🌮 Wrap Protocol 🌮

Whitepaper v1.0
Bender Labs

Abstract

In this paper we present Wrap, a decentralized bridge between the Ethereum and Tezos blockchain which allows anyone to wrap ERC20 tokens into FA2 tokens. Wrap is a solution for liquidity and asset interoperability between different blockchains, and is the first step to build a healthy and growing DeFi ecosystem on Tezos. We introduce the $WRAP token, a governance token allowing participants in the Wrap Protocol to vote on protocol upgrades, parameters change, and to access a portion of the fees generated by the wrapping activity. Wrap Protocol will go live in Q1 2021 with more than 15 ERC20 tokens supported.

Contents

I. Cross-chain liquidity and interoperability

A. The first step towards building DeFi on Tezos

To understand how and why we're building Wrap Protocol, we need to take a step back on the mission of Bender Labs.

Bender was created to build an open financial system running on Tezos. This open financial system is made up of smart contracts replicating the services offered in the traditional financial industry in a decentralized, open and unstoppable way. All the protocols built by Bender Labs share 3 important common characteristics: they are open to anyone, built with pragmatism and owned by their users.

But how good is such an open financial system if there's no assets or liquidity to power it? *"All dressed up, nowhere to go"* goes the saying, which is precisely what we're trying to avoid by building Wrap Protocol.

Wrap is our way to bring assets and liquidity to Tezos. Since the 2017 ICO boom, Ethereum has been the reference blockchain for companies and projects to issue tokens, thanks to the ERC20 standard. Today, there exists close to 350,000 ERC20 tokens[1] cumulating hundreds of millions of dollars in market capitalization. The recent advent of DeFi on Ethereum was arguably fostered by the wide range of assets and the depth of liquidity available to DeFi developers and users. Building a DeFi ecosystem on Tezos starts with bringing liquidity and assets on the Tezos blockchain: this is the purpose of Wrap Protocol.

B. A larger problem: decentralized interoperability of different blockchains

The liquidity problem that we're trying to solve on Tezos is nothing but an instance of a broader problem: how to achieve decentralized interoperability of two different blockchains. To put it simply, wrapping is a way to transform any blockchain A into a pegged sidechain of a blockchain B. This way, one can use the consensus mechanism and specific infrastructure of blockchain A to use assets or information stored with both A and B.

Blockstream defines a sidechain in their original whitepaper[2]:

- A sidechain is a blockchain that validates data from other blockchains.
- Two-way peg refers to the mechanism by which coins are transferred between side chains and back at a fixed or otherwise deterministic exchange rate.

[1] Source : Etherscan

[2] https://www.blockstream.com/sidechains.pdf

- A pegged sidechain is a sidechain whose assets can be imported from and returned to other chains.

In our case, we want to transform Tezos into a sidechain of Ethereum first, then potentially other blockchains. The Tezos blockchain is equipped with its own consensus, which means we can narrow down our focus to the peg mechanism. In a way, this pegging mechanism is strongly related to the oracle mechanism : an external program or user must watch for facts that the sidechain cannot observe by itself. These facts then trigger some computation on the sidechain.

Our use case does not require us to build a general oracle mechanism. Events that must be observed are transactions that occurred on other blockchains, and the computations are limited in number :

- lock assets in the mainchain
- mint tokens in the sidechain
- burn tokens in the sidechain
- unlock assets in the mainchain

Today, there are 5 ways to solve this problem:

- **Centralized solution:** a trusted oracle gathers facts, and trigger computations on both chains.
- **Off-chain federation, aka federated peg:** a group of oracles gather facts, run their own consensus mechanism, and trigger transactions on both chains.
- **Oracle Aggregation**: instead of relying on an off-chain consensus, oracles provide facts to the sidechain right away, and a contract is in charge of aggregating and validating before running the actual computation.
- **Synthetic Peg**: do not try to enforce a 1:1 peg, but instead, try to enforce a price peg.
- **Atomic swaps:** which allow the exchange of assets from one blockchain to another through the use of smart contracts.

C. Existing solutions

1. Centralized Custodians

The most straightforward solution to make two blockchains interoperable is to use a (or several) centralized custodian. The custodian is an entity which holds assets from the main chain on behalf of users and issues corresponding assets on the sidechain. Often, the custodian needs to be regulated for such an activity.

By and large, this is the model of very successful bridges, such as WBTC (BTC on Ethereum) and USDC (USD on Ethereum , c*onsidering the US Dollar relies on a - very centralized - basechain of its own*).

2. Off-chain Federation, aka Federated Peg

The second model which comes to mind is the off-chain Federation, or Federated Peg. The founding principle of an off-chain federation is that its members cannot be trusted individually, but the federation can. Thanks to a set of incentives (rewards to behave correctly, and deterrents not to behave maliciously), the off-chain federation as a whole can be trusted.

