Tizen Web Runtime Update

Ming Jin
Samsung Electronics
Table of Contents

• Quick Overview of This Talk
  ✓ Background, Major Updates, Upcoming Features

• What Have Been Updated
  ✓ Installation/Update Flow, WebKit2, Privilege Levels, Sandboxing, etc.

• Upcoming Features
  ✓ Content Security Policy (CSP), Navigation Policy, etc.
Quick Overview
Revisit: What Is Web App and Web Runtime

• **Web Application**
  • Using Web based technologies (HTML/JS/CSS/etc.)
  • Accessing local device / platform resources
  • Can be installed on the device

• **Web Runtime**
  • Package management (installation, update, etc.)
  • Execution and lifecycle (launching, pause, resume etc.)
  • Runtime security (API/network access, sandboxing, etc.)
  • Platform integration
Revisit: Web Apps on Tizen Devices

User → Interaction → Device Resources → Networking → Web
Revisit: Tizen Web App Ecosystem

Developer

SDK

Packaging & Signing

Web App

Upload

Store

Download & Install

Verification & Filtering

User

Tap

Launch

Play

Developer

Web App

SDK

Download & Install

Verification & Filtering

Store

Upload

Develop

Packaging & Signing
What Have Been Updated

- “Hosted” Web App support
- Updated installation and update flow
- WebKit2/EFL based process model
- Faster launching flow
- Update on suspend/resume, and localization
- Resource encryption/decryption
- Privilege-based API security
- Process sandboxing
- Web Dynamic Box support
Upcoming Features

• CSP (Content Security Policy)
• Navigation policy restriction
• AppCache download
Major Updates
Revisit: Web Runtime on Tizen

Tizen Web Framework
- Web UI F/W
- Web Runtime

Tizen Native Framework
- App / Shell
- Graphics / UI
- Content
- Web / Xml
- Base / Io
- Uix
- Media
- Messaging
- Net
- Telephony
- Social
- Text / Locales
- System
- Security

Core
- App Framework
- Graphics / UI
- Location
- Multimedia
- Messaging
- Web
- Security
- System
- Base
- Connectivity
- Telephony
- PIM

Linux Kernel and Device Drivers
Tizen Web Runtime Internal Blocks
Hosted Web App Support

• **What is “hosted” web app?**
  • A Web App with all of its resources located on a remote server
  • For now, it needs to be packaged with minimal resources: config.xml, signature(s)

• **Restrictions:**
  • Start page should be specified in <tizen:content> extension
  • All of rendered pages are remote pages
  • Only W3C/HTML APIs are allowed in remote pages
  • Created browsing context has the same origin as remote server origin
Installation Flow Update

• **New verification steps introduced:**
  • Higher-level privilege declaration is not allowed
  • Invalid signature is not allowed

• **New installation steps introduced:**
  • Resource encryption (if necessary)
  • Smack rules population

• **Deprecated steps:**
  • W3C Widget P&C `<feature>` tag checking is ignored (used as Store side filtering purpose for now)
Web App Update Flow

• **Update criteria:**
  • Must have the same Tizen App ID
  • Must have the same author signature

• **Update triggering point:**
  • At installation request time, if the request satisfies the “update criteria”

• **Update result:**
  • Web App data (settings, cookies, local storage etc.) must be preserved
Installation/Update Flow

1. **Installation Start**
   - Initial Config Parsing
   - Installation Task
   - Update Task
   - Unzip Package
   - Check Configuration
   - Check Certificate
   - Privilege Check
   - SMACK rules

2. **Needs Encrypt?**
   - [Yes]: Encrypt Resource
   - [No]: Install Files

3. **Is Update?**
   - [Yes]: Update Files
   - [No]: Install Files

4. **Update DB**
   - [Security Error]
   - [Invalid Certificate]
   - [Invalid Format]
   - [Decompression Failure]

5. **Installation Fail**
   - Installation Success
WebKit2/EFL based Web App Process Model

- Each Web App has 1 UI Process and 1 Web Process
  - UI Process manages lifecycle, and Web Process is responsible for rendering
  - NPAPI plugins (if any) will run in separate processes
Launching Procedure

- **Wrt-launchpad** is introduced to preload WebKit and WRT libraries

- **Caller application**
  - launch

- **Wrt-launchpad**
  - fork, UID & Smack
  - launch pid

- **AMD**
  - appcore events

**Wrt Client** (UI Process)
- Create Ewk Context
- Init AppCore
- Create Window
- Ewk View Add

**Web Process**
- WebProcess Init
- Plugin Preloading
- Resource Loading

**Libraries preloaded by wrt-launchpad:**
- libappcore-efl.so
- libappcore-common.so
- libisf-imf-module.so
- libewebkit2.so
- libinjected-bundle.so
- wrt-client
Suspend / Resume

• By default, webviews will be suspended by WRT when the Web App goes to background, unless the developer explicitly enables it via `<tizen:setting>`
  • Suspended activities include: JavaScript executions, timers, animations
  • Even if a webview is not suspended in background, the painting operations will not happen in background and the backing store will be purged
• When switched to foreground, the suspended webviews will be resumed
Localization

