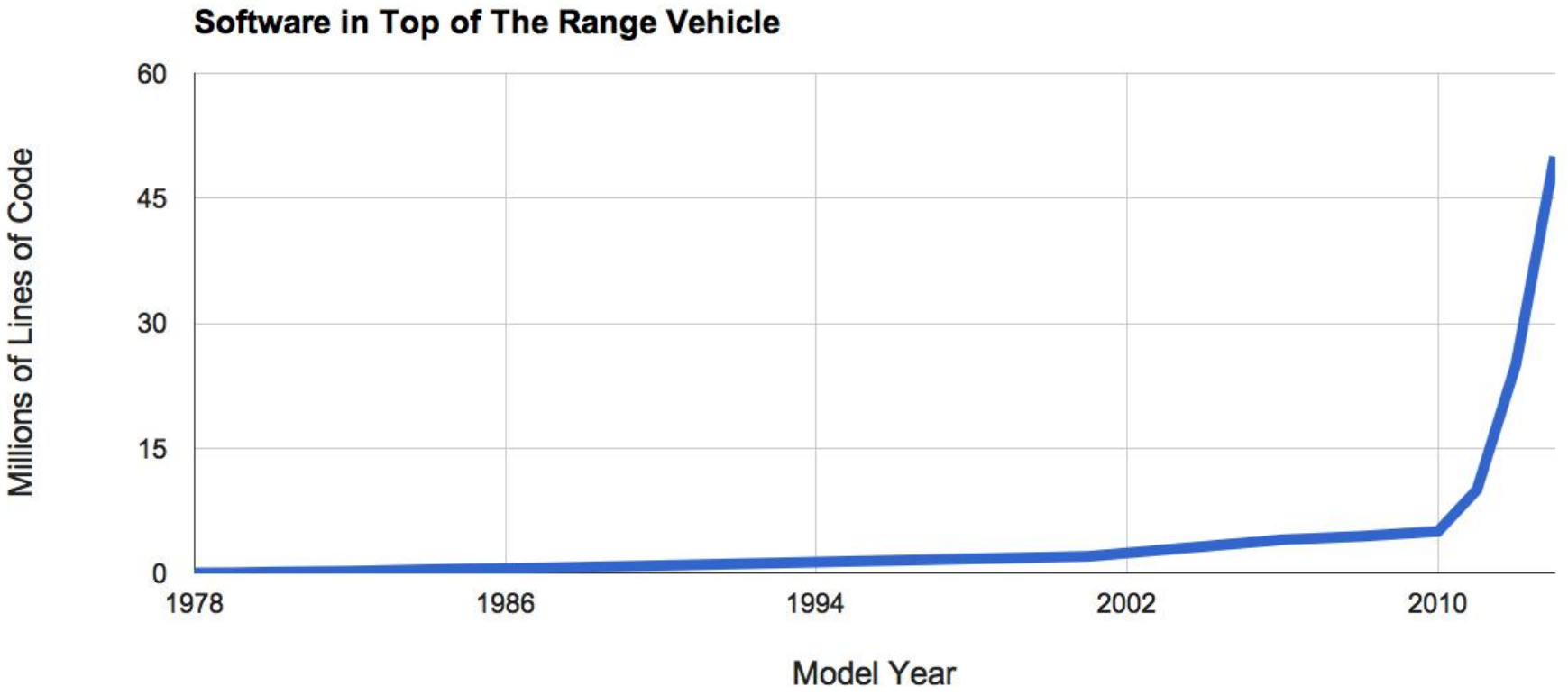
Mainline, LTS, LTSI and GENIVI

Thanks to Greg Kroah-Hartman, John Ellis

Mainline, LTS, LTSI and GENIVI

- Context/scale of GENIVI-based systems
- Explain Linux kernel mainline, LTS, LTSI
- Show how they relate to GENIVI systems
- Highlight the gaps between LTS, LTSI and production IVI

Growth of codebase in vehicle



Source: internet + anecdotes

CodeThink Providing Genius.

