

Outlook and Invitation to Contribute

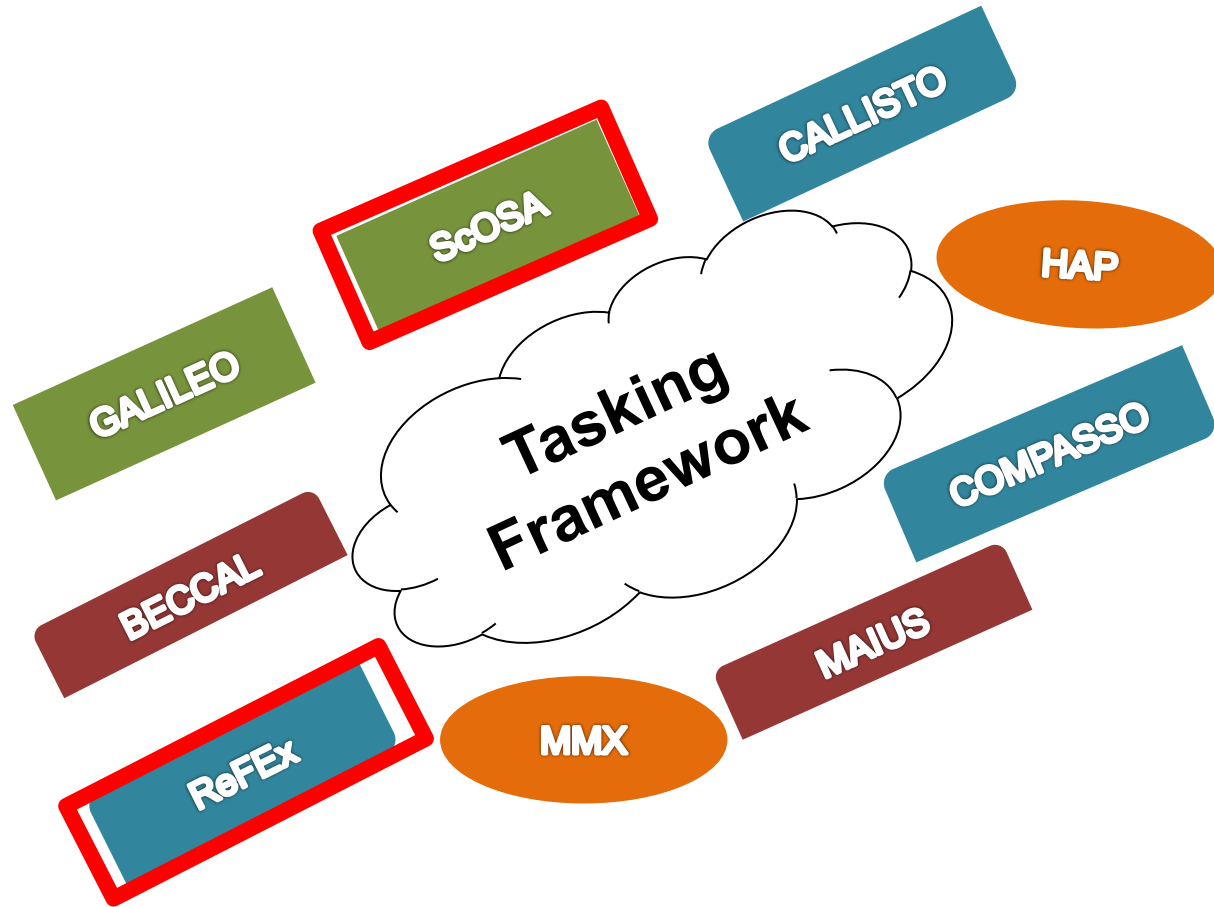
Embedded System Week 2020
Zain Hammadeh



Knowledge for Tomorrow



