flask

For the backend of our dashboard, we will utilize Flask, a lightweight and flexible microframework for web development in Python. Flask is part of the category of micro-frameworks, which typically have minimal dependencies on external libraries, making it lightweight and efficient. [2] Flask's simplicity and ease of use make it a suitable choice for our project, allowing us to create API endpoints to serve data to the frontend and handle user interactions efficiently. Despite its simplicity, Flask can be used to build robust and scalable backend systems. It offers support for secure cookies, enhancing the security of user sessions. It’s compatible with Google App Engine, providing flexibility in deployment options. [2]

Its minimalist design and extensive documentation make it an ideal candidate for rapid prototyping and development. Flask's versatility and extensive ecosystem of extensions provide the necessary tools to build a robust and scalable backend for our dashboard.

Streamlit

To streamline the development of our dashboard interface, we will utilize Streamlit, a powerful Python-based open-source framework specifically designed for machine learning and data science applications. Streamlit's straightforward API and focus on rapid prototyping make it an ideal choice for creating interactive dashboards with minimal code, allowing developers to quickly build data-driven applications [5]. Its ability to seamlessly integrate with popular Python libraries like Pandas and Matplotlib facilitates efficient data visualization and manipulation, enabling the creation of intuitive and user-friendly interfaces [5].

Streamlit's simplicity and focus on user experience make it a suitable candidate for developing an interactive and user-friendly dashboard interface. The framework's automatic updates and easy deployment options streamline the development process, allowing for quick iterations and collaboration [5]. Streamlit's ability to transform complex algorithms and data visualizations into interactive web applications aligns well with our project goals, as demonstrated by its successful implementation in creating a loan prediction web application[5].

However, it's important to note that Streamlit may have performance limitations for very large-scale applications or complex real-time data processing, and its simplicity can sometimes limit advanced customization options compared to more comprehensive web frameworks. Additionally, Streamlit's primary focus on single-page applications may pose constraints for more complex multi-page dashboards. Despite these potential drawbacks, Streamlit's strengths in rapid development, ease of use, and integration with data science tools make it a compelling choice for our dashboard interface, enabling us to quickly transform our data analysis and machine learning models into an interactive and user-friendly web application.

HeBERT/AlephBERT

HeBERT and AlephBERT are language-specific models specifically trained on Hebrew data, which allows them to better understand the nuances and complexities of Hebrew text. This makes them highly effective for tasks like anonymization, where recognizing the context and meaning of specific words is crucial.
Pretrained on a large Hebrew corpus, they have already learned the patterns and structure of the language, leading to higher accuracy in recognizing and anonymizing entities like names and locations. AlephBERT, in particular, is trained on a larger dataset and vocabulary than any previous Hebrew PLM, including 20.9M sentences from Oscar, 71.5M sentences from Twitter, and 6.3M sentences from Wikipedia. [1]

As BERT-based models, they are bidirectional, meaning they read the entire context of a word in a sentence, making them adept at identifying named entities and sensitive data that require anonymization. AlephBERT has shown substantial improvements on essential tasks in the Hebrew NLP pipeline, including Named Entity Recognition (NER), which is crucial for anonymization [1]

The reliance on Twitter data (72% of sentences) in AlephBERT's training set may introduce biases towards informal language use [1], which can give us good results in the anonymization of private and group chats.

However, their limitation lies in being specifically trained for Hebrew, meaning they are not suitable for anonymizing non-Hebrew text or multilingual datasets, which can be a constraint if the data involves multiple languages.

In addition, resource scarcity in Hebrew NLP may still limit the model's performance compared to PLMs for resource-rich languages.

The reason we picked this tool is because we tested anonymization using LLMs, we got good results, and we believe we will get better results with LLMs that support hebrew.

Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a framework that combines retrieval-based and generative approaches to enhance the performance of models in tasks like question answering, summarization, and anonymization. RAG integrates an external retrieval system that fetches relevant information from large corpora or knowledge bases, which is then used by a generative model to produce informed responses. This hybrid approach is particularly effective for tasks requiring extensive external knowledge, as it bridges gaps in the model's training data and reduces hallucinations by grounding outputs in retrieved content [6].
The RAG process involves indexing, retrieval, and generation. Indexing prepares documents by chunking and encoding them into vector representations, enabling efficient similarity searches. Retrieval selects the most relevant chunks based on semantic similarity to the query, while generation synthesizes this information into coherent responses. This synergy allows RAG to provide continuous updates and integrate domain-specific knowledge, making it highly adaptable for real-world applications [6].

However, challenges persist. The retrieval phase can struggle with precision, occasionally pulling irrelevant or redundant information, while the generation phase may still produce hallucinations or fail to integrate retrieved content cohesively. To address these issues, RAG has evolved through paradigms like Naive RAG, Advanced RAG, and Modular RAG. Advanced RAG improves retrieval quality with pre- and post-retrieval optimizations, while Modular RAG introduces flexible components such as iterative retrieval and adaptive pipelines for better task alignment [6] .

Despite these challenges, RAG's ability to enhance contextual understanding and integrate external knowledge makes it a critical tool for improving model performance in complex tasks like anonymization, where understanding sensitive contexts is essential [6].

Bibliography

1. Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Shaked Greenfeld, Reut Tsarfaty (2021, April 8). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv.org https://doi.org/10.48550/arXiv.1810.04805
2. Fankar Armash Aslam, Hawa Nabeel Mohammed, Prof. P. S. Lokhande (2017, January 13). Efficient Way Of Web Development Using Python And Flask. ijarcs.info https://doi.org/10.26483/ijarcs.v6i2.2434
3. H. Li, Y. Wu, R. Huang, X. Mi, C. Hu and S. Guo, Demystifying Decentralized Matrix Communication Network: Ecosystem and Security. 2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS), Ocean Flower Island, China, 2023, pp. 260-267, doi: 10.1109/ICPADS60453.2023.00047.
4. Schipper, G.C., Seelt, R., & Le-Khac, N. (2021). Forensic analysis of Matrix protocol and Riot.im application. Digit. Investig., 36 Supplement, 301118. https://doi.org/10.1016/j.fsidi.2021.301118
5. Saurabh Shukla, Arushi Maheshwari, Prashant Johri (March 2022). Comparative Analysis of Ml Algorithms & Stream Lit Web Application. ieeexplore.ieee.org 10.1109/ICAC3N53548.2021.9725496
6. Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, Haofen Wang (2023, Dec 18). Retrieval-Augmented Generation for Large Language Models: A Survey. ArXiv.org https://doi.org/10.48550/arXiv.2312.10997
7. Amazon, mturk, available at: https://www.mturk.com/help (Accessed: 09 December 2024).
8. The National Internet Observatory, available at: https://nationalinternetobservatory.org/faq.html (Accessed: 09 December 2024).
9. Brandwatch, available at: https://www.brandwatch.com/ (Accessed: 09 December 2024).
10. Md. Shazibul Islam Shamim, Farzana Ahamed Bhuiyan, Akond Rahman (2020, Oct 21) XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices. ieeexplore.ieee.org 10.1109/SecDev45635.2020.00025
11. Rad BB, Bhatti HJ, Ahmadi M. An introduction to docker and analysis of its performance. International Journal of Computer Science and Network Security (IJCSNS). 2017 Mar 1;17(3):228
12. Makris, A., Tserpes, K., Spiliopoulos, G. *et al.* MongoDB Vs PostgreSQL: A comparative study on performance aspects. *Geoinformatica* **25**, 243–268 (2021). https://doi.org/10.1007/s10707-020-00407-w

