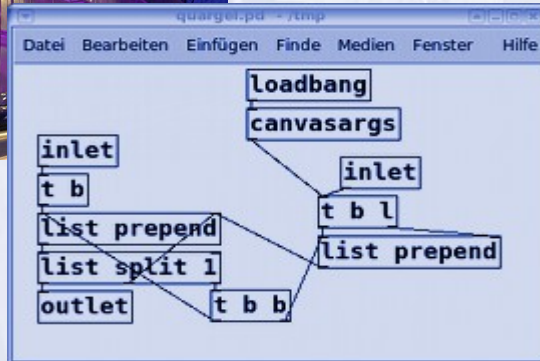


The IEM Demosuite

A large scale jukebox for the MUMUTH concert venue



Peter Plessas
 IOhannes m zmölnig

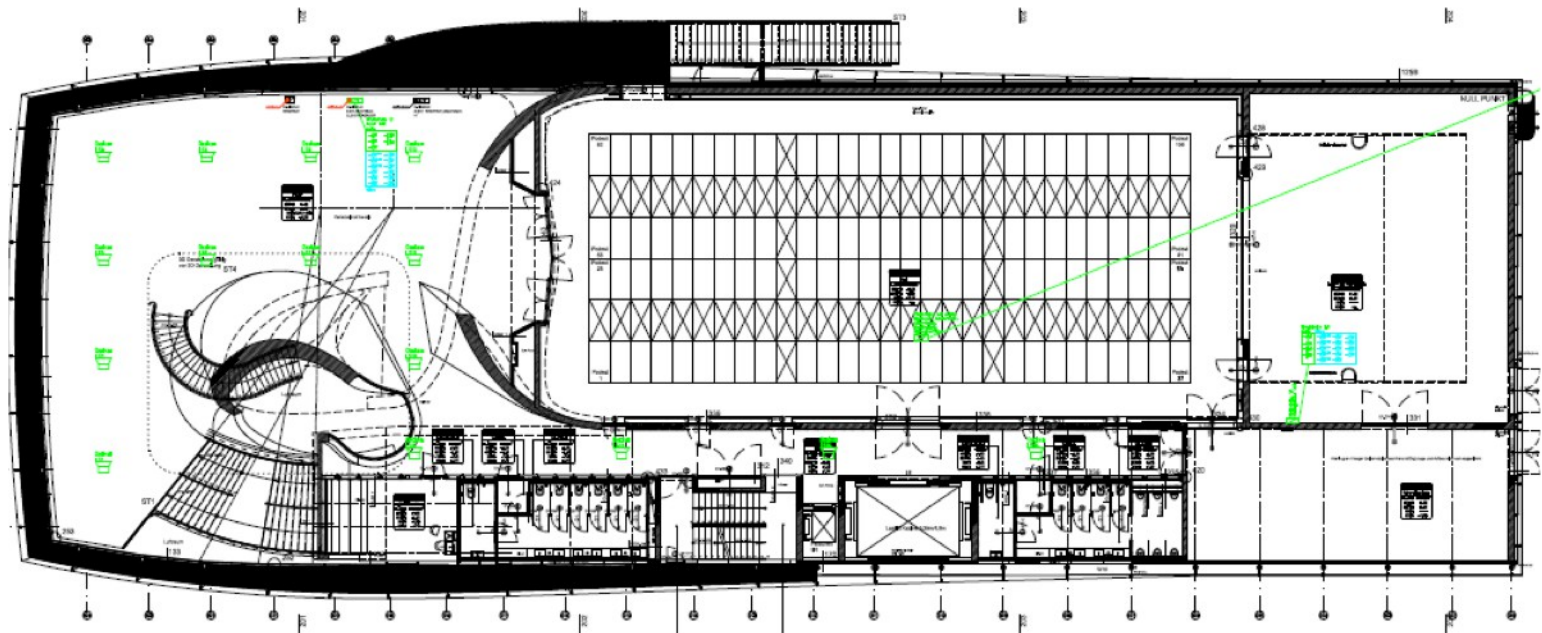


MUMUTH - House for Music and Music Theatre



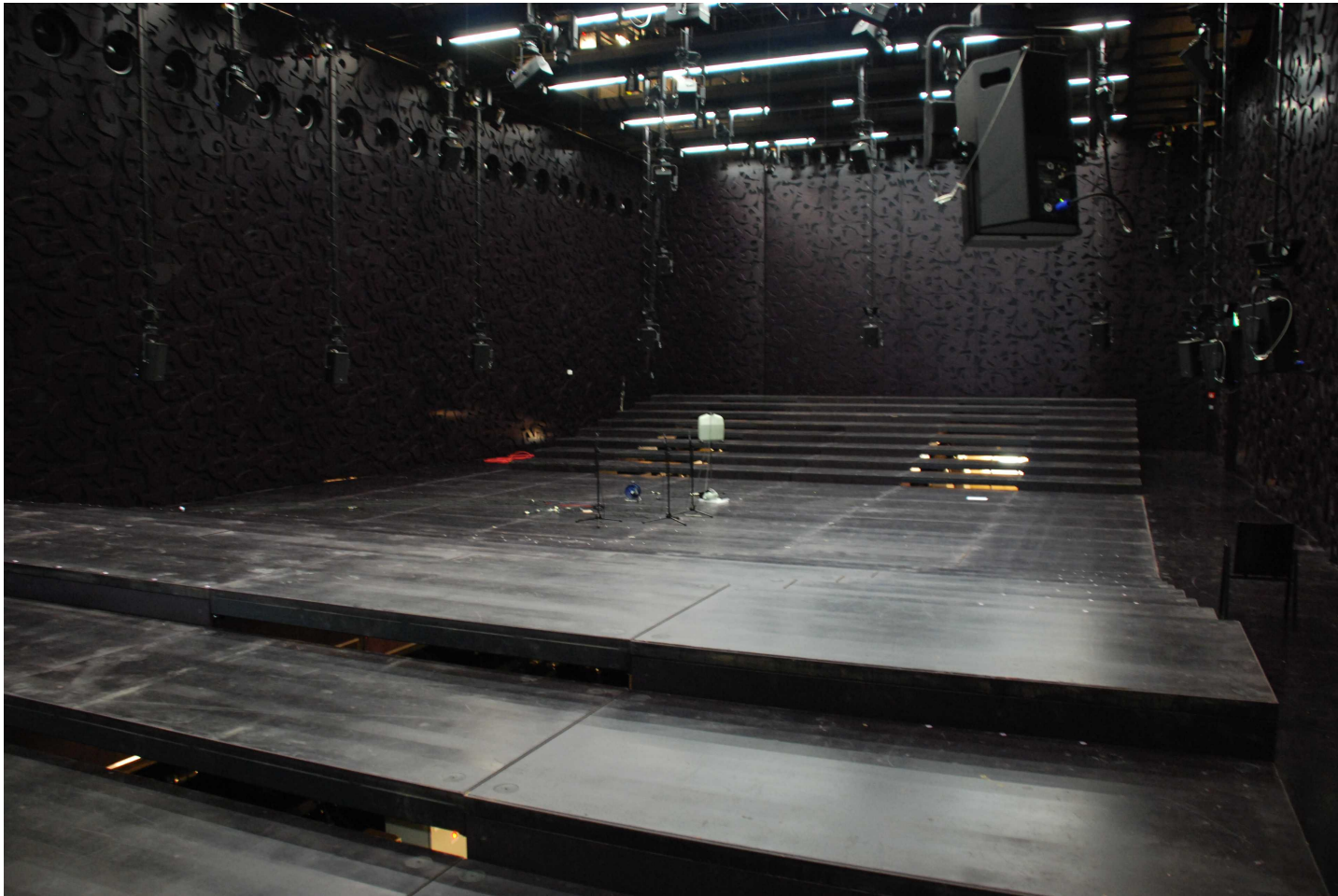
University of Music and
Performing Arts Graz

