# PowerPool

A decentralized automation network for Defi

BUIDL. AUTOMATE!

# Hey! We're building automation since 2020

- **PowerPool:** initially launched as a protocol/DAO creating smart baskets of tokens ('indices').
- **Baskets proposers/gov participants:** initially launched as a protocol/DAO creating smart baskets of tokens ('indices').

PIPT YETI ASSY YLA

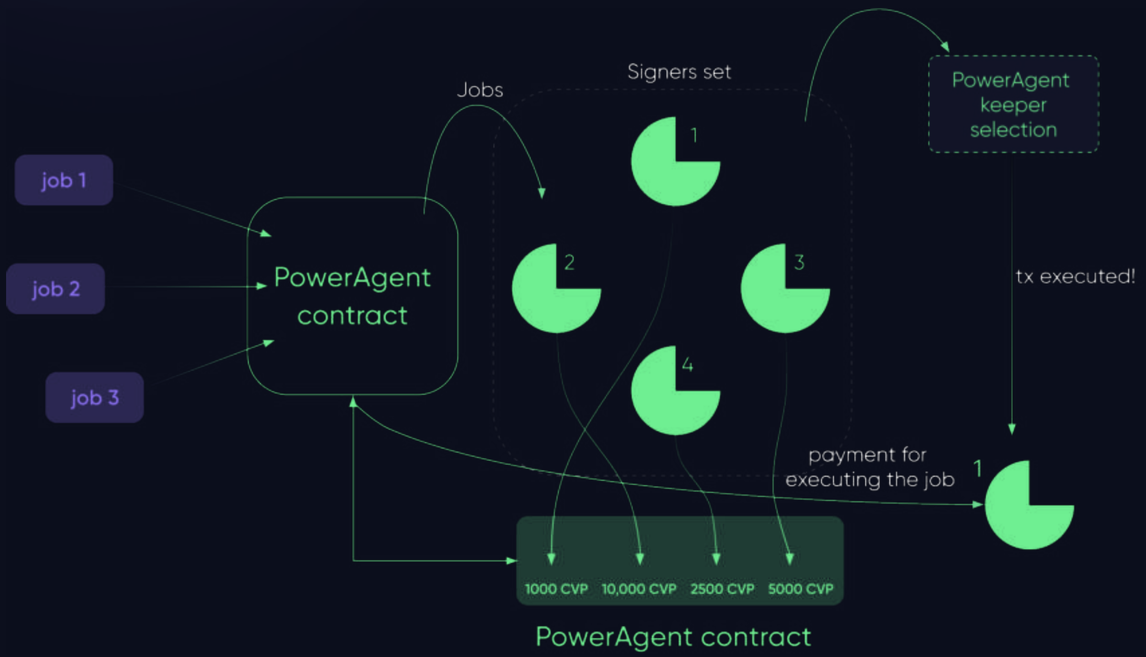
- **We are automating Defi for the last 3 years:)** We built our own Power Agent V1 automation network back in October 2020 and used it for Smart Baskets.
- Featured: *Bankless*, *Paul Veradittakit's newsletter*, *Binance Research*, *Cointelegraph*, *Messari Crypto Thesis* and *articles*.
- **CVP** is listed on Binance.



# PowerAgent v2: How it works

The main function of the PowerAgent v2 network is to provide automated guaranteed execution for Jobs (Tasks) created by users and protocols

1. You deploy a smart contract on an EVM chain with a function that has either a  time-based (interval job) or logic-based (resolver job) execution conditions.
2. You add a job using the user interface to the Power Agent contract, deposit ETH (or native token) for payment, and specify the stake range.
3. The network automatically and randomly assigns a Keeper for execution from hundreds to thousands of Keepers, depending on your specified stake range.
4. The Keeper executes the job, receives payment, and the network randomly assigns the next Keeper.
5. If the Keeper doesn’t execute the job on time, other Keepers will slash him and execute the job.



