

SoSy-Lab

Software Systems



fm-weck

Containerized Execution of
Formal-Methods Tools

Dirk Beyer
> Henrik Wachowitz

10.10.2024 – ConVeY, Ettal

HOW DO YOU RUN AN FFM TOOL?

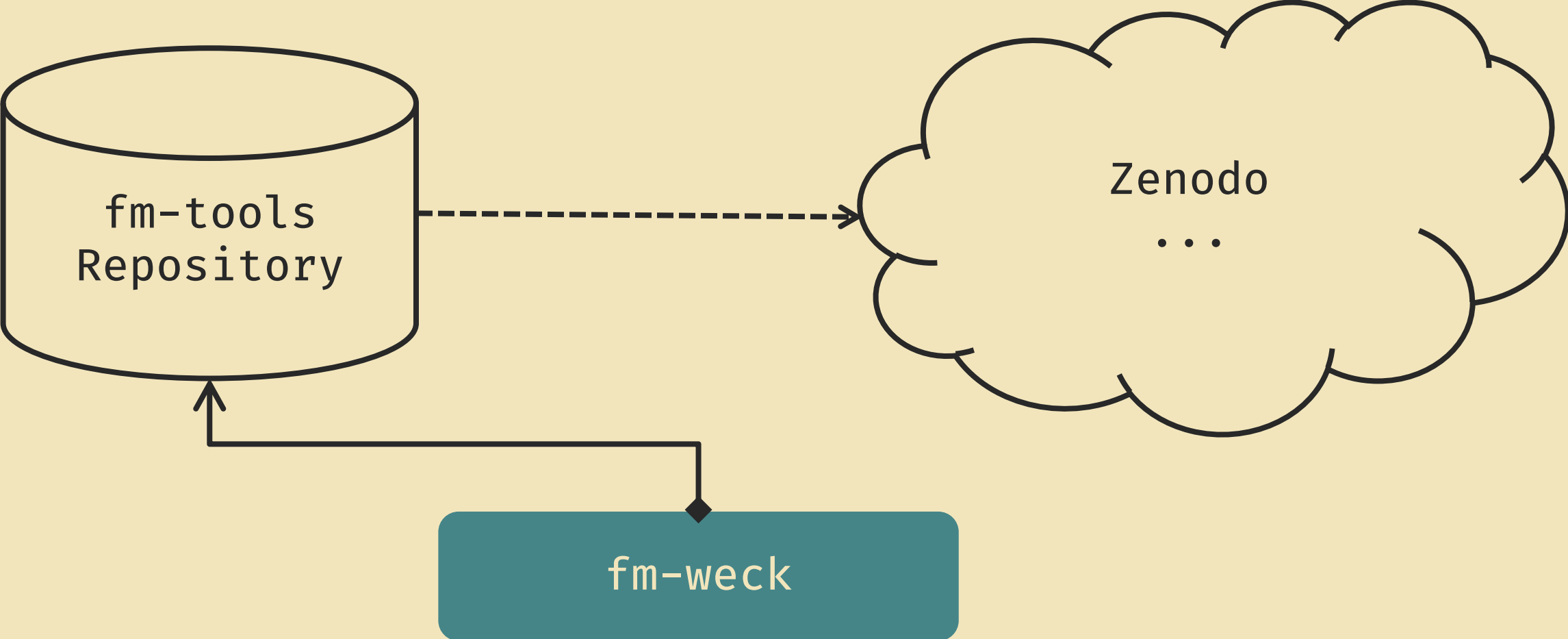
How do others solve it?

```
npx create-react-app my-app
```

```
pipx run pycowsay
```

```
fm-weck run cpachecker
```

Data Source











D

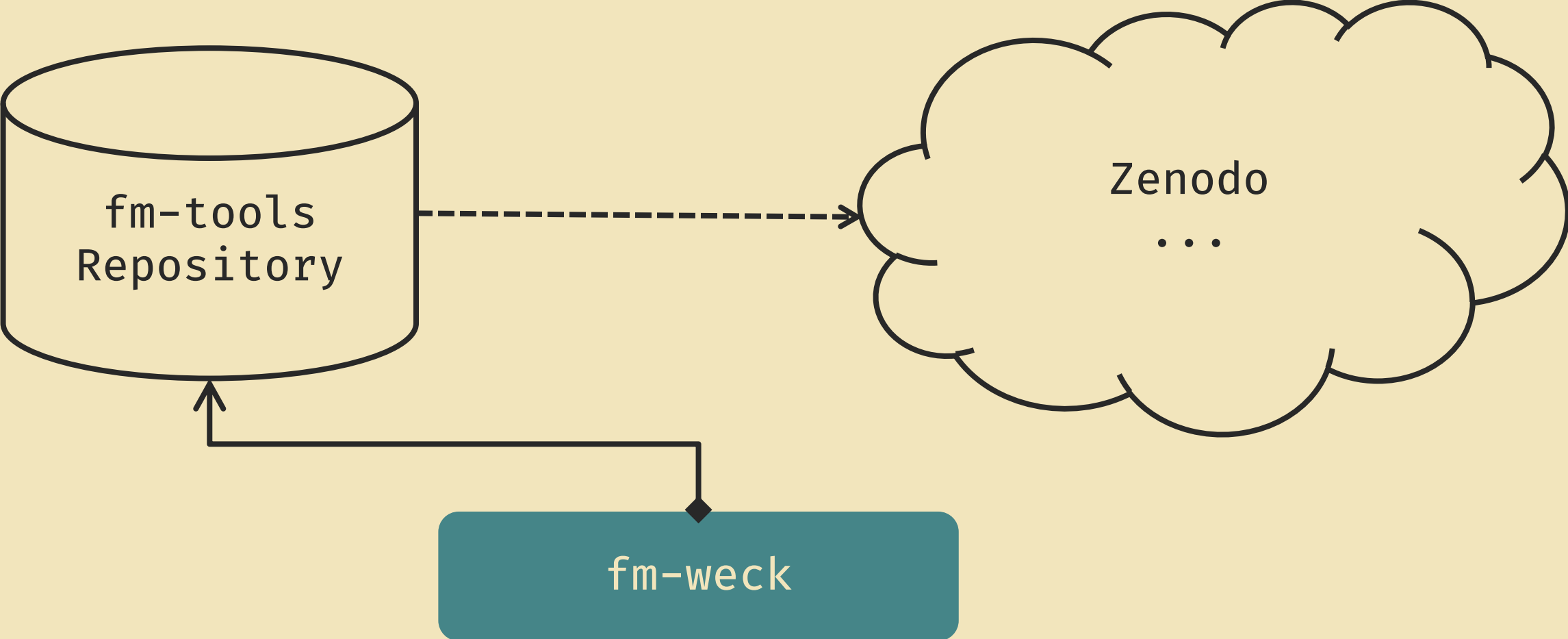
Dirk Beyer authored 1 week ago

Code owners Assign users and groups as approvers for specific file changes. [Learn more.](#)

