

“Particles”

Whitepaper v2.0 - A pipeline that protects, tracks, and makes it easier to do business with new technology from the first line of code through distribution to the end user.

Part One: Background

What is a Blockchain?

A blockchain is a distributed database where information links across a network of decentralized computers. In a blockchain, data groups are in units called blocks. These blocks link to the ones before and after, creating a chain of information.

In a blockchain, data is decentralized. Decentralization means that, unlike a centralized system where centralized entities like companies or governments manage data, decentralized databases share information across many nodes, not controlled by any single entity. Altogether, this means data cannot be controlled or manipulated by one source, and information stays safe and unmodifiable. A well-known example of blockchain technology is Bitcoin, a digital currency that uses blockchain technology to be decentralized, operating without a centralized bank or authority. In Bitcoin's utilization of blockchain technology, financial transactions are verified and recorded on the blockchain's network of nodes, with each node being a computer for storing the blockchain's data. The network of nodes can be considered a distributed server room, with data held across many different machines worldwide. Another famous example is Ether, which, upon the Ethereum blockchain, functions as a decentralized digital currency and introduces functionality for smart contracts.

What is a Smart Contract?

Smart Contracts are computer programs that control and execute events and actions according to the terms of a legal agreement in a transactional manner. Smart contracts are viewable as a self-executing, decentralized enforcement mechanism.

In a legal sense, smart contracts can manifest as a machine-readable layer to a legal agreement or function as the legal agreement itself, with the terms of the agreement written directly into the smart contract's code. For example, a smart contract can act as an enforcement mechanism for a rental agreement. As a machine-readable layer to the rental agreement, the smart contract records the agreement between the landlord and tenant on the blockchain, then collects money from the tenant and verifies payment received on time. If the landlord later terminates the rental agreement, the smart contract can prevent the tenant from sending any further rent payments and send them notice of the agreement termination. This rental agreement is one example of many legal applications for smart contracts.

Outside of the legal ecosystem, smart contracts have gained a utility within software development, with their self-executing and transactional nature appealing to developers. A prominent use for this is virtual currencies. On the Ethereum blockchain, for example, cryptocurrencies other than the native currency of the Ethereum blockchain, Ether, have emerged. These currencies, called tokens, are created through smart contracts deployed on Ethereum, which dictate how they function as financial instruments. One of the most popular tokens on Ethereum, called Tether, is a stablecoin, which through its smart contract, forms a digital currency that anchors itself to the value of the United States Dollar.

In recent times, one application of smart contracts has been gaining popularity: the decentralized application or dapp. Decentralized applications are computer programs that operate autonomously through smart contracts running on blockchain systems. Many dapps currently available on the market, ranging from financial tools and digital artwork to games and new-age technologies, include CryptoKitties, Decentraland, and OpenSea. What makes dapps appealing to consumers compared to traditional apps is the increased functionality, such as interoperability with other applications and security made possible by blockchain technology. Dapps are free from the control of a single authority, resistant to downtime, not subject to censorship, safer for user privacy, provide an increased sense of ownership for their users, are transparent, and introduce greater flexibility for developers.

Tokens Explained

On the blockchain, a token is a stored asset. This stored asset can take various forms, usually seen in use across different dapps, where the most common tokens are Governance Tokens, Non-Fungible Tokens, and Finance Tokens. Governance tokens grant ownership and voting rights, meaning that holding a governance token gives the holder voting rights and membership in a project or organization. In non-fungible tokens, more commonly known as NFTs, the token represents ownership of a digital asset, and every token is unique. NFTs can be used to represent a digital collectible or as an authentication tool thanks to their unmodifiable, unique nature. Finance tokens include digital currencies and other applications of tokens as financial instruments, such as a reward for investment or an ownership interest in a business venture. These three types of tokens, when on the Ethereum blockchain, are typically created using the ERC-20 Token Standard, ERC-777 Token Standard, or ERC-721 Non-Fungible Token Standard.

The most significant differentiating factor to keep in mind between the different token standards is fungibility. For something to be fungible means it is interchangeable and not unique from other parts of a greater whole. For example, Euros are fungible since a specified amount of Euros equals that same amount of Euros, no matter the denomination (such as two Ten Euro banknotes or one Twenty Euro banknote). This idea of fungibility with fiat currencies applies to fungible tokens, like Bitcoin, Ether, and tokens created using the ERC-20 and ERC-777 Token Standards. In non-fungibility, something is not interchangeable because it is unique. You cannot exchange *Mona Lisa* for *Starry Night* because they are both non-fungible assets, each having unique characteristics that make them one of a kind and give them different values. This same principle applies to tokens created using the ERC-721 Non-Fungible Token Standard, more commonly known as NFTs.

