Section 01

Abstract

NEAR is a decentralized application platform with the potential to change how systems are designed, how applications are built and how the web itself works.

It is a complex technology with a simple goal — allow developers and entrepreneurs to easily and sustainably build applications which secure high value assets like money and identity while making them performant and usable enough for consumers to access.

To do this, NEAR is built from the ground up to deliver intuitive experiences for end users, scale capacity across millions of devices and provide developers with new and sustainable business models for their applications. In doing so, NEAR is creating the only community-run cloud strong enough to extend the reach of Open Finance and power the future of the Open Web.

The following sections will describe the approach NEAR takes to designing and implementing the core technology of its system. Wherever possible, we will use language that is accessible and we will describe relevant sections starting from first principles, values and design intent before digging into their technical implementation. Additional depth on technical topics can be found in the relevant topic-specific papers and blog posts.

It must be noted that, as with all complex systems under active development, the contents of this guide and the technology they explain are both subject to change. In fact, one of the hallmarks of the NEAR approach is rapid and pragmatic iteration. The latest information about the protocol can be found in posts on the blog at https://www.nearpages.wpengine.com/blog, live chat channels at http://near.chat and in the reference code base at https://github.com/nearprotocol.

Section 02

Introduction

The richness of today's web emerged from the combined efforts of millions of people taking advantage of "permissionless innovation" — the ability to create content and applications without asking anyone first. Unfortunately, the lack of freedom for data has resulted in an environment which is actively adversarial to the interests of its participants.

A small number of companies have enticed vast numbers of users to join by luring them in with network effects and then captured them by holding their data to prevent them from seeking alternatives. Similarly, these massive platforms have enticed applications to build atop their ecosystems before either cutting off access or actively opposing their interests when the applications became too successful. As a result, these walled gardens have stifled innovation and effectively monopolized vast sections of the web.

In the future, we can fix this by using new technologies to re-enable the permissionless innovation of the past in a way which creates a more open web where users are free and applications are supportive rather than adversarial to their interests.

We have already seen the power of this kind of freedom in the financial sector, where decentralized digital currencies like Bitcoin and their underlying blockchain technology have facilitated billions of dollars of peer-to-peer transfers at a fraction of the price of the traditional banking system. The same underlying technology also allows participants in the $50B+ virtual goods economy to track, take ownership and trade these goods permissionlessly among themselves. It allows real world goods to cross into the digital realm, with verified ownership and tracking just like the digital ones.

Beyond what we've seen today, a web where freedom of data enables permissionless innovation by default will drive a new form of software development. In this web, developers can quickly construct applications from open state components and power their efforts with new business models which are enabled from within the software itself rather than rely on parasitic relationships with their users. This doesn't just accelerate the creation of applications which have a more honest and collaborative relationship with their users but it also allows for the emergence of entirely new businesses built on top of them.

These new applications and the open web that powers them can only be enabled by the right kind of infrastructure. The platform of the new web cannot be controlled by a single entity nor have its usage limited by insufficient scalability. It must be as decentralized in design as the web itself and supported by a widely distributed community of operators so the value it stores cannot be censored, modified or removed without the permission of the users on whose behalf that value is stored. It should be secure and stable enough to form the backbone of the new economy.

This is the infrastructure of NEAR.

NEAR is a decentralized application platform which is designed to enable the open web of the future and power its economy. It uses the same core underlying technology that made Bitcoin an unkillable currency and combines it with cutting edge advances in community consensus, database sharding and usability. On this web, everything from new currencies to new applications to new industries can be created, opening the door to a brand new future.

Why decentralization matters

On the surface, many of the design goals of a decentralized blockchain-based platform can be accomplished both faster and cheaper by using existing platforms.

For example, the cost to store data or perform computation on the Ethereum blockchain are between thousands and millions of times higher than the cost of performing the same functions on Amazon's Web Services. A developer can always build a "centralized" application or even a centralized currency at a fraction of the cost of doing the same on a decentralized platform because the decentralized platform will, by definition, have many redundancies in its processes and storage.

Why is it important to pay the added cost to support decentralization?

Because not all data is created equal.

Certain elements of value, for example the bits representing ownership of digital currency, personal identity or titles to assets, are highly sensitive. In a centralized system, the following players can all directly change the value of any balances they come into contact with:

1. The developer who controls the release or update of the application's code
2. The platform where the data is stored
3. The servers which run the application's code

Even if none of these players intend to operate with bad faith, the actions of governments, police forces and hackers can easily turn their hands against their users and censor, modify or steal the balances they are supposed to protect.

A typical user will trust a typical centralized application, despite its potential vulnerabilities, with everyday data and computation. Typically, only banks and governments are trusted sufficiently to maintain custody of the most sensitive information — balances of wealth and identity. But these entities are also subject to the very human forces of hubris, corruption and theft.

For example, the Global Financial Crisis in 2008 showed the fundamental problems of trusting an over-leveraged banking system. It also provided a timely example of how governments around the world implement substantial capital controls on citizens during times of crisis. Beyond this example, it has become a truism that hackers now likely own most or all of your sensitive data.

By contrast, a fully decentralized system doesn't have an "off" switch and it doesn't have a way for nefarious forces to impose their will on the applications built on top of it. To accomplish this, the system requires substantial redundancy in both computation and storage of data because any points of failure in these areas can be exploited. The more sensitive the information being stored, the more redundancy and security is required... and the more decentralization matters.

Blockchain-based systems are the substrate for this decentralization because their immutability provides the primitives — tokens, for example — necessary to incentivize cooperation and coordination among the numerous actors who make up these systems and power their redundancy. Once these systems are launched, they become essentially "unkillable".

The benefits of building applications on top of such a system are substantial. Not only is highly sensitive information secured and available globally, but currency is now a native primitive of the medium. These decentralized applications operate on a more complex infrastructure than today's web but they have access to an instantaneous and global pool of currency, value and information that today's web, where data is stored in the silos of individual corporations, cannot provide. As importantly, the data these apps secure is fully owned and controlled by their end users rather than the apps themselves. This opens up a wealth of new use cases which could not exist without a decentralized infrastructure.

While decentralization is crucially important, not all blockchain-based systems are decentralized. Decentralization is a scale which can be measured along a number of dimensions but, fundamentally, it comes down to how many players in the system must be corrupted in order to break the system itself and how likely that is to occur. The more important the assets the system must protect, the more important it is that true decentralization is achieved rather than a system which merely pays it lip service. Later sections will describe the technical architecture which achieves decentralization for NEAR.

A brief summary of NEAR

NEAR is a decentralized application platform which runs atop the NEAR Protocol blockchain. This blockchain, which runs across hundreds of machines around the world, is organized to be permissionless, performant and secure enough to create a strong and decentralized data layer for the new web.

Essentially, NEAR is a platform for running applications which have access to a shared — and secure — pool of money, identity and data which is owned by their users. More technically, it combines the features of partition-resistant networking, serverless compute and distributed storage into a new kind of platform.

For comparison, Amazon's Web Services and Microsoft's Azure operate much of the infrastructure of the web today and are two of the most common "clouds" where applications are deployed. Each of the individual servers which make up these computing and storage clouds are controlled by a single entity. This means that anything run on or stored within them is completely at the mercy of those companies or the government agencies which require them to do things against their will. Data can easily be lost, censored, altered, sold or hacked. This is because there is only a single point of failure.

Applications which are deployed to these cloud servers can also be continuously modified by their original developers or whoever holds their credentials. This makes software updates easy for developers but it also means that any data accessible by an application can be censored, modified or stolen by these same developers, whether at their own direction or because they were hacked or forced by a governmental authority to do so. Because user data is generally stored in large pooled databases, these developers become juicy targets for exactly those activities.

Together, the vulnerability of the developers and the platforms themselves makes any sensitive data stored on these platforms vulnerable.

When the cloud which hosts these applications is instead run by a global community which anyone can be a part of, the programs and assets stored by it become transparent and essentially "unkillable", allowing users to store meaningful things like money, identity and digital assets and securely transact them with anyone without requiring someone else's permission or platform. There is no single point of failure because there are multiple redundancies around the world and there is security because of the consensus which is programmatically achieved among community members who make up the cloud network.

To eliminate the vulnerability of the developer, applications deployed to this cloud can be programmatically locked so no further updates can modify the state they access. Essentially, once they achieve this state, they become autonomous and can be trusted to continue to perform their functions without fail or interference. This allows the secure storage of high value assets like money, identity and key pieces of data.

Bitcoin can be thought of as the first, very basic, version of this global community-run cloud, though it is primarily used only to store and move the Bitcoin digital currency.

Ethereum is the second and slightly more sophisticated version, which expanded the basic principles of Bitcoin to create a more general computing and storage platform, though it is a raw technology which hasn't achieved meaningful mainstream adoption.

NEAR represents an evolution beyond what has come before it and is the first decentralized application platform to solve all of the three key challenges to gaining mainstream adoption: usability, scalability, and security.

Challenges of Creating a Community Cloud

A community-run system like this has very different challenges from centralized "cloud" infrastructure which is run by a single entity or group of known entities. For example:

1. It must be both inclusive to anyone and secure from manipulation or capture.
2. Participants must be fairly compensated for their work while avoiding creating incentives for negligent or malicious behavior.
3. It must be both game theoretically secure so good actors find the right equilibrium and resistant to manipulation so bad actors are actively prevented from negatively affecting the system.

A free-market-based system is the only way to seamlessly align these incentives and so the NEAR Platform uses a token — also called "NEAR" — to glue it all together. This token allows the users of these cloud resources, regardless of where they are in the world, to fairly compensate the providers of the services and to ensure that these participants operate in good faith.

To remain decentralized, it's important that a community-run system like this be permissionless, meaning anyone has the opportunity to participate. To ensure this, anonymity is crucial and so revealing a party's identity is not required. While this provides for decentralization, it also opens up a wide range of misbehavior so all the mechanisms of the system must assume that one individual actor might be controlling a single account or a million accounts. Thus we operate with a principle of "one token equals one vote" to participate and govern the system.

These systems must also balance the need to be robustly decentralized with the reality that technologies and communities must be given the freedom to iterate or risk being rendered obsolete. Thus the long term health of the community requires maintaining a broad degree of decentralization and strong security guarantees in the platform itself while also creating efficient processes for evolving its technology over time.

Section 03

Why NEAR?