Performance space:
György Ligeti Hall



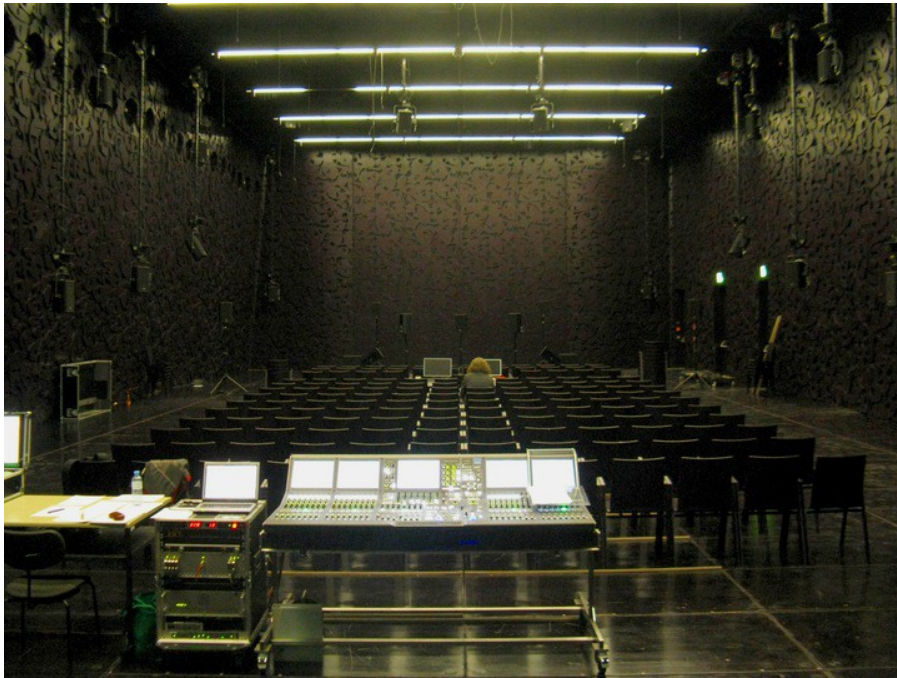
Györgi Ligeti Hall - True Multi Purpose

Open space for all departments and music styles



Györgi Ligeti Hall - True Multi Purpose

No single stage-audience position/orientation



No single sweet spot

Loudspeaker Installation

- How to combine an Ambisonics hemisphere with general purpose sound reinforcement needs?
- How to determine which loudspeaker setup fits all?

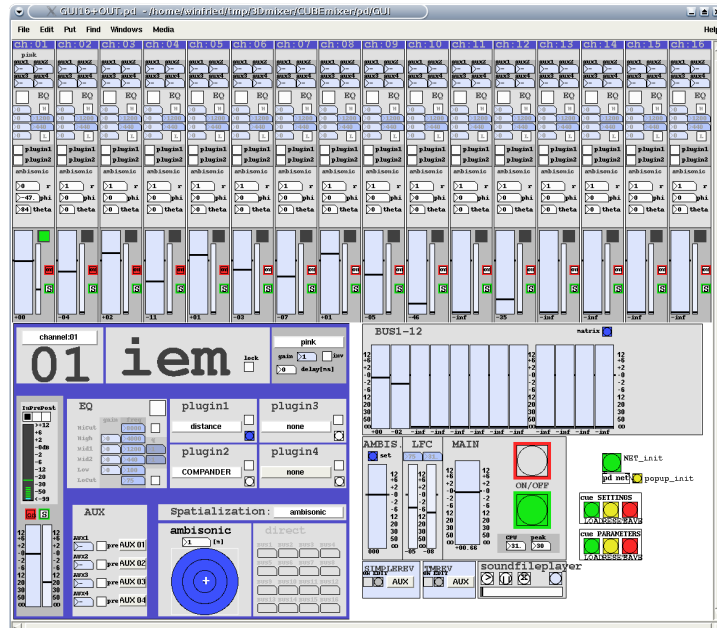
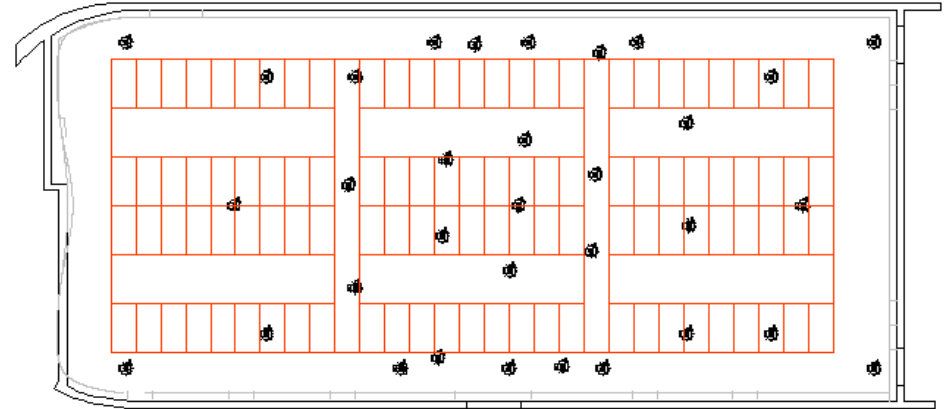
33 speakers on motor telescopes
with variable pan/tilt

