TrustChain

TrustCerts GmbH
info@trustcerts.de

18. Mai 2020

Zusammenfassung

Die TrustChain ist ein Vertrauensdienst, der ein Key-Management-System und einen Signaturspeicher zur Verfügung stellt. Beide Funktionen basieren auf einer dafür konzipierten modularen Blockchain, mit deren Hilfe jegliche Änderungen dokumentiert und nachweisbar sind. Zusätzlich wurde zur Reduzierung von Single Point of Failures und zur Erreichung einer hohen Dezentralität ein ressourcenschonender Konsens implementiert.

1 Einleitung

Digitale Informationen wie Rechnungen, Zertifikate oder Nachweise benötigen einen digitalen Schutz, der nach Möglichkeit flächendeckend anwendbar ist. Entscheidungen sollen zu einem späteren Zeitpunkt auch widerrufen werden können, sodass zukünftige Fehlerkorrekturen möglich sind. Gleichzeitig soll die Sicherheit durch Dezentralität, Transparenz und Überprüfbarkeit gesteigert werden, ohne Einbußen bei Skalierbarkeit oder den nötigen Ressourcen in Kauf nehmen zu müssen.

Bei der TrustChain handelt es sich um eine private permissioned Blockchain, die einen Proof-of-Authority-Konsens mit BFT-Eigenschaften implementiert. Die notwendigen Informationen für die Autorisierung und den Nachweisspeicher werden dabei aus den in der Blockchain gespeicherten Transaktionen abgeleitet.

2 Pre- und Postfunktionen

Bitcoin-Miner berücksichtigen bei der Blockgenerierung nur Transaktionen von Konten, bei denen eine positive Deckung des Kontos vorliegt. Somit landen nur gültige Einträge in der Blockchain, die einen Kontostand nachweislich verändern. Die jeweilige Adressierung ist nur über die Einzahlungsadressen möglich und erschwert die Implementierung von zusammenhängenden Transaktionen, um weitere Anwendungsfälle zu realisieren.

Bei Ethereum werden nach der Persistierung der Blöcke die Eingaben von Smart Contracts ausgewertet und ermöglichen eine automatisierte Zustandsänderung basierend auf definierten Regeln. Allerdings fehlt eine Validierung der Änderungen, bevor diese in Form von Transaktionen persistiert werden. Folglich landen auch ungültige Einträge in der Blockchain, die zu keinen Änderungen der Zustände bei Smart Contracts führen. Der dafür benötigte Speicher muss langfristig belegt werden und erhöht somit die benötigten Ressourcen.

Die TrustChain kombiniert die beiden Stärken der Verfahren durch vorherige Prüfung und späterer Auswertung. Dabei werden nur solche Transaktionen persistent gespeichert, die nachweislich zu einer autorisierten Zustandsänderung führen. Für die erforderliche Prüfung werden nach der Persistierung eines neuen Blocks die Transaktionen interpretiert und die notwendigen Datenbanken aktualisiert.

3 Transaktionstypen

Dank der Unabhängigkeit von anderen Blockchains, kann der modulare Ansatz der Architektur direkt bei den Transaktionen beginnen.

So beinhaltet jede Transaktion einen Hinweis auf ihren Typ und gibt somit Auskunft, wie sie zu interpretieren ist. Die notwendigen Funktionen für die Validierung oder Auswertung werden von den verantwortlichen Modulen eines Knotens bereitgestellt. Eine weitere Ergänzung von Transaktionstypen oder die Änderung des Formats im laufenden Betrieb ist ebenfalls möglich.

4 Dezentrales Key-Management-System

PGP gilt mit seiner nicht vorhandenen Hierarchie als Gegenpol zur klassischen PKI. Die Vorteile der Unabhängigkeit in Form von Anonymität und Souveränität werden bei Kryptowährungen genutzt. Gleichzeitig wird der Single Point of Control umgangen, da kein Root-Zertifikat benötigt wird. Das Vertrauen bei PGP sowie bei anderen blockchainbasierten Lösungen wie Self-Sovereign-Identity muss jeder Nutzer für sich selbst definieren. Für die Implementierung eines Proof-of-Authority-Konsens für die Blockgenerierung sowie die Herleitung einer Chain of Trust ist eine Hierarchie jedoch von Vorteil.

