

Calibration Software Package [pr-cal-3-4]

pmdtechnologies ag – 2019-05-16

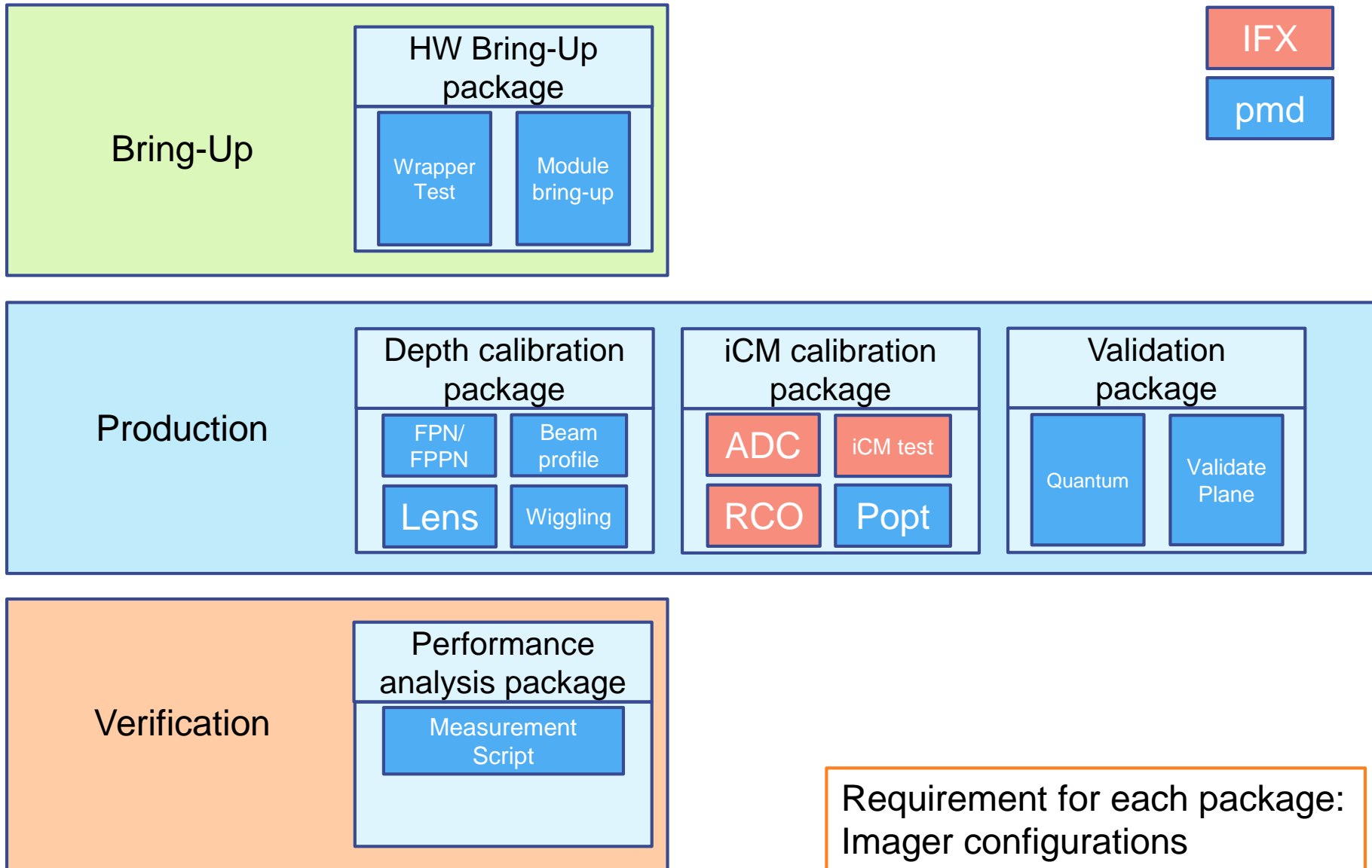
Outline

- Background information
- pmd provides a complete python software package, which consists of:
 - Hardware bring-up package
 - Depth calibration package
 - iCM calibration package
 - Validation package
 - Performance analysis package
- An API is provided for the frame grabber and the linear translation stage
- Explanations for implementation are given
- Roles and Responsibilities

Background

- Today situation:
 - pmd / IFX gives example software to explain the functionality of the calibration steps
 - The implementation is done at ODM side in building their own software scripts
 - pmd / IFX supports on implementation
- New situation:
 - pmd gives calibration package
 - ODM only needs to implement custom frame grabber functions
 - pmd / IFX supports on implementation
- Disadvantages:
 - Custom frame grabber functions need to be implemented into the wrapper
 - Wrapper of the software package needs to be used
- Advantages:
 - Custom frame grabber functions need to be implemented into the wrapper
 - Functions for frame grabber need to be implemented anyhow, only small efforts to be done at ODM
 - Flow and functions are written by pmd / IFX
 - No additional code writing necessary at ODM side
 - Bring-up of new modules
 - Much easier as basic ODM functions most probably stay the same
 - Error handling during production
 - pmd / IFX can support much better and faster as same software scripts are used
 - calibration package updates:
 - Update installers are provided by pmd
 - Only installation needs to be done at ODM side
 - No code writing necessary at ODM side (except Wrapper changes)

