

Tizen Web Device API

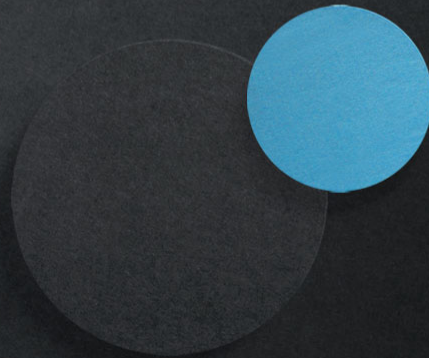
Kisub Song
Samsung Electronics

TIZEN™
**DEVELOPER
CONFERENCE**
2013
SAN FRANCISCO

Contents

- **Introduction**
 - Why Tizen Web Device APIs?
- **Tizen Web Device API**
 - Tizen Web Device API Background
 - 25 Modules of Tizen Web Device API
 - Updates from Tizen 1.0
- **Device APIs in Detail**
 - Application Control / Hybrid Application / Searching with Filter
- **Summary**
- **Appendix**

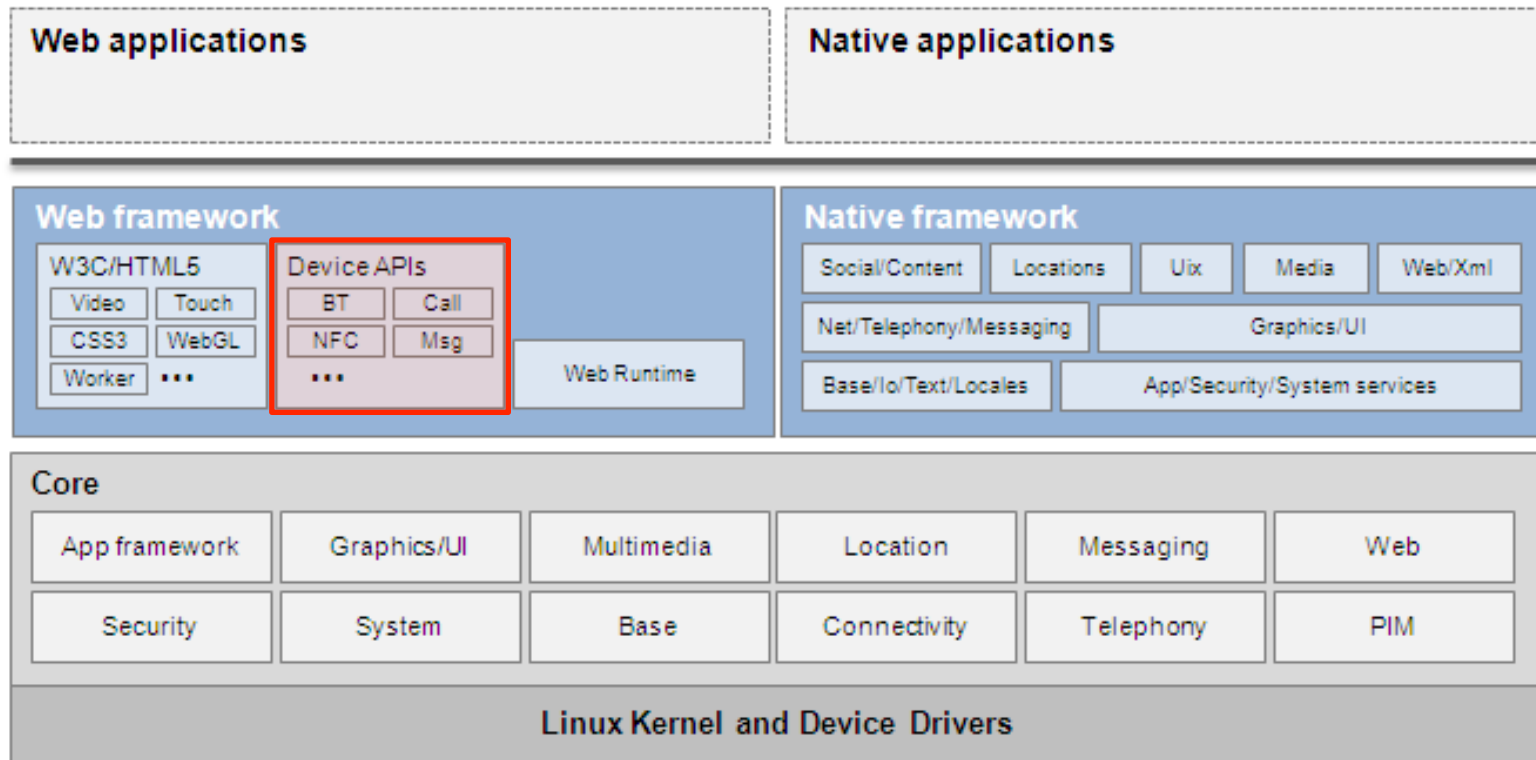
Introduction



Introduction

- Tizen is a **Web-based platform**
- A **Web application** is composed with Web technologies, such as HTML5, JavaScript, and CSS
- **Tizen Web device APIs provide the characteristic features of the Tizen platform**

Web Device APIs in Tizen Architecture



Tizen Web APIs

Tizen Web APIs

Device API

Application♪

Bluetooth♪

Calendar♪

Contact♪

Messaging♪

...♪

W3C

HTML5♪

CSS3♪

Web Worker♪

Geolocation♪

WebAudio♪

Touch Event♪

WebSocket♪

File♪

Widget♪

...♪

Miscellaneous

Web GL♪

Typed Array♪

Full Screen API♪

JSON♪

URI Scheme♪

...♪

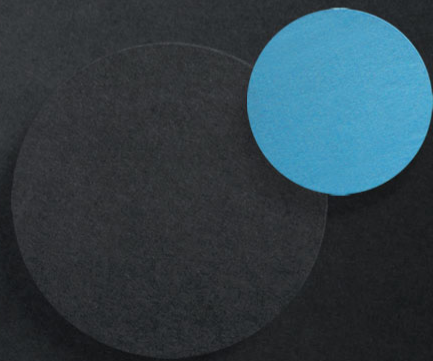
Why Tizen Web Device APIs?

