

FREE runtime

\$3,000 — Non-exclusive software license

October 2026 · Seller: Martin Gettig · Pennsylvania, USA

The offer

FREE runtime is a local operator software stack. The archive contains the HTTP service, operator sign-in, scheduled job runner, operator desktop, phone and mobile source, deployment examples, and tests.

You run it on a machine you control. You supply the model, host, accounts, and operating environment.

This is a bounded, technically demonstrable software deliverable. The packaged runtime has been booted, its health endpoint inspected, and its reported test suite executed. This sheet identifies the checks available to you and the limitations of their evidence.

FREE runtime is not a production-certified platform, a finished commercial service, or a promise of business results. You are not buying a customer list, hosted service, company, development team, or the private environment in which the software was built.

The price is \$3,000 for one non-exclusive license. You receive permission to use, modify, and operate the software under the FREE Runtime Software License Agreement. You do not acquire its copyright.

Why this instead of something else?

You have cheaper alternatives: a hosted chatbot, a model framework, a free starter project, or a developer hired to build a comparable system. FREE runtime is worth considering only if its existing combination of operator access, scheduled work, desktop visibility, and governance is what you want to acquire.

Not a hosted assistant

A hosted chatbot or SaaS agent runs on infrastructure controlled by its provider. This runtime is intended to run on a machine you control. You supply and pay for your own model endpoint and hosting environment. The seller does not charge you a runtime subscription under this offer.

You do not receive a hosted service, an existing customer base, or a product already earning revenue. If you need a public chatbot that someone else operates for you, this is not that purchase.

Not just a model framework

LangChain, LlamaIndex, n8n, and prompt collections can help connect models, tools, and workflows. FREE runtime packages a different combination: an HTTP service, operator sign-in, a scheduled job runner, a desktop interface, and a constitution-based policy gate.

The central design distinction is that a model's output is not, by itself, authorization to perform an action. The runtime includes policy and confirmation mechanisms intended to govern actions. The checks in this document demonstrate particular behaviors; they do not establish complete enforcement across every execution path.

Here the model can talk. It cannot approve its own work. That separation is the purchase.

Not a one-route starter

A minimal FastAPI example can answer an HTTP request. This archive contains a larger software tree with a health endpoint, session handling in front of administrative routes, scheduled jobs, an operator desktop, deployment examples, and governance-related code.

The packaged copy was reported to have booted, and 41 tests were reported passing against that packaged tree. The evidence and commands are described below. A successful boot and passing tests are a starting point for evaluation, not proof of hours saved, sustained operation, or production readiness.

Not a hiring contract

The software includes an existing implementation of these components. The price is not a claim that a particular number of development months has been saved; no equivalent build was timed for this comparison.

One payment buys the licensed software archive. It does not buy a retainer, a development team, ongoing operations, or the seller's continued availability.

Walk away if you require guaranteed revenue, a hosted chatbot, an included App Store listing, or a system independently certified for production security. Walk away if your business model requires redistributing the runtime itself to your customers; this license prohibits that.

Consider the offer if you want an operator runtime to inspect, adapt, and run on infrastructure you control, and the existing implementation is worth more to you than starting from an empty project.

What the archive contains

The stated inventory is:

- **HTTP runtime:** A Python HTTP service with a `/health` endpoint that reports service status, build information, and governance-related fields.
- **Operator sign-in:** Login and session handling intended to protect administrative routes.
- **Scheduled job runner:** Named functions and a scheduler that determines when jobs are due and records execution state.
- **Operator desktop:** A desktop interface for inspecting service state, logs, and jobs.
- **Model connection:** Administrative chat can connect to a model endpoint you supply, including a local Ollama endpoint or an OpenAI-compatible endpoint.
- **Phone and mobile source:** Client source that communicates with the runtime service. The App Store listing and `node_modules` are not included in the archive.
- **Deployment examples:** Examples for nginx, systemd, and Caddy. The seller's real network addresses have been replaced with documentation-only address ranges.
- **Tests and inspection scripts:** The reported test suite and the buyer demonstration described later in this document.

The service can be started without requiring a model to be available for startup when model warmup is disabled as shown in the proof procedure. This document does not use that test to claim that every model-dependent feature works without a configured model.

What you can do with it

You may install the runtime on a machine you control and adapt it for your own personal or commercial use. You supply your own host, model endpoint, credentials, jobs, and operating configuration.

You may operate a product or service of your own that depends on the runtime. You may charge your customers for that product or service, subject to the license below.

The archive does not provide customer registration, a billing system, public hosting, or an already operating commercial service. You are responsible for integrating and operating whatever you build around it.

