

Sailfish OS

How to contribute?



Who am I?

- Marko “Sage” Saukko
 - Chief Engineer at Jolla and responsible of Hardware Adaptation team, ODM discussions, factory process, hardware adaptation architecture, ...
 - Worked for Jolla since March 2012
 - Before Jolla worked with MeeGo project 2009-2012. Part of team responsible of keeping the ARM port of MeeGo functional (Nokia N900 :))

Jolla & Sailfish OS

- Jolla Ltd. is Finnish technology company developing Sailfish OS
- In 2011 MeeGo was discontinued and the passionate team behind part of MeeGo wanted to continue that effort.



Sailfish OS

- <https://sailfishos.org/>
- Still very young operating system (~5 years)
- Mobile operating system based on Linux
- Lots of familiar open source components
 - rpm, systemd, dbus, wayland, pulseaudio, bluez, connman, ofono, ...
 - Using wayland instead of X11 compared to many desktop Linux operating systems

jolla

Some Sailfish OS milestones

- 2012 Announced Sailfish OS UI/SDK
- 2013 Jolla Phone with Sailfish OS 1.0 Beta
- 2014 Sailfish OS 1.0 and Hardware Adaptation Development Kit
- 2015 Sailfish OS 2.0 and Jolla Tablet
- 2016 Sailfish OS with Multi-SIM support
- 2016 Sailfish Community Device Program

jolla

Where can I find Sailfish OS?

- Products using Sailfish OS
 - Jolla 1
 - Jolla Tablet
 - Intex Aqua Fish
 - Jolla C
 - Turing Phone
- 40+ Community ports



jolla


SAILFISH OS

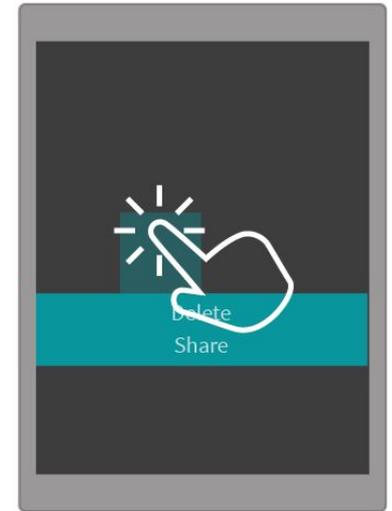
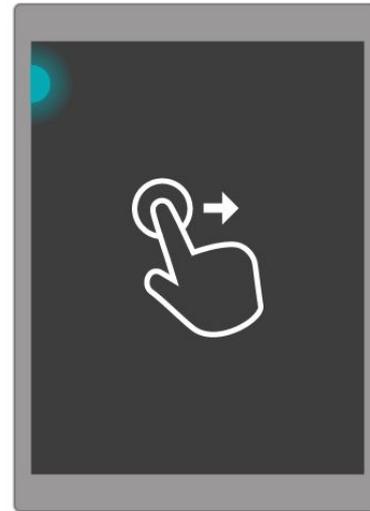
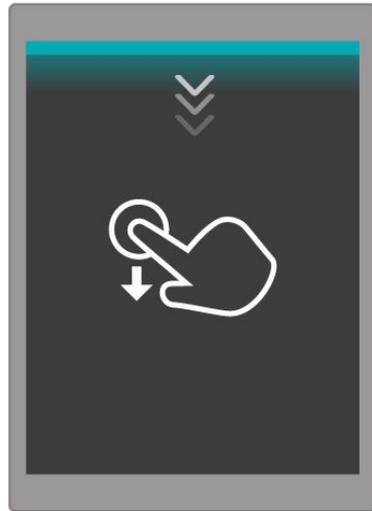
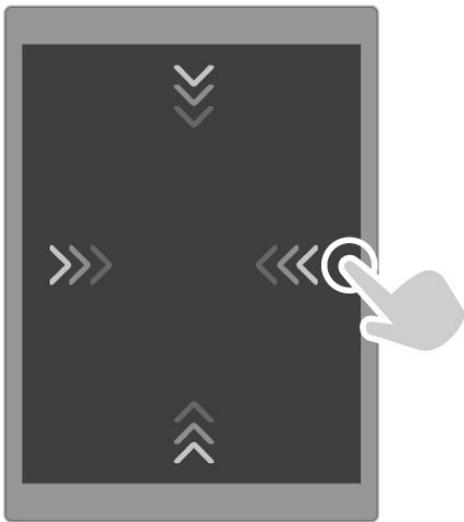
Sailfish OS some key things

