



webOS 부문 기술설명회

제 19회 임베디드 SW 경진대회





1

webOS는?

webOS가 무엇인지 알아보자



webOS는?

- 안드로이드, iOS 같은 Linux 기반 운영체제
- 효율적인 버스 시스템 (Luna-Bus)
- 다양한 App Framework 지원
- 시장에서 검증된 플랫폼
 - Palm의 Pre, HP의 Touch Pad
 - LG전자의 TV와 Signage, 스마트워치 등

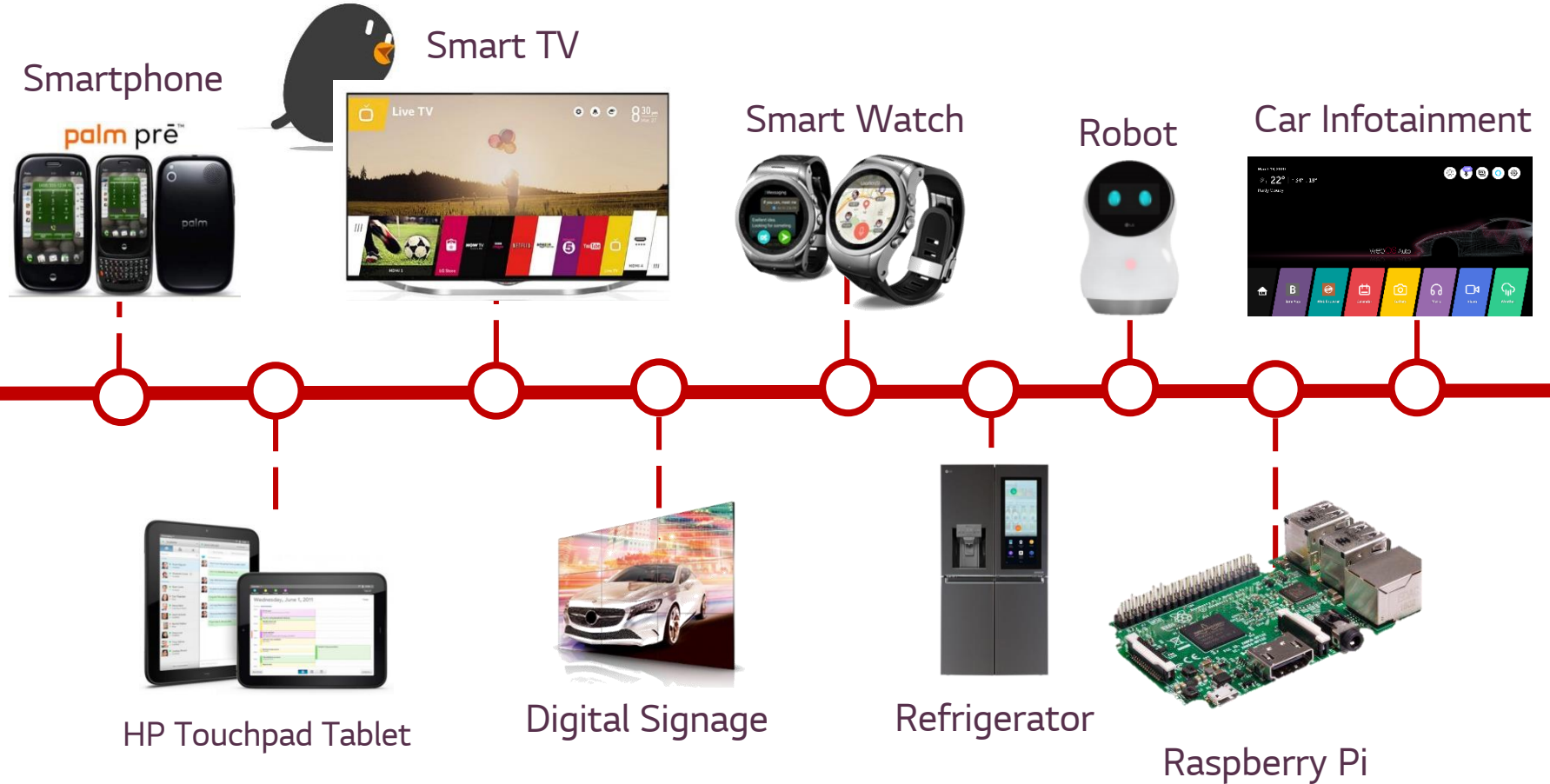
