Analyse verschiedener Blockchain-Frameworks und Implementierungen

Daniel Kucko, Johannes Schwickerath, Sebastian Sidortschuck

-

Im Rahmen des Projektstudiums bezüglich der aktuellen Blockchain-Technologie unter der Betreuung von Prof. Skornia wurden neben dem Bau einer eigenen Blockchain auch verschiedene Frameworks und Implementierungen von Blockchains auf diverse Faktoren untersucht.

Die untersuchten Frameworks und Implementierungen sind:

* BigchainDB
* BitCoin
* Ethereum
* Hyperledger Sawtooth
* Hyperledger Fabric
* NameCoin
* ScoreX

Im Folgenden werden die Ergebnisse der jeweiligen Untersuchungen präsentiert.

[BigchainDB](https://www.bigchaindb.com/)

Bei BigchainDB handelt es sich um eine sogenannte Blockchain-Datenbank. Das bedeutet es ist eine „Big Data“ Datenbank, mit typischen Blockchain-Features, wie Dezentralisierung und Unveränderbarkeit der Daten kombiniert. Während Blockchains üblicherweise eher wie Logfiles funktionieren und „nur“ Transaktionen zwischen Adressen speichern, bietet BigchainDB die Möglichkeiten einer voll funktionalen Datenbank durch die Integration von NoSQL Datenbanken wie MongoDB. Durch die Dezentralisierung entsteht eine sehr große Daten Redundanz, die verhindert, dass Daten verloren gehen. Zudem wird es dadurch eventuellen Angreifern erschwert, die Datenbank zu verändern. Daten können durch verschiedene Transaktionen in die Datenbank gespeichert und an Nutzer übertragen werden.

Sprache: Python

Vorteile:

* Flexible Definitionen von Übertragungsbedingungen, sprich Signaturen der Vertragspartner

Nachteile:

* Bietet keine Unterstützung für Smart Contracts.
* Da die Blockchain öffentlich einsehbar ist, eignet sich die Datenbank nicht zum Speichern sensibler Daten

Anmerkungen:

* BigchainDB befindet sich derzeit noch in Entwicklung und wird nicht als „production ready“ empfohlen.
* BigchainDB bietet derzeit keine Unterstützung für Windows oder MacOS Systeme

BitCoin

BitCoin ist die erste und bekannteste Implementierung einer Blockchain. Durch den extremen Erfolg ihrer Kryptowährung, löste BitCoin eine ganze Welle von neuen Blockchains und Kryptowährungen aus. Durch verschiedene Updates, die zu sogenannten Hard Forks in der Chain führten entstanden auch mehrere „Altcoins“, also alternative Währungen, die auf der gleichen Technologie beruhen, die bekanntesten darunter Litecoin und NameCoin. BitCoin setzt bekanntermaßen auf einen Proof-of-Work Algorithmus, um neue Blöcke zu validieren und zu minen. Die Blockchain verfügt über eine eigene stackbasierte Skriptsprache, um Transaktionen zu validieren und durchzuführen, diese ist jedoch sehr stark beschränkt in ihrer Funktionalität und bietet daher keine Unterstützung für Smart Contracts.

Sprache: C++

Vorteile:

* Gut getestet und sehr stabil und sicher

Nachteile:

* Proof-of-Work ist sehr Ressourcen (Strom + Hardware) intensiv

Anmerkungen:

Ethereum

Ethereum ist eine dezentralisierte Plattform basierend auf der Blockchain Technologie.

Ethereum kann als öffentliche oder private Blockchain verwendet werden.

Durch die Ethereum-Virtual-machine (EVM) ist es möglich dezentralisierte Programme (dApps), auch Smart Contracts genannt, auszuführen. Zum programmieren dieser dApps mehrere Programmiersprachen exisitieren, die populärsten sind Solidity (Syntax ähnlich zu Javascript/C) und Vyper (Syntax ähnlich zu Python). Um die Nodes welche Smart Contracts ausführen (aktuell alle Miner) für den Rechenaufwand zu entlohnen (besonders auf der öffentlichen Blockchain) existiert die Kryptowährung Ether mit der dies bezahlt wird.

Sprache: Go

Vorteile:

* Smart Contract Unterstützung
* Einige dApps sind bereits online

Nachteile:

* Aktuell noch Proof-of-Work (siehe Bitcoin Nachteile)

Anmerkungen:

* Ethereum nutzt aktuell Proof-of-Work, wird aber in Zukunft auf Proof-of-Stake umsteigen

[Hyperledger Sawtooth](https://www.hyperledger.org/projects/sawtooth)

Bei Hyperledger Sawtooth handelt es sich um ein Framework zur Entwicklung eigner Blockchain-Anwendungen. In dem Framework herrscht eine strenge Teilung zwischen Kernfunktionalität (Netzwerk) und Anwendungslogik. Dies ermöglicht eine angenehme Entwicklung von eigenen Anwendungen auf Blockchainbasis. Das Framework unterstützt sowohl einfache Transaktionen als auch Smart Contracts, letztere können sogar auf der Ethereum Blockchain ausgeführt werden.

