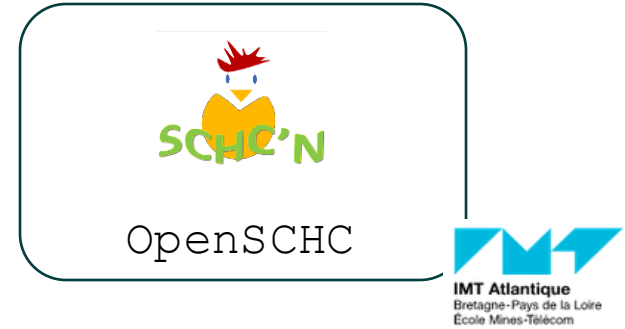
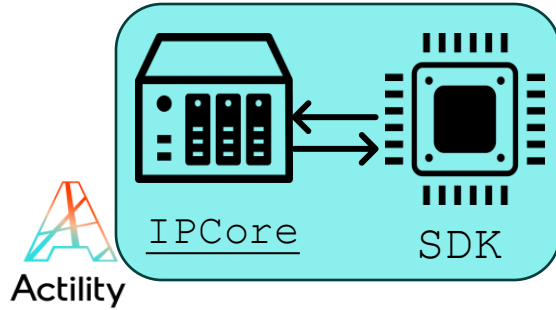


# lab . SCHC Full SDK

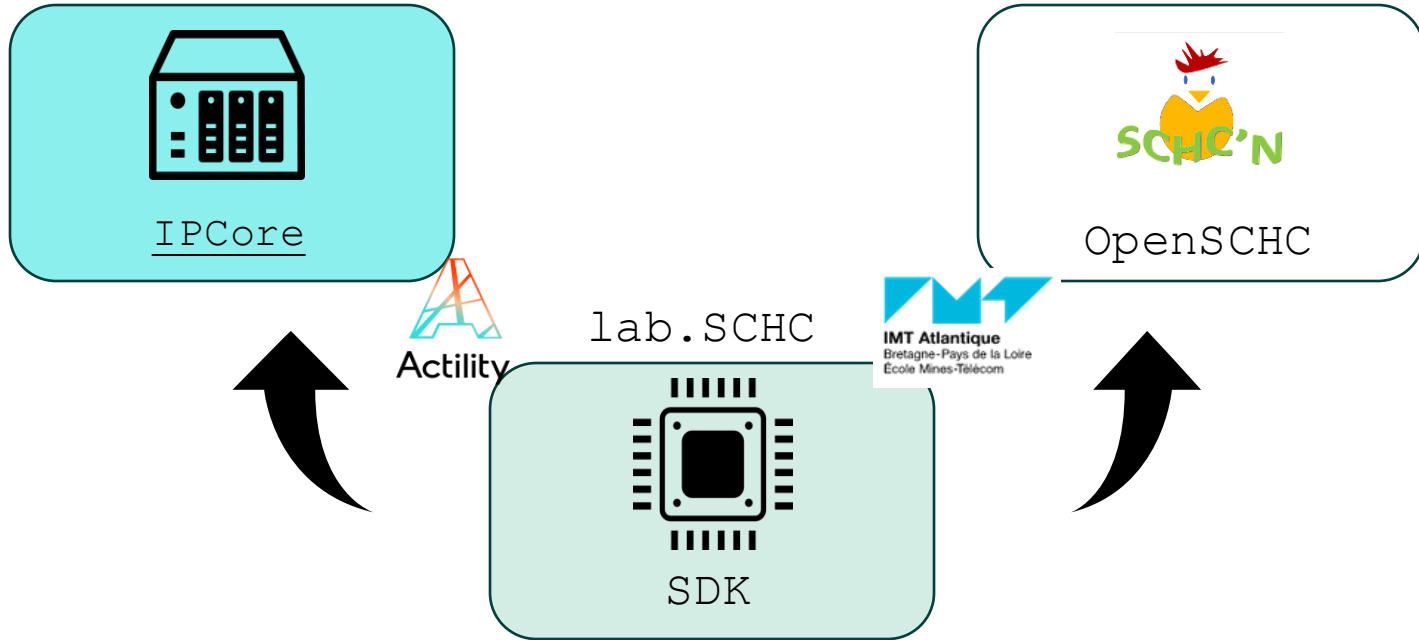
## A SCHC implementation in C

Javier Fernandez

# lab.SCHC



# lab.SCHC



## Meet lab(dot)SCHC

Welcome to lab.SCHC, the research and development laboratory dedicated to advancing the SCHC mechanism for the Internet of Things (IoT). Our mission is to make SCHC accessible to everyone by developing innovative solutions that optimize communication and data transfer. Join us as we pave the way for a more connected and sustainable future.

