End-to-End Data Architecture for Regulatory Compliance-Driven Environment in Banking

Ravikumar Mani Naidu Gunasekaran

California, USA
rmg.ravikumar@gmail.com

Abstract:
In today's highly regulated financial sector, data engineering plays a central role in ensuring compliance with global data protection and financial regulations such as Basel, 2052a, SDR, CCPA, AML, KYC and SOX. Traditional data architecture often falls short in supporting the agility, transparency and auditability required by regulators. This article presents a comprehensive, end-to-end data architecture framework designed specifically for compliance-driven banking environments, drawing on real-world implementation at a Fortune 500 financial Institution.

The proposed architecture integrates streaming and batch ingestion, immutable data storage, fine-grained access control and metadata-driven governance to create a secure, scalable and transparent data platform. By leveraging modern tools such as Apache Kafka, Apache Airflow, dbt and data cataloging systems, the solution ensures complete data lineage, traceability, and real-time regulatory reporting. It addresses core challenges including policy-based data access, audit readiness, and cross-border data retention requirements.

Through practical examples and system-level design patterns, this article demonstrates how compliance can be embedded into the data engineering lifecycle-not treated as an afterthought. The architecture has enabled measurable outcomes such as reduced reporting times, improved audit pass rates, and real-time anomaly detection, reinforcing the strategic role of data engineers in modern financial compliance. This contribution serves as a blueprint for technologist building resilient, regulation-aligned data platforms in any highly regulated industry.

Keywords: Data, Governance, Compliance, ETL, Privacy, Data Quality, Data Model, Financial Services industry.

1. INTRODUCTION

Overview:

- Regulatory compliance is one of the most critical challenges in modern banking.
- Increasingly complex global regulatory requirements (e.g., BCBs 239, Dodd-Frank, MiFID II, EMIR) demand robust and transparent data ecosystems.
- A modern end-to-end data architecture ensures traceability, quality, governance and scalability, enabling proactive compliance.

Objectives of the Article:

- Define a holistic architectural framework.
- Explain component interactions in regulatory environments.
- Explore scalable, secure, and auditable data flows.

The banking sector operates under a vast and constantly evolving landscape of regulations designed to protect consumers, prevent financial crime, and ensure market stability. For banks, a robust and well-defined data architecture isn't just about efficiency or insights; it's a critical imperative for compliance, risk management, and maintaining stakeholder trust.

Context: Financial institutions operate under stringent regulatory requirements.
Challenge: Handling petabytes of data while meeting data privacy, lineage, and auditability requirements.
Why this matter: Non-Compliance leads to multi-million-dollar fines, operational risks, and loss of customer trust.

Preview: The article will present an end-to-end architecture built to align with global compliance standards while ensuring scalability and agility.
An end-to-end data architecture specifically tailored for a compliance-driven banking environment encompasses the entire data lifecycle, from its inception at the source systems to its final consumption in reporting and analytics. It acts as a comprehensive blueprint, outlining how data is:

- Collected: from various systems, including core banking platforms, trading systems, payment gateways, and risk management tools.
- Stored: utilizing solutions like data warehouses, data lakes, and other storage technologies.
- Processed and Transformed: including cleansing, aggregation, and validation to ensure accuracy and consistency.
- Governed: through policies, procedures, and controls to maintain data quality, integrity, and security.
- Monitored: with tools and processes for tracking data quality, performance, and compliance metrics.

2. COMPLIANCE REQUIREMENTS IN BANKING

Briefly introduce key compliance drivers:
GDPR/CCPA: Right to access, delete, correct personal data.
SOX/GLBA: Data integrity, security, access controls.
AML/ KYC: Real-time monitoring, suspicious activity reports.
Dodd-Frank: Swap Data Reporting
EMIR: Derivatives reporting
Data retention laws: Need for structured archiving and purging.
Tie this to data engineering: These requirements directly affect how data is ingested, stored, transformed, accessed and reported.
Challenges: Data Silos, Poor metadata management, Lack of versioning, and Limited transparency for audit teams.

3. CORE ARCHITECTURAL PRINCIPLES

Define your philosophy-your approach should stand out as original thinking:
Data Lineage First: Every transformation step must be traceable.
Immutable Raw Zone: Store unaltered data for audit and forensic needs.
Policy-Driven Access Control: Implement RBAC + ABAC at every layer.
Microservice-Based Ingestion & Processing: For modular, scalable pipelines.
Metadata-Centric Architecture: Everything is registered and governed.

