

PREDICTIVE LIQUIDITY

Using AI and Machine Learning to Improve Cash Flow Forecasting

From Uncertainty to Control

Finance Professional. Practitioner.

Michael Krahn CMA, CSCA, CPA, PMP, PMI-ACP, PMI-CPMAI · Michael.Krahn@remarcdata365.com

Agenda

01

Why Cash Forecasts Miss

How behavioral timing—not balances or averages—creates liquidity risk

02

From Point Forecasts to Predictive Liquidity

How AI and machine learning produce more realistic, decision-useful forecasts

03

Making the Forecast Trustworthy

Explainability, human judgment, governance, and organizational adoption

04

What Predictive Liquidity Teaches Us

The changing role and capabilities of finance professionals

*The number was defensible. The cash
wasn't there when you needed it.*

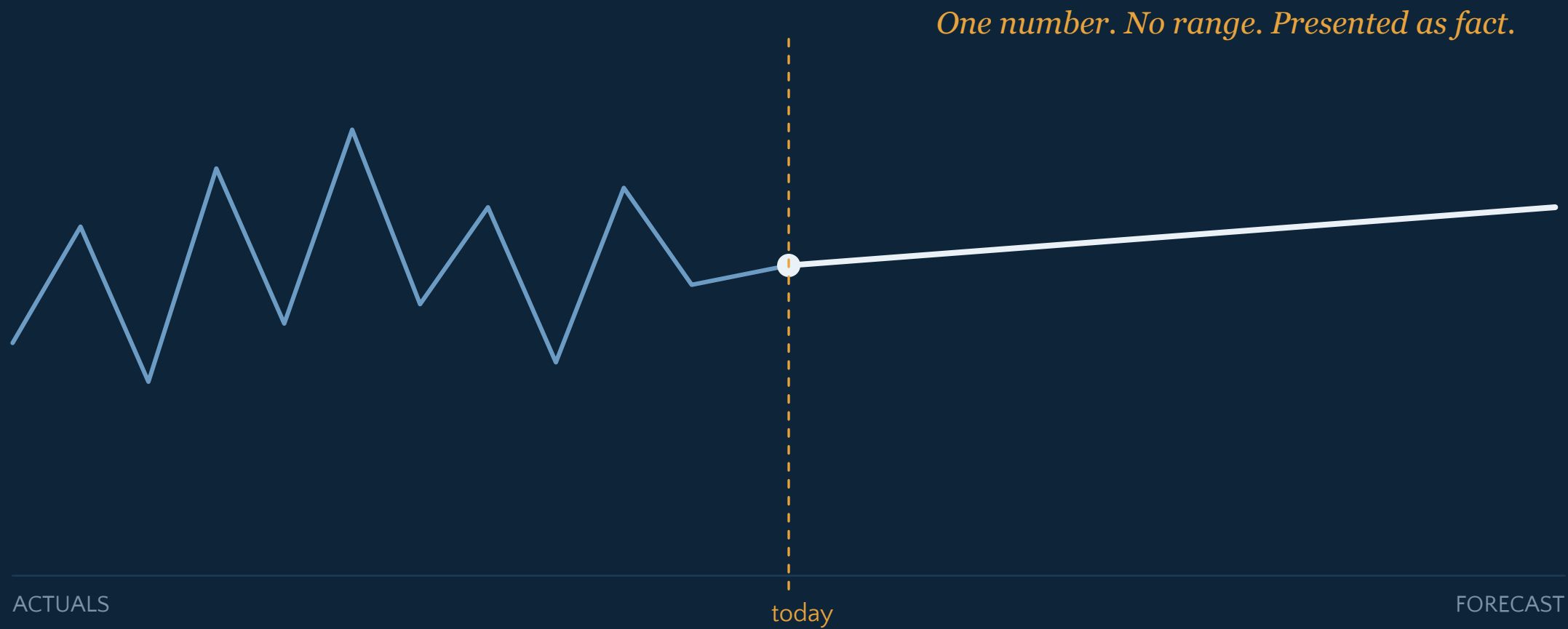
What This Costs

The decision	What late or incomplete timing information costs
Hold cash, or draw on the revolver?	Idle cash earning nothing — or an unnecessary draw and the interest on it
Pay the vendor now, or hold?	A strained supplier relationship, or a missed discount
Greenlight the investment, or wait?	Opportunity deferred on cash you actually had — or a commitment on cash you didn't
Reassure the board, or flag risk?	Credibility spent on a number that moved by week three

