# Visual prototyping of audio applications

David García (UPF)

Pau Arumí (UPF)

Xavier Amatriain (UCSB)

Linux Audio Conference 2007

Berlin, 24th March

## Index

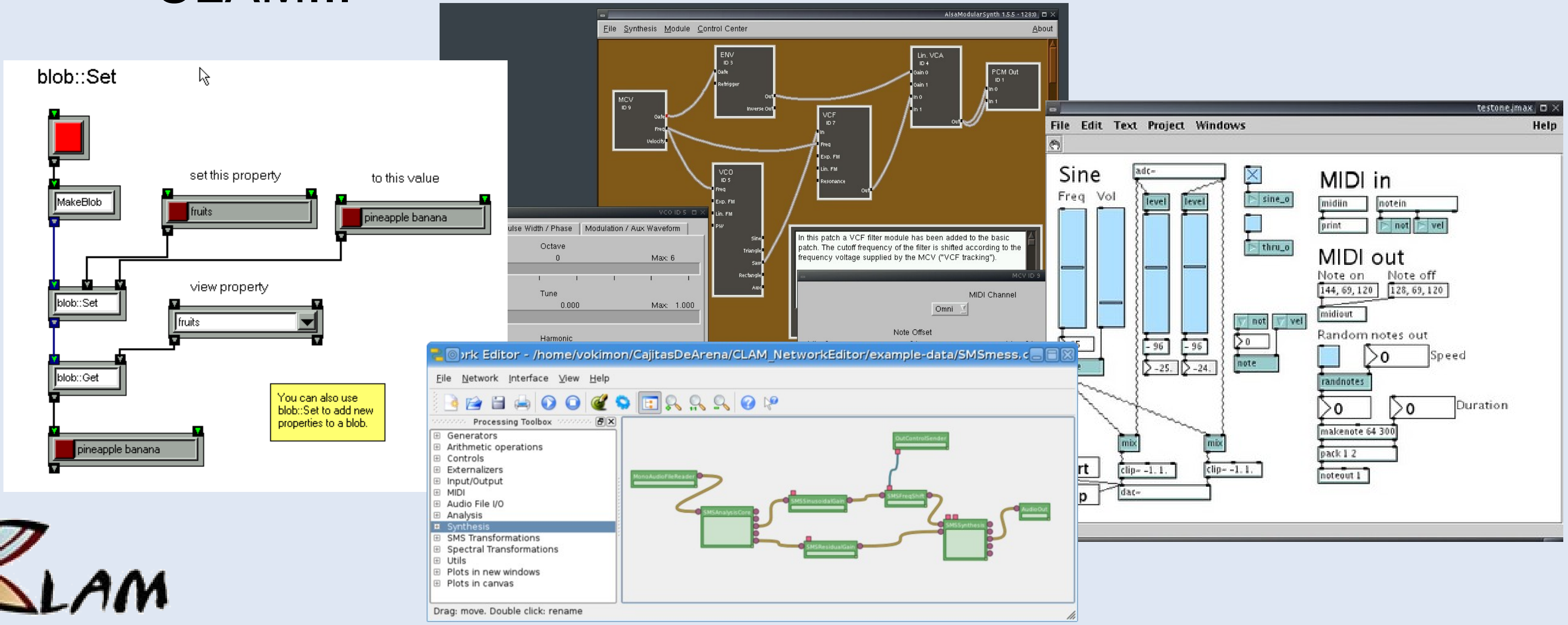
## Index

## Introduction

- How can we improve the audio software development process?
- Framework offers system models for a specific domain
- Mature frameworks offer visual builders

## Introduction

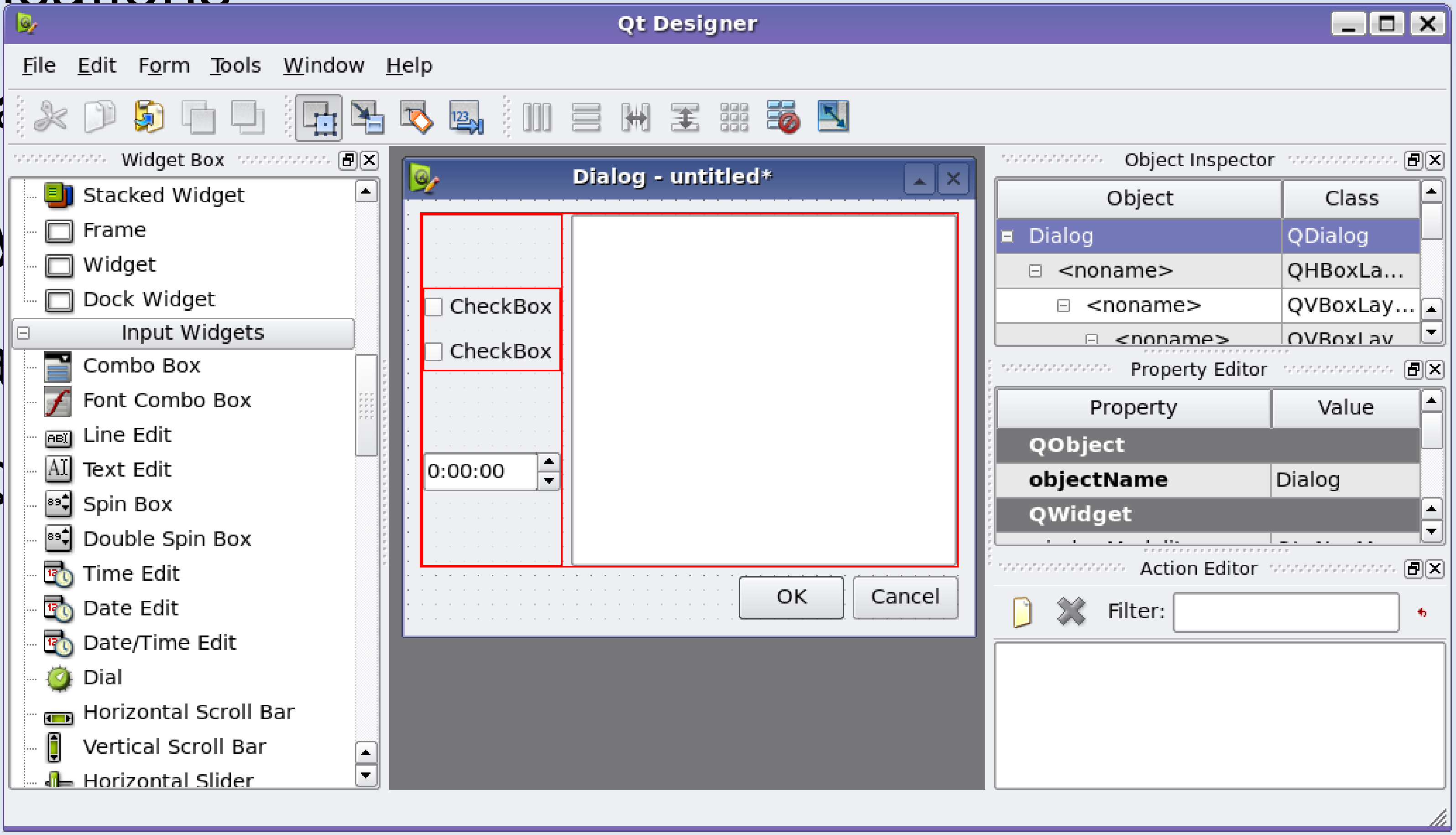
- What about audio and music domain?
- Long history of frameworks and environments
  - PD, MAX, SuperCollider, MARSYAS, Open Sound World, CLAM...


You can also use blob::Set to add new properties to a blob.

In this patch a VCF filter module has been added to the basic patch. The cutoff frequency of the filter is shifted according to the frequency voltage supplied by the MCV ("VCF tracking").

