NOCIP

No Cloud Inference on Personal data

Handbook: automated via MCP server

For vendors, their AI agents and build systems: the NOCIP label through the MCP server or the REST interface

Version 1.0 · 2026-11-11 · released







1 Overview

On the way “Automated via MCP server”, your AI agent takes the product through the NOCIP audit with the MCP server, a build system with the REST interface. The second way is “With documents”. The same criteria, audit checklist and audit report apply as in the audit with documents. NOCIP measures for itself what can be measured, writes every step to a signed case log, and a person signs the award.

Why automated via MCP server

- No forms: self-assessment and evidence arrive as data; the interface reports missing items at once.
- No appointment for the test: the lab tests your release as soon as your agent requests it; the result is there within minutes.
- The state at any time: get_status says what has been received and checked and what to do next.
- New versions without paperwork: your build system announces them, a green re-check extends the label.
- Traceable: every step is in the signed case log, which you can verify yourself.

2 Two ways to the label

	With documents	Automated via MCP server
Self-assessment	fill in a DOCX, send it by e-mail	JSON through the interface, checked at once
Evidence	as attachments by e-mail	an address is enough, NOCIP fetches it; files up to 25 MB
Technical test	agree test access, appointment	on request in the lab, result within minutes
State of the audit	on request by e-mail	at any time, with the next steps

	With documents	Automated via MCP server
Record of the audit steps	audit report	audit report and signed case log
New versions	report by e-mail, re-check by NOCIP	announced by the build system, a green re-check extends the label
Platforms	all	Linux programs for now; Windows, macOS and cloud services follow
Criteria, audit report, cost	the same, free of charge	the same, free of charge

3 Who is active when

Your agent

Your AI agent (through the MCP server) or your build system (through the REST interface), with the token of your case. It sends statements, evidence and answers. Everything it sends is a claim, not a finding.

NOCIP systems

The interface on nocip.com accepts submissions and checks them for completeness. The audit server takes them every minute, checks domain and self-assessment, fetches evidence itself, downloads the release itself, tests it in the isolated lab and writes every step, signed, to the case log. Program code decides measurable criteria by the thresholds of the catalogue.

Person at NOCIP

An auditor of be next GmbH. Opens the case after the application, reads terms and documentation, decides on disputes and signs award, refusal and suspension. No interface and no agent can trigger these steps.

All steps at a glance

Step		Agent	Systems	Person	Decided by	Duration
S0	Access	x	x	x	a person at NOCIP	once; reply by e-mail
S1	Prove the domain	x	x		NOCIP systems (program code)	1 to 2 minutes
S2	Self-assessment	x	x		NOCIP systems (program code)	at once and after about 2 minutes
S3	Evidence and connection manifest	x	x		NOCIP systems (program code)	1 to 2 minutes
S4	Test target	x	x		NOCIP systems (program code)	depends on the file size, usually under 5 minutes
S5	Technical test in the lab	x	x		NOCIP systems (program code)	a few minutes (LexITanon: under one minute)
S6	Review of the documents		x	x	a person at NOCIP	working days, depending on the scope
S7	Findings	x	x	(x)	NOCIP systems (program code)	minutes; with a person, working days

Step		Agent	Systems	Person	Decided by	Duration
S8	Draft report	x	x	x	NOCIP systems (program code)	1 to 2 minutes
S9	Award		x	x	a person at NOCIP	working days
S10	Seal and disclosure	x	x	(x)	NOCIP systems (program code)	ongoing
S11	New versions	x	x	(x)	NOCIP systems (program code)	a few minutes

(x): only when needed

4 Prerequisites

- Case number and token from NOCIP (application through the contact form, topic “Application for an audit”).
- Access to DNS or the website of the domain(s) the label is to cover.
- A Linux program (x86-64) that starts without its interface and processes a document through a local HTTP interface or the command line, with a download address and a SHA-256 checksum.
- The connection manifest at /.well-known/nocip.json on your domain.

5 Set-up

MCP server (for AI agents). The token names the case; the agent starts with get_status, whose “next” list names the next steps. Example Claude Code:

```
claude mcp add --transport http nocip https://nocip.com/mcp/ \
  --header "Authorization: Bearer nct_..."
```

Other MCP clients: address, transport “Streamable HTTP” and the header Authorization:

```
{
  "mcpServers": {
    "nocip": {
      "type": "http",
      "url": "https://nocip.com/mcp/",
      "headers": {
        "Authorization": "Bearer nct_..."
      }
    }
  }
}
```

REST interface (for build systems and scripts). Answers are JSON; submissions are accepted with 202 and processed within about a minute.

```
curl -H "Authorization: Bearer nct_..." https://nocip.com/api/v1/cases/NOCIP-A-2026-0001
curl -H "Authorization: Bearer nct_..." -H "Content-Type: application/json" \
     --data @self-assessment.json https://nocip.com/api/v1/cases/NOCIP-A-2026-0001/self-assessment
```

