# Common Specification F-CS-B-0005e:

# Integration of the Beam Position Monitoring System into the Controls & Operation Environment

https://edms.cern.ch/document/1823368/

prim. authors: O. Chorniy (BI), J. Fitzek (ACO), K. Lang (BI), R. Singh (BI), R. Steinhagen (FC²-PL)

checked-by: FAIR-C2WG-ALL (e-group), F. Gressier (QA), D. Ondreka (SYS), S. Reimann (Head OP), P. Schütt (OP), M. Schwickert (BI)

approved by: P. Gerhard (UNILAC), F. Hagenbuck (HEBT), F. Herfurth (CRYRING / HITRAP), R. Hollinger (Ion Sources), K. Knie (p-Linac & p-bar Separator), I. Koop (CR), D. Prasuhn (HESR), H. Reich-Sprenger (CS), H. Simon (Super-FRS), P. Spiller (SIS-18/SIS-100), M. Steck (ESR), R. Bär (Head ACO), M. Schwickert (Head BI), S. Reimann (OP), R. Steinhagen (FC²-PL)

# Scope

- BPMs are *‘bread-and-butter’* instruments for acc. operation and used in parallel in a variety of use cases

- Primary goals:
  - improve use of beam-based optimisation techniques
    → better qualitative and quantitative beam/machine performance
    - better reproducibility of beam parameters
    - faster/more reliable setup of new BPCs
  - simplification and deployment of same optimisation, tools, OP paradigm for all FAIR (and existing) accelerators
    → no resources to support for “isolated island” solutions
    - abstraction of vendor specific HW & SW interfaces

# List of known Users/Use-Cases
## ...ie. those directly depending on BPMs

**A** Cycle-to-Cycle Feedbacks

1. *Injection Steering* <orbit & trajectory>
2. *Extraction Steering* <orbit & trajectory>
3. *Cycle-to-Cycle Orbit-Feedback (& Radial-Loop)* <orbit interface only>

**B** Archiving System

1. *Regular/OP Archiving* <orbit & trajectory>
2. *Post-Mortem System* <trajectory interface>

**C** Machine-Protection

1. *Software Interlock System / MASP* <orbit & trajectory>

**D** Optics Measurement

1. *LOCO-based* <orbit & trajectory>
2. *Optics via phase-advance* <trajectory interface>

**E** Machine Specific Clients

1. *Multi-Turn Injection Optimisation (SIS18, CRYRING)* <orbit & trajectory>
2. *Collimator/Cleaning set-up (SIS100)* <orbit interface>
3. *MPS validation (SIS100)* <orbit & trajectory>
4. *Luminosity Steering (HESR)* <orbit & trajectory>

**F** *Machine-Development*

1. *bunch-by-bunch expert diagnostics* <bunch-by-bunch>
2. *machine development specific applications* <orbit & trajectory>

**G** *Tune (Q) and Chromaticity (Q') Diagnostics*

1. *Q/Q' control after injection/ramp/pre-extraction* <trajectory & bunch-by-bunch>
2. *Q/Q' control during injection (MTI)* <trajectory & bunch-by-bunch>
3. *Q/Q' control for slow-extraction*

extended due to Libera-FEC capabilities and BI/ACO's Detailed BPM Specification integrations

Important constraint:

use-cases have to be served and data has to be provided *in parallel* for all beam production chains (BPCs)

