SDR++

A modular cross-platform SDR utility

```
$ whoami
```

Hello World! — 2002
C/C++ — 2011
SDR (use) — 2017
SDR (dev) — 2019
SDR++ — 2020
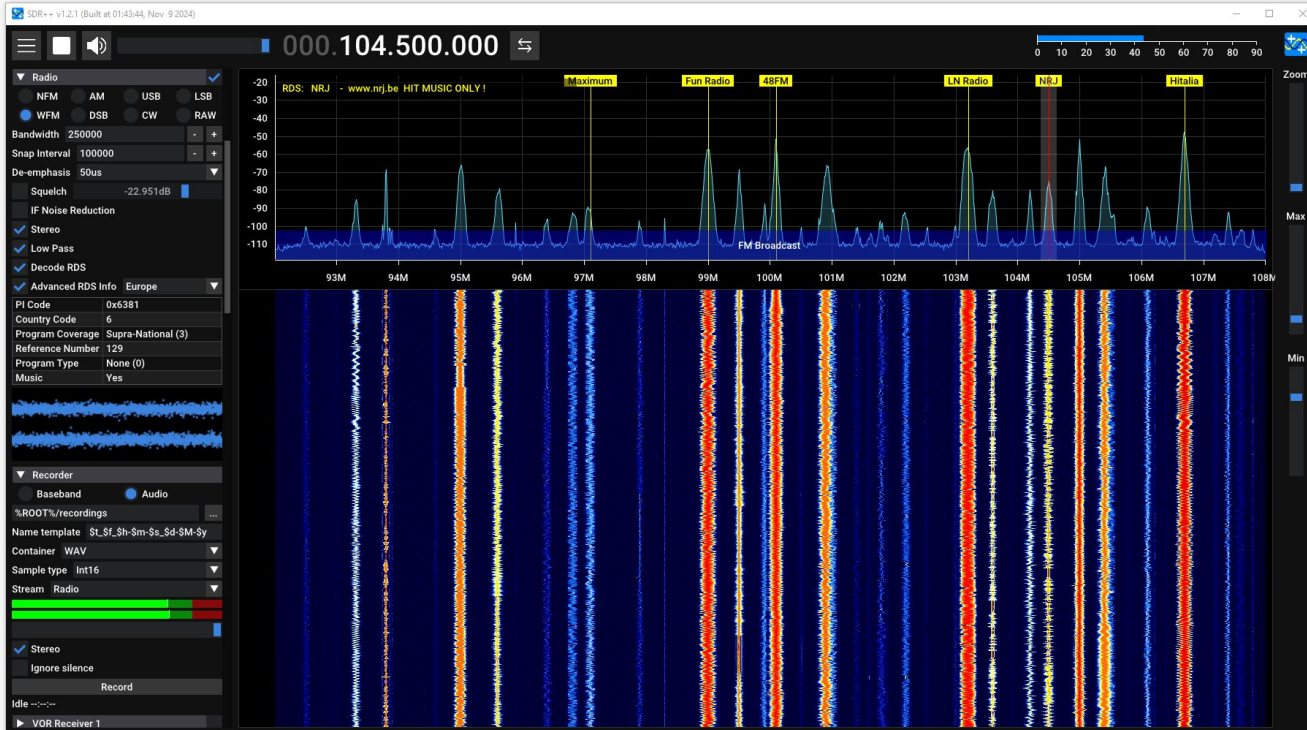Amateur Radio License — 2021
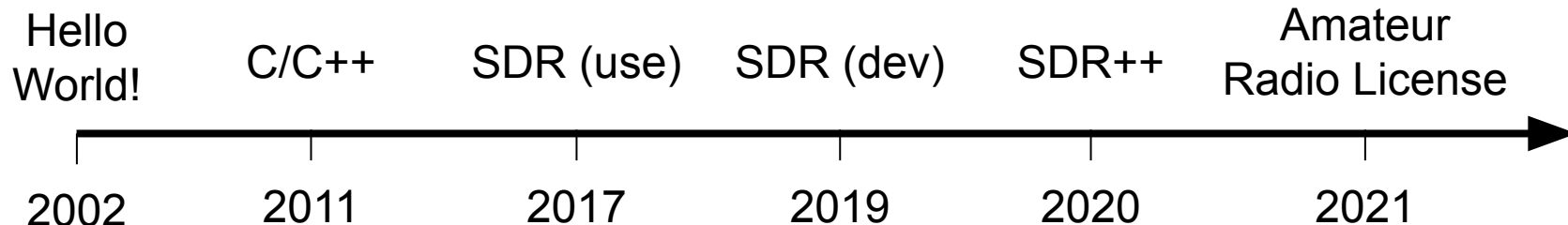
Education

BSc. Engineering (2020 - 2023)

MSc. Aerospace Engineering (2023 - 2025)

PhD. Electrical Engineering (2025 - TBD)

What is SDR++?

