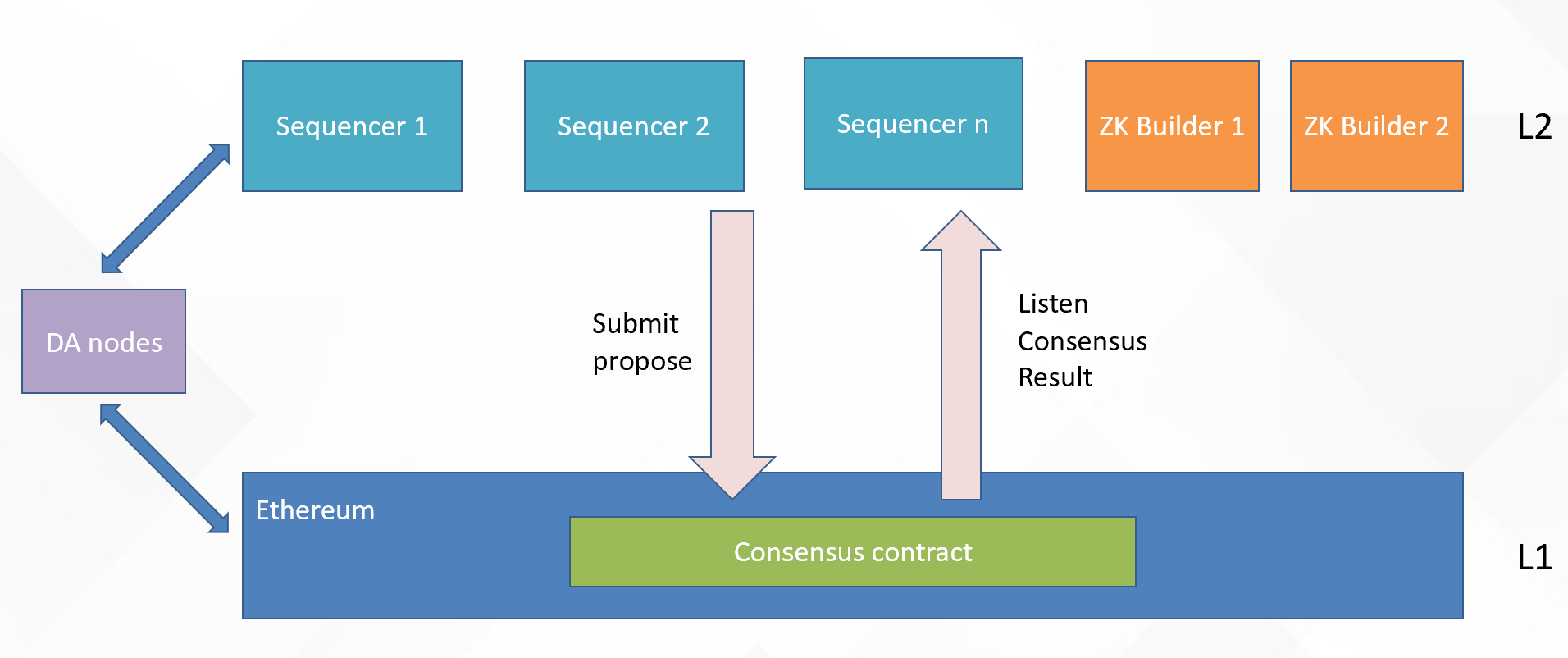
# Decentralizing ZK RollUp

# Processes



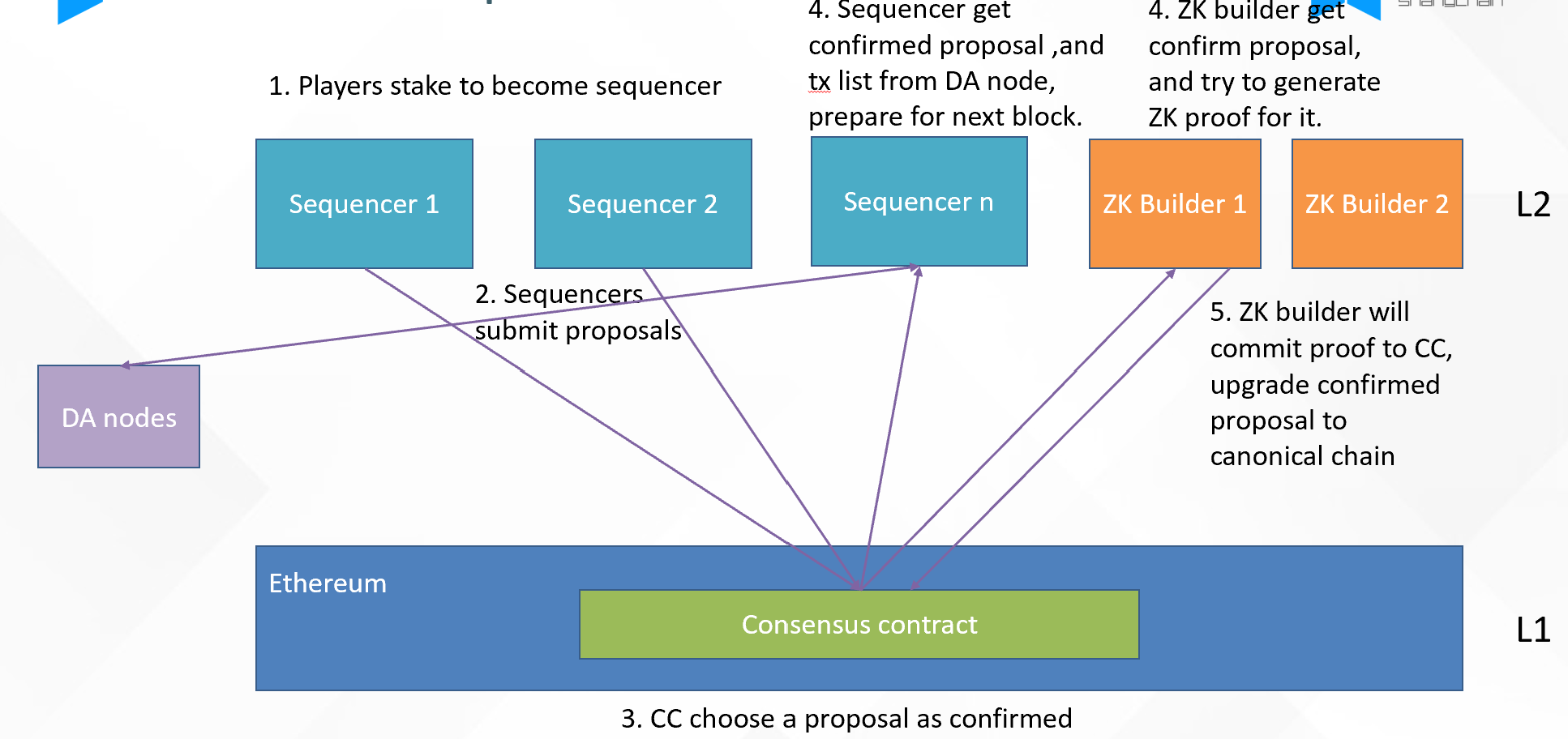
On L1, we have “consensus contract”, and on L2, we have sequencers and ZK builders. DA nodes supply reliable data access , L1 and L2 can share data via it. Sequencers need to stake on L1 to mine.

