



Leveraging webOS Technologies for Automotive



Lokesh Kumar Goel / Steve Lemke



Agenda

- LG Silicon Valley Lab (SVL) Background
- LG SVL AGL Demo at CES 2018
- webOS Open Source Edition (OSE)
- webOS Web App Runtime
- Enact Web App Framework

LG Silicon Valley Lab (SVL)

LG Silicon Valley Lab Background

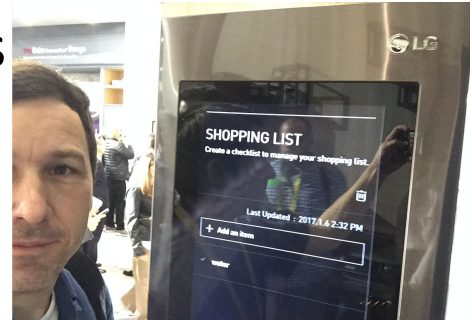
It all started with Palm Pilot...



- 1996-2008: Palm OS and Palm Pilot; Treo Smartphone
- 2009: Palm launched webOS & Palm Pre @ CES
- 2010: HP acquired Palm and webOS platform
- 2011: HP released TouchPad tablet, then stopped hw
- 2012: HP open-sourced webOS as Open WebOS

LG Silicon Valley Lab Background

- 2012: HP and LG ported webOS to TV (POC)
- 2013: LG acquired webOS sw and team
- 2014: LG launched webOS Smart TV Platform @ CES
- 2015: LG webOS reference platform
- 2016: Whiteboards, Fridges, and Automobiles
- 2017: Automotive Grade Linux
- 2018: LG open-sources webos OSE



LG Silicon Valley Lab Background



AUTOMOTIVE
GRADE **LINUX**

SVL investigations with AGL last year:

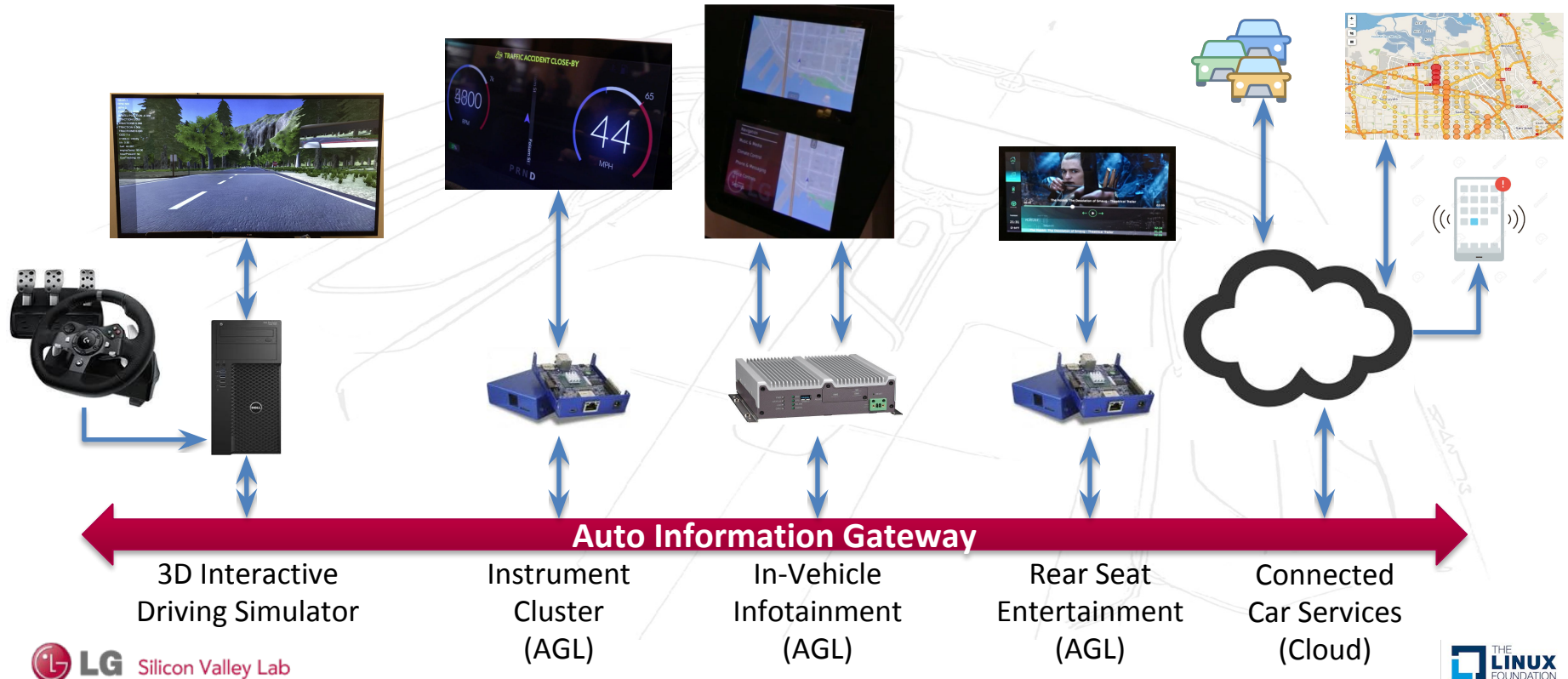
- AGL on IVI, Cluster, and RSE: new apps and compositors
- 3D driving simulator; V2V/V2C connected car cloud services
- Presented multi-system demo at CES

