Tizen Web Application Checker

Xu Zhang
(xu.u.zhang@intel.com)

Agenda

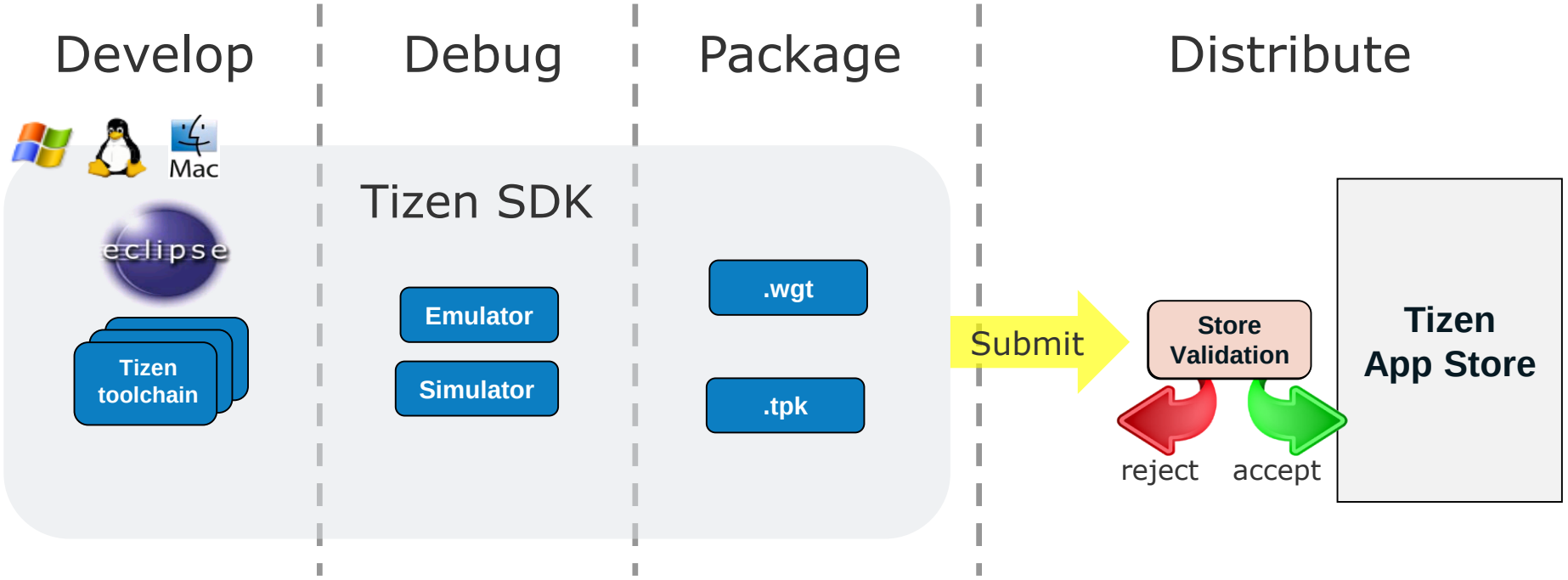
- Tizen Compliance and Application Compatibility
- Tool Features
- Design Overview
- Checker Modules
- Getting Started
- Contributing to Tizen Web Application Checker

Tizen Compliance and Application Compatibility

- Tizen Compliance Goal: ensure Tizen device implementations and applications work together.
- Tizen Compliance Specification (TCS) is a set of requirements for devices and applications to ensure that they work together smoothly.

  TCS can be found here: https://source.tizen.org/compliance
- Tizen Application Requirements:
  - API USE
  - Application Packaging
  - Security