Wide Hardware Support
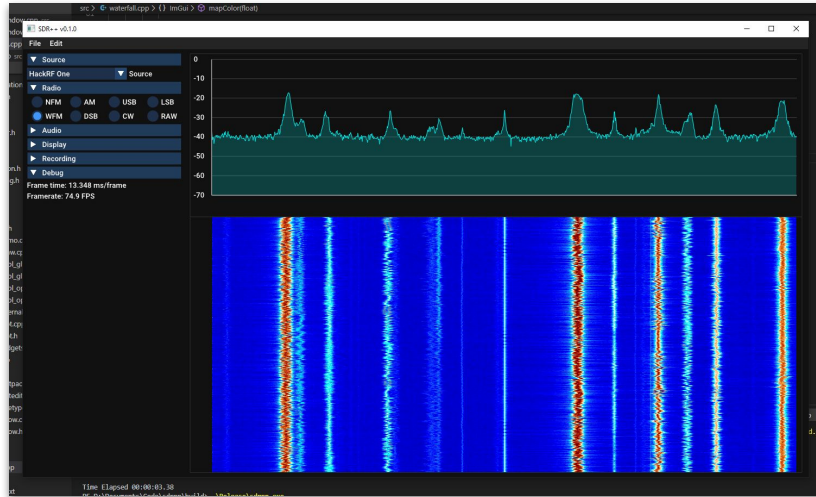
Wide OS Support

Fully Modular

Multi-VFO

RX only (for now)
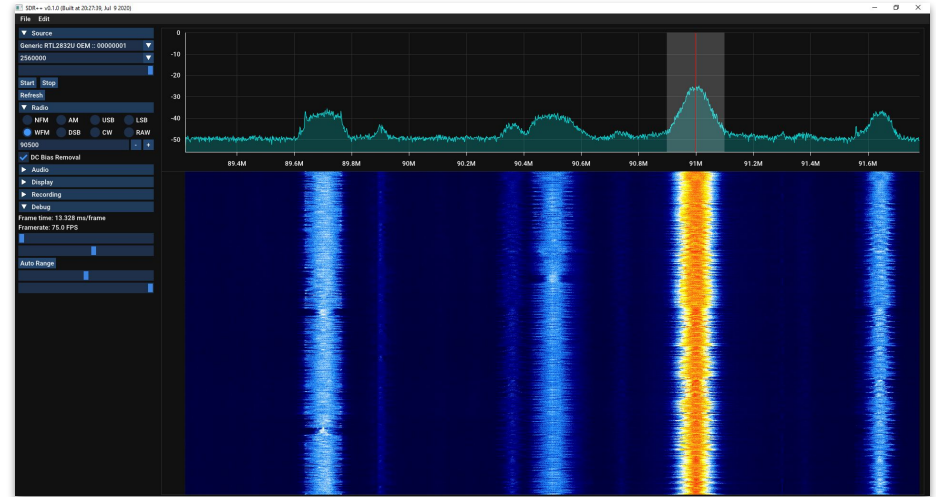
Fully custom DSP

GPLv3 License

A brief history segment…

June 2020

July 2020

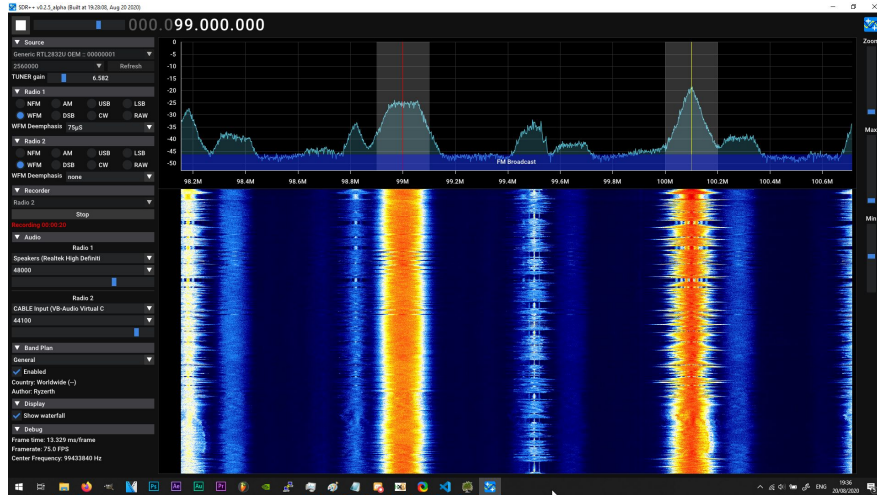Basic Proof of Concept

Layout inspired by SDR#

Waterfall code rewritten

First public release!
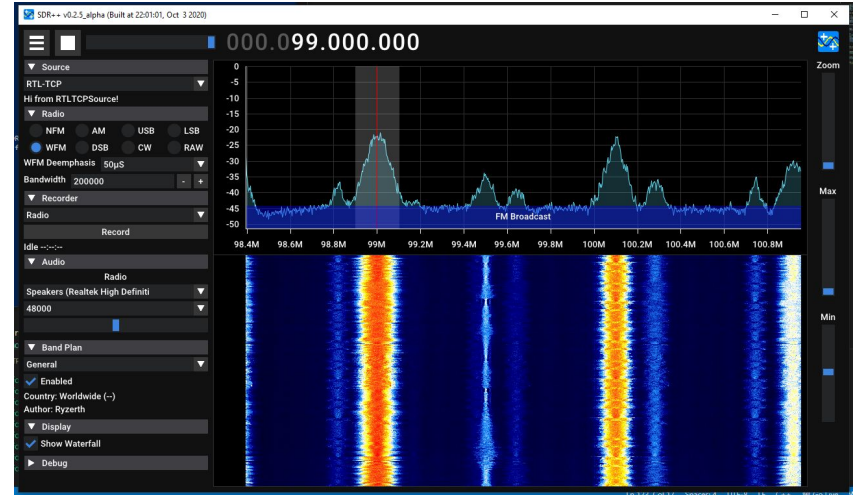
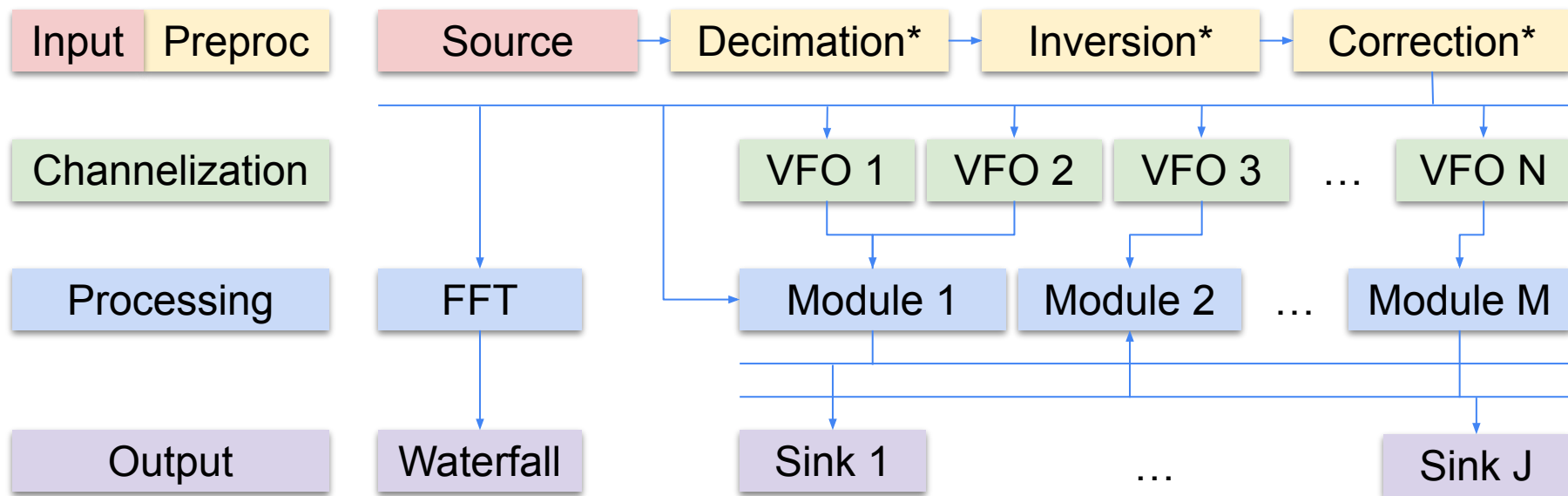# A brief history segment…

