**A black background with a black square

AI-generated content may be incorrect.**  
Hyderabad, India.

**Design Blueprint: GCP-Based Data Classification and Masking Framework**

* **Team HashHackers  
  Ashirbad Ray  
  Ashwini Bagade  
  Nikita Pardeshi  
  Neelesh Kerekar  
  Shreya Shree**

**Problem Statement**

In the current data-driven environment, ensuring the security and privacy of sensitive information is paramount. Our team operates on a large-scale data pipeline that processes around 30 million records. At the organizational level, this volume of data is significantly high. However, there are no access control mechanisms in place to restrict user access to specific columns or fields. Additionally, Personally Identifiable Information (PII) within the dataset remains unencrypted, posing a significant security and regulatory risk. To address these concerns, we must develop a mechanism that restricts access to PII, ensuring that only authorized users can view or interact with this sensitive information. Implementing such a solution will enhance our data security posture and help us comply with relevant data protection regulations.

**Objective**

To establish a robust, scalable, and automated data protection framework on **Google Cloud Platform (GCP)**, this solution leverages a combination of native GCP services—**Cloud Data Loss Prevention (DLP)**, **Data Catalog/Dataplex**, **Cloud Functions/Cloud Run**, and **BigQuery User-Defined Functions (UDFs)**, it’s primary objective is to dynamically identify, classify, and safeguard sensitive data stored in **BigQuery**.

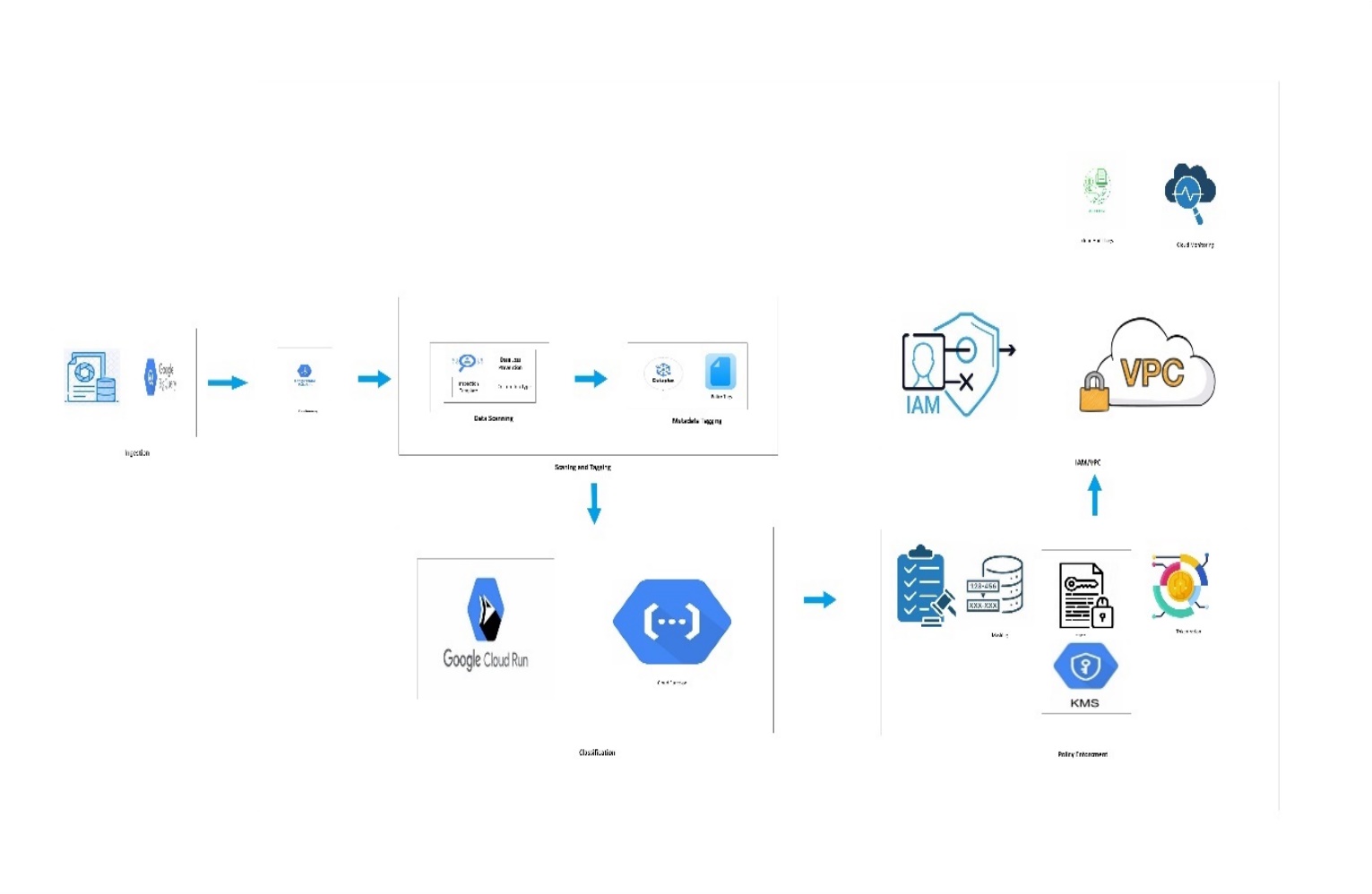
The framework begins by using **Cloud DLP** to automatically scan and detect sensitive information such as personally identifiable information (PII), financial data, or health records. Once identified, **Data Catalog/Dataplex** is employed to tag and classify the data based on its sensitivity level, enabling fine-grained data governance and discoverability.

