LHC Post Mortem

Data Transfer from client systems

Markus Zerlauth, Nikolai Trofimov AB-CO

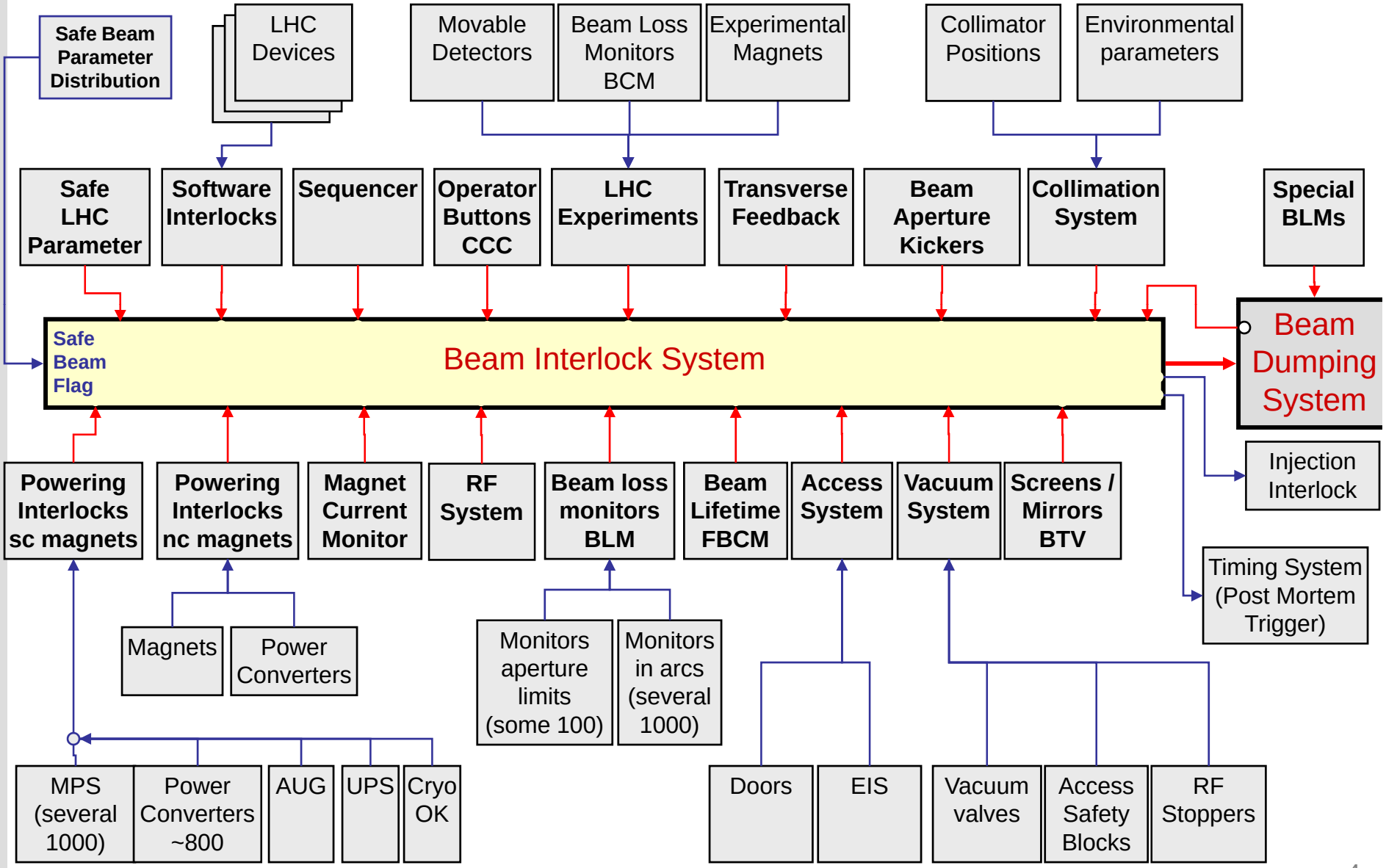
Outline

- The LHC Post Mortem System
  - Main requirements and purpose
  - Infrastructure
  - Client systems during HWC
- Client API and data formats
- Conclusions

