

Doichain (DOI) and Wrapped Doichain (wDOI)

Technical Integration Specification for Exchanges

Version 2026.1, 17 September 2026. Applies to Doichain Core v31.1.5 and later (current release v31.1.6).

Contacts: technical integration Nico Krause (Doichain@le-space.de), listing and commercial Ottmar Neuburger, DOI Labs AG (ottmar.neuburger@webanizer.de). Please open integration questions by e-mail; we answer within one business day.

1. Scope and summary

This document gives an exchange everything needed to run a Doichain node, credit deposits, process withdrawals and monitor the network for DOI, and to integrate the ERC-20 token wDOI on Ethereum. Doichain is a Bitcoin Core 31 derivative. Anyone who has integrated Bitcoin, Namecoin or another Bitcoin-family coin will find the RPC interface, the transaction format and the wallet model familiar. The Doichain-specific points are:

- 1. Merged mining (AuxPoW).** Block headers carry an auxiliary proof of work from Bitcoin and are therefore larger than 80 bytes. This only matters if you parse headers yourself.
- 2. Name operations.** Some outputs carry Namecoin-style name scripts. They are not payments and must be excluded from deposit detection.
- 3. The September 2026 upgrade.** The chain forked at block 431017. Legacy nodes follow a different chain. A start-up check is mandatory, and pre-fork coins have no replay protection.
- 4. No fee market.** Fee estimation returns no data. Use a fixed fee rate.

Quick facts:

Item	Value
Ticker	DOI
Decimals	8 (1 DOI = 100,000,000 satoshi-equivalents)
Reference client	Doichain Core v31.1.6, https://github.com/Doichain/doichain-core/releases
Docker image	doichain/core:v31.1.6 (also latest), linux/amd64
P2P port / RPC port	8338 / 8339
Block interval	600 s target
Recommended deposit confirmations	12 (see section 6.3)
Coinbase maturity	100 blocks
Address formats	N... (P2PKH), 6... (P2SH), dc1q... (Bech32 SegWit v0)
Chain check	getblockhash 431018 = 71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67
Explorer	https://doi-explorer.le-space.de
wDOI	ERC-20 on Ethereum mainnet, 0xA2B6c1a7EFB3dFa75E2d9DF1180b02668c06da72, 18 decimals

2. Network parameters (mainnet)

Source: `src/kernel/chainparams.cpp` and related files, tag v31.1.5. The `chainparams` file is byte-identical in v31.1.6.

2.1 Identification

Parameter	Value	Source
Network magic (pchMessageStart)	f8 b2 b2 ff	chainparams.cpp:191
Default P2P port	8338	chainparams.cpp:195
Default RPC port	8339	chainparamsbase.cpp:47
Genesis block hash	000006fdd8b4d786fd9bdde5bae9486c464e3aa4336c5f8415dfdd3fc1679134	chainparams.cpp:202
Genesis merkle root	234651063df5f8b01ecc2fc3a134fa1cb9dc9da9cce0149049483ba1b1469dfb	chainparams.cpp:203
Genesis time	1522756358 (3 April 2018, 11:52:38 UTC)	chainparams.cpp:200
DNS seeds	dnsseed.doichain.org, seed.doi.works (both may return no addresses, use fixed seeds)	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
Message signing prefix	Doichain Signed Message:\n	wallet constants

2.2 Addresses and keys

Parameter	Value	Source
Base58 P2PKH version	52 (0x34), addresses start with N	chainparams.cpp:213
Base58 P2SH version	13 (0x0d), addresses start with 6	chainparams.cpp:214
WIF private key version	180 (0xB4)	chainparams.cpp:215
BIP32 extended keys	0488B21E / 0488ADE4 (same as Bitcoin, xpub/xprv)	chainparams.cpp:216
Bech32 human-readable part	dc	chainparams.cpp:220
SegWit	active since block 216,500 (P2WPKH and P2WSH)	chainparams.cpp:128
Taproot	not activated (NEVER_ACTIVE), dc1p... addresses are invalid	chainparams.cpp:157
SLIP-0044 coin type	7070 (path m/44'/7070'/...)	SLIP-0044 registry

Validation rules for user-supplied withdrawal addresses:

- Base58Check with version byte 0x34 or 0x0d, 25 bytes decoded, valid checksum. Regular expression for a first-pass filter: `^[N6][1-9A-HJ-NP-Za-km-z]{33}$`.

