



# Teamcenter 2005: Guidelines for Planning and Operation

**Volume ITI-MTI-A**  
Written by  
**Erich Brauchli**

March 2, 2008  
**Version 9.24**

Reflecting Teamcenter<sup>®</sup> Enterprise 2005

Email: [erich@brauchli.info](mailto:erich@brauchli.info)  
Internet: <http://www.brauchli.info>  
(Total 28 pages)

**Entire List of Teamcenter<sup>®</sup> Enterprise Documents:**

ITI-MTI-A	Guidelines for Planning and Operation
ITI-MTI-B	Guidelines for Setup and Configuration
ITI-MTI-C	Guidelines for Oracle Setup and DB Configuration
ITI-MTI-D	Guidelines for Customizing
ITI-MTI-E	Rules

# Content

<b>1</b>	<b>General Remark</b> .....	<b>3</b>
1.1	How to start a PDM project.....	3
1.1.1	Top Down Phase.....	3
1.1.2	The Bottom Up Phase.....	4
1.2	The Project.....	4
<b>2</b>	<b>Application Management</b> .....	<b>4</b>
2.1	Naming Conventions.....	4
2.2	Mapping Conventions.....	7
2.3	Teamcenter® Enterprise Instances.....	7
2.4	Customization Rollout.....	8
2.4.1	Rollout Processes.....	8
2.4.2	Rollout Types.....	9
2.4.3	Rollout Checks.....	9
<b>3</b>	<b>Facility Management</b> .....	<b>11</b>
3.1	Teamcenter® Enterprise Topology.....	11
3.1.1	Single Database Environment.....	11
3.1.2	Distributed Environment.....	11
3.1.3	How to distribute Data in Distributed Databases.....	14
3.1.4	How to distribute Databases.....	15
3.1.5	Sample distribution schema.....	16
3.1.6	Database Topology using the Configuration Editor.....	16
3.1.7	Federated Systems.....	17
3.1.8	PDM and MRP/ERP.....	17
3.1.9	Configuration Editor – Instance preparation.....	17
3.1.10	Files Systems / Installation Location.....	18
3.2	Training Concepts.....	19
3.3	Teamcenter® Enterprise Administration.....	20
3.3.1	General Level.....	20
3.3.2	Level 1 Administration: <i>site independent</i> .....	20
3.3.3	Level 2 Administration: <i>site dependent</i> .....	20
3.3.4	Level 3 Administration: <i>site dependent</i> .....	20
3.3.5	Operation Management.....	20
3.4	User tips.....	20
<b>4</b>	<b>Teamcenter® Enterprise Concepts</b> .....	<b>21</b>
4.1	Dictionary and Data Services.....	21
4.2	Network usage of Teamcenter® Enterprise.....	22
4.3	Compromise.....	23
<b>5</b>	<b>PDM Basics</b> .....	<b>23</b>
<b>6</b>	<b>Processes</b> .....	<b>24</b>
<b>7</b>	<b>Teamcenter® Enterprise Development Organization</b> .....	<b>25</b>
7.1	Instances.....	25
7.1.1	Teamcenter® Enterprise Instance.....	25
7.1.2	Oracle Instance.....	26
7.1.3	Comparison between DB terms of Oracle and Teamcenter® Enterprise.....	26
7.2	Logical structure of Teamcenter® Enterprise instances.....	26
<b>8</b>	<b>Platform specifics</b> .....	<b>28</b>
8.1	AIX.....	28
8.1.1	Shared Libraries.....	28
8.2	HP-UX 10 and later.....	28

## 1 General Remark

This document intends to give a first input for topics to resolve prior to start a Teamcenter® Enterprise implementation. It is not a real manual, because a lot of topics will be solved in different ways depending on company contexts and on instance complexity.

This document should help to plan the important details for a productive Teamcenter® Enterprise environment. Company specific solutions are adequate. This can help to find the appropriate solutions.

Today's definition of PDM<sup>1</sup> is rather management of the company's "intellectual capital".

### 1.1 How to start a PDM project

Two facts are common for all PDM projects:

- No project got to success that had not strong commitment by high level management; a PDM tool is mainly a backbone technology to reduce sources of same data and to support overall information management.
- No project got to success that was not driven by end users. If these people do not accept it and see soon an advantage in using the tool, then it may be impossible to get the brake through.

This means, that any PDM project must start with a corporate conceptual phase.

#### 1.1.1 Top Down Phase

In this state there is not yet any discussion about the tool to apply.

**A prerequisite for such a project is a strong commitment of the management and a firm committed and documented business model and policy, enforced for implementation.**

- **Process definitions**, eventually even process re-engineering. But first of all you need to be aware of the actually in place processes for design, manufacturing, delivery, sales, maintenance and recycling.
- **Generic data summary**: decide about the overall data to maintain for design, manufacturing, delivery, sales, maintenance and recycling as well as for legal liability reasons.
- **Target setting**: split the above into manageable pieces and partial goals, which are committed by the company's management.<sup>2</sup>
- **Guidelines**: Today's PDM and MRP solutions are complementary<sup>3</sup>. They share some data during partial lifetime of products. Thus guidelines about coexistence and communality are required from a business standpoint to avoid civil wars between internal product freaks.
- **Not more detailed than necessary**: be aware that you want to create an application which must map all the changes in your business constantly
- **Keep it simple**: PDM project tend to be complicated, while technology freaks want to use the possible features. It is easier to manage, to achieve goals and maintain if you start from what you need rather than from what could be done.

Here may follow an intermediate phase, where the tool may be chosen. The goals and targets define essential constraints.

---

<sup>1</sup> ) Born as CAD drawing management tool and evolved to general „Product Data Management“.

<sup>2</sup> ) Degree of details must not be higher than a general management statement can cover.

<sup>3</sup> ) Actually neither system can really fulfill the requirements of the other, despite any promises of sales forces. But coexistence models and interfaces exist and are deployed successfully.

### 1.1.2 The Bottom Up Phase

The implementation then should start in bottom up method, so that a small community of users in a limited area may profit as quick as possible from first results.

- **Sub Projects:** small and effective sub projects are launched, starting with small user community and the cooperation of these communities, but strongly respecting the general rules and targets.
- **Sub Project should give a positive visible result within 6 months.**
- **Data Migrations:** from legacy system to the new data model are usually under estimated, due to over estimation of the data quality. The correction loop for such programs are normally 4 times longer than estimated. Such sub projects usually break the rule of “small sub projects”. That is why this type of sub project is hard to be the starting point, even your PDM get no value if you have no data in.

## 1.2 The Project

One of the essential goals of a PDM system must be flexibility. Conditions and requirements of your company are changing constantly. If you define a project, approve it, develop it, and release it in five years from now, then your result will be completely out of date and can no more be accepted, not because of mismatch with definition, but because of meanwhile stale definitions.

Thus the whole PDM project cannot use all of the standard Project Rules. It has to fulfill generic goals, which may change more often than you like. But it is real life. Thus we need steps and milestones, more detailed for short term, rather generic for long term.

Break down the step to next milestone into reasonable sub projects. These must then undergo a strong project management with all details according your company's project methodology.

