PROJECT

Night's Watch

Catching a portfolio company's cash trouble before it becomes a missed payment.

Real-time cash monitoring & early-warning for the Credit team
“First to see the winter coming”

Alexandre Cela · Product assignment

EXECUTIVE SUMMARY

Night's Watch in one page

Night's Watch protects Liquidity's **near-zero credit-loss record** by spotting, in real time, portfolio companies at risk of distress — and turning each warning into a **prioritised, actionable, and adopted** intervention workflow for the Credit team.

1

The signal

Flag every loan at risk of **missing its next cash payment**.

Real-time coverage: expected transferable cash at repayment vs. cash due.

2

The workflow

Prioritised, actionable, **adopted** — save the most money at risk per day.

Ranked worst-first; Cash at risk X Recoverability prioritisation framework

3

Sharper over time

Phased **data, engineering & ML**, plus logged intervention impact.

Grounded-first; ML adjudicates only the borderline names, as labels accrue.

4

Delivered well

A **product & engineering** effort behind a tool the team trusts.

Trust-gated rollout: a grounded signal first, suite maturity as data lands.

One signal, one workflow — grounded now, sharpened in phases, and adopted by augmenting a process that already works.

OUTLINE

What we'll cover

1	**Why Night's Watch?**	How is NW helping Liquidity? By tackling which problem statement?
2	**The signal**	With which solution? 1st Part: The Signal
3	**The workflow**	2nd Part: The Loan Intervention Operating System
4	**The prototype**	How does the solution look like?— prioritised, understandable, trusted, actionable.
5	**How we deliver**	How do deliver the solution? Data roadmap, engineering & design, what can go wrong, internal rollout.
6	**How we measure success**	How does success look like? Six layers — from signal on time to the business outcome.
•	**Limitations & considerations**	What this first draft does and doesn't cover yet.
•	**Appendix**	Prototype detail · data model · medallion · recovery signal · UAE compliance.

Putting this project in perspective

The firm's goal	Generate **ever-greater returns** from ever-more private credit deals — while keeping the record that sets Liquidity apart: a **0.00% credit loss** rate since 2019
Why it matters	That near-zero loss record is one of the **firm's edge**. Liquidity's own technology already aims to be "the first to know and the first to act when any issues occur" — this project sharpens that edge for the cash-monitoring job.
Where this project fits	If it works, **Night's Watch** raises the likelihood that Liquidity keeps its loss **record intact** — by catching a company's cash trouble before it becomes a missed payment.
How we help	Spot, **ahead of time**, **ALL** loans at risk of missing their next cash interest payment — and give the **Credit team** what they need to **act**, saving as much money at risk as possible, as fast as possible.

The objective: one signal, one workflow

A Signal

Flag, in **real time**, **every** loan at risk of missing its next cash interest payment.

A workflow

Give the Credit team what they need to **act** on each warning and **save** the most money at risk per day.

How we will know we are succeeding

Type	What we measure	Constraint
Leading — warning	Share of at-risk loans flagged in time (aim: all)	Keep false alarms low, stay accurate
Leading — action	Money at risk covered by action, per day	While working through as many loans as possible
Lagging — outcome	Defaults that caught us unaware; money lost (aim: none)	—

"Defaults that caught us unaware", not "zero defaults": some companies fail however early we warn — so we promise no surprise, and the most money saved on the rest.

How do we get the signal?

A guiding principle: as much actual grounded facts as possible. Accompanied by careful and precise forward looking estimation

Ideal — but not achievable

Available cash for debt repayment at next repayment (T1) ÷ Cash to repay at T1

Not achievable because cash at a future date cannot be known — only estimated.

