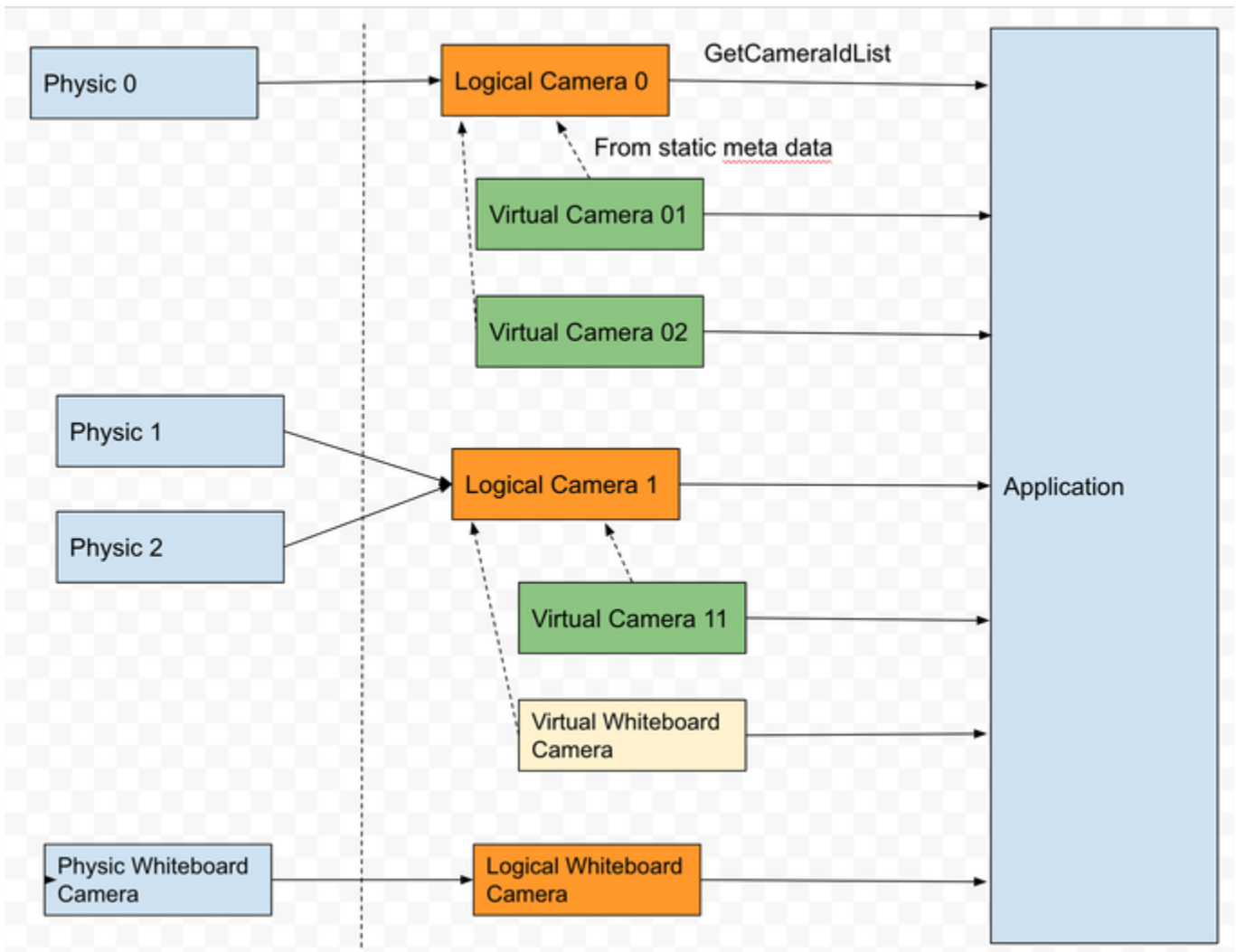# Zoom Rooms Appliance Camera Smart Gallery