GENIVI system software scale

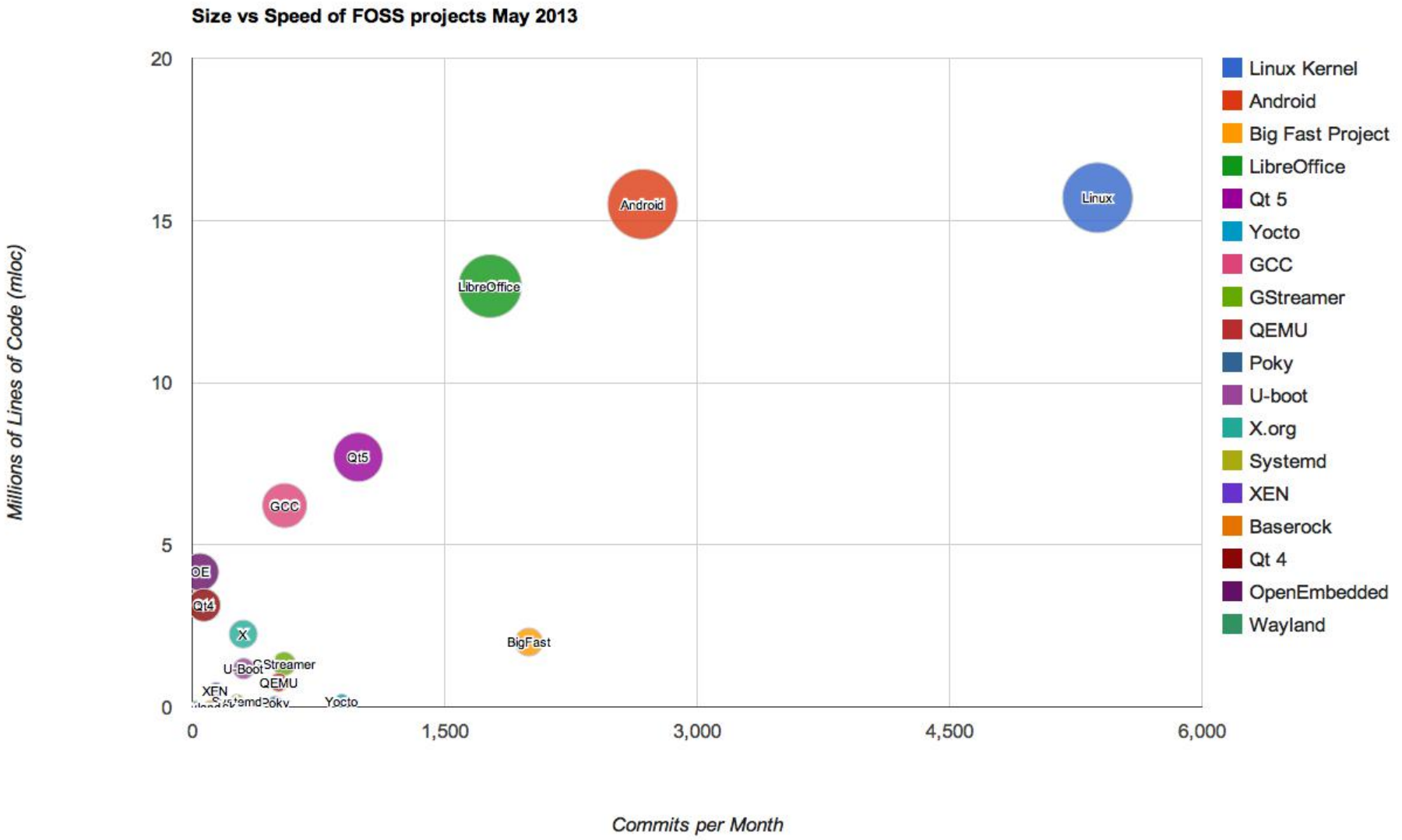
- GENIVI Baseline system ~ 35M LOC
- Plus mediaplayer, web browser, satnav, HMI etc
- Easily 50M - 75M LOC in a full IVI system
- Linux kernel is the biggest component ~ 10MLOC *

* Linux Foundation says 17MLOC – my figures are from D.A. Wheeler's 'SLOCCount'

Whole Life Support for IVI software

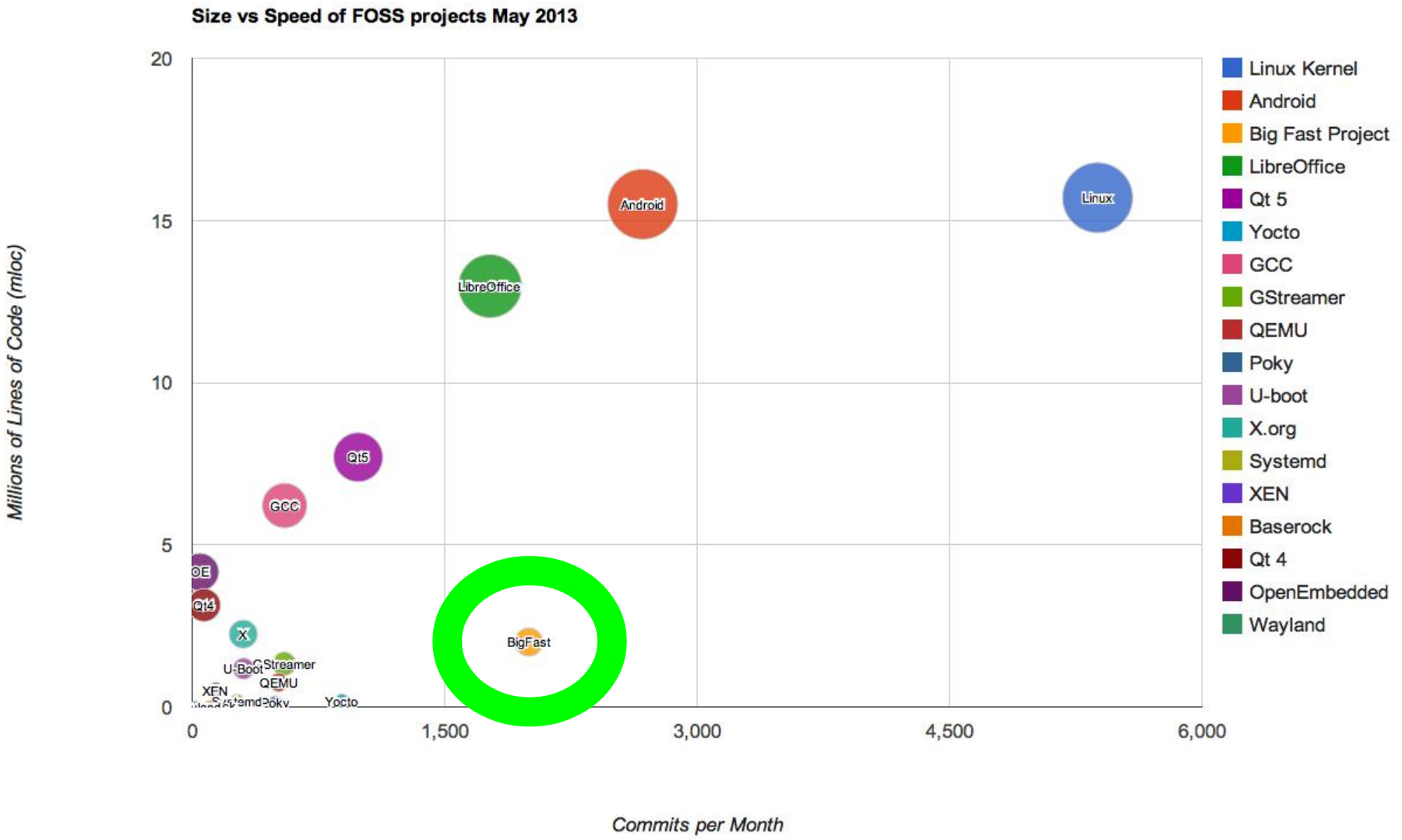
- IVI project development takes 2 to 4 years
- Vehicle to be maintained for 8 to 10 years
- Model range based on same 'platform' for 2 to 5 years
- So total project lifetime is (2 to 4) + (8 to 10)+ (2 to 5)
- Project software maintenance lifetime is 12 to 19 years
- SoC vendor won't be supporting BSP for most of this time

