

Version: 1.0

Author: Stephan
Toeppen



Architecture Guideline for PLM Core

Subtitle

History

Version	Status	Date
1.1	Review version	26.04.2007
1.0	Send out for review	20.04.2007

Storage location for temporary versions of this Document: \\server

Contents

1	Revision Log	4	
2	Stakeholders	4	
3	Key Findings / Recommendations / Open Items	4	
4	Architecture Goals and Guiding Principles	5	
4.1	Architecture Goals	5	
4.2	Boundary conditions and decisions	5	
4.3	Target Architecture	7	Deleted: 6
4.3.1	Architecture building blocks	8	Deleted: 7
4.3.2	PLM Core UI	8	Deleted: 7
4.3.3	ERP access and BO layer	8	
4.3.4	Logic layer	9	Deleted: 8
4.3.5	Enterprise SOA Approach	9	Deleted: 8
4.4	Overview on Architecture Topics	9	Deleted: 8
4.4.1	Handling of ERP DDIC elements	9	Deleted: 8
4.4.2	UI development approach	10	Deleted: 9
4.4.3	RFC UI communication	10	Deleted: 9
4.4.4	Authorizations for collaboration	12	
4.5	List of ERDs	12	
5	Technology Decisions and Constraints	13	Deleted: 12
5.1	Technology Selection	13	Deleted: 12
5.1.1	Development environment	13	
5.1.2	Floorplan manager	13	
5.1.3	Collaboration decisions and constraints	15	
5.1.3.1	Comparison and aspects of DMZ alternatives...	Error! Bookmark not defined.	
5.2	Prohibitions for technology issues	16	
5.3	Exceptions for technology issues	16	
6	Reuse Decisions and Constraints	17	
6.1	Selection and Prohibitions for reuse issues	17	
6.2	Exceptions for reuse issues	17	
7	Software Layers and Components	17	
7.1	Software Layers	17	
7.1.1	PLM installation scenarios	19	
7.2	List of Components	19	
7.2.1	Core PLM packaging	21	Deleted: 20
7.2.2	Own Application and Software Components	21	
7.2.3	Non-SAP Software Components	22	
7.3	Deployment Scenarios	23	

7.3.1	PLM Deployment.....	23
7.3.2	PLM release cycle	24
7.3.3	PLM Core and ERP Enhancement package dependencies	25
7.3.4	PLM Core development constrains.....	25
7.3.5	ERP cross demand handling.....	25
7.3.6	IPPE ERP2005s dependencies	27
7.4	Constraints	28
8	Relations to Standards and Guidelines	28
8.1	Related Program Architecture Guidelines	28
8.2	Impact from SAP architecture adoption guidelines	28
8.3	Product Standards.....	28
9	Development landscape.....	29
10	Glossary	29

1 Revision Log

Version	Date	Who	Remarks

2 Stakeholders

Name	Role
Stephan Toeppen	[Author]
Gregor Rieken, Thomas, Moritz; Bohnengel, Tilman; Haehner, Peter; Iselborn, Bernhard; Morrison, Iain; Walter, Wolfgang; Bauerndistel, Karin; Hausen	[Involved Architects]
Stephan Többen	[Program Architect]
Marco Bettenhausen	[Program Lead]
Hans Thalbauer	[Solution Management Program Lead]
Bachmann, Martin; Voigt, Michael	[Involved SM]
Fereshteh Fakhar	[QPL]
Ernst Berger	[Component owner]
Andreas Linke	[DLL, DSS]
Tobias Stein	[Lead Architect SCM/PLM/MAN]

Formatted: German (Germany)

3 Key Findings / Recommendations / Open Items

Key Issues / Recommendations	Comments / Consequences

4 Architecture Goals and Guiding Principles

4.1 Architecture Goals

1. Significantly improve usability and user experience
 - Improved UI acceptance
 - Improved perceived UI performance
 - Easy and flexible enhancement concept for user interfaces
2. Smooth integration to existing PLM and ERP solutions
 - Re-use existing ERP objects as platform for PLM core to limit effort and lower TCO
 - iPPE as backbone for SAP engineering solution (mid term)
3. No destabilization of ERP
 - Create exposed Business Object Layer
 - Encapsulate business logic with services
4. Collaboration with direct access (in addition to existing cFolder scenarios)
 - Separation of presentation layer and backend/data
5. Follow SAP's roadmap for enterprise Service Oriented Architecture
 - Allow partners to create composite applications based on PLM platform

4.2 Boundary conditions and decisions

In addition to the existing (and still valid) cFolders based collaboration scenarios and use cases a collaboration with direct access shall be supported.

1. Network security requires a separation of backend system and UI system for external collaboration scenarios for direct access.
2. PLM Core shipment shall not depend technically on ERP EhPs → PLM AddOn
3. PLM Core requires ERP backend access, since all PLM master data is persistent in the ERP layer.
4. ERP 2005 with NetWeaver 2004s

Comment:

- It is not required to install the PLM Core UI on a pure NetWeaver system in the DMZ without an ECC stack.

Summary:

SAP will ship PLM Core as a new Add-On in 3 releases for ERP 2005. PLM Core has one component with three strcuture packages:

1. PLM Core integration for ERP backend(ERP adapters for PLM and new IPPE version)



2. PLM Core UI
3. PLM Core Basis (prerequisite for PLM Core integration and PLM Core UI)

4.3 Target Architecture

Driver for the system architecture are network security requirements arising from internet based collaboration scenarios. This requires the separation of presentation layer in the DMZ and ERP/PLM data stored in a separated system behind the firewall.

However the UI component shall be installable on the PLM/ERP system additionally for internal users behind the firewall.

