Upcoming Technologies: oFono

Marcel Holtmann
Intel Open Source Technology Center

Happy Anniversary

Three years of Open Source Telephony

big THANK YOU to everybody

Aki Niemi
Alexander Kanavin
Amit Mendapara
Anand Ramakrishna
Anders Gustafsson
Anderson Briglia
Anderson Lizardo
Andras Domokos
Andres Salomon
Andrzej Zaborowski
Antoine Reversat
Antti Paila
Arun Ravindran
Benoît Monin
Bernhard Guillon
Bertrand Aygon
Caiwen Zhang
Carlos Pargada
Christian Lam
Christophe Guiraud
Christopher Vogl
Claudio Takahasi
Daniel Orstadius
Daniel Wagner
Dara Spieker-Doyle
Denis Kenzior
Faiyaz Baxamusa
Florian Steinel
Forrest Zhao
Frank Gau
Frédéric Dalleau
Frédéric Danis
George Matveev
Grant Erickson
Guillaume Lucas
Guillaume Zajac
Gustavo F. Padovan
Hannu Lyytinen
Helen Clemson
Iiro Kaihlaniemi
Inaky Perez-Gonzalez
Ismo Puustinen
Jan Luebbe
Jarkko Lehtoranta
Jarko Poutiainen
Jeevaka Badrappan
Jessica Nilsson
João Paulo Rechi Vita
Johan Hedberg
John Mathew
Jukka Rissanen
Jukka Saunamaki
Jussi Kangas
Jussi Kukkonen
Kai Vehmanen
Kalle Valo
Konrad Slowinski
Kristen Carlson Accardi
Lasse Kunnasluoto
Lei Yu
Lucas De Marchi
Luiz Augusto von Dentz
Marcel Holtmann
Mario Tokarz
Marit Henriksen
Marko Saukko
Martin Xu
Matthias Günther
Michael Brudevold
Michael Dietrich
Michael Schloh von Bennewitz
Miia Leinonen
Mika Liljeberg
Mikel Astiz
Minjun Li
Naresh Mehta
Neil Jerram
Nicolas Bertrand
Oleg Zhurakivskyy
Olivier Guiter
Olivier Le Thanh Duong
Oskari Timperi
Paavo Leinonen
Pasi Miettinen
Patrick Porlan
Patrik Flykt
Pekka Pessi
Petteri Tikander
Philippe Nunes
Rafael Ignacio Zurita
Rajesh Kadhiravan Nagaiah
Rajyalakshmi Bommaraju
Regis Merlino
Rémi Denis-Courmont
Renat Zaripov
RISKÓ Gergely
Rolf Offermanns
Ronald Tessier
Ryan M. Raasch
Ryan Raasch
Samuel Ortiz
Santiago Carot-Nemesio
Santtu Lakkala
Sébastien Bianti
Shane Bryan
Simon Lethbridge
Sjur Brændeland
Syam Sidhardhan
Szymon Janc
Thadeu Lima de Souza Cascardo
Tomasz Bursztyka
Tomasz Gregorek
Torgny Johansson
Vinicius Costa Gomes
Yang Gu
Zhenhua Zhang
Zhigang Li

day 1 – every start is hard

first email ...

To: ofono@ofono.org
Subject: waste of resources...?
Date: Mon, 11 May 2009 23:51:41 +0200

Hi ofono,

after many, many other initiatives of free, open mobile development platforms that do not fit the communities requirements, why again starting a new one?

why don't u use already given apis and platforms, help development of those and enhance them?

why not use for example pyneo? it brings a clean modern design not only for telephony but for all needs of mobile devices and a reference implementation is in place...

best regards,
michael

... and now

pyneo-pygsmd - the gsm daemon
(obsolete, use ofono instead!)

irrel.deviantart.com

What is oFono actually about?

