Lite IDMEF Library

API specification

(Release v0.1)

Author: R.Lupu

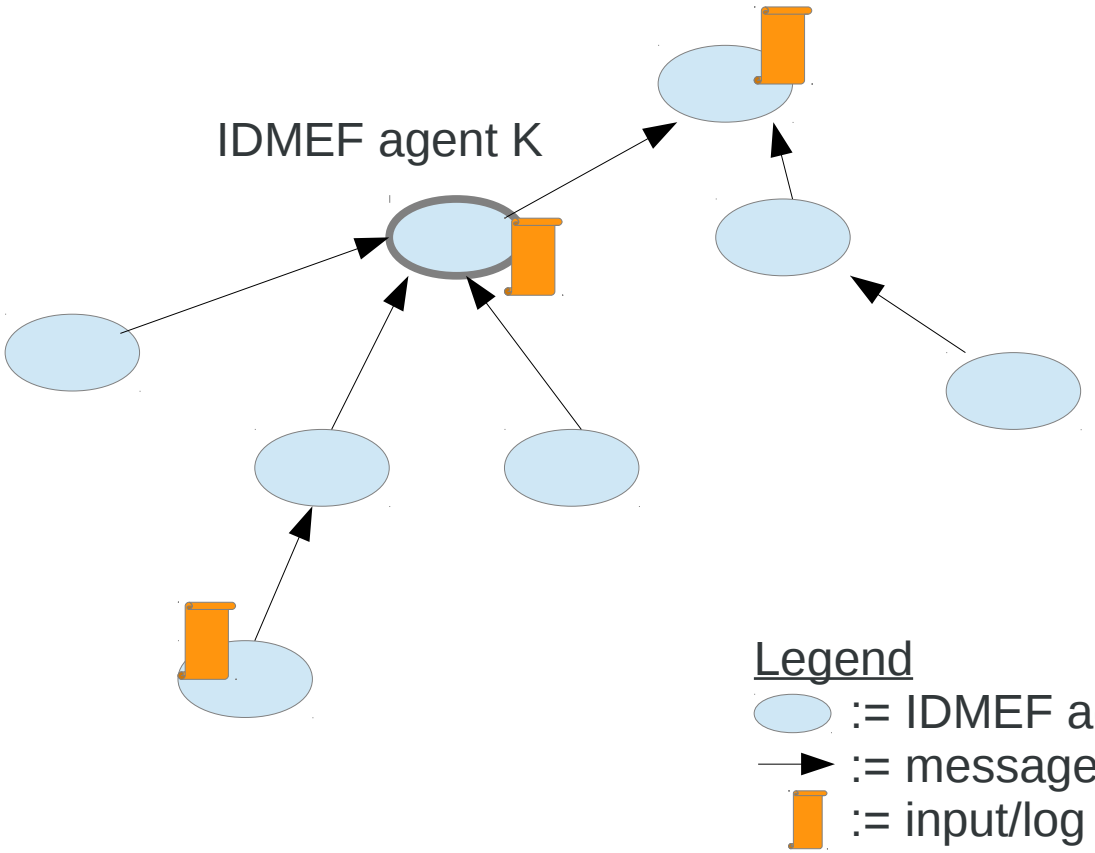
December, 2015

Lite IDMEF Library
-outline-

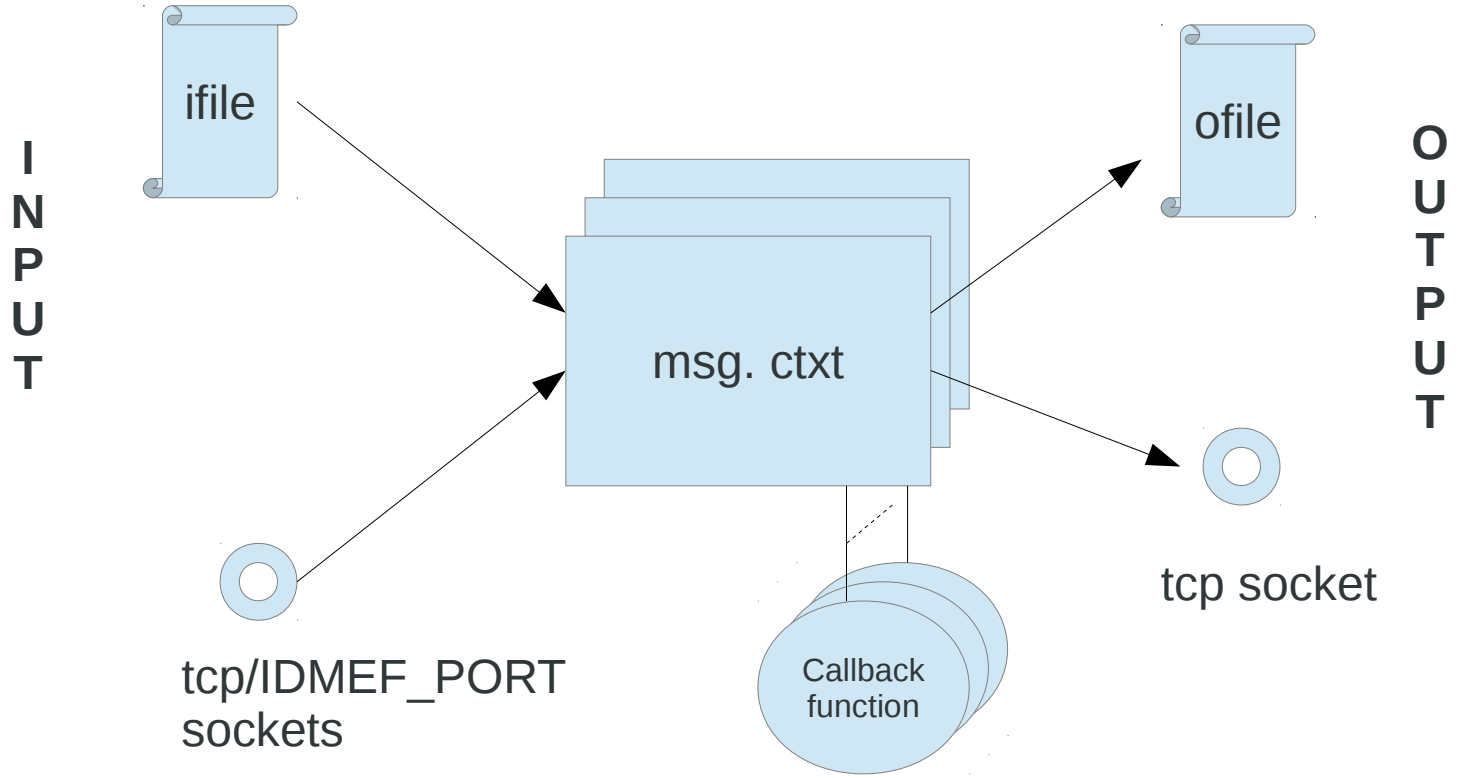
- Overall concept
- APIv0.1 specification
- APIv0.1 grammar
- References

Lite IDMEF Library

Overall goal: to support IDMEF-based signaling plane deployment for dynamic distributed ID architectures.



Lite IDMEF Library
-overall concept-



Lite IDMEF Library

The message context concept is at the core of the library's design in order to provide improved flexibility and efficiency. Within the user application process that uses the library several IDMEF contexts may coexist. Only one IDMEF message can be processed (i.e. build up, modified or received and parsed) inside the message context at a given moment. Typically, each context specifies a different message structure (e.g. Alert, Heartbeat) and distinct input and output interfaces.
Nevertheless, once it has been built up the message structure can be reused as many times as needed by similar successive messages. The context's interface could operate in one of the following modes: IDMEF_MODE_SOCK, IDMEF_MODE_FILE, IDMEF_MODE_FS.
Whenever a new message is parsed by the receiver, the user-defined callback function is run. Sending the current message through the output interface is optional by invoking write operation from inside callback function left at the programmer decision. At the receiver side efficient parsing is achieved through tag/attribute enabling mechanisms leveraged in a similar way the sender builds up a new message. In other words, the receiver shall enable the tags/attr. he/she wants to read/write before a new message has to come.

Beware, each time the message structure is modified (i.e. add/delete tag or (re)/set attribute) it shall be (re)compiled.

A common quite compact and versatile API has been designed to support context operation and message manipulations at sender, as well as at receiver.