The idea of federated peg is developed by Blockstream in a paper[3] of 2016.
A federated peg relies on a group of separate centralized custodians. Instead of letting each custodian manage their own custody wallet, and issue assets on the sidechain, each of them has to participate in a consensus to forge transactions - both on the main chain and the side chain. All assets locked on the main chain are controlled by a consensus of custodians, and so is the process of issuing and burning assets on the side chain. The consensus can be achieved through multisignature, or threshold multiparty computation.

To make sure that custodians do not collide with each other to steal the “honeypot”, a set of incentives has to be added to the federation, such as a fee paid by users of the peg to members of the federation behaving correctly, as well as a bond to be posted by federation members, which can be slashed in case of malicious activity.

One of the first and most used off-chain federation is Liquid by Blockstream which acts as a fast settlement network for Bitcoin traders and exchanges.

There have been further efforts to make off-chain federations more decentralized, by creating on-chain mechanisms to open them to new members and decentralizing their core logic. RenBTC is one example of such a federation, working in a fully decentralized fashion to offer a bridge between BTC, BCH and ZEC and the Ethereum blockchain.

3. Oracle Aggregation

This approach is similar to the off-chain federation: several members gather and interact together in such a way that the group can be trusted but not its individual members. Rather than holding keys to sign transactions on both chains, members interact with smart contracts as oracle. This obviously requires the chains to support smart contracts.

Instead of having a mechanism allowing federation members to “talk” together to forge a valid transaction on either chain, each member here directly sends observations on the state of the two blockchains to an oracle smart contract. The oracle smart contract is then in charge of

[3] https://arxiv.org/abs/1612.05491

aggregating the results, computing a consensus and acting on it (release funds on the basechain, mint assets on the sidechain). Likewise, incentive mechanisms such as bonds and rewards put in place to limit the risk of a sibyl attack have to be coded on-chain on a smart contract.

In this approach, the protocol relies heavily on smart contracts :

- no need for gossip and network communication between oracles : the smart contracts act as a central place to gather and share information on the state of both the basechain and the sidechain
- no need to forge an off-chain consensus : the rules of the consensus are enforced by the smart contracts directly before accepting the actual computation

One of the main oracle systems operating this way is Chainlink, which provides data and information from off-blockchain sources to smart contracts on several different blockchains.

4. Synthetic Pegs

Somewhat different than previously mentioned solutions, synthetic pegs are a way to guarantee a 1-1 correspondence in value between the asset on the sidechain and the asset on the basechain. In the traditional financial world, this is similar to holding a synthetic asset to gain exposure to an underlying asset. Convertibility of assets is necessary to impose a perfect peg.

In most cases, synthetic pegs rely on bonds that have to be deposited on a smart contract by a user. Based on the relative value of the asset in bond and the asset on the mainchain, the smart contract issues assets on the side chain. If the price varies too much, the smart contract either sells the bond, asks for a larger bond or releases some of the bond to account for relative price fluctuations. Synthetic pegs require a stability fee (which can be positive or negative) to address the inherent risk of printing synthetic assets against collateral.

DAI is a prime example of a synthetic peg. By depositing ETH on one of MakerDAO's smart contracts, users can withdraw DAIs, worth 1 dollar each. The value of DAI is backed by the dollar value of ETH deposited by the user in the smart contract. If the ETH dollar price drops below a certain level, the smart contract will liquidate the ETH collateral to maintain the DAI peg to USD.
MakerDAO has been one of the longest running DeFi projects on Ethereum.

5. Atomic Swaps

In his paper "Atomic Cross-Chain Swaps"[4], Maurice Herlihy defines an atomic swap as *"distributed coordination task where multiple parties exchange assets across multiple blockchains, for example, trading bitcoin for ether."*

[4] https://arxiv.org/pdf/1801.09515.pdf

Atomic swaps allow users to exchange their assets on two different blockchains in a decentralized way. They rely on smart contracts called hash time locked contracts, or HTLC which release the funds they store upon presentation of a secret or after a certain time.

A basic explanation of how an atomic swap works:

- Alice wants to trade her ETH for Bob's XTZ
- Alice deposits her ETH into an HTLC and generates a key to access it
- Alice shares a hash of this key with Bob. At this point, Bob cannot access the ETH because he only has a hash of the key - not the key itself
- Bob uses the hash provided by Alice to create an HTLC on which he deposits his XTZ
- To claim the XTZ, Alice has to reveal her key, which can then be used by Bob to access her ETH
- If Alice doesn't reveal her key, or if Bob doesn't deposit his XTZ, the HTLCs send back the funds to respectively Bob and Alice after some time

A fairly recent innovation, Atomic Swaps have a great potential to solve trustless interoperability of blockchains, especially in the context of decentralized exchanges. One of the drawbacks of atomics swaps is that they do not pass consensus operations: they only make sure that two operations on two different chains happen atomically - without moving the state or value across chains.

