Praktikum Innovative Mobile Applications:
Blockchain Applications

Blockchain Introduction

Andreas Sedlmeier | LMU Munich | Mobile and distributed systems group

Blockchain Building Blocks

- Hash
- Block
- Blockchain
- Distributed Blockchain
- Tokens
- Coinbase Transactions

Basierend auf:
http://blockchain.mit.edu/blockchain/
NISTIR 8202: Blockchain Technology Overview

1: Hash

- Ein Hash ist ein „Fingerprint“ von Eingabedaten
- Er ist die Ausgabe einer Hash-Funktion
- Technisch: Eine Hash-Funktion bildet Eingabedaten beliebiger Länge auf Ausgabedaten fester Länge ab
- Für Blockchains von Interesse: Cryptographic Hash Functions
  Spezielle Eigenschaft: Einwegfunktion
  D.h. es ist „intractable“ vom Output auf den Input zu schließen

1: Hash

Länge des SHA256 Hash ist immer 64 (Hex-codiert),
unabhängig von der Lände des Input
→ Outputlänge: 64*16 = 1024 Bit

z.B. Hashfunktion SHA256

Data	hello world
Hash	b94d27b9934d3e08a52e52d7da7dabfac484efe37a5380ee9088f7ace2efcde9

Data	Beispieltext mit Zeilenumbrüchen etc.
Hash	36258485c58ee45af253da45adaeb313b94b16f987c8d84f5bc4a6ec844215f8

2: Block

Erweitere den Input um 2 weitere Felder. D.h. der Hash wird jetzt gebildet über:

- Block-Id
- Nonce
- Data

Block-Id	1
Nonce	72608
Data	hello world
Hash	dbc1d5036d48dff0b133f2199f25b2fbeb7a34ba47872953d5103ef19fc0a698

2: Gültiger (signed) Block

Definiere eine (willkürliche) Restriktion auf dem Hash:
z.B.: Nur Hashes die mit 0000 beginnen sind gültig (signed)

Block-Id	1
Nonce	72608
Data	hello world
Hash	dbc1d5036d48dff0b133f2199f25b2fbeb7a34ba47872953d5103ef19fc0a698

2: Gültiger (signed) Block

Durch Ändern des Nonce Wertes verändert sich der Hash.
→ Ändere den Nonce Wert so lange, bis ein Hash mit 0000 am Anfang entsteht

Block-Id	1
Nonce	72608
Data	hello world
Hash	dbc1d5036d48dff0b133f2199f25b2fbeb7a34ba47872953d5103ef19fc0a698

2: Gültiger (signed) Block

Durch Ändern des Nonce Wertes verändert sich der Hash.
→ Ändere den Nonce Wert so lange, bis ein Hash mit 0000 am Anfang entsteht

Block-Id	1
Nonce	726099
Data	hello world
Hash	95673d3c7d434bfc159a239ee579fa58bcebdcbec090fabc5b546858a22434ee

2: Gültiger (signed) Block

Durch Ändern des Nonce Wertes verändert sich der Hash.
→ Ändere den Nonce Wert so lange, bis ein Hash mit 0000 am Anfang entsteht

Block-Id	1
Nonce	18014
Data	hello world
Hash	0000d3d3702b7dce97cfd9f35705650ff9cfbc8102594352568024167859403 3

Dieser Prozess des „Findens eines gültigen Hash-Wertes durch variieren der Nonce“ nennt sich „**mining**“.

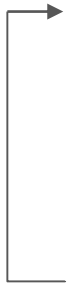
3: Blockchain

- Verknüpfen von Blöcken durch Erweiterung um ein Feld „Previous“
- Dieses „Previous“ Feld enthält den Hash des vorhergehendes Blocks
- Aus Sicht der Datenstruktur erhalten wir damit eine **linked list**
  die als Pointer Hashes verwendet

3: Blockchain

Verknüpfen von Blöcken durch Erweiterung um ein Feld „Previous“ (Prev)

Block-Id	1
Nonce	85460
Data	hello world
Prev	0000000000000000000000000000000000000000000000000000000000000000
Hash	0000a51677b3cc64056c498c1d6c88130528ca64b9c4de33ef694b5bf0007028