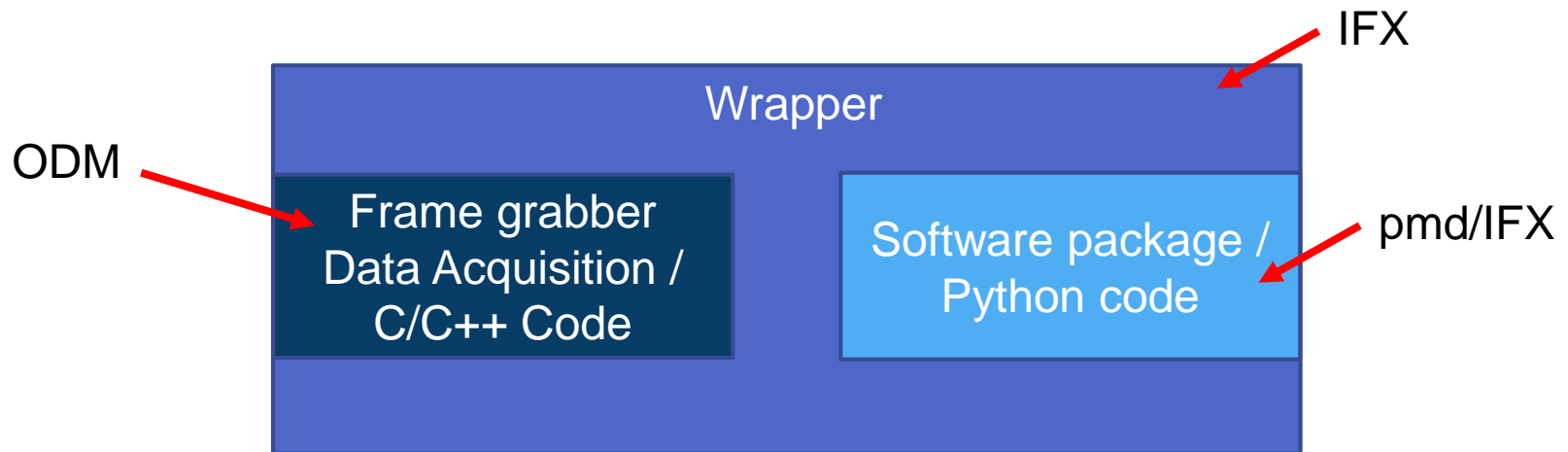
Overall software package



confidential

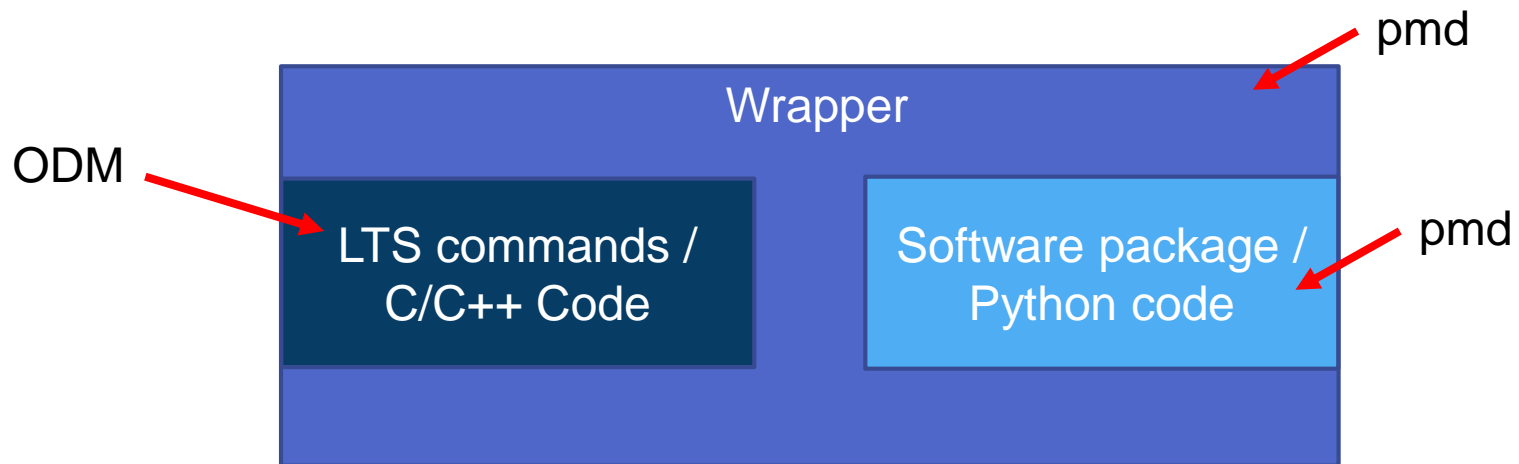
Frame grabber API

- Usage of animator wrapper
 - ODM to implement all required framegrabber functions in C/C++
 - Python software packages communicate with frame grabber via wrapper
- ODM frame grabber functions only have to be implemented once
- Update of software package does not result in ODM/IFX/pmd efforts



Linear translation stage (LTS) API

- Usage of custom LTS wrapper
 - ODM to implement all required linear translation stage functions in C/C++
 - Python software packages communicate with linear translation stage via wrapper
- ODM LTS functions only have to be implemented once
- Update of software package does not result in ODM/IFX/pmd efforts



