The Krate Paper

v3.4

A Cloud Marketplace for Self-Sovereign
Identity, Compute, Storage, and Network Resources

Prove Anything. Solve Everything. Disclose Nothing.

Abstract

The KRATE Cloud will be introduced as a public and permissionless cloud offering consisting of storage, compute, and network resources that are supplied by peers. It employs a unique combination of software and hardware technologies that are used in distributed systems, identity management solutions, cryptographic schemes, clustered file systems, content delivery networks, and blockchains.

The KRATE Cloud will allow data to be stored, transmitted, and computed without ever requiring decryption. As a result, Krate's users can prove anything and solve everything while disclosing nothing to others.

This is a living document that will receive updates as the KRATE Cloud is developed.

Abbreviations & Acronyms

Abbreviations

- Mempool: *Memory Pool*
- PubSub: *Publisher-Subscriber*

Acronyms

- 2FA: *Two-Factor Authentication*
- API: *Application Programming Interface*
- CDN: *Content Delivery Networks*
- CLI: *Command Line Interface*
- CPA: *Chosen Plaintext Attack*
- CVP: *Closest Vector Problem*
- DCL: *Distributed Compute Labs*
- DDoS: *Distributed Denial of Service*
- DoS: *Denial of Service*
- FHE: *Fully-Homomorphic Encryption*
- IoT: *Internet of Things*
- IPU: *Intelligence Processing Unit*
- ISP: *Internet Service Provider*
- KC: *KRATE Cloud*
- KCCA: *KRATE Cloud Client Application*
- KHG: *KRATE Hashgraph*
- KVM: *KRATE Virtual Machine*
- LRP: *Lattice Reduction Problem*
- MTP: *Merkle Tree Proofs*
- NIST: *National Institute of Standards and Technology*
- NP: *Nondeterministic Polynomial (time)*
- OS: *Operating System*
- PHE: *Partially Homomorphic Encryption*
- PoC: *Proof of Capacity*
- PoS: *Proof of Stake*
- PoW: *Proof of Work*
- SC: *Smart Contract*
- SED: *Self-Encrypting Drive*
- SPS: *Sequenced Proof System*
- SSN: *Social Security Number*
- SVP: *Shortest Vector Problem*
- SWHE: *Somewhat-Homomorphic Encryption*
- UI: *User Interface*
- ZKP: *Zero-Knowledge Proof*

Table of Contents

1. Introduction

1.1. What is Krate Distributed Information Systems?

Krate Distributed Information Systems Incorporated, herein referred to as Krate, is a Canadian business based out of Saskatoon, Saskatchewan and incorporated under the Business Corporations Act of the same province on November 21, 2017.

1.2. Who is the Krate Team?

The Krate team includes eight employees with specializations in systems analysis and development, software development, internet technology, network administration, computer information systems, blockchain, data privacy, cryptology, cryptocurrencies, decentralization, marketing, journalism, and graphic design.

Five R&D partners are included on the Krate team with specializations in distributed systems, ubiquitous computing, data science, applied mathematics, artificial intelligence, and machine learning.

The Krate team is advised by seven experts with specializations in finance, international business, enterprise selling, strategy & practice, complex contract negotiation, mergers and acquisitions, marketing strategy and analysis, customer experience, information technology, information security, project management, and governance.

1.3. What is the Mission of Krate?

Our mission is to disprove the idea that data breaches are an intractable problem and that advanced technology requires the sacrifice of privacy. As such, we provide data management technology that will prevent data breaches in an efficient manner and empower data owners to self-determine their privacy.

1.4. What is the Vision of Krate?

Our vision is a future where the sovereignty of data owners is recognized as a global human right and where technology empowers them to effectively and efficiently enforce their rights.

1.5. What is the KRATE Cloud?

The KRATE Cloud (KC) will be the new gold standard of cloud services. It will feature a decentralized structure, offer storage, compute, identity, and communications solutions, introduce a new cryptocurrency, provide data owners with total control of their online information, and leverage the world’s first commercial implementation of a security technique that has been sought by cryptographers for nearly fifty years: Fully Homomorphic Encryption (FHE).

It will offer the ability to publicly but securely store, analyze, and transmit data without the necessity of ensuring that the devices used in the process are secure and trustworthy. Additionally, it will enable users to loan and rent their devices’ storage, compute, and network resources over the KC in exchange for its cryptocurrency, and it will provide an advanced distributed ledger (i.e., blockchain) that operates orders of magnitude faster than similar solutions. Conceptually confounding but practicable nonetheless, it will:

1. process encrypted data without ever decrypting it;
2. validate user identities without ever disclosing them; and
3. facilitate safe transactions with untrusted parties.

These feats will be accomplished by combining cloud and blockchain technologies, such as smart contracts and Krate's new implementation of Hashgraph, with FHE, zero-knowledge proofs (ZKP), erasure coding, tiered checksums, masternodes, and other techniques. It will revolutionize data management and cybersecurity, as it is capable of conclusively ending the identity theft/data breach crisis, and it will offer the ultimate system for secure and confidential rule-matching.

1.6. Who is the KRATE Cloud For?

Despite its intimidating technological underworkings, the KC will be useful and usable for everyone.

Blockchain, a central component of the KC, is viewed by many as being too complicated to bother with, but this is largely due to a misconception that complex concepts must be implemented in a complex manner. While the technical details of a blockchain's construction and operation are indeed complicated, this doesn't prevent their implementation from being designed with the user in mind. Our developers understand end-user needs because they're end users too. As such, the KC has been designed to be easily used by everyone—they won't even need to know what a blockchain is. Storing data through Krate's service will be as simple as uploading and moving files on a computer in either traditional or drag-and-drop fashion, so its advanced capabilities come packaged in a familiar and intuitive interface.

As developers, we love to work with systems that make our job easier, so we have ensured that the KRATE Cloud will be developer-friendly. Its entire tech stack will be built on .Net., which means that developers will be able to build modules and write smart contracts for the KC in any language capable of compiling to .Net Core or .Net Standard. Additionally, the on-ramping process of transitioning to the KC will be eased by its modular design, as it will allow developers to pick and choose individual components to begin with. This will allow developers to continue using their preferred tools and coding language throughout the process.

Furthermore, many industrial sectors are searching for a data storage solution that offers greater security than current offerings without compromising accessibility. Some medical institutions, for example, are resorting to storing sensitive documents in vulnerable services like Dropbox, simply for lack of a better solution. Such practices will no longer be necessary, since the KC will provide unrivaled security without introducing tedious security measures that make its use impractical.

1.7. What is the KRATE Cloud Council?

The KRATE Corporation will be responsible for the initial development of the KRATE Cloud in order to ensure that it exceeds regulatory standards. After releasing its first production version, the project will be opened to the community to permit and promote the continuous development of and contribution to the codebase. To maintain its high standard of software development and regulatory compliance, the KRATE Cloud Council will continue to oversee the project as a curator from this point on. In this capacity, the KRATE Cloud Council will rely on the advice of its volunteer experts to approve or deny governance proposals submitted by the community. Criteria for approval or denial of will depend heavily on ethical and legal considerations, its feasibility, and its effect on the network as a whole.

2. The KRATE Cloud's Design

2.1. Privacy by Design

The architectural design of the KC has been guided by the methodology of privacy by design. Its seven foundational principles are:

1. Privacy embedded into design
2. Privacy as the default setting
3. Proactive rather than reactive measures; preventive rather than remedial actions
4. Full functionality achieved through positive-sum rather than zero-sum objectives
5. End-to-end security – full lifecycle protection
6. Visibility and transparency in component parts and practices – keep it open
7. Respect for user privacy – keep it user-centric

2.2. Components of the KRATE Cloud

Most generally, the KC will be comprised of:

1. **The KRATE Core Network:** This will be the primary network that connects together the Storage, Compute, Identity, and Communications Network. It will protect data with end-to-end encryption and also facilitate secure connections between devices. It will dynamically optimize transmission of data along the fastest routes. Additionally, users and organizations with multiple devices will be able to ensure that they communicate their data only between them and never to a third party.
2. **The KRATE Storage Network:** This will be the foundation of the KC's distributed file system. It will pool storage resources from connected devices and allow device owners to own and rent them in exchange for our cryptocurrency, the KRATE Token. It will give users unprecedented control of their data by enabling them to upload, share, revoke access, and permanently delete data at any time, and it will provide an excellent solution for archiving data over long periods of time, thanks to integrated disaster recovery and fault tolerance techniques.
3. **The KRATE Compute Network:** This will allow users to access enormous data-processing resources, while also providing the unparalleled security of FHE. It will connect users of the KC to the service offered by Distributed Compute Labs (DCL), a partnering company that specializes in distributed compute resources.
4. **The KRATE Identity Network:** This will enable users through ZKP technology to create self-enrolled digital identities that are linked to official documents like passports and driver's licenses. Issuers of such documents will be able to produce a cryptographic signature that validates the identity and signs a ZKP on the KRATE Identity Network, and the ZKP will allow other third parties to verify the entity without the entity being disclosed.
5. **The KRATE Communications Network:** This will create dedicated and secure connections between a user's own devices whenever needed, which will provide a solution to the challenge of securing the rapidly-growing number of Internet of Things (IoT) devices.