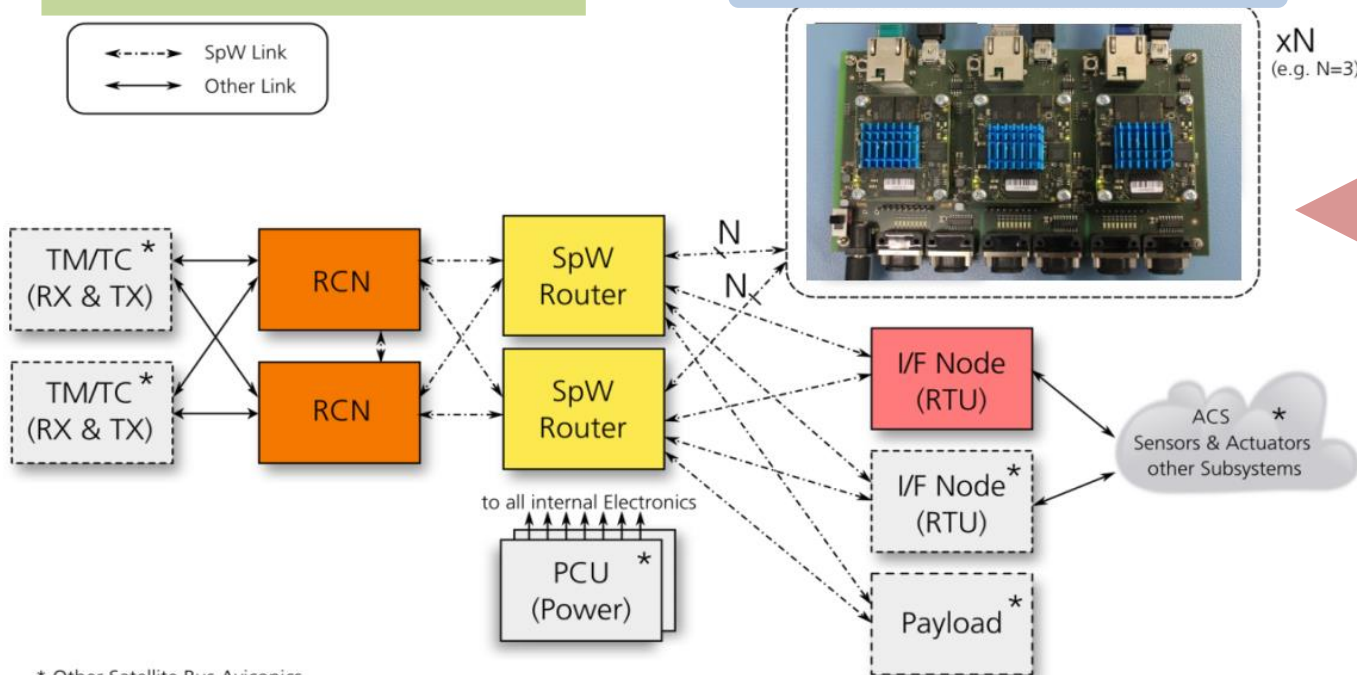
Where are we using Tasking Framework currently?



Scalable On-board Computing for Space Avionics (ScOSA)

