Crosswalk Security on Tizen 3.0

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

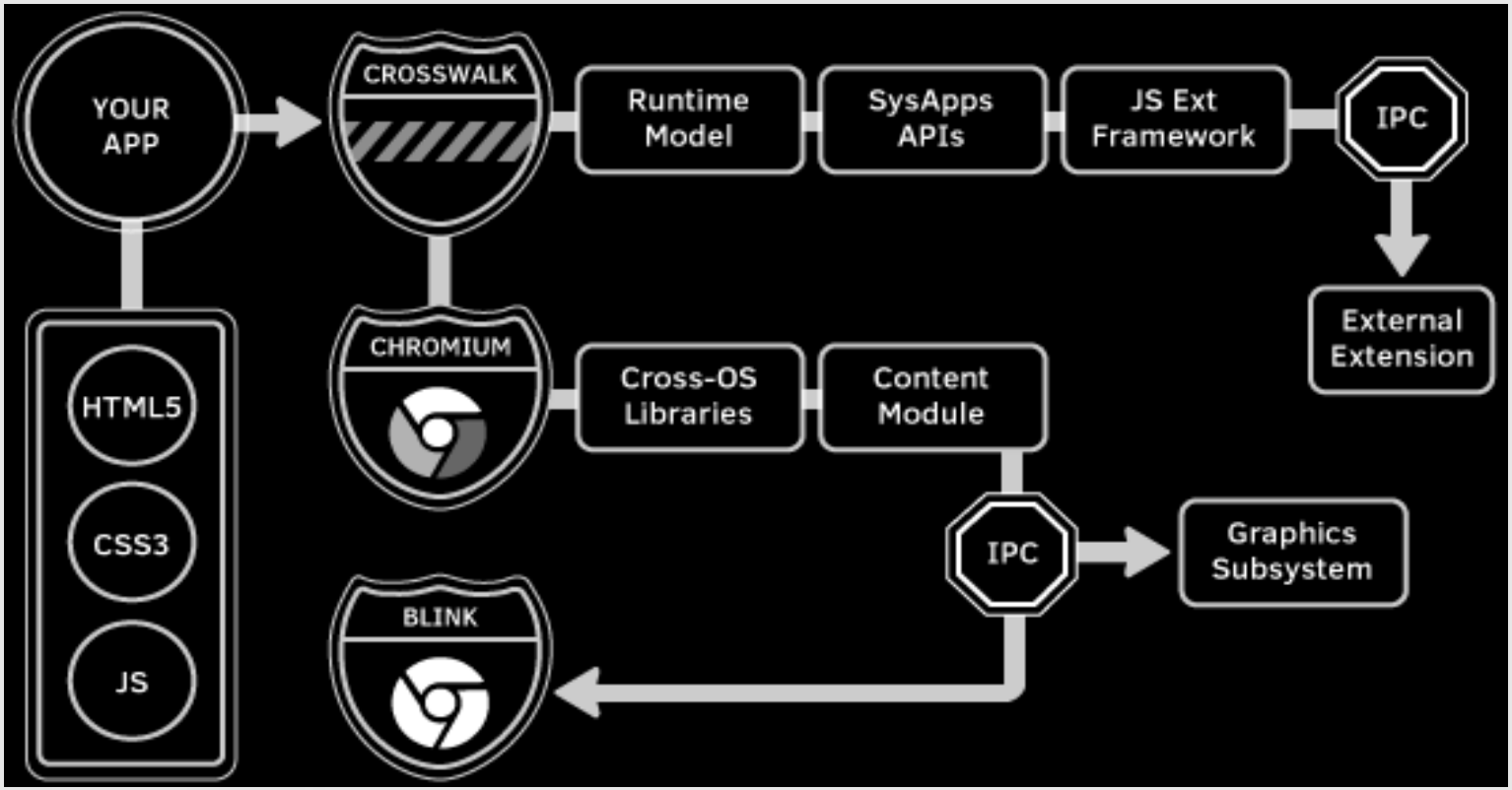
Agenda

- Tizen 3.0 Security Overview
- Crosswalk Overview
- Crosswalk Security Features
- Conclusion
- Contribute to Crosswalk

