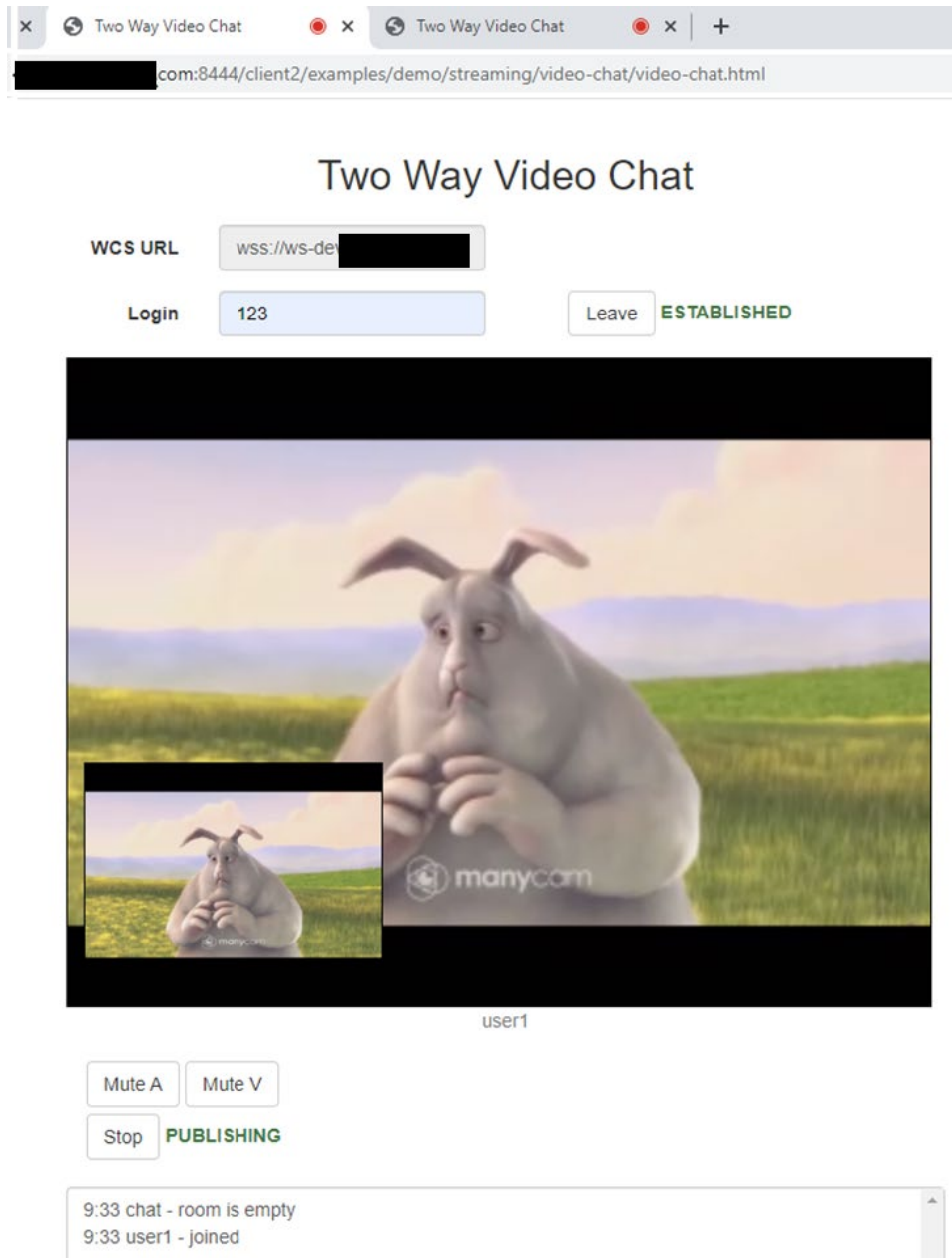


Алгоритм тестирования.

1. Публикуем потоки участников видеочата с помощью стандартного примера "Two Way Video Chat"



2. С помощью REST запроса на WCS сервере получаем имена потоков участников видео чата

```
curl -H "Content-Type: application/json" -X POST  
http://localhost:8081/rest-api/stream/find -d  
'{"published":true}'
```

имена потоков в нашем тестировании:

"room-e59fb5-user1-e6e6"

И

"room-e59fb5-123-d7b1"

3. Создаем микшер с помощью REST запроса

```
curl -H "Content-Type: application/json" -X POST  
http://localhost:8081/rest-api/mixer/startup -d  
'{"uri":"mixer://mixer1", "localStreamName":"st3"}'
```

