$Alpha Dragon

Decentralizing token data to discover alpha.

Co-created by Harshan and Frog

Last updated: June 25, 2025

email: baddragonmygoodness@gmail.com

Table of Contents

1 Introduction

2 Market Dynamics & Opportunity

3 Dragon

4 AlphaSwarm, Magic Engine & Alpha Engine

5 Project Token

6 Development Goalposts

7 Founders' Note & How to Contribute

1

Introduction

Every trader can discover alpha as composable data sourced by open intelligence.

Dragon is an open project powered by community-built token analyses. Each one 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 combine them with the best insights from KOLs, Dune analysts, and insider group chats. Meaningful token data are scattered on these platforms or gated by the chads with custody of them. Retail traders lose by managing open browser tabs or pinging telegram bots or paying subscription fees just to get to some of the information they need to trade with intelligence. This is the stacked game of getting to **alpha**—but nobody has tried the approach of methodically discovering volume signals in the composable data.

Dragon gives analysts an opportunity to contribute token analyses that they believe will signal volume changes, by building bite-sized **data modules** that integrate into an open browser extension. Each Dragon user's extension is a customizable panel of these analyses, allowing traders to research tokens in the way they believe is most effective. Modules that see high usage will receive agentic AI training in order to learn how their output relates to volume changes across token data in the ecosystem. Dragon also monitors user trades on-chain in relation to their module use, developing strategic trader profiles that personalize their trading companions.

As the project grows, we discover where the alpha-signals are in the token analyses through the collective intelligence of the builders and the data from the traders. This platform builds retail's contribution layer on the blockchain, enabling everyday consumers to impact future token design and development.

2

Market Dynamics & Opportunity

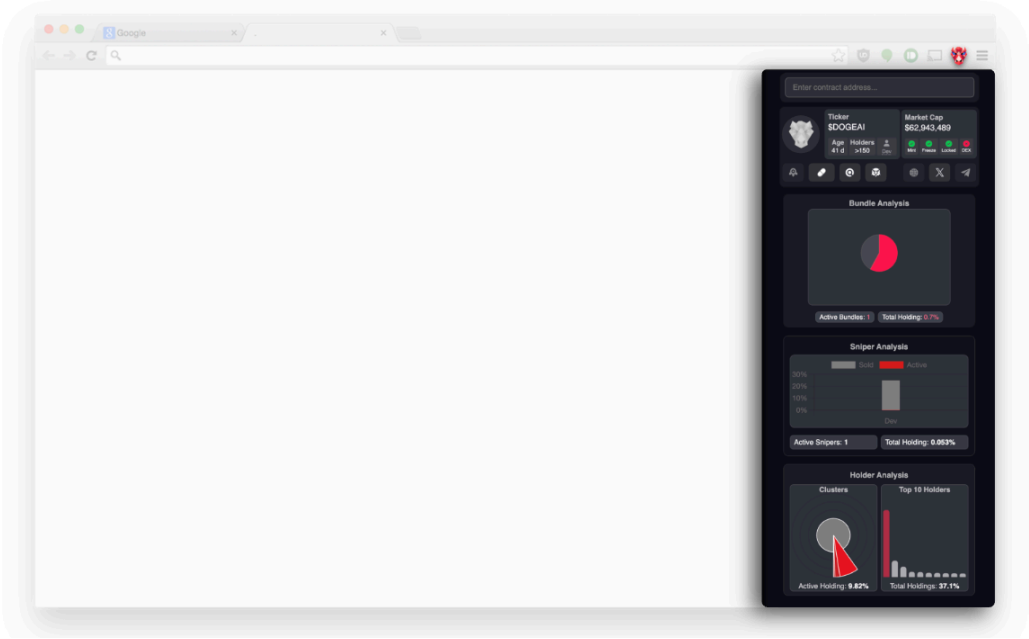
Trading bots across blockchains have seen over $80B in lifetime volume[1] and bot trading alone on Solana surpassed a volume of $770m in one day earlier this May[2]. As the chain with the highest number of total network transactions[3], Solana's users have found value in a new category of tools called **token scanners**, like Syrax, SarumAI, and trench.bot. These DYOR assistants currently see ~500k monthly active users with rugcheck.xyz in the lead[4]. Even Solana's trading platforms Photon, BullX, and Axiom are featuring token scanning data adjacent to the charts, acknowledging that data transparency is a top of mind concern for consumers. Due to recent high profile scandals[5], extractive pump.fun launches[6], and even obscure behaviors at institutional levels[7], token data demands a better standard for effectiveness and accuracy in retail trading.