6. Comparison and technical choices

To design our own version of a bridge between Tezos and Ethereum, we've compared these existing solutions across 6 characteristics:

- **Complexity:** how complex is the solution? How hard is it to develop?
- **Level of Trust:** what is the degree of decentralization of the solution? Is it trusted, and if so what is the level of trust required by its users? Can the solution be censored or is it unstoppable
- **Security**: How secure is the solution? Is it easy to attack? Is it prone to errors or mistakes?
- **Transaction costs**: is the solution costly to use? Does it require a lot of on-chain operations?
- **Speed**: from a user perspective, how fast is the solution?
- **Dependencies**: last but not least, does the solution require a stack of existing tools on the Tezos or Ethereum blockchain?

The following table lays out our conclusions.

From 🤖 to 🤖🤖🤖	**Centralized Custodian**	**off-chain Federation**	**Oracle Aggregation**	**Synthetic Peg**	**Atomic Swap**
Complexity	🤖	🤖🤖	🤖🤖	🤖🤖🤖	🤖
Level of Trust	🤖🤖🤖	🤖🤖	🤖🤖	🤖	🤖
Security	🤖	🤖🤖🤖	🤖🤖🤖	🤖🤖	🤖🤖🤖
Tx costs	🤖	🤖	🤖🤖	🤖	🤖🤖🤖
Speed	🤖🤖🤖	🤖	🤖	🤖🤖🤖	🤖
Dependencies	🤖	🤖	🤖🤖	🤖🤖🤖	🤖🤖

One of the core statements of Bender Labs is to build with pragmatism. Full decentralization and trustlessness, although appealing, often stand in the way of usability and user-friendliness. While this will likely change in the years to come, we believe that today there is a fine line to find between centralization and decentralization to optimize for user experience. Said otherwise, given two solutions with an acceptably low level of trust required from users, we will choose the one which is the easiest to implement and use.

Moreover, we see Wrap Protocol as a means to an end, which is to build DeFi protocols running on Tezos. The mission of Bender Labs is not to build a perfectly decentralized solution to blockchain interoperability, but to build an open financial system on Tezos. For this to happen, we designed Wrap Protocol as a quick-to-implement, easy to use and understand, and with a reasonably low level of trust decentralized protocol that will allow to quickly and securely bring assets and liquidity on Tezos.

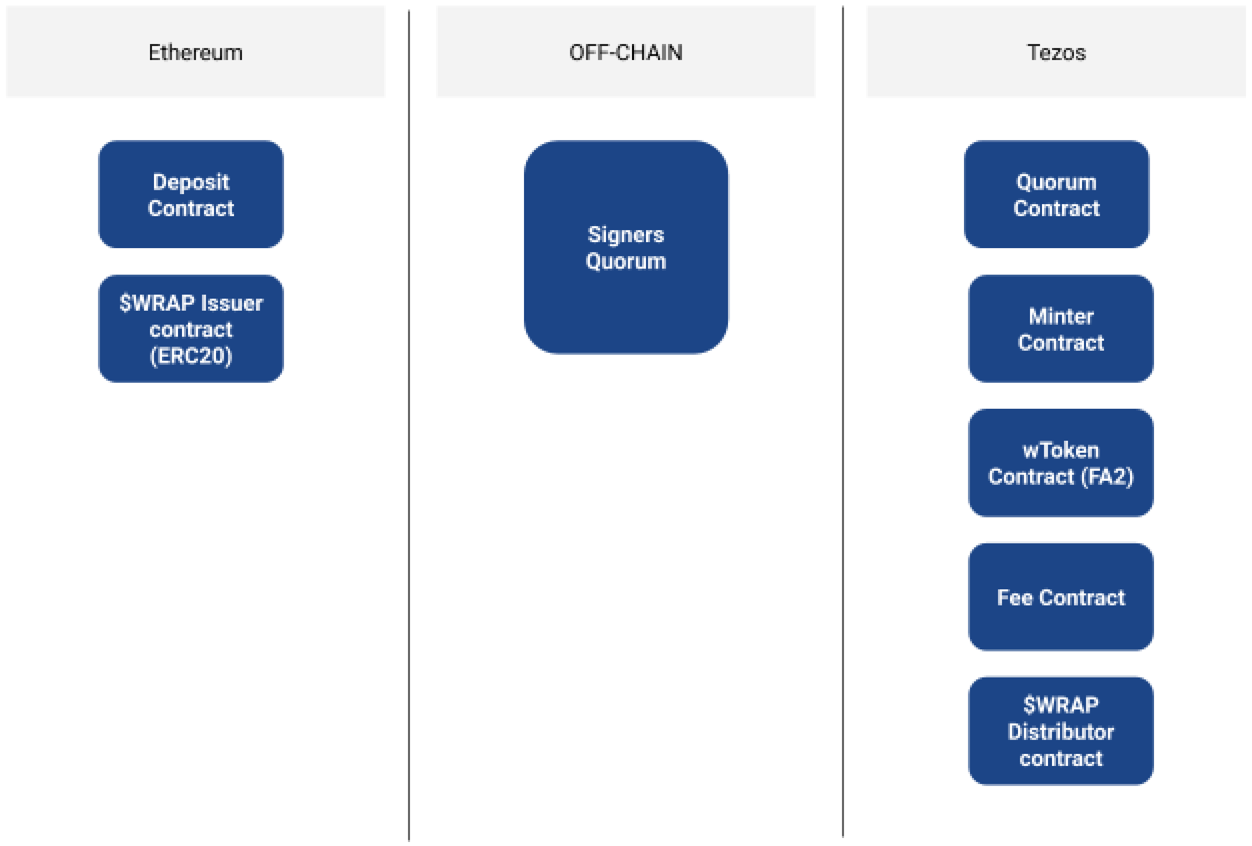
For all these reasons, we've decided to rely on an off-chain federation to build Wrap Protocol. off-chain federations offer a good level of "trustlessness" (its members cannot be trusted but the federation can - under certain conditions). It is easy to implement, without important technical dependencies and offers interesting features in terms of speed and transaction cost. The next part explains in more detail how Wrap Protocol works.

