



Version 2.2

Training Plan for Implementation of AML Transaction Monitoring with Jube (May 2025)

This document outlines comprehensive remote training for implementing Jube, an open-source Anti-Money Laundering (AML) transaction monitoring system in line with Jube's compliance guidance document.



Amendments	2
Introduction	3
Program Outcomes	3
Training Plan	4
Day 1: Configuring Models, Messaging and Rule Strategy	4
Day 2: Suppressions, Sanctions, Machine Learning and Case Workflow	12
Day 3: Permissions, Database, Reporting, Reprocessing and Extensibility.....	19
Day 4 (Optional): Developer Workshop	24
Conclusion	29

Amendments

Date	Author	Version	Description
14 th March 2025	Richard Churchman	2	Updated for version 2 of the training materials.
15 th April 2025	Richard Churchman	2.1	Updated to reflect remote only.
1 st May 2025	Richard Churchman	2.2	Harmonisation with the Training Plan for Predictive Analytics with R document.



Introduction

While transaction monitoring encompasses a broad spectrum of activities, this remote training specifically focuses on the AML (Anti-Money Laundering) use case. The remote training is designed to support compliance efforts and is structured around this guidance. This guidance is intended to be adapted by compliance managers to align with an organization's specific regulatory obligations, which are often derived from Financial Action Taskforce (FATF) guidelines and further reflected in the Wolfsberg Principles.

It is important to note that Jube operates as a real-time system. As a result, while the remote training emphasizes implementation of this guidance other transaction monitoring use cases will also be comprehensively addressed by implication. The underlying concepts are largely adjacent and require similar methodologies, albeit with subtle differences in application.

Program Outcomes

- Confidently Implement and Manage Jube: Master Jube's AML transaction monitoring (FATF/Wolfsberg-aligned), UI navigation, API integration (e.g., Postman), and rule strategies (Gateway/Abstraction/Activation Rules) for fraud/AML detection.
- Optimize Jube for Performance & Security: Configure caching, sanctions screening, and suppression techniques (keys/total/rule-based); use TTL counters for high-volume transaction tracking.
- Accelerate Adoption with Best Practices: Build/evaluate predictive models (fraud/credit risk); implement ML via Exhaustive training; design case management workflows (tracking, forms, audits).
- Troubleshoot & Debug (Developers): Explore Jube's architecture (Docker/Redis/Postgres), debug with INFO-level logs, and manage async processes (AMQP) and background threads.
- Leverage Advanced Compliance Features: Apply fuzzy matching (Levenshtein) for sanctions, trigger real-time alerts/cases via Activation Rules, and create custom dashboards for oversight.



Training Plan

The training spans 3 to 4 days, with each day lasting 6 to 8 hours, depending on elective modules. Elective modules provide in-depth training on advanced administrative concepts using dedicated servers. These are ideal for technical participants responsible for system administration but do not require developer expertise.

For organizations with developer teams, an additional day is available to familiarize them with the Jube codebase for business continuity. This session covers software patterns, frameworks, coding standards, and hands-on debugging.

Day 1: Configuring Models, Messaging and Rule Strategy

This session introduced Jube, an open-source AML transaction monitoring tool aligned with FATF and Wolfsberg Principles. Key topics included:

The Jube UI and core concepts like navigation, rule building, and data extraction via Request XPath. Participants learned to configure AML models, enable sanctions screening, and integrate APIs using Postman.

Rule development was a major focus—Gateway Rules for filtering transactions, Abstraction Rules for data aggregation, and Activation Rules for triggering actions like alerts or case creation. Hands-on exercises reinforced these concepts.

An optional deep dive covered Jube's architecture, including Docker deployment and log management. The training equipped attendees with foundational AML monitoring skills, setting the stage for advanced topics in Day 2.

Time	Topic	Description
09:00 – 09:00	AML Monitoring Compliance Guidance Overview	<p>While transaction monitoring covers a wide range of activities, Jube specifically targets the Anti-Money Laundering (AML) use case. To enhance clarity and precision, the Jube AML Monitoring Compliance Guidance has been developed and is meticulously maintained. Guidance is designed to assist compliance managers in tailoring it to meet their organization's unique regulatory requirements. These obligations are often rooted in the Financial Action Task Force (FATF) guidelines and further reflected in the Wolfsberg Principles, which are updated periodically to reflect evolving standards.</p> <p>This document provides a comprehensive framework for monitoring compliance with Anti-Money Laundering (AML) regulations using Jube, an open-source fraud prevention and transaction monitoring tool. The guidance aligns with guidance from the Financial Action Task Force (FATF) and the Wolfsberg Principles, focusing on transaction monitoring and risk-based approaches to AML compliance.</p>

		<p>Jube facilitates the execution of the AML compliance guidance, making it the most relevant tool to highlight its features, ensuring an assured compliance outcome. By leveraging its advanced capabilities, Jube enables financial institutions to effectively monitor transactions, detect anomalies, and adhere to regulatory requirements, thereby enhancing overall compliance efforts.</p> <p>This session aims to offer an overview of the guidance and establish the foundation for the features and capabilities that will be introduced.</p>
09:00-09:10	Break	Break
09:10-09:30	Introduction to Jube and Key User Interface Concepts	<p>In the context of the AML Monitoring Compliance Guidance, this session will provide a comprehensive introduction to Jube. The first session will offer a quick overview of its functionality, followed by a deeper dive into the core user interface concepts. These concepts include recurring themes and consistent design patterns that apply across all pages of the platform. Key elements to be covered are:</p> <ul style="list-style-type: none"> • Authentication: How to log in and out of Jube, as well as change passwords. • Main Menu Navigation: An overview of Jube's layout and where to locate key features. • Parent Objects: The primary groupings of functionality in the system and how these entities are created. • Child Objects: The tree-based navigation used to configure child objects associated with the main entities. • Locked and Active States: How critical configurations are made immutable to ensure sensitive abstractions (e.g., activation rules and machine learning dependencies) remain unaltered. • Response Payload: The flags that send data back to calling systems or emit to a service bus. • Report Table: Fields displayed in an expanded and indexed format to enable direct reporting from the database. • Rule and Code Builder: The component used to create rules across various areas of the system. <p>During the session, participants will log into their respective tenant environments on the server to explore these features through hands-on practice. This</p>

