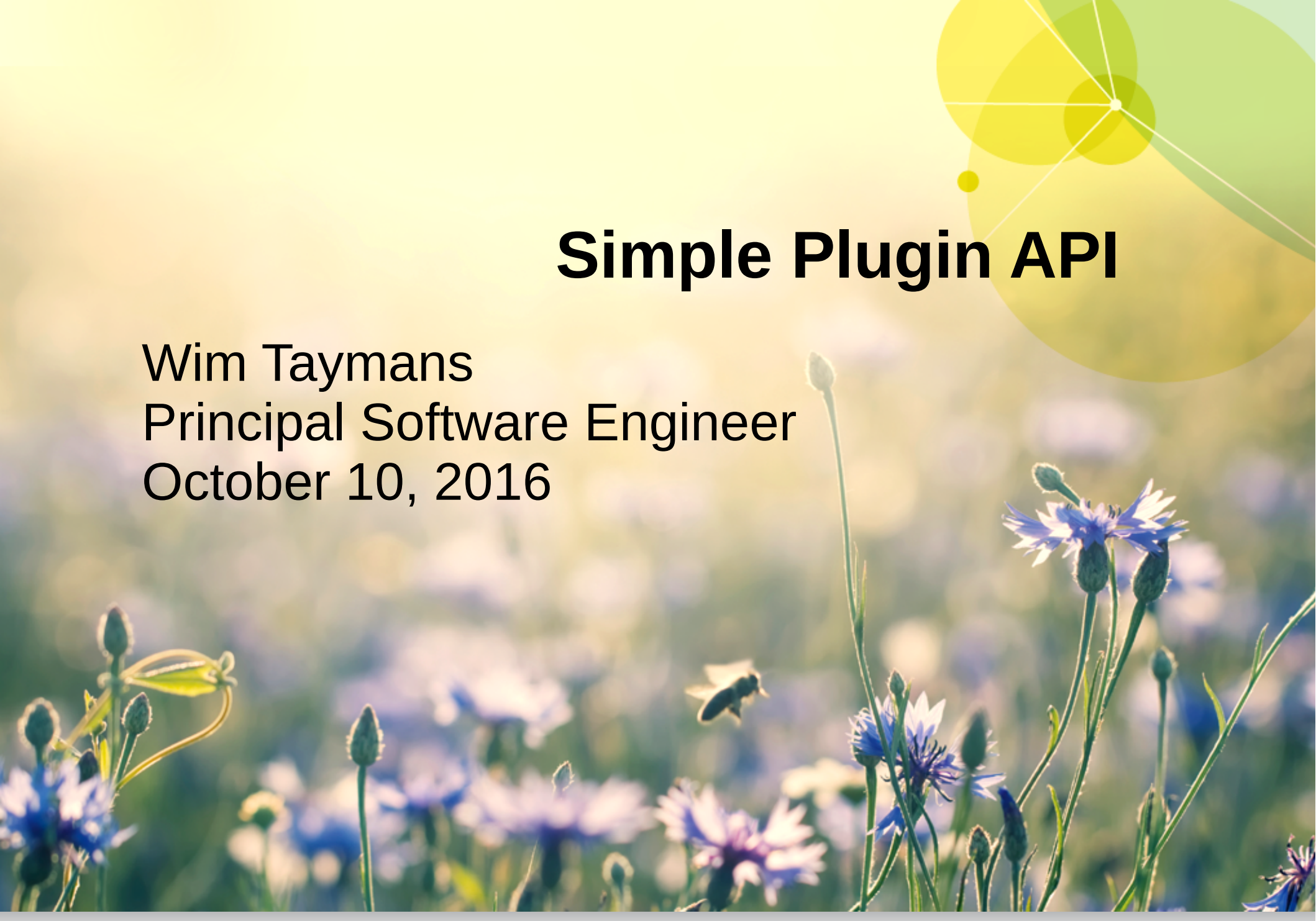


# Simple Plugin API

Wim Taymans
Principal Software Engineer
October 10, 2016

