Plasma DAG

한겨레 (kr8534@gmail.com)

Plasma DAG 팀 (김동한, 김재윤, 박상현, 박성완,
박준모, 박찬, 유진영, 이규택, 한겨레)

Decipher
Blockchain Research Group at SNU

It’s Me

한겨레

- ✓ 서울대학교 전기공학부 학사 졸업
- ✓ 서울대학교 Petabyte-scale In-memory Database 연구실 박사과정
- ✓ 서울대학교 블록체인 학회 디사이퍼 공동 조직

블록체인 관련 연구

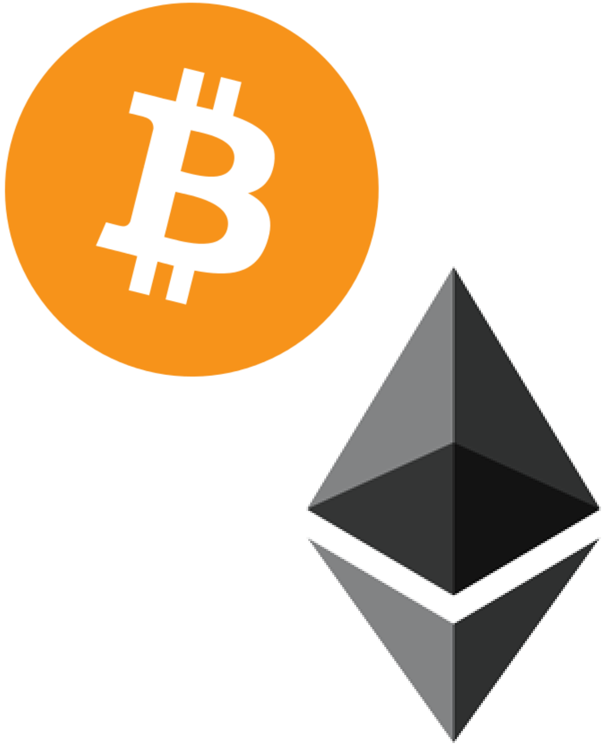
- ✓ Blockchain Scalability Solutions Survey
- ✓ Queryable Blockchain
- ✓ Plasma DAG (On-going)

Agenda

- 연구 목적
- Plasma DAG (Directed Acyclic Graph)
  - ✓ 기본 개념도
  - ✓ 동작 방식
  - ✓ 공격 대응
- 맺음말
  - ✓ 시행착오
  - ✓ 백서 및 구현 현황
  - ✓ 마일스톤

연구 목적

Crossing the Chasm: Blockchain Scalability



Crossing the Chasm: Blockchain Scalability

확장성

보안

상호운용성

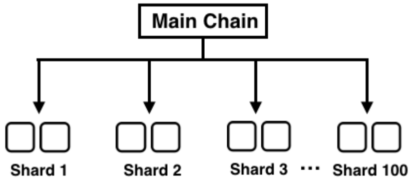
연구 목적

Crossing the Chasm: Blockchain Scalability

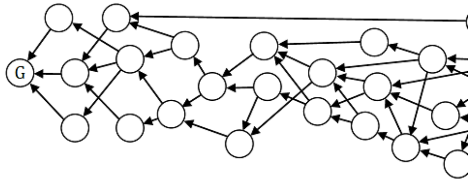
확장성

보안

상호운용성



State channel



Sidechain
/Interchain

Feasible Scalability Solution – Plasma Variants

- Plasma MVP ➔ 관찰비용 너무 큼, 이중서명으로 인한 불편함
- Plasma Cash ➔ 관찰비용을 줄였으나 사용이 불편함
- Plasma Debit ➔ Proof size가 너무 큼
- Plasma Ignis / SNAPP ➔ 결국 모든 트랜잭션 관찰하는 셈, ZKP가 DA(Data Availability) 문제를 해결해주지는 않음

