University of Limerick

MSc in Software Engineering

AI Driven Compliance and Secure Data Sharing: Integrating Blockchain and Kafka for Cross Border Media Collaboration

Rohan Sikder
Student ID: 24165816

`https://github.com/rohansikder/MSc_FYP`

Supervisor:
Dr. Salim Saay

August 2025

Abstract

Cross-border media collaborations have many challenges in securely sharing data while meeting various international regulations. This problem is critical because non compliance with frameworks such as GDPR can result in legal penalties, reputational damage and disruption of media workflows. Previous research has explored AI for legal interpretation, blockchain for auditability and Kafka for real-time data streaming. However, these approaches are typically studied in isolation and there is limited work integrating them into a unified system. Existing solutions lack the ability to dynamically enforce compliance on data in motion while ensuring transparency and trust. This dissertation proposes a novel framework that combines Artificial Intelligence (AI), Apache Kafka and Hyperledger Fabric to automate compliance in cross-border media data exchange. The contribution lies in designing and evaluating a system that unites AI-driven policy interpretation, scalable event streaming and blockchain-backed auditability. A design science methodology was followed to develop a prototype. The AI module reads data protection policies (e.g., GDPR) and makes allow/block decisions in real time. Kafka allows low-latency event streaming, while Hyperledger Fabric records immutable logs and enforces compliance rules usings smart contracts. The prototype was tested in simulated multi-jurisdictional scenarios. Evaluation showed high compliance accuracy in blocking unlawful transfers and permitting lawful exchanges. Kafka ensures low decision latency and blockchain added transparency with moderate overhead. The results demonstrate that AI-augmented, blockchain-backed data pipelines can possibly enforce regulatory compliance while maintaining efficiency. The main takeaway is that integrating these technologies offers a transparent, scalable and secure approach to cross-border media collaboration. This work is practical proof of concept and shows future research directions, including advanced AI for legal interpretation, full Kafka–Fabric integration and scaling the system for real-world deployment.

Acknowledgements

"My success is only through Allah. Upon Him I have relied, and to Him I return."
(Qur'an 11:88)

All praise is due to Allah (Alhamdulillah), without whose guidance and blessings this work would not have been possible.

I would like to express my deepest gratitude to my supervisor, Dr. Salim Saay, for his invaluable guidance, encouragement and unwavering support throughout the course of this dissertation. His expertise and insights were crucial in shaping the direction of this work and his patience and constructive feedback helped me navigate challenges with confidence.

I am grateful to my family, whose love, support and belief in me have been my constant source of strength. Their understanding and encouragement, especially during the most demanding periods of this journey, made all the difference.

I would also like to thank my friends for their motivation, late night discussions and for simply being there. Their help provided balance and joy amidst the pressures of academic life.

This dissertation is as much a product of my own efforts as it is of the collective support of those around me. Thank you all.

FH + IH

Declaration of Authenticity

I, Rohan Sikder (Student ID: 24165816) declare that the work presented in this dissertation titled:

"AI Driven Compliance and Secure Data Sharing: Integrating Blockchain and Kafka for Cross Border Media Collaboration"

is entirely my own work, except where otherwise stated and acknowledged.

All sources of information have been appropriately referenced and acknowledged.

Name: Rohan Sikder
Student ID: 24165816
Date: August 2025

Signature: _____

Declaration: Consent for Publication

I, Rohan Sikder (Student ID: 24165816), give my permission for this dissertation to be made available for academic and research purposes by the University of Limerick.

Name: Rohan Sikder
Student ID: 24165816
Date: August 2025

Signature: _____

AI/GenAI Usage Statement

As part of this MSc dissertation, I acknowledge the use of Generative Artificial Intelligence (GenAI) in accordance with the University of Limerick's academic integrity policies and the guidelines for the use of AI/GenAI in assessments.

Specifically, my use of AI corresponds to **Level 2: AI-Assisted Idea Generation and Structuring**, as outlined by O'Sullivan et al. AI tools such as ChatGPT were used solely during the early stages of the dissertation process to assist with:

- Brainstorming ideas,
- Organizing themes and arguments,
- Structuring the overall flow of the document.

No AI-generated content has been included in the final dissertation. All writing, critical analysis and academic content are my own. AI was not used to write, paraphrase, summarize, or generate any content within the submitted work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern media production increasingly relies on collaboration between stakeholders operating across national borders. In such globally distributed environments, the secure and lawful exchange of media data-such as video files, metadata and sensitive user-generated content presents significant technical and regulatory challenges. Laws like the General Data Protection Regulation (GDPR) and the ePrivacy Directive impose stringent requirements on how personal data may be processed and transferred, particularly when it crosses jurisdictional boundaries. Failure to meet these requirements can lead to severe legal penalties, reputational damage and disrupted workflows .

This dissertation addresses the urgent need for a robust, automated and scalable framework that enables real-time, cross-border media collaboration while ensuring legal compliance and data security.

The research focuses on designing and evaluating a prototype that performs three key functions: (1) using AI to interpret and enforce privacy rules in real time; (2) employing Kafka for low-latency, event-driven communication; and (3) leveraging Hyperledger Fabric blockchain to provide immutable auditability and smart contract enforcement of legal obligations. By simulating international data exchanges between jurisdictions (e.g., Ireland, Germany and Belgium), the prototype demonstrates how data transfers can be evaluated dynamically for regulatory compliance and transparently recorded on a distributed ledger.

1.1 Scope

The scope of this project is to design and implement a secure, scalable and compliant data-sharing framework for cross-border media collaboration. This system will use:

- **Apache Kafka** for real-time data streaming across distributed systems.
- **Hyperledger Fabric** for smart contract-based compliance enforcement and immutable data traceability.

- **Artificial Intelligence** to automate interpretation and enforcement of data protection laws (e.g., GDPR, ePrivacy).

The system will simulate international data exchange scenarios involving multiple jurisdictions (such as Ireland, Germany and Belgium). The prototype will be evaluated based on compliance enforcement, latency, scalability and architectural soundness.

1.2 Research Questions

This dissertation is guided by the following research questions:

1. How can AI be used to interpret and automate enforcement of international data privacy regulations in real-time media exchanges?
2. What are the benefits and limitations of using blockchain smart contracts (Hyperledger Fabric) for automated legal compliance?
3. How effective is Apache Kafka in enabling scalable, low-latency and secure cross-border data transmission?

1.3 Identified Gaps and Research Motivation

The literature reviewed below highlights several strengths of AI, blockchain and Kafka for compliance, but also reveals critical gaps, particularly in their integration for cross border media data exchange. These are summarized below:

AI Application Gap

AI shows strong potential in interpreting legal texts and automating compliance checks [2, 3]. However, most implementations focus on static analysis such as document review or clause classification rather than in line, real time compliance decision making. There is a lack of systems where AI monitors data streams and determines compliance dynamically. Furthermore, trust in AI decisions remains a challenge due to issues with explainability and bias [4]. The gap lies in embedding AI driven policy interpretation directly into operational systems that handle data in motion, in a trustworthy and auditable way.

Blockchain Utilization Gap

While blockchain offers secure ledgers and smart contract based enforcement, there is a gap in applying it to media data sharing scenarios where laws intersect across copyright, privacy and jurisdictional restrictions. Additionally, most smart contracts assume static rules, but legal norms are dynamic and context sensitive. There is a need for blockchain based compliance architectures that are both

legally adaptable and aligned with data protection laws such as the GDPR, especially regarding the right to erasure [5, 6, 7].

Kafka (Streaming) Gap

Kafka provides real time streaming, but it has no built in concept of legal compliance. Enforcement is offloaded to applications consuming the data. There are no reference architectures where Kafka is tightly integrated with compliance engines or smart contracts capable of intercepting or regulating data flows in real time. Kafka lacks native compliance awareness for enforcing geographic, consent based or regulatory routing decisions [8, 9].

Integration Gap

Most crucially, current literature lacks a unified approach combining AI, blockchain and Kafka into a single, coordinated system. While pieces exist in industry or academia, these are not tightly integrated nor evaluated holistically. Current peer-reviewed literature lacks case studies or prototypes demonstrating a compliance by design data sharing platform that is intelligent (AI enabled), responsive (stream based) and accountable (blockchain audited) [10].

Research Motivation

Bridging these gaps motivates the research presented in this thesis. An integrated AI–Blockchain–Kafka system can:

- Automatically interpret and update policy rules using AI.
- Enforce these rules through smart contracts and programmatic checks in real time.
- Permanently record compliance outcomes or violations using an immutable ledger.

1.4 Research Methodology

This thesis adopts a *design science* methodology [11, 1], focusing on building and evaluating an artifact: a compliance-aware data-sharing platform that integrates Apache Kafka, Hyperledger Fabric and an AI/rules service.

