Tizen-Meta as Security and Connectivity Layers For Yocto Project

Dominig ar Foll
(Intel Open Source Technology Centre)
dominig.arfoll@fridu.net

October 2014

Tizen-Meta

- What is Tizen
- How to build Tizen with Yocto tools
- Which Connectivity is available with Tizen
- How Security is enforced in Tizen
- What's next.

Tizen, an OS for Connected Devices

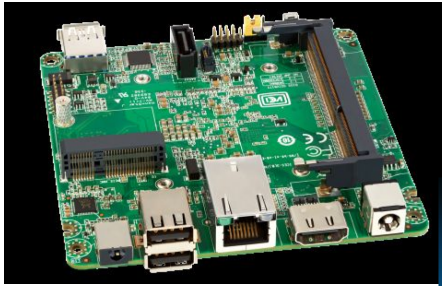
Multiple profiles:

- Mobile
- IVI
- TV
- Household equipments
- Wearables

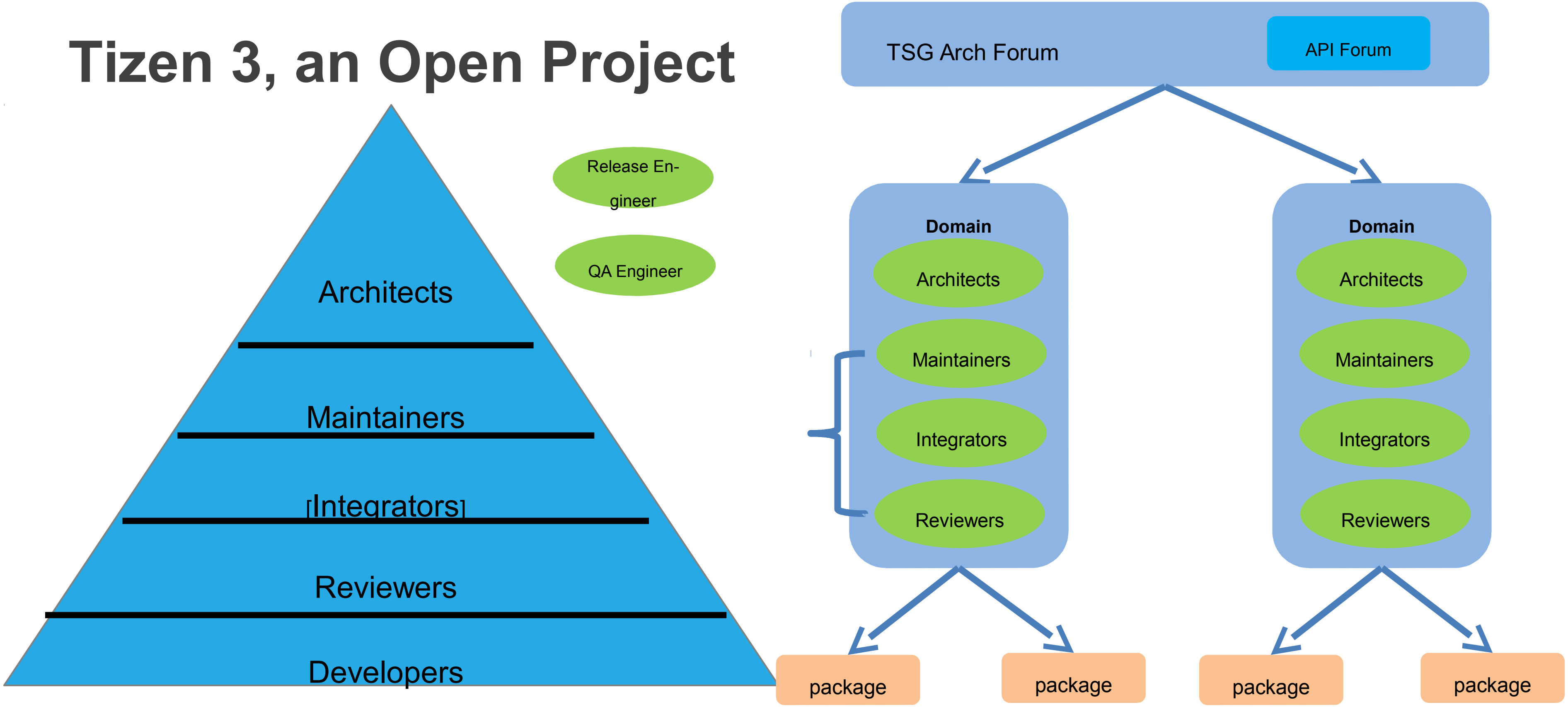
Hacker Friendly supported platforms

- Intel
  - NUC
  - MinnowBoard Max
  - Galileo-2

- ARM
  - Odroid U3

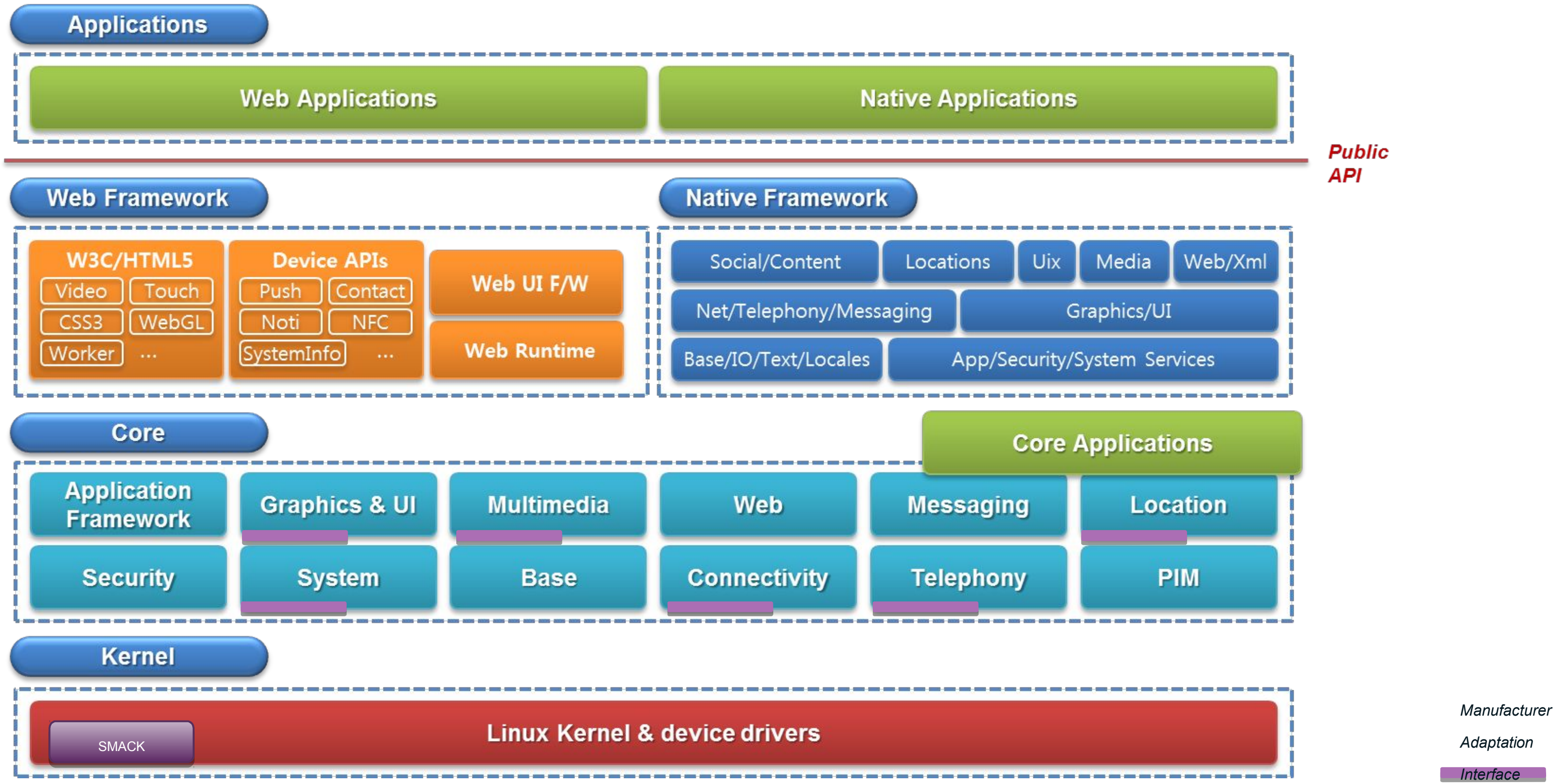


Tizen 3, an Open Project



Dominig ar Foll
Intel Open Source Technology Centre

Architecture Overview (Mobile Profile)