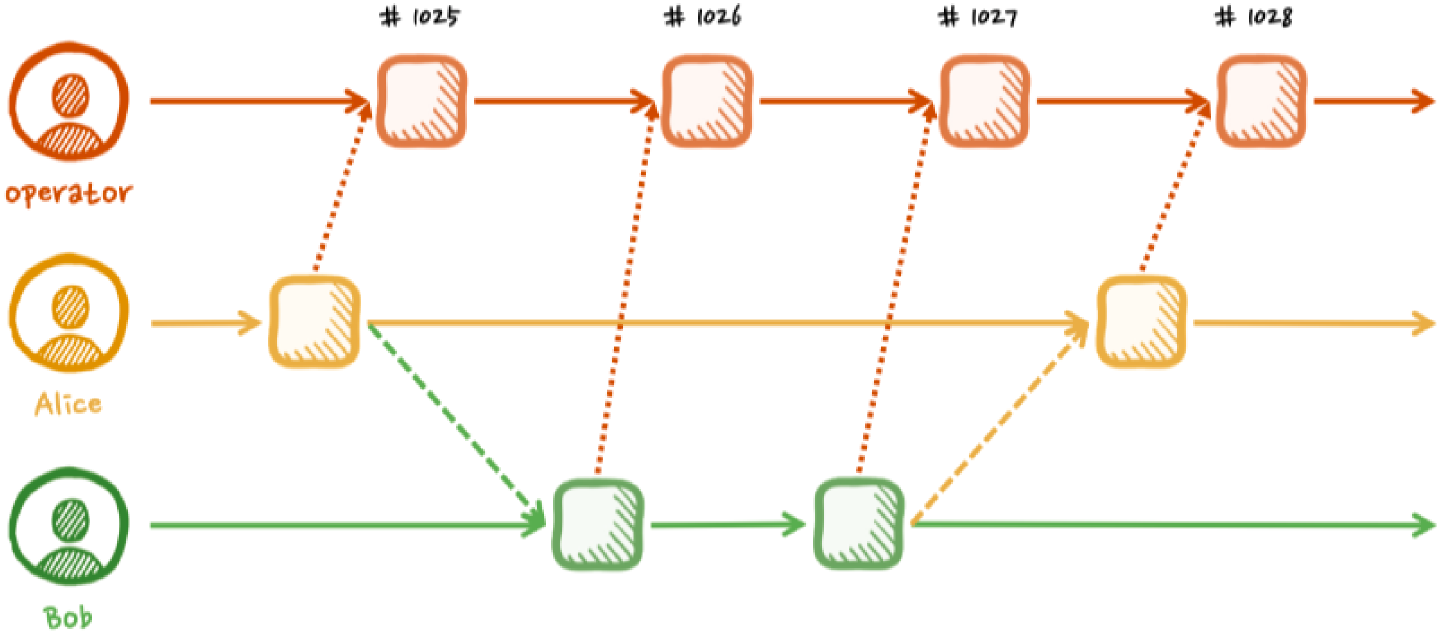
Plasma DAG

기본 개념도

동작 방식

공격 대응

기본 개념도

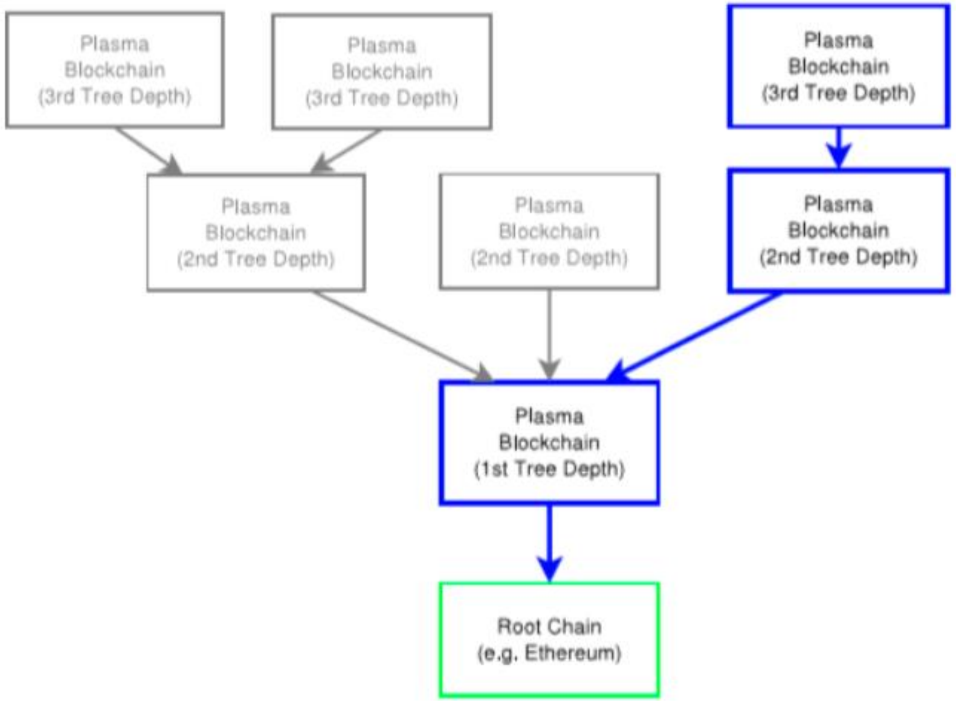


- Plasma DAG의 참여자들은 각자 스스로와 관련된 트랜잭션을 포함하는 블록을 연결, 자신만의 블록체인 유지 (NANO-like DAG)
- Operator는 참여자들이 생성하는 블록을 검증하고, 검증된 블록을 본인의 체인에 연결해나가며 global state를 관리
- Operator는 주기적으로 부모 체인 (e.g. 메인체인)에 aggregate 된 TX 요약을 전송하여 확률적 finality 보장

관찰비용의 절감

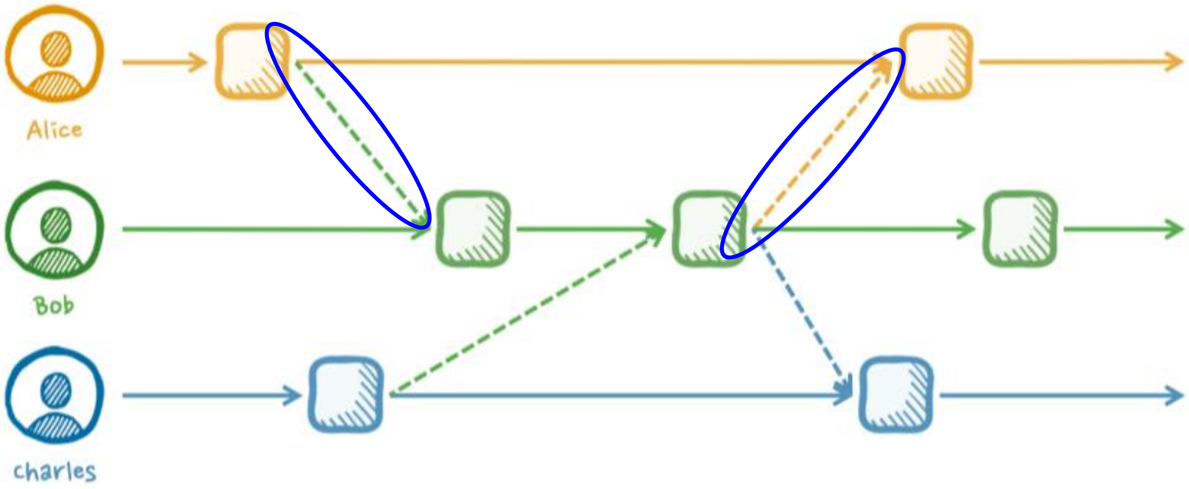
AS-IS

소속된 체인의 모든 트랜잭션을 지켜봐야 함


Plasma Blockchain (3rd Tree Depth)
Plasma Blockchain (3rd Tree Depth)
Plasma Blockchain (3rd Tree Depth)
Plasma Blockchain (2nd Tree Depth)
Plasma Blockchain (2nd Tree Depth)
Plasma Blockchain (2nd Tree Depth)
Plasma Blockchain (1st Tree Depth)
Root Chain (e.g. Ethereum)