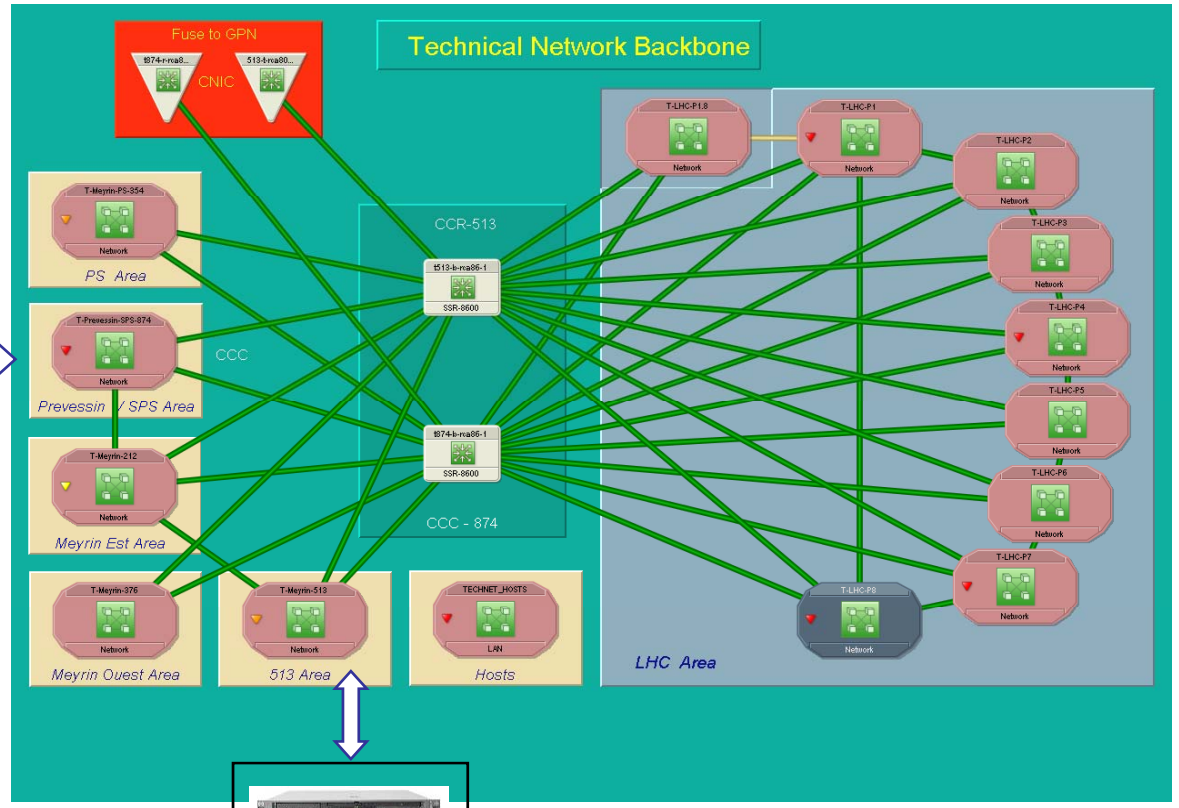
Main Requirements for LHC (beam) Post Mortem:

- Data repository for transient data recordings from equipment systems (LHC Logging for slow acquisition rates)
- needs to reveal cause of emergency beam abort / possible equipment damage to improve operational procedures and protection systems
  - Initiating event
  - Event sequence leading to dump/incident
- Validate correct funtioning of protection systems (redundancy within system, etc..)
- Automated analysis modules in view of systems, complexity and data volume

LHC Machine Interlocks



PM infrastructure – Preparing for the LHC (beam)



