Demystifying Data Collection on Facebook

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Jack Schefer

with

Stephen Shamaiengar

November 11, 2020

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Jack Schefer

Technical advisors: Daniel Graham, Department of Computer Science
Luther Tychonievich, Department of Computer Science

Demystifying Data Collection on Facebook

Jack Schefer
js7ke@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Stephen Shamaiengar
sas7dd@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

ABSTRACT

Modern social media companies profit by selling targeted advertisements powered by data harvested from their users. In addition to profile information and intentionally published posts and images, Facebook and other platforms covertly collect data on users' interactions on websites and apps. To enhance data collection, these platforms take steps to maximize user engagement, and their efforts have been shown to contribute to societal problems such as misinformation, political polarization, and mental health issues. The lack of public awareness of these practices prevents social media companies from being held accountable for their exploitative actions.

We propose an online educational platform to help social media consumers understand what type of data gets collected about them online and how it can be used. By combining a mock social media platform with real-time insights into what is collected behind the scenes, we aim to improve the general public's awareness of these processes. This will empower users to be more mindful of their personal data, to combat the pernicious effects of social media.

ACM Reference Format:
Jack Schefer and Stephen Shamaiengar. 2020. Demystifying Data Collection on Facebook.

INTRODUCTION

In the 2016 U.S. presidential election, Donald Trump's upset win in the electoral college came down to a margin of 77,744 votes in Michigan, Pennsylvania, and Wisconsin, representing six hundredths of a percent of all votes cast [1]. The campaign's secret weapon was Facebook advertising, enabling him to win each of these battleground states by margins less than 1 percent and to get the 46 electoral votes he needed to win the election [21]. The incredible power of Facebook's advertising offerings is what drives its economic success and its ability to influence society. In 2018, Facebook reported revenue of $55.8 billion, of which $55.0 billion, or 98%, came from selling advertisements [12]. Other large technology companies are no different. Google, a subsidiary of Alphabet, reported that $116.5 billion of their $136.6 billion total revenue in 2018 came from their advertising business [11].

Corporations like Facebook and Google are members of a system that one scholar has called "surveillance capitalism" [24]. In this system, online platforms extract behavioral data from users through any means possible, and analyze that data to generate predictions of user behavior that are sold to advertisers. Data collection occurs primarily through free services provided to users. Companies go to great lengths to expand the offering of these services, such as by developing or acquiring new products (like Facebook's acquisitions of Instagram and WhatsApp, or Google's acquisition of FitBit) [24]. Through partnerships with external sites, these companies can even collect data on user behavior outside of their own domains using third-party HTTP cookies [16]. The immense amount of user data flowing into these companies is what drives the targeted advertising industry.

Economic researchers have analyzed the targeted advertising business model and concluded that the better an advertiser can analyze and predict user interests, the higher per-view revenue they will achieve [15]. Thus, a profit-motivated corporation has incentives to produce the most accurate models of human behavior possible. Building these models requires an immense amount of user data, which social media platforms have at their fingertips. Platforms can track how long each post stays on screen, analyze each image and video the user watches, and correlate this information with other users around the world with similar interests. To enhance collection of these data, platforms like Facebook take steps to maximize user engagement, creating features like infinite scrolling on the News Feed and the like button [18].

While economic incentives drive targeting on the platform side, social scientists have repeatedly found that users are averse to intrusive data collection practices. One research team found that 66% of users do not want advertisements tailored to their interests and that consumers "increasingly refuse to disclose sensitive information online" [15]. In another study, 79% of participants identified as "nervous" regarding online data collection, and researchers noted a "discrepancy between the practices of platforms and the users' normative expectations" [14]. Online consumers are uncomfortable with social media data harvesting without even knowing its full extent.

Perceived discomfort is only the beginning of the dangers social media platforms pose. Excessive social media use has been linked to poor job performance [23]. Particularly in youth and young adults, social media has been classified as addictive and contributing to lower self-esteem, life satisfaction, and general motivation levels [8, 22]. Since Facebook became a worldwide phenomenon, rates of depression and suicide among adolescents have steeply risen, attributable in part to social media usage [18]. Social media platforms have also been used by terrorist movements around the world to recruit and radicalize new members [20], and to promote racial violence and genocide in countries like Myanmar [17]. The Facebook platform promotes polarization by design, creating echo chambers where constant interaction with like-minded peers further cements preconceived biases [19, 21].

When it comes to data collection practices, Facebook deserves special attention because of the immense power it has through its user base. Outside of those operating only in China, Facebook owns four of the five largest global social media platforms: Facebook, WhatsApp, Facebook Messenger, and Instagram [21]. The immense amount of data that is mined through these platforms powers Facebook's advertising services, which enable advertisers to precisely target ads based on factors such as location, age, race, religion, and political beliefs, to reach groups as small as 20 people [21]. These are valuable tools for small businesses and political campaigns alike; nationalist politicians — including Donald Trump in 2016, Narendra Modi, and Rodrigo Duterte — relied primarily on Facebook for the advertising and outreach that led to their electoral victories [21].

