Ad-Tech: Decentralized Marketing, a promising direction for the future of blockchain technology

Dr. Alan Tominey
contact@gorki.dev

February 5, 2024
v1.0

Abstract

This paper presents a novel blockchain-based solution that addresses critical challenges in the integration of Web 2.0 and Web 3.0 technologies. Building on the foundational ease of use and global reach of early 2000s web applications, it leverages the freemium business model's success in attracting and engaging users. The solution uniquely combines the decentralization of blockchain with a new capability to handle on-chain data effectively, run a concurrent leaderless consensus protocol for real-world scaling, and interoperate with existing Web 2.0 technologies.

One significant challenge in blockchain adoption is the fee-based nature of transactions, which acts as a barrier to entry for users accustomed to the free access of Web 2.0 applications. Our approach introduces a groundbreaking API that allows blockchain resources—such as data storage, retrieval, and smart contract execution—to be funded through sponsored content. This innovation significantly reduces the direct cost burden on users, fostering wider acceptance and use of decentralized apps (dApps).

Moreover, our solution, Gorki, bridges the gap between legacy marketing systems and the emerging Web 3.0 paradigm. It ensures compliance with security and privacy regulations, such as the EU's right to be forgotten, and seamlessly integrates with existing content systems and marketing channels. Gorki enhances interoperability, enabling the implementation of marketing strategies previously infeasible in the blockchain space. It offers users control over their data and advertising preferences through tokenized incentives, aligning with modern privacy concerns.

In essence, this paper introduces a transformative blockchain technology that harmonizes the user-centric approach of Web 2.0 with the decentralized, privacy-respecting ethos of Web 3.0, paving the way for a new era of digital interaction and economic models in the blockchain domain.

1 Introduction

1.1 Historical context and application

In the early 2000s, the emergence of HTML/HTTP technology marked a significant shift in application development and accessibility. This period witnessed the creation of a new category of applications, characterized by their ease of use and global reach. Individuals for the first time were empowered to develop comprehensive applications, sparking a widespread enthusiasm in the field. Concurrently, a novel model for content delivery emerged, predominantly centred around sponsorship. This model enabled web-based applications to provide intricate content and services to a worldwide audience at no direct cost to the users. The underlying economic structure of this model relied on advertising revenue. In essence, it made an exchange of value possible, users provide access to their data in exchange for access to applications that provide value for them. This development, the creation of the "freemium" model, signified a paradigm shift in the digital economy, intertwining user experience with commercial interests in an unprecedented manner. In today's digital landscape, the freemium business model has emerged as a powerful strategy to attract and engage users. By offering core functionalities at no cost and reserving advanced features for a premium, businesses are effectively lowering the barriers to entry, encouraging mass adoption, and fostering user loyalty. This approach has been notably successful in various web-based and mobile applications, allowing companies to establish a broad user base quickly. When executed with precision, the freemium model can be a win-win: users benefit from free access to basic tools and services, while businesses generate revenue from those who see value in upgrading. Leveraging this model for newer technologies can revolutionize how consumers interact with platforms, especially in sectors where upfront costs traditionally deter user engagement. Blockchain's ability to decentralize data and compute services extends beyond payment systems, potentially revolutionizing applications like email, social media, maps, commerce, and logistics. However, decentralized apps (dApps) face challenges, particularly in their business model, where transactions related to blockchain, such as data storage, retrieval, and compute actions, incur immediate fees. This contrasts with traditional Web apps that operate on cloud services with more extended reconciliation periods. Similarly, a significant pain point for blockchain adoption is the need to continually load wallets with tokens to perform even rudimentary activities. In many cases leading people to centralised exchanges to on-board tokens they then must perform numerous steps to get tokens to individual wallet accounts. Creating a portal for the freemium model in the hands of app developers is a methodology to potentially lower this barrier to entry, whilst simultaneously creating a secondary avenue for revenue generation for the developers. A significant opportunity arises from these challenges: developing an API that allows blockchain resources (data storage, retrieval, and smart contract execution) to be funded through sponsored content. This approach would alleviate the burden of direct costs on users, a major friction point in dApp adoption. It proposes a model where users can access stored data either by paying a fee or receiving the data with sponsored content, thus removing direct cost barriers. This innovation, adds a crucial layer to the blockchain, enabling dApp creators to offer services without imposing direct costs on end users. This shift not only eases user adoption but also opens new opportunities for dApp developers and their consumers. Essentially, it integrates the commercial sponsorship model of Web 2.0 into the blockchain space, harmonizing user experience with the decentralized nature of blockchain technology. To do this there is a need to bridge between legacy marketing systems and any new web 3.0 paradigm. At the same time there is also a need to overcome the "fee-based" nature of modern blockchains. Integrating marketing APIs into a native blockchain may be a great way to "kill two birds with one stone". Marketing and engagement models will need to be able to bridge from existing systems and databases, for several reasons, security, and privacy being some of them (things like the right to be forgotten in the EU), as well as integration into existing content systems and marketing channels, interoperability is critical for this. Interoperability allows developers to onboard marketing paradigms that were not previously possible in the blockchain space, moving from Web 2.0 to Web 3.0. Imagine a decentralised application that would allow the user to turn off advertising through a function of their wallet. Alternatively, think of an application that would allow the user to choose which information they would be willing to share based on tokenised incentives. Gorki opens the possibility for data intense platforms to monetise through marketing or through utility token payments, depending on the user's preference for privacy.