## **2 Application Management**

### **2.1 Naming Conventions**

It is recommended to fix some conventions before starting a productive Teamcenter® Enterprise installation. Main subjects for conventions are naming schemas.

- **User Names / OS User Names:**

OS User Names are generally the names under which a user logs in on the OS system. These are often cryptic names. It is recommended to maintain one single OS User Name per user over all systems. Teamcenter® Enterprise can identify the user based on this OS User Name on systems requiring a password at login; thus Teamcenter® Enterprise does not require an additional login dialog for password submission (except for WEB clients). Avoid multiple persons with same OS User Name. If several Teamcenter® Enterprise users shared the same OS User Name as the Teamcenter® Enterprise “super user”, and then some system functions of Teamcenter® Enterprise may not properly work!

The Teamcenter® Enterprise User Name therefore is rarely to be typed in, but is very often used to identify an individual as owner or actor as well as in sign-off. This fact shows, that it is preferable to here the individuals full name in some conventional format (like family name, first name, spec). Be aware that this must be a unique name among all users, user group names role and vault names.

*Samples:*

*Miller John*  
*Adams Marc, Sen.*  
*Adams Marc, Jun.*

- **Vault Naming Convention:**

Decide whether you use Vaults by Geography or Organizational Units or other **ownership relevant** criteria. Be aware (in distributed case) that an entire Vault is stored in ONE database (except replicated object classes).  
Use meaningful names, so that users can decide where to query.

It is usually not recommended to create Vaults by projects or status; these facts are maintained in their respected attributes. (To query for released documents, fill the LifeCycleState in the query dialog). Rules by project are not depending on a Vaultname but rather to be designed by roles! This is by far more effective and performant. Be aware: **The Vault is the OWNER** of the contained objects, neither less nor more.

It is useful to have structured Vault Names (like "Vault\_Specific"). This simplifies tremendously the Condition "IsInVault"

```
(IsInVault(obj,env) := $obj.OwnerName like 'Vault*');
```

valid for all your Vaults and distinguishes easily ownership of persons, teams and vaults.

Often sites want to have the Vault names on top of the owner name lists (selection box for transfers and checkins). To achieve this, use Vault names like "@Vault Specific". The list is always sorted alphabetically; thus names starting with the at sign will be on top.

In a distributed environment you select a vault for CheckIn; but you need to know the DB name for query, to be really efficient. Thus use names for Vaults and DBs, which easily may be brought together in users mind.

Use as few User DBs as possible.

**Queries over more databases (having more than one database in the users "SCOPE") tend to be slow, especially if some are remote bases.**

- **Vault Location Names and Distribution:**

Vault Locations are directories on some host for storage of file system items (files) of a Vault.

A Vault Location belongs exactly to one Vault (the Vault is the owner of all items in the Vault). A Vault may have as many Vault Locations as necessary on as many hosts as necessary, even replicated ones. But all its items have their meta data in the same database. Thus a Vault residing in a European database may still have some of the Vault Locations in US, if this makes sense.

Use Vault Location on secure systems only. Distribute them to have data as near as possible to where they are most frequently used. And grant its backup in synch with the relative DB.

Reflect in the Vault Location Name both the Vault Name as well as the site where the location is physically. Objects may easily be moved from one Vault Location to another, if there is a need.

Recommended names are "<Name of the Vault> <n>" or "<Name of the Vault> <Site> <n>", where "n" is just a sequence number. You need to add new locations to a Vault, when disk space get short, or when a location contains more than some thousand files (suggest 5000), because you get in performance problems on OS level, when the number of files in a directory is to high. (Don't forget to update your location selection rules, when you add a Vault Location).

If there is not an explicit need for maintaining original file names in Vault Locations<sup>4</sup>, you should select randomized file names in Vault Locations. This avoids file name conflicts (several users tend to use same file names all over and over, or several versions of the same file may coexist in a Vault). The users will get back the files anyway always with the original file name. And on OS level nobody should “know” which file is which.

- **File Naming:**

Generally files are attached to a business item. Business items are identified by Teamcenter<sup>®</sup> Enterprise keys, files do not have a key; thus the system may host any number of equal file names (but not in the same location; except Vault Locations. In case of name conflicts upon checkout into a users Work Location the local file name is automatically and temporarily slightly modified.)

Today operating systems allow some fancy file names. Especially, since Windows allows long file names, we see often file names containing European characters and spaces. Teamcenter<sup>®</sup> Enterprise can handle this. But even not all Microsoft application will do so. And some backup tool cannot deal with it. Therefore it is strongly recommended, to **not allow** this type of file names.

The easiest way to avoid such files in your system and archive is to define a “picture” (pattern) on the file name attribute. This prevents from creating or registering such files.<sup>5</sup>

- **Group Naming:**

User may be grouped in user groups for organizational purpose, for project independent rule and as distribution list.

It is recommended to use “speaking” Group names and to add the suffix “Grp” to all. It is easier to select in some of the selection boxes.

- **Roles and Groups:**

Roles are used for project depending Rules.

Groups are used for project independent Rules.

*Recommendation: use*

*one word names for roles*

*two or more word names for individuals*

*use suffix “Grp” for user groups*

- **Condition Naming:**

Use a clear schema and “speaking” names. Only persons in charge of “Life Cycles” and “Rules” use them. But it is very slow to select one, if you need “Get Item Info” for each one first, to know what they are for.

Avoid conflicts with Teamcenter<sup>®</sup> Enterprise standard conditions by adding on all you create yourself a lower case prefix, recommended having a digit as second character.

- **Life Cycle and Process Naming:**

Even end users are sometimes confronted with a selection box to select a process, the object has to go in. Thus use again “speaking” names.

Do not worry about changes in the processes. They have revisions (may be revised, getting a new revision number and the same name). But only one revision is valid and selectable at a given time.

---

***Teamcenter 2005: Guidelines for Planning and  
Operation***

*March 2, 2008 (V9.24) / Erich Brauchli **Erich***

- ❑ **Industrialization environment:**  
(eventually two or more; depending on company need and application management traditions)  
Responsibility: Deployment Administration, Testers, ...
  - **No Development Tests**
  - **No Modifications!**
  - **Only Installation and site defined TEST SUITE!**
  - create user information, quick reference manuals, “cooking receipts”, ...
- ❑ **Test and Training environment(s):**  
Responsibility: ...
  - Training classes
  - User may try things after training before work in productions system. Try out, where you do not really know, how to do things.
- ❑ **Production environment:**  
Responsibility: Deployment Administration
  - **No Tests**
  - **No Modifications!**
  - **Only Installation and Deployment!**

## **2.4 Customization Rollout**

### **2.4.1 Rollout Processes**

- ❑ **Rollout Processes**
  - **Development Cycle** (clear separation of responsibility between development subcontractor and company by contract)  
As a general rule development should deliver a Teamcenter® Enterprise standard installation file set.
- ❑ **Rollout from Development to Integration**  
using "cfgedit2"
  - upgrade System
- ❑ **Rollout from Integration to Industrialization(s)**
  - Reset Data (Clear and Reload Database and Files)
  - using "cfgedit2"
  - Create New System Backup
  - Execute Test Suite(s), adapt User Info
  - Expand Test Suite relative to new functionality (to be documented)
- ❑ **Rollout to Production System**  
(using same source as above)
  - using "cfgedit2"
  - eventual Data Upgrade by special Procedures

