



Android and Long-Term Stable Kernel

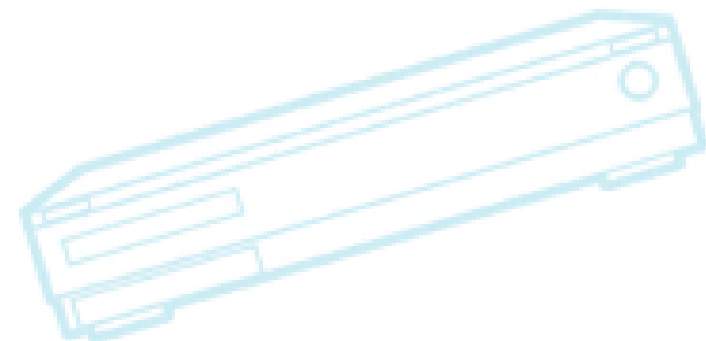
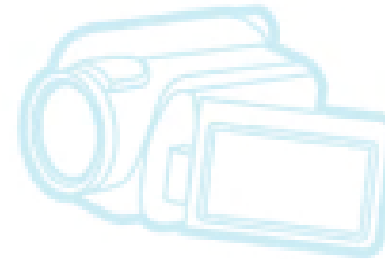
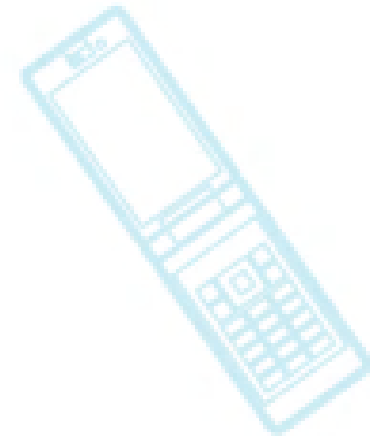
15 Feb, 2012

Tsugikazu Shibata

CE WG, The Linux Foundation

CONTENTS

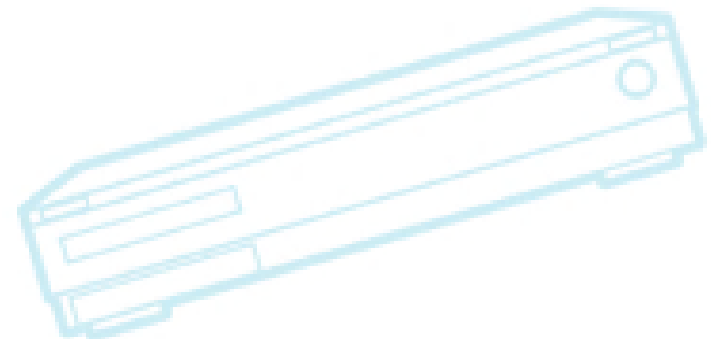
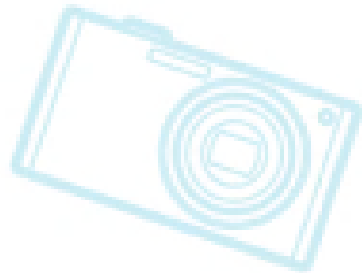
- Background
- Problem Identified
- LTSI Activities
 - Long-Term Stable kernel
 - LTSI Kernel tree
 - Industry support by LTSI
- Conclusion



Background



- Building a product using Android
 - Complicated merging process
 - Rapid release cycle becomes too hard
 - Product lifetime and release cycle not muched
 - Upstream patch submission is inactive



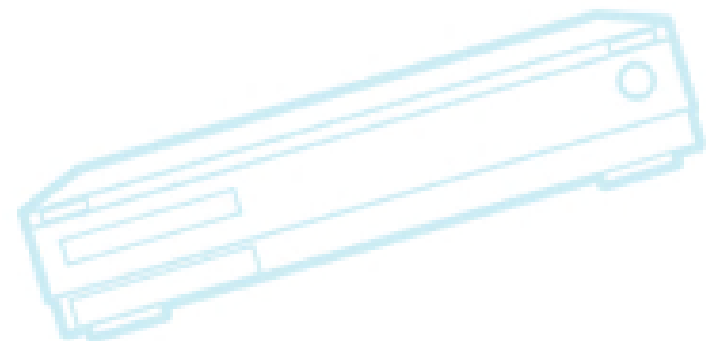
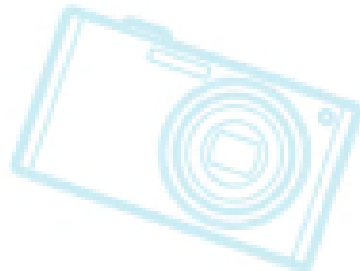
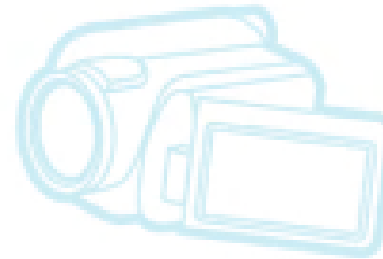
Building an android product -1-