Video:

- SVL: <https://www.youtube.com/watch?v=W36EA0SVEaQ>
- Intel: <https://www.youtube.com/watch?v=NyuTlOlP8YY>

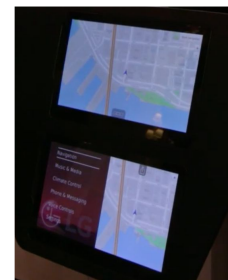
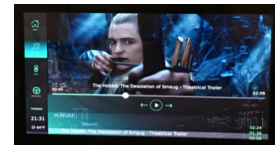


LG SVL AGL Demo at CES 2018



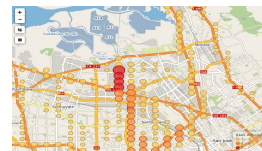
LG SVL AGL Demo at CES 2018

- Rear Seat Entertainment on AGL
 - Premium streaming media support
- Instrument Cluster on AGL
 - Simple compositor and application
 - streams navigation info from IVI
- In-Vehicle Infotainment on AGL
 - New Wayland compositor
 - Custom System UI
 - Multi-screen and multi-surface support



LG SVL AGL Demo at CES 2018

- 3D Driving Simulator
 - Bi-directional video streaming to/from AGL IVI
 - Renders backup camera view for IVI screen
 - Renders IVI screen real-time on dashboard display
 - Hundreds of NonPlayerCars as V2C clients
- Auto Information Gateway (virtual CAN bus)
- Connected Car Cloud Services
 - Real-time reporting of vehicle location and other info
 - Real-time notification of collisions and other events



webOS Open Source Edition

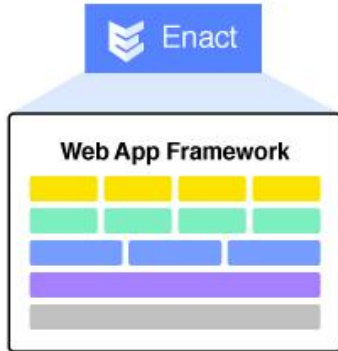
webOS Open Source Edition (OSE)

webOS Open Source Edition(OSE) is now available at <http://webosose.org>

- Powerful and Easy-to-use Open Software Platform
- Highly Optimized Open Source Web App Framework
- Open Platform for All Industries and Participants

