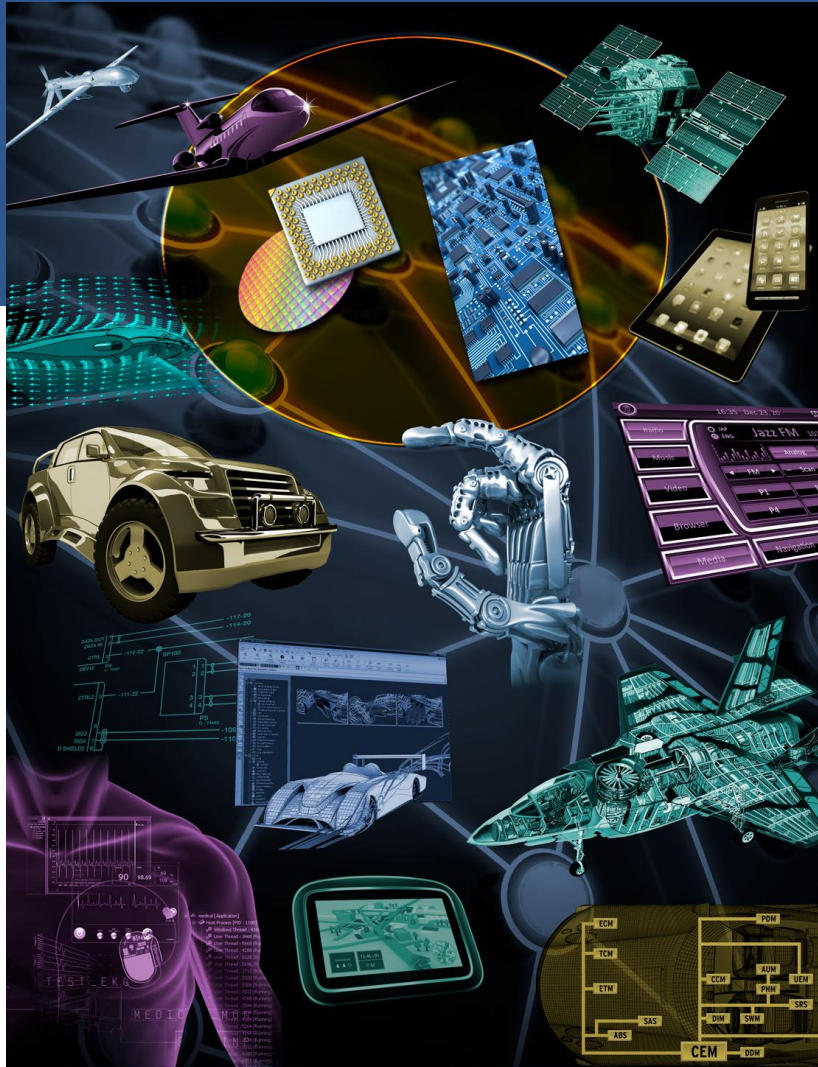


# WINDOW MANAGER

AGL F2F Technical discussion  
San Jose

July 11, 2017



# Agenda

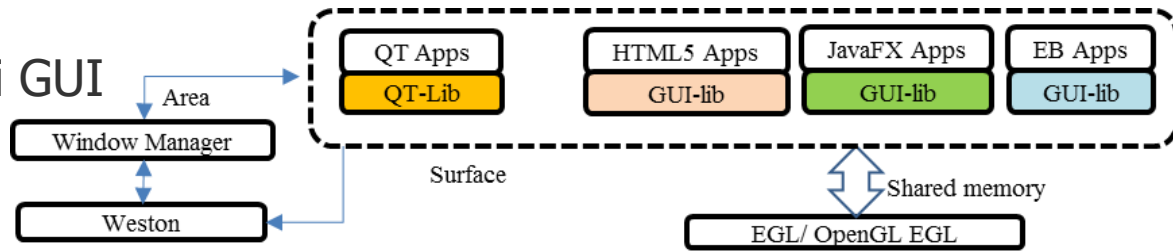
---

- Window Manager
  - Requirements
  - Plan
  
- Window manager Features
  - Overview architecture
  - Internal components
  - Features
  - Use cases (sample)
  - Sequence chart (draft)
  
- Window manager APIs (draft)

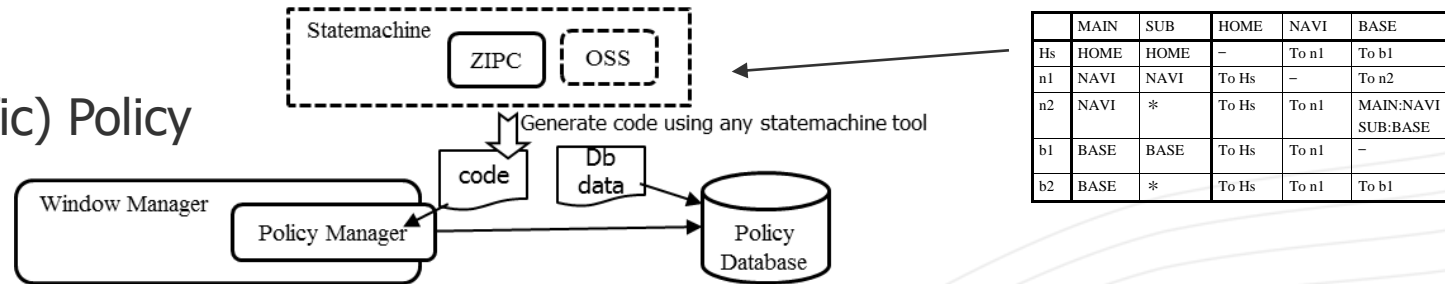
# Window Manager

## Requirements

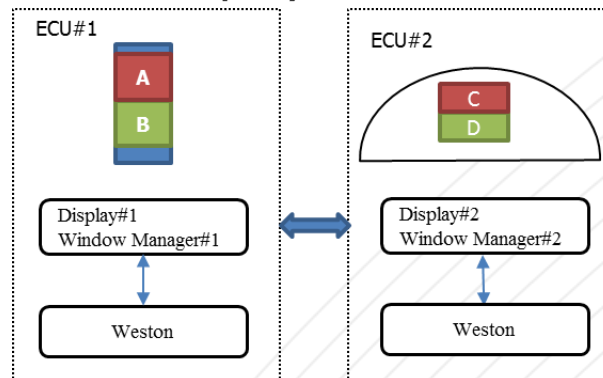
- To support multi applications and multi GUI (e.g., Qt, HTML5, JavaFX, EB)



