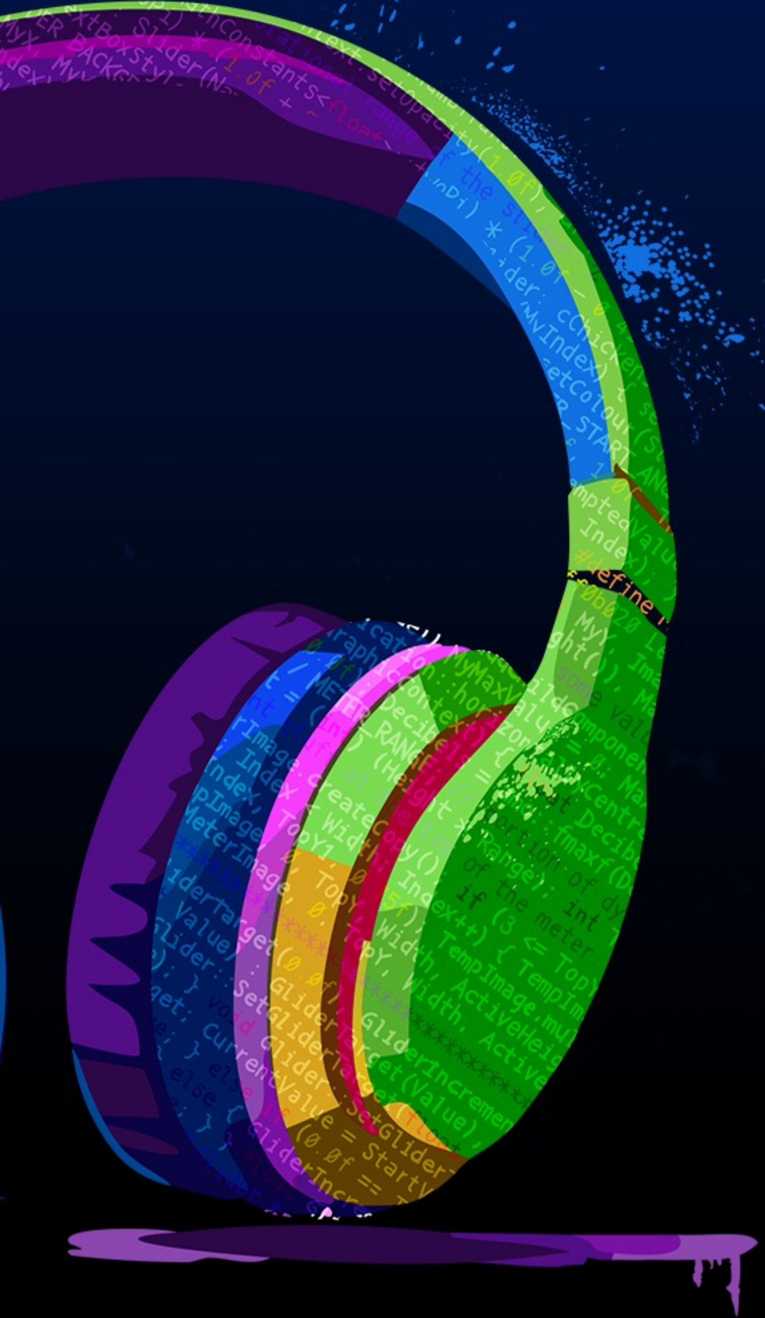


ADC24 Bristol

INTER-PLUGIN COMMUNICATION

BREAKING OUT OF THE CHANNEL STRIP

PETER SCIRI

Agenda

- (Who?)
- Why?
- What?
- How?
- Limitations

Who?

sonible | Graz, Austria

We make smart plugins

The Channel Strip = Living In Isolation

- Processing context: only a channel's samples
- Am I alone?
- Who else is around?
- What do they do?

Not that smart, huh?

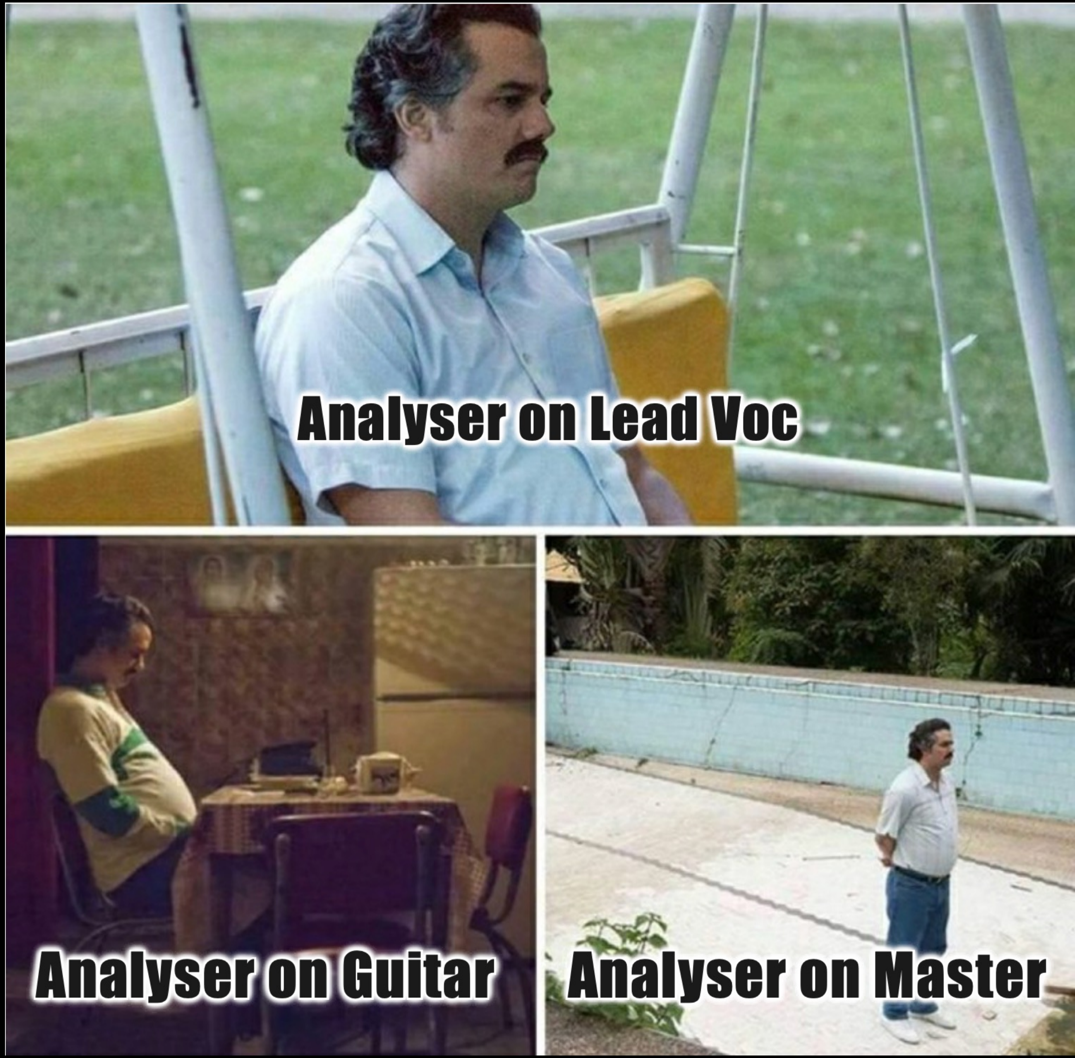
Why do Inter-Plugin Communication?

