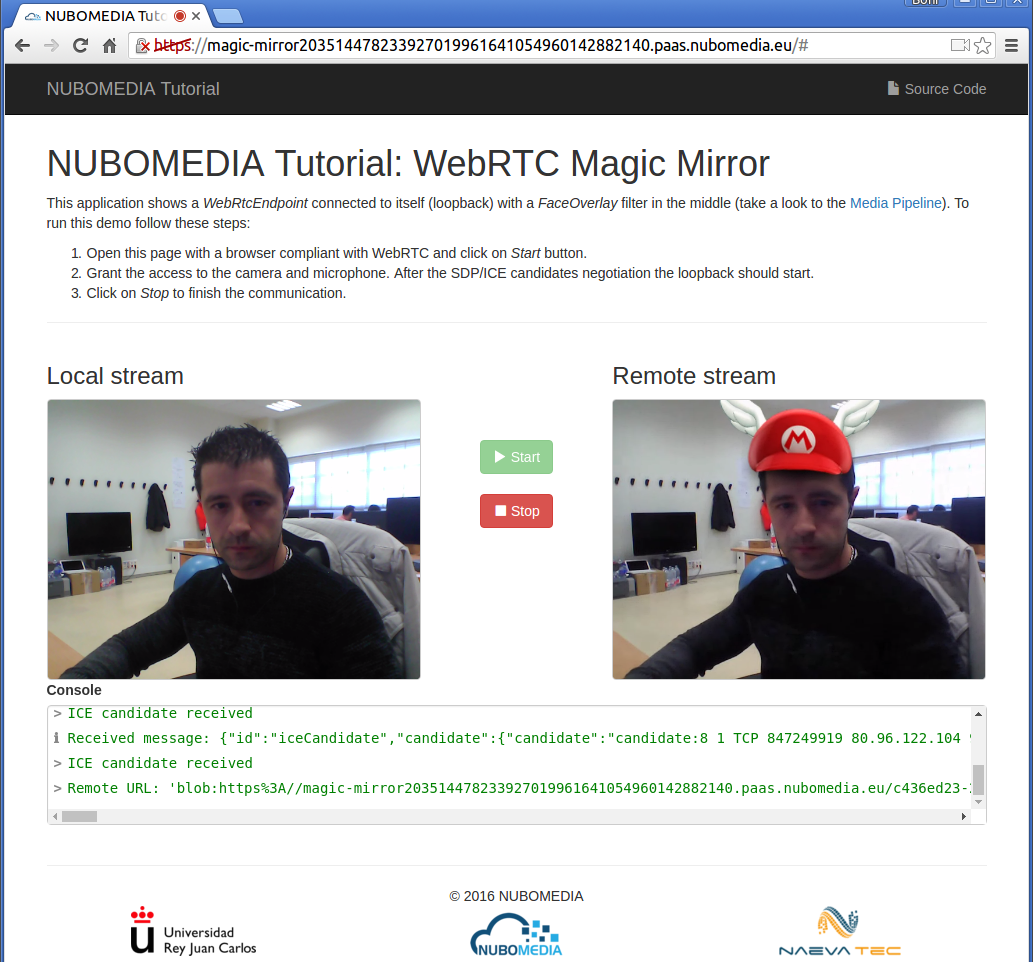
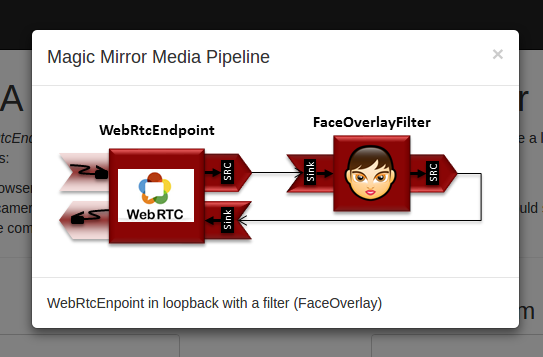
**Problema.**

Actualmente estamos usando el filtro ***FaceOverlayFilter*** para pintar imágenes sobre impresionadas en el stream de video suministrado por el KMS.