II. Wrap Protocol by Bender Labs

A. High Level Overview

Wrap Protocol is our solution to bring liquidity on Tezos and solve the cross-chain interoperability problem in a decentralized yet easily usable fashion.
At a high level, Wrap allows anyone to wrap ERC20 tokens on the Ethereum blockchain into FA2 tokens on Tezos, then use them on Tezos-native DeFi protocols. Wrap relies on an off-chain federation, that we called the Signers Quorum, which makes sure that at any given time the amount of wrapped tokens (that we'll call wTokens) issued on the Tezos blockchain corresponds to the amount of original tokens locked on Ethereum.

Wrap Protocol: High-Level Architecture



Wrap protocol is made up of different components, both on and off chain.

Ethereum Smart Contracts:

Deposit Contract: multisig contract managed by the Signer Quorum on which users deposit ERC20 tokens to be wrapped. The Deposit Contract also unlocks ERC20 assets during the unwrapping process. The specificity of the Deposit contract is that anyone can call the transfer method, given the proper signatures from the Signers Quorum. Initially, the contract is a 3-of-5 multisig contract.

$WRAP Issuer: ERC20 token smart contract, used to generate the initial supply of $WRAP.

Tezos Smart Contracts:

Quorum Contract: multisig contract managed by the Signers Quorum. Quorum Contract is the only address which can call the mint method of the Minter Contract. Initially, the contract is a 3-of-5 multisig contract.

Minter Contract: smart contract which encompasses all the wrapping logic on Tezos (including wrapping fee levels and recipients). It is the only address which can call the mint and burn methods of the wToken Contract.

wToken Contract: FA2 token contract, which regroups all wTokens on the Tezos blockchain. Mint / Burn methods can only be called by the Minter Contract.

Fee Contract: Smart contract on which wrapping and unwrapping fees are sent directly from the wToken Contract. Fee Contract allocates fees based on a preset repartition - that the community of $WRAP holders will be able to change though voting.

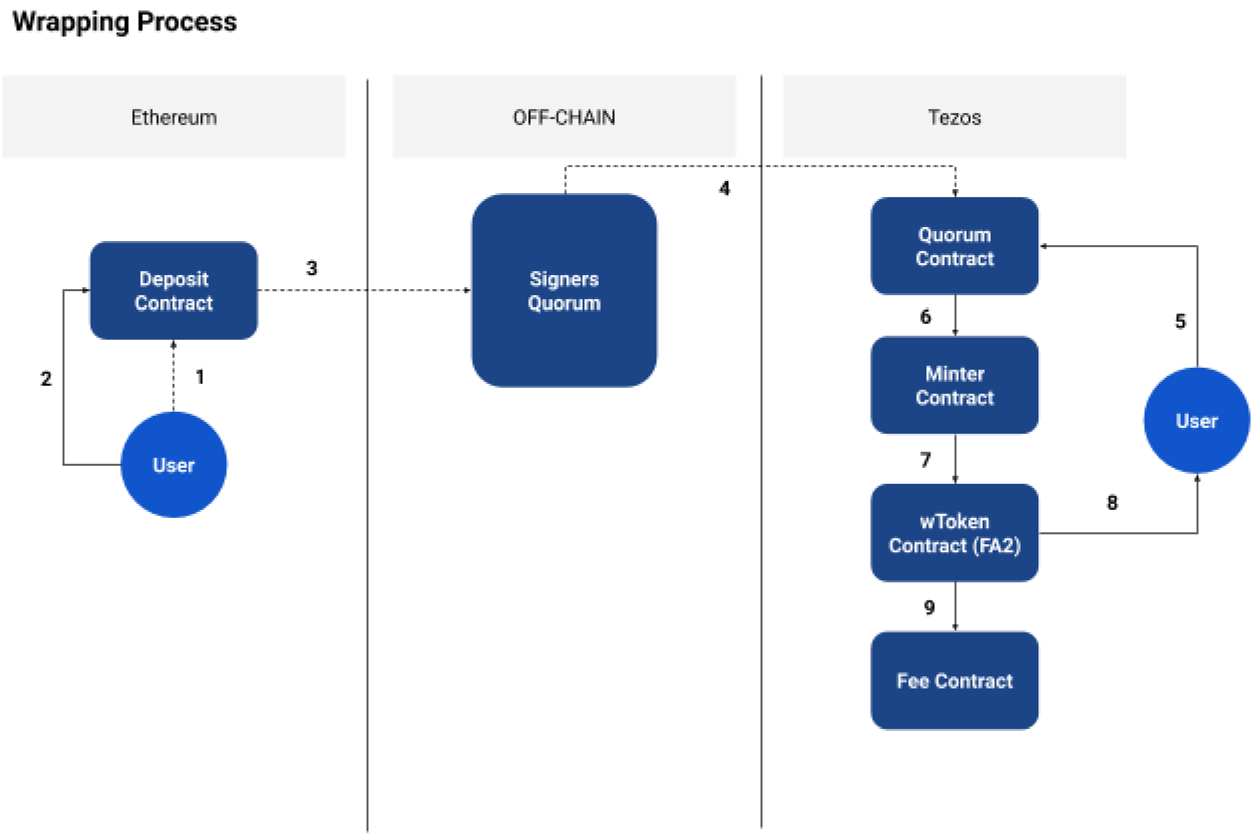
$WRAP Distributor Contract: Smart contract which manages the distribution of $WRAP tokens to Wrap Protocol participants.

