Mercury Protocol

Radical App International

08/23/2017
Revised: 10/1/2017

Abstract

Modern communication is outdated. Centralized communication platforms built on private servers are only as secure as their weakest defense, user privacy is habitually violated as service providers sell behavioral data to advertisers [10], and content is restricted to a single platform. We aim to fix these issues in the communication industry with the Mercury Protocol, an open source project for communication platforms to reap the benefits of decentralized blockchain technology at minimal cost. Any existing or future communication platform that integrates the Mercury Protocol will theoretically be able to exchange content across previously isolated privatized applications, increase user privacy by creating monetization strategies that do not depend upon their behavioral data, leverage tokens to encourage user participation, and provide stronger network security than a private system that has a single point of failure. The first application to implement this protocol is Dust, soon to be followed by Broadcast. We believe that the Mercury Protocol can solve the pervasive issues of the current communication industry, and will be the future standard of communication platforms.

Introduction

We built the Dust application with the goal of giving ownership of the conversation back to the user. Unlike other platforms that store data on hard disk, Dust stores all messages in-memory (RAM). This makes it several orders of magnitude more difficult for a malicious party to gain access to any message data, and renders the data forensically unrecoverable from the platform's servers once deleted. As a result, Dust provides users with a private platform that is designed to be safe from prying eyes.

Similarly, blockchain technology also gives back more power to the end user in the form of transparent information, privacy, security, and utility. All data on the blockchain is stored in a decentralized manner that is public, transparent and auditable. Yet because a user's real identity is not associated with the blockchain address, users benefit from privacy through pseudonymity while still generating behavioral data for large scale analysis. Moreover, the decentralized consensus mechanism inherent to blockchains eliminates the possibility of a single point of failure within the network and helps to guarantee the validity of the events that occur on it.

We are taking advantage of these highly advantageous characteristics of the blockchain technology in order to improve the communication industry by building the Mercury Protocol. The protocol implements various communication features and will utilize a Global Messaging Token, an ERC20 based token built on the Ethereum blockchain, to drive these features. This token will encourage user participation in platforms that integrate the Mercury Protocol, by allowing users with GMT to transact with Mercury Protocol services on the blockchain. Additionally, the Mercury Protocol will enable a platform agnostic communication network, where a user on one messaging platform could more securely send a message through the blockchain to another messaging platform if those platforms integrated the Mercury Protocol.

The Mercury Protocol will be first integrated into Dust as a proof of concept and shipped to market by end of Q3 2017. Shortly after, we will integrate the protocol into Broadcast, a new platform designed to tackle the issues endemic to social media, including harassment ("trolling") and echo chambers, among others. Broadcast will be released by end of Q4 2017. Once both apps are in market with Mercury Protocol integration, we will conclude any remaining security audits and open source the project so that others can join us in the decentralization of the communication industry.

Current Messaging Ecosystem

The current communication platform model in vogue has not significantly evolved since the advent of the radio. Service providers offer "free" services that aggregate user behavioral data and attention, which they in turn sell to advertisers for profit. This system creates a strong incumbent bias, as newer platforms with smaller user bases struggle to survive. In 2016, Facebook and Google collected 77% of American spending on digital marketing, and nearly half of all global spending [8]. This system not only consistently impacts user privacy, but also bears all the technical problems associated with centralized platforms, including "single point of failure" security and trust-based services (e.g. there is no method for unbiased verification that a private server actually sent your message, deleted a post, etc).

How the Mercury Protocol Helps

The Mercury Protocol will be designed to combat these problems. It will inherit the benefits of blockchain technology, including the improved network security, user privacy, and data transparency previously discussed. Moreover, it will integrate Global Messaging Token, a new Ethereum blockchain token, that can also be leveraged to encourage positive user behavior, rewarding the timeliness, quality, or reach of communications, among others. The protocol and token combined will provide a scalable solution for smaller platforms to combat the incumbent bias contributing to the industry oligarchy.

Additionally, the protocol will enable decentralized and platform agnostic communication. This means it becomes possible to send communications from one Mercury Protocol integrated platform to another, thus expanding the ecosystem to include all users across all Mercury Protocol integrated platforms, as opposed to a single platform's users. The Mercury Protocol is applicable to all communication platforms, and can render these benefits for all types of messaging service providers.

Need for a Global Messaging Token

As previously mentioned, the Mercury Protocol will integrate a new blockchain token called "Global Messaging Token."