One of the misunderstood market dynamics of blockchain is **Maximum Extractable Value (MEV)**, a feature that affects ~25% of all transactions on Ethereum, Solana, and Binance Smart Chain[8]. During periods of peak volatility and high network traffic, MEV can even surge to impact over 66% of all block transactions[9]. It has been called the double-edged sword of crypto[10] because it incentivizes validators and drives block production, but also creates information asymmetries that favor insiders over everyday traders. Infrastructure-level projects like Flashbots have emerged to illuminate MEV on Ethereum, but there is no equivalent awareness built for the retail layer; no reliable way for traders to analyze how MEV impacts volume on the tokens they trade daily.

So far, most **Web3 data initiatives** have focused on building defenses for the industry—like institutional compliance, threat detection, or top-down market reporting. The few retail-facing analytics tools lock their insights and lack the completeness to adequately inform daily trader decisions. Yet the market has proven an appetite. Dune Analytics validates the demand for community-driven blockchain data and Nansen proves that trading intelligence is valuable when it's made available to the public. This project aims to combine the two into one community-driven platform made by traders, for traders.

3

Dragon



Dragon's extension stays open inside your browser, going with you as you navigate the internet, trade tokens, or conduct research.

To use Dragon's browser extension, simply paste in a Solana token contract address. The panel becomes a dashboard of real-time information on that token using analyses drawn from on- and off-chain data. These analyses update in real-time and follow a user as they browse the internet, becoming an AI companion that goes with you where you research.

Dragon uses wallet-based single sign-on (SSO) to sync settings across devices and track trading behavior. When a user purchases a token, Dragon **snapshots** the analyses active on their extension. Over time, this builds a trader's profile and shapes their strategic identity in order to match them to tokens they haven't yet discovered. Using read-only methods, this keeps wallet security risks low while generating rich contextual data for users.

Modularity

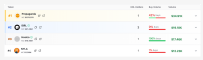
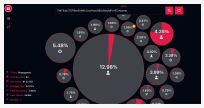
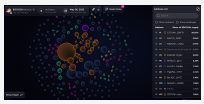
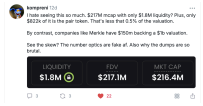
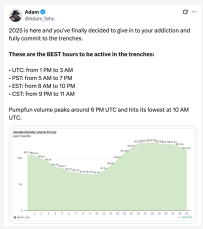
Dragon's key design feature is its modular architecture. Each token analysis is packaged as a bite-sized **data module** that evaluates one specific aspect of a token project, like liquidity flow, smart money, team wallet behaviors, developer history—the list goes on.

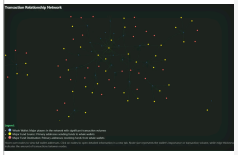
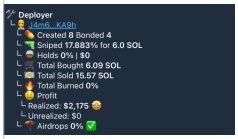
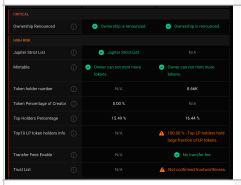
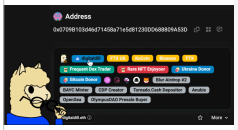
Dragon launches with four prototype modules that scrape data from third-party sources. These four modules are currently bountied for a Web3 developer to connect them to RPCs. Future modules can source off-chain data as well as on-chain, because alpha signals can come from anywhere. The goal is an entirely community-generated intelligence platform where anyone contributes a module based on an idea to detect alpha.