TO-BE

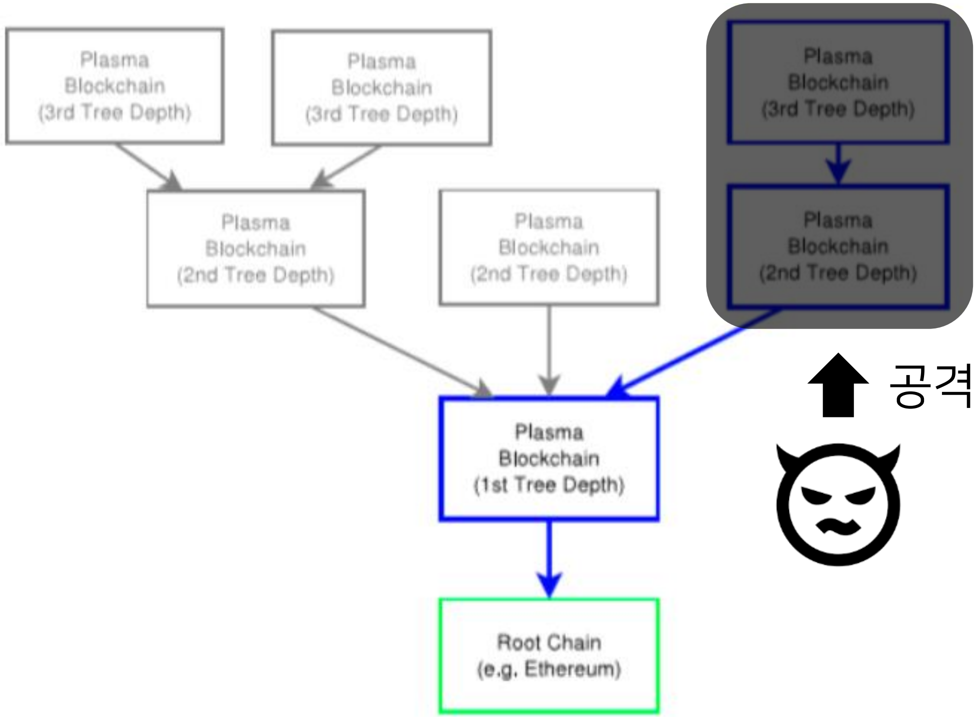
본인과 관련된 트랜잭션만 보면 됨



사이드 체인 안정성 증가

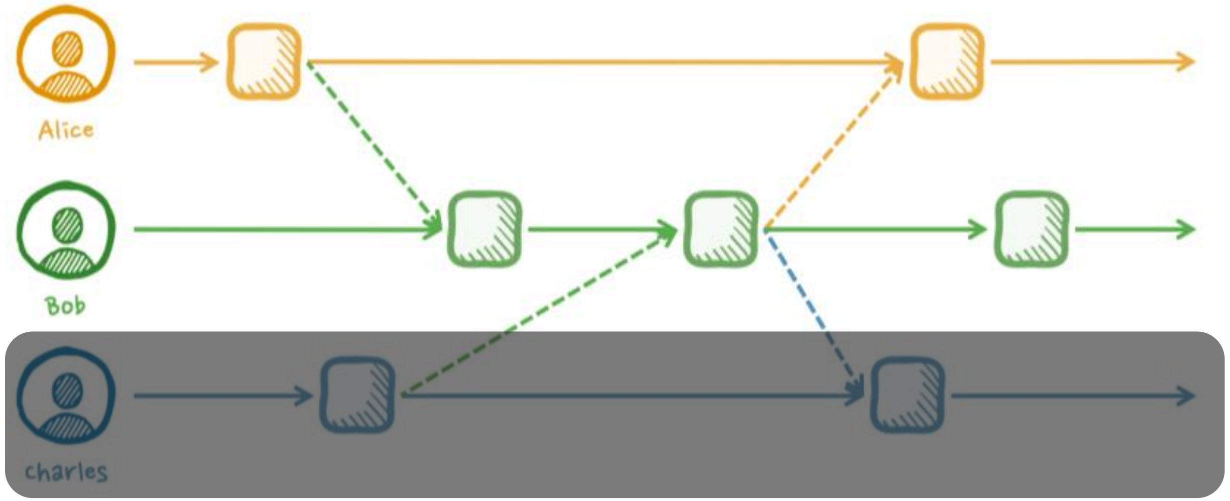
AS-IS

한 명의 잘못된 행동으로 체인이 파괴

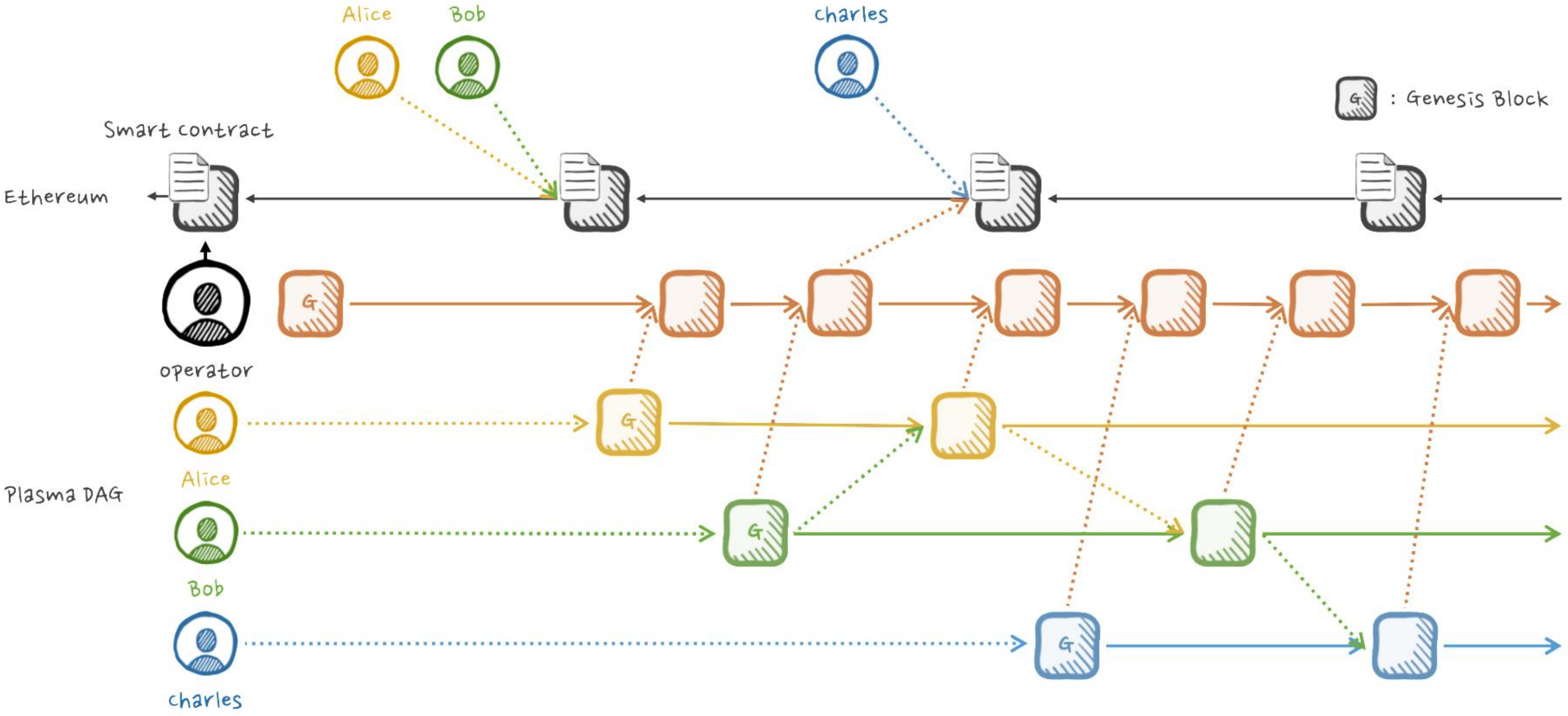

Plasma Blockchain (3rd Tree Depth)
Plasma Blockchain (3rd Tree Depth)
Plasma Blockchain (3rd Tree Depth)
Plasma Blockchain (2nd Tree Depth)
Plasma Blockchain (2nd Tree Depth)
Plasma Blockchain (2nd Tree Depth)
Plasma Blockchain (1st Tree Depth)
Root Chain (e.g. Ethereum)
공격