Other Cannels matter!

- A mix is more than just the sum of its components
  - Good mixing requires observing a multitude of channels at once
- Channel strip isolation
  - Requires working with multiple instances at once
  - Prevents smart decisions

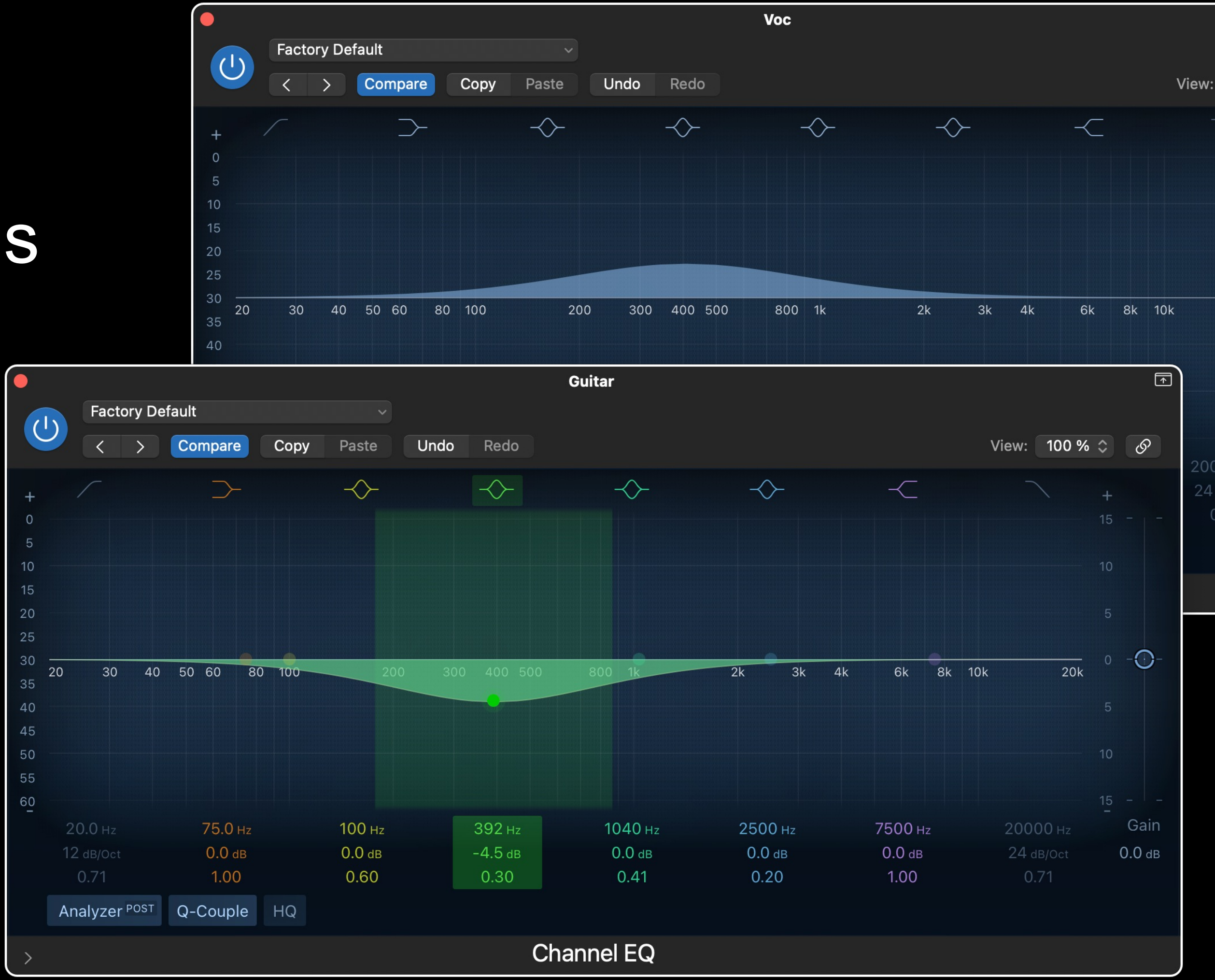
Example 1: Analyser

- One instance view shows multiple channel spectra at once
- No need for multiple windows
- Levels aligned 🎉
- Clashing frequencies easier to identify



