## Slido Q and A from Week 1

### 'Ethereum will achieve consensus by checkpointing'. Can you explain more about checkpointing?

```
 The checkpoint vote is used to finalize blocks to grow the
finalized chain.
```

### Reading up on ZK Maths, why can only addition/multiplication be applied to fields/polynomials and not subtraction for example."

```
 Field elements does support
addition/subtraction/multiplication and division. Although
when it comes to division, they behave differently from your
traditional division.
```

### Any particular reason for exact number of shards?

```
 Mostly based on performance and scalability. It can be
scaled up in the future if needed.
https://notes.ethereum.org/KbEyHiaSRQW_KS7dDK0OFw
```

### Are you referring to D-LOG problem with limited classes?

```
 Yes, especially when we choose parameters such that the
Discrete Logarithm Problem is hard to solve.
```

### Blobs are going to make ethereum kind of mutable in some parts? Able to remove the blobs when the data is not necessary anymore?

```
 Yes, I believe they are automatically deleted after 1-3
months. https://www.eip4844.com/
```

### By using the elliptic curves short length of the key offers more security, why? How we are getting this security?

```
 It comes from the elliptic curve discrete log problem which
is hard to solve computationally.
```

### Can we verify proofs on L2, if it is EVM compatible does it mean we can use pairing precompiled function?

```
 Yes, you can verify proofs on L2. If the L2 is EVM
compatible then it should support the usage of precompiled
functions.
```

**Can you speak more about why a system would use a stack versus a register?**

```
 Mostly because of a design choice and cryptographic
functions. https://faizannehal.medium.com/understanding-the-
stack-based-architecture-of-evm-af45dc9819f2
```

**Could you explain what goes in each category concretely, I mean for execution, settlement/consensus, and data availability?**

```
 This article gives you a detailed overview about it:
https://volt.capital/blog/modular-blockchains
```

**Cyclic groups and generator is equivalent to `map()` in rust right we are applying the same operator to each element? Just to be sure?**

```
 I would say it's different as one is mathematics and the
other one is programming. The map transforms an iterator.
```

with the generator you are repeatedly applying the operator, so for example if we start at 7 and our operator is + we would get 7 + 7 + 7 ...

**Do all failed transactions executed on the sequencer get propagated and included in l1 state?**

```
 Yes, reverted/failed transactions are included as well.
```

**Do we have a support of BN128 EC on zkEVM?**

```
 Yes, we will talk about elliptic curves in in the upcoming
lessons.
```

**Do ZK rollups submit only the proof or is that an addition to the execution result?**

```
 L2's first submit the proof to the L1 and if correct they
submit the state diff to L1 to update the state (this one is
the data availability problem that we will see).
```

**Don't we need transaction data from the very first rollup block to compute state root?**

For zkEVMs, the proof shows validity of execution trace of a transaction, so a zkEVM like zkSync simply sends account updates to L1 (along with the proof of correct execution). These account updates can be used to compute state root in the case of sequencer failure.

**Follow-up on that last one, is it byte length that makes it less zk friendly? Or the process of generating the hash?**

zk friendly hash functions are optimized for modular arithmetic in a finite field which reduced the circuit complexity, while for example SHA256 are unsuitable due to the large number of bitwise operations.

**How Avalanche reaches finality that fast?**

Don't know much about Avalanche, but AFAIK, it's because of their Avalanche consensus where it operates through repeated sub-sample voting. This enables quick finality and low latency.
I guess is their use of Subnets which is are a dynamic subset of Avalanche validators working together to achieve consensus

**How can one access to these data blobs if the EVM does not have access? Also with an RPC call to the node or differently?**

These blobs are on the protocol level and not for the EVM to be used.

**How can rollup smart contract on L1 'enforce' correct execution on layer 2? Rollup contract only verifies execution I think.**

The L2 sends a proof to the L1 and the L1 verifies if the proof is valid. If the proof is valid, then the state is updated.

**How can states be rolled-back on L1 once a fraud/validity proof has been sent? Or must it be sent before they're sent to L1?**

```
 In   Optimistic rollups, the state transitions are not
finalized on L1 until the challenge window has passed.
```

### How does rollups differ from hyperchains?

```
 Mostly their approach to validate and finalize the
transactions.
https://era.zksync.io/docs/reference/concepts/hyperscaling.ht
ml
```

### How does sharding differ from a sidechain-based system?

```
 "Well both are scalability solutions however they operate
differently. Sharding is as Laurence explained in the
lecture, addresses the scalability by breaking the ETH
network in smaller pieces which allows parallel transaction
processing. Sidechains is a separate blockchain network and
they rely on their own security and it runs in parallel to
the Ethereum chain.
https://ethereum.org/en/roadmap/danksharding/"
```

