

TOSHIBA

Leading Innovation >>>

Using RT Preempt patch with LTSI kernel

Yoshitake Kobayashi
Advanced Software Technology Group
Corporate Software Engineering Center
TOSHIBA CORPORATION

29 Apr - 1 May 2014

Who am I?

- **Yoshitake Kobayashi (YOSHI)**
 - Chief Specialist at
Corporate Software Engineering Center,
TOSHIBA CORPORATION

- Work on embedded operating systems
 - Linux
 - RTOS
 - TOPPERS (uITRON), VxWorks
 - Open source software license

Focus of talk

- **How to use RT patch with LTSI kernel**
 - Source code is available at the following URL:
<https://github.com/ystk/linux-ltsi>

- **Expected experience level: Beginner**

Overview

- **Recipe**

- **Four steps to make LTSI-RT**

- Step 1: Basic steps to use LTSI kernel patch
- Step 2: Merge RT patch with LTSI kernel
- Step 3: Resolve conflicts
- Step 4: Test

- **Conclusion**

Recipe

■ Ingredients

- Stable kernel
 - <http://git.kernel.org/?p=linux/kernel/git/stable/linux-stable.git>
- LTSI kernel
 - <http://ltsi.linuxfoundation.org/>
- RT Preempt patch
 - <http://git.kernel.org/?p=linux/kernel/git/rt/linux-stable-rt.git>
 - <https://www.kernel.org/pub/linux/kernel/projects/rt/>

■ Commands

- Git
- grep
- Text editors
- Merge tools

References for Real-time patch

- **A realtime preemption overview**

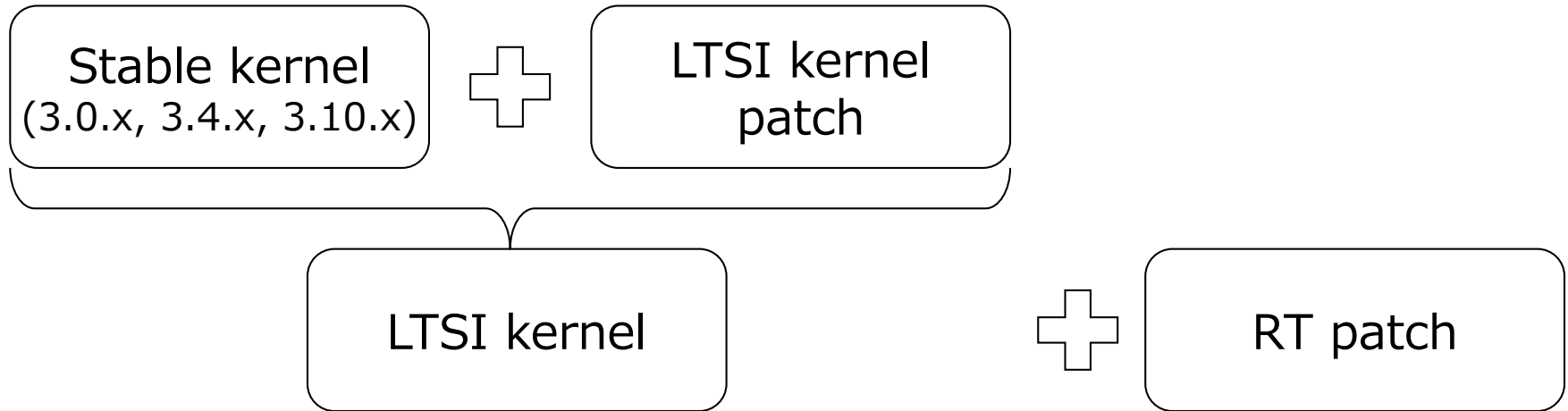
- <http://lwn.net/Articles/146861/>

- **Presentation materials**

- Frank Rowand
 - Real-Time Failure
 - http://elinux.org/images/b/be/Real_time_linux_failure.pdf
 - Status of Linux 3.x Real Time and Changes From 2.6
 - http://elinux.org/images/5/54/Status_of_real_time.pdf
- Steven Rostedt
 - Inside The RT Patch
 - http://elinux.org/images/b/ba/Elc2013_Rostedt.pdf

