L1/L2 Solutions의 고민들

2018-10-04
KAIST KI Colloquium

이종협

Scalability

Decentralization

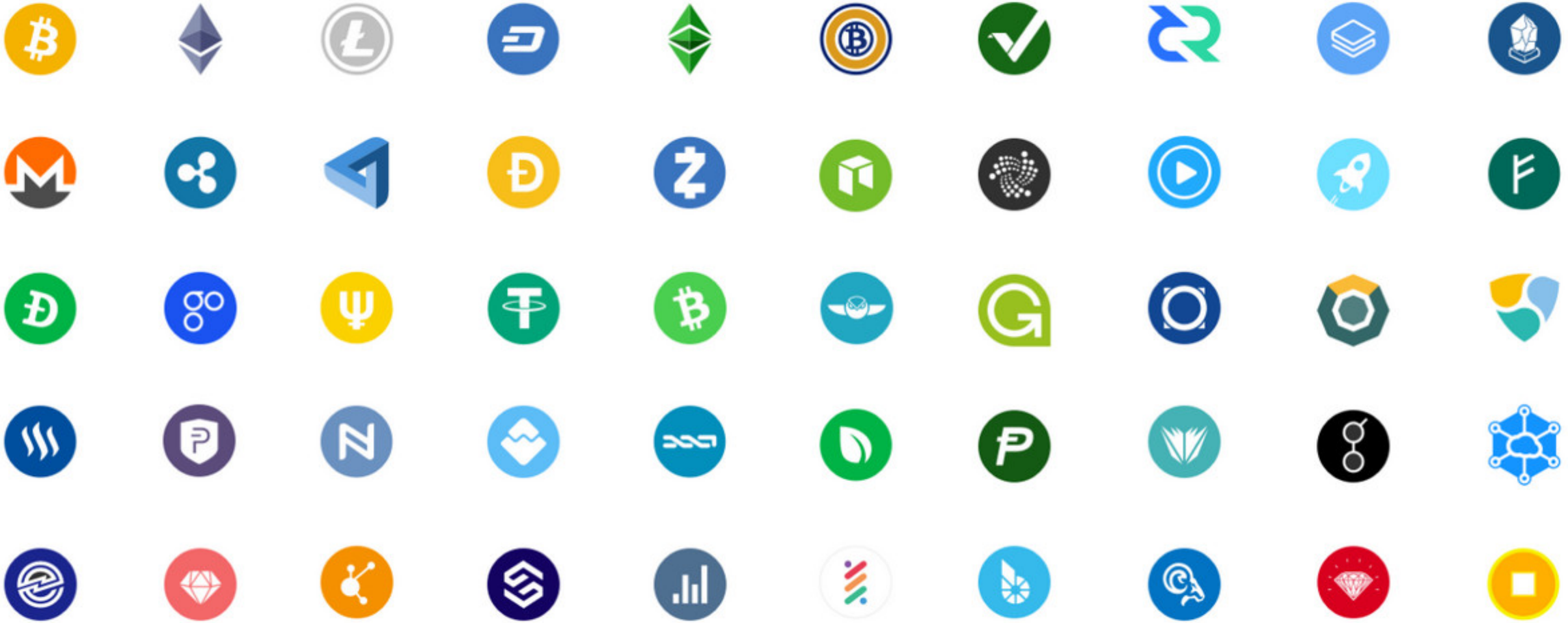
Safety

모두 중요하지만,
무엇이 당장 발목을 잡고 있는가?

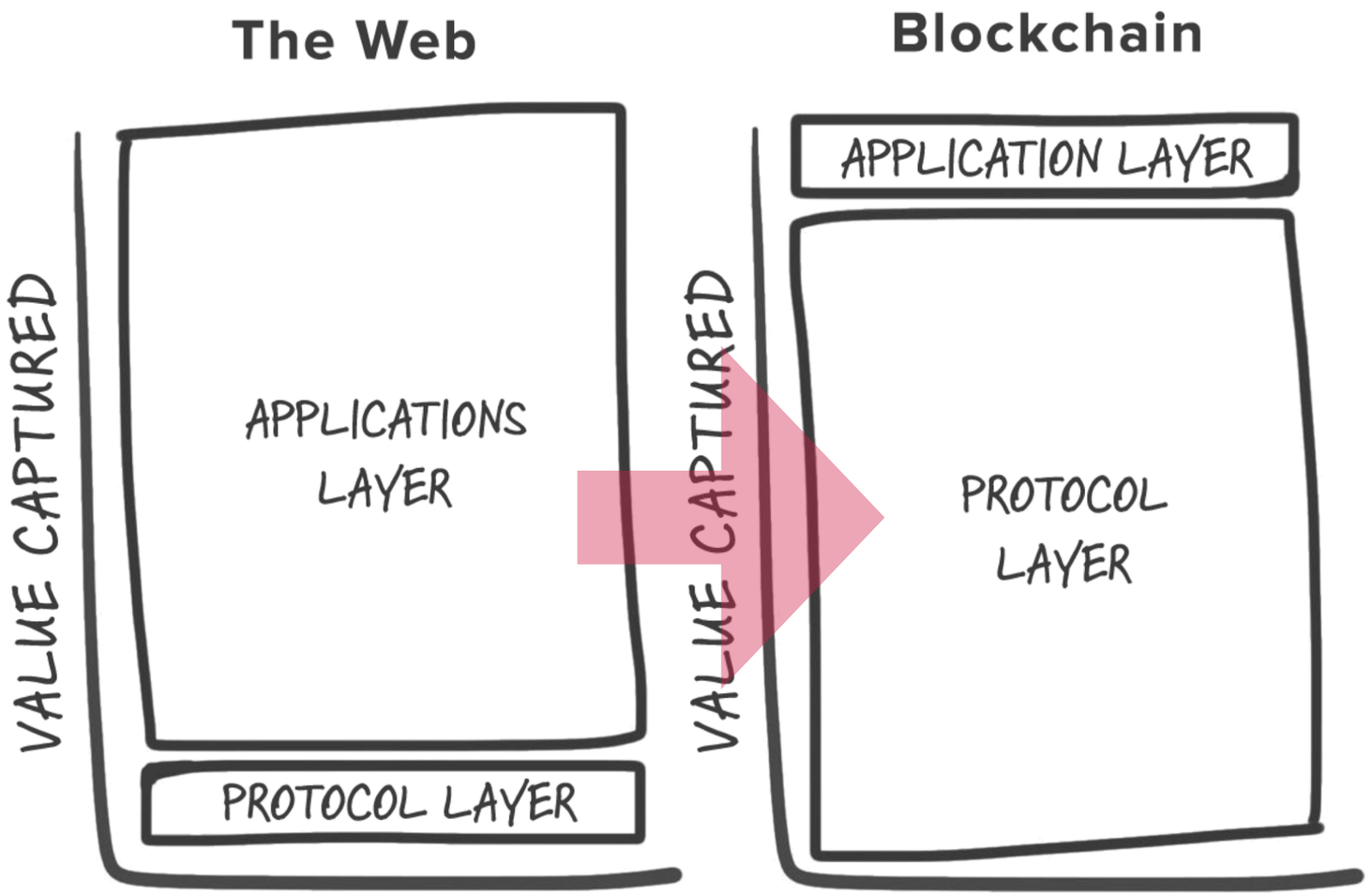
우선, **Scalability.**

“Protocol Fever”

새로운 메인넷과 프로토콜은 왜 자꾸 나타나는가?



“Fat Protocols”



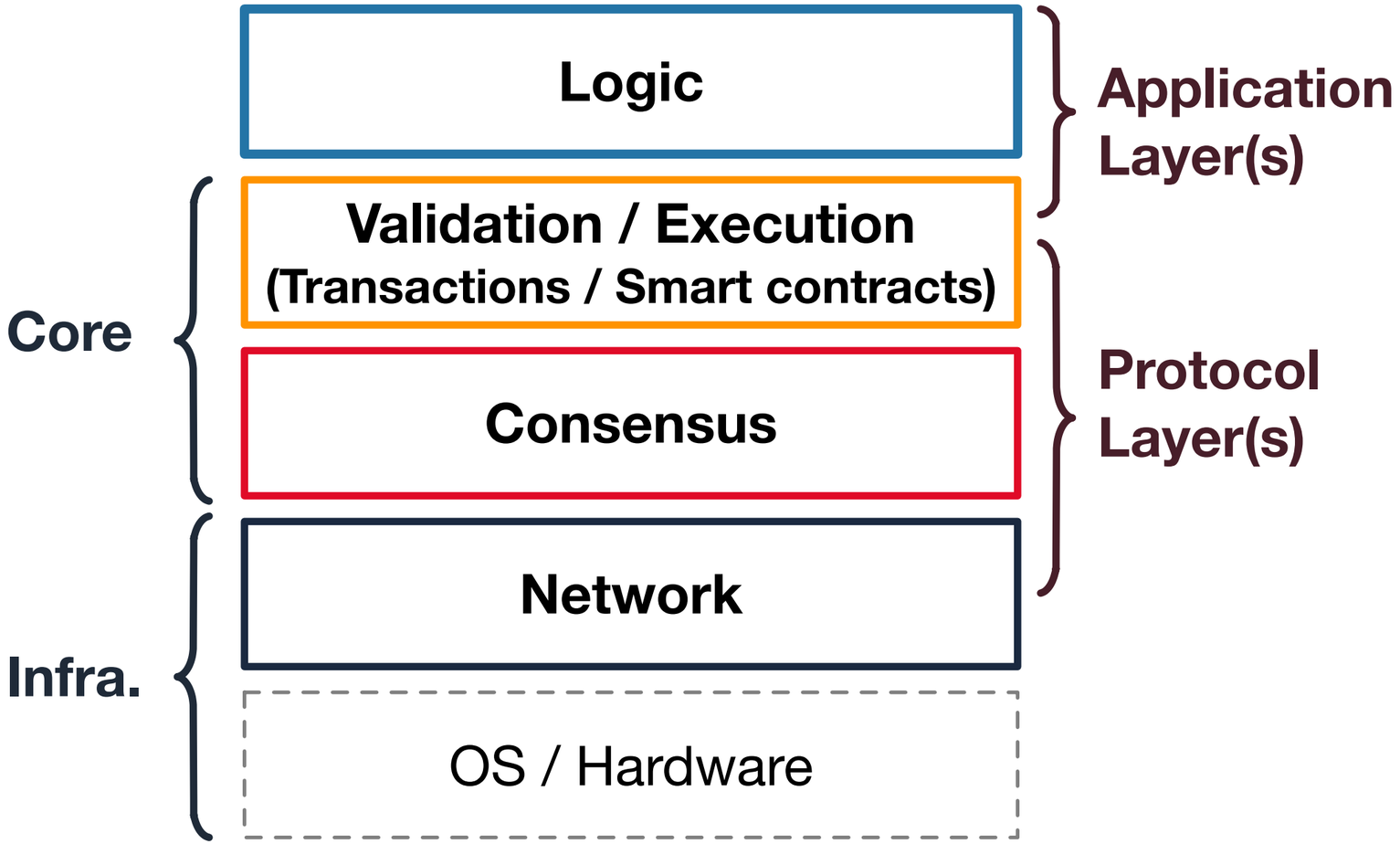
Protocol에서의
기능의 정의/구현이
중심이다.

