

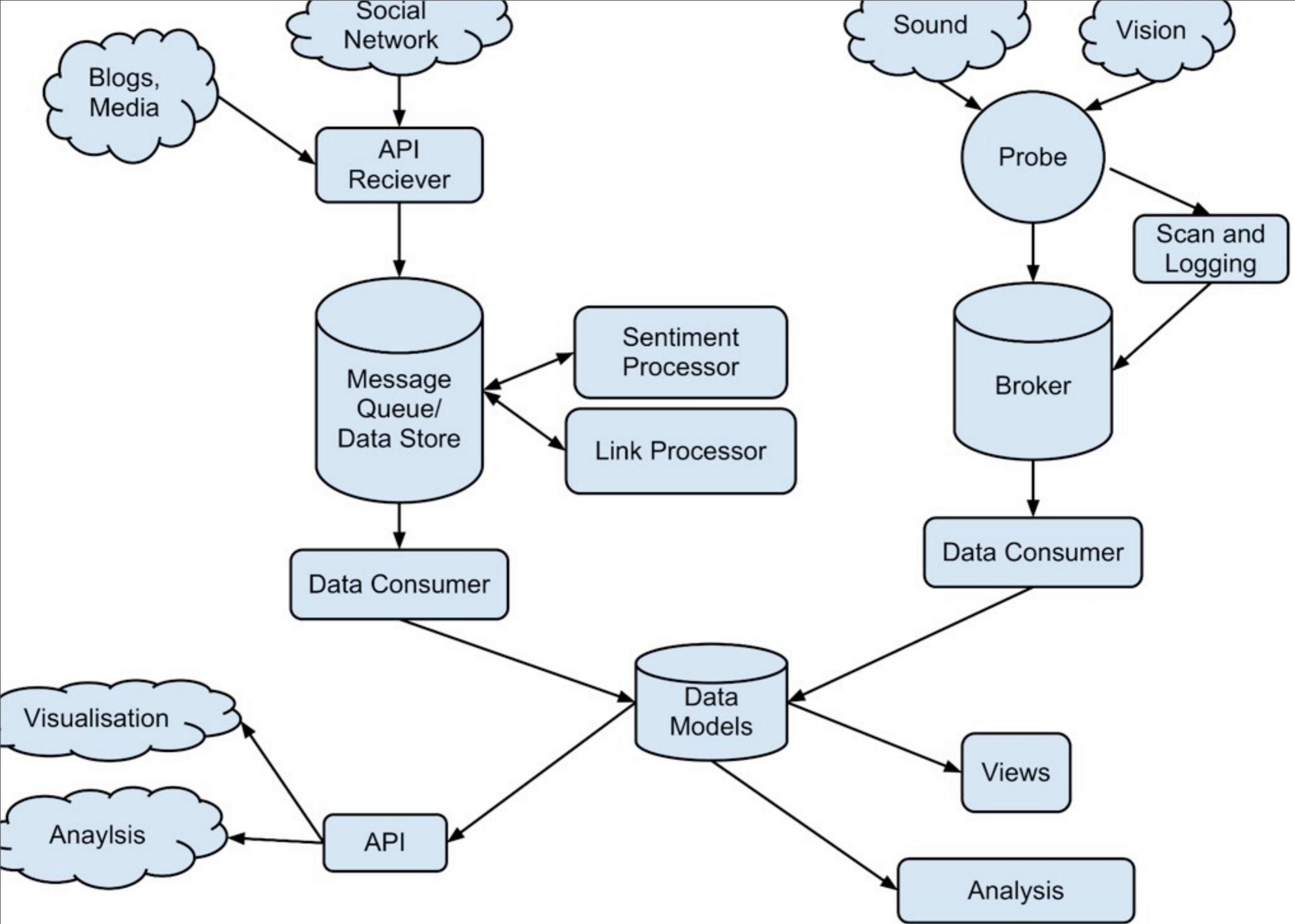


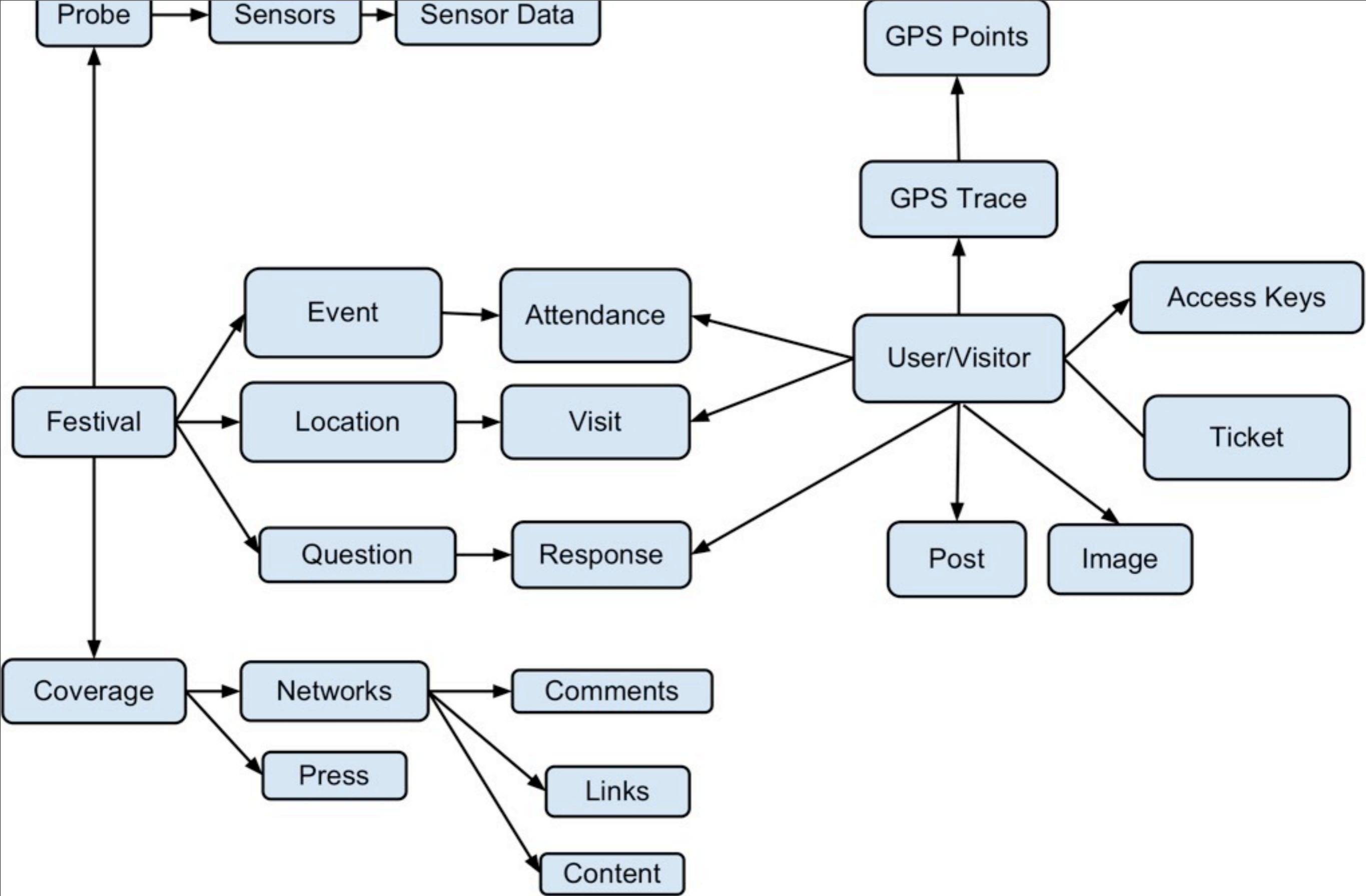
Qualia-Tech

| Christopher Hunt

The last few months...