Tokens built on the blockchain can represent any digital asset. For example, tokens can be used to represent voting power in an organization, paid credits within an application, and almost anything imaginable. We see the invention of such a blockchain token as an opportunity to change how communication is done. More specifically, a token can be used as a utility metric for communication platform participation. The token would be completely run on the blockchain, and create a system that enables the creation of decentralized communication platforms that give more power to the end user in the form of more transparent information, privacy, security, and utility. If a user generates content of value (as determined by any given platform), they could be awarded tokens that can be exchanged for premium services within the platform. This access to more services can be leveraged to encourage users to participate in the platform. This is the core idea behind the Global Messaging Token.

While it would be possible to use an existing token like Bitcoin or Ether into the Mercury Protocol, existing tokens are valued against their parent blockchains fluctuating value. This fluctuation can cause problems when trying to balance a platform's ecosystem. For example, if a user spends one Bitcoin to send a premium message, and the value of Bitcoin spikes for a reason unrelated to the messaging platform, it could be argued that the user is losing out on that increase in value. By creating a new Global Messaging Token (GMT) that is valued by the utility of Mercury Protocol integrated apps, the token is more directly tied to the applications it's built into. This stability makes it possible to properly calibrate the token cost of a platform's premium services, and consistently promote positive network behavior.

It is our hope that as more platforms recognize the benefits of the Mercury Protocol and adopt it, the token will increasingly gain utility beyond the initial use cases outlined further down in this whitepaper.

Overview of the Technology

Global Messaging Token

“Global Messaging Token” or “GMT” is an ERC20 standard token that is built on the Ethereum blockchain. Dust is the first application to use this token, followed by the Broadcast application. For more details on the token, please refer to section “Overview of Global Messaging Token”.

Why Ethereum

Ethereum was designed with the explicit goal of doing more than just creating and recording transfers of a blockchain network’s native tokens. Instead, it can be thought of as a generalized blockchain technology with a built-in Turing-complete programming language. The language enables anyone to write programs with custom transaction formats and state transition functions, essentially specifying whichever rules they want that can then be uploaded to the blockchain, and the blockchain will automatically interpret the rules for them.

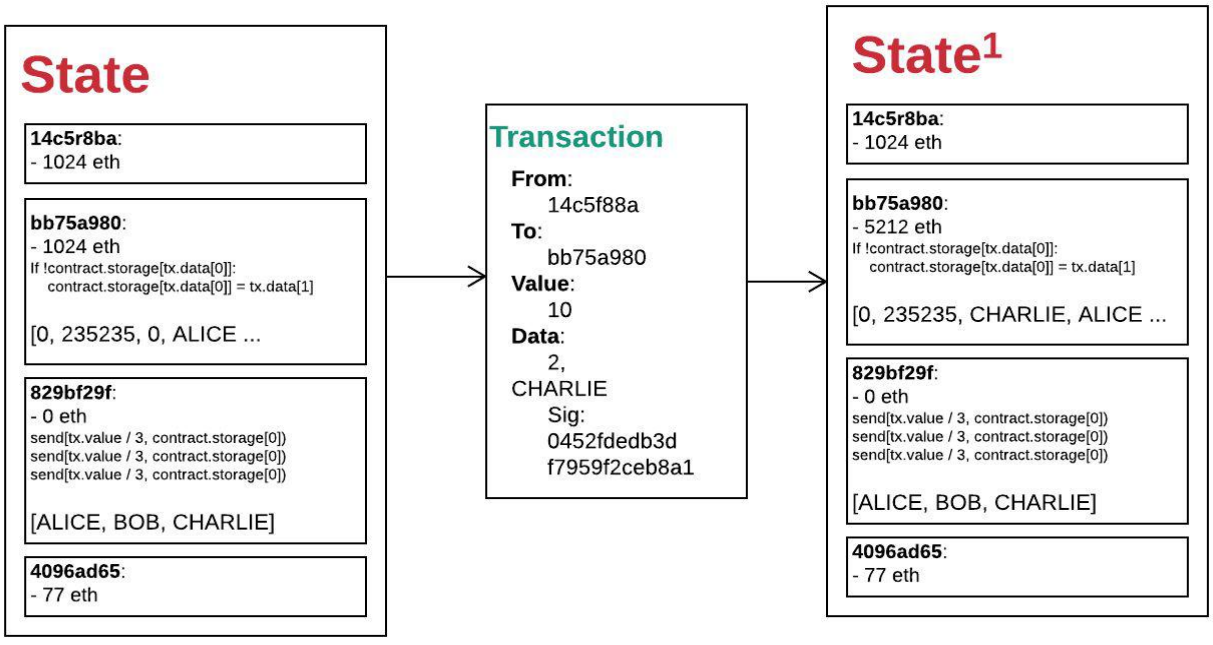


Figure 1: Ethereum is simply a transaction-based state machine: we begin with a “genesis state” and incrementally execute transactions to transform it into some final state

The generalized technology has enabled what are known as “smart contracts”, which are computer programs that directly control digital assets (e.g. tokens, domain names, IDs, etc) and can be used to

encode protocols on the blockchain. Contracts have their own addresses and own digital assets. A contract that owns digital assets can only send the asset to another party via the rules defined in the contract's code, and this transfer of digital assets is visible to every party in the network.

