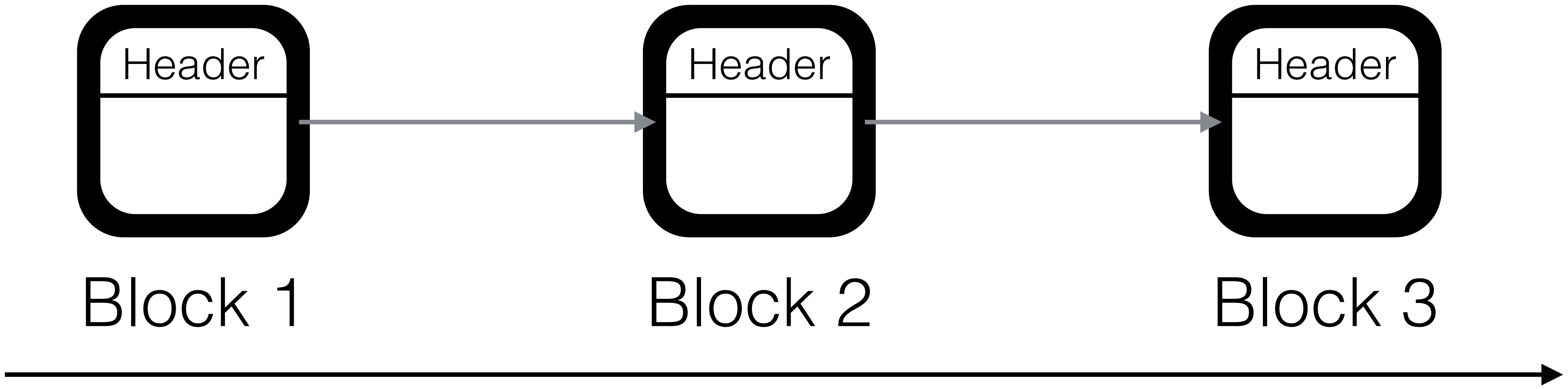
Introduction to Ethereum

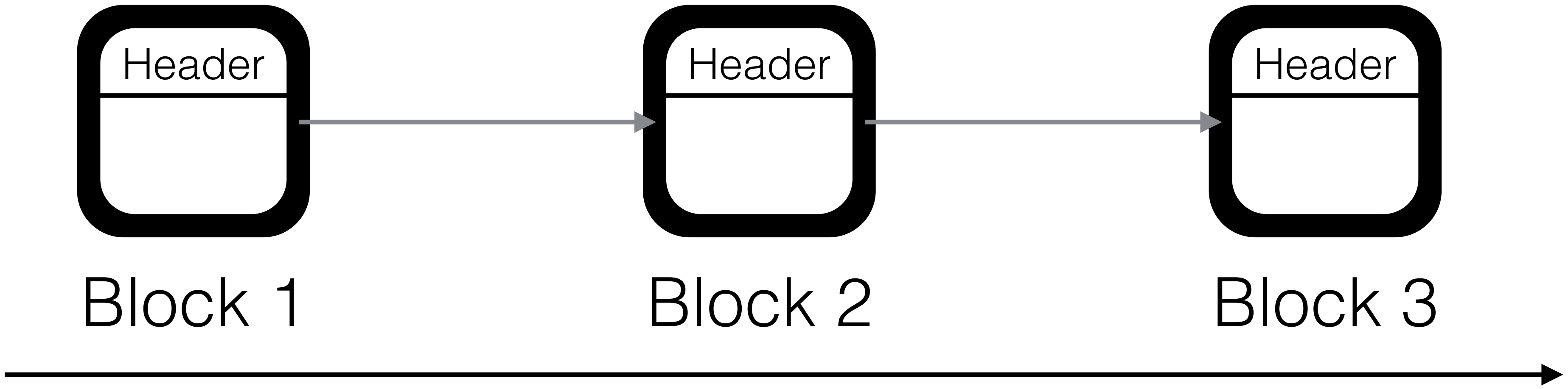
State changes in Blockchain



Consensus (nonce): ∅

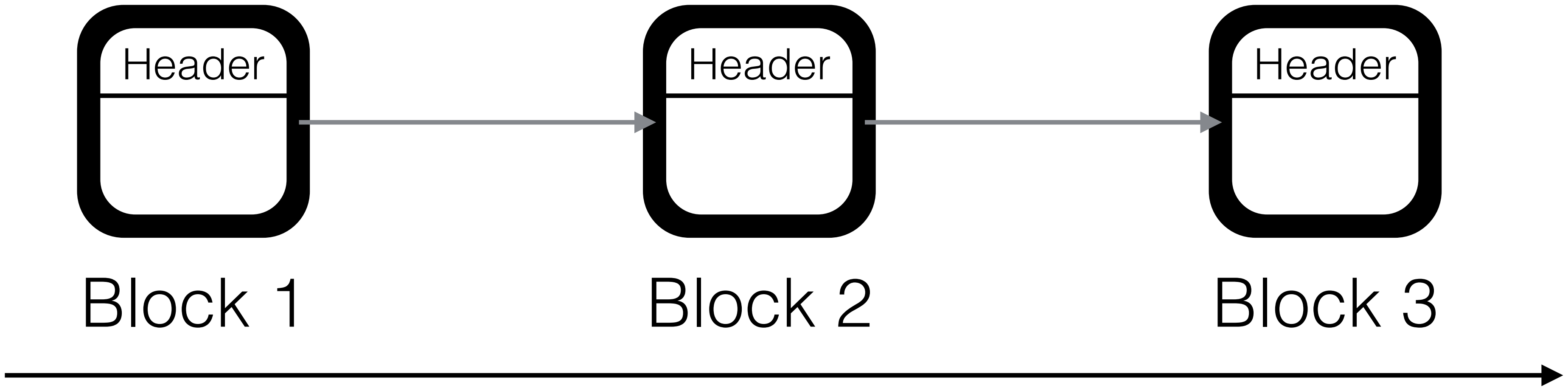
State change: ∅

State changes in Blockchain



