Data Streaming

Efficient Handling of Large and Small (Detector) Data at the Paul Scherrer Institute

S. Ebner[1], H. Billich[1], H. Brands[1], Ezequiel Panepucci[1], L. Sala[1]
[1] Paul Scherrer Institut (PSI), CH-5232 Villigen PSI, Switzerland

Paper ID: WED3O06

PAUL SCHERRER INSTITUT

The Paul Scherrer Institute, PSI, is the largest research centre for natural and engineering sciences within Switzerland

SwissFEL
Swiss Free Electron Laser

SLS
Swiss Light Source

SINQ
Swiss Spallation Neutron Source

HIPA
The high-intensity proton accelerator

...

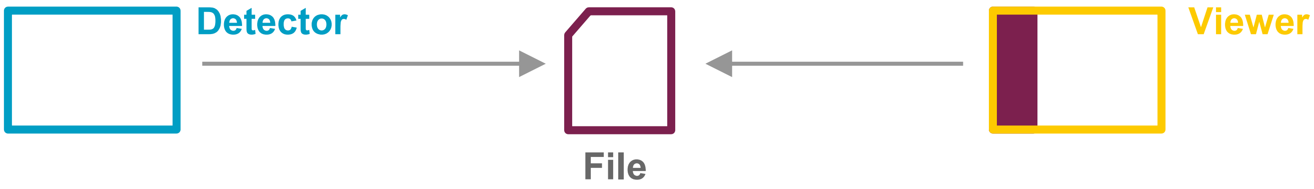
http://www.psi.ch

Outline

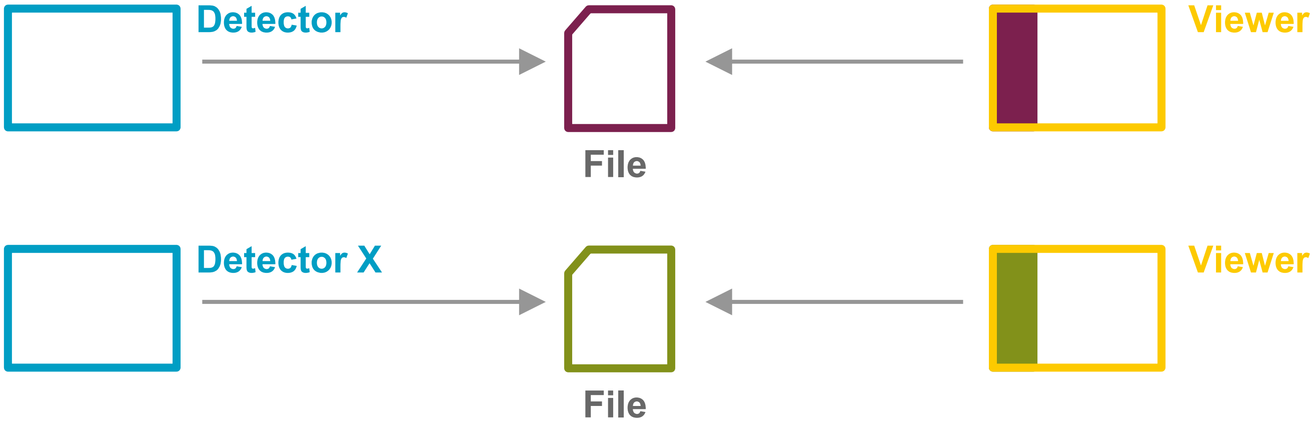
- Motivation
- Objectives
- Design
- Lessons Learned
- Outlook
- Questions