# In the begining

- Pinos
  - DBus service for sharing camera
  - Upload video and share

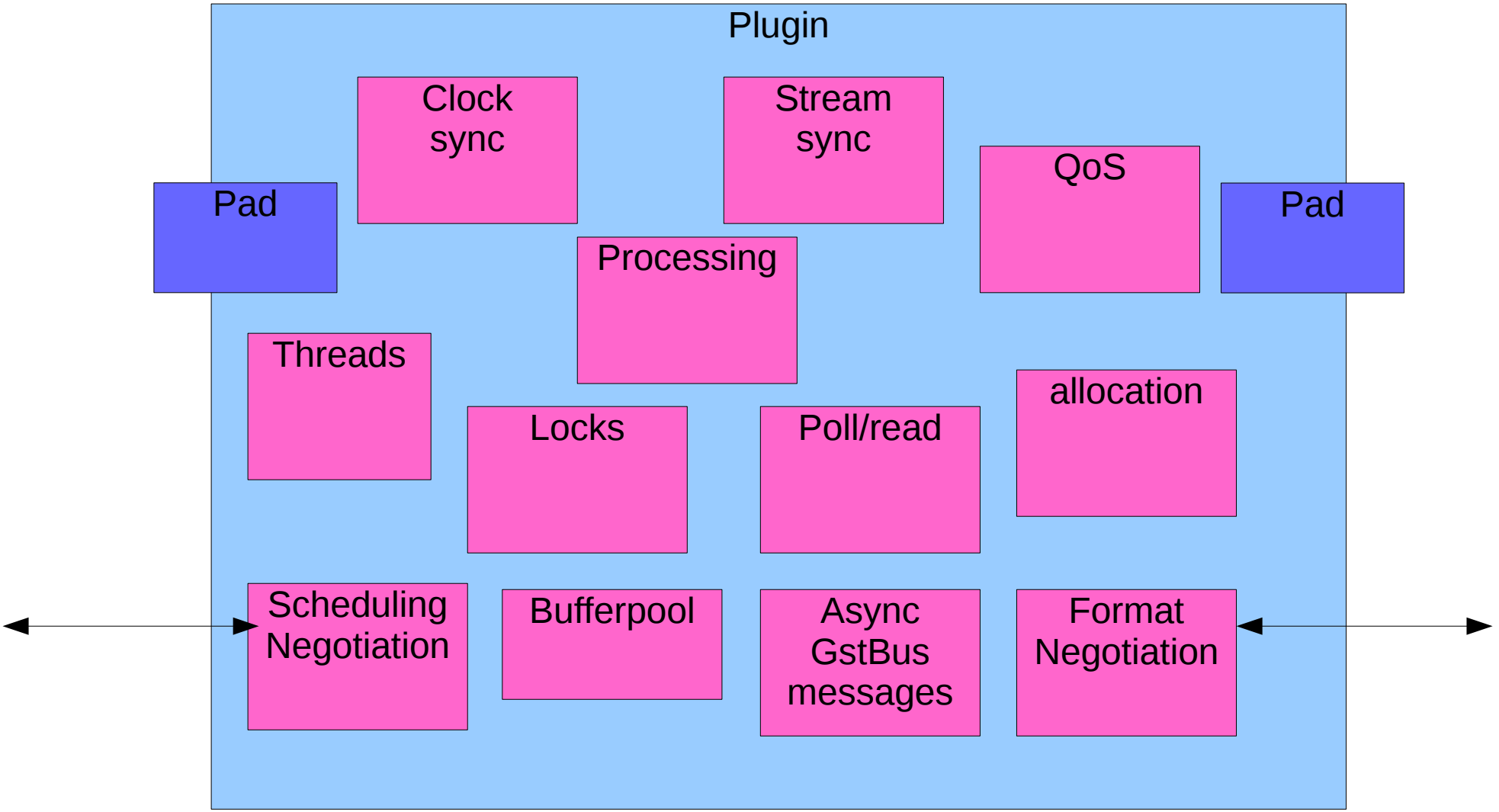
# And then...

