Deep Science Publishing
https://doi.org/10.70593/978-93-49910-78-2_11

Chapter 11: Building data engineering pipelines for payment processing and risk analysis

11.1. Introduction

As long as financial transactions are recorded and exchanged, it has always been unavoidable that irresponsible actors, or risk factors, engage in various acts of fraud to gain wealth unethically. Such unauthorized transactions include stealing of payment method information, creating unauthorized accounts for receiving gain from illegal activities such as money laundering, creating unauthorized merchant accounts to trick customers into fraudulent transactions, and phishing to swindle customers. Payment processing companies should detect such behaviors to protect legitimate users and save their companies from losing a considerable amount of money (Ghemawat et al., 2003; Kimball & Ross, 2013; Akidau et al., 2015). Growing up in an internet-enriched environment, millennials and Generation Z are the largest active user segment. They are highly sensitive to payment transaction efficiencies and prone to try new payment methods compared to Generation X and Baby Boomers. These new trends require payment service providers to design novel transaction processing systems to accommodate different payment methods, account users, and merchants, while keeping real-time fraud detection at tolerable costs. Acquiring, storing, and processing the intensive data streams generated by users during payment transactions are core challenges for payment processing services. Streaming data engineering is the cornerstone for building such payment processing and risk analysis pipelines. Data such as transactions, partner systems, and payment gateways are collected to a data lake from internal and external sources. Thereafter, the collected data are processed using batch or streaming pipelines to provide real-time transactional and risk insights for the payment operations organizations and partners. Those insights are then ingested into dashboards for tracking transaction activity and risk detection performance (Sadoghi & Jacobsen, 2011; Zhang & Xu, 2020).

11.2. Understanding Payment Processing

Processing consumer transactions is a critical infrastructure operation executed by payment systems that directly impacts most, if not all, enterprises in the world and is a required functionality in most enterprise applications. Businesses must provide consumers with a secure mechanism to pay for products or services purchased from those enterprises.

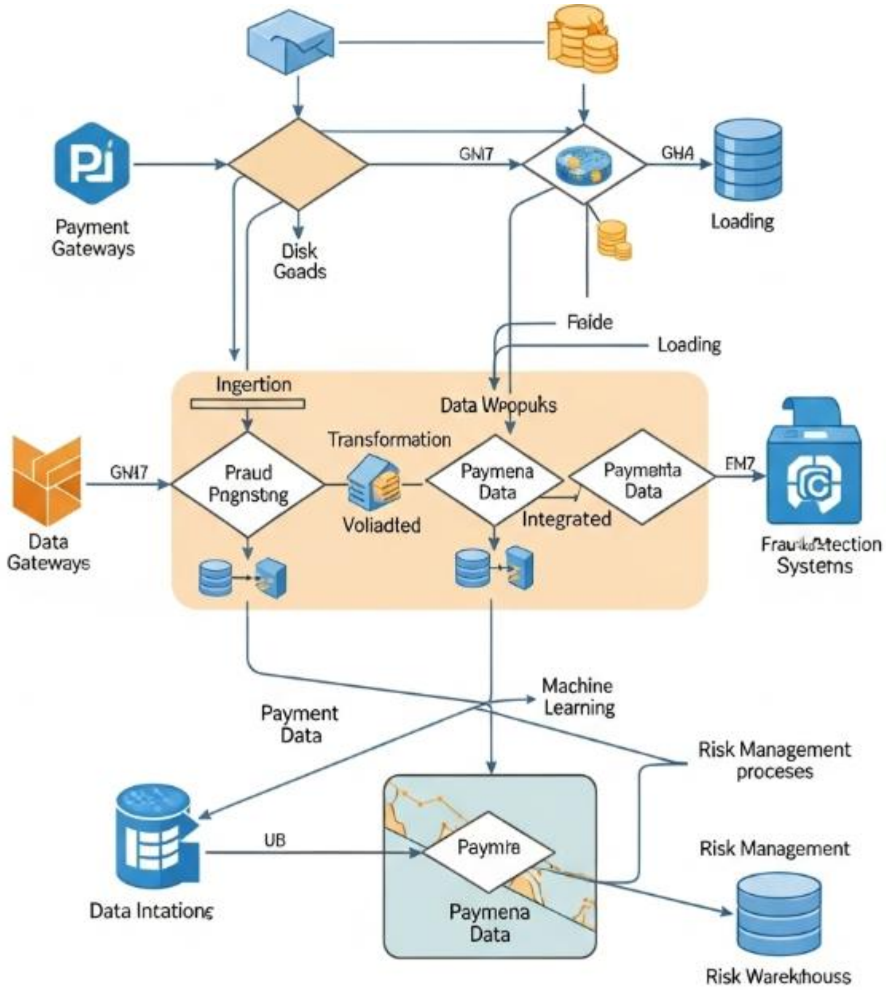

Fig 11.1: Data Pipeline Architecture

Merchants must accept forms of payments that consumers are comfortable with. Payment processing is the operation that actually moves a consumer’s money from their bank accounts to the merchant’s bank account. While payment processing appears to be a straightforward operation, underlying restrictions, intricacies, and operating protocols must be adhered to by the parties involved in the transaction. This is no different from businesses involved in contract negotiations and trades. This chapter explores payments systems and payment processing. We also cover the supporting infrastructure that enables payment processing at the core and examines why risk analysis algorithms must accompany payment processing service implementations.

Payments system is a shared infrastructure that facilitates the transmission of monetary value from one party to another. Payment processing refers to the act of moving monetary value from a consumer to the bank associated with the merchant in a transaction for goods or services purchased from the merchant. Interbanking setups such as systems have been in existence for many decades and enable deposits and withdrawals between bank accounts associated with deposits at other banks. Transaction types typically processed through these systems are direct deposits associated with payroll processing, vendor payments setups, consumer-to-consumer payments, and bank transfers.

11.2.1. Overview of Payment Systems

Payments are essential to how society and the economy function. They are the glue that holds the economy and society together, part of the fabric of commerce and social contracts. In the modern global economy, payments need to be efficient, safe, and secure. Naturally, we want cheaper, faster, easier, and better payments. We want consumer confidence that individuals as well as businesses will get what they are owed when they are owed it. Payment systems need to be available around the clock, with predictable and reasonable costs that are clear ahead of time. Nobody likes hidden fees.

Critically, payments are a substantial revenue source for certain businesses such as banks and credit card processing companies, who often hold on to funds for several business days or longer. Payment-processing technology trends are towards pushing costs down and increasing the speed of payments. Instant payments are a now-a-reality with various instant credit transfer systems. Open banking standards that force banks to make account balances and transaction data available to any bank-accredited third party equally created instant payment competition. Blockchain-based solutions, especially cryptocurrencies, hold the promise of doing instant payments for free, directly person-to-person, and on a global scale. These technologies often get pointed at large banks and global credit card networks for price-gouging consumers. Yet, digital currencies can sometimes act like a digital gold in being slow, for people seeking a safe place to store wealth.

In the meantime, electronic payment systems and processing have become essential infrastructure for the world’s economy. The world of payment processing certainly lives up to the hype of massive size, with more than $5 trillion in credit and debit card volume processed by merchant acquirers.

