KOMPUTE

Decentralized cloud computing protocol for Web3.

Whitepaper

13/07/2022

v1.1.2

Contents

1. Disclaimer

The directors of Kompute OÜ have issued this white paper and taken all the reasonable care to ensure that the facts stated on the document are true and accurate, and that there are no other facts omitted which would make any statement in the document misleading. The Kompute tokens are a utility token currently in development by Kompute OÜ developers, the company website can be found at https://kompute.network

2. Introduction

As with all the new technologies, cloud computing was not adopted and implemented overnight. It took time for people to understand what the cloud was, and even more time for tech companies to build their infrastructure on top of it. Taking it from a proof of concept to a fully operating business is not a small task.

In the modern IT landscape, cloud computing has been steadily gaining adoption due to its obvious benefits like high availability, scalability and on-demand self service. By allowing provisioning of resources in response to changes of workload requirements, cloud computing lets individuals and corporations configure flexible architectures.

Enterprises continue to accelerate cloud computing adoption, ~36% of the organizations around the world are spending over $12 million per year on public cloud. It is clear that cloud computing has become a technology that corporations can no longer ignore. Making the most of the cloud is not easy, but a lot of organizations are finding ways to optimize their spending without breaking the budget.

Although cloud computing advantages are relevant, they are not enough as the services are distributed but managed by centralized authorities. Centralization entails multiple security and anonymity issues. Kompute's vision is creating a decentralized computing network based on the Ethereum blockchain and Kubernetes.

Kompute network is designed for all the types of consumers. It connects consumers of cloud services with resource providers of the network who can generate revenue by providing their computing resources. Users can deploy any type of application, service or even virtual environments interacting with the Ethereum blockchain.

3.1 Decentralized cloud computing

In contrast to centralized permissioned providers, decentralized cloud computing leverages infrastructure that is designed to mitigate undue control or influence. Using a permissionless model enables developers to employ their services with reduced restrictions.

Making use of the web interface or the command line and an ERC-20 wallet, users of our network are able to interact with the Kompute contracts through Web3 libraries and request computing resources. The computing resources can be requested via services of the protocol, the entry services that will be launched with the mainnet are:

- **Serverless computing:** Serverless is a cloud computing application development and execution model that enables developers to build and run application code without provisioning or managing servers or backend infrastructure.
- **Container service:** Highly scalable, high performance container management service that supports docker containers and allows you to easily run applications.
- **Virtual machines:** Allows to run an operating system that behaves like a completely separate computer.

Resource providers can contribute computing to the network and generate an annualized return by staking a specific amount of tokens. The reward will be adjusted based on staking period, resources contributed and other factors that are automatically calculated.

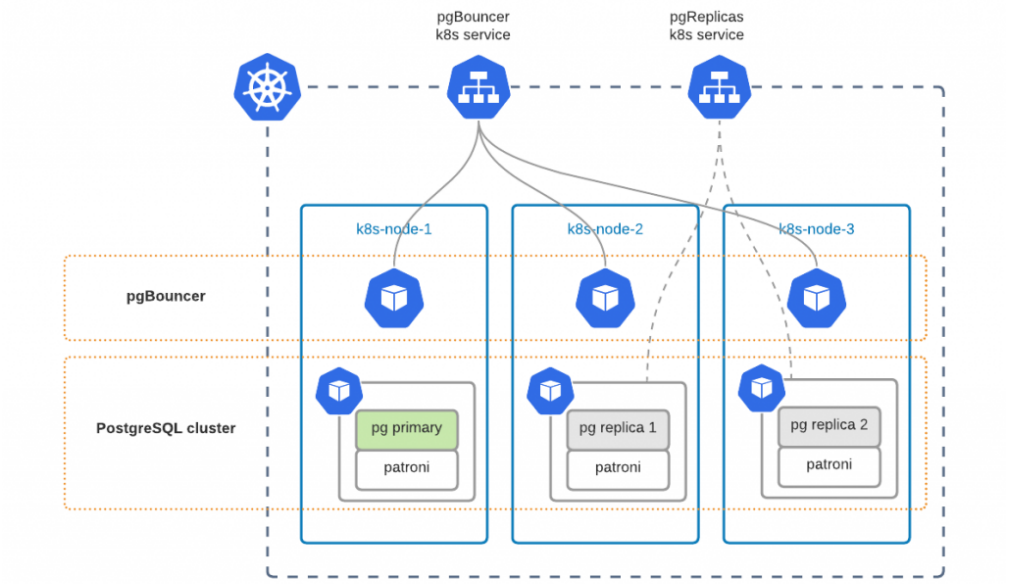
3.2 Based on kubernetes

The off-chain layer running on the nodes is composed by Kubernetes nodes. Kubernetes, often abbreviated as “K8s”, orchestrates containerized applications to run on a cluster of hosts. The K8s system automates the deployment and management of cloud native applications. It distributes application workloads across a Kubernetes cluster and automates dynamic container networking needs. Kubernetes also allocates storage and persistent volumes to running containers, provides automatic scaling, and works continuously to maintain the desired state of applications, providing resiliency.

