Understanding the Permission and Access Control Model for Tizen Application Sandboxing

Onur Aciciemez & Andrew Blaich, Samsung
Motivation

- **The Tizen application model** is based on Web technologies:
  - HTML5 + JS + CSS + Web APIs
- **Tizen WRT** supports **Tizen widgets** and multiple APIs: **W3C**, **non-W3C** (e.g. WebGL) and **Tizen Web API**
- **Web-Runtime** is the application that handles widget **installation** and **execution**
- **Security** of **WRT** and **widgets** is crucial for the ecosystem
- **Our talk**:
  - Overview of Tizen Security Framework and SMACK (Simplified Mandatory Access Control Kernel)
  - Widget access control and permissions
  - WebRunTime access control enforcement
  - Widget Sandbox
Overview of Tizen Security Framework

- **SMACK** as the main system-level access control mechanism
- **Web Runtime** enforces fine-grained controls over Tizen WebApps
- **SMACK-based** process sandbox over widget processes
Contents

• Overview of the Tizen Security Framework
  – SMACK Overview
• Widget Permissions and Access Control Model
  – Feature Declarations in Manifest
  – User Prompt Types
  – Widget Access Request Policy (WARP)
  – Sample Manifest and Policy Files
• Setting Security Configurations in Tizen SDK
• Access Control Enforcements on Tizen WebApps
  – WRT Access Control Engine
  – SMACK Sandbox
• Conclusions
SMACK Overview

- **Simplified Mandatory Access Control Kernel**
  - Linux Security Module included in the Linux Kernel

- **SMACK Terms:**
  - **Subject**
    - an active entity that performs the access
  - **Object**
    - a passive entity that is accessed
  - **Access**
    - an access attempt from Subject to Object
  - **Label**
    - a “security tag” applied to subjects (i.e., processes) and objects (i.e., file-system objects, sockets, processes). Used to identify the entity SMACK
SMACK Overview

- **SMACK Labels:**
  - Two label types: process labels and object labels
  - Extended file attributes to store SMACK label configuration
    - **SMACK64:** XATTR for file-system objects
    - **SMACK64EXEC:** XATTR for executables. Becomes process label upon exec()

- **SMACK Accesses:**
SMACK Overview

- **SMACK Rules:**
  - **Rule format:**
    - `[subjectLabel] [objectLabel] [access(rwxa)]`
  - `/usr/bin/cat → SMACK64EXEC = catApp`
  - `/home/user/documents/file1.txt → SMACK64 = myFile`
  - Example Rule to allow cat to read file1.txt
    ```
    catApp myFile r
    ```
  - Rule to allow cat to read & write file1.txt
    ```
    catApp myFile rw
    ```
Widget Permissions and Access Control Model

- A **subset** of the **JavaScript APIs** supported in Tizen are considered **restricted**
  - **Restricted** refers to any JS function that can **access** the **private data** on a **device** such as location, contacts, calendar, etc.
- **Widgets** need **authorization** to invoke **restricted APIs**
- Permission declarations and authorization:
  - Declaration in **manifest file**:
    - `<feature>` element for device APIs
    - `<access>` element for network resources
  - Authorization:
    - **prompt type** decision according to WRT ACE **policy**
    - **user confirmations**
Widget Permissions and Access Control Model

- **Developers** must declare in the **manifest file** of a widget, which **features** the widget wants access to.

**Feature Declaration “template” from W3C**

```xml
<widget xmlns="http://www.w3.org/ns/widgets">
   <feature name = "http://example.com/api/contact" required = "false"/>
</widget>
```

**Feature Declaration “implementation” for Tizen**

```xml
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:tizen="http://tizen.org/ns/widgets" version="1.0">
   <feature name="http://tizen.org/api/contact" required="false"/>
</widget>
```
## Widget Permissions and Access Control Model

<table>
<thead>
<tr>
<th>API Group</th>
<th>Feature / Device Capability</th>
<th>API Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td><a href="http://tizen.org/api/time">http://tizen.org/api/time</a></td>
<td>All</td>
</tr>
<tr>
<td></td>
<td><a href="http://tizen.org/api/time.read">http://tizen.org/api/time.read</a></td>
<td>All except setCurrentDateTime()</td>
</tr>
<tr>
<td></td>
<td><a href="http://tizen.org/api/time.write">http://tizen.org/api/time.write</a></td>
<td>setCurrentDateTime()</td>
</tr>
</tbody>
</table>