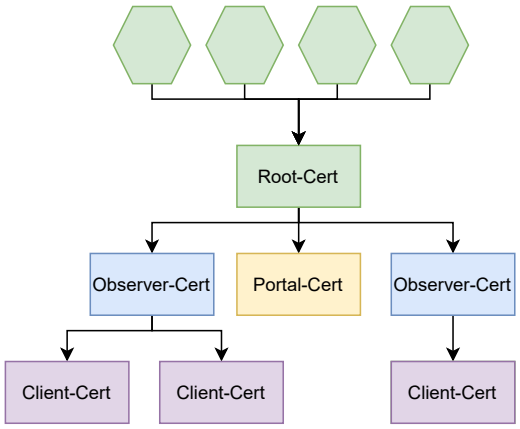

Abbildung 1: *Hierarchie mit geteiltem Root-Zertifikat*

Auf der obersten Ebene befinden sich die Validatoren, die gemeinsam ein Root-Zertifikat generieren. Es beinhaltet die Liste aller öffentlichen Schlüssel der Validatoren. Diese Liste wird anschließend von jedem Validator signiert, sodass das Zertifikat anschließend aus N öffentlichen Schlüsseln und N dazugehörigen Signaturen besteht. Mit der Signatur der Liste überzeugt der Validator die anderen Teilnehmer davon, dass er im Besitz des dazugehörigen privaten Schlüssels ist. Das eigene Schlüsselpaar kann durch eine zusätzliche Beglaubigung durch einen externen Teilnehmer in Form eines X.509-Zertifikats gestärkt werden.

Abgeleitet vom Root-Zertifikat befinden sich auf der nächsten Ebene die Knoten des Netzwerkes, deren öffentliche Schlüssel durch die höhere Ebene beglaubigt werden. Knoten auf dieser Ebene müssen sich nicht gegenseitig vertrauen, sondern verwenden das beidseitig genutzte Zertifikat der Validatoren. Limitierungen zur Nutzung des Schlüssels werden direkt im Zertifikat hinterlegt. So ist jedem Teilnehmer auf dieser Ebene der Zugriff auf das Netzwerk erlaubt, jedoch ist das Erstellen von Transaktionen nur bestimmten Knoten gestattet.

Auf der untersten Ebene befinden sich die Zertifikate der Clients, die mit dem Signaturspeicher interagieren. Durch die Beglaubigung eines Clients übernimmt der Observer die Verantwortung für dessen Handeln. Es liegt also im Interesse des Observer-Betreibers, dass eine ausführliche Identitätsprüfung vor der Beglaubigung durchgeführt wird. Mit einem gültigen Schlüssel kann der Client anschließend auch über andere Observer mit dem Blockchain-Netzwerk interagieren, da das Zertifikat auf allen Knoten lokal verfügbar ist.

4.1 Dezentrale Redundanz

Durch die direkte Integration der PKI in die Blockchain kann jeder Knoten unabhängig von den anderen Teilnehmern den kompletten PKI-Baum lokal herleiten. Die für neue Zertifikate oder Sperrungen notwendigen Informationen sind in der Blockchain in Form von Transaktionen vorzufinden. Im Genesis-Block befindet sich das erste Root-Zertifikat, das den Ausgangspunkt der Chain of Trust definiert. In einem darauffolgenden Block befindet sich die Ausstellung eines Observer-

Zertifikates, und im nächsten Block jenes eines Client-Zertifikates. Die Verkettung der chronologischen Blöcke stärkt das Vertrauen in die Chain of Trust, da per Definition die zur Überprüfung des vorliegenden Zertifikates notwendigen Informationen in einem bereits angehangenen Block zu finden sind.

