



Kernel-Mode NVRAM

Jun 2011



Ralink Website document

for joedian@sunnex.com.tw
And company Use Only

Agenda

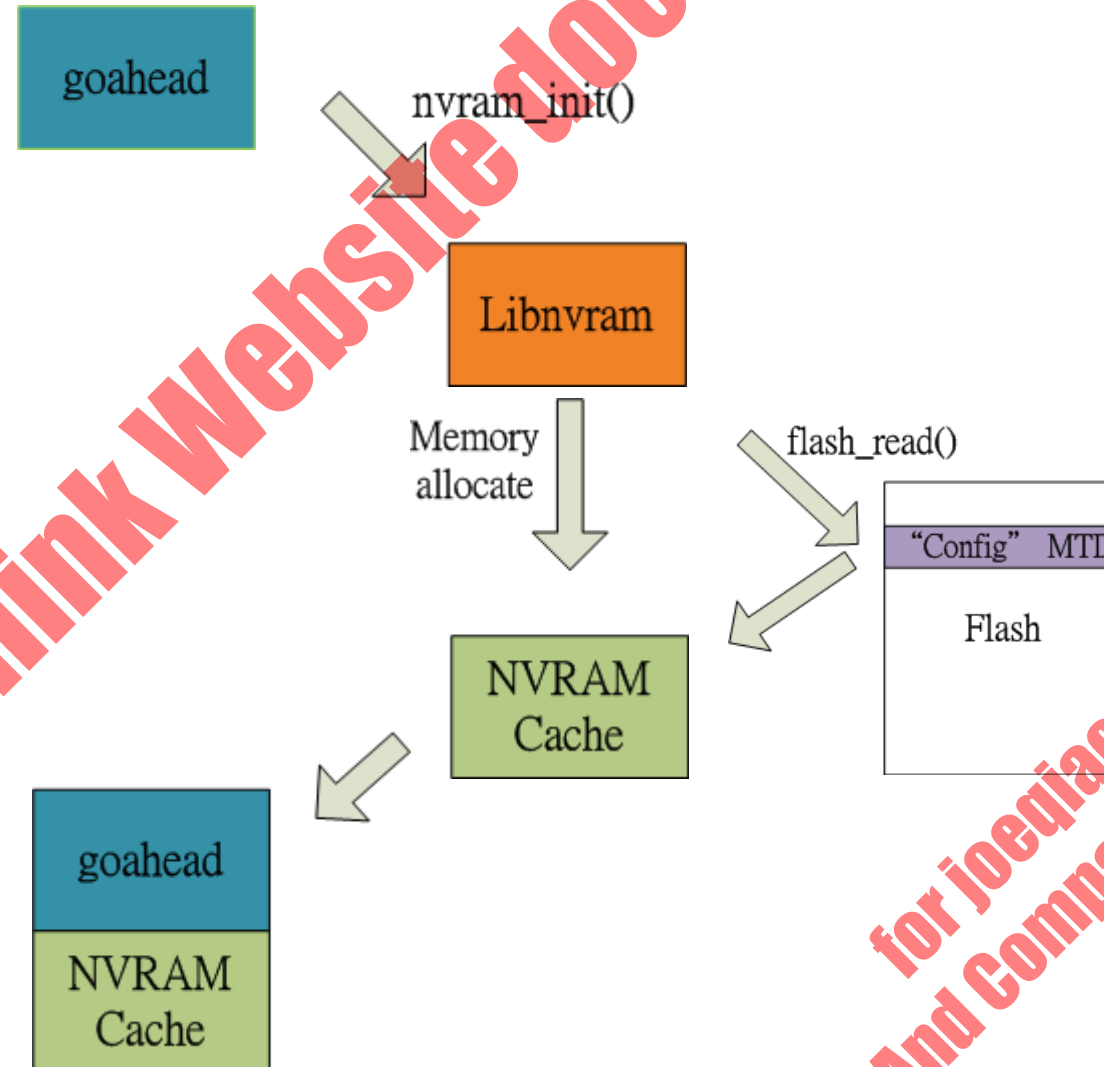
- Original Design
- Issues
- New Design
- Code Review
- Test Scenario

Original Design

- User-space library
- Provide APIs for NVRAM operations
- Allocate a user-space memory region to store NVRAM data from Flash, and each user-space program has individual one
- Data is stored in ASCII format
- API:
nvram_init, nvram_get, nvram_bufget, nvram_set,
nvram_bufset, nvram_commit, nvram_close,
nvram_clear

