

Dragon

Decentralizing token data for
retail analysis.

Co-created by Harshan and Frog



email: baddragonmygoodness@gmail.com

Table of Contents

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

1

Introduction

Why aren't you in control of your analyses while you research new tokens?

Dragon is a research data platform powered by community-built token analyses. Each one sources on- or off-chain data and produces knowledge about any aspect of a token project's design. Think of the best analyses from Bubblemaps, Trench Bot, Defillama, GMGN and combine them with the best public insights from KOLs, Dune analysts, and anything else you've always wanted to have in the trenches. Token analyses today are scattered and platforms have custody over their development. Traders end up paying the price when they are forced to open browser tabs and ping telegram bots or pay subscriptions just to have everything.

Dragon grants any token analysis the opportunity to be developed in a bite-size modular format, to be used and customized in a trader's browser extension that will stay with them wherever they trade. This is "DYOR" in the next era, an ability to customize your research with only the analyses you care about, never sifting through walls of unnecessary statistics again. With use over time, Dragon becomes a smart companion with AI that learns your strategic trader identity, matching tokens you've bought and seen profit on to new tokens you haven't discovered yet, based on the analyses you use and their readings when you buy.

Dragon composes a picture for retail trader preferences across the token analyses in highest demand, constituting a new definition for **Alpha** where modular data is the leading indicator in shifting volume patterns. This becomes an opportunity for the participants of Web3 to inform a smarter building of its trading platforms and design by demand token projects that adhere to an "alpha" that is entirely data-driven.

2

Market Dynamics & Value

Trading bots across all blockchains have seen over \$80B in lifetime volume¹. Solana bots alone surpassed a trading volume of \$770m in a day earlier this month². With the growth of trading bots, we've also seen increased use of token scanners like Bubblemaps, Trench Bot, and DefiLlama. On Solana, these DYOR projects total 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, thorough token research methods for traders and token design equilibrium at the infrastructure level still goes without a solution in Web3. On Solana, the chain with the most transactions and fees of all blockchains⁷, major trading platforms like Photon, BullIX, and GMGN are incorporating features that share real-time token data adjacent to the trading chart, acknowledging this concern as forefront among traders.

Maximum Extractable Value (MEV) is an integral part of blockchain development, interacting with or affecting on average ~25% of the transactions in every single block on Ethereum, Solana, and Binance Smart Chain⁸. Its impact can increase to over 66% of all block transactions during periods of peak volatility and high network traffic⁹. MEV has been called “the double edged sword¹⁰” of crypto as it is the essential incentive for searchers and validators to produce blocks, but it also creates opportunities for bad actors along 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 opportunity to capture it, but no current projects work at a consumer level and support retail trading or promote participation in the market design for it.

Web3 user data to date is all about defense - institutional compliance, threat detection, or top-down market analysis. Dune Analytics has shown us that community-developed data is in demand and Nansen has shown us that institutions are benefiting from our trading data. Dragon will go on the offensive and put data analysis at the retail level in the retail trader's control.