# Types of Jobs

**Interval Jobs** - regularly executed Jobs (every dT)

- **Selector Job**

  The simplest type of a Job. Call contract with no input data.

  Harvest Vault every 12 hrs

- **Job with predefined CallData**

  used when the target smart contract needs predefined input parameters (CallData). The Job Owner sets the CallData, no computation required.

  DCA Strategy - execute swap_exact_amount_in in AMM every 12 hrs, buying some token for  X ETH per one swap

# Types of Jobs

**Jobs with a Resolver** - an advanced type of Jobs executed based on on-chain logical conditions. The Job Owner provides a separate function (Resolver) that processes on-chain data and outputs true/false and *CallData to execute*.

Read X, Read Y,

if X<Y -> call method A (X,Y), if X>Y -> call method B (X,Y)

if X=Y - condition for execution isn't met, Job is not ready for execution

# Keeper assignment and execution

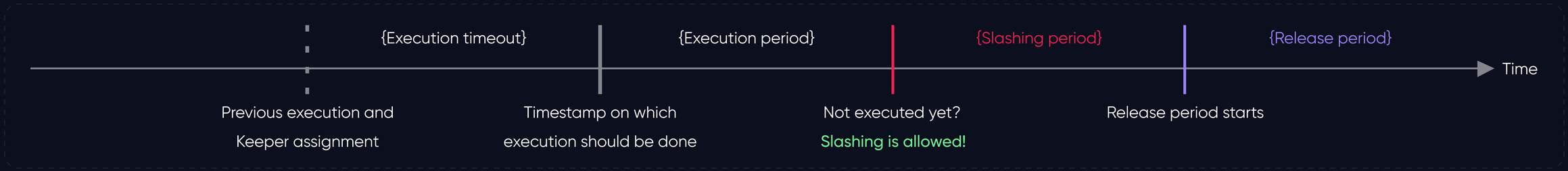
A truly decentralized Keeper network should be powered by random mechanisms and not by off-chain task assignment rules.

**Power Agent V2** uses pseudorandom numbers from the Ethereum consensus layer to assign Keepers for job execution.

- Currently, Keepers are assigned equally, regardless of their stake. Stake-weighted selection is upcoming.
- Job owner defines an interval of Keeper stake (lowest and highest bound) that is eligible for the Job. If stake < lowest bound, Keeper cannot be assigned for the task, if > highest bound, slashing is applied only upto the higher bound.
- Keeper reward for executing a Job = **BaseFee + 10% + stake*multiplier** in native token (ETH, BNB, MATIC, etc).
- The number of tokens that can be slashed is counted as a % of the stake.

# Slashing with Interval jobs

Slashing is needed to penalize malicious Keepers behavior and prevent no-execution of transactions that can lead to losses.




1 When Job is executed the next Keeper is assigned (grey dot line). Keeper has to execute Job when the Execution interval is over.

2 The Keeper has a specific period of time to execute the Job - Execution period, starting from the particular timestamp (grey line).

3 When Execution period ends (red line), the Slashing period starts. The particular Slasher (selected by the Round Robin algo) can slash the Keeper and execute the Tx.

4 If the Job cannot be executed (nor by Keeper, nor by Slasher) due to its malicious design, then the assigned Keeper can release this Job (violet line).

# Slashing with Resolver jobs

Slashing is needed to penalize malicious Keepers behavior and prevent no-execution of transactions that can lead to losses.


1. The Keeper is selected randomly when the previous execute took place (dot grey line) and continuously calls Resolver waiting to receive TRUE+CallData to start executing the task (green tick).
2. Slasher is determined by the Round Robin algorithm and calls the Resolver every block to get information regarding the Job status (is it TRUE or not).
3. If after Execution interval the  Job hasn't been executed, slashing is initiated by proofing exec possibility via virtual execute. At this moment the possibility to execute slashing is secured by Keeper who initiated slashing (yellow line).