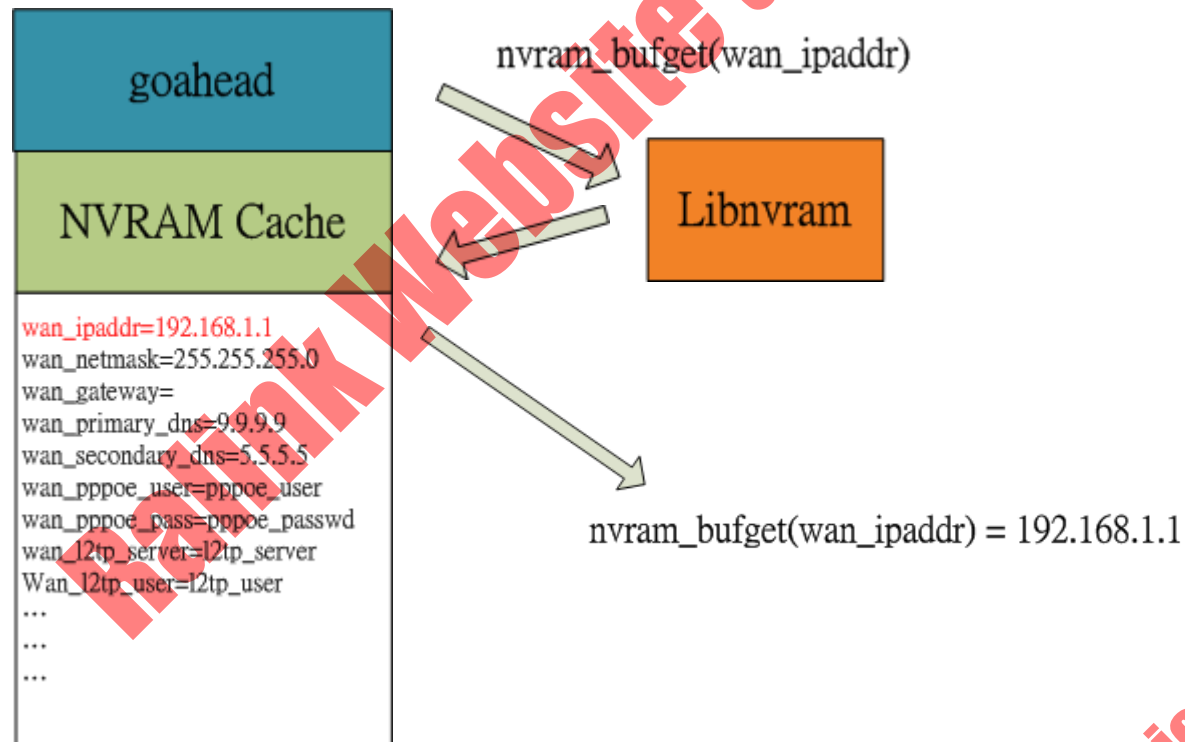
for Joeqiao@synnex.com.tw
And Company Use Only

Original Design: nvram_init()



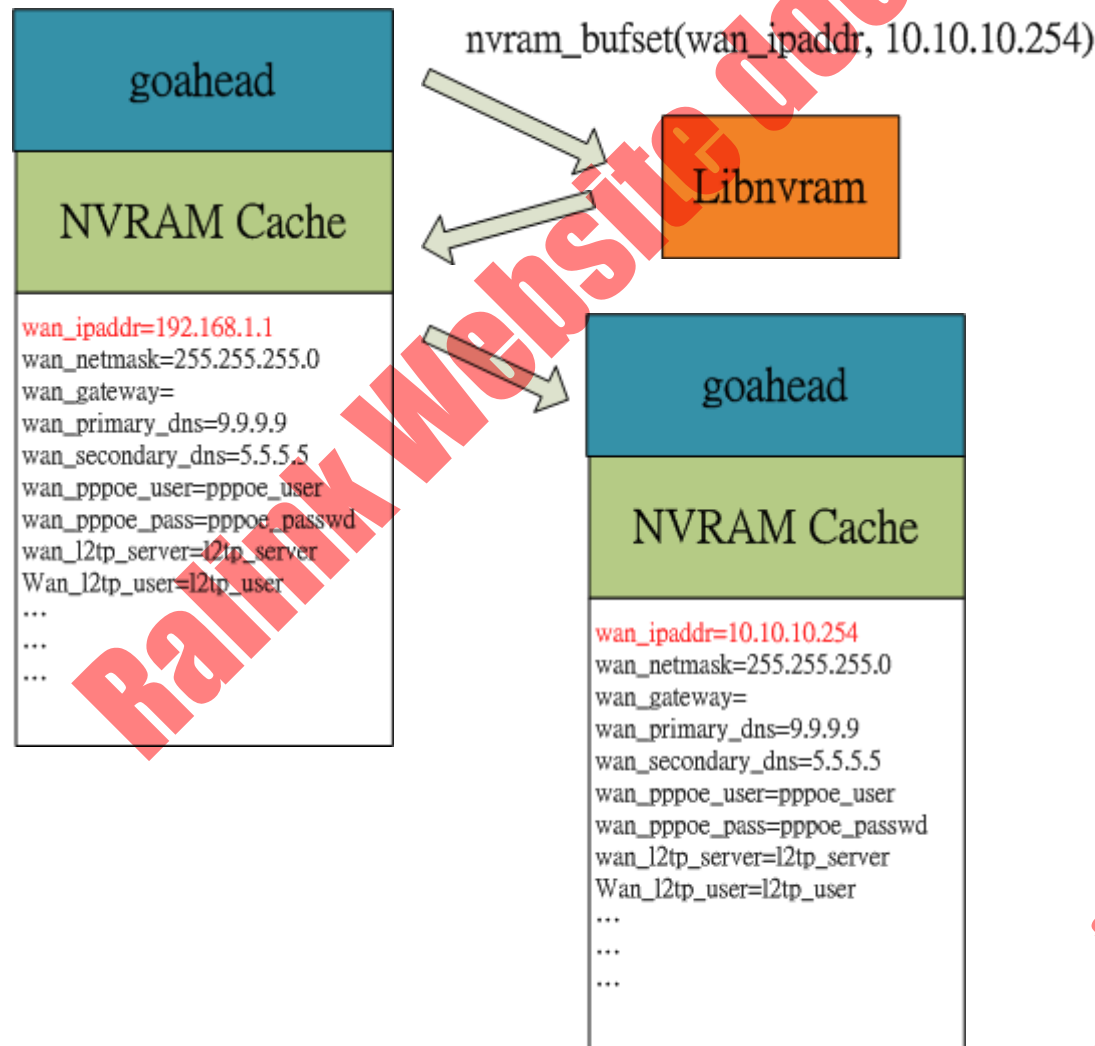
for joeqiao@synnex.com.tw
And Company Use Only

