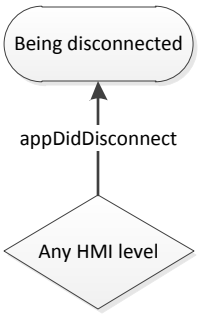
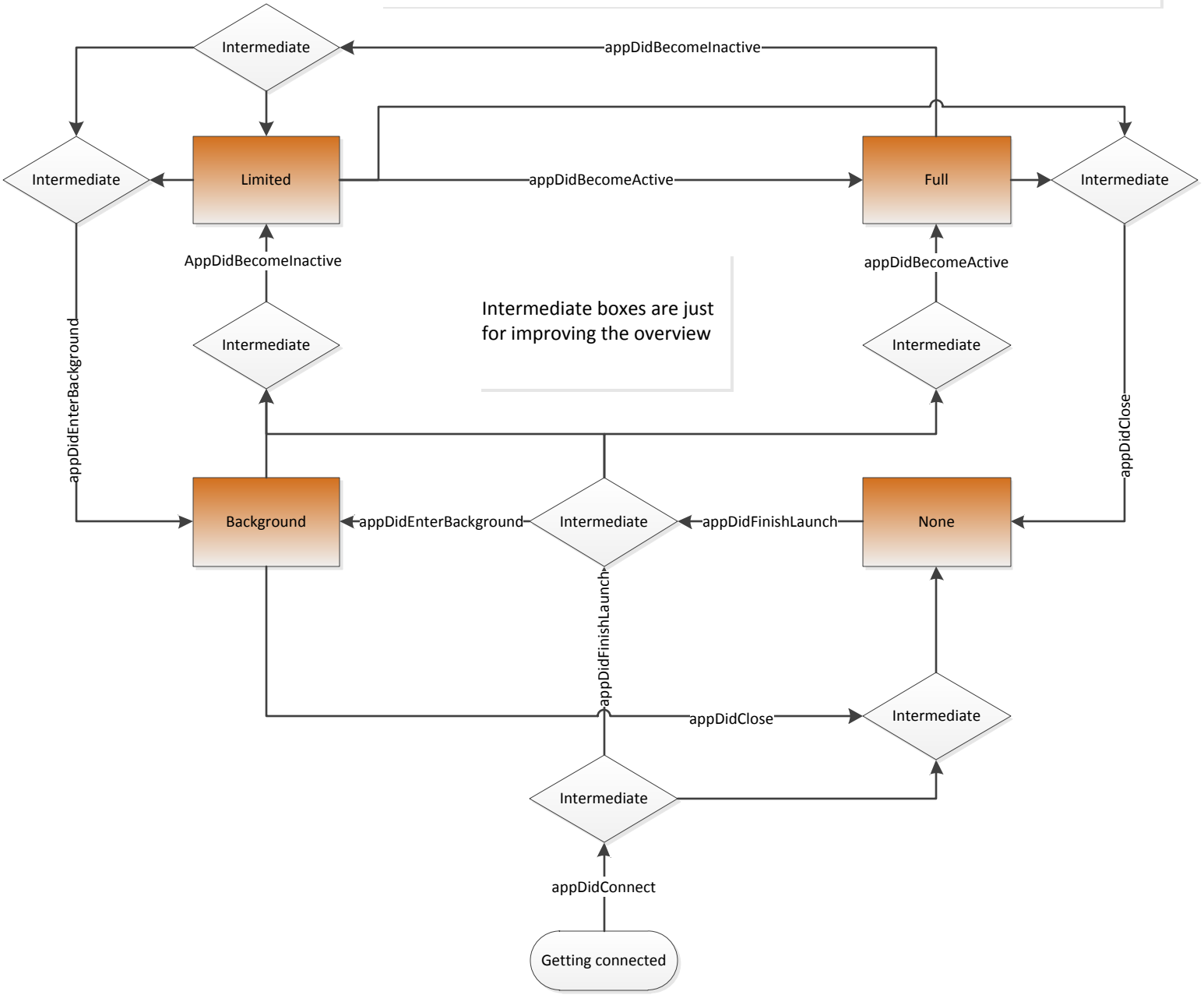