High computational demand

Commercial off-the-shelf HPN



Reconfigurable Distributed System

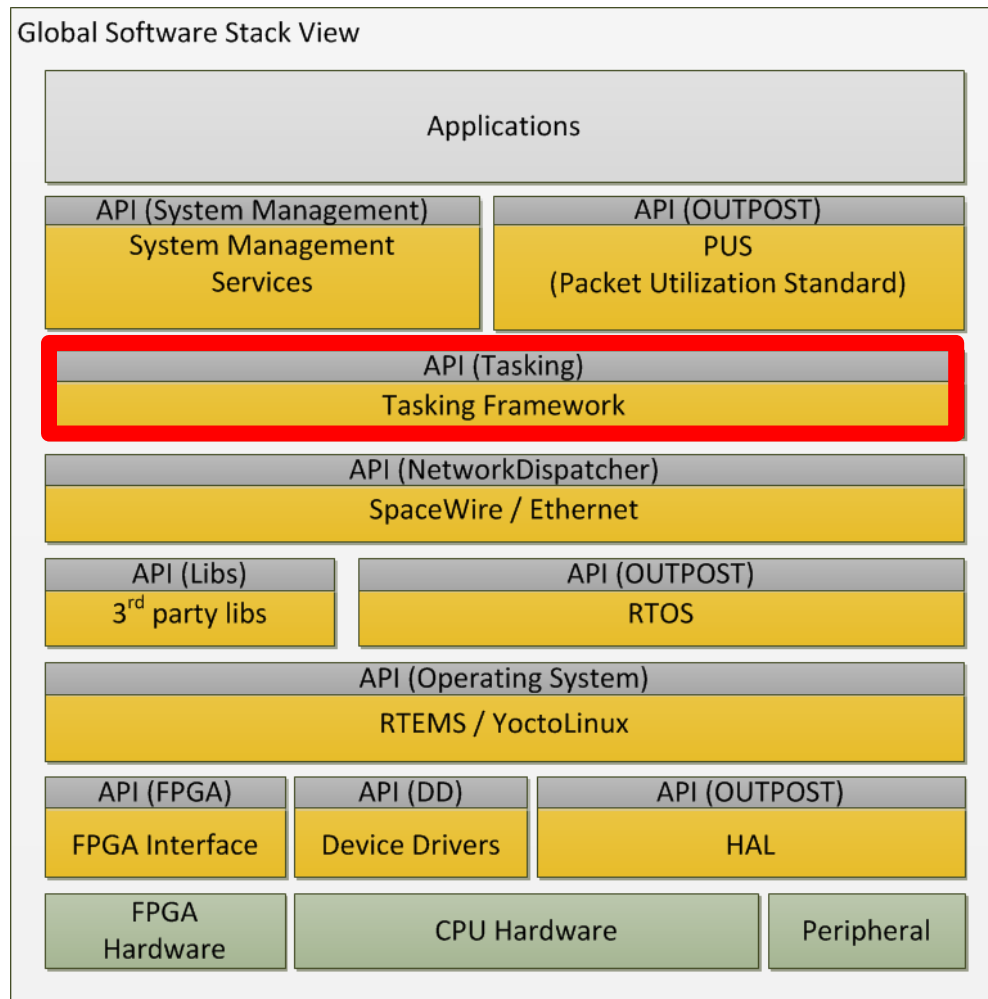
* Other Satellite Bus Avionics

RCN: Reliable Computer Node
 HPN: High Performance Node
 I/F Node: Interface Node
 SpW: SpaceWire
 RTU: Remote Terminal Unit
 PCU: Power Converter Unit
 TM/TC: Telemetry/Telecommand
 ACS: Attitude Control System