Web3 Explained

The World Wide Web is continuously evolving, going through different iterations that change how we interact with technology and how we use the internet. The first iteration of the world wide web, called Web 1.0, mainly was static read-only content. This read-only content was what comprised the early days of the internet. In the early 2000s, Web 2.0 emerged with the advent of dynamic and interactive content. Content on the web was user-generated and easy to use, giving rise to social media platforms, publications, and online communities where people could create and share content easily. Web 2.0 also increased interoperability, making content compatible with various devices and systems. Having entered the 2020s, the advent of Web 3.0, more commonly referred to as Web3, is upon us. Web3 is the newest iteration of the world wide web, introducing blockchain technology and creating a decentralized web that provides greater privacy, security, scalability, interoperability, and interactiveness. Using the blockchain, Web3 has a high amount of peer-to-peer computation, communication, and decision-making to run Web3 applications, leading many to believe that Artificial Intelligence will play a significant role. These changes are an improvement over Web 2.0 because they increase the ease of access to automation and computing power. It was previously more difficult to obtain and incorporate into applications for smaller developers, companies, and organizations. Dapps and digital currencies are examples of early cases of Web3.

Open-Source v. Closed-Source

Open-Source is the source code of a computer program that is made publicly available. In open-source, also sometimes called Free Open-Source Software (FOSS), anyone can access source code and modify it freely to fix, improve, or enhance the source code. Open-source code management often uses Git, and many open-source projects are available on open-source software hubs like GitHub, Gitlab, and Bitbucket. Git is a version control system that helps developers keep track of changes to their code over time. In a project managed by Git, the latest official code version is in the Main branch of the code's repository. A repository is an archive of the code worked on in a project, which stores the source code. It can also contain documentation, notes, assets, and other information for the project. Any changes made since the last update to the Main branch will become branches of the repository. One use for branches is to create new features and ensure compatibility before incorporating them into the Main branch's code. New projects can also emerge from branching off the original project in actions called forks, where the forked branch becomes the new Main branch of a new repository for a new project based on the original. A great example of an open-source project is Ethereum, which has had several forks, one of the most prevalent being Ethereum Classic, a fork of Ethereum maintaining an unaltered version of the Ethereum Network.

Open-source code differs from closed-source, also known as proprietary software. In closed-source code, the source code of a computer program is generally not made publicly available, which is seen as the opposite of open-source. The creator of a closed-source project maintains all rights to use, modify, and distribute the code, which often presents opportunities for monetization of their work and preventing third parties from modifying and distributing their work without prior consent. An example of closed-source code would be the operating system Windows, where licensing from Microsoft is required to use the software, and modification by developers is not permitted.

Legal Entities

In business, companies exist as legal entities. In the United States, for example, businesses can typically form as a Partnership, a Limited Liability Company, or a Corporation. Creating a legal entity for a business provides separation between the business owner and the business itself, allowing treatment as legally separate from the owner or owners. By forming a legal entity, the company has its legal rights and responsibilities different from the owners. An owner's assets are treated as separate from the assets of a business, offering greater protection for the owners in the cases of legal and regulatory matters. Legal entities can also enter contracts and have separate finances, tax liabilities, and other rights and obligations. An easy way to think about this is as if the business became a legal person. People can enter into contracts, own property, borrow money, and enter lawsuits. So in this sense, a legal entity gains the right to do so.

Decentralized Autonomous Organizations

In the world of Web3, it is crucial to remember what constitutes the recognition of a legal entity. World governments have different types of business structures they choose to recognize, which typically follow a centralized, hierarchical structure. In Web3, there is a move towards decentralized, non-hierarchical, "flat" ways of management and doing business, often done through the Decentralized Autonomous Organization, commonly abbreviated as DAO. In a DAO, the way the organization runs is dictated by a smart contract, with token holders in connection with that smart contract being members of the DAO. Because DAOs are non-hierarchical, they lack a centralized leadership structure and adopt management by the DAO members in connection with their token holdings. In other words, a DAO automates doing business, where token-holding members have the right to propose changes to how the company operates, and no one individual has the power to dictate everything. Additionally, due to their decentralized nature, DAOs can easily have members from many different countries, allowing for a borderless means of collaboration and business, free from the regulations of any single centralized government or authority. Many notable Web3 projects function as DAOs, such as Uniswap, MakerDAO, and ConstitutionDAO.

