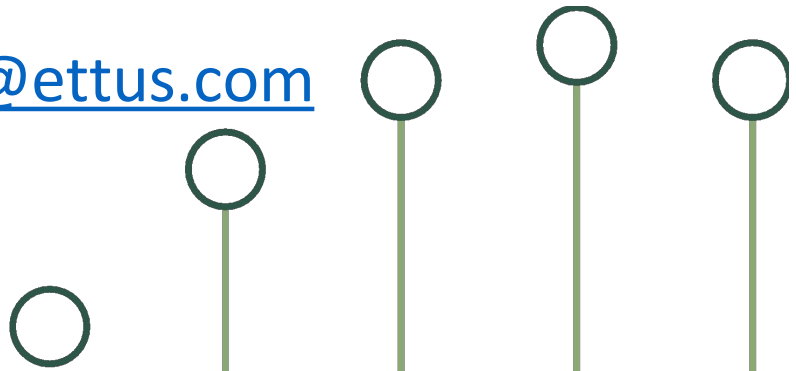
Ettus Research™
A National Instruments Company

# FPGAs: Why, When, and **How to use them** (with RFNoC™) - Pt 2

Martin Braun, Nicolas Cuervo

FOSDEM 2017, SDR Devroom

nicolas.cuervo@ettus.com



# Outline

Ettus Research™ A National Instruments Company

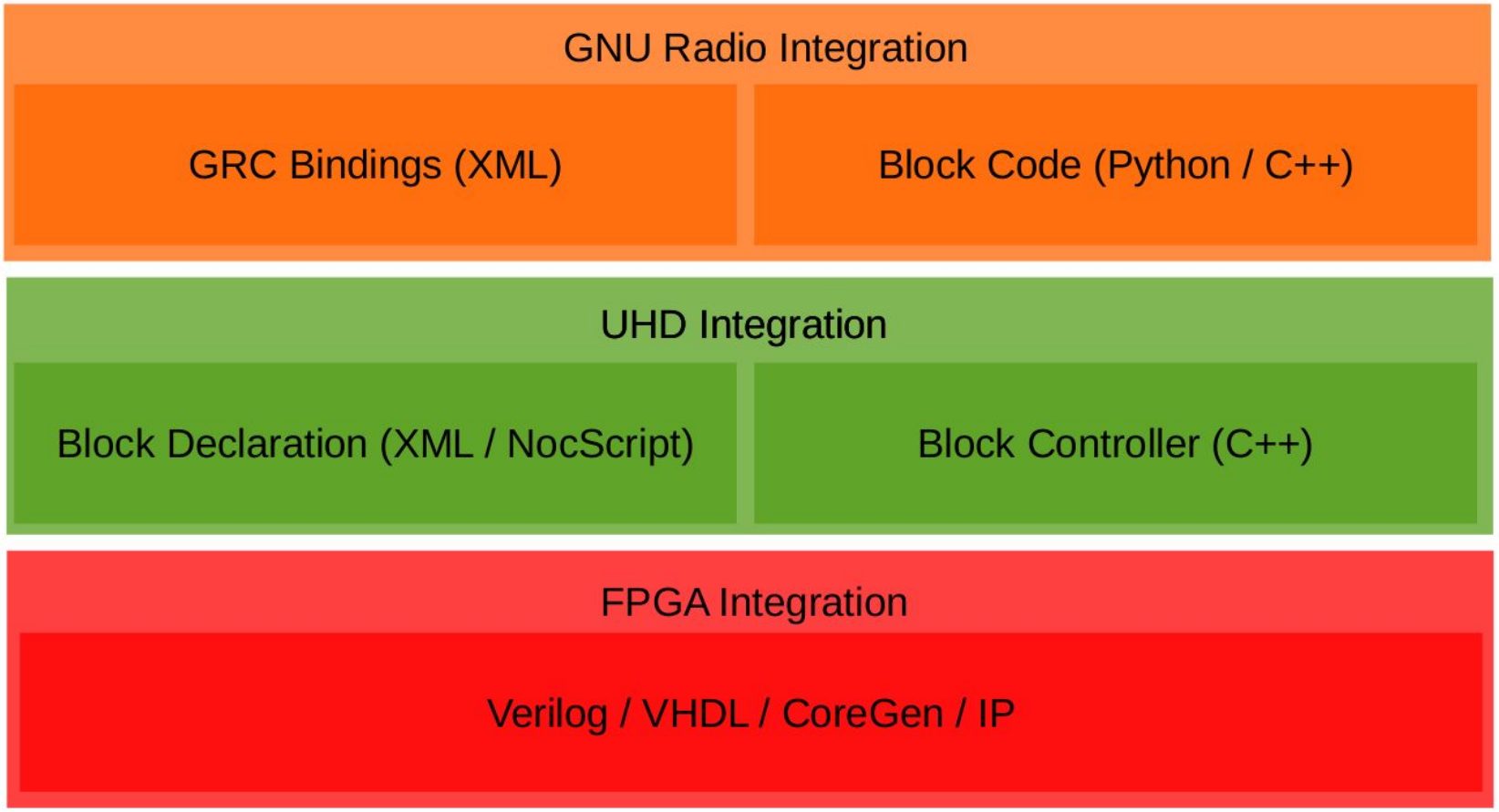
- Who am I?
- Creating your own RFNoC Block
  - rfnocmodtool
  - uhd_image_builder
  - uhd_image_builder_gui
- Demo (parallel)
- Q&A

