OPEN IPTV FORUM (OIPF) DAE OBJECTS AND <VIDEO> EXTENSIONS

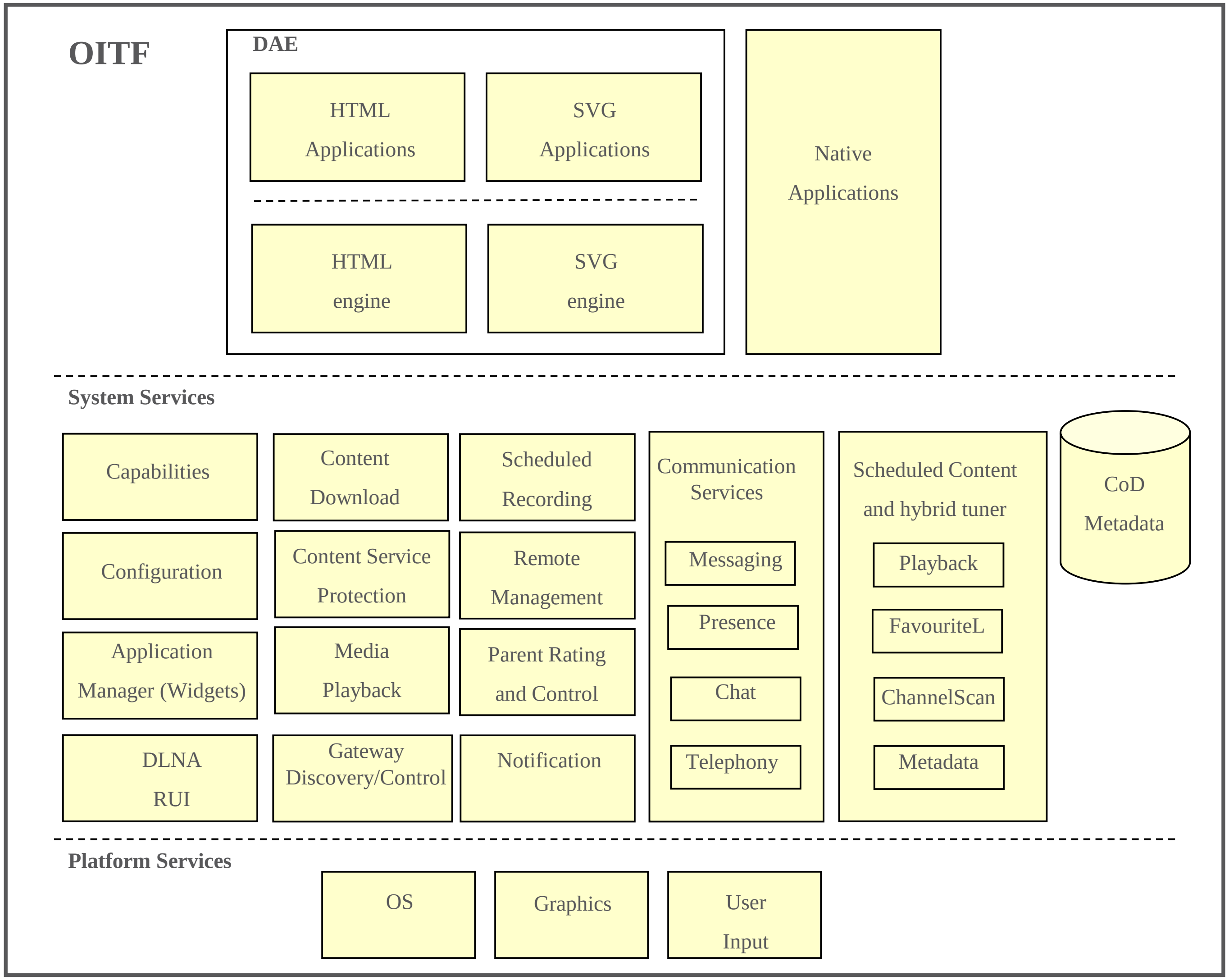
PRESENTATION BY

JAN LINDQUIST

OVERVIEW

- OITF Reference Architecture & High Level Functions
- DAE Sample APIs
- Comparison of HTML5 <video> and DAE video objects
- Reaching Expectations
- W3C Collaborations

