

# Kumo

## Serverless Cloud Infrastructure on the Blockchain

### Abstract

Serverless computing is the next step in the cloud computing revolution. At the dawn of the internet, developers set up their own servers, built their own racks, and managed every facet of the server-application stack. As applications evolved and increased in complexity, the advent of cloud computing -- led by Amazon Web Services -- allowed developers to scale their apps without building new datacenters or manually adding servers.

As apps continue to become more complex, infrastructure overhead for those applications is increasing as well, in the form of operational complexity as well as SREs and Software Engineers. Serverless computing solves these issues for developers by abstracting away the server altogether, and running an application's functions without regard to the server that runs the function. Serverless computing allows developers to scale their applications without any regard for the underlying infrastructure.

The serverless movement is not only good for developers -- it's more cost effective for providers, as well. With serverless, servers are only running when they have functions to execute. They can also run multiple small units simultaneously, which further drives down costs.

We plan to introduce KumoCoin (KMC), a token that can be used for resources on the Kumo blockchain network, the first blockchain-based serverless distributed cloud computing network.

Current blockchain technology to run real world, industrial and enterprise grade applications have been either focused on specific smart contract execution or have been limited to high latency tasks such as image rendering. To date every attempt to add a user compute layer onto blockchains has been in the form smart contracts, which define their own language and VM environment, or have been incredibly domain specific -- not beginning to scratch the surface of being generally useful to a wide audience of developers.

Kumo will bring the blockchain to the world of the average developer and user by giving a direct and seamless way to leverage the blockchain for tasks within existing applications/use cases as well as providing the ability to build complete, highly functional new applications that give 1 to 1 performance and user experience to their current cloud based apps. One way we will solve for this is by having off chain immediate execution of user code; allowing developers to run any type of quick application on Kumo.

Kumo will build a serverless blockchain infrastructure that will allow users to run tasks within their current applications on the Kumo blockchain, within the time and response standards they need, and with the languages they are used to developing in.

The traditional blockchain protocols call for a massive number of servers to be constantly calculating hashes (Proof-of-Work), which is wasteful in the sense of energy as well as computing resources; we will put those computing resources to use executing real work, instead of hashing simply to prove they've done work. We will use a Proof-of-Stake (PoS) protocol to reduce the load of compute resources needed by the network to maintain the blockchain, and then retool the nodes in the network to use their compute to run user functions.

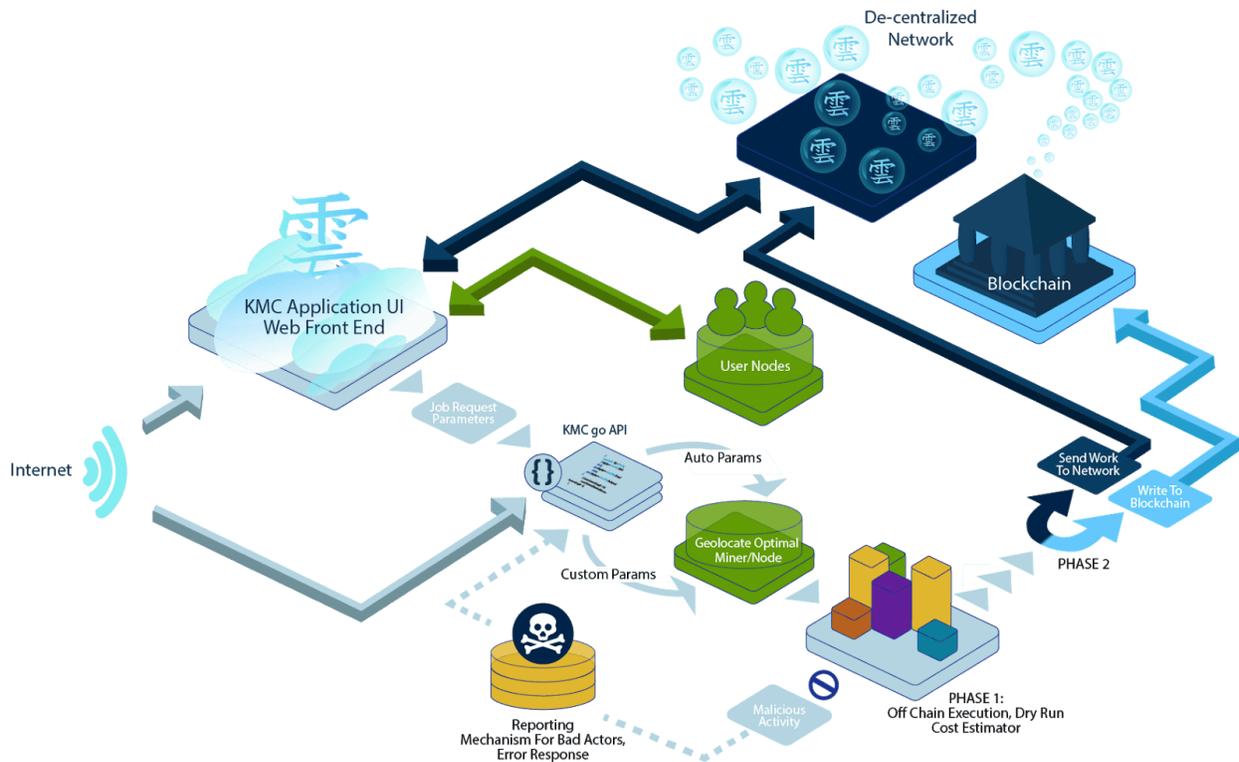
The Kumo network will have an API that makes it easy to submit serverless-esque jobs, paid for in units of user specified KMC/hour fees, collected by miners to run jobs. While the main network (chain) will operate on the public internet, we believe many enterprise software users will also be interested in running their own private chains in order to capture the advantages of serverless computing and our platform while meeting their security requirements.