### JavaScript:

```javascript
...  
var current_dt = tizen.time.getCurrentDateTime();
var is_leap = tizen.time.isLeapYear(current_dt.getFullYear());
if (is_leap)
    console.log("This year is a leap year.");
...
```

### Manifest File:

```xml
...  
<feature name="http://tizen.org/api/tizen"/>  
<feature name="http://tizen.org/api/time.read"/>  
...
```

*See Appendix for the full Tizen Web API list*
Widget Permissions and Access Control Model

- **W3C Widget Access Request Policy (WARP)**
  - All network accesses by widgets are denied by default
  - A widget must declare in its manifest which network resources it will access (such as XMLHttpRequest, iframe, img, script, etc.)
  - `<access>` element in config.xml. Developers can specify protocols, domains, and sub-domains.

```xml
<widget xmlns="http://www.w3.org/ns/widgets">
  ...
  ...
  <access origin="https://example.net"/>
  <access origin="http://example.org" subdomains="true"/>
  <access origin="http://example.org:8080" subdomains="false"/>
  <access origin="http://example.com"/>
  ...
</widget>
```
Sample Manifest file:

```xml
<?xml version="1.0" encoding="UTF-8"?>

<widget xmlns="http://www.w3.org/ns/widgets" xmlns:tizen="http://tizen.org/ns/widgets" version="1.0"
    id="http://YourDomain.com/SampleContact" viewmodes="fullscreen">
    <icon src="icon.png"/>
    <name>SampleContact</name>
    <content src="index.html"/>
    <description>Sample application for Tizen contact module.</description>
    <license/>

    <feature name="http://tizen.org/api/tizen" required="true"/>
    <feature name="http://tizen.org/api/contact" required="true"/>
    <feature name="http://tizen.org/api/contact.read" required="true"/>
    <feature name="http://tizen.org/api/contact.write" required="true"/>

    <access origin="http://jquerymobile.com" subdomains="true"/>
</widget>
```
A feature will be granted by the WRT based on the policy and the confirmation of the user to various prompt types

- Various types of prompts are available (table)
- WRT ACE Policy specifies which prompt type will be used in a specific situation

### Prompt Types

<table>
<thead>
<tr>
<th>Prompt Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blanket Prompt</td>
<td>User is prompted for confirmation the first time the API function is called by the widget, but once confirmed, prompting is never again required.</td>
</tr>
<tr>
<td>Session Prompt</td>
<td>User is prompted once per session.</td>
</tr>
<tr>
<td>One-Shot Prompt</td>
<td>User must be prompted each time the restricted API is invoked.</td>
</tr>
<tr>
<td>Permit</td>
<td>Use of the device capability is always permitted, without asking the user.</td>
</tr>
<tr>
<td>Deny</td>
<td>Use of the device capability is always denied.</td>
</tr>
</tbody>
</table>
Widget Permissions and Access Control Model

- The **type of prompt** for each **API** is **determined** by the **policy**
- **Policies** are **driven** by **Operators** and **Device Manufacturers**
- Users can affect a policy through preference configuration, but only in a more restricted way

### Blanket Prompt
- Widget requires access to: contact.read
- Keep setting as permanent
- Deny | Permit

### Session Prompt
- Widget requires access to: contact.read
- Remember for one run
- Deny | Permit

### One-Shot Prompt
- Widget requires access to: contact.read
- Deny | Permit
Sample Tizen Policy File

```xml
<policy-set id="Tizen-Policy" combine="first-matching-target">
  <policy id="Tizen-Policy-Trusted" description="Tizen's policy for trusted domain" combine="permit-overrides">
    <rule effect="prompt-session">
      <!-- rules for specific resources -->
      <condition combine="and">
        <condition combine="or">
          <resource-match attr="device-cap" func="equal" match="XMLHttpRequest"/>
          <resource-match attr="device-cap" func="equal" match="externalNetworkAccess"/>
          <resource-match attr="device-cap" func="equal" match="messaging.send"/>
        </condition>
        <environment-match attr="roaming" match="true"/>
      </condition>
    </rule>
    <rule effect="permit">
      <!-- all other matches -->
    </rule>
  </policy>
</policy-set>
```
Setting Security Configurations in Tizen SDK