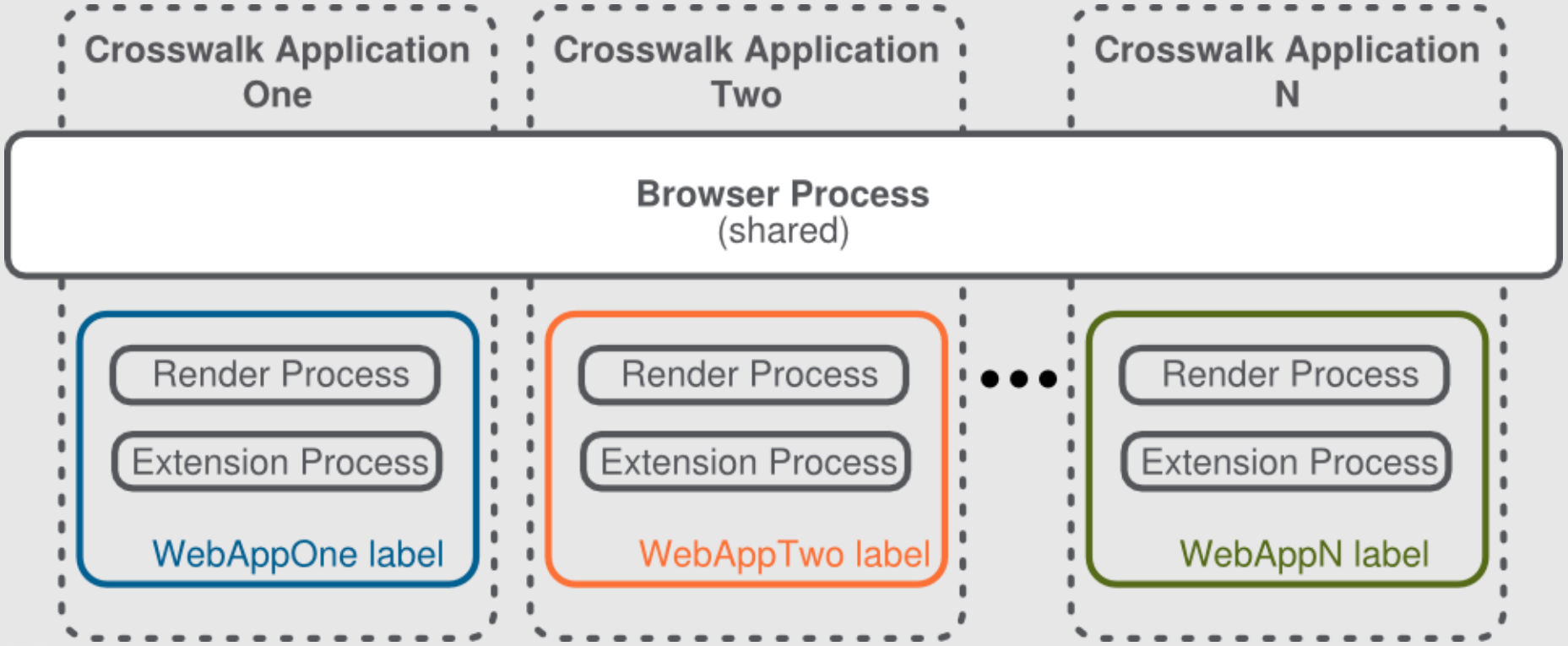
Tizen 3.0 Security Overview

- **The Objectives:**
  - Protect user data
  - Protect system resources (including the network)
  - Provide application isolation
- **Tizen 3 provide key security features:**
  - Discretionary Access Control (DAC)
  - Mandatory Access Control (MAC) with the kernel LSM SMACK
  - Cynara the permission management and check service
  - Security manager

