

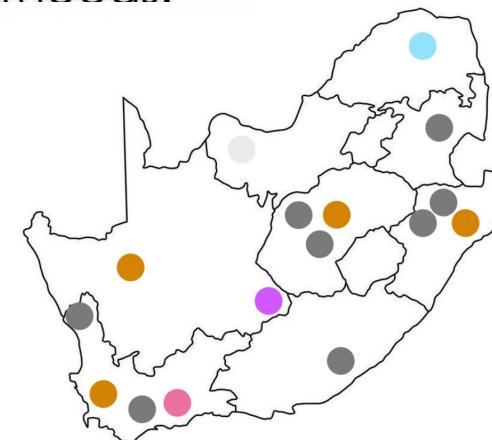
# *Technology in Religious Community Broadcasting*

Using Liquidsoap to facilitate community based broadcasts

Umar

# Introduction

- My background and experience
- Brief South African historic context and the formation of ethnic communities
- Post 1994 on the backdrop of racial segregation, ethic and religious freedoms facilitated the amalgamation of communities. This allowed the establishment of various religious centers in traditional “white” neighborhoods.

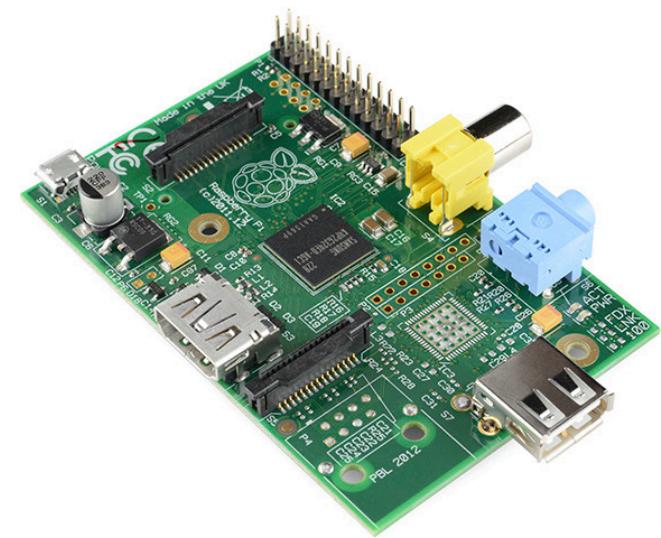


# Challenge

- The Mosque (place of worship) is central to a Muslim and the heart of the community
- Mosques serve the communities as places of worship, education, social welfare, building unity, youth development, etc.
- 5 Daily prayers (Salaah) are performed at Mosque @ Sunrise, Midday, Afternoon, Sunset, early Night
- Each prayer is preceded by the Arabic Athaan (Call to Prayer) as a reminder that the time for prayer has set in
- Traditionally called out from the Minaret (a high place), later with technology, given over loud speakers from towers of the Mosque

