AML BASE LLC

PROFESSIONAL FINANCIAL CRIME TRAINING | 94+ COUNTRIES

Finding What the Rules Cannot See

Predictive and Machine-Learning Models for Undetected FC / ML / TF / PF at Scale

A Technical Guidebook for a US Tier 1 Bank

PROBLEM	Suspicious activity that no transaction monitoring rule generates an alert for
APPROACH	Unsupervised anomaly detection · graph neural networks · sequence models · weak supervision · document and media NLP
SCALE	~48M transactions/day · ~28M parties · 180M-node / 2.4B-edge entity graph · 7 years of history
GOVERNANCE	SR 11-7 / OCC 2011-12 model risk · 2018 Joint Statement on Innovative Efforts · AMLA 2020
CLASSIFICATION	Internal Use Only — Financial Intelligence Unit / AI Risk
VERSION	1.0 — July 2026

Prepared by AML Base LLC. Illustrative institution: 'Meridian National Bank, N.A.' — a hypothetical US Tier 1 bank.

How to Read This Guidebook

This is a technical guidebook, not a policy. It assumes the reader already has a functioning rules-based transaction monitoring system — at Meridian National Bank (MNB), our hypothetical US Tier 1 institution, that system is NICE Actimize SAM running 24 scenarios — and asks a harder question: **what is that system structurally incapable of seeing, and what can be built to see it?**

The answer is not 'a better rule'. A rule encodes a typology someone already knew about, tested against a threshold someone already chose. It cannot, by construction, detect a typology nobody has written down, a pattern that lives below every threshold, or a structure that only exists across entities the rule engine examines one at a time. Those three gaps are the subject of this book.

The intended reader is a quantitative analyst, data scientist, AML technologist or model validator at a large bank. Financial-crime domain knowledge is assumed; machine-learning knowledge is developed as we go. Code is illustrative PySpark / PyTorch and is written to be read, not copy-pasted.

> **A note on what this is not**
> This book does not describe a system that decides whether a customer is a criminal. It describes a system that decides *where a human being should look next*. Every model in it is a prioritisation model. The consequence of a model error is a misallocated hour of investigator time — never an adverse action against a customer. That constraint is not a limitation of the design; it *is* the design, and Chapter 11 explains why any architecture that relaxes it should be rejected outright.

Notation and Abbreviations

Term	Meaning	Term	Meaning
AE / VAE	Autoencoder / Variational Autoencoder	LOF	Local Outlier Factor
AIS	Automatic Identification System (vessel tracking)	MRM	Model Risk Management (SR 11-7 / OCC 2011-12)
AMLA	Anti-Money Laundering Act of 2020	PF	Proliferation Financing
AUC / AUPRC	Area Under the ROC / Precision-Recall Curve	PSI	Population Stability Index
BOCPD	Bayesian Online Change-Point Detection	PU learning	Positive-Unlabelled learning
BoL	Bill of Lading	R-GCN	Relational Graph Convolutional Network
CUSUM	Cumulative Sum (change detection)	SHAP	SHapley Additive exPlanations
EWRA	Enterprise-Wide AML Risk Assessment	SAM	Suspicious Activity Monitoring (the incumbent rules engine)
FC	Financial Crime (the umbrella term)	TF	Terrorist Financing
GAT / GCN	Graph Attention Network / Graph Convolutional Network	TM	Transaction Monitoring (rules-based)
GNN	Graph Neural Network	UBO	Ultimate Beneficial Owner
HS code	Harmonised System commodity code	VASP	Virtual Asset Service Provider
iForest	Isolation Forest		

Contents

1. The Detection Gap

1.1 What a Rule Can and Cannot Be

A transaction monitoring rule is a conditional statement over an aggregate: *if the sum of cash credits to a party over seven days exceeds T across at least N transactions, alert*. It has three properties, and all three are load-bearing weaknesses.

Property	Consequence	What it makes invisible
It encodes a known typology. Somebody had to have seen the behaviour before, understood it, and written it down.	The rule set is a museum of yesterday's laundering.	Any typology that is new, or that the bank has simply never encountered — which, for a novel scheme, is by definition every scheme at the moment it matters most.
It has a threshold. A number was chosen, and it partitions the world into 'alert' and 'silence'.	There is always a region immediately below the line where the rule is guaranteed to be silent — and a sophisticated actor knows roughly where that line is.	Sub-threshold activity; value split across time, accounts, products or channels until every fragment is beneath every line.
It is entity-centric. It evaluates one focal entity at a time, over one product at a time, over one window at a time.	It cannot express a proposition about a *structure*.	Mule rings, layered corporate ownership, funnel networks, PF procurement chains, and the general class of 'forty individually-unremarkable customers who are in fact one organisation'.

The three quadrants a rules-based TM system structurally cannot reach

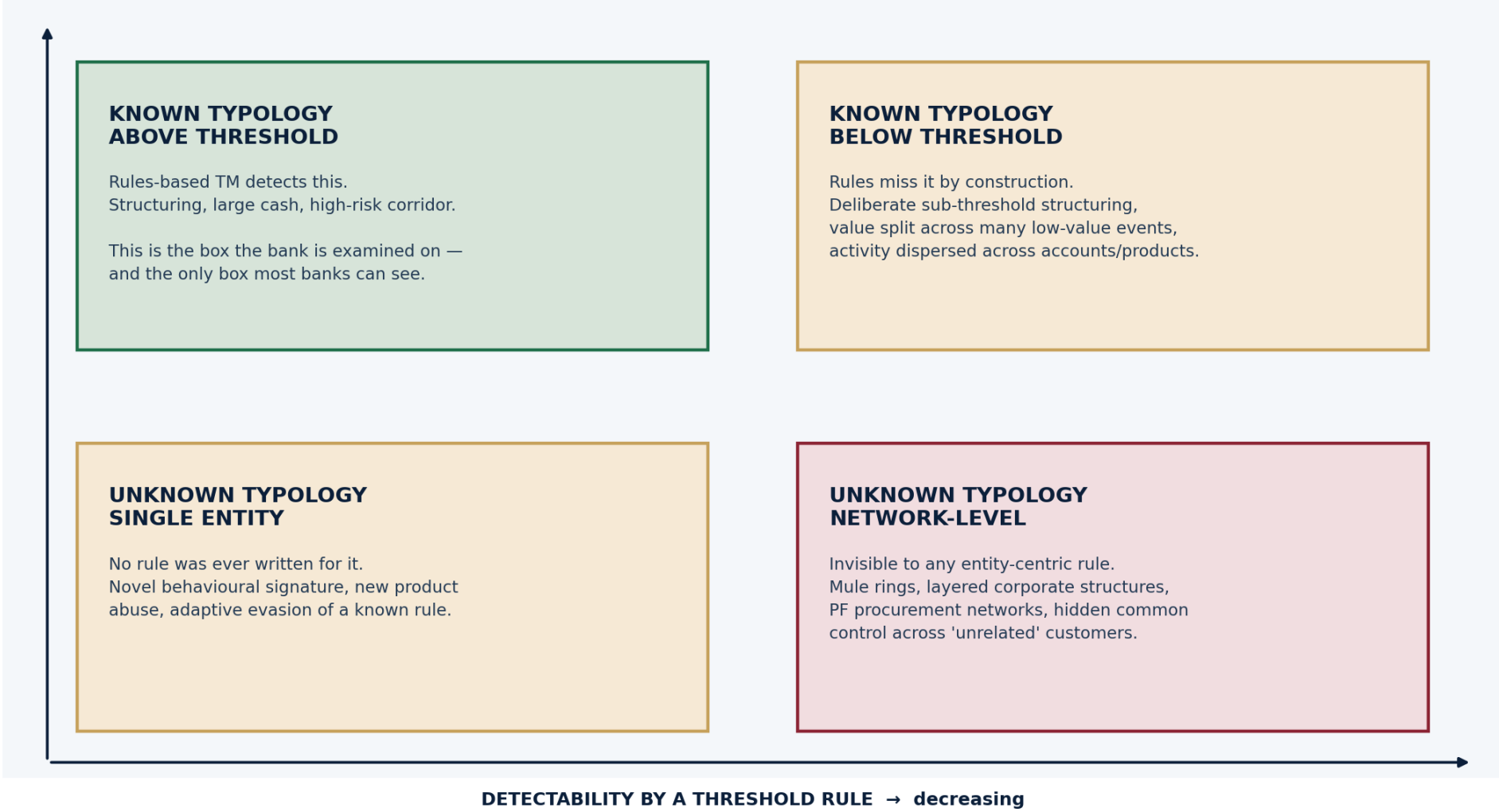

Figure 1 — The detection gap. A rules-based system operates in one quadrant. The other three are not edge cases; at a Tier 1 bank they are where the organised money is.

1.2 The Threshold Is the Signal

The deepest irony of threshold-based detection is that the threshold itself creates a detectable artefact — and the rule cannot see it. A population of honest cash deposits has a smooth distribution. A population containing deliberate structuring has an excess mass immediately below the reporting threshold and a corresponding deficit immediately above it. That discontinuity is statistically loud. It is invisible to a rule set at the threshold, and obvious to any model that looks at the *shape* of the distribution rather than at individual transactions.

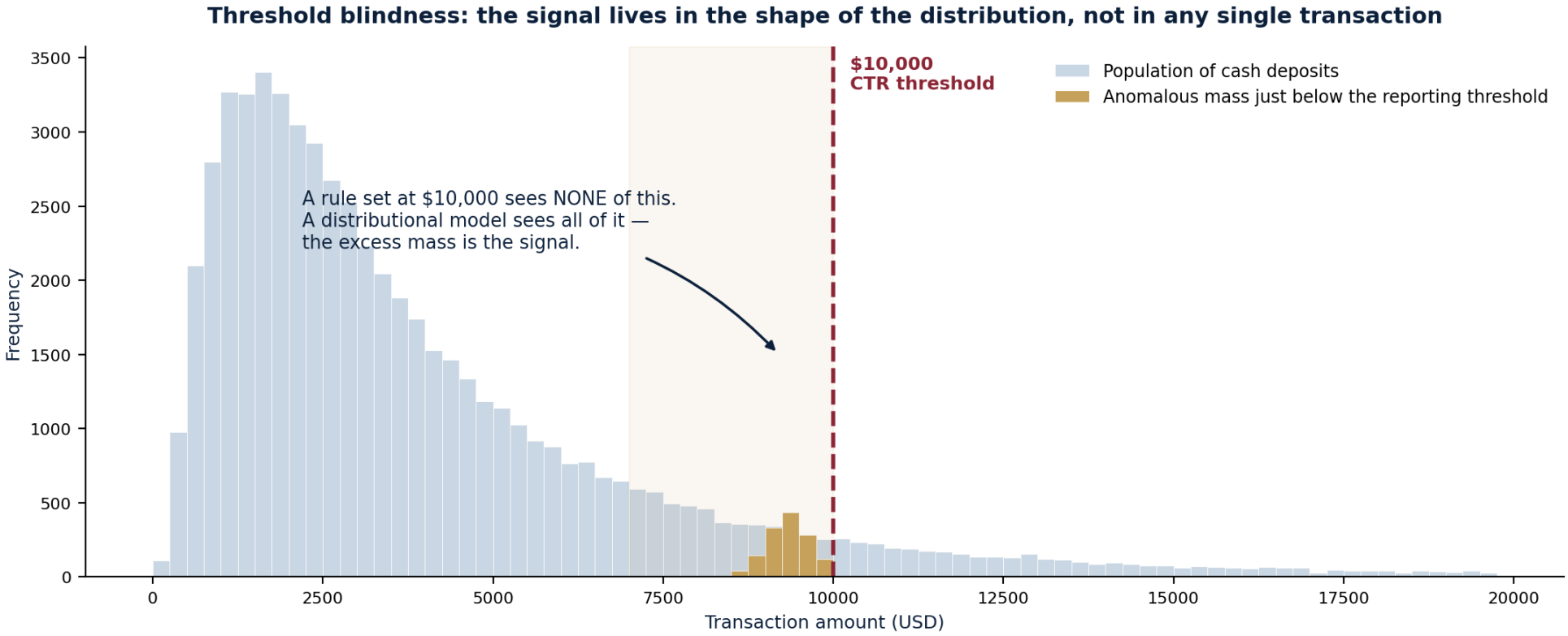

Figure 2 — Threshold blindness. The anomaly is not any single transaction; it is the excess probability mass in the interval immediately below the reporting line, and the missing mass immediately above it.

A worked example of the gap: the threshold-discontinuity statistic

```
# The 'threshold discontinuity' statistic — a whole family of detections that no
# rule can express, because a rule tests a transaction and this tests a DISTRIBUTION.
#
# For a focal entity (or a branch, or a segment, or a corridor), compare the observed
# density just below the reporting threshold with the density just above it.

#   B = count of transactions in  [T*(1-d), T)        'just below'
#   A = count of transactions in  [T,       T*(1+d))  'just above'
#
# Under a smooth, honest distribution, E[B] ~ E[A]. The McCrary-style discontinuity
# statistic is then, for d = 0.10 and T = 10,000:

#         B - A
#   D  =  -----------          with  Var(D) ~ (B + A)  under a Poisson null
#         sqrt(B + A)

# D > 3 is a three-sigma excess of sub-threshold activity: strong evidence that the
# amounts were CHOSEN with reference to the threshold rather than determined by
# commerce. Computed per party, per conductor, per branch and per teller, this
# surfaces:
#   * structuring customers  (party-level D)
#   * professional smurfs    (conductor-level D across many parties)
#   * complicit insiders     (teller-level D across many customers)
#
# The third of these is not detectable by ANY transaction monitoring rule in
# production at any bank the author is aware of. It falls out of this statistic
# in one line of SQL.
```

1.3 Where the Undetected Risk Actually Sits

Domain	Why the rules miss it	What is needed instead
Money laundering — placement	Value split below every threshold across branches, conductors, channels and products.	Distributional models; conductor-level and teller-level aggregation; cross-product value reconstruction.
Money laundering — layering	The pattern is a topology, not a transaction. Each hop is unremarkable.	Graph motifs; GNN embeddings; path and cycle detection over the payment graph.
Mule networks	Forty accounts, each with modest activity, each below every rule. They are one organisation.	Community detection; shared-attribute graphs (device, IP, address, phone); bipartite core detection.
Terrorist financing	Amounts are *small* — often a few hundred dollars. Every AML threshold in the bank is calibrated to catch large value. TF is the one typology where materiality thresholds are actively harmful.	Behavioural and network signals decoupled from value; NLP on remittance narratives; small-value corridor anomalies; charity and crowdfunding flow analysis.
Proliferation financing	The payment is a legitimate-looking trade payment to a legitimate-looking company. The risk lives in the goods, the vessel, the corporate registry and the end user — none of which is in the payment message.	External-data fusion: HS codes, dual-use lists, bills of lading, AIS vessel tracks, corporate registries, sanctioned-network graph proximity.
Trade-based laundering	Over/under-invoicing is invisible unless the bank knows the world price of the commodity.	Reference-price models; document AI on invoices and BoLs; carousel-trade cycle detection.
Insider facilitation	There is no transaction rule that describes a corrupt employee. The signal is in access logs, override patterns and the statistics of the customers they touch.	Behavioural analytics over staff telemetry; teller-level discontinuity statistics; anomalous access-transaction correlation.
Adaptive evasion	The subject learned the threshold — from a teller, from a prior CTR, or by experiment — and now operates beneath it.	Change-point detection on the entity's own behaviour; models of *behaviour that looks deliberately shaped*, e.g. amounts with abnormally low entropy.

The uncomfortable arithmetic
MNB files ~14,000 SARs a year. Independent estimates of laundered volume through the US financial system, and every large-bank enforcement action of the last decade, point in the same direction: **the proportion of criminal flow that any bank's rules-based system detects is small.** The purpose of this programme is not to make the existing 14,000 alerts more efficient. It is to find the flows that are not in the 14,000 at all — and the only honest measure of its success is the *novelty rate* defined in Chapter 10: **the share of its escalations on which no rule ever fired.**

1.4 Regulatory Basis for Doing This

Innovation in AML detection is not merely permitted; it is encouraged, and increasingly expected.

