EBOOK

A Guide to Data Access Governance

With Immuta and Databricks

2020

IMMUTA™ databricks

Table of Contents

Introduction

Data-driven insights are no longer optional — they're necessary. Organizations that leverage data to make critical business decisions have a definitive competitive advantage. While consumers today appreciate and often pay more for customized products and services, delivering that experience often requires the use of sensitive personal data.

Sensitive data is the most valuable asset for analytics and data science. But if data engineers and architects can't securely prepare it and maximize its utility for real-time access by analysts and data scientists, is it still as valuable?

Centralizing data and making it discoverable and actionable are key to data-driven innovation, personalized user experiences, and revenue growth. But the advent of new data privacy regulations like California's CCPA and Europe's GDPR, in addition to increasingly stringent internal data rules and security policies, have put a tax on data engineers and architects, who are charged with translating these rules and regulations into executable policies so that data consumers can gain access to critical data for innovation. This complicated and time-consuming process — which can lead to personal liability if a data leak or beach occurs — can delay or halt advanced analytics and data science projects, which require fast access to data.

To maintain the balance between data security and speed to data, Immuta integrates with Databricks to provide fine-grained access control capabilities, automated and dynamic data security controls, and data access auditing. This whitepaper outlines those capabilities and considerations for when each should be used, so that data teams can unlock even more use cases with Databricks and Immuta.

Why might data engineers and architects need Immuta for Databricks?

- They're responsible for managing complex policies across many tables — as well as the ensuing role explosion.
- Their existing platforms don't enable dynamic row-, column-, and cell-level access control and security, so access controls risk being too restrictive or too broad.
- They're unable to implement global policies and access controls for each platform in their cloud data ecosystem, so must manually do it for each individual platform.
- They're unable to dynamically restrict access based on time, geography, purpose, data sharing agreements, or other scenarios that may arise.
- They're responsible for consistent security and auditing across multiple database systems, but don't have a centralized way to manage the process or ensure its uniformity, which causes confusion and frustration.
- They're expected to stay up-to-date on evolving regulations and to implement sufficient data security measures accordingly — otherwise they could be held personally liable for leaks or breaches.
- They spend their time managing case-by-case data access within an organization, as opposed to supporting democratized data.
- They need to be able to mask data while preserving its original format for non-production use.
- They're expected to enable compliant, secure collaboration on data sets without inadvertently granting unauthorized access or hindering analytics initiatives.
- Their policies are difficult to scale because they're unable to draw upon existing organizational glossaries or underlying metadata.

Attribute-Based Access Control

What is it?

Attribute-based access control (ABAC) would be more accurately defined as dynamic and scalable access control. Why? It is akin to writing code using variables, eliminating the need to repeatedly write the same blocks of code — or in this case, policies.

The list of variables to utilize — including user, object, action, and environmental attributes — allows data engineers and architects to separate who the user is, what they are doing, and what policy should be dynamically applied at query time, rather than predefining all policy up front by conflating who has access to what based on roles, as with role-based access control (RBAC).

Similar to how Databricks separates compute and storage, Immuta separates access control policy logic from the data platform. Within Databricks, Immuta extracts access control policy logic from both compute and storage, eliminating the need for data engineers and architects to spend time and energy recreating access control policies for each individual data tool. The flexibility to utilize the same access control policies — whether role-, attribute-, or purpose-based — consistently across distributed data storage and compute providers is a powerful capability, particularly in satisfying data use compliance, changing business rules or working within a cloud data ecosystem that includes Databricks.

Additionally, cloud-based data teams leveraging multiple systems do not need to define every single user across systems within Databricks in order to implement secure access controls. Instead, Immuta draws upon existing authentication and metadata systems, including attributes from external systems. As cloud adoption continues to accelerate, this is — and will continue to be — a key capability that substantially increases data teams' efficiency and productivity.

What problems does it solve for data teams?

For data engineers and architects using Databricks, Immuta's attribute-based access controls reduce time and risk by enabling them to create data policies once and enforce them dynamically across all tables.

Data teams can draw upon data attributes to write and scale ABAC policies across hundreds of roles, without the static limitations of role-based access control methods and tools like Apache Ranger.

Role-based access control, though manageable in small organizations, creates "role explosion" as data consumers and roles increase. This forces data engineers to manually manage hundreds or thousands of user roles in an effort to control access to data in specific tables or databases. As a result, the reason for implementing RBAC in the first place — saving time on enabling access control on an individual basis — is directly reversed. At the same time, the surface area of risk jumps up.