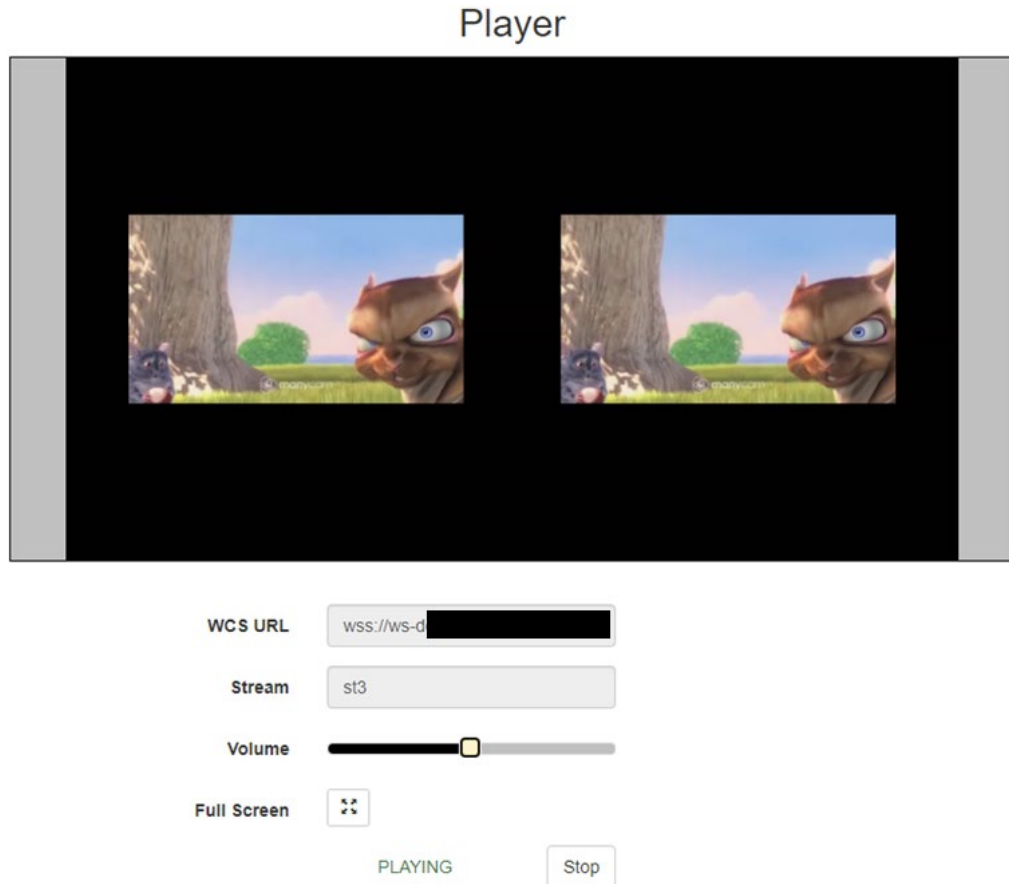
4. Добавляем в микшер потоки участников видео чата:

```
curl -H "Content-Type: application/json" -X POST  
http://localhost:8081/rest-api/mixer/add -d '{ "uri":  
"mixer://mixer1", "remoteStreamName": "room-e59fb5-user1-e6e6",  
"hasVideo": "true", "hasAudio": "true"}'
```

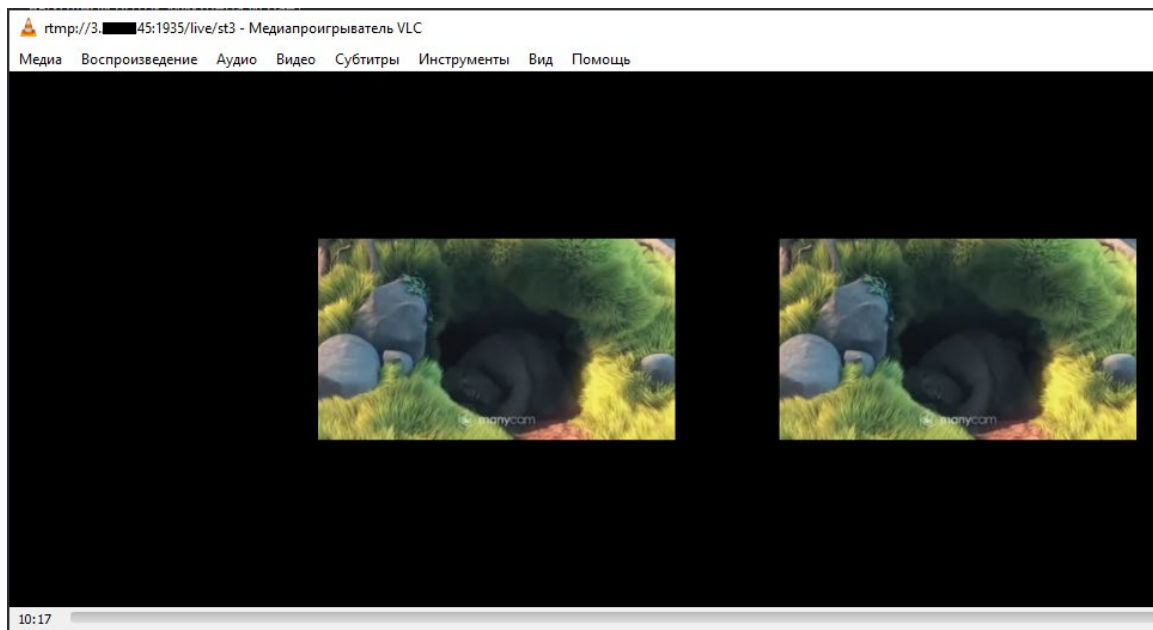
И

```
curl -H "Content-Type: application/json" -X POST
http://localhost:8081/rest-api/mixer/add -d '{ "uri":
"mixer://mixer1", "remoteStreamName": "room-e59fb5-123-d7b1",
"hasVideo": "true", "hasAudio": "true"}'
```

