HACKING TIZEN

THE OS OF EVERYTHING

AJIN ABRAHAM | @ajinabraham

YODLEE®

WHOMAI

- Application Security Engineer ,Yodlee
- Blogs at **opensecurity.in**
- Spoken at NULLCON, ClubHack, OWASP AppSec, BlackHat, Ground Zero Summit….
- Loves to learn NEW things.

DISCLAIMER

- All Images, Logos and Trademark belongs to their respective owners.
- All vulnerabilities discussed are responsibly disclosed to Tizen Security community.
- Personal View/Research, doesn’t reflect the views of my employer.

AGENDA

- **What is Tizen**
- **Why Tizen?**
- **Types of Tizen Application**
- **Tizen Architecture**
- **Tizen Application Structure**
- **Tizen Security Model**
- **Sandbox – SMACK**
- **WebKit2 on Tizen**
- **Quick Comparison – Android vs Tizen vs iOS**

- **Hacking Tizen**

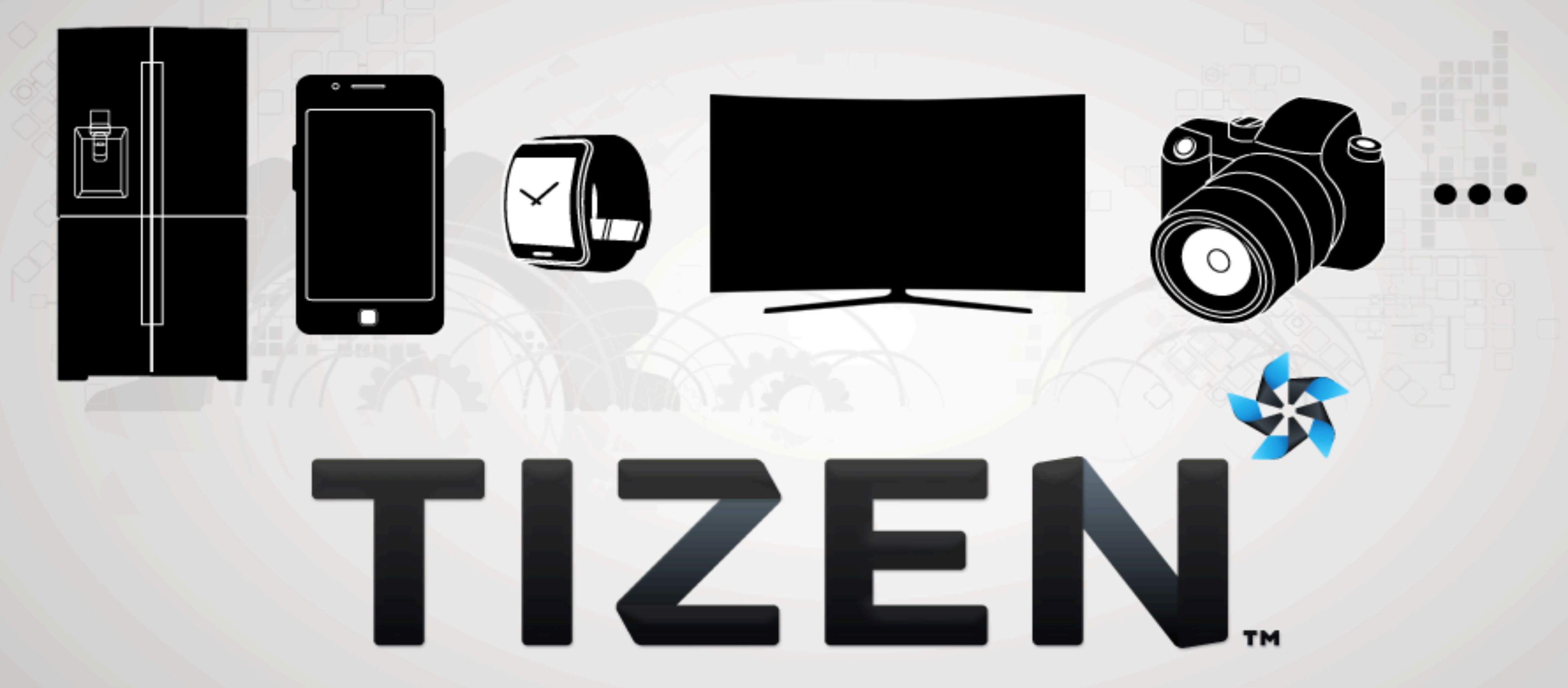
  * Android vs Tizen Web App
  * Shellshock
  * Issues in DEP
  * Broken ASLR
  * CSP Bypass
  * URL Spoofing/Content Injection

- **Pentesting Methodology**

  * Static Analysis
  * Dynamic Analysis
  * Network Analysis

- **Security Concerns in Tizen**
- **Conclusion**

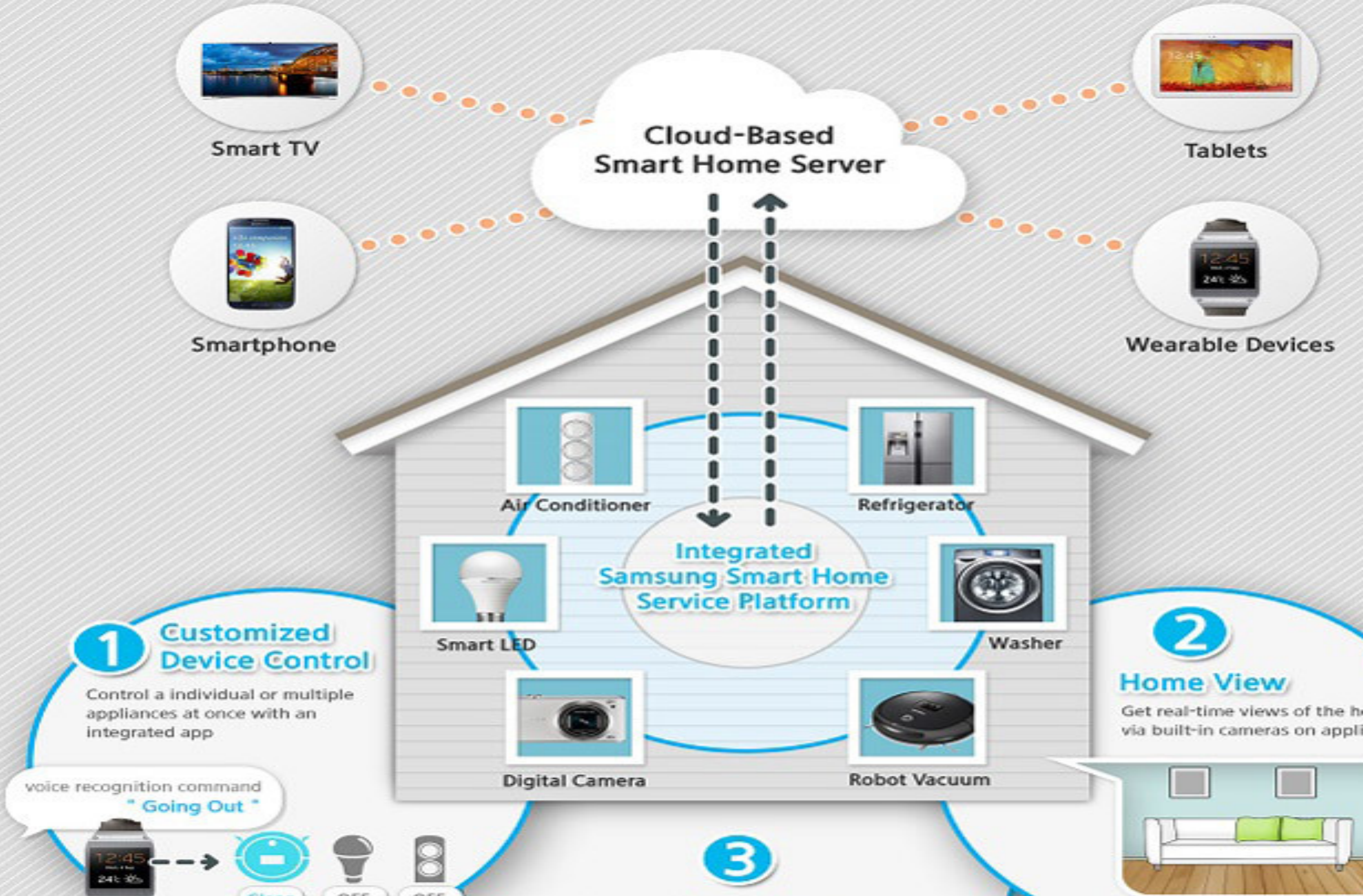
TIZEN : The OS of Everything



IoT (Internet of Things)

Tizen –A Linux Foundation Project.

Samsung Smart Home | 'Smart Living & Beyond'



Smart TV

Smartphone

Cloud-Based Smart Home Server

Tablets

Wearable Devices

Air Conditioner

Refrigerator

Smart LED

Integrated Samsung Smart Home Service Platform

Washer

Digital Camera

Robot Vacuum

1 Customized Device Control

Control a individual or multiple appliances at once with an integrated app

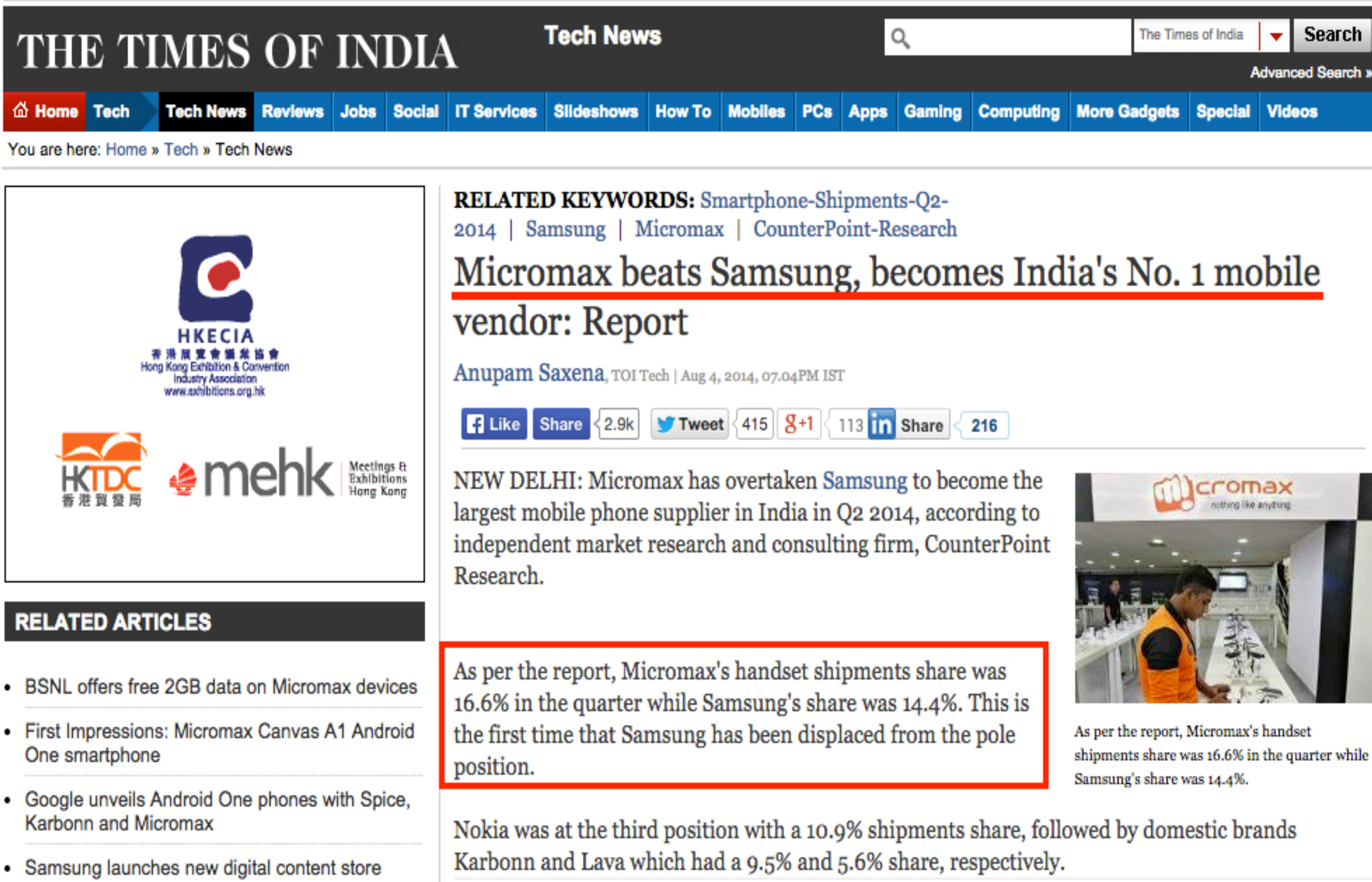
voice recognition command " Going Out "

2 Home View

Get real-time views of the h via built-in cameras on appli

3

Why TIZEN?



RELATED KEYWORDS: Smartphone-Shipments-Q2-2014 | Samsung | Micromax | CounterPoint-Research

Micromax beats Samsung, becomes India's No. 1 mobile vendor: Report

Anupam Saxena, TOI Tech | Aug 4, 2014, 07.04PM IST

NEW DELHI: Micromax has overtaken Samsung to become the largest mobile phone supplier in India in Q2 2014, according to independent market research and consulting firm, CounterPoint Research.

As per the report, Micromax's handset shipments share was 16.6% in the quarter while Samsung's share was 14.4%.