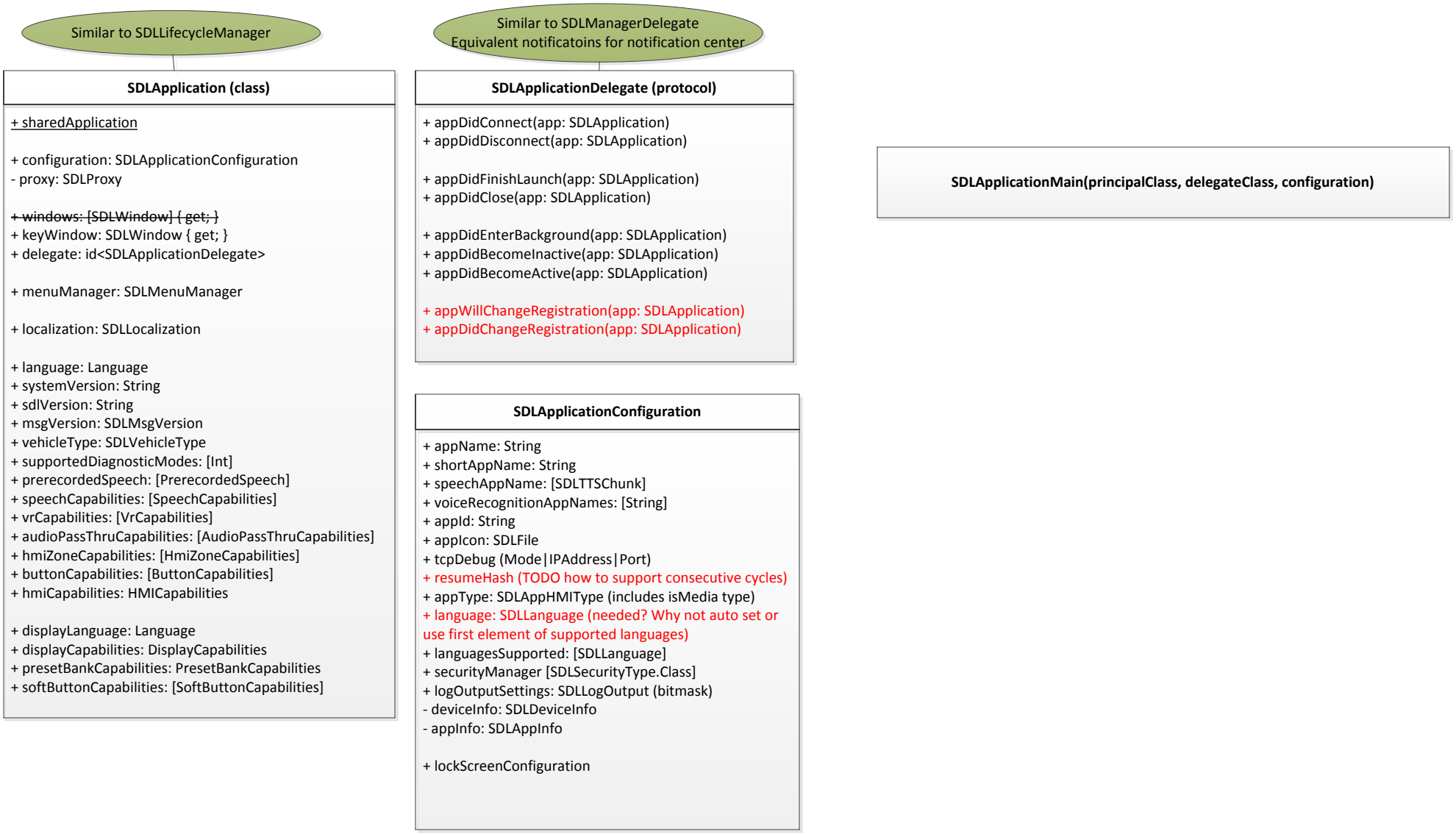