Still, some of these problems may not be felt on the level of individual users, so why would they care about the collection of their data on Facebook? First, the very nature of data collection raises privacy concerns. Facebook has a troubled past in this regard, with lapses like the Cambridge Analytica scandal, in which user data was improperly mined and given to political campaigns. If one does not care about preserving the privacy of their data, they might care that their data can be used to manipulate their behavior. Many features specifically exploit behavioral tendencies to trigger or influence certain behaviors in the future; for example, a push notification alerting a user to a friend's activity encourages the user to get back on the app [18]. Facebook itself has published research documenting its ability to affect moods and even voter turnout simply by adjusting what shows up in users' News Feeds [3, 7]. Additionally, as stated in the privacy policy, the use of the platform not only shares a user's own data, but their connections' data, too [6, 21]. So, even if one's friends and family on Facebook are not individually affected, one's usage could embroil them in the aforementioned problems. Finally, users might care that they are producing and giving away data for free when there are alternatives. Scholars and technologists have proposed an inverted model in which the act of generating data (e.g. browsing Facebook) is considered labor, so users are compensated for and given full control of that data [2].

Today, most users are not aware of how much data Facebook collects on them because most collection occurs behind the scenes. According to its data policy, Facebook collects data about user devices, including "operating system, hardware and software versions, battery level, signal strength, available storage space, browser type, app and file names and types, and plugins"; data about user activities on third-party websites that use Facebook's business tools, including "websites you visit, purchases you make, the ads you see, and how you use their services"; and data about user interactions with other entities on the platform, including "people, Pages, accounts, hashtags and groups you are connected to" [6]. Unlike profile information, comments, likes, and other public activity, users cannot see in real time exactly what Facebook is collecting for these kinds of data. Having a way to visualize this would increase user awareness of the data collection overall, which could encourage them to think consciously about their Facebook usage and even reduce it to mitigate the aforementioned problems. Therefore, we propose a software tool to demystify the data collection practices of Facebook.

RELATED WORK

Currently, there are limited ways for users to learn what data is collected about them on Facebook. The Facebook Data Policy is the source of truth, but it is vague and hard to understand for many users. For example, on the use of data for personalization, the policy appears to understate the role of targeted advertising. Other literature available online, such as articles by reputable news sources like *The New York Times* [4] or *Harvard Business Review* [13], can give more contextual insights, but these take more effort to find and may cost money to access. Additionally, this kind of mass messaging about data collection and privacy threats is likely to be less effective than individual-focused approaches because of the Third-Person Effect. This effect is a psychological phenomenon by which "people tend to believe that others are more likely to be impacted by privacy threats" than themselves [5].

Considering individualized methods, one extreme is monitoring raw network activity and cookie usage while browsing Facebook, since all data collected on the client side necessarily must be sent back to Facebook's servers. This approach indicates exactly what data is being collected at any given time, but it falls short due to the need for technical know-how to access this data; the difficulty of deciphering the data after encryption and ever-changing obfuscation by Facebook; and the challenge of identifying what data is specifically related to one's usage or actions rather than to simply loading posts and images. Simpler solutions to this problem are provided by Facebook itself, but they are also limited. Facebook provides a data download tool accessible through privacy settings, enabling users to download all data associated with their accounts. The output is an archive of files that, while accurate and complete, is overwhelming and hard to interpret, especially since the data cannot be easily correlated to a user's specific activities on the platform. Another tool gives users visibility into and control over off-Facebook activity data (e.g. visits and purchases on third-party sites), but it has similar deficits and only covers a small subset of all data collected.

USE CASE

The primary audiences for a tool illustrating Facebook data collection practices are current social media users and youth in particular. Current Facebook users must be critical when consuming information and understand how their usage affects the posts they see and even their behavior. Youth and parents need to be made aware of potentially addictive platforms so they can be properly educated in avoidance and recognition of addiction symptoms.

We consider several approaches for how to best illustrate data collection to users, but they generally fall into two categories: overlaying real-time information on Facebook or creating a separate, sandbox platform.

In the first approach, as users browse the Facebook News Feed, real-time information about what data is getting tracked would appear beside the feed to promote awareness of data collection, likely using a browser extension. This would achieve the goal of real-time, individualized illustration of tracking metrics. Additionally, since there is no separate platform, the barriers to user adoption would be low. However, one challenge would be technical feasibility. Though browser extensions access site data after decryption, proprietary

obfuscation techniques would prevent strict monitoring of network calls to observe data. Instead, the extension would have to track behavior on its own, potentially misrepresenting what is actually getting tracked. Showing these educated guesses of collection metrics on top of Facebook itself may give a false impression of what Facebook actually does and could provide a false sense of security if incomplete in scope. Further, a browser extension is only useful to those that access the site on the web platform. Anecdotal user surveys suggest that a significant portion of the population only accesses social media platforms on mobile devices.

The mock platform approach provides its own benefits and challenges. Designing an entire site gives full control over what data gets collected, how it is shown, and the degree of transparency users receive. Furthermore, while designers would still be making educated guesses on collection methods, it is more obviously illustrative in the context of a separate platform than when overlaid on Facebook itself. Lastly, a separate platform more naturally extends to mobile platforms if desired. However, barriers to adoption would present an issue, as it may be hard to convince users to join a new platform. Designing a new platform would also require more development work, though not as much as one might initially expect. For the browser extension approach to reveal any aggregated user data across sessions, it would require much of the same backend logging and analysis as the full platform would.