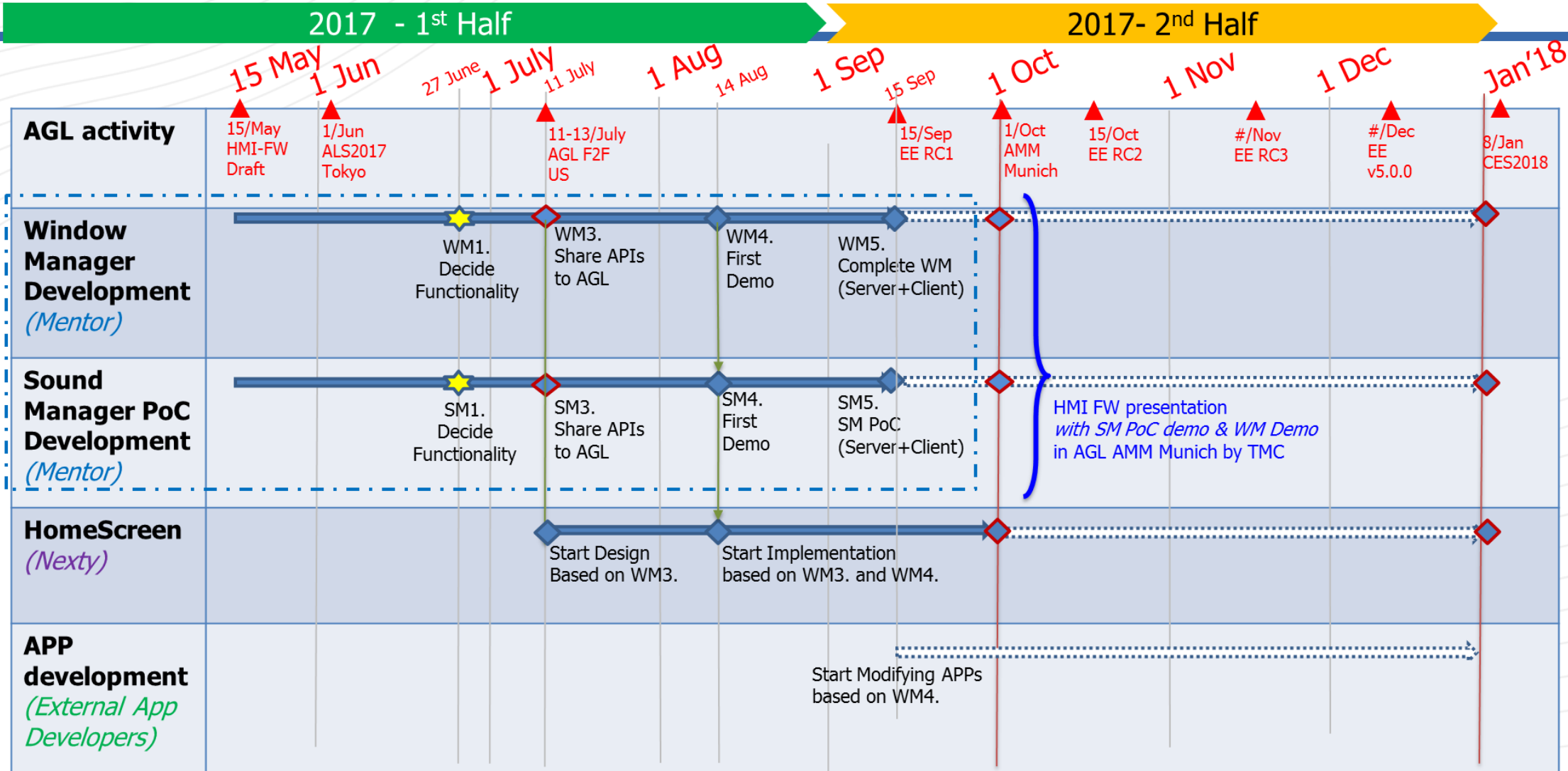
- To support (OEM specific) Policy



- To support multi ECU and Multi Display



# AGL HMI FW development – TMC CY2017 roadmap

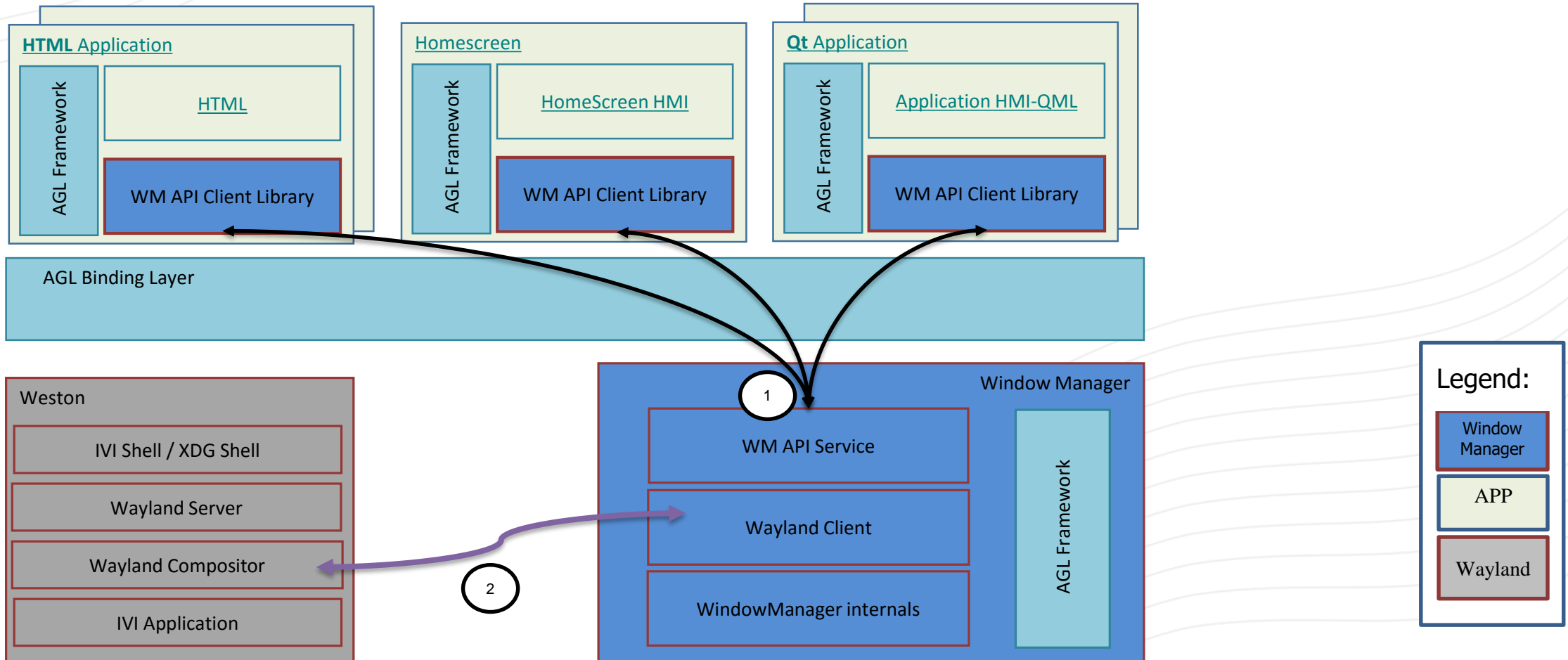


Legend: ◆ Critical Milestones ◆ General Milestones ★ Completed ⋯ TBD → Dependency