Example 2: EQ

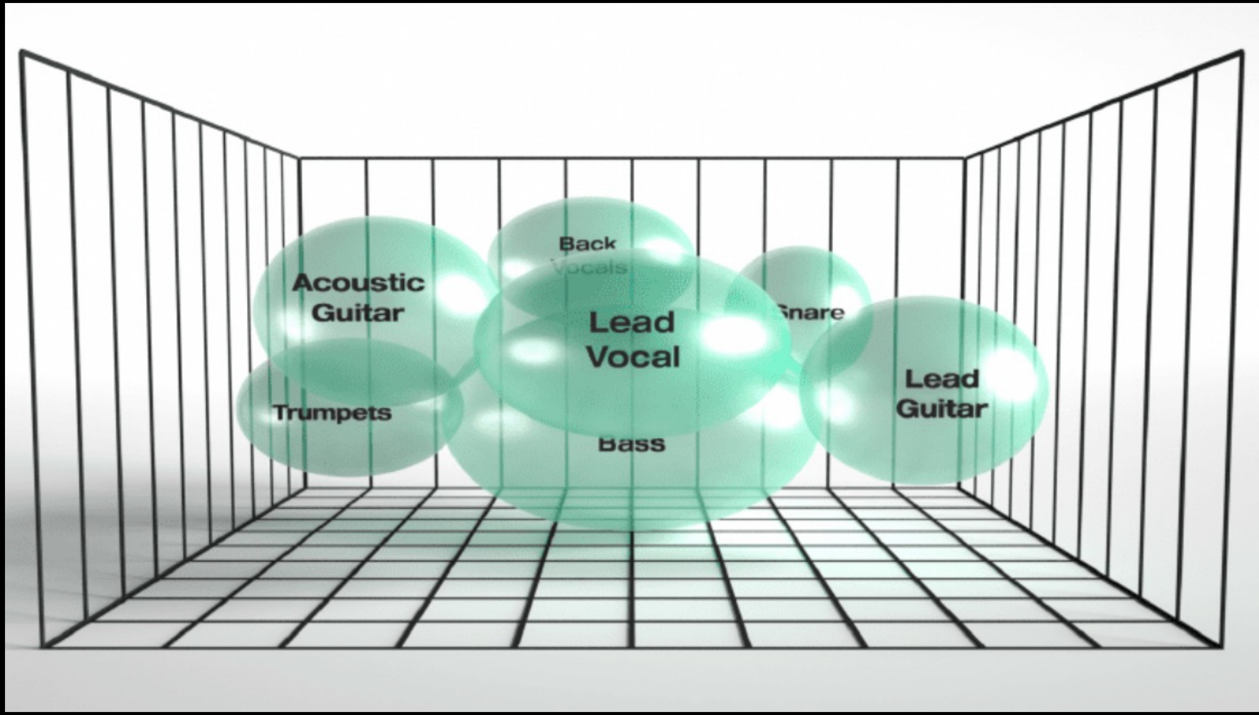
- Spectral mixing = retain/remove specific frequency regions to focus on instruments
- Depends on source types (=importance) and acutal signal (=presence)



20.0 Hz	75.0 Hz	100 Hz	392 Hz	1040 Hz	2500 Hz	7500 Hz	20000 Hz	Gain
12 dB/Oct	0.0 dB	0.0 dB	-4.5 dB	0.0 dB	0.0 dB	0.0 dB	24 dB/Oct	0.0 dB
0.71	1.00	0.60	0.30	0.41	0.20	1.00	0.71	

Example 3: Reverb

- Depth impression in mixing:
  - Proportion of direct sound, ER and tail crucial
  - Latency of direct sound (=time alignment)
- Different but interdependent settings per channel



How do we get this information?

What is Inter-Plugin Communication?

Definition

- Inter-Plugin Communication enables a single plugin instance to
  - Discover other plugin instances
  - Open communication channels
    - Send/receive information about channel strip (audio content, parameter values, etc…)
  - Create logical groups

Main Components

1. Service Discovery

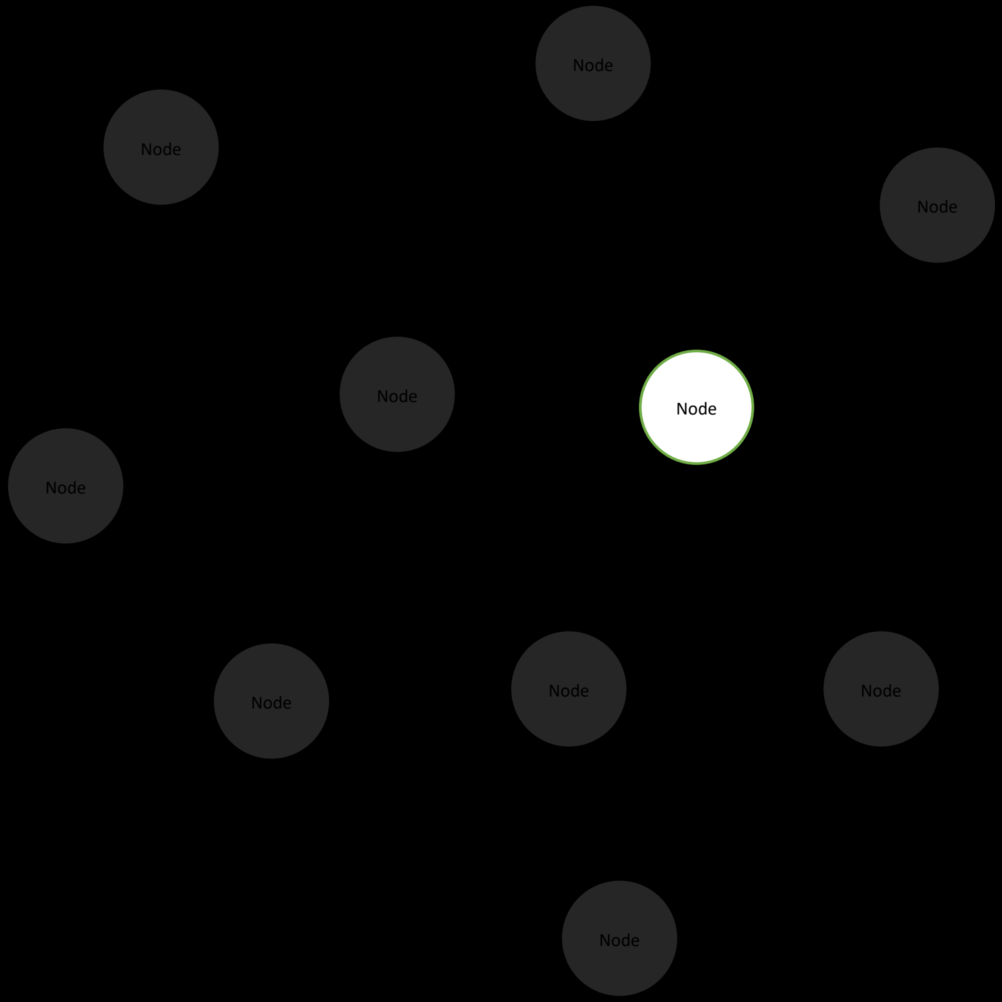
- See and be seen: advertise willingness to mingle
- One-To-Many => Broadcast

