

Introduction to the Live Python API

Juan Pedro Bolívar Puente

Ableton AG

17 May 2014

About me

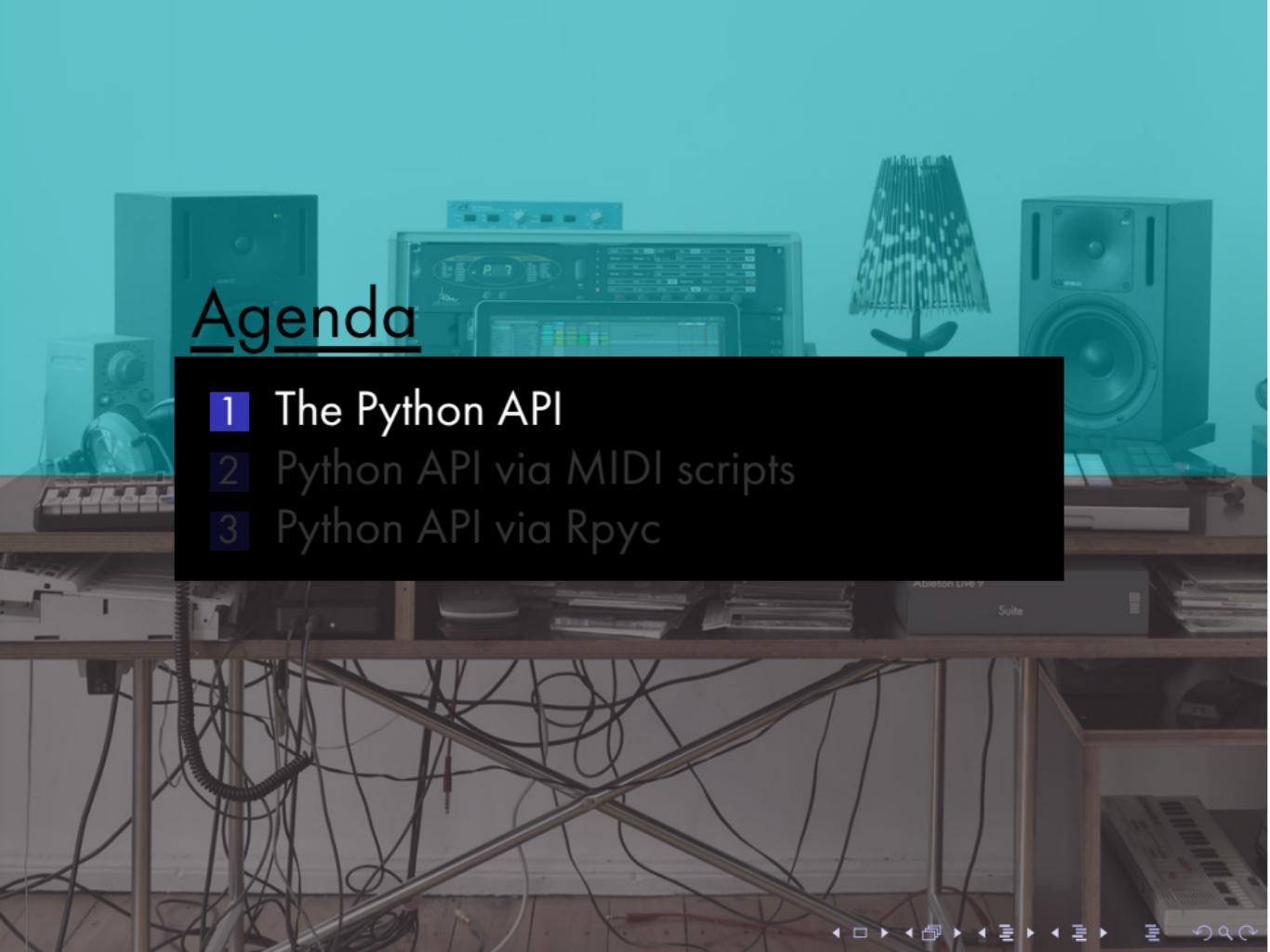
- 
- 1988 The beginning
 - 2005 GNU developer
 - 2011 Ableton Push developer

