

BLENDERCAVE

MULTIMODAL SCENE GRAPH EDITOR
FOR VIRTUAL REALITY



D. Poirier-Quinot,
D. Touraine and B.F.G. Katz

Presentation Plan

1. Framework

- ▣ Scene Graph Editors in VR
- ▣ BlenderCAVE context

2. BlenderCAVE

- ▣ Architecture
- ▣ Sound Rendering Engine

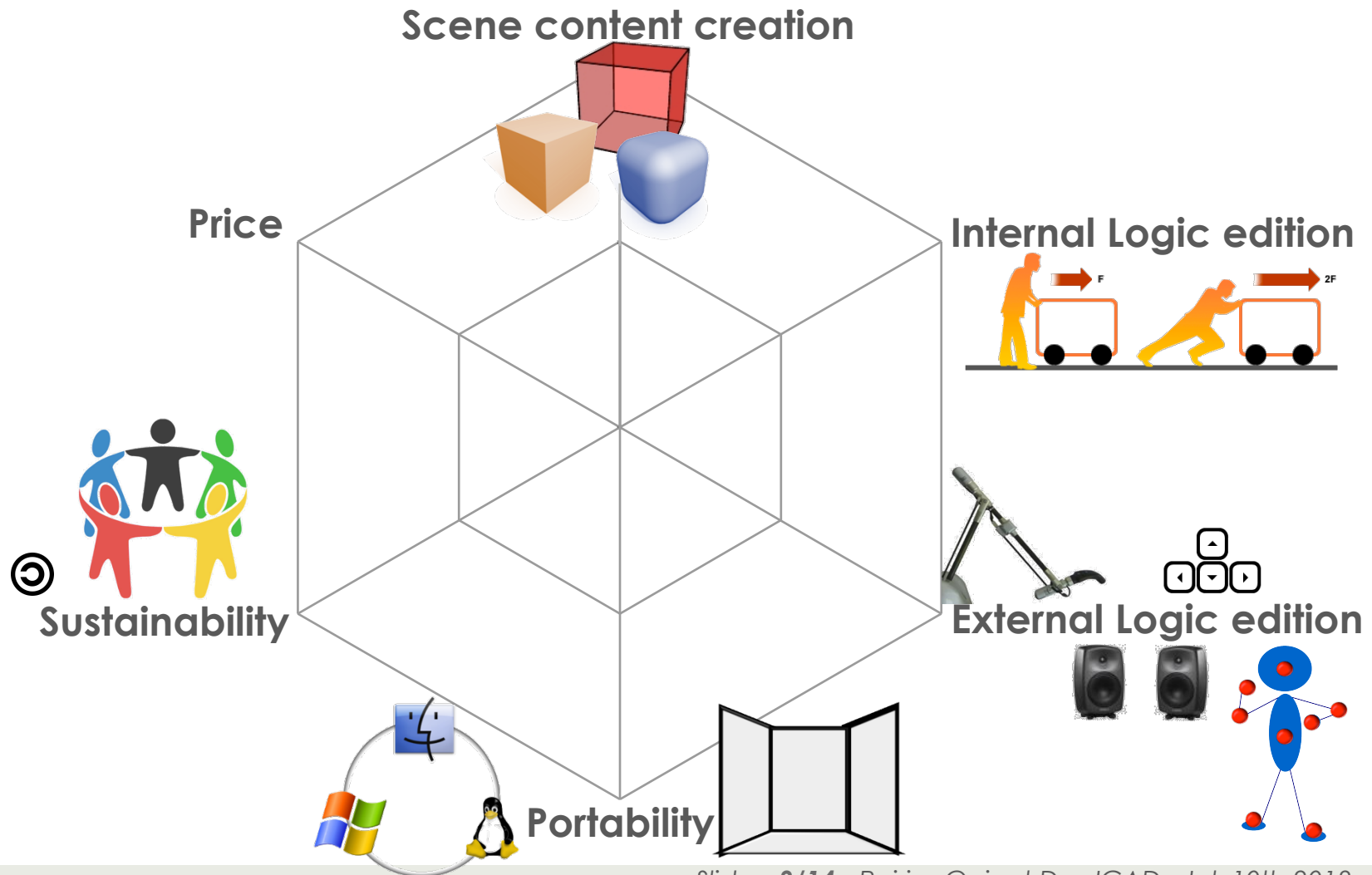
3. Usage Considerations

- ▣ Performances
- ▣ Scene Creation



1. Framework– SGE in VR research

■ Key features



1. Framework– SGE in VR research

Commercial and Public Domain Solutions: Pro and Cons

Commercial

- ✓ Sustainability
- ✓ Available Features
- ✓ Community and After-Sale Service.

