



Tizen Application Validation

Sul Jung
Tizen Validation Team

TIZEN™
**DEVELOPER
CONFERENCE**
2014
SAN FRANCISCO

Table of Contents

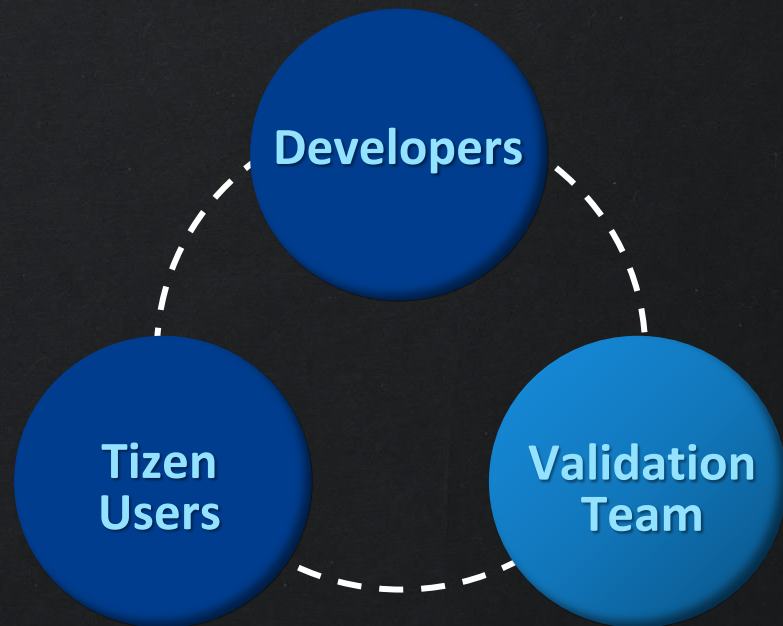
1. Introduction to Tizen Validation
 2. Validation Process
 3. Developer Support
 4. Frequently Reported Defects
-

1. Introduction of Tizen Validation

Introduction to Tizen Validation

Purpose

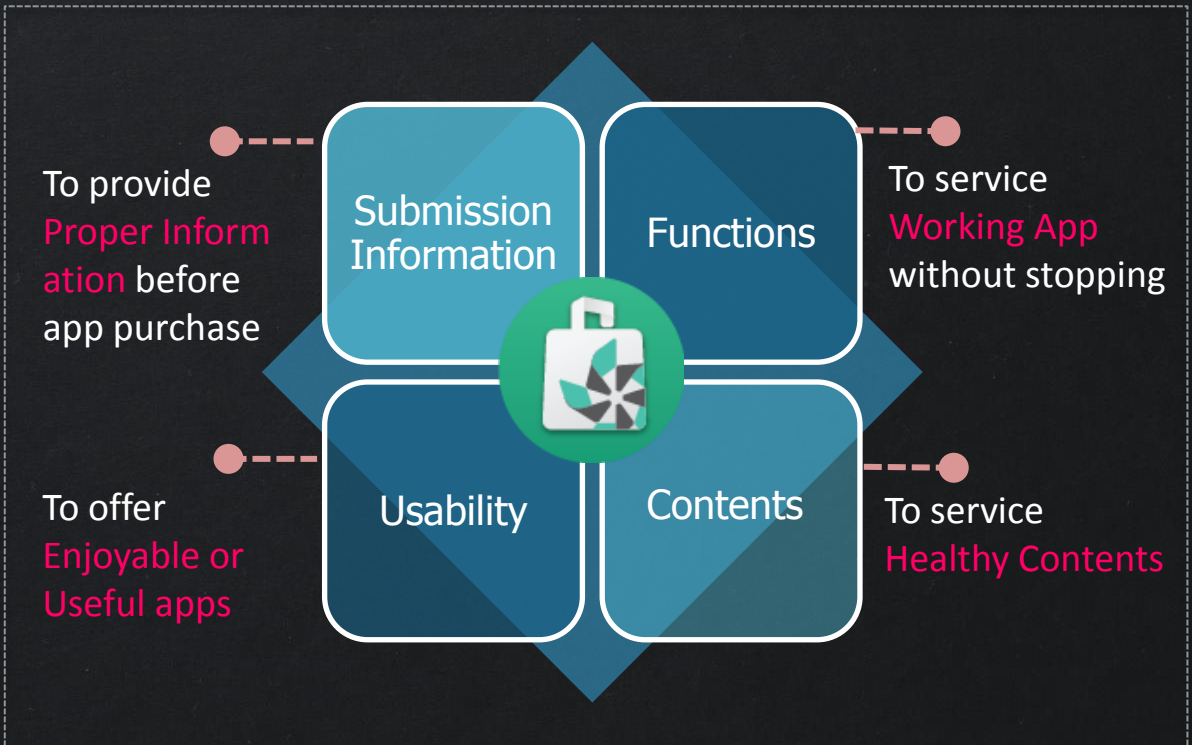
- Enable Successful Business for Developers
- Provide Qualified Application for Users



