Getting Started Series:

Using Composited or Custom Layout

The webcast will begin shortly. Please stand by.

vidyo.io™

Getting Started Series:
Using Composited or Custom Layout

Sachin Hegde

Developer Evangelist

August 10, 2017

vidyo.io™

September 22-24, 2017

https://tadhack.com/2017/global/

https://vidyo.io

Twitter - @Vidyo_io

LinkedIn – vidyo-io

vidyo.io™

Upcoming Webinar

Vidyo.io Getting Started Webinar Series: Topic TBD – What do you want to see?

September 7 @ 2:00pm ET (11:00am PT)

Presented by:

Sachin Hegde, Developer Evangelist

We will email you an invitation next week!

vidyo.io™

Poll Time!

What topic would you like our next session to cover?

vidyo.io™

Getting Started Series:
Using Composited or Custom Layout

Sachin Hegde

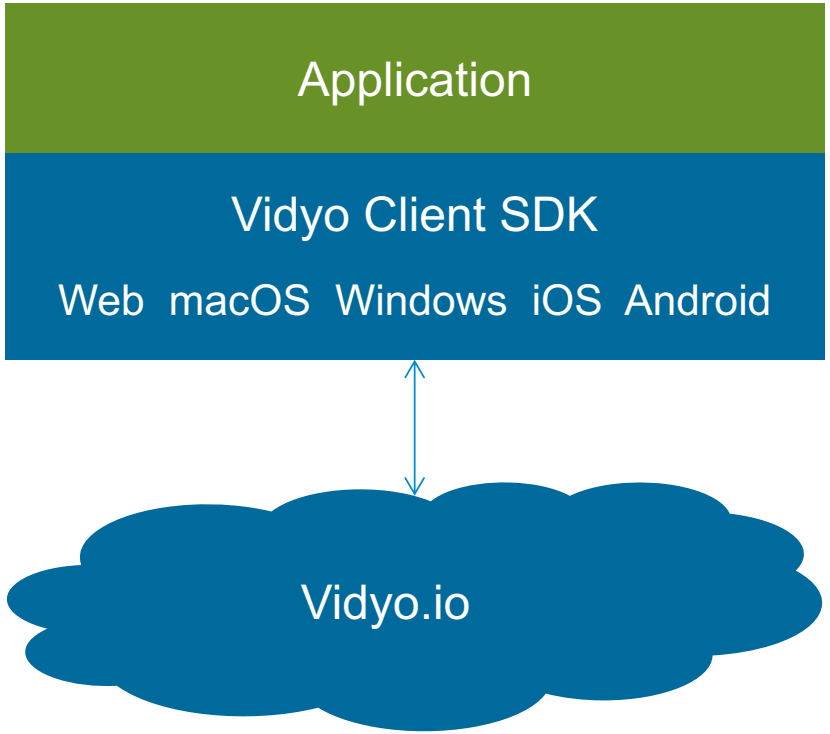
Developer Evangelist

August 10, 2017

vidyo.io™

Vidyo.io Intro

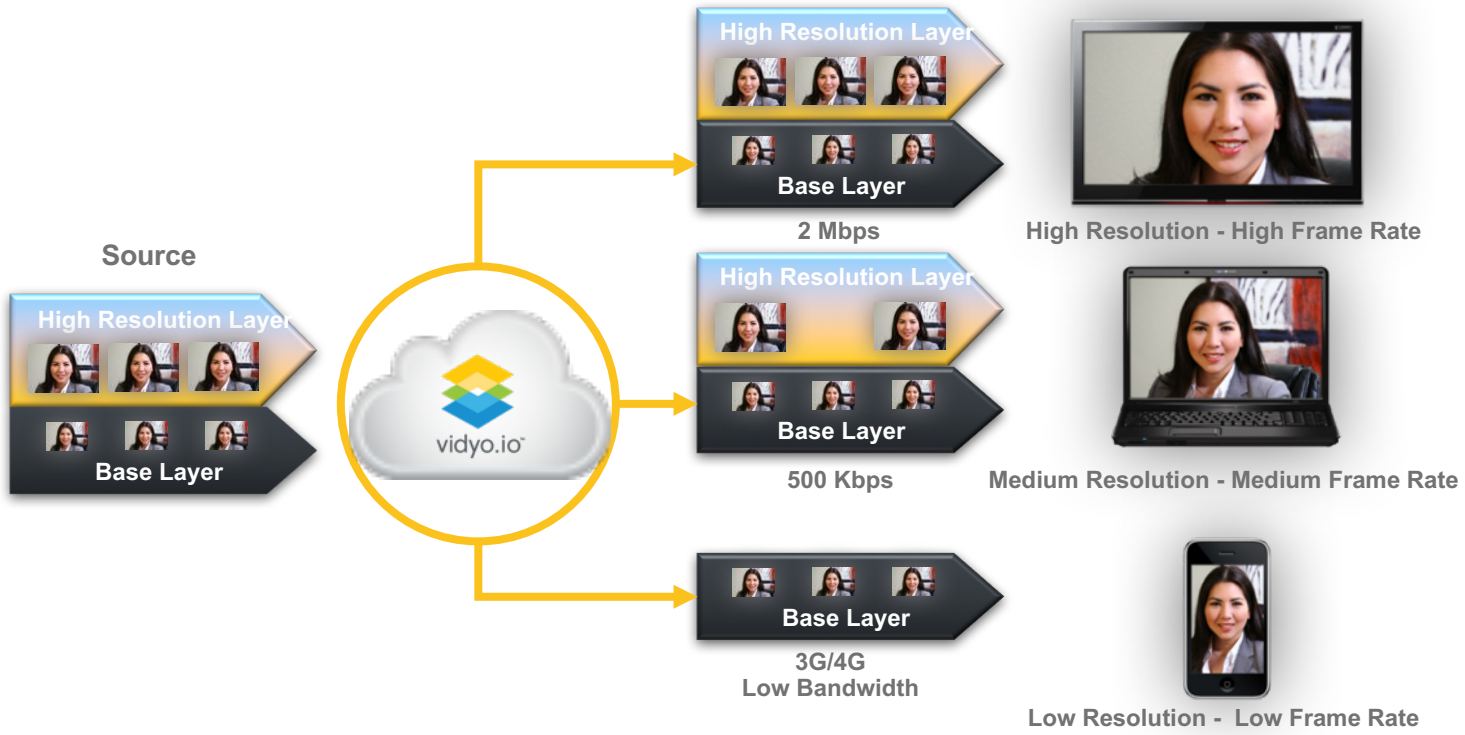
- CPaaS solution
- Globally distributed network
- Group video chat SDKs for WebRTC, mobile, and desktop apps
- Consistent API across various platforms



Vidyo.io: Powered by Industry Leading SFU

Benefits

- Massive Group Video Calls
- Low Latency, Conversational Video Chat
- Optimized Video Quality Per User
- Resilience to Unreliable Networks, Such as Wireless
- Globally Distributed Calls
- **Personalized Layouts**



vidyo.io™

Two approaches to group video layout

Composited Layout

- Assign one view
- Vidyo SDK composites the videos for you
- Simple to use
- Less flexibility on how the video tiles are displayed

Custom Layout

- Assign view for each remote participant
- Application composites the video tiles
- More involved
- Highly flexible since the application has all the freedom to display each video

Configuration for composited layout

VidyoConnector constructor

```
VC.CreateVidyoConnector({
    viewId: rendererId,                                  // Assign the platform specific View ID
    viewStyle: "VIDYO_CONNECTORVIEWSTYLE_Default",       // Visual style of the composited renderer
    remoteParticipants: 16,                              // Maximum number of participants to render
    logFileFilter: "warning",                            // Log filter levels
    logFileName: "",                                     // Log file name, leave blank for default
    userData: 0                                          // Arbitrary user data that can be retrieved later
})
```