		approach ensures users gain practical experience with Jube's interface and functionality.
09:20-09:40	Models and HTTP Messaging	<p>Jube is designed with flexibility in mind, operating without reliance on a fixed data model or predefined API for data processing, despite its strong emphasis on Anti-Money Laundering (AML) compliance. This session will introduce the concept of models and their critical role within the platform.</p> <p>Participants will configure a model specifically tailored for AML use cases, with a detailed explanation of the associated transaction flows to demonstrate how data is processed within the system.</p> <p>The session will cover the following key components:</p> <ul style="list-style-type: none"> • Keys: Configuring date and identifier fields. • Cache Storage: Enabling and setting up cache storage for full transaction payloads or time-to-live counters. • Sanctions: Enabling sanctions functionality. • Activation Watcher: Enabling Activation Watcher and setting limits to prevent system saturation. • Response Elevation: Configuring limits in response elevations to avoid severe outcomes upstream of Jube or system saturation. <p>Additionally, the session will introduce the concept of HTTP API calls. Using Postman, the process of integrating with Jube will be demonstrated, providing a hands-on example of how to interact with the platform programmatically. This practical exercise will help users understand how to connect to Jube and leverage its capabilities within their own environments.</p>
09:40-11:00	Request XPath Payload	<p>As previously highlighted, Jube operates without relying on a fixed data model or predefined API. Instead, it is dynamically constructed to suit specific use cases, with a strong emphasis on Anti-Money Laundering (AML). Certain values and processes hold special significance, particularly when used in abstraction rules or machine learning models. This session will focus on the Request XPath page and explore key concepts, including:</p> <ul style="list-style-type: none"> • XPath: Extracting basic data from the payload based on its position in the provided JSON. • Data Types: Assigning specific data types to extracted data, which is critical for rule

		<p>processing (e.g., cannot apply "greater than" logic to text).</p> <ul style="list-style-type: none"> • Suppression: Flagging to determine if a string value can be used with the suppression functionality, which will be covered in Day 2. • Search Keys and Cache: Determining whether the extracted value can be used to query the cache. • Time To Live (TTL): Deciding if the extracted value can be used to increment TTL counters. • Date Handling: A discussion on date handling, including suggested formats for optimal processing. <p>By the end of this session, the integration environment setup will be finalized, enabling the course to transition into implementing the guidance with a focus on compliance requirements. This will provide participants with a strong foundation for applying Jube's capabilities to real-world AML scenarios, ensuring they are well-equipped to meet regulatory and operational demands.</p>
11:00-11:10	Break	Break
11:10-11:30	Inline Functions	<p>Inline functions are used to combine or process data extracted via Request XPath (e.g., First Name and Last Name) to create new fields (e.g., Full Name) for real-time processing, such as sanctions screening. In this session, we will introduce the VB.net syntax for implementing inline functions. Participants will learn how to concatenate elements (e.g., combining First Name and Last Name into Full Name) to prepare data for presentation to the Sanctions subsystem. These concepts will be further explored and applied in subsequent relating to Sanctions Fuzzy matching.</p>
11:30-11:50	Gateway Rules	<p>Gateway Rules serve as a pre-filtering mechanism, reducing the volume of transactions that require more intensive processing in later stages. By applying upfront sampling and filtering criteria, Gateway Rules streamline workflows, ensuring only the most relevant or high-risk transactions are forwarded to resource-intensive functions like advanced analytics or sanctions screening. This approach enhances efficiency and optimizes system performance by directing resources toward the transactions that matter most.</p> <p>In this session, participants be introduced to building rules in Jube for the first time. Participants will learn how to create rules using both the builder (a visual,</p>

		user-friendly interface) and the coder (a more advanced, code-based approach). This foundational knowledge will prepare the participants in designing and implementing effective filtering and processing rules within the Jube platform.
11:50-12:30	Abstraction Rules	<p>The Abstraction Rules functionality is where rules are defined to aggregate data stored in the cache database. This is one of the most critical components of the model invocation process, as it generates aggregate statistics based on transaction or event data retained in the cache. These rules transform raw data into meaningful insights, enabling the system to analyse trends, patterns, and summaries that support decision-making and further processing. Abstraction Rules are essential for creating a structured and efficient framework for data analysis within the system.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Abstraction Function Types: Learn about the different types of functions used to aggregate and process data. • Rule Building and Coding: Explore how to create and implement Abstraction Rules using both visual and code-based approaches. • Offsets: Understand how offsets are used to define time windows or data ranges for aggregation. <p>Additionally, the session will introduce the implications of Abstraction Rules and their application in Activation Rules and Machine Learning, highlighting their role in enabling advanced analytics and decision-making processes.</p>
12:30-12:50	Abstraction Calculations	<p>An Abstraction Calculation involves comparing the return values of two Abstraction Rules using straightforward arithmetic operations or VB .Net code fragments. For example, it can be used to divide one value by another to create a ratio, which is often highly useful in machine learning applications. This process is most employed to generate ratios from the outputs of Abstraction Rules, providing valuable insights for further analysis.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Demonstrate how to create common ratios using Abstraction Calculations.

