Your benefit with PCV 2.0

- very little implementation effort
- transparent implementation using filter views on database level
- Customizing to limit data visibility also possible on DB-level (Oracle Virtual Private Database)
- high flexibility, expandability and adaptability
- extremely high compatibility with Agile PLM e6 standard
- the use of the role model is not a prerequisite, but will be used internally invisible and can be used later on to deploy the role model
- extremely little ongoing maintenance and administration effort

Service

about 5 - 10 days for implementation plus once effort to create initial data for visibility of data if necessary

Prerequisits

- min. Agile PLM e6.0
- Oracle Datenbase min. 9.x, for DB implementation via Oracle Virtual Private Database min. 10.2 (Enterprise Edition required)

References

Artec, Balcke Dürr, framatome, SPX Cooling, ThyssenKrupp Presta, Skidata, Werner Turck, Wittenstein

Additional information is available at:

ots
Unternehmensberatung GmbH

Duftbachweg 8 1/2
83471 Berchtesgaden, Germany

Geschäftsführer
Dipl.-Ing. Jochen Kaiser

Tel +49 (0)8652 / 974 370 - 0
Fax +49 (0)8652 / 974 370 - 9
Internet: www.o-t-s.de

PCV 2.0: protects your intellectual property in Oracle Agile e6

Efficient and easy protection for your precious data against unauthorized access and leaking

organisation
technologie
systeme

More and more sensible data nowadays is stored in a PLM-system. So the protection of intellectual property (as one can name this data) becomes more and more important, esp. for global and/or project specific work also with external partners. Therefore a functionality in your PLM system is necessary, which is efficient enough to fulfil the typical requirements and also causes just a minimum of implementation effort as well as ongoing work for administration and support. If the Agile standard module (MPAR/MOAR) is to heavy and complex and other implementations do not fit your requirements then PCV 2.0 (project controlled visibility) could be exactly the solution of choice for you.

The goal is not only to control the access rights based on projects. The users also shall only see the data which is qualified via a relation to an access object (in this case a project). This makes it possible to "give" certain data to the users and also to encapsulate other projects, so that only project members can see the data. The users daily work shall be as less affected as possible and may not obstruct the users even if they are working for several projects. Of cause it shall be a fast and easy solution as well as causing only a minimum of additional effort.

Short description

Based on a typical scenario, where the project is used as the central element to control visibility, we exactly used this object to implement the PCV functionality. But visibility control is not restricted to a typical project: you also can change the definition to a "visibility object" and fulfil other requirements or even mixed environments. It is also possible to enhance the normal project relation used for visibility control to meet additional requirements for confidentiality - this enables even several different levels of visibility.

The visibility is inherited over the product structure (P1 - P3) as well as over the product structure (A1 - A15 and D1 - D16) - making all data (e. g. projects, items, documents) within the whole structure automatically visible even if there exist a project relation only on root level.
Users are separated into restricted (U1 - U3) and unrestricted (U4) users with restricted or unrestricted data visibility. Thus "internal" users could be unrestricted without changed visibility, whereas "external" users would be restricted with limited visibility.
At login the type of user and therefore the data visibility is determined and stored to be used throughout the whole session; it is even possible to reload the visibility definition of the user in case of some major changes - to be used without re-login. This is also useful for batch processes running with a special user: it is now possible to switch the visibility to the user of the job currently processed.
A user with restricted visibility only can see data related to "his/her" projects, directly associated or via inheritance within project- or product-structure.

gedacht. getan.

