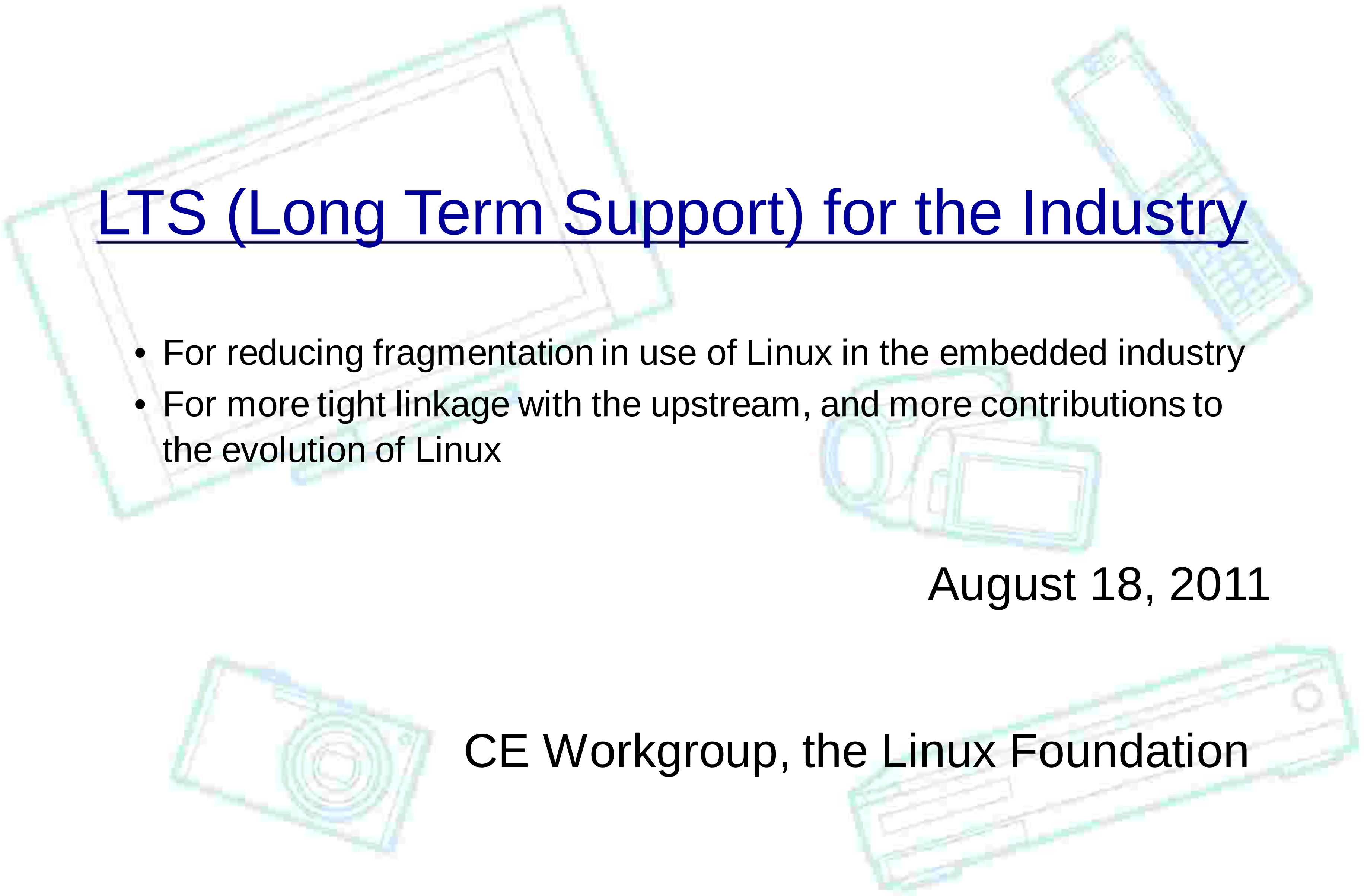


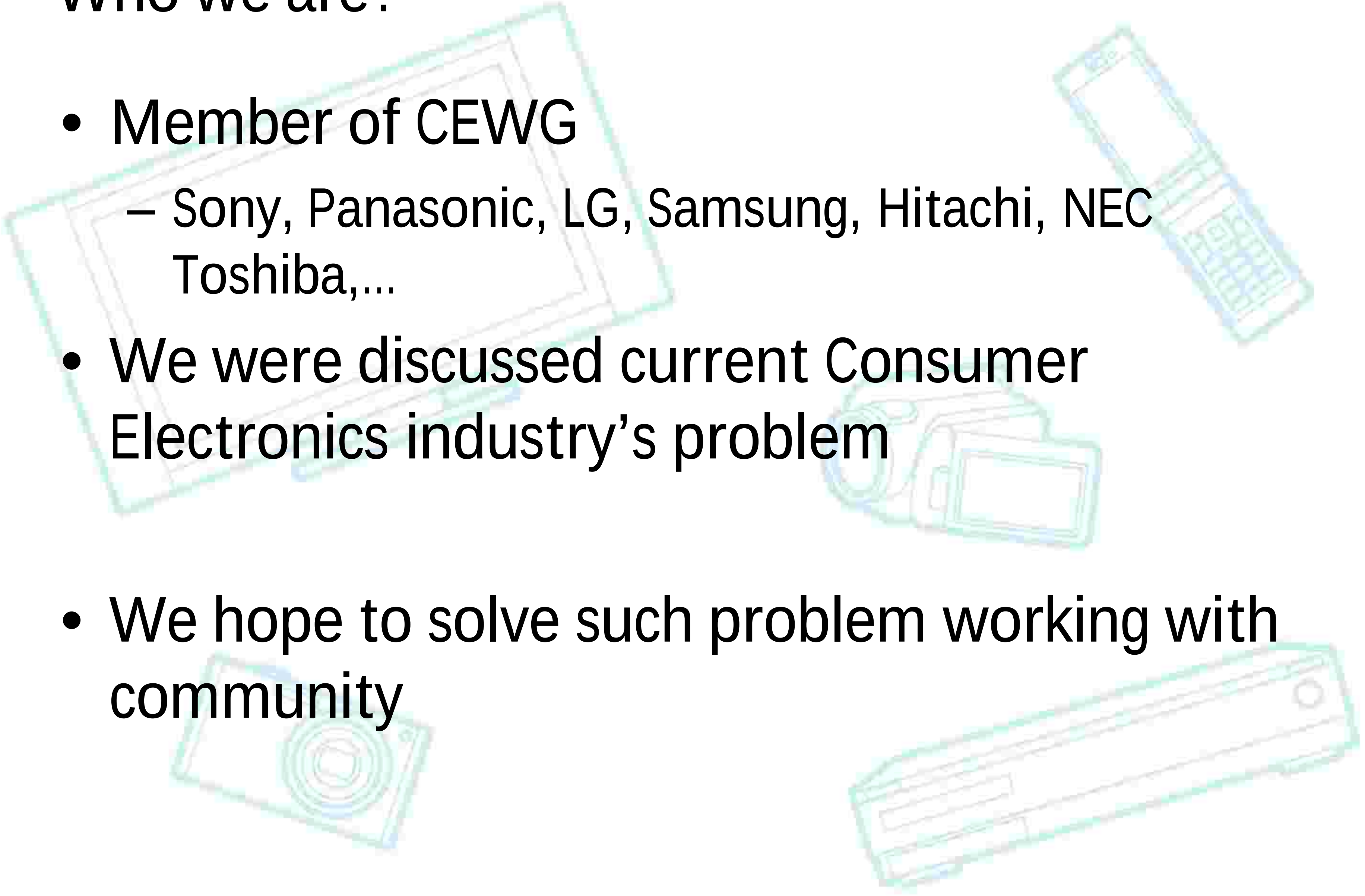
LTS (Long Term Support) for the Industry

- For reducing fragmentation in use of Linux in the embedded industry
- For more tight linkage with the upstream, and more contributions to the evolution of Linux

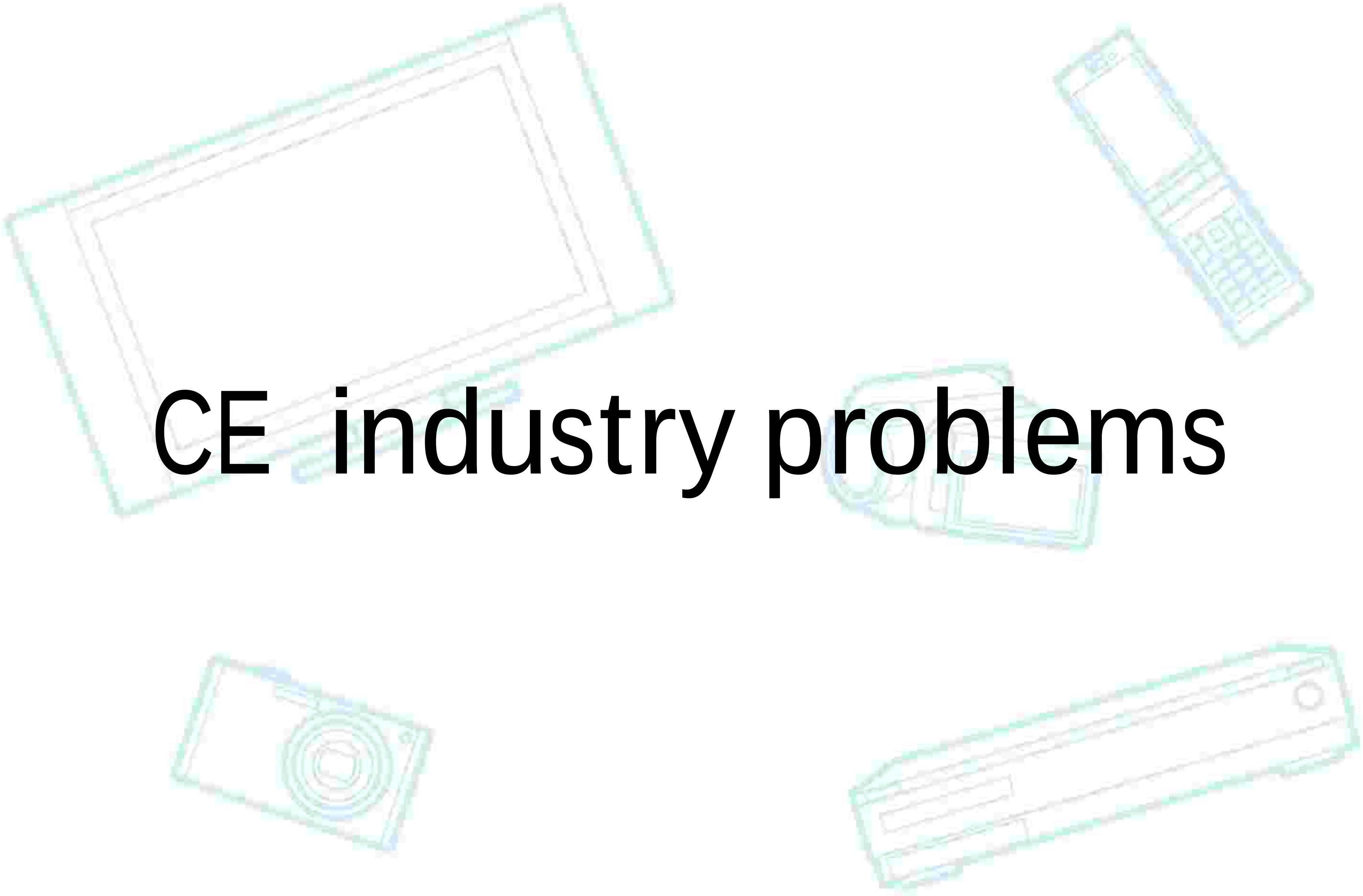
August 18, 2011

CE Workgroup, the Linux Foundation

Who we are?