OpenAPI: https://nocip.com/assets/api/nocip-api-v1.yaml

6 The process step by step

S0 Access

Your agent	A person of your company applies through the contact form (topic “Application for an audit”): product, tier, domain(s), contact.
NOCIP systems	Creates the case, pins the catalogue version (with SHA-256), creates the domain code and keeps only the hash of the token. The first case log entries are signed.
Person at NOCIP	Checks the company (commercial register) and the contact once, opens the case and sends case number and token by e-mail.
Decided by	a person at NOCIP
Duration	once; reply by e-mail
Log entries	case.opened, criteria.pinned

S1 Prove the domain

Your agent	Sets the DNS TXT record _nocip.<domain> with the code, or puts it as domain_proof into https://<domain>/.well-known/nocip.json, then prove_domain.
NOCIP systems	Queries DNS and the file and records the result. It then fetches the connection manifest of the domain.
Person at NOCIP	not involved
Decided by	NOCIP systems (program code)
Duration	1 to 2 minutes
Tool	prove_domain
Log entries	domain.proof_requested, domain.proved, domain.proof_failed

S2 Self-assessment

Your agent	Sends the self-assessment as JSON: one answer per criterion of the tier, the AI functions with their location, the index points claimed, evidence as an address or file hash.
NOCIP systems	At once: missing criteria, wrong tier or catalogue version (answer 422). Within about two minutes: schema, contradictions (e.g. Gold with an AI function outside the customer), answers “no” or “partly” as findings; every address named is fetched.
Person at NOCIP	not involved
Decided by	NOCIP systems (program code)
Duration	at once and after about 2 minutes
Tool	submit_self_assessment
Log entries	self_assessment.received, self_assessment.validated, finding.raised

S3 Evidence and connection manifest

Your agent	Names further evidence (address or file up to 25 MB) and publishes the connection manifest at /.well-known/nocip.json: every destination with purpose and frequency (I-5).

NOCIP systems	Fetches every address itself and keeps a snapshot with SHA-256. Checks the manifest against the schema and against the tier.
Person at NOCIP	not involved
Decided by	NOCIP systems (program code)
Duration	1 to 2 minutes
Tool	add_evidence
Log entries	evidence.received, evidence.snapshot, manifest.checked

S4 Test target

Your agent	Names the release: public download address (or one behind a login, through an uploaded header line), SHA-256, size and how the program starts without its interface and processes a document.
NOCIP systems	Downloads the file itself and compares checksum, size and, if given, the published checksum. A file sent by the agent is never tested.
Person at NOCIP	not involved
Decided by	NOCIP systems (program code)
Duration	depends on the file size, usually under 5 minutes
Tool	set_test_target
Log entries	test_target.received, test_target.verified, test_target.failed

S5 Technical test in the lab

Your agent	Requests the test (at most 5 a day).
NOCIP systems	Starts the program in an isolated network. Every connection attempt is recorded, every name lookup answered and noted. A test corpus with canary values generated for the case is processed; then all files and output of the program are searched for the canary values, and the destinations compared with the manifest. Result for G-01, G-02, C-04 and C-06.
Person at NOCIP	not involved
Decided by	NOCIP systems (program code)
Duration	a few minutes (LexITanon: under one minute)
Tool	request_test
Log entries	test.requested, test.started, test.finished, result.set

S6 Review of the documents

Your agent	Nothing; questions come as findings.
NOCIP systems	Provides the stored snapshots with their checksums. Planned: an AI model that runs at NOCIP itself prepares the review with verbatim quotes; program code checks that every quote is in the snapshot.
Person at NOCIP	Reads terms, privacy policy and documentation for C-01, C-02, C-03, C-05, C-07, C-08 and the index groups, enters result and reasons (German and English) and raises findings.

Decided by	a person at NOCIP
Duration	working days, depending on the scope
Log entries	result.set, index.set, finding.raised

S7 Findings

Your agent	Answers every open finding: “fixed” (with a new version or document), “evidence” or “dispute”.
NOCIP systems	Checks newly sent statements and tests new versions; what no longer applies is closed automatically.
Person at NOCIP (only when needed)	Becomes active on a dispute, after the third answer to the same finding and for findings from the review of the documents.
Decided by	NOCIP systems (program code)
Duration	minutes; with a person, working days
Tool	answer_finding
Log entries	finding.answered, finding.closed

S8 Draft report

Your agent	Can read the draft (German and English).
NOCIP systems	Builds the draft audit report from the case log: tests, result per criterion, index, open points. The draft is marked as such.
Person at NOCIP	Triggers the draft once the review of the documents is finished.
Decided by	NOCIP systems (program code)
Duration	1 to 2 minutes
Tool	get_report_draft
Log entries	report.drafted