Applications

Web Applications | Native Applications

Public API

Web Framework

W3C/HTML5: Video, Touch, CSS3, WebGL, Worker, …

Device APIs: Push, Contact, Noti, NFC, SystemInfo, …

Web UI F/W

Web Runtime

Native Framework

Social/Content, Locations, Uix, Media, Web/Xml

Net/Telephony/Messaging, Graphics/UI

Base/IO/Text/Locales, App/Security/System Services

Core

Core Applications

Application Framework, Graphics & UI, Multimedia, Web, Messaging, Location

Security, System, Base, Connectivity, Telephony, PIM

Kernel

SMACK

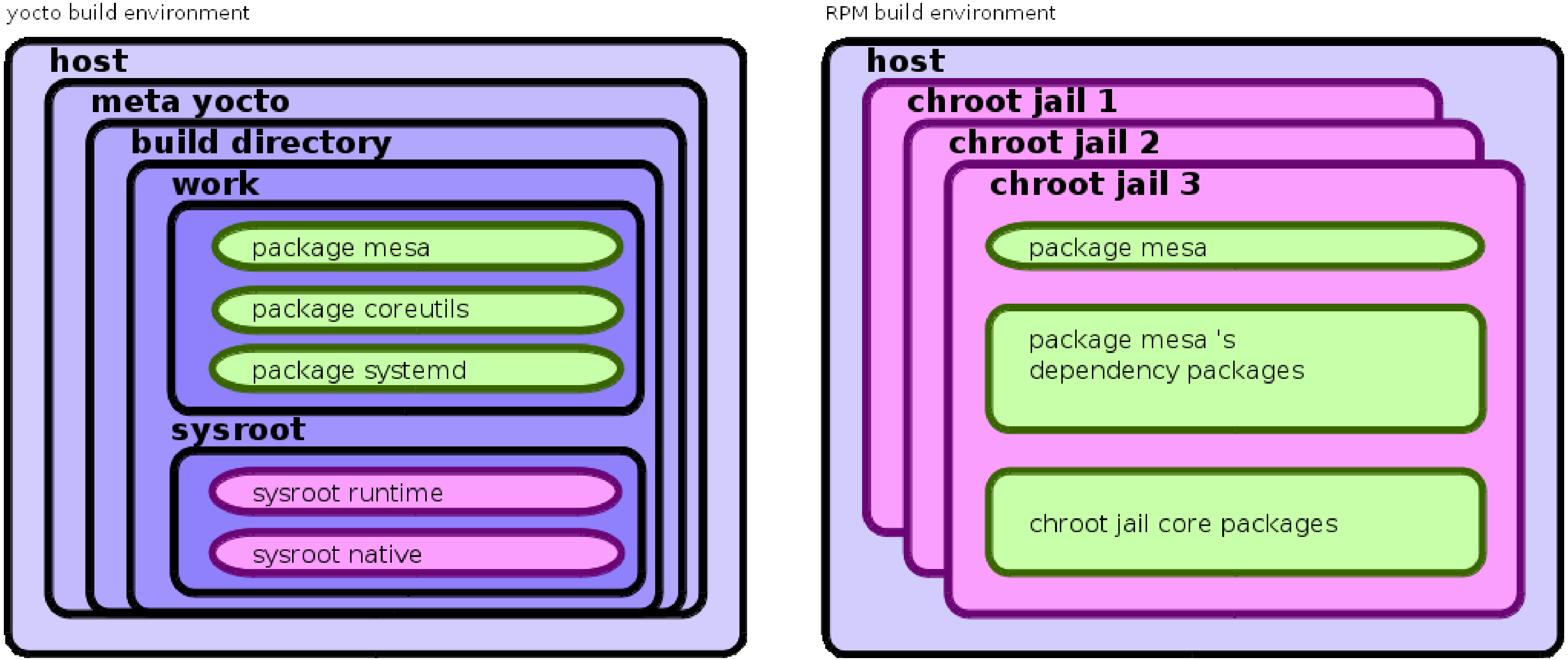
Linux Kernel & device drivers

Manufacturer

Adaptation

Interface

By default Tizen is built with an OBS



yocto build environment

host

meta yocto

build directory

work

package mesa

package coreutils

package systemd

sysroot

sysroot runtime

sysroot native

RPM build environment

host

chroot jail 1

chroot jail 2

chroot jail 3

package mesa

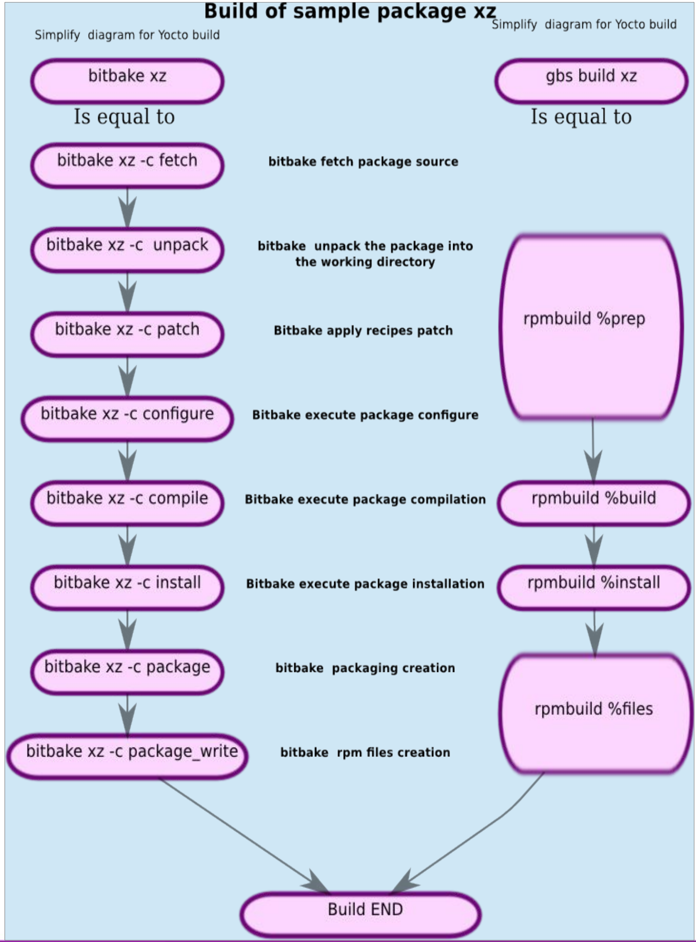
package mesa 's dependency packages

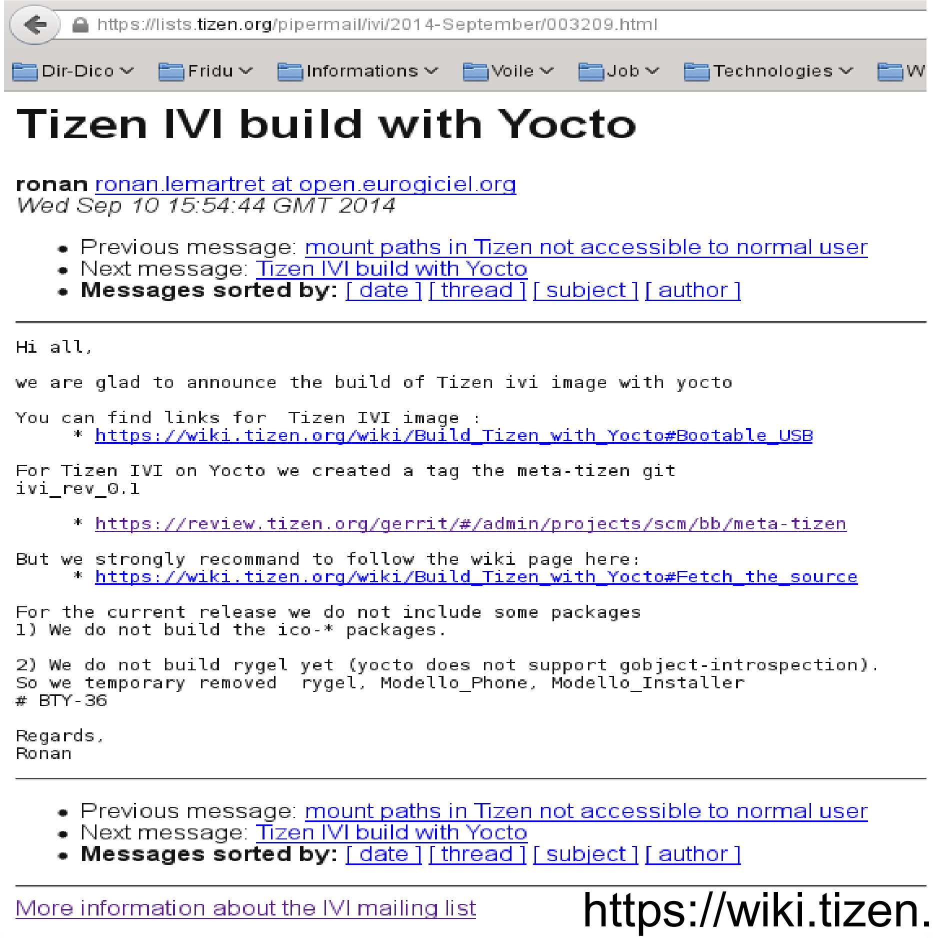
chroot jail core packages

But Nothing stop you yo build it with Yocto

From OBS to Yocto

- Use spec2yocto tool to generate a first recipe
- Correct the recipe to get the package built
- Modify the spec2yocto tool to directly get a correct recipe
- Use Yocto 1.7 to get the updated tools.


Build of sample package xz
Simplify diagram for Yocto build
Simplify diagram for Yocto build
bitbake xz
Is equal to
gbs build xz
Is equal to
bitbake xz -c fetch
bitbake fetch package source
bitbake xz -c unpack