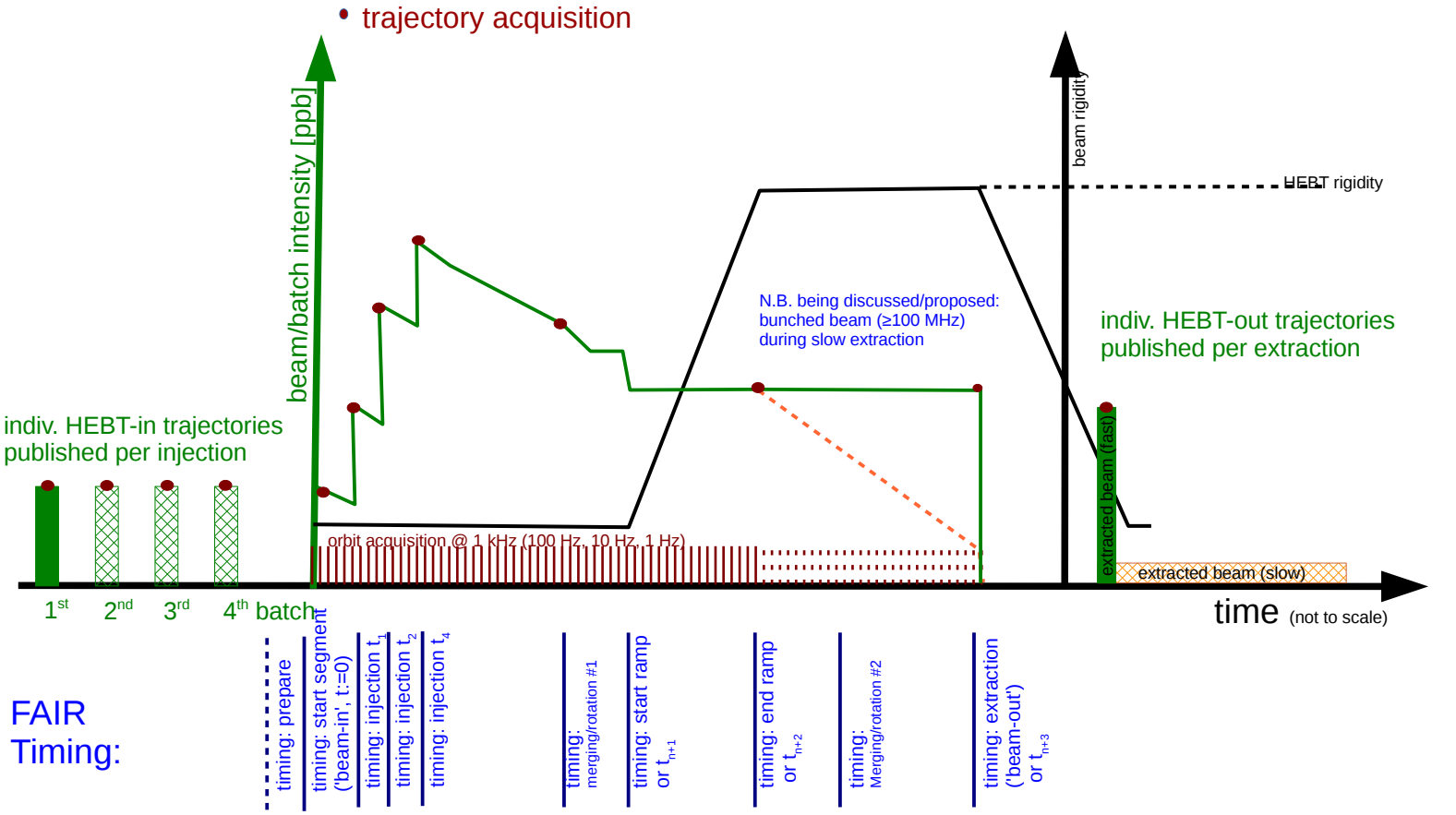
↔ multi-user || optimisation

specific use-cases being implemented as self-contained beam-based micro-services

→ expect ~14 services/machine
↔ 1 dev/acc. expert per BMS paradigm

# BPM Aggregate Definitions
# Data Acquisition, Client API Interfaces & Synchronisation

https://edms.cern.ch/document/1823368/




**transfer line 1**
e.g. SIS18 → HEBT → SIS100
or: SIS18 → HEBT → ESR
or: ESR → HEBT → CRYRING

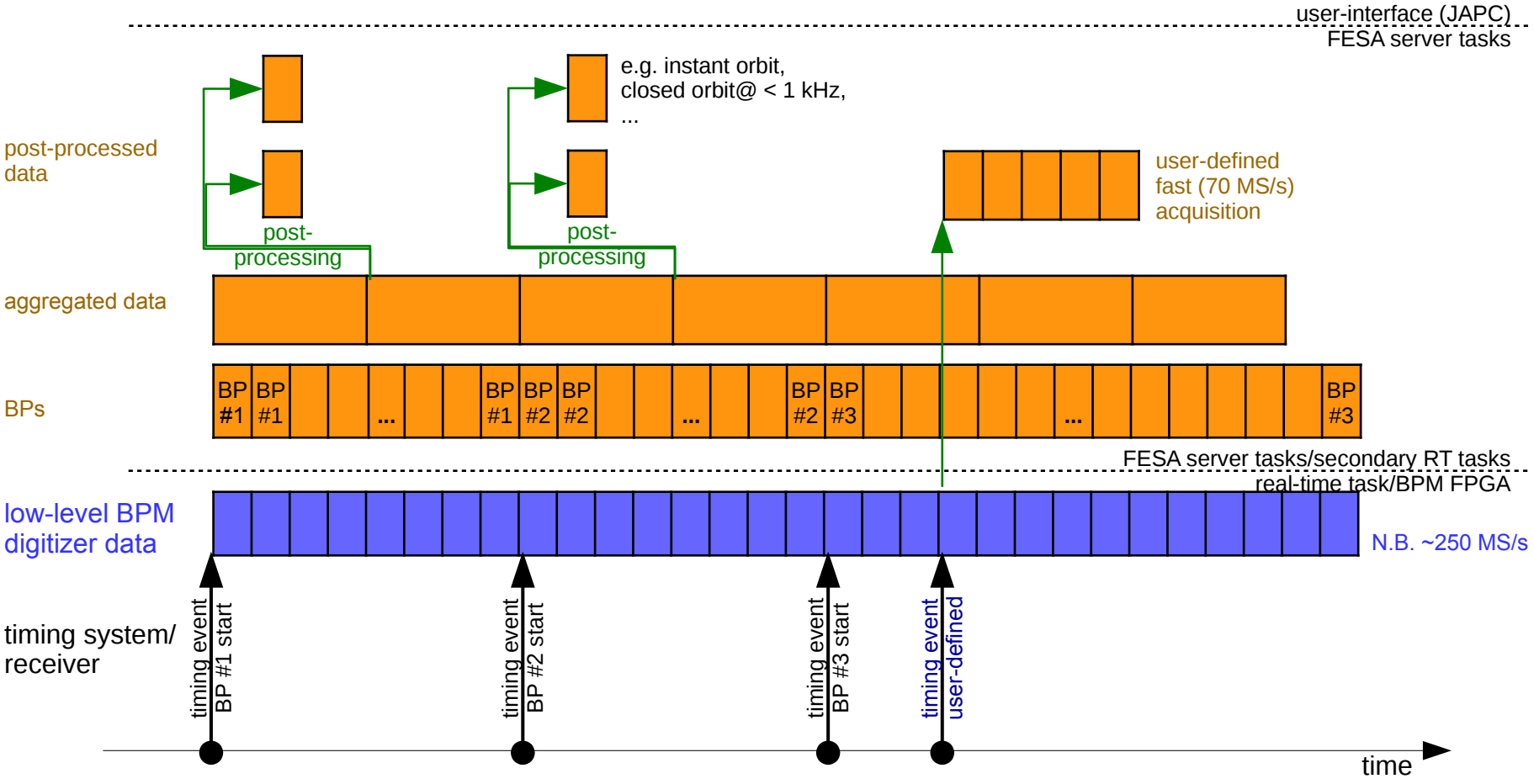
**Circular Machine Cycle**
e.g. SIS100, CR, HESR, CRYRING, SIS18, ESR