# Who am I?

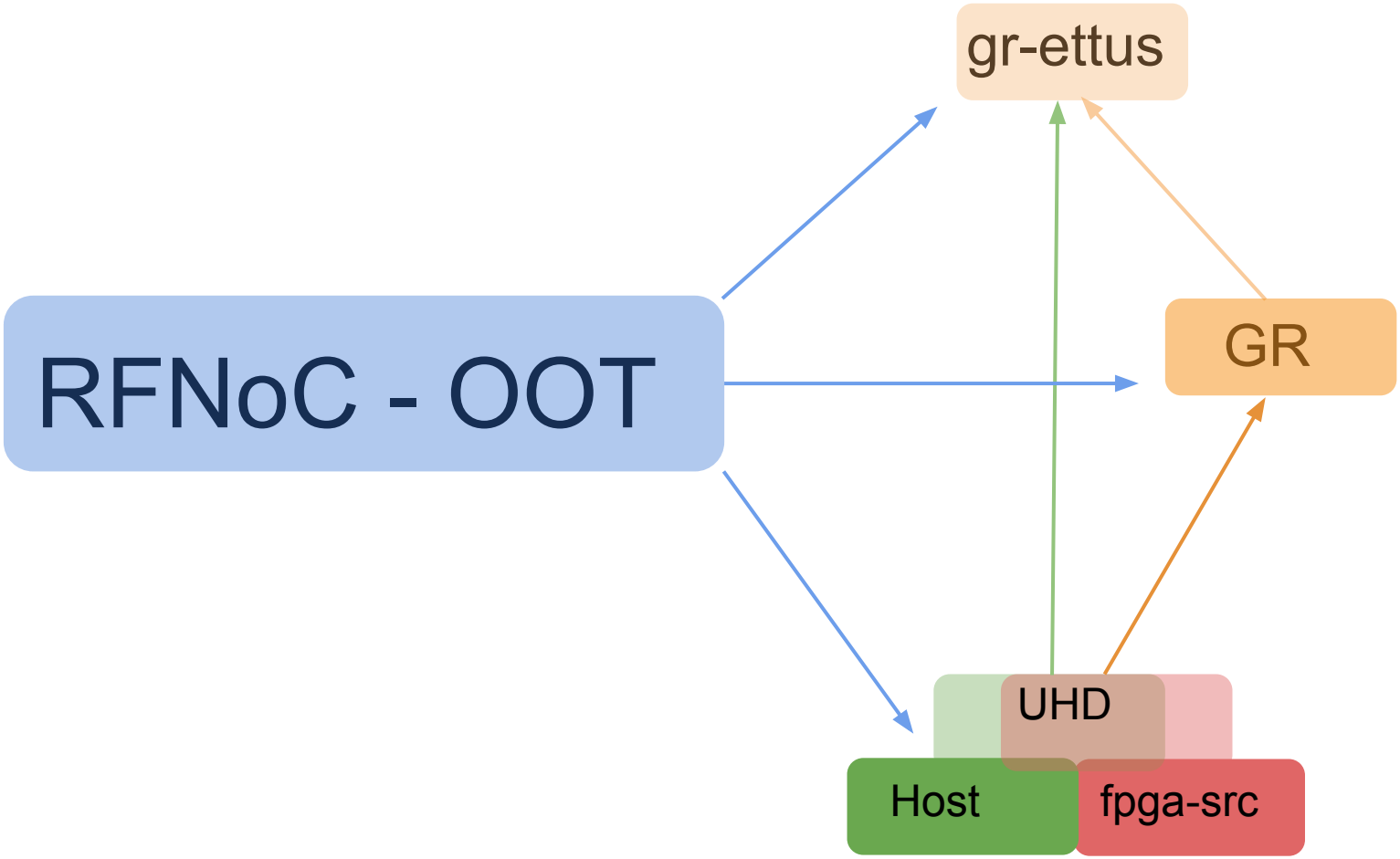
- MSEE Student - Comms and Info theory- KIT
- Former Ettus-Research's Intern
- Currently working at Ettus as "customer support" engineer