As the number of roles grows, so does the complexity of tracking and applying the appropriate permissions to roles. Role explosion creates a situation in which the data engineers and architects can no longer easily monitor which roles belong to which access permissions, so translating a user need to an actual role assignment can be very complex to manage. In turn, the risk of human error in the manual administration process grows even more quickly than the roles themselves.

Additionally, manual processes are not scalable: For organizations using Databricks in tandem with other cloud data and analytics tools, data teams are forced to implement access controls system-by-system, generate new views or copy data. The cloud architecture is engineered to scale, but manual RBAC processes prevent scaling user adoption. Immuta's attribute-based access controls eliminate these extra steps, avoiding role explosion and unmanageable data copies, while saving data engineers and architects time and increasing productivity.

Data teams are also able to use R, Scala, Python, and SQL for both coarse- (table-level) and fine- (row-, column-, cell-level) grained access controls within Databricks. Immuta supports all core languages in order to satisfy a wide range of data science and BI needs. Most importantly, though, Immuta's attribute-based access controls accelerate easy, secure data scalability across users and functions. As WorldQuant Predictive has experienced firsthand, this reduces the burden on data engineers and architects and reduces time to access and insights — all without ever having to navigate outside of Databricks.

How is it implemented?

To understand how Immuta's attribute-based access control policy is deployed within Databricks, consider the case of WorldQuant Predictive.

WorldQuant Predictive is a data science firm that leverages AI, machine learning and quantitative finance approaches to develop predictive models and potential solutions for organizations to leverage in making business decisions. WorldQuant Predictive's team of researchers — with backgrounds in AI, data science, machine learning, and data modeling — is based throughout the world and leverages hundreds of thousands of data sources, both public and proprietary, within Databricks.

While this volume of data, which includes sensitive data, is highly valuable from an insights perspective, its use on a shared cloud platform by such a large, decentralized population of data consumers is inherently risky. WorldQuant Predictive needed an automated, scalable way to quickly ingest new data and apply access controls that would be neither overly broad nor overly restrictive, and would not delay speed to data access.

Watch WorldQuant Predictive CTO Slava Frid explain how his team used Databricks & Immuta to enable research at scale