Configuration for custom layout

VidyoConnector constructor

```
VC.CreateVidyoConnector({
    viewId: nil,                                          // Assign nil/null
    viewStyle: "VIDYO_CONNECTORVIEWSTYLE_Default",        // Visual style of the composited renderer
    remoteParticipants: 16,                               // Maximum number of participants to render
    logFileFilter: "warning",                             // Log filter levels
    logFileName: "",                                      // Log file name, leave blank for default
    userData: 0                                           // Arbitrary user data that can be retrieved later
})
```

Custom layout setup – Local camera

VidyoConnector - RegisterLocalCameraEventListener

- void OnLocalCameraAdded(VidyoLocalCamera localCamera)
- void OnLocalCameraRemoved(VidyoLocalCamera localCamera)
- void OnLocalCameraSelected(VidyoLocalCamera localCamera)
- void OnLocalCameraStateUpdated(VidyoLocalCamera localCamera, VidyoDeviceState state)

Custom layout setup - Remote Camera

VidyoConnector - RegisterRemoteCameraEventListener

- void OnRemoteCameraAdded(VidyoRemoteCamera remoteCamera, VidyoParticipant participant)
- void OnRemoteCameraRemoved(VidyoRemoteCamera remoteCamera, VidyoParticipant participant)
- void OnRemoteCameraStateUpdated(VidyoRemoteCamera remoteCamera, VidyoParticipant participant, VidyoDeviceState state)

Custom layout setup – Audio devices

VidyoConnector - RegisterLocalSpeakerEventListener

- void OnLocalSpeakerAdded(VidyoLocalSpeaker localSpeaker)
- void OnLocalSpeakerRemoved(VidyoLocalSpeaker localSpeaker)
- void OnLocalSpeakerSelected(VidyoLocalSpeaker localSpeaker)
- void OnLocalSpeakerStateUpdated(VidyoLocalSpeaker localSpeaker, VidyoDeviceState state)

VidyoConnector – RegisterLocalMicrophoneEventListener

- void OnLocalMicrophoneAdded(VidyoLocalMicrophone localMicrophone)
- void OnLocalMicrophoneRemoved(VidyoLocalMicrophone localMicrophone)
- void OnLocalMicrophoneSelected(VidyoLocalMicrophone localMicrophone)
- void OnLocalMicrophoneStateUpdated(VidyoLocalMicrophone localMicrophone, VidyoDeviceState state)

Custom layout setup – Assign and Show

VidyoConnector methods to assign a camera to a view -

- AssignViewToLocalCamera(Object viewId, VidyoLocalCamera localCamera, boolean displayCropped, boolean allowZoom)
- AssignViewToRemoteCamera(Object viewId, VidyoRemoteCamera remoteCamera, boolean displayCropped, boolean allowZoom)
- ShowViewAt(Object viewId, int x, int y, int width, int height)

Custom layout setup – Remote Participants

VidyoConnector – RegisterParticipantEventListener

- void OnParticipantJoined(VidyoParticipant participant)
- void OnParticipantLeft(VidyoParticipant participant)
- void OnDynamicParticipantChanged(ArrayList participants, ArrayList remoteCameras)
- void OnLoudestParticipantChanged(VidyoParticipant participant, boolean audioOnly)

Custom layout setup – Extended features

- VidyoConnector - RegisterLocalCameraFrameListener

boolean RegisterLocalCameraFrameListener(IRegisterLocalCameraFrameListener RegisterLocalCameraFrameListener, VidyoLocalCamera localCamera, int width, int height, long frameInterval)