		<ul style="list-style-type: none"> Highlight the implications of these ratios in Activation Rules and Machine Learning, showcasing their role in enhancing analytical capabilities and decision-making processes. <p>By the end of the session, participants will understand how to leverage Abstraction Calculations to derive meaningful metrics and apply them effectively within the system.</p>
12:50-13:50	Lunch	Lunch
13:50-14:40	Activation Rules	<p>Up to this point, all processing has focused on generating a wealth of data related to the event or transaction as it moves through the model, but no concrete action has been taken. For instance, in a transaction fraud prevention platform, the goal might be to decline a transaction if it exceeds a specific volume threshold.</p> <p>Activation Rules are designed to take actionable steps, such as:</p> <ul style="list-style-type: none"> Declining transactions (though this may not always be suitable for AML use cases). Creating cases in the case management system. Incrementing TTL counters. Sending email or SMS notifications with tokenization, either directly or as a sample. <p>In this session, the participant will learn how to bring all these efforts together to invoke a response escalation. Specifically, the participants will configure the system to forward an email notification with tokenization. Case management functionality will be covered in detail on Day 2. This session will focus on integrating the components necessary to trigger actionable responses based on the data processed.</p>
14:40-15:00	Lists and Dictionaries	<p>The Lists Page enables the creation of data lists that can be referenced as operators in all rules within the models. For example, it can be used to maintain a list of flagged customer account identifiers, which can then be referenced using either the Builder or Coder. Lists are particularly useful for managing large Deny and Accept lists without embedding excessive complexity into individual rules—instead, only the list reference is embedded.</p> <p>Dictionaries function similarly to lists but with an added layer of functionality. Instead of simply matching a</p>

		<p>string value, dictionaries pair a key from the payload during model invocation and return a corresponding value. This makes dictionaries an effective way to enrich payload data with small amounts of external information without relying on inline scripts or complex integrations. A key advantage of dictionaries is that they can be created manually via the user interface, uploaded as a CSV file, added directly to the database table, or integrated via an API. This flexibility allows end-users to manage these lookups without requiring technical intervention.</p> <p>Together, Lists and Dictionaries help reduce the volume of activation rules and enhance their elegance by simplifying data management and integration.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Creation of List Entities: Learn how to set up and manage lists. • Manual creation of list values: Add individual entries to lists. • File upload of list values: Import list data via CSV files. • Creation of Dictionary entities: Set up dictionaries for key-value pair lookups. • Manual creation of dictionary key-value pairs: Add individual key-value pairs. • File upload of key-value pairs: Import dictionary data via CSV files. • Integration of lists into Activation Rules: Use lists to streamline and simplify rules. • Integration of dictionaries into Activation Rules: Enrich payload data and enhance rule functionality. <p>By the end of the session, participants will understand how to effectively use lists and dictionaries to optimize rule sets and improve system efficiency.</p>
15:00-15:10	Break	Break
15:10-16:20	Workshop	<p>By this stage, participants will have acquired a foundational knowledge of Jube, and the skills needed to build a basic transaction monitoring system capable of real-time transaction decline and notification.</p> <p>The session covers:</p>

		<ul style="list-style-type: none"> • Present key areas from the AML Monitoring Compliance Guidance Overview to provide context and regulatory insight. • Facilitate a collaborative exercise where participants work in pairs or groups to design and implement a rudimentary transaction monitoring strategy. • Guide participants in creating a few Activation Rules to support the strategy, focusing on real-time decision-making and notifications. <p>This hands-on activity will reinforce the concepts learned so far and demonstrate how to apply them in a practical, compliance-focused scenario. By the end of the session, participants will have a clearer understanding of how to build and deploy a functional transaction monitoring system within Jube.</p>
16:20-16:30	Dismissed	<p>To conclude the day, we'll recap the key concepts and skills covered, ensuring participants have a solid understanding of the foundational elements of Jube and how to apply them in building a basic transaction monitoring system.</p> <p>Session Wrap-Up:</p> <ul style="list-style-type: none"> • Review of Key Takeaways: A summary of the day's learning objectives, including rule creation, data aggregation, and real-time decision-making. • Q&A Session: An opportunity for participants to ask questions and clarify any uncertainties. • Next Steps: An overview of what to expect on the next day, including deeper dives into case management and advanced rule implementation. <p>Class Dismissal:</p> <ul style="list-style-type: none"> • Participants are free to leave unless they have chosen to stay for elective sessions. • Those staying behind for electives will receive further guidance on the additional topics or hands-on exercises available.
16:30-17:30	Elective: Architecture and Caching, Environment Variables,	<p>In this elective session, participants will dive deeper into the architecture of Jube and gain hands-on experience installing a version of Jube on clean Digital Ocean servers using the provided Docker Compose file.</p> <p>The session covers:</p>

	Installation and Log Configuration.	<ul style="list-style-type: none"> • Jube Architecture Overview: Explore the core components of Jube's architecture, including its modular design and data flow. Understand the roles of key services such as the cache database, rule engine, and activation modules. • Setting Up Digital Ocean Servers: Step-by-step guidance on provisioning a clean Digital Ocean server for Jube installation. Overview of server requirements and configuration best practices. • Installation via Docker Compose: Walkthrough of the Docker Compose file provided for Jube. Detailed instructions on deploying Jube using Docker Compose. • Environment Variables and Configuration: Discussion of essential environment variables and their role in customizing Jube's behaviour. Tips for managing and securing environment variables in a production setup. • Log Configuration and Management: Overview of Jube's logging framework and how to configure log levels and outputs. Best practices for monitoring and troubleshooting using logs. <p>By the end of this elective module, participants will have a deeper understanding of Jube's architecture, and the practical skills needed to deploy and configure Jube in a cloud environment. This knowledge will be invaluable for those looking to implement Jube in their own organizations or for further experimentation and learning.</p>
--	-------------------------------------	---

Day 2: Suppressions, Sanctions, Machine Learning and Case Workflow

The training day focused on advanced AML monitoring capabilities in Jube, beginning with suppression management techniques for controlling automated actions on flagged accounts. Participants then explored sanctions screening using fuzzy name matching powered by the Levenshtein algorithm, followed by an examination of TTL counters as an efficient method for high-volume transaction tracking without full data storage.