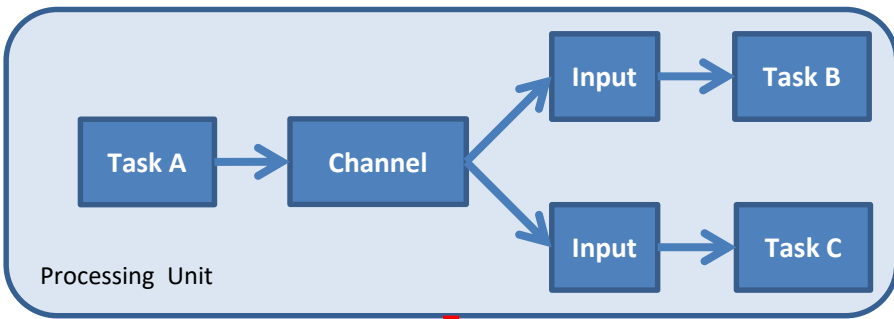
Höflinger – ASTRA 2017

ScOSA – Software stack

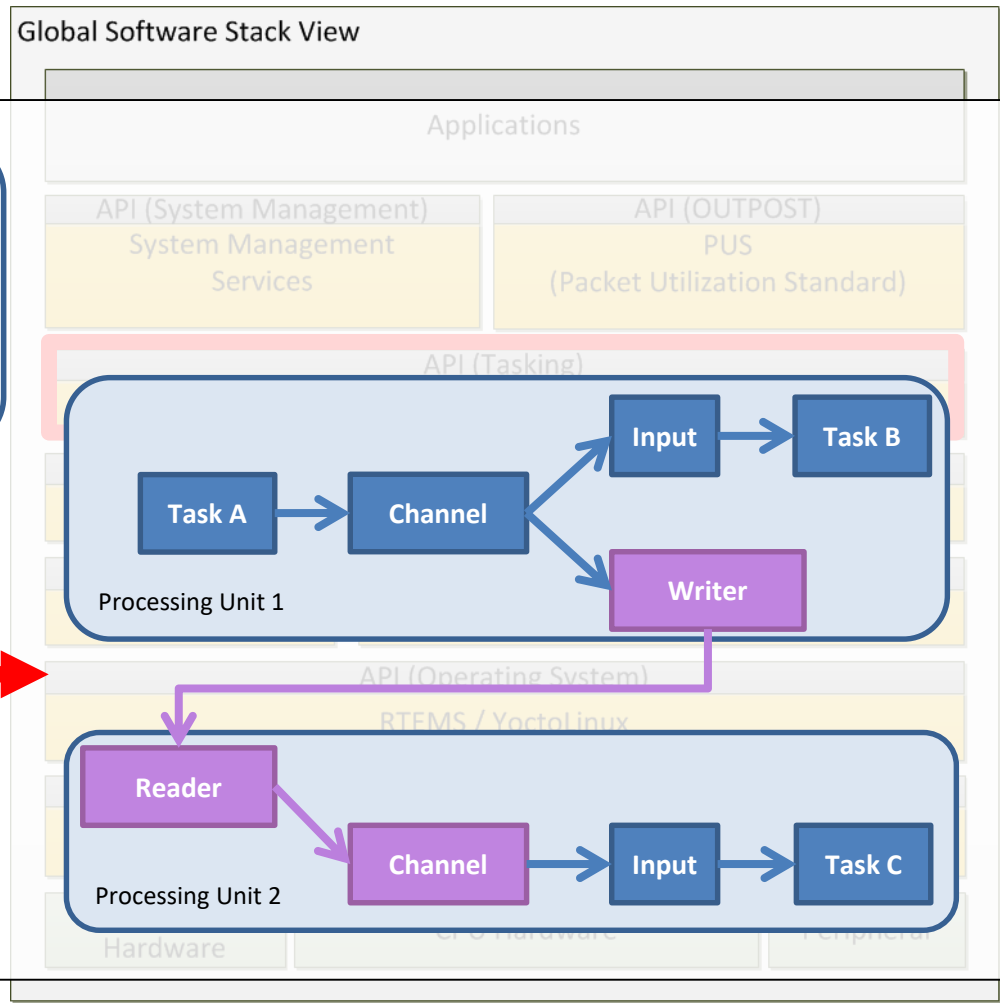
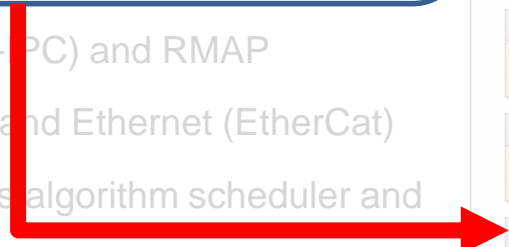
- Used hardware: FPGAs, ASICs, SoCs
- Multi-OS capable system
- FPGA as co-processor
- Reliable process communication network protocol (SpaceWire-IPC) and RMAP
- Supports Spacewire and Ethernet (EtherCat)
- Tasking framework as algorithm scheduler and communication manager
- System management and PUS services
- Application developer friendly API