The Problem

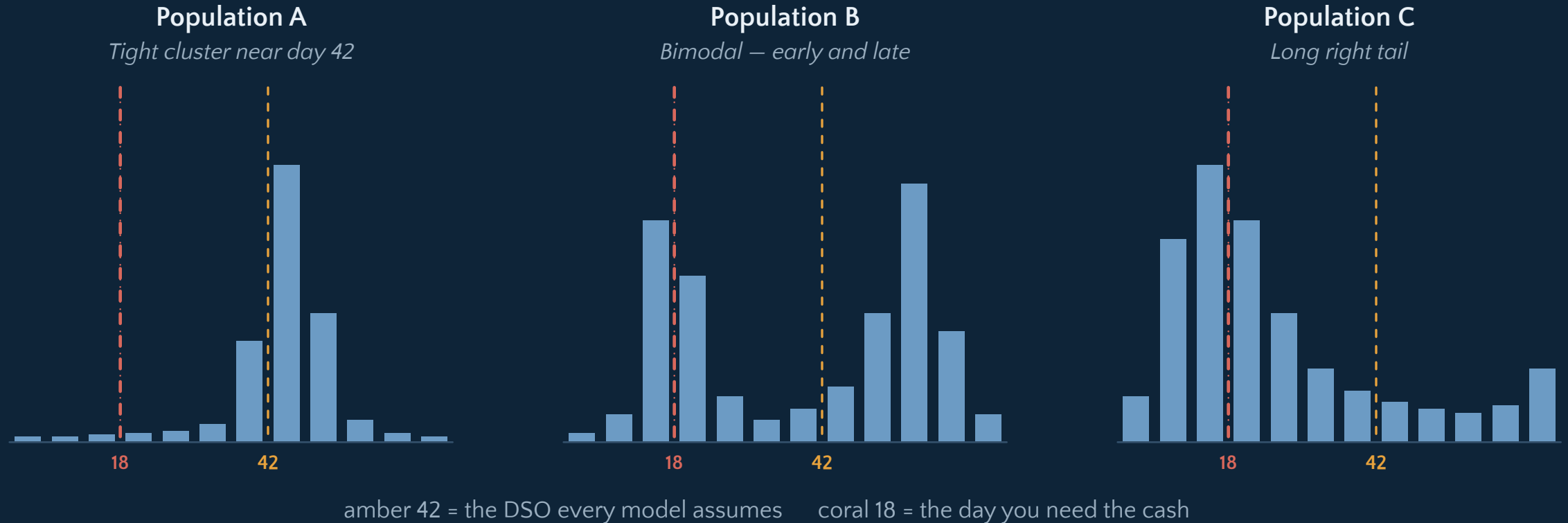


The Honest Diagnosis



Where Forecasts Actually Break Down

DSO = 42 days — identical for all three populations



Same average. Three different timing realities. Three different cash positions on day 18.

Why AR and AP Timing Matter Most



ALREADY KNOWN — APPROXIMATELY

- ✓ How much they'll collect
- ✓ How much they'll spend

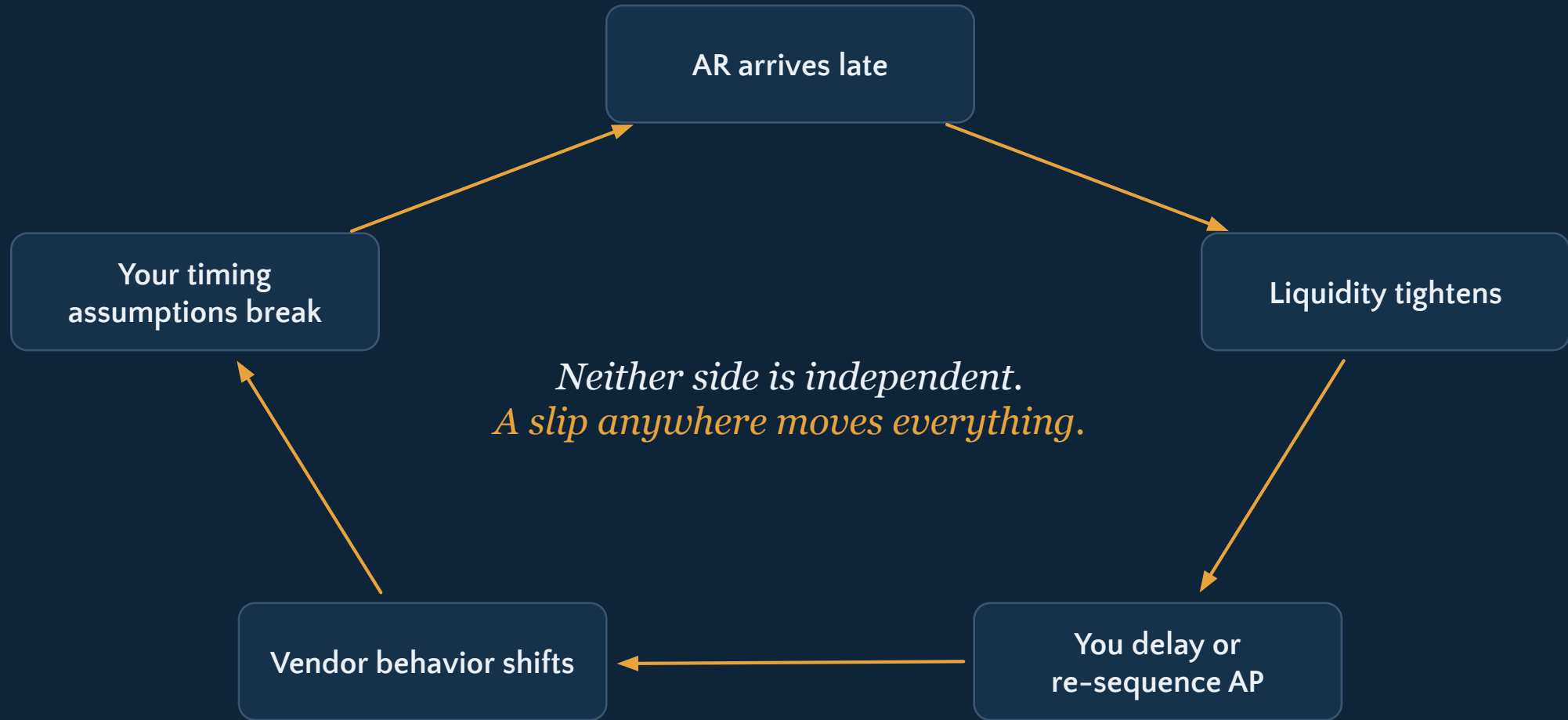
THE HARD PART

WHEN

the timing of every dollar in and out

*Both amounts are roughly known. **The timing isn't** — and that's what decides liquidity.*