4.2 Aktualisierung der Hierarchie

Neben der Erweiterung der Hierarchie ist auch das Verändern oder Löschen von Schlüsseln möglich. Als Basis dient die Blockchain als Transaktionsspeicher, über die das Erstellen oder der Widerruf von Zertifikaten realisiert wird. Statt der Führung von separaten Ausstellungs- und Sperrregistern werden die Transaktionen chronologisch interpretiert und der aktuelle Zustand in einer relationalen Datenbank festgehalten. Es ist somit möglich, zu einem definierten Zeitpunkt zu ermitteln, welcher gültige Schlüssel zu einem spezifischen Teilnehmer gehört.

Bei einer Signatur gibt der Aussteller zusätzlich das Zertifikat an, welches die Gültigkeit des Schlüssels beurkundet. Der Prüfer kann anschließend über die Abfrage an einen beliebigen Knoten das Zertifikat des Clients herunterladen und die Signatur mit dem dazu passenden Schlüssel überprüfen. Für die komplette Herleitung der Chain of Trust werden die notwendigen Zertifikate der Aussteller nacheinander oder als Batch heruntergeladen und lokal validiert.

Bei der Neuzuweisung eines Schlüsselpaares, das kompromittiert, verloren oder beschädigt wurde, wird dieses mit Hilfe der Blockchain an die bereits vorhandene Identität geknüpft. Der Nutzer kann anschließend Aktionen mit dem neuen Schlüsselpaar durchführen, die in Abhängigkeit von dem alten Schlüsselpaar standen. Der Widerruf einer Signatur mit einem anderen Schlüsselpaar ist für die anderen Teilnehmer also gerechtfertigt, wenn das neue Schlüsselpaar der gleichen Identität zugeordnet wurde.

Die Verwendung einer Hierarchie ermöglicht es einer oberen Schicht ohne Einverständnis der unteren Schicht ein neues Schlüsselpaar zuzuweisen. Diese Eigenschaft ist bei der Neuzuweisung eines neuen Schlüsselpaares relevant. Im Gegenzug wäre ein Identitätsdiebstahl eines Clients durch einen Observer also möglich. Dieser wäre jedoch für alle Teilnehmer transparent nachverfolgbar, da die Änderung des Schlüsselpaares in der Blockchain dokumentiert ist und zwingt die Teilnehmer zum ordnungsgemäßen Handeln.

5 Signaturspeicher

Einige Dateiformate wie z.B. PDF bieten die Möglichkeit, eine Signatur direkt in der Datei zu speichern, welche dann von unterstützten Programmen beim Öffnen validiert wird. Bei anderen Dateiformaten hingegen muss die Signatur als Anhang mitgesendet werden. Dies gestaltet sich in vielen Prozessen als umständlich, da der Versand zusätzlicher Informationen berücksichtigt werden muss.

Ein Signaturspeicher ermöglicht die Speicherung von Zuständen einer Signatur und nicht nur ihrer bloßen Existenz. Im Falle eines PDF-Dokument müsste immer zusätzlich geprüft werden, ob die Signatur nicht widerrufen wurde. Als Referenz auf die passende Signatur dient die Prüfsumme der Datei, die lokal berechnet werden kann und als eindeutiger Index genutzt wird. Die zusätzliche Speicherung von Transaktionen, in denen die Signatur zu finden ist, wird damit überflüssig. Es muss lediglich bekannt sein, gegen welche Blockchain geprüft werden soll.

5.1 Präventive Maßnahmen

Ein Eintrag im Signaturspeicher beinhaltet aus Datenschutzgründen lediglich die Prüfsumme der Datei. Somit werden keine personenbezogenen oder kritischen Daten vom Client aus über das Internet an fremde Server versendet. Durch Filter in Form von Eingabeüberprüfungen kann vorab sichergestellt werden, dass nur definierte Algorithmen wie SHA-256 oder SHA-3 verwendet werden.

Bei Passwörtern können Webseiten Vorgaben zur Mindestlänge und Komplexität geben, da das Passwort im Klartext vom Server verarbeitet und dort entsprechend geprüft werden kann. Diese Art von Prüfungen können die Knoten nicht vornehmen, da sie die Datei nicht im Klartext vorliegen haben. Aus diesem Grund muss bereits im Vorfeld clientseitig sichergestellt werden, dass die Information einen gewissen Grad an Variation aufweist.