- **Tizen SDK** supports **feature** selection
  - **Developers** need to **manually choose** which **features** their applications require
- A **check box** on the **left** of a feature name indicates the "**required**" attribute. If this is **checked**, **config.xml** is as follows.
  ```xml
  <feature
    name="http://tizen.org/api/accelerometer"
    required="true"/>
  ```
- **Add Feature Dialog Box** allows a feature to be added in one of 3 ways:
  - **Internal**: It is possible to select a feature from a fixed list.
  - **Feature name**: A URL with a feature definition should be entered.
  - **File**: A name of a file with a feature definition (*.xml, *.wdlprocxml) should be entered.
Applications **CANNOT** access external network resources by default (WARP - W3C Access Requests Policy).

Developers **must request permissions** for their widget to retrieve network resources.

You can **enter multiple URLs** using the Add button.

For each **URL**, you can **indicate** if you want to allow a widget to access the **subdomains** for a URL. The "Allow subdomain" column contents can be toggled with a mouse click.

**Manifest file:**

```xml
<access origin="http://jquerymobile.com" subdomains="true"/>
```
Access Control Enforcements on Tizen WebApps

- **Tizen WRT** supports **Tizen WebApps** and multiple APIs: **W3C APIs**, and **non-W3C APIs** like WebGL and **Tizen Web API**
- **WRT** has a **multi-process** model
  - **WebKit** based
  - **Widget instances** are executed in **separate processes**
  - Provides **runtime isolation** and allows the system to **enforce custom process-level containment** (sandbox) on each **instance**
- Two levels of access control enforcements
  - **WRT Access Control Engine (ACE):** Fine grained access control on JS APIs
  - **Application Sandbox via SMACK:** Process-level containment by the kernel on system calls
Access Control Enforcements on Tizen WebApps

• Access Control Engine (ACE) – General Design

PEP: ACE interface for WRT
PIP: Responsible for obtaining attribute values from WRT, Resource Information and OS
PDP: Policy Decision Point, evaluates policies; Interacts with the user if necessary
Policy Translator: Parses policies (XML)
Verdict Manager: Responsible for caching the verdicts
Access Control Enforcements on Tizen WebApps

• **ACE Policy Evaluation – General Flow:**

  - Widget requests access to JS API
  - WRT formulates a query to ACE
  - PEP receives a request from WRT
  - Subject Advocate forwards a request to Policy Evaluator
  - Policy Evaluator checks for request attributes in Verdict Manager
  - Policy Evaluator obtains attributes value

  - Policy Evaluator checks verdict in ACE database with Verdict Manager
    - Yes: verdict is in the ACE DB
    - No: verdict is NOT in the ACE DB

  - Get the verdict from Verdict Manager
  - Policy Evaluator evaluates the policies
  - Store verdict in Verdict Manager
  - PDP returns a verdict

  - User interaction?
    - Yes: Prompt Window is shown to the user
    - No

  - User selects a reply from the window

  - Subject Advocate returns verdict to PEP
  - PEP returns verdict to WRT
Access Control Enforcements on Tizen WebApps

• **Widget Process Sandbox via SMACK**
  The **SMACK Policy File** is updated with the appropriate **rules** for a **widget** during the **install**, **update**, or **uninstall** operations, as well as **at run-time**. The rules are based on the device features a widget requests in the **manifest file** packaged with a widget, **user confirmations**, and **security files** on the system that describe what **labels** and **permissions** are needed for each **device feature**.

  ![Flowchart Diagram]

  - **No**: denied by ACE
  - **Yes**: allowed by ACE

  **Intercept system calls with SMACK**

  - **Allow** or **Deny** access based on the **subject**, **object**, **permission** 3-tuple

  - Corresponding files on the system have been **labeled**
Access Control Enforcements on Tizen WebApps

• Why do we sandbox widget processes?
  – WebKit vulnerability analysis results
    • CVE: Common Vulnerabilities and Exposures

![Graph showing Total CVEs for WebKit and Total CVEs over years from 2006 to 2012.]