Disclaimer

The Python API is not supported by Ableton,
use this information at your own risk

Agenda

- 1 The Python API
- 2 Python API via MIDI scripts
- 3 Python API via Rpyc







OSC_Data_Monitor

IP address: 192.168.2.100

NOW LISTENING TO OSC PORTS:

- 9001

click on item to stop listening

LIST WITH COMMON OSC PORTS

- 3333
- 8000
- 8338
- 9001

click on item to start listening

ADD PORT

VIEW UPDATED VALUES

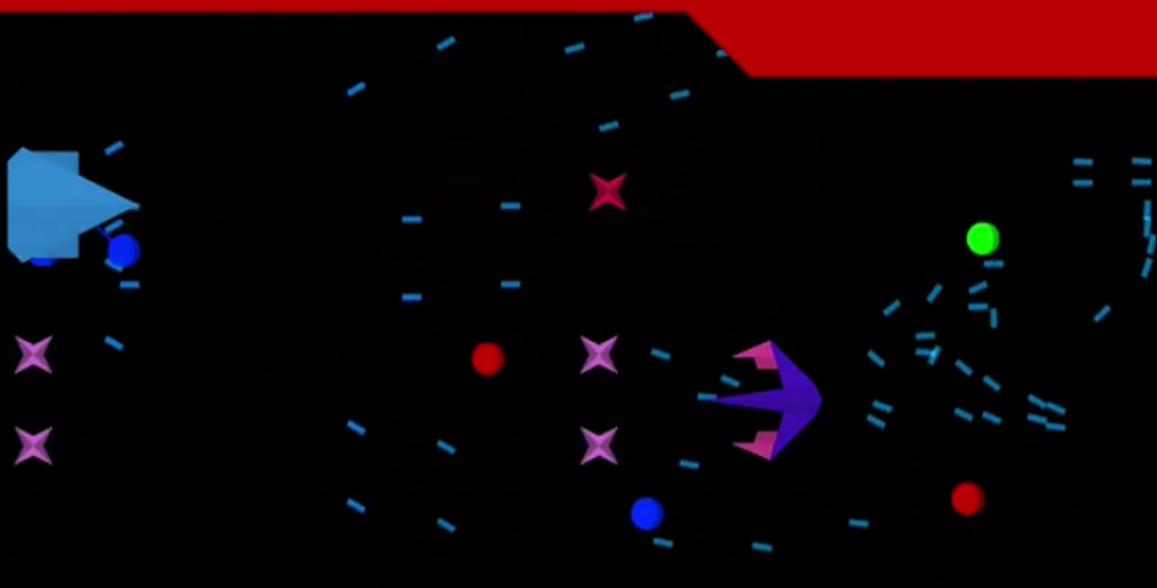
```
[9001] /live/name/track 3 (int) 4-Audio (string) 0 (int)
[9001] /live/name/track 4 (int) 5-Audio (string) 0 (int)
[9001] /live/name/trackblock 1-MIDI (string) 2-MIDI (string) 3-MIDI (string) 4-Audio (string)
[9001] /live/name/track 0 (int) 1-MIDI (string) 1 (int)
[9001] /live/name/clip 0 (int) 0 (int) (string) 8912743 (int)
[9001] /live/name/clip 0 (int) 2 (int) asdasd (string) 8912743 (int)
[9001] /live/name/track 1 (int) 2-MIDI (string) 1 (int)
[9001] /live/name/clip 1 (int) 0 (int) (string) 8912743 (int)
[9001] /live/name/clip 1 (int) 2 (int) asdasd (string) 8912743 (int)
[9001] /live/name/track 2 (int) 3-MIDI (string) 1 (int)
[9001] /live/name/clip 2 (int) 0 (int) (string) 3947580 (int)
[9001] /live/name/clip 2 (int) 1 (int) (string) 3947580 (int)
[9001] /live/name/clip 2 (int) 2 (int) BASS (string) 3947580 (int)
[9001] /live/name/track 3 (int) 4-Audio (string) 0 (int)
[9001] /live/name/track 4 (int) 5-Audio (string) 0 (int)
[9001] /live/name/trackblock 1-MIDI (string) 2-MIDI (string) 3-MIDI (string) 4-Audio (string)
[9001] /live/device/param 0 (int) 0 (int) 0 (int) 1.000 (float) Device On (string)
[9001] /live/device/param 0 (int) 0 (int) 1 (int) 1.000 (float) LFO Type (string)
[9001] /live/device/param 0 (int) 0 (int) 2 (int) 0.732 (float) Amount (string)
[9001] /live/device/param 0 (int) 0 (int) 3 (int) 0.699 (float) Frequency (string)
[9001] /live/device/param 0 (int) 0 (int) 4 (int) 4.000 (float) Sync Rate (string)
[9001] /live/device/param 0 (int) 0 (int) 5 (int) 180.000 (float) Phase (string)
[9001] /live/device/param 0 (int) 0 (int) 6 (int) 0.000 (float) Spin (string)
[9001] /live/device/param 0 (int) 0 (int) 7 (int) 0.000 (float) Stereo Mode (string)
[9001] /live/device/param 0 (int) 0 (int) 8 (int) 0.000 (float) Offset (string)
[9001] /live/device/param 0 (int) 0 (int) 9 (int) 2.000 (float) Waveform (string)
[9001] /live/device/param 0 (int) 0 (int) 10 (int) 1.000 (float) Shape (string)
[9001] /live/device/param 0 (int) 0 (int) 11 (int) 0.500 (float) Width (Random) (string)
[9001] /live/device/param 0 (int) 0 (int) 12 (int) 1.000 (float) Invert (string)
[9001] /live/device/selected 0 (int) 0 (int)
[9001] /live/device/param 0 (int) 0 (int) 4 (int) 5.000 (float) Sync Rate (string)
[9001] /live/device/param 0 (int) 0 (int) 4 (int) 6.000 (float) Sync Rate (string)
```

