Overview of the Tizen Native Application Framework

Byung Woan Kim

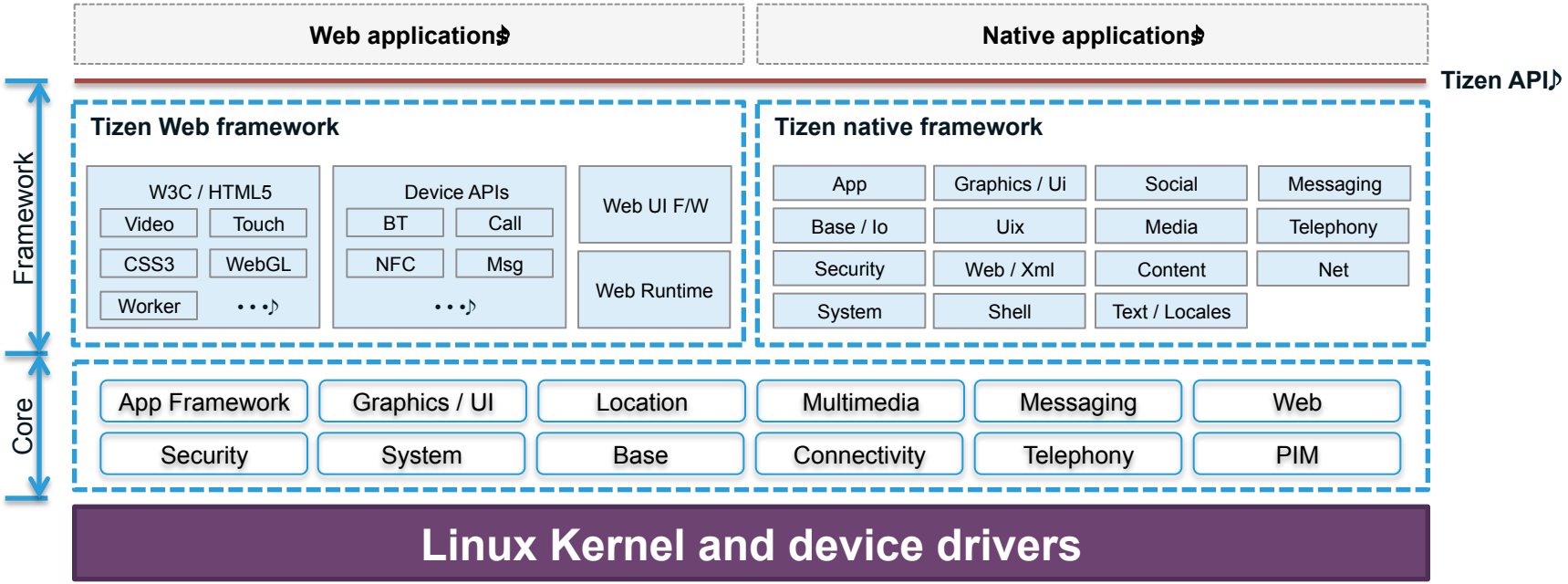
TIZEN™
DEVELOPER CONFERENCE
2013
SAN FRANCISCO

