Confidential B

MEDIATEK

NVRAM Customization

Outline

- What is NVRAM
- How to add NVRAM Items
- NVRAM Description and Versioning

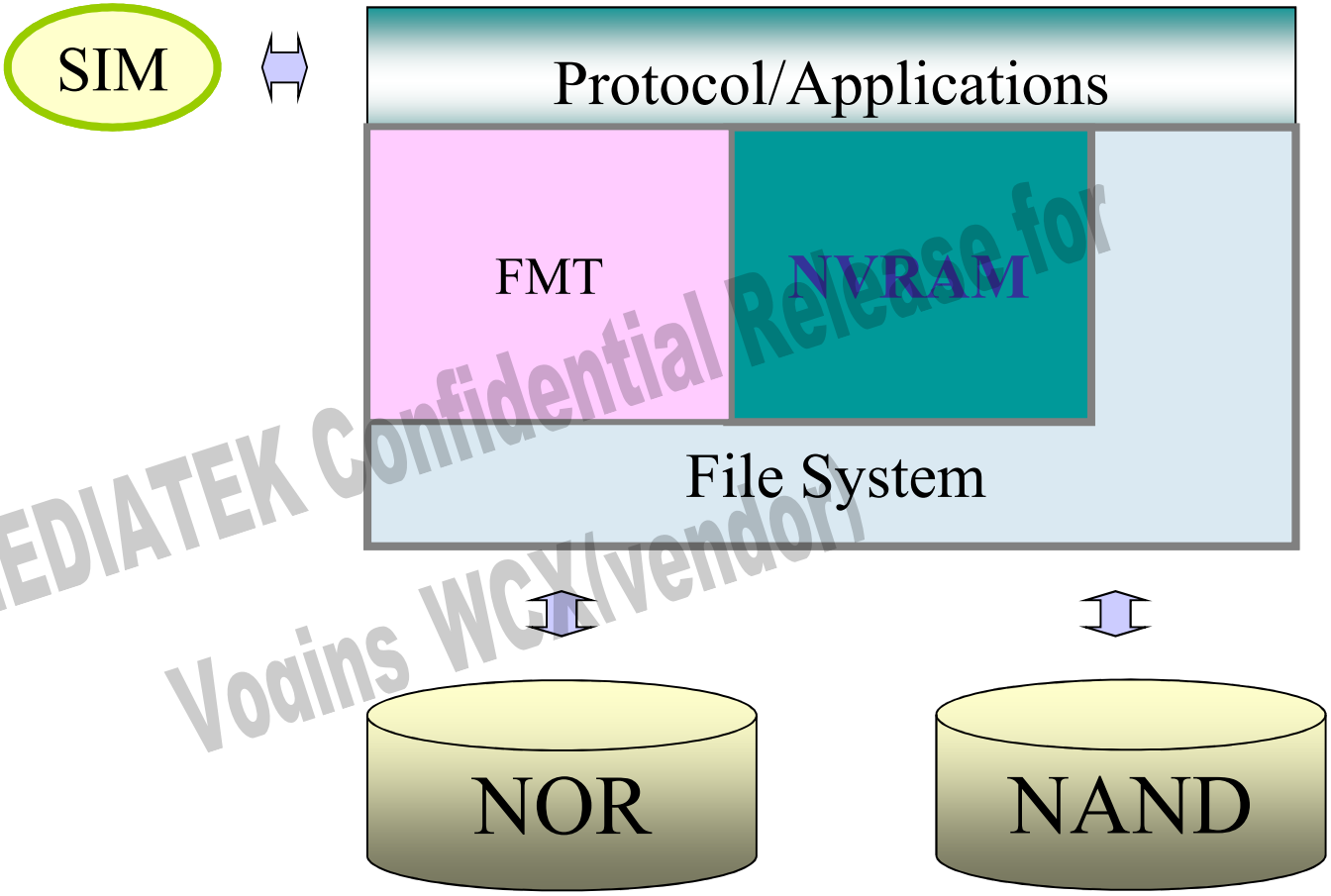
NVRAM Introduction

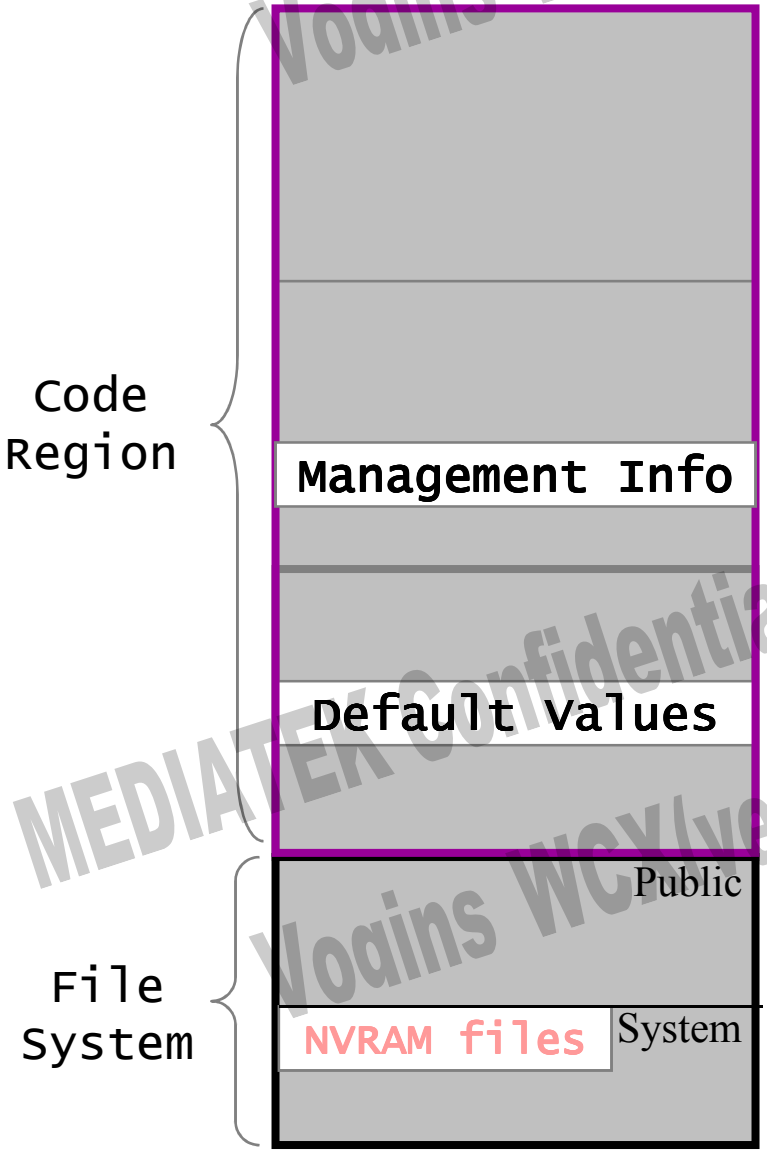
- NVRAM is an abstract, uniform interface to access data stored in the underlying non-volatile storage.
- NVRAM manages fixed-sized critical data of our system, a.k.a (logical) data items.
- NVRAM data items are stored as files in the File System.

Services NVRAM Provides:

- Data integrity:
  - Multiple copies of mission critical data
  - Auto Recovery (reset to factory default on checksum errors)
  - System-Record protection.
- Data security:
  - Encryption for confidential critical data
  - Write protection by S/W Lock