# Introduction

- Data-flow builders are not enough to build full applications
  - Pa
- They
- Wha
  - Q



| Object | Class |
|---|---|
| Dialog | QDialog |
| <noname> | QHBoxLa... |
| <noname> | QVBoxLay... |
| <noname> | QVBoxLay |

| Property | Value |
|---|---|
| QObject | |
| objectName | Dialog |
| QWidget | |

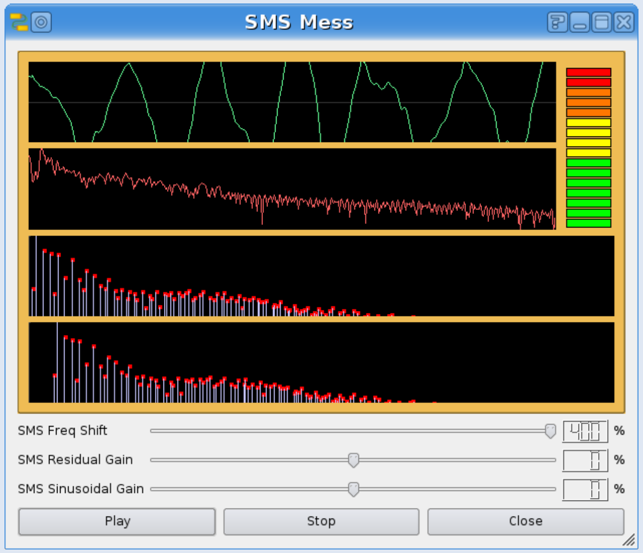
## Introduction

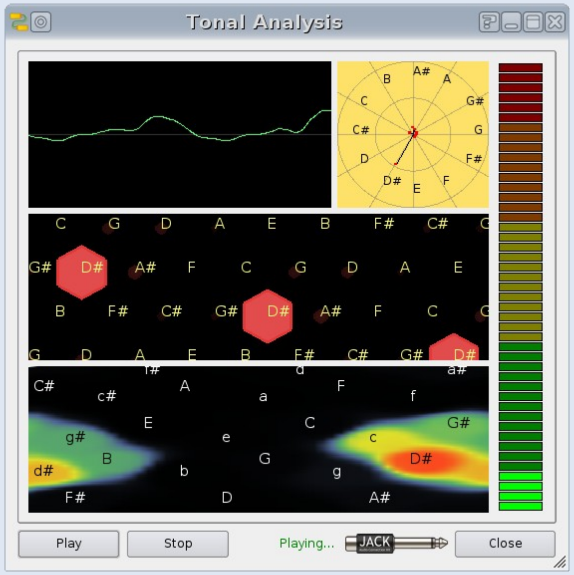
- The problem:
  - To integrate both worlds we still need low-level programming.
- The proposal:
  - Define an architecture that could enable the visual development of full audio applications including the processing core, the interface and the application logic.

## Index

## Target applications

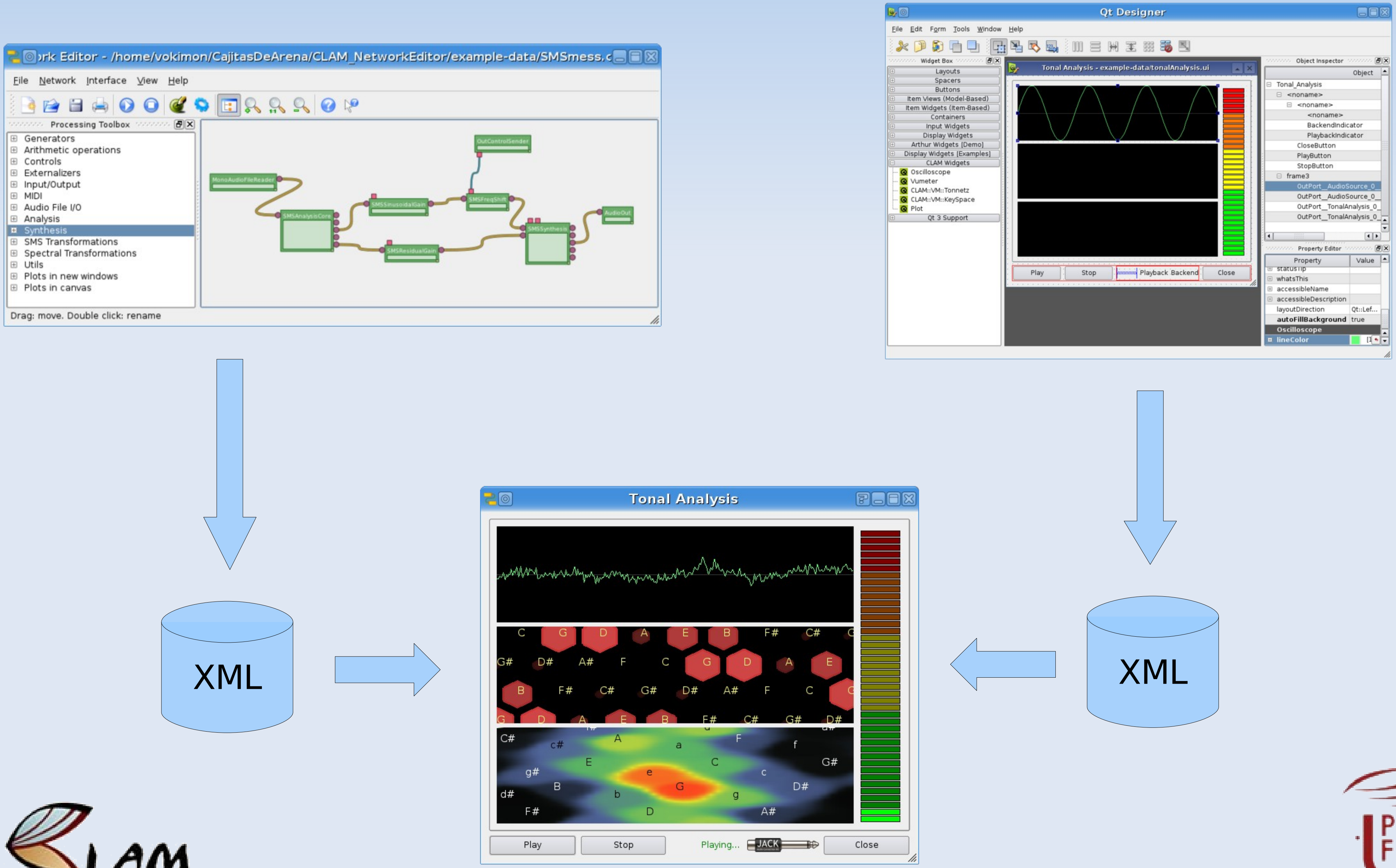
- First goal: Simple application logic
- Real-time synthesizers, analysers and effects
- Still extensible by coding





