

Biicode C and C++ deps manager with a hosting service

C++Now 2015

Diego Rodriguez-Losada, PhD
@diegorlosada



SETUP

- Signup www.biicode.com
- Last bii www.biicode.com/downloads
- CMake ≥ 3.0 in the path
- MinGW or gcc or clang (VS for zmq, last part)
- [git]

OUTLINE

- Setup
- Build system: CMake with steroids
- Dependencies:
 - Publish, depend
 - Tags: DEV, STABLE
 - Versions, updates
- ToDo: ZMQ + Protobuf
- Analysis & Future:
 - Packet

FILE BASED ANALYSIS



\$ bii build
HelloBii!

main.cpp

```
#include "hello.h"
main() {
    hello();
}
```

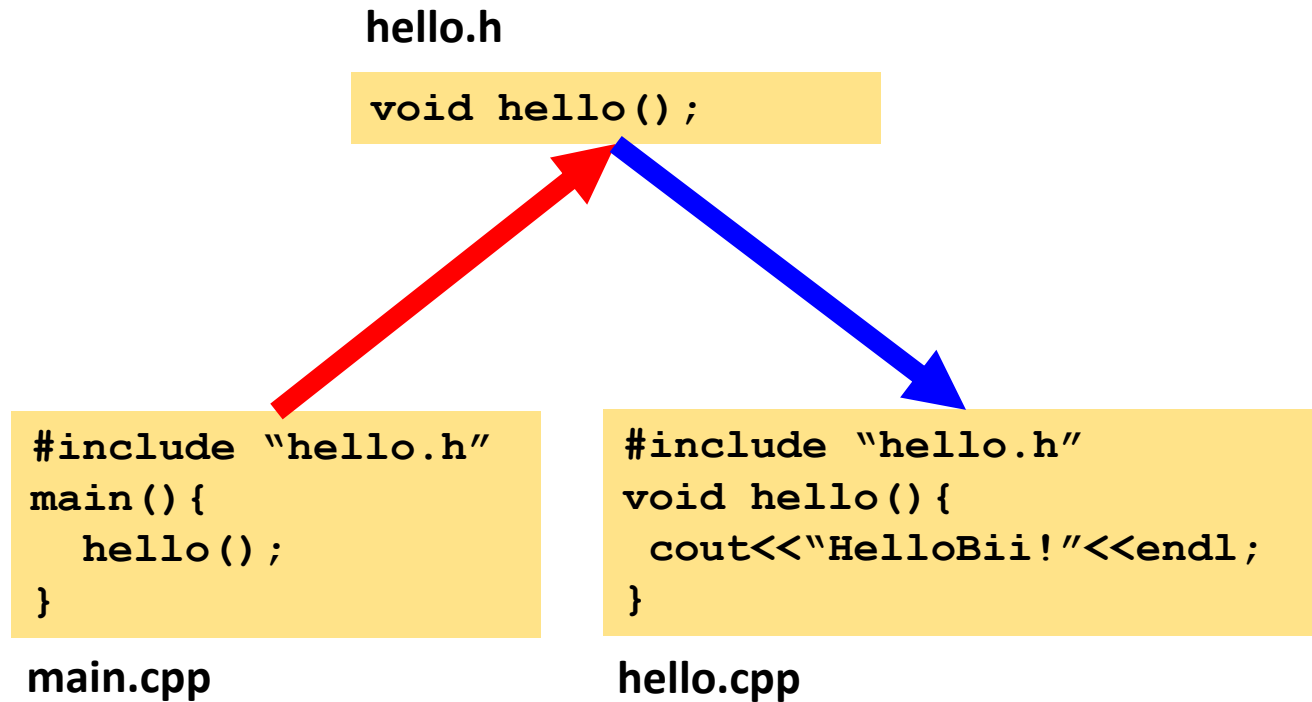
hello.h

```
void hello();
```

hello.cpp

```
#include "hello.h"
void hello() {
    cout<<"HelloBii!"<<endl;
}
```