North star — what we tend toward

Range of expected **available** cash at T1 ÷ Cash to repay at T1

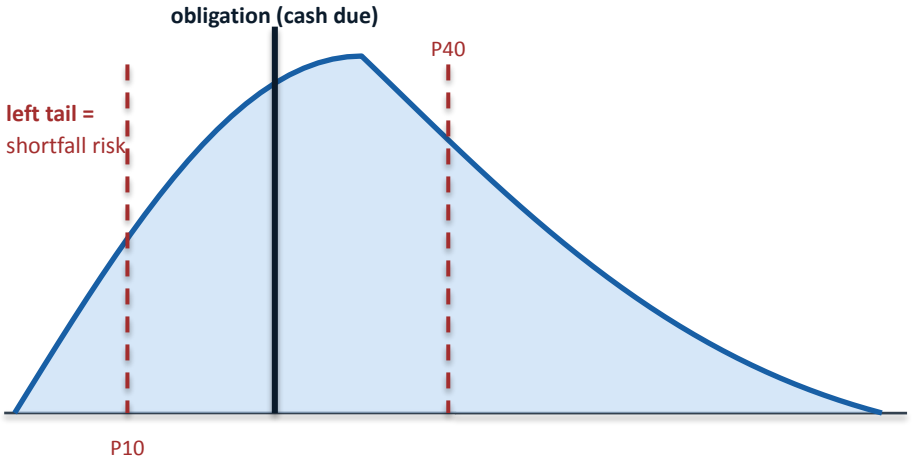
A range, this way we can tune our conservatives for recall and precision tradeoffs. Always as of T1 = NEXT cash repayment date (fallback tomorrow if not available), this way it rolls automatically for any cash repayment time

APPENDIX

Illustration Example: Liquidity-at-Risk

To be calibrated for recall first, then f1 target

The signal is a **quantile of forecast cash at repayment** — a liquidity-VaR. Where we set the threshold sets the recall/precision trade; where the obligation lands sets the action.



Where the obligation lands → the action

Obligation left of P10

No flag — even the pessimistic tail covers it. Monitor only.

Obligation between P10 and P40

Borderline — the names ML earns its keep on. Watch, verify.

Obligation right of P40

Flag — shortfall risk beyond tolerance. Triage by money × recoverability.

Obligation right of the median

Act now — more likely than not to fall short. Highest priority.

The quantile is a liquidity-VaR; an expected-shortfall view (how far below in the bad cases) sizes the money at risk. The threshold is set empirically on held-out snapshots — and owned by the Credit team as a recall floor.

How we get each piece of the metric: Data Roadmap

Metric item	Data source	Availability	Method	Conf.	Difficulty
Current transferable cash	**Company banking data**	Now	Proxy: local-currency cash likely to be exchangeable. NBFI?	4.5/5	Low
History of transferable cash	Company banking data	Now	Collect all the snapshots	4.5/5	Low
Proxy of transferable cash at T1	Company banking data	Now	Project from historical cash trend (with **burn rate + runway)**	2.5/5	Medium, mitigate with conservativ e proxy
Time of next repayment	**Loan structure** (fallback: tomorrow)	Very soon If in CRM	Read from record	4.5/5	Low
Coupon / cash to repay at T1	Loan structure	Very soon If in CRM	Read from record; flag cash vs PIK (PIK = no cash due)	4.5/5	Low
Robust forecast of cash at T1	**Cashflow statements + revenue-lifecycle data**	Later	Model inflows and outflows, not just run-down	TBD	High
Rates + input-cost forecasts (e.g. energy)	External sources	Later	Adjust floating-rate payments & margin pressure	3/5	Medium

Nice-to-have, later: rates + input-cost forecasts matter most for floating-rate loans, where they move both the payment due and the borrower's margins.

What we can build — now, soon, later

NOW

Bank files history. Loan structure in CRM

Current transferable cash ÷ cash to repay at snapshot t0 (repayment date, or tomorrow as worst case). Plus a proxy of transferable cash at T1 ÷ cash to repay, with trends input. Paired with the history of labelled missed payments per company — to calibrate the warning against what actually happened.

MEDIUM

Loan structure · balance sheet · cashflow statements

A more robust estimate of transferable cash at T1. Begin work toward an ontology and a golden early-warning data model. (More loan info, more ability to assess ‘availability for debt repayment’ cash)

LONGER

Business-lifecycle data · external rate & cost forecasts

Forecast inflows, not just cash running down (contracts pipeline, product usage). Add rates + input-cost forecasts for floating-rate loans.

What does the Credit team need to act on?