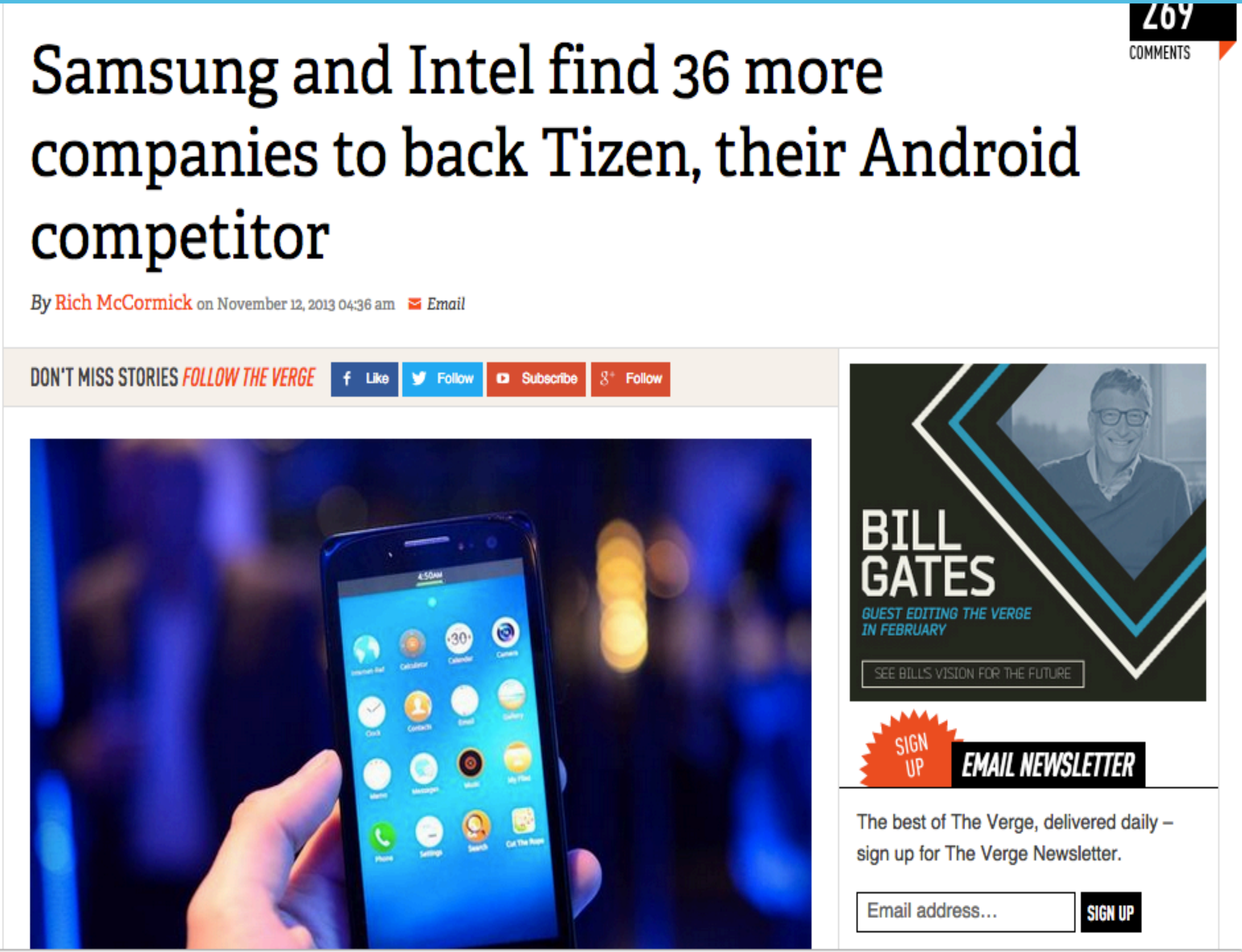
As per the report, Micromax's handset shipments share was 16.6% in the quarter while Samsung's share was 14.4%. This is the first time that Samsung has been displaced from the pole position.

Nokia was at the third position with a 10.9% shipments share, followed by domestic brands Karbonn and Lava which had a 9.5% and 5.6% share, respectively.

RELATED ARTICLES

- BSNL offers free 2GB data on Micromax devices
- First Impressions: Micromax Canvas A1 Android One smartphone
- Google unveils Android One phones with Spice, Karbonn and Micromax
- Samsung launches new digital content store

Source: http://ti mesofindia.indiatimes.com/tech/tech-news/Micromax-beats-Samsung-becomes-Indias-No-1-mobile-vendor-Report/articleshow/39630245.cms



269

COMMENTS

Samsung and Intel find 36 more companies to back Tizen, their Android competitor

By Rich McCormick on November 12, 2013 04:36 am ✉ Email

DON'T MISS STORIES *FOLLOW THE VERGE*

SIGN UP EMAIL NEWSLETTER

The best of The Verge, delivered daily – sign up for The Verge Newsletter.

Email address...

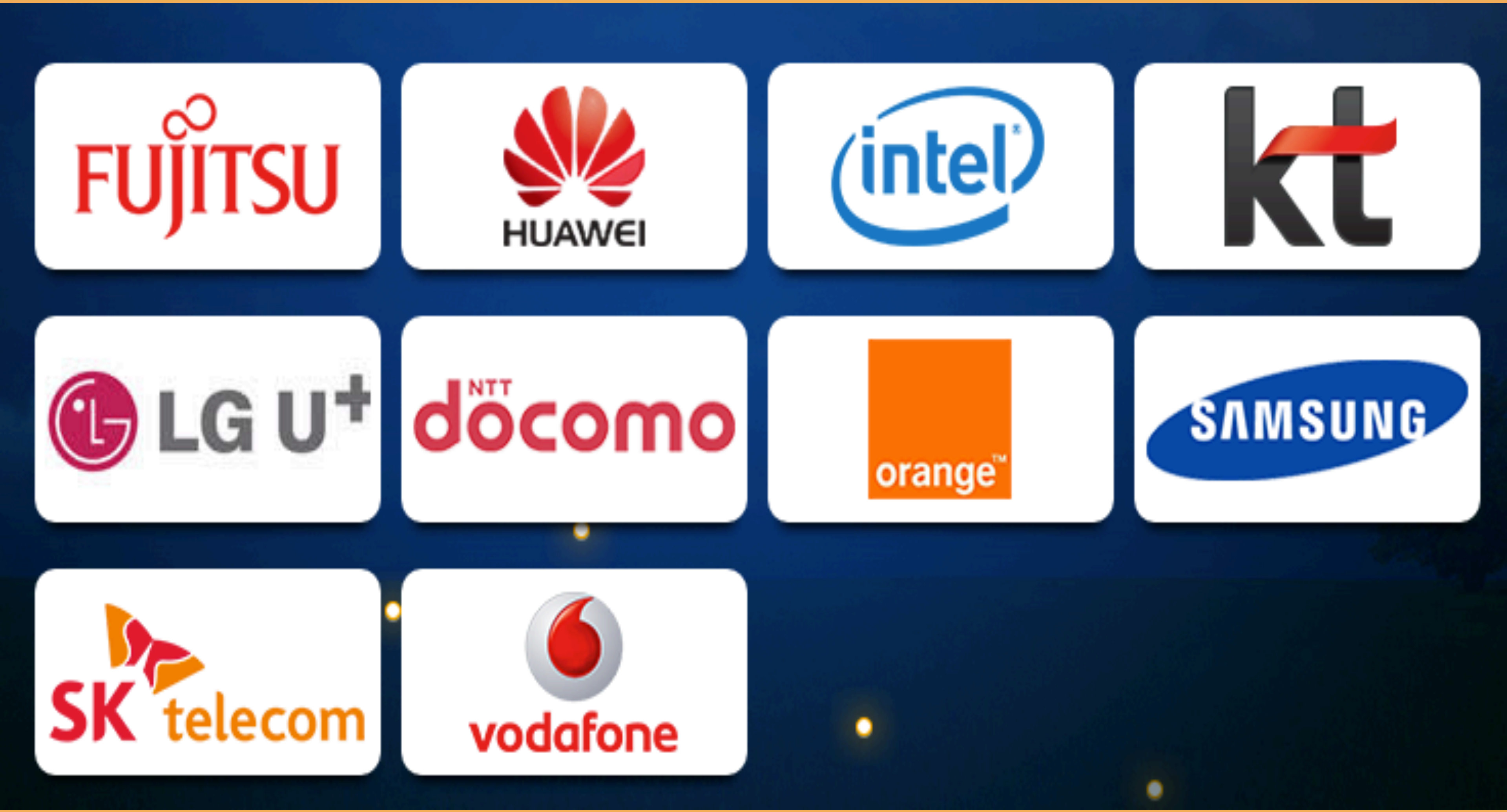
SIGN UP

Source: http://www.theverge.com/2013/11/12/5093588/tizen-open-operating-system-partners-with-36-companies

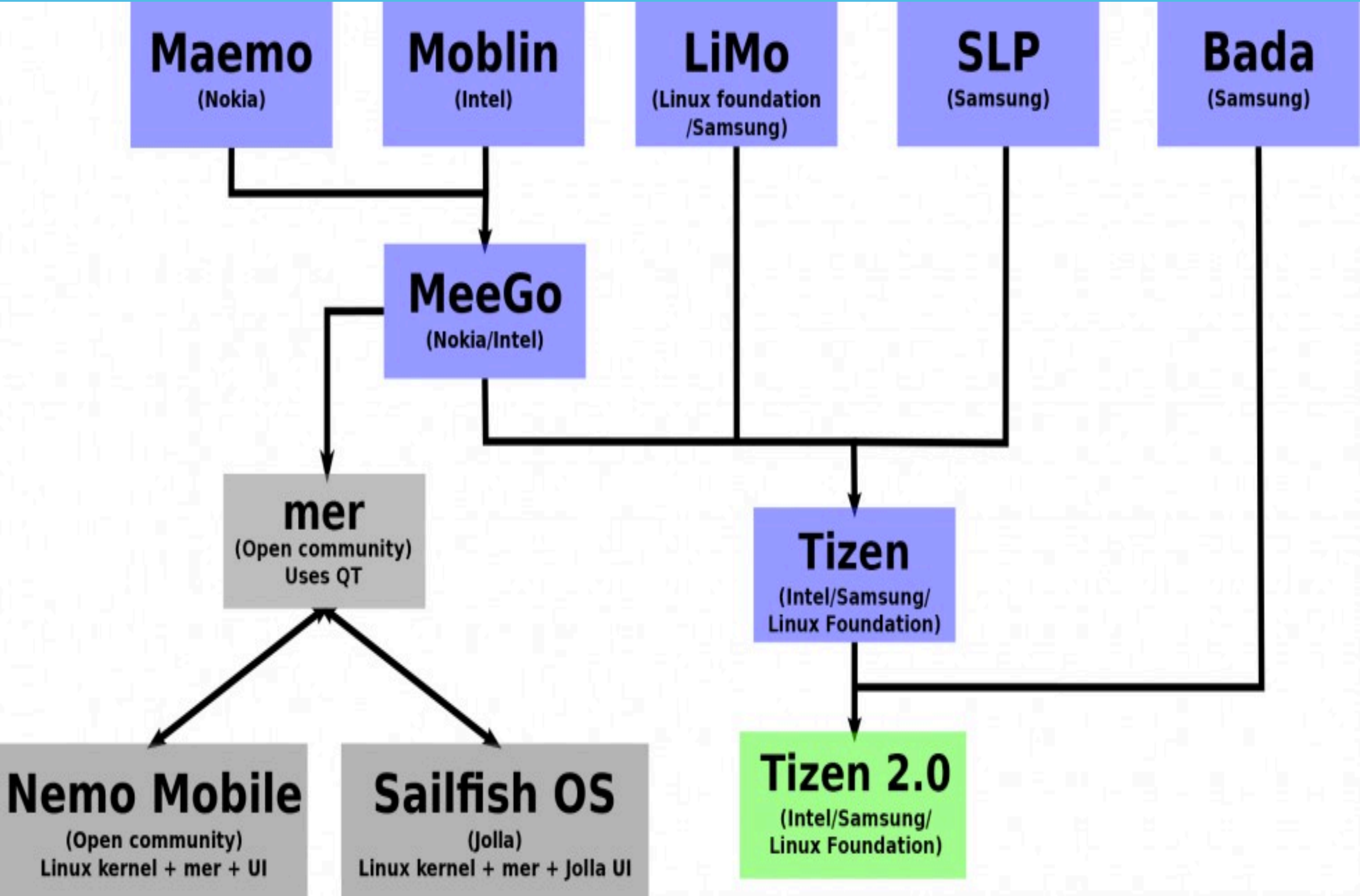
Samsung 2015 Tizen TV range now available at Curry's in the UK

News Smart TV Apr 9, 2015

Source: http://www.tizenexperts.com/2015/04/samsung-2015-tizen-tv-range-now-available-at-currys-in-the-uk/


FUJITSU
HUAWEI
intel
kt
LG U+
NTT docomo
orange™
SAMSUNG
SK telecom
vodafone

THE FAMILY



Maemo (Nokia)

Moblin (Intel)

LiMo (Linux foundation /Samsung)

SLP (Samsung)

Bada (Samsung)