A kubernetes cluster is composed of master and worker nodes. Master nodes assume the *“Gatekeper”* role inside our network, hosting the control plane, a set of components that makes global decisions about the cluster (for example, scheduling), as well as detecting and responding to cluster events. Worker nodes assume the *“Resource provider”* role inside the network, and are responsible for hosting the containerized workloads of the applications.

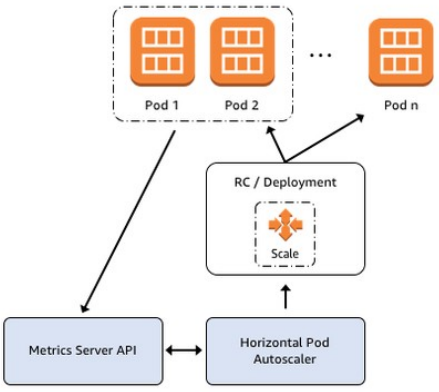
3.3 High availability

Due to Kubernetes design, the network is redundant and provides high availability at multiple layers. Worker nodes are allocated across multiple regions ensuring a distributed network. Users are able to deploy multiple replicas of their application distributed by Kubernetes between all the workers based on the available capacity, this is a key feature to not overcommit node resources. If one of the replicas of your application gets terminated by any reason (for example a node failure) it is automatically reallocated by Kubernetes on another worker.



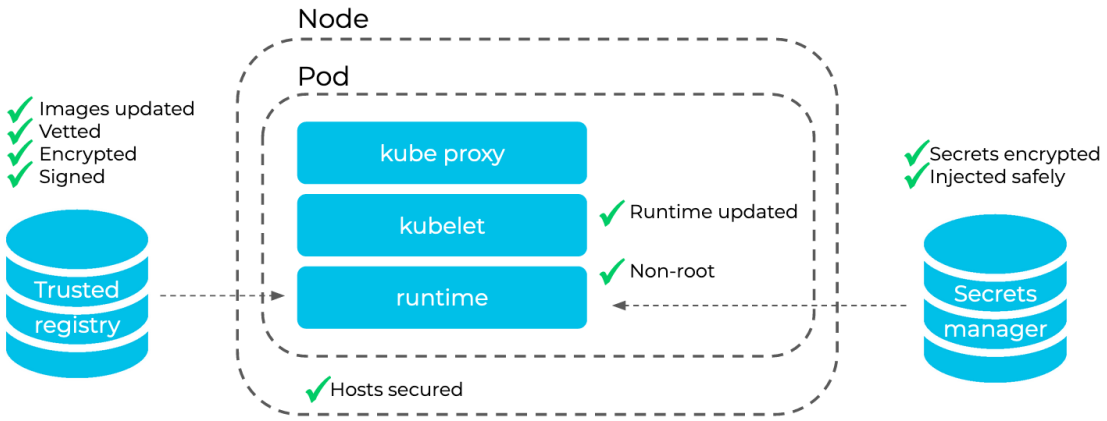
3.4 Scalability

The applications running on Kompute network have the ability to scale based on the resource usage on the container, for example automatically create multiple replicas of your container if it reaches a certain cpu / memory threshold, or remove replicas if the usage of the resources is under a specific threshold. This allows the end user to dynamically provision resources without worrying about wasting resources.

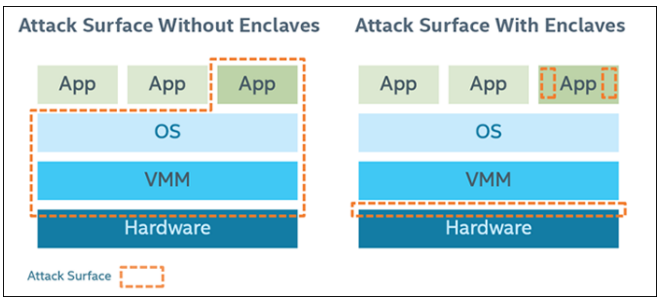


3.5 Security

The containers that perform computational tasks on the network use CRI-O as container engine, which removes all the linux capabilities from the container itself. This secures the runtime from malicious actors gaining access to it. We have also implemented other security measures at kubernetes level which you can refer to below.