Settings can be stored and recalled

- Change room configuration easily
- Switch setups in concert
- Compare changes aurally and instantaneously

Ambisonics in the Ligeti Hall

- Hemisphere
Minimum energy approach
(Thomas Musil)



Higher-order Ambisonics:
IEM CUBEmixer (LAC 2008)

Audio Infrastructure - 1

- Lawo HDcore + mc² 66 fanless mixing console

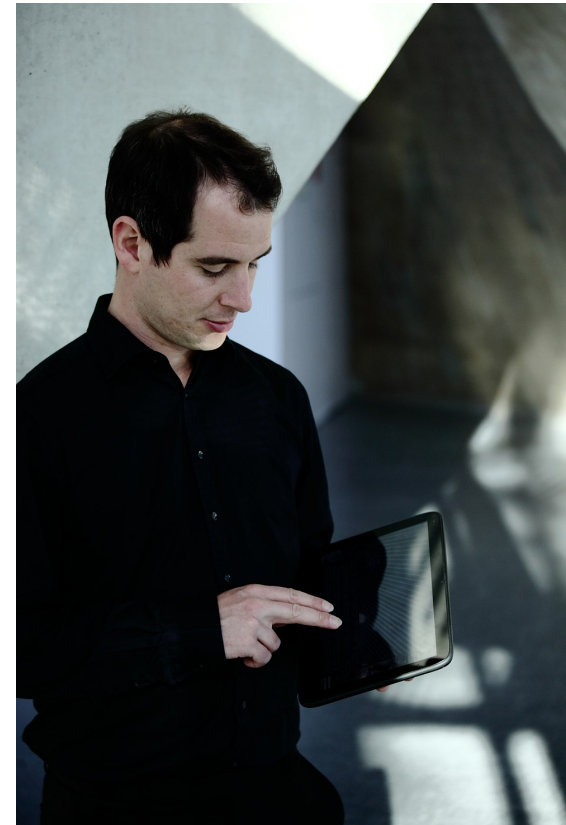


Audio Infrastructure - 2

- Lawo Dallis I/O interfaces (4096 physical channels)
- MADI fiberoptic interconnections in and between buildings
- Audio computer with dual MADI cards (RME) running Debian/GNU Linux in separate server room
- Virtual room acoustics system (Meyersound Constellation)
- $33 + 12 + 64 + 12$ subs = 121 individual loudspeaker channels!

Demosuite - Motivation

- We invite everyone to listen and explore!
- An exciting large instrument at your fingertips?
- An easy way to present your work? Sure!
- It's gotta be
ready
fun
safe
.... to use



Demosuite - Challenges

- Ready:
 - Minimum preparation time
 - Usable during other projects and room configurations
- Fun:
 - Intuitive + appealing
 - Unobstrusive + portable
 - Growing repertoire + invite contribution
- Safe:
 - Don't hurt any ears!

-> The quest for the ultimate jukebox!

Challenge: Remote Controlling

- How to Remote Control a Demosuite?
- available system
 - Mixing Desk + KVM-Extender
 - allows tight integration with existing system
 - BUT time (and space) consuming to setup
- preferred system
 - portable wireless device
 - WiFi-connection to DSP-System

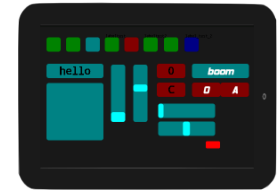
Software Choice

- Which DSP engine to use?
 - Pure Data
 - well known
 - IEM's prototyping platform of choice
- What to use as GUI toolkit?
 - Pd
 - well known
 - visual by nature
- Pd drawbacks
 - not much visual appeal
 - no auto-scale to different screen-sizes
 - DSP/GUI entanglement

→ let's ignore them...