bitbake unpack the package into the working directory
bitbake xz -c patch
Bitbake apply recipes patch
rpmbuild %prep
bitbake xz -c configure
Bitbake execute package configure
bitbake xz -c compile
Bitbake execute package compilation
rpmbuild %build
bitbake xz -c install
Bitbake execute package installation
rpmbuild %install
bitbake xz -c package
bitbake packaging creation
rpmbuild %files
bitbake xz -c package_write
bitbake rpm files creation
Build END


https://lists.tizen.org/pipermail/ivi/2014-September/003209.html
Dir-Dico
Fridu
Informations
Voile
Job
Technologies
W

Tizen IVI build with Yocto

ronan ronan.lemartret at open.eurogiciel.org
Wed Sep 10 15:54:44 GMT 2014

- Previous message: mount paths in Tizen not accessible to normal user
- Next message: Tizen IVI build with Yocto
- **Messages sorted by:** [date] [thread] [subject] [author]

```
Hi all,

we are glad to announce the build of Tizen ivi image with yocto

You can find links for  Tizen IVI image :
     * https://wiki.tizen.org/wiki/Build_Tizen_with_Yocto#Bootable_USB

For Tizen IVI on Yocto we created a tag the meta-tizen git
ivi_rev_0.1

     * https://review.tizen.org/gerrit/#/admin/projects/scm/bb/meta-tizen

But we strongly recommand to follow the wiki page here:
     * https://wiki.tizen.org/wiki/Build_Tizen_with_Yocto#Fetch_the_source

For the current release we do not include some packages
1) We do not build the ico-* packages.

2) We do not build rygel yet (yocto does not support gobject-introspection).
So we temporary removed  rygel, Modello_Phone, Modello_Installer
# BTY-36

Regards,
Ronan
```