Why the governance is there

A runtime that can call a model and execute jobs needs to distinguish a requested action from an authorized action. FREE runtime includes a constitution file and policy-related mechanisms intended to make that distinction explicit.

The constitution describes rules including:

- Actions that are forbidden.
- Actions requiring human confirmation.
- Actions requiring two approvals.
- Directories the process may write to.

Changing the constitution changes the configured rules. It does not establish that the file has integrity.

The documented integrity procedure checks the constitution against `constitution.sha256` and checks `constitution.sig` using a signing key available to the process. When the integrity check fails, the tested policy gate refuses the demonstrated action with an integrity-related refusal. The `/health` endpoint exposes status fields related to the loaded charter and governance state.

You choose the rules, generate the key, and sign the constitution you intend to use. The clean-install procedure appears below.

The archive does not include the seller's private keys, credentials, private roles, logs, memory, or household environment. The governance-related code is included; the original private environment is not.

What a valid signature establishes—and what it does not

A passing signature establishes integrity relative to the key the process actually reads. It indicates that the signed content matches the expected hash and signature for that key.

It does not establish that the rules are safe, that the key is protected against theft or replacement, or that every sensitive execution path consults the same policy gate. No penetration test, independent security audit, or production certification is included.

The mechanism is inspectable and testable. Its existence must not be confused with proof of complete security enforcement.

The license

The grant and the restrictions are stated in the FREE Runtime Software License Agreement, version 1.2, October 2026. That file ships in the archive as `FREE-SOFTWARE-LICENSE.md`. The same text is published at <https://app.voxryn.ai/offer/free-software-license.md>. What follows is the summary.

Price: \$3,000 for one non-exclusive license.

The license grants permission to use the software. Copyright remains with Martin Gettig, who may license the same software to other buyers. Payment does not transfer copyright ownership or make the buyer the exclusive owner of the codebase.

Subject to the license terms, you may use, modify, and run the software on machines you control for your own personal or commercial purposes. You may operate your own product or service using the runtime and charge your customers for that product or service.

Your customers receive the product or service you operate. They do not receive this license, the runtime source, or a copy of the runtime to install and operate independently.

Redistribution restrictions

You may not sell, give, rent, sublicense, or relicense the software, its source, or a substantial part of the runtime in a form that lets another party install, modify, or operate it as their own copy.

You may not distribute the archive, a fork of it, or the runtime tree inside an installer, image, or appliance intended for the recipient to run. You may not offer the runtime itself, including a relabeled copy, as a product.

Dependence is not delivery. If your product calls a FREE runtime service that you operate, you are using the runtime to provide your product. If you distribute the runtime or a substantial part of its code so recipients can run it themselves, that is redistribution under the intended boundary of this license, even if you wrap it in your own software.

Commercial use does not require you to hand the runtime codebase to your customers.

These paragraphs summarize the Software License Agreement. If a summary and the Agreement differ on the grant or a restriction, the Agreement controls. This sheet remains the product description: what is in the archive, how to inspect it, and what evidence was reported. Neither document is a legal opinion from a law firm. Read both, and have counsel read them, before purchasing. The source archive is not attached to the public website.

What the \$3,000 does not buy

The payment does not include:

- Revenue, customers, or a billing system.
- The iPhone app listing on the App Store or an existing subscriber product.
- The seller's website or domain.
- Hosting or ongoing operation by the seller.
- Passwords, API keys, exchange keys, domain keys, or private household data.
- A model subscription or the cost of running a model.
- A support retainer or support after delivery. Any further assistance requires a separate agreement.

- The right to resell or redistribute the runtime codebase as your own product.
- A guarantee of performance, security, compatibility, revenue, or return on investment.

The software archive is the deliverable. The buyer supplies the environment in which it will be used.

What the price must be weighed against

The \$3,000 price is an investment to weigh against building an equivalent system or assembling and integrating alternatives. This document identifies the software, provides inspection steps, and describes evidence reported from the packaged tree. It does not calculate the buyer's return.

Time saved: No equivalent implementation was timed. This sheet makes no claim that the runtime saves a specified number of hours, weeks, or months.

Sustained operation: The reported packaging proof started the service and queried `/health`. It did not establish a multi-hour or multi-day soak, a load limit, or long-term reliability.

Execution-path coverage: The checks below demonstrate a login, a due-job decision, an allowed action, a forbidden action, and integrity failures and recovery after signing. They do not demonstrate that every sensitive action follows the same integrity check or that the runtime is hardened against a determined attacker. No penetration test, independent audit, soak test, or production certification is included. These limitations bound the claims made here; they do not, by themselves, prevent the archive from being delivered.