Introduction to Tizen Validation

Validation Criteria

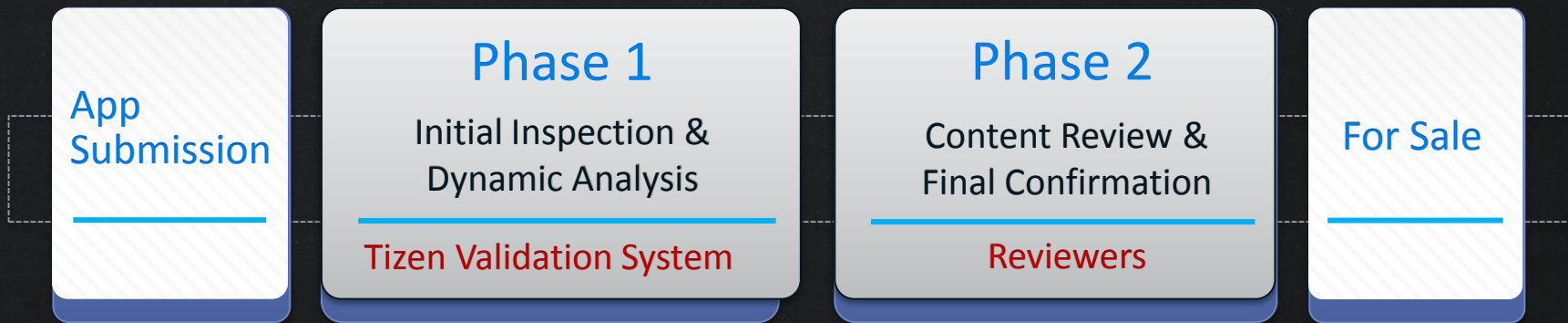
- Validation team checks four criteria for acceptance in Tizen store



2. Validation Process

Validation Process: 2 Phases

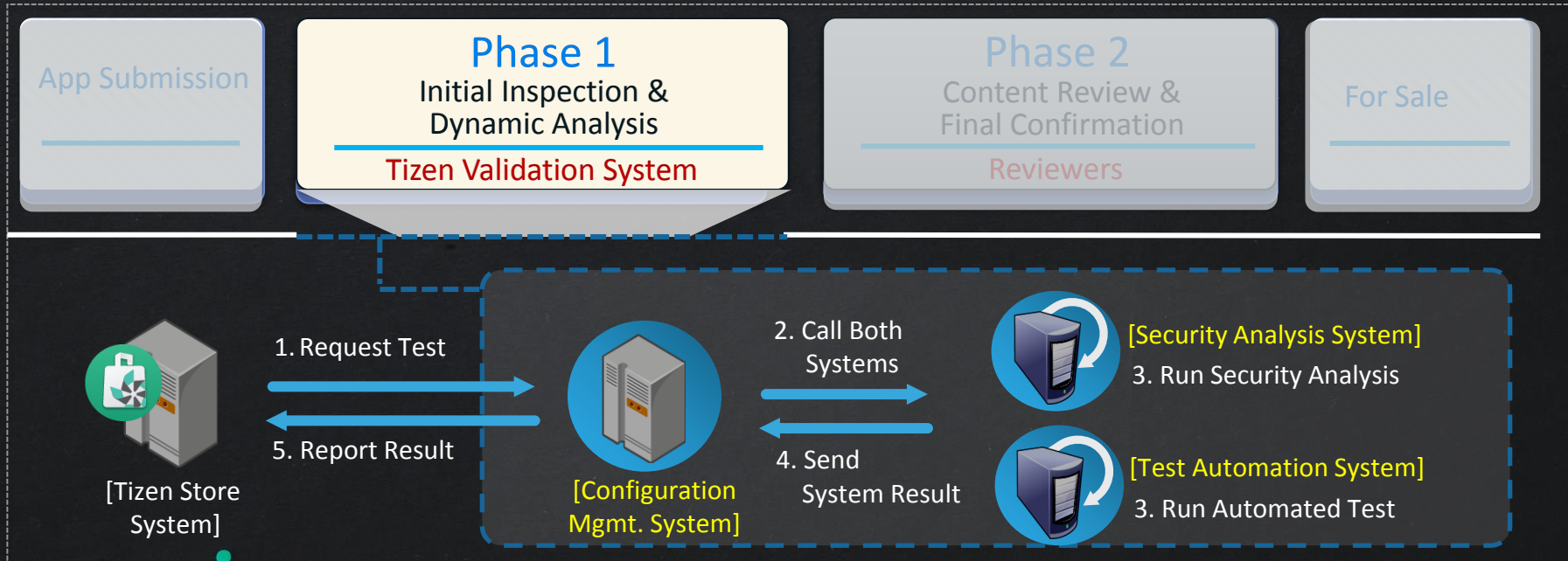
“WITHIN 3 DAYS”



Tizen Validation Process(Phase 1)

Tizen Validation System

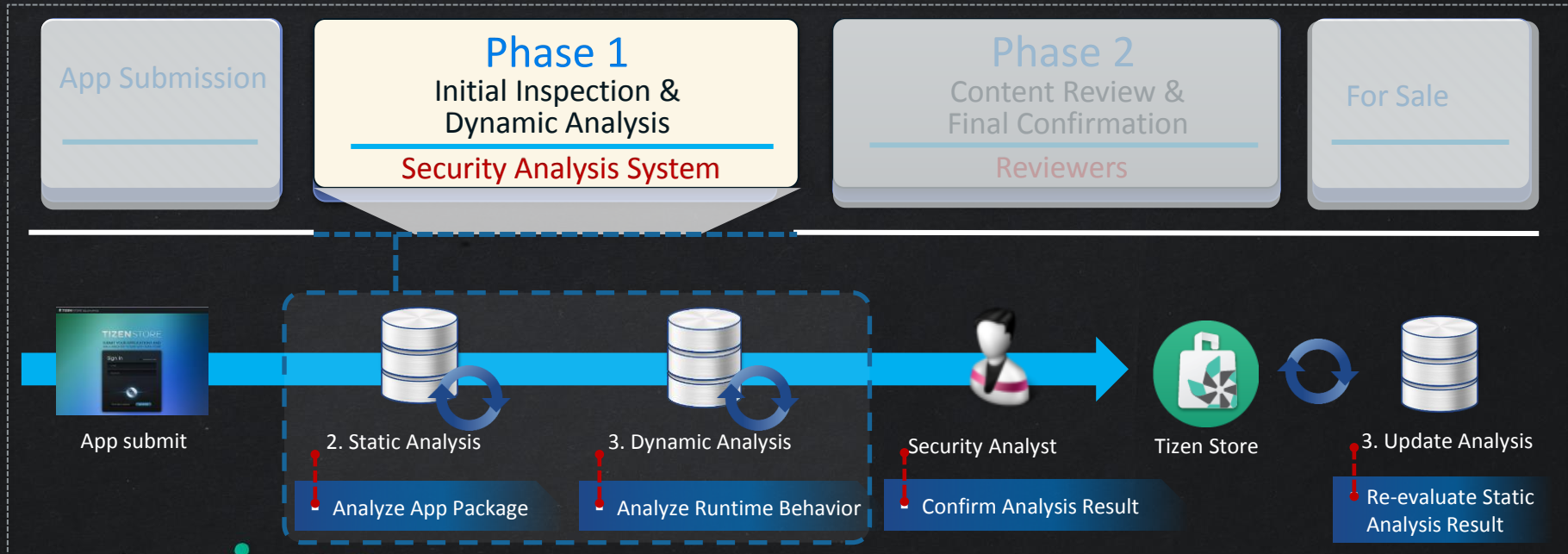
: Consists of Three systems – Security Analysis , Test Automation, Configuration Mgmt.



