Write Once and Run Everywhere with **TOAST** for Smart TV

삼성전자 김효진

Write Once and Run Everywhere with **TOAST** for Smart TV

삼성전자 김효진

Agenda

- Fragmentation for Smart TV platform
  - The challenge for developers
- One source for multiple Smart TV platforms
  - Simplification with TOAST
- Technical Specifications
  - Nuts and bolts of the TOAST plugin
- TOAST Configuration
  - Introducing the TOAST plugin
- Using TOAST for Smart TV
  - Getting started

Fragmentation for Smart TV platform

The challenge for developers

OS fragmentation

Developers faced a **new challenge** with various platforms for Smart TV.

**Legacy OS
2012~2014**

**Tizen
2015 & 2016**

For deploying a new service to many users, developers should make **two apps.**

Maintenance

Developers need to **deliver the same app experience** in barely different platforms using slightly different tools.

Common ground

Smart TV platform is **based on web technologies for both *Legacy OS* and *Tizen OS***, relying on TV platform specific APIs.
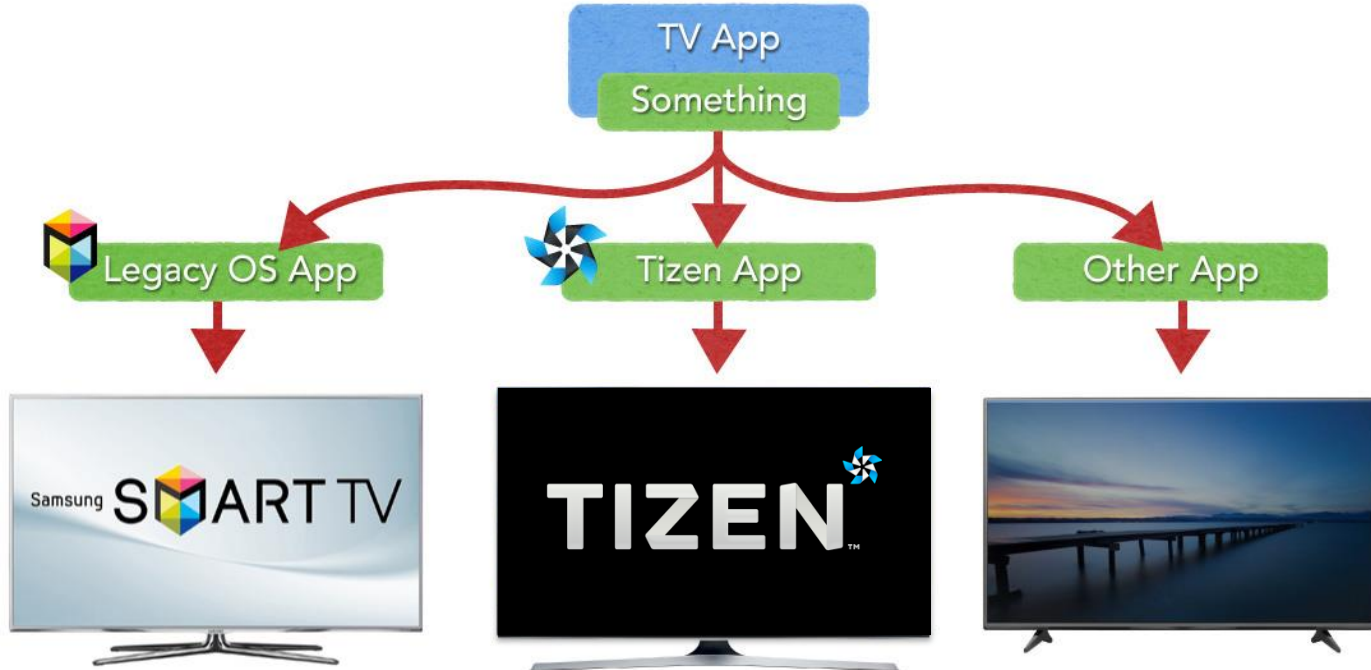
One source for
multiple Smart TV platforms