Block-Id	2
Nonce	127567
Data	ich bin block zwei
Prev	0000a51677b3cc64056c498c1d6c88130528ca64b9c4de33ef694b5bf0007028
Hash	0000dd3e6b03bdeef93e8e4d60b9baae96065d7355314567a4bd873cae2336da

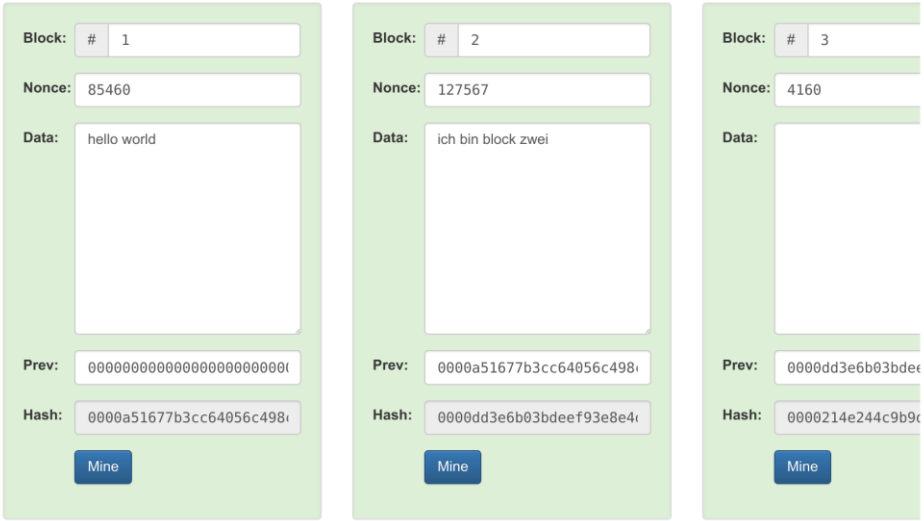
3: Blockchain

- Änderungen an den Daten eines Blocks
  (ändern den Hash dieses Blocks)

- führen nun zu Änderungen an allen Hashes der folgenden Blöcke!
  (da jeweils das „previous“ Feld des nächsten Blocks geändert werden müsste, was wiederum den Hash dieses Blocks ändert usw.)

