Version 2.1

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

This document outlines comprehensive remote training for implementing Jube, an open-source Anti-Money Laundering (AML) transaction monitoring system. Jube uses Machine Learning to enable real-time transaction and event monitoring, helping organizations meet regulatory compliance requirements through its AML Monitoring Compliance Guidance.

JUBE

JUBE

Amendments

Date	Author	Version	Description
14th March 2025	Richard Churchman	2	Updated for version 2 of the training materials.
15th April 2025	Richard Churchman	2.1	Updated to reflect remote only.

JUBE

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.

This document is an expanded version of the overview available via https://www.jube.io/training, whereby each agenda item is expanded upon.

Program Outcomes

By the end of the remote training, participants will:

- Confidently implement and manage Jube within their organization.
- Optimize Jube for performance, scalability, and security.
- Accelerate adoption through hands-on experience and best practices.
- Troubleshoot and debug the system effectively (for developers).

Why Attend?

- Gain hands-on experience with Jube's features and tools.
- Learn best practices for optimizing performance, scalability, and security.
- Accelerate adoption and ensure successful implementation.
- Develop skills to troubleshoot, debug, and manage Jube effectively.

Who Should Attend?

The remote training program is ideal for professionals involved in implementing, managing, and optimizing Jube within their organizations. Key roles include:

- Developers:
    - Responsible for integrating, customizing, and maintaining Jube.
    - Need to understand the codebase, debug issues, and ensure business continuity.
- Heads of Fraud:
    - Leaders overseeing fraud detection and prevention strategies.

    - Will leverage Jube's advanced features like Machine Learning training, suppression rules, and sanctions fuzzy matching.
- Heads of Compliance:
    - Compliance officers ensuring regulatory adherence.
    - Will use Jube for sanctions screening, case management, and reporting.
- Financial Crime Prevention Professionals:
    - Focused on combating money laundering, fraud, and sanctions evasion.
    - Will benefit from Jube's AI-driven solutions and performance optimization.
- System Administrators:
    - Responsible for Jube's administration, configuration, and maintenance.
    - Will manage performance monitoring, caching, and high availability.
- Data Analysts and Reporting Specialists:
    - Involved in data discovery, visualization, and reporting.
    - Will utilize Jube's SQL database discovery and reporting tools.
- IT and Security Professionals:
    - Ensure the security, scalability, and performance of Jube implementations.
    - Manage environment variables, log configurations, and authentication.

Remote Training Agenda

The remote training spans 3 to 4 days, with each day lasting 6 to 8 hours, depending on elective modules. Elective modules provide in-depth remote 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

- 09:00 – 09:00: AML Monitoring Compliance Guidance Overview:
  - Introduction to Jube's AML compliance framework, aligned with FATF and Wolfsberg Principles.
  - Focus on transaction monitoring, risk-based approaches, and leveraging Jube's features for regulatory compliance.
- 09:00-09:10: Break.
- 09:10-09:30: Introduction to Jube and Key User Interface Concepts:
  - Overview of Jube's functionality, including authentication, navigation, parent/child objects, and rule/code builders.
  - Hands-on practice with the platform.
- 09:20-09:40: Models and HTTP Messaging:
  - Introduction to Jube's flexible data models and HTTP API integration.
  - Hands-on configuration of AML-specific models and transaction flows.
- 09:40-11:00: Request XPath Payload:
  - Focus on extracting and processing data using XPath, data types, and cache storage.
  - Finalizing integration setup for compliance-focused implementation.
- 11:00-11:10: Break.
- 11:10-11:30: Inline Functions:
  - Using VB.net syntax to combine data (e.g., First Name + Last Name) for real-time processing like sanctions screening.
- 11:30-11:50: Gateway Rules:
  - Introduction to pre-filtering transactions using Gateway Rules to optimize system performance.
  - Hands-on rule creation using visual and code-based approaches.
- 11:50-12:30: Abstraction Rules:
  - Aggregating transaction data to generate insights and support decision-making.
  - Covers abstraction functions, rule building, and offsets.
- 12:30-12:50: Abstraction Calculations:
  - Creating ratios and metrics from Abstraction Rules for machine learning and activation rules.
- 12:50-13:50: Lunch.
- 13:50-14:40: Activation Rules:

    - Configuring actionable responses like transaction declines, case creation, and notifications.
    - Focus on integrating components for actionable outcomes.
- 14:40-15:00: Lists and Dictionaries:
    - Managing data lists and key-value pairs for streamlined rule creation and data enrichment.
- 15:00-15:10: Break.
- 15:10-16:20: Workshop:
    - Collaborative exercise to design and implement a basic transaction monitoring system.
    - Hands-on creation of Activation Rules for real-time decision-making.
- 16:20-16:30: Session Wrap-Up and Dismissal:
    - Recap of key concepts, Q&A, and preview of Day 2 topics.
- 16:30-17:30: Elective: Architecture and Caching, Environment Variables, Installation, and Log Configuration:
    - Deep dive into Jube's architecture, installation on Digital Ocean servers, and log management.
    - Hands-on deployment using Docker Compose.

Time	Topic	Description
09:00 – 09:00	AML Monitoring Compliance Guidance Overview	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. 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. 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.

		This session aims to offer an overview of the guidance and establish the foundation for the features and capabilities that will be introduced.
09:00-09:10	Break	Break
09:10-09:30	Introduction to Jube and Key User Interface Concepts	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: • 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. During the session, participants will log into their respective tenant environments on the server to explore these features through hands-on practice. This approach ensures users gain practical experience with Jube's interface and functionality.
09:20-09:40	Models and HTTP Messaging	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.

		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. The session will cover the following key components: • 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. 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.
09:40-11:00	Request XPath Payload	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: • 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 processing (e.g., cannot apply "greater than" logic to text). • 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. 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.
11:00-11:10	Break	Break
11:10-11:30	Inline Functions	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.
11:30-11:50	Gateway Rules	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. 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, 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	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. The session covers: • 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. 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.
12:30-12:50	Abstraction Calculations	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. The session covers: • Demonstrate how to create common ratios using Abstraction Calculations. • Highlight the implications of these ratios in Activation Rules and Machine Learning, showcasing their role in enhancing analytical capabilities and decision-making processes. 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.
12:50-13:50	Lunch	Lunch

13:50-14:40	Activation Rules	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. Activation Rules are designed to take actionable steps, such as: • 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. 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.
14:40-15:00	Lists and Dictionaries	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. Dictionaries function similarly to lists but with an added layer of functionality. Instead of simply matching a 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.

		Together, Lists and Dictionaries help reduce the volume of activation rules and enhance their elegance by simplifying data management and integration. The session covers: • 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. By the end of the session, participants will understand how to effectively use lists and dictionaries to optimize rule sets and improve system efficiency.
15:00-15:10	Break	Break
15:10-16:20	Workshop	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. The session covers: • 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. 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.
16:20-16:30	Dismissed	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. Session Wrap-Up: • 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. Class Dismissal: • 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, Installation and Log Configuration.	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. The session covers: • 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. 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.

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

- 09:00-09:20: Suppression:
  - Tools to manage automated actions: Suppression Keys, Total Suppression, and Suppression by Rule to pause or stop actions for specific accounts or conditions.
- 09:20-09:40: Sanctions Fuzzy Matching:
  - Introduction to fuzzy matching using the Levenshtein Distance Algorithm for sanctions screening.
  - Covers real-time and batch name matching against sanctions lists.
- 09:40-10:00: Time to Live (TTL) Counters:
  - Efficient method for creating model memory without storing full transaction data.
  - Covers creating, incrementing, and using TTL Counters in Activation Rules.
- 10:00-10:10: Break.
- 10:10-11:00: Introduction to Machine Learning:
  - High-level overview of supervised learning concepts: linear regression, logistic regression, neural networks, feature engineering, and overfitting.
  - Brief introduction to unsupervised learning.
- 11:00-11:10: Break.
- 11:10-11:50: Exhaustive Machine Learning Training:
  - Introduction to Jube's embedded machine learning algorithm, Exhaustive, for simplifying model training and recall.
  - Covers topology evolution, training statistics, and real-time recall.
- 11:50-12:00: Break.