n FE servers

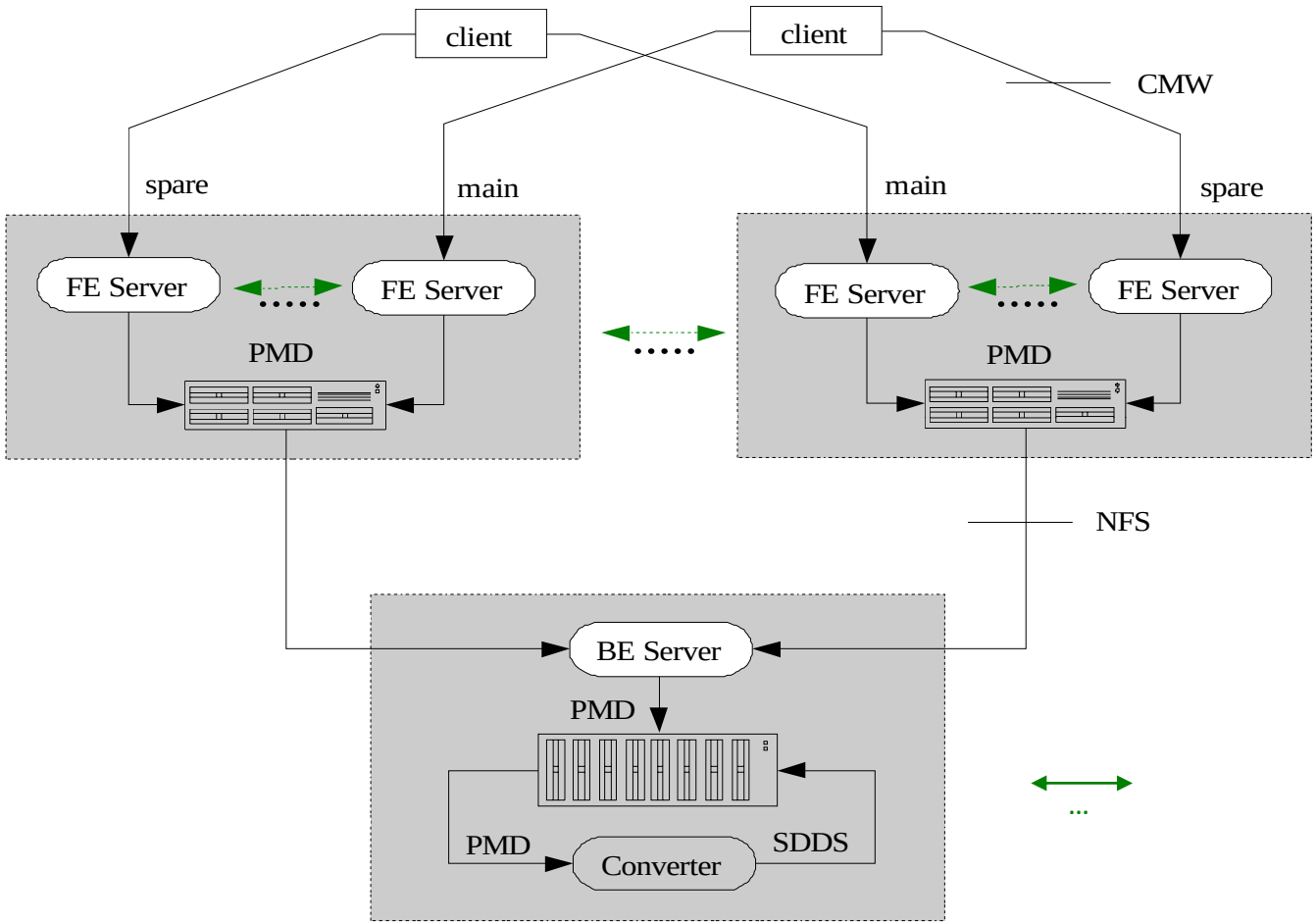
BE server

- Redundant and scalable services in CCR and TCR
- Redundancy for client connections
- Multiple FE servers for primary data storage, BE servers with large disks for complete data image

n FE servers

BE server

PM infrastructure - details



- n front end servers for primary data collection (main and spare server for each client)
- Backend server(s) for complete data image, event building (and data conversion)