MeeGo (Nokia/Intel)

mer (Open community) Uses QT

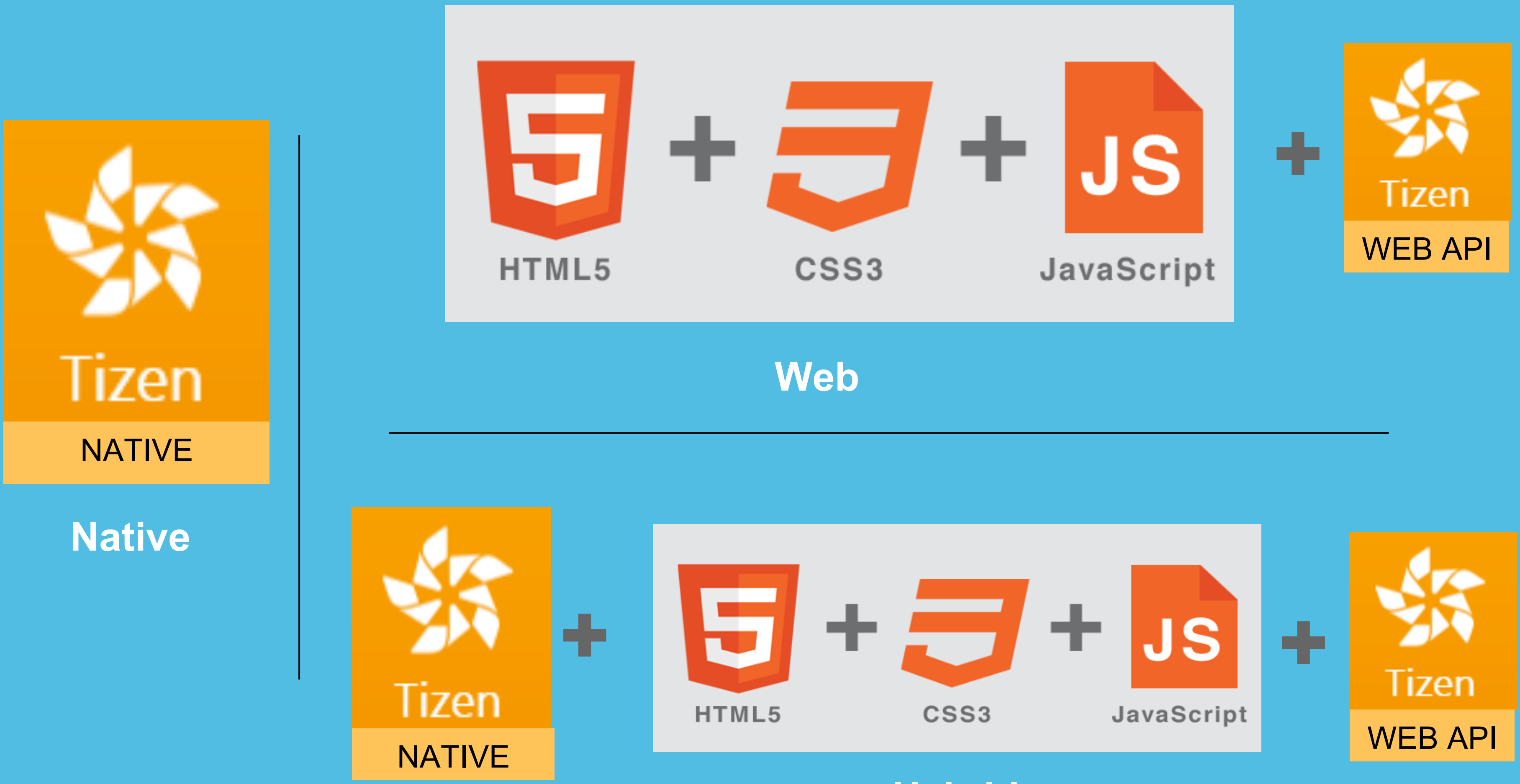
Tizen (Intel/Samsung/ Linux Foundation)

Nemo Mobile (Open community) Linux kernel + mer + UI

Sailfish OS (Jolla) Linux kernel + mer + Jolla UI

Tizen 2.0 (Intel/Samsung/ Linux Foundation)

TYPES OF TIZEN APLICATIONS



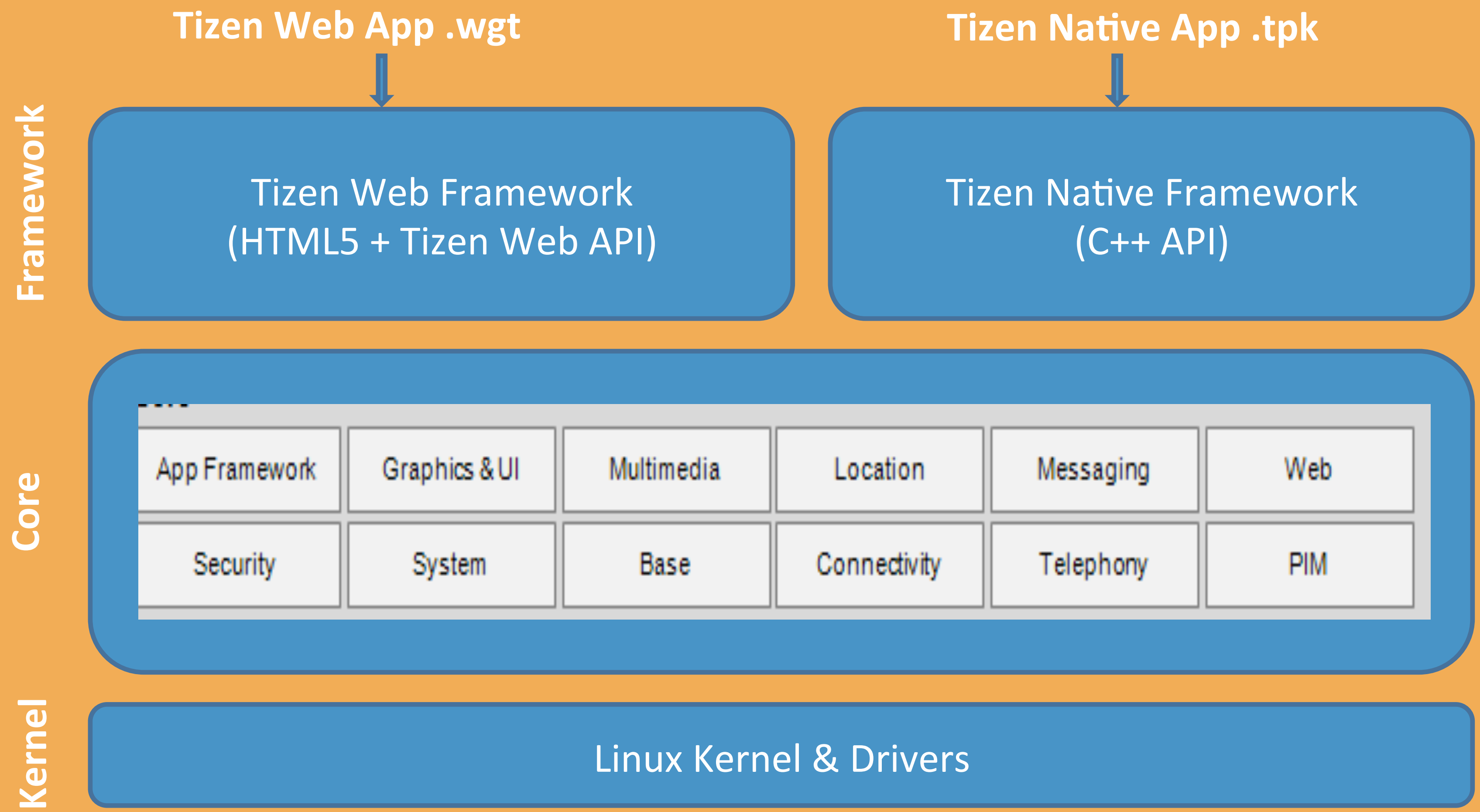
Native

Web

Hybrid

Supports Android application with Tizen Application Compatibility Layer (ACL).

TIZEN ARCHITECTURE



Tizen Web App .wgt

Tizen Native App .tpk

Framework

Tizen Web Framework
(HTML5 + Tizen Web API)

Tizen Native Framework
(C++ API)

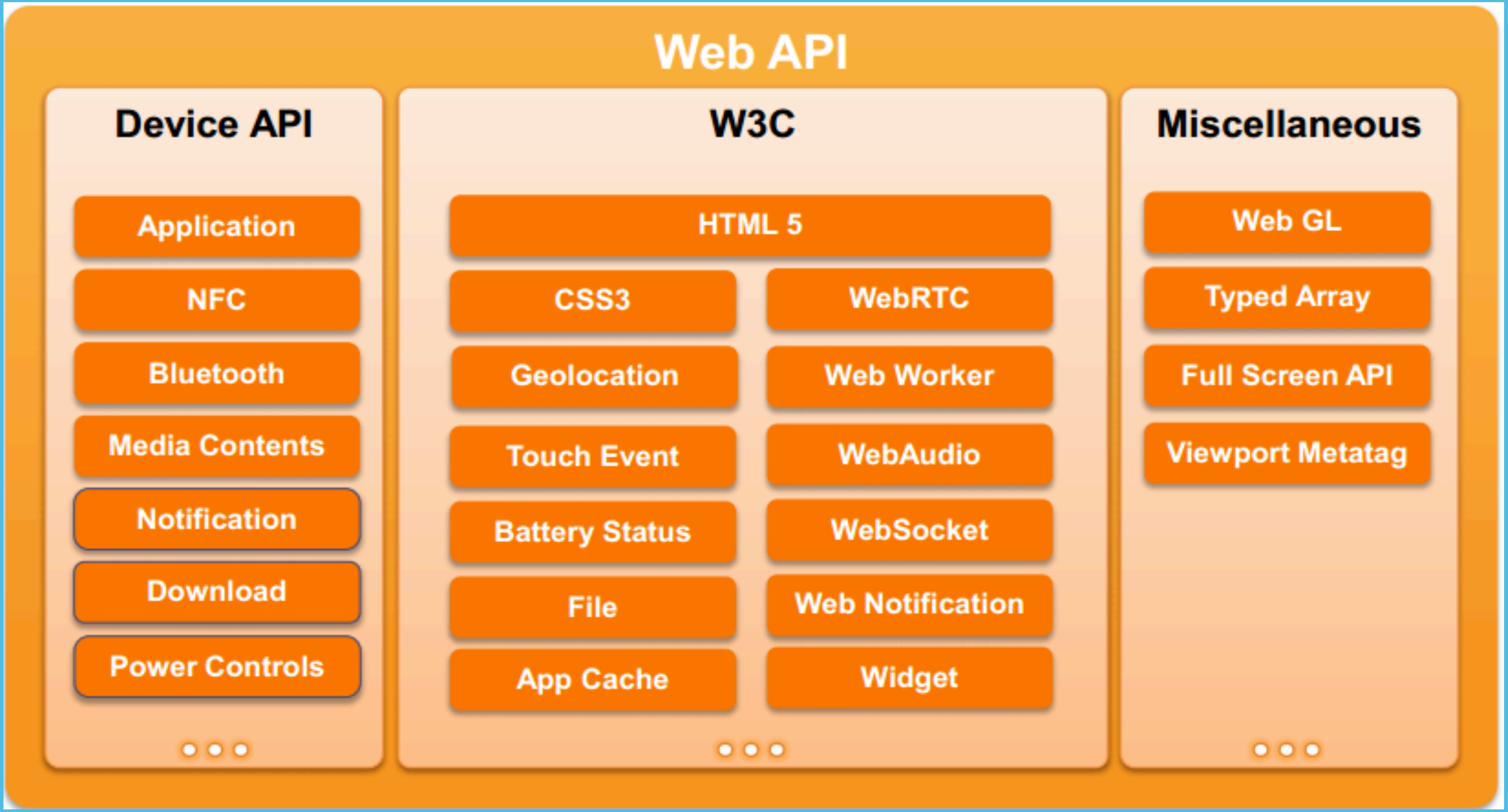
Core

App Framework	Graphics & UI	Multimedia	Location	Messaging	Web
Security	System	Base	Connectivity	Telephony	PIM

Kernel

Linux Kernel & Drivers

Web API = Standard HTML5 + Tizen Device API



Web API

Device API

- Application
- NFC
- Bluetooth
- Media Contents
- Notification
- Download
- Power Controls
- ○○○

W3C

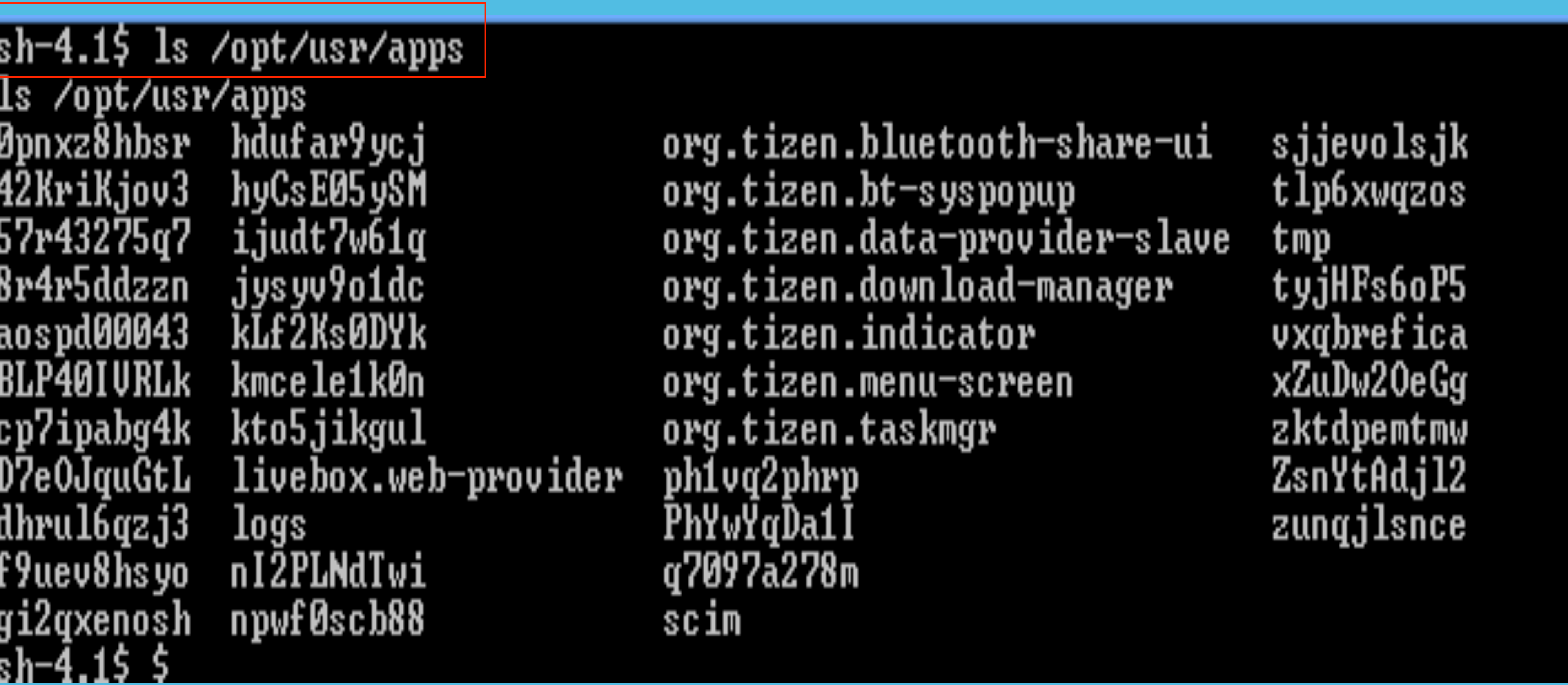
- HTML 5
- CSS3
- WebRTC
- Geolocation
- Web Worker
- Touch Event
- WebAudio
- Battery Status
- WebSocket
- File
- Web Notification
- App Cache
- Widget
- ○○○

