HakiLend's Big Data Modernization Project
by Linda Kelida (169589)

1. Introduction to Big Data (2 Marks)

Looking at **volume** first, HakiLend handles an enormous amount of data. Every day, their systems process 20 million transactions, while also maintaining two decades of historical records from their legacy mainframes. This massive scale of data processing creates significant technical demands, especially as mobile banking adoption continues to grow across their markets. The **velocity** of data presents another critical challenge. HakiLend needs to detect fraudulent activities in near real-time, requiring immediate analysis of incoming transactions.

When it comes to **variety**, HakiLend deals with a complex mix of data types. Their systems must process traditional structured banking records alongside semi-structured credit reports from external agencies. This diversity of data formats a sophisticated approach to data integration and management. The **veracity** of data is particularly crucial given HakiLend's role as a financial institution. They must ensure extremely high accuracy in their transaction processing and risk assessments. Additionally, operating across multiple African countries means complying with various regulatory frameworks, including GDPR and local data sovereignty laws, each requiring specific data governance and encryption standards.

Finally, the **value** dimension of HakiLend's data is solid. Their Big Data platform aims to enhance credit risk analytics by combining historical patterns with external credit ratings. It enables real-time fraud detection to protect against financial losses and ensures regulatory compliance through comprehensive audit trails and access controls. Looking at the complete picture

2. Big Data Architecture & Components (13 Marks)

HakiLend's Big Data Architecture

HakiLend's architecture is designed to integrate legacy systems, handle large-scale transactions, support real-time fraud detection, and ensure compliance with data sovereignty regulations. The proposed architecture consists of the following layers:

1. Data Sources Layer

- *Legacy Mainframe (COBOL):* Stores 20 years of historical transactional data.
- *Mobile Banking Stream*: Processes 20 million daily transactions.
- *Credit Agency APIs:* Integrates external credit risk data.
- *Customer Data:* Manages customer information across multiple countries.

2. Data Ingestion Layer

- *Apache NiFi*: Extracts, transforms, and loads (ETL) data from legacy systems and enforces data sovereignty rules.
- *Apache Kafka:* Handles real-time transaction streams and supports immediate fraud detection.

3. Data Storage Layer

- *HDFS Data Lake*: Stores raw structured, semi-structured, and unstructured data while ensuring compliance with sovereignty requirements.
- *Apache HBase*: Provides fast, real-time access to transactional data.
- *PostgreSQL*: Stores processed, structured data for advanced querying and reporting.