새로운 기능은
어디서 구현해야 하는가?

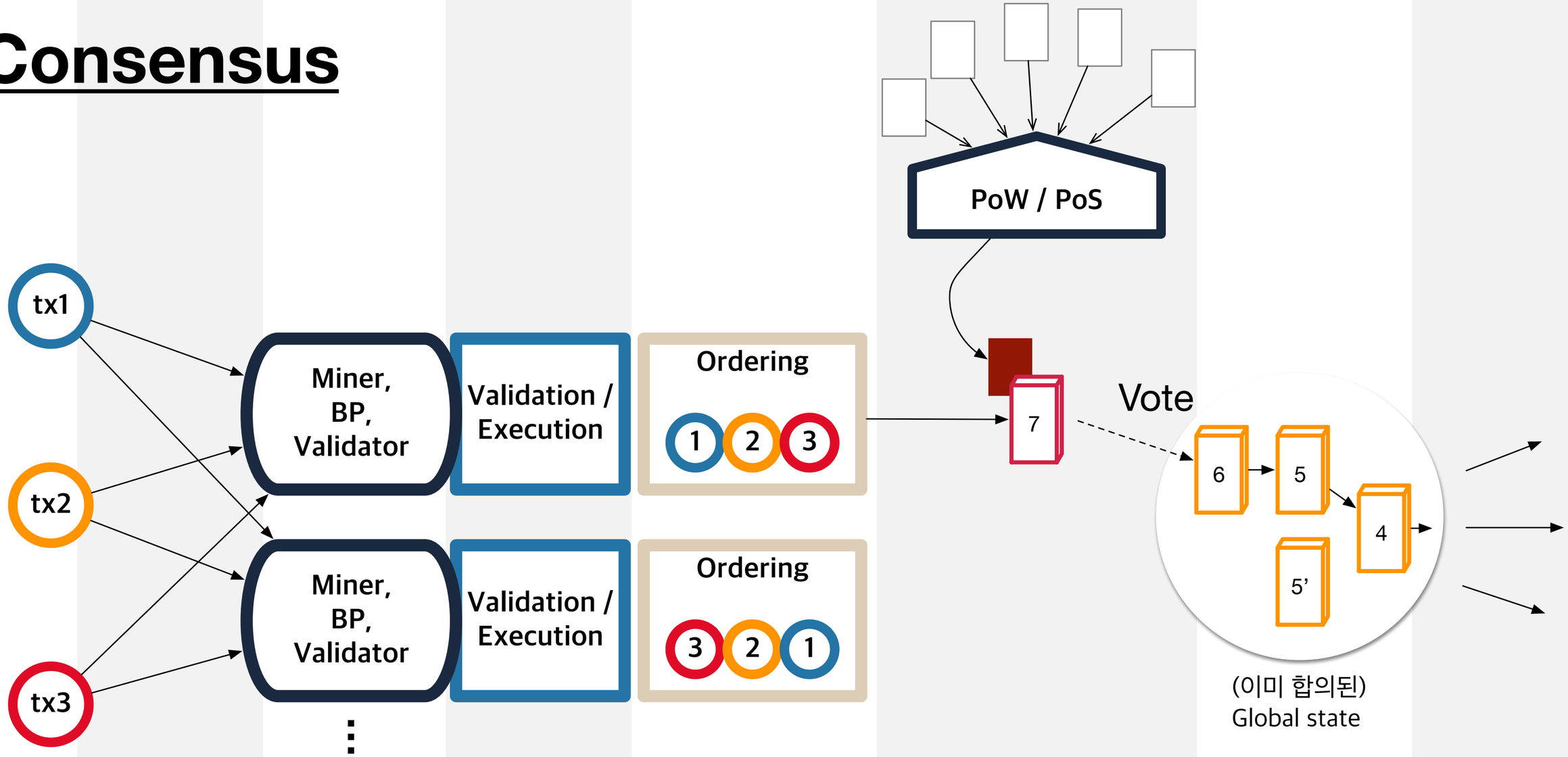
Decentralization은?

(From: www.usv.com/blog/fat-protocols)

Blockchain Architecture (stack)



Consensus



Network propagation

Rule이 명확한 과정

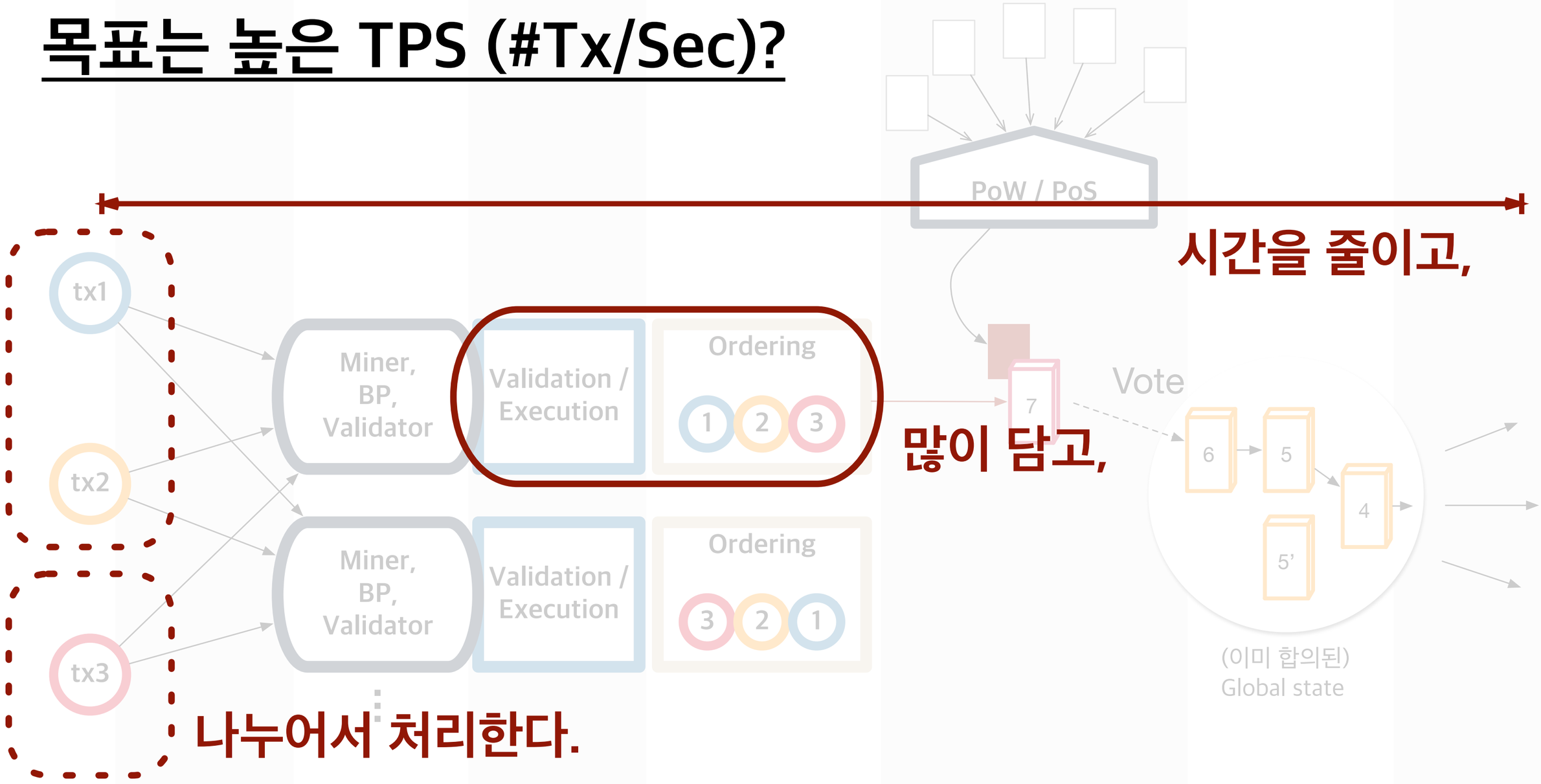
각자의 View에 따른 차이

Decentralization + Anonymity가 중요하다면

Consensus

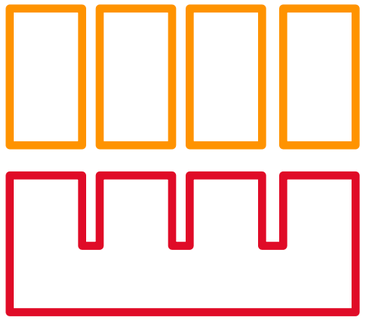
결과 전파 (Publish)

목표는 높은 TPS (#Tx/Sec)?