![Bar chart showing CVE mean score for WebKit over years from 2006 to 2012.]

![Bar chart showing access control enforcements on Tizen webapps over years from 2006 to 2012.]

![Table showing years and corresponding total CVEs for WebKit and total CVEs.]

![Bar chart showing high, medium, and low access control enforcements from 2006 to 2012.]

22
Access Control Enforcements on Tizen WebApps

- Why do we sandbox widget processes?
  - WebKit vulnerability analysis results
Access Control Enforcements on Tizen WebApps

- **Widget Sandbox via SMACK:**
  - Each **widget** runs in a **different security domain** (they have unique SMACK labels).
  - A **widget** process cannot access the files of another **widget**, system files (such as a contacts database), or communicate with other processes (such as a telephony daemon) unless the required SMACK rules are in place.
  - **SMACK rules** for a widget are configured:
    - during **install**, **uninstall**, and **update** operations by Package Manager.
    - at **runtime** by the WRT Security Server.
    - as a result of **user prompts** according to which features are granted to that widget.
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - "Widget A" contains the following code snippet:
    ```java
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ...
    ```
  - Read access to the Contacts DB file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)

- **Read** access to the **Contacts DB** file
- **Assume device policy requires blanket prompt** (*depends on the actual policy on the device)
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
  ```java
  ... addressbook = tizen.contact.getDefaultAddressBook();
  addressbook.find(...);
  ...
  ``
  - **Read** access to contacts DB file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device*)

http://tizen.org/api/tizen
http://tizen.org/api/contact.read
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    
    ```
    ...
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ...
    ```
  - Read access to **contacts DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)

- **Flow:**
  - Widget A contains this code snippet
    ```
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ```
  - Read access to **contacts DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)

- **Flow:**
  - Widget A contains this code snippet
    ```
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ```
  - Read access to **contacts DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)

- **Flow:**
  - Widget A contains this code snippet
    ```
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ```
  - Read access to **contacts DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    ```java
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ...
    ```
  - **Read** access to contacts DB file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)

  - **Assume device policy**
  - **SMACK policy**
    - com.tizen.widgetA
    - com.tizen.widgetB
    - Contacts DB
    - Other System Files

  - **Contacts DB**
  - **Other System Files**

  - **SMACK policy**
    - com.tizen.widgetA
    - com.tizen.widgetB
    - Contacts DB
    - Other System Files

  - **Set label to “widgetA”**
    - Rule: “widgetA contacts r”

  - **Launch widget**
    - AUL
    - Menu Screen

  - **WRT**
    - Widget A
      - Parental Mode
      - Certificate
      - Policy-based Access Control
      - Local Resource Control

  - **User Prompt**
    - Yes
    - User

  - **App Store**
    - Install widget
    - Purchase

  - **Package Manager**

  - **Client**

  - **AppStore**

  - **Dev Site**
    - Upload

  - **SDK**
    - Develop

  - **Manifest**

  - **User**

  - **Developer**

  - **http://tizen.org/api/tizen**
  - **http://tizen.org/api/contact.read**
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
  - **Read** access to **contacts DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device*)

