baumann.at – concepts & solutions – DI Dr. Christian Baumann

Dokumenten-Notarisierung “DocNoS” – Spezifikation Datenstruktur

V1.1 (für Private Sector Blockchain)

AUSTRIA PRO

Arbeitskreis Blockchain

Inhalt

1. Einleitung

Mittels „Dokumenten-Notarisierung“ kann bewiesen werden, dass ein elektronisches Dokument zu einem bestimmten Zeitpunkt in einer bestimmten Form existiert hat und seither nicht verändert wurde. Dokumente werden dabei durch ihre „digitalen Fingerabdrücke“ (Hashwerte) identifiziert, d.h. es werden keinerlei (im Klartext lesbare) Daten übertragen, verarbeitet oder gespeichert.

Die Sicherheit und das Vertrauen, dass die hinterlegten Daten nicht manipuliert werden können, wird dabei durch die Blockchain-Technologie gewährleistet.

Dieses Dokument beschreibt die Datenstruktur des Systems „DocNoS“ (Document Notarisation System), welches beim Projekt „Private Sector Blockchain“ (Arbeitstitel) eingesetzt wird, im Rahmen der Initiative „Blockchain-Infrastruktur für die Privatwirtschaft“[1]

Anmerkung: Die „PSBC“ ist das B2B-Pendant der „Austrian Public Service Blockchain“[2], in dieser wird die Dokumentennotarisierung unter dem Titel „Daten-Zertifizierung“[3] betrieben. Die dort verwendete Datenstruktur enthält grundsätzlich dieselben Informationen, ist aber etwas anders aufgebaut, was anderen Anforderungen seitens mancher öffentlicher Stellen geschuldet ist.

2. Organisatorisches

„DocNoS“ ist die erste Anwendung im Rahmen der PSBC. Weitere Anwendungen werden folgen, z.B. auch solche, wo nicht nur Hashwerte in die Blockchain geschrieben werden, sondern auch echte Daten. Dabei müssen natürlich Regeln eingehalten werden (z.B. keine „kritischen“ Daten wie personenbezogene, Gesundheitsdaten usw.) diese Regeln sind gerade in Ausarbeitung.

Im Rahmen der Anwendung „DocNoS“ verpflichten sich alle Teilnehmer der PSBC, dass die von ihren Systemen eingetragenen Daten der jeweils aktuellen Spezifikation (dzt. v1.1) entsprechen.

3. Blockchains und Streams

Im Rahmen der PSBC wird die Blockchainumgebung „MultiChain 2.x“ als Konsortialchain mit „Proof Of Authority“ als Konsensmechanismus eingesetzt. Es stehen derzeit zwei Systeme zur Verfügung:

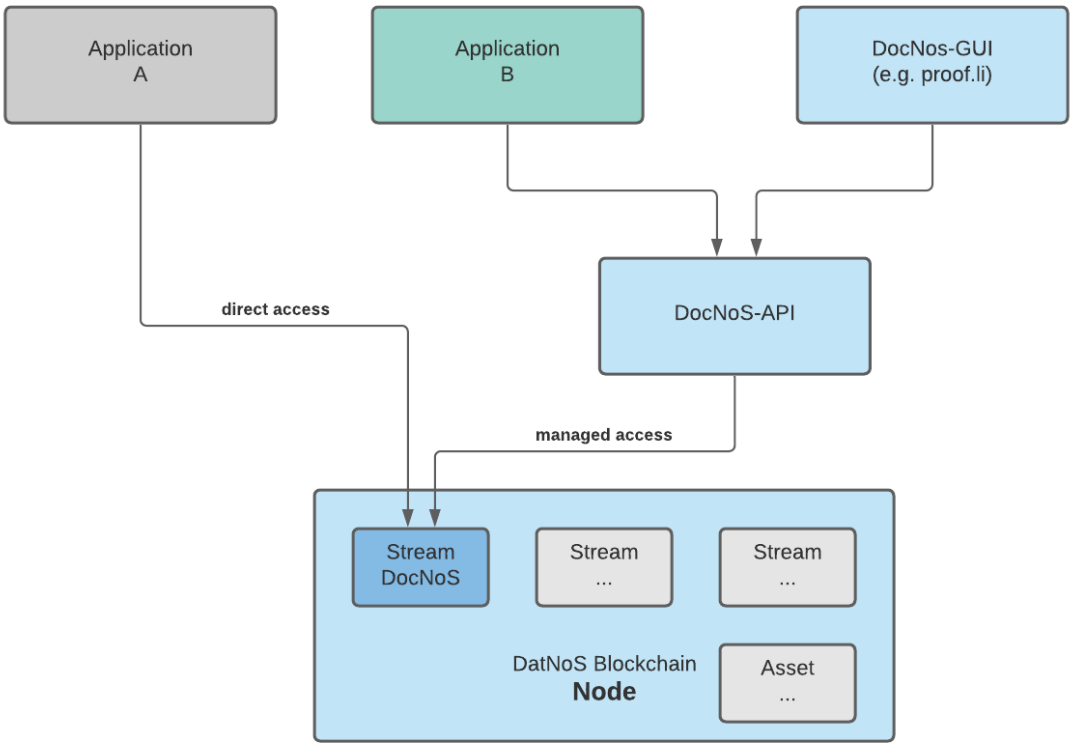
Umgebung	**Chain-Bezeichnung**	**In Betrieb seit**	**Knoten dzt.**	**DocNoS-Stream**	**Anmerkung**
Test	mc2b1	18.1.2019	ca. 15	docnos-test-1	u.a. im Rahmen des AustriaPro Blockchain Labs
Produktiv	datnos-20200220	20.2.2020	ca. 10	docnos	

[1] https://www.wko.at/service/netzwerke/blockchains-oesterreich.html
[2] Vgl. https://www.wko.at/service/netzwerke/blockchain.html
[3] Vgl. https://www.wko.at/service/innovation-technologie-digitalisierung/blockchain.html

4. Zugriffsmethoden

Um die Daten in den Blockchain-Stream zu schreiben (bzw. auch zu suchen/lesen), gibt es grundsätzlich zwei Möglichkeiten:

1) Direkter Zugriff von der Anwendung auf den Blockchain-Node (per Multichain-RPC-API).
2) Zugriff über ein DocNoS-API, welches die Komplexität des Blockchain-Zugriffes kapselt.



Der gemanagte Zugriff über das DocNoS-API[4] ist i.d.R. zu bevorzugen, da das API die DocNoS-Datenstruktur plus alles Keys spezifikationsgemäß erstellt und darüber hinaus den Zugriff von unterschiedlichen Applikationen auf mehrere Blockchain-Nodes bzw. Streams verwalteten und mit Zugriffsberechtigungen schützen kann. Auch kommende Erweiterungen können im DocNoS-API einfacher implementiert werden.

Der direkte Zugriff bietet bei hohen Transaktionszahlen bessere Performance, ist jedoch deutlich komplexer zu implementieren, da er die komplette Datenstruktur plus alle Keys erstellen muss.

Jedenfalls müssen die eingetragenen Daten und Keys den in Folge definierten Bedingungen entsprechen, was zukünftig auch automatisiert geprüft werden kann.