11.2.2. Key Players in Payment Processing

Payment processing allows merchants to accept payments directly from their customers. This payment processing service can be provided by banks or financial technology companies, with a model called Payment Facilitator. In this model, payments from customers pass through the payment facilitator company acting as a merchant service provider and aggregating the customer's transaction information, allowing merchants not to enter into bank contracts to accept transactions using different means, such as credit cards. Payment facilitation has reduced the burden of entering agreements to be able to accept payments, favoring the expansion of online business.

Payment facilitators also handle the technical difficulties involved in acting as a merchant service for digital merchants. The transactions for digital merchants are not always direct sales with the end customer. The digital merchants may have collaborations with other merchants acting as suppliers of the items or services sold, commonly known as drop shipping, in which the facilitator acts as an intermediary for the transfer of money between the end user's credit card and the supplier providing the products or services. The transfer of information about the transaction between the merchant and the digital merchant is needed in this case. The payment facilitator handles the different means used by customers to make transactions, providing integrated payment solution services for merchants, through which they can accept e-wallets, debit and credit card payments, and bank transfers.

To facilitate and act as a merchant services provider in the payment transaction, payment facilitators need to interact with banks, card schemes, gateway processing companies, and other parties. The overall transaction flow passes through these companies to process and approve the transactions, which usually take only a few seconds.

11.2.3. Transaction Flow in Payment Processing

Discussions surrounding transaction flow in payment processing are often hindered by impracticality and ambiguity; impracticality because we dilute the usefulness of the supply chain by combining payment processing with other parts of society, such as consumer goods commerce. In this manner, for nearly every merchant, from a sole pizza vendor to an aircraft manufacturer, there exists a different payment transaction flow. The second issue is that payment processing is sandwiched between multiple other agents who have conflicting perspectives - the agent handling the payment often doesn't know how they interact with the merchant, and hence their data cannot be trusted to reflect that reality correctly. Card networks and blockchains also have regulated fees associated with them, which incentivizes merchants to interact with them in a specific fashion distorting the nature of a payment processing transaction.

Notwithstanding these two difficulties we can interpret payment processing transactions as interactions between a web of agents mediated by Reserve Currency accounts. This interpretation works for all card network transaction flows, bank transfer systems, real time payment systems and crypto-based systems. The only system that doesn’t fit this pattern, are card networks during market opening times, especially, who settle transactions in traditional payment currency banks during non-business hours by credit to a merchant’s account, debit to the consumer’s accounts. However, more and more, card networks are mimicking crypto behavior by issuing reserve currency wallets to merchants and consumers and authorizing transactions directly on-chain. They have also rolled out settlement processing on-chain.

11.3. Risk Analysis in Payment Processing

The fraud detection and prevention framework used by different parts of the payment processing ecosystem defines the economy's impression of risk. The risk perception can lead to adoption inertia: alternative, higher-risk systems may be unacceptable to customers who know the potential consequences when a risk does get acted upon. Alternatively, if a system is deemed fully secure against all known risks, growth will expand rapidly, creating joint action and coordination problems. The goal of risk analysis is to balance risk and optimal payment utility. To avoid these extremes is a delicate business.

Payments systems face many types of risk, including fraud, settlement, settlement delay, liquidity, credit, operational, legal, damage and volatility. The goals of payment systems risk management is to remove, mitigate and/or allocate as many payment risk elements as possible. Credit risk is often a large differentiator in the payment systems business model. Fraud risk, intrinsic to digital payments, is the difference between the system's initial design and projected actual state after taking mitigating actions in an asset liquidation window that is often dependent on the type of processing party who is victimized. Party victimization during certain periods by certain payment methods is also known to the industry. For these parties, fraud risk is the applicable legal doctrine and the system's projected patching speed compared with the industry liquidity insurance costs for the period.

11.3.1. Types of Risks in Payment Systems

Risk analysis in payment processing is aimed at reducing those events that would impair the integrity, reliability, and availability of payment systems. Risks may arise from the inadequate or failed internal processes, people and systems, or from external events. Considering the types of events that cause harm, there are two types of risks in payment systems. The first type is process risk that is caused by internal factors. The second type is enterprise risk that is caused by external factors. The basic building blocks for both types of risks are discussed below, with enterprise risk being a broader aggregation of process risk.

Process Risks Payment processing activities include authorization, clearing, payment, currency settlement or liquidity management, or settlement finality or reconciliation. There are four major types of process risks associated with these functions – operational risk, applicability or design risk, implementation risk, and latent risk. Operational risk is the risk associated with the action of employees. An excessive, inherent, irreversible, unjustified risk associated with the performance of action of employees in one or another payment processing category causes losses. For example, security risks during the process of verification of identity of a customer, verification of the availability of funds, analysis of information technology risk, etc. In fact, the concept of risk appetite is originated from operational risk, describing the circumstances in which outsized operational losses should be viewed as acceptable. The action of the employees in payment processing activities is concentrated in two areas – approval and risk assessment at the customer level, approval of identity transactions and risk assessment at the identity transaction level.

In many instances, actions at both levels are automated. However, at the customer level, it is also critical to decide the level of risk associated with different types of customers, types of expected payment transactions and types of expected payment sources and routes, and the degree of automation. The general principle is that the higher the risk associated with a specific employee action within a payment processing activity, the more stringent should be the rules assessing risk and requiring verification.

11.3.2. Risk Assessment Frameworks

Fraud risk assessment is typically performed on a qualitative basis or a combination of qualitative and quantitative basis using methods such as heuristic assessments, the use of scoring models or predictive models, and a combination of all three. The assessment is either a stand-alone periodic review or a component of other assessments conducted for a financial institution or a payment processor. A risk framework outlines the financial institution’s business strategy, products and services, financial condition, and the complexity of its operations which when combined with the operational internal control environment typically forms the broad basis for the qualitative component of a fraud risk assessment. The risk framework may also target areas or indicators that tie business characteristics to fraud risk. These indicators may include geographical location of customers, industry type and characteristics, KYC process, business model expenditures, volume and velocity, abnormal changes over time, transaction attributes, historical exception reporting, customer complaints, statistical analyses, and transaction endpoints.

The qualitative assessment is usually conducted at the payment system, payment type, and payment channel level with the output of the qualitative assessment being a risk ranking of the fraud risk. The quantification of the qualitative assessment serves to validate that the risk ranking aligns with the actual fraud loss experience. The quantitative component can also provide an early warning indicator. Empirical estimations of fraud risk can be obtained through a number of statistical and econometric techniques using data from the internal and external sources. The quantitative methods use historical loss data to predict fraud risk via either contagion, descriptive, limited dependent variable, or time series models. Data mining, extreme value theory, loss distribution convolution, or predictive analytics can be applied to transaction data to estimate the fraud distributions or fraud indicators.

11.3.3. Impact of Fraud on Payment Systems

Frad stands as one of the most serious problems that payment systems deal with today. The networks that transfer funds during the payment process take the presence of fraud on high volumes and large values of transactions with low margins that are characteristics of the payment domain. In this domain, it is often argued that the existence of fraud in high levels reduces the value of all participants in a payment system, and it is up to the responsibility of payment system stakeholders to devise ways to contain it and ultimately reduce it to levels considered manageable.

Payment systems need to question the purpose of high value for low margin payments that are often done by citizens or business entities that cannot afford losing even a small part of their wealth. If not deterred by a risk analysis framework, the payment systems will easily become an environment fertile for misconduct and illicit actions. The outcome will be unpredictability and instability and this will be felt by the economy. Payment systems are enablers of other economic systems: those of goods, services, securities and currencies. Uncertainty caused by the inability of reducing fraud volumes and rates on the payment system will spill over to those inbound systems and will be costly in terms of monetary displacements, not to mention the intangible costs associated with economic growth mismatches that are possible predictions or even consequences of channeling large volumes of payments with high risk of fraud through the payment system.

