Máster en Sistemas Electrónicos Avanzados (MSEA)

Co-simulación y verificación funcional con VHDL, C/C++ y Python/m

{sim}

Unai Martinez Corral

unai.martinezcorral@ehu.eus umarcor umarcor

Escuela de Ingeniería de Bilbao
Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU)

2021/05

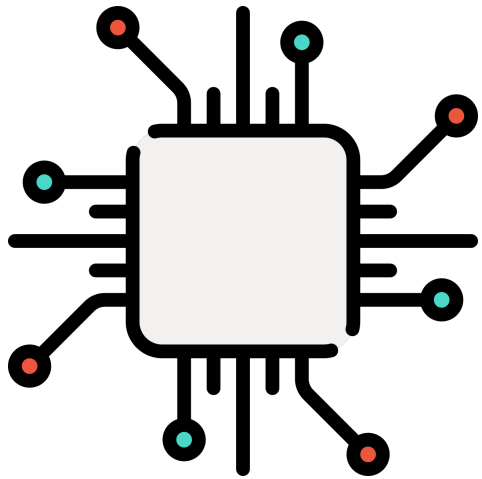
Work environment

- GHDL (LLVM or GCC backends)
- Python $>= 3.6$
- Editor: Visual Studio Code (VSC), vim, emacs, Sigasi...
- GtkWave
- Language server: ghdl-ls, rust_hdl...

HDL organisation in GitHub

A joint effort from communities and companies such as GHDL, VUnit, PLC2, Symbiflow, Antmicro, Google, Chips4Makers, etc. for gathering references, resources and pre-built/packaged solutions related to (open source) EDA tooling.

- hdl/awesome
- hdl/packages
  - hdl/MINGW-packages
  - hdl/containers
  - Conda, Bazel, Termux, etc.
- hdl/smoke-tests
- hdl/constraints



Installation: on Windows (MSYS2)

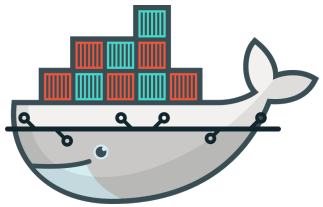
hdl.github.io/MINGW-packages/#_usage

1. Download the latest self-extracting installer or the tarball from repo.msys2.org or msys2/msys2-installer.
2. Extract/install MSYS2 and follow the guidelines in msys2.org/#installation for the initial sync/update.
3. On a MINGW64 shell, install the dependencies and the tools:

```
$ pacman -Syu --noconfirm
$ pacman -S --noconfirm p7zip git \
    mingw-w64-x86_64-yosys \
    mingw-w64-x86_64-gtkwave \
    mingw-w64-x86_64-python-pip

$ git clone --recurse-submodules https://github.com/VUnit/vunit
$ cd vunit
$ python setup.py install
```

Installation: using containers (locally)



hdl.github.io/containers/#_usage

- Engine:
  - Docker
  - Podman
- X server:
  - x11docker
  - runx
- VSCode Remote Extensions
  - Containers
  - Windows Subsystem for Linux (WSL)