The machine learning segment introduced fundamental concepts before detailing Jube's proprietary Exhaustive training system that automates model development. The case management deep dive covered the complete workflow lifecycle from initial alert to resolution, including status tracking, custom forms, and audit capabilities. A practical workshop allowed attendees to implement these features in building a functional AML monitoring system.

For those staying late, the elective session provided insights into operational aspects like transaction tracing and configuring high availability environments. By day's end, participants gained comprehensive hands-on experience with Jube's advanced compliance features,

equipping them to deploy real-world AML solutions. The training successfully bridged theoretical knowledge with practical implementation skills across the platform's sophisticated monitoring and investigation tools.

Time	Topic	Description
09:00-09:20	Suppression	<p>In scenarios where a customer's transactions are declined in real-time, but further investigation reveals a legitimate reason (such as being on holiday), it may be necessary to prevent any automated actions triggered by system rules for that account.</p> <p>This session will cover the following approaches to managing such cases:</p> <ul style="list-style-type: none"> • Suppression Keys: Tools to temporarily pause actions for specific accounts or conditions. • Total Suppression: Completely stopping all automated actions for an account. • Suppression by Rule: Selectively preventing actions based on specific criteria or rules.
09:20-09:40	Sanctions Fuzzy Matching	<p>Sanctions lists are published by various regulatory bodies and contain names of individuals or entities with whom business is prohibited. Jube provides functionality to load these sanctions lists into its engine, enabling name matching using fuzzy logic based on the Levenshtein Distance algorithm. This matching capability can also be integrated into the Entity Analysis Model for real-time processing. To ensure optimal performance, sanctions data is stored in-memory, allowing for extremely fast recall.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Introduction to the Levenshtein Distance Algorithm: An overview of how this fuzzy logic algorithm works for matching names. • The Sanctions Page in Jube: A dedicated feature for performing individual and batch name matching against loaded sanctions lists. The matching logic used here is identical to the real-time functionality available during model invocation. • Sanctions Matching via Model Invocation: How sanctions matching can be performed during model invocation by extracting and processing the multipart string from the data payload.
09:40-10:00	Time To Live (TTL) Counters	<p>Time-to-Live Counters (TTL Counters) offer an efficient method for creating model memory without storing complete transaction or event data in cache tables. TTL</p>

		<p>Counters are incremented when an Activation Rule is triggered and are designed to expire after a specified period, after which they automatically decrement.</p> <p>Unlike traditional caching, which stores the full details of a transaction or event, TTL Counters only store lightweight counter entries. This approach significantly reduces storage requirements, making TTL Counters particularly useful for handling ultra-high transaction volumes or scenarios requiring long-term retention of historical data. By storing minimal information, TTL Counters enable extended retention periods compared to storing full transaction data in cache. For rules that rely on basic counting (e.g., tracking occurrences), TTL Counters are a more efficient alternative to Abstraction Rules.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Creating TTL Counters: How to set up TTL Counters, including defining their lifespan (tenure). • Incrementing TTL Counters: How TTL Counters are incremented when an Activation Rule is triggered. • Using TTL Counters in Activations: How to incorporate TTL Counters into Activation Rules for efficient event tracking.
10:00-10:10	Break	Break
10:10-11:00	Introduction to Machine Learning	<p>This session will provide a high-level exploration of regression models and neural networks within the context of supervised learning, specifically aimed at fraud prevention. While the discussion will not focus on Jube, it will cover general concepts and principles applicable to machine learning.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • The Process of Supervised Machine Learning: An introduction to the workflow and key steps involved in supervised learning. • Dataset Terminology: Key terms and concepts related to datasets, such as features, labels, training data, and test data. • Linear Regression Primer: A basic overview of linear regression and its applications. • Logistic Regression Primer: An introduction to logistic regression and its use in classification tasks.

		<ul style="list-style-type: none"> • Neural Networks Primer: A foundational explanation of neural networks, including their structure and function. • Topology: An overview of neural network architectures and how they are designed. • Feature Engineering: Techniques for selecting and transforming features to improve model performance. • Overfitting: Understanding overfitting and strategies to prevent it. • Unsupervised Machine Learning: A brief introduction to unsupervised learning and its differences from supervised learning. <p>The session aims to provide participants with a solid understanding of fundamental machine learning concepts. It will also highlight how Exhaustive integrates these machine learning principles with abstraction concepts to create effective fraud prevention solutions.</p>
11:00-11:10	Break	Break
11:10-11:50	Exhaustive Machine Learning Training	<p>Machine learning is inherently complex, but Jube simplifies this process with its embedded training and recall algorithm, Exhaustive. When Abstraction Rules are properly defined, Exhaustive eliminates the guesswork involved in topology exploration and model training.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Exhaustive Adaptation Concepts: An introduction to the core concepts behind Exhaustive. • Topology Evolution and Adaptation: How Jube dynamically evolves and improves neural network topologies. • Training Statistics: How to interpret and understand the statistics generated during the training process. • Promoted Model: A breakdown of the components that make up the promoted model. • Manual Recall: How to manually recall the model for testing and evaluation. • Real-Time Recall: How the model is recalled and applied in real-time scenarios. <p>This session will provide a comprehensive understanding of how Exhaustive streamlines machine</p>

