Gitariq

WhitePaper

With the rapid growth of the decentralized ecosystem, there is an ever-increasing need for an open collaboration Platform.

This paper outlines Gitariq's vision for a decentralized code collaboration

Table of Contents

Table of Contents

1. Introduction

> **In real open source, you have the right to control your own destiny. - Linus Torvalds, Inventor of git**

Open source software has grown rapidly in recent years . Nearly every application we use today depends on open -source components .It has transformed the software industry and firmly established itself as the standard for development.

A key factor in the rise of open source is the emergence of version control systems such as Git and Mecurial ,along with platforms like GitHub and GitLab ,which simplified collaboration and made global developer participation easier and more convenient.

The growing popularity of Git, and later, Git hosting platforms , gradually lowered the barriers to joining open source projects . With platforms like GitHub ,contributors no longer had to search for repository links ,prepare manual patches , or subscribe to mailing lists just to ensure their work reached the right maintainers .Instead ,they could fork a repository with a single click and submit a pull request directly from their chosen branch .This simplified process made contributing significantly easier ,encouraging more developers across the world to engage actively with open source communities.

Code hosting platforms have enabled social coding and fostered global developer communities. By making code freely and publicaly, accessible, they have significantly, reduced the cost and complexity of software development, fueling rapid growth and accelerating technological innovation worldwide. Code hosting platforms have grown into essential resources for both education and business.

GitHub remains a prime example.

In addition to offering installers for countless applications, GitHub also hosts the source code for millions of projects, making it accessible for anyone to review and learn from. By archiving past versions of code, it allows users to trace the entire development journey of a project.
This feature has made GitHub an invaluable teaching resource for developers worldwide.

The popularity and advantages of code hosting platforms have made them central to modern software development. Today, the majority of open-source work takes place on platforms like GitHub and GitLab. However, this reliance on centralized providers is concerning.

It places critical open-source projects at the mercy of private companies, whose policies and business interests may dictate access and control. Such centralization undermines the very principles of the open-source movement, which are rooted in freedom, transparency, and community ownership.

Conventional code hosting platforms also present several additional significant challenges, including:

- **Dominance of centralized providers**
- **Content suppression**
- **Absence of community role in governance**
- **Opacity in operations**
- **No rewards for users in platform growth**
- **Single source of failure**
- **No native open source motivation**
- **Not optimized for decentralized ecosystem**
- **Lack of collaboration portability**

These challenges harm open-source development communities, creating the urgent need to reinvent collaborative platforms for open-source software development.

Gitariq overcomes the limitations of traditional platforms by delivering a decentralized, developer-centric code hosting and collaboration system, transparently governed by its community of users.

Gitariq introduces a system for storing gitrepositories on adecentralised networkthat is accessible, censorship -resistant ,andempowers communities to create ,co-own,contribute ,andgovern theiropen-source software collaboratively.

Gitariq aims to reinvent open-source funding and ensure the sustainability of open- source software development. It will incentivize contributors by building a global, virtual economy centered on their work and impact.

Gitariq's mission is to establish a new model for open-source software development and decentralized collaboration where every user is a stakeholder, innovators earn rewards, contributors stay motivated, and the community retains true power.

2. Nomenclature

- **Version Control System (VCS) –** A version control system is a process management tool that records and manages changes to a file system. Every update is stored as a version, allowing users to revisit and track specific revisions later.

- **git -** Git is a free, open-source, distributed version control system built to track changes in computer files and coordinate collaboration on those files among multiple contributors with reliability, efficiency, and speed.

- **Git Hub -** Centralized Repository Hosting Platform

- **Repository -** A set of source-code files and folders spanning multiple versions of a projects

- **Commit -** A commit is an atomic update to the repository, showing lines added and removed, also referred to as the diff.

- **Branch -** Independent development branch

- **master/main -** The main branch is GitHub's default branch.

- **Organization -** Organizations let companies or individuals group multiple repositories together under a single shared view.

- **Packfile -** Git works internally with objects such as commits and blobs. These objects exist on the user's system, but transferring them can become costly if the repository has many changes. To optimize this, Git compresses the objects into a single file called a packfile. A packfile is essentially a collection of multiple objects stored together using an efficient delta compression scheme.

- **Pull Request**-When an individual wishes to contribute to an open-source project,

the typical GitHub workflow is: Fork the repository, push changes to the fork, and create a pull request. The pull request is then reviewed by the project maintainer, who either merges the contribution or requests further changes before approval.