- 12:00-12:20: Activation Watcher:
  - Real-time monitoring of Response Elevations triggered by Activation Rules.
  - Streaming activations, replaying past activations, and real-time transaction monitoring.
- 12:20-12:40: Tags:
  - Collecting feedback to minimize errors and power advanced analytics.
  - Configuring tags and invoking tagging via remote HTTP endpoints.
- 12:40-13:40: Lunch.
- 13:40-15:00: Case Workflows:
  - Comprehensive guide to creating and managing case workflows for AML and fraud prevention.
  - Covers case statuses, filters, activation rules, case operations, notes, dispositions, macros, forms, and status elevations.
- 15:00-15:10: Break.
- 15:10-16:20: Workshop:
  - Hands-on implementation of a case management system to support AML compliance efforts.
  - Focus on monitoring, investigation, and reporting workflows.
- 16:20-16:30: Session Wrap-Up and Dismissal:
  - Recap of key concepts, Q&A, and next steps.
- 16:30-17:30: Elective: Tracing Transaction Flow, High Availability, Performance Counters, AMQP:
  - Tracing Transaction Flow: Identifying bottlenecks in transaction processing.
  - Performance Counters: Monitoring system throughput.
  - High Availability: Configuring Jube for distributed environments.
  - AMQP: Implementing message brokers for integration and data engineering.

Time	Topic	Description
09:00-09:20	Suppression	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. This session will cover the following approaches to managing such cases: • 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	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. The session covers: • 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	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 Counters are incremented when an Activation Rule is triggered and are designed to expire after a specified period, after which they automatically decrement. 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. The session covers: • 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	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. The session covers: • 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. • 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. 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.
11:00-11:10	Break	Break
11:10-11:50	Exhaustive Machine Learning Training	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. The session covers: • 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. This session will provide a comprehensive understanding of how Exhaustive streamlines machine learning within Jube, making it more accessible and efficient.
11:50-12:00	Break	Break
12:00-12:20	Activation Watcher	The Activation Watcher serves as a real-time ticker for Response Elevations, triggered when an Activation Rule is configured to send messages to it. This session covers: • Setting up rules to send activations to the Activation Watcher. • Streaming activations in real time. • Replaying past activations. It offers an engaging and practical insight into real-time transaction monitoring within Jube.
12:20:12:40	Tags	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. This session covers: • 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). *Note:* If future updates enable tagging within the user interface, it will be integrated into the case management system.
12:40-13:40	Lunch	Lunch
13:40-15:00	Case Workflows	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. The session covers: • Case Workflow: How to create and configure a case workflow. • Case Status: An introduction to case statuses and status elevations. • 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. This session will provide a comprehensive overview of case management functionality, equipping participants with the knowledge to effectively manage and streamline workflows.
15:00-15:10	Break	Break
15:10-16:20	Workshop	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). Using the Anti-Money Laundering (AML) Monitoring Compliance Guidance, the class will now proceed to implement a case management system designed to support broader compliance efforts. This system will enable effective monitoring, investigation, and reporting in line with regulatory requirements.
16:20-16:30	Dismissed	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. Session Wrap-Up: • 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. Class Dismissal:

		• 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.	In this elective session, participants will explore advanced administrative concepts using the DigitalOcean installation created during the previous day's elective module. The session covers: • 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

- 09:00-09:30: Users, Roles, and Permissions:
  - Focus on authentication and authorization in Jube.
  - Covers creating roles, assigning permissions, and managing user access for different responsibilities (e.g., customer service, analysts).
- 09:30-10:30: SQL Database Discovery:
  - Introduction to Jube's Postgres SQL database for application and archive data.
  - Covers logical table mapping, JSONB storage, common queries, and performance monitoring tables.
  - Prepares participants for reporting and advanced configurations.
- 10:30-10:40: Break.
- 10:40-11:40: Visualization and Reporting:
  - Introduction to Jube's reporting tool for creating grids and visualizations using SQL queries.

