

What is....



# Moonsight





Moonsight is an advanced AI-powered Telegram bot designed to scout early opportunities in the cryptocurrency market.

**By leveraging artificial intelligence, blockchain analytics, and real-time data processing, Moonsight identifies promising cryptocurrency tokens before they gain mainstream attention.**



# Core Features



## Pre-pump Detection

Flags tokens with accelerating on-chain activity (volume, transactions) and emerging whale accumulation.

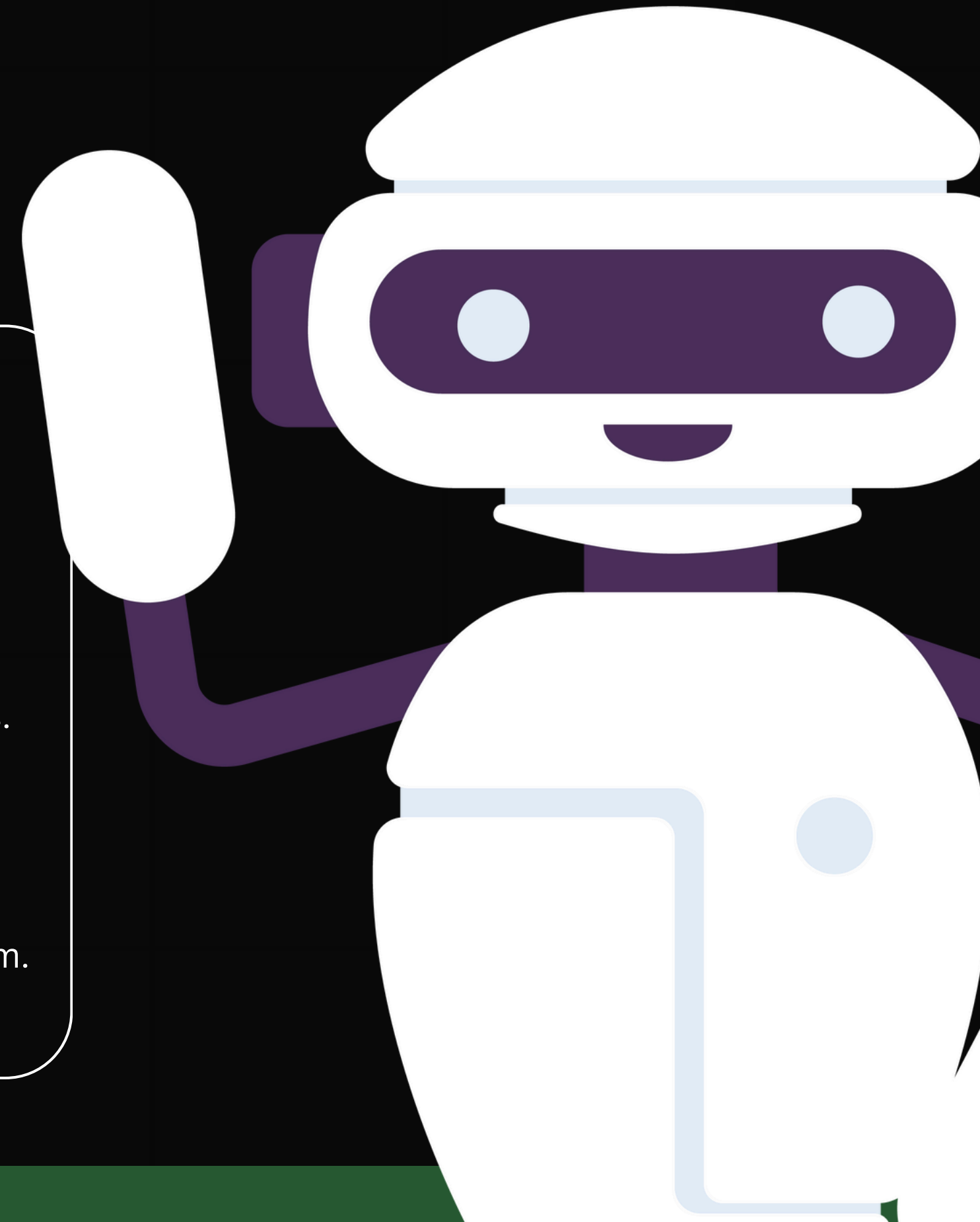
## Multi-Source Data Aggregation

Collects valuable information both via socials/sentiment, and on-chain (Glassnode, DexScreener, Dune...)

