



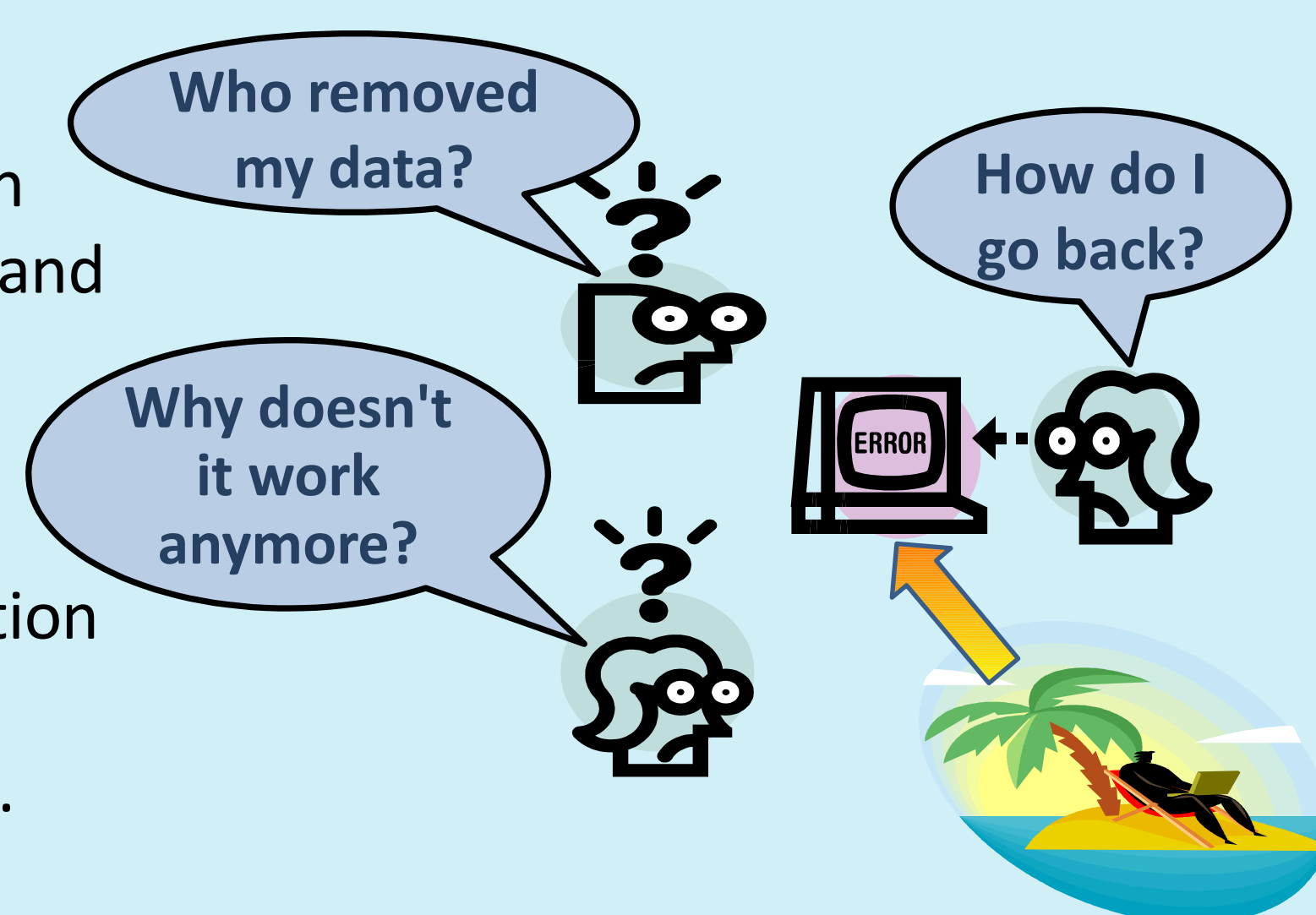
The version control service for ATLAS data acquisition configuration files



Igor Soloviev • University of California, Irvine, US • isolov@cern.ch

Why do we need version control service for files?

- There are hundreds of megabytes of DAQ-related configuration information stored in thousands of OKS^[1] XML files updated in parallel by many system and detector experts.
- From time to time there were problems caused by XML syntax errors or inconsistent state of files from the point of view of the overall ATLAS configuration. That was not always possible to know who made a modification causing problems or how to go back to a previous version of the file.
- Write access control of files based on UNIX groups was not flexible enough.