Software usage

- Win 7 / Win 10 pc needs to be available
- Installer for calibration package is provided by pmd
 - Information about installer will be send additionally
- Installation of Newport power meter needs to be done (iCM calibration)
- Frame grabber wrapper needs to be implemented
 - please see slide 8
- LTS wrapper needs to be implemented
 - please see slide 9
- Example executable (.bat) files are included in the package
 - Information about execution of the different sub-packages can be found later in this document

confidential

Frame grabber implementation

- Software required by ODM: Compiler (C++) and SWIG
- Header files and Swig file can be found in the software package:
 - wrapper\animator*
- According to the header files, basic functions of the frame grabber need to be written and connected to the wrapper (open, i2cwrite, i2cread, spiwrite, spiread, getnextbuffer, ...)
- After running SWIG and compilation the following files should be generated⁽¹⁾:
 - _animator.pyd (generated by Compiler)
 - animator.py (generated by SWIG)
- These files and the additional linked frame grabber binaries need to be copied into one folder
- This folder can then be used in the config .ini files for bring-up, production and verification (frame_grabber_path = "path_to_this_folder")

(1) cmake, Visual Studio 2017 and build_wrapper.bat can do the compilation:

CMakeLists.txt

- SWIG_EXECUTABLE_DIR
- FRAMEGRABBER_INCLUDE_DIR
- FRAMEGRABBER_LINK_DIR

build_wrapper.bat

- Path to Visual Studio 2017
- Path to cmake.exe

Run build_wrapper.bat from VS command prompt

LTS implementation

- Software required by ODM: Compiler (C++) and SWIG
- Header files and Swig file can be found in the software package.
 - Scripts\subfunctions_ch\custom_lts\custom_lts_wrapper\
- According to the header files, basic functions of the LTS need to be written and connected to the wrapper (open, moveRef, moveAbs, getPosition, ...)
- After running SWIG and compilation the following files should be generated⁽¹⁾:
 - _Custom_LTS.pyd (generated by Compiler)
 - Custom_LTS.py (generated by SWIG)
- These files and the additional linked LTS binaries need to be copied into the following folder:
 - Scripts\subfunctions_ch\custom_lts\

(1) cmake, Visual Studio 2017 and build_wrapper.bat can do the compilation:

CMakeLists.txt

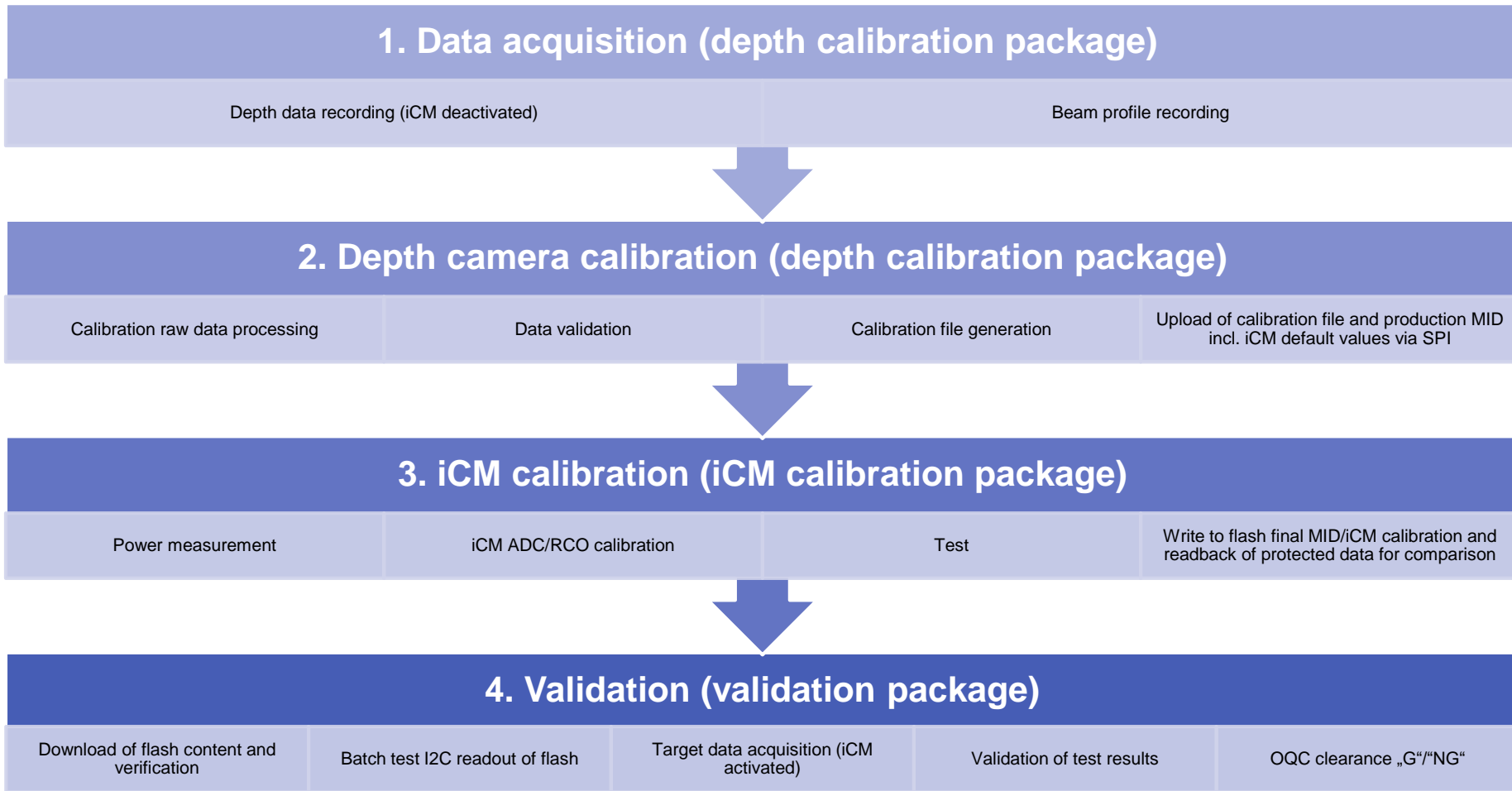
- SWIG_EXECUTABLE_DIR
- CUSTOM_LTS_INCLUDE_DIR
- CUSTOM_LTS_LINK_DIR

build_wrapper.bat

- Path to Visual Studio 2017
- Path to cmake.exe

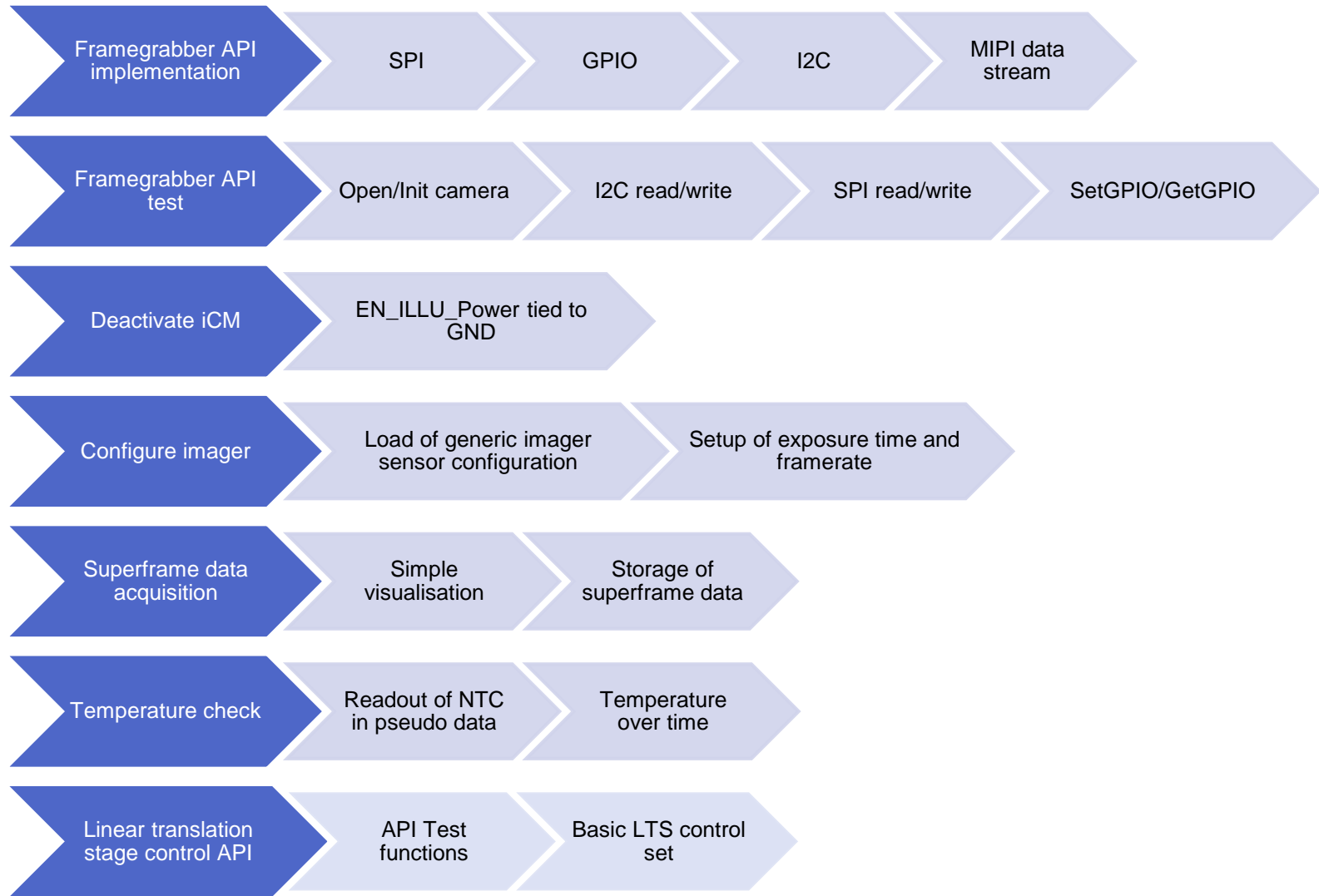
Run build_wrapper.bat from VS command prompt

Production line flow overview



Comment: The readback during iCM calibration happens after Final MID flash. This means that the module can be fail but has already the final MID. The OQC clearance has to take care of this situation.