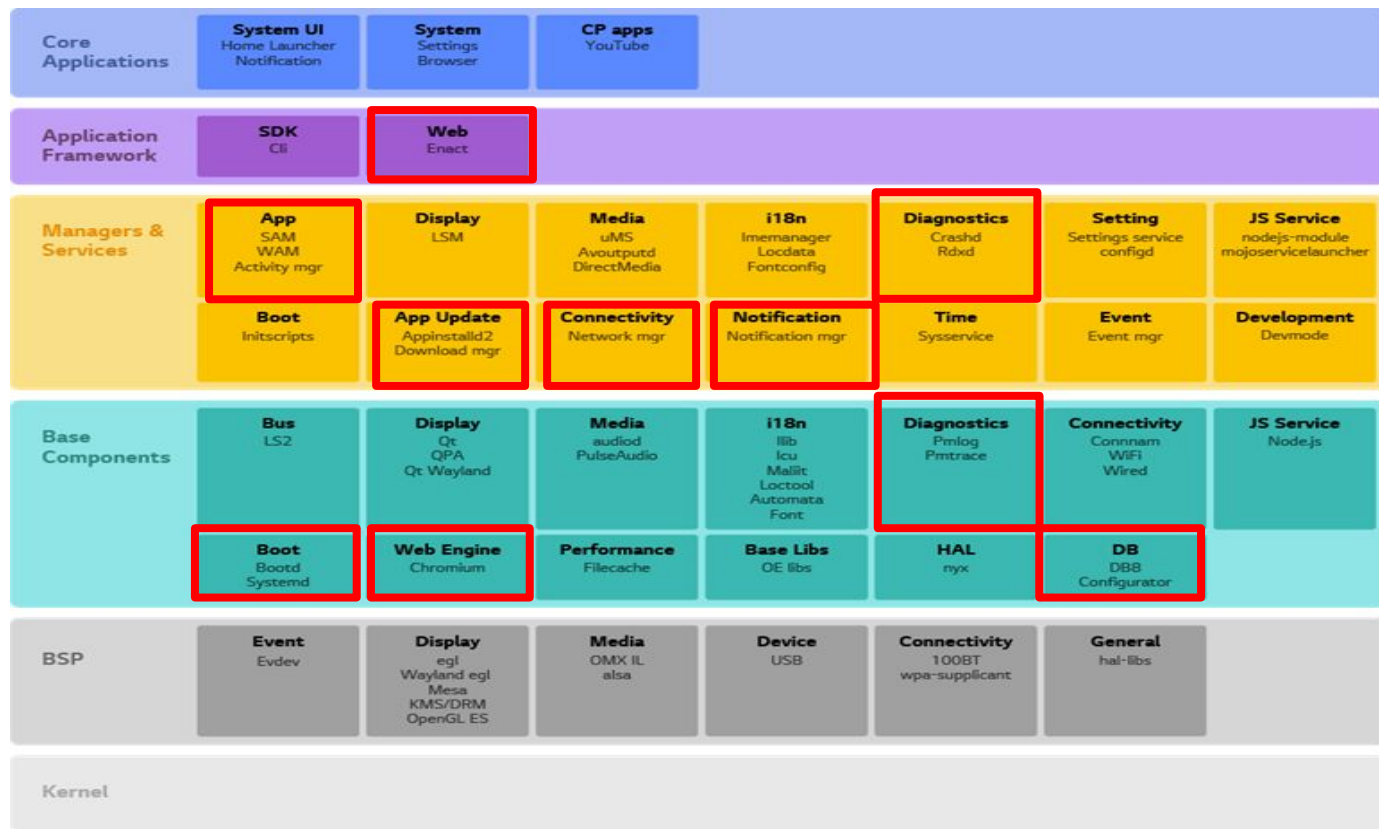


<http://enact.js>



