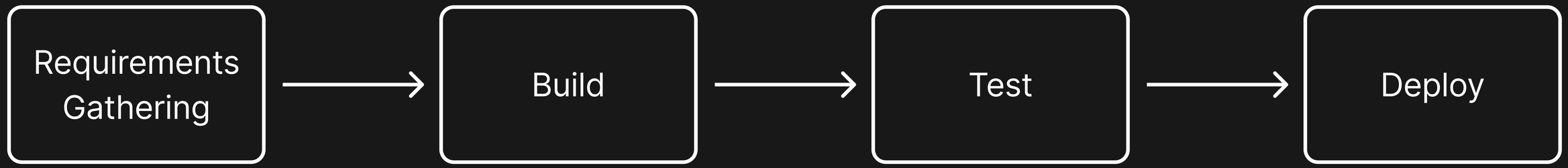
From PoC to Product

Surviving Generative AI

Traditional SDLC



Traditional SDLC

Classes of Problems → Solution Patterns → Predictable Product Value

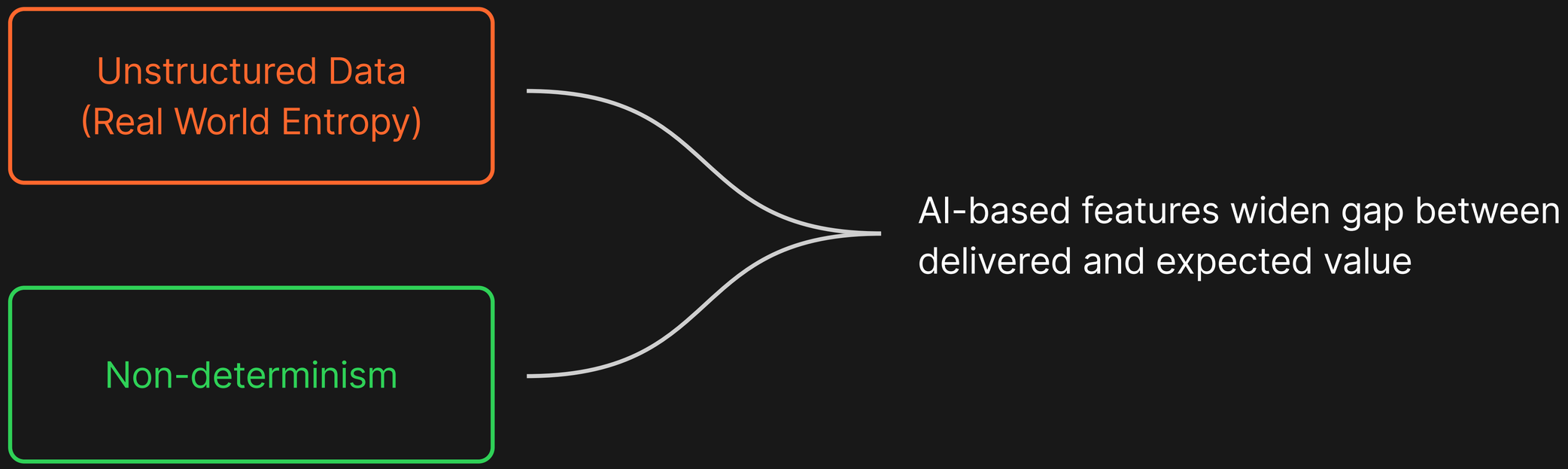
Traditional SDLC

Classes of Problems	Solution Patterns	Predictable Product Value
Read-only Accounts	RBAC	Users can’t edit