Today's blockchains have achieved significant progress — Bitcoin, the original blockchain which launched in 2008, is a store of value whose network has been priced at over $300 billion while Ethereum, the original "global computer" which launched in 2014, boasts thousands of innovative applications spanning from gaming to decentralized finance.

Unfortunately, neither these original networks nor any of those which followed have managed to bridge the gap towards mainstream adoption of the applications which are built on top of them nor provide the kind of scale which supports an entire Web.

This is a result of two key factors:

1. System design

2. Organization design

System design is relevant because the technical architecture of other platforms creates substantial problems with both usability and scalability which have made adoption nearly impossible by any but the most technical innovators. End-users experience 97-99% dropoff rates when using applications and developers find the process of creating and maintaining their applications endlessly frustrating.

Fixing these problems requires substantial and complex changes to current protocol architectures, something which existing organizations haven't proven capable of implementing. Instead, they create multi-year backlogs of specification design and implementation which result in their technology falling further and further behind.

NEAR's platform and organization are architected specifically to solve the above mentioned problems. The technical design is fanatically focused on creating the world's most usable and scalable decentralized platform so global-scale applications can achieve real adoption. The organization and governance structure are designed to rapidly ship and continuously evolve the protocol so it will never become obsolete.

The following section will highlight the key features which address these problems.

Key Features

Each of the key problems faced by current platforms, their developers and their end-users are addressed below. More information about the specific implementation of these features is left to the following sections of this paper.

Usability First

New blockchain-based platforms generally claim to differentiate themselves based on providing scalability relative to existing platforms. The scalability of a platform isn't relevant, however, unless the platform has sufficient adoption to require that throughput. As an analogy, it makes no sense to create an enormous stadium that can seat 100,000 people in the middle of the desert where no one wants to go in the first place.

Thus, the more important immediate problem to address is how to allow developers to **easily** create **useful** applications that users **can actually use** and which will **capture sustainable value** for those developers.

While some changes along these dimensions can be handled in the second layer of the technical stack, the most important ones must be made at the protocol level and cannot be bolted on afterward.

End-User Usability

Developers will only build applications which their end users can actually use. NEAR's "progressive security" model allows developers to create experiences for their users which more closely resemble familiar web experiences by delaying onboarding, removing the need for user to learn "blockchain" concepts and limiting the number of permission-asking interactions the user must have to use the application.

1. **Simple Onboarding:** NEAR allows developers to take actions on behalf of their users, which allows them to onboard users without requiring these users to provide a wallet or interact with tokens immediately upon reaching an application. Because accounts, which have human-readable names like foobar.near, keep track of application-specific keys, user accounts can also be used for the kind of "Single Sign On" (SSO) functionality that users are familiar with from the traditional web (eg "Login with Facebook/Google/Github/etc").
2. **Easy Subscriptions:** Contract-based accounts allow for easy creation of subscriptions and custom permissioning for particular applications.
3. **Familiar Usage Styles:** The NEAR economic model allows developers to pay for usage on behalf of their users in order to hide the costs of infrastructure in a way that is in line with familiar web usage paradigms.
4. **Predictable Pricing:** NEAR prices transactions on the platform in simple terms which allow end-users to experience predictable pricing and less cognitive load when using the platform.

Developer Usability

A number of key usability improvements support developers of the platform, allowing them to more easily learn, develop, test and deploy their applications than they can with any other platform.

1. **Familiar Languages:** NEAR nodes run Web Assembly (WASM), which can be compiled from a host of popular languages. Initially, smart contracts can be built using Rust, a very secure and comprehensive programming language that is rapidly gaining popularity. NEAR also supports contracts written in AssemblyScript which is very similar to TypeScript, a Microsoft-developed modification of JavaScript that has types and a very broad adoption among developers. In the future, more common programming languages will be supported so developers don't have to learn an entirely new language to build applications atop the platform.
2. **Robust Tooling:** NEAR's development suite is created to support the developer workflow with a unified set of tools so developers can easily build, test and deploy applications. The tooling on top of the platform and the APIs exposed by it provide developers with the kind of development experience they are used to from traditional web apps. This includes one-click deploy, integrated unit testing, easy front-end integration and debugging from the web browser's developer console.
3. **Developer Business Models:** The NEAR Protocol supports developers by helping them monetize the open components they create for the ecosystem by natively rewarding them with rebates based on the usage of those components. This is addressed specifically in a following section.
4. **Predictable Pricing:** NEAR prices transactions on the platform in simple terms which allow the developer to experience predictable pricing and less cognitive load when using the platform.

Scalability Second

A future-proof protocol must shard both state and processing in order to scale. With significant adoption of the platform, no single machine would otherwise be capable of storing all the information on the chain or verifying all of the transactions.

NEAR uses a sharding approach which allows the network to increase its capacity as additional nodes participate. This is done by dynamically splitting the network nodes into multiple shards when usage is high enough to require it and parallelizing computation across those shards. With this approach, the network can scale continuously as demand increases.

A lot of recent sharding research in the blockchain community separates transactions into intra-shard and cross-shard categories, optimizing for the former and providing a much slower solution for the latter. *The NEAR Protocol assumes that transactions will touch multiple shards by default*, which is the likely behavior for arbitrary smart contracts, and optimizes performance accordingly.

Efficient Development and Evolution

A key challenge facing both existing and new platforms is how they handle development and evolution of the platform. While the platform itself must be decentralized, there are multiple approaches to updating and evolving it.

Existing protocols have proven insufficiently iterative to keep up with the pace of innovation and a successful next-generation protocol must ensure that the network maintains its decentralization while still allowing for an efficient development process which prevents it from being disrupted by the next wave of technology.

NEAR's initial development is being done by one of the strongest teams of engineers, entrepreneurs and technologists in the world and its governance is designed to ensure that the protocol is developed subject to ongoing community oversight but with sufficient efficiencies in the process that it will remain competitive and relevant long after its launch.

This ensures that NEAR will not only avoid the "failure to launch" problem which has plagued one half of the industry but also the "failure to improve" problem which has held back the other half.

Real Decentralization

Even though Bitcoin and Ethereum are often lauded for their level of decentralization, they actually suffer from many centralization issues. Bitcoin, for example, has 53% of its mining power controlled by just three pools. On top of that, running mining nodes currently requires expensive hardware, which increases barriers to entry and reduces the incentives for nodes to join over time.

Newer networks often trade the hope of decentralization for the operational efficiencies provided by more centralized implementations which use either limited validator sets or fully "permissioned networks". This violates one of the fundamental tenets of a truly decentralized network — that its value is protected by its level of redundancy among independent nodes.

In order to maintain real decentralization, the network needs to allow permissionless participation from prospective node operators and not incentivize pooling. To address these concerns, NEAR uses a staking mechanism called "Thresholded Proof of Stake" which is specifically designed to be both deterministic and broadly fair so it doesn't incentivize pooling of large validators and it encourages broad participation from nodes.

Lowering the barriers to entry for nodes accomplishes more than simply decentralizing the network. In a horizontally scaling system like NEAR's, the more nodes which can participate, the more it can scale as well.

A New Business Model for Developers and Entrepreneurs

An early use case for blockchains like Ethereum was allowing projects to create their own tokens and raise funds through "Initial Coin Offerings" (ICOs). This initially appeared to be a revolutionary new way of allowing infrastructure developers to access capital and to bootstrap network effects for their projects, something which had long been lacking in the world of open source software and infrastructure. Unfortunately, creating application-layer tokens also put major usability hurdles in front of users and the willful speculation and fraud that subsequently occurred made it clear that this was not a viable path forward for most developers.

NEAR provides developers and entrepreneurs with a more robust, less intrusive and more legitimate way of monetizing their infrastructure. When a contract is called, a portion of the fees generated by the network are automatically allocated to that contract and might (if coded as such) be withdrawn by its developer. This both incentivizes early infrastructure development (because the early contracts will build network effects that increase usage) and provides a business model so application and infrastructure developers can benefit from their creations without creating ill-advised tokens of their own.

Section 04

Design Principles

Both the design and development of the NEAR platform are guided by a handful of key principles. These principles reflect the problems inherent in both the centralized and decentralized systems of today.

1. **Usability:** Applications deployed to the platform should be seamless to use for end users and seamless to create for developers. Wherever possible, the underlying technology itself should fade to the background or be hidden completely from end users. Wherever possible, developers should use familiar languages and patterns during the development process. Basic applications should be intuitive and simple to create while more robust applications should still be secure.

2. **Scalability:** The platform should scale with no upper limit as long as there is economic justification for doing so in order to support enterprise-grade, globally-used applications.

3. **Simplicity:** The design of each of the system's components should be as simple as possible in order to achieve their primary purpose. Optimize for simplicity, pragmatism and ease of understanding above theoretical perfection.

4. **Sustainable Decentralization:** The platform should encourage significant decentralization both in the short term and the long term in order to properly secure the value it hosts. The platform — and community — should be widely and permissionlessly inclusive and actively encourage decentralization and participation. To maintain sustainability, both technological and community governance mechanisms should allow for practical iteration while avoiding capture by any single parties in the long run.

Section 05

How NEAR Works

NEAR provides a community-operated cloud infrastructure for deploying and running decentralized applications. It combines the features of a decentralized database with others of a serverless compute platform. The token which allows this platform to run also enables applications built on top of it to interact with each other in new ways. Together, these features allow developers to create censorship resistant back-ends for applications that deal with high stakes data like money, identity and assets and open-state components which interact seamlessly with each other.

These application back-ends and components are called **"smart contracts,"** though we will often refer to these all as simply "applications" here.

The infrastructure which makes up this cloud is created from a potentially infinite number of "nodes" run by individuals and organizations around the world who offer portions of their CPU and hard drive space — whether on their laptops or, more likely, professionally deployed servers. Developers write smart contracts and deploy them to this cloud as if they were deploying to a single server, which is a process that feels very similar to how applications are deployed to existing centralized clouds.

Once the developer has deployed an application, called a "smart contract", and marked it unchangeable ("immutable"), the application will now run for as long as at least a handful of members of the NEAR community continue to exist. When end users interact with that deployed application, they will generally do so through a familiar web or mobile interface just like any one of a million apps today.

In a centralized cloud hosted by Amazon or Google, developers pay for their applications each month based on how much usage they required, for example based on the number of requests generated by users visiting their webpages. The NEAR platform similarly requires that either users or developers provide compensation for their usage to the community operators of this infrastructure. Like today's cloud infrastructure, NEAR prices usage based on easy to understand metrics that aren't heavily influenced by factors like system congestion. Such factors make it very complicated for developers on alternative blockchain-based systems today.