August 2020

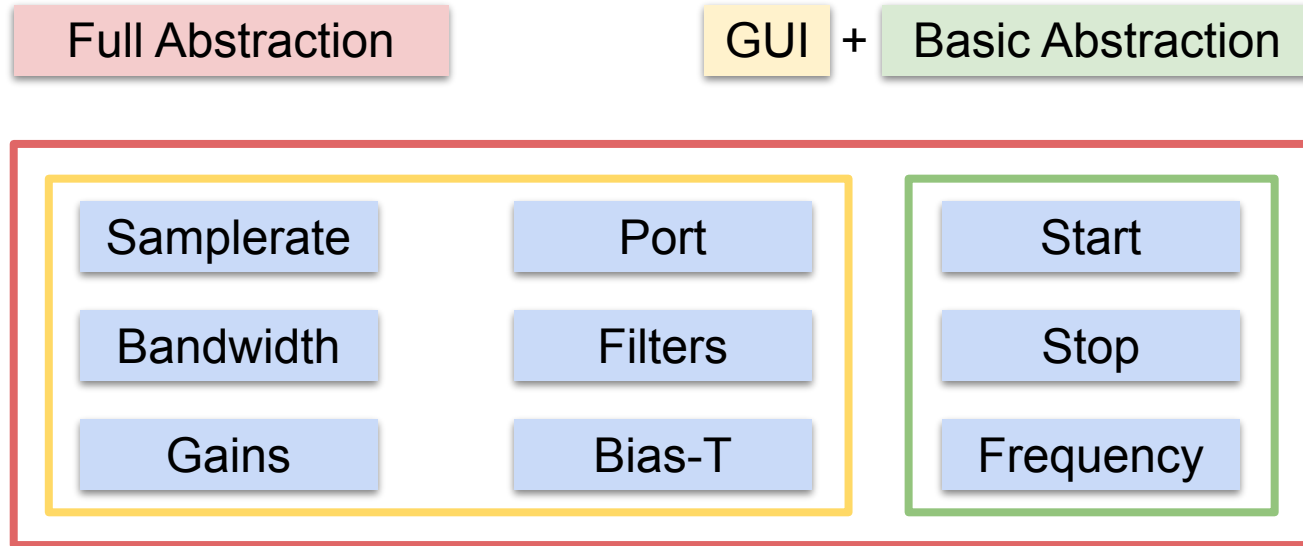Module system

Band plans

October 2020

Source system

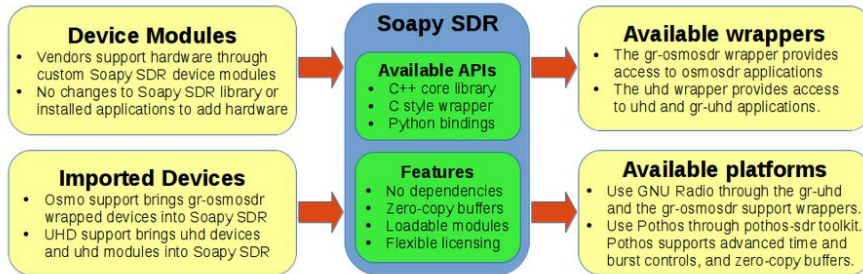Last alpha release!

The SDR++ Signal Path

* Optional

Device Handling

Two common implementations.

Device Handling - Full Abstraction

Most common SDR device handling scheme due to libraries like SoapySDR.

| **Device Modules**
• Vendors support hardware through custom Soapy SDR device modules
• No changes to Soapy SDR library or installed applications to add hardware | **Soapy SDR**

Available APIs
• C++ core library
• C style wrapper
• Python bindings | **Available wrappers**
• The gr-osmosdr wrapper provides access to osmosdr applications
• The uhd wrapper provides access to uhd and gr-uhd applications. |

| **Imported Devices**
• Osmo support brings gr-osmosdr wrapped devices into Soapy SDR
• UHD support brings uhd devices and uhd modules into Soapy SDR | **Features**
• No dependencies
• Zero-copy buffers
• Loadable modules
• Flexible licensing | **Available platforms**
• Use GNU Radio through the gr-uhd and the gr-osmosdr support wrappers.
• Use Pothos through pothos-sdr toolkit. Pothos supports advanced time and burst controls, and zero-copy buffers. |

Source:
github.com/pothosware/SoapySDR/wiki

Platforms

Platforms are graphical applications and command line utilities that use SoapySDR to interface with the ecosystem of SDR hardware.

Graphical

- QSpectrumAnalyzer is a python based spectrum analyzer based on SoapySDR python bindings.
- Welle.io DAB/DAB+ is an open source DAB and DAB+ software defined radio (SDR). It supports high DPI and touch displays and it runs even on cheap computers like Raspberry Pi 2/3 and 100€ China Windows 10 tablets.
- Cubic SDR is a cross-platform Software-Defined Radio application which allows you to navigate the radio spectrum and demodulate any signals you might discover.
- GQRX is a C++ based spectrum analyzer which uses SoapySDR through bindings in GrOsmoSDR.
- SDRangel is an Open Source Qt5/OpenGL 3.0+ SDR and signal analyzer frontend to various hardware.
- HABDEC is a RTTY decoder for High Altitude Balloons.
- LinHPSDR is an HPSDR application for Linux based on GTK+ 3.
- QUISK is a graphical receiver and transmitter application.
- SigDigger is a free digital signal analyzer based on Suscan
- SdrGlut is a simple software defined radio player. Using glut for drawing and glui for its dialogs makes it tiny compared to programs that use QT5 or wxWidgets.
- QRadioLink is a VOIP GNU/Linux SDR transceiver application using Internet protocols for communication.
- OpenWebRX is a multi-user SDR receiver that can be operated from any web browser.
- SDR++ is a cross-platform and open source SDR software with the aim of being bloat free and simple to use.
- Abraca DAB radio is a DAB/DAB+ Qt6 cross-platform receiver application.