Lite IDMEF Library
-API v0.1-

Context manipulation	Message structure configuration
✗ idmef_new() ✗ idmef_free() ✗ idmef_setcb() ✗ idmef_chcon() ✗ idmef_compile() ✗ idmef_write()	✗ idmef_<tagname>_addtag() ✗ idmef_<tagname>_deltag() ✗ idmef_<tagname>_gettag() ✗ idmef_<tagname>_setattr() ✗ idmef_<tagname>_rstattr() ✗ idmef_<tagname>_wrattr() ✗ idmef_<tagname>_rdattr()

Note: see the list of tag names supported by the current release.

Lite IDMEF Library
-interactions definitions-

- **char idmef_new(idmef_ifs_t *, idmef_t **, void (*)(void))**

  It establishes a new message context; it can be called several times while working with the library; the function arguments are: the pointer to the context input-output specification structure, the return-value argument of the newly allocated context and the pointer to the user-defined callback function.

  Returns: 0 for SUCCESS.

```
typedef struct {
        unsigned char     mode_in;
        unsigned char     *filename_in;
        unsigned char     mode_out;
        unsigned char     *filename_out;
        unsigned char     *ipaddr_out;
} idmef_ifs_t;
```

  char idmef_free(idmef_t *)

  Release the message context specified as pointer argument.

  Returns: 0 for SUCCESS.

Lite IDMEF Library
-interactions definitions(cont'd)-

- **char idmef_setcb(idmef_t *, void (*)(void))**

  NOT YET IMPLEMENTED. Given the context as argument it changes the related callback function with the one specified.

  Returns: 0 for SUCCESS.

- **char idmef_chcon(idmef_t *, idmef_ifs_t *)**

  UNTESTED. It changes the given context output receivers.

  Returns: 0 for SUCCES.

- **char idmef_compile(idmef_t *)**

  It prepares internal context data structures for receiving or idmef_write() operation; it shall be called after each modification of the message structure in order to have changes take effect.

  Returns: 0 for SUCCESS.

- **char idmef_write(idmef_t *)**

  Sends current message through the previously configured context output interfaces.

  Returns: 0 for SUCCESS.

Lite IDMEF Library
-interactions definitions(cont'd)-

- **char idmef_<tagname>_addtag(idmef_<tagname>_t *, unsigned int, void **)**

  Adds a new child IDMEF tag to the (parent)tag specified as argument. The tag id. is delivered as second argument together with related options (see library's header file “**idmef_plugin.h**” for the set of tag identifiers and attributes supported). The reference to the new tag is available via third return-value argument. It may be called by sender side to build up/modify an IDMEF message structure and by receiver side to enable parsing of this type of tags(with attributes) specified (e.g. for the use cases where the IDMEF agent is just relaying messages, no parsing is possible and therefore increased performance achievements).

  Returns: 0 for SUCCESS.

- **char idmef_<tagname>_deltag(idmef_<tagname>_t *, unsigned int, unsigned char)**

  Deletes the IDMEF tag specified as second argument from the parent tag specified as first argument (see library's header file “**idmef_plugin.h**” for the set of tag identifiers supported). It may be called by sender side and by receiver side to disable parsing of this type of tags (efficiency). Whenever there are siblings tags the third arg is assigned the related index, otherwise 0 value is assigned.

  Returns: 0 for SUCCESS.

- **void *idmef_<tagname>_gettag(idmef_<tagname>_t *, unsigned int, unsigned char)**

  It returns the pointer to the specified child tag as 2nd argument from the given parent tag as 1st argument; the 3rd argument provides the tag's index (zero for not siblings tags).

  Returns NULL if either failed to found the tag or an internal error has been encountered.

Lite IDMEF Library
-interactions definitions(cont'd)-

- **char idmef_<tagname>_setattr(idmef_<tagname>_t *, unsigned int)**

  Callable by sender side to define message structure and by receiver to enable parsing of this attribute. The IDMEF attribute is specified by second argument (several attributes may be specified).

  Returns: 0 for Success.

- **char idmef_<tagname>_rstattr(idmef_<tagname>_t *, unsigned int)**

  Disables IDMEF attributes of some tag specified as first argument.

  Returns: 0 for SUCCESS.

- **char idmef_<tagname>_wrattr(idmef_<tagname>_t *, unsigned int, unsigned char *, unsigned int)**

  Writes attribute specified as 2nd argument of the tag specified as 1st argument. The new value shall be formatted as an ASCII sequence and given via 3rd argument with the length specified by the 4th argument. This action shall be achieved subsequently either to tag+attribute definition followed by message compilation or new message parsing (i.e. inside user-defined callback function).

  Returns: 0 for SUCCESS.