11.4. Data Engineering Fundamentals

Data engineering is the 'engineering' side of data science. It is what makes the more experimental phase of data science actualizable and scalable for the real world in order to deliver tangible products and services. Data engineering typically consists of the creation, management, and orchestration of 'data pipelines.' Data pipelines, in turn, are a special case of a generic software pipeline, which is a specific orchestration of discrete data processing units (called 'steps' or 'tasks'), which may run sequentially or concurrently, and which are designed to transform data from one state to another. As such, each data transformation pipeline is built up through the careful selection and organization of data task units, defining the specific form of 'data flow' in 'data processing' mode.

Data pipelines can be classified based on their function and based on their implementation. Based on their function, data pipelines can be classified as: (1) Data ingestion pipelines, which transfer data from source to target systems; (2) Data transformation pipelines, which perform transformation operations on data; (3) Data preparation pipelines, which are responsible for supplying 'data ready for use' to analytic or other applications; (4) Data quality pipelines, which check and certify that the data are indeed of 'good quality'; (5) Data orchestration pipelines, which coordinate the execution of the aforementioned specialized pipelines and therefore have a supervisory role in the entire data pipeline environment; and (6) Data monitoring pipelines, which are responsible for continuously checking the status of production data pipelines and the quality of their output data.

11.4.1. Introduction to Data Engineering

Data engineering enables organizations to make data-informed decisions that are often based on machine learning algorithms that directly benefit the organization. Data engineering refers to the processes involving the design and development of systems that allow the collection, storage, and analysis of data that generates actionable insights and advice, often through interactive dashboards, constrained dimensional queries, or advanced visualization techniques. Such support for data-informed, real-time decisions often require the data systems to be fast, responsive, resilient and available. The guidance often involves running what-if scenarios, simulations, and other adjacent data activities that may be done outside the critical day-to-day operations. In fact, the tools that involve using the core data systems mainly from a secondary operational perspective are known as data management tools. Such tools provide governance, control, and aid in the ingest processes to allow data scientists to build their machine learning models on reliable and clean data.

While there may be some overlap between data science and data engineering, they are distinct disciplines. While data scientists will often design and build one-off systems for exploratory analysis or specific predictions, data engineers will often design systems that work at scale, and that have high reliability and long lifetimes. While data scientists should understand and appreciate the tools and methods used by data engineers, most of their work will be with the data that is produced and made available by data engineers. There may be some cases where data scientists will sink to the level of pipelines and middleware and get their hands dirty with the engineering side of the problem when their tasks are time-critical.

11.4.2. Data Pipelines: Concepts and Components

Data Pipelines: Concepts and Components Although the term data pipeline is often used generically, without clear distinctions, there is a clear and definitive definition of what data pipelines are. Data pipelines are a means of obtaining, storing, and processing dynamic or streaming data. They are used to automate the flow of data from sources to destinations. Data pipelines ingest data from sources, prepare it for analysis by validating, cleansing, and converting data formats, and publish the data to central storage repositories or other endpoints for further processing, analysis, or visualization. Out-of-the-box data pipelines are easy to deploy and configure, and require minimal management and maintenance. They are reliable, secure, and do not require programming or technical expertise to design and deploy.

These capabilities distinguish data pipelines from simple data ingestion and movement tools like traditional data extract, transfer, load tools and commercial enterprise service bus solutions. ETL tools and ESB solutions can only extract data from source systems, validate it for errors, and send data to single destination systems. Unlike data pipelines, ETL tools and ESB solutions do not provide data cleansing, transformation, and publishing capabilities. Because ETL tools and ESB solutions can only work with "batch" or static data obtained from relational databases, they do not support data "publishing." This is the key difference between traditional ETL tools and data pipelines that can transport, process, and distribute streaming data in more varied formats.

11.4.3. ETL vs. ELT: Understanding the Differences

ELT and ETL are two very similar strategies for performing data ingestion, guaranteeing data availability, and making it pipelined for further data analysis and experiments. In many situations, data ingestion and storage are the only things done with the data. Only for more advanced or demanding tasks is processing performed using one of the data processing engines. Therefore, for simplicity, speed, and productivity, in many instances, we can perform the ingestion, storage, and data availability using one strategy, which can be many times a simpler variation of any of the higher-level data pipeline strategies. Specifically, ETL is the traditional and state-of-the-art strategy for ingesting, transforming, and loading data for data analysis, while ELT is a new rival to ETL and newer data processing pipelines, involving a decision to first ingest and copy the data to the storage, and only later on demand or periodically into availability or visualization. At a high level, the fundamental difference between ETL and ELT is that the latter is a simpler strategy, where the loading phase is made the most important step of the entire end-to-end data pipeline, performed in the background and constantly updating the data for further processing. ETL focuses on transforming data before it enters data storage, and can achieve better preprocessing power, but in so doing, it makes only one loading step of the data pipelines. In contrast, ELT lets simple raw data copies circulate, but it has the flexibility of many periodic data availability steps and allows better processing and availability of data from multiple sources.

11.5. Designing Data Pipelines for Payment Processing

Every financial transaction generates observable transient metadata, such as timestamps, sending and receiving institutions, or addresses. Analyzing this transaction metadata can uncover rich information about the nature of the entities involved, how they conduct business, in what volumes, and how funds flow between them. Payment networks connect a large number of users and institutions and act as intermediaries in execution and settlement of transactions, typically exchanging value in fiat currencies. They keep specific data collected from users and institutions connected to their network to process and clear instant settlements, and from all market participants for at least as long as required by local laws, regulations, or policies. Thus, these networks create, store, and transit rich transaction metadata and are potentially susceptible to external risks or involved in crisis situations.

Payment processing metadata facilitates the actual clearing of financial transactions. Missing payment processing metadata can severely delay transaction settlements, which are now conducted instantaneously on payment networks. Payments with visible metadata enable empirical research to determine the relationships between payment institutions interacting with users. The microscopic and partially visible universe of payment flows provides the financial context – who pays what to whom? – for investigating how interactions between payment flows and other data sources reciprocally affect each other's dynamics and whether some of them may become sudden crisis triggers. Modern payment networks must adhere to each jurisdiction's rules and standards, as well as Data Protection regulations in place. Payment metadata disclosures thus generate reputational and legal risks for these payment processing companies because they would be helping regulators – and therefore, the government – access user data.

11.5.1. Defining Requirements for Payment Data Pipelines

The requirements for payment pipelines can differ significantly from those for other types of data pipelines. In this case, support for payment workflows is paramount. Payment processing needs to be executed optimally at every stage of the transaction lifecycle. Furthermore, payment events should take priority and be handled as fast as possible. However, creating solutions for real-time payments presents their own set of challenges. Payment transactions are different from other kinds of events because they are often irreversible. Payment transactions introduce multiple technical and business complexities due to an absence of transactional guarantees; they may not conform to properties. Merchants therefore need access to their transaction data as fast as possible. Making this data available to various merchant solutions and ensuring it is accurate is one of the primary goals of payment systems. Consequently, requirements for payment pipelines are specific and may diverge from other data pipelines.