iOS



TIZEN

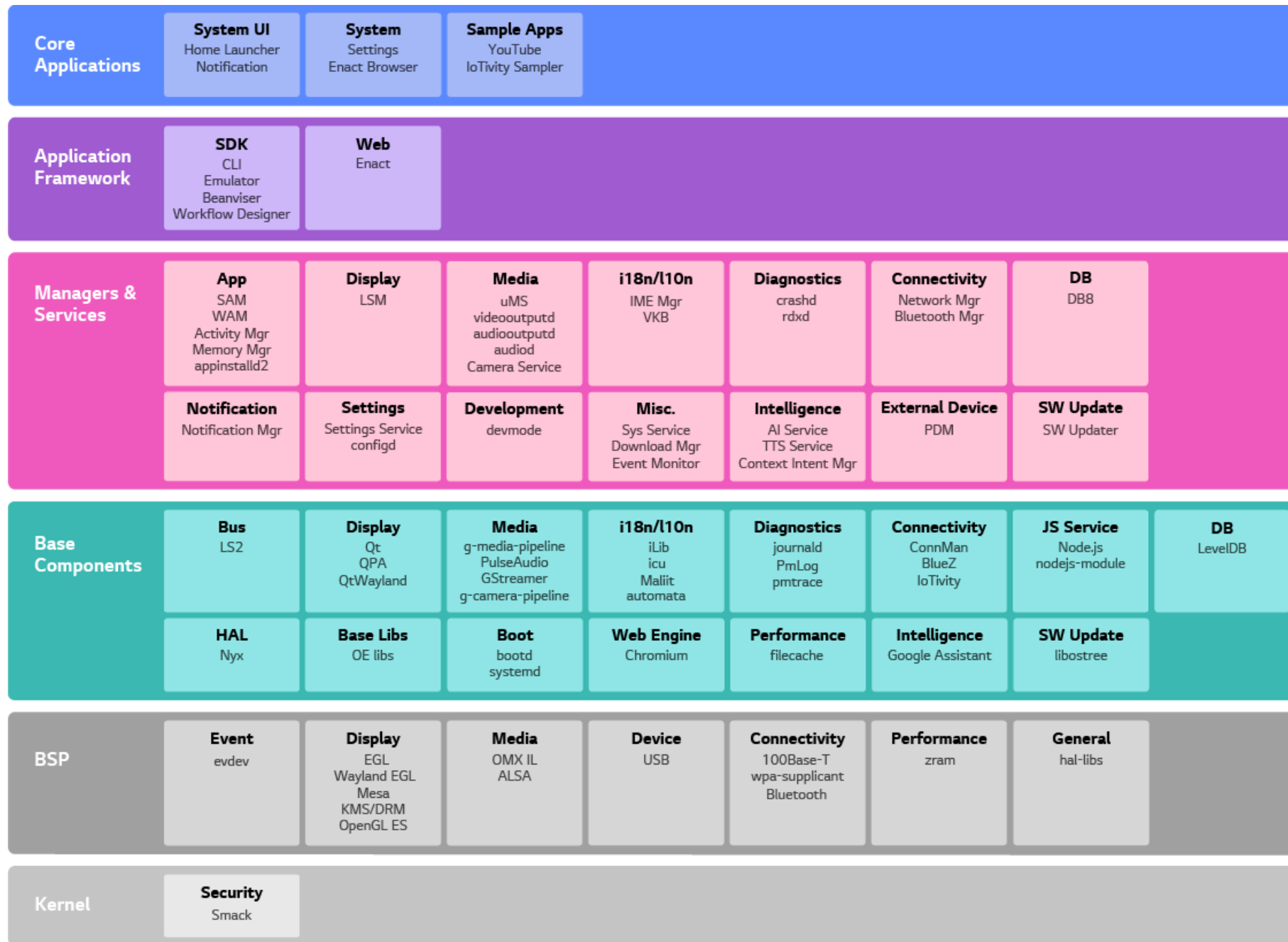
webOS

webOS의 역사





webOS OSE로 살펴보는 구조



<https://www.webosose.org/docs/guides/core-topics/architecture/architecture-overview/>



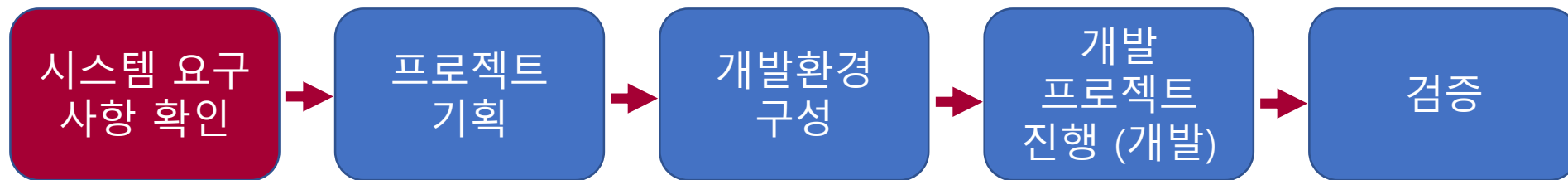
2

어떻게 시작하지?