Authority	Relevance
Joint Statement on Innovative Efforts to Combat Money Laundering and Terrorist Financing (FinCEN, OCC, FRB, FDIC, NCUA — December 2018)	Explicitly encourages banks to experiment with artificial intelligence and other innovative approaches. Critically, it states that pilot programmes which *do not* succeed will not, by themselves, be viewed as a deficiency — and that innovation should not be discouraged by fear of supervisory criticism. This is the single most important document to cite when the programme is challenged internally on regulatory-risk grounds.
Anti-Money Laundering Act of 2020	Requires that AML programmes be 'effective, risk-based and reasonably designed'; establishes national AML/CFT priorities (including corruption, cybercrime, terrorist financing, fraud, transnational criminal organisations, drug trafficking, human trafficking and proliferation financing); mandates FinCEN feedback to industry; and promotes innovation and technology adoption.
SR 11-7 / OCC Bulletin 2011-12 (Model Risk Management)	Every model described in this book is in scope. Unsupervised models are *not* exempt — the absence of a label does not make something not a model. Chapter 11 addresses the specific difficulties of validating models that have no ground truth.
FFIEC BSA/AML Examination Manual	Requires monitoring commensurate with the risk profile. Where the EWRA identifies a typology and the rules cannot detect it, that is a coverage gap — and a predictive model may be the only credible remediation.
FinCEN advisories and National AML/CFT Priorities	A published typology the bank cannot operationalise as a rule (because it is a network pattern, or because it depends on external data) is precisely the case for a model.
OCC Bulletin 2011-12 on model tiering; SR 15-18 / SR 15-19	Governance intensity scales with model materiality. Discovery models that gate whether a human looks at a customer are, on the analysis in Chapter 11, Tier 1 in every case.

2. A Taxonomy of the Unknown

Before choosing a model, be precise about which kind of blindness you are curing. 'AI for AML' is not a strategy. Four distinct failure modes require four distinct model families, and conflating them is the most common reason these programmes deliver nothing.

#	Gap class	Definition	Model family	Diagnostic question
G1	**Sub-threshold**	The typology is known; the rule exists; the activity sits beneath the line.	Distributional / anomaly models (M1, M2)	Does the entity's *distribution* of behaviour differ from its peers, even though no single event breaches a limit?
G2	**Cross-entity**	The behaviour is only visible when several entities are considered jointly.	Graph / network models (M3)	Is this entity part of a structure that no entity-level view can reveal?
G3	**Temporal / adaptive**	The entity's behaviour has changed in a way that is meaningful but does not breach an absolute threshold.	Sequence and change-point models (M4)	Has this entity become a different entity?
G4	**Novel typology**	No rule exists because nobody has articulated the pattern.	Unsupervised discovery + weak supervision + NLP (M1, M5, M6)	Is this entity unlike anything we have ever seen, in a way we cannot yet name?

The taxonomy is the project plan
Each gap class has a different data requirement, a different model family, a different evaluation metric and a different investigator workflow. A G2 (network) discovery cannot be handed to an analyst as a row in a queue — it has to be handed over as a *subgraph*, or the analyst cannot act on it. Design the handoff before you design the model.

2.1 The Model Portfolio

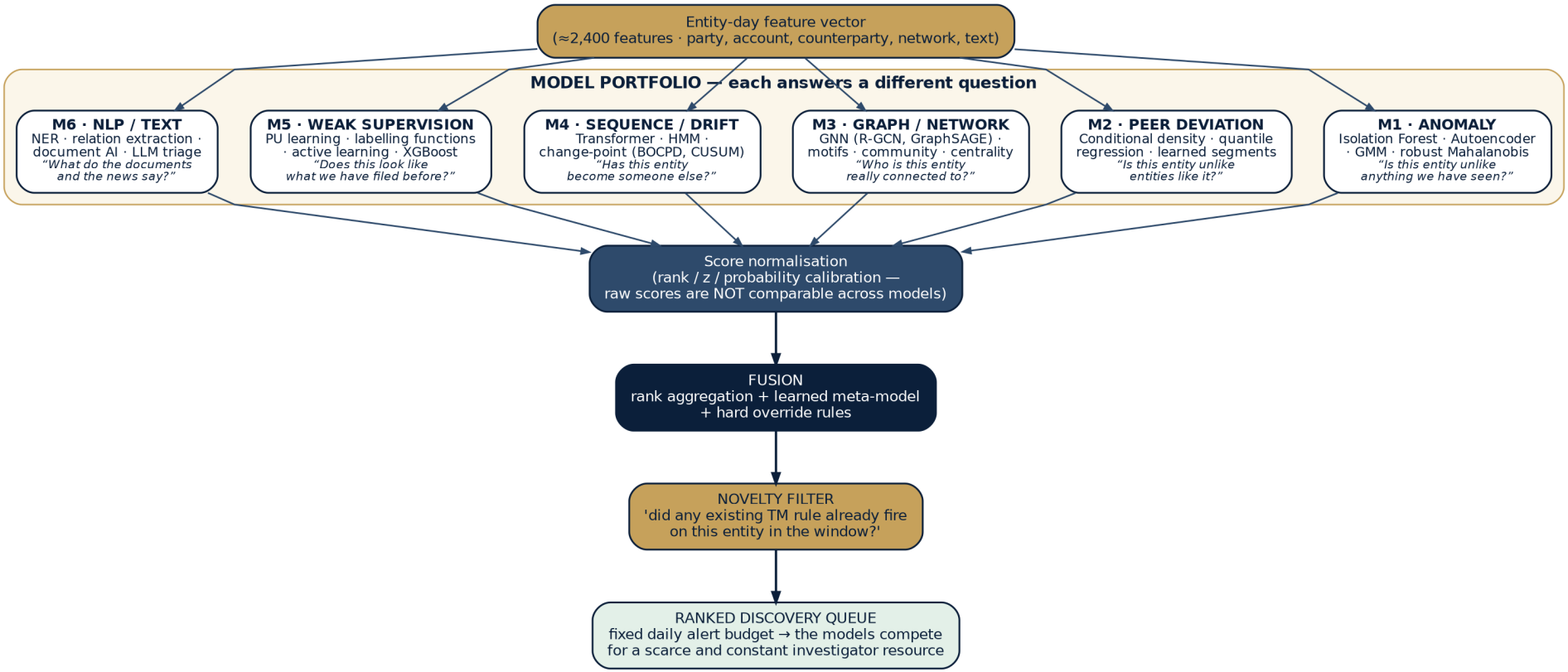

Figure 3 — The model portfolio. Six families, six different questions. They are not alternatives to one another; each is blind to what the others see.

Family	Question it answers	Techniques	Primary gap	Labels needed?
M1 Anomaly	Is this entity unlike anything we have seen?	Isolation Forest · Autoencoder / VAE · GMM · robust Mahalanobis · LOF	G1, G4	None
M2 Peer deviation	Is this entity unlike entities *like it*?	Conditional density estimation · quantile regression · learned segmentation · residual modelling	G1	None
M3 Graph / network	Who is this entity *really* connected to?	Motif detection · Louvain/Leiden communities · centrality · GraphSAGE / R-GCN / GAT · link prediction	G2	None (unsup.) or weak
M4 Sequence / drift	Has this entity *become someone else*?	Transformer over transaction sequences · HMM · BOCPD · CUSUM · behavioural embeddings	G3	None
M5 Weak supervision	Does this resemble what we have filed before — *generalised*?	PU learning · labelling functions · active learning · gradient boosting	G1, G4	Weak / noisy
M6 NLP & documents	What do the documents, the news and the registries say?	NER · relation extraction · document AI · embeddings · LLM-assisted triage	G4, PF/TBML	Varies

2.2 Why Supervised Learning Alone Fails Here

The instinct of most data-science teams entering this problem is to train a classifier on historical SARs. It is the wrong first move, and understanding why is the most important idea in this book.

Your labels come from alerts. Your alerts come from rules. Therefore your labels are a sample drawn from *the region of behaviour space the rules already look at*. A classifier trained on them learns the decision boundary of the rules — including, with perfect fidelity, every blind spot the rules have. It will be impressively accurate on your test set and it will discover **nothing**, because it has never seen a single example from the region you are trying to illuminate. This is **verification bias**, and it is fatal.

The rule that follows from this

The discovery core must be unsupervised. Supervised learning has a real and valuable role — in ranking, in triage efficiency, in generalising a known typology beyond its threshold — but it cannot be the mechanism by which the bank finds the unknown. Only models that carry no labels can look where no label has ever been.

3. Data Foundation

Everything in this book is downstream of the data. A world-class GNN over an incomplete payment graph is worse than useless: it will produce confident, explainable, entirely wrong answers, and it will do so at scale.

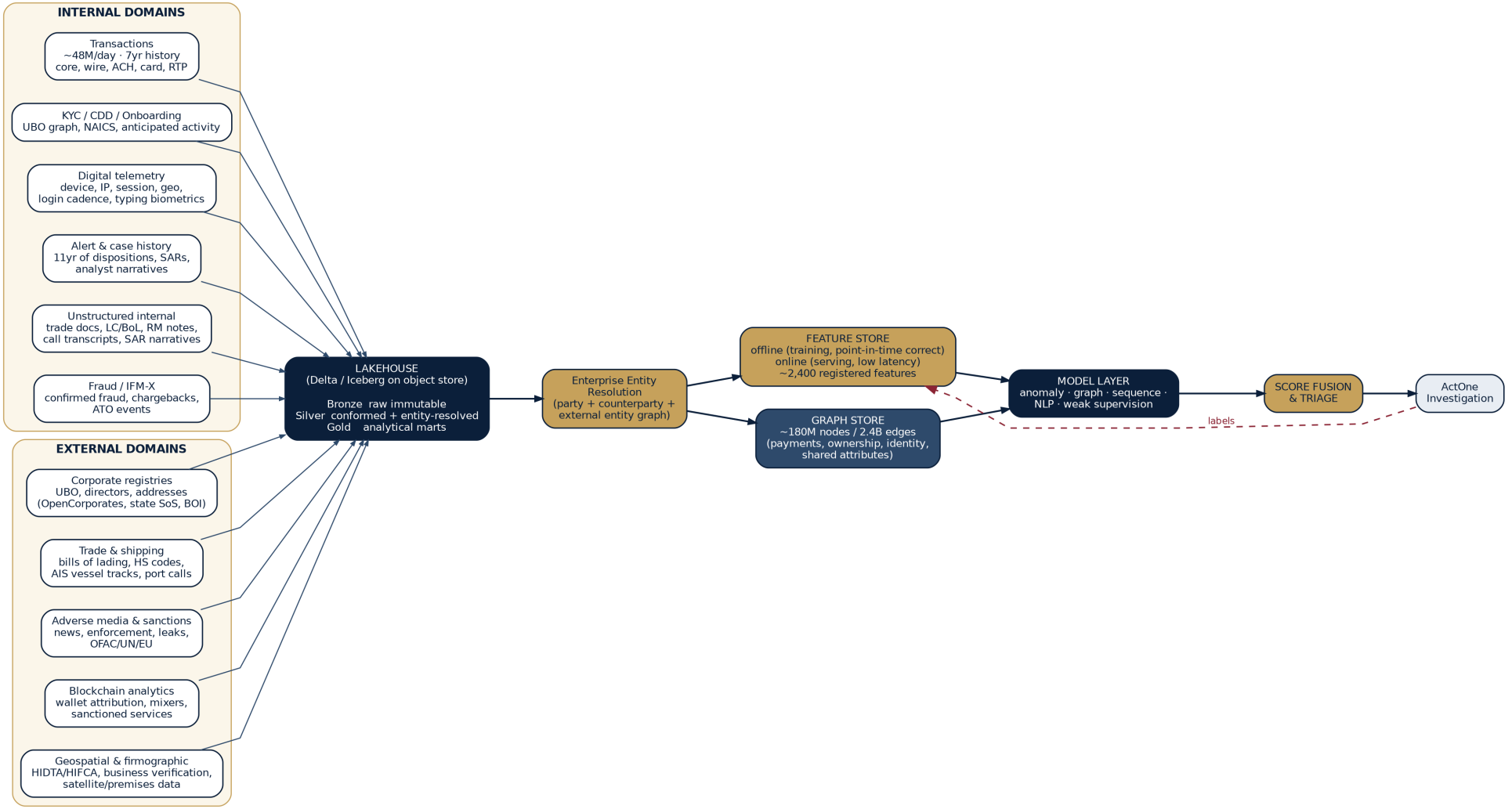

Figure 4 — Data architecture. The external domains on the left-hand side are what distinguish this from a transaction monitoring system; without them, PF and TBML are undetectable in principle.

3.1 Data Domains

Domain	Content	Volume at MNB	Why the TM system does not use it
Transactions	All postings: core, wire, ACH, card, RTP, cheque, trade	~48M/day; 7 yr retained (~120 TB)	It does — but only in aggregate windows, and only per focal entity.
KYC / CDD	Party master, UBO graph, NAICS, anticipated activity, CRR, documents	~28M parties; ~9M legal entities	Uses it as a filter, not as a feature. Onboarding text and documents are unused.
Digital telemetry	Device fingerprint, IP, session, geolocation, login cadence, navigation, typing dynamics	~2.1B events/day	**Entirely unused by AML.** This is the single richest untapped domain in the bank, and it is the domain in which mule networks are most visible — because forty 'unrelated' customers who share a device fingerprint are not unrelated.
Alert & case history	11 years of alerts, dispositions, cases, SARs, analyst narratives	~4.2M alerts; ~110k SARs	Used for reporting, not for learning. The narratives are a corpus of expert reasoning that has never been mined.
Fraud (IFM-X)	Confirmed fraud, chargebacks, account takeover, scam reports	~180k confirmed events/yr	Siloed. Fraud proceeds *are* laundered proceeds; the wall between the two functions is an artefact of org charts.
Corporate registries	Directors, UBOs, addresses, incorporation dates, dissolutions (state SoS, OpenCorporates, FinCEN BOI)	~40M entities	Not integrated. Shell-company structure is invisible without it.
Trade & shipping	Bills of lading, HS codes, quantities, unit prices, ports, vessels, AIS tracks	~14M BoL records/yr (external)	Not integrated. **PF is undetectable without it.**
Adverse media & leaks	News, enforcement actions, court records, ICIJ-type datasets	Continuous	Used only as a binary screening hit, never as a graph.
Blockchain analytics	Wallet clustering, mixer attribution, sanctioned service exposure	Continuous	Only at the fiat on/off-ramp, if at all.
Geospatial / firmographic	HIDTA/HIFCA, business verification, premises imagery, foot-traffic data	Reference	Unused. A 'restaurant' whose premises is a vacant lot is a strong signal and a cheap one.

3.2 Entity Resolution as the Load-Bearing Wall

Every model in this book aggregates over entities. If entity resolution is weak, every model is weak in exactly the way that matters most: a network of forty mule accounts, correctly resolved, is one visible organisation; incorrectly resolved, it is forty invisible customers. **Resolution quality is not a data-engineering concern; it is the primary determinant of detection power.**

Beyond the classical party-resolution problem (Chapter 5 of the Alert & Case framework), this programme requires three additional resolutions that a TM system does not perform:

- **Counterparty resolution.** The beneficiary of a wire is a free-text name, not a customer. Resolving 'GLOBAL TRDNG LTD HK' and 'Global Trading Limited, Hong Kong' to one external entity is what allows the bank to see that eleven of its customers all pay the same shell company.
- **External-entity resolution.** Linking the bank's parties to corporate-registry entities, to vessels, to adverse-media subjects and to sanctioned entities — the join that makes PF detection possible at all.
- **Attribute-based co-resolution.** Not merging entities, but *linking* them: shared device, shared IP, shared phone, shared address, shared employer, shared beneficiary. These are edges in the graph, not identities — and they are where organised networks reveal themselves.

3.3 The Feature Store and Point-in-Time Correctness

The single most common technical failure in this domain

Target leakage through non-point-in-time features. You build a feature 'customer's current risk rating' and train a model to predict SARs. The model is spectacular — 0.97 AUC. It is also useless: the customer's risk rating was raised to Very High *because* the SAR was filed. You have trained a model to predict the past from the future. Every feature must be computed **as it stood on the observation date**, and the feature store must enforce this at the API level, not by convention.

