



# JCTVC-Q0096/JCT3V-H0030: MV-HEVC/SHVC HLS: Overlay info SEI message

*Jill Boyce and Stephan Wenger*

# Introduction

- Overlay info SEI message proposed to support user selectable overlays
- Two syntax options proposed
  - Option 1: Allocate a value of AuxId for overlays
  - Option 2: Use already allocated “Unspecified” range of AuxId
- Option 2 is identical to proposal in JCTVC-Q0045. Related to JCTVC-O0358, JCTVC-P0092, JCTVC-P0135.
- Notes from San Jose meeting about P0092/P0135 say:
  - *“At the concept level, it was agreed to plan to support such a capability. Further study was needed to work out details.”*

# Proposal use case overview

- 3 auxiliary picture types are used to encode an overlay, each with a different value of `nuh_layer_id`
  - Overlay content
  - Overlay layout (optional)
  - Overlay alpha (optional)
- Each auxiliary picture is associated with a primary (non-auxiliary) picture
- Multiple overlays may be associated with the same primary picture, selected by end user, aided by information in SEI message

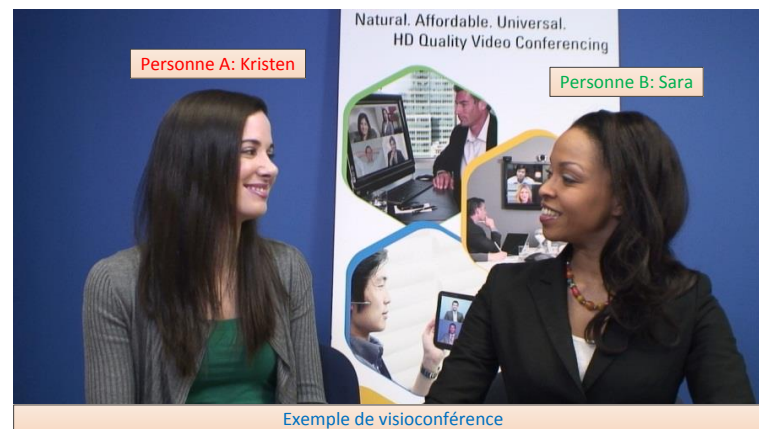
# Use case example



**Figure 1. Primary picture**



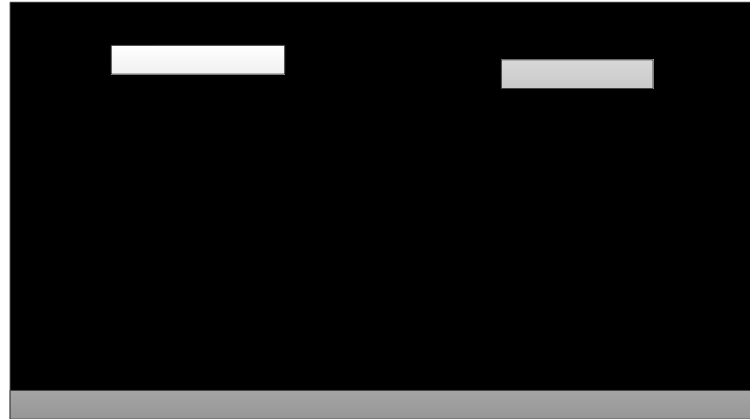
2(a)



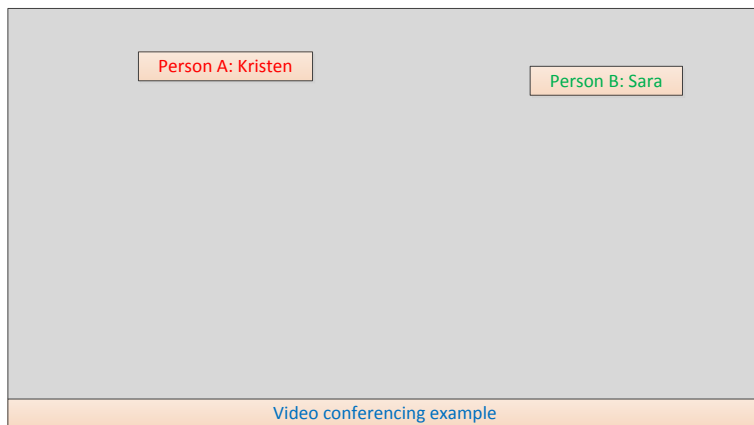
2(b)

