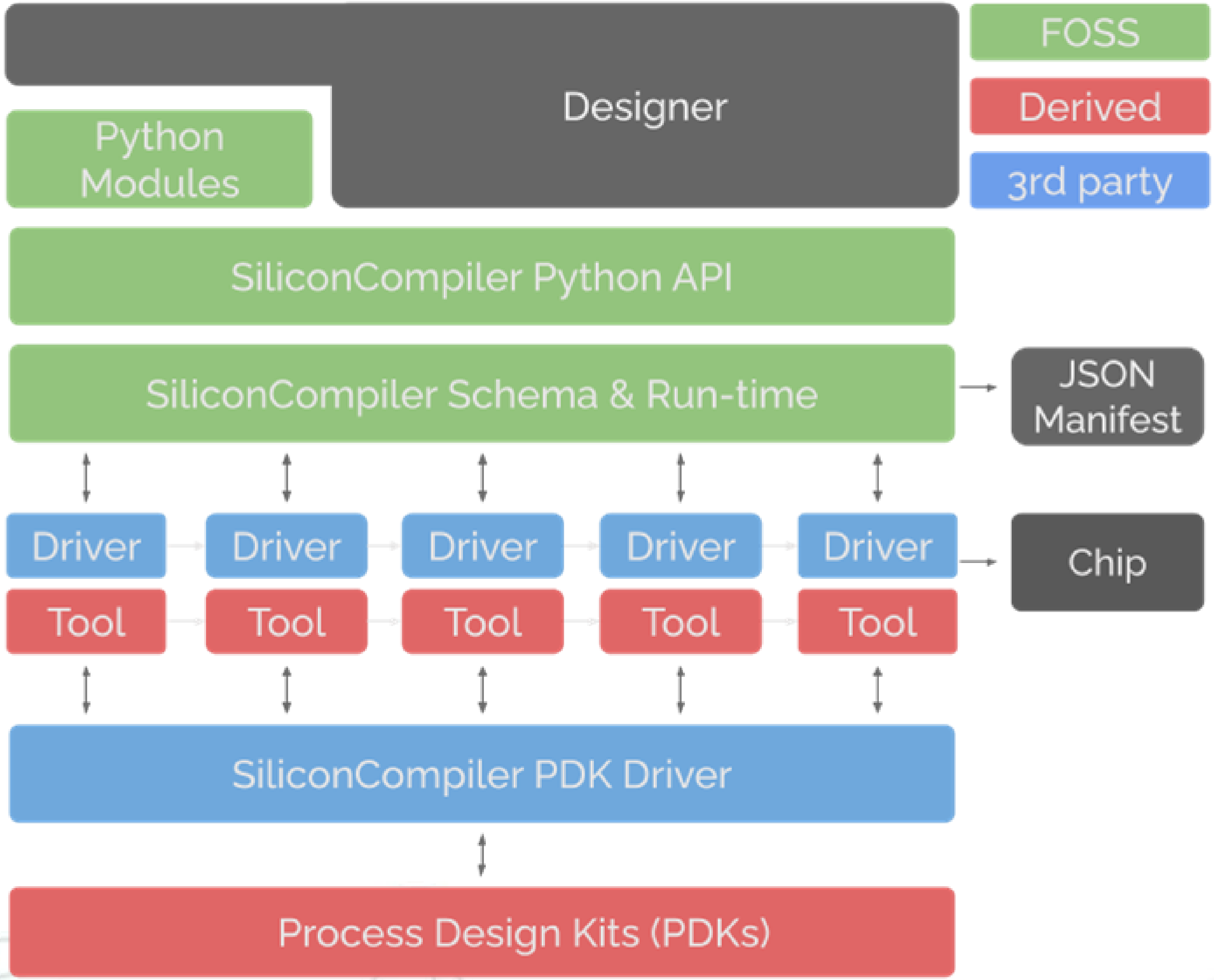
59 DESIGN AUTOMATION CONFERENCE

SiliconCompiler: “Make for HW”