Simplification with TOAST

One-source, multi-use

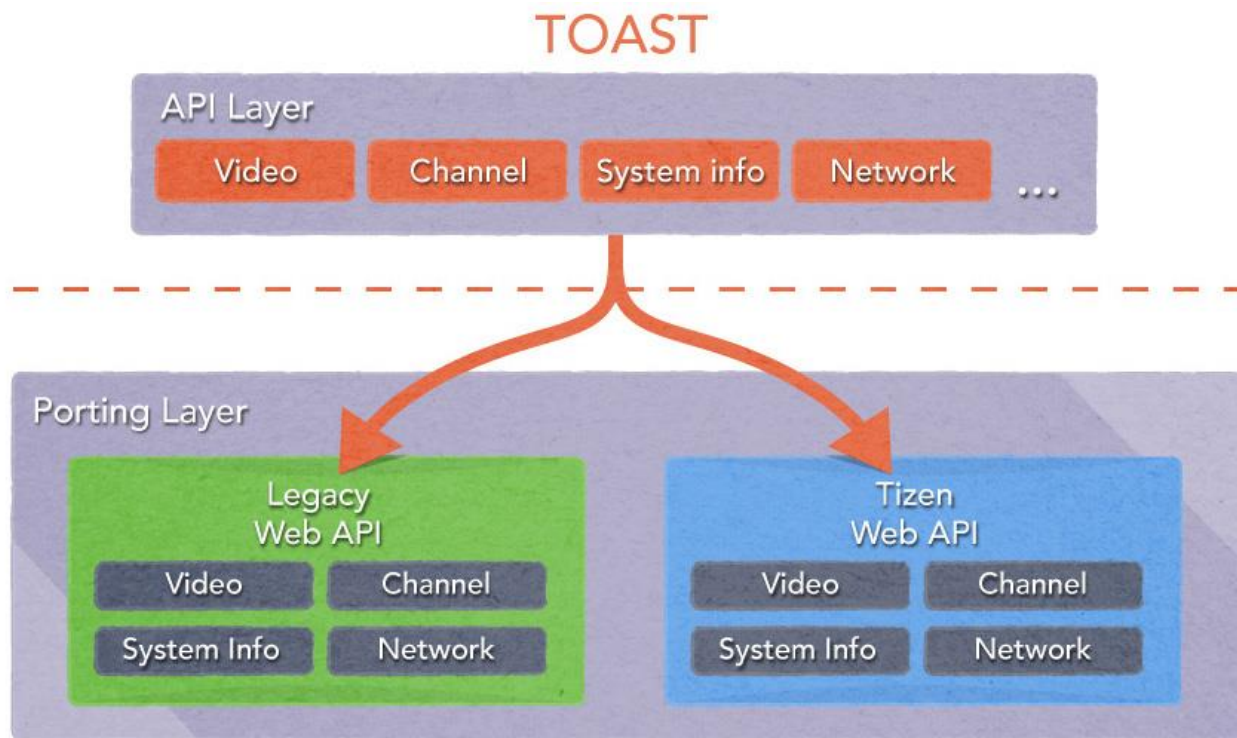As in mobile development, we have a common experience that needs to be delivered to multiple devices.

Introducing TOAST!

TV **O**riented **A**PI **S**implification for Mul-**T**-platform
The Anti-API fragmentation for Smart TV .

TOAST Framework

Technical Concept

The TOAST approach

Deliver the **same experience** in Smart TV with TOAST using **fewer lines**.

Before

Legacy OS	Tizen	webOS
``var sef = document.getElementById('sef');
sef.Open('Player', '1.0', 'Player');
sef.Execute('InitPlayer', url);
sef.Execute('StartPlayback', 0);```	```webapis.avplay.open('url');
webapis.avplay.prepare();
webapis.avplay.play();```	```var videoEl = document.createElement('video');
videoEl.src(url);
videoEl.play();```

After

TOAST
``var media = toast.Media.getInstance();
media.open('url');
media.play();```

Same result, one code base

Deliver the **same experience** in Smart TV with TOAST using **fewer lines**.