- Previous message: mount paths in Tizen not accessible to normal user
- Next message: Tizen IVI build with Yocto
- **Messages sorted by:** [date] [thread] [subject] [author]

More information about the IVI mailing list

10 Sept 2014

https://wiki.tizen.org/wiki/Tizen_on_yocto

Tizen Connectivity*

- Bluetooth 4 (Low energy)
- Ethernet AV
- Wifi P2P
- GSM 3G/4G
  - Phone
  - Messages
  - Data

- Tethering
- Hand Free support
- Miracast
- DLNA
- Shared Drive
- Multi Screen

* hardware dependent

3 kinds of security

- Isolation of the applications
  - An application cannot access the data of other application
  - How? Use of Smack and DAC
- Restriction of the services
  - An application cannot access the services without authorisation
  - How? Use of Smack and Cynara
- Restriction of the network
  - An application cannot access network without authorisation
  - How? Use of Smack and netfilter

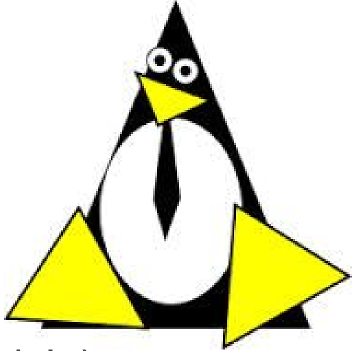
Isolation of applications

- The file system is cut in user parts using traditionnal Unix DAC uid partition
  - A user can access its own $HOME
  - A user cannot access the home of other users

- The file system is cut in application parts using the Smack MAC labels
  - Each application has its own label
  - An application can only access its own labelled files

	AppX alice	AppY alice	AppX bob	AppY bob
AppX alice	**YES**	NO (MAC)	NO (DAC)	NO (DAC+ MAC)
AppY alice	NO (MAC)	**YES**	NO (DAC+ MAC)	NO (DAC)
AppX bob	NO (DAC)	NO (DAC+ MAC)	**YES**	NO (MAC)
AppY bob	NO (DAC+ MAC)	NO (DAC)	NO (MAC)	**YES**

Short overview

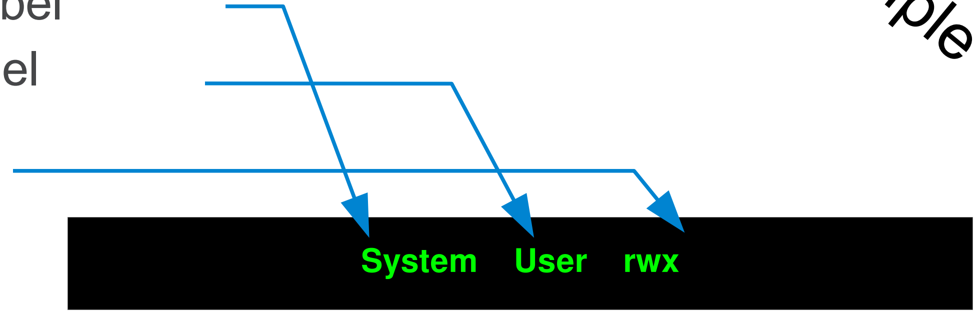


- The author of Smack is mainly Casey Schaufler.
- In Linux since kernel 2 6 25 – 17 April 2008 – as a LSM (Linux Security Module)
- Evolving since this first days.
- Inside Tizen since the first days (2012).
- Use extended file attributes to store data relating to files.
- Controlled via a filesystem interface: smackfs.
- Controls accesses of processes to files, IPC, sockets and processes (ptrace, signals, ...).
- Controls CIPSO labelled IPV4 packets

The Smack rules

- Smack's rules have 3 items:
  - the subject's label
  - the object's label
  - the access

Simple !!!



```
System   User   rwx
```

This rule tells to allow **read**, **write** and **execute** access to objects labelled **User** for the processes labelled **System**.

What are labels? **What are subjects?** **What are objects?** **How to set?**

The Smack vocabulary

- **Labels** are just text (of valid ASCII characters) without any special meaning: they are compared to equality (case sensitive: a≠A).
- **Subjects** are running processes: any running process has a smack label.
- **Objects** are **files**, **IPC**, **sockets**, **processes**.
- The label of a running process is called its **context**.
  - The commands id, ps (option -Z or -M), ls (option -Z) are prompting the contexts of the current process, the running processes, the files.
- The grantables **access modes** are: **read** (r), **write** (w), **execute** (x), **append** (a), **lock** (l), **transmute** (t).