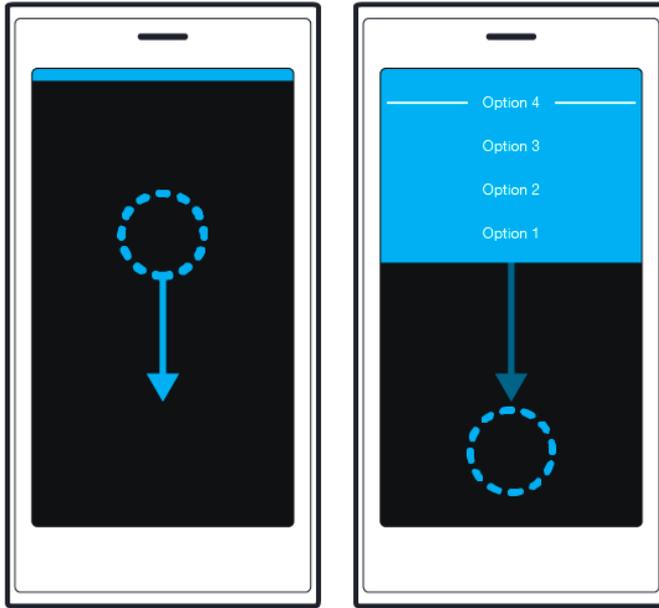
- UI written in Qt and QML
- Mostly C/C++ in the middleware
- Android support, you can run android apps without any modifications
- Compatible with hardware running Android
- Multitasking, application covers can have functionality when apps are minimized
- Gestures based operations, less buttons to press
- Easier one handed use, e.g., pull menu

jolla

Gestures

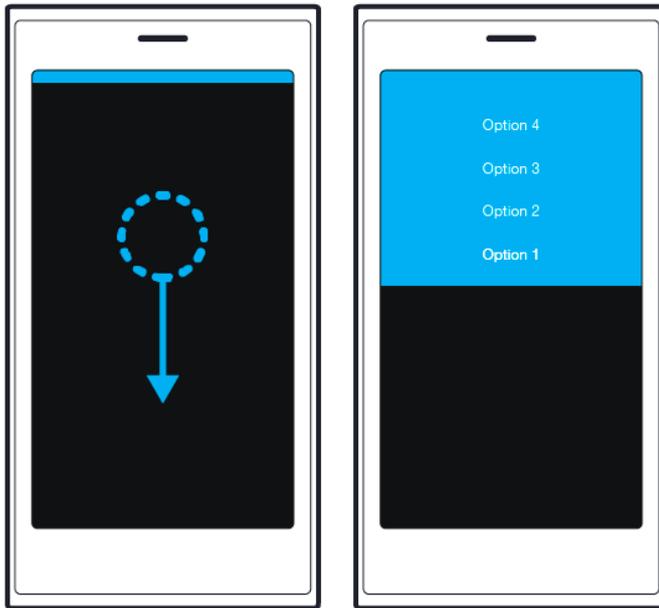
- https://sailfishos.org/wiki/User_Interface_Development
- Tap, Double tap, Edge swipe, pull menu, sub page, long-press, ...





Method A:

Move content down without releasing your finger to select an action. Release when desired action is highlighted.



Method B:

With a fast motion, pull the menu open and tap to select desired action.

Sailfish OS Architecture

jolla


SAILFISH OS

Sailfish OS Architecture

Customer configuration

More info on next slide(s)

Customer Configuration API

Sailfish OS

Sailfish Home/UX

Sailfish Silica, Lock screen, Home screen, Switcher, Launcher, Eventsview, ...

Sailfish Applications

Phone, Messages, People, E-Mail, Browser, Clock, Calendar, Camera,...

Sailfish OS Core

Multimedia (gstreamer), Virtual Keyboard (maliit), Qt5, Connectivity (connman, ofono, bluez), System libraries (systemd), Graphics (wayland), Essentials (glibc, dbus, dconf), Software management (rpm, libzypp, packagekit), ...

Sailfish OS Additional
Features
More info on next slide(s)

Publicly available hardware adaptation API that follows closely standard Linux

Hardware Adaptation

More info on next slide(s)



Customer configuration

Super app(s)

Preinstalled apps: Sailfish,
Android (*.apk)

Configurations/informations:
e.g. default timezone,
certification data (SAR values),
default browser search engine,
default photo format and
size, ...

Ambiences
(background
pictures,
sounds,...)

Preloaded content: Pictures,
Videos, Browser bookmarks,...