**transfer line 2**
e.g. SIS100 → HEBT → $\bar{p}$-target/CR
or. SIS100 → HEBT → CBM

# BPM Aggregate Definitions

https://edms.cern.ch/document/1823368/

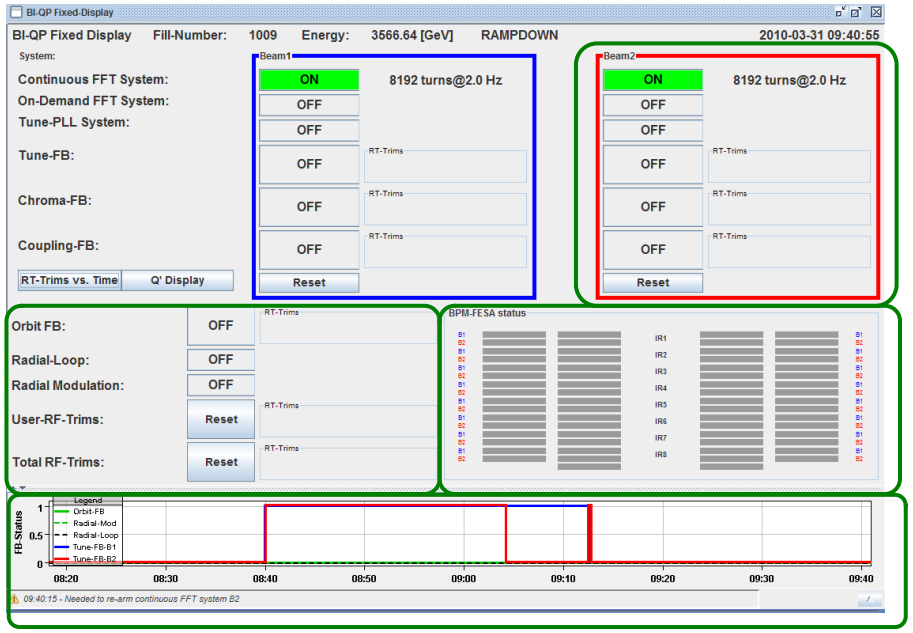
low-level acquisition scheme is flexible enough to handle GSI & FAIR requirements regardless of fast-ramping or storage mode – ?!complete!?



# Top-Level Abstraction: Beam-Based Services

(N.B. operator needs to deal with 4-5 accelerators and more systems in parallel)

- “The best hardware is only as good as its software integration”
- Limited bandwidth of expert knowledge & most operators need to keep a wide focus to tackle many different tasks: KISS – Keep it Simple and Safe → hierarchy of views with increasing/decreasing complexity
- Some reference examples (older but still valid and in active use):




simple beam-based service status and dependent systems ‘ON’-OFF’-’PARTIAL’-’FAILURE’

service performance metric: max trim strengths, latencies, simple user UI failure mitigation actions,

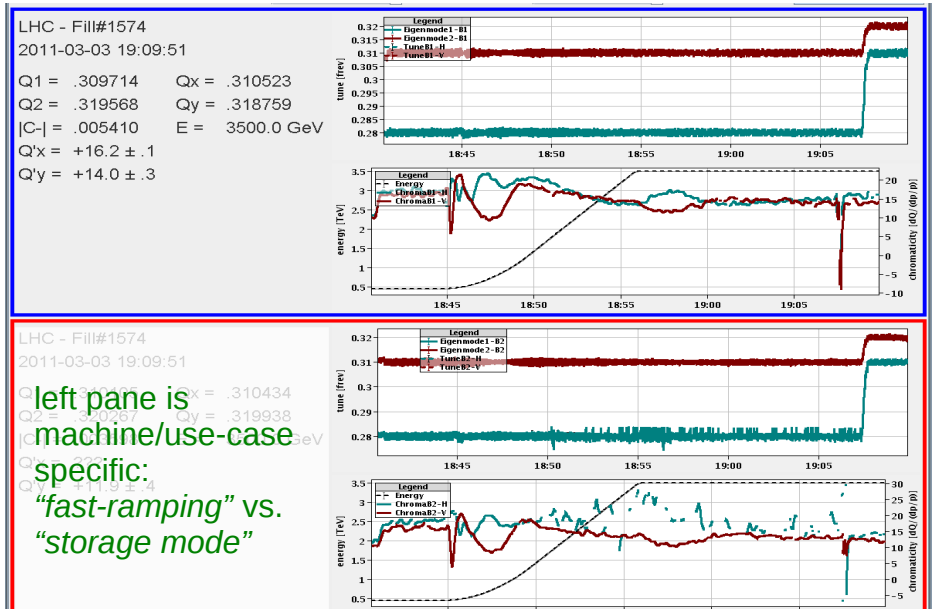
service status history (ie. when was it switched on/off), failure indicators, ...