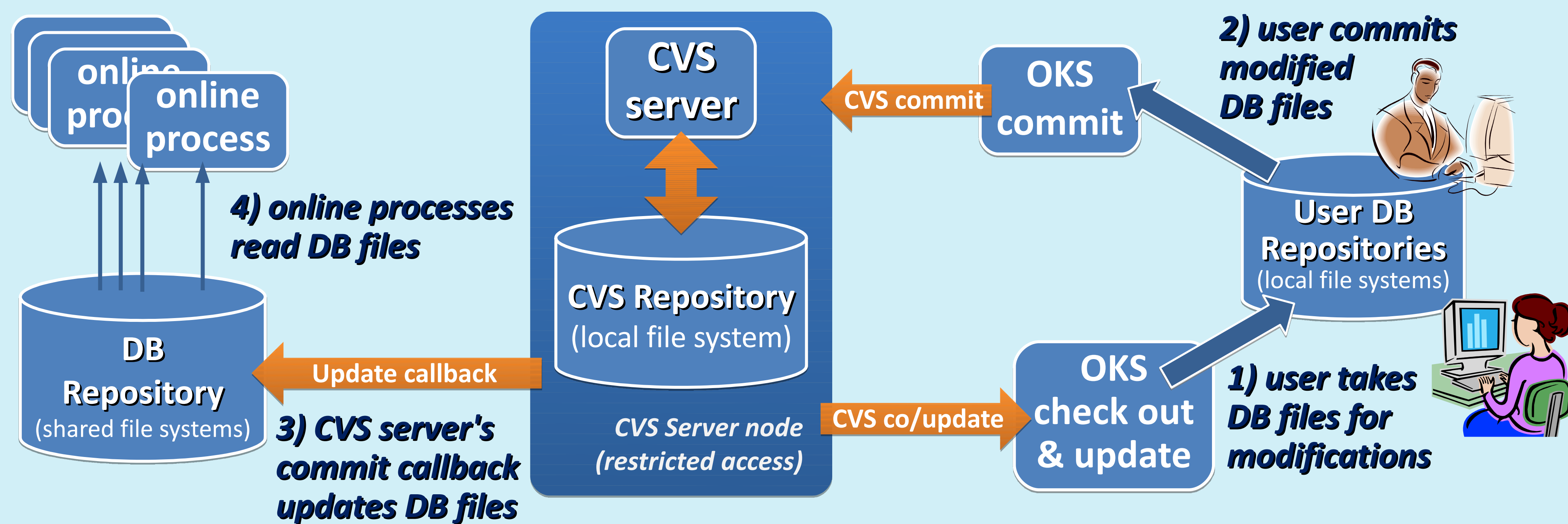


2008: Requirements

- The files stored in a repository have to be consistent at all times, i.e. they may not have syntax errors or dangling references on other files and data.
- It has to be possible to know who made modifications in a file and when, and to access any previous version of a file to see differences or to recover changes.
- The Access Manager tool responsible for ATLAS users authorization in the ATLAS online system has to be used to check update permissions on files and directories in the repository.

Implementation

From the above requirements it was clear that direct write access to the repository by experts had to be revoked. Instead an update of the repository should be performed from a single point by a process running under a privileged account on a dedicated node. It was decided to implement a utility validating the changes and applying them to the repository (named "OKS Server"), and to use popular Concurrent Versions System (CVS) to provide version control of every file modification.