Powerful and avoids needing hardware but complex to implement (well) and inflexible.

Device Handling - Basic Abstraction with GUI

One of the earliest SDR device handling scheme.
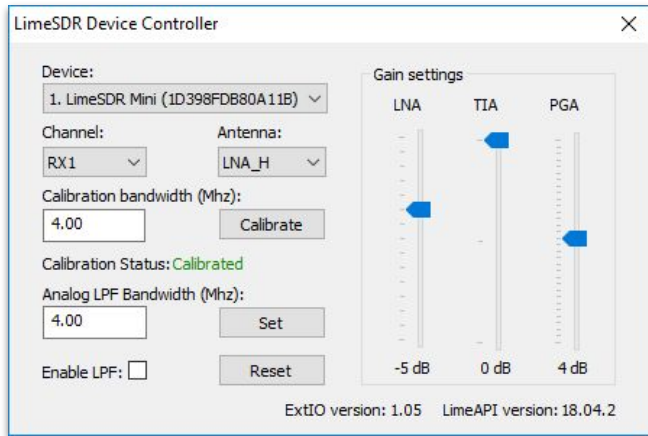
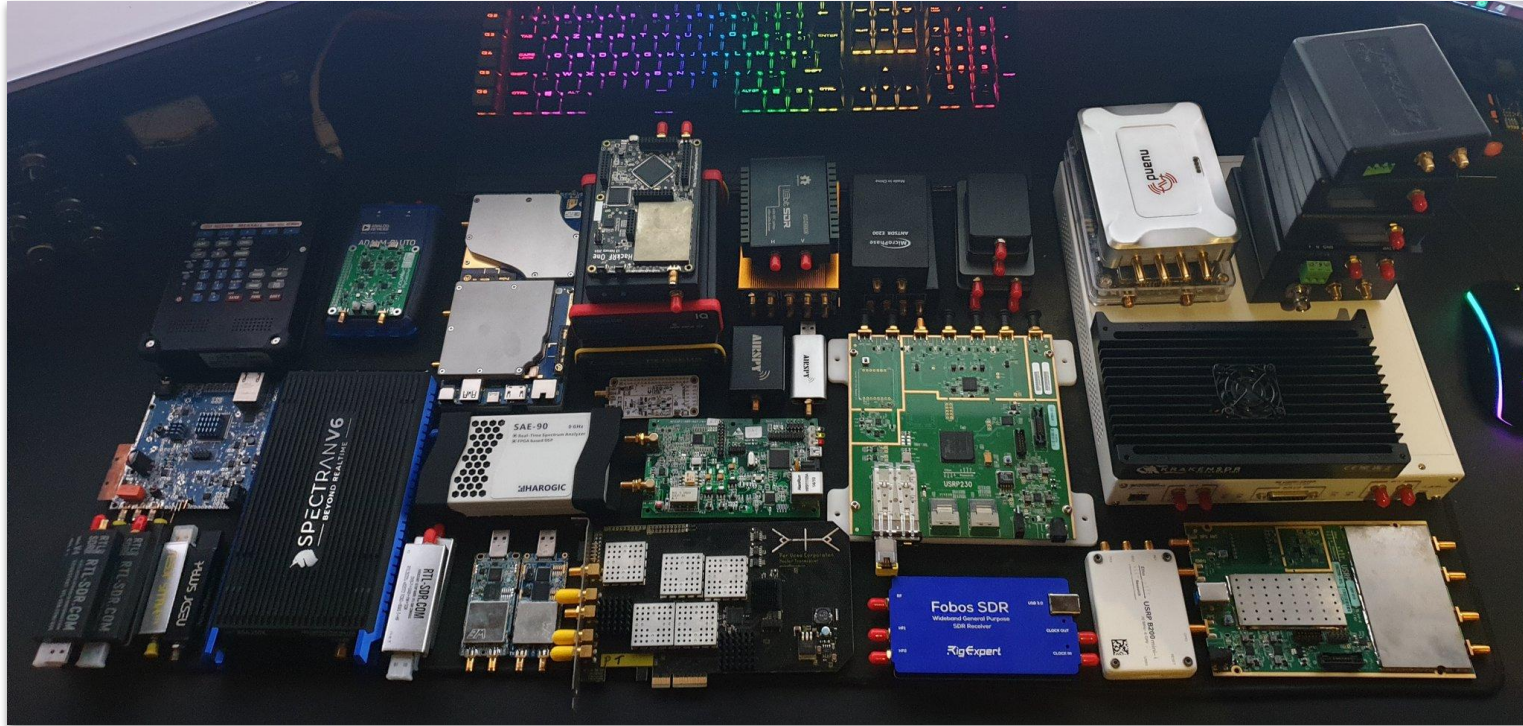| ExtIO Standard |
|---|
| Winrad (2006) |
| HDSDR (2009) |
| Win32 ONLY! |

Software-specific API
SDRsharp (2012)
SDRangel (2015)
SDR++ (2020)

Flexible but no programmatic configuration and requires having the hardware.
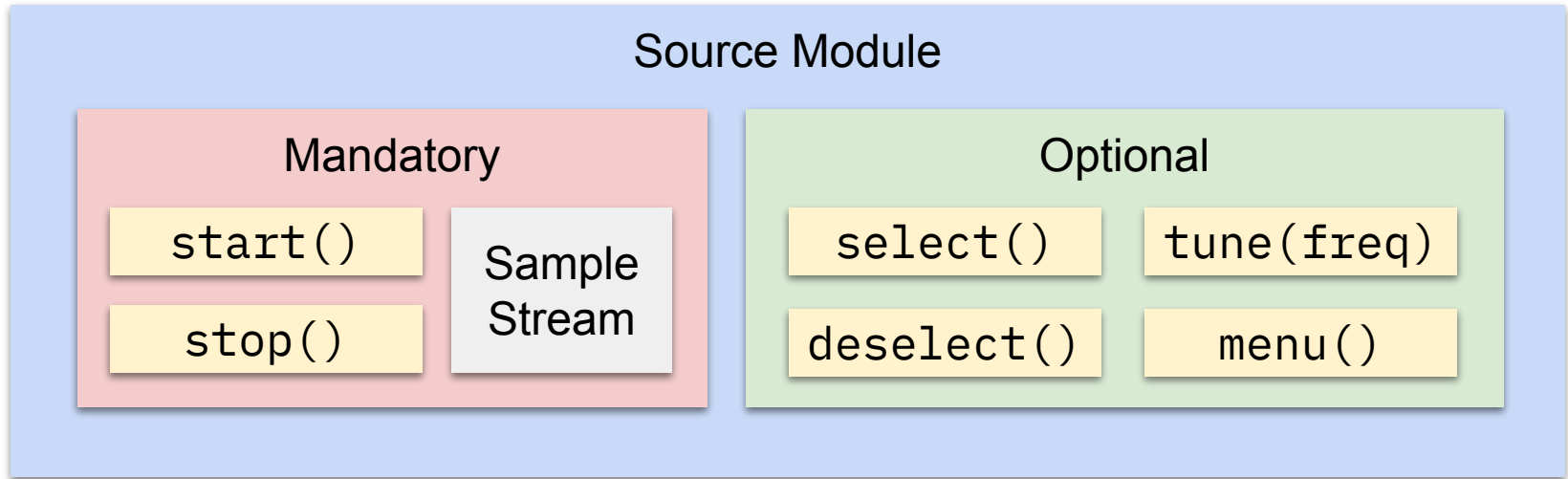
Device Handling - Basic Abstraction with GUI