Why We Need Tizen Web Application Checker

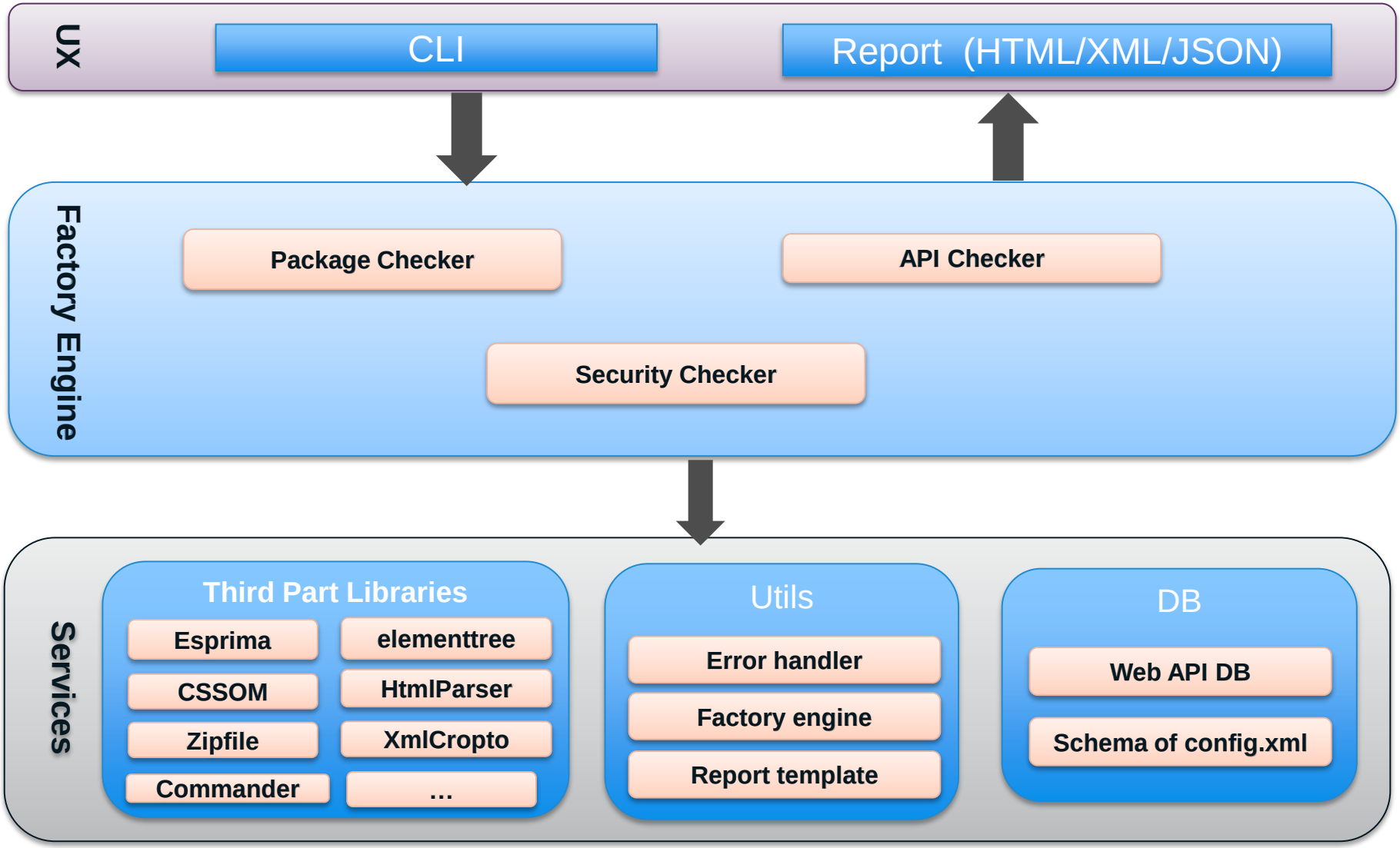


- Developers need to know how to create compatible applications
- Developers need a tool to provide guidance during development of Tizen application.
- App Store needs a tool to provide to assist in screening submitted Tizen applications before acceptance

Tool Features

- Checks Tizen web application using rules defined in TCS
- Identifies incompatible behavior and provides a clear report
- Runs on Linux/Windows/Mac environment
- Implemented with Node.js
- Difference from other JS analyzing tool
  - Checks JavaScript syntax as well as Tizen package and security risks
  - No JS code style checking as JSLint
  - Does not stop at the first issue like the closure compiler. Reports all possible issues
  - Checks Tizen Web APIs usage

Design Overview



Checker Modules – Packaging

- **Ensure package format is compliant with Tizen standards (W3C Widget)**
  - Validate the given package is a valid WGT package

TestName	Status	Reason
zip validity Checking whether the file is a complete zip archive.	FAILED	Error The given package is not complete.

