

Doichain (DOI) und Wrapped Doichain (wDOI)

Technische Integrationsspezifikation für Börsen

Version 2026.1, 17. September 2026. Gilt für Doichain Core v31.1.5 und neuer (aktuelles Release v31.1.6).

Kontakte: technische Integration Nico Krause (Doichain@le-space.de), Listing und Kommerzielles Ottmar Neuburger, DOI Labs AG (ottmar.neuburger@webanizer.de). Integrationsfragen bitte per E-Mail, Antwort innerhalb eines Werktags.

1. Umfang und Zusammenfassung

Dieses Dokument gibt einer Börse alles, was sie braucht, um einen Doichain-Node zu betreiben, Einzahlungen gutzuschreiben, Auszahlungen abzuwickeln und das Netzwerk zu überwachen, sowie den ERC-20-Token wDOI auf Ethereum zu integrieren. Doichain ist ein Derivat von Bitcoin Core 31. Wer Bitcoin, Namecoin oder einen anderen Coin der Bitcoin-Familie integriert hat, findet RPC-Schnittstelle, Transaktionsformat und Wallet-Modell vertraut. Die Doichain-spezifischen Punkte sind:

- 1. Merged Mining (AuxPoW).** Block-Header tragen einen zusätzlichen Arbeitsnachweis aus Bitcoin und sind deshalb größer als 80 Byte. Das ist nur relevant, wenn Sie Header selbst parsen.
- 2. Namensoperationen.** Manche Outputs tragen Namens-Scripts im Stil von Namecoin. Sie sind keine Zahlungen und müssen aus der Einzahlungserkennung ausgeschlossen werden.
- 3. Das Upgrade vom September 2026.** Die Kette hat sich bei Block 431017 geteilt. Alt-Nodes folgen einer anderen Kette. Eine Startprüfung ist Pflicht, und Coins aus der Zeit vor dem Fork haben keinen Replay-Schutz.
- 4. Kein Gebührenmarkt.** Die Gebührenschatzung liefert keine Daten. Verwenden Sie eine feste Gebührensrate.

Kurzübersicht:

Punkt	Wert
Kürzel	DOI
Dezimalstellen	8 (1 DOI = 100.000.000 Satoshi-Äquivalente)
Referenzclient	Doichain Core v31.1.6, https://github.com/Doichain/doichain-core/releases
Docker-Image	doichain/core:v31.1.6 (auch latest), linux/amd64
P2P-Port / RPC-Port	8338 / 8339
Blockabstand	600 s Ziel
Empfohlene Bestätigungen für Einzahlungen	12 (siehe Abschnitt 6.3)
Coinbase-Reife	100 Blöcke
Adressformate	N... (P2PKH), 6... (P2SH), dc1q... (Bech32 SegWit v0)
Kettenprüfung	getblockhash 431018 = 71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67
Explorer	https://doi-explorer.le-space.de

Punkt	Wert
wDOI	ERC-20 auf dem Ethereum-Mainnet, 0xA2B6c1a7EFB3dFa75E2d9DF1180b02668c06da72, 18 Dezimalstellen

2. Netzwerkparameter (Mainnet)

Quelle: `src/kernel/chainparams.cpp` und zugehörige Dateien, Tag v31.1.5. Die Chainparams-Datei ist in v31.1.6 byteidentisch.

2.1 Identifikation

Parameter	Wert	Quelle
Netzwerk-Magic (pchMessageStart)	f8 b2 b2 ff	chainparams.cpp:191
Standard-P2P-Port	8338	chainparams.cpp:195
Standard-RPC-Port	8339	chainparamsbase.cpp:47
Genesis-Block-Hash	000006fdd8b4d786fd9bdde5bae9486c464e3aa4336c5f8415 dfdd3fc1679134	chainparams.cpp:202
Genesis-Merkle-Root	234651063df5f8b01ecc2fc3a134fa1cb9dc9da9cce0149049 483ba1b1469dfb	chainparams.cpp:203
Genesis-Zeit	1522756358 (3. April 2018, 11:52:38 UTC)	chainparams.cpp:200
DNS-Seeds	dnsseed.doichain.org, seed.doi.works (liefern unter Umständen keine Adressen, dann Fixed Seeds nutzen)	chainparams.cpp:210
Fixed Seeds (Port 8338)	94.130.57.196, 116.203.99.217, 128.140.44.20, 195.201.129.97, 88.99.189.178, 128.140.7.173, 65.108.176.156, 88.99.175.240	contrib/seeds/ nodes_main.txt
Präfix für Nachrichtensignaturen	Doichain Signed Message:\n	Wallet-Konstanten

2.2 Adressen und Schlüssel

Parameter	Wert	Quelle
Base58 P2PKH-Version	52 (0x34), Adressen beginnen mit N	chainparams.cpp: 213
Base58 P2SH-Version	13 (0x0d), Adressen beginnen mit 6	chainparams.cpp: 214
WIF-Version für private Schlüssel	180 (0xB4)	chainparams.cpp: 215
BIP32 Extended Keys	0488B21E / 0488ADE4 (wie Bitcoin, xpub/xprv)	chainparams.cpp: 216
Bech32 Human-readable Part	dc	chainparams.cpp: 220
SegWit	aktiv seit Block 216.500 (P2WPKH und P2WSH)	chainparams.cpp: 128

Parameter	Wert	Quelle
Taproot	nicht aktiviert (NEVER_ACTIVE), dc1p...-Adressen sind ungültig	chainparams.cpp:157
SLIP-0044 Coin Type	7070 (Pfad m/44'/7070'/...)	SLIP-0044-Register