- **Issues -** Tickets represent bugs, issues, features, or enhancements proposed by contributors or maintainers.
- **Proposals -** Decentralized projects must propose improvements, features, or updates for voting. Typically, the network proceeds with implementation only if the proposal is approved.
- **Destructuring -** Unpacking packfiles into objects
- **Lazy-Loading -** Retrieving objects dynamically whenever requested by the user.
- **Clone -** Cloning locally, typically with versioning, the source code of a repository.
- **TEE -** Secure Execution Environment
- **Fork –** Forking a repository allows copying it to another user or organization while preserving a link, enabling updates whenever the original base repository receives changes.
- **Remote -** Remote transport URL, for example: git:// or gitariq/.
- **TGE -** A Token Generation Event (TGE) is a limited -time business and technical process that includes the creation of a token on a blockchain network and its subsequent release and distribution into the open market for trading.
- **Validator -** A blockchain validator is an entity responsible for verifying transactions within a blockchain network. After validation, these transactions are confirmed and permanently recorded into the distributed public ledger.

3. Domain Outlook

The open-source software market has witnessed rapid growth in recent years. Increasingly, individuals, organizations, and enterprises are recognizing the potential of open source and actively engaging in the movement. The OSS community has produced many of the world's most critical and widely used technologies, including operating systems, web browsers, and databases. Modern life would not function, or at least not function effectively, without open- source software . A 2008 study by The Standish Group estimated that "free open -source software saves consumers $60 billion annually in IT costs."

Code hosting and collaboration platform have been on intrumental in the exapnsion of open source, introducing it to mainstream audiences. These platforms foster large comunnities dedicated to building, improving, and maintaining open source software on a global scale.

Developers worldwide rely on code collaboration platforms to connect, innovate, and build software. Over the past decade, nearly 60 million developers have used these platforms to create more than 300 million source code repositories. GitHub, the largest centralized platform, hosts around 56 million developers and over 220 million repositories on its servers. In just the past year, developers made 1.9 billion contributions and launched 60 million new repositories on GitHub, reflecting the enormous scale and influence of centralized platforms in the open-source ecosystem.

Most leading open -source projects today are built and maintained on GitHub . While large corporations possess the resources and funding to sustain development , many open-source projects are managed by individuals in their spare time. However, maintaining these projects still requires effort , time , and often incurs additional costs . Financial support remains essential, as monetary contributions play a crucial role in enabling continued development and advancing the growth of open-source initiatives.

3.1 Impact on Open Source

By engaging actively with OSS communities — whether by sponsoring events ,funding developers ,or backing project initiatives — companies enhance employer branding and strengthen their presence among large developer networks . The primary motivations behind corporate contributions to open-source projects include visibility ,innovation ,cost efficiency and ecosystem growth.

- Generating revenue through complementary services like training ,consultancy, certifications, and support.
- Boosting innovation by launching new products quicker, superior, and ahead of competitors.
- Lowering development and maintenance expenses by leveraging contributions from the wider open source community.

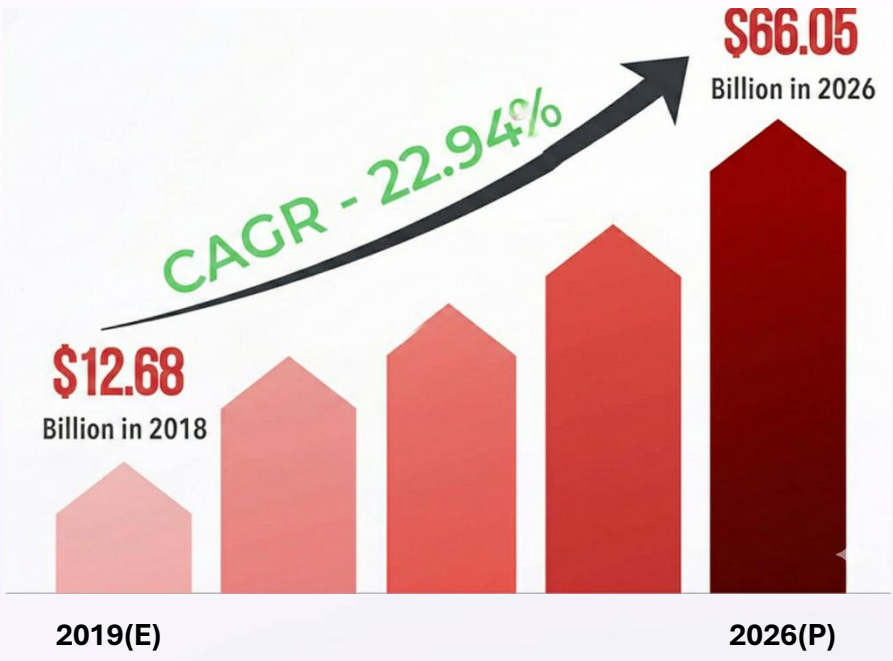
Microsoft ,Google ,Intel ,and Facebook —none originally open -source companies, now actively contribute to numerous open -source projects on GitHub .In 2018 ,Microsoft employees ranked among the most prolific contributors, recording nearly 7,700 unique submissions (with 4, 550 total contributors in 2017). Google employees were also significant participants, collectively making 5,500 contributions that year. While many supported smaller independent projects , a large portion strengthened Google's own open- source initiatives, including Kubernetes, Istio, and Knative, emphasizing how corporations strategically leverage open - source contributions to drive innovation, expand influence, and advance the broader ecosystem in targeted ways.