- Extend scope
  - Add audio too upload, playback, capture
  - Need for processing pipeline…
  - Jack-like graphs?
  - Need for real-time processing with extemely low latency 0.3ms <32 bytes buffers

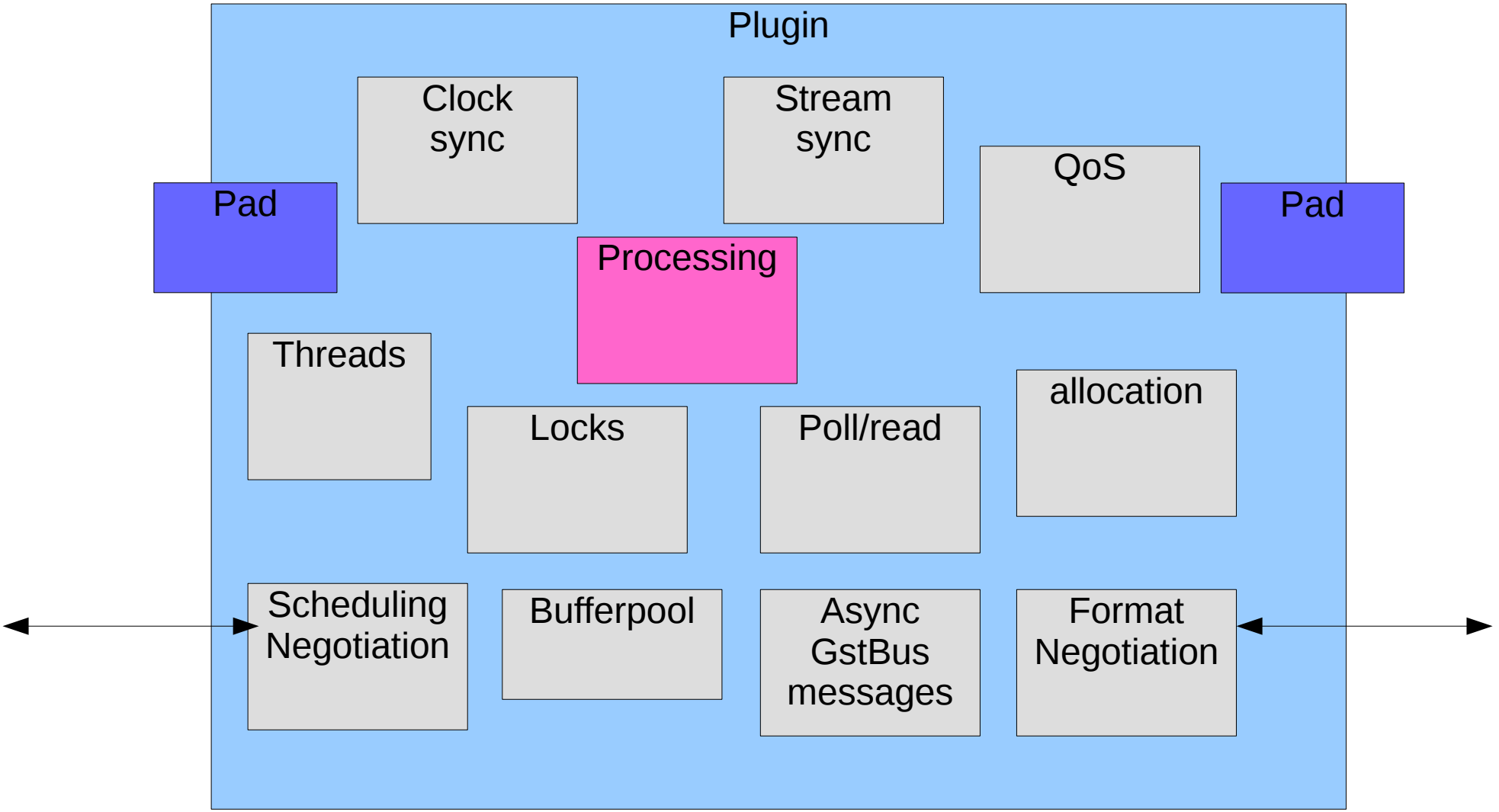
# GStreamer

- Creates a lot of threads
  - Sources, sinks, demuxers
  - No way to combine threads
- Does all kinds of locking and allocation in processing threads
  - Buffers, events, caps...
- Lots of allocations while negotiating
  - Difficult to predict real-time behaviour
  - We can do better

