



Das GenAI Playbook

GenAI für Wealth Data bauen

Wie aus komplexen, fragmentierten Portfolios eine sichere, nachvollziehbare und tatsächlich nützliche dialogische Anwendung wird. Die technischen und strategischen Entscheidungen dahinter.

Dieses Playbook fasst die technischen und strategischen Entscheidungen zusammen, die Héctor Borobia, Head of AI bei Flanks, in der Interview-Reihe [IA lo hicieron](#) von LambdaLoopers geteilt hat.

01 Von manuellem Reporting zu Chat-to-Data im Finanzbereich

Flanks hat die **Wealth-Data-Infrastruktur** hinter einer neuen KI-Generation gebaut. Die modulare Plattform automatisiert die Aggregation und Anreicherung komplexer, fragmentierter Portfolios und macht daraus eine **einzigste Datenschicht hoher Qualität**, eine echte 360-Grad-Sicht über jede Anlageklasse. Das Ziel ist es, *Vermögensverwaltung zu demokratisieren*: manuelle Reibung herauszunehmen, damit Family Offices, Privatbanken, Asset Manager und Fintechs sich auf schnellere, klügere und persönlichere Beratung konzentrieren können.



Vorher

- Manuelle Portfolioabfragen
- Excel-Exporte und Ad-hoc-Reports
- Abhängigkeit von manueller technischer Analyse
- Hohe Reibung beim Vergleich über Portfolios, Währungen und Benchmarks



Nachher

- Fragen zu Portfolios in natürlicher Sprache
- Analyse des Währungsexposures in Echtzeit
- Automatischer Vergleich gegen Benchmarks wie den S&P 500
- Reporterstellung nahezu in Echtzeit

Zentrale Erkenntnis

GenAI nimmt die Reibung heraus, mit der man an einen Wert kommt, den Flanks längst besaß: Finanzdaten hoher Qualität.

02

Produktives GenAI ist nicht nur KI, es ist Produkt, Daten, Software und Fachdomäne

Der schwierige Teil war nie, ein Modell über eine API aufzurufen. Es ging darum, herauszuarbeiten, **welche Daten der Berater tatsächlich braucht**, wo das Risiko sitzt und wie man eine Antwort zurückgibt, der er vertrauen kann. Fünf Disziplinen müssen als eine arbeiten.



Funktionale Discovery

Präzise definieren, welche Fragen das System beantworten soll und welche nicht.



Datendesign

Wissen, welche Daten existieren, wo sie liegen und in welcher Qualität, bevor modelliert wird.



Software Engineering

APIs, Tools, Traces und bestehende Systeme belastbar integrieren.



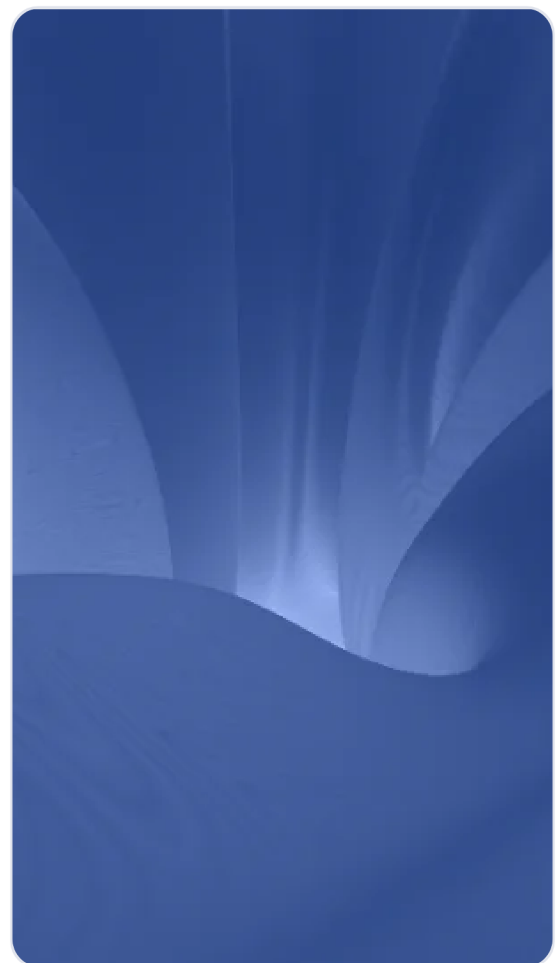
Finanzdomäne

Portfolios, Benchmarks, Risiko und Reporting mit Urteilsvermögen interpretieren.



Dialogische UX

Mehrdeutigkeit und Rückfragen sauber behandeln und Vertrauen in die Antwort aufbauen.



Zentrale
Entscheidung

GenAI als **Produkt- und Datenprojekt** behandeln, nicht als einmaliges KI-Experiment.

