**Web 3 and the Reach for a Better Tomorrow Bounty Hack**

"Programming in Reach is really easy!" enthused Karen "We'll teach it to you."

I was at a meetup of the All Things Blockchain group in Brisbane, dipping my toes into the realm of blockchain development. I wanted to find out more about how blockchain technology actually works. Ideally that meant learning how to program it.

Karen Cohen, founder of AlgoHub, the Australian centre for the Algorand cryptocurrency, was there with a team to promote the Australian Reach Bounty Hack - one of a series of hackathons set up by the Reach Foundation to disseminate the Reach programming language among budding developers.

Chris Swenor created Reach with the aim of making blockchain programming easy, safe, and widely accessible, not just the preserve of a technological elite. The more developers and the more diverse the developers, the greater the likelihood of cool new ideas emerging, possibly from unexpected quarters. It’s a numbers game. The jackpot would be a "killer app" taking blockchain technology truly into the mainstream.

Well, I thought, if a non-standard background is a plus then this is definitely for me - a retired female aged 60+ with a background in scientific but not commercial programming.

I found myself musing about the role of consensus, not just in blockchains but in our lives. Consensus underpins our societal definition of what constitutes “reality” and, crucially, what doesn’t.. Consensus is expensive for us too. We can invest an inordinate amount of brain power checking and verifying that we see what others see. The use of consensus in blockchain technology could even be regarded as a special-case extension of the role of consensus in our normal perceptions and interactions….

.

...Enough of philosophy! Encouraged by Karen's enthusiasm and Chris' presentations I signed up for the Bounty Hack, the objective of which was to create and deploy a functioning application on the Algorand blockchain within a six-week time frame. From first perusal of the online documentation I realised I was in for a steep learning curve. Reach might indeed be easy for full-stack Web developers familiar with Javascript and React. The rest of us have to get to grips with those. There's also Docker - a mysterious black box where the executable images live. I installed it blindly, resolving to look into it later. All of this operates within a Linux environment, that at least well within my comfort zone.

The Reach site features a tutorial on how to deploy a Rock Paper Scissors game on the any of the Algorand, Etherium or Conflux networks. Our first task was to complete that tutorial using the Algorand option. We were helped along by Zoom sessions with the experts. The Reach Discord was available for posting questions. I stumbled enthusiastically through the tutorial and made a tentative start on my own project.

As I began to branch away from the tutorial examples I realised that the apparently simple Reach language is quite constrained by the complicated infrastructure beneath it. Further constraints are added by Reach’s automatic verification engine. Of course these are the very properties that ensure that Reach is easy, accessible and safe to use. Knowing that did little to mitigate an almost physical sensation of pushing at immovable rocks. Reach adamantly refused to be wangled or finangled into doing what I was trying to make it do. So I geared myself up to dive deeper, ask the experts and hopefully learn more about using the language as intended ….

...At this point my crypto other-world was rudely interrupted by real life! There were torrents of real rain falling from the real sky as the real creek rose to flood the lower level of our real house. The next day our real kidults were made homeless when the duplex they were renting from us was also flooded. The ensuing merry-go-round of mud cleanup, insurance quagmires and helping the kids house-hunt left me with little head space for much else. The Australian Reach Bounty Hack began to drift away like the flood waters.

However the hack deadlines got pushed back and, working within my new limits, I put aside my solo project to join the Laughing Otter team with whom I already shared mentor Paul Ogwulumba. I was assigned to do the artwork, one of my hobbies, as the team already had two developers with much better Javascript and React credentials than mine.

The Laughing Otter's Music challenge was a wide-ranging project seeking to reward participants with tokens or crypto on completion of a set of music-related tasks devised by Jeff, our team leader. I set to work on the web graphics for each task page. Connie created NFTs and Laughing Otter tokens. Rasmus set up a GitHub account and a Jira board. Jeff brimmed with cheery optimism... But as it turned out, the other Laughing Otter crew were not immune to real life issues either. Jeff caught Covid. Connie and Rasmus had to prioritise their University coursework as their academic year got into full swing. Soon we were less than two weeks out from the deadline…..where was the Reach code?… At least in my case the flood issues had been stabilised if not resolved. Guided by the tutorial I was able to string together a skeleton in Reach and React which could deploy a smart contract and take a participant through the music tasks outlined by Jeff and displayed in my web view pages.

That in the end was what we went with. Rasmus and I were still fixing bugs up to and during the submission. But that's how real life happens, and it was fun. I was glad I'd eventually got to do some (admittedly crappy) coding, even if that failed to make it into Jeff's presentation in which I was the artist and we'd all performed superbly within our assigned roles.

Through its ups and downs the Australian Bounty Hack turned out to be a worthwhile experience. I learned a great deal. My journey with Reach programming will continue. Meanwhile the Reach documentation is being steadily improved and augmented and there are additional mini-hacks to keep us active and engaged.

I agree with Chris Swenor’s view that blockchain development should be easy and widespread. I created my first web page in 1994 at the dawn of Web 1 when I was a particle physics PhD student. Early Web 1 was decentralised and wide open for sharing information and even for fun stuff. We didn’t see commercial prospects. Network speeds ranged from bytes to kilobytes per second. But it took off explosively because it was so simple to program, versatile and readily accessible. The infrastructure grew to keep pace.

If the blockchain can indeed provide a technological extension of our innate wish for consensus then Web 3 could take hold in a powerful way if appropriate use cases emerge.

That’s not without danger. Consensus can exclude as well as include, and technology itself is amoral. Such dangers could be minimised if Web 3 were to be accessible, versatile and programmable by as wide a demographic as possible. Reach is a good step in this direction. Hopefully there will be further steps and breakthrough developments.

But watch out, there are sharks about!