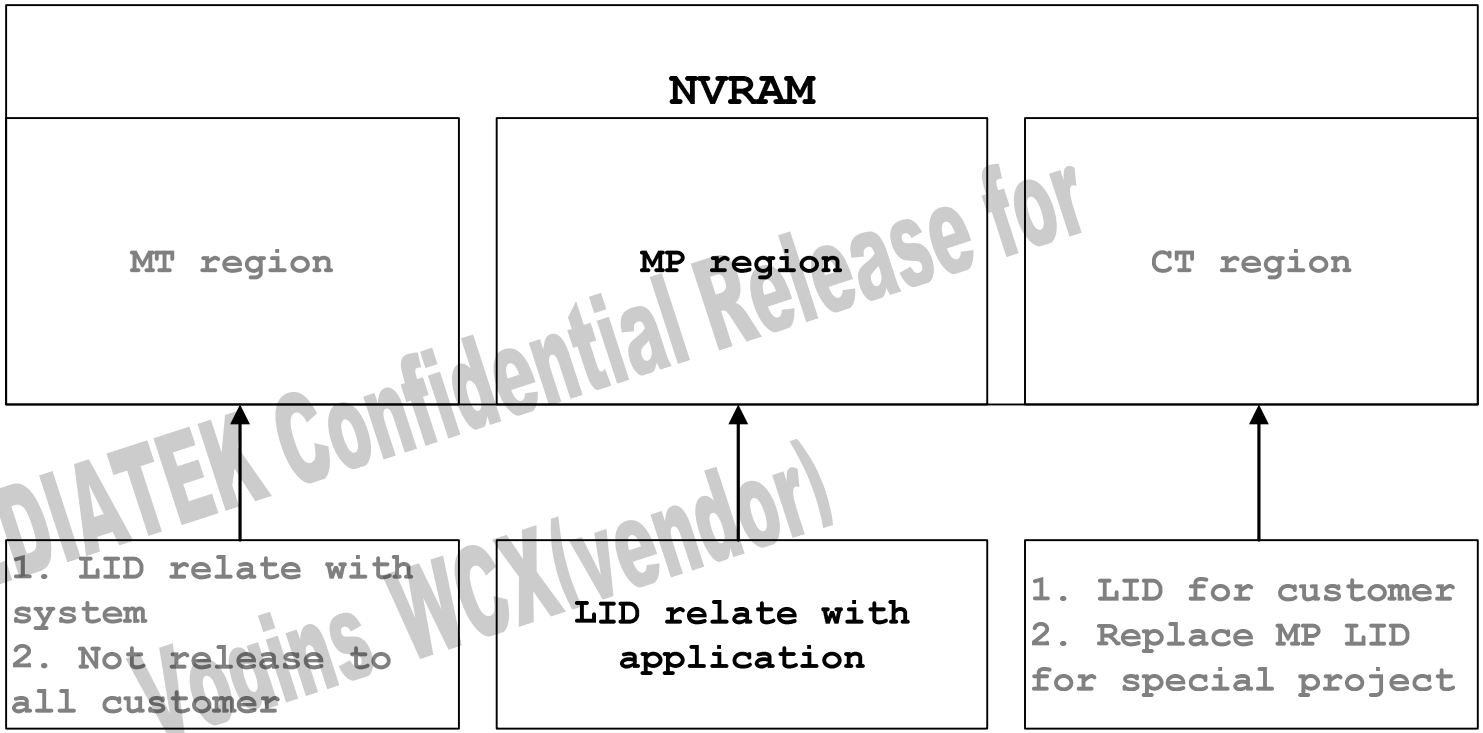
Maui Storage Overview





- NVRAM (Non-Volatile Random Access Memory)
  - A SW task to manage system data
  - The 1st task to be initiated
- Data Item
  - The logical unit conducted by NVRAM
- Management Info
  - details of every data item (in RW)
- Default values
  - Initial data of data items (in RO)
- NVRAM file
  - Instance of data item