*Token analyses are packaged as **data modules**—each focusing on one specific aspect of a token's design. Users contribute modules that they believe will signal changes in token volume.*

Inspiration

Valuable token analyses today are scattered among key individuals and platforms. Dragon invites these data custodians into the shared research community where their insights can reach broader audiences, market their other services, and remain accessible to the public for continuous improvement. With an open platform, we create incentives to compete for better, faster, and more creative approaches in developing token analyses.

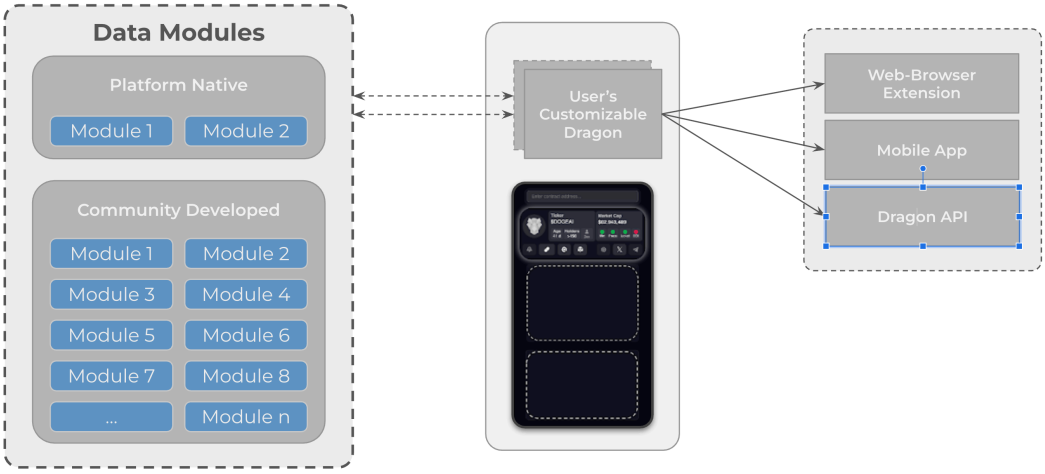
Inspiration	Example Analysis	Current Custodian
	KOL / smart money holders	StalkChain
	Sniper & insider buy counter	GMGN
	JITO bundle visualizer	trench.bot
	Top 10 holder awareness	Solscan
	Clustered wallets	Bubblemaps
	Liquidity - market cap ratio	@kompreni
	CT / whale holder % and movement	@dethective
	TikTok / X sentiment matching	@PixOnChain
	Time of day notifications	@AdamTehc

Inspiration	Example Analysis	Current Custodian
	Token narrative profiling	Dexu / @cryptokoryo
	Liquidity pool sandwich score	sandwiched.me
	Top holder previous transaction behavior	RugChecker
	Dev / deployer overview	Trenchy Bot
	Holder trends by time held	HolderScan
	Essential security heuristics	Birdeye (and others)
	Holder profiles / style tags	DefiLlama
	DEX paid? Project links and aesthetics	DEX Screener
...		

Development

At launch, developers propose new modules via pull requests on GitHub following these guidelines. Soon there will be a **Module Forge**—a web-based drag-and-drop sandbox where developers (including non-JavaScript-native users) can build modules in Python or Rust, compile them to WebAssembly (WASM), and test them against token data in real time.

A complementary **Module Market** will allow users to upvote or downvote these analyses, request and bounty new modules for development, and toggle modules on/off instantly to their own Dragon. This creates a fully composable and permissionless platform where the community's demand drives development and discovery via effective token analyses.



Dragon includes a few platform native modules and many community developed ones. The end product syncs across a user's browser extension, mobile app, and a public API.

Modules will be ranked by their weighted impact in signaling changes in token volume, as well as adoption across the user base. High-impact and high-visibility developers gain more influence in the project's direction through project token accumulation, aligning an incentive between module quality and community value. Modules ranked higher will also receive the most focus in data collection and AI model training. At launch, developers will receive token bounties for their contributions, but the goal is to stream revenue to the developers with the highest impact on the project. More information can be found on that in the Revenue section below.

