

-
-
-
- - [Hackathons](#)
 - [For organizations](#)

-
- [Portfolio](#)
-

- [Settings](#)
-

- [Log out](#)
-



-

- [Hackathons](#)
- [For organizations](#)

-



-

- [Back](#)
- [Back](#)
-
- [Portfolio](#)
-
- [Settings](#)
-
- [Log out](#)
-

Notifications

Loading...

StretchVM - The CBC Casper cross-shard messaging

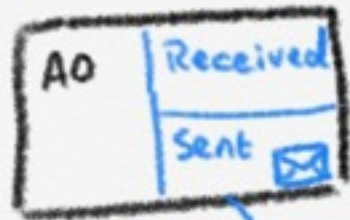
An Ethereum's Virtual Machine (EVM) that supports cross-shard messages (built into the block structure)

[Like](#)
[Comment](#)

- [Story](#)
- [Updates](#)

Stretch VM - Cross Shard Messaging for CBC Casper (Part of the Ghostly Trio, of Casper and friends)

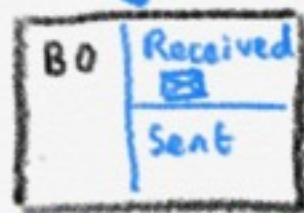
Shard A



Block A0

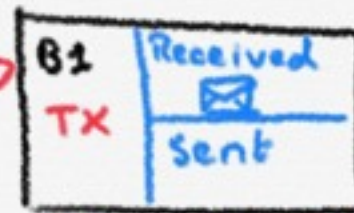
APPLYING TX:

Shard B



Block B0

TX



Block B1

•

Shard A/Block0 (A0) sends a message, shard B/Block0 (B0) receives it, and shard B/Block1 (B1) applies the transaction

- 1.
- 2.

Inspiration

While there have been a lot of research and discussions around how Ethereum could soon see its transaction processing capacity increase dramatically through the use of so-called 'sharding', we took a pragmatic approach to create, what we call, a Stretch VM (Stretch is part of the Ghostly Trio, with Casper and friends), which supports for the first time, cross-shard messaging, which is build into the block structure.

What it does

It is a fully compliant Ethereum Virtual Machine, that allows for the first time the ability to send messages between shards (using the shard_id), as part of the block structure.

Specifically, we added a new VM type, called StretchVM. Stretch VM that has cross shard messages (XMessages) built into the block structure... so that it's super easy to send and/or receive such messages across shards.

We are able to demonstrate 2 types of X-Shard transactions:

1. An ETH transfer, and,
2. A contract call.

Stretch VM - Cross Shard Messaging for CBC Casper

How we built it (Part of the Ghostly Trio, of Casper and friends)

From scratch, we implemented the CBC Casper sharding specification and defines 3 new block fields:

- Sent XMessage Log Root: The root of the sent XMessage trie, which are sent in this block
- Received XMessage Log Root: The root of the received XMessage trie, which are received in this block
- ShardID: An int identifier for which the shard to which the block belongs

Any transaction that has a destination address in another shard is processed as a sent XMessage. A sent XMessage is seen by the receiving shard in the Sent XMessage Logs of that blocks, and then is inputted as a received XMessage in their shard. This will be processed, and the appropriate transaction will be applied in the receiver shard while also putting the XMessage in the Received XMessage Logs.

The X-Shard contract calls work seamlessly, just like any other transaction. This is due to the transaction data being transmitted as-is through the sent XMessage into the received XMessage, and finally into a transaction applied on the destination shard.

Challenges we ran into

1. Lack of sleep
2. It took a while to add a custom BlockHeader in StretchVM.
3. We needed to make our own header called StretchBlockHeader with some new fields such as Sent XMessage Log Root, Received XMessage Log Root and Shard ID.
4. In order to incorporate our changes we had to take a deep dive into the codebase and modify the Chain, VM and the RLP serialization and encoding for the XMessages.

Accomplishments that we're proud of

We were able to extend an existing Ethereum VM, to support the ability to exchange (send and receive) messages across shards, as a new transaction type, in a way that will allow us and others to implement both CBC Casper as well other protocols, while keeping the same workflow and lifecycle of Ethereum transaction processing.

What we learned

We learned the depth and details about the full lifecycle of Ethereum transactions, but now, from within the EVM. We spent a considerable amount of time on making sure that our implementation is built as a "first class citizen" of the VM, by leveraging the hierarchy of Chains, VMs, their coordination, as well as their State.

It took us multiple iterations to get the design right - and now we believe that this layer can be used out of the box, both for our research, for follow up implementations as well as other protocols that will leverage sharding, at scale.

What's next for StretchVM - The CBC Casper EVM

Let's begin from the end: **An Upstream Pull Request back to the py-evm project**, as well as:

1. **CBC Casper client** This implementation is built to implement the EVM for the Casper CBC sharding proposal. We will use it to implement a CBC Casper client (after a suitable network stack is built), and to run network emulation tests to determine the performance of the sharding proposal!
2. **Book-keeping Manager** There are a lot of discussions about automatic sharding and ongoing sharding optimization algorithms. We would like to create an automated tool that will let users submit normal transaction as XMessages, by filling in the bookkeeping information involved in these XMessages
3. **Optimized Serialization** While this was done very quickly through a hackathon, there are probably ways to construct (and serialize) the XMessages better, by choosing an optimal serialization encoding, and structure.
4. **Research, Research, Research!** Apart from this implementation, there is research to be done to make the specification better (& more complete). For example, coming up with (& implementing) a light client protocol for this proposal!
5. **An Upstream Pull Request (PR) back to the EVM project** Since we worked very hard on the design this solution, not just as a one-off hack, but as work that will be contributed back to the community, the extended project (of py-evm) so that others can use it - seamlessly.

Does this actually work? Can I see it?

Yes and Yes!

Please clone <https://github.com/cbc-casper/py-evm> by executing 'git clone <https://github.com/cbc-casper/py-evm>', and run:

```
python3 tests/cbc/test.py
```

Built With

- ethereum's-evm
- py-evm

Try it out

- [GitHub Repo](#)

[Edit project](#)

Submitted to



About a minute to submit
[Edit hackathon submission](#)


Created by

- [edit](#)

Describe your contribution

Save

[Cancel](#)

E.g., I worked on the backend and cleaned up our landing 



[Jonathan Levi](#)



-

[Barnabé Monnot](#)





[Aditya Asgaonkar](#)



Researcher @ Casper CBC



[Shreyas Bhat](#)



[+ add team members](#)

[Report inappropriate content](#)
[Hide project from my portfolio](#)
[Like](#)

Share this project:





×

Updates



Post an update about StretchVM - The CBC Casper cross-shard messaging

Keep a log of how StretchVM - The CBC Casper cross-shard messaging has evolved. Post about new features, app store releases, screenshots, or even code snippets. Your followers will see your updates in their feeds and can comment on them.

Use markdown for formatting.

Post update

- [Turn off notifications](#)



[Jonathan Levi](#) started this project — [about 2 hours ago](#)

Leave feedback in the comments!

Comments



Write a comment

Post comment

Devpost

- [About](#)
- [Careers](#)
- [Contact](#)
- [Help](#)

Jobs

- [Employers](#)
- [Pricing](#)
- [Job advice](#)

Hackathons

- [Browse hackathons](#)
- [Explore projects](#)
- [Host an in-person hackathon](#)
- [Host an online hackathon](#)
- [Hackathon best practices](#)

Connect

-
-
-

Legal

- [Privacy policy](#)
- [Community Guidelines](#)
- [Terms of service](#)