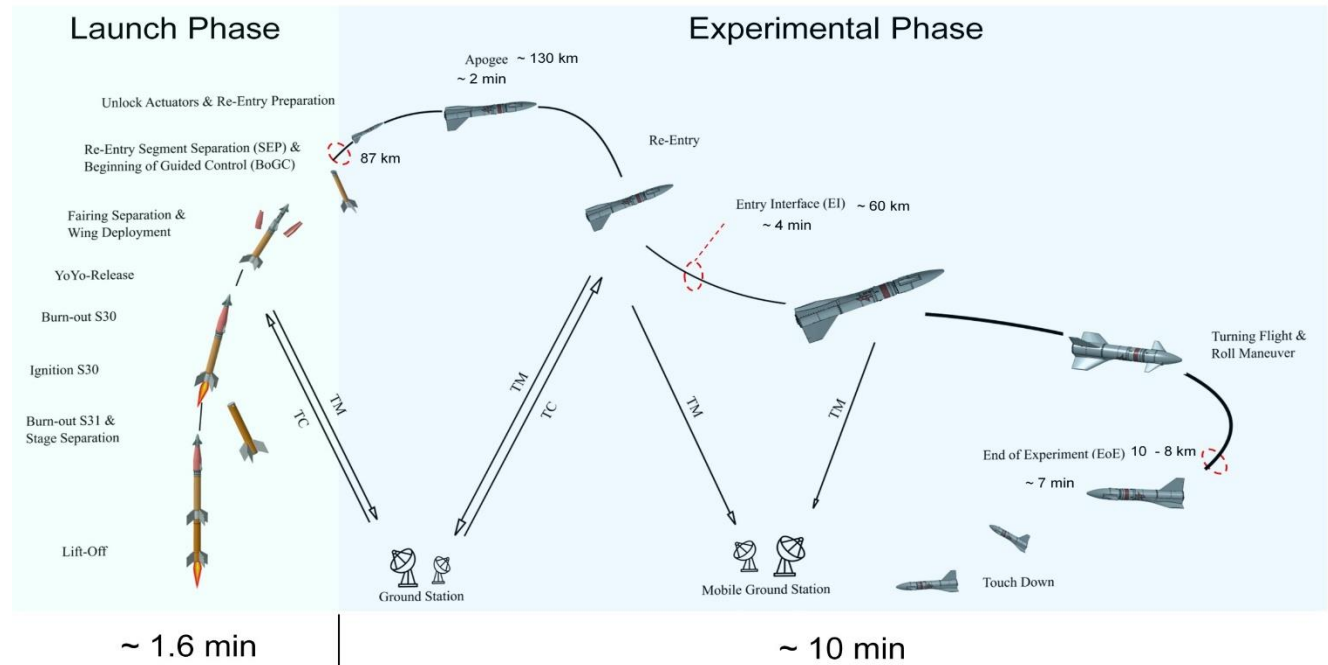
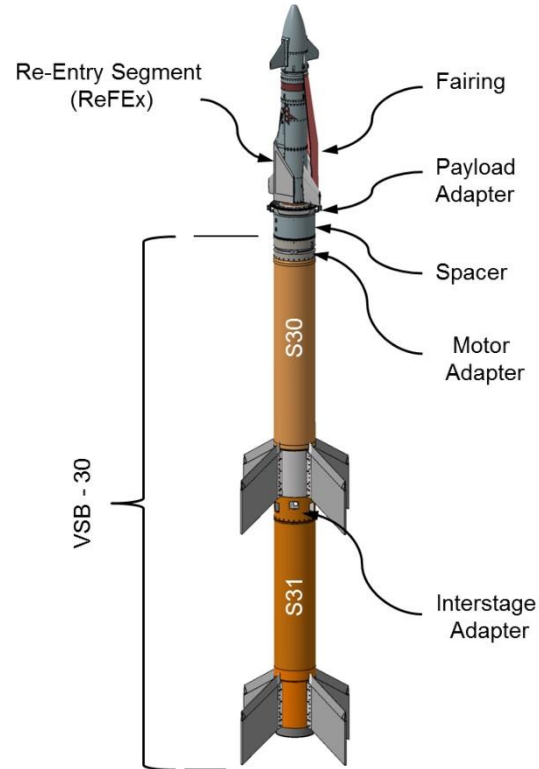
ScOSA – Software stack



- protocol (SpaceWire-PC) and RMAP
- Supports Spacewire and Ethernet (EtherCat)
- Tasking framework as algorithm scheduler and communication manager
- System management and PUS services
- Application developer friendly API