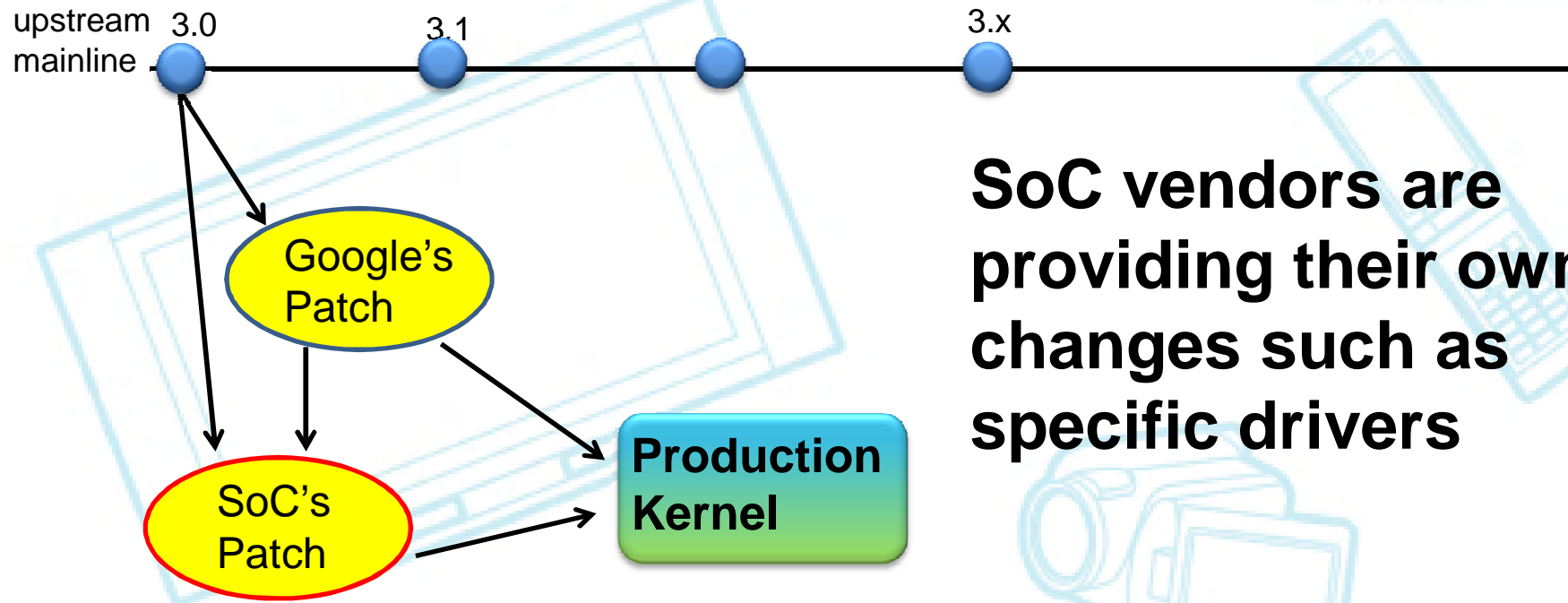
Google's Patch

Production Kernel

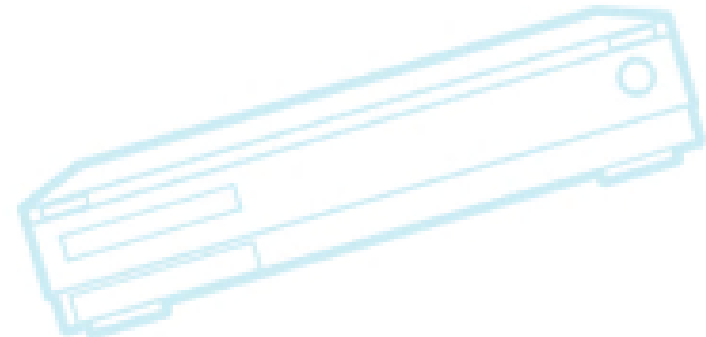
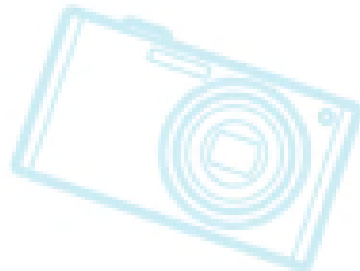
Google's patch on AOSP Is fundamental part



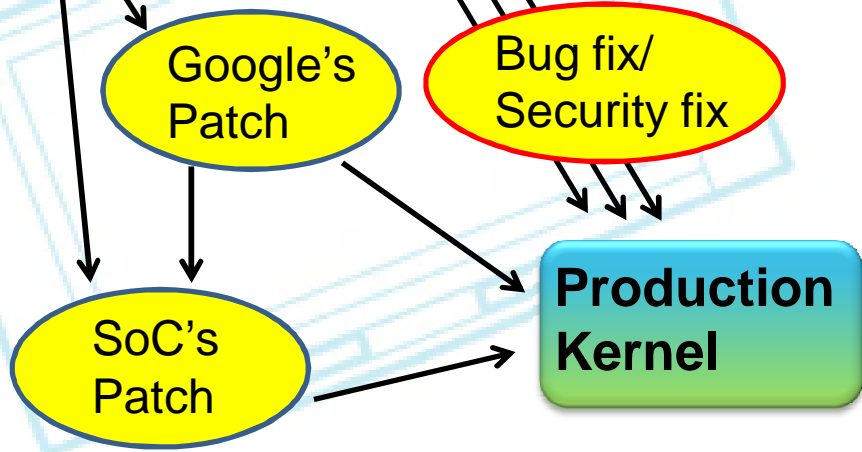
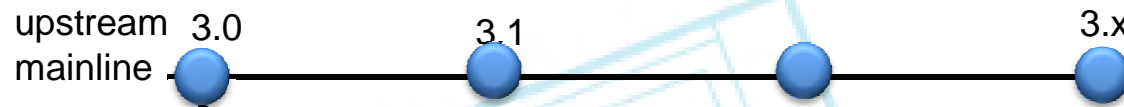
Building an android product -2-



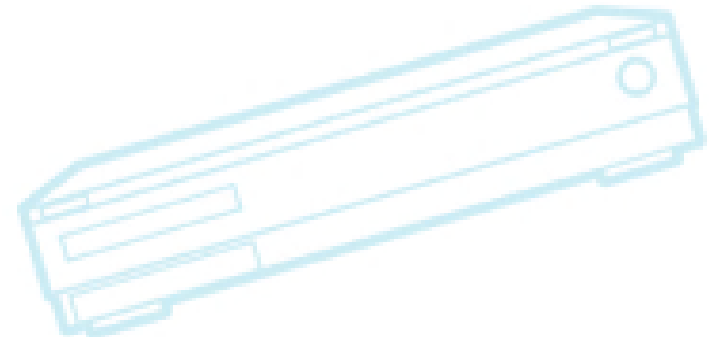
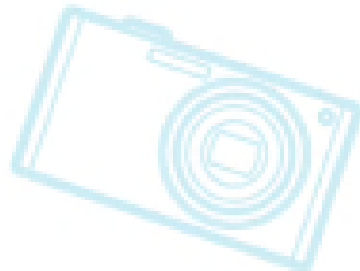
SoC vendors are providing their own changes such as specific drivers



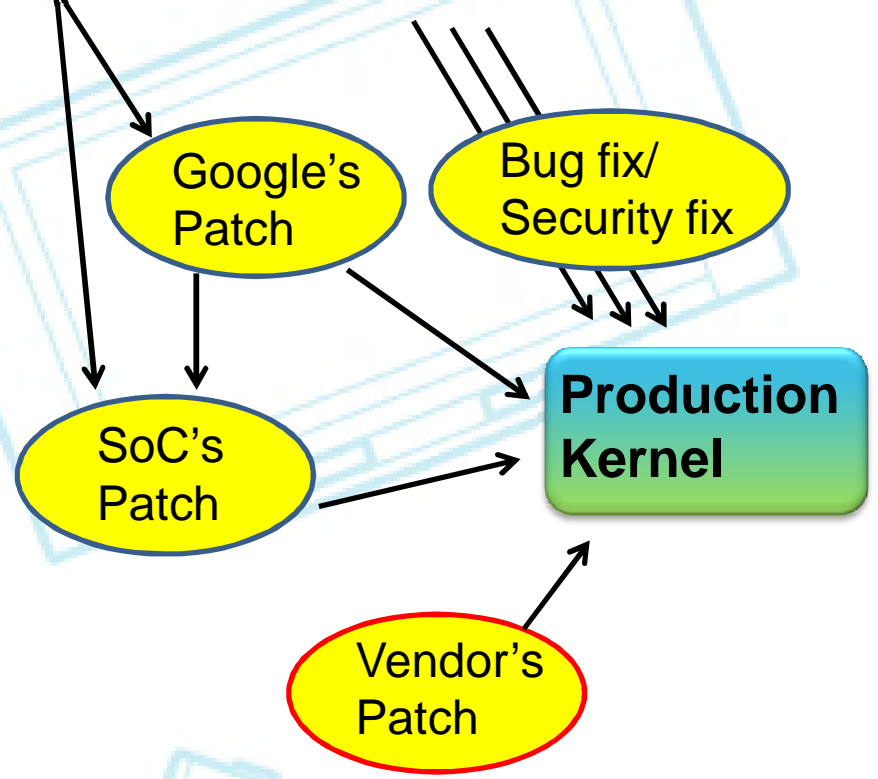
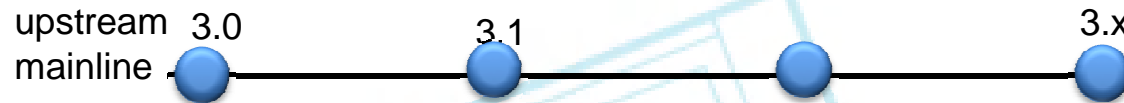
Building an android product -3-



