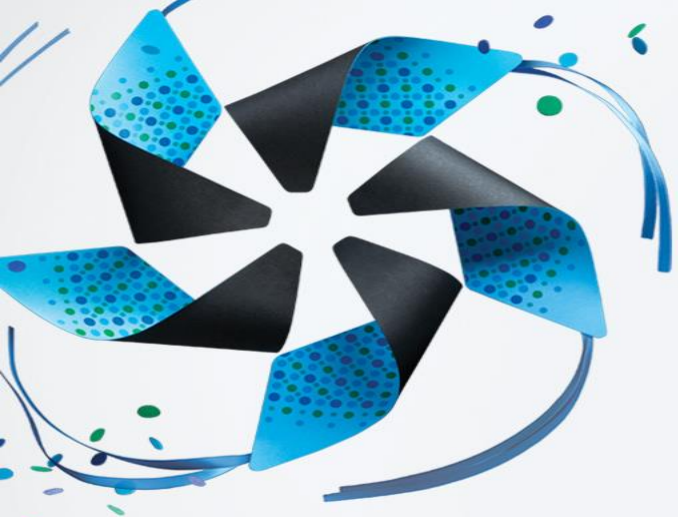


Tizen Compliance

Mats Wichmann
Open Source Group
Samsung Research
mats@osg.samsung.com

A World of Smart Devices

SAMSUNG
Open Source Group

Tizen already can run on all of these!

- **What does this mean for an application developer?**
  - It's all just HTML5, right? ☺
- **A bit more to the story**

Compliance???

Compliance means acting in accordance with a rule, policy, regulation, standard or law.

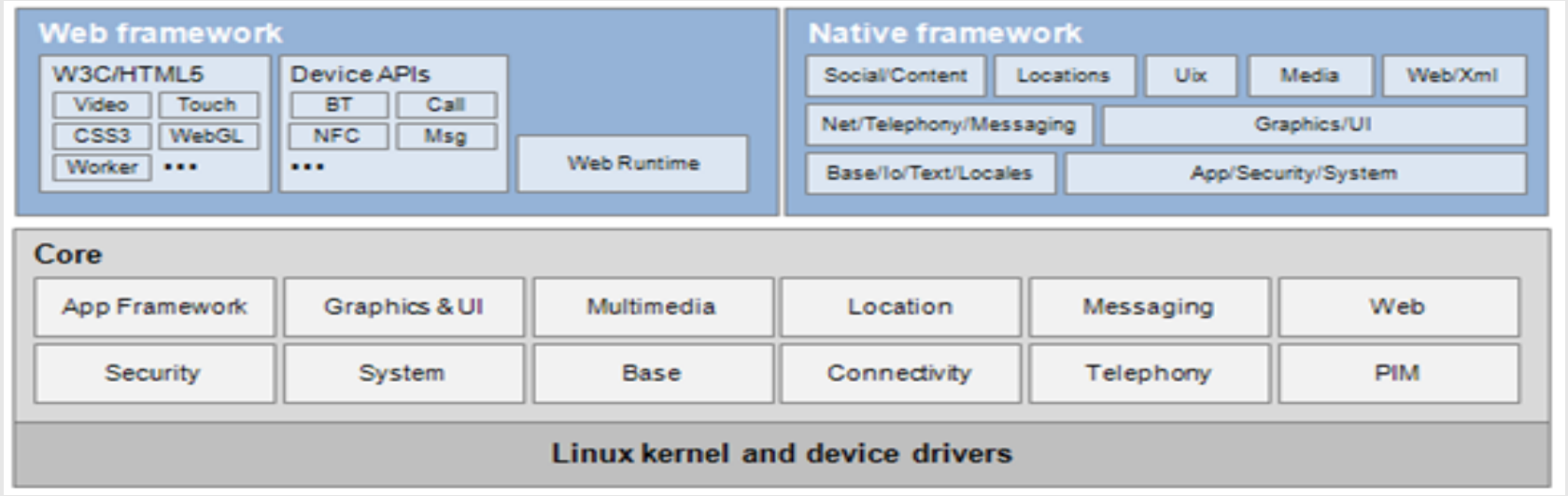
Example: A company compliance program is designed to ensure that a company and its employees act in accordance with all laws, rules and regulations necessary to perform their duties and thereby minimize the risk presented by violation of such laws, rules, and regulations

For Tizen, the compliance program is designed to ensure that implementations of the platform are really Tizen, and will run apps that conform to the standard.