		learning within Jube, making it more accessible and efficient.
11:50-12:00	Break	Break
12:00-12:20	Activation Watcher	<p>The Activation Watcher serves as a real-time ticker for Response Elevations, triggered when an Activation Rule is configured to send messages to it.</p> <p>This session covers:</p> <ul style="list-style-type: none"> • Setting up rules to send activations to the Activation Watcher. • Streaming activations in real time. • Replaying past activations. <p>It offers an engaging and practical insight into real-time transaction monitoring within Jube.</p>
12:20-12:40	Tags	<p>Jube relies on Response Elevations to drive decisions, which, like all systems, may occasionally produce errors. Tags enable feedback collection to power advanced analytics and machine learning, aiming to minimize such errors.</p> <p>This session covers:</p> <ul style="list-style-type: none"> • Configuring tags in Jube. • Invoking tagging via a remote HTTP endpoint (currently unavailable in the user interface, as it's designed for external data sources). <p><i>Note:</i> If future updates enable tagging within the user interface, it will be integrated into the case management system.</p>
12:40-13:40	Lunch	Lunch
13:40-15:00	Case Workflows	<p>A case workflow represents a logical sequence of tasks assigned to an agent or team. For instance, the workflow for evaluating transaction fraud (e.g., stolen debit cards) may differ significantly from an anti-money laundering workflow. When work efforts or team responsibilities vary, organizing these into distinct workflows ensures efficiency and clarity.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Case Workflow: How to create and configure a case workflow. • Case Status: An introduction to case statuses and status elevations.

		<ul style="list-style-type: none"> • Case Request XPath: Defining the fields to be displayed in case history. • Case Workflow Filters: Creating custom views to organize and prioritize outstanding work. • Activation Rules: Configuring case creation based on activation rules. • Searching Cases and Case Sessions: Searching for cases using filters or manual searches, including saving search definitions to a session. • Basic Case Operations: Actions such as locking, unlocking, and adding diary entries. • Case Notes and Actions: Creating unstructured notes and assigning actions that can drive integrations. • Case Dispositions: Managing case statuses, star grading, locking, and diary entries. • Skim and Lock: Moving to the next record based on the filtering criteria defined in the user's session. • Audit: Tracking previous actions on a case, including detailed audit logs. • File Upload: Uploading and associating files with a specific case. • Case Macros: Creating macros to trigger notifications or integrations, leveraging JavaScript. • Case Forms: Designing custom forms for data capture, integrations, and notifications. • Case Display: Creating custom displays for the triggering transaction. • Case Status Elevation: Understanding the upward (but never downward) classification of case statuses. <p>This session will provide a comprehensive overview of case management functionality, equipping participants with the knowledge to effectively manage and streamline workflows.</p>
15:00-15:10	Break	Break
15:10-16:20	Workshop	<p>By this stage, the class will have not only created a rudimentary transaction monitoring solution but will also have the tools to manage the process across the first line of defence, second line of defence, and escalation to the Money Laundering Reporting Officer (MLRO).</p> <p>Using the Anti-Money Laundering (AML) Monitoring Compliance Guidance, the class will now proceed to implement a case management system designed to</p>

		support broader compliance efforts. This system will enable effective monitoring, investigation, and reporting in line with regulatory requirements.
16:20-16:30	Dismissed	<p>To conclude the day, we'll recap the key concepts and skills covered, ensuring participants have a solid understanding of the foundational elements of Jube and how to apply them in building a basic transaction monitoring system.</p> <p>Session Wrap-Up:</p> <ul style="list-style-type: none"> • Review of Key Takeaways: A summary of the day's learning objectives, including rule creation, data aggregation, and real-time decision-making. • Q&A Session: An opportunity for participants to ask questions and clarify any uncertainties. • Next Steps: An overview of what to expect on the next day, including deeper dives into case management and advanced rule implementation. <p>Class Dismissal:</p> <ul style="list-style-type: none"> • Participants are free to leave unless they have chosen to stay for elective sessions. • Those staying behind for electives will receive further guidance on the additional topics or hands-on exercises available.
16:30-17:30	Elective: Tracing Transaction Flow, High Availability, Performance Counters, AMQP.	<p>In this elective session, participants will explore advanced administrative concepts using the DigitalOcean installation created during the previous day's elective module.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Tracing Transaction Flow: Enabling transaction tracing to log detailed transaction flows and timings, helping identify response-time bottlenecks. • Performance Counters: Utilizing various counters to monitor system throughput briefly and detect potential backlogs. • High Availability: Detailed guidance on configuring Jube in a horizontally distributed and highly available environment. • AMQP: Implementing AMQP message brokers, including a detailed discussion of integration and data engineering use cases.

		This session will provide participants with the knowledge and tools to optimize and manage Jube installations effectively in advanced scenarios.
--	--	--

Day 3: Permissions, Database, Reporting, Reprocessing and Extensibility

The day began with user access control, covering role-based permissions for secure UI access, followed by SQL database exploration of Jube's Postgres databases for reporting and archival data. Participants learned to extract JSONB data and write common queries.

The visualization module introduced Jube's reporting tools, teaching dashboard creation, parameterized SQL visualizations, and Kendo charting. Reprocessing demonstrated how to reevaluate historical data against new rules.