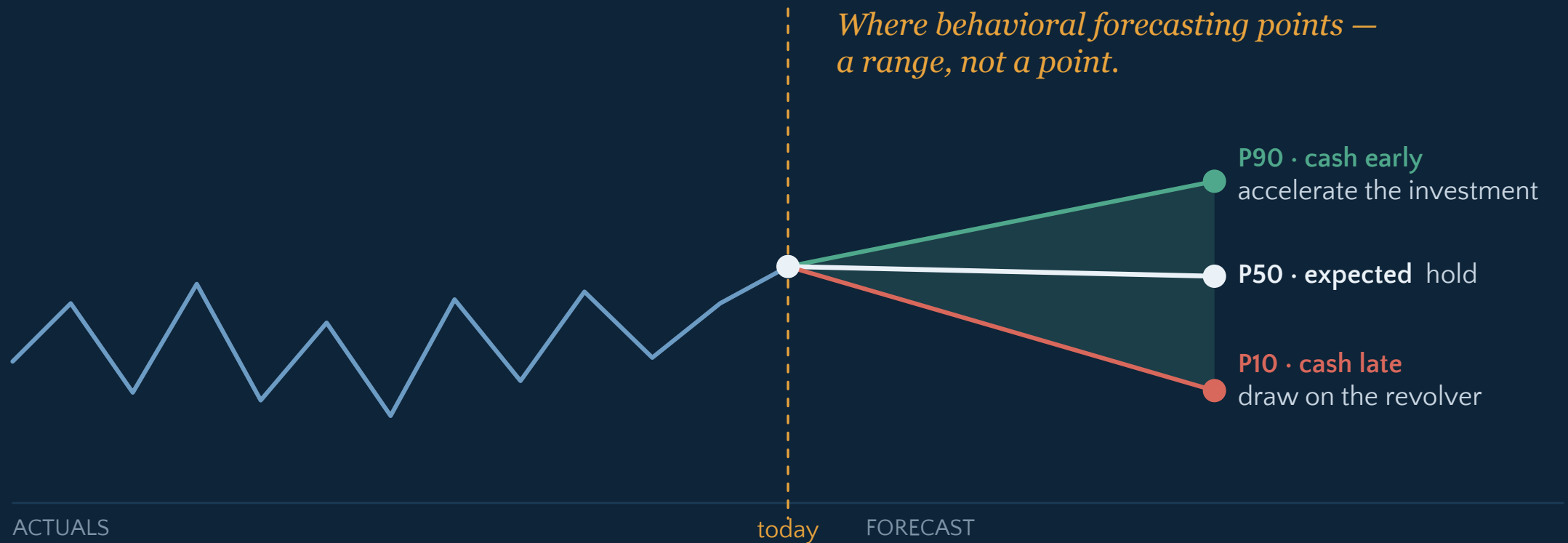
The Compounding Problem



Now run this loop across hundreds of customers and vendors, every week, simultaneously.

From Single Number to Confidence Range

Three plausible outcomes, not one prediction.



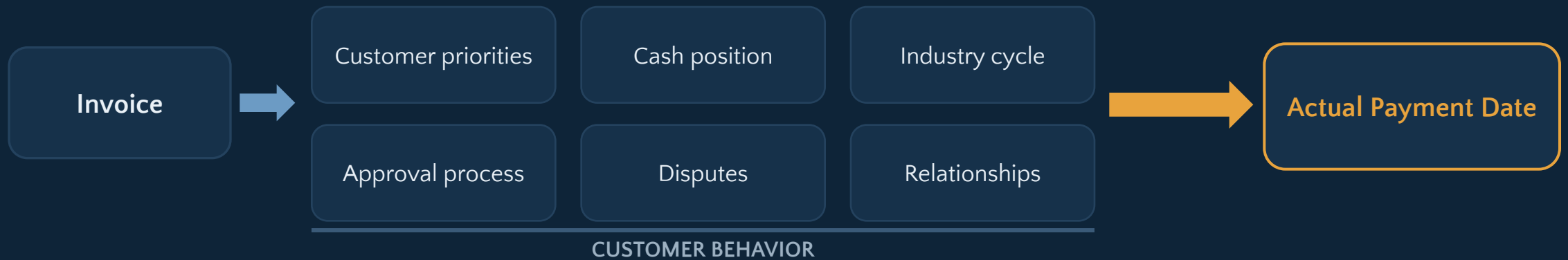
When using AI/ML, this isn't a generic band bolted onto a spreadsheet number. **It's built from how this specific account actually behaves.**

Timing is Behavior

THE TEXTBOOK ASSUMPTION

Invoice + Payment Terms → a fixed Payment Date

WHAT ACTUALLY DECIDES THE DATE



*Behavior creates **timing**. Timing creates **liquidity**. Liquidity drives **decisions**.*

Why Traditional Models Reach Their Limit

Spreadsheet Forecasting

Deterministic assumptions
Averages
Static relationships
Limited segmentation

Human Analysis

Recognizes exceptions
Understands context
Limited working memory
Difficult to scale



Machine Learning

Detects thousands of behavioral patterns
Updates as new behavior arrives
Scales to every account
Better timing assumptions

TRADITIONAL MODELS

The limit was never the spreadsheet — it's maintaining thousands of shifting timing models at scale. Machine learning doesn't replace the spreadsheet; it improves the assumptions feeding it.

What Skeptics Are Right About

“I can’t defend an output I can’t explain.”

“AI doesn’t clean my data — and my data is messier than any demo admits.”

“I’ve sat through the failed technology project before. Twice.”

“A more sophisticated forecast is still a forecast. It can be confidently wrong.”

MEANWHILE, THE MARKET

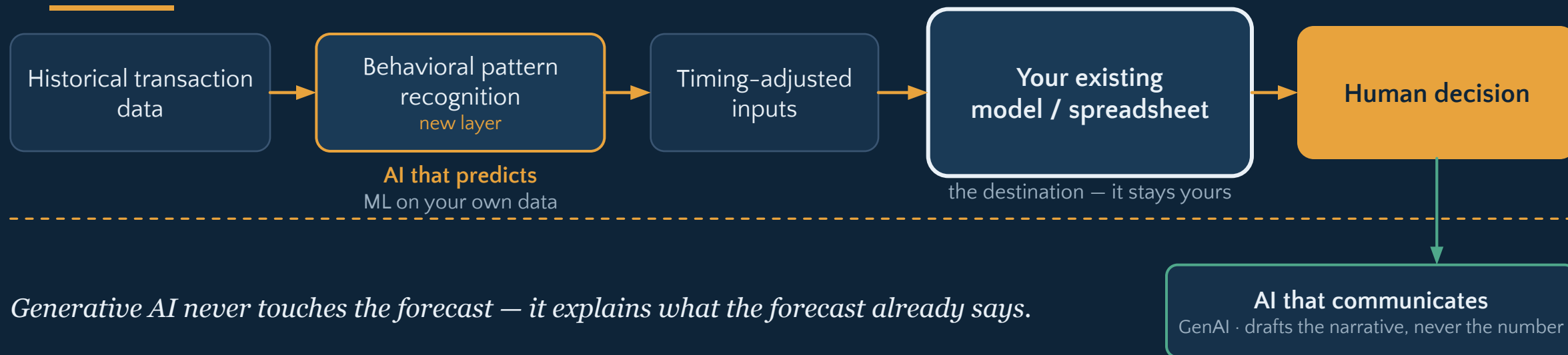
The first three are fair — and vendors are already answering them: the major ERPs and treasury platforms now ship payment-timing prediction (*see appendix*).