[4] Siehe „DocNoS API Spezifikation v1.40“

5. Aktuelles Datenformat – v1.1

5.1. Daten

Die Daten werden im JSON-Format in den DocNoS-Stream eingetragen und sind nach folgender Struktur aufgebaut:

```
{
    "timeStamp": "2020-10-23T09:52:34+02:00",
    "client": "proof.li\/c2",
    "version": "DocNoS-v1.1",
    "data": {
        "id": "d3ff8bdf-727c-470f-9999-b54317b38fd3",
        "hashes": {
            "sha256":
"2118529b9a93c83d3cc087153f366934d22635c46c97300d08dddf30fb8526ea",
            "sha512":
"a31266ac7ab751702dd5c21608390b75e32f8e5c15c08420c83ba56a4a142ace6d79161b64a7796000
f0ed23a3c960ce6aefd0789746c6791953aac98cbe18d3",
            "sha3\/512":
"e86e575de580b46ecf582da0b6cab2aa215ae9c43525aea8e1822c9202359bd4d2d8b561b3520da603
554e7eeb1870b9d81306e2299dd897d00b18216007830d"
        }
    }
}
```

Feld	Beschreibung	Beispiel
timeStamp	Zeitstempel nach ISO 8601; mandatory	2020-10-23T09:52:34+02:00
client	Kürzel des Systems, welches die Daten generiert; optional	proofl.li/c2
version	Verwendetes Datenformat; mandatory	DocNoS-v1.1
data	Struktur der Dokumentendaten; mandatory	
id	UUID[5], z.B. zur Definition eines Dokumentes; mandatory	d3ff8bdf-727c-470f-9999-b54317b38fd3
hashes	Ein oder mehrere Hashwerte des Dokumentes mit Bezeichnung des verwendeten Verfahrens. Mindestens ein Hashwert (sha256) ist mandatory.	"sha256": "211852…8526ea"

Zur Bezeichnung von Hash-Verfahren werden die entsprechenden Token laut folgender Tabelle verwendet (Kleinschreibung):

Hash-Verfahren	Token
SHA2– prefix „sha“	
Beispiel SHA2 256 Bit	„sha256“ (muss mindestens vorhanden sein)
Beispiel SHA2 512 Bit	„sha512“
SHA3 – prefix „sha3/“	
Beispiel SHA3 512 Bit	„sha3/512“ [6]

[5] Lt. RFC 4122

[6] Bitte beachten: Beim Codieren nach JSON wird dies zu „sha3\/512“

5.2. Keys

In der eingesetzten MultiChain Umgebung können Keys verwendet werden, um Informationen effizienter suchen zu können (vergleichbar mit indizierten Feldern einer Datenbank). Für die zu verwendenden Keys gelten folgende Regeln:

1) Jeder verwendete Hashwert kann als key eingetragen werden, z.B.

```
Key 1 sha256:2118529b9a93c83d3cc087153f366934d22635c46c97300d08dddf30fb8526ea
```

Es muss mindestens der sha256 Hash eingetragen werden.

2) Für die „id“ (UUID) muss ein key angelegt werden, z.B.

```
Key 0 id:d3ff8bdf-727c-470f-9999-b54317b38fd3
```

3) Für die Kurzbezeichnung des Clients kann ein key angelegt werden, z.B.

```
Key 4 proof.li/c2
```

Beispielhafte Darstellung des gesamten Datensatzes incl. Keys:

Key 0	id:d3ff8bdf-727c-470f-9999-b54317b38fd3
Key 1	sha256:2118529b9a93c83d3cc087153f366934d22635c46c97300d08dddf30fb8526ea
Key 2	sha512:a31266ac7ab751702dd5c21608390b75e32f8e5c15c08420c83ba56a4a142ace6d7916
Key 3	sha3/512:e86e575de580b46ecf582da0b6cab2aa215ae9c43525aea8e1822c9202359bd4d2d8b
Key 4	proof.li/c2
JSON data	{ "timeStamp": "2020-10-23T09:52:34+02:00", "client": "proof.li\/c2", "version": "DocNoS-v1.1", "data": { "id": "d3ff8bdf-727c-470f-9999-b54317b38fd3", "hashes": { "sha256": "2118529b9a93c83d3cc087153f366934d22635c46c97300d08 "sha512": "a31266ac7ab751702dd5c21608390b75e32f8e5c15c08420c8 "sha3\/512": "e86e575de580b46ecf582da0b6cab2aa215ae9c43525aea } } }

5.3. Automatisierte Prüfung der Spezifikation

Es können die hier definierten Regeln innerhalb der Blockchain automatisch von einem „Smart-Filter“ geprüft und fehlerhafte Einträge somit vermieden werden. Der Smart-Filter Code ist aktuell in Ausarbeitung.