Linux is bigger, harder, faster...



CodeThink Providing Genius.

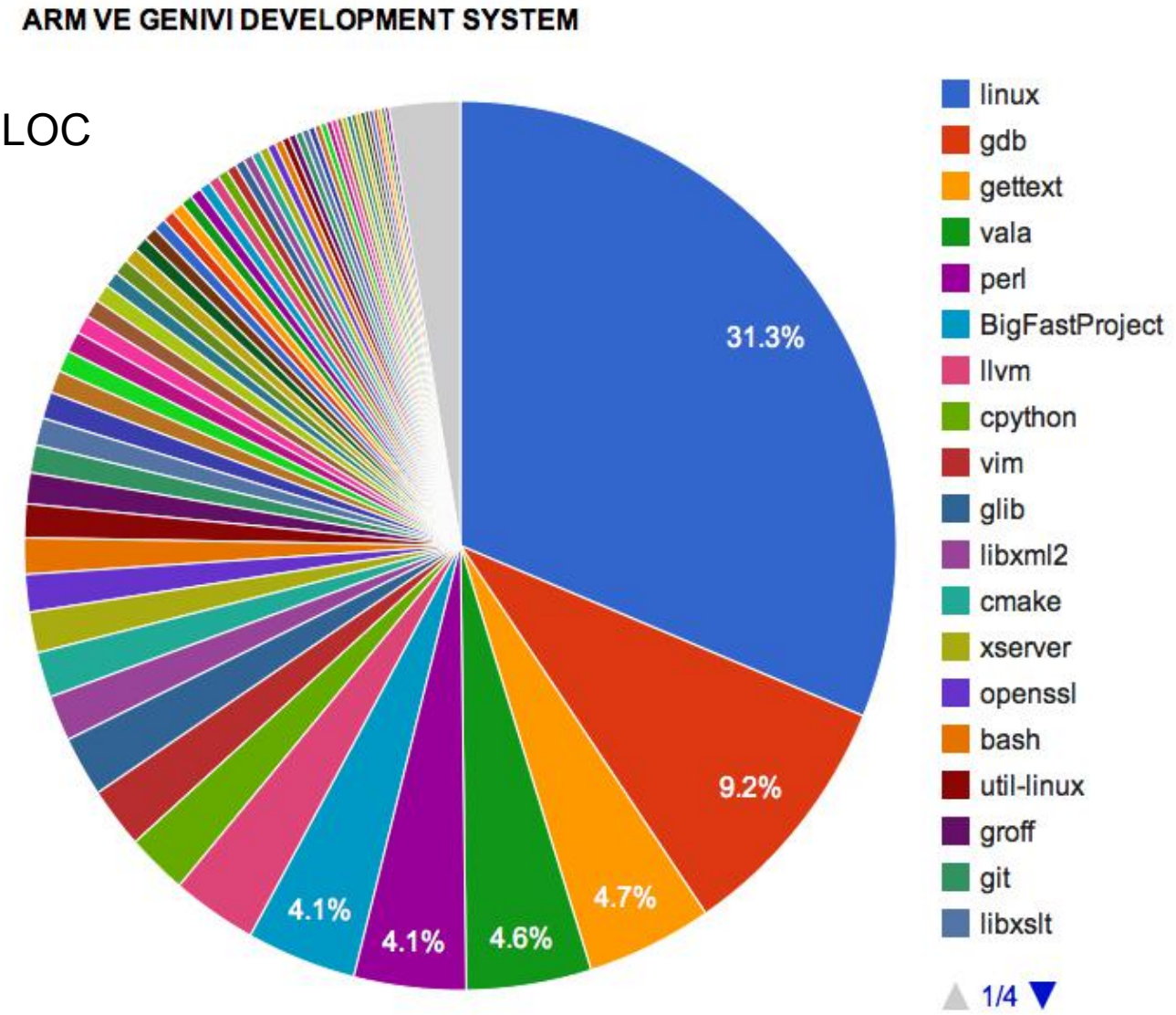
...than your project



CodeThink Providing Genius.