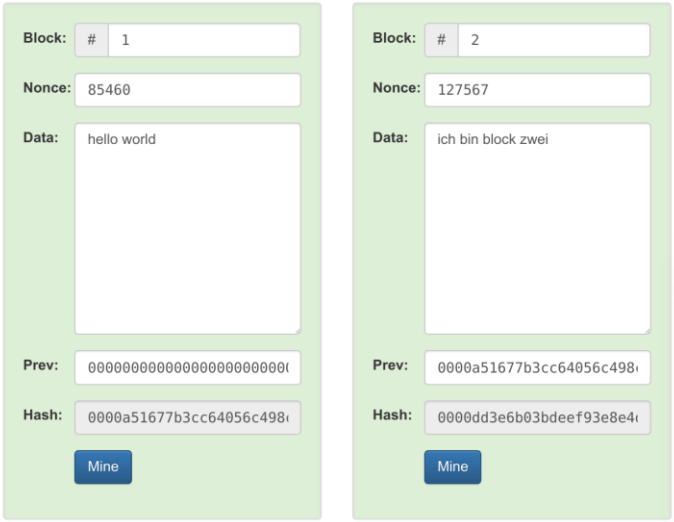
3: Blockchain

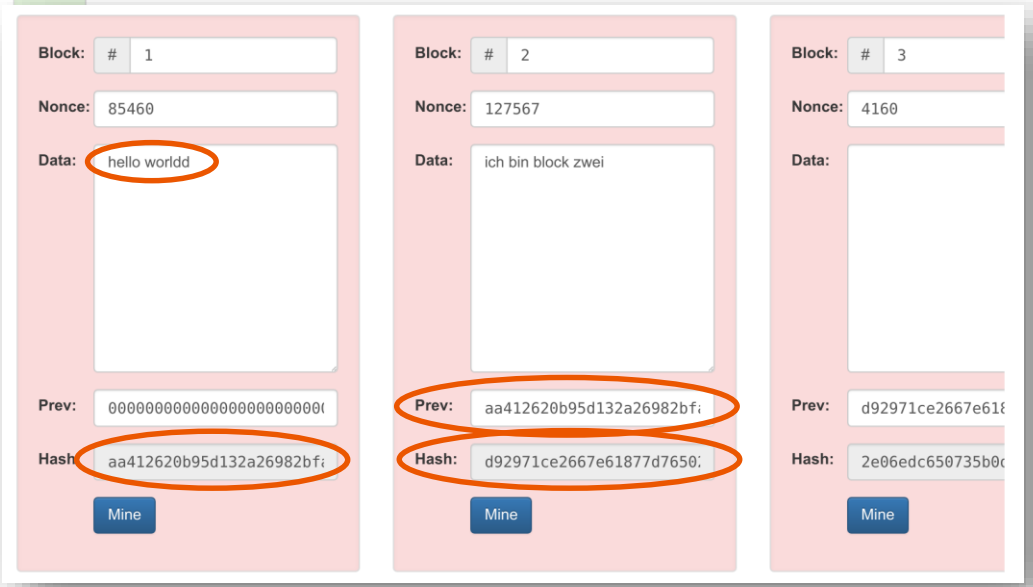
http://blockchain.mit.edu/blockchain/



3: Blockchain

http://blockchain.mit.edu/blockchain/





Um die Kette wieder gültig zu machen müssten alle Blöcke, beginnend mit dem veränderten Block neu ge-mined werden

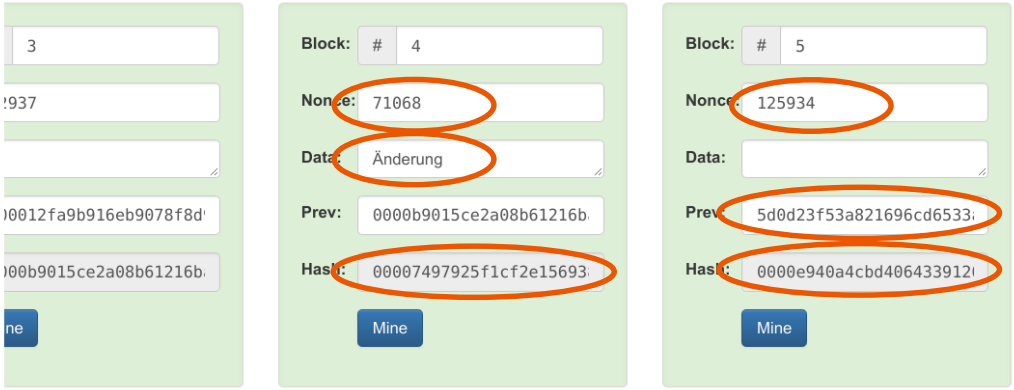
3: Distributed Blockchain

- Wie können wir feststellen, dass die Chain „re-mined“ wurde?
- Durch Replizieren der kompletten Blockchain auf allen Teilnehmern (Nodes) des Netzwerkes
- Um die Übereinstimmung der (kompletten) Chain zu überprüfen reicht es aus die Hashes des letzten Blocks zu vergleichen