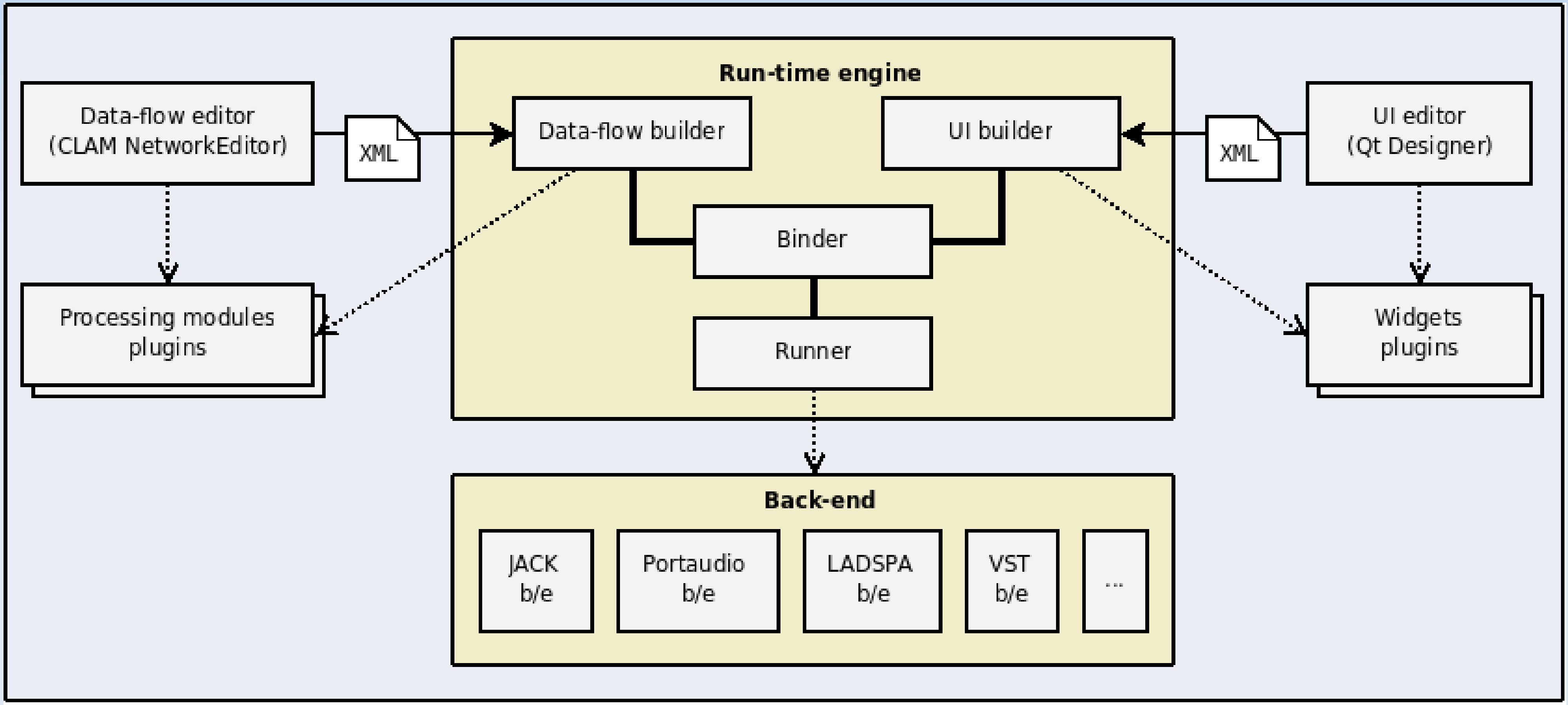
## Index

# Demo

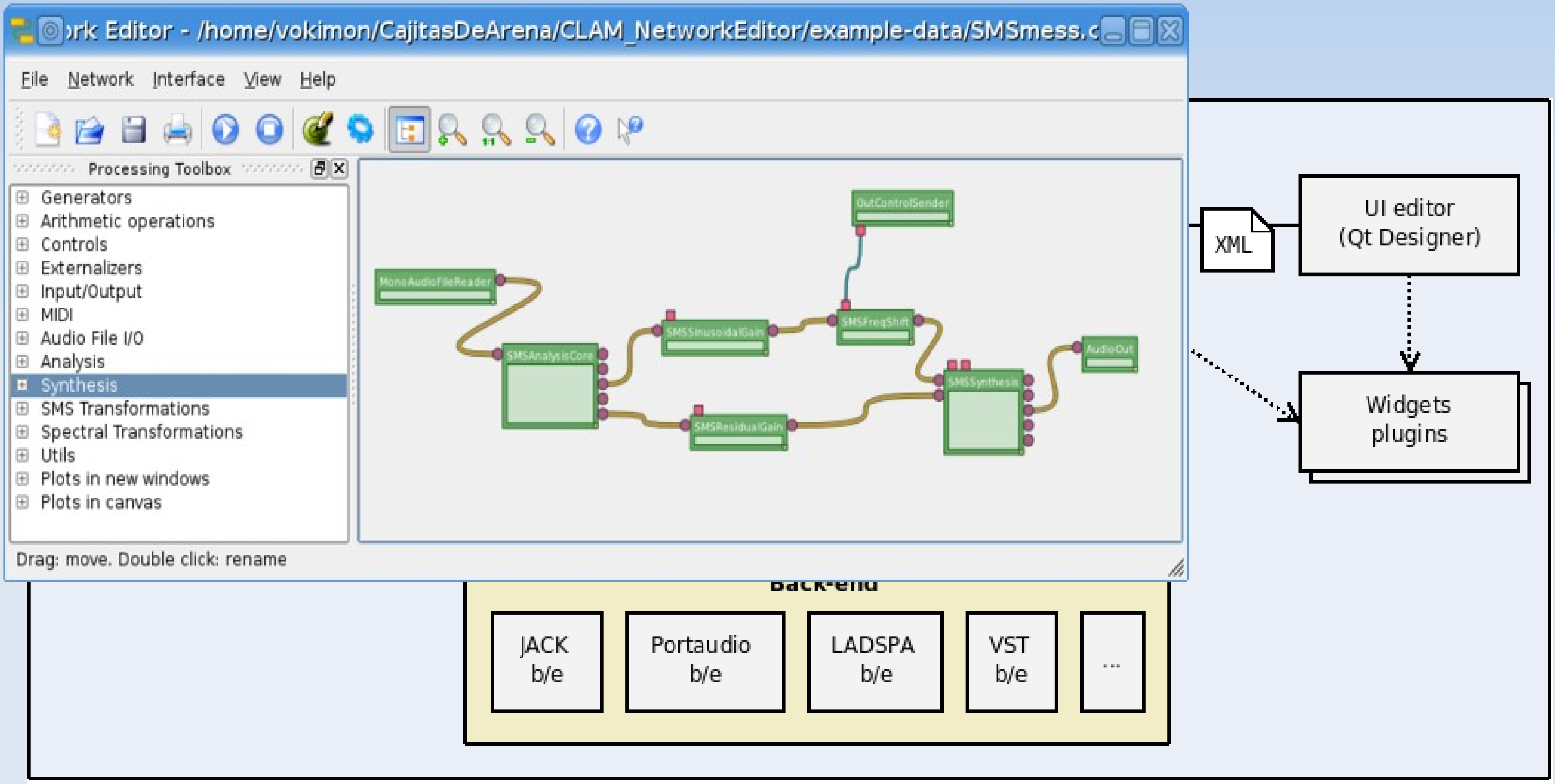

ork Editor - /home/vokimon/CajitasDeArena/CLAM_NetworkEditor/example-data/SMSmess.c
File Network Interface View Help
Processing Toolbox
Generators
Arithmetic operations
Controls
Externalizers
Input/Output
MIDI
Audio File I/O
Analysis
Synthesis
SMS Transformations
Spectral Transformations
Utils
Plots in new windows
Plots in canvas
Drag: move. Double click: rename
Qt Designer
File Edit Form Tools Window Help
Tonal Analysis - example-data/tonalAnalysis.ui
Widget Box
Object Inspector
Property Editor
Play
Stop
Playback Backend
Close
XML
Tonal Analysis
Play
Stop
Playing...
Close
XML