Tizen Validation Process(Phase 1)

Security System

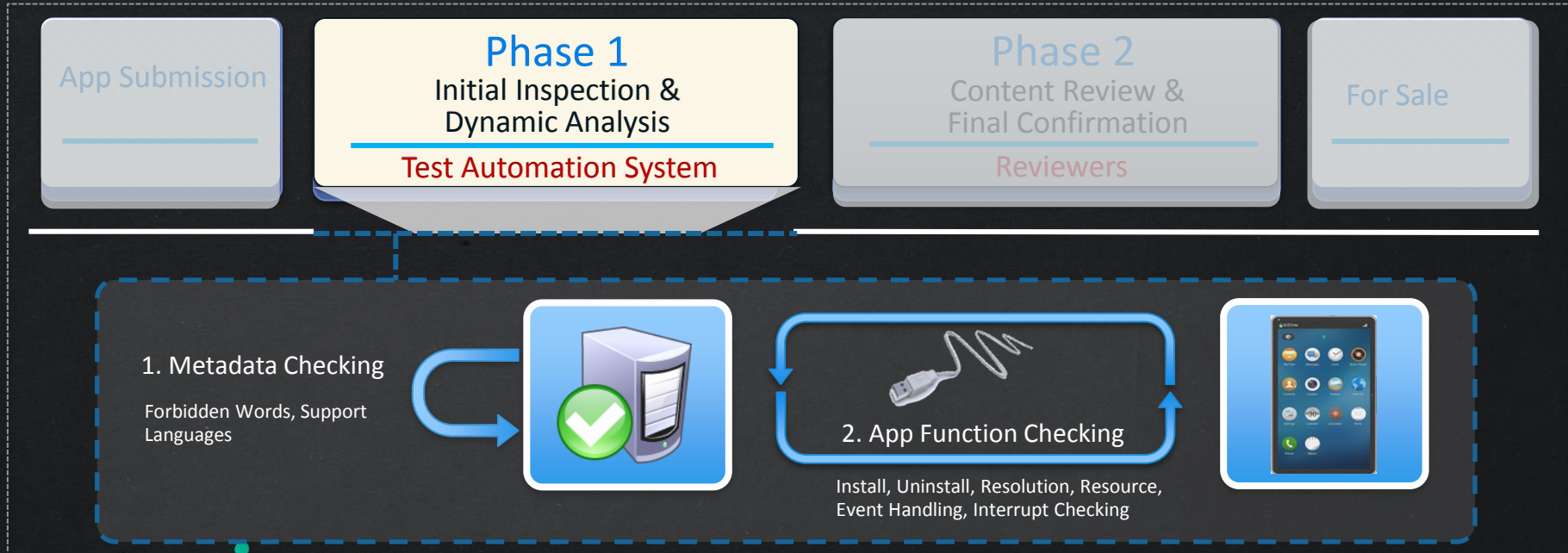
: Filters security threats such as Malware, Unauthorized API privilege, Web Attack Patterns



Tizen Validation Process(Phase 1)

