**AgoraAbstractComponent,ngOnInit()**

*Setup placeholder remoteStreams + video containers, join session + channel, publish local streams, and setup callbacks from Agora stream*

Get current session

Acquire Mutex Lock on Component

Set the current session, including assigned seats

Messages arrive from server to set current view based on session state

Build remoteStreams list based on assigned seats from

isPlaying := false, isSubscribed := false; isHidden := false

localStream variable points into this list

Placeholder video containers/components are built via list

setupClient()

NgxAgoraService.setLogLevel()

NgxAgoraService.createClient()

joinChannel()

SessionApiService.getAgoraSessionToken(user via token, session acronym) -> Agora Token, userNumber

NgxAgoraService.client.join(Agora Token, session acronym, userNumber)

NgxAgoraService.client.enableDualStreams()

publishLocalStreams(userNumber, userId)

Select hi/low stream parameters based on role

stream.init()

Store information on localStream remoteStreamInfo object which points into the remoteStreams list

Get UserInfo from API service

stream.play(videoContainerId) to show local stream preview

Setup callbacks on channel(streamAdded(), stream-subscribed, peer-leave)

Release Mutex Lock on Component

**(on) stream-added**

streamAdded()

Get userInfo via streamId() from API service

Acquire Mutex Lock on Component

Update remoteStream with userInfo, isSubscribed := true

NgxAgoraService.client.subscribe(stream)

Will trigger (on) stream-subscribed in the future, after Mutex is released

Release Mutex Lock on Component

**(on) stream-subscribed** (handler from Agora once stream is subscribed)

Acquire Mutex Lock on Component

Update remoteStream isPlaying := true

stream.play(videoContainerId)

Release Mutex Lock on Component

**(on) peer-leave**

removeVideo()

Acquire Mutex Lock on Component

Update remoteStream isSubscribed := false, isPlaying := false

stream.stop()

NgxAgoraService.client.unsubscribe(stream)

Release Mutex Lock on Component

**(evt) ChangeViewAll**

handleGlobalEvents()

*Handles all local and server global events. View change global events are created when the instructor clicks buttons*

Acquire Mutex Lock on Component

switchToGroup()  
switchToSpotlight(userId)  
switchToInstructor()

*Call appropriate method based on global event*

remoteStream isSpotlight is set for instructor

showRemoteStream()

remoteSteram isSubscribed := true isHidden := false

NgxAgoraService.client.subscribe(stream)

Will trigger **(on) stream-subscribed** in the future, after Mutex is released

hideRemoteStream()

remoteStream isHidden := true; isSubscribed := false; isPlaying := false

stream.stop()

NgxAgoraService.client.unsubscribe(stream)

Release Mutex Lock on Component

Key

* Entry points are in **bold**
* Methods are underlined, such as setupClient()
* Mutex lock operations are highlighted in green
* Stream operations are highlighted in yellow
* remoteStream properties are highlighted in grey, such as isSubscribed := false