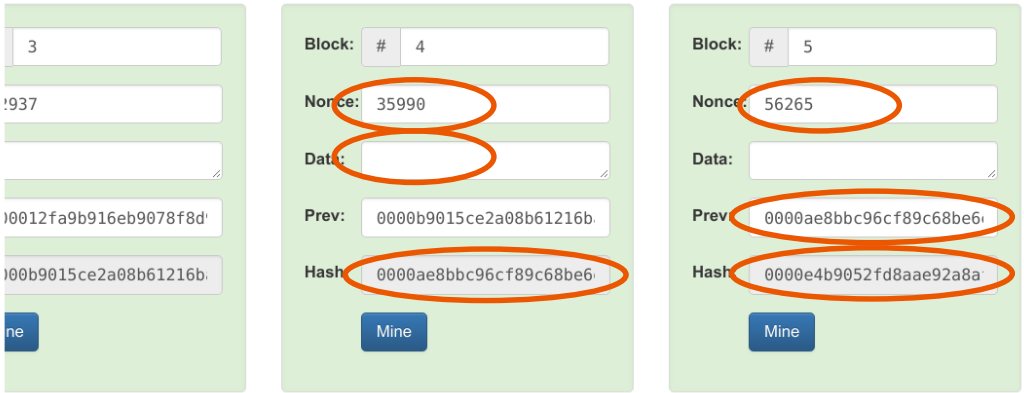
3: Distributed Blockchain

http://blockchain.mit.edu/blockchain/

Peer A



Peer B



3: Blockchain

- Änderungen an den Daten eines Blocks (ändern den Hash dieses Blocks)
- führen nun zu Änderungen an allen Hashes der folgenden Blöcke.
- Durch verteiltes Replizieren und Vergleich der Hashes können wir Abweichungen feststellen

- Dies wird oft als **immutability** („Un-veränderbarkeit“) bezeichnet
  Das würde bedeuten Änderungen sind unmöglich (**Tamper-proof**)

- Aber: Das System ist definiert durch seine gewählten Regeln!
  (z.B: Jeder Hash muss mit 0000 beginnen, ein gültiger Block muss ein Feld data enthalten, …)
- Damit stellt sich die Frage: Wer legt die Regeln fest und wer kann sie ändern?
  (siehe Hard-Fork der Ethereum Blockchain nach The DAO incident)

- Korrekt wäre somit: **Tamper-evident**, d.h. Änderungen können erkannt werden

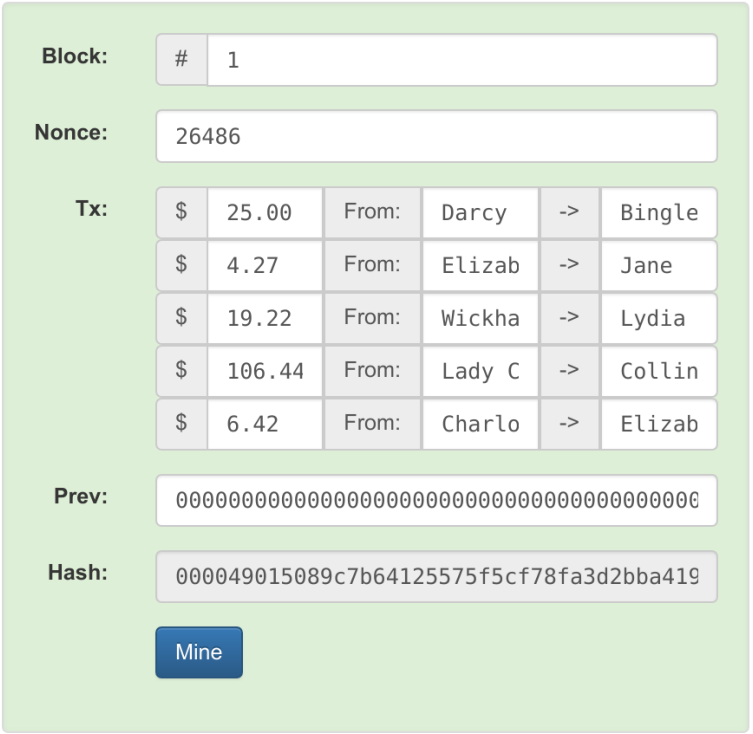
4: Tokens

- Wofür kann das Ganze nun genutzt werden?
  D.h. was schreiben wir wirklich als **data**?

- Einführen sogenannter **Tokens** (z.B. mit dem Namen $ oder BTC)
- Diese existieren als Teil von **Transactions** (als Token, Sender, Empfänger)
- Von diesen Transactions können mehrere in einem Block enthalten sein

- Note: Struktur dieser Token und Transactions ist frei gewählt,
  genauso könnte etwas Anderes definiert werden

4: Tokens



http://blockchain.mit.edu/blockchain/

- **Data** Feld wurde ersetzt durch **Tx**
- Dieses enthält mehrere sogenannte Transactions
- Jede Transaction enthält einen Anzahl an **Token** einen **Sender** und einen **Empfänger**

- **Hinweis:**
  Hier wird nichts „versendet“ im Sinne von „es werden Daten oder Netzwerkpakete vom Sender zum Empfänger übertragen“ oder Ähnlichem.
  Es ist zu verstehen als Kassenbuch (**Ledger**).
  D.h. als Dokumentation von Transaktionen.