2. Communication Channels

- One-To-One, Bi-Directional
- Agnostic of transmitted data
- (Auth, Encryption)

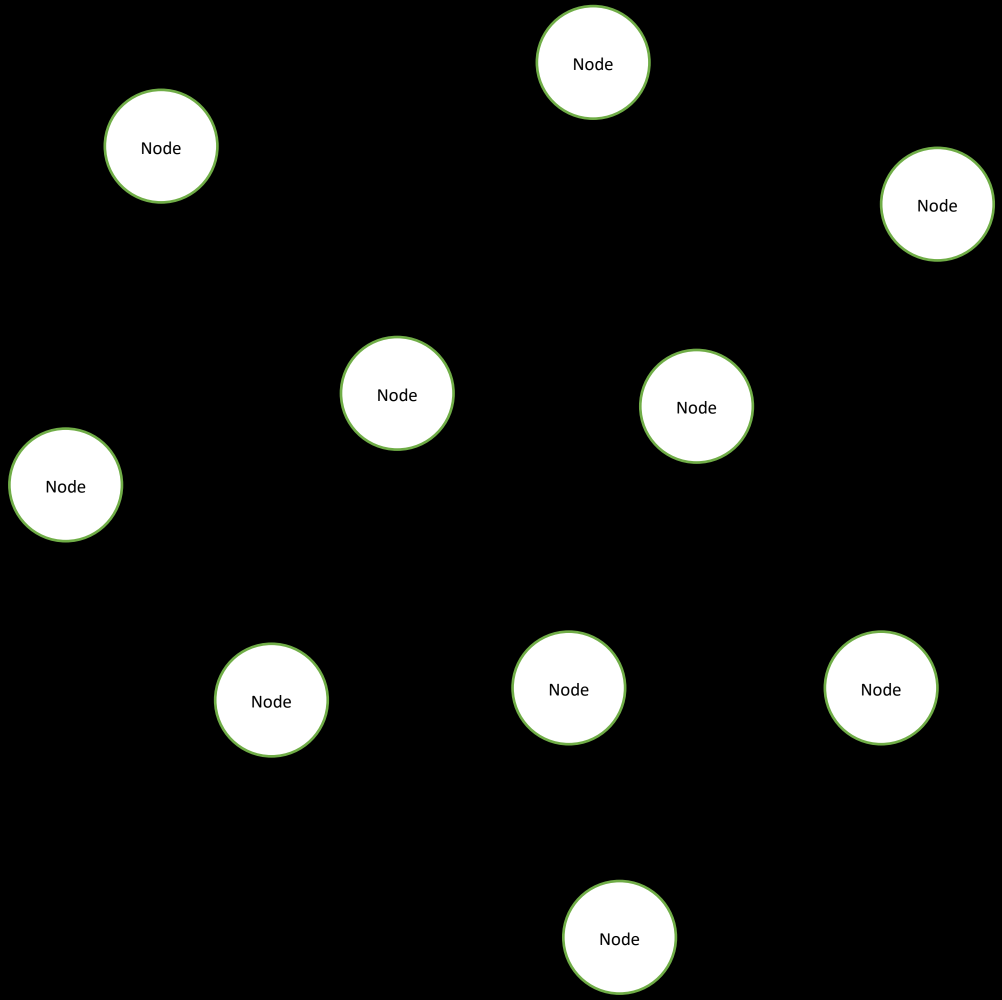
Service Discovery

- Node or Entity
  - A single instance of a plugin
  - Possibly even separate Processor and View
  - Advertises its
    - existence and identity
    - supported comm channels (=endpoints, protocols)
    - Group affiliations



