MEDIA PROTOCOL

Earn Tokens while Reading, Watching and Sharing Content

Protocol Architecture

August 2018 - Version 1.0

TABLE OF CONTENTS

TABLE OF CONTENTS **2**

INTRODUCTION **3**

1. MEDIA PROTOCOL **5**

1.1 Consumer-Focused 5
1.2 Why the Blockchain? 6

2. MEDIA PROTOCOL ARCHITECTURE 7

2.1 Design Goals 7
2.2 Major Building Blocks 7
2.3 Tokenisation of URLs 9
2.4 URL Promotion 9
2.5 URL Consumption 10
2.6 Paywalled Content 10
2.7 Identity 11
2.8 Affiliates 11
2.9 Delegation 12
2.10 The Smart Contract 12
2.11 Scalability/Off-chain Service 13
2.12 Security 14

REFERENCES **15**

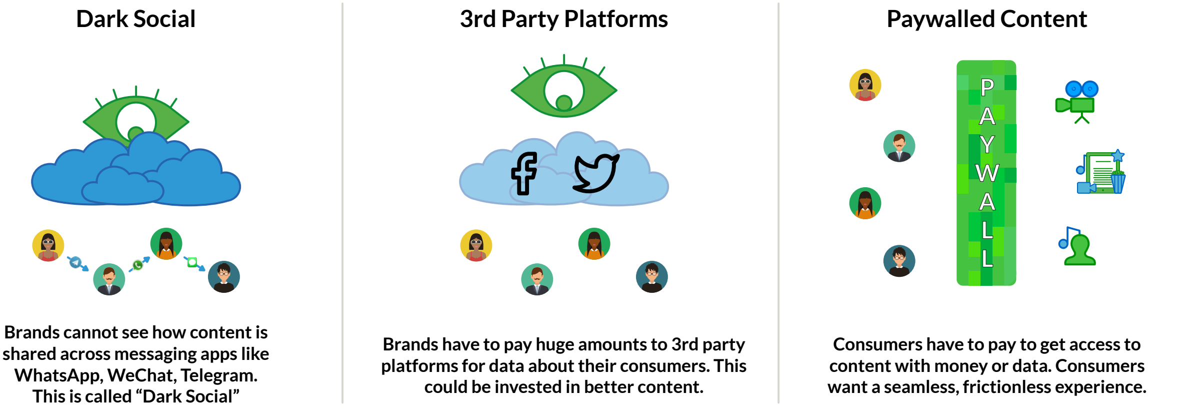
INTRODUCTION

Content has never been more important than it is today. Whilst online advertising is a pillar of the modern Internet, brands increasingly rely on exposure to audiences through content marketing to stay relevant.

However, online audiences and the data around the content they consume are controlled by a small number of centralised platforms, such as social platforms and search engines. that limit the flow of data from consumers to the parties who make content. These platforms harvest huge amounts of valuable engagement and preference data from consumers, but offer brands and publishers little in return.

These platforms present themselves as networks, when they are simply nodes in the network. The social web exists on messaging platforms (IMs) such as WhatsApp, WeChat, Telegram and SMS. When a consumer finds a piece of content, they share it via these channels. It is this vital data that centralised platforms do not, and are not incentivised to provide - these channels are referred to as “Dark Social”.

The experience of consumers is worse as a result, because instead of having a more direct relationship with their audience, brands and publishers who make content rely on clickbait or expensive paid media campaigns to reach them. To try and create a more direct relationship, consumers are forced to pay for content in many different ways e.g. paywalls, email addresses, micro-credit-card-payments, when all they want is a seamless experience.



Dark Social

Brands cannot see how content is shared across messaging apps like WhatsApp, WeChat, Telegram. This is called “Dark Social”

3rd Party Platforms

Brands have to pay huge amounts to 3rd party platforms for data about their consumers. This could be invested in better content.

