

Release the mops!

How to let coding agents loose
on MPS projects without driving you crazy

Sergej Koščejev / specificlanguages.com



Motivation: agents and MPS

- Agents can *read* individual MPS models out of the box
 - “The problem is in the e3XyzB with id 12fGTH, it should reference 25ghKX instead”
- Agents get lost trying to *edit* MPS models
 - Guessing IDs, guessing structure, adding references, unable to validate
- Agents need help: tooling and guidance



How should the tooling work?

- The default answer is: an MCP plugin for MPS
- But MPS is single-instance, single-repository, single-user
 - “I run the agent and I watch it”
- How do I run multiple agents?



8 levels of AI adoption (Steve Yegge)

- **Level 1:** no AI
- **Level 2:** Coding agent in your IDE, permissions turned on
- **Level 3:** Coding agent in IDE, “YOLO mode.” Your trust is going up.
- **Level 4:** you’re starting to not look at the diffs anymore, but at what the agent is doing. You’re not reviewing as much, and you’re really focused on the conversation with the agent.
- **Level 5:** “I just want the agent to work, and I’ll look at the code in my IDE later, but I’m not coding with my IDE”.
- **Level 6:** several agents. You’re bored because your agent’s busy and you want to do something, so you fire up another agent, then another.
- **Level 7:** 10+ agents, managed by hand.
- **Level 8:** you build your own orchestrator to coordinate more agents.



How do I run agents in parallel?

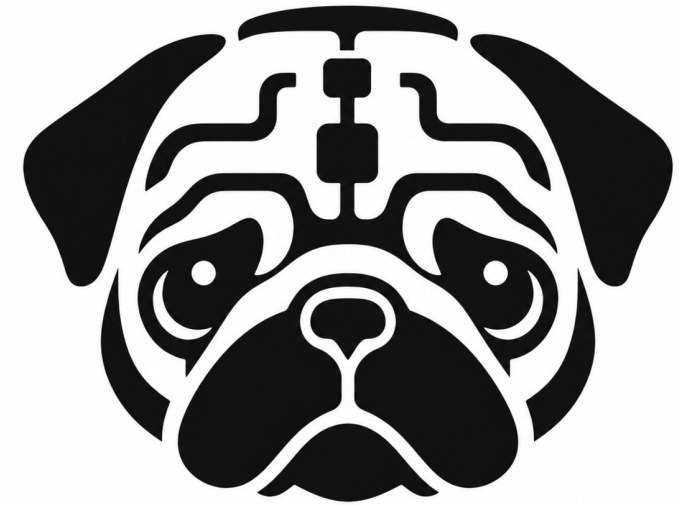
- Give each agent an isolated, headless instance of MPS
- This is how `mops` 🐶/🧹 works