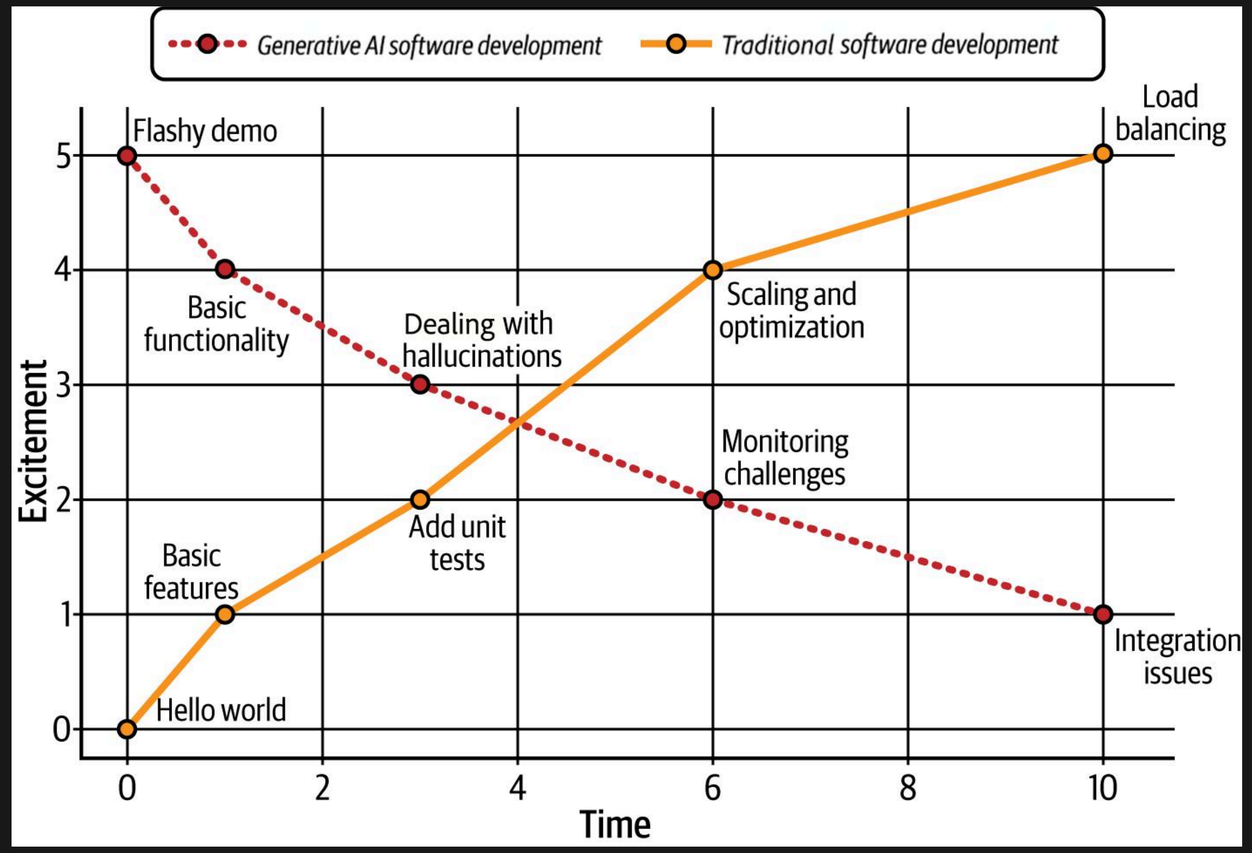
Delivered Value / Expected Value

AI-based features widen gap between delivered and expected value

Delivered Value / Expected Value



POC Purgatory



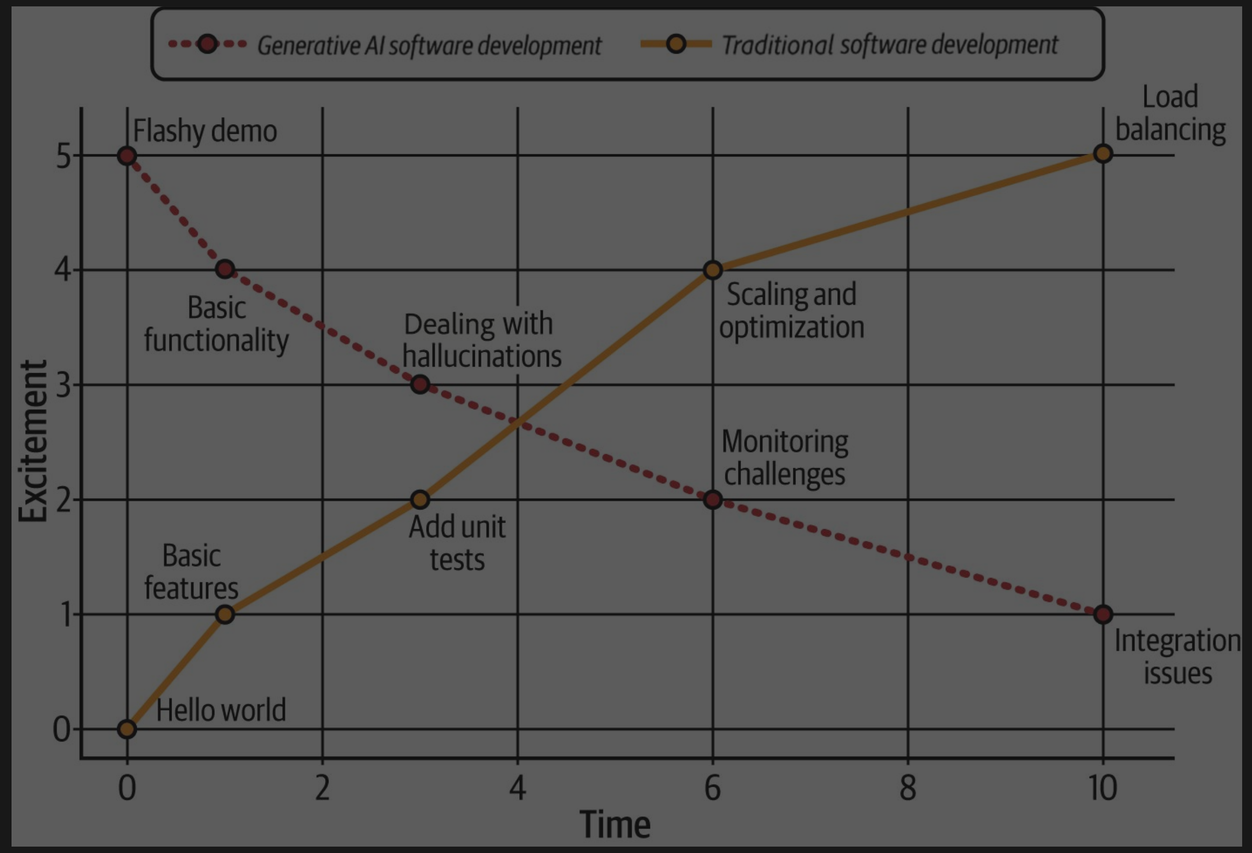
Traditional versus GenAI software: Excitement builds steadily—or crashes after the demo.

From GenAI POC to Product:

- Hallucinations
- Edge Cases
- Logging and Monitoring
- Iteration Cost
- Coordination Tax

https://www.oreilly.com/radar/escaping-poc-purgatory-evaluation-driven-development-for-ai-systems

POC Purgatory



Traditional versus GenAI software: Excitement builds steadily—or crashes after the demo.

From GenAI POC to Product:

- Hallucinations
- Edge Cases
- Logging and Monitoring
- Iteration Cost
- Coordination Tax

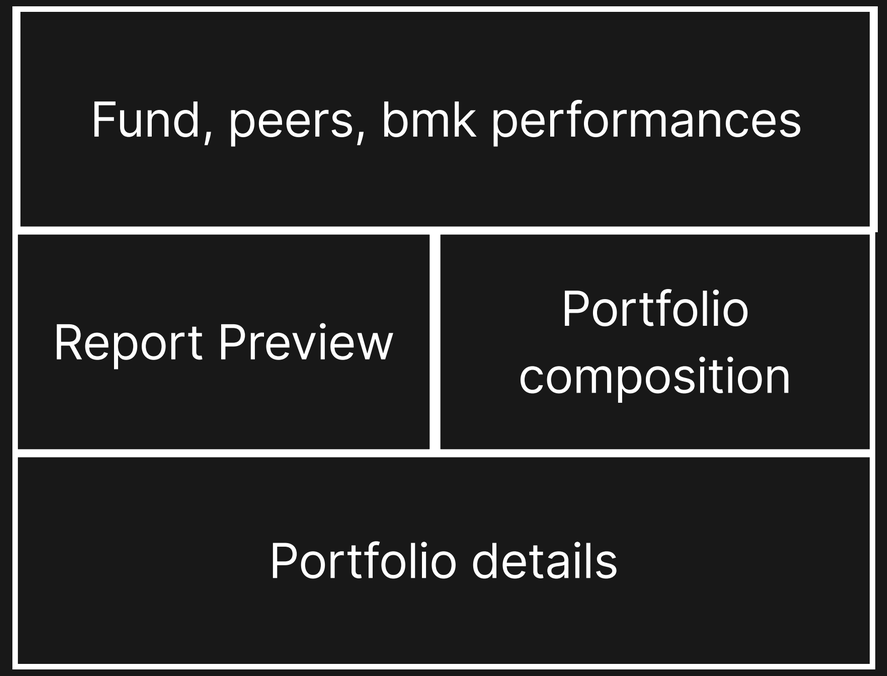
How can we escape this purgatory?

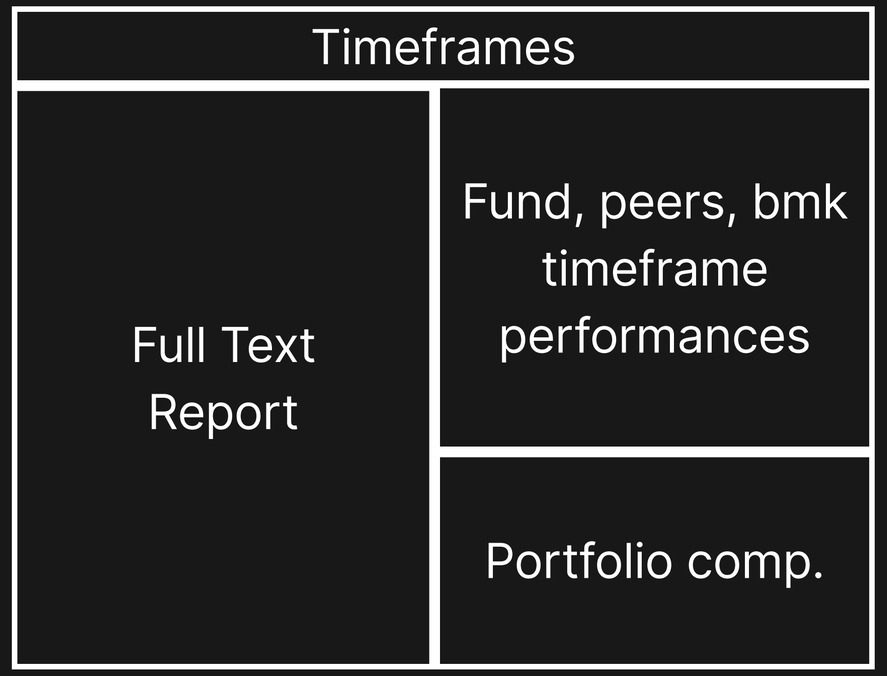
https://www.oreilly.com/radar/escaping-poc-purgatory-evaluation-driven-development-for-ai-systems