Test Automation System

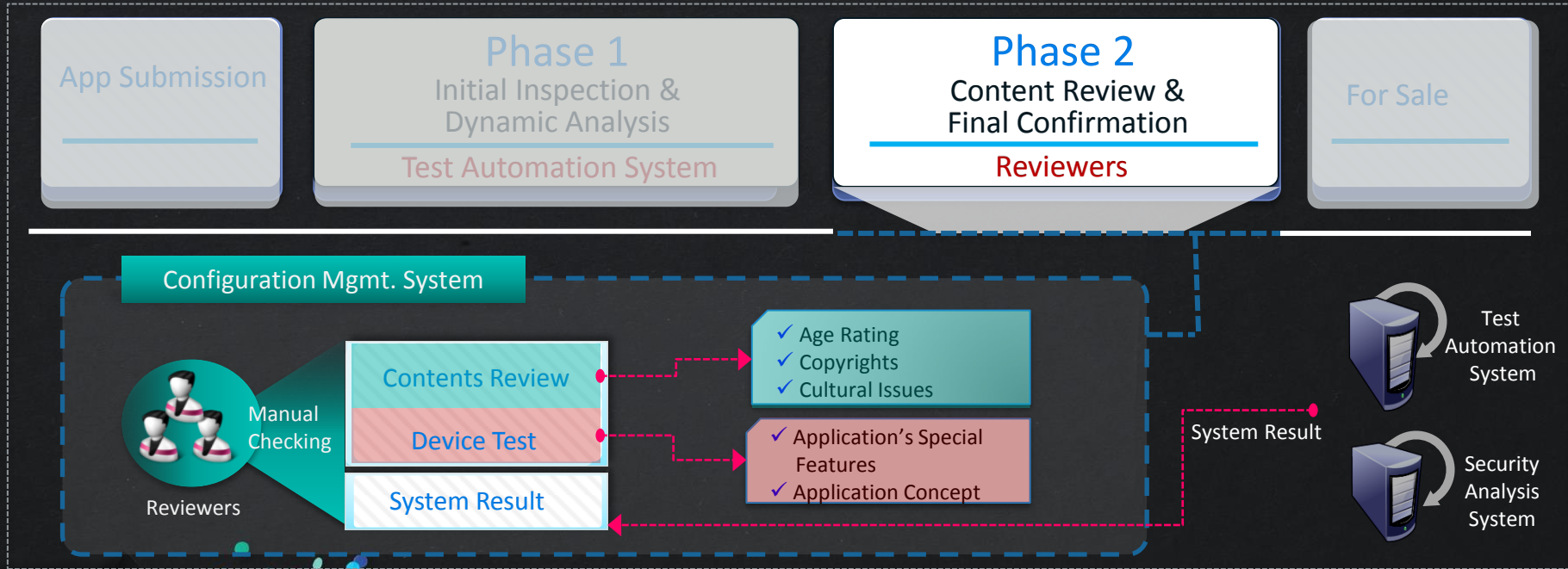
: Tests metadata and application's basic functions according to test cases



Tizen Validation Process(Phase 2)

Final Review

: Makes the final decision depending on system result and content review



3. Developer Support

Developer Support

3 Types of Validation Guidelines

: are provided for developers to get ready for successful application validation.

Validation Guide

- Enable developer to understand the publishing process

Self-Checklist

- Enable developer to check essential points for app function and validation policy

Top 5 Failures

- Enable developer to know mostly detected defects before submission

4. Frequently Reported Defects

Frequently Reported Defects

Privilege defects

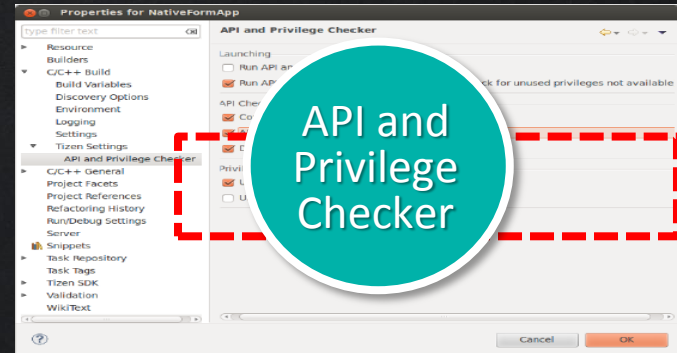
: are Detected by security analysis system

1. Unused Privilege Problems

: Delete an unused privilege

2. Undefined Privilege Problems

: Assign proper privileges in a configuration file



Frequently Reported Defects

Function Defects

: are Detected by test automation system

1. Installation Error



31%

&

2. Execution Error



29%

=

60%

of all functional defects

Tips

Use RTL!

: Developer can use the Samsung RTL(Remote Test Lab) to test and tune the application before submission

Frequently Reported Defects

🌀 Contents defects

: are checked by Reviewers before confirming the validation

- **Inappropriate Description**

➔ Write a description in accordance with Tizen, NOT other platforms

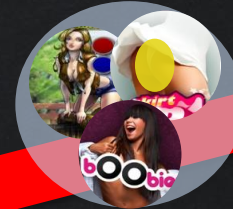
- **Not for All Ages**

➔ Make a Metadata (App name, description, icon and screenshots, etc) for everyone regardless of age