### How long for a fraud/validity proof to be sent before a block is finalized on L1?

```
 In general, a few hours for validity proofs.
```

### How proto-danksharding blobs used as checkpoints differ from custom weak subjectivity checkpoints (cache most recent finalize block) used in light client helios

```
 "Blobs do not serve as checkpoints, but rather as temporary
storage for data that can be discarded when it's no longer
needed.
On the other hand, the weak subjectivity checkpoints in
Helios are part of the consensus mechanism and are state
roots that all nodes on the network need to agree.
https://ethereum.org/en/developers/docs/consensus-
mechanisms/pos/weak-subjectivity/"
```

### How the new state roots are generated on L2?

The Sequencers bundles up the transactions and generates a new state which then a proof is generated and sent to the L1 Prover.

**How this is can be atomic? for the bridge? This is not in the same transaction correct? As this is not on the same layer by definition this is not atomic right?**

One mechanism when transferring assets through a bridge "lock and mint". You can read more about it here https://wormhole.com/

**I am a Python, JS, TS developer. I do understand C, C++, but do not know Rust. Would not knowing Rust make it difficult for me to do the practicals?**

Not at all, we will provide some interactive homework to help you and the lectures will explain the syntax which is different. The most difficult part might be the reference / borrowing which is different from the languages you mentioned. You can check the rust book for more: https://doc.rust-lang.org/book/

**I couldn't understand the l2 architecture after sequencer. What happening exactly? How correctness is calculated and verified?**

L2 submits a proof to the L1 contract and this verifies for correctness. If this proof is valid, then the state is updated.

**Could do this edge case make the probabilistic check fail?**

Yes, edge cases can make a probabilistic check fail. If a probabilistic algorithm is used to solve such problems, and if it’s not designed to handle such edge cases, it might fail or produce incorrect results.

**If correctness of execution is the main criteria for zk rollups. Is there any performance benefits in order to generate the proof?**

Higher throughput and increased efficiency since we bundle transactions. Other benefits are data compression and lower

gas fees.

**If for optimistic rollups transactions are checked (on demand) on L1, does it mean that optimistic rollups should be EVM-based?**

Yes, Optimistic Rollups are designed to be compatible with EVM-based.

**If Plasma has the disadvantage of possible congestion to exit, don't rollups have the same issue?**

Well, both have different implications for network congestion. I believe if there is a "mass exit" such as withdrawing assets on Plasma, then it would lead to congestions, while on a Rollup this wouldn't lead to the same level of congestions as on Plasma since the transactions are bundled together and then posted on L1.

**If we are storing data itself off-chain, then what limits us to use multiple l1's to store proofs and make cross chain operations?**

You run in a decentralization issue by storing it off-chain. In addition there are other factors you need to think about such as security, interoperability between those chains etc. So the problem is quite a complex one.

**If we can "send assets directly to the contract", why do we have bridges? or are those contracts the bridges?**

If you want to move assets from L1 to L2 you are using a bridge for that which inherently would be a smart contract or several.

**Is the operation always needs to be binary one?**

Yes, a group is defined as a set of elements together with a binary operations.

**Is the sequencer in for ex solana what orchestrates the parallel execution?**

```
 Isn't that Sealevel in Solana that does the parallelization? That is not a sequencer AFAIK
```

**Is there a possibility of different polinomials having a similar range of points?**

```
 Yes, although there may be overlap see the Schwarz zippel lemma
```

**Log handling same for zkevm as L1 ETH?**

```
 Yes, by emitting events.
```

**So is 1 blob = 1 tx?**

```
 Not necessary. Blobs are a new transaction type known as blob-carrying transaction. They are like normal transactions but carry an extra piece of data, a blob.
```

**Synthetix is in mainnet ETH and OP. Does it mean they've deployed a similar contract on OP (as if it were a sidechain) or I use OP to interact with L1(as a L2)?**

```
 "Sorry, I'm not that familiar with Synthetix, but it's possible to move asset between L1<>L2 similar to L1<>L2 messaging. But maybe you can find more info on their blog. https://blog.synthetix.io/author/synthetix/"
```

**The difference between Validium and rollup is not clear.**

```
 Validium doesn't store data onchain while Rollups do. https://ethereum.org/en/developers/docs/scaling/validium/ https://ethereum.org/en/developers/docs/scaling/zk-rollups/
```

**Tx are sent to modify L2 state (balance, tokens transfer, etc). Does a smart contracts has to be deployed on L2, and assets transfered to L2 too?**

```
 Not sure if I understand the question. If you want to move your assets, you would be using a bridge between L1 and L2. Then you can use those assets on L2 as you please. Is that what you asking?
```

**What are bulletproofs?**

```
 A ZKP implementation.
https://blog.pantherprotocol.io/bulletproofs-in-crypto-an-introduction-to-a-non-interactive-zk-proof/
```

