FROM EMBEDDED LINUX BUILD SYSTEMS TO EDGE COMPUTING

CEZARY DYNAK

2018-10-24

CEZARY DYNAK

ABOUT ME

SOME TIME AGO...

- Wrocław University of Science and Technology
- Faculty of Electronics / W4
- Department of Cybernetics and Robotics / K7
- Laboratory 07/C-3 (diablo/panamint)
- Automatyka i Robotyka / Embedded Robotics

CURRENTLY

- Node.js developer
- IoT back-end (cloud)
- **cdynak@spyro-soft.com**

- **github.com/cdynak**
- **linkedin.com/in/cdynak**
- **last.fm/user/cezdyn**

PLAN

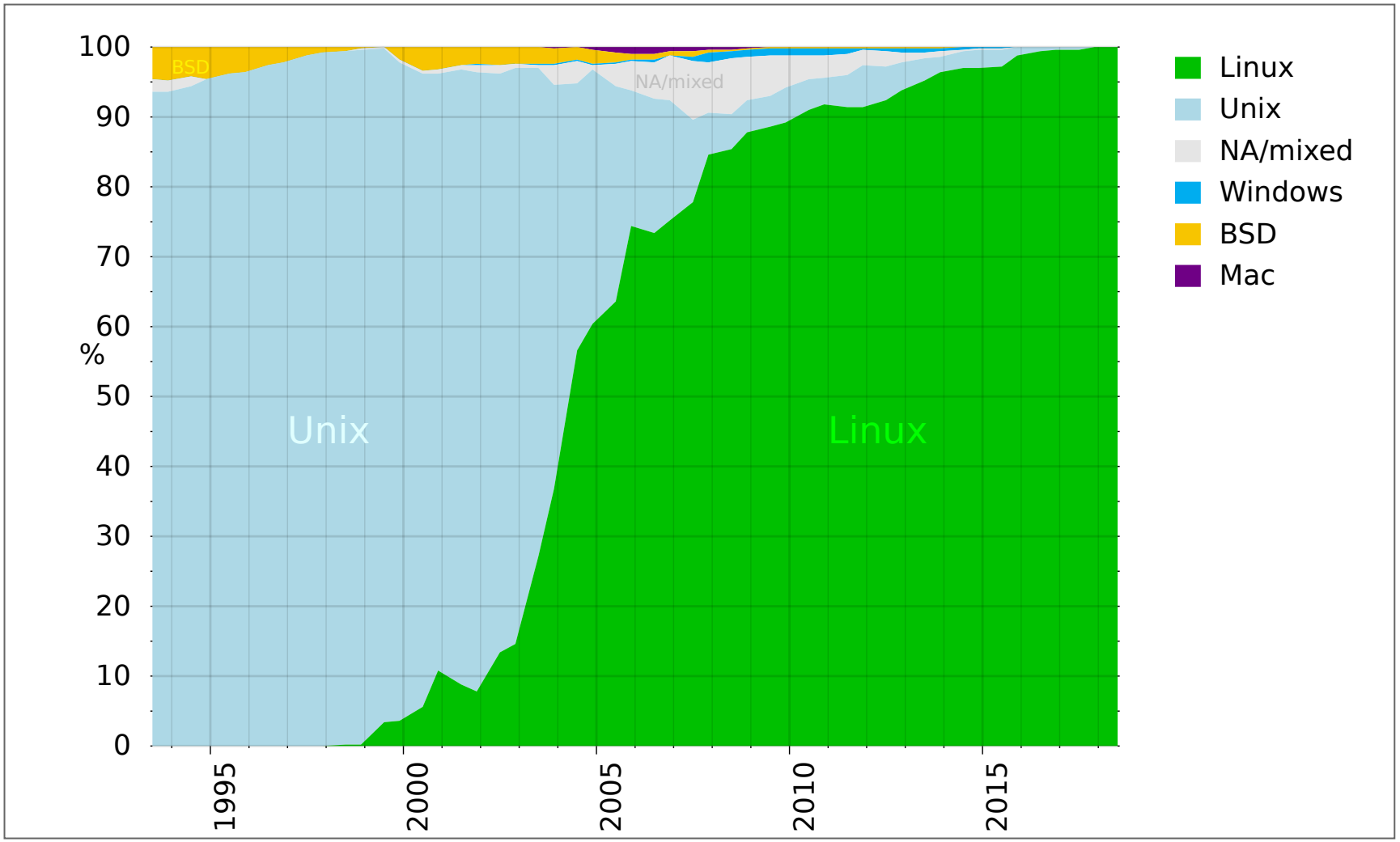
CDYNAK.GITHUB.IO/LINUX2018

- Embedded Linux build systems - comparison
  - introduction
  - definitions
  - examples
- Embedded Linux build systems - applications
  - IoT
  - cloud
  - edge