1.2 The vision

Blockchains revolutionize backend computational infrastructure by tokenizing its operational costs and decentralizing the system. This approach democratizes access to datacentre economies, similar to how Uber broadened the hired transport market to include millions of individuals outside traditional taxi companies. In this context, it is possible to tokenize the sponsorship and advertising aspects of Web3.0. By expanding the attention economy, which has already driven significant platform adoption and user engagement, this strategy presents an economic opportunity for decentralized applications (dApps) and end-users. It sets the stage for a new era of platform adoption and engagement, leveraging the unique benefits of blockchain technology.

1.3 Motivation and drive

The initial decade of the web (1990-2001) witnessed enormous speculative investment in a plethora of ideas about how the web would revolutionize our lives. However, the lack of robust business models led to the dot com boom and subsequent bust (2000-2001). Following this, advertising evolved on the web, paving the way for major companies to profit from internet usage. This era saw users unknowingly agree to End-User License Agreements (EULAs) that effectively transferred their data to corporate databases, marking a significant and rapid wealth transfer in human history, akin to the exploitation of natural resources like oil. This period can be compared to the early years of blockchain technology. Over the past decade, thousands of blockchain projects have emerged, many lacking sustainable business models and relying on speculative token prices. This parallels the dotcom era, suggesting that the blockchain sector is at a juncture where it needs sound business models post the 'crypto winter' of 2018. Blockchain's appeal lies in its potential for disintermediation, promising to revolutionize sectors like banking, supply chains, voting systems, online identity, and the sharing economy by eliminating central authorities. However, the key question remains: how can we decentralize surveillance capitalism, the predominant business model of the digital era? The rise of web companies post-2001 was fuelled by centralized advertising. Similarly, the recovery of blockchain companies from the 2018 downturn may hinge on the concept of Decentralized Marketing, a promising direction for the future of blockchain technology. All this requires the need to connect and interoperate. This will also require smart contract execution to be done concurrently because there will be latency and compute involved (Gorki is particularly designed for this).

2 Orchestration

At its core, the distribution of sponsored content is no different to any regular smart contract interaction on the blockchain. Curators, creators and developers want to be able to share content, this is a primitive operation on the blockchain: [sending some content to a location on-chain]

$$location!(\,content\,)$$

When a developer, or dApp wants to retrieve this data then it is a simple matter of waiting for that content to arrive and be presented to the user. [when content arrives at location, it is served to the User through an App]

$$for\,(content \leftarrow location)\{\; presentToUser(content)\;\}$$

This ability of curators and content creators to apply their material on-chain and have developers access that material allows groups to create their own flourishing economies and marketplaces. Communities of content creators can arrange to tokenise their involvement and democratise the availability of their content and sharing of the data that they have access to, creating a platform that global communities can use to share data through. For example, membership might be required:

$$for\,(content \leftarrow getLocation(\, membershipToken\,))\;\{\; presentToUser(\, content\,)\;\}$$

For example, or similarly some level of ranking/qualification:

$$for\,(content {\leftarrow} getLocation(\, communityEsteem\,))\;\{\; presentToUser(\, content\,)\;\}$$

2.1 Evergreen content demand

An important capability is demonstrated here, in Gorki code will wait for messages to be received until it is executed. This fits succinctly into modern content delivery systems. In many systems the development and creation of content is second by second, marketing emails and newsletters are compiled at the time they are opened and not when they are sent to the individual. This enables up-to-the-minute content to be delivered and is essential to keep up with the pace of the modern world.