3

Dragon

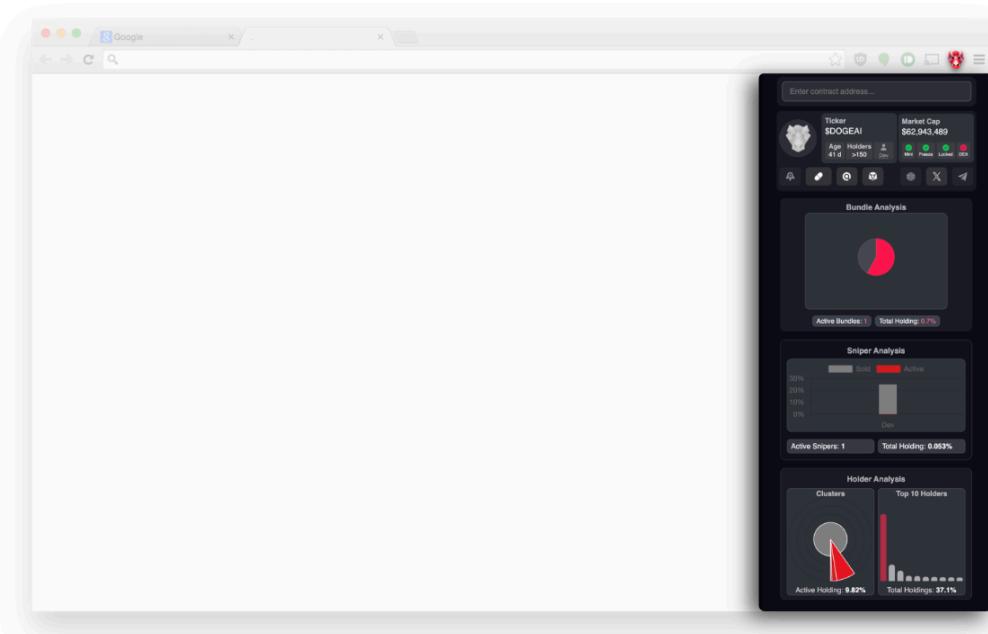


Figure 1: The Dragon browser extension stays open alongside your preferred trading platform.

Dragon's browser extension becomes the trader's companion in the trenches. Users paste a token's contract address to produce a dashboard of real-time analyses using various inputs 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, updating 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 for your trading profile. Taking advantage of only needing to use read-only methods, this approach to on-chain mixed with platform usage minimizes wallet security risks.

Modularity

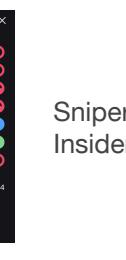
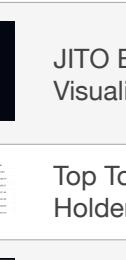
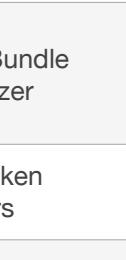
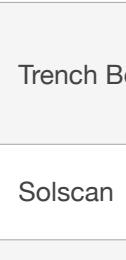
Token analyses are divided into bite-sized **data modules** that each analyze one specific aspect of a token's design or project quality. Dragon's release comes equipped with four basic prototype modules, but the goal is for a community of developers to propose and contribute new ideas from this point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



Figure 2: Token analyses are designed to be plug-and-play **data modules**—each focusing on a single token design metric—that users can mix, match, or even build themselves.

Module Inspiration

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

Example	Token Analysis	Current Custodian
	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

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 validate them against historical 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.

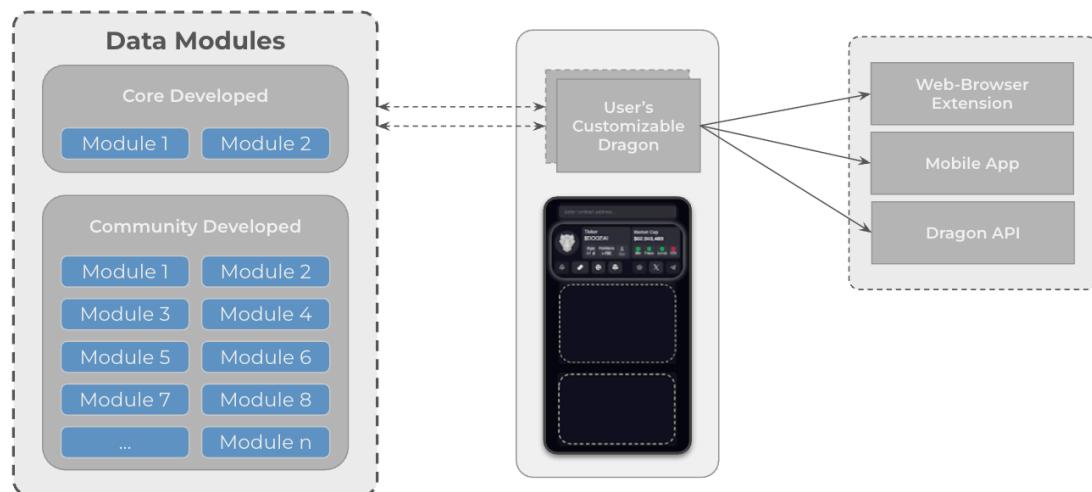


Figure 3: Dragon's architecture includes both core team and community developed modules into a product that syncs across the user's browser extension, mobile app, and public API.

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 orchestrated approach powered by selective LLM training that captures **Alpha** with users' selected modules, tracking on- and off-chain token data analytics.