Setting Smack

- How to set context? You can't! Except if you have the capability CAP_MAC_ADMIN.
-

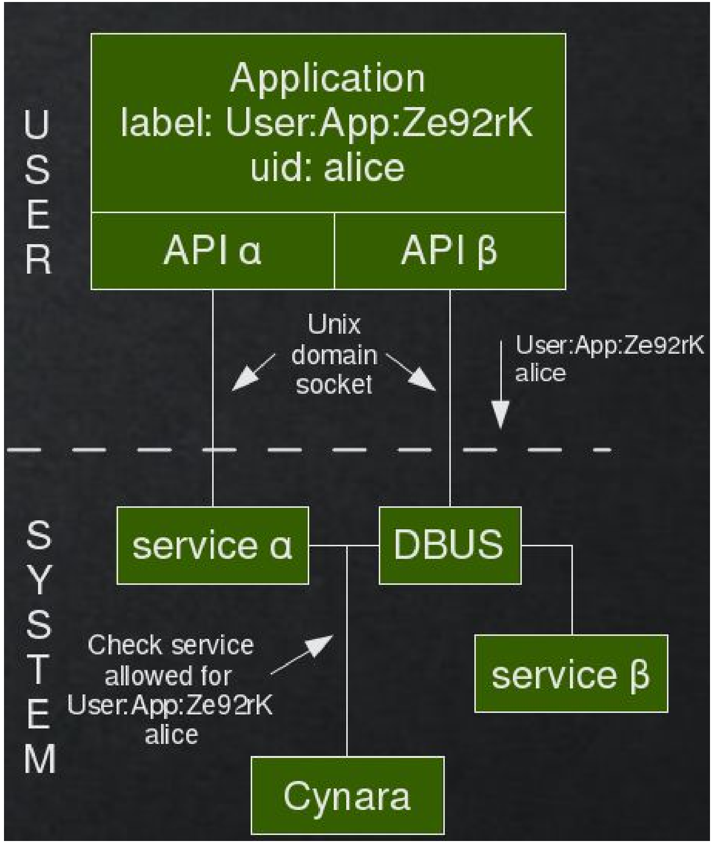
```
# chsmack --access label file
# echo -n label > /proc/$$/attr/current
```

- How to set rules? You can only reduce accesses for the current thread (inherited by cloning). But if you have the capability CAP_MAC_ADMIN, you can change all rules.

```
# echo “subject object rwt” > /sys/fs/smackfs/load-self2
# echo “subject object rwt” > /sys/fs/smackfs/load2
# echo “subject object rwt” > smackload
```

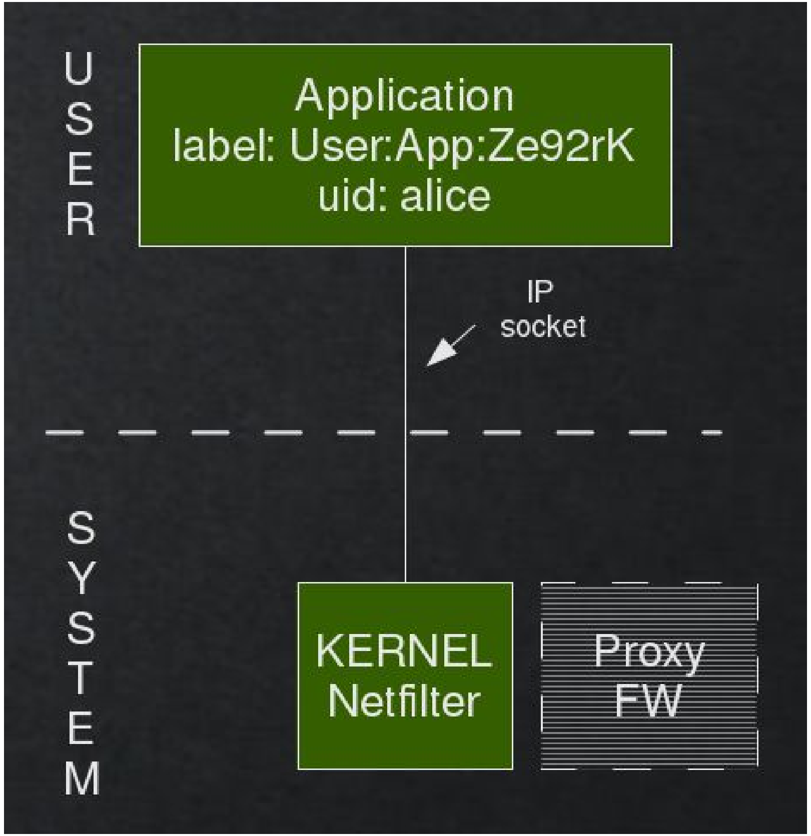
Restriction of services

- The invocations of services are using UDS
- The UDS expose the credentials of the pair: Smack label, uid, pid
- Before servicing, the service ask cynara for the authorisation using the smack label, the uid and some session id
- Cynara scans its database and reply
  - A fast cache is enable
  - Cynara can request user decision through HMI