Version: 1.2.1

Camera Structure

Camera Structure

- Zoom Rooms App uses Android's standard API `CameraManager.getCameraIdList` to get all camera ids including logical and virtual cameras from the system.
- Zoom Rooms App uses Android's standard API `CameraManager.getCameraCharacteristics` to get the logical and virtual camera's static Metadata(with vendor tags).
- From the camera's static Metadata, we can get the camera's type(logical/virtual) by vendor tag:`ZR_INFO_DEVICE_IS_VIRTUAL_CAMERA (us.zoom.zr.vendoros.smartCamera.isVirtualCamera)`
- From the virtual camera's static Metadata, we can get which logical camera it is related to by vendor tag: `ZR_INFO_RELATED_LOGICAL_CAMERA_ID (us.zoom.zr.vendoros.smartCamera.relatedLogicCameraId)`
- If the logical camera is closed and non of its virtual cameras are opened, then all of the virtual cameras related to it will also become unavailable.
- ZR registers Android's standard API `CameraManager.AvailabilityCallback` to handle the logical and virtual camera's availability change.
- The virtual camera needs to support capture video stream with resolution 720P and aspect-ratio 16:9.

Camera Metadata

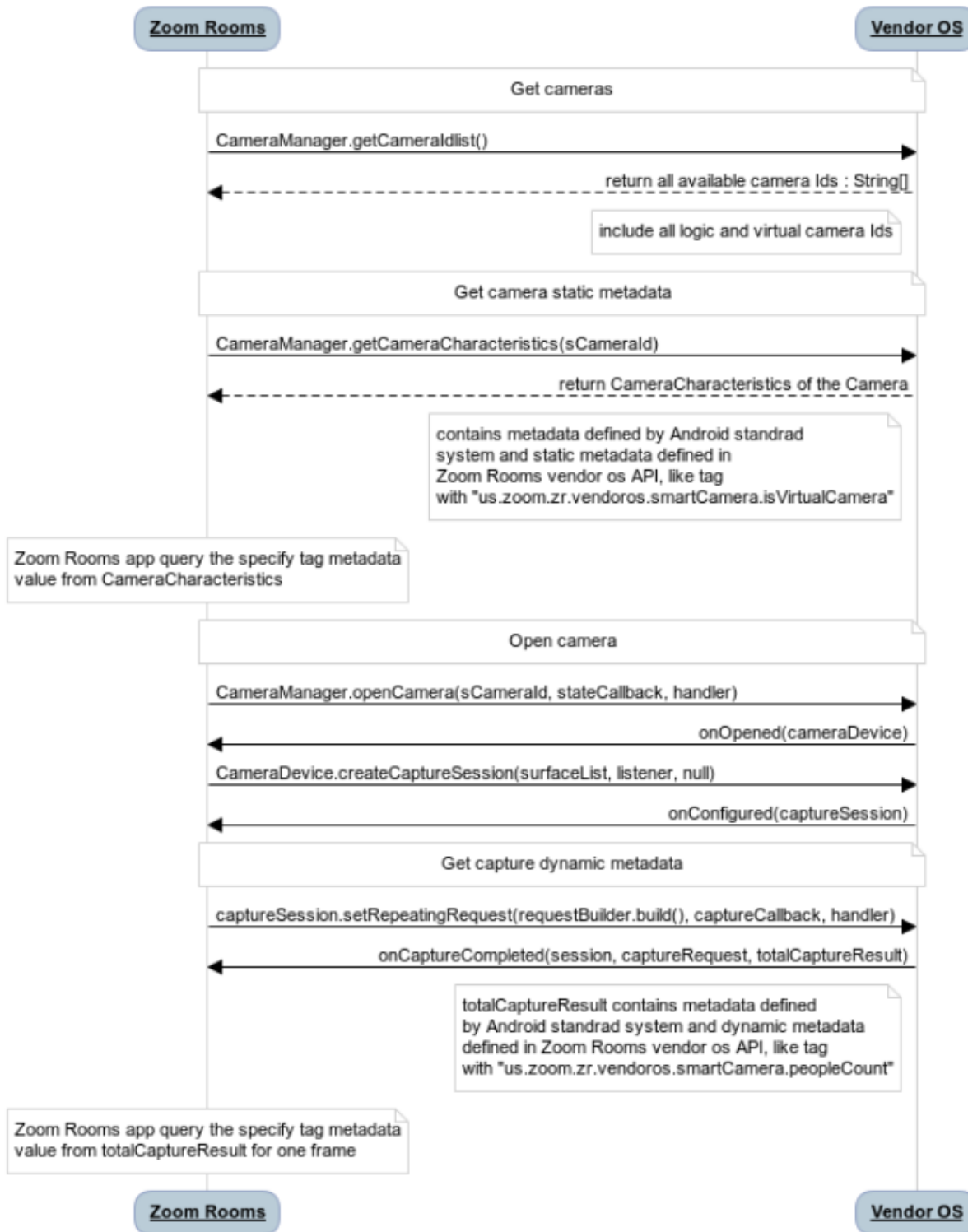### Static Metadata

- ZR uses `CameraManager.getCameraCharacteristics` with vendor static Metadata tags to get the Static Metadata.

- ZR uses `List<CameraCharacteristics.Key<?>> keyList = characteristics.getKeys();` to get all camera Metadata and query the Vendor Tag.
- The Metadata contains the virtual/logic camera's type and related logical camera id for the virtual camera.
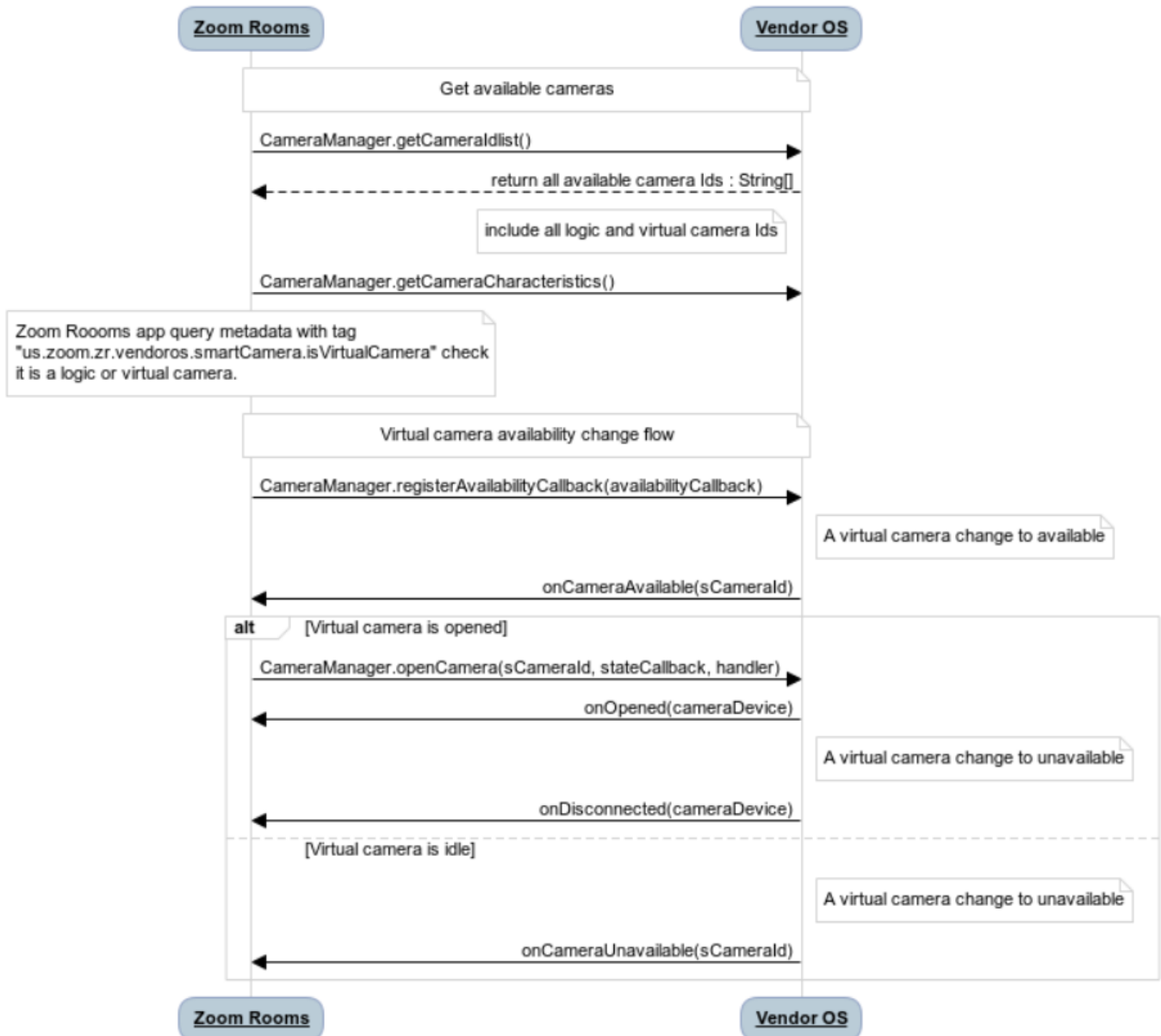
Dynamic Metadata