By creating a parallel and fully functional cloud infrastructure on the blockchain we are able to fully decouple data centers, servers, cloud providers, and other hardware infrastructure from applications - allowing developers to write code and deploy without having to consider scaling, failover or server maintenance - this isn't just serverless; it's cloudless computing on the blockchain.

## Why We're Building Kumo

- Developers want to run real world, responsive applications on a low cost, decentralized network - Kumo is the first team working to make this a reality by leveraging blockchain technology.
  - Provides a decentralized open source "cloud" - with Kumo applications run on infrastructure that is everywhere (and nowhere) at once vs current centralized, private cloud data centers.
- Infinite scale at a task level - no resource planning for scalable jobs/functions; simply write code and deploy; no failover resources wasted running redundant systems. If a task fails another node in the network runs it.
- Allows developers to use common programming languages while building applications on the blockchain, all without blockchain or cryptography experience.
- Interoperability with current serverless applications and use cases such as those that currently run in production on AWS Lambda - this allows current production users to move seamlessly from a legacy cloud providers to running tasks on the Kumo blockchain.

# Technical Specifications



The KMC network will provide low latency access to a large number of nodes. Miners will be incentivized through the fee structure to run quality hardware and servers such as what are currently available from cloud providers such as AWS EC2, Google's GCE or bare metal servers hosted in a data center. This will give users access to bulk commodity infrastructure through allowing miners of all types to participate.

Alongside running jobs, KMC nodes will be responsible for maintaining a blockchain of transactions to track fee collections for executing user functions as well as exchanges of KMC between peers. We plan to accomplish this with a Proof-of-Stake (PoS) system similar to the likes of the one employed by Nxt [2], in order to cut down on computational resources used to verify transactions. It may also be possible to retool existing Proof-of-Work protocols to maintain the actual chain, however since they are hashing for such specific outputs we expect it to be less friction to move to a Proof-of-Stake protocol, as well as being more energy efficient.

Clients will be responsible for submitting function executions to nodes, specifying the fee they are willing to pay per compute hour (KMC/hour) and also some parameters about the shape of their job, such as how long they expect it to run, expected bandwidth and RAM -- these kind of parameters (less the fee) are common in job processing frameworks, such as Hadoop, Spark and other orchestration tools such as Kubernetes and will feel familiar to developers. Adding

these mechanisms will entail making clients have a lot of functionality so that they can route to nodes which will start executing their functions quickly and also to their liking. A key part of the startup for users will be running test jobs to identify nodes that are geo located and performing to their liking (whether it be close to their client or their database or other functions, etc) and then users can scale up or down from there as needed. We will also utilize penalizations through additional and refunded fees for users that specify their jobs by large incorrect margins or miners that are running slow nodes, in order to discourage running slow servers on low bandwidth networks, miners to over subscribe and encourage users to accurately specify their jobs in order for miners to run as many jobs as possible. A lot of this functionality we will build into clients so that users will not have to think about it but they will still get the benefits.