3.1.1 Open Source Framework

Open-source service describes software in which copyright holders release the source code for users to access, modify, and study. The software can then be redistributed across any platform. Globally, open-source services help organizations achieve IT strategic goals, increase efficiency, improve performance, and enhance productivity while driving innovation and delivering sustainable technology-enabled outcomes. Key drivers of the

open -source services market include universal accessibility of source code , reduced costs compared to proprietary alternatives , and seamless interoperability .The open- source services ecosystem is supported by major vendors such as Red Hat , Accenture , Wipro , IBM , Infosys , Cisco Systems , Atos , HCL , HPE , and Oracle , all playing active roles.

The open -source services market was valued at USD 12 .68 billion in 2018 and is projected to reach USD 66.05 billion by 2026, growing at a CAGR of 22.94% from 2019 to 2026.



3.2 Dispersed Open Source Funding

Funds secured by open-source projects are allocated to compensating contributors , covering operational expenses like hosting , and fostering the creation of new features or innovations that drive the project's sustainable development and long-term growth.

The common approaches used by open-source projects to secure funding are outlined below:

3.2.1 Development Grants

Various software foundations and companies offer grants to fund open-source initiatives . Such funding is usually distributed in batches, with a designated amount released each year and allocated across multiple projects . Grant-based support enables developers to plan strategically and implement projects over extended periods , ensuring more sustainable growth and providing consistent backing for continuous contributions within the open-source ecosystem.

4. Constraints

4.1 Centralization Monopoly

This reflects the most visible aspect of centralized platforms . While open source and decentralization inherently challenge monopoly , large corporations still dominate and establish themselves at the center of the open -source ecosystem , directly undermining its core guiding principles.

GitHub , the leading centralized platform ,hosts over 56 million developers .In the past year alone , these developers created more than 60 million repositories. With developer participation and open -source contributions rising rapidly , it becomes critical to assess the long -term permanence and reliability of GitHub as the primary hub for global open -source collaboration and community-driven software innovation.

Open source depends entirely on millions of developer contributions, with GitHub expected to reliably host their code indefinitely.

As most developers depend on GitHub for their projects , a challenge emerges where open- source communities feel pressured to host solely on GitHub to attract contributors , leaving them with few viable alternatives for meaningful collaboration.

Certain decentralized projects recognize the contradiction and are transitioning toward self-hosted open-source alternatives.

4.2 Access Restriction

Centralized platforms function as monopolies, making censorship both simple and inevitable. GitHub ’s core weakness is enabling stakeholders to censor repositories or content at will, frequently disadvantaging developers and stifling broader open-source innovation.

For instance, GitHub recently took down the popular open-source repository youtube-dl, along with its public forks, after receiving a DMCA takedown notice submitted by the RIAA. The decision triggered criticism and protest from youtube-dl users and the broader open-source community. Public attention to the takedown created a Streisand effect, further amplifying the issue. Users widely reposted the software's source code across the internet in multiple formats. On Twitter, some even shared images encoding the entire youtube-dl source code with varying pixel colors. Ultimately, GitHub was forced to reverse its action and reinstate the youtube-dl repository on its platform, highlighting flaws in centralized control.

GitHub's move has made the crypto community quite suspicious of the platform, and some proponents are scared that government forces can take down cryptocurrency codebases like Bitcoin's. This possibility has encouraged many developers to start looking at alternative platforms to host their code.

Because of GitHub's monopoly and heavy concentration of developers, it becomes an easy target for governments and nations seeking to impose restrictions or advance political agendas through centralized control.

GitHub has repeatedly faced censorship attempts by governments in countries such as China, India, Russia, and Turkey, employing tactics like denial-of-service attacks and local ISP blocks on GitHub's servers. In each case, GitHub was eventually unblocked either due to backlash from technology businesses and users or through compliance measures taken by GitHub itself to meet government demands.

