Dataline Protocol Whitepaper

Autonomous AI agents are only as good as the data behind them. Today's data infrastructure was built for humans, not AI. Data is fragmented across dozens of providers, inconsistent in schema, and silent on quality. There is no standard way for an agent to know whether the data it received is fresh, reliable, or safe to act on.

Dataline is the decentralized data network for AI agents. It provides a single integration for all markets: cross-verified feeds with AI-generated confidence scores, and an open job network where anyone can request or fulfill custom data. Quality is enforced economically via the $DAT network and data access token, not by any central authority.

One integration. Every market. Confidence on every call.

---

1. The Problem

1.1 Fragmented Data Stacks

Crypto and AI data are a series of islands. Each protocol, exchange, and prediction market exposes its own interface with its own schema, authentication method, and failure mode. A production-grade agent consuming price data, funding rates, and event odds today must stitch together:

- Separate APIs for token prices, perpetuals, and prediction market signals
- Custom normalization logic for each source
- Ad hoc outlier detection with no shared standard
- No way to express "how confident are you in this value?"

Before executing a single step, an agent is already operating on top of a heavily engineered coordination layer it shouldn't have had to build.

1.2 No Quality Primitive

Existing data APIs return a value. They do not tell you how much to trust it. For human analysts, stale or outlier data is a nuisance. For autonomous agents executing trades, DeFi strategies, or on-chain transactions at speed, it is a critical failure mode. A mis-priced fill or a stale signal can cascade into real losses with no native mechanism to detect or prevent it.

The market has no standardized quality primitive for AI-consumed data. Dataline introduces one.

1.3 No Open Data Market

When an agent needs data that doesn't yet exist as a structured feed (for instance, a custom sentiment index, a niche on-chain metric, a proprietary signal) there is no market to request it. The agent either goes without or the developer builds a one-off integration. There is no open, incentive-aligned network for requesting, fulfilling, and verifying custom data on demand.

---

2. Market Opportunity

2.1 Agents Are the New User

AI agents are rapidly becoming the primary consumers of market and on-chain data. Unlike human dashboards, agents:

- Query continuously, not episodically
- Require machine-readable quality signals, not charts
- Operate across multiple markets simultaneously
- Can pay autonomously via machine-payment rails

The infrastructure built for human data consumers is not designed for this use case. Agents need a data layer built for agents.

2.2 The Routing Layer for AI Data

Just as a routing layer for LLMs abstracts model selection, fallback, and pricing into a single API, Dataline is the routing layer for AI data. Via a single endpoint, every response carries a confidence score so agents can act on data with the same reliability guarantees they expect from any other system primitive.

2.3 The Gap in Current Infrastructure

Need	Current Approach	Problem
Price feeds	Multiple exchange APIs	Inconsistent schemas, no quality signal
Perpetuals data	Exchange-specific SDKs	Siloed, no normalization
Prediction markets	Direct provider access	No aggregation across venues
Normalized output	Custom ETL per team	High maintenance, no confidence scoring
Custom data	One-off integrations	No open market to request or fulfill

3. The Dataline Protocol

Dataline operates as a two-sided decentralized network:

- Supply side: Data providers publish feeds and fulfill custom data requests
- Demand side: AI agents and developers consume verified, normalized data on demand

Every response carries a confidence score. Every transaction settles instantly. Quality is enforced by economic stake, not by a central gatekeeper.

3.1 Curated Data Feeds

Dataline aggregates 20+ (and growing) verified data sources across crypto markets and prediction markets into a unified schema. Every query runs through a 5-stage pipeline:

```
Intent Parse → Route → Normalize → Aggregate → Output
```

Intent Parse: Natural language or structured intent is parsed into a typed query. Agents can express requests in plain language or via typed API parameters.

Route: The query is routed to the relevant subset of Dataline's integrated sources based on asset class and data type.

Normalize: Each source response is mapped to Dataline's unified schema. Field names, decimal precision, and timestamp formats are standardized across all sources.

Aggregate: Normalized values are aggregated across sources. Outliers are detected and filtered. A confidence score is computed from source agreement, recency, and coverage.

Output: A single structured response is returned (value, metadata, and confidence score) in under 87ms at $0.0001 USDC per query.

3.2 AI Confidence Scores

Every Dataline response carries a confidence score. This is a first-class primitive, designed to plug directly into any app consuming Dataline feeds. An AI trading app, for instance, could restrict trades to price data scoring above 80 in confidence and discard anything lower as unreliable. It is computed during aggregation as a function of:

- Consistency: how closely independent sources agree on the value
- Freshness: how recent each contributing data point is
- Coverage: how many sources contributed to the aggregate

Agents can gate execution on a confidence threshold and refuse to act when the score falls below a defined level. This makes agent behavior auditable and risk-bounded by design.

3.3 Available Data

Market Data

Endpoint	Returns
`getPrice`	Spot or swap price for any base/quote pair
`getFundingRate`	Annualized funding rate for a perpetual contract

Prediction Markets

Endpoint	Returns
`getOddsCategories`	List of available event categories
`getOddsEventList`	Paginated list of prediction market events
`searchPredictionEvents`	Full-text search across events
`getPredictionEventDetail`	Full detail on a single event

`getOddsEventOrderbook`	Bids and asks for a specific outcome
`getPredictionMarketQuote`	Best bid, best ask, and mid-price

Data partners in the launch cohort include Hyperliquid, Polymarket, Kalshi, and others.

3.4 Developer Integration

Dataline is available via REST API and MCP server, compatible with Claude, Cursor, and Codex. Authentication uses an API key and secret key pair.

```
import { DatalineClient } from "@dataline-xyz/dataline-mcp";

const client = new DatalineClient({
  apiKey: process.env.DATALINE_API_KEY,
  secretKey: process.env.DATALINE_SECRET_KEY,
});

const price = await client.getPrice({
  base_currency: "BTC",
  quote_currency: "USDT",
});

// Returns: price, metadata, confidence score
```

Payments settle via x402, an open standard for machine-to-machine HTTP payments that enables agents to pay per API call autonomously, without human-managed subscriptions or billing cycles.

---

4. The Open Data Network

Beyond curated feeds, Dataline operates an open job network for custom data through a permissionless marketplace where anyone can request data and anyone can fulfill it.

4.1 How It Works

1. A requester posts a bounty specifying the data they need and the $DAT reward
2. A fulfiller claims the task and stakes $DAT as a quality bond
3. The fulfiller submits the data, verified against the request specification
4. Outcome is determined economically:
    - Good data: stake returned + bounty earned
    - Bad data: stake slashed

Requesters can be human developers or autonomous AI agents. Fulfillers can be humans, automated pipelines, or AI agents. The network is indifferent to who participates because it enforces quality through stake, not identity.

4.2 Zero-Knowledge Attestation

An optional ZK attestation tier allows providers to attach a verifiable proof bundle to every response. Consumers can verify data provenance cryptographically without exposing the provider's underlying sources or credentials. Sellers' credentials never leave the gateway. This is particularly useful for more sensitive data sets, where the verification of authenticity may be important, while keeping parts of the underlying data set private.

4.3 Economic Quality Enforcement

Quality in the Dataline network is enforced by economic stake, not moderation. A fulfiller who submits bad data loses their stake. The lost stake is added to the original data requester’s bounty as an additional bonus on top of the original bounty of the request - this provides an additional incentive for data fulfillers to provide replacement data. A fulfiller who consistently delivers good data earns compounding rewards. This creates durable alignment between data quality and fulfiller incentives at scale, without central oversight.

---

5. $DAT Token

$DAT is the network and data access token of the Dataline protocol. It serves three functions: access, alignment, and work.

5.1 Access

Model	Description
Pay-per-call	Per-query pricing, settled via x402
Subscription	Flat monthly access with a credit allocation
Stake	Stake $DAT to unlock access without recurring payment

Staking replaces the monthly subscription with a composable, agent-native access primitive. Rather than paying a fee that is consumed and forgotten, users stake $DAT and gain access proportional to their stake. This is the next-generation subscription model for the AI era, giving users skin in the network's success.

5.2 Alignment

Stakers are aligned participants, not just users. When the network grows and data quality improves, their stake increases in value. This creates a direct feedback loop: users who care about data quality have an incentive to hold and stake, which funds the protocol, which improves the data, which attracts more users.

5.3 Work

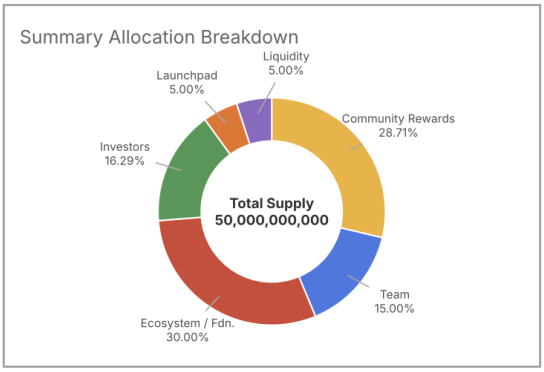
$DAT powers the open data job network end to end:

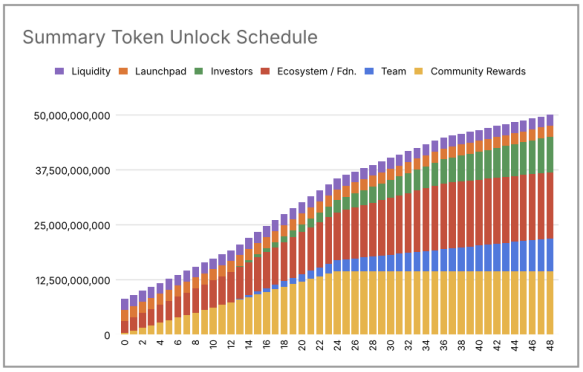
- Requesters post bounties in $DAT for custom data
- Fulfillers stake $DAT to claim tasks
- Slashing penalizes bad submissions
- Rewards are earned in $DAT for verified, high-quality data

The token is the economic backbone of the entire data supply chain from request to fulfillment to verification.

5.4 Tokenomics

Token Model Total Supply: 50,000,000,000





Summary Allocation	(%) FD	# Tokens	*Day 1 Unlock (%)*	Tokens Unlocked Day 1 (#)
Community Rewards	28.71%	14,355,000,000	*0.50%*	250,000,000
Team	15.00%	7,500,000,000	*0.00%*	0
Ecosystem / Fdn.	30.00%	15,000,000,000	*5.50%*	2,750,000,000
Investors	16.29%	8,145,000,000	*0.00%*	0
Launchpad	5.00%	2,500,000,000	*5.00%*	2,500,000,000
Liquidity	5.00%	2,500,000,000	*5.00%*	2,500,000,000
Total	**100.00%**	**50,000,000,000**	***16.00%***	**8,000,000,000**

Token Release Schedule	Lock Up Period (Months)	Unlock Period (Months)	Total Period (Months)
Community Rewards	0	24	24
Team	12	36	48
Ecosystem / Fdn.	0	36	36
Investors	12	36	48
Launchpad	0	0	0
Liquidity	0	0	0

Note: Lock Ups & Unlock Periods denote averages

Token Allocation
The total supply of 50,000,000,000 tokens is distributed across six summary categories designed to balance long-term ecosystem growth, fair community participation, and sustainable incentives for core contributors and investors.

Community Rewards: 28.71% (14,355,000,000 tokens)
The largest allocation is reserved for the community, funding incentive programs, staking rewards, airdrops, and user growth initiatives that drive long-term network participation and decentralization. This category receives the largest share because Dataline's value depends on network effects, more stakers and query volume improve confidence-score reliability across the aggregation pipeline, which in turn attracts more data providers, so the allocation weights toward sustaining that usage loop over time rather than front-loading value to insiders.

Ecosystem / Foundation: 30% (15,000,000,000 tokens)
Allocated to support ongoing protocol development, grants, partnerships, and strategic initiatives that expand the ecosystem's utility and adoption over time. This allocation is sized to fund multi-year development without repeated fundraising, reflecting that integrating and maintaining data sources, onboarding new chains, and building out FlowAgent, GhostDriver, and ChatPilot are ongoing engineering commitments rather than one-time costs.

Launchpad: 5% (2,500,000,000 tokens)
Dedicated to launchpad distribution at launch. Unused Launchpad tokens will be funneled back into the Ecosystem/Foundation bucket.

Liquidity: 5% (2,500,000,000 tokens)
Dedicated to initial exchange liquidity, ensuring a healthy and stable trading environment from day one.

Team: 15% (7,500,000,000 tokens)
Reserved for founders, core contributors, and future hires, subject to a lock-up period to align incentives with the project's long-term success.

Investors: 16.29% (8,145,000,000 tokens)
Allocated to early backers and strategic partners who provided the capital and support necessary to bring the project to market, subject to a lock-up period to align incentives with the project's long-term success.

Initial Projections
At TGE, 16.00% of the 50B token supply (8B tokens) will be unlocked on day one. On a cumulative basis, 100% of the token supply will be unlocked by month 48.

---

6. Products

Dataline's portfolio is a financial AI product suite spanning data, orchestration, execution, and user-facing interaction. The products share a common data foundation, confidence-scoring layer, non-custodial execution boundary, and agent orchestration model.

At a product level, the portfolio is composed of four complementary systems:

- **Dataline Data:** the real-time Web3 data layer.
- **FlowAgent:** the multi-step planning and orchestration layer.
- **GhostDriver:** the browser-native execution layer for Web2/Web3 interfaces.
- **ChatPilot:** the live conversational entry point for users.

Dataline Data

Product role: Dataline Data is the shared data foundation for Dataline's financial AI products and partner-facing integrations. It turns natural-language or structured intent into routed, normalized, multi-source outputs with confidence scores.

Core capabilities:

- Chain-agnostic market and asset data across supported ecosystems including BSC, Solana, Ethereum, Sui, and TON.
- Real-time prices, historical charts, liquidity data, DEX quotes, wallet balances, token metadata, portfolio context, and order-related state.
- Multi-source reconciliation across DEX aggregators, blockchain RPCs, and third-party APIs.
- Normalized responses with freshness metadata, source attribution, and confidence scoring.
- Low-latency data access, with current benchmark targets including median response time under 87ms.

Primary users and integrations: ChatPilot uses Dataline Data for real-time conversational answers and swap preparation. FlowAgent uses it for planning, decisioning, and risk evaluation. Trading agents, bots, and partner applications can consume it directly through API or MCP-style integrations.

Strategic value: Dataline Data gives the company a reusable data layer that can support multiple products and partner channels. It also reduces dependency on any single data source by combining chain abstraction, normalization, caching, and confidence scoring into one infrastructure product.

FlowAgent

Product role: FlowAgent is the orchestration layer for multi-step DeFi workflows. It decomposes user goals into graph-based workflows, evaluates confidence at each step, calls the right data and tool services, and generates task-specific UI when needed.

Core capabilities:

- LangGraph-based orchestration for Router, Plan, Decision, Answer, and UI nodes.
- Confidence-driven interrupts when intent, chain, token, amount, risk tolerance, or data quality is ambiguous.
- MCP service access for market data, DeFi protocol data, information aggregation, transaction execution, and GhostDriver-powered web automation.
- Dynamic UI generation for trading panels, staking dashboards, charts, and portfolio views.
- Support for complex DeFi strategies such as LP pledge arbitrage, revolving loans, and cross-DEX arbitrage.

Primary users and integrations: FlowAgent serves advanced users, trading agents, and product surfaces that require multi-step reasoning rather than a single answer. It can operate behind ChatPilot, power dashboard workflows, and coordinate execution paths across chains and protocols.

Strategic value: FlowAgent is the control plane of the product suite. It makes Dataline more than a data API or chat interface by giving the system a reusable way to plan, branch, interrupt, and execute complex financial workflows with explicit confidence management.

GhostDriver (Beta)

Product role: GhostDriver is the browser-native execution layer. It turns intent and operation sequences into real browser actions across Web2 applications, DeFi front ends, dashboards, wallets, and dApps.

Core capabilities:

- Chrome-based automation that works in real user browser contexts, including authenticated sessions and interactive pages.
- AI-assisted page recognition for DOM structure, visual elements, functional regions, and workflow paths.
- Structured site knowledge through sitemap indexing and MCP access.
- Partner-extensible business planning, allowing external B2B teams to bring their own workflow policies or vertical logic.
- User-controlled execution boundary for wallet signatures, irreversible transactions, and asset movement.

Use cases: Airdrop task automation, DeFi rebalancing, wallet dust cleanup, cross-platform operational workflows, and workflows where a protocol or platform does not expose a reliable API.

Traction signal: GhostDriver has processed 400,000+ requests from 10,000+ users, providing an early usage signal for browser-native Web3 automation. The supporting measurement window and source dashboard can be presented during diligence.

Strategic value: Many crypto workflows still happen across fragmented websites rather than clean APIs. GhostDriver gives Dataline an execution layer for that reality, extending the product suite from data and planning into real-world browser action.

ChatPilot (Live)

Product role: ChatPilot is Dataline's live user-facing conversational product on Telegram. It gives users a natural-language interface for market questions, swaps, dashboards, and portfolio-oriented workflows.

Core modes:

- **Ask:** source-grounded Q&A for Web3, market, token, and project questions.
- **Swap:** route discovery and transaction preparation across DEX liquidity sources.
- **Dashboard:** live pinned charts, portfolio views, and market monitoring surfaces.

Core capabilities:

- Deterministic intent routing from user input to specialized executors.
- Real-time data grounding through Dataline Data.
- Context-sensitive memory for preferences, chains, recurring workflows, and multi-turn interactions.
- Confidence-scored responses that expose source quality, freshness, cross-source agreement, and completeness.
- Non-custodial signing boundaries for swap and transaction-related workflows.

Primary users and integrations: ChatPilot is the clearest consumer-facing wedge for Dataline. It is useful for retail DeFi users, communities, and partners that want a conversational interface for trading, research, and monitoring without requiring users to manually navigate multiple DeFi tools.

Strategic value: ChatPilot packages Dataline's data, orchestration, and execution capabilities into a simple user interface. It can drive adoption while the underlying infrastructure products support agents, partners, and more advanced workflows.

---

7. Technical Design

7.1. Dataline Data: Web3 Data Layer for Financial AI

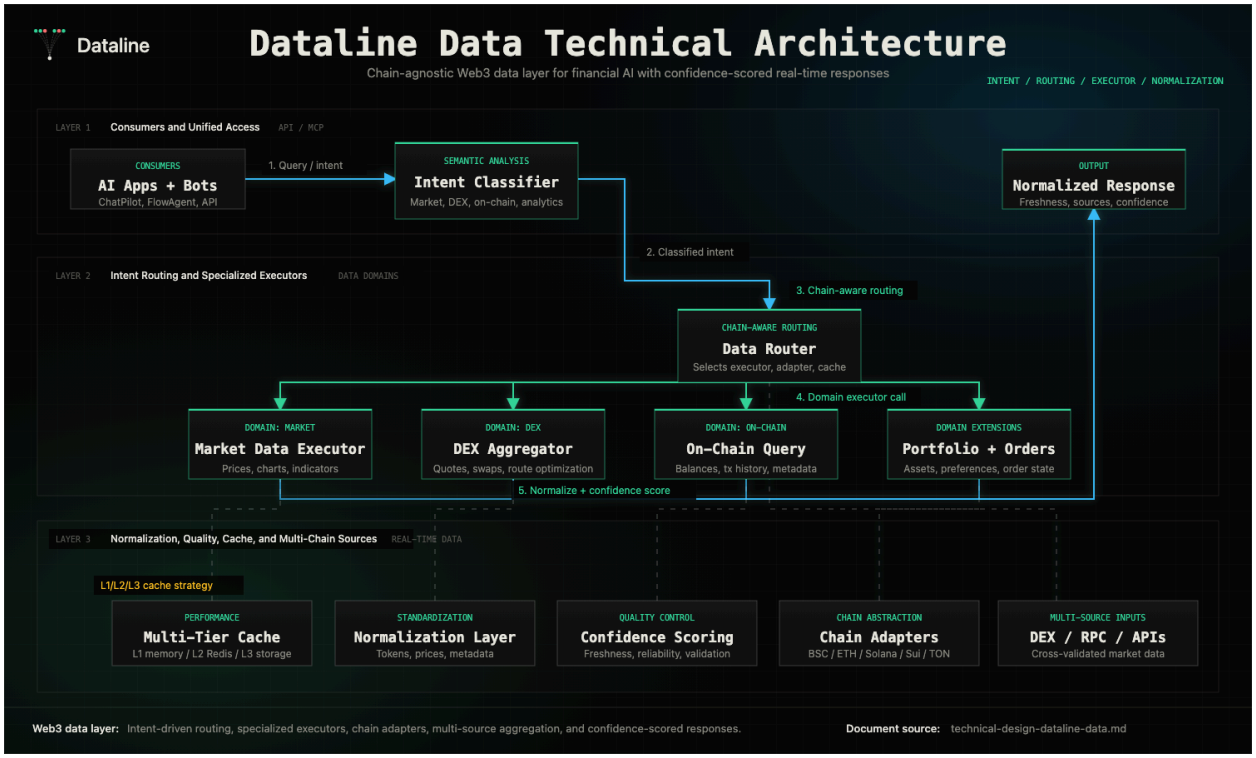


Figure: Dataline Data Technical Architecture

Dataline Data is the Web3 data layer that powers Dataline's financial AI products, trading workflows, and partner-facing integrations. It transforms natural-language requests and structured intents into routed, normalized, multi-source data responses with confidence scores across supported markets and chains.

The system is designed to solve a practical problem for financial AI: agents cannot make reliable decisions from stale, chain-specific, or single-source data. Dataline Data provides a chain-agnostic access layer for real-time market data, DEX routes, on-chain state, portfolio information, and order-related context.

Core Design Principles

1. Chain-Agnostic Abstraction: Consumers interact with one consistent interface while the platform handles BSC, Solana, Ethereum, Sui, TON, and future chain-specific implementation details.

2. Real-Time Data Freshness: Every important data read carries timestamp and freshness metadata. Cache invalidation and live feeds are used to keep time-sensitive responses current.

3. Multi-Source Aggregation: Market and on-chain data can be reconciled across DEX aggregators, blockchain RPCs, and third-party APIs before a normalized response is returned.

4. Intent-Driven Routing: Natural-language queries and structured API requests are classified and routed to specialized executors optimized for market data, DEX aggregation, on-chain queries, portfolio analysis, and order workflows.

5. Confidence-Scored Output: Responses include quality indicators so downstream AI systems can decide whether to act, ask for confirmation, or request additional verification.

Core Architecture

Figure 3 summarizes the unified data pipeline from intent classification through routing, specialized executors, normalization, confidence scoring, and the final normalized response. The component descriptions below focus on each module's responsibility.

Key Components

Intent Classifier: Determines whether a request requires market data, DEX quotes, on-chain state, portfolio context, historical charts, or order-related information.

Data Router: Selects the appropriate executor, chain adapter, cache tier, and data source strategy. Routing decisions incorporate chain context, data freshness, rate limits, and source reliability.

Market Data Executor: Retrieves prices, volumes, technical indicators, historical charts, and market analytics across supported chains.

DEX Aggregator Executor: Handles quotes, liquidity discovery, route optimization, approval preparation, and unsigned transaction generation for swap-related workflows.

On-Chain Query Executor: Retrieves wallet balances, transaction history, token metadata, smart-contract state, and multi-chain portfolio information.

Portfolio and Order Executors: Manage user asset views, order state, limit order lifecycle data, and execution triggers where supported.

Normalization Layer: Converts chain-specific or source-specific responses into a consistent schema with standardized tokens, prices, timestamps, metadata, and source attribution.

Confidence Scoring: Scores each response based on freshness, source reliability, cross-source agreement, and completeness.

Multi-Chain Abstraction Layer

Dataline Data treats each supported chain as a pluggable adapter. Adding a chain requires implementing the adapter interface rather than changing consumer-facing APIs.

Chain Adapter Pattern: Each adapter encapsulates address formats, RPC behavior, token standards, DEX integrations, and chain-specific error handling.

Unified Token Registry: A shared token registry maps equivalent assets across chains and standardizes metadata across ERC-20, SPL, Jetton, and other token standards.

Chain-Aware Routing: When a user or agent does not specify a chain, the router can use liquidity depth, gas cost, user preference, and historical behavior to choose a default route.

Data Domain Architecture

Market Data Domain: Real-time prices, historical charts, technical indicators, and market analytics. This domain prioritizes freshness and cross-source validation.

DEX Aggregation Domain: Liquidity discovery, quote aggregation, swap routing, and transaction preparation. This domain prioritizes route quality, slippage, and execution safety.

Portfolio and Asset Domain: Wallet balances, token discovery, user preferences, and cross-chain asset visibility.

Order Management Domain: Limit order state, execution triggers, order history, and lifecycle monitoring.

Confidence Scoring System

Each response receives a 0-100 confidence score based on:

- **Data Freshness:** Time elapsed since last update, weighted at 40%.
- **Source Reliability:** Historical quality and availability of the source, weighted at 30%.
- **Cross-Validation:** Agreement across independent sources, weighted at 20%.
- **Completeness:** Presence of required fields, weighted at 10%.

Scores above 85 indicate high-confidence data suitable for automated or semi-automated workflows. Scores between 70 and 85 are suitable for informational use. Scores below 70 trigger warnings, fallback retrieval, or manual verification.

Caching Strategy

L1 - In-Memory Cache: Sub-millisecond access for frequently queried stable data such as token lists and supported-chain metadata. Typical TTL: 5 minutes.

L2 - Redis Distributed Cache: Low-latency access for prices, liquidity, and market data. Typical TTL: 15-60 seconds depending on volatility.

L3 - Persistent Storage: Historical data, user preferences, order history, and long-lived analytical records.

Cache invalidation is event-driven for critical data such as price updates and order state changes, and time-based for stable data such as token metadata.

Integration with Financial AI Applications

ChatPilot: Uses Dataline Data for real-time market queries, swap preparation, and confidence-scored answers inside conversational workflows.

FlowAgent: Uses Dataline Data through MCP services for planning, decisioning, risk evaluation, and generated UI.

Trading Agents and Bots: Consume normalized data for market analysis, route selection, risk checks, and execution monitoring.

Automation Platforms: Use market intelligence and on-chain context during browser automation and cross-platform workflows.

Performance Characteristics

The service is designed for low-latency, high-throughput data access. The following service envelope defines the current target and benchmark profile for the data layer.

Latency Targets:

- Median API response time: <87ms
- P95 response time: <150ms
- P99 response time: <300ms

Throughput Targets:

- Sustained: 10,000 requests / second
- Peak: 50,000 requests / second
- Concurrent connections: 100,000+

Data Freshness Targets:

- Real-time prices: <5 second delay
- Liquidity data: <15 second delay
- Historical data: immediate from pre-computed storage

Reliability Targets:

- Uptime: 99.95%
- Error rate: <0.1%
- Data accuracy: >99.5% when validated against source-of-truth on-chain data

Security and Privacy

Data Protection:

- API communication encrypted with TLS 1.3.

- Sensitive identifiers such as wallet addresses and transaction hashes are redacted or hashed in logs according to retention and privacy policy.
- API keys and scoped permissions protect partner and application access.
- Rate limits reduce abuse and protect shared data infrastructure.

Non-Custodial Architecture:

- Dataline Data does not hold user funds or private keys.
- Swap workflows generate unsigned transactions for user-side signing.
- Wallet connections are read-only for balance and portfolio queries.
- Irreversible transaction execution requires explicit user approval through the relevant wallet or execution surface.

Operational Workflow

A query arrives from ChatPilot, FlowAgent, an API client, or a trading bot. The Intent Classifier determines the data domain and required chain context. The Data Router selects the executor, cache tier, chain adapter, and data sources. The executor retrieves and reconciles data, the Normalization Layer standardizes the response, and the Confidence Scoring system attaches freshness, reliability, validation, and completeness metadata. The final normalized response is returned to the consumer.

Summary

Dataline Data is the shared data foundation for Dataline's financial AI stack. Architecturally, it combines chain abstraction, specialized data executors, multi-source aggregation, normalization, caching, and confidence scoring into a reusable layer for conversational products, agents, bots, and partner integrations.

7.2. FlowAgent: Dynamic Orchestrator for Planning, Decisioning, and UI Generation

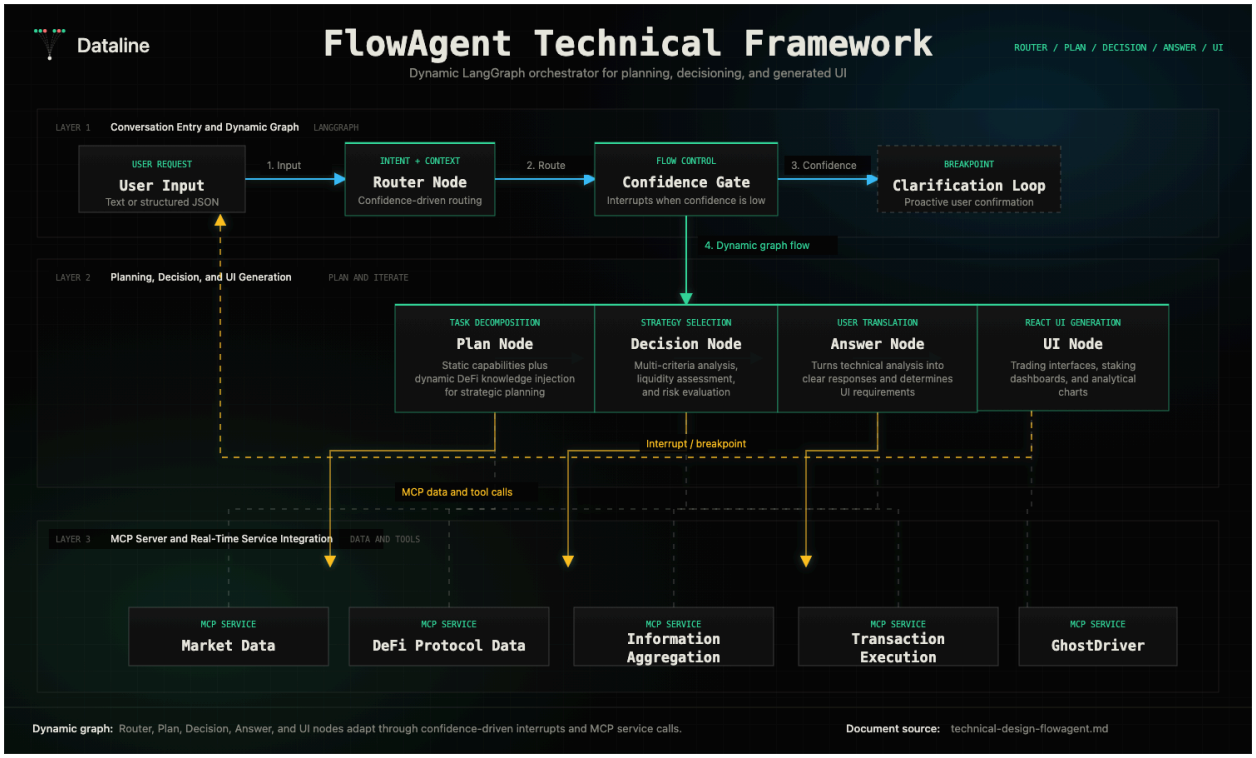


Figure: FlowAgent Technical Framework

FlowAgent is Dataline's orchestration layer for multi-step planning, decisioning, confidence management, and dynamic UI generation. It uses a LangGraph-style graph architecture to coordinate Router, Plan, Decision, Answer, and UI nodes, allowing the system to adapt its workflow based on user intent, available data, and confidence thresholds.

FlowAgent is designed for DeFi workflows where a single answer is often insufficient. Users may need intent clarification, data retrieval, plan generation, risk evaluation, transaction preparation, and a purpose-built interface. FlowAgent gives Dataline a controlled way to manage those steps without turning every conversation into an unbounded agent loop.

Core Design Principles

1. Dynamic Graph Control: FlowAgent uses graph nodes and explicit edges to control the conversation and task lifecycle. The graph can branch, interrupt, or return to the user when confidence is too low.

2. Text / JSON-Centric Interfaces: Internal node communication is based on natural-language text and structured JSON. This keeps intermediate state inspectable, testable, and easier to route into LLMs, tools, and UI generators.

3. Confidence-Driven Interrupts: The system can pause and ask clarifying questions before execution. This is especially important in financial workflows where ambiguity around chain, token, amount, wallet, or risk tolerance can create user harm.

4. Browser-Independent Orchestration: FlowAgent does not depend on a browser extension. It can coordinate APIs, MCP services, generated UI, and other execution layers such as GhostDriver when web automation is needed.

Core Architecture

Figure 5 summarizes the dynamic graph flow across routing, confidence gating, clarification, planning, decisioning, answering, UI generation, and MCP service calls. The component descriptions below focus on each node's responsibility.

Key Components

Router Node: Performs intent recognition, context selection, and initial routing. It decides whether the request can proceed or requires clarification.

Confidence Gate: Evaluates whether the current state is sufficiently complete and reliable. Low-confidence paths trigger a clarification loop rather than uncertain execution.

Plan Node: Decomposes the user request into steps. It combines baseline capabilities with dynamic DeFi knowledge injection for trading, staking, yield, portfolio, and research tasks.

Decision Node: Selects a strategy using factors such as liquidity, route quality, risk, fees, user preferences, and data confidence.

Answer Node: Converts technical analysis into a user-facing response and determines whether a UI component is needed.

UI Node: Uses a ReAct-style approach to generate task-specific interfaces, including trading panels, staking dashboards, analytical charts, and monitoring views.

Technical Features

Static + Dynamic Balance

- **Static capabilities:** Stable baseline behavior for common workflows.
- **Dynamic injection:** Context-specific domain knowledge for protocols, chains, markets, and user goals.
- **Adaptive optimization:** The graph balances reliability with specialization based on confidence and task complexity.

Confidence Management

- Node-level and workflow-level confidence scoring.
- Automatic interrupts when key fields are missing or ambiguous.
- Clarification before high-risk actions.
- Explicit state checkpoints for monitoring and debugging.

Modular Design

- Clear separation between routing, planning, decisioning, answering, and UI generation.
- Standardized node interfaces for adding new protocols and data services.
- Error handling and fallback paths at graph breakpoints.
- Compatibility with MCP services and external execution layers.

MCP Server Architecture

FlowAgent uses MCP (Model Context Protocol) services to give the graph access to real-time data, domain tools, and execution support.

Service	Function
Market Data	Prices, trading volumes, technical indicators, and historical context
DeFi Protocol Data	Liquidity pools, APY calculations, governance data, and protocol risk
Information Aggregation	News, social signals, project fundamentals, and source verification
Transaction Execution	Simulation, verification, monitoring, and execution-state tracking
GhostDriver	Browser automation for workflows that require non-API web access

Table 1: Core MCP Services

Intelligent Mechanisms

Context-Aware Service Selection: FlowAgent selects the most relevant MCP services based on intent, required data, and confidence.

Multi-Source Verification: Critical data can be cross-checked against independent sources before being used in planning or user-facing answers.

Performance Optimization: Caching, predictive fetching, and tool selection reduce latency while preserving data freshness for time-sensitive workflows.

Fault Tolerance: Circuit breakers, fallback routes, and graceful degradation help keep the experience usable when a service is slow or unavailable.

Summary

FlowAgent is the planning and orchestration layer of Dataline's financial AI stack. It provides a controlled dynamic graph for multi-step DeFi workflows, combining confidence management, MCP services, strategy selection, and generated UI into a system that is adaptable without being uncontrolled.

7.3. GhostDriver: Browser-Native Execution Engine for Web2 and Web3 Automation

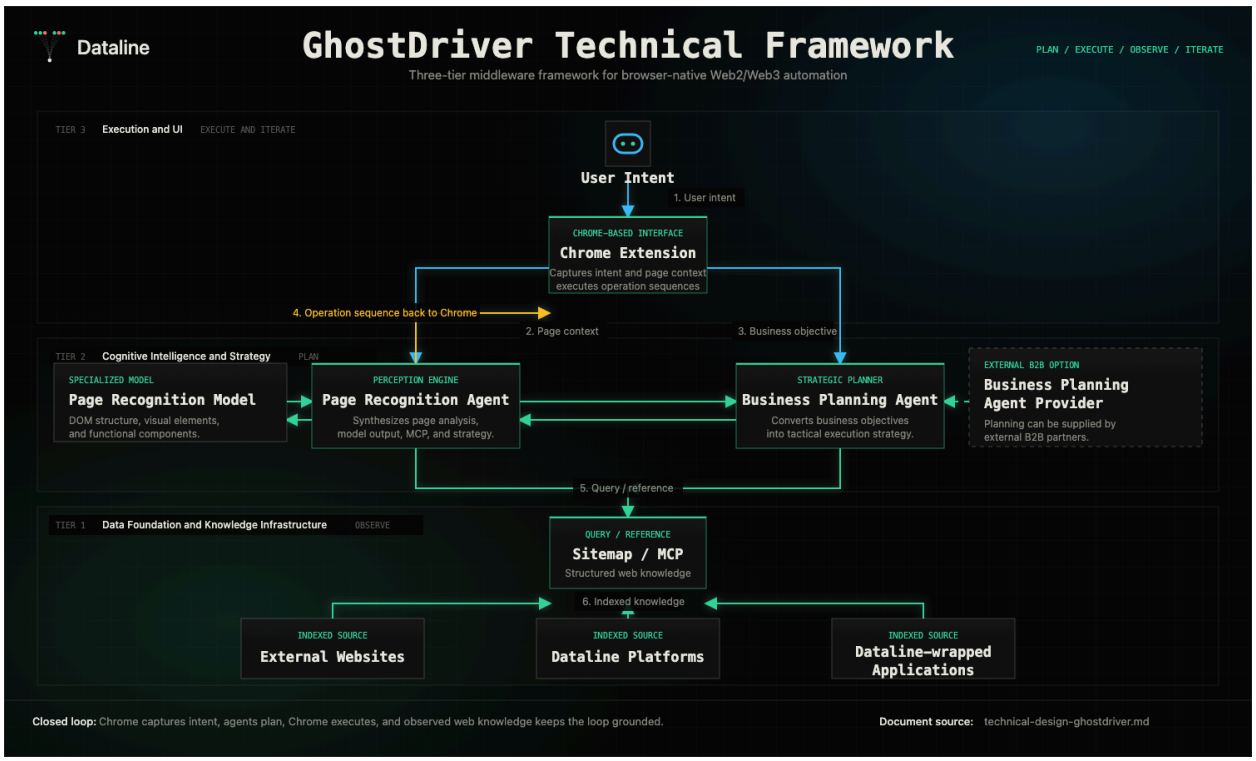


Figure: GhostDriver Technical Framework

GhostDriver is Dataline's browser-native execution engine for turning user intent into reliable actions across Web2 applications and Web3 interfaces. It combines a Chrome-based execution surface, AI-assisted page recognition, structured site knowledge, and optional business-planning logic into a closed loop: observe the page, plan the action sequence, execute in the browser, and iterate from the result.

The architecture is designed for environments where scripted selectors are brittle: changing DeFi interfaces, authenticated Web2 pages, and multi-step workflows that require page understanding, user context, and runtime validation.

Core Design Principles

1. Browser-Native Execution: GhostDriver works inside the user's browser context through a Chrome extension, allowing it to operate on real pages, authenticated sessions, and interactive Web2/Web3 interfaces without requiring every target system to expose an API.

2. Perception-Cognition-Execution Loop: The system separates page understanding, strategy generation, and browser execution. This separation improves reliability because each layer has a clear responsibility and can be monitored independently.

3. Structured Web Knowledge: Indexed websites and Dataline-wrapped applications are converted into queryable sitemap knowledge. The MCP interface gives agents a consistent way to retrieve page structure, workflows, and implementation context.

4. Partner-Extensible Planning: Business planning can be supplied by Dataline or by external B2B partners. This allows the same execution engine to support different workflow policies, compliance constraints, and vertical-specific logic.

Core Architecture

Figure 4 summarizes the three-tier middleware framework across browser execution, cognitive planning, page recognition, business planning, and structured web knowledge. The sections below describe the responsibilities of each tier.

Tier 1: Data Foundation and Knowledge Infrastructure

The foundation layer builds and maintains the structured knowledge that grounds browser automation.

- **Web Indexing:** GhostDriver indexes external websites, Dataline platforms, and Dataline-wrapped applications. The indexed representation captures page structure, UI patterns, functional components, and common workflow paths.
- **Sitemap / MCP:** Indexed data is transformed into structured sitemaps and exposed through MCP (Model Context Protocol). This allows agents to query page knowledge at runtime instead of relying only on static prompts or brittle selectors.

Tier 2: Cognitive Intelligence and Strategic Planning

The planning layer converts page context and business objectives into executable operation sequences.

- **Page Recognition Agent:** The agent analyzes DOM structure, visual elements, functional regions, and model output from a specially trained page recognition model. It also queries Sitemap / MCP to recover page-specific context.
- **Business Planning Agent:** This component translates high-level business objectives into tactical workflow strategy. In B2B deployments, the planning component can be supplied by a partner to reflect partner-specific products, policies, and operating rules.

- **Operation Sequence Generation:** The Page Recognition Agent synthesizes page analysis, model output, sitemap knowledge, and business-planning directives into an ordered operation sequence with action steps, validation points, and fallback paths.

Tier 3: Execution and User Interface

The execution layer turns operation sequences into browser actions while keeping the user in control of sensitive operations.

- **Chrome Extension Interface:** Captures user intent, observes page context, receives operation sequences, and executes browser automation commands.
- **Real-Time Execution Engine:** Converts operation sequences into concrete browser actions, validates intermediate states, and adapts when a page changes or a step fails.
- **User-Controlled Sensitive Actions:** For Web3 workflows, GhostDriver preserves non-custodial boundaries. Wallet signatures, irreversible transactions, and asset movement require explicit user approval.

Operational Workflow

The workflow starts when a user expresses intent through the Chrome extension. The extension captures intent and current page context, then sends them to the Page Recognition Agent. The agent analyzes the page, queries Sitemap / MCP, incorporates business-planning constraints, and generates an operation sequence. The Chrome extension executes the sequence, validates intermediate states, and returns observations to the planning loop when the workflow needs adjustment.

Summary

GhostDriver is the execution layer that allows Dataline to act across real browser environments, not just answer questions. Its value comes from combining browser-native automation, structured web knowledge, AI-assisted page recognition, and partner-extensible planning into a closed loop suitable for Web2/Web3 workflows.

7.4. ChatPilot: Conversational Gateway for Web3 Trading and Intelligence

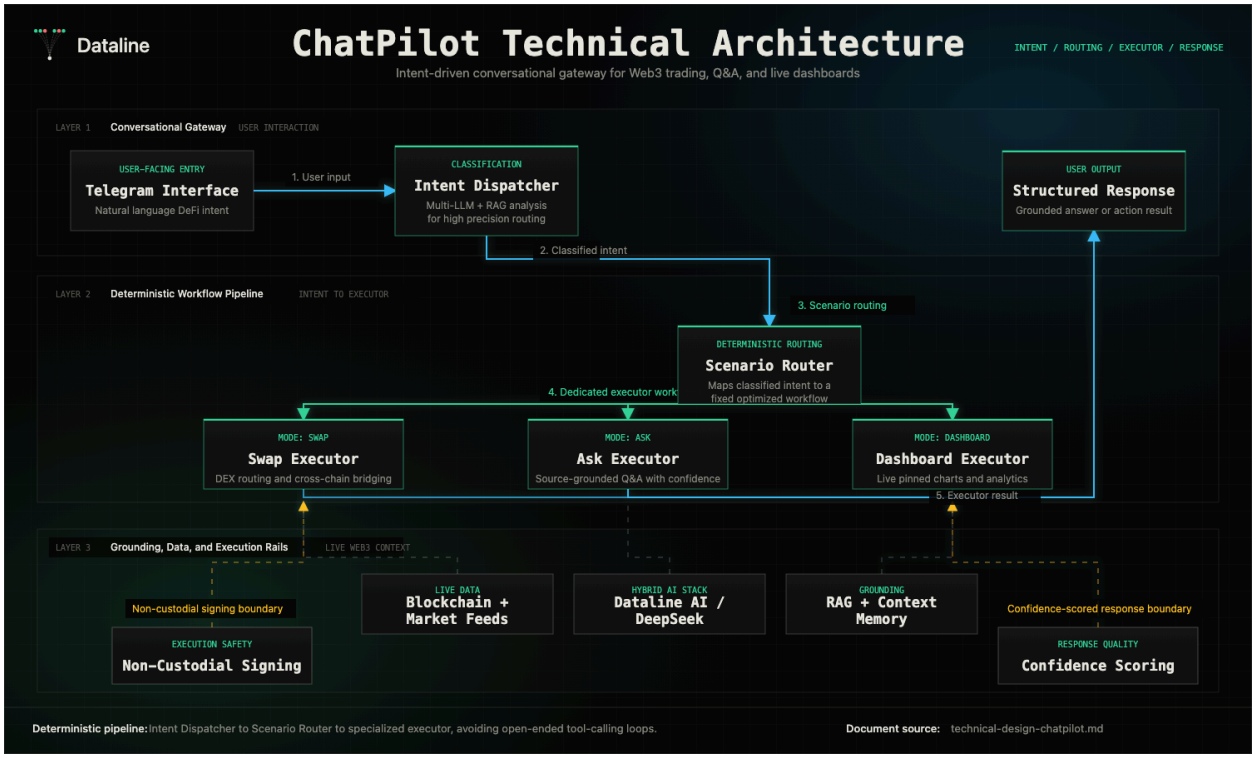


Figure: ChatPilot Technical Architecture

ChatPilot is Dataline's user-facing conversational gateway for Web3 trading, market intelligence, and live portfolio workflows. It converts natural-language requests into deterministic execution paths, allowing users to ask questions, route swaps, and generate dashboards without navigating multiple DeFi interfaces manually.

The product is built around a core problem in financial AI: generic chatbots can produce plausible but ungrounded answers, and many cannot access live blockchain state. ChatPilot addresses this by combining intent classification, retrieval-augmented grounding, real-time data feeds, specialized executors, and confidence-scored responses.

Core Design Principles

1. Workflow-Based Determinism: ChatPilot avoids open-ended tool-calling loops for high-stakes DeFi operations. User intent is classified, routed to a known scenario, and handled by a specialized executor with a predictable workflow.

2. Grounded Real-Time Responses: Market prices, on-chain state, and project context are retrieved from live or verified sources before the final response is generated. Each response includes quality signals such as confidence, freshness, and source grounding where applicable.

3. Multi-Model Specialization: Different model families can be used for conversation, reasoning, retrieval, and agent operations. This lets ChatPilot optimize for accuracy, latency, and cost rather than forcing every task through one model.

4. Non-Custodial Execution: Trading operations are prepared and routed by the system, but asset movement remains user-controlled. Signing and irreversible transaction approval stay outside the custody of ChatPilot.

Core Architecture

Figure 2 summarizes the intent-driven pipeline from conversational entry through deterministic routing, specialized executors, grounding rails, and structured response. The component descriptions below focus on each module's responsibility.

Key Components

Telegram Interface: The primary conversation entry point. Users express trading, research, or monitoring intent in natural language.

Intent Dispatcher: Performs initial classification with a hybrid AI stack and retrieval-assisted context. It determines whether the request is a swap, market query, project analysis, dashboard request, or general Q&A.

Scenario Router: Maps classified intent to a deterministic workflow. This design gives Dataline clearer control over execution behavior, error handling, and safety boundaries.

Specialized Executors:

- **Swap Executor:** Generates DEX routes, evaluates liquidity, supports cross-chain bridging flows, and prepares non-custodial transaction steps.
- **Ask Executor:** Produces source-grounded answers for market, project, and Web3 knowledge queries.
- **Dashboard Executor:** Generates live charts, pinned market views, and portfolio or token dashboards.

Structured Response: Returns grounded answers, action results, dashboards, or transaction-preparation output with confidence and source-quality indicators.

Technical Features

Hybrid AI Stack

Capability	Role in ChatPilot
Dataline AI	General conversation, product context, and Web3 domain knowledge
Reasoning Model	Complex analysis, multi-step reasoning, and risk evaluation

Real-Time Retrieval Model	Fresh market data, news, and external-source synthesis
Agent Operations Model	Tool orchestration and structured task execution
RAG + Memory	Source grounding, user preference continuity, and context recall

Table 1: Model Capability Matrix

Multi-Chain Support

ChatPilot supports user workflows across major blockchain ecosystems. The conversational interface hides chain-specific complexity while downstream executors handle routing, chain selection, and data normalization.

Context-Sensitive Memory

The memory layer stores conversation context, preferred chains, trading preferences, and recurring user patterns. This allows multi-turn interactions to remain coherent while still requiring explicit user confirmation for sensitive operations.

Confidence-Scored Responses

Responses include confidence indicators based on source quality, data freshness, cross-source agreement, and completeness. This does not eliminate model error, but it makes uncertainty visible and gives users a basis for deciding whether to act.

Operational Workflow

A user submits a natural-language query through Telegram. The Intent Dispatcher classifies the request and retrieves relevant context. The Scenario Router selects the proper workflow and executor. The executor performs the required operation, such as generating a swap route, retrieving verified information, or creating a dashboard. The system returns a structured response with grounding and confidence metadata, while maintaining conversation context for follow-up turns.

Summary

ChatPilot is the conversational layer of Dataline's financial AI stack. Its main strength is the combination of natural-language UX with deterministic Web3 workflows, live data grounding, specialized executors, and non-custodial safety boundaries.

8. Traction

Metric	Value
On-chain transactions powered	19.4 million
Agent task success rate	96.4%
Community members	300,000+
Data sources integrated	22+
Median query latency	~87ms

Dataline was named a winner of the 2025 BNB Chain AI Hackathon.

9. Roadmap

Phase	Status	Focus

Foundation	Live	Core data API, ChatPilot, MCP server, API access
Launch	Now	$10.6M round, Data Launch cohort onboarded
Expansion	Near-term	Open data job network, $DAT staking for access
Scale	Upcoming	ZK attestation tier, machine-payment rails, Solana support
Future	Ahead	Fully autonomous agent-to-agent data economy

10. Summary

Data is the lifeblood of AI. Autonomous agents are capable enough to know what data they need (sometimes better than their builders).

Dataline delivers:

- One integration across all markets and prediction feeds
- Confidence scores on every response, computed from 22+ cross-verified sources
- An open data network where agents and humans can request and fulfill custom data, enforced by economic stake
- $DAT: the alignment layer between data quality and network participants