## 1.1 Consensus contract on L1

Consensus contract(abbreviated as CC) is the most important, it has below contents:

* The sequencers list, and how much they stake.
* Proposals of current round of mining.   
  A proposal contains the info needed by L1 to know the L2 chain, be aware that proposal contains transactions root, but doesn’t contain state root and receipts root, because we don’t know whether the sequencer is honest, if it lies on it, it’ll be complicated to correct it. So I wish ZK builder to submit it.
* Confirmed proposals list.  
  It’s a proposals list which had been chosen by consensus, but not proved by ZK proof.   
  Whenever a proposal got proved, it will be removed and be used to update the canonical chain head.
* Canonical chain head.  
  It’s the head of L2 chain. It includes the most important block info needed by L1, e.g. block height, transactions root, state root, receipts root, etc.
* Interfaces:
  + Submit proposal.   
    Sequencers call this to submit their proposals.
  + Consensus algorithm.  
    How to choose a proposal as the confirmed block.
  + Prove.  
    When ZK builder generated proof for a confirmed proposal, they’ll call this to prove it, if it works, the proper proposal will be upgraded to canonical chain. ZK builder will also submit state root and receipts root for this proposal.

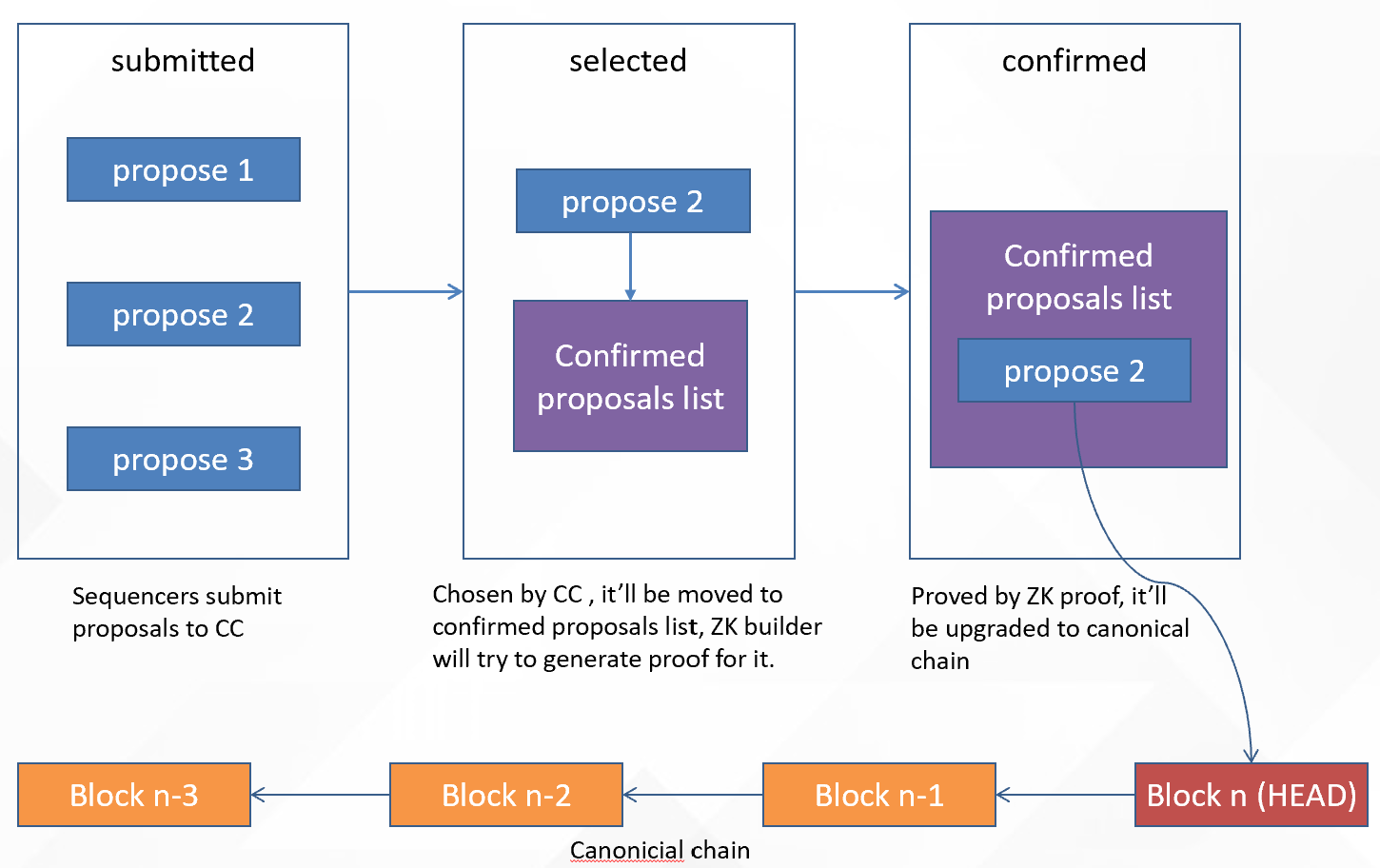
## How does it works



The work process is:

1. Players wish to be L2 sequencer will stake on CC. Normally there shouldn’t be too much sequencers, the contract will pick top N players as sequencer.
2. On L2, the sequencers collect transactions and build new block, and generate proposal on it. The proposal contains block height, previous block hash, transactions root, and anti-censorship proof (optional), CID of transactions , etc. They submit their proposals separately to CC. (Maybe later we can collect them together and send to CC, to save gas.)
3. When any sequencer submitted proposal, CC will check whether there’s enough proposals to do the selection (the check mainly consider stake ratio, also considering the number of proposals).   
   When the proposal is enough, CC runs the consensus algorithm, it picks a proposal from the proposals list as confirmed, it’ll be added to the confirmed proposals list. It will be the next block on L2.
4. L2 sequencers will listen to the result of 3. When a new proposal is chosen, they’ll get the transaction list from DA node, extend their chain, and prepare for the next block.  
