

Take your jig to a second machine, and keep it up to date there

The CTO's round trip. Someone builds a jig from a CSV, pushes it to a new private repository on GitHub, installs it on their second machine with a token, finds its skill already in their agent there, changes it on the first machine, pushes, and updates the second. Nothing is written by hand: Ozwala does the git work. The run records how big the pushed repository is, and that the update keeps the jig's data. Asking the jig a question through MCP needs a model, so it is in the human run.

development · 2026-11-08

Take your jig to a second machine, and keep it up to date there

The CTO's round trip. Someone builds a jig from a CSV, pushes it to a new private repository on GitHub, installs it on their second machine with a token, finds its skill already in their agent there, changes it on the first machine, pushes, and updates the second. Nothing is written by hand: Ozwala does the git work. The run records how big the pushed repository is, and that the update keeps the jig's data. Asking the jig a question through MCP needs a model, so it is in the human run.

What you can count on

- A new jig is pushed to a new private GitHub repository, its data with it
- A second machine installs it from the private repository with a token
- Its skill is in the second machine's agent as soon as it is installed
- A change pushed from the first machine reaches the second, skill and all, and the data stays

Where it works

In the desktop app and on the command line. Each of these is checked, on every surface named, by the ozwala-usecases harness.