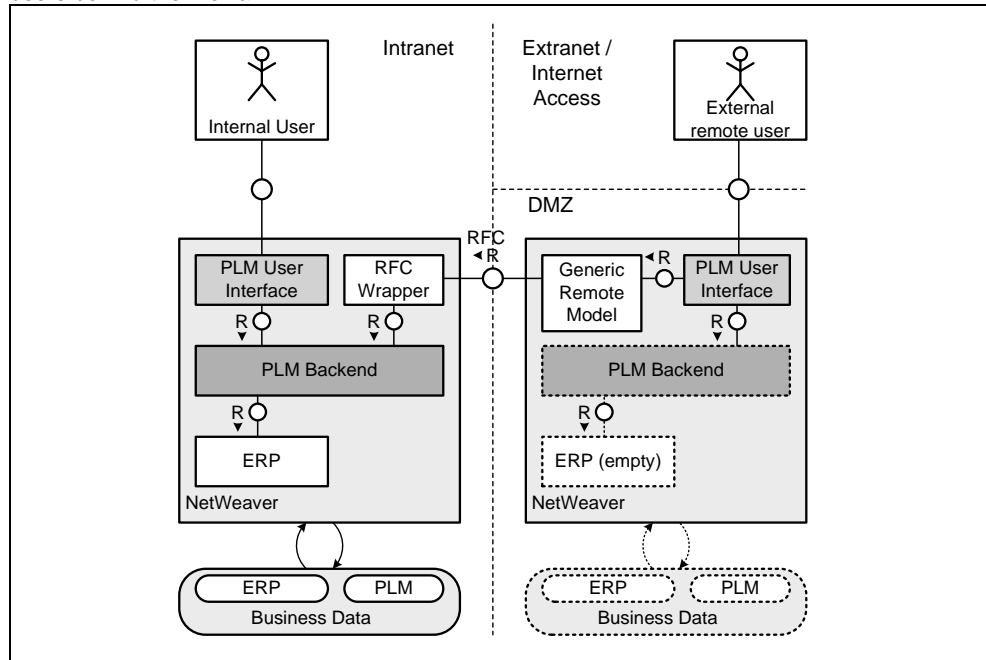


Figure 1: Network security requirements for extranet access

4.3.1 Architecture building blocks

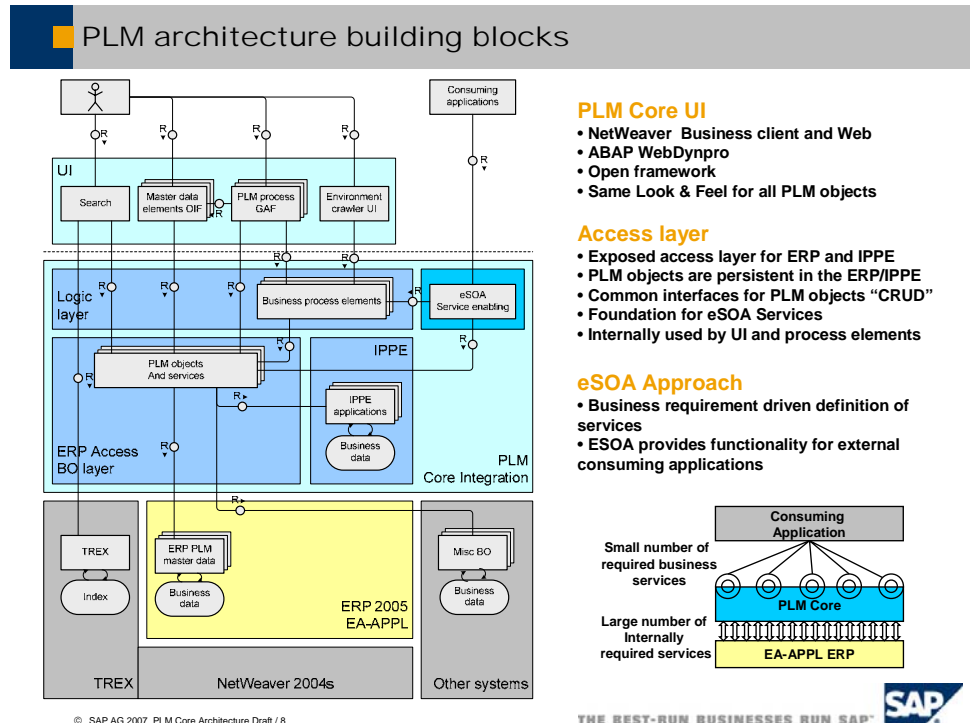


Figure 2 PLM Architecture building blocks

4.3.2 PLM Core UI

PLM Core offers all UI elements required for PLM. The main elements for the new UI are:

1. Search and work center elements
2. Object instance floor plan to maintain master data → Material, BOM, Documents and Change numbers
3. Guided activities providing wizard like functionality to guide the user through multiple steps
4. Environment crawler that allows a browsing along the complex PLM object relations

This layer must not contain business logic at all. The communication to the PLM Core backend is done via RFC. In order to grant an adequate UI performance the RFC communication between presentation layer and PLM backend has to be optimized which means to reduce the number of communication calls the chunk size of data transmitted.

Remark: Since ESA services are stateless they cannot be used for complex UI maintenance.

4.3.3 ERP access and BO layer

This layer is used to grant access to ERP or IPPE master data. The intention of this layer is to harmonize the access to the native ERP APIs that are persistent in the ERP/IPPE layer.

4.3.4 Logic layer

The logic layer provides business and engine functionality besides pure backend access. This is the foundation for guided activities and environment crawler logic.

4.3.5 Enterprise SOA Approach

Business requirement are driving the definition of services which are provided to be consumed externally by customer or partner applications.

eSOA will not be used internally for UI development for the following reasons:

- Services are stateless → Critical usability issues
 - A large number of “change” services is required.
 - Each service call requires a “Save”.
 - Locking issues for complex applications
 - BO Buffering does not speed up performance
 - Performance issues (forward/backward XML conversion)
- Very large number of internally required services
 - Very high effort
 - Internal development requirement drives the service definition
 - Technical interfaces instead of business driven interfaces
 - Limited number of services is used by customers

Open topic: Shipment alternatives (TDB) for PLM eSOA

- Part of PLM Core → Preferred by Solution management
- Part of ERP EhP

→ TBD together with ERP

4.4 Overview on Architecture Topics

4.4.1 Handling of ERP DDIC elements

A UI decoupling from ERP would require the duplication of many ERP data elements for PLMBAS. In order to limit the effort and risk concerning development, documentation and translation effort new DDIC structures shall be created that shall be used in the UI layer.

However ERP data elements shall be reused inside those structures to reduce documentation and translation efforts.

General rule: Do not keep data inside a DMZ which holds business information. This could be security critical (e.g. rabat groups from customizing,,).).

→ Thus, do not use frontend-local search helps but remote-enabled ones. This is valid for master data as well as for customizing.

Master data RFC enabled search helps shall be attached that consider collaboration specific authorizations. If possible a generic search help should be attached that is RFC enabled. This approach may not work for all search helps.

- Search help for collaboration scenarios shall consider collaboration authorizations. For performance reason this requires an own determination approach.