## Index

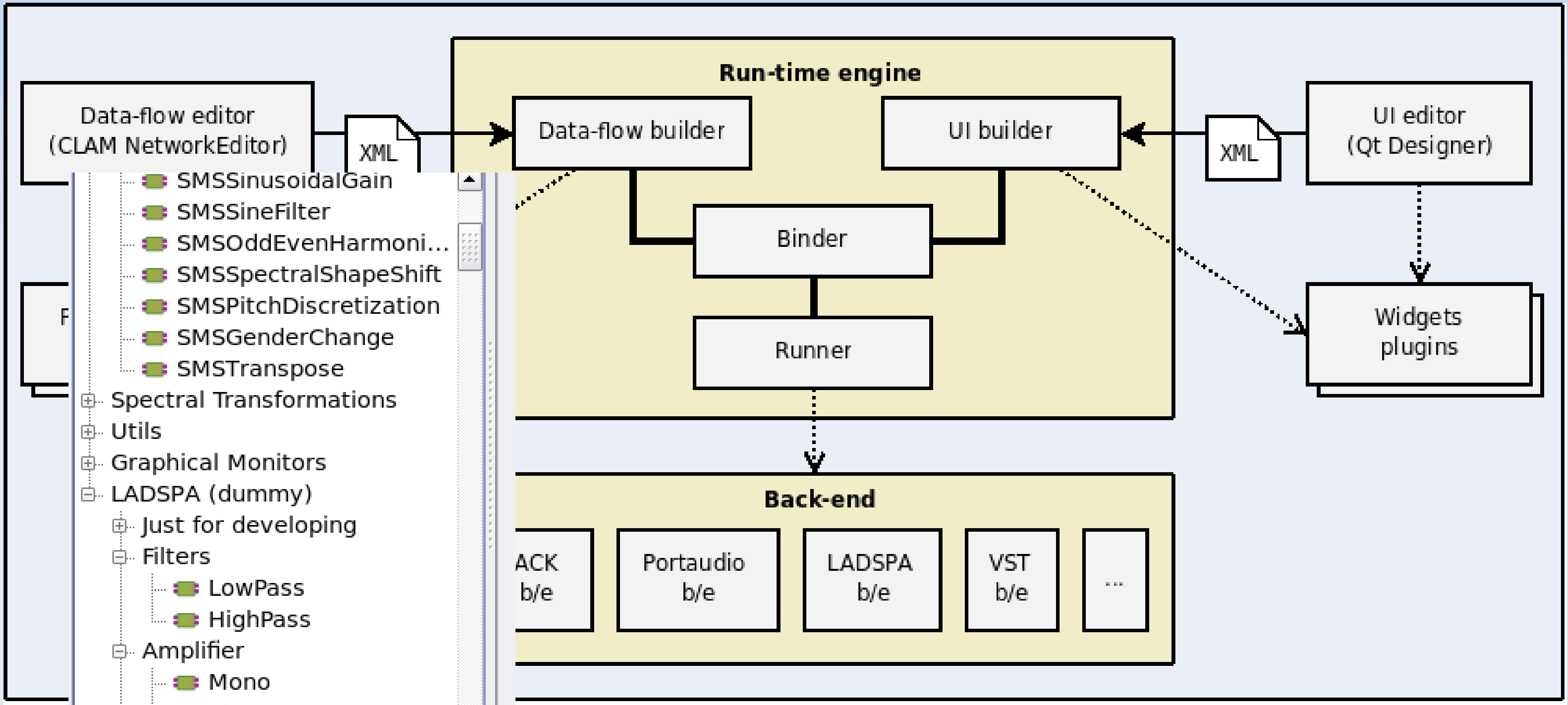
# Architecture

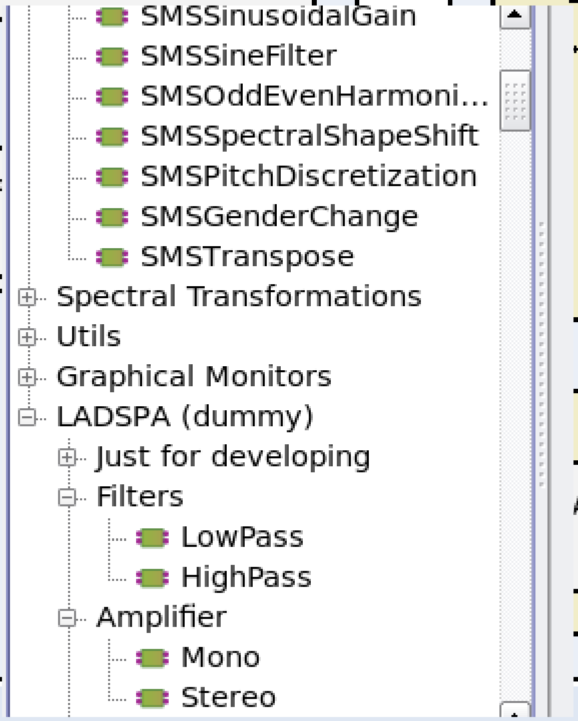


# Architecture: Data-flow editor

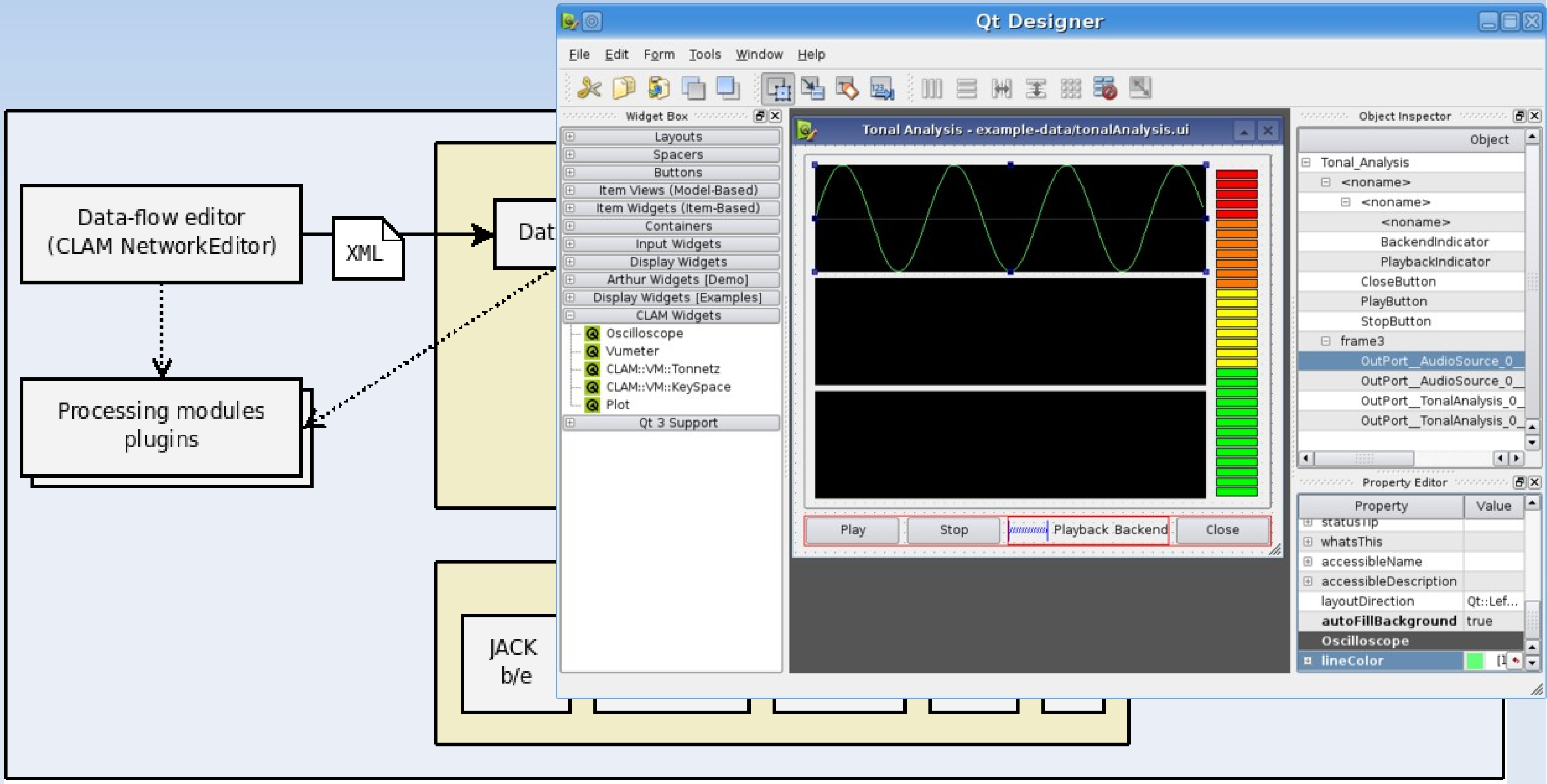


# Architecture: Processing plugins

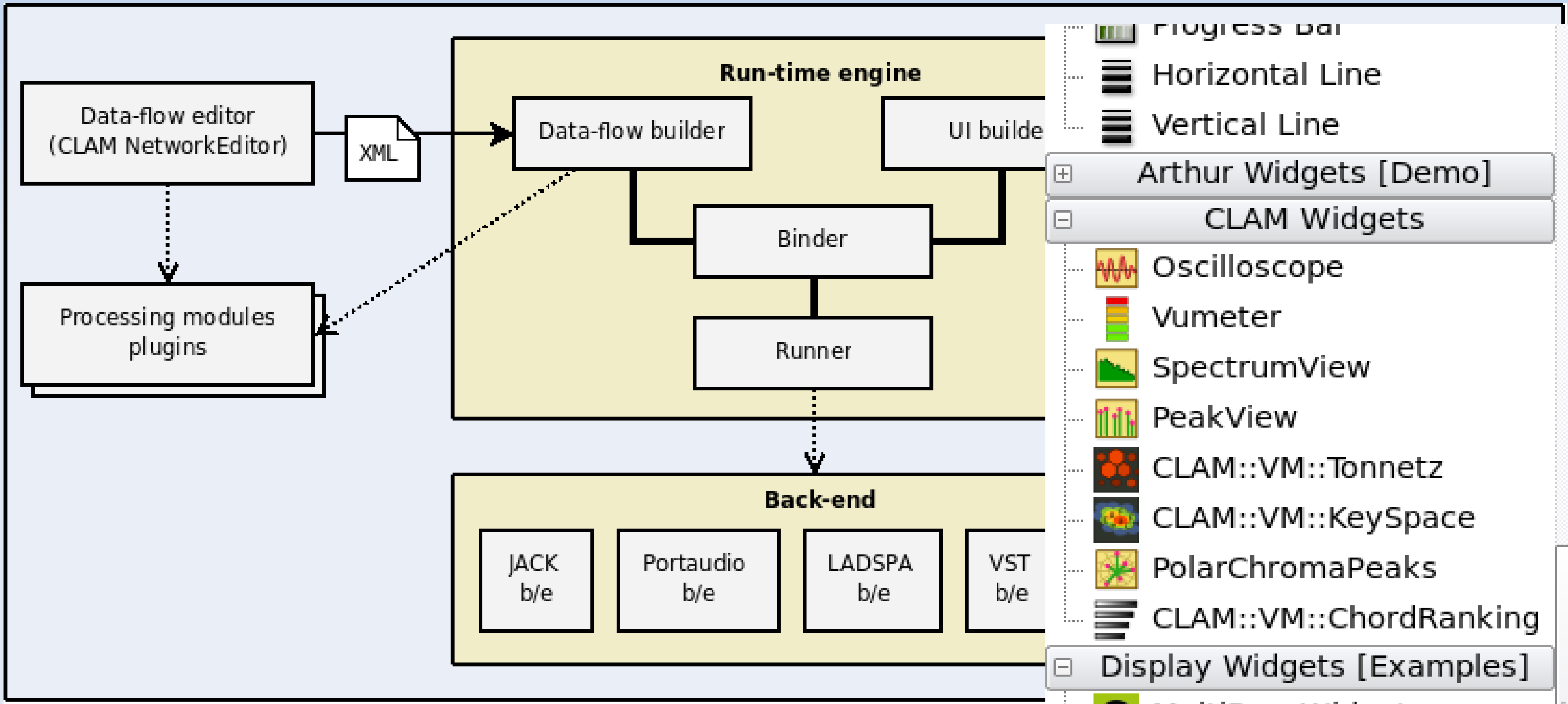