Code base

TOAST

```
var media = toast.Media.getInstance();
media.open('url');
media.play();
```

Result

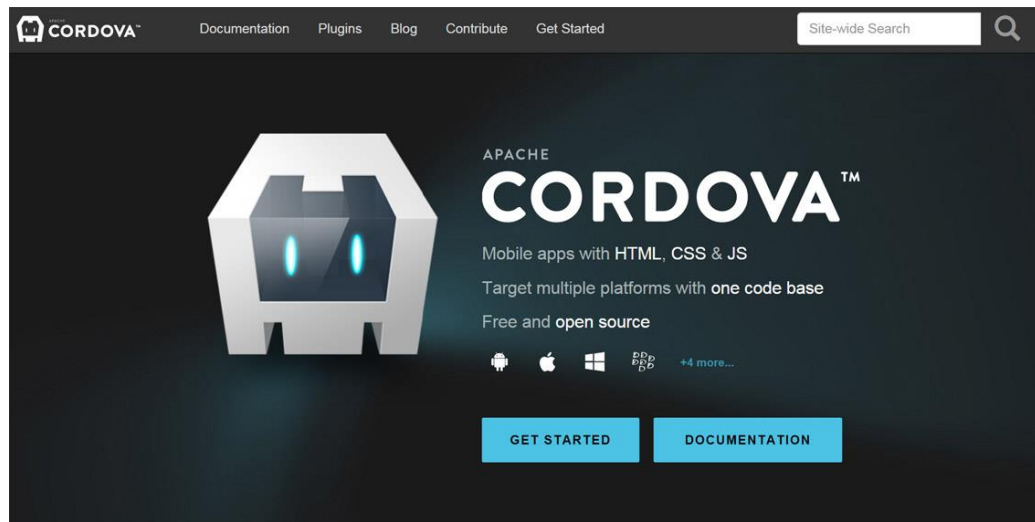| Legacy OS	Tizen	webOS

`var sef = document.getElementById('sef');`
`sef.Open('Player', '1.0', 'Player');`
`sef.Execute('InitPlayer', url);`
`sef.Execute('StartPlayback', 0);`
	`webapis.avplay.open('url');`
	`webapis.avplay.prepare();`
	`webapis.avplay.play();`
		`var videoEl = document.createElement('video');`
		`videoEl.src(url);`
		`videoEl.play();`

Technical Specifications

Nuts and bolts of the TOAST plugin
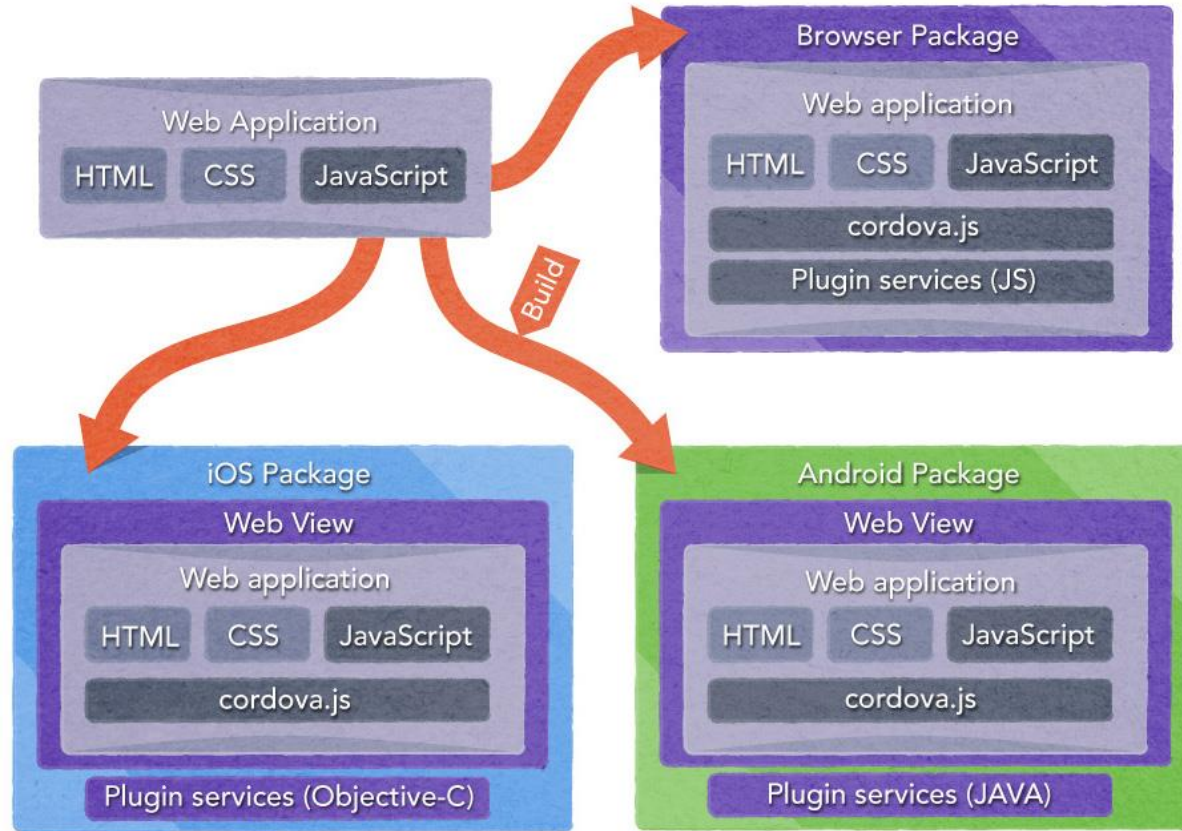
Open source technologies

