EPICS Collaboration Meeting June 2019

Contribution to Nominal Device Support V3 for standardizing device drivers

Moreno, Javier[1] (*jmoreno@gmv.com*); Ruíz, Mariano[2]; Melis, Stefano[1]; de Gracia, Alberto[2]; Sanz, Diego[1]; Esquembri, Sergio[2]; Astrain, Miguel[2]; Lange, Ralph[3]
[1] GMV
[2] UPM. Technical University of Madrid, Instrumentation and Applied Acoustic Research Group
[3] ITER International Organization

iter
china eu india japan korea russia usa

GRUPO DE INVESTIGACIÓN EN INSTRUMENTACIÓN Y ACÚSTICA APLICADA

POLITÉCNICA

gmv
INNOVATING SOLUTIONS

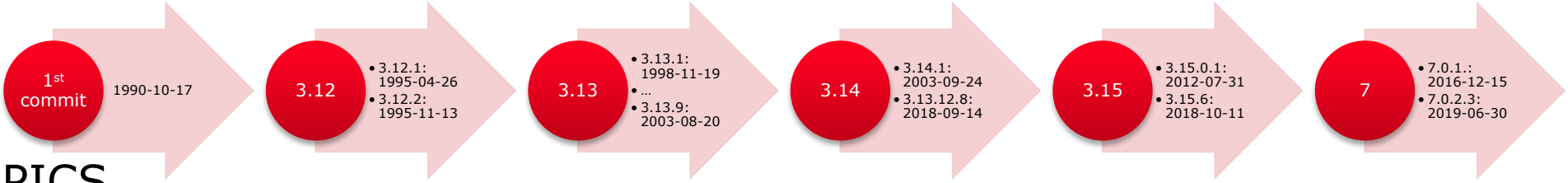
INDEX

- Motivations
- NDS overview
- Contribution to NDS3
- Use cases
- Conclusions and references

NDS

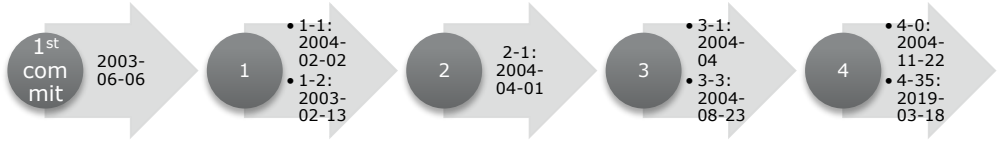
Motivations

MOTIVATIONS

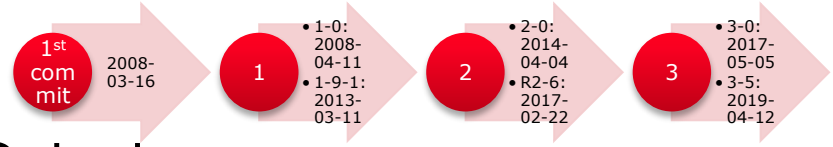


EPICS

- The higher the use of EPICS, the more support layers appear to ease IOC development
- Standardization claims to minimize IOC development efforts
- *asyn* drivers are typically more friendly than just *device support*
- *areaDetector* is based on *asyn* to interface with detectors and cameras, as well as devices providing waveforms
- *NDS* is a generic solution for device driver development



