

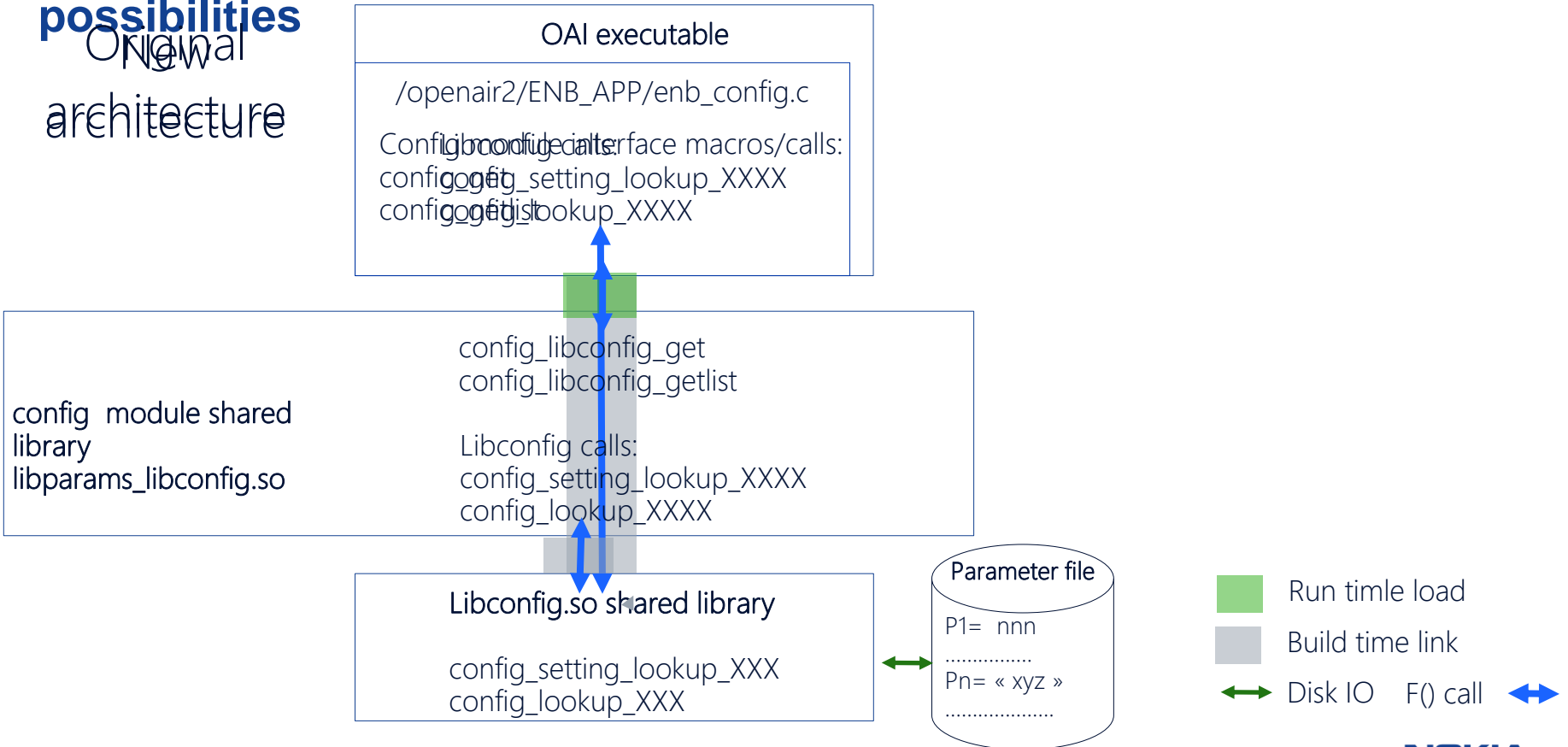
Toward an OAI Modular Architecture: The Configuration Module Example