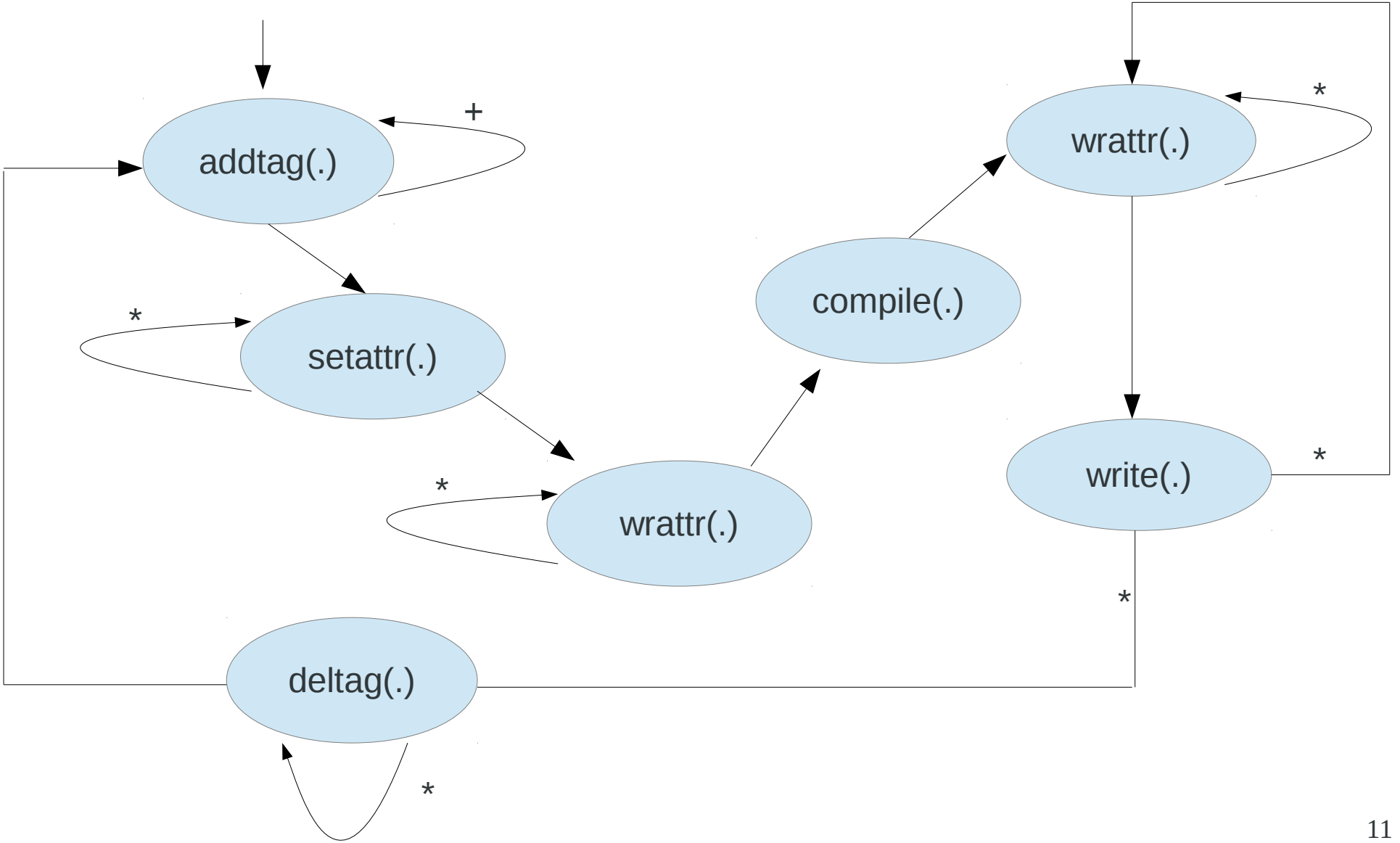
- **char idmef_<tagname>_rdattr(idmef_<tagname>_t *, unsigned int, unsigned char **, unsigned int *)**

  It provides access to the value of a specified attribute via the 3rd argument which is of type return-value. 1st argument specifies the related tag, the attribute is specified by the 2nd argument.

  Returns: 0 for SUCCESS.