Hardware Donators
- Aaronia AG
- Airspy
- Alex 4Z5LV
- Analog Devices
- CaribouLabs
- Deepace
- Ettus Research
- Harogic
- Howard Su
- MicroPhase
- Microtelecom
- MyriadRF
- Nuand
- RFNM
- RFspace
- RigExpert
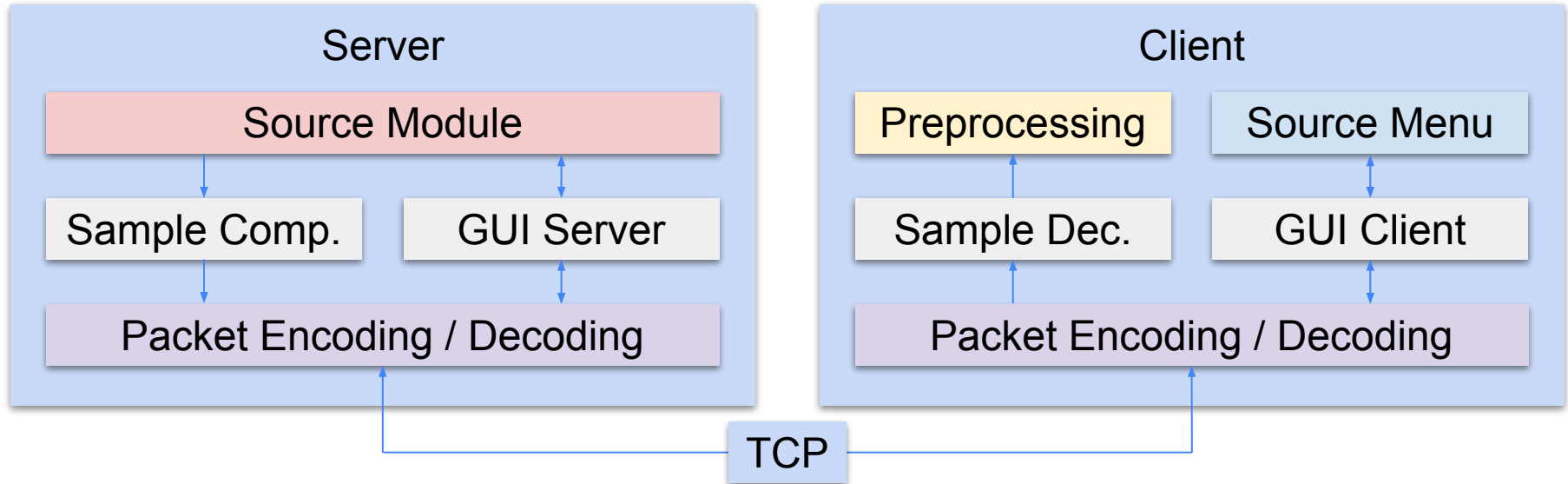- RTL-SDRblog
- SDRplay

Device Handling - SDR++

Source modules must implement start and stop while tune and menu are optional.

Enough to implement support for any SDR hardware, protocol or recording format.
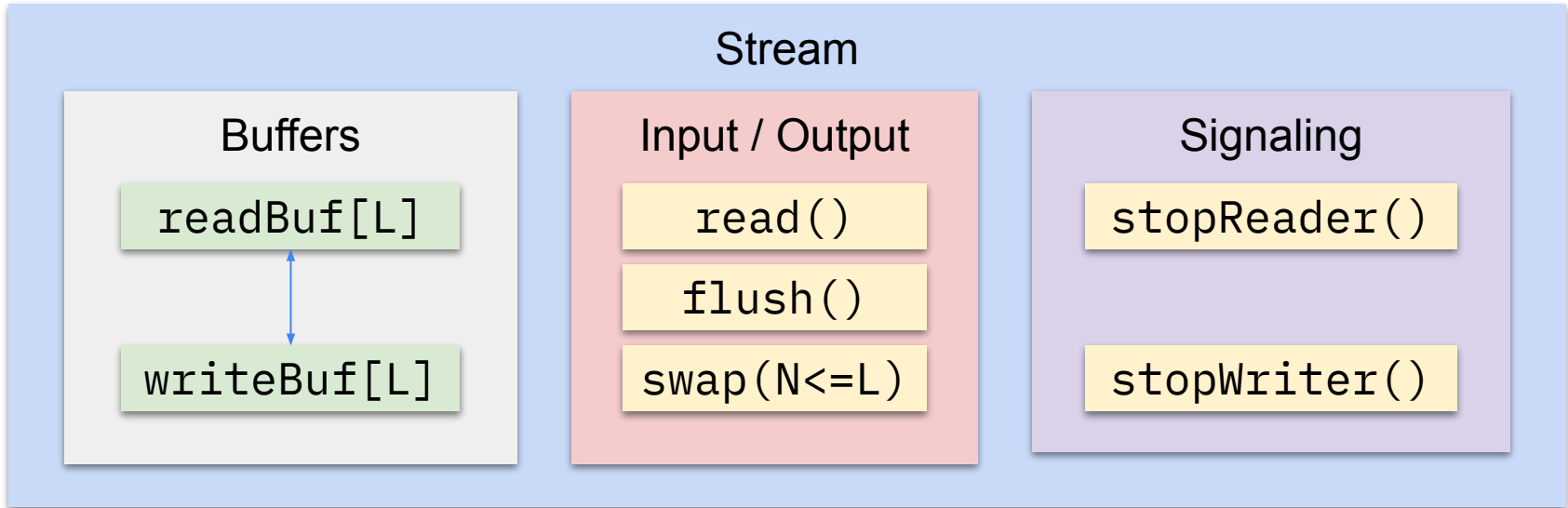
Device Handling - SDR++ Server

Remote usage of SDR hardware is often necessary.

