Challenger #2 - Tools 

`_GLIBCXX_USE_CXX11_ABI` and GCC > 5.x

```cpp
#define _GLIBCXX_USE_CXX11_ABI 1

#include <string>

void function(std::string input) {}
```

Force C++11 ABI

```
$> cat /etc/os-release
NAME="Ubuntu"
VERSION="14.04.6 LTS, Trusty Tahr"
```

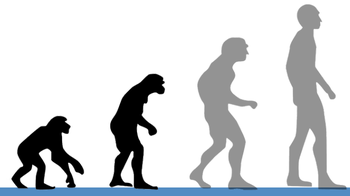
Old distro (no dual ABI)

```
$> g++ --version
g++ (Ubuntu 6.5.0-2ubuntu1~14.04.1) 6.5.0 20181026
```

New compiler

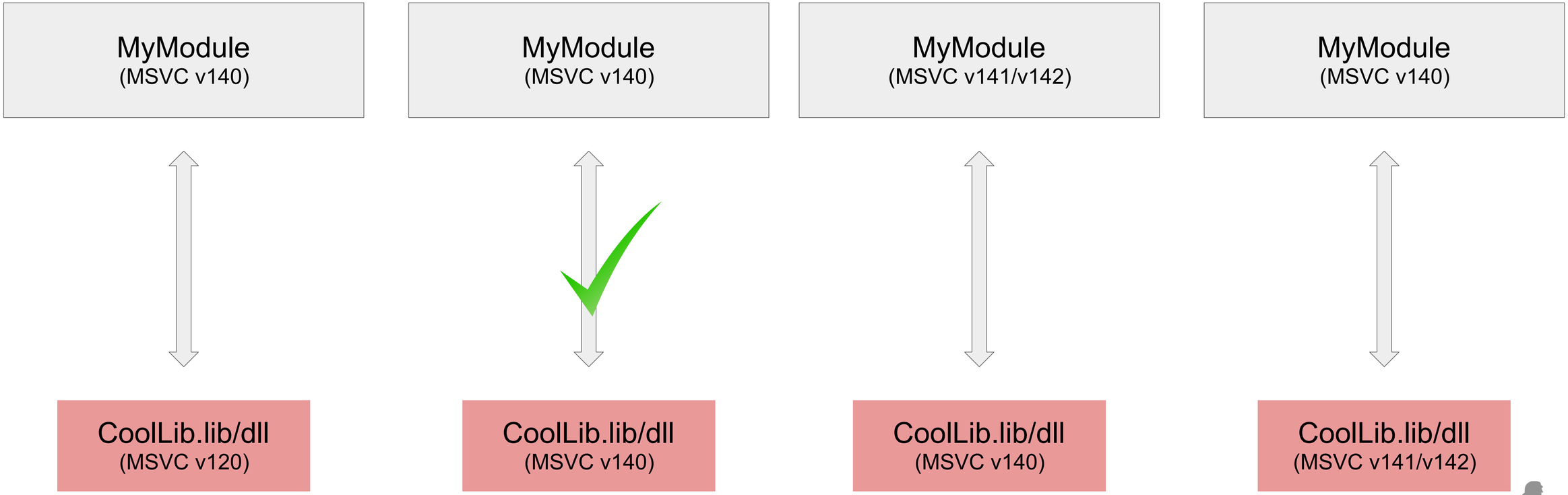
```
$> nm -gC lib.o
0000000000000000 T function(std::string)
```

No C++11 ABI



Challenger #2 - Tools

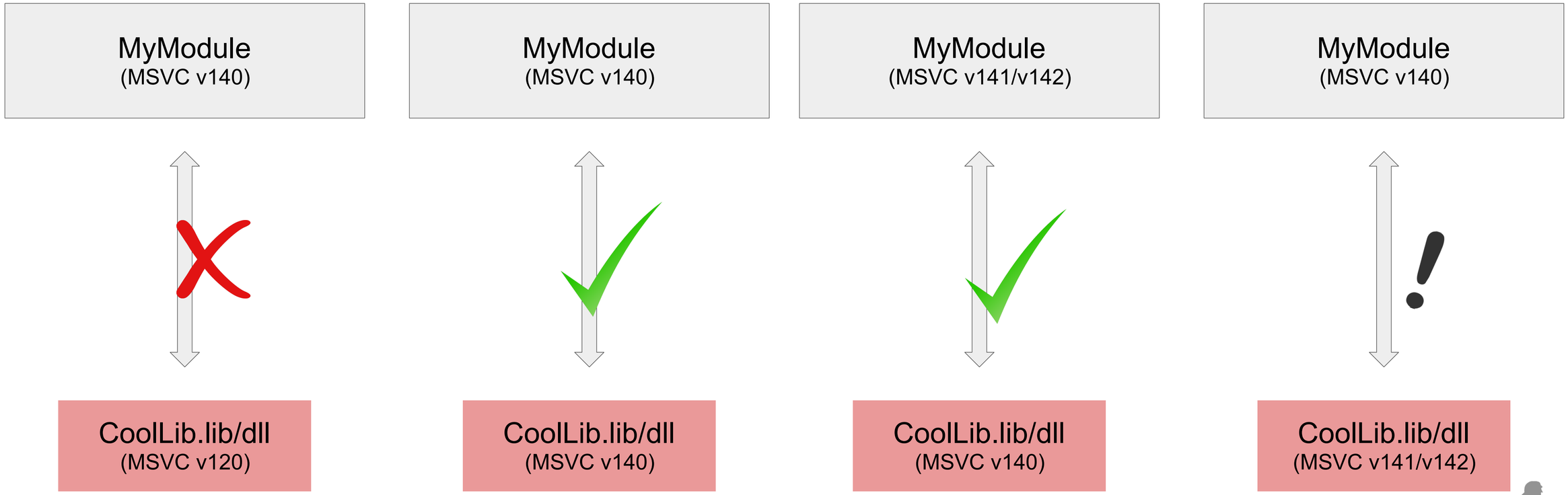
Visual Studio: *“VC Runtime in the latest MSVC v142 toolset is binary compatible with v140 and v141”*