5.2 Recht auf Vergessenwerden

Da die Berechnung der Prüfsumme clientseitig erfolgt und lediglich die resultierende Prüfsumme an das Netzwerk gesendet wird, ist aus Sicht des Knotens nicht ohne Weiteres festzustellen, ob eine erhaltene Prüfsumme tatsächlich aus der Signatur einer Datei entstanden ist, oder ob sie beispielsweise beliebig konzipiert wurde.

Dadurch ist es theoretisch denkbar, dass beliebige Daten in der Blockchain persistiert werden, solange sie den semantischen Regeln einer Prüfsumme folgen. Dadurch können beispielsweise sensible personenbezogene Daten böswillig in der Blockchain hinterlegt werden.

Eine Rekonstruktion der Blockchain ab dem Zeitpunkt, ab dem die Daten persistiert wurden, wäre unter Umständen sehr kostspielig, da zwischen der Persistierung der Daten und dem Bekanntwerden des Vorfalls mehrere Wochen vergehen können. Auch Verfahren wie Sperrlisten sind unzureichend, da sie sich nur auf die Businesslogik, und nicht auf den darunterliegenden Datenspeicher beziehen. Es wird lediglich die Auskunft verweigert, die Daten bleiben allerdings weiterhin vorhanden und können vom Betreiber des Knotens eingesehen werden.

Das von TrustCerts entwickelte Verfahren ist in der Lage die Daten nachträglich zu verändern, ohne die Verkettung der Blöcke zu beschädigen. Grundlage hierfür ist der implementierte Proof-of-Authority-Algorithmus. Da sich das Verfahren in der Patentvorbereitung befindet, kann zum jetzigen Zeitpunkt in diesem White Paper nicht weiter darauf eingegangen werden.

5.3 Transparente Verwendungsnachweise

Bei kompromittierten Schlüsseln ist es nicht möglich, alleine anhand der PKI zu ermitteln, für welche Aktionen das Schlüsselpaar missbraucht wurde. Aus Sicherheitsgründen müssen daher alle Teilnehmer darüber informiert werden, dass sie möglicherweise unautorisierte Signaturen erhalten haben. Der Signaturspeicher gibt exakte Auskunft über die ausgestellten Signaturen. Der Zeitstempel der Transaktionen dient als Filter bei der Eingrenzung der potenziellen Transaktionen. Der Abgleich mit den eigenen Signaturen durch die Prüfsummen identifiziert die Daten, die mit dem gestohlenen Schlüsselpaar signiert wurden, bevor dieses gesperrt wurde. Diese Transaktionen können durch den rechtmäßigen Besitzer der Identität nachträglich widerrufen werden. Prüfsysteme werden bei einer einer erneuten Prüfung der Daten benachrichtigt, dass die Daten widerrufen und somit nicht mehr gültig bzw. nie gültig gewesen sind.

6 Netzwerkstruktur

Im Gegensatz zu Ethereum oder Bitcoin existieren drei verschiedene Arten von Knoten. Dies erlaubt einen modularen Aufbau, sodass das Netzwerk durch seine Aufgabenverteilung effizienter arbeiten kann. Als Vorlage dient das Ring-Modell von Hyperledger Indy, welches erweitert wurde. Die Kommunikation findet immer nur zwischen benachbarten Ringen statt, innerhalb der Ringe findet mit Ausnahme von Ring 1 keine Kommunikation statt. Die Ringverteilung basiert auf den Ebenen der PKI und definiert neben den Verbindungen auch die jeweiligen Verantwortungen.