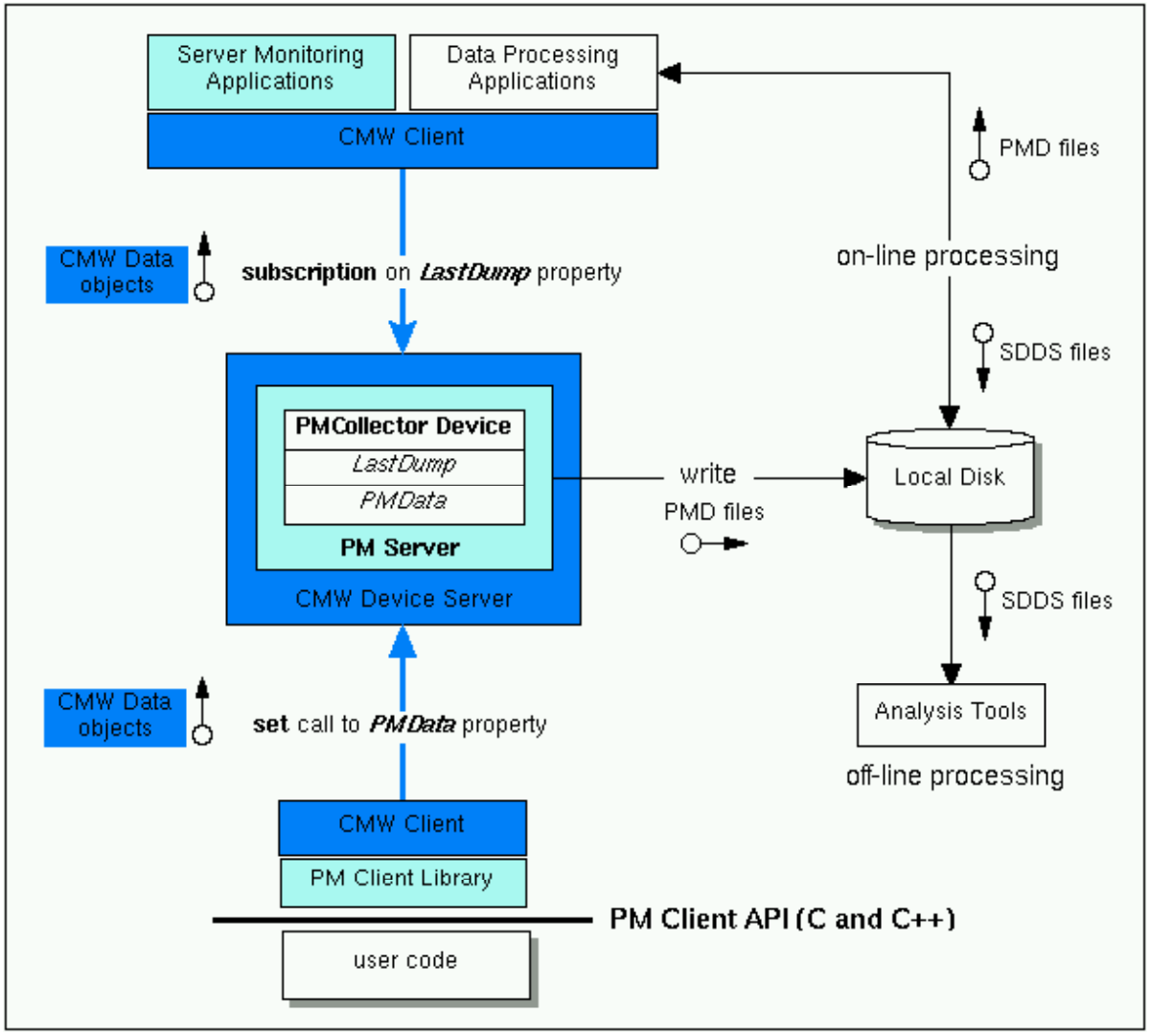
Current clients of PM system

- Post Mortem system already in use during ongoing HWC
  - LHC Power Converters
  - Quench protection system
- First beam client tests ongoing
  - BI (BLM, BPM, BCTs,…)
  - FMCM
  - Collimators, RADMON?
- Two main possibilities for sending data
  - FESA (mainly used by VME based systems)
  - C++ client API (used by Power PCs / gateways)
- Recommended data format .pmd