Both ideas present some technical limitations. First, the extent to which data collection practices are shown will be limited by designers' knowledge of real platforms, since both require tool designers to decide what metrics to track and visualize. Additionally, neither approach will allow visualization of third-party tracking on external sites. That information is stored by Facebook but not directly shown on the News Feed so extensions wouldn't get access, and no new mock third-party tracking pixel would be able to match Facebook's in terms of usage and reach.

In deciding between these approaches, it is important to consider the goal of the tool: to educate and inform users about data harvesting techniques. The primary benefit of the extension approach is to provide a constant reminder of online tracking, whereas the mock platform approach is more amenable to a tutorial-style learning experience. Additionally, while both approaches are limited by designer knowledge of tracking mechanisms, incompleteness is more appropriate in the context of a separate platform than when overlaid on a real site. Also, with full control of data stored and aggregation algorithms, the mock platform is more extensible for new tracking methods. Lastly, presuming lower barriers to adoption for the browser extension overlooks mobile-only users limiting the potential user base. The mock platform is more naturally extensible to a mobile application, and in the context of an interactive, tutorial-style learning experience, it makes more sense to require browser usage.

PROCEDURE

As an online platform, users of the proposed tool would navigate to the website in their browser. The interface (illustrated in Figure 2) would include a content feed on the left, resembling the Facebook News Feed, and a separate data feed on the right. Users would then be instructed to browse the platform as they would any other social media site. The data feed would initially be empty, but would fill up with insights of data being collected as users interact with the platform. A user would browse the content feed just like the Facebook News Feed — including viewing, liking, and commenting on posts and ads — and would be able to see all of the data being collected from those interactions in the data feed in real time. For best results, a user could visit the platform multiple times. Users would have the option to log in through a modal so that data can be aggregated and displayed through multiple sessions. If logged in, additional data would show up in the data feed, including asynchronously generated aggregations such as associations to certain topics, interests, or groups.

It is important to note that each user would see their own sandbox environment when visiting the tool. The profiles present in the system would be made up of two groups: real site visitors that login and "bot" accounts that are run by tool maintainers. Site visitor accounts would not be able to create top-level posts and would not be able to view the comments that other site visitors make. This decision is further discussed later on, but has security and performance benefits with few user experience drawbacks given the strictly pedagogical goal of the tool.

SYSTEM DESIGN

General Architecture

Figure 1 contains an overview of the major system components as well as an indication of how data will flow in the application.

The first pathway for data flow follows a client-server model and is used for most aspects of the platform. When a user wants to load posts on the feed, login, or create an account, these requests are sent to the web server, which forwards them to the appropriate services, which interact with the data stores to fulfill the request and return a result.

All data collection and logging will go through a different pathway because of the greater amount of data and the one-way nature of communication. When metrics are collected in the frontend client, they will immediately be reflected in the data feed before getting logged to the server. In the backend, rather than being immediately written to the database through service calls, they will be buffered in event queues and handled asynchronously. This allows client requests to be fulfilled successfully as soon as events are enqueued rather than waiting for persistence in the database. Periodic batch jobs can then consume these queues in bulk for efficiency, to train and evaluate the kinds of models that Facebook would typically use to power targeted advertising.

Frontend

The frontend client of the application will be split into two logical sections, shown side-by-side. Figure 2 illustrates what this will look like.

Content Feed. The first important section is the feed of content replicating what users see on Facebook. Just as on the Facebook News Feed, posts, images, and advertisements are shown in a card style. Content is ordered by a backend feed algorithm based on predicted user interest. As the user scrolls through the posts, new ones will be loaded in the background to reproduce the infinite scrolling

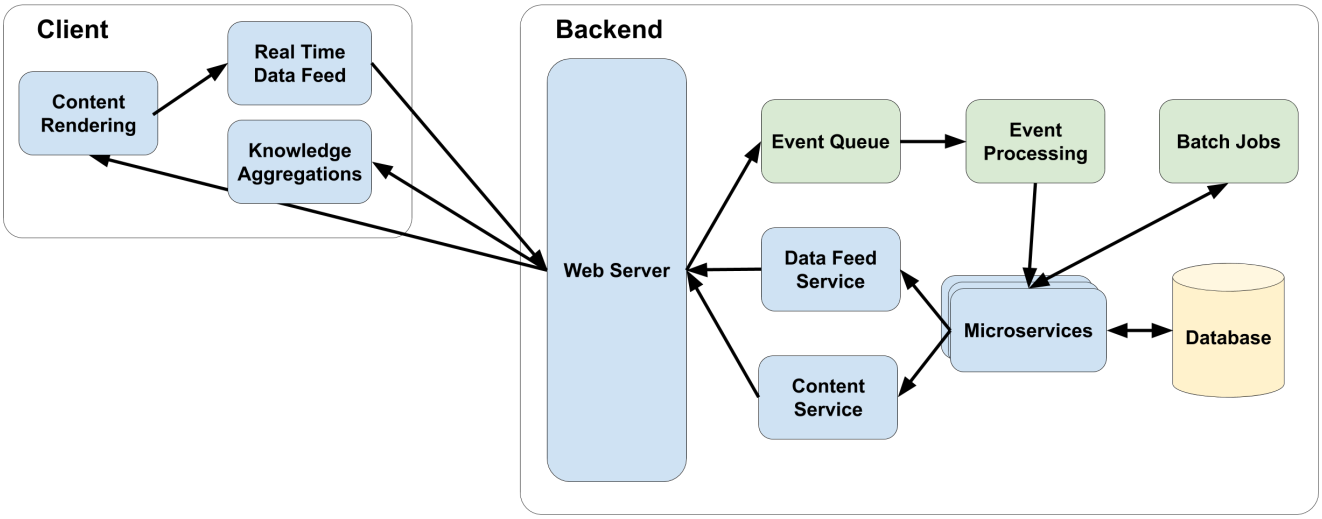

Figure 1: Major system components and illustration of data flow.