Scenarios to create LTSI-RT

■ Scenario 1

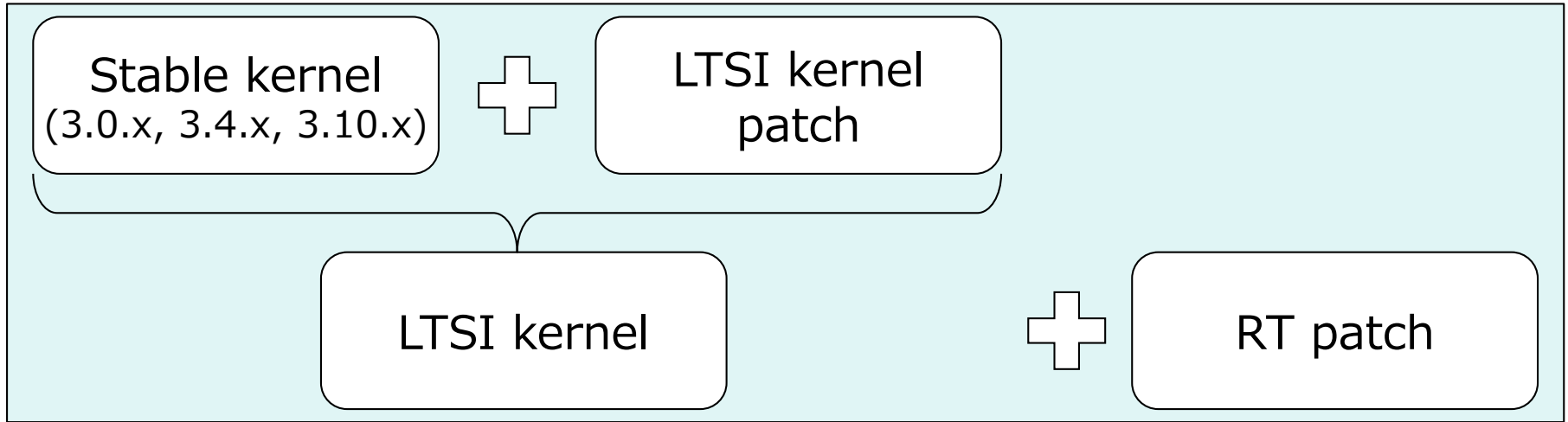


■ Scenario 2



Scenario

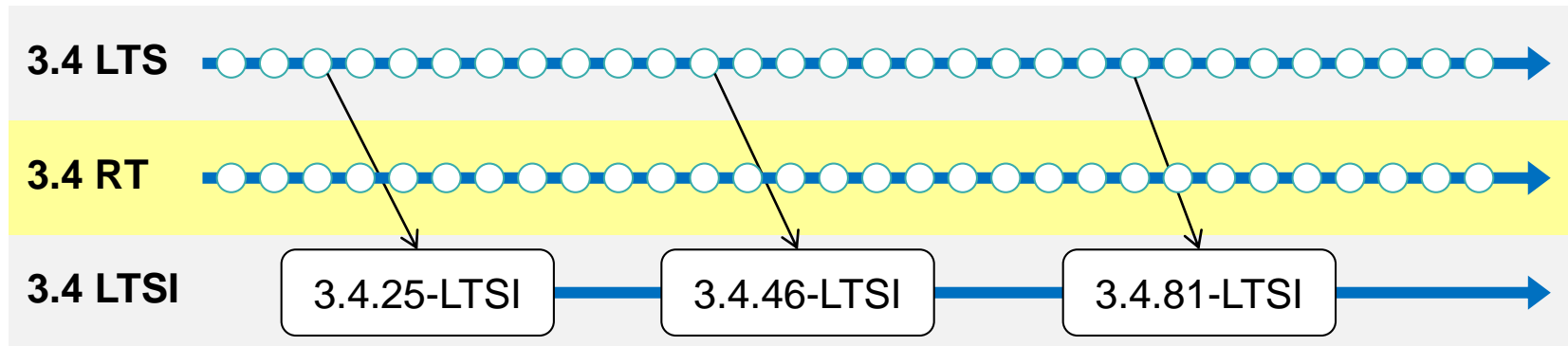
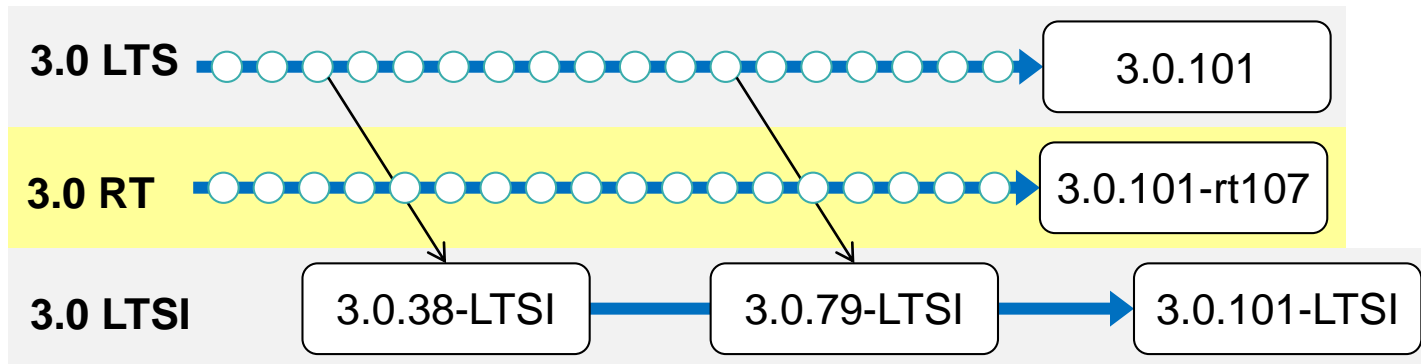
■ Scenario 1