- **Twist:** Das reicht (fast) aus um ein Geldsystem zu etablieren. Es ist kein „physischer“ Tausch von Geldscheinen etc. notwendig.

5: Coinbase

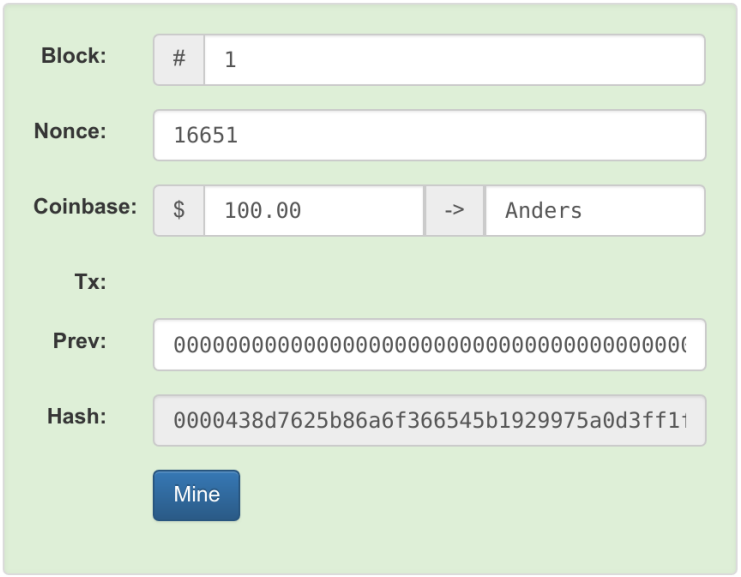
- Wo kommen die Tokens (bzw. das Geld) her?
- Bisher nur Transaktionen, keine Information darüber ob/wieviele Token ein Sender besitzt

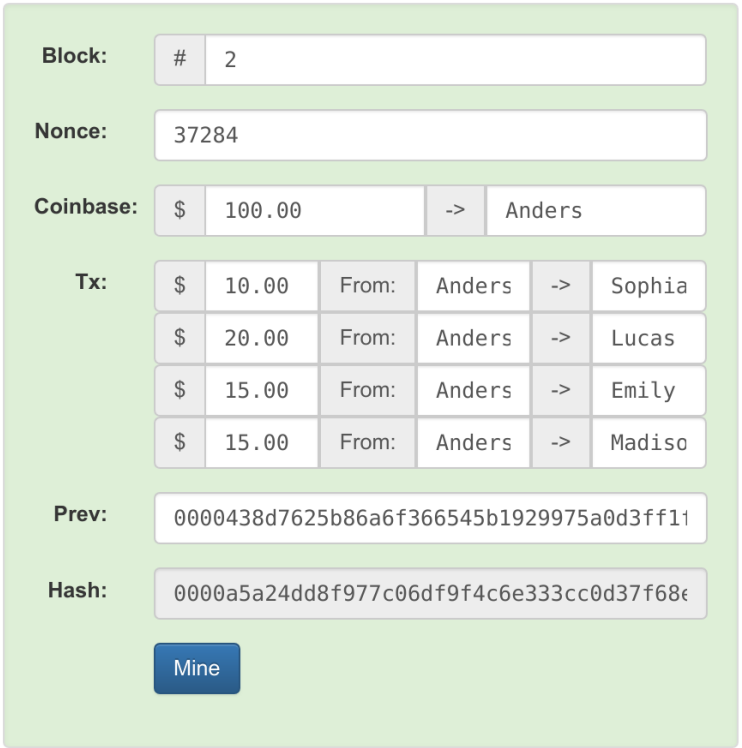
Lösung:

- Führe ein Feld **Coinbase Transaction** in jedem Block ein
- Diese bestehen aus Token Anzahl und einem Empfänger (D.h. sie erzeugen Tokens „aus dem Nichts“)

5: Coinbase

http://blockchain.mit.edu/blockchain/





5: Coinbase

- Wieviele Tokens wann an wen über die Coinbase Transactions „ausgeschüttet“ werden ist wieder beliebig definiert (als weitere Regel des z.B. Bitcoin Systems)
- In der Regel trägt sich jeder Miner selbst als Empfänger der Coinbase Transaction ein
- Derjenige, der als Erster einen gültigen Hash für den nächsten Block gefunden hat erhält so die Belohnung für seinen Rechenaufwand