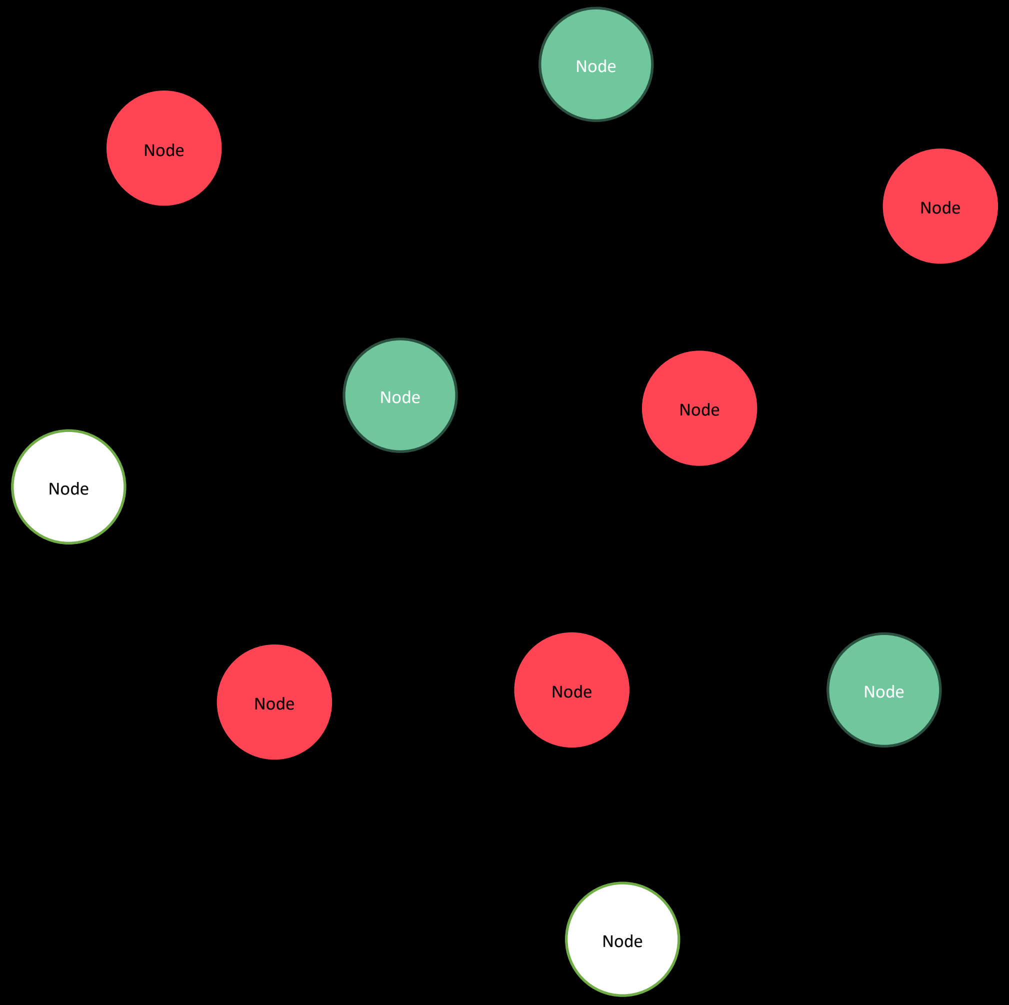
Service Discovery

- Swarm
  - Entierty of all discoverable Nodes
  - Each Node keeps register of other Nodes



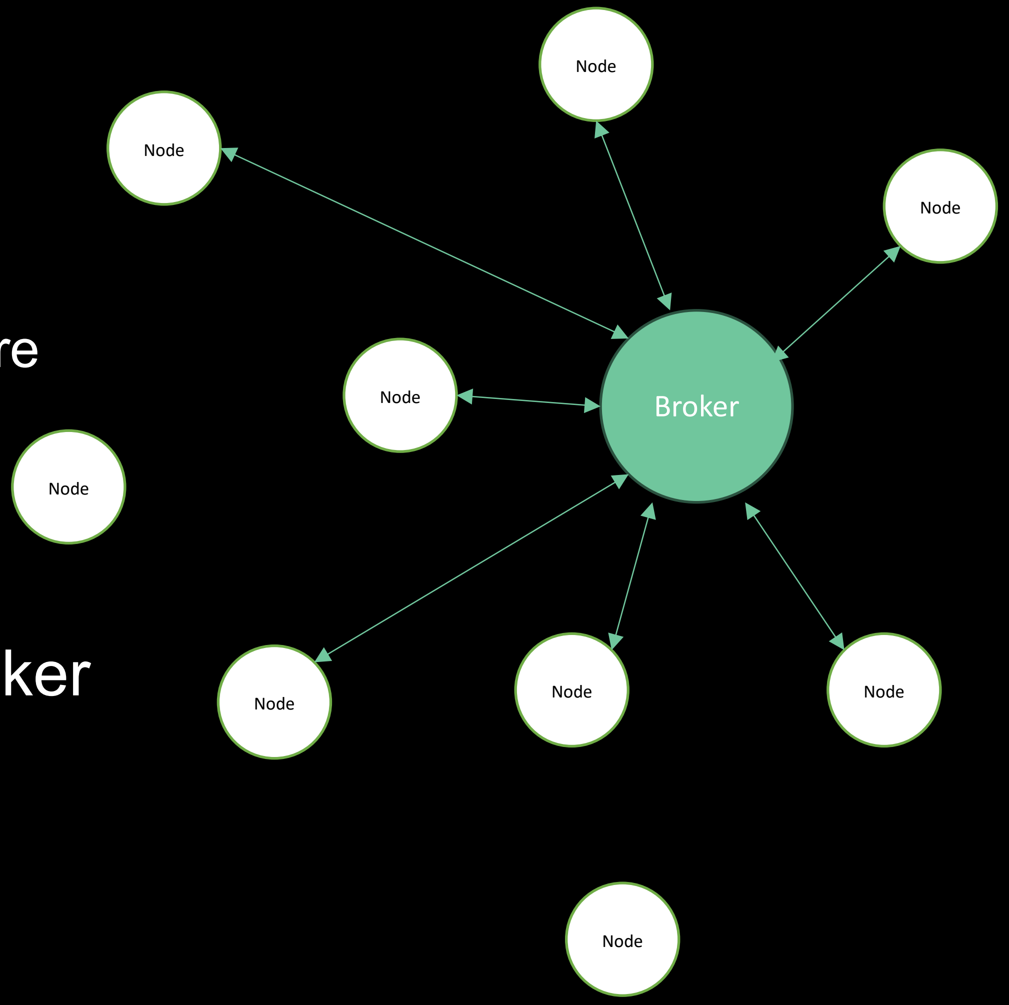
Service Discovery

- Group
  - A subset of Entities within the Swarm sharing one or more properties
- Group Leader
  - An instance within the Group elected by its members
  - Some special duties/capabilities