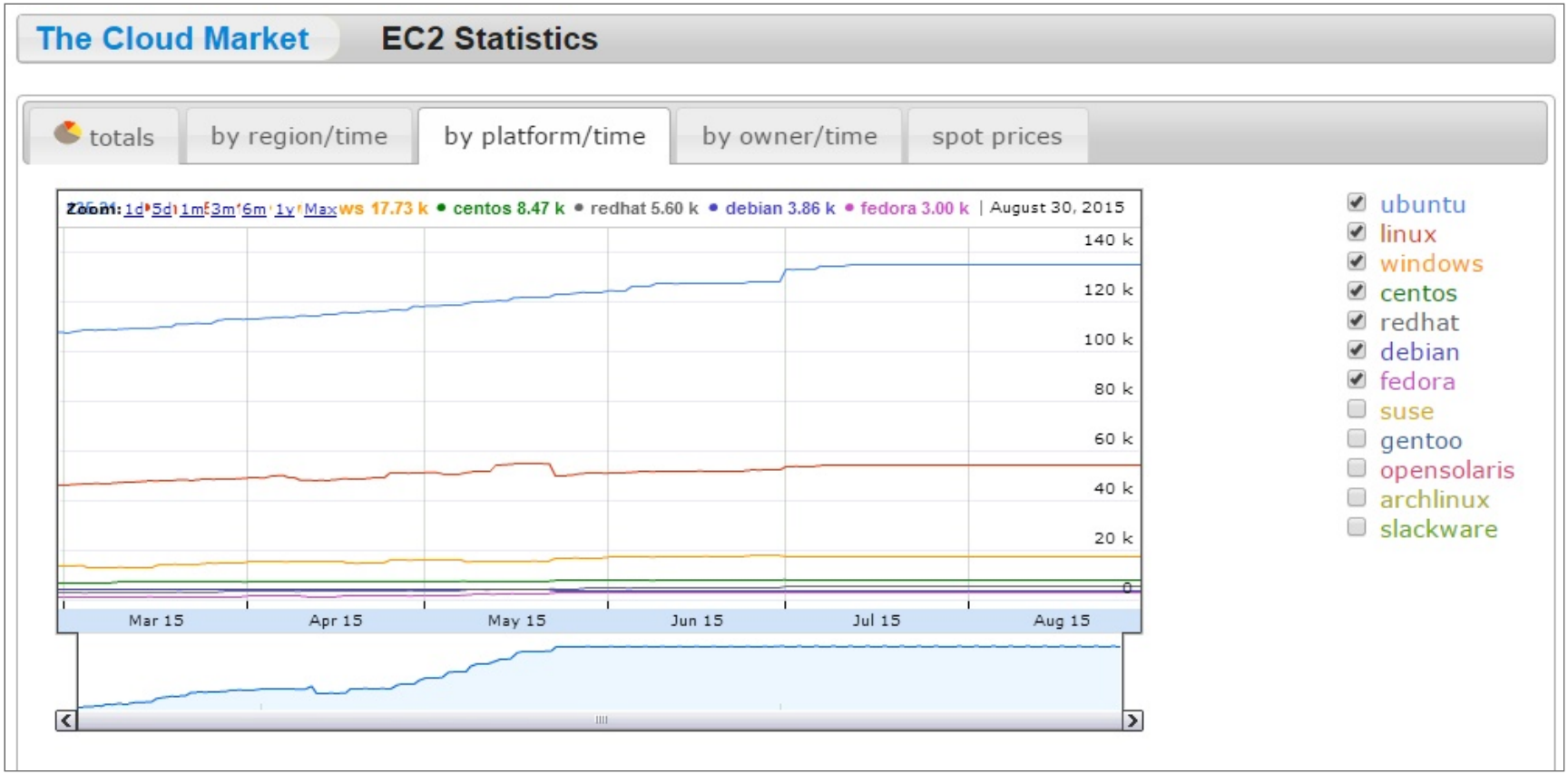
LINUX

CLOUD / DESKTOP / EMBEDDED

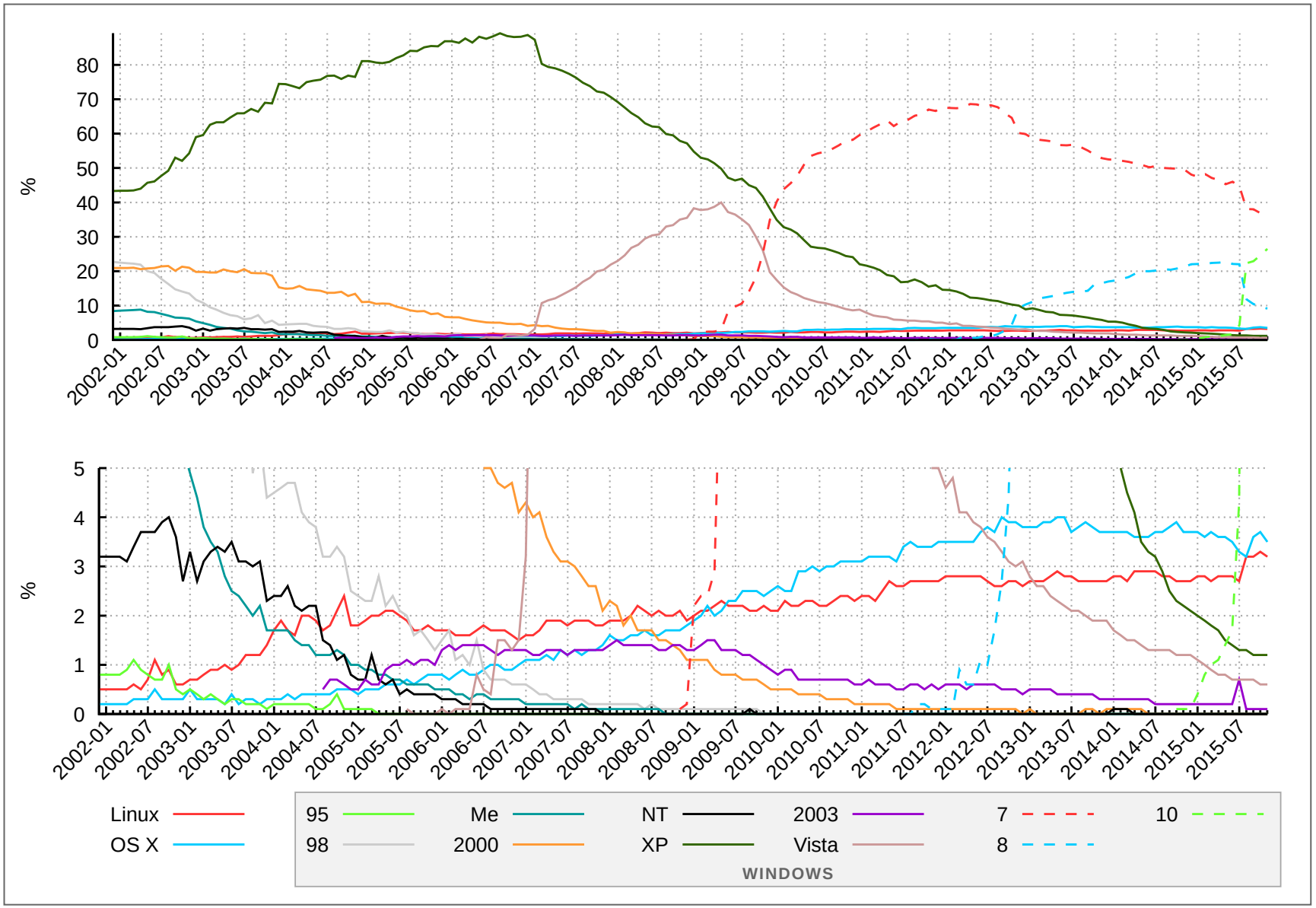
SUPERCOMPUTERS



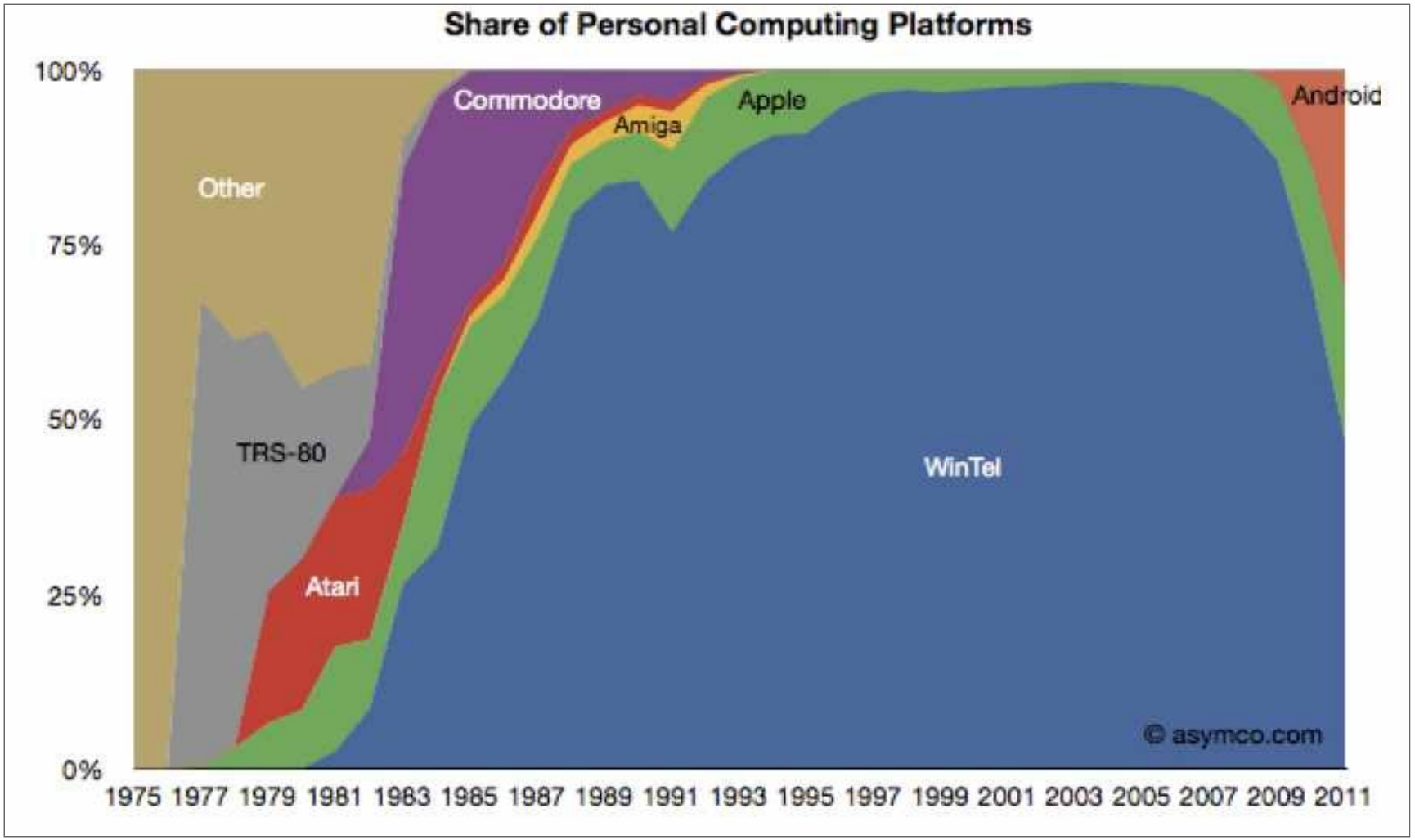
CLOUD


The Cloud Market
EC2 Statistics
totals
by region/time
by platform/time
by owner/time
spot prices
Zoom: 1d 5d 1m 3m 6m 1y Max
ws 17.73 k
centos 8.47 k
redhat 5.60 k
debian 3.86 k
fedora 3.00 k
August 30, 2015
140 k
120 k
100 k
80 k
60 k
40 k
20 k
0
Mar 15
Apr 15
May 15
Jun 15
Jul 15
Aug 15
ubuntu
linux
windows
centos
redhat
debian
fedora
suse
gentoo
opensolaris
archlinux
slackware