# Top-Level Abstraction: Beam-Based Services – Tracking Service performance

- Example

“what the beam parameter does”
ie. summary of the achieved parameter stability w.r.t. reference

“what the beam-based service does”
ie. summary of required trim corrections/time evolution with the BPC

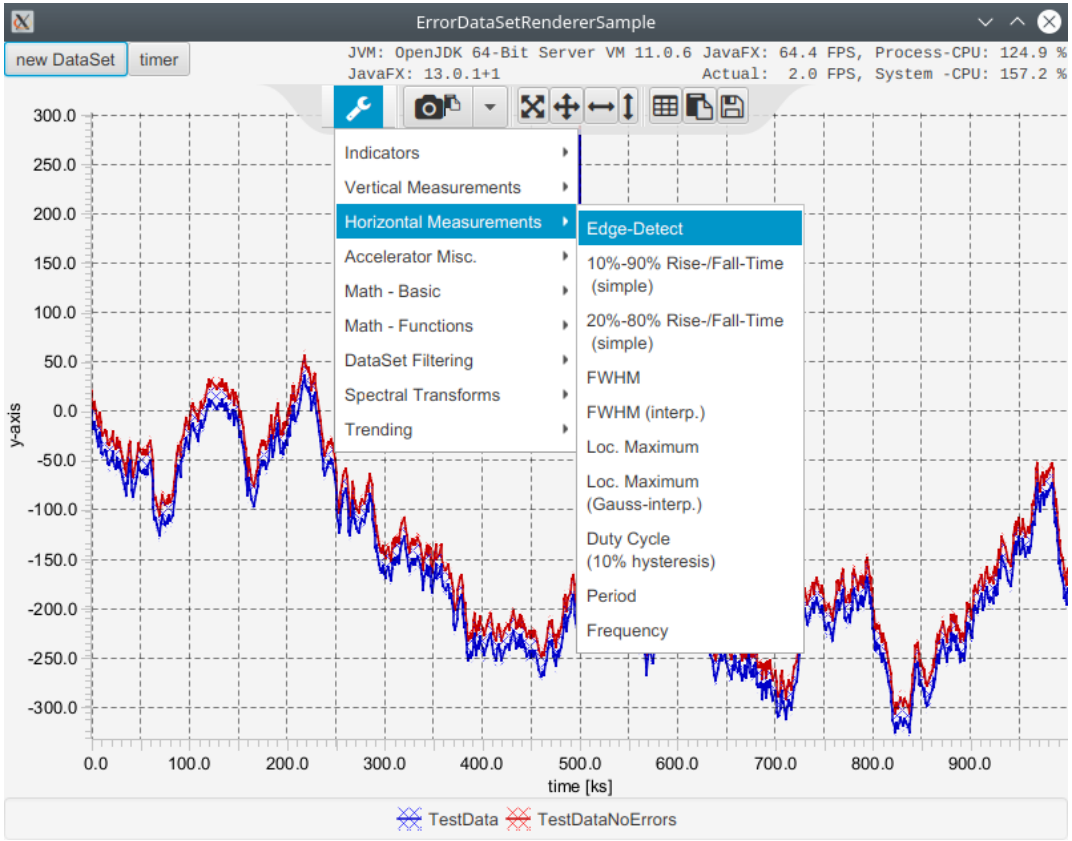




- most UI visualisation features are now out-of-the box Chart-fx core features and require little- to no-customisation, e.g.
  - OP/Expert use-case within the same UI and separated via a tab-style configurations
    → @see DataViewer and related sample …
  - generic CMW → JavaFX/ChartFX bindings (A. Krimm, prototype)
  - under development: long-term maintainable boiler-plate-code-free service framework (REST, CMW, … integration)