Motivation

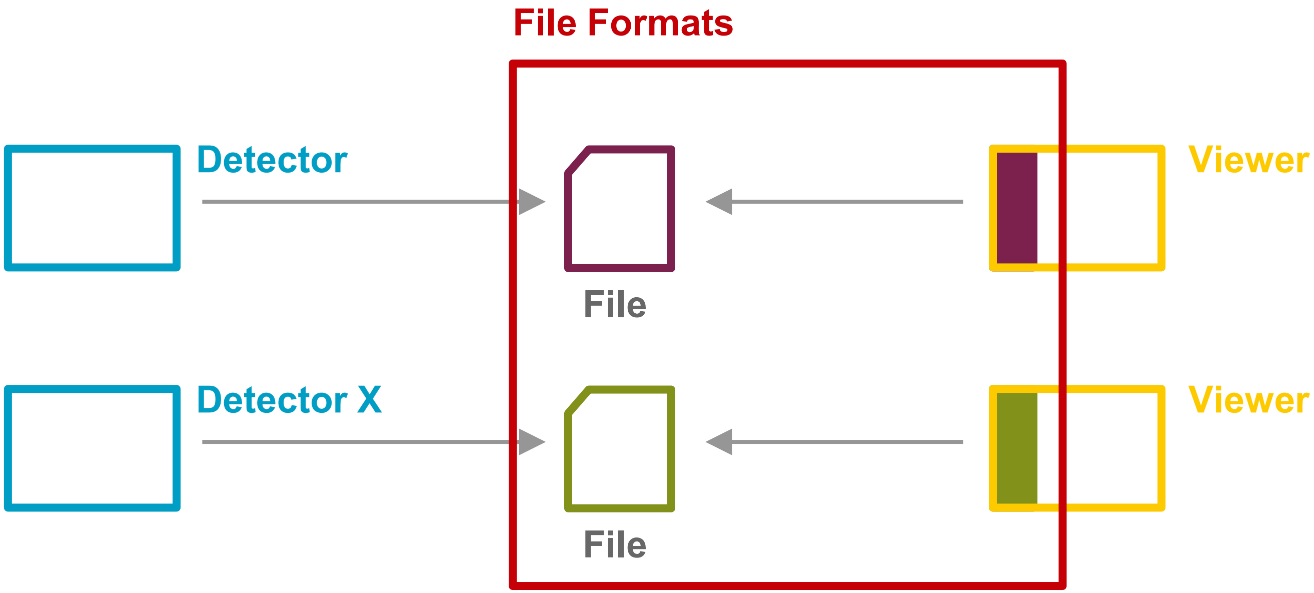
Motivation



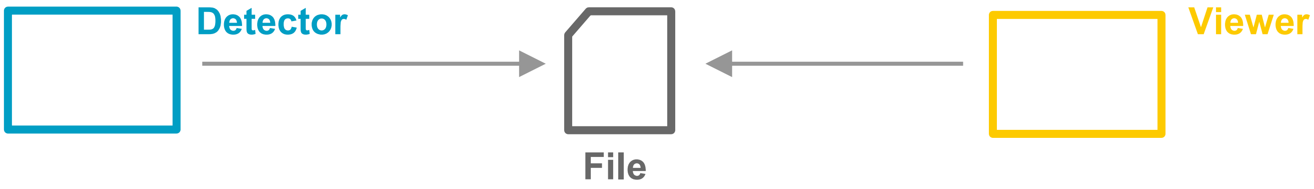
Motivation



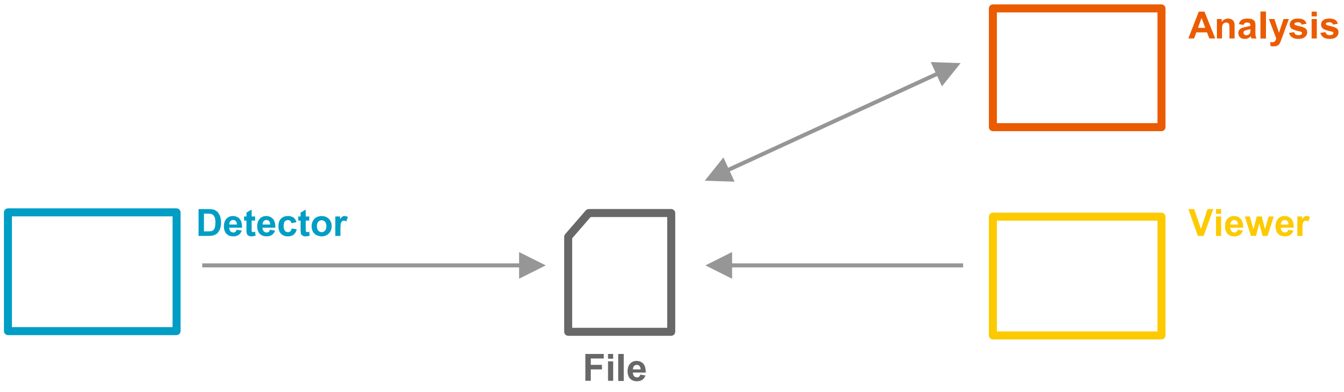
Motivation



