NVIDIA FLARE DAY 2026 · UPDATED 23 SEPTEMBER 2026

Catching cross-bank money laundering with federated graph learning

Henrik Axelsen · Founder, Node16.ai · Postdoc, Copenhagen Business School

henrik@node16.ai · node16.ai

THE PROBLEM

A societal problem shaped wrong for those policing it..

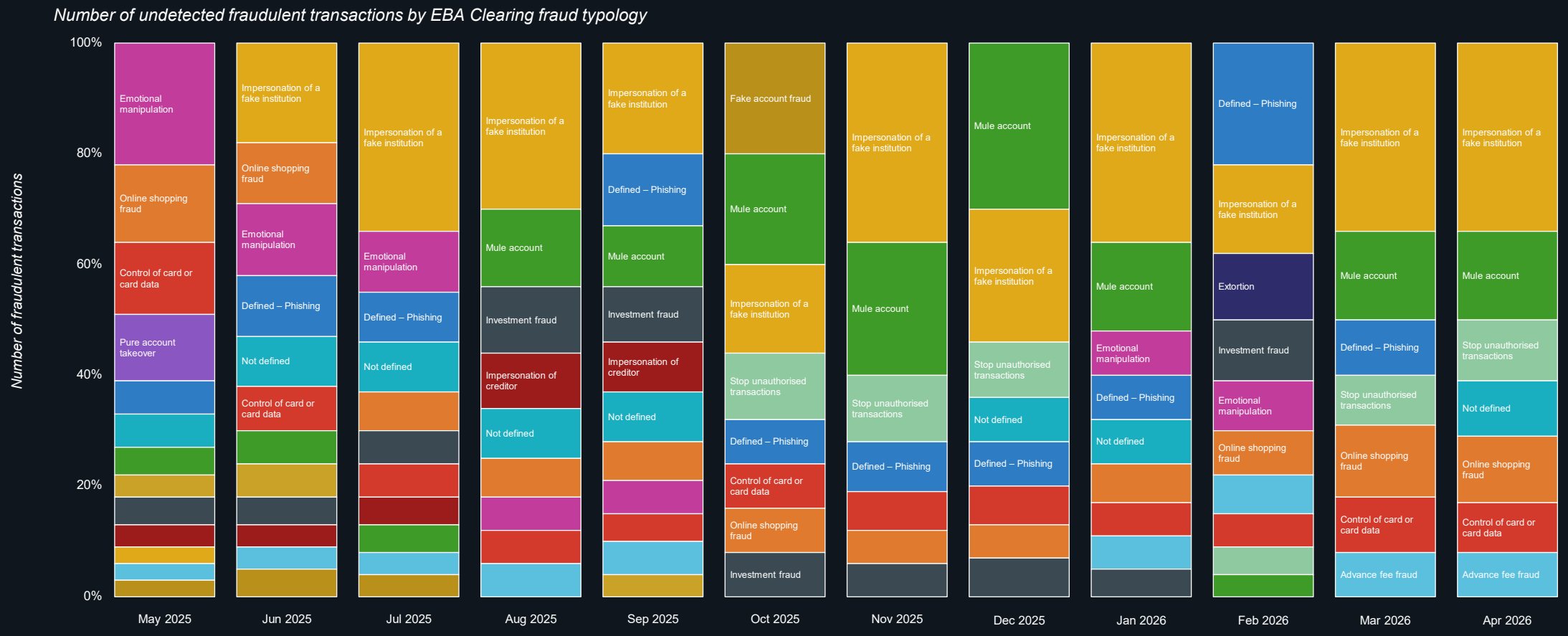
- **Money laundering is mostly a network crime**
  - Layering chains, mule webs, trade-based loops etc designed to cross institutions
  - App $4T laundered annually, <1% caught
- **Detection is institution-bound**
  - Each bank monitors its own slice and files its own reports, >90% FP
- **Every bank sees a locally-legitimate fragment**
  - The pattern only exists in the union – **placement** / introduction in one bank, **layering** and movement through other banks, **integration** and cash-out elsewhere
- **Pooling raw data is not allowed**
  - Bank secrecy, competition law, GDPR; a central financial panopticon / mass surveillance is the wrong answer. Hence federated learning relevance
- Criminals largely go free
  - Banks spend >$200Bn pe year, mostly human effort for KYC and clearing false positives + get fined if they miss reporting a suspicious transaction