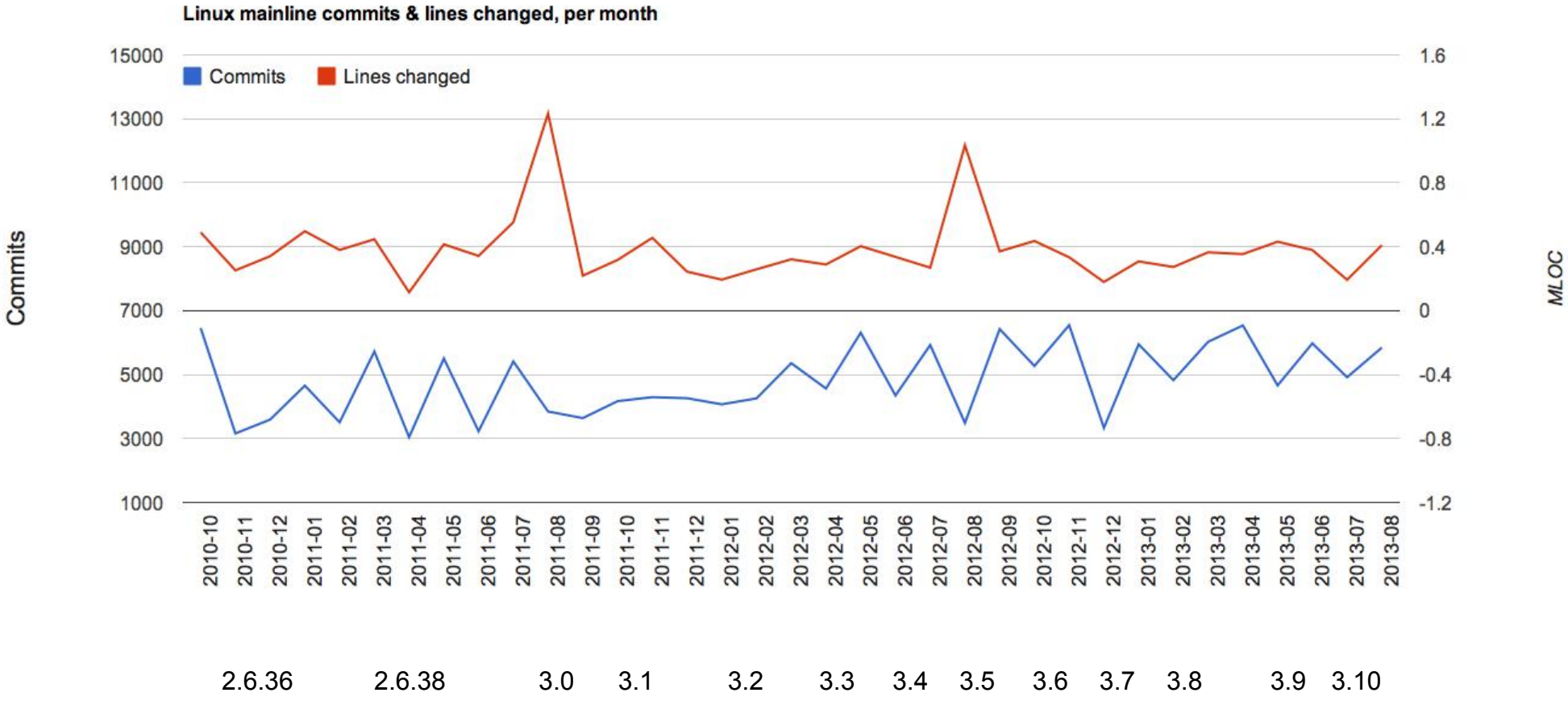
Example Baserock GENIVI development system

47 MILLION LOC



CodeThink Providing Genius.

Linux kernel mainline



2.6.36 2.6.38 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10

CodeThink Providing Genius.

LTS and LTSI

- Are led/sponsored by Linux Foundation and community
- LTS = Long Term Stable kernel trees
  - mainly backported regression and security fixes
- LTSI = Long Term Support Initiative
  - based on LTS
  - mainly SoC vendor BSP stuff (aiming for) upstream

https://www.kernel.org/category/releases.html

Mainline

Mainline tree is maintained by Linus Torvalds. It's the tree where all new features are introduced and where all the exciting new development happens. New mainline kernels are released every 2-3 months.

Stable

After each mainline kernel is released, it is considered "stable." Any bug fixes for a stable kernel are backported from the mainline tree and applied by a designated stable kernel maintainer. There are usually only a few bugfix kernel releases until next mainline kernel becomes available -- unless it is designated a "longterm maintenance kernel." Stable kernel updates are released on as-needed basis, usually 2-3 a month.