Likewise, communication between multiple people over video, audio, and text will be handled by secure networks that connect their devices. In both scenarios, Krate will be unable to view these communication networks because they will operate over encrypted and dedicated peer-to-peer connections.

6. **The KRATE Blockchain:** This will be a multipurpose distributed ledger that will manage KRATE Tokens and the KC's internal economy. It will provide a secure, private, and auditable record of actions executed by smart contracts (SCs), which will augment the blockchain with the ability to assign and oversee complex tasks carried out by user devices. SCs will also facilitate transactions between parties, removing the need for a trusted middleman. The KRATE Blockchain will provide users with flexible privacy controls that will determine who may view their transaction history, granting access to auditors, to some or all network peers—or to no one at all.

The hardware and software components that undergird the KC's architecture include:

1. **Device Hardware:** The KC tech stack will be built on the .Net Core 3.1 framework and according to .Net Standard 2.1 formal specifications. As such, the KC will be both platform- and hardware-agnostic, and it will have only one code base for ease of maintenance. This will ensure that the KC's software has a high degree of portability.
    a. **Operating System:** Many operating systems (OS) will be supported by the KC, including Windows 7 SP1+, 8.1, 10 1607+, Windows Server 2012 R2+, Windows Nano Server 1803+, macOS 10.13+, Red Hat Enterprise Linux 6+, CentOS 7+, Oracle Linux 7+, Fedora 30+, Debian 9+, Ubuntu 18+, Linux Mint 18+, openSUSE 15+, SUSE Linux Enterprise 12 SP2 +, and Alpine Linux 3.8+.
    b. **CPU Architecture:** The KC's CPU architecture will support x86 (Windows Client and Windows Server only), x64 (all OSs), and both ARM32 and ARM64 (dependent on OS support).
    c. **GPU:** Nvidia, AMD, and IntelHD Graphics.
    d. **Additional Accelerators:** The KC will exploit the Graphcore C2 Intelligence Processing Unit (IPU).
2. **The KRATE Core:** The KRATE Core will form the foundation of the KRATE Cloud Client Application (KCCA) and, in line with Krate's modular design methodology, will interface with various software modules that expand its capabilities, including those that will make up the KC's essential software suite. The KRATE Core and the following modules will provide the full host of features that will create the KRATE Client Application's user experience:
    a. **Database:** UpscaleDB is an embedded key-value database purpose-built for storing specific data types that are aligned with a host's hardware requirements. It will be used in the KC to optimize the functionality of its algorithms and to attain incredible speeds—90x faster than leveldb and 140x faster than BerkeleyDB (both used by Bitcoin)—which will be necessary in order to avoid devices bottlenecking when adding new data to a user's local copy of the blockchain.
    b. **Micro-HTTP Server:** On "headless" machines (i.e. no keyboard/mouse/monitor), the Core Daemon will use ASP .NET Core to self-host a web user interface (UI).

c. **Governance Module:** The governance module will allow certain blockchain and consensus parameters to be changed without requiring a change to the code.

d. **Blockchain:** The KC's blockchain, which is a shared, time-stamped, append-only, immutable, cryptographically secured ledger of transactions that can be independently validated and verified, will serve as the foundation for smart contract transactions and the internal economy.

    i. **Consensus System:** The KRATE Cloud will employ a complex, novel consensus system called the Sequenced Proof System (SPS), which will combine Krate's implementation of Hashgraph with a threefold proof mechanism in order to solve many of the security problems facing existing blockchains. Its nested proof structure, which will require confirmation from each proof before the next block can be mined, will enable the network to achieve truly secure decentralization will ward off centralization and the threat of bad actors achieving a majority of the network's hashpower.

        1. **KRATE Hashgraph:** KRATE Hashgraph (KHG) is a lattice-structured consensus algorithm based on Hedera Hashgraph and enhanced by Krate. It will determine the sequential order of transactions that occur on the network at a rate of millions per second, and it will maintain security by dynamically filtering out incorrect data, such as double spends and fraudulent transactions.
        2. **Proof-of-Work Module:** The SPS will incorporate Proof of Work (PoW), a consensus mechanism that requires a client (user) to succeed at a computational challenge in order to mine each block.
        3. **Proof-of-Stake Module:** The SPS will incorporate Proof of Stake (PoS), a consensus mechanism involving an auction for the right to mine blocks, which users bid on with cryptocurrency tokens.
        4. **Proof-of-Capacity Module:** The SPS will incorporate Proof of Capacity (PoC), a consensus mechanism that requires users to prove how much storage capacity they have available. The more storage space they have, the greater their chance of mining blocks.

    ii. **The KRATE Virtual Machine:** The KRATE Virtual Machine (KVM) will execute a node's compute, storage, and network tasks, which it will receive from individual smart contracts.

        1. **Smart Contract Module:** This module creates self-executing scripts that will manage blockchain transactions and enable the blockchain to perform computations that allow for distributed and heavily automated workflows. They will be compliant with .Net Standard 2.1 specifications, allowing for easy integration with external systems.

3. **Application Programming Interface:** The KC will use gRPC as its application programming interface (API) because its speed, low latency, and support for both encryption and compression will make the KC viable for financial applications in which millions of secure

transactions must be processed around the world each second. The API will manage communication between the KRATE Core and external modules and services, including:

a. **Command Line Interface:** Developers and administrators who wish to automate tasks using shell scripts can do so using the Command Line Interface (CLI).
b. **Graphical User Interface:** There will be a fully functional GUI for devices, such as laptops, desktops, and mobile devices. It will utilize the Unity framework so that both a single, easy-to-maintain code base and design can be used across multiple types of devices.
c. **Peer Interface:** The peer interface will allow for timely and efficient communication by leveraging publisher-subscriber (PubSub) patterns.
d. **HTTP/2 Interface:** Web browsers connecting to the KRATE Core can do so via HTTP Version 2, the latest version of the HTTP network protocol used by the World Wide Web.
e. **Additional Interfaces:** To ensure a smooth transition when switching from an organization's services to those provided by the KRATE Cloud, interfaces to existing client devices will be provided in an à la carte fashion, allowing organizations to seamlessly retire currently implemented protocols at their leisure.

3. The Four Pillars of Privacy-as-a-Service

A pillar is defined as a firm, upright support for a superstructure. As a digital superstructure, the four pillars of the KRATE Cloud will be its identity, storage, compute, and communications networks, since they will provide the resources that empower the KC's Privacy-as-a-Service (PaaS) layer. The PaaS layer will in turn support the Software-as-a-Service (SaaS) solutions that will be built on it. Through the KC, Krate hopes to reduce the amount of time that SaaS organizations spend on ensuring compliance with applicable regulations so they can instead focus on their own products and services.

3.1. The KRATE Identity Network

3.1.1. Attacks Against Identity

Deepfakes are a type of attack on an individual's identity that involve editing an image or video of a person in order to insert someone else's likeness in their place. The original person is made to look like someone else, so that they appear to say or do things that they never did. Forgery of content is nothing new, but deepfakes leverage powerful machine learning and artificial intelligence techniques to manipulate or generate visual and audio content that is highly deceptive. The main machine learning methods used to create deepfakes are based on deep learning and involve training generative neural network architectures, such as autoencoders or generative adversarial networks.

Synthetic identities or synthetic identity fraud is a problem that is growing in sophistication, intensity, and frequency. A synthetic identity is a combination of fabricated credentials where the implied identity is not associated with a real person. Fraudsters may create synthetic identities using potentially valid social security numbers (SSNs) with accompanying false personally identifiable information (PII). For example, the synthetic may have a real, "shippable" address and the SSN may appear valid, but the SSN, name, and date of birth combination do not match with any one person. There are two prominent methods used to create Synthetic Identities.

- **Manipulated Synthetics:** Synthetic identities based on a real identity where limited changes are made to the SSN and other elements. These are often used by individuals to hide a previous history and gain access to credit and may or may not be used with malicious intent. Individuals with bad credit histories may create fictitious identities to be approved for new credit for legitimate purchases that they intend to repay. The good news for enterprises is that manipulated identities can be detected. The key to identifying manipulated synthetics is that they often collide with the real identity they are augmenting and don't pass validity checks.

- **Manufactured Synthetics:** Originally, these synthetic identities were composed of valid data assembled from multiple identities—sometimes referred to as 'Frankenstein' identities because fraudsters cobble together bits and pieces of personally identifiable information (PII) from real people to create fake identities. More recently, they are built from invalid information, including SSNs that fraudsters choose from the same range of numbers the Social Security Administration now uses to randomly issue SSNs. The PII used to create the account does not belong to any known consumers. This new form of manufactured synthetics is difficult to identify with current techniques.

Another threat to identities is the Sybil attack wherein an identification system is subverted by creating multiple identities. An identification system's vulnerability to a Sybil attack depends on the cost at which identities can be generated, the degree to which the reputation system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically.

3.1.2. Current Solutions

There is a lot of research currently being invested into decentralized identities (DiDs), as they successfully solve the challenge of creating independently verifiable identities, although they don't address the cost of creating the identity or whether they can be independently validated. Several DiDs also apply centralized thinking to their decentralized identity solution by making it impossible to create an identity linked to existing identity systems, such as passports or a driver's license. Some also allow creation of different identities for different purposes. An entity could create a different identity for an online gaming platform from the identity they used for filing their taxes.