Hypothesised framework, to validate with the team: See what's at risk → understand why → prioritise → decide what to do → act and record. To validate with Credit Team. Data analysis decision making framework reverse engineering

See Triage	Output: List of companies with higher warning level. Ranked by coverage ratio + size	**Signal + loan principal (bank account size fallback)** Now *(Bank Data History + CRM)*
Understand Why	The cash story: balance trend, transferable vs trapped, runway — plus whether the company has been at risk before *+ bonus: country & industry sentiment, competitor analysis*	**Signal inputs + labelled history** Now
Prioritise Work order	Order by money at stake × can-they-recover — so the biggest savable money is worked first (the action framework).	**Principal + recoverability read** Now / Medium (Balance Sheet, CFS))
Decide What to do	Loan context — seniority, security, cash vs PIK, covenants — Assets — which sets the action.	**Loan structure, Balance Sheet** Now / Medium
Act & record	Capture the action and the outcome — the labelled history that trains the model.	**Action log** Now (basic)

How the Credit team prioritises action

Hypothesised framework — to validate with the team. Order by money at stake × can-they-recover. Same goal in every box: avoid default. Priority sets the order of work, not whether a loan is worth saving. Most likely actions mentioned, more actions can be ideated with further research and discussion with the team

P3 Help them recover

Small money · can recover

Watch closely, call them, ask owners to add cash
→ back on track, loan intact

Auto (low capacity): auto-schedule closer monitoring

P1 Roll up our sleeves

Large money · can recover

Agree to hold off, rewrite terms, bring in advisors
→ protect a big position, get repaid

Auto (low capacity): none — stays fully human

P4 Handle leanly

Small money · unlikely to recover

Standard playbook, stop lending; still try to save first
→ contain it cheaply

Auto (low capacity): email to CEO, standard notices, freeze advances

P2 Stop the bleeding

Large money · unlikely to recover

Demand repayment, claim collateral, sell and exit
→ recover as much cash as possible

Auto (low capacity): none — stays fully human

Grounded & Cautious automation can be considered where reversible and low-stakes, if the need for it arise. Irreversible calls always stay human.

How we earn adoption

The team already runs at near-zero loss — so we augment what works, we don't disrupt it. — Fast process to keep momentum

1	**Shadow existing workflow, gather pain points**	With at least >= 2 different credit team members. To assess common patterns and user specificities
2	**Augment the current workflow while keeping it very simple**	Get a clear end to end map of their operating ways. And don’t complicate too much
3	**Give them a prototype. Iterate from fast feedback loops**	Daily 30 min sessions; where everybody uses the prototype. Feedback gathering, and iteration. Use AI for multiple versions of the prototype with 1 key element changed if useful
4	**Earn trust with drill-down & grounding**	Every number traces to the raw file — they can verify, not just believe. Data Dictionary
5	**Quantify Success - Before vs After**	AB test, or just get a feel of the uplift of the amount of principal and loans covered per unit of time. NPS

What we're assuming — and how we'd test it fast

We assume…	How we test it
The team prioritises by principal at risk × can-they-recover	Shadow a real triage session; do they work the list in that order?
The following flow mimics how the team operates: See what's at risk → understand why → prioritise → decide what to do → act and record	Shadow sessions in real time. Data on prototype usage
Transferable (not total) cash is the right basis for repayment risk	Confirm convertibility classes with the team; check against real FX friction
Recoverability can be assessed with a combination of cash-flow history, assets and liabilities information, and loan structure information (and later inflow cash-flow forecasts)	Validate. Ask how they currently assess it, if they do. More research on best practices
Right now, a smarter signal matters more than scale or automation	Run a state-of-the-union of the distress-warning operations with the team — where is the real bottleneck today?

How we validate, throughout: Understand current workflow, gather pain points, prototype, fast feedback loops, iterate

PART 3 · THE PROTOTYPE

The analysis & action experience

A working prototype on synthetic data. Four principles — prioritised, understandable, trusted, actionable. — More analytics and actionability capabilities on the actual prototype. Demo in https://alexandrecela10.github.io/liquidity-project-nights-watch/web/agent.html



Prioritised

Ranked worst-first — an Excel-like queue. The biggest savable money rises to the top.

Understandable

Drill into any company: the cash story, why it's flagged, what the loan looks like.

Trusted

Every number traces to its raw source file and a plain-English data dictionary.

Actionable

Log or trigger the action in the same page — end to end in seconds, not days.

For more, see the live prototype → https://alexandrecela10.github.io/liquidity-project-nights-watch/web/product.html

Data acquisition roadmap

What the signal and the workflow each need — acquired in phases.