# **WINDOW MANAGER FEATURES**

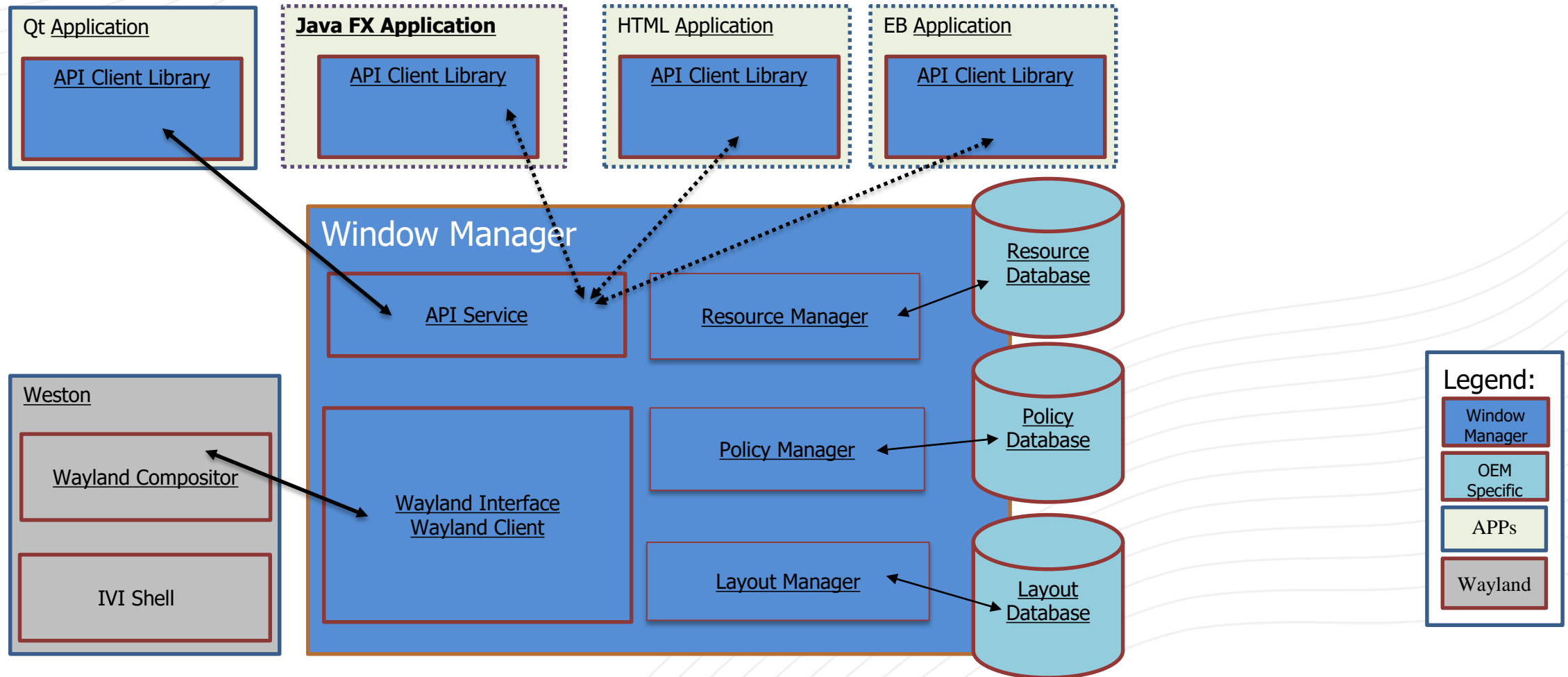
# Window Manager – Overview 1/3

## Architecture

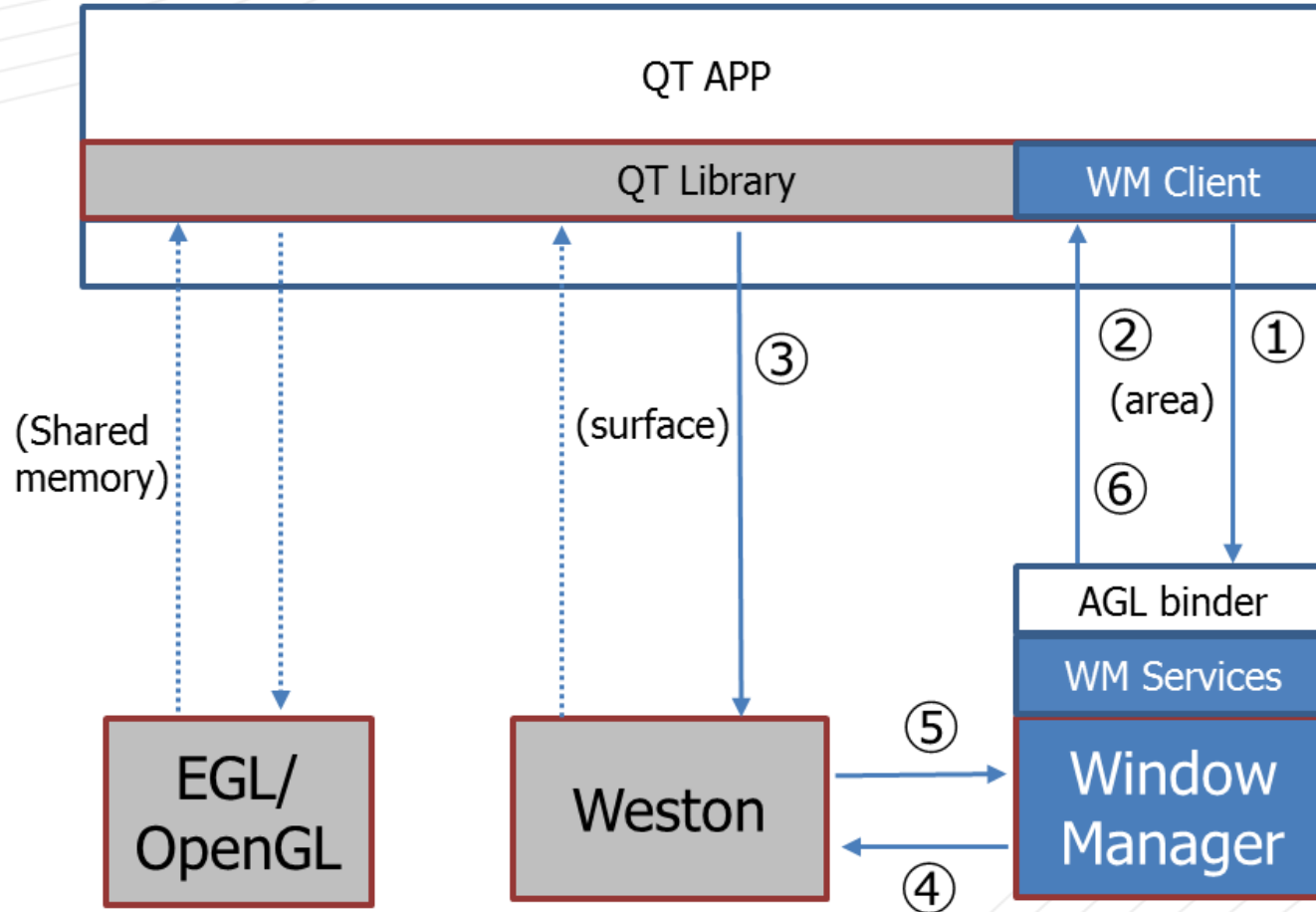