Longterm

There are usually several "longterm maintenance" kernel releases provided for the purposes of backporting bugfixes for older kernel trees. Only important bugfixes are applied to such kernels and they don't usually see very frequent releases, especially for older trees.

Longterm release kernels

Version	Maintainer	Released	Projected EOL
3.10	Greg Kroah-Hartman	2013-06-30	Sep, 2015
3.4	Greg Kroah-Hartman	2012-05-20	Oct, 2014
3.2	Ben Hutchings	2012-01-04	2016
3.0	Greg Kroah-Hartman	2011-07-22	Oct, 2013
2.6.34	Paul Gortmaker	2010-05-16	Mid-2013
2.6.32	Willy Tarreau	2009-12-03	Mid-2014

https://www.kernel.org/category/releases.html

Mainline

Mainline tree is maintained by Linus Torvalds. It's the tree where all new features are introduced and where all the exciting new development happens. New mainline kernels are released every 2-3 months.

Stable

After each mainline kernel is released, it is considered "stable." Any bug fixes for a stable kernel are backported from the mainline tree and applied by a designated stable kernel maintainer. There are usually only a few bugfix kernel releases until next mainline kernel becomes available -- unless it is designated a "longterm maintenance kernel." Stable kernel updates are released on as-needed basis, usually 2-3 a month.

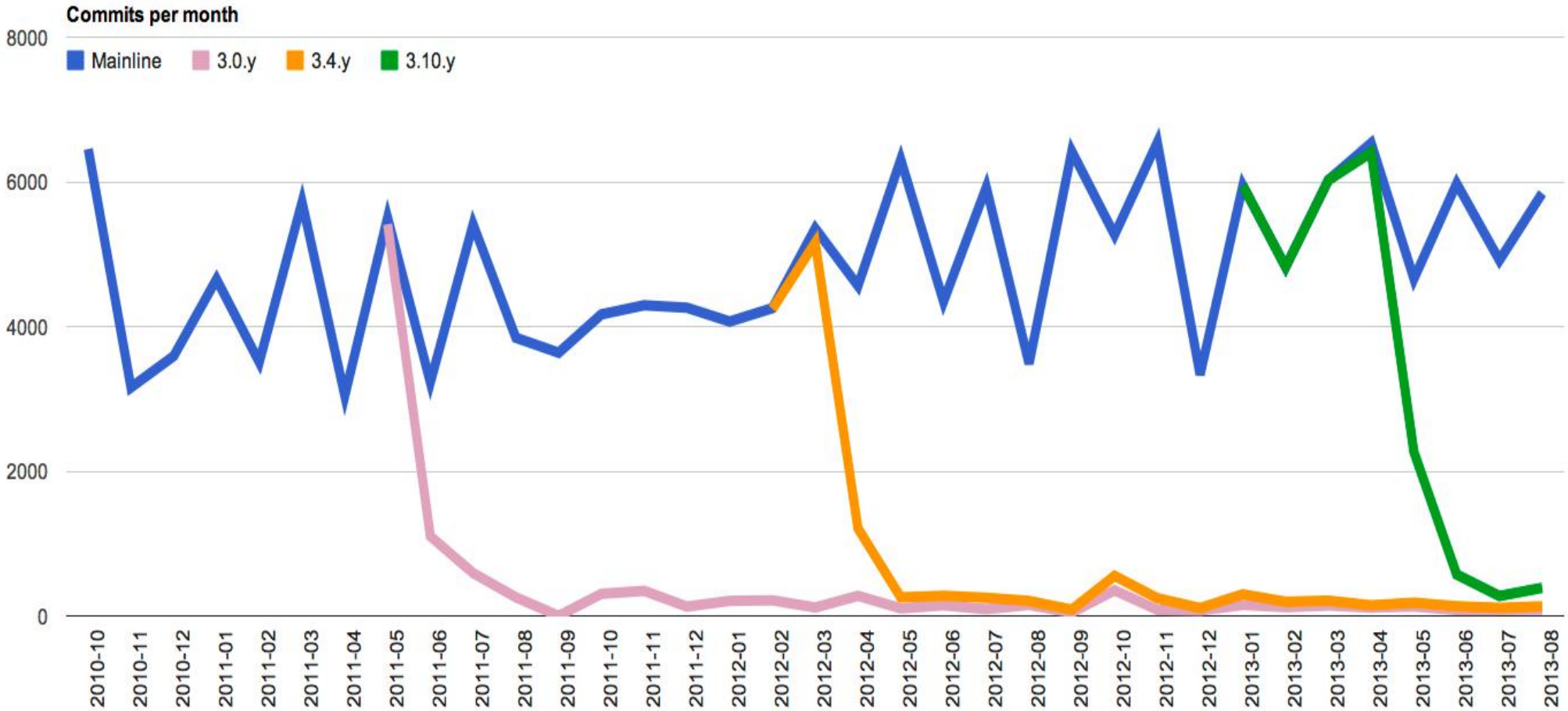
Longterm

There are usually several "longterm maintenance" kernel releases provided for the purposes of backporting bugfixes for older kernel trees. Only important bugfixes are applied to such kernels and they don't usually see very frequent releases, especially for older trees.