Prüfregeln für von Nutzern eingegebene Auszahlungsadressen:

- Base58Check mit Versionsbyte 0x34 oder 0x0d, 25 Byte dekodiert, gültige Prüfsumme. Regulärer Ausdruck als Vorfilter: `^[N6][1-9A-HJ-NP-Za-km-z]{33}$`.
- Bech32 (BIP173, nicht Bech32m) mit HRP dc, Witness-Version 0, Programmlänge 20 (P2WPKH) oder 32 (P2WSH). Nur Kleinbuchstaben. Regulärer Ausdruck: `^dc1q[02-9ac-hj-np-z]{38}$` oder `^dc1q[02-9ac-hj-np-z]{58}$`.
- Witness-Version 1 und höher ablehnen.

`validateaddress` auf dem Node führt die vollständige Prüfung aus und liefert `isvalid`.

2.3 Konsens und Proof of Work

Parameter	Wert	Quelle
Hashfunktion	SHA-256d	Bitcoin-Familie
Mining	AuxPoW Merged Mining, Chain-ID 0x0002, ab Block 1, <code>fStrictChainId = false</code>	chainparams.cpp:179
Verwendete Blockversion	0x00020104 (Chain-ID 2, AuxPoW-Flag, Version 4)	Header-Probe, Block 431018
Header-Größe	variabel, typisch 0,6 bis 2,3 kB (80-Byte-Header plus AuxPoW)	Release-Notes v31.1.5
Ziel-Blockabstand	600 s	chainparams.cpp:146
Difficulty-Algorithmus	DigiShield v3 ab Block 431017: Mittelungsfenster 17 Blöcke, höchstens +16 % / -32 % je Block	chainparams.cpp:135, params.h:221
Difficulty-Reset	erste 28 Blöcke nach 431017 mit <code>nBits 0x1a0400cd</code>	chainparams.cpp:139
Notventil	nach 3.600 s ohne Block darf der nächste Block höchstens viermal leichter sein	chainparams.cpp:140
Legacy-Retarget	alle 2.016 Blöcke über 14 Tage, gültig bis Block 431016	chainparams.cpp:145
<code>powLimit</code>	<code>0000ffff...</code>	chainparams.cpp:144
Maximales Blockgewicht	4.000.000	consensus.h:13
Maximales Standard-Transaktionsgewicht	400.000	policy.h:37
Coinbase-Reife	100 Blöcke	consensus.h:19
Checkpoints	keine	chainparams.cpp
<code>defaultAssumeValid</code>	4f5e8c0e4efb3504f8923ea175e4e5e688963819dcf3e33cc7f5a28c33616823 (Block 431016)	chainparams.cpp:177

2.4 Geldmengenregeln

Parameter	Wert	Quelle
COIN	100.000.000	amount.h:15
MAX_MONEY	21.000.000 DOI	amount.h:26

Parameter	Wert	Quelle
Anfängliche Blockbelohnung	50 DOI	validation.cpp:1945
Halving-Intervall	210.000 Blöcke	chainparams.cpp:118
Aktuelle Blockbelohnung	12,5 DOI (Blöcke 420.000 bis 629.999)	abgeleitet
Umlaufmenge bei Block 432.057 (17.09.2026)	15.901.912 DOI (Explorer), 15.900.725 DOI theoretisch	doi-explorer.le-space.de/ext/getmoneysupply

2.5 Standardwerte der Gebührenpolitik

Parameter	Wert	Quelle
DEFAULT_MIN_RELAY_TX_FEE	100.000 sat/kvB = 0,001 DOI/kvB	policy.h:69
DUST_RELAY_TX_FEE	3.000 sat/kvB	policy.h:67
DEFAULT_INCREMENTAL_RELAY_FEE	100 sat/kvB	policy.h:47
Öffentliche ElectrumX-Server melden relayfee	0,00001 DOI/kB (ihre Backend-Nodes laufen mit niedrigerer minrelaytxfee)	Live-Messung 17.09.2026
estimatesmartfee / ElectrumX estimatefee	liefert keine Schätzung (-1), weil es keinen Gebührenmarkt gibt	Live-Messung 17.09.2026

Praktische Folge: Eine Transaktion mit 0,001 DOI/kvB (100 sat/vB) wird von jedem Node mit Standardeinstellungen angenommen und in den nächsten Block aufgenommen. Die Mining-Nodes setzen diese Untergrenze durch, obwohl die öffentlichen ElectrumX-Server günstigere Transaktionen weiterleiten, solche Transaktionen bleiben im Mempool liegen und werden nie geschürft (Release-Notes Electrum-DOI dc4.1.7, 18. September 2026). Typische Transaktionen kosten zwischen 0,0002 und 0,0005 DOI. Eine am 17. September 2026 beobachtete Zahlung mit 226 Byte zahlte 0,000146 DOI und wurde im nächsten Block bestätigt.

2.6 Namensoperationen (müssen erkannt werden)

Doichain erbt die Namens-Scripts von Namecoin und ergänzt ein eigenes.