L1 Solutions

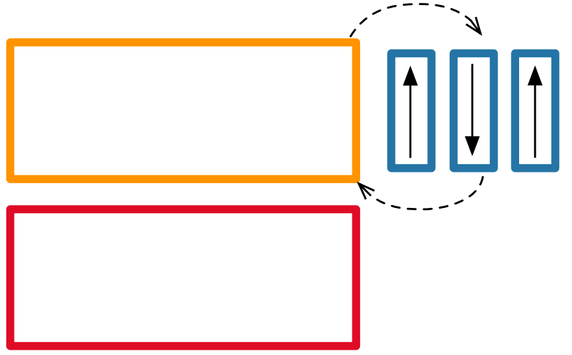
Reparametization
(tuning)



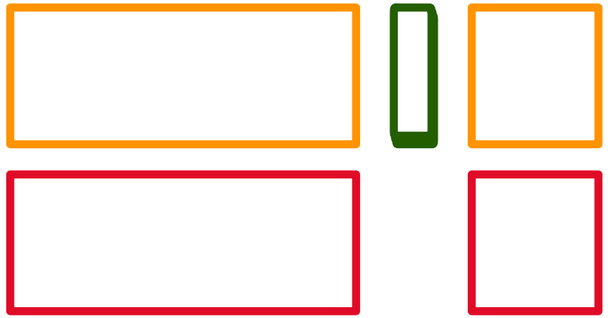
Sharding

Network layer는?

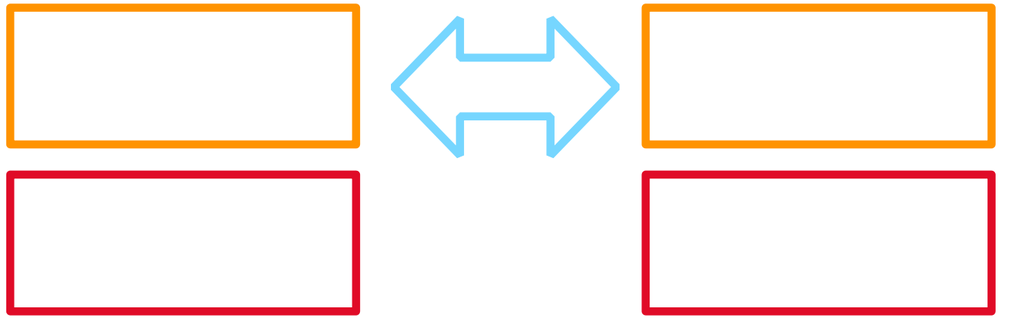
L2 Solutions



State Channels



Sidechains



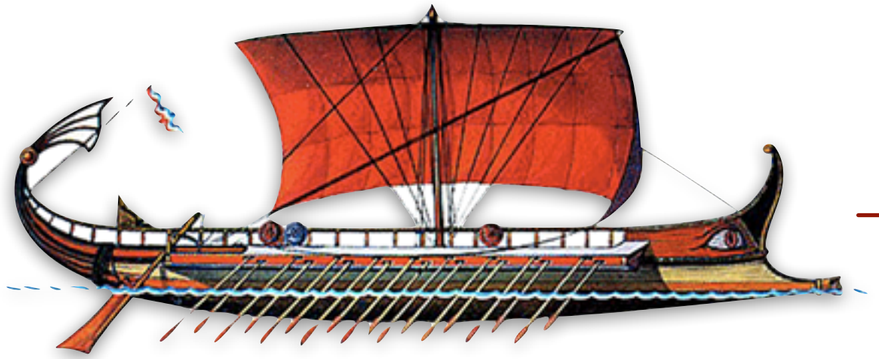
Inter-blockchain

The Unanswered Question

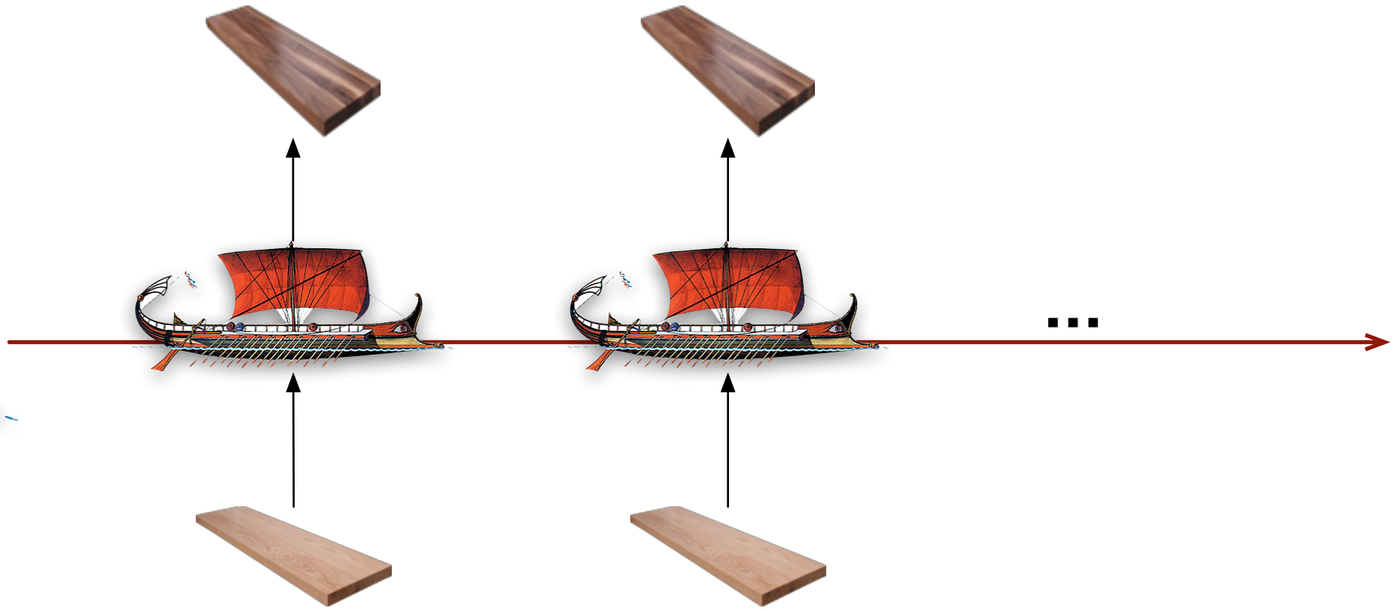
블록체인이란 무엇인가?

블록체인이란 무엇인가?

(당신의 블록체인은 잘못되었다.)



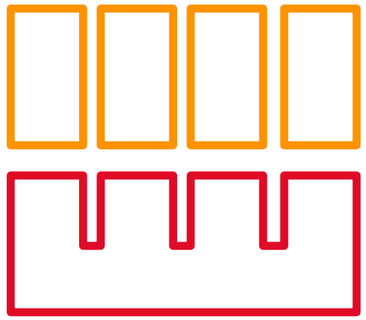
Theseus' Paradox



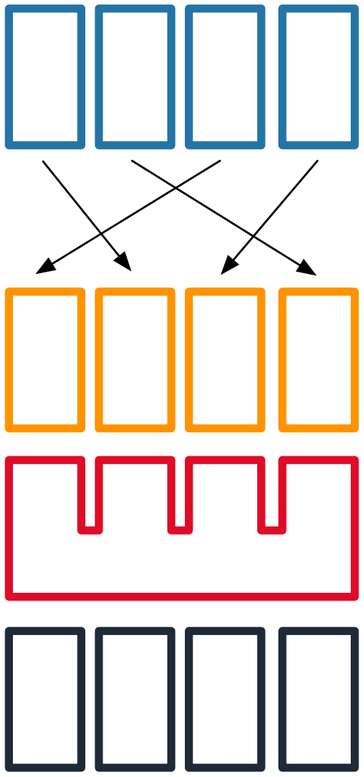
무엇을 가정하고, **어디에 가치를 두는가?**
(혁신은 어느 곳에 있어야 하는가?)

L1 Solution - Sharding

Hard fork가 필요 (Decentralized 환경에서 쉽게 바꿀 수 있는가?)

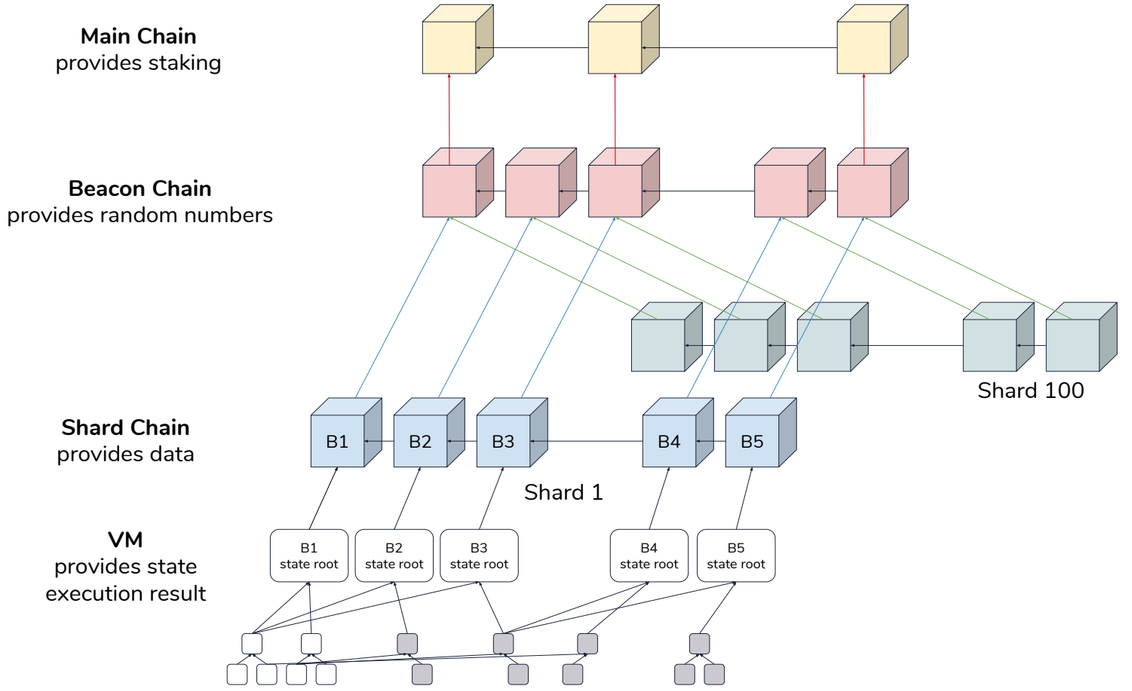


Shard takeover?
Cross-shard communications?



Random Selection

State는?

