# Cloud Computing Architecture

**AMCP**

**Performer**

**Participants**

**Developer**

### Mobile Device

Music Player **1**

AirPlay Client

### Mobile Device

Web Browser **3**

### TouchOSC **2**

Custom Layouts

### Laptop

OPC Visualizer **4**

3D Model

---

### Raspberry Pi

AirPlay Server (shairport)

Music Analyzer

Shared Memory

ALSA

UDP

FakeDNS Server

Web Server (nginx)

Fake web sites

TCP

USB Serial

### Teensy 3.0 **5**

USB Driver

Keyframe Interpolator

Dithering & Gamma

OctoWS2811

#### Python Process

OSC Server (pyliblo)

Effect Modules **7**

python-mplayer

TPM **8**

Light Mixer

OPC Client

Open Pixel Control server **6**

Cloud Sound FX

mplayer **9**
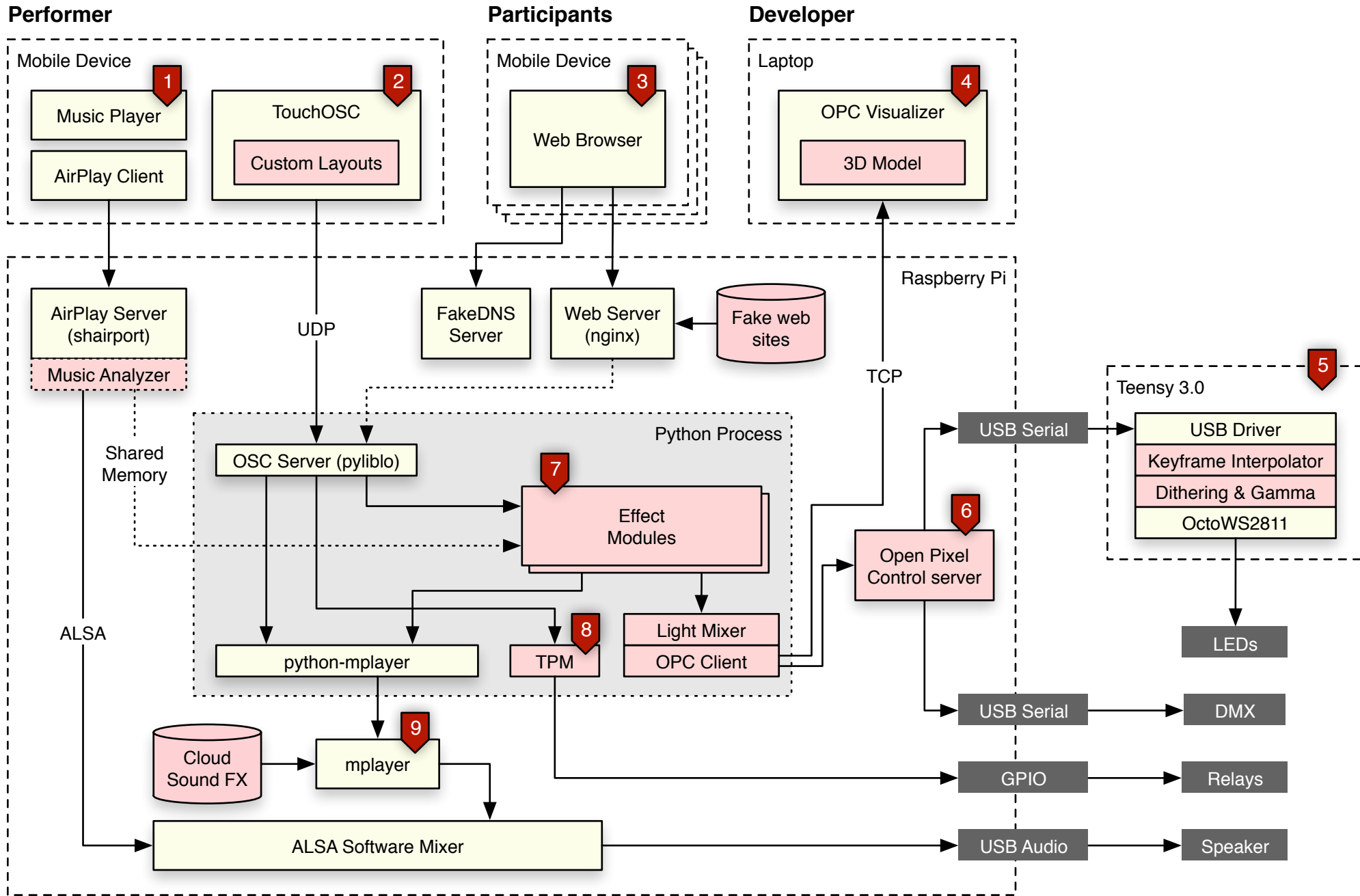
ALSA Software Mixer

LEDs

USB Serial → DMX

GPIO → Relays

USB Audio → Speaker

# Cloud Computing Architecture Notes

**1** Music from any AirPlay-compatible device can stream to The Cloud, optionally mixing with other sound effects. For extra credit, we can modify the shairplay server to add some basic signal processing, allowing light effects which react to music in real-time.

**2** Performers control light sequences, sound effects, and precipitation using a multi-touch UI developed using TouchOSC.

**3** Participants who connect to The Cloud will be greeted by fake web site content for any URL they visit. For extra credit, some of these fake web sites might interact with other cloud functionality.

**4** Using the Open Pixel Control LED protocol, developers can preview effects on a software simulator without needing any physical cloud infrastructure.

**5** The Teensy MCU handles performance-critical interpolation and dithering tasks, so that LED animation remains smooth even if the Python process isn't especially speedy.

**6** Our OPC server speaks to both the Teensy and the DMX controller, allowing the Python process and the visualizer to process LED effects and DMX effects with the same code path.

**7** Multiple light & sound effects may run simultaneously, allowing performers to smoothly blend and transition between effects.

**8** The Trusted Precipitation Module controls water valves and our pump for safe and effective moisture delivery.

**9** Sound effects play with mplayer, a light-weight media player that's easy to command from Python. It also supports simple audio filters and volume control.

## Goals

- Easy to hack on, high-level code
- Easy to develop light effects in simulation
- High quality light and sound
- Creative expression for devs and performers
- Keep the performance-critical code small

## Legend

New Software

Existing Software

Hardware Interface

·········· Extra Credit ···········

Last modified: Wed Jul 10 2013