mops – MPS OPerationS



OR

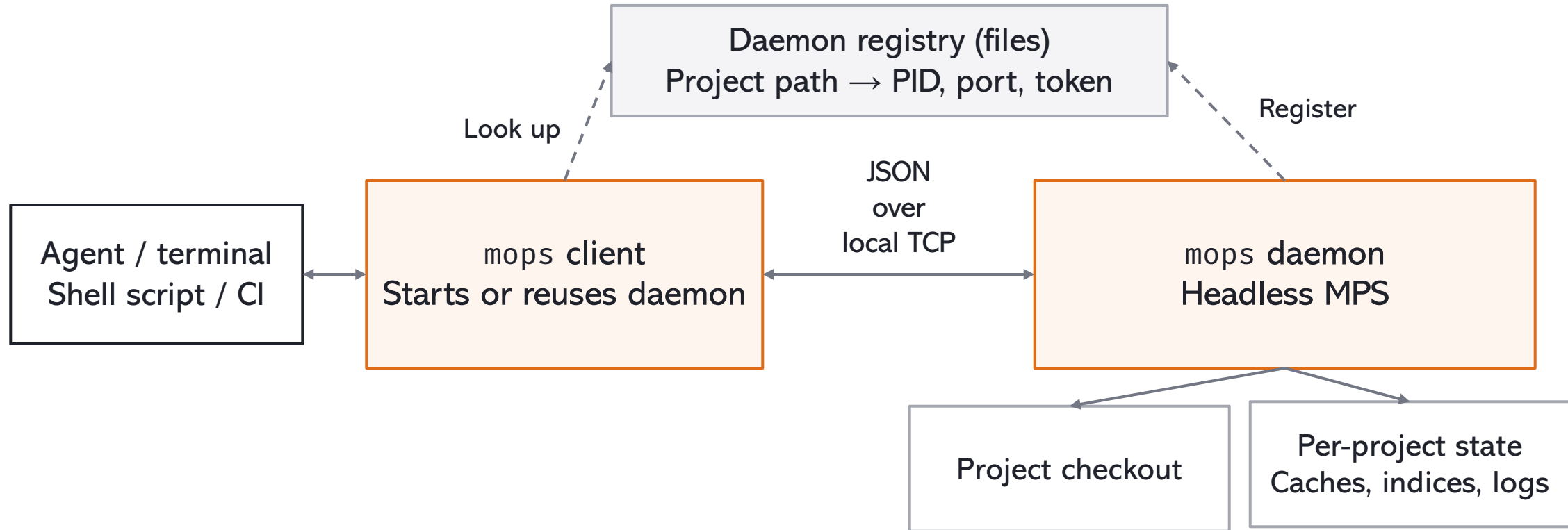


mops – Architecture

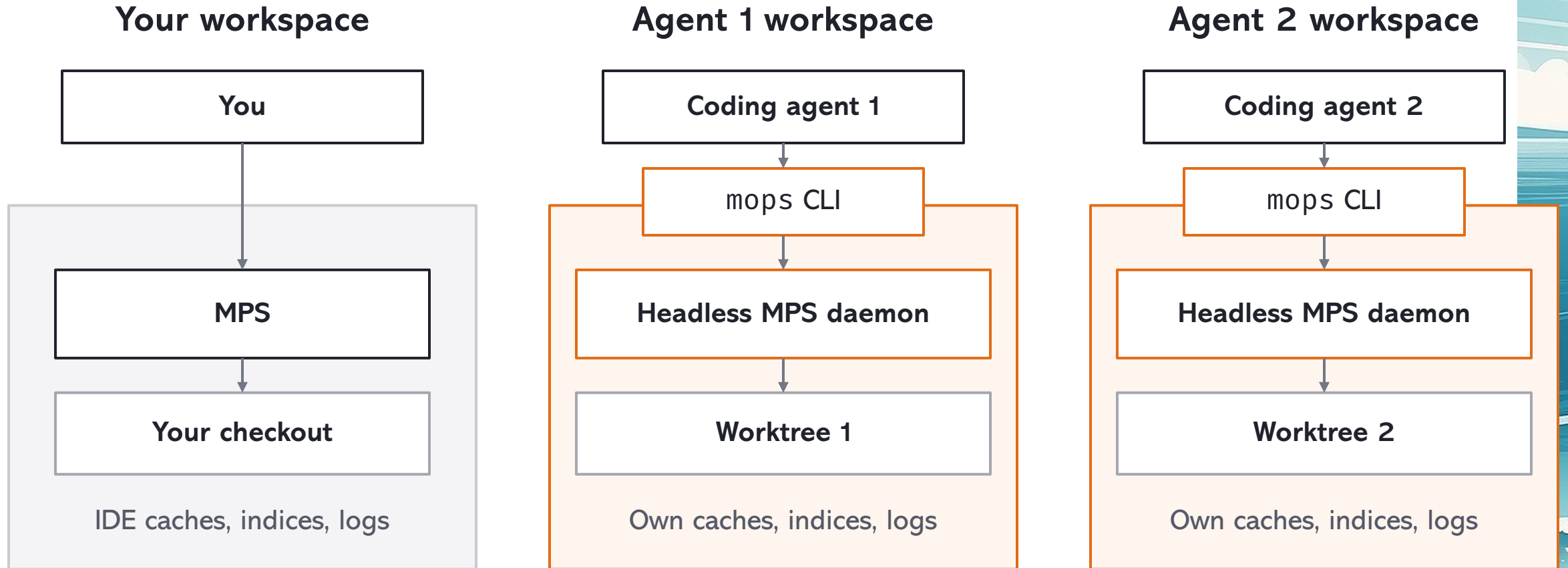
- Headless MPS "daemon" in the background
 - Inspired by Gradle
- Isolated daemon instance per MPS project
 - Separate caches, indices, logs
 - Project-specific MPS version
- mops starts the daemon if not running, then talks to it
- The daemon persists for a certain time, then dies
- mops client is a command-line interface (CLI)