OITF REFERENCE ARCHITECTURE



HIGH LEVEL FUNCTION

High-level Function [which can include several embedded objects]	**Description**
Application Manager	Overall application management, behavior and tasking such as Support for multiple simultaneous applications, inter-application communication and application signaling
Capabilities	Access to the terminal’s capability description
Configuration	Access to device configuration and user settings.
Remote Management	Access device diagnostics and perform remote management including trigerring upgrades
DRM	Integration with video and audio objects for communication with content protection systems and acquisition of content rights for protected content
CoD Manager	Control of presentation of unicast media, recordings and downloaded media items, including trick play as well as access to catalogues of on-demand content
Download Manager	Basic initiation of media download for protected and unprotected content as well as management of the media download queue and downloaded items
Scheduled Content	Control of broadcast video presentation including trick play & time-shifting & synchronization of applications to video
Parental Control Manager	Control of the parental control functionality in the receiver & PIN management for access control
Channel List Management	Discovery and management of channel lists and favorite lists, including channel scanning
Recording Management	Scheduling of local and network recordings and storage and retrieval of bookmarks
Metadata Management	Support for searching program guide information & VoD content catalogues

EXTENSION FOR PLAYBACK

```
interface {
        const COMPONENT_TYPE_VIDEO = 0;
        const COMPONENT_TYPE_AUDIO = 1;
        const COMPONENT_TYPE_SUBTITLE = 2;
        function onSelectedComponentChanged( Integer componentType );
        AVComponentCollection getComponents( Integer componentType );
        AVComponentCollection getCurrentActiveComponents( Integer componentType );
        void selectComponent( AVComponent component );
        void unselectComponent( AVComponent component );
}
interface AVComponent {
    readonly Integer componentTag;
    readonly Integer pid;
    readonly Integer type;
    readonly String encoding;
    readonly Boolean encrypted;
};
interface AVVideoComponent : AVComponent {
    readonly attribute float aspectRatio;
};
interface AVAudioComponent : AVComponent {
    readonly String language;
    readonly Boolean audioDescription;
    readonly Integer audioChannels;
};
interface AVSubtitleComponent : AVComponent {
    readonly String language;
    readonly Boolean hearingImpaired;
};
typedef Collection<AVComponent> AVComponentCollection
```

AVComponentCollection **getComponents()**

void **selectComponent()**

function **onSelectedComponentChanged()**

EXTENSION AV CONTROL FOR BUFFERING

```
interface {
    function onReadyToPlay() ;
    Boolean readyToPlay ;
    function onRepresentationChange(Integer bandwidth, Integer position) ;
    function onPeriodChange(Integer bandwidths[], Integer position) ;
    readonly Integer availableRepresentationsBandwidth [ ] ;
    readonly Integer currentRepresentation ;
    readonly Integer maxRepresentation ;
    readonly Integer minRepresentation ;
    readonly StringCollection supportedStrategies ;
    Integer getAvailablePlayTime( Boolean fromPlayPosition ) ;
    Boolean setBufferingStrategy( String name ) ;
    Boolean setRepresentationStrategy(Integer maxBandwidth, Integer
      minBandwidth, Integer position) ;
}
```

readonly Integer **availableRepresentationsBandwidth** [ ]

function **onRepresentationChange**(Integer bandwidth, Integer position)

function **onPeriodChange**(Integer bandwidths[], Integer position)

CONTENT SERVICE PROTECTION

interface {

function **onDRMMessageResult**(String msgID, String resultMsg, Integer resultCode) ;

function **onDRMSystemStatusChange**(String DRMSystemID) ;

function **onDRMSystemMessage**(String msg, String DRMSystemID) ;

String **sendDRMMessage**(String msgType, String msg, String DRMSystemID) ;

Integer **DRMSystemStatus**(String DRMSystemID) ;

}

Integer **DRMSystemStatus**(String DRMSystemID)

String **sendDRMMessage**(String msgType, String msg, String DRMSystemID)

function **onDRMMessageResult**(String msgID, String resultMsg
Integer resultCode) ;

CONFIGURATION

```
interface LocalSystem {
        const OFF = 0;
        const ON = 1;
        const PASSIVE_STANDBY = 2;
        const ACTIVE_STANDBY  = 2;
        const PASSIVE_STANDBY_HIBERNATE = 2;
        readonly String deviceID ;
        readonly Boolean systemReady ;
        readonly String vendorName ;
        readonly String modelName ;
        readonly String softwareVersion
        readonly
        readonly
        readonly
        readonly
        readonly
        readonly
        readonly
        readonly Boolean [illegible]Enabled ;
        readonly Integer powerState
        readonly Integer previousPowerState ;
        readonly Integer timeCurrentPowerState ;
        function onPowerStateChange( Integer powerState ) ;
        Integer volume ;
        Boolean mute ;
        readonly AVOutputCollection outputs ;
        readonly NetworkInterfaceCollection networkInterfaces ;
        readonly TunerCollection tuners ;
        readonly Integer tvStandard ;
        readonly Integer pvrSupport ;
        Boolean setScreenSize( Integer width, Integer height ) ;
        Integer setPvrSupport( Integer state ) ;
        Boolean setPowerState( Integer type ) ;
        Boolean setDigestCredentials( String protocol, String domain, String username, String password ) ;
        Boolean clearDigrstCredentials( String protocol, String domain ) ;
}
```

readonly String **deviceID** ;

readonly AVOutputCollection **outputs** ;