6. Ausblick

6.1. Optionale Metadaten der Dokumente

Derzeit ist folgende Erweiterung des Datenformates in Diskussion, mit welcher Metadaten von Dokumenten optional angegeben werden können – Beispiel:

```
{
    "timeStamp": "2020-10-23T09:52:34+02:00",
    "client": "proof.li\/c2",
    "version": "DocNoS-v1.2",
    "meta":{
        "version":"1.4.2",
        "updated_by":"6ca0c3f6-3e38-4463-a639-5aab971df8d2",
        "pid":"ZT_BF_1234"
    },
    "data": {
        "id": "d3ff8bdf-727c-470f-9999-b54317b38fd3",
        "hashes": {
            "sha256": "21185…526ea",
            "sha512": "a31266a…cbe18d3"
        }
    }
}
```

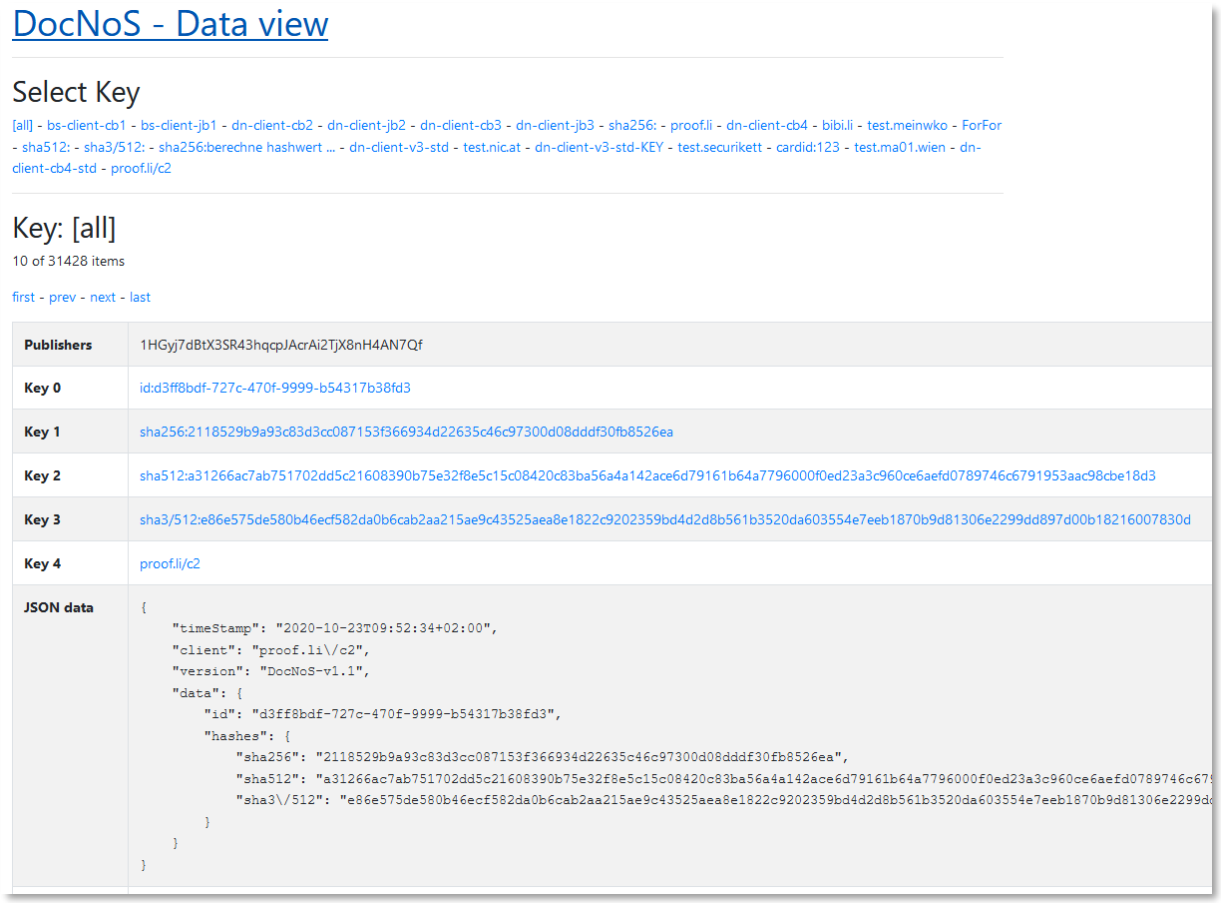
Wesentlich dafür ist eine klare Definition, dass keinesfalls „kritische“ Daten im Blockchainsystem gespeichert werden, z.B. personenbezogene Informationen (wg. Recht auf Löschung lt. DSGVO) etc..