- **Limitations of standard specifications**
 - Most of them are still working drafts
 - No full support for the Tizen characteristic features
- **Tizen Web device APIs**
 - Device APIs give the characteristic features of the Tizen platform
 - Device APIs have been updated agilely

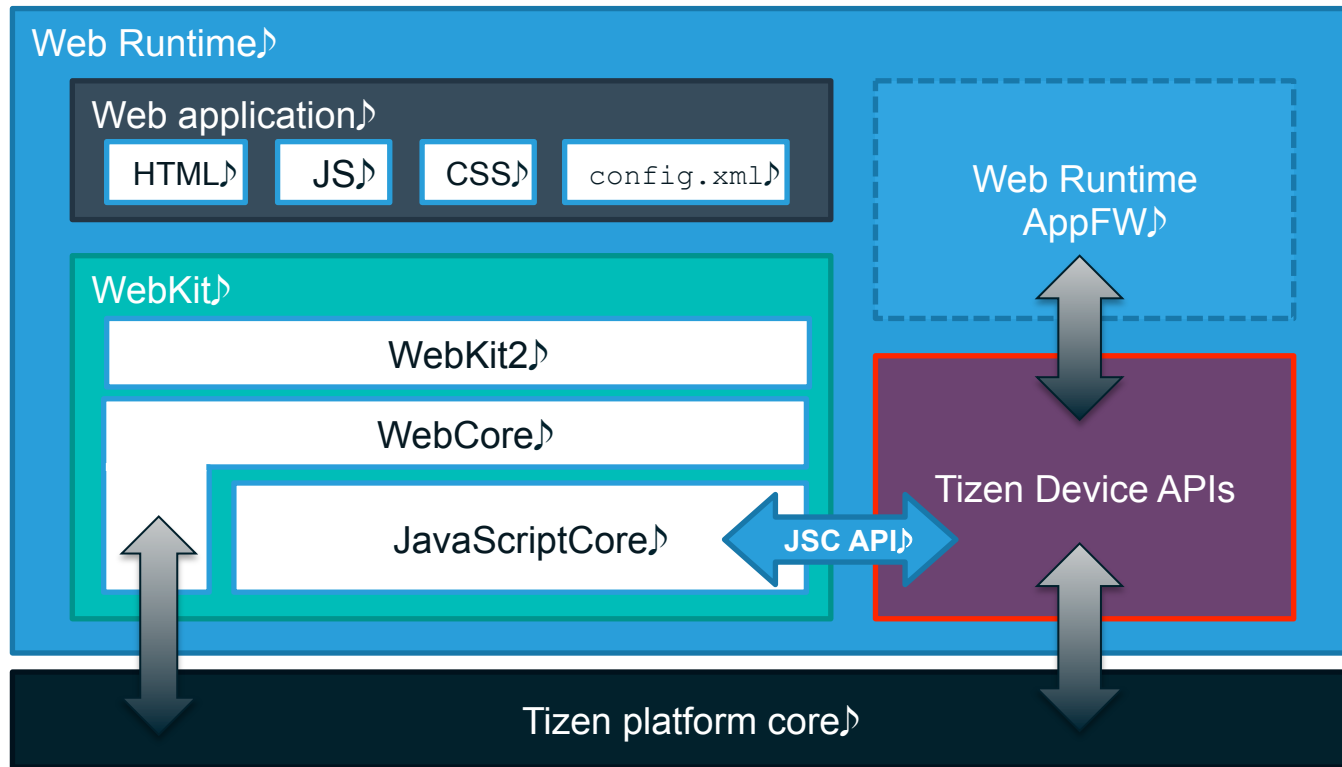
Tizen Web Device APIs Design

- **Follows W3C specification API style**
 - Numerical constants avoided but string enumerations used
 - Most methods are asynchronous
 - Success and error callback, and constructors used
- **Based on standard technologies**
 - Tizen Web device API specifications are written in **WebIDL**
 - Implementation follows the fundamental rule of **ECMAScript 5.1** and W3C **WebIDL** recommendation (for example, type conversion)
- **Defined `tizen` namespace**
 - All Tizen device APIs exist in the `tizen` namespace
 - `tizen` is the global object to which all Tizen device APIs are bound

Tizen Web Device API



Web Runtime



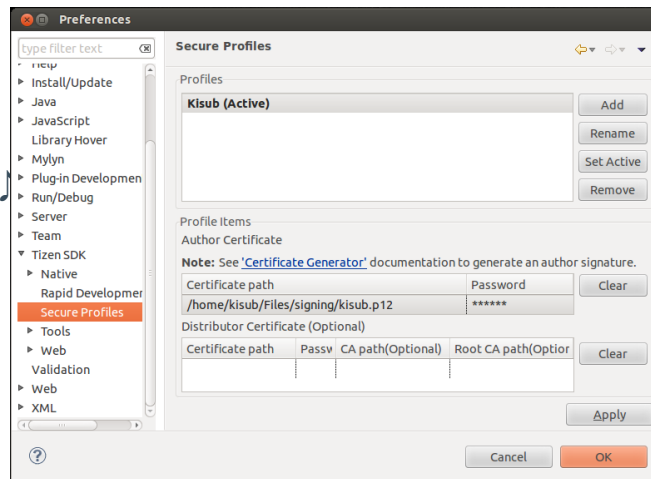
Configuration for Web Applications

- **config.xml file**
 - Defined in the W3C *Packaging and XML Configuration* specification
 - Tizen namespace is defined for describing Tizen specific features
- **Properties related with device APIs**

Element	Description	Example
tizen:application	Describes the application ID and package ID.	<pre><tizen:application id="UJedH64lXL.Test" package="UJedH64lXL" required_version="2.1" /></pre>
tizen:app-control	Defines the application control that this application provides.	<pre><tizen:app-control> <tizen:src name="controller.html" /> <tizen:operation name="http://tizen.org/appcontrol/operation/view" /> </tizen:app-control></pre>
tizen:privilege	Describes the privilege that this application needs.	<pre><tizen:privilege name="http://tizen.org/privilege/appmanager.kill" /></pre>

Application Signing

- **Applications must be signed with 2 signatures:**
 - **Author signature**
 - Determines the integrity of an application package as intended by the developer
 - Confirms that all applications that are signed with the same author certificate are trustworthy
 - **Distributor signature**
 - Generated by an application publisher
 - Determines the privilege level of the application.



Privilege Levels

- **Privileges are categorized into 3 levels**
 - `public`: open to all Tizen application developers
 - `partner`: for partners registered on the Tizen store
 - `platform`: for managing the Tizen platform

Application distributor certification \ API privilege level	Non-privileged	public	partner	platform
<code>public</code>	O	O	X	X
<code>partner</code>	O	O	O	X
<code>platform</code>	O	O	O	O

Privileges

- **Some APIs require privileges for using them from an application**
 - Privilege level: application needs to have higher privilege level than the required one to use the API
 - Privilege: needs to be described in `config.xml`

launch

Launches an application with the given application ID.

```
void launch(ApplicationId id, optional SuccessCallback? successCallback,  
optional errorCallback? errorCallback);
```

Since: 2.0

The `ErrorCallback()` is launched with these error types:

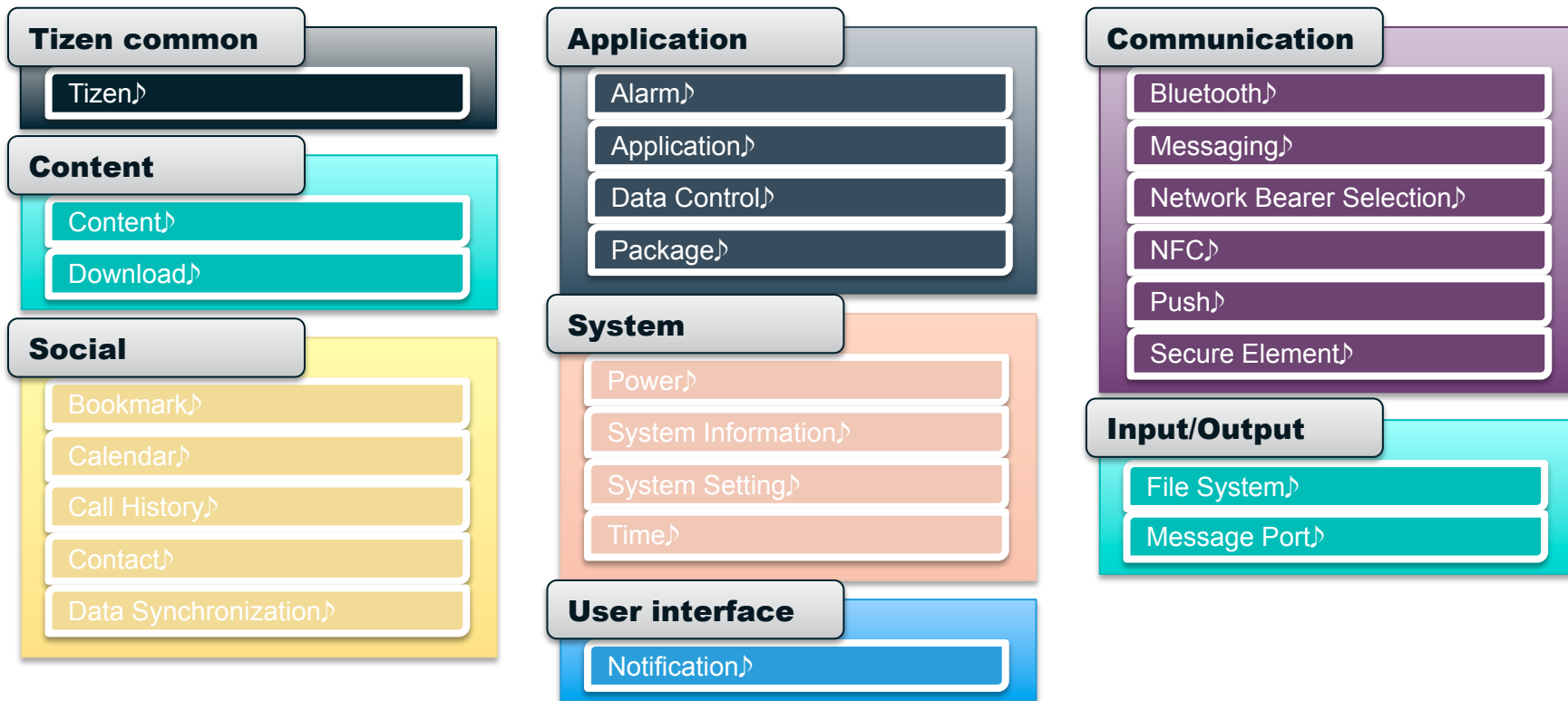
- `NotFoundError` - If the application is not found with given ID.
- `InvalidValuesError` - If any of the input parameters contain an invalid value.
- `UnknownError` - If any other error occurs.

Privilege level: public

Privilege: <http://tizen.org/privilege/application.launch>

Required privileges are described in API specification documents

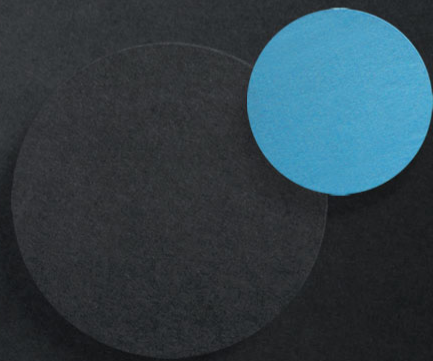
Tizen Web Device API Modules



Updates from Tizen 1.0

	Modules		
Newly introduced	Download	Bookmark	Data Synchronization
	Data Control	Package	Power
	System Setting	Notification	Network Bearer Selection
	Push	Secure Element	Message Port
Obsolete	Geocoder	LBS	
Renamed	Call	→	Call History
	Media Content	→	Content