asyn



areaDetector

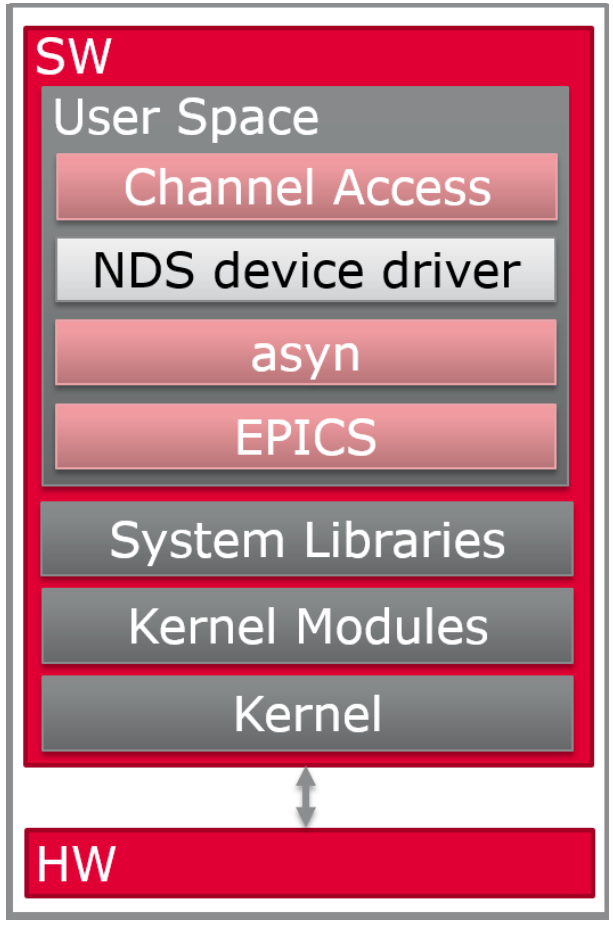
NDS

NDS

Overview

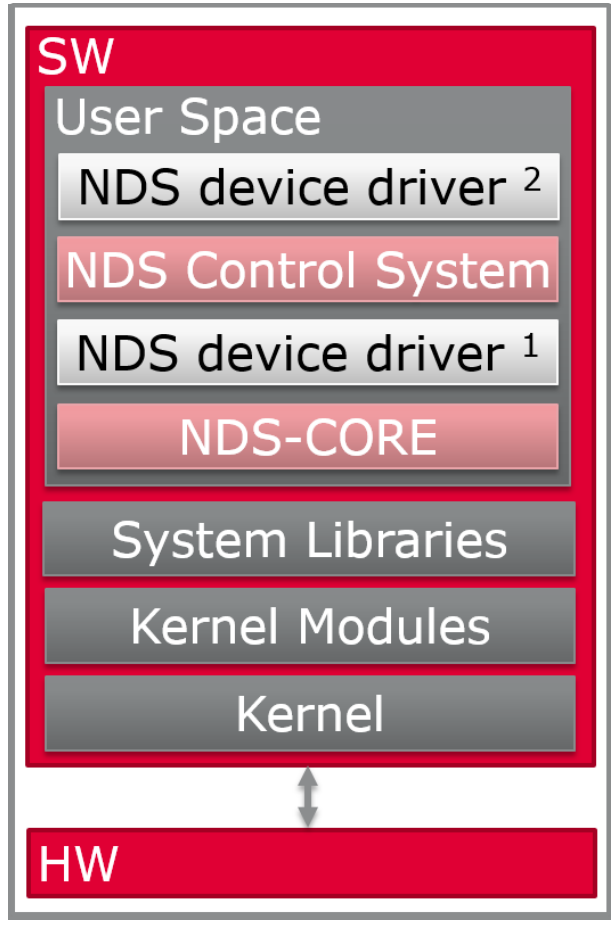
NOMINAL DEVICE SUPPORT

- NDS introduced the Device Model
- Based on *asyn* to provide common interface to EPICS IOCs
- Device drivers based on:
  - Device
    - State machine
  - ChannelGroup
  - Channel
    - Image
    - Analog/Digital Input/Output
- EPICS Libraries developed in C++



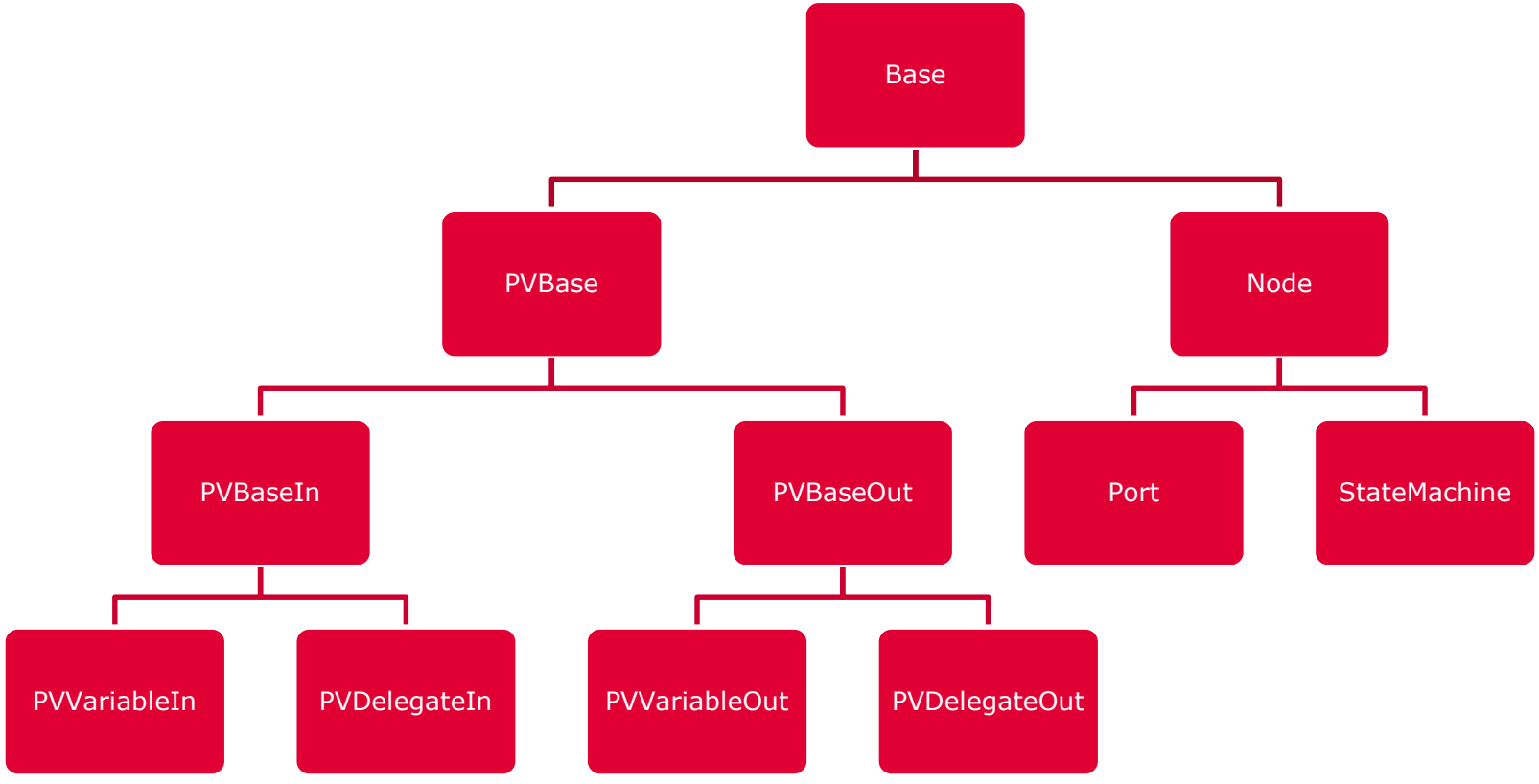
NOMINAL DEVICE SUPPORT V3

- NDS3 drivers do not depend on EPICS*
- Based on its own Control System (*Factory*) with a common interface to IOCs
- Device drivers based on:
  - Node
    - Port
    - State machine
  - PVs
    - Input
    - Output
- C++11 libraries