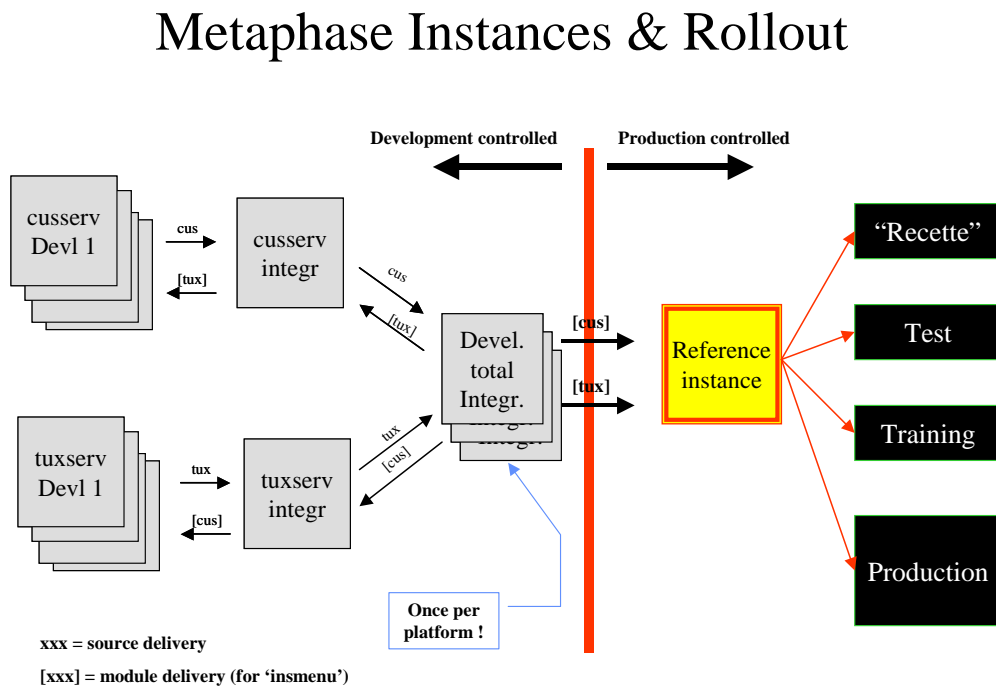


## 2.4.2 Rollout Types

- **Types**
  - Operating System Upgrade
  - RDBMS Upgrade
  - Customization Upgrade
  - Teamcenter® Enterprise Release Upgrade
  - Integrated Application Upgrade
- **Rollout Sequence always the same:**
  - Development ⇔ Industrialization ⇔ Production
  - All same formalism
- **Development Layers**
  - To be defined by company

More details see in chapter 7 Teamcenter® Enterprise Development Organization.

Sample:



- A file dump of all the directories, which are registered as file systems in this Teamcenter® Enterprise test instance.
- Complete dump of this Teamcenter® Enterprise instance's database(s). [Refer to backup / restore descriptions].

The functional test step for EACH rollout must contain the following steps:

1. Add test steps to the "test suite" for any new functionality.
2. Drop the database(s), i.e. Oracle users, for this Teamcenter® Enterprise test instance.
3. Remove the contents of the directories registered as file systems in this Teamcenter® Enterprise test instance.
4. Reload the data base(s) from the starter set (see above).
5. Restore the files from the starter set dump.
6. Install the upgrade.
7. Add initial data for test of the new functions.
8. Recreate the "starter set" (see above)
9. Go through the test suite and write a test log mentioning all actions and their results (for standard, old and new functionality).

Remark: Steps 7 and 8 can be skipped, if absolutely no new functionality is added by the current upgrade. This "test suite" should be executed on both Teamcenter® Enterprise upgrade and customization upgrade. When upgrading Teamcenter® Enterprise standard, this is always an increase of functionality; may be your site will not (or not yet) use it.

### 2.4.3.2 Installation verification

After any type of upgrade it may be good to verify if all has passed correctly. The first step is to go through all the logs generated by the upgrade. Verify that all step "completed successfully", that database schema updates have been successfully applied to all databases.

There is a very simple method to verify that the database schema corresponds to the new data dictionary: execute the following command against each of your databases:

```
objdump -d <DB-name> -o /dev/null -w OBID='0' PdmRoot
```

This command will dump no object, but issue a correct complete SQL select for each table; thus it will crash, if there is a database table missing or if a column is missing in a table.

To verify, that the new binaries have been installed correctly in the system, you list the files in "\$MTI\_ROOT/bin" (%MTI\_ROOT%\bin on Windows). The files of following list of files need to have the new compilation date:

UNIX:	CUSserv	Windows:	CUSserv.exe
	libsCUS.so or libsCUS.sl		libsCUS.dll

Where CUS is your custom servers short name (normally in lower case).

The command "mtiwhich tmti.prd" shows you the full path of the actual data dictionary. Verify, that creation date of this file corresponds to your upgrade install; its length in most cases is different from the file "tmti.bak" at same location.

### 2.4.3.3 Data migration after model changes

In most upgrade cases there is no need to update existing data. Whenever this is required, you need first to do the entire upgrade (data dictionary, database schema, binaries) and after that you run a data update job.

To be 100% sure what you do, you have two alternatives for such an update job:

- a) You use a well-tested SQL script. This required good knowledge about how Teamcenter<sup>®</sup> Enterprise uses the databases.
- b) The development delivers an update API with the new release (this is standard Teamcenter<sup>®</sup> Enterprise behavior). This API must be organized, that it treats only data, which has not yet been upgraded, so that it may be call several times, until all data is processed. This API may be called by a menu option (with permit only to the super user) or through a specific client program. (⇐ This is the recommended version).

## **3 Facility Management**

### **3.1 Teamcenter<sup>®</sup> Enterprise Topology**

Teamcenter<sup>®</sup> Enterprise offers 3 Topology models:

- Single Database Environment
- Distributed Database Environment
- Federated System (option to each of the models above)<sup>6</sup>

See *OMF Administrators Manual, chapter 1 Configuration about concept, advantages and disadvantages of distributed installations.*

**Your capital is the **Data**,  
not the application!  
And **Processes** make money out of it!**

#### **3.1.1 Single Database Environment**

All meta data resides in one database, one Oracle instance, one Oracle user. No problems with cross database links. No need to select a database scope.

The Teamcenter<sup>®</sup> Enterprise services still may be distributed on several servers.

Good solution for development environment and for heavy centralized systems. The entire application depends on the availability of the only one DB server.

**Migration to a distributed environment is very hard, if a lot of data exists.** The Teamcenter<sup>®</sup> Enterprise manuals do not enough show the type of difficulties that could occur. If a future distribution may be planned, it is preferable to start in distributed option, but having only one "User Database" at the beginning, and having all DBs in the same Oracle instance.

#### **3.1.2 Distributed Environment**

The distributed environment allows spreading meta data over different servers and geographical sites. This grants a relative independence of major sites from each other, but sharing still the entire data pool.