# Creating your own block

Ettus Research™
A National Instruments Company



# Creating your own block



# RFNoCModtool - Usage

Ettus Research™ A National Instruments Company

rfnocmodtool help
linux; GNU C++ version 4.8.4;Boost_105400;UHD_4.0.0.rfnoc-devel-162-g335a1317

Usage:
rfnocmodtool <command> [options] -- Run <command> with the given options.
rfnocmodtool help -- Show a list of commands.
rfnocmodtool help <command> -- Shows the help for a given command.

List of possible commands:

| Name | Aliases | Description |
|---|---|---|
| disable | dis | Disable block (comments out CMake entries for files) |
| info | getinfo,inf | Return information about a given module |
| remove | rm,del | Remove block (delete files and remove Makefile entries) |
| makexml | mx | Make XML file for GRC block bindings |
| add | insert | Add block to the out-of-tree module. |
| newmod | nm,create | Create a new out-of-tree module |

# RFNoCModtool - Use it!

Ettus Research™ A National Instruments Company

$ rfnocmodtool newmod fosdem

linux; GNU C++ version 4.8.4;Boost_105400;UHD_4.0.0.rfnoc-devel-162-g335a1317

Creating out-of-tree module in ./rfnoc-fosdem... Done.

Use 'rfnocmodtool add' to add a new block to this currently empty module.

# RFNoCModtool - Use it!

Ettus Research™ A National Instruments Company

## $ cd rfnoc-fosdem & rfnocmodtool add example

linux; GNU C++ version 4.8.4; Boost_105400; UHD_4.0.0.rfnoc-devel-162-g335a1317

RFNoC module name identified: fosdem

Block/code identifier: example

Enter valid argument list, including default arguments:

Add Python QA code? [Y/n]

Add C++ QA code? [y/N]

Block NoC ID (Hexadecimal):

Random NoC ID generated: 7B19F553110E186

Skip Block Controllers Generation? [UHD block ctrl files] [y/N]

Skip Block interface files Generation? [GRC block ctrl files] [y/N]