Contents

Introduction

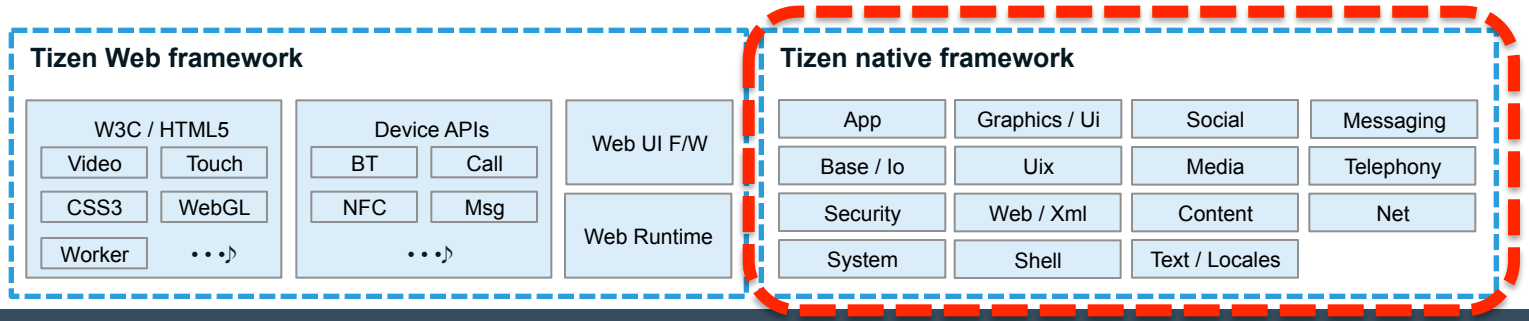
Introduction

- Tizen has 2 frameworks : Web and native



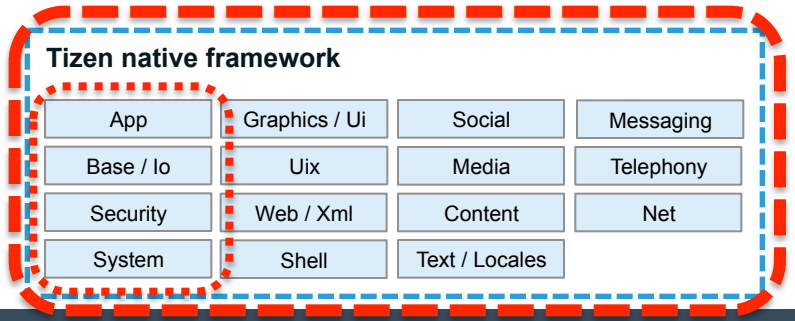
Introduction

- **Tizen native framework (released with Tizen 2.0):**
  - Supports full-featured native app development and provides a variety of features, such as background service, IP push, and TTS
  - Provides a chance to reuse existing C/C++ engines and libraries in writing apps
  - Includes popular standard open source libraries:
    - glibc, libstdc++, libxml2, Open GL® ES, Open AL®, Open MP®♪



Introduction

- **Tizen native application framework provides:**
  - Package installation, uninstallation, and upgrading
  - Application model of the native framework
  - Application life-cycle management
  - System event handlers for applications
  - Application configuration management
  - And more…



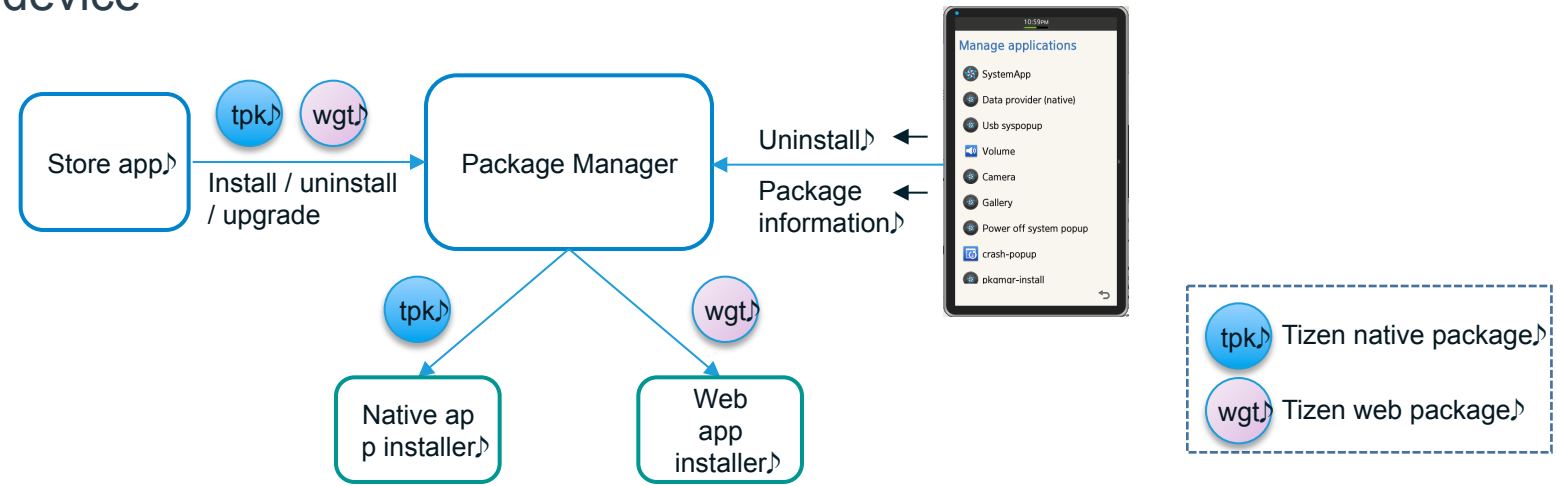
Application (Basic)

Application | Package

- **Package**
  - Container of executable binaries and content
  - Package has a `PackageId` , a global unique ID of the package
- **App**
  - Base class of an executable entity as a process
  - An App has:
    - `AppId`, global unique ID of the App
    - `AppName`, locale-independent name of the App
    - `AppVersion` (shared among all Apps in a package)
    - `AppResource` (shared among all Apps in a package)
    - Privilege (shared among all Apps in a package)