Kompute is also compatible with trusted execution environments, ensuring the confidentiality of the software running on the network. Intel SGX technology is used for this purpose, it allows user-level code to allocate private regions of memory, called enclaves, which are designed to be protected from processes running at higher privilege levels.



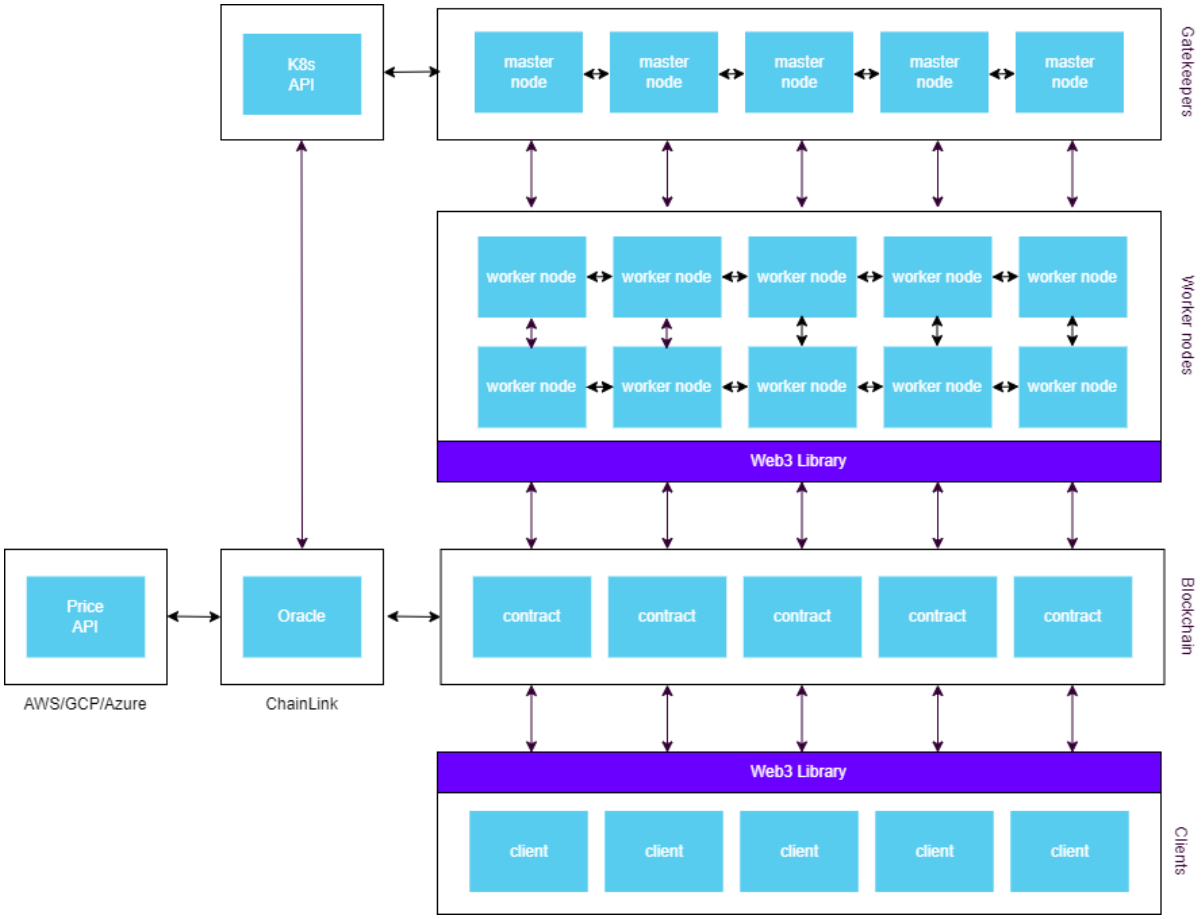
The memory of secure enclaves is also encrypted to physical attacks. These security measures prevent even system administrators with physical access to the SGX workers from tampering with the application once it has been started. Intel SGX offers data sealing support which allows enclaves to persist the data securely such that the data can only be accessible by the enclave. Through remote attestation, it enables third parties to verify that an application is indeed running inside a secure enclave and the application has not been tampered.