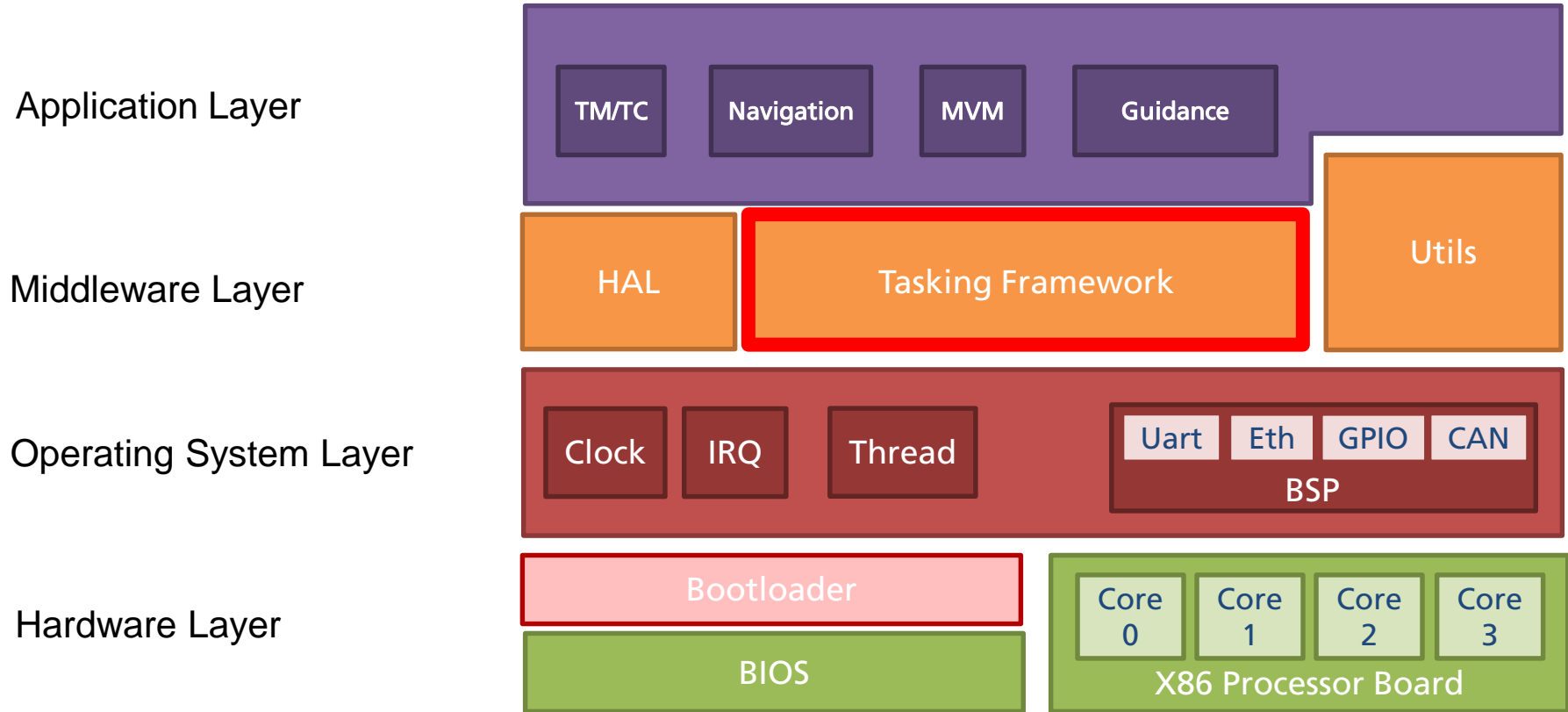


Reusability Flight Experiment (ReFEx)



<https://elib.dlr.de/131099/1/IAC-19.D2.6.9x49695.pdf>

ReFEx – Software stack



Outlook on Tasking Framework's Roadmap

C++11 Platform for Scheduler

TaskProvider as a C++11 tuple class for better type safety

Executor Stealing from other Scheduler

More channel types than only FIFO channel

User defined Boolean expression for InputArray to express more complex activation conditions



Potential research

Timing analysis (WCET + WCRT)

Multicore support (partitioning strategies)