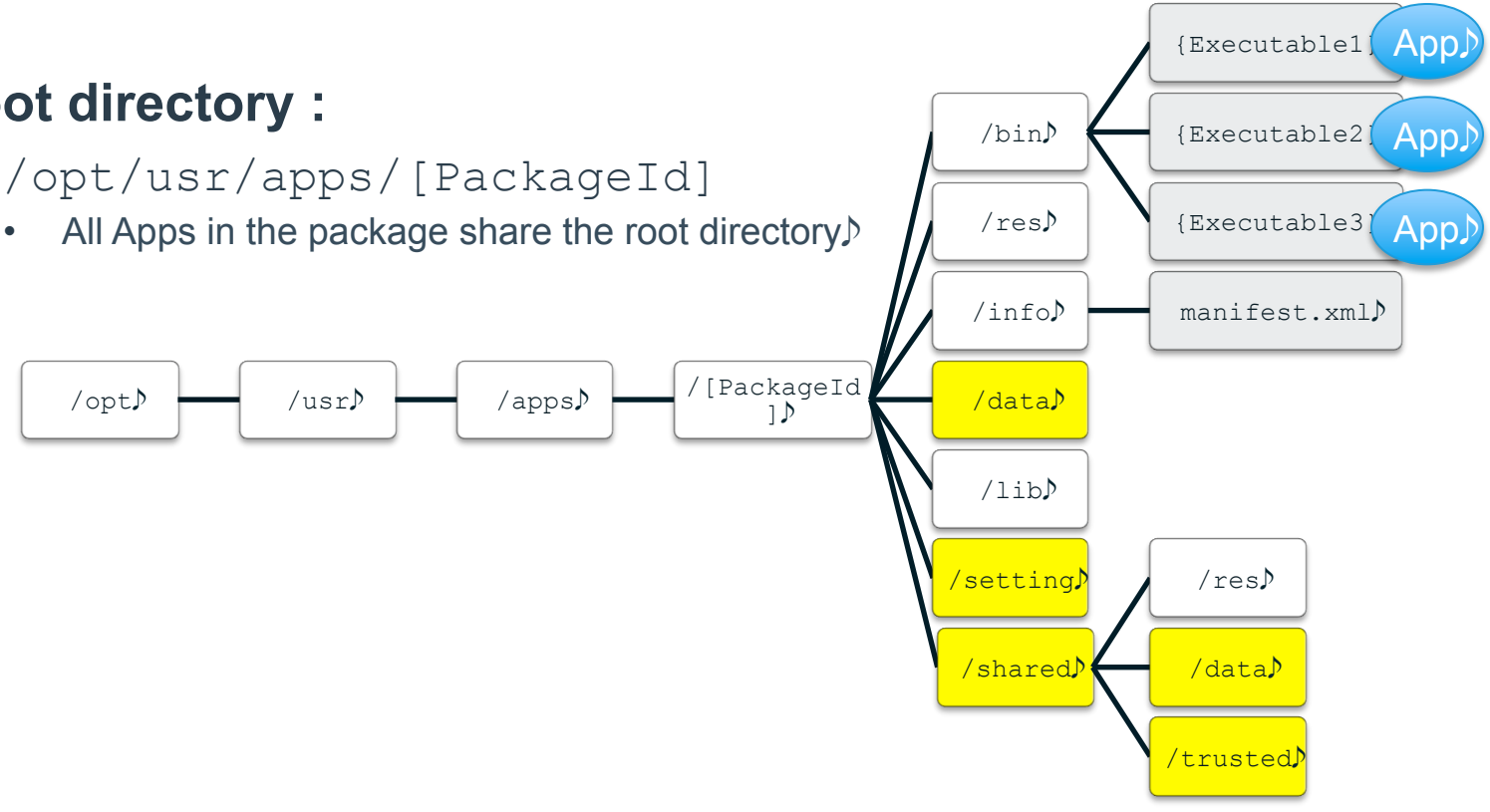
Application | Package

- **Tizen::App::Package::PackageManager**
  - Install, uninstall, and upgrade packages
  - Retrieve information related to the packages that are installed on the device



Application | Package

- **Root directory :**
  - `/opt/usr/apps/[PackageId]`
    - All Apps in the package share the root directory♪



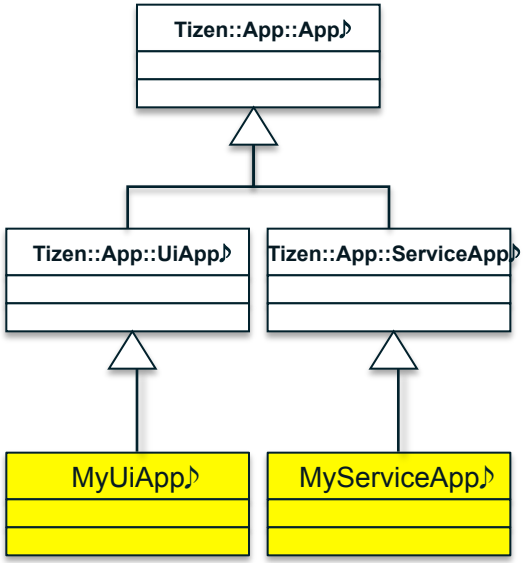
Application | Package

- **Upgrade policy**
  - All read-only files are removed
  - All the files in writable directories are overwritten

Directory♪	Upgrade policy♪
[App Root]/bin [App Root]/res [App Root]/shared/res [App Root]/info [App Root]/lib [App Root]/shared/res	These files (binaries, resources, icons, and information fil es) are read-only. The files are deleted and replaced with the upgraded files.♪
[App Root]/data [App Root]/shared/data [App Root]/shared/trusted [App Root]/setting♪	These files are writable files. All files are not removed. The existing file is overwritten by a new file if they have th e same name.♪

