Managing Differential Privacy in Large Scale Systems

Abhishek Tiwari

Citation: *A. Tiwari*, ”Managing Differential Privacy in Large Scale Systems”, Abhishek Tiwari, 2024. doi:10.59350/e2sdz-hnb96

Published on: September 22, 2024

The promise of differential privacy is compelling. It offers a rigorous, provable guarantee of individual privacy, even in the face of arbitrary background knowledge. Rather than relying on anonymization techniques that can often be defeated, differential privacy works by injecting carefully calibrated noise into computations. This allows aggregate statistics and insights to be extracted from data while masking the contributions of any single individual. Major players like Apple, Google, and the US Census Bureau have already adopted differential privacy for various applications. As privacy regulations like GDPR and CCPA raise the stakes for data protection, we're likely to see differential privacy become increasingly mainstream.

While differential privacy provides strong mathematical guarantees (see [1]), it's not a panacea for all privacy concerns. It protects against certain types of privacy attacks and inferences, but can't prevent all possible misuse of data. There's also the risk that the complexity of these systems could lead to a false sense of security. If not properly understood and configured, differential privacy mechanisms could be circumvented or provide weaker protections than expected. Particularly, deploying differential privacy in large, real-world systems is far from trivial.

In case of large scale systems, one of the key challenges stems from the fact that differential privacy operates on a "privacy budget" - a finite resource that gets depleted as queries are run against the data. Once the budget is exhausted, the privacy guarantees no longer hold. In a system with multiple applications and users accessing the same underlying data, careful management of this shared privacy budget becomes critical. How do you allocate the budget across different applications? How do you handle continuous data streams where the budget would eventually run out? These are the types of thorny issues that a recent paper by Kuchler et. al aims to address (see [2]).

Cohere

In this paper, authors introduce Cohere, a new system that simplifies the use of DP in large-scale systems. At the heart of Cohere's approach is its innovative architecture, which provides a comprehensive framework for managing differential privacy across large-scale systems. Understanding this architecture is key to appreciating how Cohere tackles the complex challenges of privacy resource management.

Architecture

Cohere's architecture is built on three main layers: the data layer, the application layer, and the privacy management layer. This structure allows for a separation of concerns while providing a unified view of data and privacy resources across the entire system. It builds on the emerging "data lakehouse" paradigm, which aims to provide a single source of truth for an organization's data assets.

By extending this foundation with privacy-specific metadata and APIs, Cohere enables fine-grained tracking and allocation of privacy budgets.