FullSDK

New!

Resources

Contact Us

SCAN ME



FIRST RELEASE  
FIRST RELEASE  
FIRST RELEASE



- [lab-schc.fr](https://lab-schc.fr)

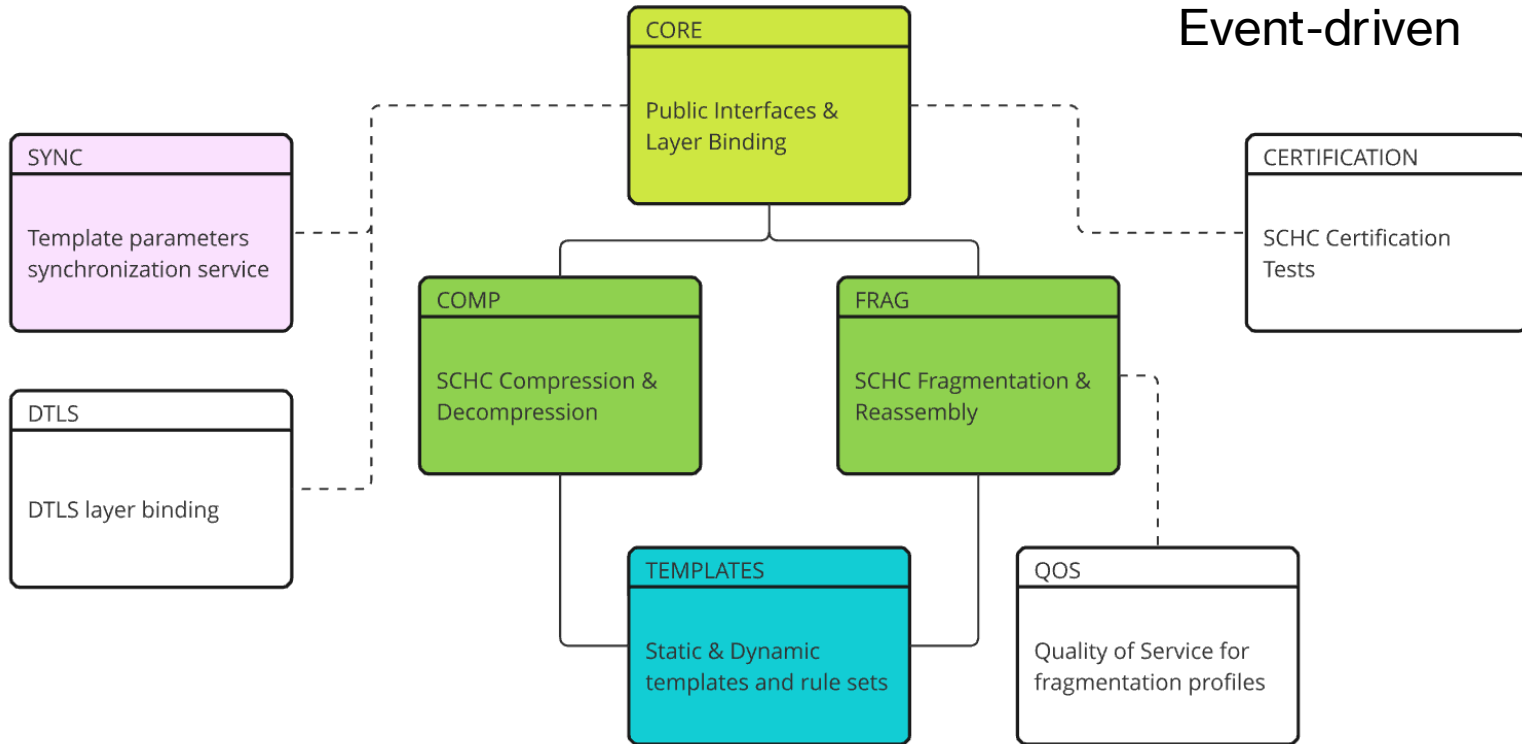
## Resources

|  |             |
|--|-------------|
| <b>lab.SCHC/fullSDK</b>                                      | <b>New!</b> |
| Industrial-grade Open Source embedded SCHC implementation    |             |
| lab.SCHC FullSDK GitLab project                              |             |
| <b>lab.SCHC FullSDK Docs</b>                                 | <b>New!</b> |
| Documentation for SCHC SDK                                   |             |
| Requisites, building and debugging, API functions, and more! |             |
| <b>Actility IPCore</b>                                       |             |
| SCHC Gateway implementation for business                     |             |
| High performance and stability                               |             |
| <b>OpenSCHC</b>  |             |
| Open source SCHC implementation in Python                    |             |
| For SCHC Gateway and Device                                  |             |