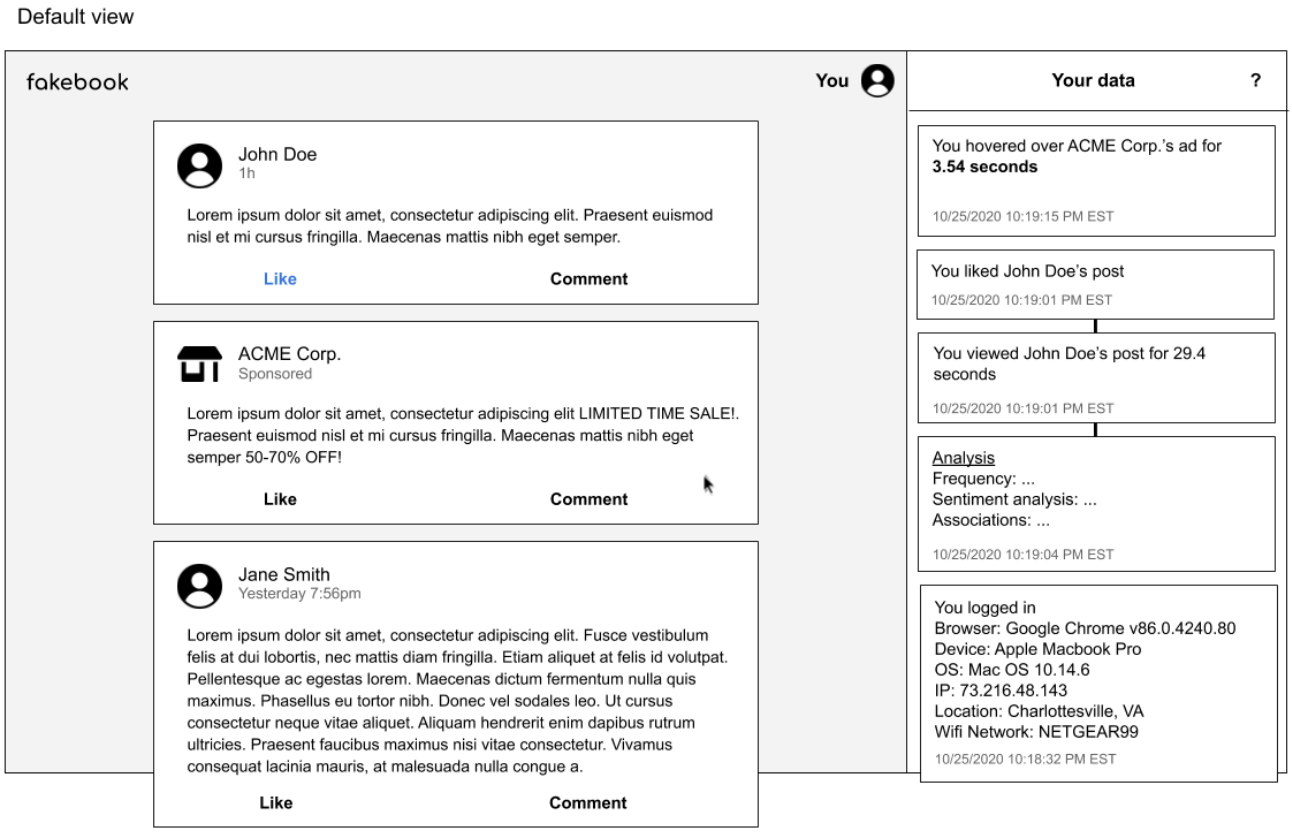

Figure 2: User interface diagram illustrating content feed and real-time data feed.

behavior of real social media platforms. On specific user actions, such as liking a post, commenting, or loading the application from a new device, events will be triggered to log data being collected.

Data Feed. Beside the content will be a section illustrating what the platform is learning. The real time feed will provide immediate feedback as user data is collected and will also forward these events on to the backend for persistence and aggregation. When a user is logged in, there will be another pane to the Data Feed showing aggregated information that the platform has learned over time. These types of aggregations include sentiment analysis of posts viewed, predicted topics or keywords of interest, geographic or device data, and relations to other users. Aggregation data is the output of periodic batch processing, but is retrieved by the frontend via the synchronous service path of data flow.

Prototype. A prototype implementation of the frontend of the application has been implemented using React. This prototype attempts to geo-locate the user and reports statistics on posts viewed, posts liked, and device information. These statistics populate in the Data Feed but are not persisted in any meaningful way. More details on the prototype can be found in the Appendices.

