

Breaking the LLDB/Python Revlock

Jonas Devlieghere

Python & LLDB


LLDB loads Python
scripts, data formatters, breakpoint callbacks
LLDB
Python
Python loads LLDB
import lldb

The Python Dependency

- The source depends on one version's C API
  - Uses the full C API
  - No stability beyond single minor version
- The binary depends on one version's library
  - Part of the install name or search paths
  - Dynamic loader will load that exact version

Failure Mode

```
$ lldb
<crash>
```

```
Python 3.14.8
>>> import lldb
<crash>
```

https://lldb.llvm.org/resources/caveats.html#python

The Limited API

- Python Limited API (PEP 384)
  - Introduced for Python 3.2 (2011)
  - Supported by SWIG since 4.2 (2023)
- What had to change?
  - Set `Py_LIMITED_API` to existing minimum (3.8)
  - Update APIs (*mostly* mechanical)
  - Update ABI tag in Python module: `_lldb.abi3.so`

Loading Python at Runtime

Search Paths

Let the dynamic loader find Python by name

Python's install names vary too much

`dlopen/dlsym`

Look up every symbol by hand

SWIG generates the calls, and data symbols can't be shimmed

Undefined Symbols

Compile against the headers but bind once Python is loaded

A missing symbol is a runtime crash rather than a link-time error

Loading Python at Runtime

Search Paths

Let the dynamic loader find Python by name

Python's install names vary too much

`dlopen/dlsym`

Look up every symbol by hand

SWIG generates the calls, and data symbols can't be shimmed

Undefined Symbols

Compile against the headers but bind once Python is loaded

A missing symbol is a runtime crash rather than a link-time error

Loading Python at Runtime

Search Paths

Let the dynamic loader find Python by name

Python's install names vary too much

`dlopen/dlsym`

Look up every symbol by hand

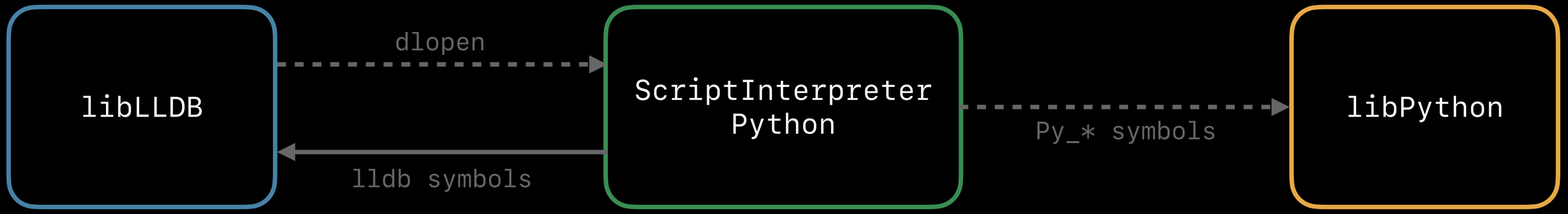
SWIG generates the calls, and data symbols can't be shimmed

Undefined Symbols

Compile against the headers but bind once Python is loaded

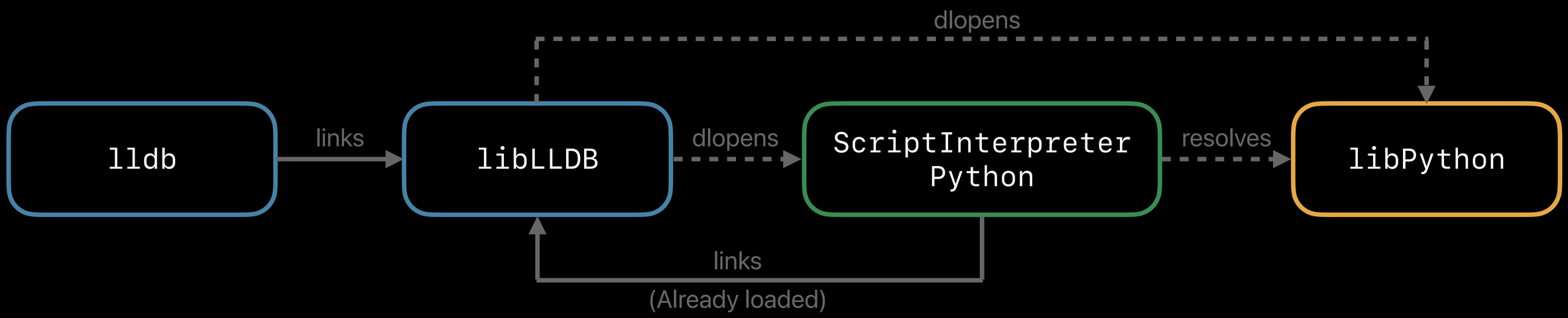
A missing symbol is a runtime crash rather than a link-time error

Script Interpreters as Dynamic Libraries



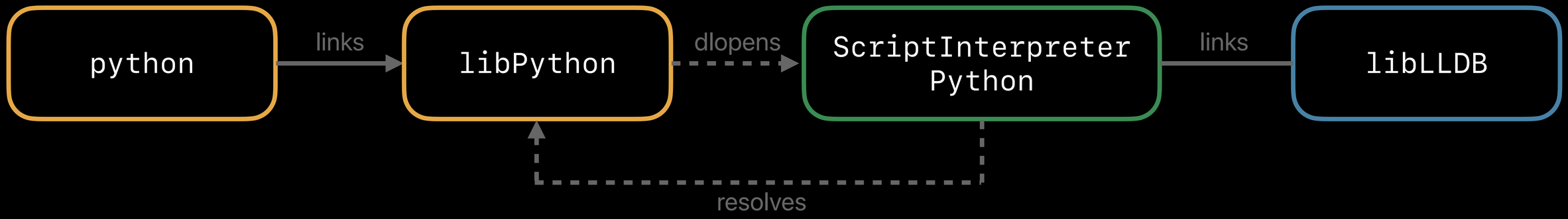
Script Interpreters as Dynamic Libraries

```
$ lldb
```



Script Interpreters as Dynamic Libraries

```
Python 3.14.8
>>> import lldb
```



Ambitious Project

- Limited API: *31 PRs*
  - 22 to adopt the Limited API
  - 9 follow-ups (CMake, SWIG, Windows)
- Dynamic script interpreters: *36 PRs*
  - 14 for the core changes + 7 changes to the PluginManager
  - 15 follow-ups (Exports, Framework, Windows, Docs)
- **Thank you** to everyone that helped make this happen

Conclusion

- LLDB is no longer revlocked to the Python it was built against
  - `import lldb` works across versions of the Python interpreter
  - The LLDB library can programmatically pick which Python to load

- LLDB still works without Python
  - No longer crashes
  - Error message + reduced feature set



TM and © 2026 Apple Inc. All rights reserved.