Public Domain

- ✓ Price
- ✓ Improvable Features
- ✓ Community and Software developers

✗ Portability

✗ External Logic Edition

Quest3D

3dvia virtools

MiddleVR For Unity

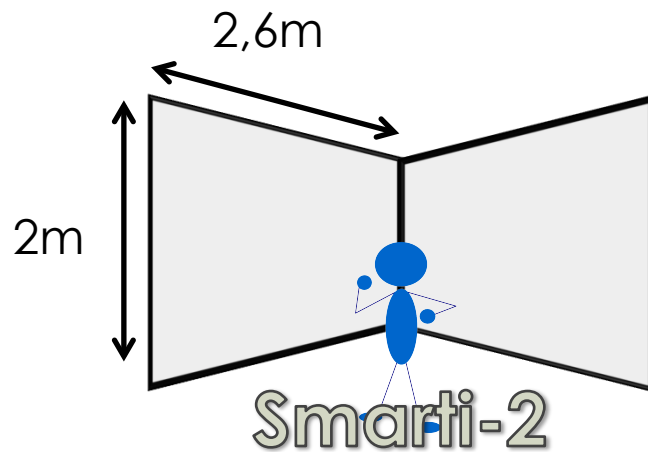
Multimodal
Affective
Reactive
Characters

CaIVR

1. FrameWork – BlenderCAVE Context

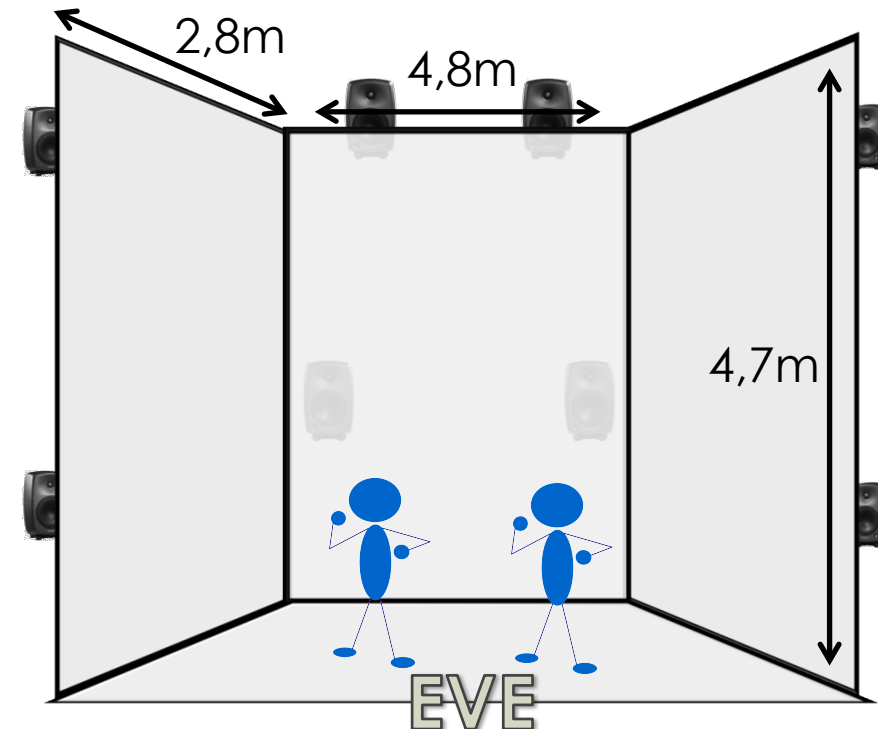
Smarti-2 system

- 2 screens / 4 projectors
- Wave Field Synthesis and Ambisonic



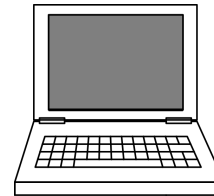
EVE system

- 4 screens / 7 projectors
- 2 users adaptive stereoscopy
- HOA Ambisonic and Binaural

