Content Concierge – Backend Intelligence & Personalization Report

1. Overview

This project implements the **backend intelligence layer** for a Content Concierge system designed to deliver **contextual, educational, and compliant investment insights** inside an investing application.

The system does **not give advice**.
Instead, it explains *what the user is seeing*, *why it may be happening*, and *how investors often interpret similar situations*, tailored to:

- **Who the user is** (experience & behavior)
- **Where they are in the app** (placement)
- **Why the insight is being shown now** (trigger)
- **What they are looking at** (focus ticker, if any)

The result is a **placement-aware, archetype-driven, non-advisory insight engine** suitable for a regulated financial environment.

---

2. Core Concepts & Definitions

2.1 Archetype (Who the user is *right now*)

An **archetype** represents the user's current behavioral state, not a permanent label.

Archetypes implemented:

- **INACTIVE** – low engagement / returning after a gap
- **EVERYDAY** – active, non-advanced investor
- **ADVANCED** – experienced investor (flagged via experience level)
- *(Prospect is planned, not yet implemented)*

How archetype is determined:

```
If inactivity_flag == true → INACTIVE
Else if investment_experience_level == "advanced" → ADVANCED
Else → EVERYDAY
```

Archetypes control **what kind of insight is appropriate**, not the UI.

2.2 Tier (How much wealth context the user has)

A **tier** is derived from total investable assets and represents *scale*, not sophistication.

Tier mapping:

- UNDER_250K
- FROM_250K_TO_1M
- OVER_1M

Used to:

- Shape tone
- Gate future sophistication
- Support business segmentation

Currently informational, but foundational for later prioritization.

2.3 Placement (Where the insight appears)

A **placement** represents the UI surface where an insight is embedded.

Placements implemented:

- **INVESTMENT_DASHBOARD** – portfolio-level orientation
- **POSITIONS** – holding-level inspection
- **PERFORMANCE** – interpretation of returns and risk

Placement determines:

- Scope (portfolio vs ticker)
- Bundle selection
- Insight framing

2.4 Trigger (Why the insight is shown now)

A **trigger** describes the user action or moment that caused the insight to be generated.

Examples:

- `APP_OPEN`
- `TAB_VIEW`
- `HOVER_TICKER`
- `DWELL_NO_ACTION` (future)
- `REPEAT_VIEW` (future)

Triggers do **not** change facts, but change *intent* and *timing*.

2.5 Focus Ticker (What the user is inspecting)

When the user is interacting with a specific holding (e.g., hovering over AAPL), the request includes:

```
"focus_ticker": "AAPL"
```

This enables:

- Ticker-scoped insights
- Provider filtering
- Contextual explanations

If absent, insights are portfolio-level.

---

3. The Insight Pipeline (End-to-End)

Step 1: Input Payload

The request includes:

- User profile
- Wealth snapshot
- Holdings
- Goals
- Activity summary
- Preferences
- **Request context** (placement, trigger, focus ticker)

Step 2: Normalization

Raw data is converted into a **normalized context**, including:

- Derived archetype
- Derived tier
- Holdings count
- Top holdings
- Dividend profile
- Goal progress
- Inactivity signals

This ensures **all downstream logic works off a clean, stable abstraction**, not raw schemas.

Step 3: Provider Fetch (Grounding)

External providers (e.g., market data, analyst context) are queried.

Outputs are:

- **Items** (neutral facts)
- **Citations** (for attribution)
- No opinions, no advice

Step 4: Bundle Planning (Core Intelligence)

This is the **heart of the system**.

A **bundle** is a structured container of facts meant for *one coherent insight*.

Each bundle has:

- `kind` – what type of insight it is
- `facts` – bullet-style truths the LLM may use
- `citations` – optional grounding sources

What bundles do:

- Decide *what* the LLM is allowed to say
- Prevent hallucination
- Enforce compliance boundaries

---

4. Bundles Implemented (signals.py)

4.1 Inactive Bundles

Purpose: Re-orient returning users.

Function:

`build_inactive_activation_signals()`

Facts include:

- User is inactive
- Difference between funding vs investing
- Diversification concepts
- Goal framing

Used on: Dashboard only

4.2 Everyday Bundles

Everyday – Performance
`build_everyday_performance_signals()`

Explains:

- How investors interpret performance
- Benchmark framing (educational only)
- Goal context

Everyday – Positions
`build_everyday_positions_signals()`

Explains:

- What investors look for when reviewing holdings
- News, earnings, macro context
- Concentration awareness

4.3 Advanced Bundles

Advanced – Performance
`build_advanced_performance_signals()`

Explains:

- Concentration signals
- Risk exposure
- Performance drivers

Advanced – Positions
`build_advanced_positions_signals()`

Explains:

- Catalysts
- Volatility context
- Scenario thinking

4.4 Market Trend Bundle

`build_market_trend_signals()`

Aggregates:

- Analyst sentiment
- Price context
- Thematic signals

Used as a **secondary context layer**, not the primary explanation.

---

5. Facts (What the LLM is allowed to say)

Facts are the most important safety construct in the system.

They are:

- Short, declarative statements
- Derived from user data or providers
- Non-prescriptive
- Non-predictive

Example facts:

- “Largest holding represents ~86% of tracked holdings value.”
- “Recent prices ranged between X and Y.”
- “User is viewing ticker AAPL.”

The LLM:

- **May only speak using these facts**
- Cannot invent numbers
- Cannot give advice

---

6. LLM Realize + Judge Pattern

Every insight passes through **two LLM steps**:

6.1 Realize

- Generates headline, explanation, personal relevance
- Uses facts only
- Educational tone enforced

6.2 Judge

- Independently classifies output as PASS / BLOCK
- Ensures no advice language
- Blocks imperatives or recommendations

This creates a **self-auditing system**.

---

7. Insight Types (kind_to_type)

Each bundle kind maps to an **InsightType** used by the UI:

Bundle Kind	InsightType
goal_portfolio	GOAL_PROGRESS
inactive_activation	PORTFOLIO_COMPOSITION
everyday_performance	PORTFOLIO_COMPOSITION
everyday_positions	PORTFOLIO_COMPOSITION
advanced_performance	PORTFOLIO_COMPOSITION
advanced_positions	PORTFOLIO_COMPOSITION
market_trend	MARKET_TREND
positions_ticker	MARKET_TREND

This separates **content intent** from **rendering logic**.

---

8. Scope & Priority

Each insight includes:

- **scope**
  - `PORTFOLIO` – whole account
  - `TICKER` – specific holding

- **priority**
    - Determines ordering when multiple insights exist

---

9. What This System Achieves

- Fully **placement-aware** insight generation
- **Behavior-driven personalization** (not static personas)
- Strict **non-advisory compliance**
- Deterministic, auditable outputs
- Clear extension points for:
    - Ranking
    - Benchmark math
    - Advisor triggers
    - Experimentation

---

10. What Is Intentionally Not Done (Yet)

- No buy/sell recommendations
- No ranking optimization
- No benchmark calculations
- No multimedia content
- No advisor escalation logic

These are **future layers**, not missing features.

---

11. Final Summary

This implementation delivers a **production-grade intelligence backbone** for a financial content concierge.
 It converts raw financial data into **contextual, compliant, and placement-specific insights**, driven by user behavior and experience rather than generic rules.

It is designed to scale safely into more advanced capabilities without re-architecting the system.