- ZR registers `CaptureCallback` and gets a `CaptureResult` per frame through the `onCaptureCompleted` callback.
- ZR uses vendor dynamic Metadata tags to get Metadata from `CaptureResult`

Zoom Rooms Applicance Smart Gallery API-- Get Metadata

Zoom Rooms   **Vendor OS**

Get cameras

CameraManager.getCameraIdlist() →

← return all available camera Ids : String[]

include all logic and virtual camera Ids

Get camera static metadata

CameraManager.getCameraCharacteristics(sCameraId) →

← return CameraCharacteristics of the Camera

contains metadata defined by Android standrad system and static metadata defined in Zoom Rooms vendor os API, like tag with "us.zoom.zr.vendoros.smartCamera.isVirtualCamera"

Zoom Rooms app query the specify tag metadata value from CameraCharacteristics

Open camera

CameraManager.openCamera(sCameraId, stateCallback, handler) →

← onOpened(cameraDevice)

CameraDevice.createCaptureSession(surfaceList, listener, null) →

← onConfigured(captureSession)

Get capture dynamic metadata

captureSession.setRepeatingRequest(requestBuilder.build(), captureCallback, handler) →

← onCaptureCompleted(session, captureRequest, totalCaptureResult)

totalCaptureResult contains metadata defined by Android standrad system and dynamic metadata defined in Zoom Rooms vendor os API, like tag with "us.zoom.zr.vendoros.smartCamera.peopleCount"

Zoom Rooms app query the specify tag metadata value from totalCaptureResult for one frame

Zoom Rooms   **Vendor OS**

Zoom Rooms Applicance Smart Gallery--Virtual camera availability

Camera Availability

Open and Capture camera

- ZR uses Android's Camera2 API to open and capture video for virtual cameras.
- ZR may open and capture logical and several virtual cameras at the same time.
- The logical camera returns the complete view of the camera, like today's normal camera, while the virtual camera returns a special feed-type view.

Related Vendor OS API Definition

Vendor Tags For Static Metadata:

- Tags identifying which logical camera the device relates to and the device type(logical or virtual) are defined in `CameraCharacteristic`

The below MetaData Keys definitions needs to be added to the Vendor System APIs
Refer to :

- https://developer.android.com/reference/android/hardware/camera2/CameraCharacteristics?hl=en https://developer.android.com/reference/android/hardware/camera2/CameraCharacteristics.Key?hl=en

```
    /**
      * Key used to lookup which logical camera device the virutal
camera is related to.
      * If the camera is a logical camera, just return a null value.
      */
   public static final CameraCharacteristics.Key<String>
ZR_INFO_RELATED_LOGICAL_CAMERA_ID =
           new CameraCharacteristics.Key<String>("us.zoom.zr.
vendoros.smartCamera.relatedLogicCameraId", String.class);


     /**
      * Key used to lookup whether the device is a virtual camera.
      */
     public static final CameraCharacteristics.Key<Boolean>
ZR_INFO_DEVICE_IS_VIRTUAL_CAMERA =
             new CameraCharacteristics.Key<Boolean>("us.zoom.zr.
vendoros.smartCamera.isVirtualCamera", boolean.class);
```