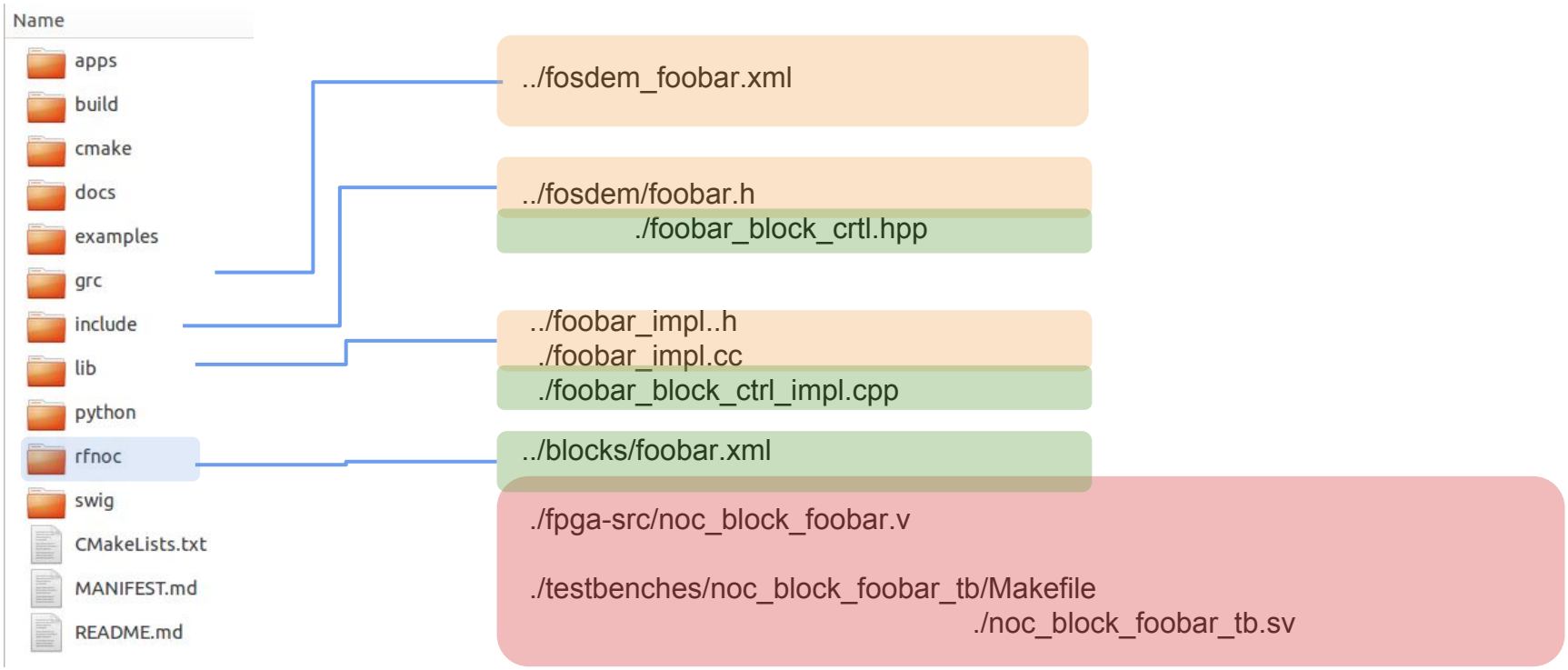
# RFNoCModtool - Use it!

Ettus Research™ A National Instruments Company

```
Adding file 'lib/foobar_impl.h'...
Adding file 'lib/foobar_impl.cc'...
Adding file 'include/fosdem/foobar.h'...
Adding file 'include/fosdem/foobar_block_ctrl.hpp'...
Adding file 'lib/foobar_block_ctrl_impl.cpp'...
Editing swig/fosdem_swig.i...
Adding file 'python/qa_foobar.py'...
Editing python/CMakeLists.txt...
Adding file 'grc/fosdem_foobar.xml'...
Adding file 'rfnoc/blocks/foobar.xml'...
Adding file 'rfnoc/fpga-src/noc_block_foobar.v'...
rfnoc/testbenches/noc_block_foobar_tb folder created
Adding file 'rfnoc/testbenches/noc_block_foobar_tb/noc_block_foobar_tb.sv'...
Adding file 'rfnoc/testbenches/noc_block_foobar_tb/Makefile'...
Adding file 'rfnoc/testbenches/noc_block_foobar_tb/CMakeLists.txt'...
```