# Slashing with Resolver jobs

Slashing is needed to penalize malicious Keepers behavior and prevent no-execution of transactions that can lead to losses.


4 During the slashing timeout, assigned Keeper still can still execute the Job. If the Keeper doesn't do it up to red line, Slasher can execute the Job and simultaneously slash the Keeper.

5 If the Job cannot be executed at all due to its malicious design, governance can release the job in the period after the violet line.

6 Additionally, if the slasher fails to execute the job during the slashing period, another slasher is permitted to reinitiate slashing (starting from the yellow line).

# PowerAgent v2: key features

- **Proven reliability**

  Power Agent V1 successfully operated on the Ethereum Mainnet from October 2020 until March 2023 , supporting the PowerPool product without any vulnerabilities or issues.
  Power Agent V2 has passed several public security audits and is running successfully on the testnet. The community bug bounty program starts in the near future.

- **Permissionless nature**

  Power Agent V2 offers a completely permissionless network where you don't need to pass KYC, be added to the team's whitelist, or sign any agreements. To run a node, all you need is to do set up a server, have access to an Ethereum/other EVM RPC (preferably an independent node) and a Power Agent node, deposit CVP tokens, and earn ETH.

# PowerAgent v2: key features

- **Random keeper selection**

  Power Agent V2 uses pseudorandom numbers from the Ethereum consensus layer to assign Keepers for job execution.

  Currently, Keepers are assigned equally, regardless of their stake. However, in the future, we plan to implement a function that will consider the stake of the Keepers.

  For EVM-based chains that do not have RanDAO, we will implement RanDAO supported by the Power Agent network itself.

- **Full decentralization**

  The network that cannot scale up and is limited to dozens of nodes makes no sense. With the PowerPool approach for random Keeper selection, the Power Agent V2 network can support from hundreds to thousands of Keepers.

# PowerAgent v2: key features

- **DappNode integration**

  To simplify node installation and ensure maximum decentralization, we've integrated with Dappnode. You can install the Power Agent package with an Ethereum node (or any other network node you need) in just a few clicks. If you're already running an Ethereum node on Dappnode, you'll simply receive additional ETH income from job execution.

- **Guaranteed execution**

  The Keeper assigned to execute a job must do so within a specified period of time. If he fails, another Keeper will slash him and execute the job instead. The Keeper who will do the slashing is chosen using a Round Robin algorithm.

- **Stake based pricing**

  When one creates a job, they should set a CVP stake range. Keepers with a stake less than the range cannot be assigned for execution. Keepers with a stake larger than the range can be assigned, but they can only be slashed up to the maximum stake. The larger the Keeper’s stake, the larger the payment for execution.

# Upcoming features

- **ZKP Resolvers (in progress)**

  A resolver is a smart contract function that checks necessary logical requirements and, if they are met, returns true and Calldata for the smart contract. Sometimes it's very expensive or even impossible to check such requirements on-chain. To solve this problem, we are working on zero-knowledge proof resolvers.

  **Value to users:** cheap and verified computations; an option to create Jobs with privately kept algorithms.

- **MEV protection (in progress)**

  Some Jobs cannot be executed in a truly decentralized manner due to potential risks associated with MEV. For this purpose, we plan to launch a special standalone solution with a limited set of Keepers having integration with industry-leading MEV relays.

  **Value to users:** an opportunity to execute MEV-sensitive tasks such as Uni v3 position mgmt, trading strategies, lending market strategies.

# Build with us!

PowerPool DAO approved Ecosystem Fund aimed on supporting builders and contributors to PowerPool:

Builder grants: up to $20,000 (per grant)

Node Runner grants

PowerPool Contributors grants (up to $20,000 per contributor)

Bug bounty (up to $25,000 per vulnerability reported)

# Thank you for your attention!

powerpool.finance

Apply for grant