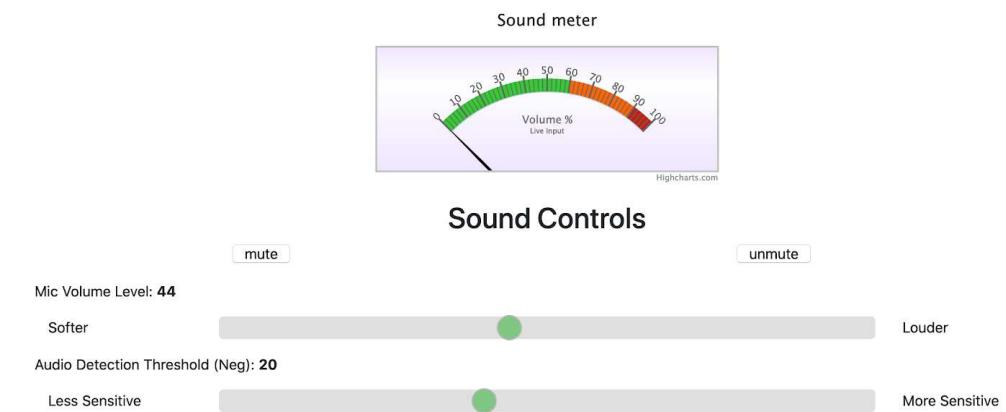
# Solution

- The creation of Livemasjid.com - a free audio streaming service for Mosques.
- Livemasjid.com platform was started just over 10 years ago (+-2012)
- Driven by the launch of the Raspberry Pi 1. Affordable and extremely versatile.
- Challenge managing data consumption and a dynamic (unknown start and stop) stream.