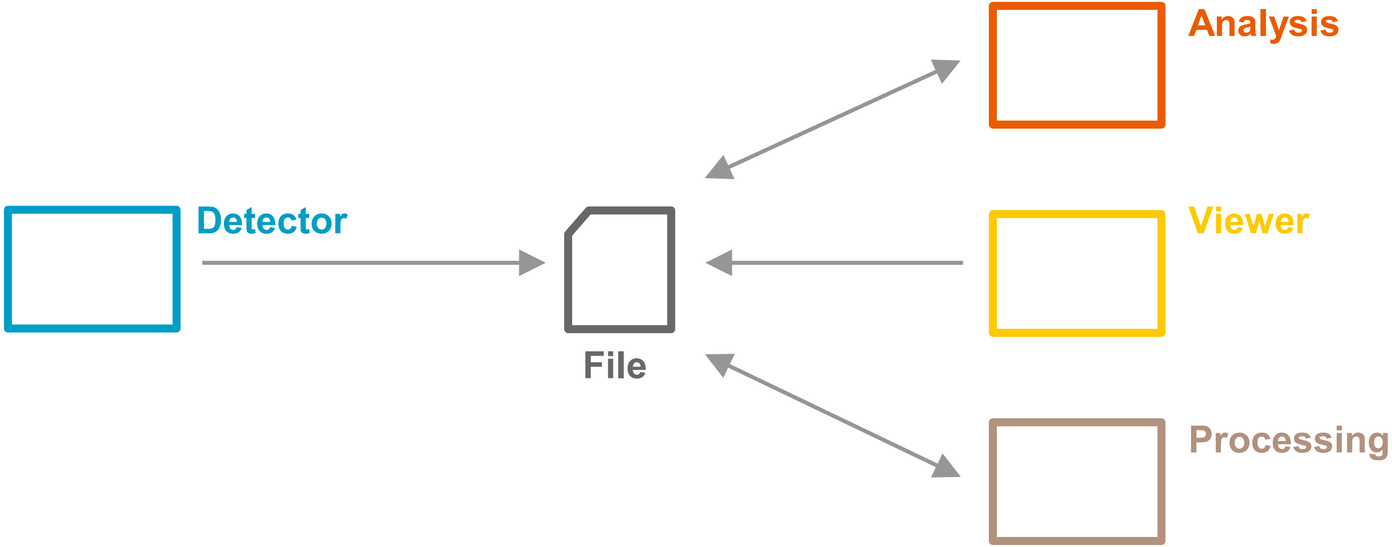
Motivation



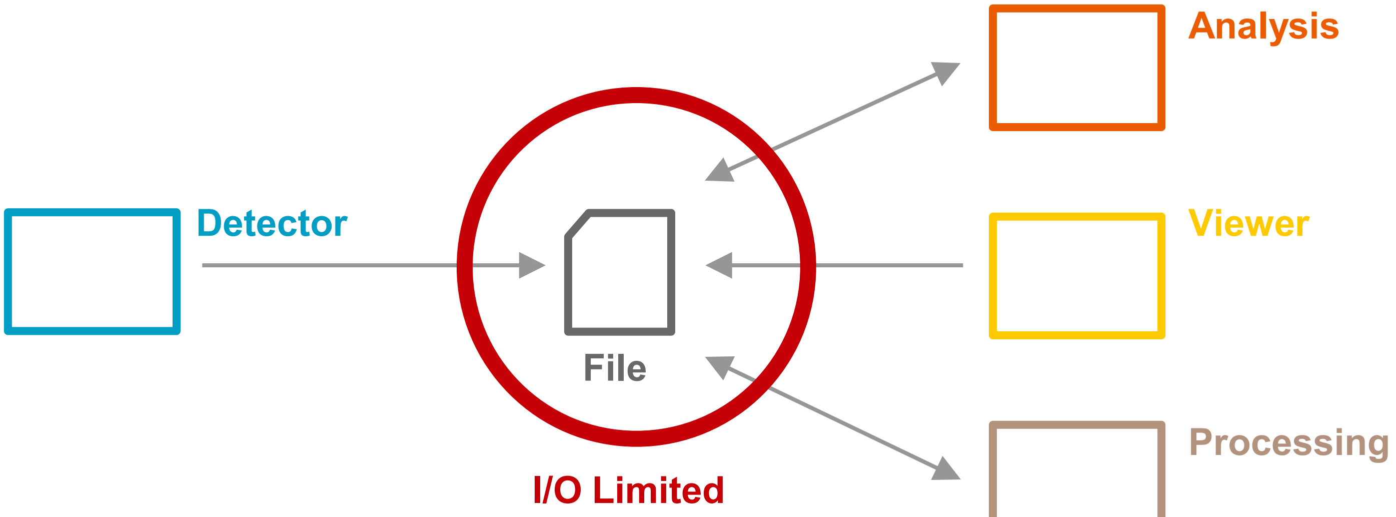
Motivation



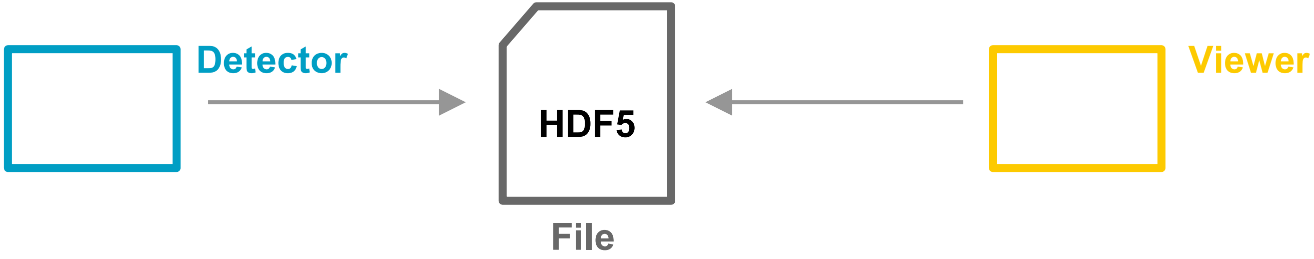
Motivation