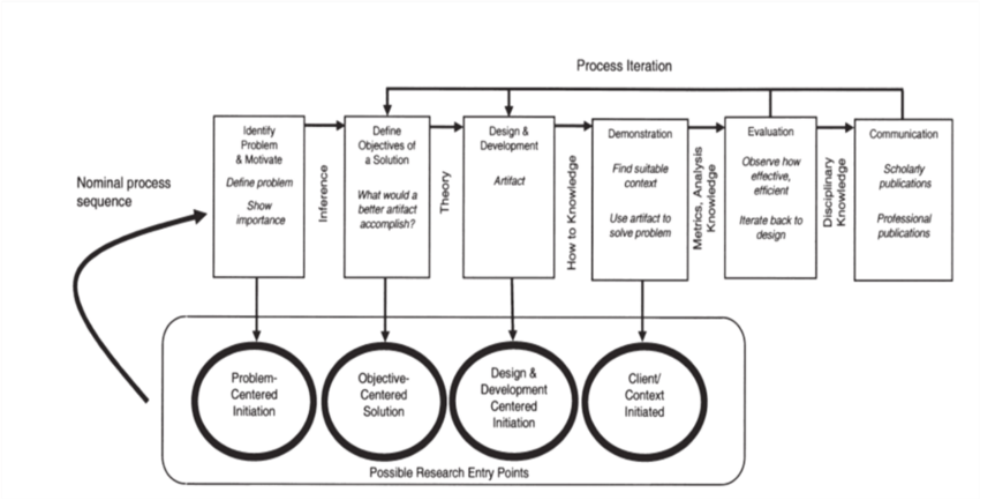

Figure 1.1: Design Science Research Methodology Process Model (adapted from Peffers et al., 2008). Source: [1]

Methodological Stages

Problem & Objectives Identify gaps in cross-border media data compliance and define objectives (real-time enforcement, auditability, secure routing).

Design & Development Specify the architecture and implement a working prototype (Kafka topics and flows, AI policy checks, Fabric smart contracts).

Demonstration Run end-to-end scenarios that simulate international exchanges (e.g., Ireland ↔ Germany/Belgium) with compliant and non-compliant events.

Evaluation Measure compliance accuracy, latency (Kafka in → AI decision → on-chain commit), throughput, fault tolerance (replay/recovery) and auditability (ledger queries).

Communication Report results, discuss trade-offs and limitations and derive design implications for real-world deployment.

1.4.1 Thesis Structure and Outline

This dissertation is organised as follows:

Chapter 2: Background and Related Work Reviews legal and technical foundations relevant to cross-border media data sharing, including GDPR/ePrivacy considerations and surveys prior work on AI for policy interpretation, blockchain for auditability and Kafka-based streaming architectures.

Chapter 3: Methodology Describes the design science approach, research design and evaluation plan, including problem framing, objectives, build–evaluate cycles and the metrics used to assess compliance, latency, throughput, auditability and security.

Chapter 4: Prototype Design and Implementation Presents the system architecture and implementation details: compliance modelling, AI/rules service, Kafka topics and flows, Hyperledger Fabric network and smart contracts and the security posture (TLS/mTLS, MSP).

Chapter 5: Evaluation Details the simulation scenarios (compliant transfer, missing consent, jurisdictional restrictions, anonymisation, load and fault tolerance) and reports quantitative/qualitative results against the stated metrics.

Chapter 6: Discussion and Limitations Interprets the results, discusses design trade-offs, governance and legal considerations (e.g., right to erasure), threats to validity and integration limits across AI, Kafka and Fabric.

Chapter 7: Conclusion and Future Work Summarises contributions, reflects on the viability of the approach and outlines directions for extending the prototype and engaging stakeholders for real-world deployment.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the research relevant to the secure and legally compliant exchange of media data across borders using new technologies. It focuses on three main domains: Artificial Intelligence (AI) for legal policy interpretation and automation [2, 12, 3], Blockchain (particularly Hyperledger Fabric) for compliance enforcement and auditability [13, 5, 6] and Apache Kafka for distributed real time data transmission.

Cross border data sharing presents different challenges due to differing data protection laws and data security requirements. Traditional systems are often not scalable or cannot dynamically enforce complex regulatory requirements. Recent advances in AI and distributed systems hint at potential solutions to these issues, but research frequently treats these technologies separately.

This literature review aims to:

- Explore how AI can be used to interpret and process legal documents
- Assess the impact of blockchain technologies on data governance and smart contract based enforcement
- Evaluate Kafka as a messaging backbone for scalable and secure media exchange

This review also identifies research gaps in both academia and industrial practice, highlighting the absence of integrated approaches integrating the 3 technologies in a compliance centric, real time data sharing environment. These gaps lay the foundation for the research questions and system design tackled in later chapters.

2.2 AI in Legal and Regulatory Compliance

2.2.1 Use of AI for Policy Analysis

Artificial Intelligence, specifically Natural Language Processing (NLP), is a powerful tool for analyzing complex legal and regulatory texts. These systems are capable of reading through legislative language, extracting obligations, identifying rights and classifying legal provisions relevant to compliance [2, 12, 14].

Transformer based models such as BERT and GPT have showed significant potential in understanding legal language, thanks to their ability to capture deep context and semantics from large corpora. LEGAL BERT (Legal Bidirectional Encoder Representations from Transformers), a domain specific adaptation of BERT, has shown improved performance in legal text classification tasks [2]. Studies like Aletras et al. [12] show how NLP techniques can be used to predict court outcomes based on textual input, while Lippi et al. [14] demonstrate the feasibility of detecting GDPR relevant clauses in terms of service agreements using machine learning.

The use of AI in policy analysis offers efficiency and consistency in interpreting regulatory frameworks across jurisdictions. However, challenges remain regarding explainability, algorithmic bias and the need for domain specific training data tailored to legal applications [4].

2.2.2 Automated Compliance Systems

Beyond text analysis, AI is increasingly used to support or automate compliance decision making. AI driven compliance systems assess an organisation's data processing activities against regulatory requirements, flag violations and recommend mitigation strategies. These systems often combine rule based logic with statistical models to apply legal standards, such as the GDPR to specific contexts.

Zetzsche et al. [3] discuss the application of AI in risk based compliance management, in particular within financial services. Enterprise platforms such as IBM OpenPages now integrate AI capabilities to streamline compliance workflows, automating tasks like obligation analysis, regulatory mapping and Data Protection Impact Assessments (DPIAs) in real time [10].

Despite these advances limitations exists, particularly around handling legal ambiguity and maintaining accountability for decisions. Human oversight remains essential in high risk scenarios, but the integration of AI into secure data sharing workflows signals a step toward more scalable and adaptive compliance infrastructures.

2.3 Blockchain and Smart Contracts in Compliance

2.3.1 Overview of Blockchain Technologies

"Blockchain is a distributed ledger technology that enables secure, transparent and tamper evident recording of transactions across a network of peers "[15]. At its core, a blockchain maintains a sequential chain of blocks, each containing a set of transactions that are linked to the previous block. This structure makes sure data integrity and resistance to manipulation is in place[16].

Blockchains are generally either public (permissionless) or private (permissioned). Public blockchains, such as Bitcoin and Ethereum, allow open participation and typically rely on consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS) [17]. While these networks emphasis decentralization and transparency, they often suffer from scalability limitations and high energy consumption. In contrast, permissioned blockchains restrict participation to verified entities, offering better performance, privacy and governance features that make them more appropriate for regulated domains [13].

Key attributes of blockchain systems include immutability, decentralized trust and auditability. These properties enable sturdy mechanisms for data provenance, consent verification and enforcement of compliance rules through smart contracts. However, blockchain's inherent immutability can conflict with privacy regulations such as the GDPR, particularly regarding the right to rectification and erasure (Articles 16 and 17) [5].

Despite these, the use of blockchain in compliance, critical applications continues to grow. Frameworks such as Hyperledger Fabric have been created to address enterprise needs by combining modular design, efficient consensus protocols and granular access control making them great candidates for secure, compliant data sharing infrastructures.

2.3.2 Hyperledger Fabric as a Compliance Tool

Blockchain has emerged as a mechanism for enforcing compliance through immutable data storage and transparent governance. Among the various platforms available, Hyperledger Fabric stands out due to its modular and permissioned design, which is particularly suited to enterprise and regulatory contexts.

Unlike public blockchains, Fabric operates within a consortium model where participants are known and authenticated. It uses *chaincode* smart contract logic deployed on private channels to enforce rules across distributed peers. This setup allows for both data segregation and accountability. Jagadeesh Sai et al. demonstrated the application of Fabric in identity and access management systems, where chaincode was employed to track and validate credential usage within a closed network. The system maintained an immutable audit trail of access events, providing "tamper evident logs" that can be cross referenced during compliance audits [18].

Healthcare applications further showcase Fabric's suitability for sensitive environments. In a study by Adlam et al., a permissioned blockchain was implemented to manage audit logs for Electronic Health Records (EHRs). The authors noted that Fabric's channel based architecture allowed for "privacy preserving audit capabilities," giving institutions the ability to maintain traceability without exposing patient level data to all participants [19]. Similar systems, such as ACHealthChain, extend this approach by combining modular consensus with policy based access control to ensure regulatory alignment while retaining system flexibility [20].

These examples reflect Fabric's ability to embed compliance at the infrastructure level, where data access, modification and retention rules are codified into the system itself.

2.3.3 Smart Contracts for Regulatory Enforcement

Smart contracts provide a way of automating the application of legal and regulatory norms. When implemented effectively, they can serve as dynamic gatekeepers verifying user rights, checking consent status and triggering alerts or actions based on compliance rules.

Islam et al. proposed a smart contract framework for consent management in healthcare that records consent metadata on chain, while sensitive data is stored off chain. Their architecture allows users to revoke access at any time and provides audit trails for data usage. In their model, contracts dynamically adapt to regulatory requirements, offering a scalable approach to managing rights under frameworks like the GDPR [21].

Despite these advancements, risks persist. The rigidity of smart contracts can become a liability in legal contexts that require flexibility or human judgment. John highlights several of these limitations, including the "inability to alter or revoke deployed contracts without administrative override," which may conflict with data subject rights under laws such as the GDPR or CCPA [22].