Technical sessions covered inline scripting (VB.Net/RPC) for custom integrations and synchronous RPC for external ML model scoring. The async messaging segment explored Jube's role in data pipelines via AMQP.

A workshop reinforced skills by building secure dashboards and managerial reports. The elective Redis deep dive explained cache optimization and transaction tracing for bottleneck analysis.

Outcome: Participants gained hands-on experience with Jube's security, reporting, and integration capabilities, from SQL queries to advanced data pipeline configurations.

Time	Topic	Description
09:00-09:30	Users, Roles and Permissions	<p>This session focuses on roles and permissions in Jube, which are essential for authentication and authorization within the user interface. Roles are collections of users and permissions, simplifying access management by assigning permissions to roles rather than individual users.</p> <p>Permissions grant access to specific resources, such as pages, and are allocated to roles, which are then assigned to users. For example, a customer service user may only need access to Case Management, Lists, Dictionaries, or Suppression, while an analyst might require access to rule and report creation tools.</p> <p>The session will cover:</p> <ul style="list-style-type: none"> • Creating roles and users. • A quick overview of all permissions available in Jube. • Focus on creating users for case management and discussing common roles and responsibility segmentation in typical deployments.

		This ensures efficient and secure access control tailored to users' responsibilities.
09:30-10:30	SQL Database Discovery	<p>In this session, participants will explore Jube's two databases: one for real-time processing (not covered here) and one for maintaining application and archive data. The focus is on understanding the database structure to support reporting, visualization, and advanced configurations.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Connecting to Postgres SQL. • Logical mapping of tables to their corresponding user interface components. • Introduction to the archive table and its JSONB storage format. • Introduction to the case tables and their JSONB storage format. • Common JSONB extraction statements. • Common reporting queries. • Counters and performance monitoring tables for integration with external tools like Splunk. • Administering entities via the dataset, including Sanction Loader configuration and Inline Scripts. <p>By the end of the session, participants will gain foundational SQL reporting skills and an understanding of advanced configurations, preparing them for the Visualization and Reporting module and enabling them to leverage Jube's data for analysis, monitoring, and customization.</p>
10:30-10:40	Break	Break
10:40-11:40	Visualization and Reporting	<p>This session introduces Jube's rudimentary reporting tool, designed to execute SQL queries and present results as grids or visualizations. The tool emphasizes quick report creation using basic SQL knowledge and the Kendo JQuery charting library, which uses JSON-like JavaScript fragments. Visualizations can be accessed via the Visualization Directory or the Case page, with two default reports available.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Recalling reports in the case management and reporting pages.

		<ul style="list-style-type: none"> • Dashboard layout concepts. • Creating parameterized visualizations. • Embedding SQL for visualizations. • Building charts and maps using Kendo tooling. • Creating HTML blocks with tokenization. <p>By the end of the session, participants will be able to use Jube as a centralized reporting tool, enabling data analysis and summarization directly within the case management pages.</p>
11:40-11:50	Break	Break
11:50-12:30	Reprocessing	<p>Reprocessing enables the retroactive re-evaluation of data stored in the Archive table, one entry at a time. This is particularly useful when new rules are created, and historical data needs to be reassessed.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Reprocessing Eligibility: Selecting which rules should be included in the reprocessing job. • Reprocessing Filtering: Filtering data to focus on specific subsets, as reprocessing large volumes of data can be resource intensive. • Reprocessing Instances: Creating a reprocessing job to define the time frame for reprocessing and monitor its progress. <p>The session will show how, or previously processed historical data, can be efficiently reassessed and aligned with new rules or requirements.</p>
12:30-13:30	Lunch	Lunch
13:30-14:00	Inline Scripts, including Remote Procedure Calls (RPC).	<p>This session delves into the embedding of VB.Net code for complex integration logic and introduces the concept of remote procedure calls (RPC) within Jube. It aims to showcase the platform's extensibility and its ability to handle advanced integration scenarios.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Registering an example inline script. • Understanding the structure of an inline script. • Reviewing a One-Time Password (OTP) example. • Reviewing a Remote Procedure Call (RPC) example. • Registering dependencies for scripts. • Calling an inline script from a model.

		<p>While the technical nature of the session may be challenging for non-technical participants, the goal is to familiarize all stakeholders with Jube's flexibility and its capability to support advanced customization and integration. This ensures that participants understand how Jube can be extended to meet complex business requirements.</p>
14:00-14:30	Scores via Synchronous Remote Procedure Calls (RPC).	<p>This session focuses on HTTP Adaptation, which allows users to integrate their own machine learning models into Jube by sending a JSON document in a POST request to a remote HTTP endpoint and receiving a single quantitative score in the JSON response. While Exhaustive simplifies machine learning integration, this method provides flexibility for custom model deployment.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Request and response payloads and protocol specifications. • Deployment of an R model using R Plumber, building on the earlier session introducing machine learning concepts with R. <p>By the end of the session, participants will understand how to extend Jube's qualitative functionality by integrating remotely available models, enabling advanced analytics and decision-making capabilities.</p>
14:30-14:40	Break	Break
14:40-15:30	Synchronous Messaging and Advanced Data Engineering Concepts with Jube	<p>This session shifts focus from synchronous RPC over HTTP to asynchronous methods of invocation and callback, highlighting Jube's role as a powerful analytics tool within data pipelines. By leveraging Jube's abstraction capabilities, participants will learn how to integrate its payload into data pipelines using common technologies.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Async query string flag and callback mechanisms. • AMQP queues for integration. • AMQP exchanges for streaming (publish/subscribe, header, or topic-based). <p>By the end of the session, participants will understand how to implement Jube in asynchronous workflows and</p>