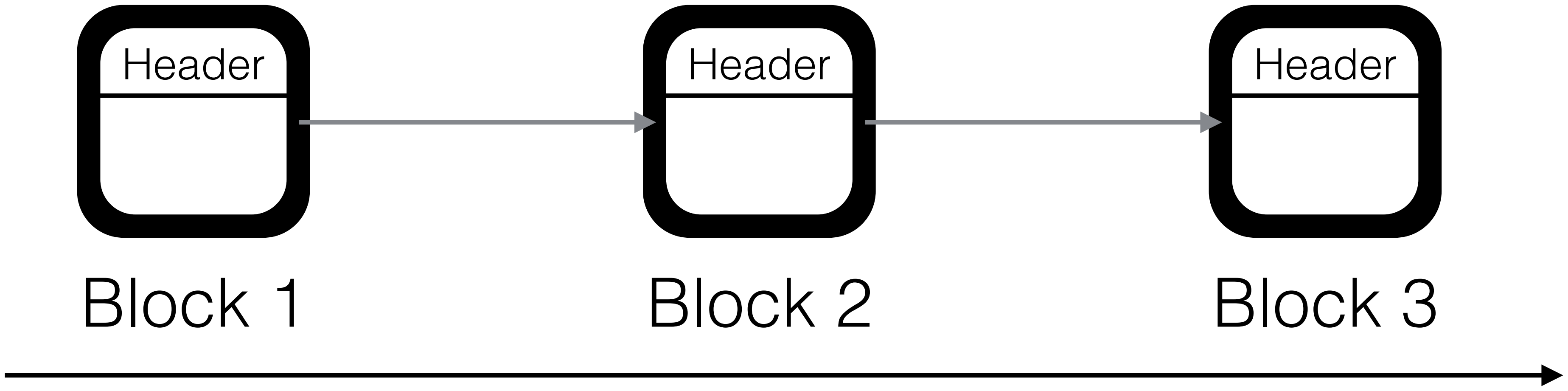
Consensus (nonce):	∅	0xab
State change:	∅	Transaction 1 A —> B, 3

State changes in Blockchain



Consensus (nonce):	∅	0xab	0xbv
State change:	∅	Transaction 1 A —> B, 3	Transaction 2 B —> C, 2