Customer Configuration API

Sailfish OS Additional Features

Sailfish OS Additional Features

Android support

Microsoft Exchange support

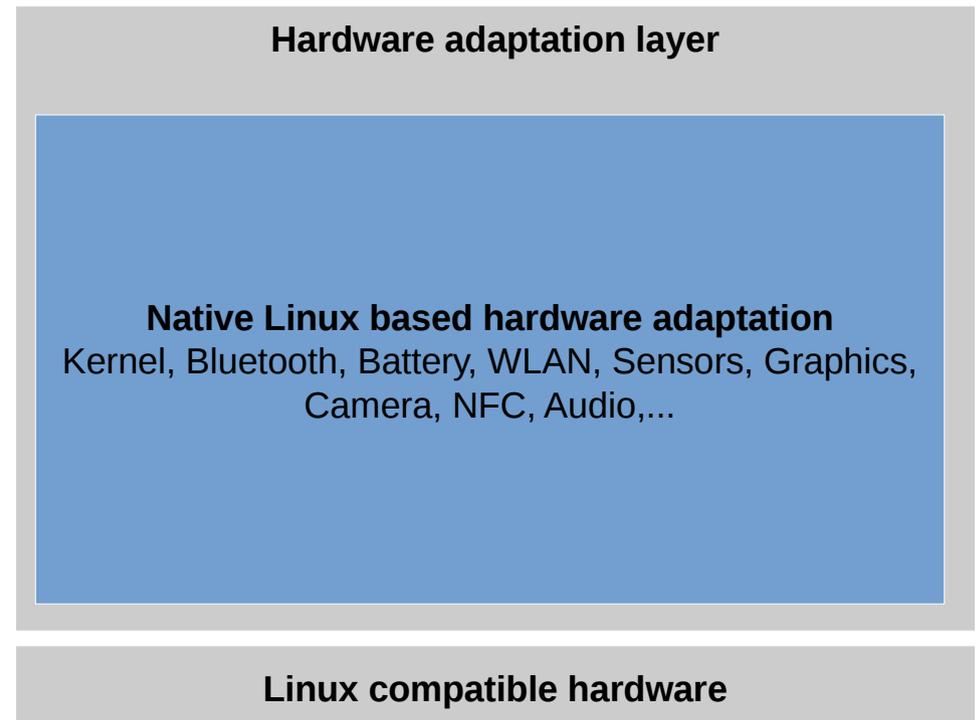
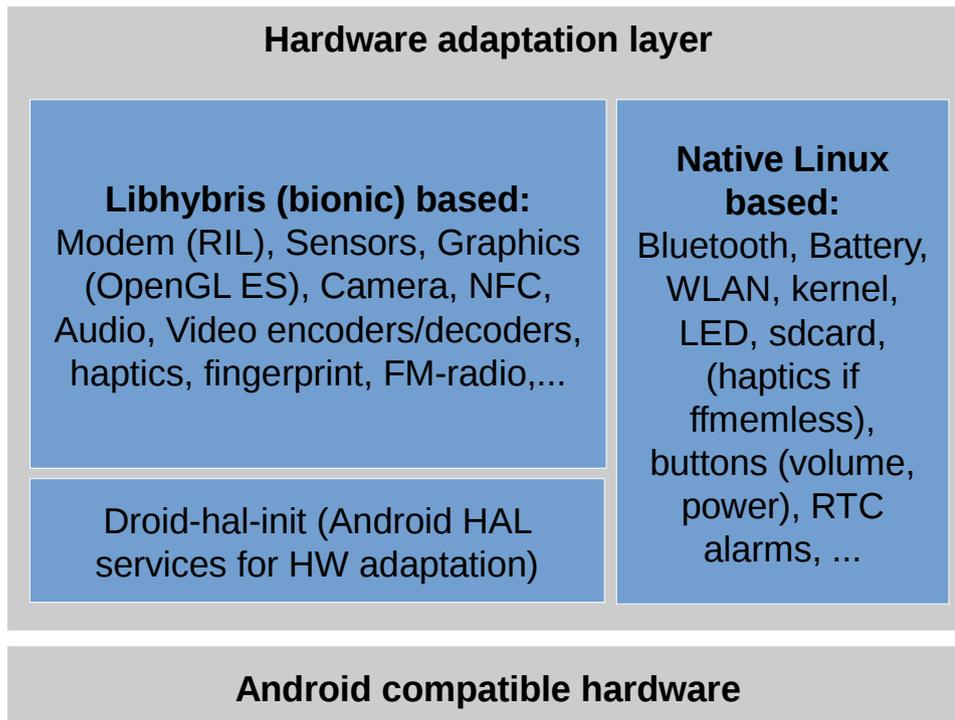
Location services by HERE.

Nuance (XT9) – text prediction
and correction



Hardware Adaptation

Publicly available hardware adaptation API that follows closely standard Linux



libhybris?