mops – Architecture



One MPS instance per checkout



Keep working in your MPS instance while agents work in their own checkouts

Why is mops a CLI?

- Could have been a local MCP server (studio)
- ~~More context-efficient~~ – maybe?
- A shared interface for humans, automation, agents, and CI
 - Spend tokens once to write a script, run it many times for free



Features

- Project isolation – unleash your agents, come back for results
- Support for multiple MPS versions (2024.1 and above)
- Strict – less guessing, more diagnosis
- Open source (<https://github.com/specificlanguages/mops>)
- Open-source “breadcrumbs”:
 - [mps-ai-benchmarks](#)
 - [mps-api-research](#)
 - mps-agent-guidance (planned)



Basic commands

- Reading: `find`, `list`, `get`, `render`
- Writing: `edit`
- Checking: `check`, `make`, `test`
- Scripting: `code run` – Groovy, with domain-specific extensions:
 - parsing Java (`mops.parsing.java`)
 - reloading build scripts
(`mops.editing.build.reloadModulesFromDisk`)
 - (more to come)



Example script: names of all classes

```
def conceptName = 'jetbrains.mps.baseLanguage.ClassConcept'  
  
return project.read {  
    def concept = mops.lookup.requireConceptByName(conceptName)  
    def names = []  
    mops.search.eachInstanceOf(concept, project.scope) { node ->  
        names << node.properties['name']  
    }  
    return names  
}
```



mops – More commands

- Diagnosing: diagnose, daemon
- Help: help, explain, example, code help
- Agent guidance: skill path
- And more added as required, as discovered by field tests



mops – Field testing

- Used so far manually for smaller, down-to-earth tasks
 - Helping with migrations 💰
 - Reproducing bugs
 - Reviewing PRs
- Goal: set up automation and use mops to
 - Reproduce bugs reported against MPS libraries (MPS-extensions, mbeddr platform, mps-qa) and write unit tests.
 - Review PRs, leave comments



Demo



Which commands get used?

- GPT 6.1 Sol (Medium)
- MPS-extensions repository
- Reviewing a new feature (PR #1909)
- Reproducing a bug (issue #843)

Command	PR review	Bug reproduction
code run	6	32
render	19	2
list	7	0
find root-by-name	8	0
find usages	2	0
check	2	9
make	0	3
test	1	3
Total	45	49



Installing mops

```
gh repo clone specificlanguages/mops # no published binaries yet
cd mops
./gradlew installMops # installs to ~/.local/bin

cd /my/project
mops wrapper          # requires a Gradle build
./gradlew assemble # build everything

codex exec "Review my current changes. Use mops CLI, see `mops --help`."

# go do something else in the meantime 🐼
```

Installing mops



Learnings & takeaways

- Isolation is more difficult
- Isolation is worth it
- Vibe-coding has been (so far) worth the tradeoffs
- Skills and guidance are as important as tools
- Consistency and guardrails will become more important
 - *How to put your MPS project on the rails – and keep it there* (submitted to MPS Community Meetup 2026)



Questions?

GitHub: [specificlanguages/mops](https://github.com/specifichumanities/mops)

I ❤️ MPS

- Working in the MPS ecosystem since 2015
- Focus on infrastructure (MPS tooling for builds and CI)
- Free office hours for MPS developers of all levels (also just curious)
- MPS consulting: migrations, build infrastructure
- Web: specificlanguages.com
- LinkedIn: [sergejkoscejev](https://www.linkedin.com/in/sergejkoscejev)