Efforts to address these flaws include the use of cryptographic tools like chameleon hashes or zero knowledge proofs. While they are promising, these solutions often introduce trade offs in transparency or decentralisation, raising further questions about trust and system design.

2.3.4 Blockchain vs GDPR: Legal Tensions

Blockchain's core attributes and data protection law particularly the GDPR has been well documented. A fundamental conflict arises from the GDPR's provision for data erasure (Article 17), which stands in opposition to the immutable nature of blockchain records [5, 6, 7].

Finck and Finck & Moscon argue that blockchain's inability to modify or delete historical data presents a structural compliance challenge, especially when personal data is involved [5, 6]. Their analysis suggests that while techniques such as off chain data storage or encryption can reduce legal exposure, they do

not fully resolve the conflict. These technical mitigations often rely on complex architectures that shift responsibility from the blockchain layer to auxiliary systems, thereby creating new risks.

A systematic review by Weber et al. reinforces this view. The study surveyed over a hundred publications and found that Articles 16 (rectification) and 17 (erasure) were the most frequently cited concerns in blockchain research. Solutions such as revocable encryption, metadata masking and hybrid storage models have been proposed, but none offer a universal fix [7].

Hyperledger Fabric, by virtue of being permissioned, does offer greater control over governance and accountability. Its design allows system architects to define and enforce data ownership and role based access policies. Nevertheless, the core issue of immutability remains. Unless future iterations of blockchain frameworks incorporate built in mechanisms for controlled data deletion, legal compatibility with the GDPR will continue to pose significant design and ethical challenges.

2.4 Apache Kafka for Real Time Data Sharing

2.4.1 Event Driven Architectures

Modern distributed systems increasingly rely on event driven architecture (EDA) to respond to real time data changes. Instead of operating through direct service calls, components react to "events" state changes that are published to a central stream. This model enables systems to be decoupled, scalable and highly responsive.

Apache Kafka plays a central role in many of these architectures. Originally developed at LinkedIn, Kafka has evolved into a widely adopted platform for managing real time data streams. It acts as a high throughput, fault tolerant event broker that captures, stores and distributes event data across services [23]. This is particularly useful in domains like finance, where operations such as trade confirmations or fraud detection must be processed instantly and consistently.

For example, Kafka can be paired with business process engines like Camunda BPM to orchestrate regulatory workflows. Events such as "transaction completed" or "threshold breached" can automatically trigger downstream compliance checks, reporting tasks or alerts, all without manual intervention. This reactive design not only increases efficiency but also improves auditability, as each action is logged and traceable across the system [24].

2.4.2 Kafka in Regulated or Distributed Systems

Kafka's architecture makes it well suited to environments where data integrity, security and traceability are non negotiable such as healthcare, finance and public services. It offers encryption in transit using TLS, support for authentication via SSL or Kerberos and fine grained access control through built in ACLs [25]. These features allow organisations to build secure, compliant pipelines for sensitive data.

In financial services, Kafka is often used to capture and retain trading data, transaction metadata and system logs all of which are critical for regulatory audits under frameworks like GDPR, PCI DSS and MiFID II [9]. Its ability to replay historical events also aids in post incident analysis, helping teams trace exactly what happened, when and why.

Several comparisons between Kafka and traditional message brokers like RabbitMQ highlight Kafka's strength in handling large scale, high throughput workloads. Its persistent log based design allows consumers to reprocess or audit data from any point in time, offering both flexibility and accountability key concerns in regulated industries [8].

2.4.3 Kafka + Blockchain Integration

While Kafka provides speed and scalability, it doesn't offer inherent immutability. Blockchain, on the other hand, guarantees data cannot be tampered with once written. Integrating the two allows systems to benefit from the strengths of both technologies.

There are several ways Kafka and blockchain can be combined:

- **Stream to Chain Pipelines:** Kafka topics can stream data (e.g., transaction records or consent events) that are later hashed and written onto a blockchain. This creates a verifiable audit trail without slowing down the system.
- **Blockchain Triggers:** Kafka events can be used to trigger smart contracts or on chain actions. For example, a suspicious login event might be streamed through Kafka and, if certain criteria are met, invoke a smart contract that locks an account or flags it for investigation.
- **Kafka Native Ledgers:** Some architectures attempt to simulate blockchain properties inside Kafka itself by chaining messages with cryptographic hashes. While not a true blockchain, this can provide lightweight immutability for use cases where formal consensus isn't required.

One real world implementation is PIEChain, which uses Kafka to coordinate auctions across multiple blockchains, ensuring atomic settlement through event based triggers [26]. Another example comes from Confluent's work on real time blockchain data integration, where Kafka Streams is used to process and validate transactions before anchoring them onto a distributed ledger [27]. These hybrid designs offer the responsiveness of Kafka with the trust guarantees of blockchain an increasingly valuable combination in compliance heavy sectors.

2.5 Cross Domain Integrations

Having examined AI, blockchain and Kafka individually, it is evident that each addresses different aspects of the secure data sharing problem. A natural question

is how these technologies can be combined to produce a more powerful, comprehensive solution. An integrated approach could, in theory, use AI to understand and encode legal policies [2, 4], blockchain to enforce and record compliance [21], and Kafka to handle the real time data flows [26, 27].

In practice, however, research that bridges all three domains is sparse. Most existing work and implementations incorporate at most two of the three technologies and often in a limited scope.

One area of integration is using AI in conjunction with blockchain based compliance rules. For instance, researchers have suggested using NLP techniques to translate legal texts (regulations, contracts, service level agreements) into formal representations or even directly into smart contract code [21]. This crosses the AI/blockchain boundary: the AI would interpret complex human language and the blockchain would execute and enforce the extracted rules. A conceptual example might be an AI system that reads data sharing agreements or privacy laws and produces a set of logical rules. Those rules could then be deployed as smart contracts on a platform like Hyperledger Fabric, which automatically check each data transaction against the rules (e.g., "if data category = medical and destination country = X, then ensure consent Y is present; otherwise block").

There have been early efforts in computable contracts and logic based regulatory compliance that echo this idea, but they tend to remain theoretical or at the prototype stage [4]. The difficulty lies in guaranteeing that the AI's interpretation is correct and legally sound, since errors in translation to code could have serious consequences. Moreover, legal language often requires context that an AI might miss without human input.

Another intersection is between AI and Kafka in the context of streaming compliance monitoring. Real time data streams can be analyzed with machine learning to detect anomalies or potential compliance breaches. For example, an AI model could monitor Kafka topics for patterns that resemble a data leak or an unauthorized access. Kafka provides the real time pipe and buffering, and AI provides the analytical layer overlaying it [10].

Kafka and AI can also enhance blockchain by serving as an oracle mechanism. In blockchain parlance, oracles are trusted data feeds that inform smart contracts about external facts. Kafka could carry processed results from AI systems (e.g., "AI has classified this document as containing personal data of type X and subject to law Y") into the blockchain environment. A smart contract could then decide what to do with a transaction, based on that information. In this way, Kafka and AI together act as an intelligent bridge between the off chain world and on chain enforcement [26, 27].

A few industrial solutions are beginning to touch on combining these elements. For example, IBM integrates AI in governance platforms like OpenPages and experiments with Hyperledger for audit logs [10]. In some supply chain or finance scenarios, IoT data is streamed via Kafka, analyzed by AI and verified through blockchain. However, these systems are usually modular and not deeply integrated.

One of the biggest gaps in cross domain integration is the absence of ref-

erence architectures or case studies explicitly uniting AI, blockchain and Kafka in a regulatory compliance context [8]. Most research isolates variables: one paper studies AI for contract parsing, another examines blockchains for consent tracking and another assesses Kafka's performance. Tri component systems are mostly discussed in visionary whitepapers or exploratory frameworks, not in detailed academic evaluations.

2.6 Summary

This chapter surveyed the current state of research in three technological domains relevant to secure, compliant cross border data sharing:

- **AI for Legal Compliance:** AI can interpret legal language and automate aspects of compliance, especially through NLP [2]. While promising, current applications are mostly static and challenges such as explainability persist [4].
- **Blockchain and Smart Contracts:** Blockchain enables immutable logs and automated enforcement. Hyperledger Fabric's modular and permissioned architecture makes it suitable for compliance contexts [13]. However, legal tensions with data protection laws such as the GDPR must be carefully managed [5, 7].
- **Apache Kafka:** Kafka provides high throughput, real time data streaming with support for auditability and fault tolerance [9]. It is widely used in regulated industries but lacks native enforcement or awareness of compliance rules.

The chapter identified critical gaps in applying each technology to media data compliance and highlighted the lack of integrated systems that combine all three. Cross domain integration remains largely theoretical, with minimal implementation in academia or practice. These gaps establish the need for a compliance by design architecture capable of adapting to evolving legal and operational conditions.

The next chapter introduces methodology for designing and evaluating such a system one that brings together AI, blockchain and Kafka into a unified solution for secure, auditable and policy aware media data exchange across jurisdictions.

Chapter 3

Methodology

This chapter details the methodology employed for designing and evaluating a secure, scalable and compliant data-sharing framework for cross-border media collaboration. The research adopts a design science methodology, focusing on the creation and assessment of a functional prototype.

3.1 Research Design: Design Science Approach

This research uses a **design science methodology** [11], which focuses on building and evaluating an innovative artifact to solve a real-world problem. This approach is particularly suitable because the goal is to create a working system a prototype that enables compliance-aware data sharing across international borders. The process followed stages of problem identification, objective definition, design and development, demonstration, evaluation and communication.