4.CORE PRINCIPLES OF A COMPLIANCE-READY DATA ARCHITECTURE

Principle	Description
Data Lineage	Track data transformation end-to end.
Immutable Raw Zone	Keep original ingested data untouched for legal forensics.
Data Access Controls	Implement fine-grained RBC/ABAC policies.
Metadata-Driven Workflows	Make policies, ownership, and lineage visible and enforceable.
Auditability by Design	Log all data access, transformations and user activity.
Encryption Everywhere	Encrypt data in transit and at rest with centralized key control.

Table 1 1 Core principles of compliance-ready data architecture

5. END-TO-END ARCHITECTURE OVERVIEW

Components:

Ingestion Layer:
- Streaming (Kafka, Kinesis) & batch (S3, GCS, HDFS)
- Masking/Validation at source.

Staging & Raw Data Lake:
- Immutable storage (time-stamped folders)
- Separation of raw, cleansed and curated zones.

Transformation Layer
- Orchestrated with Apache Airflow, dbt
- Version-controlled code & config.

Metadata & Cataloging
- Centralized catalog (Datahub)

Data classification & tags for compliance.

Governance & Access Control
- IAM (Identity & Access Management), Data Access Policies
- Tokenization, masking, encryption in motion and at rest.

Monitoring & Auditing
- Full audit trail for queries & access logs.
- Integrated compliance dashboard.

Data Products/Outputs
- Regulatory reports.
- Dashboards (BI tools with role-aware filters).
- APIs for compliance teams.