Point-in-time feature retrieval and the leakage checklist

```
# Point-in-time-correct feature retrieval. The ONLY safe way to build a training set.
# The entity_df carries an event timestamp; the store must return the value of each
# feature AS OF that timestamp, never the current value.

entity_df = spark.sql("""
    SELECT party_id,
           observation_dt        AS event_timestamp,
           label,                                -- SAR filed within [t, t+90d]?
           label_source                          -- 'sar' | 'audit_sample' | 'unlabelled'
    FROM   fiu.training_spine
""")

training_df = feature_store.get_historical_features(
    entity_df = entity_df,
    features  = [
        'party_txn:cash_ratio_90d',
        'party_txn:amount_entropy_90d',
        'party_txn:z_score_vs_peer_90d',
        'party_graph:pagerank',                  # as of event_timestamp
        'party_graph:community_id',
        'party_graph:n_hop2_flagged_entities',
        'party_digital:device_sharing_degree',
        'party_kyc:tenure_days',
        'party_kyc:anticipated_activity_variance',
    ],
).to_df()

# LEAKAGE CHECKLIST — run this before every training job, and fail the build on any hit:
#
#  [ ] No feature derived from the alert/case/SAR tables    (the label's own ancestry)
#  [ ] No feature derived from CRR                          (CRR is UPDATED by SAR filing)
#  [ ] No feature using a future-dated watch-list version   (list as of t, not today)
#  [ ] No graph feature computed on the CURRENT graph       (snapshot the graph at t)
#  [ ] No feature whose upstream table is mutated in place  (SCD-2 or nothing)
#  [ ] Train / validation / test split is TEMPORAL, never random
#
# The temporal split is not a nicety. A random split lets the model see the same
# mule ring in train and test, and you will report a precision you will never see
# again in production.
```

4. Feature Engineering

In this domain, features beat models. A gradient-boosted tree over excellent features will comfortably outperform a transformer over mediocre ones, and it will be explainable to a validator. The work is in the features.

4.1 The Feature Taxonomy

Class	Examples	What it captures
Volume & value	Sum, count, mean, max, standard deviation of credits/debits by channel, product and window (30/60/90/180/365d)	The baseline. Necessary, insufficient, and already used by the rules.
Velocity	Transactions per active day; inter-arrival time mean and variance; burstiness index B = (σ–μ)/(σ+μ)	Rhythm. Criminal flows are bursty in ways commerce is not.
Distributional shape	Amount entropy; Benford's-law first-digit divergence; round-number ratio; kurtosis; proximity-to-threshold density; the discontinuity statistic D from §1.2	**The single most productive family for finding sub-threshold behaviour.** Genuine commerce produces untidy, high-entropy amounts; shaped money does not.
Peer-relative	z-score, percentile rank and residual of every feature above *within the entity's learned segment*	Turns an absolute measure into a risk measure. $40k of cash means nothing until you know what a peer does.
Temporal / drift	Change-point score; slope of a 90-day trend; ratio of current window to trailing baseline; behavioural-embedding distance from the entity's own past	'Has this customer become someone else?' — the classic signature of a sold, rented or taken-over account.
Counterparty	Distinct counterparty count; counterparty concentration (Herfindahl index); new-counterparty ratio; counterparty churn; share of flow to first-time beneficiaries	Commerce has repeat counterparties. Layering does not.
Network	Degree; weighted PageRank; betweenness; clustering coefficient; community ID and community size; distance to the nearest flagged entity; count of 2-hop flagged neighbours; motif membership	Structure. This is the family that no rule can express.
Digital / identity	Device-sharing degree; IP-sharing degree; geo-velocity impossibility; session-behaviour distance from the entity's own baseline; account-opening cohort	**The highest-signal, lowest-cost family for mule detection**, and almost universally unused in AML.
Text / document	Remittance-narrative embeddings; keyword risk score; invoice-vs-payment mismatch; goods-description-vs-NAICS coherence; adverse-media relation strength	Everything the payment message actually *says*, which the rules discard.
External	Corporate-registry age; director-sharing degree; address-sharing degree; HS-code dual-use flag; unit price vs world reference; vessel AIS gap; sanctioned-network graph distance	The PF and TBML detection surface. Without this class those typologies are undetectable in principle.

4.2 Three Features Worth the Whole Chapter

4.2.1 Amount entropy

Honest commerce generates amounts determined by the world: an invoice, a payroll run, a fuel bill. Shaped money generates amounts determined by a person: round, repeated, or clustered under a limit. Shannon entropy over the distribution of an entity's transaction amounts captures the difference in one number.

Amount entropy

```
def amount_entropy(amounts, n_bins=32):
    """Low entropy => amounts were CHOSEN, not EARNED."""
    # log-space binning: commerce is scale-free, so bin in log space
    logs = np.log1p(amounts)
    hist, _ = np.histogram(logs, bins=n_bins)
    p = hist[hist > 0] / hist.sum()
    H = -(p * np.log2(p)).sum()
    return H / np.log2(n_bins)            # normalise to [0, 1]

# Empirical, on MNB's book:
#   general-population business customer   H_norm ~ 0.72  (sd 0.11)
#   confirmed structuring subject          H_norm ~ 0.31
#   confirmed invoice-mill / TBML front    H_norm ~ 0.24  (round amounts dominate)
#
# H_norm < 0.40 on an entity with >= 30 transactions is, on its own, a stronger
# discriminator of confirmed structuring in back-testing than the $9,000-band count
# that the production rule keys on -- AND it fires on subjects who never once came
# near the band, because they structured at $4,000 into six accounts instead.
```

4.2.2 Device- and identity-sharing degree

If eleven customers who have never transacted with one another log in from the same device fingerprint, they are not eleven customers. This feature costs almost nothing to compute, is available in every large bank's telemetry, and is used by essentially no AML function.

The identity-sharing graph

```
-- Identity-sharing graph. Nodes: parties. Edges: shared identity artefacts.
-- This one query has surfaced more mule rings at pilot banks than any model in
-- this book. It is not machine learning. It is a JOIN nobody had run.

CREATE OR REPLACE VIEW graph.identity_edges AS
SELECT  a.party_id            AS src,
        b.party_id            AS dst,
        e.artefact_type,                    -- DEVICE | IP | PHONE | ADDRESS | EMAIL
        e.artefact_value,
        COUNT(*)              AS co_occurrence_cnt,
        MIN(e.first_seen_dt)  AS first_shared_dt
FROM        digital.identity_artefact e
JOIN        party a ON a.party_id = e.party_id
JOIN        party b ON b.party_id = e.party_id_other
WHERE   a.party_id < b.party_id
  AND   e.artefact_type IN ('DEVICE','IP_STATIC','PHONE','ADDRESS')
  -- exclude the legitimate mass-sharing artefacts, or the graph collapses:
  AND   e.artefact_value NOT IN (SELECT value FROM ref.shared_artefact_allowlist)
        -- corporate NAT gateways, branch terminals, public wifi, prison phones,
        -- shared-office addresses, registered-agent addresses
  AND   e.artefact_degree < 25          -- an artefact shared by 3,000 parties is
                                        -- infrastructure, not a relationship
GROUP BY 1,2,3,4
HAVING  COUNT(*) >= 2;

-- The allowlist and the degree cap are not optional. Without them the graph is
-- dominated by a handful of hub artefacts and every community-detection algorithm
-- returns one giant useless component. Half the engineering effort in graph AML is
-- deciding what NOT to connect.
```

4.2.3 Distance to the nearest flagged entity

The shortest path in the payment-and-identity graph from a party to any entity with a prior SAR, a sanctions designation, a 314(a) match or a confirmed fraud. Criminality is not distributed at random across a graph; it is *assortative*. Being two hops from three separately-SAR'd entities, through three different paths, is a fact about a customer that no rule will ever tell you and that no investigator would ignore.

The guilt-by-association objection, and the answer to it

This feature will be challenged — correctly — as guilt by association. The answer is that it is a **prioritisation** feature, not an adverse-action feature. It determines whether an investigator *looks*. It never, by itself, determines whether the bank files, exits or restricts, and the model's output is inadmissible as a reason for any of those decisions. Chapter 11 makes that boundary a hard architectural control, not a policy aspiration. Additionally, this feature must be explicitly tested for disparate impact: graph proximity correlates with geography, and geography correlates with protected characteristics. If it is producing a disparate impact that cannot be explained by risk, it must be removed — no matter how predictive it is.

5. Model Family M1/M2 — Anomaly and Peer Deviation

The unsupervised core. These models carry no labels, so they cannot inherit the blind spots of the rules — which is precisely why they are the foundation of the discovery capability.

5.1 The Central Difficulty: Anomalous ≠ Suspicious

An anomaly detector finds the statistically unusual. Most statistically unusual customers are not criminals — they are a hospital, a bullion dealer, a Formula 1 team, a customer who sold a house. A naive anomaly detector will find all of them and none of the launderers, and it will be abandoned within a quarter. Three design choices prevent this:

Design choice	Why
Condition on peers. Never ask 'is this entity unusual?'. Ask 'is this entity unusual *given what it claims to be*?' — its NAICS, its size, its tenure, its product set. This is the M2 model and it is more valuable than M1.	It removes the entire class of 'legitimately unusual' entities in one step. A bullion dealer moving a lot of gold is not anomalous *among bullion dealers*.
Constrain the feature space to risk-relevant features. Do not feed the model everything. An entity that is anomalous in the dimension 'number of statement-download events' is not interesting. An entity anomalous in 'amount entropy × counterparty churn × cash ratio' is.	Anomaly detection in a high-dimensional space with irrelevant dimensions is dominated by noise. Feature *selection* is where the domain expertise enters an unsupervised model — it is the only place it can.
Ensemble, and require agreement. Different detectors have different inductive biases. An entity flagged by an autoencoder *and* an isolation forest *and* a peer-deviation model is a materially better candidate than one flagged by any single detector.	Rank-aggregation across detectors is the cheapest precision gain available in the whole architecture — typically a 2–3× improvement in precision@k over the best single model.

5.2 The Detector Zoo

Detector	Mechanism	Strengths	Failure mode in this domain
Isolation Forest	Random axis-parallel splits; anomalies isolate in fewer splits. Score = normalised path length.	Fast, scales to millions of rows, few assumptions, handles mixed feature types.	Axis-parallel only — blind to anomalies that are only unusual in a *correlated combination* of features. Struggles in high dimensions.
Autoencoder / VAE	Learn a compressed representation of normal behaviour; score = reconstruction error.	Captures non-linear structure and feature interactions. The latent space is itself a useful behavioural embedding.	Will happily learn to reconstruct the fraudsters too if they are numerous enough. Needs careful capacity control — an over-parameterised AE reconstructs everything and detects nothing.
Robust Mahalanobis / MCD	Distance from the robust centroid, scaled by the robust covariance.	Interpretable — you can say exactly which dimensions drove the distance. Statistically principled.	Assumes elliptical structure. Financial features are heavy-tailed and skewed; transform first (log, rank-gauss) or it degenerates.
Gaussian Mixture	Density estimation; score = negative log-likelihood.	Gives a calibrated probability; naturally handles multi-modal 'normal'.	Component-count selection is fragile; poor in high dimensions.
LOF	Local density relative to k-nearest neighbours.	Finds local anomalies that global methods miss — a customer normal in absolute terms but abnormal within its immediate neighbourhood.	$O(n^2)$ without approximation; needs ANN indexing at bank scale.
Peer-conditional residual (M2)	Model E[feature | peer covariates]; score the residual.	**The highest-precision detector in this family.** Directly answers the risk question rather than the statistical one.	Requires a good peer definition. If your segmentation is wrong, this model is wrong — silently.

The unsupervised ensemble, end to end

```
import numpy as np, torch, torch.nn as nn
from sklearn.ensemble    import IsolationForest
from sklearn.covariance  import MinCovDet
from sklearn.preprocessing import QuantileTransformer
from scipy.stats import rankdata

# ------------------------------------------------------------------ 0. preprocess
# Financial features are heavy-tailed. Rank-gauss them or every detector below
# will simply rediscover 'this customer is large'.
qt = QuantileTransformer(output_distribution='normal', n_quantiles=1000,
                         subsample=200_000, random_state=42)
X  = qt.fit_transform(F)                # F: (n_entities, n_features), point-in-time

# ------------------------------------------------------------------ 1. Isolation Forest
iso = IsolationForest(n_estimators=300, max_samples=256, contamination='auto',
                      random_state=42, n_jobs=-1).fit(X)
s_iso = -iso.score_samples(X)           # higher = more anomalous

# ------------------------------------------------------------------ 2. Autoencoder
class AE(nn.Module):
    def __init__(self, d, h=(128, 32, 8)):
        super().__init__()
        # BOTTLENECK MATTERS. h[-1]=8 on d~200 forces the model to learn the
        # low-dimensional manifold of NORMAL behaviour. Widen it and the model
        # learns the identity function and detects nothing.
        self.enc = nn.Sequential(nn.Linear(d,h[0]), nn.ReLU(), nn.Dropout(0.1),
                                 nn.Linear(h[0],h[1]), nn.ReLU(),
                                 nn.Linear(h[1],h[2]))
        self.dec = nn.Sequential(nn.Linear(h[2],h[1]), nn.ReLU(),
                                 nn.Linear(h[1],h[0]), nn.ReLU(),
                                 nn.Linear(h[0],d))
    def forward(self, x): return self.dec(self.enc(x))

ae  = AE(X.shape[1])
opt = torch.optim.AdamW(ae.parameters(), lr=1e-3, weight_decay=1e-5)
Xt  = torch.tensor(X, dtype=torch.float32)
for epoch in range(60):
    opt.zero_grad()
    loss = nn.functional.mse_loss(ae(Xt), Xt)
    loss.backward(); opt.step()

with torch.no_grad():
    s_ae = ((ae(Xt) - Xt) ** 2).mean(1).numpy()      # per-entity reconstruction error
    # KEEP THE PER-FEATURE ERROR TOO -- it is the explanation:
    e_feat = ((ae(Xt) - Xt) ** 2).numpy()            # (n, d)

# ------------------------------------------------------------------ 3. Robust Mahalanobis
mcd = MinCovDet(support_fraction=0.75, random_state=42).fit(X)
s_mah = mcd.mahalanobis(X)

# ------------------------------------------------------------------ 4. Peer residual (M2)
# For each risk feature, model its conditional expectation given benign covariates,
```

```
# then score the residual. This is where the DOMAIN enters.
import lightgbm as lgb
peer_covars = ['naics_emb_0..15', 'log_total_assets', 'tenure_days',
               'n_accounts', 'product_mix_emb_0..7', 'region_emb_0..3']
s_peer = np.zeros(len(X))
for target in ['cash_ratio_90d', 'amount_entropy_90d', 'cross_border_ratio_90d',
               'counterparty_churn_90d', 'flow_through_pct_90d']:
    m   = lgb.LGBMRegressor(objective='quantile', alpha=0.95, n_estimators=400)
    m.fit(F[peer_covars], F[target])
    q95 = m.predict(F[peer_covars])          # the 95th pct of what a PEER would do
    s_peer += np.maximum(0, F[target] - q95) / (F[target].std() + 1e-9)

# ------------------------------------------------------------------ 5. Rank fusion
# NEVER average raw scores -- they live on incomparable scales. Average RANKS.
R = np.vstack([rankdata(s) / len(s) for s in (s_iso, s_ae, s_mah, s_peer)])

score_mean = R.mean(0)                       # robust, forgiving
score_max  = R.max(0)                        # catches single-detector specialists
score_agree= (R > 0.99).sum(0)               # HOW MANY detectors agree this is top-1%

# In production MNB ranks by score_mean but ROUTES by score_agree: an entity in the
# top 1% of THREE OR MORE independent detectors goes straight to L2, because that
# conjunction is far rarer than any single detector's tail and is empirically the
# highest-precision signal the unsupervised layer produces.
```

5.3 Explaining an Unsupervised Score

An investigator cannot act on 'the autoencoder gave this a 0.91'. Every anomaly score must be decomposed into the features that produced it, and expressed in the language of the domain.