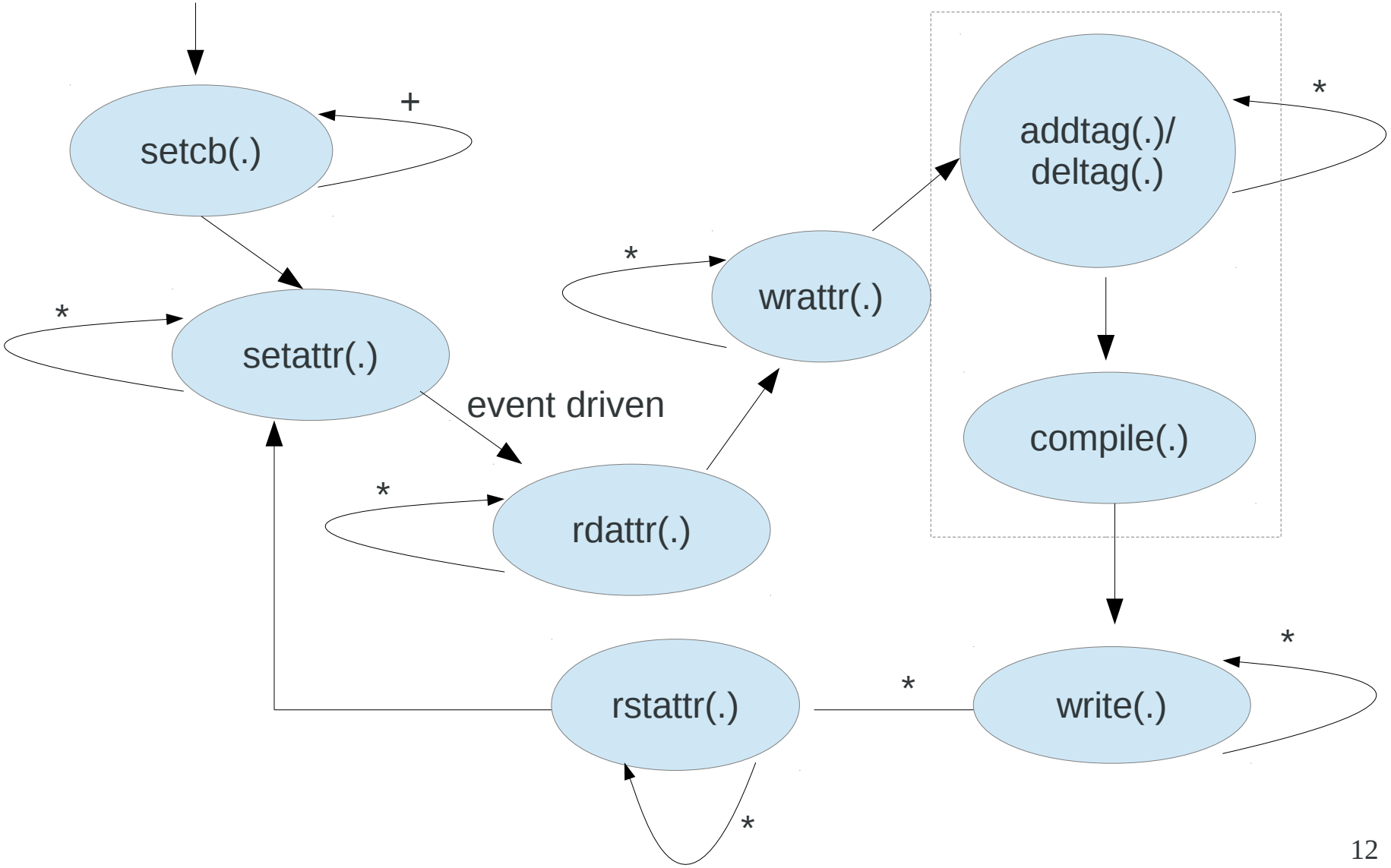
Lite IDMEF Library

-sender's API grammar-



Lite IDMEF Library

-receiver's API grammar-



Lite IDMEF Library
-API v0.1-

Tag names supported:

ALERT, ANALYZER, CREATETIME, TARGET, SOURCE, NODE, ADDR, NETMASK, SERVICE, PORT, CLASSIFICATION, REFERENCE, NAME, URL, ...

Lite IDMEF library
-bibliography-

[RFC4765] H. Debar, et al. “The intrusion detection message exchange format (IDMEF)”, IETF, March 2007, www.ietf.org/rfc/rfc4765.txt