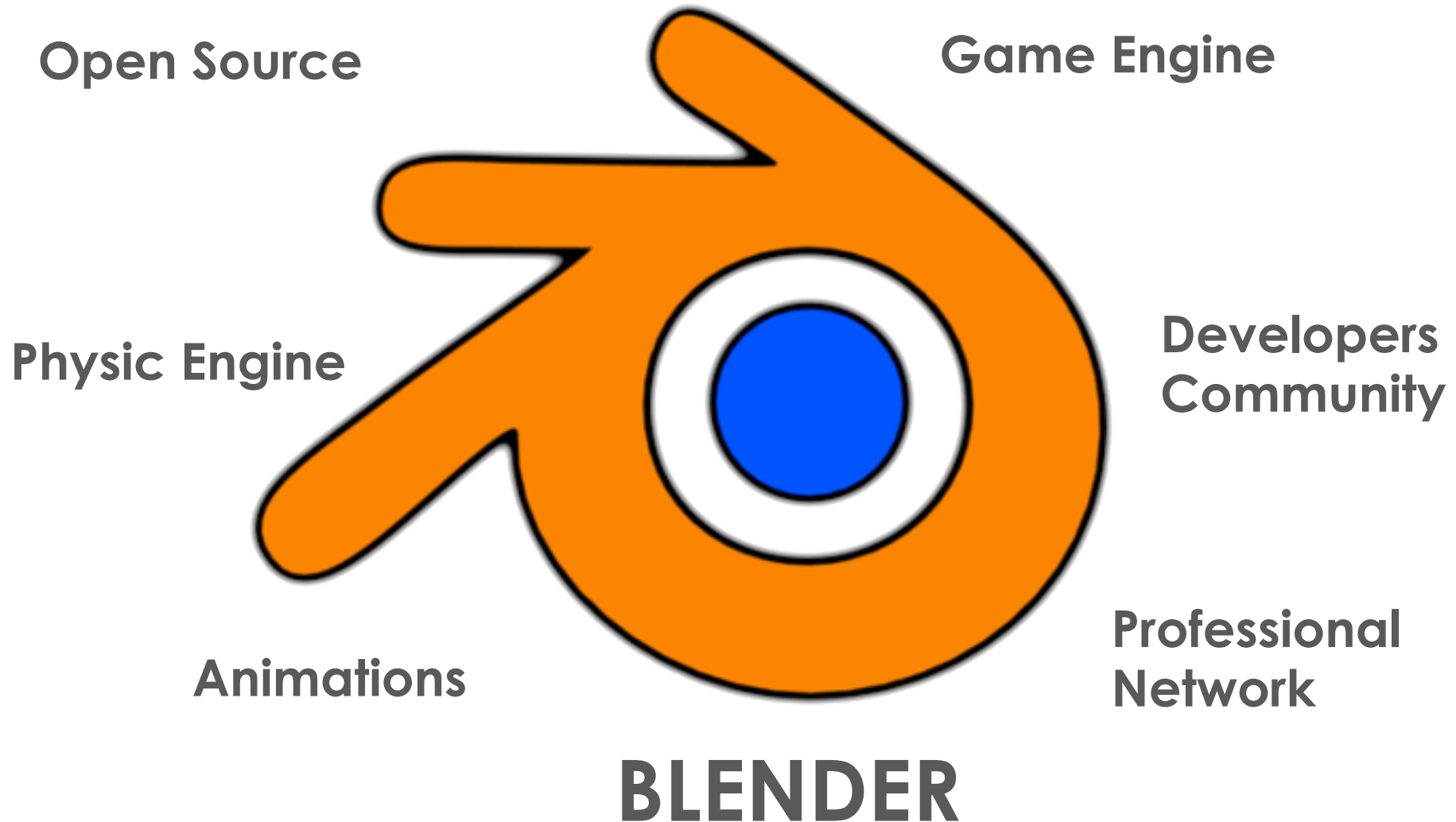


Laptop

- 1 screen
- Stereo and Binaural



1. FrameWork – BlenderCAVE Context



1. FrameWork – BlenderCAVE Context

■ From Blender to 2013 BlenderCAVE

[www.gmr.v.es/
~jgascon/BlenderCave/](http://www.gmr.v.es/~jgascon/BlenderCave/)