Beispiel Bitcoin:
Satoshi Nakamoto, Bitcoin’s creator, set the block reward schedule when he created Bitcoin: The block reward started at 50 BTC in block #1 and halves every 210,000 blocks.
Since blocks are mined on average every 10 minutes, 144 blocks are mined per day on average. At 144 blocks per day, 210,000 blocks take on average four years to mine. *

> **Total circulation will be 21,000,000 coins. It’ll be distributed to network nodes when they make blocks, with the amount cut in half every 4 years. first 4 years: 10,500,000 coins next 4 years: 5,250,000 coins next 4 years: 2,625,000 coins next 4 years: 1,312,500 coins etc**
>
> - Satoshi Nakamoto

* https://www.bitcoinmining.com/what-is-the-bitcoin-block-reward/

That‘s it.

That‘s it.

… well basically ;)

- Wie werden Teilnehmer adressiert? (Public-key Cryptography + Key-Hashes als IDs)
- Wie beweisen Teilnehmer ihre Identität? (Cryptographische Signaturen)
- Wie wird Konsens erzeugt, darüber welche Chain die „korrekte“ ist?
  (Forking, Consensus making → Longest chain rule, etc.)
- Welches P2P-System wird verwendet? (Robustheit gegen Angriffe, DDOS, etc.)
- Welche Hashfunktion wird verwendet?
- Können auch komplexere „Logiken“ als nur simple Transaktionen eingeführt werden?
  (siehe Smart Contracts, etc.)
- Wer darf am System teilnehmen, d.h. Lesen? Schreiben? (Permissioned VS Permissionless)
- Wie wird in einem dezentralen System Ordnung, d.h. Reihenfolge hergestellt?
  (Proof-of-work as a decentralized clock: https://grisha.org/blog/2018/01/23/explaining-proof-of-work/)

Geht das nicht auch einfacher?

Geht das nicht auch einfacher?

Antwort: Absolut, kommt ganz auf die Anforderungen an!

→ Was sind die Anforderungen, die die Grundlage für die Entwicklung der (bisherigen) Blockchain Systeme waren?

Key characteristics of blockchain technology:
NISTIR 8202: Blockchain Technology Overview (NIST: National Institute of Standards and Technology)

- Append only Ledger
- Cryptographically secure (Ledger is attestable / tamper-evident)
- Shared (the ledger is shared amongst multiple participants → Transparency)
- Distributed (scaling the number of nodes makes it more resilient to attacks by bad actors)

Sind das auch die Anforderungen, die ich an mein System stelle?

Blockchain Flowchart

United States Department of Homeland Security (DHS) Science & Technology Directorate