# Missing UI Functionality?
## new Chart-fx Options for the Control Room

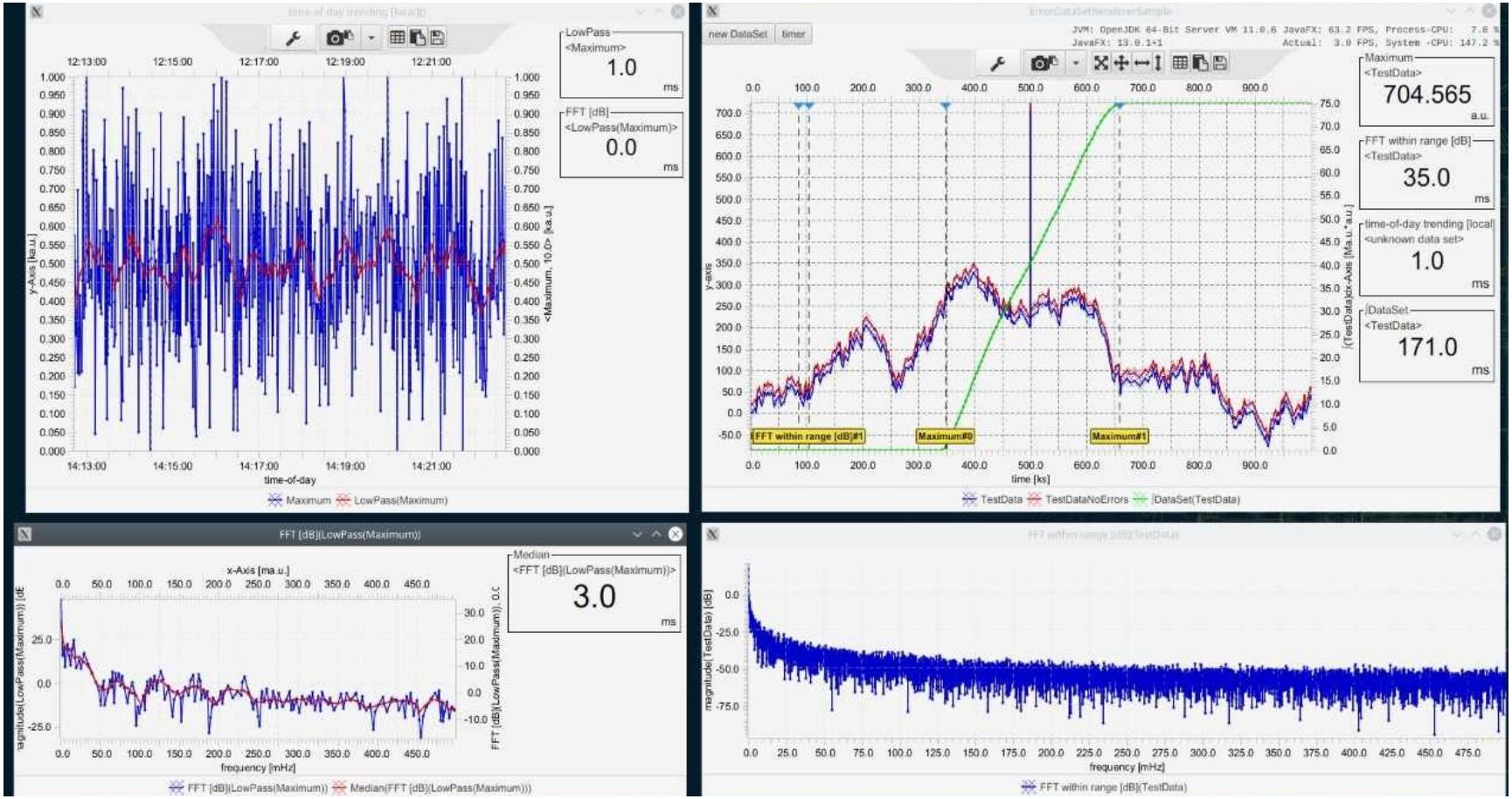
- '*Parameter Measurements*': add expert/rarely used/missing or experimental features on-the-fly




- Importance – puts the operator & machine experts "in the driver seat":
  → evaluate/prioritise features during operation that may eventually enter default implementation
  - benefits: loose-coupling beam-based service provider can focus on acc. numerics/error handling
  - long-term goals: end-users can construct their own WYSIWYG UI w/o requiring string Java/C++ knowledge and also perform simple trims based on these parameter measurements (N.B. we are not too far off from this goal)

# Missing UI Functionality?

## new Chart-fx Options for the Control Room – Para.-Measurements



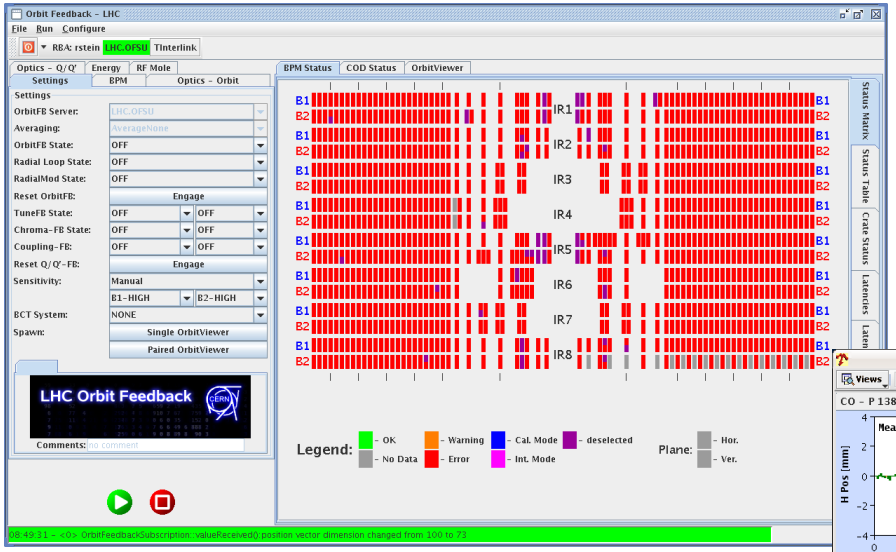

None of the shown functionality (DataSet generation/DAQ aside) needs to (but may) be hard-coded and can be “tried out” before adding this as an official functionality to the OP-approved application.