One of the core assets of KMC will be the API for users to submit their functions. This will be an extremely simple way for users to get started running functions, inasmuch that no account creation is even necessary as there is with every other functions platform currently available. While also being easier to get started than the cloud provider functions platforms, we aim to provide a much easier to use API to run specific compute units than any other blockchain-based computing protocol. These protocols - to date - are limited in function and (in our estimation) not able to run real world applications at scale with the 1 to 1 functionality of a centralized cloud provider. This means that essentially users will specify a Docker image (from Docker Hub or any Docker registry) to be executed on the KMC network, along with a fee and some other parameters (input, memory, etc).

We anticipate tooling and other services to continuously make the ease of use and efficiency of the network evolve naturally via a robust community driven ecosystem economically incentivised to provide SaaS, PaaS, and other developer centric applications via KMCs. This ecosystem could initially be thought of as similar to what has developed within developer tools and services “app stores” such as Heroku, Azure Marketplace, or the AWS marketplace.

Key to all of this is the fee structure, and we believe that using fees to penalize miners for being slow will ultimately encourage them to run high quality servers. On the other end, we have to make sure that users aren't specifying unsatisfiable parameters for their jobs simply to pay reduced fees and penalize miners. We think that penalizing either party for being malicious solves this. Miners will still take the common sense approach of attempting to run functions that pay the highest fees, however there will be additional parameters around bandwidth and expected run time and memory so that a miner may elect not to run a function if it does not believe it can currently meet the user's desired parameters, thus avoiding being heavily penalized for insufficiently running a function. We expect that ultimately there will need to be a 2 phase protocol for submitting user functions, where functions will be run “off chain” after being submitted (so that they can run instantly) and then (post execution) an execution record will be posted to the next block so that the miner may collect fees from the user.

Unlike cloud platforms, since this is running on a public block chain there is no authentication mechanism, no billing mechanism -- everything is pay as you go and getting started is as easy as downloading our CLI tool and having a Docker image and some KMC. We also plan to make it so that users do not have to think about scaling or routing their functions, all provided through our tooling. In addition, we'll make use of existing open source project code bases, such as [IronFunctions](#) - the brainchild of and heavily contributed to by former Iron.io CTO and Kumo project technical advisor Travis Reeder which allows use of wrappers for users to easily transfer their existing and currently in production functions from AWS Lambda (and possibly some other platforms) to run on the KMC network so that Lambda users can move over seamlessly. This ease of use and immediate portability will allow what may possibly be the easiest way currently available for large scale production users to begin leveraging and utilizing blockchain technology to run their existing production applications. Examples of large enterprises currently using AWS Lambda in production include:

- [AOL](#)
- [Localytics](#)
- [Nordstrom](#)
- [Netflix](#)
- [Thomson Reuters](#)

## Security

One of the primary concerns of any FaaS deployment is security, and KMC is no exception.

While systems like Lambda will assign a fully qualified VM to each function execution, we will explore using containerization as the unit of security. While existing systems like Lambda, Google Cloud Functions and OpenWhisk run on "trusted" servers they manage and operate in-house, this also contributes to centralization and vendor lock-in. With the KMC network, we will deploy a combination of virtual machines and docker to ensure that neither users nor miners are able to access one another's container images.

Another security concern we'll be addressing is the potential vulnerability for functions to execute outside of their containers and look around the host, thus creating the potential for malicious user applications to run. We plan to run a modified security system that would allow miners to simply blacklist images. Miners could share common blacklists to other miners to stay up to date or getting started with sensible blacklist defaults. This mechanism is simple, however it does give quite a lot of power to miners in that they can decide which images to run to some degree and could penalize certain users even if the users weren't building malicious images (the intention of use), but perhaps that could be useful in some capacities. At least, as long as miners are staying active in a community with one another, any user that manages to create a malicious image can be blocked across the mining pool rather quickly, as opposed to trusting a

set of image validators (a separate entity with separate goals) to catch this. In this way, it would be similar to how the DNS / spam system works currently at the ISP level. However, by facilitating these blacklists to live in the public on github or similar vs a centralized ISP's database it would be more efficient to prohibit use by bad actors. Ultimately we will continue to hone in on how to make the network more secure including replying on contributors in the community for solutions as well as providing bounties and airdrops to contributors to our open source code base.

## Token Sale Structure

KumoCoins (KMC's) will initially be issued as ERC-20 tokens to give our users immediate control of their purchase. As Kumo continues to develop our proprietary blockchain we will create and issue KMC's to allow a direct exchange of existing Kumo ERC-20 tokens for KMC's for use on the Kumo Network and Kumo Application Marketplace

KMC has chosen to introduce new KMC tokens via a crowd sale structure. We will allow users to purchase KMC's starting July 21st, in a pre-sale at a rate of 1 Ethereum = 500 KMC tokens. This pre-sale will be limited to 6,000,000 KMCs and will close immediately once they are sold out.