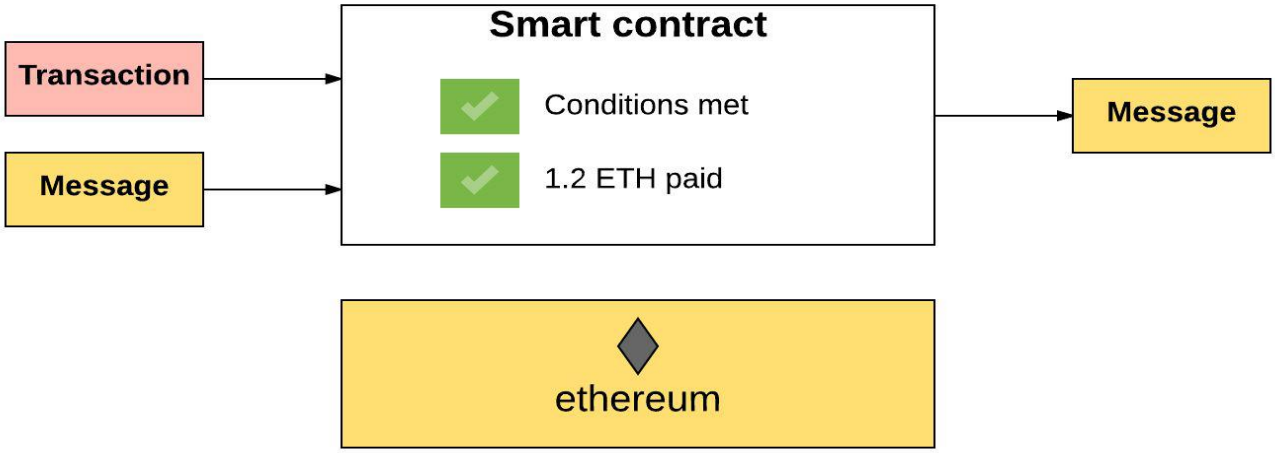
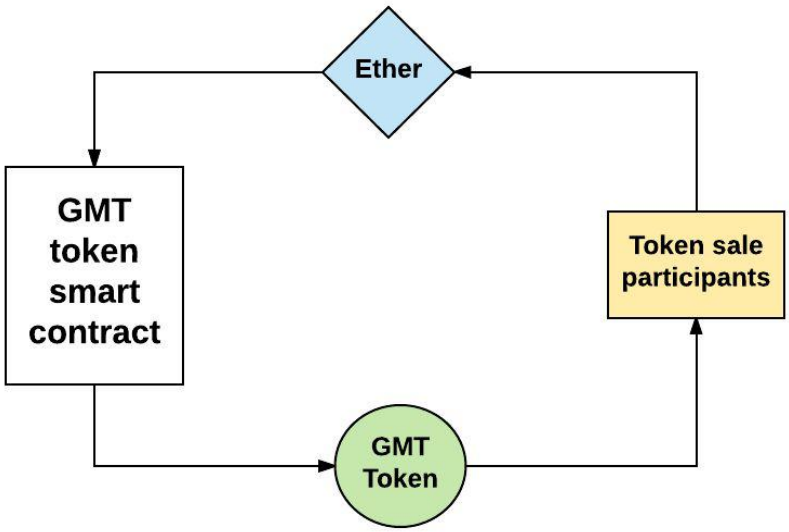


Figure 1: A smart contract is a distributed contract that is represented in code that is intended to facilitate, verify, or enforce the negotiation or performance of the contract. In Ethereum, smart contracts can accept and store Ether and data, and can send that Ether to other accounts or even other smart contracts.

Smart contracts can serve many different functions because they can encode and enforce complex issuance rules and automated incentive structures for cryptocurrencies, digital financial contracts, escrows, multi-party protocols (e.g. auctions), and more.

As a result, using smart contracts on the Ethereum blockchain is currently the industry standard for issuing custom digital assets. The advanced features and active ecosystem of Ethereum make it a natural fit for GMT and the Mercury Protocol.



State channels

The number of transactions on the Ethereum network is growing at a significant pace— daily transactions increased from 40,085 to 240,465 from Q2 2016 to Q2 2017 [3], which represents a 500% year-over-year growth. The daily transaction volume is currently between 225,000 to 300,000, and as Ethereum adoption grows, the transaction volume will continue to grow.