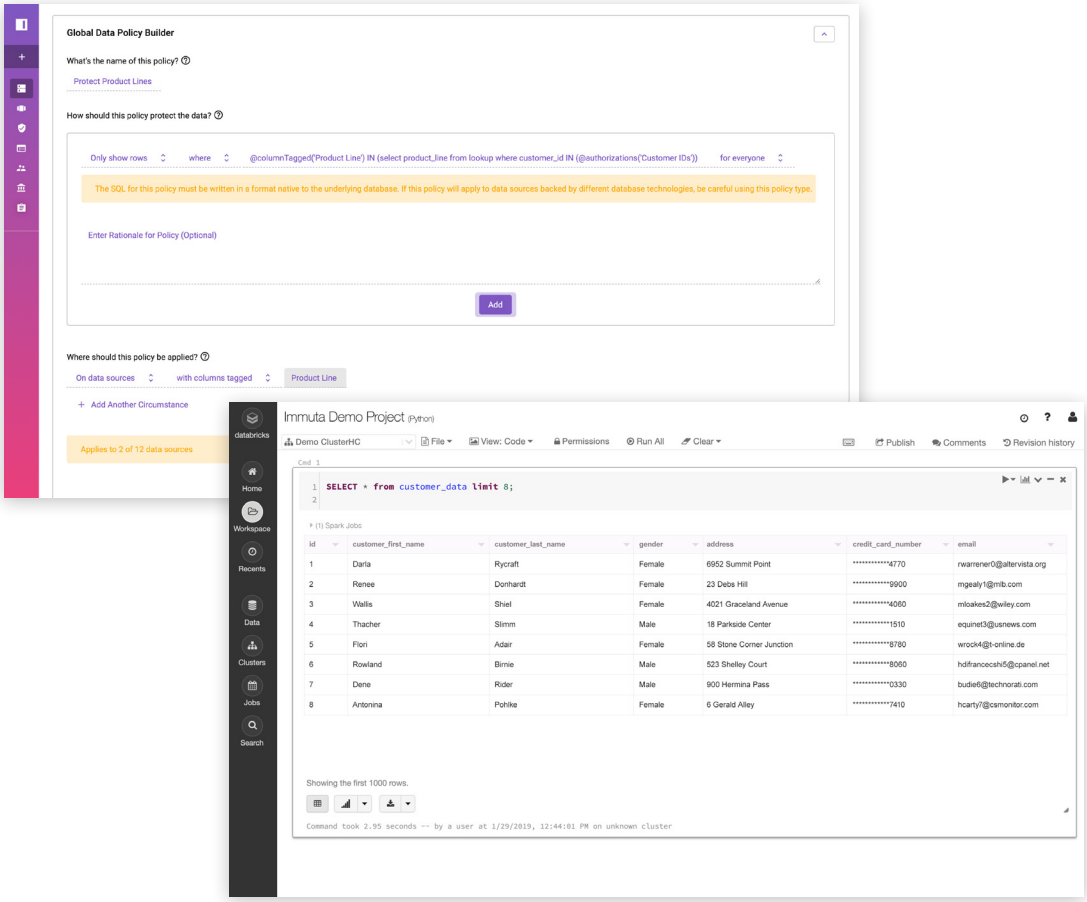
* Recorded for 2020 Spark+AI Summit

WATCH VIDEO

Immuta's native integration streamlines this process as new data is introduced to Databricks. When a WorldQuant Predictive researcher uploads any data source to their Databricks workspace, Apache Airflow picks up the new data so it can be analyzed in Notebooks. Once the analysis is complete, Databricks notifies the researcher with a link to a newly created Dashboard, which the researcher accesses to verify

the findings. Once the researcher approves the analysis, the data is moved into raw and trusted data stages. From there, Databricks and Immuta data sources are created and scanned for sensitive data and attribute-based access controls are dynamically enforced on Spark jobs.

This process means that data access is determined at query time. For an organization like WorldQuant, which needs to give its contracted researchers quick access to data, applying access controls at runtime eliminates the need to have all data users defined in a specific system, which vastly reduces the burden on data engineers and architects. Consequently, data teams can quickly and securely scale data access for both internal and external data consumers.



Automated Privacy Control

What is it?

Manual processes, as evidenced by role-based access control approaches, are simply not practical for an increasingly cloud-based data ecosystem. Users rely on Databricks — in addition to their other cloud platforms — for fast, automated data storage and compute. Why should data teams expect any less from their data governance solution?

With Immuta and Databricks, automated data security and privacy controls layer on top of dynamic attribute-based access controls to serve as additional buffers against unauthorized access, data leaks, and re-identification.

There are a few prominent examples of automated security and privacy controls that Databricks users can implement with Immuta.

De-identification

To understand de-identification, it's important to first understand the two types of identifiers — in other words, the personal information that can be used to help identify an individual.

1. **Direct identifiers** are the pieces of personal information that are unique to an individual and can be used in isolation to identify that single person. For this reason, direct identifiers are highly sensitive and strictly regulated. Examples of direct identifiers include social security numbers, passport numbers, taxpayer identification numbers, full facial images, and medical record numbers.

2. **Indirect identifiers**, also known as quasi-identifiers, are pieces of personal information that are not unique to a single person. While indirect identifiers cannot alone be used to identify a specific individual, they are still considered sensitive because they can often be combined with other information to single somebody out. Examples of indirect identifiers include height, ethnicity, hair color, car make and model, and occupation.

With that baseline knowledge, there are two types of de-identification that teams can utilize with Databricks and Immuta: masking and pseudonymization.

Masking is the process of removing or obscuring identifiers, most commonly by way of generalization or suppression. Generalization assigns the same broad value for any given attribute — for example, replacing an attribute, `hair color`, with the value any. Suppression works the opposite way, by removing values entirely or replacing them with a constant — for instance, replacing an attribute, `hair color`, with the value `redacted`.

Pseudonymization is a de-identification method that is generally a preliminary step in protecting direct identifiers. In this approach, direct identifiers are replaced with non-sensitive values on a one-to-one basis, so that data subjects remain distinct. The non-sensitive values must be randomly selected and fully independent of the actual value — this can be achieved through cryptographic hashing. An example of pseudonymization would be replacing the identifier `John Smith` with the value `James Johnson`.

Sensitive Data Discovery & Tagging

As with role-based access control, manually tagging sensitive data as it is uploaded to Databricks is a significant time commitment for data teams, in addition to introducing the risk of human error. As data becomes available faster than ever before, it's easy to see how this process can quickly become unmanageable for data engineers and architects, who could also potentially be held liable for any sensitive data that slips through the cracks and ends up in the wrong hands.