More details on the economics of NEAR can be found in the Economics section.

In a centralized cloud, decisions are made unilaterally by the controlling corporation. The NEAR community-run cloud is decentralized so updates must ultimately be accepted by a sufficient quorum of the network participants. Updates about its future are generated from the community and subject to an inclusive governance process which balances efficiency and security.

More details of the governance process can be found in the Governance section.

In order to ensure that the operators of nodes — who are anonymous and potentially even malicious — run the code with good behavior, they participate in a staking process called "Proof of Stake". In this process, they willingly put a portion of value at risk as a sort of deposit which they will forfeit if it is proven that they have operated improperly.

More details of the staking process can be found in the Technology section.

Elements of the NEAR Platform

The NEAR platform is made up of many separate elements. Some of these are native to the platform itself while others are used in conjunction with or on top of it.

The NEAR Token

NEAR token is the fundamental native asset of the NEAR ecosystem and its functionality is enabled for all accounts. Each token is a unique digital asset similar to Ether which can be used to:

1. Pay the system for processing transactions and storing data.
2. Run a validating node as part of the network by participating in the staking process.
3. Help determine how network resources are allocated and where its future technical direction will go by participating in governance processes.

The NEAR token enables the economic coordination of all participants who operate the network plus it enables new behaviors among the applications which are built on top of that network.

Other Digital Assets

The platform is designed to easily store unique digital assets which may include, but aren't limited to:

- **Other Tokens: Tokens bridged from other chains ("wrapped") or created atop the NEAR Platform can be easily stored and moved using the underlying platform. This allows many kinds of tokens to be used atop the platform to pay for goods and services. "Stablecoins," specific kinds of token which are designed to match the price of another asset (like the US Dollar), are particularly useful for transacting on the network in this way.**
- **Unique Digital Assets:** Similar to tokens, digital assets (sometimes called "Non Fungible Tokens" (NFTs) ranging from in-game collectibles to representations of real-world asset ownership can be stored and moved using the platform.

The NEAR Platform

The core platform, which is made up of the cloud of community-operated nodes, is the most basic piece of infrastructure provided. Developers can permissionlessly deploy smart contracts to this cloud and users can permissionlessly use the applications they power. Applications, which could range from consumer-facing games to digital currencies, can store their state (data) securely on the platform. This is conceptually similar to the Ethereum platform.

Operations atop the platform which require computation, network usage or storage require payment to the platform in the form of fees which the platform then distributes among its community of validating nodes. These operations can include the creation of new accounts, the deployment of new contracts, the execution of code by a contract and the storage or modification of data by a contract.

Details of these costs is laid out in the Economics section. Details of how the nodes work are provided in the Technology section.

The platform can be interfaced with permissionlessly. As long as the rules of the protocol are followed, any independent developer can write software which interfaces with it (for example, by submitting transactions, creating accounts or even running a new node client) without asking for anyone's permission first.

The NEAR Development Suite

The NEAR platform is designed to be used independently and permissionlessly but a set of tools and reference implementations is being created to facilitate its use by those developers and end users who prefer them. These tools include:

- **NEAR SDKs:** NEAR supports Rust and AssemblyScript (JavaScript with types) languages to write smart contracts. To provide a great experience for developers, NEAR has a full SDK which includes standard data structures, examples and testing tools for these two languages.
- **Gitpod for NEAR: NEAR uses existing technology Gitpod to create zero time onboarding experience for developers. Gitpod provides an online "Integrated Development Environment" (IDE), which NEAR customized to allow developers to easily write, test and deploy smart contracts from a web browser. The NEAR Examples website contains templates that can be deployed in one-click to make the process of building on NEAR for both new and old developers as simple as possible.**
- **NEAR Wallet:** A wallet is a basic place for developers and end users to store the assets they need to use the network. NEAR Wallet is a reference implementation that is intended to work seamlessly with the progressive security model that lets application developers design more effective user experiences. It will eventually include built-in functionality to easily enable participation by holders in staking and governance processes on the network.
- **NEAR Explorer:** To aid with both debugging of contracts and the understanding of network performance, Explorer presents information from the blockchain in an easily digestible web-based format.
- **NEAR Command Line Tools:** The NEAR team provides a set of straightforward command line tools to allow developers to easily create, test and deploy applications from their local environments.

All of these tools are being created by the community in an open-source manner so they can be modified or deployed by anyone.

Section 06

Economics

The ecosystem which makes up the NEAR platform is driven primarily by economic forces. This economy creates the incentives which allow participants to permissionlessly organize to drive the platform's key functions while creating strong disincentives for undesirable, irresponsible or malicious behavior. In order for the platform to be effective, these incentives need to exist both in the short term and the long term.

Fundamentally, the NEAR platform is a marketplace between willing participants. On the supply side, operators of the validator nodes and other fundamental infrastructure need to be

incentivized to provide these services which make up the “community cloud.” On the demand side, the developers and end-users of the platform who are paying for its use need to be able to do so in a way which is simple, clear and consistent so it helps them.

Further, economic forces can also be applied to support the ecosystem as a whole. They can be used at a micro level to create new business models by directly compensating the developers who create its most useful applications. They can also be used at a macro level by coordinating the efforts of a broader set of ecosystem participants who participate in everything from education to governance.

The specific application of each of these forces is described in the sections below. We will begin by benchmarking how each of the key design principles of NEAR apply to its economics and survey the landscape of existing approaches.

NEAR Economy Design Principles

NEAR’s overall system design principles are used to inform its economic design according to the following interpretations:

1. **Usability:** End users and developers should have predictable and consistent pricing for their usage of the network. Users should never lose data forever.
2. **Scalability:** The platform should scale at economically justified thresholds.
3. **Simplicity:** The design of each of the system’s components should be as simple as possible in order to achieve their primary purpose.
4. **Sustainable Decentralization:** The barrier for participation in the platform as a validating node should be set as low as possible in order to bring a wide range of participants. Over time, their participation should not drive wealth and control into the hands of a small number. Individual transactions made far in the future must be at least as secure as those made today in order to safeguard the value they modify.

Overview

The NEAR economy is optimized to provide developers and end-users with the easiest possible experience while still providing proper incentives for network security and ecosystem development.

Here is a summary of the key ideas that drive the system:

1. **Thresholded Proof of Stake:** Validating node operators provide scarce and valuable compute resources to the network. In order to ensure that the computations they run are correct, they are required to “stake” NEAR tokens which guarantee their results. If these results are found to be inaccurate, the staker loses their tokens. This is a fundamental mechanism for securing the network. The threshold for participating in the system is set algorithmically at the lowest level possible to allow for the broadest possible participation of validating nodes in a given “epoch” period (½ of a day).
2. **Epoch Rewards:** Node operators are paid for their service a fixed percentage of total supply as a “security” fee of roughly 4.5% annualized. This rate targets sufficient participation levels

among stakers in order to secure the network while balancing with other usage of NEAR token in the ecosystem.

3. **Protocol treasury:** In addition to validators, the protocol treasury receives 0.5% of total supply annually to continuously re-invest into ecosystem development.
4. **Transaction Costs:** Usage of the network consumes two separate kinds of resources — instantaneous and long term. Instantaneous costs are generated by every transaction because each transaction requires the usage of both the network itself and some of its computation resources. These are priced together as a mostly-predictable cost per transaction, which is paid in NEAR tokens.
5. **Storage Costs:** Storage is a long term cost because storing data represents an ongoing burden to the nodes of the network. Storage costs are covered by maintaining minimum balance of NEAR tokens on the account or contract. This provides an indirect mechanism of payment via inflation to validators for maintaining contract and account state on their nodes.
6. **Inflation:** Inflation is the combination of payouts to validators and the protocol treasury minus the collected transaction fees (and a few other NEAR burning mechanics like the name auction. Overall, the maximum inflation is **5%**, which can go down over time as network gets more usage and more transactions fees are burned. It's possible that **inflation becomes negative** (total supply decreases) if there are enough fees burned.
7. **Scaling Thresholds:** In a network which scales its capacity relative to the amount of usage it receives, the thresholds which drive the network to bring on additional capacity are economic in nature.
8. **Security Thresholds:** Some thresholds which provide for good behavior among participants are set using economic incentives. For example, "Fishermen" (described separately).

The justifications for each of these principles is described in more detail in the following sections.

Resources Provided

A blockchain-based cloud provides several specific resources to the applications which run atop it:

- **Compute (CPU): This is the actual computer processing (and immediately available RAM) which run the code in a contract.**
- **Bandwidth ("Network")**: This is the network traffic between participants and users, including messages which submit transactions and those which propagate blocks.
- **Storage:** Permanent data storage on the chain, typically expressed as a function of storage space (eg kilobytes).

Existing blockchains like Ethereum account for all of these in a single upfront transaction fee which represents a separate accounting for each of them but ultimately charges developers or users for them only once in a single fee. This is a high volatility fee commonly denominated in "gas".

Developers prefer predictable pricing so they can budget and provide prices for their end-users. The pricing for the above-mentioned resources on NEAR is an amount that is slowly adjusted based on system usage (and subject to the smoothing effect of resharding when usage grew sustainably) rather than being fully auction-based. This means that a developer can more predictably know the cost of running transactions or maintaining their storage.

Initially, all of these resources will be priced and paid in terms of NEAR tokens. In the future, they may also be priced in terms of a stable currency denomination (for example a token pegged to the $USD).

Compute and Bandwidth ("gas")

Compute (CPU) is a momentary resource spent on executing a transaction. The cost of each CPU instruction is denominated in "gas" units and its price is determined based on the slowly adjusted price of gas (denominated in NEAR tokens). Bandwidth is usually measured in bytes, but in the NEAR platform, it is converted into gas units by using a simple coefficient of overhead which has been estimated on reference hardware.

Each chunk (a piece of a block) is given a specific maximum gas limit which is determined based on how much a single block can "fit" when executed on reference hardware. The block limit can also be adjusted by network participants in order to account for performance improvements or the participants deciding to run better hardware. This is done via the normal system governance processes.

The current gas price is predictable but not fixed. Each block, it is adjusted in the following way:

- If the prior block is more than half full, the gas price from the previous block is increased by an amount given by the parameter called `alpha`.
- If the prior block is less than half full, the gas price from the previous block is decreased by an amount also given by the parameter called `alpha`.

Gas usage can trigger Resharding, as explained below.