- nota bene: working on a simple Chart-fx parameter – LSA interface (using REST)

# Top-Level Abstraction: Beam-Based Services – Expert Operations/Debug Tracking

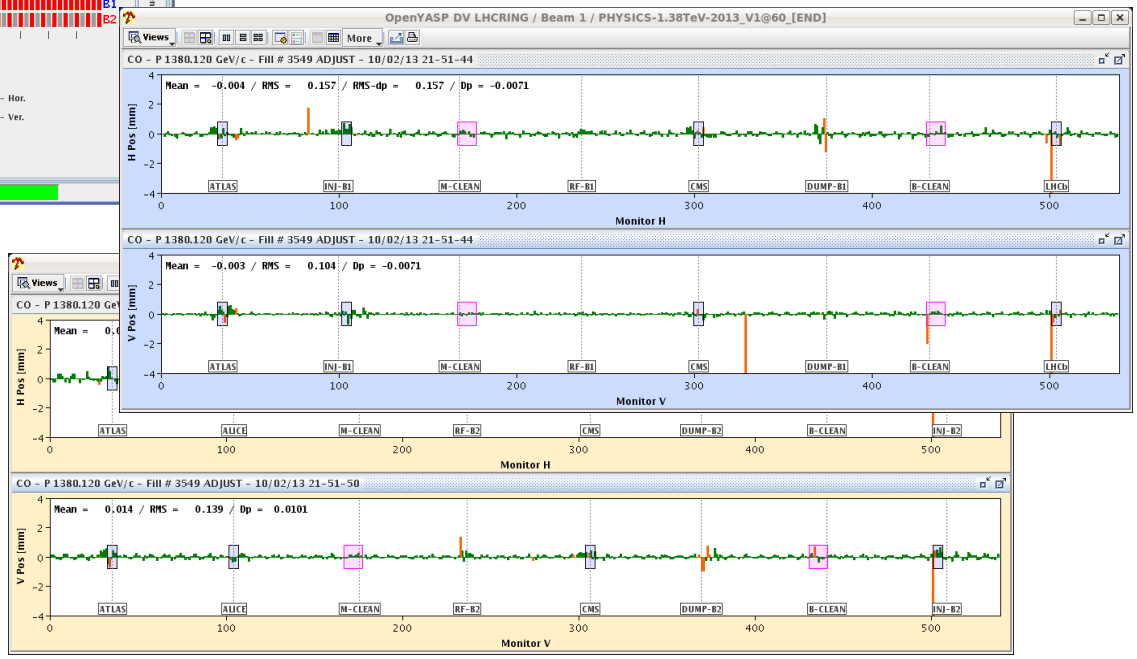
## “Overview of BPM statuses”