The Use Case
GRATIA ARTI
TRADE
MARK
©

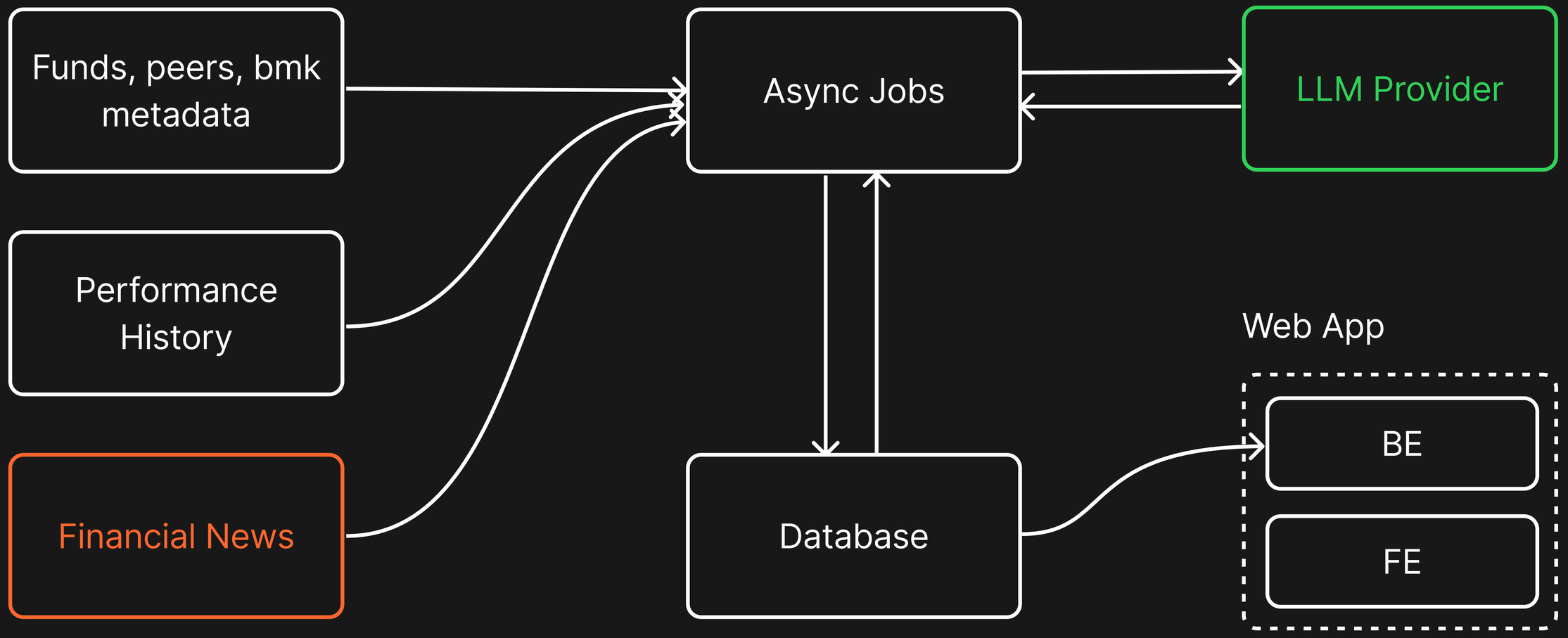
The Product

Objective
Streamline the Financial Managers job by automating the creation of funds reports

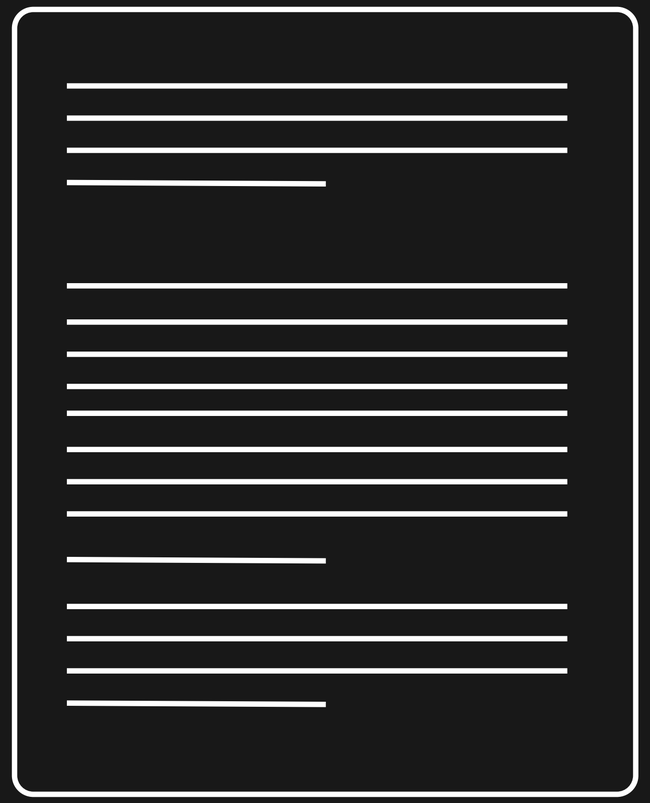




Architecture



Funds Reports



- One A4 page report per fund
- Multiple time frames per report (last month, YTD, last quarter, etc.)
- Hundreds of funds

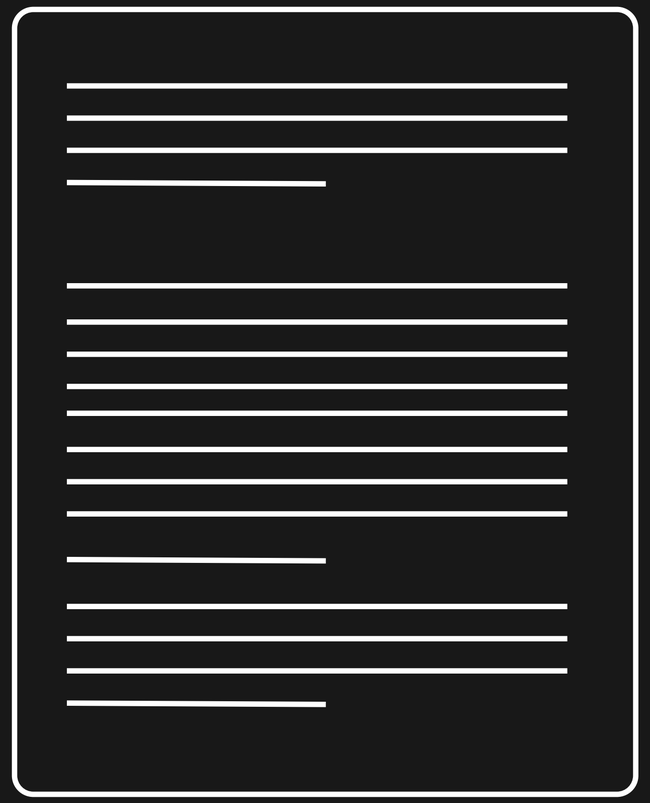
Ideal reports provided by SME?

Funds Reports

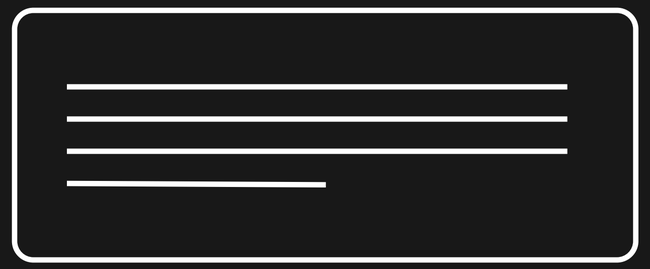
- One A4 page report per fund
- Multiple time frames per report (last month, YTD, last quarter, etc.)
- Hundreds of funds