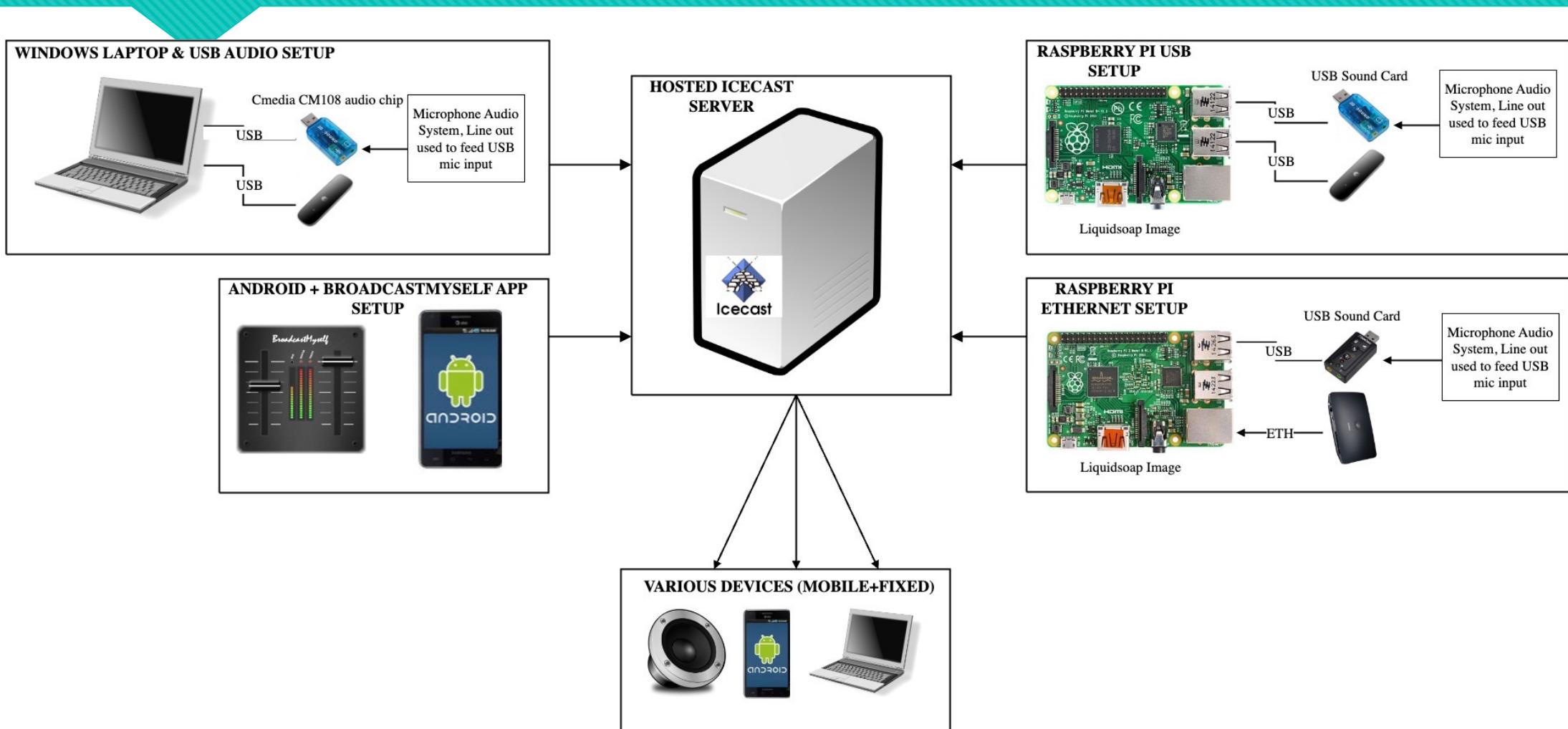


# Solution – Liquid Soap

- Automatic broadcast was facilitated through use of Liquidsoap Track Processing function:
- `stream = strip_blank(max_blank=90.,min_noise=0.5,threshold=-20.,start_blank=true, stream)`



# High Level Architecture



# Solution – Liquid Soap

- 2<sup>nd</sup> challenge was around the use of multiple separate audio systems.
- A mixer and amp was used specifically for Athaan, while a separate mixer and amp is used for sound within the Masjid.
- One solution might be adding a third ‘mini’ mixer to combine these channels however to save costs the extra USB ports of the Raspberry pi can be used to connect additional USB sound cards.
- Mixing is then facilitated within Liquidsoap!
- Simple as:

```
stream = add([mic1,mic2])
```