Quoted: https://devblogs.microsoft.com/cppblog/cpp-binary-compatibility-and-pain-free-upgrades-to-visual-studio-2019/

Challenger #2 - Tools 

Visual Studio: *"... linking all of it together (with the latest linker)..."*
"... VCRedist can't be older than any of the toolset versions used to build…"



Quoted: https://devblogs.microsoft.com/cppblog/cpp-binary-compatibility-and-pain-free-upgrades-to-visual-studio-2019/

OS, arch, distro: C/C++ runtime, ABI

Tools: compiler (flags, toolset), distro

Francisco de Goya. *El Aquelarre* o *El gran Cabrón*. 1823
Museo Nacional del Prado, Madrid (Spain)

Challenger #3

The sources

Challenger #3 - The sources 

lib.h

```h
int add_1(int a = 0);

int add_2(int a = 0) {
    return a + 2;
}
```

lib.cpp

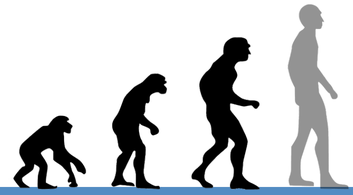
```cpp
#include "lib.h"

int add_1(int a) {
    return a + 1;
}
```

main.cpp

```cpp
int main(int argc, const char* argv[]) {
    std::cout << "add_1: " << add_1() << "\n";
    std::cout << "add_2: " << add_2() << "\n";
```

Compile and link shared:

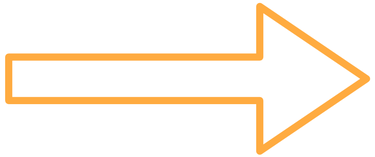
```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
g++ -o executable bin/main.cpp -lfoo

./executable
add_1: 1
add_2: 2
```



Challenger #3 - The sources 

lib.cpp

```cpp
#include "lib.h"

int add_1(int a) {
    return a + 1;
}
```



lib.cpp

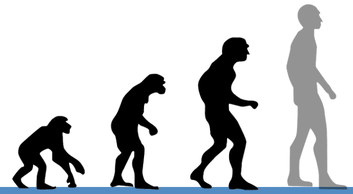
```cpp
#include "lib.h"

int add_1(int a) {
    return a + 10;
}
```

```
./executable
add_1: 1
add_2: 2
```

Compile only the shared library:

```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
~~g++ -o executable bin/main.cpp -lfoo~~
```

```
./executable
add_1: 10
add_2: 2
```



Challenger #3 - The sources 

lib.h

```
int add_1(int a = 0);

int add_2(int a = 0) {
    return a + 2;
}
```

lib.h

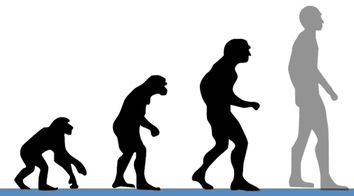
```
int add_1(int a = 100);

int add_200(int a = 0) {
    return a + 200;
}
```

Compile only the shared library:

```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
g++ -o executable bin/main.cpp -lfoo
```

```
./executable
add_1: 10
add_2: 2
```

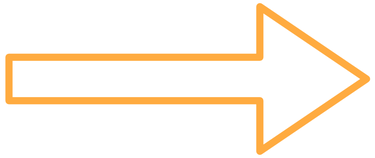
```
./executable
add_1: 10
add_2: 2
```



Challenger #3 - The sources 

lib.h

```
int add_1(int a = 0);

int add_2(int a = 0) {
    return a + 2;
}
```



lib.h

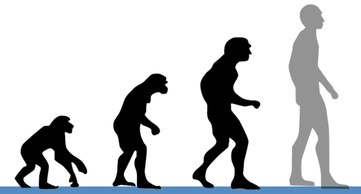
```
int add_100(int a = 0);

int add_2(int a = 0) {
    return a + 2;
}
```

Compile only the shared library:

```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
~~g++ -o executable bin/main.cpp -lfoo~~
```

☺️

```
./executable
./executable: symbol lookup error: ./executable: undefined symbol: _Z5add_1i
```



Challenger #3 - The sources 

lib.h

```
struct Sum {
    int a;
    int b;
};

int sum(const Sum&);

int sum_inline(const Sum& s) {
    return s.a + s.b;
}
```

lib.cpp

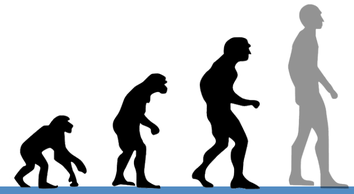
```
int sum(const Sum& s) {
    return s.a + s.b;
}
```

main.cpp

```
int main(int argc, const char* argv[]) {
    Sum s{10, 10};
    std::cout << "sum: " << sum(s) << "\n";
    std::cout << "sum_inline: " << sum_inline(s) << "\n";
```

