

<https://gitlab.cern.ch/caribou>

# Updates on Peary

**Simon Spannagel, DESY**

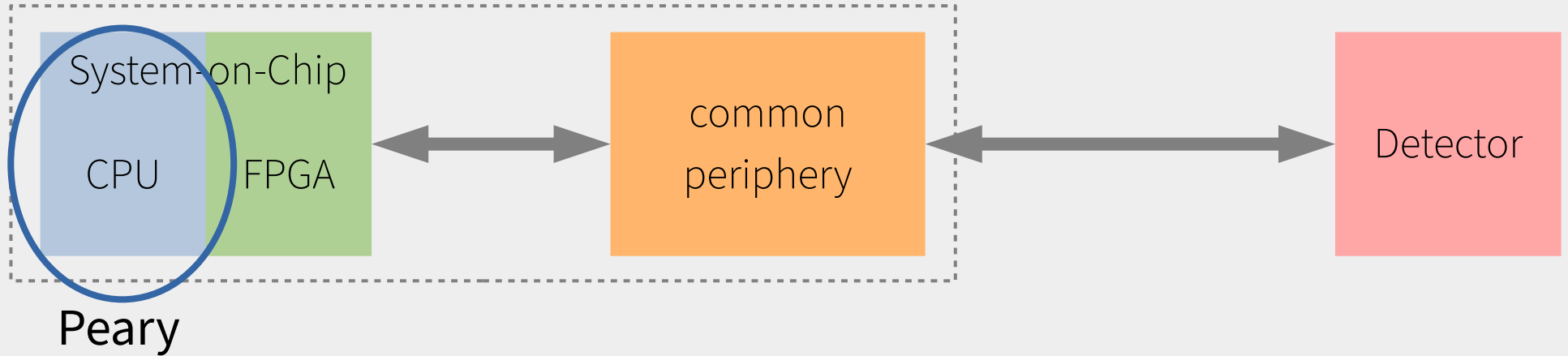
Caribou User Meeting

28 May 2021

Virtual

# Overview: Caribou

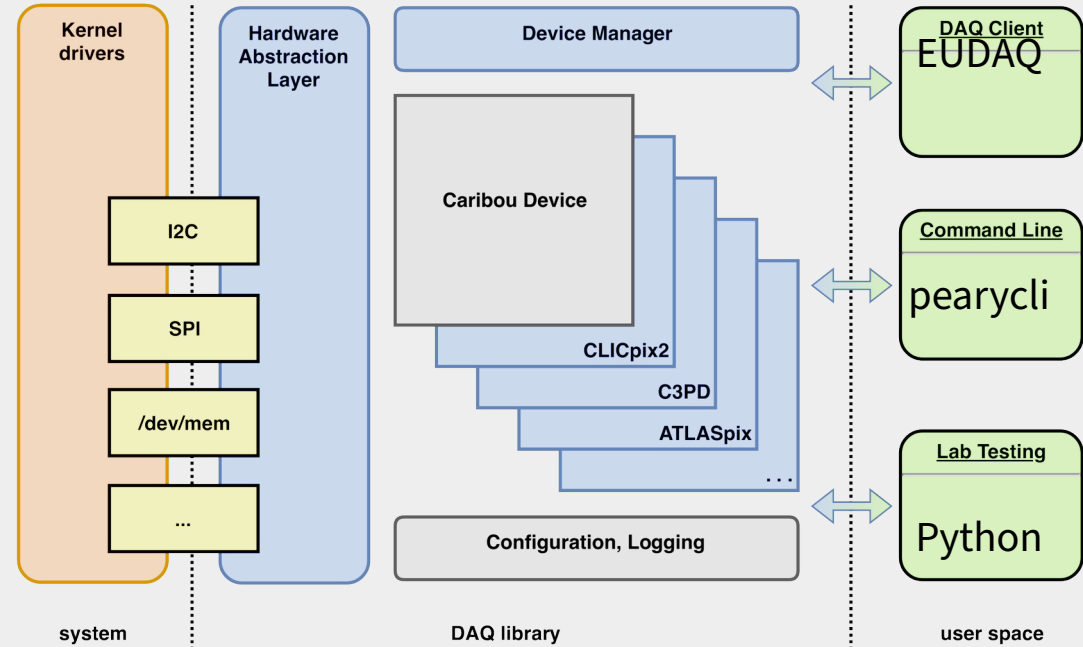
- Generic R&D requires flexible DAQ system, minimal effort for supporting new prototypes



- Using System-on-Chip devices to combine
  - Programmable logic (PL) – FPGA fabric for detector control, data handling
  - Processing system (PS) – CPU for data acquisition, user interface, full Linux system

# Overview: Peary

- Hardware Abstraction Layer (HAL) to handle peripherals
  - Functions to control periphery boards
  - Set/measure voltages, capture ADC, ...
- Device management
  - Multiple devices/detectors in parallel
- Various user interfaces:
  - Command line interface
  - Client interface for integration with a superior DAQ, system daemon
  - Python interface for scripting
  - EUDAQ2 producer: full integration of Caribou into EUDAQ2 test beam DAQ system