S9 Award

Your agent	Nothing.
NOCIP systems	Allows the award only if every criterion of the tier is passed, every index group is scored, no blocking point is open and the last test of the current release is green. Then builds audit report, certificate, seal and register entry.
Person at NOCIP	Reads the result and the differences between statement and measurement and signs at the audit server with a typed confirmation: award, or refusal with reasons. The register entry appears with the next update of nocip.com.
Decided by	a person at NOCIP
Duration	working days
Log entries	case.awarded, case.refused

S10 Seal and disclosure

Your agent	Fetches seal, badge and HTML code (get_seal) and states the tier and where the AI runs on the website and in the documentation (C-01).
NOCIP systems	The badge always shows the state of the register; the verification page detects whether it is embedded on a registered domain. Planned: automatic check of the statements after 7 and 30 days.
Person at NOCIP (only when needed)	Checks the disclosure at re-checks.
Decided by	NOCIP systems (program code)
Duration	ongoing
Tool	get_seal

S11 New versions

Your agent	Announces every new version before its release (C-05): test target, release notes and whether the AI architecture changes.
NOCIP systems	Downloads the new version and tests it. Green and no change to the AI architecture: the label also covers this version (register entry with the next update). Red, a new connection or a changed AI architecture: a suspension proposal to the auditor.
Person at NOCIP (only when needed)	Decides only if the re-check is not green or the AI architecture changes.
Decided by	NOCIP systems (program code)
Duration	a few minutes
Tool	notify_release
Log entries	release.notified, label.version_added, label.suspension_proposed

Tools and REST calls

MCP tool	REST	Purpose
get_status	GET /cases/{case}	state of the case and next steps
prove_domain	POST /cases/{case}/domain-proof	have the domain proof checked
submit_self_assessment	POST /cases/{case}/self-assessment	send the self-assessment
add_evidence	POST /cases/{case}/evidence	evidence: address or file
set_test_target	POST /cases/{case}/test-target	name the release to test
request_test	POST /cases/{case}/test-runs	request a lab test
list_findings	GET /cases/{case}/findings	read the findings
answer_finding	POST /cases/{case}/findings/{id}/answer	answer a finding
get_case_log	GET /cases/{case}/log	signed case log
get_report_draft	GET /cases/{case}/report-draft	draft report
get_seal	GET /cases/{case}/seal	seal and HTML code after the award
notify_release	POST /cases/{case}/releases	announce a new version

MCP tool	REST	Purpose
get_criteria	GET /criteria	criteria catalogue with SHA-256 (no token)
withdraw	POST /cases/{case}/withdraw	withdraw the application

7 Formats

Self-assessment (excerpt)

https://nocip.com/assets/api/self-assessment.schema.json


```
{
 "schema": "nocip-self-assessment/1",
 "criteria_version": "1.0",
 "tier": "G",
 "variant": null,
 "product": {
  "name": "LexITanon Desktop (Linux)",
  "version": "0.5.0",
  "platforms": [
   "linux-x64"
  ]
 },
 "ai_functions": [
  {
   "name": "Erkennung personenbezogener Angaben (Pseudonymisierung, Schwärzung)",
   "location": "customer",
   "model": "eigenes Erkennungsmodell 1.3.3 (BERT, Token-Klassifikation), ONNX Runtime auf der CPU",
   "provider": "be next GmbH"
  }
 ],
 "answers": {
  "C-01": {
   "answer": "yes",
   "text": "Startseite und KI & Berufsrecht: vollständig lokal; Startseite nennt NOCIP Gold Nr. G-2026-
0001.",
   "evidence": [
    "https://lexitanon.com/",
    "https://lexitanon.com/ki-und-berufsrecht/"
   ]
  },
  "G-01": {
   "answer": "yes",
   "text": "Die Erkennung läuft im Programm auf der CPU, ohne Netzwerk.",
   "evidence": [
    "https://lexitanon.com/assets/docs/LexITanon_Handbuch.pdf"
   ]
  }
 },
 "index": {
  "I-5": {
   "claim": 2,
   "text": "Handbuch Kap. 2.3: Ziele, Zweck, Häufigkeit.",
   "evidence": [
    "https://lexitanon.com/assets/docs/LexITanon_Handbuch.pdf"
   ]
  }
 }
}
```