iOS: (no equivalent exist) -> SDL:appDidConnect
(telling the app it got connected and registered)
iOS: (no equivalent exist) -> SDL:appDidDisconnect
(telling the app it got disconnected or unregistered)

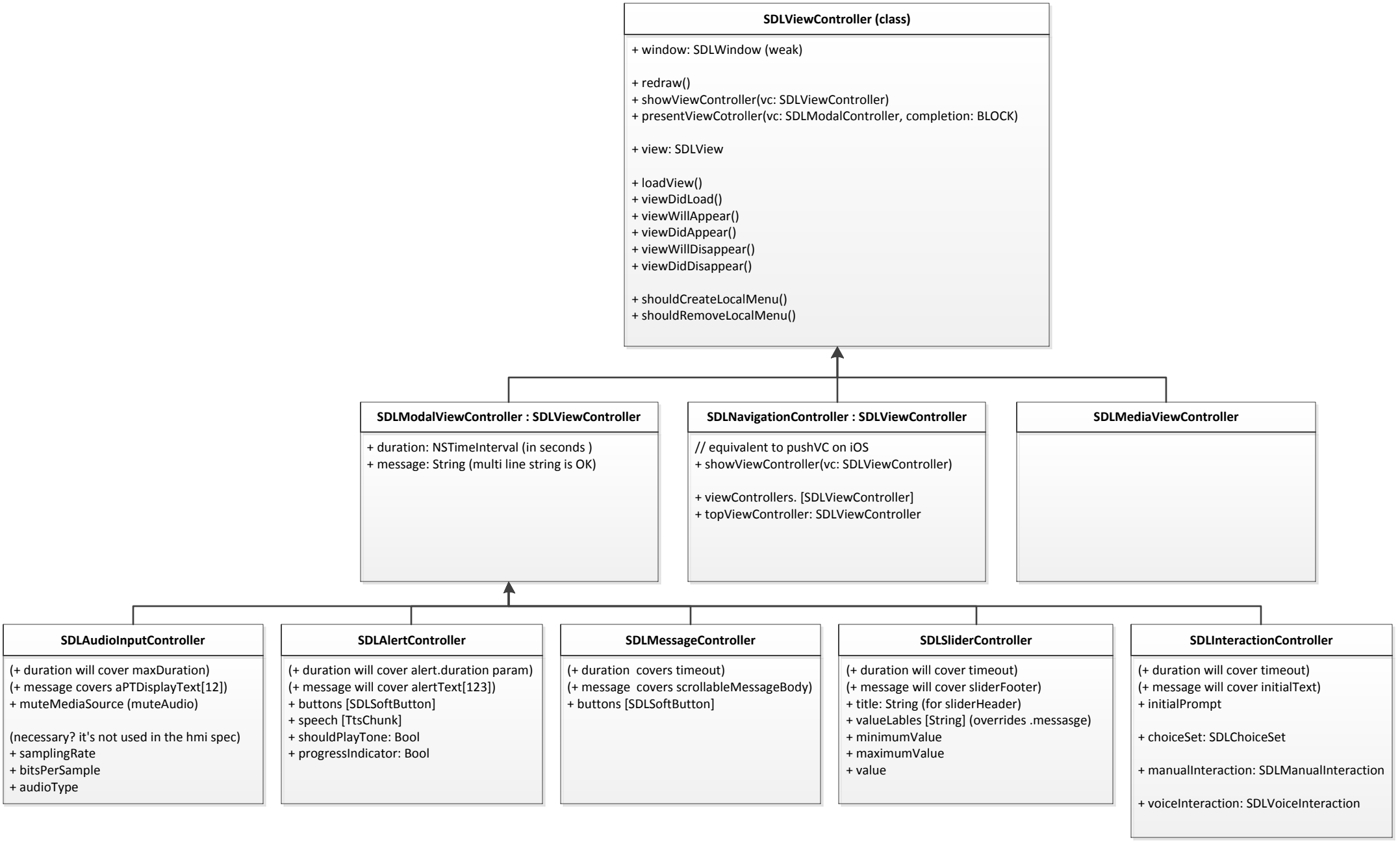
~~iOS:applicationWillFinishLaunchWithOptions -> SDL: (no equivalent planned)~~
~~(Can be added between onProxyOpened and RegisterAppInterface)~~
~~(Used hardly ever in the wild therefore not planned)~~
iOS:applicationDidFinishLaunchWithOptions -> SDL:appDidFinishLaunch
(called on first transition to any HMI level other than NONE)
iOS:applicationWillTerminate -> SDL: appDidClose
(HMI NONE can be treated as an unlaunched app)
(also called when getting connected and immediately entering any level other than NONE)