## Alpha Leak Prevention

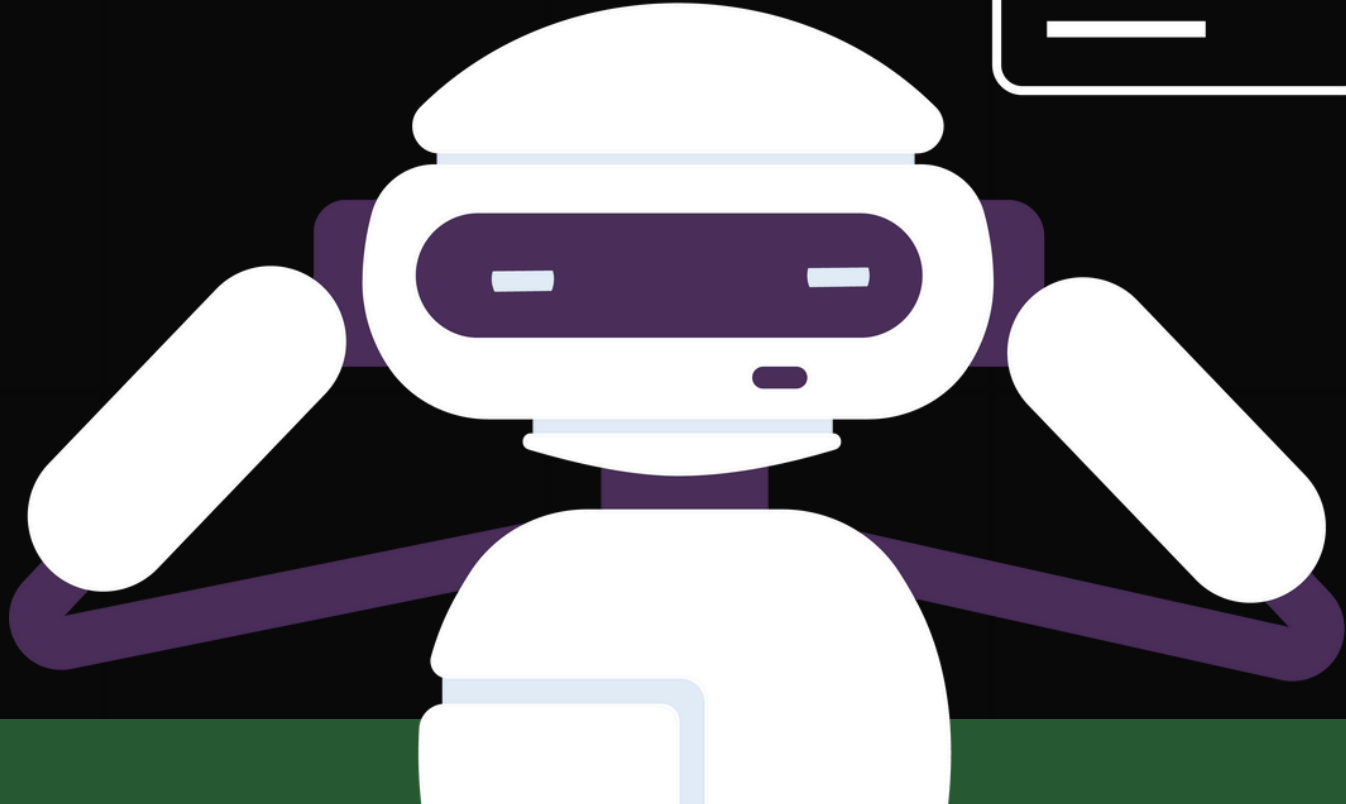
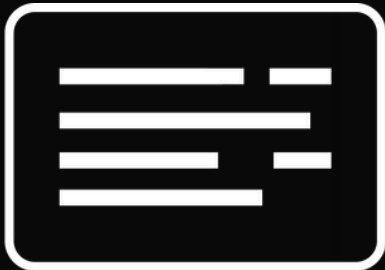
Unlike scanners that alert after pumps, Moonsight uses:

- Predictive modeling (e.g., regression on holder growth vs. price).
- Cross-chain support (Solana + EVM via RPC nodes).
- Low-Latency Alerts: <5s delay from on-chain event → Telegram.





# Key Indicators Tracked



	METRICS	SIGNAL RELEVANCE
WHALE ACTIVITY	Large wallet inflows, OTC deals	Early accumulation phase detection
ON-CHAIN VOLUME	Uniswap/DEX volume spikes, tx count	Rising trader interest
HOLDER COUNT	Top holders %, new unique holders	Risk of manipulation vs. organic growth
SOCIAL SENTIMENT	Twitter mentions, Telegram member growth	Retail FOMO potential



# Technical Architecture



## Data Ingestion Layer

```
import asyncio
from datetime import datetime, timedelta

class OnChainFetcher:
    GLASSNODE_ENDPOINT = "https://api.glassnode.com/v1/metrics/market/price_usd_close"

    def __init__(self, api_keys):
        self.api_keys = api_keys
        self.session = aiohttp.ClientSession()

    async def fetch_glassnode(self, metric, asset, since):
        params = {
            'api_key': self.api_keys['glassnode'],
            'asset': asset,
            'since': int((datetime.now() - timedelta(days=since)).timestamp()),
            'interval': '24h'
        }
        async with self.session.get(self.GLASSNODE_ENDPOINT, params=params) as resp:
            return await resp.json()
```

## Real Time Monitoring System

```
class TokenMonitor:
    def __init__(self, fetcher, detector):
        self.fetcher = fetcher
        self.detector = detector
        self.watched_tokens = {}

    async def watch_token(self, contract_address):
        while True:
            # Get live data from multiple sources
            txs = await self.fetcher.fetch_etherscan(contract_address)
            volume = await self.fetcher.fetch_dexscreener(contract_address)

            # Run detection algorithms
            whales = self.detector.detect_whales(txs)
            volume_alert = self.detector.volume_anomaly(
                self.watched_tokens[contract_address]['volume_history'],
                volume['24h']
            )

            if whales or volume_alert:
                await self.trigger_alerts(contract_address, {
                    'whales': len(whales),
                    'volume_change': volume['24h']
                })

            await asyncio.sleep(60) # Check every minute
```

## Data Quality

Our technology ensures only the best signals are forwarded to the bot. With an extremely accurate filtering.

## How It Works

When Moonsight's AI identifies a high-probability opportunity, it instantly delivers a detailed alert through Telegram. These alerts include:

- Token identification and contract address
- Key metrics summary
- Potential risk factors
- Market context and indicators
- Detection confidence score