Longterm release kernels

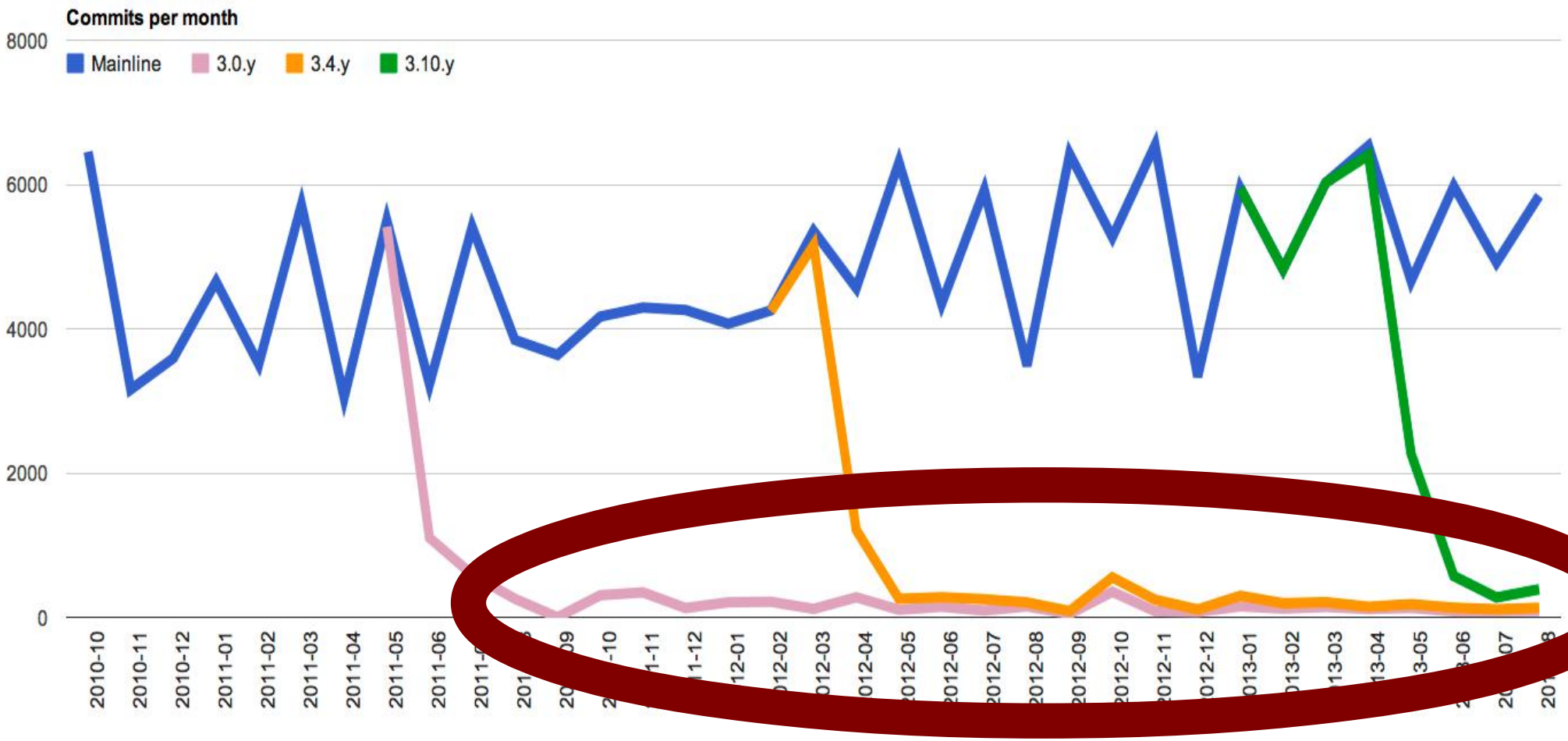
Version	Maintainer	Released	Projected EOL
3.10	Greg Kroah-Hartman	2013-06-30	Sep, 2015
3.4	Greg Kroah-Hartman	2012-05-20	Oct, 2014
3.2	Ben Hutchings	2012-01-04	2016
3.0	Greg Kroah-Hartman	2011-07-22	Oct, 2013
2.6.34	Paul Gortmaker	2010-05-16	Mid-2013
2.6.32	Willy Tarreau	2009-12-03	Mid-2014

Changes in Mainline vs Long Term Stable trees



CodeThink Providing Genius.

Changes in Mainline vs Long Term Stable trees



CodeThink Providing Genius.

What 'stable' really means

- **100 to 200 kernel patches per month**
- new release every 2 to 3 weeks
- regression fixes and security patches
- there is no reason to expect zero issues after two years
- there IS reason to expect this work gets harder over time
- GKH says the numbers are higher, and INCREASING

“All users of the 3.x kernel series must upgrade.”