The "global" business model of banking suggests an inherent risk assessment must find many patterns in fragmented (small) local datasets, where the problem is invisible

THE PROBLEM

Fraudsters are agile and adapt – EBA clearing example



Source: Courtesy of Spuerkeess /Banque et Caisse d'Épargne de l'État, Luxembourg (BCEE) @ EBA Clearing, June 2026, EBAday

Five banks learn one detector from local inputs

- This case is broad AML, not fraud
  - For those of you modelling fraud, typologies overlap around 10%
- Simulating 80% of Danish banking
- Matched local and federated full-GNN training with the same bank-local graph and temporal context
- Federated GNN with richer local context; secure aggregation and DP remain production gates
- Four retained investigation windows per bank
- Synthetic AML labels across the 187-template experimental universe.
- Shared-host execution simulates separate banks; 85 rounds completed, stopped as gains flattened

5
“bank" participants

85 × 50
rounds × local epochs

1.2 TB
synthetic corpus

3 × B200
Sovereign host

187
typologies

4m
customers

SYNTHETIC DEVELOPMENT VALIDATION: independent test and real-bank detection unmeasured

Federation improves PR-AUC in every synthetic bank

5/5

banks improve with federation. PR-AUC gain: +0.046 to +0.080

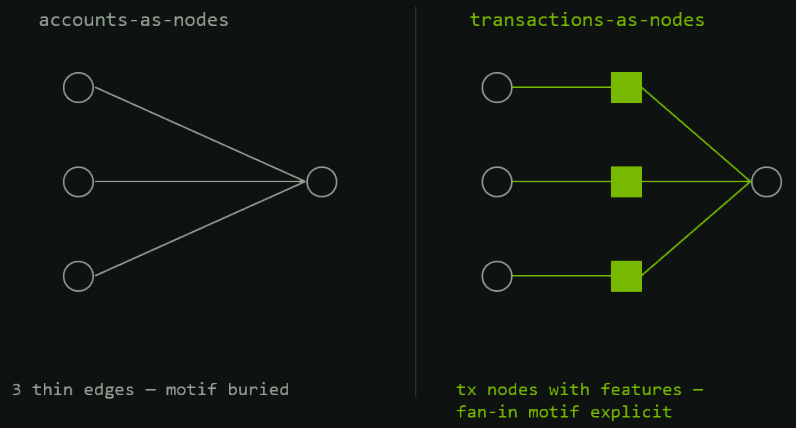
Bank	PR-AUC: local → federated
A	0.689 → 0.735
B	0.678 → 0.742
C	0.708 → 0.775
D	0.721 → 0.776
E	0.668 → 0.748

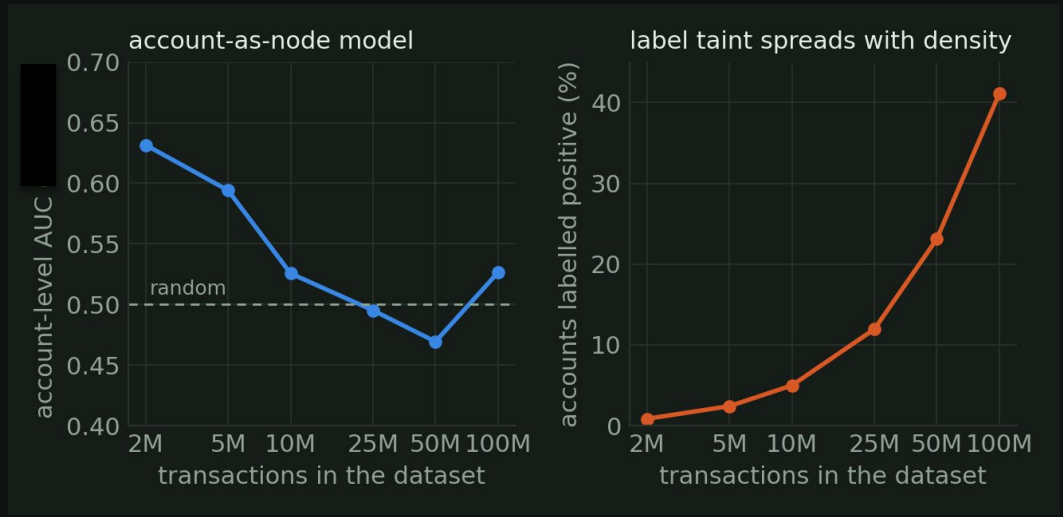
Same per-bank development-validation rows and local inputs. Thresholds are not independently tested. Round 85 was booking-limited; round 80 had slightly higher PR-AUC.