TO-BE

Mal-acting 당사자만 slash 되고
나머지 사용자는 그대로 진행

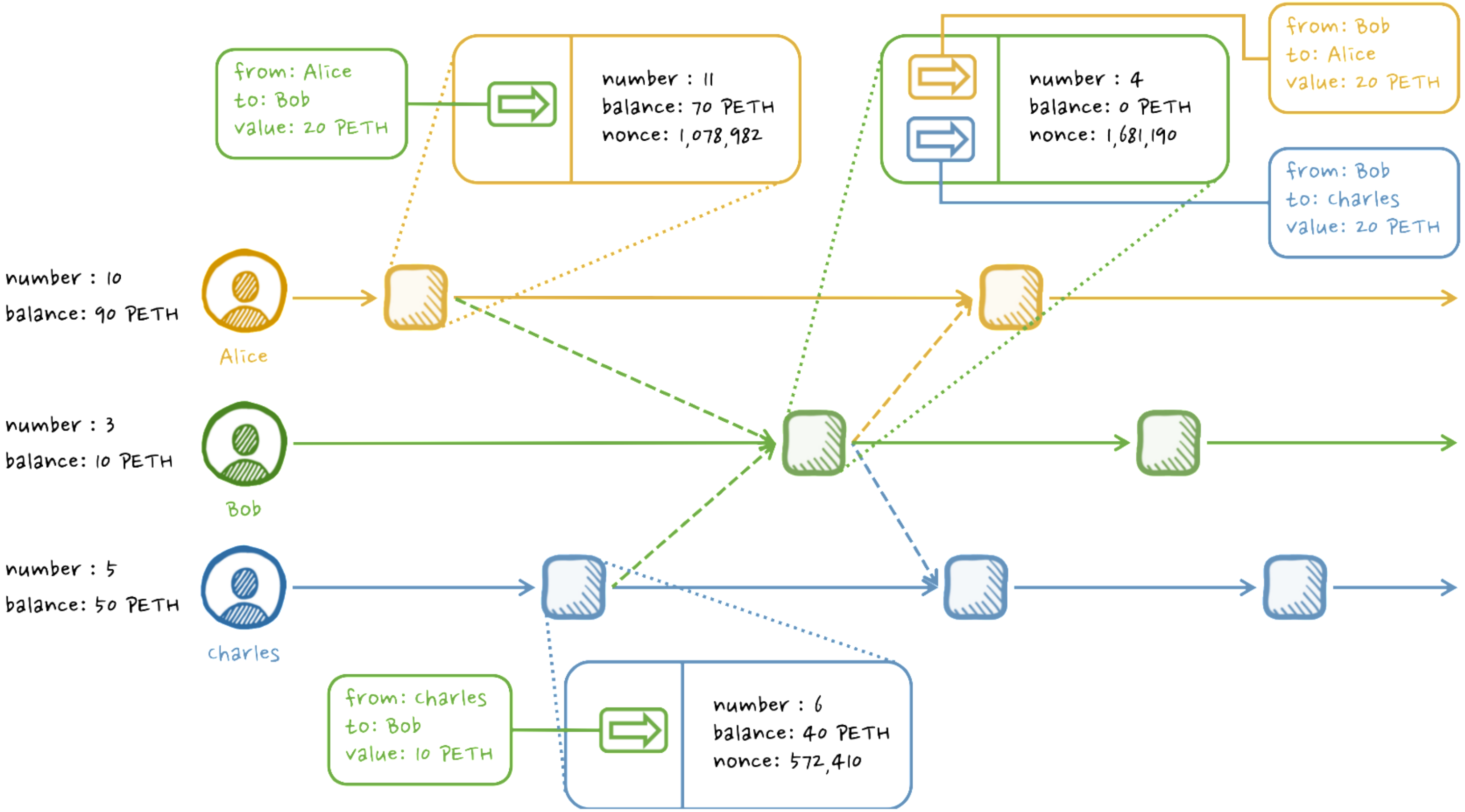


동작 방식 – 초기화 및 사용자의 참여



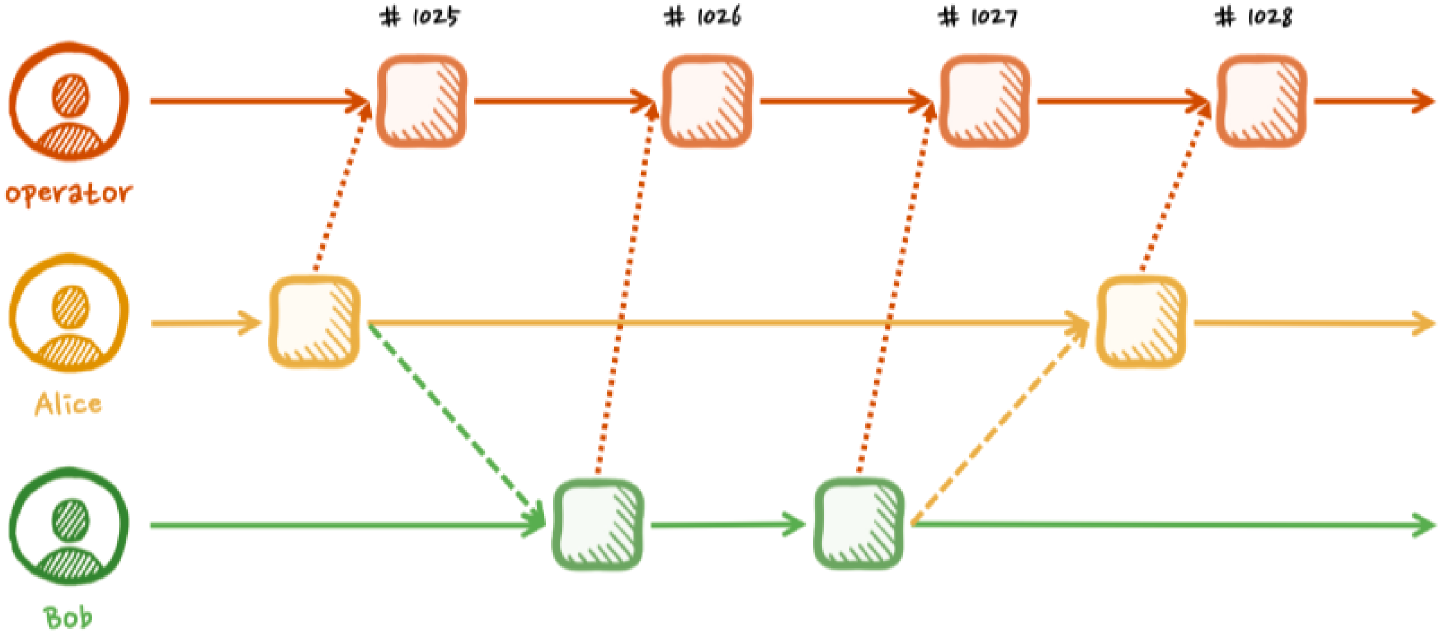
- Operator가 deposit을 smart contract에 예치하고 사이드체인 초기화
- 참여자들은 본인의 deposit 만큼을 smart contract에 예치한 후, 본인의 블록 생성 후 operator에게 승인 요청
- 참여자들의 deposit 총량은 operator의 deposit을 넘을 수 없음

동작 방식 – 트랜잭션 발행



- 송금을 원하는 참여자는 본인의 서명과 함께 해당 트랜잭션을 포함하는 블록을 생성하고 이 정보를 전달
- 입금 받는 참여자의 경우 본인이 해당 트랜잭션을 받았다는 것을 알려야 함. 그 전에 생성된 트랜잭션들을 포함하는 블록을 생성한 후 이를 공표함으로써 다른 사용자들이 이를 알 수 있도록 함

동작 방식 – 트랜잭션 승인



- Operator 또한 블록체인을 유지하지만, 스스로 트랜잭션을 발행하지 않음. 참여자들이 생성한 블록을 검증하고 해당 블록이 올바를 경우 이를 본인의 체인에 연결
- 참여자들은 operator의 체인을 보면서 본인의 블록이 담겼는지를 확인하여 승인 여부를 판단할 수 있음
- Operator가 검증을 올바르게 하지 못해 특정 참여자가 손해를 봤을 경우, 해당 손해를 operator의 deposit에서 제외하기 때문에 성실하게 검증할 유인이 있음

공격 대응 – Double Spending

- 참여자의 double spending
  ➔ operator가 승인을 해주지 않음 (제대로 검증하지않고 승인 시 본인의 deposit이 깎이기 때문에)

- Operator의 double spending
  - ✓ 전체 supply가 바뀔 경우 (없는 돈을 추가할 경우) smart contract 에서 revert
    ➔공격에 성공하려면 다른 누군가의 state를 조작해야 함
  - ✓ 특정 참여자의 state를 조작할 경우 (전체 supply는 불변) 해당 참여자는 조작 전 블록으로부터 탈출 요청
    ➔ challenge period 가 시작되고 operator가 탈출 시점 이후
    참여자 signature 가 담긴 블록을 증명하지 못할 경우 사용자는 탈출 가능