**Figure 2. Resulting display with (a) English overlay (b) French overlay, with all elements displayed**

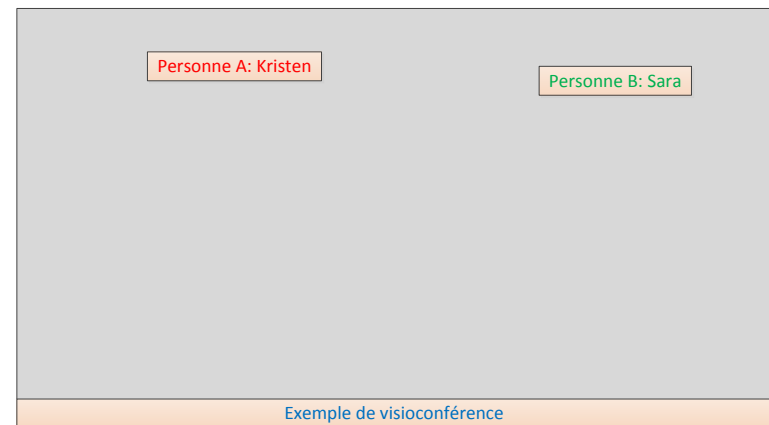
# Example case example continued



**Figure 3. Overlay label**



4(a)



4(b)

**Figure 4. Overlay content for (a) English overlay (b) French overlay**

# Overlay info SEI message

- Two syntax options proposed
- Indicates nuh\_layer\_id values for the auxiliary picture layers used to define an overlay, and presence for optional aux pics
- Describes contents of the overlay layout
  - How regions of the picture are divided into individually controllable overlay elements
- Describes names of overlay and individual overlay elements

# Option 1. Allocate a value of AuxId

Table F-2 – Mapping of AuxId to the type of auxiliary pictures

AuxId	Name of AuxId	Type of auxiliary pictures
0	AUX_ALPHA	Alpha plane of non-auxiliary picture
1	AUX_DEPTH	Depth picture
2	AUX_OVERLAY	Overlay picture
3-127		Reserved
128-143		Unspecified
144-255		Reserved

- One value of AuxId can be used for all 3 types of overlay auxiliary pictures

# Option 1: SEI message syntax

overlay_info( ) {	
<b>overlay_info_cancel_flag</b>	u(1)
if (!overlay_info_cancel_flag) {	
<b>num_overlays_minus1</b>	ue(v)
<b>overlay_info_name_len</b>	ue(v)
for( i = 0; i < num_overlays_minus1; i++ ) {	
<b>overlay_name[ i ]</b>	f(v)
<b>overlay_content_layer_id[ i ]</b>	u(6)
<b>overlay_label_present_flag[ i ]</b>	u(1)
if (overlay_label_present_flag[ i ]) {	
<b>overlay_label_layer_id[ i ]</b>	u(6)
<b>overlay_alpha_present_flag[ i ]</b>	u(1)
if (overlay_alpha_present_flag[ i ]) {	
<b>overlay_alpha_layer_id[ i ]</b>	u(6)
if (overlay_label_present_flag[ i ]) {	
<b>num_overlay_elements_minus1[ i ]</b>	ue(v)
for( j = 0; j <= num_overlay_elements_minus1[ i ]; j++ ) {	
<b>overlay_element_name[ i ][ j ]</b>	f(v)
<b>overlay_element_label_min[ i ][ j ]</b>	u(v)
<b>overlay_element_label_max[ i ][ j ]</b>	u(v)
}	
}	
}	
<b>overlay_info_persistence_flag</b>	u(1)
}	
}	



## Option 2: Use “Unspecified” AuxId values

---

- Use (possibly) different values of AuxId for each of the three type of overlay auxiliary pictures, from the Unspecified range
- Syntax identical to that proposed in JCTVC-Q0045.