Compile and link shared:

```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
g++ -o executable bin/main.cpp -lfoo

./executable
sum: 20
sum_inline: 20
```



Challenger #3 - The sources 

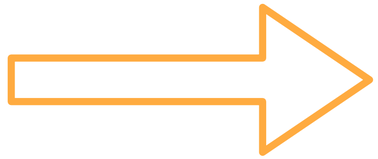
lib.h

```
struct Sum {
    int a;
    int b;
};
```

```
int sum_inline(const Sum& s) {
    return s.a + s.b;
}
```

lib.cpp

```
int sum(const Sum& s) {
    return s.a + s.b;
}
```



lib.h

```
struct Sum {
    int a;
    int b;
    int c;
};
```

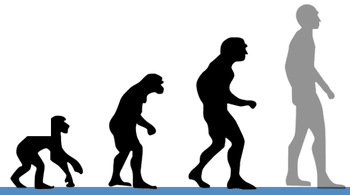
```
int sum_inline(const Sum& s) {
    return s.a + s.b + s.c;
}
```

lib.cpp

```
int sum(const Sum& s) {
    return s.a + s.b + s.c;
}
```

```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
~~g++ -o executable bin/main.cpp -lfoo~~

./executable
sum: -1743113196
sum_inline: 20
```

😟



Challenger #3 - The sources 

lib.h

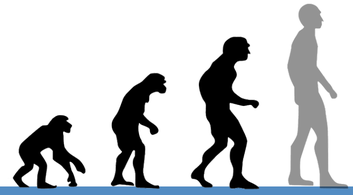
```
class Person {
public:
    Person(const std::string&);
    ~Person();

    std::string name;
};
```

main.cpp

```
void greet(const std::string& name) {
    std::cout << "Hello " << name << "!\n";
}

int main(int argc, const char* argv[]) {
    Person s{"Manuel"};
    if (argc == 1) { greet(s.name); }
    else { greet(argv[1]); }
}
```

Compile and link shared:

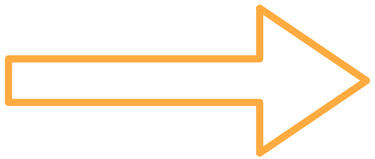
```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
g++ -o executable bin/main.cpp -lfoo

./executable
Hello Manuel!
./executable Javier
Hello Javier!
```



Challenger #3 - The sources 

lib.h

```
class Person {
public:
    Person(const std::string&);
    ~Person();

    std::string name;
};
```

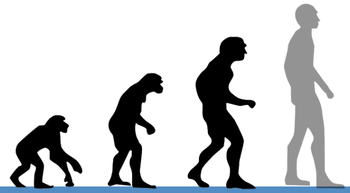


lib.h

```
class Person {
public:
    Person(const std::string&);
    ~Person();
};
```

Compile and link shared:

```
g++ -c -fpic lib/lib.cpp
g++ -shared -o libfoo.so lib.o
g++ -o executable bin/main.cpp -lfoo

./executable
Segmentation fault
./executable Javier
Hello Javier!
```