- **Copyright Infringement**

➔ Make sure your application does not infringe someone else's copyrights

Sexuality



Violence



LOGO
MOVIE **MUSIC**
IMAGE **E-BOOK**



Tizen Application Security Analysis System

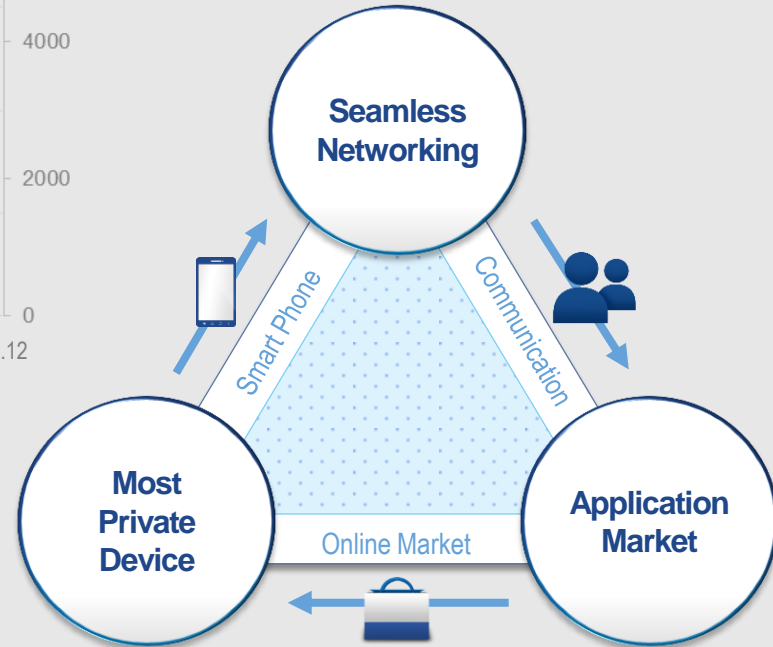
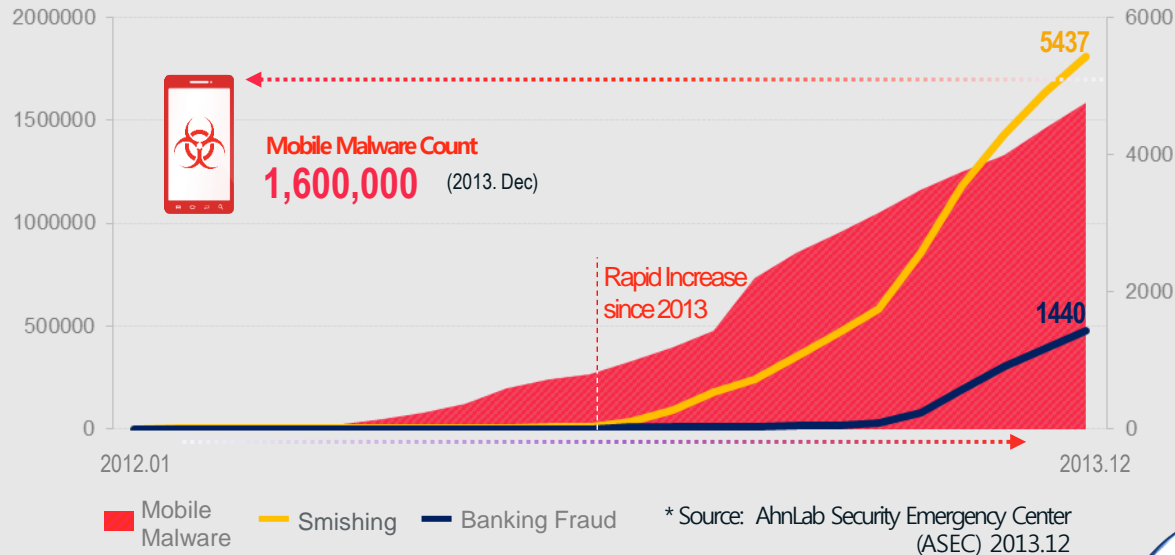
Seung Won Ko
Jackson Kang

TIZEN[™]
**DEVELOPER
CONFERENCE**
2014
SAN FRANCISCO

Security Analysis System



Need for Security Solution (1/2)



Need for Security Solution (2/2)

Google
Play

Basically laissez-faire,
Post verification method



Hard to manage quality and security,
most malware apps



Bouncer Introduction



Bypass method found,
Manufacture and telecommunication
companies are conceiving security plan

Compare to Google Play
provide Safer app store



Maintain Clean store status,
Benefit for developer and user

Tizen
AppStore

New Open source Web platform
With progress
Quality and security needs stabilization