- Member of CEWG
  - Sony, Panasonic, LG, Samsung, Hitachi, NEC Toshiba,…
- We were discussed current Consumer Electronics industry’s problem
- We hope to solve such problem working with community



CE industry problems

Problem 1: Shorter product lifetime compared to Enterprise product

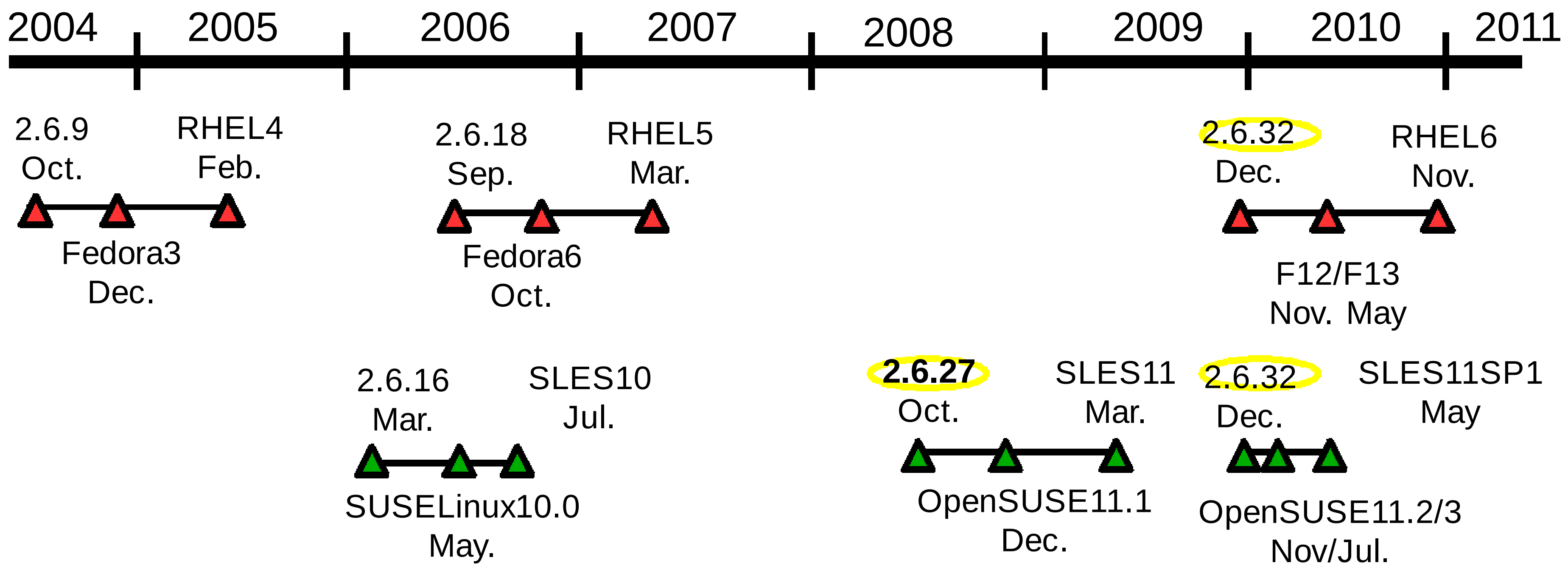
- Lifetime of the Enterprise industry is about 5 to 10 years
  - We expect that RHEL will be refresh the kernel version about 4-5 years of cycles
  - Latest RHEL is using 2.6.32.x (long term) and new long term may need to establish 3-4 years later
- Lifetime of the Consumer products are 1 to 3 years
  - CE industry need to refresh kernel every year but there no such infrastructure.
  - 2.6.35 had been established as long term just last year. (No follow on long term had been discussed)

Linux upstream kernel and distros

Enterprise distro take 9-12 months for the integration/ testing to provide commercial quality products from upstream Kernel

Release period is about 3-4 years

RHEL6 and SLES11 is using 2.6.32 LTS



Problem 2: no common ground for embedded (version gaps makes collaboration tough)

- Android and MeeGo is releasing every 6 month with latest kernel
  - Latest kernel have innovative features
  - SoCs is providing BSP for the kernel every time without support
  - Every Manufacture use the BSP and do QA work for every time
- Manufacture and SoC needs some time for system level verification with single solid kernel.
  - Industry want to reuse same kernel for a few generation before move to next kernel.

Problem 3: upstream patch submission from embedded is still very inactive

- Embedded industry production team engineer pays huge attention to Linux, especially device driver code quality.

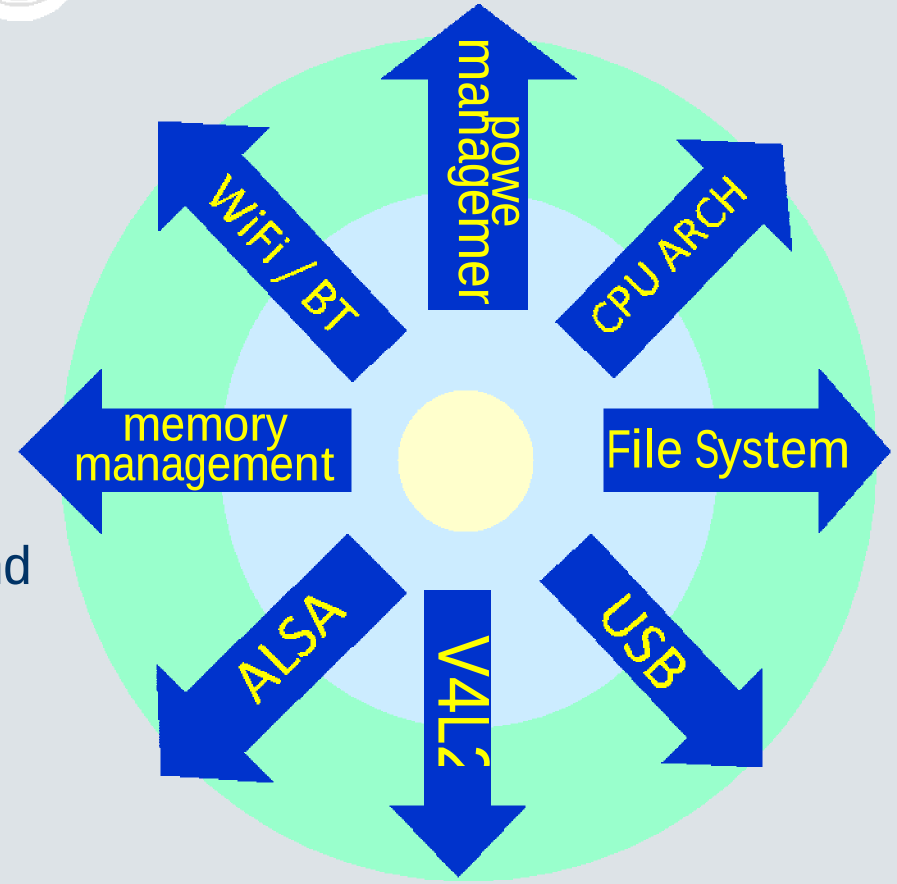
- And they pretty likely modify device driver code to improve system stability and/or performance.

- Therefore embedded industry developer should own some good and important patches in house, however patch submission from these developer is very low.

- As these codes are not mainlined, they need to apply same enhancement when they adopt new kernel.

Upstream principals = divergence

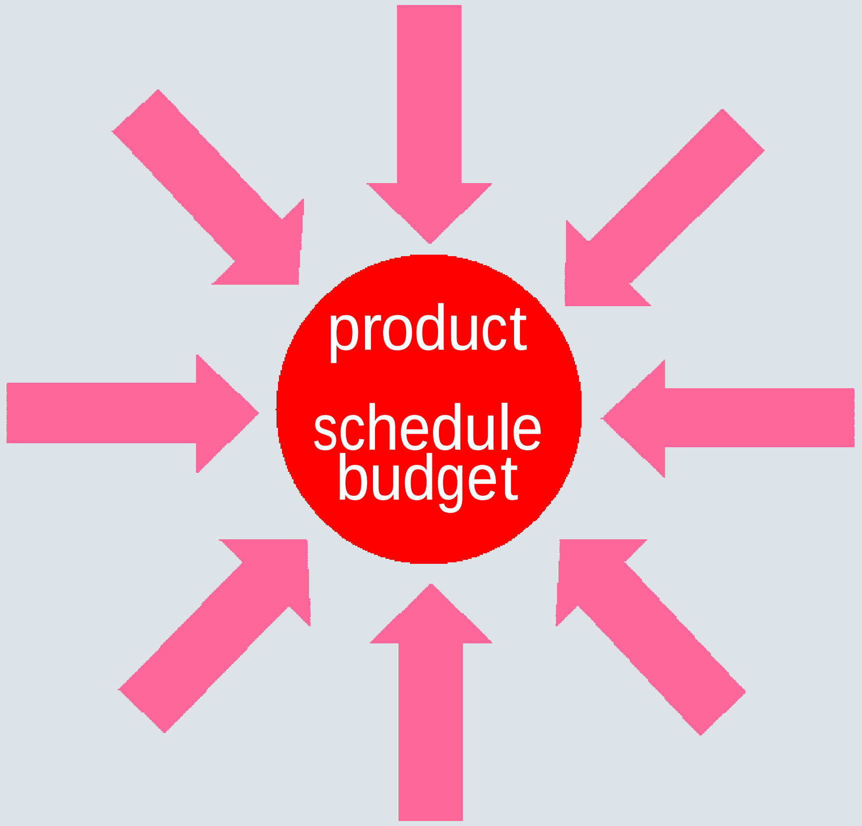
- Think sustainable evolution
  - random technical improve
  - no specific shred narrow target
  - allow diversity
- Ever lasting development
  - no specific due date
  - Think for better future
  - incremental improve
  - moving target depends on demand
- Fair governance
  - Completely open
  - purely technical (for best)
  - volunteer contribution basis



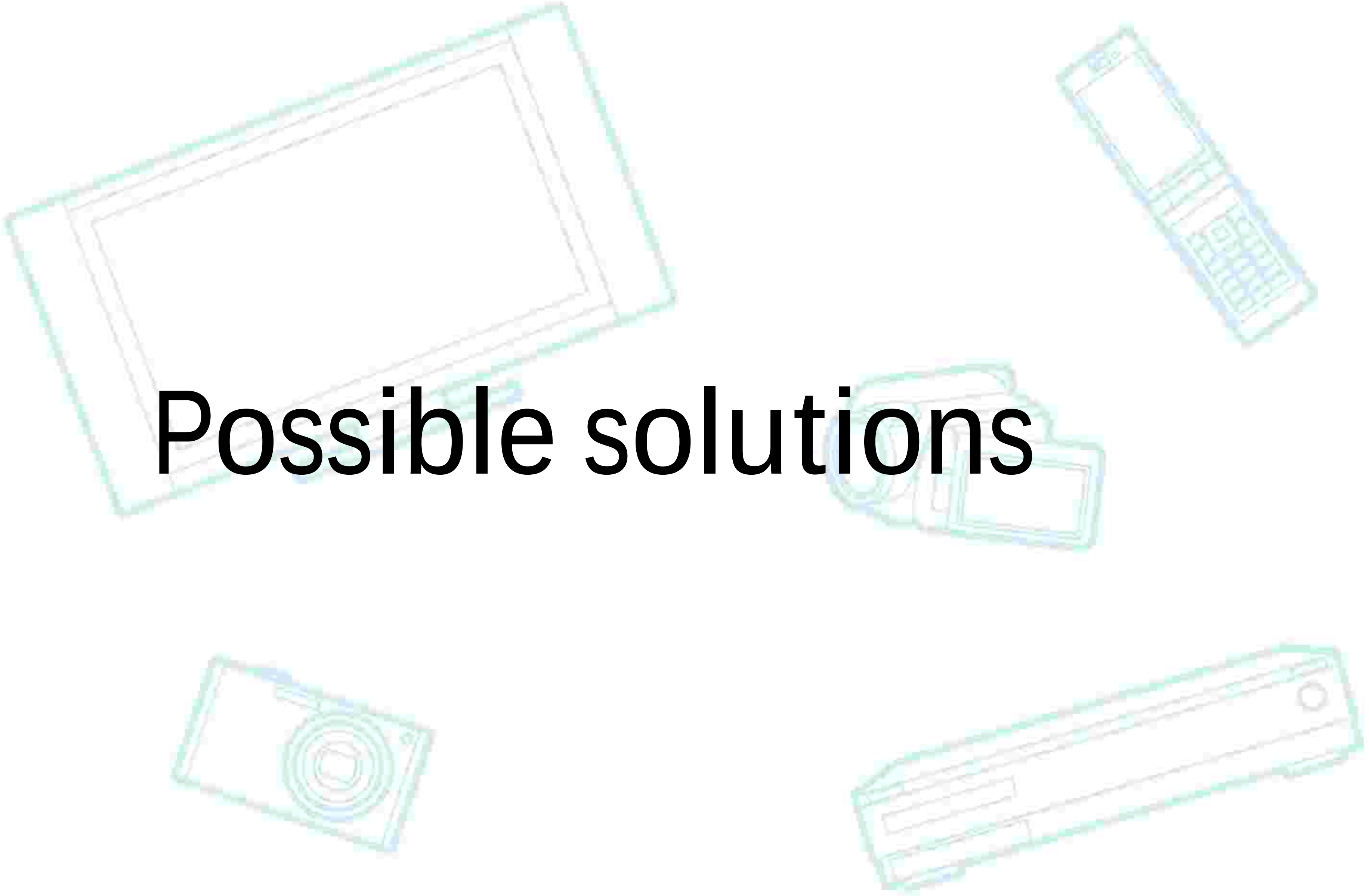
Upstream guys work for unified better future for all

Production principals = convergence

- Clear production goal
  - strict release due date
  - sever performance target
  - high cost pressure
- One shot development
  - allow interim solution
  - average skilled engineer
  - relatively large team
- Quality requirement
  - product liability demand
  - limited use case
  - reset is not allowed