Requirements and Initial Specification

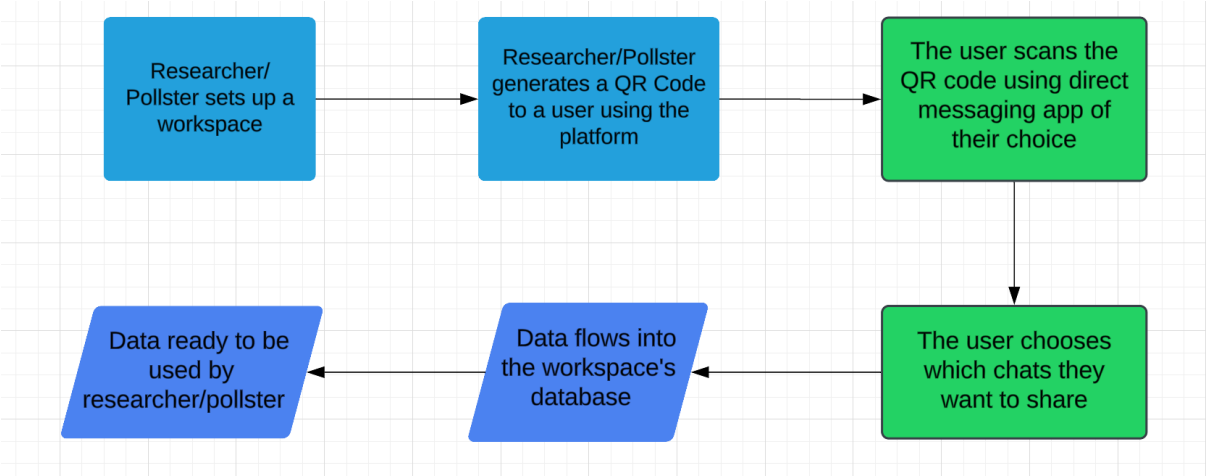
Platform's goals

To allow researchers and pollsters to collect data in Hebrew from private chats (taken from direct messaging applications such as Whatsapp, Telegram and Signal) in order to perform researches and polls based on collected data.

1. The researcher/pollster will be able to easily interact with the platform and to allow users to connect and send data.
2. The data will be anonymized and protected.
3. The researcher/pollster will be able oversee the activity of the platform and data flow.

Requirements Research Summary

Business Process



Functional Requirements

- The platform will be able to connect to the customer's database.
- The platform will generate a QR code for users to connect their apps.
- The platform will set up bridges that connect to the different apps.
- The users will be able to choose which chats they want to share.
- The platform will pull data from the apps into the chosen database.
- The platform will anonymize the data.
- The customer will be able to manage the bridges and users' data.
- The customer will be able to view the users' data and statistics.