Device APIs in Detail

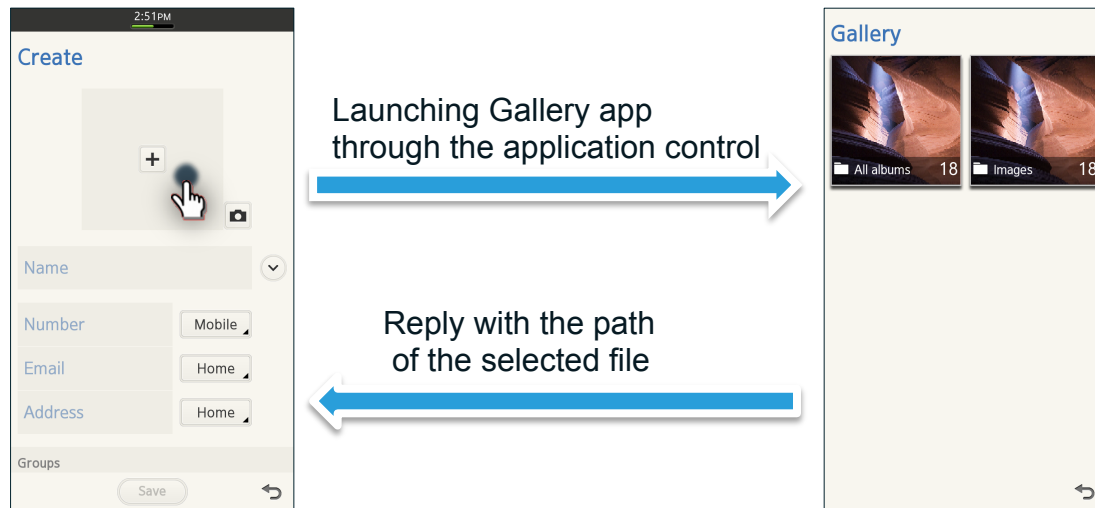




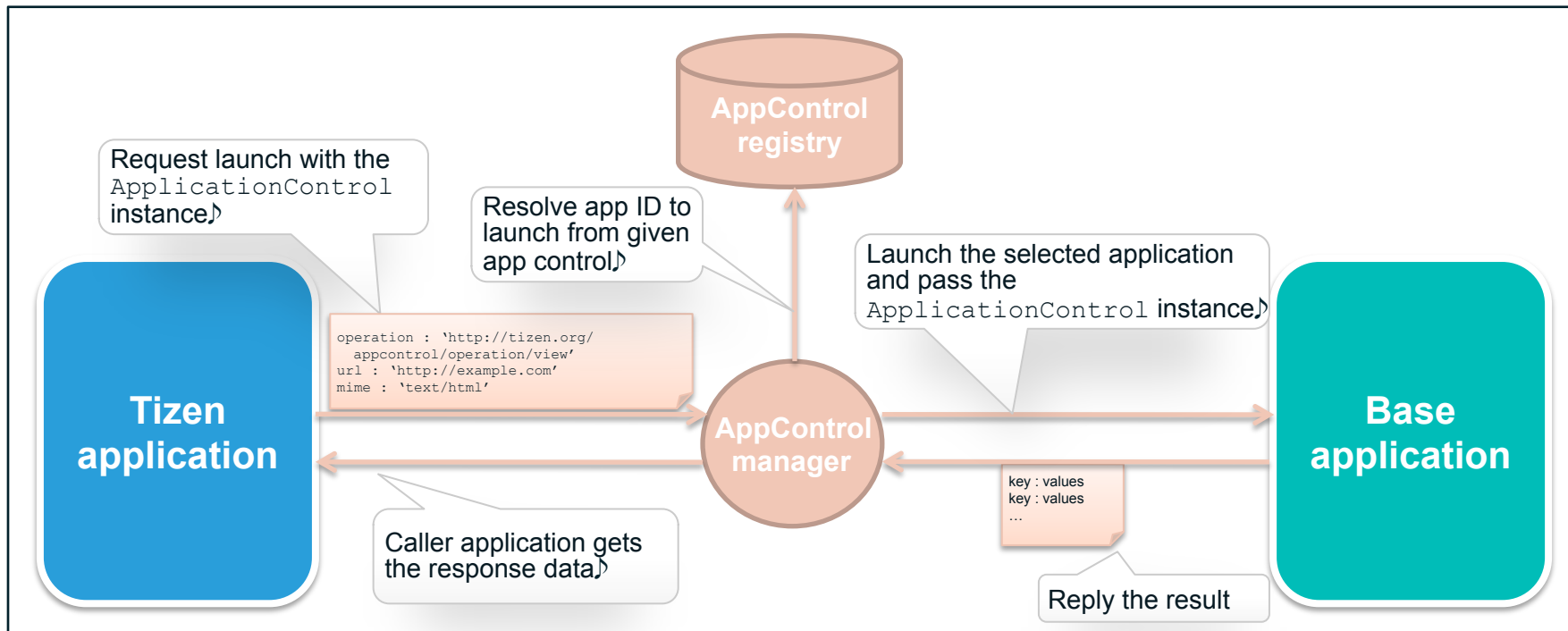
Application Control (AppControl)

Application Control

- **Application control enables launching other applications based on the functionalities needed**
 - For example: Contacts app uses the Gallery app to select an image file



Application Control



Example: Requesting Application

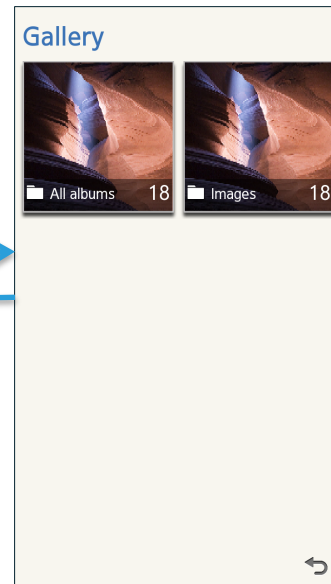
- Requesting AppControl:

```
var appControl = new tizen.ApplicationControl(  
    'http://tizen.org/appcontrol/operation/pick', null, 'image/*', null);  
  
tizen.application.launchAppControl(appControl, null,  
    function() { console.log("Launching AppControl succeeded"); },  
    function(e) { console.log("Launching AppControl failed"); },  
    {  
        onsuccess : function(data) {  
            console.log("AppControl returned success");  
            data.forEach(function(v, i) {  
                console.log("[ "+i+" ] key : "+v.key);  
                v.value.forEach(function(vv, vi) {  
                    console.log("---- value#"+vi+" : "+vv);  
                });  
            });  
        },  
        onfailure : function() { console.log("AppControl returned failure"); }  
    }  
);
```

onsuccess function is invoked when the base application returns the result data

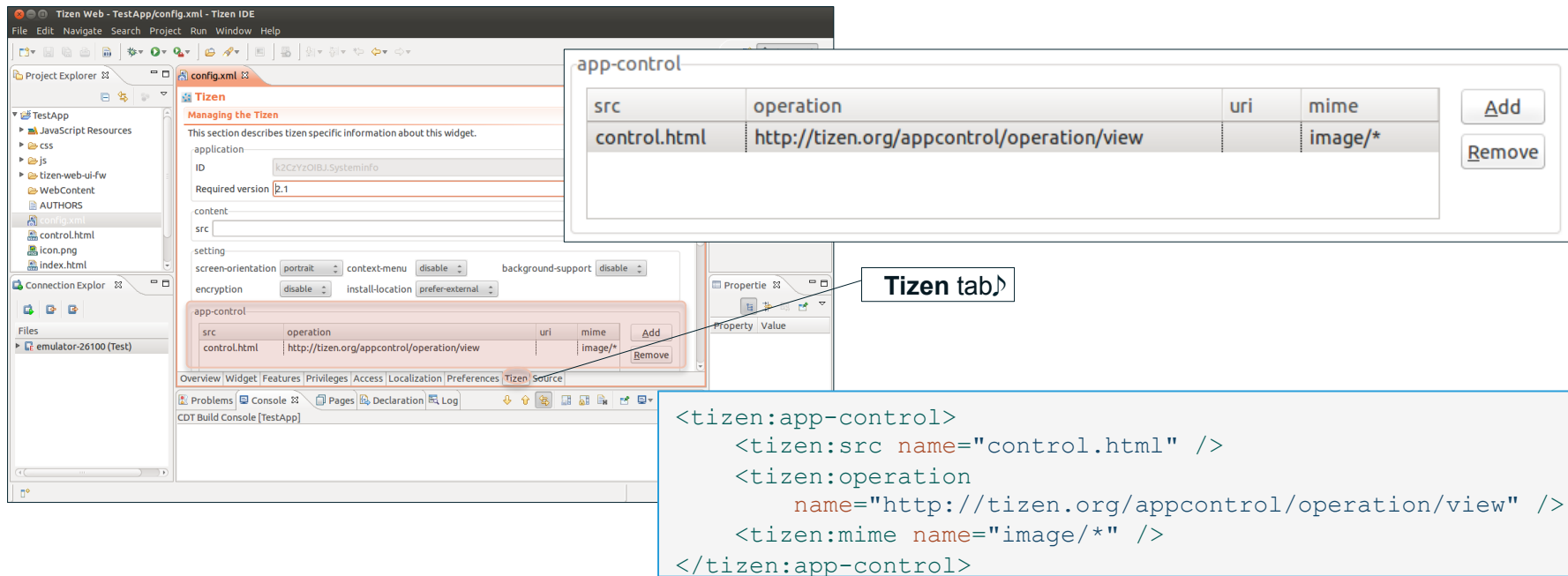
Printed log

```
... ConsoleMessage ... :Launching AppControl succeed  
... ConsoleMessage ... :AppControl returned success  
... ConsoleMessage ... :[0] key : http://tizen.org/appcontrol/data/selected  
... ConsoleMessage ... :---- value#0 : /opt/usr/media/Images/image2.jpg
```