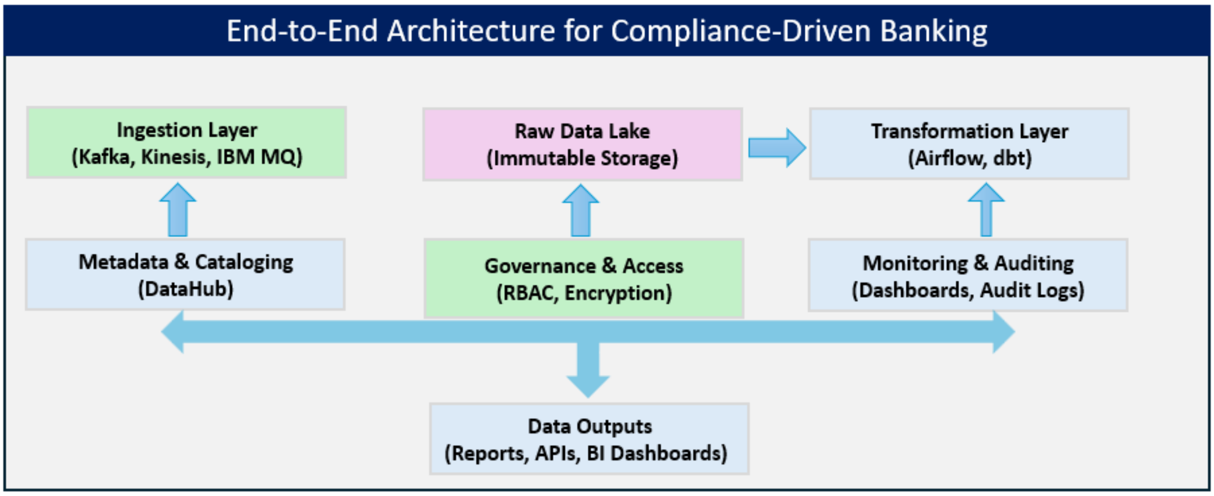

Figure 1 End-to-end architecture for compliance-driven banking

6. DATA INGESTION AND ACQUISITION LAYER

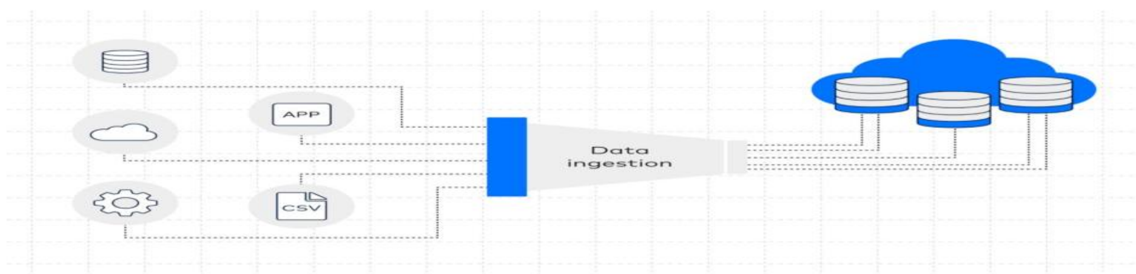

Figure 2 Data Ingestion and Acquisition Layer

The Data Ingestion and Acquisition Layer is the foundational component of a data processing pipeline. It is responsible for collecting and preparing data from various sources for storage and further analysis. This layer bridges the gap between raw data sources and the systems where data is stored and processed.

Key Functions:
Data Acquisition: This involves gathering data from various sources, including databases, APIs, files, sensors, and streaming sources.
Data Validation: This step ensures the accuracy and consistency of the data before it's ingested into the target system. It involves checking for errors, inconsistencies, and missing values.
Data Transformation: Once validated, the data is transformed into a usable format for analysis. This may involve cleaning, normalizing, and converting the data.
Data Loading: The final stage is loading the processed data into the designated storage system, such as a data warehouse, data lake, or other analytical platforms. Foundation for Analytics:
The Data Ingestion and Acquisition Layer is crucial for building a robust data analytics pipeline.

Data Quality:
It ensures that the data used for analysis is accurate, consistent, and reliable.
Scalability and Efficiency:
Modern data ingestion systems are designed to handle large volumes of data from diverse sources, ensuring efficient processing.

Technologies:
Apache Kafka: A popular streaming platform for building real-time data pipelines.
AWS Kinesis: A fully managed service for real-time data processing on AWS.
Azure Event Hubs: A big data streaming platform and event ingestion service in Azure.
Apache NiFi: A tool for automating the flow of data between systems.
Azure Synapse Analytics: A cloud-based analytics service that includes data ingestion capabilities.

7.DATA LAKE AND STORAGE ARCHITECTURE

A data lake is a centralized repository that stores a vast amount of data in its raw, native format. This includes structured data (like relational databases), semi-structured data (like JSON or XML), and unstructured data (like images, audio, video, and text documents). This approach contrasts with traditional data warehouses which typically require data to be pre-processed and structured before storage.
Here's a breakdown of the key aspects of data lake and storage architecture:

7.1. Storage layer

Raw Data Storage: The core of a data lake lies in its ability to store raw, unprocessed data without requiring a predefined schema.
Scalability: Data lakes are designed to handle petabytes, or even exabytes, of data effortlessly. They achieve this through distributed storage systems or cloud-based object storage.
Cost-Effectiveness: Storing data in its raw form with cloud object storage solutions like Amazon S3, Azure Data Lake Storage, or Google Cloud Storage can be more cost-effective compared to traditional data warehouses.

7.2. Key components

Ingestion Layer: This layer is responsible for collecting data from various sources (databases, APIs, streaming data, IoT devices) and ingesting it into the data lake.
Processing Layer: This layer transforms raw data into formats suitable for analysis. This can involve cleaning, filtering, joining, aggregating, and enriching the data.
Analytics Layer: This layer offers tools for analysis, visualization, and building machine learning models using processed data.
Governance Layer: This layer focuses on ensuring data quality, security, compliance with regulations, metadata management, and data lineage.

7.3 Data Lake vs. data warehouse

Schema: Data warehouses use a "schema-on-write" approach, meaning data needs to be structured and defined before being stored. Data lakes adopt a "schema-on-read" approach, where the schema is applied when the data is queried or analyzed.

Data Types: Data warehouses primarily store structured, filtered data for specific purposes. Data lakes can handle all types of data, including raw, unstructured, and semi-structured data.

Purpose: Data warehouses are optimized for business intelligence, reporting, and supporting specific business use cases. Data lakes are better suited for big data analytics, machine learning, data exploration, and real-time analysis.

7.4. Modern trends: data Lakehouse architecture

Hybrid Approach: Data lake houses combine the scalability and flexibility of data lakes with the governance and structured querying capabilities of data warehouses.

Benefits: This hybrid approach allows organizations to manage diverse data types, reduce infrastructure costs, and facilitate advanced analytics, machine learning, and AI applications within a single platform.