# lab.SCHC full-sdk

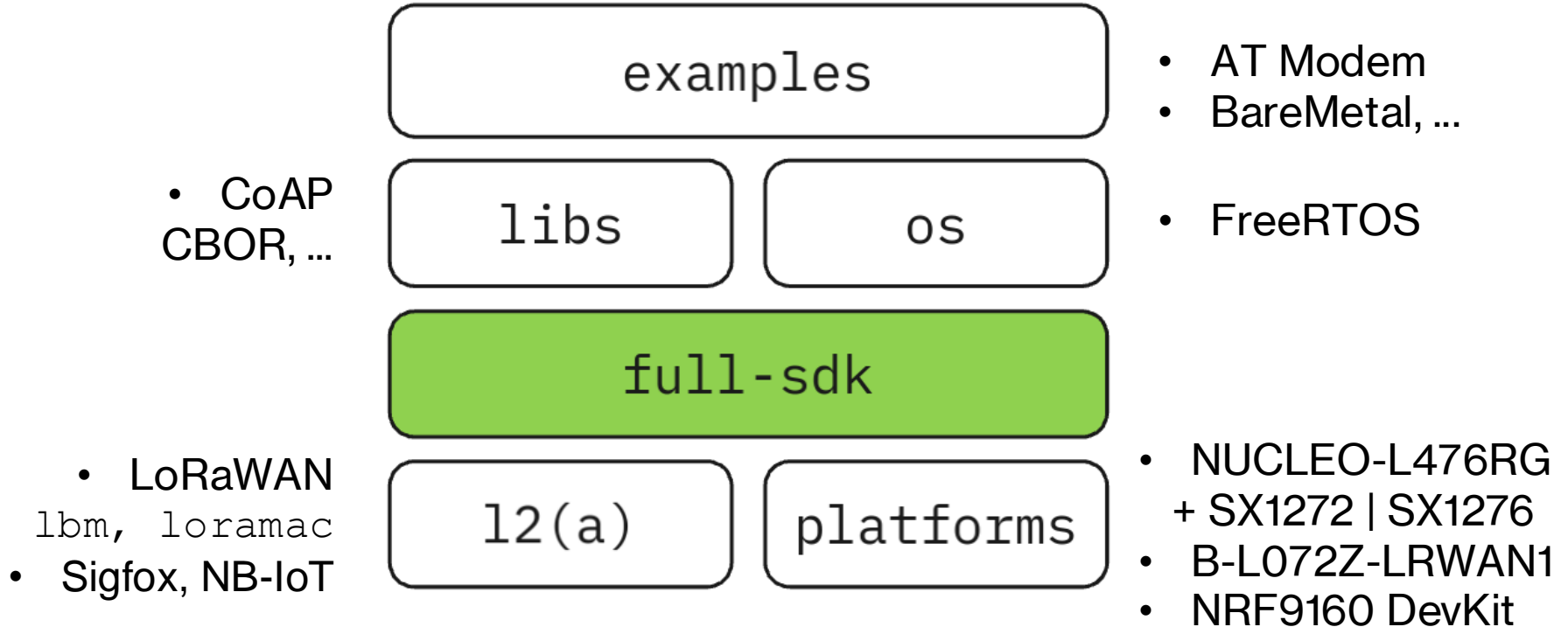
- [lab-schc.fr](http://lab-schc.fr)

Layered &  
Event-driven



# lab.SCHC full-sdk-delivery

- [lab-schc.fr](http://lab-schc.fr)



# lab.SCHC Docs

• [lab-schc.github.io](https://lab-schc.github.io)

## 1. The SCHC Mechanism:

- Learn the basics of SCHC
- The SCHC adaptation layer

## 2. Getting started:

- Requisites and compatibility
- Building and debugging applications

## 3. FullSDK Concepts:

- General architecture
- Available interfaces
- How to use

## 4. FullSDK Reference Manual:

- Public API functions
- Sample applications

## Requisites and Compatibility

The hardware, operating systems, libraries, and third-party software in this section are recommended.