Andreas Olofsson, William Ransohoff, Noah Moroze
Zero ASIC Corporation
Cambridge, MA
{andreas,will,noah}@zeroasic.com

SiliconCompiler

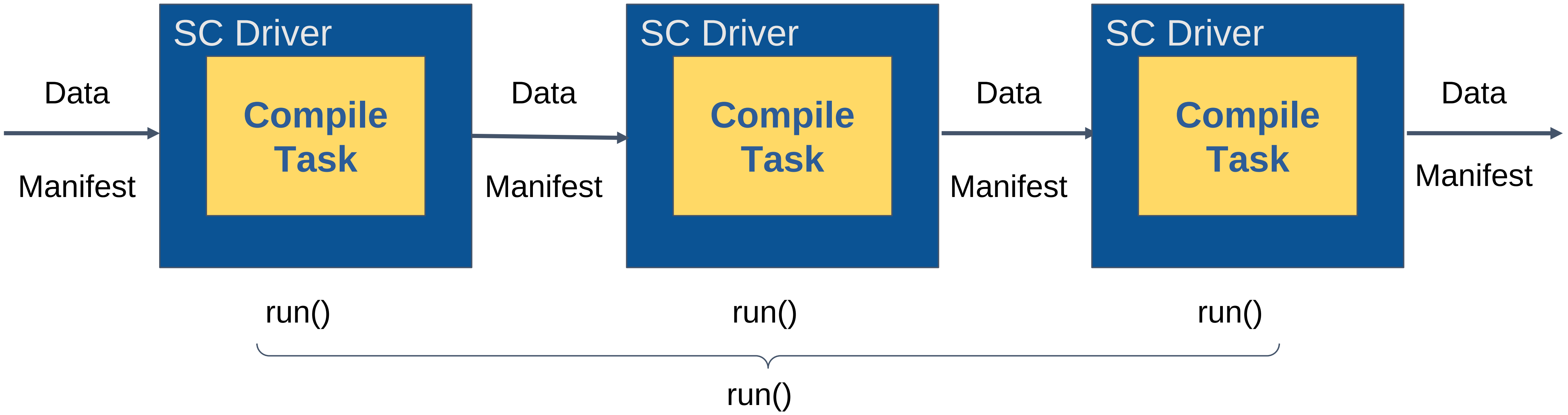
SiliconCompiler: *A modular build system for hardware*



Designer
FOSS
Derived
3rd party
Python Modules
SiliconCompiler Python API
SiliconCompiler Schema & Run-time
JSON Manifest
Driver
Driver
Driver
Driver
Driver
Tool
Tool
Tool
Tool
Tool
Chip
SiliconCompiler PDK Driver
Process Design Kits (PDKs)

- “Make for hardware”
- Standardized build schema (json)
- Python OO API
- Flowgraph based execution
- Developed with cloud first approach
- Automated actions/metrics tracking
- Built for commercial AND open source ASIC/FPGA tools.
- https://github.com/siliconcompiler