TOAST based on **Cordova**.

- An open source project of the Apache Software Foundation.
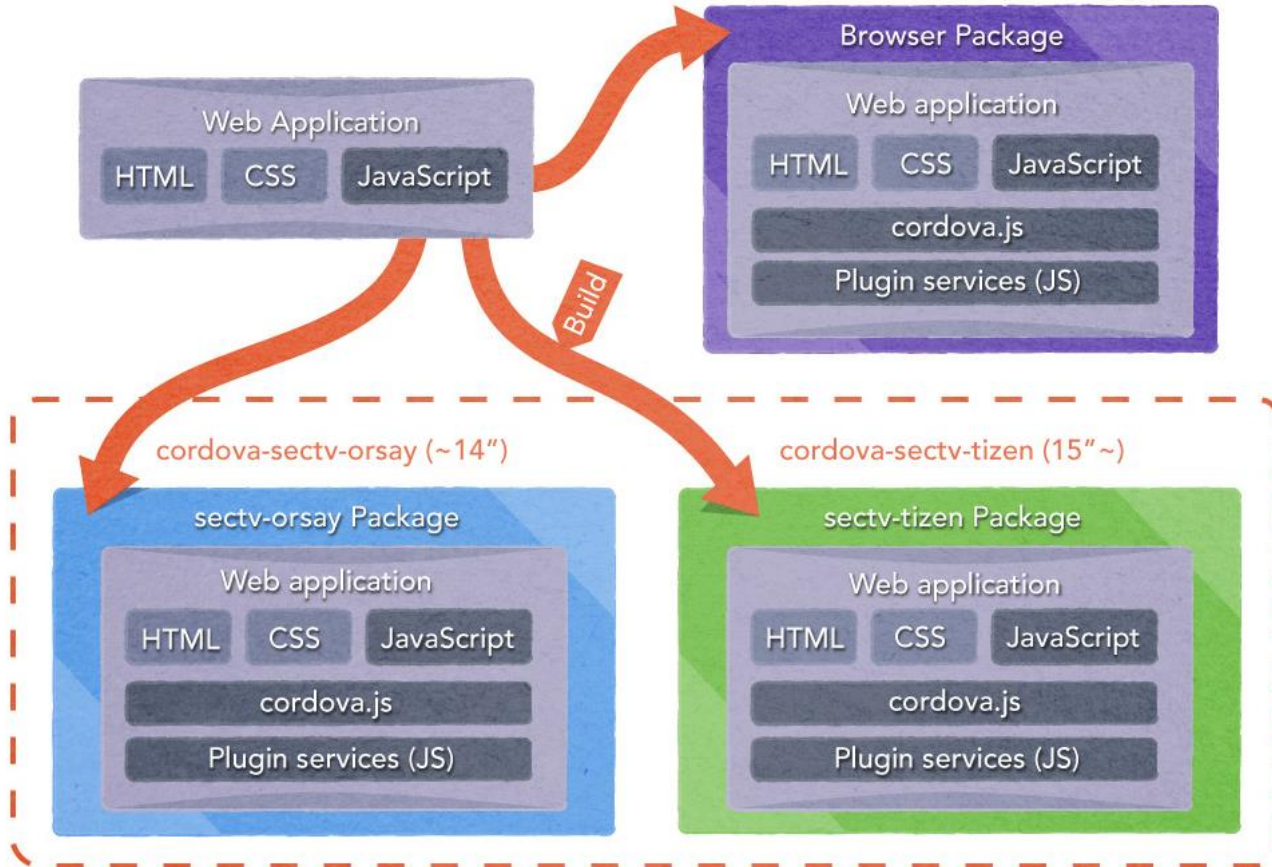