4. Data Processing Layer

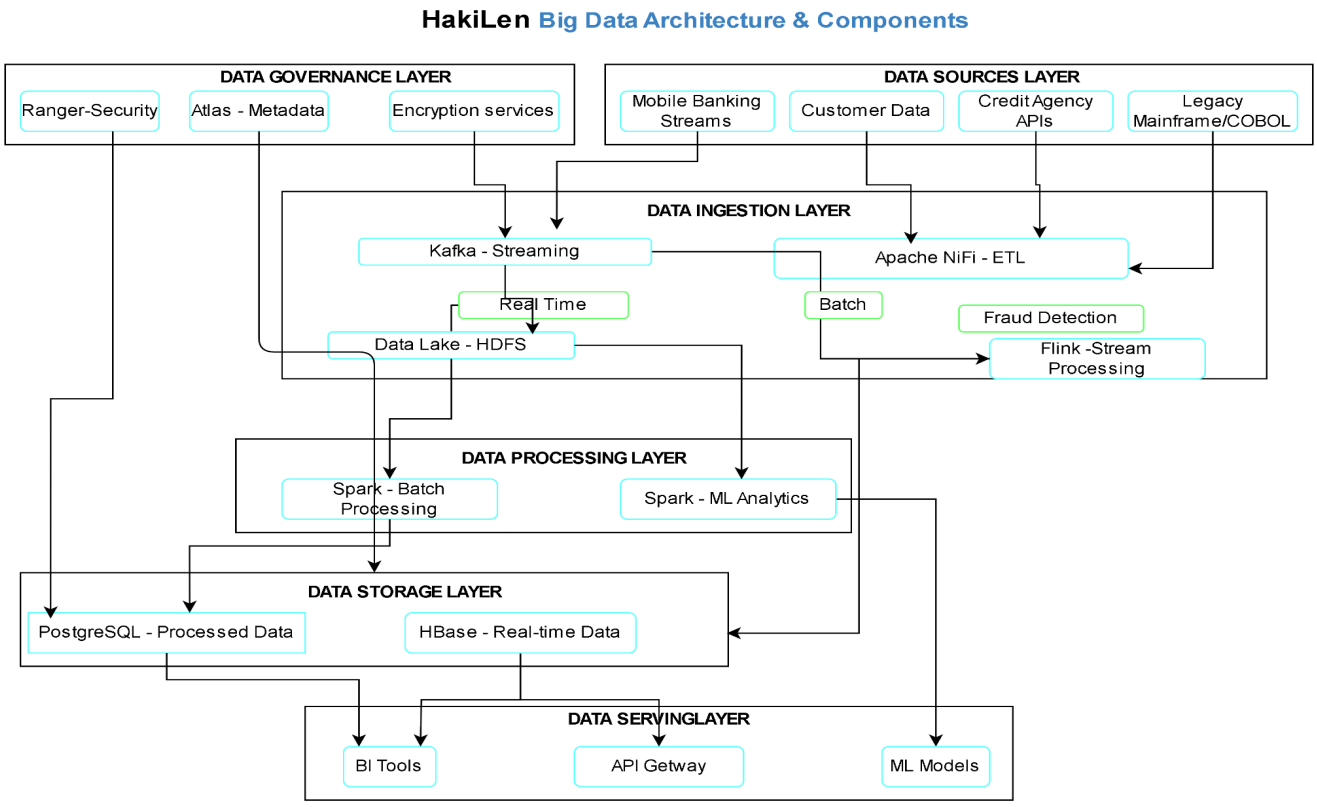
- *Apache Spark (Batch Processing):* Handles historical data processing for credit risk modeling and machine learning analytics.
- *Apache Flink (Streaming Processing):* Detects fraud in real-time by processing high-velocity transaction streams.
- *Spark ML*: Develops credit scoring models and customer behavior analysis.

5. Data Serving & Analytics Layer

- *BI Tools (Power BI)**:*** Provides business intelligence dashboards and analytics for risk assessment and operational insights.
- *API Gateway:* Facilitates integration with external services, such as third-party credit agencies and regulatory bodies.
- *ML Model Deployment*: Enables real-time scoring of loan applications and fraud detection.

6. Data Governance & Security Layer

- *Apache Atlas*: Manages metadata, ensures data lineage, and supports regulatory compliance.
- *Apache Ranger:* Implements fine-grained access controls and security policies.
- *Encryption Services:* Ensures secure data storage and transmission, complying with in-country data protection laws.



The above is the high-level Big Data architecture diagram for HakiLend. It visually represents the key components and their interactions, covering data ingestion, storage, processing, analytics, and governance.

3.Common Big Data Challenges in Banking

Challenge 1: Data Quality & Integration

***Problem*:** Banks often face difficulties in integrating data from legacy systems while ensuring data quality. For HakiLend, this means efficiently incorporating 20 years of COBOL-based data without compromising accuracy.

***Solution*:** A staged data validation strategy can streamline this process:

- Utilize Apache NiFi for initial validation and transformation.
- Implement automated data quality checks with robust error handling.
- Begin with a small dataset to test and refine transformation rules.

This incremental approach enables HakiLend's small team to establish dependable data pipelines over time.

Challenge 2: Regulatory Compliance Across Jurisdictions

***Problem*:** With operations spanning 20+ countries, HakiLend must navigate varying data sovereignty regulations while upholding consistent security protocols.