Manage branch rules

Name	Last commit	Last update
..		
 2ls.yml	Use URLs using the correct tag	1 month ago
 aise.yml	Add frameworks/solvers to the tools and schema	3 months ago
 aprove.yml	Set the correct GitLab user for the maintainer; add the images for co...	3 months ago
 brick.yml	Use URLs using the correct tag	1 month ago
 bubaak-split.yml	Add used_actors key to a few more tools	1 week ago
 bubaak.yml	Use URLs using the correct tag	1 month ago
 cbmc.yml	Add used_actors key to a few more tools	1 week ago
 cetfuzz.yml	Add frameworks/solvers to the tools and schema	3 months ago

Data Source



fm-tool YAML

```
name: CPAChecker
input_languages:
  - C
project_url: https://cpachecker.sosy-lab.org
repository_url: https://gitlab.com/sosy-lab/software/cpachecker
spdx_license_identifier:
  Apache-2.0
benchexec_toolinfo_module:
  benchexec.tools.cpachecker
fmtools_format_version: "2.0"
fmtools_entry_maintainers:
  - dbeyer
  - ricffb
```

name: CPAchecker

fm-tool YAMI

input

- C

project

repository

spdx_

Apache

benchmark

benchmark

fmtool

fmtool

- db

- ri

```
class Tool(benchexec.tools.template.BaseTool2):  
  
    def executable(self, tool_locator):  
        return tool_locator.find_executable("cbmc")  
  
    def version(self, executable):  
        return self._version_from_tool(executable)  
  
    def name(self):  
        return "CBMC"  
  
    def project_url(self):  
        return "http://www.cprover.org/cbmc/"  
  
    def cmdline(self, executable, options, task, rlimits):  
        ...
```


fm-tool YAML

name: CPAChecker

input_languages:

- C

project_url: <https://cpachecker.sosy-lab.org>

repository_url: <https://gitlab.com/sosy-lab/software/cpachecker>

Anatomy of a version

spdx_license_identifier:

Apache-2.0

benchexec_toolinfo_module:

benchexec.tools.cpachecker

fmtools_format_version: "2.0"

fmtools_entry_maintainers:

- dbeyer

- ricffb

versions:

- version: "svcomp24"

doi: 10.5281/zenodo.10203297

benchexec_toolinfo_options:

["-svcomp24", "-heap", "10000M", "-benchmark",
"-timelimit", "900 s"]

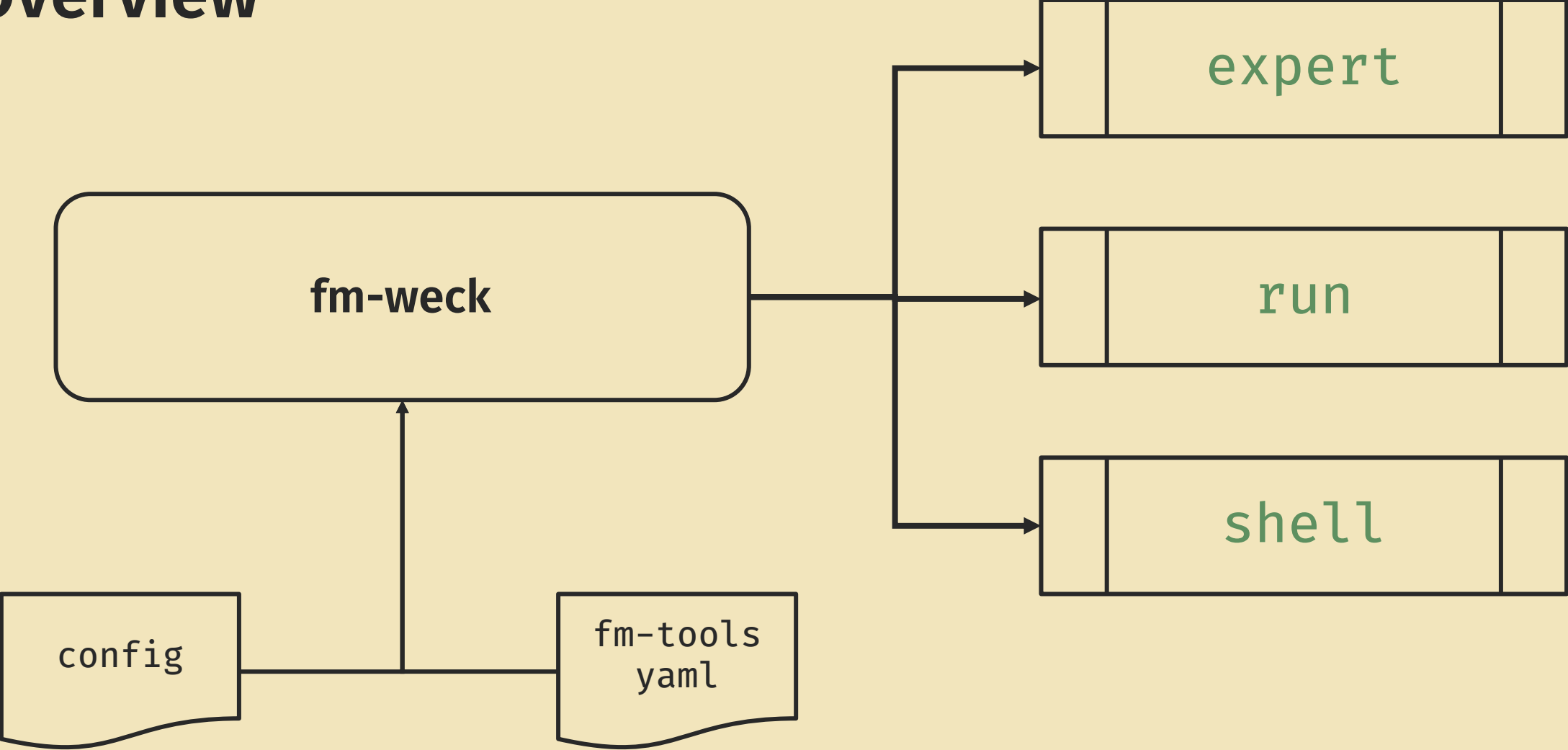
container_base_image: "ubuntu:22.04"

