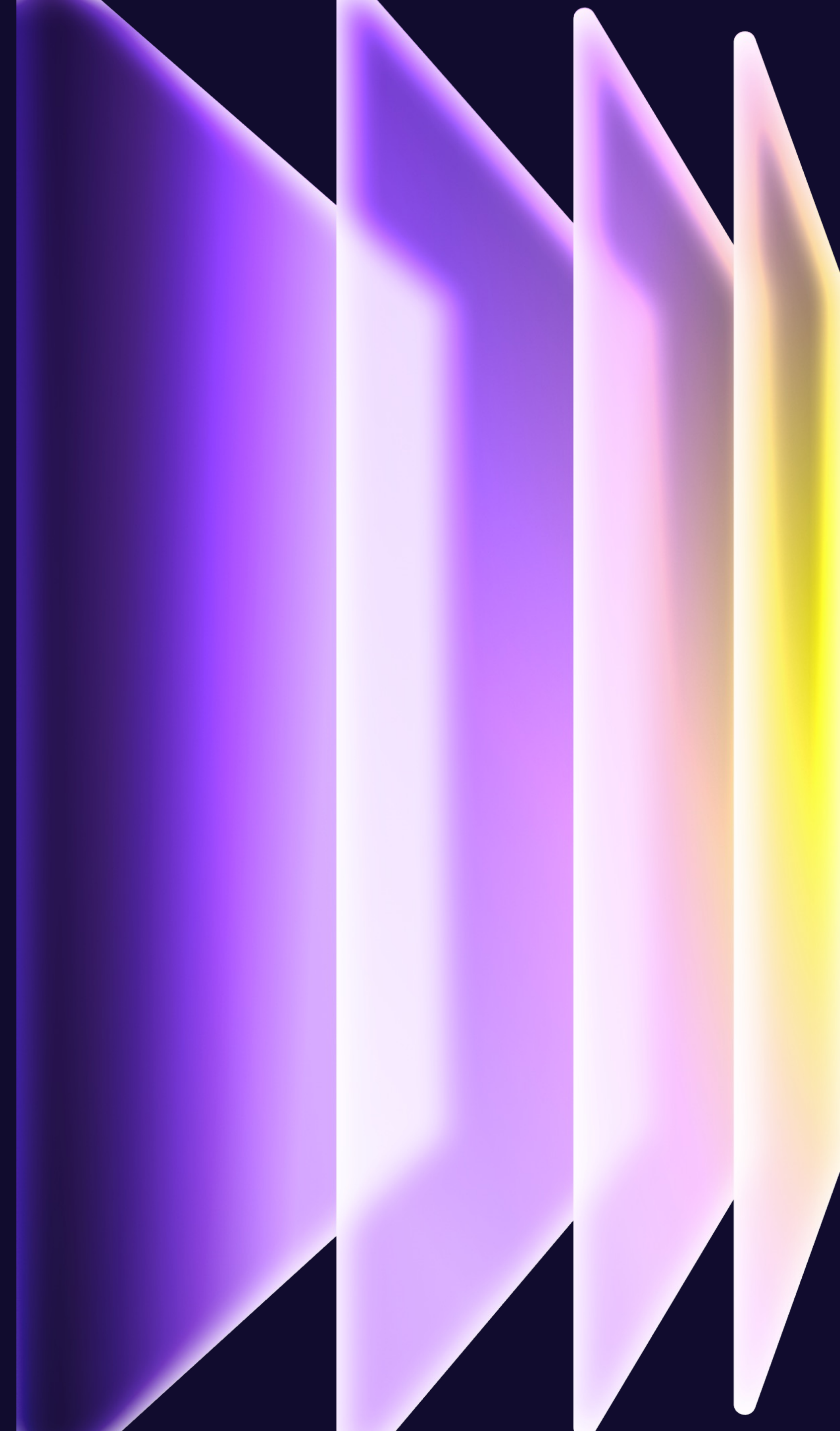


Внутреннее устройство бинарных сборок Python

Артамонов Александр,
руководитель службы инструментов сборки



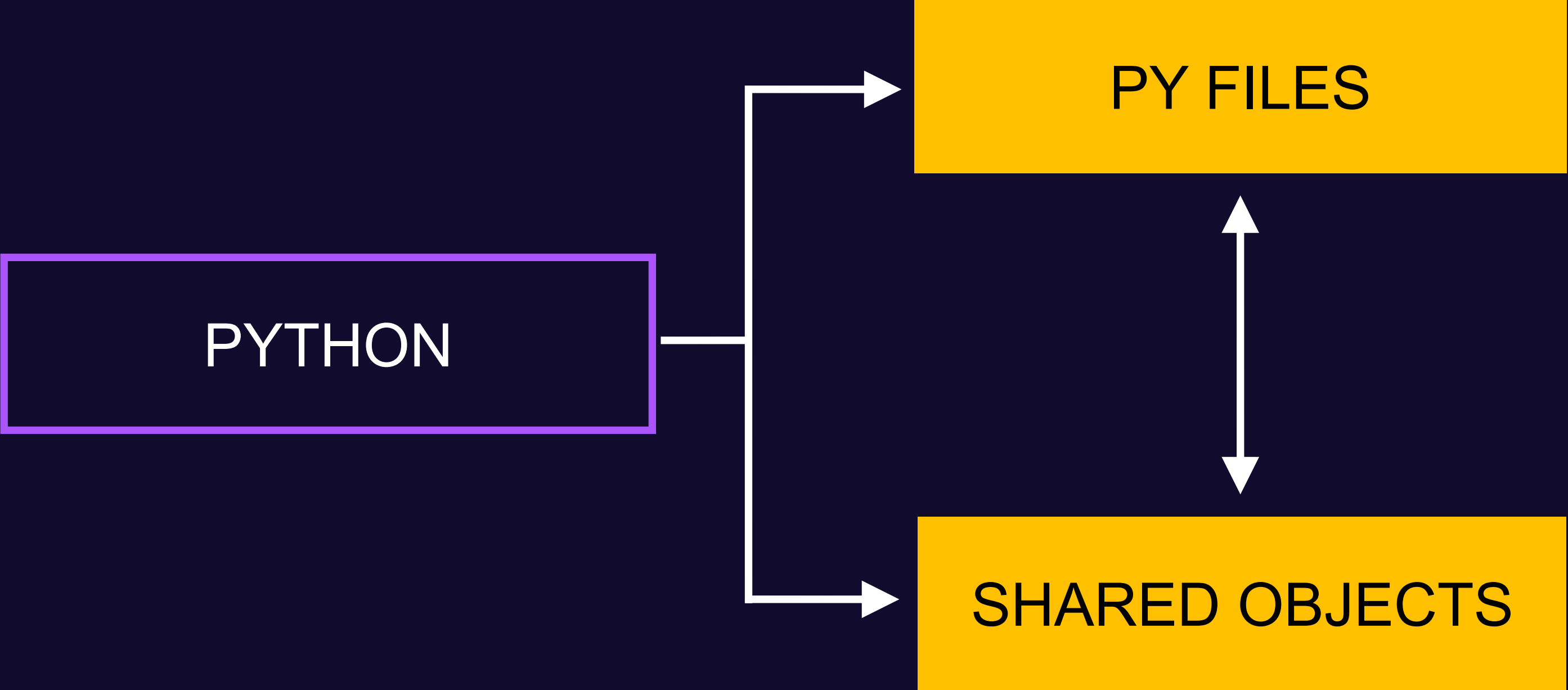
План доклада

- 1 Бинарная сборка Python
- 2 Обзорно про PyInstaller, Nuitka, yatool
- 3 Стадия поиска зависимостей
- 4 Упаковка зависимостей
- 5 Устройство Runtime
- 6 Вместо заключения

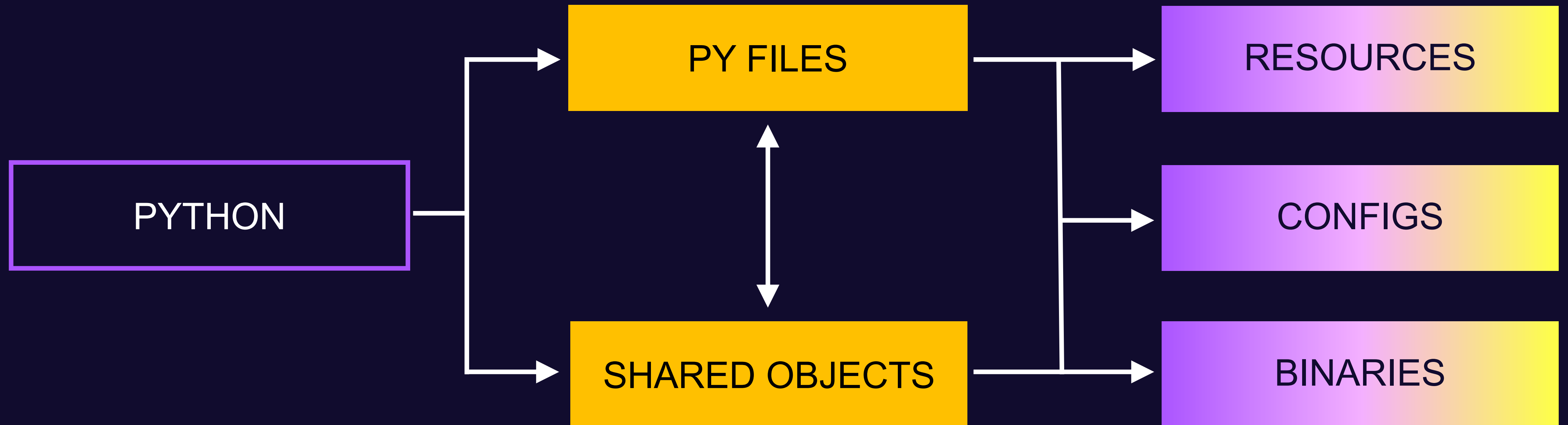


Бинарная сборка Python

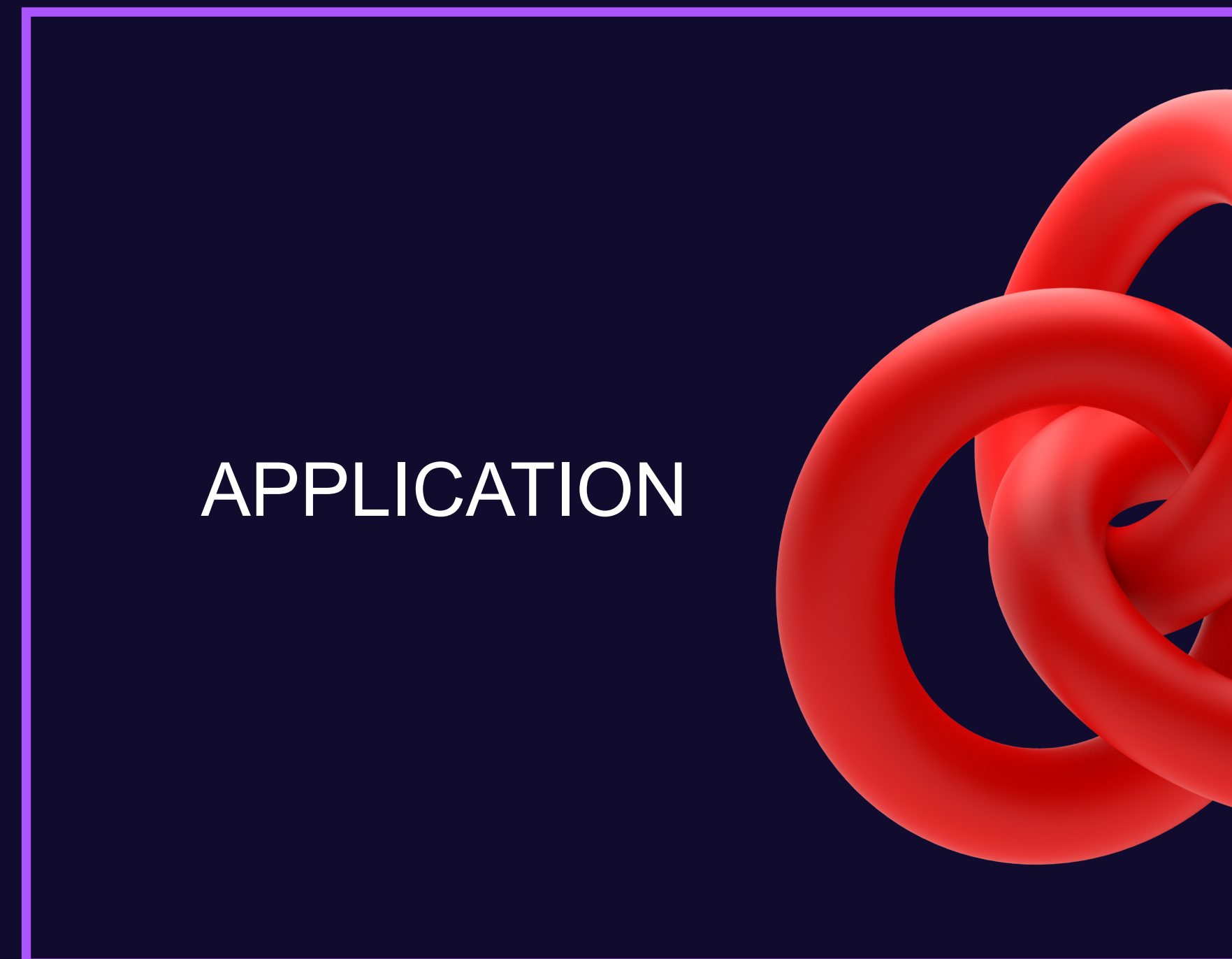
Конвенциональная дистрибуция



Конвенциональная дистрибуция



Герметичная автономная дистрибуция



Герметичная автономная дистрибуция



(чемодан в питоне)

Область применения

Desktop Applications

Optimizations

Tooling

Games

Инструменты дистрибуции

cx-freeze

docker

wheel

vagrant

PyInstaller

pip

PyOxidizer

deb

poetry

single file

yatool

PythonZipper

Nuitka

Hatch

zip

py2app

pipx

Герметичная автономная дистрибуция

cx-freeze

docker

wheel

vagrant

PyInstaller

pip

PyOxidizer

deb

poetry

single file

yatool

PythonZipper

Nuitka

Hatch

zip

py2app

pipx



PyInstaller, Nuitka, yatoool

PyInstaller



Bundler

SFX with Python
and dependencies

Self contained*

Feature parity

Uses system
Python

Nuitka



Transpiler Toolchain

Transpiles Python to C
and bundles Python with
dependencies

Env. deps

Requires compiler,
patchelf, ldd, ccache, etc

Feature agnate

CPython
approximation

yatool



**Build
system**

Builds CPython with
dependencies

Self contained

Linux experience

Feature parity

Uses fixed
Python

Стадии для анализа

Сборка

- 1 Поиск зависимостей
- 2 Упаковка зависимостей

Эксплуатация

- 3 Исследование особенностей Runtime

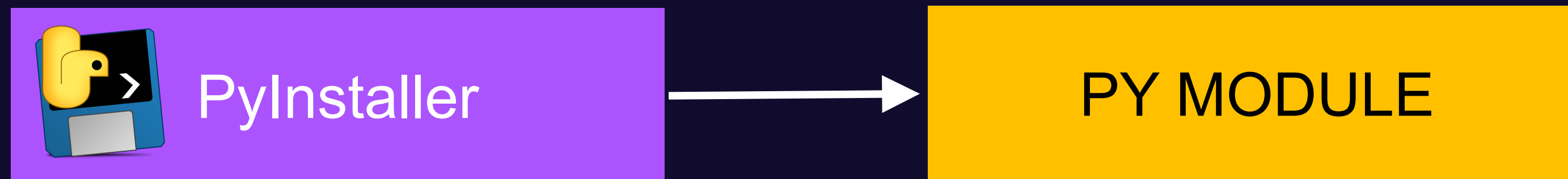
Disclaimer



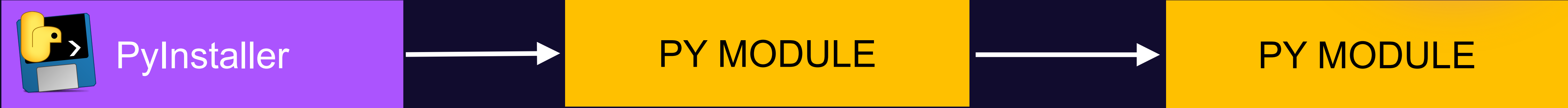


Поиск зависимостей

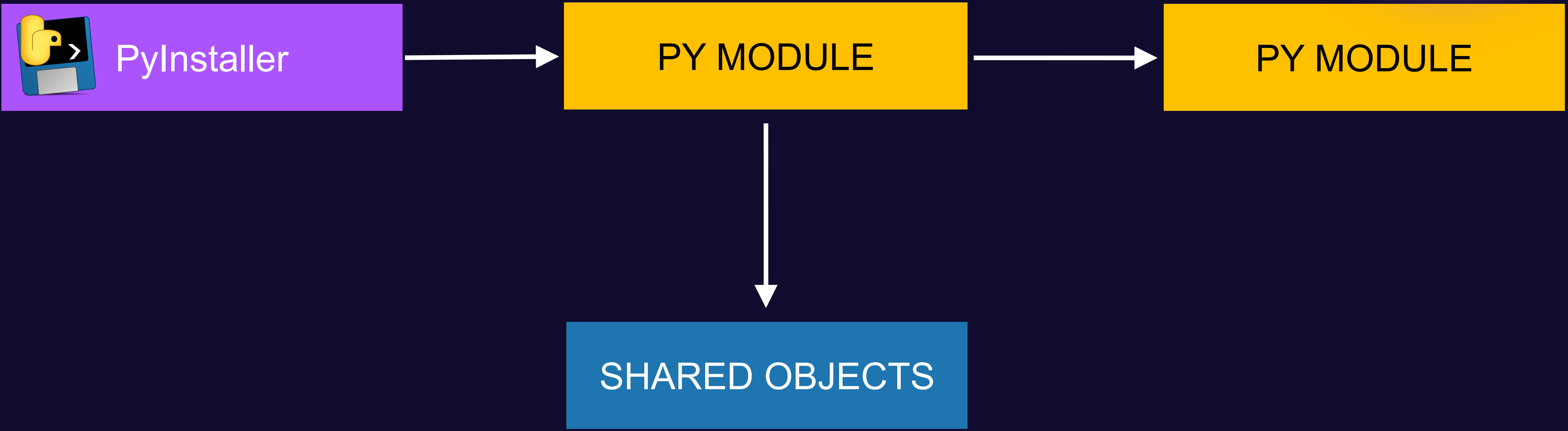
Поиск зависимостей - PyInstaller



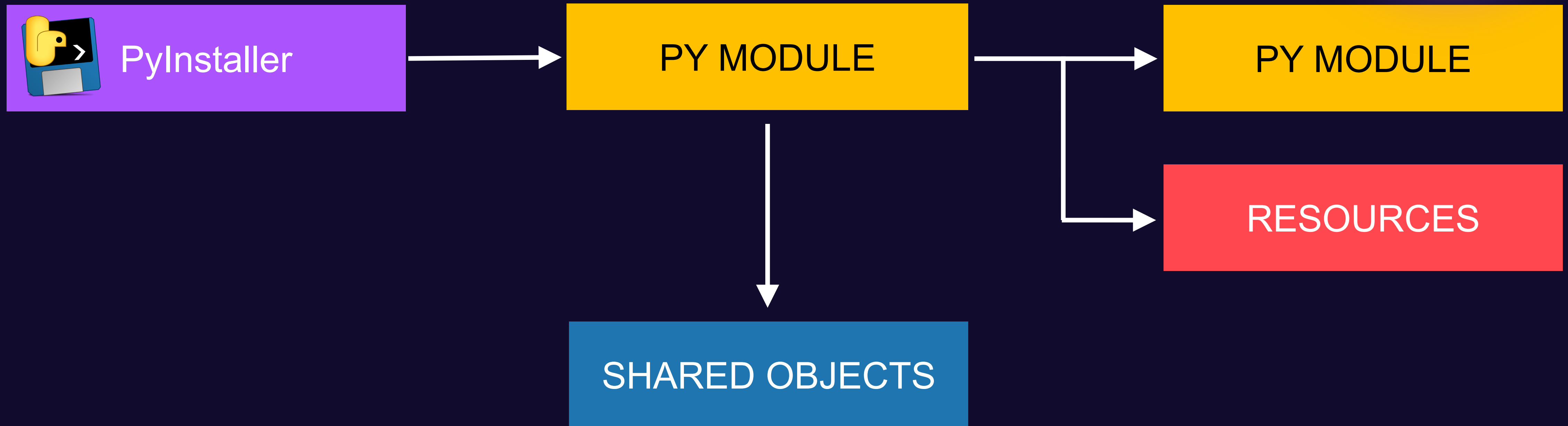
Поиск зависимостей - PyInstaller



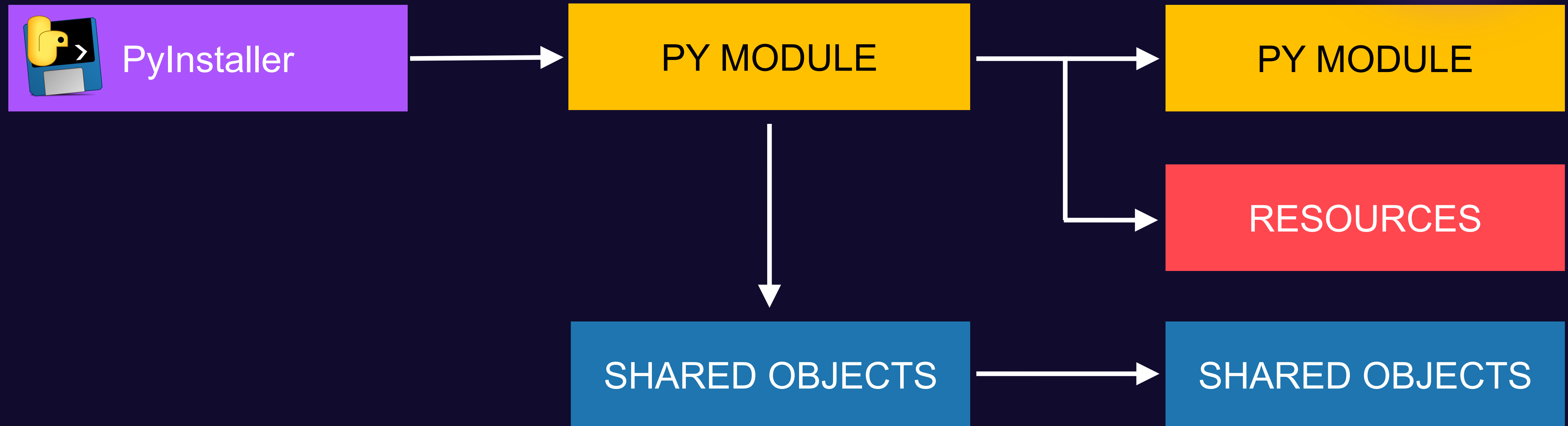
Поиск зависимостей - PyInstaller



Поиск зависимостей - PyInstaller



Поиск зависимостей - PyInstaller

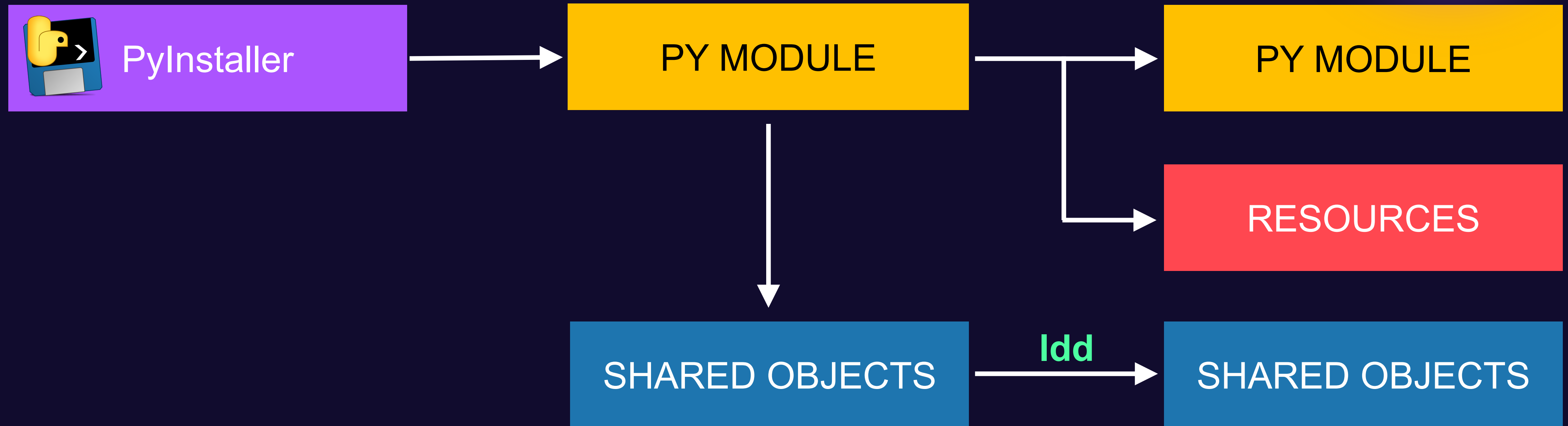


Поиск зависимостей - PyInstaller

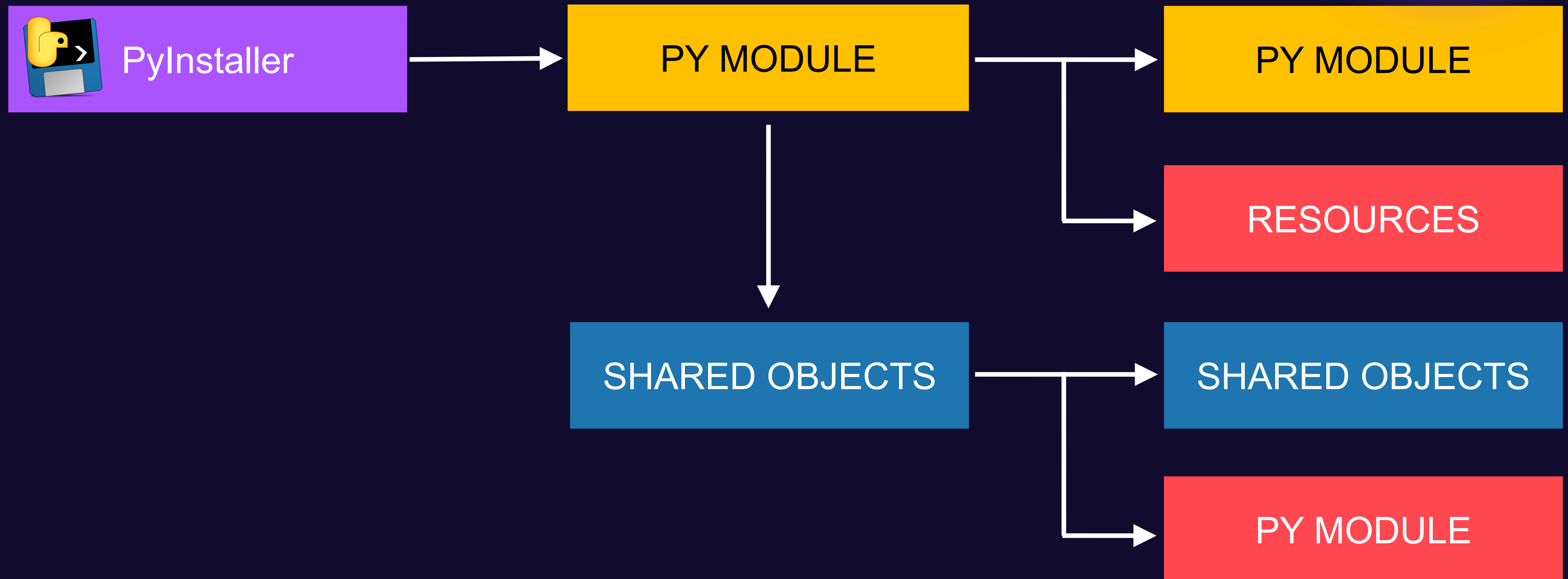


```
linux ~> ldd <binary>
linux-vdso.so.1 (0x00007ffd319fe000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007eff0ba9c000)
librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1 (0x00007eff0ba92000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007eff0ba6f000)
libcrypt.so.1 => /lib/x86_64-linux-gnu/libcrypt.so.1 (0x00007eff0ba34000)
libutil.so.1 => /lib/x86_64-linux-gnu/libutil.so.1 (0x00007eff0ba2f000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007eff0b83b000)
/lib64/ld-linux-x86-64.so.2 (0x00007eff0bab7000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007eff0b6ec000)
```