Part Two: The Introduction of Particles

With the onset of Web3, there is an abundance of new technology entering the market. However, our current means of management for this new technology have not evolved simultaneously. Software developers lack a direct interest in what they build. Often, within the corporate world, all their work becomes the sole intellectual property of the company that employs them. Individuals who wish to enter this market face a high entry level, with demands of time, capital, strategic planning, business knowledge, and leadership required on top of the idea they wish to develop, shutting out many people from contributing to this new technology boom. As this new technology emerges, how we protect and share it is not advancing, meaning a new approach to managing our intellectual property needs to be adopted to combat this unequal advancement.

To assist in the transition to Web3, we need to make an effort to deliver more developer rights, lower the entry-level, and modernize our intellectual property management methods. To accomplish this, we propose introducing a pipeline that protects, tracks, and makes it easier to do business with technology from the first line of code through distribution to the end user. We call this solution "Particles." Particles consist of a Smart Licensing system for both consumer and developer software licensing, a method of project management for early-stage and decentralized projects; all managed through a tokenized approach allowing for token-based funding opportunities.

Repository Particles (Smart Licensing)

Repository Particles are a means of tokenizing intellectual property and providing "smart licensing" for code, allowing for the formation of hybrid-source codebases. This method serves as a middle ground between open-source and closed-source code, offering a dynamic system that tailors to the needs of each developer. In an enterprise market, this simplifies the process of joint ventures and gives the means to reward developers more directly for their work by giving them direct ownership of what they build. For a consumer market, Repository Particles provide the end user more power over how they use their purchased software, with no restrictions on resale or trading, opening the door to open secondary markets.

Developer Licensing

In smart licensing, smart contracts mint ERC-721 Non-Fungible Tokens, which embody modernized copyright licenses. In traditional software licenses, such as the MIT and GNU licenses, you can find in open-source projects, terms and conditions are static, often contained in plaintext or markdown. Smart licenses construct from modular components, each with a human-readable and a machine-readable layer, as a modernization to traditional licensing. The experience is nearly identical to conventional copyright licenses in the human-readable layer, with terms and legal language available in plaintext or markdown. An overview of the human-readable modules is at the end of this Whitepaper. Each component has its functionality written into the smart contract on the machine-readable layer, allowing for a more dynamic licensing experience with actions able to execute automatically. Automatic actions include, but are not limited to, tracking, bilateral licensing, and royalty distribution and are a feature that the repository's host must support.

In an isolated environment, a smart license acts as a tradable authorization to adapt a project in the software development or record of software ownership in the consumer market. All financial transactions must occur on-chain, and users must report any off-chain activity to avoid violating the copyright and voiding the license to minimize any potential for abuse of the license in an isolated environment. Any new forks of the licensed work must also report new repository creation as a means of license validation to avoid copyright infringement. While these technological limitations exist now, this is primarily due to the domination of Web 2.0 technologies in the market, bringing confidence to the issue diminishing as Web3 adoption grows.

Particle Licenses are issued as ERC-721 Non-Fungible Tokens, meaning every license is unique. This approach is necessary for automated actions since every license will have a unique identifier. Crypto wallets are required to access repositories using smart licensing by holding a token and supplying SSH keys. Access to the repository grants utilizing the host's API to add SSH access to the token holder. Tokenized licenses are fully transferable and eligible for resale by the holder at any time. Upon the sale or transfer of a license token, the previously associated SSH key is removed from the repository to prevent maintaining access.

Tracking derivative work and executing bilateral licensing agreements can be accomplished by considering the downstream work from the source repository as a tree. All derivative works branch out from the original project, and even as more branches form, any point on the tree has a direct path back to the source. In derivative work tracking and bilateral licensing, any time a new branch emerges, the maintainers of the source repository have the right to view the branch and claim a license to work with the branch's adapted work of the source repository. This system is compatible with both public and private repositories. Through this dynamic licensing system, developers can customize the exact terms of how their work gets used, creating a spectrum of options between open-source and closed-source, called hybrid-source code.