2011 BlenderCAVE, by GMRV

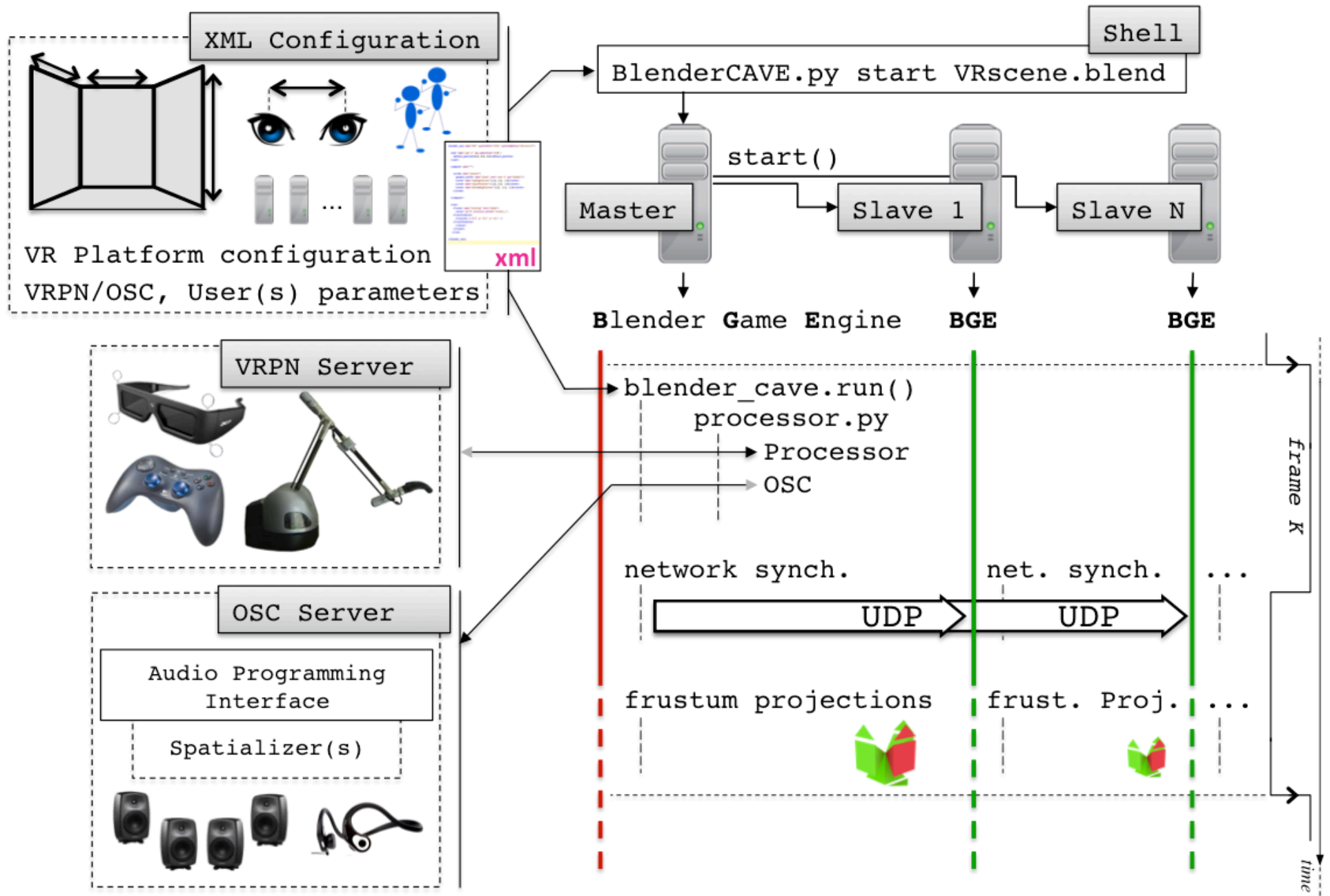
- ✓ Video Wall Display
- ✗ “Manual” synchronization
- ✗ Stereoscropy

2013 BlenderCAVE Improved, by LIMSI

- ✓ Adaptive Stereoscopic Rendering
- ✓ Master / Slave synchronization process
- ✓ External Messages processing
- ✗ Patching Blender sources