- What is libhybris?
 - Hybris loads "Android libraries, and overrides some symbols from bionic with glibc" calls, making it possible to use Bionic-based software, such as binary-only Android drivers, on glibc-based Linux distributions.
 - <http://mer-project.blogspot.fi/2013/04/wayland-utilizing-android-gpu-drivers.html>
 - <https://github.com/libhybris/libhybris>
- Why it is used?
 - Proprietary drivers = binary blobs
 - Chipset/device manufacturers do not want to do extra effort to create drivers for new platforms
 - Saves on maintenance burden to maintain new set of drivers for new platform
 - Alternative is to have native port, which is actually what our VM is doing in the Sailfish OS Application SDK
 - Pure ports, i.e., using native drivers instead of android ones. e.g., Raspberry Pi 2 has one(?)
 - https://sailfishos.org/wiki/Application_Development
 - Who uses libhybris?
 - Sailfish OS, Ubuntu Touch, Open webOS, AsteroidOS, ...



Different ways of contributing to Sailfish OS



How to contribute

- Discuss with community at <https://together.jolla.com/>
 - Report issues, do feature requests, discuss about Sailfish OS, ...
- Help doing translations at <https://translate.sailfishos.org/>
 - 40+ languages are translated for multiple releases during a year, that include new features, ...
 - https://sailfishos.org/wiki/Translate_the_OS
- Create apps and submit those to <https://harbour.jolla.com/>
 - https://sailfishos.org/wiki/Application_Development
- Help us to improve the Sailfish OS Core
 - Core consists of 500+ components, help us to maintain the packages
 - https://sailfishos.org/wiki/Platform_Development
 - **DEMO:** Live coding demo how to do core platform contribution coming up a bit later
- Hardware adaptations
 - <https://sailfishos.org/develop/hadk/>
 - <https://wiki.merproject.org/wiki/Adaptations/libhybris>



Contributing to the Sailfish OS Core



Things you should familiarize yourself

- Linux in general
- rpm packaging
- git
- Sailfish OS

Sailfish OS Core contribute steps

- https://sailfishos.org/wiki/Platform_Development
- Quick summary of the steps
 - Install Platform SDK
 - Install SB2 target, for building the components
 - Find package to contribute and create task (bug) for that
 - Clone git tree
 - Making modification
 - Building a package
 - Verifying the package
 - Creating pull request
 - Wait for review

Different parts of the Platform SDK for Sailfish OS Core package development

Host OS (Fedora/Ubuntu/OpenSUSE/....)

One can also install Linux to VirtualMachine on Windows or Mac OS if native Linux PC not available

Platform SDK (https://sailfishos.org/wiki/Platform_SDK)

SB2 target

which is specific for each device, if you do hardware adaptation development, because we install device specific headers inside it. If you do platform development you can do it inside any sb2 target really

SB2 uses QEMU to build for different architectures.

Used to package:

- The HABUILD output to rpm's

Used to build:

- Hardware adaptation middleware packages, such as, libhybris, sensorfw module, geoclue module, ...
- Sailfish OS core packages or basically any package

DEMO: Live Platform SDK demo

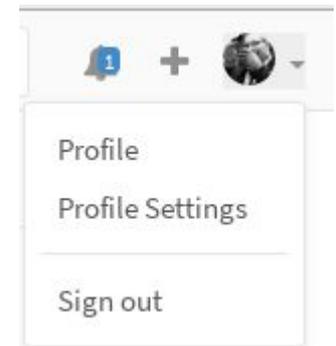


What is tar_git packaging?

- tar_git, all lines starting with [something] are taken from git commit messages to .changes file.
 - <https://git.merproject.org/mer-core/mce/commit/ea655bf14af7464ac9836f29eed492dfcf02559f>
 - It does not need to be first line
 - There can be multiple lines in one commit message
 - Line should always have bug reference, e.g., MER#
- Tag defines when the changelog is taken
 - Changelog is generated from commit messages that are between two tags
 - One bug reference between two tags required for promotions
- If there is .changes file in rpm/ directory that is added to the end of .changes file generated by tar_git
- No changes should be done to .changes files anymore. .changes files kept only for history reasons

Packaging demo 1: get required accounts

- Creating and setting up the required accounts
 - Create account to <https://bugs.merproject.org/createaccount.cgi>
 - NOTE: Currently requires contacting to lbt or sage at #merproject IRC channel @ Freenode
 - Login to https://git.merproject.org/users/sign_in
 - NOTE: You can also browse the content without login by going to <https://git.merproject.org/explore>
 - Upload you SSH key: Logo > Profile Settings > SSH Keys
- Search for the package you want to contribute to.
 - Search a package that you want to update. In this example we use: freetype
 - Fork the selected package Fork > Select your account to fork to
 - NOTE: At times the fork page gets stuck, but the process gets done so go manually to the URL
- Create a bug to which we can refer to while working <https://bugs.merproject.org/>
 - Search for existing bug or ...
 - ... create new bug https://bugs.merproject.org/enter_bug.cgi
 - Select Mer Core: which you can also find as a group prefix in the git.merproject.org component that was selected
 - Select component, which is actually the package name in this case
 - Write summary and description e.g. Update Freetype to version 2.6.5
 - Get bugzilla number that we can use and refer to with MER#xyz