Opcode	Bedeutung	Script-Muster
OP_1	name_new	OP_1 <hash> OP_2DROP <p2pkh/p2wpkh>
OP_2	name_firstupdate	OP_2 <name> <rand> <value> OP_2DROP OP_2DROP <p2pkh/p2wpkh>
OP_3	name_update	OP_3 <name> <value> OP_2DROP OP_DROP <p2pkh/p2wpkh>
OP_10	name_doi (Doichain)	OP_10 <name> <value> OP_2DROP OP_DROP <p2pkh/p2wpkh>

Namens-Transaktionen tragen die von Namecoin geerbte Transaktionsversion 0x7100 (28928), woran sie sich schnell erkennen lassen. Ein Namens-Output enthält mindestens 0,01 DOI (MinNameCoinAmount, params.h:60). Dieser Betrag wird weder verbrannt noch ist er eine Gebühr: Er bleibt im Namens-UTXO gebunden, wandert bei jeder Aktualisierung oder Übertragung mit dem Namen mit und wird erst frei, wenn der Name abläuft (36.000 Blöcke nach der letzten Operation). Der scriptPubKey beginnt mit dem Opcode und endet mit einem gewöhnlichen Zahlungs-Script.

Byte-Layout einer echten Mainnet-Registrierung des Namens test mit dem Wert hello an einen P2PKH-Eigentümer (Version 0x7100):

```
5a          OP_10 (OP_NAME_DOI)
```

```

04 74657374      push "test"
05 68656c6c6f    push "hello"
6d 75            OP_2DROP OP_DROP
76 a9 14 <20 Byte> 88 ac P2PKH des Eigentümers

```

ElectrumX indexiert Namen unter dem Script-Hash sha256(0x53 || push(name) || 0x00 || 6d 75 6a) (Bytes umgekehrt), abfragbar mit `blockchain.scripthash.get_history`. Der Index liefert auch abgelaufene und überholte Einträge, den Ablauf muss der Client gegen die Kettenspitze berechnen. `name_doi` registriert in einem Schritt ohne das Commit-Reveal von Namecoin (`name_new`, `name_firstupdate`), eine im Mempool wartende Registrierung kann daher überholt werden. Für den Börsenbetrieb ist das unerheblich, erklärt aber, warum Namens-Outputs im Mempool auftauchen und wieder verschwinden können.

Die Einzahlungserkennung muss jeden Output ignorieren, dessen Script mit OP_1, OP_2, OP_3 oder OP_10 beginnt, sonst werden die 0,01-DOI-Pfänder als Zahlungen an die Adresse gutgeschrieben, die das nachfolgende Script enthält. Hot Wallets einer Börse müssen nie Namens-Outputs erzeugen.

2.7 Soft-Fork-Status

Funktion	Aktivierung
P2SH (BIP16)	Block 0
BIP65, BIP66	Block 130.000
CSV (BIP68, 112, 113)	Block 216.500
SegWit (BIP141, 143, 147)	Block 216.500
Taproot (BIP340 bis 342)	nie
BIP34	nicht erzwungen (Höhe 100.000.000)

3. Node-Betrieb

3.1 Software

Verwenden Sie Doichain Core **v31.1.6** oder neuer von <https://github.com/Doichain/doichain-core/releases>. Bereitgestellt werden statische Linux-x86_64-Binaries (musl) und ein Docker-Image. Release-Tags sind mit dem Schlüssel 52A2 1926 E3F2 74AA 1F2D 9CDA 9BA0 ED44 F14E 8118 GPG-signiert. Prüfen Sie `SHA256SUMS.asc` vor dem Einsatz.

Verwenden Sie **kein** Release der Reihen 0.20.x oder 0.16.x und kein Datenverzeichnis, das je mit einem solchen Release synchronisiert war. Diese Nodes folgen der Altkette (Abschnitt 7). Eine frische Synchronisation ab Genesis dauert auf aktueller Hardware wenige Minuten, weil die Kette klein ist (rund 2,7 Millionen Transaktionen).

Wer ein fertiges Deployment bevorzugt: Das Projekt pflegt einen Referenz-Stack als Docker Compose unter <https://github.com/Doichain/doichain-install> (Abschnitt 3.5). Dessen `docker-compose-mining.yml` startet Doichain Core, einen ElectrumX-Indexer, einen Merged-Mining-p2pool und einen gekürzten Bitcoin-Node. Eine Börse braucht davon nur den Core und optional den ElectrumX-Dienst.

3.2 Hardware

Ressource	Empfehlung
CPU	2 Kerne
RAM	4 GB (mindestens 2 GB)
Speicher	20 GB SSD (Kette rund 5 GB inklusive Indizes, langsam wachsend)
Netzwerk	stabile ausgehende Verbindung zu Peers auf Port 8338

3.3 Konfiguration

Beispiel `doichain.conf`:

```
server=1
daemon=1
txindex=1
rpcuser=<benutzer>
rpcpassword=<starkes passwort>
rpcport=8339
rpcbind=127.0.0.1
rpccallowip=127.0.0.1
listen=1
port=8338
maxconnections=40
# Fixed Seeds, falls die DNS-Seeds nichts liefern
addnode=94.130.57.196:8338
addnode=116.203.99.217:8338
addnode=128.140.44.20:8338
addnode=195.201.129.97:8338
addnode=88.99.189.178:8338
addnode=128.140.7.173:8338
addnode=65.108.176.156:8338
addnode=88.99.175.240:8338
# Gebührenpolitik (siehe Abschnitt 6.4)
minrelaytxfee=0.001
paytxfee=0.001
# Wallet
wallet=exchange_hot
```

Docker-Äquivalent:

```
docker run -d --name doichain -v /srv/doichain:/root/.doichain -p 8338:8338
doichain/core:v31.1.6
```

Das offizielle Image schreibt `doichain.conf` beim **ersten** Start aus den Umgebungsvariablen `RPC_PASSWORD` (Pflicht) und `RPC_ALLOW_IP`, erzeugt `bind=` für das aktive Netz und kündigt die in `EXTERNAL_IP` angegebene Adresse an. Es veröffentlicht nur den P2P-Port, der RPC-Port bleibt absichtlich im Docker-Netz. Zusätzliche Daemon-Argumente lassen sich über `DOICHAIN_EXTRA_ARGS` übergeben. Die Docker-Hub-Tags `v31.1.5` und `v31.1.6` sind aus den signierten Release-Tags gebaut.