To enforce protection policies, **Cloud Functions/Cloud Run** are triggered in response to classification events or data access patterns. These functions apply appropriate data protection techniques—such as **masking**, **encryption**, or **tokenization**—based on predefined sensitivity rules. **BigQuery UDFs** are then used to transform or redact sensitive fields during query execution, ensuring that data privacy is maintained even during analytics operations.

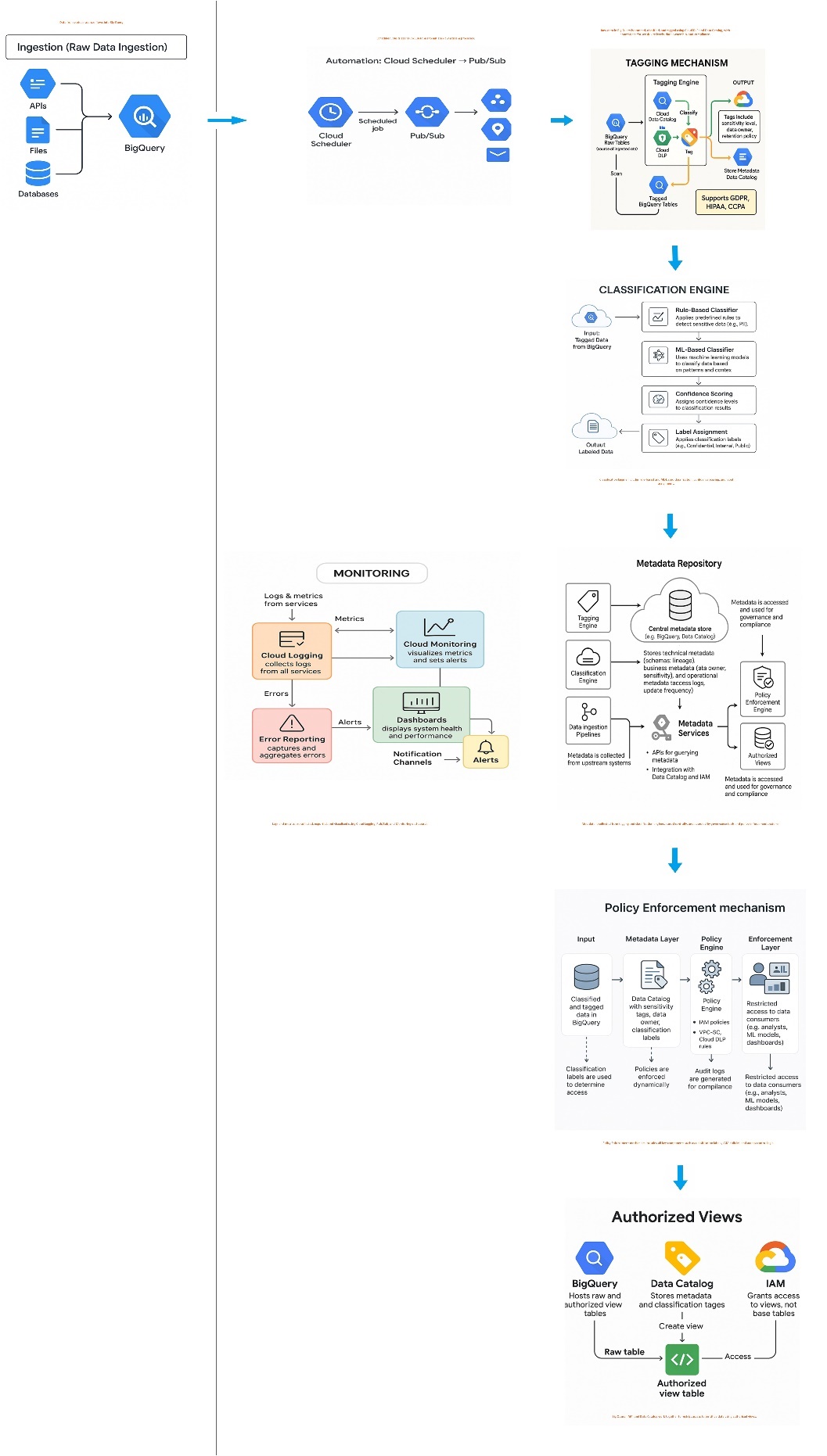
This end-to-end framework not only automates the detection and protection of sensitive data but also ensures compliance with regulatory standards such as **GDPR**, **HIPAA**, and **CCPA**, by enforcing consistent and auditable data protection practices across the data lifecycle.