RECEIVED OSC ADDRESSES:

/LIVE/TRACK/INFO

MONITOR ONLY OSC ADDRESSES:

◀ □ ▶ ▷ ▸ ▲ ▾ ▴ ▾ ▷ ▸ ▲ ▾



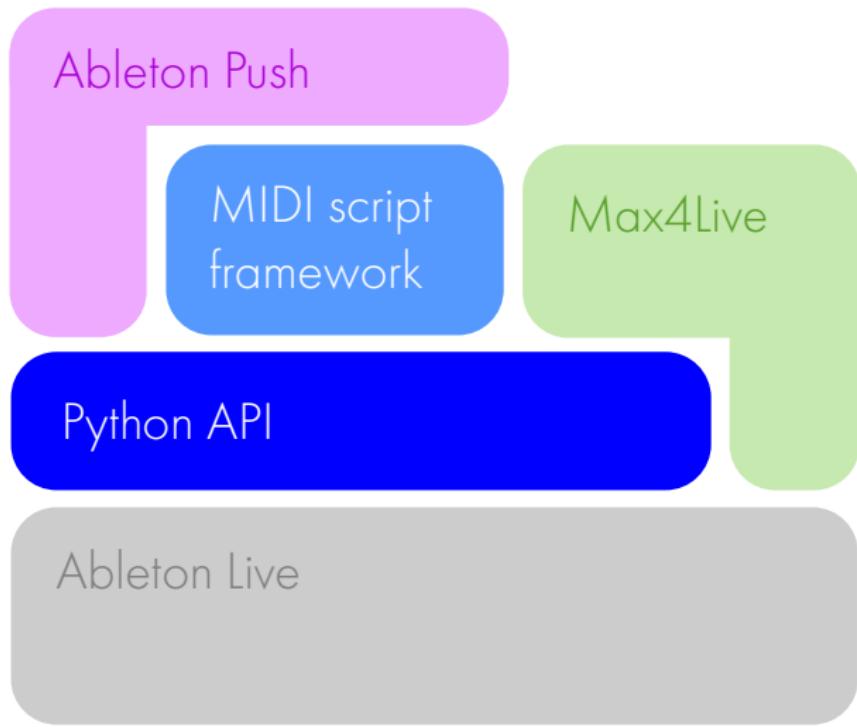
Untitled

EXT TAP 120.00 ||||| 4 / 4 ○● - 1 Bar ➞ 1. 1. 1 ► ■ ● + ⚡ + ○ NEW ↻

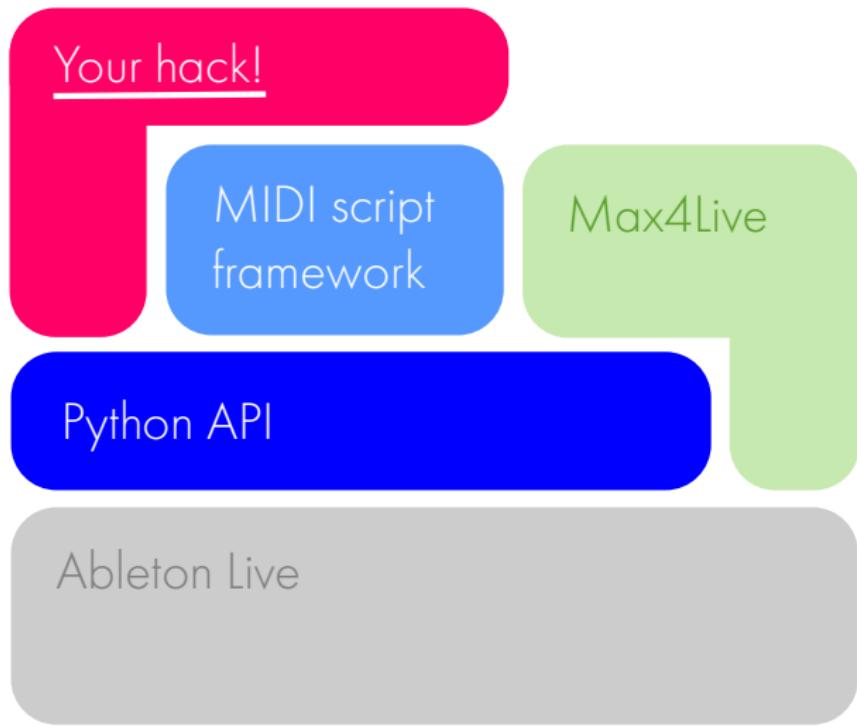


Drop Files and Devices Here

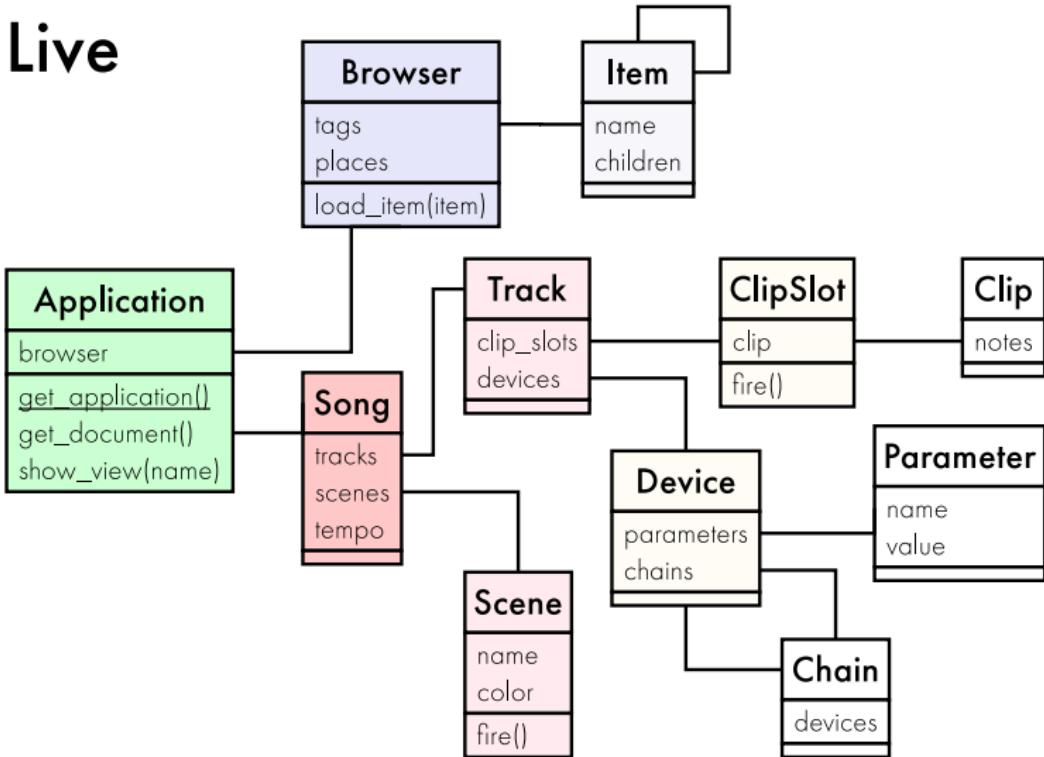
Architecture