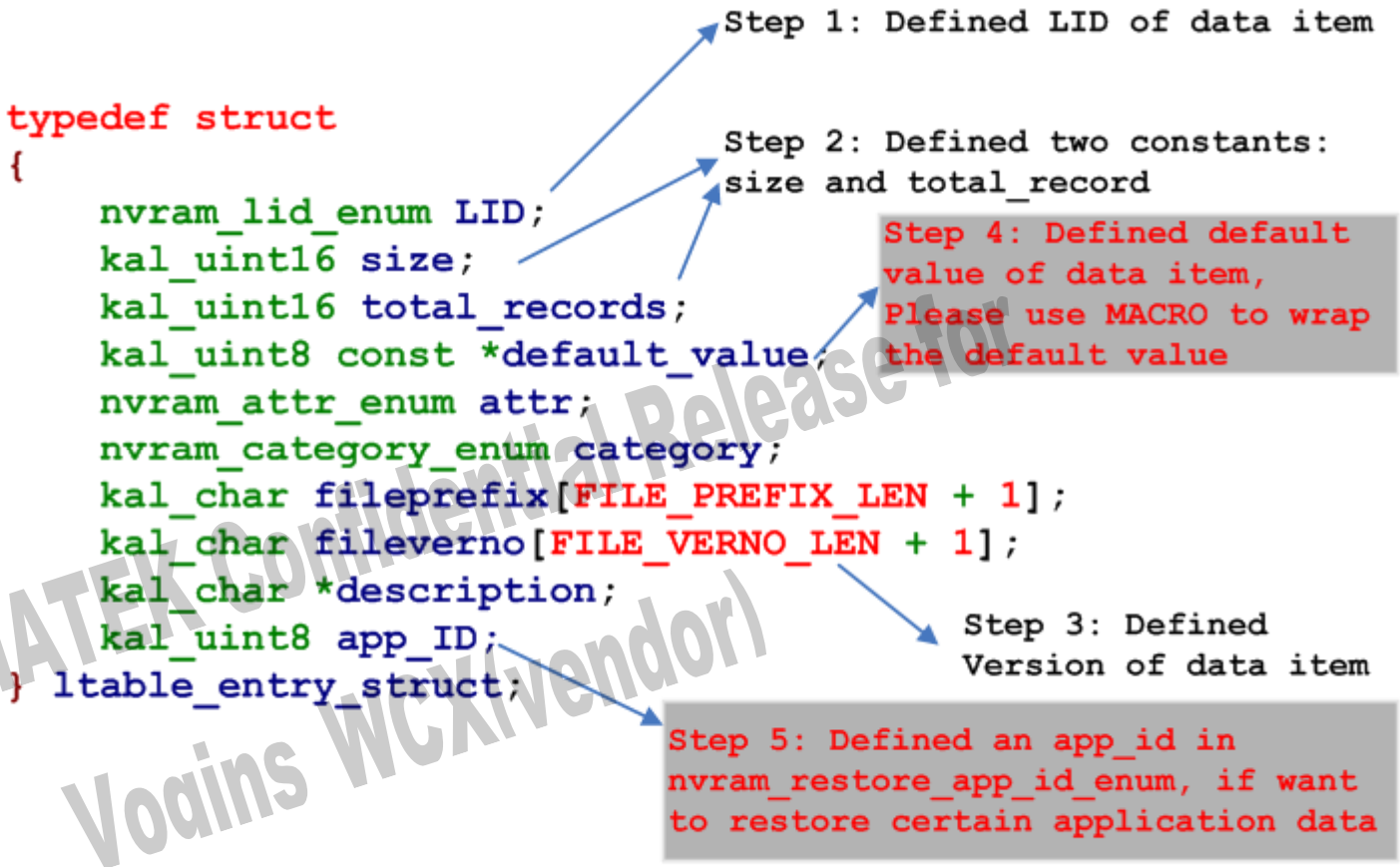
Configure data item – Add LID into which region?



How to Add NVRAM Item

- Define new NVRAM Item LID, size, and total number
  - nvram_data_items.h
  - nvram_common_defs.h
  - nvram_user_defs.h
- Setting NVRAM Item data items and default values
  - nvram_data_items.c
  - nvram_common_config.c
  - nvram_user_config.c
- Provide NVRAM Item version and structure
  - nvram_editor_data_item.h
  - common_nvram_editor_data_item.h
  - custom_nvram_editor_data_item.h

