従来のRAGは、なぜ間違えるのか？

ベクトル検索の限界を突破する「推論ベース」RAG：PageIndexの全貌

RAG 1.0 (Vector): 類似度への依存

RAG 2.0 (PageInDex): 構造と推論への進化

エグゼクティブ・サマリー

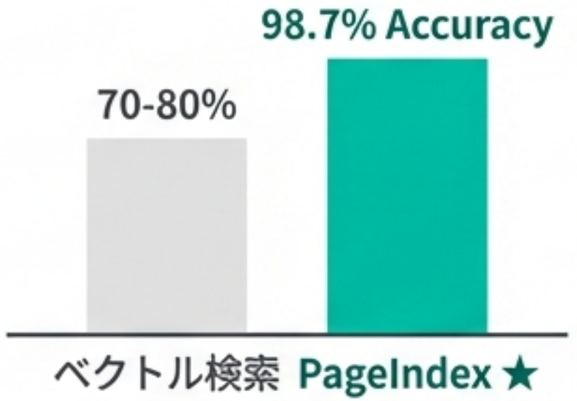
1. 課題 (The Problem)

類似性は関連性ではない
(Similarity ≠ Relevance)。
従来のベクトルRAGは単語の一致に依存し、複雑なロジックや文脈理解でハルシネーションを起こしやすい。

2. 解決策 (The Solution)

PageIndex。
ベクトルDBを使わず、ドキュメントの「目次構造」を構築。LLMが人間のように目次を推論して情報を探索する新フレームワーク。

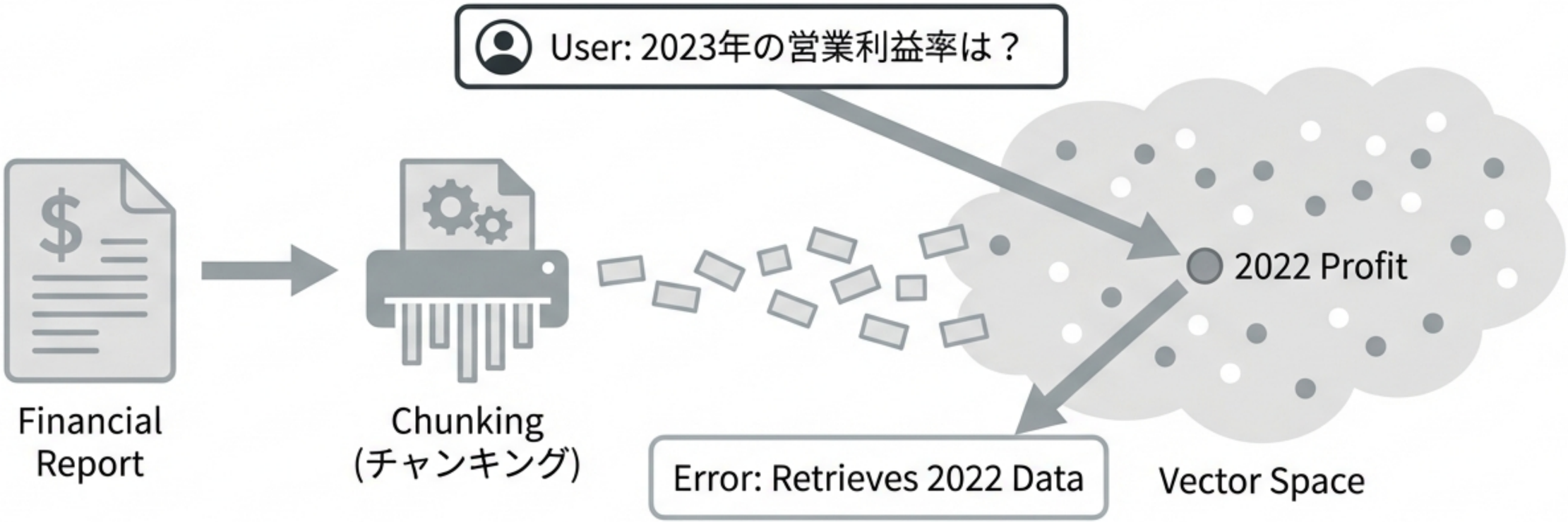
3. 実績 (The Result)