  - Validate the content of a given WGT package

TestName	Status	Reason
package elements attributes Checking whether attributes of elements in config.xml are valid.	FAILED	Error About the attribute "name" of the element "tizen:privilege", its value "http://tizen.org/privilege/calandar.read" is invalid. in file config.xml, line 12 Evidence: <tizen:privilege name="http://tizen.org/privilege/calandar.read"/>

Checker Modules – Security

- Ensure no violation in widget access request policy, API privilege use, and security signature

TestName	Status	Reason
HTML resource validation Checking whether html resources are valid.	FAILED	Error 'index.html' requesting access to 'http://google.com' is not allowed: no origin area matches. in file index.html, line 22, column 32
CSS resource validation Checking whether CSS resources are valid.	PENDING	Warning CSS file 'css/style.css' requires access to '../images/noexist.png', but it cannot be found in the package.
authorized use of API Checking whether the APIs used exceed the privilege level.	FAILED	Error Current privilege level is 'public', but 'http://tizen.org/privilege/appmanager.kill' need 'partner' privilege level. Warning Privilege 'http://tizen.org/privilege/calandar.read' is declared in config.xml, but app neen't 'http://tizen.org/privilege/calandar.read'
integrity of package files Checking whether all signed files are included in the package and whether all files in the package are signed by the author.	FAILED	Error The author has signed file 'invalid.html', but it can't be found in the package. Error File 'app/app.js' is included in the package, but the author hasn't signed it.
author signature Checking whether hash value of reference and signature value in signature are valid.	FAILED	Error Hash error happens on 'config.xml'.

Checker Modules – API Validation

- Ensure all APIs used are supported on Tizen compliant systems

TestName	Status	Reason
access of used api Checking the API used exists and is accessible.	FAILED	Error API 'tizen.bluetooth.getDefaultAdapter' is not allowed to be used, according to the 'tizen:privilege' list in the config.xml file. in file js/main.js, line 476, column 18 Evidence: `this.adapter = tizen.bluetooth.getDefaultAdapter();`
usage of api Checking if the API is used correctly	PENDING	Warning Type error on symbol 'tizen.alarm.PERIOD_HOUR', maybe used attribute as a method or called abstract class, etc. in file js/main.js, line 518 Evidence: `new tizen.alarm.PERIOD_HOUR();`
build-in namespace Checking whether the default built-in namespace has been changed.	PENDING	Warning The existing API 'tizen.alarm' has been overwritten. in file js/getMessage.js, line 202, column 8 Evidence: `tizen.alarm = {};`

Checker Modules – API Validation

Known Issues

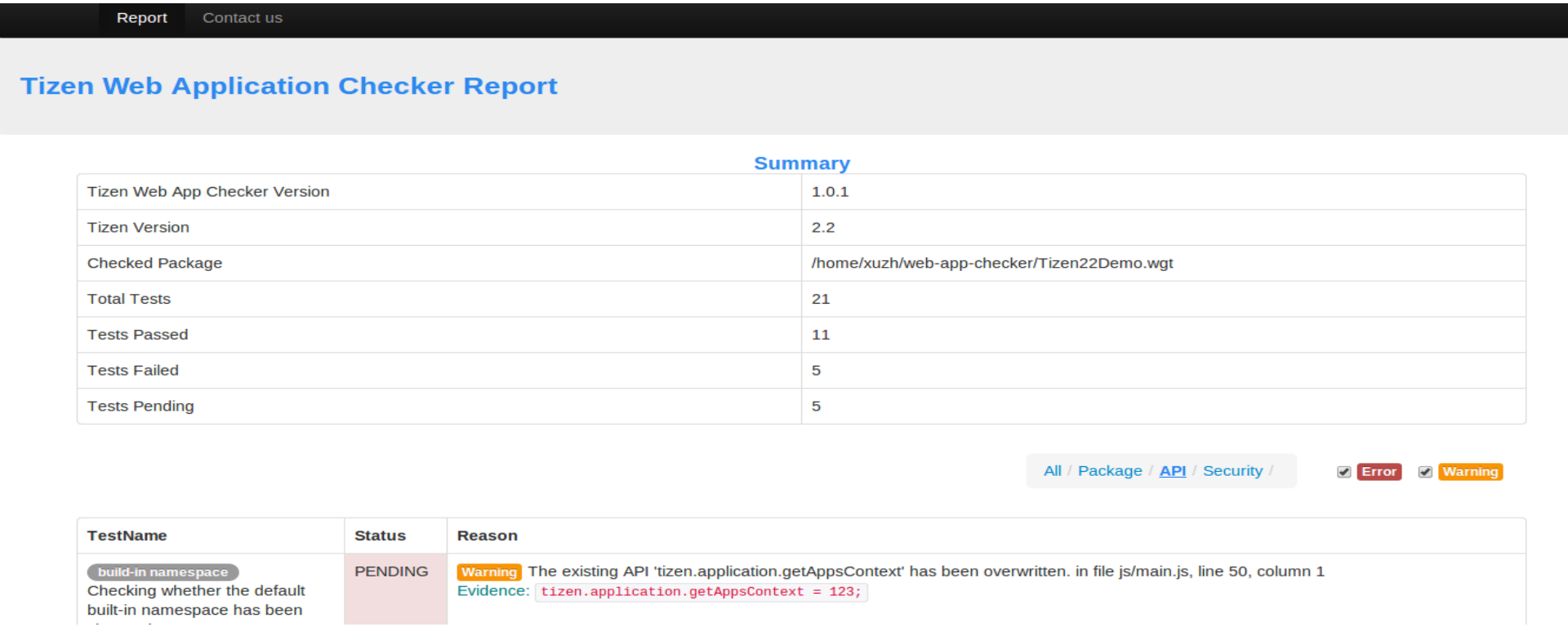
```
// no define of "test" before
   if (condition) {
       var  test = "hello.World";
       var  person = function () {...};
   } else {
       test.split(".");
       person = "a string";
   }
   test.indexOf("o");
   p = new person;
```

```
"<div id="footer"/>"        // in HTML
   footer.setAttribute();        //in JavaScript code
```

```
document.getElementById('touchable').addEventListener('touchend', function(ev){
       document.write('Remaining on element: '  + ev.targetTouches.length);
       // other usage of argument ev
       ....
   }, false);
```