François TABURET Nokia Bell-Labs
Nov 7th and 8th 2017

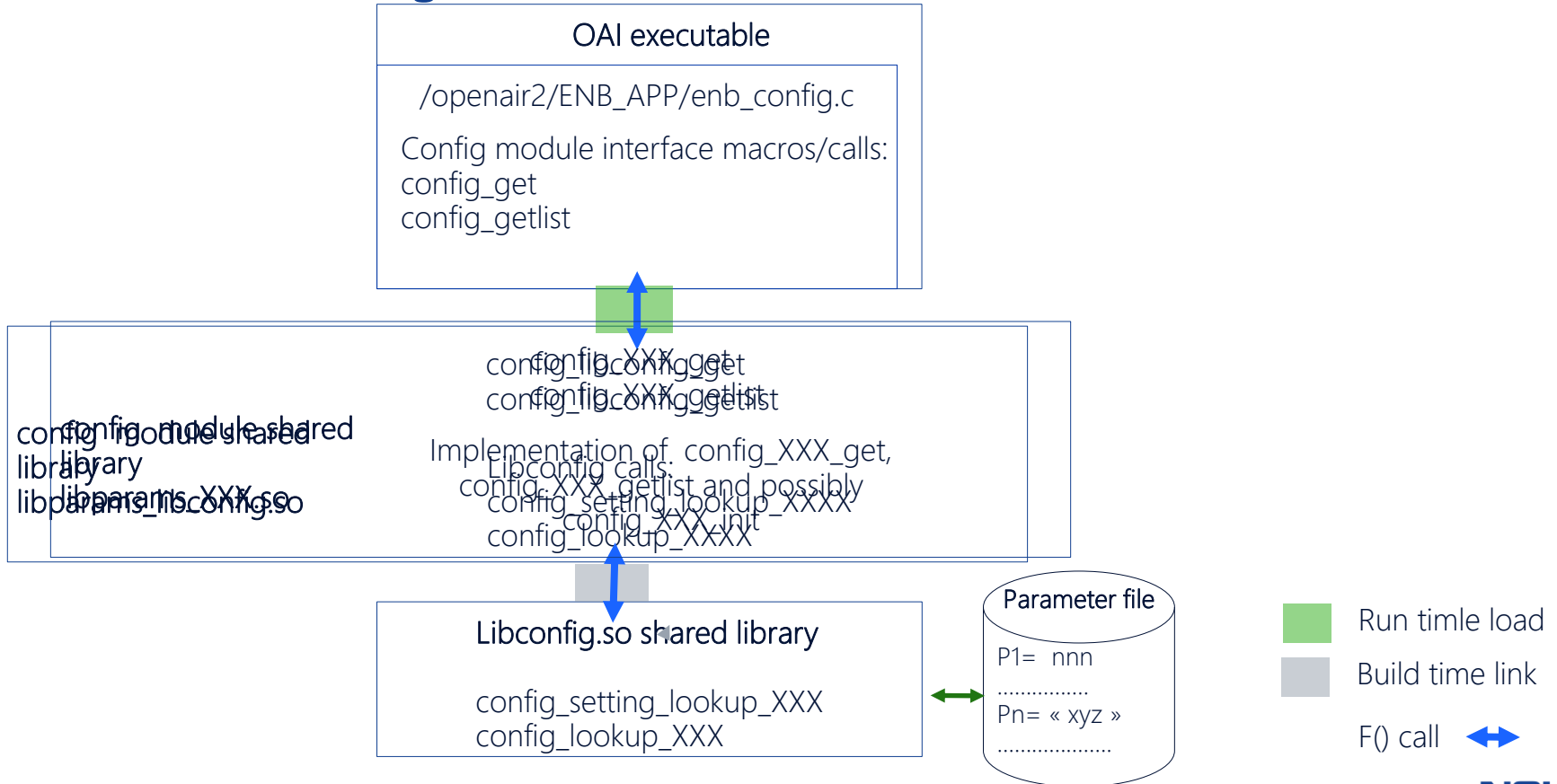


Using shared libraries to get a flexible architecture, enhancing OAI possibilities