Doichain Core 31 verwendet Descriptor-Wallets (SQLite). Alte Berkeley-DB-Wallets aus früheren Releases müssen einmalig mit `migratewallet` migriert werden.

3.4 Verpflichtende Startprüfung

Sobald der Node in `getblockchaininfo` den Wert `initialblockdownload: false` meldet, führen Sie aus:

```
doichain-cli getblockhash 431018
```

Erwartet: `71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67`.

Liefert der Node `bab49c132328d09664261c3061608442408d4f667eaa457ed874b879923f2d34`, steht er auf der Altkette. Stoppen Sie ihn, löschen Sie das Datenverzeichnis und synchronisieren Sie mit dem aktuellen Release neu. Automatisieren Sie diese Prüfung in Ihrem Monitoring und alarmieren Sie, falls sich das Ergebnis je ändert.

Weitere Plausibilitätsprüfungen: `getblockchaininfo` zeigt `chain: main`, die Höhe liegt innerhalb weniger Blöcke von `https://doi-explorer.le-space.de/api/getblockcount`, und `getnetworkinfo` zeigt `subversion /Satoshi:31.1.x/` (der User-Agent-String ist in der Reihe 31.1 noch nicht auf Doichain umbenannt, ein Alt-Node meldet `/Satoshi:0.20.x/`).

3.5 Referenz-Stack (doichain-install)

`https://github.com/Doichain/doichain-install` ist das Referenz-Deployment des Projekts, im September 2026 für Core 31.1 aktualisiert (Pull Requests 5 bis 9, Dokumentation in `docs/stack-31.1.md`). Relevante Fakten für eine Börse:

Punkt	Wert
Compose-Datei	<code>docker-compose-mining.yml</code> (Dienste <code>bitcoin-init</code> , <code>bitcoin</code> , <code>doichain</code> , <code>p2pool</code> , <code>electrumx</code>)
Core-Image	<code>doichain/core:v31.1.5</code> in der Compose-Datei festgelegt, <code>v31.1.6</code> auf Docker Hub ist ein Drop-in-Ersatz
ElectrumX-Image	<code>doichain/electrumx:v2.0.0-doi1</code> , gebaut aus <code>Doichain/electrumx</code> Tag <code>v2.0.0-doi1</code> (Commit 11877d7b), Upstream ElectrumX 2.0.0 mit den Doichain-Coin-Klassen (AuxPoW, SegWit, Namensindex inklusive <code>name_doi</code>)
ElectrumX-Protokoll	1.4 bis 1.7.0, Namens-RPC <code>blockchain.name.get_value_proof</code> ab Protokoll 1.4.3
ElectrumX-Datenbank	RocksDB (DB_ENGINE=rocksdb, seit 2.0 Pflicht), ein mit dem 1.15-Image gebauter Index lässt sich nicht migrieren, das Volume <code>electrumx-db</code> muss leer starten
Indexdauer aus leeren Volumes	rund 22 Minuten für die gesamte Kette (gemessen 16.09.2026)
ElectrumX-Ports	50001 TCP, 50002 SSL, 50004 WSS, selbstsigniertes Zertifikat beim ersten Start, für den öffentlichen Betrieb eigenes Zertifikat einbinden
Daemon-Verbindung	<code>DAEMON_URL=http://<benutzer>:<passwort>@doichain:8339/</code> (expliziter Port vom Entrypoint verlangt)
Voraussetzungen	Docker Engine 20.10+, Docker Compose v2.17+ (der ElectrumX-Dienst nutzt <code>additional_contexts</code>)

Von den Maintainern genannte offene Punkte: Die erzeugte `doichain.conf` enthält noch ein zu weites `rpccallwip`, das eingeschränkt werden sollte, die Images `p2pool` und `bitcoind` auf Docker Hub sind ältere Builds, die die Compose-Datei zur Laufzeit ausbessert, und der User-Agent des Nodes lautet noch `Satoshi`. Keiner dieser Punkte berührt Konsens oder RPC-Schnittstelle.

4. RPC-Schnittstelle

Die RPC-Schnittstelle entspricht Bitcoin Core 31 (JSON-RPC über HTTP, Port 8339). Alle Standardaufrufe verhalten sich wie in Bitcoin Core: `getblockchaininfo`, `getblockcount`, `getblockhash`, `getblock`, `getblockheader`, `getrawtransaction`, `decoderawtransaction`, `sendrawtransaction`, `getmempoolinfo`, `getrawmempool`, `listsinceblock`, `listtransactions`, `gettransaction`, `getnewaddress`, `getaddressinfo`, `validateaddress`, `listunspent`, `createrawtransaction`, `fundrawtransaction`, `signrawtransactionwithwallet`, `walletcreatefundedpsbt`, `walletprocesspsbt`, `sendtoaddress`, `sendmany`, `send`, `estimatesmartfee` (liefert keine Schätzung, siehe 6.4), `getnetworkinfo`, `getpeerinfo`.