So What makes Tizen?

- **Web Frameworks**
  - HTML5/W3C APIs, additional Tizen device APIs and Web UI Framework
- **Native Frameworks**
  - Full-featured native application development with feature set similar to Web, can also use EFL and other supported native libs
- **Core**
  - Core subsystems required by the web and native frameworks

Tizen Architecture



Terms

- **Tizen *Platform***
  - The code for a Tizen release, such as “2.3”
- **Tizen *Profile***
  - Adaptation of platform to a specific device category
- **Tizen Compliance *Profile***
  - The application programming interface and application execution environment for a specific device category (e.g “Tizen Mobile”).
- **Tizen Compliance Specification (*TCS*)**
  - The document which defines a compliance profile
- **Tizen Compliance Tests (*TCT*)**
  - measures how well a device matches the TCS

Tizen Compliance Goals

- **Reduce fragmentation**
- **Enable applications written for one platform to run on another platform**
- Clarify what “Tizen” means via specification and test suite.
- Communicate Tizen platform and product features
- Increase quality of Tizen products and Tizen brand
- Increase confidence to participate from partners, operators, OEMs, ISVs, and developers

Tizen Compliance In Brief

- A subset of Tizen platform, presented as a supported API
- Devices promise to provide
  - They can’t call themselves Tizen otherwise
- Apps get compatibility promise if they code to this API set

Tizen Compliance Spec (TCS)

A TCS describes a profile-specific App Environment

- **Mandatory / optional software:**
  - APIs, codecs, …
- **Mandatory / optional hardware:**
  - Connectivity, display, ..
- **Portability aids**
  - list of application controls
  - list of feature keys
  - list of privileges
- **One TCS per profile**
  - Try to keep #profiles to a minimum
  - A dozen profile-TCS/TCT pairs would not be great…
  - A refrigerator is Mobile, right?

Example of TCS Items

Public API
Implicit API (AppControl)
Execution environment (WRT)
Common user interface
app's resource related (codec)
Interface for common tools
Events which apps receive
Common HW
Common security model
Common input method
Others (only if impacts application compatibility on devices)

Tizen Compliance Program

- **Tizen certification registers the compliance of a device**
  - Certification is against a specific Tizen Profile
  - Pick the best fit where device meets the requirements
- **Intentionally designed as a lightweight program**
  - Self-test
  - Not restrictive: plenty of room for customization outside the scope of the TCS
- **Application developers**
  - TCS guides how to correctly target a profile
  - No separate certification for apps
  - However, compliance of app may be an acceptance requirement of app stores

Tizen Mobile Profile Overview

- Defined for various mobile devices
- Flexibility to cover low to high-end, and could be used for other devices like camera and appliance
- 2.3 includes both Native and Web APIs
- Full version of source code is available at part of 2.3 platform at http://source.tizen.org

Tizen Profile – Product Mapping

Current

Mobile

Wearable

In Progress / Considered

Phones / Tablets
(Mobile Profile)

Wearables
(Wearable Profile)

Automobile
(IVI Profile)

TVs
(TV Profile)

Cameras
(Mobile Profile*)

Appliances
(Mobile Profile*)

*TBD, but application runtime requirements are the same as mobile devices so *could* share the same profile..