These solutions allow people to create at little cost an identity made up of any information they choose, and the information can be repeatedly tailored to each use case as needed. To combat this, some solutions add a weight to identities based on how old the identity is, how many other entities have verified an identity, or a combination of both. This seems sufficient at first glance, but it introduces other problems to this already-complex identity dilemma.

In theory, anyone could create a Bill Gates identity and successfully pass it off as legitimate if the identity were to gain strength solely from the age of the identity, assuming it was created before the real Bill Gates creates his own. On the other hand, if the identity gains strength from peer verifications, then a wealthy person could create many identities (or even the identity of a rival) and validate it by paying a very large number of friends to attest to it.

3.1.3. A Superior Solution

Some solutions promise a self-sovereign identity system but place restrictions on what the identity can contain, or they give some semblance of control over your identity while a third party maintains control of the cryptographic keys that allow access to it.

Zero-knowledge Proof-of-Identity from trusted public certificates (e.g., national identity cards, ePassports, eSIMs) is used on the KC to provide a solution that will end identity theft and prevent deepfakes, Sybil attacks, and synthetic identities. Entities (people, businesses, organizations) will be able to create truly self-sovereign identities that also may be voluntarily connected to official ID documents like passports, driver's licenses, etc.

Using the KC to create an identity that is connected to such documents will offer users a high-value identity at the trivial cost of a single blockchain transaction. The process will begin by first taking a picture of an identity card, such as a driver's license. Next, the KCCA will do a liveness test to create biometric data that is compared to the photo on the passport. The liveness test and passport data are used to generate a zero-knowledge proof, which is signed with the user's private key and then published to the blockchain.

At this point, the information used to create the identity will be deleted from the device. It's important to note that this information is not contained inside the ZKP, so all that remains is a proof *about* the information, not the information itself. Now the published identity can be independently validated by the original document issuer (in this case, the Passport Office) without disclosing any identifiable information, and, once validated, the passport office will sign the identity with their private key, completing the process.

The end result will be an independently validated and independently verifiable identity that is published to a public and permissionless blockchain. Once identities are enrolled with official ID documents, only the real identity can be independently verified without disclosure, which will filter out synthetic identities. It will also prevent deepfakes because it offers no data for bad actors to work with. Lastly, individuals, businesses, and organizations will be able to conduct trade between themselves without disclosing or storing identity information while still maintaining compliance with regulations concerning Know Your Customer/Business, Anti-Money Laundering, and Countering Financing of Terrorism.

3.2. The KRATE Storage Network

The KRATE Storage Network will represent a new class of decentralized file storage solutions that grows in strength as more users join the network because its storage will be acquired through the network from users around the world. Somewhat analogous to a homeowner renting out their empty basement, users will be able to rent out empty hard drive space in exchange for blockchain-based reward tokens. This sounds both unsafe and unreliable, but we demonstrate in Section 7.3. "Securing Data in the Cloud" that this is far from the case. In fact, centralized data storage solutions are far less trustworthy.

3.2.1. The Pitfalls of Centralized Data Storage

To begin with, such services are inherently liable to corruption and censorship; they give consumers only the illusion of control over their data, since it is managed by third parties, and they're susceptible to data breaches, since it's possible for untrusted servers, providers, and cloud operators to retain user data after the user decides to stop using their services. They often struggle to scale and can suffer from a single point of failure in the event of accidental erasure, man-in-the-middle attacks, force majeure events, etc. Additionally, solutions like redundant data arrays and off-site data backups are inefficient, resource-intensive, and time-consuming, but with the current shift in favour of consumer data rights, the market is ready for alternatives.

3.2.2. Controlling Data in the Cloud

The KC will succeed in offering users total control of their data by combining key-pair-based access control lists with state-of-the-art encryption. Using Krate's decentralized model, users will be able to control every aspect of their online data, such as where and how it is stored for compliance with applicable regulations, how frequently it can be accessed, and when to permanently delete it, assuming no legal hold requirements are needed. Users will be able to audit how their data is being used and revoke access at any time. The audit data can be used to support any complaint or action against any entity that misuses a user's data. Additionally, it will be impossible for any entity to force data to be surrendered and decrypted because of the KC's autonomous functionality and distributed structure.

3.3. The KRATE Compute Network

One of Krate's business partners is Distributed Compute Labs (DCL), a company that has developed a distributed compute protocol by which the processing power of many computers can be combined in order to achieve supercomputer capabilities. Krate's clients will be able to access this incredible service, since Krate will be integrating DCL's protocol into the KC. Additionally, Krate will be augmenting this service with its FHE technique, which will secure the data while processing it.

3.4. The KRATE Communications Network

The KRATE Communications Network will set the standard for network speed, reliability, privacy, security, and control.

3.4.1. Networks on Demand

The recent development and rapid expansion of internet connectivity for devices of every kind has far outpaced security measures necessary to maintain the privacy and safety of their users. The vast scope of this problem makes it a daunting challenge to tackle from the standpoint of any one organization seeking to standardize and unify the IoT, but Krate will offer individuals a solution by allowing them to bring their own devices to the KC in order to take advantage of its secure, on-demand networks. These networks will operate within the KRATE Communications Network exclusively between the user's own devices, much like a VPN, and will require no central server to coordinate communication. Depending on the application and network needs, these VPN-like connections will operate at either Layer 2 or Layer 3 of the Open Systems Interconnection model.

3.4.2. Points of Presence

Migration over to the KC will be a logical next step in the evolution of existing Content Delivery Networks (CDNs), since every node in the Communications Network will add another point of presence. The primary purpose of a CDN is to serve content more locally to the end user, and the Communications Network will help to achieve this by making CDN points of presence not just regional to a user but potentially as local as the same neighborhood. This will be of great benefit to any service requiring a CDN with many global points of presence, such as software distribution and multimedia streaming platforms. With over 70% of internet traffic in North America dedicated to streaming services, the ability to deliver fast, reliable, and secure data is more important than ever. The Communications Network will give streaming services a global, built-in distribution platform, providing unparalleled uptime and speed to their users. Game servers will be able to host matches between the players alone, ensuring minimal latency and eliminating the possibility of Distributed Denial of Service (DDoS) attacks. It will also allow users to authenticate the source of online content, and publishers will be able to ensure that licensed content is used only by those who are authorized to do so.

3.4.3. Network Map

Provider and consumer nodes will not be allowed to directly connect to each other over the Communications Network and will be required to pass requests through at a minimum of two different masternodes in order to reduce the chance of bad actors mapping out the network. If a user wishes to grant a third party access to their data, they will be able to provide a network map that will reveal the location of the data's shards and will provide a key that will decrypt them. In order to revoke access, the data owner will simply invalidate the map, making the key useless (you can't use a key if you can't find what it unlocks).

3.4.4. Self-Healing Network

The Communications Network will be self-healing, i.e., it will automatically establish multiple paths between nodes and dynamically select the best connection using the Open Shortest Path First routing protocol. This is important because nodes and general network topography frequently change. The Communications Network will cope with this by keeping open several unique, low-latency paths, which will allow it to immediately resend data along another path if the current one fails.

3.4.5. A National Network

Due to the academic/research value of the KC, Krate has been granted access to Canada's Canary Network. This means that Krate will be able to guarantee that, if needed, sensitive data will never be routed over nodes located outside of Canada if the source and destination are also connected to the Canary Network. In cases where a transmission's source is on the Canary Network but its destination isn't (or vice versa), the KC will be able to minimize its route over non-Canadian networks, although some of the network's path may extend beyond our borders.

4.0. Blockchain

Blockchain will be foundational to the KRATE Cloud. To give a concise definition of this complex construct, a blockchain is a shared, time-stamped, append-only, immutable, cryptographically secured ledger of transactions that can be independently validated and verified. All of this needs to be unpacked, so we will start with the transactions that the ledger records.

4.1. Overview

4.1.1. Transactions

Users of the KC will transact with each other using a pair of cryptographic keys; one of these keys will be public and the other private, which is an implementation of public-key cryptography that will imbue transactions with a) integrity; b) irrefutability; and c) the ability to be authenticated. Transaction data itself will not be encrypted, but the KRATE Blockchain will instead use a ZKP generated from the transaction data that will be signed by the user, allowing anyone else with that user's public key to verify the transaction's source and integrity while maintaining privacy (see the section on ZKPs further down). Completed transactions will be transmitted one "hop" (one node away) over the network in order for other nodes to verify that they conform to the network's rules, passing on those that are valid and discarding those that aren't. In this way, all nodes will eventually receive all valid transactions.

4.1.2. Mining

Each node will store a pool of verified transactions that can be grouped together into a block of data, but since blockchains are intended to grow only one block at time, many nodes will compete in a timed competition for the right to produce the latest block and to earn cryptocurrency in doing so. This process is called mining, and it consists of one or more tasks required of miners by the network. Once mining is complete, the resulting block will be proposed to the other nodes on the network in order to verify that it is acceptable. The block's cryptographic hash (a blockhash) will allow all other nodes to instantly determine whether its answer satisfies the requirement because hash functions are one-way: it is all but impossible to obtain the input from the output, but you can obtain the output from the input. If the block contains a) the preceding block's hash, b) valid transactions, and c) the earliest valid timestamp out of all blocks proposed during that round, it will be universally accepted, its transactions will be applied to the network's world state, and it will be added onto the chain.