   ZK builder will also listen to the result of 3. When a new proposal is chosen, they’ll consider whether to generate proof on it, if yes, they’ll try to do it. (Because there can be many ZK builders, they work parallel and compete to each other, only the 1st suppling proof on L1 can get paid.)
5. When ZK builder generated the proof, it’ll submit to CC. if succeed, the proposal will be upgraded to canonical chain head(if it’s height is the next, it’ll be upgraded. if its height is beyond the next block, it’s just be marked as proved, and will be upgraded when possible).  
   ZK builder will also submit the state root and receipts root, which can be proved by the proof.

## Life of a proposal



## Consensus algorithm

Any algorithm can choose someone from a list can work here. Here’s just an example.

When the sequencer submits a proposal, we give the proposal some number, the numbers are continued, the number of the numbers, is the sequencer’s stake. And the begin number of it, is the previous proposal’s end add 1. It’s simple in an example.

Think we have 4 sequencers, their stake is : 100, 200, 500, 1000. And they submit proposals as below sequence, and they get the numbers:

|  |  |  |  |
| --- | --- | --- | --- |
| Sequencers | Stake amount | Submit sequence | Got numbers |
| Sequencer 1 | 100 | 1 | 1-100 |
| Sequencer 2 | 200 | 2 | 101-300 |
| Sequencer 3 | 500 | 3 | 301-800 |
| Sequencer 4 | 1000 | 4 | 801-1800 |

In the consensus algorithm, we’ll get a random number (e.g. from oracle), and modulo on 1800, which’s the total number of the numbers, and get the remainder. The proposal holds the remainder will be the chosen one. E.g. the random number is 5000, modulo it with 1800, so the remainder is 1400, sequencer 4 holds it, so proposal 4 is chosen. In this way , the chosen ratio is based on the stake.

## Exceptions.

If a sequencer had malicious behavior , it may generate a fake block and cause the ZK builder can’t generate proof (invalid transactions can be marked as executed fail and won’t have influence on ZK proof, so this should be rare case). If this happened, the canonical chain will stop.

In this case , we’ll set a gap, e.g. 500 L1 blocks. If the proposal which should be the next canonical chain head, can’t be proved after the gap, we’ll consider it be invalid. Then the “confirmed proposals list” will be cleared and an event will be sent to remind the L2 sequencers to go back and submit new proposals on that height again. And we’ll consider to add the sequencer generated that invalid proposal to a “watch list”.