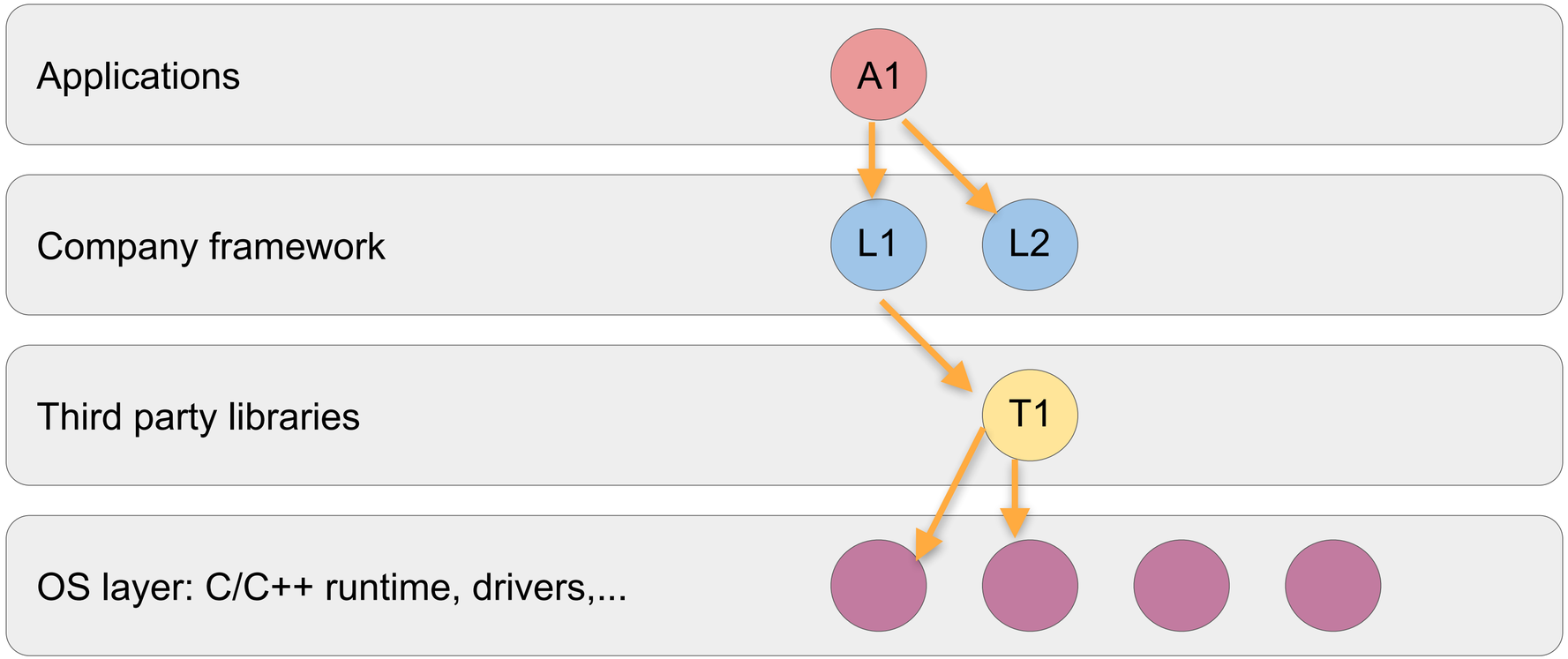
Challenger #4

The environment

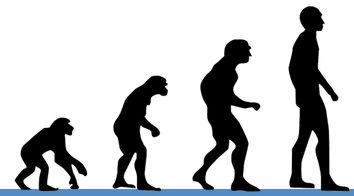
javierg@jfrog.com | @jgsogo | JFrog Ltd.

Challenger #4 - The environment

When shared libraries enter the game...

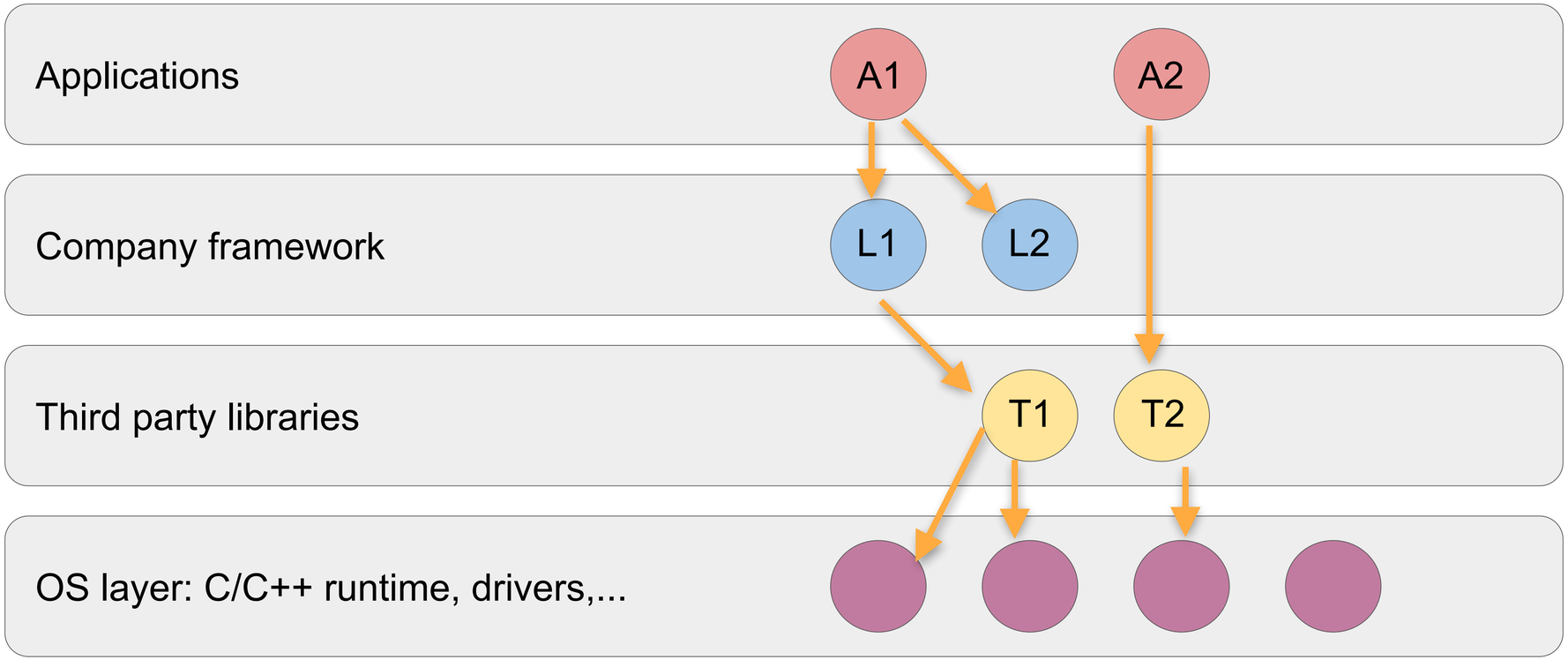


- `n` deploy environments

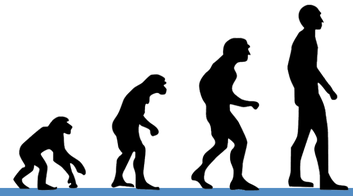


Challenger #4 - The environment 

When shared libraries enter the game...