Consumer Licensing

It is possible to create a consumer licensing system where all upstream rights to revenue are fulfilled at every point of sale, utilizing the same functionality in the developer smart licenses. This function is made possible because of the tracking introduced in the developer license. To best explain the consumer licensing process, see the example below:

Developer A has a project they have made available for other developers to remix via smart licensing called Project V. Later down the line, Developer B forks Project V, obtaining a developer license and creating Project W. Developer B spends time refining their work on Project W and later decides they wish to sell their work commercially. Project W also utilizes code from Projects X, Y, and Z. These projects are works of Developers C, D, and E. Because Developer B was licensed in all cases through a smart license, by verifying their token ownership, they can generate a consumer license of Project W, designed for delivering compiled code to the end user. Using this consumer license, Developer B can sell the Project W application. At any point of sale, all entitlements to revenue distribute automatically to Developers A, C, D, E, and any project maintainers they have recognized as having an entitlement to royalties.

In the example, consumer licenses generate using the developer license that automatically distributes royalties. This consumer licensing system creates a framework where the smart contract can split revenue per fractional intellectual property ownership through automated actions. For consumers, the software purchasing process remains nearly identical to buying a game, app, or utility software from any storefront, except with the added ability to resell and trade freely and across multiple marketplaces rather than being locked into a single option. This interoperability brings back freedom with digital goods that were only possible in the early days of the internet when everything was disk and cartridge-based. The only requirement for the consumer is to have a crypto wallet where tokens are stored, which can be an invisible component in some marketplaces where a marketplace user account can have an associated underlying wallet. This framework makes transitioning to Web3 technologies on the consumer side, reducing the need to educate people on new processes. For marketplaces, consumer smart licensing gives a base framework any marketplace can build itself off of rather than needing to design its own from the ground up.

Neo-Project Particles

Neo-Project Particles are a method of organizational management for the next generation of projects, compatible with traditional hierarchical organizational structures and newer Decentralized Autonomous Organizations (DAOs). Neo-Project Particles form a smart contract initiating a token supply through either ERC-20 or ERC-777 Fungible Token Standards. Tokens are held by individuals, organizations, or entities, making them Particle Members. Particle membership functions as the regulatory/executive power, with influence corresponding to token holdings.

Regarding intellectual property rights, Neo-Project Particles adopt the Repository Particle approach to management, with maintainer or source repository developer rights split amongst the token holders. This approach entitles every token holder to rights such as a share of royalties from licensing and allows every token holder to make personal use of the Particle's intellectual property if permitted by the Particle Rules.

The organizational structure of Neo-Project Particles is left open to interpretation so that the members of the projects utilizing them may decide for themselves what works best. The organizational structure, along with other terms of the Particle's existence, are contained in the Particle Rules, acting similarly to the bylaws of a corporation. The frame can range from mirroring traditional business hierarchies to a DAO. Neo-Project Particles empower the next generation of projects with for-profit fiscal sponsorship and tokenized member structures. This solution allows for engagement in economic and business activity from day one without an entity structure's bureaucratic and capital demands so early in its journey. For DAOs, this allows for a legal footing amidst uncertainty surrounding their existence and avoids the costly process of establishing a legal wrapper in a state where capital to finance such may be limited.

Fiscal sponsorship is an essential component in Neo-Project Particles, allowing them to engage in economic and business activities without themselves being legal entities. Fiscal sponsorship traditionally refers to non-profit organizations offering their legal and tax-exempt status and financial management to projects related to the non-profit organization's mission. With Particles, this practice applies in a for-profit sense, where a business entity offers its legal status and financial management services to a project or DAO in exchange for a fee, a share of the project or DAO's token supply, or both. The sponsoring entity enables Particles to own their intellectual property via the sponsoring entity and engages in financial transactions using fiat currency. This system allows early-stage projects to avoid bureaucratic hurdles and enable others to enjoy their decentralized nature and avoid high costs.

Additionally, this allows Particles to engage in a borderless means of business, optimizing international work and teams. This model is made possible by the sponsoring entity having multiple instances in different jurisdictions and economic areas. Lastly, this allows Particles to obtain trade names and other forms of intellectual property and to enter into legal agreements, to some extent permitted by the fiscal sponsor.