03

Das Modell zählt, aber weniger als das Datensystem darum herum

Ein Modell zu wählen ist die einfache Entscheidung. Der dauerhafte Vorteil liegt in dem, was man darum herum baut: der Kontext, den man einspeist, die Tools, die man anbindet, und wie souverän man mit Mehrdeutigkeit umgeht.

Modellauswahl

Mit einem großen Modell starten, um Variablen zu isolieren. Nicht nach generischen Benchmarks entscheiden, sondern Qualität, Latenz, Kosten und korrekte Tool-Nutzung messen, und für unterschiedliche Aufgaben unterschiedliche Modelle einsetzen.

Frameworks

LangGraph, LlamaIndex, CrewAI und Agent-SDKs übernehmen Boilerplate und State Graphs. Was sie *nicht* übernehmen, ist der Entwurf Ihrer Produktarchitektur.

Context Engineering

Prompting entscheidet, **welcher Kontext in welcher Reihenfolge eingeht** und welche Tools angebunden sind. Mehr Kontext ist nicht automatisch besser, er bringt Rauschen, Kosten und Fehler mit.

Dialogische UX

Ist eine Frage mehrdeutig, soll das System fragen, nicht raten. Wird eine Aktie an mehreren Börsen gehandelt, ist die richtige Reaktion: „Welchen Markt meinen Sie?“

Zentrale Entscheidung

Nicht „*dem besten Modell*“ nachjagen. Das **richtige Modell für die jeweilige Aufgabe** wählen, innerhalb einer Architektur, die man messen kann.

04

Tools sind die Grenze zwischen probabilistischer KI und deterministischem Geschäft

Ein produktionsreifes Tool muss so gebaut sein, dass das Modell **weiß, dass es existiert, und weiß, wann es danach greift**, saubere Argumente erhält und sich sauber erholen kann, wenn etwas bricht.



Semantisches Design

Leicht interpretierbare Eingaben, strukturierte Ausgaben, klare Funktionsbeschreibungen und Berechtigungen, definiert je Nutzer und Datendomäne.



Funktionale Fehler

Keine Daten für ein bestimmtes Jahr? Das Tool wirft keinen „error“, es antwortet „*Keine Werte für diesen Zeitraum, probieren Sie einen anderen Bereich*“, sodass das Modell einfach erneut ansetzen kann.



Deterministische Architektur

Kritische Logik liegt in **kontrollierten Tools**. Das LLM interpretiert und formuliert, die Berechnungen, Berechtigungen und echten Daten kommen aus deterministischen Systemen.



Vorfilterung und Validierung

Programmatische Validierungen vor dem Modell. Feldbezogene Vorfilterung, um unnötiges Rauschen zu vermeiden. Latenz wird je Tool gemessen.

Zentrale Erkenntnis

Ein schlecht entworfenes Tool lässt den ganzen Agenten unbeholfen wirken. Ein **gut entworfenes** nimmt den größten Teil des Systemrisikos still heraus.

05

Ein Protokoll, um jedes Modell an Ihre Daten und Tools anzubinden

Ein gutes MCP sollte mit mehreren Modellen funktionieren. Wenn es nur mit dem leistungsfähigsten läuft, **ist es wahrscheinlich falsch spezifiziert.**

MCP als Infrastrukturprodukt behandeln

+ Was MCP bringt

- Standardisierte Verbindung zwischen Modellen, Tools und Daten
- Wiederverwendbare Finanzfunktionen, offen bereitgestellt
- Kunden binden ihre eigenen Agenten an Flanks-Daten an
- Trennung der Endoberfläche von der austauschbaren Datenschicht
- Ein möglicherweise neues Geschäftsmodell: die „Data Engine“

🔒 Verpflichtende Kontrollen

- OAuth, Access Tokens und Client-IDs
- Allow Lists und Auditing von Tool-Aufrufen
- Nutzungslimits und Latenz-Observability
- Mehr MCPs bedeuten mehr Kontext, mehr Kosten, mehr Rauschen

Zentrale Entscheidung

MCP als **Infrastrukturprodukt** behandeln, nicht bloß als technischen Konnektor.

06

Memory heißt nicht, mehr zu speichern, sondern den richtigen Kontext abzurufen

01 · Erfassen

**Working Memory**

Das laufende Gespräch, der Kontext des aktiven Session-Threads.

02 · Verdichten

**Zusammenfassung beim Abschluss**

Ein LLM erfasst die Kernpunkte und entscheidet, welche Kandidaten es wert sind, behalten zu werden.

03 · Speichern

**Long-Term Memory**

Meilensteine und Beratermuster: Benchmarks, bevorzugte Formate, Risikoprofil.

04 · Abrufen

**Kontextbezogenes Retrieval**

Ein Tool holt den nützlichen Kontext genau dann zurück, wenn der Nutzer ihn braucht.