**Architecture Overview**

**Abstract View**



**Detailed View:**



**End-to-End Data Protection Workflow on GCP**

1. **Raw Data Ingestion**  
   Structured or semi-structured data from various sources (e.g., application logs, transactional systems, third-party APIs) is ingested into **BigQuery**, serving as the central data warehouse. This step initiates the data lifecycle within the secure analytics environment.
2. **Sensitive Data Scanning with Cloud DLP**  
   **Cloud Data Loss Prevention (DLP)** scans the ingested BigQuery tables using predefined **inspection templates**. These templates contain rules and detectors to identify sensitive data types such as names, email addresses, credit card numbers, and dates of birth. The scan results include findings with confidence scores and likelihood levels.
3. **Metadata Tagging via Data Catalog/Dataplex**  
   Based on the DLP findings, **Data Catalog** automatically applies **column-level metadata tags** (e.g., pii, email, dob) to the identified sensitive fields. These tags serve as classification labels that inform downstream data governance and protection policies.
4. **Automated Classification Engine**  
   A **Cloud Function** is triggered to process the metadata tags and classify each column according to its sensitivity level (e.g., public, internal, confidential, restricted). This classification logic can be customized to align with organizational data protection policies and regulatory requirements.
5. **Policy Enforcement through BigQuery Views**  
   Depending on the classification, **BigQuery Authorized Views** and **User-Defined Functions (UDFs)** are used to enforce data protection techniques such as **masking**, **tokenization**, or **encryption**. These views ensure that only appropriately transformed data is exposed to users, based on their access privileges.
6. **Secure Access Control with IAM and VPC-SC**  
   **Identity and Access Management (IAM)** roles are configured to grant users access only to the protected views, not the raw data. Additionally, **VPC Service Controls (VPC-SC)** establish secure perimeters around GCP services, preventing unauthorized data movement and enhancing the overall security posture.

**Component Details**

**1. Tagging Engine**

* Sensitive Data Detection with Cloud DLP  
  The process begins with Cloud Data Loss Prevention (DLP) scanning BigQuery tables using predefined inspection templates. These templates include a set of infoTypes—such as EMAIL\_ADDRESS, CREDIT\_CARD\_NUMBER, DATE\_OF\_BIRTH, etc.—that help identify sensitive data patterns across columns.
* **Mapping to Custom Tags**Once sensitive fields are detected, each infoType is mapped to a corresponding custom tag that reflects the organization's data classification taxonomy. For example, EMAIL\_ADDRESS might be mapped to a pii\_email tag, while CREDIT\_CARD\_NUMBER could be tagged as financial\_data.
* Tag Application via Data Catalog APIs  
  Using Data Catalog APIs, these tags are programmatically applied at the column level using tag templates. This metadata tagging enables consistent classification, discoverability, and policy enforcement across the data lifecycle.

**2. Classification Engine**

* Automated Classification with Cloud Functions  
  A Cloud Function is triggered to read the applied tags and execute classification logic. Based on the presence and type of tags, each column is assigned a sensitivity classification level:
  + Public: No sensitive tags detected; data is safe for open access (e.g., City, Country).
  + Limited: Contains low-risk tags like email, city; suitable for internal use with minimal restrictions.
  + Confidential: Includes moderately sensitive tags such as account\_number, date\_of\_birth; requires protection mechanisms.
  + Highly Confidential: Contains high-risk tags like PAN, biometric\_data, password; demands strong encryption and restricted access.
* **Auditability and Metadata Storage**The classification results are stored in a centralized repository—either a BigQuery table or Firestore—to maintain an auditable record of data sensitivity levels. This supports compliance reporting and traceability.