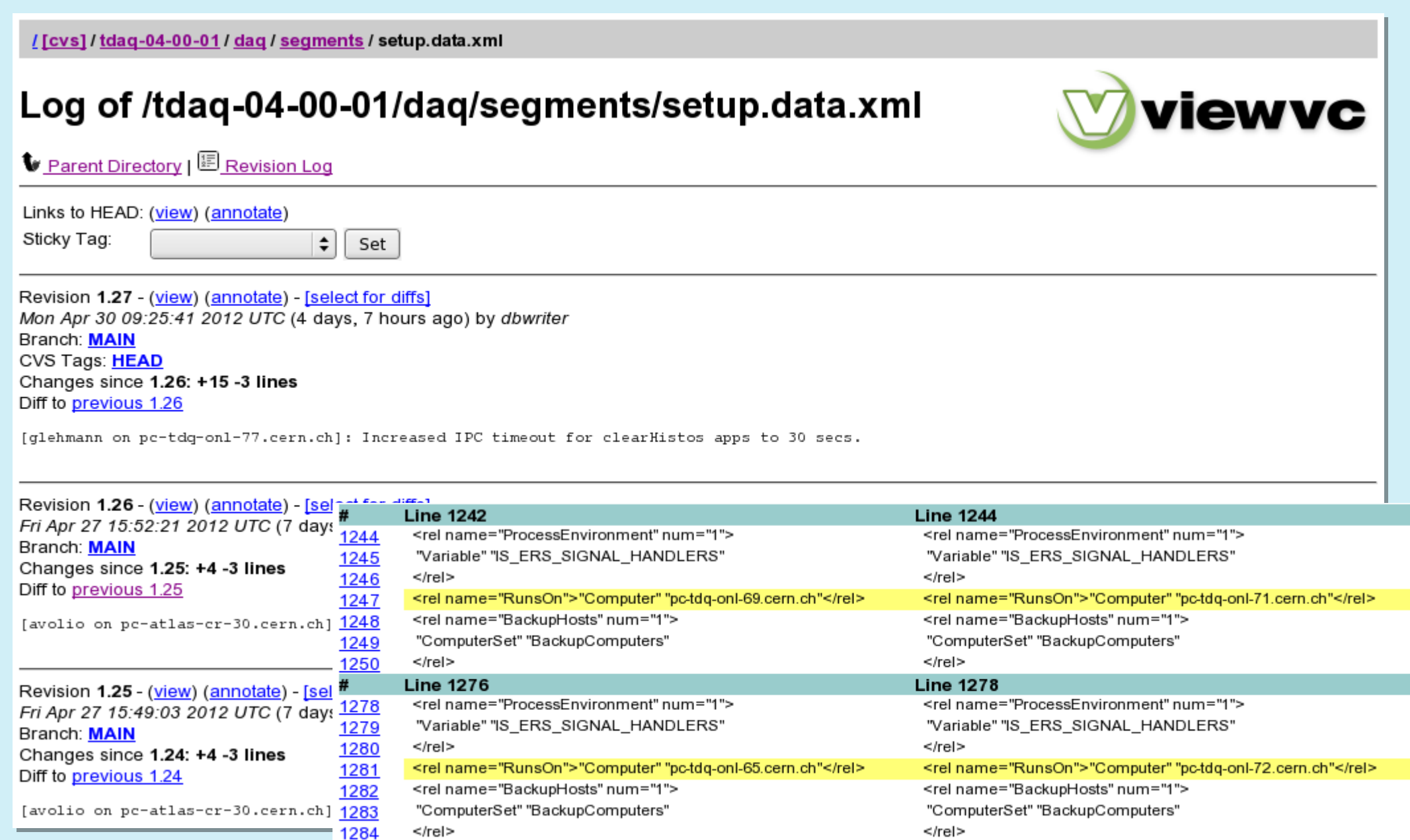


The architecture of the OKS Server is based on CVS server usage

- The user creates or checks out files into a private database repository, makes modifications and commits the files on CVS (using special OKS commit utility), and then the CVS server's callback updates database repository.
- The CVS server is running on a dedicated node (accessible by administrators only) under privileged account allowing to modify DB repository files. The file system volume storing OKS DB repository is mounted read-write on the CVS server node and is mounted read-only on all other nodes.
- The OKS commit performs consistency checks of updated files and user's authorization on the Access Manager service. If the consistency check is passed and the user is authorized for changes, the OKS commit performs necessary steps for CVS user authentication and passes updates to CVS server. The OKS commit has execution sticky bit to be started under privileged account able to get CVS server commit authorization that protects OKS server from CVS commits bypassing OKS validation.

OKS Server Interfaces

- The application programming interface (API) to the OKS Server is integrated into all configuration service packages and is transparent to all applications reading or modifying the database.
- The user programming interfaces include three groups of tools:
 - The command line utilities are similar to CVS ones: a user can check out XML files into private repository and modify using any tools including text editors; the updated files can be committed on the OKS server using the OKS commit script as well as updated from the OKS Server or released.
 - The graphical database editors allows user to modify objects stored on OKS server.
 - The ViewVC Web interface allows to see who, when and why changes were made in OKS Server files. One can download any previous version of a file or visualize differences between versions.



Experience in ATLAS since 2008

- No OKS Server errors or problems** with inconsistency of the data stored on it **have been observed**.
- The OKS Server manages more than **1300 XML files for ATLAS data taking configuration** and their total size exceeds **160 megabytes**. The overhead introduced by the OKS Server to commit changes in such configuration is typically **less than 5-10 seconds** including validation of changes, commit to CVS server, and updating of the files on the shared file system.
- This or a similar technology is advantageous wherever in all areas where human readable data files are writable by several experts and their consistency and traceability of modifications are key requirements for functionality of a system.

[1] Soloviev I, Jones R, Mapelli L and Ryabov Y "The OKS persistent in-memory object manager" 1998 IEEE Trans.Nucl.Sci. 45-4-1, 1958-1964