Doichain-spezifische Ergänzungen (für den Börsenbetrieb nicht nötig): `name_show`, `name_history`, `name_list`, `name_new`, `name_firstupdate`, `name_update`, `name_doi`, `name_scan`, `name_pending`. `bumpfee` und `psbtbumpfee` verweigern Transaktionen mit Namens-Outputs.

Wissenswerte Unterschiede:

- `getblock` mit Verbosity 2 liefert die AuxPoW-Daten in einem Objekt `auxpow` des Headers. Der hash eines Blocks ist wie üblich der Hash seines 80-Byte-Kern-Headers.
- `getblockheader` in Hex-Form liefert den vollständigen serialisierten Header inklusive AuxPoW, seine Länge variiert daher.
- `help` war in v31.1.2 bis v31.1.5 defekt (RPCHelpMan-Fehler) und ist in v31.1.6 repariert. Die RPC-Aufrufe selbst funktionierten in allen Versionen.
- Beträge sind DOI mit 8 Dezimalstellen, genau wie BTC in Bitcoin Core.
- Wie in Bitcoin Core seit Version 22 enthält `scriptPubKey` in dekodierten Transaktionen einen einzelnen String `address`, das alte Array `addresses[ ]` aus Doichain Core 0.20.x gibt es nicht mehr. Werkzeuge für den Altclient müssen angepasst werden (`spk.address ?? spk.addresses?.[0]`).
- Namens-Outputs erscheinen in dekodierten Transaktionen mit einem zusätzlichen Objekt `nameOp` (`op, name, value`), das nachfolgende Zahlungs-Script wird wie üblich gemeldet. Filtern Sie deshalb auf `nameOp` oder auf den führenden Opcode, nicht auf die Adresse.

5. Block- und Transaktionsformat

Punkt	Wert
Block-Header	80-Byte-Bitcoin-Header, gefolgt von der AuxPoW-Struktur (Coinbase-Transaktion des Parent-Blocks, Coinbase-Merkle-Pfad, Chain-Merkle-Pfad, Parent-Block-Header). In jedem Block seit Höhe 1 vorhanden.
Block-Hash	SHA-256d des 80-Byte-Headers
Transaktionen	Bitcoin-Format inklusive SegWit (BIP141). <code>txid</code> und <code>wtxid</code> wie in Bitcoin.
Script	Bitcoin Script plus Namens-Opcodes (Abschnitt 2.6). Kein Taproot.
Locktime, Sequence, RBF	wie in Bitcoin Core 31: Der Mempool wendet Full Replace-by-Fee an, jede unbestätigte Transaktion kann unabhängig von ihren Sequence-Nummern durch eine konkurrierende Transaktion mit höherer Gebühr ersetzt werden. Einzahlungen nie bei null Bestätigungen gutschreiben.
Serialisierung für das Hashing	AuxPoW ist nicht Teil des gehashten Headers, Standard-Bibliotheken für Bitcoin-Header-Hashing funktionieren, wenn Sie die AuxPoW-Daten abtrennen.

Wenn Sie einen Indexer eines Drittanbieters nutzen (etwa ElectrumX oder einen Block-Parser), muss er AuxPoW-Header verstehen. Die öffentlichen ElectrumX-Server (Abschnitt 8.2) tun das bereits.

6. Einzahlungen und Auszahlungen

6.1 Adresserzeugung

Erzeugen Sie Einzahlungsadressen mit `getnewaddress "" bech32` (bevorzugt, niedrigste Gebühren) oder `getnewaddress "" legacy`. Verwenden Sie nicht `bech32m`. Beide Formate werden von allen aktuellen Doichain-Wallets unterstützt. Öffentliche ElectrumX-Server und der Explorer indexieren beide.

6.2 Erkennen von Einzahlungen

Rufen Sie bei jedem neuen Block `listsinceblock <letzterblockhash> 1 true` ab oder abonnieren Sie mit `-blocknotify`. Für jede empfangene Transaktion:

1. `gettransaction <txid>` oder `getrawtransaction <txid> true` abrufen.
2. Nur Outputs gutschreiben, deren `scriptPubKey.type` `pubkeyhash`, `scripthash`, `witness_v0_keyhash` oder `witness_v0_scripthash` ist und deren Adresse Ihnen gehört.
3. Outputs verwerfen, deren Script mit `OP_1`, `OP_2`, `OP_3` oder `OP_10` beginnt (Namensoperationen), unabhängig von der nachfolgenden Adresse.
4. Coinbase-Outputs erst nach 100 Bestätigungen als ausgebbar behandeln (`COINBASE_MATURITY`).

Empfohlene Mindesteinzahlung: 1 DOI (deutlich über Dust- und Namensbeträgen).

6.3 Bestätigungstiefe

XT.com schreibt DOI-Einzahlungen nach 10 Bestätigungen gut. Wir empfehlen **12 Bestätigungen** aus folgenden Gründen: Die Netzwerk-Hashrate ist klein (rund 3,4 PH/s im September 2026), ein einzelner Merged-Mining-Pool produziert derzeit die meisten Blöcke, und der Difficulty-Algorithmus ist seit dem 11. September 2026 im Einsatz. Zwölf Blöcke dauern beim Zielabstand rund zwei Stunden. Für sehr große Einzahlungen erwägen Sie 24 Bestätigungen. Reduzieren Sie die Tiefe, sobald sich die Hashrate breiter verteilt hat, und schreiben Sie nie unter 6 Bestätigungen gut.