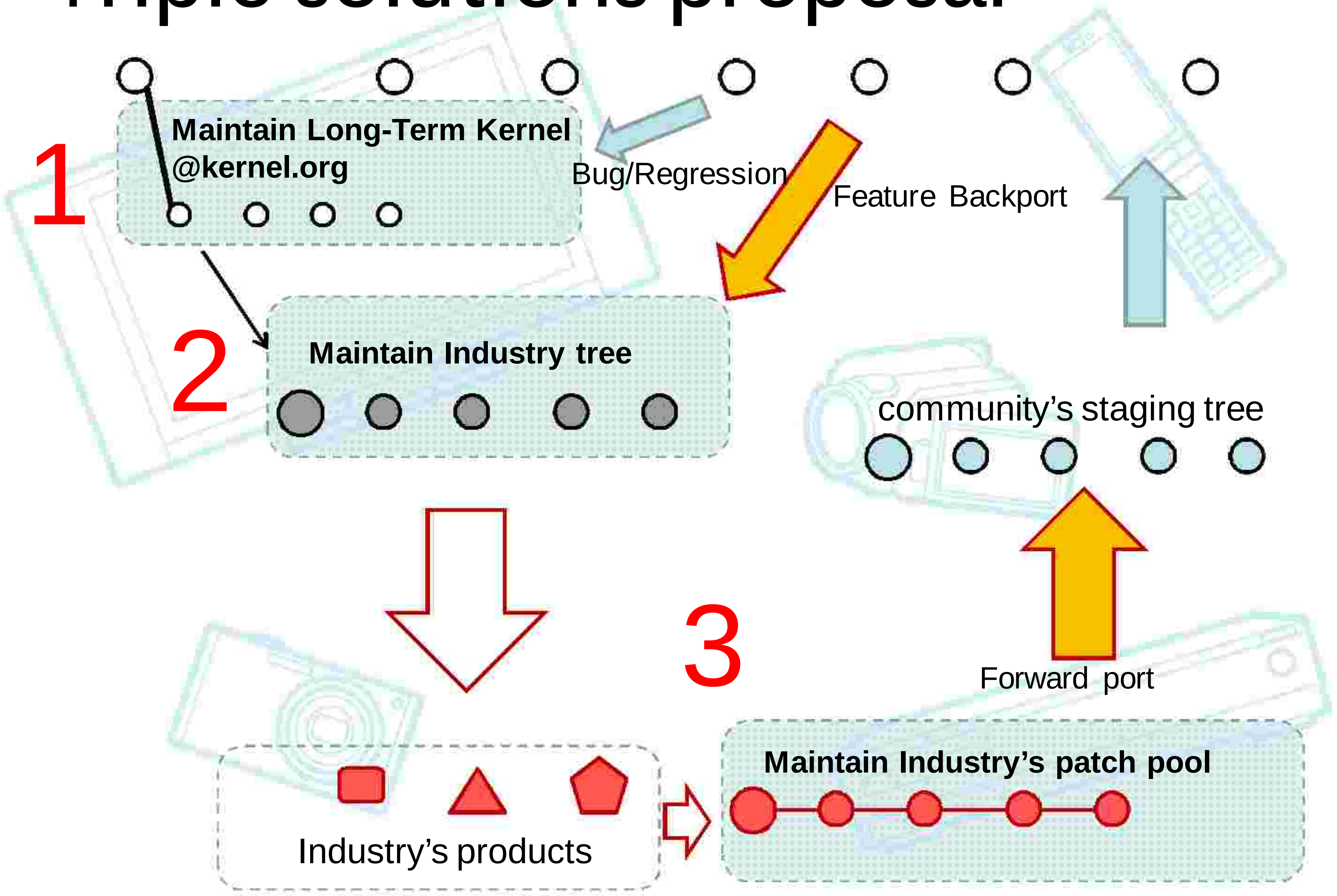


Industry developer work for their current particular product

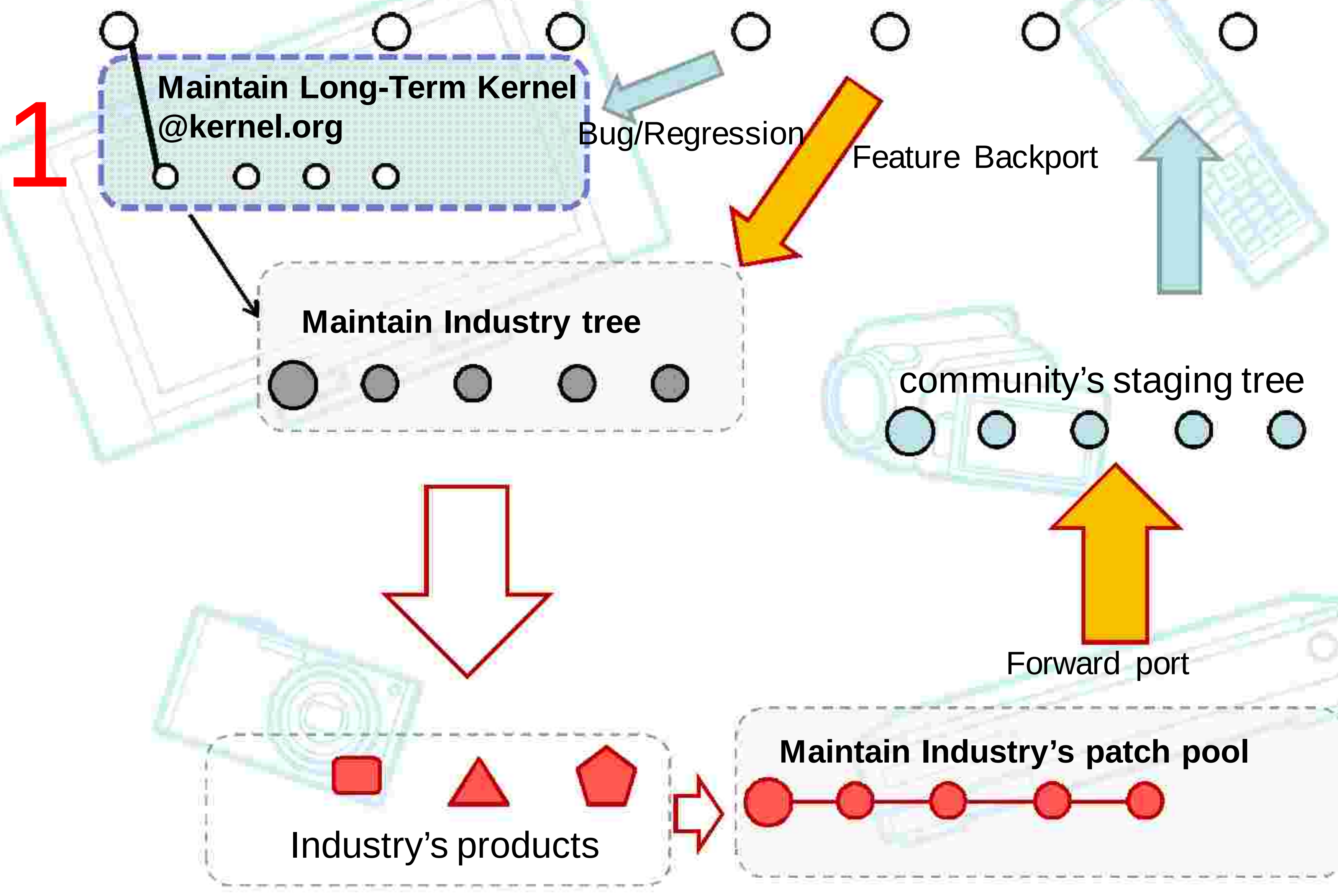


Possible solutions

Triple solutions proposal

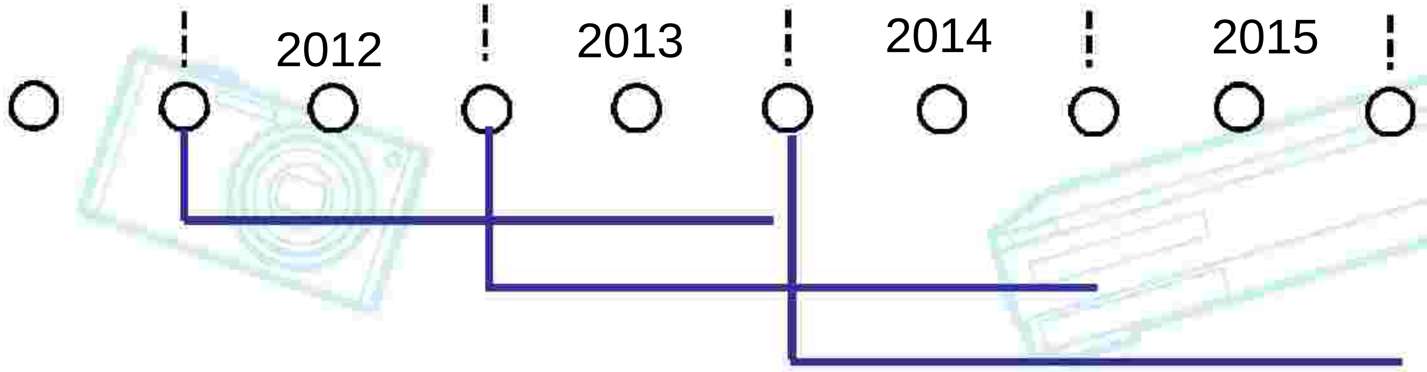


1st solution = community LTS

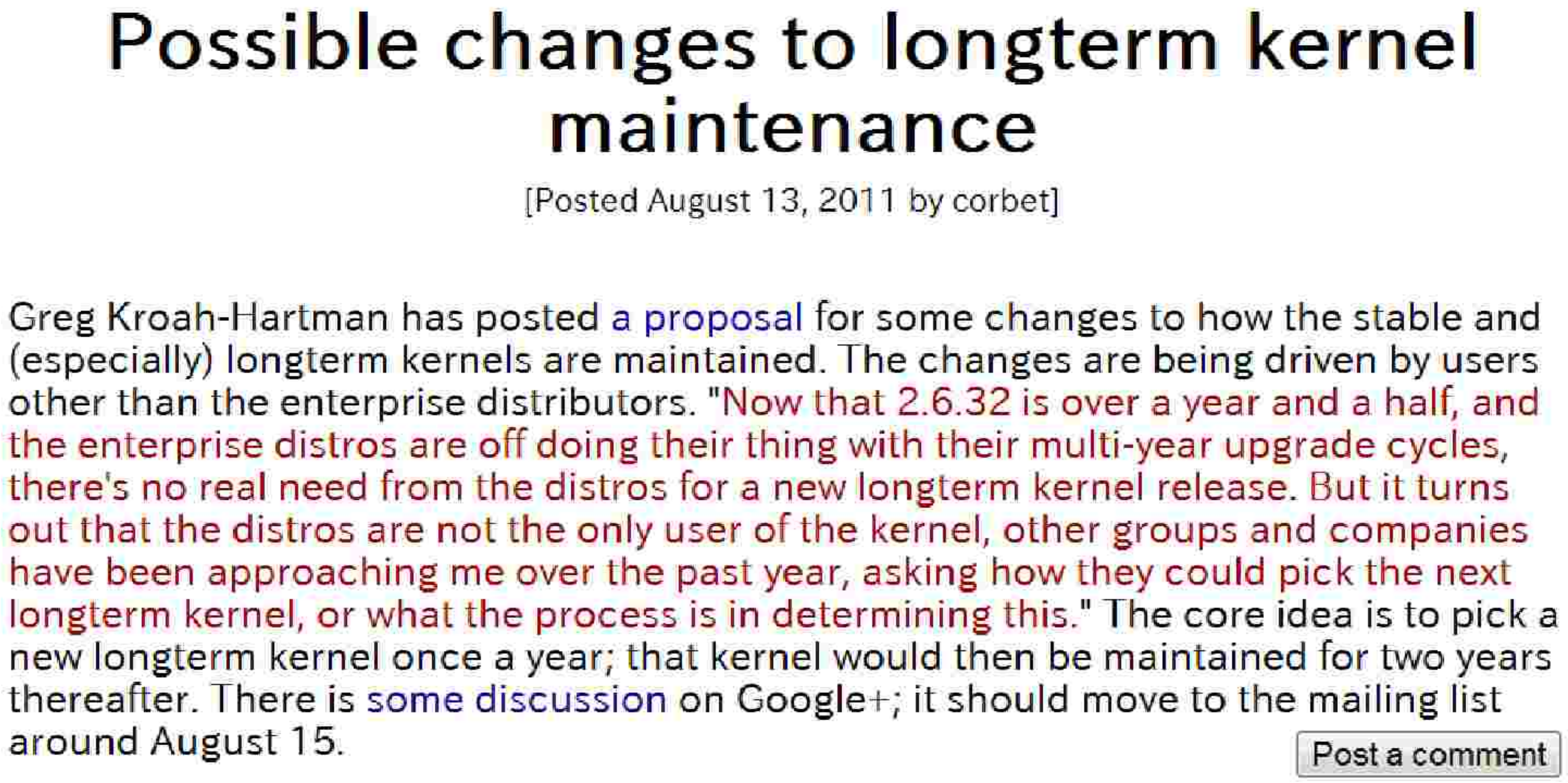


1st solution = community LTS

- CEWG would like to establish and maintain long term kernel
  - Pick one candidate kernel in year and maintain it for 2 years lifetime.
  - Up to 2 of long-term kernel will be maintained
  - Maintained by the stable-kernel rule
  - CEWG would like to contribute this activity and publish it on kernel.org