Providing AppControl in Your Application (1/2)

- Edit AppControl in the IDE:



The screenshot shows the Tizen IDE interface. The 'Project Explorer' on the left lists files like 'config.xml', 'control.html', and 'icon.png'. The 'Tizen' tab is active, displaying the 'Managing the Tizen' section. A callout box highlights the 'app-control' configuration area, which contains a table with the following data:

src	operation	uri	mime
control.html	http://tizen.org/appcontrol/operation/view		image/*

Buttons for 'Add' and 'Remove' are visible next to the table. A callout box labeled 'Tizen tab' points to the 'Tizen' tab in the IDE's tab bar. Below the IDE, a code block shows the XML representation of the app-control configuration:

```
<tizen:app-control>
  <tizen:src name="control.html" />
  <tizen:operation
    name="http://tizen.org/appcontrol/operation/view" />
  <tizen:mime name="image/*" />
</tizen:app-control>
```

Providing AppControl in Your Application (2/2)

- Handle passed **ApplicationControl** instance and reply the result

RequestedApplicationControl instance has the received **AppControl**, the caller's **applID** and functions for reply

ApplicationControl instance has the same value with the **AppControl** instance sent from caller application

Replying the result, if any

```
function initAppControl()
{
    var currentApp = tizen.application.getCurrentApplication();

    var reqedAppControl = currentApp.getRequestedAppControl();

    var appControl = reqedAppControl.appControl;

    console.log("operation:" + appControl.operation +
        " / uri:" + appControl.uri + " / mime:" + appControl.mime);

    // Do something

    reqedAppControl.replyResult( [
        tizen.ApplicationControlData('key0', ['value0_0']),
        tizen.ApplicationControlData('key1', ['value1_0'])
    ] );
}

window.addEventListener('load', initAppControl, false);
```



Hybrid Application with Message Port

Message Port



- **Features**

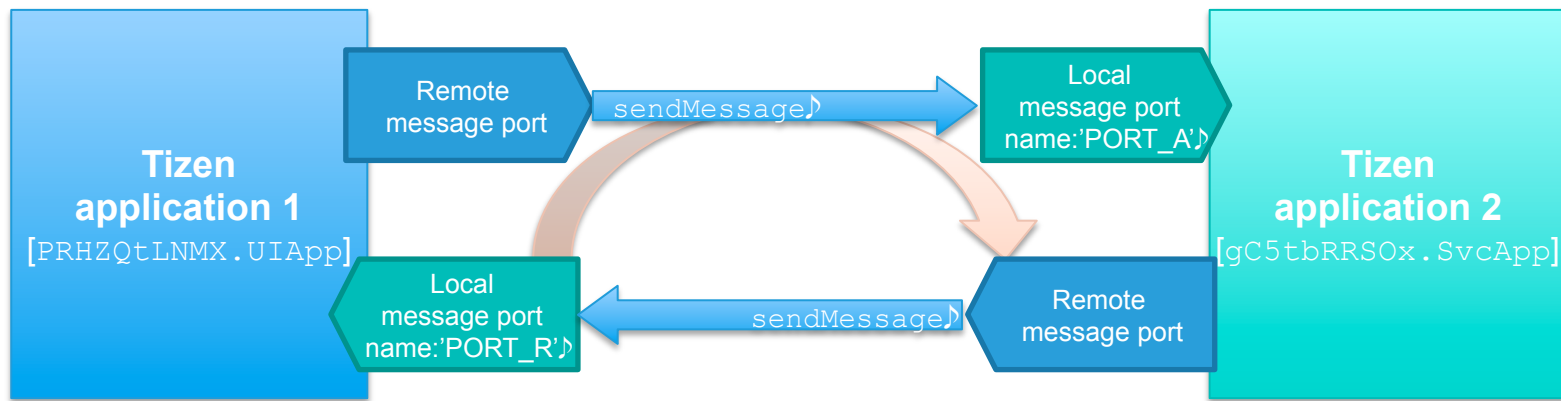
- Application identifies message ports by application ID and port name
- Application can open a message port that restricts access from unidentified application
- Native API also has a message port that is compatible with the Web message port

- **Limitations**

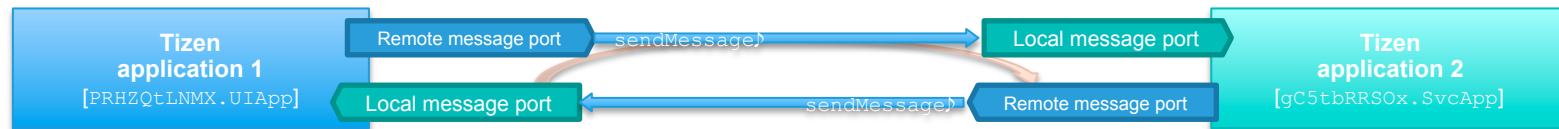
- Only string data can be sent through a message port
- It can send a limited amount of data at a time (4KB)

Communication through Message Port

- `LocalMessagePort` is used to receiving messages.
- `RemoteMessagePort` is used to send messages.
- Application can send messages using the `sendMessage()` method of `RemoteMessagePort` object.



Example: Communication through Message Port



```
var remoteMsgPort =
  tizen.messageport.requestRemoteMessagePort(
    'gC5tbRRSOx.SvcApp', 'PORT_A');
var localMsgPort =
  tizen.messageport.requestLocalMessagePort(
    'PORT_R');

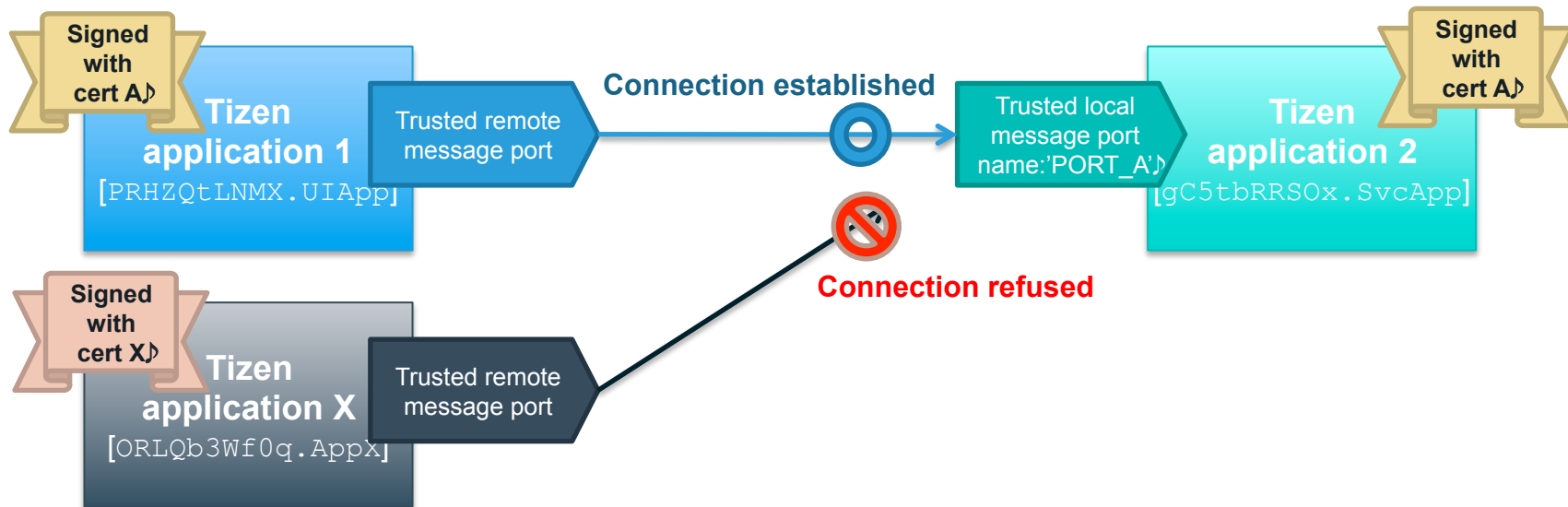
remoteMsgPort.sendMessage(
  [
    { key:'CMD', value:'openWindow' },
    { key:'OPTION', value:'bx' }
  ],
  localMsgPort
);
```

```
var localMsgPort =
  tizen.messageport.requestLocalMessagePort(
    'PORT_A');

var watchId = localMsgPort.addMessagePortListener(
  function (data, remoteMsgPort) {
    console.log('Received data is...');
    // .....
    remoteMsgPort.sendMessage( [
      { key:'ANSWER', value:'OK' },
      { key:'PARAM1', value:'WFWQ' }
    ] );
  }
);
```