Tizen 2.3 Profiles

All

- W3C/HTML5 APIs, Tizen mandatory Web Device API modules

Mobile Profile

- Native API
- User Interface sections including Theme and Notification Tray
- HW requirements for mobile incl RAM size, Camera, Display resolutions and matching themes, NFC, Wi-Fi Direct
- Developer Tools: Dynamic Analyzer
- Mandatory platform AppControls

Tizen 2.3 Profiles

Wearable Profile

- Use Samsung Extension to offer new API modules, e.g. IrLED and SAP
- Download API
- Wearable-appropriate display resolutions, hardware

TV profile:

- Support for remote control
- TV-appropriate display resolutions, matching themes, hardware

IVI Profile

- NOT 2.3: Uses features only in Tizen 3.0, like multi-user:
  - each “seat” in vehicle may need different settings and behaviors

Tizen Profile Strategy

The following profiles have been defined:

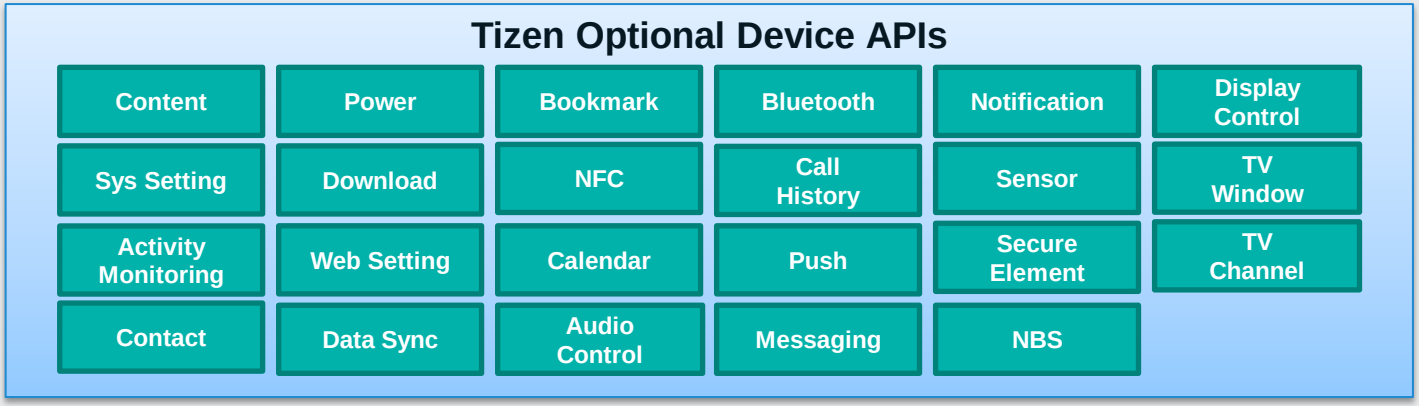
- Mobile: low to high-end phones and tablets
- IVI: automobile console and entertainment system
- Wearable: wearable device adaptation (partial in 2.2, full in 2.3)
- TV: smart televisions

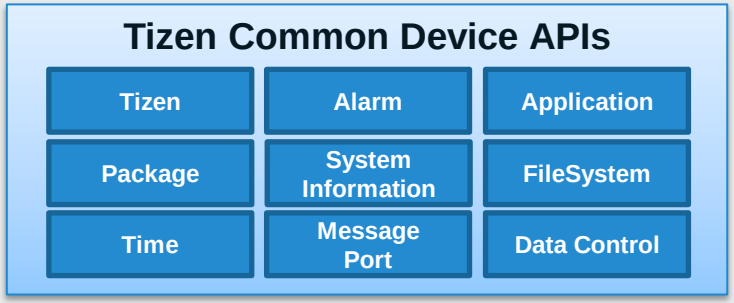
Strategy:

- Limit number of profiles as much as possible
- Each profile has specific feature keys to identify features present/absent
  - applications indicate what features they need