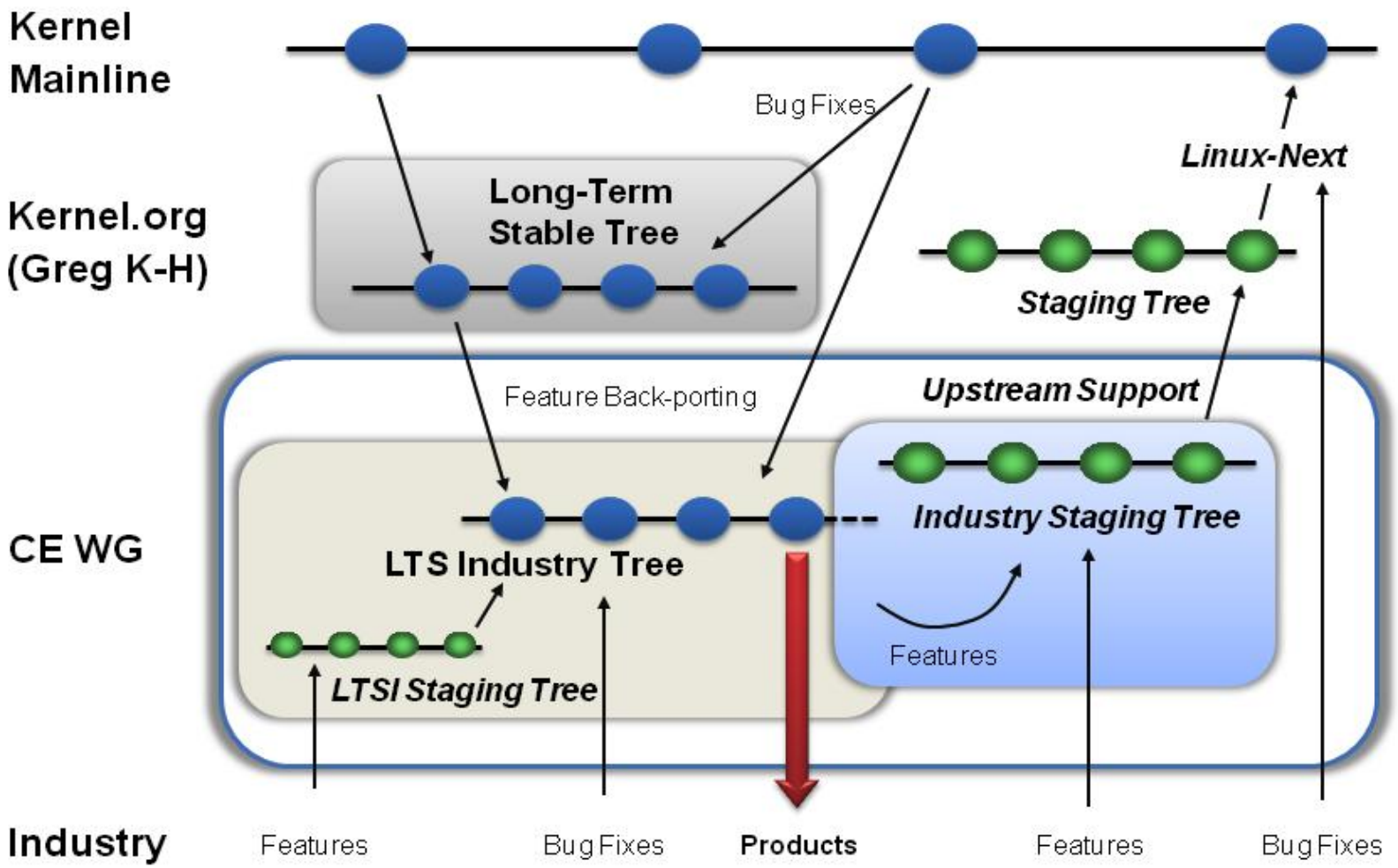
- Because they are security fixes
- And fixes for serious bugs eg. crashes, lockups
- But these update releases happen **every few weeks**
- Which IVI projects **actually take** these upgrades?
- GENIVI baselines are not required to, for example.

LTSI => Long Term Support (Stable?) Initiative (Industry?)