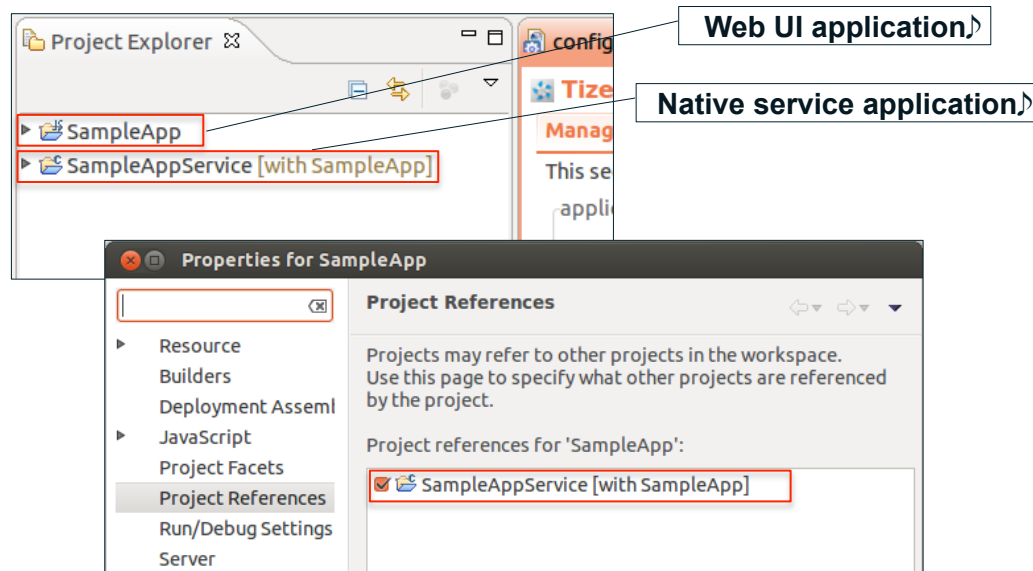
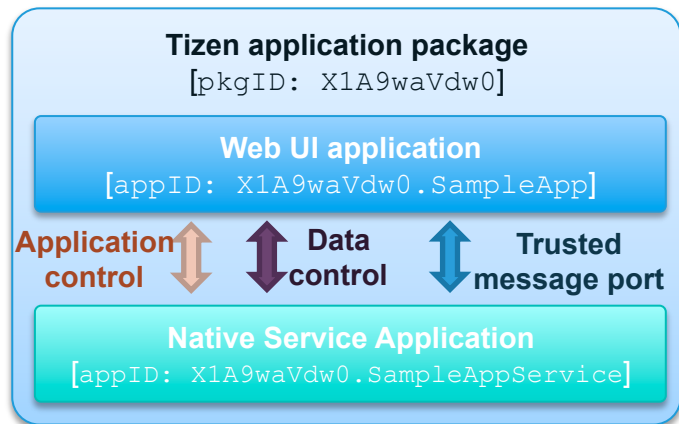
Trusted Message Port

- Trusted message port allows access from the application signed with the **same certificate** with the current application



Hybrid Application Package

- Combines a web application and native applications in a package
- Create in the SDK IDE: Project > Properties > Project References





Searching with Filters

Filters

- **Filters are Tizen common interfaces to make complex queries**
- **Filter types:**
 - Attribute filter, attribute range filter, and composite filter
- **Filters are used with various Tizen device API modules**

Module	Function [Interface::Method()]	Search for (result type interface)
Contact	ContactManager::find()	Person
	AddressBook::find()	Contact
Calendar	Calendar::find()	CalendarItem (CalendarTask or CalendarEvent)
CallHistory	CallHistory::find()	CallHistoryEntry
Content	ContentManager::find()	Content (VideoContent or AudioContent or ImageContent)
Messaging	MessageStorage::findMessage()	Message
	MessageStorage::findConversations()	MessageConversation
	MessageStorage::findFolders()	MessageFolder

Attribute Names

- **Name of the attribute to be searched**
 - Generally, matches with the name of the attribute.
 - For example: to search the Contact objects `birthday` attribute, use the attribute name `birthday`
 - The attribute name can have 2 or more levels separated by a period
 - For example: `phoneNumbers.number`
 - Available attribute names are described on the SDK documents.

Contact Filter Attributes

The following table lists the filter types you can use with specific contact attributes in the methods of the [AddressBook](#) interface.