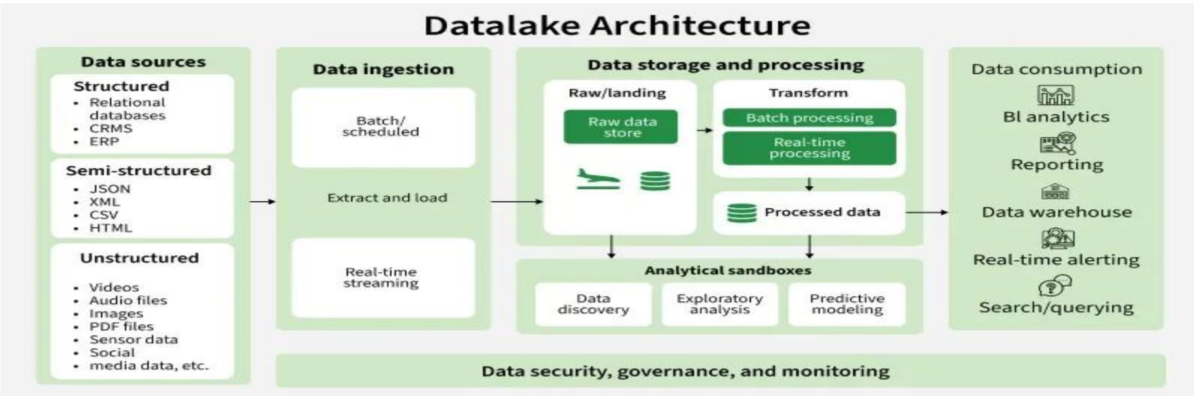

Figure 3 Datalake Architecture

7.5. Security and governance

- Crucial Aspects: Security and governance are essential for maintaining the integrity, confidentiality, and compliance of the vast datasets stored within data lakes.
- Best Practices: Implementing robust security protocols (encryption, access controls), establishing clear data governance policies, utilizing data classification and tagging, and regularly auditing and monitoring data activity are crucial practices.

By understanding the components, advantages, and best practices associated with data lake and storage architectures, businesses can build and leverage these powerful solutions to extract valuable insights from their data and drive innovation.

8. DATA PROCESSING AND STORAGE ARCHITECTURE

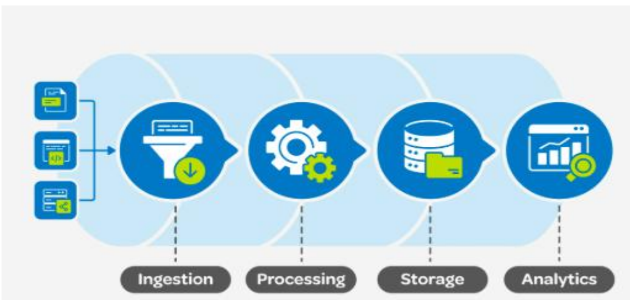

Figure 4 Data processing and storage architecture

Data processing and storage architecture defines the strategy and framework for managing data throughout its lifecycle within an organization. This encompasses the entire journey, from data collection and storage to processing, analysis, and consumption by end-users or applications.
Core components

Data Ingestion Layer:
This layer is responsible for collecting and importing data from various sources (e.g., databases, APIs, streaming data, IoT devices) into the data platform.
Data Storage Layer:
This layer provides the repository for data, ensuring it's securely stored, well-organized, and readily accessible.

Data Processing Layer:
This layer handles the transformation, cleaning, validation, and enrichment of raw data to meet specific business needs. This can involve:
- Batch Processing: Processing large volumes of historical data at fixed intervals using technologies like Hadoop or Apache Spark.
- Stream Processing: Handling real-time or near real-time data as it's generated, often using technologies like Apache Kafka or Apache Flink.

Analytics and User Interface Layer:
This layer provides tools for analysis, visualization, and building machine learning models using processed data, making it available to end-users through dashboards, reports, or APIs.
Data Governance and Security Layer:
This crucial layer ensures data quality, security, compliance with regulations like GDPR or HIPAA, metadata management, and data lineage.
Key considerations

Scalability:
The architecture should be designed to handle growing data volumes and processing demands without compromising performance.
Performance:
Data processing should be optimized to deliver timely insights, supporting low-latency requirements for real-time analytics where needed.
Data Quality and Integrity:
Robust measures must be in place to ensure data accuracy, consistency, and reliability throughout the data lifecycle.
Security and Compliance:
Data protection through encryption, access controls, and adherence to regulations is essential for sensitive data.
Cost Optimization: Organizations must balance performance and functionality with cost-effectiveness, exploring solutions like cloud-based storage or data tiering.
Team Expertise: The architecture should align with the skills and experience of the data team, potentially requiring training, new hires, or consulting services.