The fourth stays open: **none yet tell you how sure they are.**

That gap is the rest of this talk.

*A spreadsheet can compute an average. It can't hold — and keep updating — **a distinct, evolving pattern for every account.***

What AI Actually Does in a Forecasting Workflow



What this AI layer does NOT do

- Replace accountant judgment
- Override governance
- Guarantee accuracy
- Make organizational politics any easier

Existing method

13-week cash flow

DSO / DPO modeling

Scenario modeling

What the AI layer changes

AR assumption → a behavioral timing pattern

Population average → account-level profiles

Scenarios informed by data-derived patterns

Predict upstream. Communicate downstream. Human judgment in between. **That line is not optional.**

The Build



Dynamic Liquidity Intelligence (DLI)

A practitioner laboratory — not a product.

WHAT IT IS

A working system I built to find out what actually helps a cash forecast — and what only looks like it helps. ***Built so you can manage AI/ML risk in vendor solutions.***

WHAT IT IS NOT

Something I'm here to sell you. ***Nobody in this room is a prospect.***

What I Got Wrong First

WHAT I FIRST THOUGHT “DONE” LOOKED LIKE

“The forecast looks correct.”



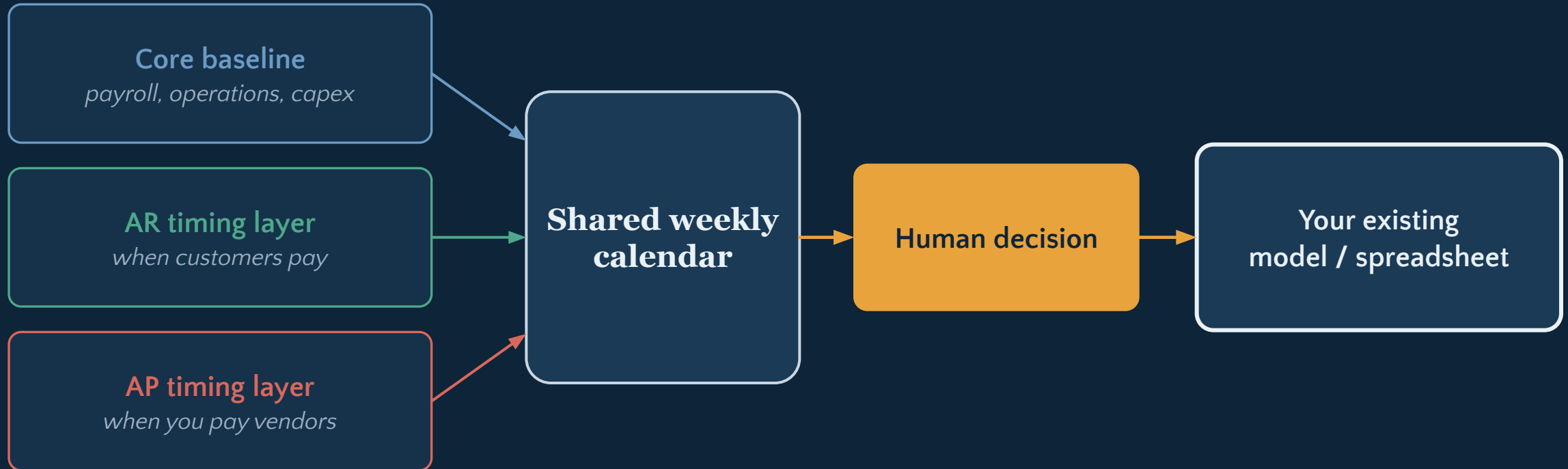
WHAT “DONE” ACTUALLY MEANS

**Reproducible. Traceable.
Explainable.
Defensible.**

— when someone asks how you got it.

*I treated all of that as “glue.” It turned out to be the **load-bearing wall.***

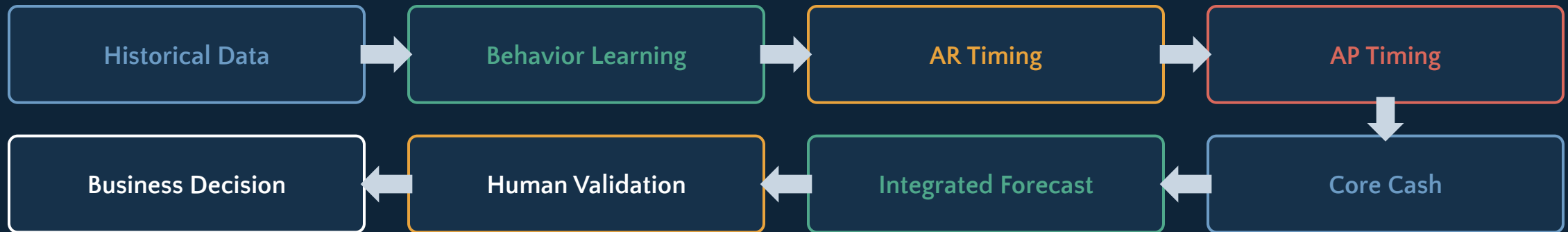
The Architecture, Briefly



Each stream forecasts only as far as its data supports — Core further out, AR and AP shorter. We show those boundaries rather than pretending to one clean number.

DLI Conceptual Architecture

The same Core, AR and AP streams from the last slide — now shown end to end.