Installation: using online services/IDEs

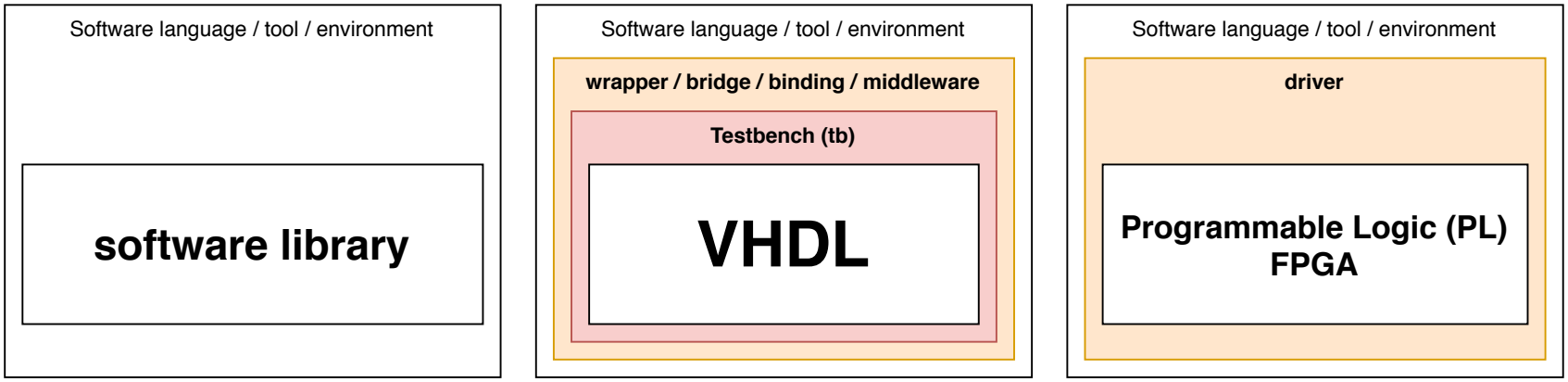
- Gitpod
  - gitpod.io/#https://github.com/umarcor/msea
  - gitpod.io/#https://github.com/cocotb/cocotb
- Play with Docker (PWD)
- EDA Playground