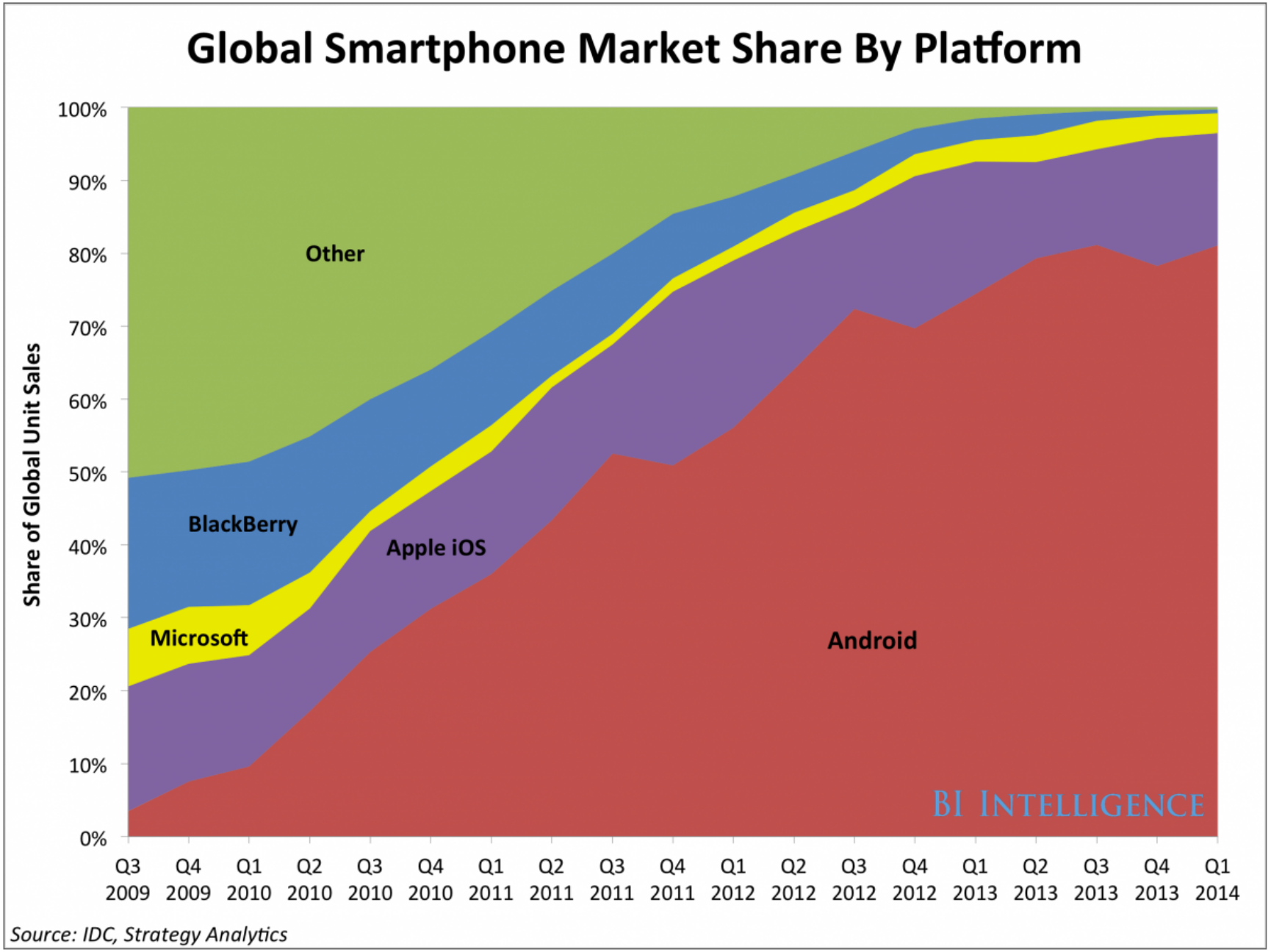
DESKTOP


%
80
70
60
50
40
30
20
10
0
2002-01
2002-07
2003-01
2003-07
2004-01
2004-07
2005-01
2005-07
2006-01
2006-07
2007-01
2007-07
2008-01
2008-07
2009-01
2009-07
2010-01
2010-07
2011-01
2011-07
2012-01
2012-07
2013-01
2013-07
2014-01
2014-07
2015-01
2015-07
%
5
4
3
2
1
0
Linux
OS X
95
98
Me
2000
NT
XP
2003
Vista
7
8
10
WINDOWS

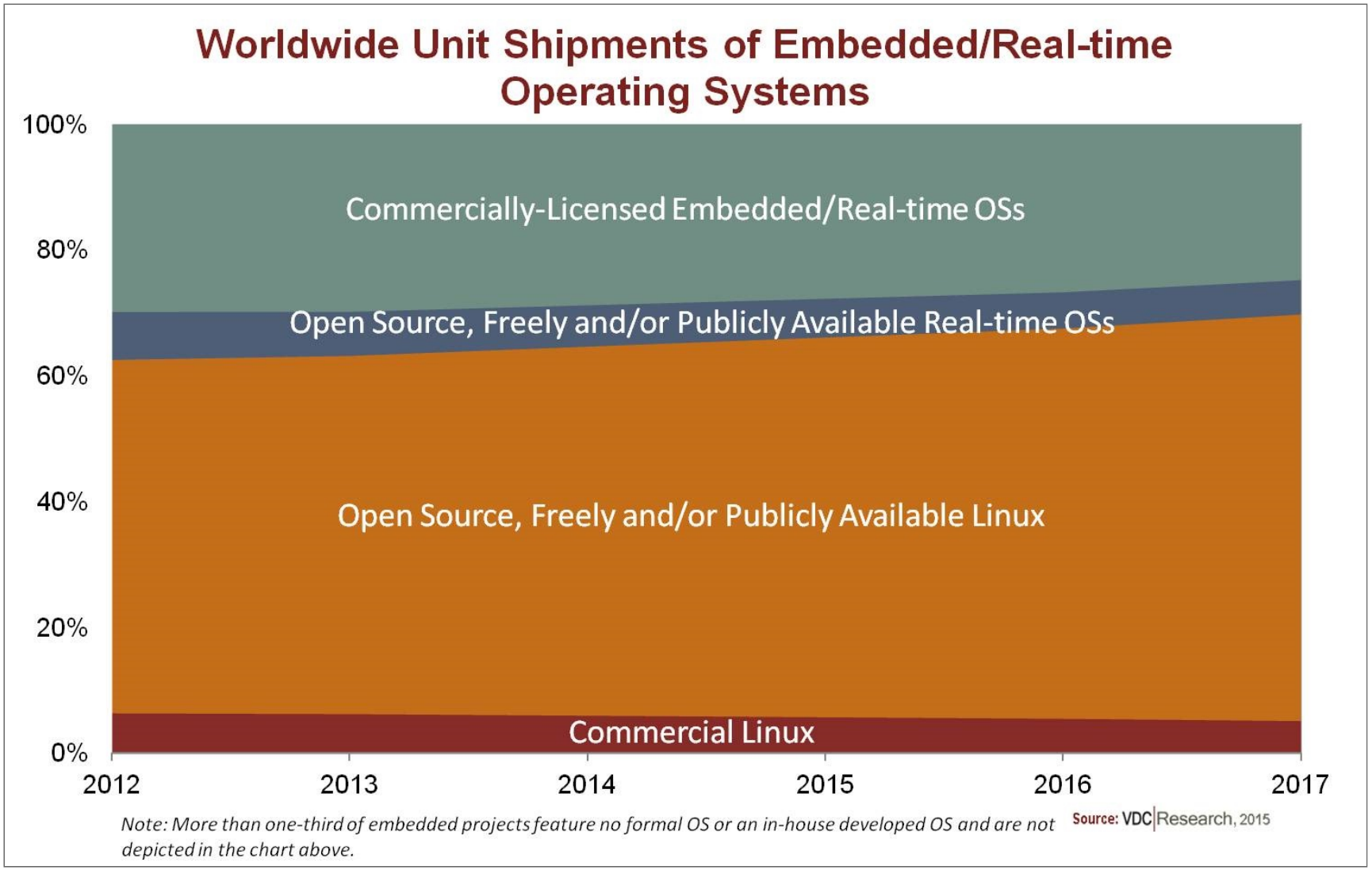
DESKTOP


Share of Personal Computing Platforms
100%
75%
50%
25%
0%
Other
TRS-80
Atari
Commodore
Amiga
Apple
WinTel
Android
© asymco.com
1975 1977 1979 1981 1983 1985 1987 1989 1991 1993 1995 1997 1999 2001 2003 2005 2007 2009 2011