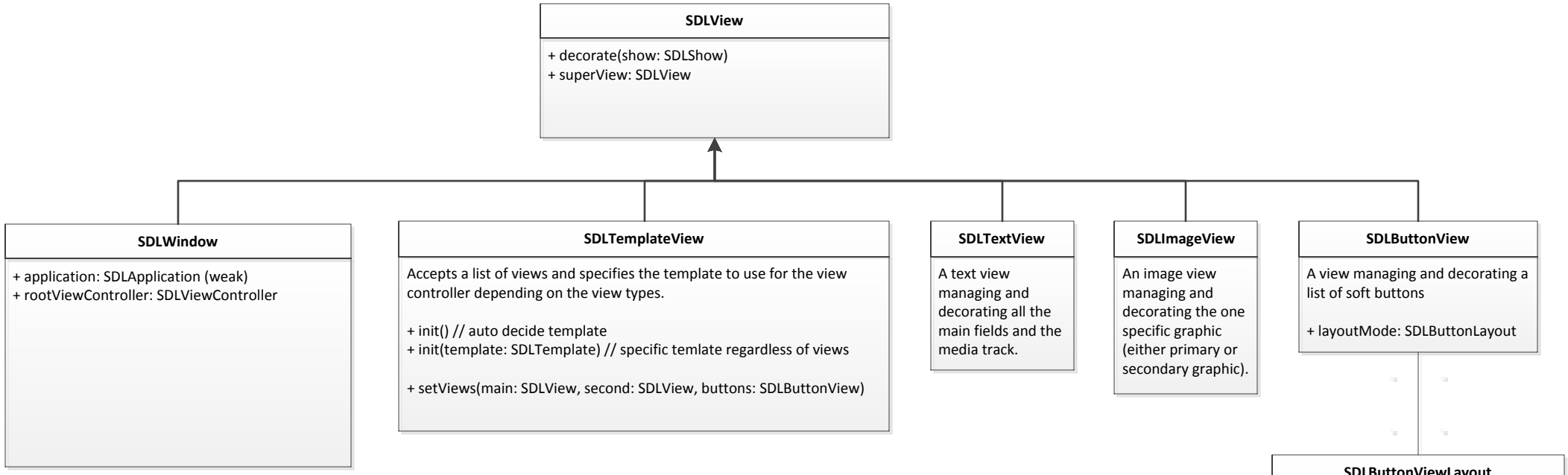
iOS:applicationDidEnterBackground -> SDL:appDidEnterBackground
(transition to HMI BACKGROUND)
iOS:applicationWillEnterForeground -> SDL:appDidBecomeInactive
iOS:applicationWillResignActive -> SDL:appDidBecomeInactive
(transition to HMI LIMITED)
(similar to the state on iOS between foreground and background)
iOS:applicationDidBecomeActive -> SDL:appDidBecomeActive
(transition to HMI FULL, the real foreground state)