맺음말

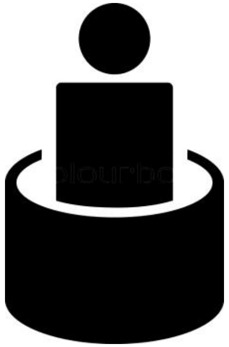
시행 착오

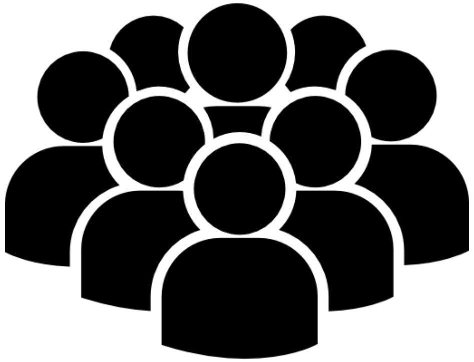
백서 작성 및 구현 현황

마일스톤

우리팀 고생했어요

시행착오





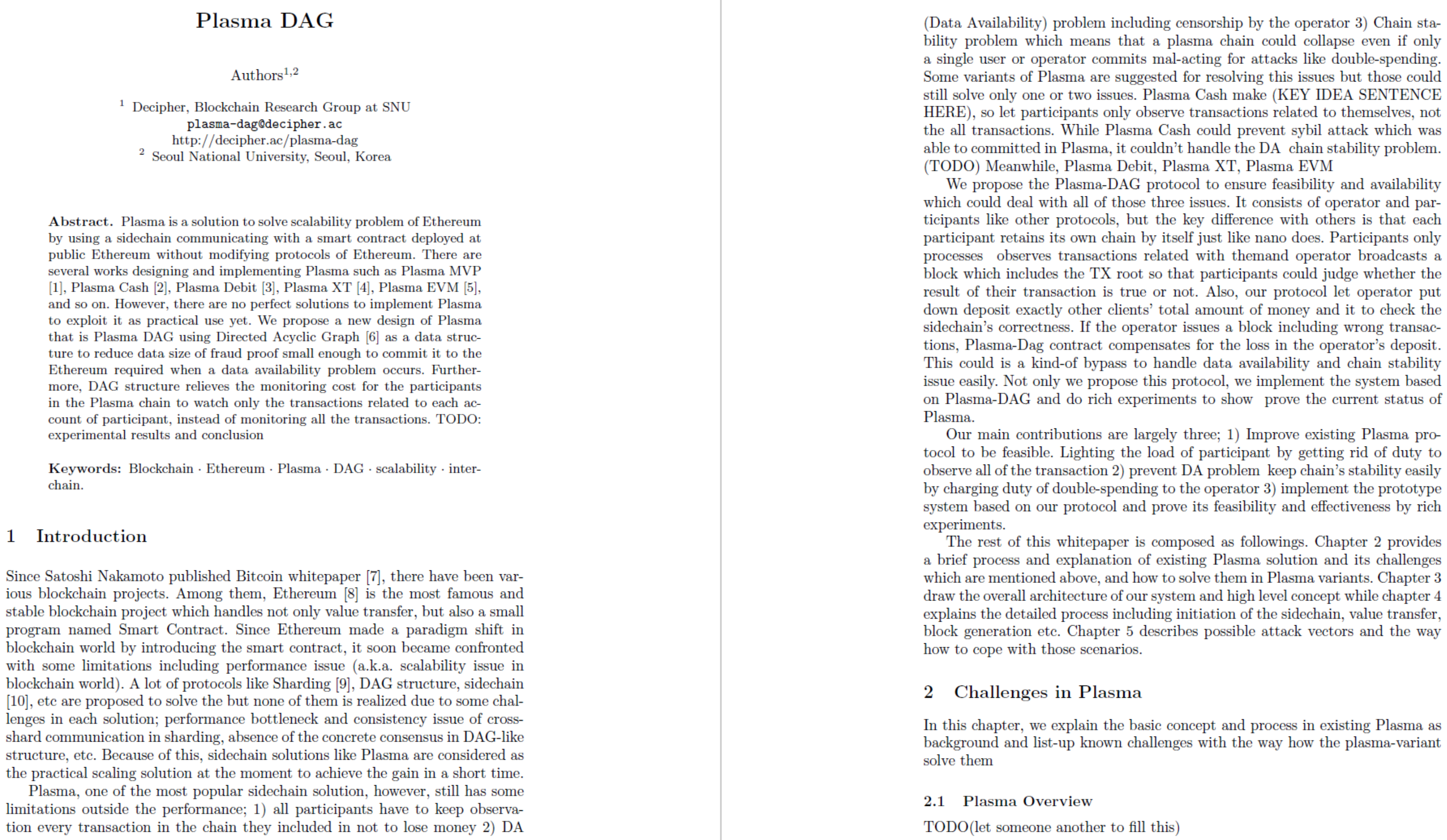
백서 작성 및 구현 현황

- 백서
  - ✓ **완료:** Intro / Related Work / Design Overview / Behavioral Process
  - ✓ **작성중:** Attack Vectors
  - ✓ **연구중:** succinct sum-check method (ZKP)

- 프로토타입
  - ✓ **완료:** 코어 모듈 (블록, 트랜잭션, 검증 등) 구현 및 테스트
  - ✓ **설계/구현중:** UI / ETH smart contract

	설계	구현	테스트
Core 모듈	O	O	O
User Interface	△	△	△
Network (Plasma DAG)	O	O	O
Network (Ethereum)	△	X	X

백서 작성 및 구현 현황



Plasma DAG

Authors[1,2]

[1] Decipher, Blockchain Research Group at SNU
plasma-dag@decipher.ac
http://decipher.ac/plasma-dag
[2] Seoul National University, Seoul, Korea

Abstract. Plasma is a solution to solve scalability problem of Ethereum by using a sidechain communicating with a smart contract deployed at public Ethereum without modifying protocols of Ethereum. There are several works designing and implementing Plasma such as Plasma MVP [1], Plasma Cash [2], Plasma Debit [3], Plasma XT [4], Plasma EVM [5], and so on. However, there are no perfect solutions to implement Plasma to exploit it as practical use yet. We propose a new design of Plasma that is Plasma DAG using Directed Acyclic Graph [6] as a data structure to reduce data size of fraud proof small enough to commit it to the Ethereum required when a data availability problem occurs. Furthermore, DAG structure relieves the monitoring cost for the participants in the Plasma chain to watch only the transactions related to each account of participant, instead of monitoring all the transactions. TODO: experimental results and conclusion

Keywords: Blockchain · Ethereum · Plasma · DAG · scalability · interchain.

1 Introduction

Since Satoshi Nakamoto published Bitcoin whitepaper [7], there have been various blockchain projects. Among them, Ethereum [8] is the most famous and stable blockchain project which handles not only value transfer, but also a small program named Smart Contract. Since Ethereum made a paradigm shift in blockchain world by introducing the smart contract, it soon became confronted with some limitations including performance issue (a.k.a. scalability issue in blockchain world). A lot of protocols like Sharding [9], DAG structure, sidechain [10], etc are proposed to solve the but none of them is realized due to some challenges in each solution; performance bottleneck and consistency issue of cross-shard communication in sharding, absence of the concrete consensus in DAG-like structure, etc. Because of this, sidechain solutions like Plasma are considered as the practical scaling solution at the moment to achieve the gain in a short time.

Plasma, one of the most popular sidechain solution, however, still has some limitations outside the performance; 1) all participants have to keep observation every transaction in the chain they included in not to lose money 2) DA (Data Availability) problem including censorship by the operator 3) Chain stability problem which means that a plasma chain could collapse even if only a single user or operator commits mal-acting for attacks like double-spending. Some variants of Plasma are suggested for resolving this issues but those could still solve only one or two issues. Plasma Cash make (KEY IDEA SENTENCE HERE), so let participants only observe transactions related to themselves, not the all transactions. While Plasma Cash could prevent sybil attack which was able to committed in Plasma, it couldn't handle the DA chain stability problem. (TODO) Meanwhile, Plasma Debit, Plasma XT, Plasma EVM

We propose the Plasma-DAG protocol to ensure feasibility and availability which could deal with all of those three issues. It consists of operator and participants like other protocols, but the key difference with others is that each participant retains its own chain by itself just like nano does. Participants only processes observes transactions related with themand operator broadcasts a block which includes the TX root so that participants could judge whether the result of their transaction is true or not. Also, our protocol let operator put down deposit exactly other clients' total amount of money and it to check the sidechain's correctness. If the operator issues a block including wrong transactions, Plasma-Dag contract compensates for the loss in the operator's deposit. This could is a kind-of bypass to handle data availability and chain stability issue easily. Not only we propose this protocol, we implement the system based on Plasma-DAG and do rich experiments to show prove the current status of Plasma.