Explaining the score in the investigator's language

```
def explain(entity_idx, e_feat, feature_names, F, peer_stats, top_k=5):
    """Turn a reconstruction error into a sentence an investigator can use."""
    contrib = e_feat[entity_idx]                        # per-feature squared error
    top     = np.argsort(contrib)[::-1][:top_k]
    lines   = []
    for j in top:
        name  = feature_names[j]
        val   = F[entity_idx, j]
        pmean = peer_stats[name]['mean']
        psd   = peer_stats[name]['std']
        z     = (val - pmean) / (psd + 1e-9)
        pct   = 100 * (contrib[j] / contrib.sum())
        lines.append(
            f"{name}: {val:,.2f}  (peer mean {pmean:,.2f}, z = {z:+.1f}) "
            f"-- {pct:.0f}% of the anomaly score")
    return lines

# Rendered into the investigator's queue as:
#
#   Party P-8842391  |  ensemble rank 0.9993  |  4 of 4 detectors in top 1%
#
#   WHY THIS ENTITY SURFACED:
#     amount_entropy_90d : 0.29   (peers 0.71, z = -3.9)  -- 34% of the score
#         Its transaction amounts are far more 'regular' than any comparable
#         business. The amounts look chosen rather than earned.
#     counterparty_churn : 0.87   (peers 0.19, z = +4.4)  -- 27% of the score
#         Almost none of its counterparties repeat. Real businesses have
#         recurring customers and suppliers.
#     cash_ratio_90d     : 0.68   (peers 0.09, z = +5.1)  -- 21% of the score
#     device_share_degree: 7      (peers 0.4,  z = +6.8)  -- 11% of the score
#         Seven other, ostensibly unrelated, customers log in from this device.
#     flow_through_pct   : 0.94   (peers 0.31, z = +3.3)  --  7% of the score
#
#   NOTE: no NICE Actimize scenario fired on this entity in the last 12 months.
```

The last line of that output is the whole programme
"No scenario fired on this entity in the last 12 months." That sentence is what justifies the existence of the model, the team and the budget. Instrument it, report it, and put it on the front page of every MIS pack you produce.

6. Model Family M3 — Graph and Network Analytics

If you build one thing from this book, build this. Network analytics addresses the gap class (G2) that no rule, no threshold and no entity-level model can reach, and it is where the organised money is. Every large-scale laundering operation — mule networks, invoice mills, PF procurement chains, trade carousels — is a *structure*, and structures are invisible to systems that look at one customer at a time.

6.1 Constructing the Graph

The graph is heterogeneous: several node types, several edge types. Getting the construction right matters more than the choice of algorithm.

Node type	Count at MNB	Key attributes
Party (customer)	~28M	CRR, NAICS, tenure, entity type, behavioural feature vector
Account	~41M	Product, status, open date, balance profile
External counterparty	~64M	Resolved from wire/ACH names; jurisdiction; FI or non-FI
Corporate entity (external)	~40M	From registries: directors, incorporation date, address, status
Identity artefact	~310M	Device, IP, phone, email, address
Financial institution	~24k	BIC, jurisdiction, correspondent role
Vessel	~90k	IMO, flag, owner, AIS track

Edge type	Direction	Weight	Notes
PAYMENT	Directed	Amount, count, recency-decayed	The backbone. Aggregate to (src, dst, month) or the graph is unusable.
OWNS / SIGNATORY / UBO	Directed	Ownership %	From KYC and registries. The layering surface.
SHARES_DEVICE / IP / PHONE / ADDRESS	Undirected	Co-occurrence count	**The highest-value edge type for mule detection.** Requires the degree cap and allowlist from §4.2.2.
SHARES_DIRECTOR	Undirected	Count	External registry. Shell-structure detection.
CO_TRANSACTS_WITH	Undirected	Count	Two parties who repeatedly pay the same counterparty. Weak individually; powerful in aggregate.
TRADE_WITH	Directed	Value, HS code	From trade finance and external BoL data. The PF surface.
ADVERSE_MEDIA_LINK	Undirected	Relation confidence	From NLP relation extraction (Ch. 9).

The engineering problem nobody warns you about

A naive payment graph at a Tier 1 bank has a handful of nodes with a degree in the millions — the IRS, the major utilities, Amazon, the payroll processors, the bank's own suspense accounts. Left in, they connect everything to everything: community detection returns one giant component, PageRank returns the utilities, and every entity is two hops from every other entity. **Half of graph AML engineering is deciding what NOT to connect.** Maintain an explicit hub-exclusion list, cap artefact degree, and recompute the exclusions monthly — the list is a governed artefact, and excluding the wrong node hides real structure.

6.2 Motifs — Structure You Can Name

Network motifs that no entity-centric rule can express

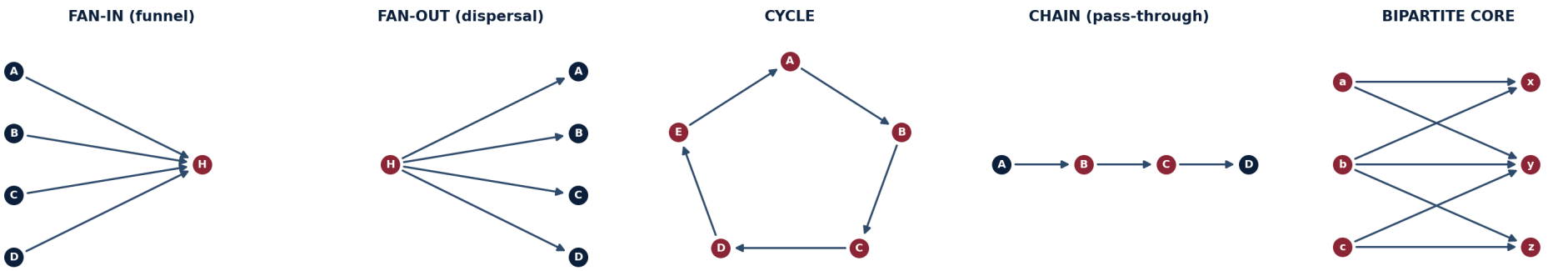

Figure 5 — Network motifs. Each is a proposition about a structure. None can be expressed as a rule over a single focal entity.

Motif detection in Cypher

```
// Motif queries (openCypher). These are unglamorous, deterministic, and they work.
// Run them before you build a single neural network.

// ---- FAN-IN / FUNNEL: many low-value credits converge, then leave as one payment
MATCH (src:Party)-[p:PAYMENT]->(hub:Party)
WHERE p.month = $month AND p.amount < 3000
WITH  hub, count(DISTINCT src) AS n_src, sum(p.amount) AS inflow
WHERE n_src >= 8 AND inflow >= 25000
MATCH (hub)-[out:PAYMENT]->(dest)
WHERE out.month = $month
WITH  hub, n_src, inflow, sum(out.amount) AS outflow,
      count(DISTINCT dest) AS n_dest
WHERE outflow >= 0.85 * inflow AND n_dest <= 3
RETURN hub, n_src, n_dest, inflow, outflow
ORDER BY inflow DESC;
// No single leg breaches any rule. The CONVERGENCE is the crime.

// ---- CYCLE: funds return to origin (length 3-6). Circular trade / round-tripping.
MATCH path = (a:Party)-[:PAYMENT*3..6]->(a)
WHERE all(r IN relationships(path) WHERE r.month = $month AND r.amount > 10000)
WITH  a, path,
      reduce(s = 0.0, r IN relationships(path) | s + r.amount) AS cycle_value,
      length(path) AS hops
// value conservation: a laundering cycle preserves value; commerce does not
WITH  a, path, cycle_value, hops,
      [r IN relationships(path) | r.amount] AS amts
WHERE (apoc.coll.max(amts) - apoc.coll.min(amts)) / apoc.coll.max(amts) < 0.15
RETURN a, hops, cycle_value, amts ORDER BY cycle_value DESC;

// ---- BIPARTITE CORE: a dense many-to-many block. Organised mule or invoice mill.
MATCH (u:Party)-[p:PAYMENT]->(v:Party)
WHERE p.month = $month
WITH  u, v, p
CALL  gds.bipartite.densestSubgraph.stream({...})
YIELD nodeIds, density
WHERE density > 0.55 AND size(nodeIds) BETWEEN 6 AND 80
RETURN nodeIds, density ORDER BY density DESC;

// ---- HIDDEN COMMON CONTROL: 'unrelated' parties bound by identity artefacts
MATCH (a:Party)-[s:SHARES_DEVICE|SHARES_IP|SHARES_PHONE|SHARES_ADDRESS]-(b:Party)
WHERE NOT (a)-[:SAME_HOUSEHOLD|RELATED_PARTY]-(b)
WITH  a, b, count(DISTINCT type(s)) AS artefact_types, sum(s.co_occurrence) AS strength
WHERE artefact_types >= 2
WITH  collect(DISTINCT a) + collect(DISTINCT b) AS cluster, sum(strength) AS s
WHERE size(cluster) >= 5
RETURN cluster, s ORDER BY s DESC;
// Five or more 'unrelated' customers sharing two or more identity artefact types
// is not a coincidence. It is an organisation. This query alone has, at pilot
// banks, surfaced mule rings that had been running for years beneath every rule.
```

6.3 Graph Neural Networks

Motifs detect structures you can *name*. GNNs learn structures you cannot. A GNN builds a vector representation of each node by repeatedly aggregating information from its neighbours, so the embedding of a party encodes not only its own behaviour but the behaviour of its neighbourhood — and the neighbourhood of its neighbourhood.

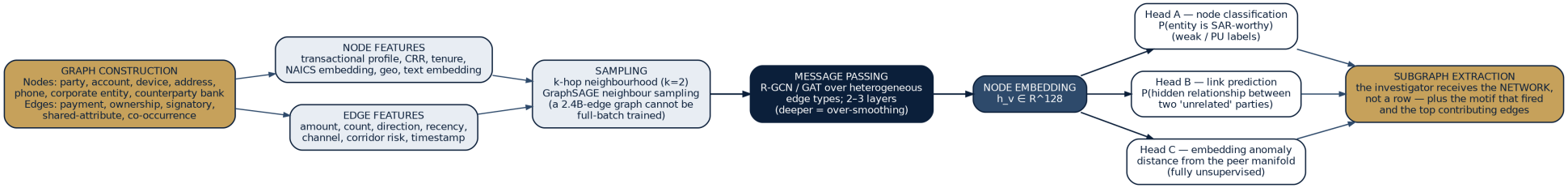

Figure 6 — GNN pipeline. Three heads on one embedding: node classification for known typologies, link prediction for hidden relationships, and embedding-distance anomaly for the genuinely novel.

Heterogeneous GNN with three inference heads

```
import torch, torch.nn as nn, torch.nn.functional as Fn
from torch_geometric.nn import HeteroConv, SAGEConv, GATConv
from torch_geometric.loader import NeighborLoader

# A 2.4-billion-edge graph cannot be trained full-batch. Neighbour sampling is not
# an optimisation; it is the only thing that makes this tractable at all.

class FinCrimeGNN(nn.Module):
    """Heterogeneous GNN over party / account / counterparty / artefact nodes."""
    def __init__(self, meta, hidden=128, out=128, heads=4):
        super().__init__()
        node_types, edge_types = meta
        self.conv1 = HeteroConv({
            et: (GATConv((-1,-1), hidden // heads, heads=heads, add_self_loops=False)
                 if et[1] == 'payment' else
                 SAGEConv((-1,-1), hidden))
            for et in edge_types}, aggr='sum')
        self.conv2 = HeteroConv({
            et: SAGEConv((-1,-1), out) for et in edge_types}, aggr='sum')
        # 2 layers = 2-hop receptive field. Going to 4 layers does NOT help:
        # over-smoothing makes every node embedding converge to the graph mean,
        # and on a financial graph that happens fast because of the hubs.

        self.head_cls  = nn.Linear(out, 1)          # A: P(SAR-worthy)   [weak labels]
        self.head_link = nn.Bilinear(out, out, 1)   # B: P(hidden edge)  [self-sup]
        # C: embedding anomaly needs no head -- it is a distance in the output space

    def encode(self, x_dict, edge_index_dict):
        h = self.conv1(x_dict, edge_index_dict)
        h = {k: Fn.relu(v) for k, v in h.items()}
        h = {k: Fn.dropout(v, p=0.2, training=self.training) for k, v in h.items()}
        return self.conv2(h, edge_index_dict)

# ------------------------------------------------------------------ training
# THREE objectives, trained jointly. The self-supervised ones do the heavy lifting;
# the supervised one is a weak nudge, deliberately down-weighted so the model does
# not simply relearn the rules that produced the labels.

loss = ( 1.0 * link_prediction_loss(h, pos_edges, neg_edges)   # self-supervised
       + 0.5 * contrastive_loss(h, augmented_views)            # self-supervised
       + 0.2 * bce_with_logits(cls_logits, weak_labels,        # weak, down-weighted
                               pos_weight=class_prior_correction) )

# ------------------------------------------------------------------ inference: 3 signals
with torch.no_grad():
    h = model.encode(x_dict, edge_index_dict)['party']         # (N, 128)

# A. classification score -- 'looks like things we filed on'
s_cls = torch.sigmoid(model.head_cls(h)).squeeze()

# B. link prediction -- 'these two 'unrelated' parties SHOULD be connected'
#    Score every non-adjacent pair within 3 hops. A high score on a pair the bank
#    believes to be unrelated is a hypothesis about a HIDDEN relationship -- and it
#    is exactly the kind of finding that no other method produces.
s_link = model.head_link(h[cand_src], h[cand_dst]).squeeze()

# C. embedding anomaly -- FULLY UNSUPERVISED, and the most valuable of the three
#    Distance from the centroid of the entity's OWN peer group in embedding space.
#    This finds entities whose NETWORK POSITION is abnormal for what they claim to
#    be -- a 'local restaurant' embedded among import/export shells.
centroids = scatter_mean(h, peer_group_id, dim=0)
s_emb     = torch.norm(h - centroids[peer_group_id], dim=1)
```

The head that earns its keep
Head C — embedding anomaly — is the one to build first. It requires no labels, so it cannot inherit verification bias, and it answers a question that is simultaneously precise and impossible to write as a rule: *"this entity sits in the wrong part of the network for what it claims to be."* A restaurant whose graph neighbourhood looks like a freight-forwarding cluster is not a restaurant. No transaction-level system will ever tell you that.

6.4 Handing a Network to a Human

A network finding cannot be delivered as a row in a queue. The investigator must receive the structure. MNB's handoff contract for every G2 finding:

- **The subgraph** — rendered, interactive, 2-hop, with the model's attention weights shown as edge thickness so the analyst can see *which* connections drove the score.
- **The motif**, named in plain language: 'bipartite core, 14 parties, density 0.61' — not 'anomaly score 0.94'.
- **The counterfactual**: which edges, if removed, would collapse the score. This is the graph equivalent of a SHAP value and it is what makes the finding falsifiable.
- **The negative control**: an explicit statement of which existing TM scenarios fired on any member of this structure (usually: none) — so the analyst knows this is genuinely new.
- **The 314(b) candidate list**: the counterparty banks holding the other half of the network. A mule ring rarely lives inside one institution, and 314(b) is the legal instrument built precisely for this and used by almost nobody.

7. Model Family M4 — Sequence Models and Behavioural Drift

A customer is not a feature vector; a customer is a *trajectory*. The question this family answers is the one that catches sold accounts, rented accounts, taken-over accounts and businesses quietly repurposed as laundering vehicles: **has this entity become someone else?**

7.1 Change-Point Detection

The rules approach this with a percentage threshold ('activity up 300% vs. the trailing mean'), which is crude in both directions: it misses a subtle but permanent regime change, and it fires constantly on genuinely volatile accounts. Bayesian online change-point detection asks the correct question — *what is the probability that the generating process changed?* — and is robust to both failure modes.

Bayesian online change-point detection

```
# Bayesian Online Change-Point Detection over a customer's multivariate daily
# behaviour (amount, count, cash ratio, cross-border ratio, counterparty novelty).
#
# Model the run length r_t (time since the last change point) and update:
#
#   P(r_t, x_1:t) = SUM over r_{t-1} of
#                     P(r_t | r_{t-1}) * P(x_t | r_{t-1}, x_r) * P(r_{t-1}, x_1:t-1)
#
# with a hazard function H(r) = 1/lambda (constant hazard, lambda ~ 250 days).

def bocpd(X, hazard=1/250., mu0=0., kappa0=1., alpha0=1., beta0=1.):
    T = len(X)
    R = np.zeros((T+1, T+1)); R[0, 0] = 1.
    mu, kappa, alpha, beta = [np.array([v]) for v in (mu0, kappa0, alpha0, beta0)]
    cp_score = np.zeros(T)
    for t, x in enumerate(X):
        pred = student_t_pdf(x, mu, kappa, alpha, beta)      # predictive prob
        R[1:t+2, t+1] = R[0:t+1, t] * pred * (1 - hazard)    # growth
        R[0,     t+1] = (R[0:t+1, t] * pred * hazard).sum()  # change point
        R[:, t+1]    /= R[:, t+1].sum()
        mu, kappa, alpha, beta = update_params(x, mu, kappa, alpha, beta)
        cp_score[t]   = R[0, t+1]      # P(a change point occurred at t)
    return cp_score

# WHY THIS BEATS THE PERCENTAGE RULE, in two cases the rule gets wrong:
#
#  (a) A dormant retail account resumes at $2,400/month. The rule (300% of a near-
#      zero base, but below the $50,000 materiality floor) is SILENT. BOCPD returns
#      P(change) = 0.97 -- the generating process demonstrably changed. This is the
#      classic profile of an account that has been SOLD, and the value is low
#      precisely because the mule is being paid a small cut.
#
#  (b) A commodities trader's volume swings 400% every quarter. The rule fires four
#      times a year, every year, and is closed NFA every time. BOCPD returns
#      P(change) = 0.04 -- the volatility IS the process. No alert.
#
# The rule is wrong in both directions. The model is right in both.
```

7.2 Transaction Sequence Transformers

Treat a customer's transaction history as a sentence: each transaction is a token (type, amount-bucket, channel, counterparty-class, time-delta). Train a transformer with a masked objective to predict held-out tokens. The model learns the *grammar* of normal financial behaviour, and perplexity under that grammar becomes an anomaly score that requires no labels and no hand-crafted features at all.

