Mid-Report: DP-accurate DAU/MAU Counter Under Deletions

Dev Sanghvi (ds221) and Lazeen Manasia (lm152)
Probabilistic Algorithms and Data Structures

14th November, 2025

1 Problem Description

Modern online services and IoT platforms routinely track distinct daily active users (DAU) and monthly active users (MAU) to monitor growth, retention, and product health. DAU measures the number of unique users active on a given day, while MAU counts users active within a rolling window of days (typically $W = 30$). In practice, activity arrives as a high-volume stream of events, and realistic systems must comply with privacy regulations such as GDPR and CCPA, which grant users a *right to be forgotten*. This means that once a user requests erasure, their contribution must be retroactively removed from all relevant aggregates.

We model user activity as a *turnstile stream* of positive and negative events. A positive event $+u@d$ indicates that user u was active on day d, while a negative event $-u@d$ encodes a deletion request to remove u from the counts for one or more days. The system must continually release DAU and MAU estimates:

$$\mathrm{DAU}(d) = \big|\{u : u \text{ active on day } d\}\big|, \quad \mathrm{MAU}_W(d) = \big|\{u : u \text{ active on some day in } [d - W + 1, d]\}\big|$$

with differential privacy guarantees, meaning that published counts should not reveal information about any single user.

Concrete Example: Consider an app with $N = 10{,}000$ daily users, of which 5% request deletion mid-month. A naive distinct-count system without deletion support would overestimate MAU by hundreds of users, potentially violating privacy laws. Our goal is to maintain accurate, private estimates of DAU/MAU even under such deletion requests.

Real-World Stakes: Accurate, privacy-preserving tracking of DAU and MAU under deletions is critical for large apps that must obey GDPR-style "right to be forgotten" regulations, while still making business decisions based on reliable metrics. Failure to respect deletions can lead to legal penalties and loss of user trust.

The technical challenge arises because classical differential privacy analysis assumes monotone streams (only insertions) or simple counters. Distinct counting under deletions violates standard sensitivity bounds, and naive DP mechanisms can produce biased or overly noisy estimates. Our project explores combining deletion-friendly streaming sketches (Theta) with flippancy-aware, user-level DP mechanisms to yield accurate, privacy-preserving DAU/MAU estimates that respect all deletion requests.

2 Literature Survey

Our project builds on two main pillars from the course: streaming distinct-count estimation and differentially private algorithms under continual observation.

2.1 Streaming distinct counting

Classic work by Flajolet and Martin on probabilistic counting [1] and its later refinement HyperLogLog (HLL) [2] introduced bit-pattern based sketches for estimating the cardinality of large sets from streaming data. These sketches are mergeable (supporting unions across shards) and provide tight relative error guarantees using sublinear memory. However, HLL is essentially append-only: registers store maxima, and there is no native support for removing a user once its contribution has been merged. More recent systems work, such as Apache DataSketches Theta sketches [3], represent sets via bottom-k sampling. Theta sketches support union and set difference (A-not-B), making them naturally better suited to model deletions and rolling windows in a turnstile setting.

2.2 Differential privacy for continual release

Classical DP mechanisms for counters and histograms under continual observation use tree aggregation or binary mechanisms to release noisy partial sums over time while controlling privacy loss through composition [4]. Specifically, Dwork et al. add Laplace noise to aggregated counts at each node of a binary tree, ensuring that the total noise scales logarithmically with the number of releases while preserving user-level privacy. These methods work well for numeric aggregates with bounded per-user sensitivity, but they assume monotone contributions or simple add-only updates.

Recent theoretical work on DP distinct counting in turnstile streams introduces the notion of flippancy w, defined as the maximum number of times a single user's indicator can flip between "present" and "absent" over the entire time horizon [5]. Jain et al. achieve user-level DP by scaling noise according to each user's flippancy, allowing continual release of distinct counts even in the presence of deletions. These results show that if w is small, for example, if we coalesce events so that each user/day contributes at most one insertion and one deletion, then it is possible to achieve user-level DP with additive error on the order of $O(\sqrt{w} \cdot \mathrm{polylog}\, T)$, where T is the number of releases. Complementary work studies space-efficient DP algorithms for turnstile distinct counting, emphasizing sublinear memory and worst-case lower bounds for event-level DP in this setting [6].

2.3 Gaps in current systems and our positioning

In production analytics systems, DAU/MAU are commonly computed using non-private sketches such as HLL or Theta embedded in data warehouses or streaming engines [9, 10, 11]. Deletions are often handled by offline recomputation or ad-hoc correction pipelines, without formal privacy guarantees [7, 8]. On the theory side, DP work on continual release typically assumes exact sets or describes mechanisms at the level of abstract vectors, rather than integrating with realistic sketch-based engineering [12]. Our project occupies the space between these two lines of work: we take the flippancy-based DP mechanisms from the theoretical literature and combine them with deletion-friendly sketches (Theta) and a simple service interface, targeting the concrete use case of DAU/MAU under GDPR-style deletions.

3 Hypothesis

We hypothesize that combining a flippancy-aware, user-level differentially private mechanism with a deletion-friendly distinct-count sketch (Theta) will yield DAU/MAU estimates whose median relative error is below 5% on realistic workloads at privacy levels $\varepsilon \leq 1.0$, while still correctly honoring all user deletion requests.

Use Case 1: Streaming Distinct Counting (HLL / Theta)

Hypothesis: We argue that using a deletion-friendly Theta sketch instead of a traditional HLL allows accurate DAU/MAU estimation under user deletions, while still supporting efficient streaming updates.

Use Case 2: Differential Privacy under Continual Observation

Hypothesis: We hypothesize that applying user-level DP with bounded sensitivity and flippancy-awareness ensures that continual MAU/DAU releases protect privacy while maintaining low relative error ($< 5\%$) for realistic ϵ values.