Our main contributions are largely three; 1) Improve existing Plasma protocol to be feasible. Lighting the load of participant by getting rid of duty to observe all of the transaction 2) prevent DA problem keep chain's stability easily by charging duty of double-spending to the operator 3) implement the prototype system based on our protocol and prove its feasibility and effectiveness by rich experiments.

The rest of this whitepaper is composed as followings. Chapter 2 provides a brief process and explanation of existing Plasma solution and its challenges which are mentioned above, and how to solve them in Plasma variants. Chapter 3 draw the overall architecture of our system and high level concept while chapter 4 explains the detailed process including initiation of the sidechain, value transfer, block generation etc. Chapter 5 describes possible attack vectors and the way how to cope with those scenarios.

2 Challenges in Plasma

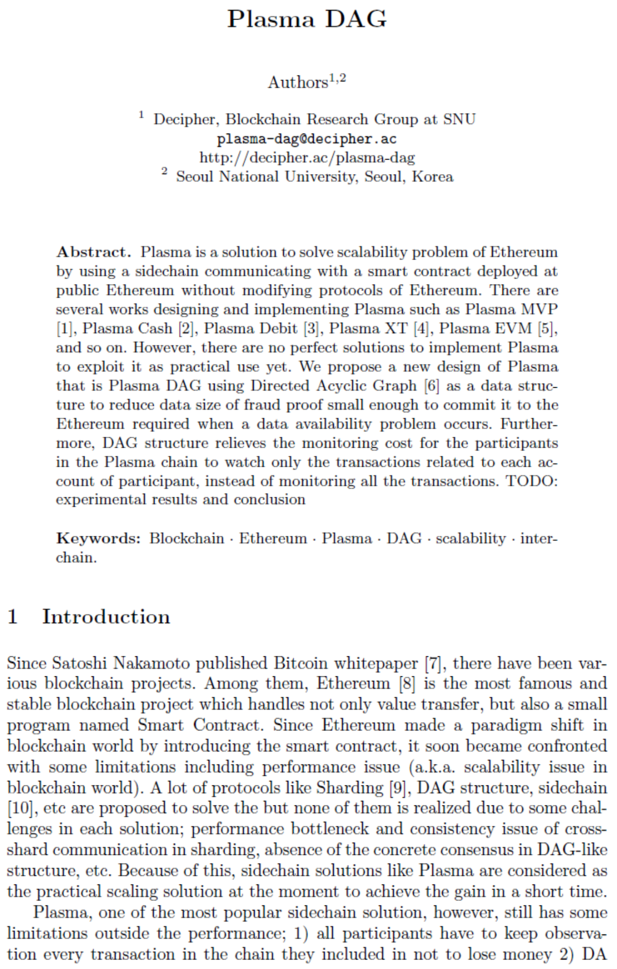
In this chapter, we explain the basic concept and process in existing Plasma as background and list-up known challenges with the way how the plasma-variant solve them

2.1 Plasma Overview

TODO(let someone another to fill this)

설계	구현	테스트
O	O	O
△	△	△
O	O	O
△	X	X

백서 작성 및 구현 현황



Plasma DAG

Authors[1,2]

[1] Decipher, Blockchain Research Group at SNU
plasma-dag@decipher.ac
http://decipher.ac/plasma-dag
[2] Seoul National University, Seoul, Korea

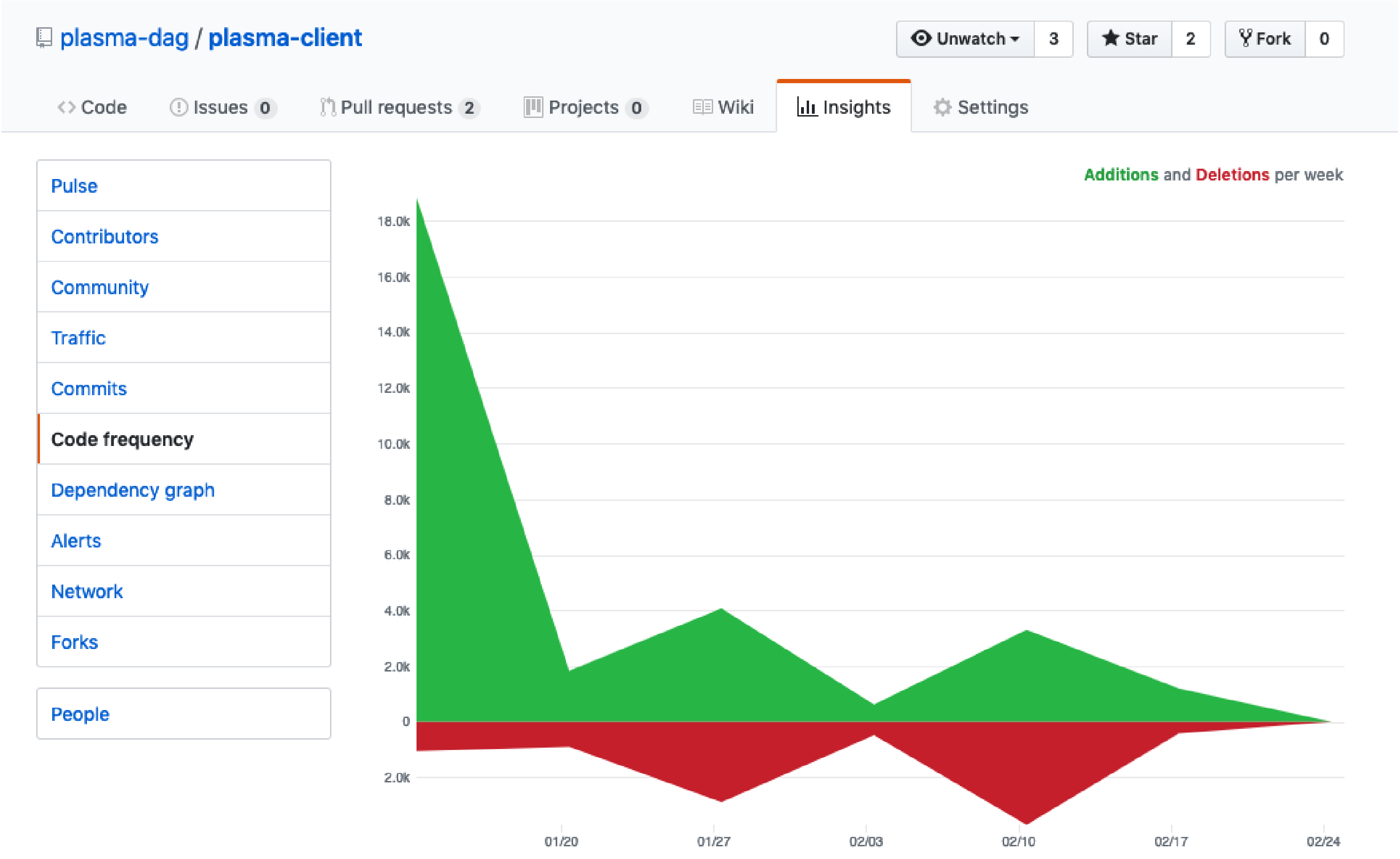
Abstract. Plasma is a solution to solve scalability problem of Ethereum by using a sidechain communicating with a smart contract deployed at public Ethereum without modifying protocols of Ethereum. There are several works designing and implementing Plasma such as Plasma MVP [1], Plasma Cash [2], Plasma Debit [3], Plasma XT [4], Plasma EVM [5], and so on. However, there are no perfect solutions to implement Plasma to exploit it as practical use yet. We propose a new design of Plasma that is Plasma DAG using Directed Acyclic Graph [6] as a data structure to reduce data size of fraud proof small enough to commit it to the Ethereum required when a data availability problem occurs. Furthermore, DAG structure relieves the monitoring cost for the participants in the Plasma chain to watch only the transactions related to each account of participant, instead of monitoring all the transactions. TODO: experimental results and conclusion

Keywords: Blockchain · Ethereum · Plasma · DAG · scalability · interchain.