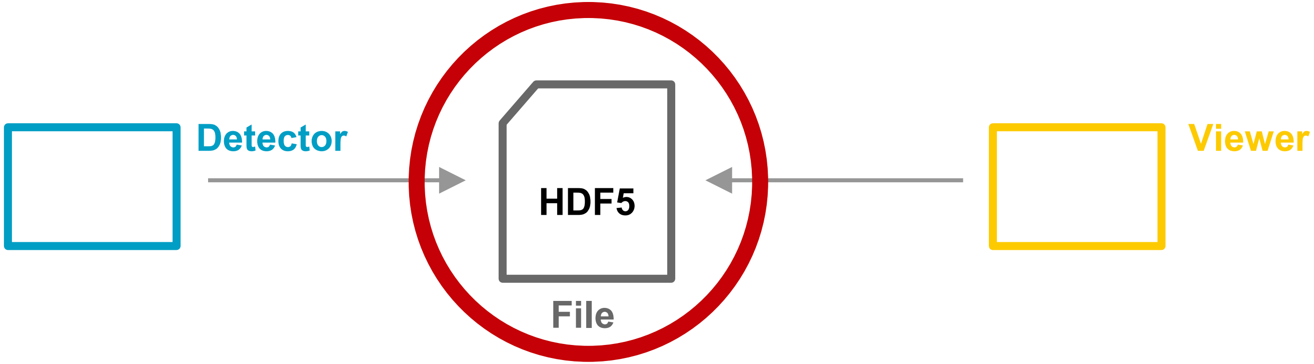
Motivation



Motivation



Motivation



Delay / Concurrency

Motivation

…

Objectives

Objectives

Code

Complexity

Design

...