As new data enters Databricks, however, Immuta's sensitive data discovery feature automatically classifies and tags direct, indirect, and sensitive identifiers for efficient human inspection. This vastly reduces the risk of human error, while simultaneously making more efficient use of data teams' time.

Data identified as being sensitive can be assigned tags that correspond to access control policies which are dynamically applied at query time. Immuta automatically tags sensitive fields, like PII or PHI, in addition to enabling data engineers and architects to create tags as needed. Tags can also map to data privacy protection laws such as CCPA and GDPR, which in turn correspond to Immuta's templated starter policies that automate policy creation and ensure compliance with regulations. This means data engineers and architects can seamlessly ingest sensitive data and apply regulatory-compliant policies without fear of personal liability or wasted time.

What problem does it solve for data teams?

According to the Immuta Data Engineering Survey: 2021 Impact Report, the most challenging aspect of managing a data pipeline for data teams is data masking and security, followed by auditing and monitoring. The least challenging part? Extract and load.

Maintaining an ETL pipeline and `GRANT`s can be complicated and cumbersome for data teams when data governance isn't built into the process. This is because data engineers and architects must transform raw data into "clean" data before it can be utilized by analysts, data scientists, or any other data consumers. The report found that this task overwhelmingly falls on the shoulders of data engineers — more than half of survey respondents said transform is done by data engineers, either before or after load. So, in order to make data operational, data engineers must apply comprehensive privacy controls as part of transform. This model also sets data engineers and architects up for having to make copies of data and apply controls to them manually.

Without an integrated system, these time-intensive, manual processes may or may not result in sufficiently protected data.

Immuta's automated privacy controls — including dynamic data masking and sensitive data discovery — make this task much faster and more secure for Databricks users. Incorporating data governance into the ETL process with Immuta's native Databricks integration eases the burden of combing through and identifying potentially sensitive data, and streamlines the process of updating pipelines and `GRANT`s as data users and policies change over time.

When data consumers run a query in Databricks, Immuta's dynamic privacy controls are applied at run time. This captures the most current data policies and permissions, simplifies the ETL process, and eliminates the need for manual processes and data copies. Dynamic, automated controls that reduce the risk of human error, missed sensitive data, and re-identification are a more efficient, secure approach for data teams.

How is it implemented?

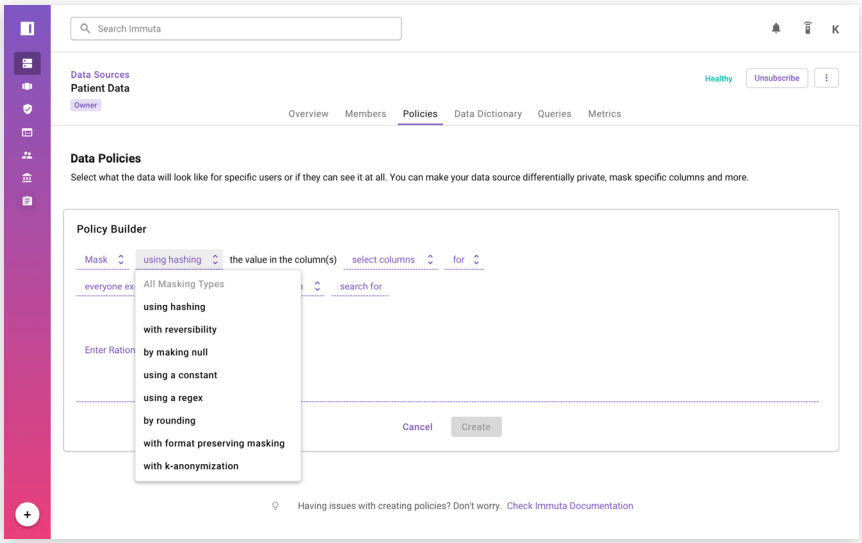
To understand automated privacy control with Databricks and Immuta in practice, use medical records as an example.

In this scenario, a data engineer receives a data set of medical records and is responsible for preparing it for use by a medical center's billing department and the city's public health department. The data set contains protected health information (PHI) including name, address, insurance information, and social security number, and the data engineer must implement policies and privacy controls that restrict data visibility by user attribute and purpose, while achieving HIPAA compliance.