Following our pre-sale we will begin taking deposits for the main sale of KMCs that is currently scheduled for October 1st, 2017. We believe that it is both against the spirit of the blockchain community and potentially a detriment to adoption of use of the KMC network to have a high concentration of KMC's in a few users hands. To this end we seek to have a broad distribution of tokens via allowing both large and small users alike to buy their tokens in the sale via a deposit mechanism.

To accomplish this there will not be a limit of KMCs sold during the deposit period. Following this the Main Sale will commence and will be limited to 20,000,000 additional KMCs.

The deposit period sale rate is 1 Ethereum = 200 KMCs

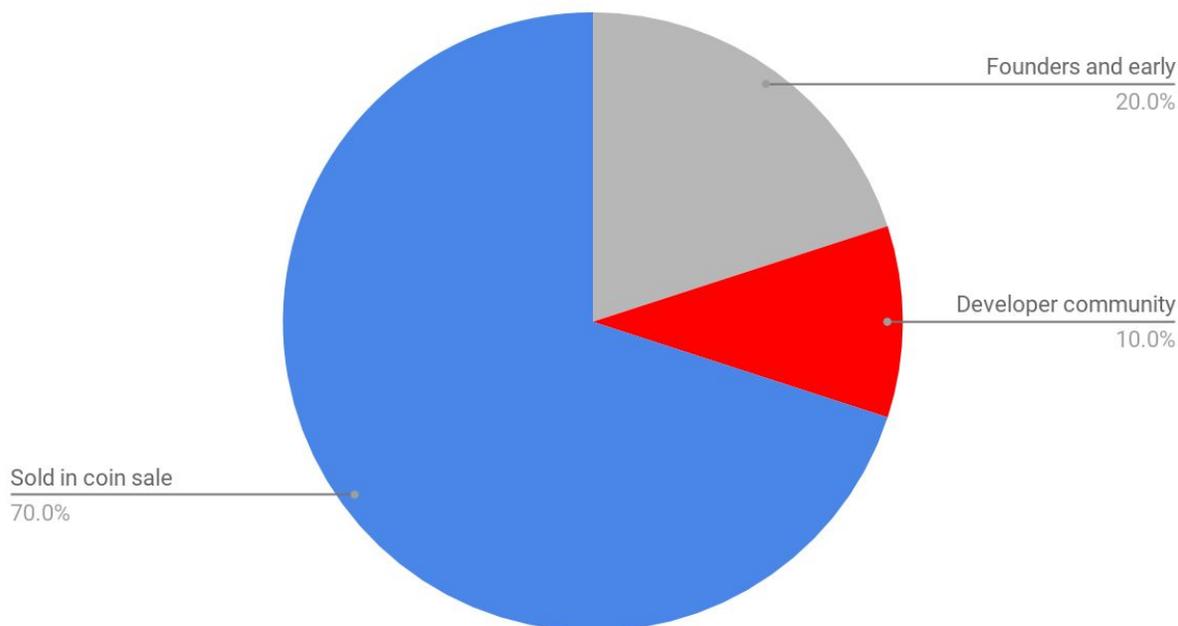
The remaining 20,000,000 KMCs of the main sale period will be sold at a rate of 1 Ethereum = 100 KMCs

Prior to the main sale we will continuously release additional documentation, notices of our hiring successes, a complete technical white paper, and a public - open source - code base that we welcome contribution to.

# Kumo FAQs

- What are KMCs?
  - KMC is the coin used to pay for services on the Kumo network and are available for pre-purchase directly from Kumo.
  - They can also be earned by providing compute resources to the network and will be used to pay for services, applications, and other tooling within the Kumo marketplace.
  
- How much is being sold?
  - KMCs will be sold in the following manner and at set rates during the sale:
    - Pre-sale - 6,000,000 KMCs are available at a rate of 1 Ethereum = 500 KMCs
    - Deposit Period - KMCs will be available to be reserved during the deposit period at the rate of 1 Ethereum = 200 KMCs; this period will be uncapped.
    - Main Sale - The main sale rate will be 1 Ethereum = 100 KMCs. The main sale will end once 20,000,000 KMCs are sold in the Main Sale.
  
- What cryptocurrencies are acceptable in the coin sale?
  - We will accept Ethereum
  
- When will the crowdsale begin and end?
  - The pre-sale will begin July 21st, 2017 and end once 6,000,000 KMCs are sold, Following this the reserve/deposit period will commence and continue through October 1st, 2017 at which point the main sale period will start and continue until 20,000,000 KMC's are sold in the Main sale period.
  