4.1.3. Block Size

Not all blocks will be the same size. The blockchain network's parameters will combat large queues in this process by using a variable block size that will adjust the amount of transactions included in each block according to the length of the que, although this mechanism won't be engaged until pending blocks pass a certain threshold.

4.1.4. Consensus

Furthermore, it will be possible to revert the blockchain by a limited number of blocks in case a problematic block is introduced. Once a block is finished and successfully appended to the chain, the mining round will end and the competition will start over. This process, which is called reaching consensus, will be driven by a distributed consensus algorithm that will maintain the blockchain's linear form because, as a distributed construct, it will be predisposed to fork into multiple strands with differing records.

4.1.5. Defense Mechanisms

It is vital for the blockchain to reach and maintain consensus because its ledger is established by agreement between nodes—in other words, the majority shapes the chain. This makes it susceptible to majority attacks in which a bad actor gains control of a majority of the nodes and therefore is able to alter the chain in order to produce illegitimate tokens. To prevent this, a blockchain's consensus algorithm uses various methods to make the mining process challenging for miners. These methods are all designed to prevent alteration of, and attacks against, the system when adding blocks to the chain (see Section 5.1. "Attack Resistance" for an extensive list), and they each have unique strengths and weaknesses. However, many blockchains, including Bitcoin, use only one of them. The KRATE Blockchain will employ four different methods that are combined in its distributed consensus mechanism in order to capitalize on each of their strengths while negating their weaknesses.

4.2. The Sequenced Proof System

4.2.1. KRATE Hashgraph

The first component of the SPS will be Krate Hashgraph, an Asynchronous Byzantine Fault-Tolerant proof. Traditional blockchains like Bitcoin process a mere 7 transactions a second, and this creates a major bottleneck. Furthermore, the speed at which a blockchain is formed is equal to the scale of the network, which means that its processing speed is quickly outpaced as it increases in scale. KHG, however, will be capable of verifying millions of blockchain transactions each second and establishing their sequential order. Instead of maintaining one linear blockchain and pruning off any forks, KHG will allow the chain to fork and then weave each divergent chain back together, thus removing the time constraints posed by traditional blockchains that are necessary in order to reach consensus. This will allow consensus to be reached on the fly at incredible speeds, which in turn will vastly increase how quickly smart contract transactions can be added to the blockchain. Also, the original Hashgraph adds a lot of additional data for the sake of communication and storage, so KHG will include efficiencies for those two problems and others.

4.2.2. Proof of Work

The second component of the SPS will be PoW, which uses computing power to "solve" hashes in a way that is extremely taxing to accomplish and yet easy for other users to verify once solved. The challenge involves "solving" a cryptographic hash that is lower than a certain threshold set by minimum difficulty requirement. Users' processors churn out hashes at incredible speed, hoping to hit the right one. Once a hash that meets the difficulty requirement is produced, the mining node is able to confirm the block's transactions and then add it onto the blockchain.

4.2.3. Proof of Stake

The third component of the SPS will be Proof of Stake. It was originally created to address the energy inefficiency of PoW, which it achieves by introducing stake-based mining. In this proof, an auction is held for the right to mine the block, which means that a user's PoS mining power is determined in part by the total number of coins/tokens that they own. Additionally, their coins are multiplied by the length of time that they have been owned, which is a factor that helps to even the disparity between rich and poor miners. Coins must be held for a maturity period before generating weight. Ultimately, both the number and the age of coins are factored together to achieve a weight value, and the highest value wins the auction. At this point, the process of gaining maturity and weight resets.

4.2.4. Proof of Capacity

The fourth component of the SPS will be Proof of Capacity, which proves through advanced mathematical functions how much storage capacity a user has available, and it generates tokens accordingly. Their storage space is filled with plot points in individual encrypted blocks, and hashes are produced from these blocks. Similar to PoW, in which more hashing power solves more blocks, the more space is used for PoC, the more blocks can be solved. This method will grant to users of the KC a very low barrier to entry, since practically everyone has at least a small amount of free hard drive space, and it will produce a more egalitarian distribution of hashing power.

4.3. The Blockchain Miner

4.3.1. Dynamic Code Generation

The process of blockchain mining is performed by an application called a miner. Normally, these miners must be optimized for the target platform to achieve the greatest efficiency, but this involves the creation and maintenance of large code libraries for various technologies like OpenCL, CUDA, AVX, SSE, and Neon. Instead of this, Krate will use ArrayFire, a high-performance software library that performs dynamic code generation, in order to remove the hassle of writing difficult, low-level device code. It will allow the miner to utilize optimized instructions on the fly by feeding hashing functions as vector equations into ArrayFire.

4.3.2. Performance

A well-known drawback to using a high-level language like C# or a framework like .Net Core, both of which are used in the KC, is that performance rarely matches that of applications written in lower-level languages such as C/C++. To remain platform agnostic without sacrificing speed, the KRATE Blockchain's miner will use a .Net wrapper for the ArrayFire C/C++ library. This also gives the miner platform-agnostic characteristics.

4.4. Transaction Life Cycle

4.4.1. Components

A number of components will be involved in the life cycle of transactions in the KRATE Cloud:

1. user memory pool (mempool);
2. peers' mempools;
3. the KRATE Virtual Machine;
4. storage, compute, and network resources of peers' devices; and
5. the Sequenced Proof System (KHG, PoW, PoS, PoC);
6. UpscaleDB.

4.4.2. Verification Process

Users on the KRATE Network will send their unverified transactions to each other as part of the consensus process, and these transactions will be temporarily stored in their local mempool. From the mempool, they will be sent first to KHG in the SPS, which will serve as a first level of validation to filter out fraudulent or invalid transactions. If KHG verifies the transaction, it will be sent on to the KRATE Virtual Machine, which is a decision-making unit that will route transactions either:

A. to peer devices, if it includes a work order that requires storage, compute, or network resources; or
B. back to the SPS to run the gauntlet of PoW, PoS, and PoC if it includes payment data.

Transactions that include both a work order and payment data will be split by the KVM, and each part will be routed accordingly.

In case A., a report on the results of the work order will be passed from the KVM to the audit log of the smart contract that initiated the transaction[1], similar to a clerk posting to an account record. A new payment transaction will then be generated by the KVM to claim compensation for completing the work order. This new transaction will be broadcasted out to the network to be verified in its own transaction cycle.

In case B., transactional data sent by the KVM directly to the SPS will be required to pass a second validation stage, the three-part proof gauntlet, which will be a more robust transaction filter than KHG. Once this type of transaction passes the SPS, it will carry cryptographic proof of its validity and will be broadcasted out to the network for its own verification round. Transactions will be finalized only when they reach UpscaleDB; everything before that point will happen within the system's memory. Anything committed to UpscaleDB will be transmitted to the network as an update to the blockchain.

[1] *This is explained in Section 5.4. "KRATE Smart Contracts".*

4.4.3. Transaction ZKPs

No user data, including but not limited to multimedia files, documents, records, databases, etc., will be stored on the chain but rather in storage space provided by KRATE users through the Storage Network. Instead, independently verifiable ZKPs[2] will be housed in the blockchain that will provide an auditable record of transactions. ZKPs will make it redundant to retain the transactional data itself, so this data will be deleted after its corresponding ZKP is finalized in the blockchain.

[2] *ZKPs are explained in Section 7.8. "Zero-Knowledge Proofs".*

5. Why a New Blockchain?

The long-term viability of any blockchain-based project is based on its security and integrity. As the entire KRATE Cloud ecosystem relies on its blockchain to function, the safety of the KRATE Blockchain is of paramount importance.

5.1. Attack Resistance

There is a wide array of attacks that may be leveled against a blockchain, including:

- Network Attacks:
    - Proof of Work Attacks:
        - Majority Attack
        - Sybil Attack
        - Selfish Mining
        - Denial of Service Attack
        - Eclipse Attack
        - Time-Jacking
    - Proof of Stake Attacks:
        - Fake-Stake Attack
        - Nothing at Stake Attack
        - Long Range Attack
    - Proof of Capacity Attacks:
        - Burst Attack (message bomb)
        - Time-Memory Trade-off Attack
- Miner Attacks:
    - Finney Attack
    - Race Attack
    - Withholding Blocks
- Mining Pool Attacks:
    - Hostile takeover:
        - Majority Attack
        - Sybil Attack
        - Selfish Mining
        - Other hashpower attacks
    - Cannibalizing pools
- Internet Service Provider (ISP) Attacks:
    - Routing Attack
    - Partition Attack
    - Delay Attack
- Server Attacks:
    - DoS Attack
    - DDoS Attack

- P+Epsilon Attack
- Transaction Malleability Attack
- Wallet Attacks
    - Electrum Wallet DoS Attack
    - User Wallet Attack

Many of these attacks (most often a majority attack) have resulted in illegitimate coin-minting, prices tanking, networks clogging, exchanges delisting the project, or the project completely ending. A non-exhaustive list of blockchains that have fallen prey to such attacks include:

- Ethereum Classic (majority attack)
- Verge (majority attack)
- XPcoin (DoS attack, Fake-stake vulnerability)
- PeerCoin (Fake-stake)
- Qtum (Fake-stake)
- BitcoinGold (majority attack)
- Monacoin (majority attack)
- ZenCash (majority attack)
- Litecoin Cash (hostile takeover)
- FLO (majority attack)
- HTMLcoin (Fake-stake)
- Bitcoin (DoS vulnerability)
- Etherbase Coin (contract-swap vulnerability)

However, the KC's novel method of block validation, the Sequenced Proof System, will be capable of defending against all known attacks. Very simply, attacks against one type of proof are useless against another, since the mechanics of each proof are unique (PoW deals with computational power, PoS deals with cryptocurrency, PoC deals with digital storage space, and Hashgraph deals with transaction signatures), so the SPS will require each block to be confirmed by all four proofs. Not only will this ensure that one proof does not outpace the other, leading to unfair minting distributions and network insecurity, it will also increase the amount of network control needed to successfully launch a majority attack. To achieve this attack in a blockchain that uses either PoW, PoS, PoC, or Hashgraph by itself, a bad actor must control 51% of the network (67% with Hashgraph), but the KRATE Blockchain will require majority control of each proof simultaneously—in other words, a supermajority of the network. This nested proof structure is the key to achieving truly secure network decentralization.