Miscellaneous

- Web GL
- Typed Array
- Full Screen API
- Viewport Metatag
- ○○○

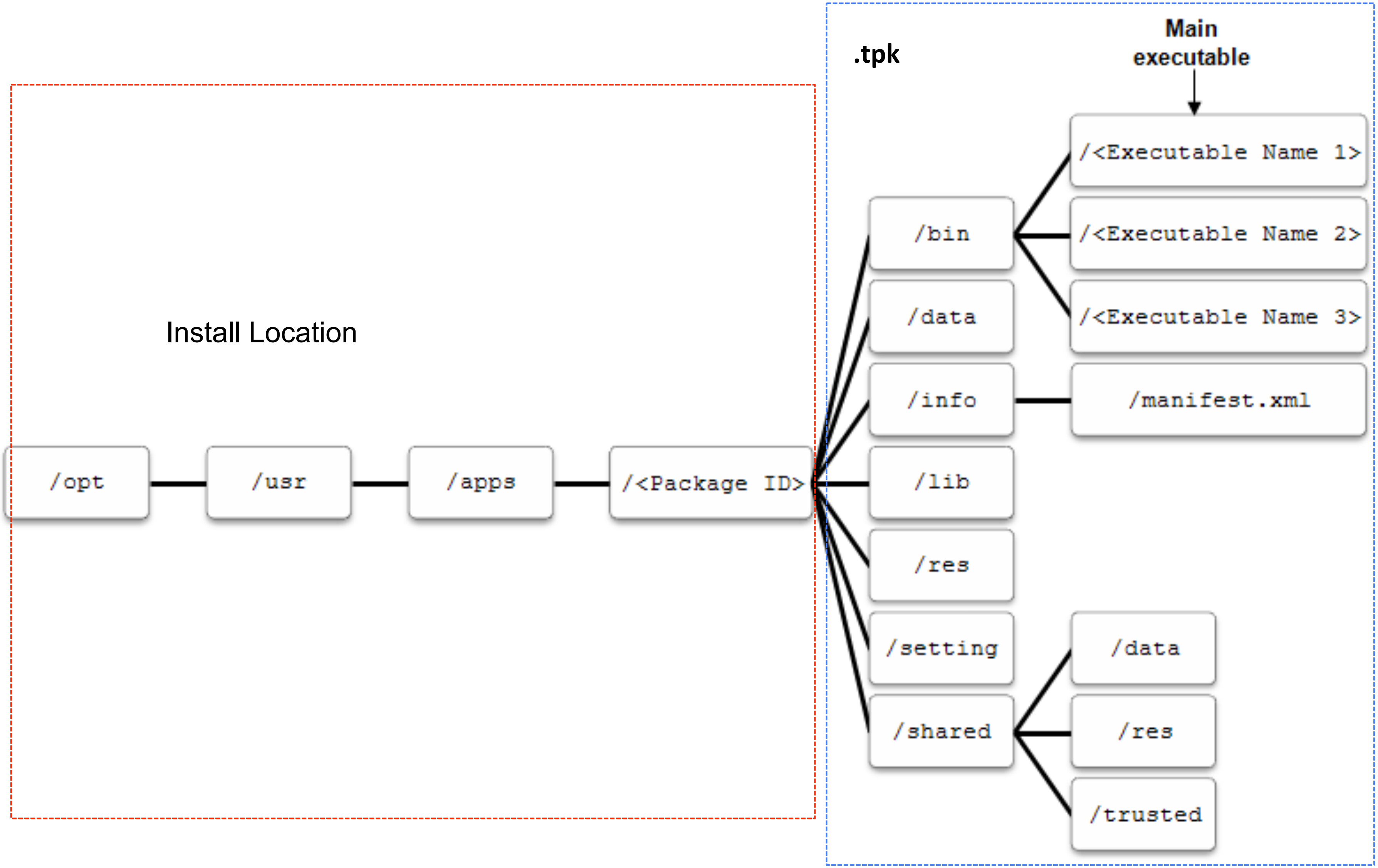
TIZEN APPLICATION STRUCTURE

INSTAL DIRECTORY

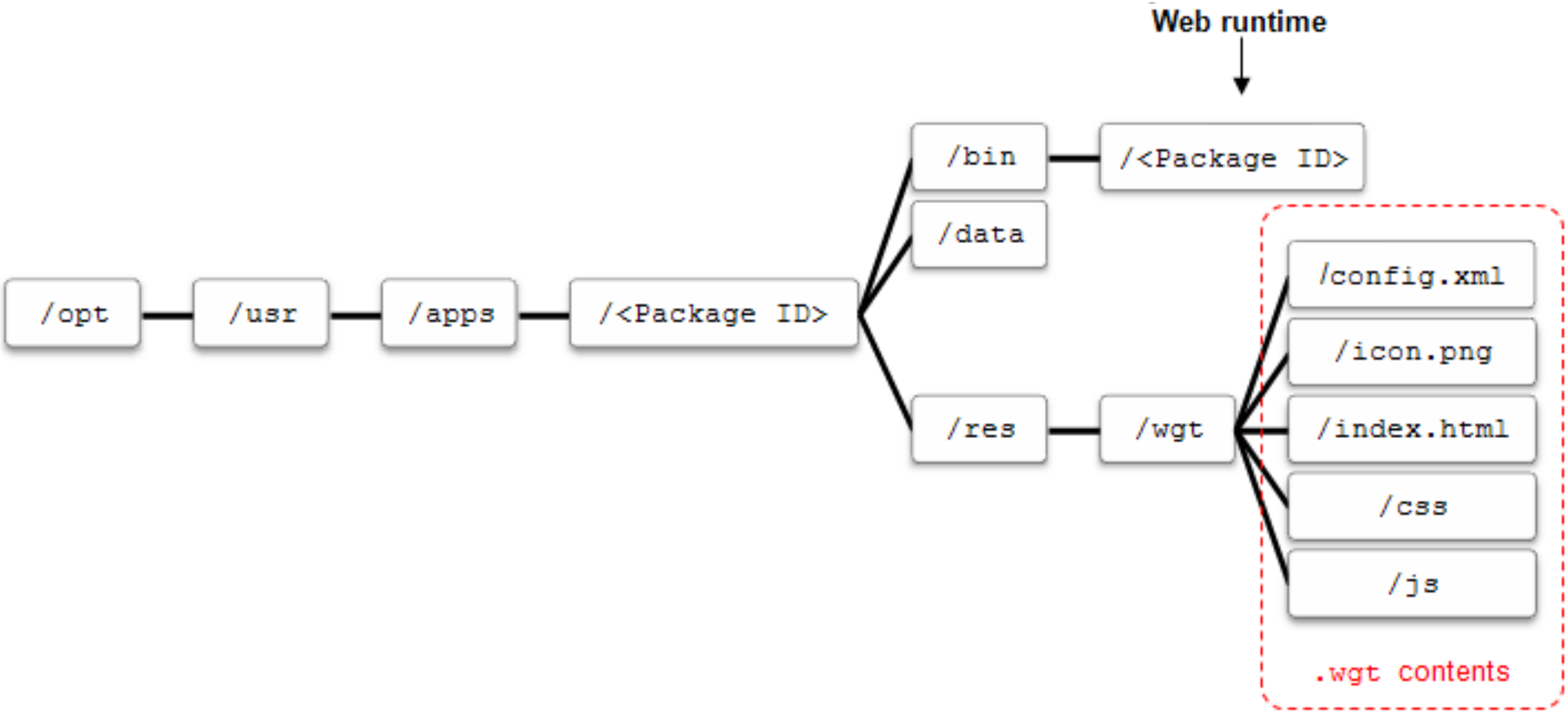


```
sh-4.1$ ls /opt/usr/apps
ls /opt/usr/apps
0pnxz8hbsr  hdufar9ycj            org.tizen.bluetooth-share-ui   sjjevolsjk
42KriKjov3  hyCsE05ySM            org.tizen.bt-syspopup          tlp6xwqzos
57r43275q7  ijudt7w61q            org.tizen.data-provider-slave  tmp
8r4r5ddzzn  jysyv9o1dc            org.tizen.download-manager     tyjHFs6oP5
aospd00043  kLf2Ks0DYk            org.tizen.indicator            vxqbrefica
BLP40IVRLk  kmcele1k0n            org.tizen.menu-screen          xZuDw2OeGg
cp7ipabg4k  kto5jikgul            org.tizen.taskmgr              zktdpemtmw
D7eOJquGtL  livebox.web-provider  ph1vq2phrp                     ZsnYtAdjl2
dhrul6qzj3  logs                  PhYwYqDa1I                     zunqjlsnce
f9uev8hsyo  nI2PLNdTwi            q7097a278m
gi2qxenosh  npwf0scb88            scim
sh-4.1$ $
```

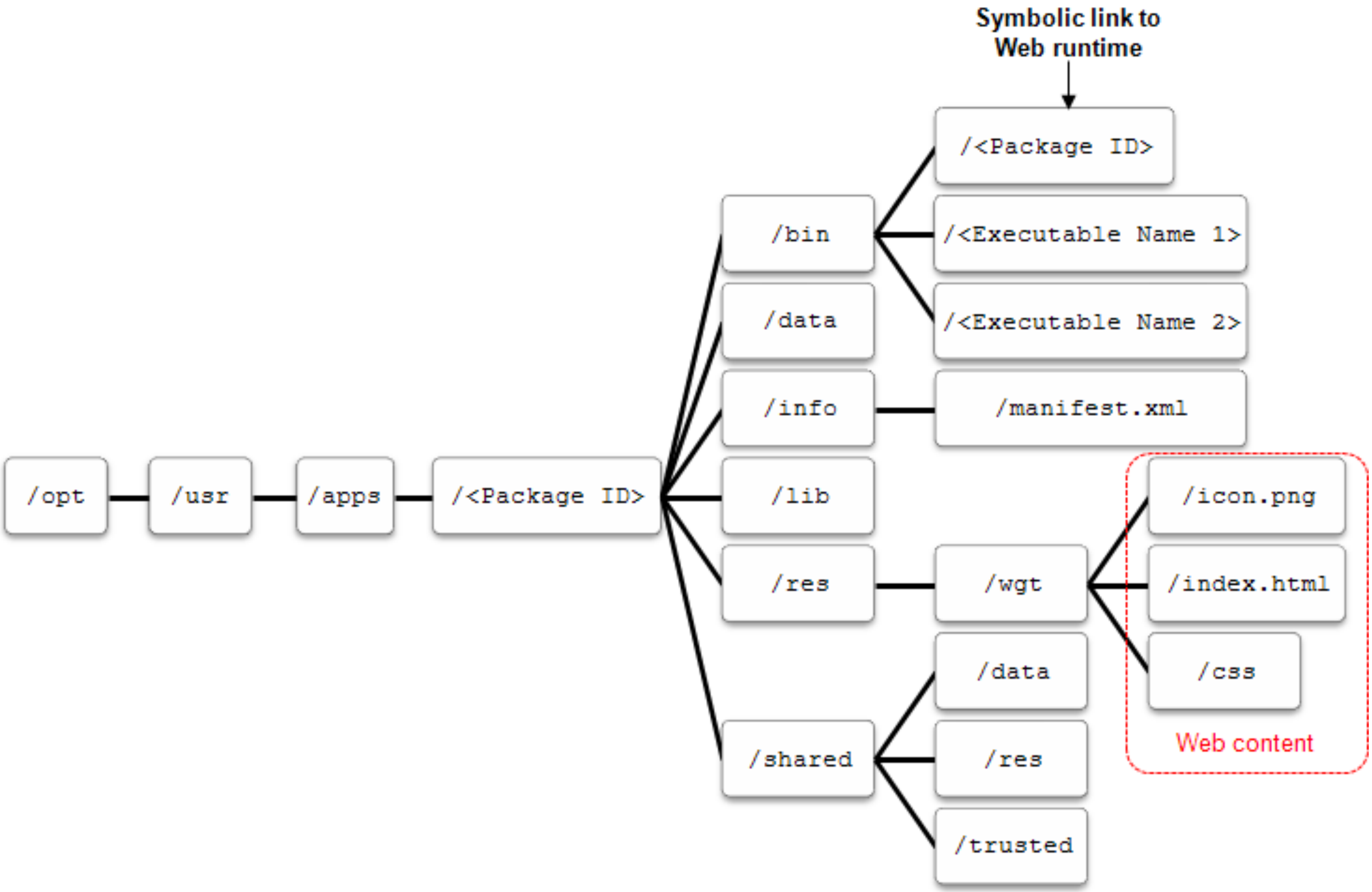
NATIVE APPS (.TPK)



WEB APPS (.WGT)


Web runtime
/opt
/usr
/apps
/<Package ID>
/bin
/<Package ID>
/data
/res
/wgt
/config.xml
/icon.png
/index.html
/css
/js
.wgt contents

HYBRID APP(.TPK)