Storage

Storage is a long-term scarce asset. For an application or users to use it, they must maintain a minimum balance on their account that scales linearly with the amount of storage such account takes. The required amount of NEAR tokens per byte is fixed and is subject to change only by major governance decisions (and, given trends in storage hardware and system capacity, it will likely be adjusted down going forward).

For example, if some contract takes 10 kilobytes of storage due to data stored under it, this contract must maintain a minimum balance of 1 NEAR. A contract like **USDT** from Ethereum would need to maintain ~10k NEAR balance to cover for storage it is using. This also means that regular user's accounts need to maintain a fraction of the NEAR minimum balance.

Using a minimum balance of NEAR on the account leads to this amount not being staked or used in other applications. Validators are getting paid indirectly for maintaining this storage from inflation and the fact that the total stake is smaller.

The NEAR system will keep shard size mostly balanced, allowing for each node to maintain roughly the same amount of state (which will be roughly the total state size divided by the number of shards). As individual shard state size changes, the assignment of accounts and contracts to shards can shift to maintain this balance.

Resharding

Accounts and contracts are each assigned to a shard. Because the usage of such contracts is not equal, the usage or size of some shards might greatly outpace the usage or size of other shards. To prevent this, NEAR uses resharding, which rebalances shards periodically based on specific conditions. The end result will be a set of shards that are expected to have a reasonable and balanced amount of transactions and storage usage.

During each epoch (½ day), statistics are accumulated regarding how full blocks are during that epoch and other relevant conditions. Each contract is examined based on its usage during the previous epoch and its current storage. Contracts are then "bucketed" into groups such that each bucket has roughly the same expected aggregate characteristics.

The system knows approximate limits on transaction usage per shard (the "gas limit") as well as expected per-node storage. If the number of resources used in the previous epoch exceeded a particular threshold (eg if a large number of blocks are more than half full), a number of new shards will be allocated, increasing the number of buckets and averaging down each of their expected usages.

Adding new shards also means there will be more seats available for validation, which in turn brings more unique validators to the table as the per-seat price (in NEAR tokens) goes down. This computation is performed one epoch in advance of actually using these new shards so validators have the time they need to re-sync the necessary state and other shard information.

Inflation

The overall inflation (minting of new tokens) of the system is determined by how large the epoch reward is for running a validating node. The rate of minting of new tokens is capped at 5% per annum. The effective rate is computed per epoch (½ a day). It is calculated from the expected inflation rate per epoch minus the fees collected during the epoch. Each portion of fees captured by the platform is removed from inflation, thus reducing the overall inflation of the system as its usage increases. Should the system's usage fees reliably exceed the tokens generated by the inflation, it will become deflationary overall.

Because the system is sharded and has hidden validators whose assignment is unknown to the system, it must socialize all rewards. This means that the "security" fee is evenly allocated to all validators independent of how many transactions or storage their shards processed.

Economic Stakeholders

- **Validators: Provide the computational resource and security for the network by running nodes.**
- **Developers:** Create the applications which run atop the network
- **Token Holders:** Accounts or applications which maintain token balances

- **NEAR Foundation:** An independent entity which coordinates the governance and technical evolution efforts of the network participants.
- **Third Party Observers:** The observers of the chain who provide extra fraud and bad behavior protection.
- **Users:** Users of applications on the network who do not maintain token balances.

The impact of economic policies on each of these stakeholders is examined below.

Validator Rewards

As a "Proof of Stake" system, the NEAR platform is secure because the validators who run nodes put some of their tokens at stake as a kind of deposit to guarantee good behavior and for validator selection (prevent Sybil attack). Should these validators produce an invalid block or create an alternative chain (eg. with the goal of creating a double spend), they will be "slashed", cutting this deposit.

Validators are chosen based on the "Thresholded Proof of Stake" model which uses an auction to determine how many "seats" will be allocated to each prospective validator (by determining the minimum threshold number of tokens for a single seat). This auction is designed to provide fair (equal opportunity) allocation and allow as many people as possible to participate in the network's validation process so it can achieve meaningful decentralization.

A validator can deterministically expect to participate in the validation process in a proportion that matches their proportion of total stake in the network. A given validator may become one of several possible roles:

1. Block Producer
2. Chunk Producer
3. Hidden Validator

Independent of the role the validator is assigned, its reward will be proportional to the percentage of the total amount staked by the validator. This means there is no need to pool stake under the minimum required to become a validator.

In exchange for servicing the network by producing blocks and chunks and providing security and data availability, validators are rewarded with a target number of NEAR every epoch. The target value is computed in such a way that, on an annualized basis, it will be 4.5% of the total supply.

Because validators are selected on a per-epoch basis and they each need to do an equal amount of work to validate chunks, provide data availability, and produce blocks, the reward gets allocated every epoch and gets divided proportionally to the stake of each participant.

All transaction fees (minus the part which is allocated as the rebate for contracts) which are collected within each epoch are burned by the system. The inflationary reward is paid out to validators at the same rate regardless of the number of fees collected or burned. Taken together, this means that system-wide inflation is reduced by an amount proportional to the amount of fees that are paid to the system and, should network usage fees exceed the system-wide inflation rate, the system will become deflationary.

The computational requirements for running a validating node are designed to be minimal. Most operators should be able to do so easily with a standard cloud-hosted virtual machine, for example with Amazon AWS's $100/month instance or a n1-highcpu-4 Google Cloud instance (and likely with even less robust hardware). For validators who anticipate staking substantial balances and thus who expect to participate in many shards simultaneously, they might want to utilize more hardware (and redundancy) to compensate for the extra storage, bandwidth and compute load they will need.

Third Party Observers ("Hidden Validators" and "Fisherman")

A key disadvantage of an unsophisticated sharded blockchain system is that the overall system security is split because only a portion of the network's validators are validating the transactions of a particular shard. NEAR employs two ways to counteract this problem: "Hidden Validators" and "Fishermen".

"Hidden Validators" are validators who are selected from the general validator pool and are assigned to validate shards that are unknown to any parties except themselves. This process, which is described in greater detail in other sections, ensures that it is extremely difficult to successfully corrupt a sufficient number of nodes to perfrom malicious behaviors in a shard.

Hidden Validators are compensated for validating and signing off on the validity of chunks and blocks as a normal part of the validator compensation process.

"Fishermen" are observing nodes who permissionlessly detect and report bad behavior. These nodes are synced with the network but are not necessarily participating in the consensus and don't actually get paid for any specific ongoing activity. They can include wallet operators, application developers or exchange infrastructure. These nodes validate parts of the chain that are important to them and, if they detect issues, they can flag those issues via challenge. To prevent "griefing" of challenges, a small bond of 10 NEAR must be posted ahead of time.

The system does not provide any reward for operating a node as a Fisherman (there is no reward for sending a successful challenge). Instead, participants who run a Fisherman node generally have outside motivations for maintaining the security of the network.

Contract Rewards

A portion of the fees generated by a particular transaction is provided back to the contracts that were run during that transaction. This "Contract Reward" may be distributed in accordance with the rules specified in the contract, for example, it may be allocated to an account controlled by the contract developer, by investors, by a DAO, etc.

The percentage of fees that are allocated to this reward is set to a minimum value as a system-level parameter, initially 30%. Developers always can charge extra outside of this mechanism by requiring user to attach funds to the call.

This creates a business model for developers who might otherwise not have a meaningful way of charging for their applications. Having a minimum fee which is set at the system level avoids a "race to the bottom" which results in zero rewards due to competition (or simply the "forking out" of the fee by another developer).

This has a powerful effect of incentivizing developers to build applications and core contracts for the network because they will be directly compensated proportional to the usage of these contracts.

Token Holders

Token holders may choose not to participate in the staking process, for example because they are only temporary holders, they are providing liquidity for trading markets or they simply prefer not to participate. Token holders who do not participate in the staking and validation process receive no additional benefits from the operation of the network itself, though their tokens have utility by powering storage of the data and their usage of applications running on the network.

Protocol Treasury

To enable continued community growth and evolution of protocol, part of the inflation (10% of the inflation, or a total of 0.5% annually per initial parameters) is allocated to Protocol Treasury. In the future, these are expected to be managed by the community with the goal of coordinating long-term development of NEAR ecosystem, particularly efforts that won't sustain themself like future protocol development.

Given the current state of decentralized governance, initially treasury will be overseen by the NEAR Foundation. The NEAR Foundation's mandate is to enable community-driven innovation to benefit people around the world. While the foundation is nonessential to the operation of the network, which is fully decentralized, it is already able to support the project's ongoing evolution in ways that other entities would find difficult. For example, it is funding education, events or infrastructure projects which benefit the commons but do not have a particular business model and which would otherwise not be funded.

The amount of inflation that is allocated to the Protocol Treasury also acts as a decentralizing economic force since it allows for the redistribution of capital back into the ecosystem to developers and other participants who support the commons who might not otherwise have stake to offer.

Special Conditions

Slashing Conditions and Progressive Slashing

There are two major types of malicious behaviour possible on the NEAR platform:

1. **Double Signing:** Signing two or more different blocks at the same height.
2. **Invalid Chunks:** Signing a chunk with an invalid data or computational result.

Malicious validators might double sign because they are trying to execute a chain reorganization which reverts certain transactions (which might allow them to conduct a "double spend" as a result).

Non-malicious double signing in a Proof of Stake system can also happen due to a misconfiguration or issue in the software.

To balance the risk of accidental slashing, NEAR uses "Progressive slashing". This is where the portion of stake that is slashed is a multiple of the amount of stake that exhibited the double

signing behavior during the epoch in question. This multiple is 3, so each malicious participant gets slashed by 3 * malicious_stake / total_satake.

For an example of a double sign which leads to slashing, assume a validator has 1% of the total tokens which have been staked in a particular epoch. Assuming the total amount of tokens staked in the epoch is 50,000,000 NEAR, this validator has 500,000 NEAR at stake. If that validator double signs and there are no other double signs, the validator will lose 3% of their stake in that epoch, so they will have 485,000 of NEAR returned to them and 15,000 NEAR will be burned. If the total amount of stake that double signed within an epoch reaches 33% (which becomes dangerous for the network), the entire stake of all involved parties will be slashed.

For an invalid chunk, the full stake of the validator gets slashed. This is because an invalid chunk is only possible if the node is actually malicious (they have modified the code).

Section 07

Technology

NEAR's community-operated cloud uses a novel consensus algorithm and a scalable sharding architecture to achieve its high level design goals.