## User-Centric Design

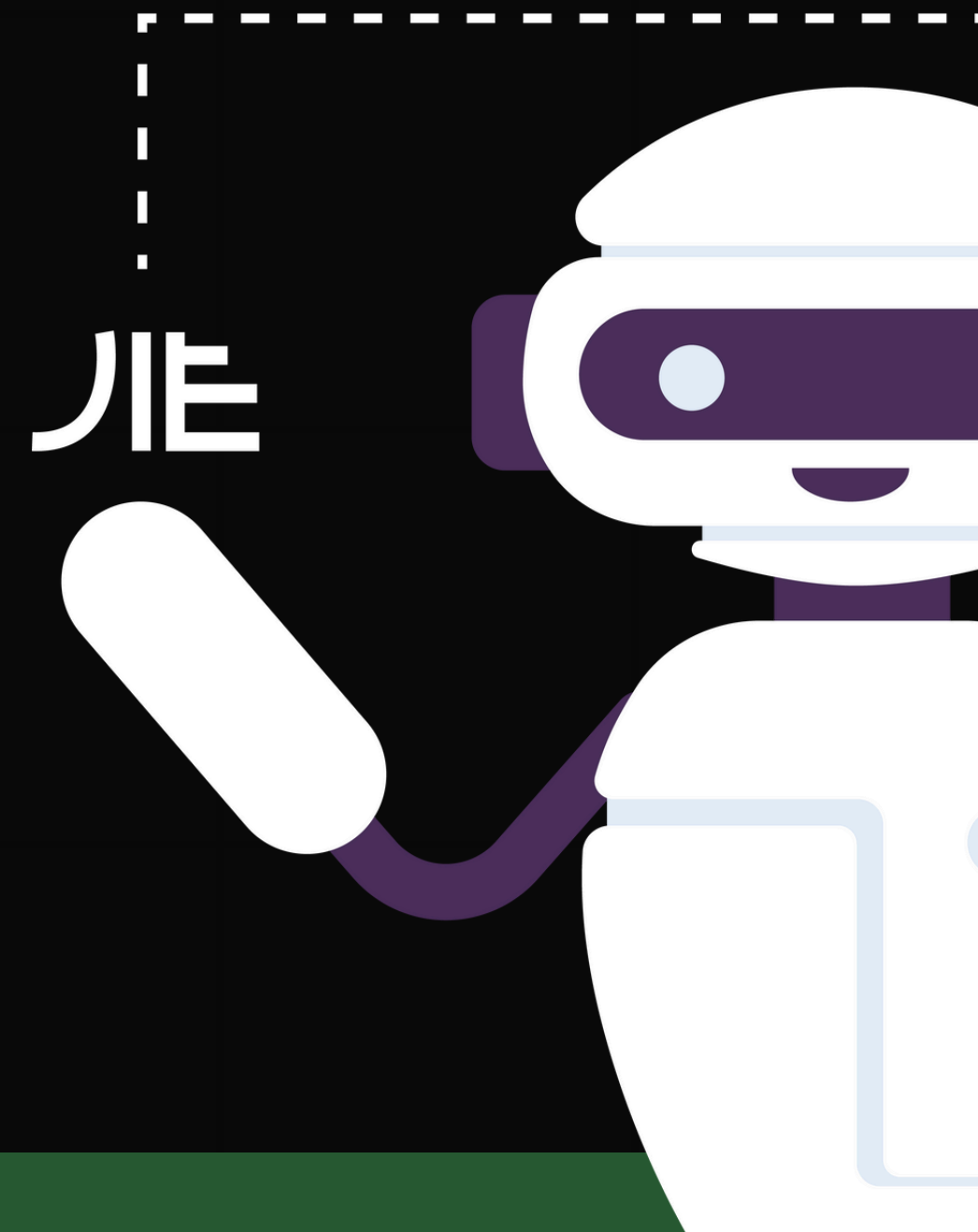
Moonsight is designed for both novice and experienced crypto users, with customizable alert parameters to match individual risk profiles, investment interests, and strategy preferences. The intuitive interface requires minimal setup while offering advanced customization for experienced users.



# Real-Time Telegram Alerts

- Risk Tolerance Settings: Adjust alert thresholds based on personal risk appetite
- Social Sentiment Focus: Customizable settings for monitoring market sentiment
- Alert Frequency Control: Manage the volume and timing of notifications
- Market Context Integration: Alerts include broader market conditions correlated assets