WHAT DLI IS NOT

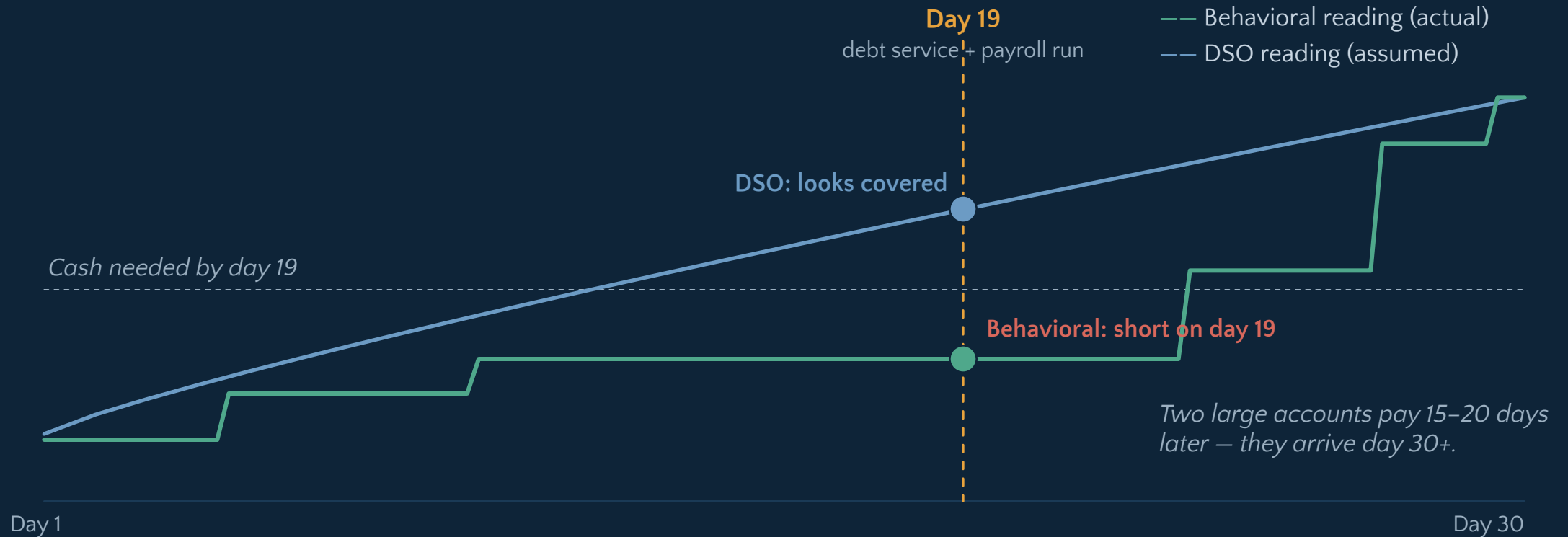
- A real-time cash ticker
- A replacement for judgment
- A single guaranteed number
- A black box you can't audit

WHAT DLI IS

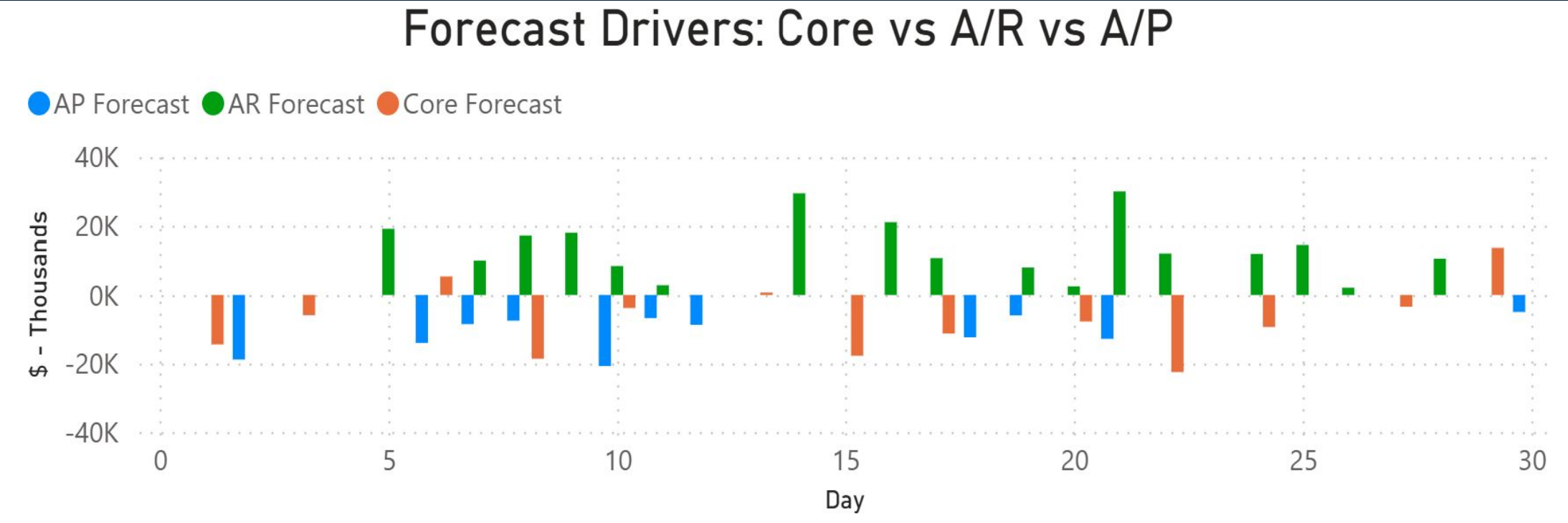
- A behavioral timing model — gradient-boosted trees (XGBoost) on your data
- Built to range, not fake precision — the calibrated band is the next layer
- Updated as new payment behavior arrives
- Auditable, human-validated, governed

DLI predicts timing, refuses false precision, and stops at human judgment — the model informs the decision; it never makes it.