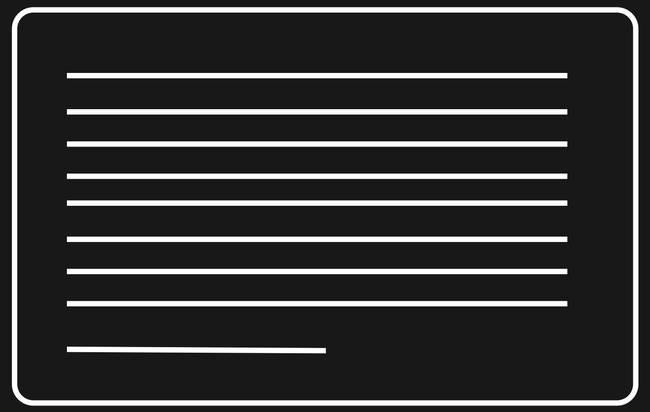
Ideal reports provided by SME? 22

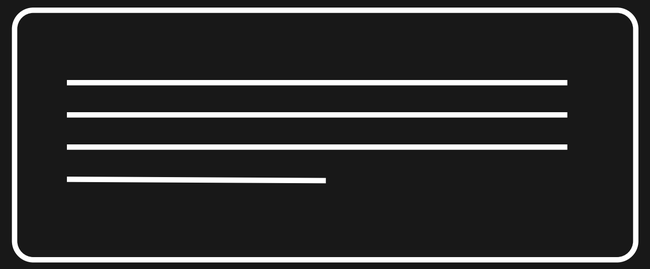
Segregation



Segregation



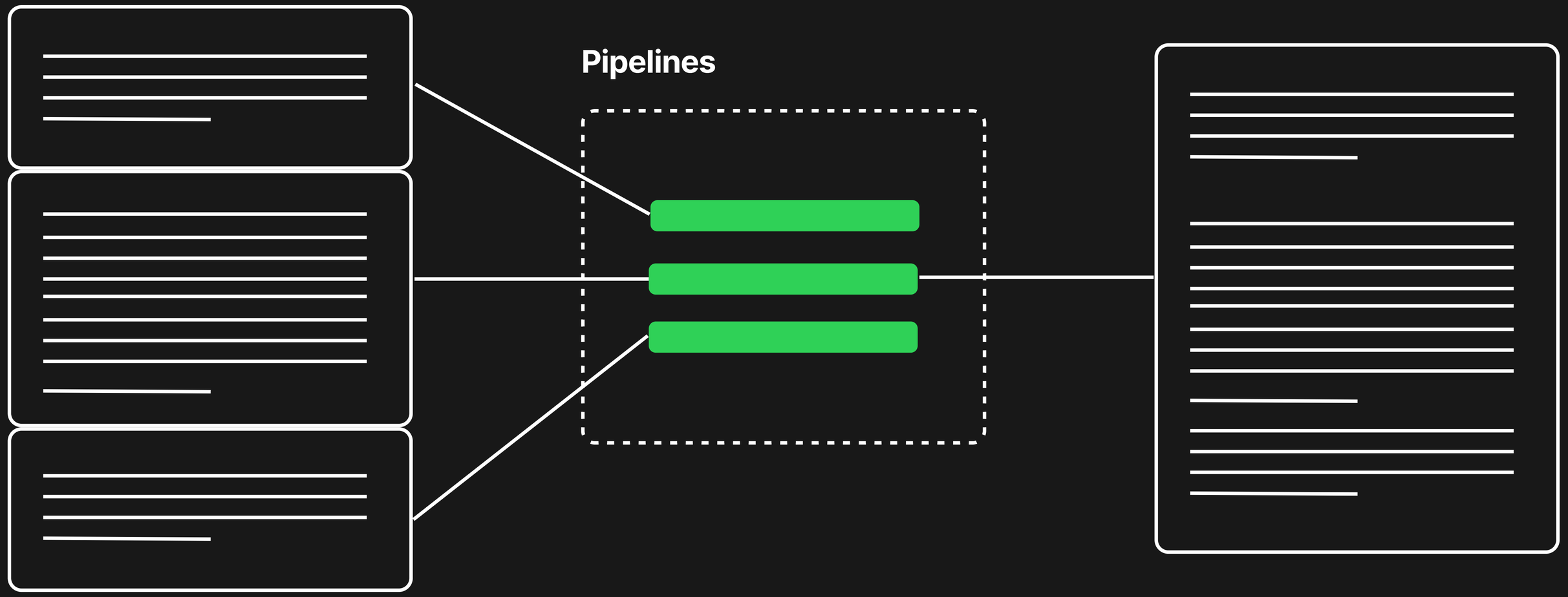




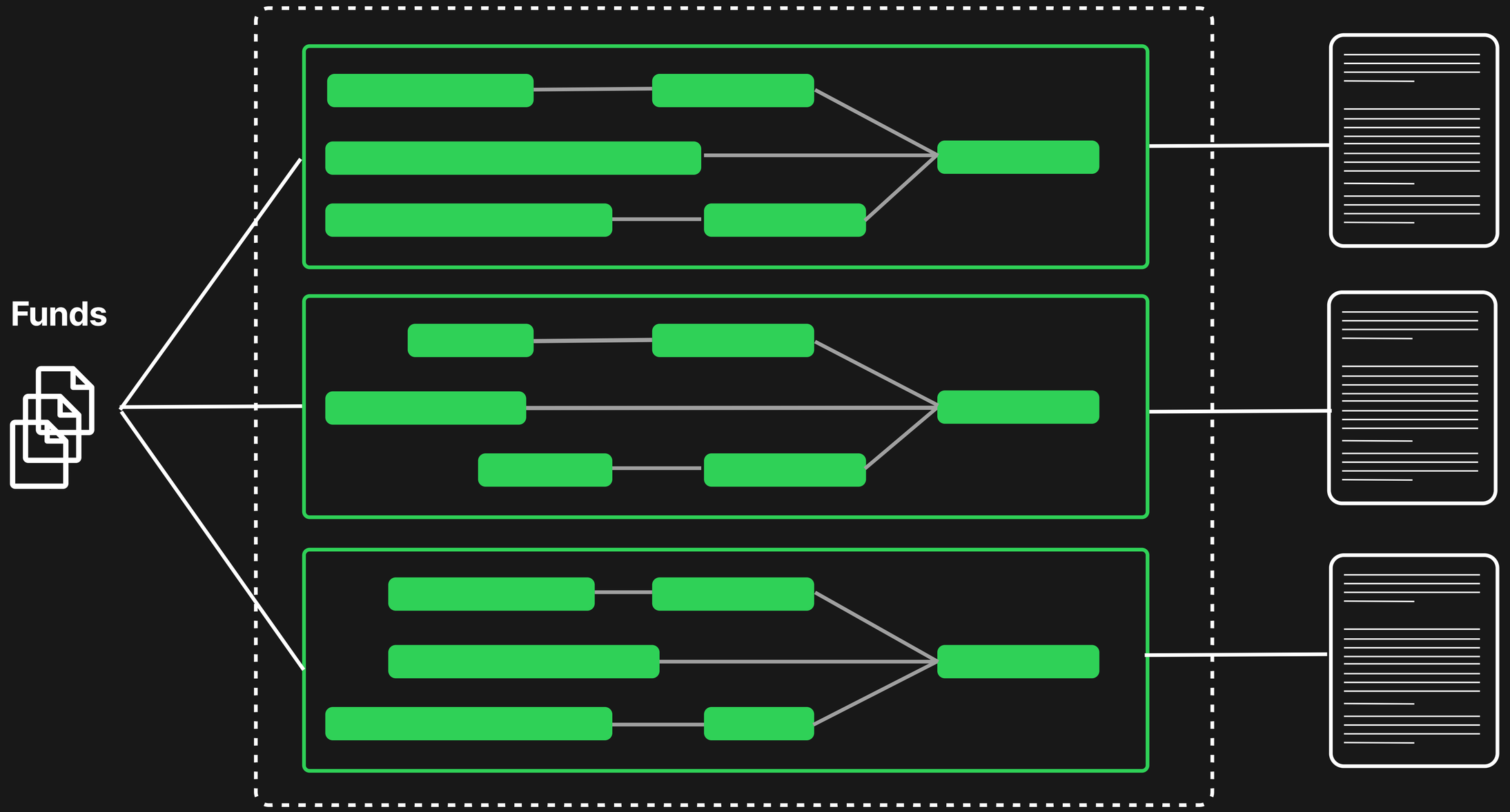
Independent life cycles:

- **Parallel team members workflows**
- **Parallel feedback loop**
- **Scoped focus**