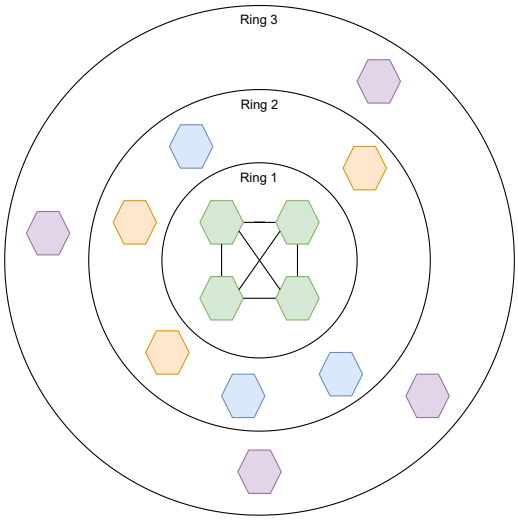

Abbildung 2: *Topologie der Blockchain*

- Ring 1: Validatoren bilden den Kern des Netzwerks
- Ring 2: Observer und Portale agieren als Schnittstellen zum Netzwerk
- Ring 3: Clients kommunizieren vom äußersten Ring aus mit der Blockchain

6.1 Validator

Im innersten Ring befinden sich die Validatoren. Zu ihren Aufgaben zählt das Bilden neuer Blöcke sowie die administrative Verwaltung von Ring 2.

Um eine flächendeckende Synchronisation des Transaktion-Pools zu vermeiden, wird eine Gruppe von Knoten bestimmt, die für das Erstellen von Blöcken verantwortlich ist. Die daraus resultierenden neuen Blöcke werden dann an alle Knoten versendet. Es ist dabei nicht zwingend notwendig, dass alle Validatoren auch bei der Blockgenerierung beteiligt sind. So kann auch nur ein Teil aktiv an dem Konsens teilnehmen, wobei die restlichen Knoten redundant als Backup bereitstehen. Grundlegende Entscheidungen, die das Netzwerk beeinflussen, müssen von mehreren Validatoren vorab bestätigt werden. Je nach Grad der Relevanz kann die Anzahl der benötigten Signaturen für die Transaktion vorab definiert werden, um einen Single Point of Control zu vermeiden. Zum Beispiel ist die Ausstellung eines Zertifikates für einen Knoten nur dann gültig, wenn mindestens zwei Validatoren die Transaktion signiert haben.

6.2 Observer

Die Observer gelten als Gateway zur Blockchain und übernehmen die Verwaltung und Zugriffe von Ring 3.

Neben den Knoten können auch außenstehende Teilnehmer aus Ring 3 Transaktionen erzeugen, die in der Blockchain persistiert werden sollen. Ein Broadcast an das komplette Netzwerk ist nicht ohne weiteres möglich, da den Außenstehenden nicht alle Knoten aus Ring 1 und Ring 2 bekannt sind. Zusätzlich nimmt auch nur ein Teil der Knoten an der Blockgenerierung teil, sodass es hier zu einer unnötigen Netzwerkauslastung kommen würde. Aus diesem Grund ist es die Aufgabe des Observers, eingehende Transaktionen aus Ring 3 an den inneren Ring weiterzuleiten. Zudem kann der Observer die Transaktion vorab auf Fehler untersuchen. Entspricht die Transaktion nicht den vorgegeben Regeln, oder versucht der Nutzer eine unautorisierte Transaktion einzuschleusen, wird diese sofort verworfen.

Sollte ein Observer nicht mehr erreichbar sein, so kann der Client einen anderen Observer nutzen, da alle Observer dieselbe Funktion erfüllen und untereinander austauschbar sind.

6.3 Portal

Ein Portal ist ein Knoten mit reiner Leseberechtigung und dient als Abfrage für interpretierte Transaktionen der Blockchain.

Wie der Observer muss ein Portal einen signierten Schlüssel besitzen, um sich mit den Knoten aus Ring 1 verbinden zu können. Obwohl es sich hier um eine bidirektionale Verbindung handelt, nehmen die Validatoren keine Transaktionen von Portalen an. Ein Portal kann somit bei einer Firma/Institution als reiner Backup-Knoten aufgesetzt werden, der wie alle anderen Knoten die vollständige Blockchain vorliegen hat.

6.4 Client

Ein Client interagiert mit dem Signaturspeicher, wobei die Daten lokal verarbeitet werden.