Payment systems usually need to capture payment events with zero loss, with the highest degree of accuracy, and with minimal impact on payment processing customers. Any transaction that is not captured or that contains errors could have dire business implications since payments are directly tied to the financial health of the merchant conducting them. Additionally, understanding transaction data is one of the most challenging but important responsibilities for payment teams. Payment data must be as detailed and specific as possible to be of benefit to people and applications interpreting the payment data. Depending on the payment channels utilized, transaction data could be extensive and time-consuming to replicate across payment systems and pipelines, reducing its utility within analytical systems.

11.5.2. Choosing the Right Technologies

In order to process large volumes of payment data quickly and reliably, payment data pipelines typically rely on powerful distributed systems. Cloud computing has made such systems and storage solutions easily accessible, allowing decision-makers to focus on how to design the data pipelines, rather than which technologies to use. As a consequence, the landscape of available technologies is exciting, and this ease of access has fueled explosive innovation. Short development cycles of cloud-based services are creating new options for payment pipelines and updating existing services with new features. However, the speed of such innovation can also be confusing and overwhelming. If you are involved in building a payment data pipeline, clear guidance on how to make technology choice is essential. In this section, we outline considerations for the choice of technology in building modern payment pipelines.

Since the earliest days of big data, there has been a dichotomy between batch and real-time processing. Initially, large-scale data processing was relegated to batch jobs that ran on a cadence dictated by end-user reporting and algorithm needs. Core batch-space technologies were architected for data-intensive jobs with large input footprints and large realizations. The introduction of more complex data systems enabled better accessibility for business users. However, data remained static between batch runs, resulting in stale data, which was typically unacceptable for any time-critical application. With the advent of stream processing technologies, the landscape of capabilities began to evolve rapidly, and the pivot to near-real-time enabled a broader spectrum of possible applications.

11.5.3. Data Ingestion Methods

Ingestion of physical payment data from different resources, services, and platforms is one of the most delicate parts of designing and building the data pipeline. Information about payments is usually divided into different payments systems based on payment methods such as debit or credit cards, wire transfers, etc. Payment data related to wire transfers is either moved to the database or stored directly into the database to which the client has provided access. Other types of payments data, like card payments, are usually transferred using APIs provided by the payment method provider, namely vendors like PayPal, Stripe, Authorize.net, or payment gateways as payment processors. Each of these payment vendors has provided their methods to process the data for payments properly. The GraphQL API by Stripe provides a strong frictionless approach to payment processing data ingestion but increases the complexity of tech stack due to different implementation languages being supported. Each of these payment method's payment processing APIs is not built the same way with all available features.

Another kind of ingestion infrastructure is built to bulk ingestion of data from data dump files which are being generated and maintained in external warehouses accessible via FTP or SFTP. For wire transfer transactions, there usually exists a set of files every day to process. These files are being generated from the banks after applying batch jobs to create a list of payments containing data about sender and recipient bank accounts with transfer-sent timestamps, references, amounts, etc. Like processing wire transfer transactions, bulk processing other types of payments is also tedious. Because of these challenges, planning voice and data pipelines for data ingestion from banks and payment transcription vendors becomes an overwhelming task, although the importance of building high-scale and low-latency systems is needed to maintain trust and relationships with payment transaction clients.

11.5.4. Data Storage Solutions

Choosing a data storage solution to store payment data depends on two factors, as follows: • What type of data do you want to store? Payment processing can generate several bulk and transactional data sets. Bulk datasets can be transactions, where each row in the data corresponds to a transaction, but transactional datasets can be for any level of granularity like user, account, and merchant level. These datasets can be in the form of raw files or in a database server. Other associated datasets can be merchants, products, order, invoice, and partnership where the same charge for the same purpose goes to different companies. Data associated with merchants and partners can often contain sensitive data. Also there are products which for example in an e-commerce setting can go from it being purchased to later returning it. • What type of data query patterns do you anticipate? For example, payment processing generates payment calls consisting of payment amount and merchant ID which will happen several hundred to thousands of times per hour for each merchant. At the same time it generates refunds of the charge where after some duration of the charge refunds may it be denied or accepted (a merchant can also file for a chargeback about a charge he/she/they refuses to accept). Reports and quality of service system can indirectly generate query patterns that may analyze/aggregate the charge data on weekly or monthly basis which would be heavily consolidated. The number of concurrent users may be large during some instances such as happy hours, daily work hours, weekends, holidays, festival or holiday sales, and concurrent users could all put a call to the same product as application servers can write but database servers may all attempt to read from the same dataset.

11.6. Building Pipelines for Risk Analysis

Risk analysis is a unique processing step still left out of common data engineering documentation. Even though it's considered a subset of feature engineering for risk models, it should still be treated as a specialized data engineering stage because it's so critical to the model development and deployment lifecycle and because of its unique data, independence, structure, and techniques. For risk, the data used is different than that used for scorecard development or backtesting, users may or may not be the same and support factors, such as rank and recency, are often not applied. Unfortunately, most discussions around transaction data engineering processes omit fraud and AML.

A full cycle risk analysis pipeline could take data from multiple product data marts, filter and augment relevant transactions with geo and behavioral data, merge with credit application data, apply historical modeling logic to classify relevant segments, and augment transaction details at scale. The final transformed data is intended to help linoleum a model to predict future penalties for the suspicious transactions, audit from the terminal/country/merchant level for countries with low/no penalties. The advantage of a high dimensional sparse dataset like this is that at the transaction level, generalization is not so an issue and it results in the auditor being able to gloss over tons of low risk transactions from tourists and missing merchants and terminals.