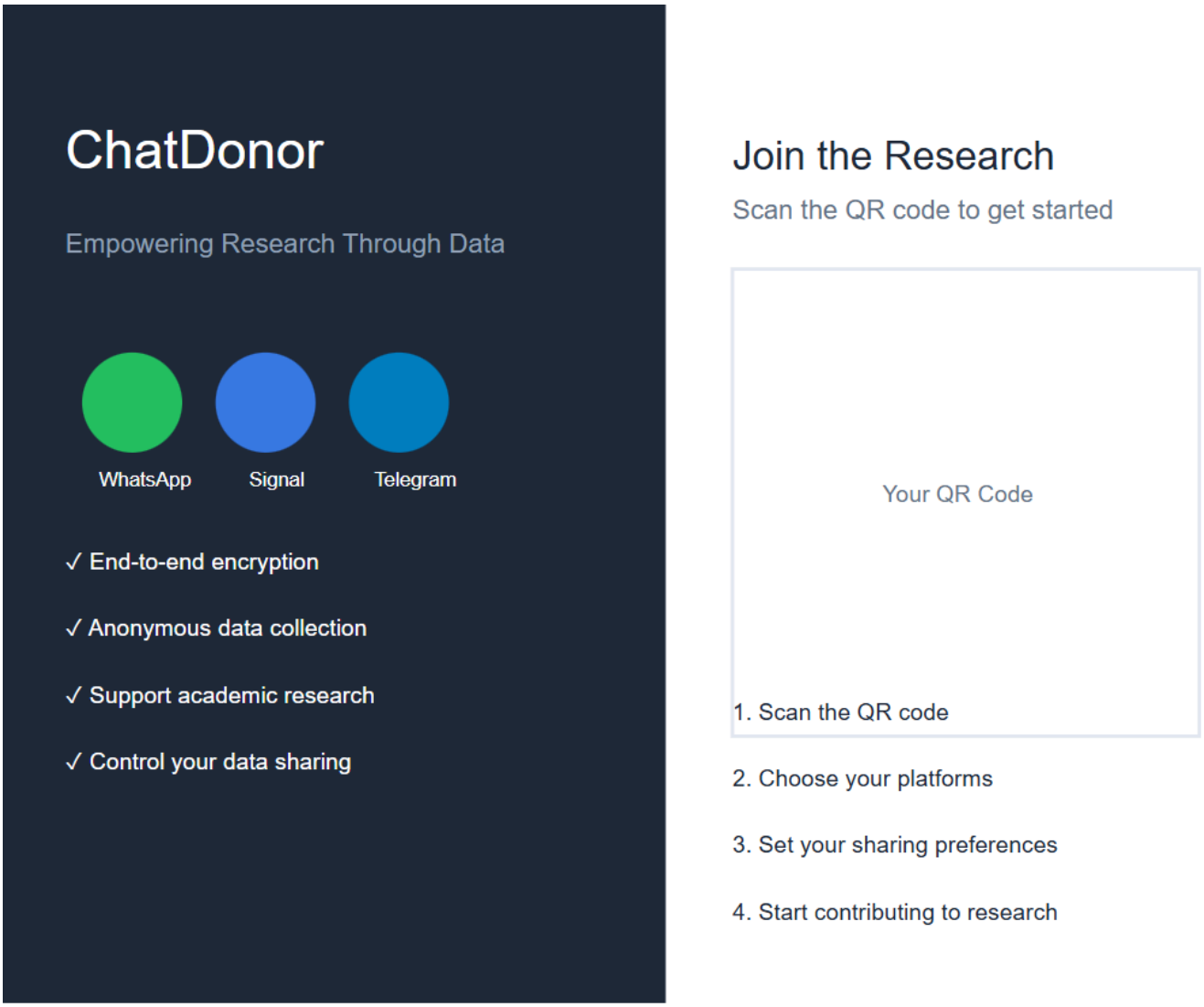
Non-Functional Requirements

The dimensions of a poll panel are subject to alteration based on the polling method employed and the extent to which it intrudes upon privacy; however, it is guaranteed that no

