External App Package 1.0

Drop a zip into Paramus - your app runs on every scientist's desktop, and its functions become tools of the Paramus agent.

What you ship

One zip, named after your app. Inside: a short manifest (paramus-app.yml, about fifteen lines), your application in whatever stack you choose, and one HTTP endpoint that answers the Model Context Protocol (tools/list, tools/call). Everything else - pages, design, data, settings - is yours.

```
my-app/
  paramus-app.yml      id, version, runtime, port
  <your code>          any language or stack
  /mcp                 one JSON-RPC endpoint
  /health              answers 200 when ready
  run.sh | Dockerfile  how it starts
```

What Paramus does

- Verifies the manifest and unpacks the zip into the user's local app folder.
- Shows your app under "Mounted" with an unverified badge; opens your UI as you deliver it.
- Registers your MCP tools with the agent. Every call is confirmed by the user until the user marks your app as trusted.
- Logs every call with its origin, so the user can always see what your app did.

What stays yours

Your code, your repository, your copyright, your licence, your user interface. A mounted app lives outside the Paramus product and is never copied into it. Paramus prescribes the manifest and the MCP endpoint - nothing about how your app looks or is built.

What is fixed - enforced by Paramus

- Nothing runs before the user confirms the first drop.
- Your app sees its own data folder only - never another app's.
- Your app cannot change Paramus settings.
- Your app is shown as unverified; the user may trust it, the badge stays.
- Only a person mounts an app. An agent cannot install one.
- Your app is reached on its assigned port only; extra listeners stop it. It holds no key to Paramus's own services.

Requirements

- Paramus 3.1 or later on the user's computer.
- Your app starts from a script (run.sh / run.bat) or a Dockerfile.
- One network port, assigned by Paramus at mount time and handed to your app (PARAMUS_APP_PORT, or the container port mapping). Bind nothing else.
- An MCP endpoint (/mcp) and a health endpoint (/health).
- A licence file of your choice inside the zip.

Get started

Start from the manifest below: put your first tool behind /mcp, add /health and a start script, zip, drop. The step-by-step guide ships with Paramus: Documentation app, how-to "Mount an app".

Questions or early access for your app:

sales@paramus.ai
paramus.ai/blog

Hello world - the whole manifest

```
id: hello-paramus
version: 1.0.0
profile: '3.0'
location: dropin
runtime: native          # or image
connection: mcp
mcp: { endpoint: /mcp }
policy: { license: MIT, confirm: always }
state: { kind: local, path: data/ }
```

External apps are entirely in the customer's hands - Paramus mounts, confines and logs them; it does not design them.