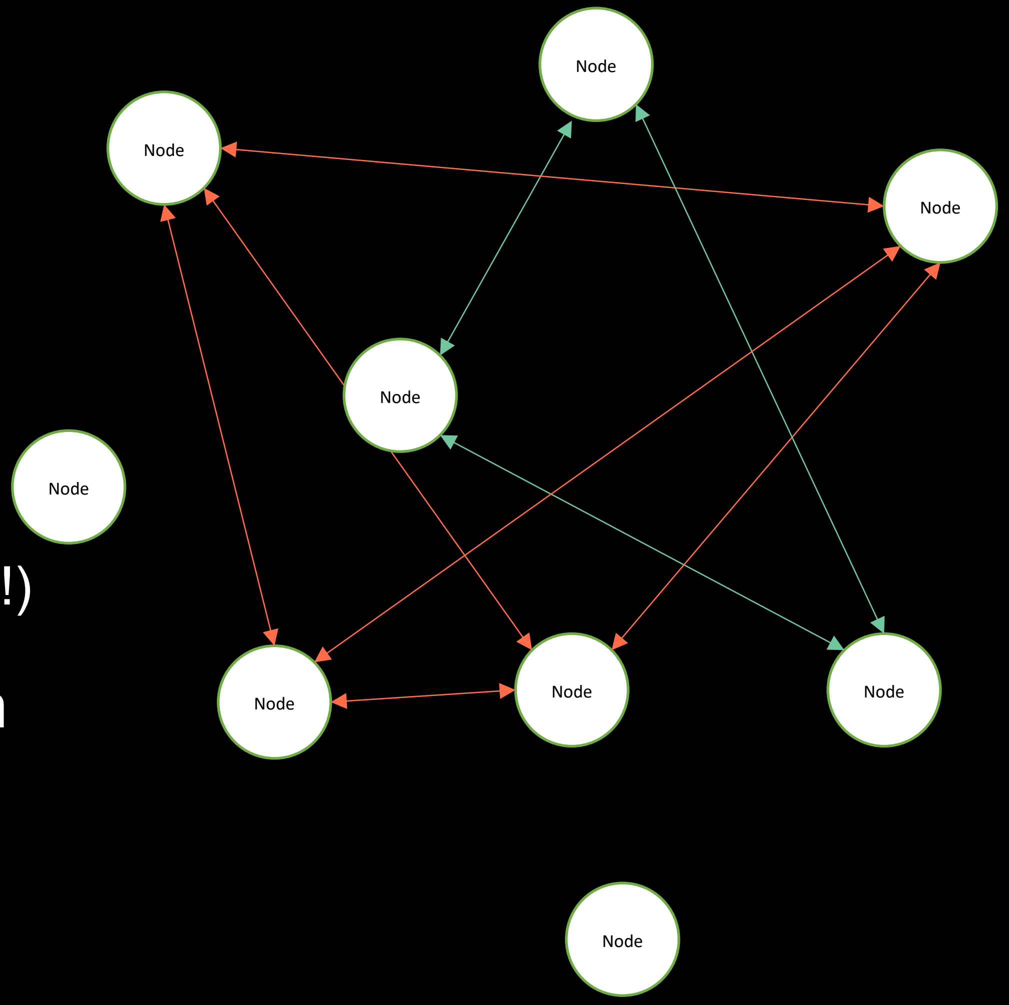
Managed vs. Distributed System

- Managed/Brokered System
  - Capabilities centralized:
    - One node knows more than the others/has more responsibilities
    - Single truth: the broker is always right
  - Communication can be routed through broker (overhead) or Peer-To-Peer
  - Single point of failure -> Lifetime management



Managed vs. Distributed System

- Distributed System
  - Everyone can do everything
  - No absolute truth
    - All nodes must agree on truth
    - Decision making mechanism required (quorum!)
  - If a node disappears another one can fill in



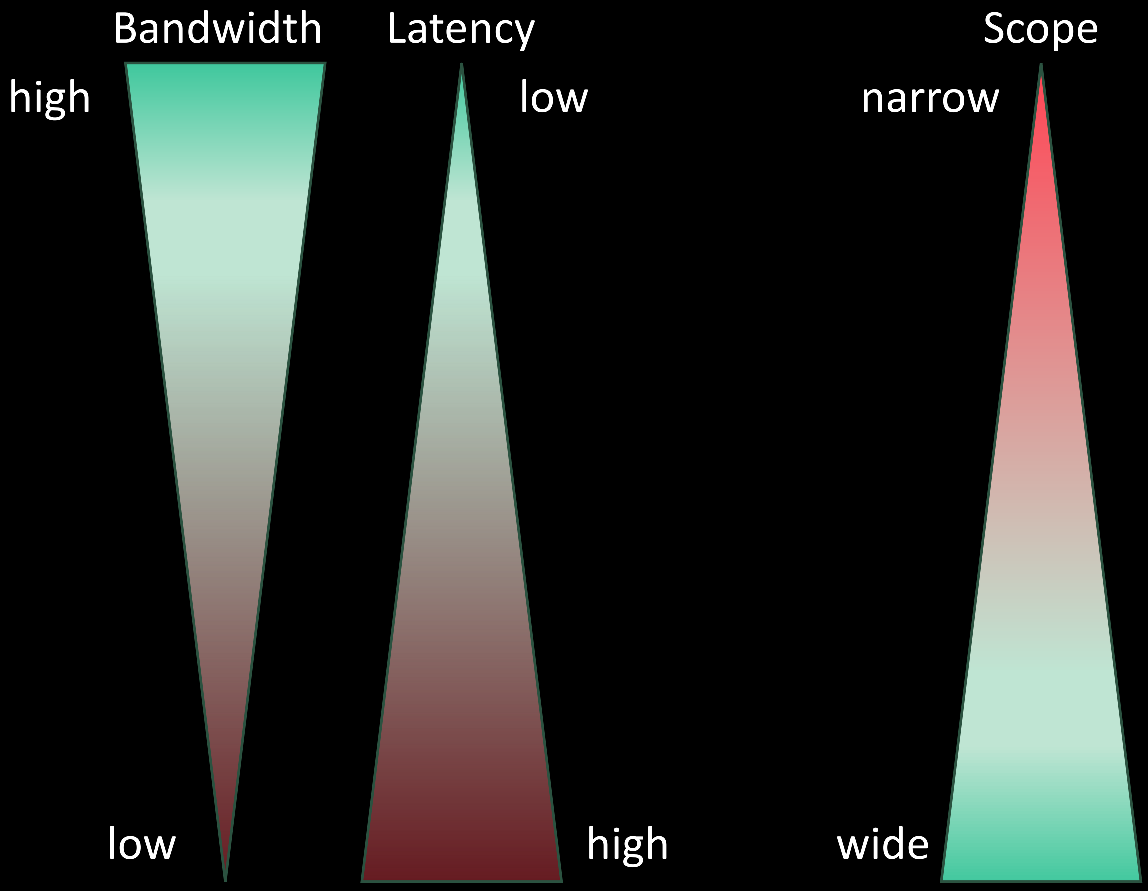
How do Inter-Plugin Communication?