    - Covers dashboard layouts, parameterized visualizations, and Kendo charting library.
    - Enables participants to create and embed reports directly in case management pages.
- 11:40-11:50: Break.
- 11:50-12:30: Reprocessing:
    - Explains how to reprocess historical data from the Archive table to apply new rules.
    - Covers eligibility, filtering, and monitoring reprocessing jobs.
- 12:30-13:30: Lunch.
- 13:30-14:00: Inline Scripts and Remote Procedure Calls (RPC):
    - Introduction to embedding VB.Net code for complex integrations and RPCs.
    - Covers script registration, structure, and examples like One-Time Password (OTP) and RPC.
    - Highlights Jube's flexibility for advanced customization.
- 14:00-14:30: Scores via Synchronous RPC:
    - Focus on HTTP Adaptation for integrating external machine learning models into Jube.
    - Covers request/response payloads and deploying models using R Plumber.
- 14:30-14:40: Break.
- 14:40-15:30: Synchronous Messaging and Advanced Data Engineering:
    - Explores asynchronous workflows using AMQP and Kafka for data pipeline integration.
    - Covers callback mechanisms, queues, exchanges, and streaming.
- 15:30-16:40: Workshop:
    - Hands-on exercise to enhance the case management system with a custom dashboard and management reports.
    - Focus on account profiling, team productivity monitoring, and audit log reviews.
- 16:40-16:50: Session Wrap-Up and Dismissal:
    - Recap of key concepts, Q&A, and next steps.
- 16:50-17:50: Elective: Tracing Transaction Flow, Bottleneck Analysis, and Cache:
    - Deep dive into Redis, Jube's real-time cache database.
    - Covers data indexing, transaction tracing, and log analysis for performance optimization.

Time	Topic	Description
09:00-09:30	Users, Roles and Permissions	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. 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.

		The session will cover: • 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	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. The session covers: • 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. 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.
10:30-10:40	Break	Break
10:40-11:40	Visualization and Reporting	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. The session covers: • Recalling reports in the case management and reporting pages. • Dashboard layout concepts. • Creating parameterized visualizations. • Embedding SQL for visualizations. • Building charts and maps using Kendo tooling. • Creating HTML blocks with tokenization. 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.
11:40-11:50	Break	Break
11:50-12:30	Reprocessing	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. The session covers: • 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. The session will show how, or previously processed historical data, can be efficiently reassessed and aligned with new rules or requirements.
12:30-13:30	Lunch	Lunch
13:30-14:00	Inline Scripts, including Remote Procedure Calls (RPC).	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.

		The session covers: • 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. 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.
14:00-14:30	Scores via Synchronous Remote Procedure Calls (RPC).	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. The session covers: • 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. 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.
14:30-14:40	Break	Break
14:40-15:30	Synchronous Messaging and Advanced Data Engineering Concepts with Jube	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. The session covers:

		• Async query string flag and callback mechanisms. • AMQP queues for integration. • AMQP exchanges for streaming (publish/subscribe, header, or topic-based). • Kafka interfaces for append-only log topic-based publish/subscribe. By the end of the session, participants will understand how to implement Jube in asynchronous workflows and integrate it into data pipelines, enabling efficient data processing and analytics in real-world scenarios.
15:30-16:40	Workshop	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. Additionally, management-focused reports will be developed in the visualization system to: • Monitor productivity of teams or individuals. • Review audit logs for compliance and oversight. 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.
16:40-16:50	Dismissed	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. Session Wrap-Up: • 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.

		Class Dismissal: • 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 Analysis and Cache	This session introduces Redis, a real-time cache database used alongside Postgres SQL in Jube. Redis stores transaction data and counters, functioning 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. The session covers: • 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. 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.