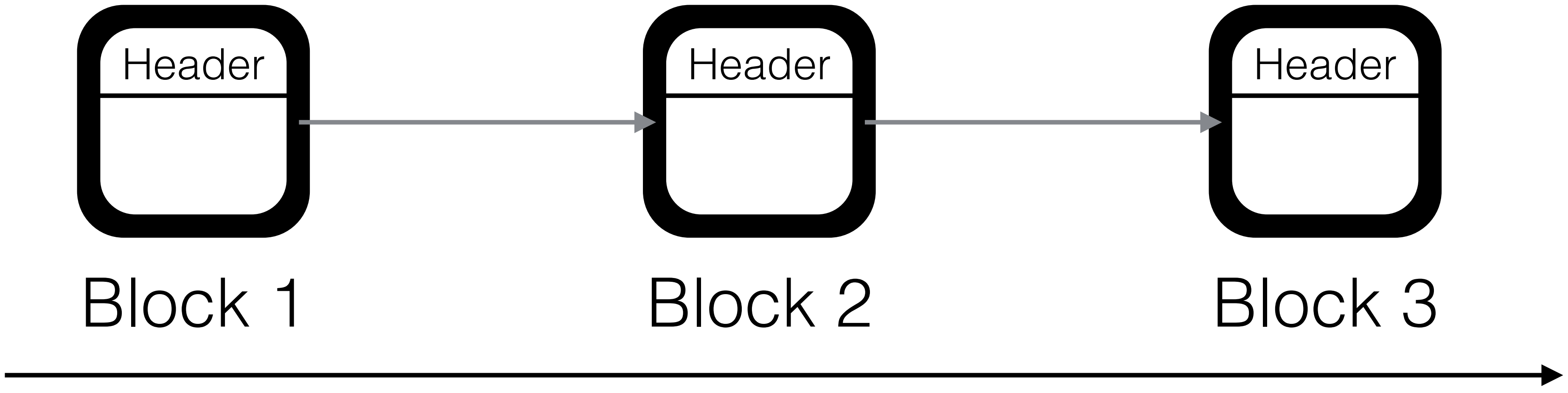
State changes in Blockchain



Consensus (nonce):	∅	0xab	0xbv
State change:	∅	Transaction 1 A —> B, 3	Transaction 2 B —> C, 2

Transaction 3
C —> D, 1

State changes in Blockchain

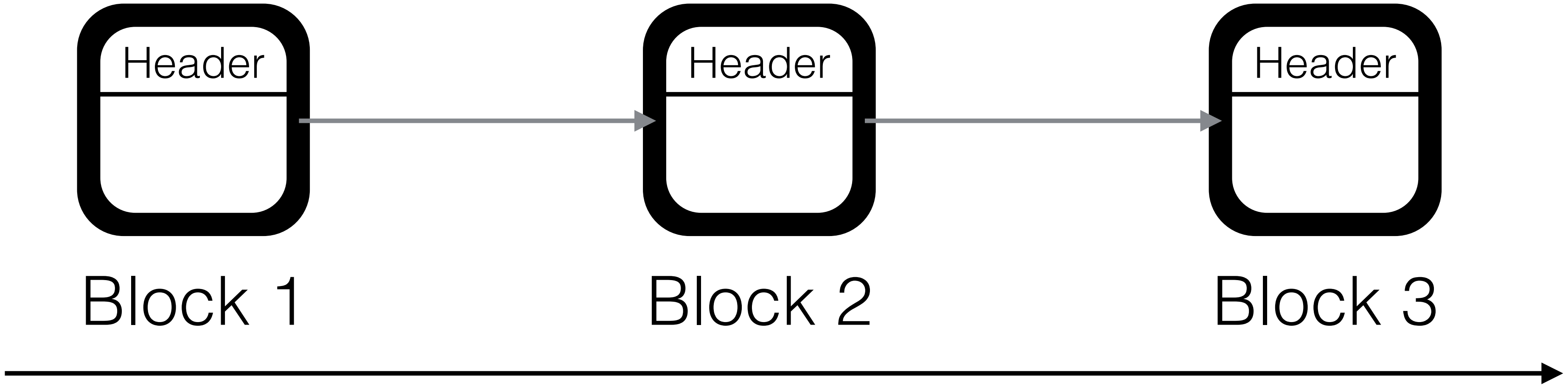


Consensus (nonce):	∅	0xab	0xbv
State change:	∅	Transaction 1 A —> B, 3	Transaction 2 B —> C, 2

Transaction 3
C —> D, 1

Is this transaction 3 valid?