- Qualia-Smile for Jazz
- Qualia-Web-Engine and Qualia-App for Science
- Qualia-Probes for Music





Key Goals

- Where possible, Open Source
- Reusable
- Maintainable
- Avoiding Technical Debts

```
# Allocate the temporary images
gray = cv.CreateImage((image.width, image.height), 8, CV_8U)
smallImage = cv.CreateImage((cv.Round(image.width / 2), cv.Round(image.height / 2)), 8, CV_8U)

# Convert color input image to grayscale
cv.CvtColor(image, gray, cv.CV_BGR2GRAY)

# Scale input image for faster processing
cv.Resize(gray, smallImage, cv.CV_INTER_LINEAR)

# Equalize the histogram
cv.EqualizeHist(smallImage, smallImage)

# Detect the faces
faces = cv.HaarDetectObjects(smallImage, faceCascade, cv.CV_32F,
                             haar_scale, min_neighbors, haar_flags, min_size)

# If faces are found
if faces:

    #print faces

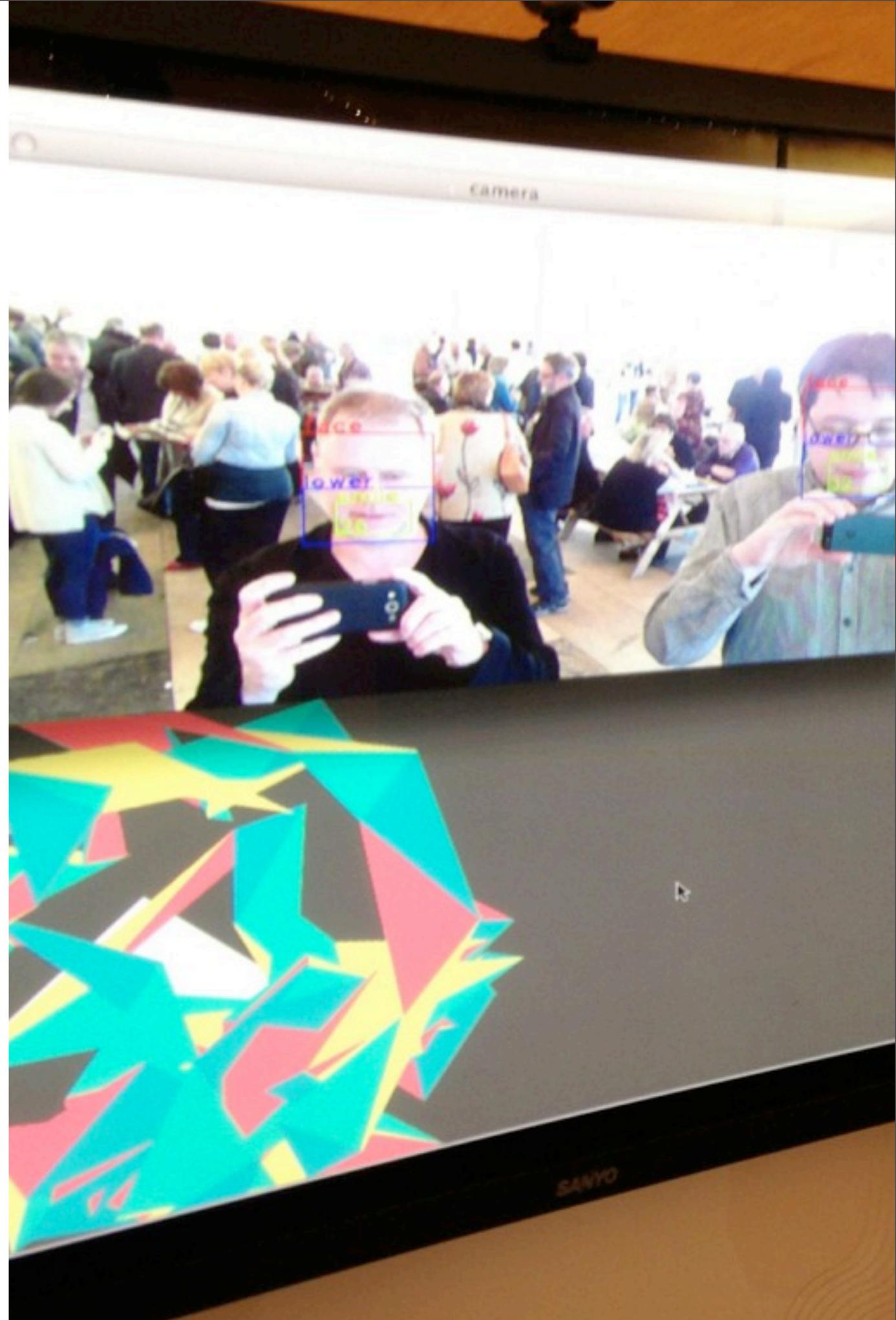
    for ((x, y, w, h), n) in faces:
        # the input to cv.HaarDetectObjects was resized,
        # bounding box of each face and convert it to two points
        #print "face"
        pt1 = (int(x * image_scale), int(y * image_scale))
        pt2 = (int((x + w) * image_scale), int((y + h) * image_scale))
        cv.Rectangle(image, pt1, pt2, facecolor, 1, 8)
        cv.PutText(image, "face", pt1, font, facecolor, 1)
        face_region = cv.GetSubRect(image, (x, int(y * image_scale),
                                             int((x + w) * image_scale),
                                             int((y + h) * image_scale)))

        #split face
```