SOTA Accuracy。
金融ベンチマーク「FinanceBench」で**98.7%**の精度を達成。従来のベクトル検索（70-80%）を圧倒。

ベクトルDB不要。チャンキング不要。あるのは推論のみ。

ベクトル検索の隠れた欠陥：「雰囲気検索」の正体



Similarity ≠ Relevance

「類似度」は単語の近さしか見ない。「関連性」は文脈（年度・指標）の理解が必要。

検索のパラダイムシフト：「単語合わせ」から「推論」へ

Old AI (Vector)

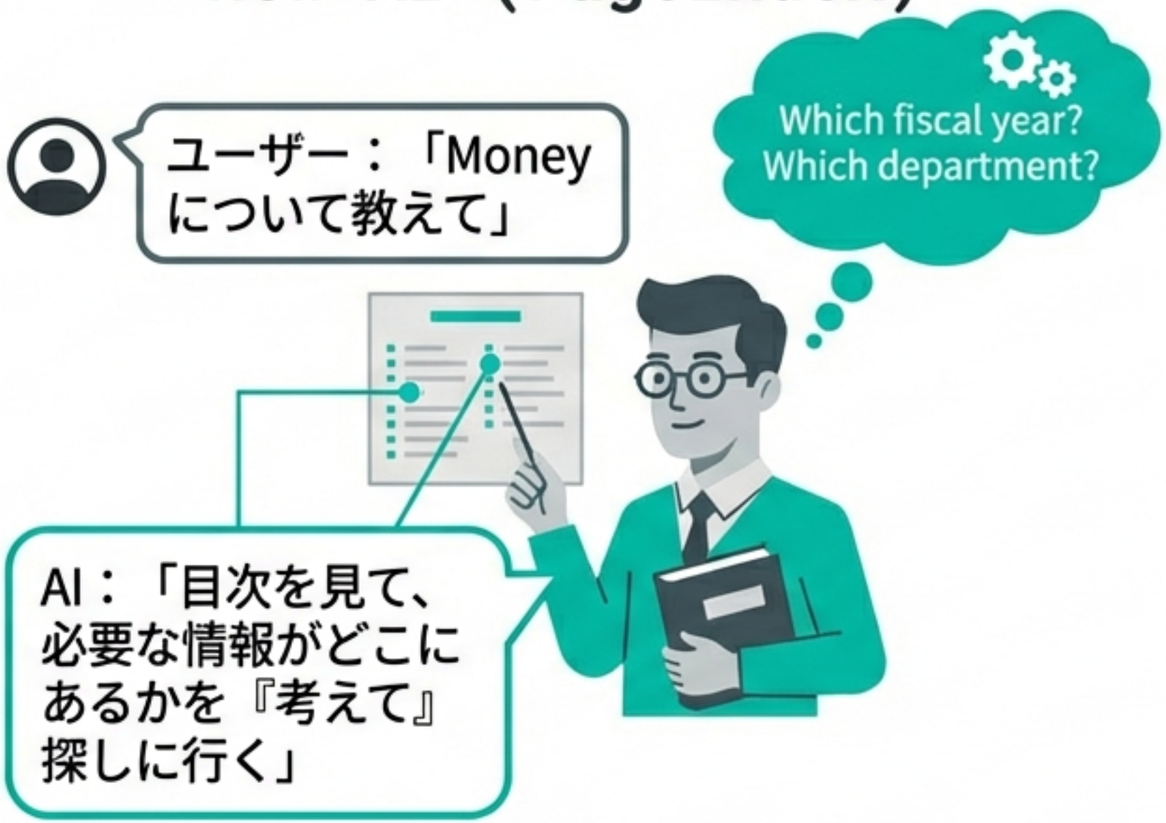
ユーザー：「Moneyについて教えて」

AI：「単語が含まれる本を片っ端から持ってくる（文脈無視）」

Similarity ≠ Relevance

New AI (PageIndex)