Adaptation: The handler table represents the scheduler's registered handlers, not a list of finished business features. Some jobs were written for the seller's original private environment and may look for files that are not included. You must supply the host, model, accounts, configuration, and constitution appropriate to your use.

Inspect the archive yourself

From the unpacked archive, run:

```
python scripts/buyer_value_demo.py
```

The demonstration uses a temporary login file and deletes it at the end. It does not open the network or execute the registered handler table. The following is the reported output from the demonstration when this sheet was prepared; run it yourself to check the archive you receive.

```
login accepted=True role=admin wrong_password_rejected=True
job due_when_never_run=True ran_when_due=True due_just_after_run=False ran_when_not_due=False
handler_table_size=75 handlers_executed=0
permitted toolchain_verify_health role=runtime allowed=True reason=allowed_runtime
refused doctrine_change role=admin allowed=False reason=constitution_forbidden_action
constitution intact_hash_ok=True intact_integrity_ok=False intact_reason=ok|missing_signature_key_env
constitution changed_word_hash_ok=False reason=hash_mismatch|missing_signature_key_env
constitution gate_allowed=False gate_reason=constitution_integrity_required
constitution unparseable=JSONDecodeError gate_reason=constitution_integrity_required
```

1. Login

The demonstration creates a temporary account file, accepts the correct password for an account with the admin role, and rejects an incorrect password. The temporary file is deleted after the run. It is not the seller's account list.

2. Scheduled job

The demonstration tests the scheduler's due check, `co_developer_autonomous_jobs._due`. A job that has never run is due; the local probe runs once. When the same job is recorded as having just run, it is not due again and the probe does not run a second time.

The registered handler table contains 75 entries, and this demonstration executes zero of them. The table describes registered handlers, not completed business features.

3. Permitted action

The demonstration checks `toolchain_verify_health` for the runtime role and receives `allowed_runtime` under the tested conditions. This demonstrates a specific allow-list result. It does not mean the action remains permitted when the integrity gate fails.

4. Forbidden action

The demonstration checks `doctrine_change` for the admin role and receives `constitution_forbidden_action`, even when the request is marked confirmed and dual-approved. In this tested case, administrator status and approvals do not override the forbidden-action rule.

5. Constitution integrity

The shipped constitution's hash matches `constitution.sha256`, but the demonstration reports `integrity_ok=False` because the signing key is not available in the environment. The included signature was generated with a key that is not supplied in the archive. There is no seller key you need to install.

Changing one word in the constitution causes a hash mismatch. Making the constitution invalid JSON causes a `JSONDecodeError` and an integrity-related refusal in the demonstrated gate. The next section explains how to generate your own key and sign the constitution you intend to run.

Prepare a clean installation and sign the constitution

The shipped signature is not intended to pass using a key supplied with the archive. Generate your own key, sign the constitution you intend to use, and ensure the service reads that same key.

From the unpacked directory, save the following as `make_signing_key.py`:

```
import secrets
from pathlib import Path

Path("constitution/signing.key").write_text(
    secrets.token_hex(32) + "\n",
    encoding="utf-8",
)
```

Run it:

```
python make_signing_key.py
```

Then sign the constitution:

```
python scripts/sign_constitution.py
```

The signing script checks for a key in this order:

- 1 `FREE_CONSTITUTION_SIGNING_KEY`

2 FREE_CONSTITUTION_SIGNING_KEY_FILE

3 constitution/signing.key

It rewrites `constitution.sha256` and `constitution.sig`. A successful run is expected to report `integrity_ok=True reason=ok`.

Start the runtime with the same key available to it. You can use the local key file or the supported environment-variable configuration. The service reads the first available key according to its configured precedence. If a higher-precedence environment key differs from the key used to sign the constitution, integrity verification should fail rather than silently accepting the mismatch.

If you edit `constitution.json`, sign it again before expecting the integrity gate to pass.

Reported signing demonstration

The buyer demonstration also signed a temporary copy of the constitution and reported these results:

```
before_sign integrity_ok=False reason=ok|missing_signature_key_env
after_sign integrity_ok=True reason=ok gate=allowed_runtime
forbidden_after_sign allowed=False reason=constitution_forbidden_action
edited_without_resign integrity_ok=False hash_ok=False gate=constitution_integrity_required
resigned integrity_ok=True reason=ok gate=allowed_runtime
```

The reported sequence shows the demonstrated integrity gate refusing the tested action before signing, allowing the tested permitted action after signing, continuing to refuse the forbidden action, refusing the permitted action after an unsigned edit, and allowing it again after the modified constitution is signed.

The temporary key is deleted at the end of the demonstration. The demonstration does not create `constitution/signing.key` in the shipped archive.