Minimize

Conflicts

Dependencies

Objectives

Stability

Programming Language

Exchangeability

Maintainability

Flexibility

Design

Parallelization

Maximize

Decoupling

Simplicity

Scaleability

(Code) Reuse

…

Design

Design

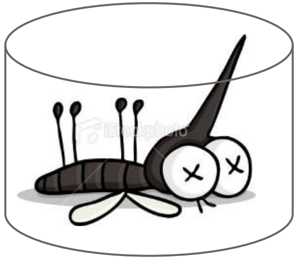
Principles

Design

FRAMEWORKS SUCK

Principles

Design



Data on Disks is DEAD

Principles

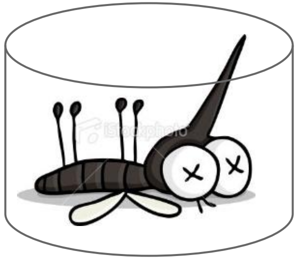
FRAMEWORKS
SUCK

Design

Control/Data

Separation of Concerns

Data taking, viewing, analysis, persistence, …



Data on Disks is DEAD

Principles

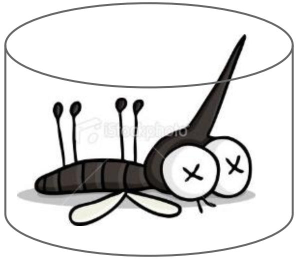
FRAMEWORKS SUCK

Design

Control/Data

Separation of Concerns

Data taking, viewing, analysis, persistence, …



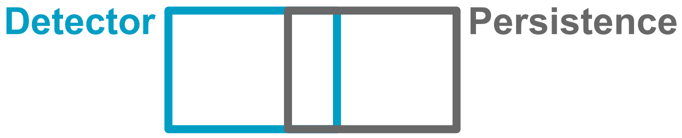
Data on Disks is DEAD

Principles

FRAMEWORKS SUCK

Standardisation/ APIs on the Network

Design

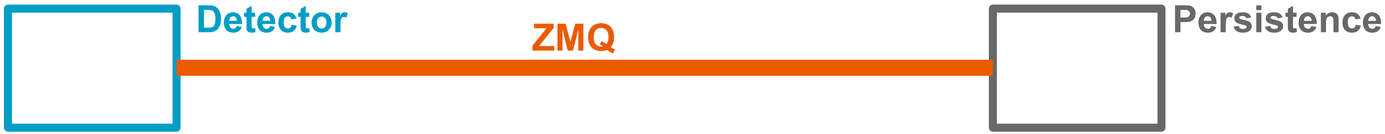


Detector

Persistence

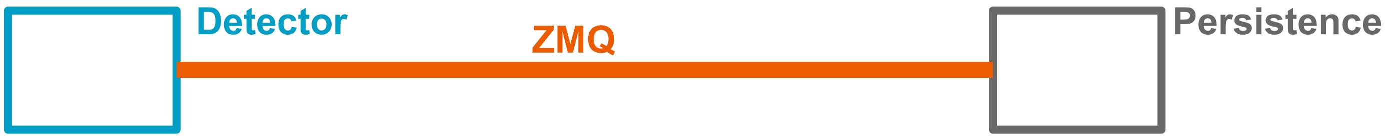
Design