FILE DEPENDENCIES



About DEPENDENCIES DETECTION

Yay C++

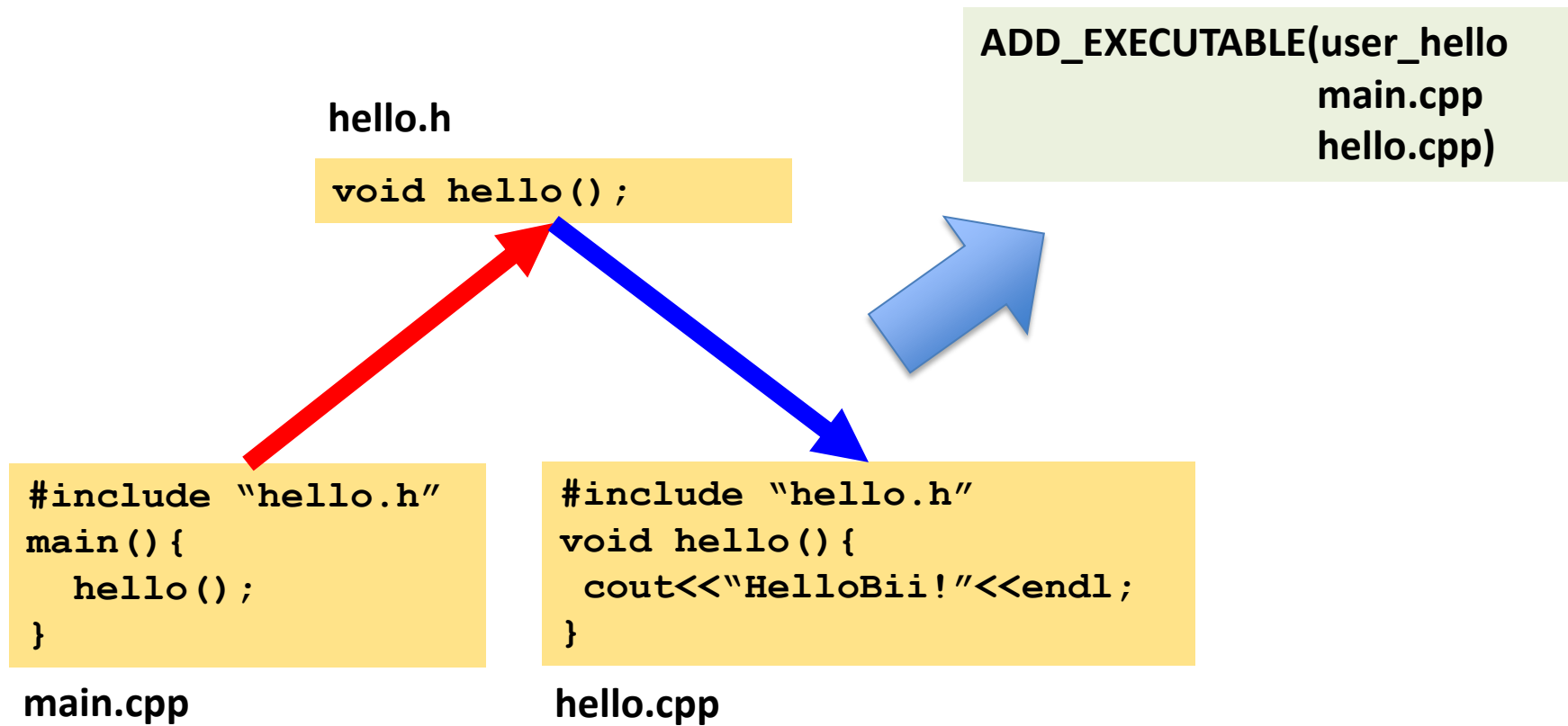
- Namespace, classes
- Includes
- Static vars
- Fast (no ply)

Nay C

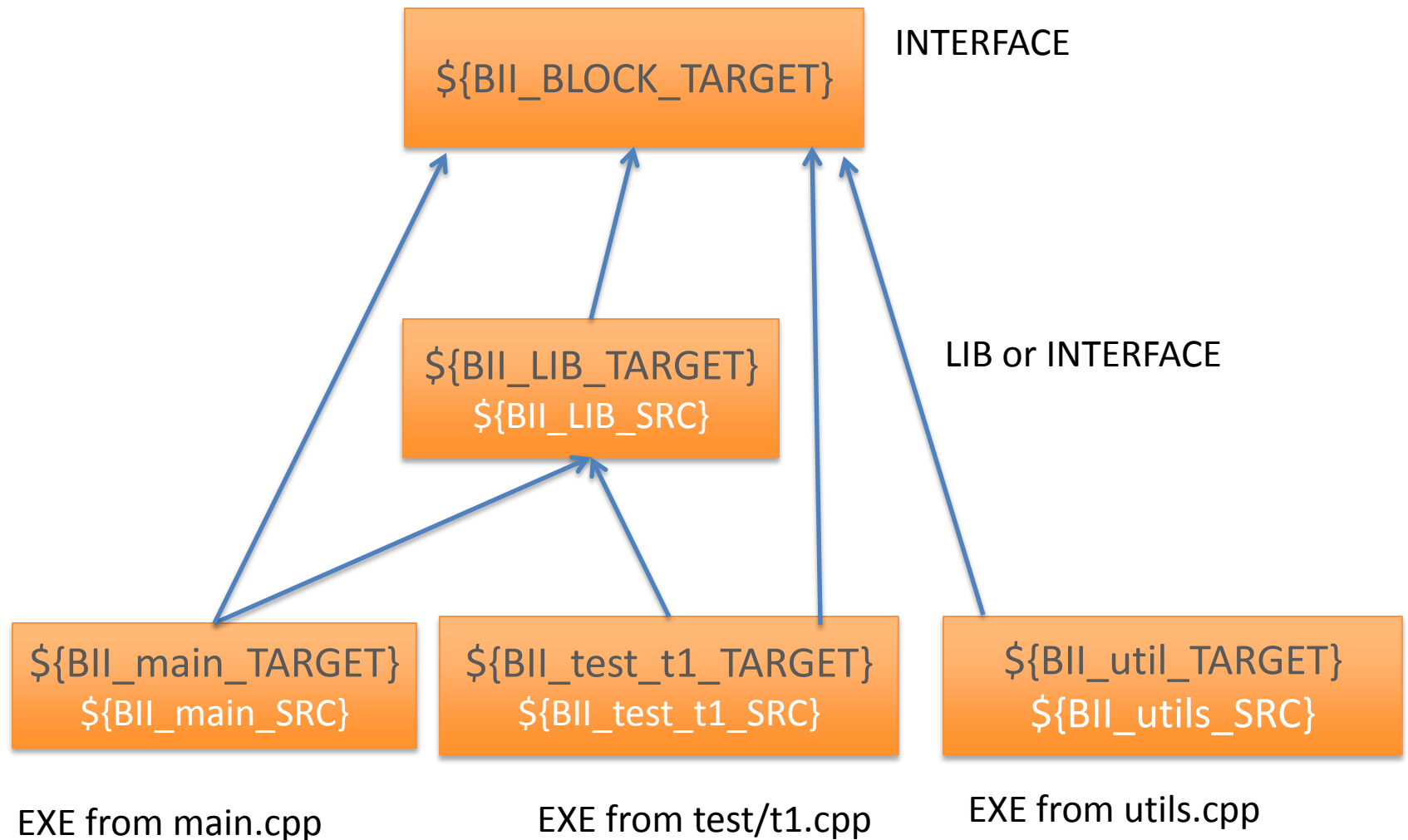
- # include
MY_INCLUDE_MACRO
- Extern, global vars linkage
- MACROS
- Preprocessor
- Same symbol implemented many times.

```
$ bii deps
```

