IBM Safer Payments

- PCI PA-DSS certified data protection
- In-memory transaction history (1 year typical) for fast rules and model building and testing
- Continuous, performance monitoring with user configured dashboards
- Nation-scale throughput of thousands of transactions per second
- Supports one or hundreds of concurrent tenants isolated or with controlled sharing
- Rich alert and case management with customizable workflows
- Tools for novice through expert model builders

Fraudsters are outwitting obsolete fraud prevention systems, stealing more from payment channels and eroding confidence while inhibiting business growth and innovation

It’s time for a better approach

IBM Safer Payments puts modern machine learning in the hands of on-site fraud management teams, which significantly improves their effectiveness and ability to stop more fraud with fewer false alerts.

This solution is designed to help you rapidly recognize and stop new fraud attacks across diverse channel segments (credit issuing and acquiring, immediate and alternative payments, processors, etc.) and use cases (card, non-card, cross-channel, online, etc). It protects multiple payment channels sharing data between them and monitors thousands of payments per second.

IBM Safer Payments provides the ability to build, test, validate and deploy machine-learning models in days. Build profile variables with point-and-click ease and deploy new rules and models without interrupting monitoring. You can leverage look-back profile variables that adapt for relevance to the current transaction, for example, “What is the frequency, average amount, maximum amount, most frequent amount and mean amount of transactions at counterparties in the same post code as the current transaction?”

Regain the autonomy to modify models and rules as needed without relying on a vendor to make the changes. The solution delivers 99.999% availability for typical installations.

Key benefits of IBM Safer Payments

IBM Safer Payments helps give state-of-the-art machine learning to on-site fraud management for on-demand rapid responses. No more layering ad hoc rules atop failing black-box models to address new fraud attacks.

You can enable rapid updates to primary models, with full testing, governance validation and high-speed deployment. With IBM Safer Payments, use your preferred tools to build neural networks, random forests, decision trees and regressions. Combine models of different types into ensembles to leverage the best of each modeling technology. Improve detection by combining short-view models focused on recent fraud attacks for lowest false-positives, with long-view models trained on behavior regularities for high detection rates.

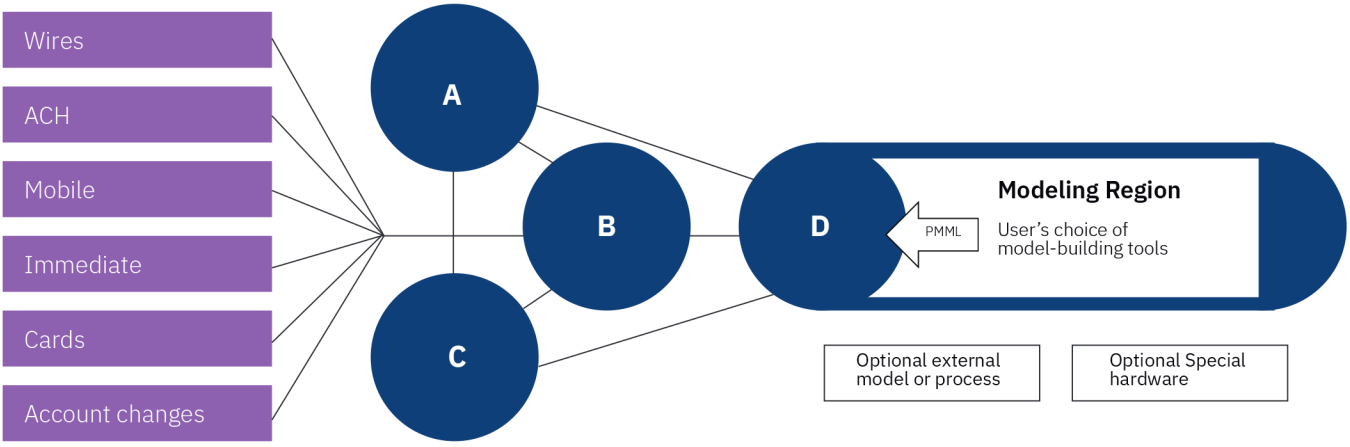
Use in-memory historical data and point-and-click feature definitions, to speed model testing and deployment. Skilled users may use their choice of model-building tools with CRAN-PMML model imports for Safer Payments deployment.

A Typical, Statistical Modeling Configuration for nation–level volume using IBM Safer Payments

Multiple payment channels, account information

3 Interconnected, Commodity Intel Linux machines at 2 or more sites

Extra instance with extra resources to support user's choice of modeling tools



Three identical Intel Linux® virtual or physical servers host duplicate Safer Payments instances. All instances are sized to handle peak volume alone, and are updated and ready to process transactions. Instance D is hosted on a machine with additional resources (especially disk space) to support the user's modeling tools and intermediate data files. All four instances have duplicate, continuously updated, in-RAM copies of the last year (configurable) of monitored transactions.

Models constructed on and imported into instance D that are promoted to production are propagated to the other instances for real-time, triply redundant production operation. A switch to new models or rules is accomplished by switching production message flow to different instances so that no interruption of full-speed production monitoring is needed.