TIZEN SECURITY MODEL

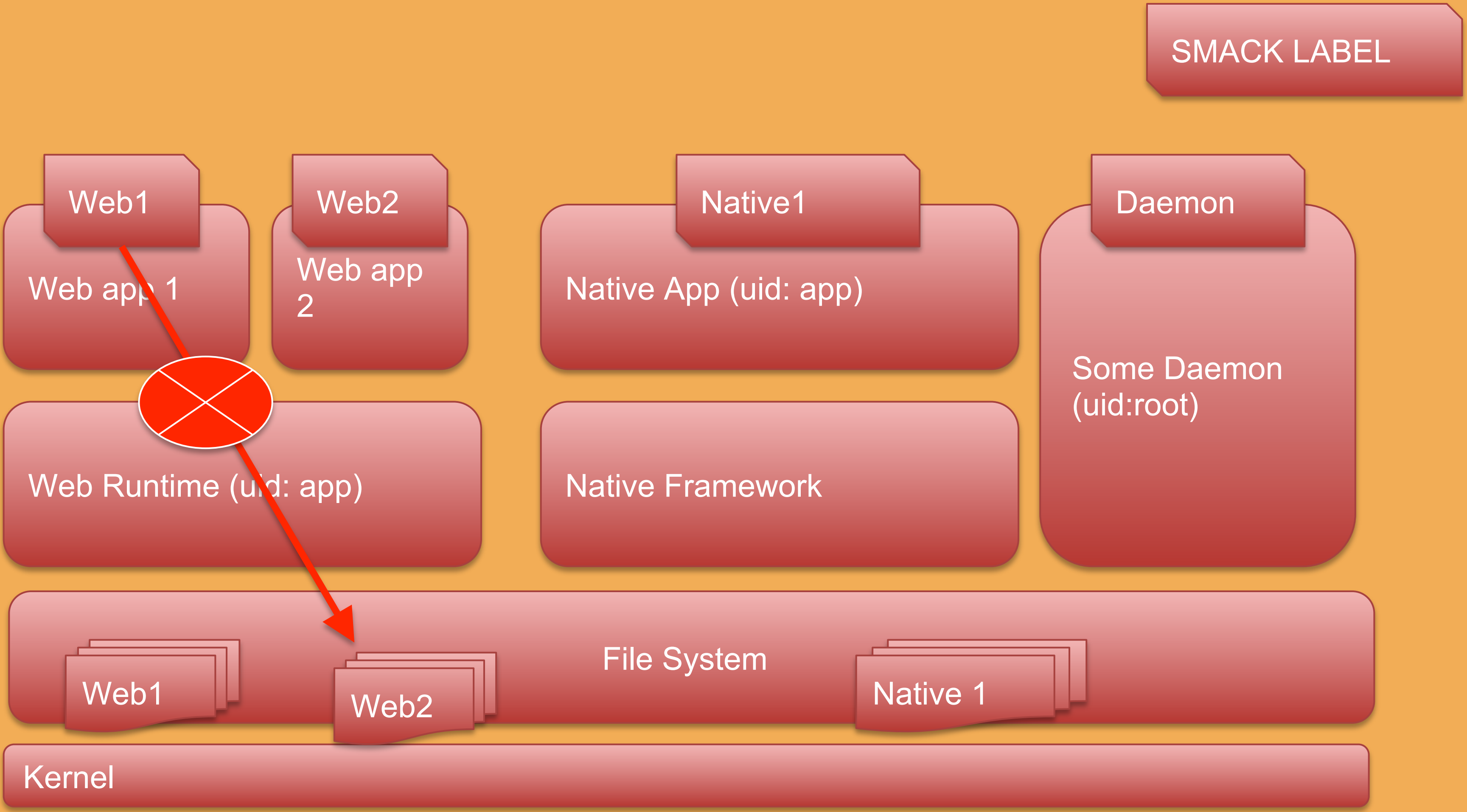
- **Non root applications**
  - All applications run under same non-root user ID, app.
  - Most of the middleware daemons will run as non-root user.
- **Application sandboxing**
  - All applications are sandboxed by SMACK.
  - An application is allowed to read or write files in it's home directory and shared media directory (/opt/usr/media)
  - Each application unable to send IPC and sockets, r/w other application files.
- **Permission Model/Least privilege**
  - All applications will have manifest file describing privileges.
  - Manifest file describes also SMACK labels and rule.
- **Application Signing** – Author and Distributor
- **Tizen CSP for Web Apps** –Web Apps have additional layer of security with Content Security Policy.
- **Encrypt HTML, JS and CSS stored in Device** - Encrypts at Install time and Runtime decryption.
- **Content Security Framework –** Provides API for AVs.

SMACK

SIMPLIFIED MANDATORY ACCESS CONTROL KERNEL

“what's mine is mine; what's yours is yours.”

SMACK allows you to add controlled exception to this basic rule.


SMACK LABEL
Web1
Web app 1
Web2
Web app 2
Native1
Native App (uid: app)
Daemon
Some Daemon (uid:root)
Web Runtime (uid: app)
Native Framework
File System
Web1
Web2
Native 1
Kernel

SMACK TERMS

– **Subject** → Any Running Process (Have Smack Label)

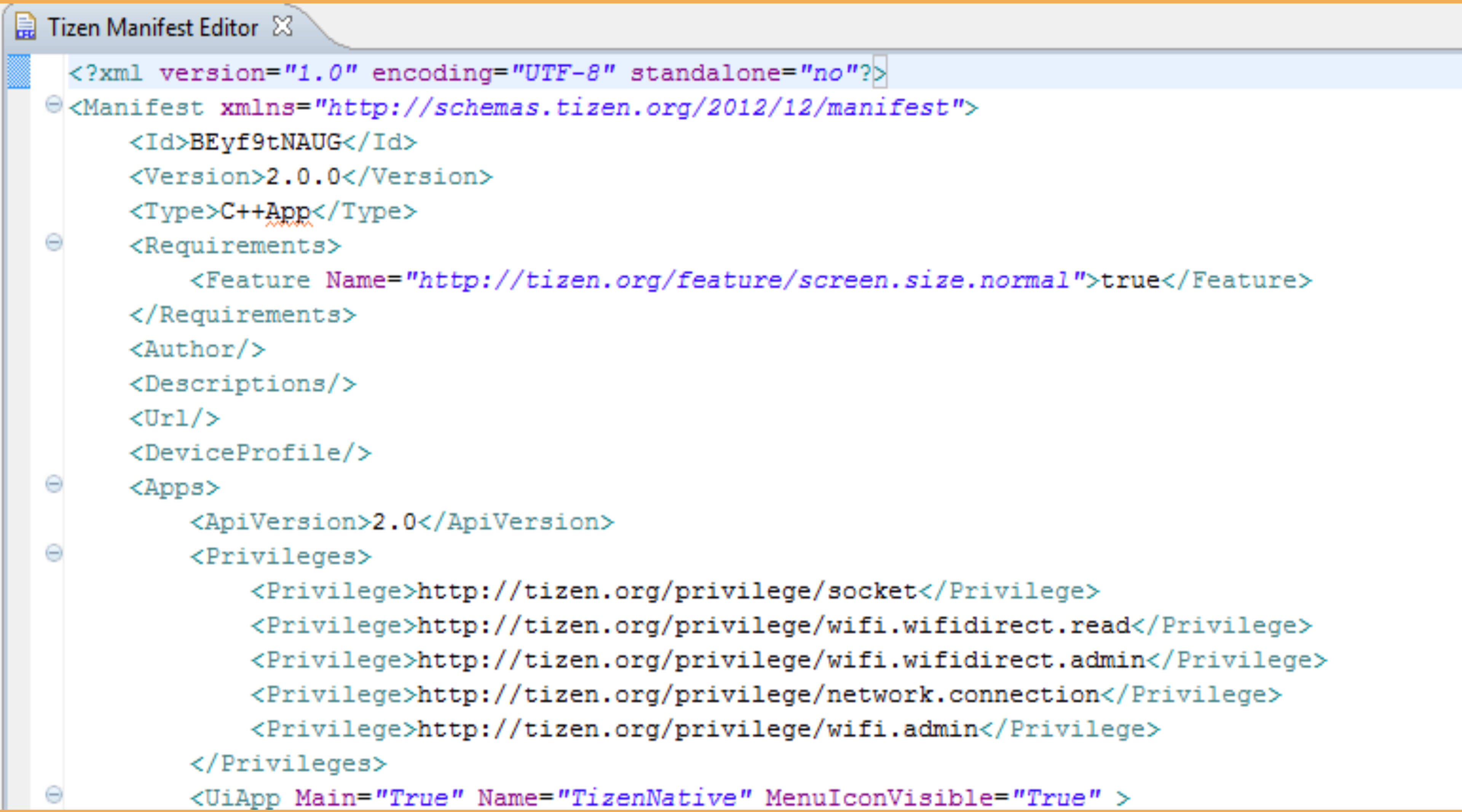
– **Object** → File, IPC, Sockets, Process

– **Access** → Read (r), Write (w), Execute (e), Append (a) , Lock (l), Transmute (t)

41,000 SMACK Rules in Tizen 2.2.1 !!

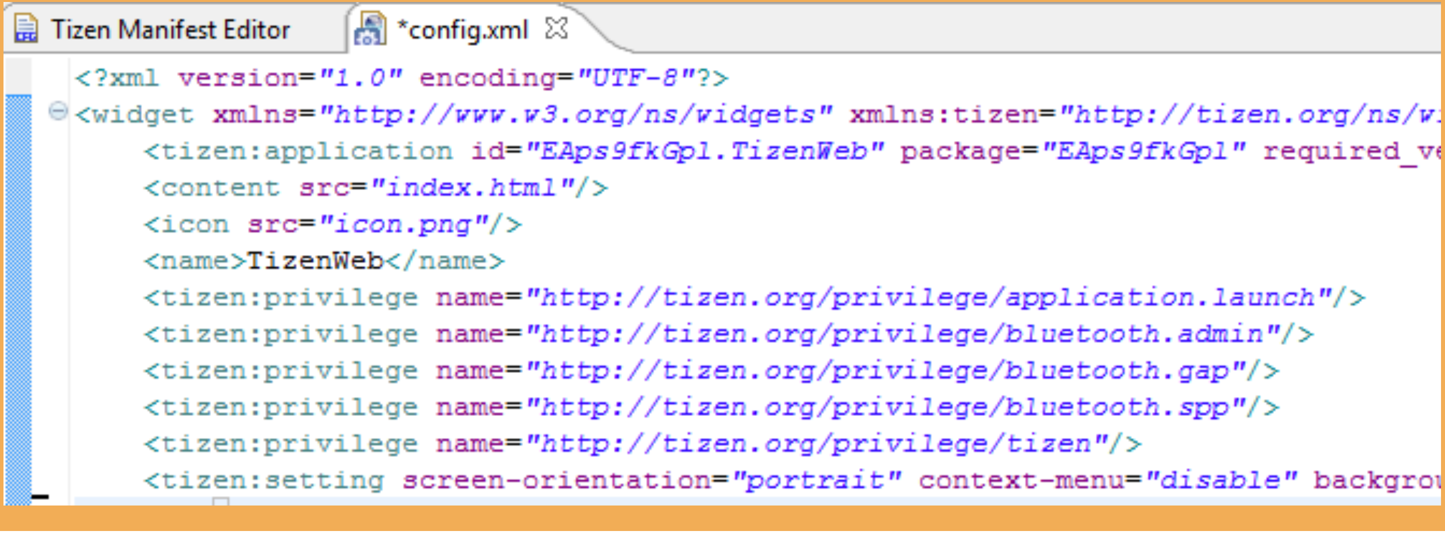
From Tizen 3.X: (Smack Three domain Model, Cynara)

NATIVE APPS – MANIFEST.XML

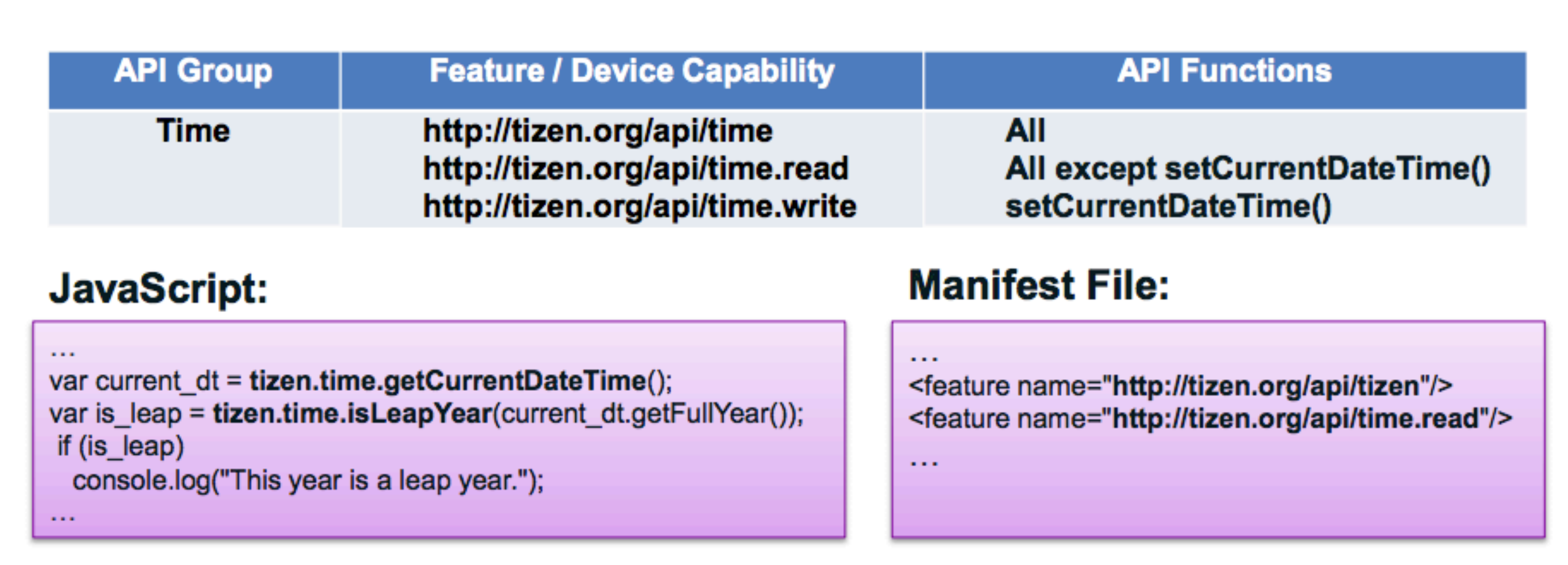


```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Manifest xmlns="http://schemas.tizen.org/2012/12/manifest">
    <Id>BEyf9tNAUG</Id>
    <Version>2.0.0</Version>
    <Type>C++App</Type>
    <Requirements>
        <Feature Name="http://tizen.org/feature/screen.size.normal">true</Feature>
    </Requirements>
    <Author/>
    <Descriptions/>
    <Url/>
    <DeviceProfile/>
    <Apps>
        <ApiVersion>2.0</ApiVersion>
        <Privileges>
            <Privilege>http://tizen.org/privilege/socket</Privilege>
            <Privilege>http://tizen.org/privilege/wifi.wifidirect.read</Privilege>
            <Privilege>http://tizen.org/privilege/wifi.wifidirect.admin</Privilege>
            <Privilege>http://tizen.org/privilege/network.connection</Privilege>
            <Privilege>http://tizen.org/privilege/wifi.admin</Privilege>
        </Privileges>
        <UiApp Main="True" Name="TizenNative" MenuIconVisible="True" >
```