However, because the Ethereum blockchain is a decentralized network that is replicated on every node, it poses a scalability problem. The blockchain is replicated on every node in the network, which means every single node on the network processes every transaction and maintains the entire state. While this offers strong benefits such as auditability, fault tolerance, authenticity and political neutrality, it limits the throughput of transactions that can occur on the network.

Under perfectly contrived conditions, Ethereum's theoretical transaction processing capacity is over 1,000 transactions per second. However, due to Ethereum network's block gas limit, a more realistic limit is 10 transactions per second for simple transactions, and 1 to 5 transactions per second for complex ones [3]. The current throughput on Ethereum's network is ~8.5 transactions per second, or approximately 740,000 transactions daily [4].

This leads us to three potential issues:

1. As the Dust application acquires more users and adds more uses cases for the Global Messaging Token, the number of transactions will grow rapidly, which would lead to severe scaling challenges
2. The user experience would suffer because long transaction confirmation times would affect the responsiveness of the Dust app
3. On-chain transactions, regardless of size, are required to pay a fee. This is likely to cause users to refrain from freely using Global Messaging Tokens for large numbers of micropayments, which would hinder the growth of token usage on the platform

Although the long term goal for Ethereum 2.0 and 3.0 is for the protocol to be able to maintain a blockchain capable of processing VISA-scale transaction levels via mechanisms such as code pre-processing using precompiled contracts and WebAssembly [3], blockchain sharding, and/or new initiatives such as Plasma.io, the Mercury Protocol will need to use an interim solution for scalable interactions with the GMT tokens: **State channels**.

State channels are essentially a mechanism by which blockchain interactions which could and would normally occur on the blockchain instead get conducted off of the blockchain, without significantly increasing the risk of any participant while providing significant improvements in cost and speed. State channels will be a critical part of scaling blockchain technologies to support higher levels of use. They can be applied to payments, smart contracts, and many other scenarios.

A state channel is designed to be implemented as follows [5]:

1. Part of the blockchain state is locked via multi-signature or some sort of smart contract, where the only way to update it is if a specific set of participants agree completely
2. Participants make updates amongst themselves by constructing and cryptographically signing transactions without submitting it to the blockchain. Each new update overrides previous updates
3. At some later point, participants submit the state back to the blockchain, which closes the state channel and unlocks the state again

Steps 1 and 3 involve blockchain operations which are published to the network, pay fees and wait for confirmations. However, Step 2 does not involve the blockchain at all. It can contain an unlimited number of updates and can remain open indefinitely. In this sense, the blockchain is used purely as a settlement layer to process the final transaction of a series of interactions for the final settlement. Ethereum enables us to easily encode the above mechanism into a smart contract.