New community kernel longterm maintenance scheme proposed by Greg Kroah-Hartman



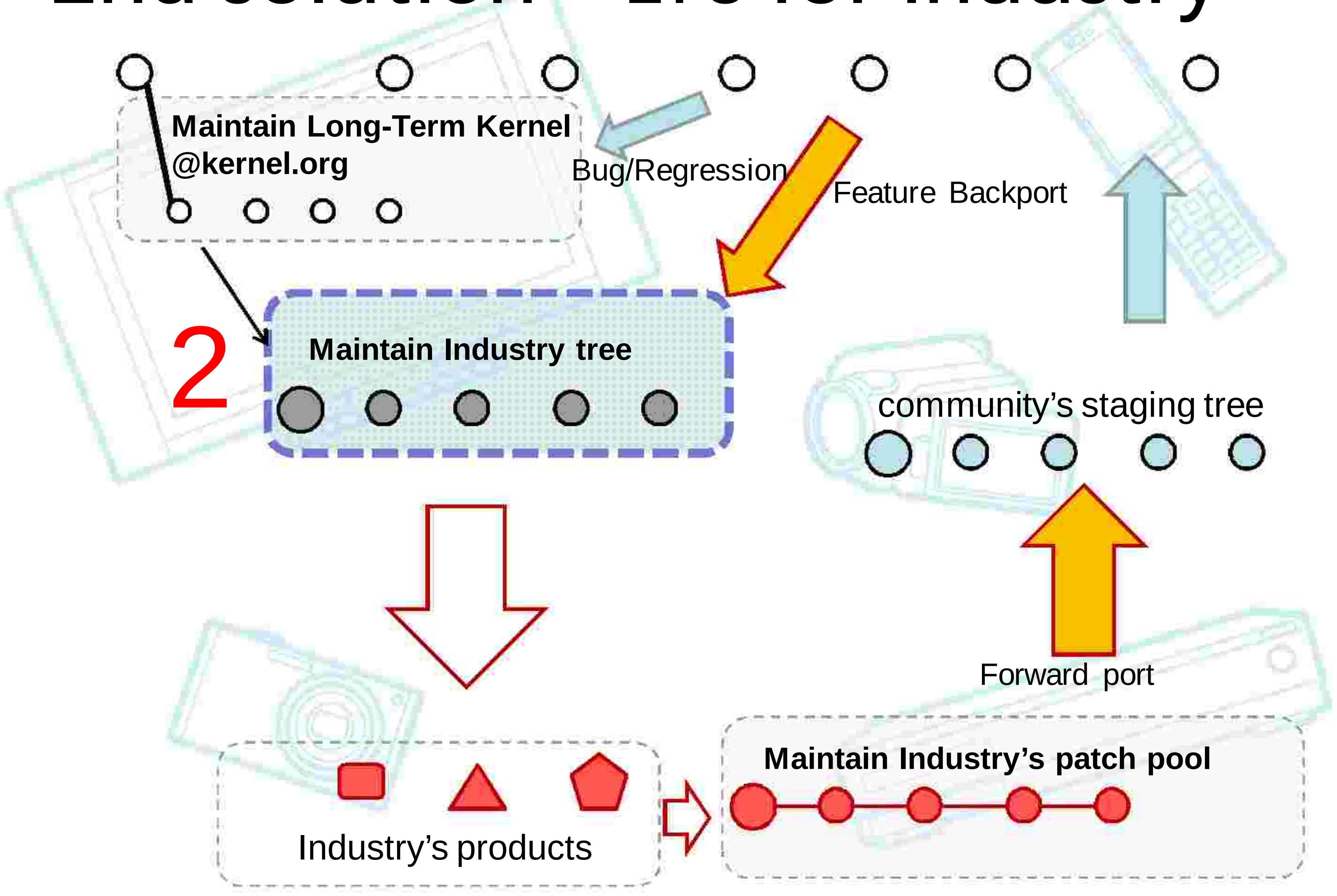
Possible changes to longterm kernel maintenance

[Posted August 13, 2011 by corbet]

Greg Kroah-Hartman has posted a proposal for some changes to how the stable and (especially) longterm kernels are maintained. The changes are being driven by users other than the enterprise distributors. "Now that 2.6.32 is over a year and a half, and the enterprise distros are off doing their thing with their multi-year upgrade cycles, there's no real need from the distros for a new longterm kernel release. But it turns out that the distros are not the only user of the kernel, other groups and companies have been approaching me over the past year, asking how they could pick the next longterm kernel, or what the process is in determining this." The core idea is to pick a new longterm kernel once a year; that kernel would then be maintained for two years thereafter. There is some discussion on Google+; it should move to the mailing list around August 15.

http://lwn.net/Articles/454886/

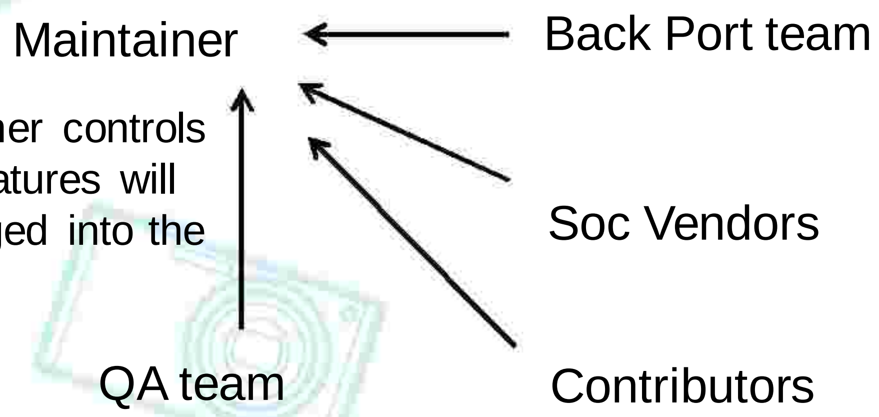
2nd solution = LTS for industry



2nd solution = LTS for industry

- Based on long term kernel (= 1st solution)
  - Accept adding new device driver*1
  - Accept performance improvement*1

    1: these codes must be mainlined (or at least queued)
  - Conduct pre-defined test before release
- The tree will be published by CEWG



Back port team will be back port some features from upstream

Maintainer controls what features will be merged into the tree

SoC vendors can contribute their specific code to the tree. (this could be good for manufactures)

Any of industry engineers can contribute to back port the features

Conduct pre-defined test case before we release LTSI code.

Expected SoC vendors correspondence

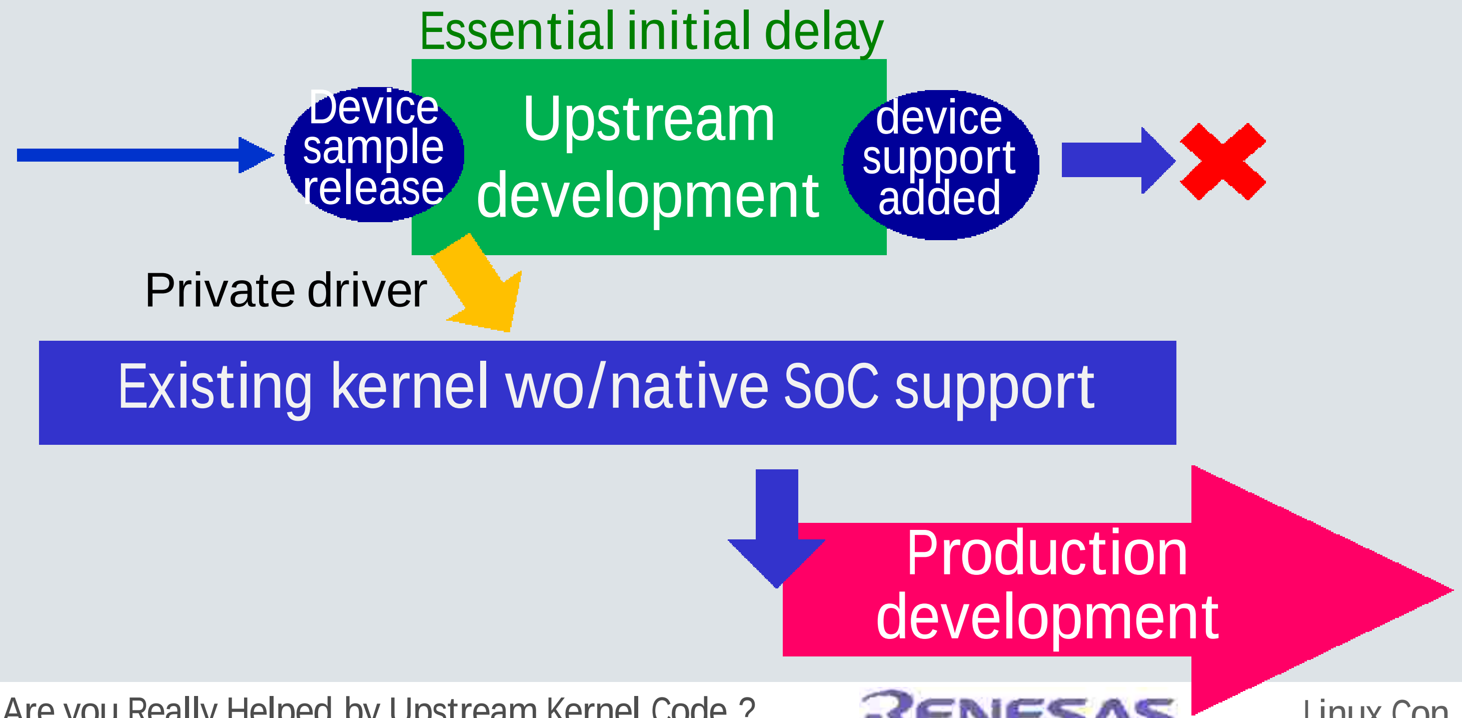
- To add new driver and/or platform to LTSI

1) Submit these code to current latest upstream
2) Get community review, feedback
3) Adjust code based on 2) to fit upstream
4) Code queued (or merged)

5) Then backport that code to LTS for industry
6) We expect test code will come with code

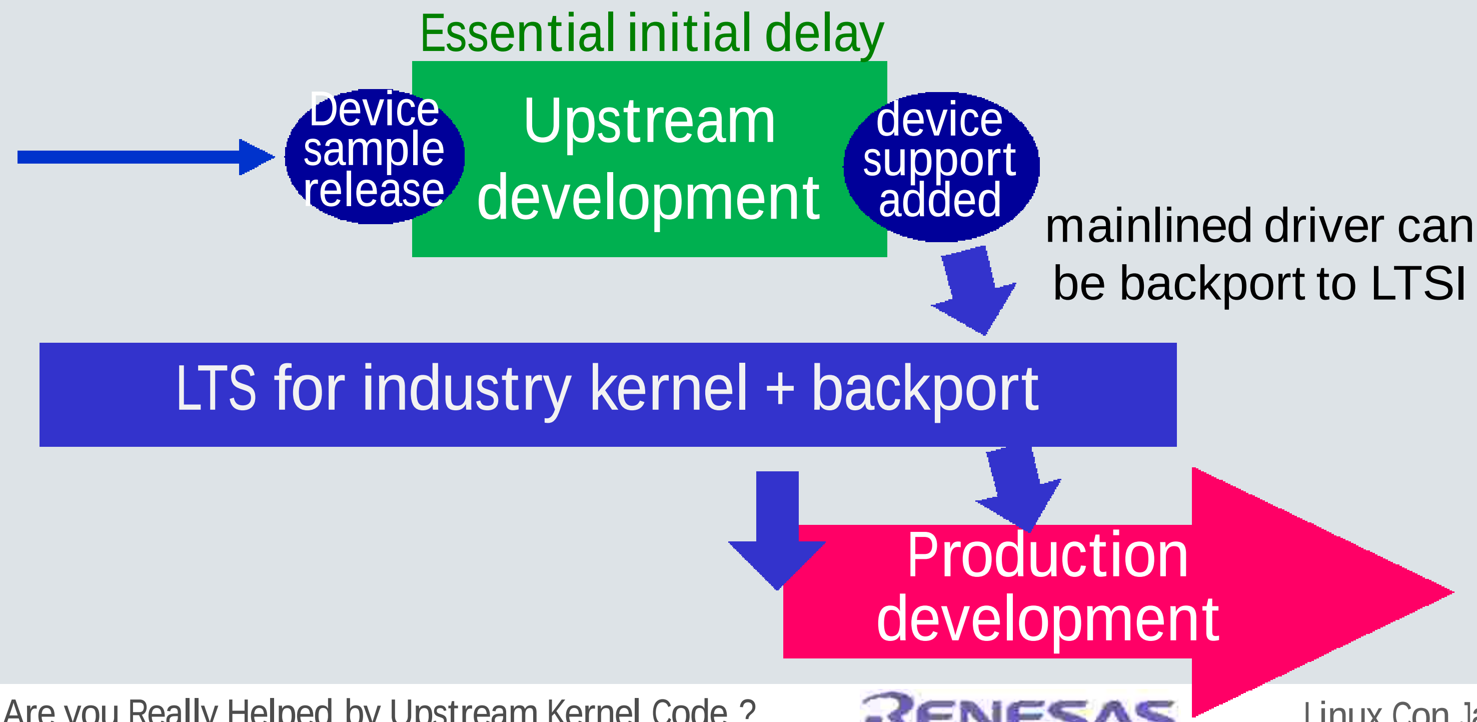
SoC vendor hope add support on current kernel

- Upstream takes some time after chip releases (3-6 month)
  - Set producer need to start development with current kernel that does not include new SoC device support



SoC vendor hope add support on current kernel

- Upstream takes some time after chip releases (3-6 month)
  - Set producer need to start development with current kernel that does not include new SoC device support



scope of "LTS for industry" (candidates)

- device driver
  - common device drivers for embedded*1
    1: Target driver list needs to be maintained with TLF/CEWG members
  - SoC specific code will not be cared for by TLF/CEWG, rather SoC vendor can send their mainlined code to this LTS kernel to add their own enhancement.
- Platform support
- Part of common kernel function*2