Other:

Signers Quorum: set of several entities each controlling one key of both the ETH Quorum and the TEZ Quorum. The role of the quorum is to process wrapping and unwrapping transactions, maintain a 1-1 peg between tokens locked on Ethereum and tokens issued on Tezos and add more assets to the list of assets supported by Wrap Protocol. Signers watch both blockchains, and create and sign relevant transactions for both the Deposit Contract and the Minter Contract. Signers interact with the protocol through IPFS. Initially, the quorum is made of 5 signers.

B. Wrapping and Unwrapping

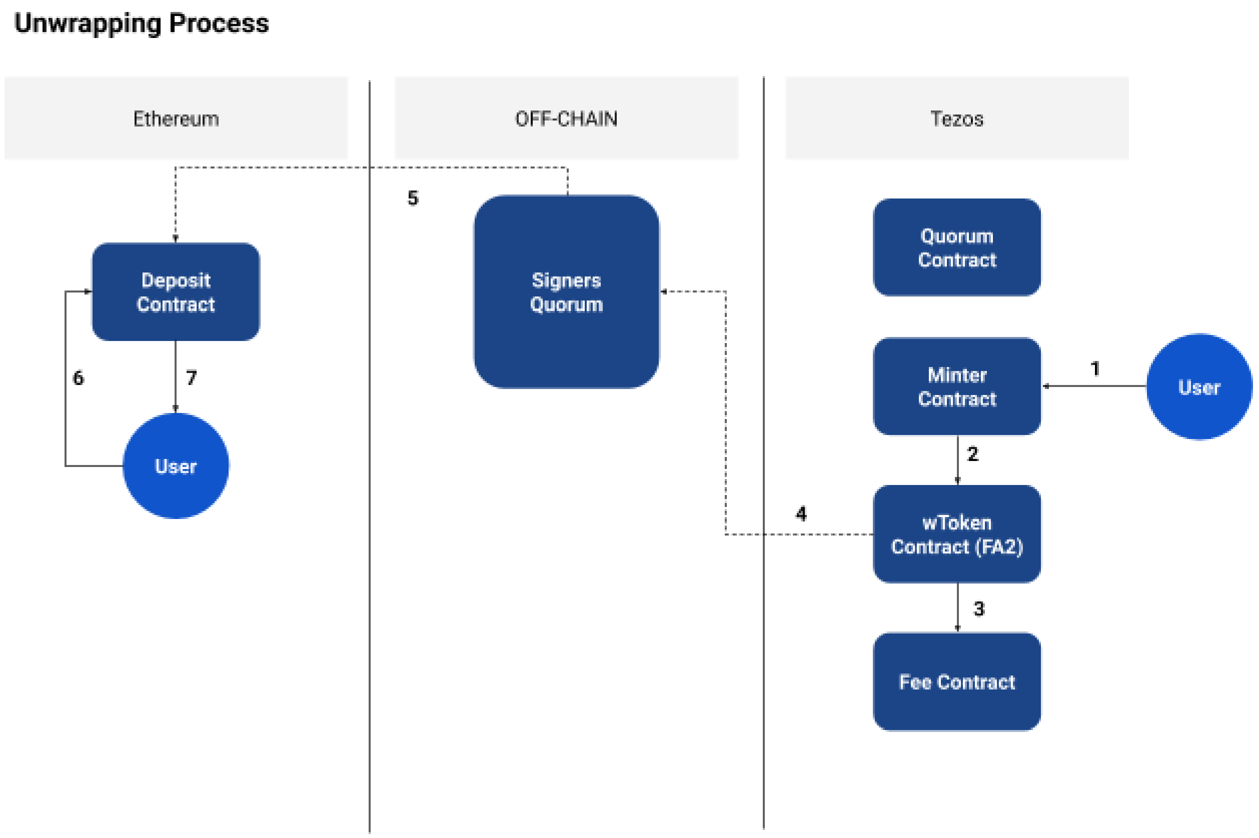
Wrapping Process



1) User allows Deposit contract to spend ABC tokens.
2) User wants to wrap N ABC (ERC20 token). User calls Deposit Contract to spend N ABC with a TEZ Destination Address. Deposit Contract transfers N ABC from User to itself.
3) Each signer then packages the ETH transaction information and corresponding minting instructions into a payload, signs it with their own cryptographic key and releases the signature on IPFS.
4) User gathers at least 3 different signatures and submits them along with an instruction to mint N wABC (wrapped ABC, FA2) to the Quorum contract.
5) If the three signatures correspond to 3 members of the Signers Quorum and match the minting instructions submitted by User, then Quorum Contract instructs Minter Contract to start the process of wrapping N ABC.
6) Minter contract sends two minting instructions to the wToken Contract (*mint xN wABC to Fee Contract and (1-x)N wABC to TEZ Destination Address*). .
7) (1-x)N wABC are allocated to the TEZ Destination Address.

8) xN wABC are allocated by the wToken Contract to the Fee Contract, which then dispatches it between participants of the protocol.