ie. if you have an accelerator with more than 12 BPMs



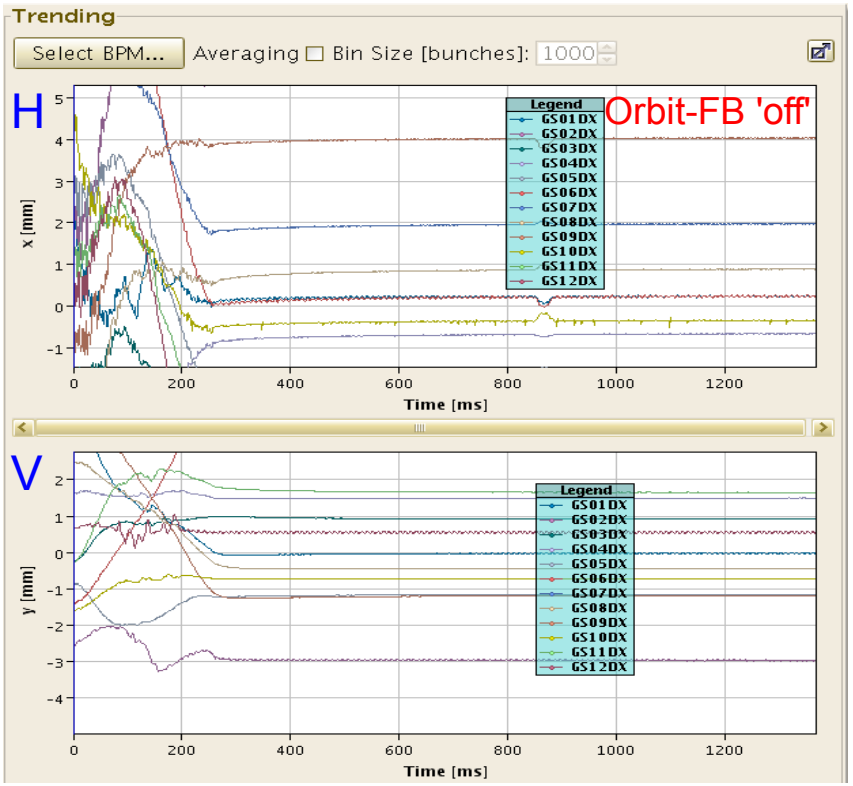

## “Individual Orbit/Trajectory view”

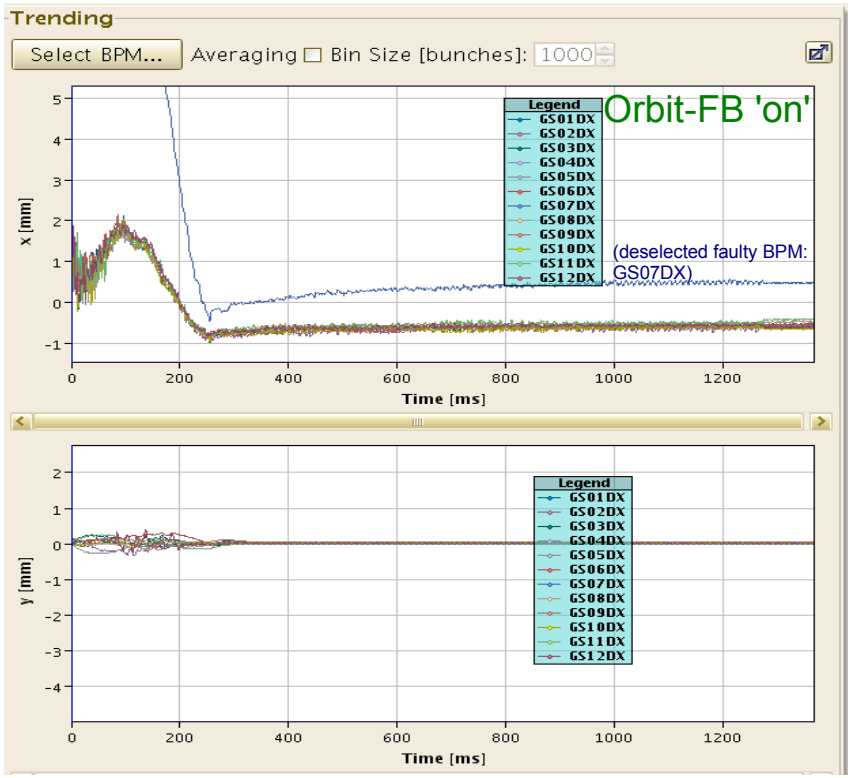
needs to be adapted to FAIR use cases (fast-cycling machines)
→ see Bernd Schlei’s application