**3. Policy Definition and Enforcement**

* **Mapping Classifications to Protection Techniques**Each classification level is associated with a specific data protection technique, ensuring that data is handled appropriately based on its sensitivity:

| **Classification** | **Protection Technique** | **Example** |
| --- | --- | --- |
| **Public** | **None** | **City, Country** |
| **Limited** | **Masking** | **Email → e\*\*\*@domain.com** |
| **Confidential** | **Tokenization** | **Account Number → TKN-12345** |
| **Highly Confidential** | **Encryption (via KMS)** | **PAN → Encrypted using Cloud KMS key** |

* These techniques are enforced using BigQuery Authorized Views, User-Defined Functions (UDFs), and Cloud KMS for encryption, ensuring that only authorized users can access or view sensitive data in its original form.

**4. Policy Enforcement Layer**

* This layer ensures that sensitive data is protected at the point of access, using a combination of BigQuery features and GCP security controls:
* **BigQuery Authorized Views**  
  These views act as secure, filtered representations of the underlying data. They expose only the **masked, tokenized, or encrypted** versions of sensitive fields, ensuring that users never directly access raw sensitive data. This abstraction allows for safe data sharing without compromising privacy.
  + **User-Defined Functions (UDFs)**  
    UDFs are used to implement **custom masking logic** within SQL queries. Techniques include:
  + **Regex-based redaction**: Hiding parts of strings (e.g., masking all but the last four digits of a credit card).
  + **Format-preserving tokenization**: Replacing sensitive values with pseudonyms that retain the original format, useful for analytics without revealing actual data.
  + **Row-Level Security (RLS)**  
    RLS restricts access to specific rows in a table based on the **user’s identity or role**. For example, a regional manager may only see data related to their region, while a global analyst sees aggregated data across all regions.
  + **Column-Level Security (CLS)**  
    CLS hides or restricts access to specific columns containing sensitive information. Unauthorized users querying the table will not see these columns at all, ensuring **principle of least privilege** is enforced.

**5. Audit & Monitoring**

* This component provides visibility, traceability, and proactive alerting to ensure the integrity and compliance of the data protection framework:
* **Cloud Logging**  
  Captures detailed logs of key operations such as:
  + DLP scan executions and results
  + Metadata tag applications and updates
  + Classification logic executions and outcomes  
    These logs are essential for **troubleshooting, auditing, and compliance reporting**.
* **Cloud Monitoring**  
  Enables the creation of **custom dashboards and alerting policies**. Alerts can be configured for:
  + Unauthorized access attempts
  + Policy violations (e.g., access to unmasked data)
  + Failures in classification or tagging workflows  
    This ensures that security teams are notified in real-time about potential risks.
* **Audit Logs**  
  GCP’s **Admin Activity and Data Access logs** record every access to sensitive datasets, including:
  + Who accessed the data
  + What data was accessed
  + When and from where the access occurred  
    These logs support **forensic investigations**, **regulatory audits**, and **internal compliance reviews**.

**Regulatory Compliance Support**

This data protection framework is architected to align with major global data privacy regulations by ensuring that sensitive data is **accurately identified**, **systematically classified**, and **appropriately protected** throughout its lifecycle. It supports compliance with the following key regulations:

**1. GDPR (General Data Protection Regulation)**

The **GDPR** is a comprehensive data protection law that governs how organizations handle the personal data of **EU citizens**, regardless of where the organization is based. Key requirements include:

* **Lawful and transparent processing** of personal data.
* **Data minimization** and purpose limitation.
* **Security and accountability** for data handling.
* **Rights of data subjects**, including access, rectification, and erasure.

**Framework Support for GDPR:**

* **Cloud DLP** identifies personally identifiable information (PII) such as names, email addresses, and national IDs.
* **Data Catalog** applies sensitivity tags to ensure traceability and governance.
* **BigQuery UDFs and Authorized Views** enforce masking or tokenization to minimize data exposure.
* **Cloud Logging and Audit Logs** provide full traceability of data access and processing activities, supporting GDPR’s accountability and auditability requirements.

**2. HIPAA (Health Insurance Portability and Accountability Act)**

**HIPAA** mandates the protection of **Protected Health Information (PHI)** in the U.S. healthcare sector. It includes three major rules:

* **Privacy Rule**: Governs the use and disclosure of PHI.
* **Security Rule**: Requires administrative, physical, and technical safeguards.
* **Breach Notification Rule**: Mandates reporting of data breaches.