Another example of government-driven censorship occurred when GitHub restricted users in Iran, Syria, Crimea, and other sanctioned regions from accessing parts of its service in order to comply with United States trade regulations.
GitHub also engages in significant censorship as a result of the Scunthorpe problem.

The Scunthorpe problem occurs because computers can easily detect strings of text but struggle to interpret words and complex phrases within diverse cultural and contextual settings, making the task highly challenging. Consequently, broad blocking rules often generate false positives, restricting legitimate phrases. GitHub also experienced this issue, leading to unintended censorship and highlighting the limitations of centralized automated moderation in balancing accuracy with freedom of expression.

Another prominent example of the censorship problem can be seen in how the Chinese government has targeted GitHub. The Chinese government, recognized for its censorship of foreign services, has witnessed rapid growth in open-source development. GitHub's centralization has enabled China to specifically target the platform, employing multiple methods of attack to exert control and disrupt developer access

4.3 Forced Directives

Developer access The 'Open Source' Phenomenon is a community-driven movement, yet GitHub has positioned itself at its center over the years, In reality, GitHub is merely a collaboration platform, while developers provide the code and sustain the ecosystem. Without the community's contribution, GitHub would lose its value and could be sourceforge Collaboration were removed. Dispite being Build by the community. Developers have influence over GitHub's product or policy decisions. This disconnect fosters distrust as the platform thrives on community efforts but denies them meaningful participation in shaping its direction or governance.

4.4 Open and Accountable

Most popular code hosting and collaboration platforms today , like GitHub , are not open source. They lack transparency regarding how user data is utilized and how specific features are implemented. Users rely on these products without fully understanding their operation or the implications for privacy and security. Platform owners retain the authority to change terms or modify product behavior at will, often without notifying or seeking consent from the user community.

4.5 Absent User Rewards

Despite GitHub's monopoly over open-source code collaboration, it lacks any incentivization mechanism that directly aligns developers with the platform's overall product growth.

4.6 Fortified Access

Source code repositories on centralized platforms remain vulnerable if security is compromised. Unauthorized actors could alter repositories, inserting malicious commits that harm users, disrupt projects, or introduce hidden vulnerabilities, ultimately undermining trust in both the platform and the software it hosts. Blockchain resolves this issue effectively,

as every modification must be cryptographically signed by the repository owner or authorized members with access, ensuring authenticity, accountability, and protection against unauthorized alterations.

4.7 Systemic Fragility

Hosting all code on a centralized platform creates a single point of failure . If such a platform experiences a catastrophic event —such as a hack , data breach , bankruptcy ,or sudden shutdown —entire codebases risk being lost , leaving developers and organizations vulnerable without reliable alternatives for continuity, resilience, or long -term preservation of their work.

Frequent service interruptions on GitHub have demonstrated that this scenario is entirely possible,reinforcing the fact that GitHub operates with a vulnerable single point of failure.

By design, each developer maintains a full copy of the repository , minimizing direct risk of data loss for companies using platforms like GitHub. The challenge emerges, however, when GitHub becomes deeply integrated into the core development workflows of multiple organizations , creating dependency risks and potential disruptions tied to its centralized control.

For instance , companies whose developers work locally and push code to GitHub for deployments have no control over the underlying infrastructure . If deployment is required while GitHub is down, they lose the ability to proceed , leaving critical operations dependent on the platform's centralized availability and stability.

4.8 Open Source Incentivization

Open-source incentivization largely depends on sponsorships or voluntary donations, despite open source forming the backbone of nearly all modern software. Furthermore, as every cryptocurrency-based decentralized network is inherently open source, incentivization should not rely solely on public donations. Instead, it must be integrated directly into the economic architecture of decentralized networks, ensuring sustainability and consistent rewards for contributors.

4.9 Incompatible with Decentralized Models

The past decade has witnessed remarkable expansion in the decentralized ecosystem.

The transition from centralized , proprietary code to decentralized networks and open-source models has highlighted the urgent need for tailored solutions that truly empower communities.

The existing systems built by centralized players enable code collaboration within a controlled , top-down environment . What is needed instead is a bottom -up approach . While core collaboration processes may remain the same , new models —incorporating proposals , governance , and community -driven access—must be explored to align with the principles of decentralization and ensure broader participation in decision-making.

4.10 Collaboration Lock-In

The absence of collaboration export is a tactic employed by centralized platforms to tightly bind workflows to their systems while detaching them from the actual source code . For developers , this represents the greatest pain point, as all collaboration history can be lost if account or repository access is restricted or removed . This creates a form of vendor lock-in, leaving no viable option for migration and trapping communities within a single platform ’s boundaries, regardless of their needs or preferences.