Unwrapping Process



1) User wants to unwrap N wABC. User sends a transaction to Minter Contract allowing them to burn (1-y)N wABC from User's address and pay yN wABC in unwrapping fees, along with an ETH Destination Address.
2) Minter Contract checks that User owns N wABC and that the unwrapping fee amount y specified by the User is larger than the minimum unwrapping fee. If so, Minter calls the burn method of wToken Contract (*burn N wABC from User's address)* as well as its mint method (*mint yN wABC to Fee Contract*)
3) wToken Contract burns N wABC from User's address, then mints yN wABC to Fee Contract
4) Signers watching the Minter Contract wait for a few confirmations after the transactions have been observed before starting the unwrapping process.

5) Each signer then packages the corresponding withdrawal transaction information (*send (1-y)N ABC to ETH Destination Address)*, signs it with their own private key and releases it on IPFS.
6) User gathers at least 3 different signatures and submits them to the Deposit contract.
7) If the three signatures correspond to 3 members of the Signers Quorum and match the unwrapping instructions submitted by User, then Deposit Contract releases (1-y)N ABC and sends them to the ETH Destination Address.

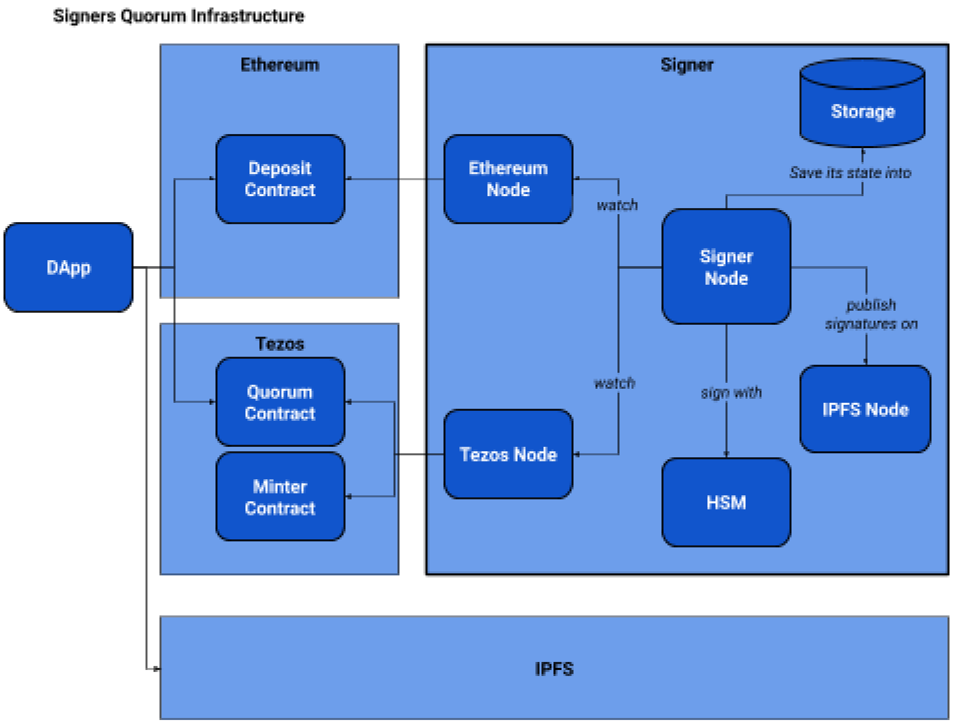
C. The Wrapping Quorum

1. Overview

The Signers Quorum guarantees the stability of the Wrap Protocol. The Quorum is a strong federation: its members cannot be trusted, but the federation itself can. Quorum Members interact with both the Ethereum and the Tezos blockchain, and create wrapping and unwrapping transactions based on user activity.
In its initial version, the quorum will be made up of 5 members of the Tezos community, including Bender Labs.