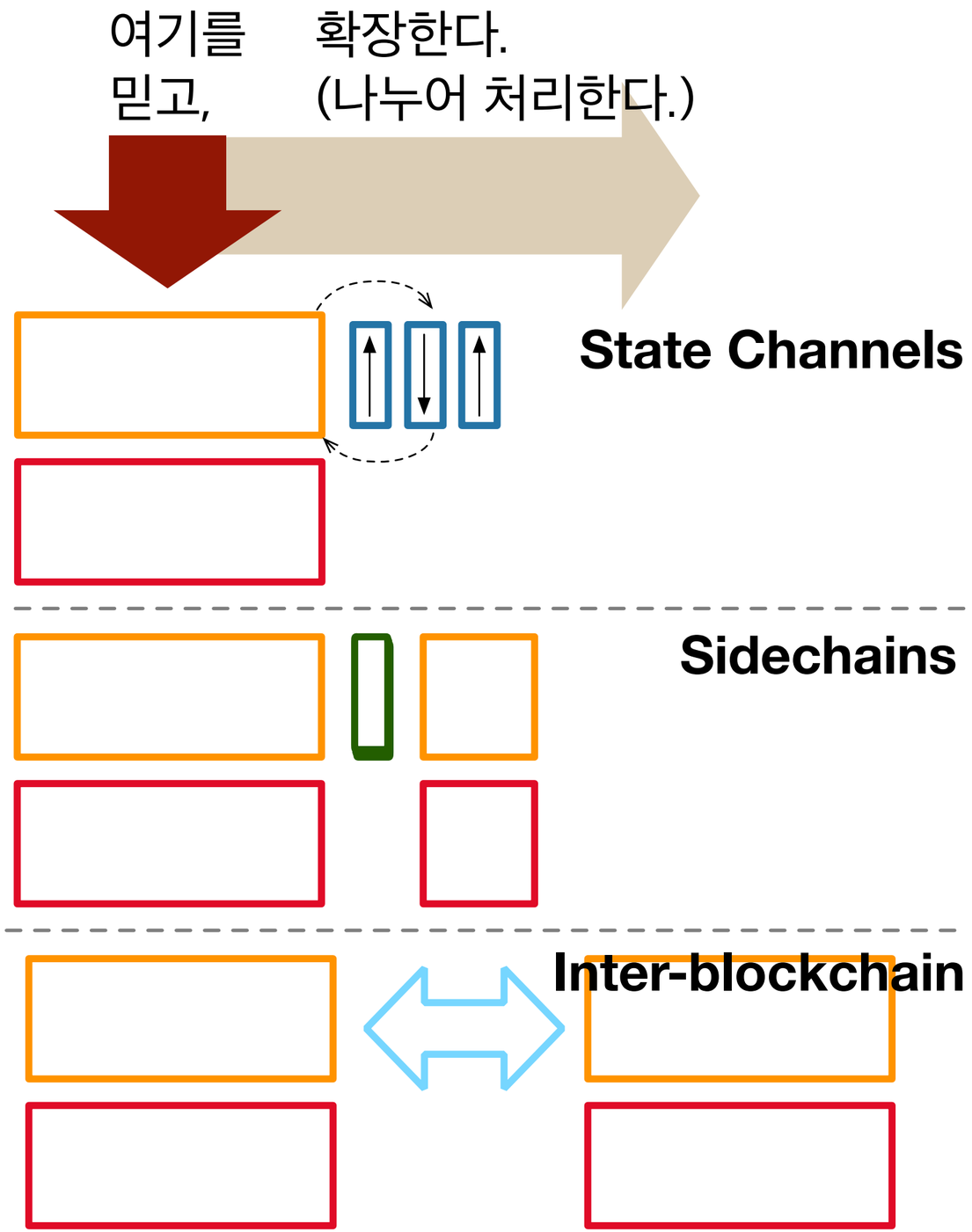


L2 Solutions

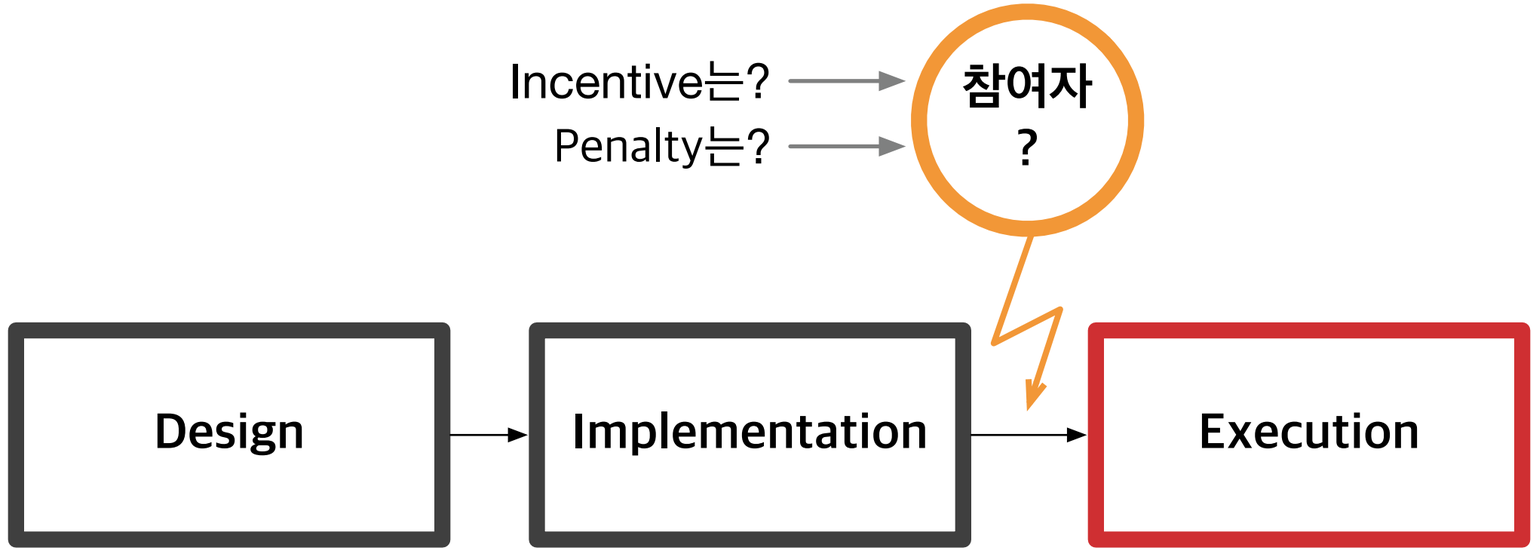
어떠한 것을 L1-chain에서 옮기는가?
어떠한 것을 L1-chain에 남겨놓는가?

자신만의 Rule을 가진다는 것의 의미는 무엇인가?
별도의 chain에서 (또는 offchain에서) 하고자 하는 것은 무엇인가?

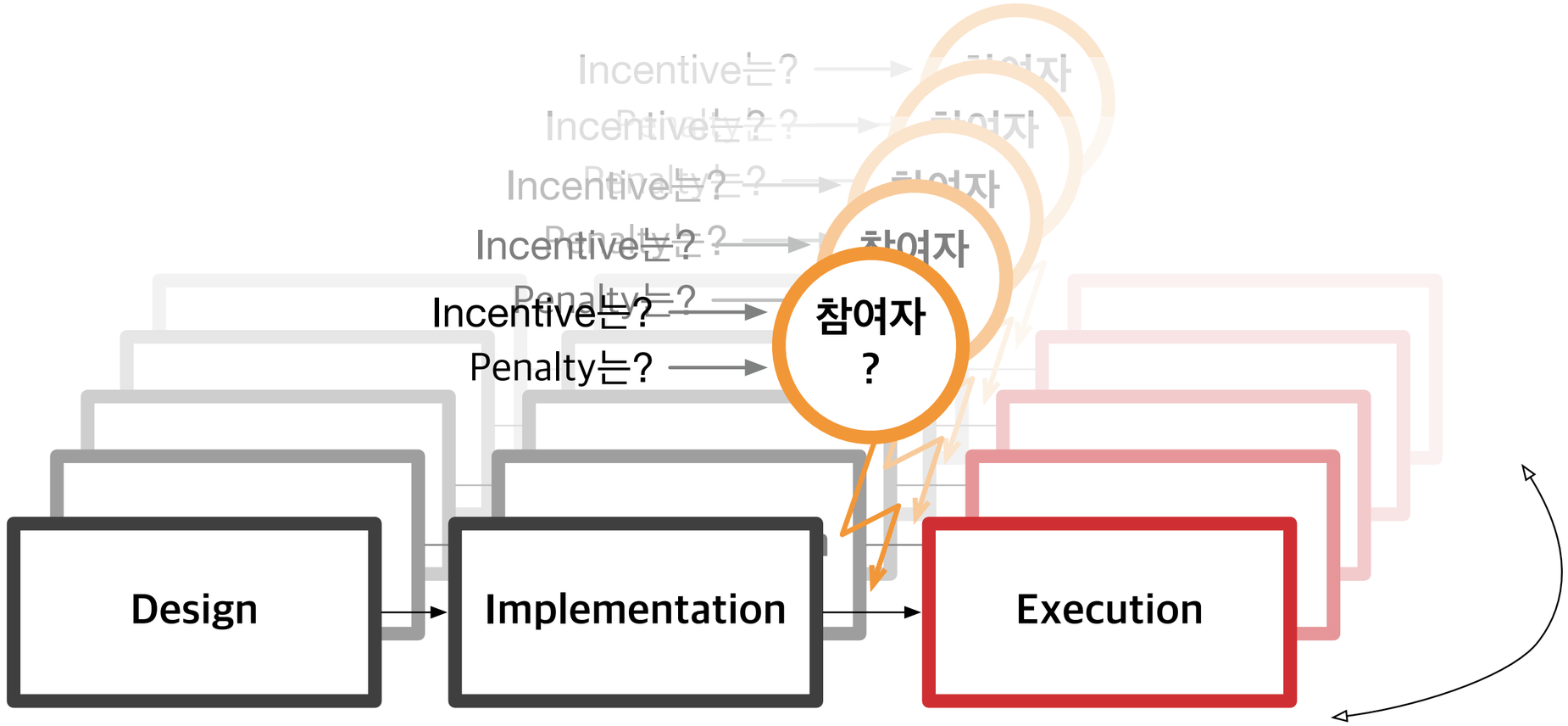
Cryptoeconomy는 어떠한 역할을 하는가?