Answer: Simplicity

Really simple APIs

Design driven by use cases

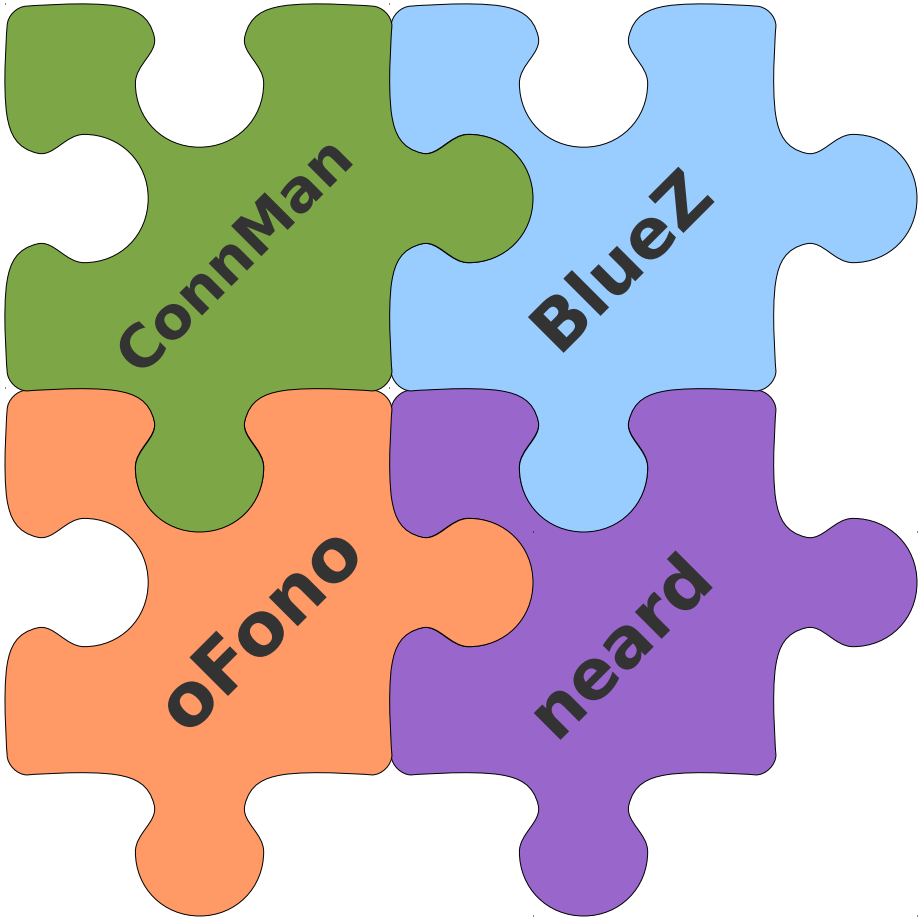
Easy to add new hardware support

Based on standards

KISS principle

Fully integrated solution

Manufactures can focus on <u>user experience</u>



out-of-the-box modem support

IMC/Infineon
ST-Ericsson
Nokia ISI
TI Calypso
Telit
SIM Com
Ericsson MBM
Option HSO

Huawei
ZTE
Longcheer
Wavecom
Novatel Wireless
Sierra Wireless
Qualcomm Gobi
Nvidia Icera

out-of-the-box features

3GPP and 3GPP2 compliant

GSM, UMTS, CDMA and LTE

Call, SS, USSD, SIM, STK, RAT, SMS, CBS, CTM/TTY, GPRS, GNSS, NITZ, Device Info, Network registration, History, Phonebook etc.

Genivi abstract component

Generic layers for 27.007 and 27.010

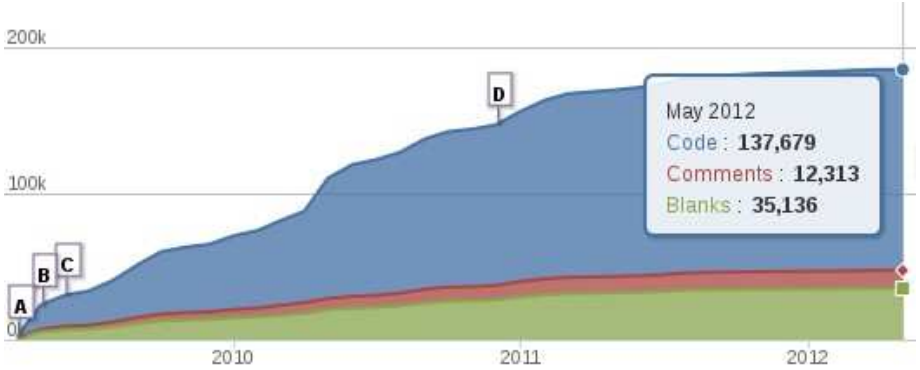
Integrated IPv4/IPv6 dual-stack
Integrated HDLC and PPP stack
Integrated AT emulator
Integrated Bluetooth HFP, DUN, SAP

Nokia's Phonet support
ST-Ericsson's CAIF support
Qualcomm's QCDM and QMI support
Infineon's RawIP support

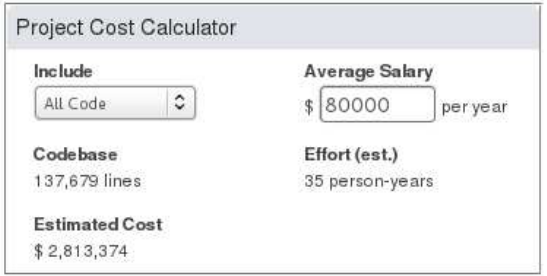
oFono core contains all telephony logic

oFono modem adaptations are tiny

oFono based applications are typically 10 times smaller



Basic COCOMO model



This is one of the largest open-source teams in the world, and is in the top 2% of all project teams on Ohloh.