***Solution*:** Adopting a "Regulatory Compliance as Code" strategy ensures systematic management:

- Leverage infrastructure-as-code to deploy storage solutions tailored to each region's legal requirements.
- Develop automated tools to continuously monitor compliance.
- Centralize policy enforcement using Apache Ranger.

This method allows HakiLend's lean team to handle compliance efficiently without relying on manual oversight.

4.HakiLend's Justification for Big Data Solutions

1. ***Why Traditional Databases Are Insufficient***

With **20 million daily transactions**, traditional relational databases struggle to keep up with performance demands. Real-time analytics at this scale is impractical, and executing complex joins across vast historical and real-time datasets leads to prohibitive processing costs. Relational databases are ill-equipped to handle **diverse data formats**, particularly semi-structured credit agency reports that require flexible schema support. Additionally, processing **real-time streaming data** necessitates a different architectural approach that traditional databases lack.

2. ***Why Big Data Solutions Are Inevitable***

To stay ahead in the financial sector, HakiLend needs **real-time fraud detection** powered by modern streaming capabilities. Advanced **credit risk modeling** requires seamless integration with machine learning, and with competitors likely adopting similar technologies, falling behind is not an option. As **mobile banking adoption** continues to rise, transaction volumes will grow exponentially. New data sources—such as **IoT devices and social media**—will need integration. Additionally, **evolving regulatory requirements** demand a more adaptable data infrastructure. With AI and machine learning becoming industry standards, building a **scalable, future-ready** platform is essential. Modern Big Data solutions provide a superior **price-performance ratio**, allowing HakiLend to scale with a **pay-as-you-grow** model. Additionally, automation reduces maintenance overhead, making long-term operations more cost-effective. The transition to Big Data isn't just about solving immediate challenges—it's about creating a **resilient, scalable foundation** that drives banking innovation and sustains a competitive edge in an increasingly digital financial landscape.

6.YARN & Resource Management at HakiLend

YARN (Yet Another Resource Negotiator) efficiently manages HakiLend's shared cluster resources through a two-tier scheduling system, ensuring optimal performance across various teams and applications.

Two-Level Scheduling System

Resource Manager (Master):
As the central coordinator, the Resource Manager functions as the cluster's **traffic controller**, tracking available resources and distributing them across applications such as fraud detection, credit risk analysis, and marketing. It ensures fair allocation through a scheduling mechanism, preventing resource control by any single workload.

Node Manager (Workers):
Each worker node is monitored by the Node Manager, which tracks **CPU and memory usage**, reports health status to the Resource Manager, and manages application containers. This ensures efficient **local resource allocation** while maintaining overall system stability.

Key Features for Balanced Usage

- *Capacity Scheduling:* Ensures that each team has access to a guaranteed minimum resource allocation.
- *Fair Scheduler:* Distributes resources equitably across competing applications.
- *Queue Management:* Prioritizes critical workloads, such as fraud detection, over lower-priority tasks like marketing analytics.
- *Resource Isolation:* Prevents one team's resource-intensive tasks from negatively impacting others.
- *Dynamic Resource Allocation:* Adapts resource distribution in real-time based on workload demands, optimizing efficiency.

By leveraging YARN's robust resource management, HakiLend ensures **scalability, efficiency, and fairness** in handling its diverse and growing Big Data workloads.

7. Hadoop Ecosystem Tools at HakiLend

HakiLend leverages Hadoop ecosystem tools to streamline data processing, analytics, and automation, ensuring efficient operations with minimal manual intervention.

Apache Hive – Credit Risk Analytics

Hive allows HakiLend to **convert legacy COBOL data into a SQL-friendly format**, enabling business analysts to work with familiar query structures. It facilitates **historical transaction analysis**, risk assessment reporting, and complex aggregations spanning years of data. By integrating Hive, analysts can efficiently extract insights without requiring deep technical expertise in distributed computing.