Mobile Application

Dragon's mobile app mirrors the extension, giving users access to their modules, dashboards, and alerts on the go. Push notifications can be configured to trigger when selected modules detect major shifts or volume anomalies.

Because Dragon tracks usage patterns across both desktop and mobile, its personalization function (powered by the Magic engine) adjusts for behavior deltas between the devices, refining token match opportunities to users wherever they are actively trading.

4

AlphaSwarm, Magic Engine & Alpha Engine

An orchestration agentic approach to selective pattern-recognition training that captures Web3 alpha with composable analyses, putting trading data to volume changes.

Introduction to AlphaSwarm

Unlike traditional on-chain scanners that only surface static metrics, **AlphaSwarm** is a multi-agent architecture that generates scenario-based scores paired with clear, human-readable explanations—so you can instantly spot genuine market anomalies rather than scan through raw data. As a modular ensemble observatory, it fuses on-chain signals (DEX volume, wallet flows, MEV activity) with off-chain indicators (social sentiment, news, alpha-group chatter) to deliver truly 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 orchestrator** (pattern recognition) that holistically associates patterns, 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 Training on Meaningful Windows

Rather than training a 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 framework. Moreover, all event-window training is conducted in an offline pipeline—so we avoid the complexity, latency and cost of continual LLM fine-tuning in production—while still ensuring our patterns stay up to date through scheduled batch retrains.

Each data module in Dragon is handled by a dedicated agent that monitors and logs its respective features over time. A numeric pattern recognizer 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 model captures multi-dimensional patterns that characterize specific market scenarios. The LLM summarizes these for the user.

During inference, when new data arrives, the orchestrator computes a similarity score to past events using the numeric model. The LLM then explains why a given situation may be unfolding, offering a more robust and context-aware form of interactive 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 hub which enable agents to validate, enrich, or challenge each other's findings.

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 orchestrator learns to recognize that pattern. When similar conditions reappear in new data, it surfaces the match—not as a hard statistical claim, but as a scenario match based on prior outcomes. In this way, AlphaSwarm resolves confounding using historical pattern coordination and an interactive analytics tier.

Qualitative & Quantitative Output Layer

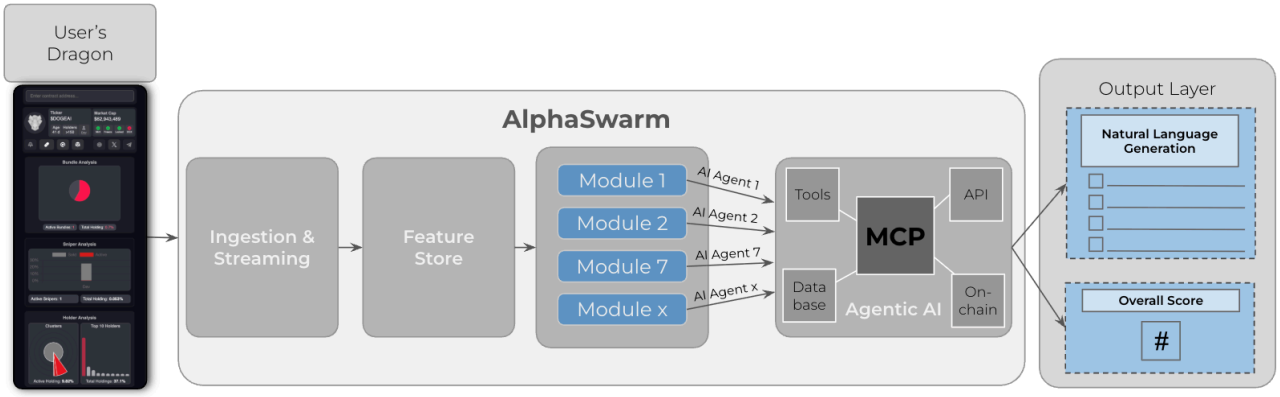
AlphaSwarm delivers a scenario-based scoring mechanism ranging from 0 to 10. A score of 0 indicates that current conditions bear no resemblance to known high-volatility scenarios from the training data, whereas a 10 reflects a strong historical match to a past pattern with significant market impact. 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 ("AI-Hub" = numeric + NLG) 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 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.