Ready for phone and tablet use cases

Ready for netbook and ultrabook use cases

Ready for automotive use cases

100% open source

Large community around the project

first they ignore you

then they laugh at you

then they fight you

then you win

(unconfirmed source)

DEVELOPER
CONFERENCE
MAY 7–9, 2012

Introduction to Wayland

Jesse Barnes, Intel Open Source Technology Center

What is Wayland?

A compositing display management architecture and protocol

- Rolls window management, compositing, and the display server into a single process
  - Main open source implementation called Weston
- Does not include a rendering API
  - Clients use what they want and send buffer handles to the server
  - Current clients use Cairo, OpenGL, GLES
  - Software rendering fully supported as well through the SHM protocol
- Consolidates experience from the last decade of extending and enhancing X

Current status

Basics work today, full desktop support is close

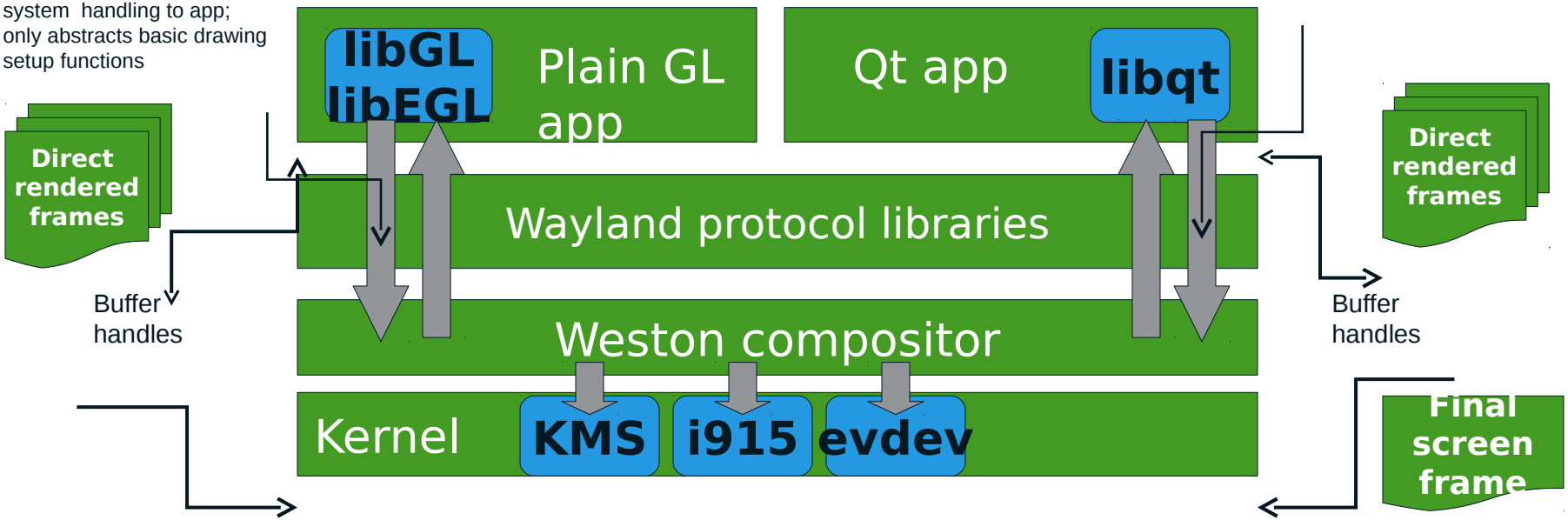
- Qt was the first toolkit; port done by Intel and Nokia for MeeGo related projects
- EGL (with OpenGL or GLES) and Cairo have been supported since the beginning
- Improved Qt support coming in Qt 5
- GTK+ work coming along nicely
  - Many apps can work without any source changes (should even be possible to avoid a recompile in some cases)
  - Client side decorations in development
  - A few missing features like drag-n-drop
- EFL in progress
  - Basic apps run, and include client side decorations
  - WebKit back-end optimization in progress
  - Media integration in progress