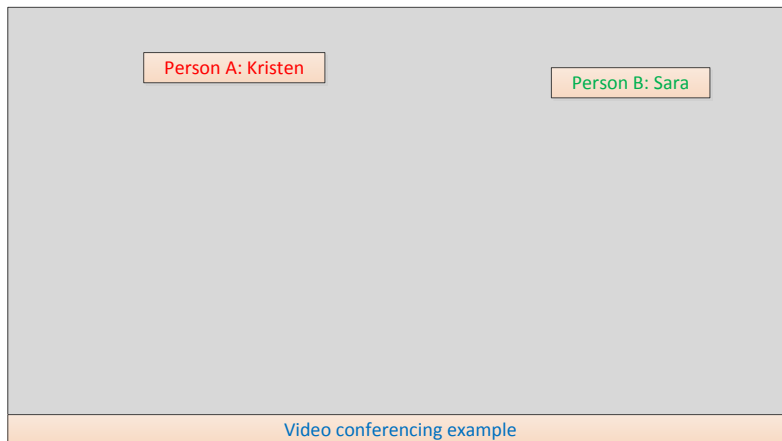
# Option 2: SEI message syntax

Differences from  
Option 1 in cyan

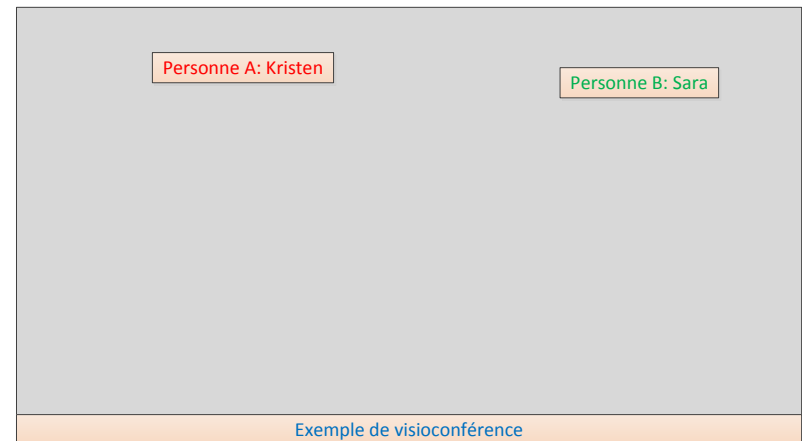
overlay_info() {	
<b>overlay_info_cancel_flag</b>	u(1)
if (!overlay_info_cancel_flag) {	
<b>overlay_content_aux_id_minus128</b>	ue(v)
<b>overlay_label_aux_id_minus128</b>	ue(v)
<b>overlay_alpha_aux_id_minus128</b>	ue(v)
<b>num_overlays_minus1</b>	ue(v)
<b>overlay_info_name_len</b>	ue(v)
for( i = 0; i < num_overlays_minus1; i++ ) {	
<b>overlay_idx[ i ]</b>	ue(v)
<b>overlay_name[ i ]</b>	f(v)
<b>overlay_content_layer_id[ i ]</b>	u(6)
<b>overlay_label_present_flag[ i ]</b>	u(1)
if (overlay_label_present_flag[ i ]) {	
<b>overlay_label_layer_id[ i ]</b>	u(6)
<b>overlay_alpha_present_flag[ i ]</b>	u(1)
if (overlay_alpha_present_flag[ i ]) {	
<b>overlay_alpha_layer_id[ i ]</b>	u(6)
if (overlay_label_present_flag[ i ]) {	
<b>num_overlay_elements_minus1[ i ]</b>	ue(v)
for( j = 0; j <= num_overlay_elements_minus1[ i ]; j++ ) {	
<b>overlay_element_name[ i ][ j ]</b>	f(v)
<b>overlay_element_label_min[ i ][ j ]</b>	u(v)
<b>overlay_element_label_max[ i ][ j ]</b>	u(v)
}	
}	
}	
}	
<b>overlay_info_persistence_flag</b>	u(1)
}	
}	

# Aux pic type: Overlay content

- Contains overly content, intended for display as an overlay over the associated primary picture
- Multiple overlay content aux pics may be associated with the same primary picture
  - For example, for different languages



4(a)



4(b)

**Figure 4. Overlay content for (a) English overlay (b) French overlay**

# Aux pic type: Overlay layout

- Describes a mapping of the picture into multiple overlay elements
- May be shared by multiple overlays
- Parameters in the overlay info SEI message are used to interpret content
  - Coded sample values determine element selection
  - Min and max sample values for each element signaled in SEI
- Overlay layout pictures may use a lower frame rate than overlay content



## Aux pic type: Overlay alpha

---

- Apply alpha to the overlay content picture, rather than to the primary picture
- May be shared by multiple overlays
- Optional

# Associations of aux pic layers with overlays

- Number of overlays
- For each overlay
  - Layer id of each of
    - Overlay content
    - Overlay layout (optional)
    - Overlay alpha (optional)