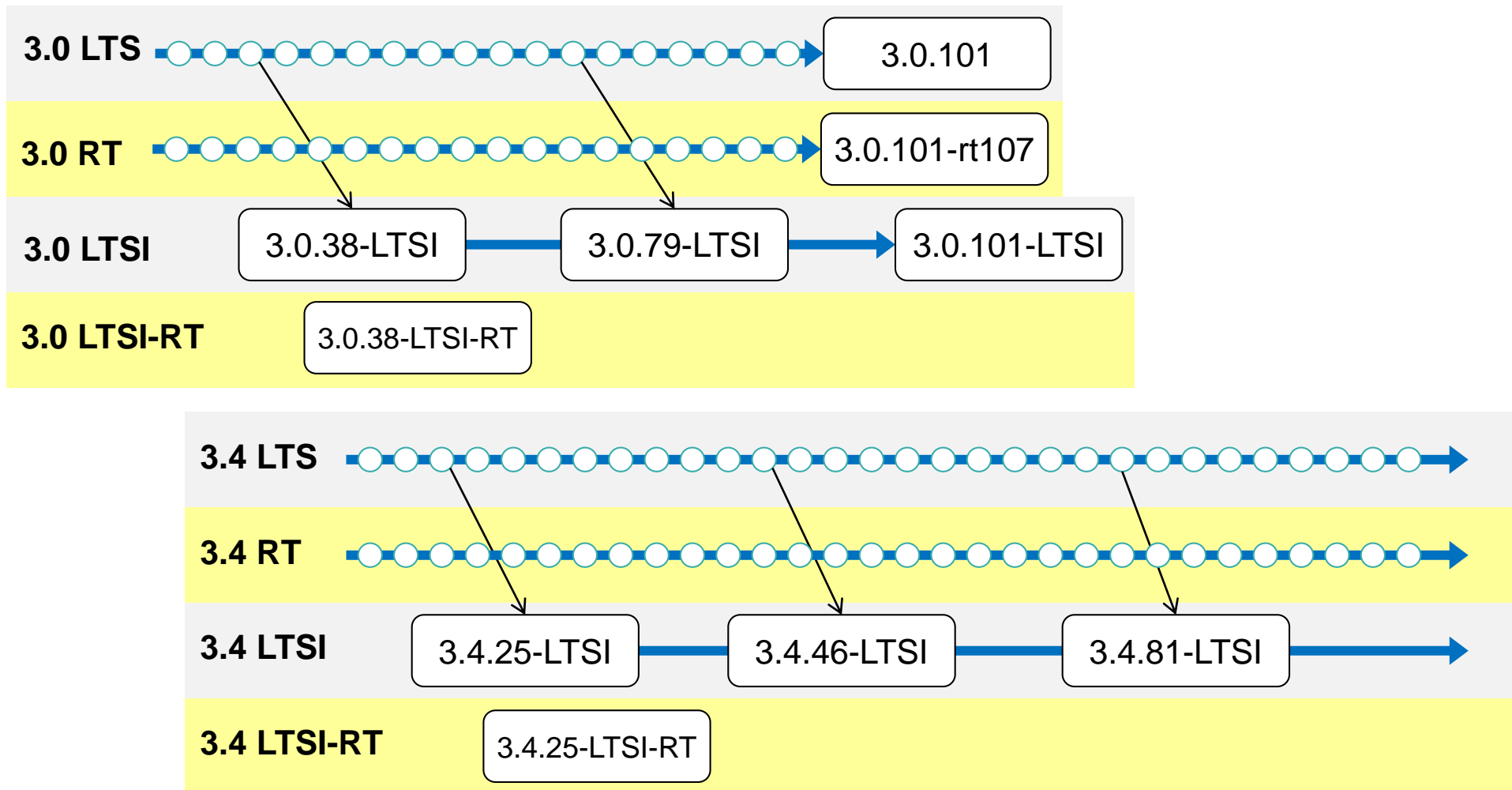
■ Scenario 2



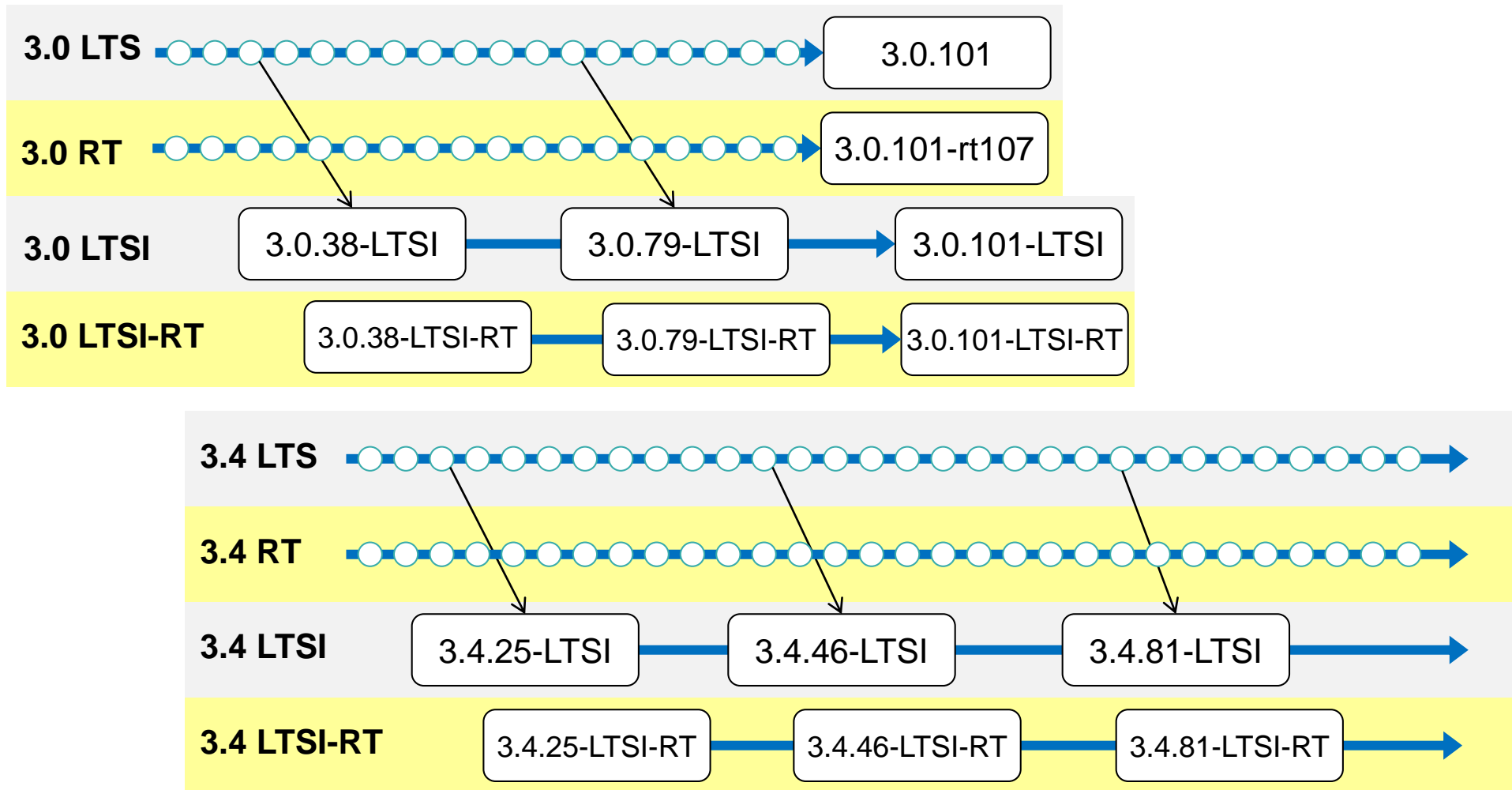
LTSI development cadence



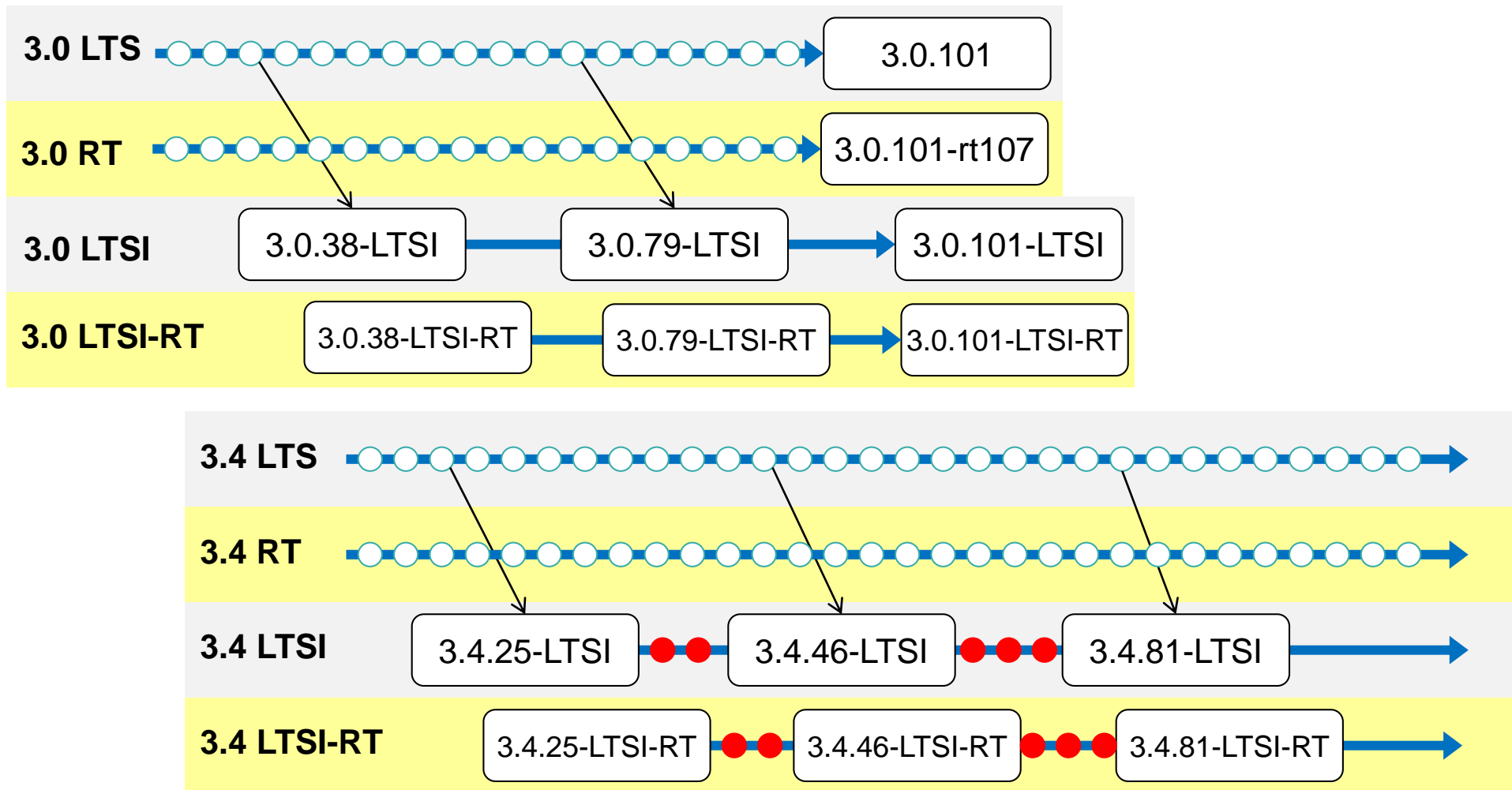
LTSI development cadence



LTSI development cadence

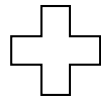


LTSI development cadence



Step 1: Basic steps to use LTSI patch

Stable kernel
(3.0.x, 3.4.x, 3.10.x)



LTSI kernel
patch

■ An example to prepare LTSI kernel

1. Prepare a stable kernel source tree

```
$ git clone git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git  
$ cd linux-stable/  
$ git checkout v3.4.46 -b v3.4.46-ltsi-tmp
```

2. Prepare a LTSI patch tree

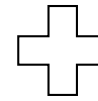
```
$ git clone http://git.linuxfoundation.org/ltsi-kernel.git  
$ cd ltsi-kernel/  
$ git checkout -b v3.4.46-ltsi-tmp v3.4.46-ltsi
```

3. Apply LTSI patch to stable kernel

```
$ export QUILT_PATCHES=../ltsi-kernel  
$ git quiltimport  
$ git tag v3.4.46-ltsi
```

Step 2: Basic steps to use RT patch

LTSI kernel
(v3.4.46-ltsi)



RT patch

■ Merge RT patch with LTSI kernel

1. Add stable-rt for reference

```
$ git remote add stable-rt git://git.kernel.org/pub/scm/linux/kernel/git/rt/linux-stable-rt.git  
$ git remote update
```