Paywalled Content

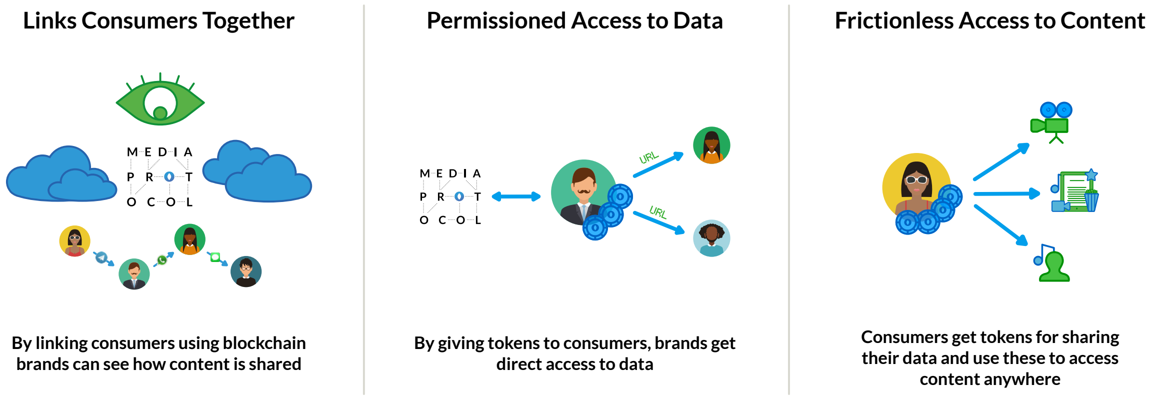
Consumers have to pay to get access to content with money or data. Consumers want a seamless, frictionless experience.

Fig.1: The challenges addressed by MEDIA Protocol

A new developer protocol – MEDIA Protocol – will provide developers, brands, publisher, consumers, etc. the opportunity and incentive structures to interact in ways that are not permitted by centralised platforms. In this way, the protocol does more than reward attention: it fundamentally enables a new set of relationships between important parties that already want to interact with one another in more efficient and rewarding ways.

MEDIA Protocol creates a direct economy for the exchange of content, data, and incentives, including financial incentives. MEDIA Protocol enables publishers, brands and content creators to deliver the most relevant content in pursuit of a more transparent, efficient and enjoyable online experience.

The economic function of the protocol is designed to create a direct channel for publishers to promote content through balanced consumption incentives directly to consumers, for consumers to pay directly for gated content, and to spend tokens rewarding favoured content creators and publishers.



Links Consumers Together

By linking consumers using blockchain brands can see how content is shared

Permissioned Access to Data

By giving tokens to consumers, brands get direct access to data

Frictionless Access to Content

Consumers get tokens for sharing their data and use these to access content anywhere

Fig.2: The benefits of adopting MEDIA Protocol

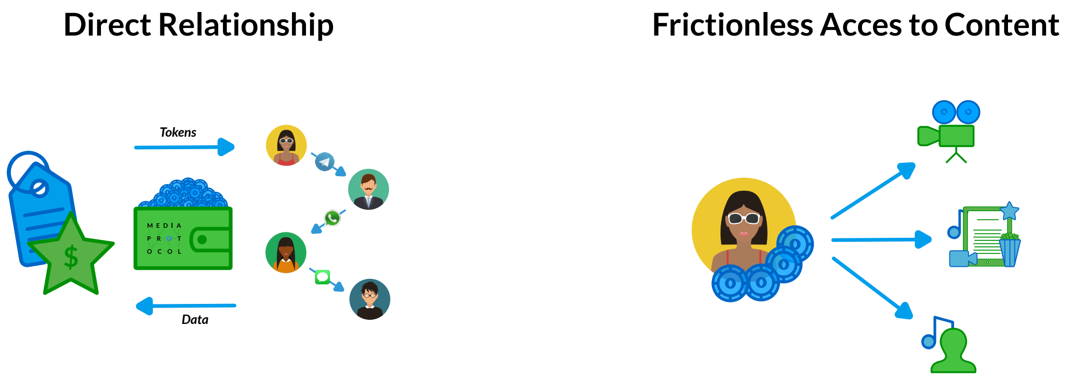
The protocol is designed to bring visibility to how consumers interact with and share content. In the future, this will foster an ecosystem of localised and community-centric content distribution that cannot currently overcome the economies of scale perpetuated by incumbent platforms.