6.4 Gebühren für Auszahlungen

Es gibt keine funktionierende Gebührenschatzung. `estimatesmartfee` liefert keine `feerate`, und die ElectrumX-Server antworten auf `estimatefee` mit `-1`. Verwenden Sie eine **feste Gebührensrate**:

- Setzen Sie `paytxfee=0.001` (0,001 DOI je kB, 100 sat/vB) in `doichain.conf` oder übergeben Sie `fee_rate=100` (sat/vB) an `send/sendtoaddress`.
- Das entspricht der Standard-`minrelaytxfee` eines unveränderten Nodes und garantiert Weiterleitung und Aufnahme in den nächsten Block.
- Verlassen Sie sich nie darauf, dass `estimatesmartfee` ein Ergebnis liefert. Wenn Ihr Framework einen Schätzer verlangt, hinterlegen Sie den Rückfallwert fest.
- Eine typische SegWit-Auszahlung mit einem Eingang und zwei Ausgängen hat rund 140 vB und kostet bei dieser Rate 0,00014 DOI.

Als Auszahlungsgebühr gegenüber Nutzern verlangt XT.com 58,7 DOI bei einer Mindestauszahlung von 587,2 DOI. Angesichts der On-Chain-Kosten ist das großzügig, eine Gebühr von 1 bis 5 DOI und ein Minimum von 10 DOI wären mehr als ausreichend.

6.5 Abwicklung von Auszahlungen

Standard-Workflow von Bitcoin Core: `send` oder `sendtoaddress` aus der Hot Wallet, oder PSBT-basiertes Signieren mit einem Offline-Signierer (`walletcreatefundedpsbt`, `walletprocesspsbt`, `finalizepsbt`).

sendrawtransaction). Bündeln Sie Auszahlungen wo möglich mit sendmany. Warten Sie mindestens 1 Bestätigung, bevor Sie eine Auszahlung als abgeschlossen anzeigen, und behalten Sie die übliche Reorg-Behandlung (Nachprüfung nach 6 Blöcken).

6.6 Replay-Vorsicht bei Coins aus der Zeit vor dem Fork

Die Blöcke bis 431017 sind mit der Altkette gemeinsam, und es gibt keinen Replay-Schutz. Eine Transaktion, die **ausschließlich** UTXOs ausgibt, die bei oder vor Block 431017 entstanden sind, ist auf beiden Ketten gültig und kann von jedem auf der Altkette wiederholt werden. Das ist relevant, wenn Sie oder Ihre Nutzer Coins halten, die älter sind als der 11. September 2026:

- Vor der Nutzung solcher UTXOs für Auszahlungen einmal „markieren“: Geben Sie sie zusammen mit mindestens einem UTXO aus, das nur auf Doichain existiert (etwa ein Output einer nach Block 431017 bestätigten Transaktion oder ein Coinbase-Output nach dem Fork). Das entstehende Wechselgeld existiert nur noch auf Doichain.
- Nach dem 11. September 2026 empfangene Einzahlungen sind bereits Doichain-exklusiv, wenn sie auf Ihrem Node bestätigt wurden (der auf der Doichain-Kette steht).
- Schreiben Sie keine Einzahlungen gut, die ein Dienst meldet, der noch der Altkette folgt (Abschnitt 8.3).

7. Das Upgrade vom September 2026 (Fork) im Detail

Punkt	Wert
Aktivierung	Block 431017, geschürft am 11. September 2026, 14:33:16 UTC
Letzter gemeinsamer Block	431017, Hash 75a4ca09bf092862061e0e1c9f066145962f222ef965f3e9ccc27c6bcd0da320
Erster Doichain-exklusiver Block	431018, Hash 71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67
Erster Altketten-exklusiver Block	431018, Hash bab49c132328d09664261c3061608442408d4f667eaa457ed874b879923f2d34 (von Doichain Core 31 mit bad-diffbits abgelehnt)
Konsensänderungen	Difficulty-Algorithmus DigiShield v3, Prüfung der deklarierten Difficulty (nBits), strikte Namens-Eigentümerschaft für name_doi
Grund	Kettenstillstand nach dem Abzug eines großen Miners (Blöcke alle 5 bis 6 Stunden, keiner in den 9 Stunden vor dem Fork) und eine Konsensschwäche des Altclients, der die deklarierte Difficulty nie prüfte
Altsoftware	Doichain Core 0.20.x (und älter), wird noch von mindestens einem Pool geschürft, die Kette ist wegen der ungeprüften Difficulty in der Blockzahl weit vorausgeeilt
Replay-Schutz	keiner
Referenz	https://github.com/Doichain/doichain-core/issues/16

Alles, was nach Block 431017 auf der Altkette geschah, existiert auf Doichain nicht: keine Coinbase-Belohnungen, keine Überweisungen, keine Einzahlungen. Der Block-Explorer unter explorer.doichain.org zeigte am 17. September 2026 noch die Altkette und darf nicht als Referenz dienen, bis sein Betreiber ihn umgestellt hat (Abschnitt 8.3).