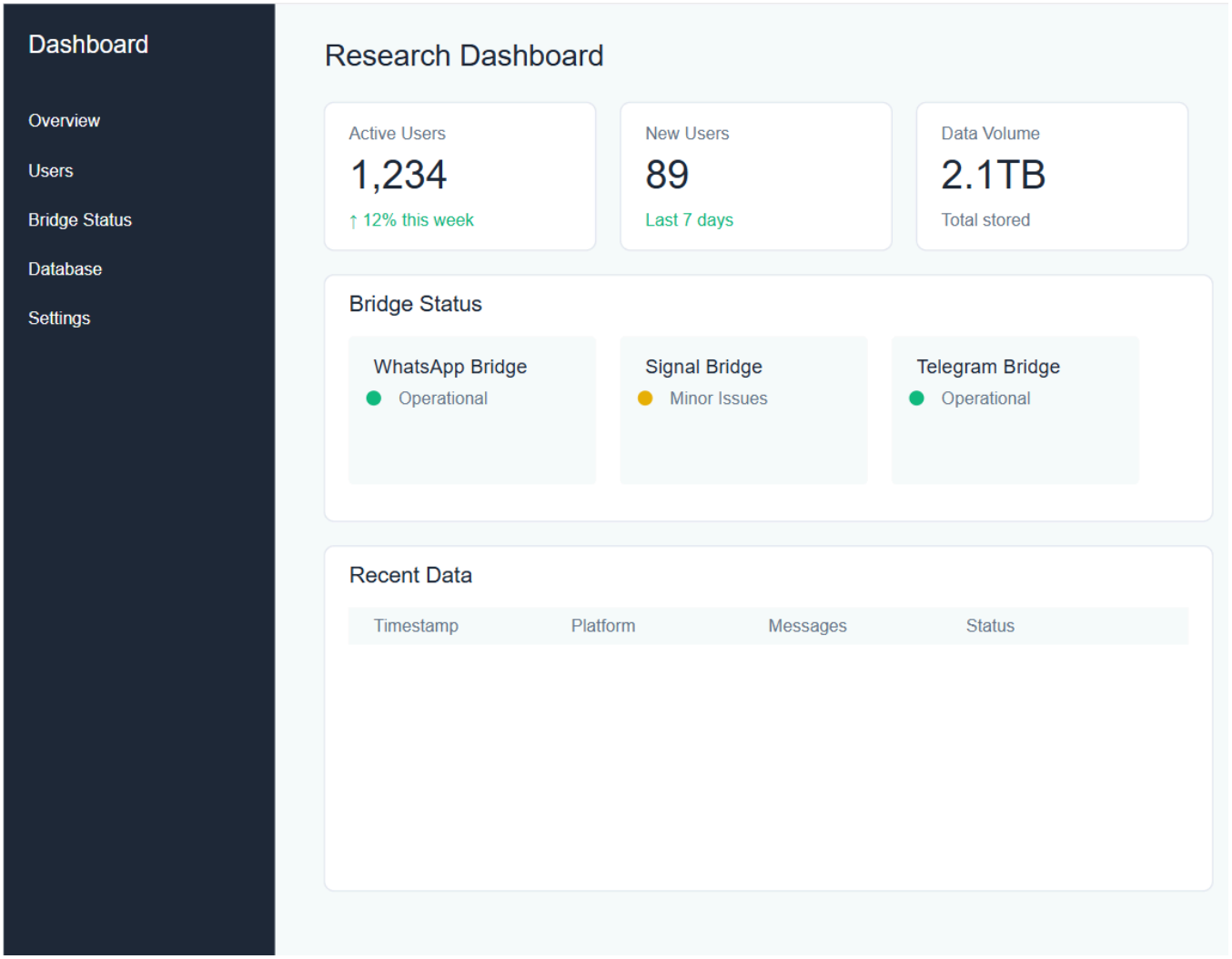
panel will comprise fewer than 50 participants. In light of these considerations, we desire that our system be capable of accommodating up to 200 participants, at the very least. Our system is intended to be deployed on a Kubernetes cluster by our users. This deployment architecture enables horizontal scaling, allowing the system to adapt to varying traffic levels. Through the implementation of Kubernetes, our users can establish redundant instances of each service, thereby mitigating potential downtime associated with individual instances. Backups should occur daily on the PostgreSQL instance to ensure that there is always a recent and reliable copy of the database in case of data loss or corruption. These backups should be automated and incremental, meaning that only the changes made since the last backup are copied. This approach minimizes the time and resources required for the backup process. As we are developing a system that is meant to be used on cloud providers such as Google Cloud Platform, a managed DB backup is available with easy configuration.

Interface Design

User's interface:



Researcher’s dashboard:



Planned Deliverables

Creating A multi-Platform System

After establishing a valid connection with Whatsapp and our platform it is intended to expand its compatibility with other text messaging platforms such as: Telegram, Signal and discord. By doing that the platform will be available for many users across various platforms.