Common (required) and Optional Web Device API

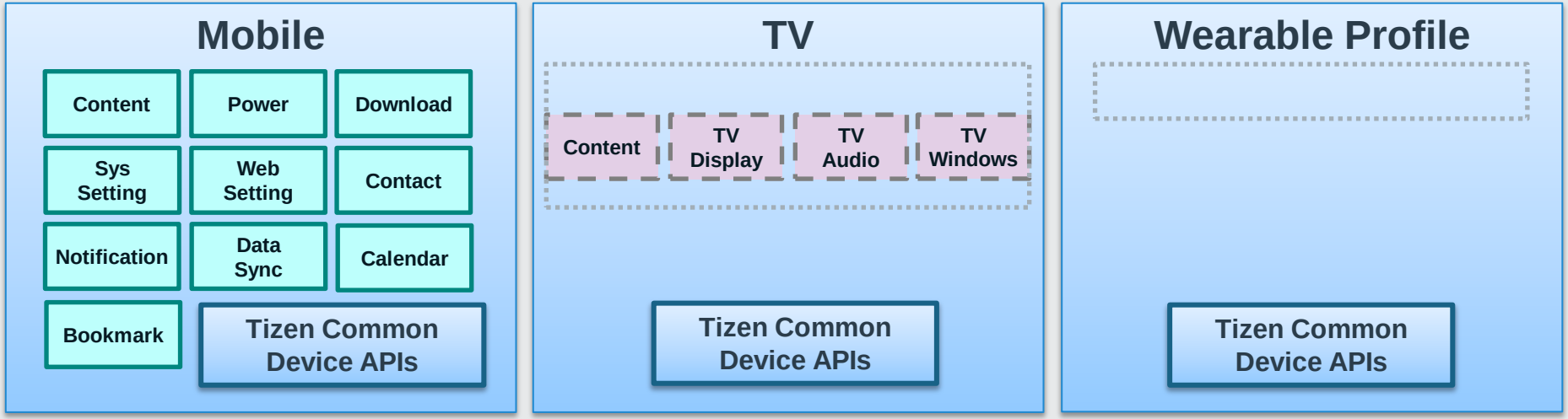
- All profiles that support the Web API must provide Tizen Common Device APIs.
- Each profile includes additional mandatory APIs. (next slide).
- Tizen optional device APIs may be included in any profile.





Profile-specific Device APIs

- Each profile includes mandatory APIs (beyond common).
- A device can include/exclude any optional device APIs.
- If an optional API is supported, it must be reported by the “feature” mechanism.
- The app store client shows apps that match the device’s supported features.



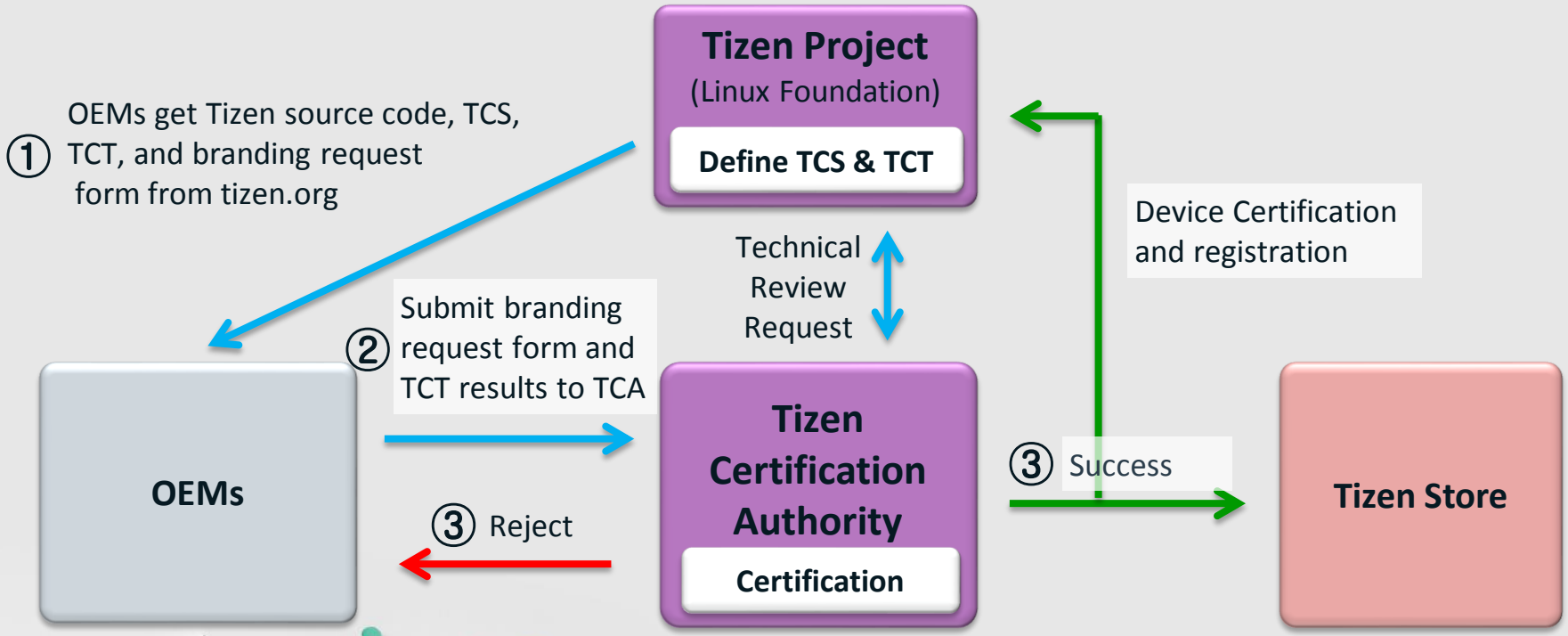
* Profile requirements are not finalized. They will be available when the platform is released.

