

Dragon

Decentralizing token data to discover alpha.

Co-created by Harshan and Frog

Last updated: June 1, 2025

email: baddragonmygoodness@gmail.com

Table of Contents

- 1 Introduction
- 2 Market Dynamics & Opportunity
- 3 Dragon
- 4 AlphaSwarm & Magic
- 5 Project Token
- 6 Development Goalposts
- 7 Project Note

1

Introduction

Why are you not in control of token analysis when you do your own research?

Dragon is a data project powered by community-built token analyses. Each analysis sources on- or off-chain data and produces specific knowledge about one aspect of a token project's design. Think of the best analyses found on Bubblemaps, BullX, Solscan, GMGN and then combine them with the best public insights from KOLs, Dune analysts, and real-time group chats. Token data today are scattered among these platforms or gated by insiders with custody of them. Retail traders lose momentum when they manage open browser tabs and ping multiple telegram bots or pay subscription fees to get to the information they need to trade with intelligence. This is the game of discovering **alpha**—yet nobody has defined it yet simply as collecting thorough, real-time data and analyses.

This project grants any data analyst the opportunity to develop a bite-size **data module** as part of an open-source browser extension that aims to discover alpha in Web3 as it is composed of an aggregation of token data analyses. This is DYOR for the next era—an opportunity to customize your research process using analyses that you find to be most effective while the community learns what is most impactful at scale. With use, Dragon becomes a trader's AI companion trained on their strategic profile or “behavioral fingerprint,” matching tokens they've bought to new tokens they haven't discovered yet.

Dragon's AI compose a picture of retail trading and volume patterns across each token analysis ranked by highest demand and highest effect, composing alpha into modular data indicators for volume change in trading activity. We aim to create a platform for consumers to participate and inform the intelligent design of Web3 by participating in its token research and development, initiating more projects that are data-driven.

2

Market Dynamics & Opportunity

Trading bots across blockchains have seen over \$80B in lifetime volume¹ and bot trading on Solana alone surpassed a volume of \$770m one day earlier this May². As the chain with the highest number of total network transactions³, Solana's users have found value in a new category of tooling called **token scanners** like Syrax, SarumAI, and trench.bot. These DYOR assistants currently see ~500k monthly active users with rugcheck.xyz leading the charge⁴. Even Solana's major trading platforms Photon, BullX, and Axiom are incorporating features that display token scanning data adjacent to the chart, acknowledging that data transparency is a top of mind concern for their users. Due to recent high profile scandals⁵, extractive pump.fun launches⁶, and even obscure behaviors at institutional levels⁷, data transparency still goes without a clear standard for retail trading in crypto.

Maximum Extractable Value (MEV) is an integral part of blockchain development, interacting with or affecting an average of ~25% of all transactions in every single block on Ethereum, Solana, and Binance Smart Chain⁸. Its impact can increase to over 66% of each block's transactions during periods of peak volatility and high network traffic⁹. MEV is called “the double-edged sword” of crypto¹⁰ as it is an essential incentive mechanism for validators to produce new blocks, but it also creates opportunities for bad actors in the order flow to abuse power. Projects like Flashbots on Ethereum have sprung up to help balance MEV’s benefits by democratizing its access at the infrastructure level, but no current project works at the consumer level to equip traders with information on its presence in any one token project, barring market participation in the demand for MEV.

To date, most of **Web3’s user data are produced defensively**—like for institutional compliance, major threat detection, and top-down market analysis information. The applications of user trading data as a constructive force in blockchain are still largely underdeveloped. Dune Analytics has proven that community-sourced data analytics has high demand and Nansen has shown that there is immense value in trading data. There is open space for a platform to combine both and create a tool made for and by retail traders.

3

Dragon

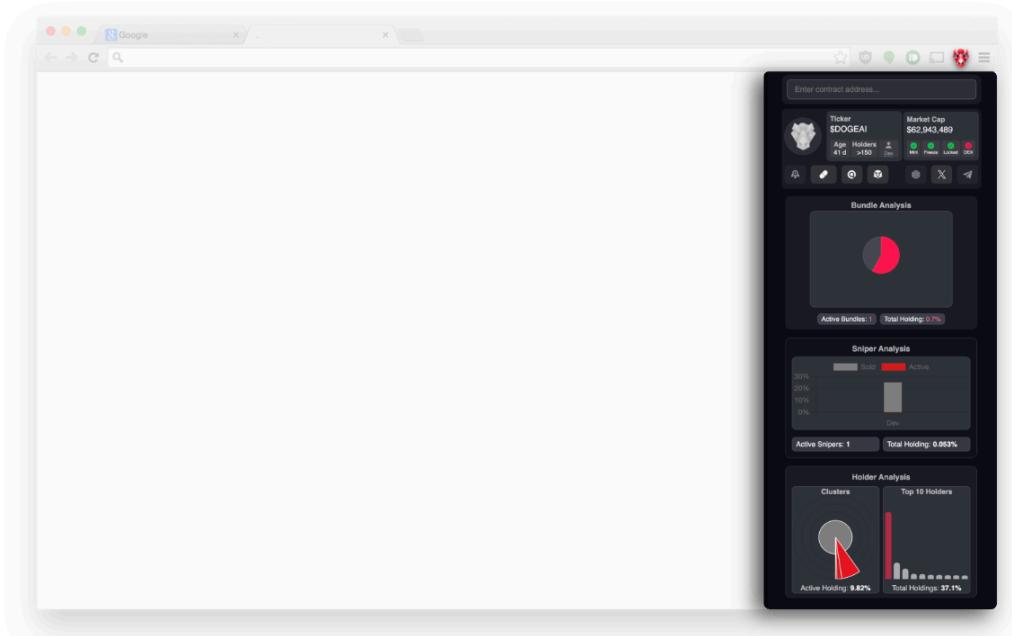


Figure 1: Dragon’s extension stays open inside your browser session, staying with you as you navigate the internet, trade tokens, or conduct DYOR.

To use Dragon’s browser extension, just paste in a Solana token’s contract address. The panel becomes a dashboard of real-time information on that token using community-developed analyses that source on- and off-chain data. These analyses remain available and update as a user navigates the internet, becoming a DYOR companion that stays with traders wherever they go.

Dragon uses a wallet-based single sign-on to save a user’s settings across devices and monitors their on-chain purchase transactions as they relate to the analyses they select. This means that when you buy a token, Dragon will **snapshot** the analyses you have in your extension in order to create a strategic trader identity and build your behavior profile. Taking advantage of read-only methods, this approach to on-chain data mixed with platform usage minimizes wallet security risks.

Modularity

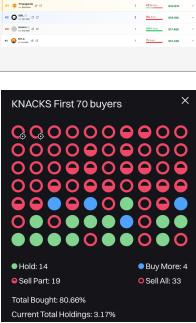
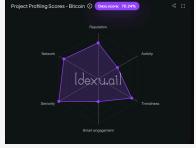
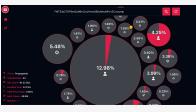
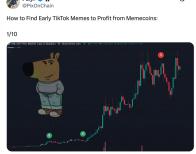
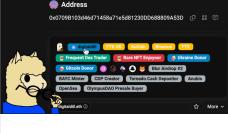
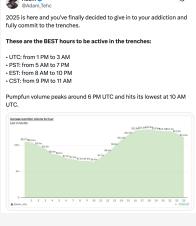
Analyses are composed into bite-size **data modules** that each analyze one specific aspect of a token's design or project quality. Dragon's release comes equipped with four basic prototype modules, but the goal is for a community of developers to propose and contribute new ideas going forward. A data module can be developed to analyze any aspect of a token project, using on- or off-chain data as its input, and is rewarded with a bounty in \$DRAGON token supply if it is ultimately included on the platform.



*Figure 2: Token analyses are designed as customizable **data modules**—each focusing on a single token design metric. Users can mix, match, and even build these modules themselves.*

Inspiration

Token analyses today are produced by various individuals and hosted on different platforms. These data custodians have the opportunity to distribute insights with more accessibility and reach, participating (and competing) in an open-sourced token research community. This will inspire more developers and analysts to build and improve on token analyses, developing more ideas as the community grows and trader demand evolves.

Inspiration	Potential Analysis	Current Custodian	Inspiration	Potential Analysis	Current Custodian
	KOL / smart money holders Sniper & insider buy counter	StalkChain GMGN		Token narrative profiling	Dexu / @cryptokoryo
	JITO bundle visualizer	trench.bot		Liquidity pool sandwich score	sandwiched.me
	Top 10 holder awareness	Solscan		Top holder previous transaction behavior	RugChecker
	Clustered wallets	Bubblemaps		Dev / deployer overview	Trenchy Bot
	Liquidity - market cap ratio	@kompreni		Holder trends by time held	HolderScan
	CT / whale holder % and movement	@dethetive		Essential security heuristics	Birdeye (and others)
	TikTok / X sentiment matching	@PixOnChain		Holder profiles / style tags	DefiLlama
	Time of day notifications	@AdamTehc		DEX paid? Project links and aesthetics	DEX Screener

Community Development

At launch, developers will propose new data modules by opening pull requests on GitHub that are to be verified, accepted, and integrated into the platform by the core team. Future extensibility is enabled upon developing a web-based module compiler, known as the **Module Forge**. This will be a drag-and-drop browser sandbox where coders and even non-JS-savvy quants can upload Python or Rust snippets, compile them to WASM (WebAssembly), and instantly test them with historical token datasets.

With the incorporation of a permissionless Module Forge, a web-based market will also be deployed where token holders can up-vote, down-vote, or request new modules for development. Traders will use this with a dual purpose to toggle modules on or off to their Dragon, without needing to restart or update the extension. This allows for an open-sourced system that takes central control out of the equation, allowing token analyses to proliferate and provide value to the community at its demand and at its users' direction.

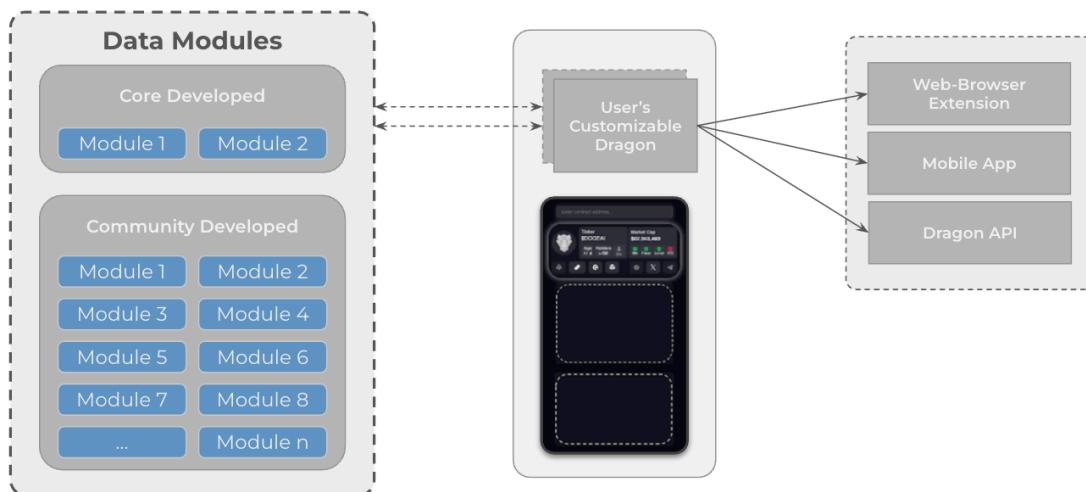


Figure 3: Dragon's architecture includes both core developed modules and community developed modules. The end product syncs across a user's browser extension, a mobile app, and a public API.

A natural development for community-sourcing data modules will be the competitive reach and incentives for those who develop modules that deliver the highest impact for users, as measured by correlation to volume changes across tokens in scope and proliferation among Dragon users. This module ranking system will be constructed as the project grows, rewarding the developers with the highest impact with more influence in the project's direction.

Mobile Application

Dragon's mobile app mirrors every feature of its browser extension by extending the module backends to traders while they are on the road. Users receive push notifications when enabled modules cross personalized data thresholds, staying aware of critical data spikes or sentiment shifts while away from the desktop. Additionally, because Dragon tracks desktop versus mobile usage, the Magic module (info below) will use the behavior deltas between devices to refine personalized token matches that it sends to both devices.

4

AlphaSwarm & Magic

An orchestration agentic approach to selective LLM training that captures Web3 alpha with a user's modules, sourcing on- and off-chain data and historical volume changes.

Introduction

AlphaSwarm is a multi-agent observatory designed to recognize token movements by fusing module outcomes, turning on-chain events (DEX volume, wallet flows, MEV activity) and off-chain signals (social sentiment, news, alpha groups) into actionable trading insights. The system doesn't rely on full-streaming data. Instead, it learns from significant market moments—volume spikes, rug pulls, pumps—and builds a behavioral map of modular signal combinations during those key events.

These snapshots of module data and volume fluctuations become training data for a central LLM-powered orchestrator agent that learns trend patterns holistically, rather than isolating the effect of a single signal. It rates new events on a 0–10 scale (quantitative) based on how much they resemble past scenarios and provides a narrative explanation (qualitative)—offloading the cognitive burden from traders.

Event-Curated LLM Training on Meaningful Windows

Rather than training the Large Language Model (LLM) on an exhaustive stream of real-time data—much of which is noisy or uninformative—Dragon adopts a more efficient and insightful approach: event-curated training. Specifically, the system identifies periods in the dataset with significant volume changes and extracts the full set of module outputs corresponding to those timeframes. These time-bounded, high-impact segments—referred to as event windows—serve as the training corpus for the orchestration LLM.

Each data module in Dragon is handled by a dedicated agent that monitors and logs its respective features over time. The LLM then learns how different combinations and permutations of these module trends correlate with actual market movements. Rather than attempting to assign linear causality to any single feature, the LLM learns complex, multi-dimensional patterns that characterize specific market scenarios. During inference, when real-time data is fed into the system, the orchestrator evaluates the new inputs against the

historical pattern library and computes a similarity score to past events. This methodology enables scenario recognition based on learned multi-signal patterns, offering a more robust and context-aware form of predictive analytics.

Agentic Confounding Resolution via Orchestration

One of the fundamental challenges in financial modeling is the issue of confounding, where multiple variables change simultaneously, making it difficult to isolate the effect of any single factor. AlphaSwarm avoids the faulty attempt of strict real-time causality per module, which is typically unreliable in the stochastic and noisy environment of crypto markets. Instead, the system relies on a team of agentic modules, each tasked with monitoring a specific metric—such as **Sniper %** or **Bundle %** in a token—individually over time.

These agents feed their observations into the orchestration LLM, which is trained to evaluate the collective context of module behaviors rather than individual contributors. For example, if past event data shows that sharp increases in **Sniper %** combined with flat **Bundle %** and rising liquidity often lead to significant frontrun-pump scenarios, the model learns to recognize that pattern. When similar conditions reappear in live data, the orchestrator surfaces the match—not as a hard statistical claim, but as a pattern-aligned scenario based on prior outcomes. In this way, the system resolves confounding, using coordination and historical pattern matching instead of assuming modular independence.

Qualitative & Quantitative Output Layer

AlphaSwarm delivers a scenario-based scoring mechanism ranging from 0 to 10. A score of 0 indicates that current conditions bear no resemblance to known high-volatility scenarios from the training data, whereas a 10 reflects a strong historical match to a past pattern with significant market impact. This scoring system reflects confidence in the model's learned memory rather than a strict statistical likelihood.

Each score is accompanied by a breakdown of per module historical precedence—indicating which components contributed most to the pattern match—and a natural language explanation. The explanation is generated through an embedded NLG (Natural Language Generation) pipeline, allowing traders to quickly interpret the rationale behind each signal. This dual output of numeric score and qualitative insight helps traders build intuition and make informed changes in their trading focus, even in situations where pure mathematical certainty is lacking.

AlphaSwarm Architecture

The AlphaSwarm is built as a robust, real-time data analytics system that integrates a full-stack modular AI pipeline. It sources data from the various community developed modules that source both on-chain and off-chain events.

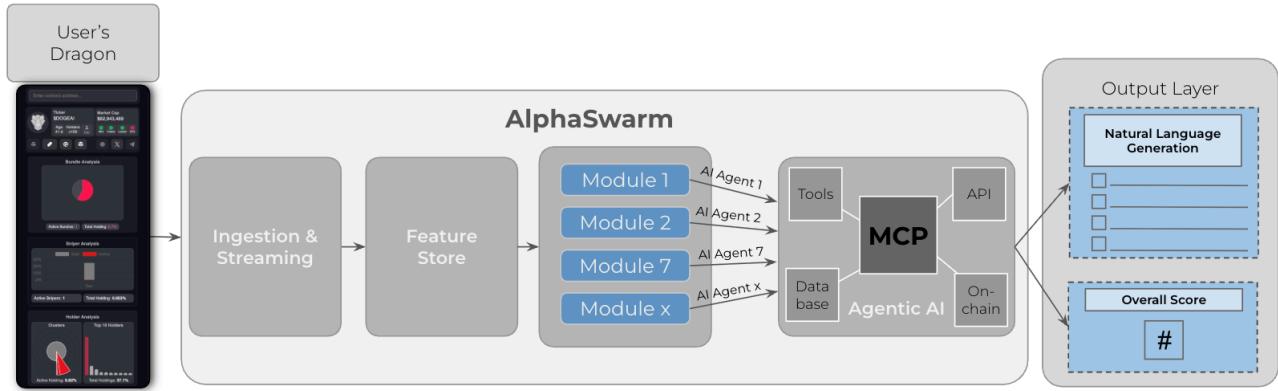


Figure 4: Dragon's Core AlphaSwarm Module

All of the data streams are ingested through a high-throughput infrastructure using technologies like AWS Kinesis or Apache Kafka, which organize the signals into time-aligned buffers. These are then written into a dual-layer feature store: a hot layer (e.g., Redis or KeyDB) for sub-second latency access and a cold layer (e.g., Apache Iceberg on S3) for long-term historical analytics and backtesting.

The core of AlphaSwarm is the modular analysis stack, composed of discrete agents working on each module. These module outputs are passed to the Agentic AI hub, or **Model Context Protocol (MCP)**, where agents using LangGraph or AutoGen frameworks validate, enrich, or challenge one another's findings to orchestrate the final output.

What is Model Context Protocol (MCP)?

MCP is an open standard, open-source framework introduced by Anthropic to standardize the way LLMs integrate and share data with external tools, systems, and data sources. Inspired by agentic AI systems like Microsoft's AutoGen, MCP ensures real-time adaptability without retraining the full pipeline, echoing a broader shift in corporate AI towards modular, context-aware intelligence.

For more information on MCP, visit Anthropic's announcement [here](#).

In Dragon, the MCP is a coordination layer that dynamically assigns trust scores to analytic modules based on how well their signals align with actual volume changes. It acts like an ensemble controller—boosting modules with strong predictive value. The MCP serves as a shared trust layer, dynamically updating the weight w_i of each module based on its historical signal quality. A final composite hype index $H = \sum_i(w_i \times signal_i)$ is computed for each token, where:

$$H = \text{total hype score}$$

$$\sum_i = \text{sum over modules}$$

$$w_i = \text{module trust weight}$$

$$signal_i = \text{module output signal}$$

AlphaSwarm's results are rendered in two formats through its output layer: a quantitative score from 0-10 (and perhaps a future radar chart!) that summarizes the weighted module contributions and a natural language summary that explains the reasons why a token may have a significant shift in volume soon. These outputs are delivered via the AlphaSwarm API to both the browser extension and mobile app, offering an immediate and accessible interface for traders. The entire system is also integrated with the **Magic** engine, enabling one-click new token discovery based on a user's trading patterns.

A Personalized Magic Module

Dragon enhances AlphaSwarm's framework with **Magic**— a personalized module designed to increase your action potential as a trader. Every time you interact with another data module or make an on-chain transaction, your behavior is captured and embedded into a dynamic user profile. This profile encodes not just your preferences, but your actual trading behaviors—the risk levels, liquidity patterns, and market sentiment signals at which you engage with tokens at the most—forming a “behavioral fingerprint” of your strategic profile based on the data analyses that you use.

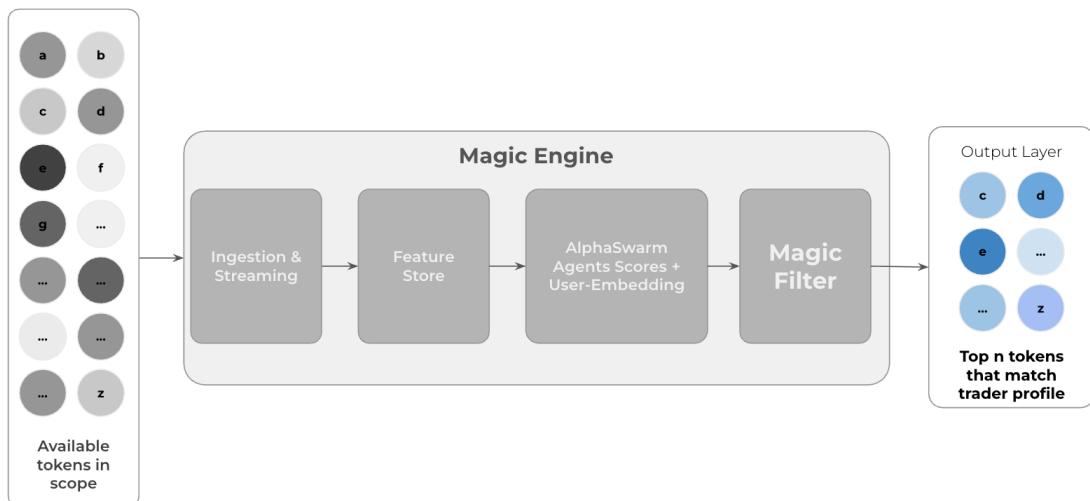


Figure 5: Dragon's Core Magic Module

This fingerprint becomes the input to a two-tower transformer architecture at the core of the Magic engine. The first tower ingests your usage history: module activation patterns, past transactions, and clickstream events across the extension. The second tower encodes AlphaSwarm's fused analytics that have been passed through the trust-weighted MCP. This means every token embedding already reflects the most relevant AlphaSwarm-derived signals.

During training, the system optimizes a learning objective that prioritizes tokens you've historically interacted with or traded. Over time, your preferences become more refined and context-aware as AlphaSwarm agents re-weight signal quality, ensuring that your new tokens are not just personalized, but also responsive to changing market dynamics.

For fast and scalable retrieval, Dragon indexes all token embeddings using FAISS's HNSW (Hierarchical Navigable Small World) algorithm, enabling near-instantaneous nearest-neighbor lookups. Your personal embedding is updated in real-time using lightweight online learning steps and consolidated with a nightly batch retrain to capture longer-term evolution. For new users without enough interaction history, the system applies clustering algorithms like DBSCAN to assign a cold-start profile by grouping you with traders showing similar early-stage signals.

The Magic Button Experience

Once the Magic engine identifies a set of top-matching tokens based on a fusion of your behavioral fingerprint and AlphaSwarm's signals, the Magic button offers a one-click bridge to execution. With a single tap, the Magic Button filters out all non-matched tokens on your selected platform (e.g., DEX Screener, pump.fun), and preloads parameters. In effect, AlphaSwarm fuels the intelligence, Magic delivers the personalization, and the Magic Button enables an action.

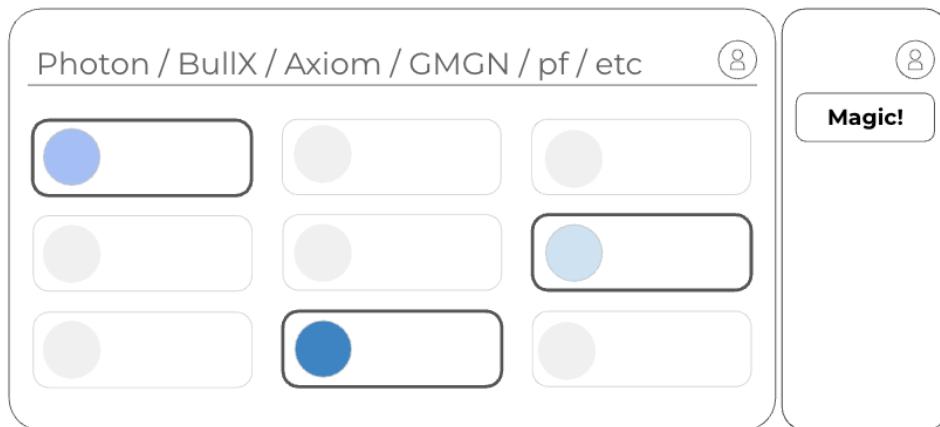


Figure 6: The Magic button interacts with your browser, eliminating the tokens onscreen that do not match your trading profile.

This integration of insights, preferences, and execution expedites the loop for intelligent trading—giving users an advantage during new token cycle iterations.

Parallel Enhancements to AlphaSwarm & Magic

Dragon will eventually include a built-in conversational chatbot—your on-demand trading assistant embedded in the extension and mobile app. This chatbot will leverage a retrieval-augmented generation (RAG) pipeline to pull in exactly the right Wikipedia passages, module data, and real-time agentic analytics you ask for. LangChain agents will orchestrate tool calls for signal queries, while Dragon connects to your encrypted profile and transaction history so every answer is personally tailored—becoming your true AI companion.

5

Project Token

CA:

Utility

There are three utilities the \$DRAGON token provides to holders.

1. Rewards a developer for a successful contribution of a token analysis module.
2. Enables a user's access to the core AI modules (AlphaSwarm and Magic).
3. Facilitates a DAO governance of the project's development and treasury allocation.

These three functions will be enabled gradually and with increased detail as the project progresses.

Rollout

- I. \$DRAGON is launched using Believe: 1% of all trading fees go to the project's treasury, 0.5% in SOL and 0.5% in \$DRAGON
- II. Project contributors are given tokens from the \$DRAGON treasury to promote community growth and project participation
- III. Project overhead and administrative costs are paid out of the SOL treasury to streamline transactions and allow for the widest scope of development opportunities
- IV. When trading fees and project growth have stabilized, the core team will build a smart-contract that automates treasury distributions and payments
- V. When ready, the smart-contract is handed over to a DAO of \$DRAGON token holders to decentralize control of the treasury and the project's development
- VI. The DAO and core team keeps building together, fully decentralized

The governance rules and logistics of the DAO will be decided in time, with both the token holders and the core team's combined input.

6

Development Goalposts

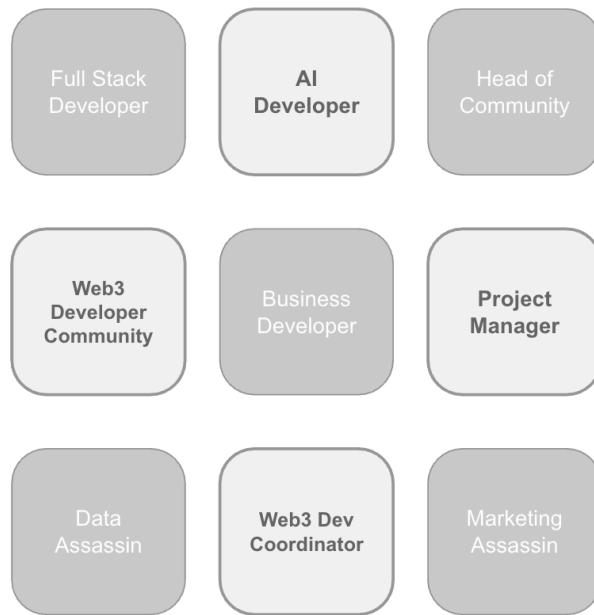
June	Team hired!	Wallet login enabled	Treasury automated and justified
July	Charts redesign for first modules	RPCs built to first modules	Logo & mascot redesigned
Aug	Customizable extension	Dataset created: AlphaSwarm v0	Web hub for customization
Sep	Training begins: AlphaSwarm v0	Module Forge developed	
Oct	Training begins: Magic v0	First release: AlphaSwarm v0	DAO infra begins
Nov	Mobile app	Magic v0 first release	Multi-database integrations
Dec		ETH / BSC / ... expansions?	DAO takes the wheel

[^]Dependent on labor & resources. Could go faster if we go higher and slower if we go lower.

7

Project Note

This project has been launched by Harshan (Harshit) and Frog (Andrew), who are an AI researcher and a project manager by background. They consider the third founding member to be a Web3 community of developers. This wasn't really the original plan, but it became a happy consequence of not having any blockchain development experience between the two of us.



If you are interested in joining the team, please send an email to baddragonmygoodness@gmail.com with an introduction and your relevant experience!

Our first goal is to hire a team of specialists who can work alongside us in a committed capacity, including a Web3 developer who will coordinate the community module integrations. A core design aspect of the project is that its success is a direct function of the market's demand. If it turns out that modular and open-sourced token data is popular among retail traders, we are prepared for the window of opportunity to publish research on user data as it relates to transactions, volume, and effective token analysis modules. The insights gained from this research promises value for blockchains, exchanges, and firms invested in better token designs and the intelligent expansion of Web3.

Our advantage lies in our daily exposure to institutional-grade AI and research, which we apply directly to this project's development. Harshit and Andrew met in a Masters of Data Science program at the University of Maryland at a time when Harshit had just completed research at IBM co-developing in-house generative AI systems. Harshit currently works at Amazon building scalable infrastructure for Bedrock AI and developing the latest implementations of MCP methodology. Andrew approached Harshit for tutoring in ML, but ended up trading memecoins during their sessions instead of learning the material. He previously co-founded the blockchain startup Cent, which raised a seed round after their Valuables project gained traction for tokenizing tweets as NFTs on Ethereum. Andrew brings a multi-disciplinary perspective to project management, drawing from an international background and experiences managing global campaigns and creative projects outside of the tech/Web3 bubble.

We see ourselves as outsiders jumping in the deep end—retail memecoin traders that lost money to sniper bots and crypto-insiders. We look forward to both stewarding and being stewarded by the vision-forward community that has a mission to capture and equip retail traders with data-driven alpha.

Project Links

Website

<https://alpha-dragon.ai/>

Download the Prototype

<https://chromewebstore.google.com/detail/dragon/ncbglgbplhnbekllhogabdefjidbkoe>

GitHub

<https://github.com/alpha-dragon-org>

X / Twitter

<https://x.com/AlphaDragonAI>

Telegram

<https://t.me/+OU0SLVfcxEZhZWQx>

Demo

<https://vimeo.com/1062123553>