http://ltsi.linuxfoundation.org

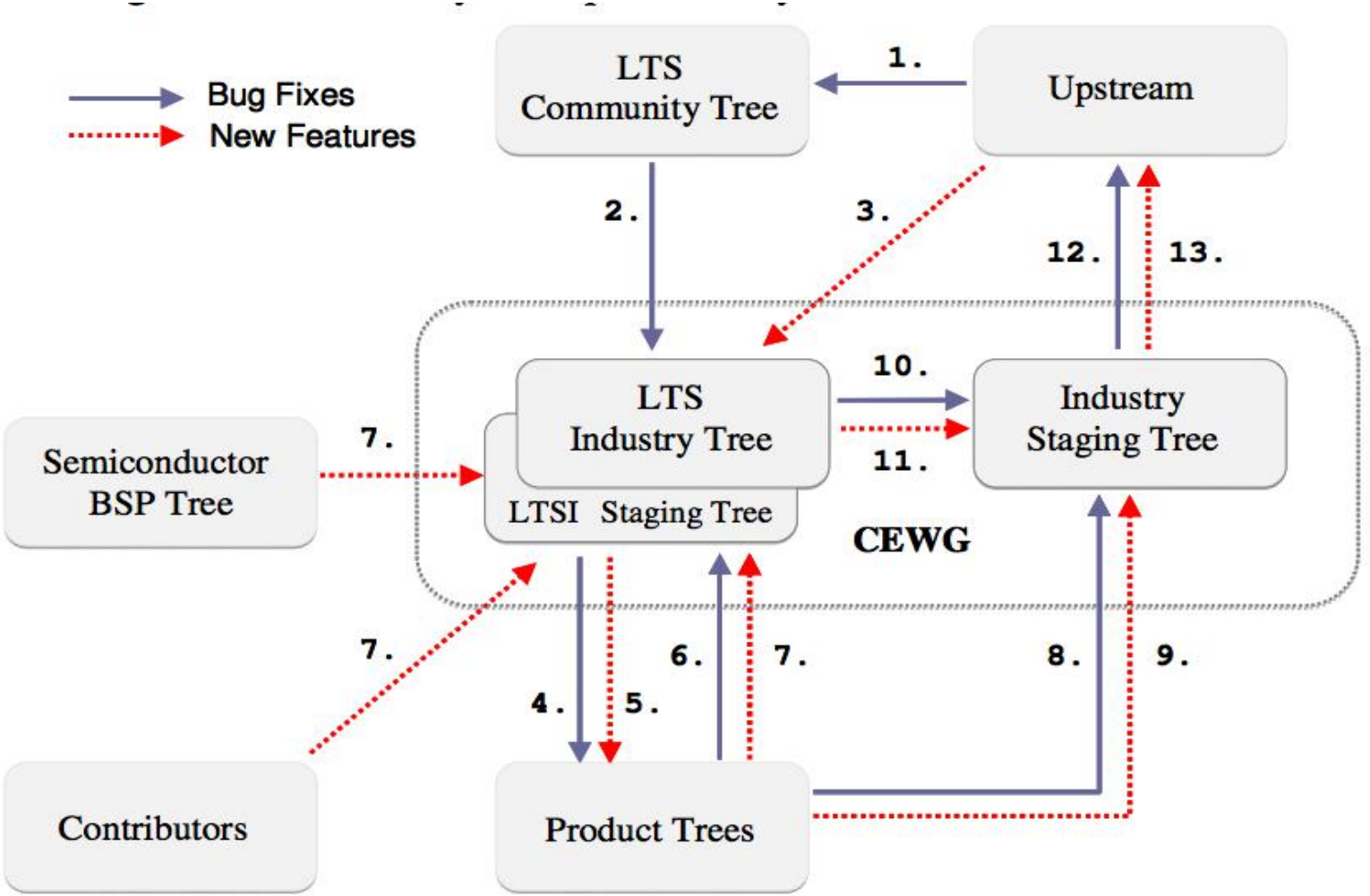
- Long Term Stable Industry patch sets (not trees)
- Based on LTS – adds extra patches, same EOL
- Device-specific patches eg Renesas
- Industry-specific patches eg AF-BUS
- These may or not get into mainline (GKH disagrees)

LTSI



http://ltsi.linuxfoundation.org/what-is-ltsi

LTSI



http://ltsi.linuxfoundation.org/sites/ltsi/files/LTSI%20Overview.pdf

LTSI in the real world...

As of 2013-10-07:

3.0.99 is reaching EOL , 3.4.65 and 3.10.4 are current

SoC Vendor 1:
- mainline work has 'almost zero peripherals'
- current BSP is 1000+ patches on 3.4.25-ltsi

SoC Vendor 2:
- Current BSP published at several places based on 3.0.35

Yocto GENIVI Baseline hands-on this Friday
- Renesas devboard is at 3.4.38 (six months old)

Why two years?

"I refuse to support anything after 2 years unless someone starts to give me tons of money to do so. The patches that apply for stuff after 2 years drops off dramatically, and the work involved in keeping stuff working and testing for problems increases greatly."

Greg K-H

What about Super Long Term Support?

"... some companies, even crazier than automotive vendors, that need 10-20 year support (traffic signals, railways, etc.) They were told they were on their own. If they want to do that, it's nothing that the community can support, especially as they are the ones getting paid to do so."

Greg K-H

Whole Life Support for IVI

- Production project starting now, EOL 2027 – 2038
- Latest possible LTS Kernel EOL today => Sept 2015
- So project needs to PLAN its own Whole Life Support
- Or can GENIVI/Linux Foundation/others offer a solution?

What about the rest of the software in the stack?

- ~ 200 regression/security patches/month for 10MLOC
- ~ (wild guess) 1000 patches/month for 50MLOC system?
- 150,000 to 300,000 patches over lifetime of project?

We could do more work to assess the reality of this

Expect **more** security issues per LOC in closed code

IVI project starting now => EOL 2025 - 2032

- All of the current LTS and LTSI kernels will be long dead
- Linux 3.50 to Linux 3.100+ will be current
- Or Linux 5, Linux 6, Linux 7
- Systemd 1000+, or systemd EOL?

=> interlude...

What does GENIVI (or AGL, or anyone) recommend?

- How should we deal with this?
- Should a production project snapshot and backport?
- Apollo>Borg>Cassini>Discovery>Excalibur>Foton>Gemini>Horizon>Intrepid etc?
- MeeGo => Tizen?

Possible approaches

- Extended Automotive LTS: 5, 10, 15, 20 years?
- Plus equivalent mechanism for the rest of the software?
- Establish principle of aftermarket support for IVI?
- Fixed term licence (eg 2 years), forced service refresh?
- Track upstream releases (incremental updates)?

IMPOSSIBLE approaches

- Base everything on arbitrary vendor SoC/GPU BSP
- Stick at a given GENIVI Compliance for whole of life
- Track every change from LTS and then stop

- Track every change from mainline on all components

Discussion

- How are OEMs planning to deal with this?
- How do platform vendors recommend dealing with this?