Application | Type

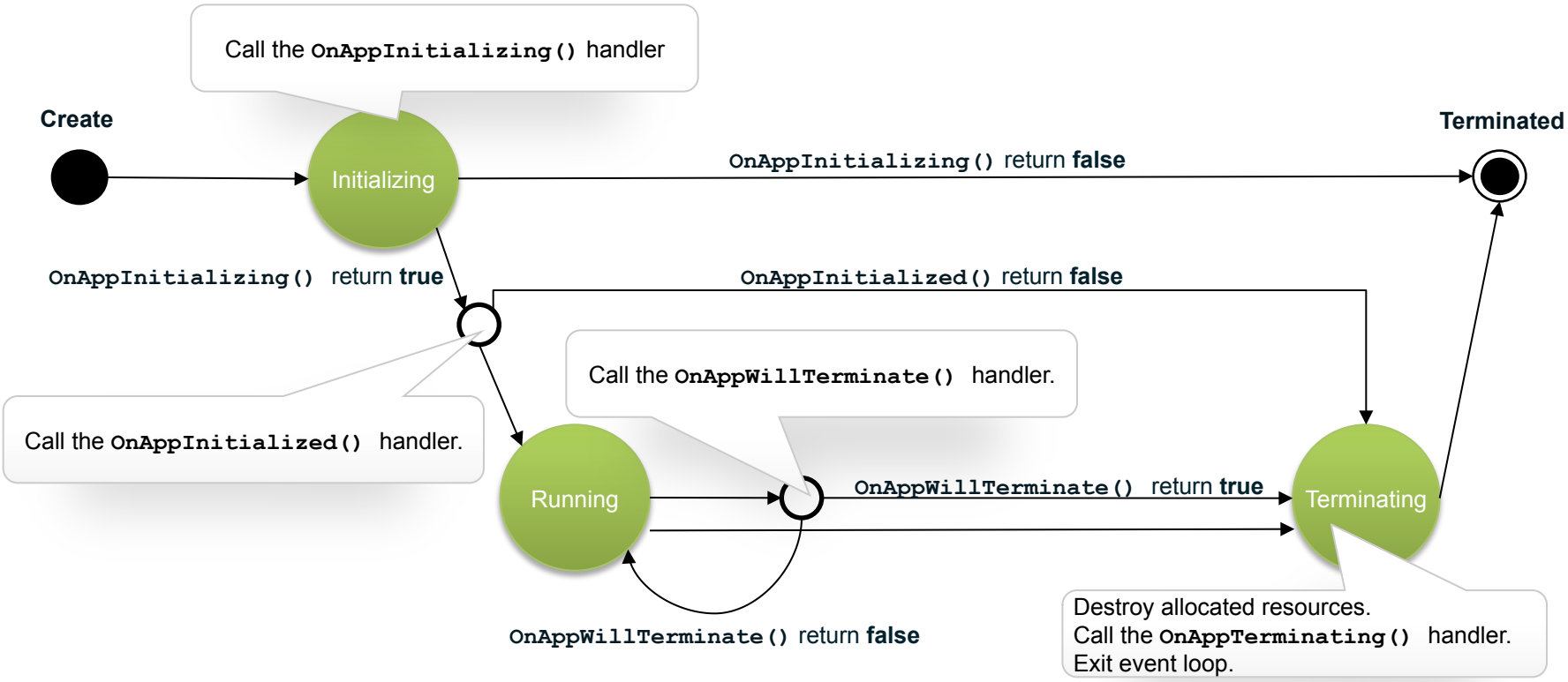
- **There are 2 types of Tizen native applications:**
  - UI application
    - Has a graphical user interface
    - Inherits from the **`Tizen::App::UiApp`** class
    - Handles the **`OnForeground`** / **`OnBackground`** events
  - Service application
    - Has no graphical user interface
    - Inherits from the **`Tizen::App::SerivceApp`** class
    - Runs on background♪