Packaging demo 1: Getting the code

```
# Clone the git tree with the url in the fork
git clone --recursive ssh://git@git.merproject.org:2222/sage/freetype.git

cd freetype

git checkout -b merXYZ # Create new branch

cd freetype # Go to the submodule

git checkout master

git pull

git checkout VER-2-7 # Package specific new tag from upstream, in this case VER-2-7

cd ..

# Edit the required files
vim rpm/*.spec

mb2 -t SailfishOS-armv7hl build

# See first errors and start fixing those.
# NOTE: change Version and check if there are patches as those need to be applied manually

# Commit your changes
git commit -a -s -m "Update freetype to version 2.7
[packaging] Update to version 2.7. Fixes MER#XYZ"

# Push changes to your fork
git push origin merXYZ
```



Handling patches in rpm/ dir with tar_git

```
# In case one has patches in rpm/ directory, those can be  
# easily applied with:  
mb2 apply
```

```
# If you want to revert the patches you can call:  
mb2 -R apply
```

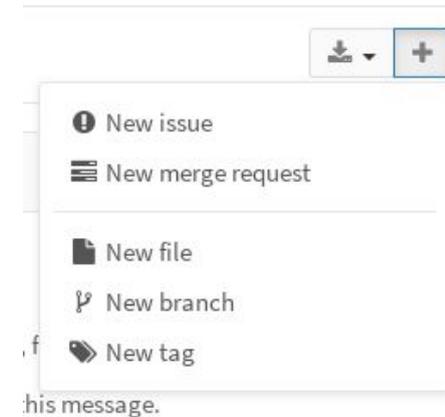
```
# Patches can be generated like shown in previous live example
```



DEMO: Install the created rpm

Packaging demo 1: Create the merge request

- User the link from the git push given by gitlab ...
- ... or go to your fork, e.g., <https://git.merproject.org/sage/freetype>
 - Click the + in the page and then New merge request
 - Select your source branch and click ***compare branches and continue***
 - Next review the changes and if everything is ok click ***Submit merge request***



Packaging Demo 2

- **NOTE:** Demo 2 was skipped during the presentation, as we ran out of time.
- Package that has not been converted to tar_git format is called “dumb package format”
 - Generating dumb packages should be avoided
- Example packages: alsa-utils, alsa-plugins, alsa-lib, desktop-file-utils, ...
 - <https://git.merproject.org/mer-core/alsa-lib/tree/master>
 - You can note these packages from the fact that there is no rpm directory and there is tarball (e.g., .tar.bz2) included in the git tree
- **DEMO:** Converting a dumb package to tar_git

Package Demo 2: Difference

- Search first the upstream git tree, e.g., desktop-file-utils
 - <https://www.freedesktop.org/wiki/Software/desktop-file-utils/>
 - <https://cgit.freedesktop.org/xdg/desktop-file-utils/>
- Then start conversion with your git fork
- Same accounts, Platform SDK and SB2 targets can be used as earlier example

Packaging Demo 2: Commands

```
git clone --recursive ssh://git@git.merproject.org:2222/sage/desktop-file-utils.git
cd desktop-file-utils

mkdir rpm

# Move existing packaging to rpm dir
git mv *.spec *.changes rpm/

# Drop old not used packaging
git rm *.tar.* *.yaml

# Next add submodule
git submodule add git://anongit.freedesktop.org/xdg/desktop-file-utils

# Get the version you want
cd desktop-file-utils
git checkout 0.23
cd ..

# Do the required changes
vim rpm/*.spec

# Submit the merge request as in example1
```

jolla



SAILFISH OS

How to find oldest packages on your Sailfish OS system?

```
rpm -qa | awk '{ print $0" " $0 }' | xargs printf "rpm -q --changelog %s | head -n1 | awk -F' ' '{printf(\\$5 \\\" \\\"\\$3 \\\" \\\" \\$4\\\" - \\\" )}'";echo %s\\n" | sh > packages-with-dates.txt
```

```
cat packages-with-dates.txt | sort -k 1,1n -k 2,2M
```

- **DEMO:** Show the package list
- Above cmdline will print that for you, even in your PlatformSDK or SB2 target as well as on your Sailfish OS device
- Should be noted that this prints the packages based on last modification time of the package, which does not necessarily mean the age of the package itself. This is because packaging changes can happen without actual package updates.