Dragon's AlphaSwarm Module

Data streams are ingested through a high-throughput infrastructure using AWS Kinesis or Apache Kafka, which organize signals into time-aligned buffers. These are written into a dual-layer feature store: a hot layer (e.g., Redis or KeyDB) for sub-second latency access using compacted 32–64 byte features, and a cold layer (e.g., Apache Iceberg on S3) for long-term 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-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.

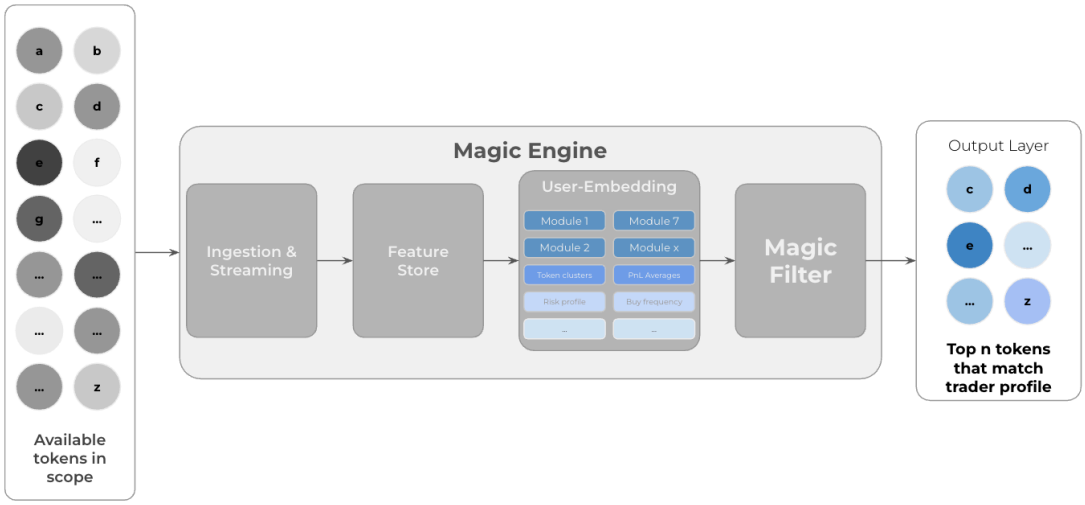
In Dragon, the MCP is a coordination layer that dynamically assigns trust scores to modules based on how well their signals align with actual volume changes. It acts like an ensemble controller—boosting modules with strong predictive value. The MCP serves as a shared trust layer, dynamically updating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

H = *total hype score*
Σ_i = *sum over modules*
w_i = *module trust weight*
$signal_i$ = *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 will be 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, market sentiment and other 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.



Dragon's Magic Engine

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, ensuring that your new tokens are personalized and responsive to changing market dynamics.

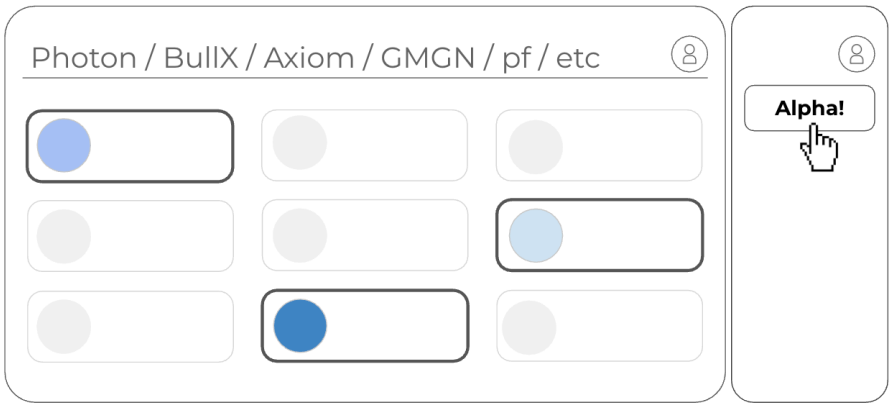
For fast and scalable retrieval, Dragon indexes all token embeddings using FAISS's HNSW (Hierarchical Navigable Small World) algorithm, enabling near-instantaneous nearest-neighbor lookups. Your personal embedding is updated in real-time using lightweight online learning steps and consolidated with a 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.