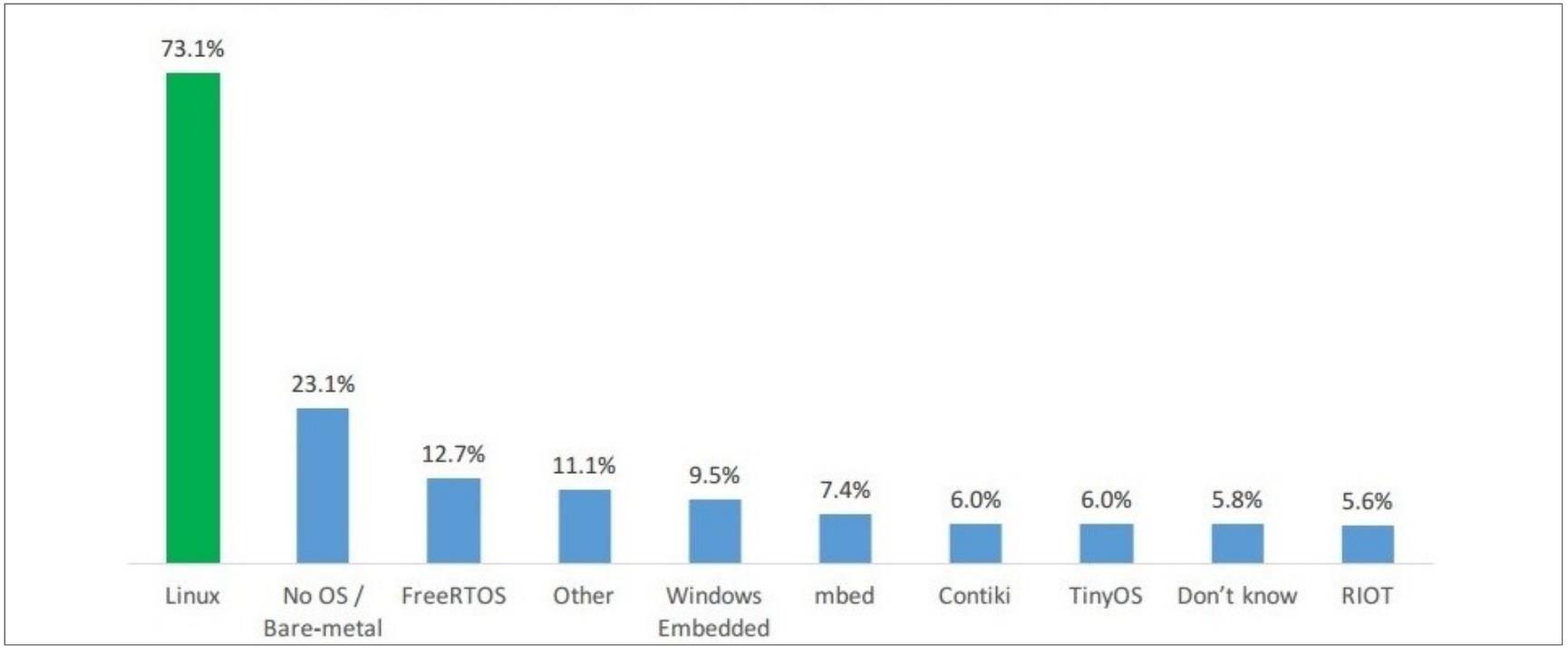
MOBILE


Global Smartphone Market Share By Platform
Share of Global Unit Sales
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Other
BlackBerry
Microsoft
Apple iOS
Android
BI INTELLIGENCE
Q3 2009
Q4 2009
Q1 2010
Q2 2010
Q3 2010
Q4 2010
Q1 2011
Q2 2011
Q3 2011
Q4 2011
Q1 2012
Q2 2012
Q3 2012
Q4 2012
Q1 2013
Q2 2013
Q3 2013
Q4 2013
Q1 2014
Source: IDC, Strategy Analytics

EMBEDDED


Worldwide Unit Shipments of Embedded/Real-time Operating Systems
100%
80%
60%
40%
20%
0%
Commercially-Licensed Embedded/Real-time OSs
Open Source, Freely and/or Publicly Available Real-time OSs
Open Source, Freely and/or Publicly Available Linux
Commercial Linux
2012
2013
2014
2015
2016
2017
Note: More than one-third of embedded projects feature no formal OS or an in-house developed OS and are not depicted in the chart above.
Source: VDC Research, 2015

EMBEDDED

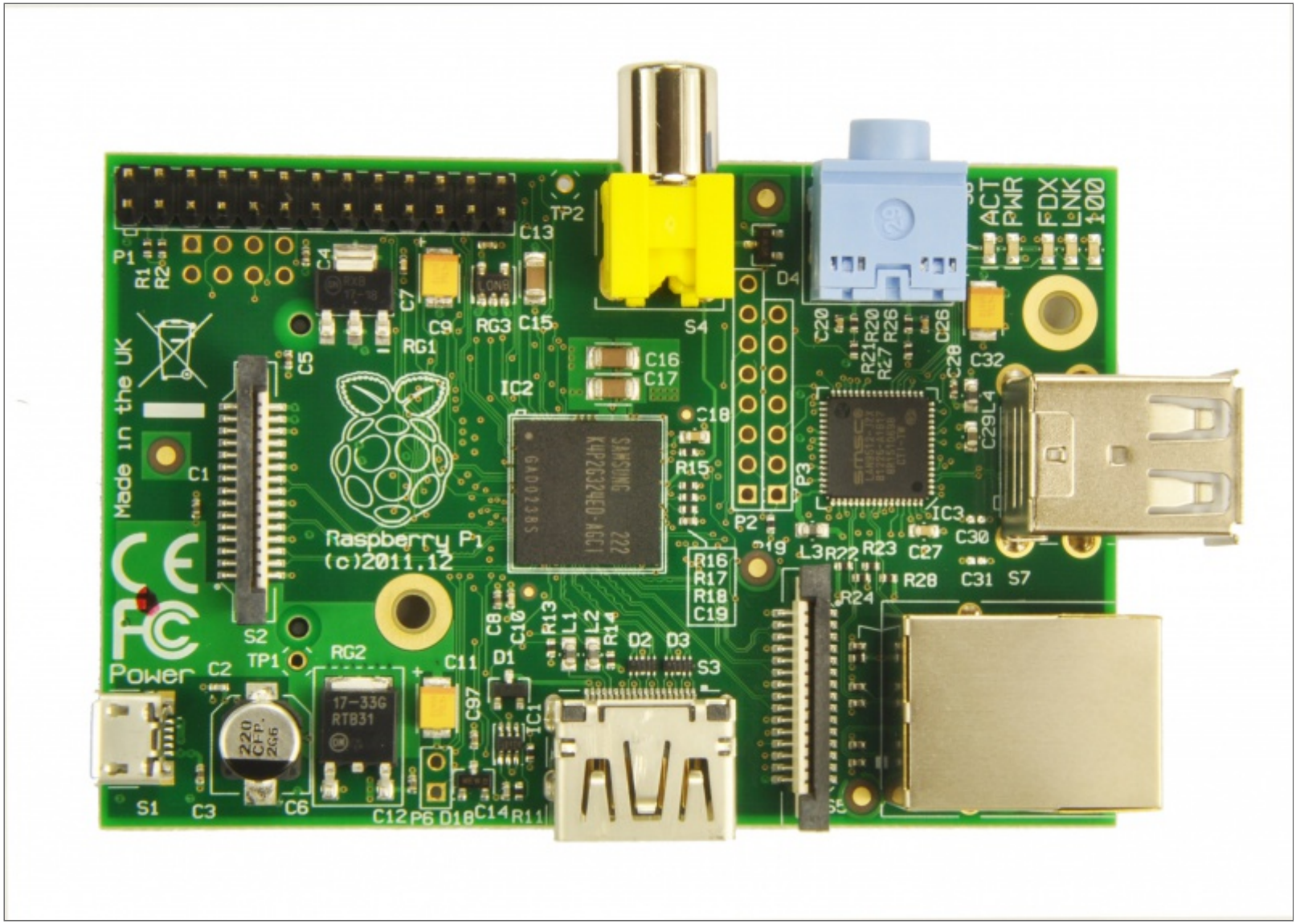

73.1%
23.1%
12.7%
11.1%
9.5%
7.4%
6.0%
6.0%
5.8%
5.6%
Linux
No OS / Bare-metal
FreeRTOS
Other
Windows Embedded
mbed
Contiki
TinyOS
Don't know
RIOT

EMBEDDED LINUX

SO...

WHAT ARE THE EMBEDDED LINUX BUILD SYSTEMS?!

FOR A LOT OF PEOPLE IT ALL STARTS WITH...



BUT THERE ARE OTHERS...

- https://raspberrypi.org
- https://beagleboard.org/black
- http://wiki.litesom.grinn-global.com
- https://wandboard.org
- http://pandaboard.org

name	Raspberry Pi 1	Raspberry Pi 2	BeagleBone Black	PandaBoard	Wandboard Quad	Asus Eee PC 1215n
release date	April 2012	February 2015	April 2013	October 2010	February 2013	August 2010
target price	$35	$35	$45	$174	$129	$499
word size	32-bit	32-bit	32-bit	32-bit	32-bit	32-bit/64-bit
SoC	Broadcom BCM2835	Broadcom BCM2836	Texas Instruments AM3358/9	Texas Instruments OMAP4430	Freescale i.MX6 Quad	Intel Atom
architecture	ARM Cortex-A7	ARM Cortex-A8	ARM Cortex-A8	ARM Cortex-A9	ARM Cortex-A9	x86
CPU frequency	700 MHz	1000 MHz	1000 MHz	1000 MHz	1000 MHz	1800 MHz
RAM size	512 GB DDR3	1 GB	512 MiB DDR3	1 GB	2GB DDR3	2GB DDR3
Power source	5 V (Micro USB/GPIO)	5 V (Micro USB/GPIO)	Mini USB / 5 V jack	5V	5V	19V
USB	2 (via the onboard 5-port USB hub)	4 (via the onboard 5-port USB hub)	USB 2.0	two USB host ports and one USB On-The-Go	USB 3.0	USB 2.0 + USB 3.0
Network	10/100 Mbit/s Ethernet on the USB hub	10/100 Mbit/s Ethernet on the USB hub	Ethernet Fast Ethernet (MII based)	10/100 Ethernet on USB hub	GbE	10/100 Ethernet (MII based)
storage	microSDHC slot	microSDHC slot	4GB eMMC / microSDHC slot	SDHC slot	microSDHC	SATA (default 320 GB HDD)

Table 2.1: Development boards comparison

BUT WHY?

- for prototyping
- for promotion
- for learning
- for fun

HOW TO MAKE IT WORK?