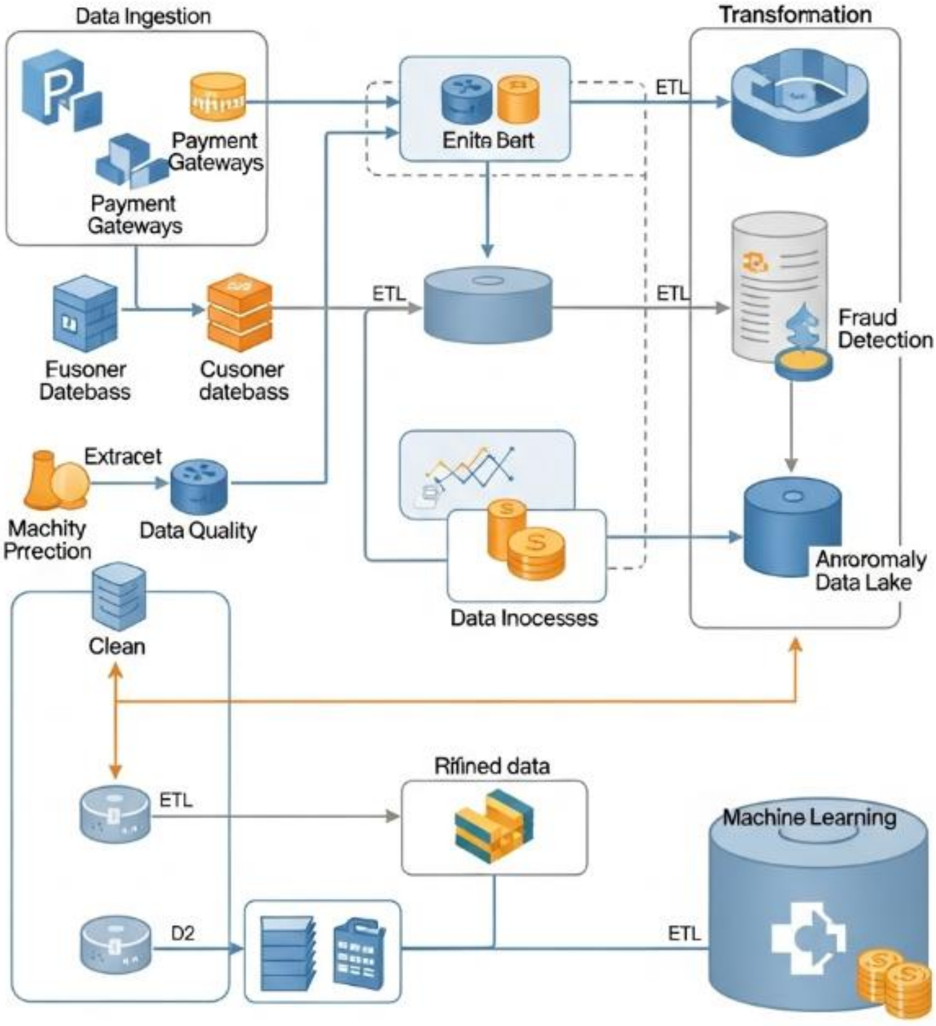

Fig 11.2: Data Engineering Pipeline

For AML and Fraud, the raw transaction/withdrawal data is neither 100% nor 0% fraud/AML/blacklist classified normally. The parameters and logic tied to risk segments and weights would be a business secret and change on a case-by-case basis. Periodic reruns of the data engineering segment would normally expose these changes while the different flows/models followed could be compressed to template files with the parameters stored elsewhere for speed of cycling.

11.6.1. Data Sources for Risk Analysis

Risk analysis is one of the most important items of business logic in every detection pipeline. To be able to calculate risks, risk analytics should have all the precise and historical information about user behaviors. Therefore, all the data sources that can be used for building detection pipelines will also be very useful for risk analytics purposes. With very low compute power costs, risk analysts can benefit from highly processed data at very low latency based on planned batch runs.

Another type of risk analytics is real-time anomaly detection. While similar to some types of detection pipelines, this quickly evolving and changing domain requires special attention to the data sources and specifics of risk analytics. Disparate data sources, such as the close connections of the user with external parties like clients and vendors, credit rating scores, behavioral info, income flow, and other business drivers quickly drift over time, and these models tend to get outdated after a while. While the detection models won’t be the same as for risk analytics, we can create their versions, relying often on the same data sources, creating efficient models, and using them in unison.

11.6.2. Data Transformation Techniques

Once we received the event data, in the pipeline for risk modeling, we usually execute some data transformations to make the information better suited for the models that we are going to train. Typical transformations include categorical encoding and time-window aggregations. For breach detection, we use time-window aggregations to measure the amount of requests over the last hour to the same target and incoming from the same user/foreign IP pair. For the detection of payments with manipulated information for verification, we apply categorical encoding to the issued country of the card and the foreign IP. Such encodings allow using gradient boosting trees for breach detection and classical neural network architectures for information manipulation detection. They also increase training efficiency. Note that the number of transactions to the same target is an important factor in anomaly detection in general. For example, in domain knowledge, it is stated that the number of image requests to be sold on a marketplace must contain redundancy within one hour. Any attempt to sell such an image must generate a large number of view requests to be considered a potential breach. Algorithms of supervised learning for anomaly detection usually judge a transaction to be a breach when it is the only one that had been issued for a target within the past hour or 90% of the transactions to that target in the past hour are breaches.

11.6.3. Real-time vs. Batch Processing

Real-time data processing and batch data processing has their respective advantages and disadvantages, and hence which type to use depend on the use case and system specifications. Real-time processing delivers instantaneous results. However, the latency to produce outputs in batch processing can be minutes, hours, or even months. The number of events that invoke pipeline invocation requests varies—one request or two requests per second for an online payment platform that generates huge amounts of payment data, and around a few requests per day for a cryptoexchange platform with different volume fluctuations throughout the day. The time gap between subsequent requests for certain payment processing services is so small that the service requesters for the payment processing service are charged overage fees if an API Gateway for these transactions is used. These considerations can make the choice lean toward processing the transactions in batch. However, for users to receive their refunds faster, real-time processing may be the only way to go with.

For attacking the transaction risk analysis pipeline, it is then important to keep the table of common attributes for both types of analyses updated as often as possible; otherwise the existing tables for risk analysis can get stale and outdated over time, and be conducive for fraudulent activities. To achieve that for the batch analysis table, the transactions involved in a previous day's game should ideally either be ingested into the batch analysis table or an updated version of the batch analysis table should be generated by triggering a batch ETL job for the game after it has ended. For the real-time analysis table, users may want to purchase their tickets for a day’s game after observing the ticket prices for the game designated by the user through subscriptions to ticket exchange platforms, for one with a good enough price drop. For such analyses in real-time, it is important to have the data in the pipelines in real-time to satisfy the requirements for alerting the users, emails, and/or mobile notifications.

11.7. Integration of Payment Processing and Risk Analysis

One of the greatest contributions of data engineering to business is the ability to integrate multiple data processing tasks to achieve synergy across functions. Often, pipelines are developed in different groups or by different individuals, with no focus on end-to-end enablement, limiting the value that can be derived. This is particularly true in the case of the integration of financial risk analysis into payment processing. The two activities are tied together by having the same data. The twin challenges are that the speed of payment processing is necessary due to the immediate business need to either accept a transaction or reject it, while risk detection is mainly a situation of batch processing done at varying intervals. An added complexity is that the outcome of the batch processing can be used to inform future payment processing. For example, if a customer is determined to be a high risk for fraud or chargebacks, the payment processing pipeline may choose to implement a more stringent transaction validation step for this customer in future transactions. Recent advances in the use of techniques in enabling near real-time risk detection augment the capability further.

This chapter will break down the process of integrated pipeline implementation into manageable steps, showing that combining risk detection into the payment processing pipeline may be less daunting than it seems. The chapter will discuss use cases to illustrate the points covered. It provides generic, implementation-agnostic examples that can be translated into specific implementation details by data engineers for any type of payment processing business. The chapter will not provide implementation details that are overly specific to one class of business because these details become rapidly outdated. The focus of the chapter is on integrated end-to-end pipelines and not on specific modules, which may have been implemented differently within different payment processing companies.

11.7.1. Combining Data Streams

Data for payment processing and risk analysis activities originate from distinct processes that run in parallel. Payment processing is responsible for making sure customers can pay, and fund transfer is correctly executed, whereas risk analysis is responsible for recommending actions to protect the company from losses. However, there is a need to combine the data streams generated by these two activities and integrate information in the same data schema, for example a document. The reason is that many regulations require risk detectors to be associated with the action triggered and associated parameters. In fact, event data pipelines for risk analysis take two data streams as input: payment processing events and risk analysis events. The output schema is a merged record that stores the information contained in both streams. However, the functions that generate the respective events for each process run on different timescales, that can have different latencies, depending on internal or external factors. Data for payment processing is becoming mature when it requires little processing/validation and can’t be modified anymore, but data for risk analysis might take longer to generate.