# A GStreamer plugin does a lot



# Can we focus on this



# The goals

- Unified plugin API
  - For muxer, demuxer, decoder, encoder, effect, mixer,…
- Software + Hardware implementations
- Synchronous and asynchronous
- Hard real-time capable
- Extensible
- Minimal
  - Does not bring in a complete framework
  - Can be used in different frameworks

# The options.. (and its plugins)

| | Unified | SW | HW | RT | ext | generic | minimal | Async |
|---|---|---|---|---|---|---|---|---|
| v4l2 | | | | | | | | |
| alsa | | | | | | | | |
| LADSPA | | | | | | | | |
| LV2 | | | | | | | | |
| MediaCodec | | | | | | | | |
| OpenMAX | | | | | | | | |
| FFMpeg | | | | | | | | |
| MFT | | | | | | | | |
| upipe | | | | | | | | |
| GStreamer | | | | | | | | |

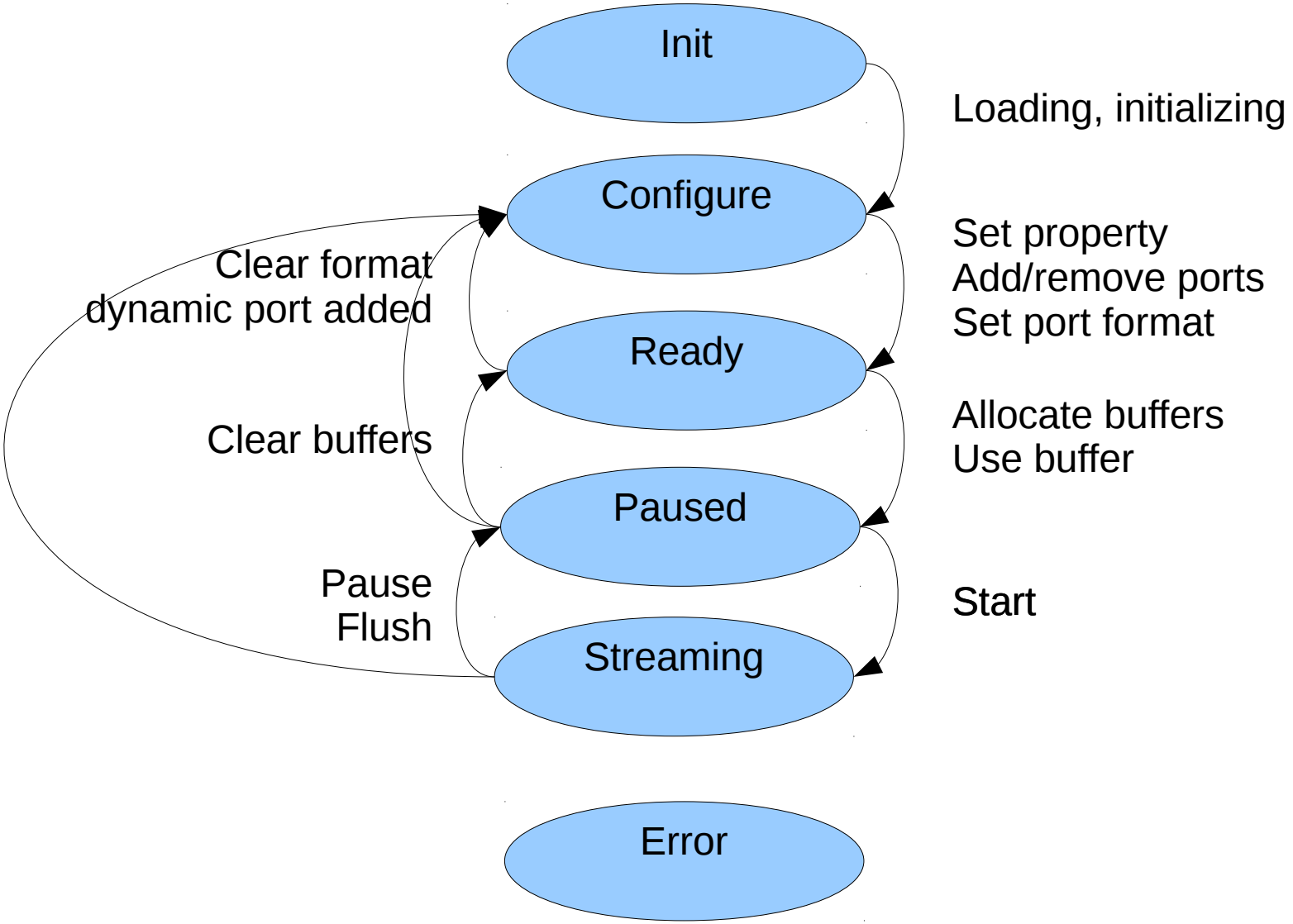
# The ideas

- Interfaces
  - Structure with methods
  - Introspection of interfaces
- URI map
  - Map a string to an id
- API is .h files
  - Methods either inline or in separate helper library

# The Node

- A basic processing module
- Dynamic input/output ports
  - Ports are ids
- Does not do allocations
  - App must do allocations of buffers
  - Allocation free format description
- Give input, produces output
- Goes through state changes

# The Node states



# The Properties

- Key/type/value
- Unset values + description of possible values
  - Lists
  - Ranges (with steps)
  - Enum/flags
- For nodes and ports