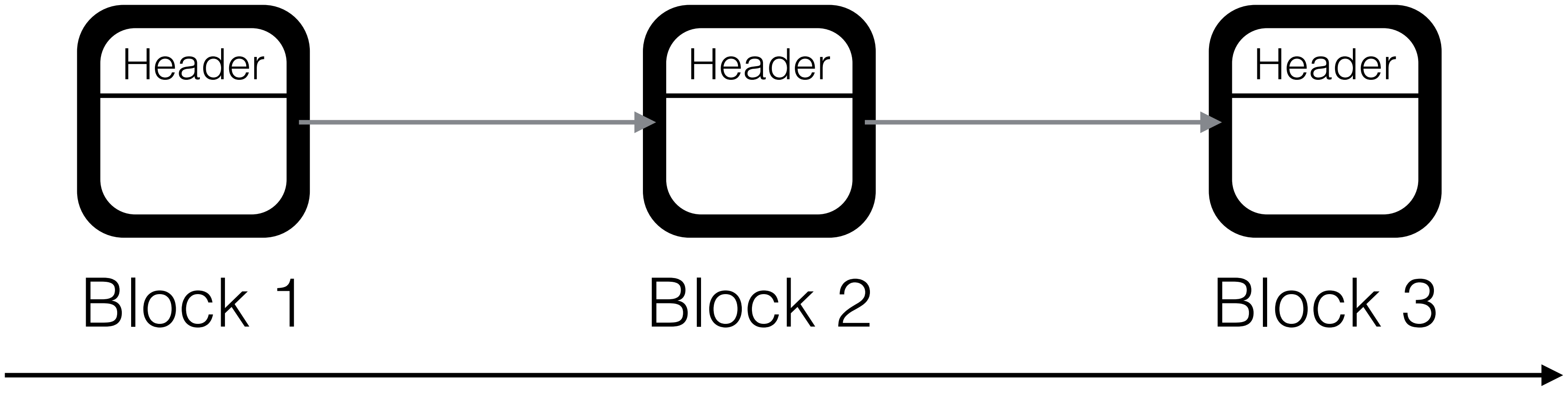
Store explicit state



Consensus (nonce):

State change:

Store explicit state



Header

Header

Header

Block 1

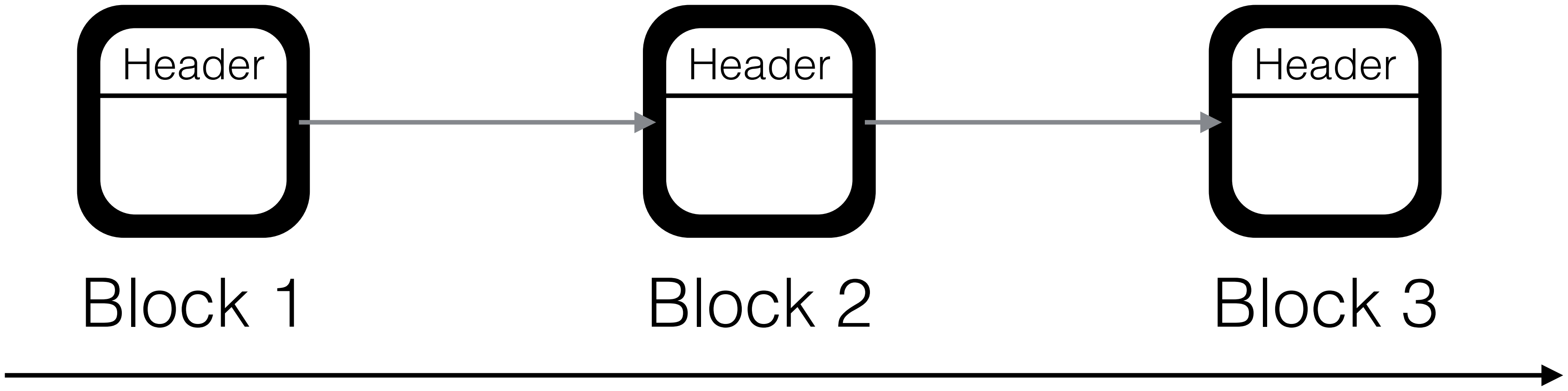
Block 2

Block 3

Consensus (nonce): 0xab

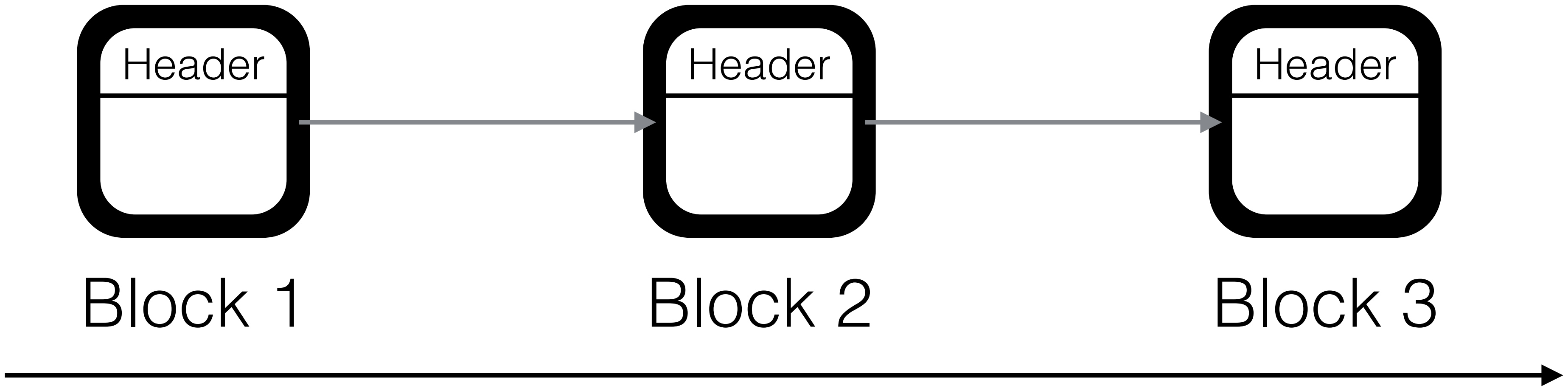
State change: Transaction 1
A —> B, 3

Store explicit state



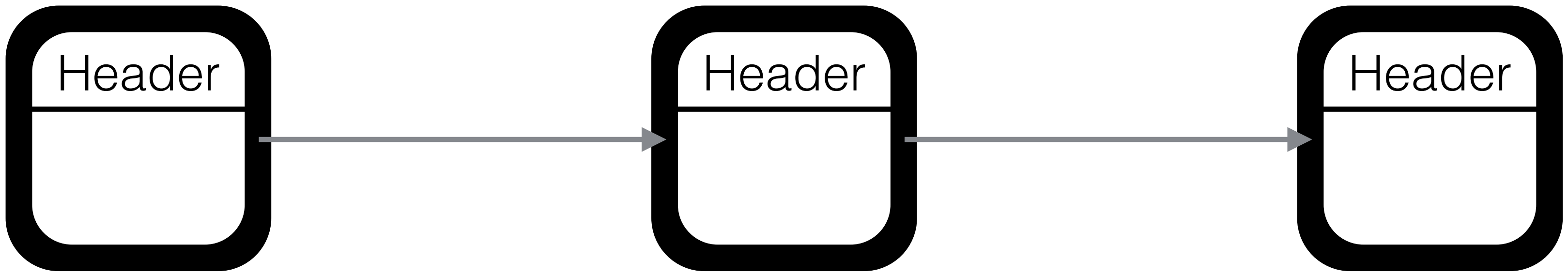