Table: Contact filter attributes

Attribute	Attribute filter supported	Attribute range filter supported
id	Yes	Yes
personId	Yes	Yes
lastUpdated	Yes	Yes
isFavorite	Yes	Yes

Tizen Web App Programming >
Programming Guide > Device Guides
> Tizen Guide

Filter Types

- **Attribute filter**

```
var filter = new tizen.AttributeFilter("name.firstName", "EXACTLY", "Chris");
```

Describe the filter with the SQL where clause:

```
WHERE name.firstName = 'Chris'
```

Flag indicating search mode

Attribute where the value is searched from

- **Attribute range filter**

```
var filter = new tizen.AttributeRangeFilter("name.firstName", null,  
new Date(1983, 0, 1));
```

Describe the filter with the SQL where clause:

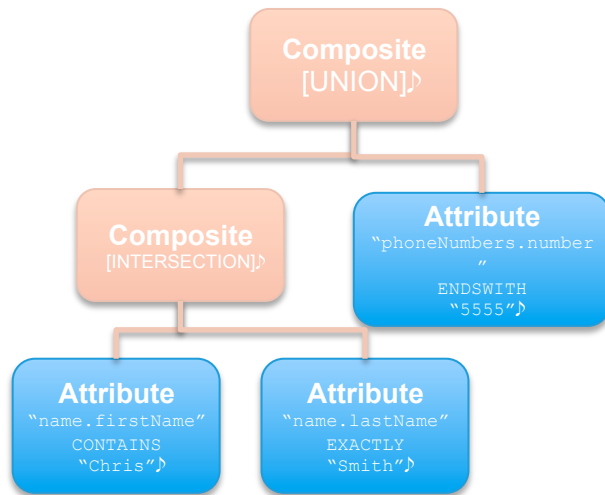
```
WHERE birthday < 410227200
```

- **Composite filter**

- Combines several filters into a set

Example: Filters in Pseudo-SQL Expression

```
var filter = new tizen.CompositeFilter({  
  type:"UNION",  
  filters:[  
    new tizen.CompositeFilter({  
      type:"INTERSECTION",  
      filters:[  
        new tizen.AttributeFilter("name.firstName", "CONTAINS", "Chris"),  
        new tizen.AttributeFilter("name.lastName", "EXACTLY", "Smith")  
      ]  
    },  
    new tizen.AttributeFilter("phoneNumbers.number", "ENDSWITH", "5555")  
  ]  
});  
  
addressBook.find(filter, null,  
  function(contacts){ console.log("Retrieved : " + contacts.length); },  
  function(err){ console.log("Error : " + err.message); } );
```



Describe the filter with the pseudo-SQL where clause:

```
( ( ( name.firstName LIKE '%Chris%' ) AND ( name.lastName = 'Smith' ) )  
  OR  
  ( phoneNumbers.number LIKE '5555%' ) )
```

Summary

- **Tizen Web Device API supports Tizen characteristic features.**
 - Tightly coupled with the structure of the Web Runtime
 - Privileges have been defined to control API access authority
- **Tizen Web Device API provides 25 modules**
 - 12 have been newly introduced since Tizen 1.0
- **Tizen Web Device API provides various special features**



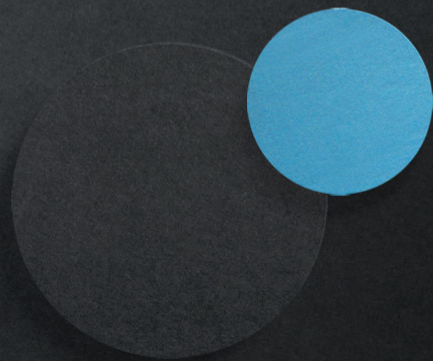
TIZENTM

**DEVELOPER
CONFERENCE**

2013

SAN FRANCISCO

Appendix



APIs with partner or public privileges

Privilege	Module	Methods
Partner	Application	kill / getAppCert
	Data Control	ALL
	Network Bearer Selection	ALL
	Secure Element	ALL
	System Information	imei, msisdn, msin
Platform	Package	install / uninstall
	Bluetooth	setVisible
	Bookmark	ALL

Application Control Model

	Description	Example	
Operation	<ul style="list-style-type: none">• Defines the action to be performed by the application control• Mandatory item for application control• IRI style	<code>http://tizen.org/appcontrol/operation/view</code> <code>http://tizen.org/appcontrol/operation/pick</code> <code>http://tizen.org/appcontrol/operation/call</code>	Used for resolution
URI	<ul style="list-style-type: none">• Data on which the action is performed	<code>http, tel, mailto ...</code>	
MIME	<ul style="list-style-type: none">• Specific URI type• If MIME is not set and URI has a <i>file</i> scheme, the MIME is automatically determined by local file	<code>audio/*, video/* ...</code>	
Application control data	<ul style="list-style-type: none">• Key-value pairs providing additional information for the service request• Dataset		

* AppControl Manager resolves the application whose operation, URI, and MIME type match the requested `ApplicationControl` instance exactly

Base AppControls

Base application	Operation	URI	MIME
Internet (Browser)	http://tizen.org/appcontrol/operation/view	http	-
MusicPlayer	http://tizen.org/appcontrol/operation/view	-	audio/*
VideoPlayer	http://tizen.org/appcontrol/operation/view	-	video/*
Phone	http://tizen.org/appcontrol/operation/dial	tel	-
Call	http://tizen.org/appcontrol/operation/call	tel	-
Camera	http://tizen.org/appcontrol/operation/create_content	-	image/jpeg, video/3gpp
FileManager	http://tizen.org/appcontrol/operation/pick	-	*/*, image/*, audio/*, video/*
Email	http://tizen.org/appcontrol/operation/compose	mailto	-
ImageViewer	http://tizen.org/appcontrol/operation/view	-	image/*
...			

* The base AppControls are provided by the system default applications

Example: Requesting Application

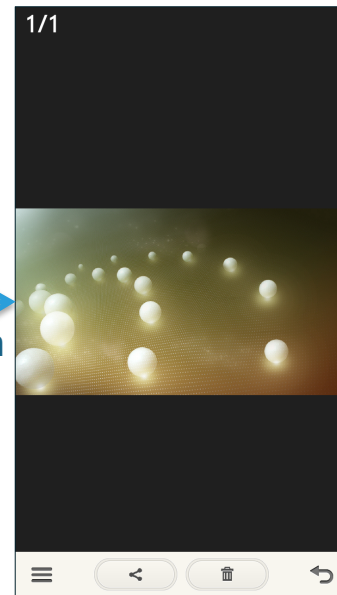
```
var appControl = new tizen.ApplicationControl(  
    "http://tizen.org/appcontrol/operation/view",  
    "file:///opt/usr/media/Images/image1.jpg",  
    null,  
    null);  
  
tizen.application.launchAppControl(appControl,  
    null,  
    function() {  
        console.log("launch appControl succeed");  
    },  
    function(e) {  
        console.log("launching appControl failed : " +  
            e.message);  
    }  
);
```

Operation▶

URI▶

MIME is null, but would be determined to be image/jpeg based on the URI value▶

Launch



ImageViewer

Providing AppControl in Your Application

- 3rd party application can provide a new AppControl by describing a new AppControl in the `config.xml` file

```
<tizen:app-control>
  <tizen:src name="[PAGE]" />
  <tizen:operation name="[OPERATION]" />
  <tizen:uri name="[URI_SCHEME]" />
  <tizen:mime name="[MIME_TYPE]" />
</tizen:app-control>
```

The **path of HTML file** launched initially when this AppControl is requested.