ユーザー：「Moneyについて教えて」



PageIndexは、人間の専門家の検索プロセスをシミュレートする。

PageIndex：RAG 2.0の定義

PageIndex Framework

No Vector DB

Noto Sans JP

不要 ✕

ドキュメント構造を利用するため、複雑なベクトルDBは不要。

No Chunking

不要 ✕

意味のない断片化をせず、章・節という自然な単位で扱う。

Reasoning-based

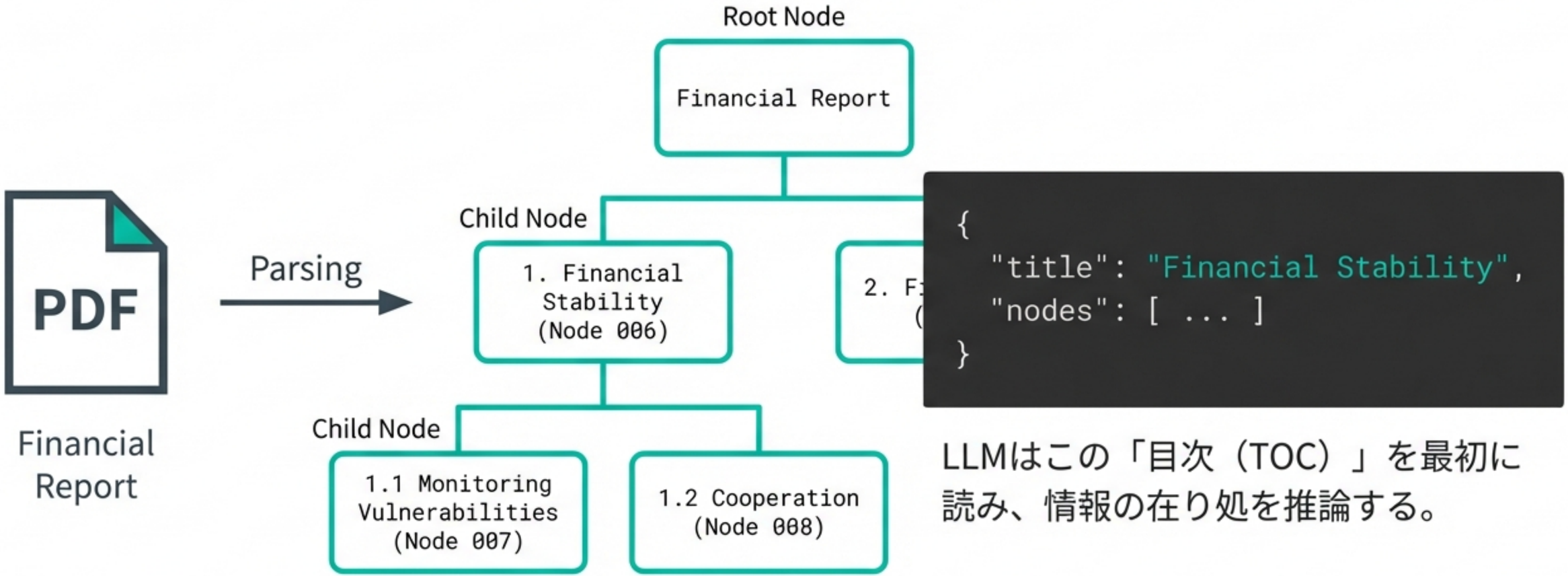
NotoSans JP

推論ベース ✓

「雰囲気」検索を撤廃し、論理的で追跡可能なナビゲーションへ。

「ドキュメントを切り刻むな。構造化された知識として扱え。」

技術的革新：セマンティック・ツリー構造



LLMはこの「目次（TOC）」を最初に読み、情報の在り処を推論する。

「推論」による検索プロセス

Index Generation (ツリー生成)

PDFから目次構造を抽出。

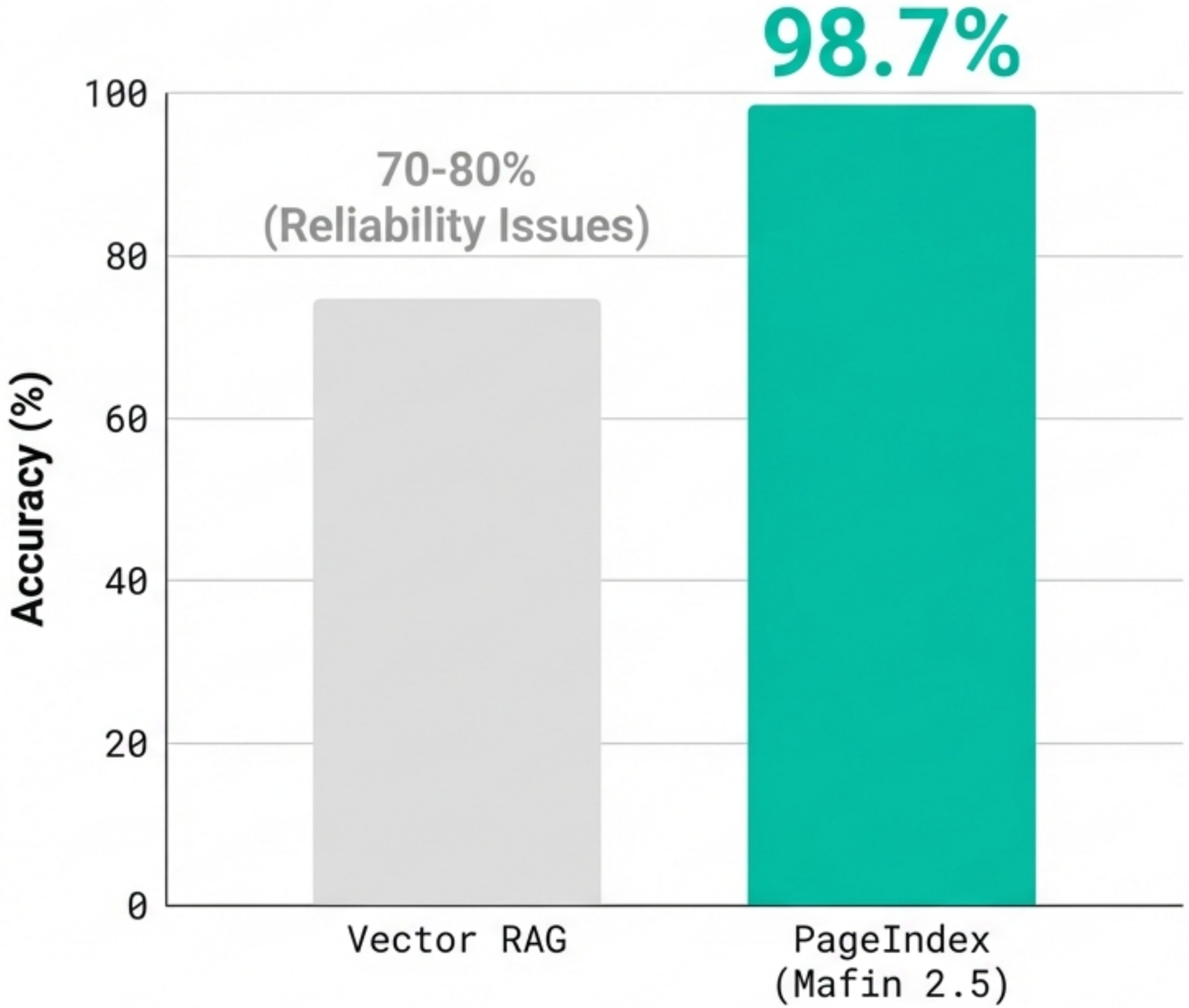
Reasoning Search (推論検索)

LLMが探索パスを決定。

Extraction (抽出)

該当ページへ直接ナビゲートし回答。

圧倒的な精度：FinanceBenchでの実証結果

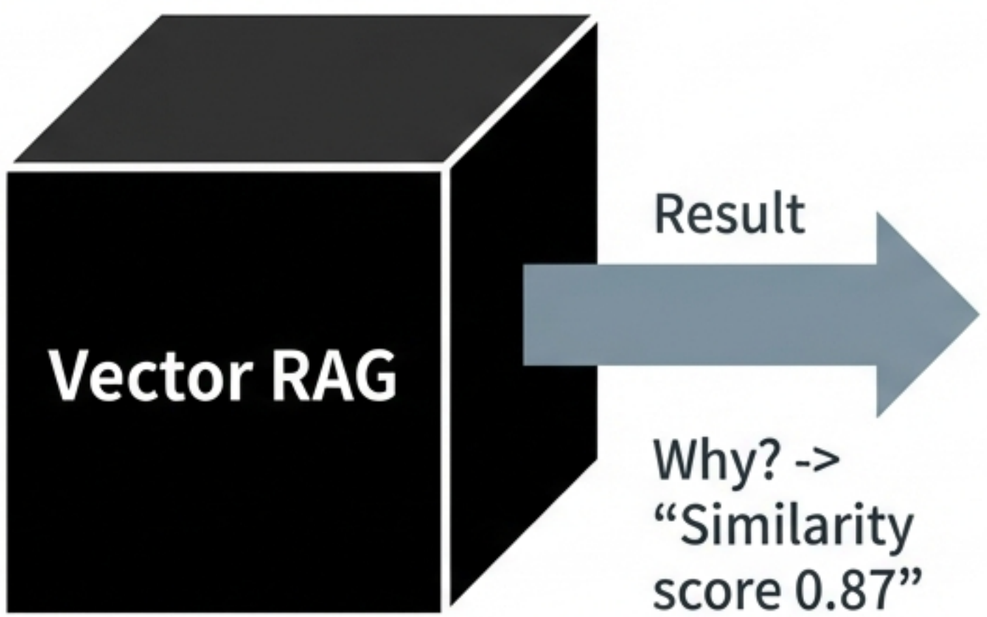


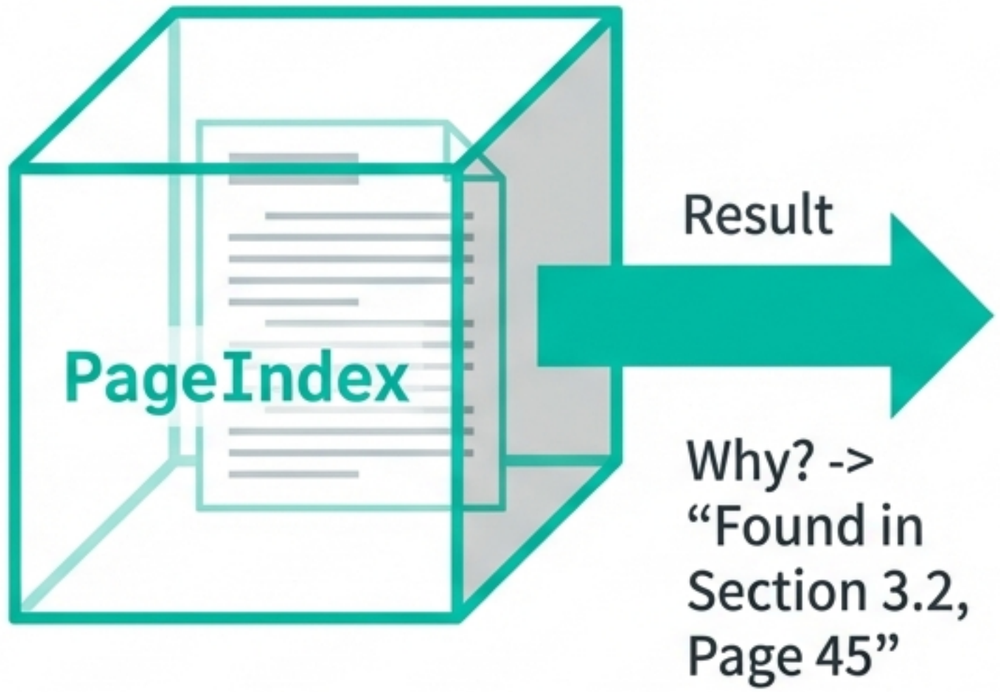
FinanceBenchは、'2022 Q3' と '2023 Q3' の区別など、高度な推論を要するベンチマーク。

80%の精度 = 5回に1回間違える（実務不可）。

98.7%の精度 = エンタープライズ利用に耐えうる信頼性。

ブラックボックスからの脱却：説明可能性





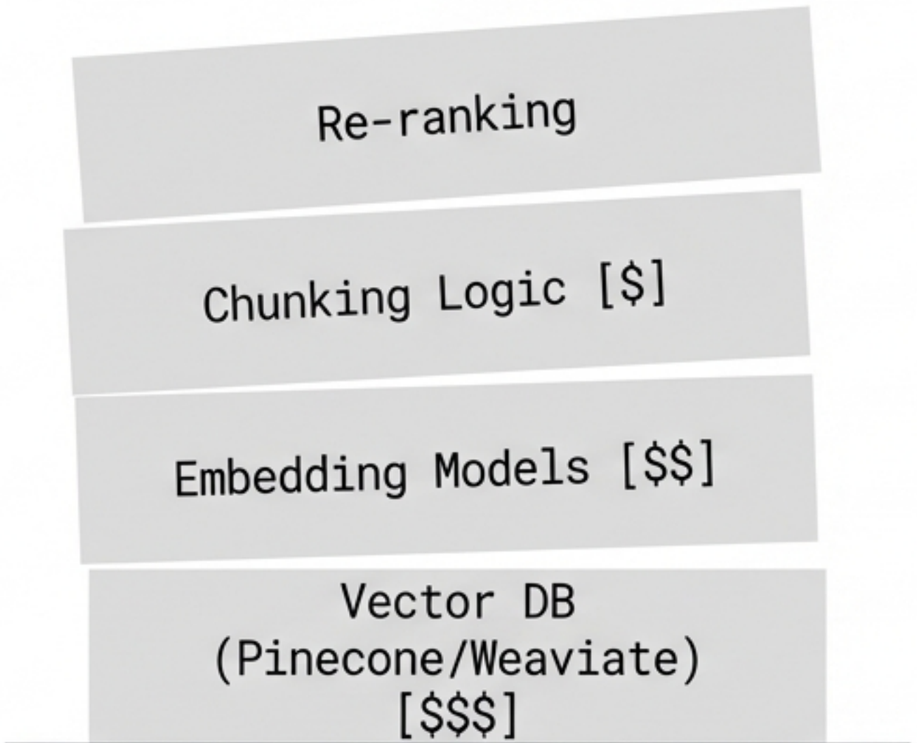
根拠が数学的すぎて検証不能（Black Box）。

誰でも検証可能。監査・法務に必須（White Box）。

インフラの簡素化とコスト革命

Stack

Legacy Stack



複雑で高コスト

PageIndex Stack

シンプルで低コスト

$0 Vector DB Cost.
$0 Embeddings Cost.

PageIndexが真価を発揮する領域

Finance

Annual Reports, SEC Filings.
特定の数値や指標の抽出。

Legal

Contracts, Regulations.
特定の条項や例外の検索。

Technical

Manuals, API Specs.
構造化された技術知識。

Not Recommended For: Random tweets, unstructured logs, simple keyword search.

「人間が目次を使って読むドキュメント」なら、PageIndexが最適解。

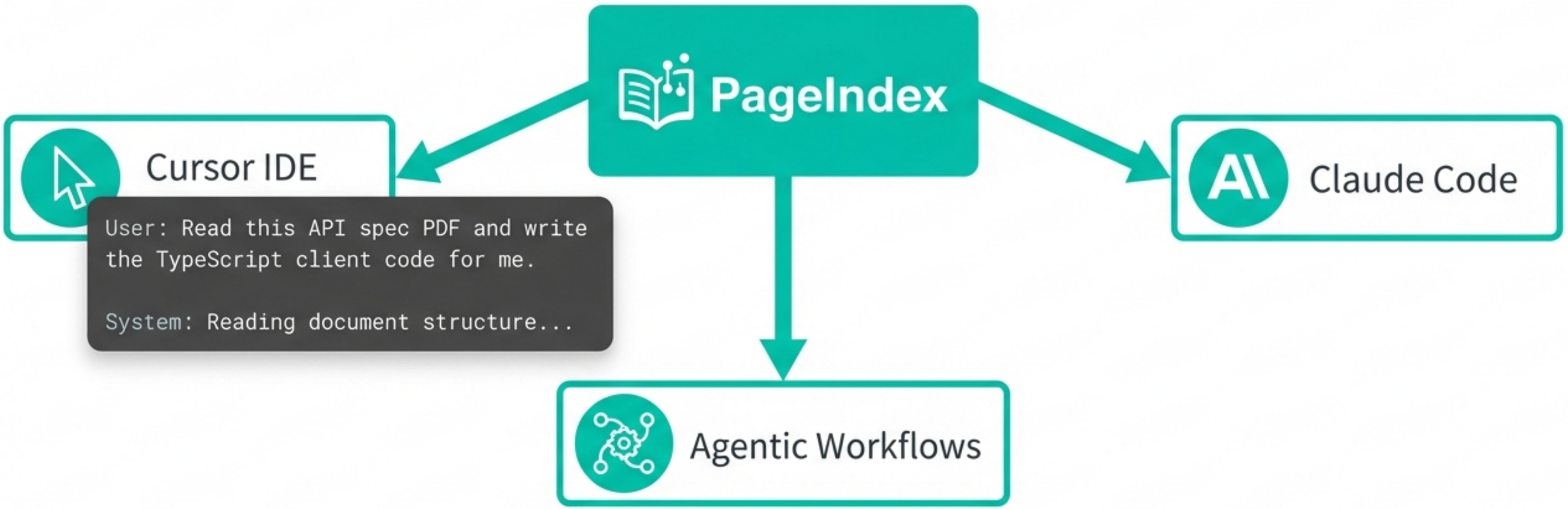
実装は驚くほどシンプル

```
# 1. Install
pip install -r requirements.txt

# 2. Run PageIndex on your PDF
python3 run_pageindex.py --pdf_path document.pdf
```

- Markdown Support: マークダウンファイルも処理可能
- **Vision-based**: OCR不要で画像ベースの処理も対応

最新のエコシステム連携：MCPとAgent



Model Context Protocol (MCP) 対応により、IDEから直接ドキュメント構造を呼び出し、コーディングを自動化。

比較まとめ：従来型 vs PageIndex

Feature	Vector RAG	PageIndex
Search Method	Similarity (Matching)	**Reasoning (Thinking)**
Data Structure	Chunks (Fragmented)	**Tree (Structured)**
Vector DB	Required ($$$)	**Not Required ($0)**
Explainability	Low (Black Box)	**High (Traceable)**
Accuracy	~70-80%	**98.7%**

推測をやめ、推論をはじめよう

Stop Guessing, Start Reasoning

GitHub (Open Source)

Web Demo

Cookbooks

Created by Vectify AI

pageindex.ai