# Window Manager – Overview 2/3

## Internal Components and External dependencies



# Window Manager – Overview 2/3



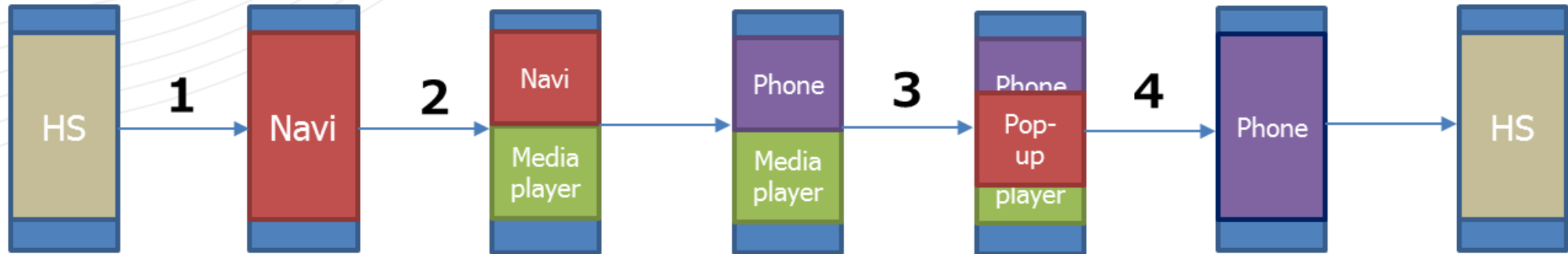
## API References:

- ① **APP ⇒ WM**  
request to generate surface-id
- ② **WM ⇒ APP**  
WM returns surface-id to APP
- ③ **APP ⇒ Weston**  
create surface
- ④ **WM ⇒ Weston**  
register surface to be able to control it later  
`registerSurface(surfaceID),`  
`configure(surfaceID, geometry)`
- ⑤ **Weston ⇒ WM**  
acknowledge
- ⑥ **WM ⇒ APP**  
Visible, Invisible, Active, Inactive

※ Agreed with The QT Company



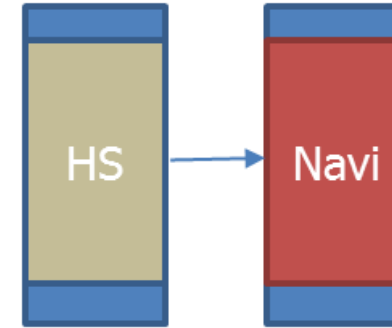
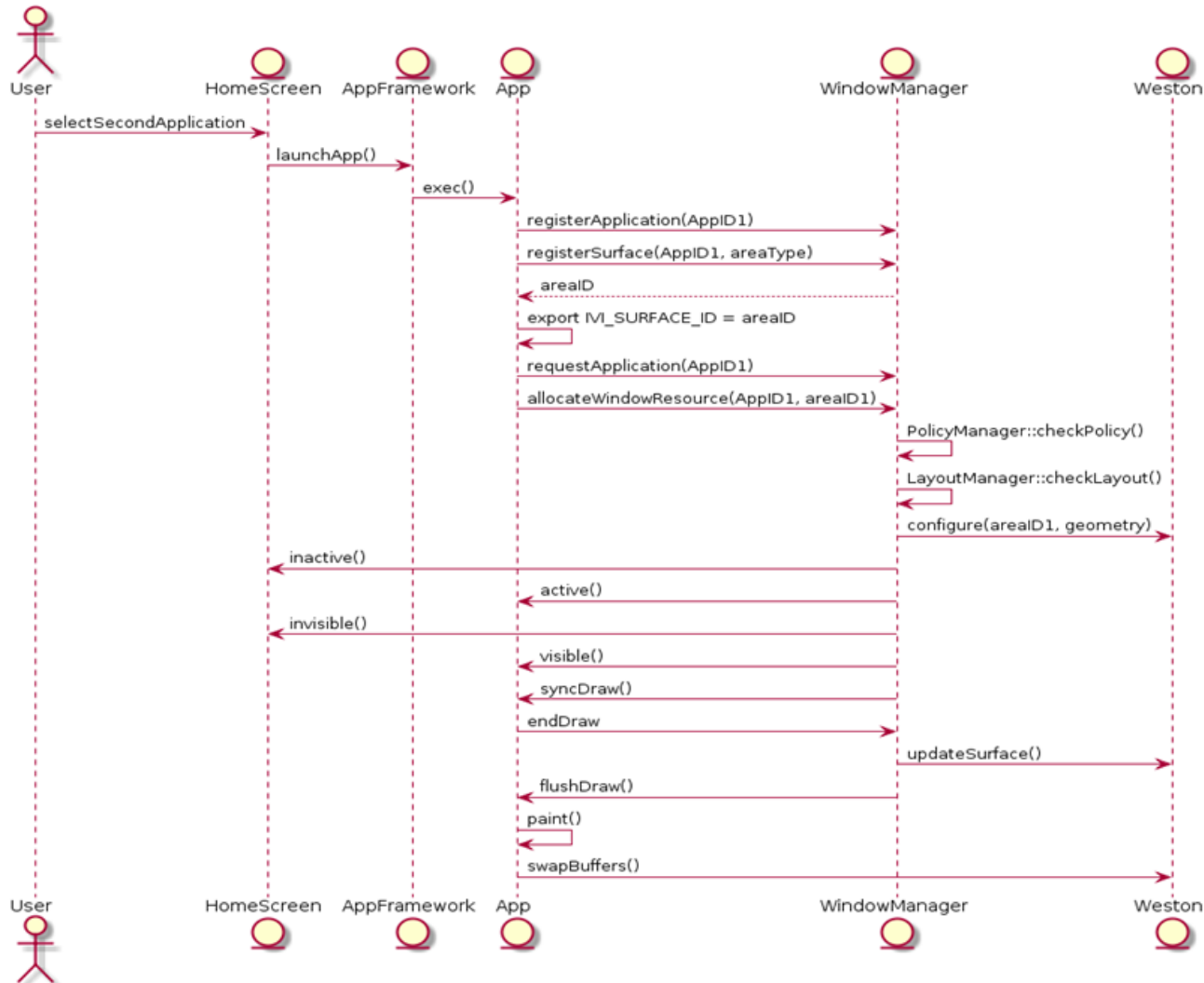
# Window Manager – Sample Use Cases