Attribute Filter - Flags

	string only	case-sensitive	SQL 'WHERE' expression	Description
EXACTLY	X	O	<code>ATTR = 'VALUE'</code>	Match exactly with the specified value
FULLSTRING	O	X	<code>ATTR LIKE 'VALUE'</code>	Match with the whole string but case-insensitive
CONTAINS	O	X	<code>ATTR LIKE '%VALUE%'</code>	Contain the specified string and case-insensitive
STARTSWITH	O	X	<code>ATTR LIKE 'VALUE%'</code>	Start with the specified string and case-insensitive
ENDSWITH	O	X	<code>ATTR LIKE '%VALUE'</code>	End with the specified string and case-insensitive
EXISTS*	-	-	<code>IS NOT NULL</code> or <code>ATTR <> ''</code>	Have any value not null nor empty string

* If the EXISTS flag is set, the `matchValue` is not necessary and can be ignored



Modules

Tizen APIs (1/2)

- **Tizen**
 - Provides Tizen's common functionalities
 - Contains object interfaces that are commonly used throughout the other modules
 - The `tizen` object is the topmost object providing the foundations for accessing Tizen device features

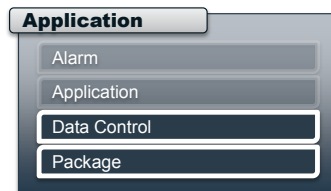
Tizen APIs (2/2)



- **Tizen's common object interfaces**
 - Generic asynchronous callback handling
 - Success callback for methods that do not require a return value
 - Error callback for methods that require an error as input parameter
 - Generic error and exception handling
 - Tizen APIs throw a `WebAPIException` object and returns a `WebAPIError` object through error callbacks
 - These follow the style and types of error of DOM4 specification.♪
 - Filters and sort modes
 - Filters are used to limit query results and compose complex queries
 - Generally, filters are used with the `find()` method in Calendar, Call History, Contact, Content, and Messaging modules♪

Application APIs (1/2)

- **Alarm**
 - Schedules an application to be launched at a specific time
- **Application**
 - Manages current application (this application)
 - Manages other applications
 - Retrieves information of applications
 - Launches other applications
 - Application control

Application APIs (2/2)



- **Package** 
 - Retrieves information of installed packages
 - Checks the updates of the installed package list
 - Installs or uninstalls packages
 - `partner` privileged
- **Data Control** 
 - Exchanges specific data with the data control provider application
 - Data control provider application can be implemented as a native application
 - Data control types:
 - `SqlDataControl`: SQL-type data control – data consists of rows and columns
 - `MapDataControl`: key-value-type data control
 - `partner` privileged

Package is a piece of applications that the system can install and uninstall, such as wgt, tpk, or rpm

Communication APIs (1/2)

- **Bluetooth**

- Manages Bluetooth devices
- Discovers nearby devices, and bonds or pairs with found devices
- Connects to devices to exchange data with them




- **NFC**

- Manages NFC devices
- Detects NFC tag and peer
- Exchanges NDEF data

- **Messaging**

- Sends or receives SMS, MMS, or email messages
- Retrieves the message storage

Communication APIs (2/2)

- **Network Bearer Selection** 
 - Sets a network bearer for a specific IP address or domain name
 - `partner` privileged
- **Secure Element** 
 - Provides functionality to communicate with applications in several secure elements, such as UICC/SIM, embedded Secure Element, or Secure SD card
 - `partner` privileged
- **Push** 
 - Receives push notifications from the Tizen push server

Content APIs (1/2)

- **Content**

- Discovers and manages images, video, music, and other files
- Scans content or directory metadata in the device and updates the content database
- Retrieves content from content database using filters
- Browses content by getting a list of content directories
- Views and edits content item details

Content APIs (2/2)

- **Download**



- Downloads files from a specific URL
- Enables applications to manage download operation details
 - Sets the stored location of a downloaded file
 - Sets the network type: cellular, Wi-Fi, or default
 - Get operation status: current received size and events
 - Gets the file MIME type

Input/Output APIs

- **Filesystem**

- Accesses the file system virtual root locations
 - Documents, downloads, images, music, videos, ringtones, wgt-package, wgt-private, and wgt-private-tmp
- Manages file storage
 - Internal and external
 - Mounted, removed, or unmountable
- Accesses files and directories
- Creates, reads, edits or deletes files and directories

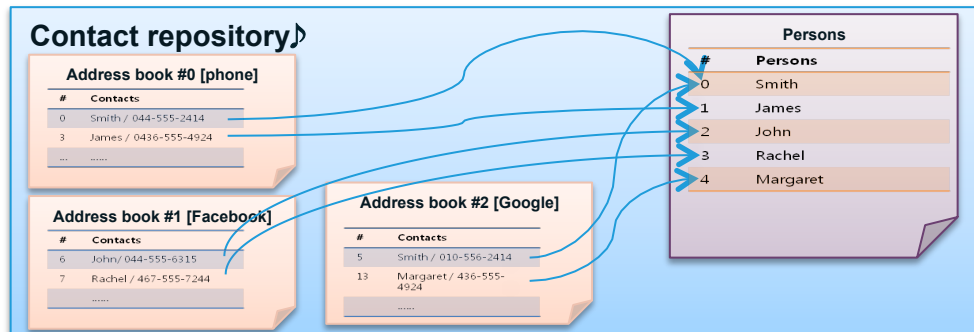
- **Message Port** 

- Communicates with other applications: IPC based on application ID

Social APIs (1/3)

- **Contact**

- Manages contacts in device address books
 - Including adding, searching, updating, and deleting
- Manages persons
 - Person is the aggregation of one or more contacts that are the information of the same person
- Supports vCard 3.0



Social APIs (2/3)

- **Calendar**

- Manages **events** and **tasks**
 - Including adding, searching, updating, and deleting
 - Each event or task has a series of attributes, such as purpose, starting time, and duration
- Monitors the changes in **events** and **tasks**
- Supports iCalendar v2.0, based on RFC 5545

- **Call History**

- Browses the call history of a device
- Removes call history entries
- Monitors changes

Social APIs (3/3)

- **Bookmark**



- Manages bookmarks and bookmark folders
- Browses bookmark folders
- `platform` privileged

- **Data Synchronization**



- Synchronizes device data to the server using the OMA DS 1.2 protocol
 - Contact data or calendar data
- Manages the OMS DS profile slots
 - Tizen platform sets a limitation on the number of supported profiles
 - Adds, updates and removes profiles

System APIs (1/2)



- **Time**

- Provides TZDate type that is an extended type of ECMAScript date
 - It contains the time zone information as well as time information
- Provides utility functions for managing system time and duration
 - Methods for getting local time or time zone
 - Methods for calculating time duration

- **System Information**

- Enables access various properties of the system
 - Bluetooth, NFC, Wi-Fi, Front/back Camera, Flash, GPS, Sensors, Platform and API version, USB host/accessory
- Monitors the change of system information

System APIs (2/2)

- **System Setting** 
 - Sets or gets the system setting values
 - `HOME_SCREEN`: Home screen background image
 - `LOCK_SCREEN`: Lock screen background image
 - `INCOMING_CALL`: incoming call ringtone
 - `NOTIFICATION_EMAIL`: email notification alert tone
- **Power** 
 - Manages the power state for the screen resource
 - Sets or gets the state of the screen: off, dim, normal, or bright
 - Monitors the changes in the state of the screen
 - Sets the CPU not to sleep

User Interface APIs

- **Notification**
 - Posts UI notifications about application events