5. Проверяем воспроизведение выходного потока микшера с помощью плеера (у нас выходной поток микшера "st3")



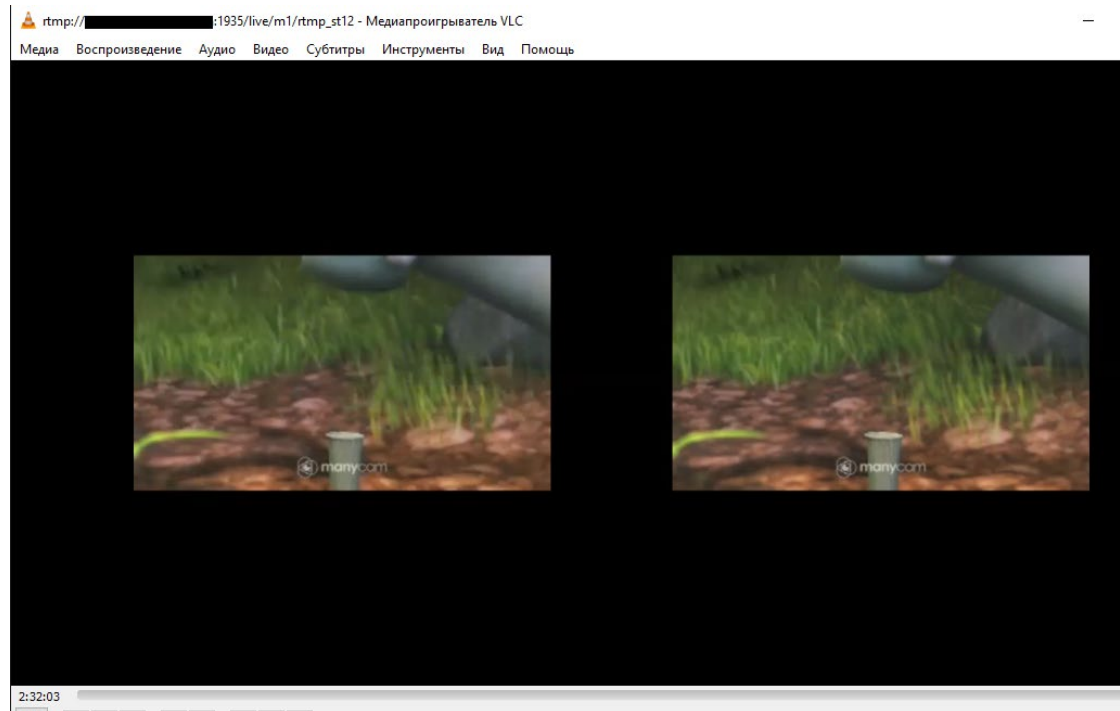
6. Проверяем воспроизведение выходного потока микшера по RTMP в VLC плеере



7. С помощью REST запроса делаем републикацию выходного потока микшера на сторонний RTMP сервер

```
curl -H "Content-Type: application/json" -X POST
http://localhost:8081/rest-api/push/startup -d '{"streamName":
"st3", "rtmpUrl": "rtmp://third-party-server:1935/live/m1"}'
```

и проверяем воспроизведение в VLC плеере (скриншот ниже от 10 итерации тестирования, поэтому имя потока микшера было "st12").



Было проведено 10 итераций тестирования. Все действия выполнялись вручную. Результат – 10 из 10 тестирований успешно. Микшер работал пока были опубликованы потоки от участников видеочата.