To do so you must accordingly separate some data types and services to reflect that independence. The consulting of an experienced Teamcenter<sup>®</sup> Enterprise architect is recommended.

All sites have the identical data model and the identical applications (even if some use only a subset):

- Each site manages and hosts the data it creates or uses mostly.

<sup>6</sup> ) Federated option will most probably disappear in a future release.

- ❑ Each site has access to all data in the entire system.
- ❑ There is no difference in the relationships between local and remote objects.
- ❑ There is no difference in data handling between local and remote objects.
- ❑ Remote data handling is slightly slower than local.
- ❑ Any user can participate to any Life Cycle.
- ❑ Life Cycles may migrate data automatically between sites.
- ❑ Very minimal dependency of local activity from WAN and remote server availability.







The data is differently organized than in single DB environment. Even with only one “User Database” data is spread out in tree types of databases:

- Centralized data: to maintain uniqueness, rare query access, write access for specific Teamcenter® Enterprise actions on object having a “key”.
- Replicated administrative data: being rarely updated, but often (and often implicitly) read everywhere
- Distribute user data: data the users deal with are stored where it makes sense.

Special conventions are required to synchronize the backups between sites.

### 3.1.2.1 Database Types

## Distributed Databases Specialized Databases

	<b>Key-Database:</b> Holds Keys for all Database-Objects Holds Unique Id Info ...
	<b>Admin-DB:</b> Holds Info about Users, Groups, Hosts, File Systems, Rules, Roles, Conditions, ... (Has few Updates, but must always be accessible for read)
	<b>LCM-DB:</b> Holds LifeCycle Objects: LifeCycles, Processes, ... (Has very few Updates, but must be accessible for read where LifeCycles are used)
	<b>Operational DB:</b> Holds Events and Logs (This Database in general has only local meaning)
	<b>User DB:</b> Holds all Data Entries owned by Users or Vaults. (multiples) At User/Vault Definition the Database must be specified.
	<b>SuperUser DB:</b> Just an ordinary User DB but holds Data of the Super-User only

**Database Toplogy as well as which classes are assigned to which DB, is defined in “dbtop.prd”**

Starting with Teamcenter® Enterprise 3.2 there is no more specific LCM DB. The LCM objects are stored in the Admin DB. However, if you upgrade from earlier system, you may still have LCM DB's. To continue this you need to modify “toclcm.dat” in the install directory. (A migration increases performance and decreases complexity).

The Operational DB on the corporate server is now put together with the “SU” DB. On other Local Area Servers (which run an advancer service), put it into one local User DB.

### **3.1.2.2 Database Assignments**

Do **NEVER** assign a user or a Vault to a database containing a Replica Set. If you do so this data will go into the assigned DB, but not be replicated. This prevents you from the ability to add a database to this type later.

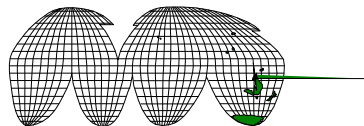
For performance reasons users should **NEVER** put a database in their SCOPE, which is not a user DB!

It is recommended that you use “translate” statements for all database names in your customization (preferably put it into file dbtop1.prd). Use an understandable translation for the DBs. But put all non-user DBs in parenthesis, so that you can advise user to not use DB-Names in parenthesis for their DB-scope!

### **3.1.2.3 Database independence**

## ***Distributed Databases***

***Improves Performance and Fault Tolerance in Large, Distributed Environments***



### 3.1.3 How to distribute Data in Distributed Databases

Remember the database types (see 3.1.2.1 Database Types) have 3 different characteristics:

- Special database, unique in whole system, no replication.  
Data which is mainly accessed in modify mode from all sites (like uniqueness info for keys, etc) is candidate for such a database. Sample KeyDB.
- Special database replicated.  
Data, which is not often updated, but heavily read by all users, is candidate for this type, like admin data, relatively fixed organizational data, ... For any DB modification all replications must be online available to complete. All query/read activities do only access the nearest copy. Replications are automatically kept in sync by Teamcenter® Enterprise. Administrators only are considered to do anything about it. Sample AdminDB.
- Distributed database (User database).  
For all the data classes not assigned to a specific database. There may be any number of them. Most common data in it are „Owned Items“ (typically documents and parts). Each data object is stored in exactly one of the, the ownership decides on which one: Each user and each Vault is assigned to a user-DB, which has to hold ALL data of that user or that vault.

A special case is the storage of Events in the Event-Queue table:

- Earlier versions of Teamcenter (Metaphase) used an extra DB, the OpsDB. This was not very useful and was as a default changed at Metaphase 3.2 level. Existing installation did often not change it.
- For best impact use, what Siemens PLM proposes as default:
  - Put all events into a user DB (one per Local Area). On corporate server site put it into “SU DB”. There is a toggle in “cfgedit2” database screen, which defines that the DB contains EventQ items; this will set the variable “OPERATE” to the correct value.
  - On Corporate server use “SU DB”.
  - On any other server use just one local user DB.
  - Any server storing EventQ items needs to run an “advancer” in this server or on an associated local server.
  - If a site still using an extra OpsDB wants to migrate, this may be done in two ways:
    - a) Change Configuration file and drop OpsDB, while all Events are worked (deleted).
    - b) Stop “advancer”, wait for only “pending” events in queue, dump all events and their relations, correct the dump file, load it in the new DB, drop OpsDB, change Config file, and restart Teamcenter.

Data is assigned through the data dictionary by object class to such a type. Thus it is not possible to have some items of an object class in a replicated DB and others of the same class in a distributed (user) DB. For such purpose you need to subclass the respective data class, so that one class may go into the replicated DB and the other class into the user DB.

Database assignment of data classes is subject to customization (file dbtop.prd). Thus you are free to assign any data class to a database type of your convenience. Just add the class to a (standard or custom) “replica set”. And the replica set is assigned to a database, or some replicated databases. After defining so, Teamcenter® Enterprise handles the data correctly; you do not need to change any method code.

If a data class is assigned to a “replica set”, then all its subclasses are assigned too, unless you assign a subclass explicitly later to another replica set. But there is no way to “unassign” a subclass. Only unassigned classes are kept in the standard distributed user DBs.

#### **Sample:**

If your company distinguishes between standard and built parts, it may be the case, that standard parts are very invariant (not often updated), but should be easily accessible any time anywhere, while normal parts are the stuff your engineers are working on.

You may decide the following:

- Create a subclass „StPart“ of Assembly (or Component) and assign it to a replica set for a replicated DB.

Queries to StPart are very effective, since you will query always in the nearest copy. If you put these standard Parts into your product structures, the behavior is much depending on how you decide for the relative part master items:

- a) Use the standard AssmMstr (or CmpnMstr), which are stored in a distributed user DB (or use a specific „StPrMstr“ but not assigned to a replica set): Both putting the StPart in your product structure and Structure Expand will run at medium speed, because the intermediate master may eventually be in a remote database. Putting to structure will not run, if this single DB is not accessible (expand show the noparking sign for the component).
- b) Use a specific „StPrMstr“ assigned to the same replica set as the „StPart“: Putting the part into your structure may be slow, since the relation between your part and the relative standard part master must be put into all replications to complete. Any expand is very quick, since it get all info from the nearest copy.