Exercises: introduction

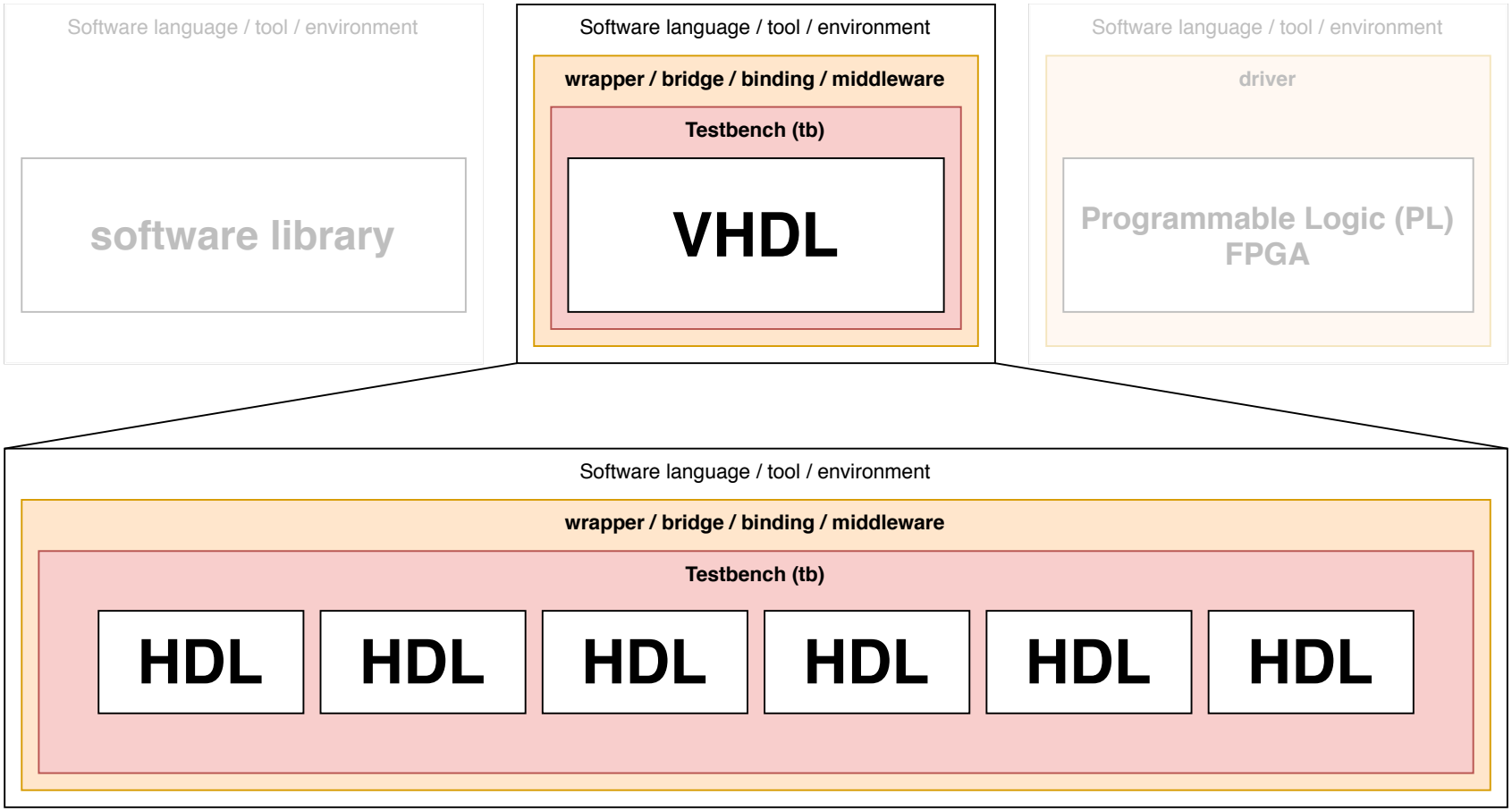
GHDL Quick Start Guide: Simulation

- Hello World
- Heartbeat
- Full-adder

Functional verification of an HDL design



Functional verification of a non-trivial HDL design

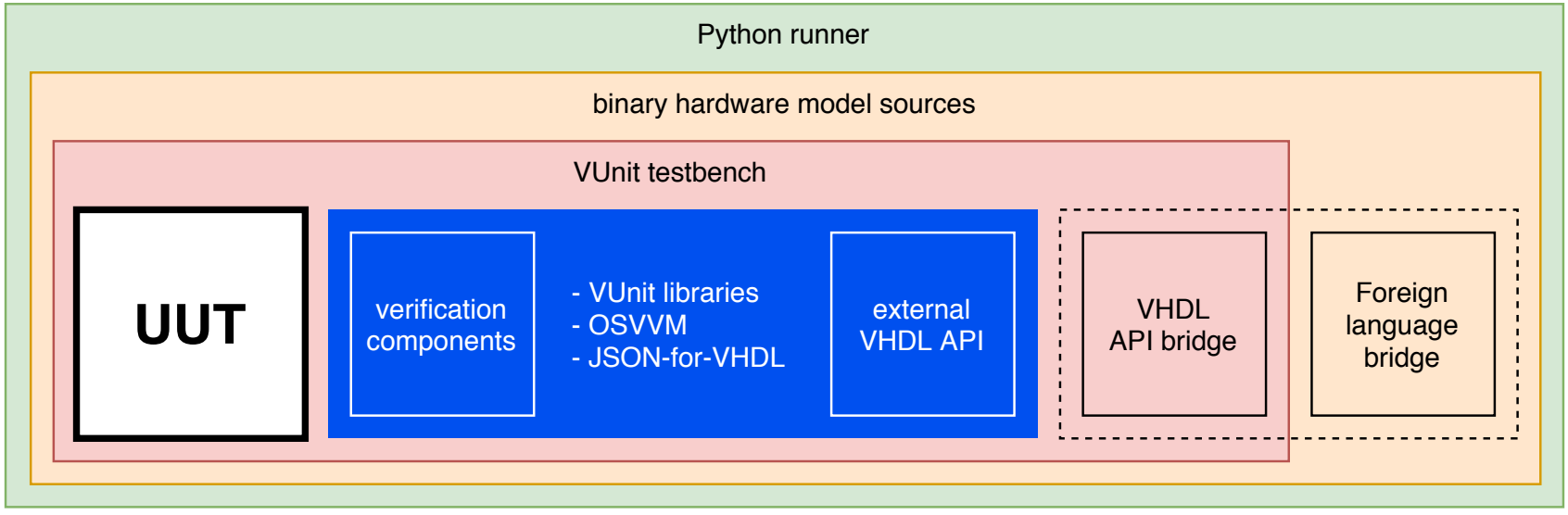


HDL simulation: VUnit

Open source unit testing framework for VHDL/SystemVerilog. It features the functionality needed to realize continuous and automated testing of your HDL code. VUnit complements traditional testing methodologies by supporting a *test early and often* approach through automation.