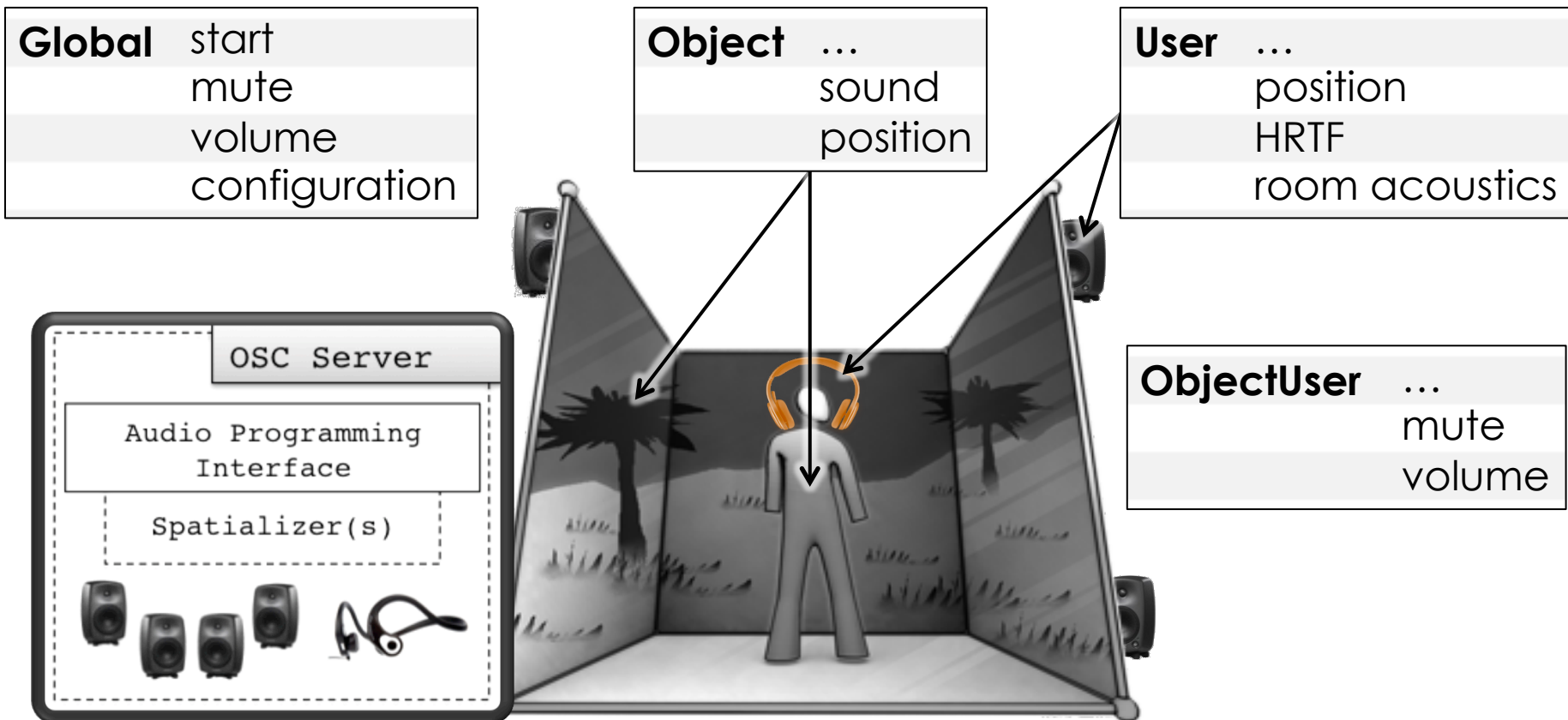
BLENDERCAVE 2011



2. BlenderCAVE – Architecture

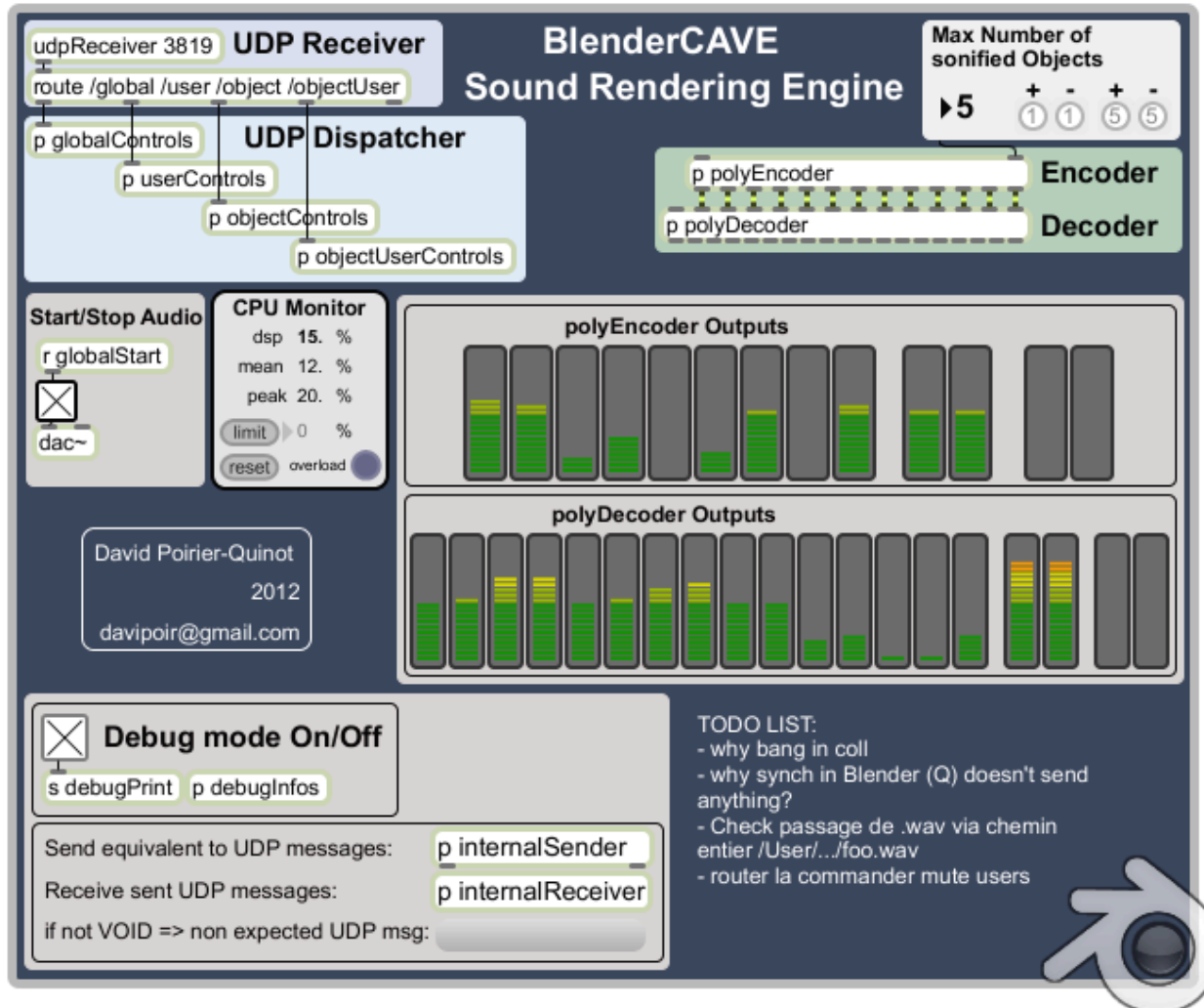
- ▣ **Embedded OSC API** – Easy communication with the Sound Rendering Engine