Ein Client selbst ist nicht im Besitz der kompletten Blockchain, sondern interagiert mit der Blockchain über die Portale und Observer in Ring 2.

Für die Abfrage von Signaturen oder Zertifikaten benötigt der Client kein gültiges Schlüsselpaar. Die Signatur wird durch die Abfrage an den Signaturspeicher des Portals ermittelt, für die Sicherstellung der Chain of Trust werden die benötigten Zertifikate nachgeladen. Da der Client die Blockchain nicht selber lokal besitzt, kann er zusätzlich ein anderes Portal um die Abfrage bitten und die Ergebnisse vergleichen. Die redundante Abfrage verhindert dabei die Gefahr, dass der Client mit einem kompromittierten Portal kommuniziert.

6.5 Kommunikation

Das komplette Netzwerk basiert auf dem P2P-Ansatz, wobei die Verbindungen je nach Ring unterschiedlich sind.

Innerhalb von Ring 1 sind alle Knoten miteinander verbunden und somit vollmaschig untereinander vernetzt. Dies ist notwendig, da die Validatoren an dem Konsens teilnehmen und die Informationen der anderen Teilnehmer direkt beziehen müssen. Falls das Blockchain-Netzwerk nicht in einem geschlossenen Netzwerk wie einem Intranet betrieben werden soll, benötigen allen Validatoren eine öffentliche IP-Adresse.

Zwischen Ring 2 und Ring 1 wird das zustandsbasierte Websocket-Protokoll genutzt. Es baut wie HTTP auf dem TCP-Protokoll auf und ermöglicht einen bidirektionalen Versand der Daten. Somit kann ein Observer eine neue Transaktion in den inneren Ring senden und erhält über denselben Kanal die Antwort in Form eines neu generierten Blocks. Zudem muss der Observer für die Verbindung nicht zwingend eine öffentliche IP-Adresse benutzen, sodass keine Ports nach außen hin geöffnet werden müssen. Alle Teilnehmer aus Ring 2 verbinden sich mit allen Teilnehmern aus Ring 1. Dies ist notwendig, da die Teilnehmer aus Ring 2 nicht wissen, von welchem Validator sie den nächsten Block erhalten werden. Die Autorisierung erfolgt über einen beidseitigen Handshake, für die Signatur der Challenge werden die hinterlegten Schlüssel aus der Blockchain verwendet.

Die Kommunikation von Ring 3 zu Ring 2 basiert auf dem zustandslosen HTTPS-Protokoll mit Hilfe einer REST-Schnittstelle. Entwickler können somit unkompliziert für die Kommunikation ein Standardprotokoll verwenden, welches in den meisten Fällen bereits implementiert ist. Ein Client schickt seine Transaktion nur zu einem Observer, der sie anschließend an die Validatoren verteilt. Für die Validierung von Informationen können beliebig viele Verbindungen zu den Portalen aus Ring 2 aufgebaut werden.

7 TBFT-Konsens

Der Konsens für die Blockgenerierung ist an den Istanbul-BFT angelehnt und ist sowohl crashresistent als auch fehlertolerant. Die Mindestanzahl an Validatoren beträgt 2 für den generellen Ablauf und 4 für die Fehlertoleranz. Das Protokoll ist dabei in 5 Phasen unterteilt, die für einen neuen Block erfolgreich durchlaufen werden müssen. Bei einem Fehler wird die Runde abgebrochen und von vorne begonnen. Bei jeder Runde gibt es zwei verschiedene Rollen: der Proposer ist für die Verteilung von Informationen unter den Teilnehmern im Konsens zuständig. Er koordiniert den Informationsfluss, sodass die Anzahl der versendeten Nachrichten linear verläuft. Würde jeder Validator seine Informationen mit jedem direkt teilen, so wäre der Versand exponentiell. Der Proposer nimmt dabei lediglich temporär die Rolle des eines ansonsten gleichberechtigten Koordinators ein, er kann den anderen Validatoren beispielsweise keine Befehle für den Start einer neuen Runde erteilen.