Track progress at http://wayland.freedesktop.org/

Wayland native applications

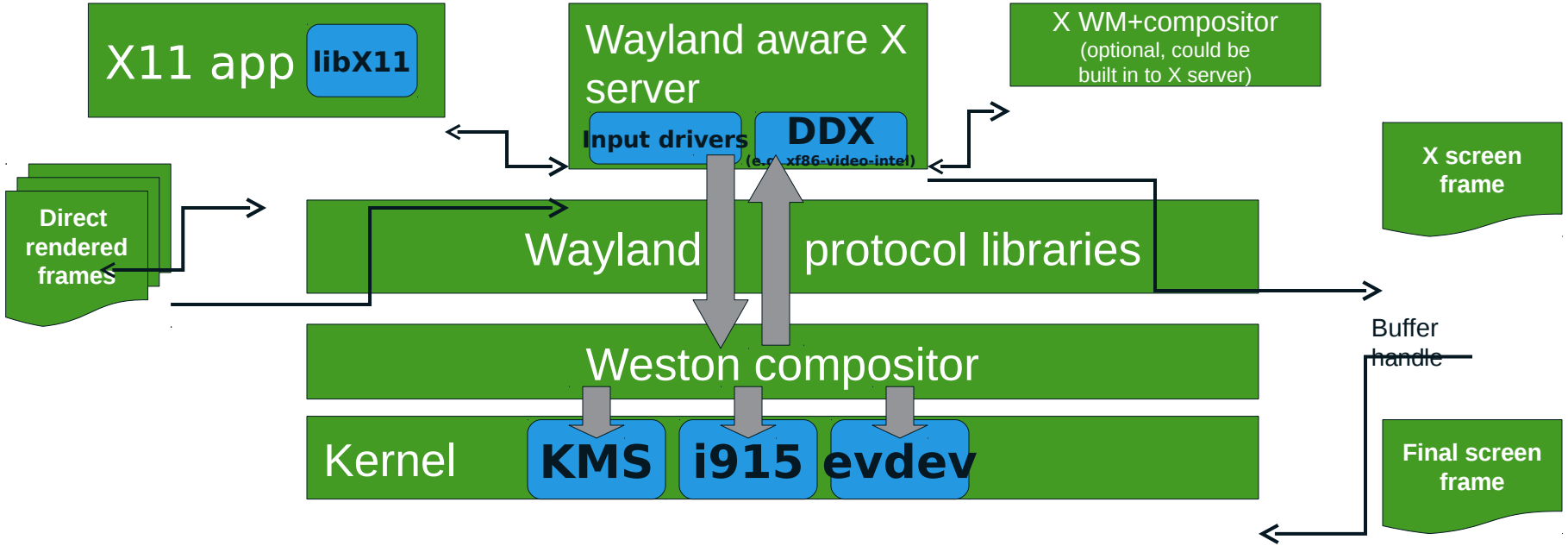
EGL leaves most window system handling to app; only abstracts basic drawing setup functions

Toolkits use window system protocol (e.g. surface allocation, window movement/resize, input handling)



- Buffer handle passing, not copying
- Toolkits generally abstract windowing system
  - Pure Qt & GTK+ apps just needs a re-compile or loading with a different window system back end
  - Plain GL apps will need some work at the EGL level

Wayland with X11 compatibility



- X server can be started on-demand when X clients connect
- “Rootless” or full screen versions of X possible (just like Mac and Windows X servers)

FAQs

- “Who are these people, why don’t they just extend X?” or “These people must not understand X.”
  - Founder Kristian Høgsberg responsible for key X improvement of the past few years: DRI2, other core contributors are/were major X contributors as well
  - X not suited by design
- “What are the platform requirements for Wayland & Weston?”
  - Mainly buffer sharing so clients can render and pass a handle to the resulting buffer to the server
  - Short story: if you have a KMS and DRI driver you probably don’t need to do much
  - Weston back ends available for DRM, OpenWF, X, and Wayland, others definitely possible
- “How is input handled?” and “Does Wayland support touch/multitouch?”
  - Wayland protocol includes input handling, Weston supports input devices through evdev, and supports multitouch including libmtdev for devices requiring it
- “What about network transparency?” or “OMG they killed Kenny!”
  - Since there is no server side rendering, the problem is much simpler
  - RDP or similar protocols can be provided by a server like Weston, and likely be more efficient than X is today

Schedule

We expect that we can release Wayland 1.0 this year:

- 0.85, developer snapshot, protocol changes planned (already out)
- 0.90, starting beta, protocol frozen
- 0.9x, release candidates
- 1.0, first stable release
  - Marks beginning of stable protocol and API
  - Not world domination
  - Somewhere in first half of 2012
  - Will include 1.0 of Weston compositor as well

DEVELOPER
CONFERENCE
MAY 7–9, 2012