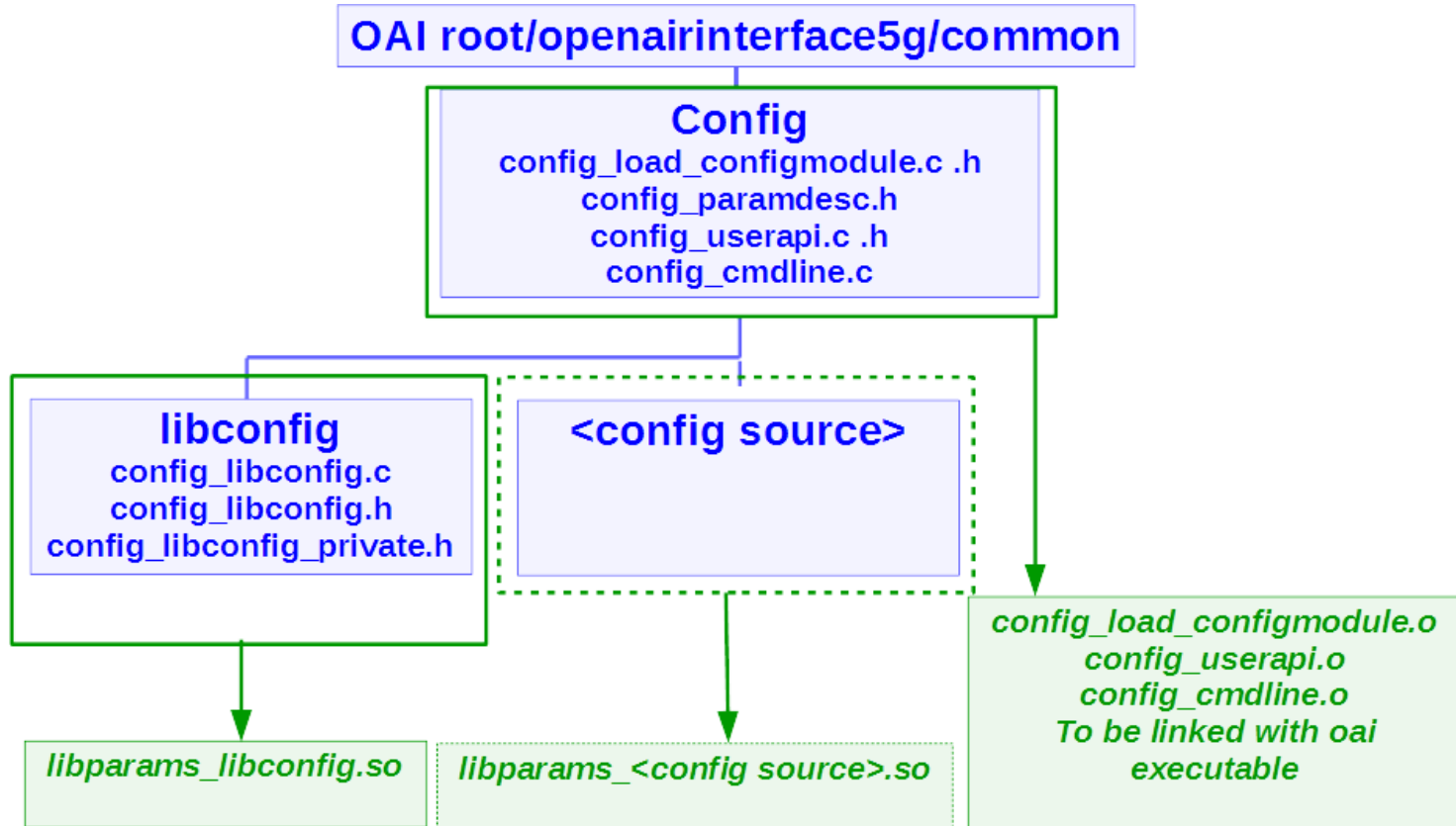
Original
New
architecture



Using shared libraries to get a flexible architecture, implementing alternative configuration sources



Configuration module implementation: source files and directories



Using shared libraries to get a flexible architecture: configuring the configuration module using the command line

Traditional command line

```
./lte-softmodem -O /usr/local/oai/enb.band7.tm1.25PRB.usrpb210.conf
```

```
[CONFIG] get parameters from libconfig /usr/local/oai/enb.band7.tm1.25PRB.usrpb210.conf  
[CONFIG] function config_libconfig_init returned 0  
[CONFIG] config module libconfig loaded  
[LIBCONFIG] config: 1/1 parameters successfully set, (1 to default value)  
# /dev/cpu_dma_latency set to 0us  
[LIBCONFIG] log_config: 4/4 parameters successfully set, (1 to default value)  
[LIBCONFIG] log_config: 37/37 parameters successfully set, (31 to default value)
```

Using shared libraries to get a flexible architecture: configuring the configuration module using the command line

Command line to get more configuration module traces

```
./lte-softmodem -O libconfig:/usr/local/oai/enb.band7.tm1.25PRB.usrpb210.conf:dbgI5
```

```
[HW][I][SCHED][eNB] eNB_thread_prach started on CPU 7, sched_policy = SCHED_FIFO , priority = 99, CPU Affinity= CPU_0 CPU_1 CPU_2 CPU_3 CPU_4 CPU_5 CPU_6 CPU_7  
Sending sync to all threads  
[CONFIG] calling config module end function...  
got sync (ru_thread)  
[CONFIG] free 1 config parameter pointers  
[CONFIG] free 12 config value pointers
```

Dbgl values:

- 1: print parameters values
- 2: print memory allocation/free debug messages
- 4: print command line processing messages

Using shared libraries to get a flexible architecture: configuring the configuration module using the configuration file

```
Config :  
{  
    Debugflags = 5;  
};
```

Using the configuration module, as a developer

Between the `load_configmodule` and `end_configmodule` calls (both are in `lte-softmodem.c`) any part of the code can use the configuration module.