- Bech32 (BIP173, not Bech32m) with HRP dc, witness version 0, program length 20 (P2WPKH) or 32 (P2WSH). Lower-case only. Regular expression: `^dc1q[02-9ac-hj-np-z]{38}$` or `^dc1q[02-9ac-hj-np-z]{58}$`.
- Reject witness version 1 and above.

`validateaddress` on the node performs the full check and returns `isvalid`.

2.3 Consensus and proof of work

Parameter	Value	Source
Hash function	SHA-256d	Bitcoin family
Mining	AuxPoW merged mining, chain ID 0x0002, from block 1, <code>fStrictChainId = false</code>	chainparams.cpp:179
Block version in use	0x00020104 (chain ID 2, AuxPoW flag, version 4)	header probe, block 431018
Header size	variable, typically 0.6 to 2.3 kB (80-byte header plus AuxPoW)	release notes v31.1.5
Target spacing	600 s	chainparams.cpp:146
Difficulty algorithm	DigiShield v3 from block 431017: 17-block averaging window, max +16 % / -32 % per block	chainparams.cpp:135, params.h:221
Difficulty reset	first 28 blocks after 431017 at <code>nBits 0x1a0400cd</code>	chainparams.cpp:139
Safety valve	after 3,600 s without a block the next block may be at most 4 times easier	chainparams.cpp:140
Legacy retarget	every 2,016 blocks over 14 days, valid up to block 431016	chainparams.cpp:145
<code>powLimit</code>	0000ffff...	chainparams.cpp:144
Max block weight	4,000,000	consensus.h:13
Max standard tx weight	400,000	policy.h:37
Coinbase maturity	100 blocks	consensus.h:19
Checkpoints	none	chainparams.cpp
<code>defaultAssumeValid</code>	4f5e8c0e4efb3504f8923ea175e4e5e688963819dcfbc33cc7f5a28c33616823 (block 431016)	chainparams.cpp:177

2.4 Monetary rules

Parameter	Value	Source
COIN	100,000,000	amount.h:15
MAX_MONEY	21,000,000 DOI	amount.h:26
Initial subsidy	50 DOI	validation.cpp:1945
Halving interval	210,000 blocks	chainparams.cpp:118
Current subsidy	12.5 DOI (blocks 420,000 to 629,999)	derived
Supply at block 432,057 (17 Sep 2026)	15,901,912 DOI (explorer), 15,900,725 DOI theoretical	doi-explorer.le-space.de/ext/getmoneysupply

2.5 Fee policy defaults

Parameter	Value	Source
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
Public ElectrumX servers report relayfee	0.00001 DOI/kB (their backend nodes run a lower minrelaytxfee)	live probe 17 Sep 2026
estimatesmartfee / ElectrumX estimatefee	returns no estimate (-1) because there is no fee market	live probe 17 Sep 2026

Practical consequence: a transaction paying 0.001 DOI/kvB (100 sat/vB) is accepted by every node with default settings and is included in the next block. The mining nodes enforce this floor even though the public ElectrumX servers relay cheaper transactions; such transactions sit in the mempool and are never mined (Electrum-DOI dc4.1.7 release notes, 18 September 2026). Typical transactions cost between 0.0002 and 0.0005 DOI. A 226-byte payment observed on 17 September 2026 paid 0.000146 DOI and confirmed in the next block.

2.6 Name operations (must be recognised)

Doichain inherits Namecoin's name scripts and adds one of its own.

Opcodes	Meaning	Script pattern
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>

Name transactions carry the transaction version `0x7100` (28928), inherited from Namecoin, which is a quick way to recognise them. A name output carries at least 0.01 DOI (`MinNameCoinAmount`, `params.h:60`). This amount is not burned and not a fee: it stays locked in the name UTXO, travels with the name on every update or transfer and is released only when the name expires (36,000 blocks after the last operation). The scriptPubKey starts with the opcode and ends in an ordinary payment script.