This means that, in general, it is not possible to merge these data streams into a single one. Instead, we build two separate event streams at different speeds, one for payment processing and one for risk analysis. The architecture of the integration module is reasonably simple. It reads payment processing events from one stream, waits for the associated data to be available in the other stream, adopts the associated parameters, and writes the integrated events into an output stream, that can then be consumed by a machine learning algorithm. This module is written as a simple stream application, a simple job, using the built-in connectors, and possibly a small amount of pipelines for any necessary transformations.

11.7.2. Implementing Machine Learning for Risk Detection

As discussed earlier in this chapter, risk analysis can be enriched by many different internal data streams. Once we start analyzing data that come from a different source than the transaction itself, internal data, we usually become less dependent on automated scripts and rules who have to monitor transactions in real time. Automated algorithms are great, but exhausting a high level of false positive or false negative would affect customers' trust or business profitability. It happens often that both numbers don't reach the desired level of accuracy and in order to do that we have to inspect a larger amount of data tied to the fraud risk. A pure parameterization of a score coming from a model usually does not exceed a certain level of accuracy, thus making highly invasive techniques unavoidable. Some companies had to stop relying only on big data techniques working on users' digital traces to build profiles against which transactions have to be validated and started embedding rules based on the knowledge of fraud patterns shared by all players in the systems that rejected genuine transactions, even if those patterns were explainable only in a limited number of circumstances defined a priori.

As a consequence, a valid proposition to reduce the dependency on business rules and heuristic work is the use of predictive algorithms trained on large user behavioral models built using both internal and external data known. By pushing fraud prevention up in the transaction monitoring funnel, and estimating the risk before a transaction enters its last stages at a payment gateway level, we actually free payment gateways from operations looking for legitimate payment going through and yet to be validated. This way they can absorb a much larger traffic load without putting at risk merchants' revenues and customers' buying experience, or even better, guarantee an even faster approval but this time backed up by a sound analysis of the risk.

11.7.3. Case Studies of Integrated Pipelines

This work contains two applied studies of data pipelines for integrated payment processing and risk analysis. The first uses a substantial set of real payment transaction data for learning and modeling transaction amount distributions. This study requires the development of a demand pipeline – a data pipeline that extracts demand signals from social media – that is scheduled to execute essentially concurrently with the transaction pipeline. The fact that both pipelines are required to execute essentially concurrently means that there are complex integration issues arising from the fact that the computational load on the demand pipeline may exceed available capacity for extended periods, so that the transaction pipeline must be allowed to operate at reduced capacity. We establish demand period templates based on analysis of past peaks, and describe data architecture, data flows, and algorithms for demand prediction during demand pipeline allocation.

The second applied study relies on an actual set of real transactions that have been tagged by transaction type, and applies machine learning techniques for fraud detection based on transaction history. Again, this study presents challenges from the need for very low-latency response times, particularly for several coalition demand application areas that most of us experience daily – the groupbuying promo code, the e-commerce shopping cart abandonment, and the online payment rejection. We conclude this work by discussing some requirements and future directions on the integration of payment processing and risk analysis, and what we feel will be directions for research.

11.8. Monitoring and Maintenance of Data Pipelines

Developing data engineering pipelines is just the beginning. Once your system is in production you need to monitor the various components for failure, performance bottlenecks, or data quality issues, so that you can quickly take action. As with your system design, monitoring should ideally be automated in order to not create unnecessary overhead but still notify you in time for issues that require human intervention. Monitoring should help detect the failures across various components in the data pipeline execution. The services you are connecting to might experience some downtime due to various issues, such as being overloaded or going through maintenance. The data transforms can also fail due to changes in the input data such as the arrival of values in non-nullable fields or any change in schema. You will need to set up appropriate alerts and notifications based on how critical the failure is. Monitoring tools can be used to track the keyword and search requests as part of run. Tracking the jobs running durations can help you detect performance bottlenecks. A manual check might be required for low-priority jobs that take longer, but performance degradation for critical hourly runs can require immediate action, which could range from running a retrain based on updated data to adjusting the load in production.

11.8.1. Setting Up Monitoring Tools

To ensure a smooth operation, we need to set up monitoring. The first thing we need to install is the web server, which provides several useful features that already cover a vast number of jobs' typical needs. There are many parameters that can be monitored, and each one of the displayed jobs can have its corresponding parameter configured through several means, either directly in the DAG, as an environment variable or in the job creation web form. These dependencies specific to the job displayed in the web server can be checked using the ’DAG Runs’ window in the UI. It is also possible to monitor whether or not the jobs have failed. In the ’Graph View’ of the web server, those jobs that failed are colored in red. We can also add notifications whenever a job fails. Another great feature is that the web interface provides a log feature. By going to the log corresponding to each task inside the specific DAG, the system will automatically log the output of both the standard output and error output when the job ran. Besides that, it is also possible to integrate other platforms to have more visibility regarding our jobs.

Other monitoring tools can be integrated with the system to send alerts. There are other need-to-have monitoring features not integrated by default that can be easily solved by integrating third-party tools. If we want to monitor data quality issues, certain integrations are recommended. On the other hand, we can implement monitoring tools with an operator, or for monitoring data summary metrics using the subject’s content. It is also possible to use a built-in job service for monitoring our predictions.

11.8.2. Performance Optimization Techniques

It is important to clarify that we are only focusing on high-throughput and low-latency data pipelines. There are more examples that delve into optimizations for very low-latency pipelines feeding ML systems that require sub-second data processing cycles. These types of pipelines serve primarily as input streaming layers for high-performance ML systems.

Data flow systems, given their goal of performing optimizations that are sound and transparent, will perform more conservative optimizations than manually managing a data pipeline. Using some well-known data pipeline technologies, we can make certain guidelines to follow in order to make the best possible optimizations. It is also important to mention here that the optimizations for pulled and pushed batch processing modes will differ. For files that are being pulled from a static source, it is better to perform optimizations that focus on high parallelism. However, for files that are pushed and may be used by various source nodes, or files that are used for fast batch iterations at low scale or random, minimizing overhead and maximizing low latency may be more important optimizations.

11.8.3. Handling Data Quality Issues

As discussed earlier, performing general data quality checks is essential, particularly if you are dependent on third-party providers or other systems for your data source. In the normalized relational storage, it’s fairly easy to check data size for tables storing orders, refunds, chargebacks, and risk processing events to check for discrepancies, as you have already identified the relationships among these systems during the modeling step that responds to business requirements. If you stop receiving data, this indicates a temporary outage in either the source system transformation or the ingestion pipeline. Once the new data ingestion resumes, you need to check your previous data storage and historical data to see if the pipeline is functional and delivering the same volume of data, as well as verify the accuracy of the data. More importantly, to make sure there was no loss of data, you will need to implement compensating mechanisms that can report these events and can be retried, such as failed data grab workflows or failed inserts into the data warehouse.

In a data pipeline that is responsible for storing order transactions in a financial institution, outlier detection checks can be implemented to check if the recent data size is more than or less than the previous time frame data size. This will also depend significantly upon the business being processed and the data movement volume. Having weekly monitoring to check if the latest First-in, First Out row for a data table type is still within the expected volume, say within two standard deviations of the three-month average, could be sufficient in detecting bad data on a real-time basis. During setup time, this check could be performed daily and increased to weekly once the activity has stabilized.