BUILD AUTOMATION



CMAKE TARGETS



TEST CREATION

- biicode.conf

```
[tests]
    hello.cpp
    bye.cpp

$ bii test
```

SHARED LIBRARY

```
$ bii configure -DBUILD_SHARED_LIBS=ON
```

CHANGING NAME EXES

CMakeLists.txt

```
SET_TARGET_PROPERTIES(${BII_hello_TARGET}  
                      PROPERTIES OUTPUT_NAME hello)  
SET_TARGET_PROPERTIES(${BII_bye_TARGET}  
                      PROPERTIES OUTPUT_NAME bye)
```

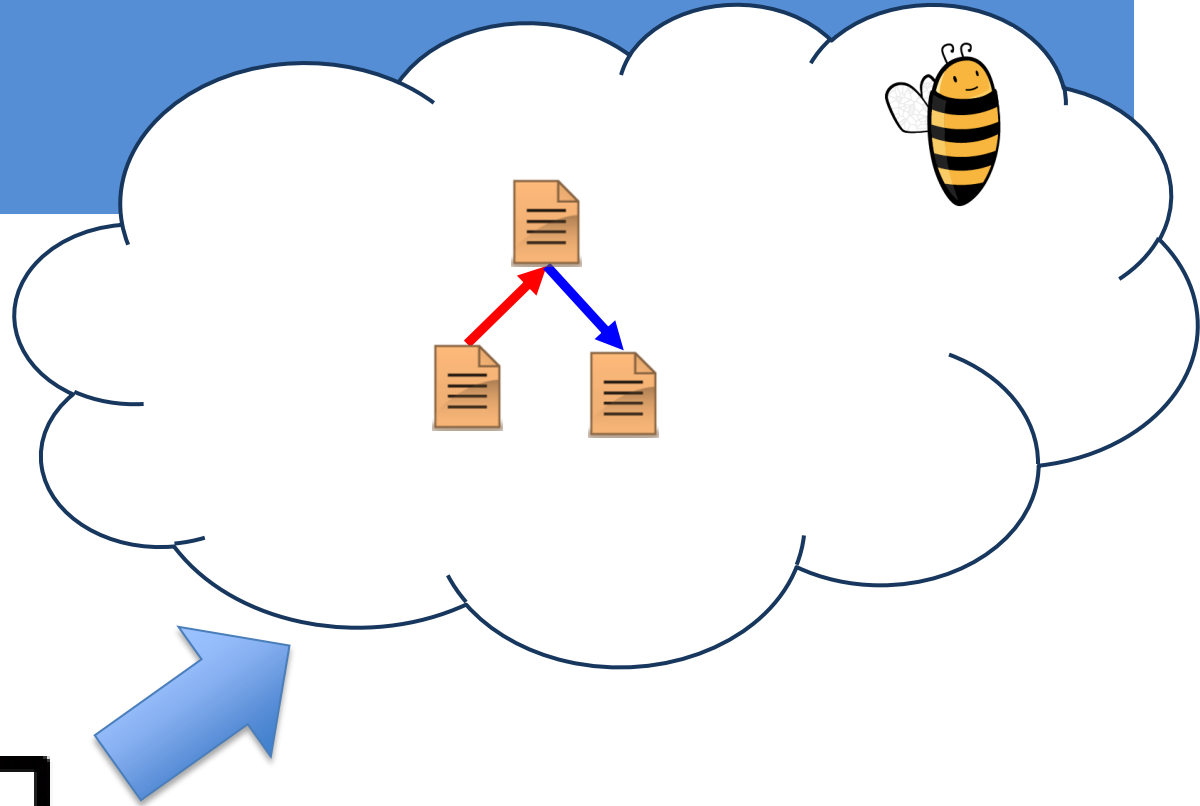
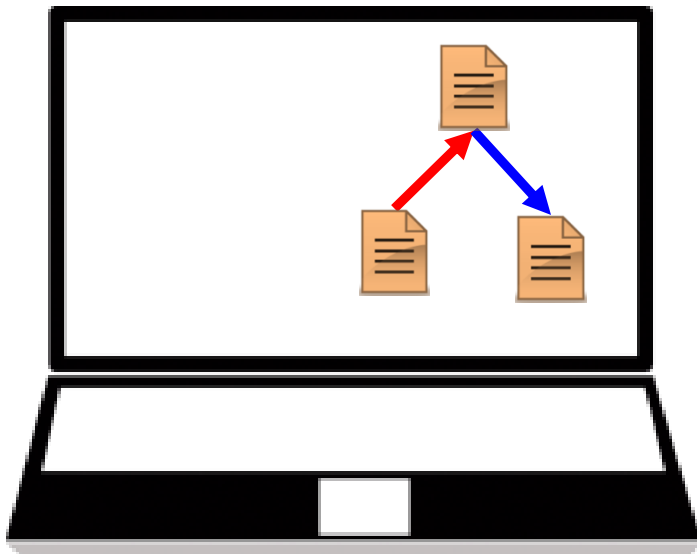
ACTIVATE C++11