Crosswalk Overview



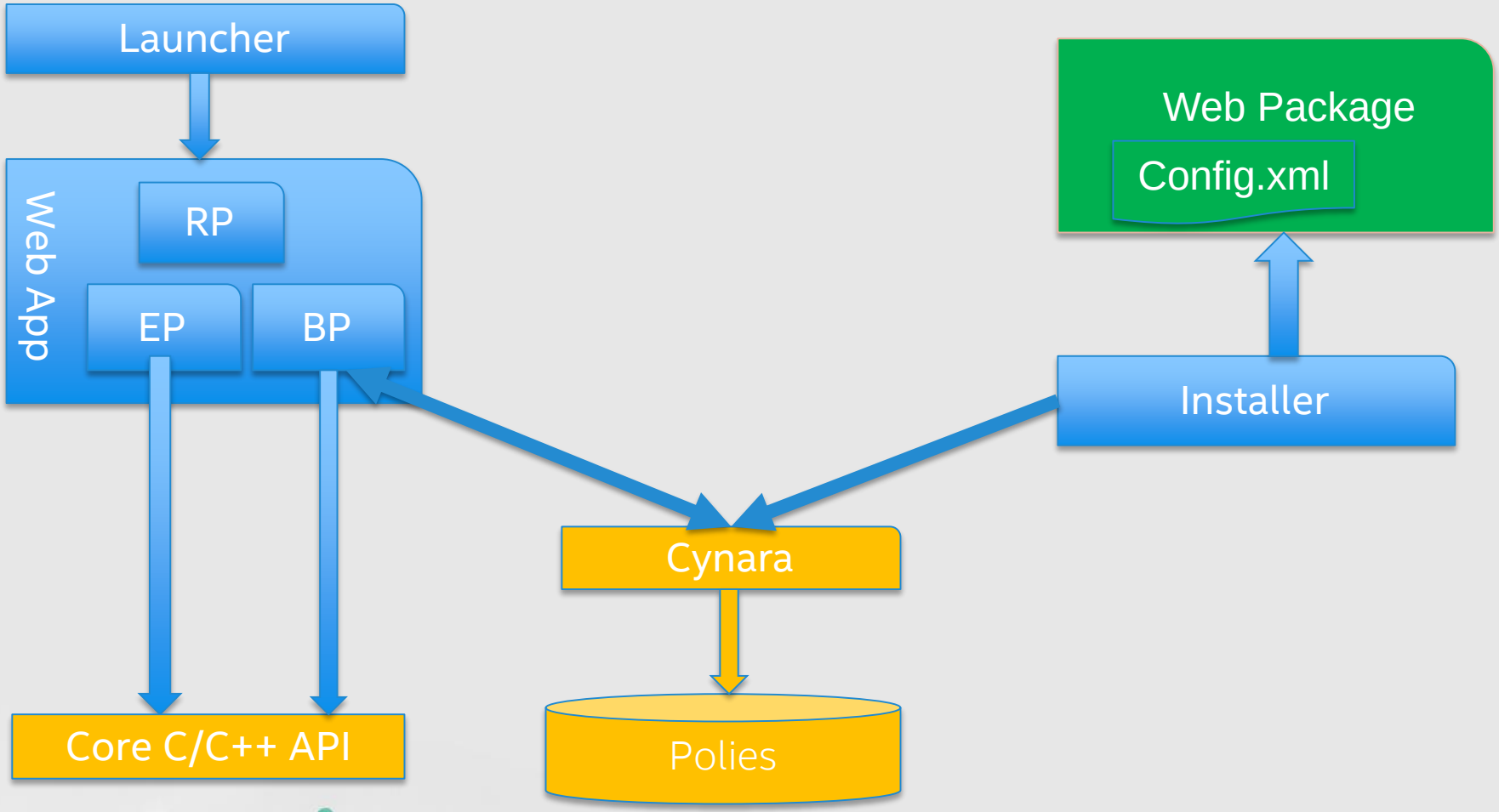
Crosswalk Share Process Mode for Per User