- **Folder-based localization**
  - Web App contents can be localized in file unit according to W3C Widget P&C specification
  - The localization requires page reloading, and the localized contents will only be reflected at next launching time

- **Element-based localization**
  - Configuration document can be localized with xml:lang attribute

- **Content-based localization**
  - System locale change event can be listened with Tizen System Info API
  - L10n routine needs to be implemented in locale change event handler
Resource Encryption / Decryption

• **Resources are encrypted during installation / update**
  - Enabled with `<tizen:setting encryption="enable"/>` in `config.xml`
  - Web App directory is scanned recursively
  - Only resources with predefined extensions (html / js / css) are encrypted
  - Information about encrypted resources are stored in WRT DB

• **Resources are decrypted at runtime**
  - UI Process informs Bundle (WebProcess) about the decryption necessity
  - Bundle performs resource decryption in `willSendRequestForFrameCallback`
  - Resources are decrypted to base64 string and read by WebKit
Resource Encryption at Install/Update Time

Web app resources
- config.xml
- resources

wrt-installer
(1) Configuration parser
(2) wrt-commons
(3) Installer
(4) wrt-commons
(5) ResourceEncryptor
(6) WrtDB

Data Flow
- Process
- DB
- Class/lib
- Modules
- Data Flow
- Call (Callback)
- IPC
- fork
Resource Decryption at Loading Time

- **UI Process**
  - wrt-client
  - WrтClient

- **Web Process**
  - wrt-commons
    - ResourceDecryptor
  - Webkit
    - Bundle
    - PageResourceLoadClient

**Data Flow**
- CALL (Callback)
- IPC
- fork

**Modules**
- Process
- DB
- Class/lib

**Tizen Developer Conference 2013 San Francisco**
Privilege Levels and API Permissions

- **Untrusted**
  - Location, web notification, media capture, full screen, unlimited storage
- **Public**
  - Alarm, application, bluetooth, calendar, call history, contact, file system, message port, messaging, power, push, setting, system, etc.
- **Partner**
  - App manager, secure element, system manager, etc.
- **Platform**
  - Bluetooth manager, lock manager, package manager, setting manager, etc.
Process Sandboxing

- **Web App (UI/Web Process) Sandboxing**
  - UID: “app”
  - GUID: “app”
  - Process Smack label: {PkgID}
  - Resource Smack label: {PkgID}
  - Smack rules: basic application rules + privilege specific rules
- **Web App private data** (e.g., localstorage, indexed db, cookie, etc.) is protected from the rest of system after sandboxing
Web Dynamic Box

• Features:
  • Included in a Web App package
  • Installed, updated, and uninstalled as part of Web App package
  • Supported sizes: 1x1, 2x1, 2x2
  • Supports periodic update: minimum 30 min term
  • Embeddable in viewer-like applications (e.g., home screen)
Tizen Configuration Extensions Update

```xml
<tizen:application id="PkgA.MyApp" package="PkgA" required_version="2.0" />

<tizen:privilege name="http://tizen.org/privilege/application.launch" />

<tizen:app-control>
    <tizen:src name="edit.html" />
    <tizen:operation name="http://tizen.org/appcontrol/operation/edit" />
    <tizen:mime name="image/jpg" />
    <tizen:uri name="" />
</tizen:app-control>

<tizen:setting background-support="enable" />
<tizen:setting context-menu="disable" />
<tizen:setting encryption="enable" />
<tizen:setting screen-orientation="landscape" />
<tizen:setting install-location="internal-only" />

<tizen:content src="https://www.tizen.org/" />
```
Typical Web App Directory Structure (on Device)

- **{PackageId}**
  - **bin**
  - **res**
  - **data**
  - **shared**
  - **wgt**
  - **config.xml**
  - **signature files**
    - **start file (index.html)**
  - **plugins**
    - **arm**
    - **x86**
    - **{NPAPI}**
  - **data**
  - **trusted**
  - **Additional files or directories**

Packaged Web App Specific Files
Coming Soon
Content Security Policy (CSP)

• New fields required in config.xml:
  • E.g.: `<tizen:content-security-policy>`, `<tizen:content-security-policy-report-only>`

• Possible default policy:
  • "default-src '*'; script-src ‘self’; style-src ‘self’; object-src ‘none’;"

• A Web App can define a fine-grained CSP policy by using different directives:
  • E.g., script-src, style-src, img-src, connect-src, frame-src, etc.
Navigation Policy

• New field required in config.xml:
  • E.g., <tizen:allow-navigation>
• Top-level window URL navigation is limited to the list of domains specified in <tizen:allow-navigation>.
• Navigation to any other domains (not listed in <tizen:allow-navigation>) may result in opening up the URL in Browser.
AppCache Download

- New field required in config.xml:
  - E.g., `<tizen:appcache-manifest-url>`
- Upon installation, download the appcache resources specified in appcache manifest URL
- Useful for hosted web apps to work offline or provide fallback routines for offline