



# An Overview of Grand Central and the Zoo

## From the perspective of Nathan West

Nathan West


U.S. Naval Research Laboratory

July 20, 2015

Disclaimer: Eric Statzer of BR Envision is the author of grand central and the zoo. I have been using it for the last two-ish weeks and have added some features and other minor development.



# My Goal

hippo 

Running flowgraphs

- \* canary
- \* snapper


Init string

Status

Command

- \* stop
- \* retune
- \* start\_record
- \* stop\_record

Command arguments

hippo 

New flowgraph

- \* canary
- \* snapper

options

filename

channels

- 1
- 2
- 3



# Android Project Perspective

my app

lte\_dl\_detect

my other flowgraphs

inherited from giraffe::flowgraph\_base

jlibgrand

giraffe::grand\_central

giraffe::flowgraph\_base

giraffe::snapper

gr flowgraph control panel

mobilesdr\_template

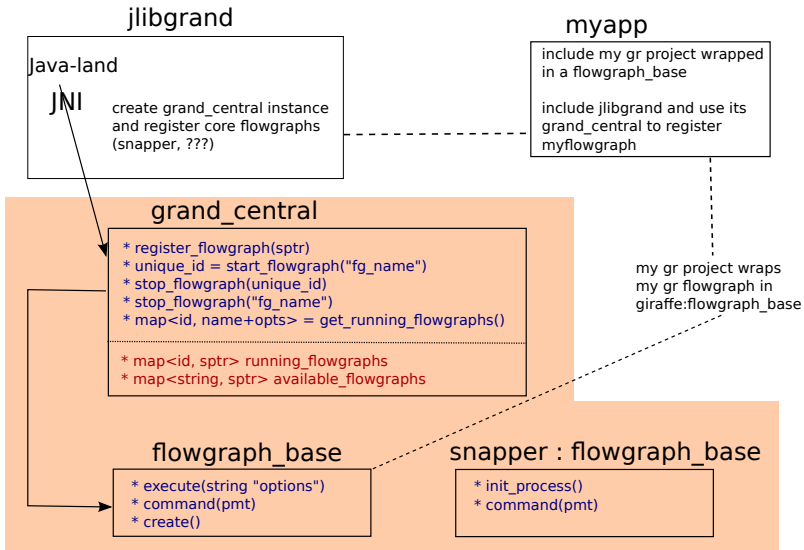
spectrum viewer

zmq source

core activities



# Some Nameless Perspective That's Useful





# Grand Central's Implementation

## Public:

- `register_flowgraph(fg_base_sptr)`
- `uint start_flowgraph(std::string name)`
- `command_by_id(uint, pmt&)`
- `stop_flowgraph_by_id(uint)`
  
- `int is_running_by_name(std::string name)`
- `vector<string> get_running_flowgraphs()`
- `vector<uint> id_from_name(std::string name)`

## Internally:

- `map<uint, fg_base_sptr> running_flowgraphs`
- `map<string, fg_base_sptr> available_flowgraphs`
- `map<uint, thread*> fg_threads()`



# A Look in flowgraph\_base

The important bits:

- `execute()` – do everything for everyone
  - 1 `Create a topblock`
  - 2 `init_src()`
  - 3 `init_phy()`
  - 4 `init_process()`
  - 5 `start_top_block()`
  - 6 `cnc_loop()`
- `create()` – a virtual factory method
- `init_process()` – virtual block connector



# IQ sources

`init_src()`

- set the src member
- either a file source or osmosdr source

`init_phy()`

- Builds `src`→`phy`→`null_sink`
- `phy` is a hier block
- `spec_est::sub` → `fft` → `complex` to `mag2` → `10log10`→`spec_est::protoize`→`spec_est::pub_msg_sink`



# `init_process()` – where you do something!

- Use `tb` (top block) created for you
- Connect your graph to `src`





# Running

- `start_top_block()` does what you expect
- `cnc_loop()` has zmq poll for commands and protoizes them



# Building an app

- Build a new "flowgraph" that inherits from flowgraph\_base
- Create a new Android Studio project
- Add jlibgrand and mobilesdr\_template as libraries
- register your flowgraphs with grand\_central



My priorities for the week:

- 1 get lte\_dl\_detect working in an app
- 2 get app flowgraph management mostly automatic
- 3 document creating new apps

Wishlist:

- grand central interrogates app for config
- cleaner failures– what happens when a GR graph dies?
- flowgraphs that can transparently share a source via zmq
- pcap viewer activity