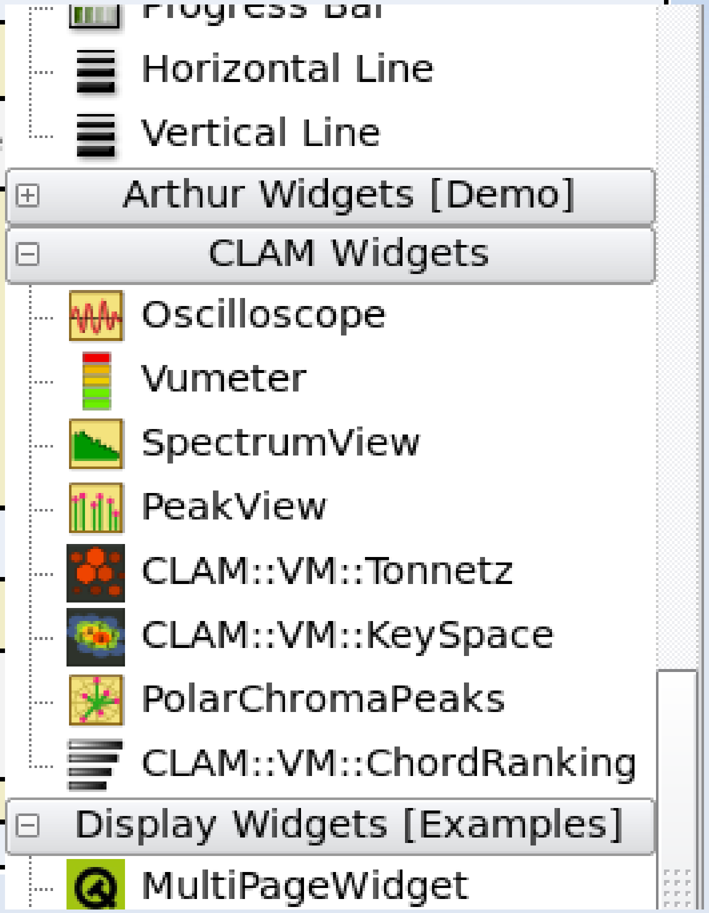

SMSSinusoidalGain
SMSSineFilter
SMSOddEvenHarmoni...
SMSSpectralShapeShift
SMSPitchDiscretization
SMSGenderChange
SMSTranspose
Spectral Transformations
Utils
Graphical Monitors
LADSPA (dummy)
Just for developing
Filters
LowPass
HighPass
Amplifier
Mono
Stereo