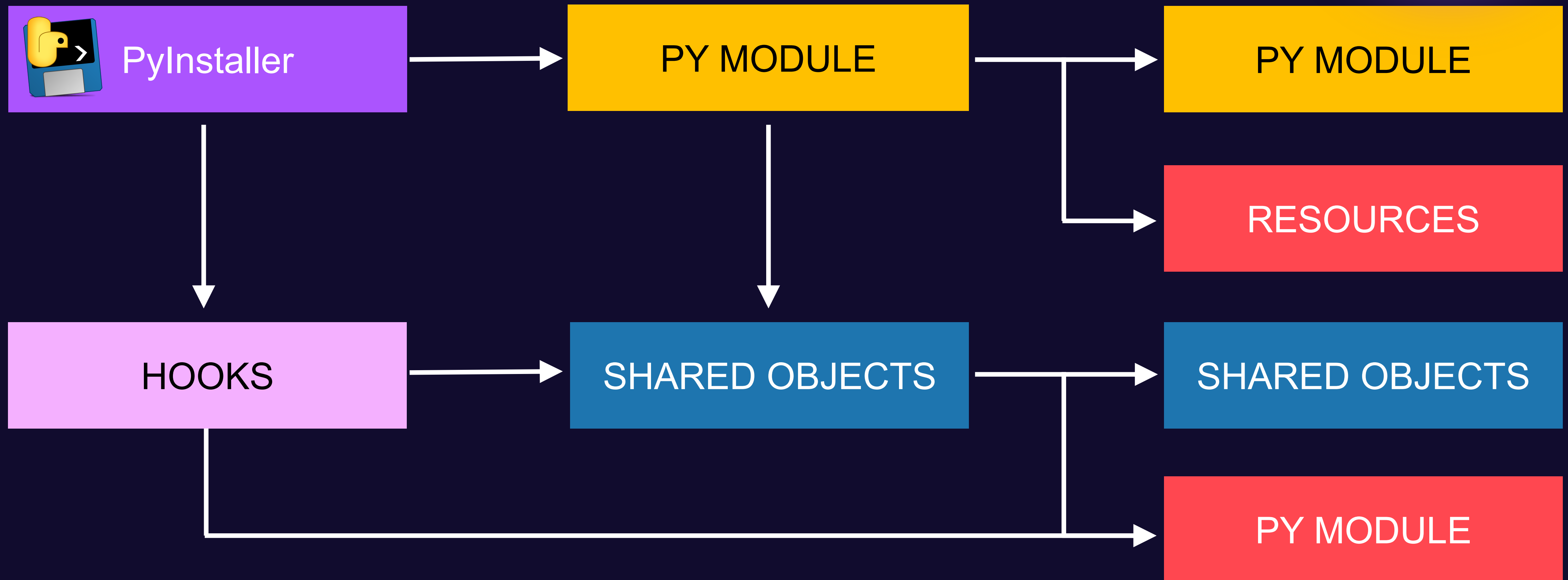
Поиск зависимостей - PyInstaller



Поиск зависимостей - PyInstaller



Поиск зависимостей - PyInstaller

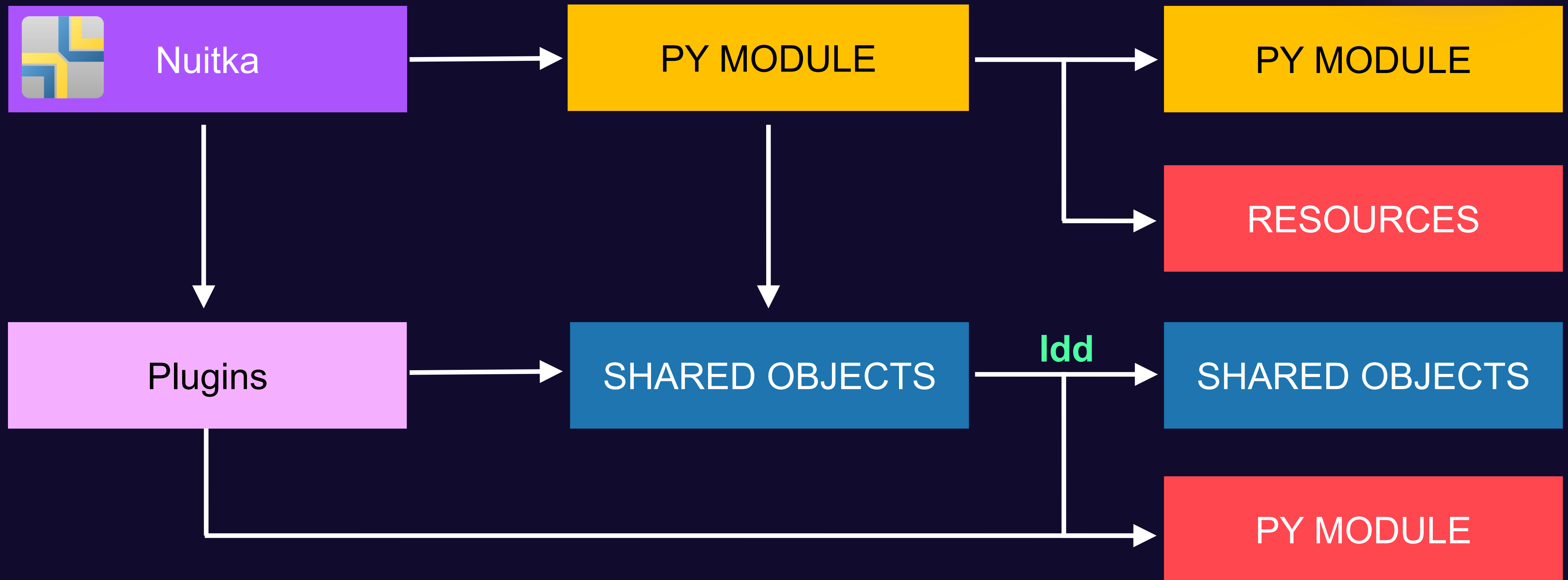




Поиск зависимостей - PyInstaller

- Разбор AST
- Реестр hook'ов
- Поиск транзитивных зависимостей
- Отсутствие индуцированных зависимостей
- Отсутствие resource auto-bundling
- Зависимость от /lib and /usr/lib

Поиск зависимостей - Nuitka



Поиск зависимостей



PyInstaller

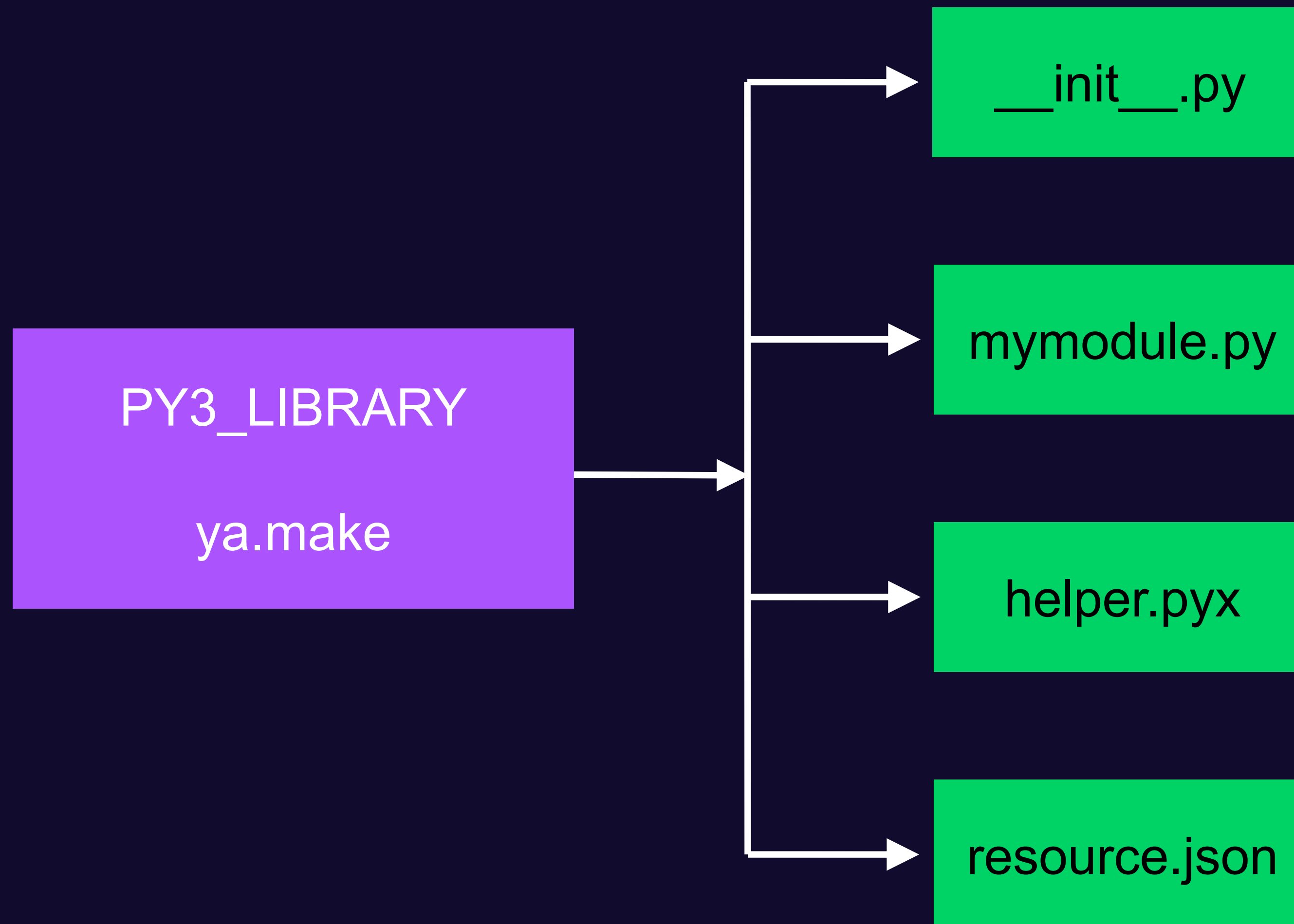
- Разбор AST
- Реестр **Plugin'ов**
- Поиск транзитивных зависимостей
- Отсутствие индуцированных зависимостей
- Отсутствие resource auto-bundling
- Зависимость от **некоторых библиотек**



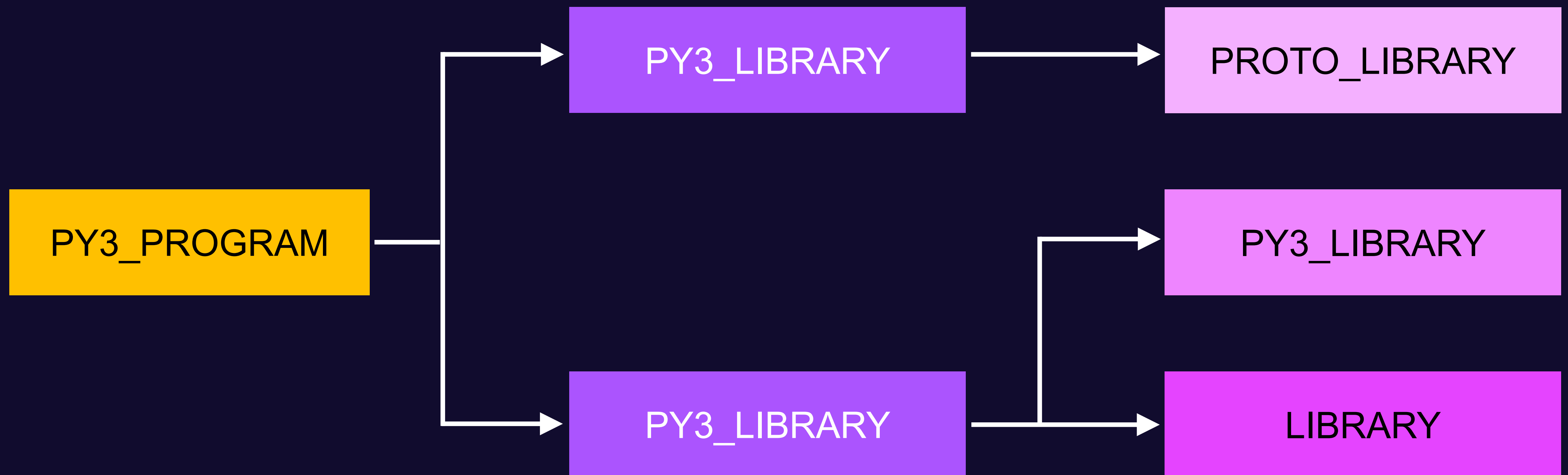
Nuitka

- Разбор AST
- Реестр **hook'ов**
- Поиск транзитивных зависимостей
- Отсутствие индуцированных зависимостей
- Отсутствие resource auto-bundling
- Зависимость от **/lib and /usr/lib**

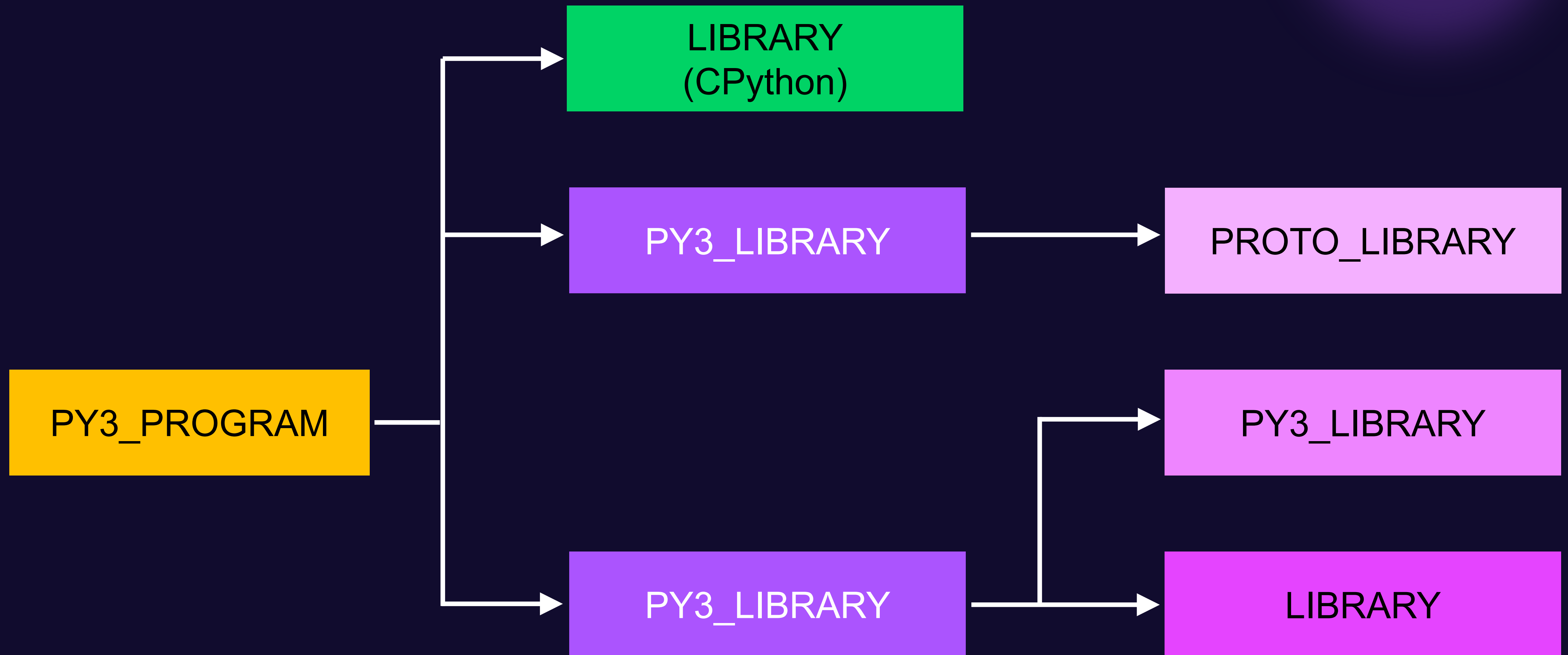
Поиск зависимостей - yatool



Поиск зависимостей - yatoool



Поиск зависимостей - yatoool



Поиск зависимостей

Yatool

- `ya.make` с описанием зависимостей
- Все исходники в репозитории
- Поиск транзитивных зависимостей
- Выведение индуцированных зависимостей
- Ресурс — зависимость библиотеки
- Платформено-агностичная сборка

Nuitka

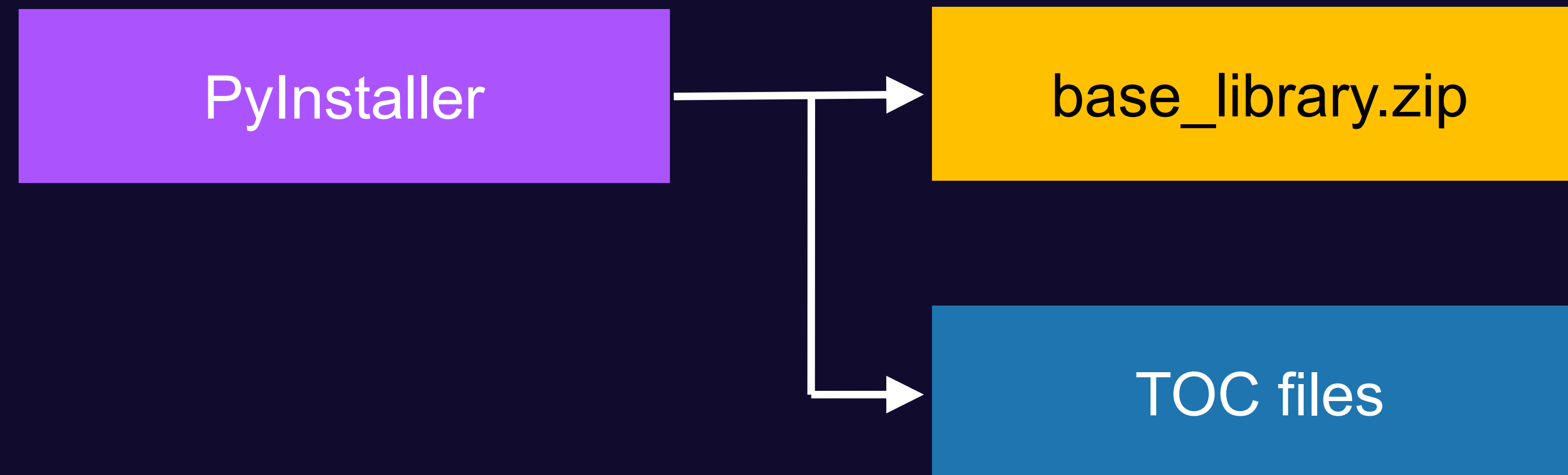
PyInstaller

- Разбор AST
- Реестр hook'ов
- Поиск транзитивных зависимостей
- Отсутствие индуцированных зависимостей
- Отсутствие resource auto-bundling
- Зависимость от `/lib` and `/usr/lib`