### **Note**

Be aware, that replication is hard in the data dictionary. For each HOST a variable in the config file defines, which copy to query. If this copy is unavailable, no automatic redirect is executed. But you can temporarily change the respective variable in the config file or override it through an OS variable. (Sample: Variable „ADMIN“ defines in which DB the „Admin“ class and subclasses are searched).

The respective variables have the same name as the Top Class(es) in the replica set, but all in upper case.

## **3.1.4 How to distribute Databases**

### **3.1.4.1 Conditions for distribution**

The distributed solution is advisable; if following numbered conditions are true:

1. Must guarantee unique identifiers within corporation.
2. Share corporate data and/or processes.
3. Federated approach is not strong enough or may pose problems in future releases.
4. At least one of the following letters is true:
  - a) You have a big number of users, working in separate areas (projects, departments, group companies, business areas,...) that share only 20% of their data with other areas: use user DBs by area. This increases performance. (Access to own data is faster than in centralized case, access to other data is slower; which is the trade off in your case?)
  - b) Users are geographically distributed with WAN's between sites and over 60% of data are mainly used by determined sites (even if site changes between life steps like development and production; it may migrate to new location at specific state change).
  - c) You want to grant physical ownership of data by legally separated group companies. (Ease of company restructuring, ease of switch to federated system, when this offers all needed features over installation borders, ...).

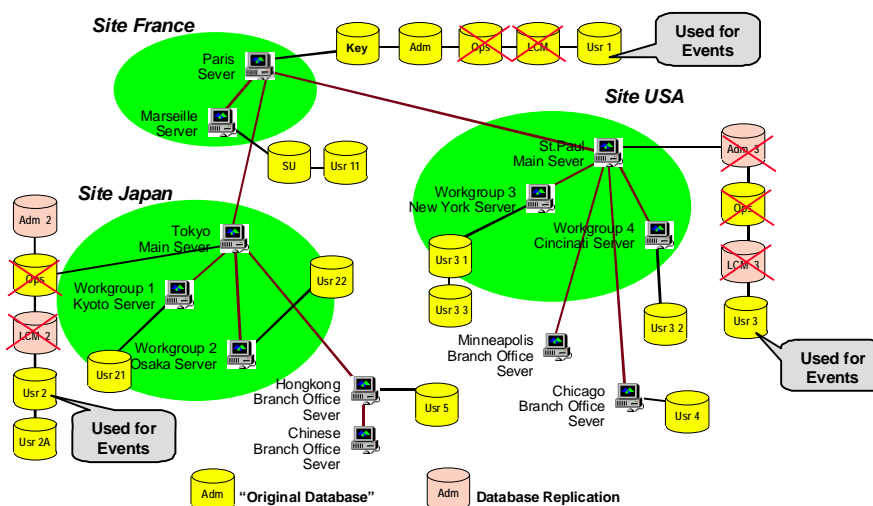
Caution: in overloaded networks a bad distribution will be worse than a centralized base.

### 3.1.4.2 Which Data Base to put Where?

- The replicated databases (AdminDB) should have a copy on the main server of each major<sup>7</sup> site (a Local Area Server).
- The KeyDB should reside on the main server of the major site (corporate server), where most “Creates”, “CheckOut/Ins”, “Revises” and “Deletes” of objects with a key assigned could happen. (These are the actions that need an implicit key access; the KeyDB is NEVER used for Queries! Thus do never put it into your DB scope!)
- Have one single and user DB assigned for Event Queue<sup>8</sup> on each major site. On same host the advancer service should run. Caution: all hosts served by this major site server need the configuration variable “OPERATE” to be set to the name of that DB! If not, this results in a lot of additional network traffic.
- User DBs should be placed at least at each major site. User DBs on minor sites or even work group servers may be considered for performance reasons (this creates 80% of the DB accesses locally, thus less net resource needed, ...).  
The SU-DB is just a user DB, containing mainly test data for administrators, which are not intended to be seen by ordinary users.

### 3.1.5 Sample distribution schema

#### Distributed Layout



### 3.1.6 Database Topology using the Configuration Editor

All these details of Configuration are defined in the “cfgedit2” of Teamcenter<sup>®</sup> Enterprise correctly.

<sup>7</sup>) A **major site** is a site, which is separated from other major sites by a big distance and/or by a WAN with poor bandwidth or poor reliability.  
A major site may contain several **minor sites** with own databases, interconnected by well performing and stable networks.

<sup>8</sup>) SUIDB on corporate, a user DB on Local Area Servers.



### 3.1.7 Federated Systems

Federated systems are different independent Teamcenter® Enterprise installation. They do not share the applications, nor the entire data model. They can give access to partial data for foreign Teamcenter® Enterprise systems.

Shared data classes must share the persistent part of their data model.

Open or close connections are on 'the fly operations' of administrators of each side.

This allows queries beyond system borders. Proxy objects in the local base represent remote objects. Local user must be mapped to remote federal users and vice versa.

A few relationships are allowed across systems (like Attach, see Teamcenter® Enterprise standard manuals). But any structure or successor relations are not allowed cross border.

"Copy Object" can be used between systems.

This is a starting point for "virtual enterprise" PDM. It is an appropriate solution for often changing company associations as well as for supplier and customer data exchange, where you give access to a subset of your data during a certain time; but you maintain the data ownership.

In the current version of "Federated Systems" Life Cycles cannot span systems.

Each of the partner systems may be mono base or distributed. The only condition is a common basic model for the data type to be exchanged.

**Caution:** The federated model will no more be supported in this form in the future common platform for Teamcenter Enterprise and Engineering!

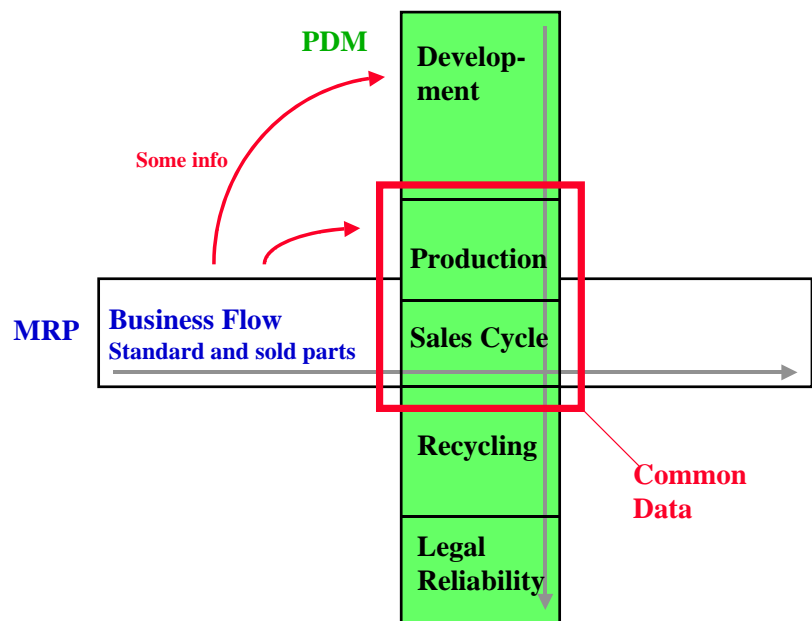
### 3.1.8 PDM and MRP/ERP

Standard products exist for such integration between Teamcenter® Enterprise and Oracle Manufacturing (by ITI) or Teamcenter® Enterprise and SAP or Baan.