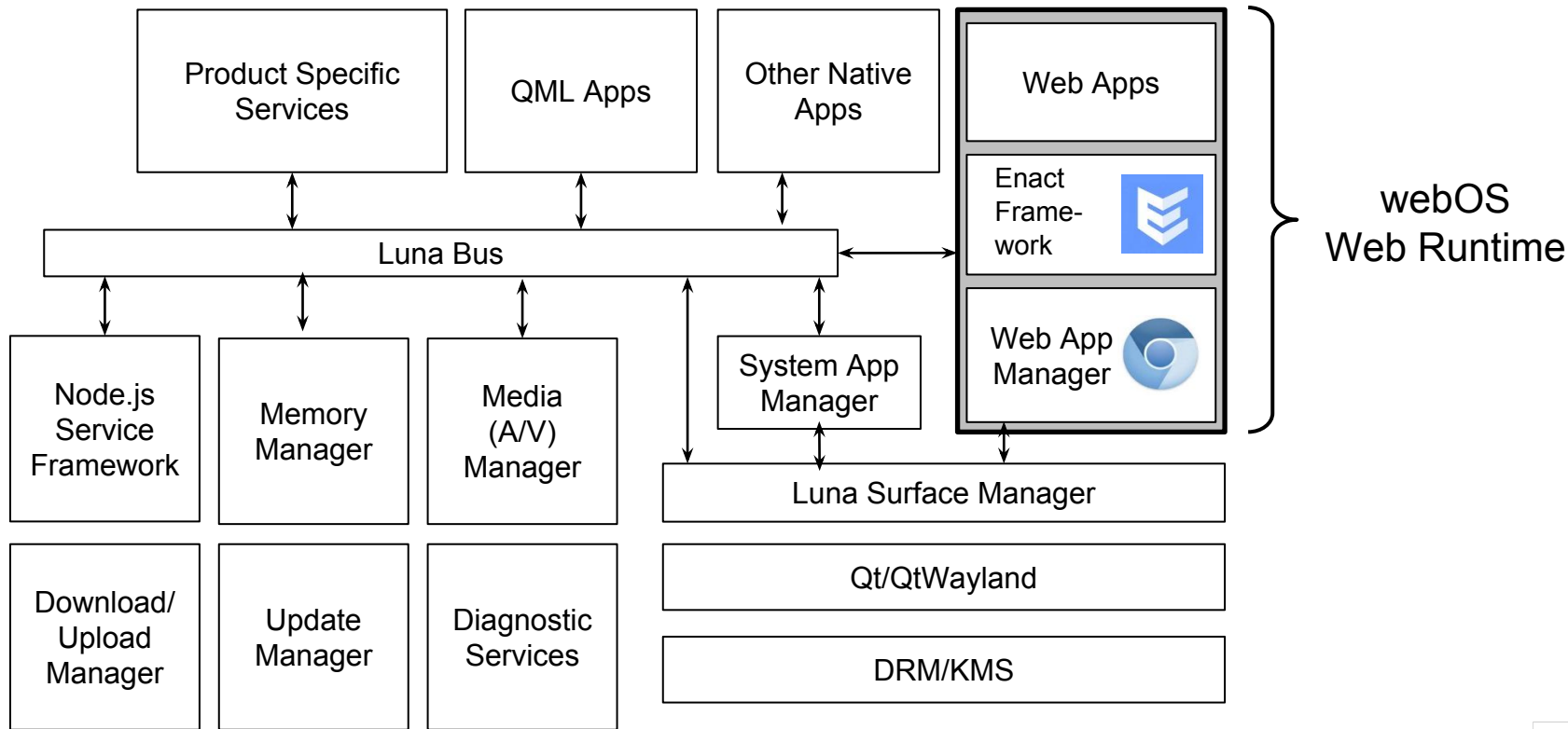
Roadmap: <http://webosose.org/discover/webos-ose-roadmap/>