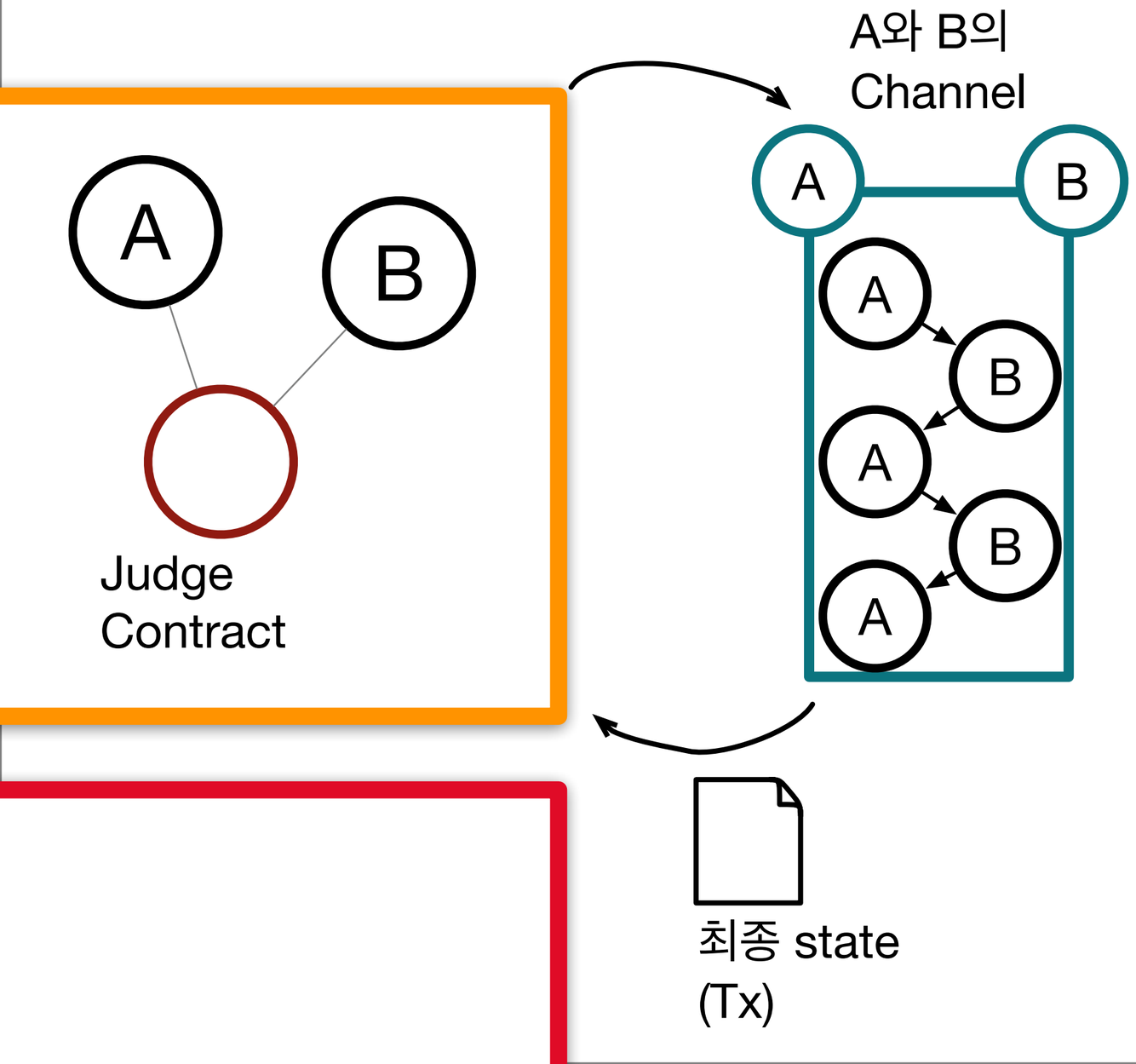
“L2 solutions are cryptoeconomic solutions”



“L2 solutions are cryptoeconomic solutions”



State Channels



목표는?

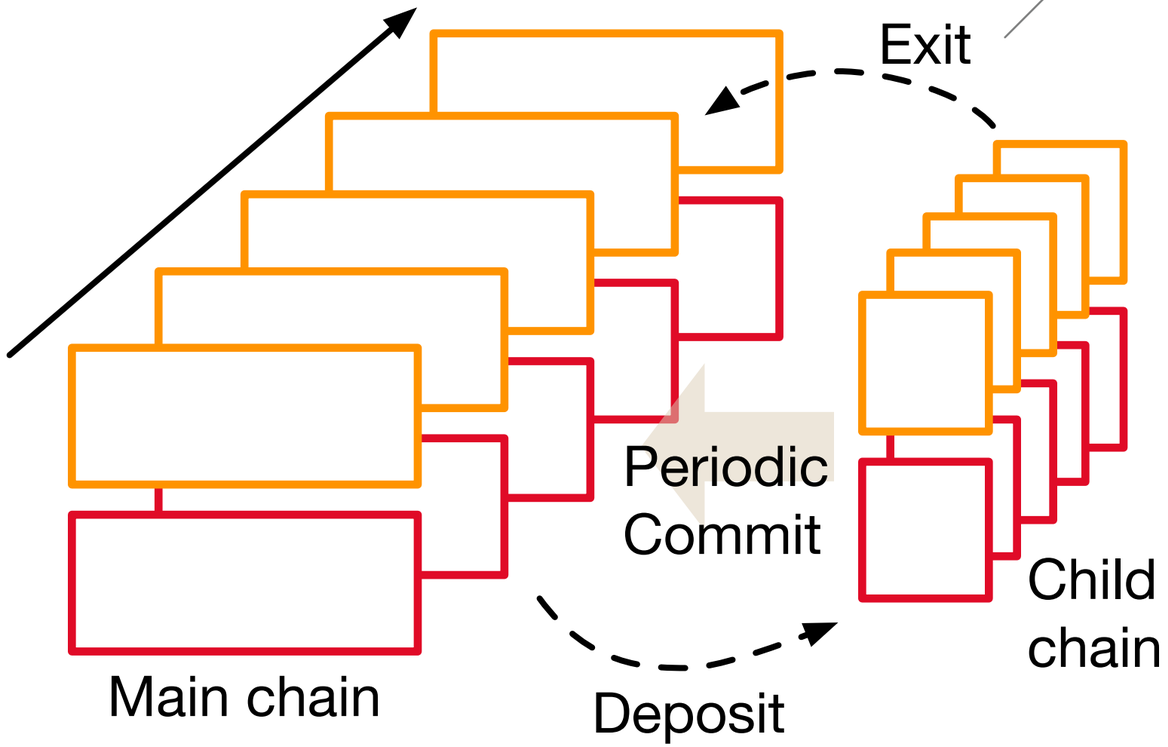
L1-chain의 transaction을 최소화 한다.
Instant finality
(Anonymity)

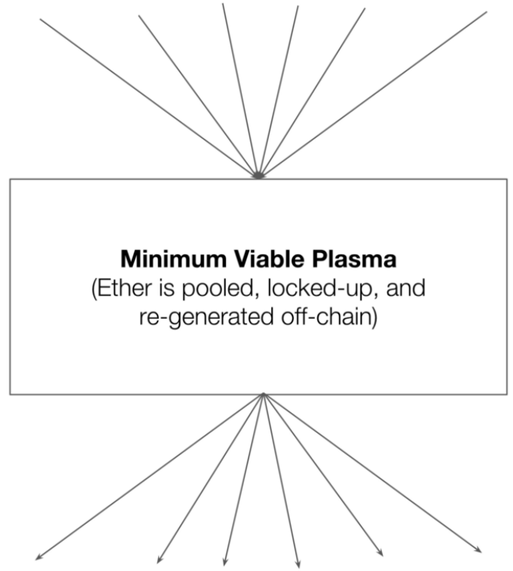
어디에 가치를 두는가?

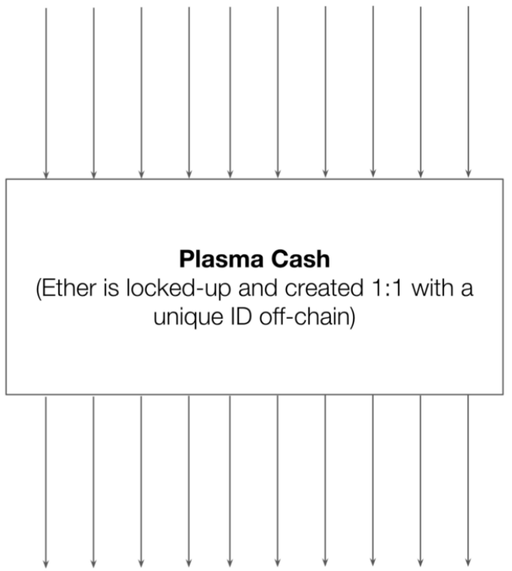
Payment	Lightning, Raiden
	Trinity, Liquidity
	⋮
General state	Celer
	Counterfactual, Perun

Plasma

Challenge 과정을 통해서 Exit을 보장한다.