Apache Oozie – Automated Data Pipeline Orchestration

Oozie automates HakiLend's **end-to-end data workflows**, ensuring seamless execution of critical processes such as regular **credit score updates, daily transaction processing, and regulatory report generation**. It efficiently handles workflow dependencies, ensuring that **legacy data migrations** and cross-system operations run smoothly without manual oversight.

By combining **Hive for data analytics** and **Oozie for workflow orchestration**, HakiLend achieves a **scalable, automated, and business-friendly** data ecosystem. These tools are well-documented, easy to manage for a small team, and offer enterprise-grade capabilities essential for a modern financial institution.

8.RDDs, DataFrames, and Datasets: Choosing the Right Abstraction for HakiLend

1. Comparison of Spark Abstractions

Feature	RDDs (Resilient Distributed Datasets)	DataFrames	Datasets
Data Structure	Low-level distributed collection	Tabular, structured format	Strongly typed, object-oriented
Type Safety	Compile-time safety	No compile-time safety	Compile-time safety (Scala)
Optimization	No built-in optimization	Catalyst optimizer	Catalyst optimizer
Memory Usage	Higher memory overhead	Lower memory footprint	Lower memory footprint
Ease of Use	Higher memory overhead	SQL-like syntax, easy for analysts	Object-oriented API, better for developers
Schema Support	No schema enforcement	Schema-based processing	Schema-based with static typing
Performance	Slower due to lack of optimization	Optimized query execution	Optimized with type safety
Best Use Case	Unstructured data, complex transformations	Structured data, SQL queries, analytics	Type-safe structured data processing (Scala)

2. Recommendation for HakiLend's Financial ETL

For HakiLend's ETL workflows, **DataFrames** are the optimal choice because:

- *Financial data is inherently structured*, making DataFrames a natural fit for transaction records and account details.
- *Built-in query optimizations e*nhance performance, reducing execution time for large datasets.
- *The SQL-like interface aligns with the finance team's expertise, e*nsuring easier adoption.
- *Lower memory usage* is essential given the high data volume HakiLend processes daily.
- *Simpler learning curve* allows the small development team to implement and scale ETL workflows effectively.

By leveraging **DataFrames**, HakiLend can achieve **efficient, scalable, and business-friendly data processing**, ensuring seamless integration with their existing analytics and reporting tools.

9. Tool Selection & Streamlining for HakiLend

To ensure HakiLend's small team can effectively manage the new Big Data platform, a **tiered approach** to tool adoption is recommended.

Tiered Approach

- *Core Tier:* Focus on **essential tools** that form the backbone of the architecture, such as **Apache Spark for processing, Apache Kafka for real-time streaming, and scalable storage solutions like HDFS or cloud-based options**.
- *Extension Tier:* Introduce **additional tools** only when a clear business need arises, ensuring minimal complexity while enhancing capabilities.
- *Experimental Tier:* Isolate and **test new technologies** before integrating them into production, preventing unnecessary disruptions.

Selection Criteria

HakiLend should prioritize tools that offer **strong documentation and active community support** for troubleshooting. A **managed service option** that would reduce operational overhead and integrate seamlessly with **existing tools and workflows**. Also provide a **clear upgrade and maintenance path** to avoid long-term technical debt.

Implementation Strategy

Start with core components to establish a stable foundation. **Incrementally add new functionality** based on business needs. **Use managed services** where possible to minimize operational burden. **Automate routine tasks** to enhance efficiency and reduce manual effort. **Emphasize self-service capabilities**, empowering non-technical users to extract insights without engineering dependencies.

Team Considerations

- Select tools with **overlapping skill sets** to simplify onboarding.
- Prioritize solutions with **intuitive UI and monitoring capabilities** for easier management.
- Ensure availability of **vendor support and community forums** for troubleshooting.
- Consider **training resources** to upskill the team and accelerate adoption.
- Opt for tools with **strong debugging and logging features** to simplify issue resolution.

By following this structured approach, HakiLend can **maintain a manageable, scalable, and cost-effective** Big Data ecosystem while gradually expanding capabilities to meet future demands.