Similar to SDLLifecycleManager
Similar to SDLManagerDelegate
Equivalent notificatoins for notification center
SDLApplication (class)
+ sharedApplication
+ configuration: SDLApplicationConfiguration
- proxy: SDLProxy
~~+ windows: [SDLWindow] { get; }~~
+ keyWindow: SDLWindow { get; }
+ delegate: id<SDLApplicationDelegate>
+ menuManager: SDLMenuManager
+ localization: SDLLocalization
+ language: Language
+ systemVersion: String
+ sdlVersion: String
+ msgVersion: SDLMsgVersion
+ vehicleType: SDLVehicleType
+ supportedDiagnosticModes: [Int]
+ prerecordedSpeech: [PrerecordedSpeech]
+ speechCapabilities: [SpeechCapabilities]
+ vrCapabilities: [VrCapabilities]
+ audioPassThruCapabilities: [AudioPassThruCapabilities]
+ hmiZoneCapabilities: [HmiZoneCapabilities]
+ buttonCapabilities: [ButtonCapabilities]
+ hmiCapabilities: HMICapabilities
+ displayLanguage: Language
+ displayCapabilities: DisplayCapabilities
+ presetBankCapabilities: PresetBankCapabilities
+ softButtonCapabilities: [SoftButtonCapabilities]
SDLApplicationDelegate (protocol)
+ appDidConnect(app: SDLApplication)
+ appDidDisconnect(app: SDLApplication)
+ appDidFinishLaunch(app: SDLApplication)
+ appDidClose(app: SDLApplication)
+ appDidEnterBackground(app: SDLApplication)
+ appDidBecomeInactive(app: SDLApplication)
+ appDidBecomeActive(app: SDLApplication)
+ appWillChangeRegistration(app: SDLApplication)
+ appDidChangeRegistration(app: SDLApplication)
SDLApplicationConfiguration
+ appName: String
+ shortAppName: String
+ speechAppName: [SDLTTSChunk]
+ voiceRecognitionAppNames: [String]
+ appId: String
+ appIcon: SDLFile
+ tcpDebug (Mode|IPAddress|Port)
+ resumeHash (TODO how to support consecutive cycles)
+ appType: SDLAppHMIType (includes isMedia type)
+ language: SDLLanguage (needed? Why not auto set or use first element of supported languages)
+ languagesSupported: [SDLLanguage]
+ securityManager [SDLSecurityType.Class]
+ logOutputSettings: SDLLogOutput (bitmask)
- deviceInfo: SDLDeviceInfo
- appInfo: SDLAppInfo
+ lockScreenConfiguration
SDLApplicationMain(principalClass, delegateClass, configuration)


Start point
Create View Controller instance because the app got into foreground and the root view controller has tobe created or another view controller was pushed
Init()
View controller is about to become visible for the first time
app should create and prepare view content in the next method
loadView
View stuff is ready and next method is for anything else (e.g. data etc.)
viewDidLoad
View is about to appear (transition into this view controller starting)
App reentered foreground or
the pushed view controller got pop'd
viewWillAppear
View is now visible on the screen (transition into this view controller done)
viewDidAppear
Now the view controller is active and visible and the user is interacting with this view controller
View controller
is now active
View is about to disappear (transition into another view controller or app hmi level changed)
View controller
is now in
background
(HMI not full
or another
view controller
is in front of it)
viewWillDisappear
View is not visible anymore
App entered background level or another
view controller got pushed above it
viewDidDisappear
Not happy about this transition but can be used to indicate unloading anything because the view cotnroler instance is about to be disposed
viewDidUnload
Memory now gets free. Latest here the app should dealloc other data related to this view controller
dealloc
end