Introduction

AlphaSwarm is a modular, multi-agent observatory designed to forecast token movements by fusing different modules making on-chain events (DEX volume, wallet flows, MEV activity) and off-chain signals (social sentiment, news, alpha groups) into actionable 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 Scoring System

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

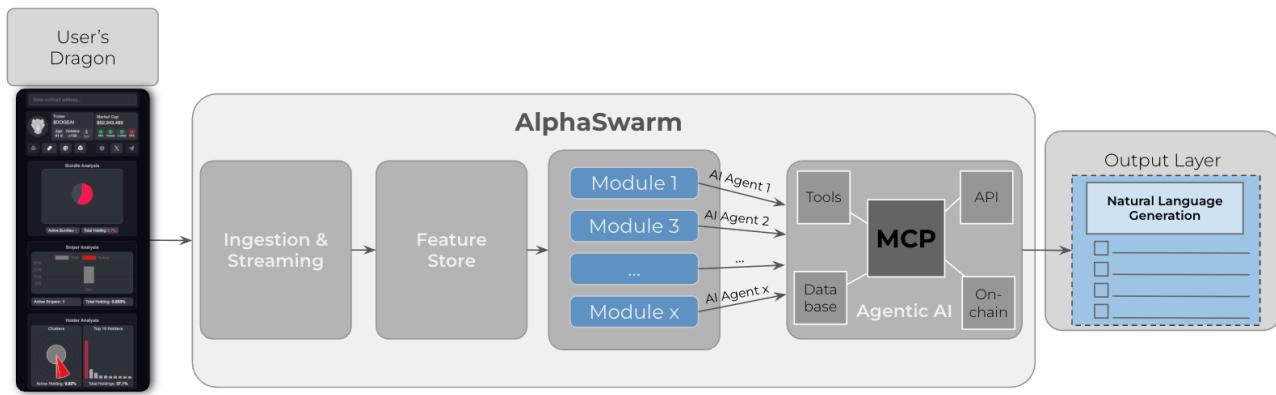


Figure 4: Dragon's Core AlphaSwarm Module - [insert summary on flow above]

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.

<https://www.anthropic.com/news/model-context-protocol>

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 trade suggestions based on personalized trading patterns.

A Personalized Magic Module

Dragon enhances AlphaSwarm's analytical framework with **Magic**— a personalized module designed to act as your trading companion. Every time you interact with any other data module or make an on-chain trade, your behavior is captured and embedded into a dynamic user profile. This profile encodes not just

your preferences, but your actual trading behavior—the risk levels, liquidity patterns, and market sentiment signals at which you engage with tokens at the most—forming a “behavioral fingerprint” of your trading strategy profile.

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 or app. The second tower encodes AlphaSwarm’s fused analytics that have been passed through the trust-weighted MCP. This means every token embedding already reflects its most relevant AlphaSwarm-derived signals.

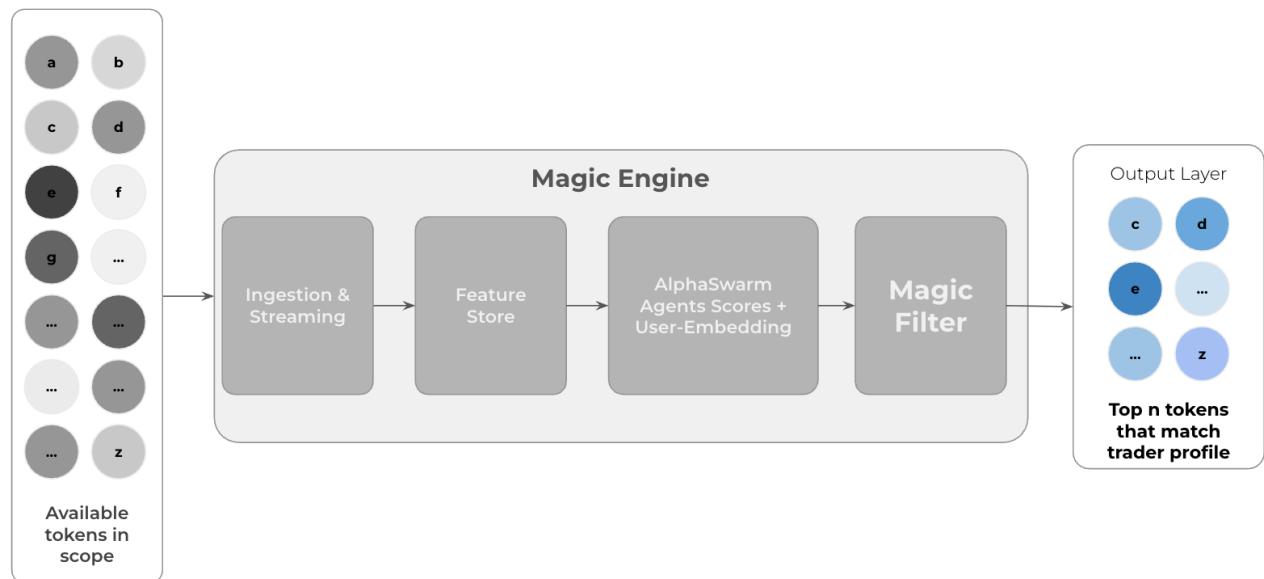


Figure 5: Dragon’s Core Magic Module - [insert summary on flow above]