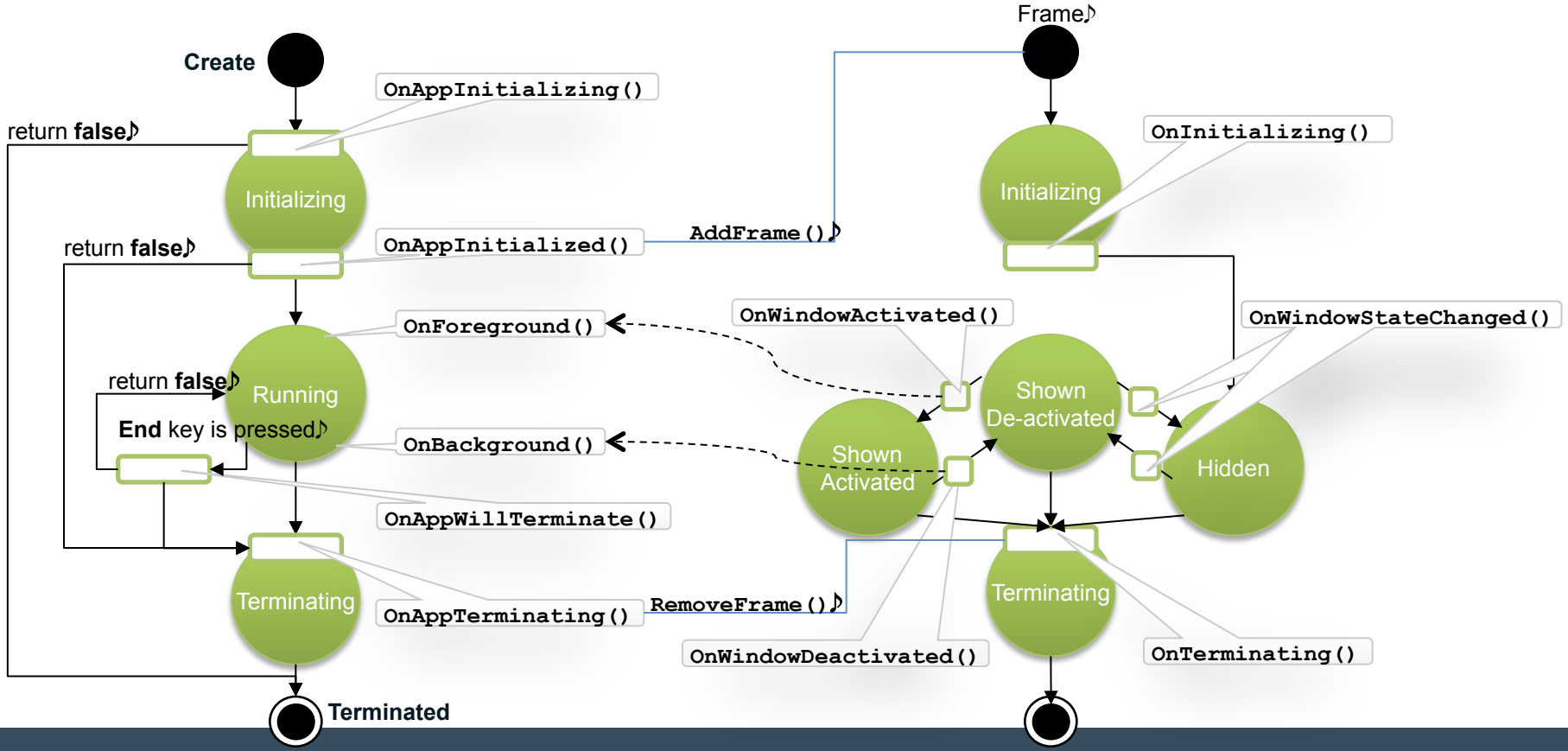
Application | Type

	UI application♪	Service application♪
Definition	Application with UI♪	Application without UI♪
Base class♪	`Tizen::App::UiApp`♪	`Tizen::App::ServiceApp`♪
Launched by♪	Main menu, task switcher, other apps, or condition♪	On booting, other apps, or condition♪
Main menu icon♪	Visible♪	Invisible♪
Task switcher♪	Visible♪	Visible (separate)♪
Foreground♪	Possible♪	Impossible♪
Background♪	Possible♪	Always♪
Tag in the manifest file♪	`<UiApp></UiApp>`♪	`<ServiceApp></ServiceApp>`♪

Application | Life-cycle



Application | Life-cycle (UI Application)



Application | Life-cycle – Launching (1/2)

- **An application is launched when:**
  - User clicks the icon in the main menu
  - Other application requests launching using launch-related APIs:
    - `Tizen::App::AppManager::LaunchApplication`
    - `Tizen::App::AppManager::RegisterAppLaunch`
    - `Tizen::App::AppControl::Start`
    - `Tizen::Shell::NotificationManager::NotifyByAppId`
- **In the application initialization phase:**
  - System loads the necessary libraries and the app executable binary to the memory
  - App instance is created and executed
  - In the **`OnAppInitializing()`** handler resources, UI components, and previous app states can be loaded or initialized

Application | Life-cycle – Launching (2/2)

- Implementing `OnAppInitializing()`:

```
bool
BasicApp::OnAppInitializing(AppRegistry& appRegistry)
{
	int lastPanelId = 0;
	…

	result r = appRegistry.Get(panelIDkey, lastPanelId);
	…
	r = appRegistry.Get(panelNamekey, lastPanelName);
	…
	// Create & add a Frame
	BasicAppFrame* pBasicAppFrame = new (std::nothrow) BasicAppFrame();
	pBasicAppFrame->Initialize(lastPanelId);
	pBasicAppFrame->SetName(L"BasicApp");
	AddFrame(*pBasicAppFrame);
♪
	return true;
}
```

Application | Life-cycle – Termination (1/2)

- **Application is terminated when**
  - The `Tizen::App::Terminate()` method is called
  - Another application requests termination by calling the `Tizen::App::AppManager::TerminateApplication()` method
  - The application is in the foreground and the user presses the END key
  - The user selects “End” in the task switcher
  - The number of concurrent Tizen applications exceeds the limitation defined by the system policy
  - The system memory or power is extremely low

Application | Life-cycle – Termination (2/2)

- **In the `OnAppTerminating()` handler, the following requirements must be met:**
  - Application returns quickly
  - Application frees the resources it has been using to avoid memory leakage
  - Application closes all pending connections to servers
  - App can save its state through the `AppRegistry` class