	NOW	MEDIUM	LATER
Signal data	Bank balances + snapshot history; loan principal, next payment & coupon (from CRM)	Balance sheet & cashflow statements → free cash after commitments	Business-lifecycle data (pipeline, usage); external rate & input-cost forecasts
Workflow data	Ranked list inputs; labelled missed-payment history; basic action log	Full loan context (seniority, security, cash vs PIK, covenants); action-outcome write-back	Automation hooks for low-principal actions; richer outcome → ground-truth loop

Each phase sharpens the same coverage metric and gives the team more to act on. We start with what's available now and design so later sources slot in without rework. See https://alexandrecela10.github.io/liquidity-project-nights-watch/web/roadmap.html for more on the data roadmap

Working with engineering & design

I build a v0 prototype first; then we agree golden success criteria and failure events to avoid; then each partner owns their how of their build and iterates against them. I understand requirements, enable fast feedback loop for improvements and tradeoffs toward the goal. And dive deep if required

Partner	What they own	Agreed success criteria
Data engineering	Bronze → Silver → Gold pipeline: ingestion, FX, entity resolution, freshness	Data true to raw files; low latency; fast querying. Ability to drill down to the raw files from any KPIs
AI engineering	Extract & standardise unstructured raw files (PDF statements, contracts, filings) into the canonical model	Extraction accuracy; grounded to source
ML engineering	The signal: range of expected cash at repayment, once the full feature set exists	Recall-first; calibrated; monitored for drift
Backend + frontend	Bonus if ever we want to build an internal app. Productionise the product and the credit-team workflow, in parallel	Fast workflows. Usable workflows. Reliable refresh; drill-down to raw works end-to-end
Design	Shape the credit-team experience: the ranked list, drill-downs and action flow	Minimise time to execute the JTBD; pleasant to use; complies with company design rules

It's a partnership. Though I have the responsibility to take initiatives in offering user facing solutions ideas and artifact, the best idea wins

What can go wrong — and how we mitigate

While constantly adapting to anything new that can go wrong in the process. Not exhaustive

Risk	Mitigation	Level if unmitigated
Raw file generation & extraction (esp. bank file)	Minimum-latency SLA; escalate if a feed goes stale — staleness is itself a flag	**High**
Data pipeline	Bugs, standardisation errors from new raw inputs; adapters + schema validation, quarantine bad rows	**High**
Data quality	Completeness, freshness & accuracy checks; reconcile to raw; surface confidence	**High**
ML signal — drift	Recall not at 100%. Data & model drift; monitoring, periodic retraining, recall-first calibration	**Critical, High**
ML signal — too many false positives	Prioritise by warning-signal strength; an ML owner dedicated to precision while keeping recall same; cautious automation for low tickets only after careful discussion	**Medium**
Hyper-scale	Prepare the end-to-end pipeline for scale; cautious automation for low tickets, with success tracking on that automation	**Medium**
Product availability. Product Market Fit	Uptime monitoring, graceful degradation, fallbacks for the morning digest. Fast feedback loops for PMF	**High**
End-to-end data governance & compliance	Access levels, confidentiality, storage & transfer — fully compliant with UAE law + gold-standard AI sovereignty	**Critical**

Rolling it out — full suite in 60 days

A tight, staged rollout with the Credit team in the loop throughout. Ambitious roll out plan, to tend towards, or even beat. Tradeoff between short term business needs with working scrappy prototype, and longer term product health with great engineering foundation

Stage	Key milestone at the end	Time	Resources
1. Prototype (v0) - Now version	Working demo on synthetic data; Credit team validates the workflow. POC Adopted	**~ 2 days**	PM + AI eng (me hands-on + advice from design))
2. Now version - MVP	Now-version live for the team. Lower version of signal, only ability to look at cashflow time series + principal and compute ratio + proxy time repayment ratio	**~2 days**	AI PM hands on - only need to aggregate snapshots + CRM + few cleaning
3. 1m Version	More robust signal, with more complete database. More robust actions with loan structure + CFS + balance sheets. *A non 'production grade ready' with already these info can be added in the meantime*	**~21 days**	1 Data Engineer, 1 AI Eng for the pipeline. 1 backend for the product service. Design. FE if capacity
4. 2m Version	Best Actual + ML Signal we can get, full data architecture with company business lifecycle data. The latter should need the most time. Compliance, availability, monitoring in place; full suite shipped	**~30 days**	ML Eng + Data Eng. Data Governance

Those ETA are ambitions. While we will tend towards them, or try to beat them, this should not come at the expense of long term product viability and health