Packages currently needing conversion to tar_git

- Some simple ones to start with if you are interested: alsa-utils, alsa-plugins, alsa-lib, desktop-file-utils,...
- Full list, around ~170 package at: <http://piratepad.net/dumb-packages>
- NOTE: Some of these can't be converted with submodule approach as there is not git based upstream.

Sailfish OS porting to new devices where to start?



Where to start with adaptation work?

- IRC - #sailfishos-porters @ Freenode
- https://sailfishos.org/wiki/Hardware_Adaptation_Development_Kit
- Status page of some of the community adaptation
<https://wiki.merproject.org/wiki/Adaptations/libhybris>
- Generic porting requirements
 - Device running Android 4.1 (API level 16) or newer (higher is not always better)
 - <https://source.android.com/source/build-numbers.html>
 - Some source codes available for the device, such as kernel
 - Rule of thumb: If there is cyanogenmod for that device available you can most probably also do Sailfish OS port for it.
 - https://sailfishos.org/wiki/Hardware_Adaptation_Development_Kit

Smartphones

Legend: - Working (y=hack), - HAL works, not hooked up to UI/MW/init yet, - Not working, - Untested, - N/A - N/A on device.

Downloadable Image	Linux Kernel	Display		Touch	LED	Audio	NFC	Bluetooth	GSM			WLAN		GPS	Camera	Sensors				Keys		Vibra	Haptics	Power Mgmt.	RTC alarms	USB		FM Radio
									SMS	Voice	Data	Connect	Hotspot		ALS	PS	Accel.	Gyro.	Magne.	Vol+/-	Home					Net.	Charge	
OnePlus X (onyx)	3.4.0	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y
Fairphone 2 (fp2)	3.4.0	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y
Find5 X909 (find5)	3.4.0	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	?	Y	Y	N/A
Find7 X9076 (find7s)	3.4.0	Y	Y	N	Y	Y	Y	Y	?	?	?	Y	?	Y	?	?	?	?	?	?	N/A	Y	Y	Y	?	Y	Y	N/A
Xiaomi Redmi 2 (wt88047)	3.10.49	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	?	Y	Y	Y
OnePlus One Bacon	3.4.0	Y	Y	N	Y	?	Y	Y	Y	Y	Y	?	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	?	Y	Y	N/A
Galaxy Nexus (maguro)	3.0.31	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	N/A
Galaxy Note (n7000)	3.0.64	Y	Y	?	Y	N/A	N	?	?	?	Y	?	?	?	N	Y	Y	Y	Y	Y	N	N	Y	Y	?	Y	Y	N/A
Huawei Ascend P6 (hwp6_u06)	3.0.8	Y	Y	N	N	N/A	N	N	N	N	Y	?	?	?	N	?	Y	N	?	Y	N/A	?	N/A	Y	?	Y	Y	N/A
Nexus One (passion)	2.6.38	Y	Y	?	?	N/A	?	?	?	?	Y	?	?	?	?	?	?	?	?	?	?	N	N	N	N	Y	Y	N/A
Nexus 4 (mako)	3.4.0	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	N	Y	Y	N/A
Nexus 5 (hammerhead)	3.4.0	Y	Y	Y	Y	?	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	N	Y	Y	N/A
HTC One (m7)	3.4.82	Y	Y	Y	Y	?	?	?	Y	N	Y	Y	N	?	N	Y	N	Y	Y	Y	?	Y	Y	Y	N	Y	Y	?