Services

The backend of the tool will make extensive use of the microservice architecture, in which reads and writes to data stores are done through many services that are loosely coupled and can be independently developed and deployed. The following list represents the anticipated services to be built in the application along with some data characteristics to inform storage requirements.

Profile Service. This microservice manages user accounts, profiles, the concept of "friends", and all user sessions. Both logins and anonymous one-time sessions will be handled here. This data is characterized by significantly more reads than writes. Reads are also highly focused on bot profiles whose data will be pulled whenever posts are loaded.

Post Service. This microservice is the source of truth for all posts, comments, and likes. This data is also characterized by a high read load with fewer writes.

Feed Service. This microservice is responsible for recommending posts to be shown in the feed of any given user session. This service uses the outputs of periodic batch analysis jobs and also ensures duplicate content is not shown during infinite scrolling in the same user session. The cache of posts that have been shown during each session will be written very frequently, but this data has low consistency requirements because infrequently showing the same post twice in a session will not negatively impact user experience.

Tracking Service. This microservice manages the logging of all user data that gets fed into batch processing for analysis. This data has a very high write load, but the buffering of writes using event queues should help with this process. Reads of this data are characterized by short bursts whenever the batch analysis runs.

Knowledge Service. This service gives access to the aggregated information the platform has learned about each user over time. When batch processing analyzes and models user data, the output is persisted here. This data is characterized by write loads that occur in short bursts, and fairly infrequent reads.

Content Service. This higher level service contacts the Feed, Post, and Profile services to provide all of the data necessary to render the content feed. This is the entry point used by the frontend on initial page load as well as for on-demand post loading during infinite scrolling.

Data Feed Service. This higher level service contacts the Knowledge service and Tracking Service as necessary to provide data shown on the data feed. For logged-in users this includes both the knowledge aggregations as well as the previous session's real-time feed contents.

Storage

As explained above, different services require different types of storage for the data they manage. All services will use relational databases — like MySQL, PostgreSQL, or Oracle Database — for full persistence of data. Services with high read-write ratios, such as the Profile and Post services, can take advantage of read locality with in-memory caches — such as Redis or Memcached — to improve performance if necessary. On the other hand, services with more writes than reads can use eventual consistency to reduce latency (if the storage needs are increased and replication is used). Services with lower reads and writes overall can use the relational databases alone.

Event Queues and Asynchronous Processing

All user interaction data collected by the frontend must eventually get persisted for later analysis. However, this represents an extraordinary amount of data given the anticipated collection metrics. Distributed event queues — such as Kafka, RabbitMQ, or Amazon SQS — will be used to buffer data collection writes so they can be handled one-by-one asynchronously. Daemon processes can then subscribe to these queues and handle data as it comes in. These daemons will then call the appropriate Tracking Service endpoint so user data is properly persisted.

Batch Job Processing

This component of the system represents the meat of what powers targeted advertising: predictive modeling of user behavior. This modeling occurs through batch jobs that asynchronously process large amounts of collected data. In Facebook, this processing may include the building of indices (including the social graph), or the training, testing, and evaluation of machine learning models. For this tool, the batch jobs will involve the creation of basic models and the use of external APIs. For example, sentiment or keyword analysis APIs can be used to connect text content to topics or categories. Naive predictions of future post engagement can be made by modeling previous post viewing times. This work will be done using distributed data processing tools such as Apache Spark or Hadoop.

Design Decisions

Sandbox Environment. The decision to eliminate interaction between users and only show posts from tool maintainers (through bot accounts) was made for a variety of reasons. First, taking the platform too far towards a real social media site would hinder the educational value of the tool. Minimizing the social aspect of the platform places emphasis on the data collection. This choice also improves the applicability of the tool. With posts generated by tool maintainers and potetially visible to all users, the tool is equally effective with one user as it is with one million, which would not be the case if content came from peer users. Mixing the two approaches (some content generation by maintainers and some by fellow users) would add complexity but little educational value.

Both real site visitors and bot accounts will share the same backend data structures, and the interaction model provides some technical benefits. In terms of security, since site visitors won't be posting, we don't need to store things like names or profile pictures if they have privacy concerns. For performance, in-memory database caching makes significantly more sense when post and profile reads are highly temporally localized to the limited set of bot accounts and their recent posts.

Microservice Architecture. A microservice architecture consists of many independently operated services that each manage reading and writing to storage for a specific logical kind of data, like profiles or posts. This architecture contrasts with a monolithic architecture, in which reads and writes of data go through one large application that provides all of the functionality. We have proposed a microservice architecture for this solution because it offers several advantages. First, as mentioned earlier, the kinds of data present in this platform may have differing storage requirements, such as use of caching and different loads on storage systems. Well-designed microservices inherently provide a separation of concerns by which each service has complete control over its own data store, which naturally supports variation in storage requirements. Microservices are also advantageous for the software development and maintenance process. Since they are by definition loosely coupled, they are ideal for independent development by separate engineers or teams. Since they are also deployed separately, they give easier and more granular control over scale for the application overall; if, say, the Post service is under much higher load than others, it alone can be scaled horizontally (by adding more instances of the microservice). These features of microservices confer similar benefits to testing and debugging. Modules that are highly cohesive and have narrowly defined functionality are easier to test, and they are also easier to debug because they lack the overlap or confusing side effects that could creep into a monolithic architecture.