Naming convention <<ERPName>> → /PLMB/<<ERPName>>

4.4.2 UI development approach

A key requirement for the PLM UI is a simple UI enhancement concept. However PLM will not introduce a home made solution but rather will be based upon the existing UI WebDynpro custom controller concept available in NetWeaver. This custom controller concept shall be used for all PLM applications. That is to say UI building block elements are pieced together to form PLM applications by use of this concept. This allows an independent development of UI building blocks.

The same concept is used for the FPM (Floorplan manager section 5). This ensures a high degree of compatibility.

A role specific user concept shall be used in conjunction with the WebDynpro custom containers to control visibility.

Events trigger communication between those UI building blocks while “real” data shall be exchanged via model only.

In order to harmonize the UI development the model for the MVC approach has to follow certain guidelines and will most likely implement a certain generic interface.

Link to PLM UI Guidelines: << in development>>

4.4.3 RFC UI communication

External collaboration is one scenario for users to interact with the PLM. In this scenario the presentation layer is on separate machine in the DMZ. However this is a rare case compared to the number of users and customers that will work directly and only on the UI of the PLM/ERP machine. For this reason collaboration driven requirements must not have a major impact on the performance of the UI on the local PLM/ERP machine.

Note: *We are facing a paradigm shift here. Applications which are not typical external ones, tend to be used more and more from outside the intranet firewall, e.g. by users in remote shared service centers; it could also be even outsourced activities. VPN access is not a real solution here, as customers try to avoid the administrative overhead and rather prefer access via internet standards.*

The RFC protocol is used for the communication between UI presentation system and PLM ERP backend system in the collaboration scenario, since it supports “state full” applications.

However this system landscape constraint has a major impact on the design of the interfaces used for the UI communication. This means the data has to be transmitted in a way that it can be used directly within the UI. The goal is to minimize the performance overhead for RFC communication. This requires finding an optimum between the number of calls, the package size of transmitted data and number of required RFC enabled function modules.

Approach:

UI elements shall not directly call the business object layer or APIs. UI elements and applications shall use a dedicated UI models that act as an application service provider. Such a model offers data and specific logic for a certain UI application in a standardized and harmonized way.



The model implementation shall be located in the PLM backend system interacting with the BO/APIs.

A model factory shall deliver a model instance to the UI:

- Local → Model implementation directly
- Remote → Generic remote model

In the remote case the generic remote model delegates the requests to an RFC wrapper that shall provide a generic RFC communication to the backend that shall retrieve the data from the original model implementation.

It has to be ensured that this RFC model proxy fulfills the security standard. Opening the backend system for a generic RFC which itself could execute more or less anything in an uncontrolled way, could be refused by customers. Therefore this model proxy shall allow only communications via and against such models.

The advantage of this approach is that the UI does not care about remote or local installation. Furthermore the RFC handling is centralized and we not need of large number of very specific RFC modules. However the drawback is the effort for data conversion and the increase of data transmitted per call.

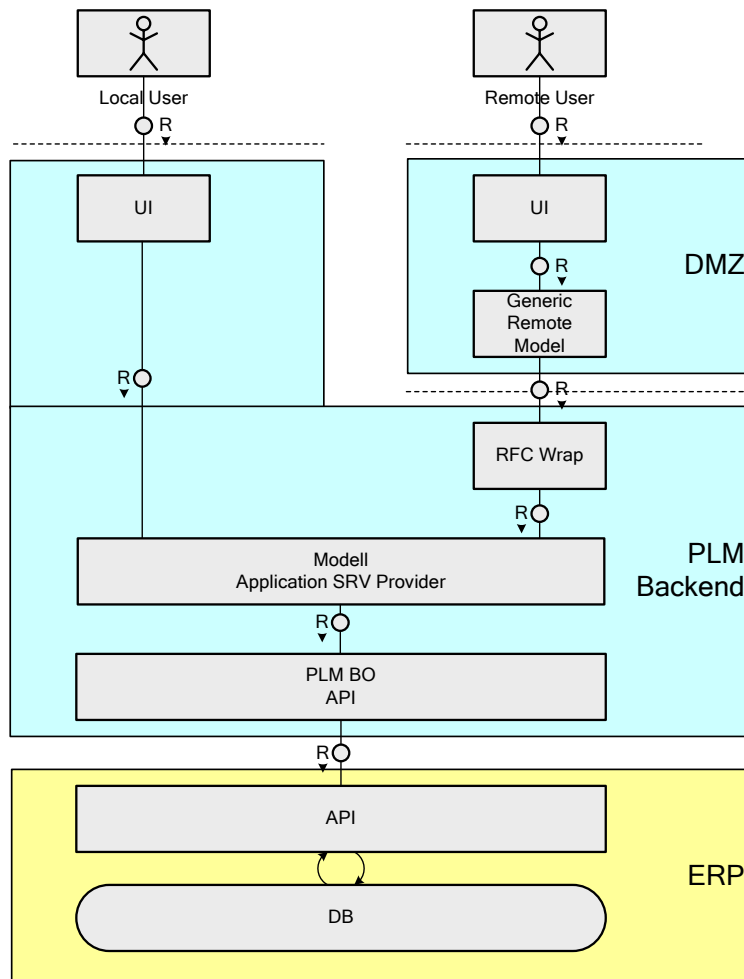


Figure 3: UI backend communication

4.4.4 Authorizations for collaboration

An enhanced authorization/team concept is required for PLM.

Those requirements cannot be handled by standard SAP and role concepts. Those authorizations shall be considered for searches and F4 directly.

Link to collaboration concept: <<in development>>

4.5 List of ERDs

Planned ERD	Responsible Architect

5 Technology Decisions and Constraints

5.1 Technology Selection

ERP 2005 with NetWaver 2004s

- ABAP WebDynpro

Open topics:

- CAD DMU (possibly an OCX-control) viewer integration

5.1.1 Development environment

- Original language for all development packages is English

5.1.2 Floorplan manager

The floorplan manager is currently in development and shall become part of ABA/NetWeaver.

All PLM OIFs and GAFs shall be technically based on the Floorplan manager.