Qualia-Smile

- OpenCV Smile Sensor
- Connected to the Triangles Visualisation and Logging Software
- Installation piece and a demonstration of technology



Qualia-Web-Engine

- Modelling Data
- Providing access and restrictions
- Data Processing and Display
- <http://qualia.org.uk/admin/>

Auth	
Groups	 Add
Users	 Add
Core	
Attendances	 Add
Event responses	 Add
Events	 Add
Festivals	 Add
GPS Hotspots	 Add
GPS Points	 Add
Images	 Add
Venues	 Add
Visitors	 Add
Musicmetric	
Tweet sentiments	 Add
Probes	
Audio answers	 Add
Face detections	 Add
Probe messages	 Add
Probes	 Add
Questions	 Add
Smile detections	 Add
Registration	
Registration profiles	 Add
Sites	

Working with others - the API

- Standard REST API
- Provides connection to the Web Engine for the App, Probes and Log Parser for Smile
- <http://qualia.org.uk/api/doc/>

Response Body

```
{
  "display_on_probe": true,
  "end": "2013-05-01T22:30:00",
  "id": 238,
  "image_url": "http://cheltenhamfestivals-...",
  "info": "As a Swedish songstress now res...",
  "name": "EMILIA MARTENSSON",
  "price": "£60",
  "quick_find": "J02",
  "resource_uri": "/api/v1/event/238/",
  "site_url": "http://www.cheltenhamfestiva...",
  "start": "2013-05-01T18:00:00",
  "venue": {
    "about": "The Daffodil began life as Ch...",
    "accessibility": "The Daffodil is wheel...",
    "address": "The Daffodil",
    "directions": "The Daffodil is around a...",
    "id": 34,
    "image_url": "http://cheltenhamfestiva...",
    "lat": "51.8927000000",
    "long": "-2.0799600000",
  }
}
```