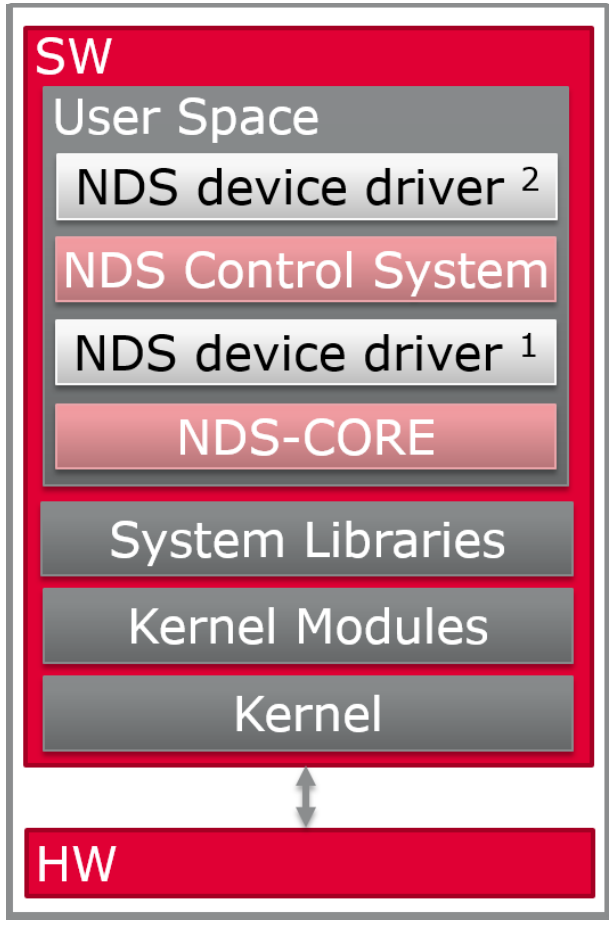
CLASS DIAGRAM

NOMINAL DEVICE SUPPORT V3



NOMINAL DEVICE SUPPORT V3

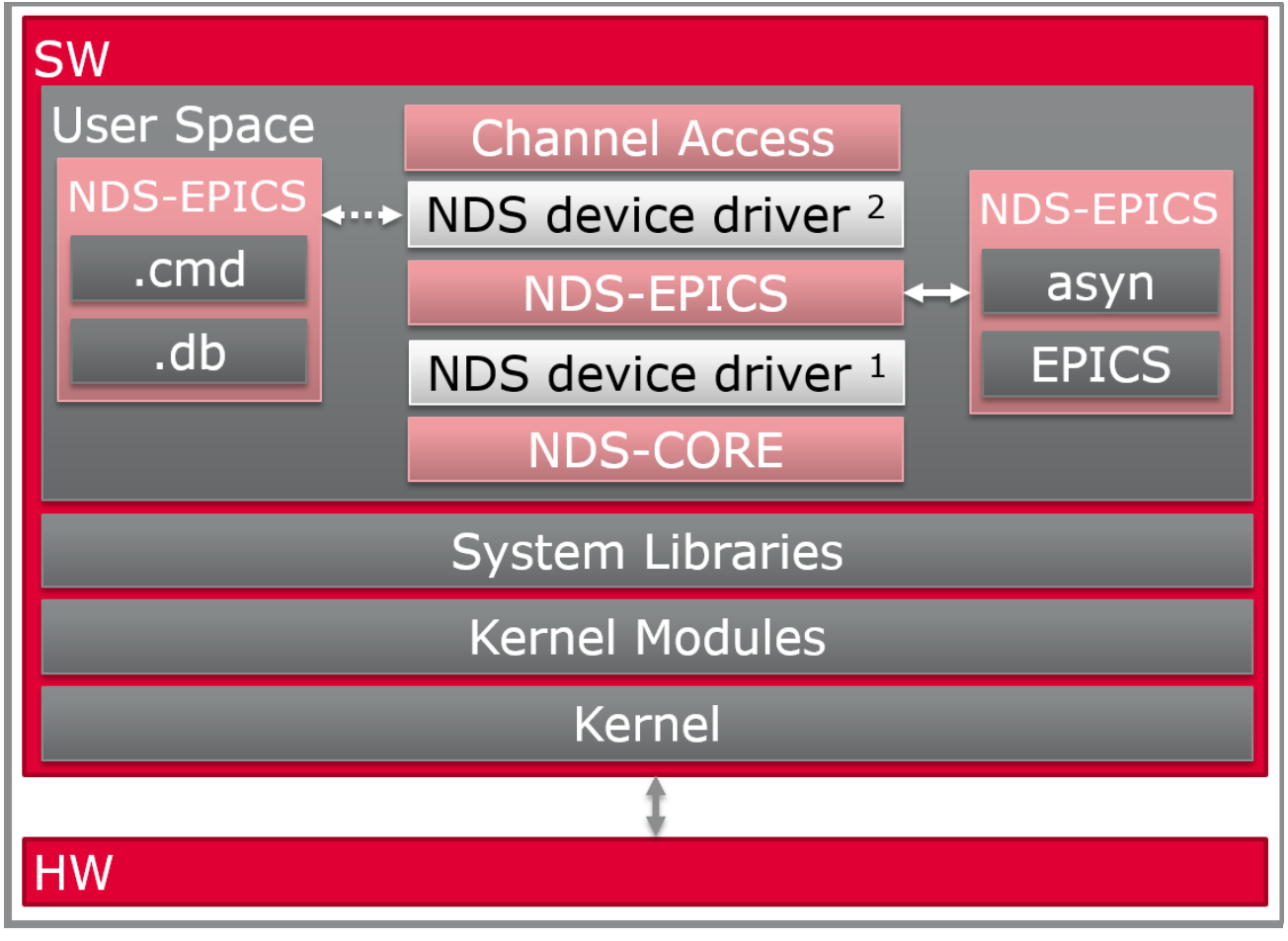
- The NDS Control System takes care of:
  - Registering device drivers
  - Creating device drivers instances
  - Communicating device drivers with rest of world:
    - Setting values
    - Getting values
    - Publishing values
    - Data sharing
- Control System classes inherit from the target system. For instance: *asynPortDriver*