  2: Not all fundamental change can be back ported due to its complexity.
  Framework change might not be applicable as it require huge code change
  - memory / power management
  - realtime enhancement
  - boot related (boot method, boot time .,etc)
  - size related (linux-tiny etc)

patch categorization

- B: Bug fix / Correctness Fix

extended backport target

- C: Clean-up
  Infrastructure Change
  Documentation Change
- F: New Feature (not performance related)
- P: Performance Enhancement
- U: Usability Enhancement
- X: Unclassified

I think that clean-up and infrastructure change patches should be back/up-ported on a best-effort basis as they are not essential.

example patch categorization

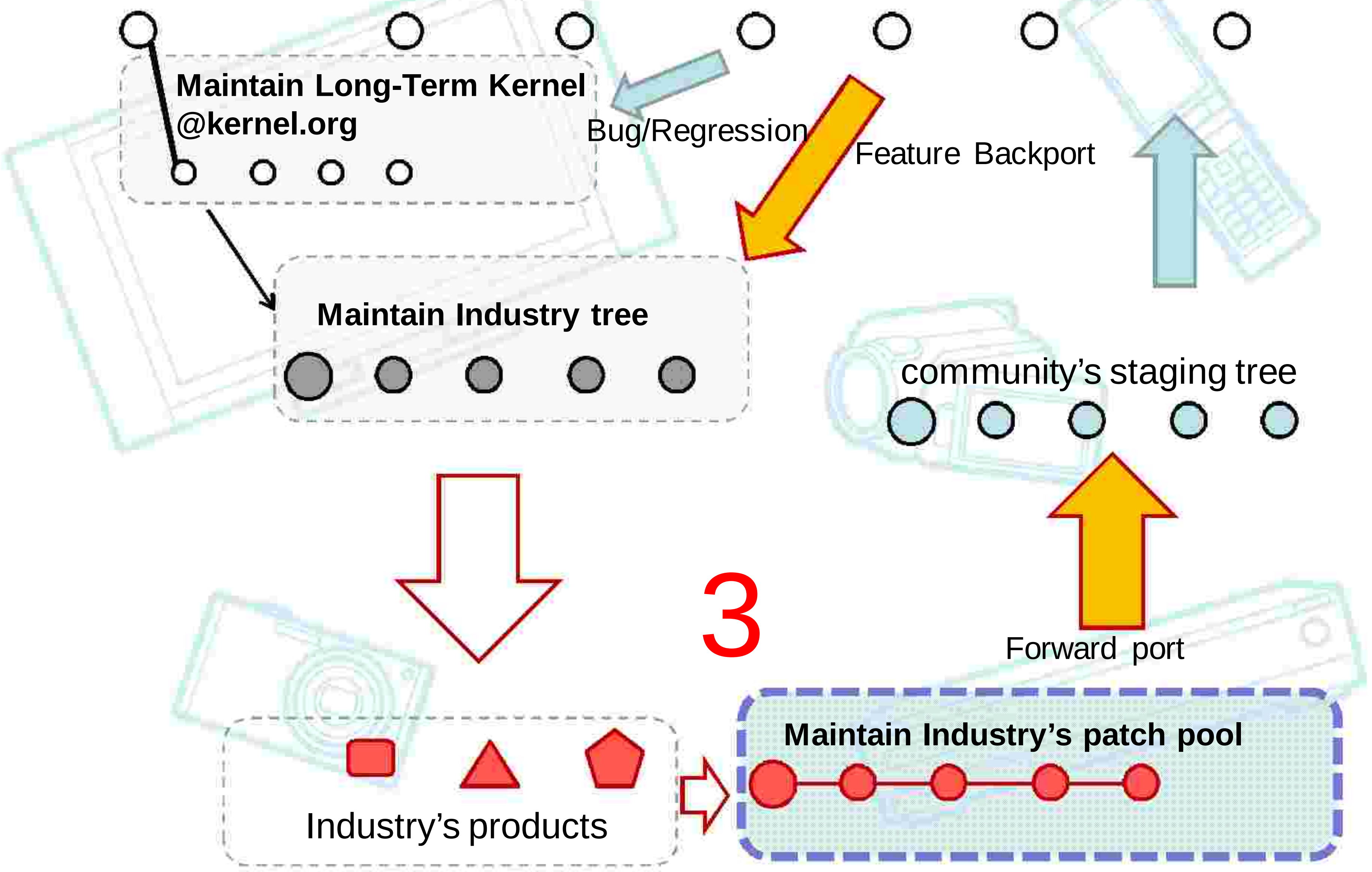
F	7fbda226 ARM: shmobile: sh73a0: Provide the I2C platform data
C	1720ee5c ARM: shmobile: sh73a0: Delete I2C4 related resources
F	c008228d ARM: shmobile: ag5evm: Provide ag5evm reset function
F	e1e72cac ARM: shmobile: Introduce shmobile_arch_reset() hook
B	3284c77c mtd: nand: sh_flctl: Cann't use full SDA area
B	21091e39 ARM: shmobile: ag5evm: Provide FLCTL and mLBA-NAND device resources
F	7400a9fb ARM: shmobile: intc-sh73a0: IRQ pins interrupt support
B	9c107648 ARM: shmobile: intc-sh73a0: Use __raw_xxx() version of I/O accessors
C	bc1ae8eb ARM: shmobile: intc-sh73a0: Rearrange the order of the PINT IRQs
C	e776aa33 ARM: shmobile: intc-sh73a0: No need to disable interrupts on pint_lock'ing
B	a6f3a8d8 ARM: shmobile: intc-sh73a0: Do not read-modify-write the PINTRR[01]A registers
C	5645e619 ARM: shmobile: intc-sh73a0: No spinlock on the PINTRR[01]A register accesses
C	67710877 ARM: shmobile: intc-sh73a0: Remove unnecessary IRQ flags maintenance
B	d1c68c2b ARM: shmobile: intc-sh73a0: Initialize the PINT mask and control registers
F	13a5a863 ag5evm: Add platform support for sdhi1
F	66a716f6 ARM: shmobile: ag5evm: Tiny LED class driver for LCD brightness control
F	565afdb ARM: mach-shmobile: ag5evm: Set MMC_CAP_SD_HIGHSPEED flag
F	13f2745 ARM: mach-shmobile: clock-sh73a0: Provide zb1, zb3, zb3s clk definitions
F	2f98edf ARM: mach-shmobile: clock-sh73a0: add con_name "sgx"
B	394362e ARM: mach-shmobile: clock-sh73a0: Fix divisors[12] for M1/M2/M3 clocks
B	facff60 ARM: mach-shmobile: clock-sh73a0: pll rate recalculation fix
B	44f62b3 ARM: mach-shmobile: clock-sh73a0: Fix mstp_clks[MSTP112] (for SGX543)

LTS for industry comes with test

- To secure backport code quality, we want to conduct system test for CEWG kernel, and we will focus on device driver behavior
  - New driver should come with test scenario.
  - New driver should come with functional coverage.
  - Use automated test system (like autotest)
  - Regression test case need to be collected and applied
  - CEWG members can share this test framework

CEWG need to organize this test management case.

3rd solution = Upstreaming

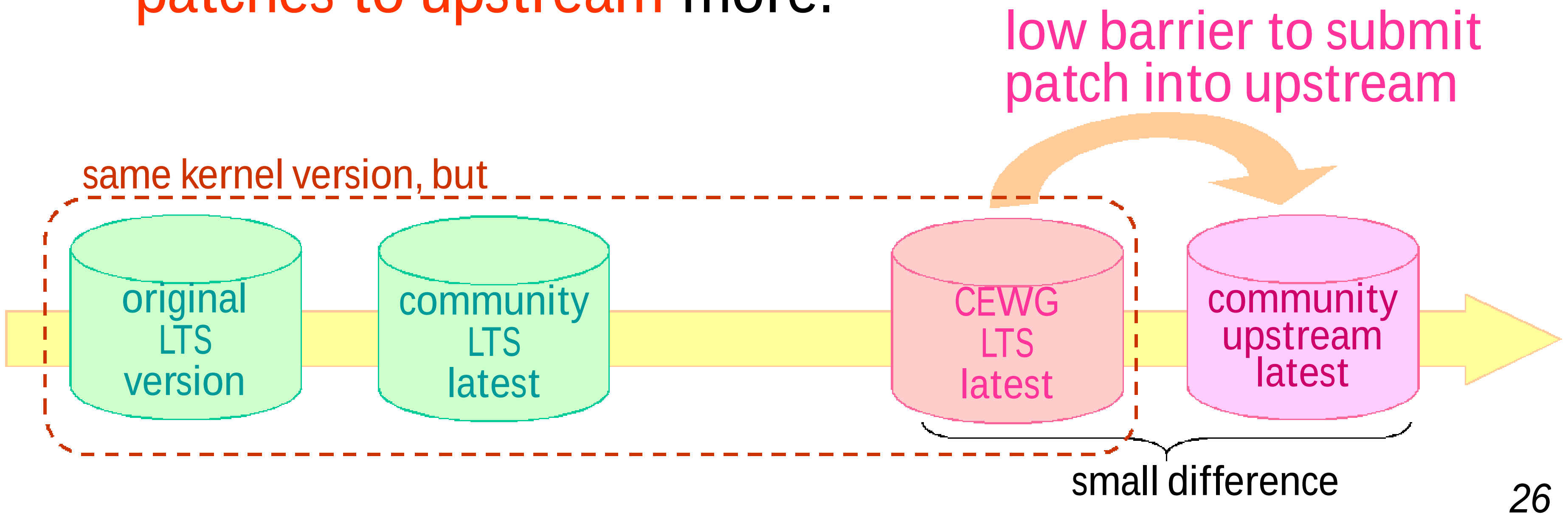


3rd solution = Upstreaming

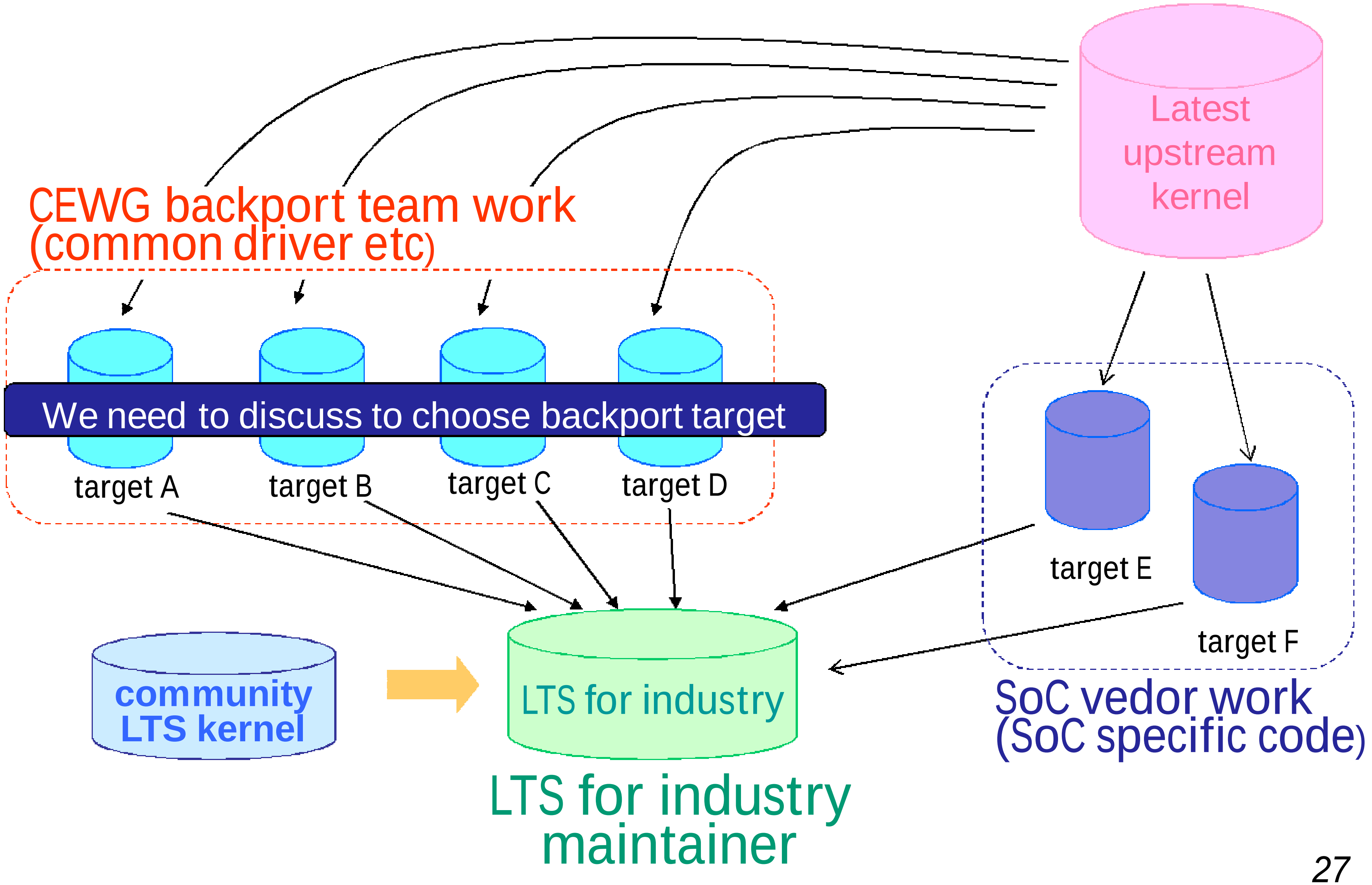
- CEWG will help industry developer to submit their work result to upstream. It might include **consolidate and forward port** action.
- We may need to consolidate each company's work those aims same fix and/or enhancement.
- If we merge some patchs written by other auther (company) we should keep all original author's signed-off-by name not to cause license violation issue.