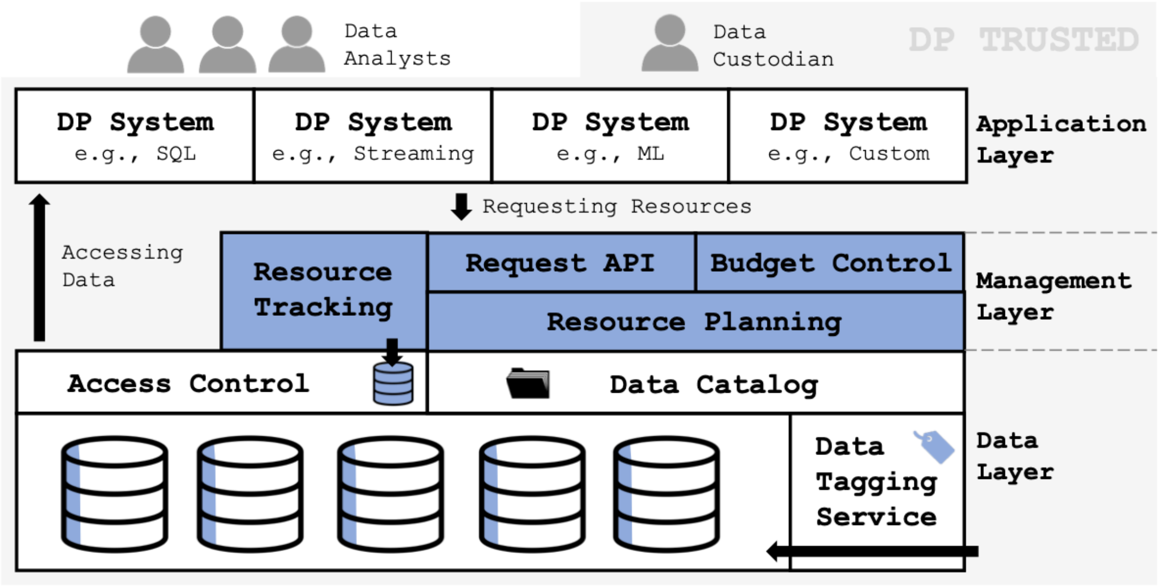


Figure 1: Cohere implements a unified interface that allows heterogeneous applications to operate on a unified view of users' data.

Data Layer

The data layer forms the foundation of the architecture. It extends existing data storage systems with privacy-specific metadata, crucial for fine-grained privacy analysis and enforcement. A key feature here is the data tagging service, which assigns each user a unique ID and tags all user data accordingly. This enables precise tracking of which data belongs to which user - a necessity for accurate privacy accounting. The data layer also introduces the concept of "partitioning attributes" and "block IDs", which allow for more efficient privacy analysis when dealing with subsets of users.

Application Layer

Building on this foundation is the application layer. This is where various differentially private systems and applications interact with the data. Cohere provides a unified API that allows these diverse applications to express their data and privacy requirements in a standardized way. This is a significant step forward from the current state of affairs, where different DP applications often operate in isolation, unable to share resources effectively.

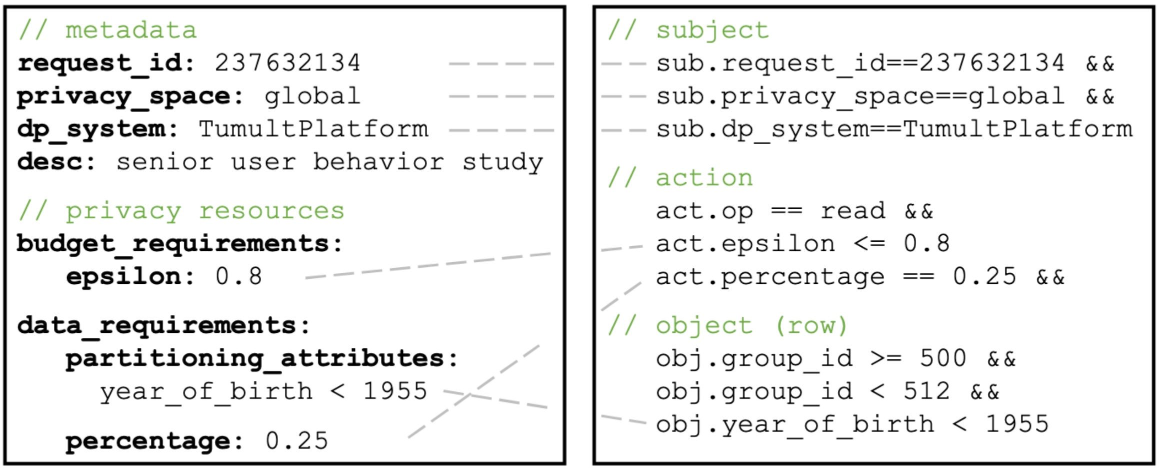
Privacy Management Layer

The real magic happens in the privacy management layer. This is where Cohere coordinates and optimizes the use of privacy resources across all applications. It includes a budget control system that manages overall privacy budgets, and a resource planner that allocates these budgets to specific requests. The resource planner uses sophisticated optimisation techniques to maximise utility while respecting privacy constraints.