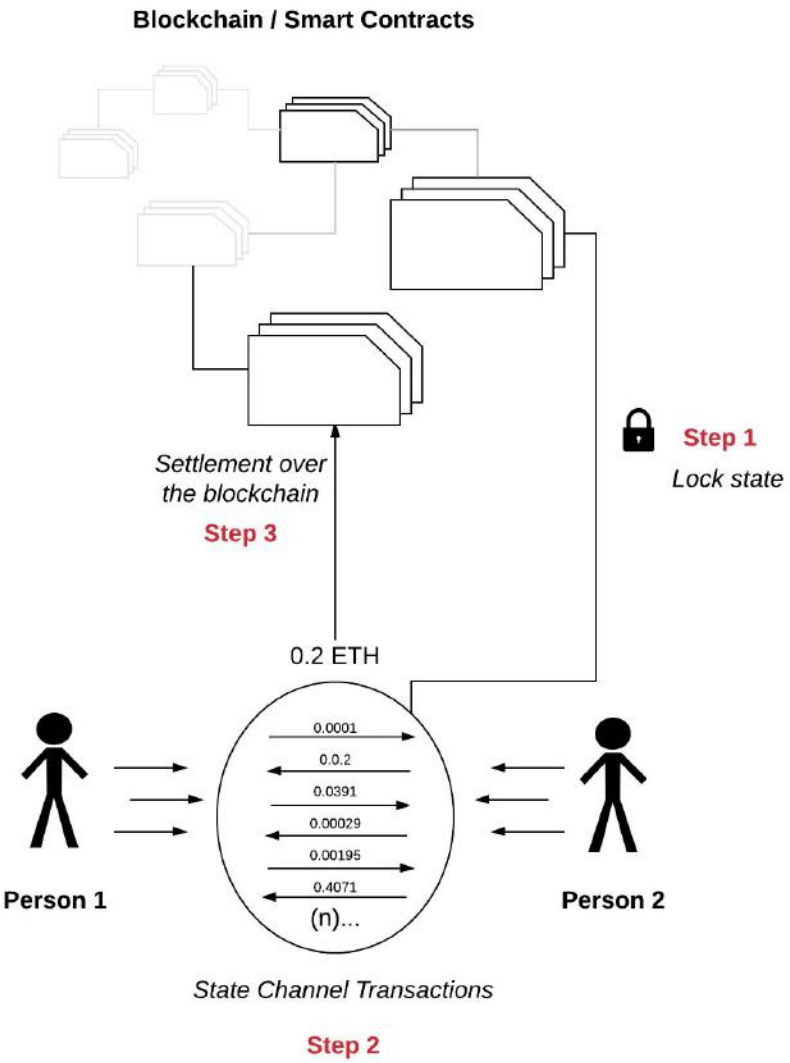


Figure 3: At any point during the process, any participant can send a transaction into the contract to close the channel and start a settlement procedure. This starts a time limit within which participants can submit transactions, and the transaction with the highest sequence number is processed. If one of the participants leaves or tries to cheat, another one can at any time publish the latest transaction to the blockchain to finalize the state, assuming all the participants completely agree on the state. [15]

In this regard, state channels enable any number of transactions to be processed instantly without facing the throughput limitations explained above. It also has potential to reduce high per transaction fees by 7x for handling large numbers of small digital transactions (i.e. micropayments) [7].

Oracles

Smart contracts are like walled gardens in that they cannot communicate with the external world. In other words, smart contracts are unable to fetch data from external APIs and data sources, and are restricted to the state maintained on the blockchain.

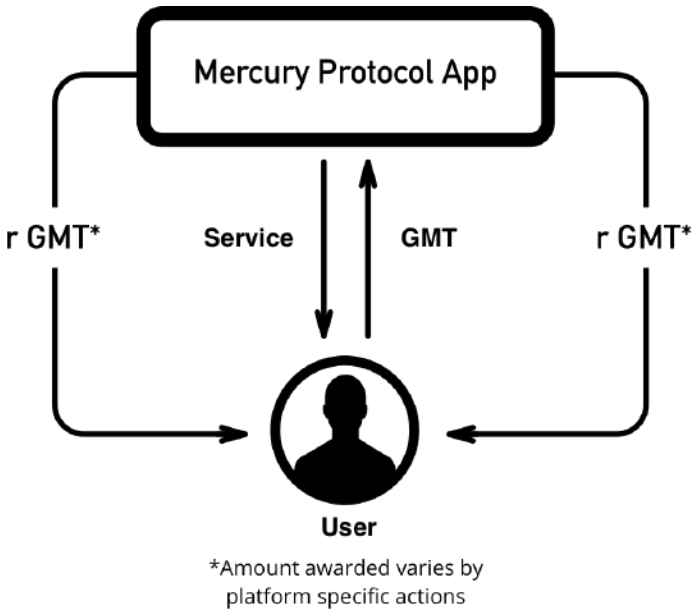
However, the Mercury Protocol will implement many features which will require the protocol to communicate with the external world; for example, to fetch content from an external CDN that then needs to be served to a set of end users, or to verify that a piece of content has been accepted and viewed by a user within an application. To facilitate such communication, we will use oracles. In the blockchain space, an oracle is a party which relays information between smart contracts and external data sources. While we plan to integrate oracles into the protocol in the future, the specific implementation is yet to be determined.

We will publish the Mercury Protocol Technical Whitepaper that further delves into the underlying technical architecture, as well as a detailed description of features and functionality provided by the protocol.

Use Cases for Global Messaging Token

Token Flow

Mercury Protocol integrated platforms will be able to integrate Global Messaging Tokens (GMT) into their ecosystem. Users can earn GMT by performing platform specific incentivized actions (e.g., consuming x posts a day, reading a message within y seconds, etc), and then use these earned tokens to receive premium services on the platform. The app will keep a small amount of tokens as payment for services rendered, and allocate the rest to award to users performing specific actions that add value to the platform (e.g., generating content, achieving n subscribers, etc).



Exchange GMT for Services

Historically, there have been three main forms of communication:

1. **1-to-1:** “An exchange” - A back and forth dialogue between two parties, such as written letters, traditional phone calls, or one-on-one text messaging.

2. **1-to-Many:** “An announcement” - A one directional broadcast, such as radio, television, or public speech.

3. **Many-to-Many:** “A discussion” - A collection of individual people or entities, such as a town hall meeting, a live digital chatroom, or cocktail party.

Global Messaging Tokens can be used to obtain access to numerous services in all three types, and across a variety of platform types. Whenever users use GMT to obtain a service, the majority of GMT paid will go to the platform rendering service. For example, Dust currently envisions many possible services for users to utilize in exchange for GMT, including but not limited to:

In the 1-to-1 private messenger portion of Dust

1. Extending the lifetime duration of a message beyond the standard twenty-four hours.
2. Allowing users to send larger message payloads such as .PDF, .MOV, etc.
3. Sending special types of messages, such as an “urgent” type that can repeatedly notify the recipient until answered.
4. Sending messages to a Mercury Protocol integrated platform outside of the Dust app.

