

# Canyon: Permanent Storage Layer for Limitless Scalability

Liu-Cheng Xu

## Abstract

TODO

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Organization . . . . .	2
<b>2</b>	<b>Related works</b>	<b>2</b>
2.1	Filecoin . . . . .	2
2.2	Crust . . . . .	2
2.3	Arweave . . . . .	2
<b>3</b>	<b>System design</b>	<b>2</b>
3.1	Consensus . . . . .	2
3.1.1	Proof of Access . . . . .	2
3.1.2	Proof of Stake . . . . .	3
3.2	Economy Model . . . . .	3
3.2.1	Perpetual Storage Cost . . . . .	3
3.2.2	Transaction Fee . . . . .	3
3.2.3	Data Oblivion . . . . .	4
3.2.4	Multi-currency Payment . . . . .	4
3.3	Transaction Pool . . . . .	4
<b>4</b>	<b>Roadmap</b>	<b>4</b>
<b>5</b>	<b>Conclusion</b>	<b>4</b>
5.1	Future work . . . . .	4
<b>A</b>	<b>Appendix</b>	<b>4</b>

# 1 Introduction

## 1.1 Motivation

TODO

## 1.2 Organization

TODO

# 2 Related works

## 2.1 Filecoin

TODO

## 2.2 Crust

TODO

## 2.3 Arweave

TODO

# 3 System design

## 3.1 Consensus

### 3.1.1 Proof of Access

$$P(\text{win}) = P(\text{has recall block}) * P(\text{finds hash first}) \quad (1)$$

$$P(\text{win}) = P(\text{has recall block}) * P(\text{claims slot}) \quad (2)$$

$$\hat{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

$$R = \frac{1}{\hat{x}_{N \rightarrow +\infty}}$$

---

**Algorithm 1:** Generation of POA

---

**Input :**

The random seed  $S$ ;  
The weave size  $W$ ;

**Output:**

The proof of accesing the recall block  $POA$ ;

```
1 Initialize the number of repeats  $x$  with 1;  
2 repeat  
3   Draw a random byte  $B$  with  $\text{MULTIHASH}(S, x) \bmod W$ ;  
4   Find the  $TX$  in which the random byte  $B$  is included;  
5    $x \leftarrow x + 1$ ;  
6 until The data of  $TX$  is available;  
7  $POA \leftarrow \text{CONSTRUCTPOA}(TX)$ ;  
8 return  $POA$ ;
```

---

### 3.1.2 Proof of Stake

#### 3.1.2.1 Staking Rewards

#### 3.1.2.2 Stake

## 3.2 Economy Model

### 3.2.1 Perpetual Storage Cost

$$P_{GBH} = \frac{HDD_{price}}{HDD_{capacity} * HDD_{mtbf}} \quad (3)$$

- $P_{GBH} =$
- $HDD_{price}$
- $HDD_{capacity}$
- $HDD_{mtbf}$

$$P_{store} = \sum_{i=0}^{\infty} (Data_{size} * P_{GBH}[i]) \quad (4)$$

### 3.2.2 Transaction Fee

$$TX_{permacost} = TX_{data\_size} * Sum \quad (5)$$

$$TX_{bandwidthcost} = TX_{data\_size} * C_{network\_per\_byte} \quad (6)$$

$$TX_{reward} = TX_{permacost} * C_{fee} + TX_{bandwidthcost} \quad (7)$$

$$TX_{total} = TX_{permacost} + TX_{reward} \quad (8)$$

- $TX_{permacost}$

- $TX_{data\_size}$
- $TX_{bandwidthcost}$
- $TX_{reward}$
- $TX_{total}$

### 3.2.3 Data Oblivion

TODO

### 3.2.4 Multi-currency Payment

TODO

## 3.3 Transaction Pool

TODO

## 4 Roadmap

TODO

## 5 Conclusion

### 5.1 Future work

## References

- [1] Protocol Labs, Filecoin: A Decentralized Storage Network, <https://filecoin.io/filecoin.pdf>.
- [2] <https://crust.network/>
- [3] Sam Williams, Viktor Diordiiev, Lev Berman, India Raybould, Ivan Uemlianin. Arweave: A Protocol for Economically Sustainable Information Permanence.
- [4] Gavid wood. Polkadot: Vision for A Heterogeneous Multi-chain Framework. <https://polkadot.network/PolkaDotPaper.pdf>
- [5] <https://github.com/paritytech/substrate>
- [6] Vitalik buterin. An Incomplete Guide to Rollups, <https://vitalik.ca/general/2021/01/05/rollup.html>
- [7] A Storage-based Computation Paradigm Enabled by Arweave, <https://medium.com/everfinance/a-storage-based-computation-paradigm-enabled-by-arweave-de799ae8c424>
- [8] <https://ever.finance/>

## A Appendix