In this paper, we will discuss the technical characteristics of MEDIA Protocol, how it will be executed and our plans for the future. To learn more about MEDIA Protocol as a business proposition, please read our Business Paper.

1. MEDIA PROTOCOL

MEDIA Protocol is an open architecture that creates a multi-directional economy, where publishers, creators and content consumers exchange content, data, and incentives, including financial incentives, directly. It enables brands and publishers to directly incentivise the consumption of content using MEDIA Protocol Tokens (MPT), the protocol's accompanying ERC20 token, and allows consumers to be compensated for their content interactions and the preference data they generate.

All actors involved in the creation and consumption of content recognise the important role that audiences play in the reception, perception and distribution of any given piece of content. Existing economic models, however, cannot compensate audiences for the role they play.


Direct Relationship

Frictionless Acces to Content

Fig.3: The key benefit for brands and publishers, and consumers

Through MEDIA Protocol, a direct two-way flow of tokens from brands and publishers to consumer, and vice versa, remedies this and facilitates payment for paywalled content, tipping and the ability for content promoters to reward creators. MEDIA Protocol is the foundation for an ecosystem where publishers, creators and any variety of content distribution platform can interact directly with consumers to further a more transparent, efficient and enjoyable online content experience for everyone.

1.1 Consumer-Focused

MEDIA Protocol is designed to help the marketing, content and advertising industries transition towards more consumer-centric models. To this end, the MEDIA Protocol team is committed to creating a multi-disciplinary industry working group, to facilitate the adoption of all media, advertising and content blockchain protocols by publishers, brands, social networks and agencies.

Using MEDIA Protocol, brands and publishers will be able to incentivise consumer interaction with content by using MEDIA Protocol to assign incentives (in the form of tokens) to specific pieces of content, represented by URLs, and distribute the tokens as specific rewards for defined content interactions via apps and websites that support MEDIA Protocol.

1.2 Why the Blockchain?

The blockchain provides several key advances that make MEDIA Protocol possible. The decentralised nature of the blockchain removes the need for a middleman, who extract more value than they contribute to the system. As the network is completely transparent, the blockchain prevents any actor from being able to manipulate the system, and means any fraud has to be in the open, where it can be combated. The blockchain additionally enables fast micropayments; if each interaction were to earn fiat money, for example, the transaction processing fees would overwhelm the payment values. With the blockchain, transaction processing costs will go to almost zero as the technology evolves.

Whilst we could derive some of these benefits by piggybacking on an existing cryptocurrency token, creating our own token has two key advantages: first, we can allow actors to define their own economic rules that are optimised around creating the highest value for content publishers, promoters, distributors and consumers, and second, we can implement innovative features oriented around providing a better, seamless user experience, such as the ability to delegate from an offline wallet to an in-app wallet.

2. MEDIA PROTOCOL ARCHITECTURE

2.1 Design Goals

In designing the MEDIA Protocol, we had the following key criteria:

- Enable dApp creators to provide a slick, frictionless user experience to consumers;

  Maximise existing crypto infrastructure (such as ERC20) and the benefits it provides, such as tried-and-tested algorithms and existing wallet software;
- Ensure we leverage existing web standards where useful and possible;
- Provide mechanisms to foster an ecosystem around the protocol (e.g., affiliate fees for dApp creators);
- Make adoption as easy as possible for publishers; and
- Protect against fraud and bad actors.

2.2 Major Building Blocks

The heart of the protocol is an ERC20[1] compatible Ethereum smart contract that provides token accounting, the handing of rewards for viewing and sharing promoted content and charging for viewing paywalled content.

ERC20 is a common interface for Ethereum smart contracts that provides a standard set of functions and rules primarily around token transfer, which mean any wallet that supports ERC20 can natively support any ERC20-compliant token.