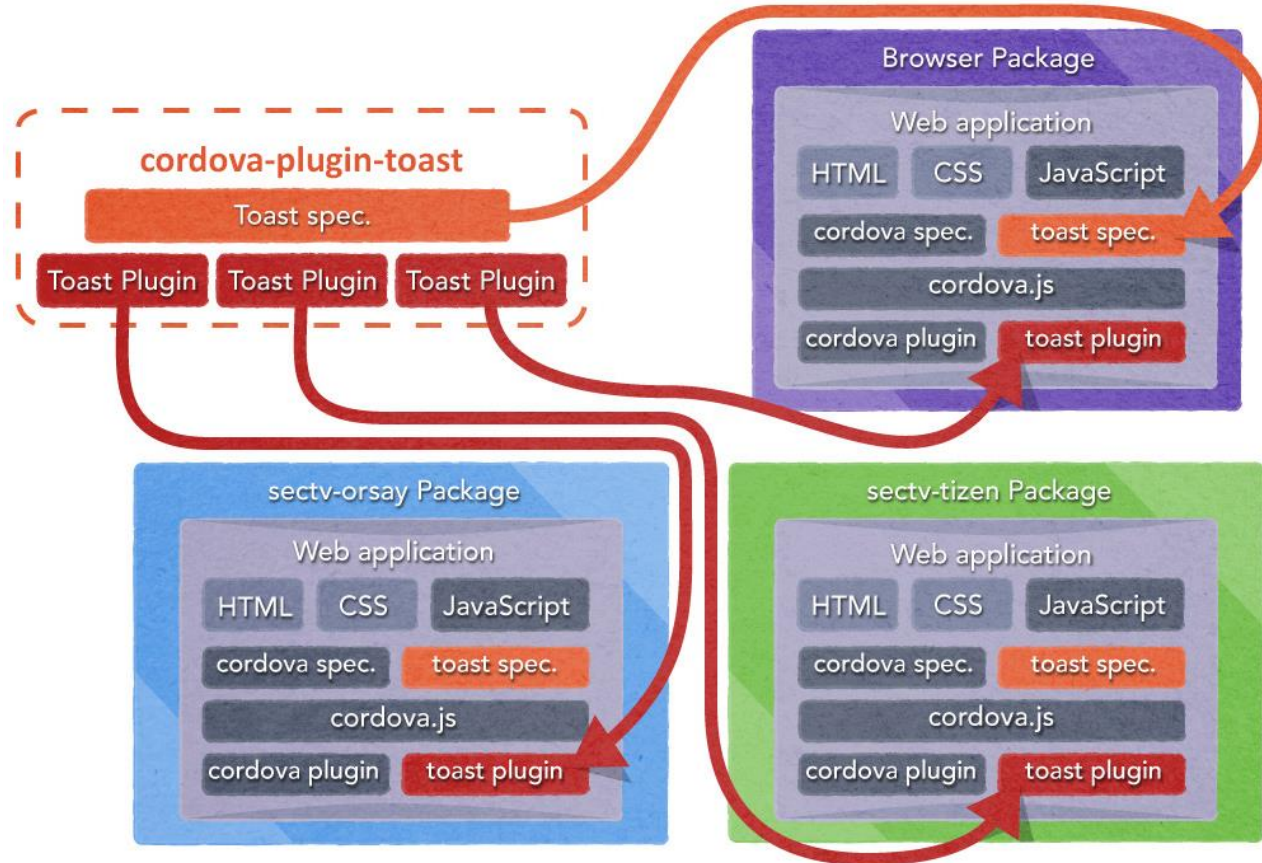- Support One-source Multi-use on a variety of Mobile Platform.

