This started as an email to Jon Charbonneau after I read his great blog post “[Rollups Are L1s (& L2s) a.k.a. How Rollups Actually Actually Actually

Work](https://dba.mirror.xyz/LYUb_Y2huJhNUw_z8ltqui2d6KY8Fc3t_cnSE9rDL_o)”. By the end it turned out to be waaay to long for an email and I thought it might be interesting for other people too. So I decided to instead post it here.

TLDR: My name is Bruno França, I’m the consensus lead at Matter Labs and I just wanted to share my opinions

with you on what is and is not a rollup.

With that out of the way, I’ve been following your posts for a few months. Sometimes I don’t agree with your conclusions (for example, on shared sequencers), but your posts always have high-quality research and are quite unbiased so I normally learn something new. Your last post on what rollups are was no exception, the difference here is that it actually forced me to think for a while on what is a rollup and I think my ramblings might be useful for you.

To start, I’ll say on what I agree with you. I do agree that:

1. There are two different concepts here. I’ll call them the “rollup bridge” and the “rollup blockchain” temporarily.
2. That a rollup bridge and a rollup chain are separate and can “fork” away from each other.
3. That L1 and L2 are relative terms and depend on where a given asset was issued / is native to.

Let’s start with a thought experiment. Imagine we have two separate existing blockchains, let’s say Ethereum and Near. Can we build a trustless bi-directional bridge between them without modifying their protocols? Yes, we can! First, imagine we develop zk validity proofs for Near’s VM and consensus (all my examples are going to be about zk rollups, it’s just simpler for me). This clearly doesn’t require any change to Near’s protocol, we simply find a way of proving that a given block is a valid state transition inside a SNARK/STARK. Now, we also create on Ethereum a smart contract to verify those proofs and we’re done.

Now anyone can submit Near blocks (or state deltas) to Ethereum together with validity proofs and that smart contract will know the state of Near’s blockchain without any trust assumptions. And then it’s trivial to send messages from Near to Ethereum, you just need to submit Merkle proofs of Near’s state to the smart contract in Ethereum, that way you can prove any part of Near’s state. That’s the first realization. We are not sending coins or data from one blockchain to another blockchain, there’s no movement of anything, these are all still independent databases. We are just proving the state of one blockchain in another blockchain. Moving coins across blockchains is just a convenient abstraction.

But so far this is uni-directional, we only prove Near’s state on Ethereum. How do we do the reverse? Simple, we develop validity proofs for Ethereum, create a verifier contract on Near and send Ethereum blocks and validity proofs to Near. Now we can prove Ethereum’s state in Near. With two uni-directional trustless bridges we get a bi-directional one. This however ponders the question, why do we need two bridges? Rollups don’t have two bridges, Arbitrum and zkSync don’t have Ethereum bridges in their states, what’s happening here? Well, they have, but it’s a different type of bridge. We force zkSync full nodes to also be Ethereum full nodes, the same happens with Arbitrum full nodes. That’s the second realization, all rollups actually have two uni-directional bridges. It’s just that on the zkSync → Ethereum direction we use validity proofs + state deltas while on the Ethereum → zkSync direction we just use a full node bridge. Evidently a possible solution to connect Ethereum and Near would be to require each Ethereum full node to also be a Near full node and vice-versa. This would in fact achieve the same thing, it’s just a little dumb because it doesn’t scale well.

But now we have two bridges with validity proofs between Ethereum and Near. We can prove one blockchain state on the other blockchain and use that to “move assets” and “send messages”. What I want to point out is that we didn’t change any protocol, we didn’t require any extra functionality from Ethereum’s or Near’s full nodes, each blockchain might not even be aware of these bridges. A completely foreign third-party like Matter Labs or Coinbase could maintain (and eventually fail to maintain) these bridges. So, are Ethereum and Near now rollups? Is Near more secure now because its blocks are posted on Ethereum? Will either blockchain’s security decrease if the bridges stop working? I think that you would agree that no, nothing changed on either blockchain. It’s just that now there’s a smart contract on each blockchain that can access the state on the other blockchain. Note that these bridges are also independent, if one of them fails, the other is completely unaffected.

Now we can talk about different types bridges then. Patrick McCorry has abrilliant post on rollups being validating bridges. I can say that I’m generally on Patrick’s camp, but I think there’s actually more types of bridges:

- The first type is obviously “full node bridges”. These are simply when the full nodes of one blockchain are also full nodes of another blockchain. This is the highest level of security for a bridge, this is what a trustless bridge actually is. The bridge has no extra security assumptions. It is also completely unworkable at scale. In the Ethereum + Near thought experiment, if we require all full nodes to be full nodes of both blockchains, then there’s no point in having two different blockchains. It can make sense in one direction though, if the node requirements for one blockchain are much smaller than for the other blockchain. That’s exactly what happens in rollups like zkSync, Arbitrum, Optimism, etc.
- The second type are “validity bridges”. This is a bridge that uses zk proofs to prove the validity of a given state transition (i.e. zk rollups). We have extra trust assumptions related to the proof system used for those validity proofs.

- The third type are “optimistic bridges”. The type of bridge used in optimistic rollups. The extra trust assumptions here are related to game theory and to the existence of at least one honest full node of the optimistic rollup.

Both validity bridges and optimistic bridges aim to approximate the security of full node bridges without requiring the same level of resources. In effect, a validity or optimistic bridge pretty much acts like a full node of another blockchain, but it’s just a smart contract.