SDLViewController (class)
+ window: SDLWindow (weak)
+ redraw()
+ showViewController(vc: SDLViewController)
+ presentViewCotroller(vc: SDLModalController, completion: BLOCK)
+ view: SDLView
+ loadView()
+ viewDidLoad()
+ viewWillAppear()
+ viewDidAppear()
+ viewWillDisappear()
+ viewDidDisappear()
+ shouldCreateLocalMenu()
+ shouldRemoveLocalMenu()
SDLModalViewController : SDLViewController
+ duration: NSTimeInterval (in seconds)
+ message: String (multi line string is OK)
SDLNavigationController : SDLViewController
// equivalent to pushVC on iOS
+ showViewController(vc: SDLViewController)
+ viewControllers. [SDLViewController]
+ topViewController: SDLViewController
SDLMediaViewController
SDLAudioInputController
(+ duration will cover maxDuration)
(+ message covers aPTDisplayText[12])
+ muteMediaSource (muteAudio)
(necessary? it's not used in the hmi spec)
+ samplingRate
+ bitsPerSample
+ audioType
SDLAlertController
(+ duration will cover alert.duration param)
(+ message will cover alertText[123])
+ buttons [SDLSoftButton]
+ speech [TtsChunk]
+ shouldPlayTone: Bool
+ progressIndicator: Bool
SDLMessageController
(+ duration covers timeout)
(+ message covers scrollableMessageBody)
+ buttons [SDLSoftButton]
SDLSliderController
(+ duration will cover timeout)
(+ message will cover sliderFooter)
+ title: String (for sliderHeader)
+ valueLables [String] (overrides .messasge)
+ minimumValue
+ maximumValue
+ value
SDLInteractionController
(+ duration will cover timeout)
(+ message will cover initialText)
+ initialPrompt
+ choiceSet: SDLChoiceSet
+ manualInteraction: SDLManualInteraction
+ voiceInteraction: SDLVoiceInteraction



Layout name	Main view	Second view	Button view	Comment
DEFAULT	any	any	any	Fallback for a non media app
MEDIA	any	any	any	Always used when app registers as a media app
NON_MEDIA	any	any	any	Alternative fallback for a non media app
ONSCREEN_PRESETS	nil	nil	nil	When no view is set as a non media app?
TEXT_WITH_GRAPHIC	SDLTextView	SDLImageView	nil	
GRAPHIC_WITH_TEXT	SDLImageView	SDLTextView	nil	
TILES_ONLY	nil	nil	SDLButtonView	If button view is set to TEXT_TILE, IMAGE_TILE or DEFAULT_TILE
or	SDLButtonView	nil	nil	If button view is set to TEXT_TILE, IMAGE_TILE or DEFAULT_TILE
TEXTBUTTONS_ONLY	nil	nil	SDLButtonView	If button view is set to TEXT or DEFAULT
or	SDLButtonView	nil	nil	If button view is set to TEXT or DEFAULT
TILES_WITH_GRAPHIC	SDLButtonView	SDLImageView	nil	If button view is set to TEXT_TILE, IMAGE_TILE or DEFAULT_TILE
GRAPHIC_WITH_TILES	SDLImageView	SDLButtonView	nil	If button view is set to TEXT_TILE, IMAGE_TILE or DEFAULT_TILE
GRAPHIC_WITH_TEXT_AND_SOFTBUTTONS	SDLImageView	SDLTextView	SDLButtonView	If button view is set to TEXT, IMAGE or DEFAULT
TEXT_AND_SOFTBUTTONS_WITH_GRAPHIC	SDLTextView	SDLImageView	SDLButtonView	If button view is set to TEXT, IMAGE or DEFAULT
TEXTBUTTONS_WITH_GRAPHIC	SDLButtonView	SDLImageView	nil	If button view is set to TEXT or DEFAULT
GRAPHIC_WITH_TEXTBUTTONS	SDLImageView	SDLButtonView	nil	If button view is set to TEXT or DEFAULT
LARGE_GRAPHIC_WITH_SOFTBUTTONS	SDLImageView	nil	SDLButtonView	
DOUBLE_GRAPHIC_WITH_SOFTBUTTONS	SDLImageView	SDLImageView	SDLButtonView	
LARGE_GRAPHIC_ONLY	SDLImageView	nil	nil	
NAV_FULLSCREEN_MAP	??	??	??	
NAV_LIST	??	??	??	
NAV_KEYBOARD	??	??	??	

SDLChoiceSet:
NSMutableOrderedSet<SDLChoiceItem>

+ dynamicSet: Bool

If dynamic it will treat internally each choice Item as a separate interaction choice set. This way the app can reuse each choice item in any desired order (e.g. spotify and „recently played“ playback queues).