LTS for industry

- TLF/CEWG LTS code is very close to latest upstream kernel for the target areas compared to community LTS, because it already includes various updates.
- Our intention to maintain TLF/CEWG LTS is to encourage the embedded industry to send their patches to upstream more.



LTS for industry backport integration

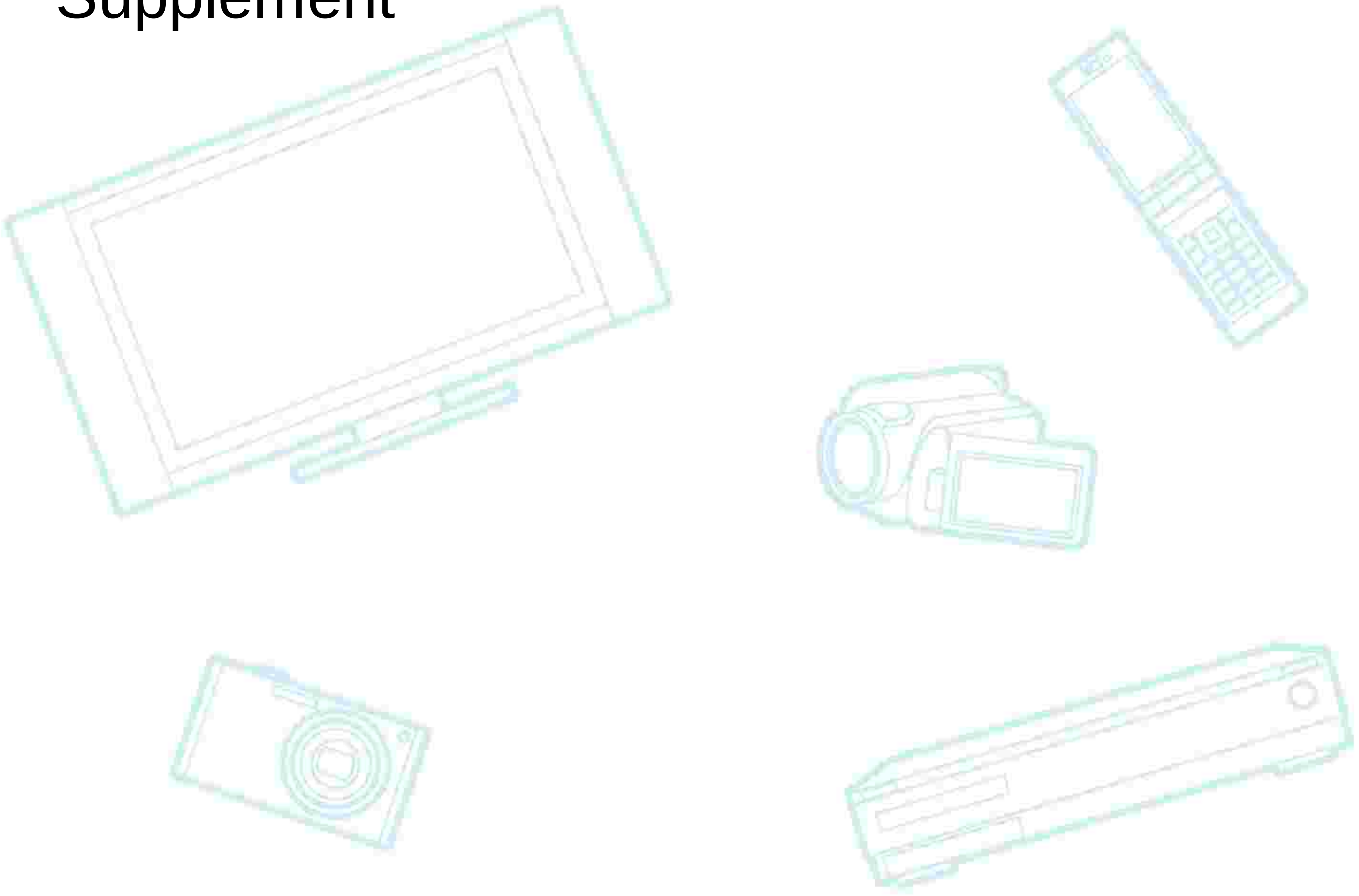

Latest upstream kernel
CEWG backport team work (common driver etc)
We need to discuss to choose backport target
target A
target B
target C
target D
target E
target F
SoC vedor work (SoC specific code)
community LTS kernel
LTS for industry
LTS for industry maintainer

Pilot Project - “Yami-nabe”

- What is “Yami-nabe” ?
  - Japanese old time’s fan party to enjoy thrill and happenings.
  - The party will be held in the dark room.
    - “Yami” = Dark room, “nabe” = Cauldron
  - Participants bring some foods without disclosing anything and put them in the cauldron. No one knows what foods are in the cauldron because of the dark room and then boil it.
  - Participants pick foods from the cauldron by using chop sticks.. He or she should eat it whether he/she likes it or not …

- Why “Yami-nabe” pilot project ?
  - Manufactures do not want to provide their kernel source code to avoid unnecessary comparisons with competitors in a source code level.
  - CEWG will provide “Yami-nabe” environment to put kernel source code without disclosing a company’s name.
  - CEWG will cook/analyze them to find out common (and good) changes which are not part of the Upstream. We need to recognize how many/large that is and make our plan to put such changes into the Upstream.
  - CEWG will also realize how much the code are fragmented, and the reasons.

Supplement



Background

A few reasons why Consumer Electronics Industry is not so active in the upstream

- Fragmentation & Version Gap
  - The use of Linux kernel is very fragmented. Some come from semiconductor vendors, some are based on an internal version of Linux Kernel which was “forked” long time before, and some come with Android. There are version gaps, and it is hard to share knowledge across projects.
  - Fragmented use of Linux and time-pressure make embedded system engineers difficult to work “Upstream First”, but force to work on locally.
  - The forked private tree is far from the upstream version due to the version gaps and numerous private fixes. It is difficult to even test with the latest upstream version which forces them to work locally.
- Challenges in differentiations
  - Traditionally, the industry tends to keep modifications and enhancements in secret within a company, since those are considered as “know-how” and differentiations which sometimes violate the GPL.

⇩

Fragmented use of Linux and local/private fixes to all are becoming major pain points

- The industry now requires Long Term Support of Linux
  - as a common base for the entire ecosystem including semi-conductor venders, set-vendors, software components providers, distributors, and hopefully Google Android and MeeGo to solve the fragmentation issue.
  - to accelerate activities toward the upstream with having a common base version of Linux

LTS for the Industry – What it is ?

The industry has slightly different requirements on LTS.

- By definition, fatal bug fixes and security fixes would be back-ported to the stable version of Linux.
- On the other hand, the industry needs back-porting of features for emerging devices support, performance improvement, better power management capability and so on.
  - Features to be back-ported may vary by industry (e.g., consumer electronics, aerospace and defense, automotive, control system, and others.)

LTS for the Industry (LTSI) ;

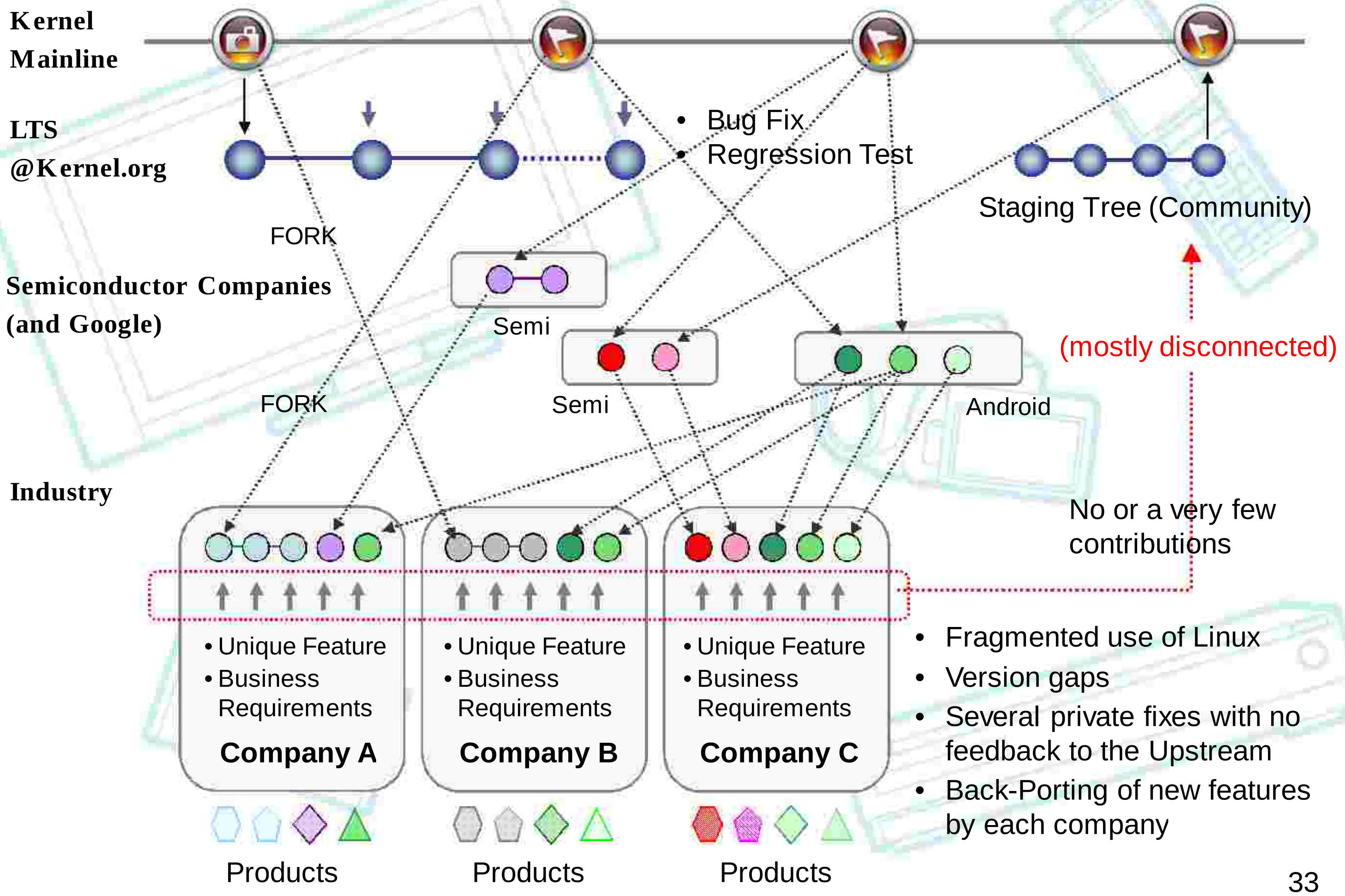
- is an industry-wide collaborative activity to minimize fragmentation and accelerate further innovation of Linux with incorporating fixes and innovations from embedded system engineers.
- is expected to be the base for majority of embedded systems
- is expected to be the base for ecosystem players (e.g., semiconductor vendors, set-vendors, software component providers, distributors, and system/application framework providers such as Google Android and MeeGo)
- is maintained by the LTS Back-Port (BP) team and ecosystem partners.
  - Base tree (MM, PM, RT, Security,..) is maintained by LTS BP Team
  - Device/Architecture trees are maintained by ecosystem partners (e.g., Processor/SoC vendors)

LTS for the Industry – What it is NOT ?