# Architecture: UI editor

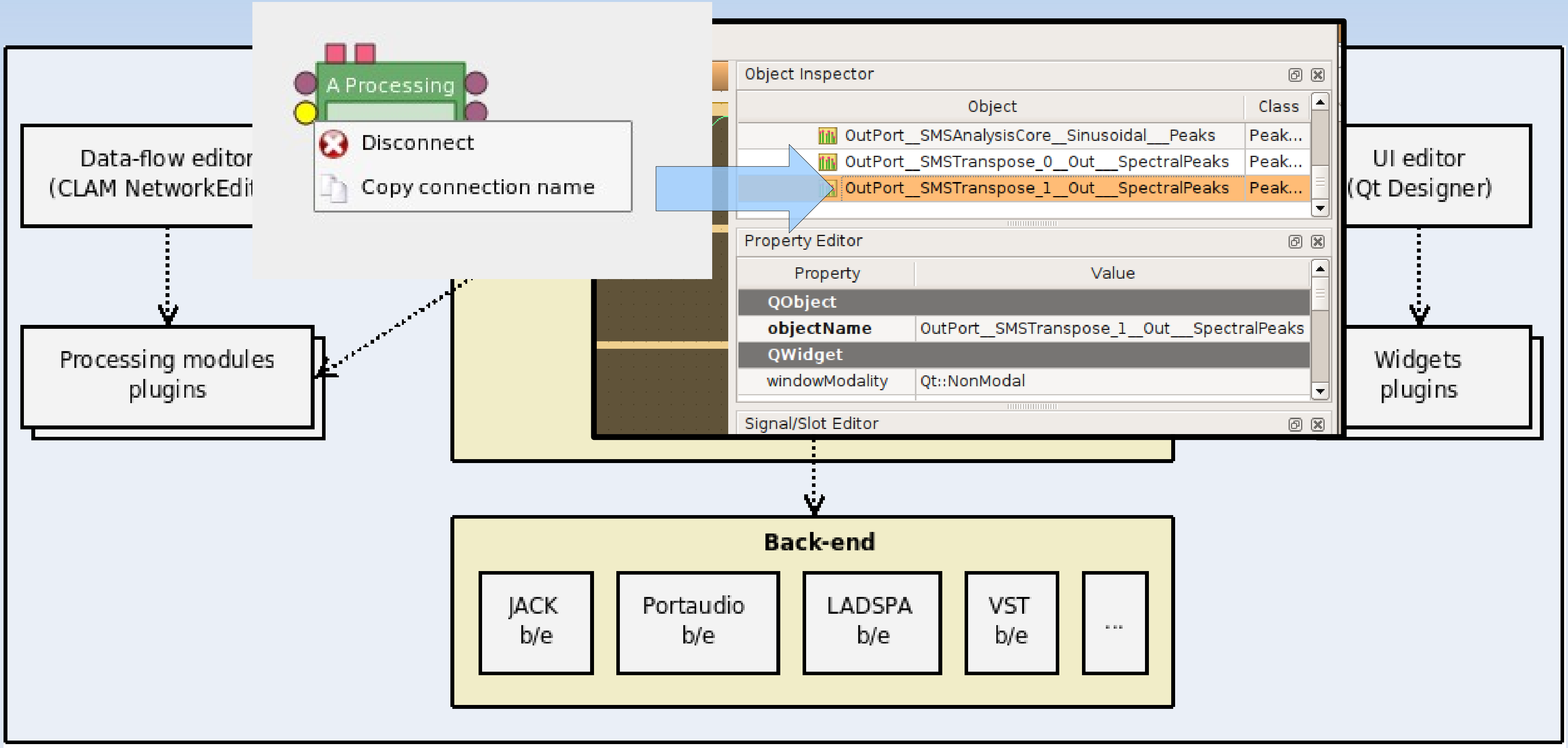


# Architecture: Widget plugins

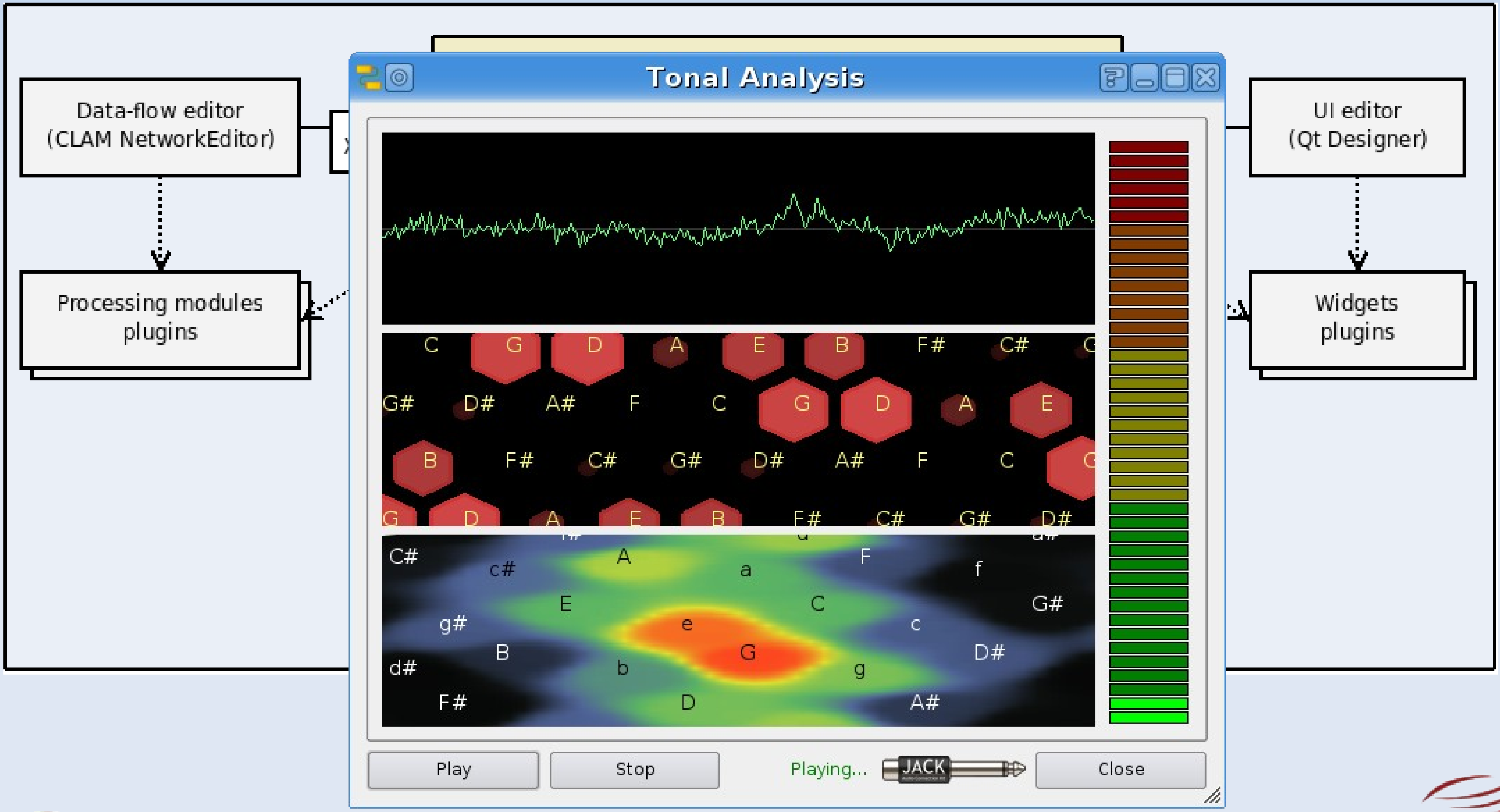




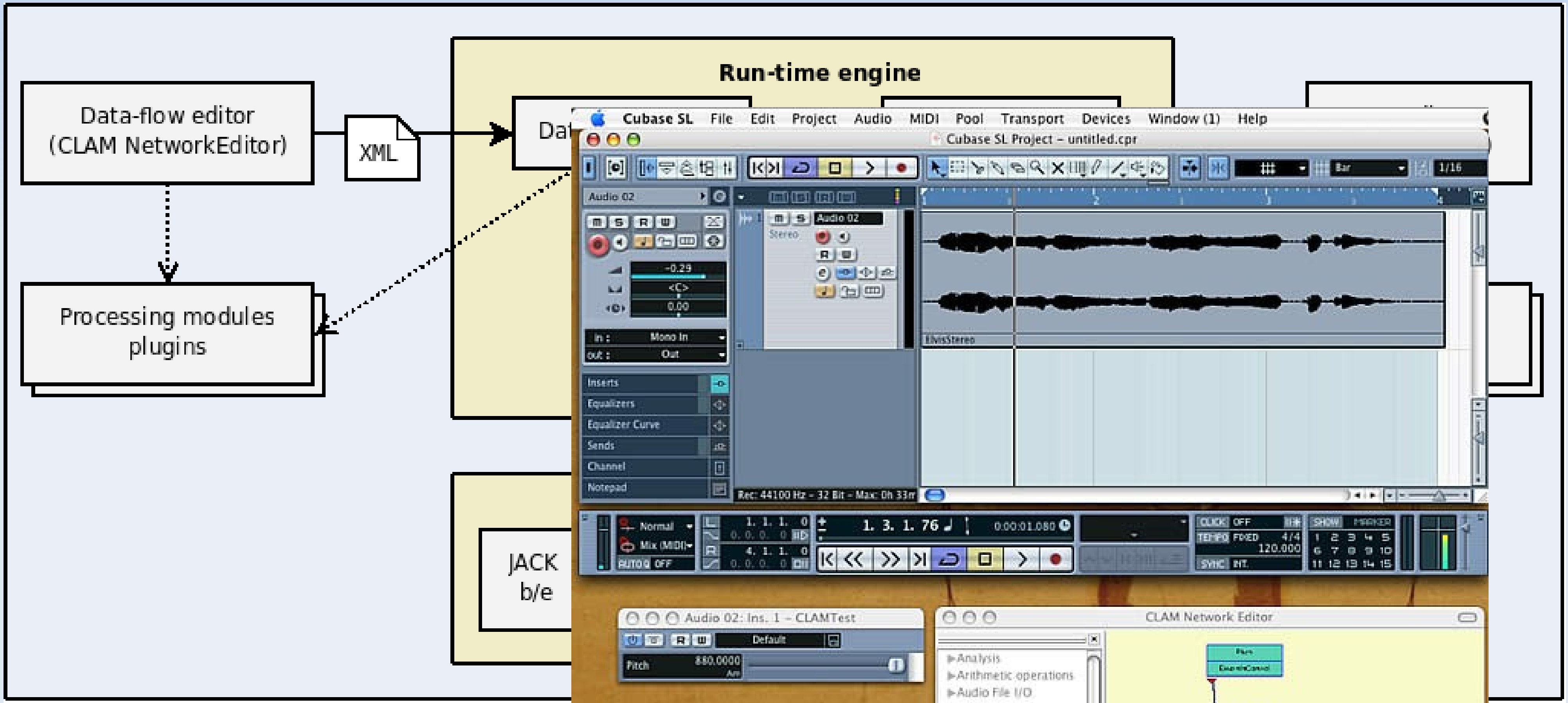
# Architecture: Binder

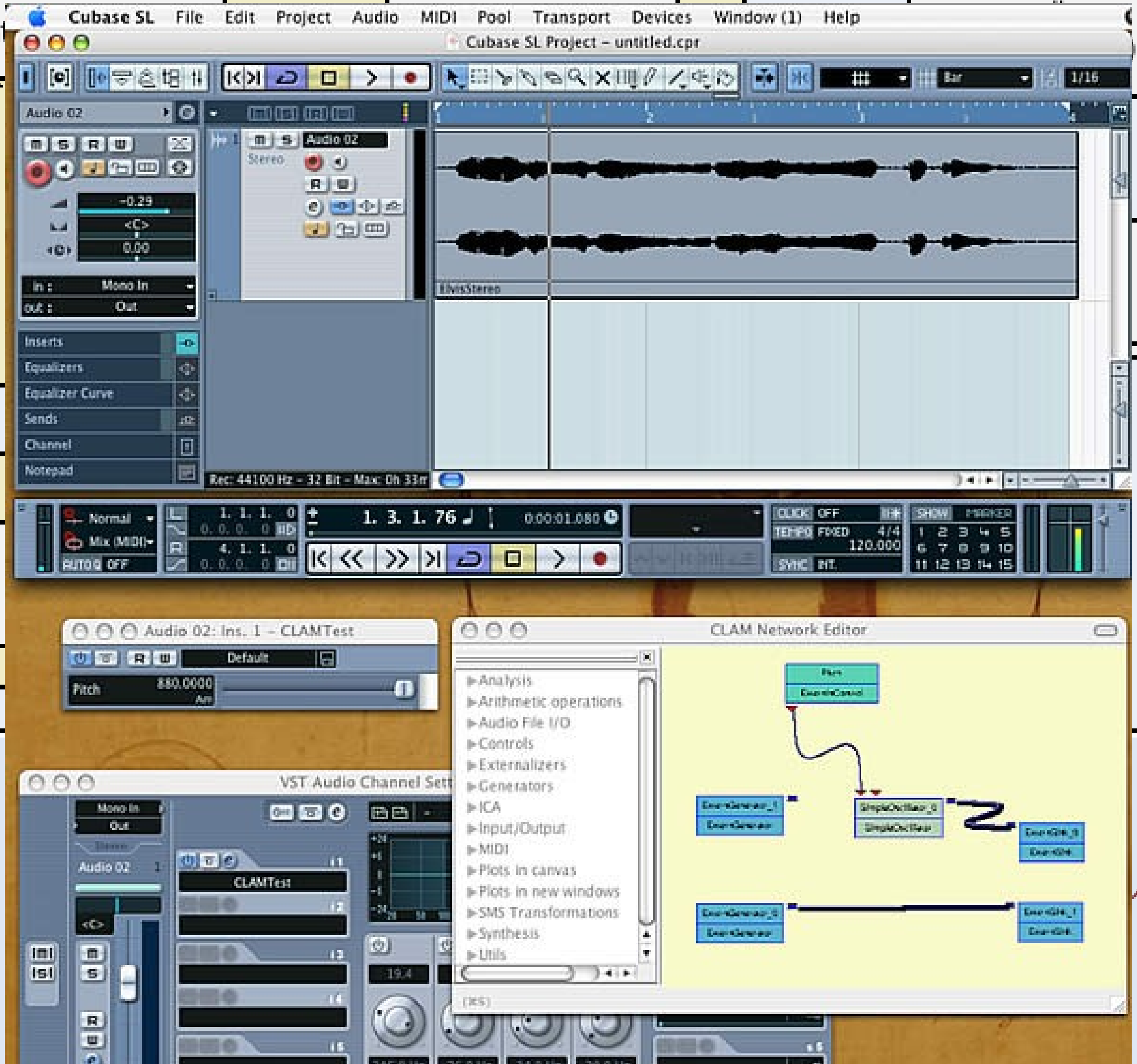


# Architecture: Runner



# Architecture: Back-ends



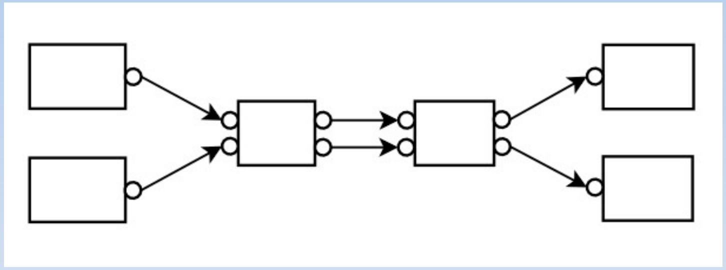


## Index

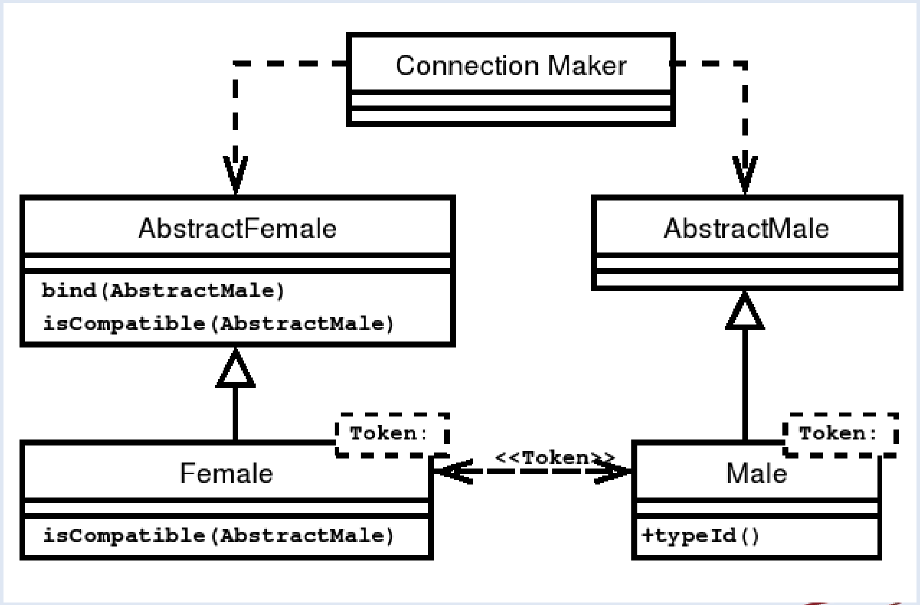
## Enabling design patterns

- **Data-flow and UI Builders:** How to build structures of objects of unknown types?
  - Factory pattern (Meyer)
- **Binder:** How to establish type safe channels between UI and data-flow?
  - Typed Connections