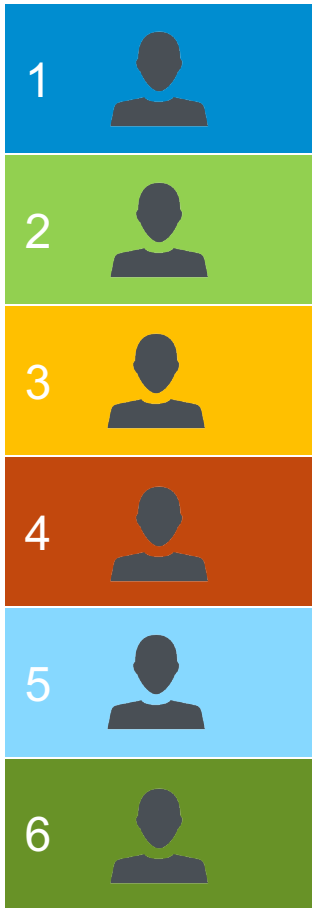
  - void OnLocalCameraFrame(VidyoLocalCamera localCamera, VidyoVideoFrame videoFrame)

- VidyoConnector – RegisterLocalMicrophoneFrameListener

boolean RegisterLocalMicrophoneFrameListener(IRegisterLocalMicrophoneFrameListener RegisterLocalMicrophoneFrameListener, VidyoLocalMicrophone localMicrophone)

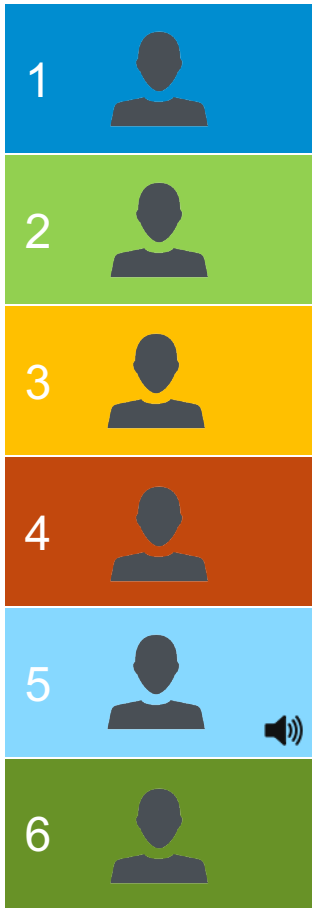
  - void OnLocalMicrophoneFrame(VidyoLocalMicrophone localMicrophone, VidyoAudioFrame audioFrame)