As the data set is uploaded to Databricks, Immuta's sensitive data discovery tags and classifies PHI so the data engineer can assign it to access control policies. The data engineer can then build policies in plain English that restrict data returned from a query based on user attributes. So, an accountant from the billing department may be able to view patients' full addresses and insurance information in order to process claims, but the data engineer can suppress diagnoses with a value of `redacted`.

Meanwhile, an analyst from the public health department may be able to view diagnoses, but the data engineer suppressed patients' names, social security numbers, and insurance information and generalized addresses by only showing the first three numbers of a zip code — for example, `021XX`. This would allow the analyst to see diagnoses by general area, so they can make public health recommendations, without identifying any particular individuals with a positive diagnosis.

Using automated privacy controls, the data engineer in this scenario preserves the original data set within Databricks — without making copies — and ensures data consumers see only the information relevant to their functional need.



Dynamic Security & Privacy Controls

What are they?

Today's consumers expect a certain caliber of service and experience — one that's tailored specifically for them. From suggestions on new shows to watch to push notifications at the point of purchase, data analytics reports and machine learning models leverage personal data to predict and influence consumers' decisions and improve customer service.

For better or worse, the evolution of technology and data analytics has accelerated the collection and use of personal data. While this has fueled innovation and enabled personalization, it comes with substantial risk. Personal information — including that which can be used maliciously — is more widely available than ever before. Its use can lead to increased risk of data leakage, misappropriation and distrust.

Simply safeguarding access to raw data inputs is no longer sufficient to protect personal data. Data teams must also consider what information can be inferred about an individual from a model's behavior or API output, as well as any risks that may arise from publishing a data set. Dynamic data security and privacy controls transform data in a manner that maximizes privacy and utility, enabling both internal or external data use with a reduced possibility of attacks on the privacy of individuals.

Databricks users can implement a range of dynamic security controls through Immuta's native integration:

Differential Privacy

The sheer amount of personal information being collected and used in today's environment means that no single piece of data exists in a vacuum. Differential privacy aims to mathematically limit an outsider's ability to confidently use the output of an analysis to make inferences about its input. This allows individuals providing their personal data to credibly deny their participation in the input.

Differential privacy requires the data analysis mechanism to give the same answers with similar probabilities over any pair of databases that differ by a single row. In other words, Immuta injects noise into the data analysis in order to render inference attacks nearly impossible. This way, an individual may claim that the output of the mechanism came from a database that did not include their data.

Randomized Response

While differential privacy enables people to credibly deny their participation in a data input, randomized response — also known as local differential privacy — makes it possible for participating individuals to credibly deny the contents of their participation records. This approach allows data subjects to answer sensitive or potentially embarrassing questions confidentially.

Like differential privacy, randomized response employs randomization to enhance privacy; however, unlike differential privacy, randomization is applied prior to submission and formal constraints are applied to the randomized substitutions. This means that any chosen substitution must be nearly — though not necessarily exactly — as likely to arise from any given input. As a result, all potential inputs look plausible to an attacker wishing to undo the randomized substitution.

Since the randomized response technique is applied prior to the data leaving a device, data subjects are assured protection from the moment of submission. This protection remains privatized — even in the case of subsequent breach.

k-Anonymization

k-anonymization is the data equivalent of hiding in a crowd; the more people — or in this case, data points — that are present and generally similar, the harder it is to pick out the details that can identify individuals. This approach reduces re-identification risk by anonymizing indirect identifiers, thereby destroying the signal of data.

In k-anonymization, *K* represents instances of tuples in a data set. A data set is k-anonymous when attributes within it are generalized or suppressed until each row is identical to at least k-1 other rows. Therefore, the higher the value of *k*, the lower the re-identification risk. Just as the larger the crowd is, the less likely you'll find exactly the person you're looking for, k-anonymization works particularly well with large data sets. However, lines of data may have to be redacted if there isn't enough data to anonymize indirect identifiers.

k-anonymization can help transform, analyze, and share secure data at scale, making it an important privacy enhancing technique for Databricks users dealing with large sets of sensitive data.

What problem do they solve for data teams?

Data teams are often responsible for walking the line between data utility and privacy. The inherent problem is that creating and implementing controls that maximize both privacy and utility is highly complex and risky.

Effective implementation that mathematically guarantees against re-identification often requires a PhD in applied mathematics.