Basic Operation: *Configure, Run, Observe*

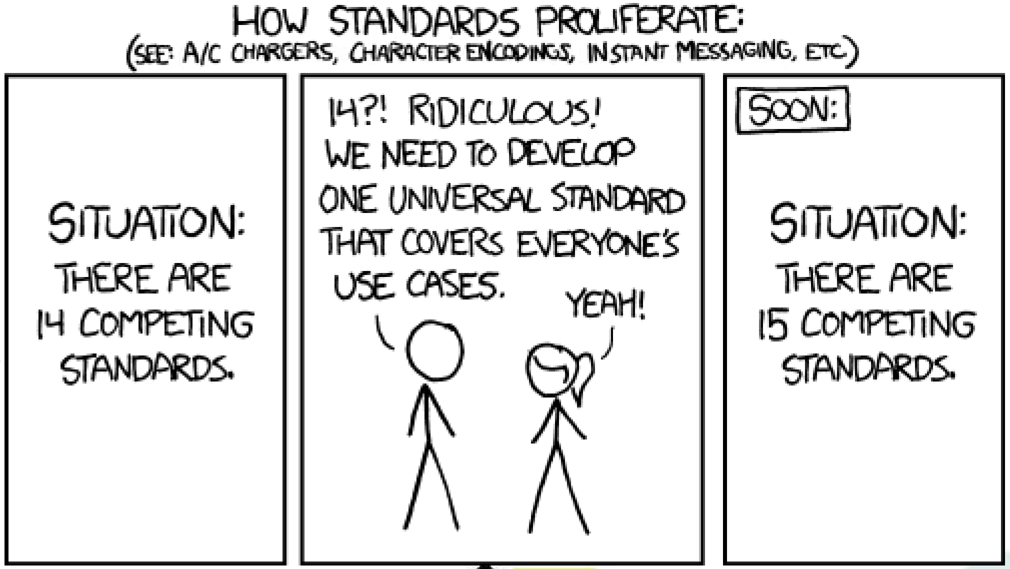


1. All compilation tasks (ie EDA tools) are wrapped with SiliconCompiler (SC) interfaces to enable configuration and results tracking in a unified JSON manifest.
2. The manifest is the golden database that defines the “what, how, when, who, why” of compilation.
3. A static flowgraph defines the sequence of tasks to be executed by the atomic run() function.
4. After running, all results and metrics accessible through the manifest.

The Manifest: *An Open “CAD Standard”*

Group	Parameters	Examples
asic	46	diearea , maxfanout, cell lists, delay model…
intput/output	2	sdc, rtl, def, …
constraint	8	PVT, SDC, checks, …
options	50	loglevel , skip, optmode, path, …
unit	10	Time, voltage, current, ...
pdk	50	Runset, stackup, process,...
tool	29	Options, exename, license,...
flowgraph	9	Inputs, weights, goals
checklist	9	Rationale, criteria,
metric	45	Setupwns, errors, warnings, …
datasheet	39	Abs voltage, setup, hold
package	32	Dependency, author, …

- A unified HW compilation manifest
- Standardize all common settings
- Bypass parameters for “one-offs”
- Validated with 5 PDKs, 35 EDA tools, ASIC/FPGA/HLS flows