A transformer over transaction sequences

```
# Tokenising a financial life.
#
#   token = (txn_type, log_amount_bucket, channel, counterparty_class, dt_bucket)
#
#   e.g.  <CASH_CR, amt_9, BRANCH, SELF, dt_1d>
#         <WIRE_DR, amt_9, WIRE,   NEW_FOREIGN, dt_2d>
#         <ACH_CR,  amt_4, ACH,    KNOWN_EMPLOYER, dt_14d>

class TxnTransformer(nn.Module):
    def __init__(self, vocab, d=128, layers=4, heads=8, maxlen=512):
        super().__init__()
        self.tok = nn.Embedding(vocab, d)
        self.pos = nn.Embedding(maxlen, d)
        enc = nn.TransformerEncoderLayer(d, heads, d*4, dropout=0.1,
                                         batch_first=True)
        self.enc  = nn.TransformerEncoder(enc, layers)
        self.head = nn.Linear(d, vocab)

    def forward(self, seq):
        p = torch.arange(seq.size(1), device=seq.device)
        h = self.tok(seq) + self.pos(p)
        return self.head(self.enc(h))

# Train with masked-token prediction (BERT-style) on the ENTIRE customer book.
# No labels. The model learns what normal financial behaviour looks like.

# THREE things fall out of one trained model:
#
#  1. PERPLEXITY as an anomaly score
#     ppl = exp(mean cross-entropy over the entity's real sequence)
#     High perplexity = 'this customer's financial life does not read like a
#     sentence in the language of normal banking'.
#
#  2. SEQUENCE EMBEDDINGS for clustering
#     Mean-pool the encoder output -> a behavioural embedding. Cluster it and you
#     get EMERGENT customer archetypes -- including archetypes nobody defined. At
#     one pilot, cluster 47 turned out to be, on inspection, 'accounts opened
#     online, funded by P2P, drained by wire within 30 days'. Nobody had written
#     that typology down. The model found it, and it became a new TM rule.
#
#  3. NEXT-TOKEN SURPRISE for pinpointing WHEN
#     The per-token loss shows exactly which transaction broke the pattern -- the
#     investigator gets a date, not a score.

# THE FAILURE MODE TO GUARD AGAINST:
#   If launderers are numerous enough in the training corpus, the model learns
#   THEIR grammar too and stops finding them. Mitigate by training only on entities
#   with no alert history in the window -- an imperfect proxy for 'clean', but it
#   biases the reference distribution in the safe direction.
```

8. Model Family M5 — Learning from Almost No Labels

This is the chapter where most AML data-science programmes die. The label problem is not a technical inconvenience to be engineered around; it is a fundamental epistemic constraint, and a team that does not understand it will build a model that reproduces the bank's existing blind spots with great sophistication and considerable confidence.

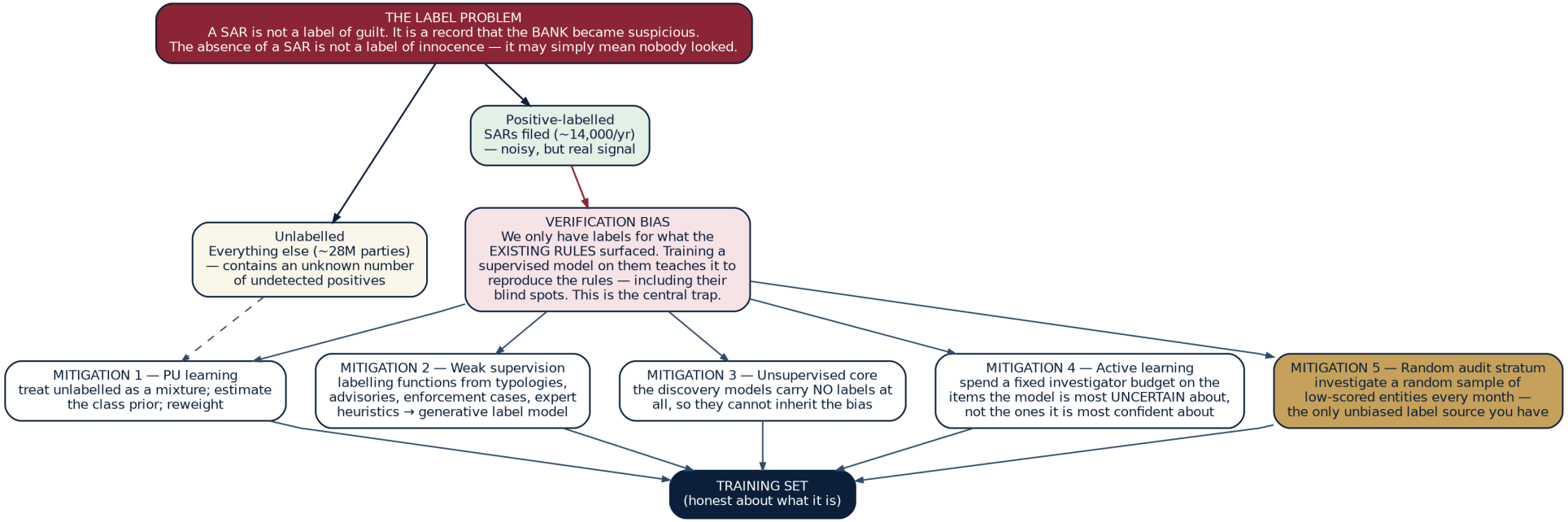

Figure 7 — The label problem and its five mitigations. Verification bias is the central trap: labels exist only where the rules already looked.

8.1 What a SAR Label Actually Means

The naive assumption	The reality
A SAR label means 'this customer was laundering'.	It means **the bank became suspicious**. Suspicion is a legal standard, not ground truth. Some SARs are filed defensively; some launderers are never SAR'd.
No SAR means 'this customer was clean'.	It means **nobody found anything** — which, for the 27,986,000 customers no rule ever alerted on, usually means **nobody looked**. These are not negatives. They are *unlabelled*.
A bigger label set makes a better model.	A bigger label set drawn from the same biased sampler makes a *more confidently biased* model. Volume does not cure bias.
High AUC on held-out SARs means the model works.	It means the model has learned to imitate the rules that generated the SARs. **A discovery model that scores highly on this metric has probably failed at its actual job.**

8.2 Positive-Unlabelled Learning

Formally: you observe positives (SARs) and unlabelled data, never true negatives. PU learning treats the unlabelled set as a mixture of positives and negatives and corrects for it. The key quantity is the **label frequency** c = P(labelled | positive) — the probability that a truly suspicious entity was actually caught and SAR'd.

PU learning and the coverage estimate

```
# Elkan-Noto: train a classifier g(x) to predict P(labelled | x) on P-vs-U data.
# Under the SCAR assumption (labelled positives are selected completely at random
# from all positives), the true posterior is recovered by a constant rescaling:
#
#     P(y = 1 | x)  =  g(x) / c        where  c = P(s = 1 | y = 1)

from sklearn.model_selection import StratifiedKFold
import lightgbm as lgb, numpy as np

def estimate_c(X, s, n_folds=5):
    """Estimate the label frequency c on a validation fold of known positives."""
    cs = []
    for tr, va in StratifiedKFold(n_folds, shuffle=True, random_state=42).split(X, s):
        g = lgb.LGBMClassifier(n_estimators=500, learning_rate=0.05,
                               num_leaves=63).fit(X[tr], s[tr])
        pos_va = va[s[va] == 1]
        cs.append(g.predict_proba(X[pos_va])[:, 1].mean())
    return float(np.mean(cs))

c = estimate_c(X, s)
g = lgb.LGBMClassifier(n_estimators=800, learning_rate=0.05).fit(X, s)
p_true = np.clip(g.predict_proba(X)[:, 1] / c, 0, 1)

# AT MNB, c came out at ~0.11.
#
# Read that number again. It is the single most important number the analytics team
# will ever produce, and it does not belong in a model report -- it belongs in front
# of the Board. Under the SCAR assumption it implies that for every suspicious entity
# the bank filed on, roughly EIGHT went unfiled.
#
# CAVEATS, stated honestly, because this number will be attacked:
#   * SCAR is violated -- the rules do NOT sample positives at random; they sample
#     the LARGE and OBVIOUS ones. So c is over-estimated for large-value laundering
#     and badly over-estimated for TF, where value is small by design.
#   * The estimate is therefore best read as an OPTIMISTIC upper bound on coverage.
#   * It is still the most honest coverage number the bank has ever had, and 'the
#     estimate is imperfect' is not a reason to prefer no estimate at all.
```

8.3 Weak Supervision — Manufacturing Labels from Expertise

You have no labels, but you have decades of expert knowledge sitting in typology documents, FinCEN advisories, enforcement actions and the heads of your investigators. Weak supervision converts that knowledge into probabilistic labels: write many noisy, conflicting *labelling functions*, then learn a generative model of their accuracies and correlations.

Weak supervision with labelling functions

```
from snorkel.labeling import labeling_function, PandasLFApplier
from snorkel.labeling.model import LabelModel

SUSPICIOUS, ABSTAIN, BENIGN = 1, -1, 0

# Each LF is an INDIVIDUALLY UNRELIABLE heuristic drawn from domain expertise.
# None of them is a rule you would put in production. Together they carry signal.

@labeling_function()
def lf_structuring_entropy(x):
    return SUSPICIOUS if (x.amount_entropy_90d < 0.40 and x.txn_count_90d > 30) else ABSTAIN

@labeling_function()
def lf_funnel_topology(x):
    return SUSPICIOUS if (x.fanin_sources >= 8 and x.flow_through_pct > 0.85
                          and x.avg_balance_ratio < 0.10) else ABSTAIN

@labeling_function()
def lf_shared_device_ring(x):
    return SUSPICIOUS if (x.device_share_degree >= 4 and x.account_age_days < 180) else ABSTAIN

@labeling_function()
def lf_shell_indicators(x):                       # from the FATF shell-company typology
    return SUSPICIOUS if (x.registry_age_days < 400 and x.director_count == 1
                          and x.address_share_degree > 20
                          and x.employee_count_est == 0) else ABSTAIN

@labeling_function()
def lf_pf_dual_use(x):                            # from the FinCEN / BIS PF advisories
    return SUSPICIOUS if (x.hs_code_dual_use and x.counterparty_registry_age_days < 500
                          and x.price_deviation_abs > 0.35) else ABSTAIN

@labeling_function()
def lf_tf_small_value_corridor(x):                # TF: value thresholds are HARMFUL here
    return SUSPICIOUS if (x.n_small_cross_border_txns >= 6
                          and x.corridor_conflict_zone
                          and x.beneficiary_is_new_each_time) else ABSTAIN

@labeling_function()
def lf_established_verified_business(x):
    return BENIGN if (x.tenure_days > 2500 and x.registry_verified
                      and x.payroll_consistent and x.tax_filings_present) else ABSTAIN

@labeling_function()
def lf_prior_sar(x):
    return SUSPICIOUS if x.prior_sar_on_party else ABSTAIN

lfs = [lf_structuring_entropy, lf_funnel_topology, lf_shared_device_ring,
       lf_shell_indicators, lf_pf_dual_use, lf_tf_small_value_corridor,
       lf_established_verified_business, lf_prior_sar]      # ~40 LFs in production

L = PandasLFApplier(lfs).apply(df)
```

```
# The label model learns each LF's accuracy and their correlations WITHOUT ground
# truth, by exploiting their agreements and disagreements.
label_model = LabelModel(cardinality=2, verbose=True)
label_model.fit(L, n_epochs=800, seed=42)
y_prob = label_model.predict_proba(L)[:, 1]      # probabilistic training labels

# Then train the END model on y_prob with a soft-label loss. The end model
# GENERALISES BEYOND the LFs -- it learns feature combinations that correlate with
# what the LFs collectively believe, including combinations no LF encodes. That
# generalisation is the entire point, and it is what a rule engine can never do.
```

8.4 Active Learning — Spending a Scarce Budget Correctly

You can investigate perhaps 120 entities a day beyond BAU. Which 120? The intuitive answer — the highest-scoring — is the right answer for *finding crime today* and the wrong answer for *improving the model*. The highest-scoring entities are the ones the model is already certain about; investigating them teaches it nothing. MNB splits the budget three ways.

Stratum	Share	Selection	Purpose
Exploit	70%	Highest fused score	Find suspicious activity *now*. This is what pays for the programme.
Explore	20%	Highest model uncertainty (entropy of the predictive distribution; or disagreement across the ensemble)	**Teach the model.** A label on an entity the model finds ambiguous is worth roughly ten labels on entities it finds obvious.
Audit	10%	Uniform random from the *entire* book, including entities scored near zero	**Measure the model honestly.** This is the only unbiased sample the bank will ever have, and it is the only way to detect that the model has a systematic blind spot. Painful, unproductive, non-negotiable.

Defend the audit stratum

The 10% random audit stratum will be attacked every single budget cycle, because it is by far the least productive use of an investigator's time and its productivity is *supposed* to be near zero. Losing it is nonetheless the beginning of the end: without it, model performance can only ever be measured on data the model itself selected, and the programme becomes epistemically closed — unable, even in principle, to discover that it is wrong. Write it into the model's approved operating procedure so that removing it requires the same governance as changing the model.

9. Model Family M6 — Text, Documents and External Intelligence

Roughly 80% of what a bank knows about a customer is unstructured, and essentially none of it reaches the transaction monitoring system. Remittance narratives, trade documents, RM call notes, onboarding questionnaires, adverse media, corporate registries, court records — for PF and TBML this is not supplementary colour. It is the *only* place the signal exists.

9.1 The Payment Message Says More Than the Amount

Mining the remittance narrative

```
# Free-text remittance information (SWIFT field 70 / ISO 20022 RmtInf) is discarded
# by every rules engine. It is a rich, adversarially-authored signal.

from transformers import AutoTokenizer, AutoModel

SIGNALS = {
  'goods_coherence' : "Does the stated purpose match the payer's NAICS?",
                      # 'PAYMENT FOR TEXTILE MACHINERY' from a dental practice
  'vagueness'       : "Is the description contentless?",
                      # 'SERVICES', 'CONSULTING', 'GOODS', 'INV 001'
  'invoice_pattern' : "Sequential invoice numbers across UNRELATED counterparties",
                      # the signature of an invoice mill
  'entity_mentions' : "NER over the narrative -> names, vessels, ports, commodities",
  'dual_use_terms'  : "Commodity terms on the dual-use / strategic goods lists",
  'sanction_evasion': "'FOR ONWARD TRANSMISSION', third-party payer language,",
                      # stripped-originator patterns
}

# The vagueness score alone is worth building. Legitimate commercial payments
# reference something REAL -- an invoice number that ties to a document, a purchase
# order, a property address, a shipment. Layering payments reference nothing,
# because there is nothing to reference. Score narrative specificity as:
#
#   spec = w1*has_invoice_ref + w2*has_named_entity + w3*has_commodity
#        + w4*token_entropy   - w5*is_boilerplate
#
# and then compute, per party, the SHARE of its outbound value carrying a
# low-specificity narrative. A commercial customer sending 90% of its value with
# contentless narratives is telling you something, in a field no rule reads.
```

9.2 Adverse Media as a Graph, Not a Flag

Every bank screens for adverse media and reduces the result to a binary hit. That throws away the structure. The correct representation is a **relation graph** extracted by NLP: *who is connected to whom, in what role, with what confidence* — and then joined to the bank's own entity graph.