```java
... addressbook = tizen.contact.getDefaultAddressBook();
addressbook.find(...);
...
```

- **Read** access to **contacts DB** file
- Assume device policy requires **blanket prompt** (*depends on the actual policy on the device*)

http://tizen.org/api/tizen
http://tizen.org/api/contact.read
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    ```java
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ...
    ```
  - Read access to contacts DB file
  - Assume device policy requires blanket prompt (*depends on the actual policy on the device)

  ![Diagram of Tizen WebApps access control](http://tizen.org/api/tizen)
  ![Diagram of Tizen WebApps access control](http://tizen.org/api/contact.read)
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    ```javascript
    ... addressbook = tizen.contact.getDefaultAddressBook(); addressbook.find(...);
    ...
    ```
  - **Read** access to `contacts DB` file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device*)

  - **User Prompt**
    - Yes
    - No

  - **Launch widget**
    - **Menu Screen**

  - **Install widget**
    - **AppStore Client**
    - **Package Manager**

  - **Upload**
    - **App Store**
    - **Certification**
    - **Code signing**

  - **Upload**
    - **Dev Site**
    - **SDK**

  - **Develop**
    - **Addressbook**
    - **Manifest**

  - **Developer**
    - **App Store**
    - **App Developer**

  - **Tizen WebApps**
    - **http://tizen.org/api/tizen**
    - **http://tizen.org/api/contact.read**

  - **Certification**
    - **Code signing**

  - **Set label to “widgetA”**

  - **Rule: “widgetA-contacts r”**

  - **Label for Widget A:**

  - **Other System Files**
    - **Contacts DB**
    - **Other**

  - **SMACK policy**
    - **com.tizen.widgetA**
    - **com.tizen.widgetB**

  - **Parental Mode**
    - **Certificate**
    - **Policy-based Access Control**
    - **Local Resource Control**
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    ```
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ...
    ```
  - Read access to **contacts DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)

  - **Read** access to **contacts DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    ```java
    ... addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ...
    ```
  - **Read** access to contacts DB file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)*

- **Flow:**
  - Widget A contains this code snippet
  - **Read** access to contacts DB file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)*

- **Related API:**
  - [http://tizen.org/api/tizen](http://tizen.org/api/tizen)
  - [http://tizen.org/api/contact.read](http://tizen.org/api/contact.read)
**Access Control Enforcements on Tizen WebApps**

**• SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    ```
    ... addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    ...
    ```
  - **Read** access to **contacts DB file**
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device*)

  ![Diagram showing access control on Tizen WebApps]

  - **Label for Widget A:** “widgetA”
  - **Rule:** “widgetA contacts r”

  - **SMACK policy**: com.tizen.widgetA, com.tizen.widgetB

  - **Contacts DB**: Contacts DB
  - **Other System Files**: Contacts DB

  ![Diagram showing access control on Tizen WebApps]

  - **WRT Initializer**: User Prompt
    - **AUL**: Menu Screen
  - **App Store**: Install widget
    - **AppStore Client**: User Prompt
      - **Package Manager**: Install widget

  ![Diagram showing access control on Tizen WebApps]

  - **Upload**: Certification Code signing
  - **Dev Site**: Set label to “widgetA”
  - **Dev Site**: Launch widget

  ![Diagram showing access control on Tizen WebApps]

  - **Flow**:
    - **– Widget A contains this code snippet**
    - **– Read** access to **contacts DB file**
    - **– Assume device policy requires **blanket prompt** (*depends on the actual policy on the device*)

  ![Diagram showing access control on Tizen WebApps]

  - **http://tizen.org/api/tizen**
  - **http://tizen.org/api/contact.read**
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet

  ```java
  ... addressbook = tizen.contact.getDefaultAddressBook();
  addressbook.find(...);
  ...
  ```

  - **Read** access to `contacts` DB file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device)*

  - [Read access to contacts DB file](http://tizen.org/api/tizen)
  - [Assume device policy requires blanket prompt](http://tizen.org/api/contact.read)

  ![Diagram](http://tizen.org/api/tizen)

  ![Diagram](http://tizen.org/api/contact.read)
Access Control Enforcements on Tizen WebApps

- **SMACK Sandbox Example Flow:**
  - Widget A contains this code snippet
    ```
    …
    addressbook = tizen.contact.getDefaultAddressBook();
    addressbook.find(...);
    …
    ```
  - **Read** access to **contacts** **DB** file
  - Assume device policy requires **blanket prompt** (*depends on the actual policy on the device*)

- **Flow:**
  - Widget A contains this code snippet
  - Read access to contacts DB file
  - Assume device policy requires blanket prompt (*depends on the actual policy on the device*)
  - Set label to “widgetA”
  - Rule: “widgetA contacts r”

### Code Snippet:
```java
addressbook = tizen.contact.getDefaultAddressBook();
addressbook.find(...);
```

### Resources:
- [Tizen API Reference](http://tizen.org/api/tizen)
- [Contact Read](http://tizen.org/api/contact.read)
Conclusions

• To developers:
  – You need to declare the required features in the manifest
    • The current SDK does not support automatic manifest configuration
    • Features need to be defined manually
  – Declare the minimum set of features you really need
    • Helps to better protect the device and user data
  – Pay attention to proper error handling in your application
    • Calls to device features may be denied by the Security system
    • Never assume a call will succeed
Thank You!

More Developer Information:
http://tizen.org
https://developer.tizen.org/documentation
## Appendix Tizen APIs

<table>
<thead>
<tr>
<th>API Group</th>
<th>Feature / Device Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tizen</td>
<td><a href="http://tizen.org/api/tizen">http://tizen.org/api/tizen</a></td>
</tr>
</tbody>
</table>
| Alarm     | [http://tizen.org/api/alarm](http://tizen.org/api/alarm)  
http://tizen.org/api/alarm.read  
http://tizen.org/api/alarm.write |
| Application | [http://tizen.org/api/application](http://tizen.org/api/application)  
http://tizen.org/api/application.read  
http://tizen.org/api/application.kill  
http://tizen.org/api/application.launch |
| Bluetooth | [http://tizen.org/api/bluetooth](http://tizen.org/api/bluetooth)  
http://tizen.org/api/bluetooth.spp  
http://tizen.org/api/bluetooth.gap |
| Calendar  | [http://tizen.org/api/calendar](http://tizen.org/api/calendar)  
http://tizen.org/api/calendar.write  
http://tizen.org/api/calendar.read |

<table>
<thead>
<tr>
<th>API Group</th>
<th>Feature / Device Capability</th>
</tr>
</thead>
</table>
| Call      | [http://tizen.org/api/call](http://tizen.org/api/call)  
http://tizen.org/api/call.simple  
http://tizen.org/api/call.history  
http://tizen.org/api/call.history.read  
http://tizen.org/api/call.history.write |
| Contact   | [http://tizen.org/api/contact](http://tizen.org/api/contact)  
http://tizen.org/api/contact.read  
http://tizen.org/api/contact.write |
| Filesystem | [http://tizen.org/api/filesystem](http://tizen.org/api/filesystem)  
http://tizen.org/api/filesystem.read  
http://tizen.org/api/filesystem.write |
| Geocoder  | [http://tizen.org/api/geocoder](http://tizen.org/api/geocoder) |
## Appendix Tizen APIs

<table>
<thead>
<tr>
<th>API Group</th>
<th>Feature / Device Capability</th>
</tr>
</thead>
</table>
| Media Content | [http://tizen.org/api/mediacontent](http://tizen.org/api/mediacontent)  
|             | [http://tizen.org/api/mediacontent.read](http://tizen.org/api/mediacontent.read) |
| Messaging    | [http://tizen.org/api/messaging](http://tizen.org/api/messaging)  
|             | [http://tizen.org/api/messaging.send](http://tizen.org/api/messaging.send)  
|             | [http://tizen.org/api/messaging.read](http://tizen.org/api/messaging.read)  
|             | [http://tizen.org/api/messaging.write](http://tizen.org/api/messaging.write)  |
| NFC          | [http://tizen.org/api/nfc](http://tizen.org/api/nfc)  
|             | [http://tizen.org/api/nfc.tag](http://tizen.org/api/nfc.tag)  
|             | [http://tizen.org/api/nfc.p2p](http://tizen.org/api/nfc.p2p)  
|             | [http://tizen.org/api/nfc.se](http://tizen.org/api/nfc.se)  |
| SystemInfo   | [http://tizen.org/api/systeminfo](http://tizen.org/api/systeminfo)  |
| Time         | [http://tizen.org/api/time](http://tizen.org/api/time)  
|             | [http://tizen.org/api/time.read](http://tizen.org/api/time.read)  
|             | [http://tizen.org/api/time.write](http://tizen.org/api/time.write)  |
| LBS          |                                                     |
| Map          | [http://tizen.org/api/lbs.map](http://tizen.org/api/lbs.map)  |
| POI          | [http://tizen.org/api/lbs.poi](http://tizen.org/api/lbs.poi)  |
| Route        | [http://tizen.org/api/lbs.route](http://tizen.org/api/lbs.route)  |

Developer Information:  
https://developer.tizen.org/documentation