# Development on GitLab

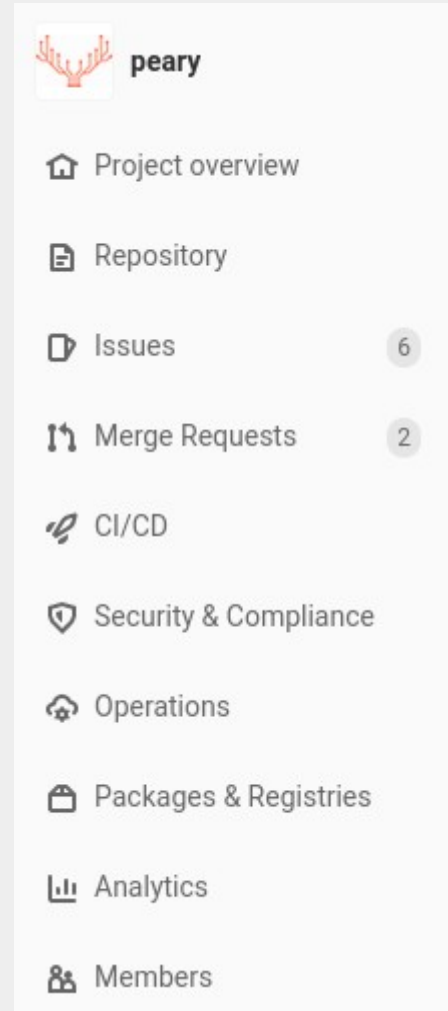
- Peary developed on CERN GitLab, alongside other Caribou projects

The screenshot displays the GitLab interface. On the left, the 'Caribou' group page is visible, showing a list of subgroups and projects: **firmware** (Owner), **hardware** (Owner), **meta-caribou** (Maintainer), **peary** (Maintainer), and **peary-firmware-modules**. The 'peary' project is highlighted. The main content area shows the 'peary' project overview, including statistics (1,912 Commits, 10 Branches, 30 Tags, 768 KB Files, 636.3 MB Storage) and a recent commit: 'Merge branch 'centos8' into 'master'' by Simon Spannagel. Below the commit, there are buttons for 'README', 'GNU LGPLv3', 'CI/CD configuration', 'Add CHANGELOG', 'Add CONTRIBUTING', and 'Add Kubernetes cluster'. A table at the bottom lists files and their last commit details.

Name	Last commit	Last update
<code>.gitlab-ci.d</code>	Rename variable	3 weeks ago
<code>cmake</code>	CMake: add compiler flags and version check	2 weeks ago

# Development on GitLab

- Repository with code
- Issue tracker
- Collaboration via forks/merge requests
  - Fork main repository (currently: 13 forks)
  - Push changes to your fork
  - Open merge request
- Container registry: Docker images



# Supporting Multiple DAQ Boards

- Peary as always been developed with flexible hardware support in mind
- Devices advertise which (periphery) hardware they run with
- Now first use cases:
  - Support for Carboard v1.4 (Eric Buschmann)
  - Support for Falconboard (Fastree3D, Adrian Fiergolski)  
completely separate project using Caribou platform but different board,  
Hardware development kit for development of flash LiDAR
- Groundwork for supporting Carboard v2.0 in the future



# Device Class Interface

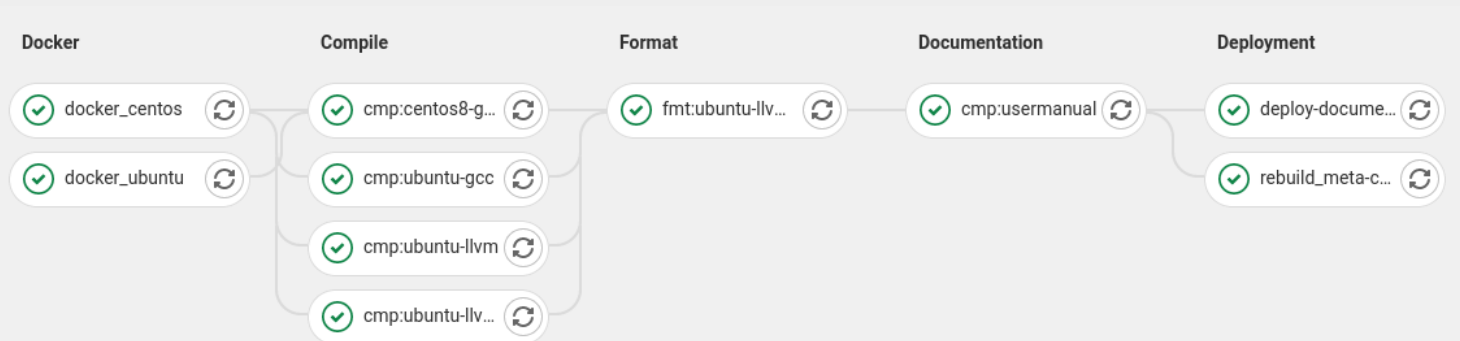
- Removed template complexity from hardware abstraction
- Now device classes inheriting from CaribouDevice specify as template parameters
  - The hardware they intend to run on/with
  - The slow-control interface they will be using
  - (in preparation: the data/readout interface from which output can be obtained)

```
17 namespace caribou {  
18  
19     /** CLICTD Device class definition  
20     */  
21     class CLICTDDevice : public CaribouDevice<carboard::Carboard, iface_i2c> {  
22  
23     public:
```



# Continuous Integration / Automated Builds

- CI ensures software can be build, additional formatting, linting, ...



- Peary: every merge request built & tested on Ubuntu 20.04 & CentOS 8
- Full Caribou Linux (including Peary) built automatically every night
  - SD card image generated, placed on EOS
  - Latest Caribou version always ready to download:

<https://project-meta-caribou.web.cern.ch/project-meta-caribou/sdimage-latest-mmcbk.zip>



# Summary & Plans

- Consolidation of Peary code ongoing
  - Reworked HAL inheritance, altered interface
  - Preparatory work for new hardware
- CI / automated builds running again, SD card images available for download
- Some more rework planned
  - Standardized data/readout interface access
  - Possibility to configure interfaces (different bus widths etc)
  - Cleanup of HAL methods, move some to specific boards
- Contributions from users welcome!