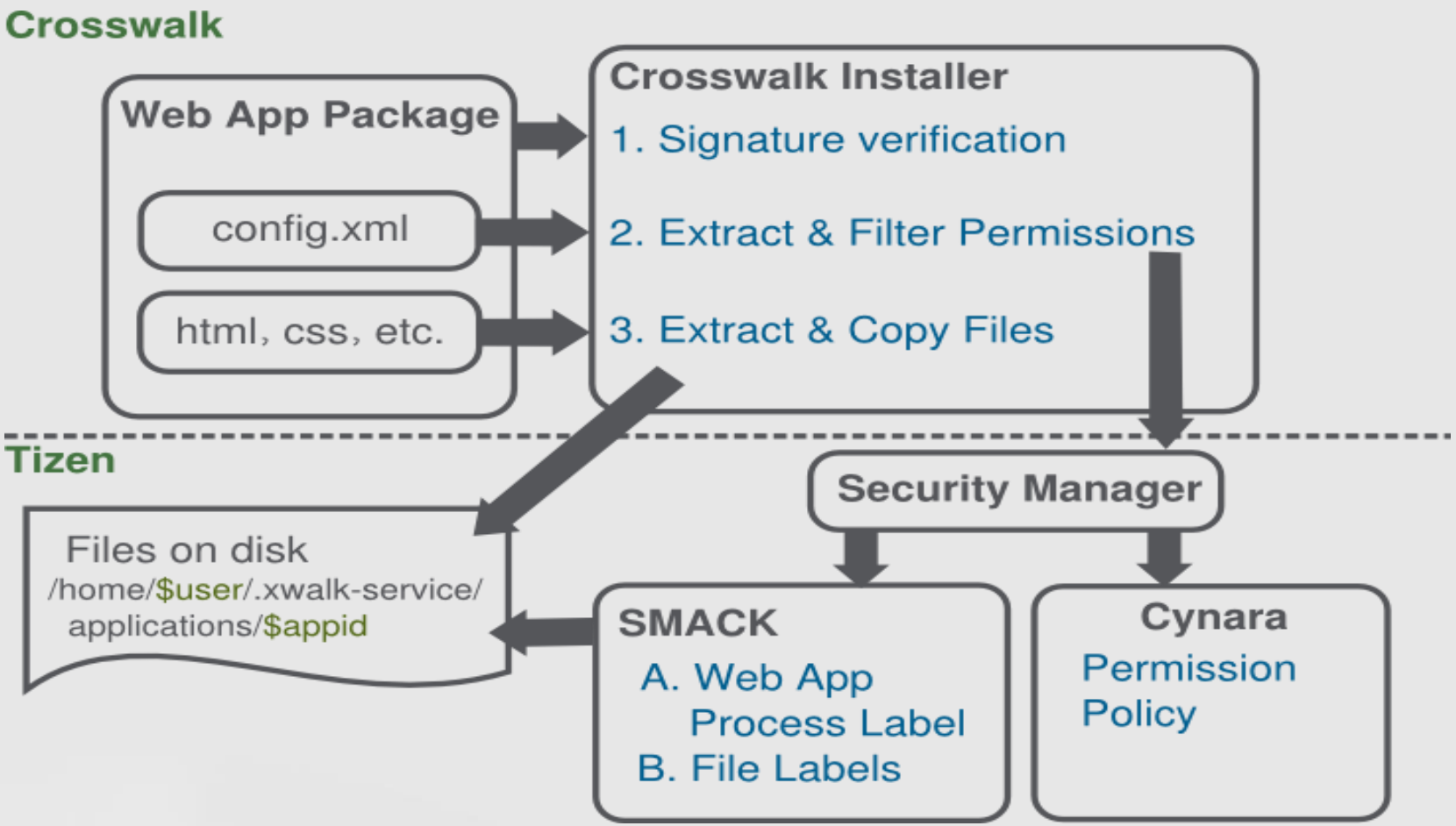
Crosswalk Security Features

- **Relationship with Tizen 3.0 security**
  - Build over: SMACK, Cynara and security manager
- **Goals of Crosswalk Security:**
  - Keep sensitive resource and data safely
  - Web applications should be run with least privileges
  - Web applications are sandboxed so they can only see their own resources