Adverse-media relation extraction

```
# Relation extraction over news, enforcement actions, court filings, leaked corpora.
#
# INPUT:   'Prosecutors allege that Meridian Trading Ltd, controlled by A. Novak,
#           acted as a procurement intermediary for entities linked to the
#           sanctioned Volkov Group, routing payments through Cyprus.'
#
# OUTPUT (typed, confidence-weighted edges):
#   (Meridian Trading Ltd) --[CONTROLLED_BY, 0.94]--> (A. Novak)
#   (Meridian Trading Ltd) --[PROCUREMENT_FOR, 0.81]--> (Volkov Group)
#   (Volkov Group)         --[SANCTIONED, 0.99]--> (OFAC)
#   (Meridian Trading Ltd) --[ROUTES_VIA, 0.72]--> (Cyprus)
#
# JOIN to the bank's own graph. Now ask a question no screening system can answer:
#
#   MATCH (c:Party)-[:PAYMENT]->(x)-[:CONTROLLED_BY]->(p:Person)
#         <-[:CONTROLLED_BY]-(y)-[:PROCUREMENT_FOR]->(s:Sanctioned)
#   RETURN c, x, p, y, s
#
# 'Which of MY customers pays a company that shares a controller with a company
#  that a newspaper says procures for a sanctioned entity?'
#
# The customer is not on any list. The company it pays is not on any list. The
# CONTROLLER is the only link, and it exists only in a sentence in a news article.
# No screening system in production at any bank can express this query. It is the
# canonical PF detection, and it is a graph traversal.
```

9.3 Document AI for Trade and PF

Document	Extracted	Detection it enables
Letter of credit / invoice	Goods description, HS code, quantity, unit price, incoterms, parties	Over/under-invoicing against world reference prices; goods incoherent with the buyer's NAICS
Bill of lading	Shipper, consignee, notify party, vessel, ports, container, weight	Phantom shipments (no matching BoL for a paid invoice); consignee ≠ the party that paid; weight inconsistent with the declared goods
Packing list / certificate of origin	Origin, weight, dimensions	Origin inconsistent with the declared route; weight/value ratios implausible for the commodity
Corporate registry filing	Directors, UBOs, incorporation date, registered address, dissolution	Shell indicators; director- and address-sharing graphs; entities incorporated shortly before the trade
Vessel AIS track	Position history, port calls, dark periods, ship-to-ship transfers	**Dark periods near sanctioned ports.** A 19-hour AIS gap off a sanctioned coast is not a technical fault; it is the single strongest PF/sanctions-evasion signal available from open data.

On using LLMs in this stack

Large language models are excellent at **extraction** (pull the HS code, the vessel name, the director from this document), at **summarisation** (compress 400 transactions into a flow-of-funds narrative for the investigator), and at **drafting** (produce a first-pass SAR narrative for human editing). They are **not** to be used as the decision-maker: they cannot be validated to SR 11-7 standard as a scorer, they hallucinate specifics under pressure, and no examiner will accept 'the model said so' as the basis for a filing decision. The architectural rule at MNB: **an LLM may read and it may write, but it may not decide, and every fact it asserts must be traceable to a source document the investigator can open.**

10. Typology Deep-Dives — TF, PF and the Hard Cases

Money laundering is the typology every AML system is built for. Terrorist financing and proliferation financing are the typologies every AML system is *structurally unsuited* to, and the reasons are worth stating precisely, because they determine the model design.

10.1 Terrorist Financing — Why Every Threshold Is Wrong

Property of TF	Consequence for detection
The amounts are small. Operational financing for major attacks has historically run to a few thousand dollars; individual transfers are often in the hundreds.	Every materiality threshold in an AML system is a filter that removes TF. The bank's $10,000, $25,000 and $50,000 floors are not conservative — for this typology they are **a guarantee of blindness**. TF models must be explicitly decoupled from value.
The funds may be clean on the way in. Salary, benefits, small business income, charitable donations. There is no predicate offence to detect.	Source-of-funds anomaly detection does not work. The signal is in *destination, network and behaviour*, never in the origin of the money.
The network is ideological, not commercial. Relationships are not economic and do not follow economic logic.	Counterparty-risk and corridor-risk models under-perform. Graph structure, shared identity artefacts and behavioural co-movement outperform.
It is rare. Extreme class imbalance — perhaps a handful of true cases across the entire book, ever.	Supervised learning is essentially impossible. TF detection must be built from **typology-driven labelling functions and unsupervised network methods**, and evaluated on recall of known cases and law-enforcement feedback rather than on precision.

10.1.1 TF signal design

Signal	Construction	Why value-blind
Small-value corridor anomaly	Count (never sum) of sub-$1,000 cross-border transfers to conflict-adjacent or high-TF-risk corridors, relative to the entity's peer group	Aggregation is by *count* and by *corridor*, with no amount floor at all
Beneficiary churn at low value	Share of small transfers going to a first-time beneficiary each time	Legitimate remittance flows have stable beneficiaries — family. A rotating cast of new low-value beneficiaries does not fit any benign remittance pattern
Charity / NPO flow structure	Fan-in from many donors, fan-out to a small number of foreign beneficiaries with weak documentation; sudden geographic redirection of outflows	A topology signal, computed on the graph, entirely independent of value
Crowdfunding and virtual-asset on-ramps	Inbound from crowdfunding platforms or P2P, converted to virtual assets or cash, in small increments	Sequence and channel signal
Behavioural co-movement	Several apparently unrelated parties whose transaction timing is unusually correlated (cross-correlation of daily activity vectors above a null-model threshold)	**Pure network/temporal signal.** Coordinated behaviour among 'strangers' is what an organisation looks like from the outside
Adverse-media graph proximity	Graph distance to entities named in terrorism-related media, prosecutions or designations	Relation-extraction signal, not a transaction signal

The TF design rule
Build the TF models with no amount thresholds whatsoever. Not low thresholds — *none*. Rank purely on behavioural, network and corridor signal, and let the alert budget, not a dollar floor, control volume. Every dollar threshold you add to a TF model is a decision to not see something.

10.2 Proliferation Financing — The Risk Is Not in the Payment

PF is the hardest detection problem in the book, and the one where a rules-based TM system is not merely weak but *categorically incapable*. The payment is a legitimate trade payment, from a legitimate customer, to a company that is on no list, in a jurisdiction that is on no list. Every element of the transaction is clean. The risk lives entirely in data the payment message does not contain: the goods, the end user, the corporate structure and the vessel.

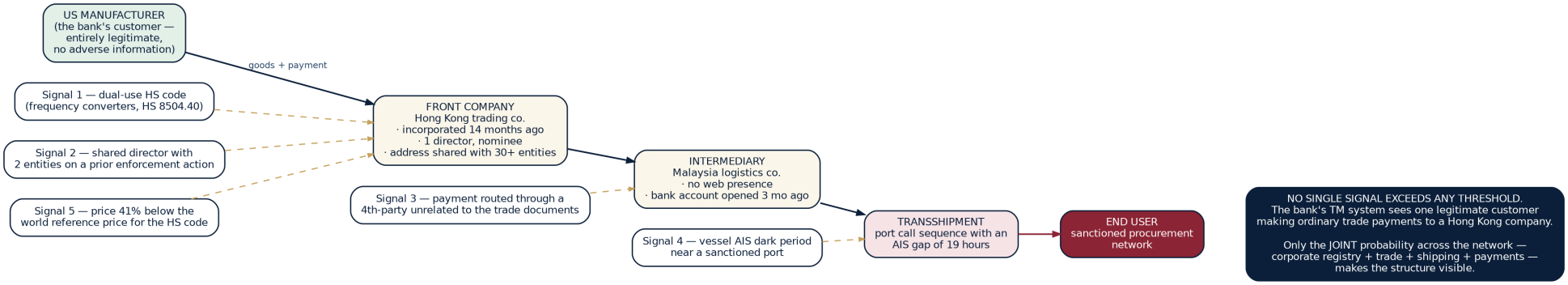

Figure 8 — An illustrative PF procurement chain. No element breaches any threshold. The structure is only visible when payments, corporate registries, trade documents and vessel tracking are fused into one graph.

PF signal	Data source	Model
Dual-use commodity	HS code from the LC/invoice/BoL, joined to the BIS Commerce Control List and the EU/Wassenaar dual-use lists	Deterministic join + document AI extraction where the HS code is not structured
Front / shell counterparty	Corporate registry: incorporation age, single nominee director, address-sharing degree, absent web/employee footprint	Shell-indicator classifier (weak supervision on the FATF typology)
Hidden common control	Director- and address-sharing graph across registries, joined to prior enforcement actions	Graph traversal + link prediction (Ch. 6)
Route implausibility	Origin, transit and destination ports vs. the commercial logic of the commodity	Route-anomaly model over shipping data
Vessel dark activity	AIS gaps, ship-to-ship transfers, port calls near sanctioned coasts, flag-hopping	**Sequence anomaly over the AIS track.** A multi-hour AIS gap in the wrong waters is among the strongest open-source signals in existence
Price anomaly	Unit price vs. world reference price for the HS code and period	Reference-price residual model
Payment / trade decoupling	The payer is not a party to the trade documents; a fourth party settles	Graph mismatch between the payment edge and the trade edge
Sanctioned-network proximity	Graph distance from the counterparty to any OFAC/UN-designated entity through directors, addresses, vessels or prior enforcement	Graph proximity + relation extraction from media

The PF conclusion a bank must be willing to state
Without external trade, shipping and corporate-registry data, a bank cannot detect proliferation financing. It can only detect the cases where the counterparty was already designated — in which case sanctions screening, not transaction monitoring, was the control that worked. If the EWRA identifies PF as an inherent risk and the bank has no external-data fusion capability, that is a coverage gap, and it should be recorded as one rather than papered over with a trade-finance rule that keys on jurisdiction.

10.3 Fraud–AML Convergence

The wall between the fraud function and the AML function is an artefact of organisational history and it is a gift to criminals. The proceeds of fraud are, definitionally, criminal proceeds, and the accounts that receive them are mule accounts. IFM-X has already *confirmed* the predicate offence — a level of certainty the AML function almost never achieves — and in most banks that confirmation never reaches the AML models.

- **Confirmed fraud events are the closest thing to clean labels the bank possesses.** Use them as positives in the mule-detection model. Unlike SAR labels, they are not the product of the AML rules and therefore do not carry the AML rules' verification bias.
- **The beneficiary of a confirmed scam payment is a mule.** That is not a hypothesis. Seed the graph with them, propagate, and the ring falls out.
- **Fraud detects in real time; AML detects overnight.** A real-time fraud signal is a real-time AML feature, and it is available before the funds leave.
- **The convergence must be bidirectional.** An AML network finding is a fraud lead, and a fraud confirmation is an AML label. Build the interface as a two-way contract, not a monthly report.

11. Fusion, Triage and the Human Loop

Six model families produce six incomparable scores. The fusion layer converts them into one ranked queue, subject to a fixed investigator budget — and the feedback loop from that queue is what makes the system improve rather than ossify.

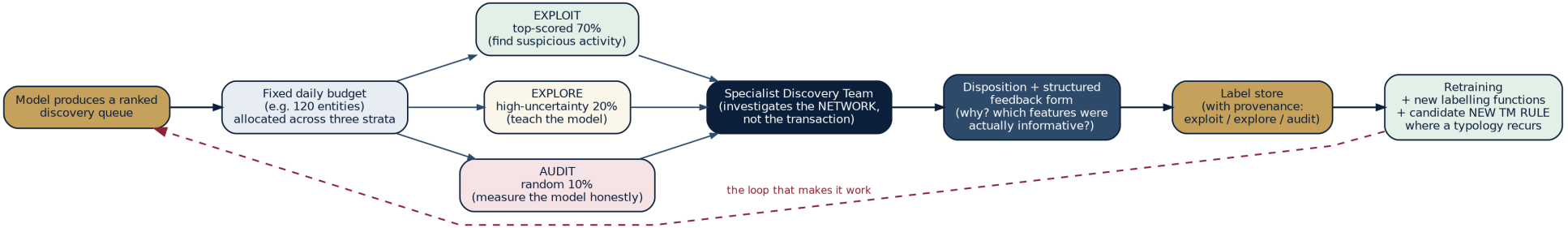

Figure 9 — The human-in-the-loop cycle. The three-way budget split (exploit / explore / audit) is what keeps the system from becoming epistemically closed.

11.1 Score Fusion

Score fusion

```
# RULE 1: never average raw scores from different models. An autoencoder's
# reconstruction error, a GNN's logit and a PU classifier's calibrated probability
# live on incomparable scales. Averaging them is arithmetic without meaning.

# Rank-normalise everything into [0,1] first.
R = {name: rankdata(s) / len(s) for name, s in model_scores.items()}

# --- Tier 1: unsupervised rank aggregation (robust, no labels needed)
fused_rrf = sum(1.0 / (60 + rank_of(name, e)) for name in R)      # reciprocal rank
                                                                  # fusion; the 60 is
                                                                  # a standard damping

# --- Tier 2: learned meta-model, once enough audit-stratum labels exist
#     Trained ONLY on the audit stratum (unbiased) plus explore-stratum labels.
#     NEVER trained on the exploit stratum: that would close the loop on itself.
meta = lgb.LGBMClassifier(n_estimators=300, max_depth=4)   # keep it SHALLOW and
meta.fit(R_audit_explore, y_audit_explore)                 # explainable
fused_meta = meta.predict_proba(R_all)[:, 1]

# --- Tier 3: hard overrides (deterministic, non-negotiable, above the models)
if entity.sanctions_nexus or entity.fincen_314a_match:
    priority = 'CRITICAL'          # models do not get a vote on this
if entity.employee_nexus:
    route    = 'SIU_RESTRICTED'    # bypasses the discovery queue entirely

# --- THE NOVELTY FILTER: the reason this programme exists
novel = not existing_tm_alerted(entity, window_days=365)
# Reported separately, ALWAYS. The programme's value is measured on the novel
# subset, and a fusion layer that quietly rediscovers what SAM already found is
# a fusion layer that has failed while appearing to succeed.
```

11.2 The Discovery Team

Findings from these models cannot be worked by the standard L1 alert queue, and attempting it is the most reliable way to kill the programme. An L1 analyst trained to disposition a structuring alert in 40 minutes, handed a 14-node subgraph with an embedding-anomaly score, will close it NFA — not out of negligence, but because nothing in their training, tooling or SLA equips them to do anything else. A dedicated **Discovery Team** is required.

Attribute	Standard
Composition	Senior investigators (5+ years), paired with an embedded data scientist. At MNB: 6 investigators, 2 data scientists, 1 lead.
Mandate	Investigate *structures and hypotheses*, not alerts. Their unit of work is a network or a behavioural cohort, not a transaction.
SLA	Deliberately generous — 15 business days for a network finding. A mule ring cannot be dispositioned in an afternoon, and pretending otherwise produces false negatives at scale.
Tooling	Interactive graph exploration; 314(b) request workflow; external-data access (registries, media, trade); ad-hoc query rights over the lakehouse.
Feedback obligation	**Structured** disposition, not free text: which features were actually informative, which were noise, what the true explanation was, and — where the finding was benign — *why*. This is the training signal, and an unstructured 'closed, no concern' is worth nothing to the model.
The rule harvest	Where a discovery recurs across several unrelated entities, the Discovery Team writes a **candidate TM rule** and hands it to AML Model Governance. **This is how the unknown becomes known** — and it is the mechanism by which the programme permanently improves the rules engine rather than merely running alongside it.

The output nobody expects

The most durable deliverable of a discovery programme is not the alerts it produces. It is the **new rules it writes for the incumbent TM system**. A typology discovered by an unsupervised model, validated by investigators and codified as a SAM scenario, becomes a permanent, cheap, explainable control that runs every night forever. Measure this. 'Number of new production scenarios originated by the Discovery Team' is a KPI that will win the programme's budget battles more effectively than any precision metric.

12. Evaluation Without Ground Truth

You cannot compute recall, because you do not know what you missed — that is the entire premise of the programme. Evaluation must therefore be built from metrics that are meaningful under that constraint, and the team must be ruthlessly honest that none of them is a substitute for the recall it cannot compute.