Implemented Classes:



2. BlenderCAVE – SRE

Max/MSP based Sound Rendering Engine Implementation



✓ **Transparent**

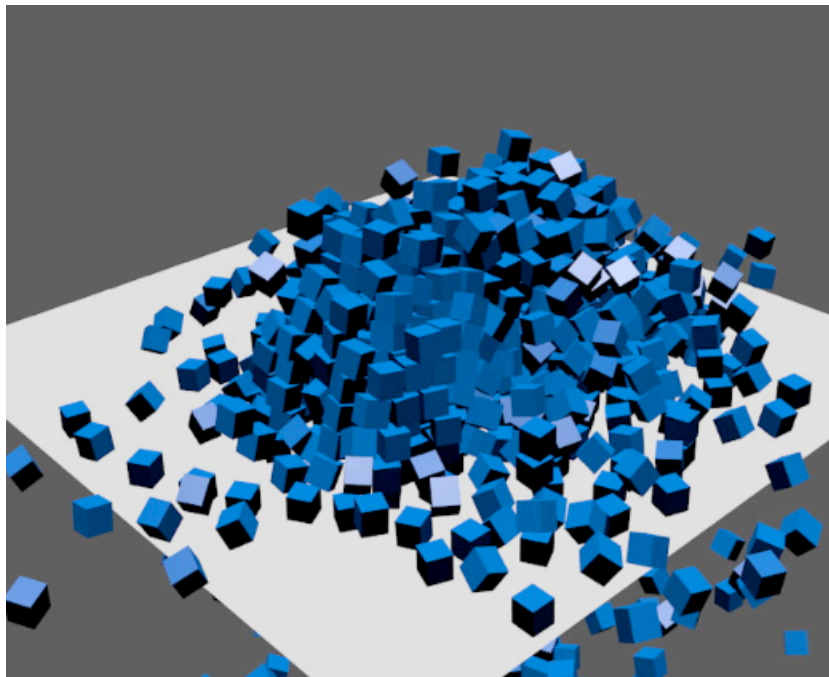
✓ **Dynamic** object instantiation (poly~)