SGSIII Int'l (i9300) xda	3.0.64	Y	Y	Y	Y	N	N	Y	Y	Y	Y	?	Y	N	Y	?	Y	?	?	Y	N	Y	Y	Y	N	Y	Y	N/A
SGSIII 4G (i9305)	3.0.64	Y	Y	?	Y	N	N	Y	Y	?	Y	?	?	N	N	?	?	Y	?	?	Y	N	N	Y	Y	N	Y	N/A
Motorola Moto G 2013 (falcon)	3.4.42	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	N/A	Y	Y	Y	?	Y	Y	N/A
Motorola Moto G 2014 (titan)	3.4.42	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	N/A	Y	Y	Y	N/A	Y	Y	Y
Motorola Photon Q (xt897)	3.0.93	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	QWERTY	Y	?	Y	?	Y	Y	N/A
HTC Desire HD (ace)	3.0.101	Y	Y	Y	Y	N/A	N	Y	Y	Y	Y	?	Y	N	Y	?	Y	N/A	Y	Y	N	Y	Y	Y	N	Y	Y	N/A
HTC Desire Z (vision)	3.0.101	Y	Y	Y	Y	N/A	N	Y	Y	Y	Y	?	Y	N	Y	?	Y	Y	Y	Y	QWERTY	Y	Y	N	N	Y	Y	N/A
HTC One Sprint (m7spr)	3.4.82	Y	Y	?	N	?	N	N	N	N	Y	?	?	N	Y	?	Y	Y	Y	Y	?	Y	Y	Y	?	Y	Y	?
HTC Sensation (pyramid)	3.0.85 / 3.4.112	Y	Y	?	?	N/A	?	?	?	?	Y	?	?	N	?	?	?	?	?	Y	N	Y	Y	?	?	Y	Y	N/A
Sony Xperia L C2105	3.4.0	Y	Y	N	Y	N	Y	Y	Y	Y	Y	?	N	N	Y	Y	Y	N/A	N	Y	N/A	Y	Y	Y	N	Y	Y	N/A
Sony Xperia SP (huashan)	3.4.0	Y	Y	Y	Y	?	?	Y	Y	Y	Y	?	Y	?	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	?	Y	Y	N/A
Sony Ericsson 2011 Xperias (anzu, coconut, haide, hallon, iyokan, mango, satsuma, smultron, urushi)	3.4.0	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	?	Y	Y	Y	Y	Y	N/A	Y	Y	QWERTY	Y	Y	Y	?	Y	Y	?
Sony Xperia Z (yuga)	3.4.0	Y	Y	N	Y	?	?	N	N	N	Y	?	?	N	?	?	?	?	?	Y	N/A	Y	?	N	?	Y	Y	N/A
Sony Xperia Z1 Compact (amami)	3.4.0	N	N	N	N	N	N	N	N	N	Y	?	Y	N	N	N	N	N	N	N	N/A	Y	N	N	N	Y	Y	N/A
Sony Xperia Z2 (sinus)	3.4.0	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	Y	N/A	Y	Y	Y	N/A	Y	Y	N/A
Sony Xperia Z3 compact (aries/z3c)	3.4.0	Y	Y	Y	Y	?	?	Y	Y	Y	Y	?	?	?	?	?	?	?	?	Y	N/A	?	?	?	?	Y	Y	N/A
Huawei C8813Q / G525	3.4.98	Y	Y	Y	Y	N/A	N	N	N	N	?	?	Y	Y	Y	Y	Y	N/A	Y	Y	?	Y	Y	Y	?	Y	Y	N/A
ZTE Open C / Kis 3 (kis3)	3.4.0	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	?	Y	N/A	N/A	Y	N/A	Y	Y	Y	?	Y	Y	Y
Xiaomi Redmi 1S (armani)	3.4.0	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y
ZUK Z1 (ham)	3.4.0	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y	N/A
Samsung Galaxy S (galaxysmtd)	3.0.101-g41a833f	Y	Y	N/A	N	N/A	N	N	N	N	Y	?	N	N	?	?	?	?	?	?	Y	Y	Y	Y	?	Y	Y	N/A
Samsung s4 mini (serrano)	3.4.0	Y	Y	N/A	Y	N/A	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	N	Y	Y	?
Xiaomi Mi 3(Mi 4)cancro	3.4.0	Y	Y	Y	Y	?	Y	Y	Y	Y	Y	?	Y	Y	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	?	Y	Y	N
Samsung Galaxy S4 GT-I9505 LTE (jfltexoc)	3.4.107-cyanogenmod-gc0cdaeb	Y	Y	Y	Y	N	Y	Y	?	Y	Y	?	Y	N	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	?	Y	Y	N/A

Tablets

Legend: - Working (y=hack), - HAL works, not hooked up to UI/MW/init yet, - Not working, - Untested, - N/A - N/A on device.