CMakeLists.txt

```
TARGET_COMPILE_OPTIONS(${BII_BLOCK_TARGET}  
                        INTERFACE "-std=c++11")
```

PUBLISH

\$ bii publish



No packaging
No binaries
No libraries
No configuration

DEPEND



\$ bii find
\$ bii build
HelloBii!
HelloBii!

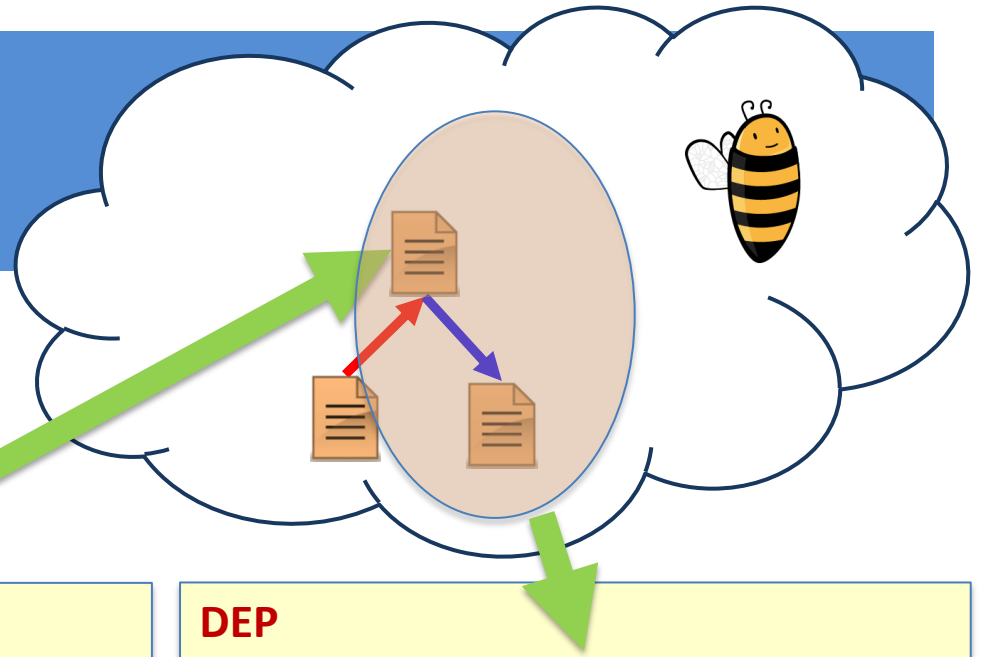
chat.h

```
#include "user/hello/hello.h"  
void chat() {  
    hello();  
    hello();  
}
```

main.cpp

```
#include "chat.h"  
main() {  
    chat();  
}
```

DEPS RETRIEVAL



SRC

```
#include "user/hello/hello.h"
void chat(){
    hello();
    hello();
}
```

chat.h

```
#include "chat.h"
main(){
    chat();
}
```

main.cpp

DEP

user/hello/hello.h

```
void hello();
```

```
#include "hello.h"
void hello(){
    cout<<"HelloBii!"<<endl;
}
```

user/hello/hello.cpp

DEPS BUILD

```
ADD_LIBRARY(user_hello
            hello.cpp)
ADD_EXECUTABLE(diego_chat
               main.cpp)
TARGET_LINK_LIBRARIES(diego_chat
                       user_hello)
```

CMakeLists.txt

SRC

```
#include "user/hello/hello.h"
void chat(){
    hello();
    hello();
}
```