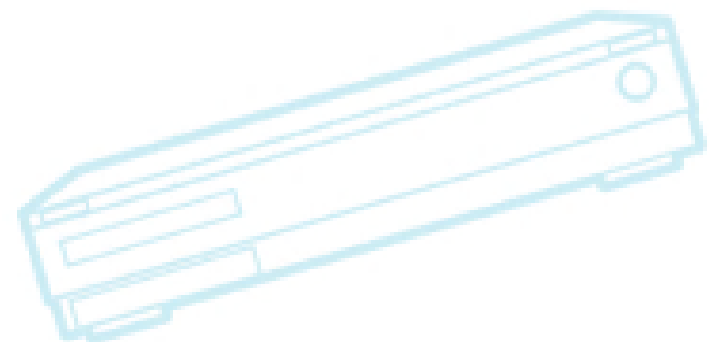
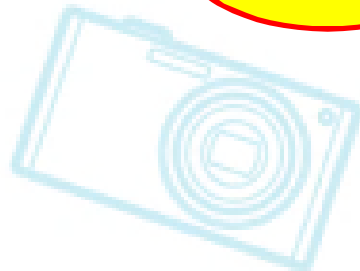
Bug fixes and security fixes which found in upstream need to be integrated into production kernel



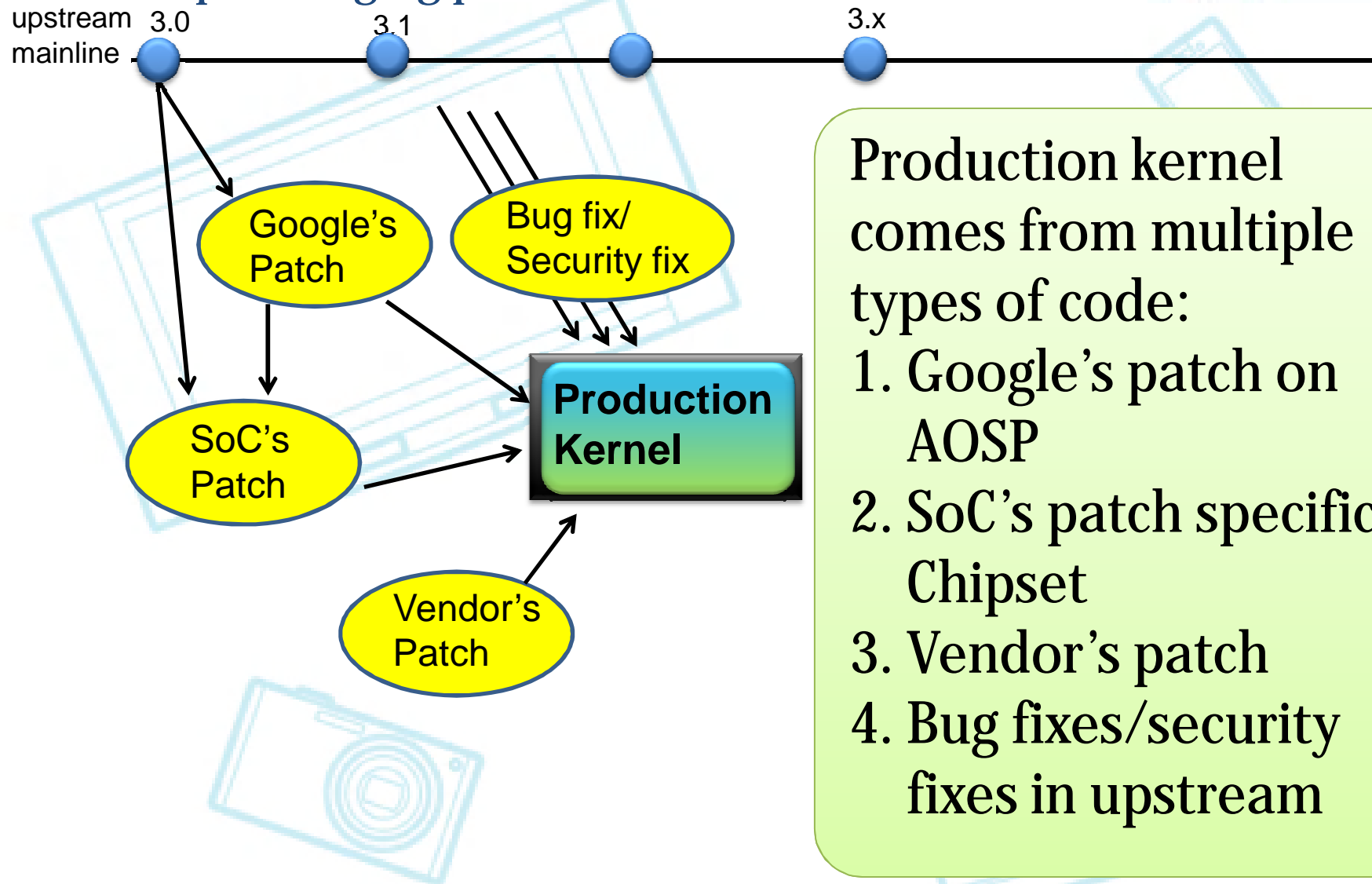
Building an android product -4-



Vendors are needed to add their own changes into their kernel



Building an android product with multiple merging process



Production kernel comes from multiple types of code:

1. Google's patch on AOSP
2. SoC's patch specific to Chipset
3. Vendor's patch
4. Bug fixes/security fixes in upstream

4 way merging process is really hard for manufacturers

Rapid Android release cycle



- Android is releasing about every 6 month
- Every time using latest kernel version

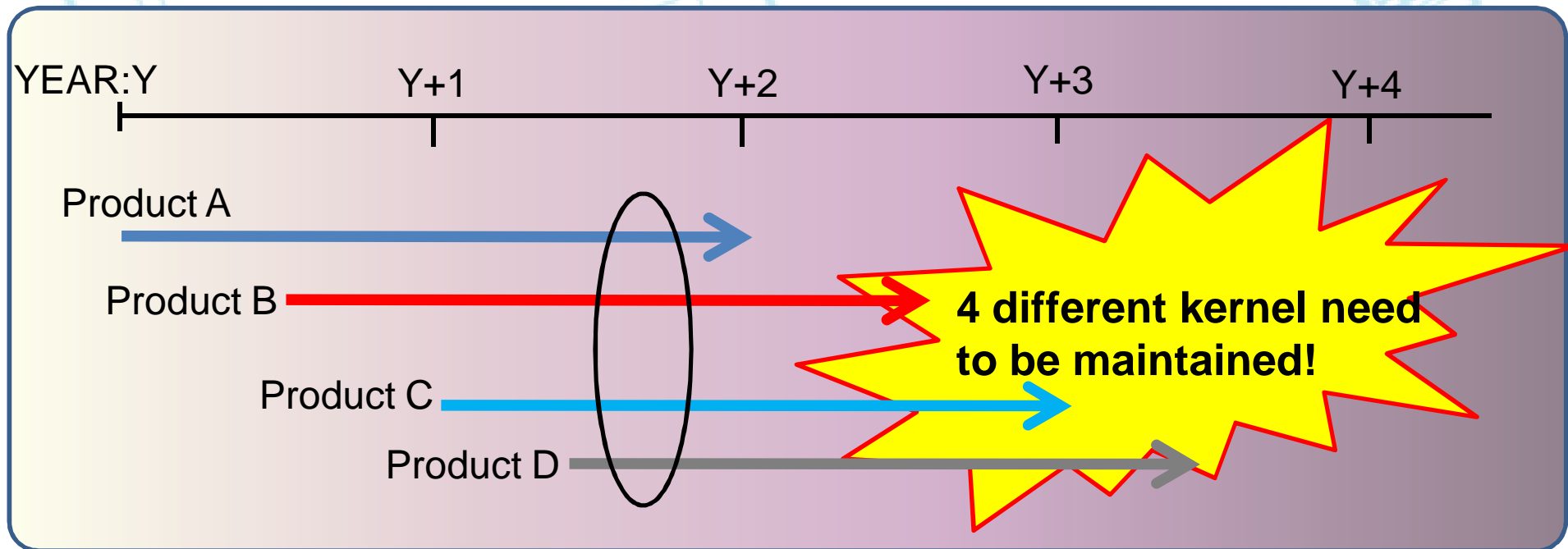
Android Version	Code Name	Kernel version	Release Date	Days
1.6	Donut	2.6.29	15,Sep, 2009	41
2.0	Éclair	2.6.29	26,Oct. 2009	206
2.2	Froyo	2.6.32	20,May. 2010	200
2.3	Ginger Bread	2.6.35	6,Dec. 2010	78
3.0	Honey Comb	2.6.36	22,Feb.2011	265
4.0	Ice Cream Sandwich	3.0	14,Nov.2011	