✓ **Commutable Spatialization Engine**

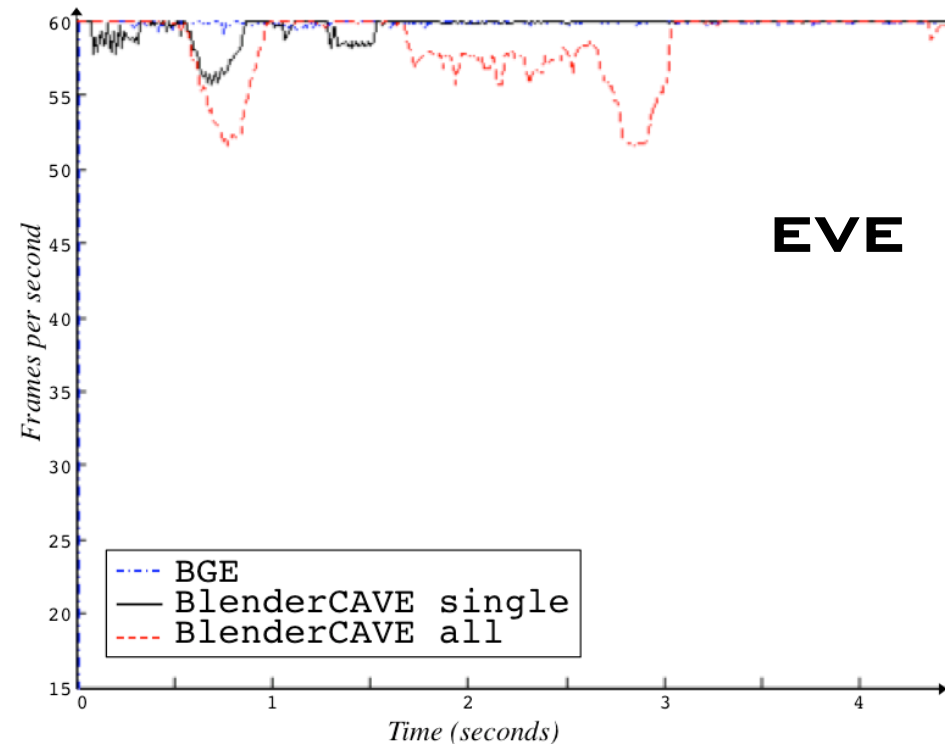
3. Usage – Performances

Work in progress

Enhance synchronization process: FPS stress-test (980 cubes)