The server system transports samples, commands and GUI elements.
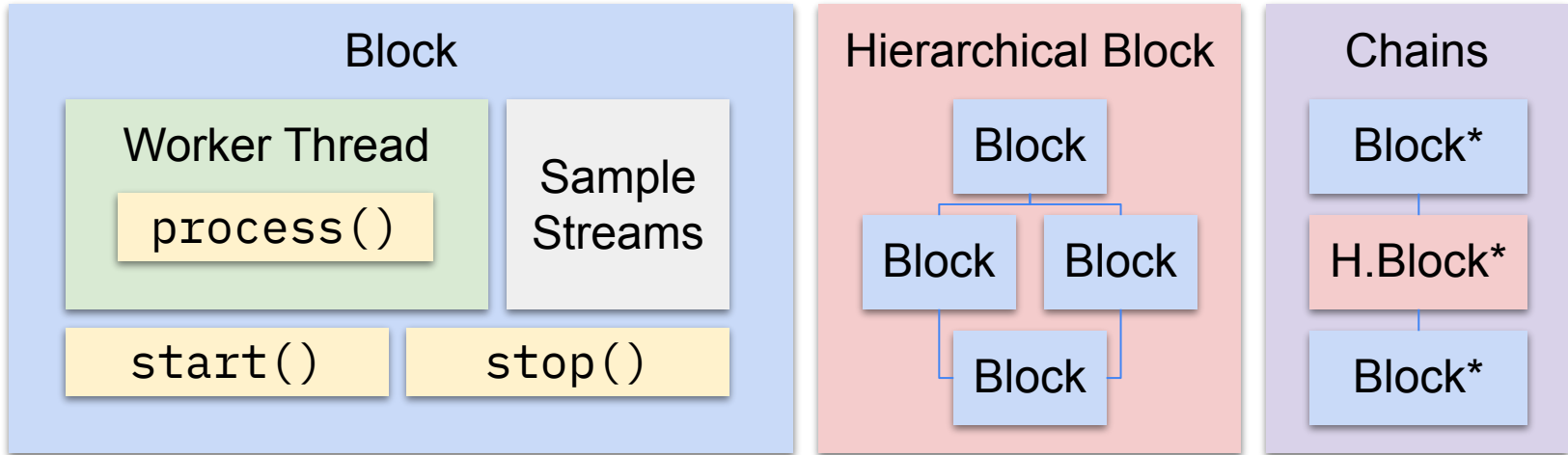
Digital Signal Processing - Streams

Streams move samples. They encapsulate two buffers, IO and signaling functions.

Using fixed size swappable buffers is fast but uses more memory.

Digital Signal Processing - Blocks

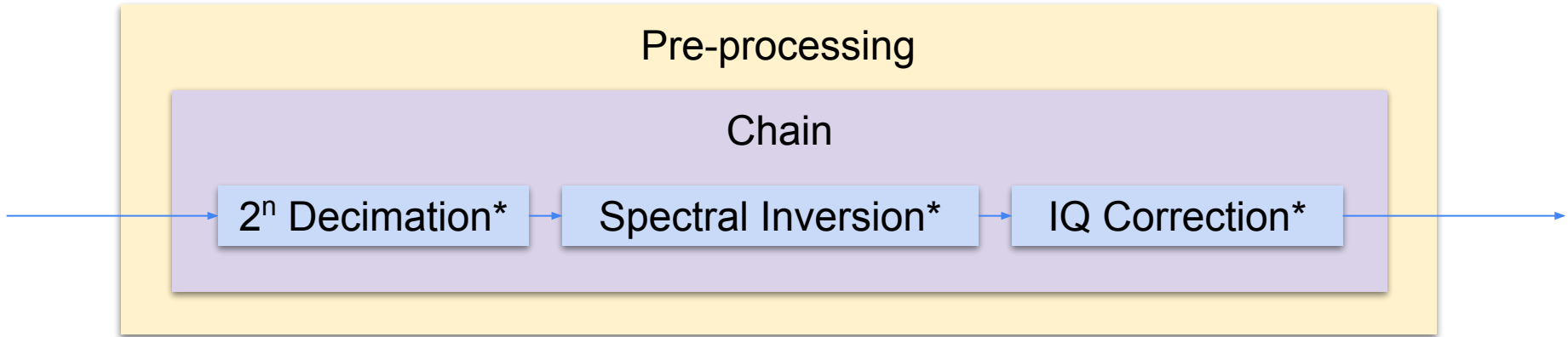Blocks encapsulate a processing function, worker thread and streams.

Multiple blocks can be combined in hierarchical blocks or chains.

* Bypassable

Pre-processing

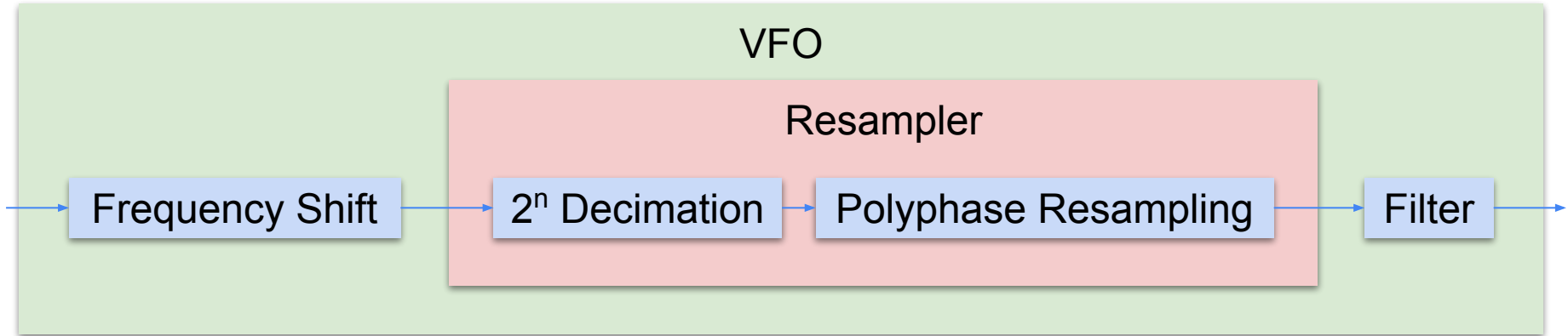Pre-processing is implemented as a chain of DSP blocks.