1 Introduction

Since Satoshi Nakamoto published Bitcoin whitepaper [7], there have been various blockchain projects. Among them, Ethereum [8] is the most famous and stable blockchain project which handles not only value transfer, but also a small program named Smart Contract. Since Ethereum made a paradigm shift in blockchain world by introducing the smart contract, it soon became confronted with some limitations including performance issue (a.k.a. scalability issue in blockchain world). A lot of protocols like Sharding [9], DAG structure, sidechain [10], etc are proposed to solve the but none of them is realized due to some challenges in each solution; performance bottleneck and consistency issue of cross-shard communication in sharding, absence of the concrete consensus in DAG-like structure, etc. Because of this, sidechain solutions like Plasma are considered as the practical scaling solution at the moment to achieve the gain in a short time.

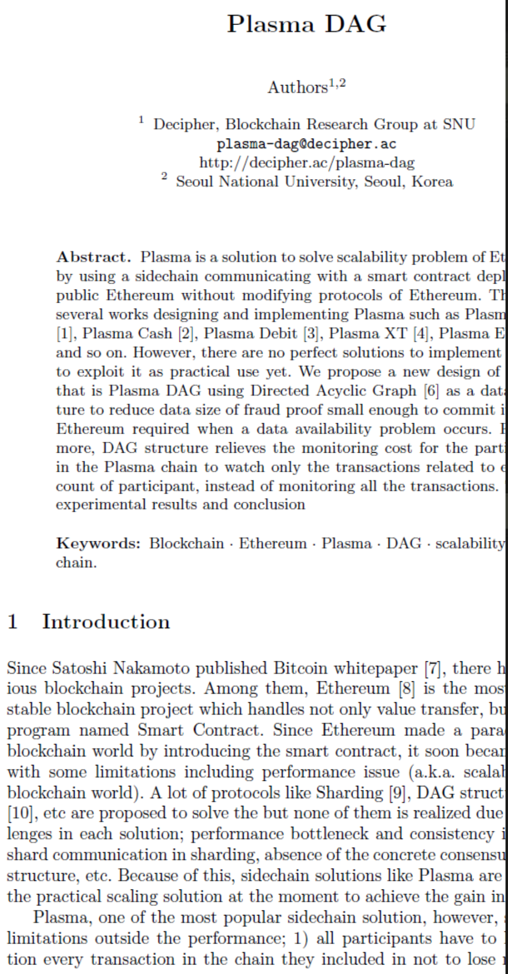
Plasma, one of the most popular sidechain solution, however, still has some limitations outside the performance; 1) all participants have to keep observation every transaction in the chain they included in not to lose money 2) DA



TODO(let someone another to fill this)

맺음말

백서 작성 및 구현 현황



Plasma DAG

Authors[1,2]

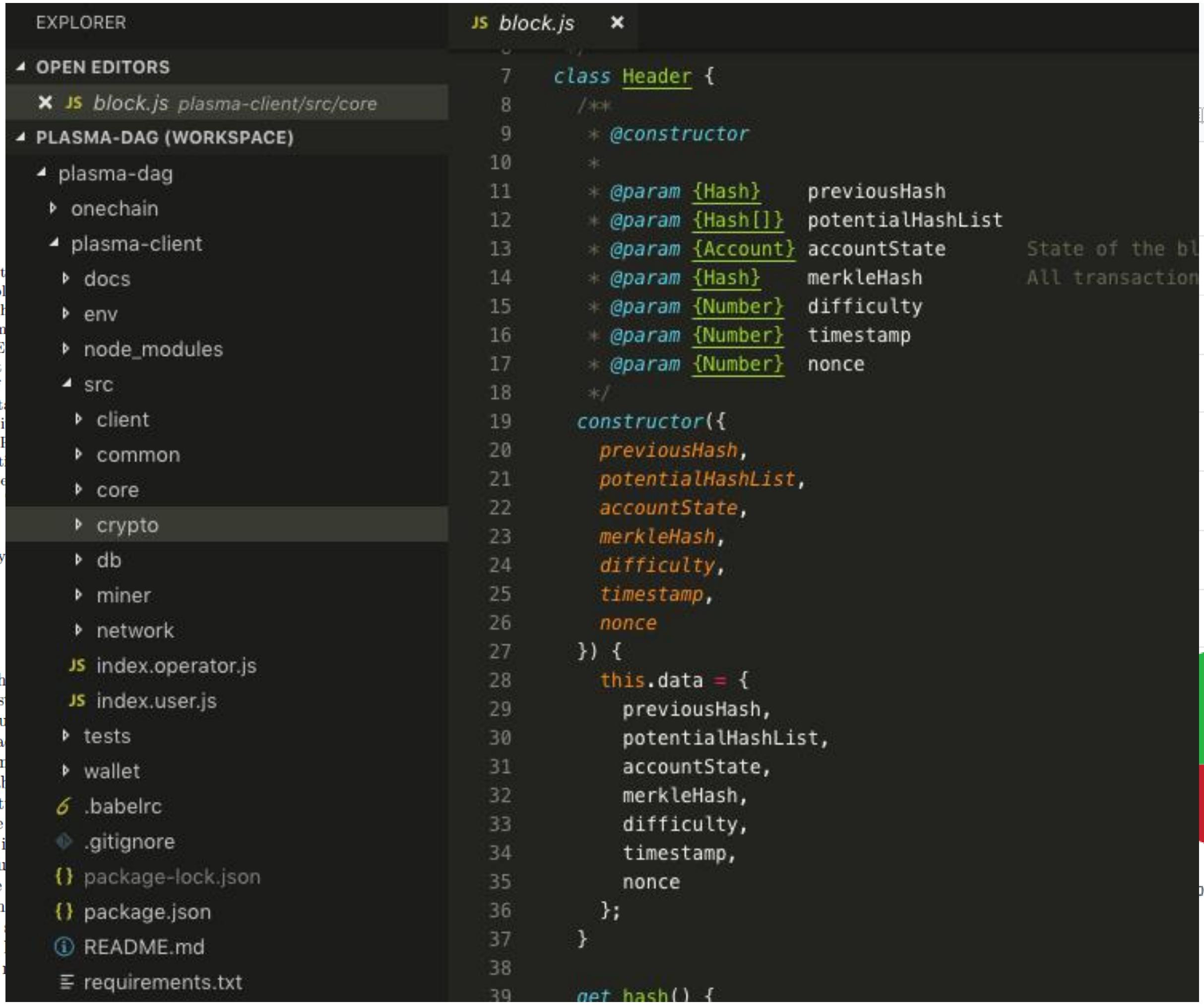
[1] Decipher, Blockchain Research Group at SNU
plasma-dag@decipher.ac
http://decipher.ac/plasma-dag
[2] Seoul National University, Seoul, Korea

Abstract. Plasma is a solution to solve scalability problem of Et
by using a sidechain communicating with a smart contract depl
public Ethereum without modifying protocols of Ethereum. Th
several works designing and implementing Plasma such as Plasm
[1], Plasma Cash [2], Plasma Debit [3], Plasma XT [4], Plasma E
and so on. However, there are no perfect solutions to implement
to exploit it as practical use yet. We propose a new design of
that is Plasma DAG using Directed Acyclic Graph [6] as a data
ture to reduce data size of fraud proof small enough to commit i
Ethereum required when a data availability problem occurs. F
more, DAG structure relieves the monitoring cost for the parti
in the Plasma chain to watch only the transactions related to e
count of participant, instead of monitoring all the transactions.
experimental results and conclusion

Keywords: Blockchain · Ethereum · Plasma · DAG · scalability
chain.