In the 1-to-many “Blast” portion of Dust

5. Increase in-app reach by sending a blast to a specified number of users who do not follow the content creator.
6. Send a blast to the followers of a specific account in order to target an existing social group or demographic.

Beyond the two forms of communications within Dust, there are several app-specific features we plan to implement to enable further use of GMT:

7. Featuring users on the front page of the discover tab to help increase their following.
8. Third party apps trade GMT to embed a Dust messenger client into their platform.

Receive GMT to Incentivize Participation

It is important to note that GMT is designed and intended to be obtained primarily from a service provider when a user performs one or more various incentivized actions. When a platform is given GMT to provide a service, a portion of the GMT should be allocated to incentivize user behavior. A given ecosystem might award GMT to users for participating in the following ways:

In the role of content creator:

1. Generating quality content, as determined by content consumer feedback by platform
2. Creating content often and/or consistently

In the role of content consumer:

3. Generating de-identified behavioral data or giving attention
4. Determining platform content quality by providing feedback
5. Responding within a certain timeframe to notifications or messages
6. Sharing/”reposting” content

Other possible incentivized actions:

7. Consistent logins
8. Rewarding users for inviting friends to the platform.
9. Giving attention to premium distributed content (ads)
10. Achieving milestones within the platform (e.g., x number of subscribers, x messages sent, etc.)

This incentive system allows users to generate a positive token balance simply by being active on the platform, and thus grant access to the premium experience without having to acquire resources from outside the ecosystem.

While not all of these GMT based features are guaranteed to be in the first version of Dust available after the token sale, many will be, and we will continuously be working to add more GMT integrated features to both Dust and Broadcast. We will communicate the progress we make in integrating these and new use cases via our blog.

Use cases in the Industry at Large

Beyond the above example use cases specific to Dust, there are many other ways to integrate a Global Messaging Token. In the 1-to-1 communication format, for example, encryption, archiving/storage, or sending a message to a user on a different Mercury Protocol integrated messaging platform. Receiving a premium message from a user not in a user's contact book or friends list could award GMT in exchange for their attention.

In the 1-to-many broadcasting type of communication, some premium use cases might include sending content to other users who are not subscribed to a content creator or scheduling the release of content for a later time or date. Positive feedback on content (likes, comments, etc.) could award GMT to a user as a reward for quality content, promoting positive engagement between content creators and consumers. The higher the content quality, the more enjoyment consumers experience and the more GMT the creator receives.

In the more chaotic many-to-many open forum discussion type of communication, a use case to could be using GMT to obtain the attention of the collective, whether in the form of speaking time before an audience or visual flair in a real-time chatroom. Another use case might be paying a GMT premium to join a shorter queue to deliver one's message. In the case of a chatroom where premium users are using GMT to speak or message without waiting in a queue, non-premium users should receive that GMT for having their queue time extended.

In each of the communication types, GMT can be leveraged to encourage positive behavior in the ecosystem for both content creators and consumers. Additionally, by providing opportunities to obtain GMT through common participation within the ecosystem, any given communication platform can allow less affluent users to partake in the same premium environment as the established users. When properly balanced, this system is theoretically self-correcting, as users who negatively contribute to the platform (a.k.a. "trolls") will have significantly less access to premium services.

Roadmap

Phase 1: Dust *(Q3 2017)*

We will build version 1.0.0 of the Mercury Protocol, conduct the token sale, and integrate GMT into Dust. The Mercury Protocol will remain closed-source at this stage. Security audits for the protocol and token smart contracts will be conducted before and after shipping a GMT integrated product. Some finances from the token sale will be used to expand the development team and increase throughput on feature development as we move forward.

Phase 2: Broadcast *(Q4 2017)*

The next step is completing version 1.0.0 of Broadcast and shipping to market. The project is currently in pre-alpha, and we are on target to have a first version in market by end of year. In addition, we will drive further development of the Mercury Protocol and use cases for GMT within Dust.

Phase 3: Beyond the Platforms *(Q1 2018 and later)*

Once Dust and Broadcast are both in market with GMT integrated features, we will open source the Mercury Protocol. We will continue the development of value added services for Dust and Broadcast. Security audits will be conducted in an on-going manner as more features and functionality is added to the protocol.

Throughout the above three phases, we will continue to evangelize the protocol and token model in pursuit of a new global standard of messaging ecosystem that is designed to be decentralized, more secure, and auditable.

Adoption

Dust is currently in-market, averaging at least 20,000 daily active users and over 200,000 messages sent per day. The existing user base has proven to be loyal and consistently active, making Dust an ideal platform to demonstrate the versatile potential of a Global Messaging Token. Moreover, having Dust in-market with a patent pending on the in-memory solution demonstrates our team's ability to tackle complex technical problems and deliver a market-ready product. We expect to see the number of daily active users and average actions per day rise significantly in the near future. As we gain knowledge about the GMT integrated features in Dust, we will look to continue to expand use cases in both Dust and Broadcast.

Similar Products

	Decentralized Token System	**In Market 2018**	**Participation Incentive**	**Open Source Protocol**	**Strong Focus on Privacy**
Status	✓			✓	✓
WeChat		✓			
Toshi	✓	✓		✓	✓
Messenger		✓			
Signal		✓		✓	✓

Dust	✓	✓	✓	✓	✓
Broadcast	✓	✓	✓	✓	✓

Status

Status is a decentralized messenger and browser that implements a token based on Ethereum. The platform also contains a DApp ecosystem, allowing developers to publish their own Ethereum based apps. This ambitious platform is currently in Alpha, however, and their official roadmap shows no intention of hitting market until Q3 2019 at the earliest.

WeChat

WeChat is the dominant communication platform in China, providing services for everything from private messaging, to publically sharing “moments” similar to a social media post, and even allowing users to send money via smart-contract-esque microtransactions. While WeChat provides similar features to those proposed in this paper, it is a centralized, non-blockchain based system, which means it has increased security risks and fails to incentivize end-users to participate.

Toshi

Toshi is a decentralized mobile app for Ethereum apps with integrated private messaging and an Ethereum wallet. Inspired by WeChat is designed with the intention of providing secure financial services to everyone who has a smartphone. Toshi does not incentivize users to participate. Moreover, since the platform is Ether based, any future implementation of user incentivization would either produce inconsistent incentives due to ether market volatility, or require significant resources to transition to a token based system.

Messenger

Facebook Messenger is primarily a private messenger, with the ability to send multimedia messages, stickers, make video or audio calls via internet, and recently added the ability to send money. While high quality, established, and stable, Messenger is the epitome of a centralized communication platform. Facebook rakes in over $8 billion USD annually [9], most of it through advertising, and fails to reward its users who generated it by providing their attention. This non-blockchain based platform has increased security risk for users by relying on the defenses of a single party, does not open source code for integration opportunities or peer review, and routes financial transactions through a centralized third-party that lacks the transparency and trustless consensus provided by decentralization.

Signal

Signal is a private communication platform featuring end-to-end encrypted messaging, VOIP, and video calling built with the intention of “dismantling the surveillance states globally” [14]. The underlying open-source Signal Protocol that encrypts communications is widely considered to be one of the more secure methods of communication, ensuring message integrity and attribution. This protocol has been

integrated into other major messaging platforms such as WhatsApp and Messenger. Signal is funded through public grants and donations, so it does not conform to the previously discussed advertising model. But Signal is decidedly centralized [14], and, unlike Dust, stores a record of messages exchanged. This puts conversation history at risk if a malicious party manages to copy the data and brute force crack it at their leisure. Even if a third party cannot gain access to the content of the messages, it is still recorded that a conversation occurred between two or more parties. Moreover, Signal does not incentivize user participation, and lacks an auditable token system.

Overview of Global Messaging Token

The token sold during the token launch is known as the Global Messaging Token, or GMT. This token is designed for various utilities in messaging apps designed to be compatible with the Mercury Protocol. Dust app is the first app that is proposed to use this token, followed by the Broadcast app.

ERC 20 token

The GMT token is implemented on the public Ethereum blockchain (using an Ethereum smart contract) as an ERC20 token [6]. ERC20 establishes a standard contract ABI for tokens on the Ethereum blockchain and has become the de facto representation for all types of digital assets. All ERC20 tokens share the same core contract interface, simplifying integration with external contracts.

Core ERC20 functions include:

- transfer(to, value)
- balanceOf(owner)
- approve(spender, value)
- allowance(owner, spender)
- transferFrom(from, to, value)

In addition to the core ERC20 interface, the GMT token smart contract will have additional functionality to create new GMT tokens, finalize the crowdsale and send money to a hardware wallet, and refund users in the case of a failed crowdsale.

The ERC20 token smart contract is written in the Solidity programming language and will be publicly accessible and free to use.

Hardware wallet

Both ETH and GMT tokens available for sale will be stored in hardware wallets.

A hardware wallet is another form of cold storage which allows digital assets to be stored offline. They provide increased security because they remove the process of having to load the private key to some software which is exposed to online vulnerabilities.