- Supported languages: VHDL (93, 2002, 2008, 2019), Verilog, SystemVerilog
- Supported simulators: GHDL, Aldec Riviera-PRO/Active-HDL, Mentor Graphics ModelSim/Questa and Cadence Incisive (experimental)
- Requires Python $>=$ 3.6: Python Interface and CLI
- Supported on Windows, GNU/Linux and macOS.
- VHDL libraries: OSVVM, JSON-for-VHDL, Run, Check, Logging, Communication, Verification Components, etc.
- Data Types with an external API for co-simulation

VUnit: overview

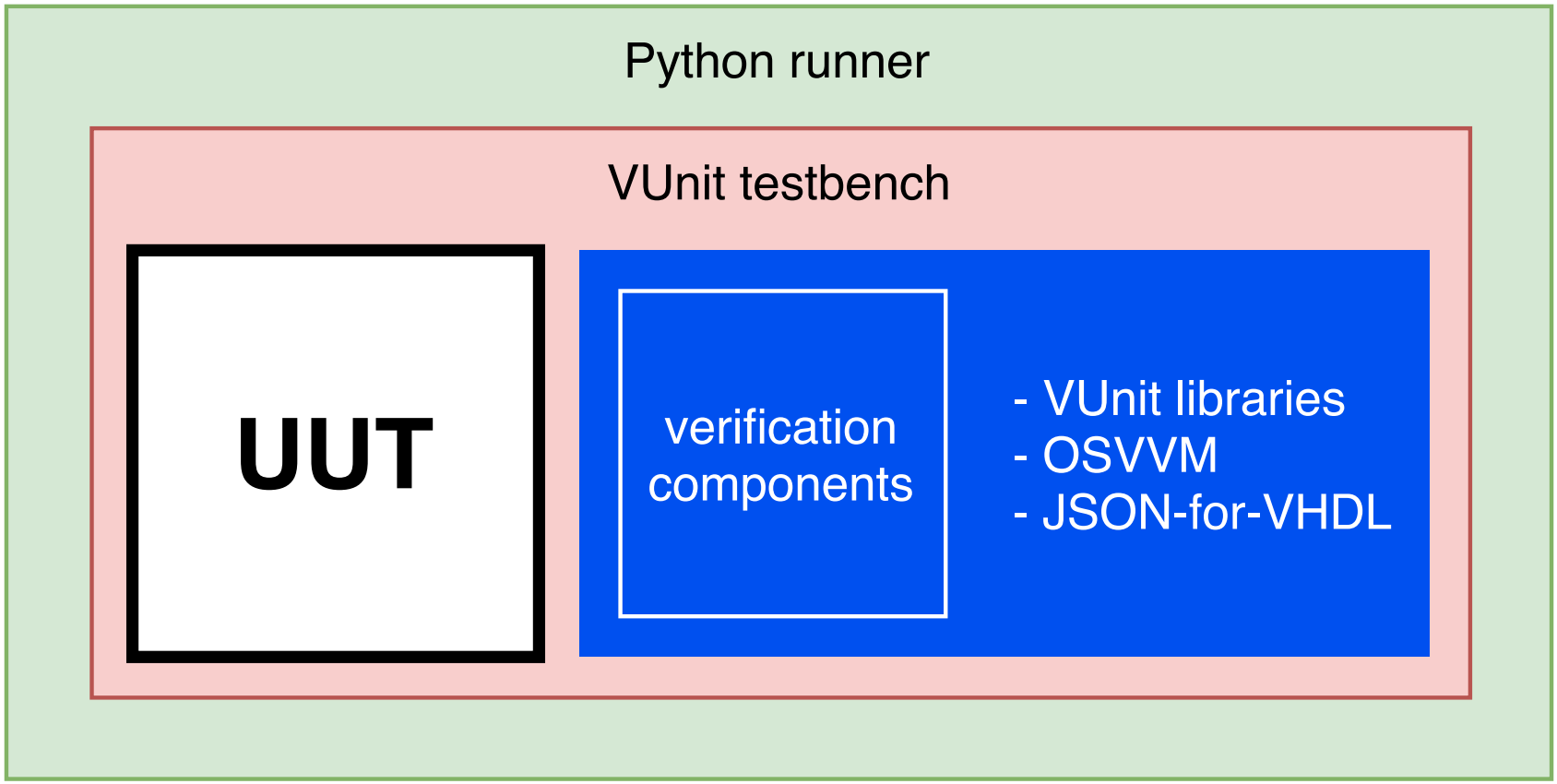


Exercises: introduction

VUnit User Guide 

- Add `run.py` to Full-adder

VUnit: tutorial



Exercises: libraries

- Run
- Logging
- Check
- Communication
- OSVVM
  - array
- JSON-for-VHDL
  - json4vhdl
  - composite_generics

Exercises: verification components

Verification Component Library (VCL)

- uart </>
- array_axis_vcs </>
- axi_dma </>

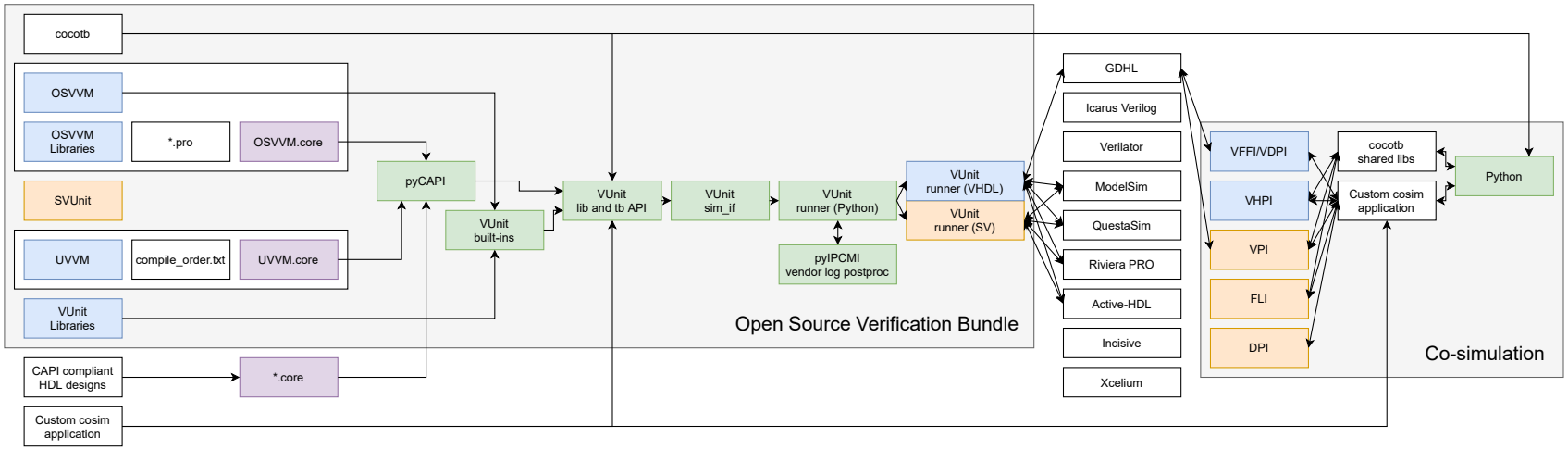
VHDL simulation: libraries/frameworks for verification