container_full_image: "registry.gitlab.com/sosy-lab/benchmarking/competition-scripts/user:latest"

required_ubuntu_packages:

- openjdk-17-jdk-headless

Overview



Modes

Tool Name & Version

```
fm-weck run cpcachecker:2.3.1 \  
-p unreachable-call \  
prog.c
```

Common Property by name



Modes



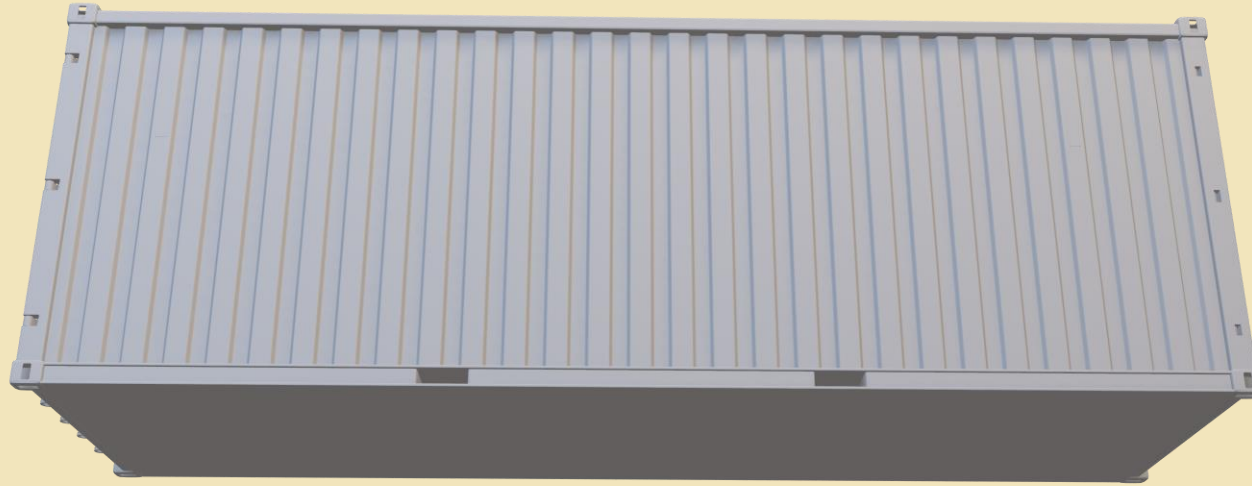
Tool Name & Version



```
fm-weck expert cpatchecker:2.3.1 \  
--witness witness.yml  
--spec unreachable.prog.c
```