Magic Engine + AlphaSwarm = Alpha Engine

The **Alpha Engine** is the advanced iteration of Magic Engine, enhanced by the scoring infrastructure of AlphaSwarm. Its architecture fuses two complementary towers. The first tower Magic Engine ingests your behavioral fingerprint from usage patterns, on-chain activity, risk profile, and buy frequency. The second tower encodes AlphaSwarm's fused analytics that have been passed through the trust-weighted MCP. This means every token embedding also now reflects the best-matching AlphaSwarm-derived signals.

The **Alpha Button** will then interact with a trader's browser to rank-match the tokens that maximize both personal fit and alpha potential. This one-click simplicity serves as the execution bridge—instantly filtering non-matching tokens across platforms like DEX Screener or pump.fun, preloading trade parameters, and converting AlphaSwarm's market intelligence into a personalized, actionable feed. By integrating insights, preferences, and

execution, the Alpha Engine and Alpha Button accelerate the trading loop and gives users a distinct edge in fast-moving token cycles.



Alpha Button works with your browser (trading platforms) to remove any tokens on the screen that don't fit your trading style or meet AlphaSwarm's signal standards.

Parallel AI Features

Dragon will eventually include a built-in conversational chatbot—your on-demand trading assistant embedded in the extension and mobile app. This chatbot retrieves the most relevant docs (Wikipedia, module snapshots, on-chain traces and real-time agentic analytics) to ground its answers. 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

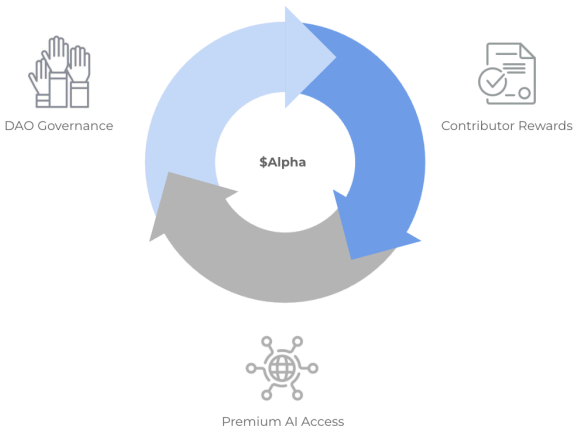
There are three major utilities that the $ALPHA token will provide.

1. Rewards developers and other collaborators for project contributions, ie. high performing data modules or high impact digital content.

2. Enables a user's access to Dragon's premium AI modules, ie. Alpha Engine.

3. Facilitates a DAO's governance of the project's development and revenue allocations.

These three functions are gradually rolled out as the project and community develops.