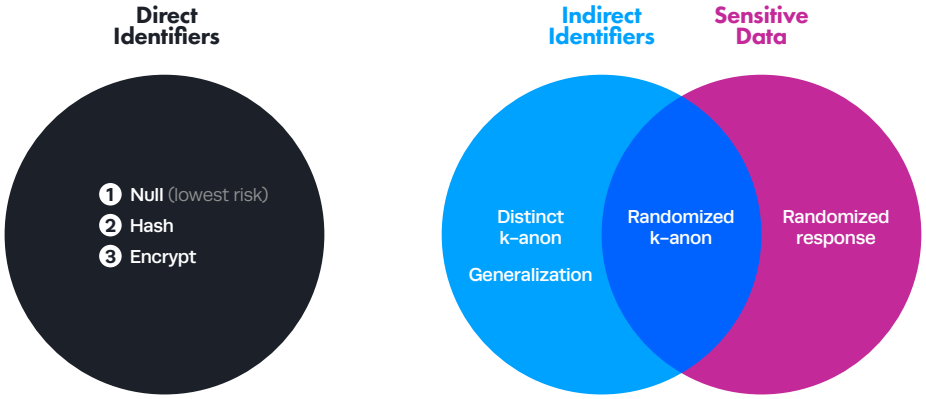
Although the dynamic security and privacy controls shield some data fields, they are better able to preserve data's utility than some other methods.

In the k-anonymized health data example, suppressing `[age]` and `[zip code]` reduces the risk of re-identification by nulling the data, but a statistically relevant number of `[age]` and `[zip code]` combinations can still be analyzed to observe diagnosis trends by age and gender cohorts, for instance. For WorldQuant Predictive researchers analyzing COVID-19 spread, this technique enabled data sharing from multiple data sources that helped streamline collaboration and quickly generate predictive models to form hypotheses without inadvertently exposing sensitive PHI or identifying information in the process.

A key benefit of dynamic security and privacy controls like k-anonymization, randomized response, and differential privacy is that they simultaneously reduce the risk of re-identification, maximize data utility and privacy, and enable data sharing between teams. Without these techniques, data teams are hindered in their ability to unlock collaboration and speed to data access without compromising data integrity, security, and compliance. Immuta's natively enforced advanced security capabilities mean Databricks users can access and share critical data — including sensitive data — quickly, efficiently, and securely.

How are they implemented?

To understand how dynamic security and privacy controls apply to specific types of data, it helps to divide data into segments: that which contains direct identifiers, indirect identifiers, and sensitive data.



As this figure demonstrates, these privacy enhancing technologies (PETs) can buffer protection for indirect identifiers and sensitive data, and vastly reduce the opportunity for inference or linkage attacks.

These advanced controls have grown in popularity across industries to enable secure data sharing. Data teams are well versed in the stringent requirements — and enforcements — associated with regulations like GDPR, HIPAA, and CCPA, which explicitly state that data must be protected such that the chances of an individual being re-identified remain as close to unfeasible as possible. History has shown that personal data is easy to re-identify if not adequately protected — Harvard professor Latanya Sweeney found that 87% of the population can be re-identified simply using birth date, zip code, and gender.

For Databricks users like the WorldQuant Predictive researchers, leveraging personal data like credit card transactions and cell phone location data to drive timely analytics is a necessity, but one that comes at a high risk. By law, a Databricks table containing PHI, such as genders, zip codes, and credit card numbers, must have PETs applied before it can be shared with data scientists and analysts.

Once the table has been registered in Databricks with Immuta, a data engineer can create a new data source and Databricks connection. Then, automated tagging flags identifiers and sensitive data in the set. Using Immuta's no-code policy builder, the data engineer can enforce one or more dynamic security and privacy controls. For example, k-anonymity can be enforced using the masking or suppression method on columns tagged as `[gender]` and `[zip code]`.

With this policy selected, Immuta scans the Databricks table to calculate the statistics required for k-anonymization. A fingerprint service runs a query against Databricks to collect counts for each possible group of values in the data source and produces custom predicates for each column. The data engineer can either use this automated return's minimum group size, *k*, or can manually specify a value for *k*. To protect identity data, the predicates only contain a whitelist of values visible to users.

This secured data set is exposed as a table in Databricks so data analysts and scientists can access and query the table. Since the controls are enforced natively on read from Databricks, the underlying data remains unmodified and not copied, and policies are applied to the plan that Spark builds for a user's query from the Notebook.

Data Access Auditing

What is it?