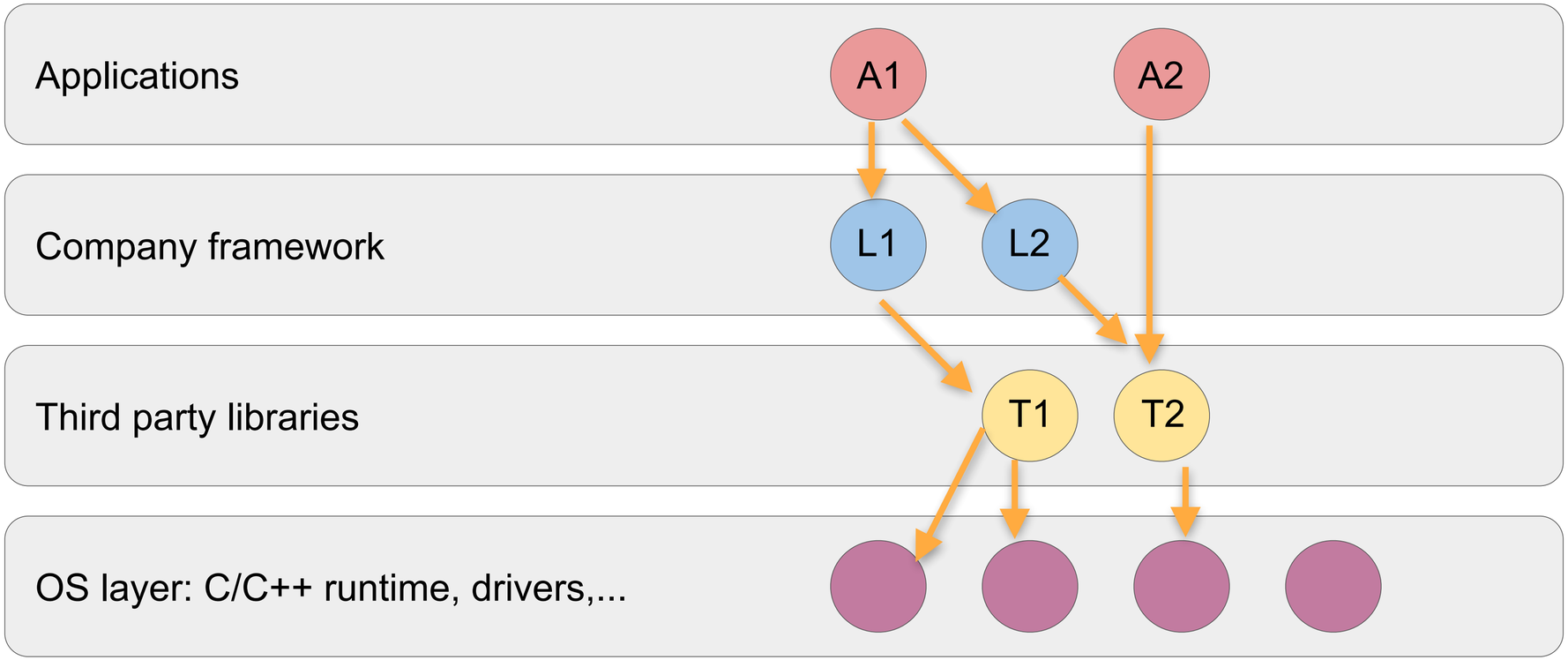


- `n` deploy environments

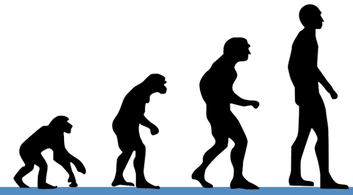


Challenger #4 - The environment 

When shared libraries enter the game...