Stay same kernel at least a year could be great!

Manufacturer's product timeline



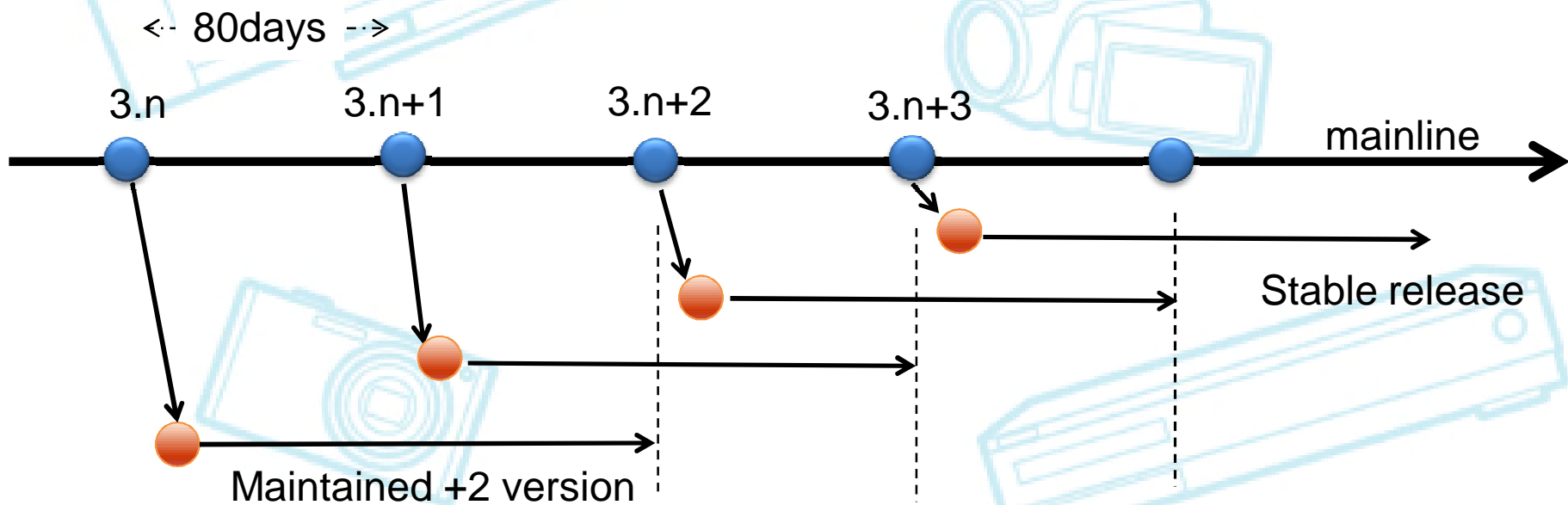
- A product life time is about 2 years
- Should provide bug fixes/security fixes for a production kernel within lifetime
- 4 different kernels need to be maintained in the future



Industry would like to use same kernel with reasonable lifecycle as a common ground of industry

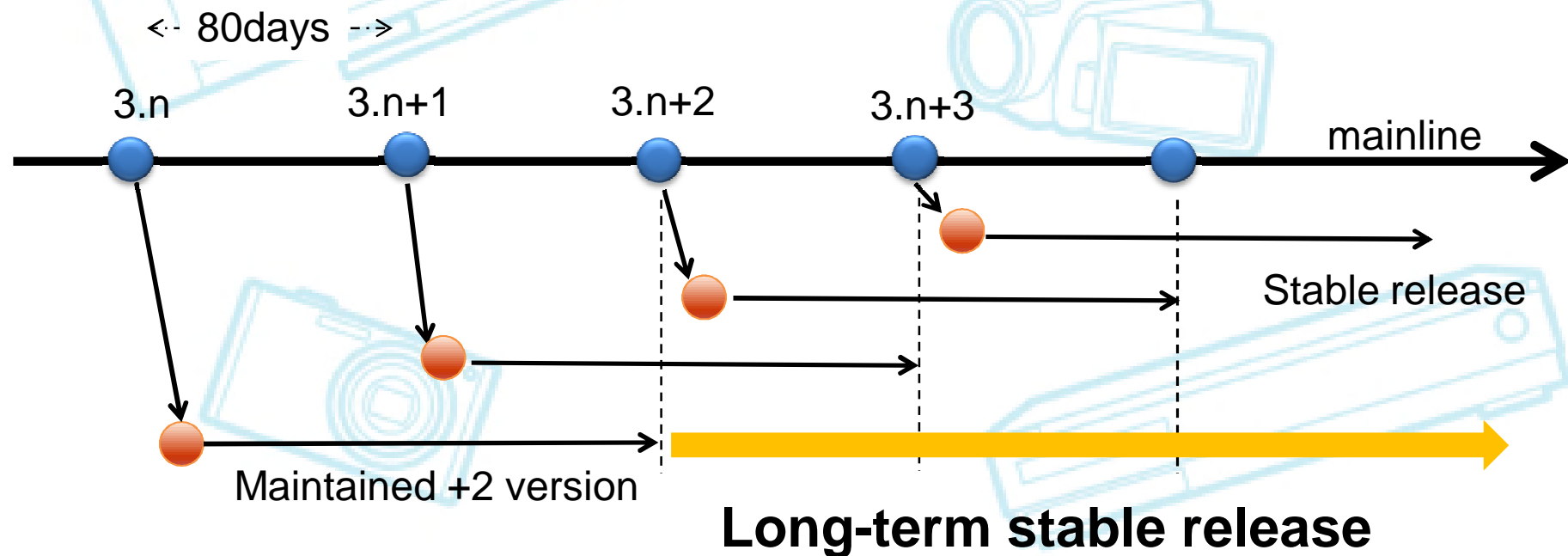
Stable kernel by the community

- After release of a kernel, stable release branch will start to maintain until $n+2$ version released
- Stable release will only include bug/security fixes found in latest version



Stable kernel by the community

- Some specific versions are maintained as “long-term stable”, LTS release
- For Enterprise use, there are Long-term stable release
 - 2.6.32 is used by Red Hat and SUSE for more that 4 years



Upstream patch submission

- Share the code, problem and fixes are valuable for the industry. Why it's inactive?