# The Formats

- Media type
- Media subtype
- Properties
- For ports
  - Enumerate formats (with filter)
  - Set format (clear by setting NULL format)

# The Port info

- Setting a format changes the info on a port
- Port features (live, can allocate,…)
- latency
- Allocation parameters
  - Size, alignment, metadata, padding

# The Allocation

- Application allocates buffers + metadata
  - Based on port info
- Allocates the buffer memory or..
- Have one of the ports allocate memory if possible
  - With alloc_buffers
  - Only if something else than malloc, really
- Does use_buffers on the ports

# Streaming

- Asynchronous or synchronous processing
- Async
  - You get events when data can be pushed and pulled from ports
- Sync
  - You push and pull from pads
  - Return code tells you what to do
    - Push more, pull, go back to configure/ready state

# Streaming

- Push buffer to input port
  - You actually only send the id of the buffer
  - Both ports know buffer from alloc/use_buffer
- Pull buffer from output port
  - Pull many ports in one go
  - Get the id of the buffer
- Event when buffer id is no longer used

# Port status

- Tells you
  - If you can push/pull
  - If the port has a format
  - If the port has buffers

# Points of interest

- Only callback is for events
- Some methods can be async
  - High bit set in return value, low bits are seqnum
  - You get event with seqnum when the command completes

