SAMPLE

Building the AI Infrastructure for Finance of Tomorrow

A Practical Guide

Aiman JoeHanz

Free sample: Introduction & Chapter 1

putrabooks.github.io

Buy the full book:

Carousell carousell.sg/p/building-the-ai-infrastructure-for-finance-of-tomorrow-ebook-pdf-1467066546

Gumroad

Ko-fi

ElevenReader

Payhip

Contents

Introduction

Chapter 1: The State of Play and Its Limits

Chapter 2: Why the Familiar Approaches Fall Short

Chapter 3: The Cost of Inaction

Chapter 4: Foundational Principles

Chapter 5: Core Architecture and Design

Chapter 6: Implementation Strategies

Chapter 7: Technology and Tools

Chapter 8: Operations and Management

Chapter 9: Scaling and Future-Proofing

Chapter 10: Case Studies and Best Practices

Frequently Asked Questions

Glossary

About this Series

Connect

This free sample contains the Introduction and Chapter 1. The complete book contains the full contents above.

Introduction

Finance has always been an information business dressed up in the clothing of a money business. Every dollar that moves through a bank, an exchange, an insurer, or a payments network is really a data point moving through a decision. Should we lend? At what price? Is this transaction fraudulent? What is the fair value of this instrument right now, given a market that will look different a millisecond from now? For most of the industry's history, those decisions were made by people using paper, then by people using spreadsheets, then by people using deterministic rules encoded into monolithic core systems. The infrastructure that supported those decisions — mainframes, relational databases, batch schedulers, message queues — was built for a world in which the volume of decisions was bounded by human attention, the complexity of models was bounded by what a quantitative team could code by hand, and the speed of decisions was bounded by end-of-day cycles.

That world is over. It has been over for at least a decade, and the pace at which its remnants are being dismantled is accelerating. Modern finance runs on models that no human can hand-write, on data volumes that no analyst can eyeball, and on latency budgets measured in microseconds rather than minutes. Fraud rings adapt in hours; customers expect credit decisions in seconds; regulators expect explainability across millions of daily determinations; markets clear across dozens of venues in parallel. The only way to meet all of these expectations simultaneously is to build financial services on top of an infrastructure explicitly designed for machine learning, for real-time decisioning, for high-throughput data processing, and for the peculiar operational and regulatory constraints of the financial industry.

This book is about that infrastructure. It is not a book about how to train a specific model, and it is not a book about a specific vendor's platform. It is a book about the durable design choices — data, compute, storage, networking, governance, security, and operating model — that make an AI-native financial institution possible. The word "AI" here is used deliberately in its broadest and most pragmatic sense: statistical learning, deep learning, reinforcement learning, and the emerging class of large foundation models, all applied to problems where correctness, latency, cost, and auditability all matter simultaneously.

Finance is a hard place to build AI infrastructure. The workloads are enormously diverse: batch risk aggregations that chew through billions of positions overnight, streaming fraud scoring that must return a decision in under one hundred milliseconds, low-latency trading systems in which every

microsecond of tail latency has a measurable P&L impact, generative assistants that summarize research or draft compliance memos, and reinforcement learning agents that adapt hedging strategies to changing market conditions. Each of those workloads has different compute, memory, and latency characteristics, and yet they all sit inside the same institution, share the same data, are subject to the same regulators, and must be operated by the same people. Building the AI infrastructure for finance means building an environment in which all of these can coexist without one starving another and without any of them producing an outcome that a regulator, a customer, or an auditor cannot accept.

This book is structured to walk you through that problem in the order you have to face it. Part I lays out the problem: the state of financial infrastructure today, where it is running out of runway, and why the cost of standing still is now higher than the cost of change. Part II then walks through solutions: the foundational principles, the reference architecture, the implementation strategies, the underlying technologies, the operating model, how to scale, and finally what the best-in-class institutions are actually doing. The FAQs address the questions that recur in every executive briefing and every architecture review. The Action Plan gives you a phased roadmap you can adapt to your own institution's starting point. The Summary distills the key ideas into a short document you can hand to a sponsor.

The reader I have in mind is broad on purpose. If you are a chief technology officer, a head of data, a head of risk, or a head of quantitative research, this book gives you a shared vocabulary and a shared blueprint that your teams can execute against. If you are an infrastructure engineer, a platform lead, an MLOps engineer, a data engineer, a security architect, or a model risk manager, this book gives you the context in which your technical choices land and the connective tissue that ties your work to the work of your peers. If you are a consultant, a vendor, or a regulator trying to make sense of how financial firms are actually being rewired, this book is a field guide.