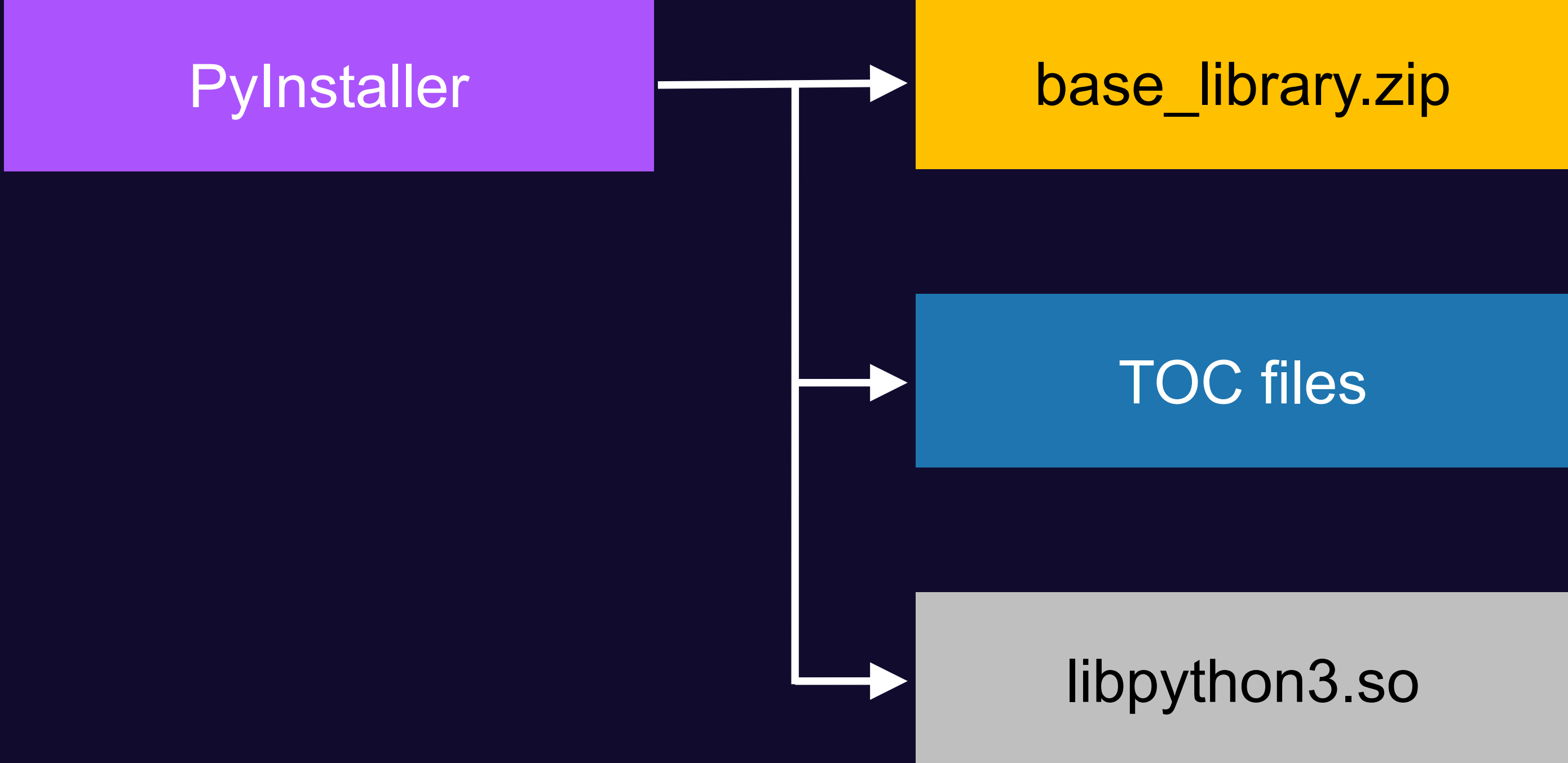


Упаковка зависимостей

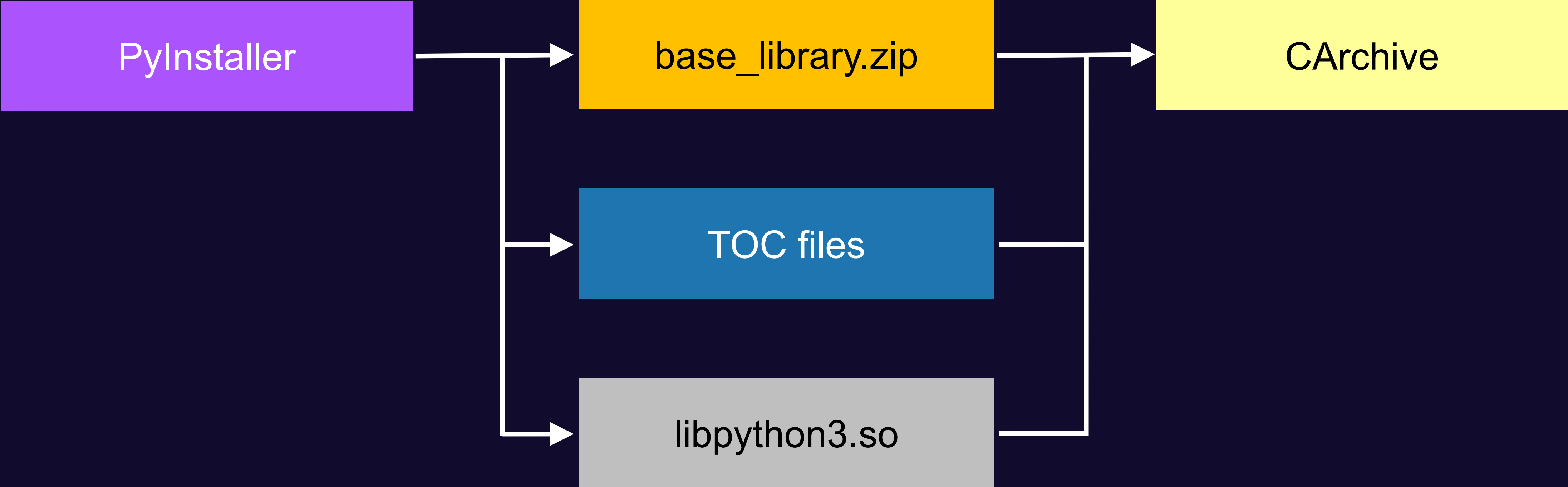
Сборка - PyInstaller



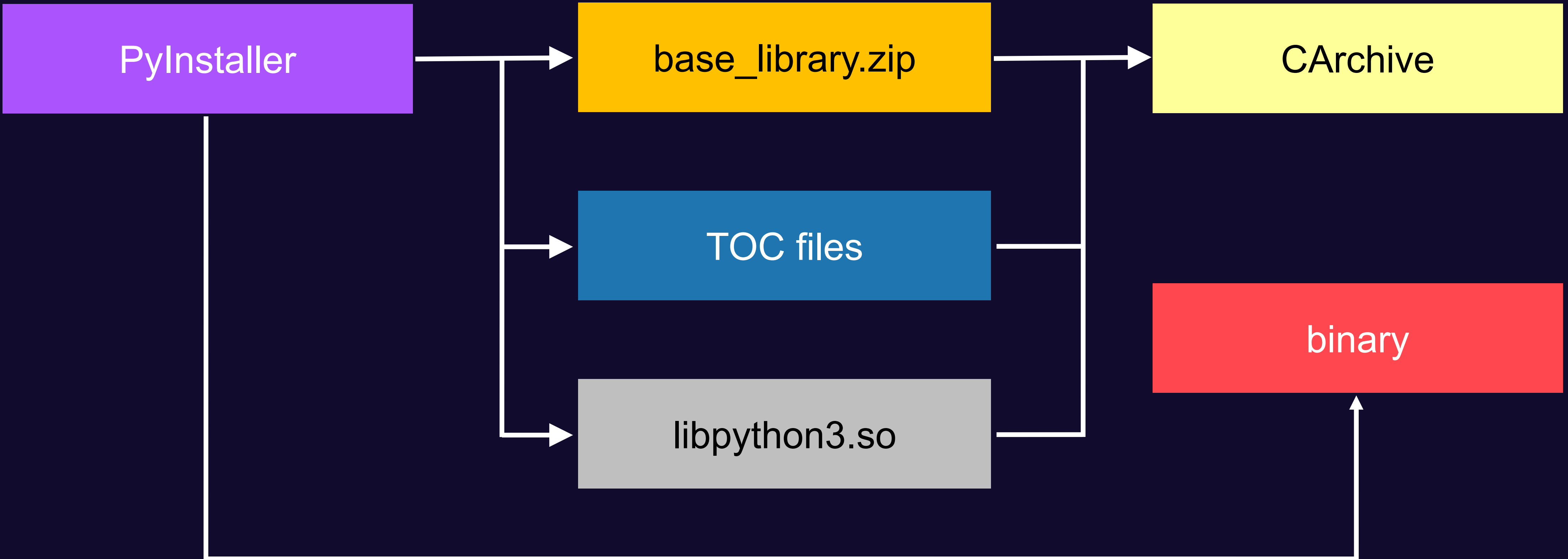
Сборка - PyInstaller



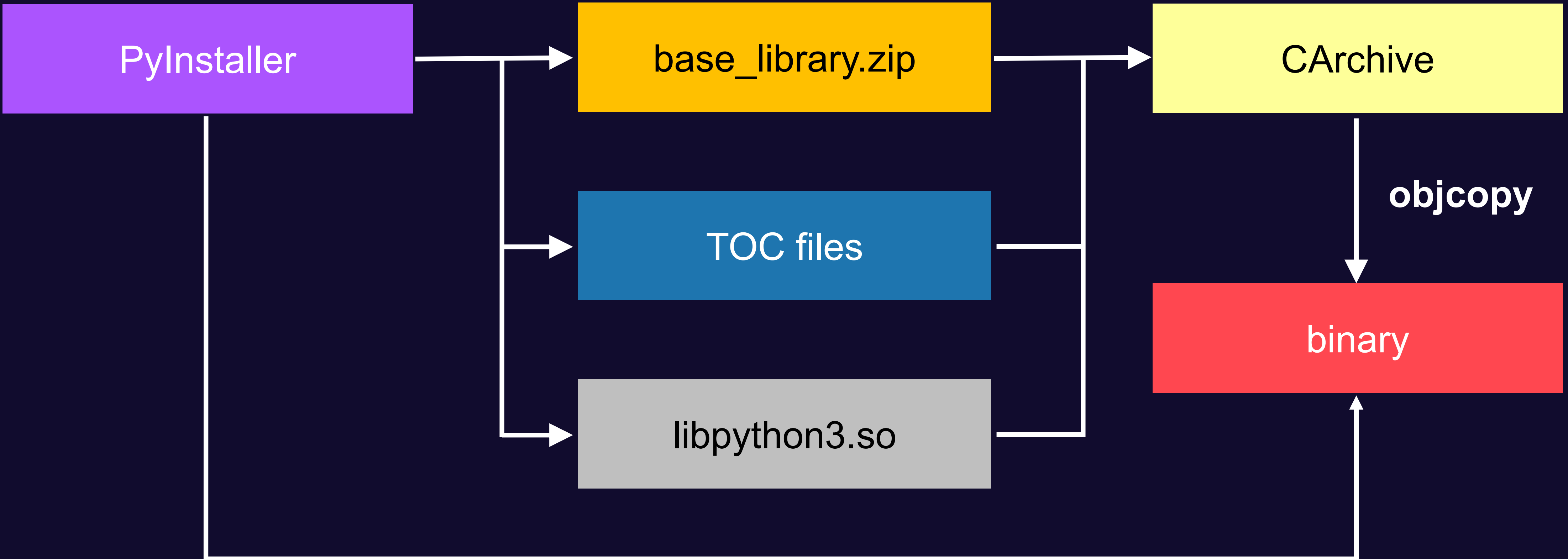
Сборка - PyInstaller



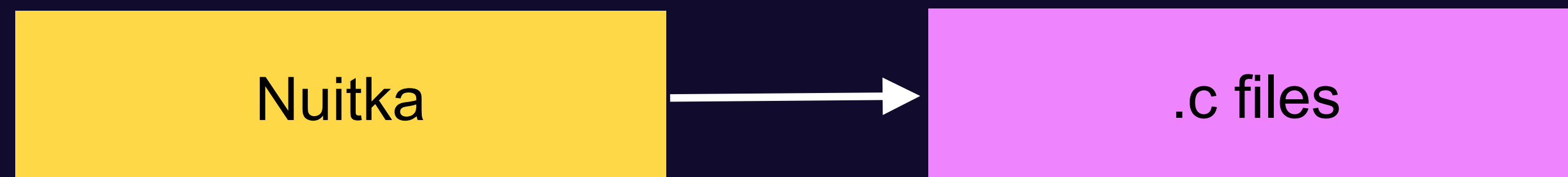
Сборка - PyInstaller



Сборка - PyInstaller



Сборка - Nuitka

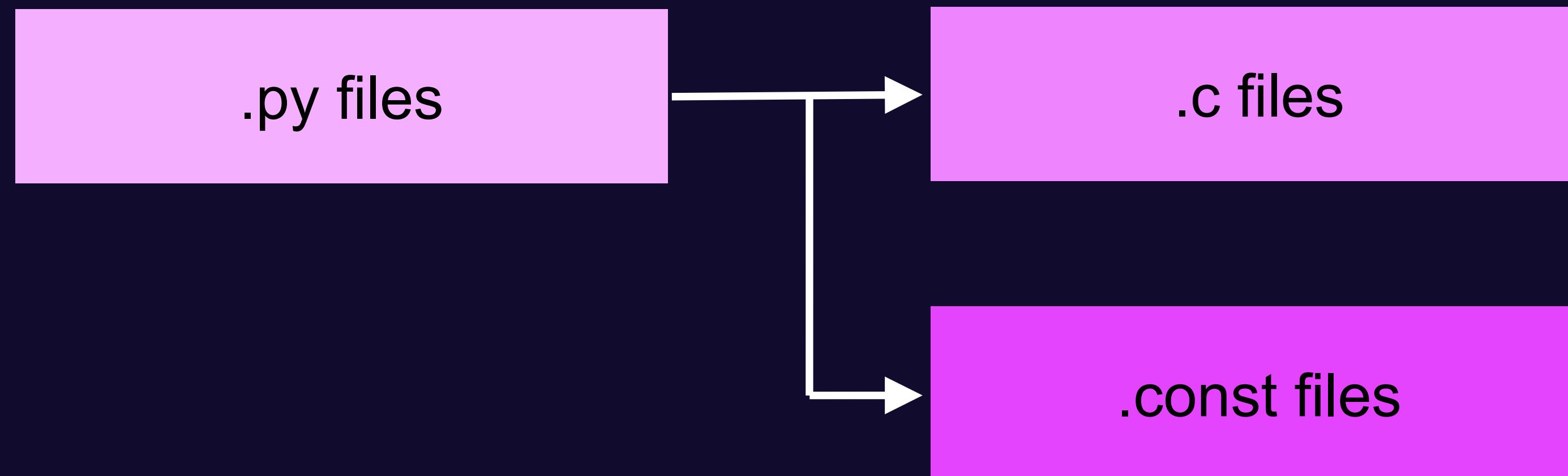


Транспиляция - Nuitka



.py files

Транспиляция - Nuitka



Транспиляция - Nuitka



Транспиляция - Nuitka



```
def func(a):  
    b = 200 * 30 * 2  
    c = b // 100  
    return b - c * a
```

Транспиляция - Nuitka



```
tmp_sub_expr_left_1 = mod_consts[0];
tmp_mult_expr_left_1 = mod_consts[1];
...
tmp_sub_expr_right_1 = BINARY_OPERATION_MULT_OBJECT_LONG_OBJECT(
    tmp_mult_expr_left_1,
    tmp_mult_expr_right_1);
...
tmp_return_value = BINARY_OPERATION_SUB_OBJECT_LONG_OBJECT(
    tmp_sub_expr_left_1,
    tmp_sub_expr_right_1);
```

Транспиляция - Nuitka



```
tmp_sub_expr_left_1 = mod_consts[0];
tmp_mult_expr_left_1 = mod_consts[1];
...
tmp_sub_expr_right_1 = BINARY_OPERATION_MULT_OBJECT_LONG_OBJECT(
    tmp_mult_expr_left_1,
    tmp_mult_expr_right_1);
...
tmp_return_value = BINARY_OPERATION_SUB_OBJECT_LONG_OBJECT(
    tmp_sub_expr_left_1,
    tmp_sub_expr_right_1);
```


Транспиляция - Nuitka



```
tmp_sub_expr_left_1 = mod_consts[0];
tmp_mult_expr_left_1 = mod_consts[1];
...
tmp_sub_expr_right_1 = BINARY_OPERATION_MULT_OBJECT_LONG_OBJECT(
    tmp_mult_expr_left_1,
    tmp_mult_expr_right_1);
...
tmp_return_value = BINARY_OPERATION_SUB_OBJECT_LONG_OBJECT(
    tmp_sub_expr_left_1,
    tmp_sub_expr_right_1);
```

Транспиляция - Nuitka



```
def func(a):  
    b = 200 * 30 * 2  
    c = b // 100  
    return b - c * a
```

Транспиляция - Nuitka



```
def func(a):  
    b = 12000  
    c = b // 100  
    return b - c * a
```

Транспиляция - Nuitka



```
def func(a):  
    b = 12000  
    c = 120  
    return b - c * a
```

Транспиляция - Nuitka



```
def func(a):  
    return 12000 - 120 * a
```

Транспиляция - Cython



```
__pyx_v_b = __pyx_int_12000;  
...  
__pyx_t_1 = __Pyx_PyInt_FloorDivideObjC(__pyx_v_b, __pyx_int_100, 0x64, 0, 0);  
...  
__pyx_t_1 = PyNumber_Multiply(__pyx_v_c, __pyx_v_a);  
__pyx_t_2 = PyNumber_Subtract(__pyx_v_b, __pyx_t_1);
```


Транспиляция - Cython



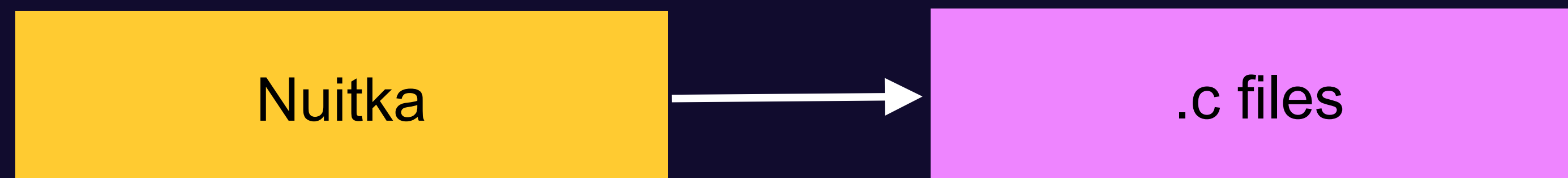
```
__pyx_v_b = __pyx_int_12000;
...
__pyx_t_1 = __Pyx_PyInt_FloorDivideObjC(__pyx_v_b, __pyx_int_100, 0x64, 0, 0);
...
__pyx_t_1 = PyNumber_Multiply(__pyx_v_c, __pyx_v_a);
__pyx_t_2 = PyNumber_Subtract(__pyx_v_b, __pyx_t_1);
```

Транспиляция - Cython

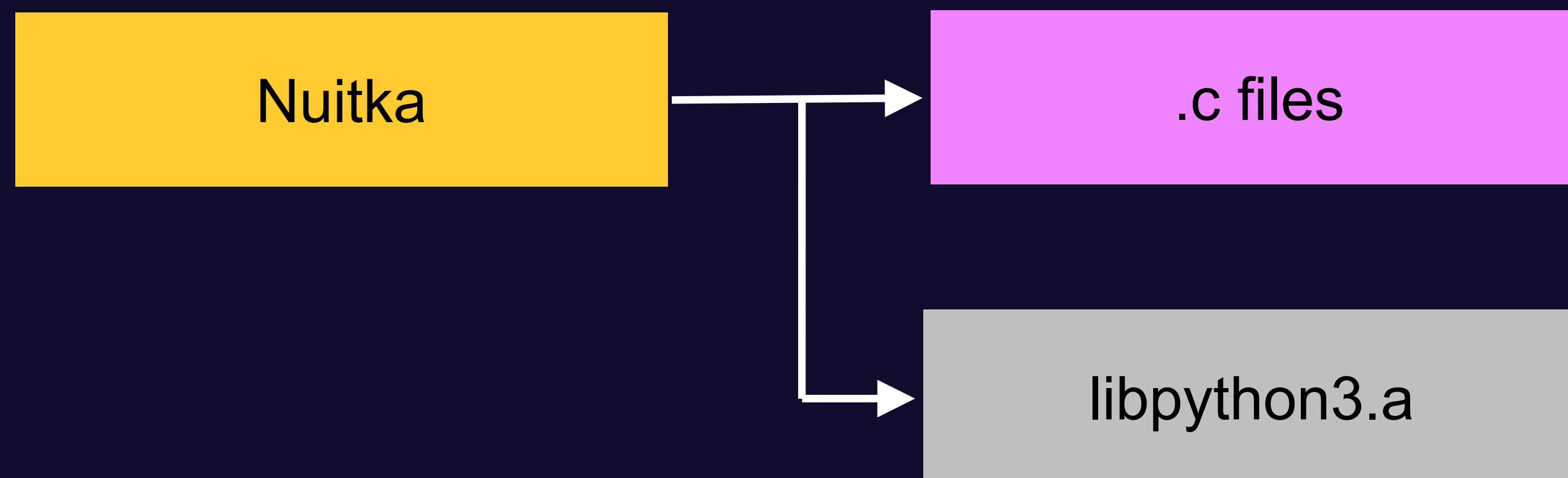


```
__pyx_v_b = __pyx_int_12000;  
...  
__pyx_t_1 = __Pyx_PyInt_FloorDivideObjC(__pyx_v_b, __pyx_int_100, 0x64, 0, 0);  
...  
__pyx_t_1 = PyNumber_Multiply(__pyx_v_c, __pyx_v_a);  
__pyx_t_2 = PyNumber_Subtract(__pyx_v_b, __pyx_t_1);
```