Byte layout of a live mainnet registration of the name `test` with value `hello` to a P2PKH owner (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 bytes> 88 ac P2PKH of the owner
```

ElectrumX indexes names under the script hash `sha256(0x53 || push(name) || 0x00 || 6d 75 6a)` (bytes reversed), queried with `blockchain.scripthash.get_history`; the index also returns expired and superseded entries, expiry has to be computed by the client against the chain tip. `name_doi` registers in a single step without Namecoin's commit-reveal (`name_new`, `name_firstupdate`), so a pending registration in the mempool can be front-run; this is irrelevant for exchange operation but explains why name outputs may appear and disappear in the mempool. `decoderawtransaction` reports such outputs with "type": "nulldata"-like custom types in older tooling; in Doichain Core they appear with a `nameOp` field. **Deposit**

detection must ignore any output whose script starts with OP_1, OP_2, OP_3 or OP_10, otherwise the 0.01 DOI name deposits will be credited as payments to whichever address the trailing script contains. Exchange hot wallets never need to create name outputs.

2.7 Soft-fork status

Feature	Activation
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 to 342)	never
BIP34	not enforced (height 100,000,000)

3. Node deployment

3.1 Software

Use Doichain Core **v31.1.6** or later from <https://github.com/Doichain/doichain-core/releases>. Static Linux x86_64 binaries (musl) and a Docker image are provided. Release tags are GPG-signed with key 52A2 1926 E3F2 74AA 1F2D 9CDA 9BA0 ED44 F14E 8118. Verify SHA256SUMS.asc before deployment.

Do **not** use any 0.20.x or 0.16.x release, and do not reuse a data directory that was ever synced with such a release. Those nodes follow the legacy chain (section 7). A fresh sync from genesis takes a few minutes on current hardware because the chain is small (about 2.7 million transactions).

If you prefer a ready-made deployment, the project maintains a reference Docker Compose stack at <https://github.com/Doichain/doichain-install> (section 3.5). Its `docker-compose-mining.yml` starts Doichain Core, an ElectrumX indexer, a merged-mining p2pool and a pruned Bitcoin node; for an exchange only the Core and, optionally, the ElectrumX service are needed.

3.2 Hardware

Resource	Recommendation
CPU	2 cores
RAM	4 GB (2 GB minimum)
Disk	20 GB SSD (chain about 5 GB including indexes, growing slowly)
Network	stable outbound connectivity to port 8338 peers

3.3 Configuration

Example `doichain.conf`:

```
server=1
daemon=1
txindex=1
rpcuser=<user>
rpcpassword=<strong password>
```

```

rpcport=8339
rpcbind=127.0.0.1
rpccallowip=127.0.0.1
listen=1
port=8338
maxconnections=40
# fixed seeds in case DNS seeds return nothing
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
# fee policy (see section 6.4)
minrelaytxfee=0.001
paytxfee=0.001
# wallet
wallet=exchange_hot

```

Docker equivalent:

```

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

```

The official image writes `doichain.conf` on the **first** start from the environment variables `RPC_PASSWORD` (mandatory) and `RPC_ALLOW_IP`, generates `bind=` for the active network and advertises the address given in `EXTERNAL_IP`. It publishes only the P2P port; the RPC port stays inside the Docker network by design. Additional daemon arguments can be passed through `DOICHAIN_EXTRA_ARGS`. Docker Hub tags `v31.1.5` and `v31.1.6` are built from the signed release tags.

Doichain Core 31 uses descriptor wallets (SQLite). Legacy Berkeley DB wallets from older releases must be migrated once with `migratewallet`.

3.4 Mandatory start-up check

After the node reports `initialblockdownload: false` in `getblockchaininfo`, run:

```
doichain-cli getblockhash 431018
```

Expected: `71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67`.

If the node returns `bab49c132328d09664261c3061608442408d4f667eaa457ed874b879923f2d34` it is on the legacy chain. Stop it, delete the data directory and resync with the current release. Automate this check in your monitoring and alert if the result ever changes.

Additional sanity checks: `getblockchaininfo` shows `chain: main`, the height is within a few blocks of `https://doi-explorer.le-space.de/api/getblockcount`, and `getnetworkinfo` shows `subversion /Satoshi:31.1.x/` (the user-agent string has not yet been renamed to Doichain in the 31.1 line; a legacy node reports `/Satoshi:0.20.x/`).

3.5 Reference stack (doichain-install)

<https://github.com/Doichain/doichain-install> is the project's reference deployment, updated for Core 31.1 in September 2026 (pull requests 5 to 9, documentation in `docs/stack-31.1.md`). Relevant facts for an exchange:

Item	Value
Compose file	<code>docker-compose-mining.yml</code> (services <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> pinned in the compose file; <code>v31.1.6</code> on Docker Hub is a drop-in replacement
ElectrumX image	<code>doichain/electrumx:v2.0.0-doi1</code> , built from <code>Doichain/electrumx</code> tag <code>v2.0.0-doi1</code> (commit <code>11877d7b</code>), upstream ElectrumX 2.0.0 with the Doichain coin classes (AuxPoW, SegWit, name index including <code>name_doi</code>)
ElectrumX protocol	1.4 to 1.7.0 served; name RPC <code>blockchain.name.get_value_proof</code> from protocol 1.4.3
ElectrumX database	RocksDB (<code>DB_ENGINE=rocksdb</code> , mandatory since 2.0); an index built with the 1.15 image cannot be migrated, the <code>electrumx-db</code> volume must start empty
Index time from empty volumes	about 22 minutes for the whole chain (measured 16 September 2026)
ElectrumX ports	50001 TCP, 50002 SSL, 50004 WSS; self-signed certificate on first start, mount your own for public use
Daemon connection	<code>DAEMON_URL=http://<user>:<password>@doichain:8339/</code> (explicit port required by the endpoint)
Requirements	Docker Engine 20.10+, Docker Compose v2.17+ (the ElectrumX service uses <code>additional_contexts</code>)

Known open items listed by the maintainers: the generated `doichain.conf` still contains a permissive `rpcallowip` that should be tightened, the `p2pool` and `bitcoind` images on Docker Hub are older builds that the compose file patches at run time, and the node's user-agent string is still `Satoshi`. None of these affects consensus or the RPC interface.

4. RPC interface

The RPC interface is that of Bitcoin Core 31 (JSON-RPC over HTTP, port 8339). All standard calls behave as 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` (returns no estimate, see 6.4), `getnetworkinfo`, `getpeerinfo`.

Doichain-specific additions (not needed for exchange operation): `name_show`, `name_history`, `name_list`, `name_new`, `name_firstupdate`, `name_update`, `name_doi`, `name_scan`, `name_pending`. `bumpfee` and `psbtbumpfee` refuse to touch transactions containing name outputs.

Differences worth knowing:

- `getblock` with verbosity 2 returns the AuxPoW data in an `auxpow` object of the header. The hash of a block is the hash of its 80-byte core header, as usual.
- `getblockheader` in hex form returns the full serialized header including AuxPoW, so its length varies.
- `help` was broken in v31.1.2 to v31.1.5 (RPCHelpMan bug) and is fixed in v31.1.6. The RPC calls themselves worked in all versions.
- Amounts are in DOI with 8 decimals, exactly like BTC in Bitcoin Core.

- As in Bitcoin Core since version 22, `scriptPubKey` in decoded transactions carries a single address string; the legacy `addresses[]` array of Doichain Core 0.20.x is gone. Tooling written against the old client must be adapted (`spk.address ?? spk.addresses?.[0]`).
- Name outputs appear in decoded transactions with an additional `nameOp` object (`op, name, value`); the trailing payment script is reported as usual, so filter on `nameOp` or on the leading opcode, not on the address.

5. Block and transaction format

Item	Value
Block header	80-byte Bitcoin header followed by the AuxPoW structure (parent coinbase transaction, coinbase merkle branch, chain merkle branch, parent block header). Present in every block since height 1.
Block hash	SHA-256d of the 80-byte header
Transactions	Bitcoin format including SegWit (BIP141). <code>txid</code> and <code>wtxid</code> as in Bitcoin.
Script	Bitcoin script plus name opcodes (section 2.6). No Taproot.