webOS OSE가 제공하는 개발키트가 무엇이고,
어떻게 활용하는지 보자



개발 프로세스



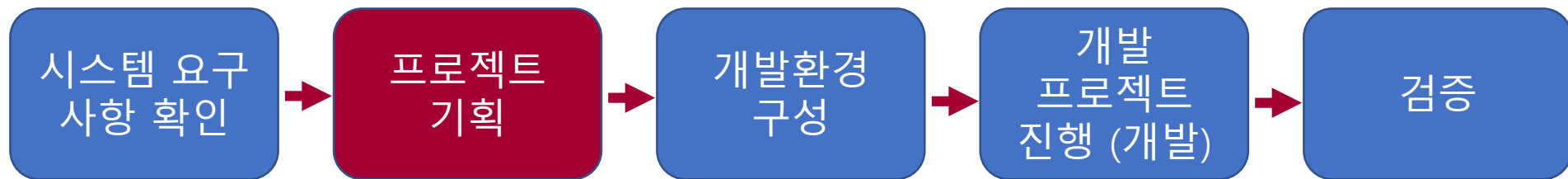
프로젝트에 필요한 HW/SW의 Spec, 빌드 환경등 확인

<https://www.webosose.org/docs/guides/setup/system-requirements/>

- webOS OSE 2.11.0
- RasperryPi 4
- Dual Display 지원
- Touch Input 지원
- Internet 연결 필요
- Linux PC에서만 빌드 지원
- 앱 개발은 Linux / Windows / macOS 지원



개발 프로세스

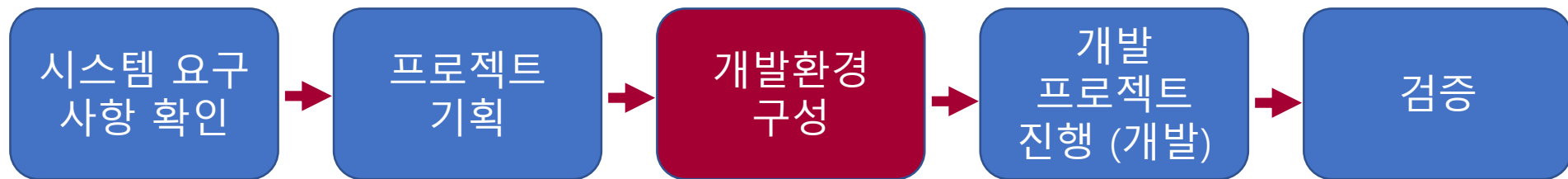


구상하고 있는 프로젝트를 기획

- 지원되는 API 확인
 - Web 표준 API
 - Luna Service API
- 앱 / 서비스 기획
- 구현이 필요한 각각의 기능 정의 및 시스템 구성
- 시스템 전체적인 Interface 구성
- 제약사항 등 확인



개발 프로세스

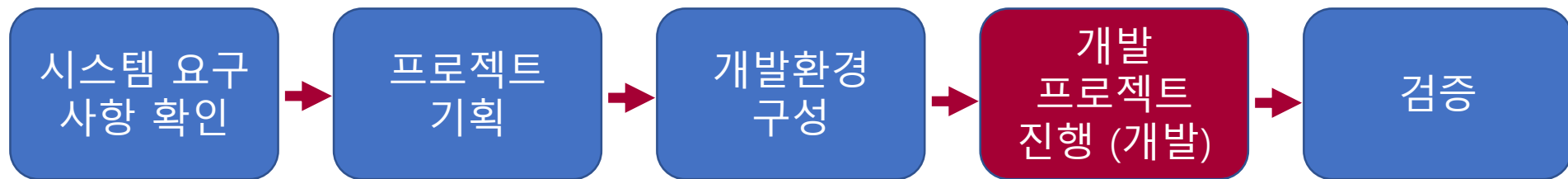


개발을 위한 환경을 구성

- IDE 선택
 - Visual Studio / Atom 등
- CLI 설치
- 플랫폼 이미지 준비
 - 플랫폼 직접 빌드 또는 빌드된 이미지
- 플랫폼 설정
 - 네트워크 / 디스플레이 등



개발 프로세스

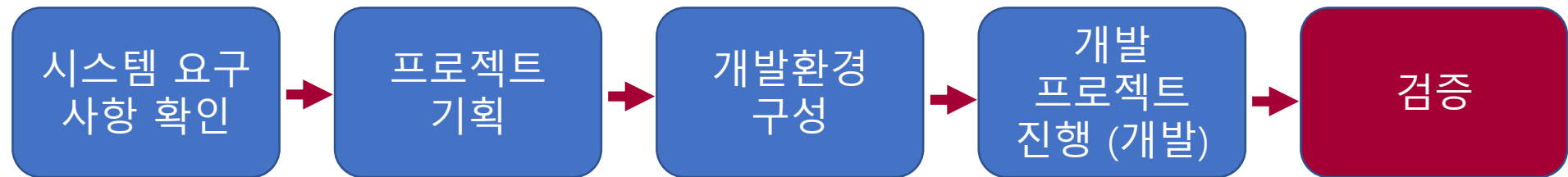


계획한 대로 개발을 진행

- 시스템 구성 (HW)
- 개발자 사이트의 문서를 참고하여 소프트웨어 개발
 - Development Guide
 - Tutorial Guide
- 동작 가능한 수준으로 개발
- 검증 단계에서 발견된 버그 수정