Original Design: nvram_bufget()



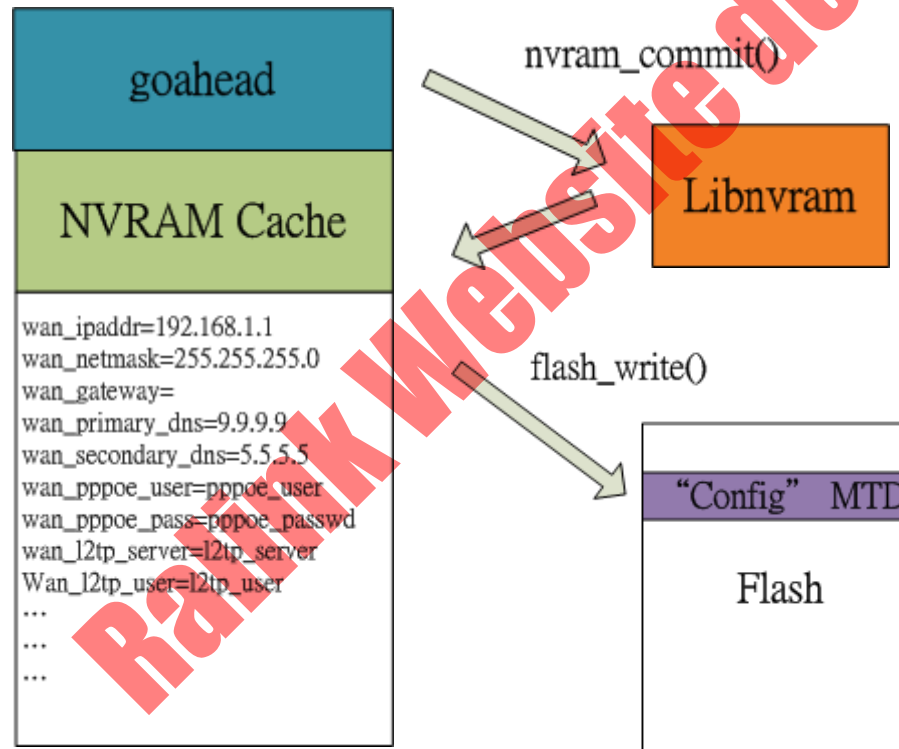
for joeqiao@synnex.com.tw
And Company Use Only

Original Design: nvram_bufset()



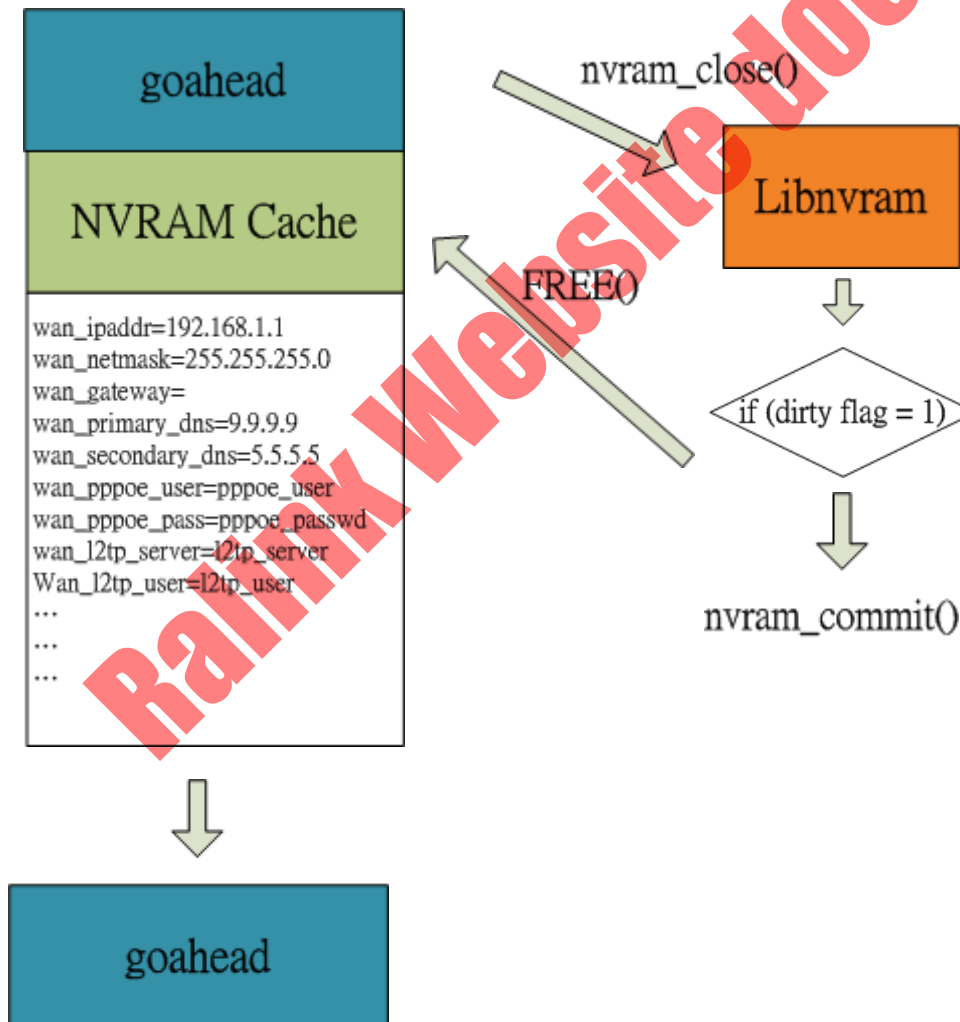
for Joeqiao@synnex.com.tw
And Company Use Only