5. The Framework

Gitariq is a decentralized code hosting and collaboration platform powered by open-source blockchain protocols .Combining modern software development practices , Gitariq delivers robust collaboration features supported by a secure and censorship -resistant decentralized storage infrastructure.

Core Features of Gitariq

- **Immutable Storage:** Through blockchain integration, Gitariq offers developers a secure, durable, and easily accessible space to preserve and recover their code.
- **Distributed:** Gitariq is powered by a network of validators and participants who secure and govern the system, removing any central authority and fostering transparent, resilient, and community-led decision-making.
- **Community Rule:** On Gitariq, every platform decision is made with open community participation. Users propose initiatives, vote transparently, and directly shape the platform's direction and long-term growth.
- **Content Freedom:** On Gitariq, no central authority can remove repositories. Instead, the community collectively defines content policies and manages moderation, ensuring fairness, resilience, and open access for all developers.
- **Resilient Infrastructure :** Gitariq ensures uninterrupted service through its decentralized architecture . A distributed network delivers stability , redundancy , and high availability , eliminating single points of failure and strengthening overall platform reliability.
- **Cryptographic Security:** Repositories on Gitariq are protected with public-key cryptography, making them resistant to unauthorized access, tampering, and malicious attacks.
- **Open Transparency:** Gitariq's source code is fully open, ensuring workflows remain visible for developers worldwide to review, verify, and improve.
- **Stakeholder Rewards:** Users of Gitariq become stakeholders, earning incentives as the platform expands and its ecosystem grows stronger.
- **Built -in Incentives :** Contributor incentivization is embedded into Gitariq 's design , motivating developers to actively engage and contribute to open-source projects.
- **Seamless Interoperability :** Gitariq integrates with Solona -native solutions to enable smooth interoperability across decentralized applications and blockchain ecosystems.

5.1 MVP Rollout

The MVP implementation of Gitariq mainly consisted of a git based Code hosting platform built using Arweave. Arweave is a blockchain that enables permanent, low-cost storage of regular digital data . Arweave provides us with permanent storage , where we pay only once for storage , and retrieve it freely forever . Arweave protocol and its incentivization mechanisms ensure that all the data are accessible permanently and do not depend on a centralized service provider's trust.

In Gitariq's initial implementation , repository objects were stored directly on Arweave , even though processing them into packfiles before storage would have been more efficient. This approach was chosen because generating packfiles required an intermediary server, which would have added complexity and overhead to the application . As a result , while Gitariq functioned effectively as a code hosting platform , this design introduced certain limitations that highlighted the need for optimization in future iterations.

5.1.1 Insights from the Gitariq MVP

During Gitariq's MVP build on Arweave:

As a built -in capability , Git equips developers with remote helpers that allow execution of custom logic directly from the command line. For example , when a developer adds a GitHub remote beginning with git://, the CLI seamlessly interacts with the git server , maintaining the same familiar workflow for developers.

When a developer uses Gitariq and adds a remote starting with gitariq://, the git command line searches for a helper named git-remote -gitariq on the user's system . If installed (distributed via npm), git forwards sub-commands (push, pull, etc.) to git-remote -gitariq for handling . On each code push , the CLI tool retrieves the head reference from storage , uploads new objects , and updates the branch reference to the latest commit hash , ensuring synchronization between the local repository and Gitariq's decentralized infrastructure.

While this method of uploading individual objects to decentralized storage allows users to lazy - load them on demand through the web interface,it creates inefficiencies for larger repositories. Such repositories often contain extensive history,which significantly increases the initial upload size and restricts performance during the onboarding process. Additionally , this process takes significantly

longer to complete and incurs higher costs. For example, the Linux repository contains ~8,000 , 000 objects , totaling nearly 60GB in raw git objects , whereas its compressed packfile is only about 5GB, highlighting the efficiency gap.

Git enables the use of packfiles to compress objects , improving data transfer efficiency . However, if Gitariq were to adopt this method and push packfiles containing new objects with every commit , several challenges emerge that complicate storage and synchronization workflows.

Clone Process

The user would be required to retrieve all packfiles (representing each git push) for that specific repository from storage, making the cloning process significantly slower and more resource-intensive.

Browser Workflow

Lazy loading of git objects would not be feasible . Instead , users would be forced to download every packfile before rendering the repository in the browser . For larger repositories , this significantly increases waiting times — often several minutes — leading to a poor user experience and reduced accessibility for developers browsing or reviewing code online.

Key Insight

- Gitariq requires optimized use of permanent decentralized storage, such as Arweave or IPFS, for efficiently managing and storing packfiles.
- Gitariq needs a scalable smart contract layer to handle state management for diverse user-driven actions across the platform.
- Gitariq requires a compute layer designed to efficiently execute git operations and support CI/CD workflows for developers.