Earlier stages identified gaps in cross-border media data compliance and defined objectives like real-time enforcement and auditability. This methodology emphasizes an **artifact-centric approach**, allowing iterative prototyping and refinement through a build-and-evaluate loop. Building a functional prototype directly tests how emerging technologies AI, blockchain and Kafka can be orchestrated to meet compliance requirements, providing both a concrete solution and generalizable insights.

3.2 Prototype Development and Technology Selection

The developed artifact is a compliance-aware data sharing platform integrating three core technologies. Each was selected for specific strengths in addressing facets of the problem of cross-jurisdictional media collaboration:

- **Hyperledger Fabric (Blockchain)**: Chosen for its **permissioned, modular architecture** and smart contract support. Unlike public blockchains,

Fabric offers robust privacy controls and identity management vital for sensitive media data under regulatory constraints. It allows for a private consortium network where only authorised organisations (e.g., media partners, regulators) participate, ensuring data logs remain confidential while benefiting from **immutability and transparency**. Fabric's ability to execute chaincode (smart contracts) facilitates encoding legal rules directly into transaction logic and offers high transaction throughput for enterprise settings.

- **Apache Kafka (Event Streaming)**: Selected as the **backbone for real-time, event-driven communication** across distributed components. Kafka's high throughput and low latency make it ideal for large volumes of media data that need instant cross-border sharing. Kafka also acts as an integration layer, decoupling data producers and consumers and bridging AI and blockchain components.

- **Artificial Intelligence (NLP and Rule Processing)**: Employed to **interpret complex regulations and automate compliance decisions**. An NLP model analyses policy text and data content to determine applicable rules. AI techniques handle multilingual regulations and nuances, monitoring data flows for anomalies or violations faster than manual oversight.

Integrating these technologies prioritised **loose coupling** through Kafka, allowing each component to evolve or scale independently. The AI component subscribes to Kafka topics and invokes blockchain transactions when compliance checks or logs are needed.

3.3 System Requirements

3.3.1 Functional Requirements (FR)

FR-1: Event Ingestion Ingest media metadata as Kafka messages on an input topic (e.g., `media-events`).

FR-2: AI Decisioning Classify each event as `approved`, `blocked`, or `flagged` with a short reason code; publish to `ai-decisions`.

FR-3: On-chain Logging Write every decision to a permissioned Fabric ledger via chaincode (store `eventId`, `decision`, `reason`, `timestamp`).

FR-4: Access Control Only authenticated identities (Fabric MSP) can invoke chaincode.

FR-5: Conditional Delivery Consumers process only `approved` events (gated by a compliance topic or flag).

FR-6: Query & Audit Support `queryDecision(eventId)` and `getAllDecisions()` for audits.

3.3.2 Non-Functional Requirements (NFR)

NFR-1: Decision Latency Kafka in to decision out ≤ 100 ms under nominal load.

NFR-2: Ledger Latency On-chain commit per decision ≤ 2 s (baseline).

NFR-3: Throughput Sustain ≥ 100 events/s ingestion in lab conditions without loss.

NFR-4: Resilience No message loss on service restarts; consumers can replay from offsets.

NFR-5: Auditability Every decision is immutably recorded and retrievable with timestamps and reasons.

3.3.3 Security Requirements (SEC)

SEC-1: Transport Security Kafka uses TLS (mTLS) and ACLs for topic access.

SEC-2: Identity Fabric MSP enforces authenticated, attributable transactions.

SEC-3: Least Privilege Only authorized organisations/roles can access channels and invoke chaincode.

3.3.4 Compliance & Policy Requirements (COM)

COM-1: Rule Model Represent policies as {`DataType`, `SenderJurisdiction`, `ReceiverJurisdiction`, `Conditions`, `Action`}.

COM-2: Two-Step Verification AI decisions are validated/finalized on-chain; rejections include explanations.

COM-3: Jurisdictional Controls Enforce rules such as "DoNotExport" before delivery.

3.3.5 Data & Interfaces (DI/INT)

DI-1: Event Schema Minimum fields: `event_id`, `data_type`, `sender`, `receiver`, `consent` (boolean). Optional anonymization and legal-basis flags.

DI-2: Topics Use at least `media-events` (ingress) and `ai-decisions` (egress); optionally `compliance-events` for anchoring/gating.

DI-3: Chaincode API Provide `recordDecision`, `queryDecision` and `getAllDecisions`.

3.3.6 Deployment & Environment (DEP)

DEP-1: Containers Kafka/ZooKeeper (and optional UI) via Docker Compose; Fabric network per Fabric 2.x lifecycle.

DEP-2: Tooling Python 3 for producers/consumers/AI; Node.js for chaincode; Fabric CLI binaries available.

DEP-3: Secrets Provision CA certs, MSP directories and client wallets securely to gateway/CLI hosts.

3.3.7 Verification

Acceptance is achieved when all FR, NFR, SEC, COM and DI/INT items pass across the simulation scenarios (compliant transfer, missing consent, jurisdiction block, anonymization, load and fault tolerance) with complete on-chain audit records.

The verification criteria ensure that all functional, non-functional, security, compliance and deployment requirements are satisfied under the defined simulation scenarios. To realise these requirements in practice, the system is designed according to an *event-driven architecture* (EDA), which provides the scalability, resilience, and real time responsiveness needed for compliance aware data sharing. The following section introduces this architectural paradigm and its relevance to the proposed solution.

3.4 Event-Driven Architecture

The system requirements outlined above are implemented within an event-driven architecture (EDA) pattern. In this paradigm, producers publish events to a broker, which routes them asynchronously to subscribing consumers. This model is particularly well suited to real time compliance enforcement because it decouples producers and consumers, supports high throughput and enables resilience through replayable event logs.

A generic event-driven architecture model is shown in Figure 3.1. In the context of this thesis, Apache Kafka serves as the event broker, mediating interactions between media event producers, the AI-based compliance checker and the Hyperledger Fabric ledger for immutable decision logging.

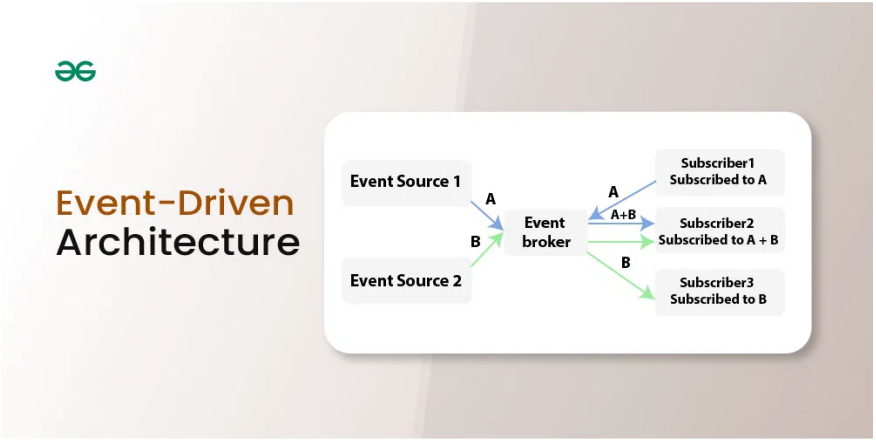

Figure 3.1: Conceptual Event-Driven Architecture (adapted from GeeksforGeeks, 2023).

3.5 System Architecture Overview

The prototype system is structured as a multi-layered, event-driven architecture that connects participants in different jurisdictions through a compliance-mediating core. Key components include:

- **Data Producer and Consumer Applications**: Simulate media collaborators in different countries (e.g., Ireland, Germany, Belgium).
- **Apache Kafka Cluster**: Handles messaging and real-time data streaming.
- **AI-driven Compliance Service**: Intercepts events from Kafka, invokes the AI/NLP module to interpret applicable compliance policies and formulates decisions (allowed/blocked/flagged).
- **Hyperledger Fabric Blockchain with Smart Contracts**: Receives transaction proposals from the compliance service. Smart contracts enforce compliance logic, rejecting or approving transactions.

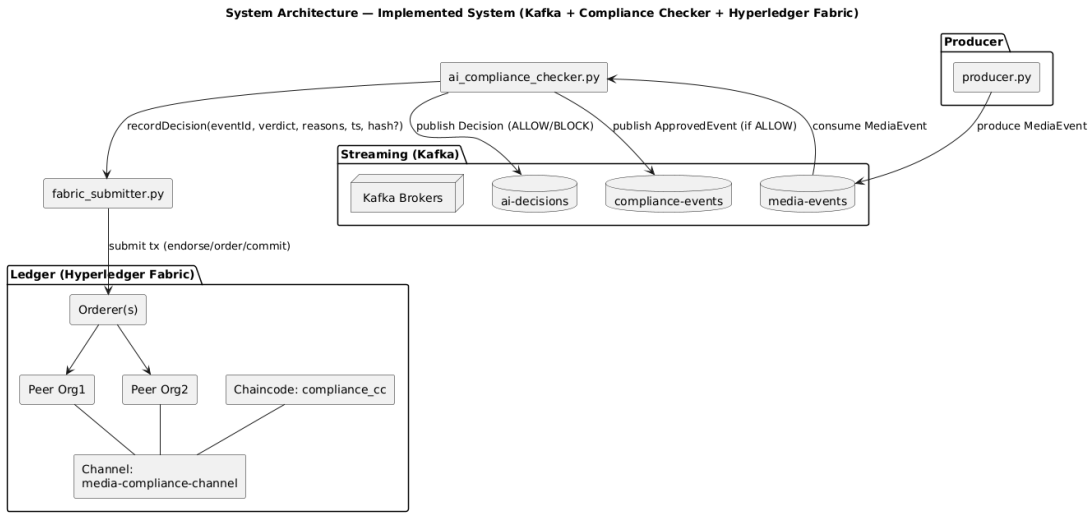

Figure 3.2: System Architecture — Implemented system (Kafka + Compliance Checker + Hyperledger Fabric).