11.9. Compliance and Security Considerations

When designing and building any payment processing pipelines that collect or manage PII or PHI data, there are certain key compliance and security considerations to follow. Payment processing has garnered recent increased scrutiny from both government actors and the general public. Government actors have argued that certain payment processing companies are not doing enough to protect against the monetization of COVID-19 related data. In response to the push by government actors, officials stated that "Cybersecurity is critical to economic security, and economic crime is a national security priority." Public complaints have also risen largely in recent months. It has been reported that certain payment processing companies have been leveraging COVID-19 related data for monetization purposes, and many consumers want assurances that protections will be enforced against these activities.

Any payment processing workflow should be built with security first in mind. Payment processing by nature involves the collection or management of sensitive data, and this means getting it right the first time is of the utmost importance. By ignoring security risks upfront, organizations risk being forced to spend more in the future fixing potential leverage points even after security best practices are implemented, resulting in payment processing workflows that will be sub-optimal from a performance perspective. This applies to both the organization building the data engineering pipeline and the organizations leveraging the data.

11.9.1. Regulatory Requirements in Payment Processing

In payment processing and analysis use cases, the volume of sensitive financial data that is captured and processed—credit card account numbers, Social Security numbers, bank account information, and debit/credit transaction details—warrants strict adherence to regulatory compliance requirements. Sensitive financial data is a lucrative target for bad actors, and to protect consumer trust in financial institutions, governments worldwide have imposed legal requirements for marrying superb customer experience with adequate security practice. In the US, regulatory bodies mandate that financial institutions become compliant with the requirements of the applicable regulations in the industry.

Both retail and eCommerce merchants are subject to PCI DSS, which is the most comprehensive data protection program that sets the global standard for protecting sensitive financial data. Merchants processing more than 6 million credit card transactions a year will be assigned the Level 1 risk category and must undergo an annual on-site assessment conducted by a qualified security assessor. PCI DSS requires merchants to meet compliance checklists covering all aspects of their operations, from the physical security in their offices and data storage facilities, to their internal technology and information security policies, to policies for their vendors. Each aspect of the company's operations must be assessed to ensure that every data security practice ensures the privacy of data traffic between consumers, banks, payment processing platforms, and credit card companies.

Like PCI DSS, the Healthcare Insurance Portability and Accountability Act also imposes strict compliance requirements for enterprises operating in the healthcare sector to protect sensitive consumer data. However, in addition to HIPAA, organizations involved in the payment processing and risk analysis of health insurance also must comply with a host of laws related to data security and privacy. Consent and breach notification laws, data protection laws, and consumer protection laws also exist at the state level. Additionally, international laws also govern payment processing in these regions.

11.9.2. Data Privacy and Security Best Practices

Data storage is a common characteristic of data engineering pipelines. In payment processing and risk analysis, data encryption is strongly encouraged for storage of sensitive data such as payment information and personally identifiable information (PII).

It is best to encrypt sensitive data before it is transmitted to the target and at the target using both client-side and server-side encryption, respectively. A Content Management Interoperability Services (CMIS)-like approach is highly recommended to ensure that sensitive information (e.g., PII) can be optionally decrypted for authorized third-party access only. Access to sensitive information should be dealt as carefully as database access control. External third-party access to sensitive information should be logged, especially when sensitive information is decrypted through a CMIS-like workflow.

It is of utmost importance that data security is in place throughout the entire data life cycle, including data transmission, storage, and deletion. Sensitive information has to be deleted securely after its designated retention mandates expire. Laser or degausser has to be used to destroy unused sensitive data in hard disk (HDD) type mass storage devices due to its high sensitivity to physical destructing devices. Even after routines like "formatting" or "distributed data destruction", the data on the HDD can be maliciously retrieved, and secure data deletion procedures for HDDs have been proposed. All the data are permanently "erased", i.e., written with zeroes and ones alternately, using at least one of the above procedures. For magnetic tape (MT) mass storage, overwriting with new data or zeroing with degausser are common secure deletion procedures. Utilize safe and secure data management software for any safe and secure storage, deletion, and transfer of sensitive data.

11.9.3. Risk Mitigation Strategies

In order to mitigate the various risks discussed in the previous sections, transactions should be processed in real time, meaning at the point-of-sale or on the e-commerce platform. This allows validation against a number of parameters. First, the transacting account should be compared with the historical risk pattern derived from machine learning models or heuristic methods. For online transactions, IP geolocation should correlate with the account's historical data. A large distance from the country where the card is usually used can generate a flag or a score. Other real-time factors such as velocity checks can further enrich the monitoring. For online e-commerce websites, do-not-serve lists of both black and white, previously known-to-be-good merchants can be used to further reduce risks. For others, lists of recently served merchants through a previously served account can be an indication of merchant engagement. All these are used as a risk response mechanism that can be completely automated, signaling either a successful authorization or the opposite either for soft decline or manual review.

Traditional machine learning classification models can be used to validate the conditional transaction score or the risk assessment. In practice, a combination of rules and models, where the model triggers a score only for certain conflictive transactions but where the majority flagging is done by thresholds, have been proven useful.

Successful automation where a good percentage of transactions are either accepted or on the other side flagged/soft-declined can save hours of manual review. Both the traffic volume and the service quality—and how quickly the system flags a conflictive transaction are important operational KPIs for the business. Manual review is necessary, but its cost must be considered when designing fraud strategy.

11.10. Future Trends in Payment Processing and Risk Analysis

Historically, innovation in payment systems has been sporadic. Periods characterized by little change are punctuated only by the introduction of new channels to existing payment methods but no real change in the payment processing itself, prompting people to ask. However, recent years have seen an explosion of effort in new payment processing technologies, and life is no longer so quiet for payments as it once was. Technology is transforming both the systems used to facilitate payment and the consumer experience of the transfer — but, while exciting, these hostile takeovers have created a swirling center of gravity. Merchant apps, shopping carts, loyalty programs, and resellers are delivering payments via off-line, online/direct, and mobile channels. Digital wallets can hold different payment types, but nobody yet knows which protocol might emerge, or if the space is so fragmented as to ensure that somebody only ever makes money off of fees.

Many parts of the world remain underbanked. As a result, payment processing will likely continue to evolve towards solutions that require little to no banking: id-based methods or credit-based methods. These methods will rely on systems that operate outside of the existing features of major banking protocols, or innovate slightly at their border, until some intern pulls the plug on various functionality once used by popular apps, but now discovered lying fallow.

As more payment can be processed instantaneously, allowing micro-transactions to leap from niche to widespread usage, some businesses will break down services and products into their atomic parts, charging people for exactly how much of a resource they consume on a much more immediate basis. While much of this phenomenon will be enabled by greater speed or lower friction, some might also be driven by social expectations around fairness.

11.10.1. Emerging Technologies in Payment Systems

The global payment industry is currently in the throes of a transformative phase, characterized by seismic shifts in operational foundations and infrastructural architectures – volatile forces of evolution resulting in the disruption and disintermediation of players who have previously maintained well-defined positions within static ecosystems. Natural ascribees to this description include providers of services relating to foreign exchange, processing, acquiring, card issuing as well as OEMs, High Street banks, primary card schemes, neobanks, and – perhaps most spectacularly – established players from outside the financial services industry such as payment facilitators, credit agencies, vendors and various companies. So, too, are payments industry stakeholders. It is perhaps inevitable that these technologies shape-shift into innovative revenue and earnings models. The extrapolation may well serve to set the path for the payment industry and, by extension, its payments solutions providers, for the next three decades. In opposite parallel, two other technological currents exemplify major challenges and serve to shape this new world order – those concerning quantum computing and artificial intelligence, as well as blockchain technology and cryptocurrency.