# 2016-06-12: Orbit-FB Proof-of-Concept MD

B. Schlei, H. Liebermann, R. Steinhagen

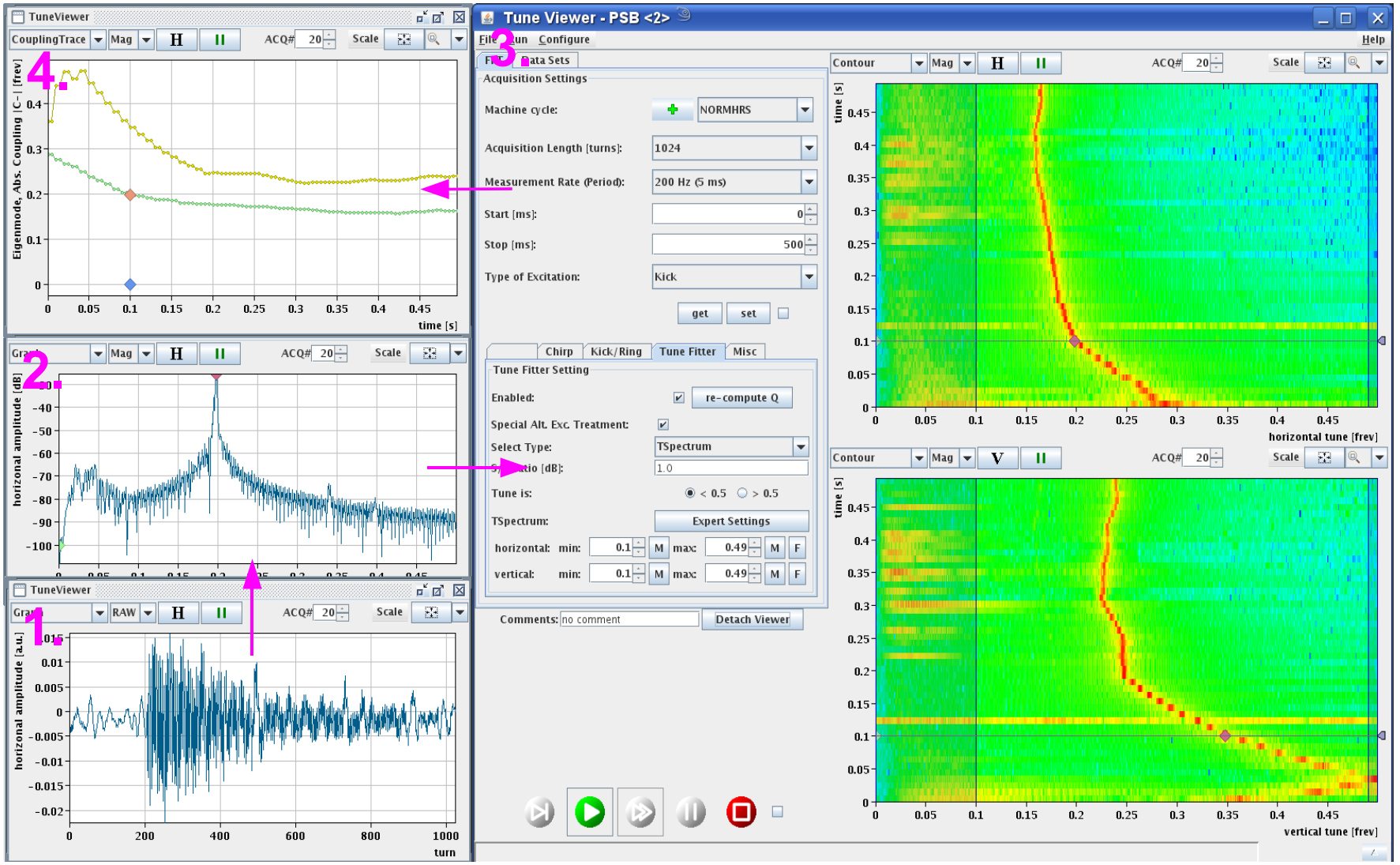



- some workarounds needed, but overall success and results look promising
  - need to follow-up: reliability, performance issues related to CO & BI
    + detailed integration before being put into regular operation
  - N.B. remaining horizontal oscillation due to uncorrected Δp/p mismatch → Radial-Loop/Energy-FB

# Top-Level Abstraction: Beam-Based Services Example Q/Q' Expert Control

here: BBQ-based but API abstraction equally allows BPM data



# List of known Users/Use-Cases - cont
## ...ie. those directly depending on BPMs

| | | | |
|---|---|---|---|
| **A** | **Cycle-to-Cycle Feedbacks** | | |
| 1. | *Injection Steering* | *<orbit & trajectory>* | |
| 2. | *Extraction Steering* | *<orbit & trajectory>* | machine-exp. interface/target steering: rstein+Hades (proto. only) & FRS |
| 3. | *Cycle-to-Cycle Orbit-Feedback (& Radial-Loop)* | *<orbit interface only>* | B. Schlei |
| **B** | **Archiving System** | | |
| 1. | *Regular/OP Archiving* | *<orbit & trajectory>* | |
| 2. | *Post-Mortem System* | *<trajectory interface>* | |
| **C** | **Machine-Protection** | | |
| 1. | *Software Interlock System / MASP* | *<orbit & trajectory>* | |
| **D** | **Optics Measurement** | | |
| 1. | *LOCO-based* | *<orbit & trajectory>* | existing proto. O. Kovalenko (HESR dev.) |
| 2. | *Optics via phase-advance* | *<trajectory interface>* | |
| **E** | **Machine Specific Clients** | | |
| 1. | *Multi-Turn Injection Optimisation (SIS18, CRYRING)* | *<orbit & trajectory>* | A. Krimm |
| 2. | *Collimator/Cleaning set-up (SIS100)* | *<orbit interface>* | |
| 3. | *MPS validation (SIS100)* | *<orbit & trajectory>* | |
| 4. | *Luminosity Steering (HESR)* | *<orbit & trajectory>* | |
| ***F*** | ***Machine-Development*** | | |
| 1. | *bunch-by-bunch expert diagnostics* | *<bunch-by-bunch>* | BI: O. Chorniy H. Bräuning |
| 2. | *machine development specific applications* | *<orbit & trajectory>* | |
| ***G*** | ***Tune (Q) and Chromaticity (Q') Diagnostics*** | | |
| 1. | *Q/Q' control after injection/ramp/pre-extraction* | *<trajectory & bunch-by-bunch>* | |
| 2. | *Q/Q' control during injection (MTI)* | *<trajectory & bunch-by-bunch>* | A. Krimm |
| 3. | *Q/Q' control for (pre-)slow-extraction* | | R. Steinhagen (macro-spill) |

### what we have:

- SVD-based orbit/trajectory control algorithms and most related numeric tools
- UI tools w/ less-boiler-plate
- optics management/interface to LSA → see OpticsViewer
- spec. BPM API (since '19/'20)
- proto. microservice framework

### what remains:

(medium-term, strategic)

- ref. management system
- full microservice framework
- scalling LSA trim performance
- containerisation & farm
- contributions from acc. experts with the required domain knowledge

# Some Open Points w.r.t. BPM integration

- BPM interfaces/aggregates are defined:
  https://edms.cern.ch/document/1823368/
  - Q: status of the implementation – are ESR integration tests/QA completed?

- To be addressed if the services should scale beyond 1-2 machines:
  - reference management of orbit/trajectory/target position
    N.B. control/optics-retrieval algorithms themselves are already solved → see e.g. *OpticsViewer*, *Coat*, ...
    - API needs to be consistent across GSI/FAIR
    - issue: many different “golden standards” and criteria depending on what needs to be achieved, e.g.
      - aperture optimisation, feed-down effect minimisations, ...
      - optics/injection/extraction/other beam physics constraints,
      - *<enter your specific machine goal here>*
      - When to use which ‘golden standard’ when setting-up a new experiment/beam?
  - LSA trim interface: both references & corrections
    - strategic decision to scaling alternative to existing ACID+2PC concept needed,
      e.g. SAGA (choreography) + Event Sourcing (as one of the options)
      ↔ which type/product to use? Definition of appropriate Events and Domain Objects?!?