Connection manifest

https://nocip.com/assets/api/connections.schema.json


```
{
 "schema": "nocip-connections/1",
 "label": "NOCIP-G-2026-0001",
 "product": "LexITanon Desktop (Linux)",
 "version": "0.5.0",
 "connections": [
  {
   "host": "lexitanon.com",
   "port": 443,
   "protocol": "https",
   "purpose": "licence-check",
   "when": "Desktop once a day, Server hourly",
   "data": "licence key, version, device hash, usage counts",
   "volume_per_day_kb": 5
  },
  {
   "host": "key.lexitanon.com",
   "port": 443,
   "protocol": "https",
   "purpose": "licence-check-fallback",
   "when": "only if lexitanon.com cannot be reached"
  },
  {
   "host": "lexitanon.com",
   "port": 443,
   "protocol": "https",
   "purpose": "update-download",
   "when": "only when the user clicks Jetzt aktualisieren"
  },
  {
   "host": "lexitanon.com",
   "port": 443,
   "protocol": "https",
   "purpose": "user-report",
   "when": "only when the user sends Fehlerhafte Erkennung melden",
   "data": "line 2 with invented data, consent version"
  }
 ],
 "ai_processing": [
  {
   "function": "recognition",
   "location": "customer",
   "network": false,
   "provider": "be next GmbH"
  }
 ]
}
```


Test target

https://nocip.com/assets/api/target.schema.json


```
{
 "schema": "nocip-test-target/1",
 "kind": "linux-x64-binary",
 "version": "0.5.0",
 "download_url": "https://lexitanon.com/programm/?os=linux",
 "download_auth": "sha256:<upload a file with: Cookie: ...>",
 "sha256": "9570b9ed3a93cd910dfdf92b1574bbc4ca0d7ad520bca4ed86760139a35f2f32",
 "size": 157309040,
```

```
 "files": {
  "licence": "sha256:<upload the test licence>"
 },
 "network": {
  "local_ips": [
   "178.63.52.93"
  ]
 },
 "selftest": {
  "args": [
   "--self-test"
  ],
  "expect_exit": 0,
  "expect_stdout": "self-test passed"
 },
 "start": {
  "args": [
   "--licence",
   "{file:licence}",
   "--ui-port",
   "0",
   "--api-port",
   "{port}",
   "--no-supervisor",
   "--licence-state",
   "{state}/licence.json",
   "--session-log",
   "{state}/session.txt",
   "--workers",
   "1",
   "--threads-per-worker",
   "2"
  ],
  "ready_http": "http://127.0.0.1:{port}/api/v1/info"
 },
 "process": {
  "http": {
   "url": "http://127.0.0.1:{port}/api/v1/redact-text",
   "json": {
    "text": "{text}"
   }
  }
 }
}
```

The complete formats are published as JSON Schema at the addresses given.

8 Verifying the case log

Every entry has the fields seq, time, case, step, action, actor, inputs, data, prev, hash and sig. hash is “sha256:” and the SHA-256 of the entry without hash and sig, as JSON with sorted keys, no spaces, UTF-8. prev is the hash of the entry before (null for the first). sig.ed25519 is the Ed25519 signature of the text of hash with the public key of NOCIP.

https://nocip.com/assets/api/nocip-case-signing.pub

```
import base64, hashlib, json
from cryptography.hazmat.primitives.asymmetric.ed25519 import Ed25519PublicKey
key = Ed25519PublicKey.from_public_bytes(base64.b64decode(PUBLIC_KEY_B64))
prev = None
for i, ev in enumerate(events, 1):
```

```
    body = {k: v for k, v in ev.items() if k not in ('hash', 'sig')}
    raw = json.dumps(body, sort_keys=True, separators=(',', ':'), ensure_ascii=False).encode()
    assert ev['seq'] == i and ev['prev'] == prev
    assert ev['hash'] == 'sha256:' + hashlib.sha256(raw).hexdigest()
    key.verify(base64.b64decode(ev['sig']['ed25519']), ev['hash'].encode())
    prev = ev['hash']
```

9 Limits and confidentiality

- The lab currently tests Linux programs (x86-64) that start without their interface and process a document through a local HTTP interface or the command line. Windows, macOS and cloud services are audited with documents for now.
- A person reads terms and documentation. Preparation by an AI model at NOCIP is planned.

Your documents, releases and test results stay on NOCIP's servers. No document goes to a cloud AI service. The test corpus with the canary values stays private and is different for every case. NOCIP keeps only the hash of the token; a new token makes the old one invalid.

10 Questions

Do I need an AI agent?

No. The REST interface can be called from any build system or script, even by hand with curl. The MCP server makes it especially easy for an AI agent.

Which agents work?

Any MCP client that supports Streamable HTTP with a header for the token, for example Claude Code. Everything else through the REST interface.

Can the agent award the label itself?

No. There is no tool and no call for it. A person at NOCIP signs award, refusal and suspension.

Is the result the same as in the audit with documents?

Yes: the same criteria, audit checklist, audit report and label. The signed case log comes on top.

What does the way “Automated via MCP server” cost?

Nothing, like the way “With documents”.

My product runs on Windows or in the cloud.

Then we audit with documents for now; you can still send the self-assessment and evidence through the interface. The lab for Windows, macOS and cloud services follows.