The flow ensures that every data exchange is checked against compliance rules and recorded immutably before reaching the recipient. All Kafka communications are secured via SSL and Fabric's MSP ensures transaction attribution.

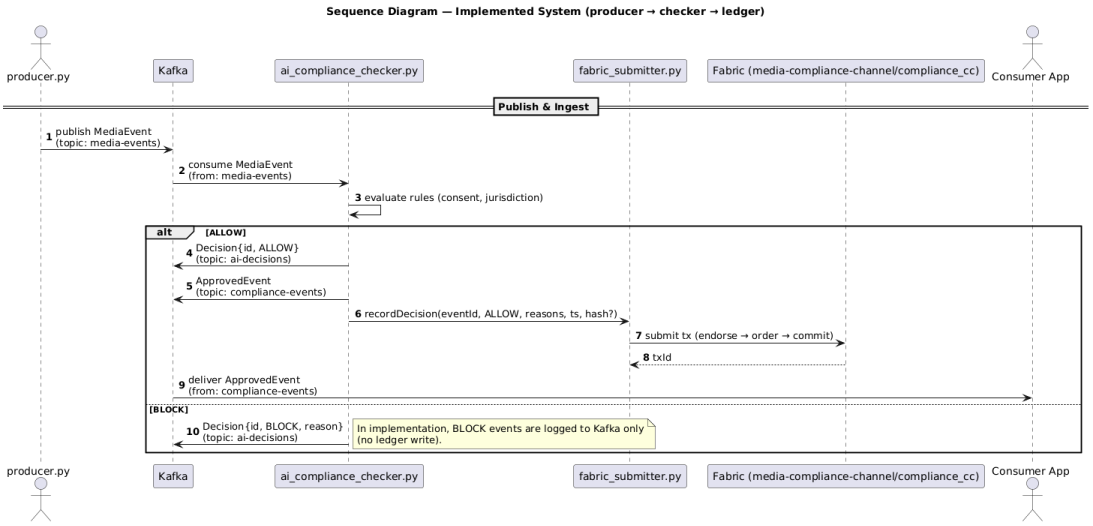

Figure 3.3: Sequence diagram — Implemented flow from `producer.py` to `ai_compliance_checker.py` to `fabric_submitter.py` and ledger.

3.6 Compliance Modeling and AI-Driven Policy Interpretation

This section covers compliance abstraction and the AI methods for real-time interpretation:

- **Compliance Modeling Strategy**: Abstracted into structured rules: {DataType, SenderJurisdiction, ReceiverJurisdiction, Conditions, Action}, enforced in chaincode and AI logic.
- **AI-Driven Policy Interpretation**:
  - *Parsing Legal Texts*: NLP extracts machine-readable rules from legal documents (e.g., GDPR).
  - *Classifying Data Events*: AI tags content (e.g., identifying personal data) and checks attributes against policy rules.

AI decisions are passed to the blockchain for final validation, enabling **two-step verification**. Decisions, especially rejections, are recorded with explanations to support accountability.

3.7 Event-Driven Communication with Apache Kafka

Apache Kafka serves as the **nervous system** of the prototype:

- **Kafka as the Data Pipeline**: Topics represent project-specific channels. Kafka offers reliable storage and ensures availability even if consumers go offline.
- **Integration via Kafka**: Kafka enables asynchronous communication. The compliance service acts as a Kafka consumer and producer.
- **Secure Delivery**: Kafka enforces security through SSL mutual authentication and ACLs. Conditional consumption ensures consumers process data only if approved via a dedicated compliance topic.

3.8 Blockchain for Immutable Logging and Smart Contract Enforcement

Hyperledger Fabric ensures trust and transparency:

- **Network Setup**: A permissioned Fabric v2.x network with four organisations. A shared channel `media-compliance-channel` maintains a unified audit trail.
- **Smart Contract Design**: Chaincode (Go) implements `SubmitDataEvent` which:
  - Authenticates submitters.
  - Evaluates rules.

– Rejects or commits the event.

- **Immutable Audit Logging**: Time-stamped, tamper-evident entries support audits. Query functions support retrieval.
- **Cross-Border Enforcement**: Jurisdiction-specific rules (e.g., "DoNot-Export" policies) are enforced via smart contracts.

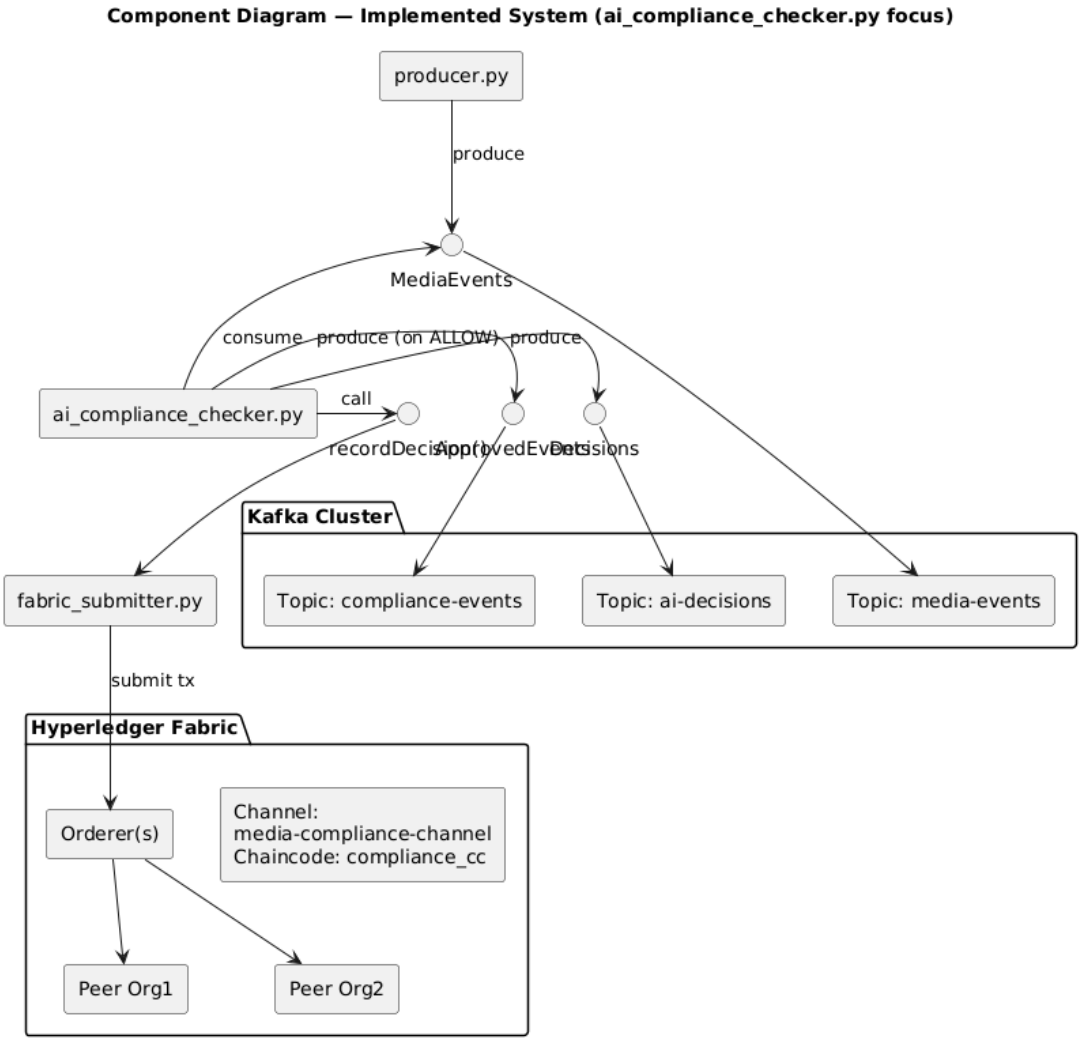

Figure 3.4: Component diagram — Implemented system focusing on `ai_compliance_checker.py`.

3.9 Simulation and Test Scenarios

Tests validate system functionality under realistic conditions:

- **Compliant Data Transfer**: Tests correct processing of compliant events.

- **Missing Consent**: Verifies blocking and logging of events missing proper consent.
- **Jurisdictional Restriction**: Ensures enforcement of local rules.
- **Data Anonymization**: Tests conditional approval based on metadata flags.
- **Throughput and Load**: Measures system under stress.
- **Fault Tolerance**: Simulates failures to assess resilience and recovery.

Monitoring tools (Kafka metrics, Fabric explorers) were used to observe and analyze system behavior.

3.10 Evaluation Metrics and Criteria

Evaluation aligns with research goals:

- **Compliance Accuracy**: Measured by True/False Allow and Block Rates.
- **Latency**: End-to-end, compliance decision and blockchain commit time.
- **Throughput and Scalability**: Events per second and resource utilization.
- **Auditability and Transparency**: Log completeness, query capabilities and intelligibility.
- **Security and Integrity**: Access control, integrity checks and recovery.