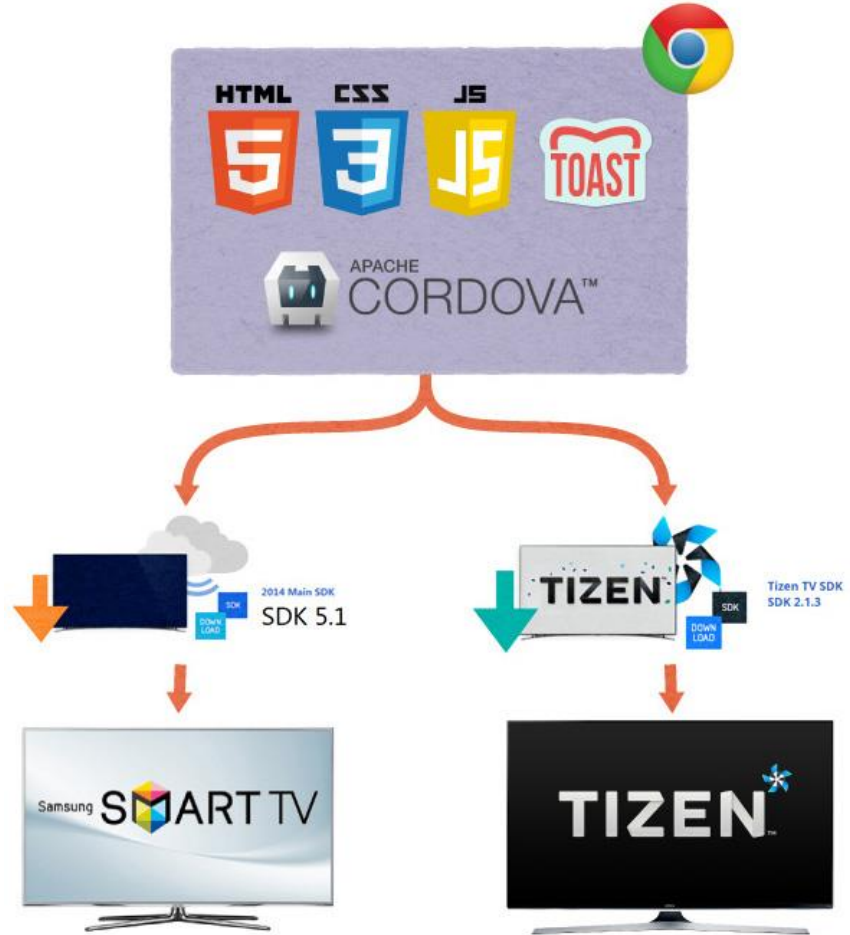# Cordova model

Smart TV as a Cordova platform

Smart TV as a Cordova platform

Development workflow

1. Development environment in PC for browser application

2. Test the SDK for each platform

3. Submit your app

TOAST Configuration

Introducing the TOAST plugin

Open source project

- Strategic reasons
  - Easy access
  - Contribution of external developers
  - Consideration of expansion to other TV platforms
- Technical reasons
  - Prompt update/release
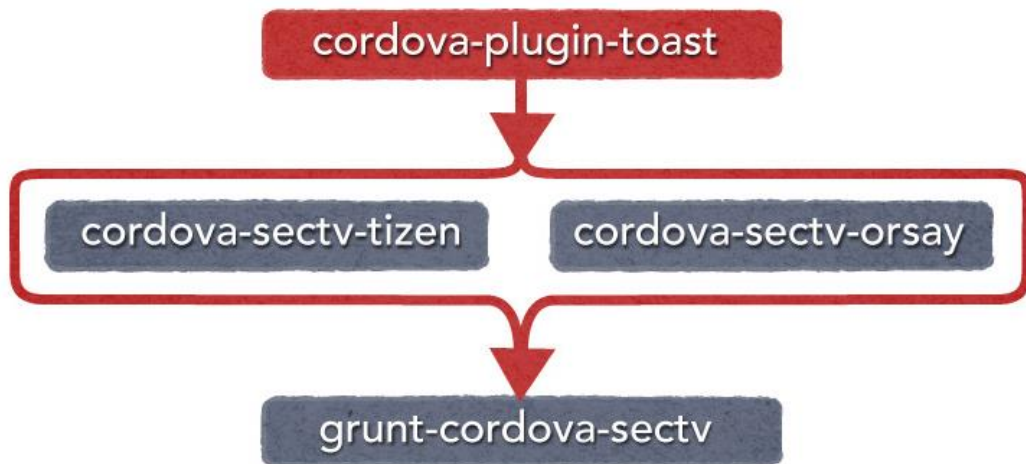 - Using public technology (Cordova)