2. Setup

To be able to participate in the Quorum, signers to run a specific infrastructure made up of different components, as described below. This infrastructure has been thought and designed as being easy to spin up by entities or people already running Tezos and Ethereum nodes.



Ethereum and Tezos Nodes: to start with, members of the Signers Quorum need to run both an Ethereum and a Tezos Node to be able to participate in Wrap Protocol. The Ethereum node allows them to watch the deposit contract for assets waiting to be wrapped, and prepare transactions out of the same contract for unwrappings. By running a Tezos node, signers can interact with the Quorum Contract during wrapping, and watch the Minter Contract during unwrapping.

HSM / Private Key Storage Infrastructure: Each signer is identified by a set of private cryptographic keys. These keys identify them clearly as signers at several crucial parts of both the wrapping and unwrapping processes. More particularly, the Quorum Contract and the Deposit Contract take into input signatures that are made by keys held by the signers. Signers need to protect and use these keys at every wrapping or unwrapping transaction. To do so, they use an HSM infrastructure (running on cloud, local or in memory for testing purposes).

IPFS Node: As described in the paragraphs above, signers do not actually broadcast transactions on the Tezos or Ethereum blockchain. Instead, they sign wToken minting instructions (resp. original asset release transactions), then store them on IPFS for users to pick them up and present them to the Quorum Contract (resp. Deposit Contract). Signers are identified by users using IPNS. Signers can also use relay nodes.

Storage: The signer node has a minimal storage requirement which relies on plain files, and not an external RDMS[5], to ease installation and maintenance routines. The signer node mostly stores certain information such as the last block observed on Tezos and Ethereum, as well as the transactions which have already been minted. add

[5] Relational Database Management System

III. $WRAP Token and Wrap Economics

A. The $WRAP Token

The Wrap protocol is built around the $WRAP token. Both the governance and the economic profit generated by the protocol are distributed to $WRAP token holders. $WRAPs are distributed weekly to users of Wrap Protocol, with the idea of distributing its ownership to members of the Wrap community.

B. Governance: the $WRAP DAO

$WRAP holders form a DAO. This DAO has governance power over Wrap Protocol which can be split in 2 categories:

- **On-chain, automated governance**: the DAO can vote on-chain on parameters of the protocol and change them using BaseDAO. Ex: wrapping fees level for Quorum members
- **Off-chain, manual or consultative governance**: the DAO can vote off-chain on a certain number of topics using Snapshot-like tools. Ex: vote on new assets to add to Wrap.

<u>On-chain Governance</u>
The DAO can vote on:

- Wrap Protocol Fee Structure
  - Wrapping/Unwrapping Fees Level
  - Fee Recipient Addresses
  - Fee Split between Recipients
- $WRAP Token Characteristics
  - Max Supply
  - Distribution Function
  - Distribution frequency and date
  - $WRAP Recipient Addresses
  - $WRAP Split between Recipients

<u>Off-chain Governance</u>
Additionally, using similar tools as Snapshot (Tezos version implemented in BaseDAO), the community can vote (in a consultative or binding way) on pretty much any topic related to Wrap Protocol.

C. $WRAP Issuance and Distribution

1. Initial issuance

In order to allow $WRAP holders to use their tokens on both Ethereum and Tezos, $WRAP is an ERC20 token 100% wrapped into an FA2 token using Wrap Protocol.
The total supply of $WRAP is 100mm tokens initially. This can be modified by the $WRAP DAO at a later stage.

The token will be issued on Ethereum and locked on Tezos as follows:

1. Mint 100mm $WRAP as standard ERC20 tokens on the ETH blockchain
2. Wrap 100m $WRAP using Wrap Protocol and send them to the $WRAP Distribution Smart Contract on Tezos
3. The $WRAP Distributor contract then manages the distribution of $WRAP on the Tezos blockchain with the rules laid out below.

2. $WRAP Distribution

A batch of $WRAP tokens is distributed weekly. There are 3 categories of recipients of $WRAPS:

- Quorum Members
- Users of the protocol
- Dev Pool

Initially, each batch of token is split as follows:

- Quorum: 50%
- Users: 40%
- Dev Pool: 10%

The distribution split above can be modified by the $WRAP DAO. The list of recipients (addresses) can be modified by the $WRAP DAO.

The initial distribution schedule can be found in Appendix 1. The distribution increments follow a rebased exponential distribution. With the initial parameters chosen, the 100mm WRAP tokens are distributed after 7 years. These parameters (Total supply and distribution function) can be modified by the DAO.

D. $WRAP Token Economics

The token economics of $WRAP rely on the distribution of wrapping and unwrapping fees paid by users of the protocol. More precisely, fees are paid by every user of the protocol at every wrapping and unwrapping transaction. Fees are paid in wTokens only.

Initially, both the wrapping and the unwrapping fee will be set at 0.15%. This number will be modifiable by the $WRAP DAO.