Apps Running on Multiple Profiles: Web only

Create app manifest (config.xml) before uploading to Tizen Store

- Specify the profiles the app will run on (at least one)
- Specify the device features the app requires to run (e.g. Bluetooth)
  - The store client will not show the app if a device does not support a listed profile or all the features required to run the app.

Tizen Compliance: Workflow



Tizen Native API

- **As described in other sessions at TDS there is also a “new” Native API**
- **Today, Native API is only in 2.3 Mobile Profile**
  - Support in other profiles is under consideration – not a technical problem, up to the profile teams
- **Native has a different packaging format (.tpk files) and a different configuration document**
- **Compliance considerations are generally the same**

Thank you.

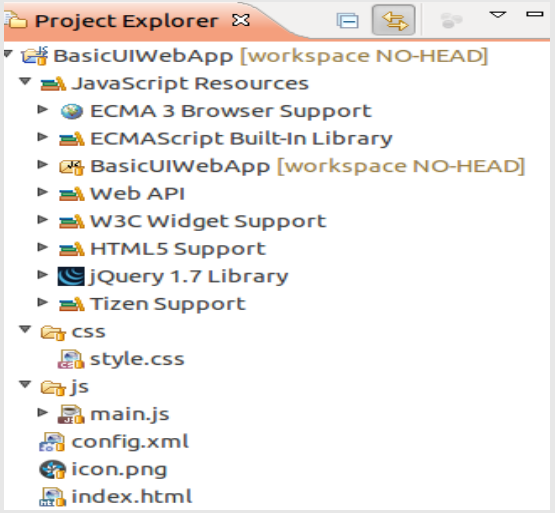
Questions?

http://source.tizen.org/compliance (lists only 2.2.1 today)
Mailing list: http://lists.tizen.org/listinfo/compliance

Backup

Web Application Packaging

- **Packaging**
  - W3C Packaged Web Apps (Widgets) spec
- **Configuration doc. config.xml**
  - Application basic info (ID, icon, name, …)
  - Localization
  - Preferences
  - Features and Privileges




```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:tizen="http://tizen.org/ns/widgets" id="http://yourdomain/BasicUIWebAPp" versior
    <tizen:application id="RM2Gaf4ztV.BasicUIWebAPp" package="RM2Gaf4ztV" required_version="2.2"/>
    <content src="index.html"/>
    <feature name="http://tizen.org/feature/screen.size.all"/>
    <feature name="http://tizen.org/feature/network.bluetooth"/>
    <icon src="icon.png"/>
    <name>BasicUIWebAPp</name>
    <tizen:privilege name="http://tizen.org/privilege/application.info"/>
</widget>
```

If feature not mandatory for all, test for it:

- Example: fallback for Geolocation API

```
if ('geolocation' in navigator)
{
    geolocation.getCurrentPosition =
        function(successCallback, errorCallback, options)
}
else
{
    // ask user to input current location manually
    // and use google map service to get complete address
}
```