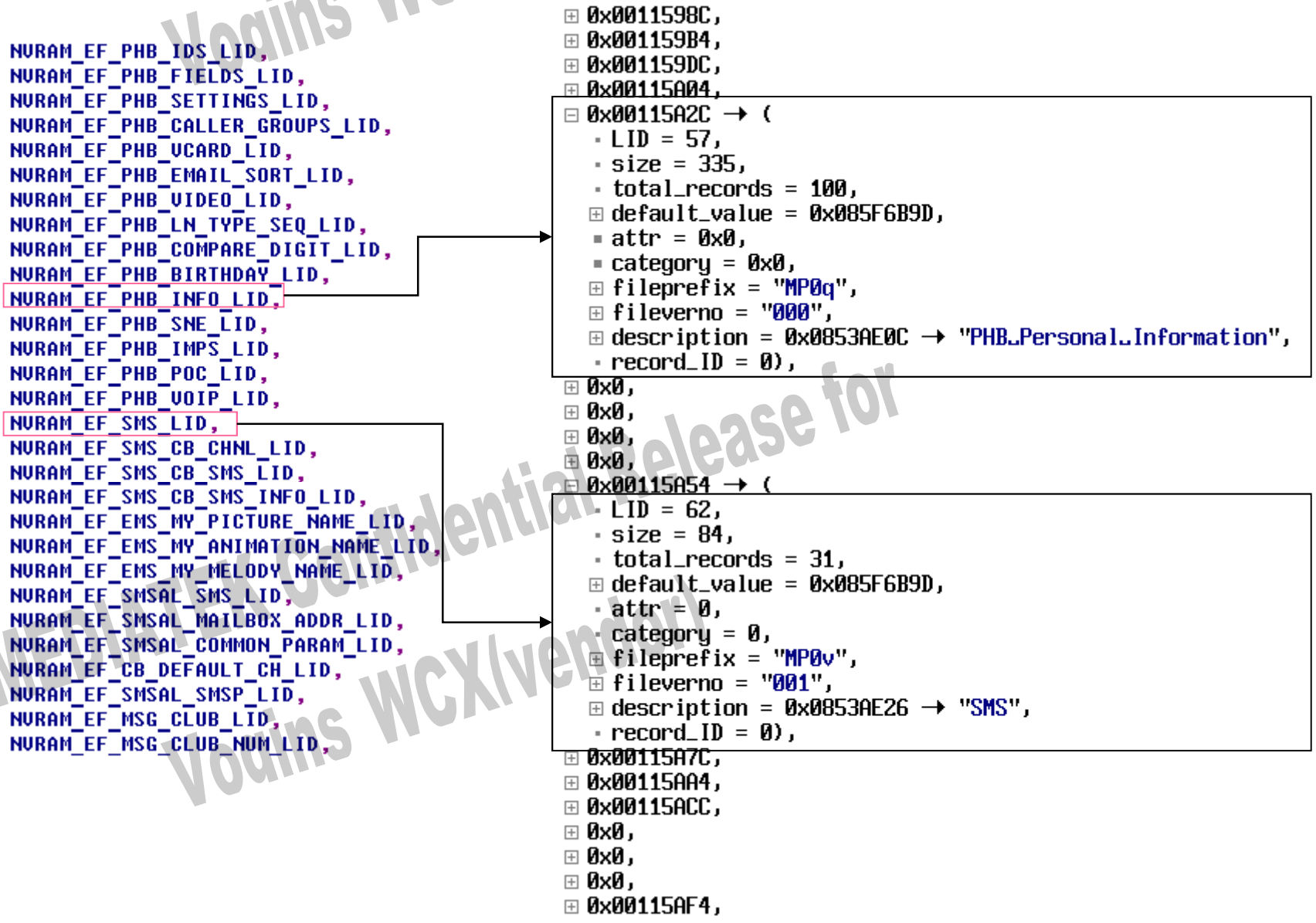
Configure date item – Overview



Step 6: Add a new entry for the