LTS for the Industry (LTSI)

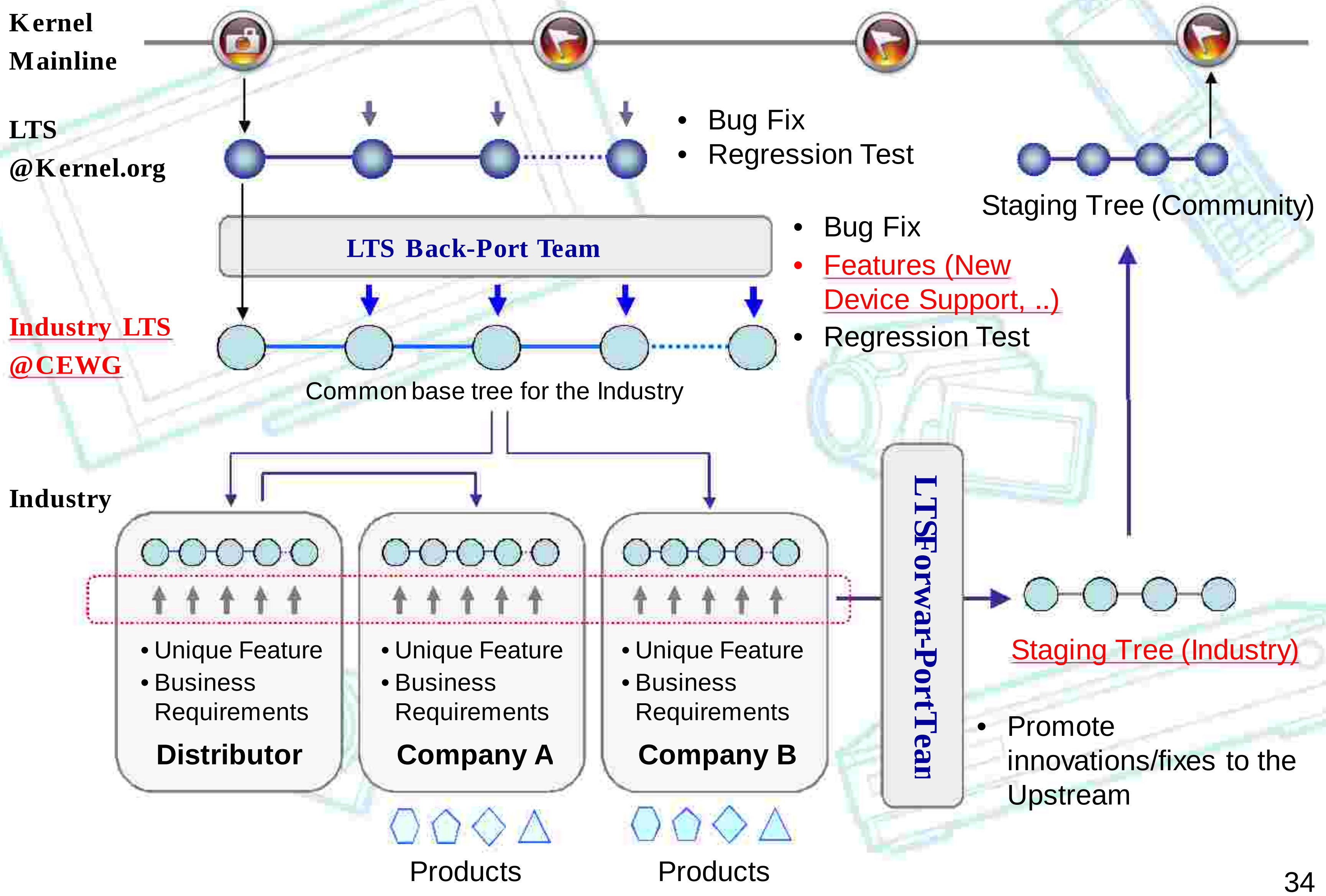
- is NOT intended to de-promote the use of the upstream.
  - ð With reducing fragmented use of Linux, CE WG expects embedded system engineers to find more time to work with the upstream, rather than working on several versions of Linux locally.
- is NOT intended to discourage embedded programmers to contribute to the upstream.
  - ð CE WG conducts a back-porting work to LTSI from the upstream kernel. Bug fixes and some key important functional enhancement that embedded programmers have contributed to the upstream, are ported into LTSI as well as new releases. This significantly reduces local works by embedded system engineers, and motivates them.
  - ð With reducing a gap between LTSI and the upstream by feature porting, CE WG expects embedded system engineer to easily test with a latest version of the upstream kernel, and work with upstream engineers to find better solutions.
- does NOT mean the embedded industry wants to stay in an old level of Linux.
  - ð Similar to Enterprise, the embedded industry may want to use a single version of Linux for a life of a particular product. But, the evolution of devices are fast enough, and innovative resource management functions are expected to be in the upstream. The industry is ready to pick up the latest version, but, expects it to be stable while a product is in a market.

Consumer Electronics Industry – Fragmented use of Linux

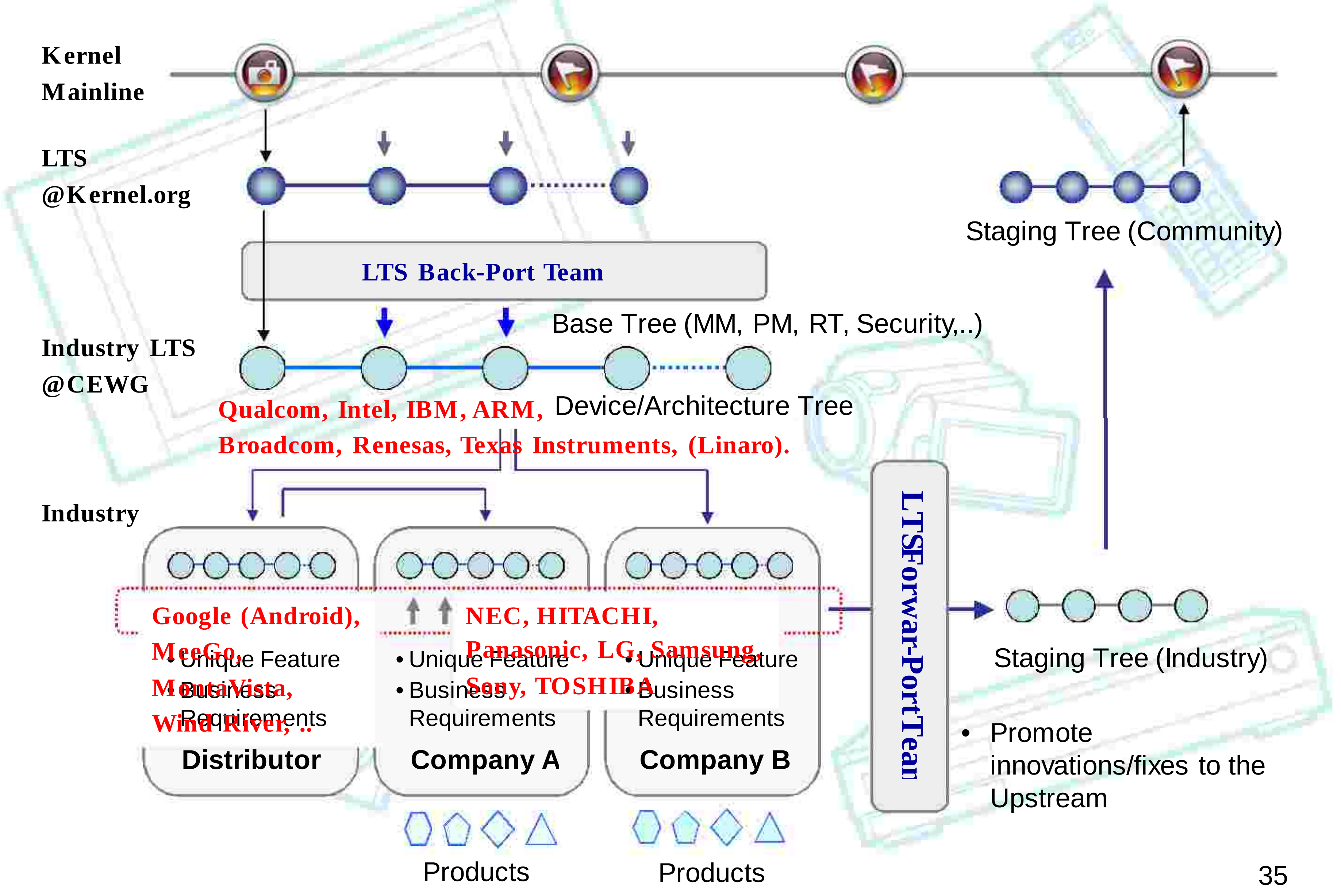


- Fragmented use of Linux
- Version gaps
- Several private fixes with no feedback to the Upstream
- Back-Porting of new features by each company

LTS for Industry is divided by 3 parts



Industry version of LTS Kernel - Expected Partners



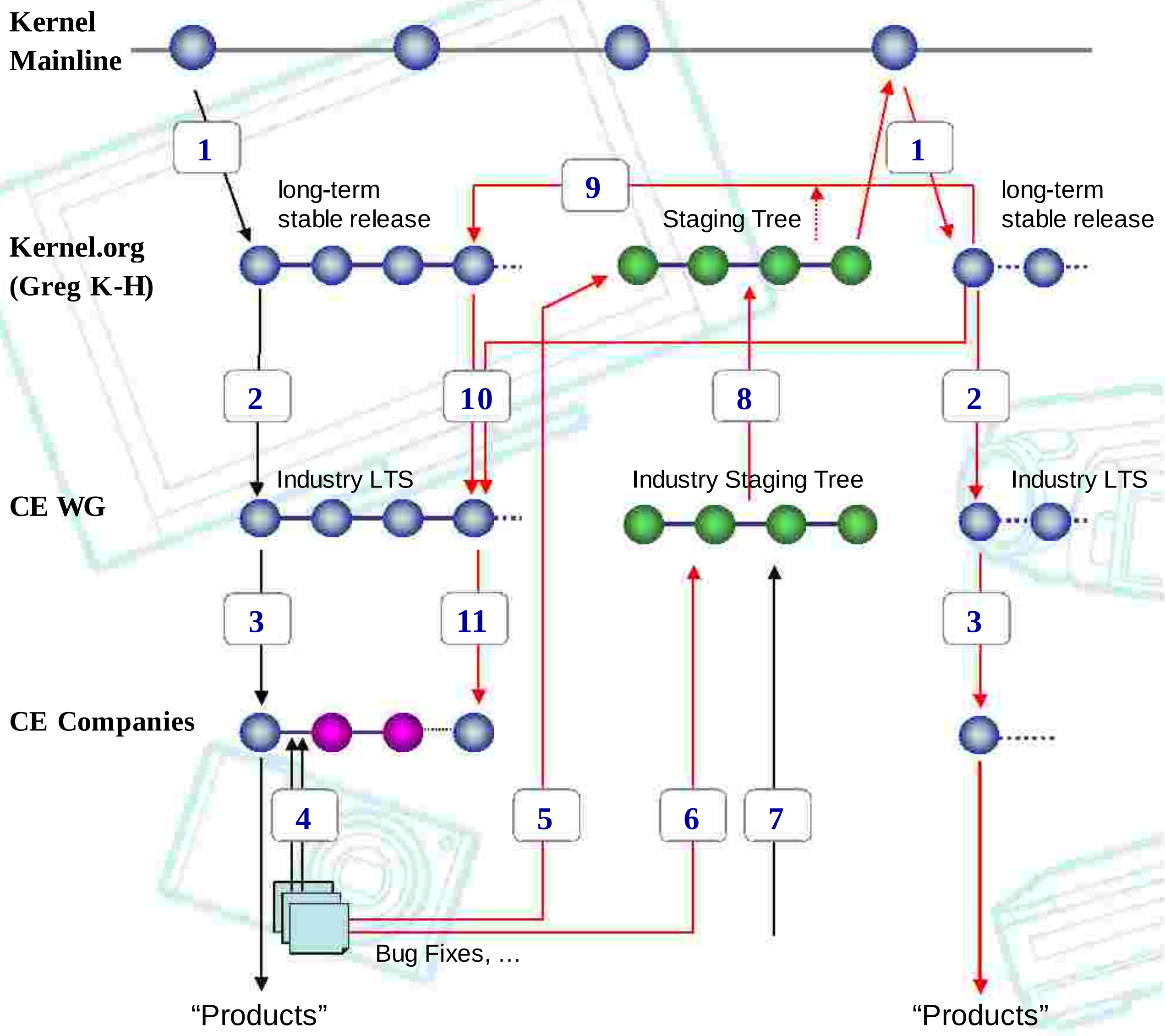
Key tasks for “LTS for the Industry” project

- Creating and maintaining a Long Term Support Industry tree working closely with Greg K-H.
  - This is the key first step not only for “LTS for the Industry”, but also for other entities who may want to make a particular version of Linux as LTS.
  - Details and processes should be discussed with the Linux Community.

<“LTS for the Industry” unique activities start from here>

- Back-Porting
  - maintains the LTS Industry base tree (MM, PM, RT, Security, ..)
    - Architecture/Device trees are expected to be maintained by ecosystem partners.
  - conducts back-porting of
    - additional bug fixes and security fixes which are relevant to embedded systems
    - new features relevant to embedded systems (e.g., new device support, performance improvement, power management enhancement, ..)
  - runs the quality assurance process
    - develops and maintains test suites.
- Forward-Porting
  - encourages Linux engineers in the Industry to work with the upstream.
  - works with representatives from the Industry, solicits fixes and enhancements, and coordinates its re-implementation with upstream maintainers if required.
  - maintains the Industry staging tree to be linked with the stating tree for the upstream.
  - analyzes causes of local fixes for further encouragement of “mainlining”.