A word on tone. The financial industry is not always well served by breathless language about AI. Every generation of technology promises to transform finance, and every generation delivers a mix of real progress and hard trade-offs. Real-time gross settlement was a genuine breakthrough. High-frequency trading was, on balance, a mixed blessing. Blockchain was oversold. Cloud is, at long last, being used as more than a lift-and-shift. AI, in my view, is somewhere between electrification and the internet in terms of the magnitude of its eventual impact on financial services, but that impact will only be captured by institutions that treat AI as an infrastructure problem first and a modeling problem second. Institutions that do it the other way around —

chasing model demos while ignoring the platform — will find themselves with impressive prototypes that never make it into production, and with production systems that cannot be safely improved.

The goal of this book is to help you avoid that fate. If you finish it with a clearer sense of what to build, what to buy, what to change about how your organization operates, and what to expect from the next few years, then it will have done its job.

The institutions that will lead financial services in the next decade are not those with the best models — models are increasingly commoditized — but those with the best infrastructure for building, deploying, monitoring, and governing models at scale. A team of talented data scientists at a bank with poor infrastructure will produce fewer deployed models per year than a smaller team at a bank with excellent infrastructure. The leverage provided by good infrastructure is not incremental; it is exponential.

This book is about that infrastructure. It is written for the people who will build, fund, and operate it: CTOs, CIOs, heads of data, heads of AI, infrastructure architects, and the senior business leaders who sponsor these programs. It assumes familiarity with financial services but not necessarily with the technical details of AI infrastructure. The technical depth is sufficient for practitioners while remaining accessible to decision-makers who need to understand what they are funding and why.

The book is organized in two parts. Part I (Chapters 1-3) diagnoses the problem: the current state of financial AI infrastructure, why existing approaches have failed, and the compounding cost of continued inaction. These chapters build the case for change. Part II (Chapters 4-10) supplies the solution: the principles, architecture, implementation, technology choices, operations, scaling, and best practices for a modern AI platform in financial services. Chapter 10 presents real-world case studies from institutions that have done this successfully — and lessons from those that have not.

Chapter 1: The State of Play and Its Limits

To understand why the AI infrastructure of tomorrow needs to look so different from the financial infrastructure of today, you have to be honest about the infrastructure of today. Most of the world's regulated financial services still run on a substrate that was designed in an era when the fastest reliable way to move data was a mainframe channel and the fastest reliable way to change behavior was to rewrite a COBOL module. That substrate has been extended, wrapped, front-ended, and cloud-hosted, but the assumptions baked into it — batch processing, deterministic rules, tightly coupled systems, and a very small number of very authoritative databases — still shape how work gets done. AI workloads violate almost every one of those assumptions.

PRACTICAL INSIGHT Before deploying AI in finance, map every regulatory requirement to a technical control. Most AI failures in finance are compliance failures, not technology failures. A model that works perfectly but cannot explain its decisions to regulators is a failed deployment.

The single most consequential design choice in traditional financial infrastructure is the reliance on batch processing. Ledgers reconcile at end of day. Regulatory reports assemble overnight. Risk aggregations run in windows that end after the last trading desk closes and finish before the first desk opens the next morning. Customer statements generate on a monthly cadence. Fraud rules refresh on a weekly cadence. Even in "real-time" systems, the underlying data pipelines that support the models being served in real time are usually being refreshed hourly at best and nightly at worst.

Batch is not a defect. Batch is an elegant answer to a specific problem — how to guarantee consistency across many systems when the cost of coordination is high — and it will remain a legitimate architecture pattern for as long as there is such a thing as a financial close. The problem is that AI does not respect the batch cadence. A fraud model trained on data that is twenty-four hours old is a model trained on a world that no longer exists. A credit model that only sees a customer's income once a month cannot react to the customer being laid off yesterday. A market-making model that only sees liquidity as of the previous session cannot price the sudden appearance of a new order book. Batch-first infrastructure means AI-second results.

The practical refinement is that freshness is not one requirement but a spectrum, and the infrastructure should price it accordingly. Not every decision needs streaming data: a quarterly-basis portfolio strategy does not benefit from a millisecond feature feed, and paying for one is a cost error, not an engineering achievement. The design discipline is to classify every model by

its decision latency — the interval at which its inputs go stale enough to change its answer — and then give each class exactly the pipeline it needs: streaming capture for transaction-timed decisions, micro-batch for intraday, nightly for reporting-adjacent workloads. The failure mode to avoid is the uniform pipeline, which either starves the fast models of freshness or taxes the slow ones with infrastructure they never asked for. When freshness is treated as a per-workload property with a per-workload cost, the batch estate stops being the villain of the story and becomes one tier in a deliberately tiered system.