Verbatim Arguments
to Tool

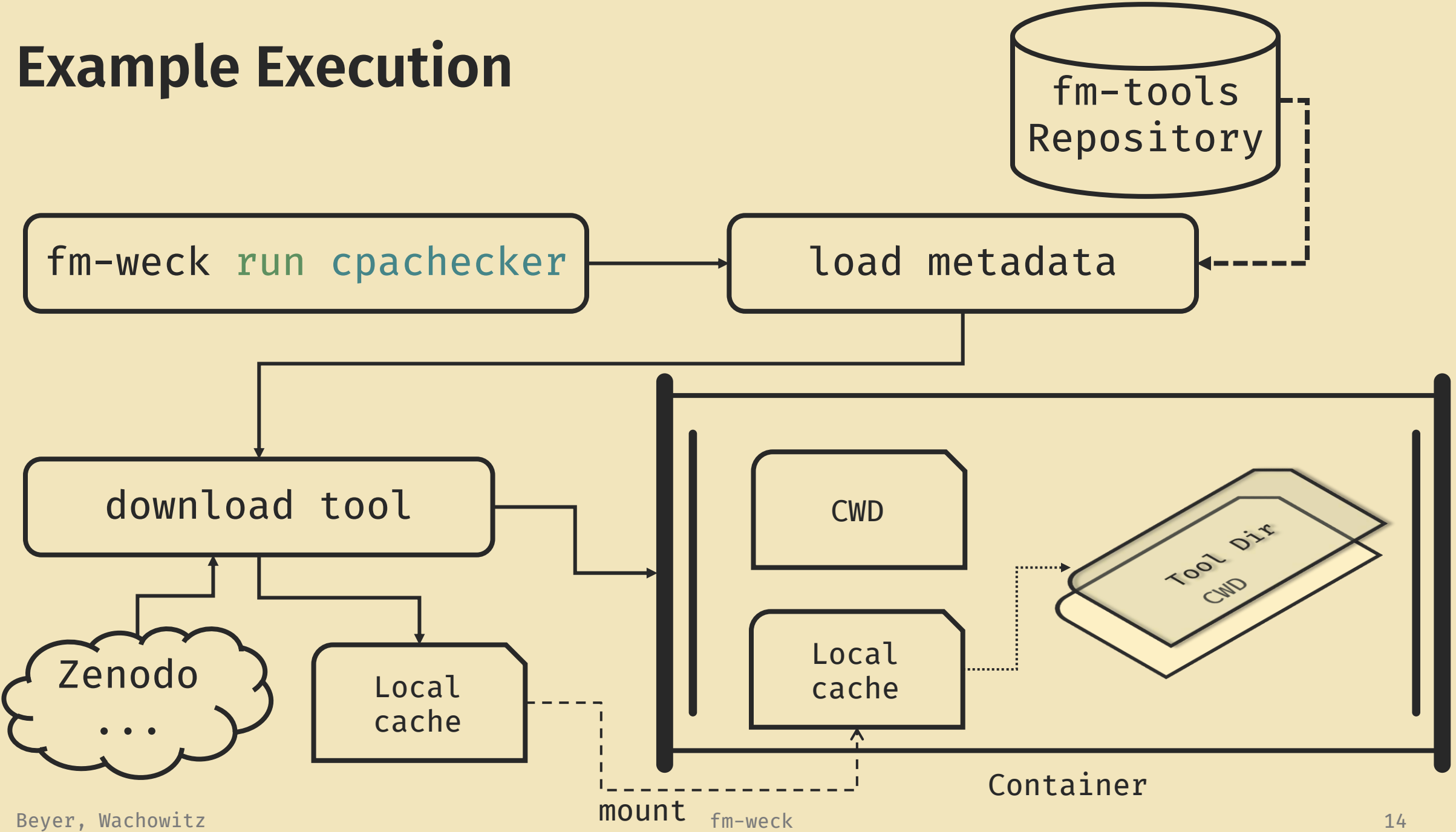
Modes



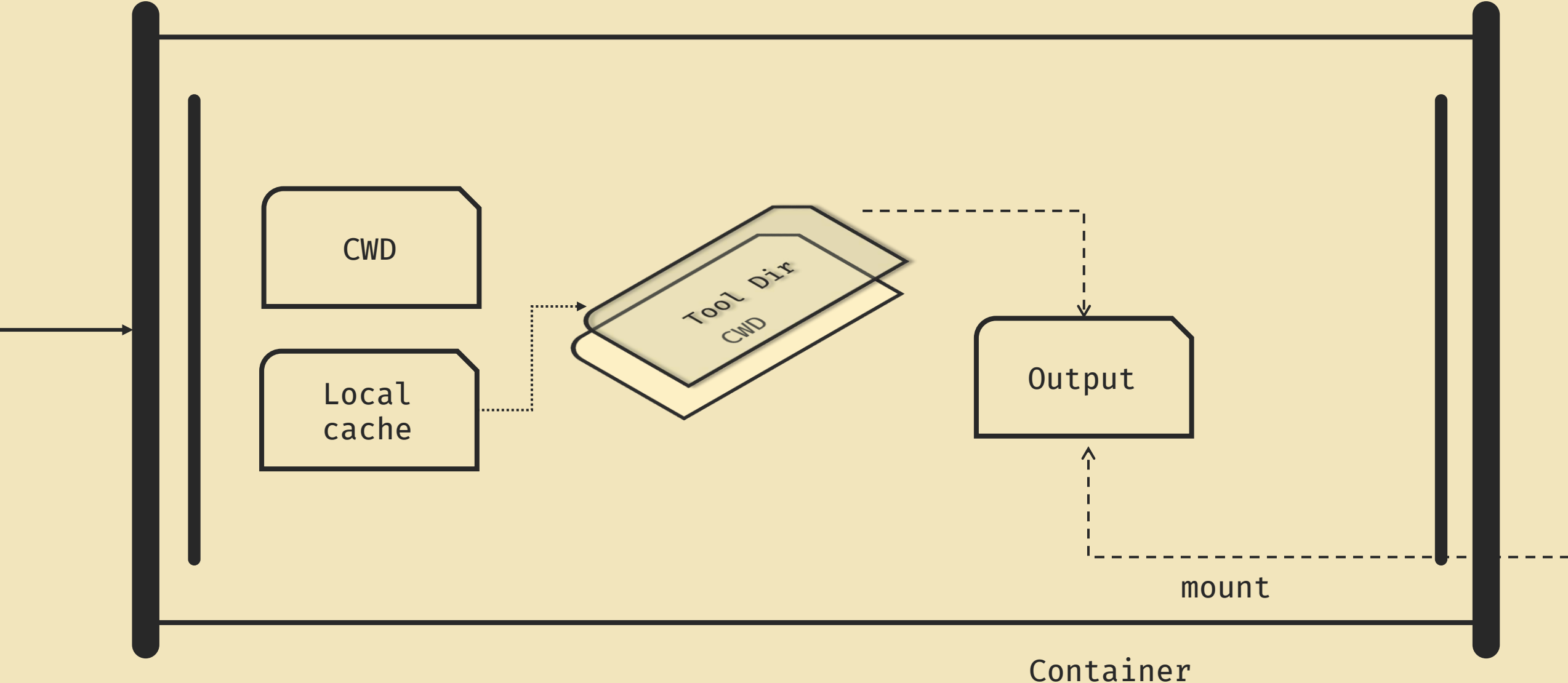
```
fm-weck shell cpatchecker:2.3.1
```

Launch interactive shell

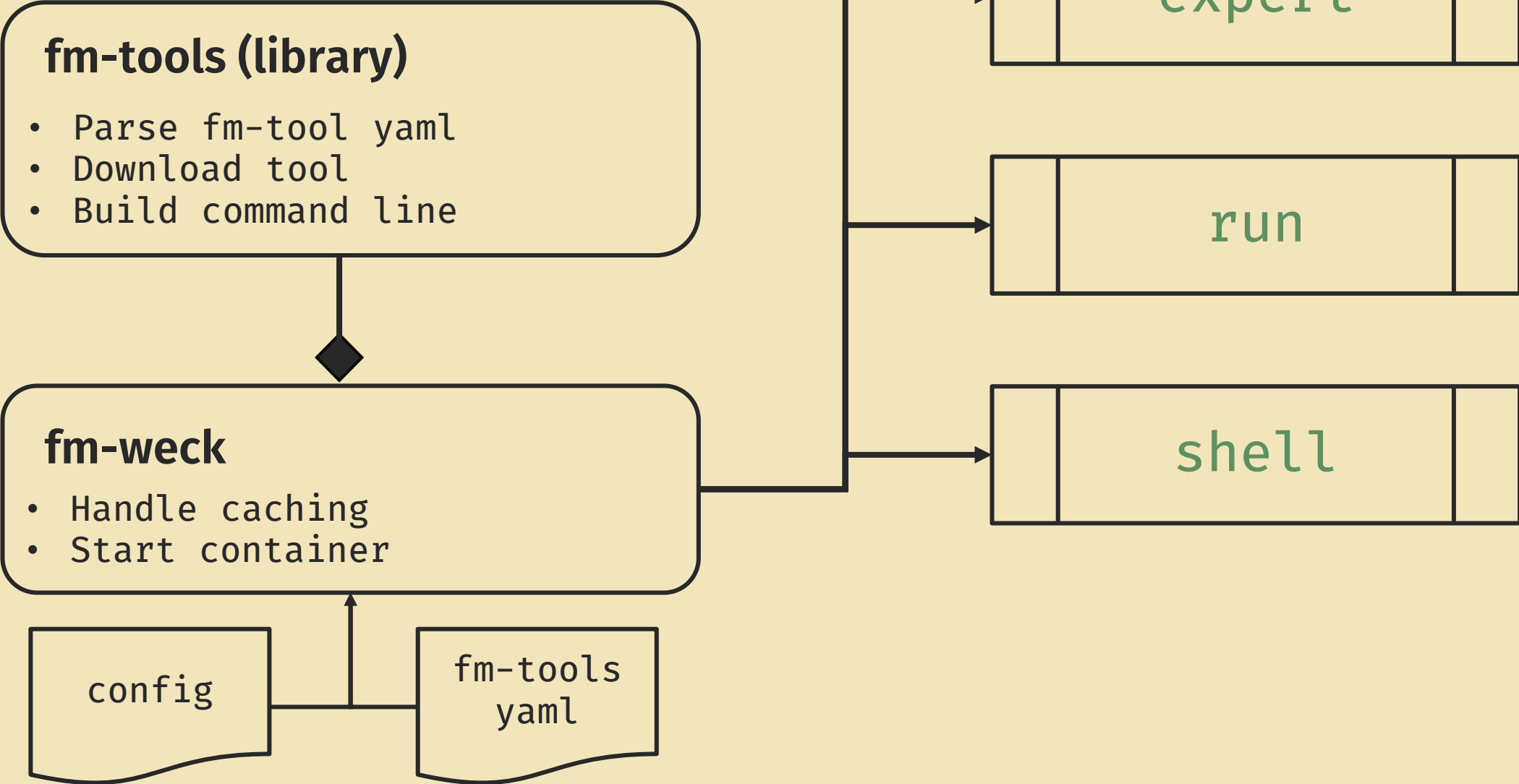
Example Execution



Example Execution