Event Processing. The use of asynchronous event processing for the persistence of user data was briefly mentioned but deserves more detail. The primary advantage of asynchronous processing is performance. Each tracking metric collected by the frontend will be sent to the server for persistence over HTTP. In this design, logging requests can be fulfilled as soon as the event is enqueued for processing due to the reliability guarantees of distributed queues. The buffered nature of processing also allows it to happen in batches to improve performance. Modern message queue systems allow for multiple consumers for a single event queue, meaning that the daemon processes listening for events can be scaled independently from the rest of the application. Finally, since these daemon processes mainly invoke Tracking Service endpoints, they can be co-located on the same physical machine to reduce the impacts of network latency.

Profile and Session Distinction. The decision was made to separate the idea of a user profile from that of a session. A user profile represents a person, whereas a session represents a person and a specific time period. Categorizing each session separately allows for the expansion of data analysis in the future. In theory, topics of interest could be correlated with the user's session location or which device they are using at the time. The use of sessions also allows for the relatively simple implementation of anonymous users. When accessing the tool without logging in, users can still receive a session token and have their data logged, but it simply won't be linked back to them when they return. Anonymous sessions allow the most privacy-focused users to skip account creation for a slightly less robust experience.

RESULTS

Risk and Cost Analysis

Risks to success. There are a few key risks to the success of the proposed system. Regardless of marketing and costs, these nonfunctional requirements have guided design and should guide future implementation.

First and foremost is security, since the tool cannot be successful unless it is a safe platform to use. Harvesting significant user data is part of the premise of the application but requires strict security enforcement. The interaction model where site visitors do not post is helpful for security purposes. Storing personally identifiable information is not strictly necessary since it will never be shown on another user's feed. The ability to provide anonymous sessions also allows users to opt for a more secure experience, albeit without features like multi-session aggregation of interests and behavior.

The accuracy of the feed and engagement analysis also presents a risk for the success of the tool. To convince users of the power in behavioral analysis algorithms, we must be able to present reasonably accurate predictions of their interests. If this fails, the system may unintentionally give users a false sense of security on platforms like Facebook which is directly contrary to the intended purpose.

The tool also cannot be successful if it is hard to use or uninteresting. Choosing to create an entirely separate platform from Facebook is a risky endeavor. While it provides a more robust educational experience, the application must be intuitive and compelling to capture user attention long enough so they have time to learn.

One aspect of usability comes from the performance of the application. To fulfill the aim of immediate, personalized feedback, the tool has to be fast. This informed significant portions of the design. Collection metrics generated by the content feed flow directly to the data feed before being sent to the server, allowing for immediate feedback in the frontend. Asynchronous processing and persistence of these metrics also should improve the performance of the tool. Finally, the caching mechanisms described can be added as necessary based on performance testing.

Table 1: Estimated initial development costs in developer-weeks. Actual dollar costs will vary by number of developers and compensation given.

Component	Cost
Microservices	10 developer-weeks
Higher-level services	1
Event processing	1
Batch job processing	4
Frontend	1
Testing	3
Content generation	2
Total	**22 developer-weeks**

Development Costs. Our estimates of the initial development costs for this tool are shown in Table 1, given in developer-weeks (1 week of full-time work for 1 developer). We estimate development and deployment of microservices to take the most time (2 developer-weeks per service times 5 services) since they involve implementing the core business logic as well as setting up a large amount of computing, storage, and networking infrastructure. We estimate development of higher level services to be much quicker because they primarily call microservices, so they rely on existing microservice logic and infrastructure. Event processing is estimated to take 1 developer-week because it requires mostly infrastructure work; the logic of reading and writing to the event queues is handled by other components. We estimate implementation of batch job processing to take 4 developer-weeks because it might require significant machine learning and modeling work. The frontend is only estimated to take 1 developer-week because the tool focuses only on replicating the News Feed aspect of Facebook along with a data feed, so there are relatively few components to implement. For reference, our development of the frontend prototype only took about 15 developer-hours, although it isn't feature-complete or tested. Finally, we reserve 3 developer-weeks for thorough testing (unit, regression, integration, and load) of all components, and 2 developer-weeks for the creation of a variety of artificial content (posts, comments, likes, etc.) that will be presented through bot accounts.

Computing Costs. The computing cost estimate in Table 2 is based on the price of an Amazon EC2 `t4g.small` instance (2 vCPU, 2 GB vRAM) for $0.0168 per hour [10]. With an initially small user base, all services and processing jobs could likely be vertically scaled on one, larger cloud VM. However, the system was designed for a larger user base, so horizontal scaling comes into play. For the final estimate, we assume the renting of 10 virtual machines: 5 for microservices and their data stores, 2 for the higher level services, 1 for the event queue, 1 for the batch processing job, and 1 for a load balancer.

Storage Costs. The storage cost estimates in Table 2 are based on the price of Amazon EBS `gp2` storage (general purpose SSD) for $0.10 per GB per month [9], and the costs vary based on number of users. To come to the final estimates, storage needs were broken down into three categories: user profiles, behavioral data, and content like posts and comments. The per-user estimates are then multiplied by the number of users before summing to the numbers in the table.