8. Datenquellen und Infrastruktur

8.1 Block-Explorer (aktuelle Kette)

<https://doi-explorer.le-space.de> (API im liquidus-Stil):

Endpunkt	Zweck
<code>/api/getblockcount</code>	aktuelle Höhe
<code>/api/getblockhash?index=<h></code>	Block-Hash nach Höhe
<code>/api/getblock?hash=<hash></code>	Block
<code>/api/getrawtransaction?txid=<txid>&decrypt=1</code>	Transaktion
<code>/api/getdifficulty, /api/getnetworkhashps</code>	Difficulty und Hashrate
<code>/ext/getmoneysupply</code>	Umlaufmenge
<code>/ext/getaddress/<adresse>, /ext/getbalance/<adresse></code>	Adressdaten
<code>/tx/<txid>, /address/<adresse>, /block/<hash></code>	Webseiten

Hinweis: Dieser Explorer zählt Bestätigungen als Blöcke oberhalb des enthaltenden Blocks und zeigt damit eine weniger als Bitcoin Core und Wallets.

8.2 Öffentliche ElectrumX-Server

`itchy-jellyfish-89.doi.works`, `big-parrot-60.doi.works`, `ugly-bird-70.doi.works`, `pink-deer-69.doi.works` (und `white-snail-54.doi.works`), alle TCP-Port 50002 mit TLS. Am 17. September 2026 liefen sie noch mit ElectrumX 1.15.0 (Protokoll 1.4 bis 1.4.2), das Referenz-Image des Projekts ist auf ElectrumX 2.0.0 umgestellt (Protokoll 1.4 bis 1.7.0, Abschnitt 3.5), die öffentlichen Server folgen. Der Genesis-Hash in `server.features` ist `000006fd...9134`. Nützlich für unabhängige Gegenprüfungen von Höhe und Salden. Nicht als primäre Quelle für die Gutschrift von Einzahlungen verwenden. Wer einen eigenen Indexer betreibt, sollte das 2.0.0-Image aus `doichain-install` statt eines 1.15-Builds verwenden.

Betriebshinweise für Clients: Die Server schließen inaktive Sitzungen nach rund zehn Minuten, auch bei aktivem Header-Abonnement, senden Sie deshalb alle ein bis zwei Minuten `server.ping` und verbinden Sie mit frischem Abonnement neu. `relayfee` unterscheidet sich zwischen den Servern um den Faktor 1.000, weil ihre Backend-Nodes unterschiedlich konfiguriert sind, leiten Sie Ihre Gebührenpolitik deshalb nie von einem einzelnen Server ab.

8.3 Nicht zu verwendende Quellen

- `explorer.doichain.org` und `cryptoscope.io/doi`: Altkette, Stand 17. September 2026 (Block 431018 = `bab49c13...`).
- `chainz.cryptoid.info/doi`: steht auf Altketten-Höhe 444.078 fest, bis der Betreiber neu indexiert.
- Docker-Tags `doichain/core:dc0.20.*` und `0.16.*`: Altsoftware.

8.4 Monitoring-Empfehlungen

- Alarm, wenn sich `getblockhash 431018` ändert oder wenn der Tip-Hash Ihres Nodes länger als zwei Blöcke vom Hash des Explorers bei gleicher Höhe abweicht.
- Alarm, wenn 90 Minuten kein neuer Block kommt (Ziel sind 10 Minuten, das Notventil macht lange Lücken unwahrscheinlich).

- `getnetworkhashps` und `getdifficulty` verfolgen, eine plötzliche mehrfache Änderung deutet auf einen großen Miner hin, der kommt oder geht, und kann eine vorübergehende Erhöhung der Bestätigungstiefe rechtfertigen.
- <https://github.com/Doichain/doichain-core/releases> und den Issue-Tracker beobachten. Konsensänderungen werden dort und gegenüber gelisteten Börsen per E-Mail angekündigt.

9. wDOI (ERC-20 auf Ethereum)

9.1 Token-Parameter

Parameter	Wert
Netzwerk	Ethereum-Mainnet, Chain-ID 1
Token-Vertrag (Proxy)	0xA2B6c1a7EFB3dFa75E2d9DF1180b02668c06da72
Proxy-Muster	aufrüstbarer Proxy nach ERC-1967, Quellcode auf Etherscan verifiziert
Implementierung (17.09.2026)	0x0b4DAF9420909A4163BCcf1C0131A65767Ef3161
Name / Symbol	Wrapped Doichain / wDOI
Dezimalstellen	18
Standard	ERC-20 (transfer, transferFrom, approve, Events Transfer und Approval)
Zusatzfunktionen	rollenbasiertes mint und burn (Custodian-Rolle), Notfall-Pause der Transfers, Tageslimit für Mint
Bereitgestellt	10. November 2025
Umlaufmenge, 17.09.2026	647.495,16 wDOI
Emittentin / Betreiberin	DOICHAINWORLD DOOEL Skopje, Nordmazedonien
Deckung	1 wDOI = 1 DOI in Reserve auf Doichain, Mint bei Einzahlung, Burn bei Rückgabe
Swap-/Bridge-Frontend	https://doilabs.com/swap

9.2 Integrationshinweise

- Standard-ERC-20-Integration: `Events Transfer(address,address,uint256)` auf Ihre Einzahlungsadressen beobachten, nach Ihrer üblichen Ethereum-Bestätigungstiefe gutschreiben, mit `transfer` auszahlen.
- wDOI ist **kein** Rebasing- und kein Fee-on-Transfer-Token. Salden ändern sich nur durch Transfers, Mints und Burns.
- Der Vertrag kann von der Betreiberin im Notfall **pausiert** werden. Während der Pause schlägt `transfer` fehl. Behandeln Sie eine fehlgeschlagene Auszahlung sauber und wiederholen Sie sie später. Die Betreiberin informiert gelistete Börsen vor und nach jeder Pause.
- Der Vertrag ist über den Proxy **aufrüstbar**. Die Betreiberin verpflichtet sich, Implementierungs-Upgrades gelisteten Börsen vorab anzukündigen. Die Token-Adresse ändert sich nie.
- Der Tausch zwischen DOI und wDOI erfolgt durch die Betreiberin über das Swap-Frontend (Mint) und durch Burn plus manuelle Freigabe (Rückgabe). Eine Börse muss die Bridge nicht selbst bedienen. Wenn Sie einen hauseigenen Tausch anbieten wollen, wenden Sie sich für eine API an die Betreiberin.