Locktime, sequence, RBF	as in Bitcoin Core 31: the mempool applies full replace-by-fee , any unconfirmed transaction can be replaced by a higher-fee conflict regardless of its sequence numbers. Never credit deposits at zero confirmations.
Serialization for hashing	AuxPoW is not part of the hashed header, so standard Bitcoin header hashing libraries work if you strip the AuxPoW data.

If you use a third-party indexer (for example an ElectrumX or a block parser), it must understand AuxPoW headers. The public ElectrumX servers (section 8.2) already do.

6. Deposits and withdrawals

6.1 Address generation

Generate deposit addresses with `getnewaddress "" bech32` (preferred, lowest fees) or `getnewaddress "" legacy`. Do not use `bech32m`. Both formats are supported by all current Doichain wallets. Public ElectrumX servers and the explorer index both.

6.2 Detecting deposits

Poll `listsinceblock <lastblockhash> 1 true` on each new block, or subscribe with `-blocknotify`. For each received transaction:

1. Take `gettransaction <txid>` or `getrawtransaction <txid> true`.
2. Credit only outputs whose `scriptPubKey.type` is `pubkeyhash`, `scripthash`, `witness_v0_keyhash` or `witness_v0_scripthash` and whose address belongs to you.
3. Discard outputs whose script begins with `OP_1`, `OP_2`, `OP_3` or `OP_10` (name operations), regardless of the trailing address.
4. Treat coinbase outputs as spendable only after 100 confirmations (`COINBASE_MATURITY`).

Minimum deposit recommendation: 1 DOI (well above dust and name amounts).

6.3 Confirmation depth

XT.com credits DOI deposits after 10 confirmations. We recommend **12 confirmations** for the following reasons: the network hash rate is small (about 3.4 PH/s in September 2026), a single merged-mining pool currently produces most blocks, and the difficulty algorithm has been live since 11 September 2026. Twelve blocks take about two hours at the target interval. For very large deposits consider 24 confirmations. Reduce the depth once the hash rate has broadened, and never credit below 6 confirmations.

6.4 Fees for withdrawals

There is no functioning fee estimation. `estimatesmartfee` returns no `feerate`, and ElectrumX servers answer `estimatefee` with `-1`. Use a **fixed fee rate**:

- Set `paytxfee=0.001` (0.001 DOI per kB, 100 sat/vB) in `doichain.conf`, or pass `fee_rate=100` (sat/vB) to `send/sendtoaddress`.
- This equals the default `minrelaytxfee` of an unmodified node and guarantees relay and next-block inclusion.
- Never rely on `estimatesmartfee` succeeding. If your framework requires an estimator, hard-code the fallback.
- A typical 1-input, 2-output SegWit withdrawal is about 140 vB and costs 0.00014 DOI at this rate.

For the user-facing withdrawal fee, XT.com charges 58.7 DOI with a minimum withdrawal of 587.2 DOI. Given the on-chain cost this is generous; a fee of 1 to 5 DOI and a minimum of 10 DOI would be more than sufficient.

6.5 Withdrawal processing

Standard Bitcoin Core workflow: `send` or `sendtoaddress` from the hot wallet, or PSBT-based signing with an offline signer (`walletcreatefundedpsbt`, `walletprocesspsbt`, `finalizepsbt`, `sendrawtransaction`). Batch withdrawals with `sendmany` where possible. Wait for at least 1 confirmation before showing a withdrawal as complete, and keep the standard reorg handling (re-check after 6 blocks).

6.6 Replay caution for pre-fork coins

Blocks up to 431017 are shared with the legacy chain, and there is no replay protection. A transaction that spends **only** UTXOs created at or before block 431017 is valid on both chains and can be replayed on the legacy chain by anyone. This matters if you or your users hold coins that predate 11 September 2026:

- Before using pre-fork UTXOs for withdrawals, "taint" them once: spend them together with at least one UTXO that only exists on Doichain (for example an output of any transaction confirmed after block 431017, or a post-fork coinbase output). The resulting change is Doichain-only.
- Deposits received after 11 September 2026 are already Doichain-only if they were confirmed on your node (which is on the Doichain chain).
- Do not credit deposits reported by any service that still follows the legacy chain (section 8.3).

7. The September 2026 upgrade (fork) in detail

Item	Value
Activation	block 431017, mined 11 September 2026, 14:33:16 UTC

Item	Value
Last common block	431017, hash 75a4ca09bf092862061e0e1c9f066145962f222ef965f3e9ccc27c6bcd0da320
First Doichain-only block	431018, hash 71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67