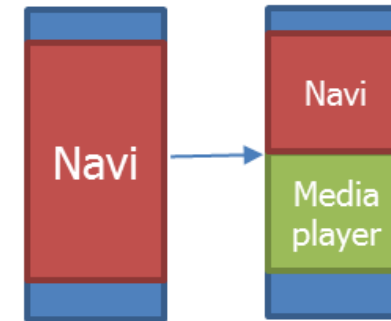
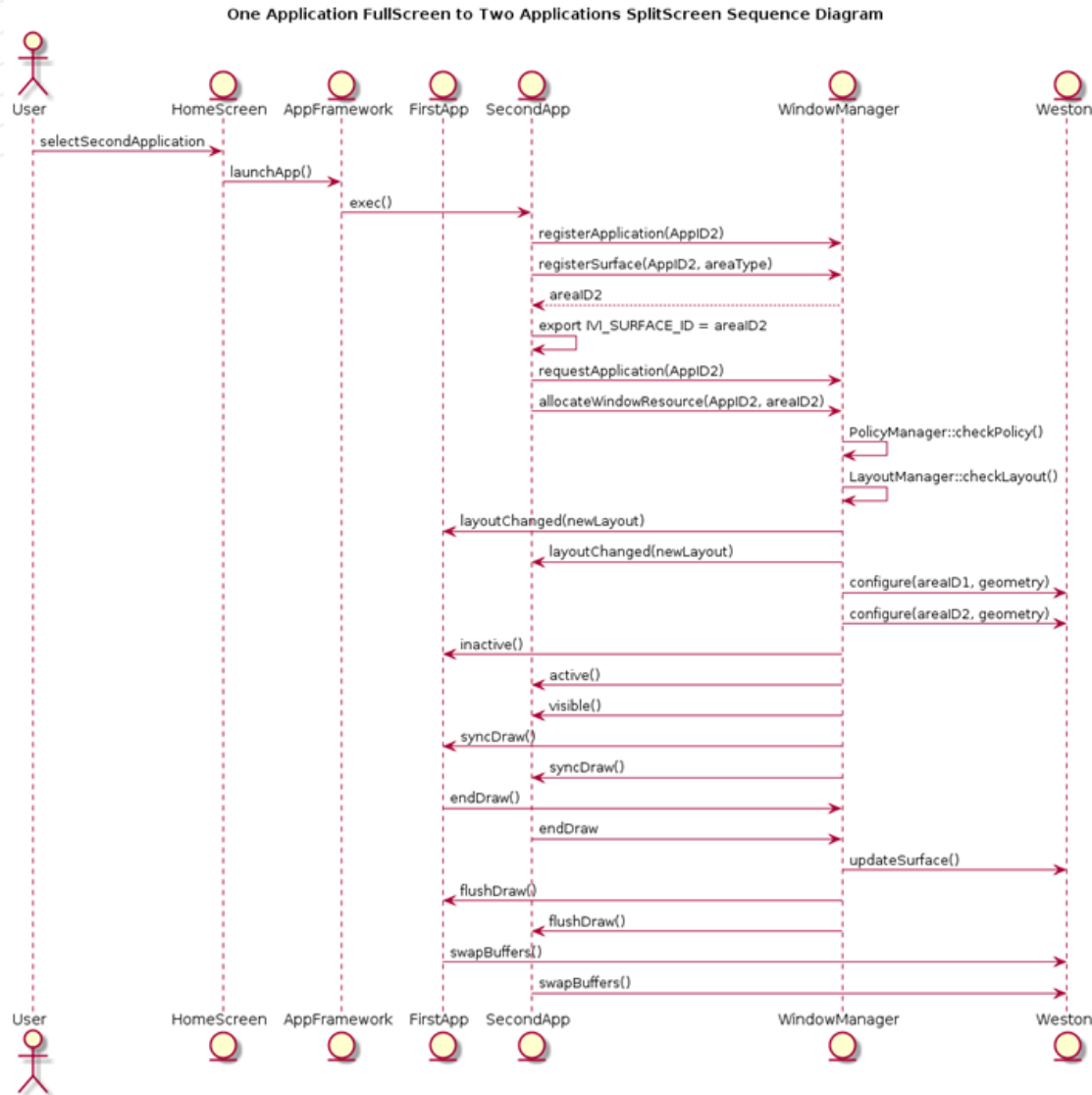
1. Full-screen -> Full-screen
2. Full-screen -> Split-screen
3. Split-screen -> Pop-up (on-screen)
4. Split-screen -> Full-screen