- Same look & feel of all PLM objects and business activities
- All UI building block elements refer to the same interface
 - UI atoms can be developed independently → Efficient UI development
- Reusable UI building block “atoms” are “glued” by the framework
- Improved extensibility and UI configuration
 - Adjustment of Guided activities to customer specific business process

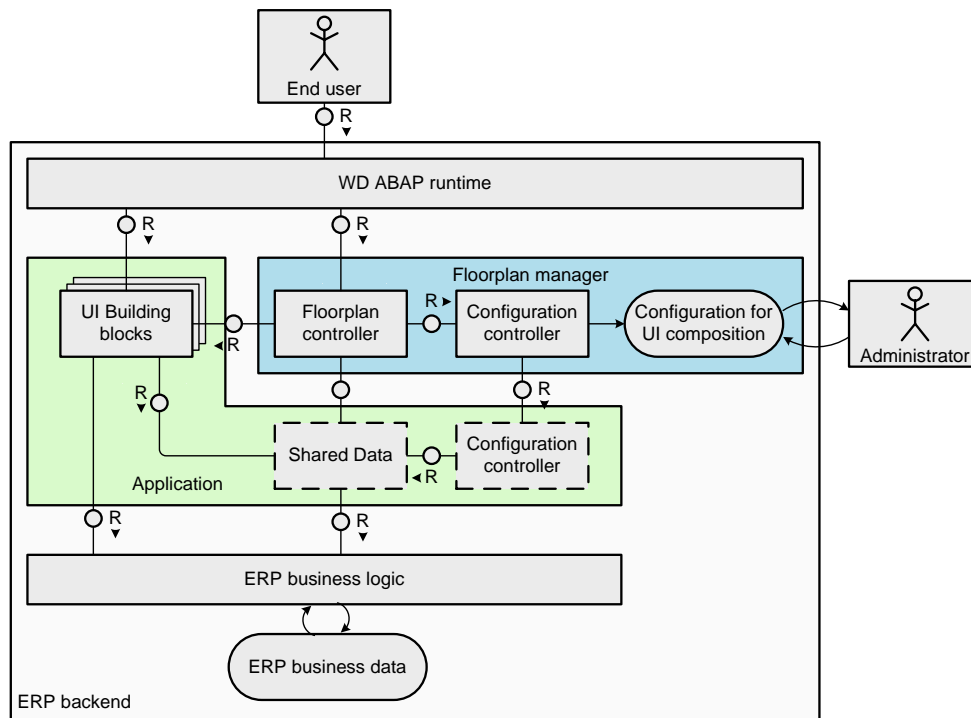


Figure 4: Floorplan manager architecture

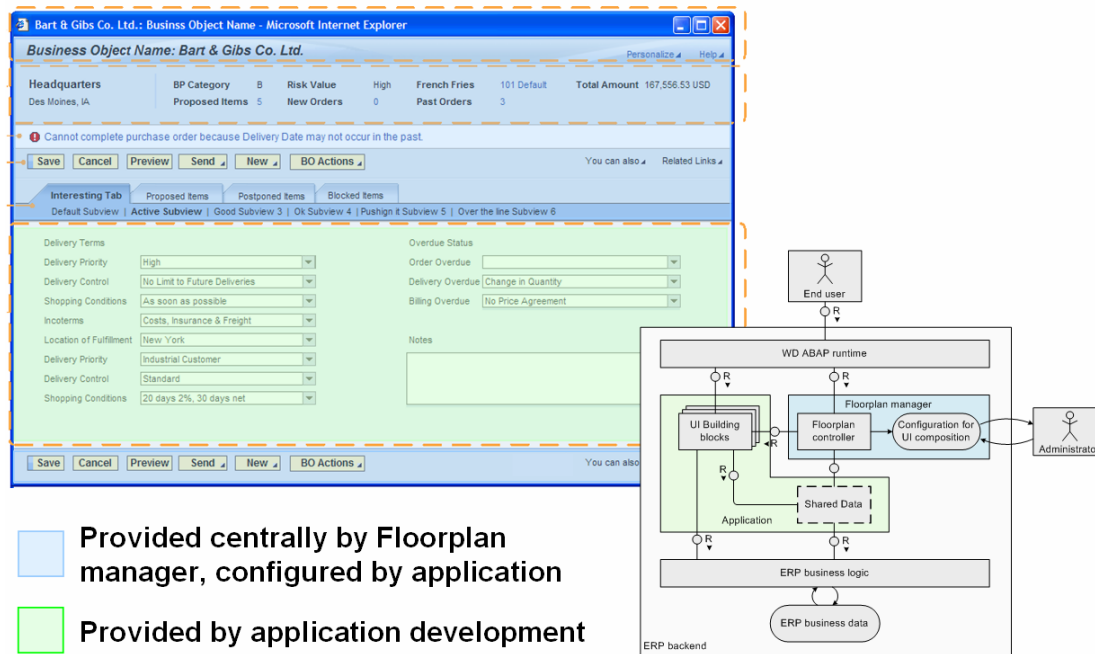


Figure 5: Example for an OIF

5.1.3 Collaboration decisions and constraints

External user has access to PLM data

Example: SAP C-User, Subsidiary in locations with low trust level

A dedicated authorization concept is required that can be maintained by the project lead. Therefore an ACL based authorization concept is required for external collaboration.

External user has access to PLM data in intranet from extranet

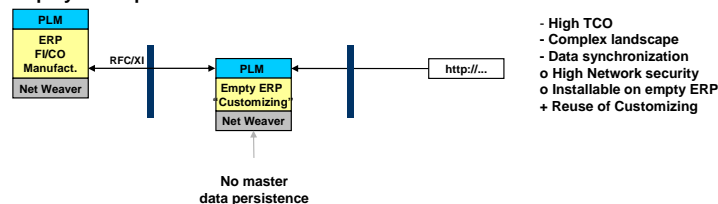
Example: Outsourced Manufacturing / Design collaboration with external development partner not working in the intranet

In addition to the authorization concept enhanced network security concepts are needed. This requirement is fulfilled by putting the presentation layer into the DMZ.

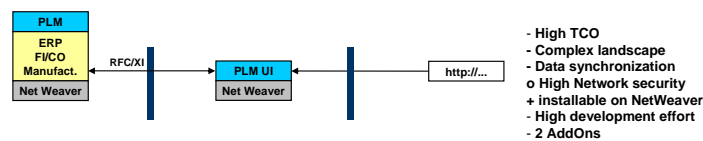
Collaboration & network security

Collaboration: External user has access to PLM data in intranet from extranet

- a. Separate PLM UI Layer → PLM has access to ERP/PLM master data via RFC
→ Deployment option 1a



- b. Separate PLM UI Layer → PLM has access to ERP/PLM via RFC
→ Deployment option 1b



© SAP AG 2007, PLM Core Architecture Draft / 21

THE BEST-RUN BUSINESSES RUN SAP®



Figure 6: Collaboration scenarios

Both approaches offer the same high degree of network security.