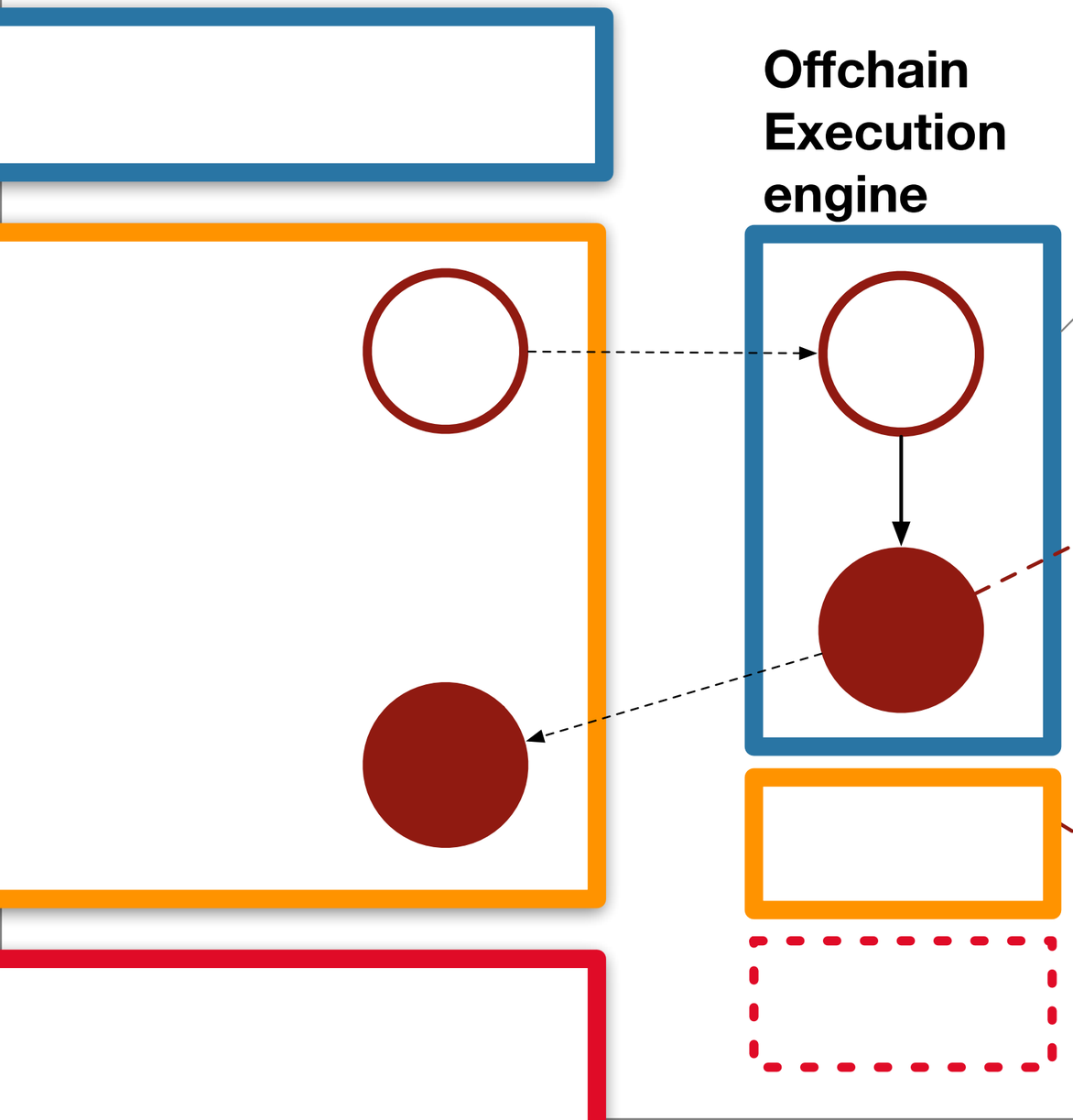






Plasma MVP, Cash, Debit, EVM, …

Offchain computation



Offchain Execution engine

일종의 delegated execution

최종 결과를 믿을 수 있는가?
(확인해야 하는가? → Verifier's dilemma)

Interactive verification game
(Bisection protocol)
e.g, TrueBit, Arbitrum

Execution engine (computing node) 자체를 믿을 수 있는가?

SGX, zkProof
e.g., Ekiden

각각 해결해야 할 문제들은 아직 많습니다.

게다가 남은 unanswered question들이 있습니다.

⋮

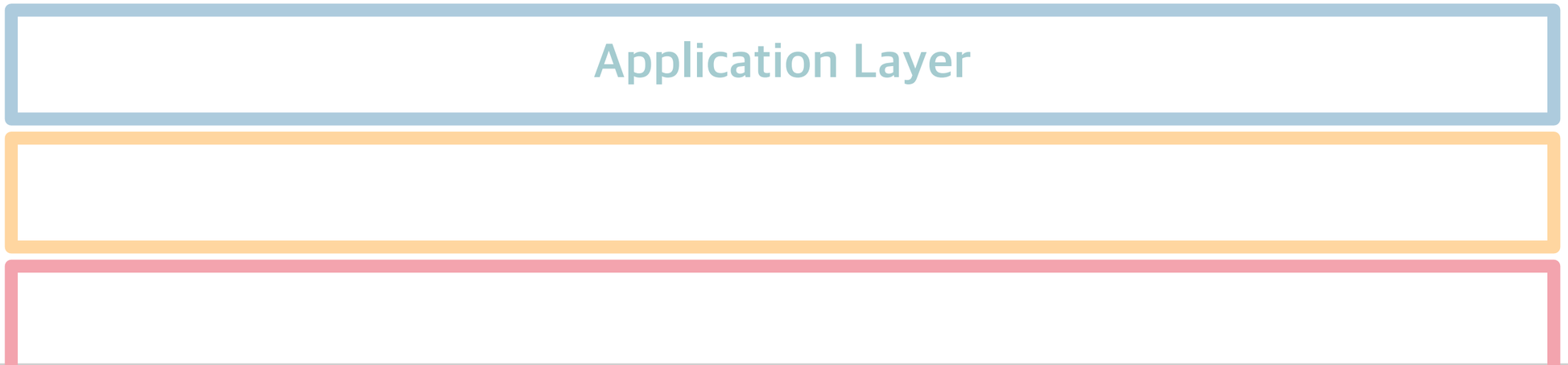
각각 해결해야 할 문제들은 아직 많습니다.
게다가 남은 unanswered question들이 있습니다.

⋮

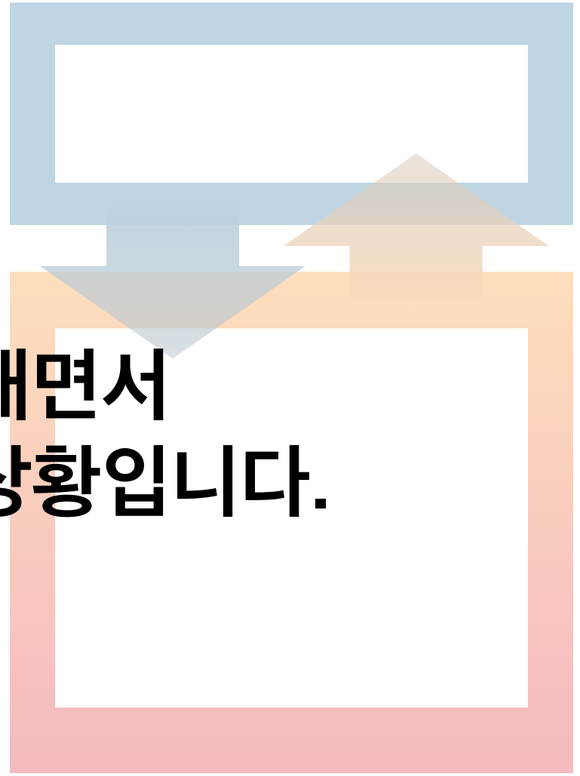
그래서 블록체인은 어디에 쓰는가?

블록체인의 Killer app은 무엇인가?

블록체인은 실생활의 무엇을 바꿀 수 있는가?



서로가 나름의 **혁신을 만들어 내면서**
상대방의 혁신을 도와야 하는 상황입니다.



Application layer의 고민

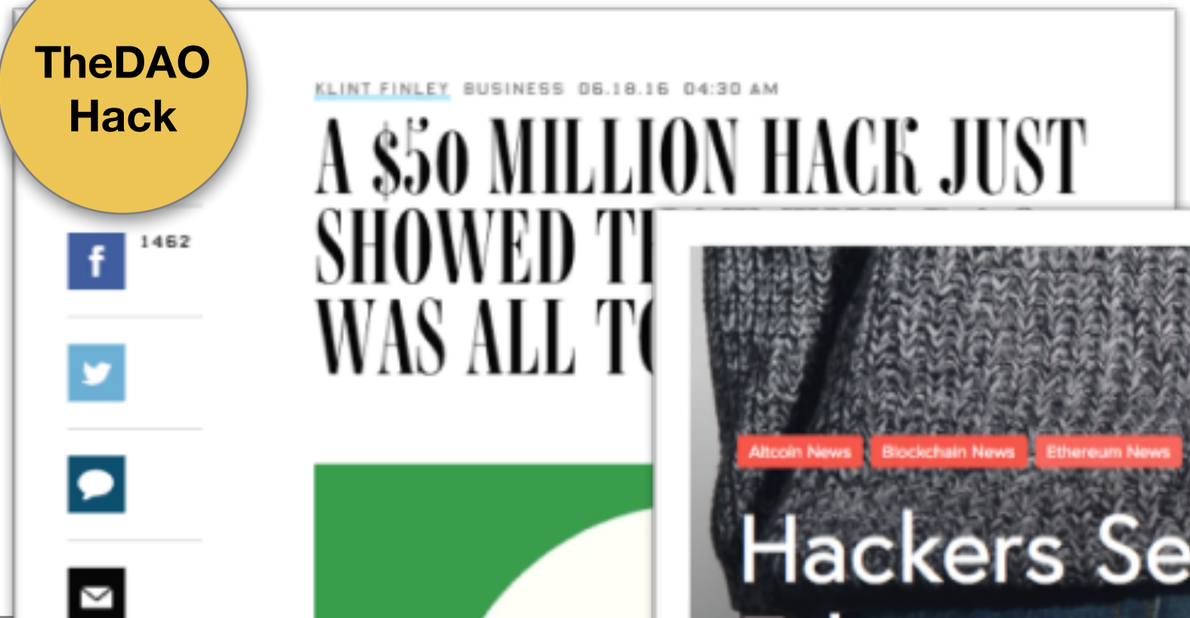
Application Layer

당장 발목을 잡는 것은 무엇인가?