WEB APPS – CONFIG.XML



```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:tizen="http://tizen.org/ns/w
    <tizen:application id="EAps9fkGpl.TizenWeb" package="EAps9fkGpl" required_ve
    <content src="index.html"/>
    <icon src="icon.png"/>
    <name>TizenWeb</name>
    <tizen:privilege name="http://tizen.org/privilege/application.launch"/>
    <tizen:privilege name="http://tizen.org/privilege/bluetooth.admin"/>
    <tizen:privilege name="http://tizen.org/privilege/bluetooth.gap"/>
    <tizen:privilege name="http://tizen.org/privilege/bluetooth.spp"/>
    <tizen:privilege name="http://tizen.org/privilege/tizen"/>
    <tizen:setting screen-orientation="portrait" context-menu="disable" backgrou
```



API Group	Feature / Device Capability	API Functions
Time	http://tizen.org/api/time http://tizen.org/api/time.read http://tizen.org/api/time.write	All All except setCurrentDateTime() setCurrentDateTime()

JavaScript:

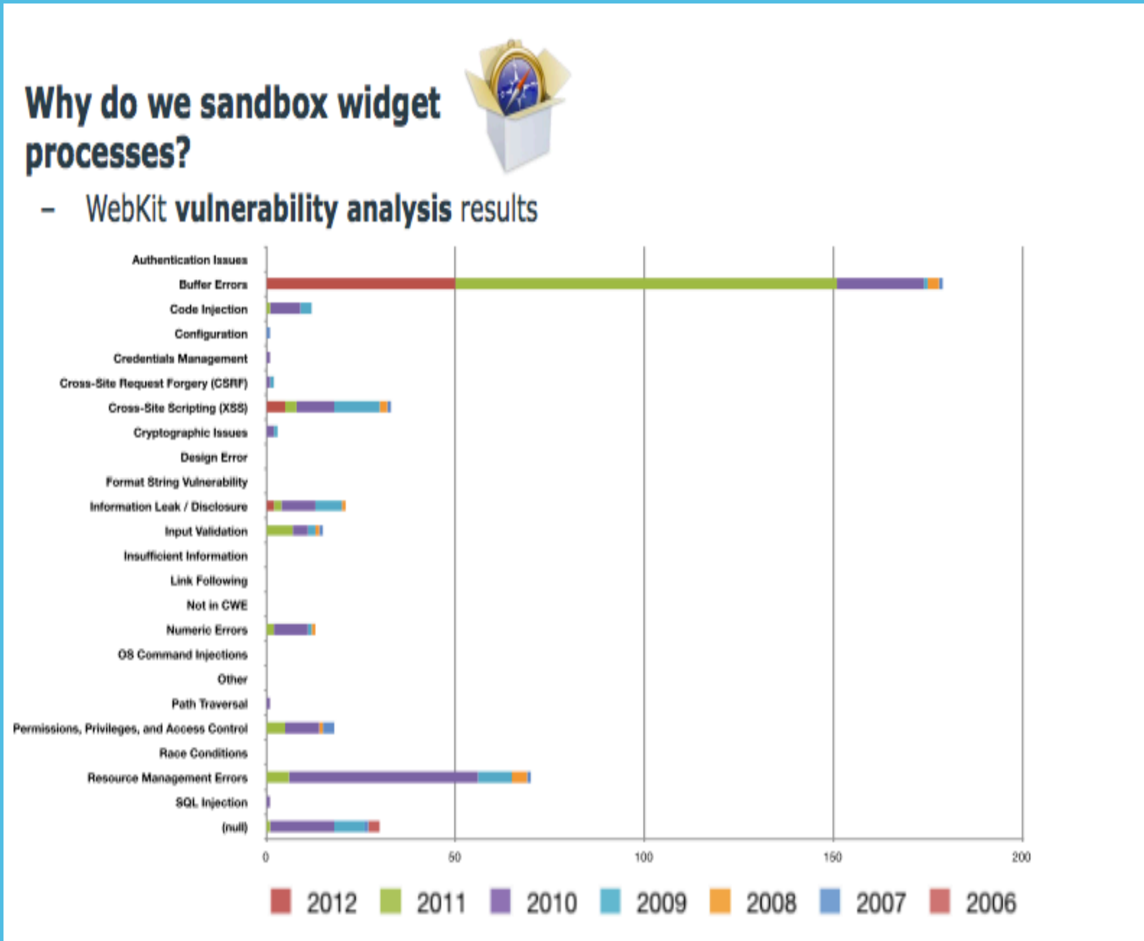
```
…
var current_dt = tizen.time.getCurrentDateTime();
var is_leap = tizen.time.isLeapYear(current_dt.getFullYear());
 if (is_leap)
   console.log("This year is a leap year.");
…
```

Manifest File:

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

WEBKIT2 ON TIZEN

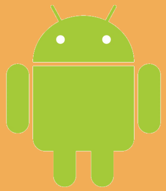
- Tizen WebApps runs on WebKit2
- New API Layer over WebKit
- Supports Split Process Model, Like your Chrome Tabs



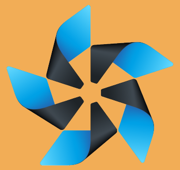
Why do we sandbox widget processes?

– WebKit **vulnerability analysis** results

QUICK COMPARISON



- Apps identified by UID
- Permission : AndroidManifest.xml
- Binder IPC using Intents
- SELinux
- Signed by Developer



- Users identified by UID (app)
- Permission: Manifest.xml & Config.xml
- MessagePort IPC using socket
- SMACK & CSP
- Content Security Framework
- Signed by Developer & Distributor

iOS

- All Apps run under user “mobile”.
- No permission model. Ask for Permission at Runtime.
- URL Schemes, x-callback URL, Extension, XPC based IPC
- Powerbox, Seatbelt
- Signed by Distributor

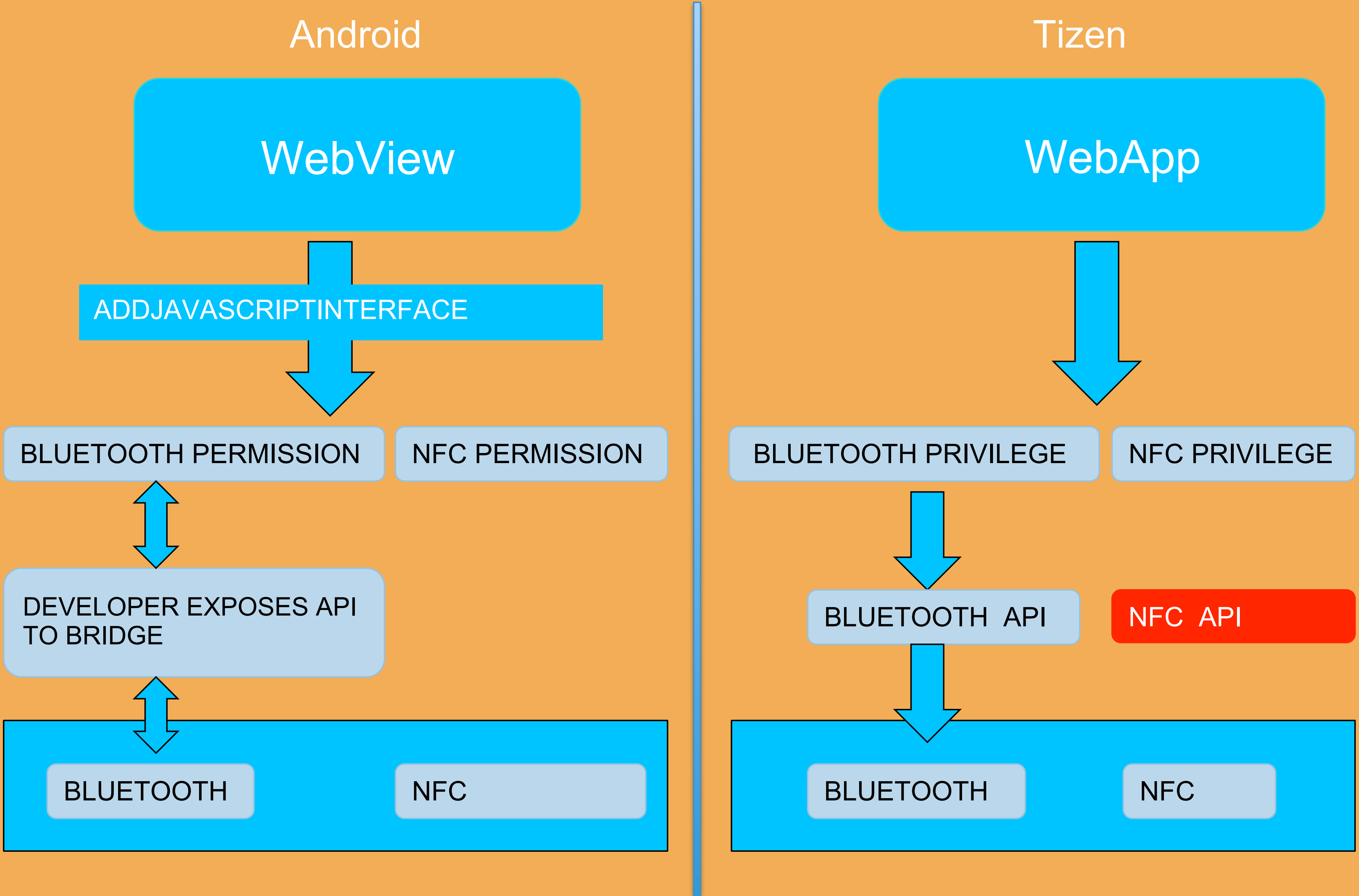
RESEARCH FOCUS

- Tizen 2.2.1 and IVI 3.0
- OS Memory Protection
- Tizen CSP and WebKit

ANDROID WEB APP vs. TIZEN WEB APP

- Tizen Web Apps are powerful and feature rich.
- In Android Web Apps in WebView and can interact with Device features using **addJavascriptInterface.**
- In Tizen, It provides Web API that allows to leverage Device features and are protected using privileges and CSP.

