Tizen Technical Overview
Contents

• What is Tizen
• Tizen architecture
• Tizen architecture (IVI profile)
• Tizen OS development tools
What Is Tizen

• A robust and flexible, open source, standards-based software platform based on HTML5.
• For smartphones, tablets, in-vehicle infotainment (IVI), PCs and more.
• A robust OS, app store, apps and open services model that consumers can take from device to device.
• Allows OEMs and Service Providers flexibility to customize the UX, the app store and services.
• Open source project resides within the Linux Foundation and is governed by the Technical Steering Group.
• Tizen Association drives ecosystem support and market adoption.
• Tizen SDK, APIs and Tools enable both HTML5 (web-based) apps as well as native.

Tizen is a trademark of the Linux Foundation
As of 2.0 release, Tizen leads all other mobile platforms in support of HTML5

- Highest on both html5test score and bonus points – 492 out of possible 500!
- Receives max bonus points of 16

Tizen also best in class for Ringmark – which measures functionality needed to develop mobile apps
Tizen 2.x source code and SDK release

- Added HTML5/W3C APIs Drag & Drop, Clipboard APIs, Media Capture
- Added hybrid and native app support
- Enhanced Web Runtime (Webkit2)
- Added Tizen device APIs for Bluetooth, NFC, Web Audio, etc.
- New and improved reference apps
- Enhanced Web and Native IDEs providing WYSIWYG design environment
- Systemd replaces init daemon
- Content Security Policy, Inter-app sharing
Architecture Overview in Detail

- Web framework
  - Provides state-of-the-art HTML5/W3C APIs, Web UI framework, supplementary APIs, and additional Tizen device APIs
- Native framework
  - Supports full-featured native application development and provides a variety of features like background service, image and face recognition, and TTS/STT
- Core
  - Underlying layer for Web and native providing common functionalities and a security mechanism
  - HW adaptation layer with plug-in architecture
  - OpenGL® ES/EGL graphics driver
Native and Web frameworks are complementary to each other

- Web is strong in portability, ease of app development, and has a minimal learning curve
- Native is relatively better in terms of performance and memory consumption
- Native enables reusing the existing engine and libraries written in C & C++ in app development
Web and Native: Mix & Match

- Different combinations for mixing Web and native, depending on the characteristics or requirements of the app to be developed
Native Framework vs. Core

- Both are native in nature but focusing on different aspects
- Core focuses on:
  - Providing common functionalities to Web and native frameworks
  - No need to guarantee app binary compatibility (ABC)
  - Performance and power optimization
- Native framework focuses on:
  - Application development productivity while guaranteeing ABC
  - Well-documented API references, developer guide, sample codes, and associated tools
Application Types

- Web and native applications
  - Apps using only public APIs to get full support for package installation and upgrade, security, backward compatibility, and so on
  - Many sample apps included in the SDK
- Core applications
  - Apps using Core APIs to fully utilize device capabilities such as telephony
  - Usually implemented and preloaded by device implementers
  - Backward binary compatibility is not guaranteed
Web Framework

- W3C standard Web APIs
  - W3C/HTML5 markup, CSS, and JavaScript APIs
- Supplementary APIs
  - De-facto APIs (such as Khronos and Mozilla)
- Tizen Device APIs
  - Advanced access to the device’s platform capabilities
- UI framework
  - jQueryMobile-based
  - Tools, such as widgets, events, effects, and animations
Web Runtime

- **Package management** (such as installation and update)
- **Execution and lifecycle** (such as launching, pause, and resume)
- **Runtime security** (such as API/network access and sandboxing)
- **Platform integration**
WebKit2 based Browser and Web Runtime

- Since 2.0, Tizen is using WebKit2 (http://www.webkit.org)
  - Split process model for web content and UI with non-blocking APIs
  - UI responsiveness, robustness, security, and better use of multicore CPUs
Web 2D and 3D Graphics

- **HTML5 Canvas** is accelerated by
  - Cairo OpenGL® ES backend
- **WebGL**
  - Directly uses OpenGL® ES
  - Triple buffering