Decision: Approach a) is selected in order to reduce the effort for development, documentation and translation. Furthermore customer enhancements would become much more expensive in approach b).

However it has to be stressed that with our current Architecture approach a later strategy change to a "pure NW" approach for the UI is not blocked. For this reason component PLMUI shall only refer to DDIC structures provided by PLMBASE. Inside these structures ERP data elements are allowed, but may be replaced at a later time in case of such a strategy change.

5.2 Prohibitions for technology issues

- Java WebDynpro
- C++, C#
- Java
- xPRA

5.3 Exceptions for technology issues

- RFC for communications between backend and UI → Necessary for statefull applications

6 Reuse Decisions and Constraints

6.1 Selection and Prohibitions for reuse issues

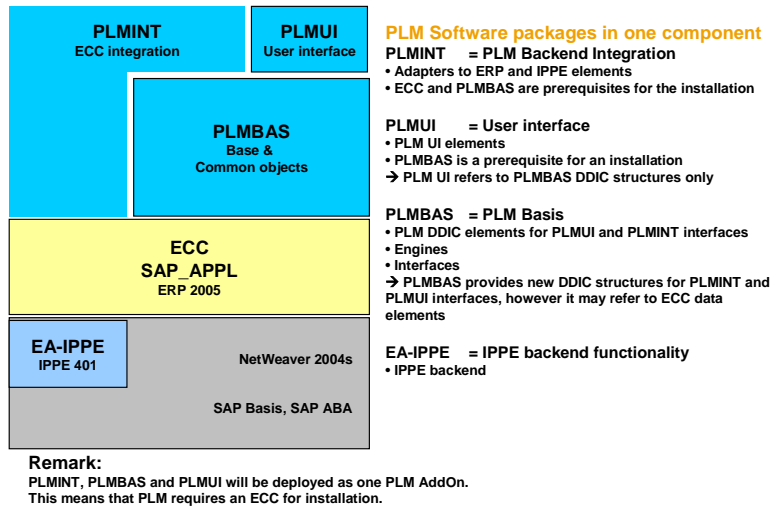
Reuse software (software component version)	Owned by (BPP / ECC / other)	Maintenance until end of extended maintenance of application?	Usage (must / should / must not)	Remark / explanation
Floorplan manager	ECC	Yes	Must	Key element for UI OIF
ERP APIs	ECC	Yes	Must	Access to ERP is key since most PLM objects are persistent in ERP.

6.2 Exceptions for reuse issues

7 Software Layers and Components

7.1 Software Layers

PLM Software layers and components – Empty ERP → 1a



© SAP AG 2007, PLM Core Architecture Draft / 8

THE BEST-RUN BUSINESSES RUN SAP®

Figure 7: Software layers and main PLM packages

PLMINT and PLMUI require a common PLMBAS software package that provides DDIC, generic functionality and engine parts.

PLMUI must not refer to ECC directly. PLMBAS shall provide DDIC structures that can be used by PLMUI. PLMBAS may refer to ECC data elements and documentation. Search helps for customizing and master data shall be RFC and collaboration authorization enabled.

7.1.1 PLM installation scenarios

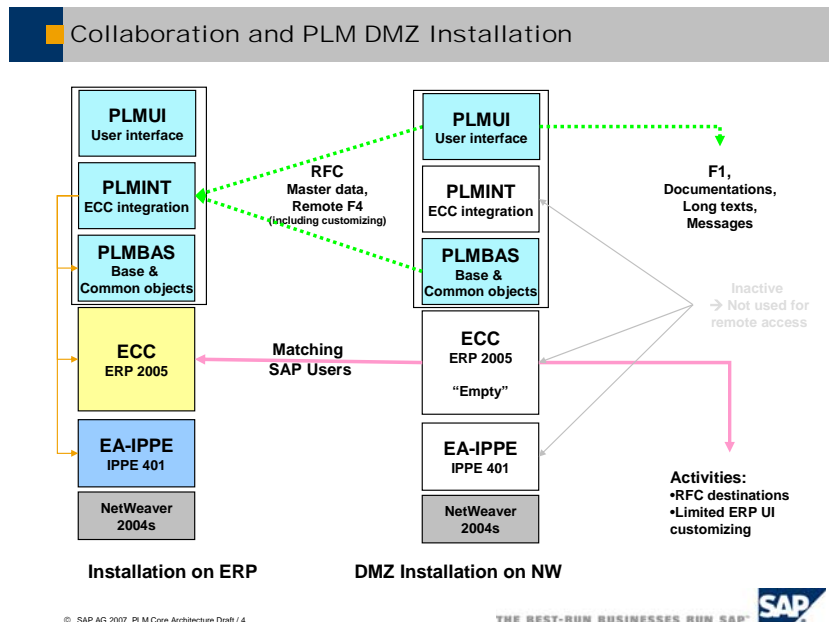


Figure 8: Installation scenarios

a) Customers not using external collaboration will install the PLM AddOn component that (PLMUI, PLMBAS, and PLMINT) on top of an ERP system. This ERP system contains customizing and all ERP master data.

b) For external collaboration a second PLM system for the presentation layer is required in the DMZ. This system contains no ERP master data nor customizing. However user and RFC destinations are required besides some limited PLM UI/ Collaboration configurations. In order to avoid conversion exit issues the related exits or configurations needs to be applied to the remote system as well (e.g. MATNR).

7.2 List of Components

PLM Core has one component with 3 main structure packages:

PLMBAS: PLM Core Basis

This package is a prerequisite for the PLM Core integration and Core UI. It contains the DDIC information necessary for the PLM Core UI to backend communication and generic engine elements for PLM Core.

An own namespace /PLMB/ is used for all elements of this layer.

PLMUI: PLM Core UI

This is the presentation layer for the PLM solution. It contains all specific UI elements based on the DDIC provided by the PLM Core Base. This layer must not contain any logic.

An own namespace /PLMU/ is used for all elements of this layer.



This package must not use and refer to any ERP DDIC structures. It shall be limited to PLMBAS DDIC only.

PLMINT: PLM Core integration

This layer grants access to the ERP backend data. It contains the adapters that are called from the UI layer. It has to be emphasized that all BO access methods have to use only PLM Core Base DDIC structures in their interfaces.