5.2. Quantum Resistance

The KRATE Cloud will utilize lattice-based cryptography. One of the attractive characteristics of this type of cryptography is that it appears likely to succeed in defending against attacks in a post-quantum setting. This is important because the advent of quantum computing is on the horizon and it appears likely to bring cataclysmic change to the world of digital security, since it will utterly destroy the foundations of traditional cryptography by providing enough computational power to brute force their defenses. In this situation, the proof security of blockchains is a double-edged sword: in the same way that it prevents adversaries from altering the blockchain by requiring the expenditure of enormous amounts of compute for each block they target, it is also required in order to change the blockchain's cryptographic algorithm throughout the chain's history. This means that existing blockchains will be completely stripped of their security by QC if they haven't been built on lattice cryptography. Therefore, it is necessary for new blockchains to be designed with a post-quantum setting in mind, as the KC has.

5.3. Privacy-First Hybrid

The KRATE Blockchain is a public/private hybrid chain utilizing zero-knowledge proofs and a dual-key system consisting of a spend and a view key. The spend key cannot be used to view transactions and vice versa, making it necessary to have both the spend and view key to create a valid transaction. View keys can be published to the network for organizations requiring full transparency or given to select individuals when oversight and auditing processes are required. For additional token security, neither key can be derived from the other, and it is possible to create either kind as a multi-signature key.

5.4. KRATE Smart Contracts

Contracts and coding are both structured on logical systems, which allows contracts to be easily translated into code that manages blockchain transactions, removing the need for a trusted middleman. Each time a new transaction on the KRATE Blockchain occurs, it will be incorporated into a smart contract that is created by the Smart Contract Module and is written to the blockchain. These smart contracts are transaction protocols that execute the terms of their own contract and facilitate various tasks, such as simple exchanges—x units of y for x units of—but they are capable of much more. If a geneticist, for example, requires access to an enormous amount of computing power in order to perform protein folding, they will be able to obtain those resources from the KC by submitting a work request. The details of the work to be performed will form the substance of the smart contract, which will split such a complicated task into numerous work units and distribute them around the network to be accomplished by available nodes. The smart contract will assign, schedule, and monitor task progress, and then relay cryptographic proof of completed work units as they accomplished their assignments. The smart contract will finish by presenting a report on the results of work requests, which will show in human-readable language what work was completed, how and when it was completed—everything that happened—allowing the results to be audited.

This will be game changing because it will drastically speed up the process of seeking legal recourse for incomplete or unsatisfactory work, since there can be no dispute over what was or wasn't done according to the terms of the agreement.

Smart contracts offer an excellent solution to the challenge of inter-entity reconciliation, as in the case of financial institutions that transact with foreign businesses or banks. For example, if someone in Brazil wants to bid on a Canadian job, a smart contract can be set up so that it will control the money involved and use it as collateral to generate a letter of credit, then automatically move it to the workers in Canada in increments as milestones are reached. The smart contract itself will hold everything in escrow without bias, conflict of interest, or impeding factors like national stability, and will solve the concern of credit-worthiness.

Before users submit data that is to be used for a specific purpose, smart contracts will be responsible for notifying them of any minimum holding periods that legally prevent them from deleting their data within a certain period of time.

With the help of McKercher LLP, the KC will provide smart contract templates for various types of legal documents. Lawyers will need to know only the fields to confirm that the contract is correct, so that programmers won't have to create a smart contract from scratch every time one is required for a new arrangement.

5.5. Non-Linear Block Validation

The linear structure of a blockchain is a beneficial characteristic in terms of security, but this has also traditionally forced applications to validate a blockchain’s history in a linear fashion, examining the chain one block at a time, since multicore and multiprocessor systems are bound to a single active thread. Validating a blockchain this way can take hours or even days, but Krate is solving this problem by implementing non-linear block validation, a proprietary technique it has developed. The KRATE Blockchain will introduce a small change to each block’s header: a field for the hash of the current block’s header, one for that of the previous block, and one for the current block’s full data without its header. This means that the KCCA will be able to speedily validate block headers by themselves. Then, as this process continues, the rest of the system’s processors will be able to engage in fully validating the blocks that have been prepared in this way. As a result, peers on the blockchain will be able to fully utilize the capability of their hardware without sacrificing linear blockchain security, and, rather than assuming that their copy is correct, they will be able to fully validate the chain’s history with minimal delay at start-up every time. Non-linear block validation will also allow nodes with limited storage space to retain only a portion of each block in its history without sacrificing its ability to validate blockchain transactions with the same confidence as nodes storing the full blockchain history.

5.6. DIFFr

To combat gamification of difficulty algorithms (applying the elements of gameplay to other activities in order to gain an advantage), the difficulty for each block in the KRATE Blockchain will be based on a timestamp derived from its blockhash. Miners will race against each other to produce a blockhash that generates a timestamp closest to the current network time, which will run on Unix time and provide a reference for each timestamp. A candidate block will need to have a timestamp later than the current network time; once the network time matches the block’s timestamp, the block may be accepted. The competition will be won by the first block to have a timestamp that the network time catches up to.

5.7. Egalitarian Proofs

Many blockchains suffer from a centralization of mining power, especially in Proof-of-Work networks like Bitcoin, as a result of never-ending competition to exceed difficulty requirements and increase computational efficiency in specially designed mining hardware, such as those based on FPGA or ASIC technology. This quickly prevents the average miner with average equipment from contributing to such networks, but Krate will prevent this from happening to the KC by implementing an egalitarian computing algorithm known as Argon2+MTP. This algorithm is a compound of Argon2, a hashing algorithm, and Merkle Tree Proofs (MTP), a cryptographically linked categorization method similar to a family tree. The KRATE Blockchain will use it to produce hashes from block and transaction data in the same manner that SHA-256 is used in Bitcoin and its derivatives. It will also make it possible for a more inclusive group of users to contribute to blockchain security by equalizing computation time between platforms that have unequal computational power. This will mean that no single device should gain a significant advantage over another at any given price-point, thus leveling the playing field.

The egalitarian properties are a result of how the Argon2 hash algorithm processes data. Argon2 a memory-hard hashing scheme optimized for a) multi-core processing; b) the organization of Intel and AMD's cache and memory; and c) x86 architecture. More resilient than its competitors and comparable in performance, it is designed for a high memory-filling rate (up to 1 GB of RAM in well under a second) that simultaneously uses its multi-core capabilities in a way that resists trade-off attacks (an exploitation of the fact that cryptographic problems are vulnerable to a memory-time trade-off when they seem to require large amounts of memory). It was designed primarily to maximize the cost of exhaustive searches on architectures other than x86, thereby removing any advantage gained by using dedicated ASICs. Argon2 has several variants that provide different trade-offs: Argon2d, Argon2i, and Argon2id. Argon2d uses data-dependent memory access and is faster, but adversaries are able to run side-channel timing attacks and password garbage-collection, while Argon2i uses data-independent memory access and is slower in order to protect against side-channel attacks by making more passes over the memory. The former is suitable for use with cryptocurrency applications and backend servers, while the latter is preferable for password hashing and password-based key-derivation in insecure settings. However, Argon2id combines the best of both and has been selected for use in the KC.

5.8. Integrated Pool

Another source of centralization of power is mining pools. The KRATE Blockchain is capable of performing pool operations for a group of miners who wish to pool their hash power. As the pool operates on the network itself, there is no centralization of mining power and no opportunity for pool administrators to skim rewards.

5.9. Economics

Blockchain solutions that are truly public and permissionless and that include an integrated cryptocurrency coin or token have always struggled with price volatility. The same is also experienced by utility tokens, usually due to runaway inflation caused by early adopters dumping the token when its emission curve begins to plateau. Since the KRATE Token will face this challenge as well, it is imperative that the economy of the KRATE Cloud be designed to handle it. It is possible to reverse the emission curve in order to start out with the maximum supply of tokens, then shrink it by burning transaction fees in order to diminish the total supply and increase scarcity over time, but the efficacy of this solution is not clear, as it hasn't been tested by a sufficient number of projects. Another option is to peg a token to an asset like fiat currency, precious metals, or gems, but the financial securities label that it carries is a significant hindrance to its adoption and fungibility.