3.11 Limitations

- **AI Generalizability**: AI is limited to predefined regulations and requires maintenance.
- **Simplified Legal Logic**: Real-world laws are more nuanced than binary rules.
- **Prototype Scale**: Evaluation was in a lab environment and doesn't represent production scale.
- **Integration Complexity**: Combining AI, Kafka and blockchain adds risk of edge-case errors.
- **Real-World Deployment**: Governance, legal acceptance (e.g., right to be forgotten) pose unresolved challenges.

- **Evaluation Gaps**: Some metrics are qualitative and lack external stakeholder validation.
- **Technology Constraints**: Fabric's complexity, Kafka replication limits and NLP resource demands are barriers to scale.

In summary, this prototype offers a viable proof-of-concept, but full deployment requires further engineering, legal review and stakeholder collaboration.

Chapter 4

Implementation

This chapter details the full-stack development of a distributed prototype that enforces data compliance across international media collaborations. It integrates Apache Kafka for scalable event streaming and Hyperledger Fabric for secure, permissioned ledger enforcement. A key design goal was modularity, enabling parts of the system (e.g., the AI module or Fabric chaincode) to function independently while supporting eventual full integration.

4.1 System Design Rationale

The system was designed with three primary aims. First, it needed to simulate realistic cross-border media transfers, such as sending video files from Germany to Ireland, in order to capture the practical complexities of international data flows. Second, these simulated transfers were evaluated against compliance policies, with particular emphasis on regulations like the GDPR. Finally, all compliant decisions were immutably recorded on a distributed ledger, ensuring traceability and accountability across jurisdictions.

To achieve this, an event-driven architecture was selected. Event streaming decouples components, allowing for:

- **Scalability**: Kafka's ability to handle thousands of events per second
- **Fault tolerance**: Consumers and producers operate independently
- **Extendibility**: AI, analytics, or logging systems can be added later

The enforcement layer was handled by Hyperledger Fabric, chosen for its modular architecture, MSP-based identity model and smart contract capabilities.

4.2 Kafka System Implementation

Apache Kafka served as the backbone for simulating data events. It included:

Topics Setup

The Kafka system was configured with three core topics to support event-driven communication. The media-events topic carried incoming media metadata, acting as the primary entry point for all simulated data transfers. The ai-decisions topic received the results of compliance evaluations from the AI component, ensuring that decisions were available for downstream processes. Finally, a compliance-events topic was planned to forward valid events directly into Hyperledger Fabric, closing the loop between real time decision-making and immutable logging.

Producer Logic

Implemented using Python's `kafka-python`, the producer generated randomized media event payloads. Each message included:

```
{
  "event_id": "001",
  "data_type": "personal",
  "sender": "Belgium",
  "receiver": "France",
  "consent": true
}
```

This mimics metadata sent during real media file transfers.

Consumer and Compliance AI Simulation

The consumer component incorporated a lightweight rule engine that simulated AI or NLP-based compliance decision-making. Each event was inspected according to predefined conditions. If consent was required but missing, the event was automatically rejected. Similarly, if the jurisdictions involved in the transfer violated regulatory restrictions, the event was blocked. Events that satisfied all compliance criteria were approved for further processing.

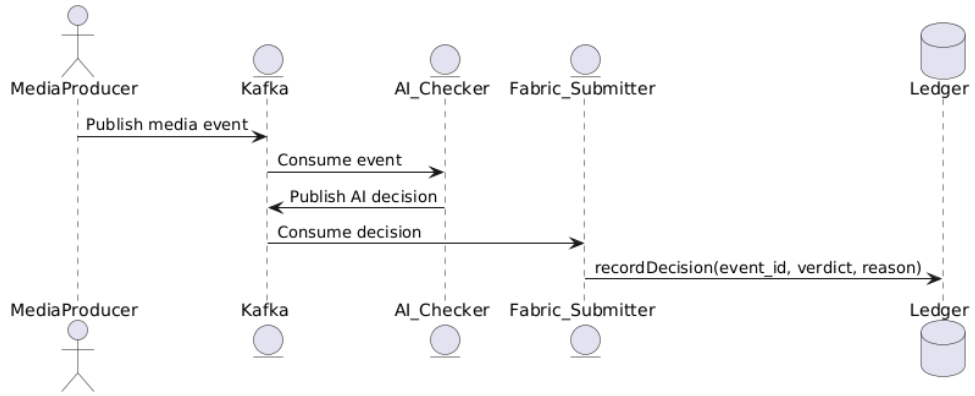

Figure 4.1: Message flow from Kafka producer to ledger (planned full integration)

The decision was published to `ai-decisions` or logged.

4.3 Chaincode (Smart Contract) Design in Fabric

The smart contract (compliance.js) was implemented using Fabric's Node.js SDK and included three principal functions. The recordDecision function immutably stored AI outputs on the blockchain, ensuring transparency and accountability. The queryDecision function enabled the retrieval of decision metadata based on a specific event identifier, while the getAllDecisions function provided bulk access to compliance records for auditing and regulatory reporting.

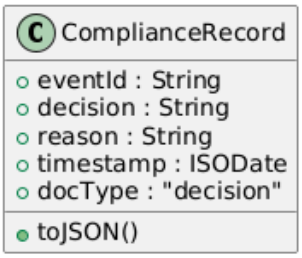

Figure 4.2: Example ledger state for a compliance record

Design Reasoning

- **Why Fabric?** For identity-based permissions, immutability and audit history.
- **Why Node.js?** Matches Kafka logic (also in JavaScript/Python) and simplifies JSON handling.
- **Why Use Ledger?** For accountability, to prove decisions were made transparently.

4.4 Chaincode Lifecycle Deployment

Figure 4.3: Chaincode lifecycle stages in Hyperledger Fabric

This CLI interaction allowed testing without needing a full SDK client during development.

Complexity Observed

- Required sequence alignment per org
- Errors like "sequence mismatch" due to improper versioning
- High sensitivity to directory structure and cert paths

4.5 Fabric Gateway Integration Attempt

To automate writes from Kafka to Fabric, a Node.js gateway client was developed using 'fabric-network':

- `enrollAdmin.js` - enrolls an admin via Fabric CA
- `registerUser.js` - registers a user, placing identity in a wallet

This required a ‘connection-org1.json‘, which was missing. It was generated via:

```
node ccp-generate.js
```

Despite successful generation, the path used by the enrollment script didn’t match where the file was created, leading to repeated failures like:

```
Error: ENOENT: no such file or directory,
open '.../test-network/organizations/.../connection-org1.json'
```

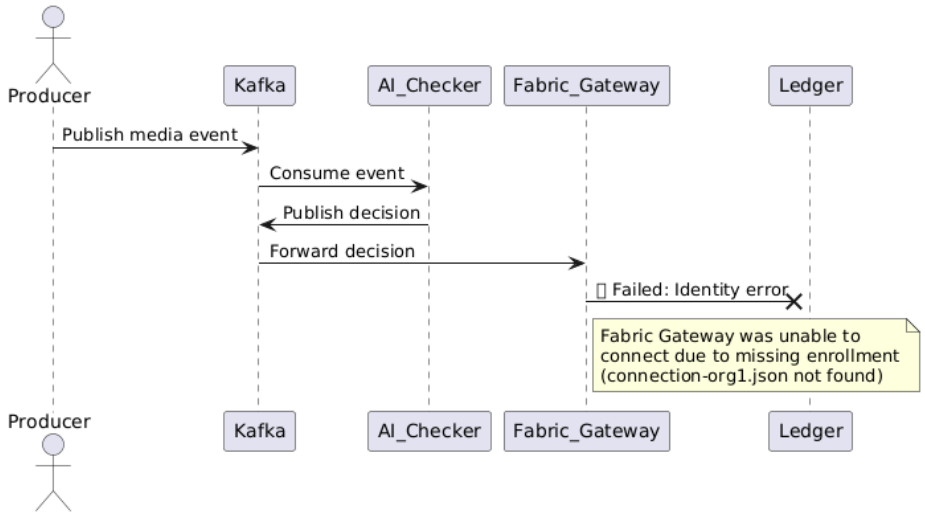

Figure 4.4: Kafka-Fabric integration attempt with failed identity enrollment

4.6 Final Architecture (As Implemented)

The final architecture incorporated a complete Kafka pipeline, connecting the producer, AI decision engine, and logging mechanisms. The chaincode was fully deployed and could be invoked manually through the Fabric CLI, while the Fabric gateway client was scaffolded but remained incomplete due to unresolved identity enrolment issues.

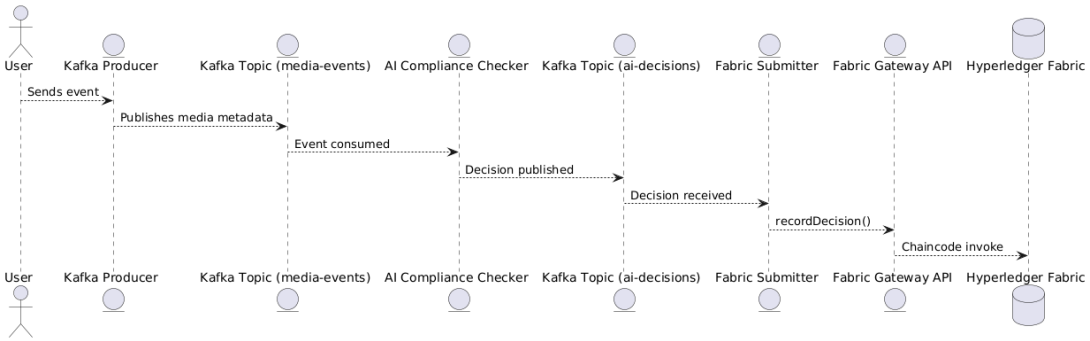

Figure 4.5: Sequence diagram of compliance-aware data sharing prototype

Table 4.1: Component Status Overview

Component	Status	Notes
Kafka Topics	✓Complete	Real-time data simulation
Kafka AI Consumer	✓Simulated AI	Rule-based compliance logic
Fabric Chaincode	✓Deployed	Manual invoke via CLI
Fabric Gateway Client	×Partial	Registration failed at identity step

4.7 Technical Complexity and Design Trade-offs

Kafka

Kafka offered several advantages in the prototype implementation. It was relatively simple to deploy using Docker Compose, making it well suited for rapid prototyping and event simulation. The availability of command line tools also provided strong support for debugging and monitoring. However, Kafka lacks native identity management and has no built in compliance enforcement capabilities, meaning that it had to be integrated with Fabric and the AI compliance engine to provide regulatory safeguards.