Five layers of success

From the signal landing on time, all the way to the business outcome — each layer builds on the one before - each layer go through objective and tight validation feedback loops

Signal + pipeline	100% of warnings flagged and shared within the agreed SLA — and we keep making it faster.	**Leading**
Signal quality	A good-enough forecast of the lower and upper range of expected cash for repayment at repayment time. Accuracy on train vs test for cash forecast. Recall / F1 Score for at Risk Labelling	**Leading**
Operation	100% of warnings processed with action taken within the agreed SLA (e.g. one day) — and we keep making it faster. Time savings when acting compared to	**Leading**
Decision	An upward trend in the amount of principal rescued — better decisions as more signal arrives. Higher % of most at risk companies covered in the first hour	**Leading / Lagging**
Business (lagging)	The same number of near-losses as before, or fewer.	**Lagging**

LIMITATIONS & OTHER CONSIDERATIONS

What this first draft does — and doesn't — cover

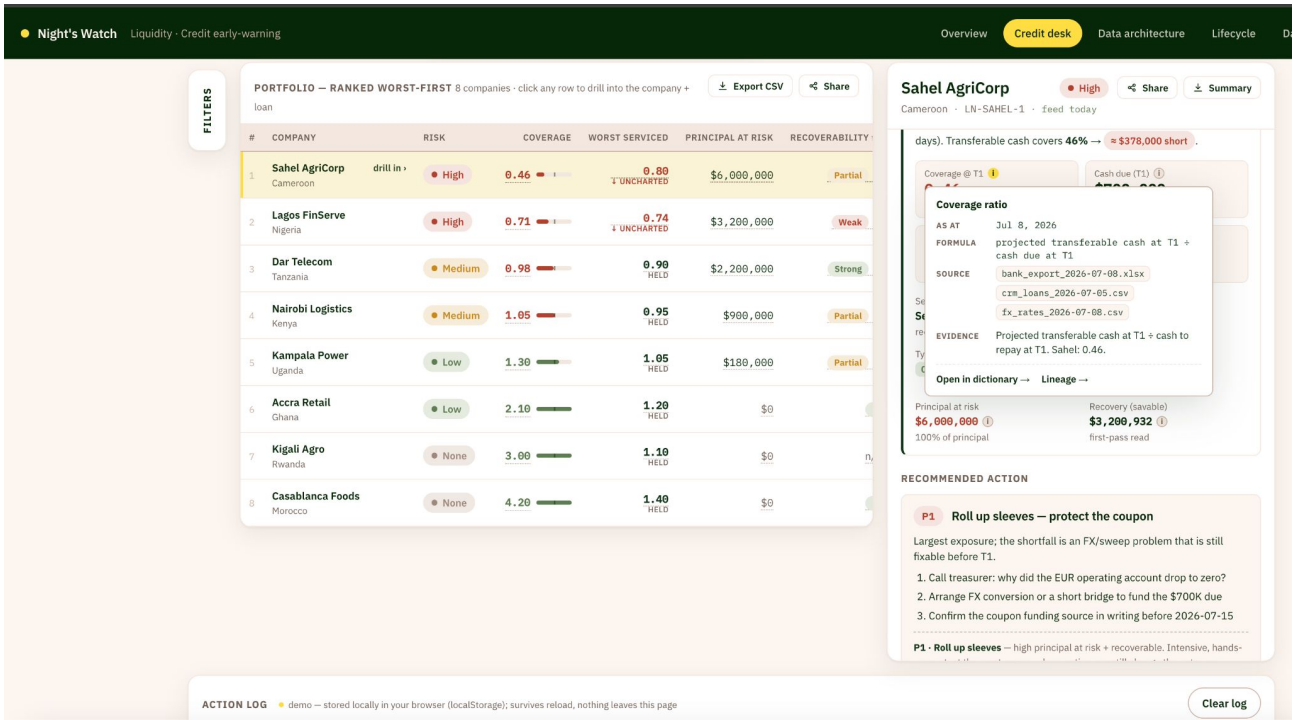
A first draft. More research, including competitor analysis, and iteration will follow to reach a definitive proposal.