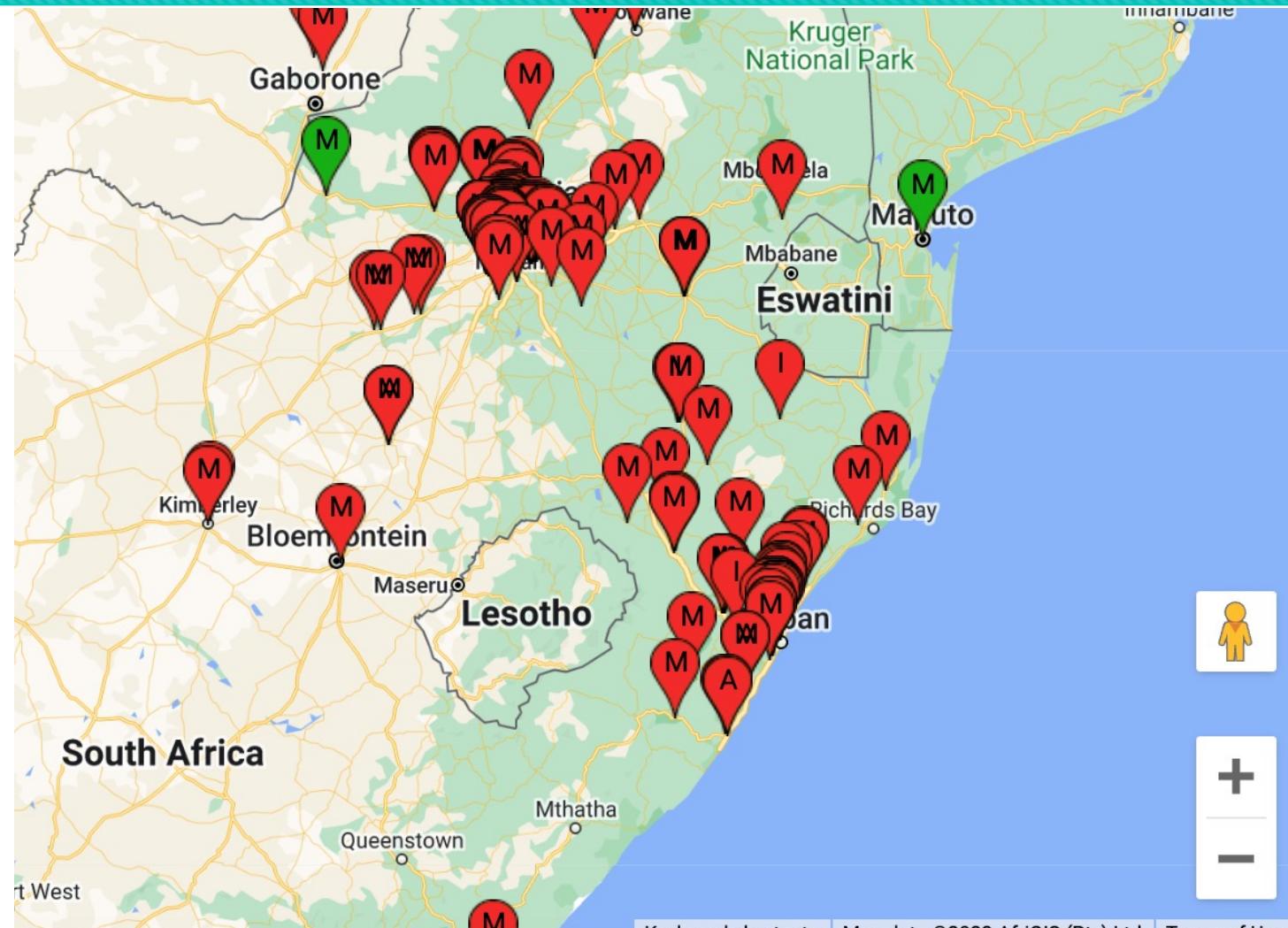
# Masjid Cube

- The missing link was enabling users easy access to audio broadcast from a specific Mosque.
- The Masjid Cube is a ESP32 based, WiFi receiver built specifically to allow live broadcast.