NOMINAL DEVICE SUPPORT V3

NDS3 drivers do not depend on EPICS* …

* Unless NDS-EPICS is considered



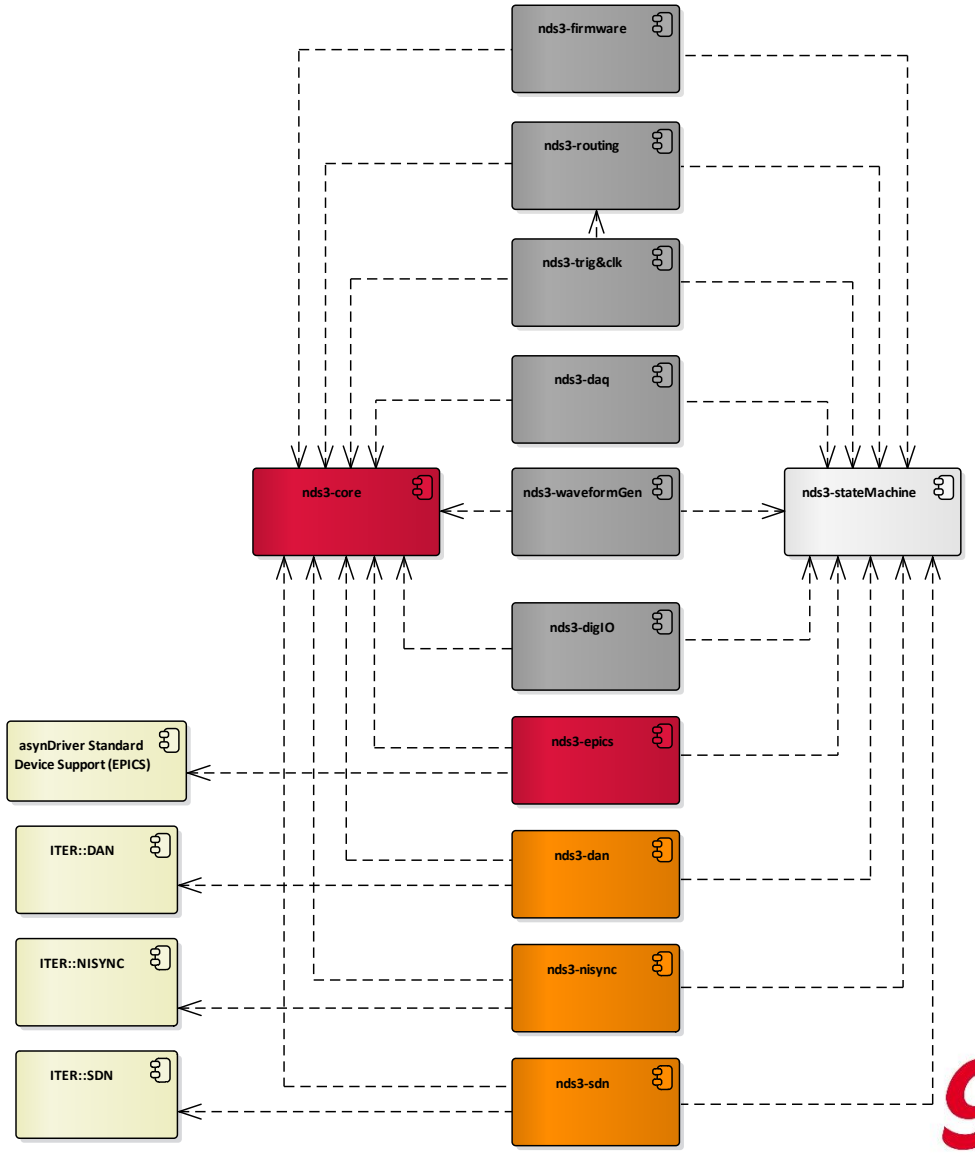
NDS

Contribution to NDS3

CONTRIBUTION TO NDS3

From a logical point of view:

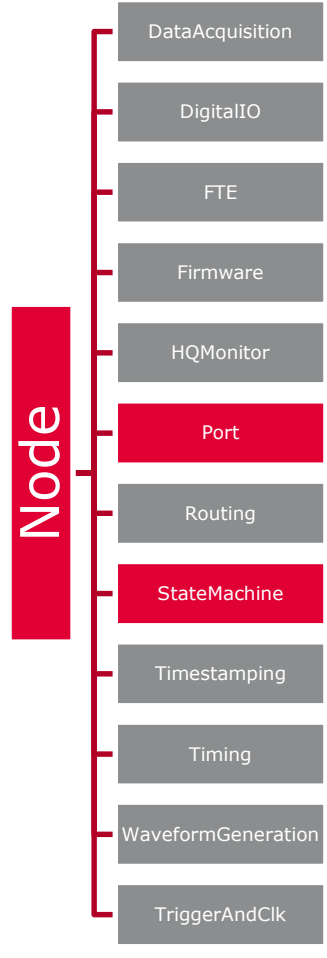
- Existing elements:
  - nds-core
  - nds-epics
  - state machine
- External elements to interface with:
  - asyn
  - ITER-DAN
  - ITER-NISYNC
  - ITER-SDN
- Added support for:
  - Firmware
  - Routing
  - Triggering and clocking
  - Data Acquisition
  - Waveform Generation
  - Digital IO



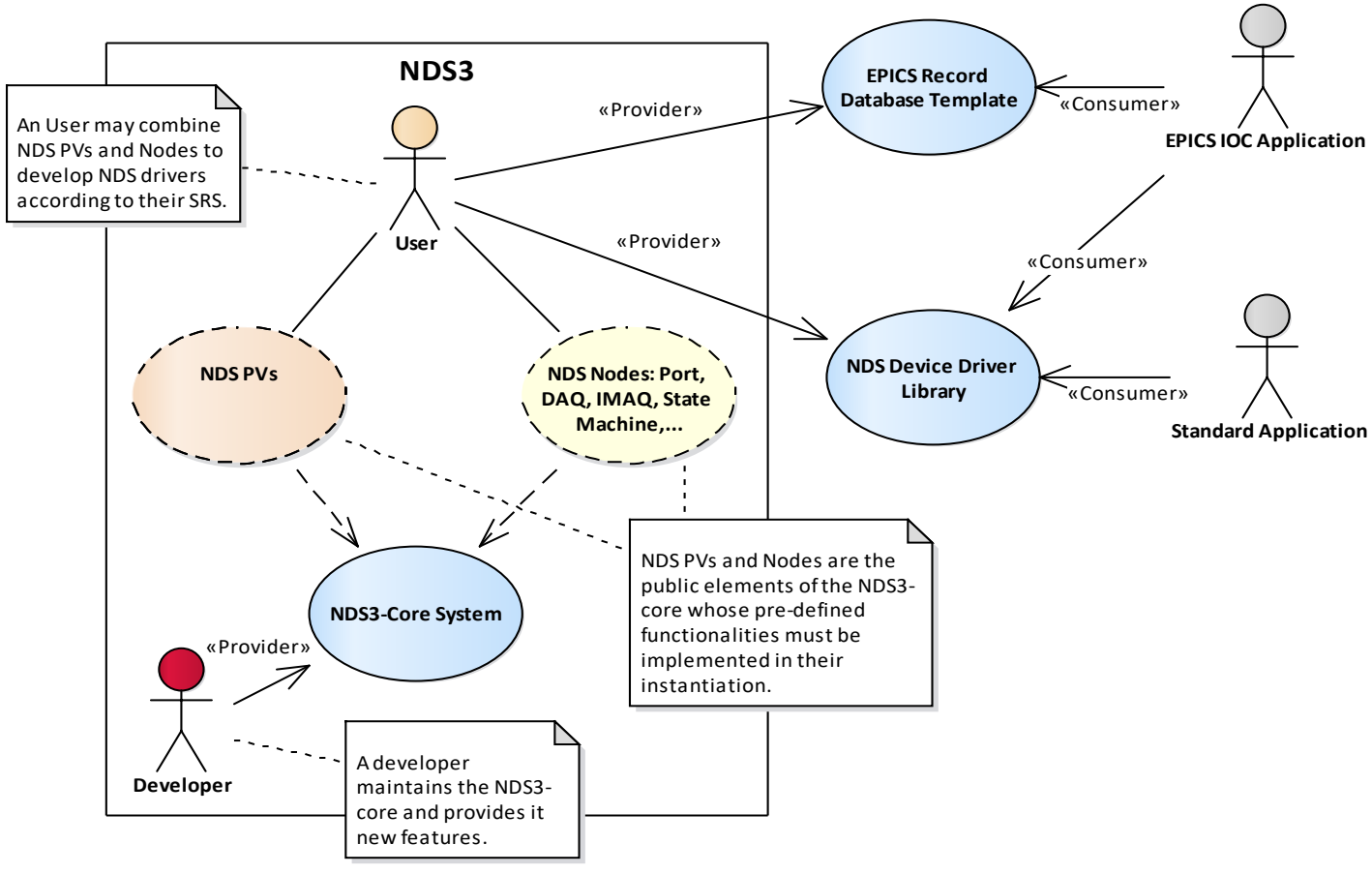
CONTRIBUTION TO NDS3

Current class diagram:

- *DataAcquisition*: acquiring data
- *DigitalIO*: handling digital signals
- *FTE*: handling Future Time Events
- *Firmware*: providing typical parameters
- *HQMonitor*: providing Health and Quality parameters
- *Port*: providing communication with the control system
- *Routing*: routing signals
- *StateMachine*: handling states
- *Timestamping*: handling timestamps
- *Timing*: providing timing parameters
- *WaveformGeneration*: generation of waveforms
- *TriggerAndClk*: generating trigger and clock signals