**Framework Support for HIPAA:**

* **Cloud DLP** detects health-related infoTypes such as medical record numbers and diagnosis codes.
* **Cloud KMS** encrypts PHI both at rest and in transit, ensuring confidentiality and integrity.
* **IAM and VPC-SC** enforce strict access controls and secure perimeters.
* **Audit logs** track all access to PHI, supporting HIPAA’s audit and breach notification requirements.

**Security Best Practices**

To ensure robust protection of sensitive data and maintain a strong security posture, the framework incorporates the following best practices:

* **VPC Service Controls (VPC-SC)**  
  VPC-SC creates **secure perimeters** around GCP services, preventing unauthorized data movement across project or service boundaries. This helps mitigate risks such as data exfiltration and enforces **zero-trust principles** within the cloud environment.
* **IAM Conditions for Context-Aware Access**  
  Identity and Access Management (IAM) policies are enhanced with **conditions** that evaluate the context of access requests—such as **geographic location**, **device security status**, or **time of day**. This ensures that access is granted only under trusted conditions, reducing the risk of credential misuse.
* **Automated Key Rotation with Cloud KMS**  
  **Cloud Key Management Service (KMS)** is configured to automatically rotate encryption keys at regular intervals. This practice maintains **cryptographic hygiene**, reduces the risk of key compromise, and aligns with compliance requirements for secure key lifecycle management.
* **Enforcement of Least Privilege Principle**  
  Access permissions are granted based strictly on the **minimum level of access required** to perform a task. This principle is applied across all roles, services, and resources, minimizing the attack surface and preventing privilege escalation.

**Scalability & Automation**

The framework is designed to scale seamlessly with growing data volumes and evolving compliance needs, using automation to reduce manual overhead:

* **Cloud Scheduler for Periodic DLP Scans**  
  **Cloud Scheduler** triggers **Cloud DLP** scans at regular intervals (e.g., daily or weekly), ensuring that newly ingested data is continuously monitored for sensitive content without manual intervention.
* **Event-Driven Pipelines with Pub/Sub**  
  **Cloud Pub/Sub** is used to publish events such as data ingestion or tag updates. These events trigger downstream workflows—like classification, masking, or policy enforcement—via **Cloud Functions** or **Cloud Run**, enabling real-time responsiveness.
* **Centralized Metadata Repository in BigQuery**  
  All tagging, classification, and policy metadata is stored in a **centralized BigQuery repository**. This repository supports **governance dashboards**, **audit trails**, and **compliance reporting**, providing a single source of truth for data sensitivity and protection status.

**Future Enhancements**

To further strengthen the framework and adapt to emerging needs, the following enhancements are planned:

* **AI/ML Integration for Smarter Classification**  
  Incorporate **Natural Language Processing (NLP)** and **machine learning models** to improve the accuracy of data classification. These models can detect contextually sensitive information that rule-based systems might miss, such as unstructured PII in free-text fields.
* **Behavioral Analytics for Anomaly Detection**  
  Implement **behavioral analytics** to monitor data access patterns and detect anomalies—such as unusual query volumes, access from unfamiliar locations, or deviations from typical user behavior. This helps identify potential insider threats or compromised accounts.
* **Multi-Cloud and Hybrid Cloud Support**  
  Extend the framework to support **AWS**, **Azure**, and **hybrid environments**, enabling consistent data protection policies across diverse cloud infrastructures. This ensures compliance and governance in multi-cloud strategies adopted by modern enterprises.

**Summary**

In today’s data-centric environment, safeguarding **Personally Identifiable Information (PII)** is not just a best practice—it’s a regulatory and ethical imperative. Organizations are increasingly required to implement robust data protection strategies that ensure sensitive information is accessed only by authorized individuals and handled in compliance with global privacy laws.

This document presents a comprehensive, high-level design for implementing a **column-level access control and real-time data masking framework** within **Google BigQuery**. The proposed solution leverages native GCP services such as **Cloud DLP**, **Data Catalog**, **Cloud Functions**, **BigQuery UDFs**, and **IAM** to dynamically identify, classify, and protect sensitive data.

By applying **fine-grained access controls**, **automated classification**, and **context-aware masking or encryption**, the framework ensures that:

* Sensitive data is only visible to users with the appropriate clearance.
* Data masking and tokenization are enforced in real time during query execution.
* All access and policy enforcement actions are logged for auditability and compliance.

This approach not only enhances **data security** but also supports **regulatory compliance** with standards such as **GDPR**, **HIPAA**, and **CCPA**. Furthermore, the framework is designed to scale with growing data volumes and can be extended to support **multi-cloud environments**, making it a future-ready solution for enterprise-grade data governance.