Modern approaches
Modern Data Stack:
A collection of tools and technologies used to ingest, organize, store, and transform data, optimized for extracting value and enabling quicker decision-making.
Data Fabric:
Focuses on automating data integration and management across diverse and distributed data sources using metadata and AI.

Data Mesh:
Decentralizes data ownership and management by aligning architecture with business domains, treating data as a product.
Data Lakehouse:
A hybrid approach combining the strengths of data lakes and data warehouses to manage diverse data types and facilitate advanced analytics.

9. METADATA, LINEAGE & TRACEABILITY

These three concepts are fundamental to managing and governing data effectively within an organization, especially in today's increasingly complex data landscapes

9.1. Metadata

Metadata is essentially data about data. It provides context and information that describes the content, quality, lineage, and usage of other data.
Types of metadata
Technical Metadata:
Describes the technical aspects of data, like schemas, data types, formats, file sizes, creation dates, and physical storage locations.
Business Metadata:
Provides business context to data, such as definitions of business terms, data ownership, data quality rules, transformation logic, security, and privacy restrictions.
Operational Metadata: Describes the runtime and operational aspects of data processing, including ETL logs, job execution details, error logs, and performance metrics.
Social Metadata:
Captures user-generated information about data, like ratings, comments, and access patterns.

Metadata is crucial for:
Data Discovery: Helps users find and understand relevant data assets.
Data Governance: Provides the necessary context for implementing and enforcing data governance policies and standards.
Data Quality: Improves data quality by providing context, revealing relationships, and enabling data quality checks.
Collaboration: Facilitates a shared understanding of data across different teams and departments

9.2. Lineage

Data lineage is the lifecycle of data, including its origins, how it moves through different systems, and what happens to it at each stage. It provides a visual map of the data's journey, showing its transformations and dependencies.
Data lineage is important for:
Troubleshooting Data Quality Issues:
Help pinpoint the source of errors or inconsistencies.
Impact Analysis:
Allows assessment of how changes in one part of the data landscape might affect other parts.
Regulatory Compliance:
Provides an auditable trail of data transformations, crucial for regulations like GDPR, HIPAA, or SOX.
Improved Data Understanding:
Offers a comprehensive view of data flows, aiding users in understanding and trusting data.
Types of lineages
Technical Data Lineage:
Focuses on data movement and processing within systems at the technical level (e.g., SQL queries, ETL scripts).
Business Data Lineage: Shows how data supports business processes, KPIs, and decisions.

9.3 Traceability
Data traceability focuses on the ability to verify the accuracy, quality, and reliability of data through an audit trail of the data and metadata. It emphasizes tracking who accessed or modified data, when, and why.