logical data item

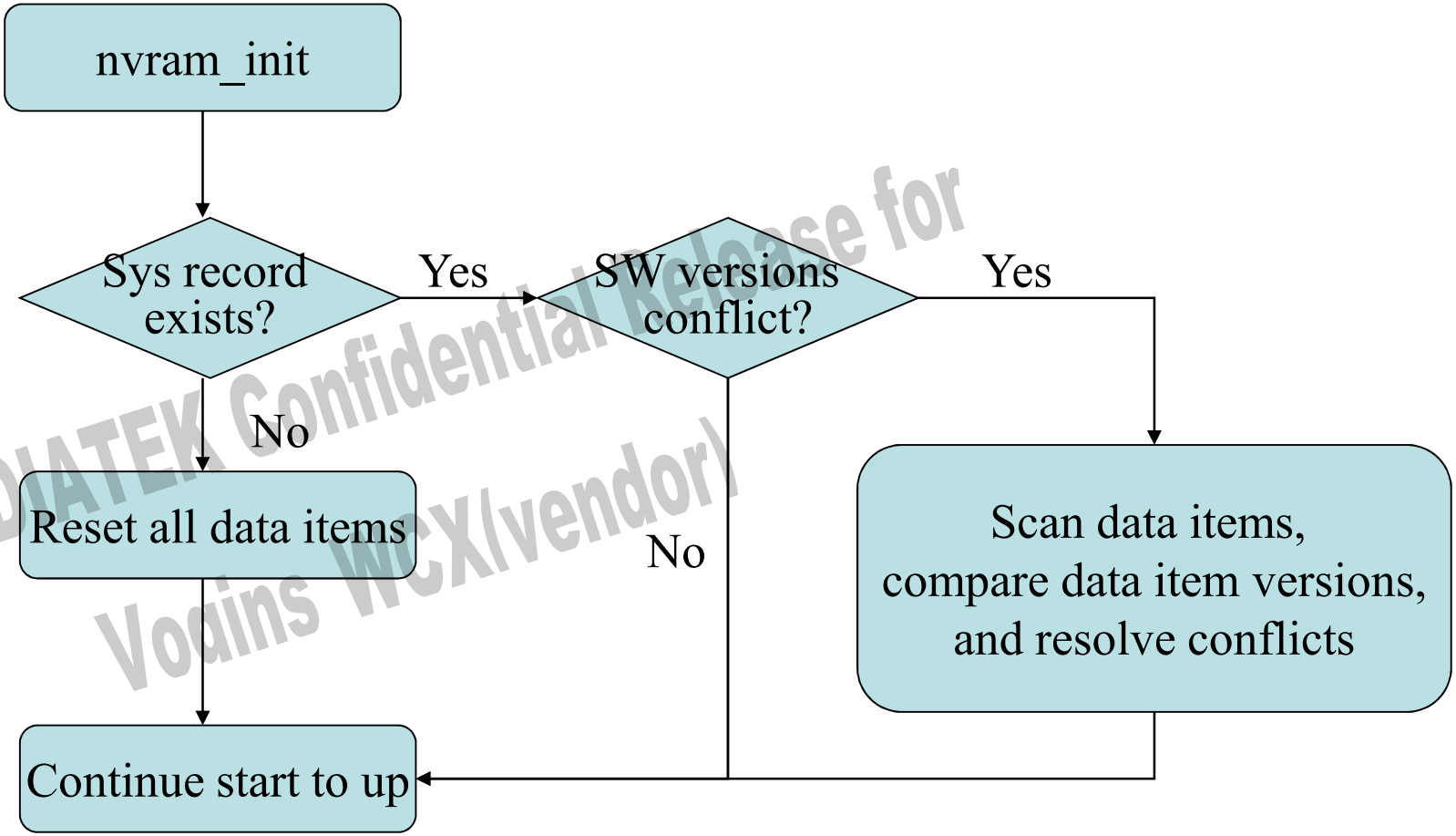
Step 7: Add bit-level description for tool