An own namespace /PLMI/ is used for all elements of this layer.

This package requires an ERP and the installation of PLM Core Base and optional PLM Core UI.

EA-IPPE: IPPE backend functionality

For the first release no IPPE enhancements are planned. However the Engineering desktop requires some minor adjustments.

7.2.1 Core PLM packaging

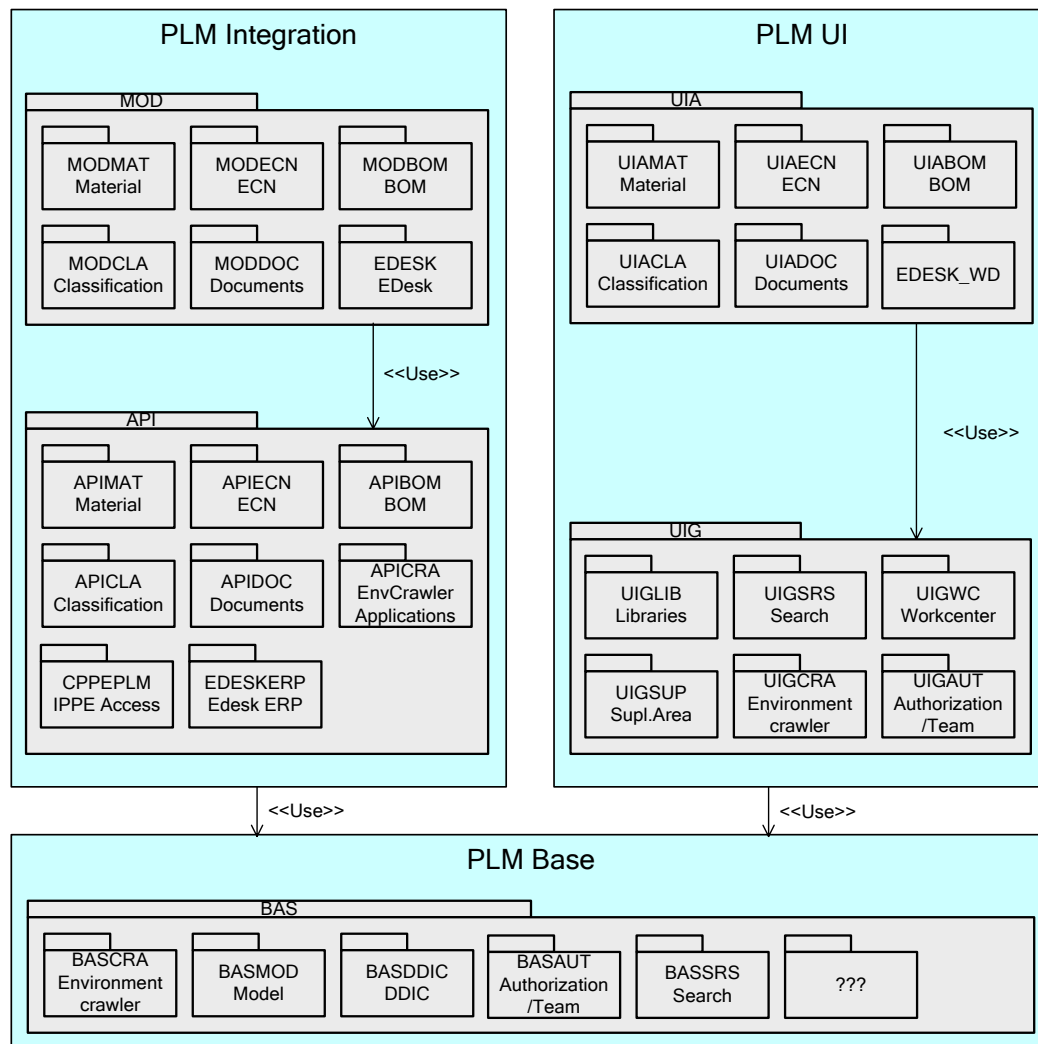


Figure 9: Rough PLM packaging concept

The packaging will be finalized in a separate document.

7.2.2 Own Application and Software Components

mySAP ERP 2005 – application release

mySAP ERP 2005 ECC ABAP Server – application release and application component

SAP NetWeaver 2004s ABAP Server – underlying application release and application component

Application Component	Required Underlying Application Release and Application Component	Software Component Version	Owner	Technically Dependent On
<i>mySAP ERP 2005 ECC ABAP Server</i>	<i>SAP NetWeaver 2004s ABAP Server</i>			
EA-IPPE				
SES	SES and TREX			

7.2.3 Non-SAP Software Components

- CAD 3d viewer

Software Component Version	Classification	Remark	Requires Legal Changes?
3D DMU viewer	<3rd party supplier>	TBD which viewer shall be used.	Yes / No

7.3 Deployment Scenarios

7.3.1 PLM Deployment

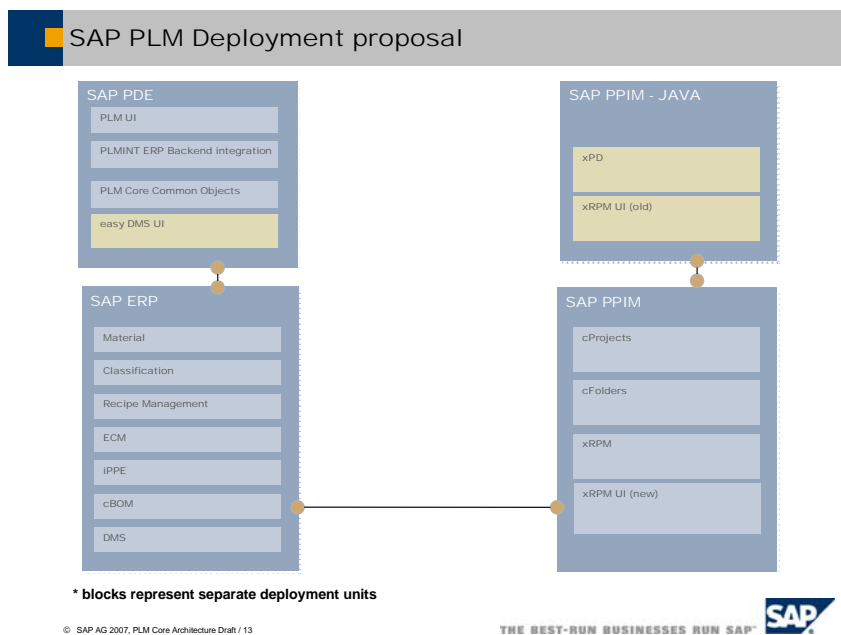


Figure 10: SAP PLM Deployment

The PLM AddOn includes one component with the 3 packages PLMINT, PLMUI and PLMBAS. Easy DMS backend shall become part of PLMINT while the UI element should be still C++ based with the intention to reuse AbapWeb Dynpro Elements for SAP document master data.