Architecture



Live

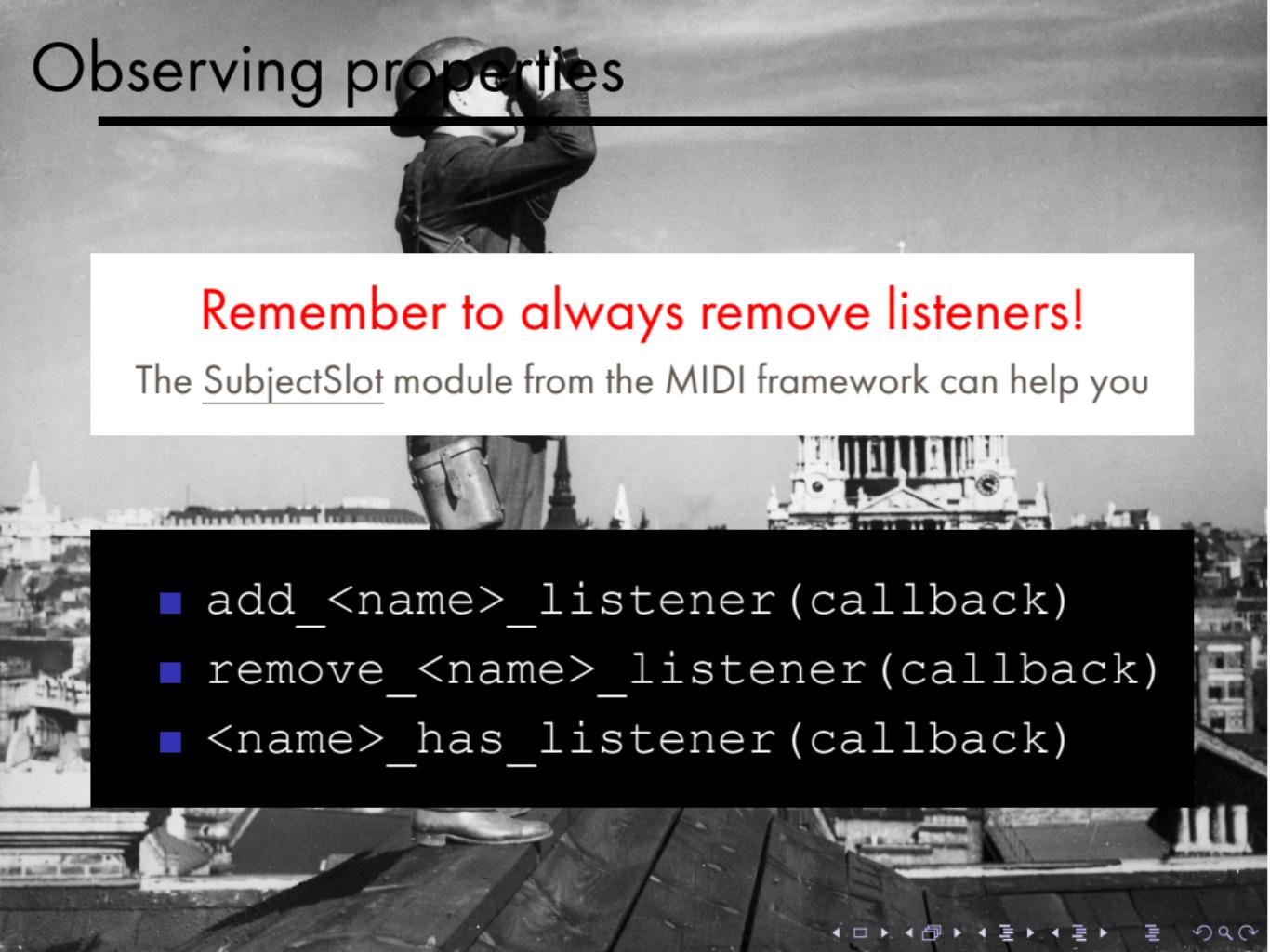


Observing properties

Most properties can be observed

- `add_<name>_listener(callback)`
- `remove_<name>_listener(callback)`
- `<name>_has_listener(callback)`

Observing properties



Remember to always remove listeners!

The SubjectSlot module from the MIDI framework can help you

