Technical Project Walk Through

Yi Zhang

Overview

- Proactively created a new iteration of Unit21's core Transaction Monitoring Engine, increasing its performance by 50%, and functionality 10-fold
- Directly synthesized requirements from working with customers
- Built a prototype in 2 weeks, then turned it into the full blown product, and built a team of ~15 around it

Background

- Unit21 makes vertically integrated software that performs transaction monitoring for compliance (e.g. anti-money laundering), and later on, fraud, use cases
- Customers ingest transactions into U21 via an API
- Customers manage a set of customizable Rules that are meant to detect suspicious activities
- Rules create alerts, which go into operational queues, to be investigated by operations specialists

Example Rule

Every day starting from 2026-03-01 at 3:00am

Look back 30 days

Alert if a user

- is based in the US

and had a total of >$100,000 of transactions that are:

- Individually between $100 and $500 each
- Originated from A, B, C countries

Benefits

- Supported a fixed set of filters on individual users & transactions
- Supported a fixed set of aggregation logic (total volume, pass through, etc.)
- Supported configurable window size & execution frequency
- Easy to understand for a simple rule as an English sentence
- Provided clear axis for extension - simply add new templates

Problems

- For moderately complex logic, an English description is very difficult to read
- Require new templates, hence engineering cycle, even for small logic changes
- As a result, number of templates quickly go out of hand, and customers don’t understand exactly what each one does or how to pick

The Realization

Facts + Condition

- A Rule is essentially a number of Facts (e.g. aggregations) stringed together into a Boolean formula
- In the previous example, the “total amount (with filters)” is a fact, and it’s >$100,000 is the condition
- They are **independent** of each other

Dynamic Model Builder Example

Facts:

user_location_us:

- User is based in the US

last_30d_risky_transaction_volume:

- transactions between $100 and $500 each
- transactions originated from A, B, C countries

Condition:

user_location_us is True **AND** last_30d_risky_transaction_volume > $100,000

Dynamic Model Builder

- Allow customers to pre-define & test out & share facts with names & descriptions
- Allow customers to write actual mathematical formulas to string facts and constants together
- Built a DSL to express rules in such a way that is 10x more flexible, which then gets compiled to a canonical Intermediate Representation (IR)
- On the backend, the rule execution engine takes the IR and generates an execution plan

Benefits

- Greatly simplified the rule's logic when inspected visually
- Much more flexible in terms of the kinds of logic it supports (e.g. facts over different time frames, arbitrary mathematical operations over facts)
- Over time, with more existing facts saved in the inventory, creating new Rules had become trivial (writing out one line conditions)
- With this added layer of abstraction, the execution was much further optimized (e.g. generated more efficient SQL, allowed streaming pre-compute & caching of results)

Rollout

- It was an immediate hit even during prototyping stage with Sales & Implementation teams, as well as the CEO
- Conducted & Supported many initiatives to accelerate its rollout:
  - Ran training sessions & wrote docs, made videos etc.
  - Wrote scripts that automatically convert prior format to the new one
  - Allowed users to compare before v. after and one-click convert
  - Created pre-built templates to help new customers onboard faster
- Made a steady & smooth transition and now 95% of executions are done through the new execution engine

Live doc today