개발 프로세스




디바이스에 설치하여 검증

- 앱 패키징
- 앱 설치
- 앱 실행
- 디버깅 도구를 활용하여 디버깅



개발 환경 구성하기 (Device)

- 시스템 요구사항
 - <https://www.webosose.org/docs/guides/setup/system-requirements/>
- webOS OSE 빌드하기
 - <https://www.webosose.org/docs/guides/setup/building-webos-ose/>
- 빌드된 이미지 플래싱하기
 - <https://www.webosose.org/docs/guides/setup/flashing-webos-ose/>
 - Prebuilt Image : <http://build.webos-ports.org/webosose/>
- 네트워크 설정
 - <https://www.webosose.org/docs/guides/setup/setting-up-networking/>
- 듀얼 디스플레이 설정
 - <https://www.webosose.org/docs/guides/setup/setting-up-dual-displays/>



개발 환경 구성하기 (SDK)

- 다운로드 SDK
 - <https://www.webosose.org/docs/tools/sdk/sdk-download/>
- CLI (Command Line Interface)
 - <https://www.webosose.org/docs/tools/sdk/cli/cli-user-guide/>
- Emulator
 - <https://www.webosose.org/docs/tools/sdk/emulator/virtualbox-emulator/emulator-user-guide/>
- Beanviser
 - <https://www.webosose.org/docs/guides/setup/setting-up-networking/>
- Workflow Designer
 - <https://www.webosose.org/docs/tools/sdk/workflow-designer/workflow-designer-user-guide/>




개발 환경 구성하기 (IDE)



Visual Studio Code | Atom | Sublime Text

- 14 Best IDEs(Integrated Development Environments)
 - <https://tms-outsource.com/blog/posts/web-development-ide/>

#이미지출처: 위 링크와 동일함



개발 환경 구성하기 (CLI)

- 요구사항

- Node.js - <https://nodejs.org/en/download/>
- npm

- CLI 설치

```
C:\Users\admin>node -v
v14.17.3

C:\Users\admin>npm -v
6.14.13

C:\Users\admin>npm install -g @webosose/ares-cli
C:\Users\admin\AppData\Roaming\npm\ares-device -> C:\Us
\ares-device.js
C:\Users\admin\AppData\Roaming\npm\ares -> C:\Users\admin
s
C:\Users\admin\AppData\Roaming\npm\ares-install -> C:\U
n\ares-install.js
```

- CLI 설치 확인

```
C:\>ares --version
Version: 2.2.0
```



개발 환경 구성하기 (Emulator)

- 요구사항

- Virtual Box (6.0 이상) - <https://www.virtualbox.org/wiki/Downloads>

- 직접 빌드

- <https://www.webosose.org/docs/tools/sdk/emulator/virtualbox-emulator/emulator-user-guide/>