Original Design: nvram_commit()



for joeqiao@synnex.com.tw
And Company Use Only

Original Design: nvram_close()



for joeqiao@synnex.com.tw
And Company Use Only

Original Design: nvram_get()

```
char *nvram_get(int index, char *name)
{
    //LIBNV_PRINT("--> nvram_get\n");
    nvram_close(index);
    nvram_init(index);
    return nvram_bufget(index, name);
}
```

- Sync data with Flash and return the freshest value

for joeqiao@synnex.com.tw
And Company Use Only

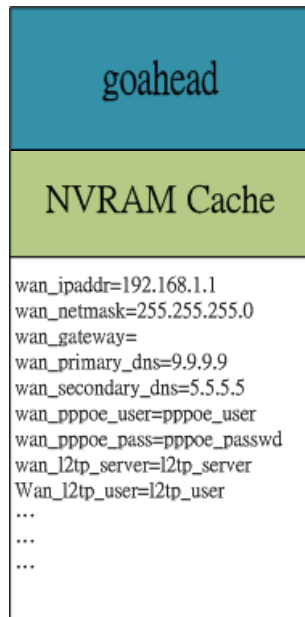
Original Design: nvram_set()

```
int nvram_set(int index, char *name, char *value)
{
    //LIBNV_PRINT("--> nvram_set\n");
    if (-1 == nvram_bufset(index, name, value))
        return -1;
    return nvram_commit(index);
}
```

- Set the value and commit current data to Flash

Ralink
for Joeqiao@synnex.com.tw
And Company Use Only

Issues?



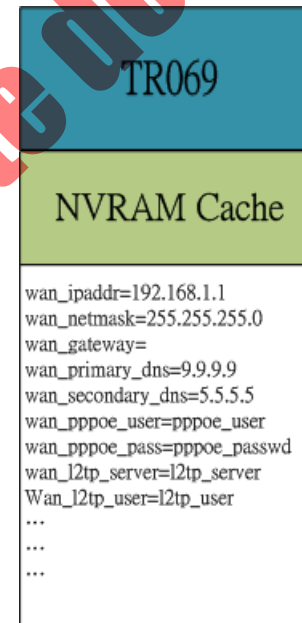
⇒ 1. nvram_bufset(wan_ipaddr, 10.10.10.254)

⇒ 3. nvram_bufget(wan_netmask) = ?



nvram_bufget(wan_ipaddr) = ?

nvram_bufget(wan_netmask) = ?



⇒ 2. nvram_bufset(wan_netmask, 255.0.0.0)

⇒ 4. nvram_bufget(wan_ipaddr) = ?

Ralink Website Document

for Joeqiao@synnex.com.tw

And Company Use Only

Solutions?

- **Use nvram_get() and nvram_set() instead**

- nvram_commit() takes long time (flash erase and write)
- Not efficient

- **Notification**

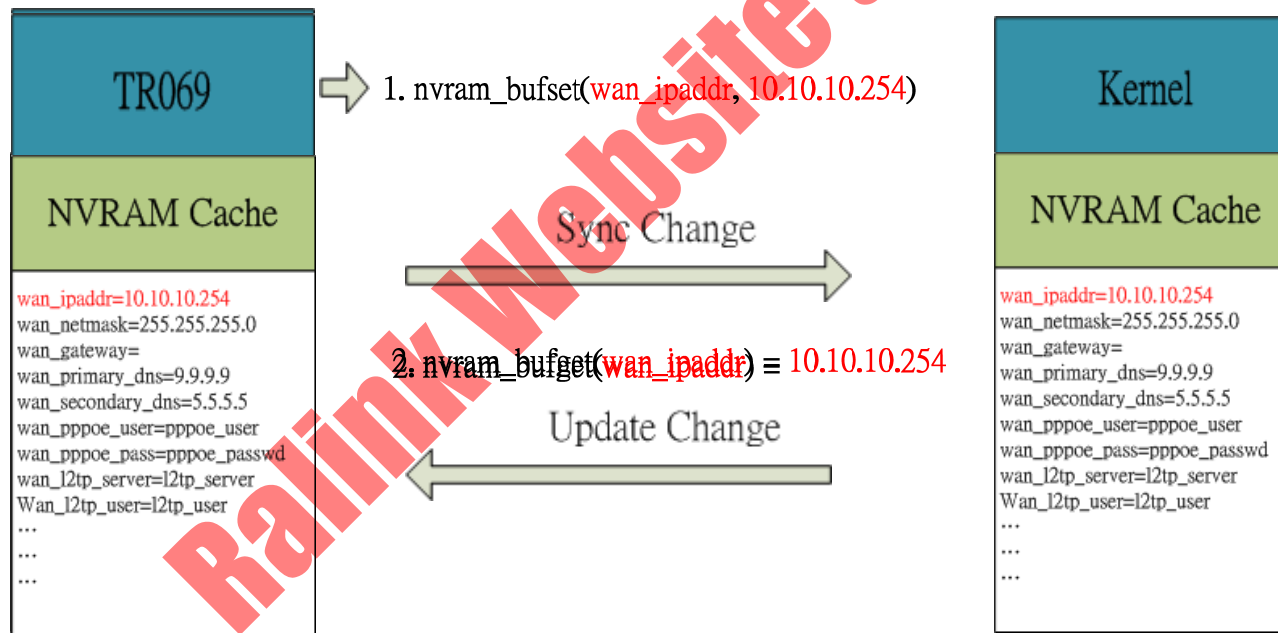
- Need to know which program needs to be notified(is using NVRAM)
- Other program may not be interested in it