  - Mitigate a broad class of content injection vulnerabilities, such as cross-site scripting (XSS) attacks
- **Features**
  - API access control
  - Web widget application signing
  - CSP (Content Security Policy)
  - WARP (Widget Access Request Policy)

API Access Control



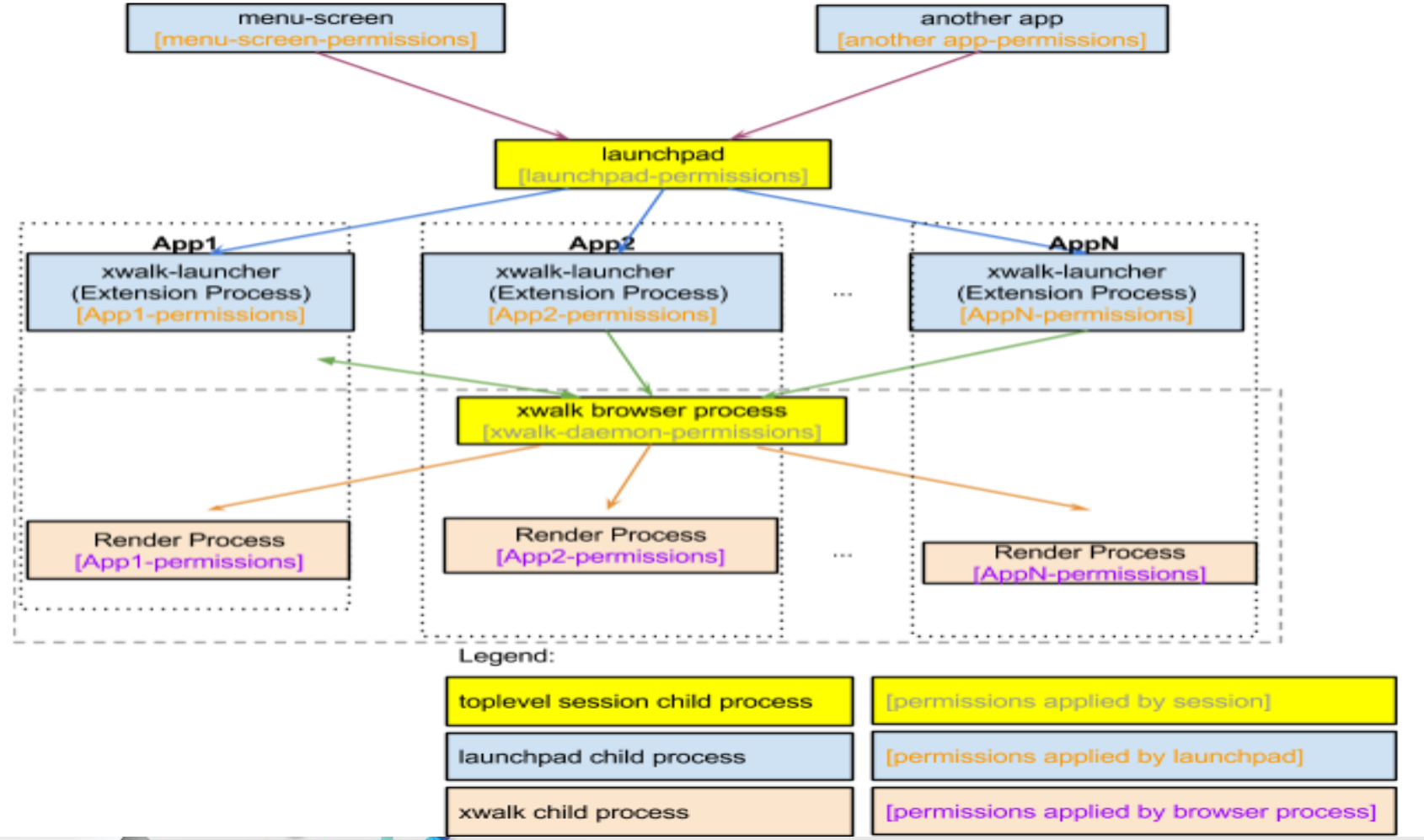
API Access Control - Crosswalk Installer