Data engineers and architects are often asked by security and compliance teams, "who accessed this data over the past 2 months?" On the surface, this seems like a simple question; in reality, it's much more complicated.

Auditing is a necessary part of the data pipeline management process but in a multi-cloud compute platform environment, it's difficult to execute. Disparate compute layers mean security and privacy controls may be enforced in the compute layer, passed through to storage or even implemented in analytical applications. Various data consumer roles and their corresponding permissions add an extra layer of complexity.

Databricks users can bypass tedious manual monitoring and auditing processes with Immuta's natively integrated capabilities. All audit logs and information, as well as cluster queries to the cloud provider, are done with system accounts so that data usage across cloud compute platforms — not just within Databricks or your cloud provider services — can be captured consistently. This includes audits of not just data queries but also of all policy actions being taken in Immuta, such as changing policies or subscribers to a table. Additionally, Databricks data teams can implement purpose restrictions that trigger consent workflows and simplify the process of monitoring and auditing data usage.

Immuta's unified audit logs, automated reports, and purpose-based access controls provide granular snapshots of which data consumers accessed specific data sources, when, and for what purpose, as well as changes to data over time. As a result, Databricks users always have quick access to a unified, centralized policy tier that proves compliant data usage across the organization.

What problem does it solve for data teams?

As personal and sensitive data has become more widely available and collected by organizations, data subjects have become increasingly aware of and concerned with how their information is being used.

Regulations have followed accordingly and now data monitoring and auditing is not an option — it's a requirement.

Yet, as an environment with multiple cloud platforms becomes the new normal, data teams' ability to audit data use consistently across platforms like Databricks and others is exponentially more complicated. This has the potential to substantially limit acceptable use cases for data in cloud analytics. Furthermore, it makes the process of responding to security and compliance requests time-consuming, if not impossible.

Databricks users avoid these concerns and roadblocks with Immuta's integrated monitoring and auditing capabilities. Data teams can efficiently

provide legal and compliance teams with detailed logs and reports at the data level, to provide full transparency about what data was accessed, by whom, when, and for what purpose. This automates the otherwise-laborious process of formalizing proof of compliance to adhere to federal, state, and industry standards and regulations, which in turn bolsters collaboration between data and compliance teams, and mitigates concerns about legal enforcement and personal liability.

How is it implemented?

The GDPR requires data consumers to have an acceptable purpose for accessing data, and consequently, that data teams can provide evidence of that purpose.

This is a complicated process because, without the right tools, purpose is difficult to capture.

Yet, GDPR and other regulations, including HIPAA, HITRUST, and SOC-2, are not optional and data teams can only avoid incurring noncompliance enforcement with comprehensive audits. Layered on top of regulations are employment and industry standards or contractual agreements, among others.

Immuta's active data catalog means all data within Databricks is searchable and discoverable, and captures details of where data is stored, who owns it, when it was added, and how recently it was queried. Immuta's regulatory policy starters and no-code policy builder mean legal and compliance teams can assess the strength of policies in meeting compliance standards. Then, when data consumers subscribe to a data set, all global and local compliance policies are automatically applied to their Spark workloads in Databricks.

Conclusion

Centralized, secure, self-service data access is the best way to maximize data's impact. In today's rapidly evolving market, that can be the secret weapon to gaining a critical competitive edge. Whether analyzing data to inform public health policy decisions or to predict consumer behavior, data consumers need secure access to data — as fast as possible.

This isn't an option without the right combination of data science and data access governance capabilities. Immuta's native integration with Databricks enables organizations to automatically secure sensitive data for analytical use in industries ranging from insurance to transportation. With attribute-based access control, automated privacy controls, dynamic security and privacy controls, and data access auditing, Databricks and Immuta together enable data teams to maximize data utility and security.

In fact, Databricks customers that take advantage of Immuta's core, native capabilities experience a 100x reduction of user roles, a 40% improvement in productivity, and a 300% increase in data utilization. This means teams are able to accomplish more and unlock more data-driven outcomes in Databricks when Immuta is natively implementing dynamic data security and privacy measures.

To find out what you can accomplish when you combine the power of Databricks and Immuta, schedule a demo today.

REQUEST A DEMO

IMMUTA™

115 Broad Street, 6th Floor, Boston, MA 02110
immuta.com | (800) 655-0982

databricks

160 Spear Street, 13th Floor San Francisco, CA 94105
databricks.com | (866) 330-0121