First legacy-only block	431018, hash bab49c132328d09664261c3061608442408d4f667eaa457ed874b879923f2d34 (rejected by Doichain Core 31 with bad-diffbits)
Consensus changes	DigiShield v3 difficulty algorithm, enforcement of declared difficulty (nBits), strict name ownership for name_doi
Reason	chain stall after a large miner left (blocks every 5 to 6 hours, none in the 9 hours before the fork) and a legacy consensus flaw that never verified declared difficulty
Legacy software	Doichain Core 0.20.x (and older), still mined by at least one pool; its chain has moved far ahead in block count because of the unverified difficulty
Replay protection	none
Reference	https://github.com/Doichain/doichain-core/issues/16

Anything that happened on the legacy chain after block 431017 does not exist on Doichain: no coinbase rewards, no transfers, no deposits. The block explorer at explorer.doichain.org still showed the legacy chain on 17 September 2026 and must not be used as a source of truth until its operator has migrated it (section 8.3).

8. Data sources and infrastructure

8.1 Block explorer (current chain)

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

Endpoint	Purpose
/api/getblockcount	current height
/api/getblockhash?index=<h>	block hash by height
/api/getblock?hash=<hash>	block
/api/getrawtransaction?txid=<txid>&decrypt=1	transaction
/api/getdifficulty, /api/getnetworkhashps	difficulty and hash rate
/ext/getmoneysupply	circulating supply
/ext/getaddress/<address>, /ext/getbalance/<address>	address data
/tx/<txid>, /address/<address>, /block/<hash>	web pages

Note: this explorer counts confirmations as blocks on top of the including block, so it shows one fewer than Bitcoin Core and wallets do.

8.2 Public ElectrumX servers

itchy-jellyfish-89.doi.works, big-parrot-60.doi.works, ugly-bird-70.doi.works, pink-deer-69.doi.works (and white-snail-54.doi.works), all TCP port 50002 with TLS. On 17 September

2026 they still ran ElectrumX 1.15.0 (protocol 1.4 to 1.4.2); the project's reference image has moved to ElectrumX 2.0.0 (protocol 1.4 to 1.7.0, section 3.5) and the public servers will follow. Genesis hash in `server.features` is `000006fd...9134`. Useful for independent cross-checks of height and balances. Do not use them as the primary source for crediting deposits. If you run your own indexer, use the 2.0.0 image from `doichain-install` rather than a 1.15 build. Operational notes for clients: the servers close idle sessions after about ten minutes even with an active header subscription, so send `server.ping` every one to two minutes and reconnect with a fresh subscription; `relayfee` differs between the servers by a factor of 1,000 because their backend nodes are configured differently, so never derive your fee policy from a single server.

8.3 Sources not to use

- `explorer.doichain.org` and `cryptoscope.io/doi`: legacy chain as of 17 September 2026 (block 431018 = `bab49c13...`).
- `chainz.cryptoid.info/doi`: stuck at legacy height 444,078 pending a reindex by its operator.
- Docker tags `doichain/core:dc0.20.*` and `0.16.*`: legacy software.

8.4 Monitoring recommendations

- Alert if `getblockhash 431018` changes or if your node's tip hash differs from the explorer's hash at the same height for more than two blocks.
- Alert if no new block arrives for 90 minutes (target is 10 minutes; the safety valve makes long gaps unlikely).
- Track `getnetworkhashps` and `getdifficulty`; a sudden multiple-fold change indicates a large miner joining or leaving and may justify a temporary increase of the deposit confirmation depth.
- Watch <https://github.com/Doichain/doichain-core/releases> and Issue tracker for announcements. Consensus changes are announced there and to listed exchanges by e-mail.

9. wDOI (ERC-20 on Ethereum)

9.1 Token parameters

Parameter	Value
Network	Ethereum mainnet, chain ID 1
Token contract (proxy)	<code>0xA2B6c1a7EFB3dFa75E2d9DF1180b02668c06da72</code>
Proxy pattern	ERC-1967 upgradeable proxy, source verified on Etherscan
Implementation (17 Sep 2026)	<code>0x0b4DAf9420909A4163BCcf1C0131A65767Ef3161</code>
Name / symbol	Wrapped Doichain / wDOI
Decimals	18
Standard	ERC-20 (<code>transfer</code> , <code>transferFrom</code> , <code>approve</code> , <code>Transfer</code> and <code>Approval</code> events)
Additional features	role-based mint and burn (custodian role), emergency pause of transfers, daily mint limit
Deployed	10 November 2025
Supply in circulation, 17 Sep 2026	647,495.16 wDOI
Issuer / operator	DOICHAINWORLD DOOEL Skopje, North Macedonia

Parameter	Value