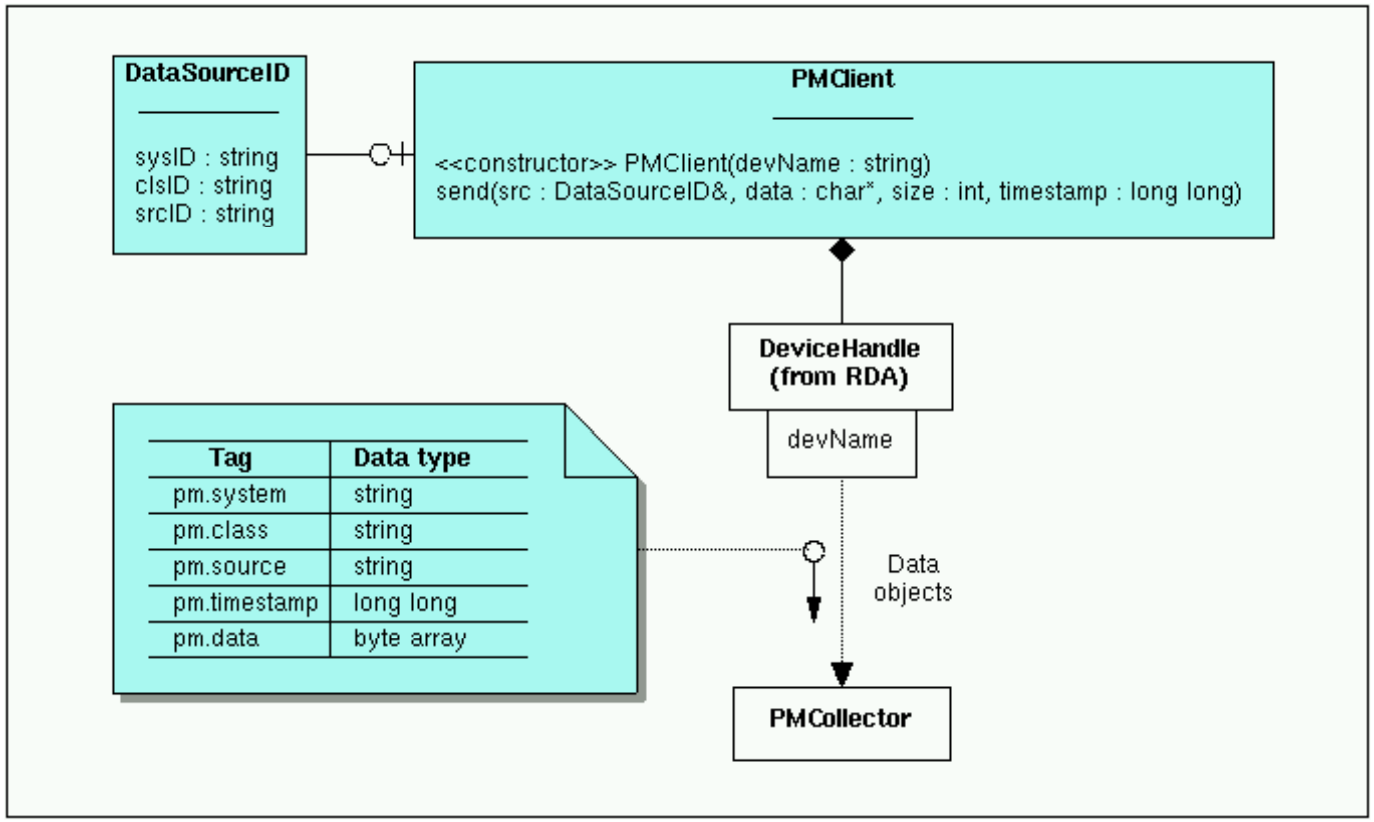
- Two possible trigger mechanisms for sending of PM data used by equipment systems
  - Self triggering of device (e.g. internal fault of power converter)
  - External trigger (PM event being sent via GMT – HX.PM1-CT, HX.PM2-CT)

PM client API

➔ Client system is preparing data file and calls the client API, which will take care of the data transfer to the PM system