The second consequential design choice is the silo. Financial institutions are organized by product line — cards, deposits, mortgages, wealth, capital markets — and by function — front office, middle office, back office, finance, risk, compliance. Each of those units accumulated its own systems over decades, each optimized for its own workflow, and each with its own view of the customer, the counterparty, and the position. A single customer may be a debit card holder in one system, a savings account holder in another, a mortgage borrower in a third, a wealth client in a fourth, and a small-business owner in a fifth, with no reliable primary key stitching them together.

For rule-based systems, silos are annoying but survivable — you write another interface, another feed, another reconciliation. For AI systems, silos are almost fatal. Machine learning models are only as good as their features, and features come from data, and data trapped in a silo cannot be a feature. Every institution I have worked with can name three or four models that could not be built, or that had to be built with severely degraded performance, because the data they needed was locked in a system that could not be integrated on a reasonable timeline.

The third consequential design choice is the compute footprint. Traditional financial workloads are dominated by transaction processing — millions of small, latency-sensitive, CPU-bound, memory-hungry operations against relational databases — and by batch analytics — long-running, CPU-bound, IO-heavy jobs against columnar stores. Compute has been sized accordingly: lots of general-purpose CPU, lots of DRAM, lots of tiered storage, and networking optimized for east-west database chatter and north-south client traffic.

AI workloads have almost none of these properties. Model training is dominated by dense matrix multiplication, which is why GPUs and increasingly specialized accelerators (tensor processors and an expanding family of custom inference accelerators from the cloud vendartup accelerators) have taken over. Large-model inference is dominated by memory bandwidth and by inter-accelerator communication, which is why fabrics like NVLink and InfiniBand

matter far more than they do for a payment system. Feature engineering pipelines are dominated by shuffle-heavy distributed compute, which is why distributed data processing engines, batch and streaming alike, have replaced hand-built extraction pipelines for large-scale ML preparation. Vector search — the backbone of retrieval-augmented generation and modern semantic search — is dominated by high-throughput approximate nearest neighbor computations against billion-scale embedding stores, which is not a workload any traditional database was designed for. Retrofitting a traditional financial compute estate to serve these workloads is possible, but it is a slow and expensive way to arrive at a mediocre answer.

The honest framing is not "CPU bad, GPU good" but workload-class separation. The transaction estate and the AI estate are both legitimate; they simply fail differently under each other's assumptions, and the mistake is running them on shared foundations sized for neither. A database server is wasted on matrix multiplication, and an accelerator cluster is wasted on ledger writes. The architecture that ages well gives each workload class its own tier — capacity, interconnect, scheduling, and failure handling — and connects them with deliberate, contracted interfaces rather than accidental proximity. The subsidiary benefit of separation is operational: the blast radius of a training job that saturates its fabric ends at the tier boundary, and the capacity conversation for one class never has to be re-litigated against the other.

Institutions that skip this separation spend their hardest years negotiating between workloads that were never in conflict until they were forced to share.

The fourth consequential design choice is the governance model. Every regulated financial institution has, at great cost, built a governance regime around its existing systems: change management, model risk management, segregation of duties, entitlement controls, audit trails, disaster recovery, business continuity, and a thick layer of policy. That regime works, in the sense that it is largely tolerated by regulators and largely respected by employees. It does not work well for AI, because AI systems change behavior without a code change — retraining a model with new data can change its predictions dramatically without touching a single line of source — and because AI systems make decisions whose logic cannot be trivially inspected by a human.

The result is that governance becomes the rate-limiting step for AI adoption in most large financial institutions. Data scientists spend more time filling out model risk documentation than building models. Platform teams spend more time proving that their infrastructure meets policy than improving it. The governance debt is not a defect in governance — most of it is warranted — it is a symptom of governance built for one operating model being asked to

accommodate another.

The fifth consequential design choice is the talent bet. Financial institutions have historically hired for depth in a small number of narrow specialties — core banking, market data, trade execution, treasury systems — and for breadth in project management and business analysis. The talent required to run AI infrastructure at scale — ML platform engineers, distributed systems engineers with accelerator experience, MLOps engineers, data engineers fluent in streaming architectures, prompt and evaluation engineers, applied research scientists — is scarce in every industry and even scarcer in an industry that competes with hyperscalers, AI-native startups, and quantitative hedge funds for the same people.

This is not solvable purely by hiring. It is solvable by building infrastructure that gives ordinary talent superpowers: platforms where the difficult things are done once by the specialists and then made easy for everyone else. That is the north star for the rest of this book, and it is the aspect of AI infrastructure most often missed when firms evaluate their options.