Tokenomics

Due to a token supply in the Neo-Project Particles, these tokens may be traded and sold freely to exchange equity in the Particles and raise capital. Tokens may initially be sold privately, in a peer-to-peer manner, to raise capital from accredited investors, friends and family, and founding Particle members. Neo-Project Particles are permitted to offer their tokens through a public sale, in compliance with necessary regulations, through an Initial Coin Offering (ICO), Initial Exchange Offering (IEO), or an Initial Decentralized Exchange Offering (IDO). All three of these methods enable Web3 initiatives, open and hybrid-source projects, and decentralized projects to sell portions of their token supplies as a means of bootstrapping their work. Anyone can trade particle tokens during and after a public sale across any exchange platform. An opt-in Particle Index Token is available to create a more dynamic token economy across Particles and as a possible means to deliver a greater return to individual Particles. The Particle Index functions similarly to stock market indexes such as the S&P 500. It is available for Neo-Project Particles, third-party Web3 initiatives, and decentralized projects regulated by token supply.

For all Particle types, the management of tokens takes place through an online platform that interfaces with the necessary blockchain. Users create accounts paired with their crypto wallets to authenticate their accounts and verify their token holdings. The functionality of this platform regarding Repository Particles is defined previously in the section on Repository Particles. For Neo-Project Particles, this platform functions as a means to create new Neo-Project Particles, with the creation and deployment of the smart contract taking place within the platform. The platform also functions as a decentralized token exchange with the functionality included for trade Particle tokens, compatible with the private and public sale methods described above.

Conclusion

Particles help encourage greater innovation through the tokenization of intellectual property rights and a system for the creation of hybrid-source codebases. Greater freedom is brought to consumer software with the freedom to trade and resell utilizing the consumer licensing method. They give more power back to developers on the enterprise level, giving them a direct stake in what they own rather than having it be the exclusive property of their employer. Particles offer a means of doing business for decentralized organizations and other projects, which would typically exclude them from such due to their legal status concerning being a legal entity. They grant a means to do business across borders seamlessly, allowing for forming international teams and engaging in activity on a global scale. Particles take luxuries previously only available to massive multinational corporations, making them available for small teams, individuals, and the wider Web3 and blockchain community. Particles help lower the entry-level to areas like software development and offer a bridge between our current world and the emerging world of Web3.

Affixed - Particle License Overview

The Particle License, derived from the Creative Commons license, is a copyright license for hybrid-source code, backed by a “smart contract” functioning as a machine-readable layer to this license and enabling compatibility with blockchain technology, decentralized systems, and other Web3 initiatives.

Attribution

Attribution <> A1: Others may distribute, adapt, and build upon this work, either commercially or non-commercial, under the condition that the original creator is credited.

Derivative Work

No Derivatives <> D1: Others may not share derivative works in their modified forms; others must give attribution.

Derivatives <> D2: Derivative work is permitted if others provide attribution.

Derivative Sharealike <> D3: Derivative work is permitted if others provide attribution; others must license any new creations under identical terms.

Derivative Bilateral <> D4: Derivative work is permitted if others provide attribution; any new creations are automatically licensed to the original creator under the terms the derivative work was licensed.

Derivative Sharealike Bilateral <> D5: Derivative work is permitted if others provide attribution; any new creations are automatically licensed to the original creator under the terms the derivative work was licensed, and others must license any new creations under identical terms.

Commercial Work

Non-Commercial <> C1: Others may not use licensed work commercially.

Commercial <> C2: Others may use licensed work commercially.

Royalties

No Royalties <> R1: No royalties are due to the original creator for any commercial use of the Licensed Material.

Transfer Royalties <> R2: No royalties are due to the original creator for any commercial use of the Licensed Material. Royalties are due for any profit generated through the transfer or sale of this license.

Commercial Royalties <> R3: Royalties are due to the original creator for any commercial use of the Licensed Material. No royalties are due for any profit generated through the transfer or sale of this license.

Commercial + Transfer Royalties <> R4: Royalties are due to the original creator for any commercial use of the Licensed Material. Royalties are due for any profit generated through the transfer or sale of this license.

Restriction

Unrestricted <> S1: The Licensed Work is unrestricted. Any sensitive information, including but not limited to source code, documentation, original files, and technical diagrams, is not required to be protected.

Restricted <> S2: The Licensed Work is restricted. Any sensitive information, including but not limited to source code, documentation, original files, and technical diagrams, is required to be protected.