# Basic structure of a generated OOT

rfnoc-fosdem




… and you are all ready to roll!!! (almost…)

# …but, as it is RFNoC…

Ettus Research™ A National Instruments Company

You have to program this block into a FPGA image, and the burn it into your device…

… WE GOT YOU COVERED ON THAT TOO!

# uhd_image_builder

Ettus Research™ A National Instruments Company

- former make.py
- Simple script that helps adding blocks into the FPGA w/o having to deal much with the code!
- @ {fpga-repo}/usrp3/tools/scripts/uhd_image_builder.py

# uhd_image_builder

Ettus Research™ A National Instruments Company

```
uhd_image_builder.py [-h] [-I INCLUDE_DIR [INCLUDE_DIR ...]] [-m MAX_NUM_BLOCKS] [--fill-with-fifos] [-o
OUTFILE] [-d DEVICE] [-t TARGET] [blocks [blocks ...]]

Generate the NoC block instantiation file

positional arguments:
  blocks                List block names to instantiate.

optional arguments:
  -h, --help                                        show this help message and exit
  -I --include-dir INCLUDE_DIR [INCLUDE_DIR ...]    Path directory of the RFNoC Out-of-Tree module
  -m --max-num-blocks MAX_NUM_BLOCKS,               Maximum number of blocks (Max. Allowed for x310|x300:10
                                                    , for e300: 6)
  --fill-with-fifos                                 If the number of blocks provided was smaller than the max
number,
                                                                fill the rest with FIFOs
  -o OUTFILE, --outfile OUTFILE                             Output /path/filename - By running this directive, you won't
build your                                                          IP
  -d DEVICE, --device DEVICE                        Device to be programmed [x300, x310, e310]
  -t TARGET, --target TARGET                        Build target - image type [X3X0_RFNOC_HG,
X3X0_RFNOC_XG,                                                      E310_RFNOC_sg3...]
```

# Getting hands on uhd_image_builder

Ettus Research™ A National Instruments Company

```
$ ./uhd_image_builder.py fft foobar fft window
    -I {OOT_moddir/rfnoc-fosdem}/rfnoc/fpga-src/
    -d x310
    -t X310_RFNOC_HG
    -m 5
    --fill-with-fifos
    [-o let_me_see_results.v]
```

# uhd_image_builder



The previous command will:

- Set up your vivado environment for the python session
- Generate a instantiation file with ...
- ... two ‘FFT’ blocks, one ‘Window’ block and one ‘demoConference’ block
- ... one ‘FIFO loopback’ block
- Either:
  - … save a file that will show how the instantiation file would look like
  - … start the FPGA image build right away!

“But I don’t recall where my OOT is, and I’d have to open a new shell to look for it…”

“I don’t know the names of the blocks I want to add!”