Backing	1 wDOI = 1 DOI held in reserve on Doichain; mint on deposit, burn on redemption
Swap / bridge front end	https://doilabs.com/swap

9.2 Integration notes

- Standard ERC-20 integration: watch `Transfer(address,address,uint256)` events to your deposit addresses, credit after your usual Ethereum confirmation depth, withdraw with `transfer`.
- wDOI is **not** a rebasing or fee-on-transfer token. Balances change only through transfers, mints and burns.
- The contract can be **paused** by the operator in an emergency. While paused, `transfer` reverts. Handle a reverted withdrawal gracefully and retry later. The operator will notify listed exchanges before and after any pause.
- The contract is **upgradeable** via the proxy. The operator commits to announcing implementation upgrades to listed exchanges in advance. The token address never changes.
- Conversion between DOI and wDOI is performed by the operator through the swap front end (mint) and by burn plus manual release (redeem). An exchange does not need to bridge itself; if you want to offer in-house conversion, contact the operator for an API.
- Supply cross-check: `totalSupply()` on Ethereum should never exceed the DOI held in the bridge reserve. The operator publishes reserve figures on the swap site and provides reserve addresses on request; a public dashboard is in preparation.

9.3 Market information

Item	Value
Existing DOI market	XT.com, spot pair DOI/USDT (since 6 March 2024), price precision 5 decimals, quantity precision 2
Market data	CoinPaprika doi-doichain (https://api.coinpaprika.com/v1/tickers/doi-doichain)
wDOI markets	none on centralised exchanges yet; on-chain swap at doilabs.com
Suggested pairs	DOI/USDT, wDOI/USDT, optionally DOI/BTC
Logo	white "D" with two vertical bars on a blue disc (vector files on request)

10. Integration checklist

1. Deploy Doichain Core v31.1.6 from a signed release or the Docker image, fresh data directory.
2. Configure `txindex=1`, fixed seeds, `minrelaytxfee=0.001`, `paytxfee=0.001`.
3. After sync, verify `getblockhash 431018 = 71d50ff1...`; automate the check.
4. Create a descriptor wallet, generate Bech32 deposit addresses.
5. Implement deposit scanning with name-output filtering and 12 confirmations (100 for coinbase).
6. Implement withdrawals with a fixed fee rate of 100 sat/vB; do not depend on `estimateSmartFee`.
7. Taint any pre-fork UTXOs before first use (section 6.6).
8. Set up monitoring against doi-explorer.le-space.de and an ElectrumX server.

9. Integrate wDOI as a standard ERC-20 with pause handling.
10. Exchange contact details with the team and subscribe to release announcements.

Appendix A: Reference hashes

Height	Hash	Note
0	000006fdd8b4d786fd9bde5bae9486c464e3aa4336c5f8415dfdd3fc1679134	genesis
431016	4f5e8c0e4efb3504f8923ea175e4e5e688963819dcfbc33cc7f5a28c33616823	defaultAssumeValid, last pre-upgrade block
431017	75a4ca09bf092862061e0e1c9f066145962f222ef965f3e9ccc27c6bcd0da320	activation block, last common block
431018	71d50ff12b090561cc918ddb560334b4350758c7eace3f058dd332fb112f4b67	first Doichain-only block

Appendix B: Version history of Doichain Core 31

Version	Date	Content
v31.1.2	11 Sep 2026	first release of the upgrade, published about five hours after activation
v31.1.3, v31.1.4	12 and 13 Sep 2026	bug fixes; v31.1.4 has a header-sync issue for fresh nodes, do not use
v31.1.5	13 Sep 2026	header-sync fix, bumpfee protection for name transactions, signed static binaries
v31.1.6	17 Sep 2026	help RPC repaired, bug-report URL updated; consensus unchanged

Appendix C: Sources for the name-operation details

Name-script byte layout, index hash, version 0x7100, expiry semantics and the Core 31 address field change are taken from the Names-on-Chain audit report of 16 September 2026 (Doichain core team), cross-checked against `src/script/names.h`, `src/names/common.h` and the ElectrumX coin class.

Appendix D: Glossary

AuxPoW Auxiliary proof of work, the merged-mining mechanism by which a Bitcoin block proves work for Doichain. **DigiShield** Per-block difficulty adjustment algorithm originally from DigiByte, v3 as used by Zcash. **name_doi** Doichain's single-step name registration opcode (OP_10). **Legacy chain** The chain followed by Doichain Core 0.20.x and older after block 431017; not Doichain. **Taint** Mixing a pre-fork UTXO with a post-fork UTXO so that the resulting outputs exist on Doichain only.

Doichain Technical Integration Specification, version 2026.1, 17 September 2026. Prepared by DOI Labs AG with the Doichain core development team. Corrections and questions to Doichain@le-space.de.