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 your behavioral fingerprint and AlphaSwarm's signal fusion, the **Magic Button** offers a one-click bridge to execution. With a single tap, the Magic Button intelligently filters out non-tradable tokens based on your selected platform (e.g., DexScreener, pump.fun), and preloads parameters. In effect, AlphaSwarm fuels the intelligence, Magic delivers the personalization, and the Magic Button enables the action.

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

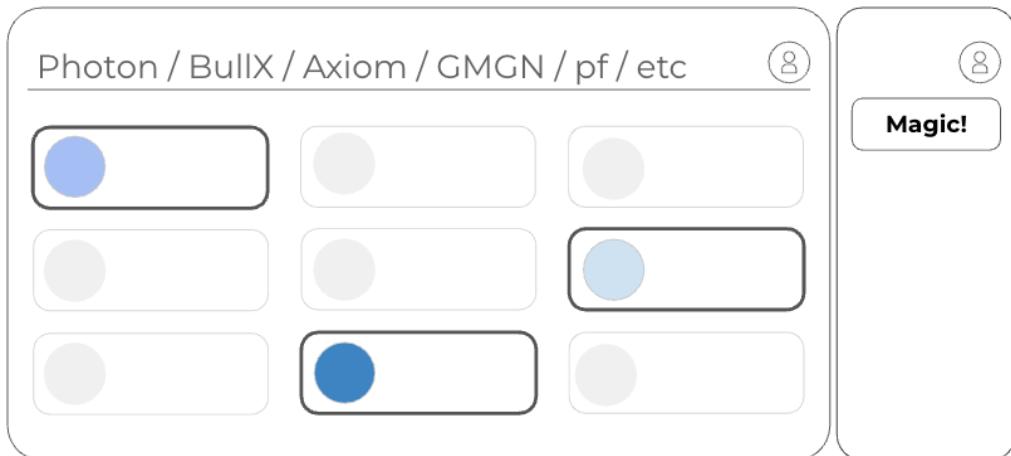


Figure 6: The Magic Button eventually takes Magic Engine to interact with your browser experience, eliminating any tokens on the screen that do not fit in your trading profile.

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.