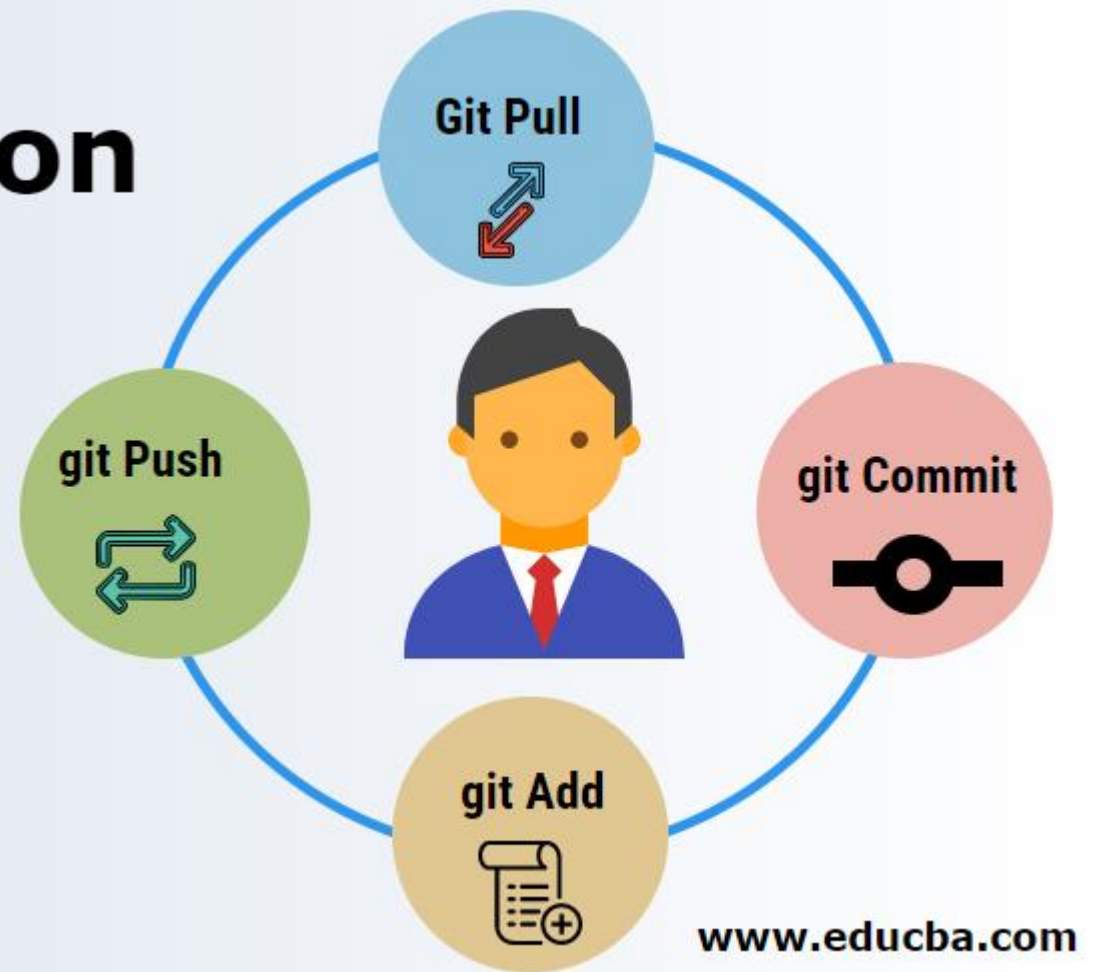
- Pre-built Image

Index of /webosose/qemux86/

../		
build-174-v2.4.0/	02-Apr-2021 10:30	-
build-197-v2.5.0/	02-Apr-2021 10:30	-
build-218-v2.6.0/	02-Apr-2021 10:29	-
build-250-v2.7.0/	02-Apr-2021 10:29	-
build-263-v2.8.0/	02-Apr-2021 10:28	-
build-295-v2.9.0/	02-Apr-2021 15:40	-
build-325-v2.10.0/	02-Apr-2021 10:28	-
build-357-v2.11.0/	25-Jun-2021 09:22	-
build-361-v2.11.0/	25-Jun-2021 09:23	-

개발 환경 구성하기 (Git)

Introduction To GIT



- 생활코딩

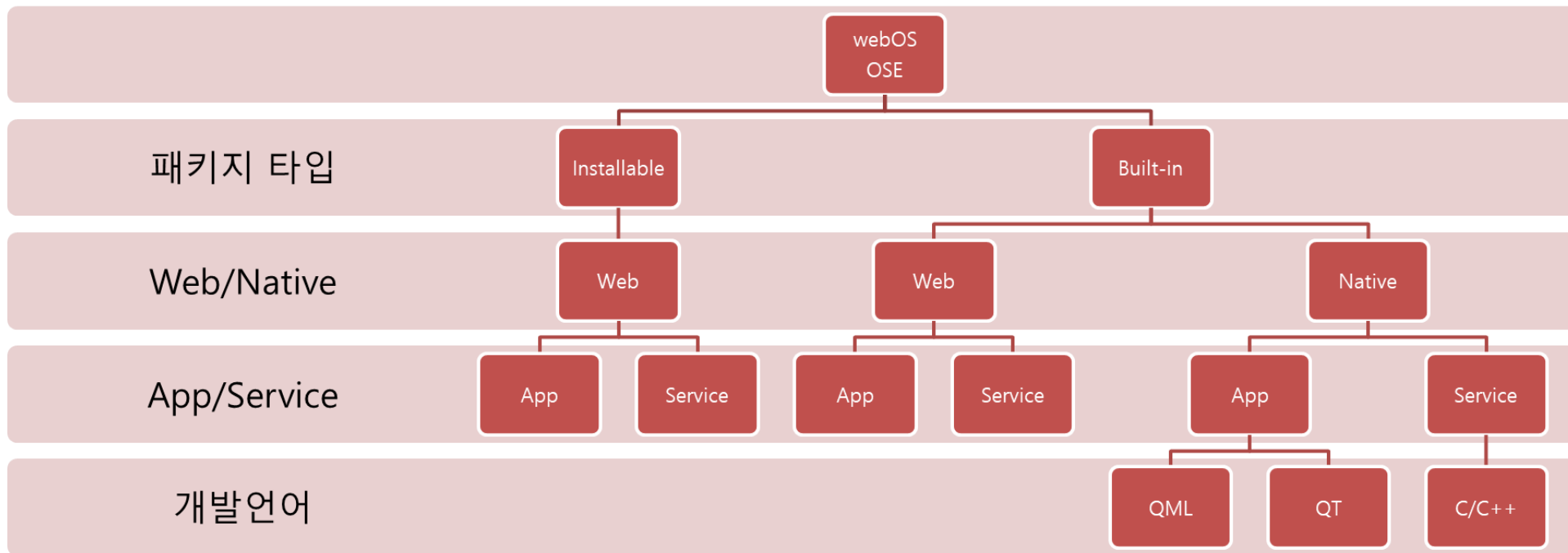
- <https://opentutorials.org/course/2708>