Rollout

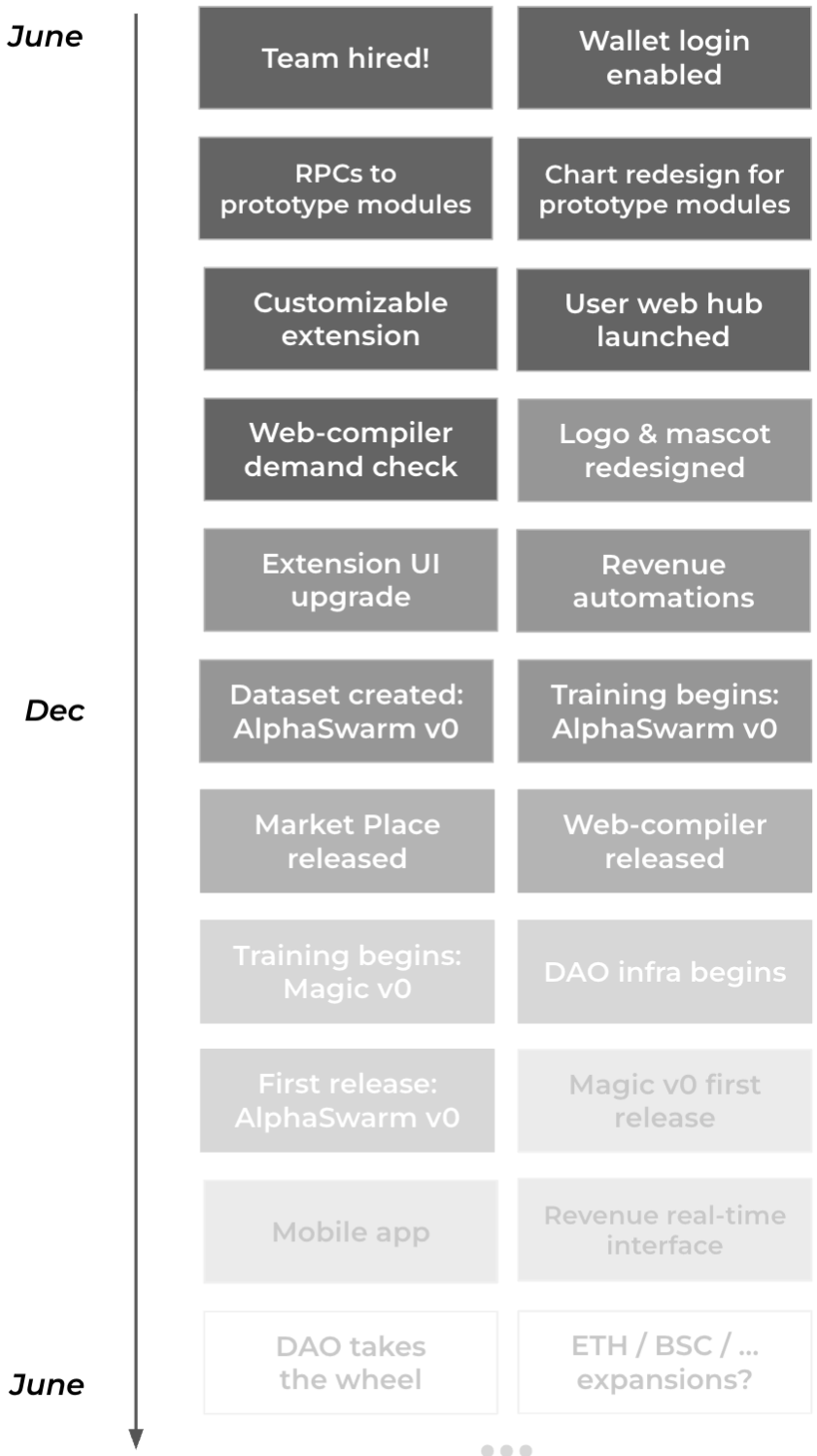
I. $ALPHA pre-sales commence and investor participation is pursued. The final allocations are set, a TGE date is planned, and the new token is launched to the public.

II. Project contributors are incentivized with $ALPHA on a bounty system coordinated by the launch team. This sparks the initial community participation at the early stages of project development.

III. As community and development normalize, the launch team implements a DAO using the $ALPHA token as the governance mechanism. The DAO gains control of the contributor reward system and the future direction of project development.