OVER PRIVILEGED ANDROID APP VS TIZEN APP



Android

WebView

ADDJAVASCRIPTINTERFACE

BLUETOOTH PERMISSION

NFC PERMISSION

DEVELOPER EXPOSES API TO BRIDGE

BLUETOOTH

NFC

Tizen

WebApp

BLUETOOTH PRIVILEGE

NFC PRIVILEGE

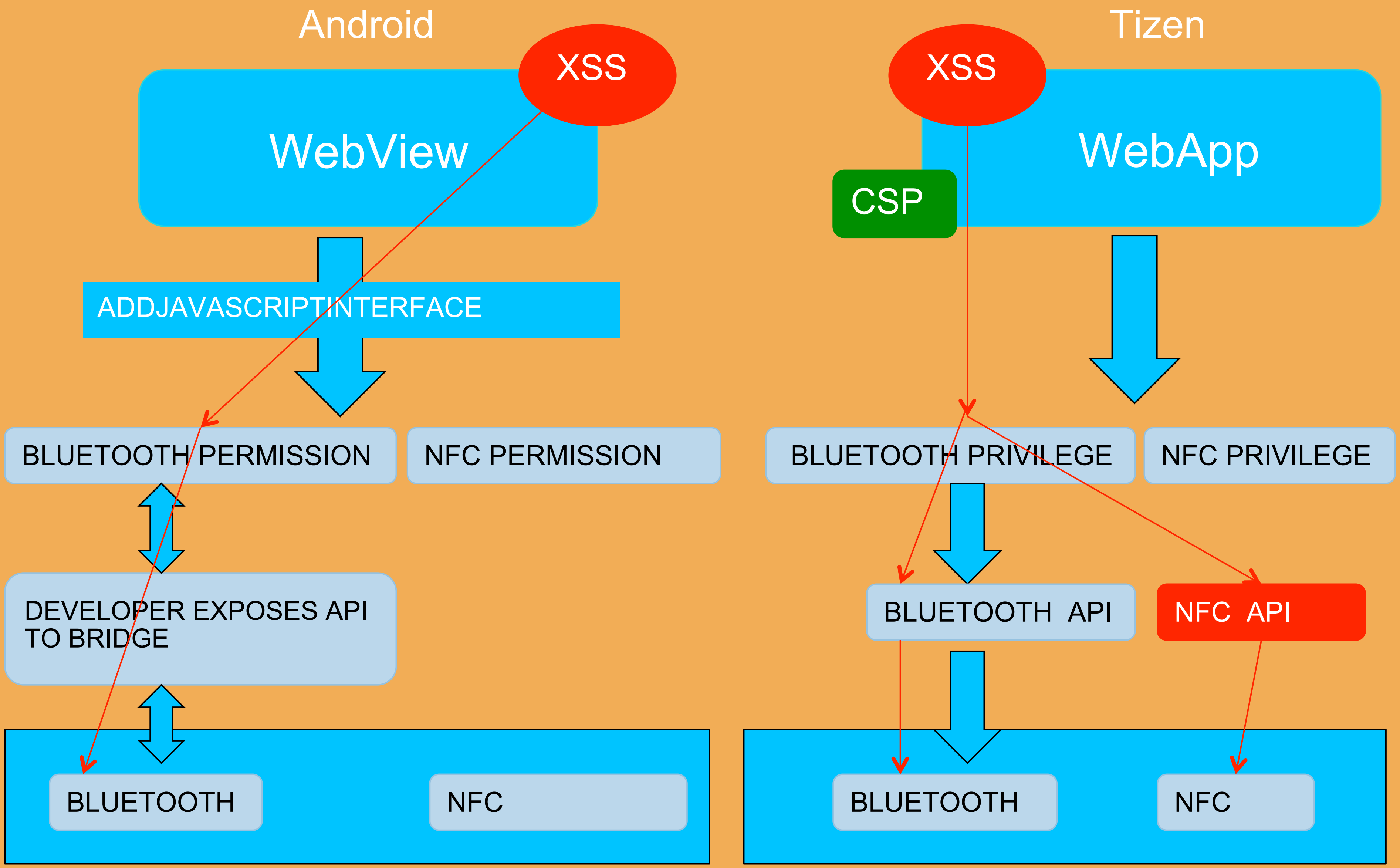
BLUETOOTH API

NFC API

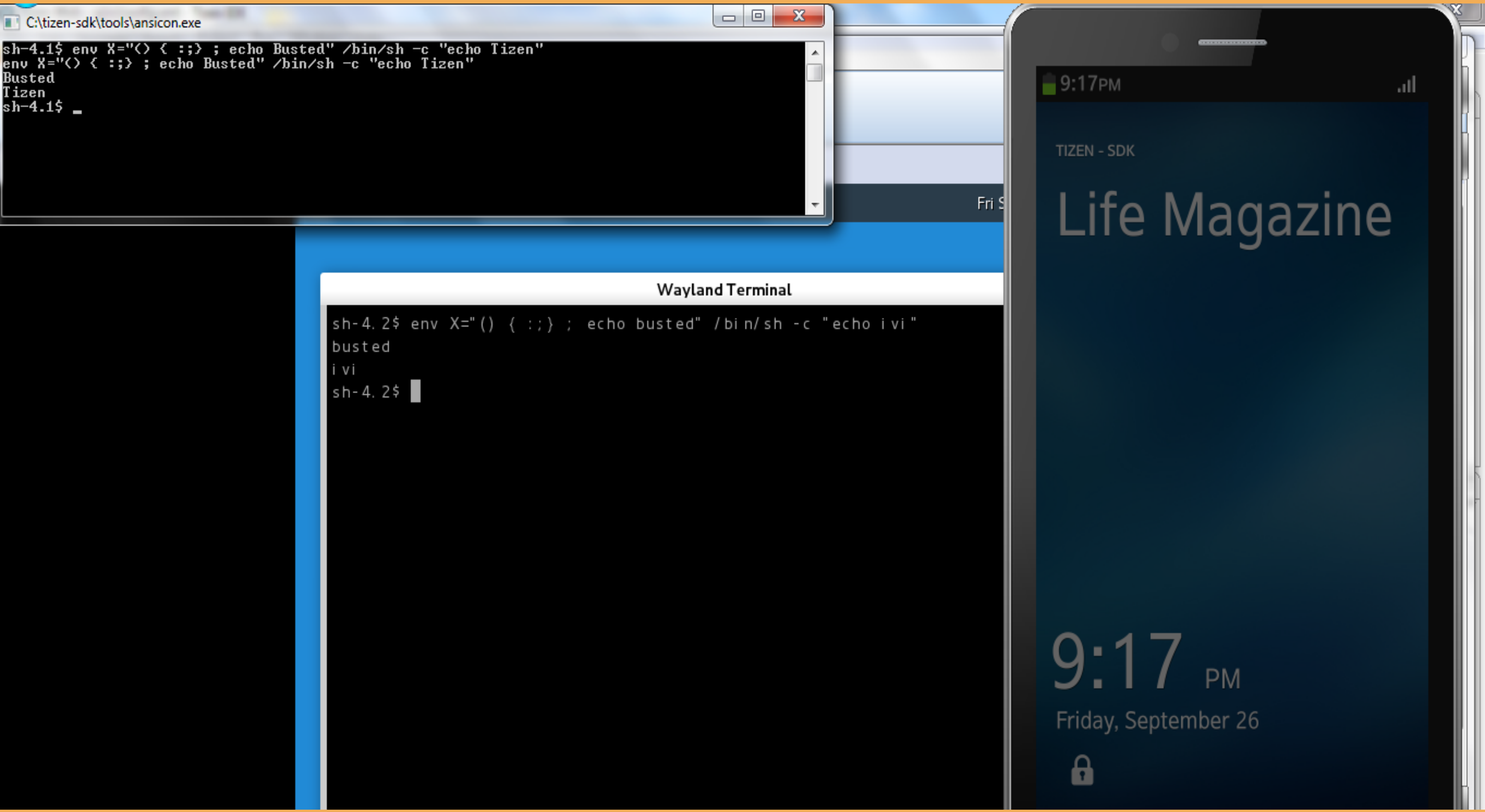
BLUETOOTH

NFC

SCENARIO : XSS



LIKE ANY LINUX DISTRO : SHELLSHOCK



```
sh-4.1$ env X="() { :;} ; echo Busted" /bin/sh -c "echo Tizen"
env X="() { :;} ; echo Busted" /bin/sh -c "echo Tizen"
Busted
Tizen
sh-4.1$ _
```

```
sh-4.2$ env X="() { :;} ; echo busted" /bin/sh -c "echo ivi"
busted
ivi
sh-4.2$
```

DEP

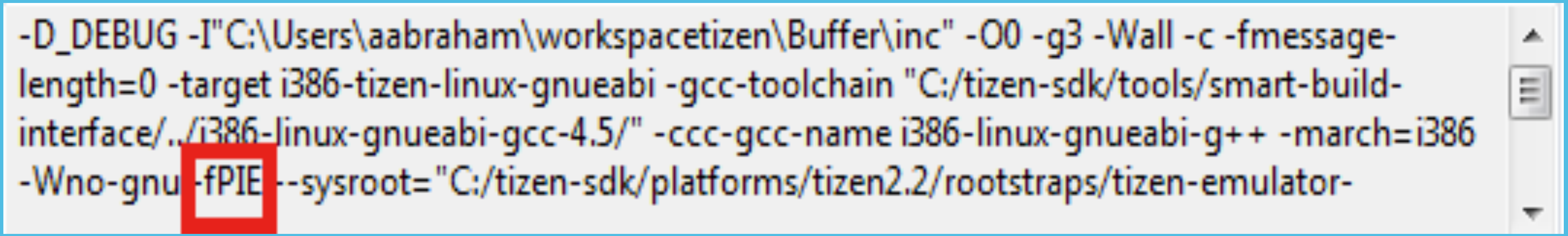
- When Data Execution Prevention is enabled, data on stack should be non-executable.
- Prevents Shellcode at Stack from Executing.
- But DEP is not seen in action.

e

DEMO

ASLR

- As per documentation ASLR is fully implemented in Tizen 2.1 itself.
- Already Broken in Tizen 2.1 , discovered by Shuichiro Suzuki
- **/proc/sys/kernel/randomize_va_space** is set to **2** which tell us that ASLR is enabled.
- The personality value at **/proc/self/personality** is set to **00040000**. which corresponds to (ADDR_NO_RANDOMIZE) disables ASLR.
- InTizen 2.2, **/proc/self/personality** is set to **00000000**



```
-D_DEBUG -I"C:\Users\aabraham\workspacetizen\Buffer\inc" -O0 -g3 -Wall -c -fmessage-
length=0 -target i386-tizen-linux-gnueabi -gcc-toolchain "C:/tizen-sdk/tools/smart-build-
interface/../i386-linux-gnueabi-gcc-4.5/" -ccc-gcc-name i386-linux-gnueabi-g++ -march=i386
-Wno-gnu -fPIE --sysroot="C:/tizen-sdk/platforms/tizen2.2/rootstraps/tizen-emulator-
```

- PIE (position-independent executable). So this will make the native application ASLR enabled.
- But due to implementation issues, it was still found that ASLR is still in broken state.
- **/proc/<pid>/maps** –Address of heap, stack and main modules remain the same.

e

DEMO

URL SPOOFING/CONTENT INJECTION

- Open a new window with URL https://facebook.com and assign it to a variable w.
- Try to write “<h1>You 've been Hacked</h1>” to DOM using w.document.write()
- Focus the window.

e

DEMO

CSP BYPASS

Content-Security-Policy: default-src 'self'; script-src 'self'

- We create a script tag with JavaScript nullbyte prepended to a SCRIPT URL.
- Tricks the browser and load the Script from a different domain and Bypass CSP.

e

DEMO

PENTESTING METHDOLOGIES

Whitebox

Access to Source and Knowledge about the application

Blackbox

No access to Source and no idea about the application

Further Classification

- Static Analysis
- Dynamic Analysis
- Network Analysis

STATIC ANALYSIS

- **Certificate Signature Analysis** – Developer and Distributor
- **Manifest Analysis** – manifest.xml/config.xml
  - * Unwanted Privileges.
  - * CSP is proper or not.
  - * Smack Labels and Rules

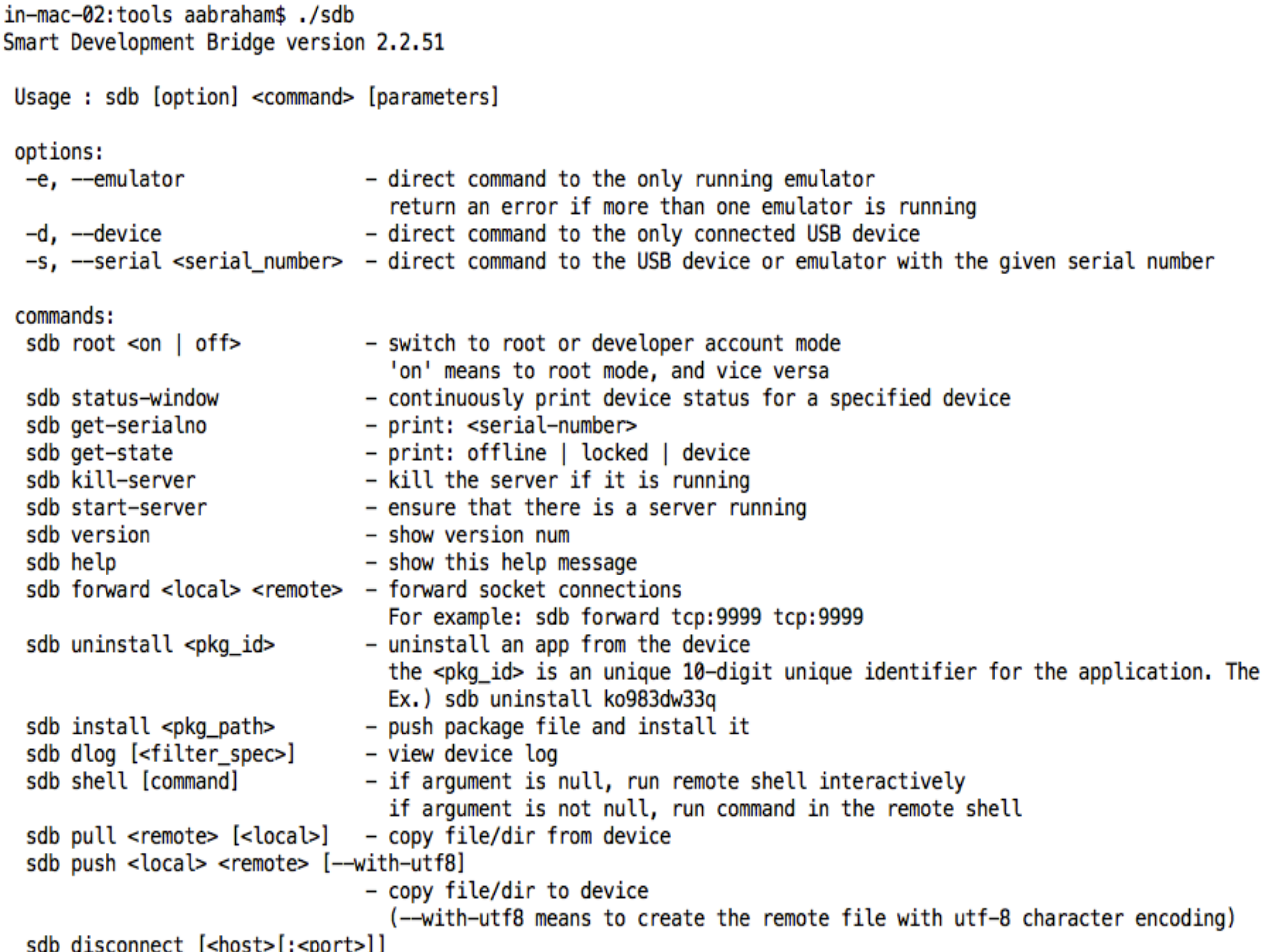
- **Decompile Native App**
  - * Apps Compiled with CLANG/CLANG++ compiler.
  - * LLVM decompiler - tizen_tpk_decompiler.py (make use of Retdec API).

- **Code Review**
  - * Weak Encryption, Crypto, Plaintext Information, SSL Overriding, Insecure File Storage, Client Side SQLi/XSS.
  - * Pretty much OWASP Mobile Top 10.

- **Couple of tools** - https://github.com/ajinabraham/tizen-security

DYNAMIC ANALYSIS

- Enable Developer Mode - ***#84936#**
- Run the App in Device/Tizen VM or Web Simulator.
- Sensitive data shared during IPC, Sensitive files written at Runtime, Temp files etc.
- Directories/ Files/DB with chmod 777 access.
- Tools: Dynamic Analyzer much like android ddms/Android Device Monitor, sdb – The adb equivalent for Tizen.