		integrate it into data pipelines, enabling efficient data processing and analytics in real-world scenarios.
15:30-16:40	Workshop	<p>The workshop aims to enhance the case management system by creating a dashboard that provides a detailed profile of an account, including useful statistics. A custom role will be created to ensure sensitive features are hidden from analysts, maintaining security and access control.</p> <p>Additionally, management-focused reports will be developed in the visualization system to:</p> <ul style="list-style-type: none"> • Monitor productivity of teams or individuals. • Review audit logs for compliance and oversight. <p>This hands-on exercise will demonstrate how to extend Jube's functionality to support both operational and managerial needs, ensuring a balance between usability, security, and insightful reporting.</p>
16:40-16:50	Dismissed	<p>To conclude the day, we'll recap the key concepts and skills covered, ensuring participants have a solid understanding of the foundational elements of Jube and how to apply them in building a basic transaction monitoring system.</p> <p>Session Wrap-Up:</p> <ul style="list-style-type: none"> • Review of Key Takeaways: A summary of the day's learning objectives, including rule creation, data aggregation, and real-time decision-making. • Q&A Session: An opportunity for participants to ask questions and clarify any uncertainties. • Next Steps: An overview of what to expect on next day (if applicable), including deeper dives into case management and advanced rule implementation. <p>Class Dismissal:</p> <ul style="list-style-type: none"> • Participants are free to leave unless they have chosen to stay for elective sessions. • Those staying behind for electives will receive further guidance on the additional topics or hands-on exercises available.
16:50-17:50	Elective: Tracing the Transaction Flow, Bottleneck	This session introduces Redis, a real-time cache database used alongside Postgres SQL in Jube. Redis stores transaction data and counters, functioning

	Analysis and Cache	<p>differently from Postgres. The session will explain how data is indexed in Redis (which is more complex than simple key-value pairs) and highlight the importance of avoiding duplicate data in transactions. Participants will enable INFO-level tracing to generate detailed logs for analysing transaction processing and other threads.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Redis data model and paths to data. • Exploring the concept of a thread context. • Using the INFO and DEBUG flags in log4net for logging. • Utilizing the trace query string for detailed transaction tracing. • Interpreting logs and tracing transactions to identify bottlenecks. <p>By the end of the session, participants will understand Redis's role in Jube, how to trace transactions, and how to analyse logs for performance optimization.</p>
--	--------------------	--

Day 4 (Optional): Developer Workshop

The training provided a comprehensive deep dive into Jube's architecture and development processes. The morning sessions covered core software patterns, including database migrations, ORM, and dependency injection, followed by an exploration of Jube's .NET solution structure and its modular project design. Participants then learned practical skills for building, running, and debugging the application, including environment setup and log analysis.

A detailed walkthrough of Dependency Injection and service initialization revealed how Jube's components are instantiated and managed. The UI debugging session connected frontend actions to backend controllers, covering JWT authentication, Swagger API documentation, and real-time communication via SignalR.

The afternoon focused on background threads (case automation, sanctions loading, model training) and transaction processing, comparing synchronous and asynchronous flows. The day concluded with an in-depth look at the Exhaustive training algorithm, demonstrating model evolution, persistence, and real-time invocation.

Outcome: Developers gained hands-on experience with Jube's core systems, enabling them to extend, debug, and optimize the platform effectively. The training emphasized both architectural principles and practical implementation, preparing participants for advanced development work with Jube.

Time	Topic	Description
09:00-09:30	Software Patterns	This session provides an overview of Jube's software architecture and its defined patterns, which follow standardized practices for building robust and maintainable systems.

		<p>Key architectural concepts and considerations include:</p> <ul style="list-style-type: none"> • Migrations: Managing database schema changes over time. • ORM (Object-Relational Mapping): Simplifying database interactions using object-oriented principles. • Controllers: Handling application logic and routing. • Separation of Concerns: Ensuring modular and maintainable code by dividing functionality into distinct layers. • Tests: Implementing automated testing to ensure reliability and quality. • Validations: Enforcing data integrity and business rules. • Environment Variables: Managing configuration settings securely and flexibly. <p>The session will explain these patterns and their importance in Jube's development process, providing participants with a clear understanding of the architectural considerations and best practices when working with or extending the platform.</p>
09:30-10:00	C# Solution and Project Overview	<p>This session delves into the structure of Jube, a .Net solution composed of multiple projects, each addressing a specific concern. Participants will gain an understanding of the purpose of each project and how it aligns with the broader software patterns used in Jube.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Overview of individual projects: Explanation of each project's role and functionality. • Alignment with software patterns: How each project fits into the overall architecture, including patterns like Separation of Concerns, ORM, Controllers, and more. • Interdependencies: How the projects interact and depend on each other to form a cohesive system. <p>By the end of the session, participants will have a clear understanding of Jube's modular design, the responsibilities of each project, and how they collectively contribute to the platform's functionality and maintainability.</p>
10:00-10:30	Building and Running	<p>This session provides a step-by-step guide to building, running, and preparing Jube for debugging, equipping participants with the skills to effectively work with the platform during development. It covers setting up the development environment, compiling the .Net solution, and configuring environment variables and dependencies to run the application.</p>