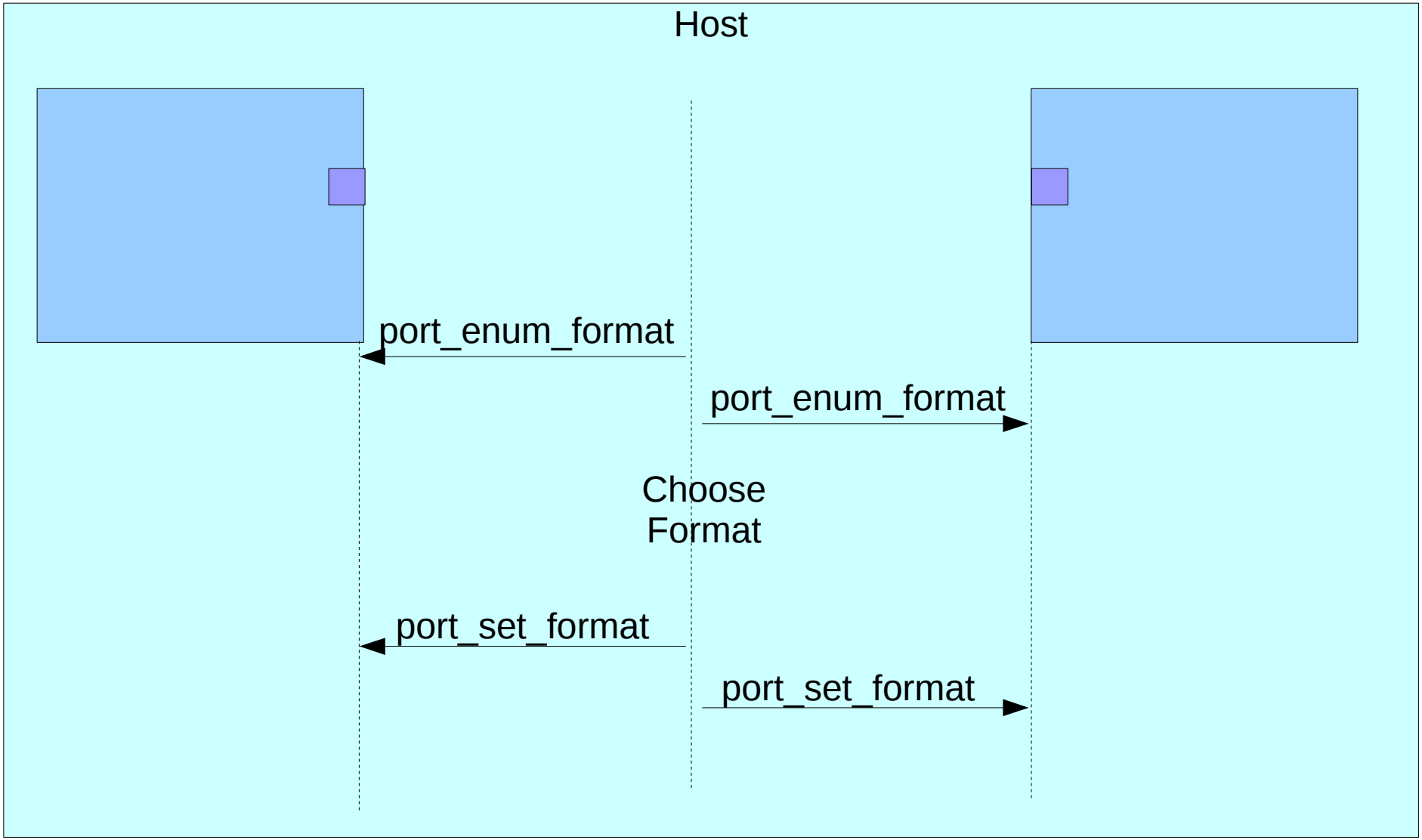
# Platform support

- A list of interfaces is given at initialization
- Logging
- Mainloop integration
  - Mainloop
  - Dataloop (for realtime processing)
- Scheduler
  - For doing things in other threads