chat.h

```
#include "chat.h"
main(){
    chat();
}
```

main.cpp

DEP

user/hello/hello.h

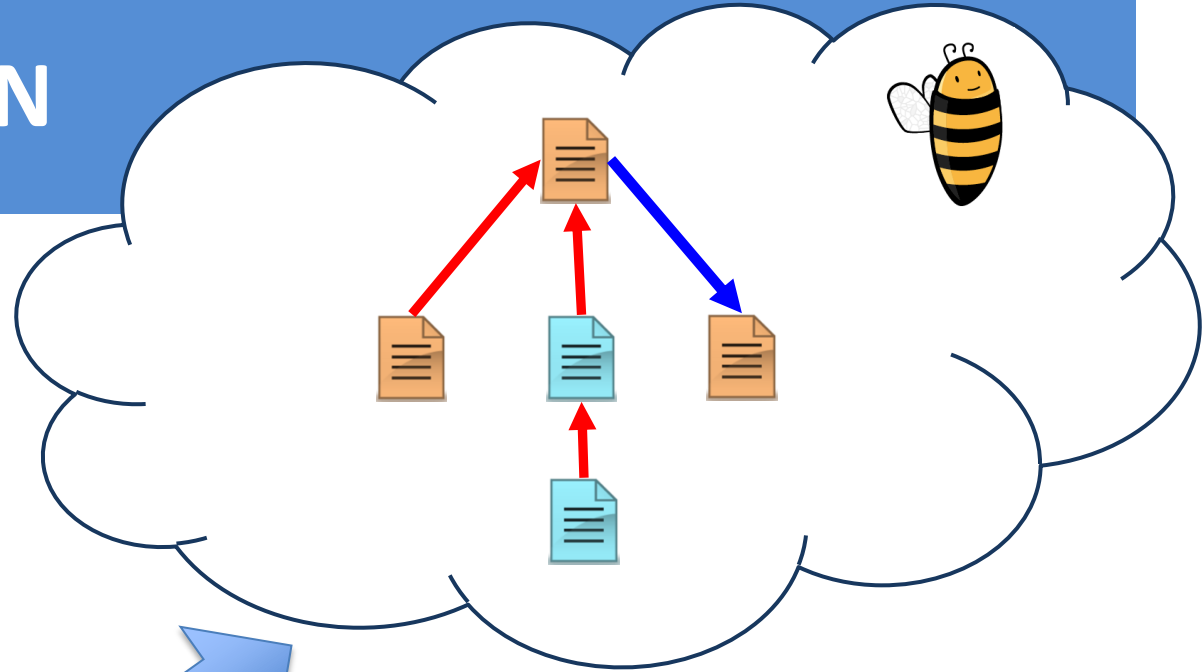
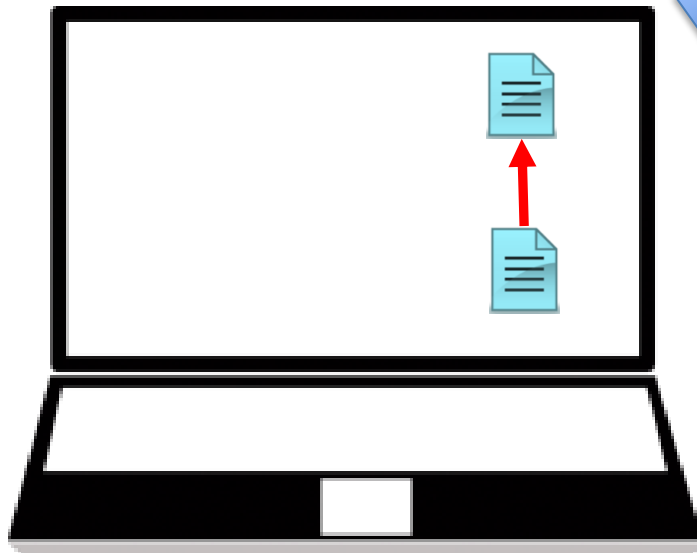
```
void hello();
```

```
#include "hello.h"
void hello(){
    cout<<"HelloBii!"<<endl;
}
```

user/hello/hello.cpp

PUBLISH AGAIN

\$ bii publish

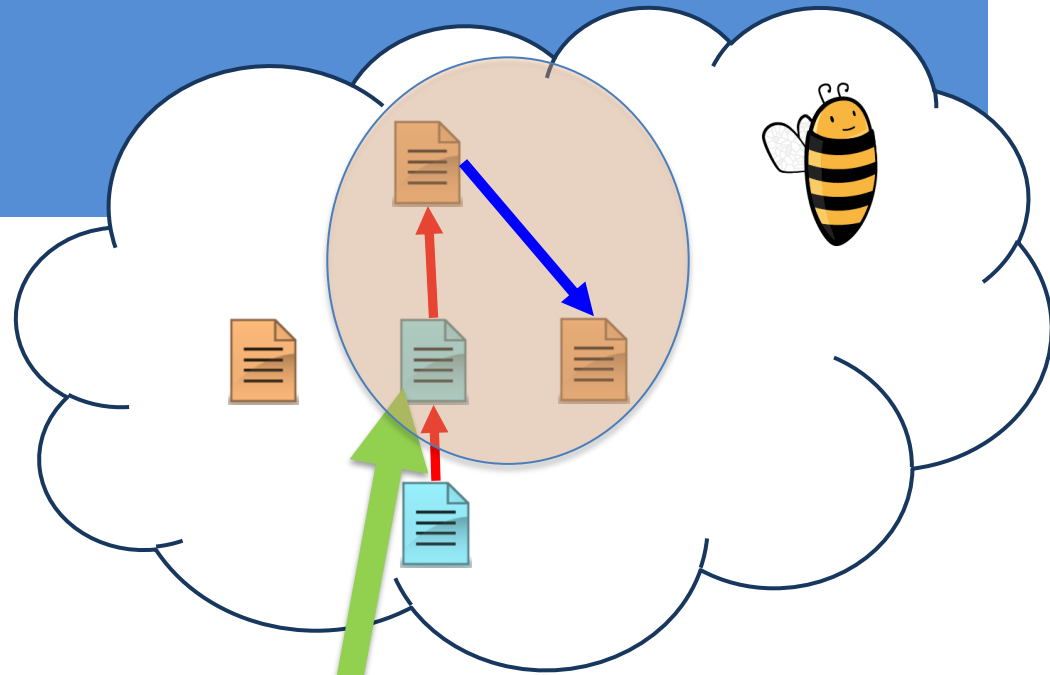


DEPEND AGAIN

@facebook



```
$ bii find  
$ bii build  
HelloBii!  
HelloBii!
```



```
#include "diego/chat/chat.h"  
main() {  
    chat();  
}
```