REAL-TIME PHYSIC ENGINE

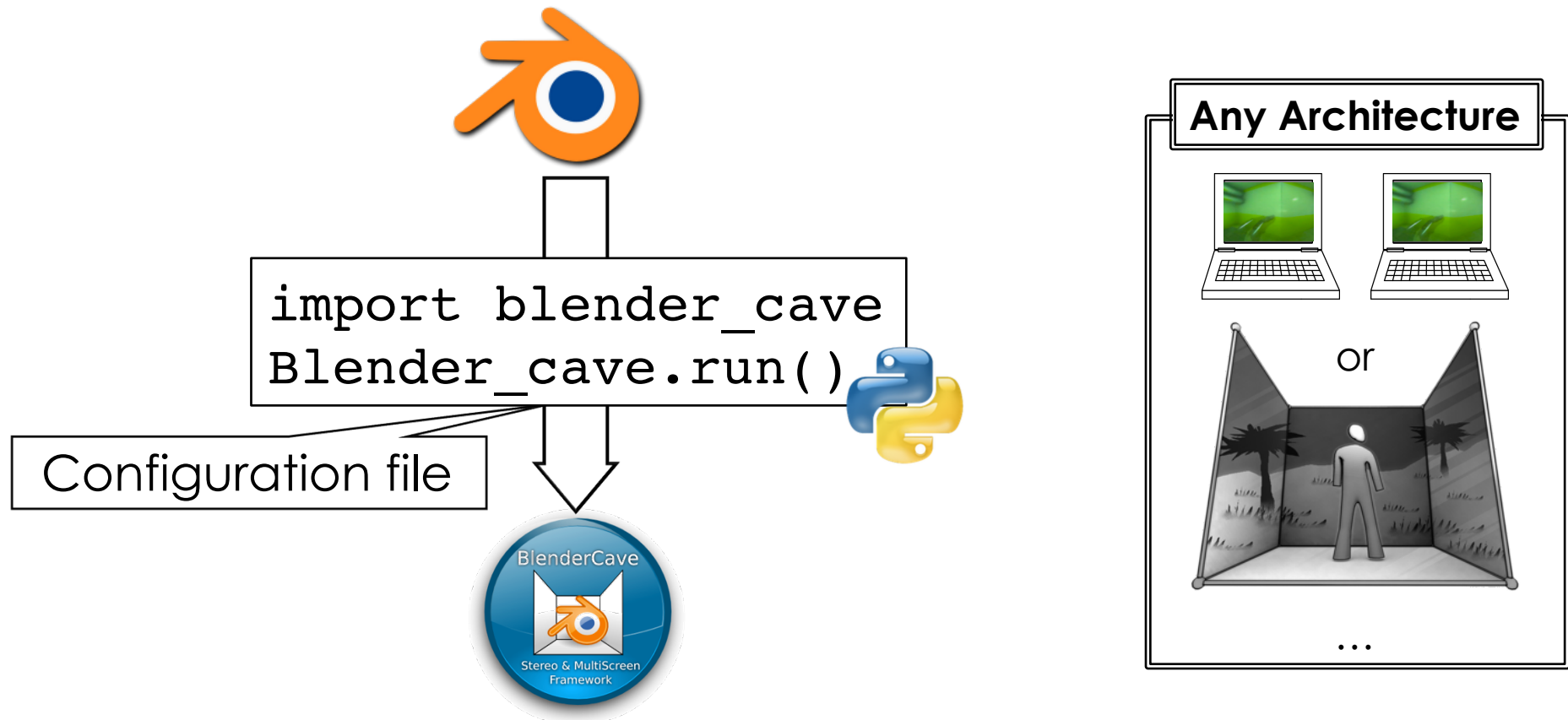


Complex scenes (Features and Sound)

Portability

3. Usage – Scene Creation

1. Create / Download a Blender Scene (Content & Logic)
2. Setup / Activate BlenderCAVE rendering



3. Usage – Scene Creation

3. Setup / Activate BlenderCAVE Sound Rendering Engine

```
import blender_cave
OSC = blender_cave.getOSC()

## Access / set OSC User
user_OSC = OSC.getUser(blender_cave.getUserByName('Binaural1'))
user_OSC.volume('%10')

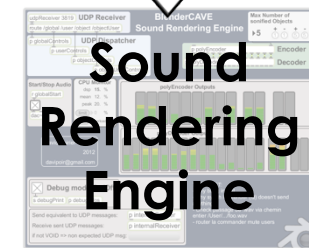
## Access / set OSC Object
object_OSC = OSC.getObject(scene.objects['Target'])
object_OSC.sound('micro2')

## Access / set OSC ObjectUser Linker
linker = OSC.getObjectUser(object_OSC, user_OSC)
linker.mute(False)
```



UDP

```
/user 2 volume %10
/object 1 sound micro2
/objectUser 1 2 mute 0
```



Thanks

My @ - davipoir@limsi.fr

BlenderCAVE mail List - Blender-cave@limsi.fr

- BlenderCAVE **sources**, **tutorial** and associated **Max/MSP Sound Rendering Engine** are available at:



<http://blendercave.limsi.fr>

- Next step: BlenderCAVE integration into Blender official trunk

BETA TESTERS ARE WELCOME!

BLENDERCAVE 2013