# Customization Options



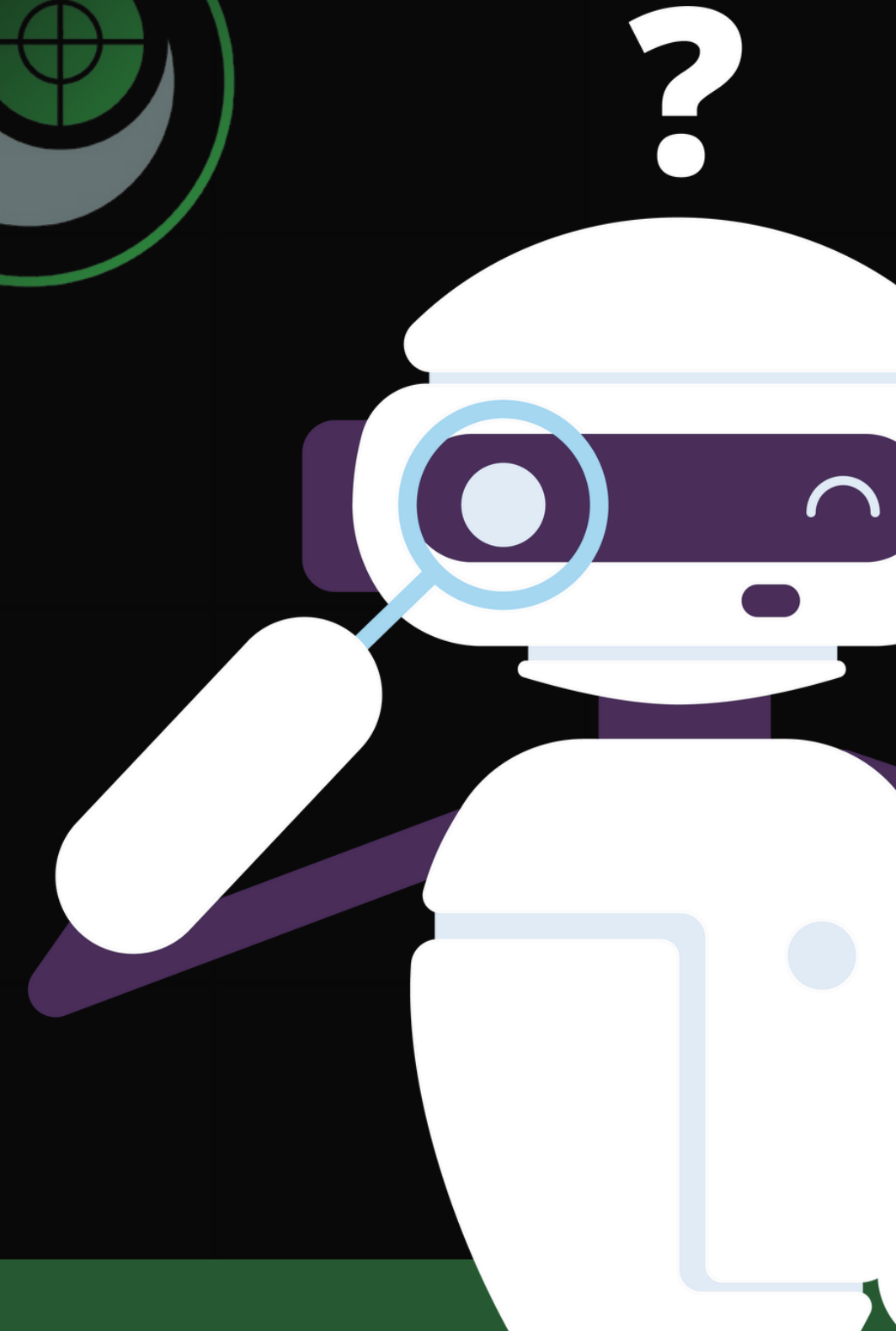
# Moonsight Tokenomics

## Token Distribution

- 40% - Public distribution (IDO, exchange listings)
- 20% - Platform development and operations
- 15% - Marketing and user acquisition
- 15% - Community rewards and incentives
- 10% - Strategic partnerships

## Value Accrual Mechanisms

- Subscription Burns: A percentage of subscription fees collected in \$MS is permanently burned
- Feature Unlock Staking: Tokens staked for feature access are locked, reducing circulating supply
- Performance Fees: A small percentage of verified profits (opt-in) is used to buy back and burn \$MS tokens







**Thank you**