5.2 Architecture Limits

Building a truly decentralized version of a complex application like GitHub introduces several engineering challenges, which are outlined in detail below:

5.2.1 Git Framework

Git employs an intelligent protocol for transferring objects between the client and server . It identifies which objects are required and compresses them into an optimized packfile format, enabling efficient transfer of git objects during fetch or push operations.

A decentralized implementation cannot replicate this optimization directly at the data layer , since blockchains allow data to be appended but not restructured . The MVP relied on storing individual git objects, which proved inefficient when handling large repositories with extensive histories.

5.2.2 Response Time

The introduction of decentralized systems leads to latency issues , caused by delays in broadcasting transactions and nodes reaching consensus. This latency creates a poor user experience and needs to be resolved.

5.2.3 Multi-Chain Integration

Blockchains are generally optimized for specific functions. To design a complex application comparable to GitHub, integrating multiple specialized solutions is essential in order to deliver the required features and overall functionality.

5.3 Gitariq Framework

Gitariq incorporates the following components to effectively address the challenges outlined above and enhance platform performance —

- Decentralized Compute: A distributed compute layer will process packfiles, making the platform significantly more efficient and scalable when managing large repositories.

- Consensus Engine: Gitariq will operate on a delegated Proof-of-Stake (dPoS) Byzantine Fault Tolerant (BFT) consensus mechanism, addressing latency issues and enhancing transaction throughput.

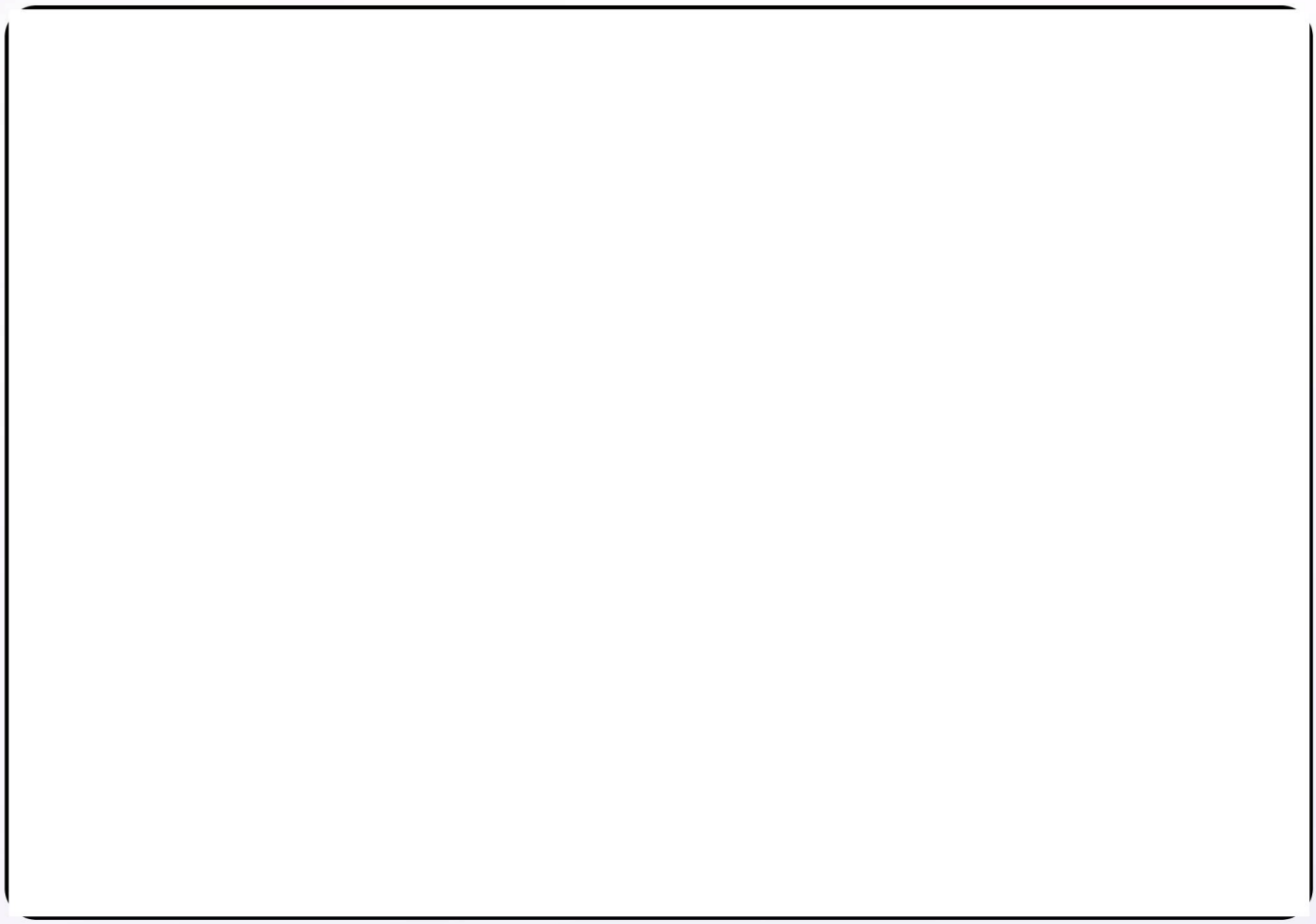
- Cross-Chain Connectivity: To enable interoperability, Gitariq will integrate with Solona-native cross-chain communication protocols, ensuring seamless interaction with other decentralized ecosystems.