Consensus (nonce):		0xab	0xbv
State change:		Transaction 1 A —> B, 3	Transaction 2 B —> C, 2

Store explicit state



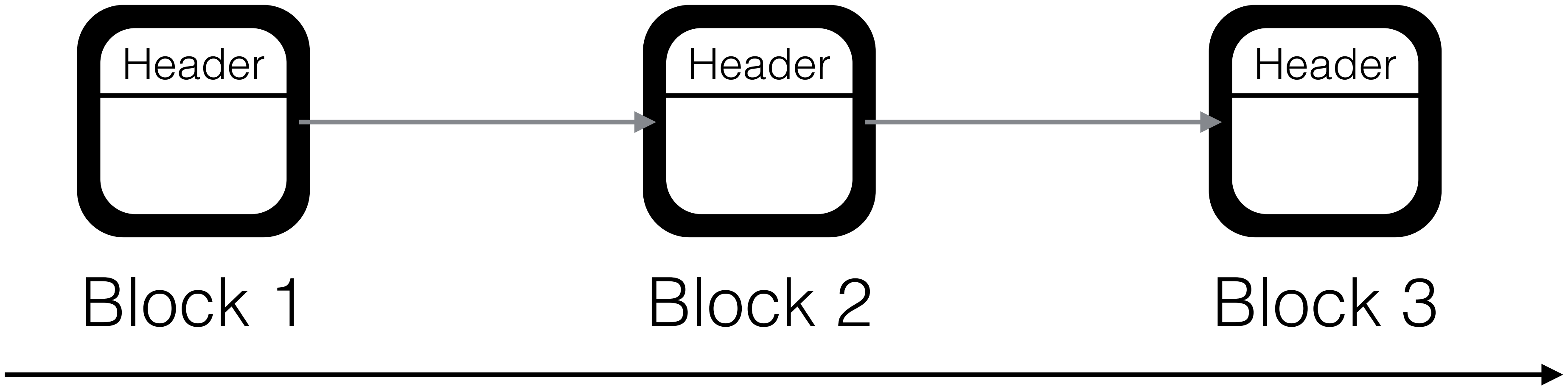
Consensus (nonce):		0xab	0xbv
State change:		Transaction 1 A —> B, 3	Transaction 2 B —> C, 2
State commitment:	{A:50}		