Design



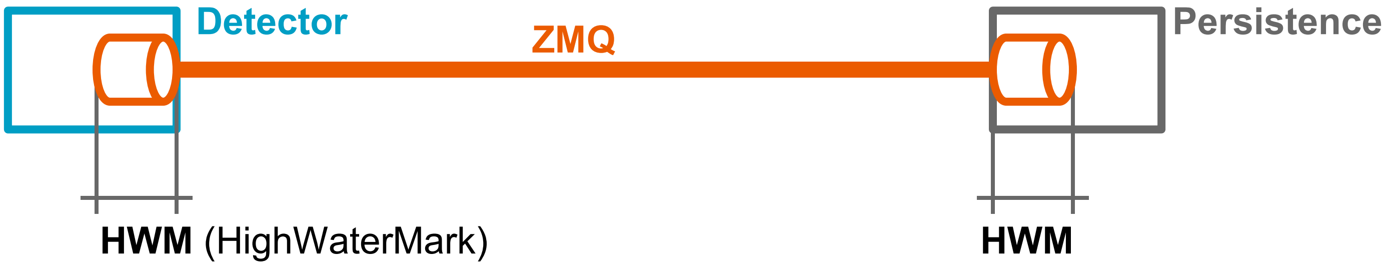
Design

Delivery Scheme: PUSH/PULL
PUB/SUB

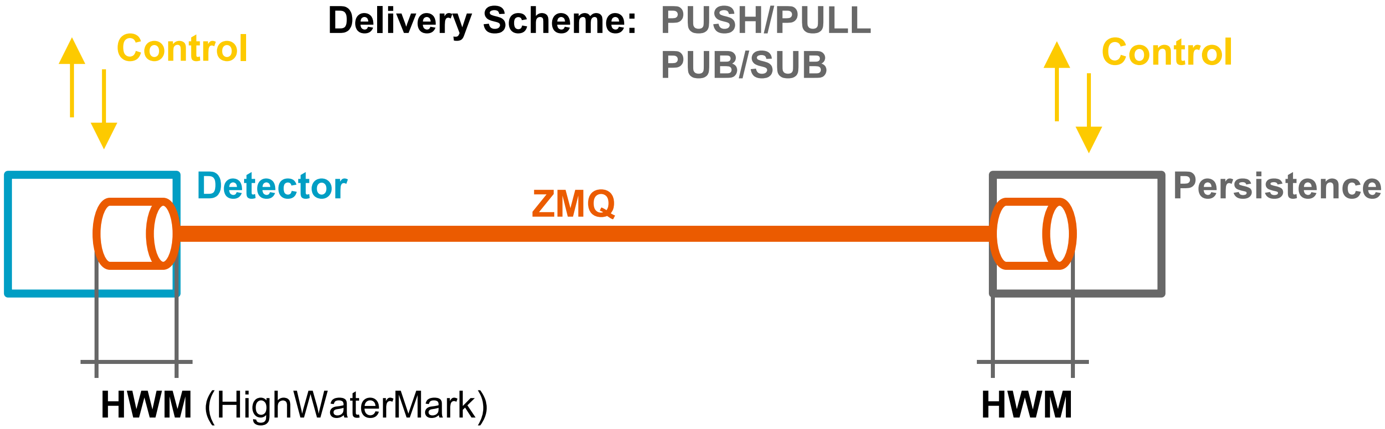


Design

Delivery Scheme: PUSH/PULL
PUB/SUB

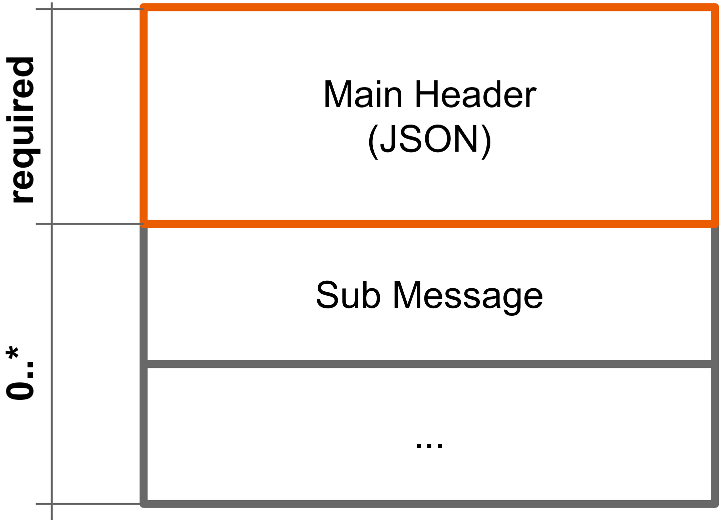


Design



Delivery Scheme: PUSH/PULL
PUB/SUB

Design - Message

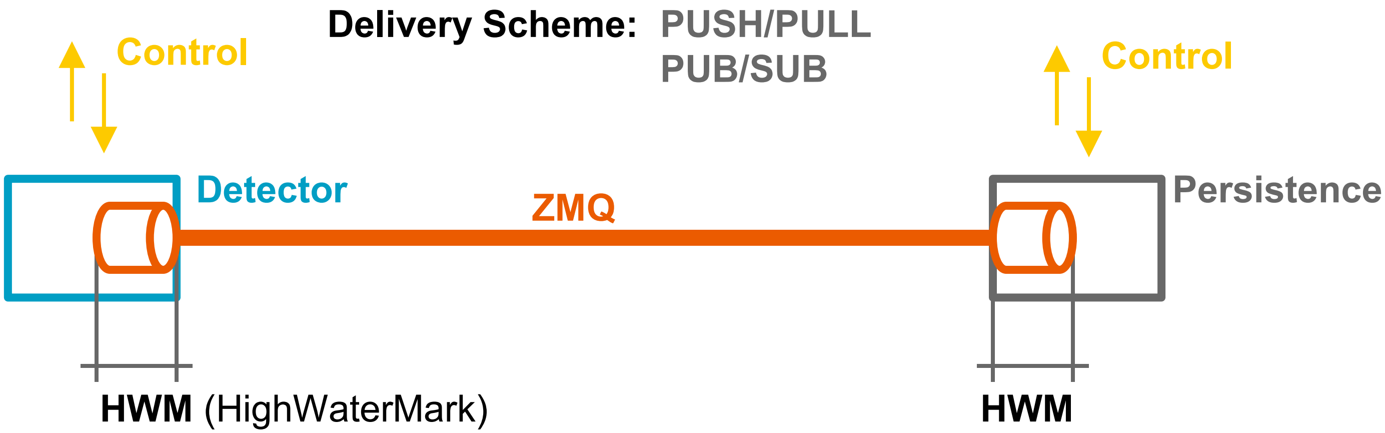


```
{"htype":"array-1.0",
"shape":[10,20],
"type":"uint16", "frame":0}
```

htype defines content of main header as well as the structure of the whole message (sub messages)

Sub Message(s) can be binary or JSON

Design


Delivery Scheme: PUSH/PULL
PUB/SUB
Control
Control
Detector
ZMQ
Persistence
HWM (HighWaterMark)
HWM

Design - Receiver Example

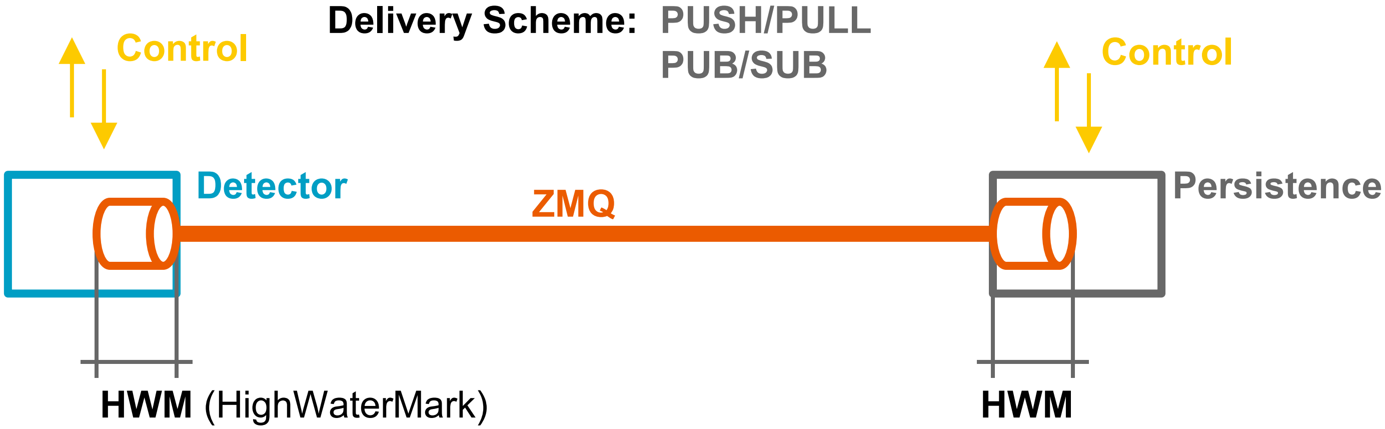
```
import zmq
context = zmq.Context.instance()
sock = context.socket(zmq.PULL)
sock.connect('tcp://hostname:9999')

while True:
    header = sock.recv()
    print header
    while socket.getsockopt(zmq.RCVMORE)
        data = sock.recv()
```