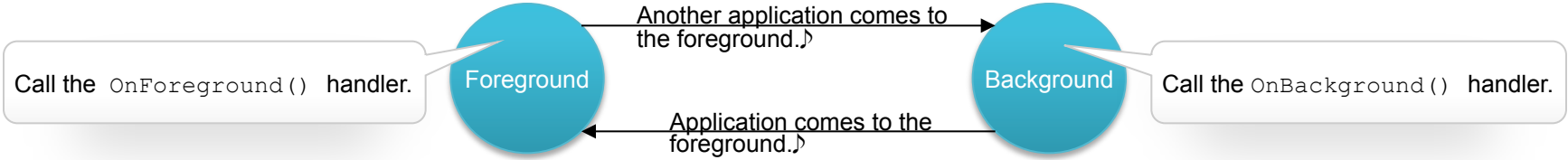
```
bool
BasicApp::OnAppTerminating(AppRegistry& appRegistry, bool forcedTermination)
{
	String panelIDkey(L"AppLastPanelId");
	String panelNamekey(L"AppLastPanelName");

	appRegistry.Add(panelIDkey, lastPanelId);
	appRegistry.Add(panelNamekey, lastPanelName);

	return true;
}
```

Application | Life-cycle - Foreground

- **Application is visible when it is in the foreground state**
  - Application is brought to the foreground from the background when:
    - Application is selected from the task switcher
    - Application icon is selected in the main menu
  - When an application is changed to be visible, the `OnForeground()` handler is called
  - When the application is brought to the foreground :
    - Application is recommended to handle graphics processing (3D or animation) since the application becomes visible
    - It is better to resume other operations that were stopped when the application was last moved to the background♪

Application | Life-cycle - Background

- **Application is invisible when it is in the background state**
  - An application is sent to the background:
    - When another application pops up and hides current application frame (for example, the user presses the HOME key, and the main menu is displayed)
  - When the application is sent to the background :
    - Graphics processing (such as 3D or animation) is better to be stopped since the graphics are not being displayed
    - It is recommended to release unnecessary resources and stop media processing and sensors manipulation♪



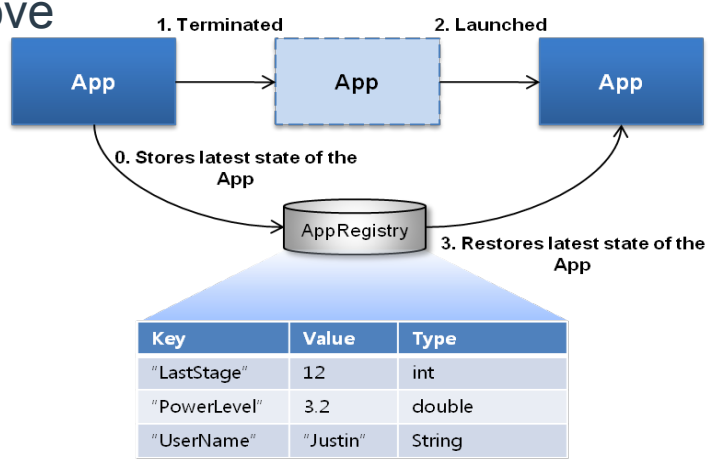
Application | Life-cycle – System Event Handling

- **Application can receive various system events**
  - Battery events
    - `OnBatteryLevelChanged()` handler
    - It is recommended that the application consuming more battery power is terminated if the battery level is `Tizen::System::BATTERY_LEVEL_CRITICAL`
  - Memory events
    - When the system-wide memory or application heap memory is insufficient to run the application any further, the `OnLowMemory()` handler is called to free unused memory resources
  - Screen events
    - `OnScreenOn()/OnScreenOff()` handlers
    - The resources must be handled efficiently by the `OnForeground/OnBackground` and `OnScreenOn/OnScreenOff` event handlers. Do not to duplicate or delete resources♪

Application | Configuration (1/3)

- **AppRegistry**
  - App can save or restore its custom states, such as user preferences, configuration information, and current game scores
  - Key-value-style data accessing
  - Supported operations: Add, Set, Get, Remove
  - Supported data types: String, Int, Double

```
bool
BasicApp::OnAppInitializing(AppRegistry& appRegistry)

bool
BasicApp::OnAppTerminating(AppRegistry& appRegistry,…)
```

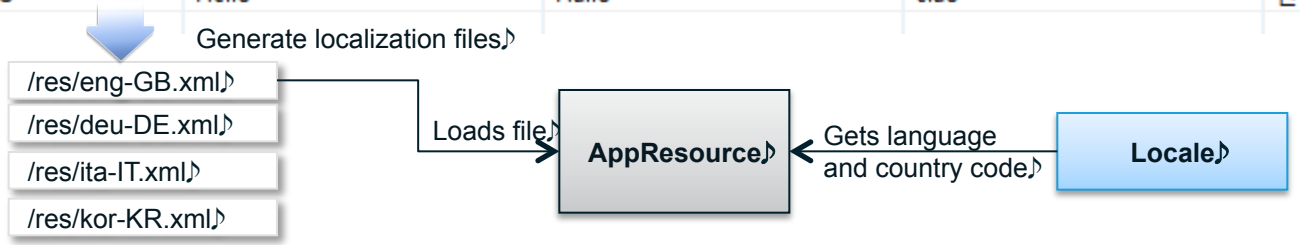


Application | Configuration (2/3)

- **AppResource**
  - `GetString()`: Load string resources for localization

String

No	ID	English (England(UK))	German (Germany)	Italian (Italy)	Korean (Korea)
1	IDS_HELLO	Hello	Hallo	ciao	안녕하세요