- **Centralized Database**

- Every NVRAM user synchronize changes with it
- Keep the freshest data

for joeqiao@synnex.com.tw
And Company Use Only

New Design

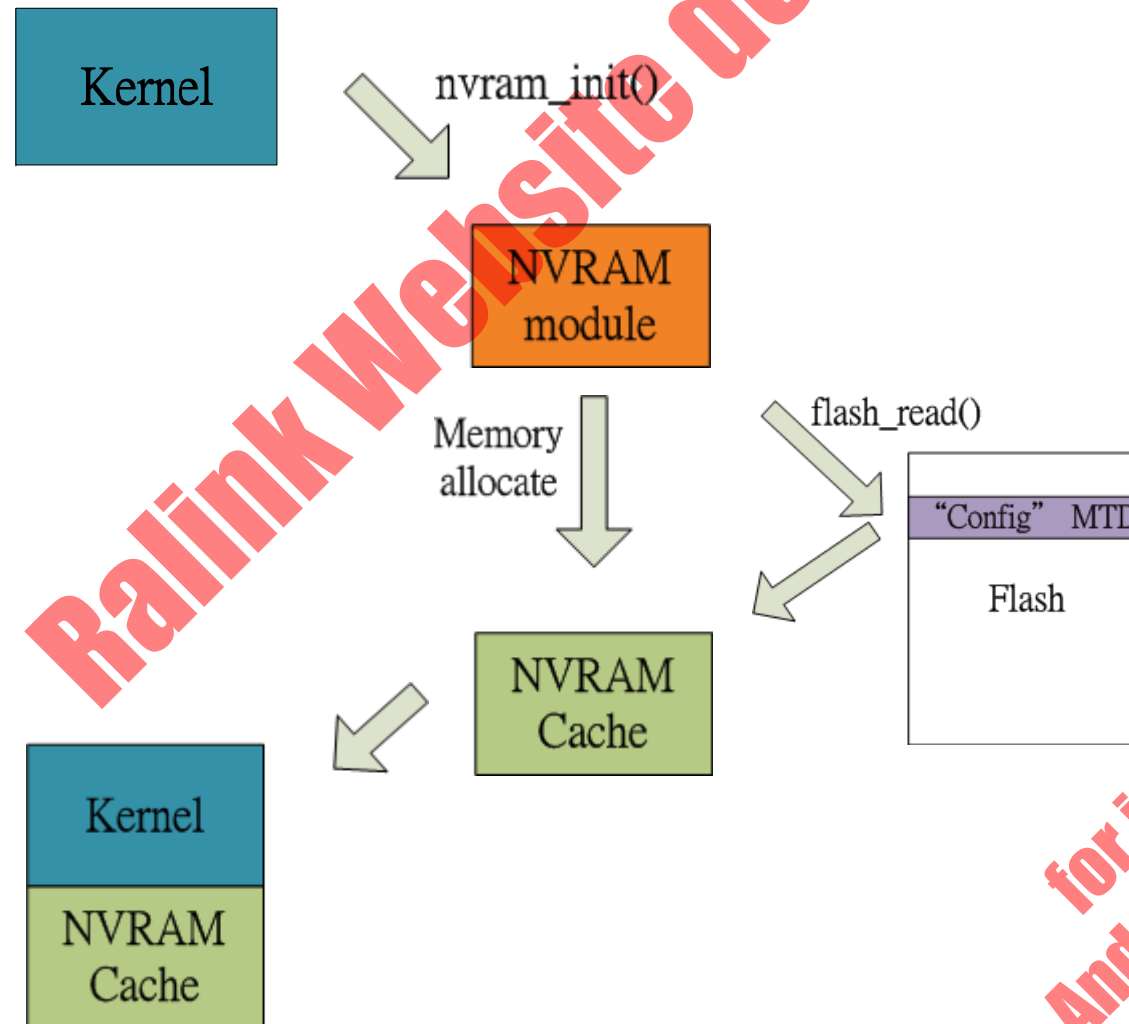


for joeqiao@synnex.com.tw
And Company Use Only

New Design: nvram_init()