Hardware bring-up steps



Hardware bring-up package (1)

- Frame grabber API test

- Update and run .bat:

framegrabber_test_run.bat

```
--frame_grabber_path ""  
--outdir ""
```

- Output (in outdir):

Framegrabber_test.log

Pass/Fail for each test

Hardware bring-up package (2)

- Superframe data acquisition and temperature check

- Update and run .bat:

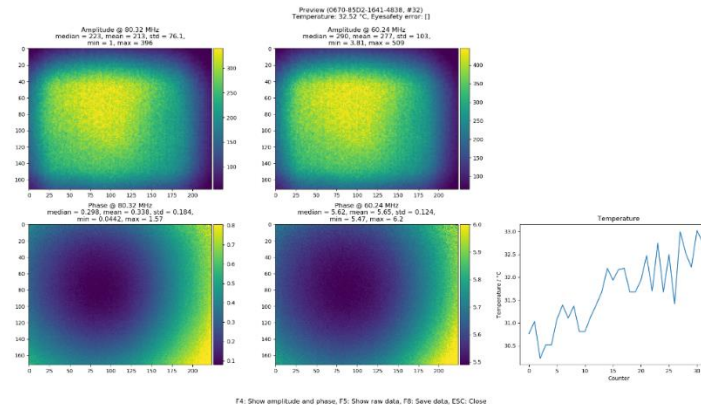
any_quick_view.ini

Parameter file (additional Docu)

Quick_view_run.bat

--config_file "path to any_quick_view.ini"

- A window should appear showing the measured amplitude, phase and temperature:



- Press F8 to save data to the OutBaseDir given in the .ini file

0000-0000-0000-0000_VIEW.png

PNG Picture

0000-0000-0000-0000_VIEW.pickle

Picture in Pickle format for Python import

0000-0000-0000-0000_VIEW.rds

Raw data

0000-0000-0000-0000_VIEW_temperatures.npy

Python array with temperature values

- Closing the window will close the application and store:

log.log

Log information

Hardware bring-up package (3)

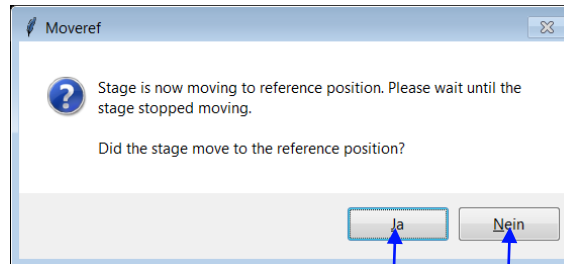
- Linear translation stage API test

- Update and run .bat:

LTS_API_test_run.bat

```
--comport "COMX"  
--minpos x.y  
--maxpos x.y  
--outdir ""
```

- Several windows will appear for user interaction to check if the LTS is doing as expected:



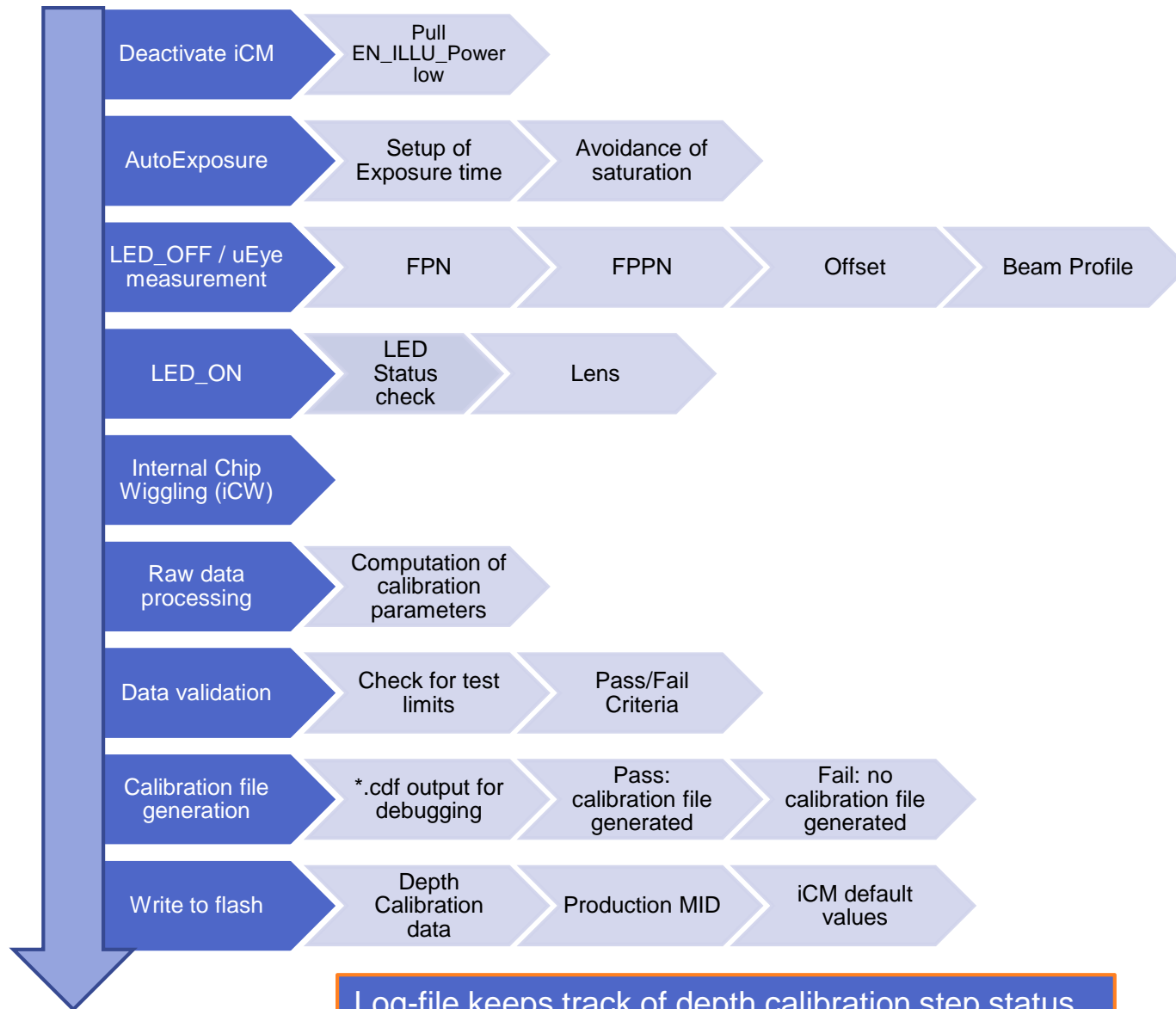
Please click: Yes or No

- Output (in outdir):