Store explicit state



	Block 1	Block 2	Block 3
Consensus (nonce):		0xab	0xbv
State change:		Transaction 1 A —> B, 3	Transaction 2 B —> C, 2
State commitment:	{A:50}	{A:47, B:3}	

Store explicit state



Consensus (nonce):		0xab	0xbv
State change:		Transaction 1 A —> B, 3	Transaction 2 B —> C, 2
State commitment:	{A:50}	{A:47, B:3}	{A:47, B:1, C:2}

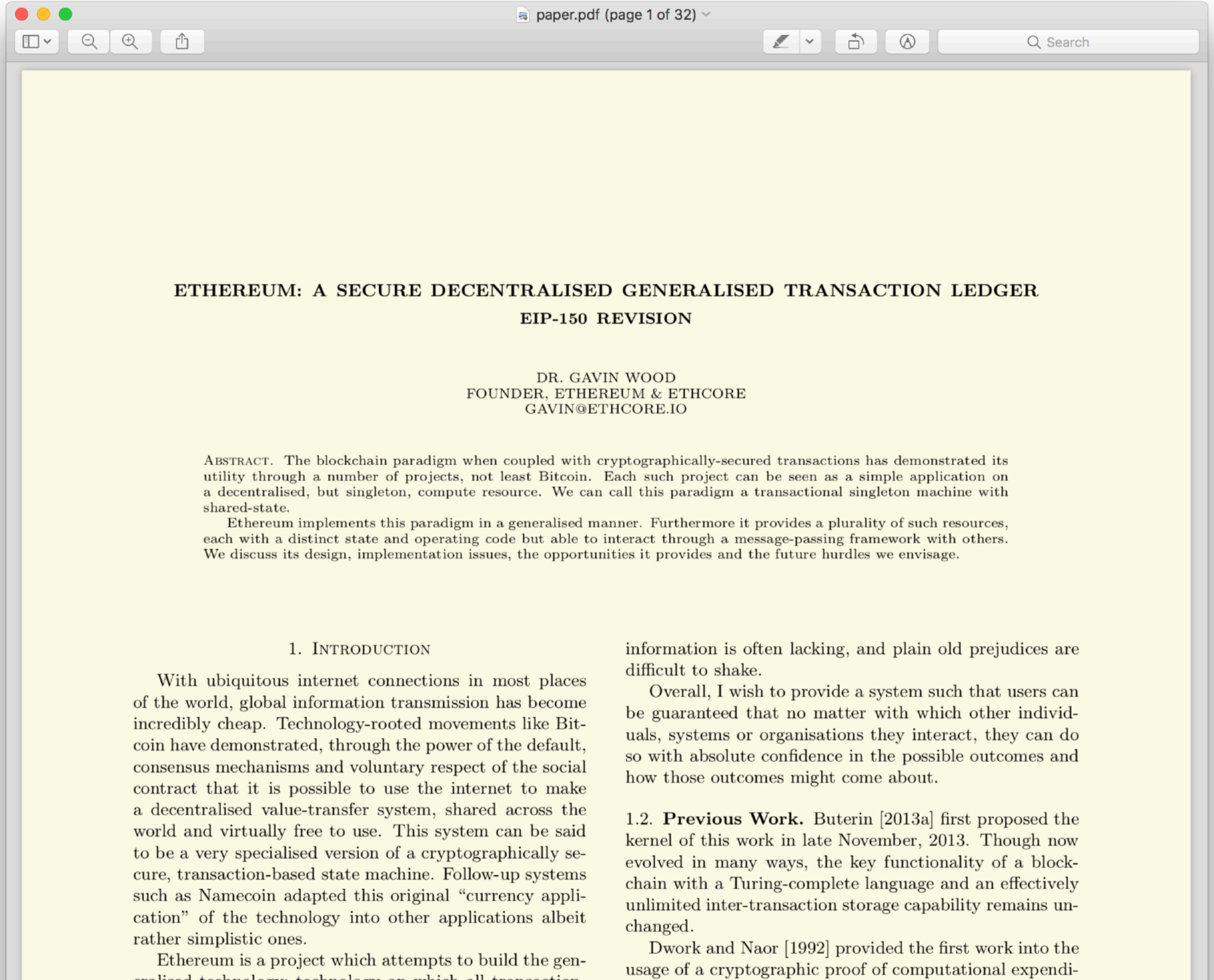
Advantages of explicit state storage

State commitment: {A:50} {A:47, B:3} {A:47, B:1, C:2}

- No need to go through whole history
- Sequence between any two blocks can be verified
- Light clients can sync up quickly

Ethereum - A universal Replicated State Machine

Ethereum - A universal Replicated State Machine



ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER
EIP-150 REVISION

DR. GAVIN WOOD
FOUNDER, ETHEREUM & ETHCORE
GAVIN@ETHCORE.IO

Abstract. The blockchain paradigm when coupled with cryptographically-secured transactions has demonstrated its utility through a number of projects, not least Bitcoin. Each such project can be seen as a simple application on a decentralised, but singleton, compute resource. We can call this paradigm a transactional singleton machine with shared-state.

Ethereum implements this paradigm in a generalised manner. Furthermore it provides a plurality of such resources, each with a distinct state and operating code but able to interact through a message-passing framework with others. We discuss its design, implementation issues, the opportunities it provides and the future hurdles we envisage.

1. Introduction

With ubiquitous internet connections in most places of the world, global information transmission has become incredibly cheap. Technology-rooted movements like Bitcoin have demonstrated, through the power of the default, consensus mechanisms and voluntary respect of the social contract that it is possible to use the internet to make a decentralised value-transfer system, shared across the world and virtually free to use. This system can be said to be a very specialised version of a cryptographically secure, transaction-based state machine. Follow-up systems such as Namecoin adapted this original "currency application" of the technology into other applications albeit rather simplistic ones.

Ethereum is a project which attempts to build the gen-

information is often lacking, and plain old prejudices are difficult to shake.

Overall, I wish to provide a system such that users can be guaranteed that no matter with which other individuals, systems or organisations they interact, they can do so with absolute confidence in the possible outcomes and how those outcomes might come about.

1.2. **Previous Work.** Buterin [2013a] first proposed the kernel of this work in late November, 2013. Though now evolved in many ways, the key functionality of a blockchain with a Turing-complete language and an effectively unlimited inter-transaction storage capability remains unchanged.

Dwork and Naor [1992] provided the first work into the usage of a cryptographic proof of computational expendi-

Ethereum

- A (slow and expensive) world computer
- Consensus among all nodes about the execution
- More transaction type flexibility than Bitcoin
- Quasi-Turing complete language

Replicated State Machine

- Set of possible states: S
- Set of possible inputs: I
- Set of possible outputs: O

- Transition function $f: S \times I \rightarrow S \times O$

- Start state $s \in S$ (genesis block)

Replicated State Machine

Arbitrary programs

- Set of possible states: S
- Set of possible inputs: I
- Set of possible outputs: O

- Transition function $f: S \times I \to S \times O$

- Start state $s \in S$ (genesis block)

Replicated State Machine

- Set of possible states: S
- Set of possible inputs: I
- Set of possible outputs: O

Arbitrary programs

- Transition function f: S × I → S × O

Execute programs

- Start state s ∈ S (genesis block)

Ethereum

- **States S** = a map from address to state

address	code	storage	balance	nonce

- **Inputs I** (transactions)

from	sig	nonce	to	data	value	gaslimit	gasprice

- **Transition f**:
    - Validate signature, nonce
    - Execute code (from, data, value, gaslimit, gasprice)

- Start state: ∅