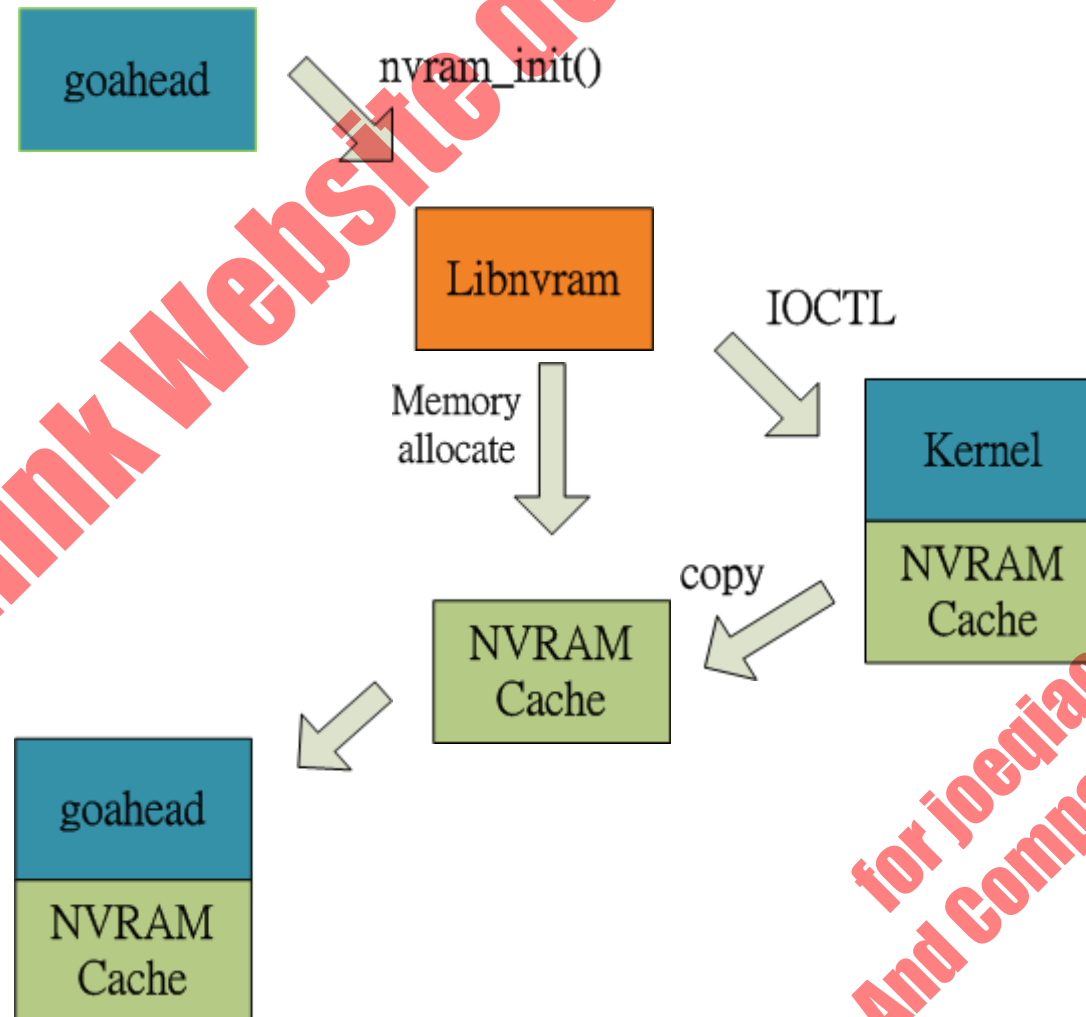
- Kernel NVRAM module will invoke nvram_init() at the boot time which does the same thing with previous user-mode nvram_init()
- User-space program also needs to call libnvram nvram_init(), but it gets the NVRAM data from Kernel NVRAM cache instead, not directly from Flash.

New Design: Kernel nvram_init()



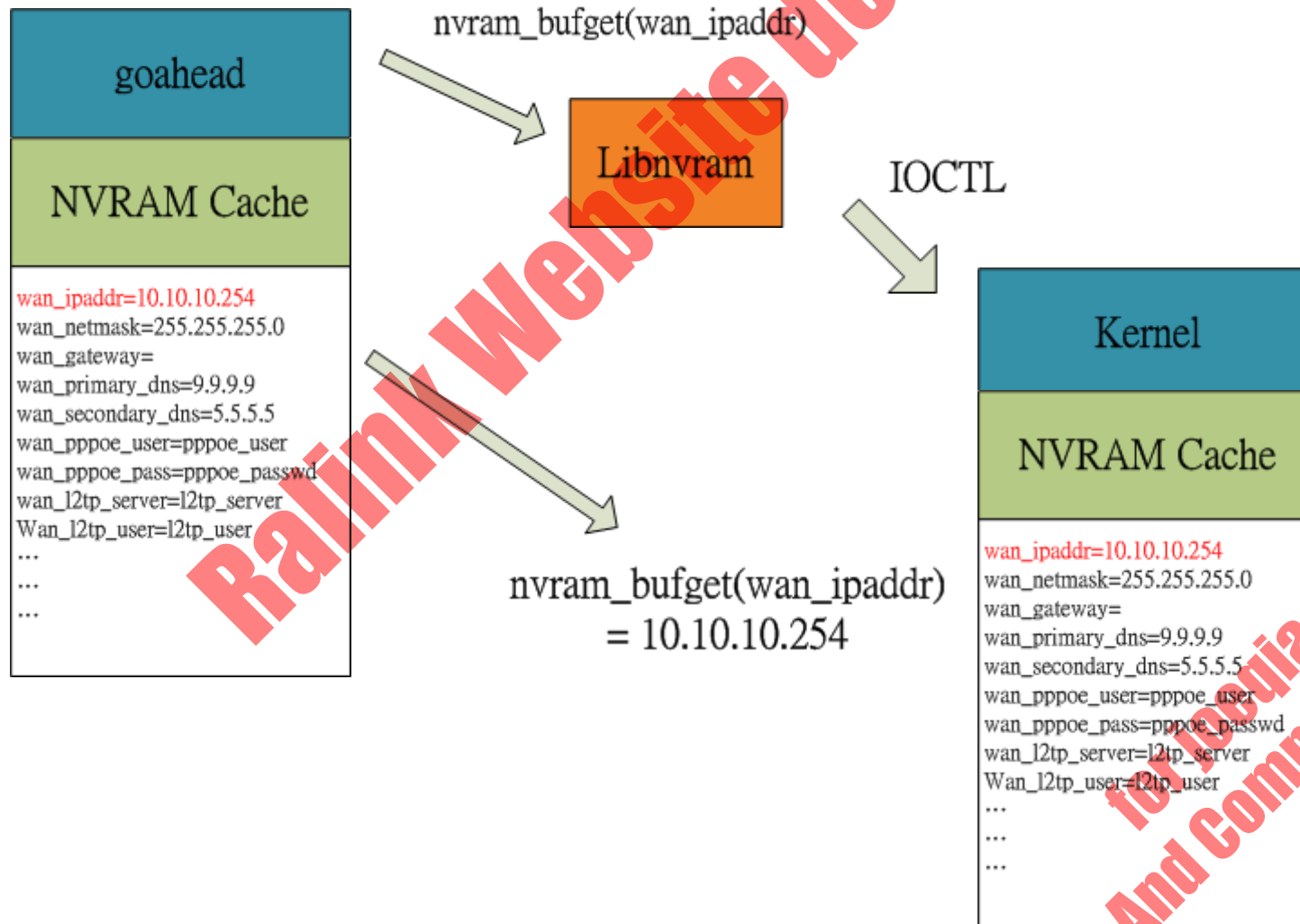
for joeqiao@synnex.com.tw
And Company Use Only