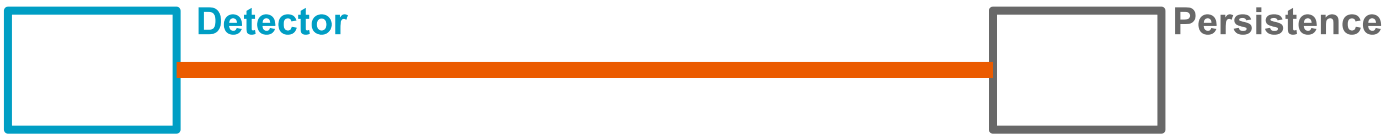
Design - Persistence Example

```
import zmq
context = zmq.Context.instance()
sock = context.socket(zmq.PULL)
sock.connect('tcp://hostname:9999')
counter = 0
while True:
    header = sock.recv()
    print header
    while socket.getsockopt(zmq.RCVMORE)
        data = sock.recv()
        open('file_{}.raw'.format(counter),'wb')
        f.write(data)
        f.close()
        counter += 1
```

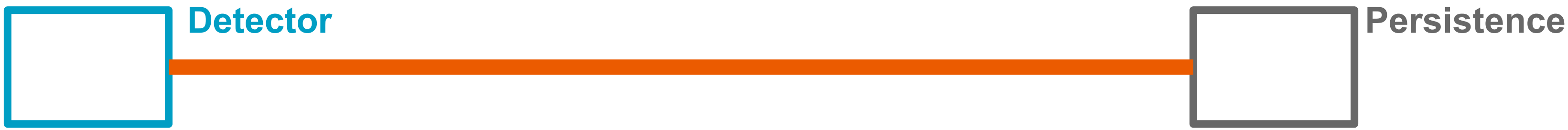
Design


Delivery Scheme: PUSH/PULL
PUB/SUB
Control
Control
Detector
ZMQ
Persistence
HWM (HighWaterMark)
HWM