Sprache: Go

Vorteile:

* Modular
* Teil des Hyperledger Projekts der Linux Foundation (Mit Unterstützung von Firmen wie Intel, SAP, Bosch)

Nachteile:

Anmerkungen:

* Die Kern Implementierung des Frameworks beruht auf der Sprache Go. Treiber für die Entwicklung eigener Anwendungen sind in verschiedenen Sprachen, unter Anderem Python, Java, C++, verfügbar.

Hyperledger Fabric

Fabric ist ein weiteres Project von Hyperledger welches die Entwicklung von Blockchain-Anwendungen erleichtern soll. Die Hauptaufgabe von Fabric ist es Smart Contracts, hier auch „chaincode“ genannt auszuführen. Wie auch bei Sawtooth handelt es sich hier um ein Framework, welches sehr viel Wert auf Modularität setzt. Dies erlaubt es eine „permissioned“ Blockchains zu erstellen welche je nach Use-Case Privatheit/Vertraulichkeit und andere wichtige Attribute unterstützt. Chaincode kann aktuell in Go entwickelt werden, aber Unterstützung für Sprachen wie Java ist bereits geplant.

Sprache: Go

Vorteile:

* Modular
* Teil des Hyperledger Projekts der Linux Foundation (Mit Unterstützung von Firmen wie Intel, SAP, Bosch)

Nachteile:

* Komplexerer Aufbau aufgrund der Modularität (im Vergleich zu beispielsweise Ethereum)

Anmerkungen:

NameCoin

Namecoin ist eines der ersten DNS auf der Blockchain. Bitcoin fork.

Sprache: C++ (Bitcoin Fork)

Vorteile:

* Gute Sicherheit durch kombiniertes Mining mit Bitcoin

Nachteile:

* Komplett öffentlich
* Sicherheit abhängig von Bitcoin

Anmerkungen:

* Eine mögliche Alternative in der Zukunft könnte MoneroDNS sein, da dieses Privatheit unterstützt im Gegensatz zu NameCoin. Jedoch existiert noch keine fertige Implementierung, was bei NameCoin der Fall ist.

[ScoreX](https://github.com/ScorexFoundation/Scorex)

Bei ScoreX handelt es sich um eine leichtgewichtige Plattform zum Entwickeln und Testen verschiedener Blockchain-Technologien. Das Framework ist sehr allgemein gehalten und erlaubt dadurch die Entwicklung verschiedenster Algorithmen und Datenstrukturen. Durch eine extreme Verallgemeinerung der Blockchain und Kryptowährung Technologien bietet ScoreX Forschern und Entwicklern die Möglichkeit, kontroverse und/oder neue Algorithmen in diesem Gebiet zu ergründen und testen. Dies ist ein großer Vorteil, da die meisten der bereits existierenden Blockchains sehr stark um eine zentrale Technologie zirkulieren, was die Forschung in diesem Gebiet, ohne eine komplett neue Blockchain zu entwickeln, extrem schwierig macht. Das Framework bietet viele Basis Implementierungen der zentralen Elemente, wie etwa Blocks oder Transaktionen zwischen zwei Teilnehmern in der Chain an. Diese können überschrieben und/oder um eigene Algorithmen und Bedingungen erweitert werden, und erlauben so eine vergleichsweise einfache und schnelle Möglichkeit, neue Ideen zu testen. Zur Zeit des Schreibens gibt es jedoch nicht viel Dokumentation über das Framework und die implementierten Basisklassen, sodass es vor allem für Neulinge in diesem Thema sehr komplex wird, eine eigene Blockchain aufzusetzen.

Sprache: Scala

Vorteile:

* Sehr allgemeines Framework
* Extrem leichtgewichtig (nur 4000 Codezeilen)
* Extrem flexibel
* Durch die Verwendung von Scala (läuft auf der JVM) plattformunabhängig

Nachteile:

* Keine/Kaum Dokumentation verfügbar
* Erfordert viel Vorwissen über Blockchain-Technologie

Anmerkungen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Typ | Programmiersprache | Smart contracts (Programmiersprache) |  |
| BigChainDB | Datenbank | Python | Nein |  |
| Bitcoin | Cryptocurrency | C++ | Nein |  |
| Ethereum | Platform + Cryptocurrency | Go | Ja (Solidity, Vyper) |  |
| Hyperledger Sawtooth | Framework | Go | Ja (?) |  |
| Hyperledger Fabric | Framework | Go | Ja (Go) |  |
| Namecoin | DNS + Cryptocurrency | C++ | Nein |  |
| Scorex | Framework | Scala | Nein |  |