The key elements of NEAR's technology are:

- **Sharding:** The system is designed to scale horizontally and near-infinitely by distributing computation across multiple parallelized shards.
- **Consensus:** Consensus is achieved across all of the nodes which make up the network operators across all of the shards using the new Nightshade algorithm.
- **Staking Selection and Game Theory:** To participate in the validation process, stakers are selected using a secure randomized process which optimally distributes seats across parties and provides incentives for them to operate with good behavior.
- **Randomness:** NEAR's randomness approach is unbiasable, unpredictable and can tolerate up to 1/3 of malicious actors before liveness is affected and 2/3 of malicious actors before any one can actually influence its output.

Technology Design Principles

NEAR's overall system design principles are used to inform its technical design according to the following interpretations:

1. **Usability:** End users should be burdened with the lowest possible security obligations for a given type of interaction. Developers should be able to easily build, test and deploy contracts in familiar languages and should be able to provide their end-users with experiences close to today's web.
2. **Scalability:** The platform should scale infinitely as its capacity is used.
3. **Simplicity:** The design of each of the system's components should be as simple as possible in order to achieve their primary purpose.
4. **Sustainable Decentralization:** The barrier for participation in the platform as a validating node should be set as low as possible in order to bring a wide range of participants.

Individual transactions made far in the future must be at least as secure as those made today in order to safeguard the value they modify.

Summary

NEAR focuses on providing solutions to the two core problems of today's blockchains — Usability and Scalability.

Usability for end users is achieved through offering a progressive security model for wallet interactions and by giving developers more opportunities to craft experiences which closely resemble the web today. These are provided by flexible and programmable key management implemented on the protocol level as a result of NEAR's contract-based account model. This allows things like meta transactions, atomic account transfers, accounts with funds that are locked for specific usage and other account programmability and restriction use cases to be easily implemented.

Usability for developers is provided by setting up the protocol to provide for browser-based debugging, familiar programming languages (like AssemblyScript and Rust) and contract usage rebates ("Contract Reward").

Scalability is provided by sharding the chain into a potentially unlimited number of subchains, each of which operates in parallel.

Performance Characteristics and Tradeoffs

One commonly referenced trilemma states that a system cannot achieve scalability, decentralization and security at the same time. NEAR's sharding and validator selection approaches provide significant scalability and decentralization while mitigating tradeoffs in security that would normally occur with such improvements.

Another classic trilemma is posed by the CAP theorem, which states that a system can only achieve 2 of Consistency, Availability (aka "liveness") and Partition Tolerance. Given that partition tolerance cannot be sacrificed in this case, the tradeoff is really between consistency and availability.

In blockchain-based systems, an illustrative example is what happens if the network splits into two parts for a week. A *consistent* system will completely shut down one (or both) of the halves until the network is restored so that the two parts do not become inconsistent. An *available* system (like Bitcoin) will continue to run both halves of the network independently and, when they are restored to unity, the operations of one half will be wiped out in favor of the operations of the other.

NEAR currently favors availability at a system level but individual users can choose to not accept blocks without >50% signature thresholds as a way of locally requiring consistency as well.

The performance of the system will be highly dependent on the types of transactions which are processed and the actual hardware which is supporting it. For simple financial transactions, per-shard throughput could range from 400-2000 transactions per second.

Sharding

Current approaches to scalability typically fall into two categories:

1. **Vertical Scaling:** Achieved by improving the performance of the existing hardware of a system. In the case of blockchain-based systems, it typically means running a network containing fewer nodes that each require *better* hardware. This creates an initial improvement in throughput while limiting future improvements to roughly the rate of increase in performance of computing hardware (often considered to be “Moore’s Law”). This leaves the network without the ability to scale at a rate commensurate with its adoption.
2. **Horizontal Scaling:** Achieved by adding *more* hardware to a system. In the case of blockchains, this is typically done by ensuring that an increase in the number of nodes participating in the network improves the performance of that network by a commensurate amount, for example by parallelizing computation across multiple “shards”.

NEAR uses a sharding approach to provide scalability horizontally, which allows it to scale capacity alongside increases in demand.

Cross-Chain and Cross-Shard Communication

One of the biggest difficulties with any form of cross-chain communication, whether this occurs within the shards of a single chain or across multiple chains, is determining that an incoming transaction from another chain is valid. There are 3 approaches for validating cross-chain transactions:

1. **Dual Validation:** Have the validators for the receiving chain also validate on the sending chain. This is used by Quarkchain. It has the downside that validators do not scale well in this approach.
2. **Trust the Transaction:** Assume that if a transaction has been received, it must be valid. In Cosmos, for example, a transaction that is copied to the main hub is considered irreversible. They keep track of the total number of tokens in each economy so you cannot create new ones but you could theoretically create invalid transfers between parties (eg steal tokens from other parties).
3. **Beacon Chain w/ Rollback:** A beacon chain verifies the state transitions all of the other chains using a small subset of validators and, if a problem is detected, all chains are rolled back. To achieve atomicity, this reversion should happen, though it should happen only rarely and should be immediately detected.

NEAR focuses on the 3rd approach. With the assumption that an adaptive adversary cannot corrupt the validators of a shard within a day, validators of each shard can be rotated daily to help add a layer of security. But presumably it **is** possible (if very difficult) for an adaptive adversary to corrupt a shard’s validators within a given day.

To help offset this, other protocols use a smaller committee which rotates far more rapidly (eg every few minutes) and validates across shards. In order for this smaller committee to perform their validations without having to download the entire state of each shard (which cannot be done in this timeframe), they receive only that portion of the state which was actually affected. But it is difficult to send the state with each change — a single transaction might affect 100mb of state at a time.

This is where the **Nightshade** sharding approach comes in.

Nightshade

Nightshade modifies the typical sharding abstraction and assumes that all of the shards combine together to produce a single block. This block is produced with a regular cadence regardless of whether each individual shard has produced its "chunk" for that specific block height. So every chunk for each shard will either be present or not.

There must be a fork choice rule to decide on the proper fork. This is still under development but will most likely resemble LMD Ghost. It will include the weight of how many validator attestations have been received for a given chunk and block.

There is a single validator assigned to produce each block. This validator must assemble the chunks which are provided to it during that block's time period into the period's block. The assignment of this validator will rotate through the existing validator set (eg 100 validators). This leader does not accept transactions, only chunks.

For each individual shard and period, a single validator is assigned to produce its chunk. If that validator is not present, the shard will stall for that period. Each shard has its own smaller pool of validators which is pulled from the main pool. The shard leader position rotates among this smaller pool (eg 4 validators) in the same way that the overall block leader is selected. Thus, if a single validator is absent and the shard chunk stalls for one period, the next validator will likely be present to continue the chain's operation in the following period.

Learn more about NEAR's sharding design in the Nightshade paper.

Hidden Validators

In order to provide additional security, NEAR uses Hidden Validators. These are a smaller committee for each shard (on average 100 validators) who verify each chunk. Rather than having this assignment be on the blockchain and thus publicly visible to all participants, the validators themselves figure out their assignment individually by drawing a set of shard ids from a Verifiable Random Function (VRF).

This way each individual validator is aware of which shards they must verify but, to corrupt them, an adversary must bribe a large percentage of total validators across all shards to reveal their masks.

Further, the number of hidden validators assigned to a particular block is randomly determined as well. This prevents an adversary from knowing exactly how many hidden validators they even need to corrupt in the first place in order to successfully pull off an attack. This prevents attacks where an adversary broadcasts their intent and waits for the fishermen to come to them (revealing which shards they are validating).

Due to the nature of the verification, any single hidden validator can present a proof that the chunk is invalid, a so-called "fraud proof".

The selection of the smaller per-shard committee is done for every epoch (½ day) from the same pool as the block and chunk producers, which is the total set of nodes which staked. For example, if there are 100 seats per shard and 100 shards, there are a total of 10,000 seats. 100 of them will be allocated to be the chunk producers and the rest will be hidden validators.

Fishermen

In addition to the hidden validators who are assigned to provide security for each shard, any other node operator can participate permissionlessly as a so-called “fisherman.” This third-party node can provide the same fraud proof as a hidden validator and thus they too can kick off the process of slashing and rollback.

This means that, even if an adversary successfully corrupted the entire hidden validator pool, they have no guarantee that their efforts will not be discovered by one of these independent fishermen and are thus highly disincentivized.

Preventing Lazy Validators

One potential problem with validators is that they can be “lazy”. After every block, a validator must receive the new chunk, download the new state and run validation on that block. They could, however, choose to do nothing unless they see another validator submitting a fraud proof and only then do they actually validate the latest block and try to submit a proof of their own. Thus a chain can end up paying validators but receive no meaningful work from them.

This is mitigated by making validators to first commit to their decision (if chunk is valid or invalid) and then reveal what they committed. This creates an incentive to do proper work because they have value at stake and will be slashed if they miss an invalid chunk.

Preventing Data Hoarding

Another problem can occur if a chunk is corrupted (by corrupting its small set of chunk producers) and the chunk producers refuse to provide sufficient data to hidden validators so they are unable to validate or make a fraud proof.

This is solved by requiring the chunk producer to send out an “erasure coded” chunk to other chunk producers in other shards. This code allows these other producers to reconstruct the chunk from just 16% of the parties and hidden validators who can validate it. If the (presumably corrupt) chunk producer did not provide this for their chunk, then no other chunk producer will attest or build on top of the chain which they don’t have parts for. The “fork choice rule” will select the chain that actually has at least 50% of parties having parts.

To prevent bad behavior, a single random part of this erasure code is deliberately made to be a “land mine” (invalid). At any time, if a validator is shown to have attested to a block which contains the land mine (which is easily proven), they will be slashed. Thus for each period there is also a small chance that a bad actor gets slashed so it highly disincentivizes bad behavior.

Randomness

Randomness in the blockchain needs to have the following properties:

1. Unbiasable
2. Unpredictable
3. Liveness, i.e. tolerates actors going offline or malicious actors

There are a few potential approaches:

1. RANDAO – unpredictable but biasable. Liveness depends on the underlying consensus protocol;

2. RANDAO+VDF – unpredictable, unbiasable, has liveness. But in practice it is hard to use it and be ASICS-resistant at the same time;
3. Threshold Signatures — unpredictable, unbiasable, has liveness. But requires a complicated mechanism to generate private keys in a particular fashion. It is an active area of research at the moment.
4. RandShare — unpredictable, unbiasable, has liveness. But requires O(n^3) network communication messages, which is a lot, where n is the number of participants. And also becomes biasable with more than 1/3 malicioius participants which is a low threshold.

