The Crosschain Layered Architecture is being implemented in the GPACT repo. Standardisation discussions around the interfaces between layers are happening in the Enterprise Ethereum Alliance's Crosschain Interoperability Working Group. This is all work in progress: please contribute your ideas!

Crosschain bridges allow value and information to be communicated across blockchains, sidechains, rollups, and other blockchain-like constructs. There are a lot of techniques that have been proposed and implemented. Invariably, the techniques have a methodology for trusting information from the remote blockchain, a technique for executing functions in the blockchain, and an application. Due to the lack of interoperability, it is usual for each organisation to recreate the whole protocol stack, and stand up an entire set of infrastructure to operate the bridge.

The diagram below shows a Crosschain Layered Architecture. My hope is that this layered architecture will allow technology and infrastructure to be more easily reused and will allow for greater interoperability.

[

crosschain-protocol-layers

2250×872 163 KB

](https://ethresear.ch/uploads/default/original/2X/4/42ae7beea373191b8b6251597da3979216a1d481.png)

Crosschain Protocol Layers

The Crosschain Message Verification

layer ensures that information can be verified as having come from a certain blockchain. Contracts on a source blockchain emit an Ethereum Event (the information). The Crosschain Message Verification layer software communicates this Ethereum Event to the destination blockchain in a way that the Ethereum Event can be trusted. This could be using threshold signing, staking and slashing, optimistic approaches, block header transfer, or a trustless approach.

The Crosschain Function Calls

layer executes function calls across blockchains. The technique could be a non-atomic approach that calls a function on a remote blockchain based on calls on a source blockchain, an atomic crosschain function call approach such as General Purpose Atomic Crosschain Transaction (GPACT) protocol [1][2][3][4] that provides a synchronous, composable programming model and atomic updates across blockchains, or some other protocol, for instance a crosschain function call protocol that focuses on privacy.

The Crosschain Applications

layer consists of applications that operate across blockchains, and helper modules that allow complex applications to be created more easily.

Advantages of Crosschain Layered Architecture

Having a layered architecture with clearly defined interfaces between layers has the following advantages:

- Different organisations can create Crosschain Message Verification, Crosschain Function Call technology, and Crosschain Applications that will be interoperable.
- A different Crosschain Message Verification technique could be used for each blockchain / roll-up used in an overall crosschain function call.
- Different Crosschain Function Call techniques can share the same deployed Crosschain Message Verification infrastructure. For example, within the one application, some low value transactions could use a non-atomic mechanism and other high value transactions could use GPACT. Additionally, a company could offer a Crosschain Message Verification bridge and allow any Crosschain Function Call approach to be used on top of the bridge by any organisation.
- Organisations can focus on creating solutions for a single part of the protocol stack stack. That is, rather than having to create the entire stack, a company might choose to focus on a better Crosschain Function Call approach.
- It should drive experimentation.

Example Non-Atomic Call Flow

The diagram below shows a possible call flow for a non-atomic crosschain function call.

[

nonatomic-call

1032×482 65 KB

](https://ethresear.ch/uploads/default/original/2X/7/7f00df349be305f5c089c01e191f8450dbeb9be9.png)

The steps are:

1. The user submits a transaction that calls Business Logic contract on Blockchain A.
2. To execute a crosschain function call, the Business Logic contract calls the Crosschain Control contract.
3. The Crosschain Control contract emits an event indicating the blockchain, contract address, and function and parameters to be called on the destination blockchain.
4. Using some Crosschain Message Verification technique, the Crosschain Control contract on Blockchain B is called with the event from the previous step. For example, the event could be signed.
5. The event is passed to the Crosschain Message Verification layer, that checks that the event can be trusted.
6. The Crosschain Control contract executes the function call, calling a function on the Business Logic contract on Blockchain B.

Example GPACT Call Flow

GPACT protocol is a protocol that allows a call execution tree of function calls to be executed across blockchains. The call execution tree is committed to by executing the start function on the root blockchain of the call execution tree. Segments of the call execution tree are then executed on various blockchains, with storage updates being stored as provisional updates. The root function is called on the root blockchain to execute the entry point function of the call execution tree. Based on whether the root function and all segments execute correctly, the root function emits an event indicating that provisional updates should be committed or discarded across blockchains. The updates are finalised on all chains by calling signalling functions on the Crosschain Control contracts on each blockchain. See [1][2][3][4] for more information on GPACT.

[

gpact-call

1048×482 71.3 KB

](https://ethresear.ch/uploads/default/original/2X/5/503e5ecb8ad98b7b87ada9337da0301db828ec9d.png)

The steps for GPACT are:

1. The start, segment, root, or signalling function is called on the Crosschain Control contract. The function takes as parameters events and signatures or proofs proving the events are valid.
2. The Crosschain Control Contract asks the Crosschain Message Verification layer to check the signatures / proofs. A different mechanism could be used for each source blockchain / each event.
3. The function in the business logic contract indicated by the call execution tree is called.
4. This could in turn call other contracts.
5. If there are any updates to lockable storage, then the updates are stored provisionally and the Crosschain Control contract is informed so that it knows that the contract will need unlocking.
6. A business logic contract can execute a crosschain function call. The Crosschain Control contract can check that the call to the other blockchain is being called with the expected parameters.

I will be giving a talk on an ERC 20 bridge implementation on top of GPACT protocol at the Ethereum Engineering Group meet-up on Oct 13, 2021 (Brisbane, Australia time zone). Register here. If you missed it, the talk will be on YouTube after the talk here.

Standardisation

The Enterprise Ethereum Alliance's Crosschain Interoperability Working Group is actively working on standardising the interfaces between layers of the Crosschain Layered Architecture. To get involved join the Crosschain Interoperability

Working Group, and join the EEA.

Get Involved

Please have a look at the interfaces in the GPACT repo. Consider contributing a component: a crosschain function call approach, and message verification approach, an application helper function or an example application. Comments and ideas are welcome. Please message me here on ethresear.ch or on LinkedIn.