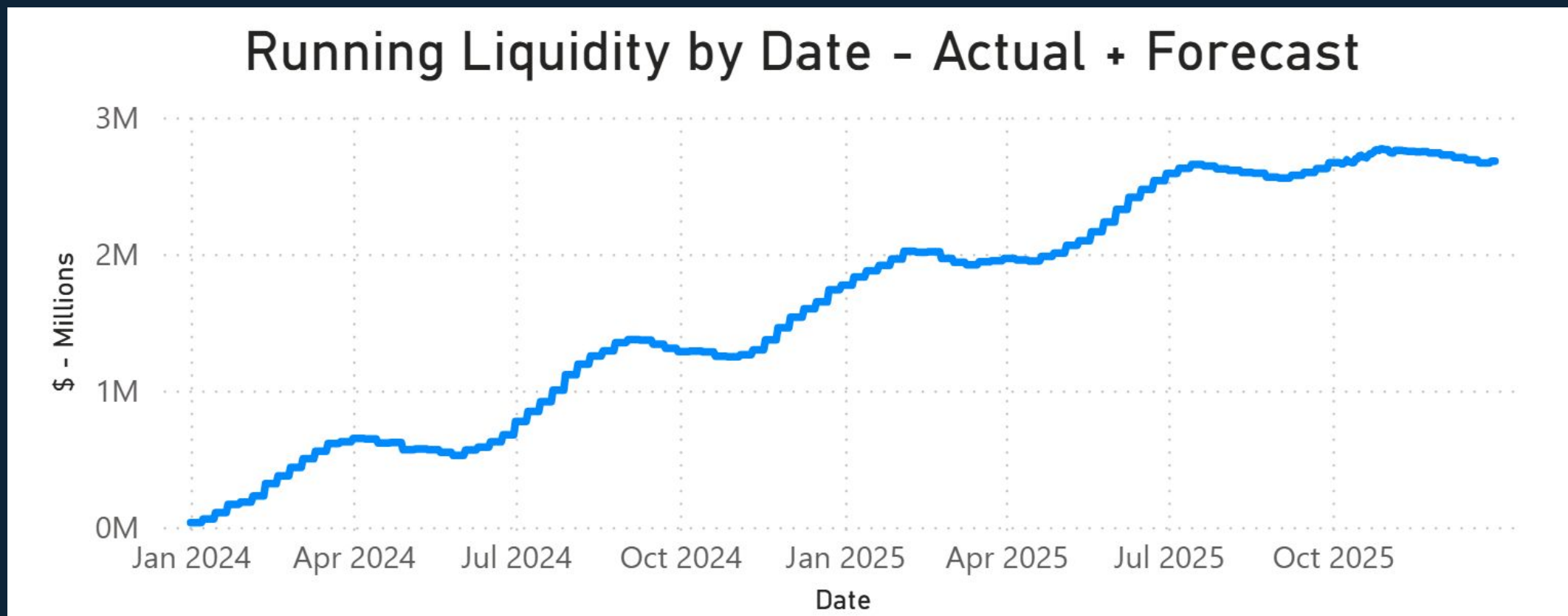
A Decision That Went Differently



Same total by month-end. **Opposite reality on the day it's due.**

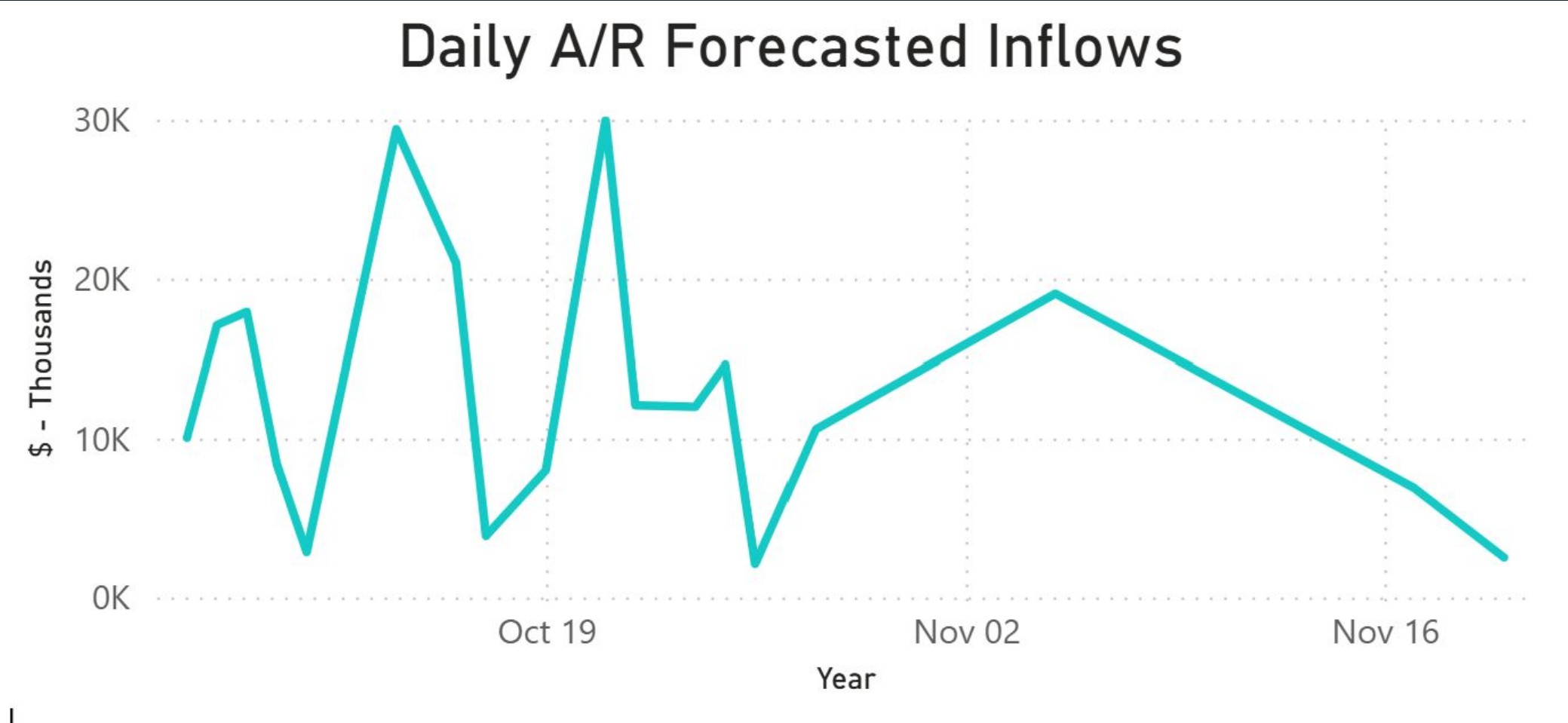


Three independent behavioral prediction problems — Core, AR, AP — measured separately, then combined.



Your actual liquidity position, continuing into a behavior-based forward view.

A single expected-case path — the full P10–P90 band is the next layer, not a claim for today.



learned from each customer's actual payment history, not an aggregate DSO.

Trust & Adoption



Who Owns the Model When It's Wrong?

A vendor can own the code. It cannot own these risks.