Compared to other platforms/standards (e.g., Counterparty), Ethereum has high adoption, several major wallet implementations, a well-regarded user experience, exchange support and generally a very strong token management ecosystem. It does have several limitations/challenges, however, which are discussed below in Section 6.11, discussing scalability.

Other than standard token transfers between addresses, the main interactions with the smart contract will be actors who set up promotions to reward consumers for impressions/likes/shares, and consumers who register impressions/likes/shares with the smart contract and, in turn, receive rewards.

As discussed in more detail below, we will be building an off-chain service to handle the initial volumes, and a verification service to verify consumers are genuine and avoid Sybil attacks.

Figure 4 illustrates a typical sequence of interactions, which are detailed later in this section.

Figure 5 shows the high-level components of the system and how they connect.

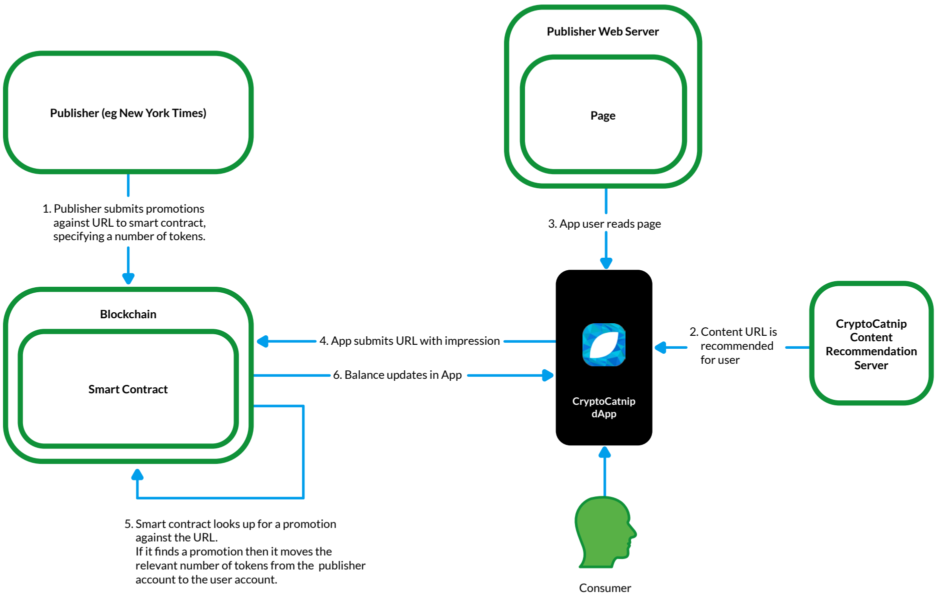

Publisher (eg New York Times)
1. Publisher submits promotions against URL to smart contract, specifying a number of tokens.
Blockchain
Smart Contract
5. Smart contract looks up for a promotion against the URL. If it finds a promotion then it moves the relevant number of tokens from the publisher account to the user account.
4. App submits URL with impression
6. Balance updates in App
Publisher Web Server
Page
3. App user reads page
CryptoCatnip dApp
2. Content URL is recommended for user
CryptoCatnip Content Recommendation Server
Consumer