6

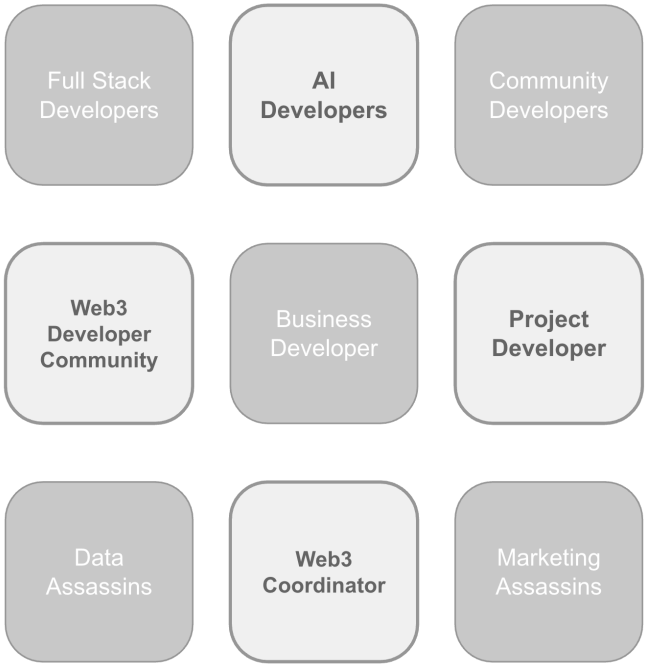
Development Goalposts



7

Founders' Note

This project is launched by Harshan (Harshit) and Frog (Andrew), an AI developer and a project developer by background. They consider the third founding member to be the oncoming community of Web3 developers. In early stages, it was a problem that neither had the skills to build on the blockchain. Now it's become a project design strength.



The goal is to hire a support team for the community, including a blockchain developer that can integrate modules and coordinate data capture with the AI team. If it turns out that modular token data becomes popular for retail, we are prepared to publish research with our findings as related to trades, token volumes, and the analyses that are useful in signaling alpha. The insights gained from this research will create value for blockchains, exchanges, and firms invested in better token design and intelligent expansion of Web3.

Our founder advantage lies in daily exposure to institutional AI research. Harshit and Andrew met in Graduate School for Data Science after Harshit had finished research at

IBM co-developing their in-house generative AI systems. Andrew approached Harshit for tutoring, but ended up trading memecoins instead of learning the material. Before that program, he co-founded the blockchain startup Cent, which raised seed funding after their Valuables project gained attention for tokenizing tweets as NFTs.

Now Harshit is working at Amazon, building scalable infrastructure for Bedrock AI and developing implementations for MCP methods. Andrew is busy designing and developing this project's early stages, as he sees the massive potential for this initiative at the financial frontier. Both of us see ourselves as shrimps jumping in the deep end of Web3; retail memecoin traders that traded money up to sniper bots, whales, and insiders. We look forward to developing a community that shares in our vision that **every trader can discover alpha as composable data sourced by open intelligence.**

How to Contribute

If you like the idea of open-sourcing alpha as composable data, we invite you to participate in the project at the ground floor. Here's how:

Build a module
Submit your idea to GitHub for an on- or off-chain analysis that surfaces real-time token insights. This can be written in Python, Rust, or JavaScript. Bounties in $ALPHA are awarded for modules accepted to the platform. Please see this guideline for how the contribution process works.

Join the community
Connect with us and other contributors in the Telegram to share ideas, plan development, and discuss upcoming features and priorities.

Collaborate on research
We're looking to collaborate with researchers on topics like token behavior modeling, retail trading patterns, and design ideas for high-impact analyses. Reach out to Andrew on Telegram with ideas.

Join the team
We're looking for a few experienced assassins, especially a Web3 dev to lead module coordination, data pipelining, and platform developments. Send us an email or send Andrew a message on Telegram if you know someone. More positions to be posted on X soon.

Project Links

Website

https://alpha-dragon.ai/

Extension (Prototype)

https://chromewebstore.google.com/detail/dragon/ncbgllgbplhnbekllhogabdefjidbkoe

GitHub

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

X / Twitter

https://x.com/AlphaDragonAI

Telegram

https://t.me/alphadragontg

Launch Video

https://vimeo.com/1091491580/5491d65057