Checker Modules – Reporting

- Three types of reports: HTML, XML, and JSON. HTML format is recommended.
- XML and JSON data can be used by other tools to generate your preferred report. Useful for automation



Checker Modules – Reporting

Report Description

Category	Description
Test Names	Lists all checked items with their names and descriptions
Status	• PASS: No problems are found for the corresponding items. • FAIL: Problems were found by the checker. • PENDING: There are warnings. The issue is suspicious but may not be an error. The Tizen Web App Checker cannot make a definitive decision and suggests the user manually double check it.
Reason	List all issues checked. Each item is a link leading to the details of the test run and results.

Checker Modules – Reporting

Report Filtering

Option	Report Filter
Error	Only show problems marked “Error”
Warning	Only show problems marked “Warning”. The developer should check each of these to verify if it is an error.
All	Show all items in all modules (Package, API, and Security)
Package	Only show items in the Package module
API	Only show items in the API module
Security	Only show items in the Security module

Getting Started

Step 1: Set Up Environment

1. Download Python and install it. Set python in your system variables (Windows Only)
2. Download Node.js from http://nodejs.org/download and install. Set Node in your environment variables.
3. Test to make sure you can use the Node.js and Python.

Step 2: Download Tizen Web Application Checker

- Download from https://source.tizen.org/compliance

Getting Started (Cont.)

Step 3: Install on your development PC

- Extract the package to your working directory
- Make sure Node.js and Python have been set in your system variables.

Step 4: Run Your Application

- Open a terminal and run command below

```
$ node ./bin/tizen-web-app-checker.js -a Tizen22Demo.wgt
```

- More helpful usage can be found by running

```
$ node ./bin/tizen-web-app-checker.js --help
```

- More options are provided as table

Command-line Options

Options	Description
h, --help	Outputs usage information
-V, --version	Outputs the version information
-l, --log <log>	Specifies the log file path
-t, --type <HTML|XML|JSON>	Specifies the report type
-r, --report <report>	Specifies report file path
-v, --verbose	shows verbose information in the console
-s, --spec <2.1, 2.2>	Specifies the Tizen API version used to check compliance. Tizen 2.2 is used by default.
-a --apiCheck	Enables the API checker. This is an experimental feature
-p, --privLevel <public | partner | platform>	Specifies to simulate the distributor granting the level

Getting Started (Cont.)

Step 5: Modify your application according to generated report if needed

More details online in User Guide:

https://source.tizen.org/compliance/application-compliance/web-application-checker-user-guide

Contribute to Tizen Web Checker Tool

- **Open source project initiated by Intel Open Source Technology Center**
- **Apache 2.0 license**
- **Source code**

  git clone ssh://username@review.tizen.org:29418/sdk/tools/web-app-checker.git

  Note: before getting source code, please read Work Flow and Developer Guide to configure your environment
- **Next Steps**
  - Integrate into Tizen SDK IDE
  - Support to check the Tizen UI Framework API

Questions?

TIZEN™

DEVELOPER SUMMIT

KOREA 2013