Hardware wallets have two parts – a connected device and a disconnected device. The connected device stores the public keys, chooses which transactions to sign, and essentially has the same functionality as any paper wallet. The one main difference is that it cannot sign a transaction because the offline device holds the private key.

When you want to sign a transaction, you must connect the device via a USB port or a QR-code. The transaction is then sent to the offline device, which then generates a signed transaction using the private key stored on the device and sends it back to the connected device, which is finally fed into the blockchain network for the standard Bitcoin/Ethereum verification process required for it to be included in the blockchain.

Our hardware wallets will have backup keys split and distributed geographically. We feel this will provide us the utmost safety guarantee against security attacks.

Liquidation

GMT tokens that are purchased by the public will be delivered as soon as ETH is received. The GMT tokens will be immediately usable after the token sale as soon as the relevant Dust integrations are publically released.

Security

The Dust team takes security seriously and will take security precautions designed to safeguard the funds. We intend to only launch the crowdsale once we are assured that all security measures have been completed.

This includes:

1) A thorough security audit by Open Zeppelin for all smart contracts used for the crowdsale
2) All GMT tokens stored by us will be stored offline in hardware wallets
3) A majority of the reserve funds will also be stored offline within hardware wallets
4) The keys for hardware wallets will be geographically distributed
5) In the future, we plan to launch a bug bounty program for all smart contracts we develop

The smart contracts for GMT token sale are currently under review by Open Zeppelin. Once completed, the review will be publicly available on our blog. We will address any and all severe or critical vulnerabilities suggested by Open Zeppelin in the final implementation of the smart contracts. The final implementation and code for the smart contracts will be released publicly prior to the token launch.

Governance

In the near future, we will define and develop a governance model around GMT such that participants who have a certain amount of stake in the overall ecosystem will have voting power to participate in the decisions made around future innovation of the Mercury protocol, and defining community standards for

platforms who integrate GMT, among other topics. We will initially manage the protocol codebase until such time as the governance model has been properly established.

We will distribute a Whitepaper that includes a more detailed explanation of the proposed governance model after the token sale.

Team

Core Team

- **Ryan Ozonian,** *Chief Executive Officer*
- **Rohit Kotian**, *Chief Technology Officer*
- **Igor Shpitalnik**, *Lead Backend Engineer*
- **Ian R. Connelly***, Project Manager*
- **Preethi Kasireddy***, Lead Blockchain Developer*
- **Alex Moir**, *Blockchain Developer*
- **Elliot Sperling**, *Blockchain Developer*
- **Sameer Khavanekar,** *iOS Team Lead*
- **Manju Deshpande***, iOS Developer*
- **Alex Rupprecht**, *Android Team Lead*
- **Brant Kortman**, *Android Developer*
- **Evan Albert***, Lead Web Developer*
- **Mayukh Das**, *Software Developer*
- **Michael Talarczyk***, Lead Systems Engineer*
- **Jon Ikemura***, Lead Designer*

Advisors

- **Mark Cuban,** Dust Investor, Dallas Mavericks
- **Nick Tomaino,** 1Confirmation, Previously at Coinbase, The Control, Runa Capital

References

1. https://etherscan.io/chart/blocktime
2. https://etherscan.io/chart/tx
3. http://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/57506f387da24ff6bdecb3c1/1464889147417/Ethereum_Paper.pdf

4. https://ethereum.stackexchange.com/questions/1034/how-many-transactions-can-the-network-handle
5. Jeff Coleman. State Channels. http://www.jeffcoleman.ca/state-channels/
6. ERC20 is the Ethereum token standard. https://github.com/ethereum/EIPs/issues/20
7. https://media.consensys.net/state-channels-ethereum-is-open-for-business-5b7cd4d7506c
8. https://www.wsj.com/articles/the-race-is-on-to-challenge-google-facebook-duopoly-in-digital-advertising-1497864602
9. https://investor.fb.com/investor-news/press-release-details/2017/facebook-Reports-Fourth-Quarter-and-Full-Year-2016-Results/default.aspx
10. http://www.investopedia.com/ask/answers/120114/how-does-facebook-fb-make-money.asp
11. https://www.usatoday.com/story/tech/personal/2013/06/15/techlicious-text-message-alternatives/2423169/
12. https://www.wired.com/2015/08/time-to-ditch-texting/
13. https://www.emarketer.com/Article/US-Ad-Blocking-Jump-by-Double-Digits-This-Year/1014111
14. https://techcrunch.com/2016/11/07/signal-app-maker-rebuts-criticism-of-dev-direction-by-calling-for-more-community-help/
15. https://hackernoon.com/blockchains-dont-scale-not-today-at-least-but-there-s-hope-2cb43946551a