2.2 Architecture

The objectives here would be to integrate this into the core Gorki APIs – creating two specialised entry points:

1. **Sponsored content API:** This API is used to create and curate content on-chain, providing the correct metadata to the content and allowing the consumption model that it is designed to be specified. When a developer is creating an app, they can use this API to provide a source of marketing material and a source of revenue for fees that are required to interact with the app. Meaning that by default a user would see a front end to their experience that mimics what the average person sees today on twitter, tik tok or Instagram.

2. **Self-disclosure API:** This API also allows the users of the app to divulge information about themselves with regards to content they would like to see, information that they are willing to share, and perhaps if they are willing to pay for services and app usage that would mean that they see less content, or none.

2.3 API structures

A simple architecture (Fig. 1), possible through the Gorki content API will be to allow content curators and creators to load content onto the Gorki network, locking value into the system to be consumed by the apps and developers for distributing this content to the end user. App developers will then be able to retrieve this content and use it when needed across their apps. These entities will be able to create names and ecosystems around this data, monetising, or tokenising access and control of their data in any way they would like. Allowing for business models that can provide merit/kudos to creators and consumers alike, and potentially allowing for transfer and trading of this property to unlock/access specific content.

In line with the needs of the user, it will also be possible for the user to govern their own data and exposure to marketing content (Fig. 2). Curators/content providers can provide templates and preferences and users can decide how they interact with different ecosystems. All data can be encrypted and stored on-chain giving the user absolute power over their experience. Essentially choosing to take part in the “freemium” business model or moving to a more “pay to play” way of interacting with the relevant apps.

3 Content distribution stories

Among the key user scenarios is the uploading of sponsored content. Figure 3 shows the process of sponsors pushing content to chain. They do so any modified interface that has been developed according to their requirements described above. After sponsored content is uploaded to chain, they may configure a campaign. During this configuration the sponsored content will be decorated with metadata that allows developers to locate that content and also determine how it is appropriate to use it. Implicit in this simplified relationship above is the assumption that the marketing content is being delivered and funded by the curator/creator – therefore a smart-contract will control allocation of funds. Then, when a dApp user takes an action that delivers content to their dApp UI the content request must provide a location for the token requirements which will be used to place the sponsored content in various places in the dApp UI.

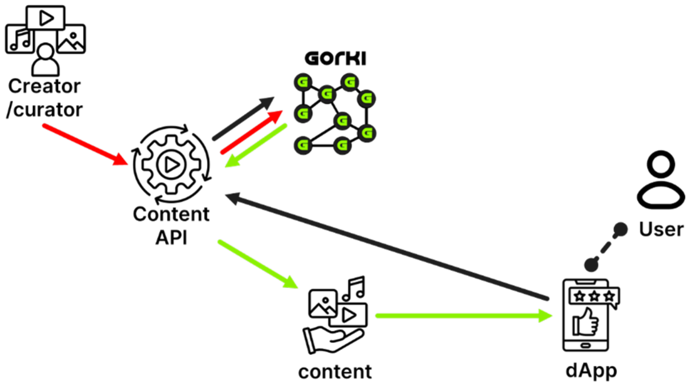

Figure 1: Content delivery through API for app developers

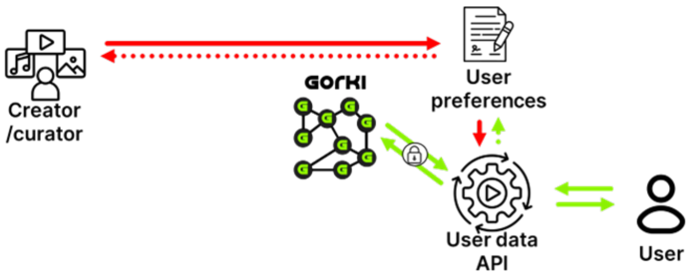

Figure 2: User data API allowing control over content delivery and personal data from the end user perspective