„Die KI lernt, während man sie nutzt“ klingt verlockend und führt in die Irre. Ein eingefrorenes Modell lernt nicht von selbst. Die Anwendung wird besser, weil das System **den richtigen Kontext abrufen und einspeist**, im richtigen Moment, ganz ohne Retraining.

Zentrale Entscheidung

Zu **Kontext als Memory** greifen, bevor man zu Retraining greift, es sei denn, der ROI spricht klar dafür.

07

Nicht alles braucht Agenten, manchmal ist der beste KI-Ansatz ein Workflow

Orchestrierung beginnt mit einer klaren Frage: *Ist die Aufgabe deterministisch?* Wenn Eingaben und Schritte nie wechseln, brauchen Sie mit hoher Wahrscheinlichkeit kein Multi-Agent-System. Komplexität muss sich ihren Platz verdienen.

Workflow

Feste Abfolge

Vorhersehbare Eingaben und Schritte. Keine Entscheidung in Echtzeit nötig.

Einzelner Agent

Entscheidung in Echtzeit

Wenig Spezialisierung. ReAct oder einfache Muster genügen meist.

Multi-Agent

Spezialisierte Teilaufgaben

Nur für parallelisierbare, aggregierbare Teilaufgaben mit echter Spezialisierung. Mehr Agenten bedeuten mehr Fehlerfläche.

Modulares Design

Flanks arbeitet mit azyklischen Graphen und entkoppelten Komponenten und kann Tools oder Fähigkeiten ergänzen, ohne das Gesamtsystem neu zu entwerfen.

Klassisches ML als Verbündeter

Nicht alles sollte mit LLMs gelöst werden. Klassifizierer im BERT-Stil können bei wiederkehrenden, schnellen, deterministischen Aufgaben die bessere Wahl sein.

Zentrale Entscheidung

Die **Aufgabe** zwischen Workflow, einzelner Agenten und Multi-Agent entscheiden lassen, nicht den Hype.

08

Echter Produktivbetrieb heißt Traces, Evals, Guardrails und kontrollierte Kosten

AgentOps macht aus einer glatten Demo ein System, das man tatsächlich betreiben kann. Observability gehört in den POC, **nicht am Ende angeschraubt**.



Tracing

Eingaben, Tool-Aufrufe, Ausgaben, Tokens, Latenz je Schritt, Fehler, Retries, Loops und Timeouts instrumentieren.



Evals und Golden Set

Von Beginn an 15 bis 50 Fragen mit erwarteten Antworten anlegen. LLM-as-a-Judge mit einem anderen Modell nutzen, um im großen Maßstab zu bewerten.



Guardrails

Eingangsseitig: Fragen außerhalb der Domäne blockieren. Ausgangsseitig: unsichere Antworten blockieren. Verhalten: Loops und unerwünschte Aktionen begrenzen.



Feedback und FinOps

Nutzerfeedback fließt in Prompt- und Golden-Set-Verbesserungen, es trainiert das Modell nicht. Tokens, Aufrufe und Kosten je Kunde messen.

Zentrale Erkenntnis

Keine Traces, **kein Debugging**. Keine Evals, **keine Verbesserung**. Keine Guardrails, **keine operative Kontrolle**.

Sieben Entscheidungen, um GenAI im WealthTech in Produktion zu bringen

Die Kernlehre von Flanks: GenAI auszuliefern heißt nicht, ein LLM einzustecken. Es heißt, **das System darum herum zu entwerfen**.

01 Das Problem zuerst validieren

Von einem echten Schmerzpunkt ausgehen: die Reibung beim Zugriff auf komplexe Vermögensdaten reduzieren.

02 Klein anfangen, damit man messen kann

Ein Modell, ein Tool, wenige MCPs, ein abgegrenzter Fall. Wenn alles gleichzeitig scheitert, erkennt man nicht, was gebrochen ist.

03 Generierung von Logik trennen

Das LLM interpretiert und formuliert. Tools rechnen, validieren, fragen ab und setzen Berechtigungen durch.

04 Tools und MCPs als Produkt entwerfen

APIs bereitzustellen genügt nicht. Sie so entwerfen, dass Modelle sie verstehen, ohne die Sicherheit zu brechen.

05 Kontext als Wettbewerbsvorteil nutzen

Das Modell wird zur Ware. Der Unterschied liegt in Daten, Kontext, Memory und Berechtigungen.

06 Nicht mit Agenten überkonstruieren

Löst ein Workflow die Aufgabe, nehmen Sie einen Workflow. Multi-Agent nur bei echter Spezialisierung.

07 Ab dem POC instrumentieren

Tracing, Golden Set, Evals, Feedback, Guardrails und Kosten müssen existieren, bevor skaliert wird.

Das Modell ist nicht der Burggraben. Der Burggraben sind die Daten, die Tools, die Orchestrierung und die Fähigkeit, GenAI **unter Kontrolle zu betreiben.**



**Vermögen,
einfach verwaltet**

Sprechen wir!

Flanks.io