

# Agent card sample (anonymised)

One-page sample of how execution agents are contracted in Agent OS 2.0. Forward internally — shows roles, boundaries and outputs. No client names.

---

## Why agent cards matter

A multi-agent system without contracts drifts fast. Each agent gets:

- a **single role** (no mega-agent),
- **least-privilege tools** (only what the task needs),
- a **clear output** (plan, diff, test report, sign-off request, handoff).

This sample mirrors the live 5-node LangGraph pipeline — names anonymised.

---

## Sample cards (LangGraph pipeline)

Agent	Core role	Allowed scope	Typical output
<b>Planner</b>	Break work into atomic tasks	Read SSoT docs, task list	Implementation plan + file list
<b>Coder</b>	Implement one module per session	Target folder only, git diff	Working code in test environment
<b>Tester</b>	Verify behaviour & regressions	Test scripts, read logs	Pass/fail report with reproduction steps
<b>Reviewer</b>	Human gate (HITL)	Approve / reject / cancel	Signed-off diff or blocked deploy
<b>Summarizer</b>	Session closure	Handoff + session notes	Handoff markdown on disk

**Pipeline:** Planner → Coder → Tester → Reviewer → Summarizer — fixed order, no skipped steps.

---

## What each agent cannot do

- **Planner** cannot deploy to production or edit live customer data.
  - **Coder** cannot change global business rules without schema approval.
  - **Tester** cannot bypass failing checks.
  - **Reviewer** is not optional on production paths — deploy stays manual by design.
  - **Summarizer** does not auto-deploy — handoff only.
- 

## Hybrid honesty

- **VPS (LIVE):** orchestration, queue, HITL, Mission Control, Langfuse.
  - **Local runner (LOCAL-ONLY):** code/git execution on dev PC — tasks wait in `WAITING_RUNNER` when PC is offline.
  - **VCMS** governs repos and SSoT — outside the LangGraph pipeline.
- 

## Stack (process proof)

- **Control plane:** FastAPI + LangGraph + PostgreSQL on EU VPS
  - **UI:** Mission Control (agent-os-ui) — tasks, queue, history, costs
  - **Measurement:** prod smoke PASS — architecture diagram and 60s demo on case study
- 

## Next step

Want this level of structure for your business operations? Start with a paid **Automation Map**.

**Book:** [services.flexgrafik.nl/book-discovery/](https://services.flexgrafik.nl/book-discovery/)

---

Norbert — Conversion Systems Architect · Quietforge · [hello@flexgrafik.nl](mailto:hello@flexgrafik.nl)