5.9.1. Decentralized Economic Growth Algorithm

Krate's solution will be to implement a circular economy, which is a concept that has been applied to hybrid blockchains (private/public and permissioned/permissionless) but not, to the best of our knowledge, to a public and permissionless blockchain. Users will earn KRATES by providing storage, compute, and network resources and will earn block rewards by securing the KRATE Blockchain when they operate masternodes or mine blocks through PoW, PoS, or PoC. Fees for cloud resources will be managed by smart contracts that will adjust the rate based on factors like quantity, as well as minimum requirements for security, fault tolerance, distribution, and performance like minimum bandwidth, maximum latency, and node reputation. To avoid runaway inflation, block rewards will be paid with transaction fees. Unlike other blockchains that have adopted this idea, the KRATE Blockchain will not pay winning miners all of the transaction fees in a block but will rather pool them. This will allow the KC's Decentralized Economic Growth Algorithm to increase and decrease block rewards based on how much the token is being used.

In order to keep inflation under control, the KC's economic state will be continuously analyzed by a protocol that tracks several different metrics, including Coin Days Destroyed (CDD), the number of transactions per block, the value of the transactions, and the number of addresses in the transactions. These metrics will determine how the pool of transaction fees should be divvied up for block rewards, which means that an increase or decrease in either the utility or accumulation of KRATES will increase or decrease the amount of rewards. Although they will have no supply cap, the amount of KRATES in circulation will be controlled by Krate.

The KC will have a very low barrier to entry in order to encourage users to join. Firstly, users will not be required to purchase tokens in order to begin earning them. Secondly, resource providers will be able to immediately contribute to PoW/PoS/PoC mining as an incentive both to bring their idle resources to the KC and to keep them there during times of low utilization. Thirdly, consumers and providers of cloud resources will benefit from the KC's economic system without learning the intricacies of cryptocurrencies, since all interactions with smart contracts and the KRATE Token will be handled in the background by the KRATE Cloud Client Application.

5.9.2. Primary and Secondary Markets

Krate will provide a primary market for KRATE Tokens. This will be necessary for several reasons: first, a primary market will satisfy regulatory requirements so that the KRATE will not be seen as a financial security, which could hinder adoption in select jurisdictions; second, it will solve the problem of realizing profits in fiat currencies without a secondary market (something no other blockchain has succeeded in doing) and will provide a stable source by which those currencies may be traded in exchange for KRATES; third, it will allow consumers to purchase cloud resources using traditional payment methods, such as credit/debit cards, PayPal, and others; fourth, it will allow cloud resource providers to be compensated in fiat currencies like CAD and USD; fifth, the primary market will be the key to the circular economy because tokens entering and exiting it will be purchased from and by Krate at a fixed rate (or may be exchanged for Krate-designed hardware or merchandise via Krate's online store).

It should be noted that it will still be possible for secondary markets to exist, such as those on cryptocurrency exchanges. Krate will neither initiate or prevent their creation, as this is a hard requirement imposed by regulators, but a third party may independently choose to start one. In this scenario, Krate will be able to provide paid technical support related to the integration of its software for such markets, as long as this is requested by the market's creator, and Krate will also be able to assist businesses with the process of accepting KRATES as an alternate form of payment for its products or services, since it isn't considered a speculative trade.

5.9.3. Equalization of Proof Profitability

The economics of the KC's Sequenced Proof System has been designed to avoid the manner in which hybrid proof systems behave in other blockchains, since they generally operate either in tandem or in a leapfrog manner that causes an inequality in their profitability. Ultimately, this leads to a) less-profitable proofs becoming a security liability; b) the entire chain stalling; or c) both outcomes. For this reason, the SPS will apply multiple proofs in a unique order by combining elements from series- and round-robin-ordering methods. The combination of the KC's economic controls and incentives, egalitarian mining algorithm, and unique order of proofs in the SPS is expected to equalize profitability across all proofs; as a failsafe, however, additional algorithms will be deployed in order to equalize rewards based on the effort required by each proof if the previous measures prove to be insufficient.

5.9.4. Dual Income Streams for Storage Providers

Users who provide storage space will have unique earning potential because they will be able profit from two streams of income. This is because encrypted user data will be indistinguishable from the high-entropy random data that will be involved in confirming PoC mining (it will be impossible to compress or deduplicate it, which will ensure that miners really are using their PoC mining space), and this will mean that such users will a) earn PoC rewards from the mining power of the storage they provide to the network; and b) earn storage fees according to the amount of storage space that is being rented from them by other users. For example, if a user provides 1 TB of available storage space that is actively storing 0.5 TB of user data, they will have 1 TB of PoC mining power with which to generate block rewards in addition to the rental income from hosting 0.5 TB of user data. This will solve the problem of attracting and keeping resources available on the network during the critical but low-utilization start-up phase.

5.9.5. KRATE Masternode Token

Blockchain projects that utilize masternodes frequently suffer catastrophic economic losses early on due to users hoarding tokens in a race to pay the required collateral to operate masternodes and then liquidating them after their profitability decreases for various reasons (the emission curve drops or the number of operators grows too large, etc.). These special nodes are desirable to own because they usually provide higher rewards in return for locking up tokens as collateral, which motivates the masternode operator to act in the best interests of the project. Unfortunately, there is always a tipping point where masternode profitability decreases enough to prompt their operators to dissolve them in an attempt to liquidate the tokens they hold on open markets. To avoid this, KRATE Masternodes will require collateral in the form of a Masternode Token that will be created from a one-way conversion of 10,000 KRATES. This will be accomplished using a one-way side-chain transaction that will inhibit operators from dumping large amounts of tokens into circulation, potentially destabilizing the circular economy and creating the need for secondary markets. Masternode Tokens will be managed by a merge-mined side chain running in parallel to the KRATE Blockchain, and it will be possible to sell them to Krate via the primary market but not to purchase them in order to ensure that the demand for more masternodes is satisfied by the conversion of KRATES into Masternode Tokens.

6. Masternodes

A masternode is a special type of node that is equipped to perform a number of important tasks on the KRATE Cloud.

6.1. Independent Arbiter

Their first function will be to act as a referee and independent witness that is able to authoritatively settle contract disputes. For example, if a storage host complains that data is held on its system erroneously or a user asserts that a storage partner is hosting data that it is not, masternodes will be able to inspect their metadata store to ascertain the truth and end the dispute.

6.2. Proactive Maintenance

Second, the KRATE Storage Network will use masternodes to periodically query storage hosts in order to track uptime. This cloud integrity scan will be a proactive measure necessary in order to weed out inactive nodes, thereby maintaining network health. As such, a masternode will consider a node as dead if it remains inactive beyond a certain interval of time and will use erasure coding to rebuild its missing data in order to redistribute it to other nodes (note that no information is exposed through this process, since the data remains encrypted). Dead nodes may be reactivated and will maintain their reputation, but their storage space and activity status will be reset.

6.3. Execute Repair

Third, masternodes will restore corrupted data if users are unable to repair it. After receiving a report about the corruption, masternodes will check whether the data was damaged in transit or at the storage node. If corrupted at the storage node, the cloud integrity scan will repair or replace the data. The KC will have a high level of fault tolerance (up to 30 nodes may fail before data will be affected), but if enough nodes are compromised and a user's data becomes inaccessible, it will compensate them for the time the data was not available with KRATE Tokens. Again, the KC will perform these functions without decrypting the data.

6.4. Reputation System

The fourth function of masternodes will be to maintain a reputation system in which nodes are ranked according to their uptime, reliability, and performance. This will allow for priority selection of trustworthy data hosts and will function much like an online auction site's feedback score but without the risk of inaccuracy from biased or fake reviews, since it will be handled by masternodes.

6.5. Network Routing

Lastly, masternodes will be responsible for routing data between user nodes.

Preventing resource consumers and providers from connecting directly to each other will eliminate the possibility of targeted attacks on individual nodes and prevent bad actors from being able to map data on the network.

6.6. Masternode Rewards

Masternodes are rewarded for providing their critical services to the KRATE Cloud. The rewards are variable and are based on the number of masternodes on the network versus the average amount of work completed by a single masternode. If there are more masternodes than required for the amount of work, rewards go down. If there are not enough masternodes, the rewards go up. Masternodes will also be subject to the reputation system and masternodes with a low reputation will receive a reduced reward.

7. Security

Ensuring the security of online data has proven to be an extremely difficult task. Even in the most secure centralized data warehouse, there are many ways by which a person or group can gain access to its information. Whether private data is leaked by a disgruntled technician, human error, digital or physical penetration of the storage site, or a government agency, breaches are a regular occurrence: in 2017 alone, for example, the Global Breach Level Index recorded over 1,700 breaches (shockingly, 9% of these were from malicious insiders) and over 2.6 billion records leaked globally. Additionally, files uploaded to a cloud storage service provider may exist on a variety of servers around the world, since these services use a degree of redundant availability, but the data may be stored in its entirety on a single hard drive under the control of a single entity. Given the stats, it's clear that a better approach to security is required. In this section, we will explore how the KC will provide its users with unparalleled security.