Remote Control - Tablet

- Pd on a tablet
 - existing projects
 - RjDj
 - libPd
 - PdDroidParty
 - BUT running Pd's *GUI* on a tablet
 - needs Tcl/Tk
 - iOS
 - Android
- Neophonie's WeTab
 - MeeGo-linux: comes with X-server!
 - i386



Demosuite - Setup

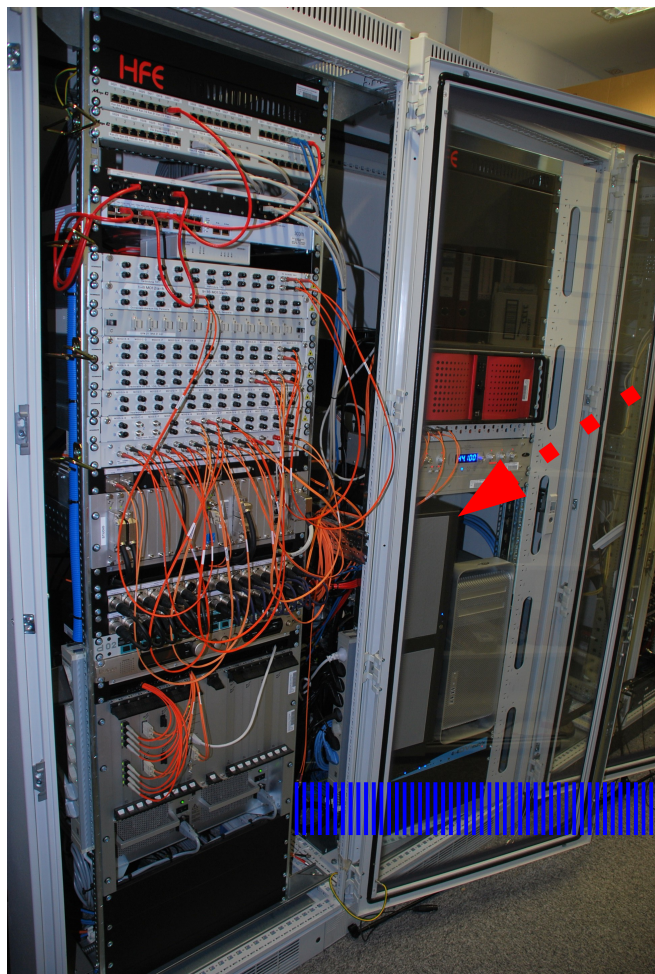


Photo: S.Warum

WiFi

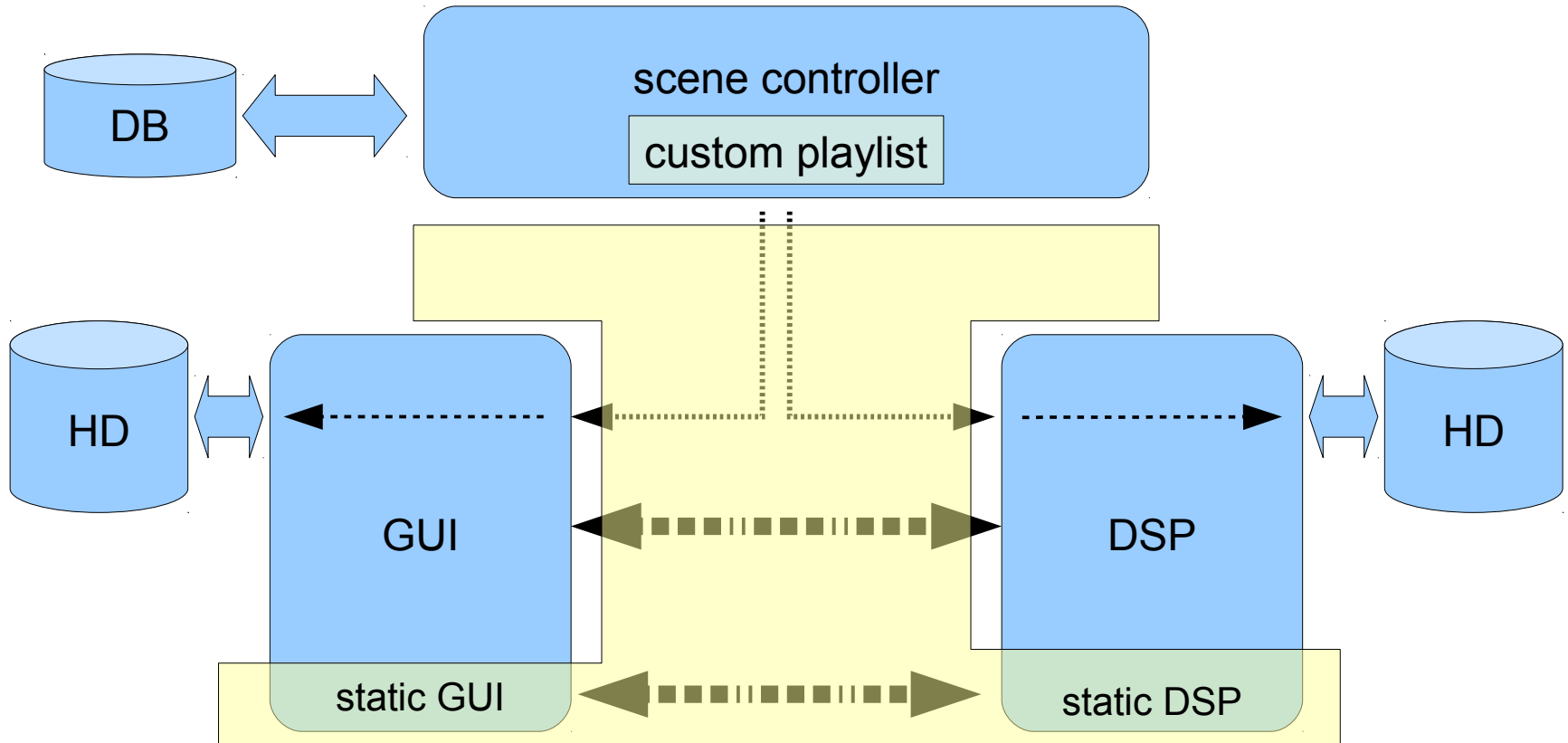


Audio



Photo: G.Eckel

Demosuite - Software modules



Static components

- Audio (Control)
 - master fader
 - Multichannel limiter
 - Mute
- scene selection / playlist
 - manages „database“ of existing scenes
 - playlist editor
- inter process communication
 - Network connection
 - Scene loading

Scenes

- variable number of *demo* scenes
- each scene consists of 2 patches on 2 computers
 - GUI
 - DSP
 - including media files
- simple deployment of scene programs/media
 - using Subversion/rsync
- consistency between hosts
 - make sure that both GUI & DSP have the same state
- isolation between different scenes
 - scenes *mutually exclusive*
 - avoid side-effects

Scene Isolation

- load-on-demand
 - only one scene instantiated at any time
 - rather than pre-loading all scenes

PRO

- Perfect isolation
- Less resources
- Scales well

CON

- Slow load times
- Audio blocked while loading

- Consistent loading of scenes
 - asynchronous loading requires synchronization constructs
 - → serial loading

Scene Isolation

- load-on-demand
 - only one scene instantiated at any time
 - rather than pre-loading all scenes

PRO

- Perfect isolation
- Less resources
- Scales well

CON

- Slow load times
- Audio blocked while loading

- Consistent loading of scenes
 - asynchronous loading requires synchronization constructs
 - → serial loading

Serialized Distributed Loading

GUI

- user selects sceneB
- delete sceneA::GUI
- enter transitory scene
- request sceneB