Сборка - Nuitka



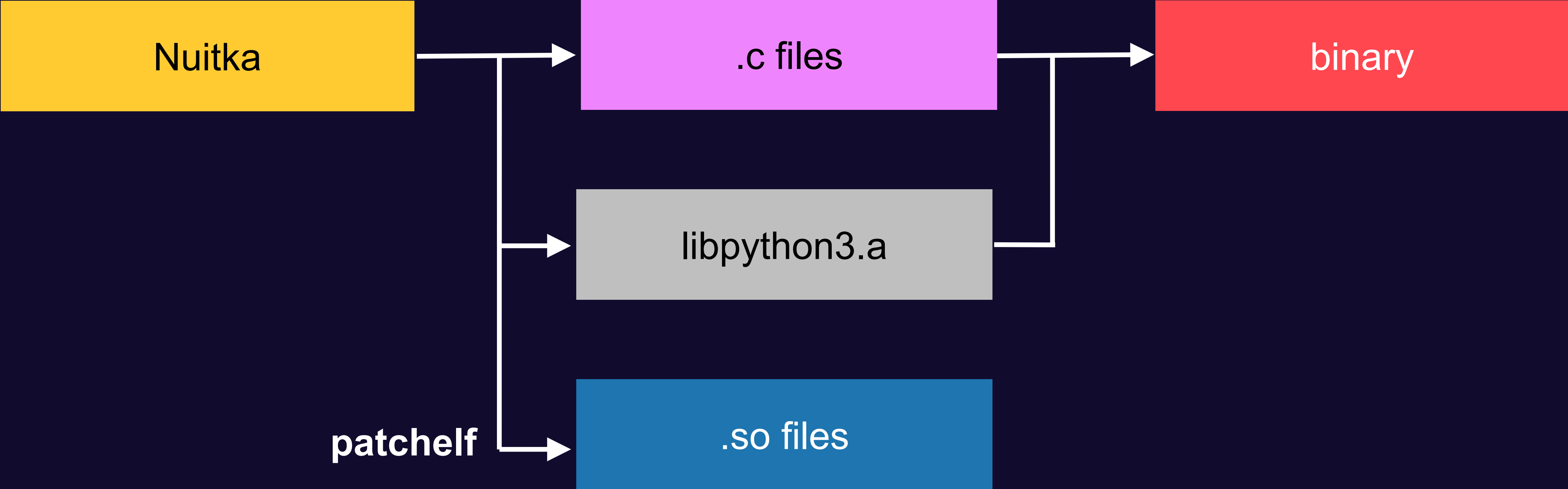
Сборка - Nuitka



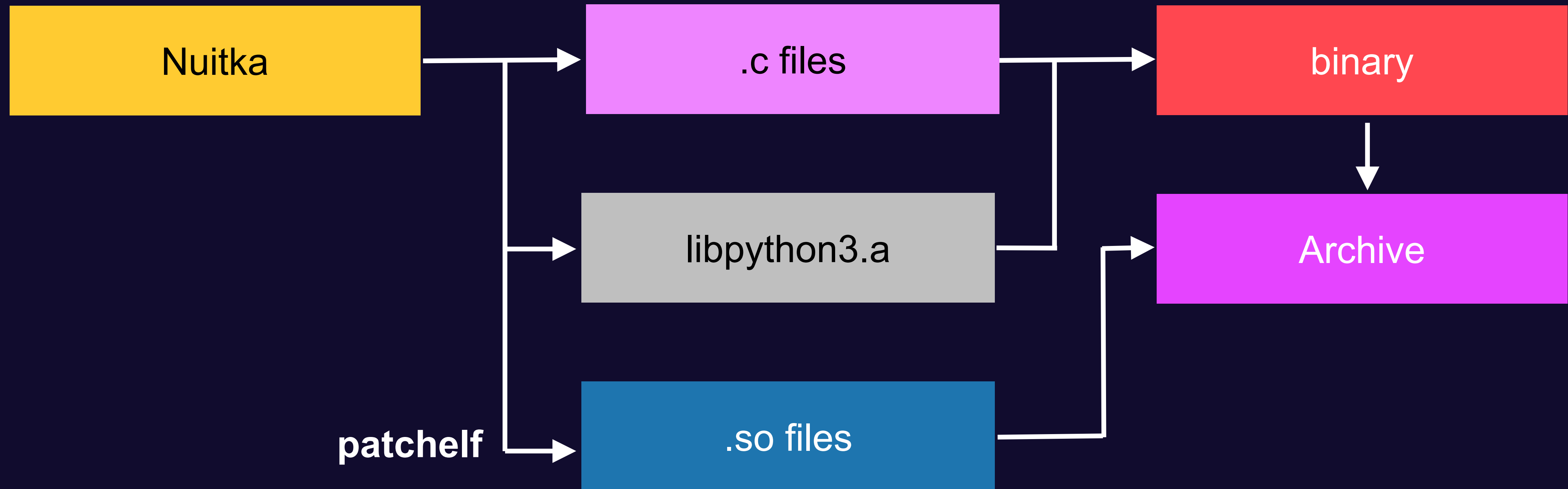
Сборка - Nuitka



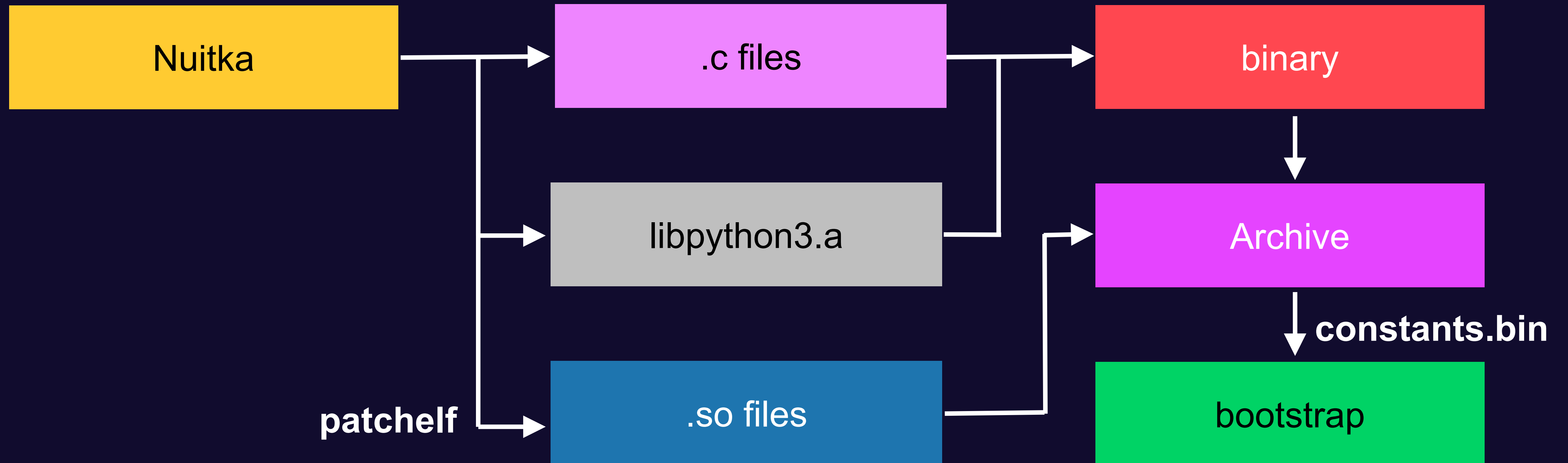
Сборка - Nuitka



Сборка - Nuitka



Сборка - Nuitka



Сборка - yatool



Сборка py файлов - уаtool

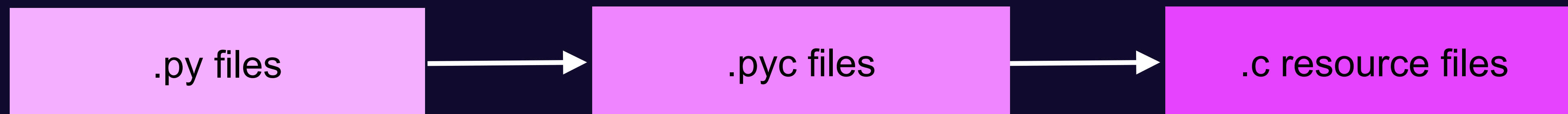


.py files

Сборка py файлов - уатоол



Сборка py файлов - yatool



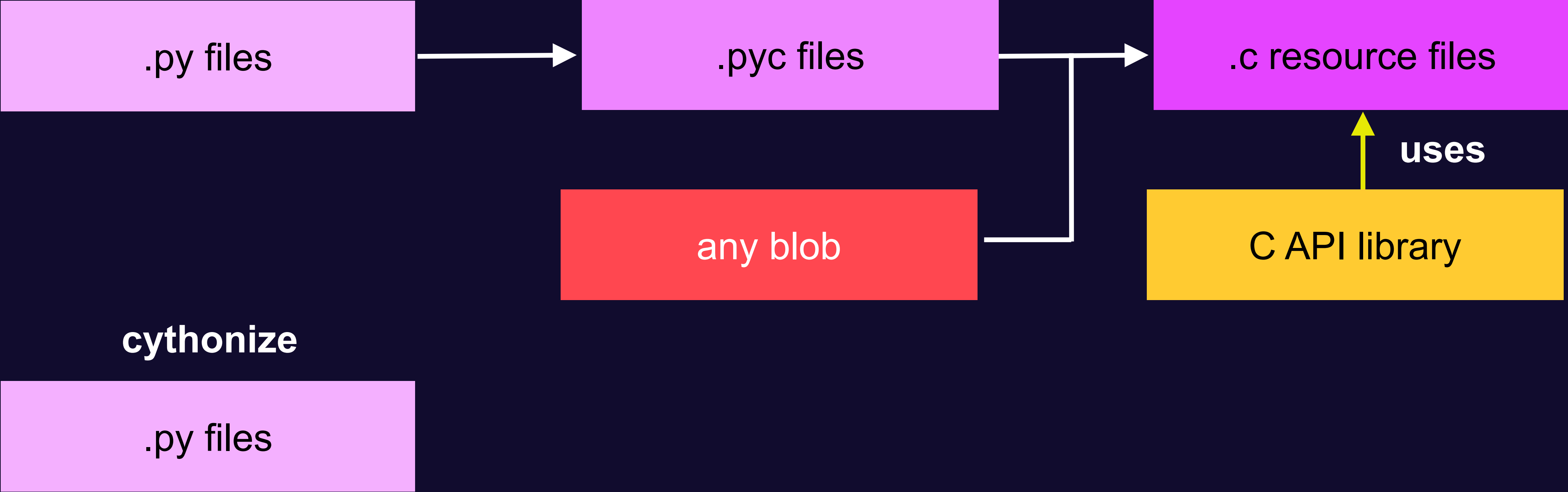
Сборка py файлов - yatool



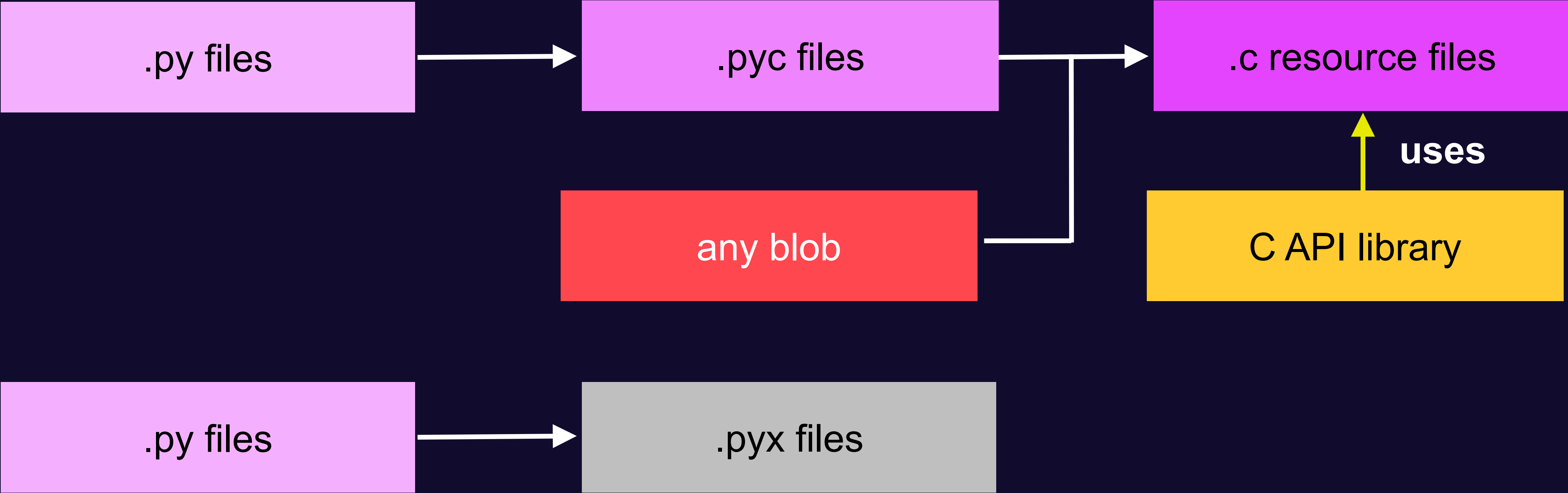
Сборка py файлов - yatool



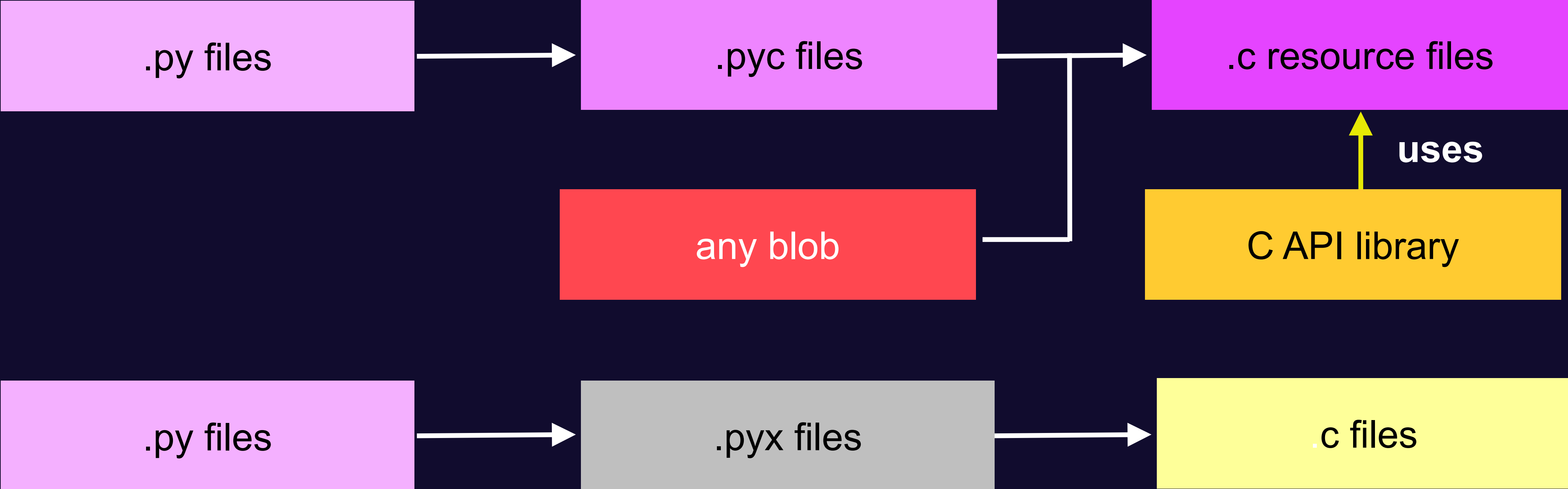
Сборка py файлов - yatool



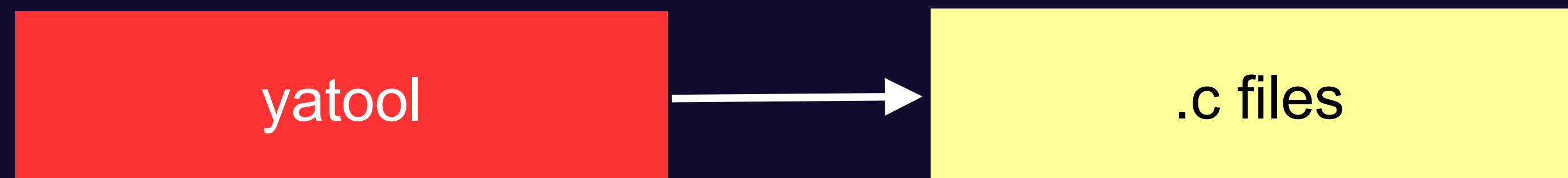
Сборка py файлов - yatool



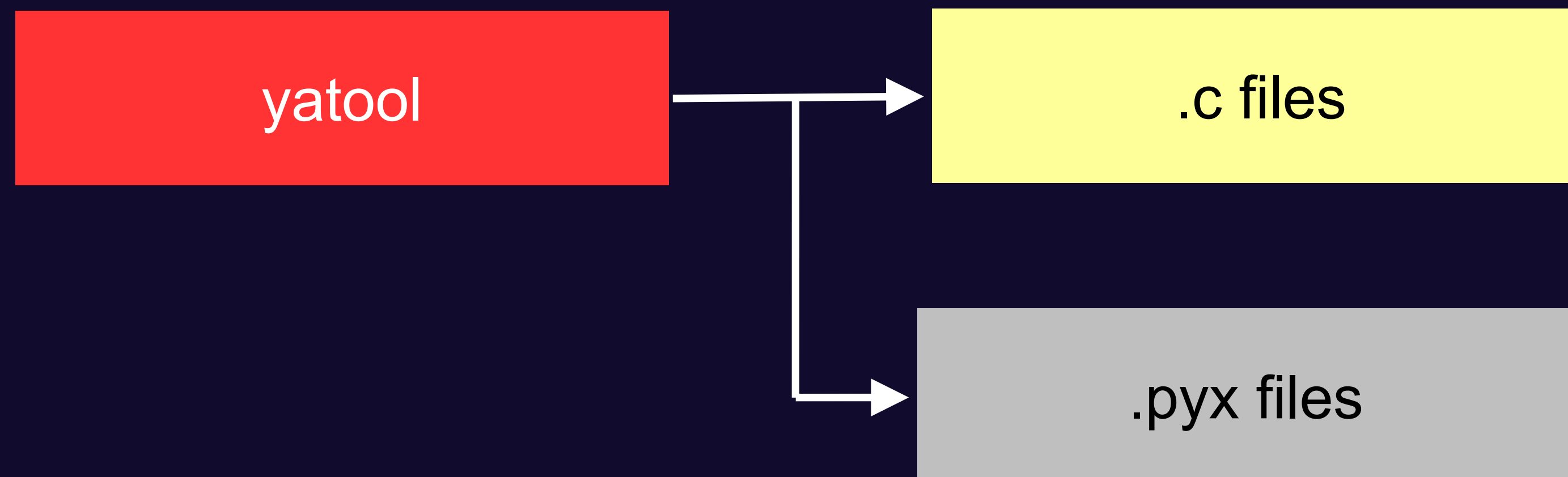
Сборка py файлов - yatool



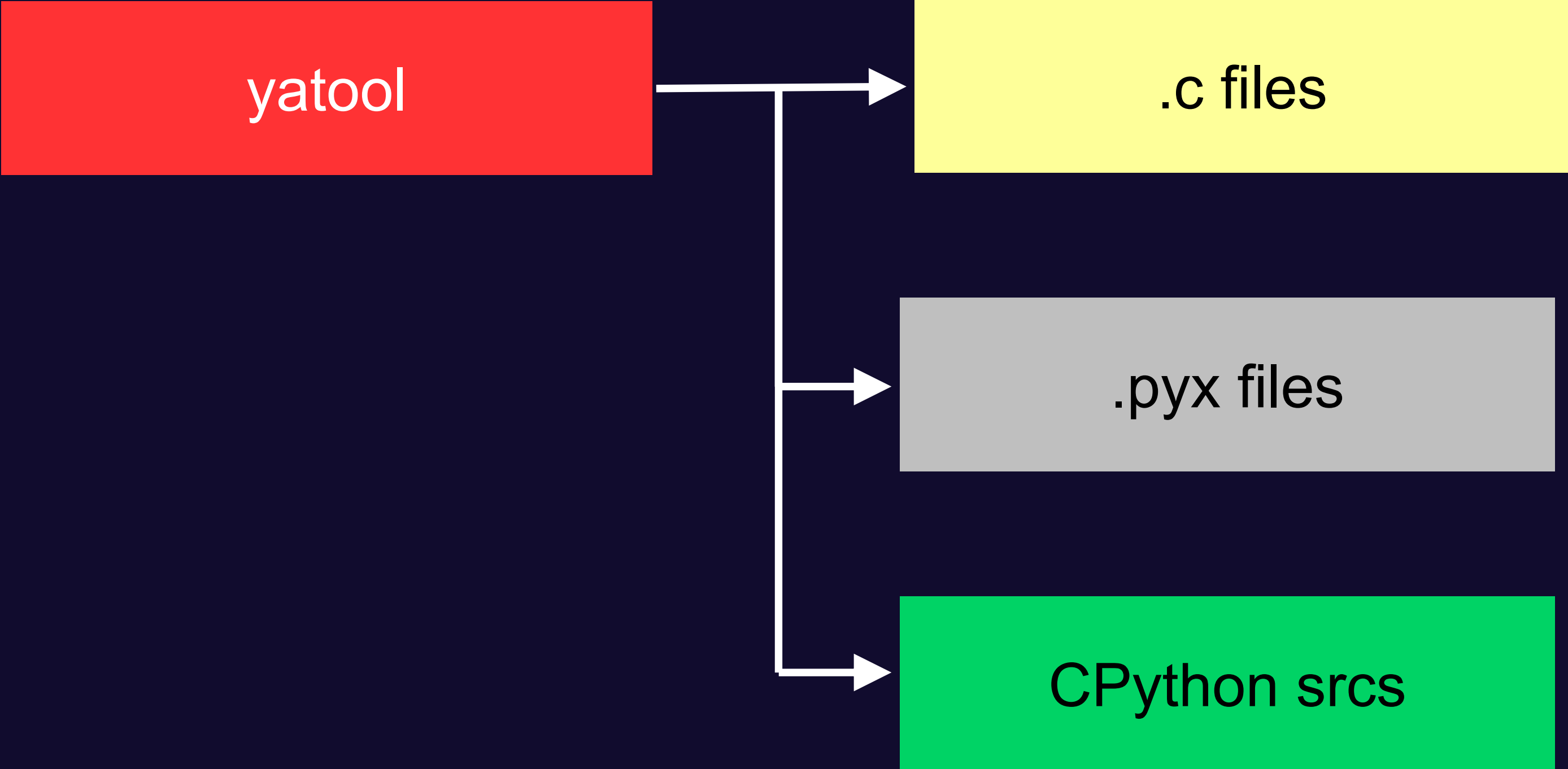
Сборка - yatool