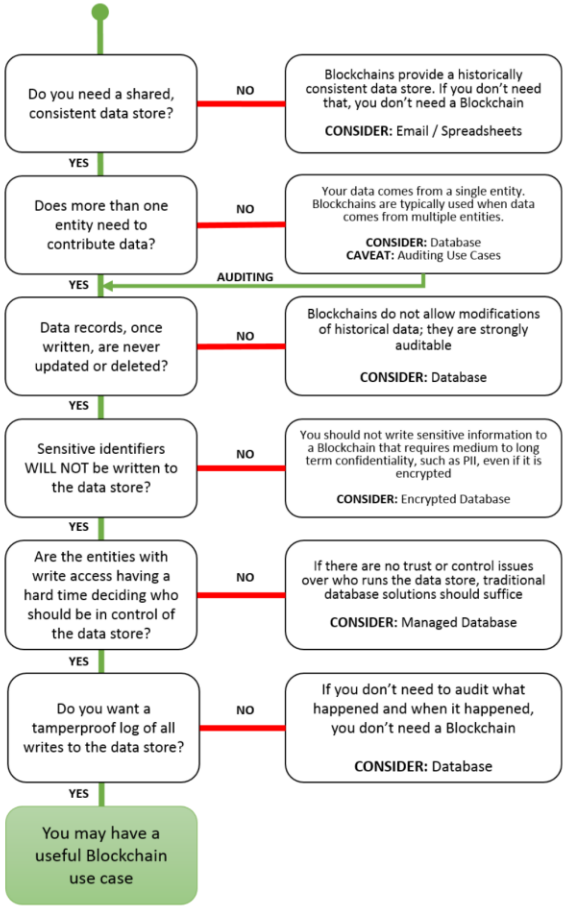
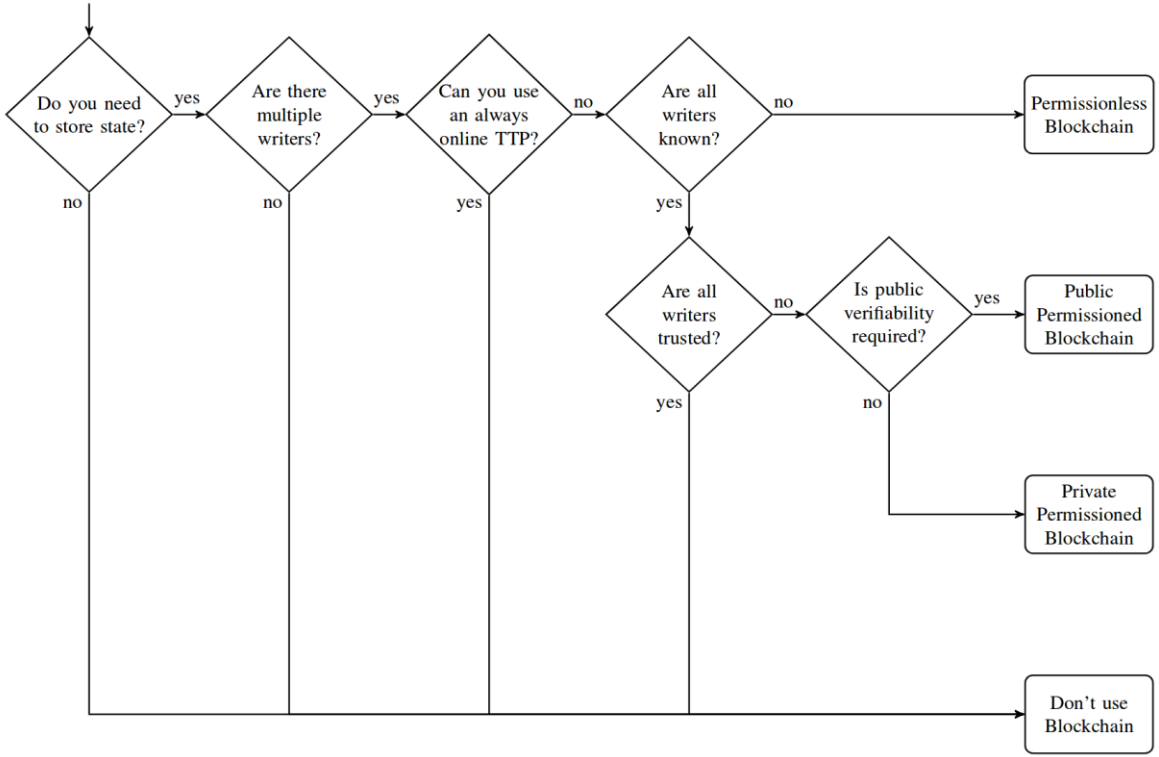


Figure 6 - DHS Science & Technology Directorate Flowchart *

*Yaga, Dylan, et al. "Blockchain technology overview." Draft NISTIR 8202 (2018). https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

Blockchain Flowchart

ETH Zürich: Do you need a Blockchain?



Wüst, Karl, and Arthur Gervais. "Do you need a Blockchain?." 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). IEEE, 2018.
https://eprint.iacr.org/2017/375.pdf

*

Where to continue?

- Interactive Blockchain: http://blockchain.mit.edu/blockchain/
- Yaga, Dylan, et al. "Blockchain technology overview." Draft NISTIR 8202 (2018). https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf
- Wüst, Karl, and Arthur Gervais. "Do you need a Blockchain?." 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). IEEE, 2018. https://eprint.iacr.org/2017/375.pdf