The sixth consequential design choice is vendor concentration. For decades, financial technology has been dominated by a small number of very large vendors — core banking, market data, trade order management, custody, general ledger — whose products are essential, whose contracts are decade-long, and whose innovation cadences are measured in years. That model produces stability, and stability is genuinely valuable. It also produces stagnation, and stagnation is fatal in a domain that is being reshaped by open-source foundation models, hyperscaler-native services, and continuous release cycles.

Every financial institution is now navigating an uncomfortable transition: from an infrastructure dominated by a small number of long-lived vendor contracts to an infrastructure that blends vendor platforms, hyperscaler services, and internally built components, each on a different lifecycle. The AI infrastructure of tomorrow is not a monolith you buy; it is a portfolio you assemble, and the assembly discipline is itself a capability you have to build.

The current state of financial infrastructure is not broken. It is working, in most institutions, well enough to keep the lights on and to satisfy regulators. But it is not the substrate on which AI-native financial services can be built at scale. Every one of the six design choices above — batch, silos, general-purpose compute, legacy governance, narrow talent, vendor concentration — was rational when it was made, and each of them is now an obstacle. The work of the next decade is to replace or reroute enough of them that AI stops being a bolt-on to existing systems and starts being the default way new financial

services are built.

The global shift to real-time payment systems is reshaping the infrastructure requirements for financial AI. Instant payment settlement has now arrived in nearly every major economy, with national rails clearing transfers in seconds rather than days, and any institution building payment intelligence must treat the real-time rails as its default operating environment. These systems process transactions in seconds rather than days, meaning fraud detection, AML screening, and risk decisions must also be made in seconds — or milliseconds.

Legacy fraud detection architectures were designed for batch processing — transactions accumulate during the day and are screened overnight. Real-time payment systems make this approach obsolete. A fraudulent transaction in a real-time system is settled before the batch screening runs, and the money is gone. The infrastructure must shift from batch screening to real-time, streaming evaluation of every transaction as it occurs, which requires fundamentally different data architectures (streaming rather than batch), model architectures (low-latency inference rather than batch scoring), and operational architectures (always-on rather than scheduled).

Open banking regulations (PSD2 in Europe, CDR in Australia, Open Banking in the UK and Brazil) require banks to share customer data with authorized third parties through standardized APIs. This creates both a data outflow (customer data shared with fintech competitors) and a data inflow (transaction data from other institutions, merchant data, and alternative data that can enrich AI models).

The infrastructure challenge is managing the API-driven data ecosystem. Data shared through open banking APIs must be consistent, accurate, and timely — discrepancies create regulatory and customer trust issues. The platform must handle API rate limiting, data consent management (tracking which third parties are authorized to access which data), and data standardization (normalizing data from multiple sources into a unified format). Institutions that treat open banking as a compliance exercise miss the strategic opportunity: the richest data ecosystem wins, and the institution with the best AI infrastructure for ingesting, normalizing, and exploiting open banking data will be the institution that offers the best customer experience.

Digital assets — cryptocurrencies, tokenized securities, stablecoins, and central bank digital currencies (CBDCs) — represent a new asset class that requires new AI infrastructure. The data structures are different (blockchain-based rather than relational), the trading patterns are different (24/7, global, high-frequency), and the risk models are different (volatility profiles, custody risks, smart contract risks that have no analog in traditional finance).

Institutions that want to participate in digital asset markets need AI infrastructure that can handle these new data types and risk patterns. This includes real-time monitoring of blockchain transactions, anomaly detection for DeFi protocols, market manipulation detection in crypto markets, and risk modeling for digital asset custody. The infrastructure gap is significant — most financial institutions' AI platforms were designed for traditional asset classes and cannot easily accommodate the characteristics of digital assets. Building this capability is a multi-year program that should start before the business need becomes urgent.

KEY TAKEAWAYS - Traditional financial workloads (transaction processing + batch analytics) are fundamentally different from AI workloads (vector search, neural inference). - AI infrastructure requires real-time, high-throughput vector computation — not a workload any traditional financial database was designed for. - The reliance on batch processing in finance must evolve to support real-time AI inference.

End of free sample. The full book is available at putrabooks.github.io

ABOUT THE AUTHOR

Aiman JoeHanz takes machines apart on paper. As the author of the Building Infrastructure of Tomorrow series, he spends his chapters tracing the load paths of the systems people trust with their money, their time, and their safety. This first volume of the finance journey opens the machinery that has crept into the heart of banking: the models, the pipelines, and the quiet decision engines that approve, deny, price, and watch.