It's Open source.

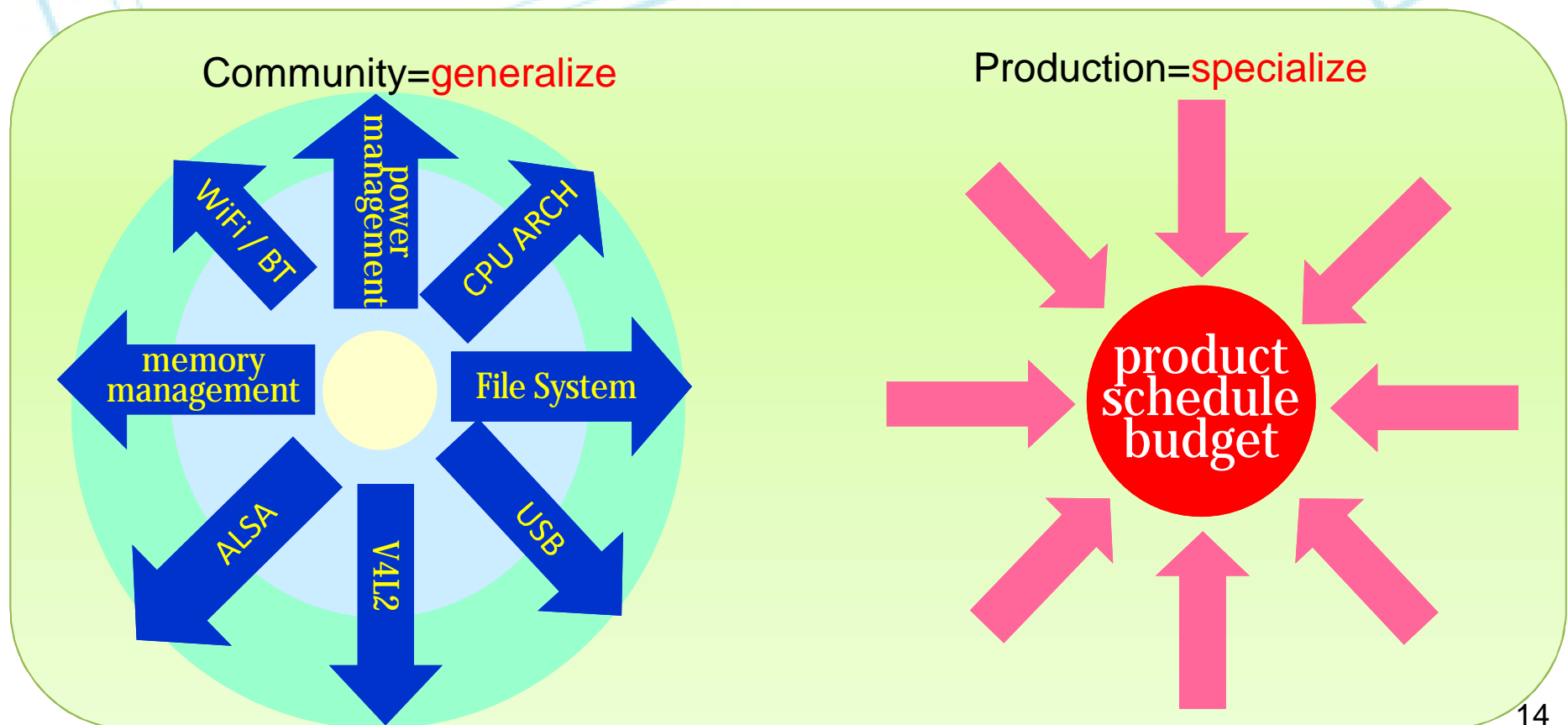
Non differentiation part could be shared among the community

Is Linux differentiator?

All the source code of Linux will be opened to public

Upstream patch submission

- Community and Industry have different direction and view
 - Such difference makes industry engineer hard to submit their code to the community

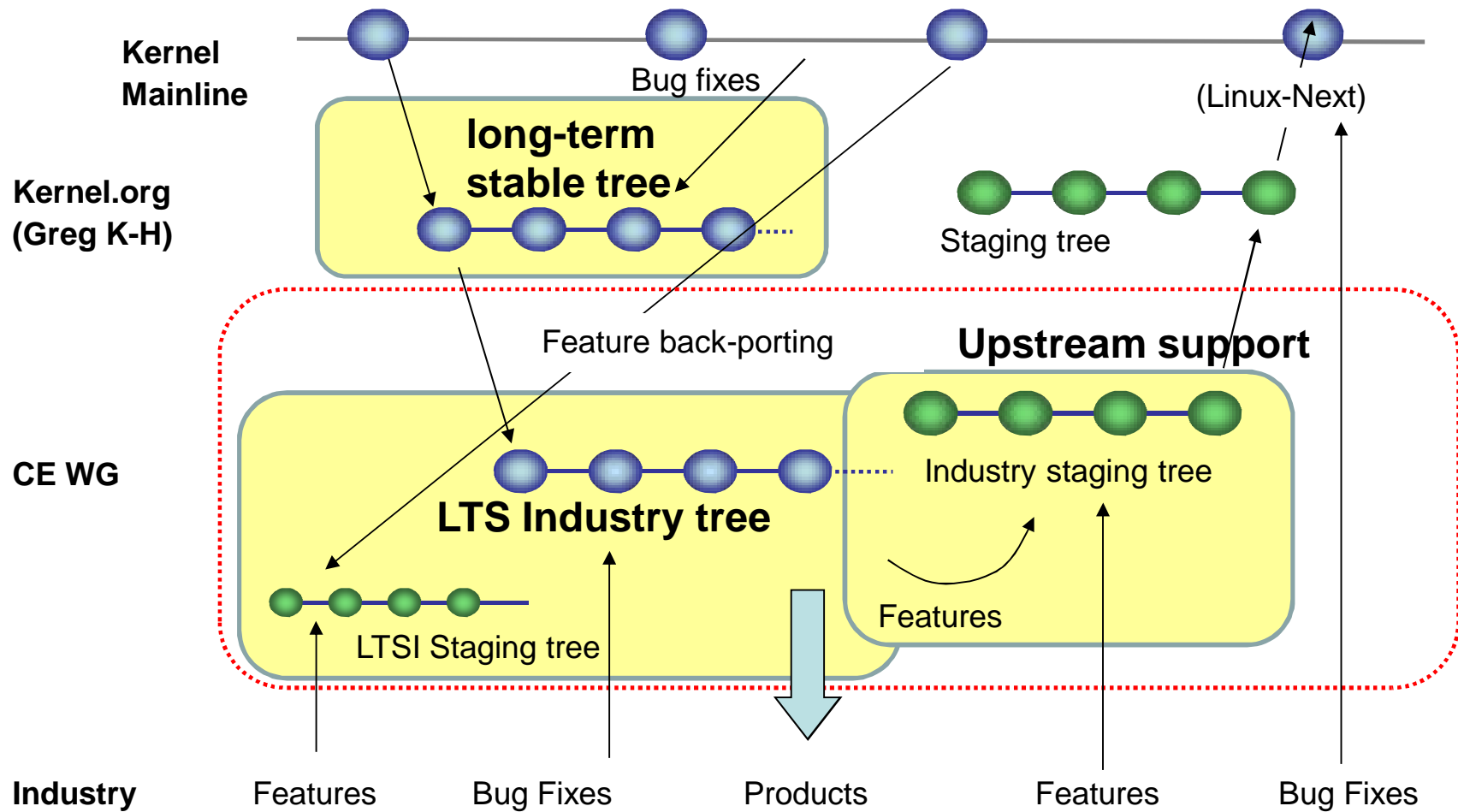


Project requirements

- **We need Long-term community based Linux kernel to cover embedded life cycle**
- **We need Industry managed kernel as a common ground for Embedded industry**
- **We need some mechanism to support the upstream activities for Embedded engineers**