CONTRIBUTION TO NDS3



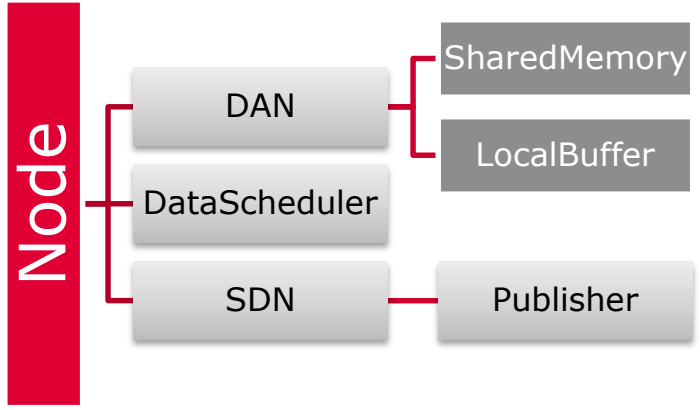
CONTRIBUTION TO NDS3

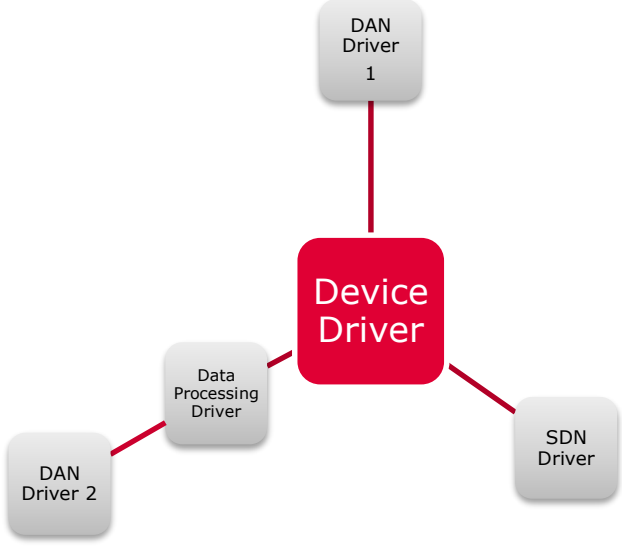
Additionally:

- Documentation: User Manuals, API, Test Plan, Software Architecture and Design Document
- Sample device drivers
- Unit tests with *PyEpics*

Upcoming features:

- DAN nodes
- SDN nodes
- Data Scheduling nodes
- Removing dependencies from device drivers with data sharing