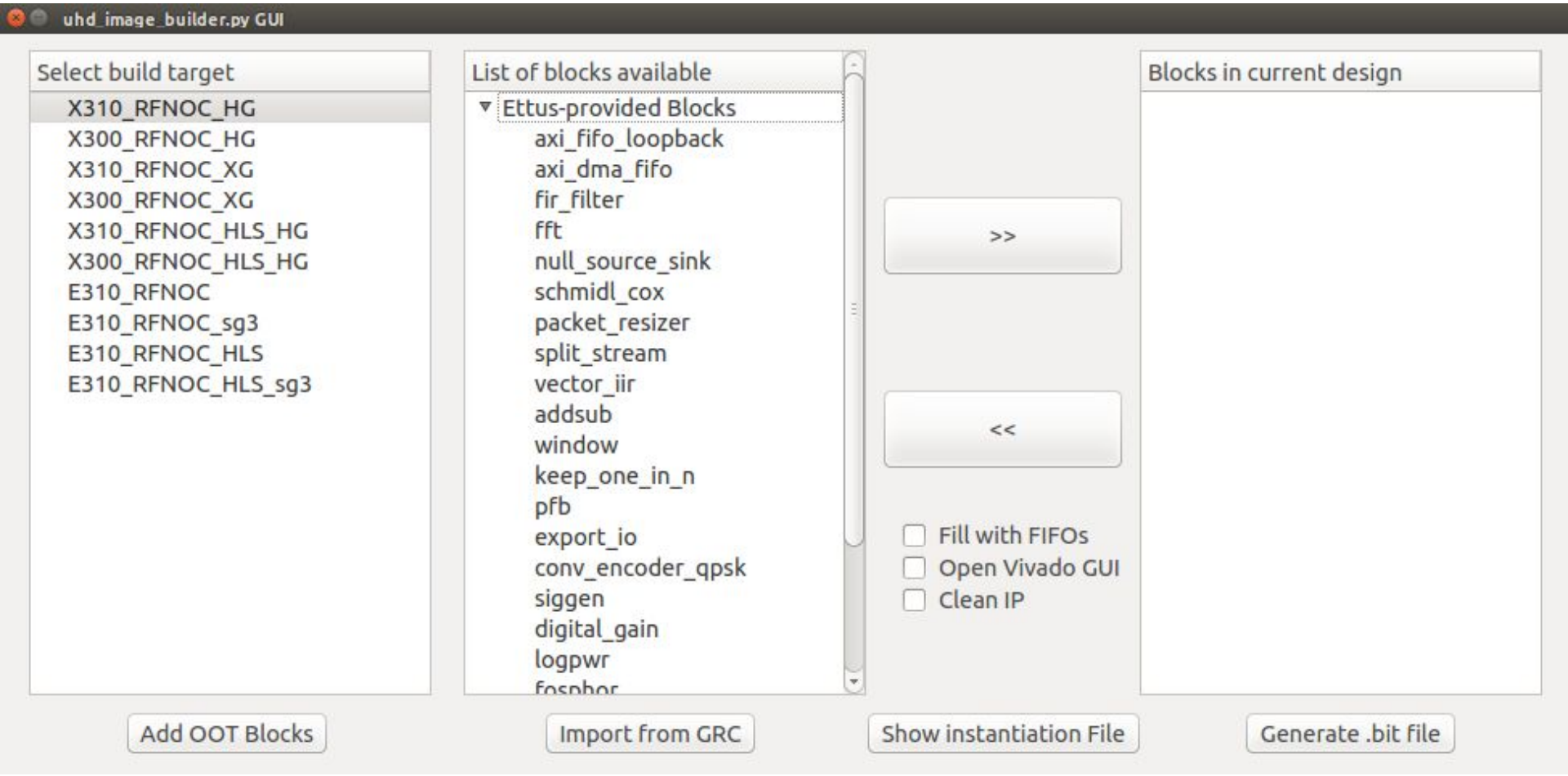
“I don’t really use terminal that much...”

“Do I have to know by heart the name of the already provided blocks to add them?”

“I always miss one of the targets of the make.py and I notice two hours later, when the wrong FPGA images finish it’s build!”

We STILL GOT YOUR BACK!

# uhd_image_builder_gui



# … that was quite a rush, huh? Do you have some questions?

Please do not hesitate to contacting us!

- Register to the mailing lists and ask away:

http://gnuradio.org/redmine/projects/gnuradio/wiki/MailingLists

(Please do so for both USRP-users and discuss-gnuradio lists)

# Do you want to give it a try?

Follow our guide in our Knowledge base!

https://kb.ettus.com/Getting_Started_with_RFNoC_Development

And if you still have more questions, post them in the mailing list.

# Thank you for your attention!