Introduction: LTSI project overview

1. Long term stable kernel by the community
2. LTSI kernel
3. Upstream support



1. Long term Stable kernel by the community



- We have discussed with community to establish long-term stable kernel as a voice of embedded industry
 - One LTS per year, and 2 years lifetime
- Greg Kroah-Hartman would agreed with our request
 - 3.0 kernel was defined as long-term stable kernel
 - He became a fellow of the Linux Foundation
- We proposed to LF to establish Industry Advisory Board to define next LTS

From: Greg KH <greg-AT-kroah.com>
To: linux-kernel-AT-vger.kernel.org
Subject: Latest kernel stable/longterm status
Date: Mon, 9 Jan 2012 16:37:05 -0800
Message-ID: <20120110003705.GA9482@kroah.com>
Cc: stable-AT-vger.kernel.org

As 3.2 is now out, here's a note as to the current status of the different stable/longterm kernel trees.

... some lines removed

Here's the different active kernel versions that I am maintaining at the moment:

3.2.y - this will be maintained until 3.3 comes out

3.1.y - there will be only one, maybe two, more releases of this tree

3.0.y - this is the new "longterm" kernel release, it will be maintained for 2 years at the minimum by me.

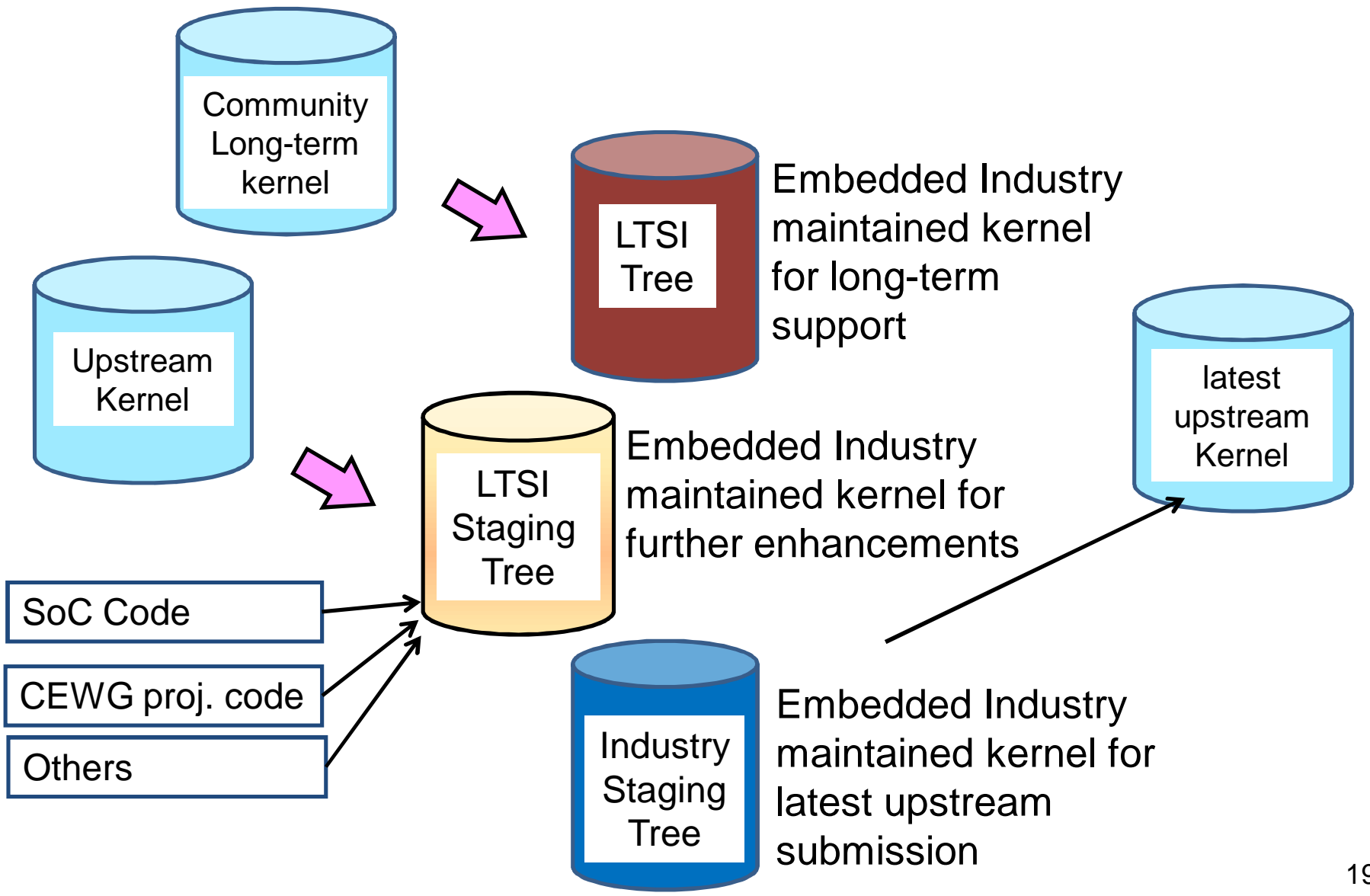
2.6.32.y - this is the previous "longterm" kernel release. It is approaching it's end-of-life, and I think I only have another month or so doing releases of this. After I am finished with it, it might be picked up by someone else, but I'm not going to promise anything.

All other longterm kernels are being maintained in various forms (usually quite sporadically, if at all), by other people, and I can not speak for their lifetime at all, that is up to those individuals. I

...

thanks,
greg k-h

2.LTSI kernel trees



LTSI tree definition

- Linux kernel maintained by Industry Initiative: LTSI
 - not a part of upstream community
- Keep maintained in 2 years
- Define a version yearly (initially 3.0)
- Use community long-term stable [LTS] version as base kernel
 - bug and security-fix will be provided timely fashion
- Useful for Industry
 - No experimental features are integrated
 - Industry engineers can be shared the problem and fix it
 - Stable and Replaceable
- Add some additional stuff to obtain industry requirement
 - new kernel feature support (e.g. Android patch set , device drivers)
 - SoC/manufacturer in-house code (after sanity check)
 - CEWG funded open source project result

LTSI staging tree definition

- Tree for the feature enhancements for LTSI
 - Same version as LTSI tree
- All code flow into LTSI kernel must be merged through this staging tree
 - Newly mainlined kernel code (back ported device driver, others).
 - CEWG funded project result
 - SoC Vendor code (SoC vendor can send merge request)
- Before feature freeze of LTSI tree, this is the place to collect the patch
- After feature freeze, proposed patch can be merged in this tree
 - Industry engineer can get latest back ported code from this tree
 - Experimental patch can be discussed here and can get help to be mainlined