- Are KMC tokens transferable?
  - Yes; following their development and release
  
- What is the distribution of KMC's?
  - 70% of coins will be sold in coin sale periods
  - 10% will be used for community
  - 20% for advisors, founders, developers, and early buyers

## Token Distribution



## Team

Patrick Coleman - Patrick has been in the developer tools, infrastructure, and enterprise application space since joining Exceptional Cloud Services. After Exceptional was acquired by Rackspace, Patrick worked with their acquired startups such as Mailgun, ObjectRocket, Redis-to-Go, and Airbrake and later joined serverless infrastructure startup Iron.io, which raised more than \$17MM in venture capital to be one of the first pioneers in the serverless computing space. Patrick has a deep understanding of how technology is built and deployed in a modern cloud environment and has helped with deployments at small startups, large venture-funded companies, and Fortune 500 organizations. Patrick began to get involved in the crypto world in early 2013 eventually running an alt-coin mining operation out of colo data center with Kumo developer Matthew Martinez and has since been active in supporting new coin adoption - buying into initial coin sales such as Ethereum, Stratis, Waves, and Antshares as well as being one of the first to both POW and POS mine Decred.

Justin Mares - Justin was the former Director of Revenue at Exceptional Cloud Services until its acquisition by Rackspace. He also co-authored the best-selling startup growth book, *Traction*, with Gabriel Weinberg, and has been founded, ran, and invested in multiple technology and physical product startups to date. Justin was also early (with Patrick and Matthew) in investing in

the crypto space, having bought into the initial Ethereum and several other coin sales, as well as working with Patrick and Matt to run an alt-coin mining operation.

Matthew Martinez - Matthew is an experienced cloud integration DevOps engineer and certified AWS architect. He Authored, maintained and improved config management and CI pipeline for Rackspace OpenStack Trove team. He's written and maintained tools to automate procedures and tasks using Python and Ruby on small and large scale. He's built distributed, highly available and resilient environments within AWS ecosystem for big and small managed support customers. Matthew is a hands on engineer who ran a mining operation out of a colo, while also running a public script mining pool (Hashville). He is passionate for building and evangelizing blockchain technology. Matthew's been involved in the crypto scene since 2010.

### Technical Advisors

Travis Reeder - Travis has 20 years of experience developing high-throughput, high scale web applications and cloud services. Travis is currently an Architect at Oracle building cutting edge cloud technology. He was co-founder and CTO of Iron.io, a pioneer in serverless computing, heading up the architecture and engineering efforts. Prior to Iron.io, Travis founded ecommStats Web Analytics and SideStripe Social Search, both dealing with large amounts of distributed data processing, and has led development teams at Siebel Systems (sold to Oracle) and db4objects.

His past experience naturally drew him to cloud infrastructure as it is a natural fit for building scalable applications. Travis holds a Bachelor's degree in Computer Science and is a known technical speaker and writer about the Go programming language, Docker and serverless computing.

Conrad Wiedenkeller - Conrad has been in the software industry for 10 years. Starting his career at Rackspace in 2008. There he worked on Openstack working on the Trove and Swift projects. His interests include brewing delicious beer, making cheese and curing meat and being an all around swell guy.

Russell Smith - Russell is the CTO and Co-Founder of Rainforest QA. Previously, he provided consultancy for startups and companies around development, ops, architecture design and capacity planning. Specialties: Development, developer workflow, devops, linux, Debian, CI, benchmarking, profiling, bug fixing, performance, scalability, ops planning, capacity planning / modeling.

## Appendix / References

- [1] <https://bitcoin.org/bitcoin.pdf>
- [2] <https://bravenewcoin.com/assets/Whitepapers/NxtWhitepaper-v122-rev4.pdf>
- [3] <https://golem.network/doc/ReleasecandidateGolemwhitepaper.pdf>
  - 1. About serverless - <https://medium.com/social-capital/understanding-abundance-part-3-the-next-big-thing-cc42bcf8a76d>
  - 2. Serverless architectures - <https://martinfowler.com/articles/serverless.html>
  - 3. What is serverless computing - <https://www.iron.io/what-is-serverless-computing/>
  - 4. Why the fuss about serverless - <https://hackernoon.com/why-the-fuss-about-serverless-4370b1596da0>

Kumo LLC - Canon's Court, 22 Victoria Street, Hamilton HM 12 Bermuda