SAP PPIM shall be shipped separately.

7.3.2 PLM release cycle

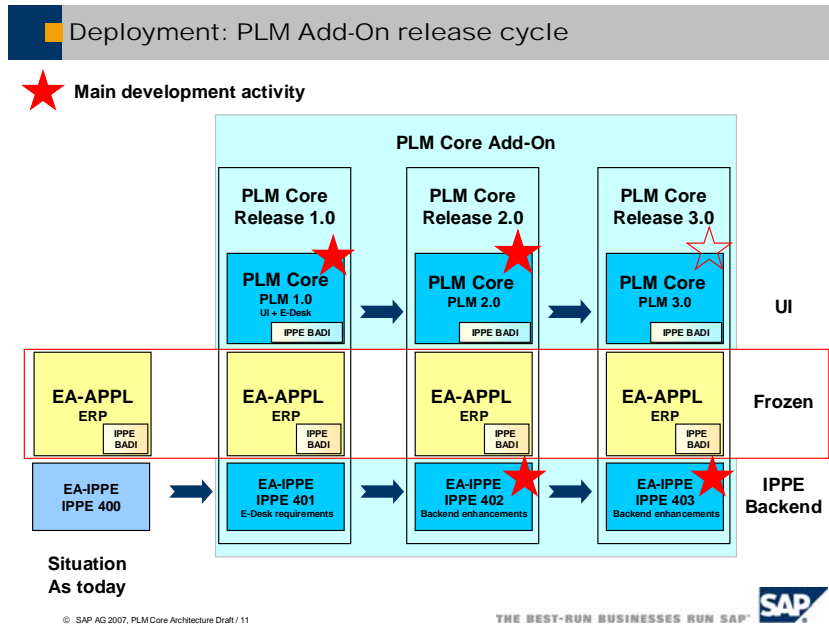


Figure 11 Add-On release strategy

ERP objects the software layer EA-APPL shall be treated as frozen.

7.3.3 PLM Core and ERP Enhancement package dependencies

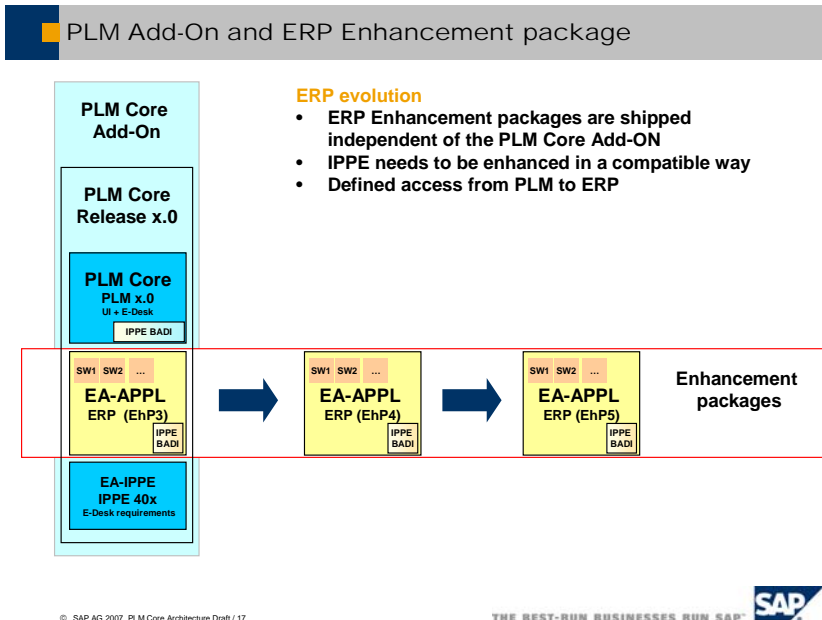


Figure 12: ERP EHP evolution

7.3.4 PLM Core development constrains

Consequence and key-findings:

- No modifications in EA-APPL → Impact and relation to ERP enhancement packages.
- EA-IPPE Enhancements need to be compatible concerning signature and functionality. This is key because of ERP applications like RM, MPL, IPPE are using EA-IPPE.
 - Existing interfaces and signatures need to stay compatible. → Only extending existing signatures and interfaces is allowed.
 - Defined API functionality must not be changed or redefined.
 - The existing IPPE UI needs to be compliant to any IPPE backend change.
- The PLM Core Add-On system will be the master system for any future IPPE development. → No IPPE development outside PLM Core.

7.3.5 ERP cross demand handling

Any sort of X-demand (cross) interferes with the PLM Add-on approach. Although X-demands are not planned for the first release yet this chapter shows the implications of the system architecture.

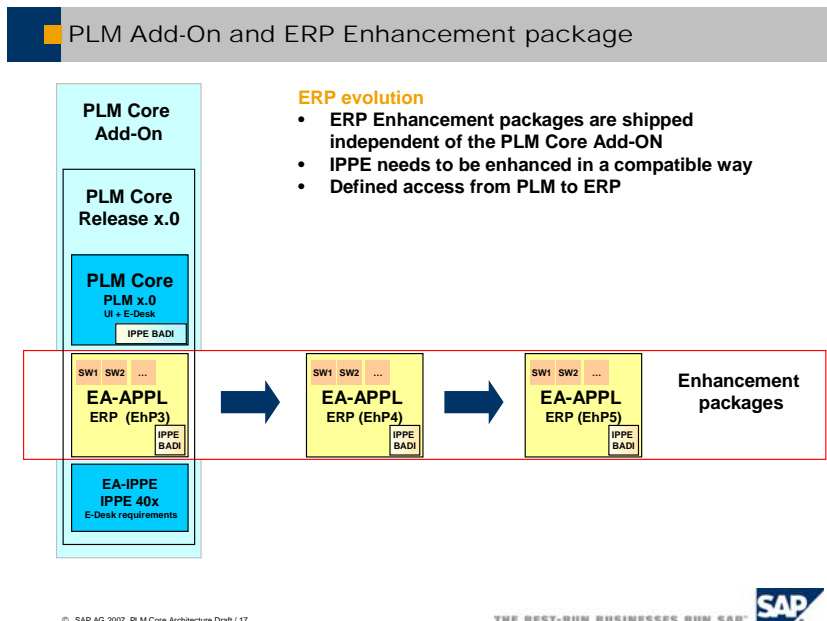


Figure 13: ERP Cross demands

PLM has ERP cross demands

This means PLM needs an enhancement in the existing ERP functionality.