ROLE	WHAT THEY OWN	THE QUESTION THEY ANSWER
Model owner	The behavioral logic and its assumptions	<i>"Why does it predict what it predicts?"</i>
Data owner	The transaction history feeding it	<i>"Is the input trustworthy and current?"</i>
Output reviewer	The forecast before it's acted on	<i>"Does it pass the sniff test before it leaves the room?"</i>
Audit trail	The record of every run	<i>"Can we reconstruct exactly how we got this number?"</i>

None of these roles is "the AI" — or the vendor that sold it. **The AI produces. People own.**

The Control Environment for AI-Augmented Forecasting

- | | | |
|---|--|--|
| 1 | Data lineage documentation | Every number traces to its source and vintage — no orphan figures |
| 2 | Model assumption disclosure | Behavioral assumptions written down and visible — not buried in code |
| 3 | Override protocols | When a human overrides the model, it's recorded — who, why, and when |
| 4 | Variance tracking with root cause | Misses logged and explained — not just noted and forgotten |
| 5 | Periodic model validation | A scheduled cadence to re-check the model against reality — not set-and-forget |

*Not bureaucracy — what makes AI output **trustworthy enough to act on**, and **defensible when you're asked to explain it**.*

What the Board Needs to See

The evidence a CFO or Treasurer tables to explain the model — and prove it is validated.

DOES IT WORK?

Performance, honestly measured

- Backtest vs. actuals, out-of-sample, by horizon
- Band calibration — the validation ranges will need next
- Champion/challenger vs. the 13-week method
- Sensible drivers (SHAP): payment history, DSO

IS IT UNDER CONTROL?

Ownership, records, and review

- Named owner and sign-off — accountable when wrong
- Override log: who, why, and when
- Version control and scheduled re-validation
- Data lineage and independent review

WHAT IF IT'S WRONG?

Known limits and tripwires

- Documented failure modes and horizon limits
- A defined fallback when it misses
- Drift tripwires that force re-validation
- Named residual-risk acceptance

Governance isn't a page of principles — it's this: better than before, sensible drivers, a named owner, and a tripwire for drift.

Industry Contexts

Manufacturing

Big customers pay around their own production and shipment cycles — not your terms.

→ Learns each major customer's true payment rhythm instead of assuming one uniform DSO.

Healthcare

Payer reimbursements arrive in fragments over an unpredictable tail, by payer and claim type.

→ Predicts the realistic collection curve by payer — not a blended average that hides the tail.

Insurance

Premium inflows and claim outflows are both lumpy and event-driven — timing, not totals, drives strain.

→ Separates the predictable premium rhythm from the volatile claim timing.

Professional Services

Revenue rides on project milestones and client behavior that varies by client and engagement.

→ Learns client-by-client payment patterns — how each actually behaves, not the master terms.

*The AI application differs by industry. The underlying problem — **behavioral timing uncertainty at scale** — is universal.*

What Good Enough to Start Looks Like

1

Disaggregate your DSO

Rank your top 20 customers by how far each one's real payment timing strays from your blended DSO. The outliers are where your forecast is already lying to you.

2

Add one column to your 13-week model

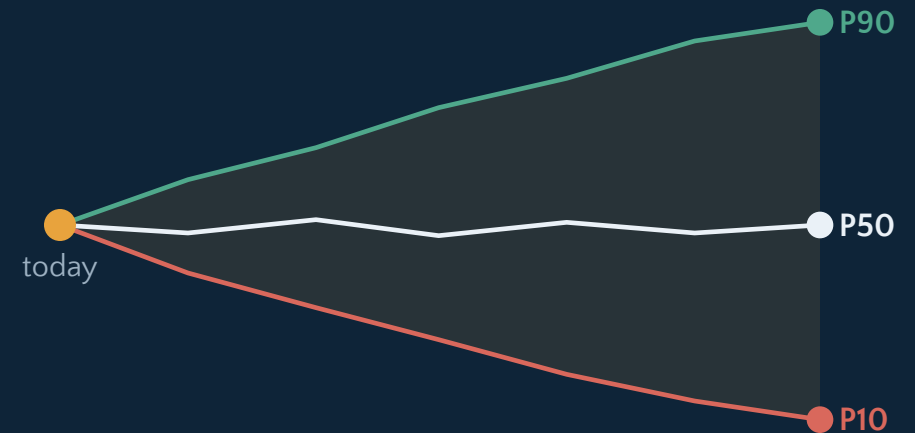
For your top 10 AR accounts, add a timing-variance column — best case, expected, late case. Ten point estimates become ten honest ranges.

3

Build one honest range

Put a manual P10 / P50 / P90 around your single biggest AR assumption. Not software — a spreadsheet and an honest conversation about how wrong it could be.

An honest range — built by hand, in a spreadsheet



*You don't need a seven-layer system. You need **one honest question — when, not just how much — applied to data you already have.***

The 90-Day Adoption Roadmap

DAYS 1–30 DIAGNOSTIC

Find where your timing actually breaks.

- Disaggregate DSO on your real book.
- Find the few accounts driving most of the uncertainty.

OUTPUT

A ranked, evidence-based picture of where the problem lives.



DAYS 31–60 PILOT

Test behavioral timing on one narrow slice.

- One stream (AR), one segment, one forecast cycle.
- Run it beside your existing model — not instead of it.

OUTPUT

A measured comparison, on your own data.



DAYS 61–90 GOVERNANCE

Build the controls around what worked.

- Now — and only now — apply the ownership matrix and controls.
- You govern something you have watched run.

OUTPUT

A defensible, documented, repeatable process.

*Build the governance structure **around something that is working** — not before you know what you're governing.*

The Bigger Question



What Building DLI Taught Me

WHAT BUILDING IT ACTUALLY TAUGHT ME

- ▲ **Timing is behavior** — not a due date; the model that respects that beats the one that assumes it.
- ▲ **Judgment got more valuable, not less** — a range is only as good as the person reading it out loud to a board.
- ▲ **Governance is the product** — defensibility is what makes a forecast usable in a decision that matters.
- ▲ **Data quality is the ceiling** — the model is only as honest as the history you feed it.
- ▲ **Start narrow, learn fast** — one stream, one segment, one cycle beats a grand rollout every time.