webOS OSE Components: Layered View



webOS OSE Components: Dynamic View

<https://github.com/webosose>



webOS Web App Runtime

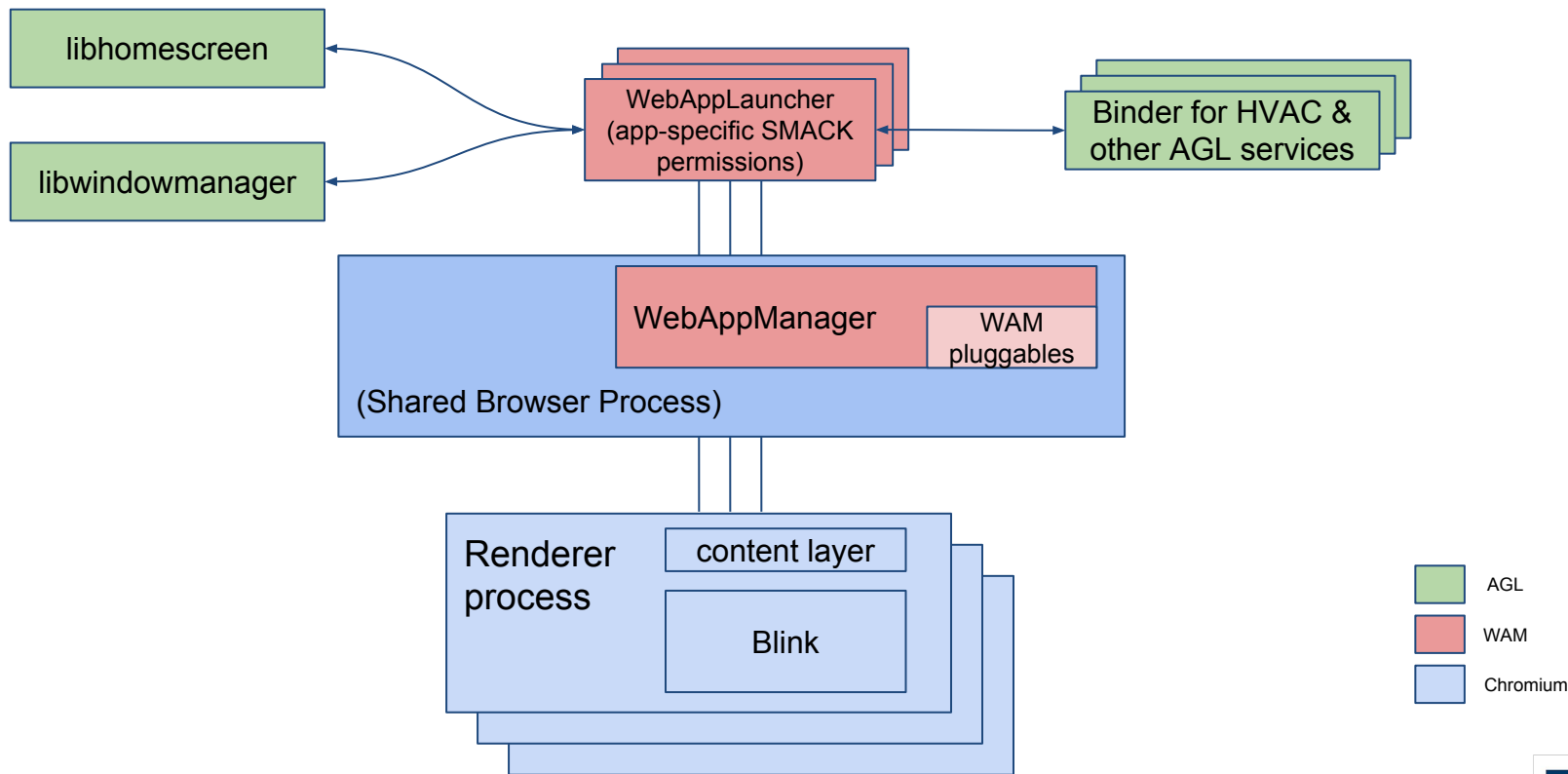
Evolution of Web App Support in webOS



Web App Runtime on AGL: Features

- Web App lifecycle (launch, close, suspend/resume)
- CPU Optimization (launch time, suspend/resume, fast task switching)
- Does not require Qt/QtWebengine
- Recovery Mechanism: recovers 'no response' apps.
- Access Control: controls access to system resources depending on the web app's trust level.
- Responding to Low Memory: provides an interface for low memory responses.

webOS Web App Runtime for AGL: Process Architecture



Web App Runtime on AGL: Source Code

Proof of concept buildable on GitHub:

AGL meta layer at webOS OSE github:

- github.com/webosose/meta-agl-lge

Source code in AGL branches:

- github.com/webosose/chromium53
- github.com/webosose/wam

WAM and Enact Demo on AGL

WebAppManager runs four different html5 apps:

- MemoryMatch
- Annex
- HVAC-Enact (Enact AGL HVAC Proof of Concept)
- YouTube

Video Demo:

- <https://bit.ly/WAMonAGL>

Next Steps: System Integration

- POC code already on GitHub
- Update for Funky Flounder
- Add support for AGL services
- Test on other reference boards
- Add to [html5] profile layer

enact



LG's JavaScript Framework

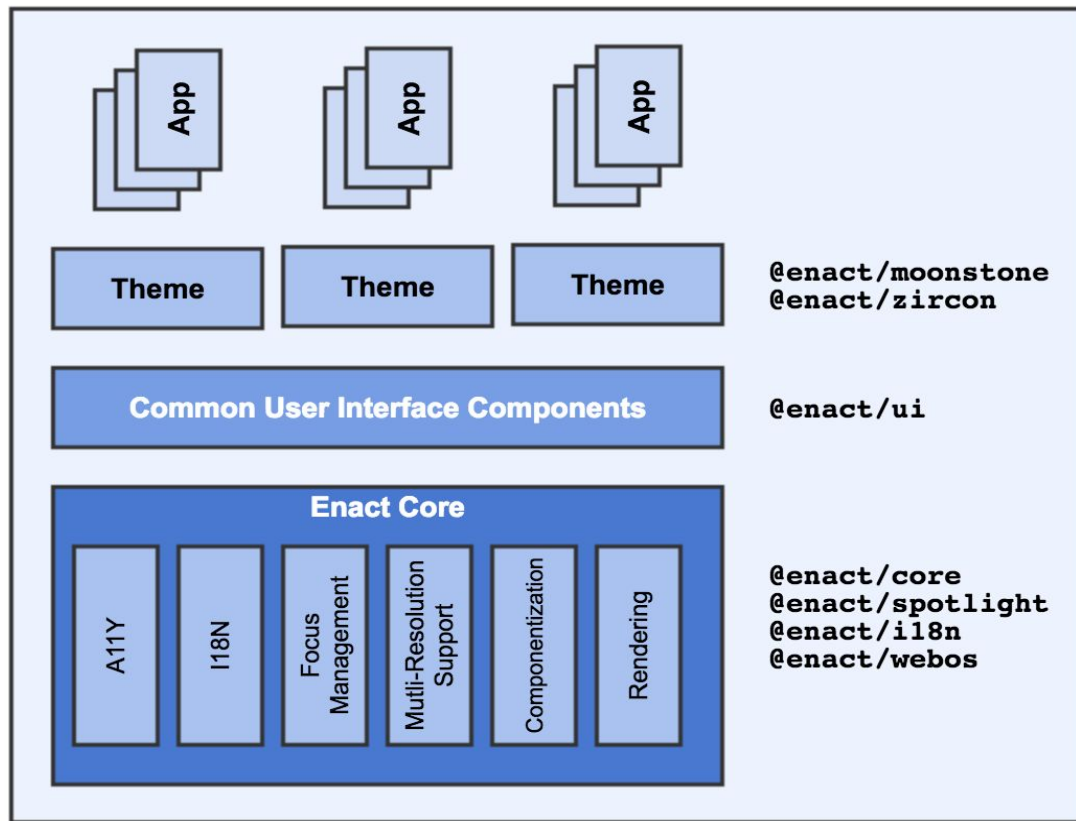
Introduction

LG Silicon Valley Lab
June 2018

Why do you need a framework?

- Consistent look, feel and behavior across apps
- Consistently high performance across apps
- Ability to make improvements across apps, quickly and efficiently
- Avoidance of duplicated effort across app teams
- Consistency of architecture, practices and style across app teams
 - Increased ability to share code between apps
 - Reduced fragmentation

Enact Components



Enact Features

- Composable widget set
- Command-line tool
- Theming and skinning support
- Internationalization & Localization
- Accessibility
- Layout
- Support modules
- Linting and testing architecture
- Documentation and samples
- Optimized scrolling and virtual list support
- V8 snapshot support

Source code on GitHub:
[https://github.com/enactjs](https://github.com/enactjs/enactjs)
(Apache License)