The two systems have different scope of data, but share some information. They are really orthogonal systems to each other.

Before installing any bridge between such a tool you need to decide some aspects:

- Who shall own the common data?
- Who can create parts?
- Who checks the uniqueness of part numbers? Who assigns part numbers?
- Do you intend a master-slave version or a peer to peer version?



### 3.1.9 Configuration Editor – Instance preparation

Starting with Metaphase 3.1 the Teamcenter® Enterprise configuration is maintained exclusively through a „Configuration Editor“. Following restrictions apply to the use of this (required) new tool:

1. The config editor may be called ONLY at the corporate server<sup>9</sup>.
2. It maintains the dbtop.prd file, the mti.prd file and the config.cfg file for the entire instance, as well as some internal files needed for the Teamcenter<sup>®</sup> Enterprise installation. Neither of these files should be manipulated manually anymore!
3. In \$MTI\_ROOT/install is a new subdirectory, containing a subdirectory for each host in the network, with all the files that need to be copied there.
4. This tool allows creating “network views” which communicate to each other exclusively through single specified hosts (network bridges).
5. If you add class replication in your customization, this can no more be done manually through “dbtop.prd”, but must be part of the installation set (as described by Teamcenter<sup>®</sup> Enterprise manuals): see file “tocCUS.dat”, key words “SEPDB” and “CLASS”.
6. Add your specific variable setting for the customization (and eventual special service definitions) in file “cfgcus.dat”.

### **3.1.10 Files Systems / Installation Location**

First of all an OS user (with admin rights) must be assigned as “trusted” user on each Teamcenter Enterprise server (try to use same OS username everywhere per instance). This user is the owner of the Teamcenter Application and is the corresponding OS user to the Teamcenter “super user”. It is not allowed, that this OS user is assigned to another Teamcenter user too, or being the owner of more than one Teamcenter instance. Any violation of these rules will render the Teamcenter Application out of function.

Physical organization of disk space for Teamcenter<sup>®</sup> Enterprise usage must be carefully planned. A couple of rules apply for effective organization of directory structures:

#### **\$(MTI\_ROOT)**

This is the top directory of Teamcenter<sup>®</sup> Enterprise installation. **This must be NOT the HOME directory** of any normal user; it may be a specific directory within trusted user’s HOME directory. This directory contains binaries, sources, libraries, support and configuration files. Use an application directory for this like for other applications:

example “/applications/TeamcenterEnterprise” or  
“/applications/TeamcenterEnterprise/<Version>” or  
“/applications/TeamcenterEnterprise/<instance>” or within HOME of the trusted user.

It is recommended to create (under UNIX) a link “/TCE” or “/TCE<Instance>”, which points to the actual productive MTI\_ROOT.

For ease of use in the application support and maintenance it is highly recommended to have as much as possible the same file paths for the same purpose on different servers of the same Teamcenter<sup>®</sup> Enterprise instance.

#### **“File Systems” for Work Locations**

##### a) UNIX

Best practice is to register the base directory of all users as Teamcenter<sup>®</sup> Enterprise File System (like “/users” or “/var/home”, depending on OS), in which users create their Work Locations, i.e. each user can (without administrator intervention) register any of his directories as Work Location.

---

<sup>9</sup>) Caution, if same host is Corporate server for more than on Teamcenter instance (like test instances): “cfgedit2” – due to Java communication restrictions – cannot be called for more than one Teamcenter instance simultaneously.

A user can then register any directory within his home directory as a Teamcenter® Enterprise Worklocation. This is nearest to his usual work.

b) Windows

To avoid the need for an administrator to go to a Window workstation for registering the C: drive as file system, or to contact a user to log in while he creates the file system, it is in most environment much more convenient to register the shared drives (which contain users private area)

### 3.3 Teamcenter® Enterprise Administration

Administration of a Teamcenter® Enterprise system contains a lot of single tasks, which can be distributed on different roles, sites and individuals. Here a list of missions grouped by some level:

#### 3.3.1 General Level

- OS System Administration
- Teamcenter® Enterprise Ownership Management
- Oracle DBA
- Database Topology Management
- Network Management
- Configuration Files
- Backup Management

#### 3.3.2 Level 1 Administration: *site independent*

- Conditions and Message Access Rules
- User Groups and Roles
- Workflows
- Events
- Notification Rules
- E-Mail Support
- Server Installations and Upgrades

#### 3.3.3 Level 2 Administration: *site dependent*

- Vaults and Vault Locations
- Location Selection Rules
- Event Log Maintenance
- Mail Log Maintenance
- Client Installations
- Client Upgrades

#### 3.3.4 Level 3 Administration: *site dependent*

- Hosts and Host Groups
- File Systems
- Users
- Role and Group Assignments
- Database Cleanup, before a PC disappears!

#### 3.3.5 Operation Management

- Problem Reporting to Teamcenter® Enterprise
- Help Desk for Users

### 3.4 User tips

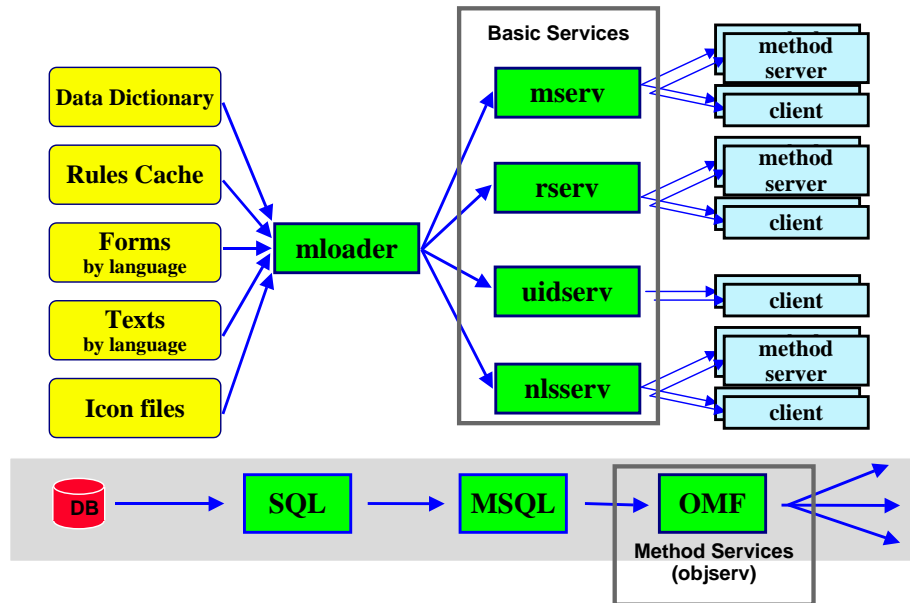
- Queries over more than one database (having more than one database in the users "SCOPE") tend to be slow, especially if some are remote bases.
- Users should only select "User" databases in their scope. Selecting one of the following database types in the scope "key", "admin"<sup>11</sup> (is no error, but) will slow down all your queries, because object that must reside in a user data base will be searched where they cannot be. Object classes of a replica set (see DB topology file "*dbtop.prd*") will anyway automatically be searched in the correct database, if replicated according variables are set in the *config* file for the specific host running *omf* service.