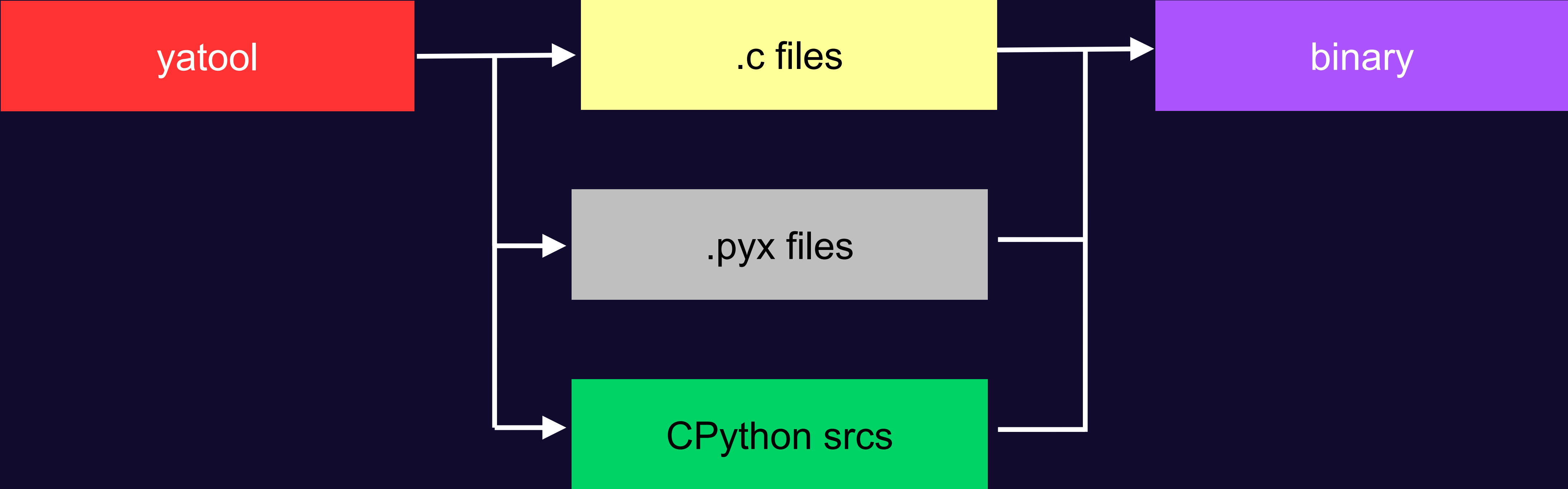
Сборка - yatool



Сборка - yatool



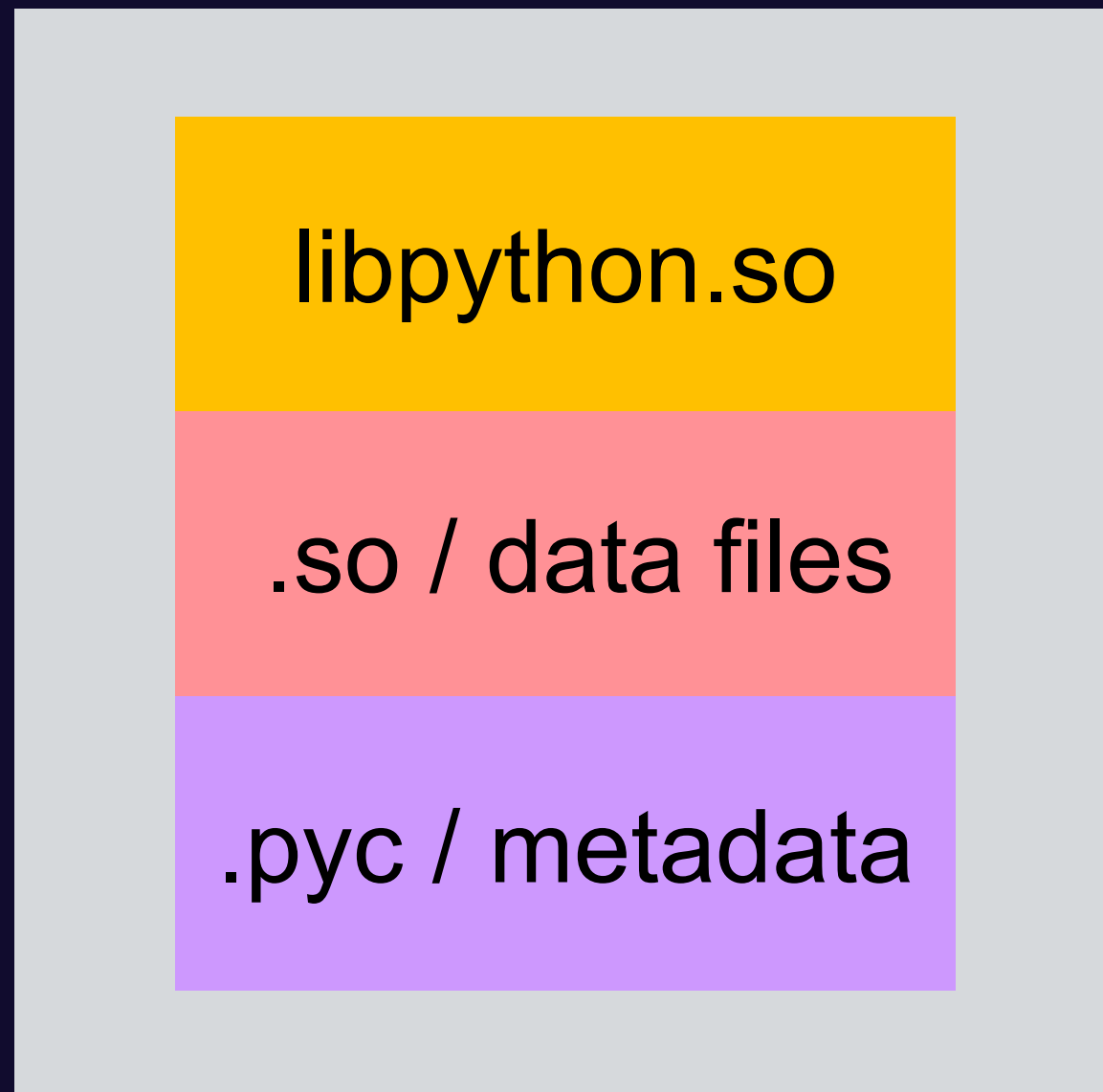
Сборка - yatool



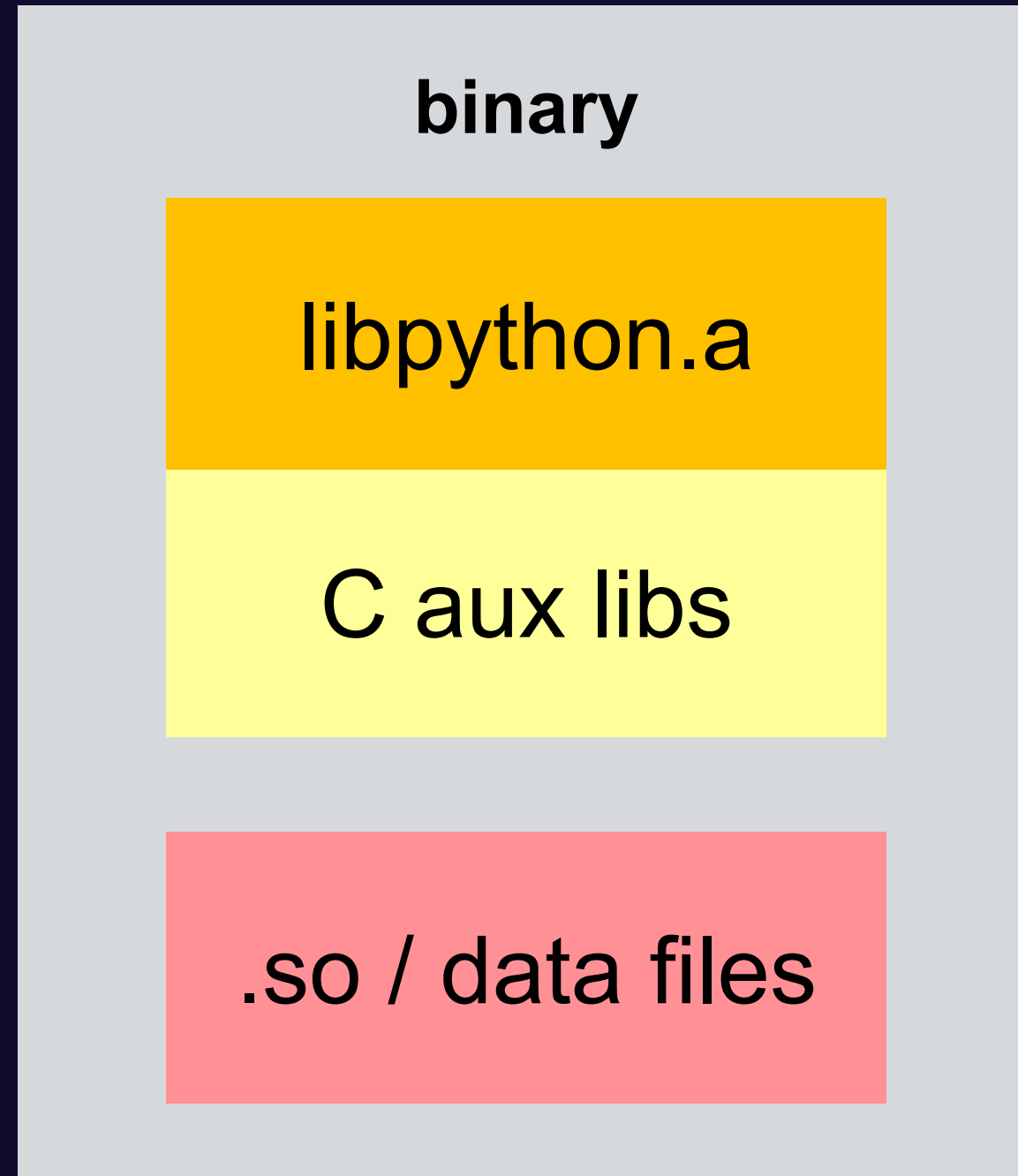
Получаемый артефакт



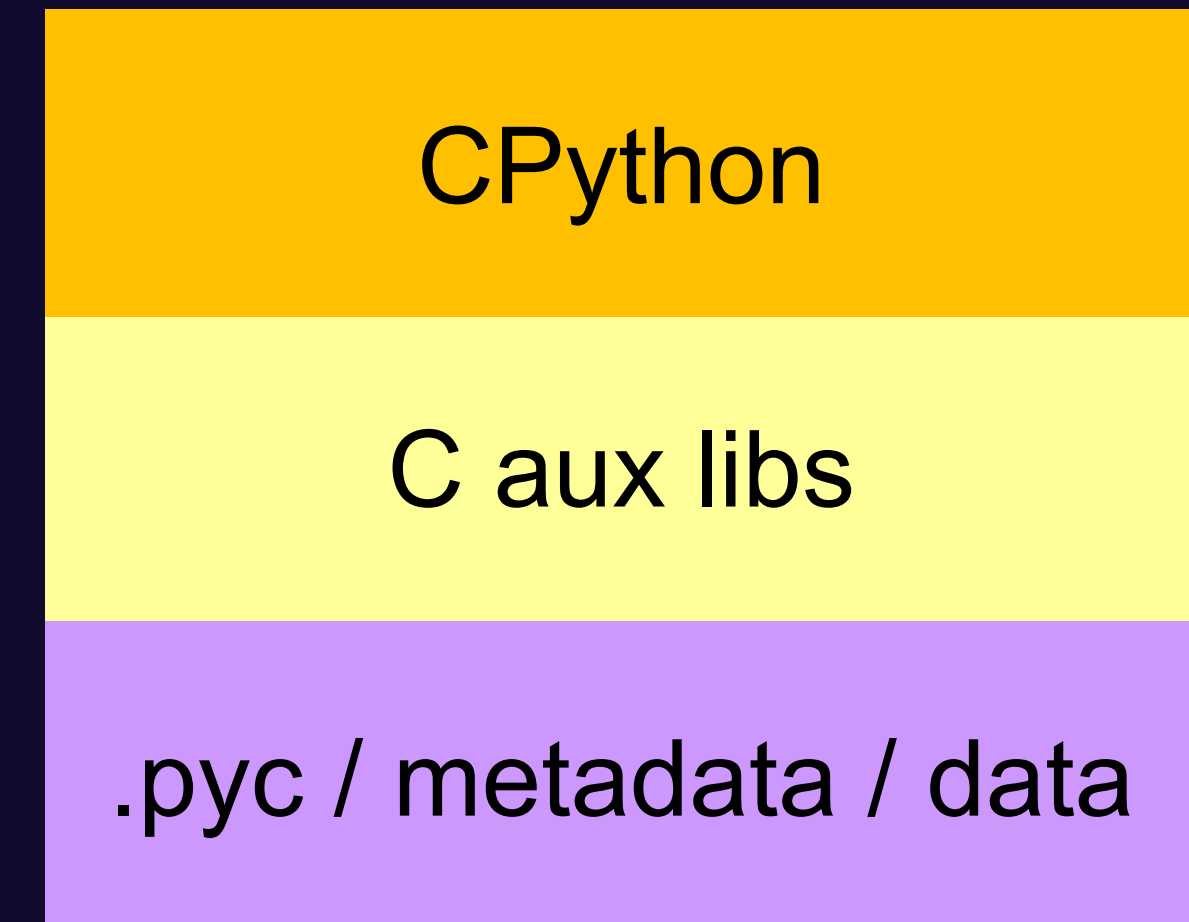
binary



bootstrap



binary



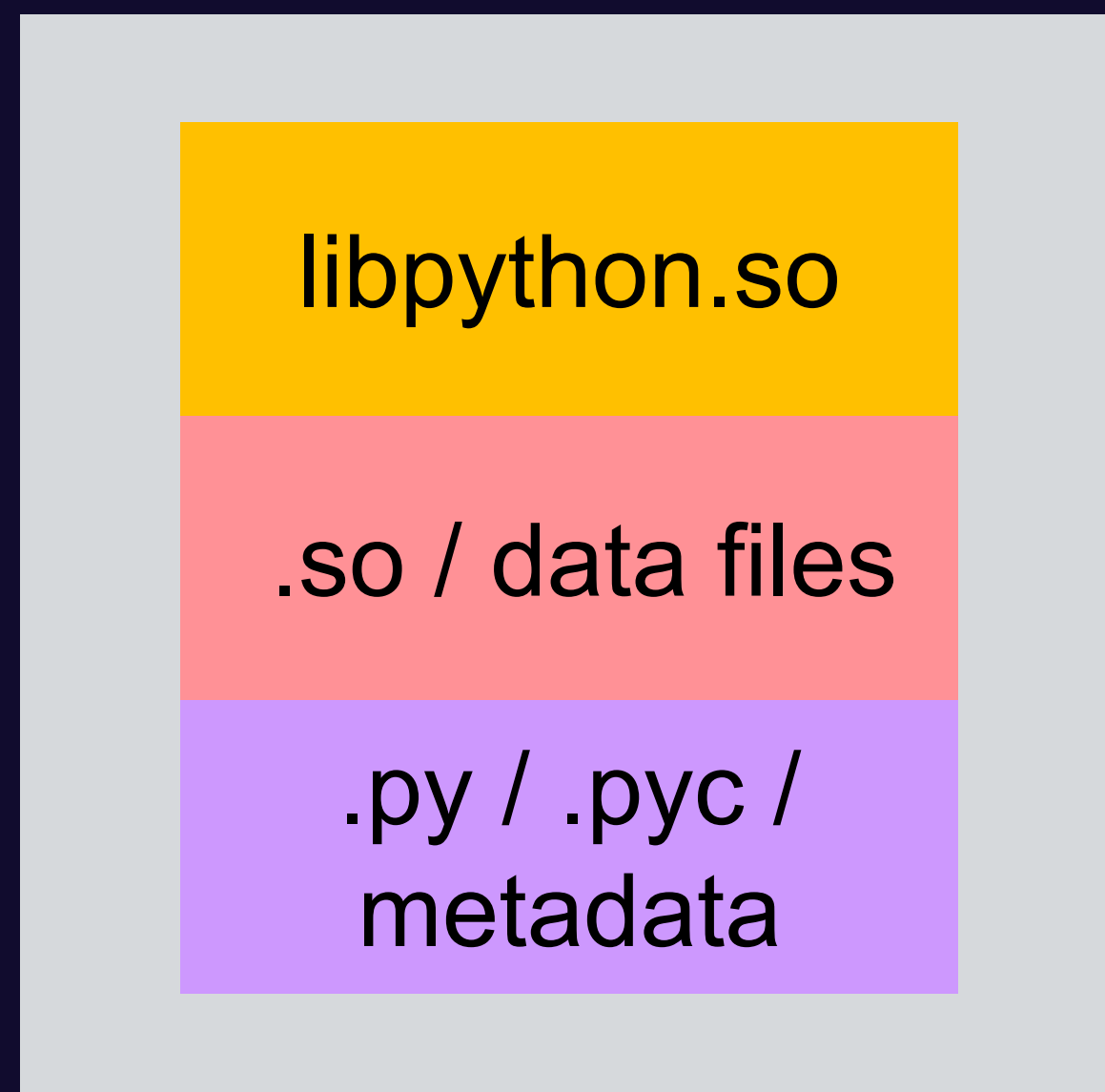


Устройство Runtime

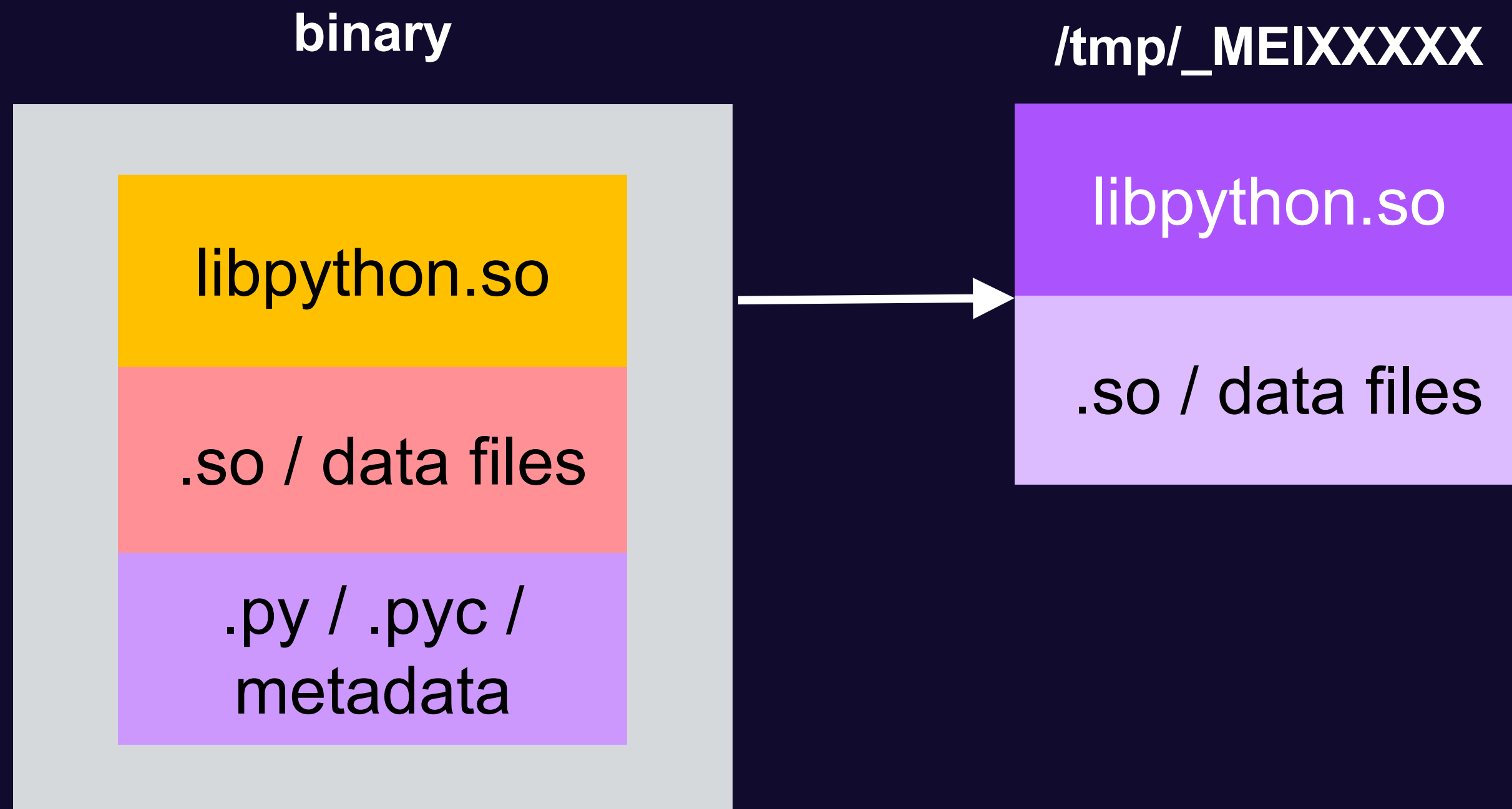
Runtime - PyInstaller



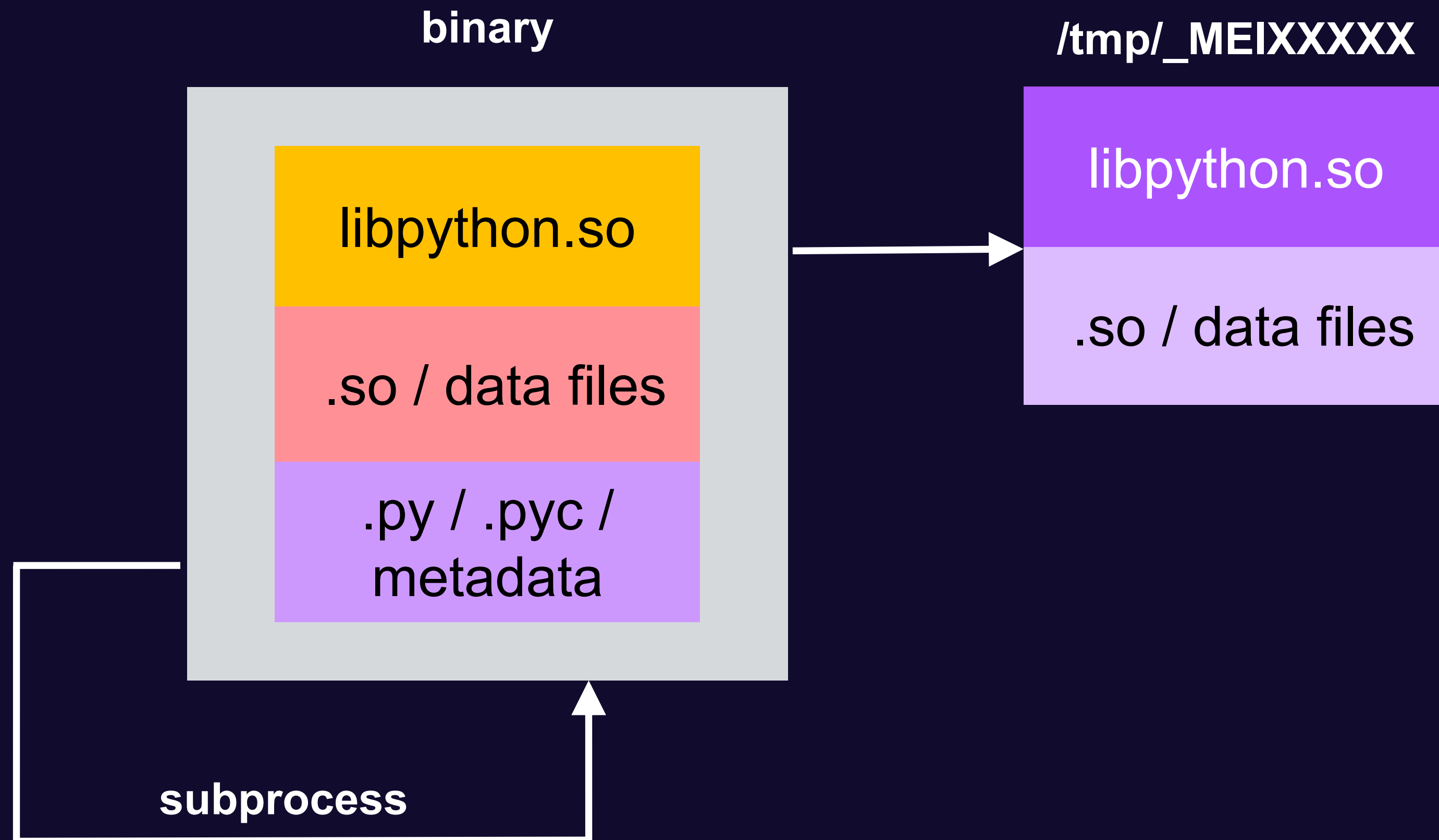
binary



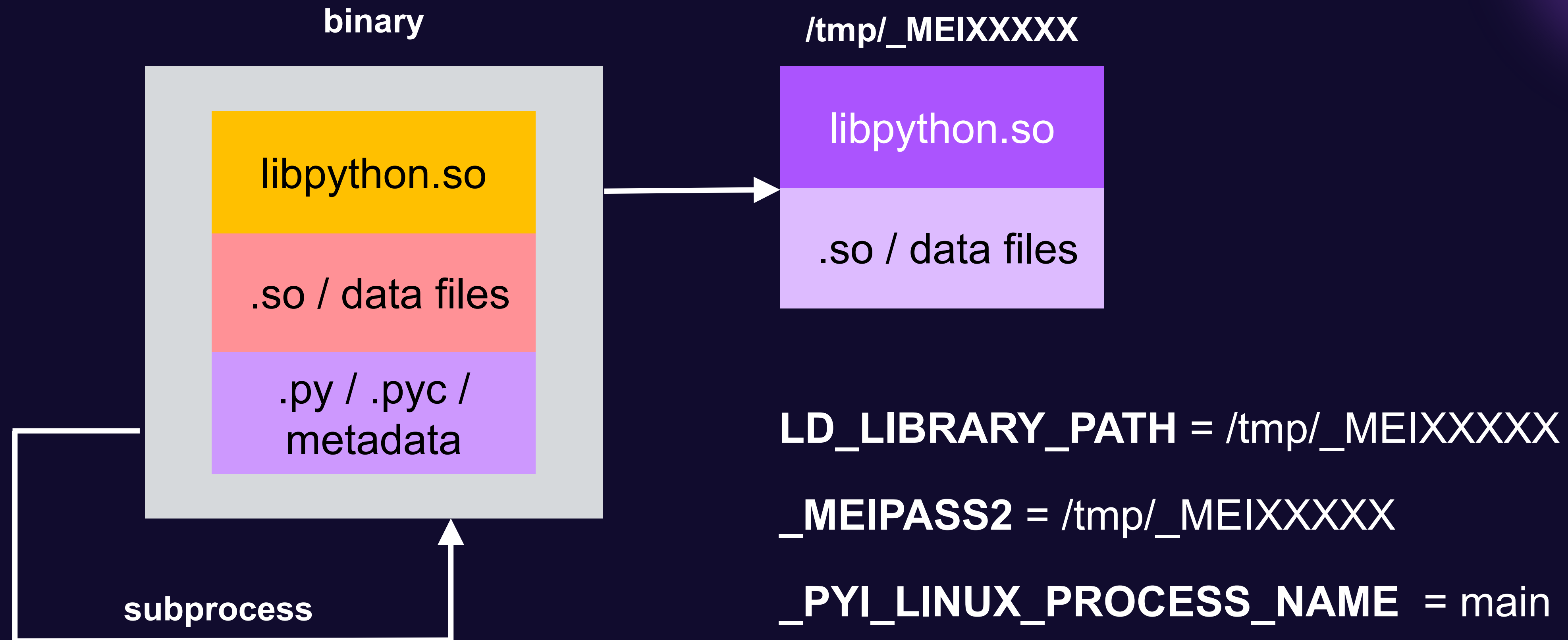
Runtime - PyInstaller



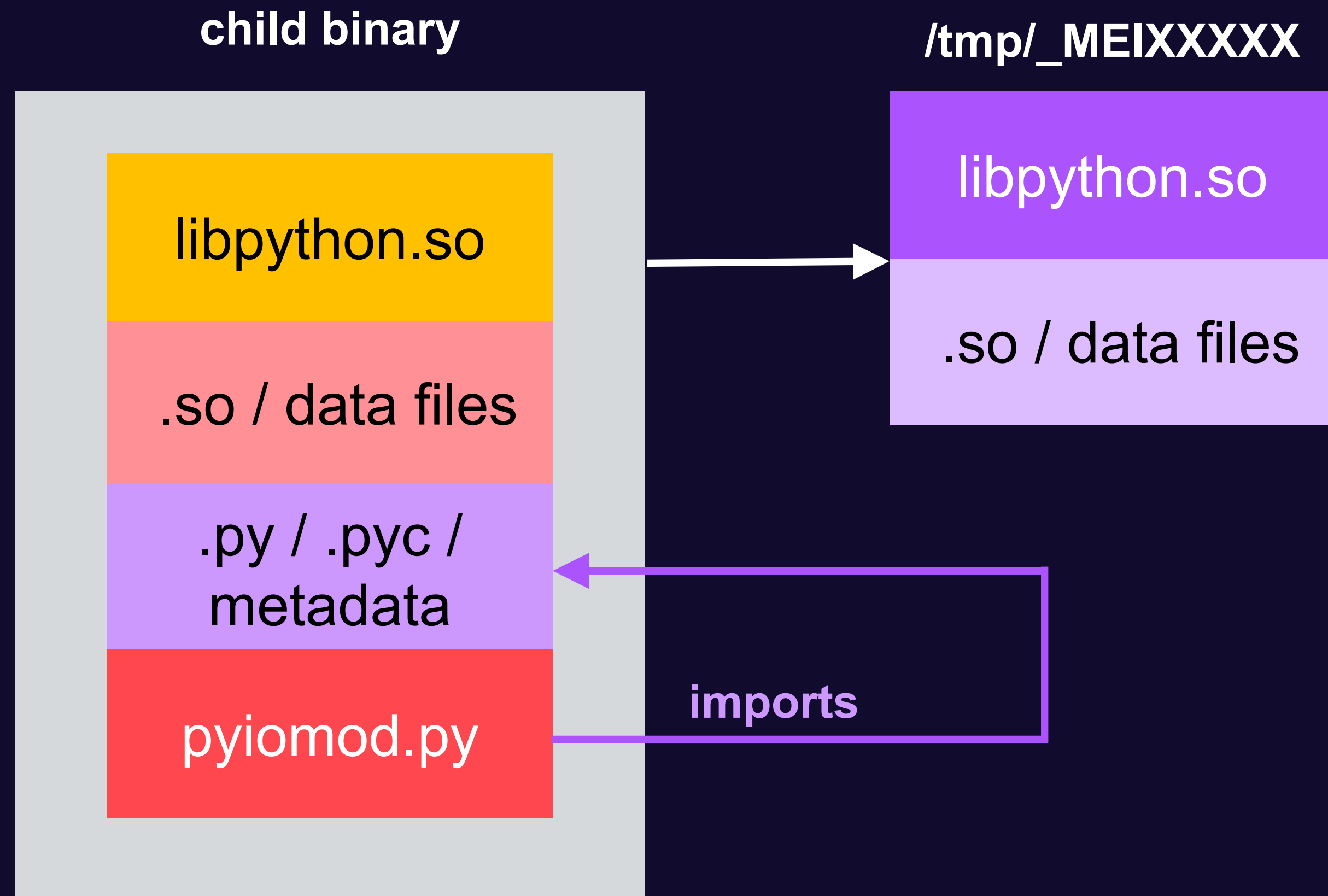
Runtime - PyInstaller



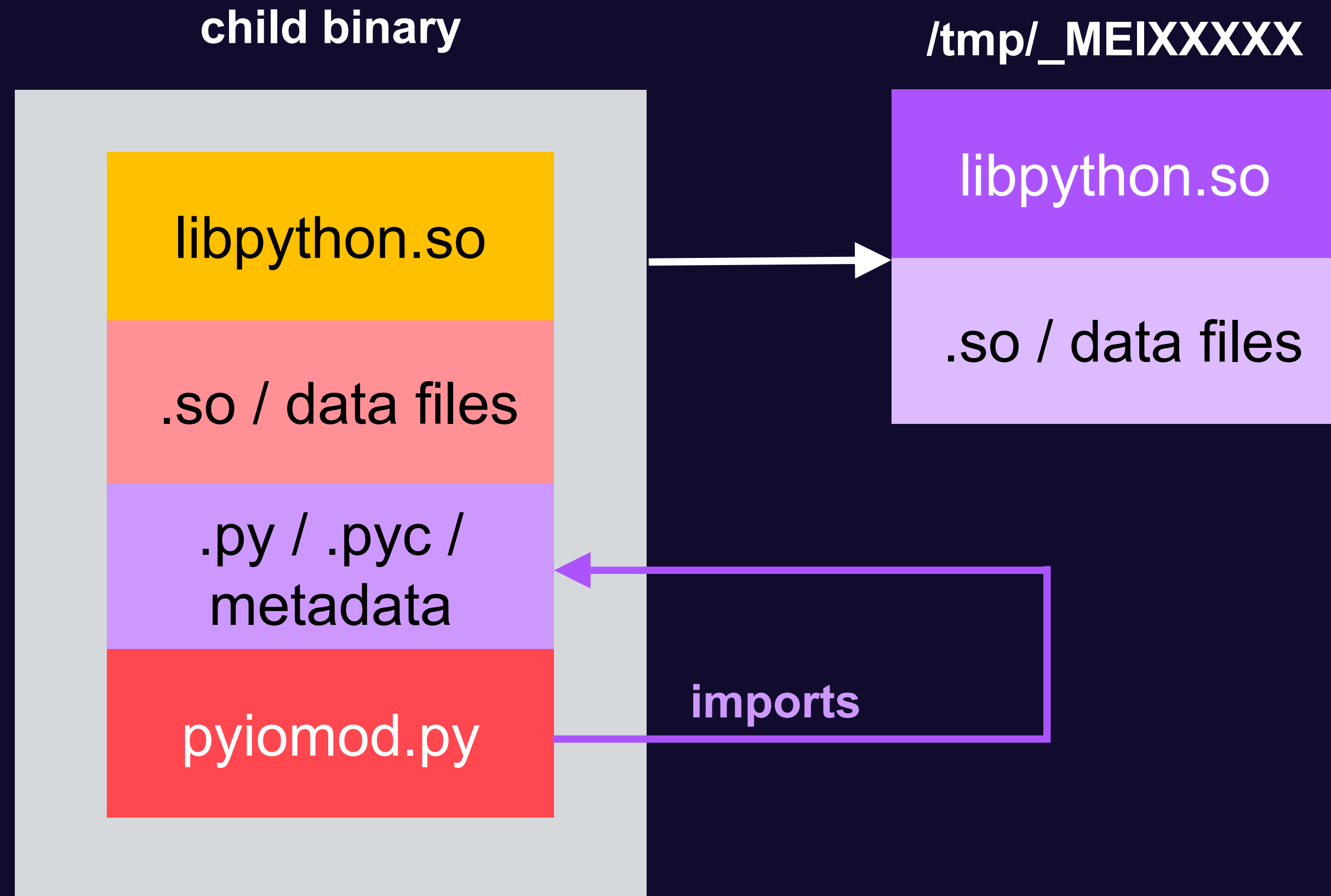
Runtime - PyInstaller



Importing - PyInstaller



Importing - PyInstaller



1. Finds module
2. Decompress
3. Imports

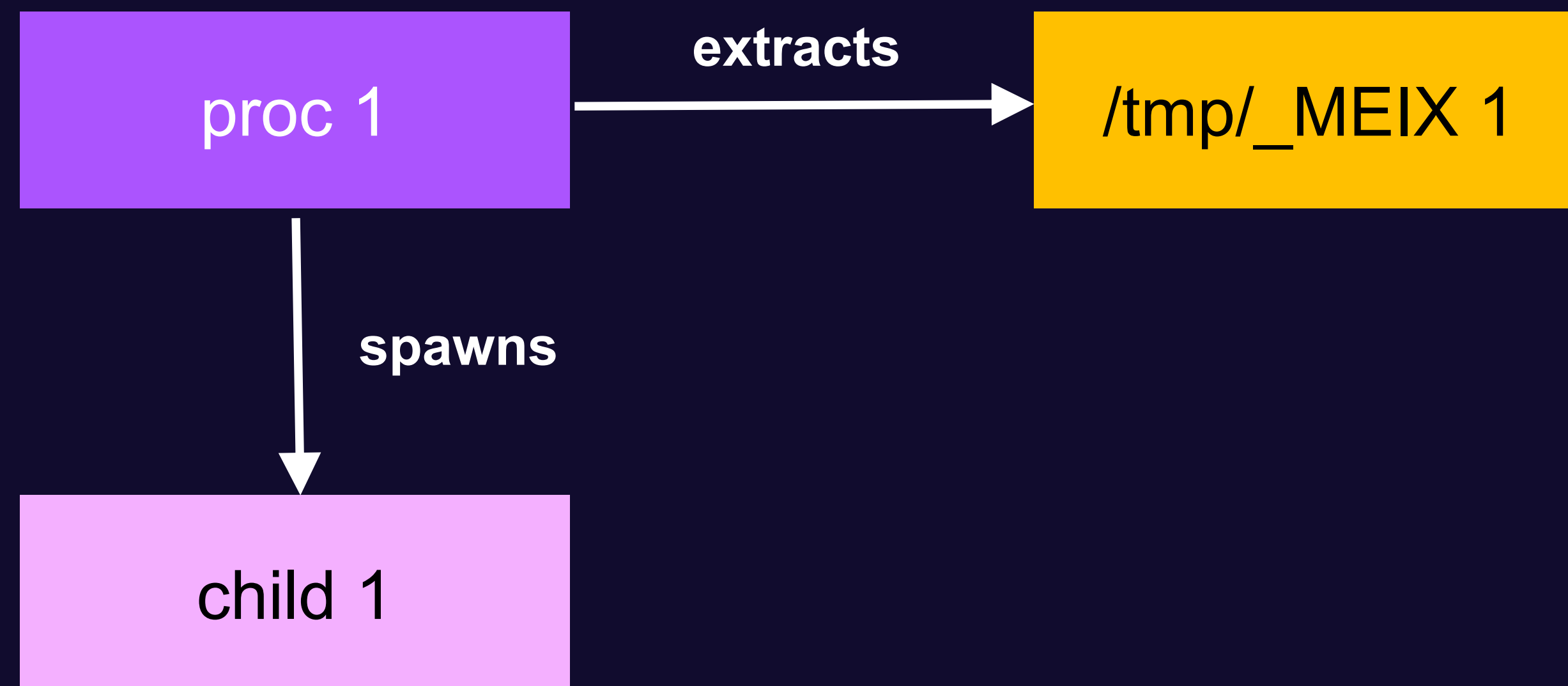


Runtime effects - PyInstaller