		<p>Additionally, participants will learn how to prepare for debugging by setting up tools in the IDE (e.g., Visual Studio), enabling INFO and DEBUG logging levels for detailed insights, and interpreting logs to identify and resolve issues. By the end of the session, participants will be able to build, run, and debug Jube efficiently, ensuring a smooth development and troubleshooting process.</p>
10:30-10:40	Break	Break
10:40-11:10	Stepping and debugging the .NET Hosting Environment, Dependency Injection, and Migrations	<p>In this session, participants will start the Jube application and step through the startup process, focusing on the instantiation of services and their inclusion in Dependency Injection (DI).</p> <p>The session covers:</p> <ul style="list-style-type: none"> Starting the application and stepping through the initialization sequence. Observing the instantiation of services and their registration in Dependency Injection. Exploring how background threads are managed and controlled. Demonstrating how environment variables influence, The instantiation of services and the behaviour of background threads. Highlighting the flexibility and configurability of the system through environment variables. <p>By the end of the session, participants will gain a deeper understanding of Jube's startup process, dependency management, and the role of environment variables in controlling application behaviour.</p>
11:10-12:00	Stepping and Debugging the User Interface (Frontend and Backend)	<p>In this session, the developer will invoke several methods in the front end and explain the connection between the front end and backend controllers, including the availability of source files and their paths.</p> <p>The session covers:</p> <ul style="list-style-type: none"> JWT Authentication: Explanation of how authentication works and its integration with the API stack. Swagger/Open API: Demonstrating the comprehensive API documentation and resources available for developers. CRUD Methodology: Exploring the straightforward implementation of Create, Read, Update, and Delete operations in the application. SignalR/WebSocket's: Highlighting the occasional use of real-time communication for specific functionalities.

		By the end of the session, participants will understand the front-to-backend workflow, the role of JWT authentication, the utility of Swagger/Open API, and the application of both CRUD and real-time communication methods in Jube.
12:00-13:00	Stepping and Debugging Background Threads	<p>This session introduces the threads created in Jube and their purposes, focusing on the thread pool used for real-time transaction processing. It explains the transaction flow, asynchronous thread usage, and their convergence. The session explores the processing of each background thread.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Cases Automation • Archiver Persist • Model Invoke Asynchronous Threads • Activation Watcher Persist Threads • Cache Prune Server • Exhaustive Training • Sanctions Loader • Reprocessing • Search Key Cache <p>Participants will gain insight into how these threads support Jube's real-time processing and overall functionality.</p>
13:00-14:00	Lunch	Lunch
14:00-15:30	Stepping and Debugging Detailed Transaction Flow and Algorithms	<p>In this session, the transaction flow in Jube will be explained, covering the use of the thread pool, asynchronous IO, and convergence. With INFO-level logging enabled, a transaction will be stepped through during an HTTP invocation, with each key part of the processing discussed in detail. After completing the lifecycle of a synchronous transaction, the process will be repeated for an asynchronous transaction, highlighting the differences in invocation methods while maintaining a similar flow.</p> <p>The session covers:</p> <ul style="list-style-type: none"> • Thread pool usage in transaction processing. • Asynchronous IO and its role in efficient processing. • Convergence of threads to complete transactions. • Stepping through a transaction with INFO-level logging for detailed insights. • Comparing synchronous and asynchronous transaction flows. <p>By the end of the session, participants will understand the transaction lifecycle, the role of asynchronous IO, and how Jube processes transactions efficiently.</p>

15:30-15:40	Break	Break
15:40-17:20	Stepping and Debugging Exhaustive Model	<p>In this session, the Exhaustive training algorithm will be explored, with a step-by-step walkthrough of training a model to demonstrate its evolution and improvement over trials. Participants will learn how the model persisted for future use and invoked in real-time to make predictions or decisions. The session will cover the training process, the persistence of the model, and its integration into real-time transaction processing, providing a comprehensive understanding of how models are developed, stored, and applied in Jube for efficient and accurate decision-making.</p>
17:20-17:30	Dismissed	<p>To conclude the day, we'll recap the key concepts and skills covered, ensuring participants have a solid understanding of the foundational elements of Jube and how to apply them in building a basic transaction monitoring system.</p> <p>Session Wrap-Up:</p> <ul style="list-style-type: none"> • Review of Key Takeaways: A summary of the day's learning objectives, including rule creation, data aggregation, and real-time decision-making. • Q&A Session: An opportunity for participants to ask questions and clarify any uncertainties. • Next Steps: An overview of what to expect on next day (if applicable), including deeper dives into case management and advanced rule implementation. <p>Class Dismissal:</p> <ul style="list-style-type: none"> • Participants are free to leave unless they have chosen to stay for elective sessions. • Those staying behind for electives will receive further guidance on the additional topics or hands-on exercises available.



Resources

<https://jube.io/JubeAMLMonitoringComplianceGuidance.pdf>

<https://jube-home.github.io/aml-transaction-monitoring/>

Conclusion

In conclusion, Jube's comprehensive training program is designed to empower professionals with the knowledge and skills necessary to effectively implement, manage, and optimize Jube within their organizations. By offering a blend of core concepts, advanced features, and elective modules, the program ensures participants gain hands-on experience and practical insights tailored to their roles—whether they are developers, compliance officers, fraud prevention specialists, or system administrators.

The structured agenda, spanning 3 to 4 days, provides a deep dive into Jube's capabilities, from AI-driven solutions and sanctions fuzzy matching to performance optimization and debugging. With optional developer training, organizations can also ensure their technical teams are well-equipped to maintain and troubleshoot the system for long-term business continuity.

By attending this program, participants will not only accelerate the adoption of Jube but also facilitate their ability to leverage its advanced features for fraud detection, compliance, and financial crime prevention. Ultimately, this training equips organizations to maximize the value of Jube, ensuring scalability, security, and operational efficiency in an ever-evolving landscape.

For those ready to take the next step, Jube's training program is an invaluable investment in building expertise and driving successful implementation.