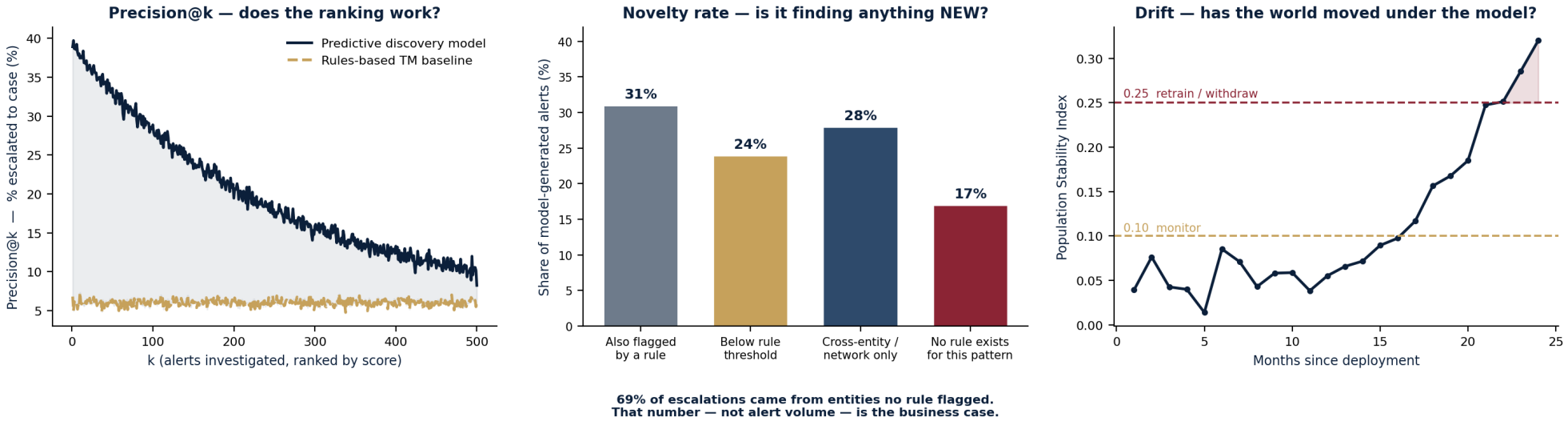

Figure 10 — The three evaluation questions: does the ranking work (precision@k), is it finding anything genuinely new (novelty rate), and has the world moved underneath it (drift).

12.1 The Metric Set

Metric	Definition	Why it matters	Target
Precision@k	Share of the top-k ranked entities that an investigator escalates to a case	The ranking's core competence. k is set by the investigator budget, not by statistics — precision at k=120/day is the only k that means anything operationally	≥ 20% at k = 120
Novelty rate	**Share of escalations where NO existing TM rule fired on the entity in the prior 12 months**	**The programme's reason to exist.** A model with 40% precision and 0% novelty has cost the bank money and taught it nothing	≥ 60%
SAR conversion	Share of model-originated cases that result in a SAR	The hardest currency available. Lags by months	≥ 8%
Lift over rules	Precision@k of the model ÷ precision of the incumbent SAM alert population	The comparison the Steering Committee will actually ask for	≥ 3×
Back-test recall	Of SARs filed in a held-out future period, what share did the model rank in its top-k *before* the alert that led to them?	The closest available proxy for true recall. Retrospective and imperfect, but it is a real number	≥ 45%
Time-to-detection lift	Median days by which the model surfaced an entity ahead of the rule that eventually caught it	Earlier detection = less laundered value. Under-used and highly persuasive to a board	≥ 30 days
Audit-stratum yield	Escalation rate in the 10% random audit sample	The bank's only unbiased estimate of the base rate. Also the early-warning signal for a systematic blind spot	Monitored, not targeted
Coverage estimate (c)	PU-derived label frequency	An estimate of what proportion of suspicious entities the bank currently catches at all	Reported to the Board
Stability (PSI)	Population Stability Index on feature and score distributions	Drift detection	< 0.10 monitor; > 0.25 retrain
Rule harvest	New TM scenarios originated by discovery findings	Permanent capability transfer into the incumbent system	≥ 4 / year

12.2 Back-Testing Honestly

Back-testing protocol

```
# The ONLY defensible back-test protocol. Everything else lies to you.
#
# 1. TEMPORAL SPLIT. Never random.
#      train  : 2019-01 .. 2023-12
#      valid  : 2024-01 .. 2024-12
#      test   : 2025-01 .. 2025-12      (strictly after everything else)
#
#    A random split lets the model see the same mule ring in train and test. You
#    will report a precision you will never see again, and you will be believed.
#
# 2. POINT-IN-TIME FEATURES. The feature vector for entity e on date t must contain
#    NOTHING that was not knowable on date t -- including the graph, which must be
#    snapshotted, and the watch-list, which must be versioned.
#
# 3. THE SIMULATION QUESTION. Do not ask 'can the model classify known SARs?'.
#    Ask the operational question:
#
#      'If we had run this model every day of 2025 with a budget of 120 entities,
#       investigated exactly those 120, and had NO other information --
#       what would we have found, and when?'
#
#    Then measure:
#       (a) how many of 2025's actual SARs appear in the model's top-120,
#           and HOW MANY DAYS BEFORE the rule that actually caught them;
#       (b) how many top-120 entities were NEVER alerted by any rule and were
#           never investigated -- these are the candidate discoveries, and they
#           MUST be manually investigated now, retrospectively, to be counted;
#       (c) the volume of value that flowed through group (b) between the model's
#           would-be detection date and today. That number, in dollars, is the
#           business case, and it is the only sentence of this report the
#           Executive Committee will remember.
#
# 4. SYNTHETIC INJECTION (red-teaming). Generate synthetic laundering structures
#    from published typologies -- a fan-in mule ring, a sub-threshold structuring
#    cohort, a PF procurement chain -- inject them into a copy of the production
#    data, and measure whether the model surfaces them. This is the closest thing
#    to a controlled experiment available in this field, and it is the only way to
#    measure RECALL on a typology you have never actually seen.
```

The one number that wins the argument

Item 3(c) above. Not precision, not AUC, not lift. *“Between the date this model would have surfaced these 43 entities and today, $310 million flowed through them, and we would have seen it on average 74 days before our current system did.”* Compute it, verify it, and lead with it.

13. Model Risk Governance

Everything in this book is a model under SR 11-7. The absence of a label does not make an unsupervised algorithm not a model, and the fact that its output is 'only' a prioritisation does not reduce its materiality — because prioritisation determines whether a human being ever looks at a customer, and a customer nobody looks at is a customer nobody files on.

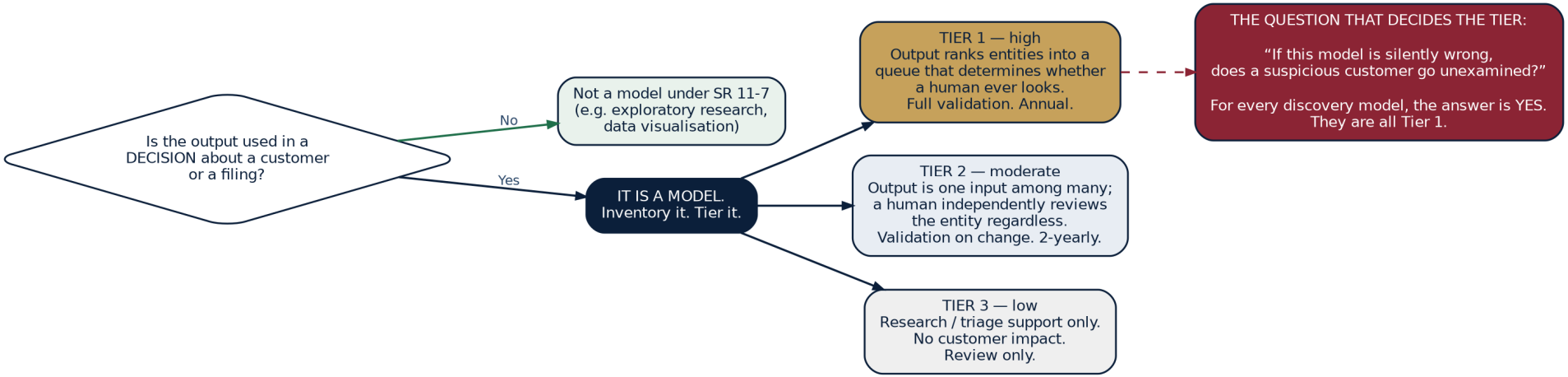

Figure 11 — Model tiering. The question that decides the tier is not how complex the model is; it is what happens if it is silently wrong.

13.1 Validating a Model With No Ground Truth

This is genuinely hard, and validators are entitled to be uncomfortable. The honest position is that conventional outcomes analysis is unavailable, and that its absence must be compensated for by greater rigour everywhere else.

Validation element	How it is done when there is no ground truth
Conceptual soundness	Carries proportionally more weight than for a supervised model. Does the feature set have a defensible causal story linking it to the typology? Is the peer definition sensible? Would a determined adversary who read this document trivially evade the model — and if so, what is the compensating control?
Stability	Bootstrap the training data and retrain. If the top-k set changes materially across bootstrap replicates, the model is fitting noise and must not be deployed. This is the single most informative test available for an unsupervised model, and almost nobody runs it.
Synthetic recall	Injection testing against typologies generated from FATF/FinCEN/Egmont publications. Reportable as a recall number on *known* typologies, which is not the same as recall on *unknown* ones — and the validation report must say so explicitly.
Benchmarking	Against a deliberately simple challenger (e.g. a robust Mahalanobis distance on twelve hand-picked features). If the deep model cannot beat that on precision@k, deploy the simple one — it is cheaper, faster and vastly easier to explain.
Explainability sufficiency	A validator sits with an investigator and works ten model outputs. If the investigator cannot articulate, from the explanation alone, what the model is claiming and how they would falsify it, the model fails validation regardless of its metrics.
Fairness / disparate impact	Test score distributions across protected classes and their proxies (ZIP code, name-origin inference, branch geography, remittance corridor). **Corridor features are the acute risk**: a model that systematically over-scores customers remitting to particular countries is a fair-lending and reputational exposure regardless of its detection performance, and 'the risk model says so' is not a defence.
Adversarial robustness	Perturb the inputs in ways an evading subject plausibly could (split value across accounts; add benign padding transactions; route through one more hop). Measure how much score decay each perturbation buys. Report the cheapest evasion found.
Ongoing monitoring	PSI on features and scores; precision@k decay; novelty-rate decay; audit-stratum yield. **Novelty-rate decay is the characteristic failure mode**: as the model's discoveries are codified into rules, it begins rediscovering things the rules now catch, and its marginal value silently approaches zero while every other metric looks healthy.

13.2 The MLOps Lifecycle

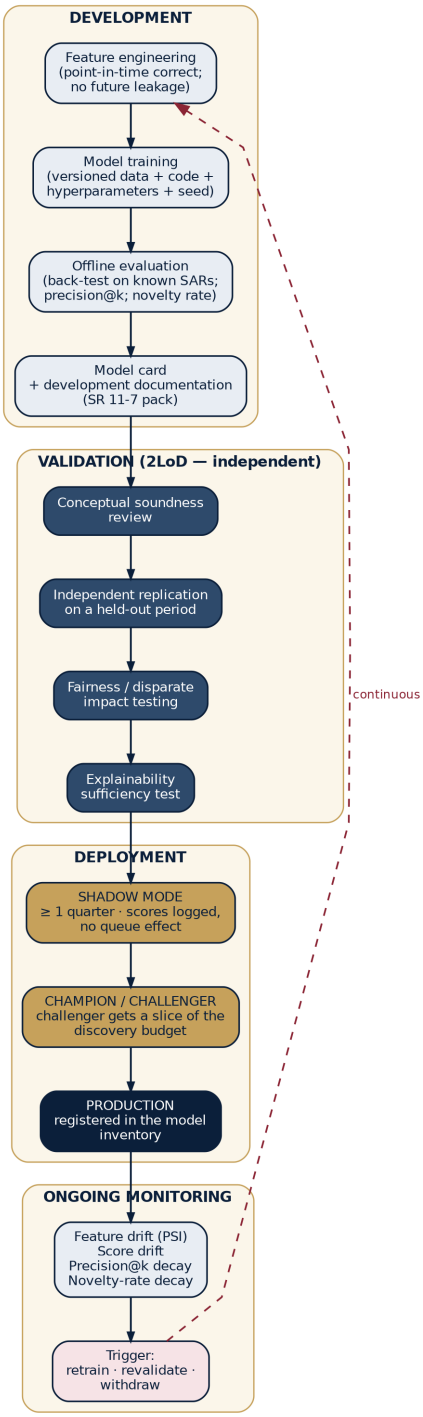

Figure 12 — The model lifecycle: development, independent validation, shadow deployment, champion/challenger, production and continuous monitoring.

Control	Requirement
Reproducibility	Every model artefact is pinned to a data snapshot hash, a code commit, a hyperparameter set and a random seed. A validator must be able to rebuild the exact model from the registry alone, two years later.
Shadow mode	Minimum one full quarter. Scores are computed and logged nightly with no effect on any queue. This produces the evidence base for the validation without exposing the bank to an unvalidated model.
Champion / challenger	The challenger receives a fixed slice of the discovery budget (MNB: 15%). Promotion requires superior precision@k **and** superior novelty rate, over at least one quarter, with committee approval.
Model inventory	Every model — including every unsupervised detector in the ensemble — is separately inventoried, tiered and owned. An 'ensemble' is not one model for governance purposes; it is n models plus a fusion model.

Control	Requirement
Kill switch	Any model can be withdrawn from the fusion layer within one batch cycle, by a named owner, without a code deployment. Test this quarterly.
Documentation	A model card per model (Annex B), a development document per model, and a validation report per model. If it is not written down, it does not exist.

13.3 The Boundary That Must Not Move

The architectural constraint that makes all of this defensible

No model output in this book may, on its own, cause an adverse action against a customer. Not an exit, not a restriction, not a de-risking, not a filing. Every model output does exactly one thing: it decides *where a human looks next*. The human then investigates, gathers evidence, and makes a decision on the evidence — and the model's score is **not part of that evidence** and never appears in a SAR narrative.

This is not defensive timidity. It is the constraint that makes an unsupervised, hard-to-validate, black-box-ish model system *lawful, examinable and fair*. Relax it — let a model score restrict an account, or let it stand as a reason for a filing — and you have built an unvalidatable adverse-action engine, with all the fair-lending, UDAAP and reputational exposure that implies. Every reviewer of this framework should treat any proposal to relax this boundary as the most serious risk on the table.

14. Delivery

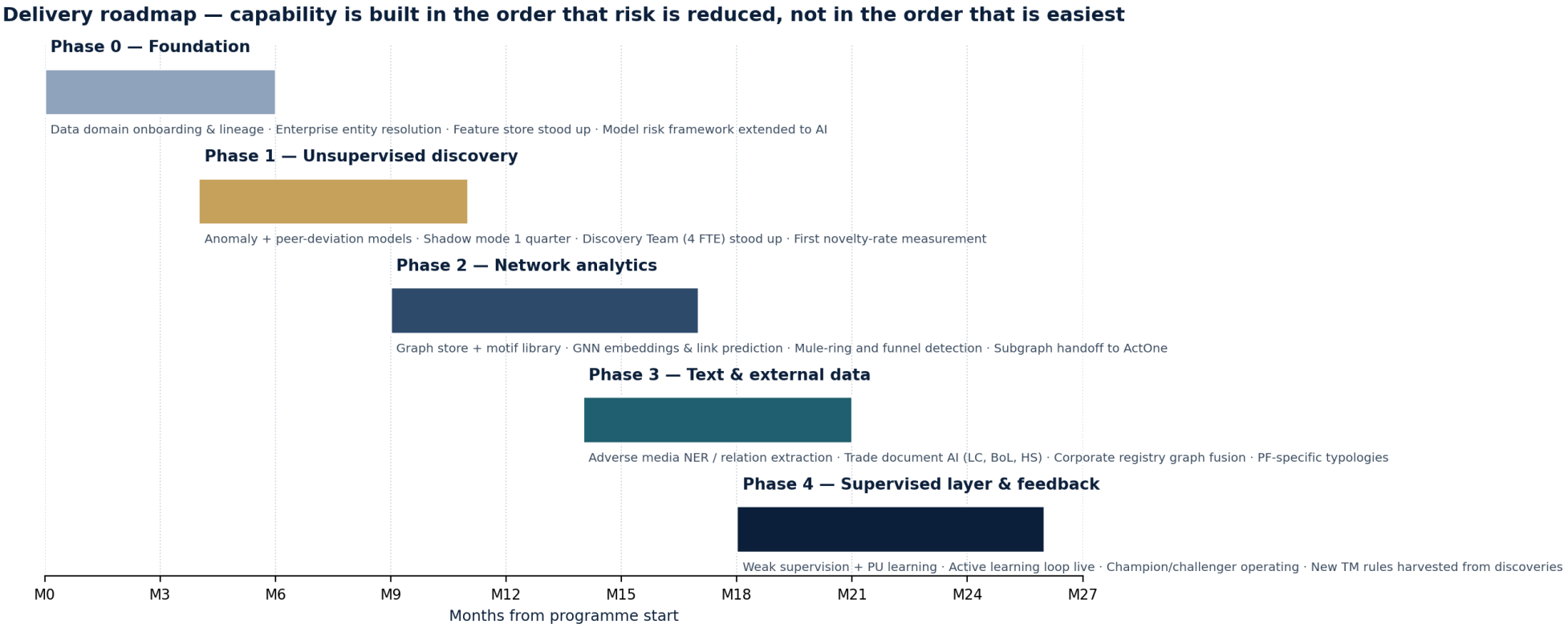

Figure 13 — A 26-month roadmap. Capability is sequenced by risk reduction per unit of effort, not by technical appeal.