DESIGN CHOICE #1

Transactions as nodes, not edges

- Accounts-as-nodes crushes the signal: amount, timing, channel become edge features; multi-edges collapse
- Bipartite graph: a payment is a first-class node; accounts connect through it
- Typology shapes – fan-in collection, circular layering – become subgraph motifs a GNN can see
- The price: node count explodes into the billions. Every scale decision downstream follows

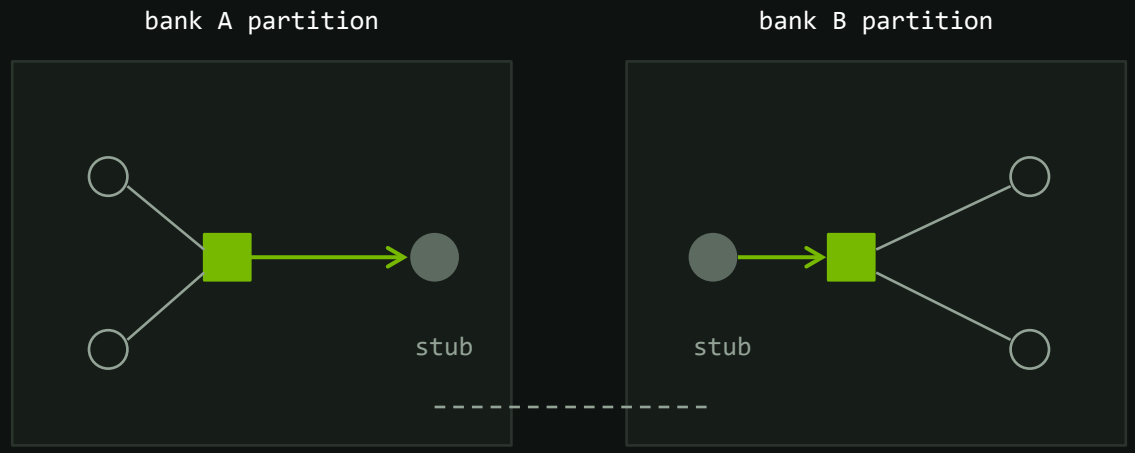




DESIGN CHOICE #2

Hermetic bank partitions, pseudonymous stubs

- One site = one bank = one hermetic partition enforced by runtime checks. Jobs refuse to start on a boundary violation
- The far end of a cross-bank payment is a pseudonymized counterparty stub: coarse properties, never an identity
- Cross-bank identity / behavioral footprint is a separate, governed, human-authorized mechanism based on recipient-disclosure rungs. Learning and linkage never share a channel.
- 80%+ of the largest synthetic bank's payments are cross-bank – stubs are the norm