- ▶ load sceneB::GUI
- initialize sceneB::GUI
- *(request state from DSP)*

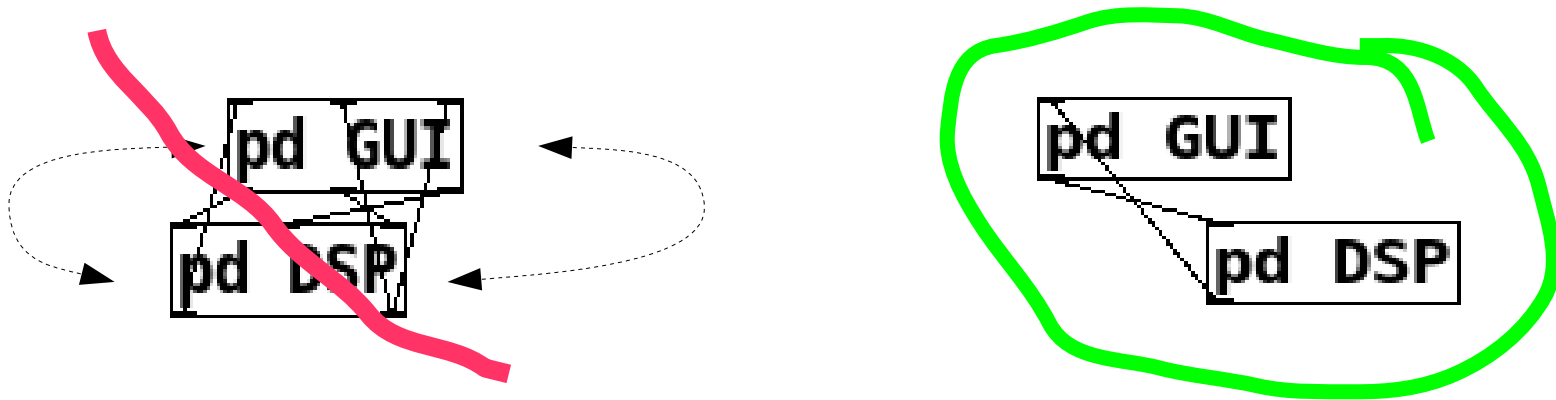
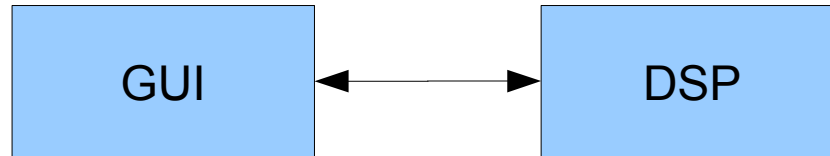
DSP

- delete sceneA::DSP
- load sceneB::DSP
- initialize sceneB::DSP
- *report loading done*
- request sceneB

something rotten...Data Consistency

- not much state in small demo-scenes
- Master-Slave architecture
- DSP
 - authoritative master
- GUI
 - can request state via special message
 - OSC-message without arguments

Bidirectional Data Connection



- Enforce single communication channel
 - transparently handled via network

Communication Protocol

- transport layer: UDP
 - TCP/IP might hang
- „OSC-style“ messages
 - OSC nice for compat

BUT

- low-level UDP-connection in Pd unstable
 - FUDI „tunnel“
 - LATER: switch to „real“ OSC

Safety and Security Considerations

- Connection Monitoring
 - heartbeat
 - ping: GUI → DSP → GUI
 - connection loss
 - DSP mutes
 - GUI: switch to transitional scene
 - autoreconnect
 - reloading of GUI-scene
 - *(request state from DSP)*
- Security
 - non-existent
 - (rely on WPA2)

Existing Repertoire of Scenes

- Pre-rendered audio scenes featuring spatialization and distance perception through HOA reverb.
- Examples of IEM productions
- Fixed media composition by IEM staff and guest composers
- Ambisonics Field recordings, soundscapes
- Comparison of different spatialization strategies
- Dynamic sound source positioning
....more to come!

Acknowledgements

- We would like to thank our colleagues:
Gerhard Eckel: Project initiation
Ulrich Gladisch: MUMUTH sound engineering
Thomas Musil: Speaker hemisphere layout
Stefan Warum: Sound reinforcement design
- Thank you Linux audio community!
- Thanks to CCRMA for hosting LAC 2012!