A successful signature is not a security certification. It does not prove that the constitution is safe, that the key cannot be compromised, or that every sensitive execution path uses the policy gate.

The purpose of this procedure is to let you inspect the mechanism and test the behavior against your own requirements.

Boot proof from the packaged tree

The packaging proof was reported against the packaged tree rather than the seller's working copy.

Reported results:

- GET /health returned `ok=True`.
- Reported version: 1.0.0.
- Reported API build: 20261009-07.
- Functions with names beginning `job_`: 76.
- Registered scheduled handler entries: 75.
- Tests executed in the packaged tree: 41 passed in 1.02 seconds.

The count of `job_` functions includes helpers that are not scheduled handlers. Neither count represents a number of finished features.

Reported boot procedure

The following commands show the reported Windows Command Prompt procedure, using Python 3.12:

```
python -m pip install -r requirements-api.txt
set FREE_MODEL_WARMUP_ENABLED=0
python -m uvicorn runtime_api:app --host 127.0.0.1 --port 8787
```

Then open:

```
http://127.0.0.1:8787/health
```

Check the JSON response and confirm that `ok` is `true` and the build information matches the archive described by this sheet.

Model warmup was disabled for the reported boot proof so startup would not depend on a local model being available. This proof does not, by itself, establish model-dependent functionality or startup behavior with warmup enabled.

The reported test result was:

```
..... [100%]
41 passed in 1.02s
```

A passing suite is evidence only for the behavior covered by those tests. It is not production certification or proof that every sensitive action follows the same integrity check.

Verify the exact archive you receive. The build number and test results in this document describe the reported packaged-tree run. They should not be assumed to describe a different archive or a later build.

What to expect after delivery

The archive is a software system, not a claim that every module is immediately useful in every buyer's environment.

Some modules and jobs were written for the seller's original private environment. Their source code is included, but the original environment's logs, keys, memory, and private data are not. A job that expects one of those withheld files may fail or report that the file is missing. That does not, by itself, indicate that the HTTP service cannot boot; the reported health check does not require those jobs to run.

The deployment examples use documentation-only address ranges, including `192.0.2.0/24` and `203.0.113.0/24`, in place of the seller's actual network addresses. They do not point to the seller's network.

The reported proof boot exposed these top-level health fields:

```
api_build
api_file
builder_ack
charter_enforcement_scope
charter_enforcement_scope_plain
charter_guardrails_version
constitution_status
governance_lane_version
house_doctrine_revision
```

```
housekeeping_model_backend
jennifer_agency
jennifer_heart_network
jennifer_hybrid
jennifer_platform
maintenance
mercy_governance_version
model_backend
model_bridge_status
ok
risk_band
runtime_chapter
safety_state_level
stop_work_event_command
toolchain_co_developer
version
```

These fields are observable runtime information. Their presence does not independently prove that every feature they describe is complete, secure, or production-ready.

Where the purchase happens

The listing price is **\$3,000**, paid through Zero to Hundred:

<https://zerotohundred.ai/>

When the sale completes on Zero to Hundred, payment forms the license in Zero to Hundred's terms: non-exclusive, worldwide, and perpetual; the buyer may use, modify, and deploy the software for the buyer's own purposes, including in the buyer's own products; the buyer may not resell, re-list, sublicense, or redistribute the software as a product; copyright stays with the seller, who may license the same archive again. The FREE Runtime Software License Agreement, version 1.2, states what that grant means for this runtime, including the line between hosting a product and shipping the runtime. It is included in the archive. On Zero to Hundred the buyer has 24 hours after purchase to report a factual problem. A limitation disclosed in this sheet is not a report ground. The five business days in the Agreement do not extend that 24-hour window.

The public product page is:

<https://app.voxryn.ai/learn>

The product sheet is published there as `FREE-asking-price.pdf`. The source archive is not attached to that public page.

The build number in this sheet refers to the reported boot of the packaged archive. The build served by the public website may differ when you visit it. Buying the license does not include the website, domain, or website infrastructure.

Seller and ownership

Martin Gettig Pennsylvania, United States Individual seller; not a company.

© 2026 Martin Gettig. Copyright remains with the author. This license grants permission to use the software; it does not transfer copyright ownership.

Final statement

FREE runtime is offered as a defined software archive under a non-exclusive license. The inventory, permitted uses, redistribution restrictions, inspection procedures, reported results, and known evidentiary limitations are stated above.

The buyer can use the documented checks to inspect the package and judge it against their own requirements. The price is the price of the license—not a forecast of savings, a guarantee of security, or a promise of commercial success.

The archive is the deliverable. The inspection is how you decide whether it is worth \$3,000 to you.