Key assumption: Money laundering is mainly a networked financial crime problem

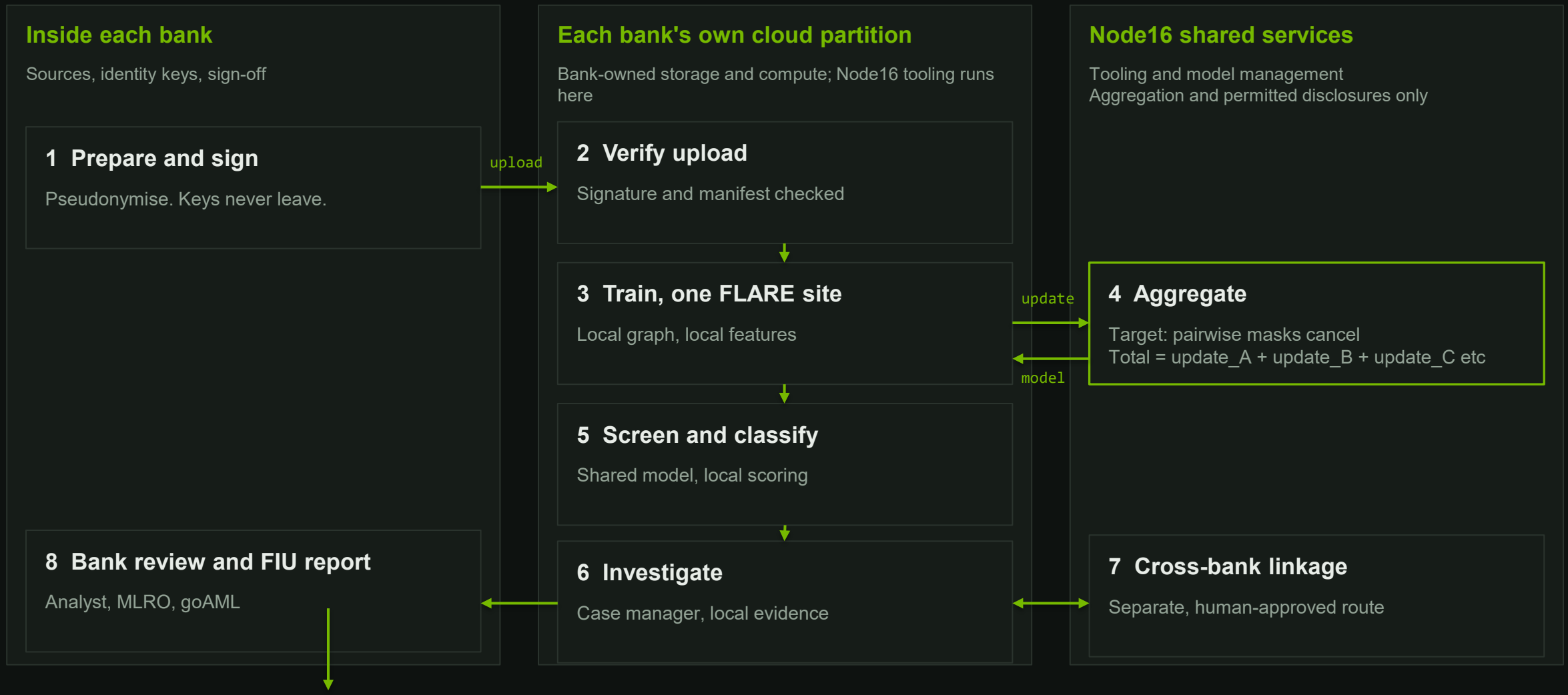
DESIGN CHOICE #3

The model collapse you won't see in the loss curve

- Averaging across banks with different class balances and degree distributions quietly cancels the minority signal
- At ~1% laundering prevalence, a collapsed model looks great on loss
- If the quality gate reads the wrong round, you ship the collapsed model
- **Fixes include** shared normalization invariants, degree caps, gate metrics keyed to the evaluated aggregate round, not the local epoch



Can we solve a societal problem through smarter collaboration?



Production gates: secure aggregation and recipient privacy

A smaller queue can be highly precise

Federation beats matched local training in all five synthetic banks at alert budget. The operating point sets the trade-off between detection and investigation workload

Transaction flagged rate	Precision (of flagged, % suspicious)	False (of flagged, % good)	Recall (% true bad found in dataset)
0.5%	42.7–64.4%	35.6–57.3%	75.2–77.4%
0.1%	98.4–99.5%	0.5–1.6%	23.2–35.5%
0.01%	100.0%	0.0%	2.3–3.6%

How do SAR/STR reports help connect networks, enrich intelligence or advance investigations ?

Real (Danish) world calibration of density, “true” positives and alert rates could improve the synthetic setup.

Calibration work progressing as part of sandbox preparations

NVIDIA at the core of combating money laundering

- Start in Denmark: develop Node16's bank-led AML case alongside the Danish FSA's priority for FT Lab data-sharing partnerships.
- Put NVIDIA technology at the core: FLARE for federated coordination and GPU computing for graph learning across bank partitions.
- Build on measured progress: 85 rounds on three B200 GPUs, with a federation advantage in all five synthetic bank partitions.
- Create a path across the EU: move from academic HPC to bank-trusted hosting on Scaleway or Polarise, then extend the model to other banks.

A Danish reference case for NVIDIA-powered AML collaboration across Europe, with potential for wider adoption

Thank you

Catching cross-bank money laundering with federated graph learning

From Danish validation to an environment banks can trust.

Henrik Axelsen · Founder, Node16.ai

henrik@node16.ai · node16.ai