7.1. Security Principles

Krate's approach to securing the KC's software and hardware has been guided by two principles:

1. **The Rule of Two:** This principle dictates the use of two different and independent forms of encryption when securing important data. The KC will exceed this, however, as it will implement three levels of encryption: first, data input and output will be secured by either FHE or AES-256, as selected by the user[3]; second, the hard drive will be secured with Self-Encrypting Drive (SED) technology; and third, in addition to the SED, the file system can be protected with a variety of encryption algorithms (as determined by individual needs or corporate policy) that are supported by the file system itself. RAM, however, can be protected only by the hardware itself. On AMD Epyc Gen 2 systems, this is done by encrypting the RAM in its entirety, whereas Arm systems accomplish this by providing a portion of the RAM as a protected space for sensitive functions and data. Encrypting the network at the hardware level requires support on both ends of the cable, but this is difficult to achieve even with enterprise deployments.
2. **Avoid Assumptions:** Failure to observe this principle is one of the greatest pitfalls in security systems. For this reason, Krate will not assume that the user's hardware is trustworthy and secure. Instead, we will push down the root of trust—the lowest level of a process that you can trust or validate—all the way to the BIOS itself (on Krate-designed hardware) by fully leveraging the latest security features present in ARM and AMD CPUs. For example, a virtualized OS will be able to verify through the hypervisor if the host OS, bootloader, or BIOS has been tampered with. Likewise, the host OS will be able to detect alterations to the underlying hardware and will not decrypt any data on the device, which may stop the boot sequence from completing. For example, if the host OS successfully boots and is compromised, the kernel (a program that facilitates interaction between

[3] *The option to choose AES-256 for data input/out encryption will be offered because some clients will be prevented from using FHE for data storage due to legal reasons. Additionally, users will be able to disable FHE while computing in order to avoid its computational overhead if it isn't warranted.*

software and hardware components) would be compromised also and would be able, therefore, to intercept the private key. Even a fully compromised device would only place the user's own data at risk; it could continue to safely store and process other users' data, since it will remain encrypted and inaccessible without their decryption key.

7.2. Data Input/Output Security

For users who require advanced levels of security, Krate will sell custom hardware that is manufactured and configured to maximize the security provided by root-of-trust technologies in ARM and AMD Epyc Gen 2 processors. They will secure data input/output in the KRATE Cloud for the following components:

a. hard drive;
b. RAM;
c. CPU;
d. operating system;
e. bootloader;
f. firmware (i.e. the BIOS); and
g. network.

7.3. Securing Data in the Cloud

Users will be able to register for the KRATE Cloud through its user application, which will create a public key-pair for them to use on the Storage Network. After that, an encrypted entry will be created in an access control list, which will give the user fine granular control of private, shared, and public access to their files, and entries will be validated by a password-protected blockchain key-pair that will protect them even if the local machine is accessed. Files uploaded to the KC will be segmented into shards, and then each shard will be individually encrypted with the user's choice of AES-256, FHE, or both. Next, erasure coding will be performed so that lost or corrupted data can be rebuilt when detected by tiered checksums, and masternodes will oversee system integrity.

All data manipulation will occur on the user's device and will be encrypted before being sent in order to ensure that it will remain secure both in transit and at rest. The KRATE Cloud Client Application will perform the data manipulation in the following order: shard; compress; checksum; deduplicate; encrypt using AES-256, FHE, or both; shard again with random block size and order; encrypt again using Serpent (optional); apply erasure coding; apply tiered checksums; and broadly distribute. Each of these steps is defined below.

1. **First Sharding:** Files will be split into a number of smaller pieces called shards, which will allow for easier distribution of data and will add a layer of obfuscation.
2. **Compression:** Shards will be compressed using the LZ4HC compression algorithm.
3. **Checksum:** Shards will be SHA-256 checksummed for local data verification.
4. **Deduplication:** Based on block checksums, the client application will determine which data is a duplicate of previously stored data. Rather than storing identical data again, it

will add an additional pointer. Deduplication will happen only for data and storage spaces owned by the user, since the KC will not deduplicate data between storage spaces.

5. **First-Stage Encryption:** The KC will make it difficult to determine which shards belong to each other by encrypting each of them individually. Even if that were accomplished, a bad actor would still have to break the decryption of each shard before they could reassemble the file.
6. **Second Sharding:** A second sharding step will be performed on the encrypted data to prevent brute-force attacks. This sharding step will randomize the sizes of the second-stage shards (1MB, 512KB, 256KB, 128KB and 64KB) and will also generate a seed to create a randomized order of the shards, further adding to the possible permutations. By performing this step, it will be impossible to decrypt the data unless an attacker has all of the second-stage shards that belong to the first-stage shard. Even with them, attackers would need to try up to 6.57×10^{143} different permutations to find the correct order in which the second-stage shards need to be assembled before they could run the decryption process on the first-stage shard. The attacker would need to also repeat this process for every first-stage shard for the file to be correctly ordered. At this point, they would also need to break the base encryption in order to make sense of the data. In this way, the chain of encryptions will be virtually impossible to crack.
7. **Optional Second-Stage Encryption:** For those who require extra privacy, an additional layer of encryption will be able to be applied to each of the second-stage shards to make it even more difficult for an unauthorized party to gain access to their data.
8. **Erasure Coding:** Erasure coding is a technique that was developed for the sake of deep-space data communication from probes like the Voyager. Data transmitted by the Voyager was often degraded by photonic action, so it was necessary to ensure that partially destroyed information could be recovered. In its early implementation, NASA was able to accomplish this as long as they received at least half of the data; today, the technique is more advanced, and it will be leveraged by the KC.

   The client application will be able to build an entire file locally, even in the event of a partial or corrupted download, and users will be able to select from several levels of fault tolerance. The minimum level will be thirty, which means up to thirty shards can be missing or corrupt before the file is no longer recoverable.
9. **Tiered Checksums:** Corruption in data must be detected before Masternodes can use erasure coding to repair it, and this is the function of checksums. However, if only one checksum is used, it is uncertain whether the data in question or the checksum itself is faulty. When this occurs, most systems assume that the data is bad and immediately rewrite the original data, then compute the checksum once again. The KC will avoid this wasteful process by using a two-tier checksum verification system that will verify the integrity of both the original data and the checksum themselves.

   The tiered checksum procedure consists of a series of comparisons in which each piece of data generates two tiers of checks: first, the data is compared to the first checksum. If the data agrees with it, the data is assumed to be correct; if the data does not agree with it, one of the two is corrupt. To determine which it is, the first checksum is compared to the second checksum. If they agree, the first checksum is assumed to be correct, the data is flagged as corrupt, and it is scheduled for repair. If they do not agree, however, one additional comparison is made: a new first checksum is generated from the original data and checked against the original first checksum. If the new first checksum agrees with the original first checksum, the second checksum is corrupt rather than the data. Masternodes will be responsible for repairing both data and checksums, which they will do without involving the client's application and with zero knowledge of the data's contents, thanks to erasure encoding being applied after encryption.

10. **Broad Distribution:** After being sharded, checksummed, encrypted, and erasure encoded, the data will be distributed over the KC. If any node were breached, it would render only an encrypted sliver of an encrypted file useless without the rest of the file, the reassembly map, and the encryption keys.

When a user wishes to retrieve their data, this process will be reversed, with files being decoded, decrypted, reassembled, returned to the KC, and unlocked by their private key.

7.4. Data Controls

Broad distribution could present a problem to some users—especially businesses and governments that may want or be legally required to store data only in certain geographical regions—so the KC will provide the option of selecting or excluding specific areas. The distribution algorithm will also use region codes to prevent clumping of data in certain geographical regions or power grids, thereby maintaining the highest level of global availability and decreasing the potential for unavailability of data if nodes go offline.

The KC will group information into blocks of data that will have a minimum size of 100 MB, since small, frequently accessed files are easy for bad actors to map and locate by observing data flows on a distributed network. This will allow the KC to divide and disseminate each block a minimum of 100 times, guaranteeing that a bad actor will face a minimum difficulty in reconstituting the data even if they succeed in obtaining all of the shards. When users upload less than 100 MB at a time, the client application will pad the remainder; for example, if 45 MB of data were uploaded, the app would add 55 MB of filler data before processing it. There will be no way to tell how much real data a block contains, and this has the added benefit of making it difficult for bad actors to determine which blocks are worth targeting.

7.5. Brute Force and Two-Factor Authentication

As with many other great breakthroughs, distributed computing can be negatively exploited. Its awesome power presents a new method of executing brute-force attacks with sufficient strength to break through many of the common data security measures relied upon by internet services around the world. As distributed computing becomes increasingly common in services like the KRATE Cloud, it will be important for online data to be protected by additional safeguards, including location-based requirements, a limit on the number of an account's login attempts, stronger identification and authentication processes, and other such methods.

As such, the KC will offer two-factor authentication (2FA) functionality. Many security solutions allow users to generate 2FA codes from web browser plug-ins in order to increase accessibility, but it doesn't make much sense to do so on the same device you are logging in with. As such, the KC will require 2FA to be generated from a different device and will allow the user to see what device generated the 2FA code. Biometric authentication of only the face or a fingerprint are not sufficient, since you need only to kill or otherwise incapacitate a person to gain access to their security credentials. Krate will implement login security through a combination of something you have and something you know, like a debit card that requires a personal identification number. Additionally, Krate may consider providing support for pre-existing solutions like the Yubikey.