Vendor Tags For Dynamic Metadata:

For example:

- Tag for returning the people count/Video position/Active speaker in the current frame

```
/**
 * Key used to return the people count in the current frame.
 */
public static final CaptureResult.Key<Integer>
ZR_INFO_FRAME_PEOPLE_COUNT =
        new CaptureResult.Key<Integer>("us.zoom.zr.vendoros.
smartCamera.peopleCount", int.class);

/**
 * Key used to return if the user in the video is the active
speaker.
 */
public static final CaptureResult.Key<Boolean>
ZR_INFO_FRAME_IS_ACTIVE_SPEAKER =
        new CaptureResult.Key<Boolean>("us.zoom.zr.vendoros.
smartCamera.isActiveSpeaker", boolean.class);

/**
  * Key for return video relative position from left to right.
Return 0 if the video of the camera is overall view, otherwise it
starts from 1.
  */
public static final CaptureResult.Key<Integer>
ZR_INFO_FRAME_VIDEO_POSITION_INDEX=
        new CaptureResult.Key<Integer>("us.zoom.zr.vendoros.
smartCamera.videoPositionIndex", int.class);
```

How to enable the feature

- Before call system API to enable Smart gallery, ZR will call the below API first to tell the system the upper bound for the number of video streams to support in smart gallery mode.

```
/**
 * Sets the upper bound for the number of video streams to support in
smart gallery mode.
 * ZR calls this API before enabling {@link
SmartCameraControlMode#ZRAPI_SMART_CAMERA_CONTROL_MODE_SMART_GALLERY}
 *
 * @param maxStreamCount the upper bound of max video streams to
support in smart gallery mode.
 * @return true if operation was successful, false otherwise
 */
boolean setMaxSupportVideoStreamsInSmartGallery(int maxStreamCount);
```

- Vendor OS needs to implement the following APIs to return the supported smart camera control modes.

```java
 /**
     * Query supported smart camera control modes of the Camera.
Require {@link CameraCapability#ZRAPI_CAMERA_CAP_SMART_CONTROL}
     * @param cameraId the camera's ID
     *
     * @return supported control modes defined in {@link
SmartCameraControlMode}
     */
    int getSupportSmartCameraControlModes(String cameraId);

    /**
     * Set the camera's smart control mode. Require {@link
CameraCapability#ZRAPI_CAMERA_CAP_SMART_CONTROL}
     * @param cameraId  the camera's ID
     * @param mode smart control mode defined in {@link
SmartCameraControlMode}
     * @return Api call success or not
     */
    boolean setSmartCameraControlMode(String cameraId, int mode);

    /**
     * Query current smart camera control mode of the camera. Require
{@link CameraCapability#ZRAPI_CAMERA_CAP_SMART_CONTROL}
     * @param cameraId the camera's ID
     *
     * @return supported control mode defined in {@link
SmartCameraControlMode}
     */
    int getCurrentSmartCameraControlMode(String cameraId);
```

- The camera needs to support at least two of the following modes:

```java
    /**
     * Multi focus mode
     */
    public static final int
ZRAPI_SMART_CAMERA_CONTROL_MODE_MULTI_FOCUS     = 0x08;

        /**
     * Smart gallery mode
     */
    public static final int
ZRAPI_SMART_CAMERA_CONTROL_MODE_SMART_GALLERY   = 0x10;
```

===

Change History:

V1.2.1:

1. Update APIs to get Static Metadata.
2. Typo fix.

V1.2:

1. change the tags, "us.zoom.zr.vendoros.XXX" ==> "us.zoom.zr.vendoros.smartCamera.XXX" .
2. Add `setMaxSupportVideoStreamsInSmartGallery` API.
3. change ZR_INFO_DEVICE_FEED_TYPE to use `ZR_INFO_DEVICE_IS_VIRTUAL_CAMERA (us.zoom.zr.vendoros.` `smartCamera.isVirtualCamera)`. Using `CameraDeviceType` to define the camera type.
4. If the logical camera is closed and non of its virtual cameras are opened, then all of the virtual cameras related to it will also become unavailable.