Response Code

200

Response Headers

Date: Sun, 11 Aug 2013 10:30:57 GMT

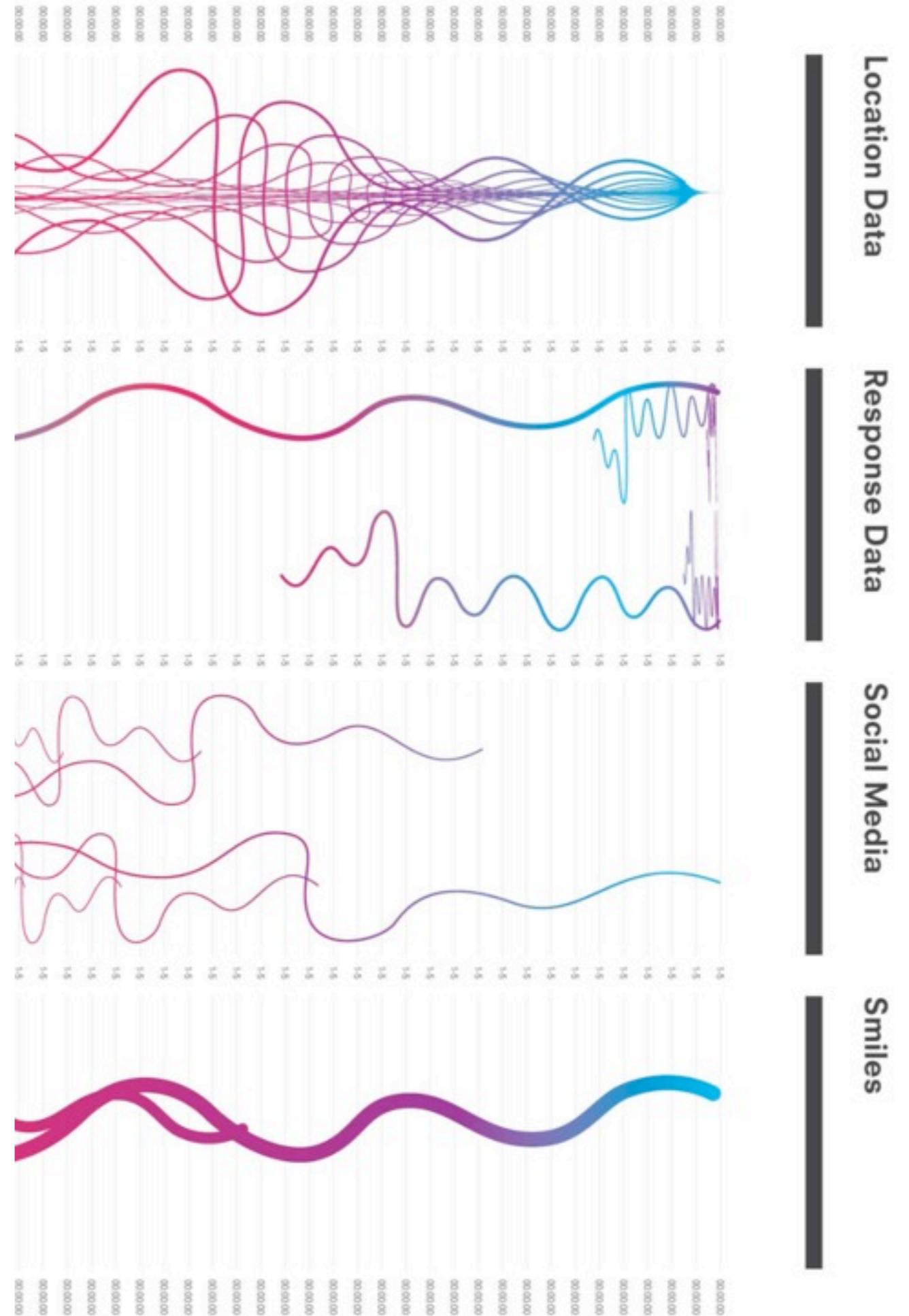
Qualia-Probe

- Kiosk Installations
- Information and interaction point for the events
- Makes use of touchscreen, camera and microphone
- Built with web technologies



Next few months...

- Full development of Dashboard interface
- Working with Elixel on the full version of the App
- Adding SMS and Voice Calling
- Qualia-Realtime
- Qualia-Sentiment



Qualia-Future

- Open Sourcing
- The business model - Product or People?
- The “responsive” event
- Should it be invisible?
Ubiquitous?