Loan-type scope	The workflow fits loans with regular expected cash payments — the bulk of private credit, which is predominantly senior secured direct lending with periodic interest. Other types (e.g. revolvers) come in a later stage, for prioritisation.
Synthetic data	All data shown is currently synthetic; the design is built to take real sources as they arrive.
Recovery signal	The "can we get the money back" read only becomes meaningful once we have loan structure + balance sheet (see appendix). Before that it is a liquidity-trend proxy.
Agentic systems	Cautious, compliant and grounded agentic systems could be considered if a clear use case emerges — especially at very high scale. Not in scope for now. Let’s start simple while gaining strong ground
Other considerations	Potential very low amount of labelled missed cash payment data. Mitigate with other correlated and more frequent sign of distress. Especially with smarter and logged intervention from credit team

APPENDIX

The prototype

Four principles — prioritised, understandable, trusted, actionable.



https://alexandrecela10.github.io/liquidity-project-nights-watch/web/product.html

Data Roadmap, Architecture and lifecycle are especially there for illustrating key elements behind this assignment. They would not necessarily fit within the end tool

APPENDIX · A FUTURE VISION

Dashboard-less, agent-run credit desk

Where this could go once the signal is proven and volume demands it. The analyst reviews and approves; the desk does the finding. TBD after many iterations
https://alexandrecela10.github.io/liquidity-project-nights-watch/web/agent.html

1
Pipeline
Gold: balances, loan terms, coverage 0.46 — already grounded & fresh

▶

2
Credit agent
Reads gold output, picks the skill for the situation → FX-bridge

▶

3
Skill
Loan-savior playbook: steps + guardrails for this case

▶

4
Deterministic automation
Templated action, rule-triggered — Accurately routed by LLM (with evals…)

Gold is already grounded (traces to Bronze) and freshness-checked — the agent inherits that trust for free instead of re-reading source systems live.

step 4 splits by stakes

P1/P2 · irreversible	P3/P4 · reversible, low-stakes
human-initiated — the analyst does it themselves	templated, rule-triggered, with a 15-min hold-and-undo buffer

As much as possible — a deterministic service firing a template

APPENDIX

Golden data model — the entities we relate

Build the bank-balance schema now; let the ontology emerge as related entities arrive. Building it before the data exists would model relationships we can't yet validate. See https://alexandrecela10.github.io/liquidity-project-nights-watch/web/diagram.html for more information

Company

Now

company_id (resolved)
name, jurisdiction
prior-distress history

Account

Now

company_id, account_key (FI + country)
currency, convertibility
balance, snapshot_ts

Loan

Medium

loan_id, company_id, principal, seniority
coupon, cash vs PIK
next repayment date

Payment schedule

Medium

loan_id, company_id, dates, amounts
cash to repay at T1

Financials

After

company_id + time_id
balance sheet
cashflow statements
free cash after commitments

Business lifecycle

Later

company_id + time_id
contracts pipeline
product usage
forecast inflows

Company 1— Account · Company 1—* Loan · Loan 1—1 Payment schedule · Company 1—1 Financials · Company 1—1 Business lifecycle*

APPENDIX

Medallion architecture — with raw-file drill-down

Bronze

Raw, source-faithful

- Bank exports, untouched
- Provenance + timestamp
- Never computed on

Silver

Conformed

- One canonical schema
- Adapters absorb the mess
- Standardisation across each key dimension
- FX re-derived, IDs resolved
- Evidence from raw files for each claim