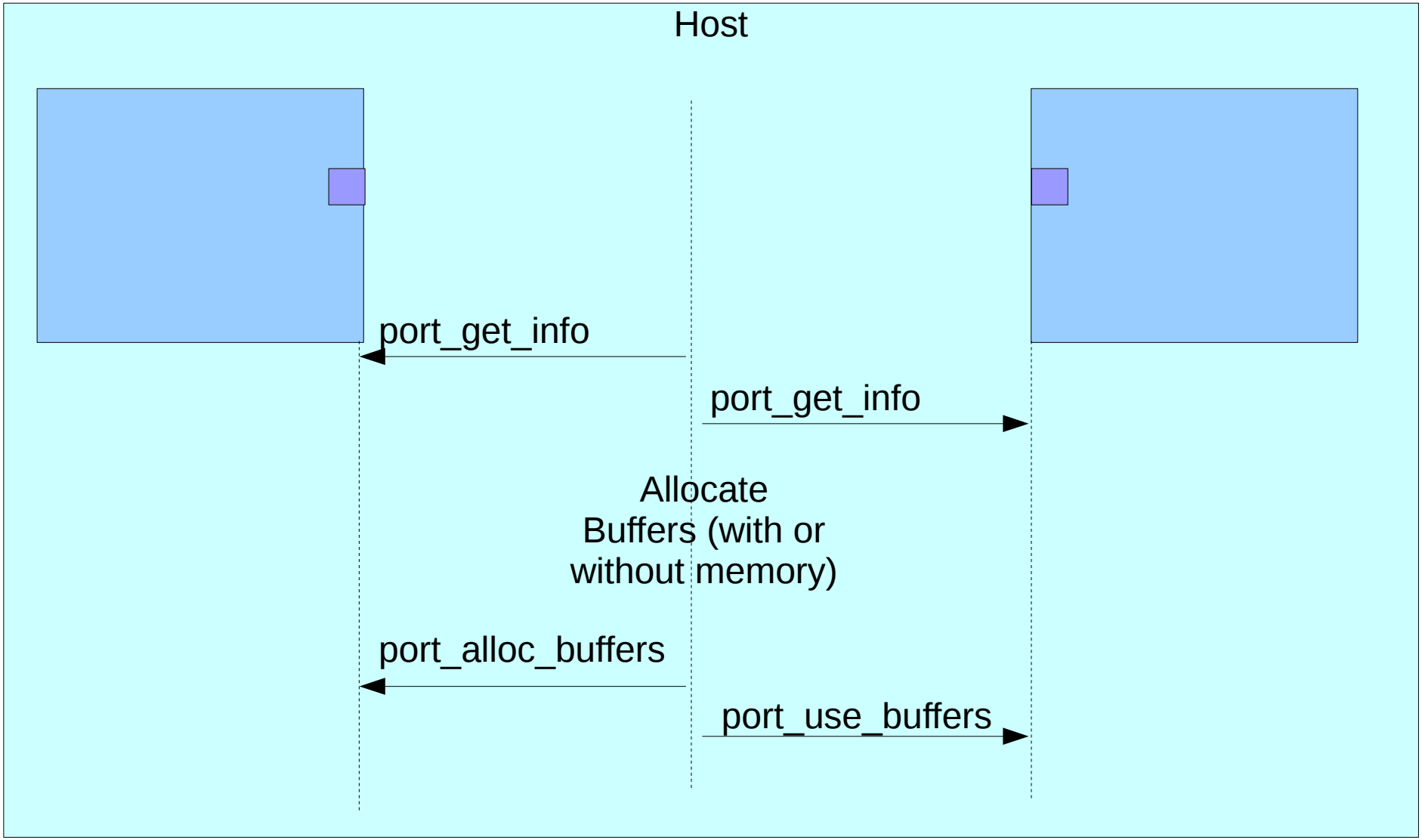
# How to use

- Host has a lot of flexibility and needs to be smart
  - Can choose negotiation
  - Can choose allocation
  - Can choose scheduling, threads, mainloops
  - Can choose synchronization
- A GStreamer plugin can be a host
- We could write higher level components working directly with the nodes