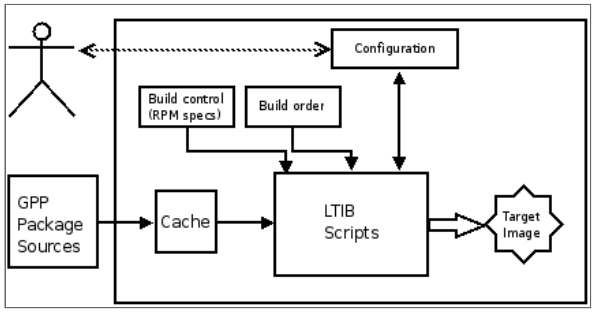
- Download dedicated OS image from web
- or... build own image!

```
wget https://downloads.raspberrypi.org/raspbian_latest

pv sdcard.img | dd of=/dev/sdb bs=4M oflag=dsync
```

TESTING AND COMPARING

- Representative set of embedded devices
- Available open source distribution builders

OpenWrt
Wireless Freedom



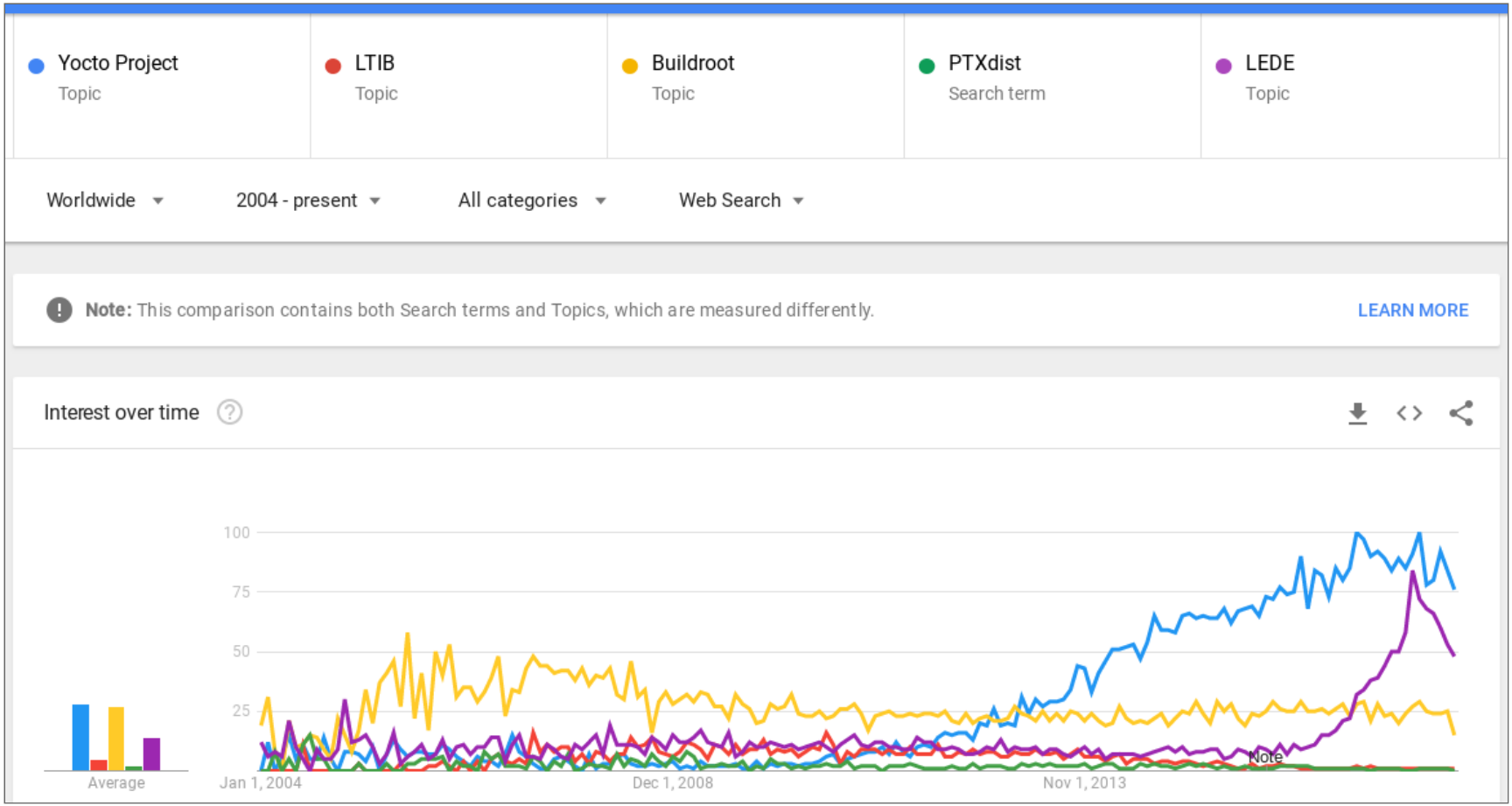
PTX DIST

name	Buildroot	OpenWrt	LTIB	PTXdist	Yocto Project	CLFS
official website	https://buildroot.org	https://openwrt.org	http://ltib.org	https://ptxdist.org	https://yoctoproject.org	http://clfs.org
source code repository	https://git.busybox.net/buildroot/	https://git.openwrt.org	http://cvs.savannah.gnu.org/viewvc/ltib/	https://git.pengutronix.de/cgit/ptxdist	https://git.yoctoproject.org	http://git.clfs.org
license	GPLv2[4]	GPLv2[27]	GPLv2[21]	GPLv2[28]	GPLv2, MIT and others[41]	OPLv1[8]
last stable release date	2017.11	2017.01	2013.02	2018.01	2017.10	2014.10
release cycle	three month[5]	irregular (aprox 1 year)	irregular (aprox 2 years)[22]	one month[29]	six months[42]	irregular[9]

Table 3.1: Embedded Linux build systems overview

BUT THERE ARE OTHERS...

- linux from scratch
- debootstrap
- wind river

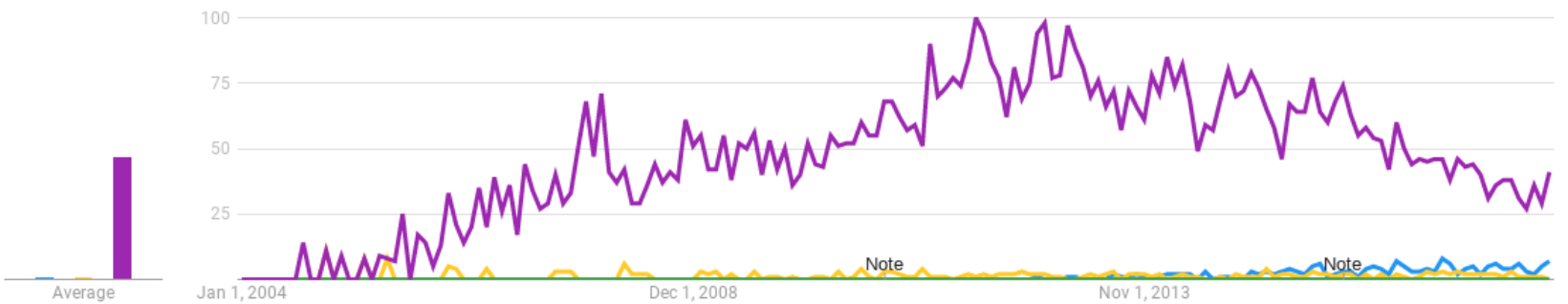

Yocto Project
Topic
LTIB
Topic
Buildroot
Topic
PTXdist
Search term
LEDE
Topic
Worldwide
2004 - present
All categories
Web Search
Note: This comparison contains both Search terms and Topics, which are measured differently.
LEARN MORE
Interest over time
100
75
50
25
Note
Average
Jan 1, 2004
Dec 1, 2008
Nov 1, 2013

Yocto Project	LTIB	Buildroot	PTXdist	OpenWrt
Topic	Topic	Topic	Search term	Operating system

Poland | 2004 - present | All categories | Web Search

Note: This comparison contains both Search terms and Topics, which are measured differently. LEARN MORE

Interest over time



GENERALIZED OS BUILDERS STRUCTURE

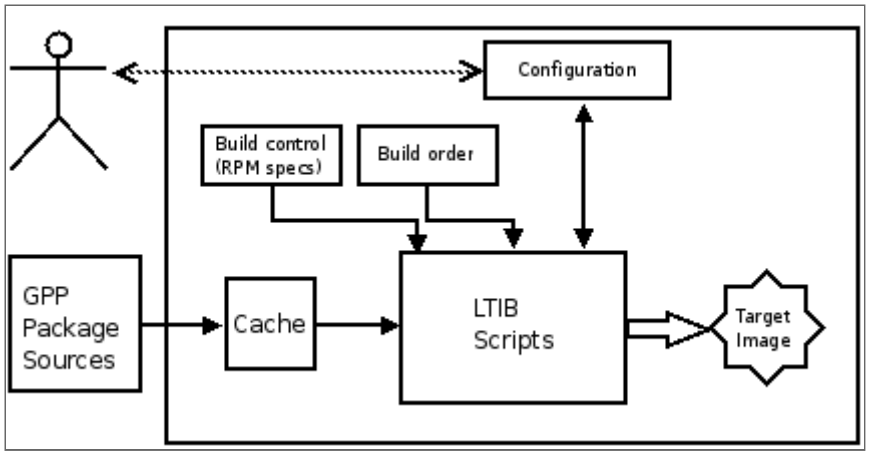
- Source code
- Host OS requirements
- Cross-compilation toolchain
- Target OS configuration
- Produced output

```
apt-get install python-dev python-pip
pip install python-openstackclient

openstack server create --flavor 'c2-30' --image 'Debian 9' --network 'Ext-Net' --key-name

ssh debian@54.37.130.68
```

```
sudo apt update
sudo apt upgrade
sudo apt install make gcc g++ libncurses-dev unzip git patch python rsync bc bzip2
wget https://buildroot.org/downloads/buildroot-2017.02.8.tar.gz
tar -zxf buildroot-2017.02.8.tar.gz
cd buildroot-2017.02.8/
make raspberrypi_defconfig
make
ls output/images/sdcard.img
```