- `add_<name>_listener(callback)`
- `remove_<name>_listener(callback)`
- `<name>_has_listener(callback)`

Observing properties



Oh, and quite often you can not change properties inside a listener

Use the Task stuff to postpone the modifications

- `add_<name>_listener(callback)`
- `remove_<name>_listener(callback)`
- `<name>_has_listener(callback)`

Agenda

- 1 The Python API
- 2 Python API via MIDI scripts
- 3 Python API via Rpyc



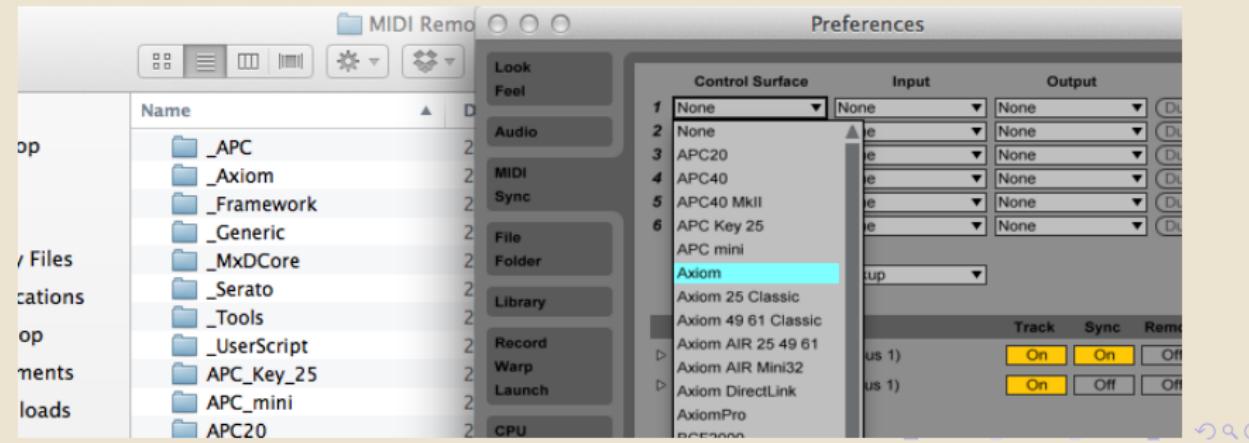
Accessing MIDI remote scripts

Windows

```
C:\Program Files\Ableton\Live 9.1\Resources\  
MIDI Remote Scripts
```

Mac

```
/Applications/Ableton Live 9.1.app/Contents/  
App-Resources/MIDI Remote Scripts
```