Then of course you have light-client bridges (like Near’s Rainbow bridge), multisig bridges and so on. Now we can actually use this model to classify different rollup projects. For example, zkSync Era is a centralized (i.e. single validator) blockchain with a full node bridge from

Ethereum and a validity bridge to

Ethereum. And that’s what most rollups today are: blockchains with a full node bridge from the base chain and a validity or optimistic bridge to that same base chain. Note that this creates a kind of hierarchy between the chains, Ethereum full nodes won’t become full nodes of whatever rollup decides to bridge to it, but the rollup full nodes are pretty much forced to also be Ethereum full nodes. So there is maybe some value in the L1 and L2 terminology, even though I agree that for a given asset L1 and L2 are relative (we probably need better naming for these concepts).

Finally note that the rollup blockchain doesn’t inherit the technical security of the parent chain. In other words, posting the data (and proofs) from some blockchain to Ethereum will not increase that blockchain’s security. The usage of Ethereum’s data availability is solely for the benefit of the rollup bridge, not of the rollup blockchain.

That implies then that the rollup bridge includes the smart contract and the data availability on Ethereum, and that the rollup blockchain is, well, just a blockchain.

This model, as neat as it seems right now, doesn’t explain validiums and sovereign rollups. Are validiums bridges or blockchains? How is it different from validity bridges? How about sovereign rollups? They don’t even have bridges! Let’s start with sovereign rollups since they are simpler to analyze.

The way sovereign rollups are normally described is as a rollups without a bridge. They essentially use another blockchain’s data availability and consensus as their own and so inherit that blockchain’s security. They do so by posting all their data to a base chain and the sovereign rollup full nodes are just full nodes of the base chain that have extra rules to interpret that data.

So is this a rollup? No, it’s not. It might sound like a rollup at first, but a lot of other things also meet this definition. Things that we certainly don’t classify as rollups, for example, Ordinals in Bitcoin. All the data for the Ordinals blockchain is on the Bitcoin blockchain. And for you to be a full node of the Ordinals blockchain you just need a Bitcoin full node and to know the Ordinals rules. It also has exactly the same security as Bitcoin. The same can be said about many other protocols on top of Bitcoin like Omni, Counterparty, Mastercoin, etc. Are all these protocols sovereign rollups?

I could probably find more examples, but the main point is that a blockchain piggybacking on another blockchain’s consensus and/or data availability is nothing new. As far as I know there’s no common term for these constructions, so I’ll try the name “dependent blockchains”. The innovation with rollups was creating a way for two blockchains to communicate in a trust-minimized and efficient way. Sovereign rollups lack that and so shouldn’t be called rollups, they are simply dependent blockchains.

This now fits nicely into our overall model. Blockchains are independent if they have their own consensus and data availability, and dependent if they rely on another chain’s consensus or data availability. Separately, they might or not have bridges to and/or from another blockchain. A sovereign rollup then is just a dependent blockchain without any bridges to

the base chain. But sovereign rollups, by definition, do have a full node bridge from

the base chain, since full nodes of a dependent blockchain need to be full nodes of the base chain.

Finally, we get to validiums. Just like rollups, they are blockchains with a full node bridge from

the base chain and some

bridge to

the base chain. That new type of bridge is basically a validity bridge where we don’t post the blockchain data (either inputs or state deltas) to the base chain, we only post the zk proofs. For lack of a better name let’s call them partial validity bridges. But how is this different from a normal rollup? To find that out we need to understand what happens when blockchains fail and bridges fork.

Let’s go back to our thought experiment on bridging Ethereum and Near. In this situation, what happens if the Near blockchain halts (let’s not worry about how that happens, just imagine that Near completely stopped producing blocks)? Evidently, the bridge from Near to Ethereum will stop being updated. If there are coins on that bridge, they are stuck until the Near chain resumes. Evidently, the assets on that bridge don’t seem to have the same security as the Ethereum blockchain. That goes against what is normally claimed about the security of rollups, so what’s happening here? Remember

that both validity and optimistic bridges act similarly to full nodes, and full nodes can fork a blockchain. Most rollups have some “escape hatch” mechanism planned, which really is just an automated fork mechanism. In our example, if the Near blockchain fails, the bridge could change into a based rollup, allowing anyone to update the state of the bridge, as long as it comes accompanied by a validity proof. If the Near blockchain then comes back online, it would have a different state from the bridge, thus reinforcing the idea that the bridge has indeed forked away from Near. This bridge would effectively have the same security as Ethereum, but it is crucial that there is some forking mechanism planned into the bridge. Imagine other situation where both Ethereum and Near blockchain are working but only Coinbase is allowed to update the state of the bridge (because that’s how the bridge was designed). If Coinbase fails for some reason and there is no forking mechanism in the bridge, then the bridge will just halt and all the assets will become stuck, even though the Near blockchain is still live. By now it should be pretty clear that bridges and blockchains really are different entities and that the security of one doesn’t influence the security of the other.

Now we can easily see the difference between validity bridges and partial validity bridges (i.e. validiums). Validity bridges are always guaranteed to have the state data (which is of course necessary to create a fork) because they post all state updates to the base chain. Partial validity bridges might not have that state data, it instead requires a honest minority of the validators in the validium blockchain to guarantee the availability of that data.

To summarize this very long post, it is blockchains and bridges all the way down

. There’s many different types of bridges, but the ones that are more interesting for the L2 space are: full node bridges, optimistic bridges, validity bridges and partial validity bridges. We can also classify blockchains into two different types, dependent or independent, depending if they use another blockchain’s consensus and data availability as their own. These are separate concepts though and we can pair any type of blockchain with pretty much any number and type of bridges.