NEAR's approach is unpredictable, unbiasable and has liveness. Unlike Randshare, it tolerates up to 2/3 malicious participants before it becomes biasable. Unlike Threshold Signatures, it is simple. Unlike RANDAO+VDF, it can not be attacked with ASICs. Learn how it works in the NEAR Randomness paper.

Section 08

Governance

Governance defines how the protocol is updated ("Technical Governance") and how its resources are allocated ("Resource Governance"). Technical governance generally includes fixing bugs, updating system parameters and rolling out larger scale changes in the fundamental technology of the protocol. Resource governance generally includes the allocation of grant funding from community-initiated sources (like the allocation provided to the foundation).

Technical governance is particularly complex because of the required coordination between potentially thousands of independent node operators around the world. Each of those nodes must go through the upgrade process in order to participate in the newest version of the network. Any who do not may end up (attempting to start) running a separate chain. Thus it is important that the upgrade process is smooth and that the nodes it affects buy into the decisions that have been made.

Many protocols perform the decision-making aspect of governance "off chain", meaning it occurs in text channels, in-person and via phone calls where key stakeholders or their representatives decide on the best course of action. This uses the dynamic nature of fluid human communication to debug major issues but is also subject to all the challenges of having big personalities and soft power structures in place.

Other protocols lean heavily on "on chain" governance, where decisions are explicitly made by the holders of the protocol's key resources (eg tokens) via an online voting mechanism. This provides explicit clarity around decision making and rollouts but suffers from the need to very clearly specify each case. It also has potential problems arising from a lack of "human common sense" around some decisions and is therefore vulnerable to certain attacks that an off-chain process would not be.

Governance Design Principles

Here is how NEAR's core design principles apply to governance:

1. **Usability:** Governance processes should be clear and understandable. Mechanisms for active participation and for voting (where available) should be simple and straightforward.

Governance should be effective and efficient so it arrives at decisions quickly and implements them efficiently. The community of stakeholders should have sufficient voice that they support the legitimacy of decisions and do not exit or fork the platform.

2. **Scalability:** Governance should scale as the scope and complexity of the platform itself grows, as the diversity of its stakeholders increases and as the breadth of participation expands.
3. **Simplicity:** The most robust processes tend to be the simplest so good governance should avoid overengineering processes and acknowledge that often human-to-human communication is the simplest approach.
4. **Sustainable Decentralization:** Governance should allow participation from the full breadth of stakeholders in the platform but be resilient against capture by any one of these over time.

It is important that governance design balances between **efficiency** and **resiliency**. Decisions must be made and implemented efficiently if a technical platform is to continue to evolve sufficiently to provide the best value for its stakeholders but that platform must ensure that it can not be captured over time by a particular group of stakeholders.

Summary

NEAR's governance is designed to provide for efficient improvement of the protocol while allowing the community sufficient voice and oversight in order to ensure the protocol maintains its independence. The long term goals are to combine community led innovation with effective decision making and execution and to receive proper representation from each of the key stakeholder roles in the network.

For example, the NEAR community initially includes token holders, validators, application developers, protocol developers, community leaders and more. Each of these stakeholders has a different set of views, opinions and inputs to various key issues.

Having proper representation means that decisions will require deliberation and discussion, which can slow down the necessary evolution of the protocol if left unchecked. To maintain a bias for efficient execution, a highly qualified entity is needed to maintain the Reference Implementation of core protocol code. This maintainer, who is called the Reference Maintainer, should be selected and overseen by the community.

Initially, governance activities are coordinated by the NEAR Foundation, an independent nonprofit entity whose mission is well-aligned with improving the long term usefulness of the ecosystem. These activities include maintaining oversight over the Reference Maintainer, supporting the build up of the governance coordination tools, certain token distributions and laying the groundwork for community operated governance.

Technical Governance

As a decentralized network, no single entity can ever force changes to the full NEAR network. Any changes made to the reference code base by its core contributors must be individually accepted by the nodes who are running the network.

It is still important to understand the core process which is used to push changes to the reference code base because these are most likely to represent the will of the community and thus receive acceptance by the network nodes which form part of that community.

NEAR's governance defines a Reference Maintainer, which is an entity responsible for technical upgrades to the NEAR network. This entity has been selected to maintain the Reference Implementation and continue to suggest improvements on the specification. All major releases will be protected with community discussion and a veto process (a 2 week challenge period), while smaller bug fixes can be rolled out fast and delivered to node operators.

Initially, the Maintainer is selected by the Foundation Board and serves until the board votes to replace them. Over time, oversight of the Maintainer will be performed through a community-representing election process.

Resource Governance

Resources provided by the network itself to the Protocol Treasury are governed and distributed by the NEAR Foundation. This foundation operates independently and will provide structured and transparent funding for projects and activities that are deemed to be most helpful to the ongoing health of the protocol's ecosystem. This may include technical projects (like the Reference Maintainer) and nontechnical projects or initiatives that support the commons and the community at large.

Section 09

Primitives

With the emergence of every new computing paradigm, there is a significant amount of uncertainty about exactly how it can be most effectively utilized and what it means for the future of innovation. This time is no different.

A blockchain-based application platform like NEAR combines two existing cloud services — compute and storage — in a trustless and permissionless manner. The combination of these services in this way creates a set of brand new primitives which can be used to build new applications, new value chains and new businesses.

With something so new, it is generally best to start at this fundamental level in order to understand what it can — and also what it should not — be used for. It is important to acknowledge that the universe of possibilities which is created when new primitives and new value chains are introduced cannot be fully known during the early phases of the technology's introduction. Few, for example, could have predicted how the camera+GPS in everyone's pocket has changed the world when the first smartphones came out in the early 2000's.

The following sections will explore what the technology is good for, what it is not good for, and many of the specific primitives that it enables today. They also provide a clear mental model for understanding when blockchain should be applied and when it should be avoided. Discussion of the future is left for the concluding section of this paper.

Technology creates primitives which enable use cases which are implemented by applications.

What the Technology is *Not* Best At

Before discussing primitives, it is important to acknowledge what this technology is *not* best for in order to dispel some persistent misconceptions.

A community-operated cloud like NEAR is neither inexpensive nor fast relative to existing compute and storage solutions alone. This is by definition — the specific benefits of using a community-operated cloud are that it leverages redundancy in both computation and storage to create the security that provides the network with its greatest benefits.

This can be understood on the most basic level by examining how these networks function — by aggregating compute and storage across a number of individual nodes.

If the network is made up of one shard containing 100 nodes and each node is running on its own individual hardware in parallel with the others, by definition running computation on the network will cost at least 100x more than running a single piece of hardware and it will be slower by an order of magnitude relative to the time it takes to communicate between these nodes on the network.

Similarly, while the storage capacity of the network is theoretically infinite, it is practically limited by the rate and cost at which new validating nodes can be added to the network. The storage capacity for each shard is fixed at a level which enables new validators to participate in the network and to sync with a new shard in time for each new shuffling action. Since each of the validators within a shard replicate its storage, a new shard must be created in order to add new storage capacity to the network. Each new shard requires the addition of a new set of validating nodes. Using the example above, this means that adding new storage capacity to the network will require bringing in another 100 validating nodes, whether as brand new validators or by enabling existing validators to span a greater number of shards. The economics of storage on the network must reflect this reality so it will always cost at least multiples more than running a single piece of new storage hardware.

Advances in storage technology do allow for some optimizations on how much storage is required but do not remove the fundamental reality of these economics.

Thus it is impossible for a blockchain-based system to claim to be faster or cheaper than a centralized cloud computing system like Amazon's AWS or Google's Cloud Provider for traditional compute or storage use cases. *Applications which require optimization along these dimensions are *not* the use cases that are best enabled by the initial implementation of the NEAR platform*.

New Benefits of the Technology

Before examining the specific primitives which are enabled by the platform, we examine the fundamental benefits that are provided by combining compute and storage in a permissionless and trustless way via the community cloud.

This high-level conceptual understanding is important because it moves us beyond simply asking how blockchain can do *existing* tasks cheaper or faster than current cloud architectures and into the realm of what *new* use cases are enabled.

Decentralization for High Value Assets

Decentralization can be thought of as a spectrum along an axis which represents how many actors must be corrupted in order to compromise the system. On one end, most of today's

systems (including today's cloud compute and storage solutions) have a centralization factor of one: by holding the access keys of a single player, data on that system can be arbitrarily modified. On the other end is a fully decentralized system, where one must corrupt dozens, hundreds or even thousands of actors in order to corrupt the underlying data.

Many of today's blockchain-based systems are actually fairly centralized, whether by design or because their systems incentivizes pooling or delegation in a way which creates a small number of very powerful players or because their governance processes are easily captured.

As previously discussed, there is an additional cost associated with the redundancy provided by decentralization and thus the strongest use cases are those which benefit sufficiently from what this decentralization provides.

Most of the data which is stored in today's applications is low value and high volume. Decentralization is most important in cases where the data being stored is highly sensitive or vulnerable to censorship, theft, corruption or other forms of modification.

In particular, this means data which represents digital money, identity and asset ownership benefit the most from storage on the NEAR platform.

A Shared Global Data Layer

The NEAR platform allows applications to access the same pool of shared data which makes it easy for multiple applications to share state with each other. This is different from traditional web applications, where each application typically stores its data in a proprietary database and these databases typically do not provide easy communication between each other.

The state which is shared across applications can be any of the data types previously mentioned — digital currency, identity, assets and more. This data is cryptographically secured by default so only applications which have the user's permission can modify their data. Because the users effectively own their own data, they are able to modify — or transfer — it without the permission of a third party application.

This means that not only is NEAR able to store high value data like monetary assets but users and applications can also transfer those assets easily between each other in a way which isn't possible using today's platforms.

It should also be noted that data can still be encrypted and protected to preserve security and privacy.

What Developers Get "For Free"

In addition to its core benefits, NEAR gives developers access to a distributed architecture that typically takes months of setup to accomplish. This includes partition-resilient networking, a high-availability database and internationally distributed endpoints.

Developers also gain easy access to a number of primitives which typically require substantial development effort to implement in the current web world. As one example, they have native access to cryptographic primitives, which allows for sensitive use cases. As another, their applications have easy access to Single Sign On (SSO) for all users of the platform so those users experience little to no friction when trying out new applications.

Native Primitives

Now that we've introduced some of the high level benefits of combining compute and storage into a single decentralized and shared data layer, let's move into what new primitives that enables. Primitives are the fundamental building blocks of use cases.