Without Enact



Developing with Enact

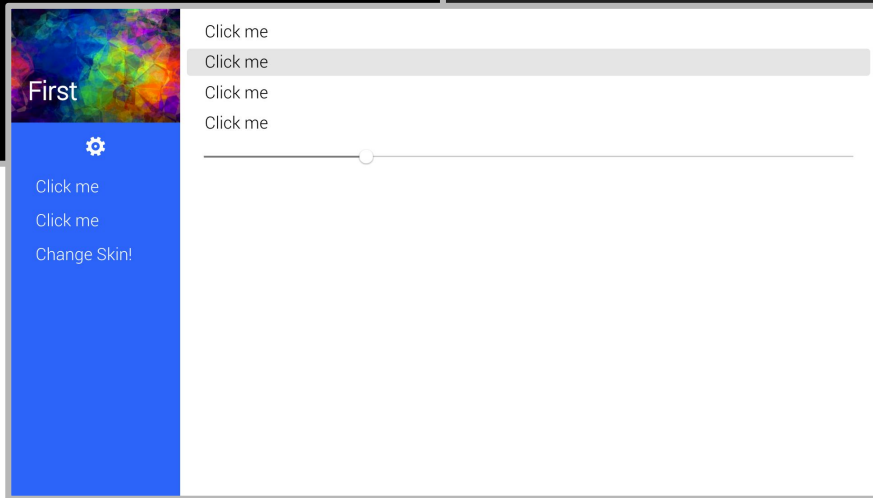
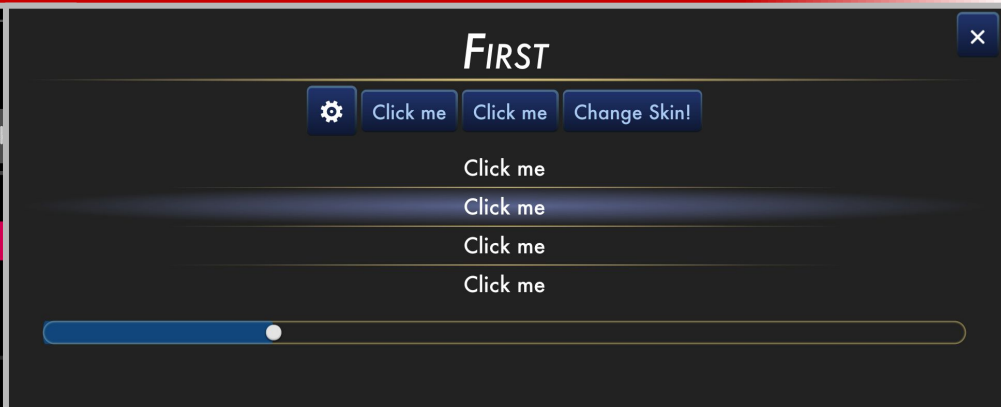
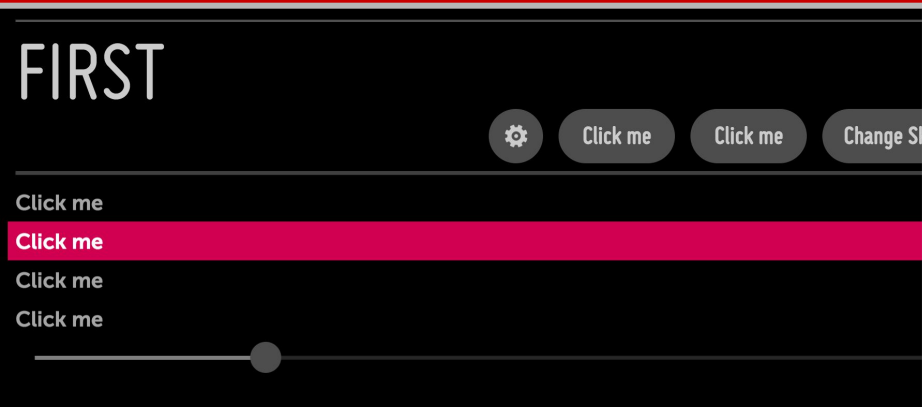
```
> npm install -g @enact/cli
```

```
> enact create myApp
```

```
> cd myApp
```

```
> npm run serve
```

Theming and Skinning



Questions?



THE LINUX FOUNDATION



AUTOMOTIVE
LINUX SUMMIT