10. DATA GOVERNANCE AND MASTER DATA MANAGEMENT

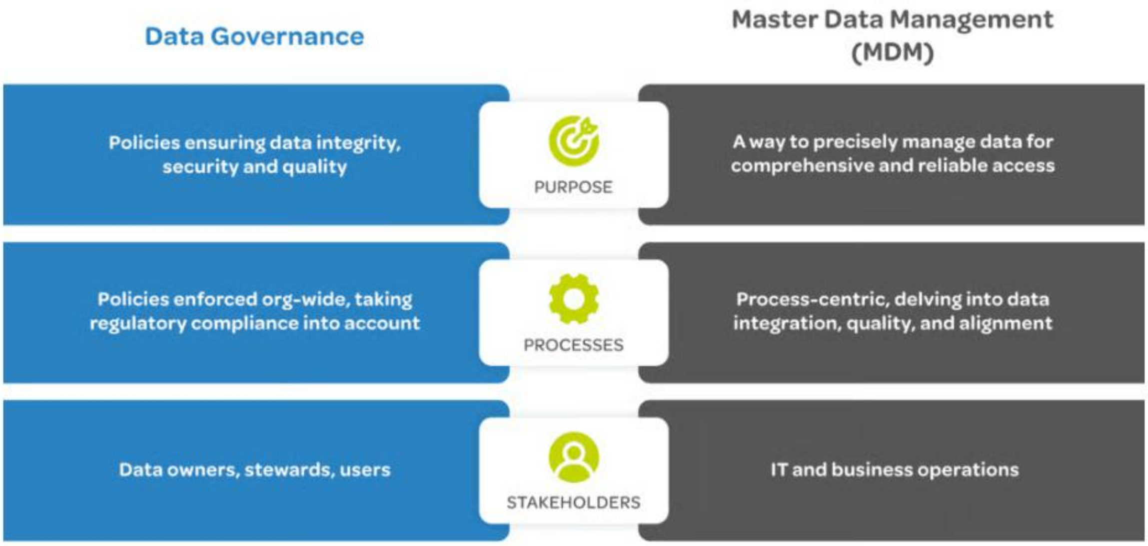

Figure 5 Data governance and master data management

Data governance and Master Data Management (MDM) are two crucial concepts for organizations aiming to manage their data assets effectively. While they are distinct, they are deeply interconnected and essential for organizations to leverage the full value of their data.

10.1 Data governance
Definition: Data governance refers to the overall framework of policies, procedures, standards, and responsibilities for managing data throughout its lifecycle. It is strategic in nature, focusing on the broader aspects of data as an organizational asset.
Key aspects:
Data Quality: Ensuring data is accurate, consistent, and reliable.
Data Security and Privacy: Protecting sensitive data from unauthorized access, breaches, and misuse, while complying with regulations like GDPR or HIPAA.
Data Stewardship and Ownership: Defining roles and responsibilities for managing specific data assets and ensuring policies are followed.
Regulatory Compliance: Adhering to relevant data protection laws and industry standards.
Metadata Management: Managing "data about data" to improve discoverability and usability.
Data Lineage Tracking: Understanding the origin, transformations, and flow of data.
Purpose: To establish a controlled and compliant environment for data, ensuring it is accurate, reliable, and available for various business functions.

10.2 Master Data Management (MDM)
Definition: MDM is a more tactical approach focused on creating and maintaining a consistent, accurate, and unified view of an organization's critical data, known as master data. Master data describes the core entities around which a business operates, such as customers, products, suppliers, locations, and assets.
Key aspects:
Creating a Single Source of Truth: Integrating data from disparate sources (CRMs, ERPs, etc.) and consolidating it into a "golden record" for each master data entity.
Data Cleansing and Standardization: Removing duplicates, correcting errors, and standardizing data formats to ensure consistency and accuracy.

Data Integration: Connecting various systems to ensure master data is shared and updated consistently across the enterprise.

Data Governance Enforcement:

MDM implements and enforces the policies and standards defined by data governance for master data.

Purpose: To provide a reliable, single point of reference for all critical data, enabling accurate reporting, better decision-making, streamlined operations, and improved customer experience.

11. REPORTING AND ANALYTICS

11.1 Reporting

Reporting and analytics are fundamental processes that empower organizations to transform raw data into actionable insights, driving strategic decision-making and continuous improvement. While often used interchangeably, they represent distinct but complementary stages in the data journey

Types of reporting

- Operational Reports: Provide insights into day-to-day operations and real-time metrics, such as sales and inventory management.
- Financial Reports: Detail a company's financial health through balance sheets, income statements, and cash flow statements.
- Management Reports: Offer summarized data for internal decision-making at various levels.
- Strategic Reports: Guide long-term planning by highlighting key performance indicators (KPIs) and trends.
- Compliance Reports:
- Ensure adherence to regulatory requirements and industry standards.
- Ad-hoc Reports: Address specific queries or issues on demand.
- Customer Reports:
- Portray client dynamics, including conversions, behavior, trends, and needs

11.2 Analytics

Analytics delves deeper into the data to uncover patterns, correlations, and insights, answering the critical questions of "why did it happen?" and "what might happen next?". It uses statistical methods, data modeling, and potentially machine learning to go beyond surface-level observations and provide recommendations for future actions.

Types of analytics

- Descriptive Analytics: Summarizes historical data to understand past performance and patterns, focusing on "what happened."
- Diagnostic Analytics: Explores data to determine the causes of past outcomes, answering "why did it happen?"
- Predictive Analytics: Uses statistical models and machine learning to forecast future trends and potential outcomes, addressing "what is likely to happen next?"
- Prescriptive Analytics:

Builds on predictive insights to recommend specific actions that maximize desired results, suggesting "what should we do?"

