

## RESEARCH PAPER

# Company OS Architecture

Self-hosted multi-agent systems need isolation, scoped authority, approval gates, and auditability. Docker plus a second environment file isolates processes. Company OS isolates authority: who may act, which capabilities a worker gets, HOLD/DENY, and receipts.

THECRINERS LLC · 2026 · COMPANY OS 0.5.0-RC1 · SECURITY MODEL 1.2.0

## Problem

On one Linux host, people now run several agents — work versus personal, two clients, a research bot next to a mail bot. They share a user account and often one environment file. If a token exists in that shared environment, any agent can use it. “Confirm before acting” is frequently just a prompt. Prompts get ignored. After a bad send, there is nothing to open in the morning except rotated logs.

Company OS is the authority layer around that mess. It is **not** a process or syscall sandbox. If the pain is “jail the process,” use Docker. That sentence is a test, not marketing copy.

## Engine versus tenant

The engine is a Python package. A company is a directory: its own `tenant.json`, sqlite database, token, and port. Names, missions, channels, and STOP lists load from disk. The package is not allowed to hard-code the owner’s brand names as if they were the product.

```
Owner
  → Director
    → Supervisors
      → Workers
```

All of those names come from `tenant.json`.  
The engine supplies leases, receipts, grants, and the dashboard bind rules.

Org shape is data. Authority machinery is code.

## Process model

One Python process per tenant for tick or dashboard. Packet Press's public life is a company on that model; Money Hunter remains a separate laboratory and is **not** a Company OS tenant. New companies land in their own roots. Dashboards bind loopback (and optional RFC1918 LAN), never 0.0.0.0.

The platform control plane can list installed tenants by path. It does not get a free look at their secrets.

## Modules that matter to an outsider

- **tenant** — identity from disk.
- **work / lease** — durable assignments; exclusive ownership of (work, stage).
- **receipts** — evidence after the fact.
- **capabilities** — ALLOW / HOLD / DENY, not API keys in JSON.
- **approvals** — owner queue. An approved row is not an executor.
- **secrets** — chmod 600, declared names, no shared-file ingest.
- **backup** — export/import without installing a vault.
- **bus** — signed envelopes between companies when the umbrella is in use.

## Implementation notes

Synchronous and small on purpose. Money Hunter's thread pool stayed in Money Hunter. Company OS 1.0 is still the wrong name; the public tag is a release candidate. The live portfolio uses the engine. That is an existence proof for tenancy, not a claim that the market has pulled the product out of our hands.

## Tests

The public tree is expected to pass `python3 -m unittest discover -s tests`. Outsider clone plus `bash scripts/docker_eval.sh` should print `EVAL_OK`, with no dashboard and no 0.0.0.0. A green run is necessary, not a certification.

## Limitations

- Not a kernel sandbox. A process with the Unix user's rights can still be that user.

- In-process sqlite access is out of scope for the frozen model.
- WAN-published dashboards are refused as a design, not as a missing feature.
- LLM supply chain and “the model will just obey” are out of scope.

## Lessons

If tenant policy is compiled into the engine, the second company never happens. If secrets travel with grants, isolation is a comment. If restore can install secrets .env, backup becomes an attack. Those are defects we actually hit and then froze tests around.

HTML is canonical. A PDF companion is offered where generated. Experimental laboratory results are labeled as such. This is not a product-market-fit claim and not a certification.

[← All papers](#) · [PDF](#)

**TheCriners LLC** · Coshocton, Ohio

Money Hunter, Packet Press, Pullsheet, Hearthlabel, Seatpack, and Company OS are projects of TheCriners LLC — not separate legal entities.