6. Gitariq Infrastructure

The core components of Gitariq include —

A Gitariq main chain built using Solona infrastructure. It manages application logic, access permissions, and metadata associated with repository storage and collaboration workflows.

- A decentralized compute layer that manages packfile processing alongside CI/ CD workflows.
- A redundant data storage layer using IPFS/Filecoin.
- Cross-chain interface for blockchains to seamlessly integrate their development workflows with Gitariq.



6.1 Gitariq Core Chain

The Gitariq main chain is an application -specific blockchain optimized for decentralized source code collaboration . It is built on Solona infrastructure and will operate using a delegated Proof-of-Stake (dPoS) Byzantine Fault Tolerant (BFT) consensus mechanism.

The Gitariq main chain will achieve interoperability with other blockchains through Solona-native cross-chain communication protocols.

The Gitariq main chain will be secured by a decentralized network of validators and the native Gitariq token, $GITQ.

The main chain will manage the application state and enforce access controls. It will be responsible for handling the following core requirements —

6.1.1 User Address Mapping

Users can reserve unique usernames linked to their public address , improving overall experience by eliminating the need to rely on long, complex addresses . Additionally , these reserved names are transferable, allowing users to reassign them to others if desired, adding flexibility and usability to identity management within the Gitariq ecosystem.

6.1.2 User, Organization, and Repository settings

User

User-specific configurations and encrypted secrets will be handled under this contract.

- Follow or unfollow other users
- Add or modify encrypted secrets

Repository

All repository -related settings , including ownership and access permissions , are managed under this contract . This ensures transparency and provides a complete audit trail for every repository-related action.

Examples of owner-controlled actions include:

- Transfer ownership of a repository
- Grant write access to new users
- Revoke access from existing users

Organization

All organization -related settings , including ownership and access permissions , are managed under this contract. Decision-making in repositories owned by organizations differs from those managed by individual users. Once an organization is created , all subsequent actions require approval from other members , ensuring collaborative governance and preventing unilateral control within organizational repositories.

Organization Workflow:

- A user creates an organization
- The owner adds another member
- Adding new members requires approval from all existing members
- Removing a member also requires voting
- Decisions in any repository managed under an organization require member consensus, eliminating the need for third-party dispute resolution since the process is hardcoded within the contract.

6.1.3 Process Framework

Collaboration workflows are encoded within the contract , ensuring that processes such as approvals , merges , and pull request closures are executed transparently . For instance , a contributor can configure a checklist (e.g., code quality, build status) in the contract for a pull request to qualify for merging . No individual can bypass this rule unless the checklist itself is modified through a stakeholder voting process , ensuring fairness and decentralized governance.

6.1.4 Governance Workflows

Governance in org/repo:

- Voting on proposals
- Treasury management
- Granting write access
- Revoking user access

Governance in Gitariq:

- Voting on Gitariq Improvement Proposals
- Oversight of community treasury
- Configuring platform fees

6.1.5 Git pointers

On every git push, git pointers for the respective branches are updated on-chain. Only authorized users can modify these pointers, and all updates are easily auditable.

6.1.6 Auxiliary data linked to git repository storage

All code submissions are efficiently stored as Packfiles , with the decentralized compute infrastructure handling their processing and caching for quick access. This ensures that the front-end application delivers loose git objects for faster file browsing, while git clients receive a single compressed Packfile to minimize data transfer.

6.2 Distributed Processing Layer

The compute layer will handle the transformation of packfiles. It will process them into:

- A consolidated packfile for users to clone.
- A decomposed packfile (loose git objects from the consolidated file) to support the web app and enable efficient lazy loading of the file browser view.