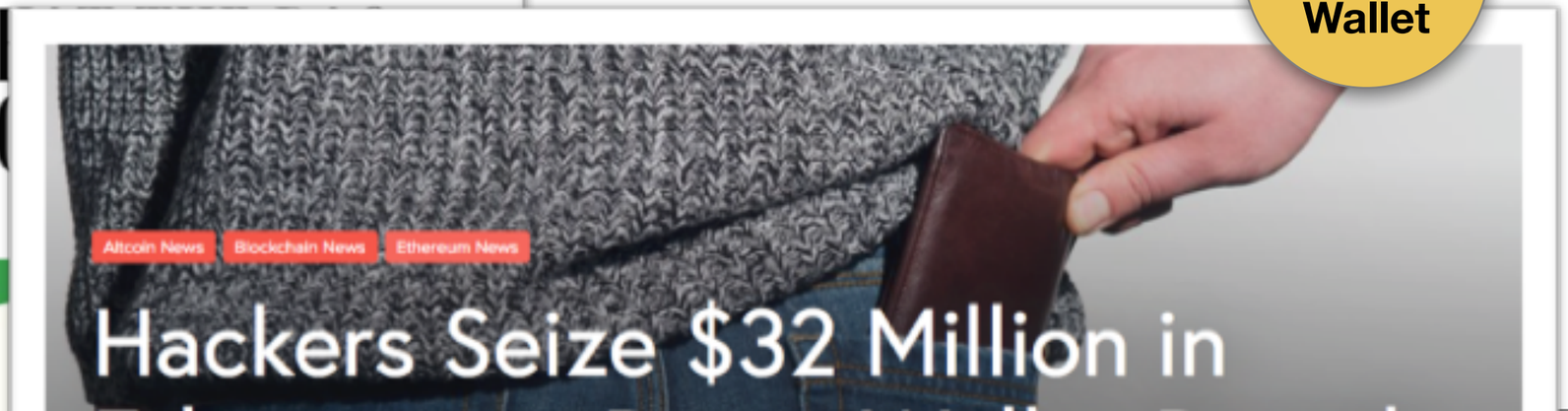
Design pattern은 어떠한가?

기존의 프로그램 모델은 괜찮은가?

TheDAO Hack



Parity MultiSig Wallet



New Programming Languages

(From Vitalik Buterin's tweet)

"Mainstream 언어는 적합하지 않다." → Things contracts require that regular code does not:

* Very small code size
* Much higher focus on safety
* Much higher focus on auditability (misleading code very bad)
* Perfect determinism

C++ (EOS)

```
void add_balance( account_name payer, account_name to, uint64_t q ) {
   auto toitr = _accounts.find( to );
   if( toitr == _accounts.end() ) {
     _accounts.emplace( payer, [&]( auto& a ) {
        a.owner = to;
        a.balance = q;
     });
   } else {
     _accounts.modify( toitr, 0, [&]( auto& a ) {
        a.balance += q;
        eosio_assert( a.balance >= q, "overflow detected" );
     });
   }
}
```

```
void transfer( account_name from, account_name to, uint64_t quantity ) {
   require_auth( from );

   const auto& fromacnt = _accounts.get( from );
   eosio_assert( fromacnt.balance >= quantity, "overdrawn balance" );
   _accounts.modify( fromacnt, from, [&]( auto& a ){ a.balance -= quantity; } );

   add_balance( from, to, quantity );
}
```

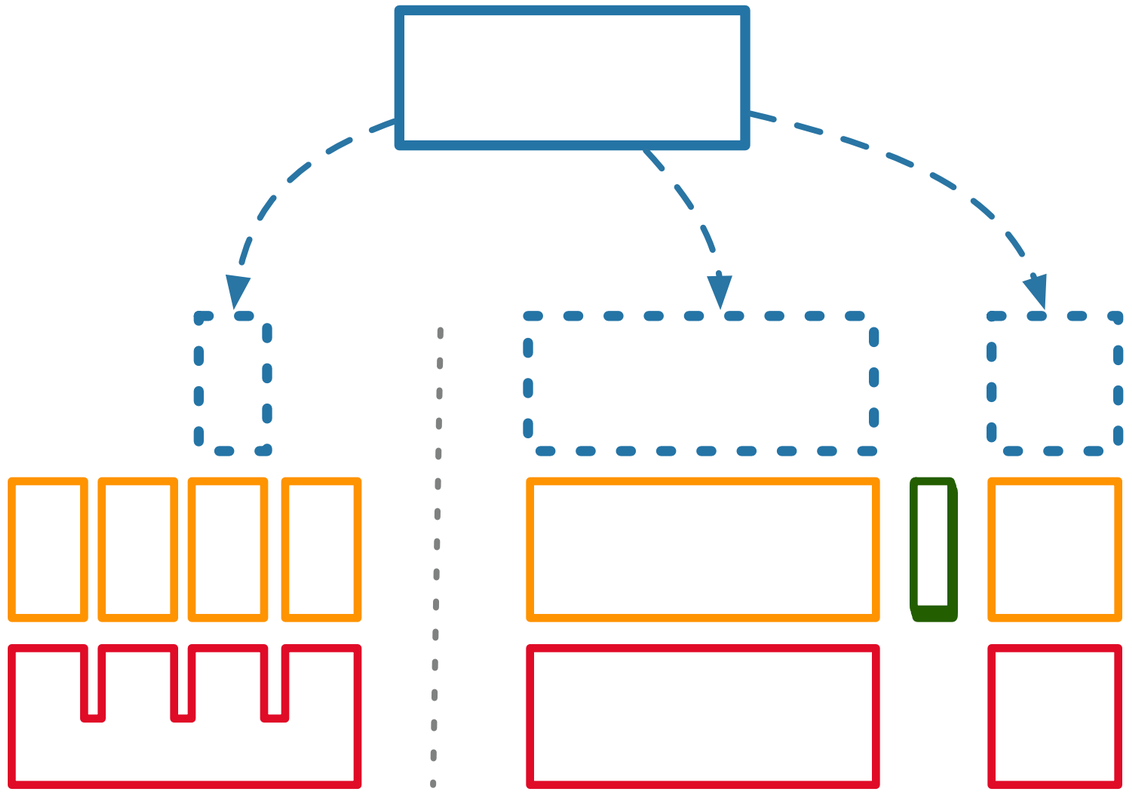
Vyper (Ethereum)

```
@public
def transfer(_to : address, _value : uint256(wei)) -> bool:
    _sender: address = msg.sender
    # Make sure sufficient funds are present implicitly through overflow protection
    self.balances[_sender] = self.balances[_sender] - _value
    self.balances[_to] = self.balances[_to] + _value
    # Fire transfer event
    log.Transfer(_sender, _to, _value)
    return True
```

Bamboo, Babbage, Liquidity,
Michelson, OWL, Plutus
Rholang, Scilla, Simplicity
Solidity, Typecoin, Vyper
…

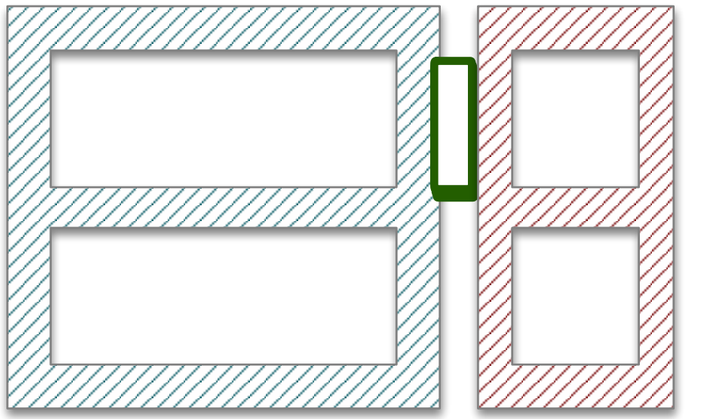
서로가 혁신을 만들 수 있는 판(platform)은 무엇인가?

무엇이 제일 잘 맞는가?



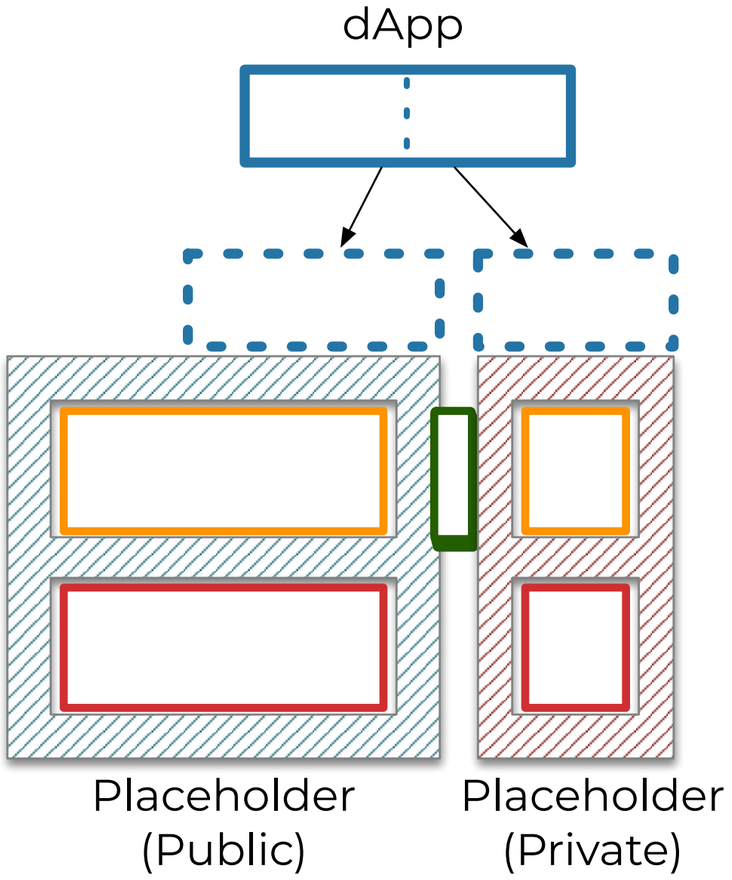
어떤 역할이 필요한가?

“어떠한 메인넷/프로토콜이 가장 혁신적인가?”



“어떠한 메인넷/프로토콜이 가장 혁신적인가?”