Anatomy of a MIDI Remote Script

MyScript/__init__.py

```
def create_instance(c_instance):
    from MyScript import MyScript
    return MyScript(c_instance)
```

MyScript/MyScript.py

```
from __future__ import with_statement
from _Framework import ControlSurface

class MyScript(ControlSurface.ControlSurface):
    def __init__(self, c_instance):
        super(MyScript, self).__init__(c_instance)
        with self.component_guard():
            pass # Here we will initialize...
```

Important concepts

ControlSurface

Base class of the object representing a MIDI device

But you can also load it without a MIDI port

ControlElement

An abstraction over a MIDI widget

A [ButtonElement](#), [EncoderElement](#), [ButtonMatrixElement](#)...

ControlSurfaceComponent

A reusable piece of behaviour that usually relates
control elements to the Live API

Advanced concepts

SubjectSlot

Manages a connection from a listener to a property of a subject

Layer

Assignment between control elements and components

Task

Allow delaying, repeating and sequencing calls

Mode

Help in constructing state machines and modal behaviour

Debugging

```
from _Framework.Dependency import depends
from _Framework.Util import const, nop

@depends(log_message=const(nop))
def some_function(log_message=None):
    log_message("I like this number", 42)
```

Windows

```
C:\Documents and Settings\<user>\Application Data\
Ableton\Live 9.1\Preferences\Log.txt
```

Mac

```
/Users/<user>/Library/Preferences/Ableton/
Live 9.1/Log.txt
```

Example time!

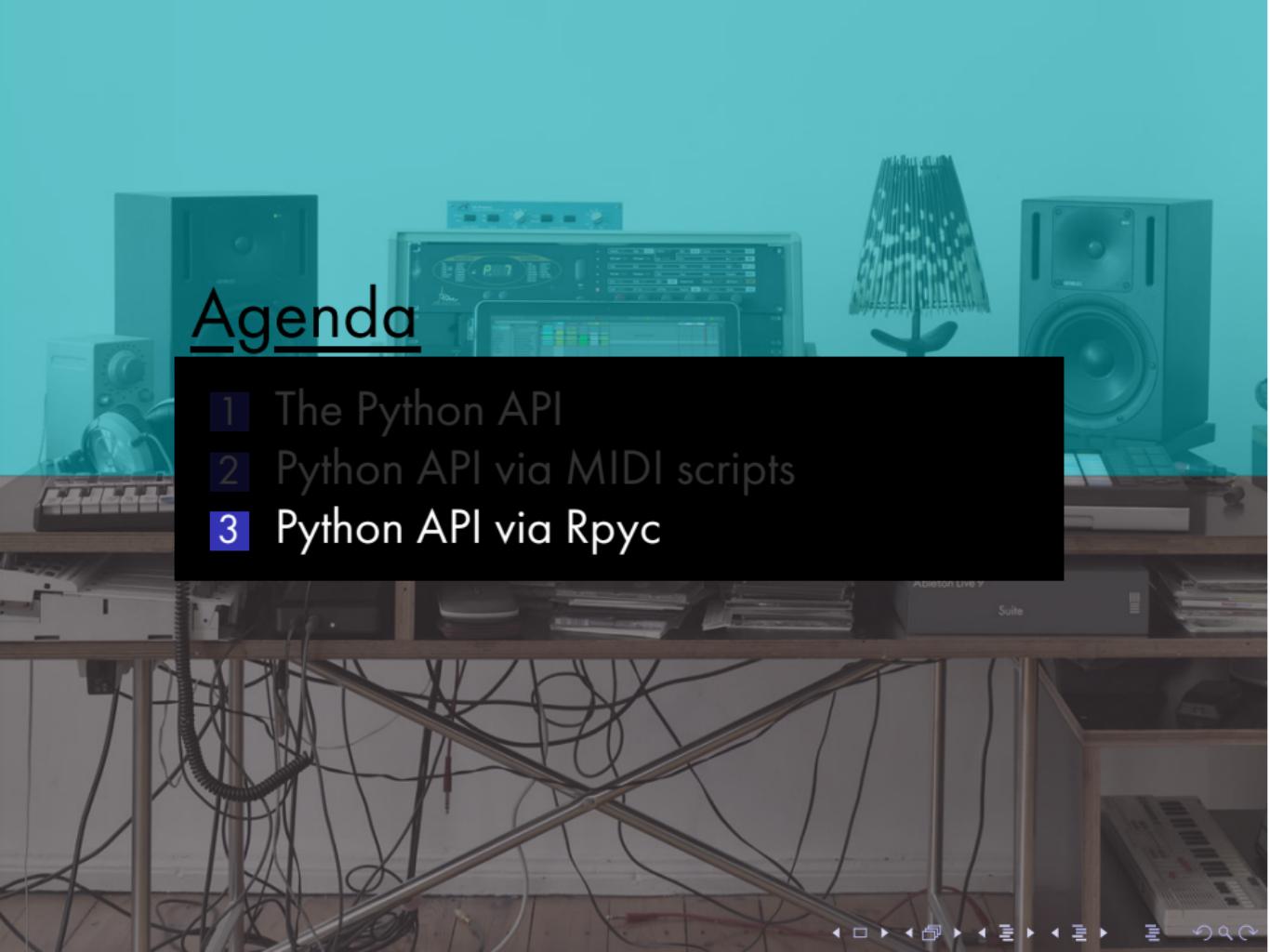
Hack_TrackColors is a script that contains...

