

## IOHK | BLOG

# Team Grothendieck move closer to Ethereum Classic goal

Working on the code in Argentina

🕒 MAY 22, 2017    📖 4 MIN READ



It took a little longer than expected but I finally made the trip to Buenos Aires, Argentina. In fact, I'm standing at a work desk by the window in Frankfurt airport waiting for my flight back to Dublin. I'm enjoying a gloriously sunny day here through the wall of glass.

It was another productive trip, a lot has happened since the Team Grothendieck trips to [Poland and St Petersburg](#). In our work to build a Scala client for Ethereum Classic there has been a lot of code written, a lot of understanding gained, and a couple of milestones reached: we now have the ability to download and execute blocks of transactions from the ETC chain. We have also evolved a lot as a team.

The remaining milestones to reach include mining, and the JSON API – to allow Mist and other dapp wallets to use our client. In parallel with that we need to focus on our codebase. It was this process that the Grothendieck team's [Alan Verbner](#) and [Nico Tallar](#), and I spent our time on in BA.



As a background, in an ideal world we would create code from day one supporting the coupling that made sense as we approached the release. However, this is an almost impossible task because we can't usually know the most sensible "final" coupling when starting out. For the ETC client we took the (oft used) approach that we would write clean unit tested code that implemented the functionality we understood at the time and then refactor as we learnt more. For example, when we finished the block download phase we had very little in the way of model classes for 'blockchain'. However, as we spun up the "Tx Execution" phase, we discovered it made a lot of sense to create a set of functions coalescing around a 'blockchain' model.

There's a school of thought that says this is the way to carry on: don't waste your time building "reusable" components that aren't reusable and won't be reused. I have sympathy for this approach because building reusable components is hard and it is embarrassing for your new component – the one you spent time and effort on – to fail at the first attempt at reuse because it doesn't *quite* do what you need it to. Better to allow the new functionality to drive refactors as and when it comes. There's a humility to this approach that appeals to me.

Guess what's coming next? We're going to look at ways to modularise our client. Why? Firstly and most importantly it's a functional requirement for the codebase to support a significant level of flexibility. Four things that might define the core of a blockchain client are the network module, the ledger, the consensus mechanism and the wallet. Closely coupling the wallet and ledger together, we would like to experiment with different types of ledger and different types of consensus. And these should be able to use a well defined network module.

So we will first attempt to isolate the 'network' module. This is a module that maintains connections to peers and sends and receives a configurable set of messages. It allows messages to be addressed to a peer or broadcast to many peers. It allows clients of the module to register for types of messages and types of message per peer. It's also functionality that we have already created. We just need to organize it so that it's reusable!

Why now? The JSON RPC API – in theory – should be controller layer code. The mining integration should – also in theory – not affect the workings of the network module. So the functionality to be reused should

already exist and when we repackage it without breaking the existing system we know it's useful. By the time we get around to examining the coupling of the ledger and consensus the same should be true, we won't be making up use cases for invented modules, we will have specific working code to repackage. Will we produce interfaces and coupling that can be reused? That's the challenge. And after that – optimization of the internals...

Mining, web API and modularisation are not trivial tasks but they will end. And with them we reach the end of the existing road map – stability, bug fixes and auditing aside. For the past five months we have been playing catch up, we didn't need to talk about future evolution of the technology because we had a clear and challenging mandate – to recreate a client from the ground up. Now that we're relatively close to doing that, the exciting process of talking about the future of the codebase can begin.

While in BA, Sergio Lerner kindly hosted the three of us at his office and over decent coffee and [alfojores](#) we had a good discussion about Ethereum tech, and some of the things he's been up to. And of course, RSK's upcoming release of their platform at Consensus 2017 in NYC. (Best of luck RSK!)

I'm always interested in how a global blockchain aimed at general purpose use can scale, with no way to delete defunct contracts from the global state trie. Sergio made the interesting point that ETC probably won't need to scale for a couple of years. He also suggested that with storage being so cheap for a network in a steady state (with most nodes staying up to date) it would be more expensive for all nodes to delete a contract than to keep it.

Apart from Rootstock, the [Bitcoin Embassy](#) in Buenos Aires, where Alan and Nico, normally work is littered with interesting people working on multiple ways of leveraging the Bitcoin blockchain. There's a great atmosphere in the building, calm and friendly but industrious and I really enjoyed my time there, so when it was suggested we attend a [live podcast](#)...we said yes!

A special shout out to Alan Verbner, a man who is proud of his city and I think the city can be proud of him. We walked the city the whole weekend and I got a real sense of it. BA is modern but you don't have to look too hard to find old world charm – French restaurants with dark wood and marble counters, majestic old cafes full of Art Deco gold fittings and the smell of cardamon infused coffee. And then there's the steak. Vegetarians, look away now. I'm delighted to report that Argentina's reputation for steak is well deserved. The variety of cuts, the sauces, the cooking...it might be worth going back just to eat steak.