API Access Control - Crosswalk Installer (Cont.)

- **General flow of installing web application**
  1. Crosswalk installer unzip Crosswalk web application package
  2. Crosswalk installer verify signature of the package
  3. Crosswalk extract permissions list from config.xml
  4. According to privilege level, Crosswalk filter invalid permissions out of permissions list
  5. Crosswalk installer calls security-manager to insert policies and set SMACK label to resource files

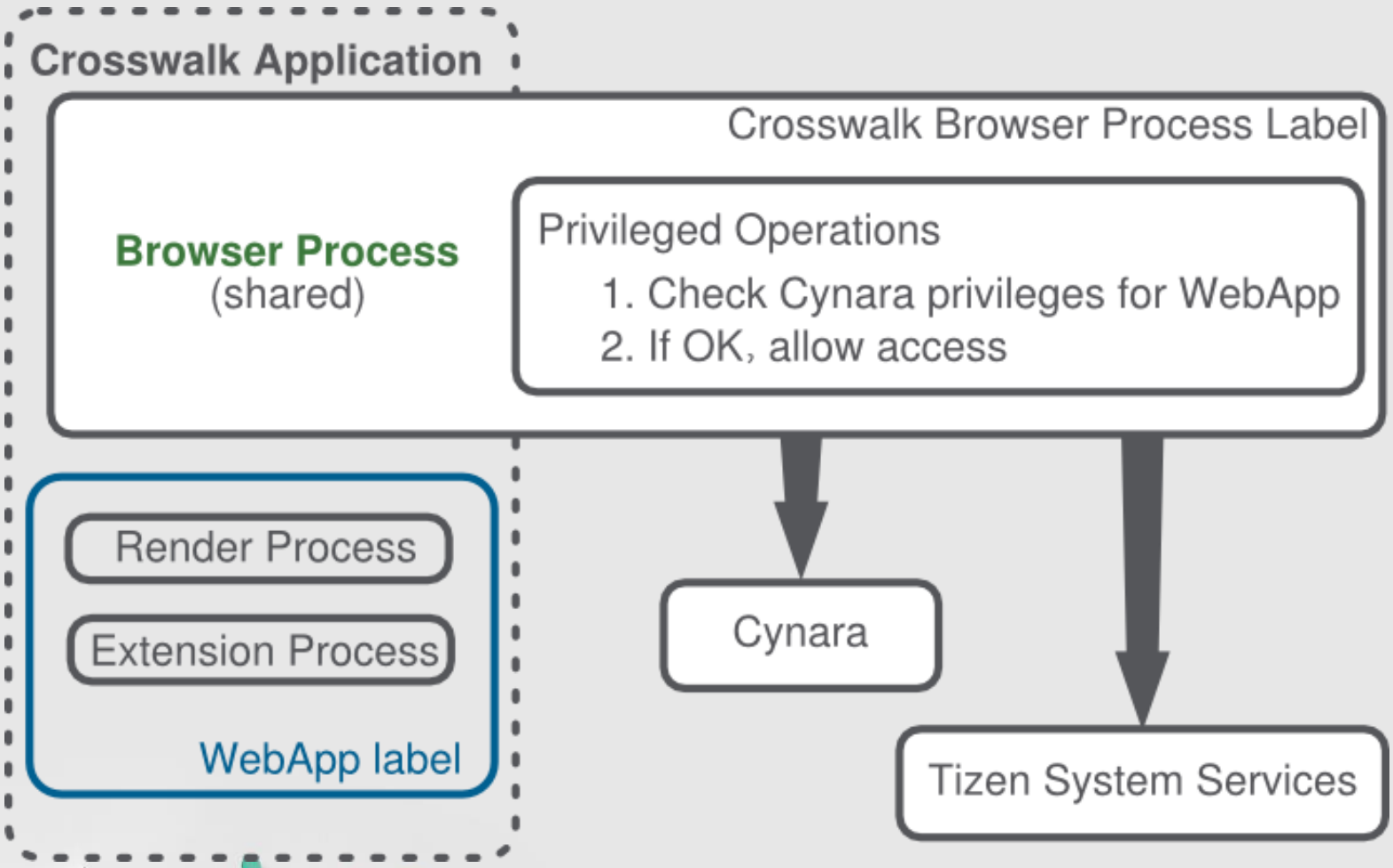
API Access Control – Launcher



API Access Control – Launcher (Cont.)

- **General flow of launching web application**
  1. User trying to launch an application by clicking the application icon in home screen or another application throwing an intent to launch web app.
  2. The launchpad_daemon fork a child process xwalk-launcher and xwalk-launcher create a new xwalk extension process instance
  3. xwalk-launcher send an dbus message to running application manager (Browser process). Application manager launch web application and start a new render view(render process)

API Access Control – Runtime Check



API Access Control – Runtime Check (Cont.)

- **General flow of checking API access in browser process**
  1. When sensitive W3C JS API is invoked, render process send IPC to browser process
  2. Browser process request Cynara to check API permission
  3. Cynara return ALLOW/DENY to browser process
  4. If the operation is allowed, browser process access Tizen system service

Application Signing

- **Tizen application must be signed with 2 signatures:**
  - Author signature
  - Distributor signature
- **Based on W3C recommendations for XML digital signature**