Hyperledger Fabric

Hyperledger Fabric also demonstrated both strengths and limitations. Its permissioned ledger model was particularly well-suited for compliance-sensitive environments and its modular design allowed chaincode to remain isolated from underlying infrastructure. On the other hand, the lifecycle was complex, requiring multi-organisation approval and strict version management. Deployment was further complicated by a rigid folder structure and a heavy SDK integration process, which required multiple certificates, wallets and configuration files.

Integration Complexity

This system required the following skills:

- Docker and container networking
- Bash scripting and CLI orchestration
- Kafka CLI and Python API
- Smart contract design
- Fabric certificate authority and identity modeling

4.8 Summary

The implemented system is modular, extensible and aligns with real-world architectures in data compliance. Kafka enables efficient event-driven processing, while Fabric provides the ledger and contract enforcement layer. While Fabric gateway integration faced challenges, the prototype effectively validates core principles of stream-based compliance enforcement and highlights a clear roadmap for full automation and AI integration.

Chapter 5

Evaluation

This chapter evaluates the prototype against the simulation scenarios and metrics defined in Chapter 3. The goal is to assess how effectively the system meets its compliance objectives while maintaining performance. Key criteria include compliance accuracy, latency, throughput (and scalability), transparency (auditability) and security.

The tests involved streaming simulated media events across jurisdictions (Ireland, Germany, Belgium), with the AI module enforcing rules such as consent requirements and jurisdictional restrictions. Each scenario (compliant transfer, missing consent, jurisdiction block, data anonymisation, etc.) was executed to gather both functional outcomes and quantitative measurements.

Compliance Accuracy

The prototype achieved 100% accuracy in applying the intended policies during simulation. All events that violated rules (e.g. missing consent or forbidden country transfers) were correctly identified and blocked, while compliant events were allowed. There were no false allowances or false blocks observed under the tested rules, indicating that the rule-based AI logic reliably enforced the policy definitions.

This high accuracy is expected given the deterministic compliance engine; it validates that the encoded rules align with the test scenarios and that the system can automate decisions consistently.

Latency

The end-to-end latency for processing an event and making a compliance decision was low, on the order of tens of milliseconds in the Kafka pipeline. Kafka's real-time streaming ensured minimal delay in delivering events to the AI consumer. The additional latency to record decisions on the blockchain was higher (often on the order of 1–2 seconds per transaction) due to Fabric's consensus and block confirmation process.

In practice, this creates a slight lag for ledger finalisation but does not impede the streaming flow since decision-making occurred near-instantaneously before ledger logging. The compliance decision latency (from event publication to AI verdict) remained negligible, demonstrating that real-time enforcement is feasible. The ledger commit latency is a trade-off for achieving immutability, but remains within acceptable bounds for post-fact audit logging.

Throughput and Scalability

Kafka demonstrated the ability to handle a high volume of events, easily sustaining hundreds of messages per second in the test environment without message loss or backpressure. The system's throughput was primarily constrained by the blockchain write rate. If every event is immediately written to Fabric, the throughput effectively falls to the blockchain's transaction rate (which in the current single-orderer setup was a few transactions per second).

However, because Kafka decouples ingestion from processing, the system could buffer and batch compliance decisions if needed, or scale out with multiple consumers and parallel blockchain channels. Resource utilisation (CPU, memory) on Kafka brokers and consumers stayed moderate even under load, indicating the design can scale with additional hardware.

The evaluation suggests that for realistic media workflows (typically bursts of events rather than constant high TPS), the architecture would handle demand, but optimisations (such as asynchronous ledger writes or Fabric throughput tuning) would be needed for truly large-scale deployments.

Transparency and Auditability

A key strength of the prototype is the end-to-end traceability of decisions. Every compliance decision (approval or rejection) is logged for audit. Approved events were recorded on the Hyperledger Fabric ledger via the `recordDecision` smart contract, creating an immutable, timestamped record.

For blocked events, the system logged the incident to the Kafka `ai-decisions` topic and console, ensuring no action goes unnoticed. Using Fabric's query functions, auditors can retrieve a full history of compliance events (by event ID or in bulk) and verify that no tampering has occurred.

The logs captured all relevant metadata (event ID, timestamps, decision result and reason codes where applicable), which aids intelligibility. In our tests, querying the ledger for sample events confirmed that the data matched the live stream outcomes, demonstrating consistency between the real-time decisions and the permanent records.

The transparent logging fulfills compliance reporting requirements and allows stakeholders to understand how and when decisions were made.

Security and Integrity

The evaluation also considered the system's resilience to security threats and failures. As a permissioned platform, Hyperledger Fabric restricts ledger access to authenticated network members, preventing unauthorised entities from reading or writing compliance records.

In the prototype, all blockchain transactions were executed by a known client identity and attempts to use an unregistered identity (during the gateway integration trial) were rightly rejected. This confirms that Fabric's Membership Service Provider is effective in enforcing access control.

Data integrity is guaranteed by the cryptographic mechanisms of Fabric — any attempt to alter logged decisions would be evident to all peers. We simulated node failures to assess fault tolerance: if a Kafka broker or consumer were taken offline, the system recovered gracefully on restart, with Kafka's persistent log ensuring no messages were lost during downtime.

Similarly, the Fabric ledger, with its ordering and endorsement process, ensured that partial transactions would not be committed, preserving consistency even under error conditions. While the prototype was not tested against malicious attacks, its use of established security frameworks (Kafka ACLs and Fabric's chain-of-trust) provides a strong baseline.

No data breaches or integrity violations occurred in the controlled tests. Overall, the security evaluation highlights that combining Kafka and Fabric can create a secure-by-design pipeline: Kafka secures data in transit (optionally via TLS and access controls) and Fabric secures data at rest in the ledger, with cryptographic audit trails for every compliance decision.

Outcomes

Across these criteria, the prototype met its primary goal of automating compliance without significant performance degradation. It successfully blocked unlawful data transfers and logged events with full traceability. The modular event-driven approach proved effective in maintaining low latency and high throughput for the streaming component, while the blockchain integration added robust audit guarantees at the cost of some throughput and latency overhead.

The simulation outcomes demonstrate the viability of AI-driven, stream-oriented compliance enforcement. Notably, the system's accuracy and transparency were high, instilling confidence that such a mechanism can uphold regulatory standards.

Limitations were also observed: for instance, the incomplete Kafka–Fabric integration meant some ledger actions were triggered manually and the evaluation was conducted in a lab setting with synthetic data. These constraints temper the results, indicating that further work (e.g. optimising integration and testing at scale) would be required before real-world deployment.

Nonetheless, the evaluation provides evidence that the concept of an AI-

augmented compliance layer for cross-border data sharing is both technically feasible and effective within the tested parameters.

Chapter 6

Conclusion and Future Work

Conclusion

This dissertation set out to investigate whether AI-driven analysis, combined with distributed streaming and blockchain, could enhance compliance in cross-border media data sharing. The research has successfully demonstrated a working prototype that addresses the core research questions.

First, showing that AI techniques (even in the form of a rule-based engine) can be used to interpret and automate the enforcement of international data privacy regulations in real time, blocking or permitting media transfers based on legal policies (addressing RQ1).

Second, by integrating Hyperledger Fabric, exploring the benefits of using blockchain smart contracts for automated legal compliance, finding that they provide tamper-proof record-keeping and distributed trust, albeit with some performance overhead and legal caveats (addressing RQ2).

Third, Apache Kafka was validated as an effective backbone for low-latency, secure data transmission across borders; it enabled the system to scale and remain responsive under load, confirming Kafka's suitability in such regulated, distributed environments (addressing RQ3).

The key contributions of this work include the design and implementation of a novel architecture that merges AI, event streaming and blockchain for compliance purposes. To the best of the author's knowledge, this is one of the first practical demonstrations of these technologies working in concert to enforce data protection rules in real time.

The project contributes to academic knowledge by providing a case study in the Design Science tradition: it identifies a pressing problem, builds an artefact to solve it and evaluates that artefact against clear criteria. In doing so, it fills a gap in existing research, which often considered AI, Kafka and blockchain in isolation. The findings show that an integrated approach is not only feasible but offers clear advantages in ensuring compliance and transparency in data sharing.

For industry practitioners, this work has practical relevance as a blueprint for next-generation compliance systems – the prototype could be adapted to various domains (beyond media, such as finance or healthcare) where data moves across

jurisdictions and must obey complex regulations. By logging every decision on a shared ledger, organisations can foster accountability and trust, reducing the need for manual audits. Moreover, automating compliance checks can significantly reduce the risk of human error and the cost of regulatory oversight.

