Licensing integration: steps

Add license checks to your own product with 1Scribe. Follow the steps in order. Platform: `https://agentskope.com` (default); `https://1scribe.xyz` is an alternative address for the same platform. Use one address consistently.

The checklist at `/1scribe/licensing` follows the same order: Set up (Steps 1 and 2), Build (Steps 3 to 6), Webhooks (Step 7), Test and go live (Step 8). A green check there means AgentSkope has verified the step.

You do not need a published listing or a payout account for these steps. You can issue licenses through the API for sales made elsewhere.

Step 1. Create your product and a license plan

1. Create your product in the dashboard and note its **product id**.
2. Add at least one **license plan**. A plan sets the features, limits and seats that a license unlocks.
3. If a plan is delivered as SaaS, turn on **verified account activation** for it.

Step 2. Create a vendor API key

1. Open **Integrations** and choose **Create key**.
2. Give it the `entitlements:read` scope (add `licenses:issue` if you issue licenses through the API).
3. Use a `scribe_test_...` key while you build. Copy the key once; it is shown only once.
4. Keep the key on your **server only**. Never put it in an app or page that customers receive.

Step 3. Choose your package

Your product	Install	Where it runs
Node.js or TypeScript backend, or a SaaS with sign-in	`npm install @1scribe/server`	Your backend
Desktop app, CLI, Electron main process	`npm install @1scribe/client`	The customer's device
Python backend	`pip install 1scribe` (import `onescribe.server`)	Your backend
Python desktop app or Windows EXE	`pip install 1scribe` (import `onescribe.client`)	The customer's device
Servers, agents, CI runners	The client package, with a machine token	The machine
Any other language	Call the HTTP API directly	Your backend

The client side never holds a vendor API key. Never ship the server package inside anything customers receive. Node.js 18 or newer for npm; Python 3.9.2 or newer for PyPI. The npm packages are ES modules and do not run in a browser page.

Step 4. Add activation

Installed app: license key, email and a one-time code. The platform emails the code to the customer.

```
import {
  OneScribeClient,
  LicenseSession,
  FileLicenseStore,
  getOrCreateDeviceId,
} from '@1scribe/client';

const client = new OneScribeClient({}); // defaults to https://agentskope.com
const deviceId = await getOrCreateDeviceId(`${appDataDir}/device-id`);
const device = { id: deviceId, name: os.hostname(), version: APP_VERSION };

const requested = await client.requestCode({ key, email, device });
const activated = await client.activate({
  challengeId: requested.challengeId,
  code, // typed by the customer
  device: { ...device, referenceName: 'Front desk PC' },
});
if (activated.ok) await session.save(activated.document, activated.keys);
```

```
from onescribe.client import DeviceInfo, FileLicenseStore, LicenseSession, OneScribeClient,
get_or_create_device_id

client = OneScribeClient()
device = DeviceInfo(id=get_or_create_device_id(app_dir / "device-id"), reference_name="Front
desk PC")
code = client.request_code(key=license_key, email=email, device=device)
result = client.activate(challenge_id=code.challenge_id, code=typed_code, device=device)
if result.ok:
    session.save(result.document, result.keys)
```

SaaS with your own sign-in: after the user signs in, your backend vouches for them.

```
import { VendorApiClient } from '@1scribe/server';
const vendor = new VendorApiClient({ apiKey: process.env.SCRIBE_VENDOR_API_KEY! });
const access = await vendor.vouchedAccess({
  productId,
  subjectId: user.id,
  verifiedEmail: user.email,
});
```

```
vendor = VendorApiClient(api_key=os.environ["SCRIBE_API_KEY"])
access = vendor.vouched_access(product_id=PRODUCT_ID, subject_id=user.id)
```

Unattended machines: an administrator creates a machine token; the machine activates with the license key and the token, with no code.

Step 5. Check the license on every start

The check works offline from the stored signed document. It contacts the platform only after the offline grace period has passed.

```
const session = new LicenseSession({
  productId,
  client,
  store: new FileLicenseStore(`${appDataDir}/license.json`),
  deviceId,
});
const decision = await session.check();
if (!decision.allowed) show(decision.reason); // not_activated, revoked, expired,
offline_expired ...
```

```
session = LicenseSession(product_id=PRODUCT_ID, client=client,
                         store=FileLicenseStore(app_dir / "license.json"), device_id=device.id)
decision = session.check()
```

- Store the license file and device-id file in the user's app-data folder, not next to the program. A copied program folder creates a new device and counts against the device limit.
- Run the check in code a customer cannot edit: a Node or Python layer or your backend, never a browser page.

Step 6. Unlock features and enforce limits

Read `features`, `limits` and `plan` from the result and unlock accordingly. The platform delivers limits; **your product enforces them**.

Step 7. Webhooks (optional)

1. Register an HTTPS endpoint in the dashboard.
2. Verify the `X-Scribe-Signature` header against the **raw request body** before parsing it (HMAC-SHA256, five-minute replay window). Use `verifySignature` / `verify_signature`.
3. Answer 2xx only after you have handled the event.

Step 8. Test, then go live

1. Issue a **test license** and activate it from your product.
2. Run the four acceptance tests: buy and activate a test license; check feature and seat limits; test expiry and revocation; test a disconnection or a removed seat.
3. Replace the test key with a **live key** on your server.

If something fails

You see	What to do
`wrong_platform`	Use one platform address for the calls and the check
`wrong_device`	The program folder was copied; activate again on this device
`clock_rollback`	Fix the device clock
`offline_expired`	Go online; the SDK refreshes automatically
`device_limit` / `seat_limit`	Remove a device or free a seat

More: package pages https://www.npmjs.com/package/@1scribe/client , https://www.npmjs.com/package/@1scribe/server , https://pypi.org/project/1scribe/ ; HTTP API guide at `/1scribe/docs` ; packages and downloads at `/1scribe/licensing` .