History and plan for TOAST

- Samsung
  - Samsung Legacy TV (2012 ~ 2014)
  - Samsung Tizen TV (2015 ~ 2016)

- WebOS
  - v1.0, v2.0, v3.0 (2014 ~ 2016)

- Android TV
  - TBD

Github structure
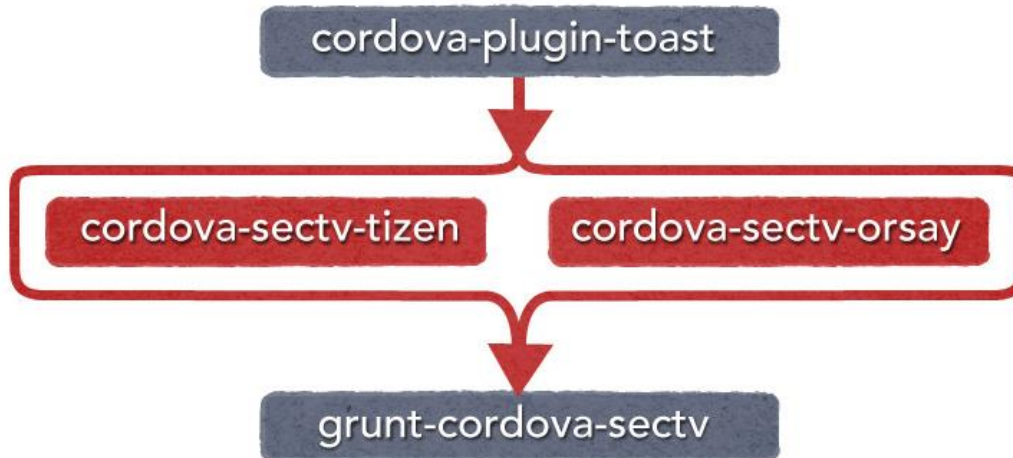
cordova-plugin-toast

• Cordova plugin that provides TV-specific functionalities.

• Defines a global ***toast*** object, which provides API for TV functionalities.

•***Toast*** object is not available until after the *deviceready* event which is provided by Cordova

TOAST Configuration: Github structure

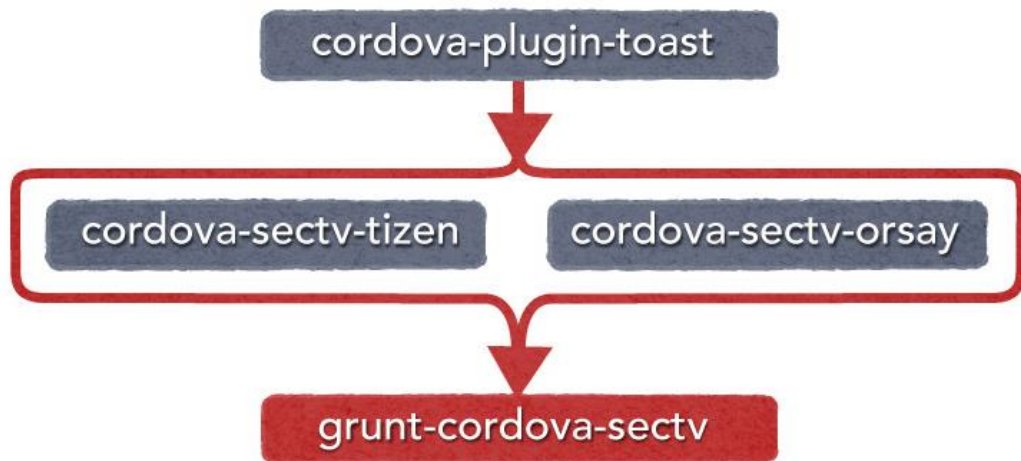## cordova-sectv-tizen & cordova-sectv-orsay

• TV application library that allows for Cordova-based projects to be built for the Samsung Smart TV Tizen OS and Legacy OS.