main.cpp

HOW TO AVOID ABSOLUTE INCLUDES

- biicode.conf

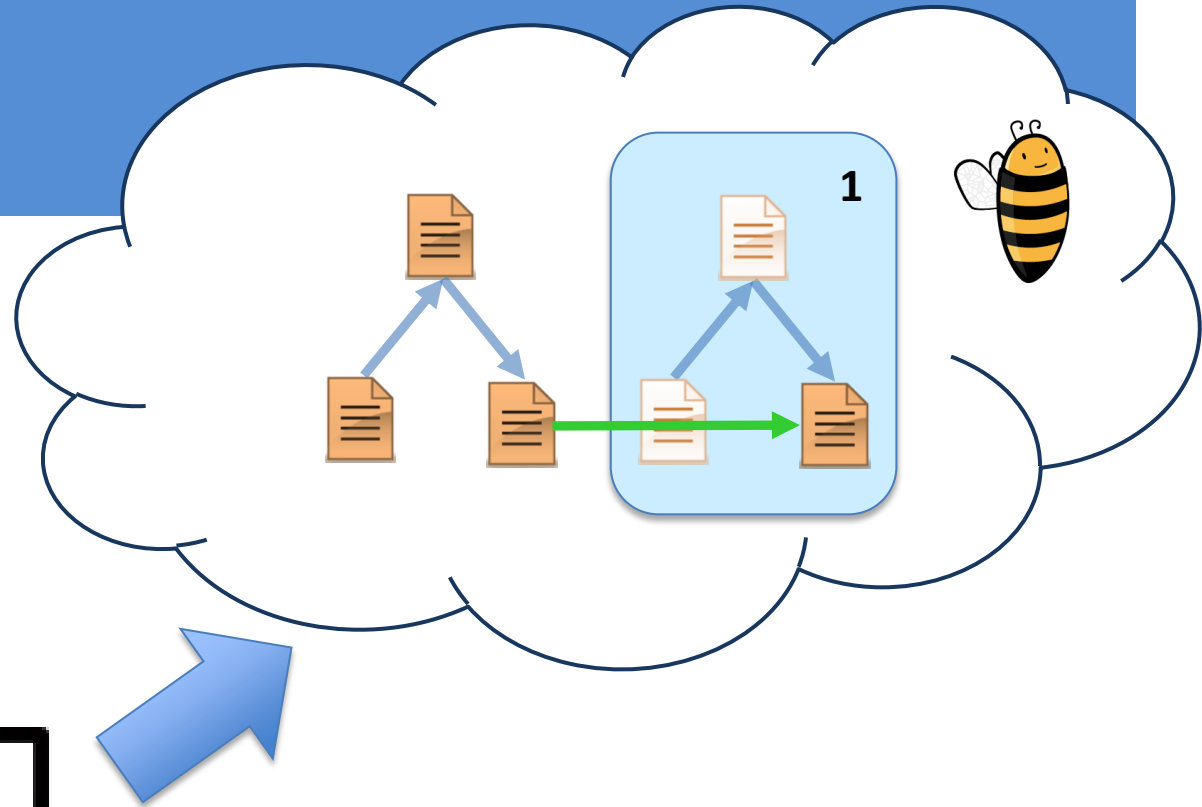
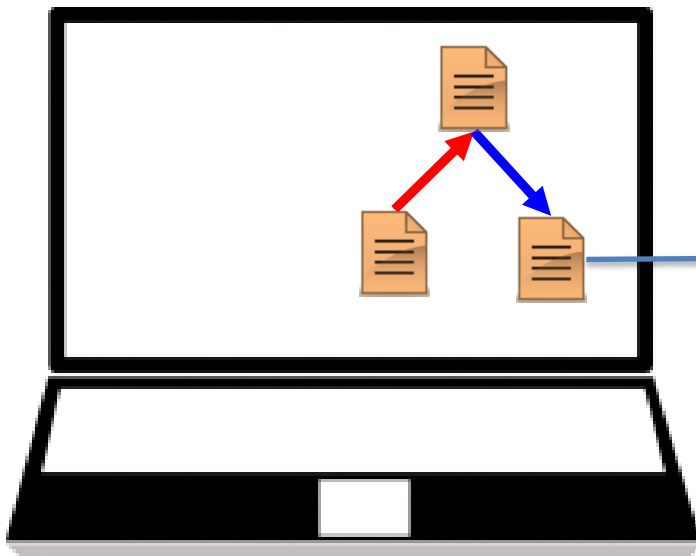
```
[includes]  
    pattern*.h: user/block/path/to
```

AUTOMATIC UPDATE DEV

- DEV version intended for testing, development
 - Cool for multiplatform (multi computer)
- It automatically updates in the consumers

VERSIONING

\$ bii publish
tag=STABLE



```
#include "hello.h"
void hello(){
    cout<<"Bii rocks!"<<endl;
}
```

hello.cpp

CHECKING DIFFS

```
$ bii diff
```

UPDATING MANUALLY

- biicode.conf

```
[requirements]  
  diego/hello: 1
```

FIND UPDATES



\$ bii find --update

\$ bii build

Bii rocks!