In a continuously running system, the privacy budget for a fixed set of users will eventually be depleted. Cohere tackles this by implementing a sliding window approach, where new users are gradually rotated in as older ones are retired. This allows the system to replenish its privacy budget over time, enabling indefinite operation. Crucially, Cohere does this in a way that maintains clear semantics for applications and avoids introducing bias into the active user population.

Attribute-based access control

One of the most innovative aspects of Cohere's architecture is its approach to access control. Rather than relying on pre-processed, privacy-safe views of data, Cohere uses attribute-based access control (ABAC) to enforce privacy budgets at runtime. This allows for more flexible and efficient use of privacy budgets, as applications can access raw data within the constraints of their allocated budget.



```
// metadata
request_id: 237632134
privacy_space: global
dp_system: TumultPlatform
desc: senior user behavior study

// privacy resources
budget_requirements:
   epsilon: 0.8

data_requirements:
   partitioning_attributes:
     year_of_birth < 1955
   percentage: 0.25
```

(a) Data Access Request

```
// subject
sub.request_id==237632134 &&
sub.privacy_space==global &&
sub.dp_system==TumultPlatform

// action
act.op == read &&
act.epsilon <= 0.8
act.percentage == 0.25 &&

// object (row)
obj.group_id >= 500 &&
obj.group_id < 512 &&
obj.year_of_birth < 1955
```

(b) ABAC Policy

Figure 2: Data access request includes precise data requirements to enable more fine-grained resource management and DP budget requirements. A set of ABAC policies defines permitted data flows and implicitly records the history of data usage.

Privacy Spaces

Another key architectural feature is Cohere's support for privacy spaces. This allows organisations to maintain separate privacy budgets for different teams or purposes, providing flexibility in how privacy resources are managed across the organisation.

Challenges

It's important to note that implementing such an architecture is no small feat. It requires significant changes to how data is stored, accessed, and processed throughout an organization. The benefits in terms of privacy protection and efficient resource use are substantial, but so too is the effort required to put such a system in place.

Moreover, while Cohere's provides a solid foundation, it's not a plug-and-play solution. Organizations will need to carefully consider how to integrate it with their existing systems and processes. They'll need to train their data scientists and analysts on how to work within this new paradigm. And they'll need to develop new governance structures to oversee the use of privacy budgets and ensure they're being allocated in line with organizational priorities and ethical considerations.

The sliding window for user rotation, while clever, does impose a limit on how long user data remains active in the system. This could be problematic for applications that require longitudinal analysis over extended time periods. There's also the question of how to set appropriate privacy budgets and parameters in the first place - a complex task that requires carefully weighing business needs against privacy risks.

Conclusion

Despite these challenges, Cohere's architecture represents offers a blueprint of how organisations can build data systems that are privacy-aware from the ground up. The system also employs sophisticated privacy analysis techniques to squeeze more utility out of a given privacy budget. By leveraging ideas like parallel composition and amplification via subsampling, Cohere can provide tighter bounds on privacy costs compared to more naive approaches. The authors claim this translates to significant real-world gains - in their experiments, Cohere achieved 6.4-28x improvements in utility compared to previous state-of-the-art systems.

Finally, there are fundamental tensions and tradeoffs at play. More aggressive privacy protections inevitably reduce utility to some degree. Different applications may have vastly different privacy vs utility needs. Cohere aims to navigate these tradeoffs through intelligent, automated resource allocation. Using optimization techniques, it can allocate privacy budgets across applications in a way that maximizes overall utility while respecting privacy guarantees. This moves us closer to the ideal of "privacy by design" - where privacy protections are baked into systems from the ground up rather than bolted on as an afterthought.

References

[1] A. Tiwari, "Mathematical Guarantee," 2024, *Abhishek Tiwari*. doi: 10.59350/ghs12-1vq60.

[2] N. Küchler, E. Opel, H. Lycklama, A. Viand, and A. Hithnawi, "Cohere: Managing Differential Privacy in Large Scale Systems," *arXiv*, 2023, doi: 10.48550/arXiv.2301.08517.