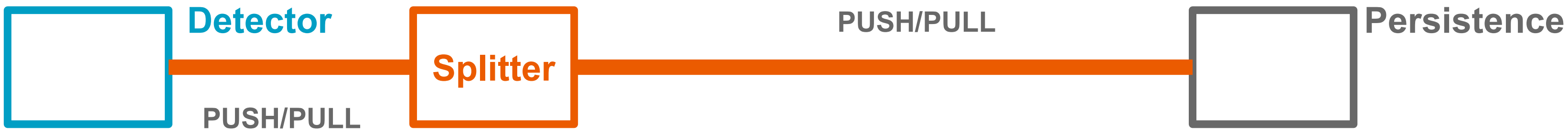
Design



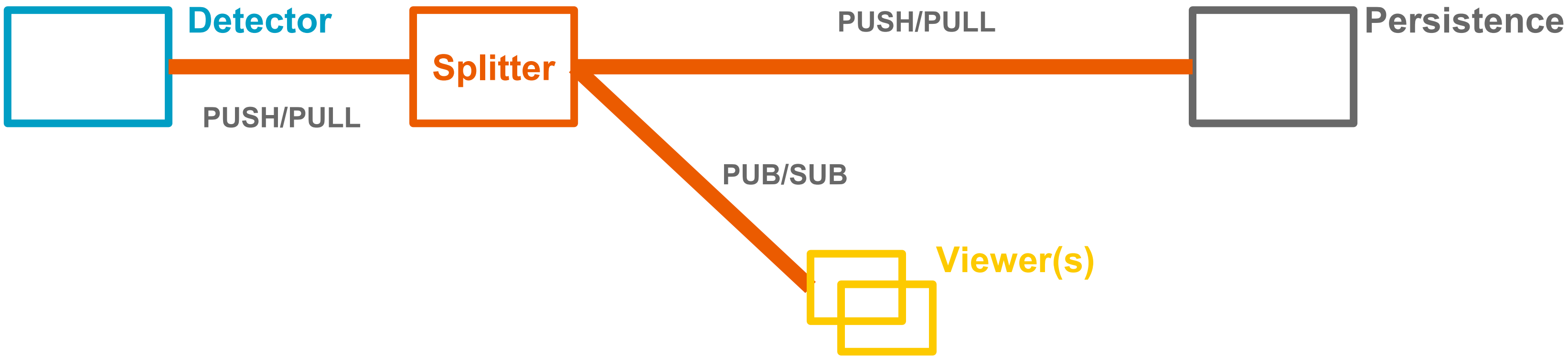
Design



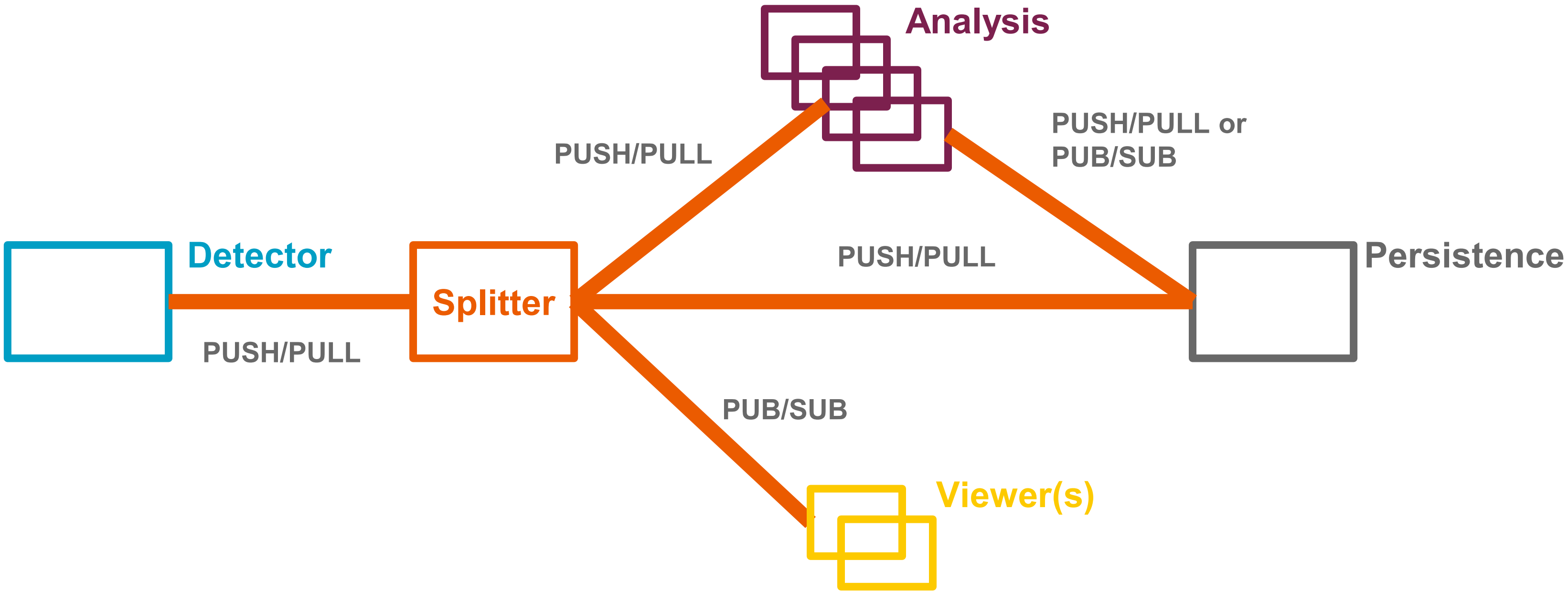
Design



Design


Detector
Splitter
PUSH/PULL
PUSH/PULL
Persistence
PUB/SUB
Viewer(s)

Design


Analysis
PUSH/PULL
PUSH/PULL or
PUB/SUB
Detector
Splitter
PUSH/PULL
Persistence
PUSH/PULL
PUB/SUB
Viewer(s)

Design - Advantages

- No delay in online analysis and viewing
- Re-use of components (e.g. file writing)
- Adding/exchange of components (e.g. detector, processing)
- Dynamic and automatic scaling (e.g. of processing) due to push/pull scheme
- Easy debugging through JSON header
- ...

Design - Disadvantages

- It is a distributed system with all its challenges ...

Detectors

GigaFROST

Pixels: 2016*2016 / 12 bit
Exposure: 2µs exposure 1255 Hz
Data Rate: 61.21Gbit/s - 7.651 GByte/s

Write Stream: 2 GByte/s
Preview Stream: 5 Hz

PCO Edge

Pixels: 2560x2160 / 16 bit
Speed: 100 Hz mit dem 286Mhz ADC Rolling Shutter
Data Rate: 1GByte/s

Write Stream: 1GByte/s
Preview Stream

Lessons Learned

Lessons Learned

- Keep code of components simple (no all-in-one-components)
- Failure tolerant design of components/software
    - Automatic restart of components
    - ...
- Take special care when starting and closing a connection
- Setting ZMQ high watermark (on client and server) "right" is very important
- Central logging is a must
- Statistics for all components are needed
    - Messages received
    - Messages dropped
- ...

Outlook

Outlook - Improvements

- Improve tooling for Python, Java, …
- Improve (central) debugging and logging
- Work on the easy configuration and orchestration of the components/modules (**black box** concept)
- …

Interested ?

Contact Us

Questions ?

Slides

Contact

Simon Ebner
Paul Scherrer Institute
WBGB/001
5232 Villigen PSI

simon.ebner@psi.ch

Colors