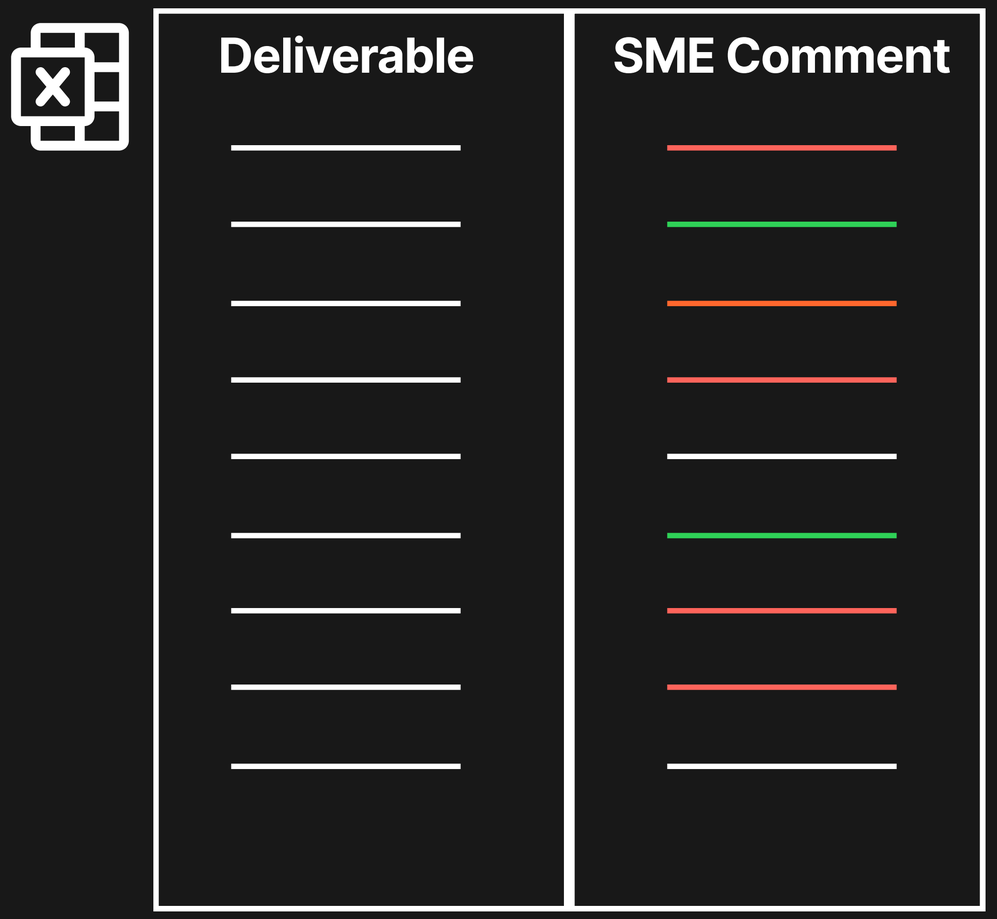
Segregation: Tooling



Prefect.io



Continuos Review: External Human

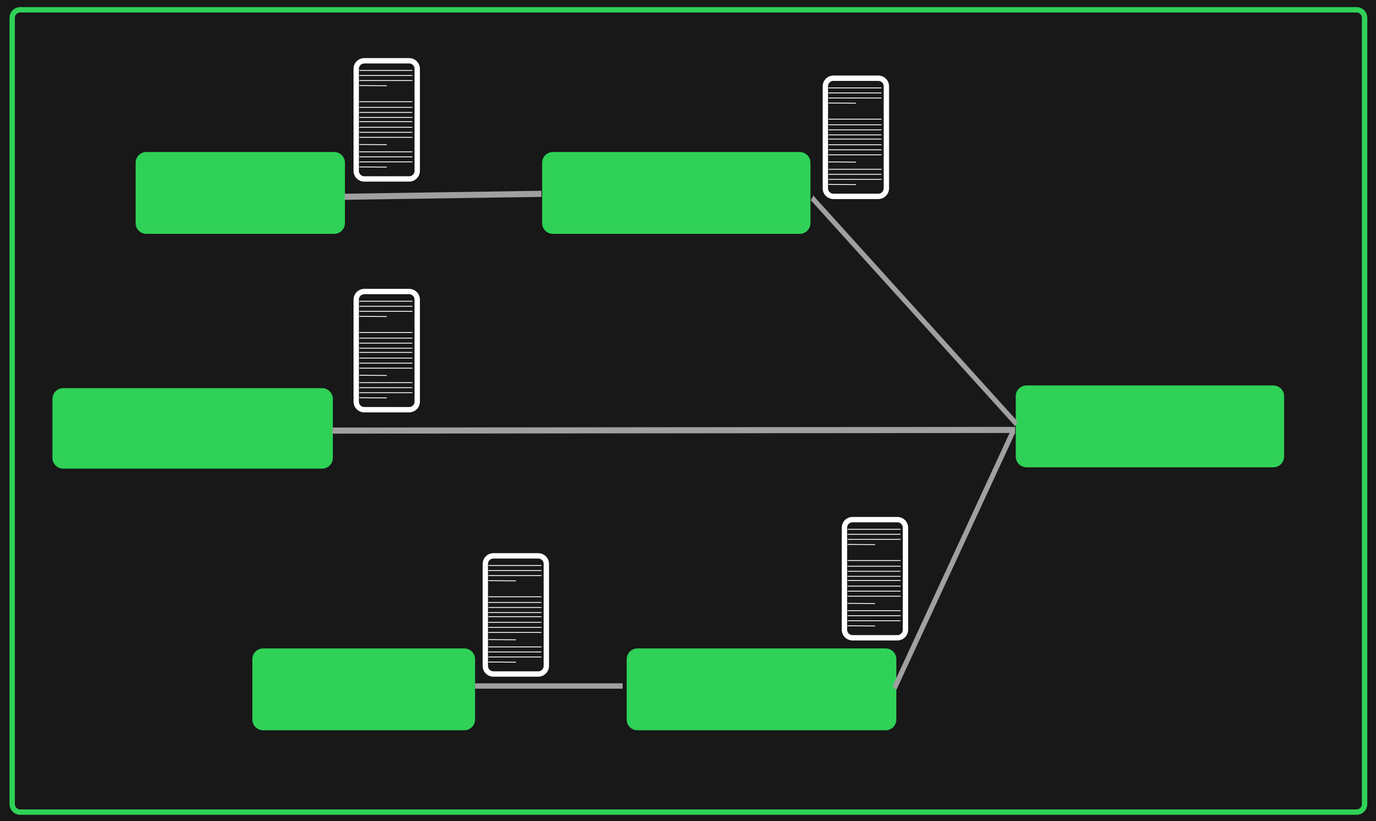


External human feedback loop:

- End-to-end
- Expensive
- Quality oriented
- End user alignment driven

Continuous Review: Internal Human

Prefect Artifacts



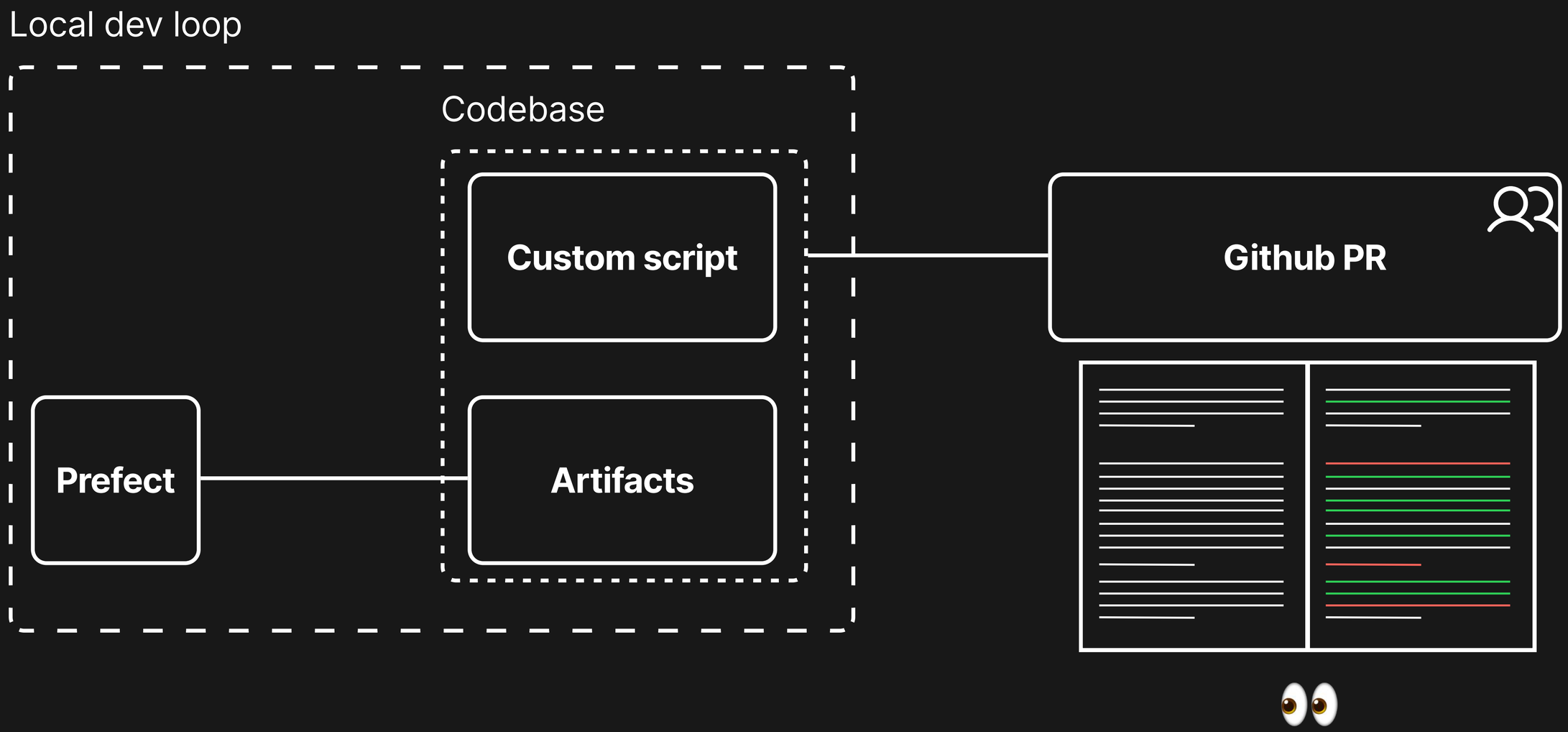
(data, prompt, output)

Internal human feedback loop:

- Internals
- Cheaper
- Quality oriented
- Product goals driven
- Formal regression barrier

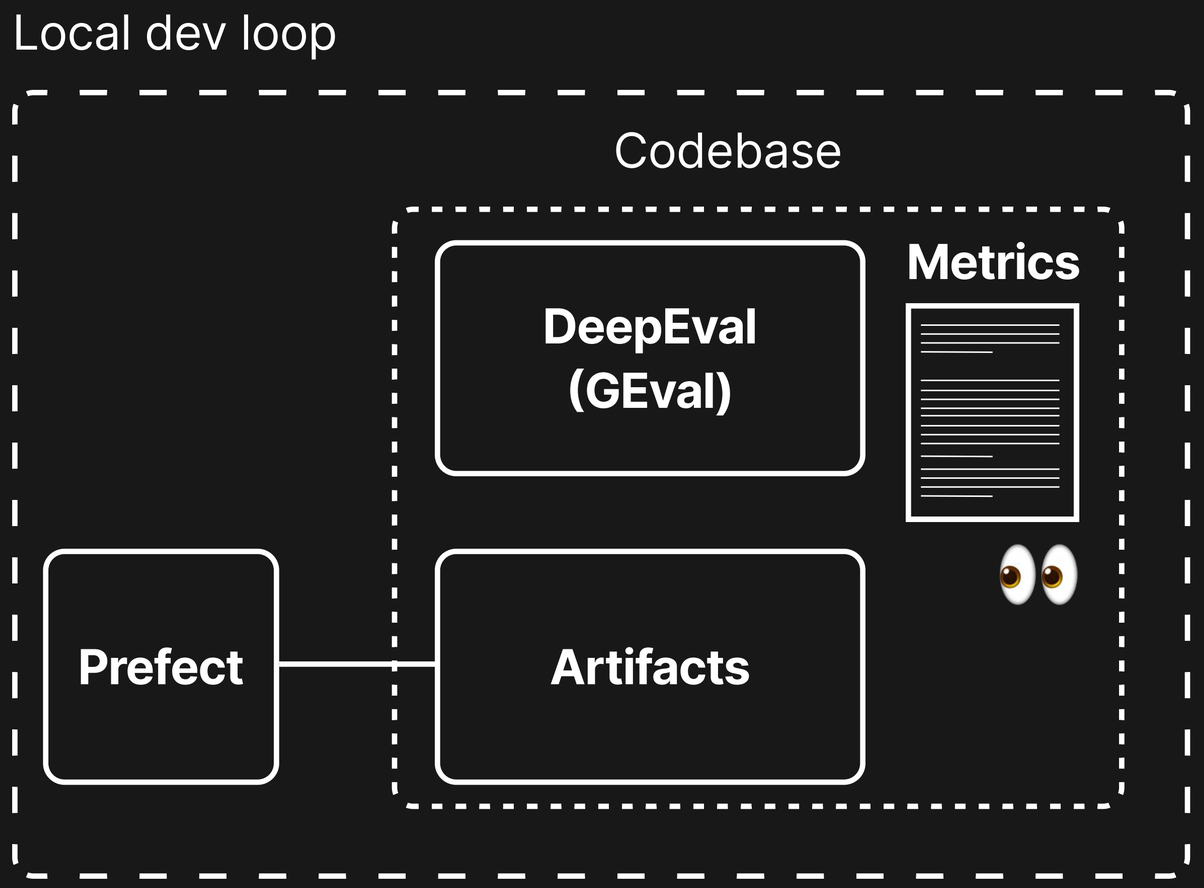
Continuous Review: Internal Human

Custom Tooling



Automatic Evaluation

DeepEval



Internal automated feedback loop:

- Internals
- Cheap
- Quantity oriented
- Automated regression barrier
- Human criteria

LLM-as-a-judge

How can we escape the POC purgatory?

How can we escape the POC purgatory?

Evaluation Driven Development

Evaluation Driven Development

Overview

We already have a POC:

- Vibes driven
- Works good in demos
- Not optimized (models, fine-tuning, etc.)

Evaluation Driven Development

Overview

We already have a POC:

- Vibes driven
- Works good at demos
- No optimized (models, fine-tuning, etc.)

We first want to:

- Define personas and scenarios
- Define success and to measure it
- Use synthetic data (SME driven)

Then we:

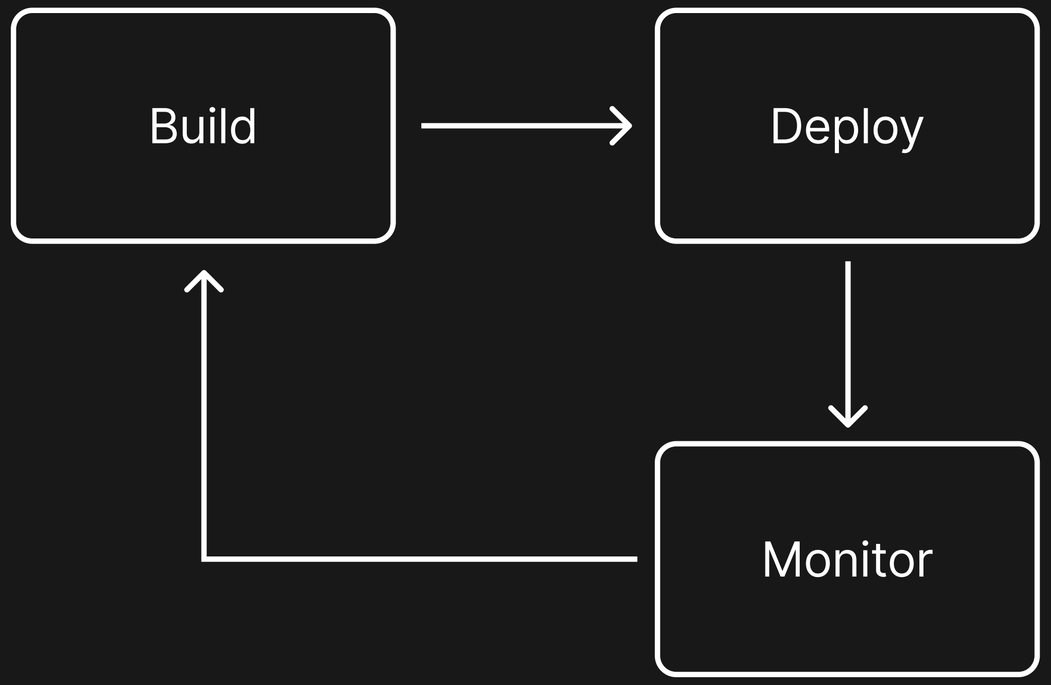
- Log and monitor
- Look at our data
- Iterate quickly