Decimation is placed first to minimize the load on subsequent blocks.
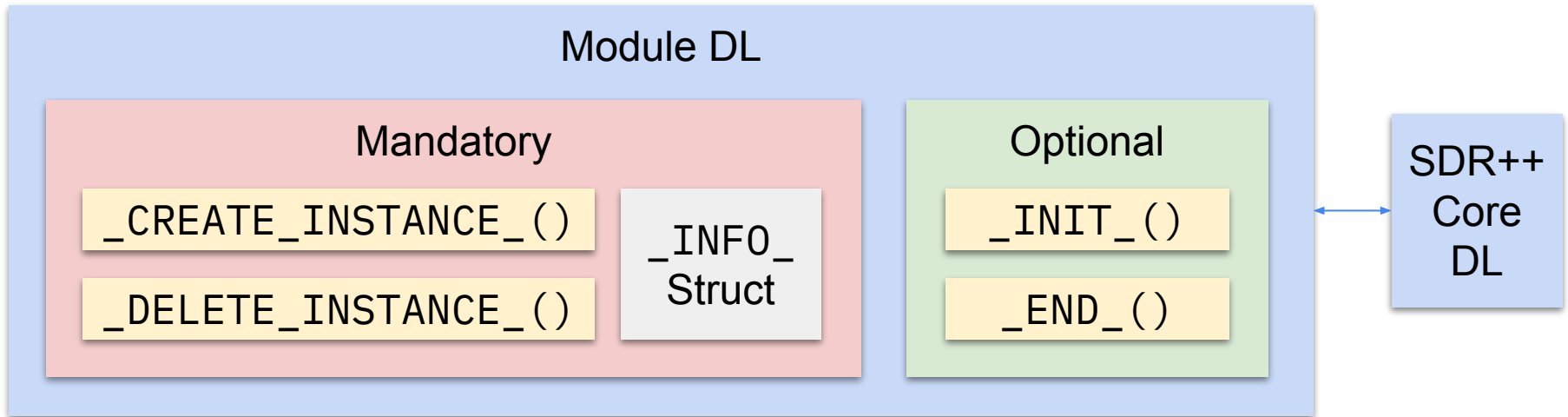
* Bypassable

Channelization

The raw IQ is frequency-shifted, then resampled and filtered.

Single thread to maximize efficiency. Minimum O.O.B. attenuation of 100dB.
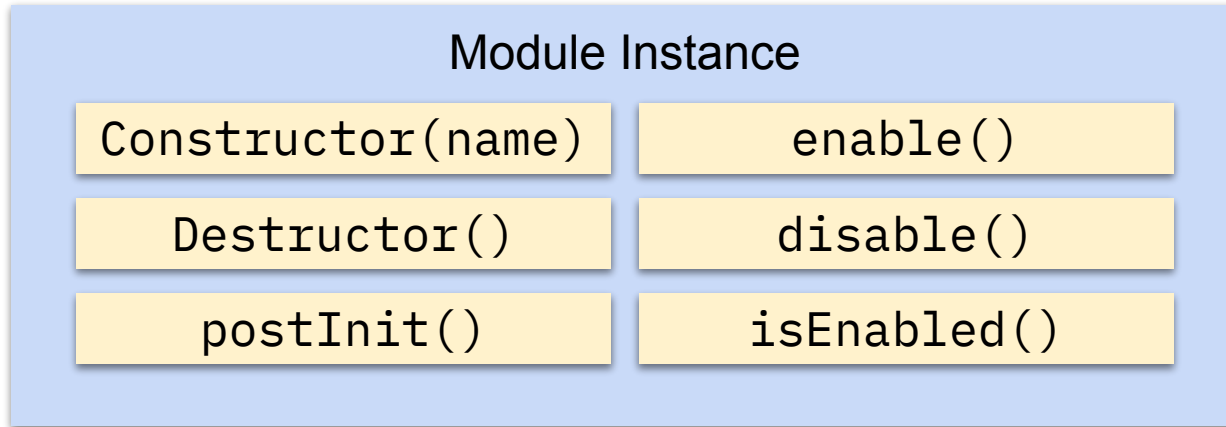
Modules - Dynamic Library

Modules are Dynamic Libraries that link themselves to the SDR++ core library.

Makes modularity trivial but means that ABI compatibility is easily broken.
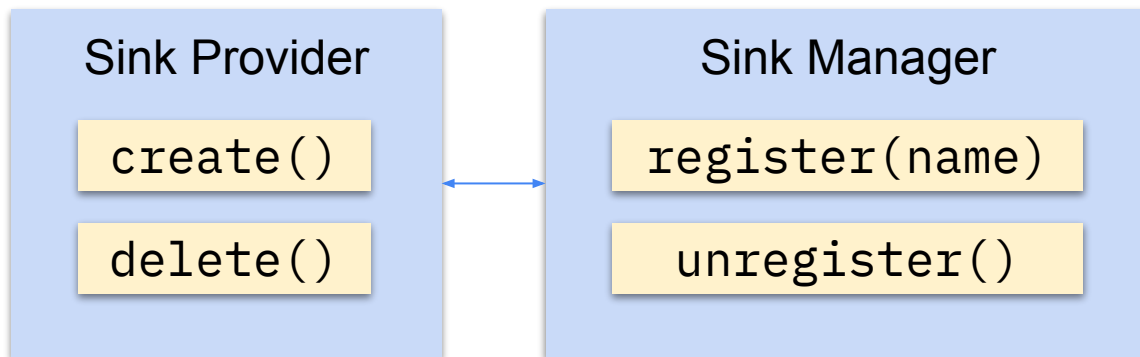
Modules - Instance

Module instances are referred by name and created by the module library.

Module instances may access any public object within the core.