2. Merge RT tree and LTSI kernel tree

```
$ git merge v3.4.46-rt61
```

..... (CONFLICTS)

Step 3: Resolve conflicts

■ **Modification policy**

- Bug fixes need to be merged
- API changes might be resolved
- When a part of LTSI patch modifies core kernel function
 - Try to fix
 - Simply ignore a patch

Conflicts to make v3.4.46-ltsi-rt

```
$ git merge v3.4.46-rt61
CONFLICT (content): Merge conflict in drivers/net/ethernet/cadence/at91_ether.c
CONFLICT (content): Merge conflict in mm/page_alloc.c
```

- **Which patch was made changes on conflicted code?**
 - RT?
 - LTSI?

```
$ lv drivers/net/ethernet/cadence/at91_ether.c
$ grep -r drivers/net/ethernet/cadence/at91_ether.c ../ltsi-kernel
$ grep drivers/net/ethernet/cadence/at91_ether.c patch-3.4.46-rt61.patch

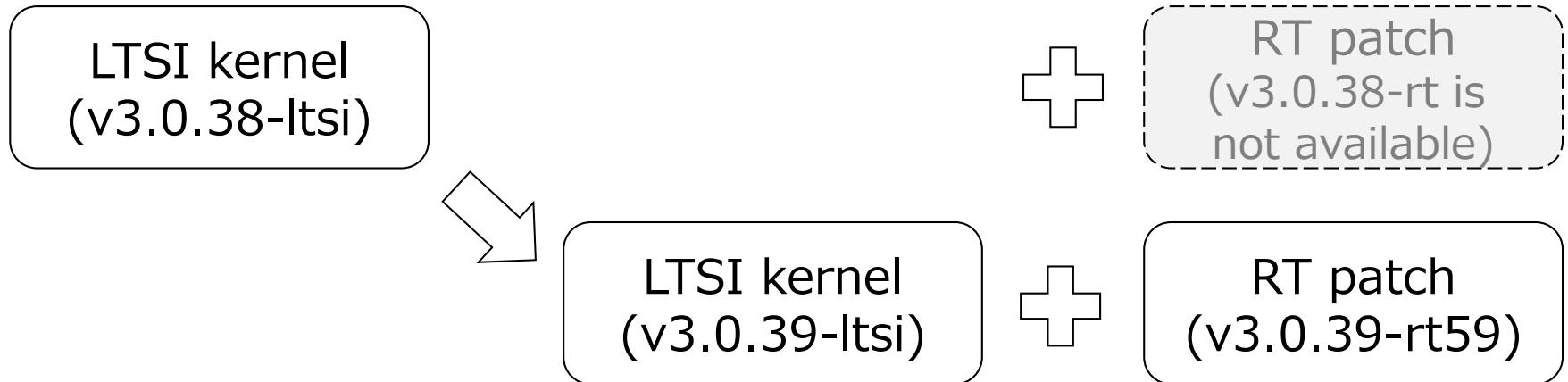
$ lv mm/page_alloc.c
$ grep -r mm/page_alloc.c ../ltsi-kernel
$ grep mm/page_alloc.c patch-3.4.46-rt61.patch
```


Make v3.0.y-ltsi-rt

1. Prepare the v3.0.38 kernel source tree and LTSI tree

```
$ cd linux-stable/  
$ git checkout v3.0.38 -b v3.0.38-ltsi-tmp  
$ cd ltsi-kernel/  
$ git checkout -b v3.0.38-ltsi-tmp v3.0.38-ltsi  
$ cd ../linux-stable/  
$ git quiltimport
```

2. Find a relative RT tree
3. Marge v3.0.39's changes with v3.0.38-ltsi
4. Merge RT path with v3.0.39-ltsi



Conflicts for v3.0.39-ltsi-rt development

```
$ git merge v3.0.39-rt59
```

```
Renaming drivers/tty/serial/8250.c => drivers/tty/serial/8250/8250.c
```

```
CONFLICT (rename/modify): Merge conflict in drivers/tty/serial/8250/8250.c
```

```
CONFLICT (content): Merge conflict in arch/arm/common/gic.c
```

```
CONFLICT (content): Merge conflict in arch/arm/common/gic.c
```

```
CONFLICT (content): Merge conflict in arch/x86/kernel/process_32.c
```

```
CONFLICT (content): Merge conflict in include/linux/irq.h
```

```
CONFLICT (content): Merge conflict in include/linux/plist.h
```

```
CONFLICT (content): Merge conflict in include/linux/rtmutex.h
```

```
CONFLICT (content): Merge conflict in kernel/Makefile
```

```
CONFLICT (content): Merge conflict in kernel/irq/settings.h
```

```
CONFLICT (content): Merge conflict in kernel/rtmutex.c
```