Table 2: Estimated baseline monthly operating costs in dollars. Costs will increase with performance and storage needs.

Component		Monthly Cost ($)
Compute		122.98
Storage		
	500 users	0.06
	50,000 users	5.78
	1,000,000 users	101.42
Total		
	500 users	**123.04**
	50,000 users	**128.76**
	1,000,000 users	**224.40**

User profile storage depends on a few factors. We estimate each profile takes up around 11 KB of data (1 KB for textual profile information and 10 KB for a small, compressed profile picture). We also make the assumption that a given user only ever accesses the site 100 times, which is a generous upper bound given the educational nature of the tool. This assumption means an additional 3.2 KB per user for session tokens (32 bytes each). These factors sum to 14.2 KB, costing the almost negligible 0.00014 cents per profile per month.

Persisted user data represents a large portion of the storage costs of the application. For estimation purposes, we assume that a data point gets logged about once every two seconds, or 1,800 metrics per hour. We also assume that a given user will only access the site for an hour a month. While this may seem low at first, we believe that the educational value of the tool can be conveyed in less than twelve hours of use total. Thus, if site access is amortized over an entire year it comes out to around 1 hour per month. If each data point is limited to 100 B of data, these metrics come to approximately $0.0001 per user per month.

Content data will vary on the rate at which posts are generated and how long that rate has been maintained. Estimating a rate of 100 posts or comments per day, where each post or comment takes up 1 KB of space, gives 100 KB per day. After holding this post rate for two straight years, this builds up to 71.3 MB of data. This constant does not vary based on the number of users because only maintainers are generating content.

Cost Burdens. The development and operating costs summarized in Tables 1 & 2 raise the question of who will fund such a project. Monetizing the site with advertising runs counter to the site's premise and would create a conflict of interest. Initial funding will likely need to come from a research grant or donation to cover the costs of development. Another option is to become open-source, though this would be more feasible for maintenance than in the early stages. As for other cloud operating costs, donations may be the most viable income source. The presented cost estimates represent the requirements for a fully operational system with a

large user base. At initial release, the application should be able to function on a handful of virtual machines with limited storage.

Broader Impact

The intended impact of this application goes beyond educating Facebook users. Consumers must think critically about their online browsing activity on any site so they can properly assess the risks in using otherwise free services. Promoting technical literacy informs not only the actions of individual users but also the society as a whole and government regulators that hold corporations accountable for their actions. Thus, while heavy social media users may be the target audience, all consumers of the internet would benefit from increased awareness of data collection techniques and the targeted advertising industry.

Even the impacts directed towards Facebook surpass data collection alone. Once the tool clearly demonstrates data collection techniques, it should expose deeper questions of why users see what they see on Facebook, and ultimately, what they truly gain from using the platform. In theory, serious consideration of these questions could help to address global social problems like misinformation, political polarization, nationalism, and radicalism. It could also raise awareness of social media addiction as users begin to understand how platforms learn what kind of posts keep them scrolling. The tool will not be able to relieve the negative psychological effects of excessive Facebook use, but it could begin to raise awareness of them.

The tool also has the potential to raise questions of implicit and algorithmic bias. The tool may be adversely affected by the implicit biases of maintainers, since they are the only stakeholders that can generate posts and comments for others to see. It also has the potential to have algorithmic bias in the way posts are recommended to each user. However, if the reasoning behind such algorithmic biases can be properly illustrated in the tool, this will be a powerful aspect of learning for consumers.

CONCLUSION

Social media platforms in general, and Facebook in particular, contribute to widespread social problems — including misinformation, political polarization, and radicalism — as a result of the approaches they take to maximize user engagement and collection of user data. We proposed a tool that displays a real-time feed of data collection and aggregation alongside a typical content feed, so that users can clearly see the amounts and kinds of data that can be collected on them as they use a platform like Facebook. Consequently, this tool would demystify the kinds of data collection practices that are used by Facebook, encouraging users to think more consciously about their usage of Facebook and other forms of social media. These outcomes are critical in beginning to solve the social problems exacerbated by these technologies and to explore better alternatives.

FUTURE WORK

There are a number of ways this tool could be expanded and improved once initially developed. First, the data collection techniques employed by the application could grow over time. Similarly, more complicated machine learning and natural language processing could be used to improve the algorithms for knowledge aggregation and post recommendation. Lastly, allowing users to step through the process of targeting an advertisement using the platform's data would naturally extend the pedagogical value of the tool by allowing users to understand how their data is exposed to advertisers.

ACKNOWLEDGMENTS

We would like to thank Professors Daniel Graham and Luther Tychonievich for advisement on this project, and Professors Mark Sherriff (Advanced Software Development) and Tom Pinckney (Internet Scale Applications) for teaching us material critical to designing the system.