- **Runner:** How to do thread safe communication among UI and data-flow?
  - Port Monitor

# Pattern: Typed Connections

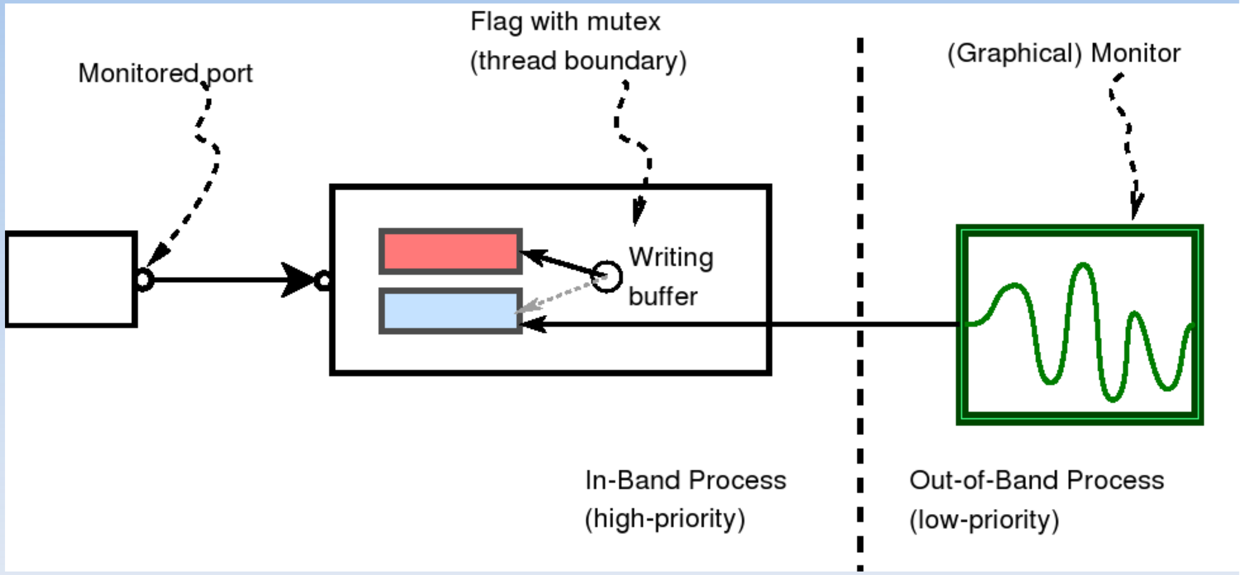


- Context
- Problem
  - Connectable entities communicate typed tokens but token types are not limited. Thus, how can a connection maker do typed connections without knowing the types?
- Forces
  - (1) Avoid type checking during process (2) Connections are done by the user, so they can mismatch the token types, etc.
- Solution ----------------------------------->



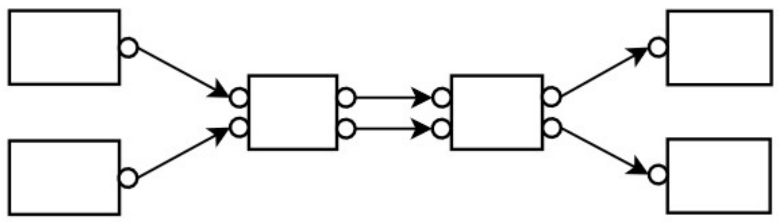
- Consequences
  - Connection maker is not coupled to token type. Modules are.