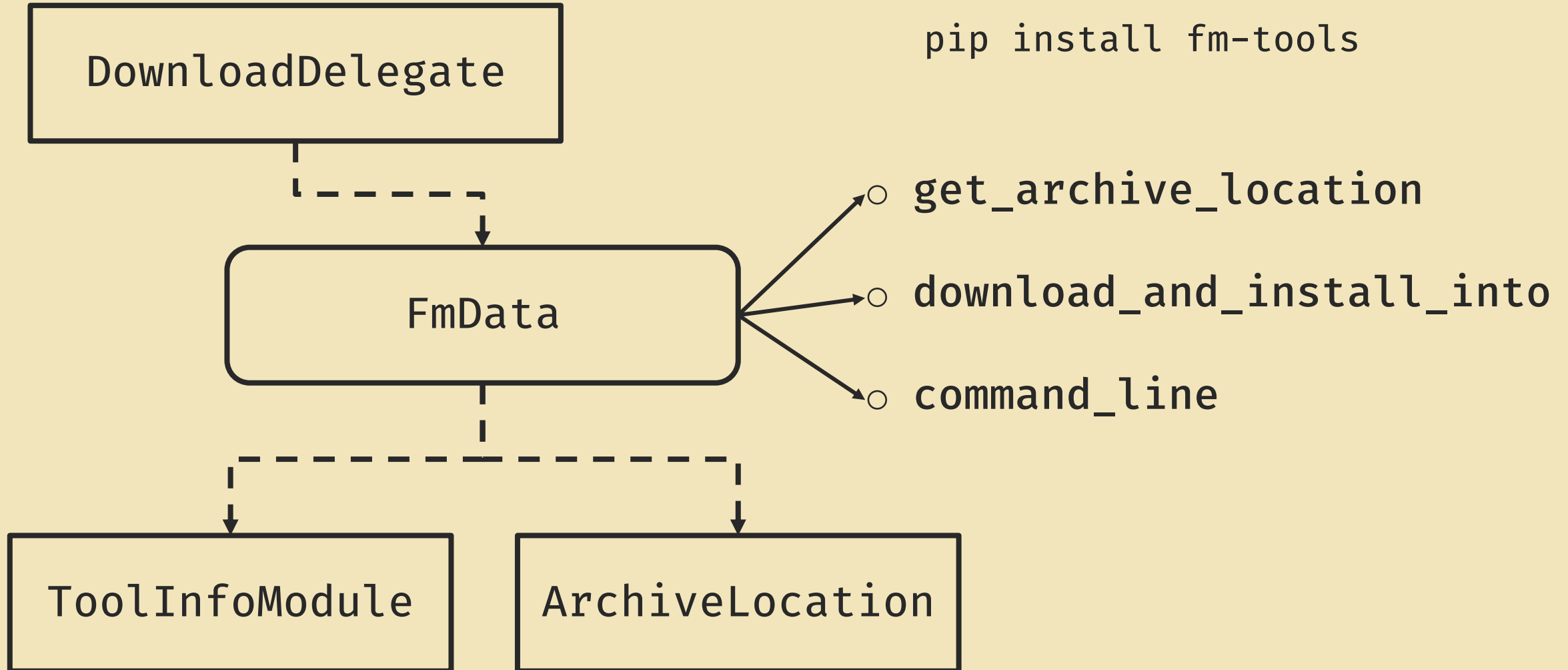


Architecture

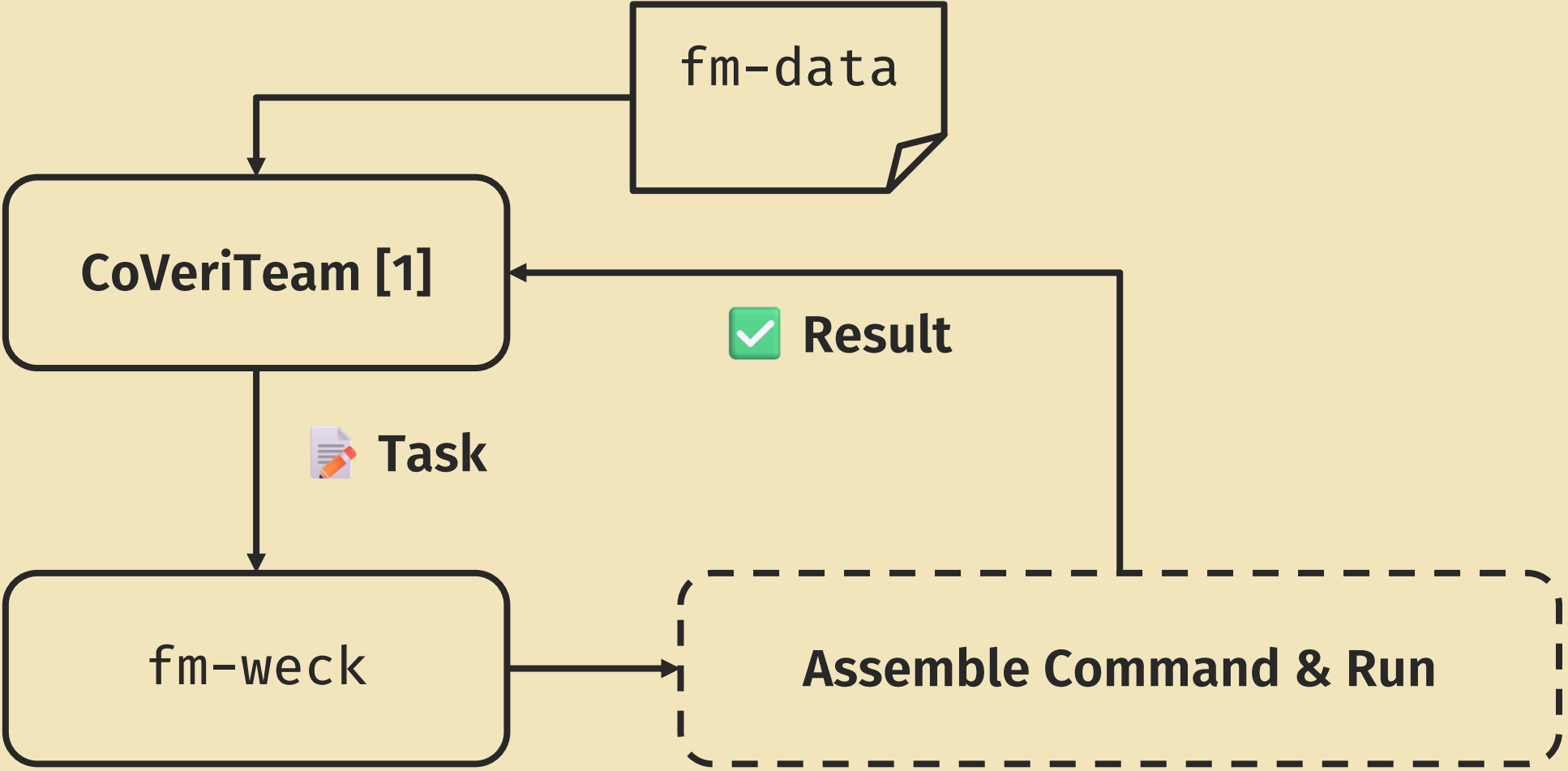


Advertisement: **fm-tools**

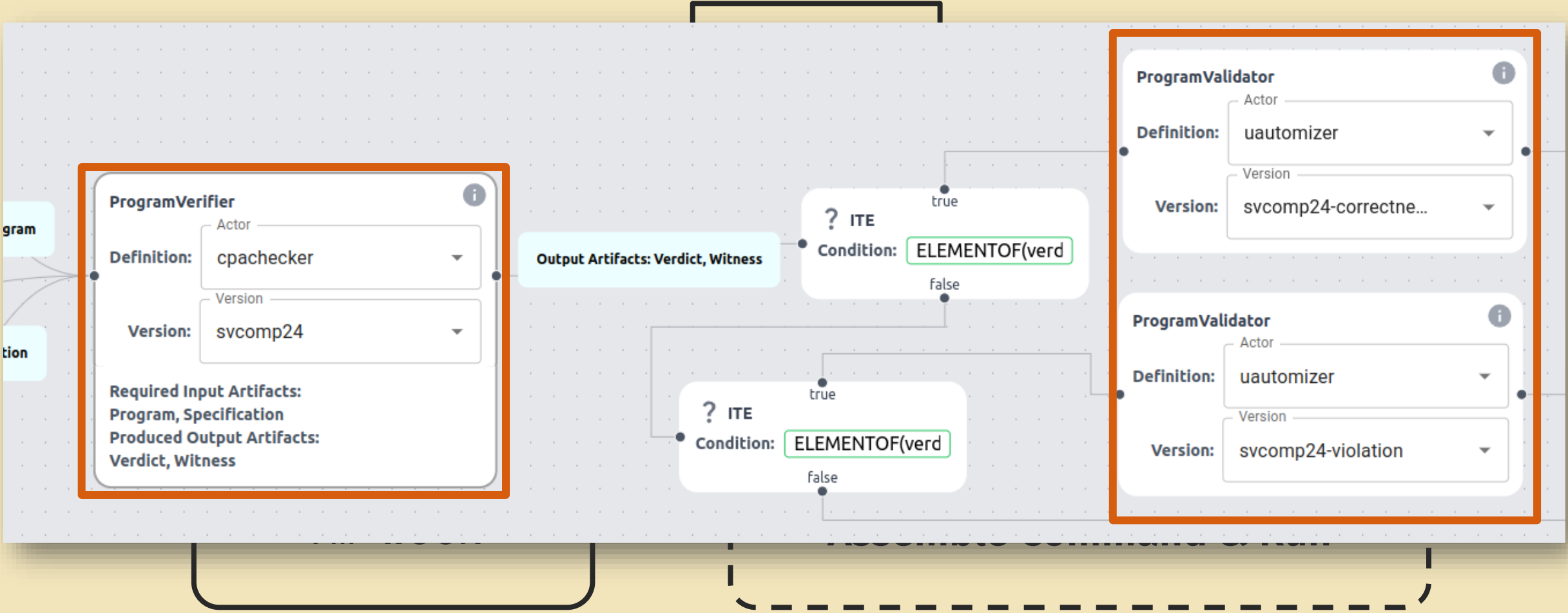
```
pip install fm-tools
```



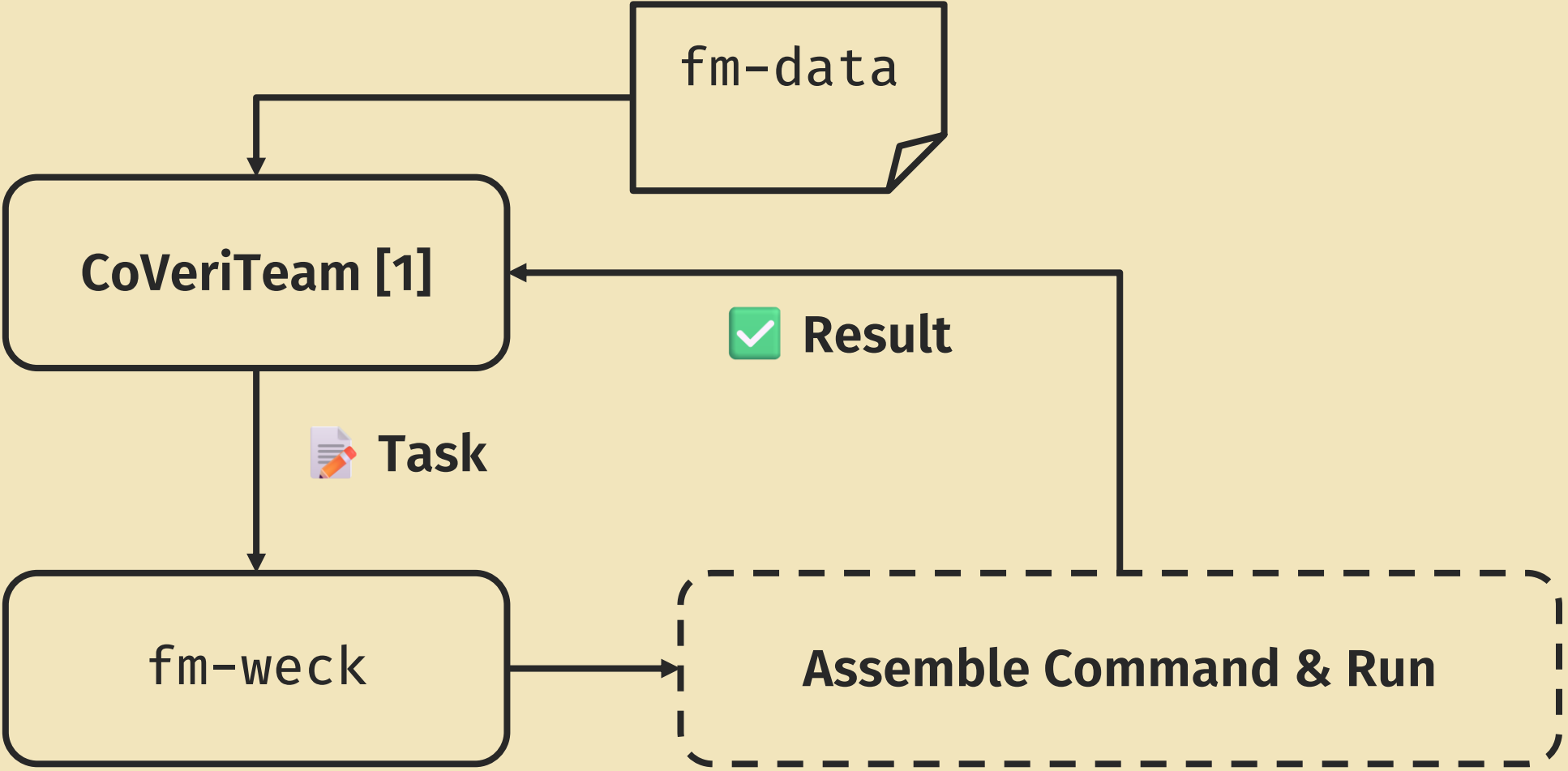
Using fm-weck in CoVeriTeam