<sup>11</sup>) or a customized replica set database.

## 4 Teamcenter® Enterprise Concepts

### 4.1 Dictionary and Data Services

# Dictionary + Data Services

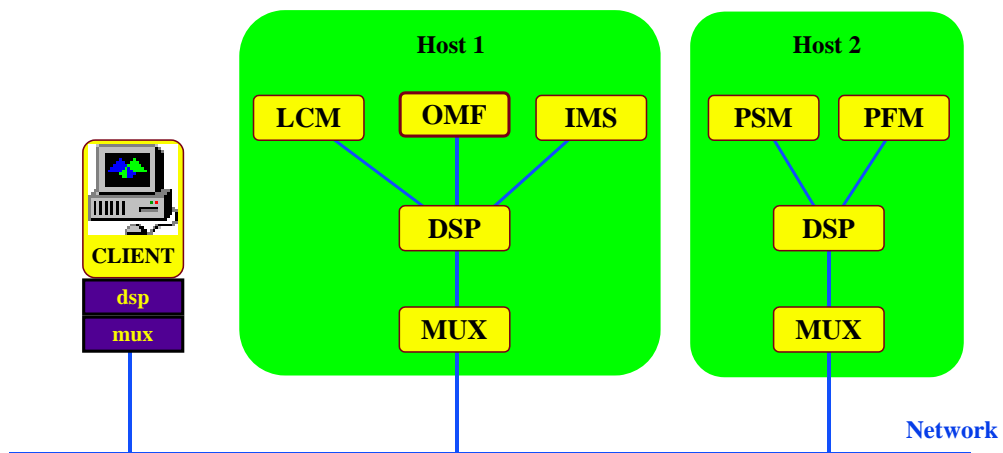


Files are read by mloader through os\_serv. Info goes to data services (or general services), to method services and client programs.

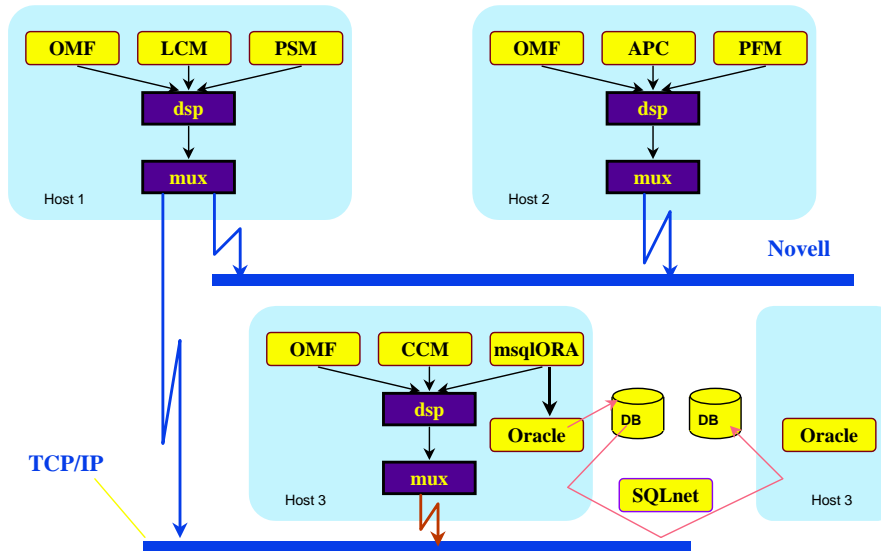
All database access goes through OMF, MSQL to Oracle and back.

DSP is the “Request Broker” (object theory). If an access has to be passed to another host, this is done through MUX. MUX is a software multiplexer, reducing all traffic between two individual hosts into one data stream.

## Distributed Servers

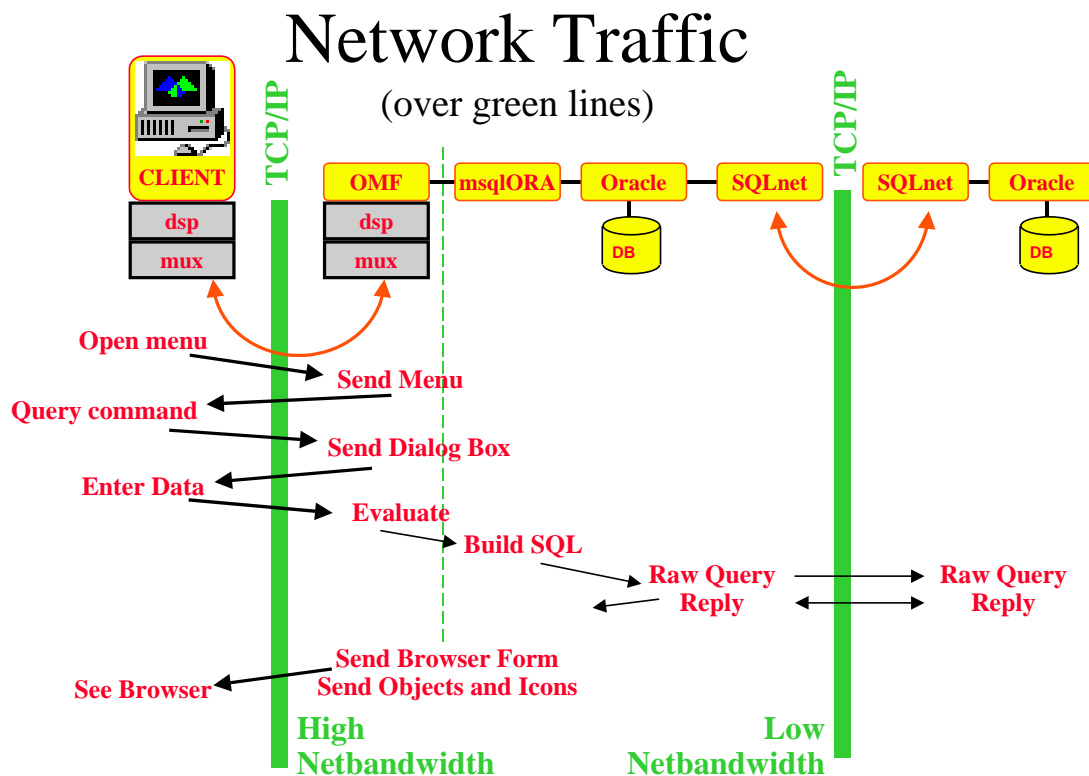


## Metaphase Communication



If more than one network type is involved into one Teamcenter® Enterprise installation, then at least one host must be defined in the config file for both network types. This host will then be the automatic bridge by its MUX.