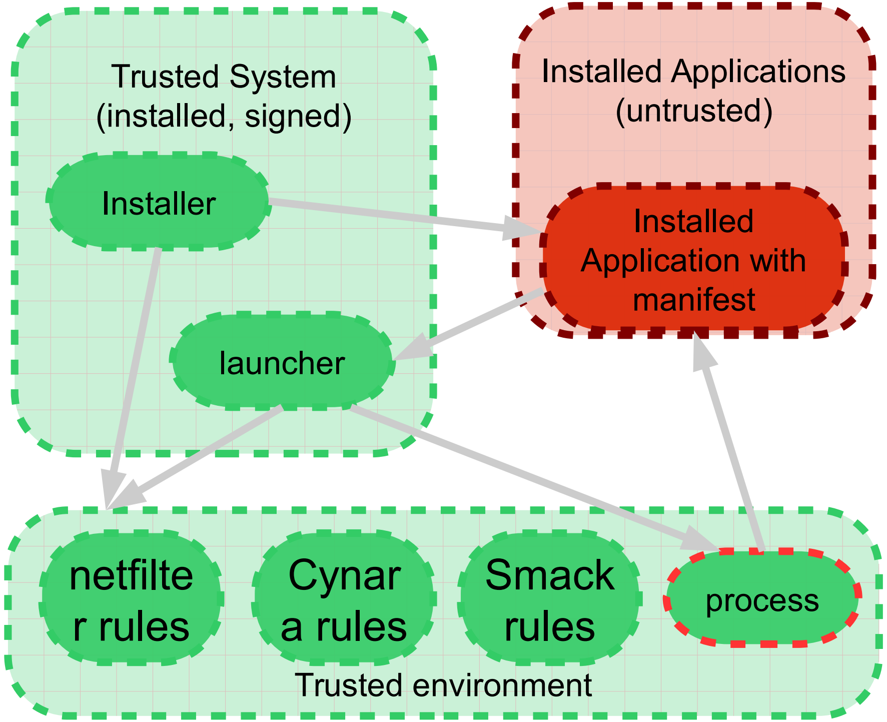
Restriction of network

- To be finalised
- Access to the network are filtered using DAC and netfilter
- A filtering proxy-firewal may be also implemented for parental control.



Application live cycle

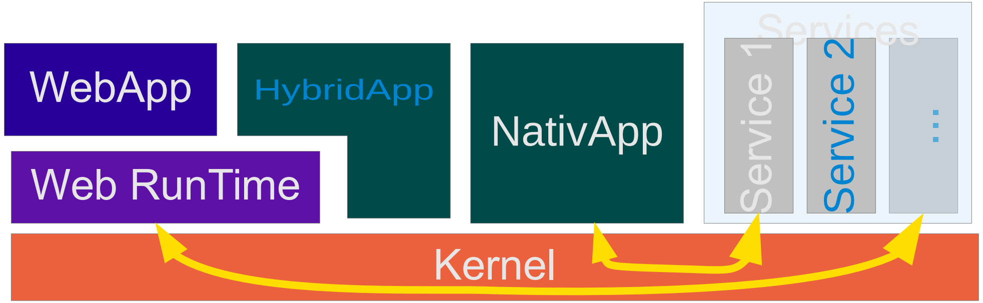
- Applications are installed by an installer
  - The installer enable the application, configure the system according to the manifest.
- Applications are launched by a launcher
  - The launcher prepare the environment in agreement with the manifest and launch the application in the trusted environment.



Trusted System (installed, signed)
Installer
launcher
Installed Applications (untrusted)
Installed Application with manifest
netfilter rules
Cynara rules
Smack rules
process
Trusted environment