Support for GPU tasks

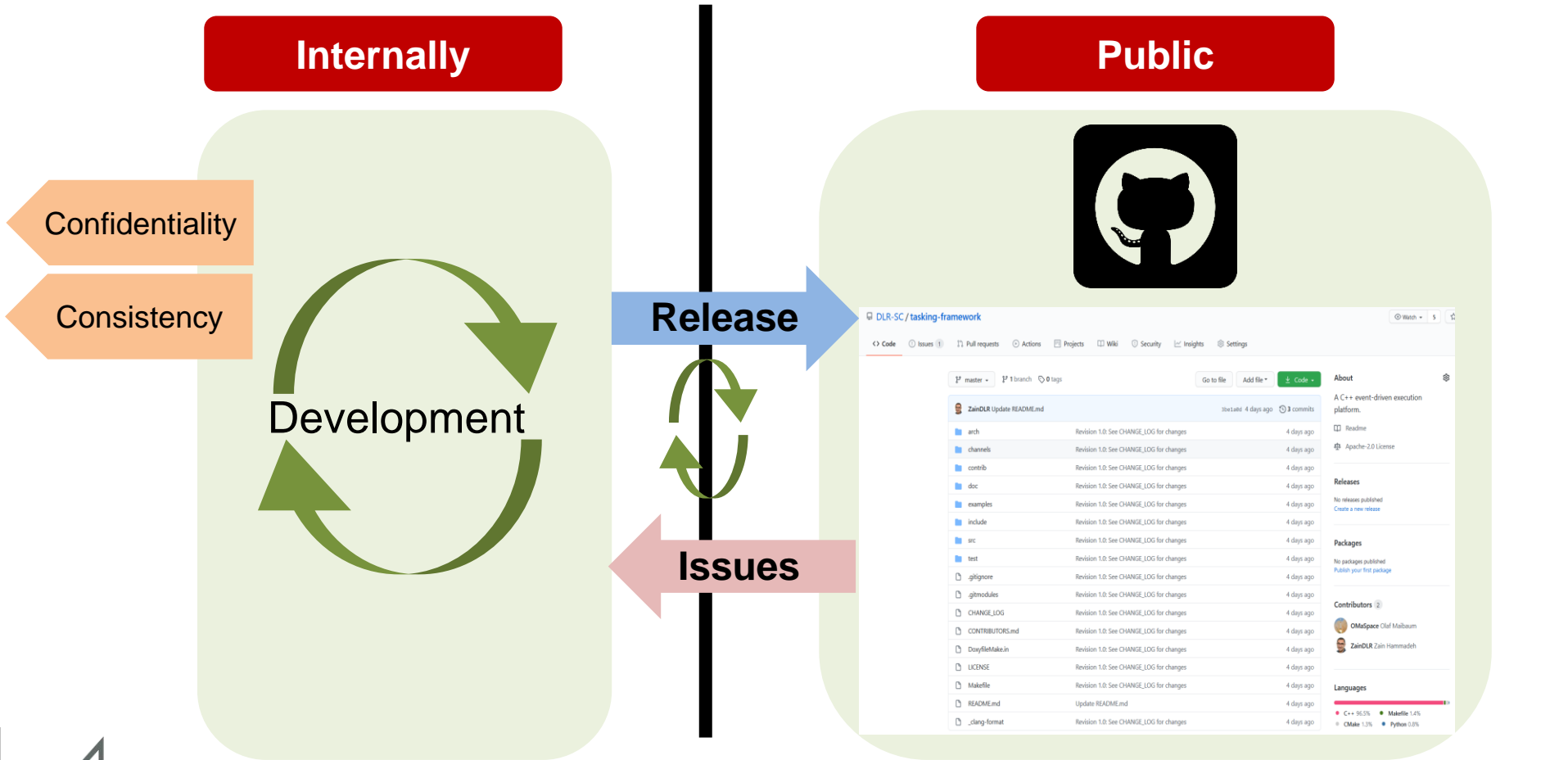
Bare-metal platform

Scheduling policy shortest laxity first

Alternative execution model for high workload



Development process



Development process

Internally

Public

Confidentiality

Consistency

- You can report bugs, suggest bug fixes, suggest features, etc.
- In case of contribution:
 - your contribution will be tested and evaluated
 - your name will be added to the contributor list (CONTRIBUTORS.md)
 - and to the related commit.