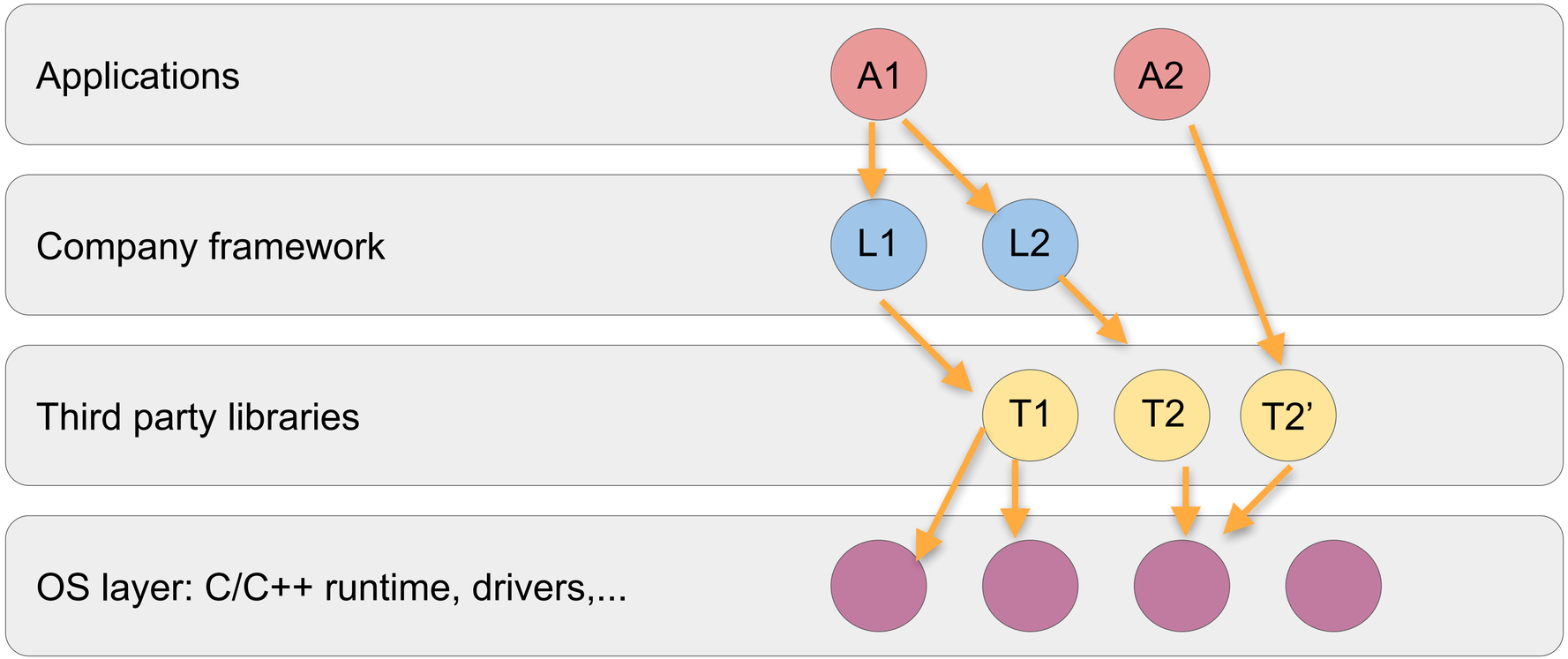


- `n` deploy environments

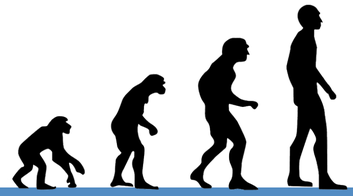


Challenger #4 - The environment 

When shared libraries enter the game...

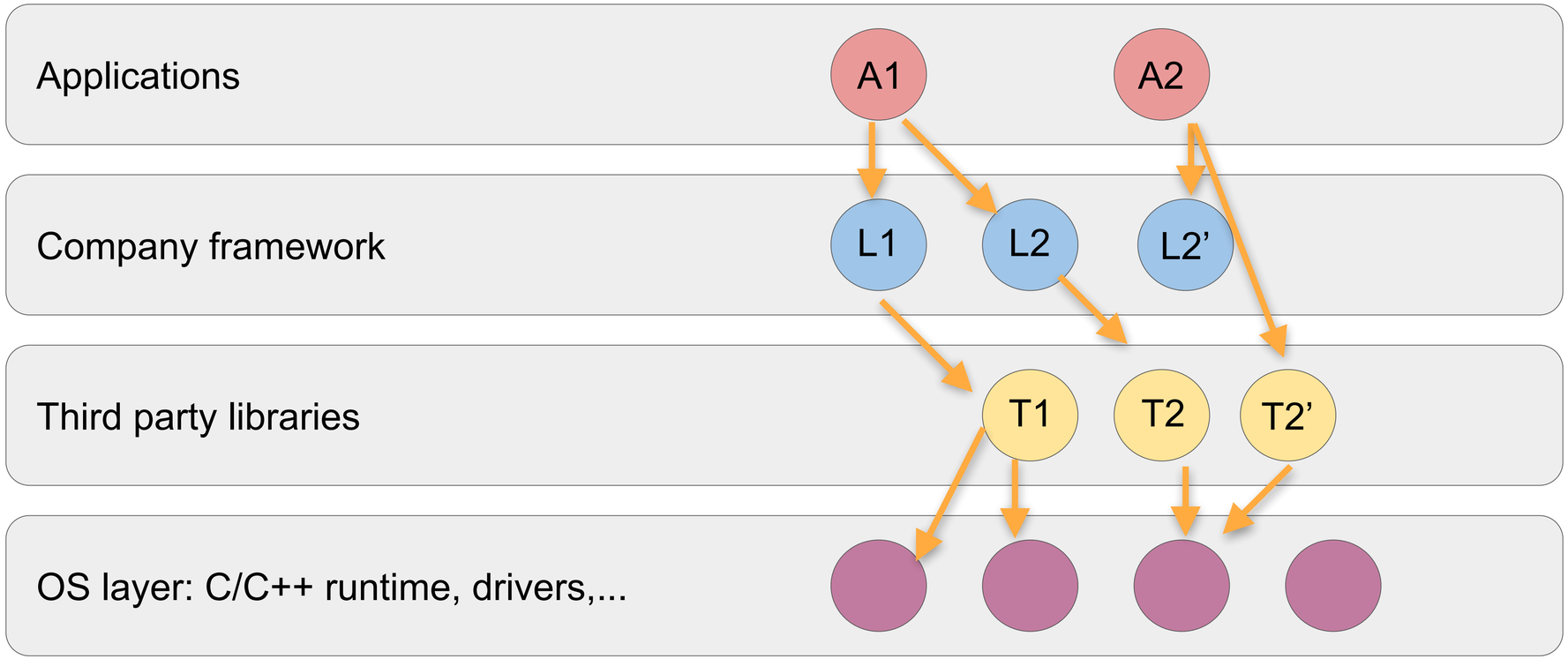


- n deploy environments

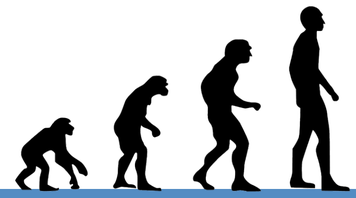


Challenger #4 - The environment 

When shared libraries enter the game...