Industry staging tree definition



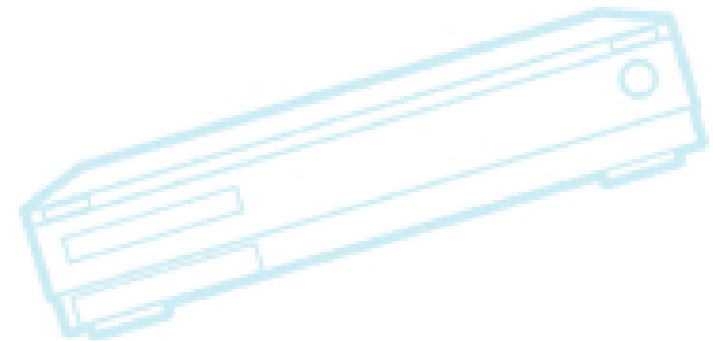
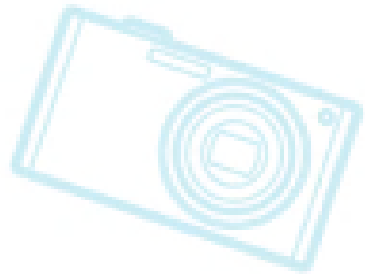
- Tree for the feature enhancements for upstream submission
 - Keep latest upstream version to send patch for current mainline or linux-next
 - Provide consultation and review, patches from industry engineers for upstream submission
- Engineers who have experience and skill to communicate with the community can use this
 - Be able to share opinion with industry engineers
 - Be able to get consultation and review from LTSI staff

3. Upstream support

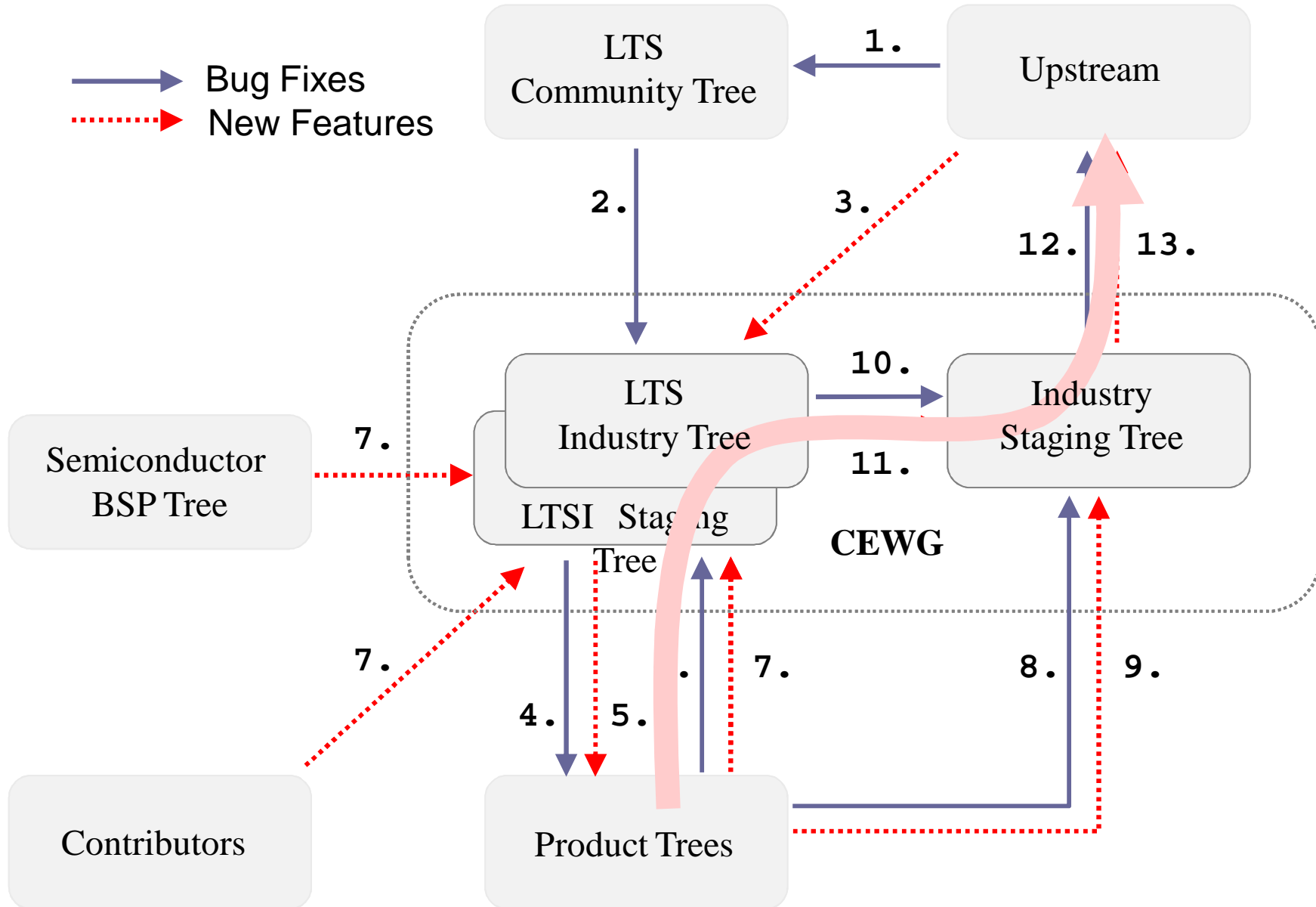
- Why upstream ?
 - 4 way merging process is running every time
 - 3 way, 2 way process gives greatly reduce actual works = Reduce development cost!
 - Holding manufacturer's own fixes also costs
 - Upstream changes sometimes makes problem not to apply own changes. In such case, Engineer should take their time to resolve it. It's also cost!
 - Finally, every single piece of code will be in public because of GPL