# Window Manager – Sample Use Cases 1/4

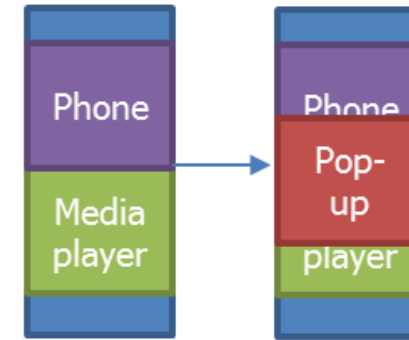
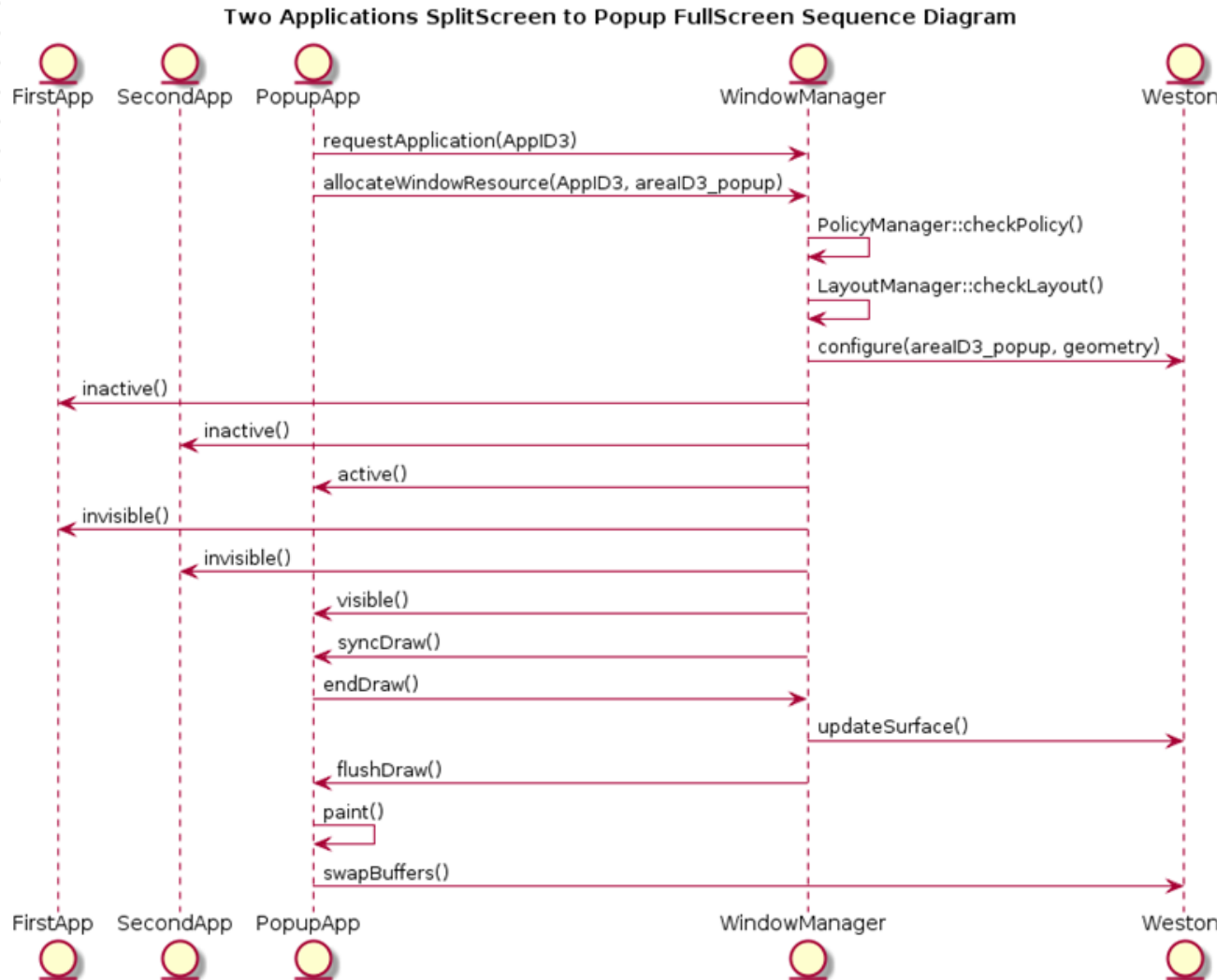
HomeScreen to Application FullScreen Sequence Diagram



# Window Manager – Sample Use Cases 2/4

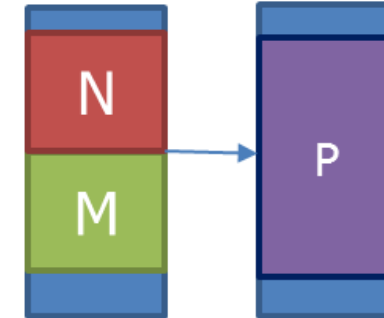
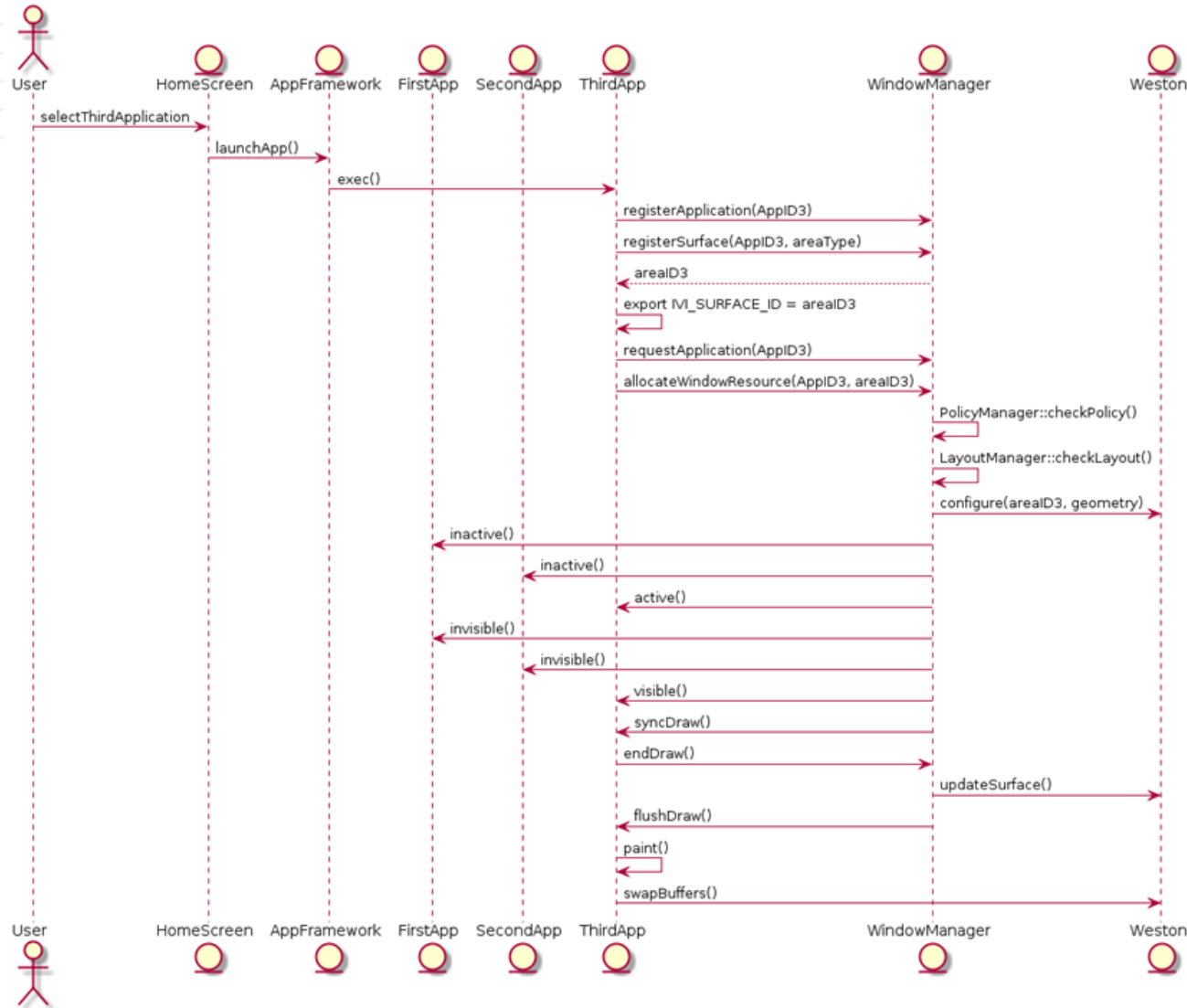