Day 4 (Optional): Developer Workshop

- 09:00-09:30: Software Patterns:
  - Overview of Jube's architecture and standardized patterns.
  - Key concepts: Migrations, ORM, Controllers, Separation of Concerns, Tests, Validations, and Environment Variables.
  - Focus on building robust, maintainable systems.
- 09:30-10:00: C# Solution and Project Overview:
  - Structure of Jube's .Net solution and its multiple projects.
  - Explanation of each project's role and alignment with software patterns.
  - Understanding interdependencies and modular design.
- 10:00-10:30: Building and Running:
  - Step-by-step guide to building, running, and debugging Jube.
  - Setting up the development environment, compiling the solution, and configuring environment variables.

    - Preparing for debugging with IDE tools and interpreting logs.
- 10:30-10:40: Break.
- 10:40-11:10: Stepping and Debugging .NET Hosting, Dependency Injection, and Migrations:
    - Exploring Jube's startup process, service instantiation, and Dependency Injection.
    - Understanding the role of environment variables in service behaviour and background thread management.
- 11:10-12:00: Stepping and Debugging the User Interface (Frontend and Backend):
    - Connection between frontend and backend controllers.
    - Key topics: JWT Authentication, Swagger/Open API, CRUD Methodology, and SignalR/WebSocket's.
    - Understanding front-to-backend workflows and real-time communication.
- 12:00-13:00: Stepping and Debugging Background Threads:
    - Introduction to Jube's background threads and their purposes.
    - Focus on real-time transaction processing and thread pool usage.
    - Key threads: Cases Automation, Archiver Persist, Model Invoke, Activation Watcher, Cache Prune, Exhaustive Training, Sanctions Loader, Reprocessing, and Search Key Cache.
- 13:00-14:00: Lunch.
- 14:00-15:30: Stepping and Debugging Detailed Transaction Flow and Algorithms:
    - Explanation of transaction flow, thread pool usage, and asynchronous IO.
    - Stepping through synchronous and asynchronous transactions with INFO-level logging.
    - Understanding the lifecycle and convergence of transactions.
- 15:30-15:40: Break.
- 15:40-17:10: Stepping and Debugging Exhaustive Model Training:
    - Step-by-step walkthrough of the Exhaustive training algorithm.
    - Model evolution, persistence, and integration into real-time transaction processing.
    - Understanding how models are developed, stored, and applied for decision-making.

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. Key architectural concepts and considerations include: • 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. 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.
09:30-10:00	C# Solution and Project Overview	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. The session covers: • 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. 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.
10:00-10:30	Building and Running	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. 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.
10:30-10:40	Break	Break

10:40-11:10	Stepping and debugging the .NET Hosting Environment, Dependency Injection, and Migrations	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). The session covers: • 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. 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.
11:10-12:00	Stepping and Debugging the User Interface (Frontend and Backend)	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. The session covers: • 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	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.

		The session covers: • Cases Automation • Archiver Persist • Model Invoke Asynchronous Threads • Activation Watcher Persist Threads • Cache Prune Server • Exhaustive Training • Sanctions Loader • Reprocessing • Search Key Cache Participants will gain insight into how these threads support Jube's real-time processing and overall functionality.
13:00-14:00	Lunch	Lunch
14:00-15:30	Stepping and Debugging Detailed Transaction Flow and Algorithms	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. The session covers: • 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. By the end of the session, participants will understand the transaction lifecycle, the role of asynchronous IO, and how Jube processes transactions efficiently.
15:30-15:40	Break	Break
15:40-17:20	Stepping and Debugging Exhaustive Model	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.
17:20-17:30	Dismissed	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. Session Wrap-Up: • 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. Class Dismissal: • 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.

Conclusion

In conclusion, Jube's comprehensive remote 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 remote 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 remote 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 remote training program is an invaluable investment in building expertise and driving successful implementation.