OpenWrt
Wireless Freedom

```
sudo apt install make gcc g++ libncurses-dev unzip git gawk file zlib1g-dev
git clone -b v17.01.4 https://github.com/openwrt/openwrt
cd openwrt/
./scripts/feeds update -a
./scripts/feeds install -a
make defconfig
make menuconfig
# Target System (Broadcom BCM27xx)
# Target Profile (Raspberry Pi B/B+/CM/Zero/ZeroW)
time make -j
cp build_dir/target-arm_arm1176jzf-s+vfp_musl-1.1.16_eabi/linux-brcm2708_bcm2708/tmp/lede-
```


Configuration
Build control
(RPM specs)
Build order
GPP
Package
Sources
Cache
LTIB
Scripts
Target
Image

```
sudo apt install make gcc g++ libncurses-dev unzip rpm bison patch tcl zlib1g-dev

wget https://github.com/downloads/midnightyell/RPi-LTIB/raspberrypi-tools-9c3d7b6-1.i386.r
sudo mkdir -p /opt/ltib/pkgs/
sudo cp raspberrypi-tools-9c3d7b6-1.i386.rpm /opt/ltib/pkgs/

sudo dpkg --add-architecture i386
sudo apt update
sudo apt install zlib1g:i386 libstdc++:i386

wget http://download.savannah.nongnu.org/releases/ltib/ltib-13-2-1-sv.tar.gz
tar -xzf ltib-13-2-1-sv.tar.gz
cd ltib-13-2-1-sv/
time ./ltib

# Platform choice (Raspberry Pi with BCM2835 SoC)

cp output/images/sdcard.img ~
```


PTX
DIST

```
sudo apt install make gcc g++ libncurses-dev unzip git gawk flex bison gettext python-de
wget http://public.pengutronix.de/software/ptxdist/ptxdist-2018.01.0.tar.bz2
tar -xjf ptxdist-2018.01.0.tar.bz2
cd ptxdist-2018.01.0
./configure
make
sudo make install
cd ..
wget http://public.pengutronix.de/software/ptxdist/ptxdist-2016.06.0.tar.bz2
tar -xjf ptxdist-2016.06.0.tar.bz2
cd ptxdist-2016.06.0
./configure
make
sudo make install
cd ..

wget https://public.pengutronix.de/oselas/toolchain/OSELAS.Toolchain-2016.06.1.tar.bz2
tar -xjf OSELAS.Toolchain-2016.06.1.tar.bz2
cd OSELAS.Toolchain-2016.06.1/
ptxdist-2016.06.0 select ptxconfigs/arm-1136jfs-linux-gnueabihf_gcc-5.4.0_glibc-2.23_bin
ptxdist-2016.06.0 migrate
time ptxdist-2016.06.0 go

git clone https://git.pengutronix.de/cgit/DistroKit/
```

```
wget http://commondatastorage.googleapis.com/git-repo-downloads/repo
chmod a+x repo
sudo mv repo /usr/local/bin/

sudo apt install make gcc g++ unzip git
sudo apt install gawk diffstat texinfo build-essential chrpath

export MACHINE=raspberrypi

mkdir yoctoproject
cd yoctoproject
repo init -u https://github.com/cdynak/yocto-manifest -m $MACHINE.xml
repo sync

source poky/oe-init-build-env
%MACHINE=wandboard DISTRO=poky source setup-environment build
% vi conf/bblayers.conf (...?)
time bitbake core-image-minimal
cp tmp/deploy/images/$MACHINE/*img ~
```

TESTS

- With respect to the target OS
- With respect to the host OS

name	Raspberry Pi 1	Raspberry Pi 2	BeagleBone Black	PandaBoard	Wandboard Quad	Asus Eee PC 1215n
real time	17m33.619s	20m15.836s	26m12.604s	11m41.240s	11m39.519s	17m40.604s
user time	73m47.968s	74m30.368s	55m52.408s	49m39.200s	48m33.172s	53m31.832s
sys time	3m54.620s	3m58.084s	3m13.860s	2m31.172s	2m27.688s	2m30.832s
buildroot/	5.4G	5.4G	6.0G	4.8G	4.7G	5.9G
sdcard.img	93M	93M	77M	69M	61M	121M
boot time	4.926830	5.575466	4.028472	4.926830	3.235903	19.957294

Table 4.1: Buildroot build comparison

name	Raspberry Pi 1	Raspberry Pi 2	BeagleBone Black	PandaBoard	Wandboard Quad	Asus Eee PC 1215n
real time	12m2.398s	12m5.585s	13m19.282s	-	31m32.177s	54m22.104s
user time	62m52.308s	62m24.408s	66m25.504s	-	43m14.808s	42m35.520s
sys time	3m19.248s	3m16.664s	3m27.880s	-	2m33.428s	2m34.988s
openwrt/	8.4G	8.1G	8.5G	-	8.1G	9.0G
sdcard.img	285M	277M	254M	-	273M	303M
boot time	8.492709	10.559345	7.952850	-	6.982082	25.640392

Table 4.2: OpenWrt build comparison

name	Raspberry Pi 1	Raspberry Pi 2	BeagleBone Black	PandaBoard	Wandboard Quad	Asus Eee PC 1215n
real time	27m17.629s	20m8.447s	20m8.447s	-	-	25m43.452
user time	87m36.840s	74m21.468s	74m21.468s	-	-	81m15.935
sys time	9m22.424s	4m31.152s	4m31.152s	-	-	7m1.295
Toolchain	15G	15G	15G	-	-	15G
real time	28m7.177s	25m44.733s	25m44.733s	-	-	27m42.724
user time	60m39.504s	58m52.088s	58m52.088s	-	-	59m54.615
sys time	4m13.768s	3m59.556s	3m59.556s	-	-	4m9.943
DistroKit/	5.5G	7.3G	7.3G	-	-	6.1G
sdcard.img	84M	81M	80M	-	-	105M
boot time	5.230498	5.259048	4.98392	-	-	20.59385

Table 4.3: PTXdist build comparison

name	Raspberry Pi 1	Raspberry Pi 2	BeagleBone Black	PandaBoard	Wandboard Quad	Asus Eee PC 1215n
real time	34m17.598s	34m24.622s	31m25.839s	19m0.417s	35m19.749s	38m29.501s
user time	201m14.080s	201m43.528s	178m14.064s	97m14.776s	205m9.384s	239m19.756s
sys time	14m52.688s	14m40.556s	13m8.788s	9m18.528s	14m0.868s	13m29.216s
yoctoproject/	24G	24G	26G	21G	24G	25G
sdcard.img	53M	53M	48M	49M	28M	26M
boot time	4.983091s	3.951380s	4.093724s	5.159032	3.539520	25.293039

Table 4.4: Yocto Project build comparison

name	C2-30	B2-30	C2-15	B2-15	C2-7	B2-7
vCPUs	8 x 3.1GHz	8 x 2.3GHz	4 x 3.1GHz	4 x 2.3GHz	2 x 3.1GHz	2 x 2.3GHz
RAM	30 GB	30 GB	15 GB	15 GB	7 GB	7 GB
price	1.542	1.049	0.765	0.519	0.395	0.272
real time	24m13.863s	24m52.772s	34m21.638s	37m10.730s	50m4.480s	57m17.304s
user time	73m29.016s	75m56.208s	71m51.860s	81m13.392s	72m39.468s	82m59.928s
sys time	8m11.556s	9m15.448s	7m35.004s	9m39.060s	7m41.560s	9m27.172s
total cost	0.62 PLN	0.42 PLN	0.43 PLN	0.32 PLN	0.33 PLN	0.26 PLN

Table 4.5: Comparison of VM size and build time (CPU instances)

name	R2-30	R2-15	S1-8	S1-4
vCPUs	2 x 2.4GHz	2 x 2.4GHz	2 x 2.4GHz	1 x 2.4GHz
RAM	30 GB	15 GB	8 GB	4 GB
price PLN/h	0.457	0.395	0.148	0.081
real time	52m33.380s	53m30.474s	65m42.849s	127m16.353s
user time	77m29.732s	80m41.752s	91m34.000s	101m2.204s
sys time	7m55.024s	7m36.960s	12m8.204s	15m2.652s
total cost	0.4 PLN	0.35 PLN	0.16 PLN	0.17 PLN