4. Architecture diagram

There are 3 roles in the network, gatekeepers, worker nodes and clients.

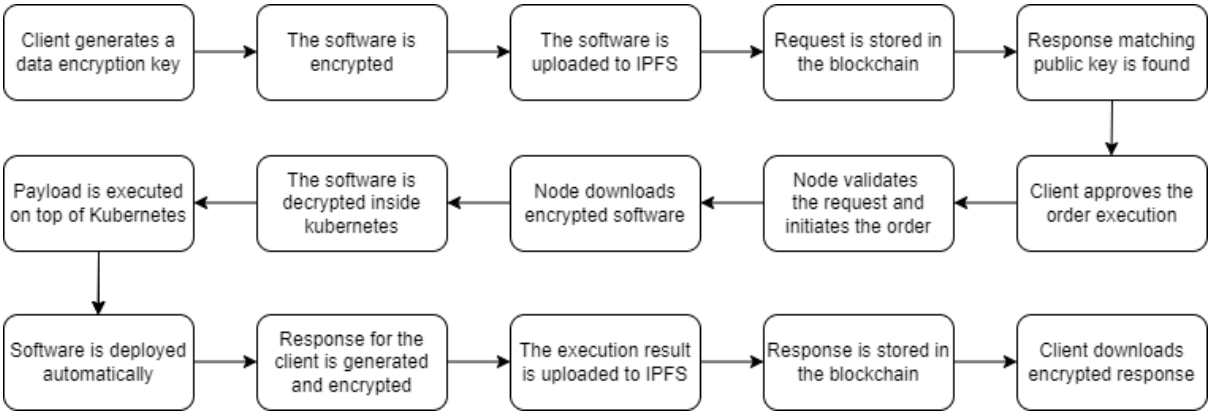
- **Gatekeepers:** Nodes responsible for maintaining the integrity of the network. Only entities with a very high reputation and selected by the network can perform this role.
- **Worker nodes:** Resource providers of the network, responsible for providing computing resources to the protocol and validating the requests on the blockchain.
- **Clients:** Users of the network, consumers of the decentralized computing services.

ChainLink is used by the contracts to retrieve live data from centralized provider price feeds, this information is used as a reference for calculations mentioned on the next points. It is also used to access the Kubernetes API to get information about the network usage.



5. End-to-end flow example

Refer to the diagram below to understand the basic flow of serverless computing on the Kompute network. The flow is initiated by the client and ends on the client.



6.1 Economy of the network

Kompute network is based on a circular economy model, the asset used to purchase decentralized computing services is the native token KOMP, at the same time the asset earned for providing resources to the network is KOMP.

Inside the economy model, users can act as customers (paying for services) or as resource providers (getting paid for providing resources).

In order to be able to become a resource provider for the Kompute network, users have to stake a specific amount of tokens (500 KOMP) and allocate computing resources for a fixed period of time. Depending on the staking period, the reward received will change.

- **1 month:** 100% reward
- **3 months:** 105% reward
- **6 months:** 110% reward
- **12 months:** 120% reward

Your annualized return will change depending on the costs you incur to become a resource provider (hardware costs, electricity, internet) and the period you set up your staking for. Initially, we plan to offer 2 ways to become a resource provider of the network.

- **Regular resource staking:** Set up your own node and stake 500 KOMP.
- **Pooled resource staking:** Set up your own node and pool 500 KOMP between multiple people, or join a staking pool by providing part of the total tokens.

6.2 Value of the nodes

In order to calculate the amount of tokens that has to be distributed to the node on a weekly basis, a value is calculated multiple times per day with a performance test and stored on the blockchain. The performance test takes into consideration all the resource types and launches a test on top of the kubernetes instance to ensure that the values retrieved by the test are accurate.

V = Sgx.score x S x B x (C1 + C2 + C3)

Sgx.score: Confidentiality value of the node.

S: Multiplier used for the staking period.

B: Multiplier used for the bandwidth, calculated by the performance test

C1: Variable used for the CPU, calculated by the performance test.

C2: Variable used for the RAM, calculated by the performance test.

C3: Variable used for the Storage, calculated by the performance test.

Nodes are publishing their V to the contract via signed transactions. Between every transaction, a certain amount of blocks must pass. Each time they publish their V, the value gets added to their sum of V values for the current cycle.

6.3 Slashing rules

If a worker node shows harmful behavior, a percentage of their staked tokens will get slashed as a penalty, meaning they will lose them.