Use Case 3: Combining DP with Deletion-Friendly Sketches

Hypothesis: We argue that integrating a flippancy-aware DP mechanism with a deletion-supporting Theta sketch achieves both privacy guarantees and correct handling of "right to be forgotten" deletions, outperforming naive DP baselines that ignore deletions.

4 Experimental Settings

4.1 Datasets and Workloads

We evaluate our system on synthetic turnstile streams designed to mimic realistic app usage over a 30-day window ($T = 30$). User populations of $N \in \{10^4, 10^5\}$ are generated. Each user u is assigned an activity probability $p_u \sim \text{Beta}(0.5, 5.0)$ to reflect heavy-tailed engagement. For each day d, user activity is sampled as Bernoulli(p_u); positive events $+u@d$ are emitted at random timestamps.

Deletion requests are modeled by selecting fractions $q \in \{0.0, 0.05, 0.20\}$ of users uniformly at random. For each such user u, a deletion day d_{del} is sampled from $\{10, \ldots, T\}$, and negative events $-u@d$ are emitted for all previous active days $\leq d_{\text{del}}$. Both random and adversarial deletion patterns are considered to stress-test flippancy bounds.

4.2 Algorithms and Baselines

We compare four approaches:

1. **Exact non-private baseline:** Hash sets per day for DAU and rolling unions for MAU. Fully implemented.

2. **Non-private Theta sketch:** Daily Theta sketches with set-difference for deletions. Implemented to isolate sketch approximation error.

3. **DP baseline ignoring deletions:** User-level DP applied to monotone streams. Planned for demonstration of bias from ignoring deletions.

4. **Proposed DP + Theta mechanism:** Theta sketches combined with flippancy-aware, user-level DP. Daily coalescing ensures $W_{\text{bound}} = 2$. Implemented for DAU; MAU release planned but experimental design specified.

4.3 Differential Privacy Parameters

User-level (ε, δ) DP is adopted with $\delta = 10^{-6}$. DAU experiments use $\varepsilon_{\text{DAU}} \in \{0.1, 0.3, 1.0\}$; MAU uses $\varepsilon_{\text{MAU}} \in \{0.1, 0.5, 1.5\}$.

Laplace noise is added proportional to sensitivity Δ. For DAU, $\Delta = 1$ per-day coalescing; for MAU with $W = 30$ days, $\Delta = W_{\text{bound}} = 2$. Privacy budget is tracked with a simple composition accountant, and hard monthly caps $\varepsilon_{\text{total,DAU}} = \varepsilon_{\text{total,MAU}} = 4.0$ are enforced.

4.4 Evaluation Metrics

Accuracy: Absolute and relative error metrics are computed per day:

$$\text{AE} = |C_{\text{alg}} - C_{\text{true}}|, \quad \text{RE} = \frac{\text{AE}}{\max(1, C_{\text{true}})}.$$

Percentiles (median, 90th, 95th, max) summarize error distributions. DP coverage is assessed as the fraction of days where the true count lies within the Laplace confidence interval.

Correctness under deletions: Track $\text{MAU}_{\text{before}}$, $\text{MAU}_{\text{after-true}}$, and MAU_{alg} to verify deletion semantics. Deviations illustrate the effect of respecting deletions.

Performance: Measure ingestion throughput (events/sec) and query latency (ms) for GET/POST endpoints under light concurrency. Two sketch configurations (exact vs Theta) and two privacy settings ($\varepsilon = 0.3, 1.0$) are tested to demonstrate trade-offs.

4.5 Implementation Notes

Implemented in Python 3.11 using FastAPI and Uvicorn. Cryptographic hashing with per-day salt ensures privacy. Random seeds fixed for reproducibility. Daily DAU fully implemented; MAU planned as rolling union with noise addition. Figures will record exact experimental configuration (N, q, ε, sketch type) to allow full replication.

References

[1] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for database applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[2] P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier. HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. In *Proceedings of the 2007 Conference on Analysis of Algorithms (AOFA)*, 2007.

[3] Apache DataSketches. Theta sketches: distinct counting and set operations. Project documentation, 2024. `https://datasketches.apache.org/docs/Theta/ThetaSketchSetOps.html`

[4] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *STOC*, pages 715–724, 2010.

[5] P. Jain, I. Kalemaj, S. Raskhodnikova, S. Sivakumar, and A. Smith. Counting distinct elements in the turnstile model with differential privacy under continual observation. In *NeurIPS*, 2023.

[6] R. Cummings et al. Differentially private space-efficient algorithms for counting distinct elements in the turnstile model. *arXiv preprint arXiv:2505.23682*, 2025.

[7] A. Podda. Shedding light on the legal approach to aggregate data under the GDPR. In *UNECE Workshop*, 2021.

[8] Data Calculus. Understanding the right to be forgotten in business intelligence and data analytics. Online article, 2023. `https://datacalculus.com/en/knowledge-hub/data-analytics/data-ethics-and-compliance/understanding-the-right-to-be-forgotten/`

[9] Wall Street Prep. Daily active users (DAU): definition and calculation. Online article, 2024. `https://www.wallstreetprep.com/knowledge/daily-active-users-dau/`

[10] Investopedia. Monthly active users (MAU): definition and overview. Online article, 2025. `https://www.investopedia.com/terms/m/monthly-active-user-mau.asp`

[11] Mixpanel. Guide to product analytics: measuring DAU, WAU, and MAU. Product guide, 2024. `https://mixpanel.com/content/guide-to-product-analytics/chapter_2/`

[12] Apple Inc. Differential privacy overview. Technical whitepaper, 2017. `https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf`