

Dragon

Decentralizing token data for
retail analysis.

Co-created by Harshan and Frog



Last updated: May 30, 2025

email: baddragonmygoodness@gmail.com

Table of Contents

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

1

Introduction

Why aren't you in control of the analyses when researching new tokens?

Dragon is a research data platform powered by community-built token analyses. Each one sources on- or off-chain data and produces specific knowledge about an aspect of a token project's design. Think of the best analyses found on Bubblemaps, Trench Bot, DefiLlama, GMGN and combine them with the best public insights from KOLs, Dune analysts, and the real-time alpha in gated group chats. Token data today are scattered among platforms and insiders that have custody over their development. Retail traders lose momentum when they must manage open multiple browser tabs and ping different telegram bots or pay subscription fees just to get the information they need to trade with intelligence.

Dragon grants any token analysis the opportunity to be developed into a bite-size module in a browser extension that stays with traders wherever they go on the internet. This is DYOR for the next era, an ability to customize your research pipeline with just the analyses you want, never sifting through unnecessary information again. With continued use, your Dragon becomes a smart companion using AI that learns your strategic trader profile or "behavioral fingerprint," matching the tokens you've bought to new tokens you haven't discovered yet, based on the analyses that you trade with.

At scale, Dragon composes a picture for retail trader preferences across the token analyses in highest demand, constituting a more specific definition for how we track **Alpha**, one where modular data become the leading indicators for volume changes in Web3. This project aims to be an opportunity for consumers to participate and inform the intelligent design of blockchain's platforms and to initiate future token projects that are specifically data-driven.

2

Market Dynamics & Value

Trading bots across blockchains have seen over \$80B in lifetime volume¹. Bot usage on Solana alone surpassed a trading volume of \$770m in one day earlier this month². With the growth of these trading bots, we've also seen increased use of token scanners on Solana like Syrax, Sarum, and Trench Bot. These DYOR assistants see around ~500k monthly active users, with rugcheck.xyz in the lead³. With recent high profile scandals⁴, extractive pump.fun launches⁵, and obscure behaviors at institutional levels⁶, a thorough token research methodology and a token design equilibrium at the infrastructure level still goes without solution in Web3. As Solana is the chain with the most transactions and fees for all blockchains⁷, its major trading platforms like Photon, BullX, and Axiom are incorporating features that display token data adjacent to the chart, acknowledging these concerns as forefront among their users.

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 blocks, but also creates opportunities for bad actors in the order flow to abuse their power. Many projects like Flashbots on Ethereum have sprung up at the infrastructure layer to help mitigate the extraction that MEV can have by democratizing the opportunity to capture it, but no current projects work at the consumer level to support retail trading or promote participation in the market demand for MEV activity.

To date, **Web3 user data has been focused on defense**—like institutional compliance, threat detection, and top-down market analysis. Dune Analytics has shown that community-developed data is in demand and Nansen has shown that institutions are benefiting from immense trading data. Dragon puts trading data on the offensive by equipping retail with fine-tuned token analyses they control.

3

Dragon

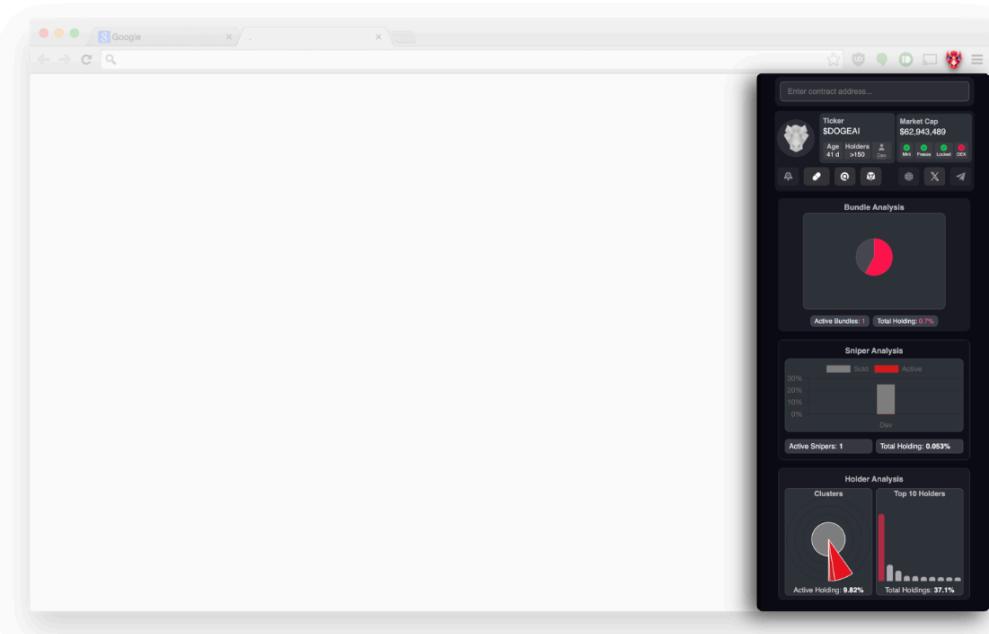


Figure 1: Dragon’s browser extension stays open alongside your preferred trading platform, and stays with you as you change tabs during your DYOR process.

The browser extension can be seen as a companion in the trenches. Users paste a token’s contract address and produce a dashboard of real-time analyses using various sources of on- and off-chain data. Having these token analyses immediately available as users navigate the internet provides traders with a DYOR companion that stays with them wherever they go, and updates in real-time.

Dragon uses a wallet-based single sign-on to save a user's settings across devices and monitors their on-chain purchase activity as it relates to their token analyses. This means when you buy a token, Dragon will snapshot the analyses that you used to research it and create a strategic identity that builds your trader profile. Taking advantage of only needing read-only methods, this approach to on-chain data mixed with platform usage minimizes wallet security risks.

Modularity

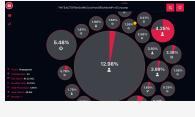
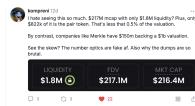
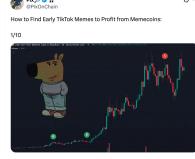
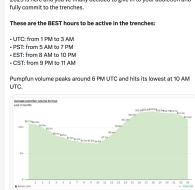
Token analyses are divided into bite-sized **data modules** that each analyze one specific aspect of a token's design or project quality. Dragon's release comes equipped with four basic prototype modules, but the goal is for a community of developers to propose and contribute new ideas from this point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

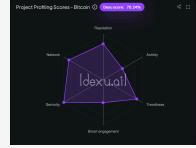


*Figure 2: Token analyses are designed as customizable **data modules**—each focusing on a single token design metric—that users can mix, match, and even build themselves.*

Inspiration

Token analyses today are produced and hosted by various platforms and individuals. These data “custodians” have an opportunity to distribute their insights and participate in an open-sourced token research community, inspiring more developers and analysts to build and improve on them.

Inspiration	Token Analyses	Current Custodians
	KOLs / Smart Money Tracker	StalkChain
	Sniper & Insider Charts	GMGN
	JITO Bundle Visualizer	Trench Bot
	Top Token Holders	Solscan
	Clustered Wallets	Bubblemaps
	Liquidity - MarketCap Ratio	@kompreni
	Whale Movements	That one guy on X who claims to have a great whale wallet tracker on a private spreadsheet
	TikTok / X Sentiment	@PixOnChain
	Time of Day Trending	@AdamTehc

Inspiration	Token Analyses	Current Custodians
	Narrative Profiling	Dexu / @cryptokoryo
	Most sandwiched liquidity pools	sandwiched.me
	Top holder transaction history	RugChecker
	Dev / deployer profiler	Trenchy Bot
	Holding trends	HolderScan
	Security issues	Birdeye
	Wallet tags	DefiLlama
	DEX paid: weblinks and imagery	DEX Screener

Community Development

At launch, developers will propose new data modules by opening pull requests on GitHub, to be verified and integrated by the core Dragon team one by one. Future extensibility is enabled upon developing a web-based module compiler, known as the **Module Forge**. This is 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 data.

With the incorporation of a permissionless Module Forge, a web-based storefront will also be deployed, where token holders can up-vote, down-vote, or request new modules for development. Users will use this space to toggle modules on or off to their own Dragon, without needing to restart or update their extension. This allows for an open-sourced system that takes the centralization out of the equation, allowing token analyses to proliferate and provide value to the Web3 community at its own direction.

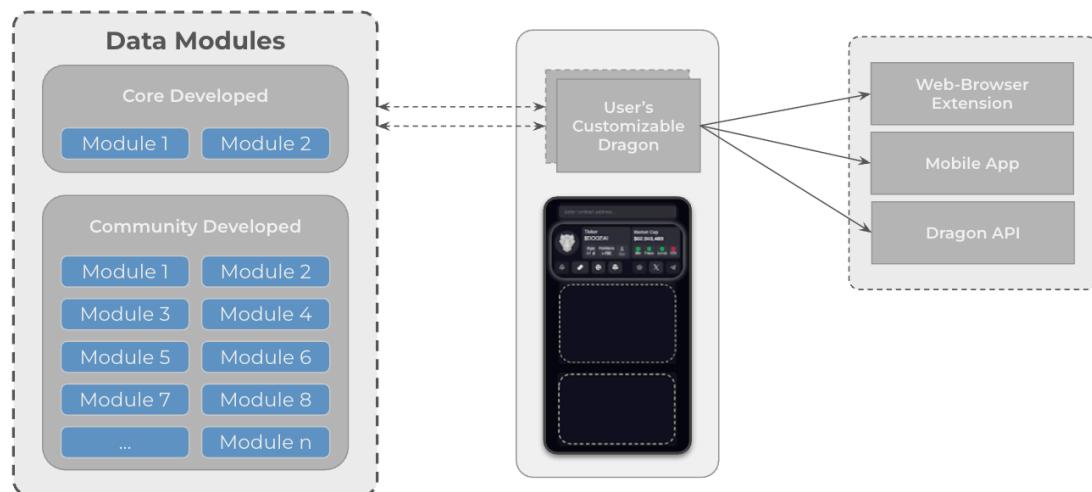


Figure 3: Dragon's architecture includes both core developed modules and community developed modules. The end product syncs across a user's browser extension, a mobile app, and a public API.

Mobile Application

Dragon's mobile app mirrors every feature of the 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, Magic Module (info below) will use the behavior deltas to refine personalized token matches that it sends back to both devices.

4

AlphaSwarm & Magic

An orchestration agentic approach to selective LLM training that captures Web3 **Alpha** with a user's modules, using on- and off-chain token data analytics.

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—Independently 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.

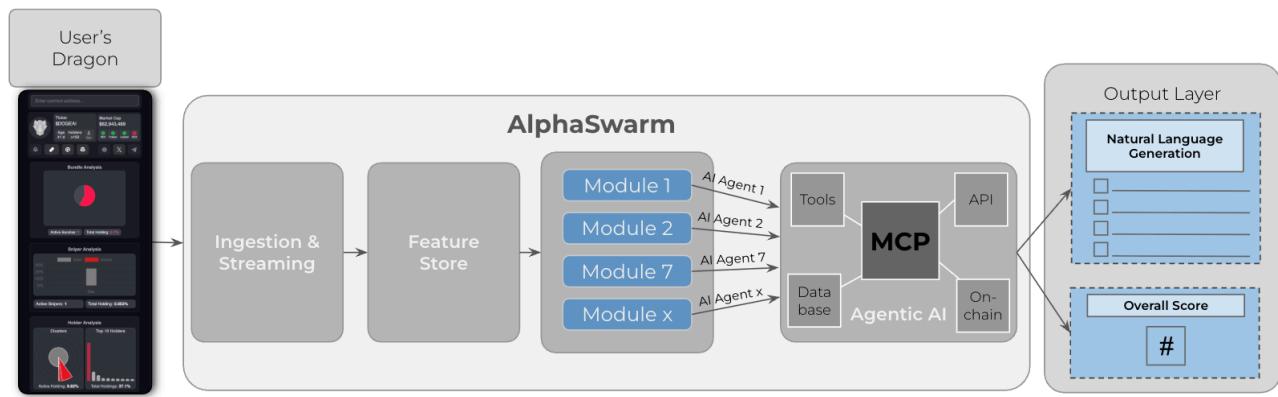


Figure 4: Dragon's Core AlphaSwarm Module

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:

$$\begin{aligned} H &= \text{total hype score} \\ \sum_i &= \text{sum over modules} \\ w_i &= \text{module trust weight} \\ signal_i &= \text{module output signal} \end{aligned}$$

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 analytical framework with **Magic**— a personalized module designed to take action as your trading companion. 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.

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.

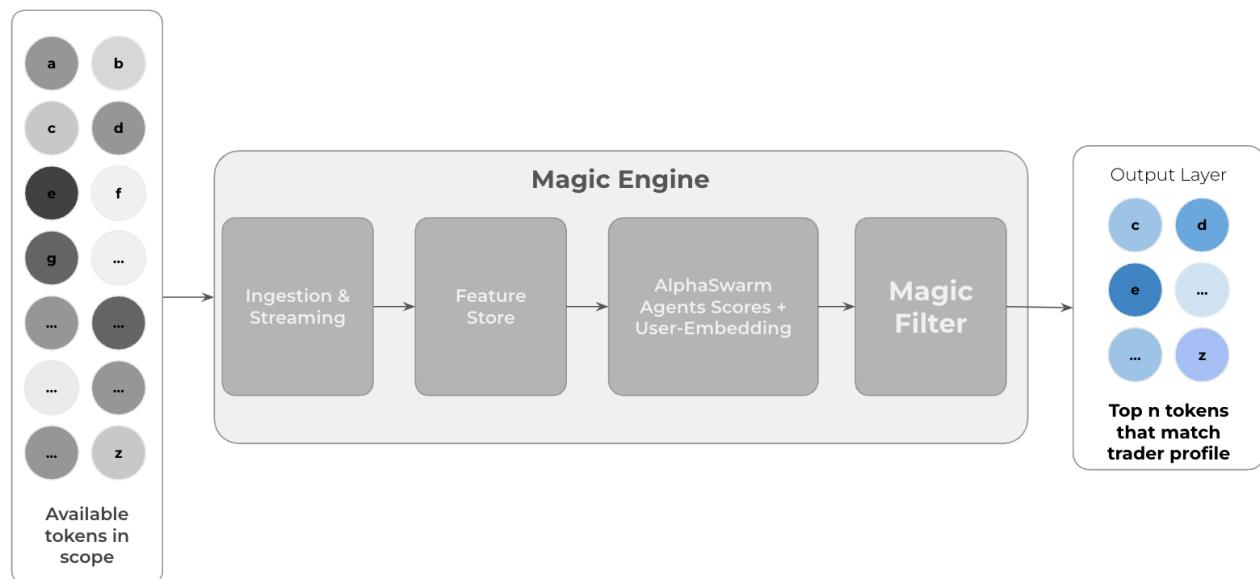


Figure 5: Dragon's Core Magic Module

The Magic engine is a multi-stage pipeline that systematically processes the set of potential tokens to be matched to a trader's profile. Initially, relevant on-chain and off-chain data is ingested and transformed into structured features, which are stored and evaluated by AlphaSwarm agents which compute signal scores and align them with the dynamic user embedding that captures individual trading behavior. The resulting scores are passed through the Magic filter, which prioritizes and selects from the list the most relevant tokens. Finally, the output layer presents the selection of top-ranked tokens that match the trader's unique profile and preferences.

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.

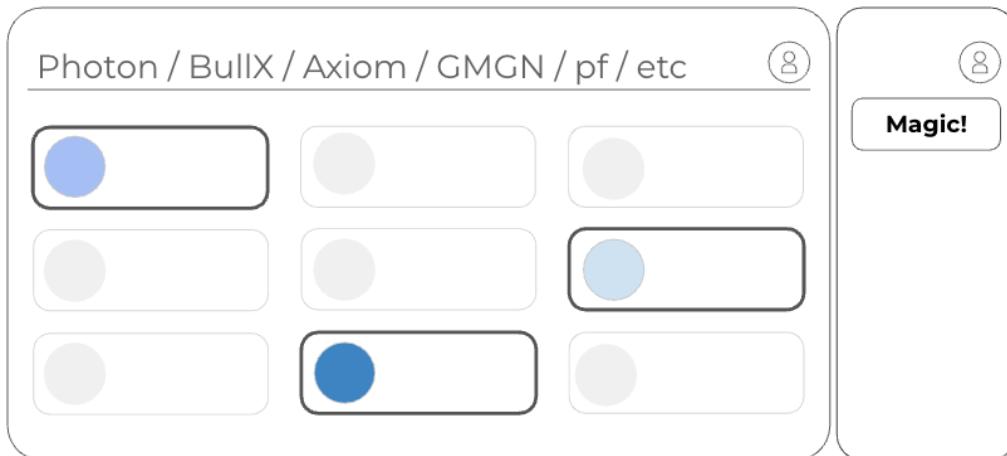


Figure 6: The Magic Button takes action to interact with your browser, eliminating all tokens on the screen that do not match your trading profile.

This integration of insights, preferences, and execution expedites the loop for intelligent trading—giving users an advantage in 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

Utility

There are three main utilities that the \$DRAGON token will provide to its holders.

1. Reward developers for contributing token analysis modules.
2. Enable user access to core AI modules (AlphaSwarm and Magic).
3. Facilitate DAO governance of project development and treasury allocation.

These three functions will be rolled out gradually as the project develops to accommodate them.

Rollout

- I. \$DRAGON is launched with 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 paid out of 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 partnership opportunity
- IV. When trading fees and project growth have stabilized, the core team builds a smart-contract that automates treasury distributions and payments
- V. When finished, the smart-contract is handed over to a DAO of \$DRAGON token holders to decentralize control of the treasury
- VI. The DAO and core team keeps building together, fully decentralized