Fig. 4: Typical sequence of interactions

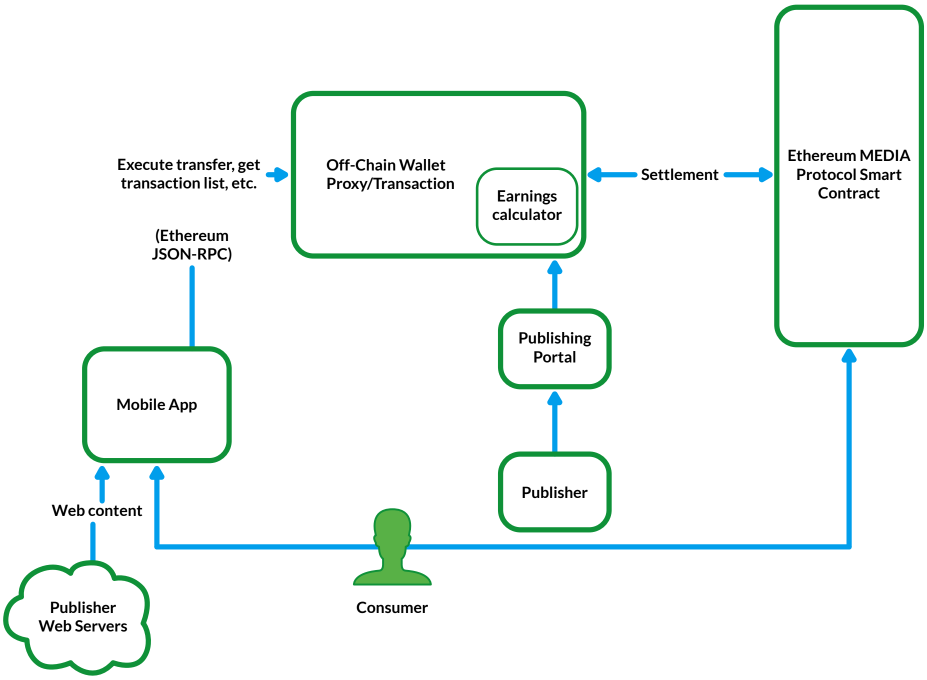


Fig. 5: High level components

2.3 Tokenisation of URLs

As discussed above, any URL (and in a later iteration of the protocol, any IPFS address) can be associated with an Ethereum address as a basis for registering a promotion, receiving tips or paywall payments.

URLs can represent any kind of media – articles, images, video, podcasts, music – and whilst the initial activity types in the protocol will focus on those that are universal across media types, we later plan to add more detailed tracking (for example, how many seconds a video or podcast was consumed for) to facilitate more precise observations and optimisations.

Whilst anyone can register a promotion against any URL, to receive tips or paywall payments the publisher must verify that they own the relevant URL by including the destination wallet address for these payments within a meta tag in the page.

We will also provide an additional mechanism for popular UGC sites that do not allow authors to customise meta tags – e.g., YouTube, Instagram, Facebook, Medium – wherein we will allow the author to embed the address somewhere only they have access to (such as a YouTube video description).

2.4 URL Promotion

Publishers and brands will promote URLs to incentivise viewership and creation. To do this, the publisher requires MEDIA Protocol tokens. For promotions aimed at consumers, using a smart contract request originating from the publisher wallet, they will submit a combination of:

- The URL to be promoted;
- Promotion parameters (e.g., the number of tokens available for promotion that day and interaction types included in the promotion);
- List of acceptable identity verification services for consumers; and
- List of acceptable identity verification services for affiliates and maximum affiliate commission level.

For promotions aimed at publishers or content creators, these parties will submit a combination of:

- The URL to be promoted;
- Promotion parameters (e.g., number of tokens available for promotion that day and interaction types included in this promotion); and
- The address that will receive the tokens (typically the address of the creator or publisher).

When the promotion is submitted, it creates an entry in the smart contract which is valid for the promotion period, after which the promotion must be renewed. This reserves the number of tokens required for the promotion from the publisher's balance until the promotion expires. If the publisher balance does not contain enough tokens, the request to create the promotion will fail.

At the end of the period, the contract will execute and tokens will be distributed as per the economic model described within Section 3.2 (note that adding timed executions to Ethereum is currently an open issue[2], but there are ways around this such as using Oraclize[3] or Ethereum Alarm Clock[4]).

All Ethereum smart contract operations require paying a small amount of Ethereum tokens to execute the operation, known as 'gas'. Whilst initially consumers will need to pay their own gas for smart contract operations when consuming content, once Ethereum Serenity is released we will leverage its features to allow gas to come from the publisher.