These primitives fall into the following categories:

1. **Assets:** Assets of all types (from money to data) are now digitally native, meaning they are verifiably unique, individually owned and completely programmable.
2. **Accounts:** Every actor in the ecosystem has an account which gives them secure storage for their assets, an easy way to verify their identity to applications and an accumulation of reputation over time
3. **Transactions:** Because assets are digitally native and accounts are part of the global pool, programmable transactions between parties are simple, cheap, secure and near-instant.
4. **Verification: Because NEAR's storage is an immutable public ledger, data and code that are saved to the platform are publicly verifiable for both timing and content.**

We'll examine each of these in greater depth in the following sections.

Asset Primitives

Assets are now digitally native, verifiably unique and fully programmable. This can be used to provide the benefits of digitization to existing assets or the creation of new categories entirely.

For example, whereas money used to be a single one-dimensional commodity, now it is fully programmable — everything from the terms of its issuance to conditions of its use can be baked directly into the asset itself.

Asset primitives include:

1. **Programmatic Ownership:** Each account has verifiable control of its money, its digital goods and its data (which can represent real-world things like identification) as well as the ability to programmatically determine (or split) that ownership.
2. **Digital Uniqueness:** A digital asset can be 100% unique, representing anything from a specific fungible coin to a completely nonfungible token.
3. **Programmable Assets:** Programmatic asset creation, evolution and destruction

Account Primitives

Every actor in the ecosystem, whether human, contract or device is treated as having a top-level account. Treating each of these as first class citizens of the platform allows for both old-world identity and new styles of interaction between autonomous or semi-autonomous actors.

Account primitives include:

1. **Autonomous: Everything on the chain gets assigned to an account. Accounts can represent a person, company, app or even a thing (eg a refrigerator). Accounts are each first-class citizens regardless.**

2. **Single Sign-on (SSO):** One account works across the whole world of apps and anything else that wants to tie into the chain.
3. **Reputation/History:** Every account's transaction history gives it reputation which can be used across services.

Transaction Primitives

The provision of Asset and Account primitives in the same shared data pool makes it trivially simple to create seamless interactions between those elements in ways which are almost impossible outside of the digitally native medium. The most commonly cited use case involves permissionless peer-to-peer transfer of money but this applies more broadly to any kind of digital asset.

Transaction primitives include:

1. **Direct:** Transfers can be made directly between accounts without requiring the code or permission of a specific application, allowing peer-to-peer marketplaces or transfers.
2. **Instant:** Financial and digital-good transactions have finality in seconds and do not require long waiting periods for confirmation.
3. **Micro:** Negligible fees make high frequency or small quantity transfers significantly more viable than current financial infrastructure allows.
4. **Conditional:** A smart contract can easily add logic to transactions, for example to create conditional escrow or time-based releases of currency or data.

Verification Primitives

The immutable public ledger used to store data on NEAR creates both a verifiable record of what has occurred in the past and a verifiable repository of the code which is being run behind particular applications. This can be used in a number of creative ways from the small (is a dice game actually using randomness?) to the large (creating audit trails for supply chains).

Verification primitives include:

1. **Checkpointing:** Cryptographic timestamping means it is easy to store verifiable checkpoints of state at a particular time, allowing applications to verify the authenticity or occurence of previous activity.
2. **Verify Process Integrity:** The code which runs apps deployed to the platform can be verified in a way that current server-side code cannot.

Combinatorial Primitives

While it isn't hard to examine the low-level primitives which are natively provided by the technology of the platform, perhaps the most exciting possibilities come from combining multiple primitives to create higher level primitives. While many of these will be discovered over time, some examples include:

1. **Permissionless Markets: Most markets today require permission from someone in order to function, for example the provider of the marketplace where activity occurs. The combination of multiple low-level primitives disintermediates this control and allows**

permissionless markets to flourish in places where there was no room to operate previously. This requires the combination of:

1. A native medium of exchange (to transact in) and unit of account (to price in).
    1. Note that dynamic cross-currency conversion can make this easier across currencies but users generally still prefer to have a single schelling point currency.
2. Verifiable ownership of an asset
3. Peer-to-peer/permissionless transfer of the asset
4. A censorship-resistant marketplace application (which provides discoverability, matching and pricing)

2. **Derivative Assets: While it's significant to provide censorship-resistant asset storage in the first place, combining multiple low-level primitives allows us to create an infinite variety of new assets which combine existing assets, transactions and logic to meet the risk management needs of anyone so inclined to use them. This requires the combination of:**
    1. Verifiable ownership of an asset
    2. Programmatic escrow of an asset
    3. Programmatic rights transfer
3. **Open State Components:** Any app has access (when granted) to the shared pool of state data, whether in regards to specific assets or users of the platform. This allows components to operate as microservices do in today's applications — performing specific functions that can be composed together to achieve larger business goals. Because they are public, competition will ensure that the best of these achieve usage. This requires the combination of:
    1. A shared data pool
    2. User-sovereign data
    3. Verifiable process integrity

Section 10

The Future

The future, like the Internet itself, is infinitely configurable. We don't know what it will look like but we can both identify some of the key forces which will govern its path and predict some of the major tools that will take us there.

Privacy: By default, activity and data on the blockchain is done in plain sight. The essence of privacy, however, is choice — whether their activity should be made transparent or hidden from view. Though the default technical tooling doesn't provide this privacy protection, a number of solutions applied on top of it make this possible.

In the weak form, data can be encrypted before writing to the chain. This generally protects the integrity of the data itself but still leaves transactions vulnerable to tracking and good analysis can often piece together what actually occurred. Thus new technologies like zero knowledge proofs hold an interesting opportunity to make not just data private but also the very computations that modify it.

This technology isn't baked into the NEAR platform day 1 but, should the community drive for it, it is possible to implement.

Private Shards: Not all blockchain use cases demand the full security and protection of the public chain for each transaction. Sometimes a consortium of users or even a single entity would prefer to run their own chain, where they control all of the validation and periodically checkpoint back to the main chain for security or verification and communication of activity. In this case, particular shards could potentially be configured to use a special predetermined validator set, thus making them "private".

Mobile Nodes: Sharding is a horizontal scaling technology where the total processing power of the network is proportional to the number of CPUs attached to it. In simpler terms, the more devices which are supporting the community cloud by participating in the validation process, the more transaction throughput the network can handle.

There are billions of devices with viable CPUs spread across the world. The network will be able to achieve extraordinary scalability by tapping into even just a fraction of the more robust of these devices. But should additional capacity be required, the requirements for running nodes could be adjusted such that even mobile devices could participate. While the engineering tradeoffs are important, this could provide access to another billion nodes running in everyone's pockets and is thus an interesting area for future exploration.

Internet of Things (IoT): IoT devices are an even more specialized case than mobile devices because they represent the lowest processing power and the highest number of available CPUs.

Composable Components (The Open Web): It starts with a global, free currency and continues with unkillable applications but, eventually, the dream of the open web becomes one where all the available applications can be easily assembled to create new functionality. Consider the hardware analogy of what the GPS, camera and an internet connection of the modern smartphone have unlocked and apply the fluidity of software to it. There's no telling how rapidly innovation can occur in a world where this is possible. With NEAR's global state accessible to all applications, this future will become reality.

What's Next?

Take the first step! Development of the protocol is open source at https://github.com/near and you can learn more about how the code works plus see examples of what you can build at https://docs.near.org. You can ask questions at https://near.chat or on Stack Overflow at https://stackoverflow.com/questions/tagged/nearprotocol.

Regardless of your experience level or skills, there is a way for you to participate so please join the journey and help NEAR build the future.

Section 11

Disclaimer

NOTICE AND DISCLAIMER

PLEASE READ THE ENTIRETY OF THIS "NOTICE AND DISCLAIMER" SECTION CAREFULLY. NOTHING HEREIN CONSTITUTES LEGAL, FINANCIAL, BUSINESS OR TAX ADVICE AND YOU SHOULD CONSULT YOUR OWN LEGAL, FINANCIAL, TAX OR OTHER PROFESSIONAL ADVISOR(S) BEFORE ENGAGING IN ANY ACTIVITY IN CONNECTION HEREWITH. NEITHER NEAR STIFTUNG (NEAR FOUNDATION) (THE **FOUNDATION**), ANY OF THE PROJECT TEAM MEMBERS (THE **NEAR TEAM**) WHO HAVE WORKED ON THE NEAR PLATFORM (AS DEFINED HEREIN) OR PROJECT TO DEVELOP THE NEAR PLATFORM IN ANY WAY WHATSOEVER, ANY DISTRIBUTOR/VENDOR OF NEAR TOKENS (THE **DISTRIBUTOR**), NOR ANY SERVICE PROVIDER SHALL BE LIABLE FOR ANY KIND OF DIRECT OR INDIRECT DAMAGE OR LOSS WHATSOEVER WHICH YOU MAY SUFFER IN CONNECTION WITH ACCESSING THIS WHITEPAPER, THE WEBSITE AT HTTPS://pages.near.org/ (THE **WEBSITE**) OR ANY OTHER WEBSITES OR MATERIALS PUBLISHED BY THE FOUNDATION.

Project purpose: All contributions will be applied towards the advancing, promoting the research, design and development of, and advocacy for community-driven innovation to benefit people around the world, focusing on the NEAR protocol. The Foundation, the Distributor and their respective affiliates would develop, manage and operate the NEAR Platform.

Nature of the Whitepaper: The Whitepaper and the Website are intended for general informational purposes only and do not constitute a prospectus, an offer document, an offer of securities, a solicitation for investment, or any offer to sell any product, item or asset (whether digital or otherwise). The information herein may not be exhaustive and does not imply any element of a contractual relationship. There is no assurance as to the accuracy or completeness of such information and no representation, warranty or undertaking is or purported to be provided as to the accuracy or completeness of such information. Where the Whitepaper or the Website includes information that has been obtained from third party sources, the Foundation, the Distributor, their respective affiliates and/or the NEAR team have not independently verified the accuracy or completion of such information. Further, you acknowledge that circumstances may change and that the Whitepaper or the Website may become outdated as a result; and neither the Foundation nor the Distributor is under any obligation to update or correct this document in connection therewith.