# Window Manager – Sample Use Cases 3/4



# Window Manager – Sample Use Cases 4/4

Two Applications SplitScreen to One Application FullScreen Sequence Diagram



# Window Manager – APIs

S. No.	API	Description
1	bool registerApplication(int appid)	Register one application with id appid, tied to the PID determined by the connection.
2	int areaid registerSurface(int appid, surface_type type)	Provide the areaid to an application to then create its owned surface with id areaid and type. Type can be something like default (generic surface, no special handling) or popup (which must be placed on a popup-layer).
3	bool allocateWindowResources(int appid, int areaid)	Activate the areaid application surface, i.e. make it visible in its assigned area according to layout
4	bool deallocateWindowResources(int appid, int areaid)	Deactivate an application surface, based on the areaid
5	void endDraw(int areaid)	Rendering Completed
6	bool requestApplication(int appid)	Request to window manager for application with appid to be displayed, this might trigger a layout change, depending on policies and current layout.

# Window Manager – APIs

S. No.	API	Description
1	<i>void visible(int areaid)</i>	Notify that area areaid has become visible
2	<i>void invisible(int areaid)</i>	Notify that area areaid has become invisible
3	<i>void active(int areaid)</i>	Notify that an area areaid is currently active (has focus)
4	<i>void inactive(int areaid)</i>	Notify that an area areaid is currently inactive (has lost focus)
5	<i>void layoutChanged(int areaid, layout newlayout)</i>	Notify a client area that its layout changed
6	<i>void syncDraw(int areaid)</i>	Redraw area after layout change
7	<i>void flushDraw(int areaid)</i>	Notify client that it should swap buffers (after SyncDraw)
8	<i>void endDraw(int areaid)</i>	<i>Notify client that the drawing process ended after swap buffers</i>
9	<i>void popupTimedOut(int areaid)</i>	Notify a client, that its popup surface timeout is expired

**THANK YOU**



# Mentor<sup>®</sup>

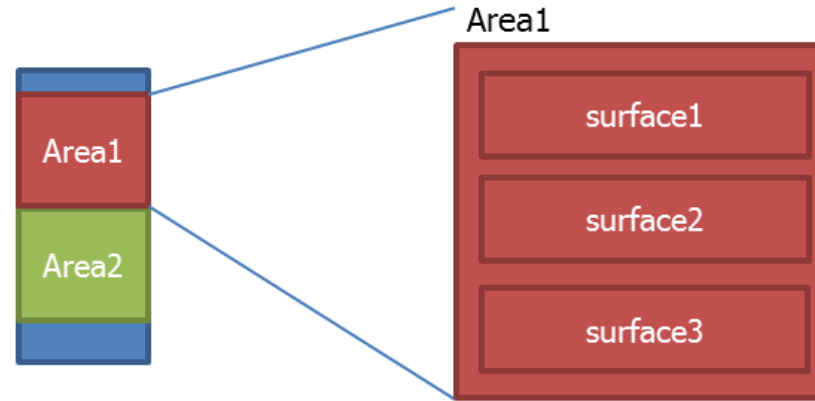
A Siemens Business

[www.mentor.com](http://www.mentor.com)

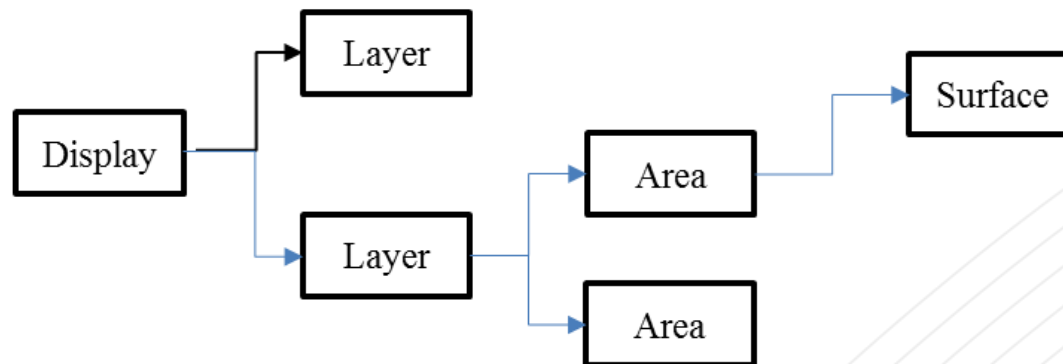
# REFERENCES

# Window Manager

## 1. Area Vs Surface



## 2. Resource Management



# Window Manager

- Visible
- Invisible
- Active
- Inactive