LTS_API_test.log

Pass/Fail for each test

Depth calibration steps



confidential

Depth calibration package

- Update and run .bat:

any_LED.ini

Parameter file (additional Docu)

calib_config.ini

Parameter and test file (additional Docu)

pmd_depth_calibration_run.bat

--config_file "path to any_LED.ini"

- Output in folder 0000-0000-0000-0000_LED :

Phase_01_LEDsON.rds

Raw data with LED on

Phase_02_LEDsOFF.rds

Raw data with LED off

Phase_03_iCW.rds

Raw data for iCW

ueye.npy

Ueye image in python file

any_LED.ini

Ini file used for calibration

log.log

Log file

calib_out

0000-0000-0000-0000.zwetschge

Calibration data written on flash

0000-0000-0000-0000.cdf

Calibration data for debugging

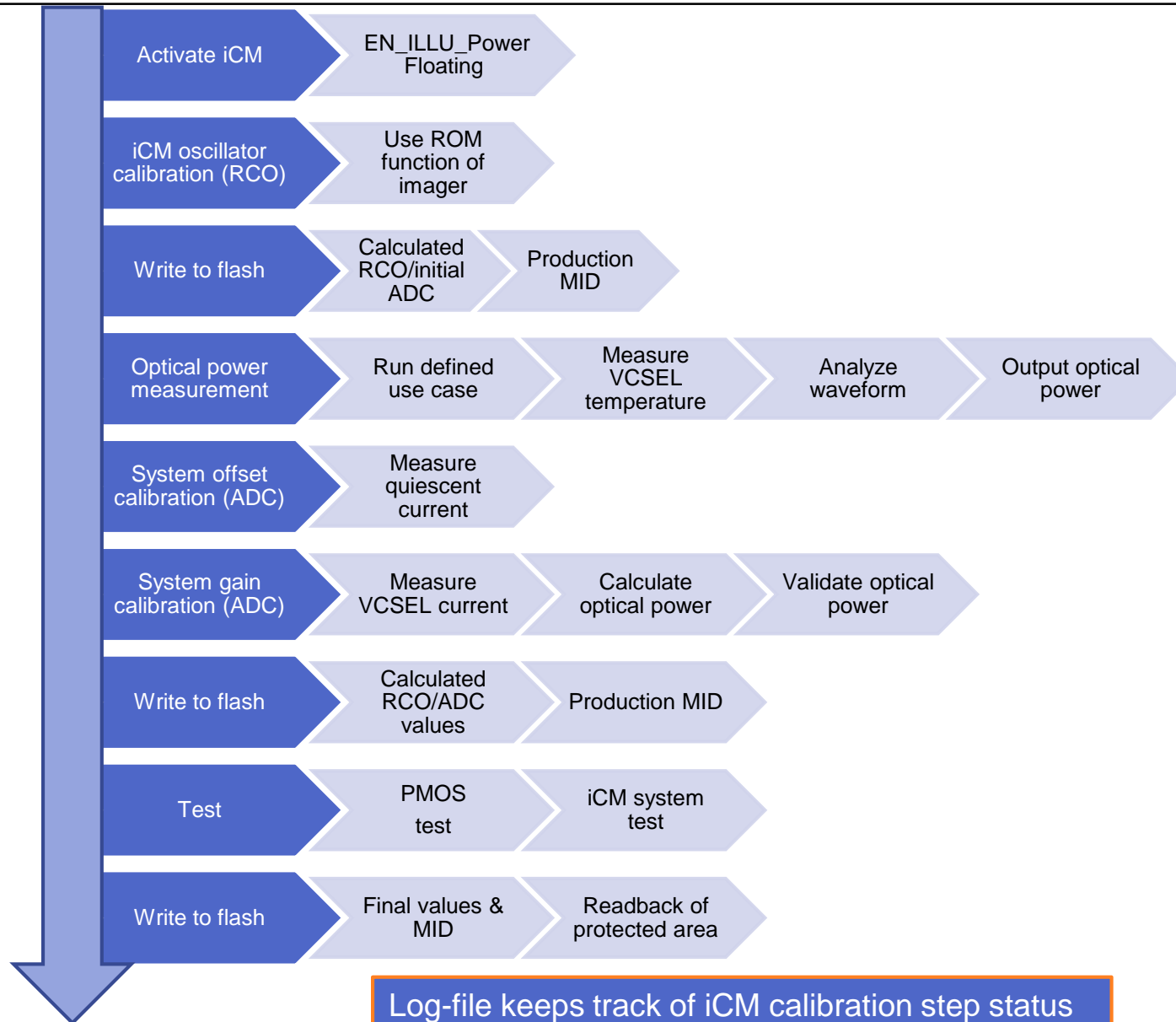
log_out

0000-0000-0000-0000.db

Log file with Test results

- If successful the calibration will be directly written on the module flash
- If not successful, no data will be written on the module flash and an error code (3) will be given as feedback

iCM calibration steps



confidential

iCM calibration package

- Update and run .bat:

any_Sphere.ini

Parameter file (additional Docu)

icm_calib_run.bat

--config_file "path to any_Sphere.ini"