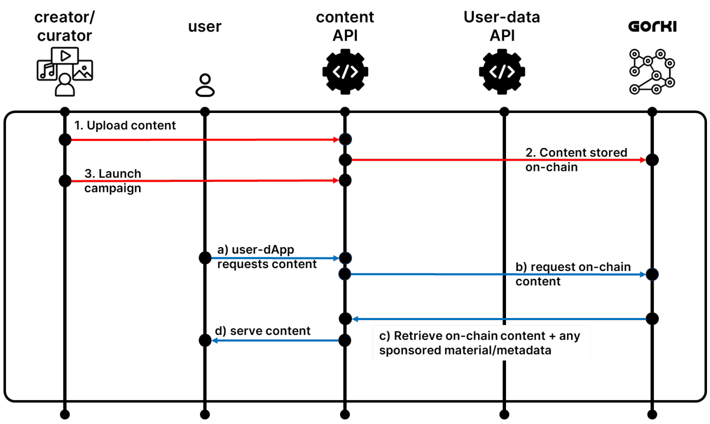

Figure 3: Uploading and distributing sponsored content

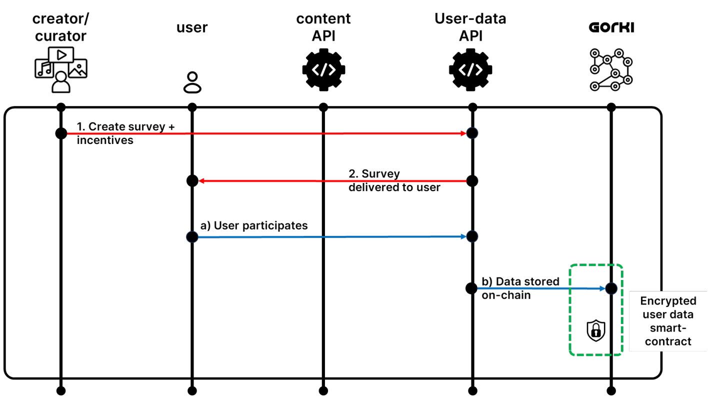

Figure 4: Empowering user control over personal data

Another key user scenario is self-disclosure of user data (Fig. 4). Today's users have become quite accustomed to providing data about themselves as they go. In this paradigm it will be possible to self-identify largely by selecting the sponsors and type of sponsored content they are willing to engage. Naturally, sponsors will want more targeted information than this. This structure will provide basic APIs to solicit additional information about users.

A central part of this new capability is the democratisation of user data and the "freemium model" through decentralised technology, users, rather than wanting to circumvent reselling information about them, actually want to be included as full partners in any such data exchange. If users are informed, and consent, and are financial participants in any such exchanges this is mutually beneficial for all. It allows users to know which downstream agencies are interested in their data and to participate in the economic responsibility of allowing their data to be used (Fig. 5).

4 Gorki: a short introduction

Critical aspects of Gorki for ad-tech and content delivery:

- Interoperability, connecting and being the communication/orchestration layer.
- Concurrent block creation. Our goal is to maximise the throughput of transactions by employing concurrency in the smart contracting language and consensus.
- Advantageous security model (OCAP), that allows fine grained control over on-chain resource.
- High level composable programming language with explicit concurrency.
- Easy data storage on chain without requirement for external solution, safe concurrent access/modification of the data
- Query and search on-chain - the ability to compose queries through the smart-contract language creates an ability to actually find content and the location of information on-chain, unlike other blockchain technologies.