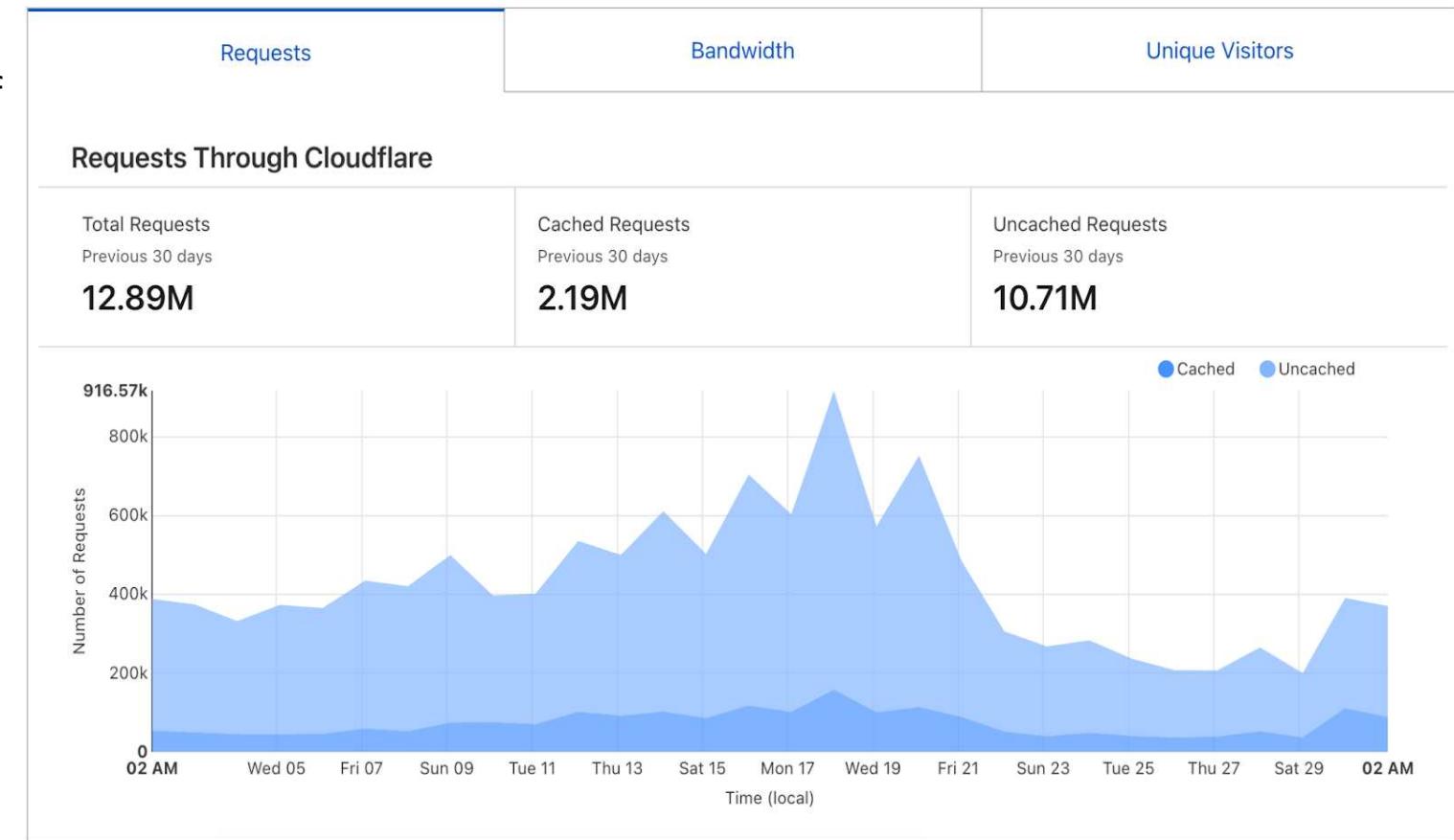
# The Platform

- Has grown to serve over 250 mosques, primarily in South Africa
- Facilitates integration within our diverse communities
- Archiving of sermons within S3 for later access



# The Platform

- Monthly stats show millions of page requests for hundreds of thousand of unique people across the world.
- Strengthens bond with the Mosque within Muslim homes



# Links and resources

- Open source git repository: <https://bitbucket.org/mitpeople>
- Website: [Livemasjid.com](http://Livemasjid.com)
- [Masjidcube.co.za](http://Masjidcube.co.za) (Instagram @Masjidcube)
- LinkedIn: [linkedin.com/in/umar-dockrat](https://linkedin.com/in/umar-dockrat)

**Thank You!**

```
#!/usr/bin/liquidsoap -v --debug
mountname = getenv("mount")
streamdesc = getenv("streamdesc")
streamurl = getenv("streamurl")
streamhost = getenv("streamhost")

# Input audio is fed in from the sound card using either direct ALSA, PulseAudio, ALSA indirectly or ALSA snoop
radio_stream = (input.alsa(device="stream",bufferize=true):source(1,0,0))

# The strip_blank function will disable the stream when 90 seconds of silence is detected
# The stream will only be re-enabled when 0.5seconds (min_noise) of noise above (threshold) is detected
# threshold values more positive means you have to speak louder to active. more negaive means it will detect softer sounds
radio_stream = strip_blank(max_blank=90.,min_noise=0.5,threshold=-20.,start_blank=true, radio_stream)
```

```
# Stream it out to icecast
output.icecast(
%opus(bitrate=16,channels=1,samplerate=48000),
host = "#{streamhost}",
port = 8000,
password = "#{strempassword}",
mount = "#{mountname}",
radio_stream,
fallible=true,
format="audio/ogg",
description= "#{streamdesc}",
url="#{streamurl}",
dumpfile="/home/pi/ram/dump.ogg",
on_start={system("/home/pi/scripts/prestream.sh")},
on_stop={system("/home/pi/scripts/poststream.sh")})
```

```
#Dump live recordings to a file
time_stamp = '%Y%m%d_%H%M%S'
output.file(
%mp3.abr(bitrate=48,max_bitrate=128, samplerate=44100, mono=true),
"/home/pi/arecordings/live_stream_#{time_stamp}.mp3",
radio_stream,fallible=true)
```



From left to right (Also Best to Worst):

1. Sound Blaster 2. UGreen 3. Cable Creation 4. Sabrent 5. China Clone 6. Adafruit 7. Black and 8. Blue