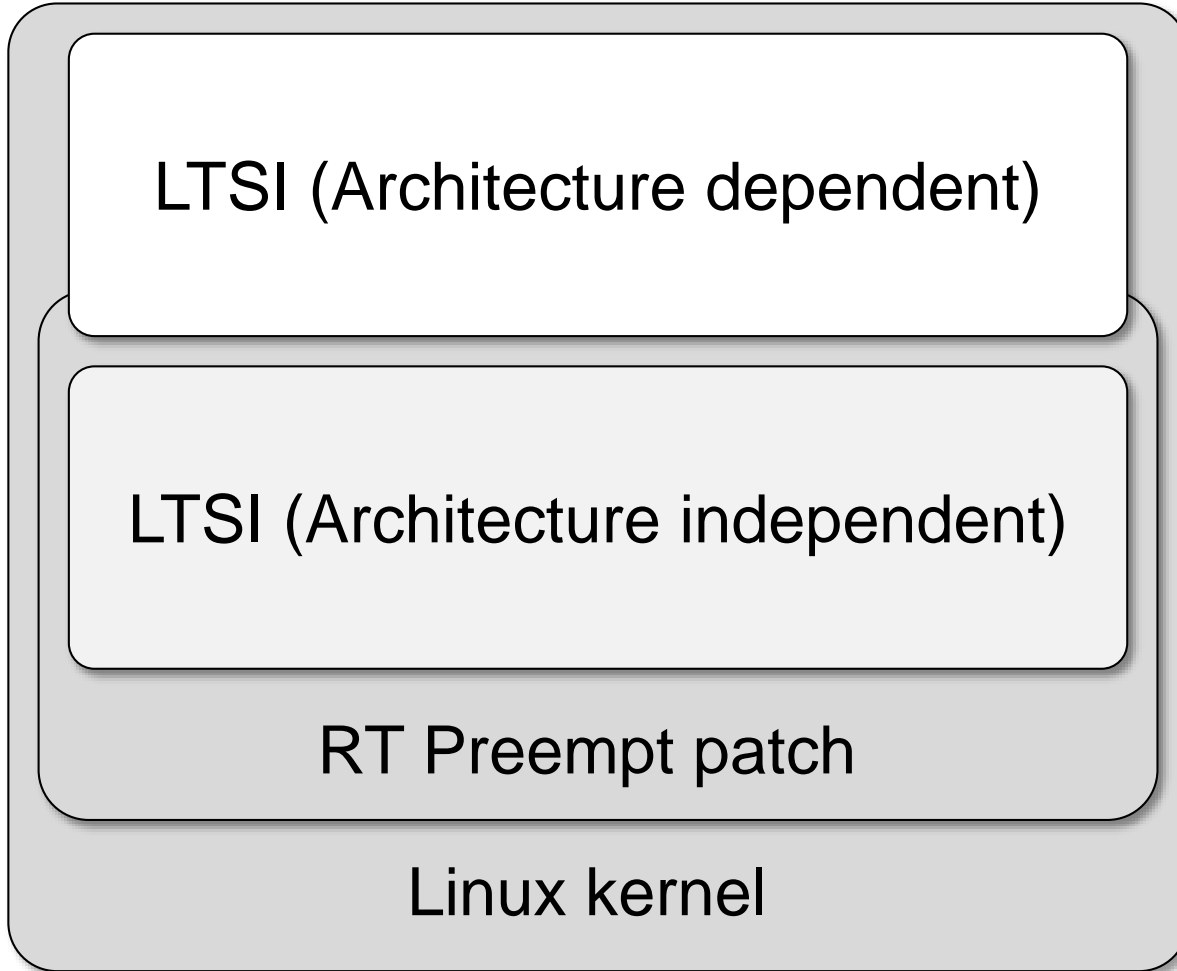
```
CONFLICT (content): Merge conflict in mm/page_alloc.c
```

■ Current solution

- Simply ignore patches which are related to PLIST

Step 3: Still missing an important thing

- This modification covers the following grey areas



Step 4: Test

- **Compilation test**

- allconfig
- allmodconfig
- Customized configuration

- **Kernel configuration file preparation**

- Configuration
 - ON: CONFIG_PREEMPT_RT_FULL , High resolution timer
 - OFF: Power management, Debug
- Tutorials
 - https://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO

Step 4: Test

■ LTP

- Compare results between original RT kernel and LTSI-RT

■ Performance test

- Latency
 - Cyclicttest
- Network
 - Netperf
- I/O
 - dd

■ Stress test

- CPU stress
- Data reliability
- Power ON/OFF

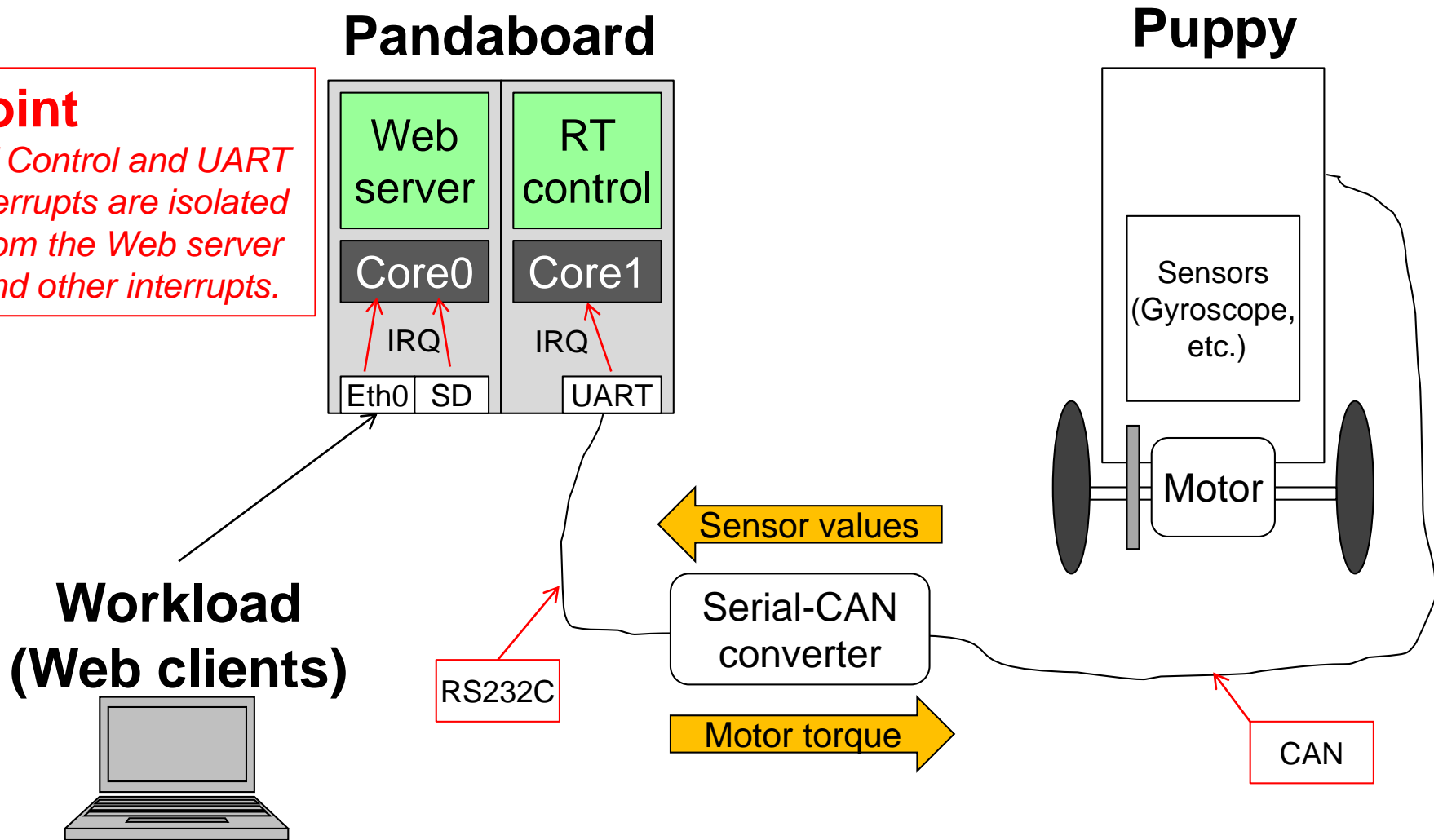
■ Customized test

- Hardware resource isolation

An example for isolation demo

Point

RT Control and UART interrupts are isolated from the Web server and other interrupts.



Reference (original idea): <https://www.toppers.jp/safeg.html>

When a system has some latency issue..

- **Find latency bottlenecks**

- Profilers
- Tracers

- **Fix it**

Conclusion

- **This presentation shows how to create LTSI-RT**
- **Source code is available at the following URL:**
 - <https://github.com/ystk/linux-ltsi>
- **LTSI-3.10-RT will be available soon**

Questions?

The latest slide is available at the following URL:
http://elinux.org/ELC_2014_Presentations