- Gegenprüfung der Menge: `totalSupply()` auf Ethereum darf die in der Bridge-Reserve gehaltenen DOI nie überschreiten. Die Betreiberin veröffentlicht die Reservezahlen auf der Swap-Seite und nennt die Reserveadressen auf Anfrage, ein öffentliches Dashboard ist in Vorbereitung.

9.3 Marktinformationen

Punkt	Wert
Bestehender DOI-Markt	XT.com, Spot-Paar DOI/USDT (seit 6. März 2024), Preisgenauigkeit 5 Dezimalstellen, Mengengenauigkeit 2
Marktdaten	CoinPaprika doi-doichain (https://api.coinpaprika.com/v1/tickers/doi-doichain)
wDOI-Märkte	noch keine an zentralen Börsen, On-Chain-Swap bei doilabs.com
Vorgeschlagene Paare	DOI/USDT, wDOI/USDT, optional DOI/BTC
Logo	weißes „D“ mit zwei senkrechten Balken auf blauer Scheibe (Vektordateien auf Anfrage)

10. Integrations-Checkliste

1. Doichain Core v31.1.6 aus einem signierten Release oder dem Docker-Image bereitstellen, frisches Datenverzeichnis.
2. `txindex=1`, Fixed Seeds, `minrelaytxfee=0.001`, `paytxfee=0.001` konfigurieren.
3. Nach der Synchronisation `getblockhash 431018 = 71d50ff1...` prüfen, Prüfung automatisieren.
4. Descriptor-Wallet anlegen, Bech32-Einzahlungsadressen erzeugen.
5. Einzahlungserkennung mit Filter für Namens-Outputs und 12 Bestätigungen (100 für Coinbase) umsetzen.
6. Auszahlungen mit fester Gebührenrate von 100 sat/vB umsetzen, nicht von `estimatesmartfee` abhängig machen.
7. UTXOs aus der Zeit vor dem Fork vor der ersten Nutzung markieren (Abschnitt 6.6).
8. Monitoring gegen doi-explorer.le-space.de und einen ElectrumX-Server aufsetzen.
9. wDOI als Standard-ERC-20 mit Pause-Behandlung integrieren.
10. Kontaktdaten mit dem Team austauschen und Release-Ankündigungen abonnieren.

Anhang A: Referenz-Hashes

Höhe	Hash	Hinweis
0	000006fdd8b4d786fd9bdde5bae9486c464e3aa4336c5f8415dfdd3fc1679134	Genesis
431016	4f5e8c0e4efb3504f8923ea175e4e5e688963819dcfbc33cc7f5a28c33616823	defaultAssumeValid, letzter Block vor dem Upgrade
431017	75a4ca09bf092862061e0e1c9f066145962f222ef965f3e9ccc27c6bcd0da320	Aktivierungsblock, letzter gemeinsamer Block
431018	71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67	erster Doichain-exklusiver Block

Anhang B: Versionsgeschichte von Doichain Core 31

Version	Datum	Inhalt
v31.1.2	11.09.2026	erstes Release des Upgrades, rund fünf Stunden nach der Aktivierung veröffentlicht
v31.1.3, v31.1.4	12. und 13.09.2026	Fehlerbehebungen, v31.1.4 hat ein Header-Sync-Problem bei frischen Nodes, nicht verwenden
v31.1.5	13.09.2026	Header-Sync-Fix, bumpfee-Schutz für Namens-Transaktionen, signierte statische Binaries
v31.1.6	17.09.2026	RPC help repariert, Bug-Report-URL aktualisiert, Konsens unverändert

Anhang C: Quellen für die Details zu Namensoperationen

Byte-Layout des Namens-Scripts, Index-Hash, Version 0x7100, Ablaufsemantik und die Änderung des Felds `address` in Core 31 stammen aus dem Names-on-Chain-Prüfbericht vom 16. September 2026 (Doichain-Core-Team), gegengeprüft an `src/script/names.h`, `src/names/common.h` und der `ElectrumX-Coin`-Klasse.

Anhang D: Glossar

AuxPoW Auxiliary Proof of Work, der Merged-Mining-Mechanismus, mit dem ein Bitcoin-Block Arbeit für Doichain nachweist. **DigiShield** Blockweiser Difficulty-Anpassungsalgorithmus, ursprünglich aus DigiByte, v3 wie bei Zcash. **name_doi** Doichains einstufiger Opcode zur Namensregistrierung (OP_10). **Altkette** Die Kette, der Doichain Core 0.20.x und älter nach Block 431017 folgen, nicht Doichain. **Markieren** Das Mischen eines UTXO aus der Zeit vor dem Fork mit einem UTXO nach dem Fork, sodass die entstehenden Outputs nur auf Doichain existieren.

Doichain Technische Integrationsspezifikation, Version 2026.1, 17. September 2026. Erstellt von der DOI Labs AG mit dem Doichain-Core-Entwicklungsteam. Korrekturen und Fragen an Doichain@le-space.de.