New Design: Libnvram nvram_init()

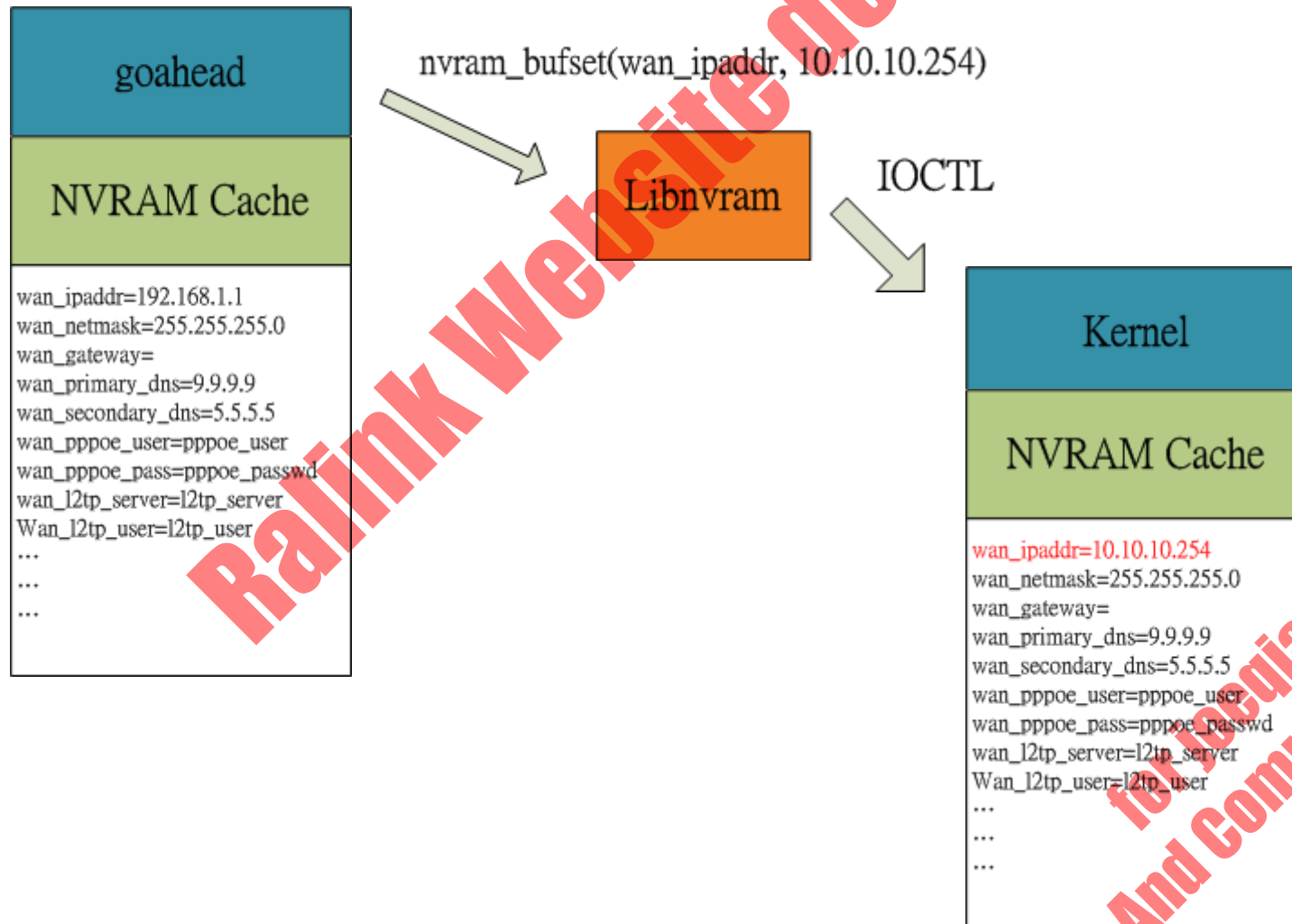


for joeqiao@synnex.com.tw
And Company Use Only

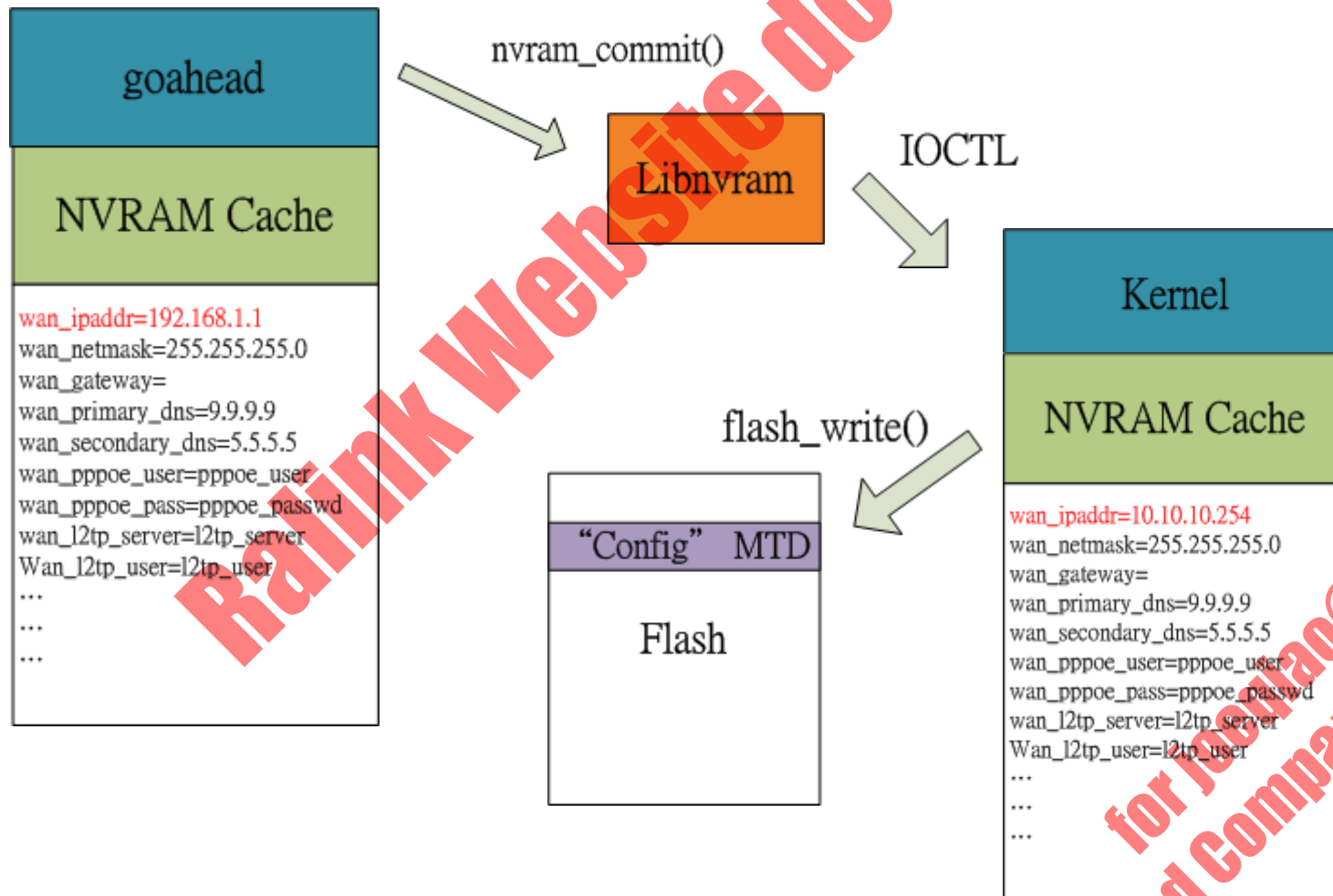
New Design: Libnvram nvram_get()



New Design: Libnvram nvram_set()



New Design: Libnvram nvram_commit()



New Design: Libnvram nvram_commit()

- Which copy of NVRAM cache will be committed to Flash?
 - → Kernel NVRAM cache
- Why?
 - → Kernel NVRAM cache keeps the freshest data

Ralink Website document

for joeqiao@synnex.com.tw
And Company Use Only

Code Review

- **New added files:**

- source/linux-2.6.21.x/arch/mips/rt2880/nvram.c
- source/linux-2.6.21.x/arch/mips/rt2880/nvram.h
- source/linux-2.6.21.x/arch/mips/rt2880/crc32.c

- **Modified files:**

- source/lib/libnvram/nvram_env.c
- source/lib/libnvram/nvram.h
- source/vendors/Ralink/RT3052/makedevlinks

Ralink Website document

for Joeqiao@synnex.com.tw
And Company Use Only

Test Scenario

- **Memory leak**

1. Program A keeps calling `nvruntime_bufget()`
2. Program B keeps calling `nvruntime_bufset()`
3. Use “free” command to check if system free memory is decreasing

- **Kernel NVRAM semaphore**

1. Program A calls X_1 times `nvruntime_bufget()`, Y_1 times `nvruntime_bufset()`, and Z_1 times `nvruntime_commit()`
2. Program A calls X_2 times `nvruntime_bufget()`, Y_2 times `nvruntime_bufset()`, and Z_2 times `nvruntime_commit()`
3. Increase global counter in Kernel NVRAM critical section
4. Check if the value of global counter = “ $X_1 + X_2 + Y_1 + Y_2 + Z_1 + Z_2$ ”

for joeqiao@synnex.com.tw
And Company Use Only

Test Scenario

- **Synchronization**

- Program A and Program B increase a nvram value in turn
- Program A increase X times, Program B increase Y times
- Check if that nvram value equals to $(X + Y)$

Ralink Website document

for joeqiao@synnex.com.tw
And Company Use Only