Slashing rules are structured on 3 different levels of severity. The percentages used on the table refer to the amount of tokens that are removed from the node based on a specific amount of time for illicit conduct on the network. All the tokens removed from nodes due to slashing are redistributed to the rewards contract.

Level 1	Worker offline	10% / 24h
Level 2	Malicious intent or mass error	50% / 24h
Level 3	Serious security risk to the network	100% / instantly

6.4 Pricing model

The reality about centralized cloud computing is that it is expensive, and while paying a few cents / hour per core might sound cheap, your total bill could end in hundreds if not thousands of dollars depending on the infrastructure you are working with.

Our pricing model is dynamic, prices are adjusted automatically via oracles that connect with AWS, GCP and Azure APIs to calculate median prices of the desired resources and always ensure that our prices are cheaper versus them while still being profitable for staking, even if you are renting a dedicated server for your node. The discount applied to the prices retrieved from the oracle will change depending on the network utilization.

(0,75 + networkUtilization^2 x 0,20) x costOfAWS|GCP|AzureService

After doing an extensive research on the computing market, we have realized that cloud providers are ~100% more expensive than regular dedicated server providers on average for on demand services, giving resource providers a big headroom if they decide to set up their nodes on datacenter solutions instead of running the nodes at home.

There is a possible scenario where regular dedicated server providers could raise their prices a lot, resulting in the prices being more expensive than cloud providers like AWS. While this scenario is unlikely, a contingency plan has been prepared for it and would be triggered via governance of the protocol.

The contingency plan stablishes a new pricing model for the network, where the prices will be set taking into consideration the cost of the hardware running on it and the network utilization instead of setting a price range below centralized cloud providers.

(1 + networkUtilization) x (costOfResource x serviceMultiplier)

A fee of 10% is taken from every purchase on the protocol as a commission and sent to the company address. This fee is used to fund the company and pay salaries along with new decentralized services development.

The formulas mentioned on the pricing models are subject to changes.

6.5 Reward mechanism

The reward mechanism works in cycles. Each cycle lasts a certain amount of blocks. During a cycle nodes are publishing their V value to the contract via signed transactions. At the end of the cycle nodes can withdraw the rewards.

Reward tokens are deposited to the contract using addReward() function. This function transfers N amount of tokens to the contract and adds the N amount to the sum of rewards for the current cycle.

A node can claim a reward from a cycle when the cycle ends. It can do that at the end of every cycle or for all the past cycles in one transaction. When it claims, it saves in the contract that the reward for the cycle was claimed.

Reward for node is calculated like this:

cycleReward / cycleTotalV x cycleAccountV

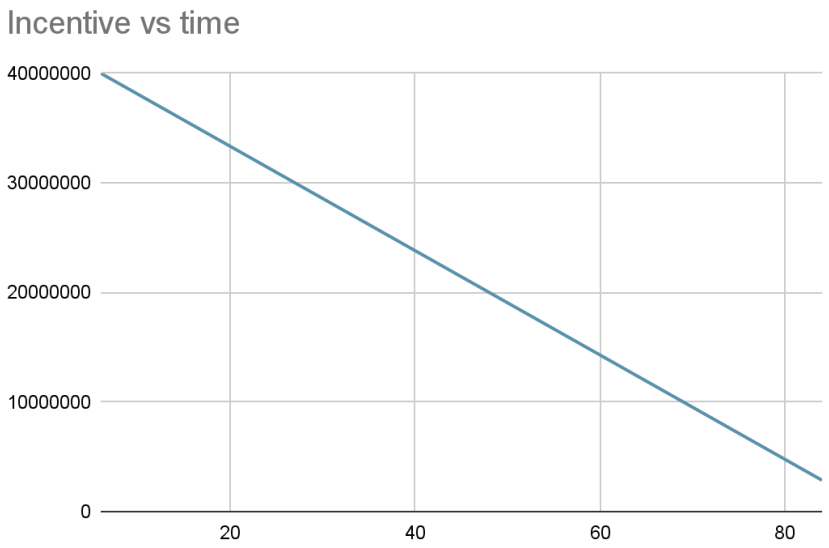
cycleReward = All the reward tokens that were deposited to the contract during the cycle.

cycleTotalV = Sum of all V values that were published by nodes in the cycle.

cycleAccountV = Sum of all V values the specific node published in the cycle

6.6 Incentive mechanism

In order to incentivize network usage there will be a mechanism that will distribute a fraction of the tokens located on the incentives contract to the nodes. This amount of tokens for incentives will decay linearly over time as the rewards from demand take over.