Dynamic participant list



vidyo.io™

Dynamic participant list



vidyo.io™

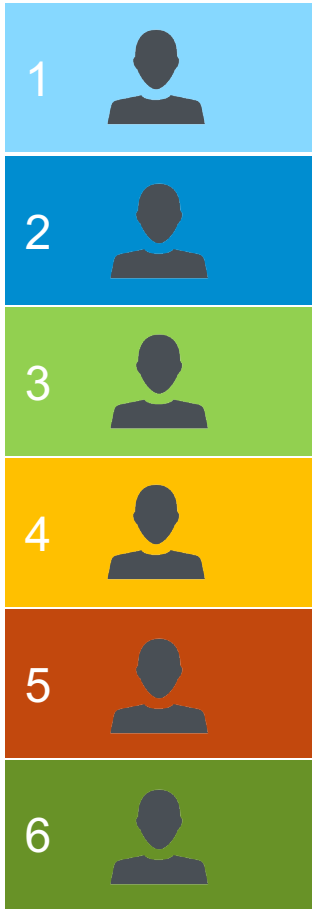
Dynamic participant list

vidyo.io™

Dynamic participant list

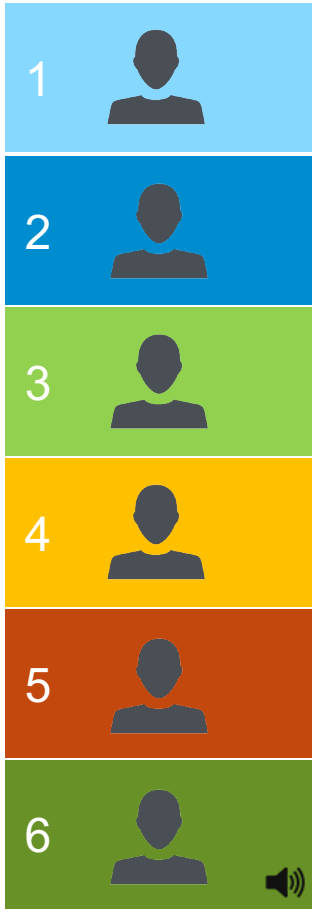
vidyo.io™

Dynamic participant list



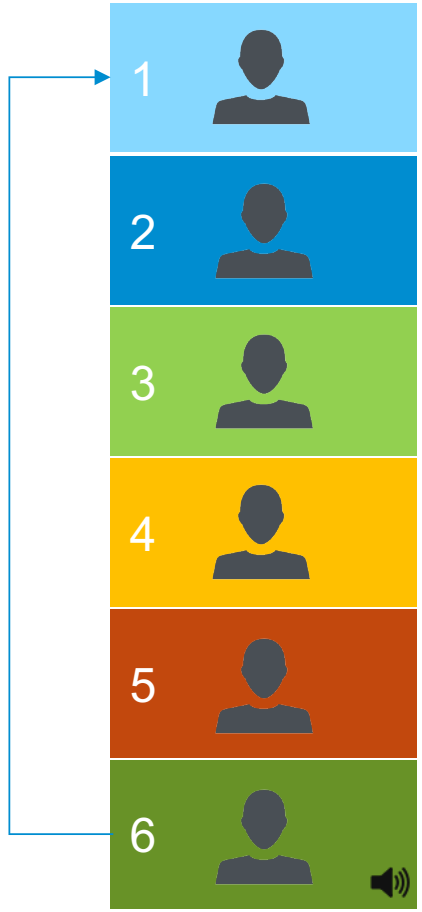
vidyo.io™

Dynamic participant list



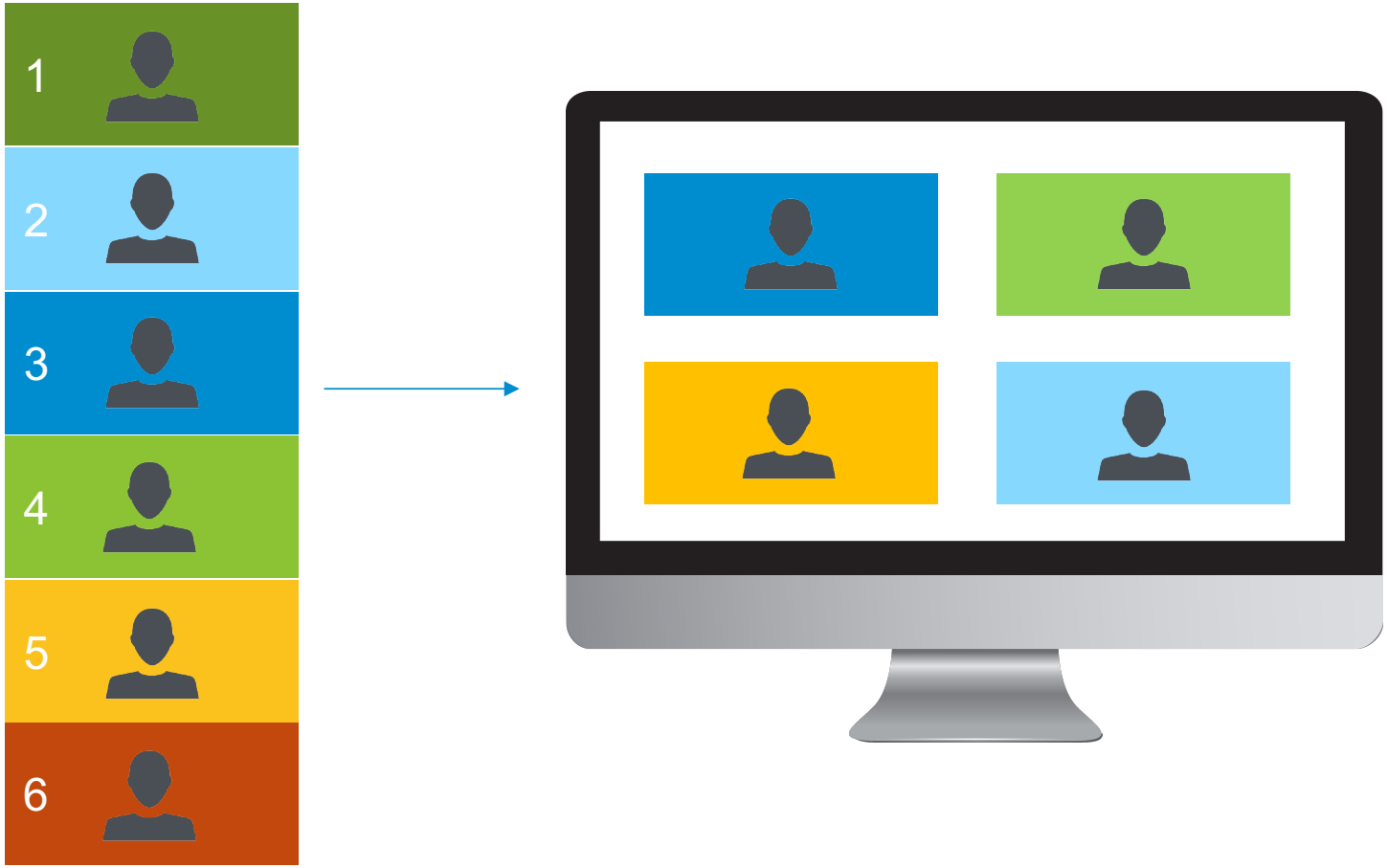
vidyo.io™

Dynamic participant list



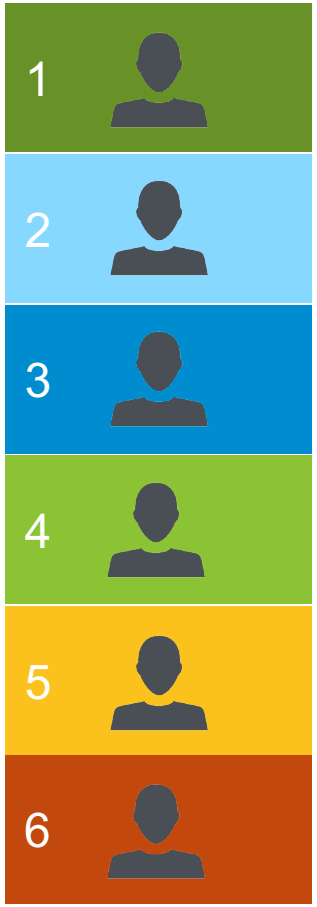
vidyo.io™

Dynamic participant list



vidyo.io™

Dynamic participant list



vidyo.io™

Demo

https://github.com/sachinVidyo/customview-swift-ios

vidyo.io™

Composited Vs Custom

Composited Setup

- Assign view Id in VidyoClientConnector constructor call
- Set maximum number of remote participants
- Done

Custom Setup

- Assign nil/null for view Id in VidyoClientConnector constructor call
- Register for local camera events
- Register for remote camera events
- Register for local speaker events
- Register for local microphone events
- Register for remote participants events
- Assign remote cameras to views
- Update views as participants leave/join

vidyo.io™

When to use custom layout?

- Custom needs like Video only, PIP only
- Spread out videos in multiple windows/screens
- Specific layout requirements e.g carousal
- Assigning particular views to particular users e.g Teacher, student views, webinars
- Conform to custom application style

Thank you!

Sachin Hegde

evangelist@vidyo.com

Twitter – twtsachin

LinkedIn – sachinh

https://vidyo.io

Twitter - @Vidyo_io

LinkedIn – vidyo-io

https://github.com/vidyo

vidyo.io™