Scope

- Beyond instance boundaries: ✅
  - e.g. EQ on Kick & Bass
- Beyond process boundaries: ✅
  - e.g. sandboxed plugin processes
- Beyond machine boundaries: 🤨
  - Might make sense?

Communication Stack

- Static Memory
  - E.g. Singleton
- Local Sockets
  - Unix Domain Sockets
  - Windows Named Pipes
- IP



Scope Of Inter-Plugin Communication

- Beyond instance boundaries: ✅ → Static Memory
    - e.g. EQ on Kick & Bass
- Beyond process boundaries: ✅ → Sockets
    - e.g. sandboxed plugin processes
- Beyond machine boundaries: 🤨 → IP
    - Might make sense?

Building A Service Discovery

- Maximum scope desired => IP
  - UDP beacon
  - Broadcast on local network or multicast on localhost
- Protocol requirements
  - Identification (UUID, IP, name, type, (host) process, endpoints, groups)
  - Defined set of messages (Hello, Bye, Ping, Elect, Leader,…)
  - Arbitrary property sets according to application needs

Building A Service Discovery

- Filter
  - Selectively include/exclude Nodes by type, process, etc
- Grouping/Leadership
  - Anyone can create or join a group
  - Only a group leader can define Group properties (e.g. name, colour, etc)
  - Election algorithm e.g. Bully Algorithm
    - Stable majority required -> beware of split-brain situations!

Building A Comm Channel

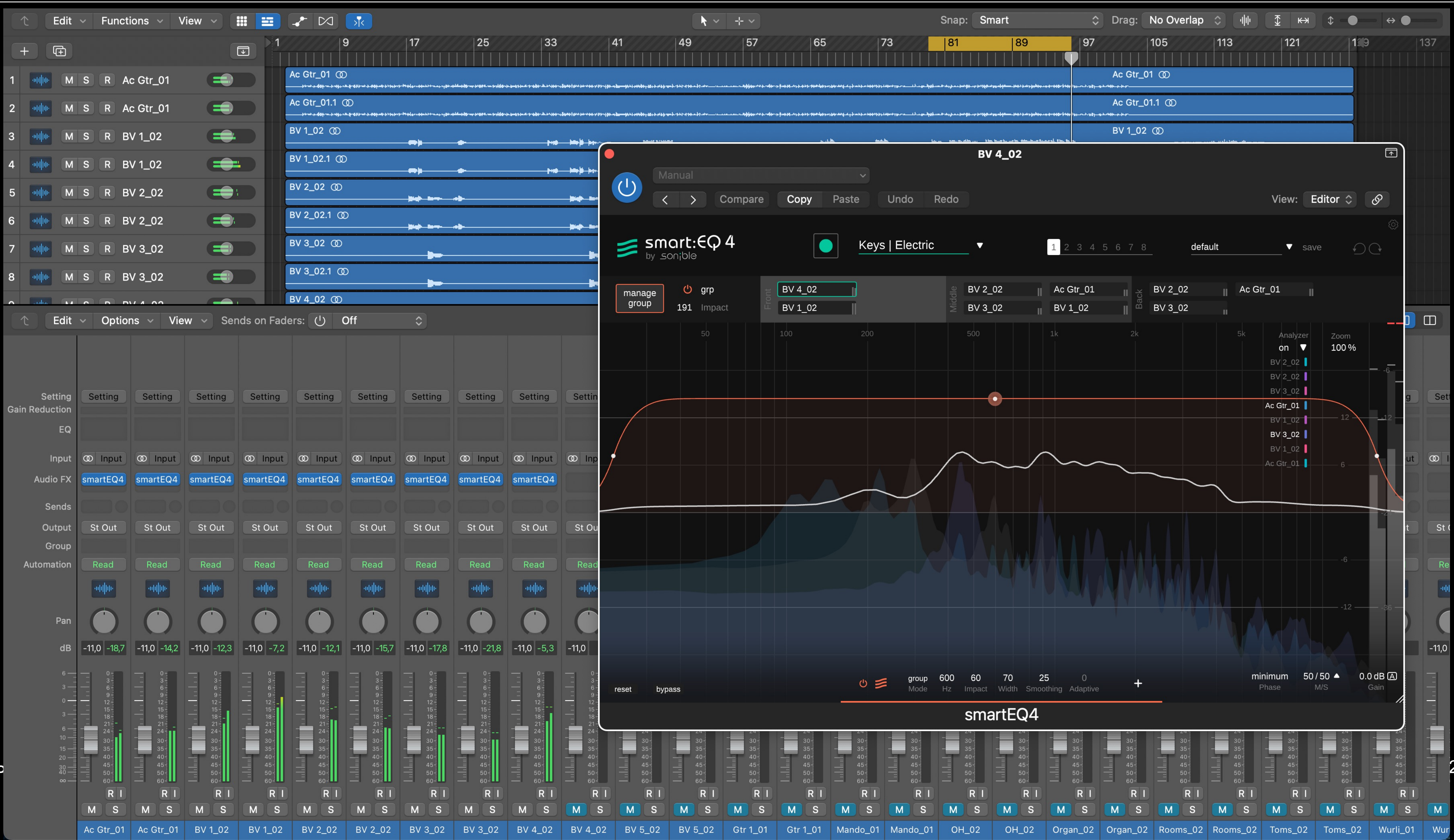
- Scope more limited & bandwidth/latency more important
  - static memory or
  - local sockets
    - zeroMQ (https://zeromq.org)
    - Nanomsg/nng + nngpp (https://github.com/nanomsg/nng)
- Transmission scheme
  - HTTP style (POST to or GET from peers)
  - RPC