6.2. Hashwerte

Weitere optionale Hashwerte können in kommenden Versionen hinzugefügt werden.

7. Anhang: Testsystem

Im Testsystem können die in der Blockchain verspeicherten Daten über ein Web-GUI abgerufen werden, der Url dazu lautet https://blockchains.web-lab.at/docnos-view



DocNoS - Data view

Select Key

[all] - bs-client-cb1 - bs-client-jb1 - dn-client-cb2 - dn-client-jb2 - dn-client-cb3 - dn-client-jb3 - sha256: - proof.li - dn-client-cb4 - bibi.li - test.meinwko - ForFor - sha512: - sha3/512: - sha256:berechne hashwert ... - dn-client-v3-std - test.nic.at - dn-client-v3-std-KEY - test.securikett - cardid:123 - test.ma01.wien - dn-client-cb4-std - proof.li/c2

Key: [all]

10 of 31428 items

first - prev - next - last

Publishers	1HGyj7dBtX3SR43hqcpJAcrAi2TjX8nH4AN7Qf
Key 0	id:d3ff8bdf-727c-470f-9999-b54317b38fd3
Key 1	sha256:2118529b9a93c83d3cc087153f366934d22635c46c97300d08dddf30fb8526ea
Key 2	sha512:a31266ac7ab751702dd5c21608390b75e32f8e5c15c08420c83ba56a4a142ace6d79161b64a7796000f0ed23a3c960ce6aefd0789746c6791953aac98cbe18d3
Key 3	sha3/512:e86e575de580b46ecf582da0b6cab2aa215ae9c43525aea8e1822c9202359bd4d2d8b561b3520da603554e7eeb1870b9d81306e2299dd897d00b18216007830d
Key 4	proof.li/c2
JSON data	{ "timeStamp": "2020-10-23T09:52:34+02:00", "client": "proof.li\/c2", "version": "DocNoS-v1.1", "data": { "id": "d3ff8bdf-727c-470f-9999-b54317b38fd3", "hashes": { "sha256": "2118529b9a93c83d3cc087153f366934d22635c46c97300d08dddf30fb8526ea", "sha512": "a31266ac7ab751702dd5c21608390b75e32f8e5c15c08420c83ba56a4a142ace6d79161b64a7796000f0ed23a3c960ce6aefd0789746c67 "sha3\/512": "e86e575de580b46ecf582da0b6cab2aa215ae9c43525aea8e1822c9202359bd4d2d8b561b3520da603554e7eeb1870b9d81306e2299d } } }

Auf der Seite werden alle in der Blockchain zu den Transaktionen vorhandenen Daten dargestellt, ähnlich einem Block-Explorer.

---

baumann.at - concepts & solutions
DI Dr. Christian Baumann
e-Mail: c.baumann@baumann.at
Tel.: +43 664 43 24 243
Web: http://www4.baumann.at