The Gorki protocol is being designed to minimise the amount of synchronisation required to avoid the infamous blockchain trilemma (liveness, safety, and fault-tolerance) to ruin user experience. Four major innovations and principles have been implemented in a blockchain model. The mathematical model backing the state of a computer is derived from process calculus, which uses process "names" as the fundamental element of that computer's state, and computation is described through the process of exchanging data between "names". To give concurrent access to this state, names are stored inside a tuple space[1], inherited from coordination language Linda. For practical reasons, this can be seen as a map that stores the content of the channels (the data passed between the names), which allows Gorki to easily compute the proof of state by maintaining this map in the form of a Merkle tree. Another benefit of this model is that it allows easy access to sharding through namespaces. Since each name can belong to one or more namespace, transferring a value from one shard into another is a matter of transferring records through the states. Rholang, the programming language of the platform, is an exact description of the state of the computer. In essence, it is the WYSIWYG principle. Rholang supports an object-capability security model that unlocks an enormous amount of value when deployed to a shared execution environment like a blockchain. Rholang is the API to a powerful concurrent state machine which prevents the user from making mistakes, restricts access and manages resources. Since process calculus has a history of usage proving concurrent programs (CCS)[2], this opens a path towards a proof system for on-chain smart-contracts. This allows formal verification of processes that are running concurrently, creating a method that allows the user to ensure their programs, smart-contracts and applications are running as intended. It is well known that systems that rely on a leader for consensus are prone to attacks, but also, there are no systems to date presented that are entirely asynchronous. One breakthrough is the Nakamoto consensus used in PoW systems, which employs economic incentives to make finality probabilistic and make attacking unprofitable. Gorki is developing a consensus algorithm inspired by the Casper CBC research branch to enable a genuinely leaderless PoS consensus. Gorki employs programming principles inherited from Rholang, building composable modules under concurrent settings. Essentially, all software manipulates data in memory, but finding the right compromise between simplicity and efficiency is crucial. Concurrency and composability are the main requirements that software should match in the upcoming decade to fully utilise hardware resources and reach the levels of performance required to facilitate the next generation of decentralised applications. Each of these components and the innovations they unlock are designed to meet specific market needs.

5 Summary

This paper introduces Gorki as a highly interoperable blockchain platform designed to revolutionize content distribution within the freemium economy model. Gorki effectively bridges the gap between traditional Web 2.0 and emerging Web 3.0 technologies, fostering a seamless integration of user-centric approaches with decentralized, privacy-focused systems. Key to Gorki's innovation are two versatile template APIs developed for use by developers, content creators, and curators. These APIs facilitate the

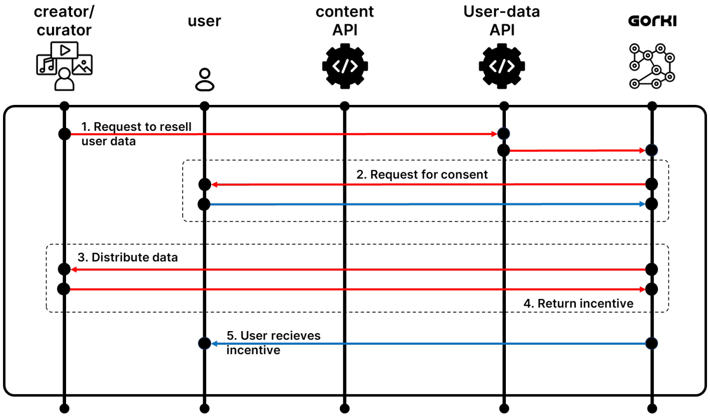

Figure 5: Incentivised redistribution of data

distribution of content in a manner that respects user preferences and privacy, allowing users to determine their level of engagement with the content. This approach enables users to participate more actively and selectively in the freemium model, choosing how they interact with sponsored content and advertisements. Gorki's blockchain architecture is uniquely equipped to support this model. It employs a fine-grained security model, ensuring robust data protection and compliance with privacy regulations such as the EU's right to be forgotten. Its scalability is achieved through a concurrent consensus algorithm, allowing the blockchain to handle a high volume of transactions and interactions without compromising performance. Moreover, Gorki's blockchain is capable of loading and managing large volumes of data on-chain, an essential feature for content-heavy applications. Its advanced smart contract language offers both security and flexibility, incorporating a powerful query language for efficiently locating resources on-chain. This feature is particularly valuable for content curators and creators, who require dynamic and secure access to blockchain-stored data. In conclusion, the Gorki blockchain presents a groundbreaking solution for the freemium economy. By combining high interoperability, robust security, scalability, and user-centric content distribution models, it paves the way for a new era of digital interaction and economic models, aligning the benefits of decentralized technology with the demands of modern web users.

References

[1] Enrico Denti and Andrea Omicini. An architecture for tuple-based coordination of multi-agent systems. Software, practice & experience, 29(12):1103–1121, 1999. Place: Chichester, UK Publisher: John Wiley & Sons, Ltd.

[2] Howard Bowman. Concurrency Theory Calculi an Automata for Modelling Untimed and Timed Concurrent Systems / by Howard Bowman, Rodolfo Gomez. Springer London, London, 1st ed. 2006. edition, 2006.