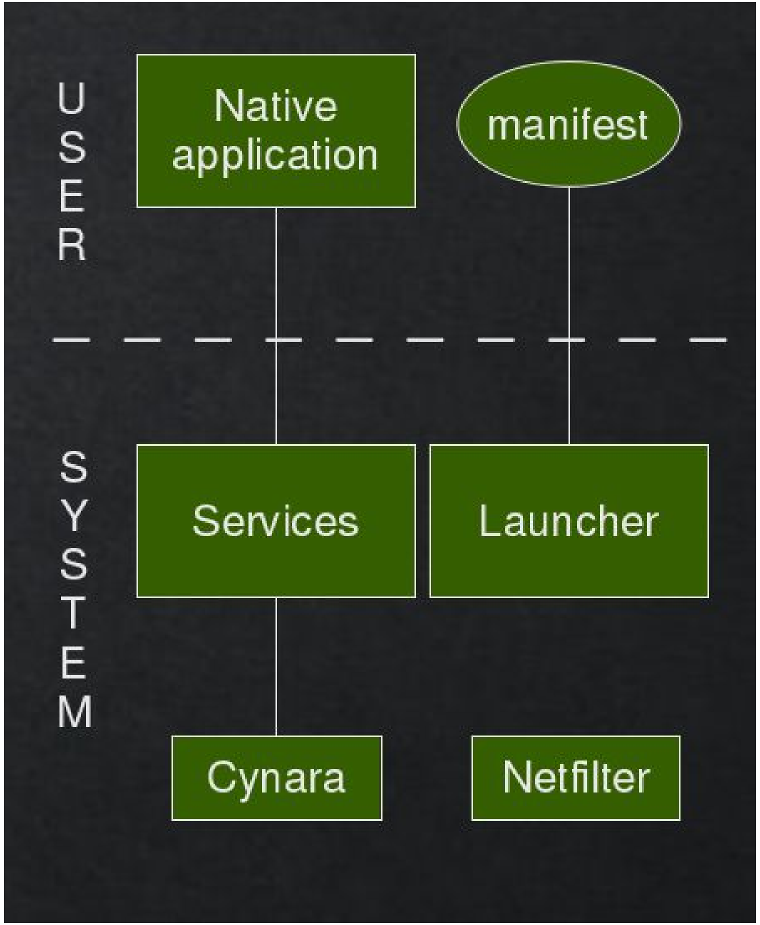
3 kinds of applications

- The web applications
  - Written in HTML5/CSS3/JAVASCRIPT
- The native applications
  - Written in any language including C/C++
- The hybrid applications
  - Mainly written in HTML5/CSS3/JAVASCRIPT
  - Includes a web runtime plugin or a some native service or application



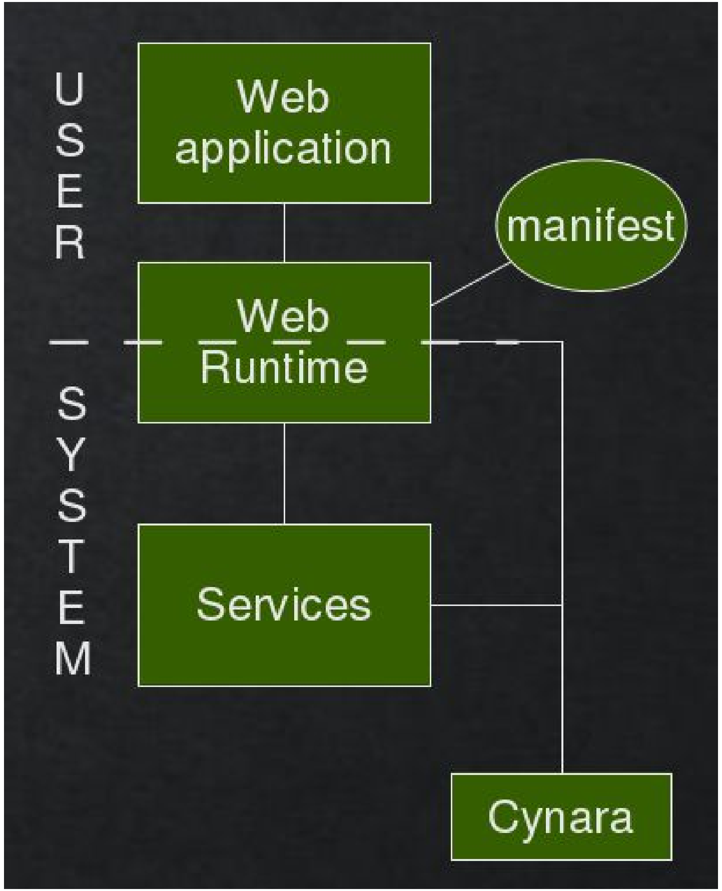
The native applications

- The applications cannot be launched directly
- The launcher is in charge of setting the runtime environment of applications
  - Specific gid
  - Netfilter data
- Services
  - D-Bus filtering
  - Service daemon



The web applications

- As natives plus:
- The Web runtime (crosswalk) is in charge of enforcing the security of the application
- Because of its model, the Web Runtime includes a trusted part (in the system space)
- The Web runtime ensure respect of the Content Security Policy (W3C)



The hybrid applications

- This applications have the two aspects of Web and natives applications.
- Their security is enforced by both:
  - Setup of the launcher
  - The Web runtime

Dominig ar Foll
Intel Open Source Technology Centre

Restriction of shared files

- Some files (like /dev/camera) are shared to users but restricted by privileges. Note that this resources can be subject to resource management (murphy)

- When no service is used as a mediator to access this ressources, then:
  - No Cynara check can be performed.
  - For this specific shared files, the access is restricted by DAC and gid to a specific group.
  - The launcher is in charge to add the group to the launched application that requires following the cynara diagnostic

How to share files?

- When files must be shared acros applications (example: an image, a pdf, a text, …) the file is copied to a directory dedicated to sharing:
  - One sharing directory per user
  - One global sharing directory
- When files must be transmitted from one user to an other, a directory specific to the destination user is used.

How applications collaborate?

- Applications sharing the same origin (as signed by a certificate) can :
  - Share some common files
  - Communicate using Message Port service

Try Tizen Meta

- HowTo
  https://wiki.tizen.org/wiki/Tizen_on_yocto

- Support
  https://lists.tizen.org/listinfo/dev

- Code
  https://review.tizen.org/gerrit/#/admin/projects/scm/bb/meta-tizen

- Bugs
  https://bugs.tizen.org/jira/browse/BTY

Q & A