Table 4.6: Comparison of VM size and build time (RAM instances)

EXAMPLE USE CASES

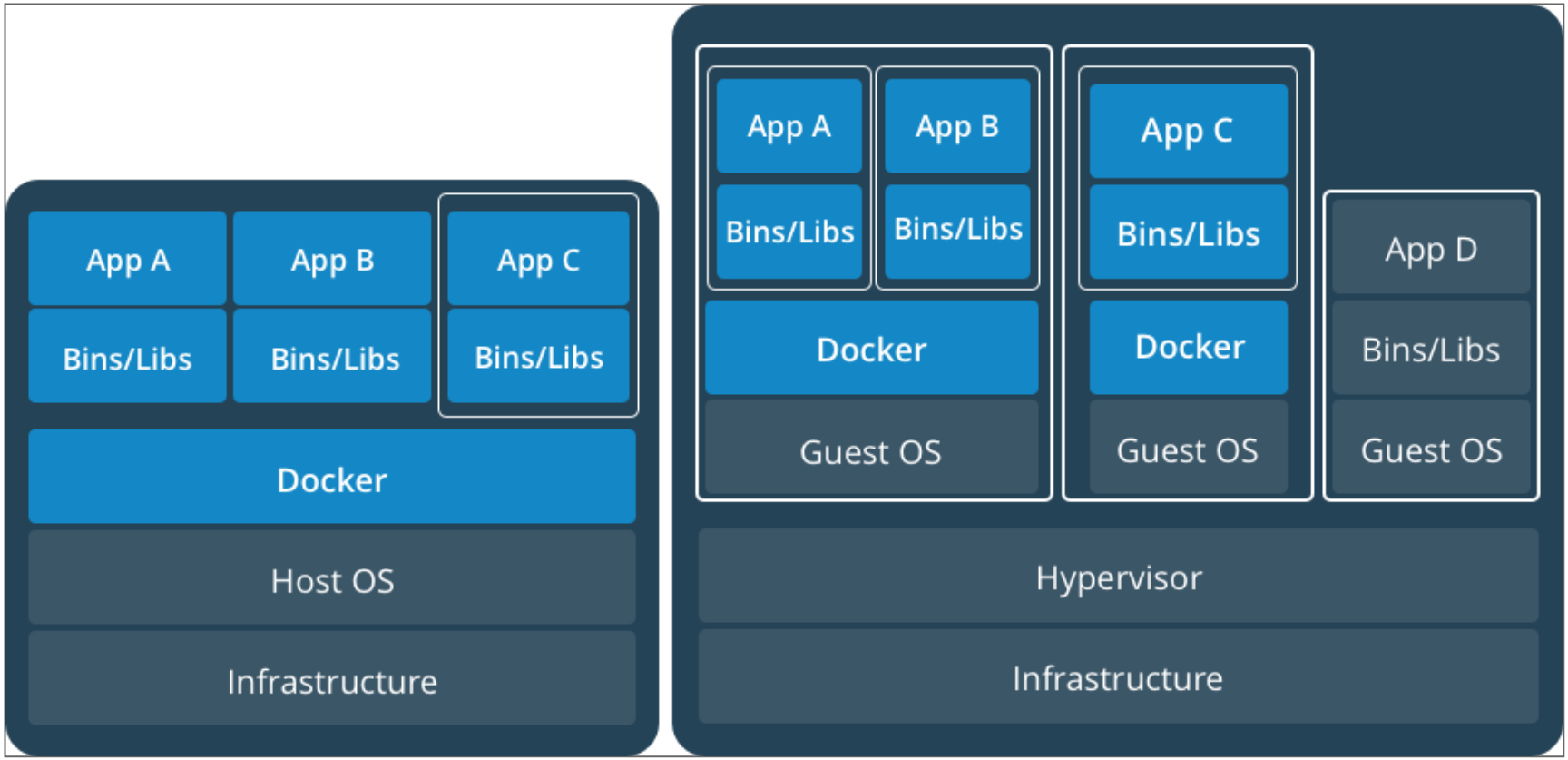
- The Node.js IoT application
- Building Containers

NODE.JS SCADA

- **github.com/newterm/szarp** (C++/wxWidgets, MIT)
- **github.com/RapidScada/scada** (C#, Apache)
- **github.com/SCADA-LTS/Scada-LTS** (Java, GPLv2)
- **github.com/trombastic/PyScada** (Python, GPLv3)
- **oscada.org/websvn/listing.php?repname=OpenSCADA** (C++, GPLv2)

LINUX CONTAINERS

CURRENTLY THERE IS ONLY ONE KING


App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Docker
Host OS
Infrastructure
App A
App B
Bins/Libs
Bins/Libs
Docker
Guest OS
App C
Bins/Libs
Docker
Guest OS
App D
Bins/Libs
Guest OS
Hypervisor
Infrastructure

HOW TO MAKE IT WORK?

- Download dedicated OS image from web
- or... build own image!

```
sudo apt install apt-transport-https ca-certificates curl gnupg2 software-properties-commo
curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg | sudo a
sudo apt-key fingerprint 0EBFCD88
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-re
sudo apt update
sudo apt install docker-ce
```

```
docker pull alpine
docker images
docker run alpine ls -l
docker run -it alpine /bin/sh
```

BUILDING LINUX CONTAINERS

```
cd output/images
mkdir extra extra/etc extra/sbin extra/lib extra/lib64
touch extra/etc/resolv.conf
touch extra/sbin/init
cp /lib/x86_64-linux-gnu/libpthread.so.0 /lib/x86_64-linux-gnu/libc.so.6 extra/lib
cp /lib64/ld-linux-x86-64.so.2 extra/lib64
cp rootfs.tar fixup.tar
tar rvf fixup.tar -C extra .
docker import - basic-system < fixup.tar
docker run -t -i basic-system /bin/sh
```

CONCLUSIONS

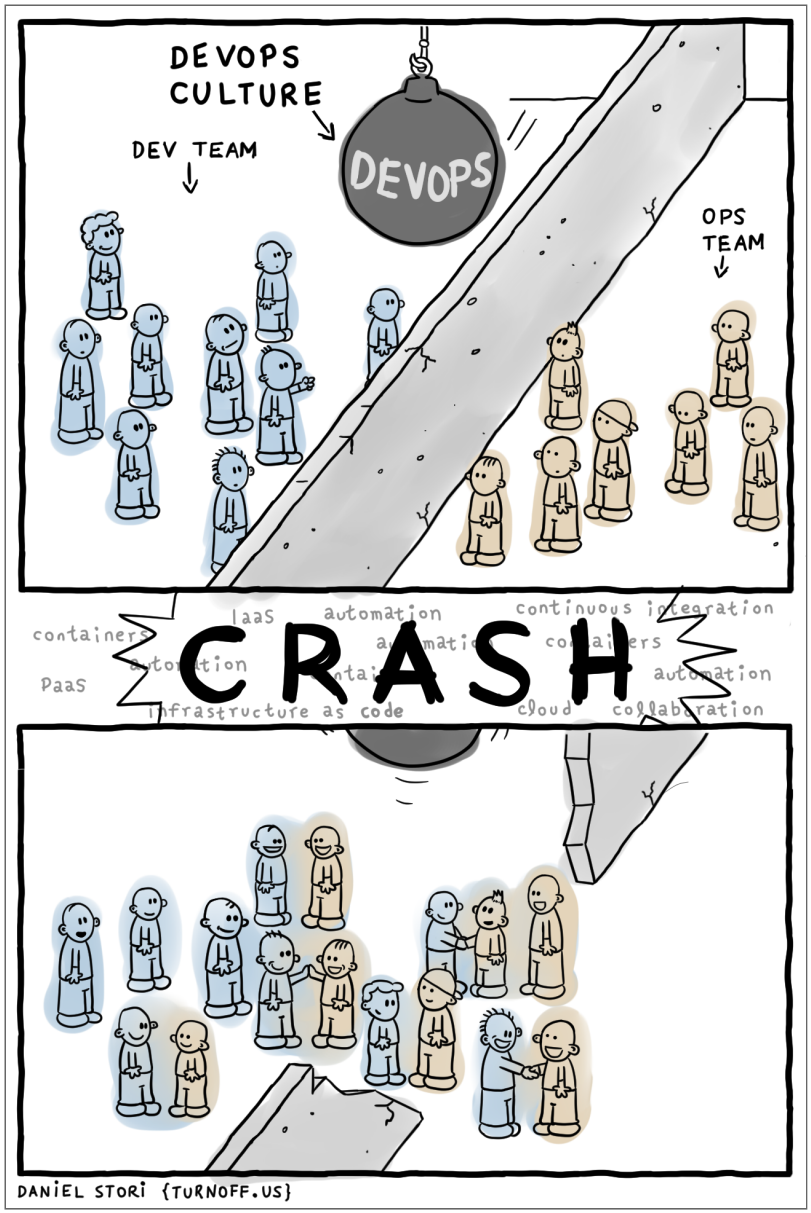
BUILD SYSTEMS ARE BETTER THAN DISTRIBUTIONS