- ClipToTrackColorsComponent
makes clip colors follow the track color
- TrackColorChooserComponent
connects MIDI buttons to choose the selected track's color
- Maps the Launchpad matrix to choose the track color

Hack_WebTrackColors is a ClojureScript web app
that uses the WebMidi API to talk to the former

Agenda

- 1 The Python API
- 2 Python API via MIDI scripts
- 3 Python API via Rpyc



What is Rpyc

Rpyc can connect to a remote Python process
and use it's objects as if they where local



Connecting to Live

- 1 Install and load the RpycHost remote script
- 2 Enjoy!

```
import rpyc
conn = rpyc.connect('localhost', 17744)
live = conn.root.Live.Application.get_application()

song = live.get_document()
print "Current tempo", song.tempo
song.start_playing()
```

A note on listeners...

Rpyc connections are bidirectional

i.e. connecting listeners just works!

but...

- They are only called when we talk to the server,
`use conn.poll()`
- You can use `select()` on the connection or a
separate thread handle the connection and other
inputs concurrently

A scene from the movie Pulp Fiction. Jules Winnfield (Bruce Willis) is on the right, wearing a dark suit and tie, looking intensely at the camera with a serious expression. Vincent Vega (John Travolta) is on the left, wearing a red and white plaid robe over a t-shirt, looking towards Jules. They are in a hallway with patterned wallpaper.

Be clean!

Failing to remove the client listeners' and close
connections can leave Live in a broken state

Example time!

Hack_LiveCurses is a Python program that...

- Controls Live's transport via a text ncurses interface
- Interleaves updates from user input and Live with select()

Conclusions

- The Python API is the **most powerful but experimental** way to hack into Live
- MIDI can be used creatively as general inter-app communication
- Rpyc and **LiveOSC** let you directly talk to Live from any system



Further readings

-  **The Live Object Model**
Max4Live documentation
http://cycling74.com/docs/max5/refpages/m4l-ref/m4l_live_object_model.html

-  **Live 9 MIDI remote script reverse engineering**
Julien Bayle
<http://julienbayle.net/ableton-live-9-midi-remote-scripts/>

-  **A introduction to the _Framework classes**
Hanz Petrov
<http://remotescripts.blogspot.de/>

Third party hacks

LiveOSC hack

Liine

<http://livecontrol.q3f.org/ableton-liveapi>

PyLive hack on top of LiveOSC

@ideoforms

<https://github.com/ideoforms/pylive>

LiveTelnet, control from a Telnet session

The LiveAPI hack project

<https://www.assembla.com/spaces/live-api/>

Questions?

Thanks for listening!