- Any functional entity can manage its configuration using the `config_get` or `config_getlist` calls → Easier project management
- Example: the log system configuration has been moved to `log.c/log.h`
- Developers just need to define an array describing the parameters he needs for his module, and pass it to `config_get` call.
- `openair2/ENB_APP/enb_config.[c,h]`: best example of configuration module usage
- `openair2/UTIL/LOG/log.c`, in `log_getconfig`: dynamic build of parameters definition

Using the configuration module functionalities at run-time

Any parameter can be specified on the command line

- Can be used to reduce the number of necessary configuration files
- Easier test
- Use dbgI5 option to get the parameter name

Example 1: modify bandwidth

```
./lte-softmodem -O  
/usr/local/oai/nokiaconf/rcc.band7.tm1.if4p5.50PRB.nuc82.alulabo.conf -  
-eNBs.[0].component_carriers.[0].N_RB_DL 25
```

Example 2: disable logs

```
./lte-softmodem -O  
/usr/local/oai/nokiaconf/rcc.band7.tm1.if4p5.50PRB.nuc82.alulabo.conf  
--log_config.global_log_online 0
```

configuration evolutions

- Bug fixes soon delivered (string parameters allocation problems)
- Support parameters propagation between entities of a splitted eNB
 - Today Eurecom implemented a simple UDP protocol
 - Might be usefull to get a generic mechanism, taking as input a parameter definition, as already implemented in the configuration module

Configuration module interface: API

`configmodule_interface_t *load_configmodule(int argc, char **argv)`

- Parses the command line options, looking for the `-O` argument
- Loads the `libparams_<configsource>.so` (today `libparams_libconfig.so`) shared library
- Looks for `config_<config source>_init` symbol and calls it, passing it an array of string corresponding to the « : » separated strings used in the `-O` option
- Looks for `config_<config source>_get`, `config_<config source>_getlist` and `config_<config source>_end` symbols which are the three functions a configuration library should implement. `Get` and `getlist` are mandatory, `end` is optional.
- Stores all the necessary information in a `configmodule_interface_t` structure, which is of no use for caller as long as we only use one configuration source.

End_configmodule()

- Free memory which has been allocated by the configuraton module since its initialization.
- Possibly calls the `config_<config source>_end` function

Configuration module interface: API

`int config_get(paramdef_t *params,int numparams, char *prefix)`

- Reads as many parameters as described in `params`, they must all be in the same configuration file section
- Calls the `config_<config source>_get` function
- Calls the `config_process_cmdline` function
- `params` points to an array of `paramdef_t` structures which describes the parameters to be read and which will be used to store the value
- `numparams` is the number of entries in the `params` array
- `prefix` is a character string to be appended to the parameters name, it defines the parameters position in the configuration file hierarchy.
- The returned value is the number of parameters which have been assigned a value or -1 if a severe error occurred

Configuration module interface: API

```
int config_libconfig_getlist(paramlist_def_t *ParamList,  
paramdef_t *params, int numparams, char *prefix)
```

- Reads multiple occurrences of a parameters array
- Used for L1s, Rus, MACRLCs, eNBs sections, which are all lists
- Calls the config_<config source>_get function for each list occurrence
- params points to an array of paramdef_t structures which describes the parameters in each occurrence of the list
- ParamList points to a structure, where paramarray field points to an array of paramdef_t structure, allocated by the function. It is used to return the values of the parameters.
- The returned value is the number of occurrences in the list or -1 in case of severe error

Configuration module interface: parameters definition

Fields	Description	I/O
Optname	Name , as used in the parameter file , without the hierarchie prefix	I
Helpstr	Help string, printed when using -h option (not yet fully implemented)	I
Paramflags	Flags to modify get behavior or inform caller	I/O
Pointers to value (union)	Config module puts the value here	O
Default values (union)	Possibly used for parameters not set in the file	I
Type	TYPE_<XXX>	I
numelt	Allocated size for string value, number of occurrences for arrays, or lists	I/O

Configuration module interface: parameters flags

Flag	Description	I/O
PARAMFLAG_DISABLECMDLINE	Parameter cannot be set from command line	I
PARAMFLAG_MANDATORY	Parameter Must be explicitly set in cconfig file	I
PARAMFLAG_DONOTREAD	Ignore this parameter	I
PARAMFLAG_NOFREE	Do not free memory allocated by the config module for this parameter value	I
PARAMFLAG_BOOL	Used only when processing command line, parameter value will be set to true when specified without any value	I

Configuration module interface: parameters flags

Flag	Description	I/O
PARAMFLAG_MALLOCINCONFIG	Memory for parameter value has been allocated by the module	O
PARAMFLAG_PARAMSET	Parameter value has been found in the configuration file or the command line	O
PARAMFLAG_PARAMSETDEF	Parameter value has been set to the default value	O

NOKIA