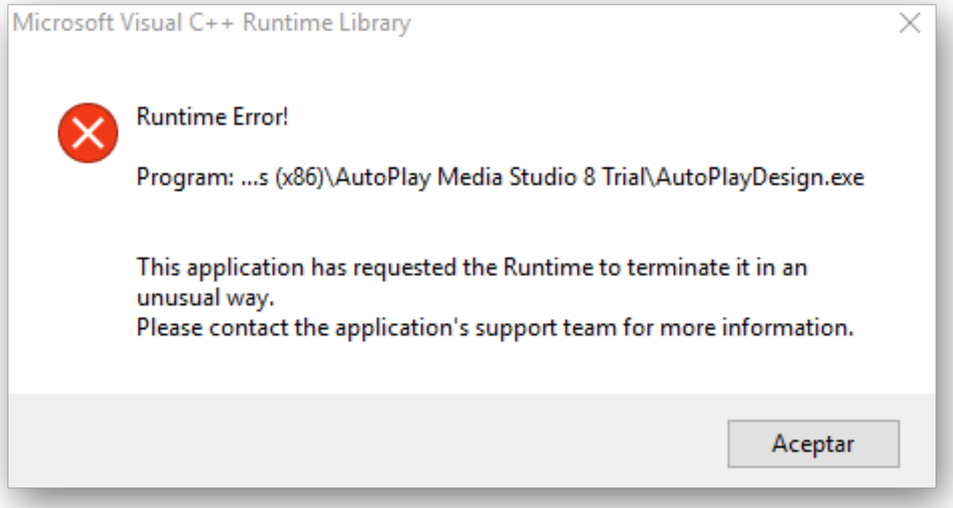
- n deploy environments

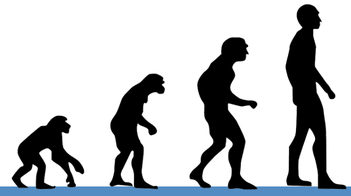


Challenger #4 - The environment 

Runtime error!

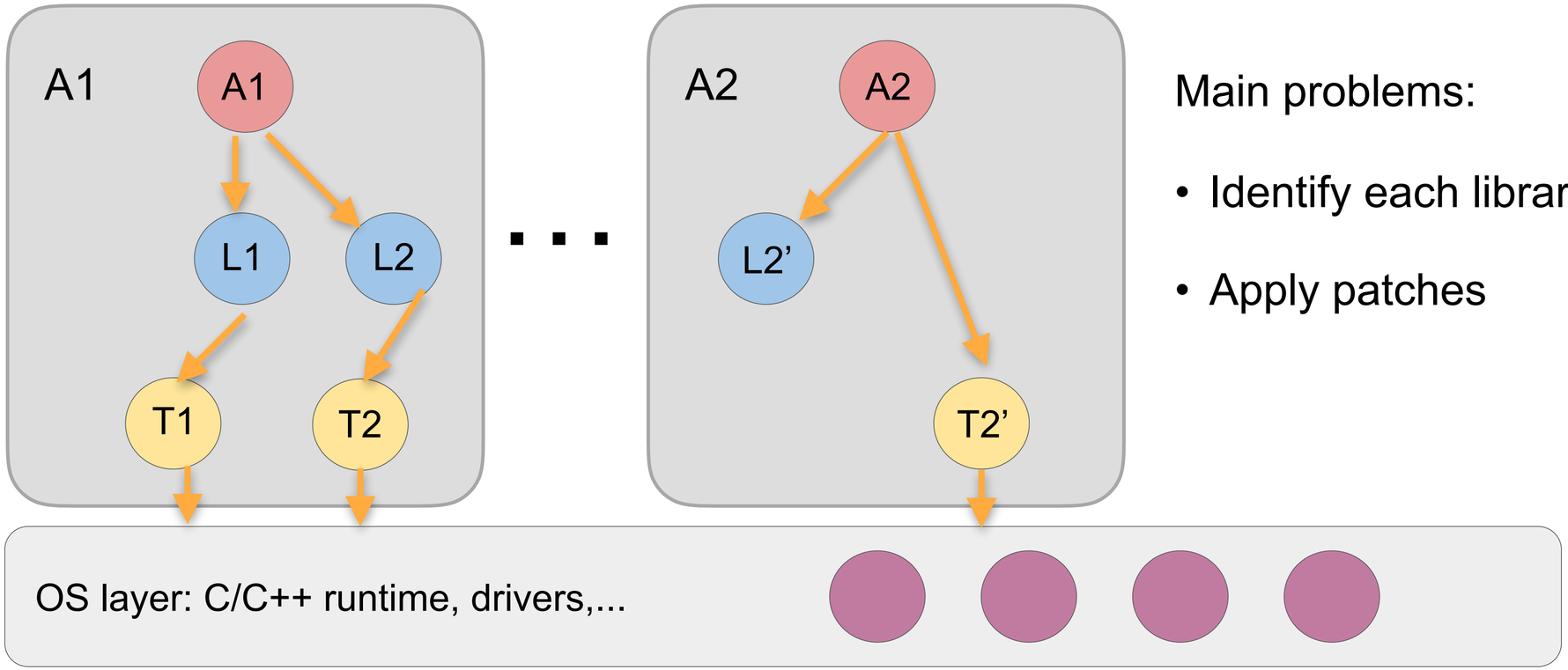
```
./executable: symbol lookup error: ./executable: undefined symbol:
_Z9get_helloRSt6vectorINSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEESaIS5_EE

./executable: error while loading shared libraries: libfoo.so: cannot open shared object file: No such
file or directory
```