PCV 2.0: protects your intellectual property in Oracle Agile e6

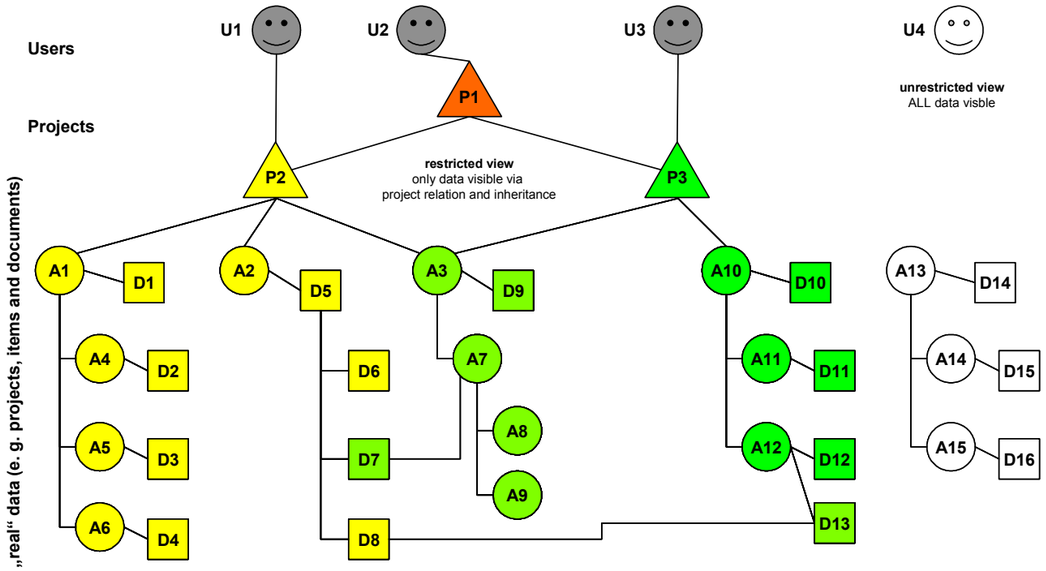
Efficient and easy protection for your precious data against unauthorized access and leaking

Technology

The functionality is provided by filter views on database level which are joined in PLM masks according to the user definition. These filter views contain all the relations between project and "real" data and other filters as well as inherited records.

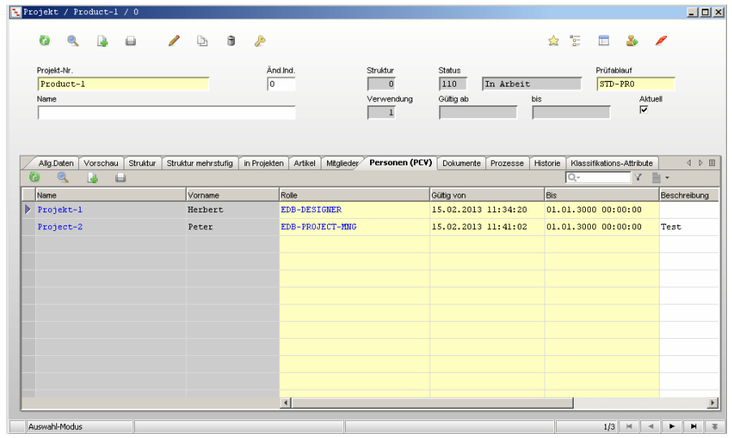
For customizing the data masks only 2 PCV LogiView-trigger are necessary (one as pre-action in all masks and another as pre-field at the key-field in relation masks only). If using Oracle Virtual Private Database only the definition of the corresponding database triggers is necessary (they are already shipped); then there is no customizing necessary in PLM to limit visibility in data masks.

Customizing for administrating the visibility records is necessary in all cases - a standard customizing is shipped. Also shipped and installed is the visibility functionality for projects, items and documents. This can be adopted and enhanced for other objects; in general the functionality can support all objects.



Because only standard PLM elements are used, the compatibility of this solution with PLM standard is extremely high. It is even possible to work without PLM role model although the corresponding tables are used internally in PCV; therefore they can be used later on as base for a role implementation. The visibility definitions are the very basic access rights within a project ("see" or "read"). Standard masks for maintaining this information are shipped with PCV.

The transparent implementation via filter views on database level on one hand makes support for PCV very easy. On the other hand adoptions and enhancements are also very easy to implement. It is for example quite easy to implement an additional confidentiality filter (esp. for document data) or customer specific rules to stop the inheritance of visibility. Also other customer specific filters are possible.



Because the product structure, which is maintained anyway, is also used to indirectly drive the visibility for PCV only very little additional effort for administration and control is necessary. It typically consists of adding the users to the corresponding project. With PCV we ship masks and functions so that a "normal" project manager can do this - even via drag&drop from the list of persons. This is a normally a part of duty for "staffing" a project.

Associating the "real" data to the projects can be done on root level - this is already enough. Thus it is normally not necessary to create additional (direct) relations from the "real" data to the project it shall be visible - this "happens" automatically as soon as the newly created record is used in the product structure. Of cause it is possible anyway to add additional project relations (also for released records) in order to control visibility on a very detailed level. Normally there are no data model changes necessary because project relations already exist for all main objects in PLM. Because relations are used instead of fields there is no limitation how many different visibilities you can define per data record.

For operation there are no background processes necessary to correct or maintain the visibility data - it is maintenance free.