.pmd data format

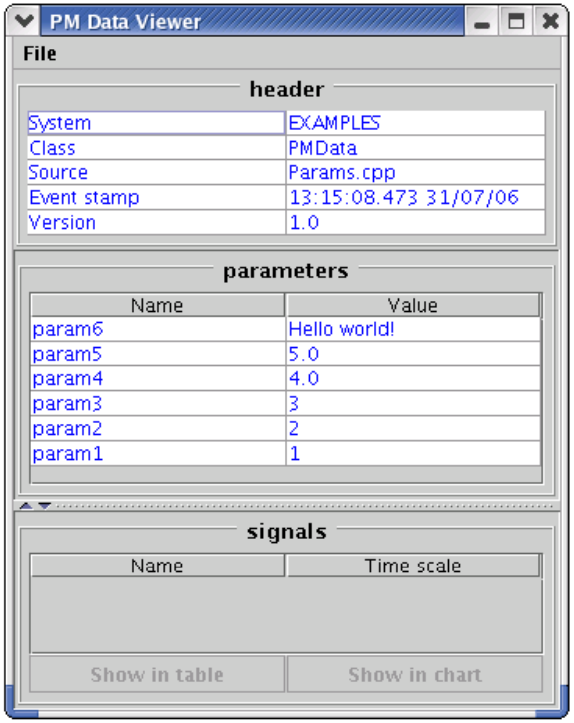


➔ Data file contains some header information (system, class, source, timestamp) and actual pm.data

– Timestamp = time of self-triggered PM acquisition or global PM event from the LHC timing system (important for correlation with beam event!)

PM Data structure - Example

- .pmd data can contain many different data objects
  - Parameters
  - Signals
  - Descriptions
  - Units
  - Enum Labels
  - BitLabels
  - 2 dim Arrays,...