7.6. Remote Attestation

The KC's modularity may also be an attack vector because bad actors will likely attempt to design malicious apps that spoof legitimate ones. Unsuspecting users who download such imitations become vulnerable to all kinds of digital attacks, but identity technology offers a solution to this problem through the use of remote attestation, i.e., the ability to remotely attest the legitimacy of software and hardware. Thus, by cryptographically signing each app and module, the KC will be able to verify that its users have installed legitimate modules.

7.7. Fully Homomorphic Encryption

One of the most important aspects of the KRATE Cloud is that it will be the first commercially viable data storage solution to implement Fully Homomorphic Encryption. This technique is a monumental breakthrough for the needs of modern data management in many different applications, but especially in the case of cloud computing. Despite their convenience and usefulness, centralized clouds are quite insecure because they are designed to process data online. The problem with this lies in fact that encryption is a two-edged sword: bad actors can't understand the information contained within a ciphertext, but neither can the data owner unless they decrypt it. This forfeits the security provided by encryption and leaves it vulnerable to attack in online settings. However, FHE solves this problem because it allows data to be processed without decrypting it. For this reason, the KRATE Cloud will be the first cloud solution to provide convenience without sacrificing security. Furthermore, this merger provides one of the tools necessary to effectively combat identity theft and data breaches, since these abuses require access to vulnerable data; with FHE, no such vulnerability exists.

7.7.1. Homomorphic Encryption Schemes

An encryption scheme is classified as homomorphic if it allows computable functions to be performed on encrypted data while preserving both the function and format of the data's features. Beyond that, HE is divided into several different schemes that are distinguished by the type of functions that they allow and the number of times that they can successively run them. Partially Homomorphic Encryption (PHE) allows only one type of operation to be used limitlessly, while Somewhat-Homomorphic Encryption (SWHE) allows a few types of operations to be used a limited number of times, but FHE offers the ultimate cryptographic system: limitless computation of encrypted data with every type of function.

7.7.2. FHE Functions

The FHE technique involves mapping an encryption in abstract algebra, which allows a computer to work with the data through algebraic functions. As an example, imagine a client who needs to run analysis on some information contained in a sensitive document, which we define as *Data*. The client conceals the information with the *Encrypt*( ) function, i.e., *Encrypt*(*Data*) = *EncryptedData*, then sends *EncryptedData* to a cloud server. Next, the client processes this data online by querying the server with some function f( ), instructing the server to compute f(*EncryptedData*). The result of the query is *EncryptedResult*, which the server sends back to the client; once received, they use the *Decrypt*() function to run *Decrypt*(*EncryptedResult*) = *Result*, allowing them to reveal the hidden information in a safe, offline setting.

This example demonstrates that the server requires knowledge only of the function f—it only interacts with encrypted data and is not involved in either the encryption or decryption process—so the client is able to keep secret and inaccessible both the message and the private key that encrypts and decrypts the data. If an adversary were to gain access to the system and observe the communication between a client querying a server with a database encrypted with FHE, they wouldn't be able to guess how many matches or lack of matches were being made, since both matched and unmatched results are encrypted—even the system itself would be unable to tell whether or not a match was made.

7.7.3. Semantic Security

Additionally, FHE is semantically secure, which means that it can withstand the strongest cryptanalytic attack, the Chosen Plaintext Attack (CPA). The CPA is so formidable because it involves an attacker being able to obtain any plaintexts of their choosing together with the corresponding ciphertexts. If the encryption depends entirely on the plaintext itself, i.e., the letter "a" is always equal to the number "5", the same plaintext will always produce the same ciphertext, and that would allow the attacker to reverse engineer the decryption key by simply comparing multiple plaintexts and their encryptions. FHE achieves semantic security by introducing into its encryption algorithm a probabilistic element of randomness that enables it to produce from the same plaintext a different ciphertext each time it is used. This randomness comes from the noise parameter.

7.7.4. Noise

In FHE schemes, noise is meaningless information that is injected into a plaintext by the encryption algorithm, which conceals the original data and creates a ciphertext, and it is added in such a way that it can be easily removed by means of a mathematical key. However, this technique is challenging to use efficiently because performing successive computations on FHE ciphertexts causes the amount of noise in it to grow, and this quickly breaks the decryption algorithm's ability to properly recover the original information. This problem was first solved in 2009 by Craig Gentry through an inventive technique called "bootstrapping". This mechanism refreshed the ciphertext by reducing its noise to a manageable level, but the process was so computationally expensive that it could never be used in a real system. Since Gentry's seminal work, however, numerous advancements have brought FHE's real-life performance metrics within range of practical use.

7.7.5. Ideal Lattices

Several approaches to FHE have been proposed that differ in their mathematical foundation, including over integers, Ring-Learning With Error, and Nth Degree Truncated Polynomial Ring Unit systems, and ideal lattices. Krate's final step in completing its FHE implementation is to determine which approach will yield the most efficient performance for the KC, although ideal-lattices are the most likely candidate. Lattices are a linear combination of independent vectors called basis vectors, and each one may serve as a basis. They are designated as "good" or "bad" depending on how orthogonal they are (not orthogonal typically being bad), with greater goodness typically corresponding to vectors of greater length. Lattices provide a number of cryptographic problems—Closest Vector Problem (CVP), Shortest Vector Problem (SVP), and Lattice Reduction Problem (LRP)—that are exploited to create a public-key encryption scheme in which a good basis is sought for a lattice that is relatively short and orthogonal. A bad basis is used to create a public key and the good basis to create a private key. Solving LRP, which involves solving CVP and SVP, renders the closest lattice point and allows the ciphertext in question to be decrypted.

7.8. Zero-Knowledge Proofs

Beyond its significance to blockchains alone, quantum computing presents an unprecedented threat to data security in general, since it will easily break through the cryptographic mechanisms that currently secure online information of every kind—social, financial, medical, legal, governmental, or otherwise. For this reason, the US National Institute of Standards and Technology (NIST) is calling for the development of post-quantum cryptographic protocols. One such avenue is that of zero-knowledge proofs (ZKPs), a technique that will be integral to the KC. Though entirely different in function, ZKPs may sound somewhat similar to FHE: they are a mathematical construct by which a prover can undeniably prove to a verifier that a theorem is true without providing any information about it, and they can be used to create data management methodologies that are quantum-secure when they are constructed on lattice-based problems. Research into ZKPs and one-way crypto hashes using lattice-based encryption may provide the key to preserving the fundamental way in which we work with data, and it could save us the colossal amount of time, money, and effort that would be required to transition to a completely new cryptographic paradigm. Additionally, Krate is considering these techniques as potential avenues of research for other projects, so developing the KC will give Krate a head start on this subject.

7.8.1. Solving Intractable Problems

To explain zero-knowledge proofs, imagine a university server that operates a database x, which is a university database containing student information, a public function F, which is a machine-learning algorithm belonging to a researcher who will use it to compute some classifier y in a study, and a commitment *cm* to x that is publicly known. Say the researcher wants to obtain $y := F(x)$ and also has some pressing need to ensure computational integrity (that the server correctly produces the output, which would require unlimited access to x). Unfortunately, x contains sensitive personal information that cannot be publicly disclosed. This presents a real conundrum: the researcher doesn't have access to the input x, and the university is not willing to share it, even though it's willing to provide y. This would seem to present an intractable problem—if not for ZKPs.

7.8.2. Logical Properties

The power of any proof lies in the properties it possesses as a logical system. In this case, there are two properties that the researcher's proof must possess in order to succeed: the soundness property, which is attained if and only if every formula that can be proved in the system is logically valid with respect to the semantics of the system, and the zero-knowledge property, which is attained if and only if the verifier of the proof learns no information beyond the fact that the statement is true. Our researcher (the prover) therefore constructs a ZKP in an attempt to prove to the university (the verifier) that the following nondeterministic polynomial time (NP) statement is true: "there is $\tilde{x}$ (i.e., not-x) such that $y = F(\tilde{x})$, and $\tilde{x}$ is a decommitment of *cm*." This proof possesses the soundness property because it guarantees that, if the NP statement is false, the prover cannot successfully convince the verifier with high probability, thereby guaranteeing computational integrity for the researcher, and it possesses the zero-knowledge property because it guarantees that, if the NP statement is true, the prover can successfully convince the verifier without leaking any information about x beyond what is provided by y.

7.8.3. Bulletproofs

Early forms of ZKPs were impossible to use in applications like coding because of their length and complexity, but recent breakthroughs have produced a type of non-interactive zero-knowledge proof called Bulletproofs that are viable for the KC. These streamlined proofs are succinct (i.e., short and easily verified), they don't require a trusted setup, and they're extremely useful for creating decentralized electronic cash systems. This class of proof revolutionizes the way that data can be exchanged, and it can be used to create automated protocols that serve in place of human auditors to guarantee computational integrity of confidential data. In the KC, ZKPs will be implemented as an element of smart contracts that will allow anyone to verify important aspects of blockchain transactions without disclosing any information. As an example, if user *A* sends 10 coins to user *B*, third parties will be able to verify that a valid transaction occurred, that the coins transferred from user *A* did in fact belong to them, that no news coins were illegitimately minted, and other characteristics of the transaction, but they won't be able to learn who either party is, how many coins were transferred, what the addresses of the wallets are, or the purpose of the transaction. Transacting parties will be able to disclose this information if required to on a case-by-case basis.