- **Crosswalk verifies that the application has been properly signed with the certificate**
- **Decide privilege level of web application**
  - Platform
  - Partner
  - Public
  - Untrusted

Content Security Policy (CSP)

- CSP works as a whitelisting mechanism for resources loaded or executed by web applications
- Policy applies to a wide variety of resources
- The policy is defined via the application's manifest as follows

```
{
    ...,
    <tizen:content-security-policy>script-src 'self'</tizen:content-security-policy>
    ...
}
```

- CSP support in Crosswalks is based on Chromium and Blink implementation

Widget Access Request Policy (WARP)

- All network accesses by widgets are denied by default
- A widget must declare in its manifest which network resources it will access (such as XMLHttpRequest, iframe, img, script, etc.)
- <access> element in config.xml. Developers can specify protocols, domains, and sub-domains.

```
<widget xmlns="http://www.w3.org/ns/widgets">
…
…
<access origin=https://example.net subdomains="true" />
<access origin=http://example.com subdomains="false"/> />
…
</widget>
```

Conclusion

- **To developers:**
  - You need to declare the required permissions in the manifest
  - Declare the minimum set of permissions you really need
  - Pay attention to proper error handling in your application

Acknowledgements

- **Thanks the contribution from Terri Oda, Casey Schaufler, Xinchao He, Yongkang You and Peter Wang**

Contribute to Crosswalk

- **Project website:**
  - https://crosswalk-project.org
- **Mail list:**
  - crosswalk-help
  - crosswalk-dev
- **IRC**
  - #crosswalk atirc://freenode.net
- **Bug Track:**
  - https://crosswalk-project.org/jira/

Question?

TIZEN™ DEVELOPER SUMMIT 2014 SHANGHAI

TIZEN开发者峰会（上海）