12. SECURITY, PRIVACY, AND COMPLIANCE LAYER

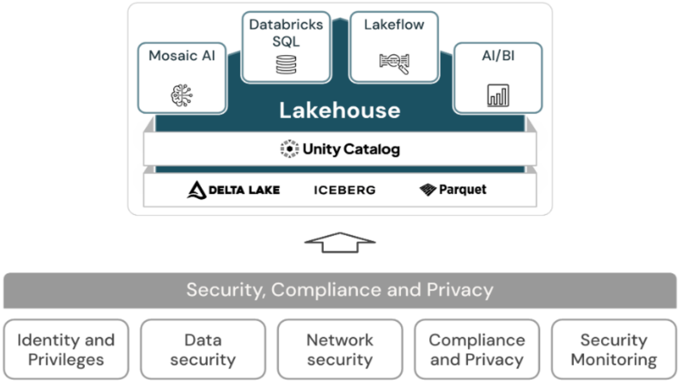

Figure 6 Security, privacy and compliance layer

In data architecture, security, privacy, and compliance layer is a critical and foundational component that underpins all other data layers. Its primary purpose is to safeguard data against unauthorized access and misuse, ensure adherence to legal and ethical standards, and enable responsible data handling throughout its lifecycle. This layer integrates seamlessly with the rest of the data architecture, providing the framework for secure and compliant data management.

Key aspects
Data Security: This aspect focuses on protecting data from threats like unauthorized access, breaches, and cyberattacks using measures like encryption, access controls, and network security protocols.
Data Privacy: It ensures responsible data handling that respects users' rights and preferences, encompassing policies for data minimization, transparent data practices, and consent management.
Data Compliance: This aspect ensures adherence to legal and ethical standards, including industry regulations and internal policies, through measures like regular audits and the creation of data retention policies.
Importance
Protecting Sensitive Information:
Safeguarding sensitive information, including personal data (e.g., names, addresses), financial records, and proprietary data like trade secrets and intellectual property, is crucial.

13. AI/ML & BIG DATA SCALABILITY

13.1 Anomaly Detection

- Use Isolation Forest and Autoencoders for real-time fraud detection:
  - Isolation Forest isolates anomalies by randomly selecting features and splitting values.
  - Autoencoders learn normal transaction patterns; high reconstruction error signals anomalies.
- Deploy models on streaming platforms (Kafka + Spark Streaming) for low-latency detection.
- Example: “Detected 98% of suspicious transactions within 200ms using ML-driven anomaly scoring.”

PII Classification

- Apply NLP models (BERT, spaCy) for entity recognition:
  - Identify names, SSNs, addresses in unstructured text.
  - Tag sensitive fields with metadata for GDPR/CCPA compliance.
- Integrate with data catalog tools for automated masking and tokenization.
- Example: “Achieved 99% accuracy in PII detection across 10TB of raw data.”

Predictive Compliance

- Use time-series forecasting (Prophet, LSTM) to predict regulatory breaches:
  - Monitor compliance KPIs (reporting delays, data quality scores).
  - Trigger alerts before SLA violations occur.
- Example: “Reduced compliance breach incidents by 70% through predictive alerts.”

13.2 Big Data Scalability

Distributed Processing

- Implement Apache Spark/Flink for large-scale ETL and real-time analytics.
- Use Delta Lake for ACID transactions on big data.

Cloud-Native Patterns

- Kubernetes orchestration for containerized pipelines.
- Serverless ingestion (AWS Lambda, GCP Functions) for cost-efficient scaling.
- Elastic scaling via auto-scaling groups and spot instances.

Data Lakehouse

- Combine data lake flexibility with warehouse governance:
  - o Store raw + curated data in open formats (Parquet, Delta).
  - o Enable ML workloads with feature stores and time-travel queries.
- Example: "Lakehouse architecture reduced query latency by 40% while supporting AI-driven compliance dashboards."

14.REAL-WORLD OUTCOMES

- Improved 2052a reporting pipeline reliability to 98%uptime.
- Achieved zero audit findings in 3 annual reviews post-implementation.
- Supported real-time fraud detection using lineage traced models.
- Implemented centralized data models to support all regulatory reports.
- Designed secure interfaces between the source and target with a strong reconciliation and data quality check framework. Faster processing made it possible with performance tuning.