Sinks - Providers

Sink providers register themselves by name.

```
Sink Provider          Sink Manager
  create()             register(name)
  delete()             unregister()
```
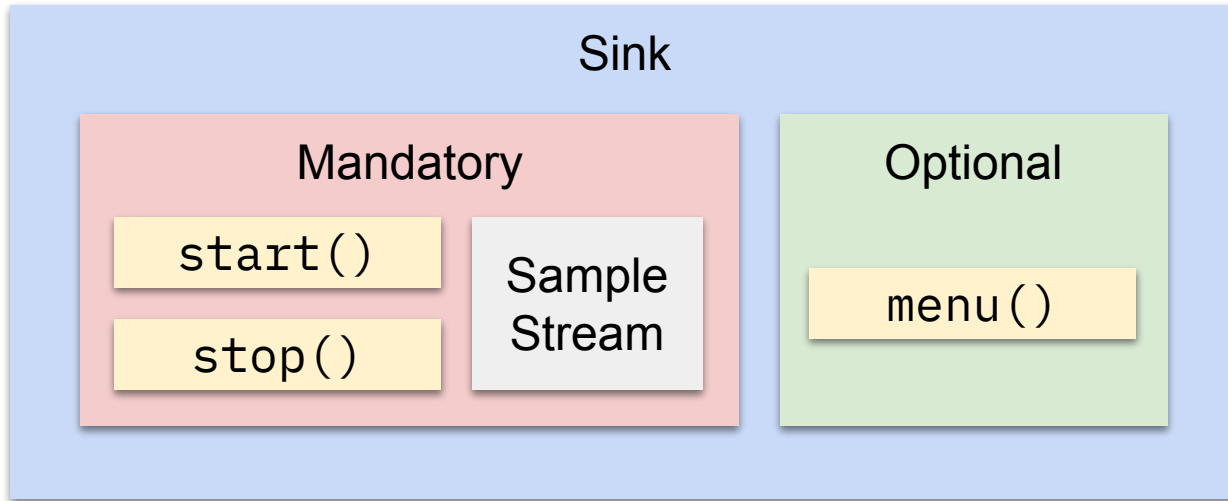
One provider can be selected to create the sink type for each audio stream.

Sinks - Instances

Sink instances are abstracted similarly to sources.

Very flexible and target agnostic.

SDR++

It's demo time!

What's next? Things that WON'T change

SDR++ 1.0 did a lot of things right.
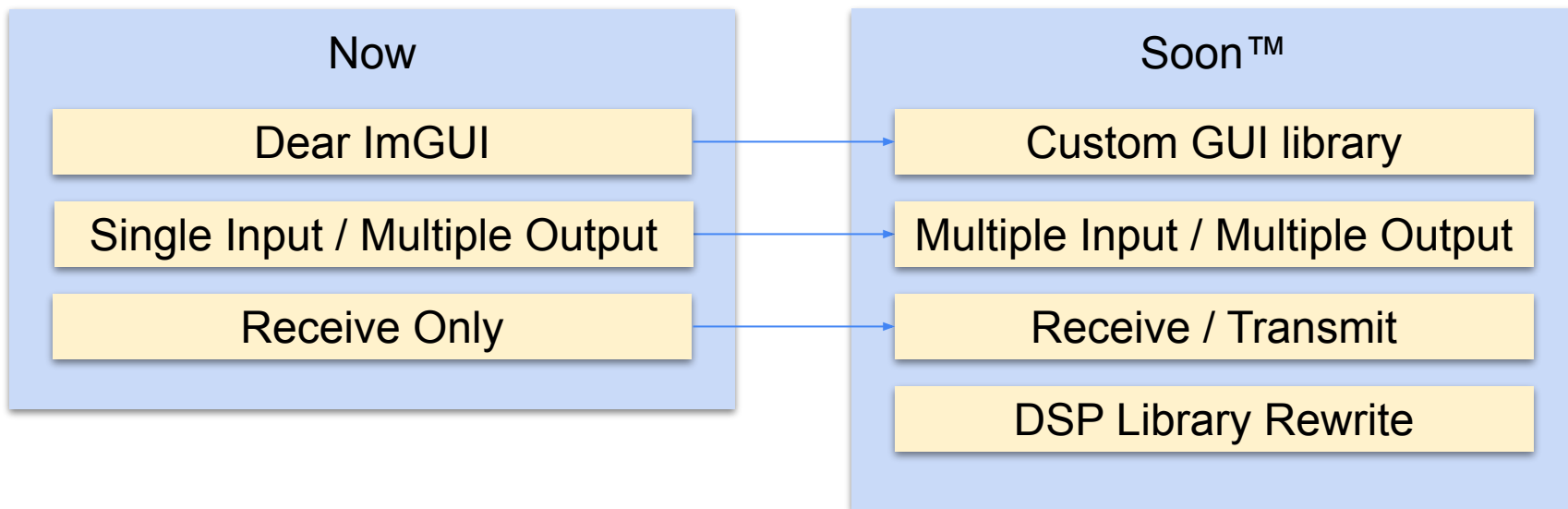
GUI Layout	OS Support
General Workflow	Hardware Support

If it ain't broke, don't fix it!

What's next?

Much of the software has to change in SDR++ 2.0.

Now	Soon™
Dear ImGUI →	Custom GUI library
Single Input / Multiple Output →	Multiple Input / Multiple Output
Receive Only →	Receive / Transmit
	DSP Library Rewrite

Near complete rewrite using lessons learned from 1.0.

Would not be possible without the Patreon supporters!

- Bob Logan
- Christian Häusler
- Croccydile
- Dale L Puckett (K0HYD)
- Daniele D'Agnelli
- David Taylor (GM8ARV)
- D. Jones
- Dexruus
- EB3FRN
- Eric Johnson
- Ernest Murphy (NH7L)
- Flinger Films
- Frank Werner (HB9FXQ)

- gringogrigio
- Jandro
- Jeff Moe
- Joe Cupano
- KD1SQ
- Kezza
- Krys Kamieniecki
- Lee Donaghy
- Lee KD1SQ
- .lozenge. (Hank Hill)
- Martin Herren (HB9FXX)
- ON4MU
- Passion-Radio.com

- Paul Maine
- Peter Betz
- Scanner School
- Scott Palmer
- SignalsEverywhere
- Syne Ardwin (WI9SYN)
- W4IPA
- William Arcand (W1WRA)
- William Pitchford
- Yves Rougy
- Zipper

patreon.com/ryzerth

And many more!

SDR++

Thank you for your attention!