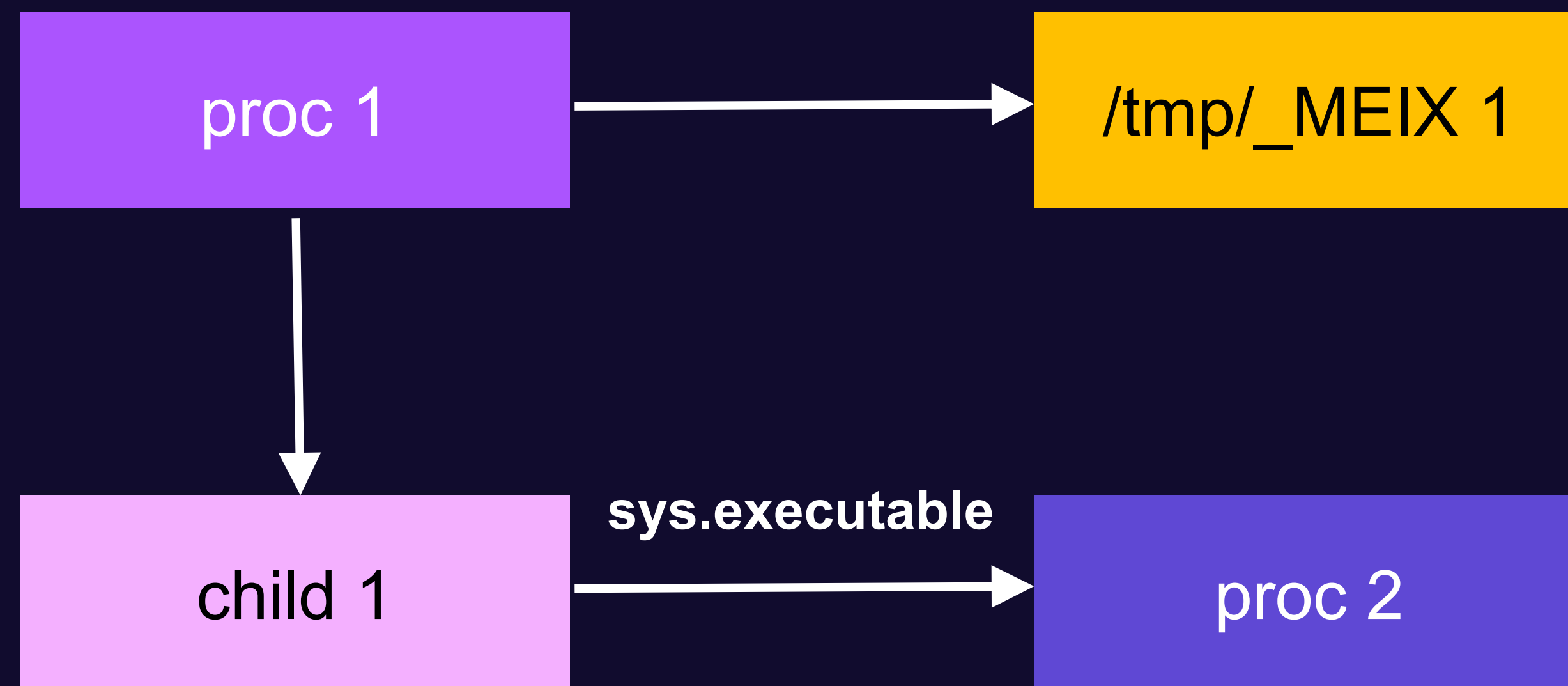
Runtime effects

- 1
Запуск подпроцессов
- 2
Защита данных
- 3
Наличие `python stdlib`
- 4
Интерактивная отладка

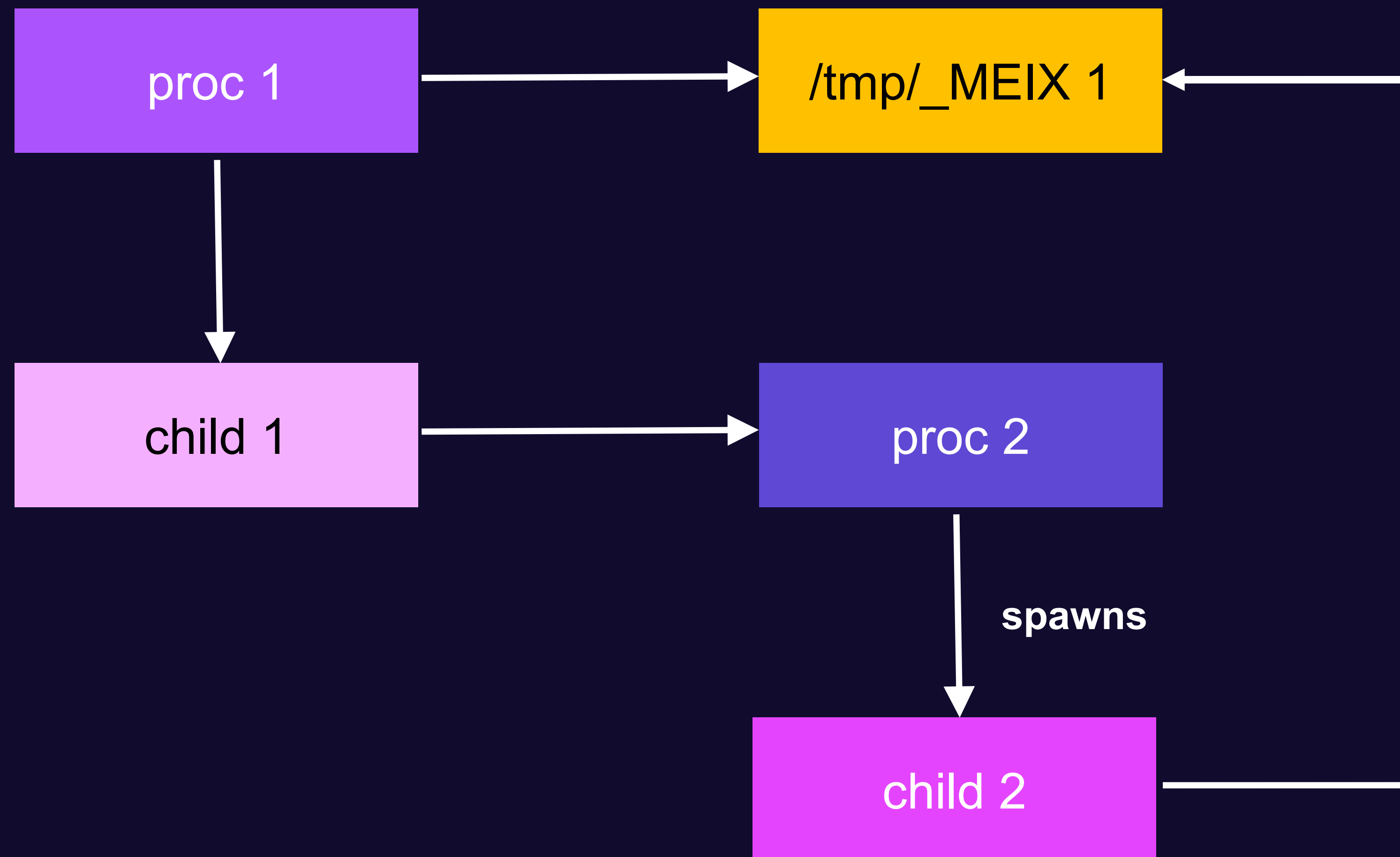
Запуск подпроцессов - PyInstaller



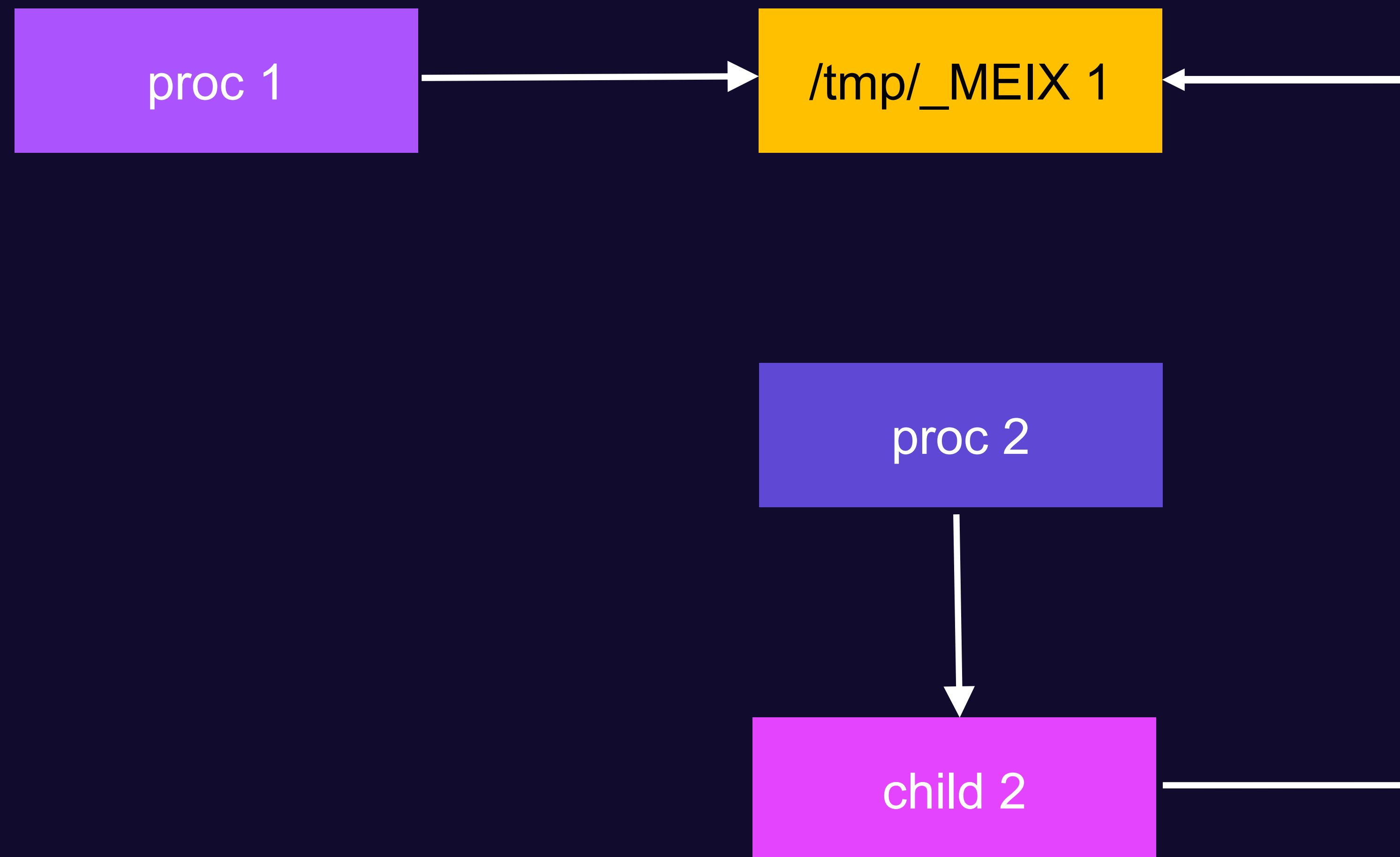
Запуск подпроцессов - PyInstaller



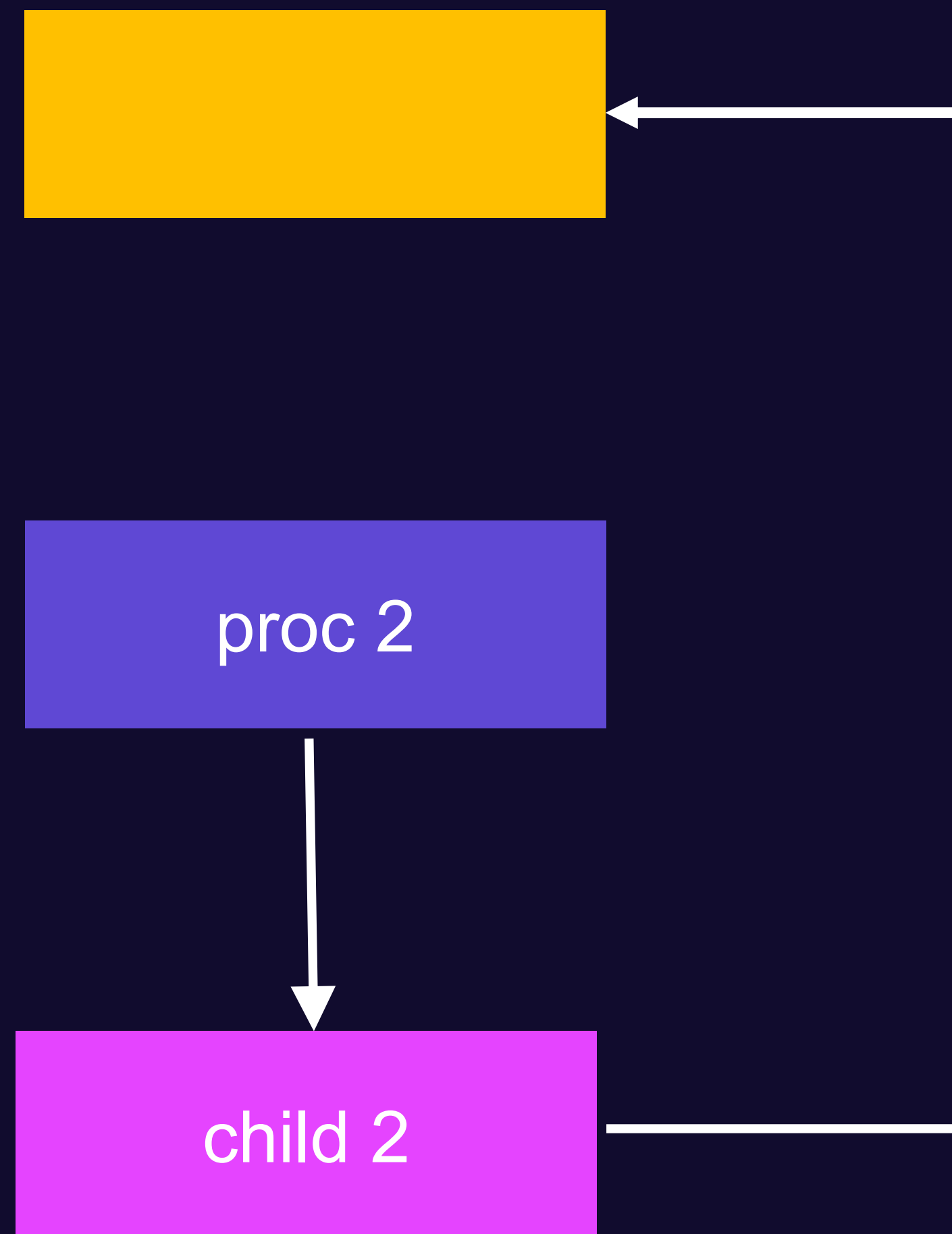
Запуск подпроцессов - PyInstaller



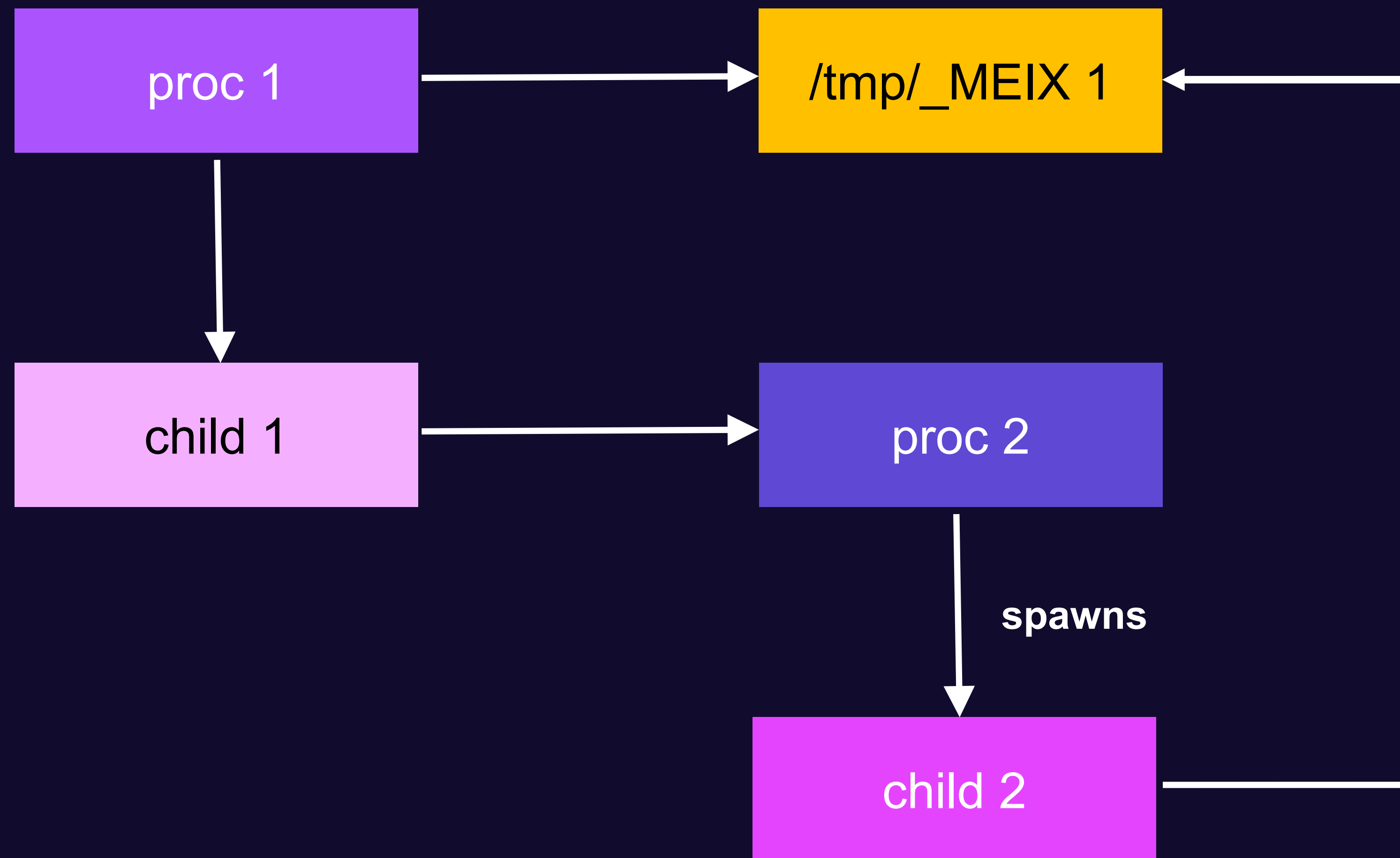
Запуск подпроцессов - PyInstaller



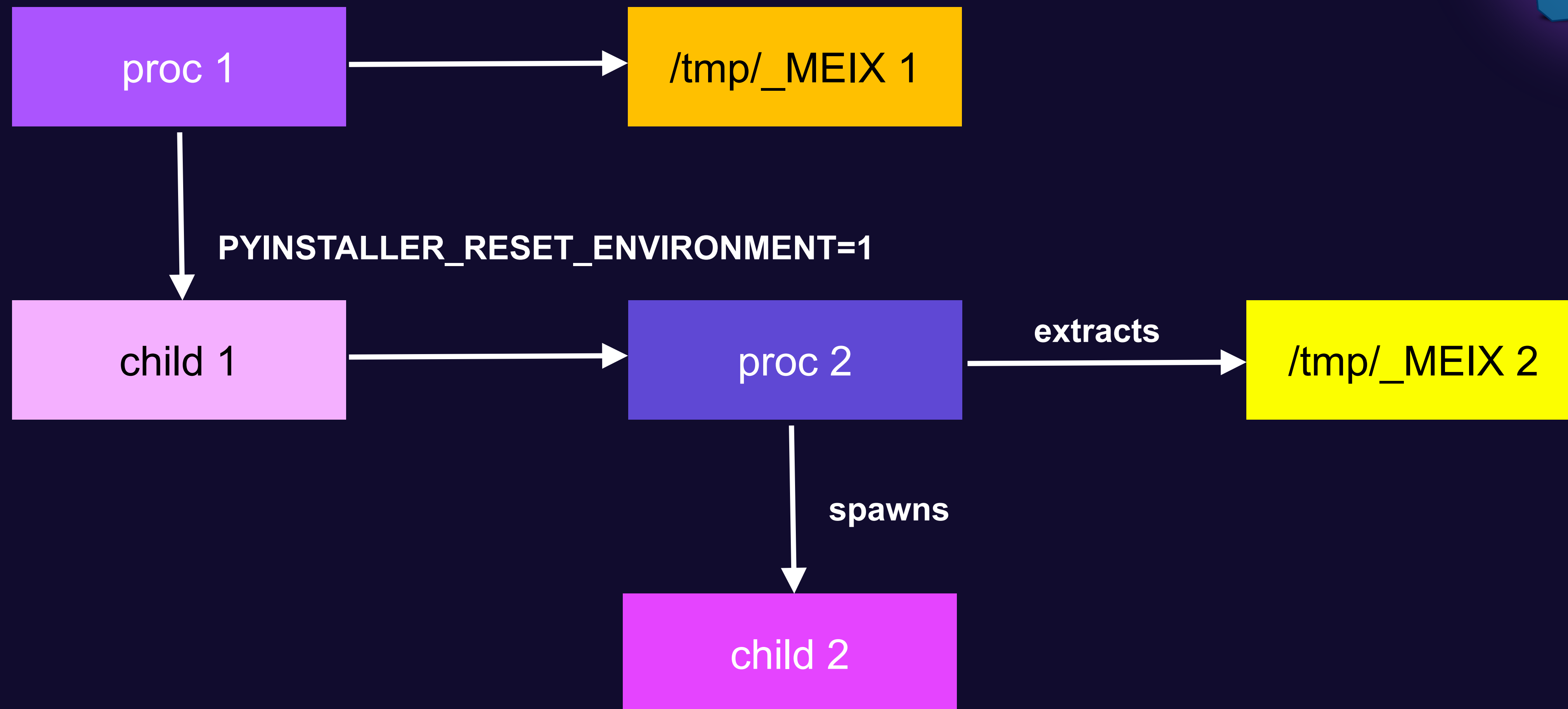
Запуск подпроцессов - PyInstaller



Запуск подпроцессов - PyInstaller



Запуск подпроцессов - PyInstaller



Запуск подпроцессов - PyInstaller



- PYINSTALLER_RESET_ENVIRONMENT режим в 6.9
- multiprocessing.freeze_support()
- Нюансы при работе с сигналами
- Невозможность запуска в read-only среде