Structure name	ltable_entry_struct	
Field Name	**Type**	**Description**
LID	nvram_lid_enum(kal_uint16)	Logical Data Item ID.
size	kal_uint16	Size of one record. Either size of entire logical data item if it is transparent, or size of a record of this logical data item if it is linear-fixed.
total_records	kal_uint16	If this value is 1, this logical data item is implicitly defined as transparent; otherwise it is linear-fixed and this value defines number of records of this logical data item.
default_value	kal_uint8 const *	Default value defined for the LID. Must be NULL if default value is not supplied
attr	nvram_attr_enum(kal_uint32)	Attribute of the data item.
category	nvram_category_enum(kal_uint32)	Category that the data item belongs to; could be associated to nvram_reset_category_enum
fileprefix	kal_char[FILE_PREFIX_LEN+1]	The predefined name of the LID for internal access purpose. Please follow the rules of naming.
fileverno	kal_char [FILE_VERNO_LEN+1]	The version number (from 000 to 999) for automatic version resolution. Please follow the versioning rules.
description	kal_char*	This is a short human readable description that is effective only when important user or calibration data need to be backup.
app_ID	kal_uint8	This LID belongs to which application.



NVRAM Versioning System (1/2)

NVRAM behavior in the start-up process:



NVRAM Versioning System (2/2)

- No re-calibration :
  - With the versioning system, only updated NVRAM data items will be reset to the new default values. This saves lots of efforts.
- The data item version must be increased for the following cases
  - Size is changed, or
  - Number of sections is changed, or
  - Attribute is changed, or
  - Data structure is changed, or
  - New default values need to be applied.

Confidential B

Q & A

Appendix

Step 1 Define New Data Item

- File:
  - nvram_data_items.h (MT region)
  - nvram_common_defs.h (MP region)
  - nvram_user_defs.h (CT region)
- Name convention:
  - "***NVRAM_EF_[new data item name]_LID***"
- *Ex:* ***NVRAM_EF_PORT_SETTING_LID***

```
typedef enum
{
    NVRAM_EF_ADC_LID = NVRAM_LID_COMMAPP_BEGIN,
    NVRAM_EF_CACHE_BYTE_LID,
    NVRAM_EF_CACHE_SHORT_LID,
    NVRAM_EF_CACHE_DOUBLE_LID,
    NVRAM_EF_CUST_HW_LEVEL_TBL_LID,
    NVRAM_EF_NOTE_APP_LID,

 /* Add new entry in the tail without its compile optoin */
    NVRAM_EF_LAST_LID_COMMAPP
} nvram_lid_commapp_enum;
```

Note: please always add the new LID at the tail but before NVRAM_EF_LAST_LID_COMMAPP

Step 2,3 Define Size, Total and Version

- File:
  - nvram_data_items.h (MT region)
  - nvram_common_defs.h (MP region)
  - nvram_user_defs.h (CT region)
- name convention:
  - ***NVRAM_EF_[new data item name]_SIZE***
  - ***NVRAM_EF_[new data item name]_TOTAL***
- File:
  - nvram_editor_data_item.h (MT region)
  - common_nvram_editor_data_item.h (MP region)
  - custom_nvram_editor_data_item.h (CT region)
- name convention:
  - ***NVRAM_EF_[new data item name]_LID_VERNO***
- *Ex:*

```
#define   NVRAM_EF_PORT_SETTING_SIZE   16
#define   NVRAM_EF_PORT_SETTING_TOTAL   1
#define   NVRAM_EF_PORT_SETTING_LID_VERNO "000"
```

Step 4-1 Default Value

- File:
  - nvram_data_items.c (MT region)
  - nvram_common_config.c (MP region)
  - nvram_user_items.c (CT region)
- Type of default value:
  - ***NVRAM_EF_FF_DEFAULT***
  - ***NVRAM_EF_ZERO_DEFAULT***
  - ***Default value defined by Application***
- ***ex:***

```
#if defined(__MMI_FWU_VIA_HTTP__)
static nvram_ef_dm_fwu_http_setting_struct const NVRAM_EF_DM_FWU_HTTP_SETTING_DEFAULT[]=
{
    {"http://wap1.mtk.com.tw/",0},
};
#endif  /* #if defined(__MMI_FWU_VIA_HTTP__) */
```