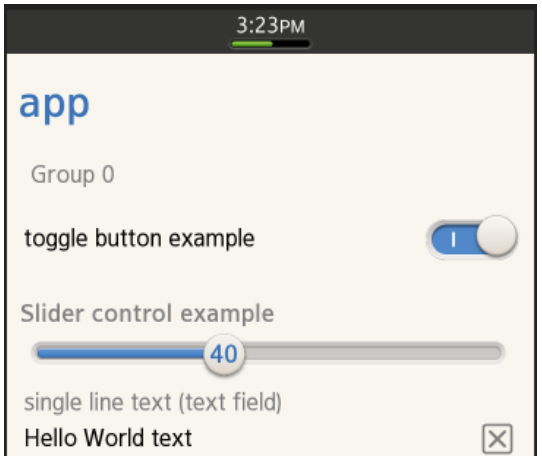


  - `GetBitmapN()` : Load graphics resources for multi-resolution (for example, HD and WVGA bitmap resources)

Application | Configuration (3/3)

- **AppSetting**
  - System setting provides setting menu for downloaded applications
  - You can design your own setting menu with IDE
  - Application can access setting values using the **`Tizen::App::AppSetting`** class

Value	
app	<setting>
id=	IDP_PAGE1
main=	true
app	<navigationbar>
Group name	<group>
id=	IDG_GRP1
toggle button example	<bool>
Slider control example	<integer>
id=	IDI_SLD1
min=	0
max=	100
value=	40
single line text (text field)	<string>



Application (Advanced)

Conditional Application Launch

- Application can register a launching condition for itself or other applications
- Framework launches a registered application when the condition is matched

Type♪	Condition format♪	Description♪
Time	`L"DateTime='mm/dd/yyyy hh:mm:ss'"`	Launch at the wall time, 'mm/dd/yyyy hh:mm:ss'. The time format matches the output format of the `Base::DateTime::ToString()` method.
	`L"DueTime='mm/dd/yyyy hh:mm:ss' LaunchPeriod='mm'"`	Launch at the wall time, 'mm/dd/yyyy hh:mm:ss', and every 'mm' minutes. The launch period must be between 60 minutes and 10 years.♪
Serial♪	`L"Serial='keyword'"`♪	Launch when receiving a keyword as serial communication input.
NFC♪	`L"NFC='command'"`♪	Launch when receiving command, as an NFC tag with NFC Data Exchange Format (NDEF) data. ♪

Inter-application Operation | AppControl (1/2)

- **Application can use an exported functionality by other application**
- **Functionality is defined by:**
  - Operation ID
  - URI
  - MIME type
- **AppControl resolution:**
  - Explicit AppControl resolution
    - **`Tizen::App::AppManager::FindAppControlN()`**
    - By AppId or operation ID
  - Implicit AppControl resolution
    - **`Tizen::App::AppManager::FindAppControlsN()`**
    - By operation ID, URI, MIME type, or category♪

```
String telUrl = L"tel:1234567900";
AppControl* pAc = AppManager::FindAppControlN(
      L"tizen.phone",
      L"http://tizen.org/appcontrol/operation/dial");
pAc->Start(&telUrl, null, null, null);
```

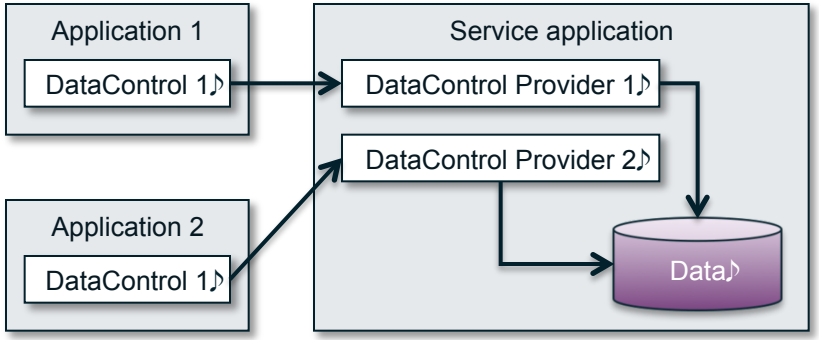
Inter-application Operation | AppControl (2/2)

- **Platform defined AppControl Operation IDs:**

Operation ID♪	Description♪
`http://tizen.org/appcontrol/operation/main`	Reserved for application main entry point
`http://tizen.org/appcontrol/operation/call`	Make a call to the specified phone number
`http://tizen.org/appcontrol/operation/create_content`	Create content, such as an image captured with the camera
`http://tizen.org/appcontrol/operation/compose`	Compose the content
`http://tizen.org/appcontrol/operation/dial`	Launch the dial screen for making a call
`http://tizen.org/appcontrol/operation/multi_share`	Share multiple content items
`http://tizen.org/appcontrol/operation/pick`	Display the list of items and return the selected item
`http://tizen.org/appcontrol/operation/share`	Share the content
`http://tizen.org/appcontrol/operation/view`	Display or play the content