AhnLab
Mobile Security Solution

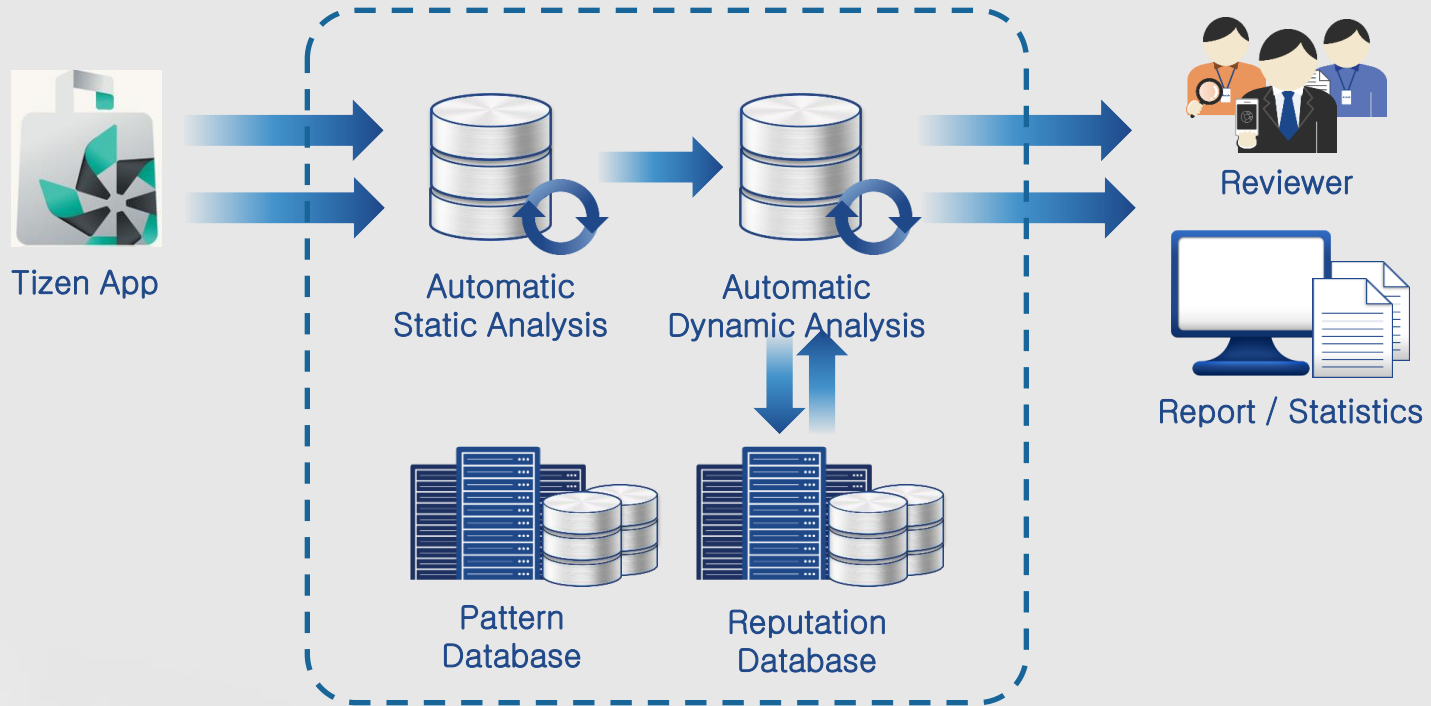


Static & Dynamic & Manual
Total security solution



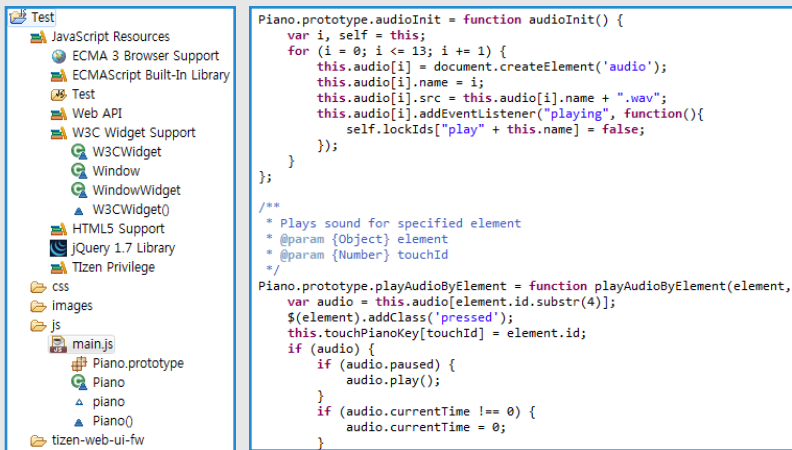
Maintain safe and attractive Tizen ecosystem
for developer and user,
Contribute platform's prosperity

Security Solution Overview



Static Analysis System (1/2)

Static Feature Analyze



The screenshot shows a static analysis tool interface. On the left is a file explorer with a tree view containing folders like 'JavaScript Resources', 'ECMA 3 Browser Support', 'ECMAScript Built-In Library', 'Test', 'Web API', 'W3C Widget Support', 'HTML5 Support', 'jQuery 1.7 Library', 'Tizen Privilege', 'css', 'images', 'js', and 'tizen-web-ui-fw'. The 'Piano.prototype' folder is expanded, showing 'Piano' and 'Piano()' files. On the right, a code editor displays JavaScript source code for 'Piano.prototype.audioInit' and 'Piano.prototype.playAudioByElement'.

```
Piano.prototype.audioInit = function audioInit() {
    var i, self = this;
    for (i = 0; i <= 13; i += 1) {
        this.audio[i] = document.createElement('audio');
        this.audio[i].name = i;
        this.audio[i].src = this.audio[i].name + ".wav";
        this.audio[i].addEventListener("playing", function(){
            self.lockIds["play" + this.name] = false;
        });
    }
};

/**
 * Plays sound for specified element
 * @param {Object} element
 * @param {Number} touchId
 */
Piano.prototype.playAudioByElement = function playAudioByElement(element,
    var audio = this.audio[element.id.substr(4)];
    $(element).addClass('pressed');
    this.touchPianoKey[touchId] = element.id;
    if (audio) {
        if (audio.paused) {
            audio.play();
        }
        if (audio.currentTime !== 0) {
            audio.currentTime = 0;
        }
    }
};
```

R
U
L
E

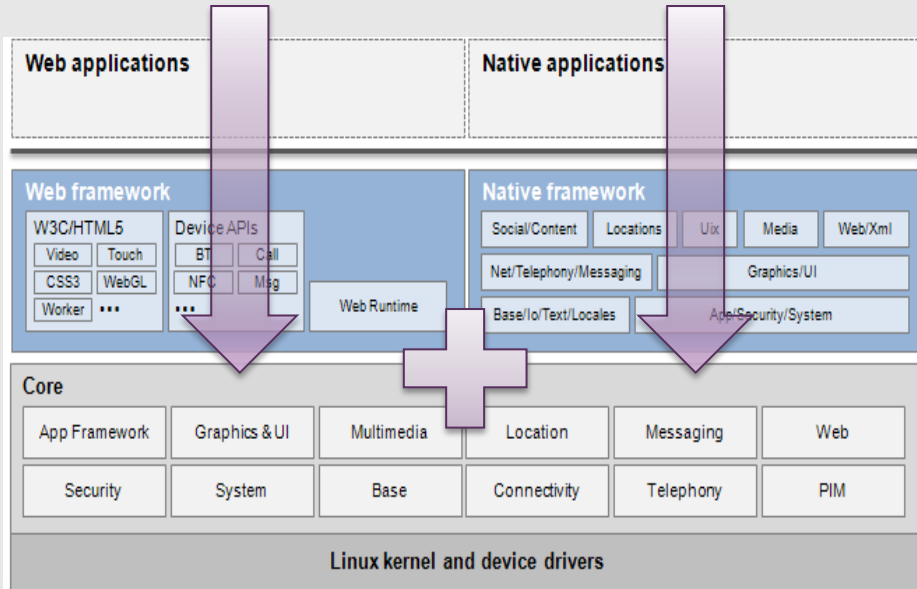
Web App

Native App

Hybrid App

- Package Binary / Signing
- Resource File / Source Code
- Etc.