Industry LTS maintenance cycle



Note : Red lines shows "inclusion of bug fixes contributed by embedded system programmers"

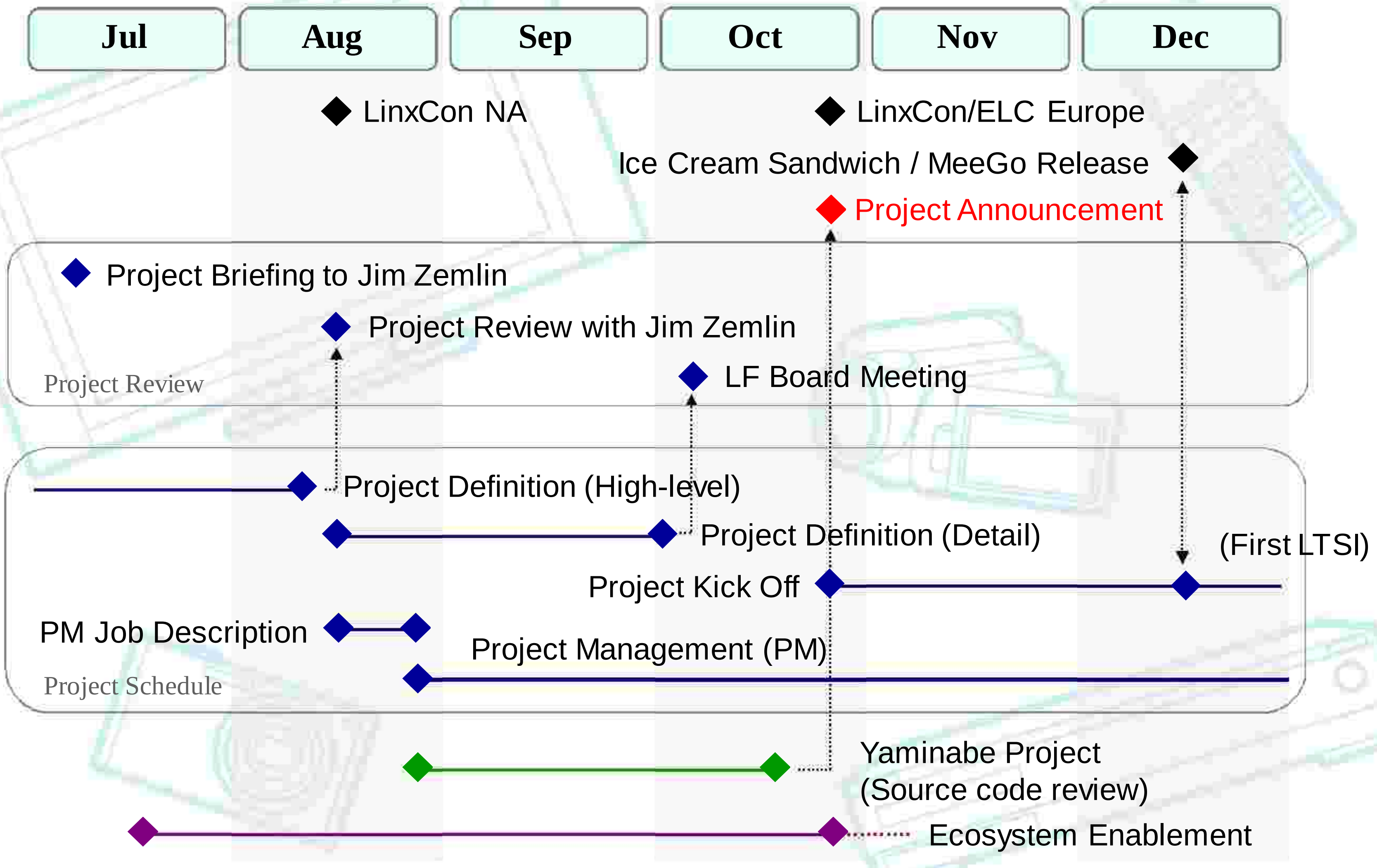
1. A certain version of Linux becomes LTS.
2. CE WG creates LTS tree for Industry.
3. Distributors, Silicon vendors and others deliver LTS based package to CE companies
4. Embedded system programmer fix problems.
5. Mature programmers may directly work with the community to include patches into the upstream.
6. Programmers work with CE WG to prepare sending patches through the industry staging tree.
7. CE WG picks up some important features to be mainlined from several sources, e.g., Android, etc.
8. CE WG makes sure the industry staging tree is merged into the community's one.
9. LTS maintainer back-ports bug fixes and security updates.
10. CE WG Industry tree maintainer back-ports bug fixes, security patches, and embedded related additions.
11. Distributors, Silicon vendors and others releases updates.

LTS for the Industry - Value Propositions

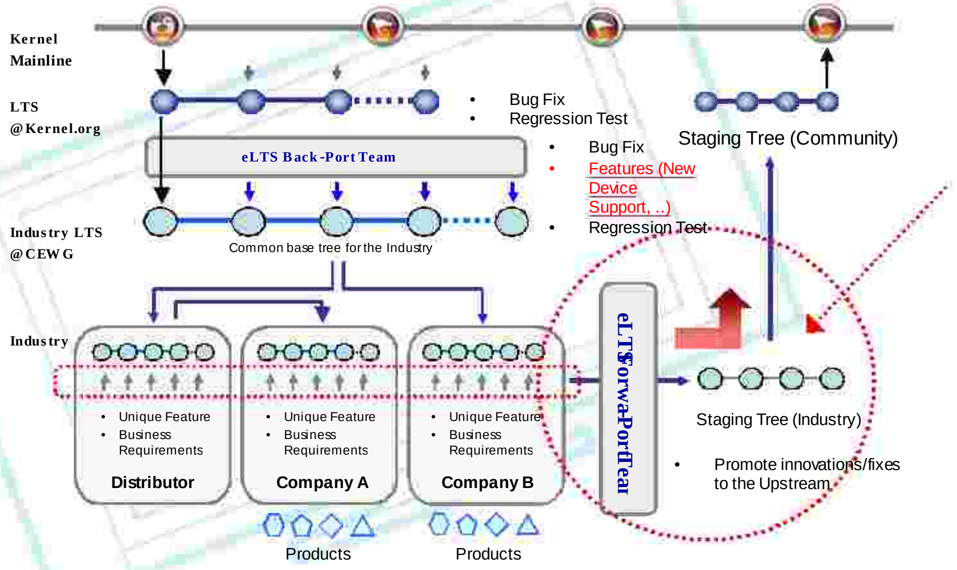
- For the Consumer Electronics Industry
  - Efficient and less-expensive operations in enabling Linux to a variety of devices with less fragmented use of Linux
    - Easy to apply latest bug fixes
    - Easy to determine problem
      - As differences between LTS and latest version of the mainline are expected to be small, embedded engineers can work with community engineers to test and fix on the latest version easily.
  - No need to apply some of local fixes anymore
    - Some of local fixes were incorporated into the mainline with the help of the LTS Forward Port Team, and back-ported into LTS by the LTS Back Port Team.
- For the Semiconductor Industry
  - Enhancements and bug fixes in the Upstream will be propagated to LTS, then, applied to distributions and actual embedded systems. This is a very effective way to enhance and maintain their code.
  - No need to work privately on an individual tree inside customers (e.g., set-vendors, automotive manufacturers, ..)
- For Distributors
  - As the LTS Back Port Team maintains the industry tree up-to-date, distributors can focus on their differentiations.
- For the Linux Community
  - Better communications with embedded engineers with a common code base.
  - More contributions are expected from embedded industry.

LTS for the Industry - Timelines

Updated on August 8th, 2011



Pilot Project - “Yami-nabe”



- To fully understand the status of fragmented use of Linux, reasons why it is diverged, and quality of local fixes, then, assess the workload of the “LTS Forward-Port Team”, a small pilot project is proposed.

Objectives:

- In order to improve development practices for embedded system, compare and contract several existing Linux kernel trees with a view to understanding the major themes in how they have diverged from “mainline” and from each other.

Targets:

- Version : Android GingerBread (kernel 2.6.35 + Android patch set)
- Trees : NEC, Sony-Ericsson, LG, Samsung (to be confirmed)

Duration and expense:

- 2-3 months, less than $50K

Pilot Project - “Yami-nabe”

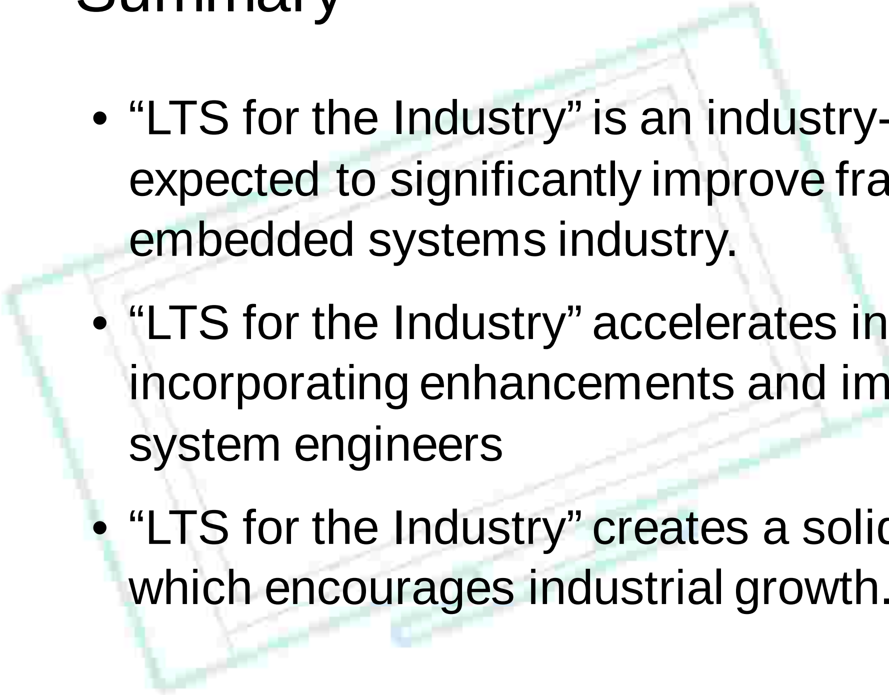
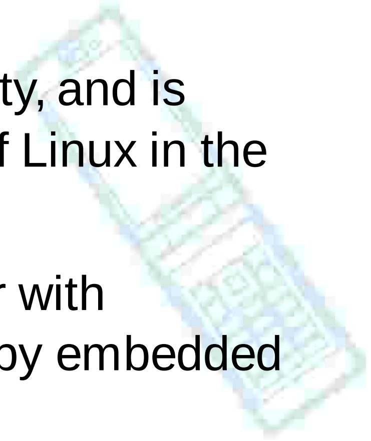
- What is “Yami-nabe” ?
  - Japanese old time’s fan party to enjoy thrill and happenings.
  - The party will be held in the dark room.
    - “Yami” = Dark room, “nabe” = Cauldron
  - Participants bring some foods without disclosing anything and put them in the cauldron. No one knows what foods are in the cauldron because of the dark room and then boil it.
  - Participants pick foods from the cauldron by using chop sticks.. He or she should eat it whether he/she likes it or not …

- Why “Yami-nabe” pilot project ?
  - Manufactures do not want to provide their kernel source code to avoid unnecessary comparisons with competitors in a source code level.
  - CEWG will provide “Yami-nabe” environment to put kernel source code without disclosing a company’s name.
  - CEWG will cook/analyze them to find out common (and good) changes which are not part of the Upstream. We need to recognize how many/large that is and make our plan to put such changes into the Upstream.
  - CEWG will also realize how much the code are fragmented, and the reasons.

Resource Estimation and Request

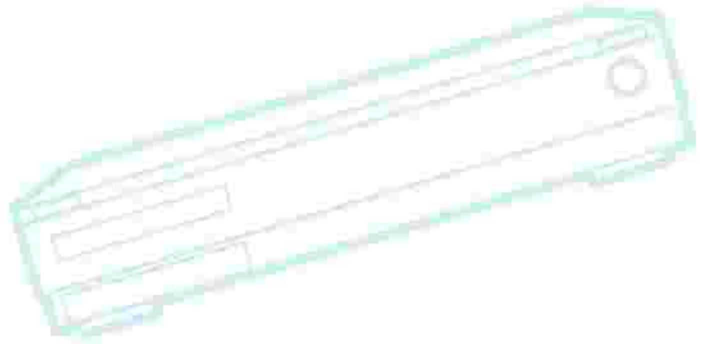
- “LTS for the Industry” is an industry-wide activity which requires coordination with several stakeholders. A project manager (PM) needs to be assigned.
  - With considering the project schedule below, a project manager should be hired for 2 months starting from middle of August, at minimum. Then, the contract should be extended upon successful launch of the project at LinuxCon/ELC-E on October.
    - Project Concept Review with Jim Zemlin (Aug., 2011)
    - Project Announcement and Kick Off @ LinuxCon/ELC-E (Oct., 2011)
- To fully understand the status of fragmented use of Linux, reasons why it is diverged, and quality of local fixes, then, assess the workload of “LTS Forward-Port Team”, a small pilot project, “Yami-nabe” is proposed.
  - The project is the key for the success of “forward porting” of fixes and enhancement provided by the Industry, and should be completed by middle of October.
  - Expected cost for this project should be less than $50K (2 months x 3 persons)
- Resource estimation for the LTS Back Port Team and the LTS Forward Port Team will be completed by end of September.

Summary

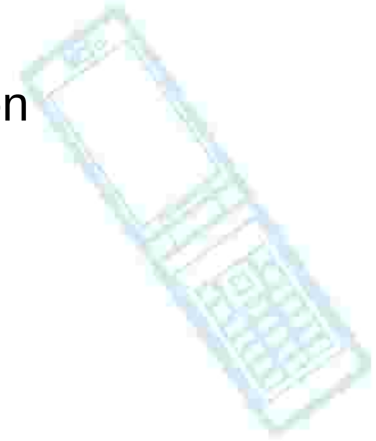



- “LTS for the Industry” is an industry-wide collaborative activity, and is expected to significantly improve fragmentation of the use of Linux in the embedded systems industry.
- “LTS for the Industry” accelerates innovation of Linux further with incorporating enhancements and improvements conducted by embedded system engineers
- “LTS for the Industry” creates a solid ecosystem in the embedded arena which encourages industrial growth.

- The Linux Foundation should promote this project for further innovation and deployment of Linux.



Contributions



Hisao Munakata	Renesas Electronics Corporation
Nobuhiro Asai	IBM Corporation
Satoru Ueda	Sony Corporation
Tim Bird	Sony corporation
Tsugikazu Shibata	NEC Corporation