Boolean **setDigestCredentials**(String protocol, String domain, String username, String password) ;

Boolean **clearDigrstCredentials**(String protocol, String domain) ;

GENERAL VIDEO

	A/V Control Object	Broadcast object	HTML5 IDL attributes	Comments
General	Number width	Integer width	video.videoWidth	
	Number height	Integer height	video.videoHeight	
	readonly Boolean fullScreen	readonly Boolean fullScreen	NS	Not in HTML5 because of security issues
	setFullScreen (Boolean fullscreen)	void setFullScreen(Boolean fullscreen)	NS	Not in HTML5 because of security issues
	focus ()		window.focus()	
	Object onfocus	function onfocus	onfocus	
	Object onblur	function onblur	onblur	
	Object onFullScreenChange	function onFullScreenChange	NS	Not in HTML5 because of security issues

COMPONENT CONTROL

	A/V Control Object	Broadcast object	HTML5 IDL attributes	Comments
Volume	Boolean setVolume(Number volume)	Boolean setVolume(Integer volume)	float media.volume	The HTML5 value is in a range between 0 and 1, whereas the DAE visual objects are between 0 and 100
			boolean media.muted	
			boolean media.controls	true if the user agent should provide its own set of controls
			onvolumechange	
		Integer getVolume()	float media.volume	
Components (ex. subtitles, languages)	AVComponentCollection getComponents(Integer componentType)	AVComponentCollection getComponents(Integer componentType)	NS	Subtitles and media annotations not currently in HTML5 (but proposals exist)
	AVComponentCollection getCurrentActiveComponents(Integer componentType)	AVComponentCollection getCurrentActiveComponents(Integer componentType)	NS	
	void selectComponent(AVComponent component)	void selectComponent(AVComponent component)	NS	
	void unselectComponent(AVComponent component)	void unselectComponent(AVComponent component)	NS	

PLAYBACK CONTROL

	A/V Control Object	Broadcast object	HTML5 IDL attributes	Comments
Playback control	String data			video.url
	readonly Number playPosition	readonly Integer playPosition	attribute float currentTime; (get)	
			readonly attribute float startTime;	
	readonly Number playTime		readonly attribute float duration	
	readonly Number playState		readonly attribute boolean paused; readonly attribute boolean ended;	
	readonly Number error		attribute int media.error and error / abort events	

	readonly Number speed	readonly Number playSpeed	attribute float defaultPlaybackRate;
			attribute float playbackRate;	

	Boolean play (Number speed)	Boolean resume()
		Boolean pause()
			void play();
			void pause();
			attribute boolean autoplay;
			attribute boolean loop;
				Recording aspects not covered

		Boolean setSpeed(Number speed)	attribute float playbackRate;	
	Boolean stop ()	void stopRecording()	NA (no recording support)	Stop functionality can be implemented with pause();currentTime=0;

PLAYBACK CONTROL CONT

	A/V Control Object	Broadcast object	HTML5 IDL attributes	Comments
Playback control		Boolean stopTimeshift()	NA (no recording support)	
	Boolean seek (Number pos)	Boolean seek(Integer offset, Integer reference)	attribute float currentTime; (set)	The HTML5 values are in seconds, whereas the DAE values are in milliseconds.
	function onPlaySpeedChanged(Number speed)	function onPlaySpeedChanged(Number speed)	events: ratechange durationchange	
	script onPlayPositionChanged(Integer position)	function onPlayPositionChanged(Integer position)	event: timeupdate	
	readonly Number playSpeeds[ ]	readonly Number playSpeeds[ ]	NS	

NO MAPPING IN W3C

- Broadcast control
  - Channel change, bind to current channel
  - Activate circular buffer

- Recording
  - Record now
  - Offset
  - Recording object

REACHING EXPECTATIONS

The following points are a list of different directions that could be taken for ensuring that the integration of Web and TV reach the expectations of all concerned parties.

1. Profiling of the web related standards to avoid duplications of methods.
   - Work done by CEA-2014, OIPF, HbbTV
2. Establish performance requirements. Simply having support of the latest standard does not ensure that the platform has the proper performance.
   - Certification process

W3C COLLABORATIONS

The following points are a list of possible areas of W3C collaboration with the OIPF.

1. Align the <video> tag for both HTML5 and SVG video control.
   – Not available when CEA-2014 selected which is base for DAE
2. Define clear boundaries for other standard forums to extend W3C standards where it is necessary.
   – Need to discuss how the boundary may look
3. Create a new interface for the control of play out of time-shifted content.
   – Jointly create an interface

ERICSSON