Защита данных - PyInstaller



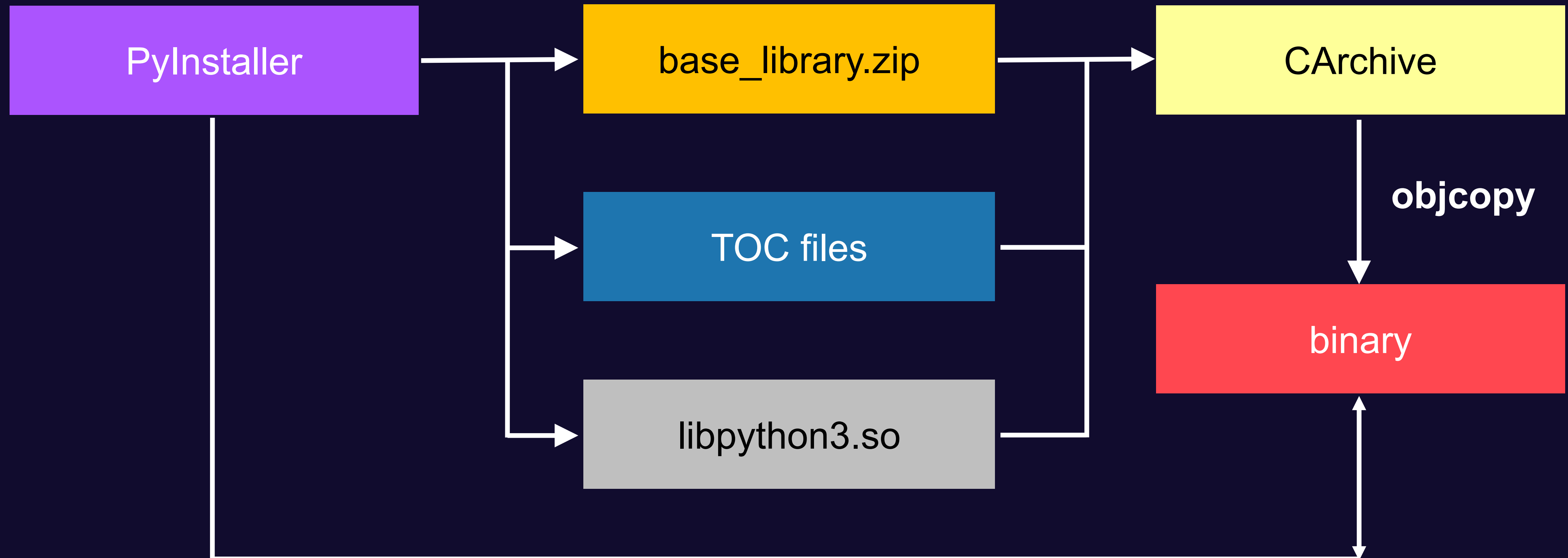
```
linux~> optrace -h -c0 -r5 dist/main
Output tracer summary report (limit: 5)
 10.0MiB /tmp/_MEINbuIzW/supersecret.txt/supersecret.txt
  5.6MiB /tmp/_MEINbuIzW/libpython3.10.so.1.0
  4.2MiB /tmp/_MEINbuIzW/libcrypto.so.3
 859KiB /tmp/_MEINbuIzW/base_library.zip
266KiB /tmp/_MEINbuIzW/lib-dynload/_codecs_jp.cpython-310-x86_64-linux-gnu.so
Total output: 22.7MiB
```

Защита данных - PyInstaller



```
linux~> optrace -h -c0 -r5 dist/main
Output tracer summary report (limit: 5)
10.0MiB /tmp/_MEINbuIzW/supersecret.txt/supersecret.txt
5.6MiB /tmp/_MEINbuIzW/libpython3.10.so.1.0
4.2MiB /tmp/_MEINbuIzW/libcrypto.so.3
859KiB /tmp/_MEINbuIzW/base_library.zip
266KiB /tmp/_MEINbuIzW/lib-dynload/_codecs_jp.cpython-310-x86_64-linux-gnu.so
Total output: 22.7MiB
```


Наличие python stdlib - PyInstaller



Интерактивная отладка - PyInstaller



```
import pdb
pdb.set_trace()
# breakpoint()
```

Интерактивная отладка - PyInstaller



```
→ module = importlib.import_module(hashlib_name)
(Pdb) l
5
6
7     def main():
8         pdb.set_trace()
9     →     module = importlib.import_module(hashlib_name)
10         md5 = module.md5()
11
12
13     if __name__ == "__main__":
14         main()
```

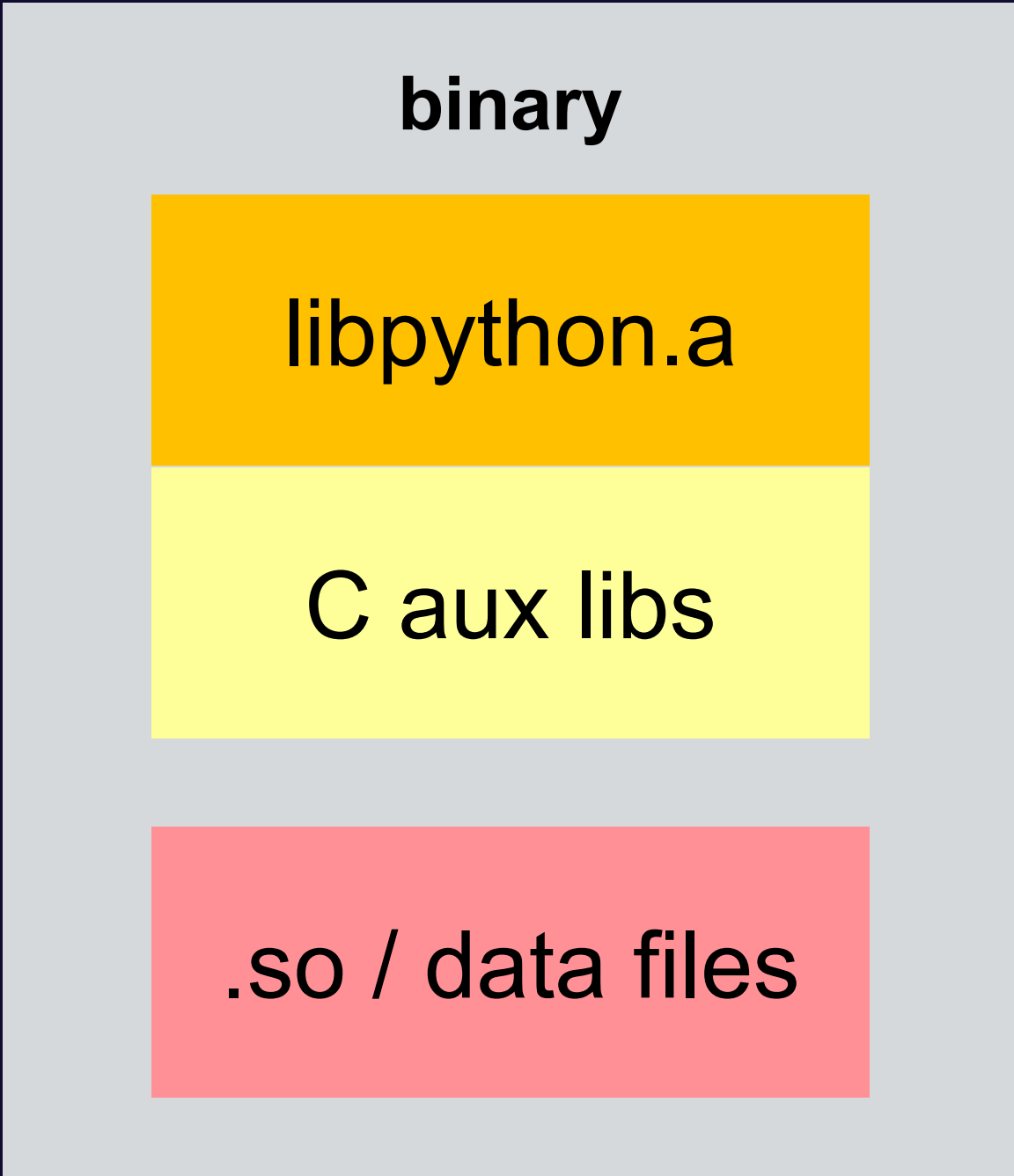


Runtime - Nuitka

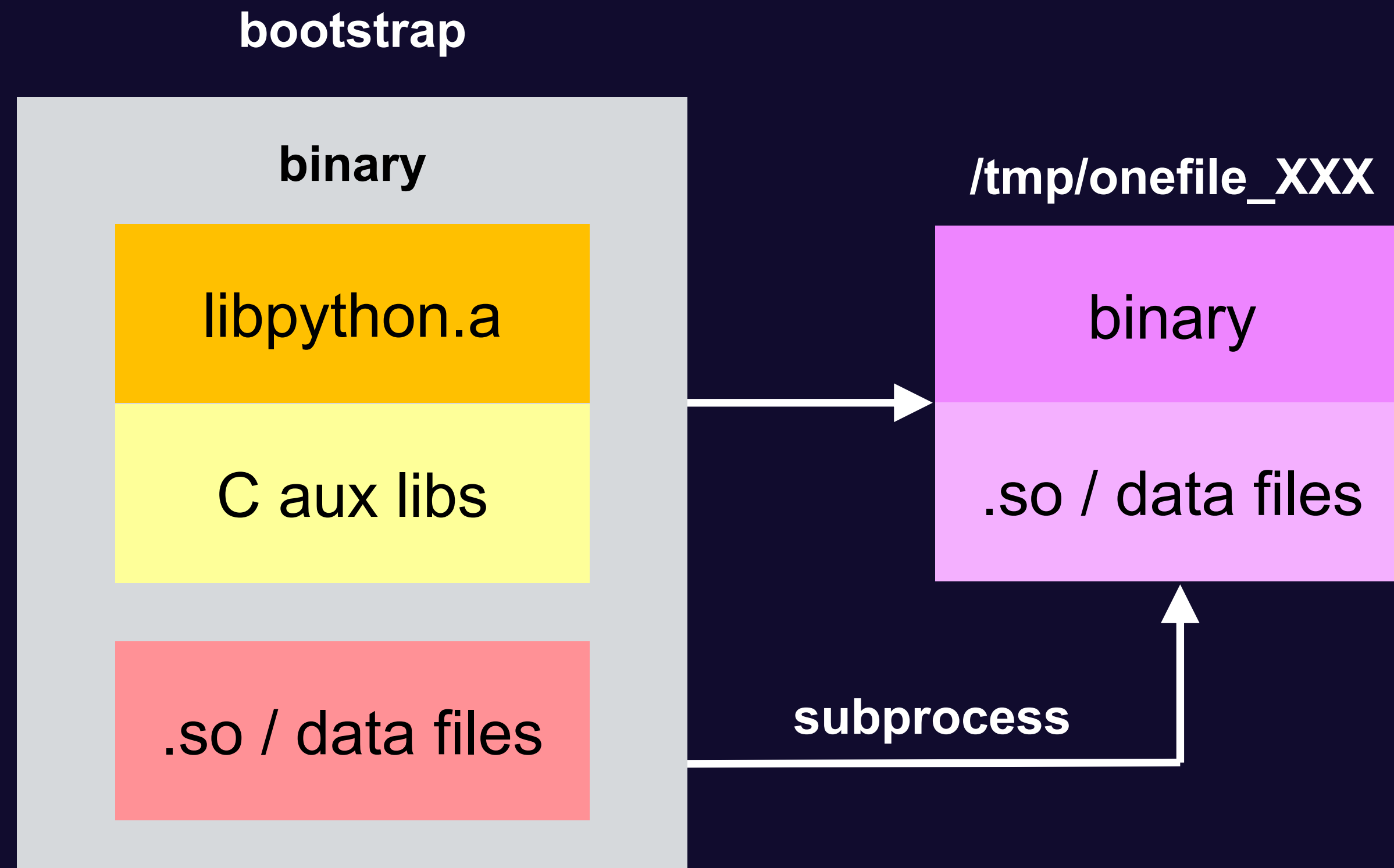
Runtime - Nuitka



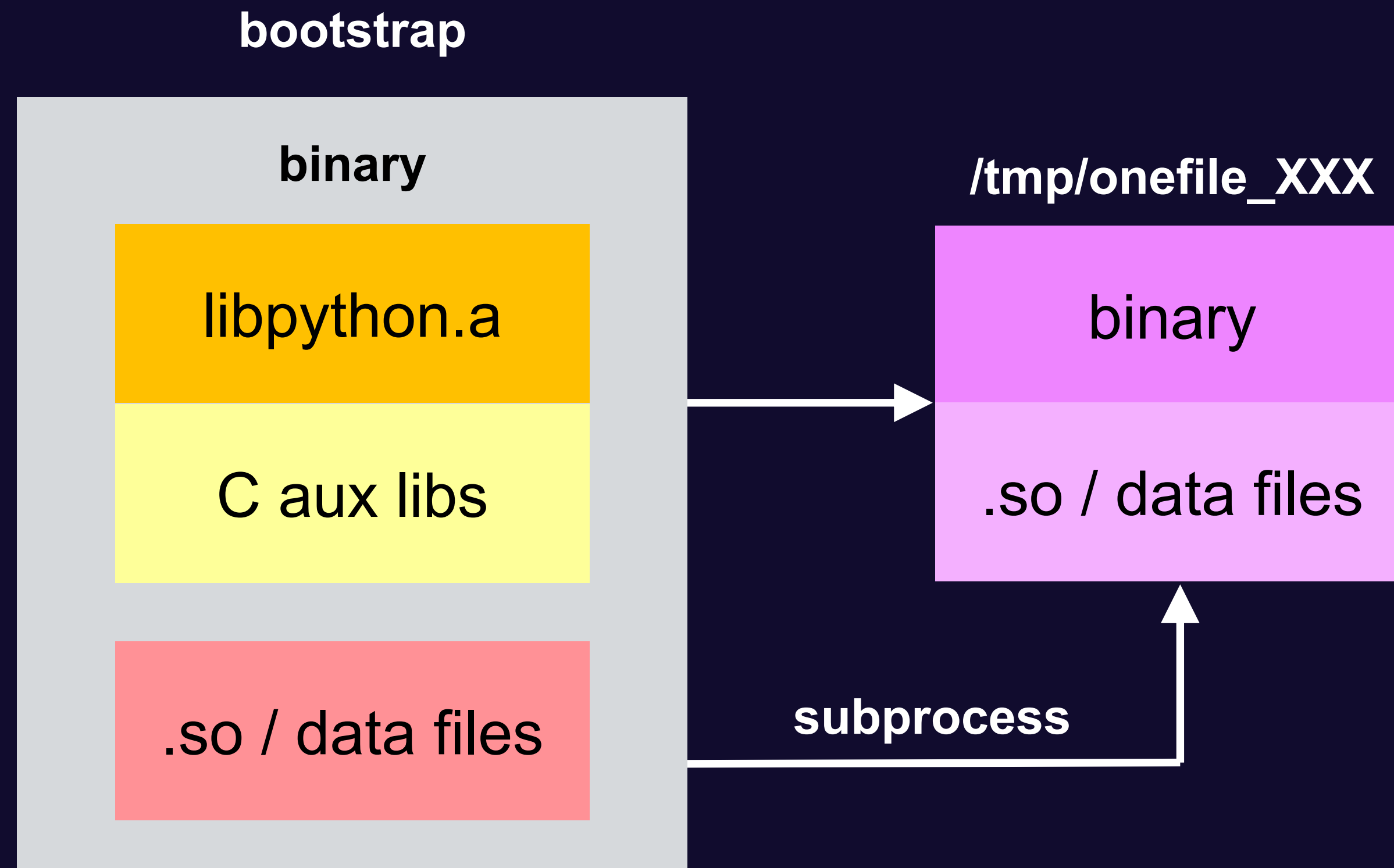
bootstrap



Runtime - Nuitka



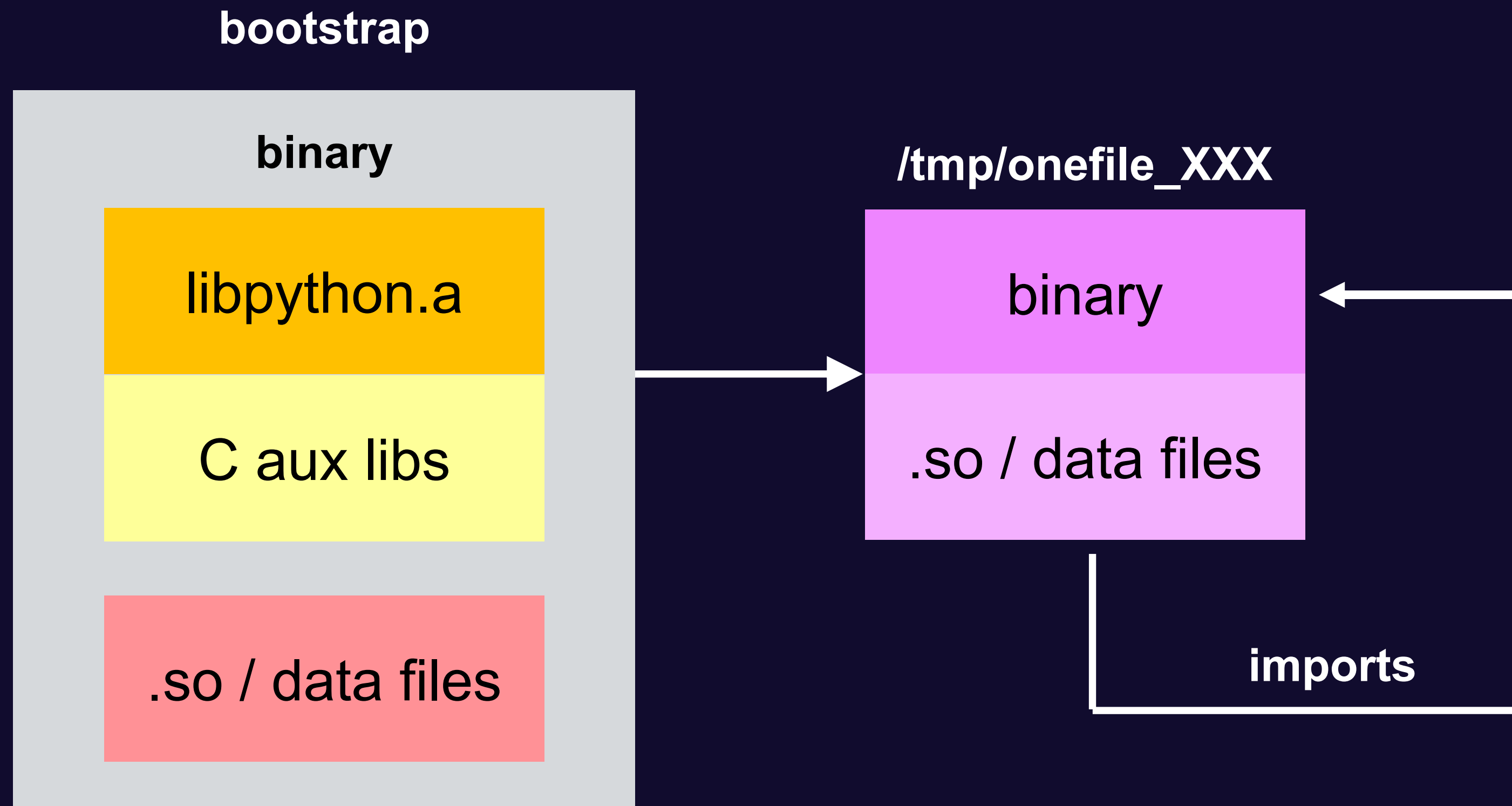
Runtime - Nuitka



NUITKA_ONEFILE_BINARY = <path>

NUITKA_ONEFILE_PARENT = <pid>

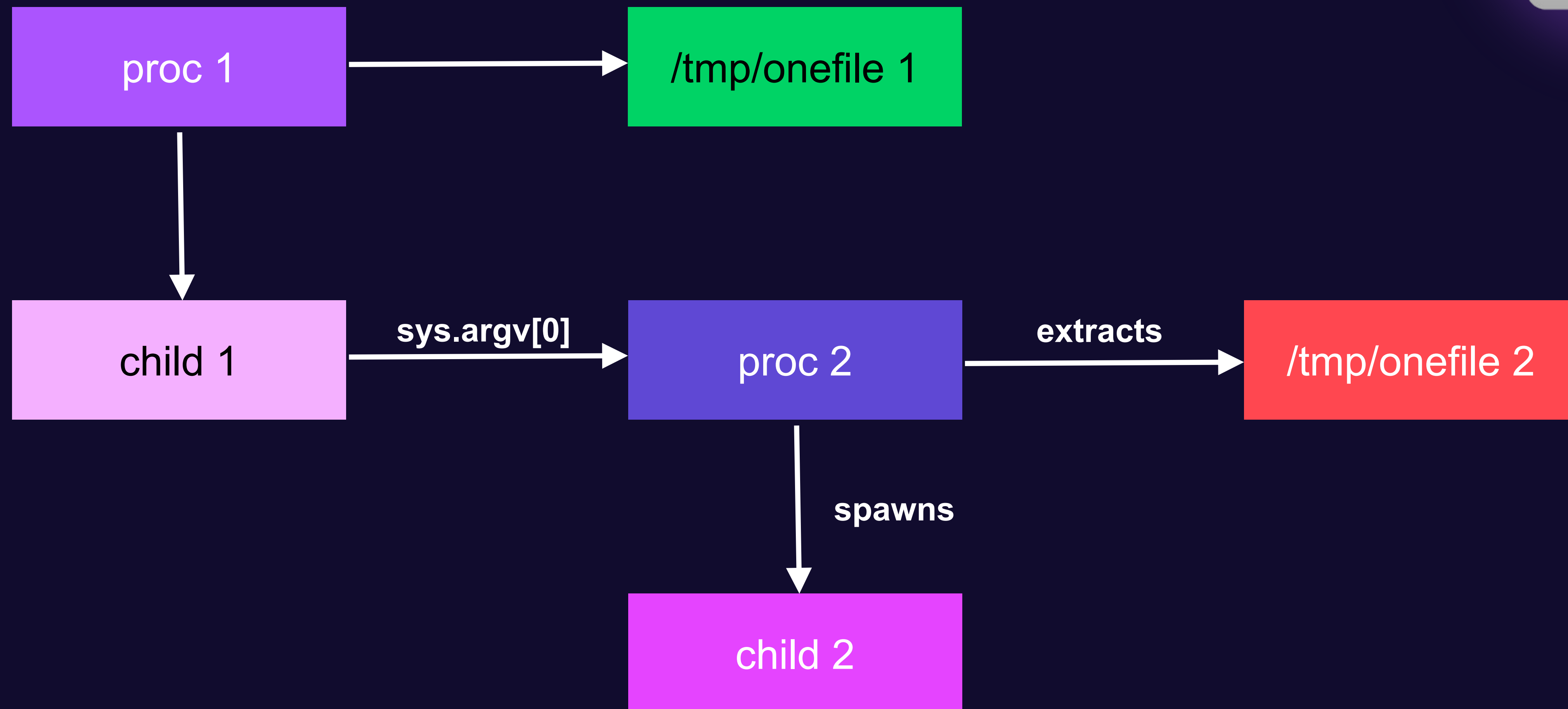
Importing - Nuitka





Runtime effects - Nuitka

Запуск подпроцессов - Nuitka



Запуск подпроцессов - Nuitka



- Стараются поддерживать multiprocessing
- Нюансы при работе с сигналами
- Невозможность запуска в read-only среде