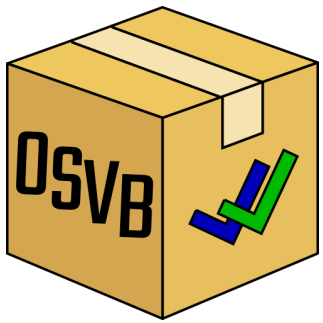
- UVM: Universal Verification Methodology W
- OSVVM: Open Source VHDL Verification Methodolody
- UVVM: Universal VHDL Verification Methodology

- cocotb: Coroutine Co-simulation Test Bench </>
- VUnit: unit testing framework </>
  - VUnit/cosim

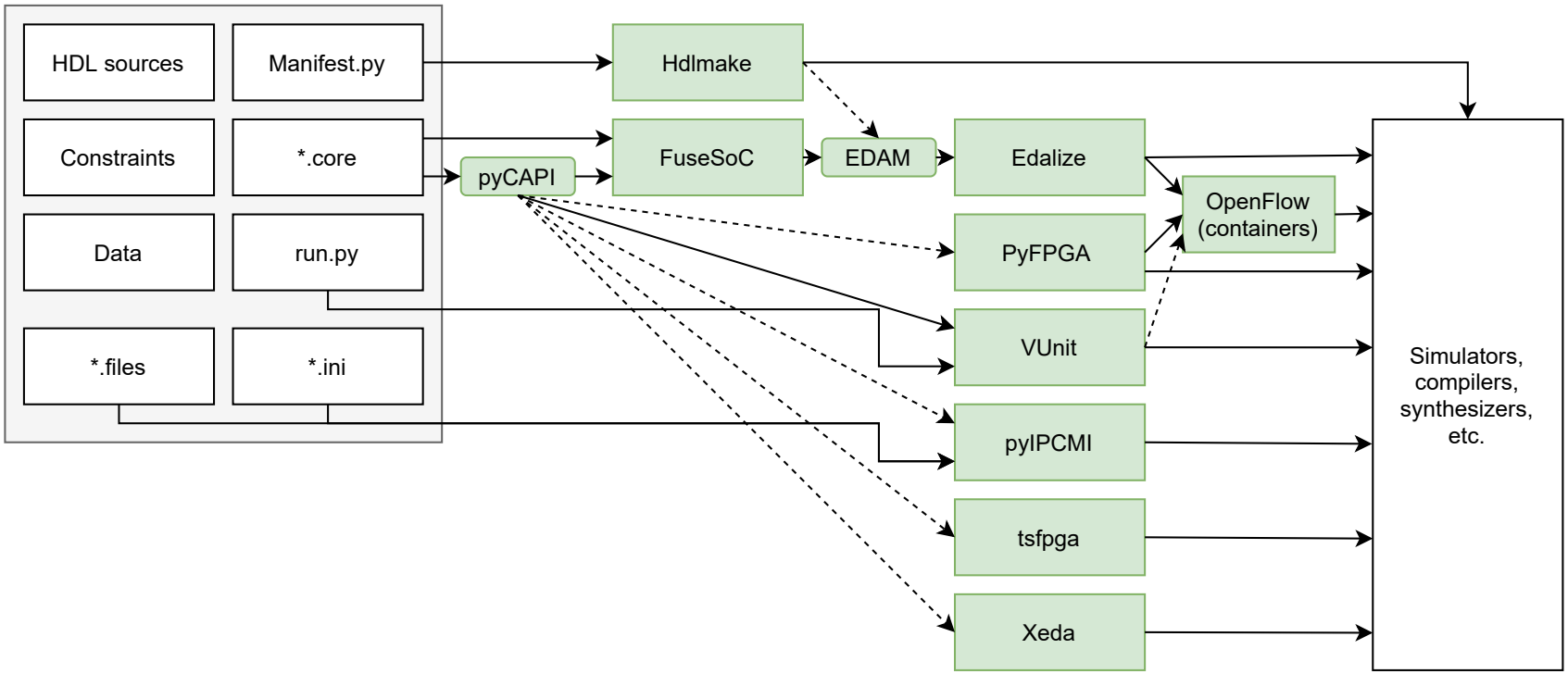
Open Source Verification Bundle (OSVB)