Static Analysis System (2/2)

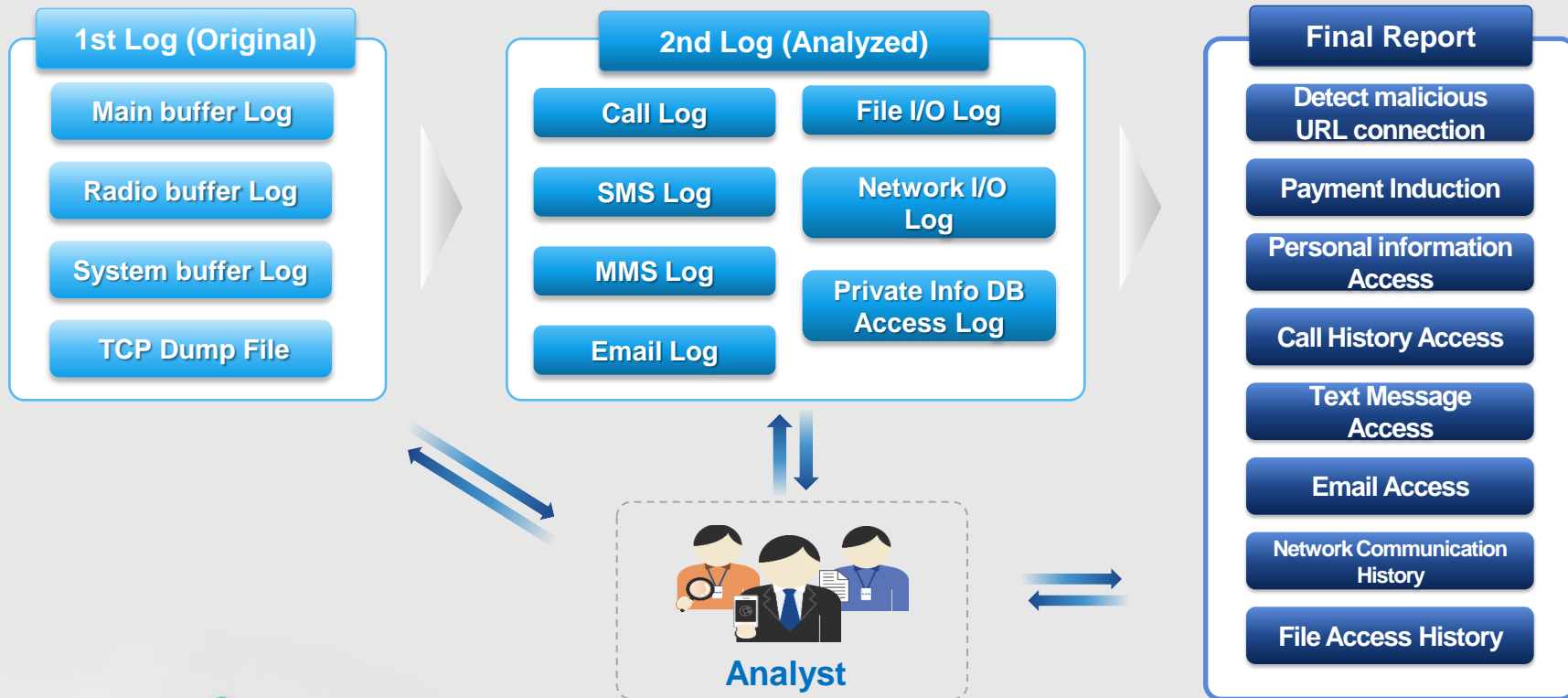


- | | |
|---------------------------|------------------------|
| Hash Check | Call Use Check |
| Sign Check | Message Use Check |
| Privilege Level Check | Contacts Use Check |
| Undefined Privilege Check | Calendar Use Check |
| Unused Privilege Check | Geocoder Use Check |
| Malicious URL Check | Device Info. Use Check |
| Specified URL Check | NFC Check |
| Specified API Check | Bluetooth Check |
| Specified String Check | VoIP Check |
| Network Use Check | China Checklist |
| Malware Check | Push Message Check |
| | White List Check |

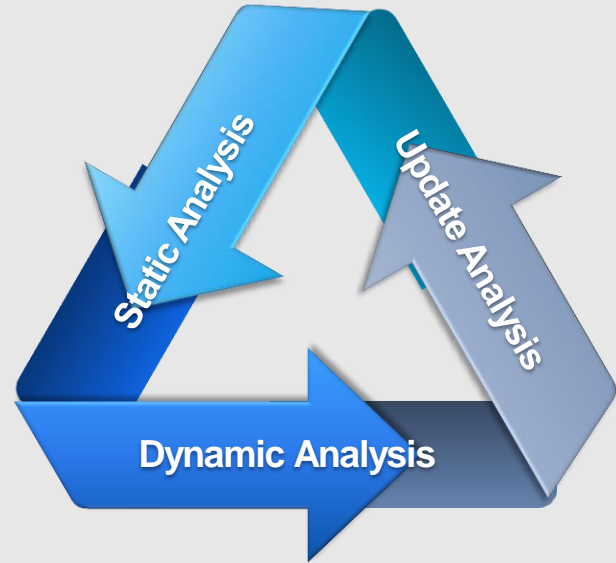
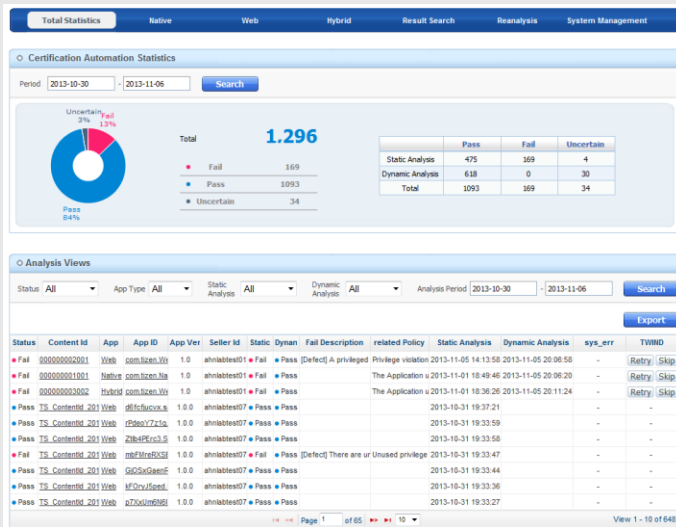
Dynamic Analysis System (1/2)