Token Documentation: Nothing in the Whitepaper or the Website constitutes any offer by the Foundation, the Distributor or the NEAR team to sell any NEAR token (as defined herein) nor shall it or any part of it nor the fact of its presentation form the basis of, or be relied upon in connection with, any contract or investment decision. Nothing contained in the Whitepaper or the Website is or may be relied upon as a promise, representation or undertaking as to the future performance of the NEAR Platform. The agreement between the Distributor (or any third party) and you, in relation to any sale, purchase, or other distribution or transfer of NEAR token, is to be governed only by the separate terms and conditions of such agreement.

The information set out in the Whitepaper and the Website is for community discussion only and is not legally binding. No person is bound to enter into any contract or binding legal commitment in relation to the acquisition of NEAR token, and no virtual currency or other form of payment is to be accepted on the basis of the Whitepaper or the Website. The agreement for sale and purchase of NEAR token and/or continued holding of NEAR token shall be governed by a separate set of

Terms and Conditions or Token Purchase Agreement (as the case may be) setting out the terms of such purchase and/or continued holding of NEAR token (the Terms and Conditions), which shall be separately provided to you or made available on the Website. The Terms and Conditions Documentation must be read together with the Whitepaper. In the event of any inconsistencies between the Terms and Conditions and the Whitepaper or the Website, the Terms and Conditions shall prevail.

Deemed Representations and Warranties: By accessing the Whitepaper or the Website (or any part thereof), you shall be deemed to represent and warrant to the Foundation, the Distributor, their respective affiliates, and the NEAR team as follows:

1. in any decision to purchase any NEAR token, you have shall not rely on any statement set out in the Whitepaper or the Website;
2. you will and shall at your own expense ensure compliance with all laws, regulatory requirements and restrictions applicable to you (as the case may be);
3. you acknowledge, understand and agree that NEAR token may have no value, there is no guarantee or representation of value or liquidity for NEAR token, and NEAR token is not an investment product including for any speculative investment;
4. none of the Foundation, the Distributor, their respective affiliates, and/or the NEAR team members shall be responsible for or liable for the value of NEAR token, the transferability and/or liquidity of NEAR token and/or the availability of any market for NEAR token through third parties or otherwise; and
5. you acknowledge, understand and agree that you are not eligible to purchase any NEAR token if you are a citizen, national, resident (tax or otherwise), domiciliary and/or green card holder of a geographic area or country (i) where it is likely that the sale of NEAR token would be construed as the sale of a security (howsoever named), financial service or investment product and/or (ii) where participation in token sales is prohibited by applicable law, decree, regulation, treaty, or administrative act; and to this effect you agree to provide all such identify verification document when requested in order for the relevant checks to be carried out.

The Foundation, the Distributor and the NEAR team do not and do not purport to make, and hereby disclaims, all representations, warranties or undertaking to any entity or person (including without limitation warranties as to the accuracy, completeness, timeliness or reliability of the contents of the Whitepaper or the Website, or any other materials published by the Foundation or the Distributor). To the maximum extent permitted by law, the Foundation, the Distributor, their respective affiliates and service providers shall not be liable for any indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including, without limitation, any liability arising from default or negligence on the part of any of them, or any loss of revenue, income or profits, and loss of use or data) arising from the use of the Whitepaper or the Website, or any other materials published, or its contents (including without limitation any errors or omissions) or otherwise arising in connection with the same. Prospective purchasers of NEAR token should carefully consider and evaluate all risks and uncertainties (including financial and legal risks and uncertainties) associated with the NEAR token sale, the Foundation, the Distributor and the NEAR team.

Token features: The native digital cryptographically-secured utility token of the NEAR Platform (**NEAR token**) is a transferable representation of attributed functions specified in the protocol/code of the NEAR Platform, designed to play a major role in the functioning of the ecosystem on the NEAR Platform, and intended to be used solely as the primary utility token on the platform. NEAR token is a non-refundable functional utility token which will be used as the medium of exchange between participants on the NEAR Platform. The goal of introducing NEAR token is to provide a convenient and secure mode of payment and settlement between participants who interact within the ecosystem on the NEAR Platform, and it is not, and not intended to be, a medium of exchange accepted by the public (or a section of the public) as payment for goods or services or for the discharge of a debt; nor is it designed or intended to be used by any person as payment for any goods or services whatsoever that are not exclusively provided by the issuer. NEAR token does not in any way represent any shareholding, participation, right, title, or interest in the Foundation, the Distributor, their respective affiliates, or any other company, enterprise or undertaking, nor will NEAR token entitle token holders to any promise of fees, dividends, revenue, profits or investment returns, and are not intended to constitute securities in Switzerland, Singapore or any relevant jurisdiction. NEAR token may only be utilised on the NEAR Platform, and ownership of NEAR token carries no rights, express or implied, other than the right to use NEAR token as a means to enable usage of and interaction within the NEAR Platform.

The NEAR token enables the economic coordination of all participants who operate the network plus it enables new behaviors among the applications which are built on top of that network, by providing the economic incentives which will be consumed to encourage participants to contribute and maintain the ecosystem on the NEAR Platform. As a decentralized network the NEAR Platform relies on various participants to provide resources for network maintenance, and so NEAR token will be used as the medium of exchange to quantify and pay the costs of the consumed resources. NEAR token is an integral and indispensable part of the NEAR Platform, because without NEAR token, there would be no incentive for users to expend resources to participate in activities or provide services for the benefit of the entire ecosystem on the NEAR Platform. Users of the NEAR Platform and/or holders of NEAR token which did not actively participate will not receive any NEAR token incentives.

NEAR token would have the following features:

1. pay within the ecosystem for various services, such as processing transactions, providing bandwidth, and storing data;
2. run a validating node (providing computational resources to validate information / produce blocks) as part of the network by participating in the staking process; and
3. help determine how network resources are allocated and where its future technical direction will go by participating in governance processes (for the avoidance of doubt, the right to vote is restricted solely to voting on features of the NEAR Platform; the right to vote does not entitle NEAR token holders to vote on the operation and management of the Foundation, the Distributor or their respective affiliates, or their assets, and does not constitute any equity interest in any of the aforementioned entities).

Disclaimers relating to the NEAR token: It is expressly highlighted that NEAR token:

1. does not have any tangible or physical manifestation, and does not have any intrinsic value (nor does any person make any representation or give any commitment as to its value);
2. is non-refundable and cannot be exchanged for cash (or its equivalent value in any other virtual currency) or any payment obligation by the Foundation, the Distributor or any of their respective affiliates;
3. does not represent or confer on the token holder any right of any form with respect to the Foundation, the Distributor (or any of their respective affiliates), or its revenues or assets, including without limitation any right to receive future dividends, revenue, shares, ownership right or stake, share or security, any voting, distribution, redemption, liquidation, proprietary (including all forms of intellectual property or licence rights), right to receive accounts, financial statements or other financial data, the right to requisition or participate in shareholder meetings, the right to nominate a director, or other financial or legal rights or equivalent rights, or intellectual property rights or any other form of participation in or relating to the NEAR Platform, the Foundation, the Distributor and/or their service providers;
4. is not intended to represent any rights under a contract for differences or under any other contract the purpose or pretended purpose of which is to secure a profit or avoid a loss;
5. is not intended to be a representation of money (including electronic money), security, commodity, bond, debt instrument, unit in a collective investment scheme or any other kind of financial instrument or investment;
6. is not a loan to the Foundation, the Distributor or any of their respective affiliates, is not intended to represent a debt owed by the Foundation, the Distributor or any of their respective affiliates, and there is no expectation of profit; and
7. does not provide the token holder with any ownership or other interest in the Foundation, the Distributor or any of their respective affiliates.

The contributions in the token sale will be held by the Distributor (or their respective affiliate) after the token sale, and contributors will have no economic or legal right over or beneficial interest in these contributions or the assets of that entity after the token sale. To the extent a secondary market or exchange for trading NEAR token does develop, it would be run and operated wholly independently of the Foundation, the Distributor, the sale of NEAR token and the NEAR Platform. Neither the Foundation nor the Distributor will create such secondary markets nor will either entity act as an exchange for NEAR token.

Informational purposes only: The information set out herein is only conceptual, and describes the future development goals for the NEAR Platform to be developed. In particular, the project roadmap in the Whitepaper is being shared in order to outline some of the plans of the NEAR team, and is provided solely for **INFORMATIONAL PURPOSES** and does not constitute any binding commitment. Please do not rely on this information in making purchasing decisions because ultimately, the development, release, and timing of any products, features or functionality remains at the sole discretion of the Foundation, the Distributor or their respective affiliates, and is subject to change. Further, the Whitepaper or the Website may be amended or replaced from time to time. There are no obligations to update the Whitepaper or the Website, or to provide recipients with access to any information beyond what is provided herein.

Regulatory approval: No regulatory authority has examined or approved, whether formally or informally, of any of the information set out in the Whitepaper or the Website. No such action or assurance has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of the Whitepaper or the Website does not imply that the applicable laws, regulatory requirements or rules have been complied with.

Cautionary Note on forward-looking statements: All statements contained herein, statements made in press releases or in any place accessible by the public and oral statements that may be made by the Foundation, the Distributor and/or the NEAR team, may constitute forward-looking statements (including statements regarding intent, belief or current expectations with respect to market conditions, business strategy and plans, financial condition, specific provisions and risk management practices). You are cautioned not to place undue reliance on these forward-looking statements given that these statements involve known and unknown risks, uncertainties and other factors that may cause the actual future results to be materially different from that described by such forward-looking statements, and no independent third party has reviewed the reasonableness of any such statements or assumptions. These forward-looking statements are applicable only as of the date indicated in the Whitepaper, and the Foundation, the Distributor as well as the NEAR team expressly disclaim any responsibility (whether express or implied) to release any revisions to these forward-looking statements to reflect events after such date.

References to companies and platforms: The use of any company and/or platform names or trademarks herein (save for those which relate to the Foundation, the Distributor or their respective affiliates) does not imply any affiliation with, or endorsement by, any third party. References in the Whitepaper or the Website to specific companies and platforms are for illustrative purposes only.

English language: The Whitepaper and the Website may be translated into a language other than English for reference purpose only and in the event of conflict or ambiguity between the English language version and translated versions of the Whitepaper or the Website, the English language versions shall prevail. You acknowledge that you have read and understood the English language version of the Whitepaper and the Website.

No Distribution: No part of the Whitepaper or the Website is to be copied, reproduced, distributed or disseminated in any way without the prior written consent of the Foundation or the Distributor. By attending any presentation on this Whitepaper or by accepting any hard or soft copy of the Whitepaper, you agree to be bound by the foregoing limitations.