So we can finally:

- Setup an evaluation harness
- Iterate to align with business metrics

Evaluation Driven Development

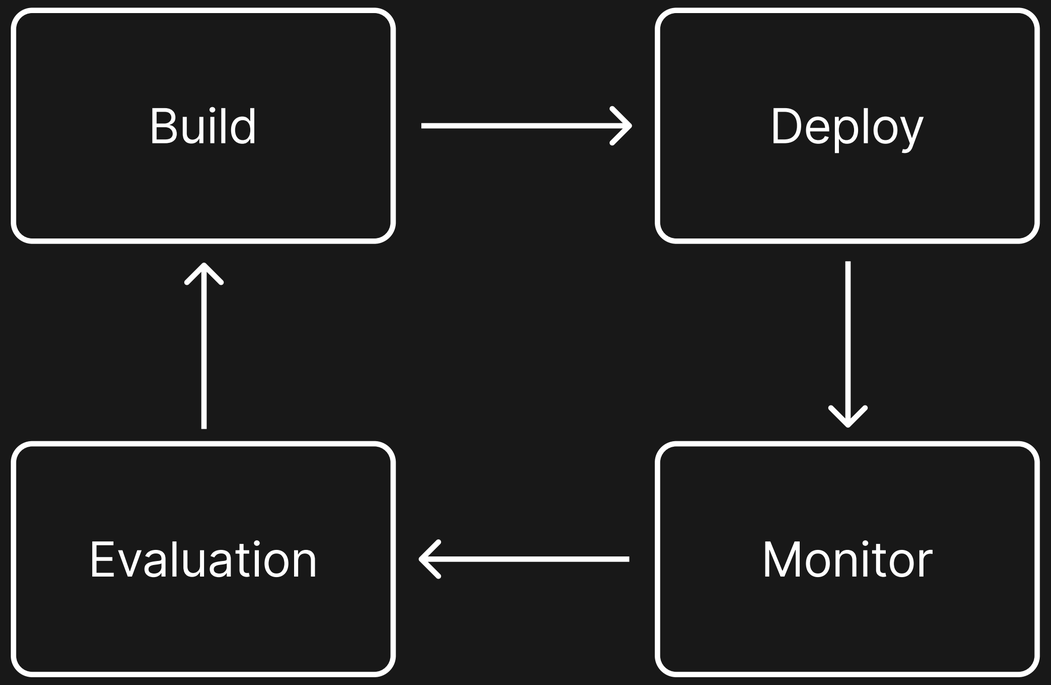
Look at Your Data



- Setup observability as a foundation
- Prompt engineering fast iterations
- Focus on prompts, not on models and fine-tunings
- Use synthetic data augmented from real one
- Internal usage catches major issues

Evaluation Driven Development

Eval Harness



Look at your data +

- Define a ground truth
- Work with Subject Matter Experts
- Use synthetic data filtered by SMEs
- Use evaluation as the steering wheel
  - Align with business metrics *based on* the evaluation outcomes

Evaluation Driven Development

Eval Harness

Scientific Approach

Look at your data +

- Define a ground truth
- Work with Subject Matter Experts
- Use synthetic data filtered by SMEs
- Use evaluation as the steering wheel
  - Align with business metrics *based on* the evaluation outcomes

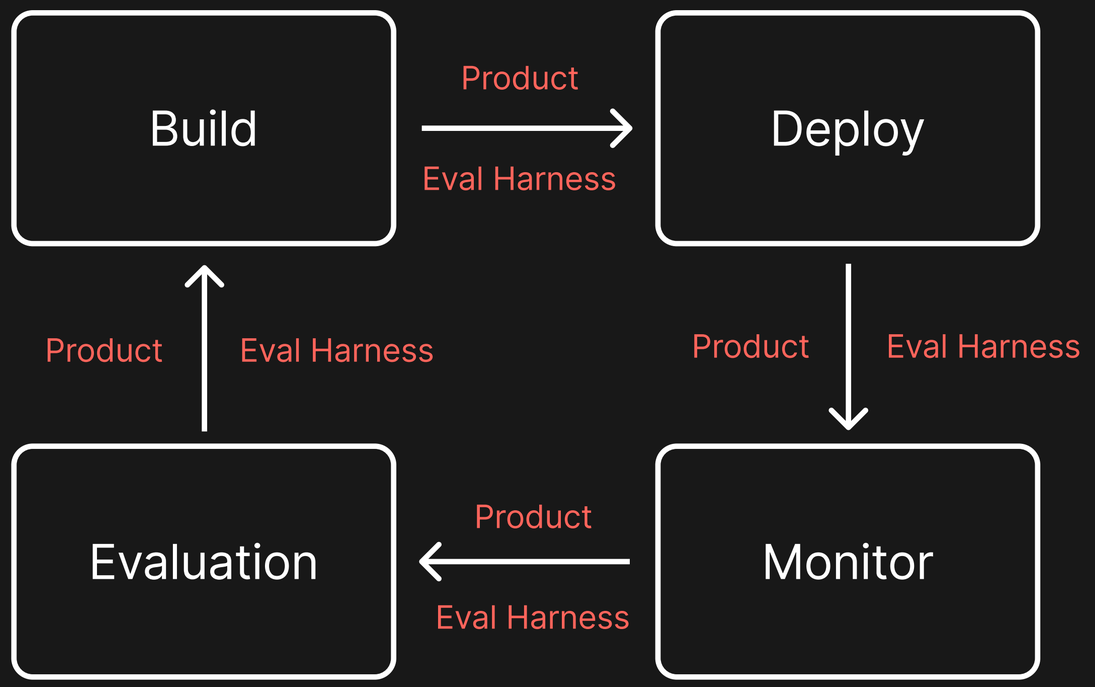
AI Features are experiments

Scientific Approach

Treat any AI feature as an experiment:

- Observe
- Question and form hypothesis
- Conduct experiments
- Analyze data to draw conclusions