Building A Comm Channel

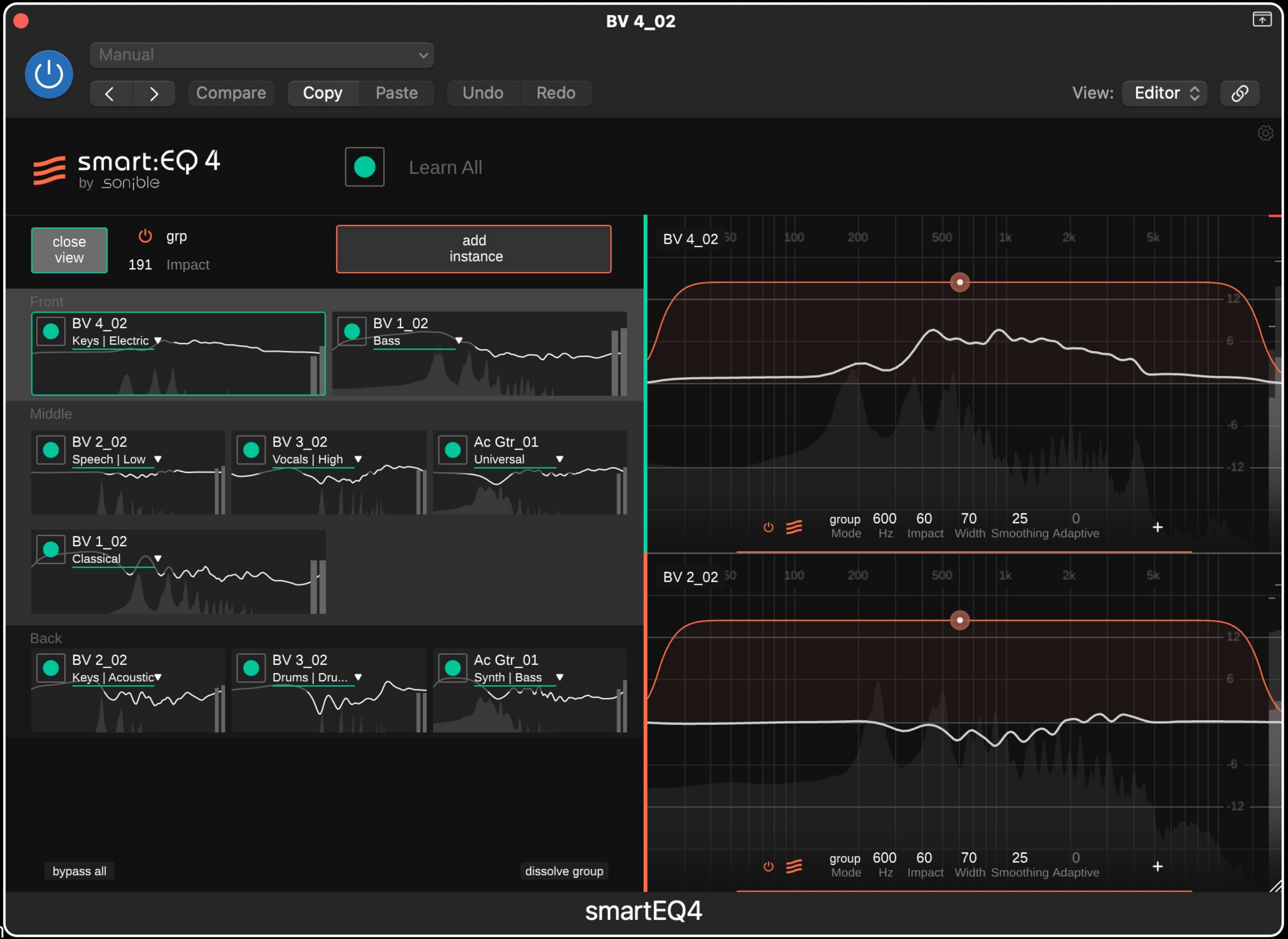
- Transmission payload
  - Static memory: direct calls into instances -> serialization can be avoided
  - Otherwise: binary blobs -> efficient serialization!
    - protobuf (https://protobuf.dev) or capnproto (https://capnproto.org)

More detail in Janos Buttgereit's talk on Wed, 15:00 in Empire

Tada! 🎉


Edit
Functions
View
Snap: Smart
Drag: No Overlap
Ac Gtr_01
Ac Gtr_01.1
BV 1_02
BV 1_02.1
BV 2_02
BV 2_02.1
BV 3_02
BV 3_02.1
BV 4_02
Options
Sends on Faders: Off
Setting
Gain Reduction
EQ
Input
Audio FX
smartEQ4
Sends
Output
St Out
Group
Automation
Read
Pan
dB
Manual
Compare
Copy
Paste
Undo
Redo
View: Editor
smart:EQ 4 by sonible
Keys | Electric
default
save
manage group
grp
191 Impact
Front
Middle
Back
Analyzer on
Zoom 100 %
reset
bypass
group Mode
600 Hz
60 Impact
70 Width
25 Smoothing
0 Adaptive
minimum Phase
50 / 50 M/S
0.0 dB Gain
smartEQ4

Tada! 🎉


BV 4_02

Manual

Compare Copy Paste Undo Redo

View: Editor

smart:EQ 4 by sonible

Learn All

close view

grp

191 Impact

add instance

Front

BV 4_02 Keys | Electric

BV 1_02 Bass

Middle

BV 2_02 Speech | Low

BV 3_02 Vocals | High

Ac Gtr_01 Universal

BV 1_02 Classical

Back

BV 2_02 Keys | Acoustic

BV 3_02 Drums | Dru...

Ac Gtr_01 Synth | Bass

bypass all

dissolve group

BV 4_02

group Mode, 600 Hz, 60 Impact, 70 Width, 25 Smoothing, 0 Adaptive

BV 2_02

group Mode, 600 Hz, 60 Impact, 70 Width, 25 Smoothing, 0 Adaptive

smartEQ4

Limitations

Limitations

- No Audio - Only control data makes sense
  - Synchronicity: DAW channels are not processed in consistent order
  - Latency compensation approach unknown plugin instace
  - Transmission delay has difference effect in realtime and offline rendered situation
  - Sample accuracy not possible. Cannot replace a routed sidechain input

Limitations

- Unknown processing graph:
  - No knowledge about other effects, rotuing, panning, level in mix
  - Cannot assume exact Node output to end up in sum bus
    - E.g. level matching across channel gets broken by user changing channel fader position

Dear DAW Developers,

Life would be so much easier if we could:

- Standardize retrieval of processing graph detail
- Standardize comm channels between plugin instances

Yours truthfully,

Peter

sonible

Thank you!

Let’s discuss

peter.sciri@sonible.com