Issues



The screenshot shows a GitHub repository page for 'OMASpace'. The file list includes:

File	Commit Message	Time
src	Revision 1.0: See CHANGE_LOG for changes	4 days ago
test	Revision 1.0: See CHANGE_LOG for changes	4 days ago
ghignore	Revision 1.0: See CHANGE_LOG for changes	4 days ago
ghmodules	Revision 1.0: See CHANGE_LOG for changes	4 days ago
CHANGE_LOG	Revision 1.0: See CHANGE_LOG for changes	4 days ago
CONTRIBUTORS.md	Revision 1.0: See CHANGE_LOG for changes	4 days ago
DoxygenMake.in	Revision 1.0: See CHANGE_LOG for changes	4 days ago
LICENSE	Revision 1.0: See CHANGE_LOG for changes	4 days ago
Makefile	Revision 1.0: See CHANGE_LOG for changes	4 days ago
README.md	Update README.md	4 days ago
_clang-format	Revision 1.0: See CHANGE_LOG for changes	4 days ago

The right sidebar shows repository metadata:

- 3 commits (4 days ago)
- 4 days ago
- 4 days ago
- 4 days ago
- 4 days ago
- 4 days ago
- 4 days ago
- 4 days ago

Repository details:

- About: A C++ event-driven execution platform.
- Readme
- Apache-2.0 License
- Releases: No releases published. Create a new release.
- Packages: No packages published. Publish your first package.
- Contributors: OMA@Space Claf Malbaum, ZainDLR Zain Hammadeh.
- Languages: C++ 95.2%, Makefile 1.4%, CMake 1.3%, Python 0.0%.

How to cite Tasking Framework?

- You can cite our paper
 - OSPERT 2019**

On-line

- <https://ospert19.tudos.org/ospert19-proceedings.pdf>
- <https://elib.dlr.de/128249/>

Plain

H. Hammadeh, Zain A. and Franz, Tobias and Maibaum, Olaf and Gerndt, Andreas and Lüttke, Daniel (2019) ***Event-Driven Multithreading Execution Platform for Real-Time On-Board Software Systems***. In: Proceedings of the 15th annual workshop on Operating Systems Platforms for Embedded Real-time Applications, pp. 29-34. 15th annual workshop on Operating Systems Platforms for Embedded Real-Time applications, 09.07.2019, Stuttgart, Germany.

Bibtex

```
@InProceedings{Hammadeh2019,
  author = {Zain A. H. Hammadeh and Tobias Franz and Olaf Maibaum and Andreas Gerndt and Daniel Lüttke},
  booktitle = {15th annual workshop on Operating Systems Platforms for Embedded Real-Time applications},
  title = {Event-Driven Multithreading Execution Platform for Real-Time On-Board Software Systems},
  year = {2019},
  month = {Juli},
  pages = {29--34},
  url = {https://elib.dlr.de/128249/},}
```



Live Q&A session

Sunday 20.09.2020
at 11:00 – 12:00 am EDT
at 05:00-06:00 pm Berlin

11:00am

Tutorial 4: Tasking Framework: An open-source software development library for on-board software systems

11:00am - 12:00pm, Sep 20

Tutorials

* This tutorial is a **live Q&A session** with **pre-recorded videos made available before the event.**

Tasking Framework is a C++ software development library and an event-driven multithreading execution platform. It is developed by the Institute for Software Technology, German Aerospace Center (DLR). Tasking Framework is dedicated to improve the reusability in developing embedded software systems and to reconcile the embedded software with model-driven software development. It can be used to develop, but not dedicated for, critical as well as non-critical embedded software on single-core as well as parallel architectures. Tasking Framework gives software developers the ability to implement their applications as task graphs with arbitrary activation patterns (periodic, aperiodic and sporadic) using a set of abstract classes with virtual methods. It is compatible with the POSIX-based operating systems, mainly Linux and RTEMS. The Tasking Framework was successfully used in, for instance, the attitude orbit

[Read more](#)