Uso:

FaceOverlayFilter faceOverlayFilter = new FaceOverlayFilter.Builder( roomManager.getPipeline(pid)).build();

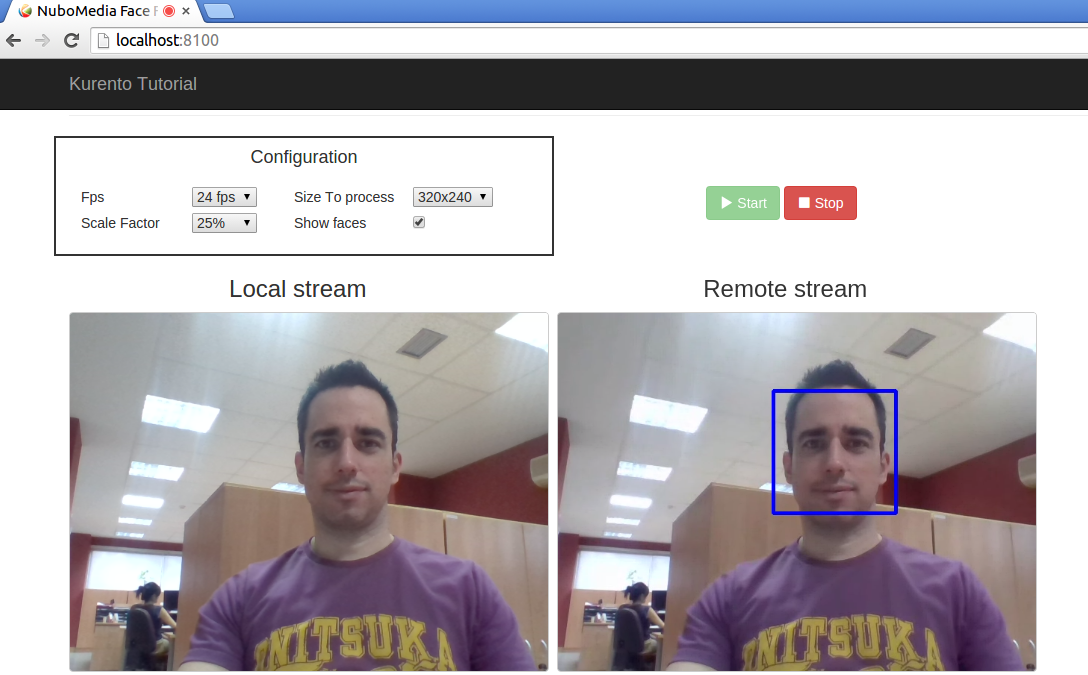
faceOverlayFilter.setOverlayedImage(this.hatUrl, this.offsetXPercent, this.offsetYPercent, this.widthPercent, this.heightPercent);

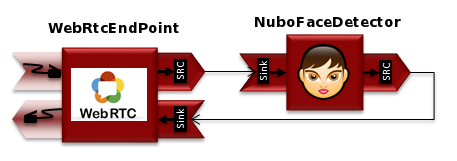
roomManager.addMediaElement( pid, faceOverlayFilter );