#이미지출처: <https://www.educba.com/introduction-to-git/>

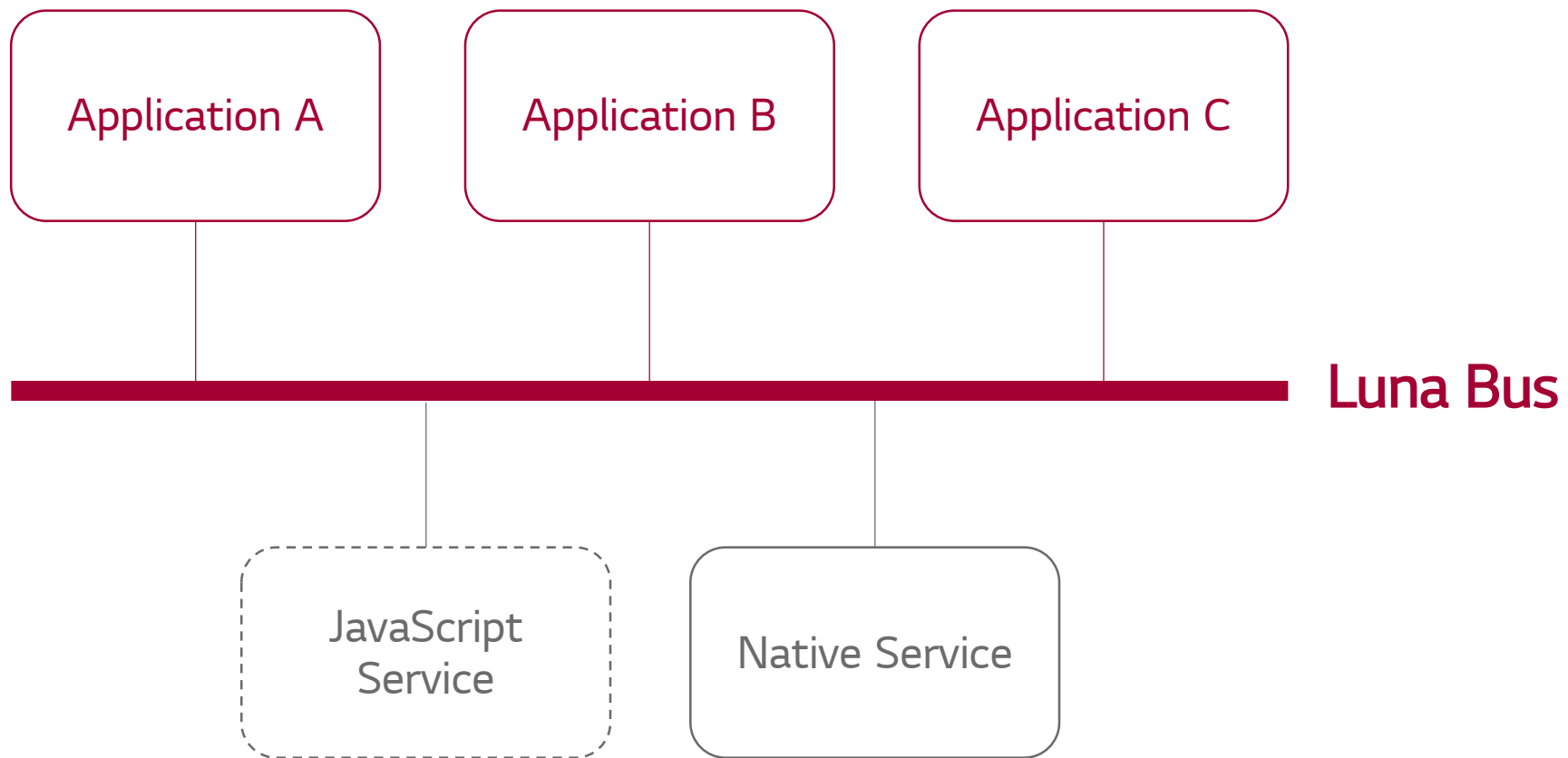


개발 시작하기 - 무엇을 개발할까





개발 시작하기 - 앱 vs 서비스





개발 환경 구성 및 앱 개발

데모

- ✓ ares 명령어 목록보기
- ✓ 각 명령어 살펴보기
 - ✓ First_App 생성
- ✓ First_Service 생성
 - ✓ Packaging
 - ✓ Installing
 - ✓ Launching
 - ✓ Inspecting



3

무엇을 참고할까?

webOS OSE가 제공하는 API들은 무엇이고
어떻게 쓰는지 보자



Web 앱 & 서비스 구성

- 웹 애플리케이션 소스, icon 파일
- 앱 메타파일
 - appinfo.json
 - <https://www.webosose.org/docs/guides/development/configuration-files/appinfo-json/>
- 서비스 메타파일
 - packageinfo.json
 - <https://www.webosose.org/docs/guides/development/configuration-files/packageinfo-json/>
 - services.json
 - <https://www.webosose.org/docs/guides/development/configuration-files/services-json/>



개발 가이드

Development Guides

Learn how to develop apps and services for webOS OSE.