This fee is shared as follows:

- Signers Quorum: 4bps
- $WRAP holders: 10bps
- Dev Pool: 1bp

The split (as well as the level) of fees can be modified by the $WRAP DAO.

For each fee recipient, there are conditions to respect to be eligible for fee sharing. Our approach is to make it easy to receive $WRAP tokens, but to ask for a commitment to benefit from the token economics.

IV. Roadmap

A. Wrap Protocol v1

Wrap Protocol v1 will launch in the first quarter of 2021 and will be immediately usable by anyone looking to wrap a set of ERC20 tokens to FA2 tokens.

The initial whitelist of assets available on Wrap Protocol will include (by alphabetical order): AAVE, BUSD, CEL, COMP, CRO, FTT, HT, HUSD, LEO, LINK, MKR, OKB, PAX, SUSHI, UNI, USDC, USDT, WBTC, WETH.

The Signers Quorum will initially be comprised of 5 members of the Tezos community, including Bender Labs. All wrapping and unwrapping transactions will be governed by a 3-of-5 governance.

$WRAP tokens will start to be issued to users and members of the Signers Quorum on a weekly basis. Each weekly distribution will be distributed as follows:

- Signers Quorum: 50%
- Wrap Protocol Users: 40%
- Dev Pool: 10%

Wrapping and unwrapping fees will both be set at 0.15% and will be paid by users in wTokens. Fees will be shared by Signers, $WRAP holders and the Dev Pool as follows:

- Signers: 0.04%
- $WRAP holders: 0.10%
- Dev Pool: 0.01%

While imperfect, Wrap Protocol v1 will serve as a fast, cheap and effective way to wrap the most liquid ERC20 tokens into FA2 tokens. Additionally, we will work with the Tezos ecosystem to integrate Wrap with existing DeFi applications so that their users can directly benefit from the additional liquidity and spectrum of assets that Wrap Protocol will unlock.

B. Wrap v2

Directly after the launch of Wrap Protocol v1, we will start implementing its next iteration.

The most important work on the v2 will be the decentralization of Signers Quorum and the strengthening of its consensus. By adding incentives such as rewards and bonds, we will further reinforce the strength of the federation - and lower the level of trust required by users. Additionally, we will create a process for adding new signers, and removing malicious members of the quorum.

The second important feature will be the implementation of a DAO of $WRAP token holders. Using the BaseDAO framework, we will allow $WRAP holders to vote on important parameters of the protocol, such as the level of fees and their attribution.

Last but not least, we will add more assets on the protocol, to bring even more liquidity to Tezos.

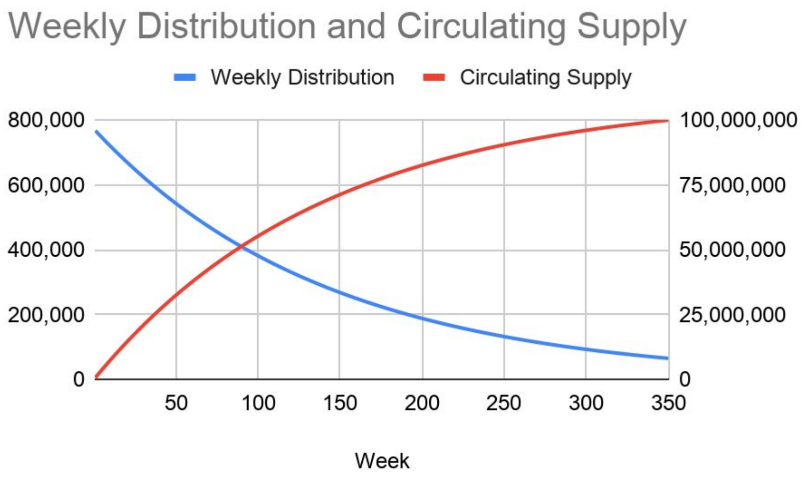
Conclusion

With Wrap Protocol, we've built a simple, yet effective way to use ERC20 tokens on Tezos. Wrap Protocol is the building block for Bender Labs' wider mission, which is to build an open financial system.

To conclude, the purpose of Wrap Protocol is to be used, owned and modified by its users and stakeholders. To echo the original Tezos whitepaper: we feel we've built an appealing seed protocol. However, Wrap's true potential lies in putting the stakeholders in charge of deciding on a protocol that they feel best serves them.

Appendix 1: $WRAP Distribution Schedule

Below is a description of the initial $WRAP distribution schedule. Weekly distributions and total supply can be modified by the $WRAP DAO.



Year	To Quorum	To Users	To Dev Pool
1	16,769,311	13,415,449	3,353,862
2	11,621,814	9,297,451	2,324,363
3	8,054,389	6,443,511	1,610,878
4	5,582,019	4,465,615	1,116,404
5	3,868,566	3,094,853	773,713
6	2,681,073	2,144,859	536,215
7	1,422,827	1,138,262	284,565
TOTAL	50,000,000	40,000,000	10,000,000