Evaluation Driven Development



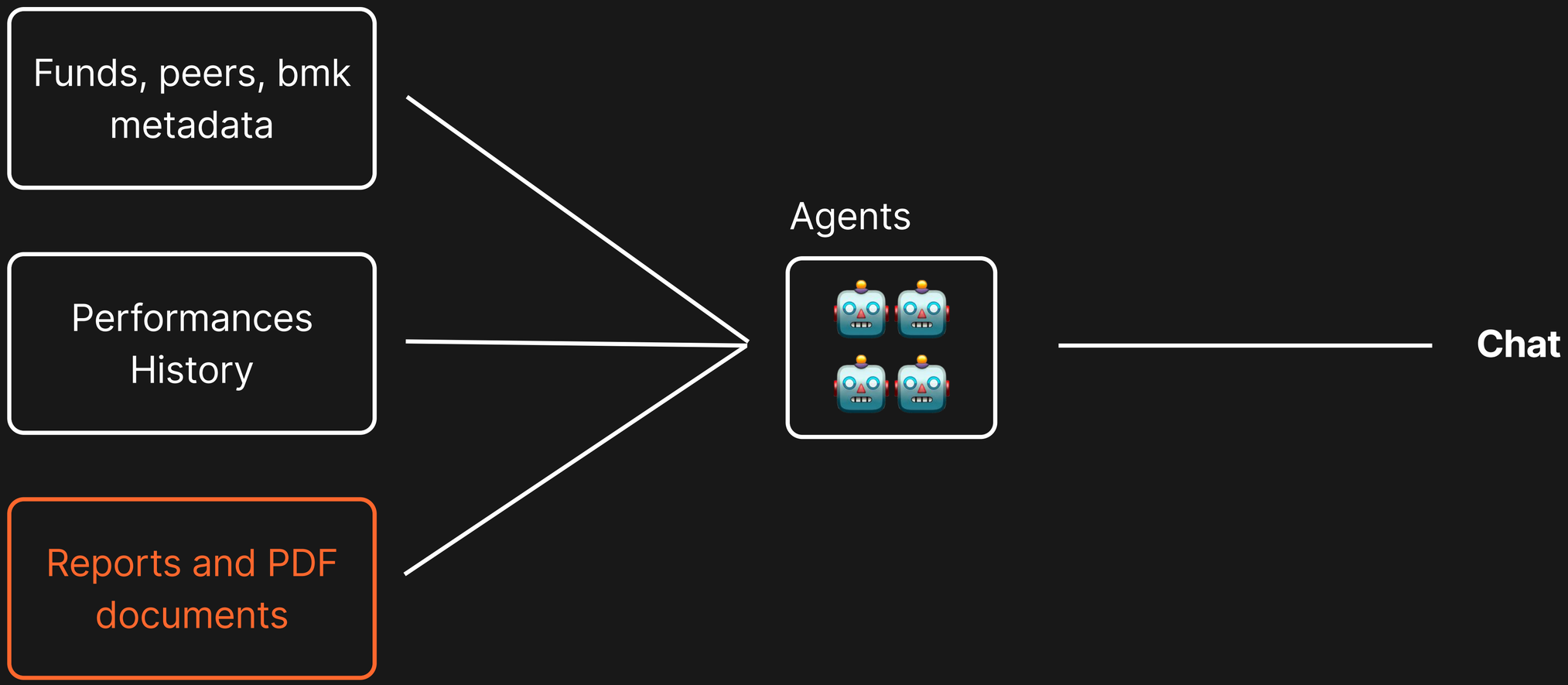
Deploy (increasing feedback loop range):

1. Internal team
2. Stakeholder
3. SMEs
4. End users

Feedback

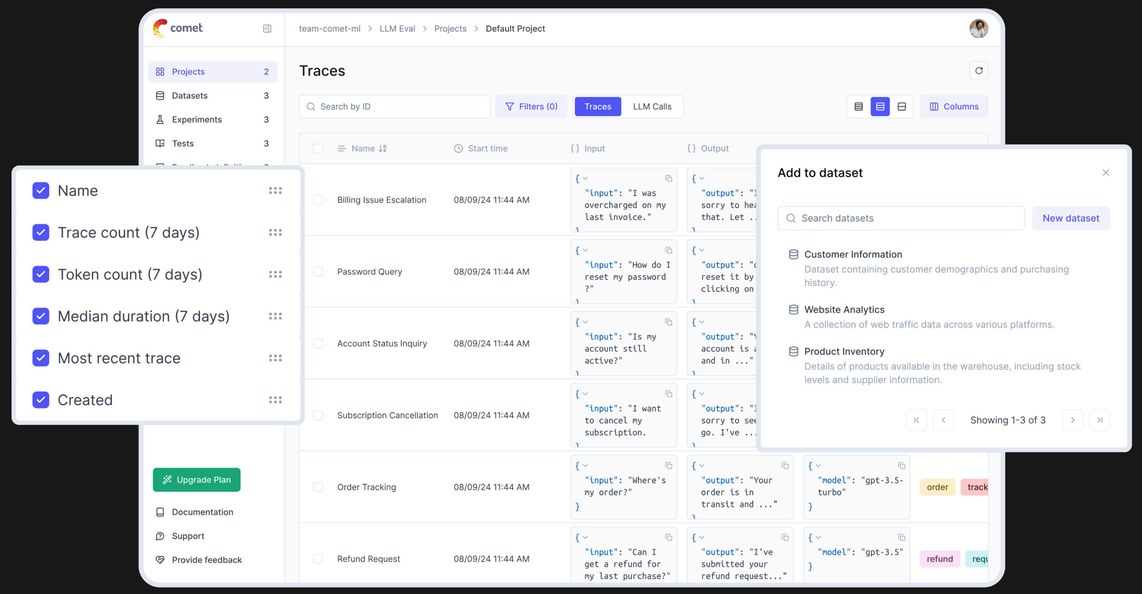

The Use Case
GRATIA ARTI
TRADE
MARK
©

Chatbot


Funds, peers, bmk metadata
Performances History
Reports and PDF documents
Agents
Chat

Observability Setup

Opik



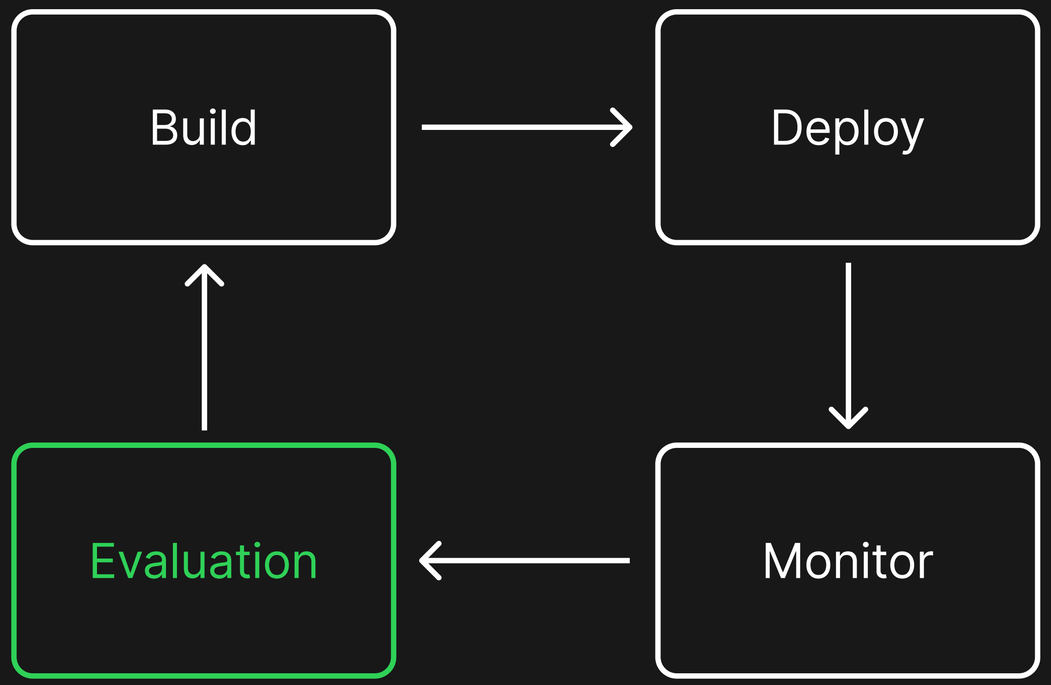
- Datasets upload
- LLM completions tracing
- General logging
- Experiments runs → Evaluation
- Dashboard GUI for human review

Ground Truth Definition

User query	Context	Response

- Features scope narrowing
- User queries listing
- Ideal responses based on current state (context)
- SMEs consultancy + synthetic data
- Versioned datasets

Evaluation Driven Development



Before any Chatbot implementation

```
class ChatBot:

    def answer(...) -> str:
        return "Hello World"
```

Takeaways

Issues

- Unstructured Data
- Non-determinism
- Iteration Cost
- Edge Cases / Halluc.

Metodology

- Evaluation Driven Development
- Scientific Approach

Process