- For professional usage
- Latest versions of kernel and packages
- Easy update and portability
- Examples: PandaBoard, Raspbian MPK


EMTEST
Raspbian GNU/Linux 8 cmreader tty1
cmreader login:
MENNICA
POLSKA

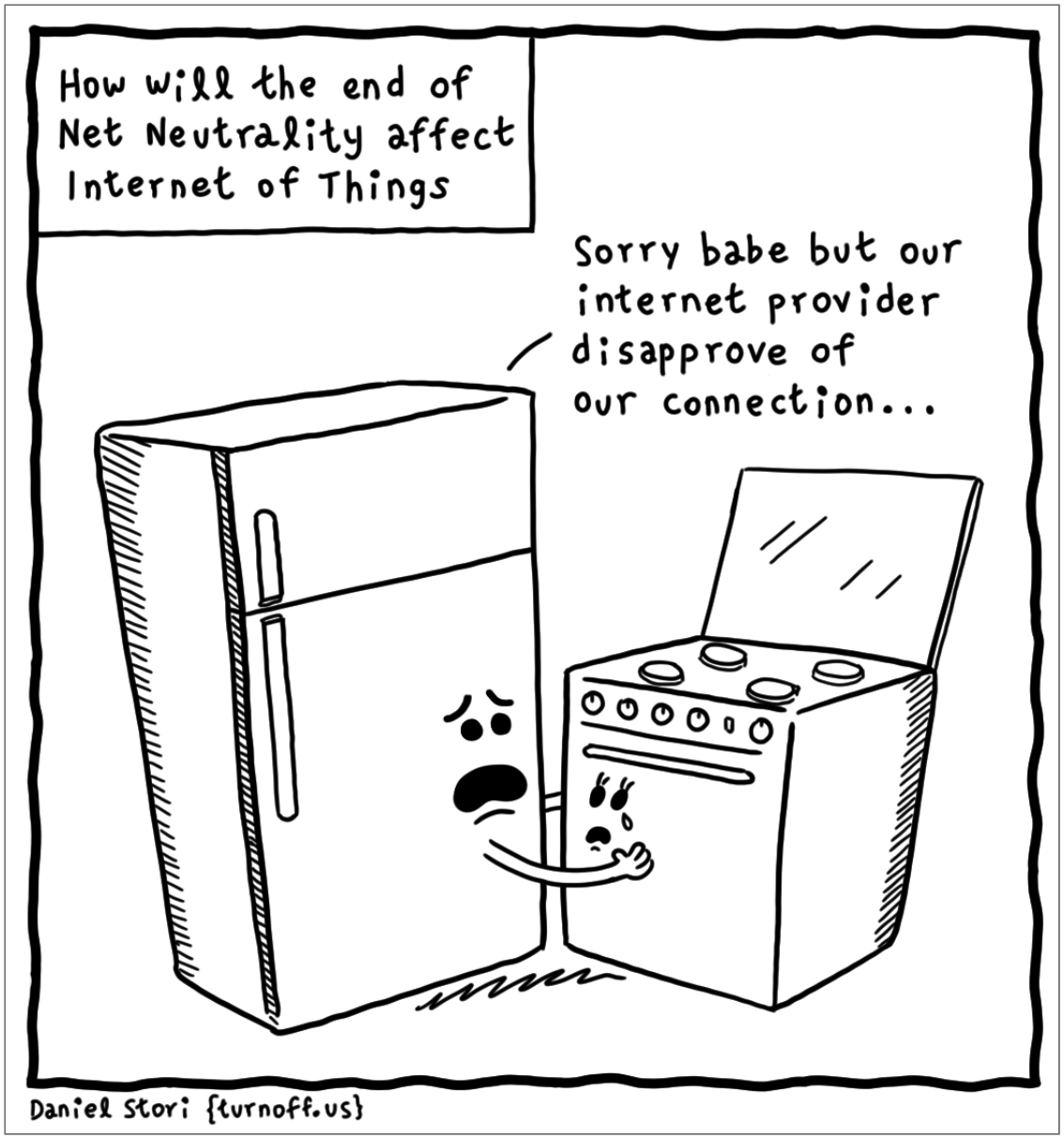
COMPARABILITY

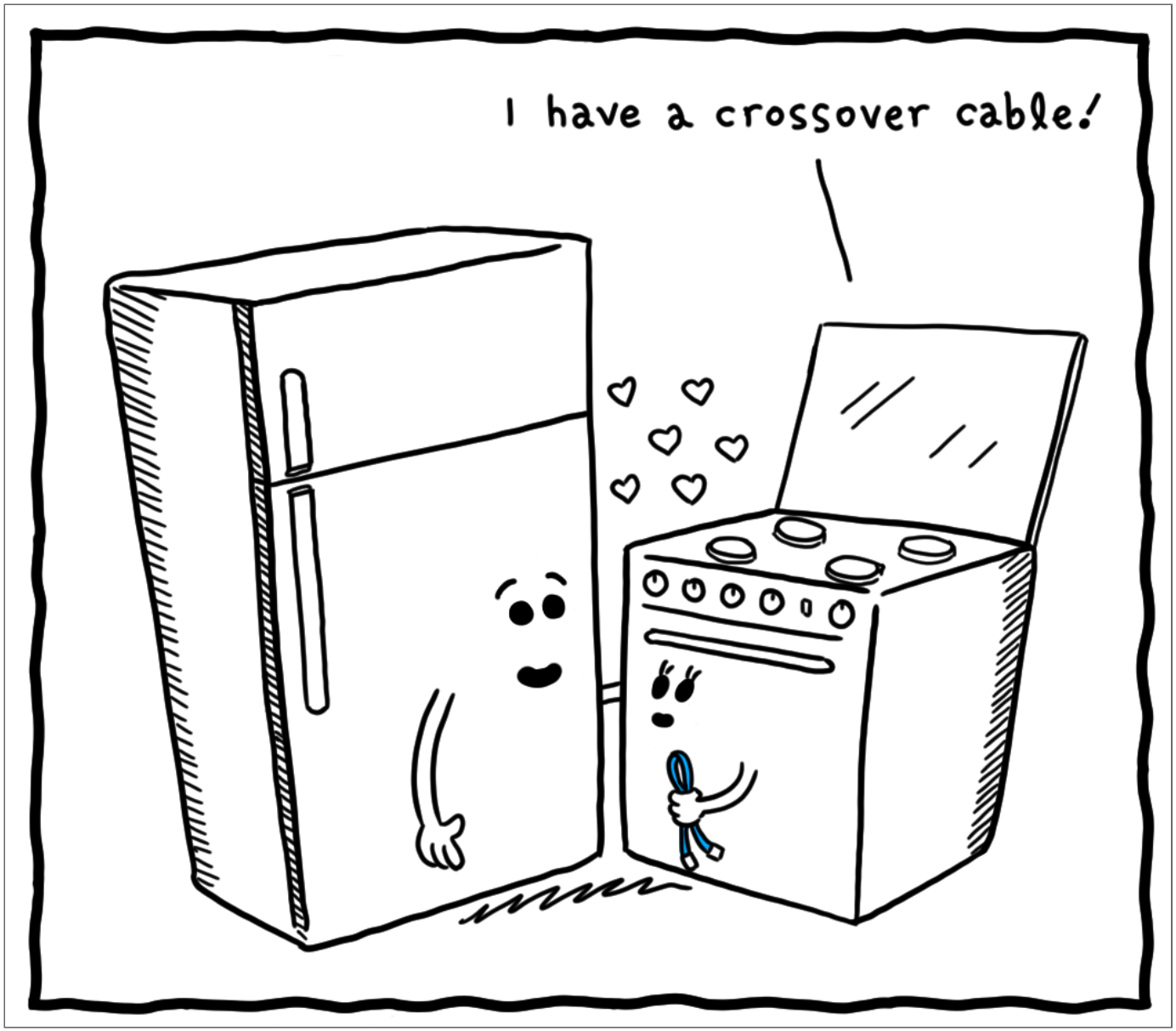
- Experience from one build system is useful on the others
- Software is easily portable between build systems
- Companies should try to experiment


DEVOPS CULTURE
DEV TEAM
DEVOPS
OPS TEAM
containers
IaaS
automation
continuous integration
PaaS
CRASH
automation
infrastructure as code
cloud
collaboration
DANIEL STORI {TURNOFF.US}

EDGE COMPUTING

- Perfect tools for edge computing
- Portable between cloud and embedded
- It is affordable to run Node.js on embedded device
- Escaping from vendor/cloud lock-in...


How will the end of Net Neutrality affect Internet of Things
Sorry babe but our internet provider disapprove of our connection...
Daniel Stori {turnoff.us}


I have a crossover cable!

OPEN QUESTIONS

- Why only Linux?
- What about MINIX, BSD, Darvin and HURD?
- Embedded Unix build systems?
- (yes, I know that Linux Standard Base and other "great standarizations" failed...)

THANK YOU FOR ATTENTION

BIBLIOGRAPHY IS INSIDE SOURCE CODE

cdynak.github.io/linux2018

github.com/cdynak/linux2018

github.com/cdynak/embedded-linux-build-systems

DISCUSSION / QUESTIONS?