2D Canvas performance score

WebGL fps test

[Source: canvaperf, ie10testdrive] [Source: Google Experiments, Tizen Demo]
Native Framework

- Released in Tizen 2.0
- Set of C++ namespaces with more than 10,000 APIs
  - Base, IO, App, Security, Graphics and UI, Net, Messaging, Social, Locations, Web, etc
- Support for standard C/C++, and popular open source libraries
  - eglibc, STL, libstdc++, libxml2, OpenGL® ES, OpenAL, and OpenMP®
- Multiprocessing support
  - OpenMP, GCD
Core

- Providing common features
  - Various native and Web APIs are implemented using the functionalities of core modules

- Unified management for:
  - Package (un)installation and upgrade
  - Launching applications
  - Windows for different apps with E17
  - Power consumption
  - Connectivity
  - Security enforcement with Smack from the kernel
  - And more..
IVI Demands more from Tizen

- Navigation GPS
- Dual Display
- TTS/STT
- Radio
- BluRay playback
- Front - video analytics
- Audio
- Mobile Device
- Display
- Video playback
- Rear - video analytics
- Audio
- Display
Tizen IVI Release History

Tizen 1.0 (2012)
- GENIVI Compliance
  - Fastboot with systemd < 5 secs
  - Rootfs < 500 Mb
  - Sample Navigation App
  - Sample Hands free dialer App
  - Media Player App
  - IVI Home Screen App

Tizen 2.0 (Apr. 2013)
- Fully functional Web framework
  - Automotive Message Broker
    - HTML5 application
  - BT HFP dialer application
  - DLNA
  - Murphy Policy Manager
  - WiFi Tethering
  - Dual Display Support
  - Sample IVI apps

Tizen Next
- Focus areas
  - Wayland
  - Fast Boot
  - Small Footprint
  - Ethernet
  - AVB Murphy +
  - NFC
  - HW Acceleration
  - Vehicle & Additional Web APIs for Automotive
  - UI Manager
Package Development Procedure

- **Local Git**: Developer
  - push
  - reject
  - 'gbs-submit'

- **Gerrit**: Reviewers
  - review
  - accept & merge to git

- **Git**: Product Project
  - submit SR to OBS which triggered by the tag
  - Image creation

- **OBS**: Release Engineers

- **Release**: QA

Flow:
- Manual-action
- Auto-action
Source Code Management

- **Git**
  - A particularly powerful, flexible, and low-overhead version control system that makes collaborative development efficient and robust
  - [https://review.tizen.org/git/](https://review.tizen.org/git/)

- **Gerrit**
  - A web-based code review system, facilitating online code reviews for projects using Git version control system
  - Gerrit optimizes the code review process, enhancing review quality
  - Gerrit simplifies the maintenance of the Git-based projects, enabling a more centralized use of Git
  - [https://review.tizen.org/gerrit](https://review.tizen.org/gerrit)
Git Building System

- **GBS (git-build-system)** is a developer command line tool that supports Tizen* OS package development
- It's used to generate tarballs based on Git repositories, to do local test buildings, and to submit code to OBS (Tizen's main build service)
- To use GBS, a development environment needs to be set up ready first
Image Creation

- MIC is an image creator. It's used to create images for Tizen*
- Users can create images of different types for different verticals, including live CD images, live USB images, raw images for KVM, loop images for IVI platforms, and fs images for chrooting
- Users can chroot into an image using MIC's enhanced chroot command
- MIC enables converting an image to another image format, a very useful function for those sensitive to image format
Tizen* OS Bug Tracking

• Tizen* uses JIRA to track bugs and to gather feature requests
  • https://bugs.tizen.org/jira/secure/Dashboard.jspa
• Developers need a Tizen account created to
  • Add a new bug
  • Comment on an existing bug
  • Submit a patch to fix bug
• To work on Tizen bug reporting and tracking, a set of guidelines are defined
  • https://www.tizen.org/community/guidelines/bug-guidelines