However, it is important to recognise the limitations and the context of these results. The prototype was developed and tested in a controlled environment with simulated data. While it confirmed the concept, real-world deployment would require handling greater complexity. Legal rules were simplified and the "AI" component was not a machine learning model but a deterministic engine – a pragmatic choice for prototyping, but one that might not scale to the full breadth of evolving laws.

Integration issues (like the incomplete Kafka-to-Fabric gateway) highlight that engineering challenges remain in achieving a seamless end-to-end system. Additionally, questions of legal acceptance loom large: for instance, whether regulators and courts would accept blockchain records as evidence of compliance, or how to accommodate legal rights (like data deletion requests) in an immutable ledger.

Despite these challenges, the core finding is optimistic: technological tools can be orchestrated to greatly improve compliance assurance in data sharing, offering faster responses and better oversight than traditional methods.

Future Work

Building on this foundation, several enhancements and investigations are recommended to fully realise the vision of AI-driven, compliance-aware data sharing:

Richer AI Compliance Engine

Future iterations should incorporate advanced Natural Language Processing models (e.g. fine-tuned legal BERT/GPT models) to interpret regulatory text and handle a wider array of rules. This could enable the system to automatically update or extend its rule set as laws change, improving generalisability beyond the hard-coded logic.

Additionally, techniques for explaining AI decisions (such as traceable rule-decision mappings or model interpretability tools) should be integrated to maintain transparency as the AI grows more complex.

Complete Hyperledger Fabric Integration

The next development priority is to fully integrate the Kafka stream with the Fabric ledger via the Gateway API. Overcoming the identity enrollment issues and enabling the automatic writing of compliance events to the blockchain would allow truly real-time end-to-end operation (no manual steps).

A successful integration would involve robust error-handling, perhaps queuing transactions when the ledger is temporarily unreachable and replaying them, to

ensure no decision is lost. It would also be useful to explore Fabric features like private data collections or off-chain storage for personal data, to better align with privacy requirements while still leveraging the immutable log for compliance proofs.

Scalability and Performance Tuning

To prepare the system for real-world use, extensive benchmarking and optimisation should be conducted. This includes testing with larger Kafka clusters, multiple Fabric peers and orders and possibly employing Fabric's support for parallel channels or sharding of data by category.

Optimisations such as batching multiple compliance decisions into a single blockchain transaction, or using a lighter consensus algorithm for Fabric (e.g. RAFT tuning), could dramatically increase throughput. Employing asynchronous processing – where the AI's decision immediately triggers the next event handling and ledger logging happens in parallel – would help decouple latency-sensitive data flow from the slower audit trail commits.

Legal and Regulatory Validation

It is crucial to collaborate with legal experts and possibly run the system in a sandbox regulatory environment. Future work should involve validating the rules and outcomes against real legal cases or compliance checklists to ensure the system's decisions truly align with law (and catch any misinterpretation early).

Engaging with regulators or standards bodies could pave the way for the system's outputs (such as blockchain logs) to be recognised as evidence of due diligence. Additionally, addressing legal concerns like GDPR's Article 17 (data erasure) and Article 22 (automated decision-making limitations) will be important – for example, by developing a policy for selective redaction or by integrating a mechanism for human override/review of AI decisions in borderline cases.

Enhanced Security and Privacy Features

While the current design leverages Kafka and Fabric security, future enhancements could include end-to-end encryption of the data payloads, more granular access controls and integration with identity management systems (for example, tying user consent to a digital identity on Fabric).

Furthermore, implementing monitoring and alerting for suspicious activities (like unusual patterns of blocked events that might indicate a misconfiguration or an attack) would turn the prototype into a more comprehensive compliance monitoring solution.

User Interface and User Experience

To drive adoption, a future version should provide intuitive interfaces for stakeholders. This could include dashboards for compliance officers to see real-time

statuses of data transfers, detailed audit logs with search and filter capabilities and interfaces for administrators to update rules or AI model settings without deep technical intervention.

A user-friendly system would allow non-technical legal staff to trust and interact with the automation, bridging the gap between the technical backend and the professionals responsible for compliance.

In conclusion, AI-driven compliance and secure data sharing is a fertile ground for further research and development. This dissertation has laid the groundwork by proving that a blend of AI, Kafka and blockchain can indeed address the core challenges of cross-border media data collaboration. The next steps will involve scaling this approach, refining its intelligence and usability and rigorously validating it in real regulatory contexts.

By continuing to iterate on this fusion of technologies, moving closer to a future where organisations can collaborate across borders with confidence, knowing that compliance is embedded in the very fabric of their data infrastructure.

6.0.1 Github Repo:

`https://github.com/rohansikder/MSc_FYP`

Bibliography

[1] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2008.

[2] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Legal-bert: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP*, 2020, pp. 2898–2904. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.261/

[3] D. A. Zetzsche, R. P. Buckley, and D. W. Arner, "Artificial intelligence in compliance risk management," SSRN, Tech. Rep., 2020. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3531711#paper-citations-widget

[4] S. Wachter, B. Mittelstadt, and L. Floridi, "Why a right to explanation of automated decision-making does not exist in the general data protection regulation," *International Data Privacy Law*, vol. 7, no. 2, pp. 76–99, 2017. [Online]. Available: https://doi.org/10.1093/idpl/ipx005

[5] M. Finck, "Blockchain and the general data protection regulation," Max Planck Institute for Innovation and Competition Research, Tech. Rep., 2018. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3080322

[6] M. Finck and D. Moscon, "Smart contracts and the gdpr: Friends or foes?" *Law and Contemporary Problems*, vol. 82, no. 2, pp. 37–58, 2019. [Online]. Available: https://scholarship.law.duke.edu/lcp/vol82/iss2/3

[7] S. Weber, Y. Xu *et al.*, "Blockchain and gdpr: A systematic literature review," *arXiv preprint arXiv:2104.00648*, 2021. [Online]. Available: https://arxiv.org/pdf/2104.00648

[8] K. Waehner, "Apache kafka and blockchain / dlt comparison — kafka-native vs hyperledger, ethereum, ripple, iota, libra," https://www.kai-waehner.de/blog/2020/07/17/apache-kafka-blockchain-dlt-comparison-kafka-native-vs-hyperledger-ethereum-ripple-iota-lib
2020.

[9] Digitalis.IO, "Apache kafka and regulatory compliance," https://digitalis.io/blog/kafka/apache-kafka-and-regulatory-compliance, 2022.

[10] IBM, "Introducing openpages 9.1: Unlocking new ai-infused capabilities for risk and compliance management," Web announcement, 2025, includes automated regulatory mapping, obligation analysis, DPIA workflows. [Online]. Available: https://www.ibm.com/new/announcements/introducing-openpages-91-unlocking-new-ai-infused-capabilities-for-risk-and-compliance-man

[11] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004. [Online]. Available: https://doi.org/10.2307/25148625

[12] N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, and V. Lampos, "Predicting judicial decisions of the european court of human rights: a natural language processing perspective," *PeerJ Computer Science*, vol. 2, p. e93, 2016. [Online]. Available: https://doi.org/10.7717/peerj-cs.93

[13] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, 2018. [Online]. Available: https://doi.org/10.1145/3190508.3190538

[14] M. Lippi, P. Pałka, G. Contissa, F. Lagioia, H. Micklitz, G. Sartor, and P. Torroni, "Claudette: an automated detector of potentially unfair clauses in online terms of service," *Artificial Intelligence and Law*, vol. 27, no. 1, pp. 117–139, 2019. [Online]. Available: https://doi.org/10.1007/s10506-019-09243-2

[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Bitcoin.org, White Paper, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[16] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, pp. 6–10, 2016. [Online]. Available: https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf

[17] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, 2014. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[18] J. Sai *et al.*, "Permissioned blockchain identity and access management system using hyperledger fabric," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/392996596

[19] J. Adlam *et al.*, "A permissioned blockchain approach to electronic health record audit logs," *ResearchGate*, 2020. [Online]. Available: https://www.researchgate.net/publication/346735005

[20] H. Liu *et al.*, "Achealthchain: A blockchain-based framework for privacy-preserving health data sharing," *Scientific Reports*, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-00757-1

[21] M. Islam, R. Hasan *et al.*, "Blockchain-based smart contract framework for dynamic consent in healthcare," *arXiv preprint arXiv:2402.06704*, 2024. [Online]. Available: https://arxiv.org/pdf/2402.06704

[22] B. John, "Data privacy and security risks in smart contracts: Compliance challenges for corporate governance," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/389881015

[23] Wikipedia contributors, "Apache kafka," https://en.wikipedia.org/wiki/Apache_Kafka, 2024, accessed July 2025.

[24] W. Adekunle and I. Odofin, "Designing event-driven architecture for financial systems using kafka, camunda bpm, and process engines," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 11, no. 2, pp. 178–184, 2025. [Online]. Available: https://www.researchgate.net/publication/392081813

[25] Apache Kafka Documentation Team, "Security overview — apache kafka 0.9 documentation," https://kafka.apache.org/090/documentation.html#security, 2016.

[26] D. Reijsbergen *et al.*, "Piechain: A blockchain auction protocol with kafka integration," https://arxiv.org/abs/1807.11378, 2023.

[27] F. Filacouris, "Blockchain data integration with apache kafka," https://developer.confluent.io/learn-more/podcasts/blockchain-data-integration-with-apache-kafka, 2023.