The distributed compute infrastructure will also manage:

- CI/CD pipelines
- Automated code validation
- Extraction of git loose objects for enhanced rendering within the web application

6.3 Cross-Chain Communication Bridge

The blockchain landscape is often viewed as a fragmented collection of isolated decentralized networks that lack the ability to exchange data or interact seamlessly, leading to tribalism and competition rather than collaboration.

Gitariq will integrate with the Cosmos IBC (Inter-Blockchain Communication) layer to enable seamless interoperability between its application layer and other blockchains that support the Cosmos IBC standard.

Gitariq brings powerful utility to the decentralized cryptocurrency ecosystem , where nearly all networks reserve portions of their tokens to incentivize community-driven development. By leveraging Gitariq 's decentralized repository management , these networks can ensure transparency and verifiable proof of code contributions , enabling developers to be fairly rewarded from network reserves.

Furthermore , the IBC cross -chain payment channel , secured by the Cosmos Hub , allows these networks to seamlessly fund tasks such as bounties, issues, and proposals directly on Gitariq.

6.4 Resilient Data Layer

The resilient data layer will enable users to back up their repositories on alternative storage networks , providing an added layer of security . Users can choose Arweave or any other preferred decentralized storage solution.

Arweave Gitariq IBC Bridge

When Arweave is used as a redundant storage layer, payments for storing data are required in AR tokens. To simplify user experience , Gitariq abstracts this complexity so users aren't burdened with handling multiple tokens directly.

To ensure all platform activity is powered by the GITQ , an Arweave –Cosmos IBC Bridge must be implemented . This bridge will enable Gitariq and other projects within the Cosmos ecosystem to leverage IBC, seamlessly integrating Gitariq with the broader network and delivering a unified developer experience.

7. Core Benefits

7.1 Open Source Incentivization

Traditional platforms lack built-in mechanisms to sustain open-source projects. Gitariq addresses this gap by embedding incentivization directly into the ecosystem through the GITQ token.

Incentivization will support funding for open-source projects, drive ecosystem growth, and reward maintainers and contributors for their valuable efforts.

7.2 DAOs Governing Code

At present , many so-called decentralized projects remain centralized in their development process . Commit access is often restricted to a handful of developers , leaving true stakeholders unaware of what changes are being implemented in the codebase.

To address this , Gitariq will integrate DAOs with source code management , ensuring full transparency in the development process . This approach gives stakeholders greater visibility and involvement in software releases.

7.3 Dynamic Open Source Licenses

Licenses are a critical element of open -source software . Without a license , open -source code is unsafe to use , as the original author can still pursue legal claims or demand royalties . Attaching a license is the only way to ensure that code is truly open source and freely accessible for use , modification, and distribution.

However , challenges remain . Open-source licenses are often subjective , with interpretations varying based on how the software is technically used . This makes it difficult to assess legal risks, particularly for developers who are not legal experts.

Take the recent example of Uniswap. In its third iteration, Uniswap introduced a license to deter direct copycats from replicating its codebase . While effective in limiting wholesale duplication , this decision marked a significant shift away from the open -source ethos that defines most cryptocurrency projects. The move sparked debate within the community , with opinions divided on whether it protects innovation or undermines openness.

Uniswap 's decision was essentially aimed at preventing copycats from exploiting their codebase and eroding potential profits. They adopted a Business Source License (BSL), which restricts others from using the code in production for two years, after which it becomes open for use.

Some argue that Uniswap 's licensing decision was justified , as competitors were freely leveraging code that Uniswap 's developers had invested significant effort in building . Others , however , see the move as a violation of open -source principles , warning that it could suppress innovation —since competition has always been a key driver of progress in the ecosystem.

While Uniswap 's new license may slow competitors from exploiting its code , enforcing such restrictions across a decentralized ecosystem remains highly challenging.

Key Issues-

- **Profits generated by users who copy the code often fail to reach the original authors challenging.**
- **Enforcing licenses against copycats across a decentralized ecosystem is highly**

Gitariq aims to solve both of these challenges within its platform . Through smart contracts , all users will be clearly informed of the open -source license attached to the code they clone or contribute to. Additionally , Gitariq 's incentivization model will ensure that original creators receive fair royalties, aligning rewards with their contributions.

8. Conclusion

Gitariq is committed to building a decentralized code hosting and collaboration platform where users can create , manage , and share open -source projects without the risks of censorship, downtime, or the limitations of centralized platforms.

Gitariq envisions a platform where user voices and experiences truly matter, with openness and transparency embedded through community-driven governance.

Gitariq aspires to be the hub where passionate developers worldwide unite to build the best open -source software . Through its GITQ -powered incentive model , Gitariq ensures contributors remain motivated and rewarded, driving sustainable open-source development.