6.7 Token economics

- Name: Kompute
- Symbol: KOMP
- Type: ERC-20
- Final tokenomics TBD

7. Governance

For the initial milestone on protocol governance we establish the foundational components of a decentralized governance system using Snapshot. To name a few of its features:

- Free (gasless) to create proposals and vote on them.
- Votes are signed messages easily verifiable online.
- Multiple voting systems - Single choice, Approval voting, Quadratic voting, and more.
- Flexible voting strategies to calculate voting results

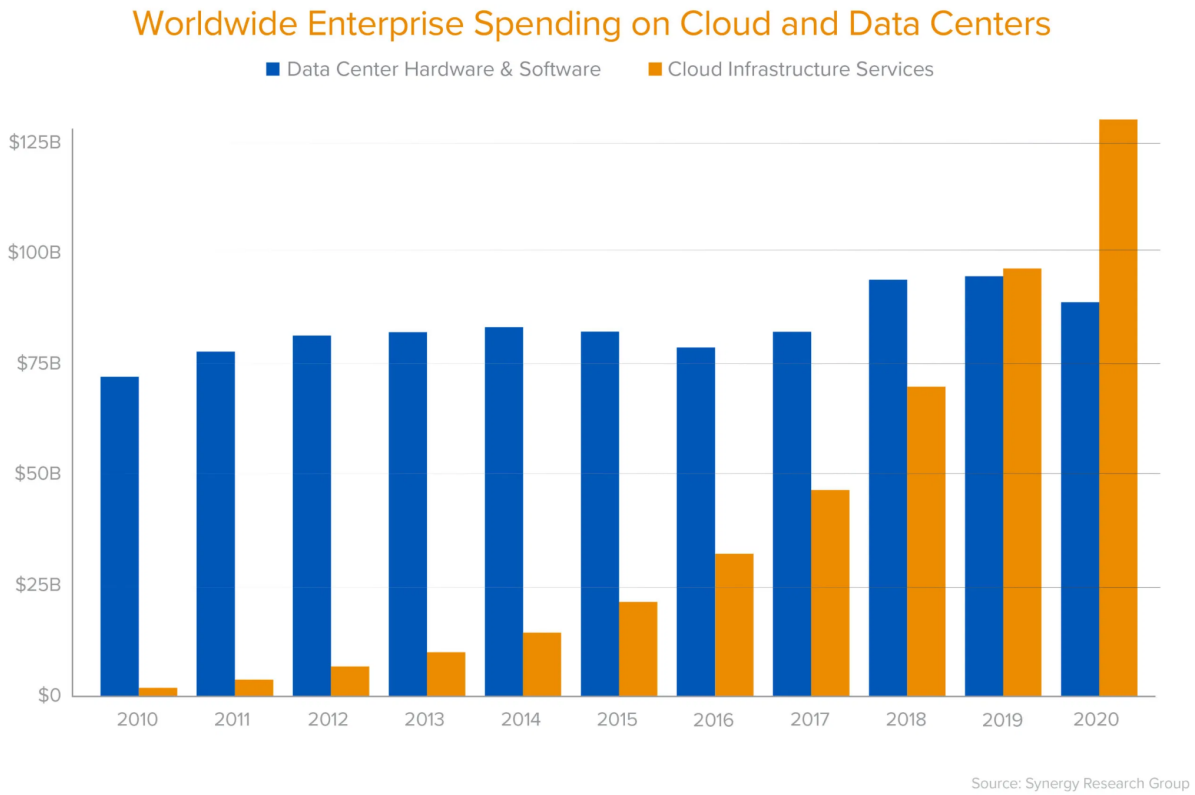
8. Project funding

We are raising a $800k seed round at a $4M valuation to build a fully operative user interface for our MVP and start generating revenue from it.

In return we offer 20% of the company equity and the ability to invest at the earliest round of the token sale.

Refer to the percentages below in order to understand how the funds will be used.

- 70% for product development.
- 10% for strategic partnerships and marketing.
- 5% for administration and operations.
- 5% for community engagement.
- 10% for contingency.



"There is no doubt that the cloud is one of the most significant platform shifts in the history of computing. Not only has cloud already impacted hundreds of billions of dollars of IT spend, it's still in early innings and growing rapidly on a base of over $100B of annual public cloud spend. This shift is driven by an incredibly powerful value proposition — infrastructure available immediately, at exactly the scale needed by the business — driving efficiencies both in operations and economics. The cloud also helps cultivate innovation as company resources are freed up to focus on new products and growth." - Source: a16z