**What do the L2's do to make smart contract wallets better / easier to implement than on L1 ethereum?**

```
 They provide Account Abstraction and act as a smart contract instead of your traditional EOA.
Lots of use cases for that, such as implementing a multi-factor authentication, session keys for gaming, etc
```

**What do we mean by valid in the context of tx? I mean are we talking about is tx valid according to current state? Can you give an example of it?**

```
 we are proving that the execution of the code happened correctly
```

**What does BFT stand for ? (consensus mechanism)**

```
 Byzantine Fault Tolerance
```

**What does the ethereum merge have to do with L2 and zkEVMs? Can you draw a clear distinction between the merge and L2s / zkEVMs?**

```
 "The Ethereum Merge is about switching from POW to POS and introducing sharding for scalability while L2s are built on top of Ethereum to solve the scalability with Validity Rollups / optimistic rollups
https://ethereum.org/en/roadmap/merge/
https://ethereum.org/en/layer-2/"
```

**What exactly is the zero knowledge aspect of the validity proof in ZK Rollups?**

```
 Currently since we use Validity proof, we don't have privacy, but usually in a zk-proof the input stays private.
```

**what is "stocknet"?**

```
 It's Starknet an L2 zk-rollup which uses STARKS proof:
https://docs.starknet.io/documentation/
```

**What is a commitment scheme ?**

A cryptographic protocol that allows one to commit to a chosen value or statement while keeping it hidden and then reveal it later.
https://en.wikipedia.org/wiki/Commitment_scheme

**What is passed in the calldata once blobs are introduced? So that zkrolluip can ver.**

Okay I see in the notes that the commitment is passed, so is that consumed? In essence yes, the commitment is passed in the calldata to verify the integrity of the data in the blobs.

**What is poly-log time?**

Refers to the speed of computation. Check the big O notation https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis/

**What is the difference between an optimistic rollup and a plasma chain ? They seem to be pretty similar regarding their security premises**

They differ in their approach to validating and finalizing the transactions. Optimistic uses fraud-proof mechanism while Plasma uses a different design.

**What is the difference between merkle and verkle trees?**

Verkle trees are more efficient in terms of proof size than Merkle trees. However, Verkle trees are more complex to implement.

**What is the meaning of a hash function is not zk friendly?**

It's not optimized for use in ZK proofs. You can compare Poseidon hash which is a zk-friendly to SHA256.

**When do we take a deep dive into sequencer? This week or next week?**

Next week I believe, in L8.

**When do we use the memory vs the stack?**

```
 The memory is used for storing data during contract execution while the stack is used for performing operations on this data.
```

**Where does the sequencer sit in L2 stack? Does the execution client submit txs to the sequencer? Do you have any diagrams that show the full lifecycle of a tx?**

```
 yes typically the txs go to the sequencer who then sends them to the VM
```

**Which hash functions are zk friendly?**

```
 Poseidon hash function for example.
```

**Which is the polynomial chosen? Is it always the same? does it depend on the protocol or on the transactions itself, to choose one or another?**

```
 The polynomial is influenced by several factors, however we will cover it in more detailed in the upcoming lessons.
```

**Why do we need to send data to L1 for zk rollups, why is it not enough to just send proof? We send only data for optimistic rollups. What am I missing?**

```
 The data that is sent to L1 is the state diff between the previous and the new state. This data allows anyone that observes Ethereum to reconstruct the current state of the L2.
```

**Why does using fields make it kinda harder for programming for eg, in zk languages differing from traditional languages?**

```
 I would say it's a different mathematical foundation since ZK proofs operate on finite fields. Limited operations could be another case where for example bitwise operations are extremely expensive for circuits.
```

**Why don't L2s support Shanghai EVM upgrades, e.g. the 0x5f opcode? afaik it would be EIP-4844, but you'll need to research that more.**

```
 In the upcoming lessons we will look at compatibility which might answer your question.
```

**Why don't we use the zero knowledge aspect to add privacy into the network ? couldn't privacy and scaling be achieved altogether with zero knowledge?**

```
 Most L2 are focused on scalability first and then will add privacy after.
```

**Why isn't calldata considered persistent?**

```
 "the calldata is different with every contract call, it's not persistent storage
```

**why nonce takes 0 space in L2?**

```
 With account abstraction it is handled differently
```

**Will MEV remain valid with the emergence of the likes of Flashbots?**

```
 The aim of Flashbots is to mitigate the negative externalities posed by MEV, so will MEV remain valid? Depends on many things.
```

**You mentioned in optimistic rollups they are able to lower a tx down to ~12 bytes. do zk based rollups optimize like this or is it not comparable?**

```
 We will have a comparison table between them on page 22 from this lesson, this is applies generally to both types of rollups, though implementations vary.
```