Защита данных - Nuitka



```
linux ~$ optrace -h -c0 -r5 ./main.bin
Output tracer summary report (limit: 5)
 10.0MiB /tmp/onefile_145920_1726242667_152599/supersecret.txt
   9.0MiB /tmp/onefile_145920_1726242667_152599/main.bin
 269KiB /tmp/onefile_145920_1726242667_152599/_codecs_jp.so
 195KiB /tmp/onefile_145920_1726242667_152599/libexpat.so.1
 161KiB /tmp/onefile_145920_1726242667_152599/_codecs_hk.so
Total output: 20.1MiB
```

Защита данных - Nuitka



```
def get_secret():  
    return "pass:i_love_piterpy"
```


Защита данных - Nuitka



```
linux ~$ strings main.bin | grep 'pass:'
```

Защита данных - Nuitka



```
linux ~ → strace -e inject=unlink:retval=0 ./main.bin
```


Защита данных - Nuitka



```
linux ~$ strings /tmp/onefile_144484_1726242172_796925/main.bin | grep 'pass:'  
upass:i_love_piterpy
```

Наличие python stdlib - Nuitka



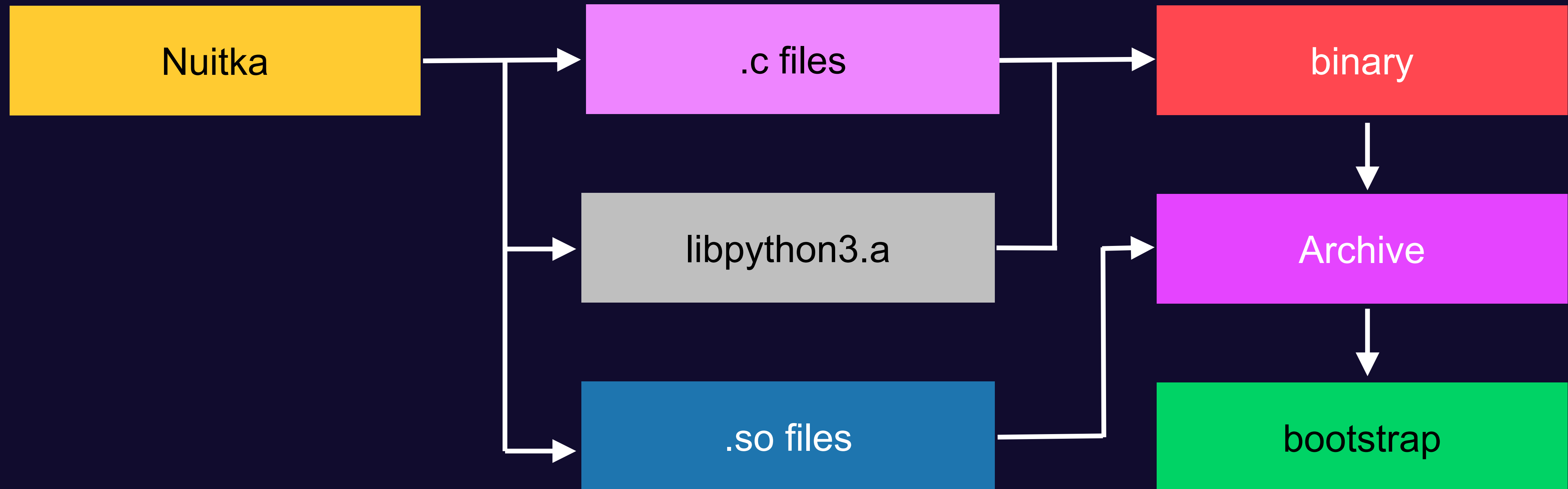
```
module = importlib.import_module(hashlib_str)
return = module.md5()
```

Наличие python stdlib - Nuitka



```
Traceback (most recent call last):
  File "/tmp/onefile_147313_1726242982_961326/main.py", line 20, in <module>
    main()
  File "/tmp/onefile_147313_1726242982_961326/main.py", line 15, in main
    module = importlib.import_module(hashlib_name)
  File "importlib.py", line 126, in import_module
  File "<frozen importlib._bootstrap>", line 1050, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1004, in _find_and_load_unlocked
ModuleNotFoundError: No module named 'hashlib'
```

Наличие python stdlib - Nuitka



Интерактивная отладка - Nuitka



```
import pdb
pdb.set_trace()
# breakpoint()
```

Интерактивная отладка - Nuitka



```
Traceback (most recent call last):
  File "/tmp/test-nuitka/main.dist/main.py", line 15, in <module>
  File "/tmp/test-nuitka/main.dist/main.py", line 9, in main
  File "/tmp/test-nuitka/main.dist/pdb.py", line 1623, in set_trace
  File "/tmp/test-nuitka/main.dist/bdb.py", line 332, in set_trace
RuntimeError: f_trace is not writable in Nuitka
```




Runtime - yatool

Runtime - yatool



binary

CPython

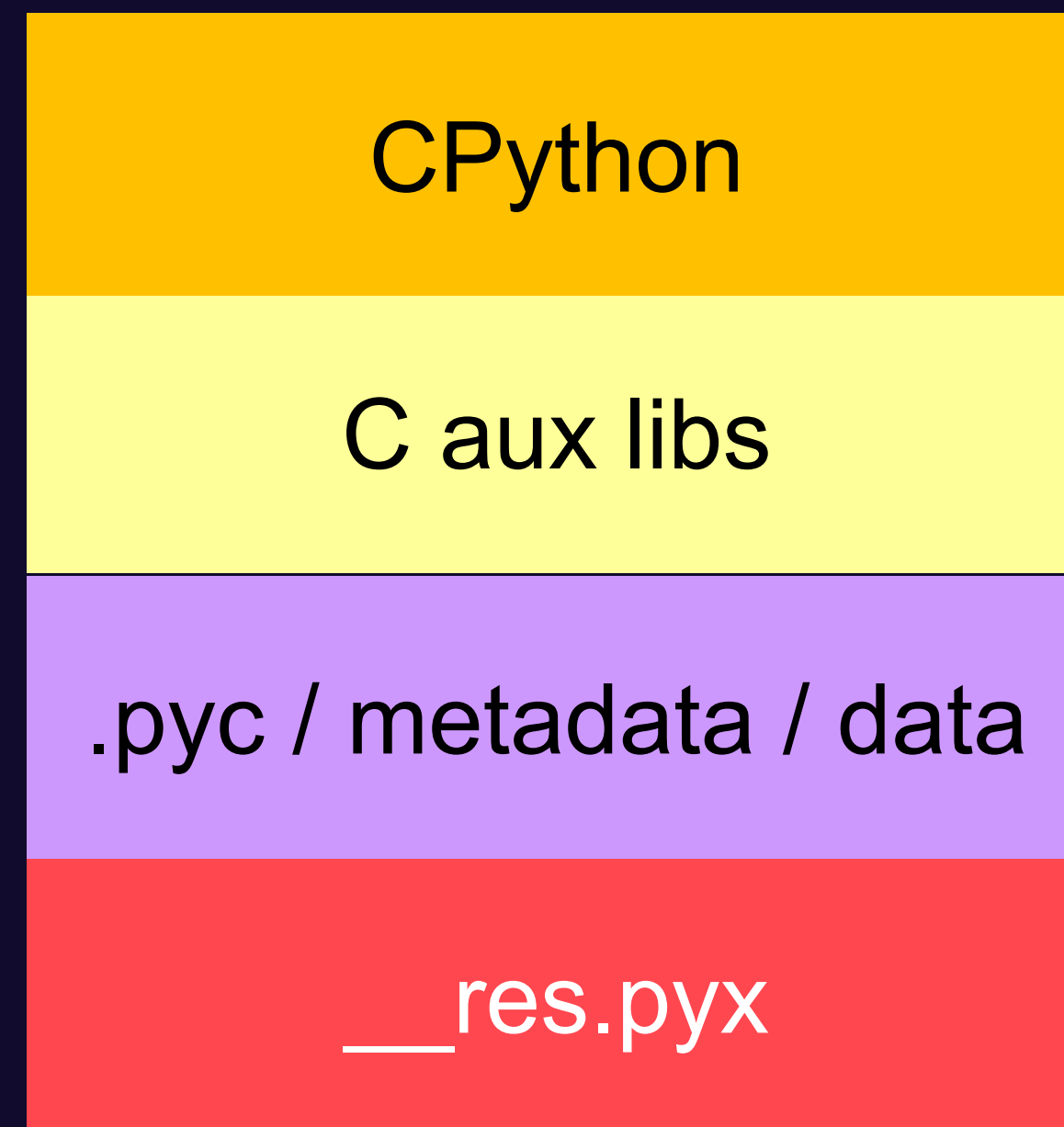
C aux libs

.pyc / metadata / data

Importing - yatool



binary



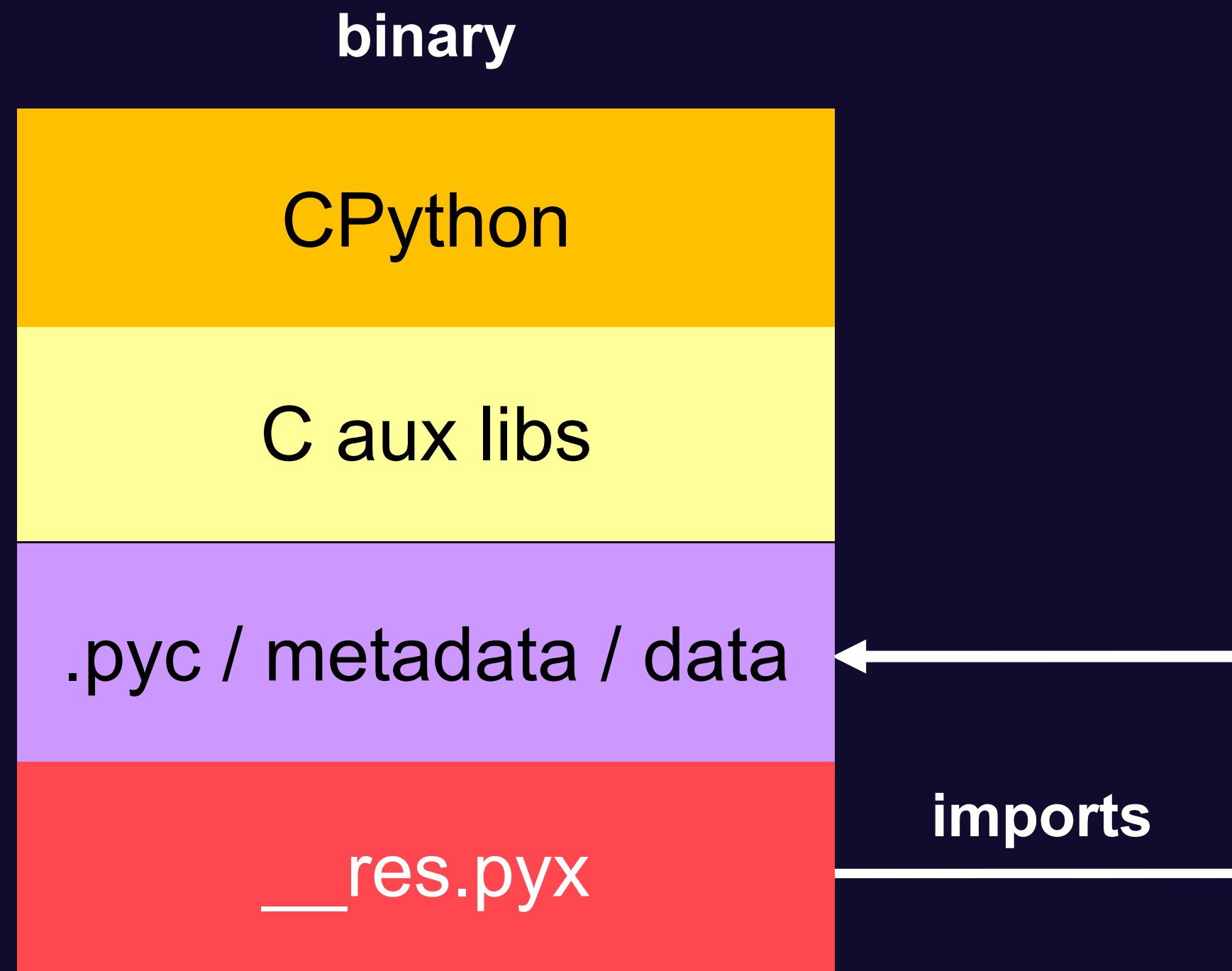
CPython

C aux libs

.pyc / metadata / data

`__res.pyx`

Importing - yatool

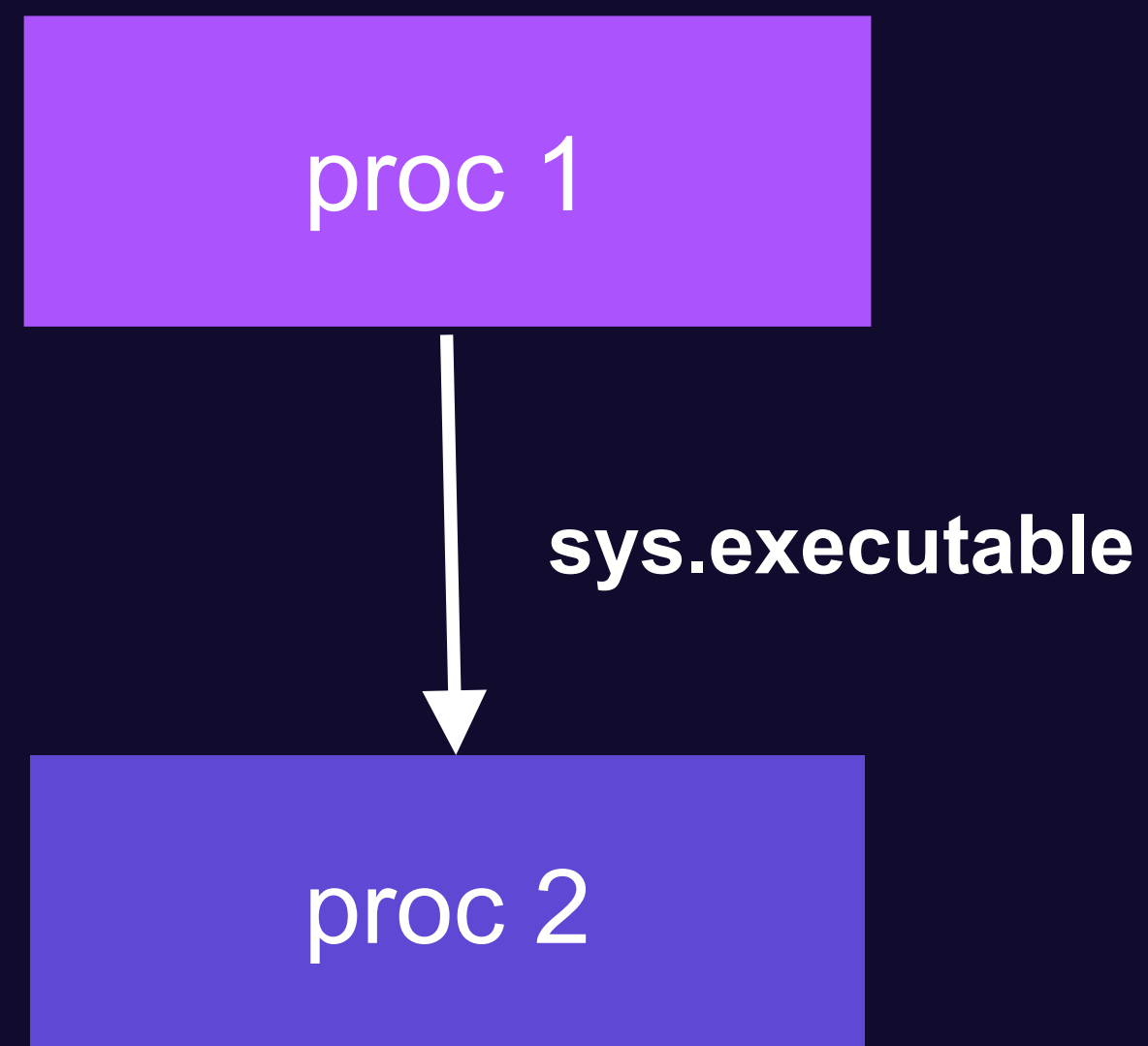


1. Finds module
2. Decompress
3. Imports



Runtime effects - yatool

Запуск подпроцессов - yatool



Защита данных - yatool



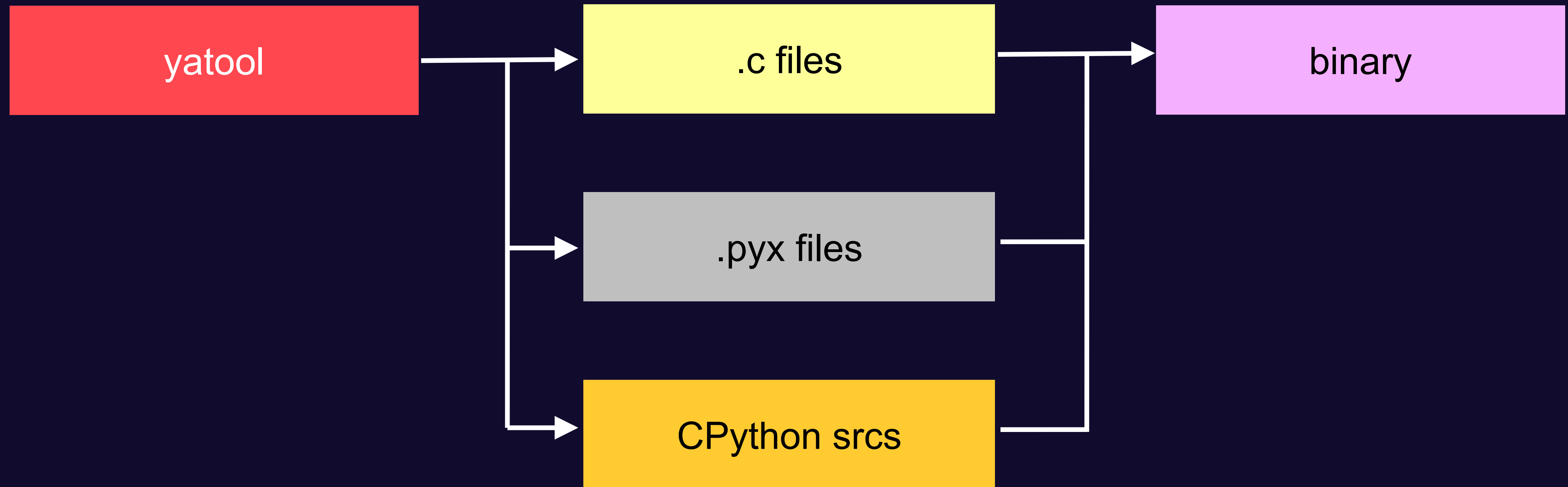
binary

CPython

C aux libs

.pyc / metadata / data

Наличие python stdlib - yatool



Интерактивная отладка - yatoool



```
import pdb
pdb.set_trace()
# breakpoint()
```

Интерактивная отладка - yatoool



```
linux → ./binary
> /tmp/binary/main.py(7)main()
→ module = importlib.import_module(hashlib_str)
(Pdb) l
 3
 4     def main():
 5         import pdb
 6         pdb.set_trace()
 7 →     module = importlib.import_module(hashlib_str)
 8         md5 = module.md5()
 9
10     if __name__ == '__main__':
```

Runtime effects



Запуск подпроцессов



Защита данных



Наличие python stdlib



Интерактивная отладка





Вместо заключения

Скорость

```
def fib(n):  
    if n < 2:  
        return n  
    return fib(n - 1) + fib(n - 2)
```


Скорость

```
import fib
import time

def main():
    start = time.time()
    assert fib.fib(40) == 102334155
    print("duration = ", time.time() - start)

if __name__ == "__main__":
    main()
```

Скорость

```
PY3_PROGRAM( )
```

```
PY_SRCS(
```

```
    MAIN main.py
```

```
    TOP_LEVEL fib.py
```

```
)
```

```
END( )
```

Скорость



Базовый случай

28.6с

34.2с

16.1с

20.4с

Скорость



Базовый случай

28.6с

34.2с

16.1с

20.4с

PGO + LTO

-

-

11.3с

-

Скорость - cythonize

```
linux ~> cython --module-name fib fib.py -3
```

```
linux ~> gcc -shared -fPIC -O2 -fno-strict-aliasing  
-I/usr/include/python3.10 -o fib.so fib.c
```

Скорость - cythonize

```
PY3_PROGRAM( )  
  
PY_SRCS(  
    MAIN main.py  
    TOP_LEVEL  
    CYTHONIZE_PY fib.py  
)  
  
END( )
```


Скорость



Базовый случай

28.6с

34.2с

16.1с

20.4с

PGO + LTO

-

-

11.3с

-

Cython2

15.2с

21.8с

20.6с

26.9с

Скорость



Базовый случай

28.6с

34.2с

16.1с

20.4с

PGO + LTO

-

-

11.3с

-

Cython2

15.2с

21.8с

20.6с

26.9с

Cython3

8.4с

10.1с

8.7с

-

Скорость - С

```
// fibc.h
long fibc(long n);

// fibc.c
long fibc(long n) {
    if (n < 2) {
        return n;
    }
    return fibc(n - 1) + fibc(n - 2);
}

// fib.pyx
cdef extern from "fibc.h":
    long fibc(long n) nogil

def fib(n):
    return fibc(n)
```

Скорость - С

```
// fibc.h
long fibc(long n);

// fibc.c
long fibc(long n) {
    if (n < 2) {
        return n;
    }
    return fibc(n - 1) + fibc(n - 2);
}

// fib.pyx
cdef extern from "fibc.h":
    long fibc(long n) nogil

def fib(n):
    return fibc(n)
```

Скорость - C

```
// fibc.h
long fibc(long n);

// fibc.c
long fibc(long n) {
    if (n < 2) {
        return n;
    }
    return fibc(n - 1) + fibc(n - 2);
}

// fib.pyx
cdef extern from "fibc.h":
    long fibc(long n) nogil

def fib(n):
    return fibc(n)
```

Скорость - С

```
linux ~> cython --module-name fib fib.pyx -3
```

```
linux ~> gcc -O2 -fno-strict-aliasing -c fibc.c
```

```
linux ~> gcc -shared -fPIC -O2 -fno-strict-aliasing  
-I/usr/include/python3.10 -o fib.so fibc.o fib.c
```

Скорость - С

```
PY3_PROGRAM( )

PY_SRCS(
    MAIN main.py
    TOP_LEVEL
    CYTHON_C fib.pyx
)

SRCS(
    fib.c
)

END( )
```


Скорость



Базовый случай	28.6c	34.2c	16.1c	20.4c
PGO + LTO	-	-	11.3c	-
Cython2	15.2c	21.8c	20.6c	26.9c
Cython3	8.4c	10.1c	8.7c	-
Cython + C	0.2c	0.2c	0.2c	0.2c

Кто победил?





[https://github.com
/yandex/yatool](https://github.com/yandex/yatool)

Вопросы