POC

At the end of the semester a proof of concept will take place where a virtual machine will run a demonstration of how a basic version of the system is going to work. It is planned to display a system that saves chat messages from a single donor, it is only a small demonstration that shows a functional system and how it is supposed to work.

Users Anonymization

Creating an anonymous platform is a key to users agreement to donate chats messages. subsequently to the proof of concept anonymization will be implemented on users in order to make sure their identity remains anonymous.

WEB Interface

The product at the end of the process is intended to be a web application where both researchers and donors will sign into. The researchers will have a separate interface from the donors, that interface will display a dashboard of conversations donated by anonymous users. The users will sign up to the system, then their chat history will be uploaded to the system. After the initial sign up the users will continue chatting through their preferred device while their text messages are transferred anonymously to the database.

collaboration with survey company

By the end of the academic year the platform should be functional to a scale of users. The platform will be shown to a survey company. This company will conduct a survey and test the platform public perception in order to see how people respond to the platform and if they trust it.

Challenges and Risks

Technological Challenges And Risks

In order to create a well functioning platform we are tasked with figuring and linking various components, such as setting up matrix servers and using matrix bridges and connecting to different platforms. We learn each of those components without any prior knowledge which could lead to setbacks and errors in our development.

Maintaining Scalable Database

The main purpose of our platform is to assist researchers by collecting data from social networks, that data is going to be stored on a database. We aren't sure yet how many users our database could handle, in other words one of our main challenges is to make sure that the database is scalable, in order to do so we must test our platform on large group of users and understand the database limits and decide how to handle a situation where the number of users and chats is increasing and how the platform keeps working while containing more chats through time.

Maintaining Data Integrity

Conducting research on data requires valid data which hasn't changed and came from a valid source. It is an important challenge to make sure that chats are being transferred to the database while preserving emojis and special characters, and It is important to maintain it across all platforms. Another challenge is inconsistency where there are mismatches in timestamps and in user's identities.

Anonymization

Anonymization is key to the platform's success since users won't agree to donate chats without knowing that their identity is kept unknown to anyone. To this day there hasn't been a Hebrew anonymization model that proved sufficient. It is a great challenge to find one and use it appropriately, without succeeding that part reaching a large audience isn't possible for our platform.

Adaptation to Hebrew

The primary users of our platform are mainly seeking Hebrew-speaking participants, which means our platform must support Hebrew. To ensure data integrity and anonymization, we need to adapt the tools we plan to use to accommodate the Hebrew language.

Tools and Methods

- Dendrite
- Mautrix Bridges
- Kubernetes
- Docker
- PostgreSQL
- Flask/Streamlit
- HeBERT/AlephBERT
- Retrieval-Augmented Generation

Timeline

December 2024:

- Dec 12: Literature review
- Dec 20: Implementation of Matrix bridge for WhatsApp (single user)
- Dec 25: Data processing and database connection
- Dec 30: Migration to k8s/swarm

January 2025:

- Jan 10: Setting up multiple bridges connecting in parallel to the same database
- Jan 10: Expanding support for Telegram and Signal
- Jan 16: POC (Proof of Concept) completion

March 2025:

- Mar 24: Hebrew text anonymization
- Mar 28: Integration with authentication service and user management

April 2025:

- Apr 10: Basic dashboard and web UI implementation

June 2025:

- Jun 1: Testing phase
- Jun 1: Collaboration with survey company

Team Responsibilities

Matrix bridge for whatsapp	Amit, Roey
Data processing and database connection	Ron, Eran
Migration to K8s and Swarm	Roey, Amit
Multiple bridges	Roey, Amit
Connection Telegram and Signal	The whole group
Anonymization	Ron, Eran
Auth service and user management	Roey, Amit
Dashboard and web UI	Ron, Eran
Testing	The whole group