Start: In dieser Phase ermitteln die Validatoren eigenständig, wer in der folgenden Runde die Rolle des Proposers einnehmen wird. Dazu pflegt jeder Validator eine sortierte Liste der

mit ihm verbundenen Validatoren. Da durch die Vollvermaschung alle Validatoren untereinander verbunden sind, können sie dank der jeweils identischen Sortierung ohne die Hilfe einer zentralen Stelle den ersten Proposer bestimmen. Sobald sich mindestens ein Eintrag im Transaktionspool befindet, startet die Runde. Ist der Index der Liste bei einer neuen Runde bereits gesetzt, wird er um eins erhöht.

Prepare: Der Proposer erstellt einen neuen Block anhand der ihm vorliegenden Transaktionen und versendet sie an alle Validatoren. Die Validatoren erwarten anschließend eine Antwort des Proposers innerhalb eines definierten Zeitfensters. Sollte der Proposer zu langsam oder zwischendurch abgestürzt sein, beenden die Validatoren die Runde und beginnen von vorne. Bei fristgerechtem Erhalt prüfen die Validatoren den vorgeschlagenen Block auf Korrektheit. Dabei werden neben dem vorgeschlagenen Index und Zeitstempel auch die Transaktionen geprüft. Dies ist notwendig, da der Proposer versuchen könnte, falsche Transaktionen in einen Block einzuschleusen. Sollte ein Regelverstoß festgestellt werden, wird die Runde vom Validator abgebrochen.

Commit: Ist der Block valide, signiert der Validator den Inhalt und schickt die Signatur zurück an den Proposer und gibt damit den anderen Validatoren seine Akzeptanz bekannt. Der Proposer gibt hierbei ebenfalls ein Zeitfenster vor, in dem er auf gültige Signaturen wartet. Ist dieses Zeitfenster abgelaufen oder haben alle Validatoren geantwortet, werden die Signaturen validiert. Sind weniger als 66% der Signaturen gültig, wird die Runde abgebrochen, da die Fehlertoleranz nicht erfüllt ist.

Persist: Sind genügend positive Signaturen vorhanden, signiert der Proposer ebenfalls den von ihm vorgeschlagenen Block. Die Liste der Signaturen wird dann an alle anderen Validatoren geschickt. Abermals erwarten die Validatoren die Antwort innerhalb eines definierten Zeitfensters, um einen Deadlock zu vermeiden. Obwohl die Signaturen alle über den Proposer als zentralen Man-in-the-Middle liefen, können sie unabhängig davon die Signaturen durch das redundante Key-Management-System vertrauensvoll prüfen. Sind alle Signaturen gültig, werden sie zusammen mit dem vorgeschlagenen Block zu dem nächsten Block zusammengefasst und an die Blockchain angehangen. Anschließend beginnt automatisch der Start einer neuen Runde.

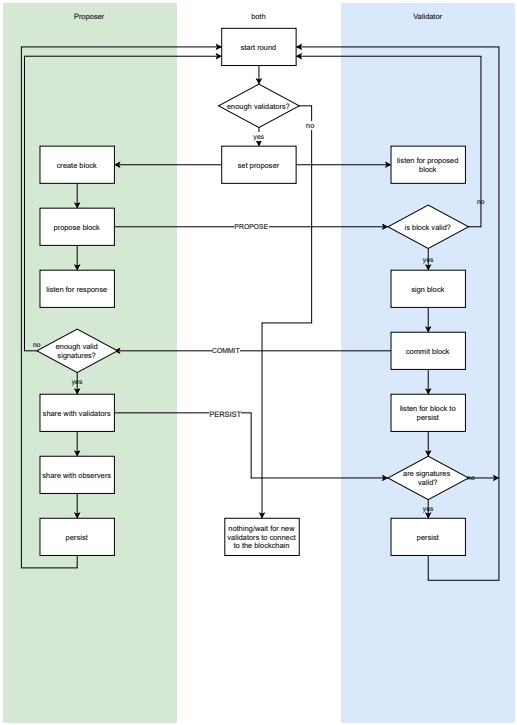

Abbildung 3: *Ablauf des Konsens*