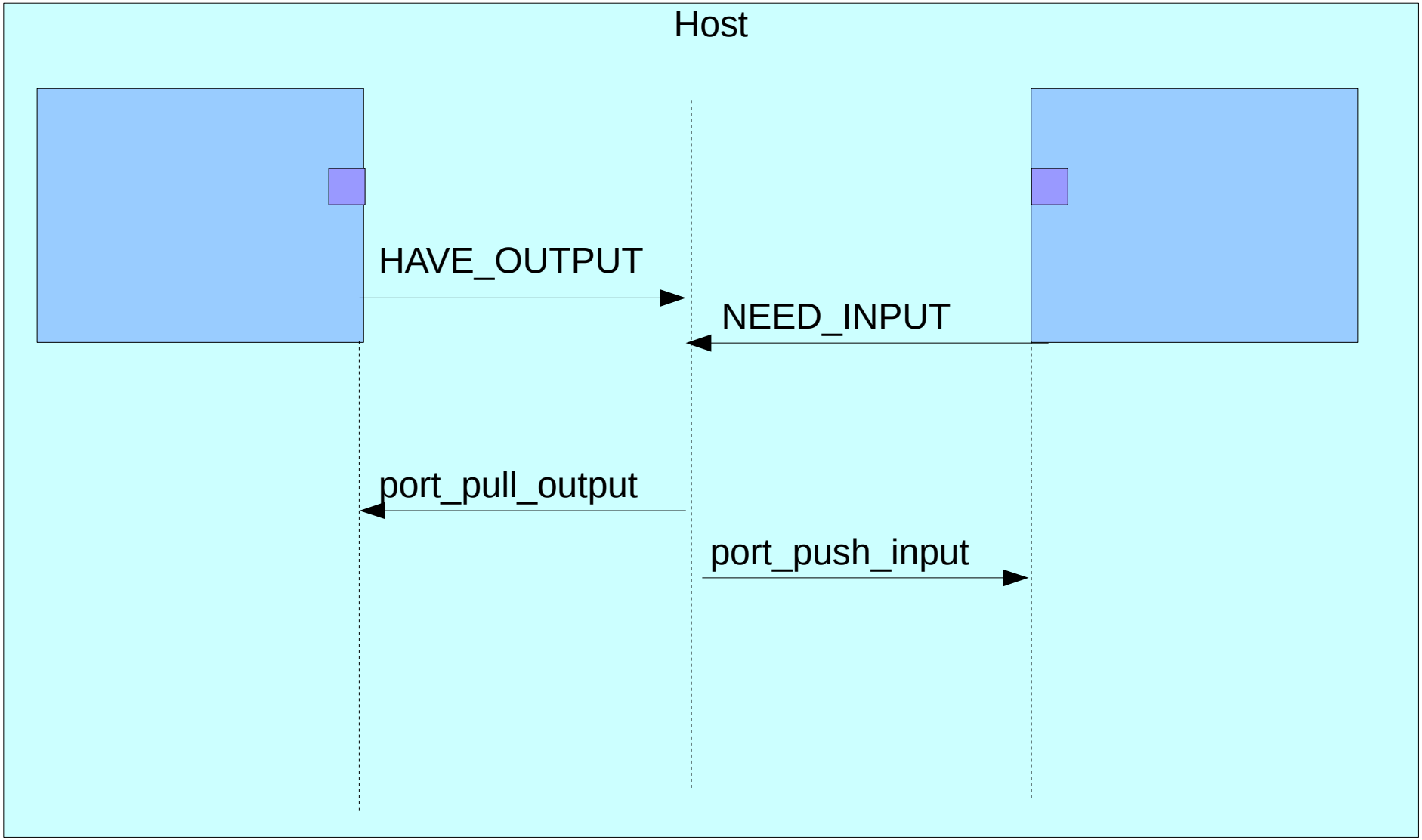
# Example.. negotiation



# Example.. negotiation



# Example.. execution



# Status

- V4l2 monitor and source
- Alsa source/sink and monitor
- Audiotestsrc/videotestsrc
- Logging/mapping
- Clock
- Negotiation, allocator in Pinos
- Scheduler in Pinos for capture->send

# Future plans

- Still early prototypes
- Plan to move some code from Gstreamer in SPA plugins
  - Audio/video conversion
  - Audiotestsrc/videotestsrc
- Work on generic scheduler for plugins
- Hope to use Gstreamer has host for plugins


http://cgit.freedesktop.org/~wtay/pinos/log/?h=work