Specific details of the innovations shaping this evolution cycle are clear. Generally, however, some caution is warranted as the rate of technological churn currently underway would appear to indicate. At one extreme, the view is expressed that some technologies, such as APIs, mobile wallets, open banking, and cryptocurrency, are nearing the end of their evolution cycles. At the same time, the news coverage on other technologies appears to reflect technologies returning from a long dormancy. So, too, do partnership models, such as Banking-as-a-Service and Banking Relationships-as-a-Service, emerge as models focusing on inter-organizational cooperation with regard to technology investment risk sharing, in contrast to traditional financial services which rely on both scale and scope to achieve cost structures suited to optimizing externally identified organizational niche targeting.

11.10.2. The Role of AI and Machine Learning

Over the past few years, AI and machine learning have changed from buzzword to application. With growing amounts of historical data available around risk, fraud, and customer experience in the financial world, coupled with improvements in algorithms and processing power, AI/ML is seeing deployment maybe inappropriately at scale. However, deep application of AI and ML in risk management, payment processing, fraud detection, and customer experience prediction is still being rolled out. The historical core systems in these businesses are cautious to embrace dependence on these new technologies. But the experimental pockets of painful learning architectures and events are scaling to enterprise-level applications. The regulatory environment has contributed to the caution. Yet at the same time, regtech is sweeping into this space with both AI methods and automation. Scalable automation is an absolute requirement to meet the increasing expectations businesses face globally from regulators, customers, and partners in the area of compliance around AML, KYC information delivery, and transaction monitoring. As a result, there is a tension between experimentation with AI/ML in risk management in payments and the immediate regulatory requirements that need to be satisfied by these businesses. Increasingly, the regtech vendors are using AI/ML in these applications to facilitate compliance with automation and speed. At the same time, the core industry players are working on AI solutions for frictionless experience improvement around payment for their consumer branches.

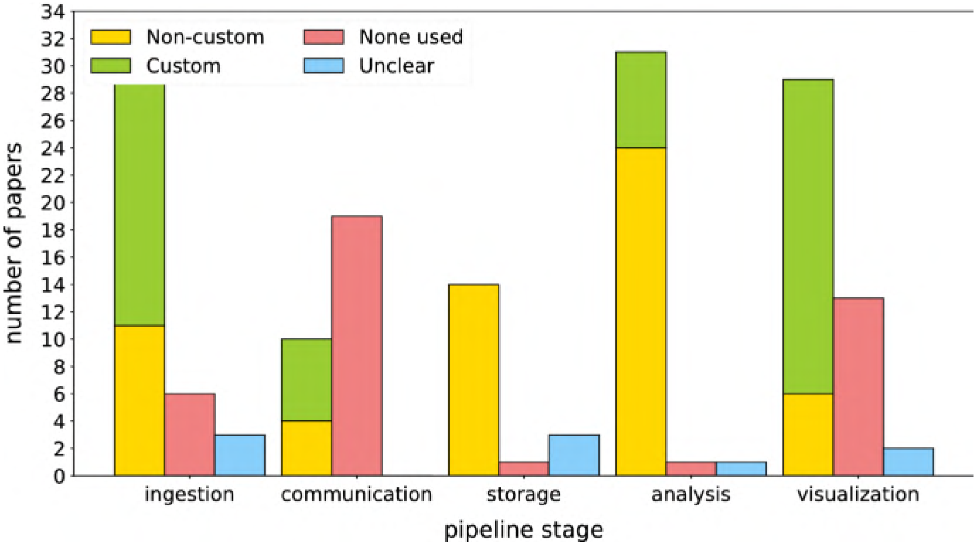

Fig : Manufacturing process data analysis pipelines

11.10.3. Predictions for the Future Landscape

Advances in secure digital payment technologies are likely to increase the inclusion of underserved classes into the payment processing universe. Blockchain payments aim to replace banks in the function of transaction intermediaries. Direct transactions between consumers, merchants, and financial product offerings have the potential to limit the profit share of banks and credit card companies while facilitating transactions of people in locations with poor access to bank facilities.

Banks are pursuing digital wallets and partnerships around using their services, such as creating bank accounts and lines of credit for underbanked customers. Traditional credit card companies respond to Blockchain payment offerings with speed and fraud reduction innovation. They are also developing digital wallets that give customers access to rewards programs as well as better transaction experience.

AI-enabled risk management innovation is expected to rely on deep data—propagating historical patterns of transactions that have detected fraudulent activities in the past—

along with contextual analysis. This contextual analysis includes considering the unique profile of the store where the transaction is taking place, or AI's real-time understanding of the context under which it is happening. It recognizes users in the immediate vicinity of the place where the transaction is taking place and considers their recent transaction profiles. Risk management innovation further relies on technology that reduces the costs of fake accounts, which is a requirement for implementing any of the various predictive models that rely on historical data patterns. It also strives to maximize the use of AI for source detection and risk scoring of third-parties involved in any transaction.

11.11. Conclusion

Due to the rapid growth of online payment processing and increasing compliance requirements in the finance domain, we have created a novel data engineering pipeline that onboards data quickly, processes many data sources, and handles large amounts of transactional data. Our industrial pipeline uses a large-scale data hub model that efficiently addresses data lake and data warehouse anti-patterns. We have demonstrated our capabilities by sharing real-world examples of how data is collected and joined in our data hub model centered on payments data, how pipelines are designed for agile onboarding of large amounts of raw data, how different sources of merchant nontransactional data are handled, the creative ways we use to detect spatial outliers in merchant activity payouts, and our automated ticketing for merchant accounts' hard and soft inquiries that issue alerts about risk or fraud. Additionally, we emphasize the importance of creating performant pipelines by combining heavy data engineering loads with modular analytic and machine learning cores that serve multiple projects within an organization. Technology has drastically changed how people buy and sell goods. Instead of shopping in a physical store, people now make their purchases online, at any day, and at any time. To support this, payment processors take on the role of intermediaries between buyers and sellers by offering merchant accounts. Payment processors accept digital transactions made with credit cards, debit cards, or mobile wallets. They pay the merchants and collect a service fee for their work from the merchant's bank account. These payments from payment processors to merchants generate millions of records every day, but not everyone lives happily ever after. Merchants can be tempted to commit tax evasion, money laundering, or fraudulent charges. What do payment processors do to help merchants when something goes wrong? They have a team of risk analysts, data scientists, and investigators helping to ascertain the legitimacy of merchant transactions. But how do these analysts get the data they need to investigate?

References:

Akidau, T., Bradshaw, R., Chambers, C., Chernyak, S., Fernández-Moctezuma, R. J., Lax, R., ... & Whittle, S. (2015). The dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. Proceedings of the VLDB Endowment, 8(12), 1792–1803.

Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google File System. ACM SIGOPS Operating Systems Review, 37(5), 29–43.

Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.). Wiley.

Sadoghi, M., & Jacobsen, H. A. (2011). BE-Tree: An index structure to efficiently match Boolean expressions over high-dimensional stream data. Proceedings of the VLDB Endowment, 4(7), 540–551.

Zhang, L., & Xu, D. (2020). Real-time fraud detection in payment systems using big data analytics. Journal of Computer Information Systems, 60(6), 562–570.