If you are early in your career, you are not behind. *You are early.*

Fluency in both the accounting fundamentals and these analytical capabilities will be unusually valuable within five years — because that combination is rare today.

The Question Changed

WHERE I STARTED

“Can AI forecast cash better than I can?”



WHERE I LANDED

“How do I keep becoming more capable as AI does more of the work?”

From Predictive Liquidity to Predictive Capability.

Read this talk as an early signal, not a sales pitch — the capability is still forming, which is why now is the time to build it.

If DLI was the alert, AIALF is what you do about it.

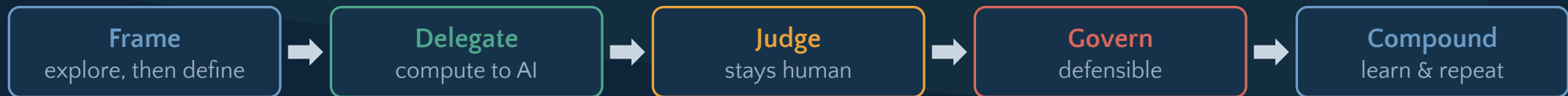
AIALF

AI-Augmented Learning Framework

Not a theory of learning — what building DLI showed me about staying capable as AI does more of the work. A practitioner's working model, still taking shape.

What I keep seeing: AI isn't mainly changing what we do — it's changing how we become capable.

A LOOP I KEEP RUNNING · AI THROUGHOUT, THE JUDGMENT STAYS MINE



*AI will keep changing what you do. **The real question is who you're becoming while it does.***

Appendix

Where the Market Is Headed

Appendix — a sample of shipping solutions, for context. Illustrative, not exhaustive; not an endorsement.

SOLUTION

WHAT'S SHIPPING TODAY (PUBLICLY DOCUMENTED)

Oracle NetSuite

Payment Date Prediction · ERP

ML forecasts each invoice's likely payment date from that customer's history — shown on the invoice record.

SAP S/4HANA

Behavioral Insights · ERP

Learns historical customer behavior to predict — and explain — late-payment risk.

HighRadius

Autonomous Finance · O2C

AI predicts invoice payment dates inside collections and cash forecasting.

Tesorio

Connected Finance Ops · ML

Predicts when each individual invoice will be paid; offered via the Workday Marketplace.

Kyriba

Treasury (TMS) · Cash AI

AI-driven cash forecasting to sharpen liquidity accuracy across bank and ERP data.

Validation, not competition — the market predicts a payment date; the calibrated P10–P90 range is where this goes next.

Q&A • Confidence Ranges

Appendix — prepared responses for the Q&A session.

“Won’t Excel — or Copilot in Excel — just build these ranges anyway?”

It can draw one, but a native Excel interval extrapolates a single aggregate series and is behavior-blind. The skill isn’t producing a range — it’s judging whether one is behavior-driven, calibrated, and governed enough to act on.

“How would you actually build the range?”

Quantile regression for the P10–P50–P90 band, wrapped in conformal prediction for a coverage guarantee, with a time-to-event framing so unpaid invoices are handled honestly — then validated on held-out periods.

“Isn’t a confidence range just false precision?”

Only if it’s uncalibrated. A band you’ve validated for coverage — does P10–P90 hold about 80% of the time? — earns trust. An uncalibrated interval you can’t audit is worse than an honest single number.

Q&A • Model & Method

Appendix — prepared responses for the Q&A session.

“Why XGBoost rather than a neural net or an LLM?”

Payment history is tabular, and gradient-boosted trees are the proven, interpretable, well-governed choice for tabular prediction. The LLM drafts the narrative — it never touches the number. Match the model to the data.

“How much data do you need, and what about data quality?”

Enough account-level history to see behavior repeat. Data quality is the ceiling — the model is only as honest as what you feed it, which is why lineage and ownership come first.

“How do you handle new customers with no history?”

Fall back to segment-level priors until the account builds its own record — and be explicit that early forecasts for them carry wider uncertainty.

“How do you know it isn’t overfitting the past?”

Test it out-of-time — train on the past, score a later period it never saw — and make it beat the existing method there. XGBoost’s own regularization and early stopping curb memorizing, and live accuracy monitoring catches decay, since anything that overfit fades fast in production.

Q&A · Model Selection

Appendix — what we tested before choosing XGBoost.

MODEL TESTED

OUTCOME

Linear Regression

Baseline benchmark

Too simplistic for the nonlinear behavior of cash timing.

Random Forest Regressor

Nonlinear ensemble baseline

Reasonable forecasts, but less consistent than boosting and less control over tuning.

XGBoost Regressor

Primary candidate · SELECTED

Best balance of accuracy, stability, speed, and explainability — the Core Cash model in v.3.

The model wasn't a default — it was a bake-off: baseline, then ensemble, then boosting.

Q&A · Governance & Audit

Appendix — prepared responses for the Q&A session.

“Who owns the model when it’s wrong?”

A named person — never ‘the AI’ or the vendor. The model produces; people own. That accountability is the whole point of the control environment.

“How is this auditable or SOX-compliant?”

Data lineage and vintage, written-down assumptions, a logged override trail (who, why, when), version control, and independent review. A SOX flag applies the moment output touches reported numbers.

“What happens when it drifts or breaks?”

Monitored drift tripwires trigger re-validation, and a defined fallback takes over. Known failure modes and horizon limits are documented up front, not discovered live.

“How do you stop people blindly trusting the output?”

The workflow forces a human validation step before any number becomes a decision — the model informs, it never decides.

Q&A • Adoption & Careers

Appendix — prepared responses for the Q&A session.

“Where do I start — and isn’t this only for big companies?”

No. Start by hand: one stream, one segment, one forecast cycle, built in a spreadsheet beside your existing model. The thinking matters more than the infrastructure.

“If AI does the modeling, what’s left for the accountant?”

The work moves up, it doesn’t disappear. Judgment on assumptions, governance design, and turning a range into a decision become more valuable, not less.

“I’m early in my career — am I behind?”

You’re not behind, you’re early. Fundamentals plus fluency in these tools is the combination that compounds fastest.

“Isn’t ‘learn to work with AI’ just hype?”

The claim is concrete: a repeatable loop — frame, delegate, judge, govern, compound — with the judgment always human. That’s a practice, not a slogan.