3. Upstream support

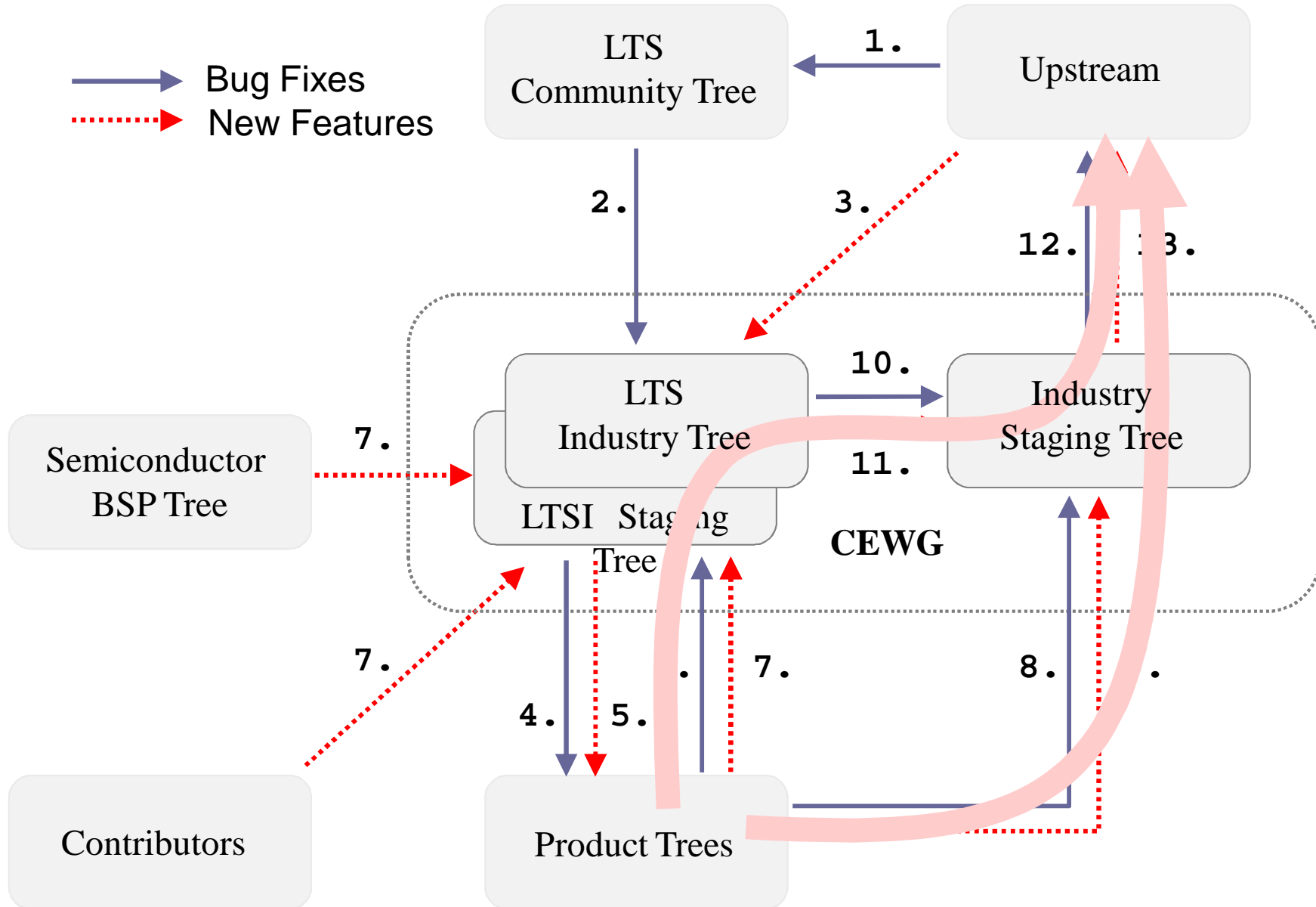
- Submitting patch into upstream. What's difficult?
 - Porting to latest kernel is not easy
 - Industry Engineer have No guidance, No consultation, No review
 - Communication with outside of company is always not easy



Upstream support in LTSI: Accept Patches for LTSI version and port it

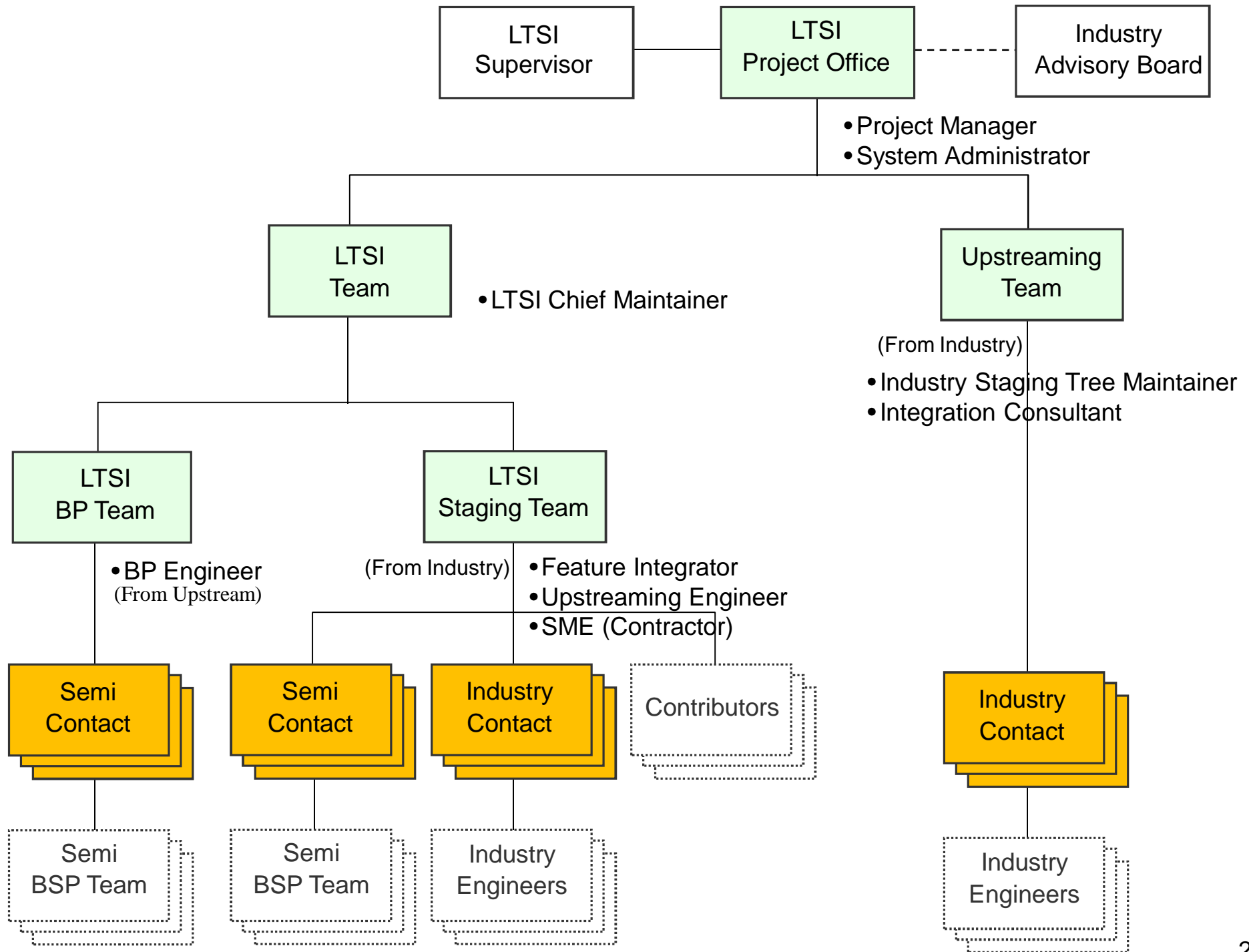


Upstream support in LTSI: Accept Patches for LTSI version and port it



Industry Contact : What?

- Industry Contact is:
 - Contact person for the company to communicate with LTSI and community
 - LTST team staff provide help to company's engineer though Industry Contact such as sending/reviewing the patch and so on.
- LTSI helps Industry Contacts :
 - More easier to talk with community
 - Provide the place to share information among Industry Contacts
 - Industry Contact Meeting
- CEWG member company can use this



How you can participate LTSI

- Follow on Twitter account:
@LinuxLTSI
- Watch the web: *will be in public soon*
<http://ltsi.linuxfoundation.org/>
- Join mailing list: *will be in public soon*
- Join CE Working group
 - CE Working Group is the place to discuss the feature to be included in the initial staging tree
 - LF corporate member can participate CEWG
 - Register Industry Contact for the company

SUMMARY

- LTSI provides stable kernel.
 - Define a kernel version in every year
 - and maintain about 2 years
 - During 2 years, bug/security fixes are come from community's long-term kernel
- LTSI includes features back ported from upstream, important features for embedded, SoC code and industry changes
 - Can reduce your development cost by sharing code
- LTSI supports mainlining industry's patch
 - Accept patches as same as LTSI version
 - Provide review and recommendations to the patches
 - Provide opportunity to share/learn upstream activities by the Industry Contact meeting



THANK YOU



Another presentation will be held in
Embedded Linux Conference
on Wednesday 15
See you then!