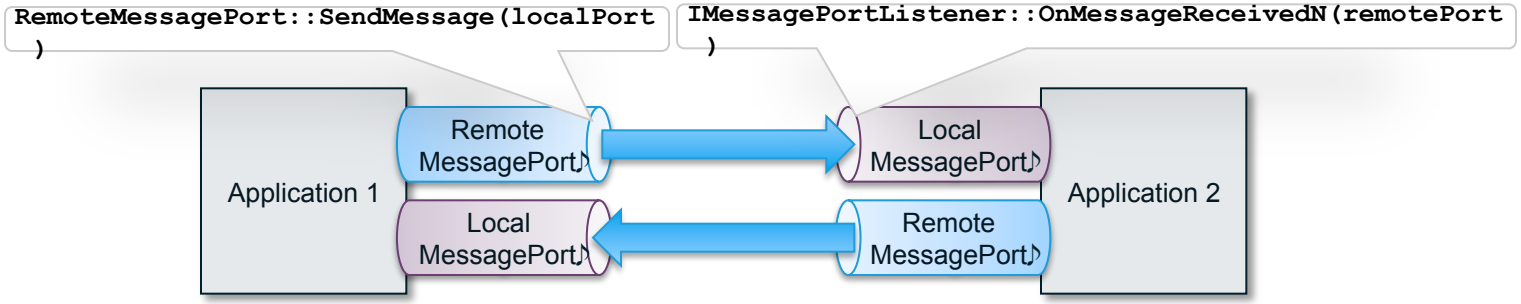
Inter-application Operation | DataControl

- **Application can use shared data by other application:**
  - Use the `Tizen::Io::File` for `[App Root]/shared/data` directory
  - Use DataControl
- **DataControl types:**
  - `Tizen::App::SqlDataControl`
    - Use SQL-type data access
  - `Tizen::App::MapDataControl`
    - Use key-value-type data access
- **DataControl resolution**
  - Explicit, by provider ID and data ID



Inter-application Operation | Message Port

- **Application can send and receive messages through a message port**
- **Message port types:**
  - **`Tizen::Io::RemoteMessagePort`**
    - Send messages to other applications
  - **`Tizen::Io::LocalMessagePort`**
    - Receive messages from another application



System Information

- **Tizen::System::SystemInfo**
  - `GetBuildInfo()`
  - `GetPlatformVersion()`
  - `GetValue()`

Key♪
http://tizen.org/feature/camera
http://tizen.org/feature/camera.back
http://tizen.org/feature/camera.back.flash
http://tizen.org/feature/camera.front
http://tizen.org/feature/camera.front.flash
http://tizen.org/feature/database.encryption
http://tizen.org/feature/fmradio
http://tizen.org/feature/graphics.acceleration
http://tizen.org/feature/input.keyboard
http://tizen.org/feature/input.keyboard.layout
http://tizen.org/feature/location
http://tizen.org/feature/location.gps
http://tizen.org/feature/location.wps
http://tizen.org/feature/microphone
http://tizen.org/feature/multi_point_touch.pinch_zoom
http://tizen.org/feature/multi_point_touch.point_count
http://tizen.org/feature/network.bluetooth
http://tizen.org/feature/network.nfc
http://tizen.org/feature/network.nfc.reserved_push
http://tizen.org/feature/network.push
http://tizen.org/feature/network.secure_element
… more …

System Information

- **Tizen::System::SettingInfo**
  - `SetValue()`
  - `SetValueAsync()`
  - `GetValue()`
  - Application can set an event listener

Key♪
http://tizen.org/setting/application.home
http://tizen.org/setting/application.lock
http://tizen.org/setting/battery.format.percentage
http://tizen.org/setting/contacts.order.firstname
http://tizen.org/setting/developer.usb_debugging
http://tizen.org/setting/device_name
http://tizen.org/setting/font.size
http://tizen.org/setting/font.type
http://tizen.org/setting/graphics.gpu.rendering
http://tizen.org/setting/locale.country
http://tizen.org/setting/locale.date
http://tizen.org/setting/locale.date.format
http://tizen.org/setting/locale.date_time
http://tizen.org/setting/locale.date_time.format
http://tizen.org/setting/locale.language
http://tizen.org/setting/locale.time
http://tizen.org/setting/locale.time.format
http://tizen.org/setting/locale.time.format.24hour
http://tizen.org/setting/locale.time_zone
http://tizen.org/setting/locale.update.auto
… more …

System Information

- **Tizen::System::RuntimeInfo**
  - `GetValue()`
  - `GetValueAsync()`

Key♪
http://tizen.org/runtime/cpu.core.all.usage
http://tizen.org/runtime/memory.allocated
http://tizen.org/runtime/memory.allocated.self
http://tizen.org/runtime/memory.allocated.video
http://tizen.org/runtime/memory.available
http://tizen.org/runtime/memory.available.video
http://tizen.org/runtime/storage.allocated.external.application
http://tizen.org/runtime/storage.allocated.external.audio
http://tizen.org/runtime/storage.allocated.external.download
http://tizen.org/runtime/storage.allocated.external.image
http://tizen.org/runtime/storage.allocated.external.video
http://tizen.org/runtime/storage.allocated.internal.application
http://tizen.org/runtime/storage.allocated.internal.audio
http://tizen.org/runtime/storage.allocated.internal.download
http://tizen.org/runtime/storage.allocated.internal.image
http://tizen.org/runtime/storage.allocated.internal.video
http://tizen.org/runtime/storage.available.external
… more …

TIZEN™

DEVELOPER CONFERENCE 2013 SAN FRANCISCO