Downloadable Image	Linux Kernel	Display		Touch	LED	Audio	NFC	Bluetooth	GSM			WLAN		GPS	Camera	Sensors				Keys		Vibra	Haptics	Power Mgmt.	RTC alarms	USB		FM Radio
									SMS	Voice	Data	Connect	Hotspot		ALS	PS	Accel.	Gyro.	Magne.	Vol+/-	Home					Net.	Charge	
Onda V820W (inet_w)	3.10	Y	Y	N/A	Y	N/A	Y	N/A	N/A	N/A	Y	?	N/A	Y	N/A	N/A	Y	N/A	N/A	Y	Y	N/A	N/A	Y	?	Y	Y	N/A
Sony Z3 Tablet Compact LTE (scorpion)	3.4.0	Y	Y	Y	Y	?	N	?	?	Y	Y	?	Y	?	?	?	?	?	?	Y	N/A	Y	Y	Y	?	Y	Y	?
Sony Z3 Compact Tablet Wifi (scorpion_windy)	3.4.0	Y	Y	Y	Y	?	?	N/A	N/A	N/A	Y	N/A	Y	N	N	N	N	N	N	Y	N/A	Y	Y	Y	?	Y	Y	?
Nexus 7 2012 WiFi (grasper)	3.1.10	Y	Y	?	Y	Y	Y	N/A	N/A	N/A	Y	?	Y	N	Y	?	Y	Y	Y	Y	Y	N/A	N/A	N/A	Y	N	Y	N/A
Nexus 7 2012 GSM (tilapia)	3.1.10	Y	Y	?	Y	Y	Y	Y	?	?	Y	?	Y	N	Y	?	Y	Y	Y	Y	Y	N/A	N/A	N/A	Y	N	Y	N/A
Nexus 7 2013 WiFi (flo)	3.4.0	Y	Y	?	N	?	Y	N/A	N/A	N/A	Y	?	Y	N	?	?	?	?	?	?	Y	N/A	N/A	N/A	?	N	Y	N/A
Nexus 7 2013 GSM (deb)	3.4.0	Y	Y	?	N	?	Y	?	?	?	Y	?	Y	N	?	?	?	?	?	?	Y	N/A	N/A	N/A	?	N	Y	N/A
Galaxy Tab 2 Wifi (p3110)	3.0.31	Y	Y	?	N	N/A	N	N/A	N/A	N/A	N	N	?	?	N	?	Y	N	?	Y	N/A	N	N/A	Y	?	Y	Y	N/A
Samsung Galaxy Tab 4 10.1 Wifi (matissewifi)	3.4.0	Y	Y	N/A	Y	N/A	Y	N/A	N/A	N/A	Y	?	?	N	?	?	Y	N/A	N/A	Y	Y	N/A	N/A	Y	?	Y	Y	?
Samsung Galaxy Tab 2 3G (espresso3g)	3.0.101	Y	Y	N/A	Y	N/A	Y	Y	Y	Y	Y	?	Y	Y	Y	Y	N/A	Y	N	Y	Y	N/A	Y	Y	Y	?	Y	N/A

Com
Confid

Tweet on October 27th

- <https://twitter.com/JollaHQ/status/791586085826625536>



Different parts of the Platform SDK

Host OS (Fedora/Ubuntu/OpenSUSE/....)

Platform SDK (https://sailfishos.org/wiki/Platform_SDK)

SB2 target

which is specific for each device, if you do hardware adaptation development, because we install device specific headers inside it. If you do platform development you can do it inside any sb2 target really

SB2 uses QEMU to build for different architectures.

Used to package:

- The HABUILD output to rpm's

Used to build:

- Hardware adaptation middleware packages, such as, libhybris, sensorfw module, geoclue module, ...
- Sailfish OS core packages or basically any package

HABUILD (minimal ubuntu chroot)

Used to build some of the hardware adaptation bits:

- Files for /system directory on device
- Android init files
- Kernel with modules
- boot.img

Platform SDK itself is used to create the repository that is used when building the device image with mic.

Porting process steps

1. Install Platform SDK, SB2 target into it and HABUILD_SDK (ubuntu chroot)
2. Build subset of android with hybris patches in HABUILD_SDK
3. Package the android built into rpm's in SB2
4. Install the resulting rpm's (e.g. headers for development) to SB2 target
5. Build libhybris and other Sailfish OS adaptation packages against the droid headers installed previously
6. Package the resulting components compiled in SB2 target
7. Create a local repository with the packages
8. Create .ks file for building an image
9. Create an image for the device – downloads the required packages from repositories
10. Install the Sailfish OS image to the device
11. Get the first black screen ;)
12. Then start iterating...



Sailfish OS boot order

- System-on-Chip (SoC)
- 1st Bootloader
- 2nd Bootloader – Aboot – little kernel (lk)
- Kernel + initrd
- /sbin/preinit
- Systemd - /bin/init
 - Starts executing the Sailfish OS system components parallel
 - Loads android init in very early phase
- **DEMO:** Bootchart, how to see what happens in the bootup

Cheat Sheet

- Sailfish OS is very similar to standard Linux, but there are still some parts that are sailfish OS generic. Thus there is cheat sheet collected that contains the commands for the commonly needed things during development
- https://sailfishos.org/wiki/Sailfish_OS_Cheat_Sheet

Follow us at social media

- <https://blog.jolla.com/>
- <https://twitter.com/JollaHQ>
- <https://www.facebook.com/jollaofficial/>
- <https://plus.google.com/+Jolla>
- <https://www.linkedin.com/company/jolla>
- <https://www.youtube.com/c/jolla>
- <https://identi.ca/jolla>
- <https://joindiaspora.com/u/jolla>



Thank you!

jolla


SAILFISH OS