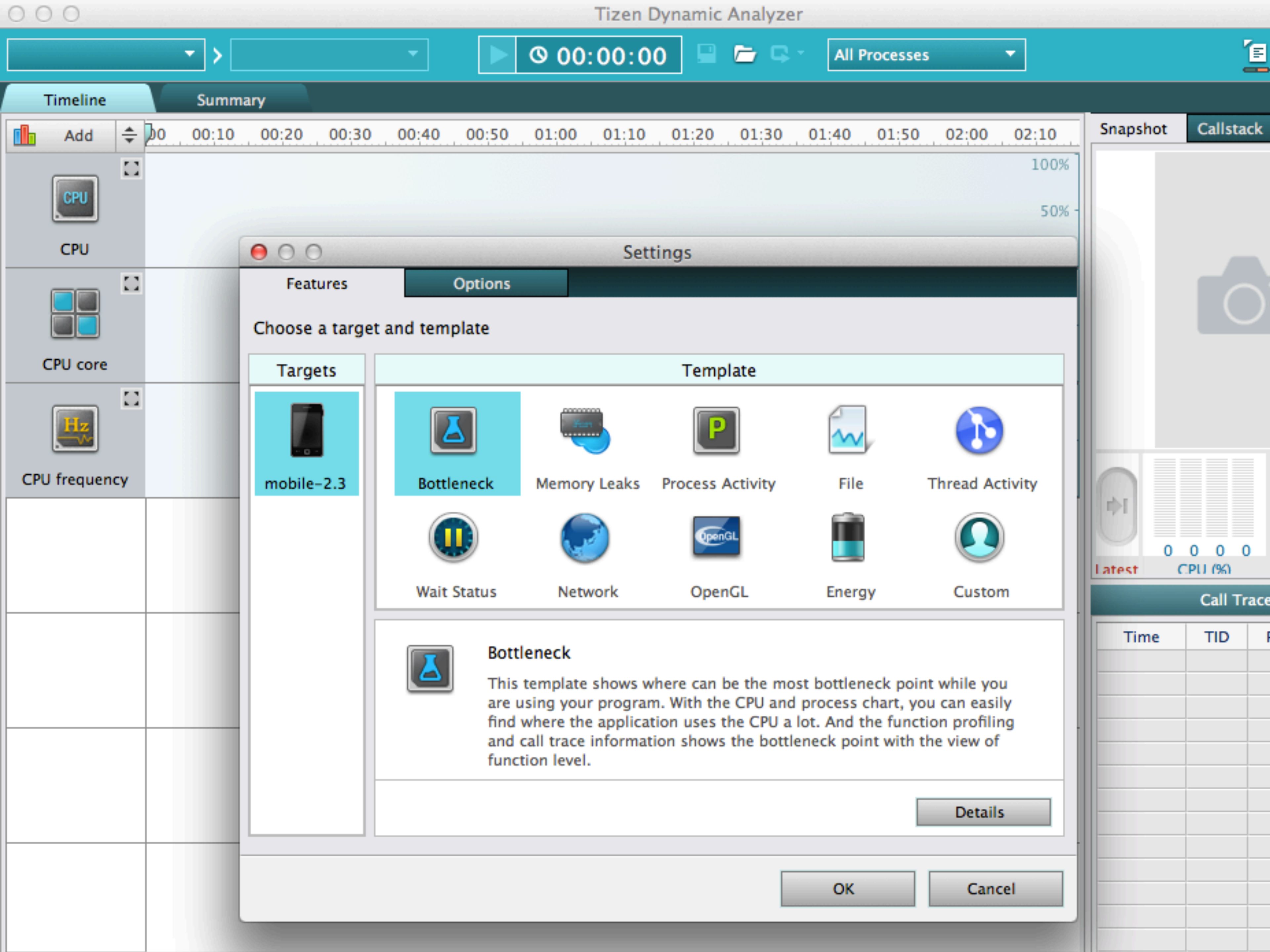


```
in-mac-02:tools aabraham$ ./sdb
Smart Development Bridge version 2.2.51

 Usage : sdb [option] <command> [parameters]

 options:
  -e, --emulator                - direct command to the only running emulator
                                  return an error if more than one emulator is running
  -d, --device                  - direct command to the only connected USB device
  -s, --serial <serial_number>  - direct command to the USB device or emulator with the given serial number

 commands:
  sdb root <on | off>           - switch to root or developer account mode
                                  'on' means to root mode, and vice versa
  sdb status-window             - continuously print device status for a specified device
  sdb get-serialno              - print: <serial-number>
  sdb get-state                 - print: offline | locked | device
  sdb kill-server               - kill the server if it is running
  sdb start-server              - ensure that there is a server running
  sdb version                   - show version num
  sdb help                      - show this help message
  sdb forward <local> <remote>  - forward socket connections
                                  For example: sdb forward tcp:9999 tcp:9999
  sdb uninstall <pkg_id>        - uninstall an app from the device
                                  the <pkg_id> is an unique 10-digit unique identifier for the application. The
                                  Ex.) sdb uninstall ko983dw33q
  sdb install <pkg_path>        - push package file and install it
  sdb dlog [<filter_spec>]      - view device log
  sdb shell [command]           - if argument is null, run remote shell interactively
                                  if argument is not null, run command in the remote shell
  sdb pull <remote> [<local>]   - copy file/dir from device
  sdb push <local> <remote> [--with-utf8]
                                - copy file/dir to device
                                  (--with-utf8 means to create the remote file with utf-8 character encoding)
  sdb disconnect [<host>[:<port>]]
```



Choose a target and template

Bottleneck

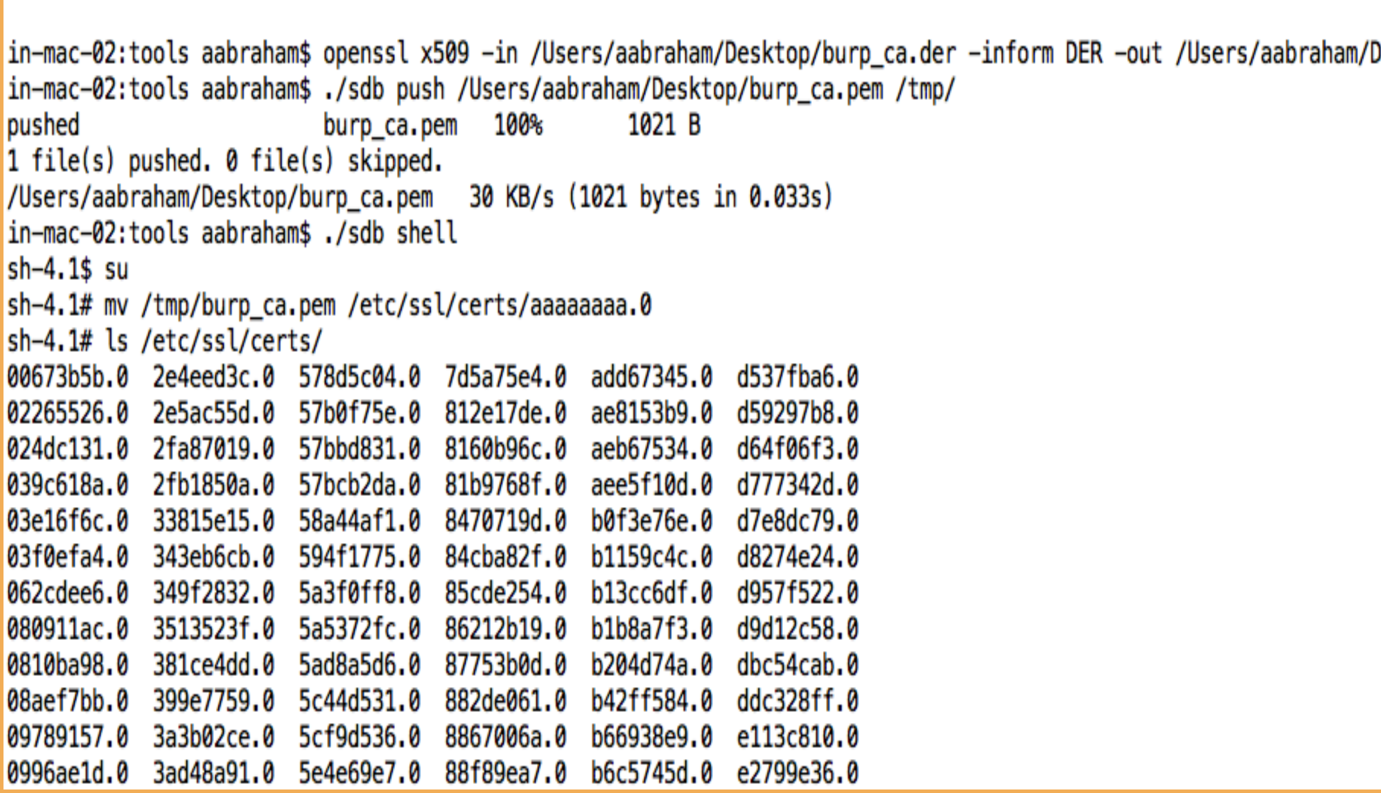
This template shows where can be the most bottleneck point while you are using your program. With the CPU and process chart, you can easily find where the application uses the CPU a lot. And the function profiling and call trace information shows the bottleneck point with the view of function level.

NETWORK ANALYSIS

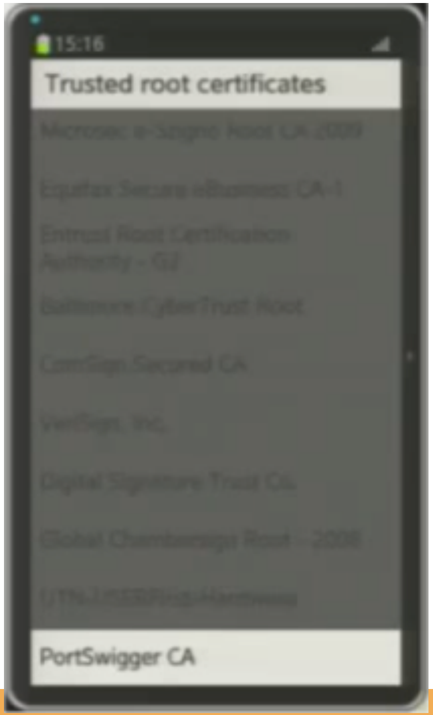
- Installing SSL Certificate and HTTPS Traffic Decryption with a Proxy like Burp/ Fiddler.
- Install Certificate to User Certificate Store: *Settings -> About device -> Manage certificates -> User certificates -> Install*.
- OWASP Top 10 Web Risks

INSTALLING CA CERT TO TRUSTED CERT STORE

- Installing CA in Device
- Trusted CA Certificates are stored under **/etc/ssl/certs**
- Filename: <**8HEXChars.0**> in PEM format.
- Copy the CA certificate to /etc/ssl/certs and it’s trusted.



```
in-mac-02:tools aabraham$ openssl x509 -in /Users/aabraham/Desktop/burp_ca.der -inform DER -out /Users/aabraham/Desktop/burp_ca.pem -outform PEM
in-mac-02:tools aabraham$ ./sdb push /Users/aabraham/Desktop/burp_ca.pem /tmp/
pushed                    burp_ca.pem   100%      1021 B
1 file(s) pushed. 0 file(s) skipped.
/Users/aabraham/Desktop/burp_ca.pem   30 KB/s (1021 bytes in 0.033s)
in-mac-02:tools aabraham$ ./sdb shell
sh-4.1$ su
sh-4.1# mv /tmp/burp_ca.pem /etc/ssl/certs/aaaaaaaa.0
sh-4.1# ls /etc/ssl/certs/
00673b5b.0  2e4eed3c.0  578d5c04.0  7d5a75e4.0  add67345.0  d537fba6.0
02265526.0  2e5ac55d.0  57b0f75e.0  812e17de.0  ae8153b9.0  d59297b8.0
024dc131.0  2fa87019.0  57bbd831.0  8160b96c.0  aeb67534.0  d64f06f3.0
039c618a.0  2fb1850a.0  57bcb2da.0  81b9768f.0  aee5f10d.0  d777342d.0
03e16f6c.0  33815e15.0  58a44af1.0  8470719d.0  b0f3e76e.0  d7e8dc79.0
03f0efa4.0  343eb6cb.0  594f1775.0  84cba82f.0  b1159c4c.0  d8274e24.0
062cdee6.0  349f2832.0  5a3f0ff8.0  85cde254.0  b13cc6df.0  d957f522.0
080911ac.0  3513523f.0  5a5372fc.0  86212b19.0  b1b8a7f3.0  d9d12c58.0
0810ba98.0  381ce4dd.0  5ad8a5d6.0  87753b0d.0  b204d74a.0  dbc54cab.0
08aef7bb.0  399e7759.0  5c44d531.0  882de061.0  b42ff584.0  ddc328ff.0
09789157.0  3a3b02ce.0  5cf9d536.0  8867006a.0  b66938e9.0  e113c810.0
0996ae1d.0  3ad48a91.0  5e4e69e7.0  88f89ea7.0  b6c5745d.0  e2799e36.0
```



Trusted root certificates

Microsec e-Szigno Root CA 2009

Equifax Secure eBusiness CA-1

Entrust Root Certification Authority - G2

Baltimore CyberTrust Root

ComSign Secured CA

VeriSign, Inc.

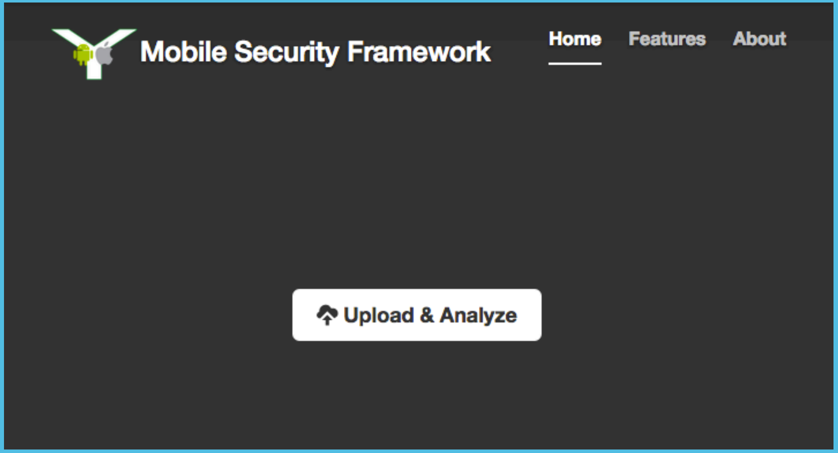
Digital Signature Trust Co.

Global Chambersign Root - 2008

UTN-USERFirst-Hardware

PortSwigger CA

MOBILE SECURITY FRAMEWORK



- Automated Mobile Application Pentest and Code Review Framework.
- Currently Supports Android and iOS.
- Tizen support is on the way.

- Download: https://github.com/ajinabraham/YSO-Mobile-Security-Framework/

SECURITY CONCERNS

- WebKit = Bugs!!

- *“WebKit is basically a collection of use-after-frees that somehow manages to render HTML (probably via a buffer overflow in WebGL)“* -the grugq

- HTML Web APIs are powerful, Improper CSP and XSS=owned !!

- Too much SMACK Rules – High chance that developers will mess up. Will be reduced from Tizen 3.

CONCLUSION

- Security Model/Architecture wise they put lot of effort compared to Android or other Operating Systems.

- They made it so complex (SMACK rules) that people can easily mess up.

- Looks promising if they can fix some silly implementation bugs.

THANKS

- Thanks to Yodlee and my awesome manager, Sachin for all the support and encouragement.
- Presentation template by SlidesCarnival & Unsplash

QUESTIONS?

Ajin Abraham

@ajinabraham

http://opensecurty.in