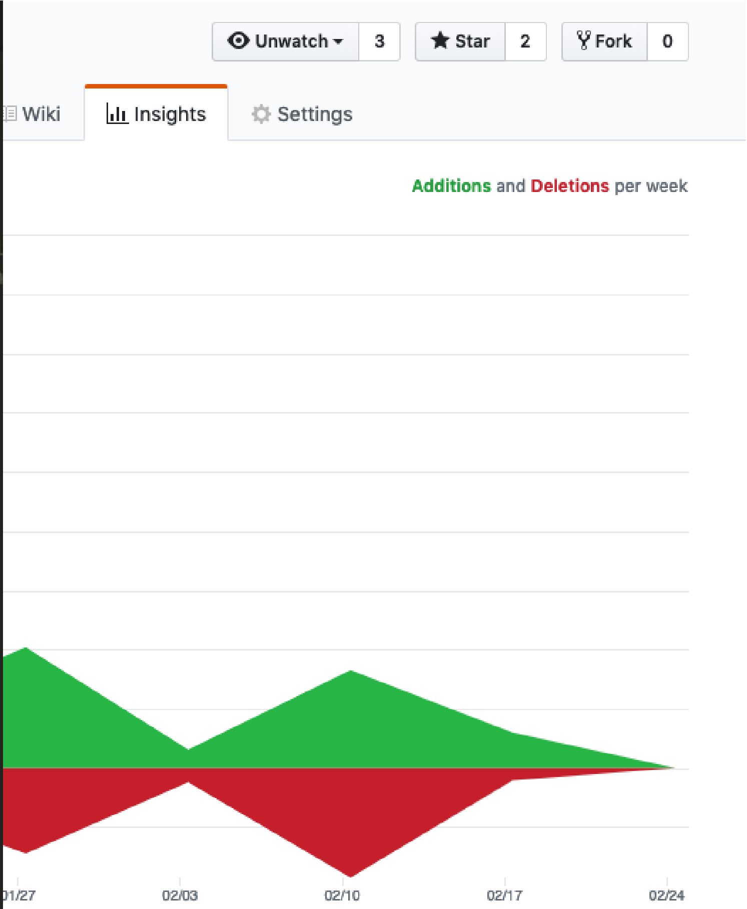
1 Introduction

Since Satoshi Nakamoto published Bitcoin whitepaper [7], there h
ious blockchain projects. Among them, Ethereum [8] is the mos
stable blockchain project which handles not only value transfer, bu
program named Smart Contract. Since Ethereum made a para
blockchain world by introducing the smart contract, it soon becar
with some limitations including performance issue (a.k.a. scalab
blockchain world). A lot of protocols like Sharding [9], DAG struct
[10], etc are proposed to solve the but none of them is realized due
lenges in each solution; performance bottleneck and consistency i
shard communication in sharding, absence of the concrete consensu
structure, etc. Because of this, sidechain solutions like Plasma are
the practical scaling solution at the moment to achieve the gain in
Plasma, one of the most popular sidechain solution, however,
limitations outside the performance; 1) all participants have to
tion every transaction in the chain they included in not to lose

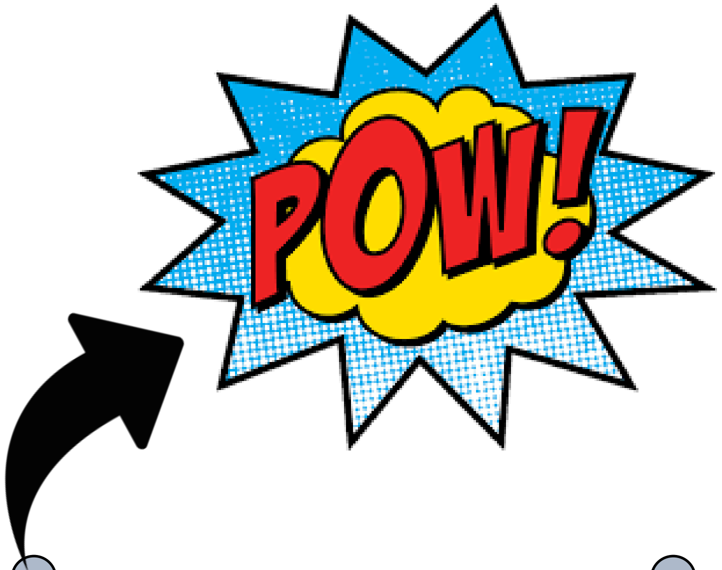


```
class Header {
  /**
   * @constructor
   *
   * @param {Hash}      previousHash
   * @param {Hash[]}    potentialHashList
   * @param {Account}   accountState         State of the bl
   * @param {Hash}      merkleHash           All transaction
   * @param {Number}    difficulty
   * @param {Number}    timestamp
   * @param {Number}    nonce
   */
  constructor({
    previousHash,
    potentialHashList,
    accountState,
    merkleHash,
    difficulty,
    timestamp,
    nonce
  }) {
    this.data = {
      previousHash,
      potentialHashList,
      accountState,
      merkleHash,
      difficulty,
      timestamp,
      nonce
    };
  }

  get hash() {
```



마일스톤



2019

1Q
- Whitepaper 초안 공개
- Github 코드 공개
- Test Network

2Q
- Zero-Knowledge Proof를 이용한 sum check 디자인 및 구현
- Application 구현

3Q
- 이더리움 메인 네트워크에 업로드하는 commitment의 용량 압축 구현 및 테스트

4Q
- 누구나 쓸 수 있게 서비스 공개
- 기업용 솔루션 제공

References

1. Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
2. LeMahieu, Colin. "Nano: A feeless distributed cryptocurrency network." Nano [Online resource]. URL: https://nano. org/en/whitepaper (date of access: 24.03. 2018) (2018).
3. Poon, Joseph, and Vitalik Buterin."Plasma: Scalable Autonomous Smart Contracts" (2017).
4. https://ethresear.ch/t/minimal-viable-plasma/426. (2018).
5. https://ethresear.ch/t/plasma-cash-plasma-with-much-less-per-user-data-checking/1298. (2018).
6. https://ethresear.ch/t/plasma-debit-arbitrary-denomination-payments-in-plasma-cash/2198. (2018).
7. https://ethresear.ch/t/plasma-xt-plasma-cash-with-much-less-per-user-data-checking/1926. (2018).
8. https://ethresear.ch/t/plasma-evm-2-0-state-enforceable-construction/3025. (2018).
9. https://ethresear.ch/t/quark-gluon-plasma-verified-plasma-chain-without-confirmation-signatures/3453. (2018)
10. https://medium.com/matter-labs/introducing-matter-testnet-502fab5a6f17. (2019)
11. https://www.learnplasma.org/en/