Dynamic Analysis System (2/2)



Update Analysis System



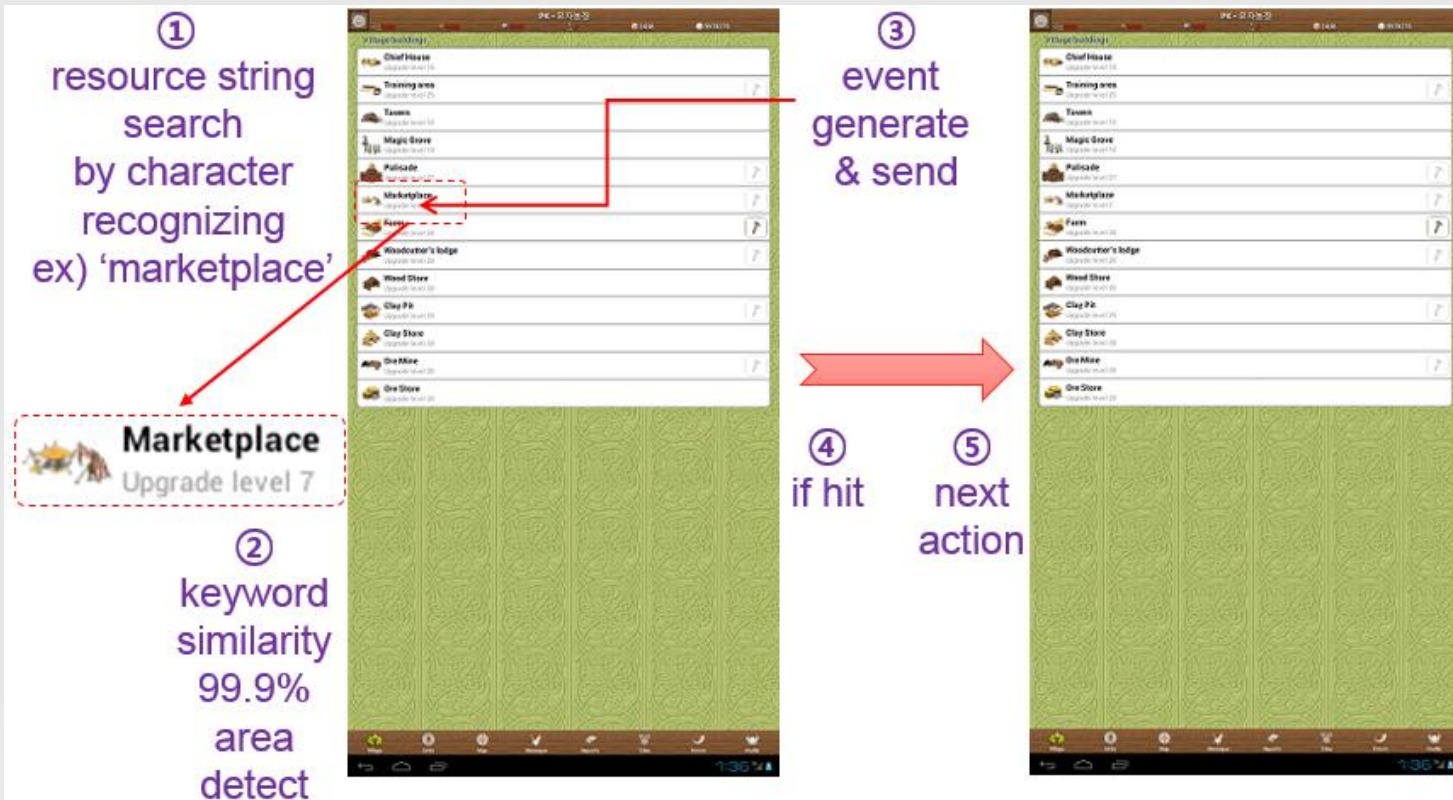
- To apply the latest verification policy and solve urgent security issue on pre-registered app
- Evaluation is not just a single verification test, but periodical purification effort

Malicious URL Check	Specified API Check
Specified URL Check	Specified String Check
Malware Check	Platform Version Check

Dynamic Analysis – Evolution (1/3)



Dynamic Analysis – Evolution (2/3)



Dynamic Analysis – Evolution (3/3)

AS IS



TO BE

Next Generation

2nd Generation

1st Generation

- **Emulator Introduction**
- **Dynamic Analysis Automation**
 - Automated process of app crawling, installation, execution, termination
- App execution and log the behavior for predetermined time

- **System performance improvement**
 - Network packet capture and analysis
 - System resource monitoring
 - Artificial system event occurrence
 - Offer processed summary information
- **Limit and Problem**
 - Lack of movement and interaction to draw all the functions of the application

- **Similar to that of humans**
 - Behavior Induction
 - Interaction
- **Effective Automation**
 - Object targeted Event Generation
 - Pattern Recognition
- **Enhanced Report**
 - Trace Route Recoding
 - All Screen Capture
- **Change of Paradigm**
 - From Passive to Active
 - Defense to Unknown App
- **Platform Independent**
 - Generalizable Model
 - Web platform, Smart TV platform, Etc.



Demonstration

Intelligent Dynamic Analysis





TIZEN™
DEVELOPER
CONFERENCE
2014
SAN FRANCISCO