Step 4-2 Default Value

- Use macro before default value
  - **NVRAM_SECUPACK**
    - data item in NVRAM_CATEGORY_SECUPACK
  - **NVRAM_CUSTPACK**
    - data item in NVRAM_CATEGORY_CUSTPACK
  - **NVRAM_NORMAL**
    - data item NOT in secupack and custpack

```
,{
    NVRAM_EF_MS_SECURITY_LID,
    NVRAM_EF_MS_SECURITY_SIZE,
    NVRAM_EF_MS_SECURITY_TOTAL,
    NVRAM_CUSTPACK(COMMON_NVRAM_EF_MS_SECURITY_DEFAULT),
    NVRAM_ATTR_MULTIPLE | NVRAM_ATTR_CONFIDENTIAL,
    NVRAM_CATEGORY_USER | NVRAM_CATEGORY_CUSTPACK,
    "MP45",
    VER(NVRAM_EF_MS_SECURITY_LID),
    "MS SECURITY",
    NVRAM_APP_RESERVED
}
```

```
,{
    NVRAM_EF_SML_LID,
    NVRAM_EF_SML_SIZE,
    NVRAM_EF_SML_TOTAL,
    NVRAM_SECRO((kal_uint8 *)&NVRAM_EF_SML_DEFAULT),
    NVRAM_ATTR_MSP | NVRAM_ATTR_CONFIDENTIAL,
#ifdef __SMART_PHONE_MODEM__
    NVRAM_CATEGORY_SECUPACK | NVRAM_CATEGORY_IMPORTANT,
#else
    NVRAM_CATEGORY_SECUPACK | NVRAM_CATEGORY_BRANCH,
#endif
    "MP33",
    VER(NVRAM_EF_SML_LID),
    "SIM-ME Lock",
    NVRAM_APP_RESERVED
}
```

Step 5 Define Application-ID

- File:
  - custom_nvram_config.h
- name convention:
  - ***NVRAM_APP_[application name]***
- *Ex:*

```
typedef kal_uint8 nvram_restore_app_enum;

typedef enum
{
    NVRAM_APP_PHNSET,
    NVRAM_APP_CAMCODER,
    NVRAM_APP_CAMERAPP,
    NVRAM_APP_EDITABLE_EQ,
    NVRAM_APP_Q05A_BROWSER,

    NVRAM_APP_TOTAL,
    NVRAM_APP_RESERVED = NVRAM_APP_PHNSET
}nvram_restore_app_id_enum;
```

Note:

If want to restore all data about certain application, RD should add an app-id. else can use NVRAM_APP_RESERED

Step 6 Add a new entry

- File:
  - nvram_data_items.c (MT region)
  - nvram_common_config.c (MP region)
  - nvram_user_items.c (CT region)
- *Ex:*

```
ltable_entry_struct logical_data_item_table_comm_app[] =
{
    {
  ,{
        NVRAM_EF_PORT_SETTING_LID,
        NVRAM_EF_PORT_SETTING_SIZE,
        NVRAM_EF_PORT_SETTING_TOTAL,
        NVRAM_NORMAL(NVRAM_EF_PORT_SETTING_DEFAULT),
        NVRAM_ATTR_AVERAGE,
        NVRAM_CATEGORY_USER | NVRAM_CATEGORY_SHADOW,
        "MP08",
        VER(NVRAM_EF_PORT_SETTING_LID),
        "Port Settings\0",
        NVRAM_RESERVED_VALUE
    }
};
```

Note: The fileprefix MUST be a new name, for example, if the last LID in the MP reign is MP07, the new one must be MP08

Step 7 Bit-Level Description

- Add bit-level description for META tool (Optional)
  - Add structure definition or include the header file if need
- File:
  - nvram_editor_data_item.h (MT region)
  - common_nvram_editor_data_item.h (MP region)
  - custom_nvram_editor_data_item.h (CT region)
- *ex:*

```
typedef struct
{
    kal_uint8 name[34];
    kal_uint8 index;
    kal_uint8 dummy;
} nvram_group_lists_struct;
```

```
LID_BIT VER_LID(NVRAM_EF_GROUPLIST_LID)
nvram_group_lists_struct *NVRAM_MAX_GROUP_NAME_TOTAL
{
    name:"CUG Name"
    {
    };

    index:"CUG Index"
    {
    };
};
```

MEDIATEK

www.mediatek.com