| Support             |   |
|---------------------|---|
| OS                  | <b>AT Modem UDP Client</b><br>This example illustrates an end-to-end communication from a U and downlink packet transmission from a device (a board for the   |
| Hardware/<br>Board  | <ol style="list-style-type: none"><li>1. The device application sends messages to the cloud application.</li><li>2. The cloud application displays every received message and</li><li>3. The device application displays every received the message</li></ol> |
| Hardware/<br>Shield | The example demonstrates how to use the SCHC SDK using the IPv6 UDP frames. The demo files can be found in the <code>/at-modem</code>   |
| Hardware/<br>Serial | <pre>cd lab-schc-examples/at-modem/udp-client/<br/>head README.md</pre>   |
| L2A                 | <b>Operation</b><br>This example considers a UDP server application with the IP in the SCHC Gateway with the IPv6 address <code>5454::2</code> and a  |

### Operation

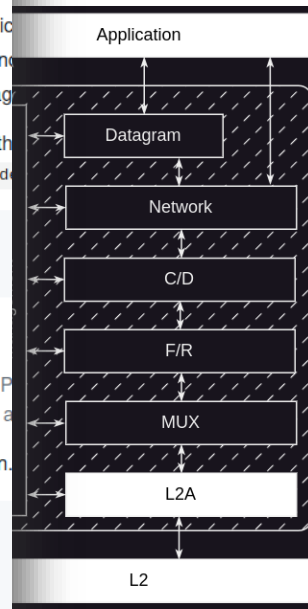
This example considers a UDP server application with the IP in the SCHC Gateway with the IPv6 address `5454::2` and a

1. Build and flash the FullSDK `ATModem` example application.

```
cd full-sdk-delivery/  
# Build:  
cmake -S . -B ./build/ -DAPP_NAME=ATModem \  
-DAPP_VERSION=4.0.0 -DFULLSDK_VERSION=3.0.0 \  
-DTOOLCHAIN=gcc-arm-none -DTARGET=m4 -DPLATFORM=STM32 \  
-DL2_STACK=semtech -DNUCLEO_LORA_SHIELD= SX1272 \
```

## General architecture

SCHC FullSDK follows an **event-driven** and **layered-architecture**. The diagram below depicts the overall architecture of the lab.SCHC FullSD





# lab.SCHC FullSDK

- Building & Flashing

- CMake
- Openocd
- GNU ARM Toolchain  
+ LoRa Semtech stack source
- fpv gcc Example output:

| FILE          | VEC | FLASH | RAM  | TOTAL |
|---------------|-----|-------|------|-------|
| libfullsdk.a  |     | 26848 | 2884 | 29732 |
| LoRaMac.c.obj |     | 11532 | 1716 | 13248 |
| libgcc.a      |     | 9880  |      | 9880  |
| command.c.obj |     | 4279  | 2588 | 6867  |



```
cmake -S . -B ./build  
-DAPP_NAME=<app_name>  
-DTOOLCHAIN=<toolchain>  
-DTARGET=<target>  
-DL2_STACK=<l2_stack> && make -C ./build
```

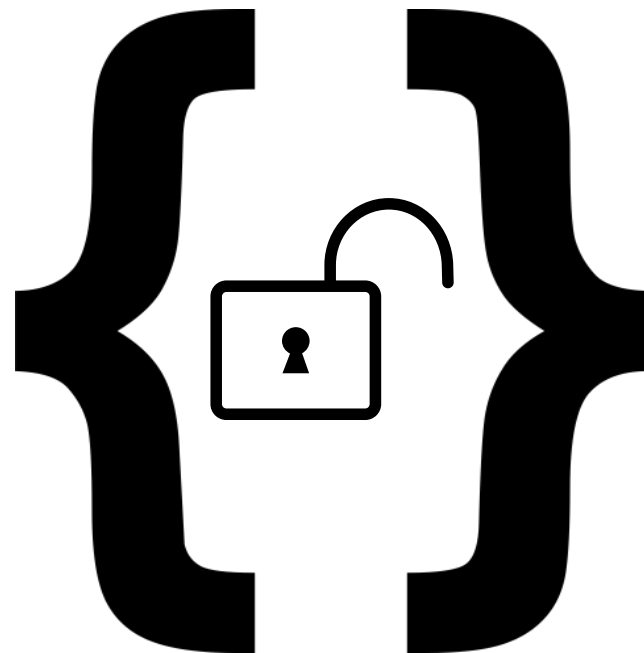


```
OPENOCD_TARGET=<openocd.cfg>  
BIN_FILE=build/<toolchain>/<target>/  
<app_name>.bin make -C openocd/ flash
```

# lab.SCHC

- Conclusion

1. Go to [lab-schc.fr](https://lab-schc.fr)
2. Clone the repo.
3. Look around & try it out.
4. Email us at [contact@lab-schc.fr](mailto:contact@lab-schc.fr)



FEC, OpenSCHC Frag  
Interoperability and more.  
Coming up!

## DEMO?