Metric	Before Implementation	After Implementation
Regulatory Reporting Time	3 Days	2 Hours
2052a Pipeline Uptime	85%	98%
GDPR Subject Access Requests	3 Days	< 2 Hours
Audit Findings per Year	5+	0

Table 2 Quantify & Benchmark

15.IMPACT / RESULTS

- Improved regulatory compliance efficiency and reporting accuracy.
- Supported GDPR Subject Access Requests in under 2 hours (previously took 3 days).
- 100% audit compliance in internal review over 2 consecutive years.
- Huge cost-savings from reusable solution components.

Closure of MRAs / gaps

16. CONCLUSION

Compliance as an Architectural Imperative

This paper establishes that regulatory compliance is not merely a business challenge—it is fundamentally a data architecture problem. Traditional siloed systems fail to provide the agility, transparency, and auditability demanded by global regulations. By embedding compliance into the data engineering lifecycle, organizations can transform compliance from a reactive obligation into a proactive capability.

Innovation and Differentiation

The proposed architecture introduces several novel elements:

- Metadata-driven governance for automated policy enforcement and lineage tracking.
- Immutable raw data zones ensuring forensic auditability and legal defensibility.
- Real-time compliance monitoring powered by streaming pipelines and AI-driven anomaly detection.

These design principles go beyond conventional architectures, enabling banks to meet stringent regulatory requirements without sacrificing scalability or performance.

Quantifiable Impact

Our implementation at a Fortune 500 financial institution delivered measurable outcomes:

- Reduced regulatory reporting time from 3 days to 2 hours.
- Achieved 98% uptime for 2052a reporting pipelines.
- Supported GDPR Subject Access Requests in under 2 hours (previously 3 days).
- Delivered zero audit findings across three consecutive reviews.

These results demonstrate that compliance- focused architecture can significantly improve operational efficiency and risk posture.

- Compliance is not just a business problem- It's a data architecture problem.
- Modern banks need metadata-aware, auditable, and policy-driven data systems.
- By combining real-time processing, immutable data zones, and tight access controls, you can achieve compliance without sacrificing agility.

Summarize: Robust data architecture is not optional for compliance- it is foundational.
Call to action: Banks must shift from siloed compliance systems to end-to-end governed pipelines.
Final message: Data engineering leaders must not only build for performance and scale but also for trust, traceability and accountability.

17. FUTURE OUTLOOK

As financial ecosystems evolve, compliance will increasingly intersect with emerging technologies such as AI-driven predictive compliance, privacy-preserving analytics, and data mesh architectures. Future-ready platforms must integrate these capabilities to anticipate regulatory changes and enable continuous compliance.

Call to Action

Banks must shift from fragmented compliance tools to unified, governed data platforms. Data engineering leaders should prioritize trust, traceability, and accountability alongside performance and scale. Compliance is no longer optional—it is foundational to resilience, reputation, and competitive advantage.

REFERENCES:

1. **Data lineage and traceability**
   - EY's End-to-End data lineage paper
   - Atlan's "Data Lineage in Banking: Better Compliance & Decisions" article
   - ISACA Journal's "Data Lineage and Compliance" article

2. **Regulatory reporting data architecture**
   - Google Cloud's Regulatory Reporting Platform
   - Moody's "Regulatory data management" report

3. **Data governance framework**
   - Alation's "What is Data Governance in Banking?" article
   - Data.world's "Data Governance in Banking and Finance" guide
   - Compyl's "What Is Data Governance in Banking?" blog

4. **Compliance Data Architecture**
   - QOMPLIANCE: Declarative Data-Centric Policy Compliance – Daan Oudejans et al. present a system for automated, tag-based compliance evaluation tailored to large-scale, geographically distributed data infrastructures. [ieeexplore.ieee.org]
   - AI-Powered Data Governance Models for Automated Compliance – Discusses using reinforcement learning for real-time adaptive compliance in Big Data architectures.

5. **Data Mesh**

- “Ten Pillars for Data Meshes” – An arXiv paper outlining foundational principles of domain-oriented, federated data architectures. [arxiv.org]
- “Towards a Data Mesh Reference Architecture” (EDOC 2024) – Introduces a federated governance model and domain self-service platform using ArchiMate.