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

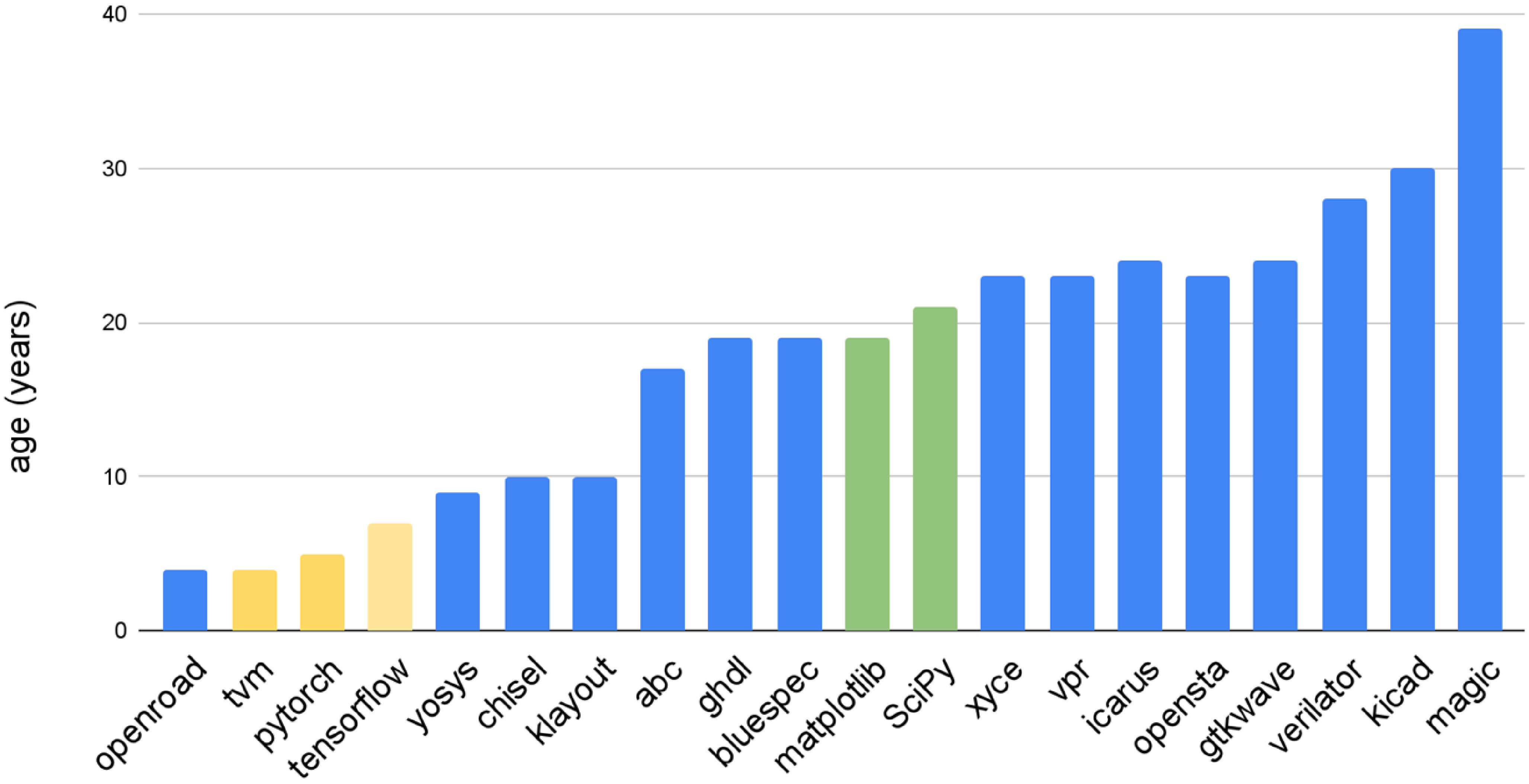
Source: https://xkcd.com/

So many options…which is the right one?

- **Manifests:**
  - SC, bender, fusesoc, metrics (openroad), mflowgen, hammer, cadre, orflow, openlane, bazelhdl, make/homegrown,...
- **Collaborations:**
  - Openroad Integration: https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts/commit/94e9240c7c5297e263cdca862b5c4b1df3fb0111
  - Caravael/skywater: https://github.com/siliconcompiler/siliconcompiler/commit/fcbf3f7f75b786f0577311fc0714d27db0b1faa5

SC is here to stay, not going anywhere!

Reminder: Software is a Lifetime of Maintenance….



Public Service Announcement

OpenROAD: https://github.com/The-OpenROAD-Project/OpenROAD

OpenSTA: https://github.com/The-OpenROAD-Project/OpenSTA

Align: https://github.com/ALIGN-analoglayout/ALIGN-public

Magical: https://github.com/magical-eda/MAGICAL

ACT: https://github.com/asyncvlsi/act

Xyce: https://github.com/Xyce/Xyce

SystemC-TLM-lib: https://github.com/Xilinx/libsystemctlm-soc

Pono: https://github.com/upscale-project/pono

LSOracle: https://github.com/lnis-uofu/LSOracle

OpenFPGA: https://github.com/lnis-uofu/OpenFPGA

PRGA: https://github.com/PrincetonUniversity/prga

BlackParrot: https://github.com/black-parrot/black-parrot

OpenFASOC: https://github.com/idea-fasoc/OpenFASOC

Were IDEA/POSH successful?

Dunno, you be the judge!

Summary of state of EDA/IP…

https://github.com/aolofsson/awesome-hardware-tools

https://github.com/aolofsson/awesome-opensource-hardware