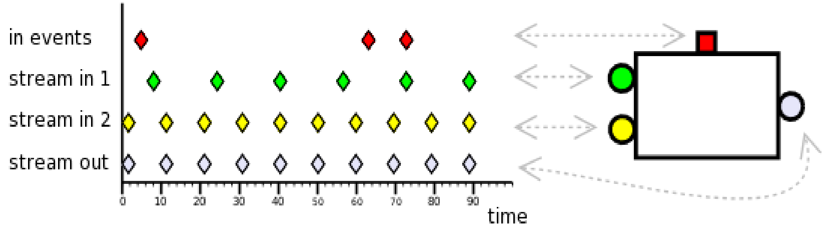
## Pattern: Port Monitor

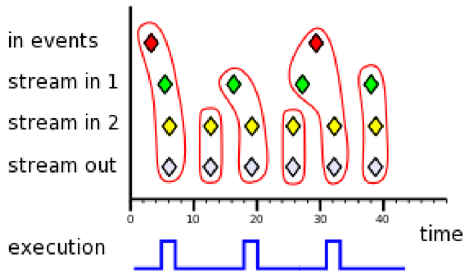

Monitored port
Flag with mutex (thread boundary)
(Graphical) Monitor
Writing buffer
In-Band Process (high-priority)
Out-of-Band Process (low-priority)

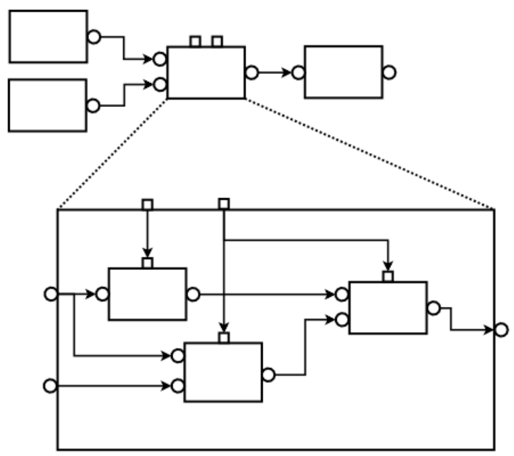
- Problem
    - We need to graphically monitor tokens being processed. How to do it without locking the real-time processing while keeping the visualization fluid?
- Solution
    - The solution is to encapsulate concurrency in a special kind of process module, the Port monitor, that is connected to the monitored out-port. Port monitors offers the visualization thread an special interface to access tokens in a thread safe way.

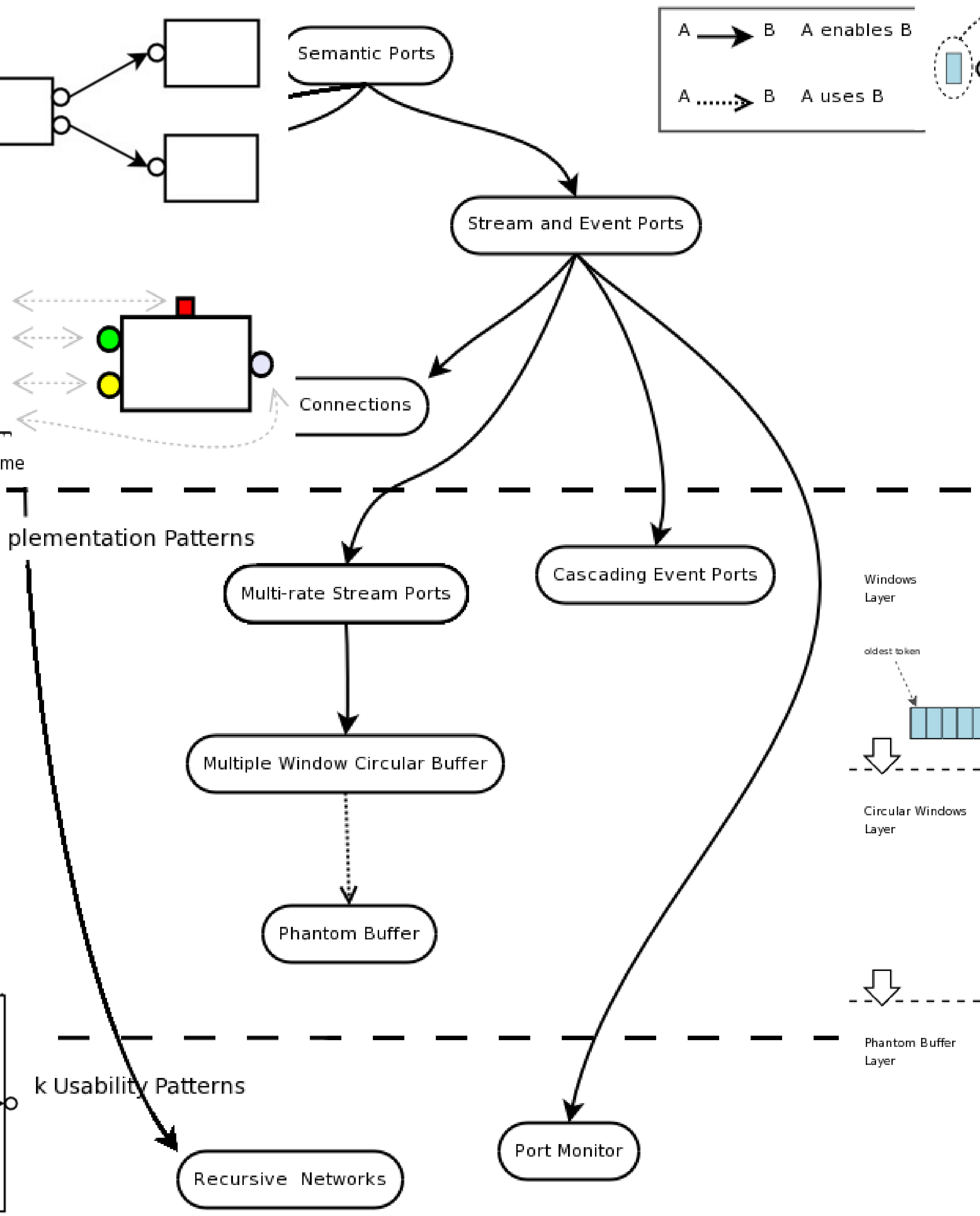
# Our Pattern Language

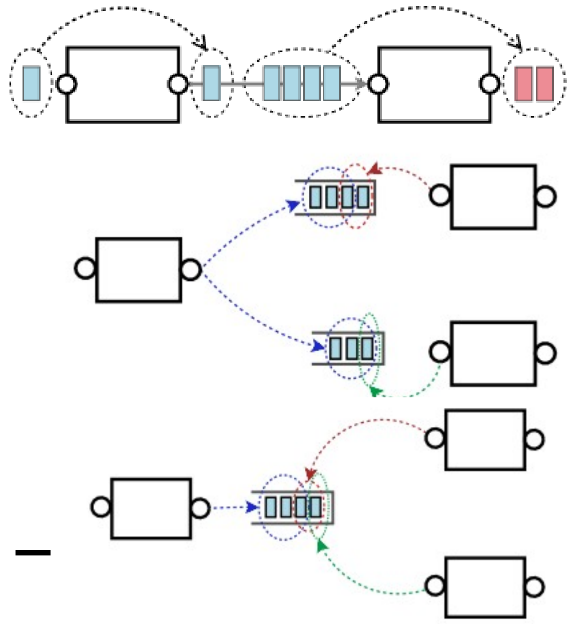


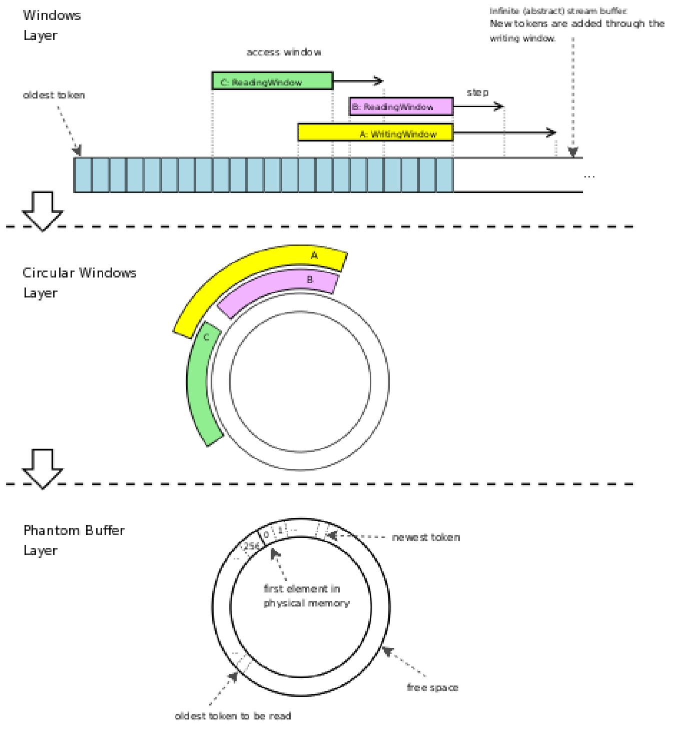












## Index

## Conclusions

- Already implemented: CLAM 1.0
- Full applications in few minutes
- Developers focus just on novel components
- Reuse processing and widgets using plugins
- Reuse design experience with patterns

## Conclusions: Future lines

- Extend the current use cases
- Address complex application work-flows:
  - Authoring tools, working with descriptors databases, batch processing...
- Enhance how to bind UI and data-flow.
- Drive the audio pattern catalogue into a community effort.

http://clam.iua.upf.edu

Questions?

Thanks