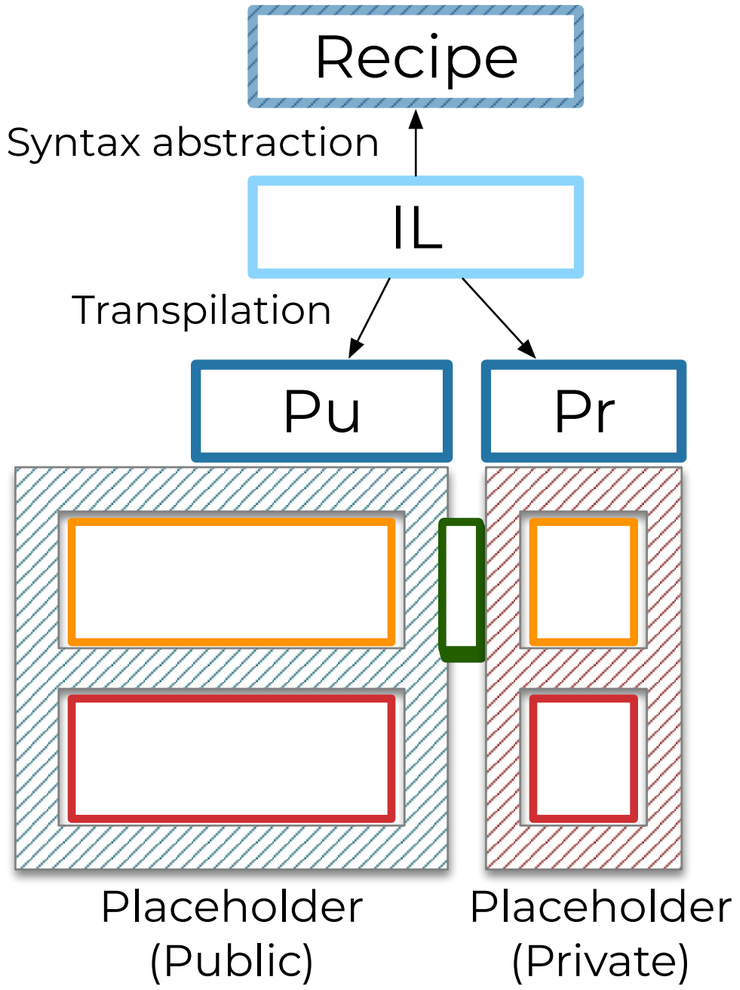
“Public/Private blockchain의 역설적 요구사항은 어떻게 해결해야 하는가?”



“어떠한 메인넷/프로토콜이 가장 혁신적인가?”

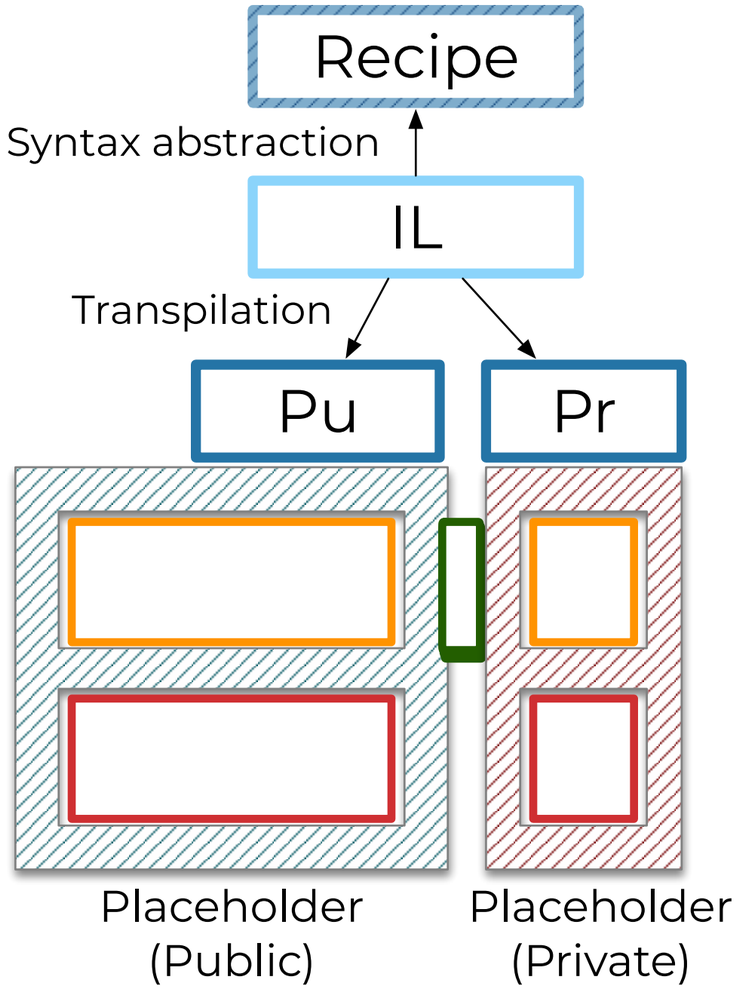
“Public/Private blockchain의 역설적 요구사항은 어떻게 해결해야 하는가?”

“Application layer가 올바르고 효과적으로 적응하기 위해서는?”



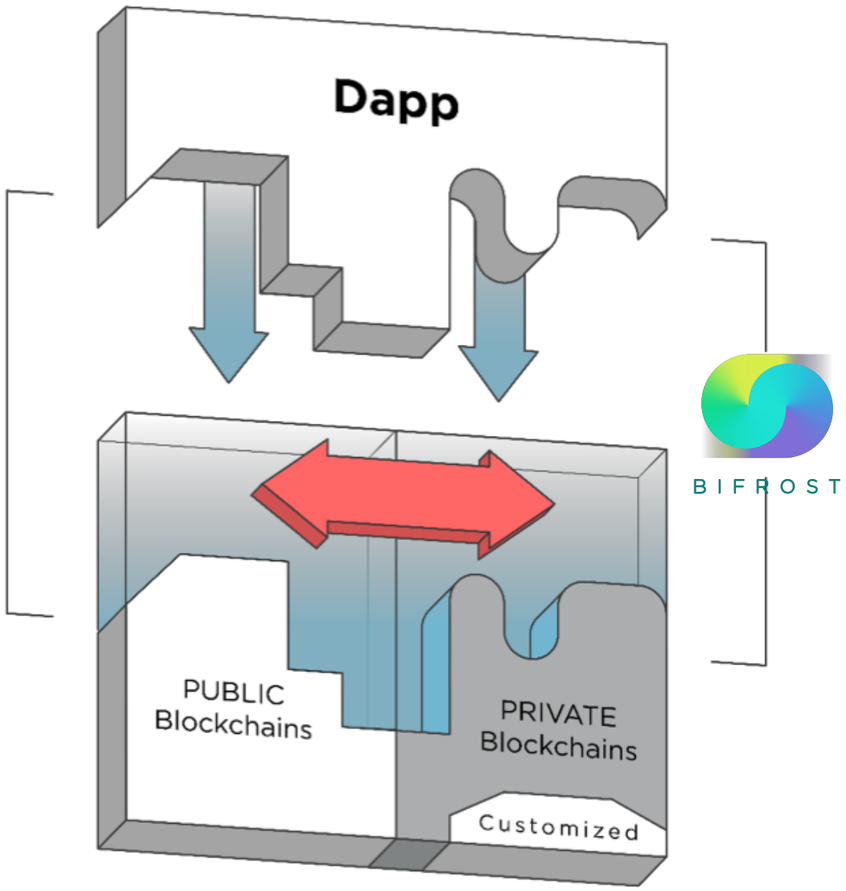
BIFROST

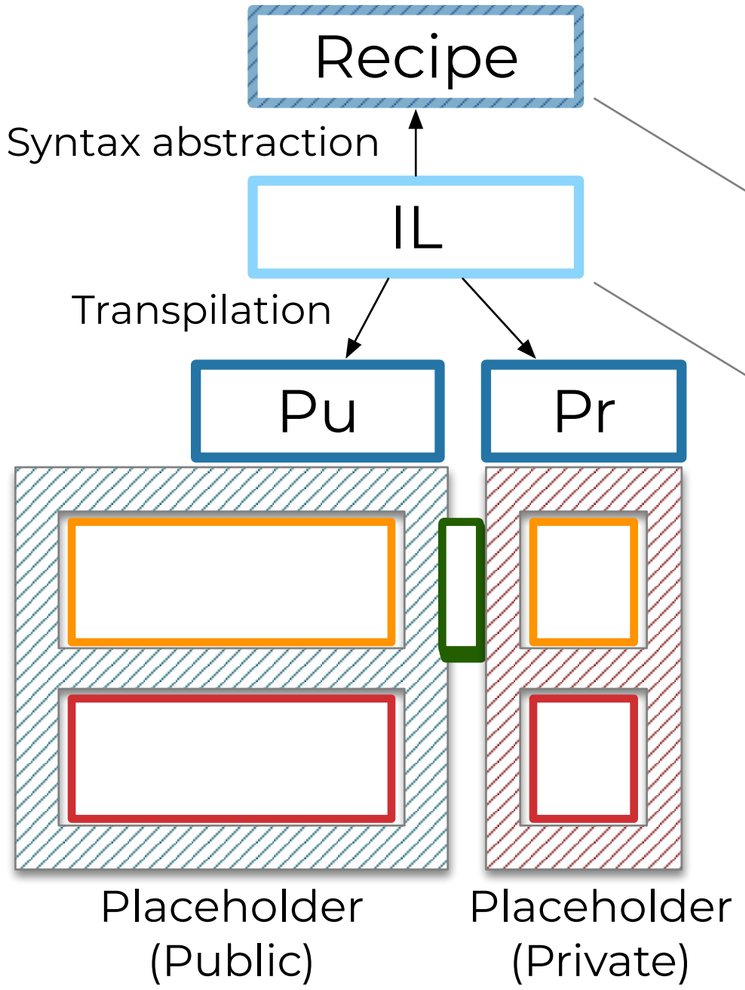
새로운 platform을 만드는 일의 즐거움과 고민



BIFROST

새로운 platform을 만드는 일의 즐거움과 고민





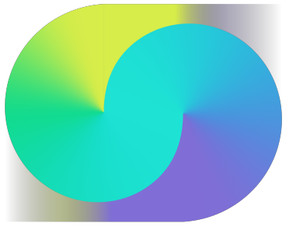
새로운 시도의 문턱을
낮추게 하는 방법은 무엇인가?

어떠한 Programming
model이 필요한가?

⋮

우리는 현재의 가정을
충분히 숙고하고 있는가?

우리가 다른 혁신을
외면하고 있지는 않은가?



BIFROST

감사합니다.

jonghyup@bifrost-platform.io

무엇인가
근본적으로
잘못되었다.