Bii rocks!

Know limitation, does not work
well with [includes]

FIND POLICIES

bii/policies.bii

```
diego/* : DEV  
* : STABLE
```

MANAGING TRACKS

\$ bii open user/hello



hello.cpp

```
#include "hello.h"
void hello() {
    cout<<"Bii super rocks!"<<endl;
}
```

\$ bii build

Bii super rocks!

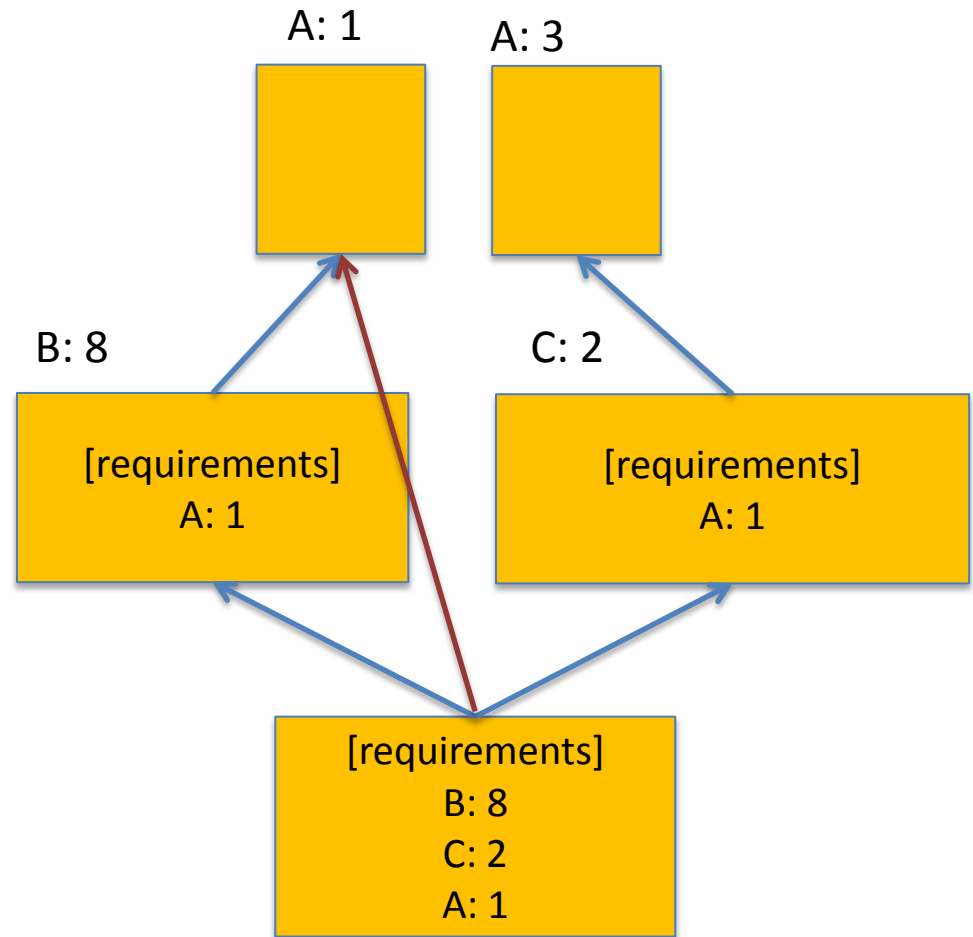
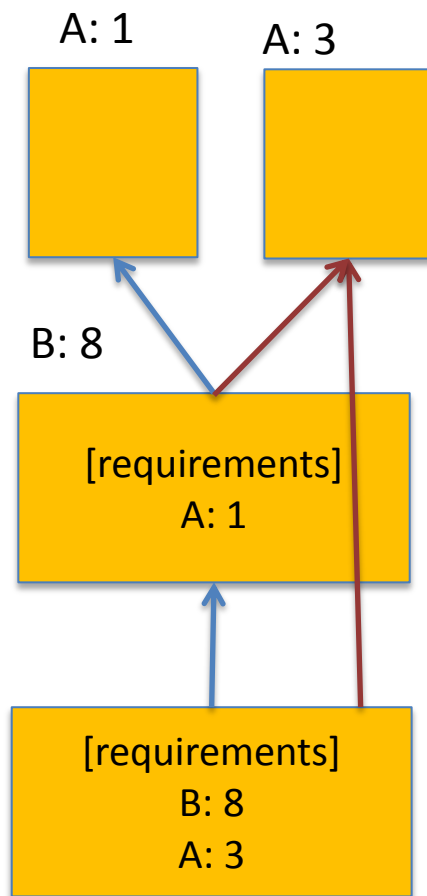
Bii super rocks!

\$ bii publish hello

biicode.conf

```
[parent]
    user/hello(test): -1
```

ADVANCED DEPS



OTHER FEATURES: HOOKS, DATA...