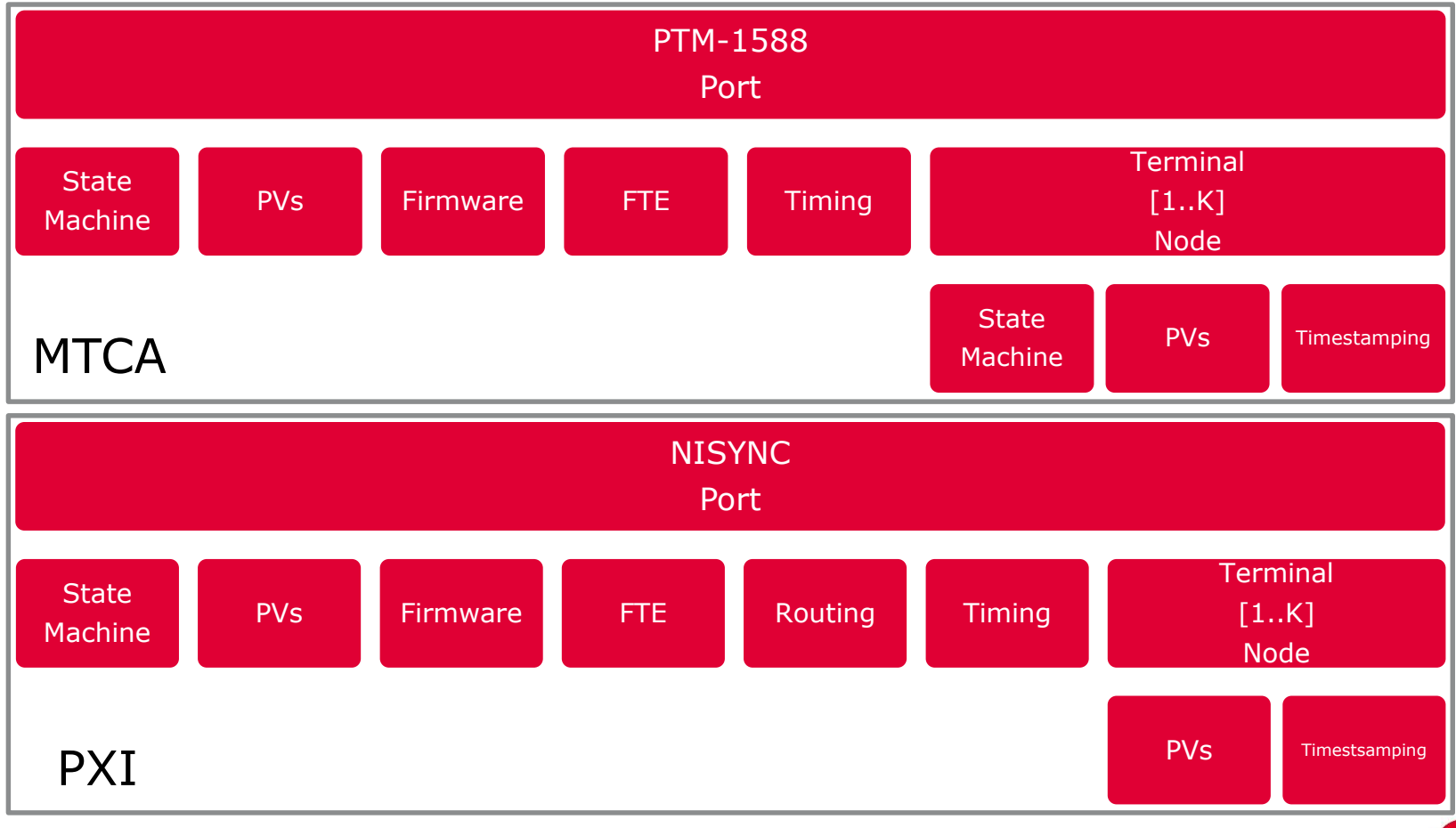




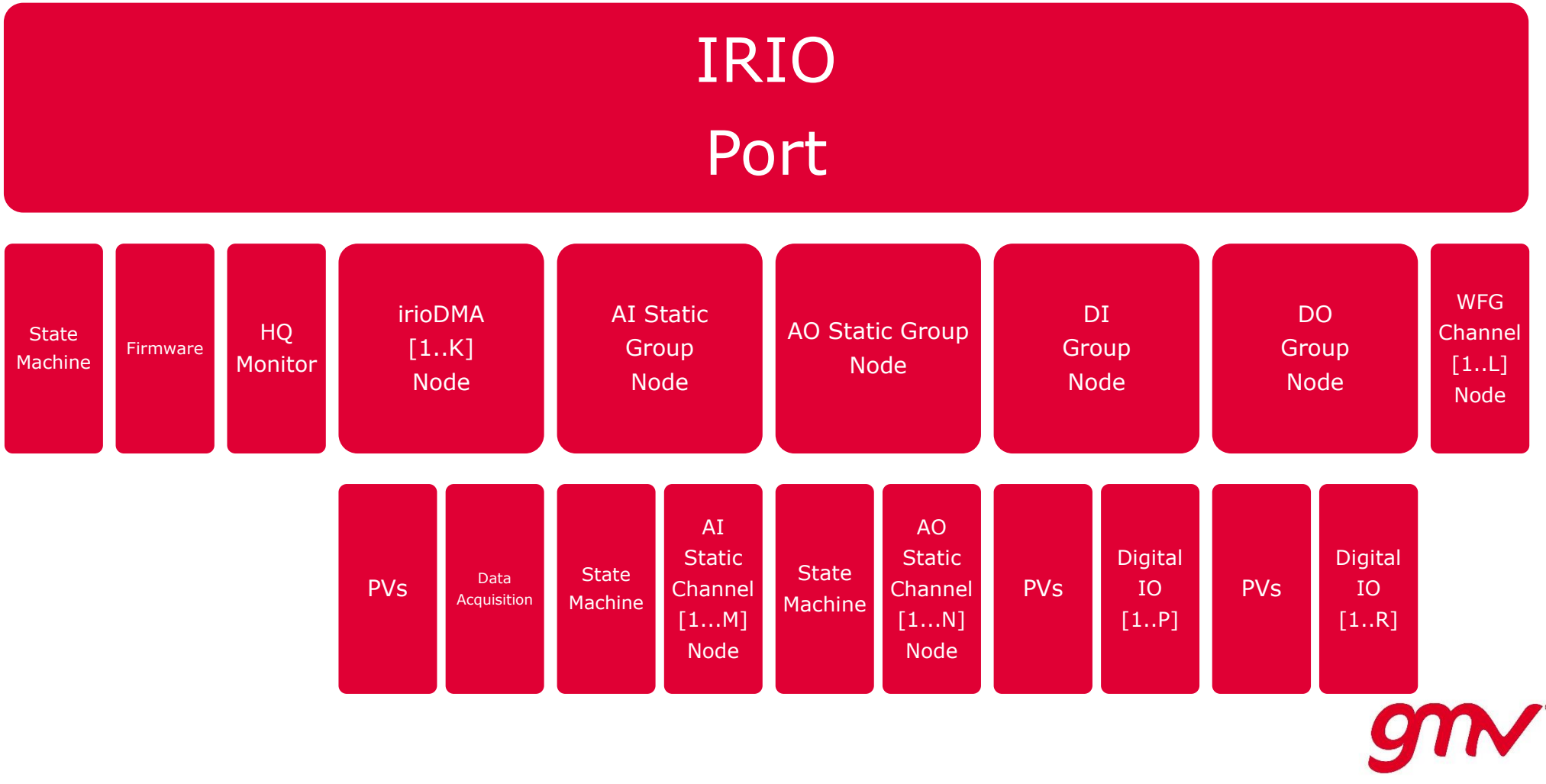
NDS3

Use cases

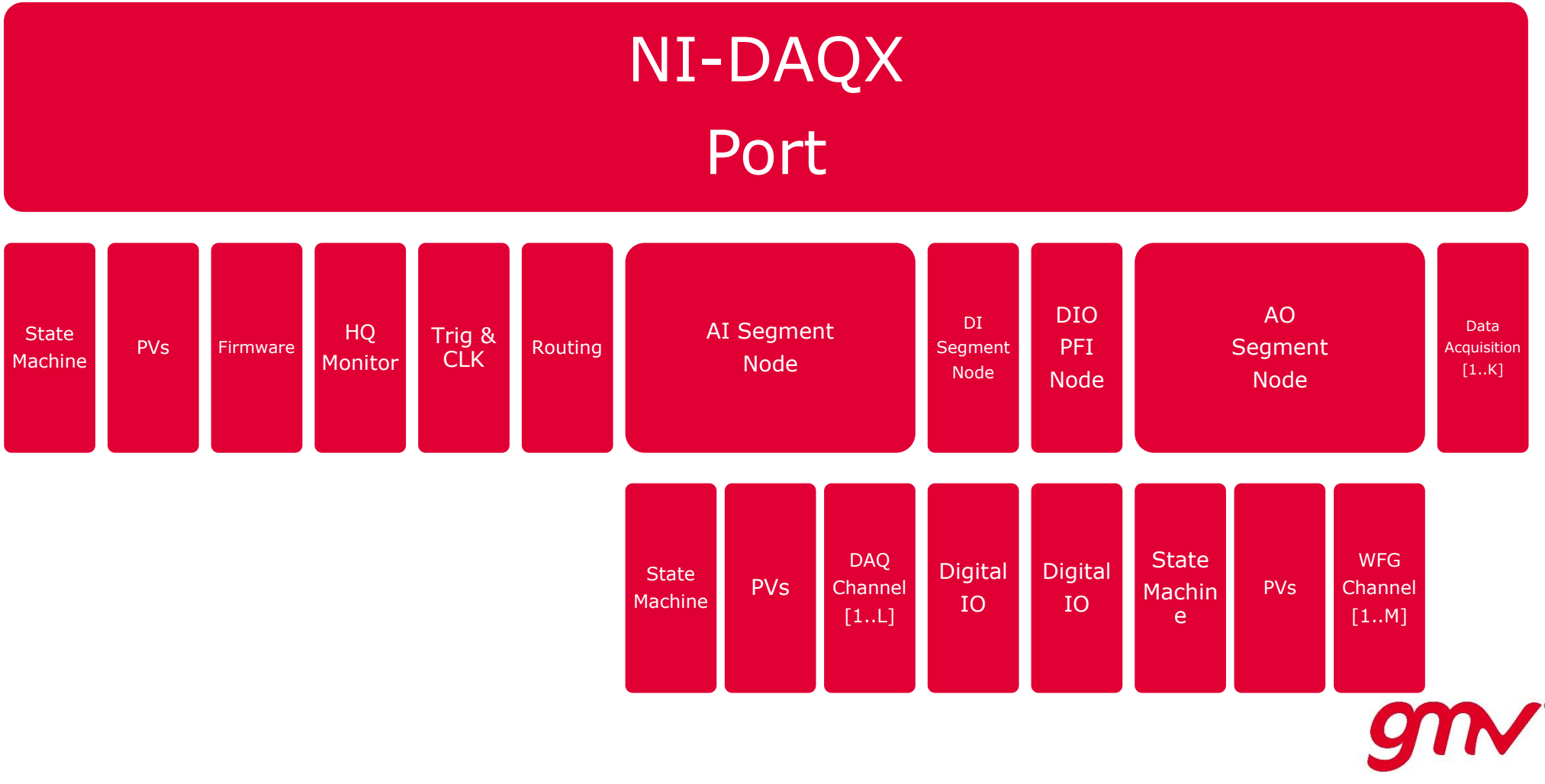
USE CASES


PTM-1588
Port
State Machine
PVs
Firmware
FTE
Timing
Terminal [1..K] Node
State Machine
PVs
Timestamping
MTCA
NISYNC
Port
State Machine
PVs
Firmware
FTE
Routing
Timing
Terminal [1..K] Node
PVs
Timestsamping
PXI

USE CASES


IRIO
Port
State Machine
Firmware
HQ Monitor
irioDMA [1..K] Node
AI Static Group Node
AO Static Group Node
DI Group Node
DO Group Node
WFG Channel [1..L] Node
PVs
Data Acquisition
State Machine
AI Static Channel [1...M] Node
State Machine
AO Static Channel [1...N] Node
PVs
Digital IO [1..P]
PVs
Digital IO [1..R]

USE CASES


NI-DAQX
Port
State Machine
PVs
Firmware
HQ Monitor
Trig & CLK
Routing
AI Segment Node
DI Segment Node
DIO PFI Node
AO Segment Node
Data Acquisition [1..K]
State Machine
PVs
DAQ Channel [1..L]
Digital IO
Digital IO
State Machin e
PVs
WFG Channel [1..M]

NDS

Conclusions and references

CONCLUSIONS AND REFERENCES

- Conclusions:
  - Standardized development of device drivers
  - Independent Control System device drivers
  - Easy usability
  - High scalability
- Suggestions:
  - Adding support for image acquisition with *areaDetector*
- References:
  - Cosylab's repositories: *NDS3* and *NDS3-EPICS*
  - ITER's repositories: *NDS3-CORE* and *NDS3-EPICS*
  - ITER's documents: *NDS User Manual*, *NDS-EPICS Device Support Developer's Guide*
- Acknowledgments:
  - *Technical University of Madrid, Instrumentation and Applied Acoustic Research Group*
  - *ITER International Organization*

www.gmv.com

THANK YOU

gmv BLOG