2.5 URL Consumption

For every impression, vote or share on social media for a URL that the dApp is aware of (e.g., via a 'share' feature built into the dApp) or survey filled, the dApp should submit a notification to the smart contract with the URL signed with the key of the consumer that interacted with the content. As controlled by the algorithm, the smart contract will then assert whether the consumer is verified with one of the verification services specified by the publisher, and will credit the consumer with tokens from the publisher.

2.6 Paywalled Content

We will have support to allow consumers to view one-off pieces of paywalled content as well as subscribe to particular publications.

Existing web paywalls generally operate via a sign-in page that provides cookies to authenticate content requests. We are making our system as easy as possible to integrate for publishers that have this type of paywall.

One-off paywalled content should contain a meta tag entitled 'MEDIA TOKEN COST' containing the number of tokens required to view the content. The process by which a consumer uses MEDIA tokens to access content through a paywall is as follows:

- The consumer/dApps sends a request to the smart contract specifying how much they're willing to pay for the content (i.e., the same as the cost from the meta tag).
- The smart contract transfers that amount from the consumer to the publisher and provides the consumer with a publisher URL.
- The consumer submits a request to the publisher sign-in URL containing the transaction ID from the smart contract transaction, together with the last block hash (for security purposes) signed with their private key (using the standard Ethereum signing algorithm[8]).
- The publisher can verify against the blockchain that the consumer with the Ethereum address corresponding to the key has paid for the content in the specified transaction, and then return a cookie providing access.

Whilst in theory a user could help others bypass the paywall by sharing either the cookie request details or the cookie itself, this is already possible with existing web-based paywall mechanisms (e.g., users can share their email and password with friends). We will rely on publishers using their existing mechanisms to mitigate this.

MEDIA Protocol also supports subscriptions, as content owner can provide access to their paywalled content through subscriptions. Subscriptions can have different durations (e.g., weekly or monthly). The content owner has full control over the subscription prices on a per-customer level, which, in combination with user behaviour data collected over the network, provides full power to create relevant, customised offerings. For example, a publisher may decide to reward a consumer who has previously interacted with a URL with a discounted subscription price. This amplifies the reward received by consumers.

2.7 Identity

It is important that the system is built in a way that publishers and promoters can avoid being gamed by malicious users, bots or Sybil attacks. The two typical strategies to mitigate these risks are reputation-based systems and external verification of accounts. As discussed earlier, in our initial implementation we will focus on external verification to allow consumers to earn rewards without having to spend time building a reputation.

Each consumer can publish a signed copy of their address to the blockchain, with the signature provided by a verification service confirming they've verified the consumer to be genuine rather than a bot. In addition, the signed data may contain an ID from the service, preventing the consumer from creating multiple accounts. Example verification services we can integrate in the future are blockchain native services such as Keybase, Civic and uPort, and non-blockchain native services such as Facebook, Twitter and reCAPTCHA, which would connect to the blockchain through a proxy service.

Any given promotion can specify a list of public keys of permitted verification services, and only users with a valid signature from one of those services will be eligible for rewards for viewing the content.

2.8 Affiliates

Affiliates enable entities that drive interaction with the token, such as dApps, to take a share of revenue – incentivizing the creation of an ecosystem around MEDIA Protocol.

To do this openly and transparently, affiliate addresses and commission levels are specified explicitly in transaction calls from consumers. Publishers will decide whether affiliates must be verified or not, and whether to introduce a maximum affiliate commission level. If a publisher specifies that they only accept verified affiliates, then the publisher must specify a list of affiliate verification services that they support, and affiliates must have a signature from one of the services stored against them in the blockchain (similar to the mechanism that verifies consumer accounts).

For example, assuming:

- Publisher X specifies that they support commissions up to 20% and affiliates verified by verification service Y.
- Affiliate A has been verified by verification service Y, and there is a copy of affiliate A's address signed by verification service Y's private key stored against them in the smart contract.
- If the consumer spends 1 token to access paywalled content from Publisher X, they can specify in the call that 15% should go to affiliate A. Given that affiliate A has been verified by one of the services that publisher X specifies, and 15% is below the maximum threshold, publisher X will receive 0.85 tokens and affiliate A will receive 0.15 tokens as commission.