TOAST Configuration: Github structure

grunt-cordova-sectv

• Grunt task for preparing and building the cordova project with sectv-*** platforms.

Using TOAST for Smart TV

Getting started

Precondition

Before using TOAST, please install these:

- Node.js
- Git
- Chrome Browser
- Samsung Tizen SDK
- WebOS SDK
- *npm* modules: *cordova* and *grunt*

```
$ npm install -g cordova
$ npm install -g grunt-cli
```

Getting Started: Git clone

Establishing the enviroment for TOAST in Cordova

- In the root directory, please git clone the follo wing repositories.

```
$ git clone https://github.com/apache/cordova-js.git
$ git clone https://github.com/Samsung/cordova-plugin-toast.git
$ git clone https://github.com/Samsung/cordova-sectv-orsay.git
$ git clone https://github.com/Samsung/cordova-sectv-tizen.git
$ git clone https://github.com/Samsung/cordova-tv-webos.git
$ git clone https://github.com/Samsung/grunt-cordova-sectv.git
```

Getting Started: Install

- You will need to *npm install* dependency modules in *cordova-js*, *cordova-plugin-toast*, *cordova-sectv-orsay*, *cordova-sectv-tizen*, *cordova-tv-webos, grunt-cordova-sectv*.
- In *cordova-js/Gruntfile.js*, please add compile tasks.

```
module.exports = function(grunt) {

    grunt.initConfig({
        pkg: grunt.file.readJSON('package.json'),
        compile: {
                ...
                "sectv-orsay": {},
        "sectv-tizen": {},
                "tv-webos": {}
    },
```

Getting Started: Configuration

- In *cordova-js/package.json*, please add these to cordova platforms.

```
"cordova-platforms" : {
        ...
    "cordova-sectv-orsay" : "../cordova-sectv-orsay",
    "cordova-sectv-tizen" : "../cordova-sectv-tizen“,
        "cordova-tv-webos"    : "../cordova-tv-webos"
}
```

Getting Started: Compile

- In *cordova-js*, please grunt *compile* for packaging

```
$ grunt compile:sectv-orsay compile:sectv-tizen compile:tv-webos
```

- In *cordova-plugin-toast*,
please grunt *compile* for packaging.

```
$ grunt compile:sectv-orsay compile:sectv-tizen compile:tv-webos
```

Getting Started: Create project

- Creating an empty project

 # Create cordova project

```
$ cordova create TestApp
$ cd TestApp
```

 # Beware of hidden file

```
$ cp -rf ../grunt-cordova-sectv/sample/* ./
$ npm install ../grunt-cordova-sectv
```

 # Grunt task for build and package

 # Install dependency modules

```
$ npm install
```

 # For toast browser simulator

```
$ cordova platform add browser
```

Getting Started: API reference

TOAST API compatibility with platforms

Feature Name	Browser	Legacy OS			Tizen		webOS		
		2012	2013	2014	2015	2016	2014	2015	2016
toast.Media	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
toast.MediaPlugin	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
toast.drminfo	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
toast.application	Partly	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
toast.inputdevice	Partly	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
toast.tvaudiocontrol	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
toast.tvwindow	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
toast.tvchannel	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
IME	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

References

Information to start using TOAST

References: Where to start?

- TOAST Wiki
  - https://github.com/Samsung/cordova-plugin-toast/wiki

- Cordova TOAST plugin
  - https://github.com/Samsung/cordova-plugin-toast

- TV application library that allows for Cordova-based projects
  - https://github.com/Samsung/cordova-sectv-tizen
  - https://github.com/Samsung/cordova-sectv-orsay
  - https://github.com/Samsung/cordova-tv-webos
  - https://github.com/Samsung/grunt-cordova-sectv

- TOAST introduction page
  - http://www.samsungdforum.com/Features/TOAST

- TOAST video clip
  - http://www.samsungdforum.com/toastAD

References: Where to start?

- Samsung Developer Forum
  - http://www.samsungdforum.com/Features/TOAST

Q & A