REFERENCES

[1] 2017. Presidential Election Results: Donald J. Trump Wins. *The New York Times* (Aug. 2017). https://www.nytimes.com/elections/2016/results/president

[2] Imanol Arrieta Ibarra, Leonard Goff, Diego Jiménez Hernández, Jaron Lanier, and E. Glen Weyl. 2017. *Should We Treat Data as Labor? Moving Beyond 'Free'.* SSRN Scholarly Paper ID 3093683. Social Science Research Network, Rochester, NY. https://papers.ssrn.com/abstract=3093683

[3] Robert M. Bond, Christopher J. Fariss, Jason J. Jones, Adam D. I. Kramer, Cameron Marlow, Jaime E. Settle, and James H. Fowler. 2012. A 61-million-person experiment in social influence and political mobilization. *Nature* 489, 7415 (Sept. 2012), 295–298. https://doi.org/10.1038/nature11421 Number: 7415 Publisher: Nature Publishing Group.

[4] Brian Chen. 2018. Are Targeted Ads Stalking You? Here's How to Make Them Stop. *The New York Times* (Aug. 2018). https://www.nytimes.com/2018/08/15/technology/personaltech/stop-targeted-stalker-ads.html

[5] Hongliang Chen and David Atkin. 2020. Understanding third-person perception about Internet privacy risks. *New Media & Society* (2020), 1–19. https://doi.org/10.1177/1461444820902103

[6] Facebook. [n.d.]. Data Policy. https://www.facebook.com/policy.php

[7] Blake Hallinan, Jed R Brubaker, and Casey Fiesler. 2020. Unexpected expectations: Public reaction to the Facebook emotional contagion study. *New Media & Society* 22, 6 (June 2020), 1076–1094. https://doi.org/10.1177/1461444819876944 Publisher: SAGE Publications.

[8] Nazir Hawi and Maya Samaha. 2016. The Relations Among Social Media Addiction, Self-Esteem, and Life Satisfaction in University Students. *Social Science Computer Review* 35, 5 (Aug. 2016), 576–586. https://doi.org/10.1177/0894439316660340

[9] Amazon Inc. [n.d.]. Amazon EBS Pricing. https://aws.amazon.com/ebs/pricing/

[10] Amazon Inc. [n.d.]. Amazon EC2 On-Demand Pricing. https://aws.amazon.com/ec2/pricing/on-demand/

[11] Alphabet Inc. 2020. Alphabet Annual Report 2019. https://abc.xyz/investor/static/pdf/2019_alphabet_annual_report.pdf?cache=c3a4858

[12] Facebook Inc. 2019. Facebook Annual Report 2018. https://s21.q4cdn.com/399680738/files/doc_financials/annual_reports/2018-Annual-Report.pdf

[13] Leslie John, Tami Kim, and Kate Barasz. 2018. Ads That Don't Overstep. *Harvard Business Review* (2018). https://hbr.org/2018/01/ads-that-dont-overstep

[14] Helen Kennedy, Dag Elgesem, and Cristinia Miguel. 2017. On fairness: User perspectives on social media data mining. *Convergence: The International Journal of Research into New Media Technologies* 23, 3 (2017), 270–288. https://doi.org/10.1177/1354856515592507

[15] Henk Kox, Bas Straathof, and Gijsbert Zwart. 2018. Targeted advertising, platform competition, and privacy. *Journal of Economics & Management Strategy* 26, 3 (Sept. 2018), 557–570. https://doi.org/10.1111/jems.12200

[16] Andrew McStay. 2012. I consent: An analysis of the Cookie Directive and its implications for UK behavioral advertising. *New Media & Society* 15, 4 (Sept. 2012), 596–611. https://doi.org/10.1177/1461444812458434

[17] Paul Mozur. 2018. A Genocide Incited on Facebook, With Posts From Myanmar's Military (Published 2018). *The New York Times* (Oct. 2018). https://www.nytimes.com/2018/10/15/technology/myanmar-facebook-genocide.html

[18] Jeff Orlowski. 2020. The Social Dilemma. https://www.netflix.com/title/81254224

[19] Cass Sunstein. 2017. #Republic: Divided Democracy in the Age of Social Media with Cass R. Sunstein. https://www.youtube.com/watch?v=_Uv-IJXVm3c

[20] Robin Thompson. 2011. Radicalization and the Use of Social Media. *Perspectives on Radicalization and Involvement in Terrorism* 4, 4 (2011), 167–190. https://doi.org/10.2307/26463917

[21] Siva Vaidhyanathan. 2018. Antisocial Media: How Facebook Disconnects Us and Undermines Democracy. https://www.youtube.com/watch?v=5IpI6PCPE7c

[22] Liu Yuchen. 2016. From Social Media Uses and Gratifications to Social Media Addiction: A Study of the Abuse of Social Media Among College Students. (2016).
[23] Suzanne Zivnuska, John Carlson, Dawn Carlson, Ranida Harris, and Kenneth Harris. 2019. Social media addiction and social media reactions: The implications for job performance. *The Journal of Social Psychology* 159, 6 (2019), 746–760. https://doi.org/10.1080/00224545.2019.1578725
[24] Shoshana Zuboff. 2019. *The Age of Surveillance Capitalism.* Profile Books.

Appendices

FRONTEND PROTOTYPE

As mentioned earlier, we developed a frontend prototype to demonstrate how data collection would be revealed through a data feed displayed alongside a content feed. This prototype is implemented as a standalone React app and thus only demonstrates frontend features, including filler content and collection of a few kinds of data (post viewing times, device information, geolocation, liking posts). The project is stored in a GitHub repository located at https://github.com/sshamaiengar/mock-platform-ui.