- Boost (depending on CMake files)

```
INCLUDE(biicode/cmake/tools)  
ACTIVATE_CPP11(INTERFACE ${BII_BLOCK_TARGET})
```

- OpenCV

```
[data]  
    myimagetest.cpp + image.jpg
```

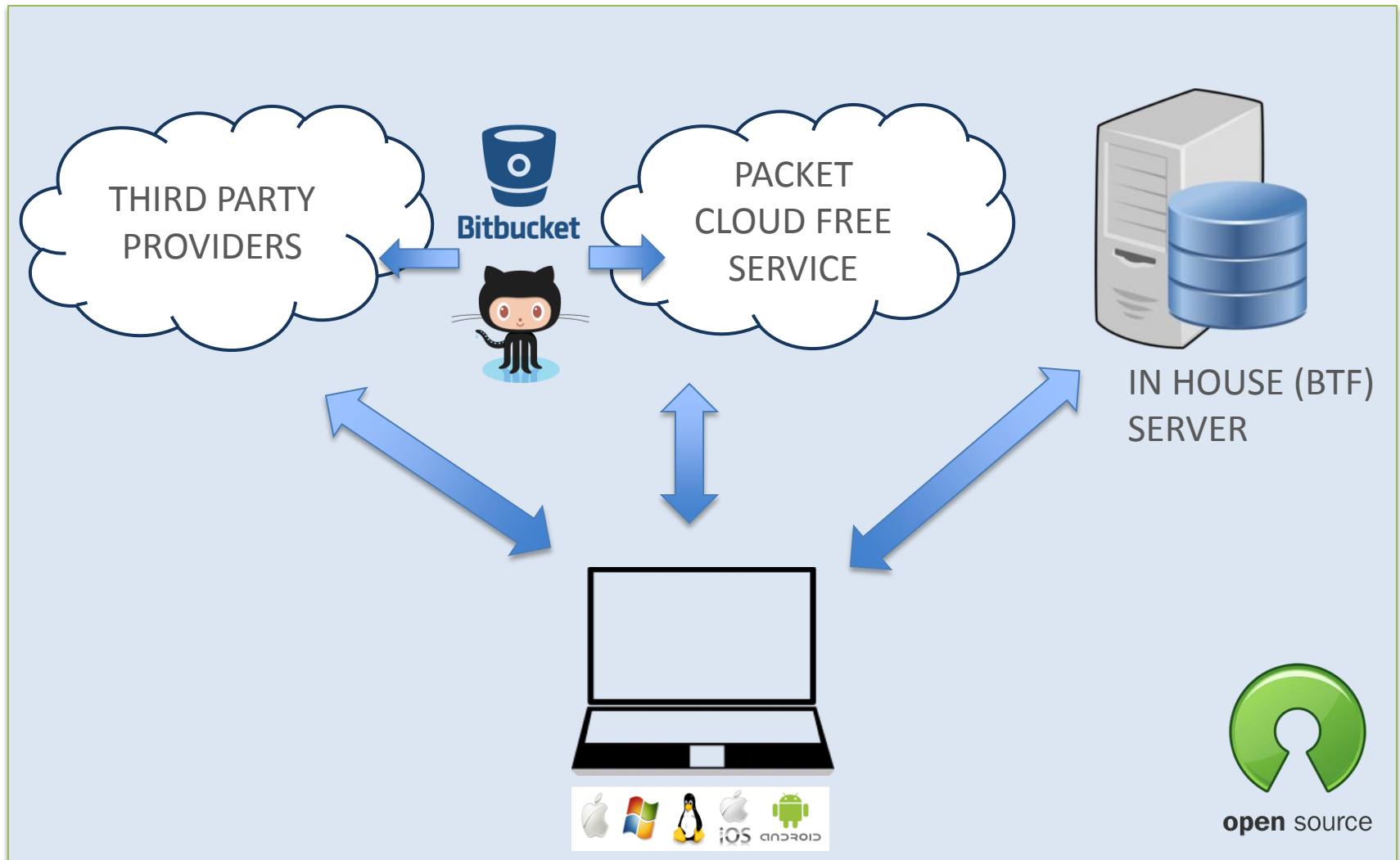
TODO: ZMQ + PROTOBUF

- Build hwserver/hwclient:
 - git clone <https://github.com/drodri/cppnowzmq.git>
 - Build (Visual Studio in Win)
- Publish and depend on message.pb.h in another project
- Change message
 - Open google/protobuf & build it
 - Use protoc:
 - `$ bin\protoc message.proto --cpp_out="."`
 - Rebuild, test & publish
 - Update in the other project

DEMO: CROSS BUILDING

- git clone
<https://github.com/drodri/median.git>
- Build R-PI
- Publish & depend on from arduino

PACKET: MULTI OS & DISTRIBUTED & OSS



PACKET: RELEASE & BINARIES MANAGEMENT

```
class BoostPkt(BasePkt):  
    name= "boost"  
    version = "2.0"  
    git: https://github.com/...  
    options = {"static": True}  
  
    def reqs(self):  
        if ...:  
            self.requires("somelib...")  
  
    def build(self):  
        if settings.os == "Windows"  
           and options.static:  
            ...
```

Boost
VS12-static-
MTd-....

Boost
OSX-Clang3.5-
shared

...

...

PACKET: DEPENDENCIES

- Contribute with your own pre-compiled binaries.
- Transitive deps, version management, deps overriding, deps conflict resolution, conditional dependencies
- **Something that can be used to release/depend on boost with 0 lockin**

NEXT TALK: TOMORROW

- Wednesday, 11:45 HUDSON
- Why in heaven there is no dependency manager in C/C++?
 - Very controversial
 - State of the art, other languages
 - Future of C++
 - **Packet proposal**

THANK YOU!



diego@biicode.com
@biicode