webOS LSM for Automotive

Jaeyoon Jung
LG Electronics
LSM: Luna Surface Manager

- Window management of application surfaces and System UI
- Input event routing from various devices
- Virtual Keyboard
- Qt/QML Wayland compositor
LSM Architecture

Control
Rendering data
Bus

webOS components
Luna bus

WAM
Qt/QML Native
Others

Wayland adaptation
Wayland protocol
LSM QML Wayland Compositor
QPA
DRM / KMS / EVDEV
Wayland EGL
LSM Core

- **QML Wayland Compositor**
  - QML application that handles Wayland surfaces
  - Renders system UI elements and Wayland surfaces together
- **QPA**
  - Platform abstraction plugin in Qt
  - Depends on hardware
Wayland and Hardware Integration

- **Wayland**
  - Display server and client protocol
  - EGL, OpenGL

- **Wayland EGL**
  - EGL layer for buffer sharing between client and server

- **Graphics / Input HW**
  - DRM/KMS or SoC specific
  - Evdev or proprietary input stack
Client Integration

- Various Wayland clients
  - WAM
    - Web app manager
    - Provides Wayland integration for web applications
  - Qt/QML
    - Runs on top of Qtwayland client plugin
  - Others
    - SDL native clients
    - Clients compatible with standard wayland protocol can run
LSM for Various Products

• Challenges
  – Various GPUs and input devices
  – Extend Wayland protocol while keeping compatibility to standard protocols
  – Commonize core compositor features and modularize extensions
  – Differentiate user interfaces
Various GPUs and input devices

- Use different QPAs
  - Proprietary QPA for LG in-house SoC
  - Open-sourced QPA
    - eg. EGLFS for DRM-based platforms

- Abstraction for Input devices
  - webOS proprietary input device stack
  - Evdev for EGLFS
Wayland Protocol Extension

• **WebOSCompositorExtension**
  – Provides Qt plug-in interfaces to Wayland protocol extension
  – A plug-in that implements a particular Wayland protocol extension can be loaded as needed
  – Defined as a Qt module
    QT += weboscompositorextension
• Compositor Base Architecture
  – Defines the baseline of webOS compositor and common System UI that can be extended or replaced by a product extension as needed
  – Key architecture
    • Plug-in interface for native core compositor
    • QML module interface that overrides the baseline QML views, controllers and types.
Core Compositor and Extensions (2/2)

- **webOS Compositor Plug-in Interface**
  - Plug-in for core class overrides
  - Extend WebOSCoreCompositor and WebOSCompositorWindow so that it can re-implement methods in a product-specific way

- **QML Compositor base and Extension**
  - WebOSCompositorBase
    - Defines the baseline of QML scene based on the common webOS UX
    - Provides QML types that can be re-defined by a product extension
  - WebOSCompositor
    - Implements product-specific layout of views
    - Re-defines controllers for views including new ones
    - Re-uses types in the baseline unless re-defined
webOS for Automotive

• Automotive Experiences
  – Cluster
  – In-Vehicle Infotainment in Center stack
  – Rear Seat Entertainment

• Specialties of webOS
  – Multiple display supports
  – Quick prototyping of window layouts and system UIs
  – DRM-enabled media stack
  – Integration with Content provider applications

➢ IVI and RSE
LSM for IVI and RSE

• Requirements
  – Multiple displays
  – Touchscreen per display
  – Integration with various applications
  – Customizable window layout and system UI
Multiple Displays

- **Hardware interface**
  - DRM with multiple connectors

- **QPA**
  - Platform screens/windows per display
    - DRM connector – Platform screen – Platform window

- **Compositor**
  - Has multiple compositor windows per display
    - Platform window – compositor window
  - One compositor window covers the screen of the display associated
Touchscreen per Display

- Hardware interface
  - /dev/input/eventX

- QPA
  - Delivers touch inputs to an associated window
    - Match touchscreen devices to each window
    - Send an event in a local coordinate of the target window

- Compositor
  - Handler events per window
Apps and System UI

• Integration with Apps
  – Supports various application types
  – Media integration
    • Punch-through
    • Textured video

• Customizable window layout and system UI
  – Custom WebOSCompositor implementation as per spec
  – Reuse common features and UI elements from the compositor base
Prototype: LSM for IVI
Prototype: LSM for RSE
webOS is Linux based and open-sourced platform constructed on the flexible and extensible architecture.

LSM is the Wayland compositor and window manager in webOS that inherits extensibility.

webOS and LSM have specialties for automotive experiences such as IVI and RSE.
Find more in
http://webosose.org
https://github.com/webosose