umarcor.github.io/osvb



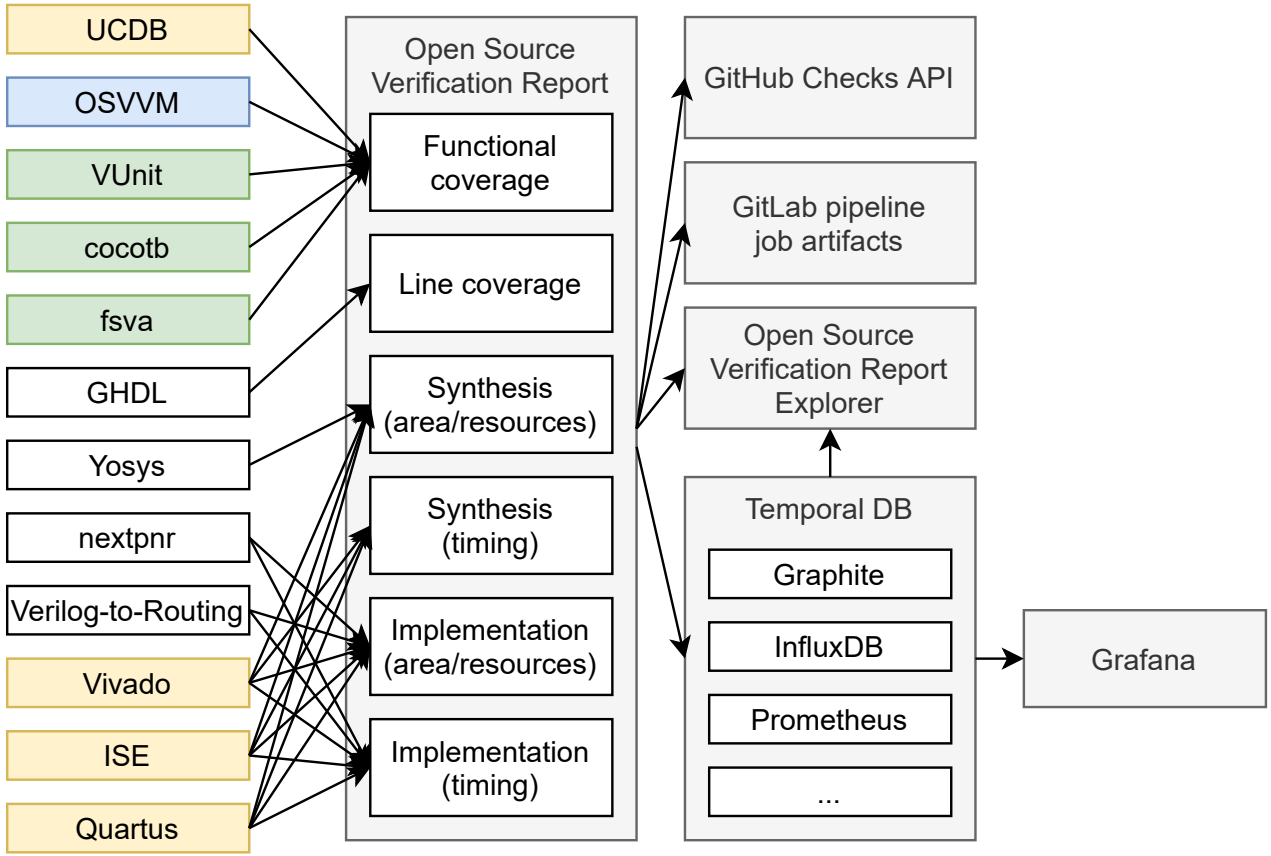


OSVB: pyCAPI



umarcor.github.io/osvb/apis/core

OSVB: Open Source Verification Report (OSVR)



umarcor.github.io/osvb/apis/logging