14.1 Sequencing Principles

- **Do the JOIN before the neural network.** The identity-sharing query in §4.2.2 and the motif queries in §6.2 are deterministic SQL. At every pilot the author is aware of, they have out-delivered the first year of model development — and they can ship in a month.
- **Entity resolution before everything.** A model over a badly-resolved graph is a sophisticated way of being wrong.
- **Unsupervised before supervised.** The unsupervised layer generates the labels the supervised layer will need, and it is the only layer immune to verification bias.
- **Stand up the Discovery Team in Phase 1, not Phase 4.** Models with no one to work their output produce nothing, and a programme that produces nothing for eighteen months will be cancelled in month twelve — correctly.
- **Instrument the novelty rate from day one.** It is the only metric that distinguishes this programme from an expensive re-implementation of the rules you already own.

14.2 Team and Cost

Function	FTE	Why
Data engineering	6	Feeds, lakehouse, entity resolution, graph store, feature store. The largest and least glamorous line, and the one that determines success.
Machine-learning engineering	4	Model development, training infrastructure, MLOps.
Financial-crime SMEs embedded in the ML team	2	**Non-negotiable.** A data scientist without a launderer's intuition will build a beautiful outlier detector that finds hospitals. Feature design is domain work performed by domain experts.
Discovery Team (investigators)	6	Work the findings. Structured feedback. Harvest new rules.
Model validation (2LoD)	2	Independent. Cannot report through the same chain as the developers.
Product / programme lead	1	Owns the novelty rate, the budget and the governance narrative.
Total	**21**	Plus infrastructure: graph store, GPU training, external data licences (trade, shipping, registries, media) — the external data licences are typically the largest single line item and the one most likely to be cut. Cutting them removes PF and TBML detection entirely.

14.3 Failure Modes

Failure mode	Symptom	Prevention
The pilot that never lands	18 months of notebooks; nothing in production	Ship the deterministic queries in month 2. Earn the right to build models.
Verification-bias trap	Beautiful AUC on SAR labels; zero novel discoveries	Unsupervised core. Instrument novelty rate from day one.
The anomaly-detector-that-finds-hospitals	Analysts stop working the queue; precision collapses	Peer-conditional models; SME-driven feature selection; kill any detector below 5% precision@k within a quarter.
Handing networks to L1	Everything closed NFA in 40 minutes	Dedicated Discovery Team with a 15-day SLA and graph tooling.
The audit stratum gets cut	Metrics look great; the model is measuring itself	Write it into the approved operating procedure. Removing it requires model-committee approval.
Governance ambush	MRM blocks deployment at month 20 over an unvalidatable model	Engage MRM in month 1. Co-author the validation approach for unsupervised models *before* building them.
Silent decay	Novelty rate drifts to zero as discoveries become rules; nobody notices	Novelty rate is a monitored KPI with a withdrawal trigger, exactly like PSI.

Failure mode	Symptom	Prevention
Boundary creep	Somebody proposes using the score to auto-exit customers	Refuse. See §13.3. This is the failure mode with the largest tail risk in the entire programme.

Annexures

Annex A — Feature Catalogue (extract)

Feature	Definition	Windows	Family	Gap
cash_ratio	cash credits ÷ total credits	30/90/180d	Value	G1
amount_entropy	normalised Shannon entropy of log-binned amounts	90/365d	Distributional	G1, G4
benford_divergence	KL divergence of the first-digit distribution from Benford	365d	Distributional	G1
round_ratio	share of amounts divisible by 1,000 with no cents	90d	Distributional	G1
threshold_discontinuity_D	(count just below T – count just above T) ÷ sqrt(sum)	365d	Distributional	**G1**
burstiness	(σ – μ) ÷ (σ + μ) of inter-arrival times	90d	Velocity	G1
flow_through_pct	debits ÷ credits	10/30d	Velocity	G1
avg_balance_ratio	mean daily balance ÷ throughput	30d	Velocity	G1
median_hold_days	median days from credit to the debit that consumes it (FIFO)	30d	Velocity	G1
cpty_herfindahl	Herfindahl concentration over counterparty value share	90d	Counterparty	G1
cpty_churn	share of counterparties appearing for the first time	90d	Counterparty	G1
new_bene_value_share	share of outbound value to first-time beneficiaries	30d	Counterparty	G1, TF
peer_z_*	z-score of any feature within the learned peer segment	all	Peer	**G1**
peer_q95_residual_*	max(0, feature – peer 95th percentile) ÷ σ	all	Peer	G1
bocpd_score	P(change point) from Bayesian online change-point detection	365d	Temporal	**G3**
seq_perplexity	perplexity of the entity's transaction sequence under the transformer	365d	Temporal	G3, G4
embed_self_distance	distance between the current and trailing behavioural embedding	90d vs 365d	Temporal	G3
pagerank_w	weighted PageRank on the payment graph	snapshot	Network	G2
betweenness	betweenness centrality (approximate, sampled)	snapshot	Network	G2
community_id / size	Leiden community assignment and its size	snapshot	Network	G2
dist_to_flagged	shortest path to any SAR'd / sanctioned / 314(a) / confirmed-fraud entity	snapshot	Network	**G2**
n_hop2_flagged	count of flagged entities within 2 hops	snapshot	Network	G2

Feature	Definition	Windows	Family	Gap
motif_flags	membership of fan-in / fan-out / cycle / chain / bipartite-core motifs	monthly	Network	**G2**
gnn_embed_anomaly	distance from the peer centroid in GNN embedding space	snapshot	Network	**G2, G4**
device_share_degree	count of other parties sharing a device fingerprint	snapshot	Identity	**G2**
address_share_degree	count of other parties sharing an address (allowlist-filtered)	snapshot	Identity	G2
geo_velocity_impossible	count of login pairs implying impossible travel	30d	Identity	G3
narrative_specificity	learned specificity score of remittance free text	90d	Text	G4
goods_naics_coherence	cosine similarity between the goods description embedding and the NAICS embedding	per txn	Text	TBML
registry_age_days	age of the counterparty's corporate registration	snapshot	External	PF
director_share_degree	count of entities sharing a director with the counterparty	snapshot	External	**PF**
hs_dual_use_flag	HS code appears on a dual-use control list	per txn	External	**PF**
price_deviation	|unit price – world reference price| ÷ reference price	per txn	External	TBML
vessel_ais_gap_hrs	longest AIS dark period on the carrying vessel	per shipment	External	**PF**
sanctioned_graph_dist	graph distance from the counterparty to any designated entity	snapshot	External	**PF**

Annex B — Model Card Template

```
MODEL CARD
==========
Model ID / name / version      :
Owner (named individual)       :
Model risk tier                :  Tier 1 (all discovery models -- see Ch. 13)
Inventory reference            :

PURPOSE
  Gap class addressed          :  G1 / G2 / G3 / G4
  Question the model answers   :  (one sentence, in domain language)
  What it explicitly does NOT  :  (e.g. 'does not estimate criminality; does not
                                   support adverse action; output is inadmissible
                                   as evidence in a SAR narrative')

DATA
  Training window              :
  Feature set (n, families)    :
  Point-in-time attestation    :  [ ] leakage checklist passed, build hash:
  Excluded features + reason   :  (protected characteristics and their proxies)
  Known data quality limits    :

METHOD
  Algorithm / architecture     :
  Hyperparameters + seed       :
  Code commit / data snapshot  :
  Simple challenger benchmarked:  (and its performance)

PERFORMANCE
  Precision@k (k = budget)     :
  NOVELTY RATE                 :  <-- the headline number
  SAR conversion               :
  Lift over incumbent rules    :
  Back-test recall / lead time :
  Synthetic-injection recall   :  (by typology)
  Bootstrap stability of top-k :

FAIRNESS
  Protected-class score deltas :
  Proxy features tested        :  ZIP, corridor, name-origin, branch geography
  Disparate impact conclusion  :

ROBUSTNESS
  Cheapest evasion found       :  (and its cost to the adversary)
  Compensating control         :

EXPLAINABILITY
  Method                       :  (SHAP / per-feature reconstruction error /
                                   attention weights / counterfactual edges)
  Investigator sufficiency test:  [ ] passed -- 10 outputs, 2 investigators

MONITORING
  Metrics + thresholds         :
  Withdrawal triggers          :  PSI > 0.25 | precision@k < X | novelty < Y
  Kill-switch owner            :

GOVERNANCE
  Developed by / date          :
  Validated by (2LoD) / date   :
  Approved by / date           :
  Next revalidation due        :
```

Annex C — Validation Checklist for Unsupervised Models

#	Check	Pass condition
1	Is the model in the inventory and tiered?	Yes — every detector separately, plus the fusion model
2	Point-in-time correctness attested?	Leakage checklist in the build log; build fails on any hit
3	Temporal train/valid/test split?	No random splits anywhere
4	Bootstrap stability of the top-k set?	Jaccard overlap ≥ 0.70 across replicates
5	Beaten a simple challenger?	If not — **deploy the challenger**
6	Synthetic-injection recall measured?	Reported by typology, with the caveat stated
7	Explainability sufficiency test passed?	Two investigators, ten outputs, both can falsify
8	Disparate impact tested, proxies included?	Corridor, ZIP, name-origin explicitly tested
9	Adversarial perturbation tested?	Cheapest evasion documented
10	Novelty rate measured and reported?	≥ 60%; below 40% the model's value is in question
11	Kill switch tested this quarter?	Evidence in the operations log
12	Audit stratum operating at ≥ 10%?	Evidence in the sampling log
13	No model output used for adverse action?	Attested by the FIU head, in writing, annually
14	MRM engaged before build, not after?	Validation approach agreed in the design phase

Annex D — Reference Feature Build (PySpark)

```
from pyspark.sql import functions as F, Window as W

# Rolling entity-day features. This is the workhorse job -- it runs nightly over
# ~48M postings and materialises the feature store's offline tables.

w90 = (W.partitionBy('party_id').orderBy(F.col('txn_dt').cast('long'))
         .rangeBetween(-90*86400, 0))

feat = (txn
  .filter(F.col('exclude_flag') == 'N')
  .withColumn('is_cash',  (F.col('txn_subtype') == 'CASH').cast('int'))
  .withColumn('is_cr',    (F.col('drcr') == 'CR').cast('int'))
  .withColumn('log_amt',  F.log1p('base_amt_usd'))

  # --- value & velocity
  .withColumn('cr_90d',      F.sum(F.col('base_amt_usd')*F.col('is_cr')).over(w90))
  .withColumn('dr_90d',      F.sum(F.col('base_amt_usd')*(1-F.col('is_cr'))).over(w90))
  .withColumn('cash_cr_90d', F.sum(F.col('base_amt_usd')*F.col('is_cash')
                                   *F.col('is_cr')).over(w90))
  .withColumn('cnt_90d',     F.count('*').over(w90))
  .withColumn('cash_ratio',  F.col('cash_cr_90d') / F.greatest(F.col('cr_90d'),
                                                               F.lit(1.0)))
  .withColumn('flow_through', F.col('dr_90d') / F.greatest(F.col('cr_90d'),
                                                           F.lit(1.0)))

  # --- counterparty structure
  .withColumn('cpty_cnt',    F.size(F.collect_set('counterparty_id').over(w90)))
  .withColumn('cpty_churn',  F.col('cpty_cnt') / F.greatest(F.col('cnt_90d'),
                                                            F.lit(1)))

  # --- distributional shape: the family that finds sub-threshold behaviour
  .withColumn('round_cnt',   F.sum(((F.col('base_amt_usd') % 1000 == 0) &
                                    (F.col('base_amt_usd') > 0)).cast('int')).over(w90))
  .withColumn('round_ratio', F.col('round_cnt') / F.greatest(F.col('cnt_90d'),
                                                             F.lit(1)))
  .withColumn('near_thresh_cnt',
        F.sum(F.col('base_amt_usd').between(9000, 9999).cast('int')).over(w90))
  .withColumn('above_thresh_cnt',
        F.sum(F.col('base_amt_usd').between(10000, 11000).cast('int')).over(w90))
  .withColumn('discontinuity_D',
        (F.col('near_thresh_cnt') - F.col('above_thresh_cnt')) /
         F.sqrt(F.greatest(F.col('near_thresh_cnt') + F.col('above_thresh_cnt'),
                           F.lit(1.0))))
)

# --- amount entropy: needs a UDF; worth every cycle it costs
@F.pandas_udf('double')
def amount_entropy(amts: pd.Series) -> float:
    if len(amts) < 8: return float('nan')
    h, _ = np.histogram(np.log1p(amts), bins=32)
    p = h[h > 0] / h.sum()
    return float(-(p * np.log2(p)).sum() / np.log2(32))

ent = (txn.groupBy('party_id', F.window('txn_dt', '90 days', '1 day'))
          .agg(amount_entropy(F.collect_list('base_amt_usd')).alias('amount_entropy')))

# --- peer-relative: the step that converts a measurement into a RISK measurement
peer = W.partitionBy('peer_segment_id', 'obs_dt')
out = (feat.join(ent, ['party_id','obs_dt'])
  .withColumn('peer_mean_cash', F.avg('cash_ratio').over(peer))
  .withColumn('peer_sd_cash',   F.stddev('cash_ratio').over(peer))
  .withColumn('z_cash', (F.col('cash_ratio') - F.col('peer_mean_cash')) /
                        F.greatest(F.col('peer_sd_cash'), F.lit(1e-6))))

out.write.format('delta').mode('overwrite').partitionBy('obs_dt') \
   .save('/gold/features/party_daily')
```

Annex E — MIS Pack: What to Put in Front of the Board

Slide	Content	Why
1	**Novelty:** 'This quarter the models surfaced 61 entities that no rule had ever alerted on. 19 became cases. 7 became SARs. $180m had flowed through them.'	This is the programme. Lead with it, every time.
2	**Lead time:** 'Where the rules eventually did catch an entity, the models saw it a median of 74 days earlier.'	Converts detection into avoided exposure — the language the Board thinks in.
3	**Coverage estimate:** the PU-derived *c*, with its caveats stated plainly.	Uncomfortable, honest, and the single most valuable thing the analytics team can tell the Board.
4	**Rule harvest:** new SAM scenarios originated by discovery findings, now running in the incumbent system.	Permanent capability transfer. This is what makes the spend compounding rather than recurring.
5	**Model health:** precision@k, PSI, audit-stratum yield, novelty-rate trend, and any withdrawal triggers hit.	Governance credibility. Report the bad quarters.
6	**Gaps that remain:** typologies the models still cannot see, and what data would be required.	A programme that reports no remaining gaps is not being honest, and the Board should be told which risks it is still carrying.

Closing

The rules-based system tells you about the crime you already know how to describe. Everything in this book exists to answer a harder and more uncomfortable question: **what is moving through this bank that we have never learned to look for?**

That question has no clean metric, no ground truth and no completion date. It has only a discipline: build models that carry no inherited assumptions, hand their findings to people who know how to investigate a structure, measure yourself on what is *new*, keep an honest random sample so the system can always be told it is wrong — and never, under any pressure, let a model score become a reason for a decision about a human being.

Prepared by AML Base LLC. Meridian National Bank, N.A. is fictional; all volumes, thresholds and performance figures are illustrative and calibrated for instructional realism.