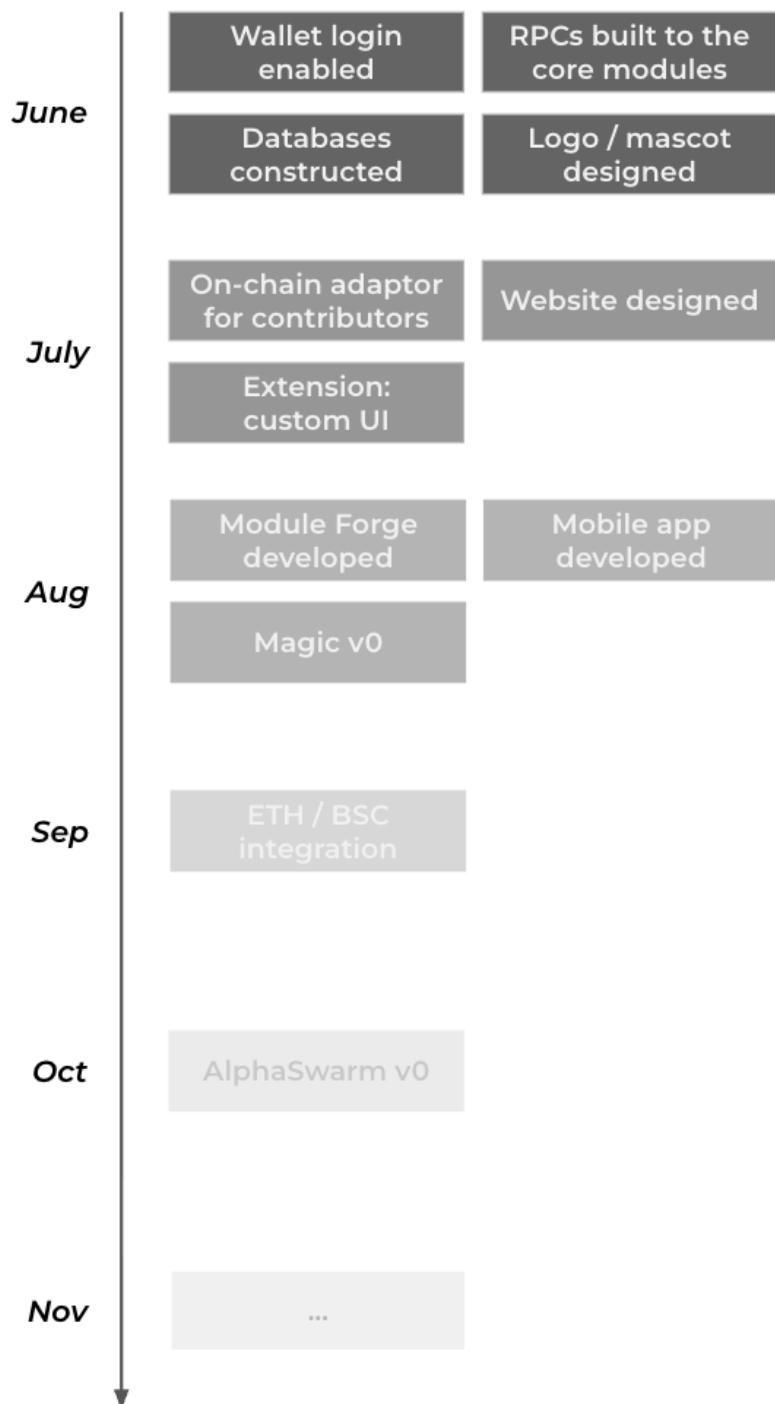
Rollout

- 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

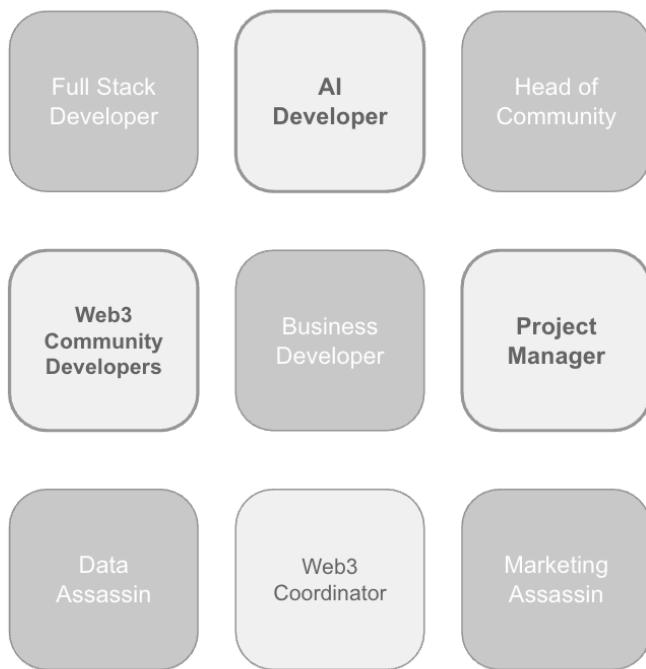
Development Timeline



7

Project Note

This project has been co-created by Harshan ([Harshit](#)) and Frog ([Andrew](#)), who are an AI Researcher and a Project Manager by previous working experience. The third founding member of this project is *the entire Web3 developer community*. While this team structure wasn't our original intent, it became a happy consequence of neither of us having experience building on the blockchain.



The plan is to hire a group of experienced assassins who will work in a full-time-esque capacity alongside us, including an experienced Web3 developer who can coordinate community module integrations. This project's development is in the hands of the people of Web3's demand for it. If it turns out that the demand for modular and open-sourced data analyses is high, we'll use the opportunity to produce research on the trading data and the most popular token analysis modules. This research promises a high value for blockchains, exchanges, and firms invested in better token design for blockchain participants.

Although we do have ample experience running Web3 start-ups¹ and participating in leading AI research¹ before this, we still see ourselves as outsiders jumping in; retail memecoin traders that lost (a lot of) money to trading bots and crypto-native insiders. The insights learned from stewarding this project will allow more people to enter crypto and exit the traditional finance system, as retail-specific trading data will allow for tailored token design on a wider spectrum of trader strategies and experience levels.

Let's build this thing and see how far it goes.

Links

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

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

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

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

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

Linktree <linktr.ee/alphadragon>

Vision <<https://dragon-12.gitbook.io/alpha-dragon>>

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