Challenger #4 - The environment 

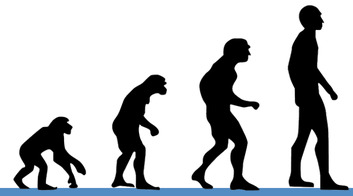
Final layout: each application with all its dependencies

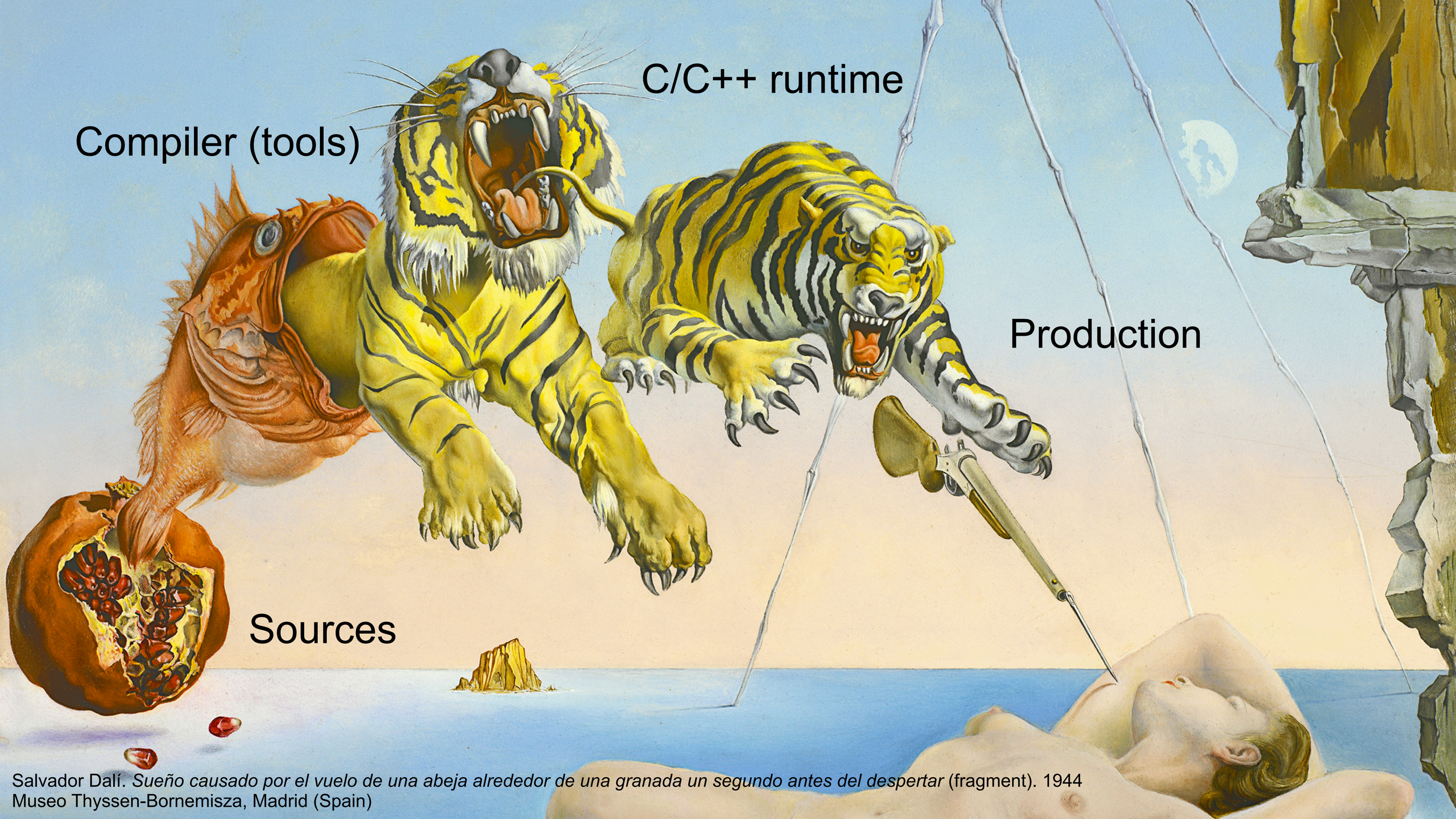


Main problems:

- Identify each library
- Apply patches

· n deploy environments





Salvador Dalí. *Sueño causado por el vuelo de una abeja alrededor de una granada un segundo antes del despertar* (fragment). 1944
Museo Thyssen-Bornemisza, Madrid (Spain)