This process works for both paywall payments from the consumer to the publisher and for reward payments from publishers to consumers (and, in both cases, it will be the consumer providing the affiliate address and commission level). More affiliate examples are outlined in the MEDIA Protocol Business Paper.

2.9 Delegation

To create a seamless user experience that does not require consumers to sign every transaction using their primary wallet, it is necessary for dApp creators to be able to create wallet addresses for dApp consumers that are isolated from their primary address, but which are able to earn and spend MEDIA Protocol tokens. To enable this, the smart contract will have a delegation facility where the owner of an Ethereum address can specify other addresses to which content consumption and paywall operations can be delegated. The smart contract will authenticate that address using a private key dedicated to the dApp, without the consumer having to compromise security of their primary private key. Note that whilst delegated addresses will be able to get rewarded for interacting with content and able to access paywalled content, they will not be provided with access to the ERC20 token transfer functions of the contract[4].

The smart contract will also provide functions to allow a user's primary account to add and remove authorised addresses.

2.10 The Smart Contract

In total, the smart contract consists of:

- Map of publisher addresses to verification service addresses they permit;
- Map of consumer addresses to verification service signatures;
- Map of URL to list of promotions (each promotion consists of an interaction type, a token amount for the promotion, a duration of the promotion and, optionally, a destination address);
- Map of address to token balance;
- Two-way map of primary addresses to delegate addresses (for delegation to dApps); and
- List of user interactions.

The conceptual model of the smart contract can be seen in Figure 6.

The contract will contain the following consumer functions in addition to the standard ERC20 functions:

- Record content event (impression, upvote, downvote, share, etc.);
- Add/remove/list delegate addresses;
- Add an account verification signature;
- Create subscription;
- List active subscriptions; and
- Cancel subscription.

And the following publisher functions:

- Add promotion against URL;
- Create a subscription ID;
- Cancel a subscription ID; and
- Charge consumer for a subscription.

And the following affiliate functions:

- Add an account verification signature.

2.11 Scalability/Off-chain Service

Currently, Ethereum has a cost per transaction of approximately USD0.03-0.04, and can only support around 5-10 token transactions per second.

At peak times, it can also have relatively long confirmation times for any given transaction. Still, Ethereum has a strong roadmap and, overtime, we expect the cost per transaction and confirmation times to dramatically decrease, and overall capacity to increase – for example, in the upcoming Ethereum Casper release Proof-of-Stake will be replacing Proof-of-Work[6], which reduces the cost of running mining nodes.

In addition, an upcoming implementation, sharding[7], means rather than every node needing to verify every transaction, only a small subset will need to reach consensus to verify, thereby dramatically decreasing transaction time and increasing scalability.

Given these current, but temporary, constraints around Ethereum cost and scalability, we propose a hybrid on-chain and off-chain solution, similar to that being operated by Kin[8] and by UnikoinGold. The off-chain service will expose a subset of the Ethereum RPC JSON API[9], with the hosting costs absorbed by MEDIA Protocol, and will regularly (at least daily) settle with the Ethereum blockchain.

The advantages of the off-chain service are that there will be no end-user transaction costs (which means that users during this initial adoption period will not need to add gas to their accounts). It will be scalable to expected volumes from the start, with low quantity, high-performance transaction executions.

This comes with trade-offs, however, as it does not provide the decentralisation and transparency of the Ethereum blockchain, and consumers and publishers will have to place trust in the centralised, off-chain service. Therefore, as Ethereum scales up overtime and is able to handle volume at a sufficiently low cost, we will migrate all transactions to occur directly on the Ethereum blockchain. We are designing the system from the start to make this eventual migration easy.

One important design factor for hybrid solutions is avoiding double-spend attacks. As discussed in Section 4, Token Economy, these are impossible for content promoters as their tokens will be held for the duration of the promotion. We will also maintain a minimum balance on-chain and only allow pending in-app transfers out of on off-chain account to the sum of the on-chain minimum balance.