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

6

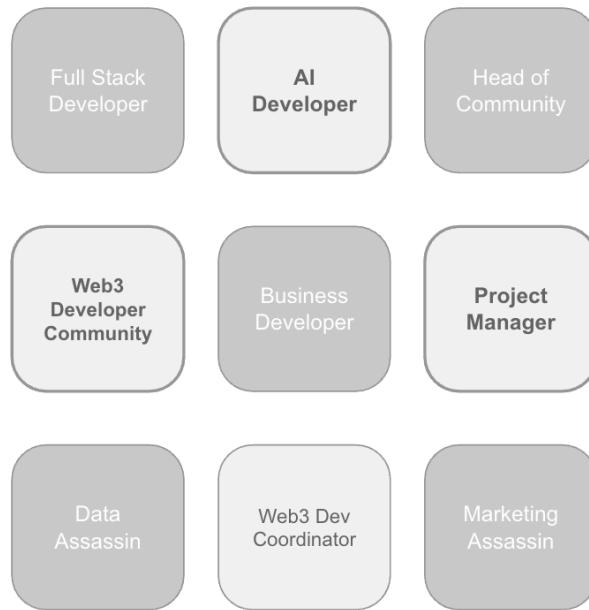
Development Goalposts

<i>June</i>	Team hired!	Wallet login enabled	
Treasury automations			
<i>July</i>	Chart redesign for first four modules	RPCs built to first four modules	
	Logo / mascot redesigned	Open-source contributor pipeline	
<i>Aug</i>	Web redesign for user's hub	Dataset created for AlphaSwarm	
	Customizable extension	User database structure finalized	
<i>Sep</i>	First AlphaSwarm training	Module Forge deployed	
<i>Oct</i>	First Magic engine training	AlphaSwarm v0 released	
	DAO infra begins		
<i>Nov</i>	Mobile App released	Magic v0 released	
	Database Integrations	ETH / BSC / ... expansions	
<i>Dec</i>	DAO takes control		

7

Project Note

This project has been launched by Harshan (Harshit) and Frog (Andrew), who are an AI Researcher and a Project Manager by previous working experience. They consider the third founding member of the project to be the Web3 community of developers. This wasn't really the original plan, but it did become a happy consequence of not having blockchain development experience between us.



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

Our goal is to hire a team of experienced specialists who can work alongside us in a full-time-esque capacity, including a Web3 developer who will coordinate the community module integrations. A foundational design of this project is that its success is a direct function of the market's need for it. If it turns out that modular and open-sourced token data is in high demand among traders, we have a perfect opportunity to publish research with user trading data as it relates to the most effective token analysis modules. The knowledge gained from this kind of

research promises valuable insights for blockchains, exchanges, and firms invested in better token design and expanding participation in Web3.

Our advantage to launching this project lies in our daily interaction with institutional grade AI research, applying the state of the science to guide our development. Harshit met Andrew in a Masters of Data Science program at the University of Maryland after coming off a research internship at IBM developing generative AI. He now works at Amazon developing their AI scaling infrastructure, where he also gleans the latest in MCP methodology. Andrew is the non-technical one, who reached out to Harshit for tutoring, and traded memecoins during the meetings. He co-founded a blockchain start-up called Cent a few years ago, which also launched a successful experiment called Valuables that tokenized tweets as NFTs on Ethereum. Andrew brings an experimental understanding to project management with his creative and international experiences working in the largest entertainment and marketing agencies known in the US. This project serves as an opportunity to combine this structured management and Web3 founder experience with a cutting edge institutional AI research exposure, and a partnership based on mutual mentoring.

<https://www.marketplace.org/story/2021/03/24/tweets-are-selling-for-millions-as-nfts-we-decided-to-see-what-all-the-buzz-is-about>

[https://openaccess.thecvf.com/content/WACV2025/papers/
Marjit DiffuseKronA A Parameter Efficient Fine-
Tuning Method for Personalized Diffusion Models WACV 2025 paper.pdf](https://openaccess.thecvf.com/content/WACV2025/papers/Marjit_DiffuseKronA_A_Parameter_Efficient_Fine-Tuning_Method_for_Personalized_Diffusion_Models_WACV_2025_paper.pdf)

We see ourselves as outsiders jumping in the deep end; retail memecoin traders that lost (a lot of) money to sniper bots and crypto-insiders. We look forward to the insights gained by stewarding this community project to allow more people to participate at the new financial frontier. We believe a key to widening access to crypto is in trading data generated by participating retail consumers, creating a new space for tailoring token design on a wider spectrum of trader profiles and experience levels.

Let's build and see how far this goes.

Links

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

Chrome Store <<https://chromewebstore.google.com/detail/dragon/ncbgllgbplhnbekllhogabdefjfdbkoe>>

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

X <<https://x.com/AlphaDragonAI>>

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

Demo <<https://vimeo.com/1062123553>>