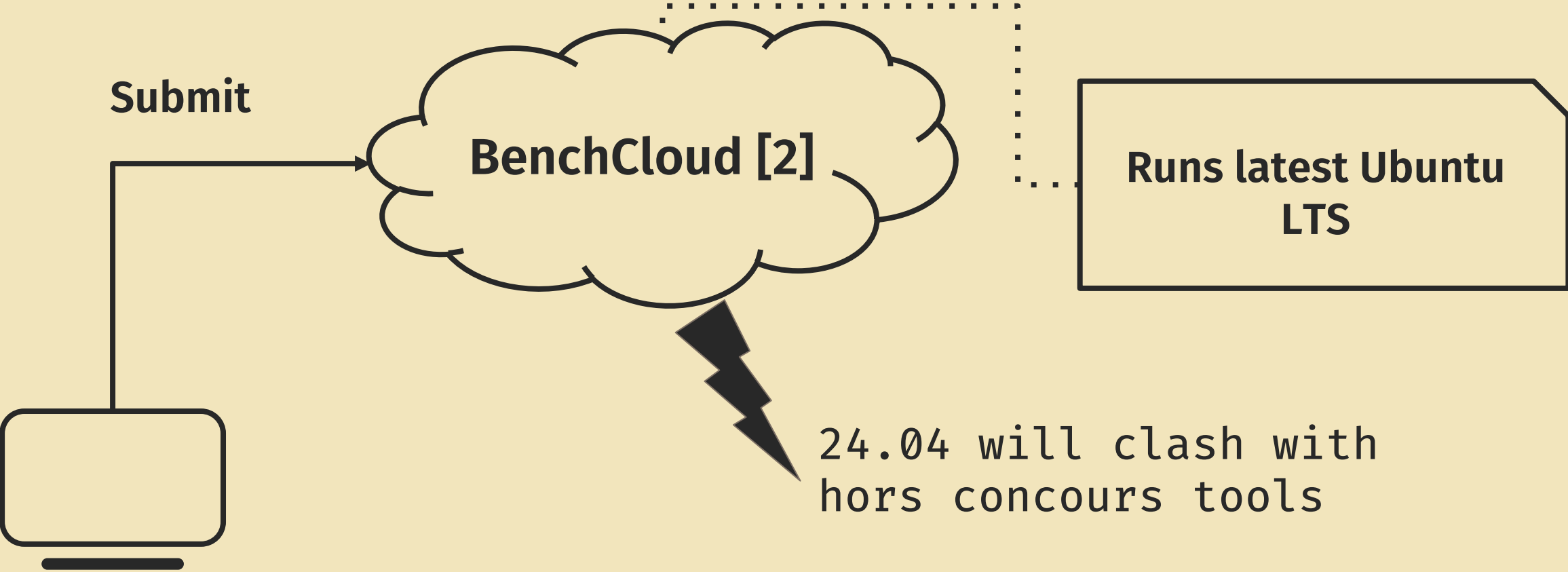
Using fm-weck in CoVeriTeam



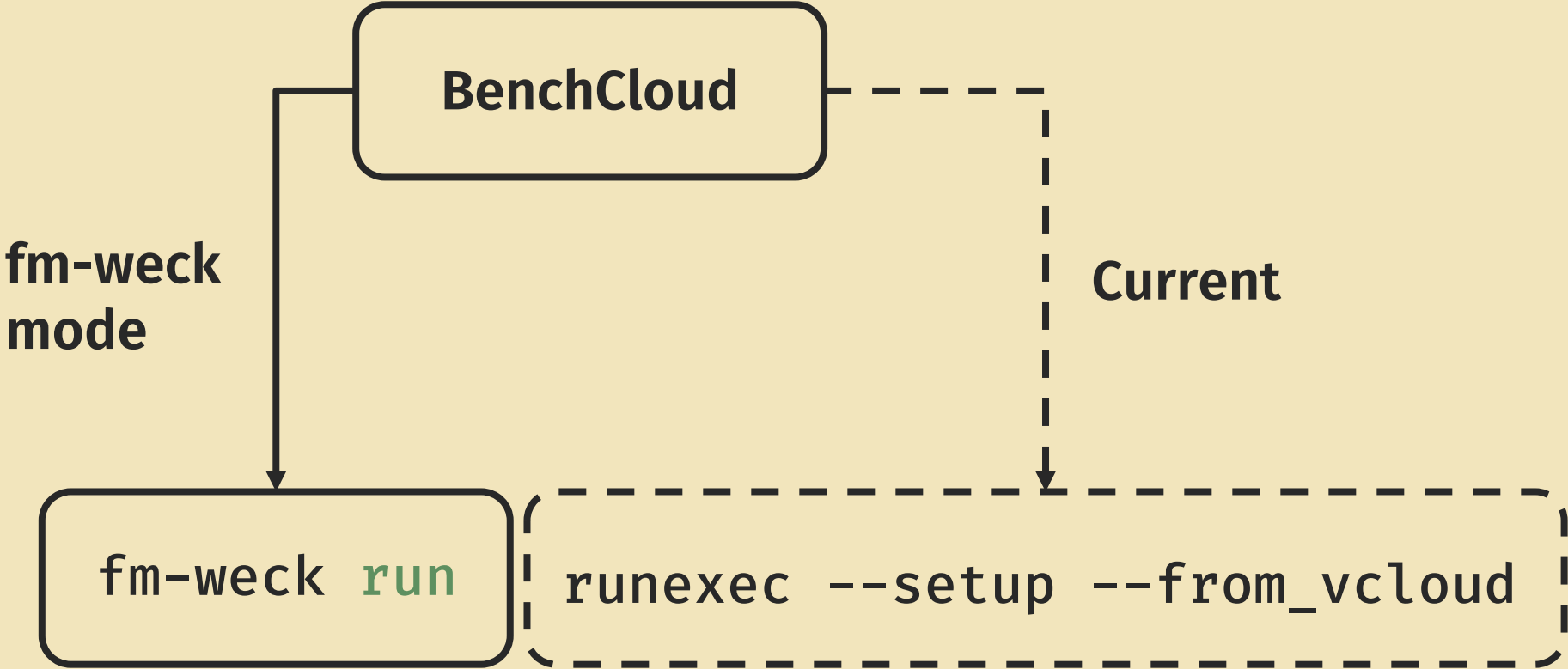
Using fm-weck in CoVeriTeam



SV-COMP 25: Support for Hors Concours



SV-COMP 25: Extending the Infrastructure



🌟 on gitlab.com/sosy-lab/software/fm-weck



fm-weck



fm-weck **expert**

- Download
- Run in Container
- fm-tool cli
processed
verbatim

fm-weck **run**

- Download
- Run in Container
- fm-tool cli
automatically
assembled

fm-weck **shell**

- Launch Container
- develop with
fm-tools

References

- [1] Beyer, D., Kanav, S. (2022). CoVeriTeam: On-Demand Composition of Cooperative Verification Systems. Proc. TACAS 2022.
https://doi.org/10.1007/978-3-030-99524-9_31
- [2] Beyer, D., Chien, P.-C., Jankola, M., (2024). BenchCloud: A Platform for Scalable Performance Benchmarking. To be published in Proc. ASE 2024.