Web Apps
JS Services
QML Apps
Native Apps
Native Services

Common Guides

See the common subjects for development.

Configuration Files
Localization Guide
Logging Guide

- 웹 앱 개발 가이드

- <https://www.webosose.org/docs/guides/development/web-apps/web-app-overview/>

- 서비스 개발 가이드

- <https://www.webosose.org/docs/guides/development/js-services/js-service-overview/>



개발 튜토리얼

Tutorials for 3rd Party Developers

Check out tutorials for developing external apps and services.

External Web Apps
External JS Services
External QML Apps
External Native Apps
External Native Services

Tutorials for Contributors

Learn how to develop apps and services for contribution.

Built-in Web Apps
Built-in JS Services
Built-in QML Apps
Built-in Native Apps
Built-in Native Services

- 튜토리얼 (샘플 앱)
 - <https://github.com/webosose/samples>
- 앱 & 서비스 개발 튜토리얼
 - <https://www.webosose.org/docs/tutorials/>



API 레퍼런스

- Web Standard API
 - <https://developer.mozilla.org/ko/docs/Web/API>
- LS2 API (For web app and service)
 - <https://www.webosose.org/docs/reference/ls2-api/ls2-api-index/>
- WebOSServiceBridge API (For web app)
 - <https://www.webosose.org/docs/reference/weboservicebridge-api/weboservicebridge-api-reference/>
- webos-service library API (For web service)
 - <https://www.webosose.org/docs/reference/webos-service-library/webos-service-library-api-reference/>
- luna-service2 library API (For Native)
 - <https://www.webosose.org/docs/reference/luna-service2-library/luna-service2-library-api-reference/>



Debug 메시지

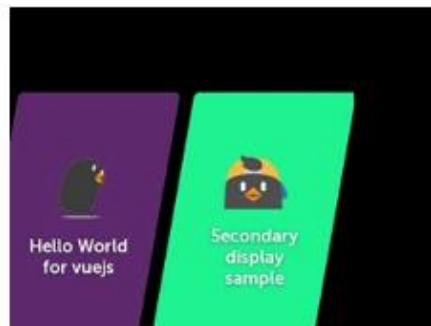
Target	journald	pmlogd
Log files	/run/log/journal/*/*	/var/log/messages*
How to follow logs	journalctl -f	tail -f /var/log/messages
Previous log files	/var/spool/rxdx/previous_ boot_logs.tar.gz	/var/spool/rxdx/previous_ boot_logs.tar.gz
How to restore previous logs	unzip /var/spool/rxdx/pre vious_boot_logs.tar.gz file > journald-* file	unzip /var/spool/rxdx/pre vious_boot_logs.tar.gz file > messages file

- PmlogLib

- Supports C/C++, JavaScript, Node.js, and QML
- <https://www.webosose.org/docs/reference/pmloglib-library/pmloglib-library-api-reference/>



블로그 아티클



Article

**Launching web app on
secondary displays**

June 11, 2020



Article

Using OpenCV in webOS OSE

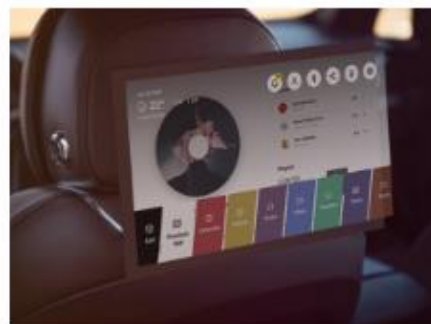
May 19, 2020



Article

FOTA in webOS OSE

December 12, 2019



Article

**LG Software Solutions Offers
webOS Auto as an Ideal
Infotainment Platform**

November 26, 2019



Article

**Web Applications and Web
App Manager in webOS OSE**

November 22, 2019



Article

**How Chromium Helps webOS
OSE Build an Ecosystem**

November 15, 2019

<https://www.webosose.org/blog/>



4

ThinQ AI – Cloud API

음성인식에 Cloud API 활용



5

프로젝트 TIP

프로젝트 개발 팁, 무엇을 주의해야 하는지 알아보자



프로젝트를 잘하기 위한 방법

• 팀원들과의 협업 하라

- 팀으로 하는 개발은 나 혼자만 잘해서 되는 것은 아니에요.
- 작업하는 내용은 항상 공유하고 서로가 알 수 있도록 해야 해요.

