

# Possible COMPASS software improvements during LS2

Antonín Květoň

February 12, 2019

# Something to keep in mind

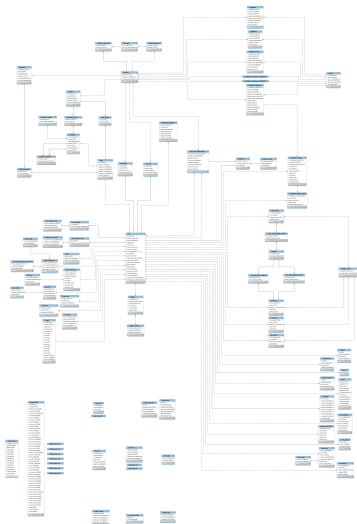
- Some of the proposed projects are just nice to have features
- Our manpower is limited and we would work on this alongside the DAQ software upgrade → can't have them all (at least during LS2)
- The goal of this talk is to start a discussion to assess the priority, feasibility as well as the **realistic practical merit** of these projects

# A small user cluster

- Users often launch CPU/memory-intensive applications on the pccogw or pccorc machines (mostly on the gateway)
- Often results in a disruption of service
- Proposal: acquire several (4-5) servers with powerful CPU's (8+ physical cores per server), install a distributed load balancer, and allow users to run their applications there
- The cluster would take place of the gateway → `ssh pccogw00`
- This is one of the easiest projects to implement, but requires new hardware

- SLC6 to be EOL'ed on November 30, 2020
- Upgrading to a new Linux distribution is a de-facto necessity and will allow us to obtain newer compiler versions (C++ 2011 standard, Qt..)
- The question is when to do it
- CC7 EOL : June 6, 2024
- CC7 has been used throughout CERN for a decent amount of time and any major bugs have already been fixed
- CC8 is in beta, do we wait?
- ⚠ We have to decide on a Root version to use long-term! ⚠

- Database not normalized, lacks relationships, constraints and DB-side referential integrity protection → a disaster waiting to happen (especially with the ongoing development of applications connecting to this DB)
- Overhaul concept created by Matouš – new DB engine, referential integrity, normalization... original structure preserved to a degree
- Goes hand-in-hand with the need for development of new Start/End of Run scripts – already being worked on by Lucie. Without this rework, a lot of problems would be foreseen after the OS upgrade.
- What's left to do: conversion logic and adjustment of the tools accessing this DB



- This DB also contains several PT tables – this rework would not include these tables (but would move them to the new DB)
- We still may need to analyze some more relationships in the data to perfect the schema
- The biggest question is whether there are any critical applications or tools accessing this DB that we do not know of.

# Old logbook phase-out

- New logbook well out of the testing phase
- Wasteful to maintain two versions at once
- Many new features added to the new version since the initial release
- This is something that should be done regardless of whether the logbook DB is reworked
- Requests for new functionality are very welcome, best to be implemented and tested before the phase-out



- Good data often lost or spoiled due to human error
- Ever-present ongoing red alerts lead to desensitization of users and result in general apathy towards actual alerts
- Tutorials, documentation either outdated, large in volume, or hard to find
- Solution: for the former, take the science of HCI (Human-computer interaction) as well as psychology, into account
- For the latter, development of a unified problem documentation application is in order

# Unified problem documentation use case

- 1 A red or orange alert appears (DAQ GUI, MTV, Message Browser, DCS)
  - 2 The user mouseovers the alert, a link to a webpage appears
  - 3 Upon opening the webpage, details concerning this alert and how to handle it are present
- Default action for warning(orange)-type alert = "Make an entry in the logbook", default action for error(red)-type alert = "Call the expert" → motivation for experts to fill in the info!
  - An easily-searchable, intuitive web interface would also be part of this (possibility to search by, e.g. detector or srcid)

# An unified connectivity solution

- Problem: experts often need to connect remotely for control/monitoring, but ssh -X slow and inresponsive.
- Solution: an unified framework for publication of COMPASS GUI's to a website
- We have a concept solution for DAQ-related software (DAQ GUI, MTV, COOOL)
- Definition of these GUIs in QML would allow for unified compilation for both web and desktop deployment
- This involves a rework of MTV/COOOL into DIALOG/Qt
- The question remains what to do about DCS – the most important and therefore the most frequently accessed system

- No normalization, no referential integrity, many columns supposed to be foreign keys are instead strings.
- Database Manually partitioned
- Problems with e.g. version control – modification date sequencing being incorrect resulted in incorrect information concerning port state (e.g. a port appearing as disabled for several weeks while in reality, it was enabled)
- Snowballing effect of bad practices → low efficiency of developers (and experts during operations)
- Compatibility/interface usability issues with new Hardware (e.g. new the new iFTDC for CEDAR and MWPC) – 64(65) channels → not trivial to insert all configurations to the DB.

# Frontend DB – some screenshots

Table						
Detector	Configuration	MODULE_program	TCS_program	ROB_slot	CableName	
<input type="checkbox"/> FILES						
<input type="checkbox"/> FILES_PROT						
<input type="checkbox"/> FRONTEND						
<input type="checkbox"/> FRONTEND_1230ns						
<input type="checkbox"/> FRONTEND_2012_06_10	GEM	NULL	HotGeSiCA RICH standard	TCS ID APV4RICH	none	TS3-2 9
<input type="checkbox"/> FRONTEND_2012_10_16						
<input type="checkbox"/> FRONTEND_20111031						
<input type="checkbox"/> FRONTEND_20120405	GEM	NULL	GeSiCA 2006	TCS Receiver V2.01 (RBT)	none	NULL
<input type="checkbox"/> FRONTEND_bak						
<input type="checkbox"/> FRONTEND_begin2010	GEM	NULL	GeSiCA #5 2006 test	TCS Receiver V2.01 (RBT)	none	NULL
<input type="checkbox"/> FRONTEND_from_backup						
<input type="checkbox"/> FRONTEND_GOOD_20120521	GEM	NULL	HotGeSiCA RICH standard	TCS ID APV4RICH	none	TS3-2 13
<input type="checkbox"/> FRONTEND_OLD_20120520						
<input type="checkbox"/> FRONTEND_primakov2009	GEM	NULL	GeSiCA 2006	TCS Receiver V2.01 (RBT)	none	NULL
<input type="checkbox"/> FRONTEND_PROGRAM						
<input type="checkbox"/> FRONTEND_sv20111027						
<input type="checkbox"/> FRONTEND_test870ns	GEM	NULL	GeSiCA 2006	TCS Receiver V2.01 (RBT)	none	NULL
<input type="checkbox"/> FRONTEND_test963ns						
<input type="checkbox"/> FRONTEND_test993ns	GEM	NULL	GeSiCA 2006	TCS Receiver V2.01 (RBT)	none	NULL
<input type="checkbox"/> FRONTEND_test1213ns						
<input type="checkbox"/> FRONTEND_test1459ns	SciFJ	FEOR_RAM F1DEBUG	catch-2008	TCS Receiver V2.01 (HEX)	MUX06_12	TG-U 13

- A complete rework from the ground up comes to mind
- Definitely the most ambitious project
- A large number of systems accesses this DB, the first step would be identify all of them
- Requires collaboration-wide cooperation
- Development very likely to last until after LS2, potentially only being deployed at the start of the first AMBER run

- 1 A small user cluster
- 2 OS upgrade
- 3 Logbook DB
- 4 Old logbook phase-out
- 5 Unified problem documentation and general CR usability
- 6 An unified connectivity solution
- 7 Frontend DB