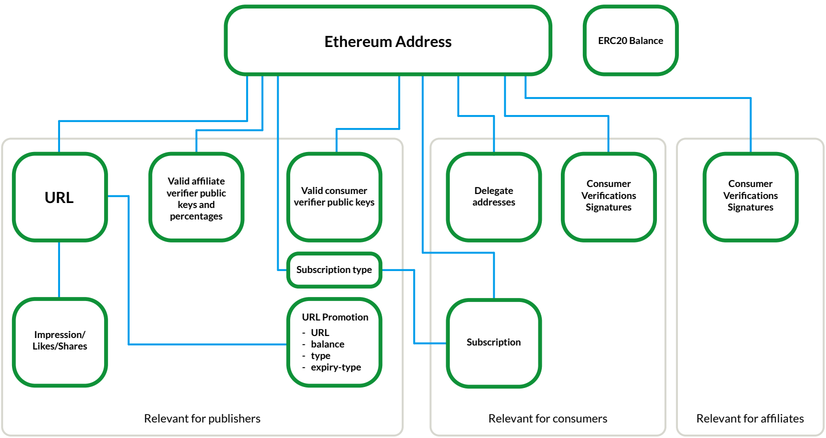


Fig 6: Conceptual structure of smart contract data

In the extremely unlikely situation that Ethereum protocol does not become able to handle scale at a reasonable cost, we have the option of migrating the token to a different blockchain, which could be our own, or another third-party alternative to Ethereum. Storj has proven such a migration is possible with an actively traded token[10]. We will examine if there is anything we can do to make allowance for this scenario in our ERC20 smart contract.

2.12 Security

Token security is very important, and we will ensure it through a variety of methods.

All future smart contract code will be open source where it can be scrutinised for security issues, and we all code published to-date has been thoroughly audited by an expert third party.

We will go through a testing period on the Ethereum test network before going into production, look at implementing circuit breakers within the contracts to allow it to be paused in case of any issues, and keep the code as clear, clean and simple as possible. Our team has rich experience writing secure code, and will stay up-to-date with Ethereum security vulnerabilities and respond promptly as required.

MEDIA Protocol is able to call upon some of the foremost security experts in the blockchain space. Our lead advisors, TLDR, have a market-leading security offering. In addition Richard Ma, the CEO of QuantStamp, is a member of our Advisory Board.

REFERENCES

[1] ”ERC20 Token Standard The Ethereum Wiki.” https://theethereum.wiki/w/index.php/ERC20 Token Standard

[2] ”ALARM opcode Issue #117 ethereum/go-ethereum GitHub.” https://github.com/ethereum/go-ethereum/issues/117

[3] ”docs/ ethereum.md at master oraclize/docs GitHub.” https://github.com/oraclize/docs/blob/master/source/includes/ ethereum.md

[4] ”Ethereum Alarm Clock.” http://www. ethereum-alarm-clock.com/

[5] Ethereum: A secure decentralised generalised transaction ledger, Dr. G Wood. http://gavwood.com/Paper.pdf

[6] ”Proof of Stake FAQ - ethererum/wiki Wiki” https:// github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ

[7] ”Sharding FAQ ethereum/wiki Wiki” https://github. com/ethereum/wiki/wiki/Sharding-FAQ

[8] ”Kin: a decentralized ecosystem of digital services for daily life - Kik.” https://kin.kik.com/papers/Kin

[9] ”JSON RPC ethereum/wiki Wiki” https://github.com/ ethereum/wiki/wiki/JSON-RPC

[10] ”STORJ.IO — Token Sale Wrap-up Details.” http://blog. storj.io/post/165553434093/token-sale-wrap-up-details

MEDIA PROTOCOL

For more information, please contact:

James Tabor, CEO
James@mediaprotocol.org

www.mediaprotocol.org

Join us on Telegram: ***Media_Protocol_Community***