1. **Modification of the ERP component** in the PLM development system is not recommended since those may interfere with enhancement packages.
→ **No Option!**
2. ERP provides functional enhancements via enhancement packages (or support packages). If PLM specific elements are necessary within the ERP code base, those are provided in a decoupled way via BADIs. In this case the BADI definition is done in the ERP enhancement package(or support packages), while the BADI implementation is done in the PLM system.
→ **ERP enhancement packages(or support packages) become a logical prerequisite for a certain PLM Add-On release functionality.** However this does not mean that a EHP is a technically installation prerequisite.
3. ERP functional enhancements are postponed to the next official ERP release. However the situation is very similar to the previous topic.

ERP has a PLM cross demands

This means ERP needs a certain PLM functionality enhanced. Two cases can be differentiated:

- a) The functionality does **not** have any impact (coding/functionality) on the PLM core release. In this case the functionality can be enhanced directly in the EA-APPL layer which is the origin of the related PLM objects that technically belong to EA-APPL.
- b) The same functionality is required in the PLM Core Add-On as well. The situation is similar to scenario 2 "PLM has ERP X-Demands".

- c) ERP needs a functionality that was already developed in a certain PLM Core release. This is a kind of a “Downport-Problem”. Such a situation would lead to a technical code redundancy that needs to be consolidated in a later PLM Core release. However this case seems to be rather unlikely.

7.3.6 IPPE ERP2005s dependencies

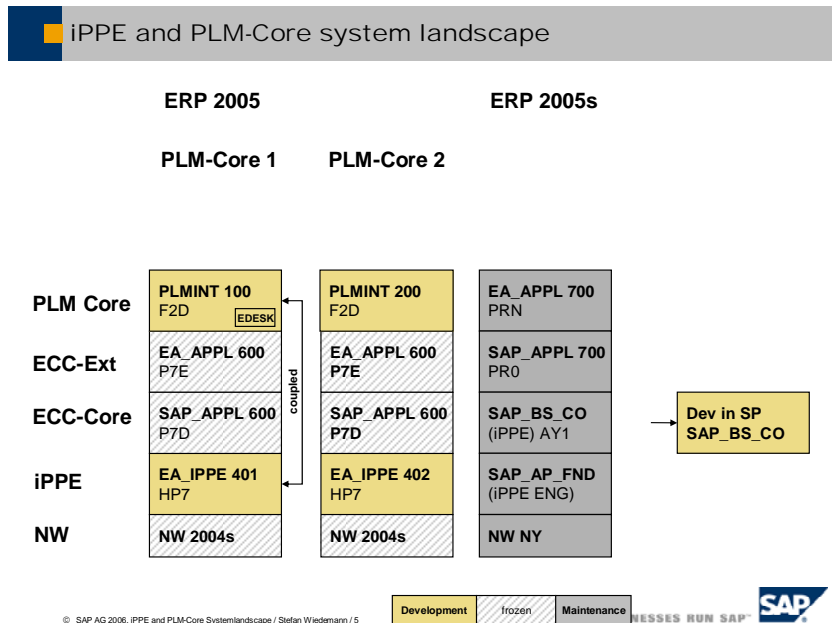


Figure 14: IPPE landscape and upgrades

EA_IPPE

IPPE is not a matter of development in release 1. However IPPE shall become the backbone of the engineering solution in release 2.

IPPE in ERP 2005s:

IPPE application objects are moved to another software component in ERP 2005s (former known as ERP 2007).

Originals of the software components are located:

- IPPE engine Foundation layer → [SAP_AP_FND](#)
→ Changes concerning the engine are not intended by PLM.
- IPPE application objects → [SAP_BS_CO](#) (System AY1)
→ Necessary for functional enhancements. This system needs to be kept in sync with the PLM Core for EA-IPPE.

This would cause issues in future for PLM AddOn customers going for an ERP2005 → ERP2005s upgrade.

Strategy:

- IPPE developed in AY1 according to EhP guidelines for component SAP_BS_CO

- Development of EA-IPPE 401 und EA-IPPE 402 enhancements according to the EhP guidelines
- New functions are documented via T061 Formular and presented to the ERP Dev&Test-Meeting

7.4 Constraints

Shipment constraints:

- It is not intended to provide the PLM Core AddOn on ERP 2005s. A shipment on top of this release is out of scope.

8 Relations to Standards and Guidelines

8.1 Related Program Architecture Guidelines

- Guideline for UI and model development
- Guideline for BO/API development
- Programming guidelines and naming conventions
- PLM packaging

Helios Guildelines WIKI, programming guidelines:

<https://wiki.pal.sap.corp:8443/display/SAPGuide/Home>

8.2 Impact from SAP architecture adoption guidelines

Adoption Guideline	Impact on Architecture / Deviations from guideline

8.3 Product Standards

Product Standard	Impact on Architecture
Accessibility (see link)	
Application Integration & Interfaces (see link)	
Customizing & Configuration (see link)	



Product Standard	Impact on Architecture
Data Archiving (see link)	
Development Environments (see link)	
Documentation (see link)	
Functional Correctness (see link)	
Globalization (see link)	
Multiple Clients (see link)	
Opensource/Third Party (see link)	
Performance (see link)	
Security (see link)	Architecture defined by network security aspects.
IT Service/Appl. Management (see link)	
Technical Implementation and Change Management (TICM) (see link)	
Usability (see link)	

9 Development landscape

10 Glossary

Application	Software unit comprising a specific collection of business processes required to address defined business needs. Example: mySAP ERP
Application release	Release of an application. Consists of application components. Example: mySAP ERP 2005
Application component	Group of technically dependent software component versions. Common technical unit of installation and upgrade. Can be seen as a "server"; corresponds to a PPMS instance. Example: ECC ABAP Server within mySAP ERP
Software component	Smallest unit of installation and maintenance. Corresponds to a Support Package track. Example: SAP KERNEL 32Bit
Software component version	Release of a software component. Building block for application components. Example: SAP KERNEL 6.40 32Bit