Gold

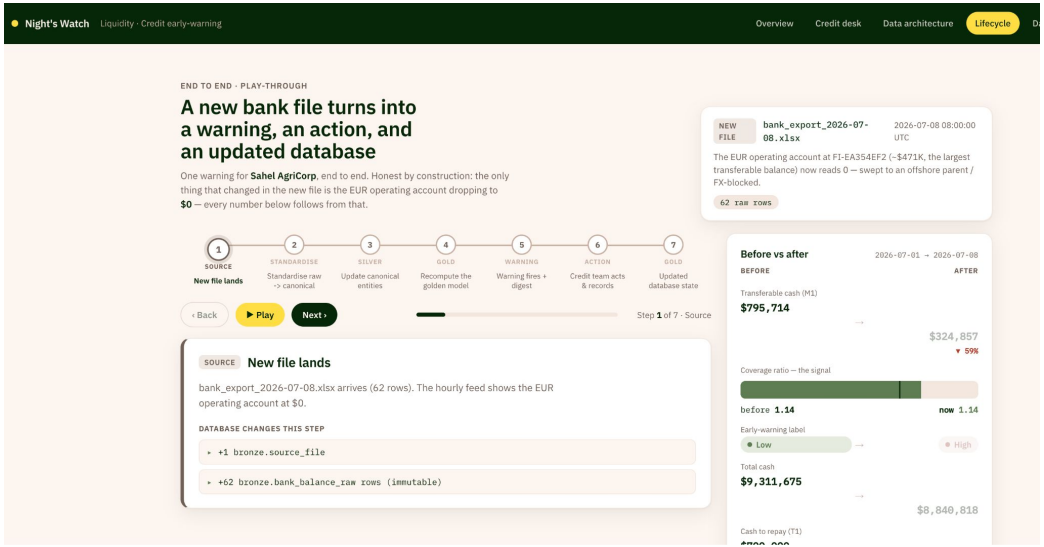
Metrics & features

- Coverage metric per company
- Every key information per company
- Drillable down at loan level
- Model features
- What the team sees

Drill-down: every claim the Credit team sees links back to the exact raw bank file behind it — Gold → Silver → Bronze. Trust by audit, not assertion. See https://alexandrecela10.github.io/liquidity-project-nights-watch/web/diagram.html for more

APPENDIX

A case in point: A new datapoint + signal lifecycle



Drill-down: Winning trust by enable to trace back the journey of any data point, from raw source to what the Credit Team sees

APPENDIX

Three already available metrics, straight from the bank file data

The sample company holds **$9.3M** in cash — but only **$796K** is in freely transferable currency.

82% is trapped in capital-controlled currencies. A loan is serviced in hard currency — so the headline cash number is misleading. This is why we measure transferable cash, not total.

1 Available cash

Transferable cash — what can actually service the loan. The level.

From one snapshot

2 Burn rate

How fast transferable cash is moving, from the snapshot history. The trajectory.

Needs history

3 Runway

Transferable cash ÷ burn — days until it runs out. The consequence.

Needs history

Only the level is computable from the single snapshot provided. Burn and runway come from the history the hourly feed builds up — so the product gets sharper the longer it runs.

APPENDIX

Can we get the money back? — the recovery signal

"Recoverable" is really two questions, answered by different data. The recovery signal only becomes meaningful once loan structure and the balance sheet arrive.

The question	What answers it	Data needed	Real from
Can they cure? (will they make the payment if given time / a restructure?)	Now: only a weak proxy — is transferable cash trending up and runway lengthening? After: the real answer — liquidity problem (fixable) vs solvency problem (impaired)	Cash trend (Now) → balance sheet & cashflow (After)	**After**
What do we recover if they can't? (how much comes back)	Where we sit in the waterfall and what we can claim — seniority, security, covenants. A senior secured position recovers even when the borrower can't cure	Loan structure (seniority, security, covenants)	**Soon**

So at Now the recoverability axis is only a liquidity-trend proxy. It becomes a genuine read as loan structure (Soon) and financials (After) arrive — and Later, business-lifecycle data shows whether they can trade out of trouble.

APPENDIX

UAE data compliance — what binds us

Liquidity operates in Abu Dhabi. If it is an ADGM entity, the ADGM Data Protection Regulations bind it — not the federal PDPL, which excludes financial free zones and credit data. Confirm the entity's regime first. While leading on this front, collaborate with Data Governance and Legal if needed.

Rule / regime	What to make sure of, concretely	Official source
ADGM Data Protection Regulations 2021 (likely the binding regime for an Abu Dhabi financial entity)	Tell the ADGM data office we're processing this data, have a real reason to use it, let people see and fix their own data, report it fast if something leaks, and keep the data properly protected.	*adgm.com — Office of Data Protection*
Federal PDPL — Decree-Law 45 of 2021 (applies onshore; note credit data & free zones are carved out)	Get permission to use personal data (or have a valid reason); be careful moving data across borders; appoint someone responsible for data protection if the law requires it; let people see and fix their own data.	*u.ae — Data protection laws*
Cross-border data transfer	Only move personal data out of the country if it's allowed — either the destination has strong-enough protection, or we've signed the right contract; keep a record of why each transfer is okay.	*ADGM DPR rulebook (Part V) / PDPL (u.ae)*
Data residency & AI sovereignty (gold standard)	Keep data in the UAE where we can; only let people see the companies they're meant to; strip out identifying details before using data to train models; keep a full record of who accessed what.	*No single official source — see UAE AI Strategy 2031*

Not legal advice — to be confirmed with Liquidity's legal/compliance team and the entity's actual jurisdiction. Links are starting points.