**Propuesta.**

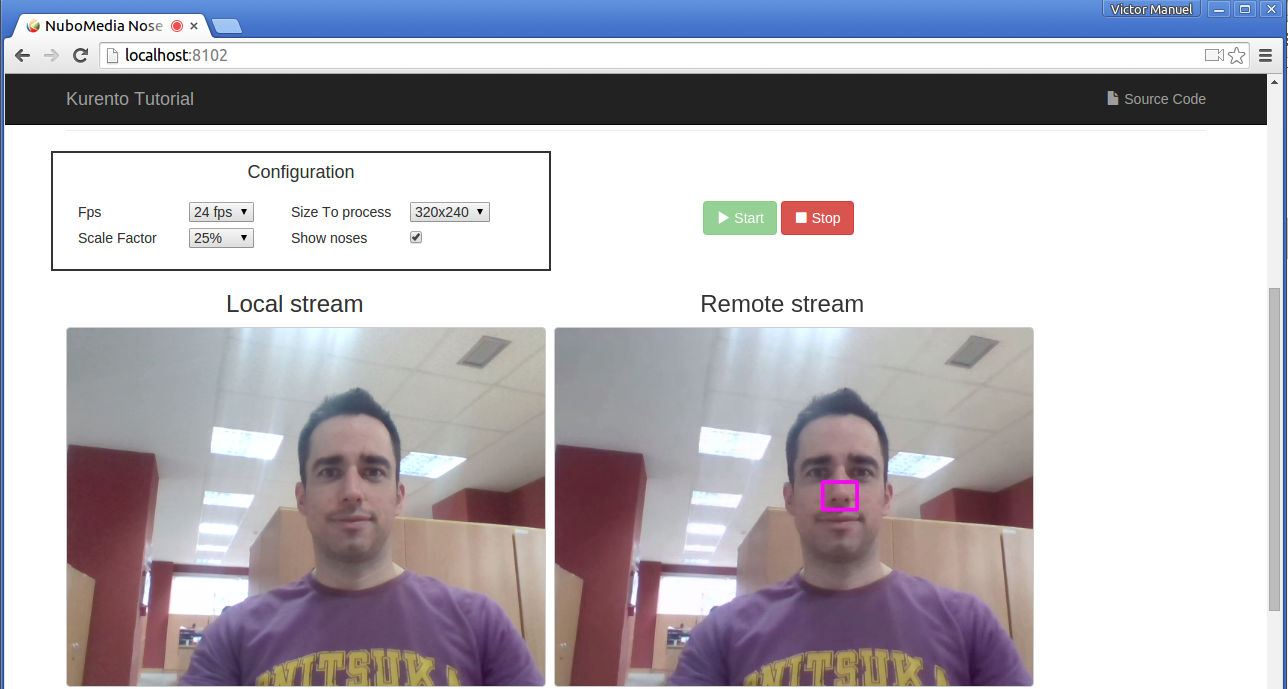
Nos gustaría usar los filtros creados por la empresa del consorcio Virtual Tools (<http://nubomedia-vca.readthedocs.io/en/latest/index.html>) que se muestran a continuación:

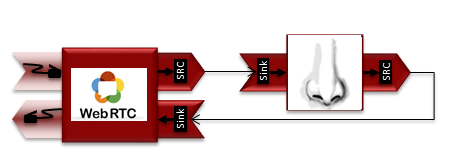
**1 .Face detector.**



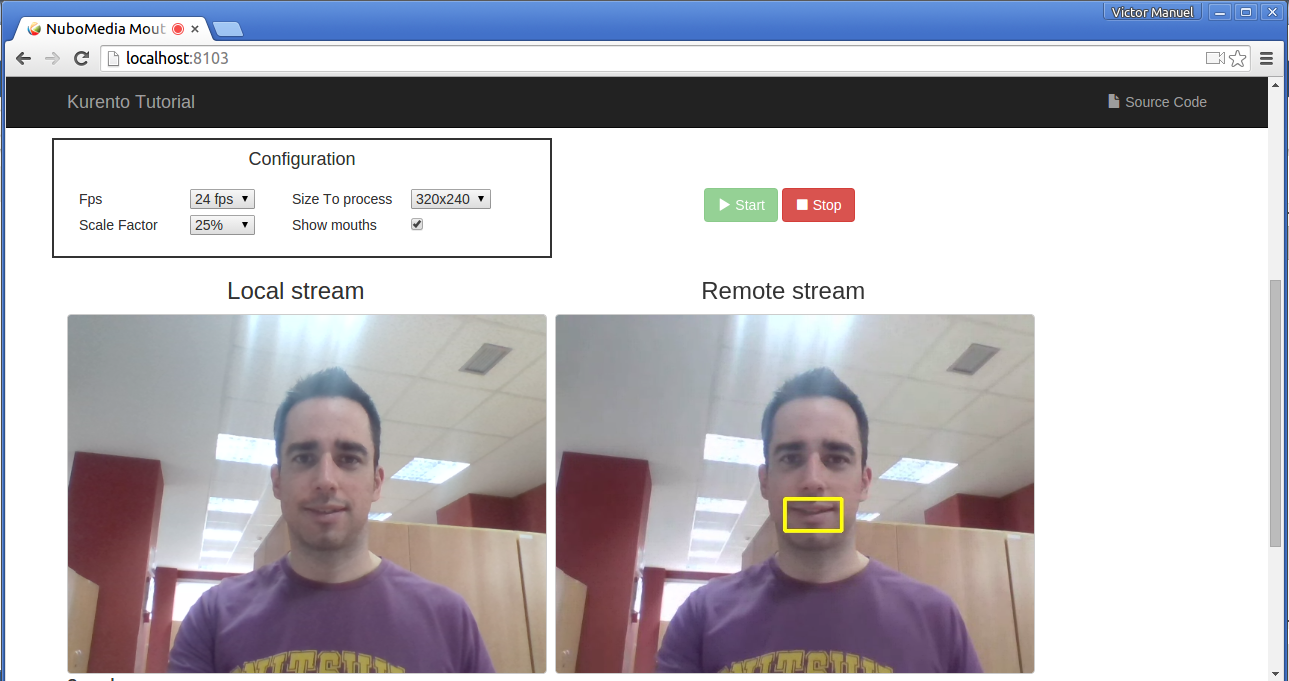


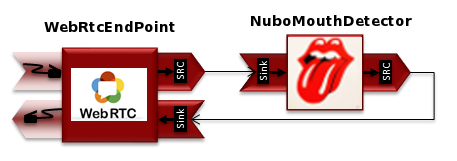
**2 .Nose detector.**



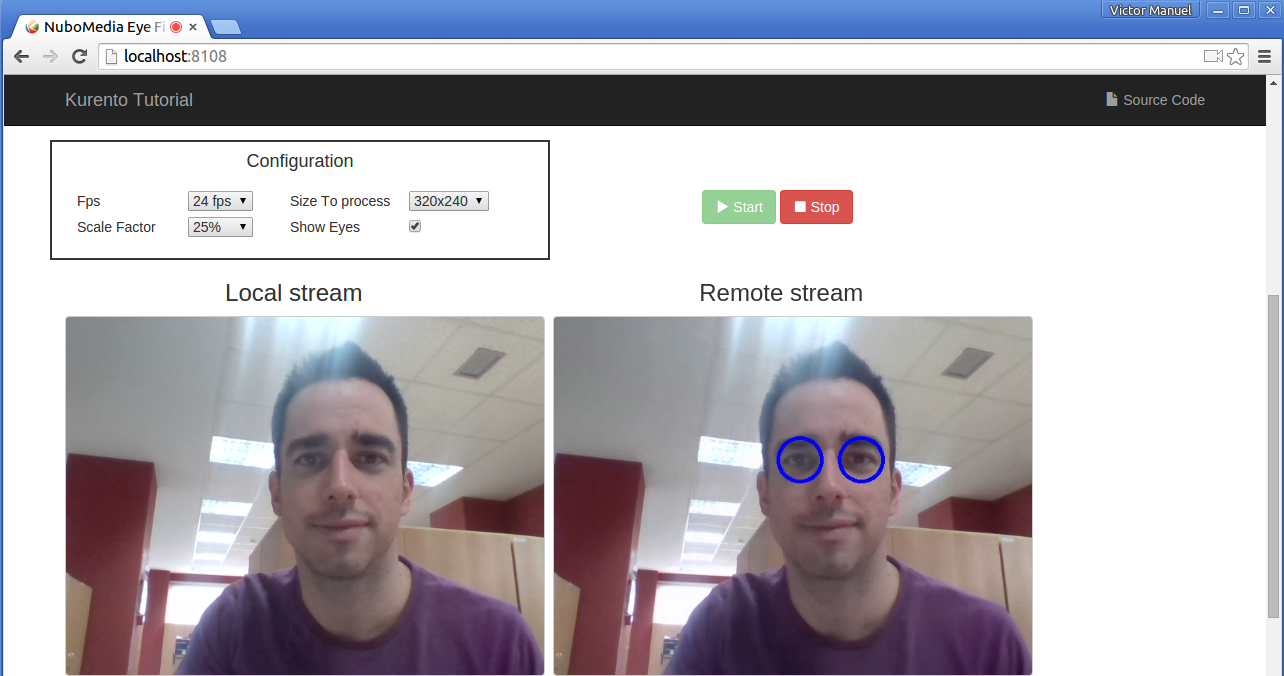


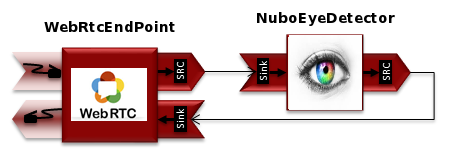
**3. Mouth detector.**





**4. Eye detector.**

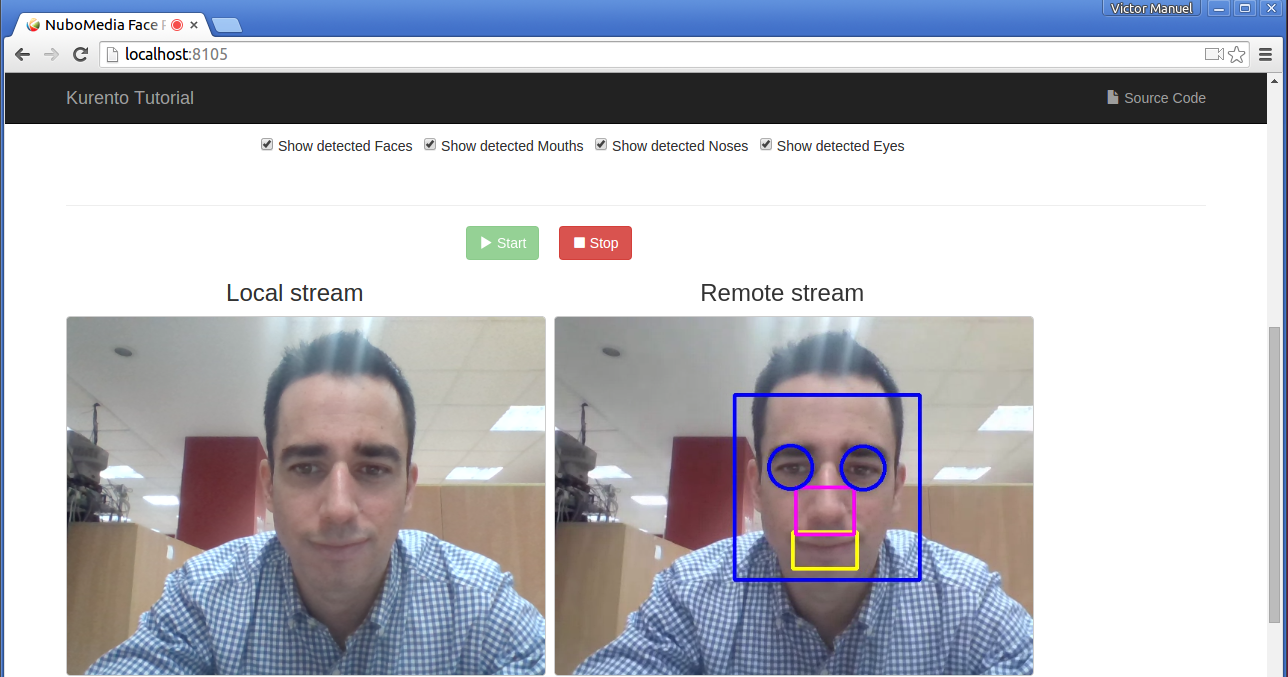




Ya que según la documentación aportada por Visual Tools es posible pasar metadatos de un filtro a otro (<http://nubomedia-vca.readthedocs.io/en/latest/face_profile.html>) como se muestra en la siguiente imagen:

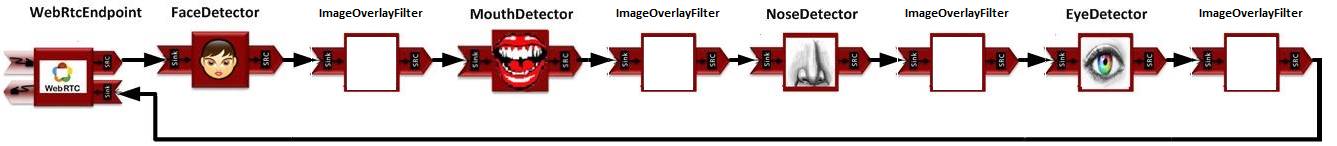


Obteniendo como resultado:

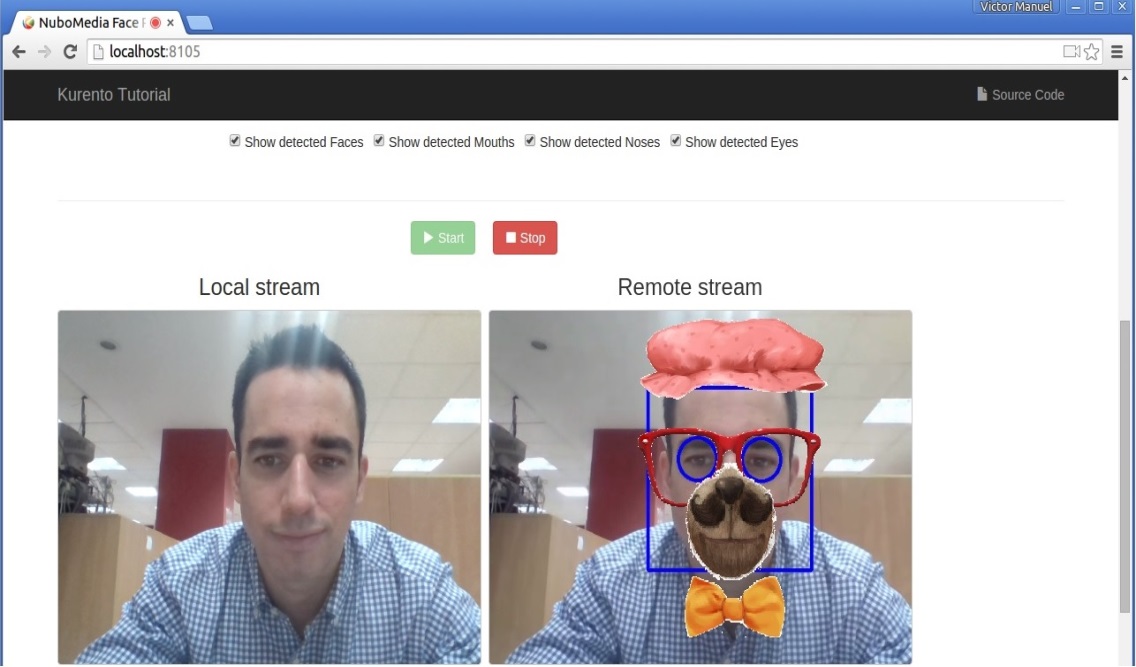


**Propuesta.**

Se querría algo similar a lo que se muestra en la imagen:



Para hacer composiciones como esta:



ImageOverlayFilter recogerá los metadatos del filtro que le precede y ubicará la imagen indicada de acuerdo con la posición dada en los metadatos recibidos. A su vez, se podrá realizar un desplazamiento en “x” e “y” y poder indicar un “alto” y “ancho” de la imagen.

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Tipo** | **Descripción** |
| imagePath | String | URL de la imagen a sobre impresionar. |
| ratioX | Float | Desplazamiento en el eje X con respecto a la coordenada superior izquierda. Los valores pueden ser positivos o negativos. |
| ratioY | Float | Desplazamiento en el eje Y con respecto a la coordenada superior izquierda. Los valores pueden ser positivos o negativos. |
| ratioWidth | Float | Ancho de la imagen. Solo positivo (0.0 a 1.0). |
| ratioHeight | Float | Alto de la imagen. Solo positivo (0.0 a 1.0). |

Este filtro no debe de realizar ningún tipo de detección, simplemente usar los metadatos que recoge del filtro predecesor, ubicar la imagen indicada por parámetro, realizar los ajustes según los parámetros y pasar el stream de video resultante a otro filtro o a un WebRTCEndPoint.

**Nota:**

La idea de este filtro es que debe de funcionar de forma similar a como funciona ***“KmsDetectFaces”*** y ***“KmsShowFaces”*** (<http://doc-kurento.readthedocs.io/en/stable/tutorials/java/tutorial-metadata.html>), donde el primero detecta la cara y el segundo pinta el rectángulo según los datos aportados por el primero.