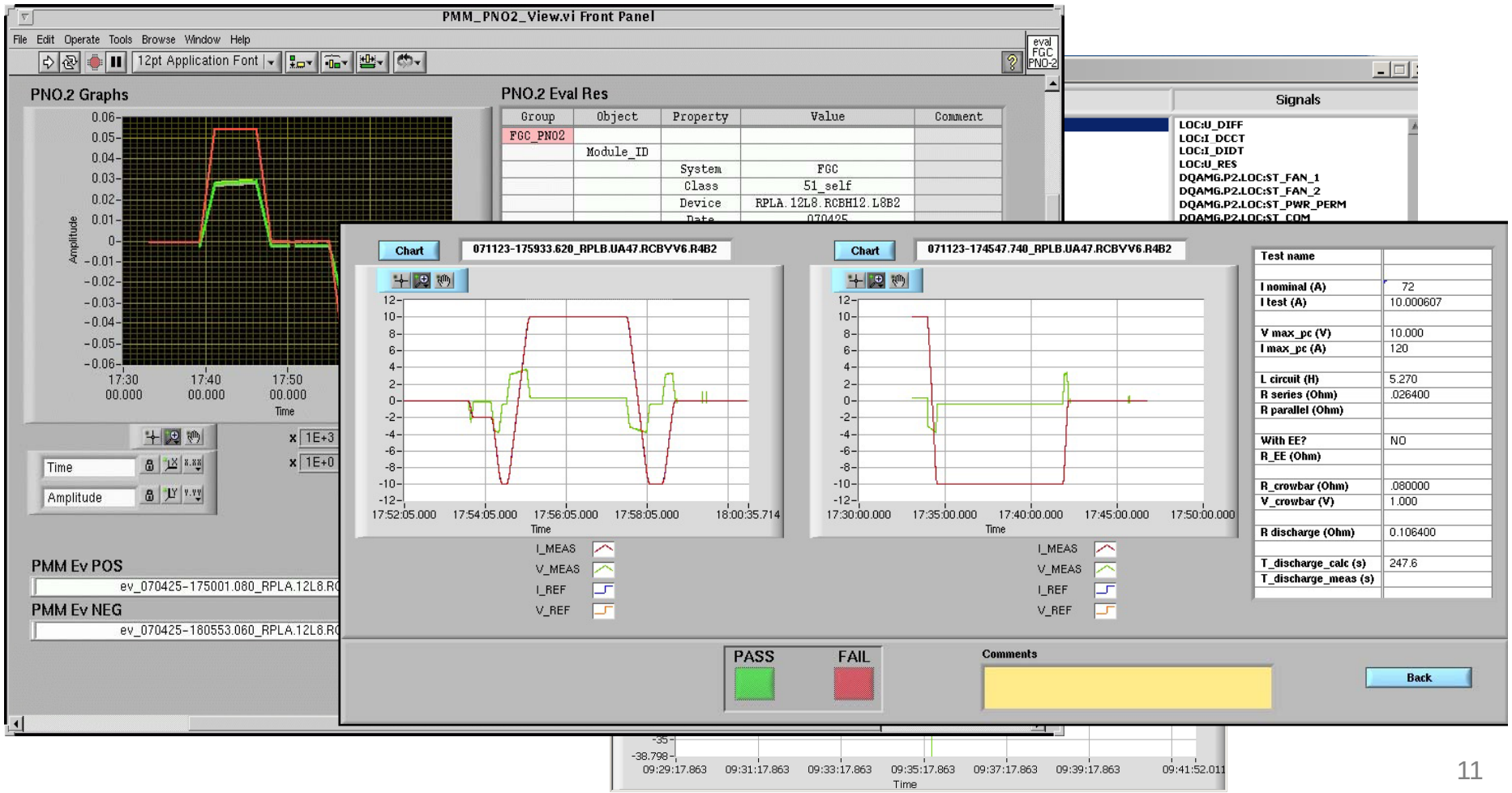


*The output file view in the **PM Data Viewer***

```
#include <iostream>
#include <pm/PMError.h>
#include <pm/PMData.h>

using namespace std;

//
// The PM data object and default output file
//
PMData pmdata("EXAMPLES", "PMData", __FILE__);
const char* fileName = "example.pmd";

//
// User variables that need to be exported to the PM system
//
char   p1 = 1;
short  p2 = 2;
long   p3 = 3;
float  p4 = 4;
double p5 = 5;
char   p6[80] = "Hello world!";

//
// Exports user variables to the PM system
//
void registerVariables()
{
   pmdata.registerParameter("param1", p1);
   pmdata.registerParameter("param2", p2);
   pmdata.registerParameter("param3", p3);
   pmdata.registerParameter("param4", p4);
   pmdata.registerParameter("param5", p5);
   pmdata.registerParameter("param6", p6);
}

int main(int argc, char** argv)
{
   if (argc > 1) fileName = argv[1];

   try
   {
      registerVariables();
      pmdata.write(fileName, PMData::timeNano());
      cout << "PM data written to file: " << fileName << endl;
   }
```

Advantages of using .pmd format

- Existing data converter to SDDS
- Existing Data viewers
- No conversion/interpretation file necessary



Conclusions

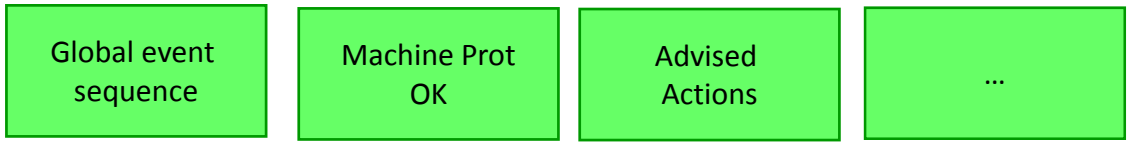
- Standardised means of data transfer to PM system via client API
- Some uncertainties for experiements to be looked at in detail (platform diversity > , RDA, TN and CMW dependencies)
- Details on client lib and examples for .pmd data format
  - General: http://wwwpsco.cern.ch/private/mw/RDA/pm/html/
  - PMData formats: http://wwwpsco.cern.ch./private/mw/RDA/pm/tmp/html/
- In case of further questions don't hesitate to contact us (Nikolai Trofimov, Markus Zerlauth)

Thanks a lot for your attention - Questions?

Client System	Platform	Data interface	Data format	Data sources	Devices	Self triggering	External Trigger
LHC Power Converters	Power PC / gateway	PM client lib	.pmd + conversion	~70	~1700	Y	Y
LHC Quench Protection System	Power PC / gateway	PM client lib	.pmd + conversion	~30	~900	Y	
FMCM	VME	FESA	.pmd	12	26	Y	Y
BLM	VME	FESA	.pmd	25	~3000		Y
BPM	VME	FESA	.pmd	64			Y
other BI	VME	FESA	.pmd				Y

Data flow and analysis

Global PM Analysis: Global Event sequence, summaries, advised actions, event DB,…



Individual System Analysis/Checks: Validation of machine protection features, pre-analysis of PM buffers into result files, flagging of interesting systems/data reduction, database catalogue

Data completeness and consistency check at system and global level (minimum data, configurable)

Upon beam dump / self triggering, systems start pushing data to PM system, Logging, Alarms, etc…