- Output in folder 0000-0000-0000-0000_Sphere:

0000-0000-0000-0000...iCM_calib.log

Log file

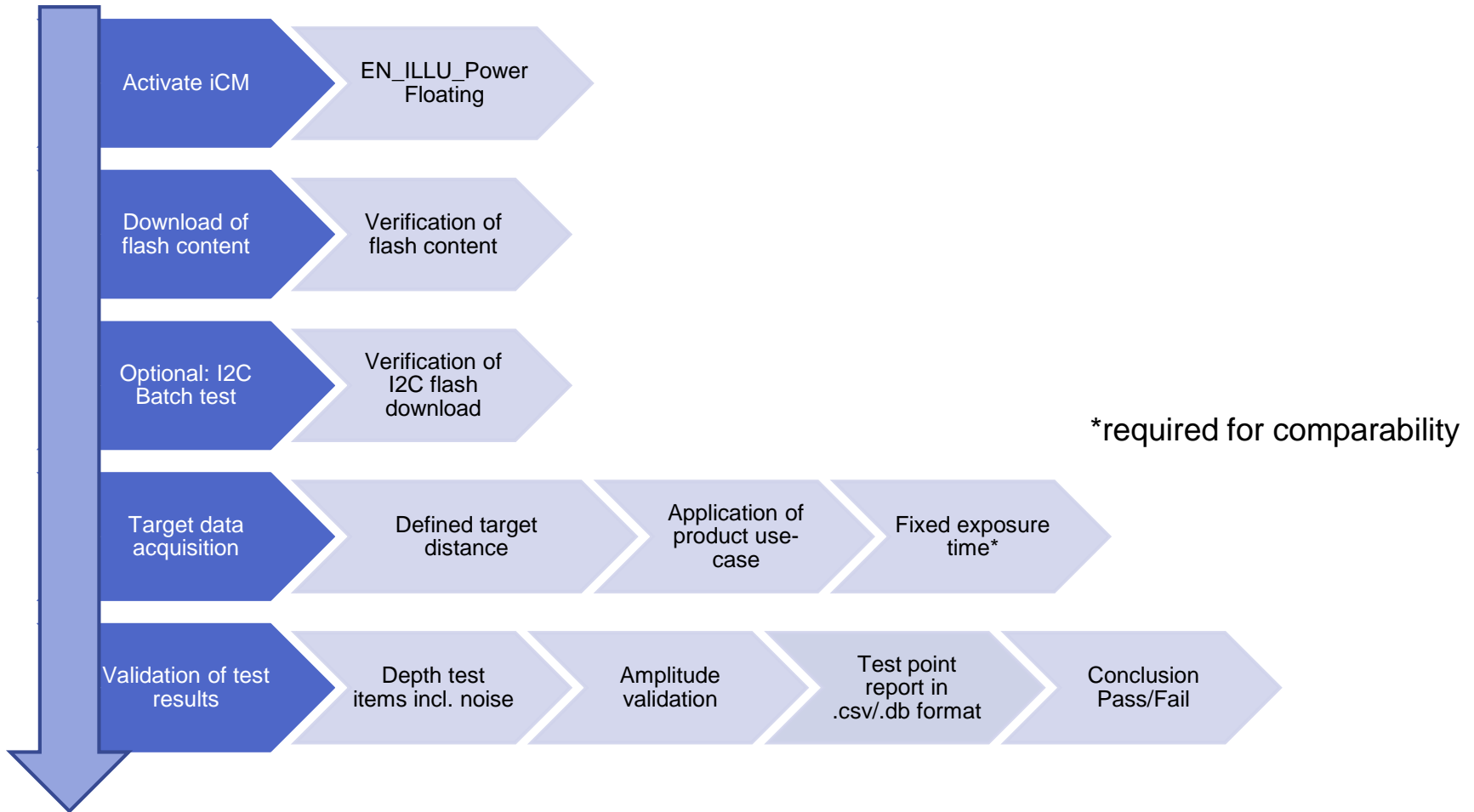
samples.txt

raw data power meter waveform

samples.npy

raw data power meter waveform python

Validation steps



Log-file keeps track of validation step status

Validation package

- Update and run .bat:

any_VAL.ini

Parameter file (additional Docu)

processing_config.json

Processing and Limits File (additional Docu)

pmd_depth_calibration_run.bat

--config_file "path to any_VAL.ini"

- Output in folder 0000-0000-0000-0000_VAL:

Phase_01_VAL.rds

Raw data

log.log

Log file

any_VAL.ini

.ini file used for validation

val_out

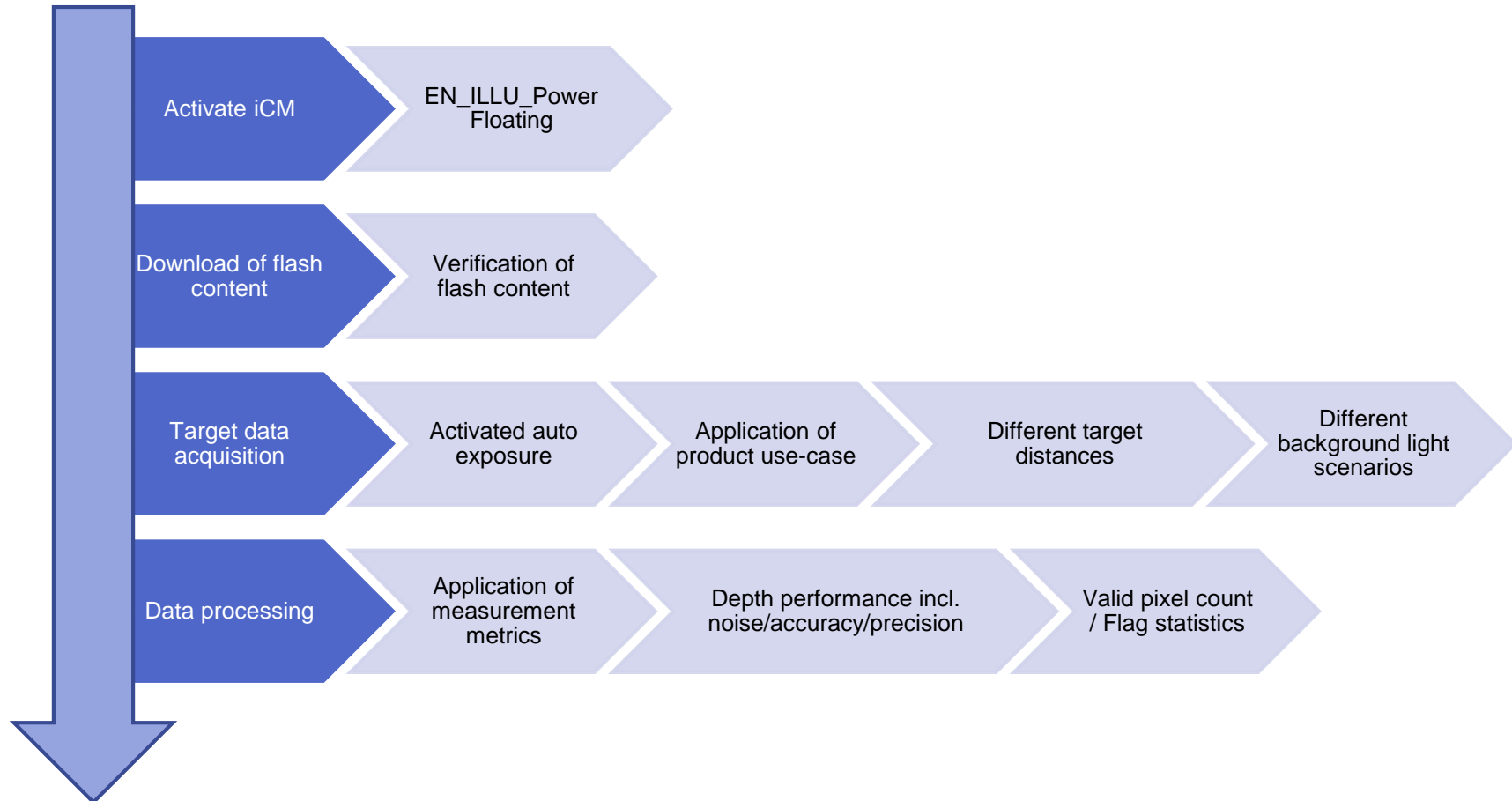
0000-0000-0000-0000....db

Validation test report

- If one of the tests fails, an error code (3) will be given back

confidential

Performance analysis step overview



Log-file keeps track of performance step status

Performance analysis package

- Update and run .bat:

any_LTS.ini

Parameter file (additional Docu)

processing_config.json

Processing File (additional Docu)

pmd_depth_calibration_run.bat

--config_file "path to any_LTS.ini"

- Output in folder 0000-0000-0000-0000_LTS:

Phase_01_LTS_XXXcm.rds

Raw data for each distance

log.log

Log file

any_LTS.ini

.ini file used for analysis

cdd_out

cdd_res_0000-0000-0000-0000.xlsx

Performance report

plane_dev_ply_0000-0000-0000-0000_X.X.ply

Plane deviation for each distance

confidential

Roles & Responsibilities with regard to work with the calibration software package

- pmd:
 - Provides software package
 - Supports on software package implementation
 - Supports on linear translation stage wrapper
 - Supports on depth calibration, validation and performance analysis topics
- IFX:
 - Supports on frame grabber wrapper implementation
 - Provides imager configurations
 - Supports on iCM calibration topics
- ODM:
 - Provides PC with Win 7 / 10
 - Implements software package into production line
 - Implements C++ frame grabber wrapper
 - Implements C++ linear translation stage wrapper
 - Installation of Newport power meter software

confidential

Roles and Responsibilities

- **pmdtechnologies ag** does support in the design process and provides consulting support in form of design and specification reviews. **pmdtechnologies ag** is not responsible for the final product of the customer. Reference designs are recommendations and may not be suitable for a claim of completeness.

Document History

- Document: Performance Evaluation [pr-se-3-2]

Revision	Origin of Change	Submission Date	Description of Change
0	MRe	2019-03-18	New presentation
1	MRe	2019-05-16	Updates with regard to wrapper changes
2	SSo	2019-05-16	Adding of Roles and Responsibility and Document History

confidential

© **pmd**technologies ag, 2019

Mailing Address: The SUMMIT, **pmd**technologies ag, Martinshardt 19, 57074 Siegen, Germany

Technical information subject to change without notice.

This document may also be changed without notice.

All texts, pictures and other contents published in this application note are subject to the copyright of **pmd**technologies ag, Siegen unless otherwise noticed. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher **pmd**technologies ag.