This can dramatically improve the performance in specific use cases and makes it extremely easy for the app developer when interacting with a choice set that receives only small but recent changes.

Limitation is that only 100 choice items can be performed within an interaction (with some rare exceptions).

It not dynamic this choice set can contain > 100 items which are divided internally to match API limitations

SDLChoiceSetManager

- choiceSets: [Int: SDLChoiceSetInternal]

+ createChoiceSet(SDLChoiceSet)
Creates one or more choice sets but with respect to the existing ones (or those that are being created).

SDLChoiceSetInternal

+ choiceSet: [Int: String]
+ status: (Creating or Created)

The first int is the choice set id that contains multiple choices. Each choice is described by the unique id and name.

A choice can occur multiple times in different sets. Therefore they will have different choice ids.

SDLChoiceItem

+ choiceName: String (internal identifier)
+ text
+ detailedText
+ rightHandText
+ voiceCommands: [String]
+ image: SDLImage
+ rightHandImage: SDLImage

SDLManualInteraction

+ layout: SDLInteractionLayout

+ keyboardProperties: SDLKeyboardProperties

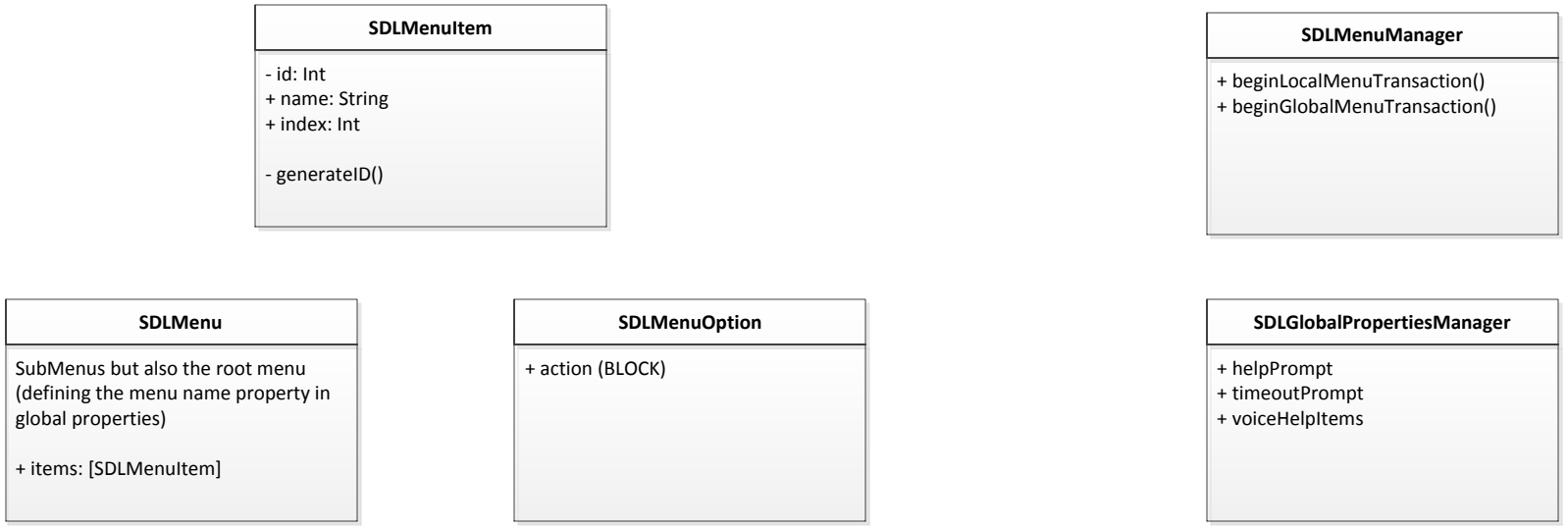
SDLInteractionLayout

+ LIST
+ TILE
+ KeyboardOnly

SDLVoiceInteraction

+ helpPrompt
+ timeoutPrompt
+ helpItems

Work in progress



Work in progress