• 사소한 것도 기록을 남기자

- 코드에는 주석을 다는 습관을 가져야 해요. 나만 보는 코드가 아니죠.
- 틈틈이 문서를 남겨요. 블로그를 쓰면 더 좋겠죠.

• 다른 사람이 이해할 수 있게 하자

- 서비스를 구현하고, 새로운 알고리즘을 설계/구현 했지만, 심사위원을 이해시키지 못 하면 의미가 없어요.

• 보기 좋은 음식이, 먹기도 좋다

- 디테일 한 부분을 신경 써야 해요. 특히, 심사위원의 눈에 보여진다면요.



프로젝트를 잘하기 위한 방법

• 내가 만들고 있는 제품이 무엇을 위한 것인지 생각하자

- 프로젝트를 시작할 때 정의했던 목적을 생각하면 좋아요.

• 동작하는 프로토타입을 개발하라

- 모든 기능이 구현이 안되었다고 기능구현에만 집중하는 경우가 많아요.
- 일단 작은 기능으로 동작하는 프로토타입을 만들고, 확장해 가세요.

• 때로는 돌아서 가라

- webOS OSE로 구현이 충분하지 않을 때, 환경적인 제약이 있을 때 좌절하지 말고 다른 방법이 있는지 찾아보자. 그것이 시간을 줄이는 방법이에요
- 필요하다면 멘토에게 물어보자

• webOS는 제대로 표기하라

- Web OS, WebOS, webos 와 같이 다양하게 사용하는데, 잘못된 것이에요
- “webOS”와 같이 소문자 w로 시작하고 붙여서 써야하며, OS는 대문자예요.



6

질문있어요

이럴땐 어떻게 하지?



궁금해요

• 대회 진행하는 동안 궁금한 부분들은 어떻게 해결하나요?

- 개발자 사이트 Forum : <https://forum.webosose.org/>
- Slack Channel : <http://ebsw2021.slack.com/>

• webOS 플랫폼에서 지원하지 않는 기능은 어떻게 하죠?

- 기능 지원을 위한 새로운 보드를 추가하여 해당 기능을 처리하고 결과를 가져와서 활용하시면 됩니다.

• 무엇을 중점적으로 보나요?

- 우선은 기획한대로 정상적으로 동작하는 것이 중요하겠습니다.
- 디스플레이 되는 영역의 완성도도 중요합니다. 그리고, 팀원 모두가 작품에 대한 이해를 동일하게 가지고 있고, 설명할 수 있다면 더 좋겠습니다.

• 센서 등을 추가하는데 제약은 없나요?

- 라즈베리파이4의 webOS OSE 플랫폼을 활용하여 다른 센서 등을 추가 연동하는 것에는 제약이 없습니다.



궁금해요

• 데이터를 저장하고 싶어요

- webOS OSE는 기본적으로 DB8이라는 Indexed DB와 유사한 Key-Value 형태로 데이터를 저장할 수 있는 DB를 제공합니다.
- 외부 DB를 사용하고 싶으시다면, 클라우드 서버나 로컬 PC 등을 활용하시면 됩니다.
- HTML5의 로컬스토리지나 세션스토리지를 활용할 수 도 있습니다.
- USB를 사용하고 싶으시다면 webOS OSE의 PDM 서비스를 살펴보세요.

• 외부 센서나 서비스와 통신하려면 어떻게 하나요?

- 다양한 방법들이 있지만, 웹앱 또는 서비스를 기준으로 보자면 Web socket을 이용하는 방법도 있습니다.

• GPIO를 사용할 수 있나요?

- Peripheral Manager를 사용하여 라즈베리파이4의 GPIO를 컨트롤 할 수 있습니다.
- <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-peripheralmanager/>



궁금해요

• 반드시 ThinQ AI 플랫폼을 사용해야 하나요?

- 프로젝트 기획 시점에서는 ThinQ AI 플랫폼의 음성인식 기능을 반드시 사용해야 하는 것으로 했었지만, ThinQ AI 혹은 webOS OSE에서 제공하는 TTS, Voice Assistant (Google) 기능 중 선택해서 사용하시면 됩니다.

• Face Recognition은 직접 구현해야 하나요?

- 네 직접 구현해야 합니다.
- webOS가 아닌 별도의 얼굴인식을 위한 시스템을 구축하면 좋습니다. webOS OSE 상에서 Python 및 YOLO를 활용하기 어렵기 때문에, 다른 플랫폼을 활용하는 것이 좋습니다.

• 웹 애플리케이션과 웹 서비스는 필수인가요?

- 네, webOS 에서 동작하는 웹 애플리케이션과 웹 서비스는 필수 입니다.



“THANK YOU FOR WATCHING”



webOS
Open Source Edition