APPENDIX

What it takes to build — by phase

Phase	Data engineering	Integration engineering	ML engineering
Now	Bronze + Silver for bank data; FX, entity resolution, freshness checks	CRM loan-structure read; banking-feed ingestion	Rules-based coverage metric; calibrate threshold on labelled history
Medium	Add loan, balance-sheet, cashflow entities; start golden model	Connect financial-statement sources; action-log write-back	Regression: forecast cash at T1; classifier on near-miss + default labels
Later	Business-lifecycle + external rate/cost feeds into golden model	Pipeline + usage data connectors; external forecast APIs	Inflow forecasting; per-segment thresholds; drift monitoring

Two model heads, shared features: a regression that estimates the cash (useful immediately), and a calibrated classifier that decides the risk (as labels accumulate).

APPENDIX

Credit-team user stories — by priority

Grounded examples of what the analyst does at each priority. A starting set — to be expanded by shadowing and research with the team.

Priority	User story	Key actions
P1 · Roll up sleeves (large · recover)	As an analyst, when a large position I can still save is flagged short, I want the cash story and the loan terms in one view so I can act fast to protect the most money.	Call treasurer; arrange FX bridge; rewrite terms; bring in advisors
P2 · Stop the bleeding (large · impaired)	As an analyst, when a large position looks structurally impaired, I want our seniority, security and recovery estimate up front so I can protect capital quickly.	Demand repayment; claim collateral; sell/exit; assemble enforcement file
P3 · Help them recover (small · recover)	As an analyst, when a small position is curable, I want a light-touch path so I can keep it on track without burning hours.	Watch closely; call them; request equity cure; closer monitoring
P4 · Handle leanly (small · impaired)	As an analyst, when a small position is unlikely to recover and I'm at capacity, I want the standard playbook to run with my sign-off so nothing is left unattended.	Standard notices; stop lending; auto email to CEO (low-stakes); freeze advances

Every action that's irreversible or needs negotiation stays human; only reversible, low-stakes, low-principal steps may be automated under capacity pressure.

APPENDIX

What a training dataset could look like

One row per **company × snapshot**. Features are what we knew at the snapshot; labels are what actually happened by the next repayment (T1) — so the data **labels itself** as time passes.

company_id	snapshot_ts	transfer_cash	burn_rate	runway_d	prior_distress	…	cash_at_T1 (y_1)	missed_pay (y_2)
CO-014	07-06 09:00	$0.81M	$42k/d	19	no	…	**$0.74M**	**0**
CO-014	07-07 09:00	$0.79M	$45k/d	17	no	…	**$0.74M**	**0**
CO-022	07-07 09:00	$0.12M	$60k/d	2	yes	…	**$0.05M**	**1**
CO-031	07-08 09:00	$3.40M	$70k/d	48	no	…	*(pending)*	*(pending)*

Features (X) — everything known at the snapshot — cash level, burn, runway, prior-distress flag, and later loan structure & financials.

Label y_1 — regression — realised transferable cash at T1. Self-supervising: every snapshot becomes labelled once its T1 passes — abundant from day one, no defaults needed.

Label y_2 — classifier — did the loan miss its cash payment? The scarce, lagging label — use proxies (covenant breach, late payment, watch-list) until real events accrue.

Pending rows (snapshot too recent for T1 to have arrived) carry features but no label yet — they mature into training data automatically.

APPENDIX · KEY PRINCIPLE

A missed warning costs far more than a false alarm

The cost function must **penalise a false negative far more heavily than a false positive** — never treat them alike. That asymmetry *is* what "recall-first" means, expressed as a cost.

False negative

"Fine" → then it misses. A surprise default.

Cost: **up to the whole principal at risk**, plus the hit to a near-zero loss record.

This is **exactly the failure** the product exists to prevent.

False positive

Flagged → turns out fine. A false alarm.

Cost: **an analyst hour** to drill in and confirm it's fine.

Bounded and small — though **too many erode trust**, so we still cap it.

Minimise expected cost = Σ [P(miss)·C□□·(missed) + P(no miss)·C_{oo}·(false alarm)]

with **C□□ ≫ C_{oo}** (scale C□□ by principal at risk), or equivalently a **recall floor** the Credit team sets.