## 4.2 Network usage of Teamcenter® Enterprise



### **4.3 Compromise**

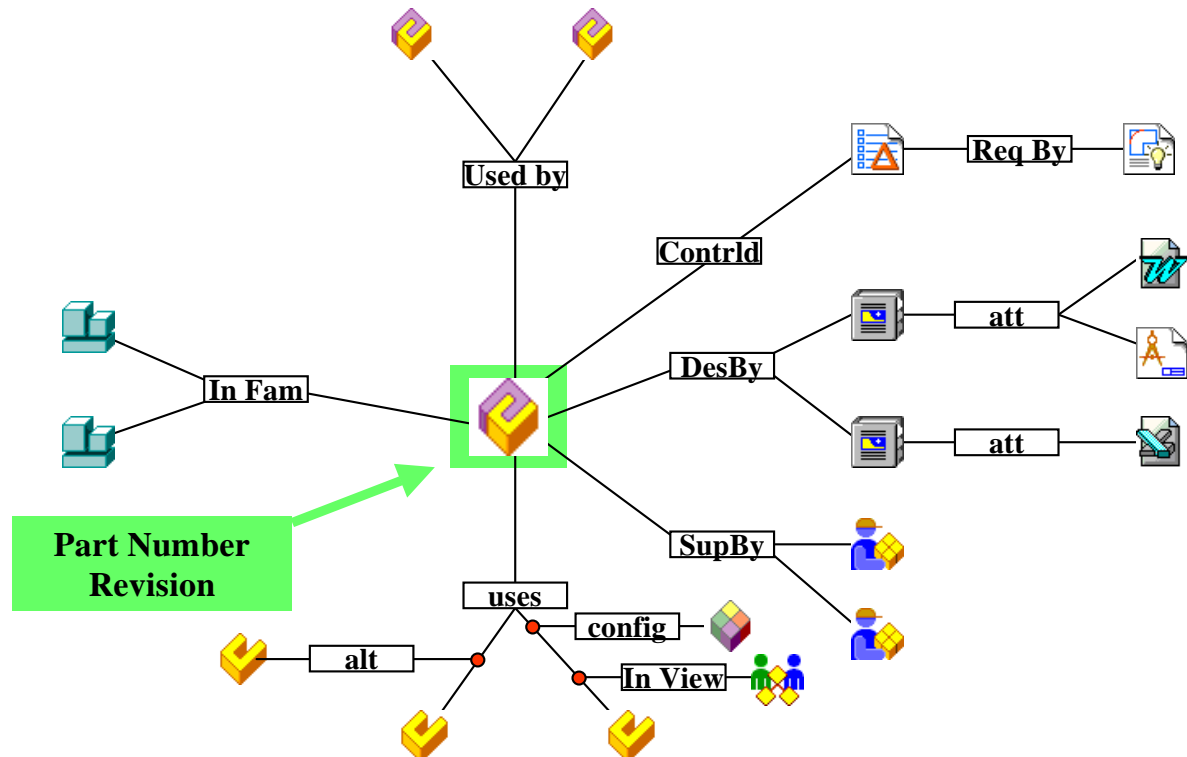
Finally any selected solution must be a compromise between mutually excluding requirements for data integrity, fault tolerance and performance.

#### ***Distributed Metadata***

***Distributed Database Solutions Require Careful  
Balancing of Competing Factors***



Or in a more complete schema, the product (part) holds within a whole network of relations:



## 6 Processes

Well organized and structured data, accessible every where in the company, is prerequisite for efficient work. It prevents from lengthy search loops and extra costs due to not found data (may be vital in legal liability procedures). And it helps for re-use of already developed parts instead of re-inventing the reel again and again.

A major gain depends on well-organized processes upon this data.

**Well-organized and structured DATA  
is a prerequisite.**

**Major gain comes from efficient  
PROCESSES upon this data.**

Teamcenter® Enterprise supports a very flexible way of mapping your processes into the system to heavily support or even enforce the company's process rules. Change management (earlier CCM, then CFM module, newer Change Solution module and tasks) and Workflow management (LCM module) are the tools for this topic.



Thus the system can support or enforce it, but you need to decide and design your processes. They are probably your company's success secret.

## 7 Teamcenter® Enterprise Development Organization

### 7.1 Instances

#### 7.1.1 Teamcenter® Enterprise Instance

A Teamcenter® Enterprise instance is one instantiation of a Teamcenter® Enterprise application, generally distributed over several computers, locations and platforms, servers and clients. On all these hosts the same version of programs is running and they communicate through one (or more) of the supported network protocols (most commonly TCP/IP).

On each host there is a system account defined as "trusted user"; which is the owner of the application on OS level. **One trusted user must own only one instance of Teamcenter® Enterprise.**

All hosts/programs of one instance have a common configuration file, which cannot be shared with other instances. They communicate over one distinct IP port, which is reserved for the instance. They have their own one common database (or set of databases).

All cross host communication is organized via a communication multiplexer program "MUX". Any Teamcenter® Enterprise instance has exactly one MUX program on each participating host (server or client).

Following table illustrates this:

instance property	same instance	different instance
configuration file	same	different
IP port	same	different or in mutually unreachable networks
data base(s)	same	different
binary version	same	may be different. If same and on same host, may be shared <sup>12</sup>
communication	yes	no, except for „Federated Systems“
platform	any	any
MUXes on same host	1	1 per instance
connect to DB instance	direct or indirect, but always on same DB(s) of one or more DB instances	different DB(s) on same or different DB instances
trusted user	Preferably everywhere same	different at least on same host
data model	same	same or different

Each instance may separately be stopped and started.

<sup>12</sup>) Generally shared binaries are not recommended, except for mutually dependent development instances! This reduces the needed disk space. But you can no more upgrade individually.

A Teamcenter® Enterprise instance may be distributed over several hardware servers and can communicate with any number of DB instances.

**Exception:**

To facilitate a Teamcenter® Enterprise release upgrade, two different levels of the standard product may run in parallel for limited amount of time using same databases (with well-defined constraints described in Teamcenter® Enterprise upgrade manual). The two version are two separate instances (in terms of the above definition); but they share the database (or set of database) and their data models may have only limited differences.

**7.1.2 Oracle Instance**

To simplify the explanation we will talk here only about Oracle as underlying DB system.

An Oracle instance is an Oracle-SID. Each instance may be started and stopped separately. They can communicate to each other only via SqlNet.

With exception of mirroring in clusters an Oracle instance cannot span hosts. But they can communicate through SqlNet.

An Oracle instance may serve several Teamcenter® Enterprise instances (using different Oracle user names), even this is not recommended. A Teamcenter® Enterprise instance may be served by several Oracle instances.

**7.1.3 Comparison between DB terms of Oracle and Teamcenter® Enterprise**

Teamcenter® Enterprise and Oracle use different expressions for same items and same expression for different items:

<b>Teamcenter® Enterprise</b>	<b>ORACLE</b>
instance = IP port, config file, data dictionary	instance = data base, SID
data base	Oracle user
object class, relation, cluster	table
attribute	column
object	row
Index <sup>13</sup>	index + index + ... (over several tables)
key <sup>14</sup>	unique index + unique index + ... + entry in key-table

**7.2 Logical structure of Teamcenter® Enterprise instances**

Use different Teamcenter® Enterprise instances for development, test, training and production. Responsibility between all development and the rest must clearly be split.

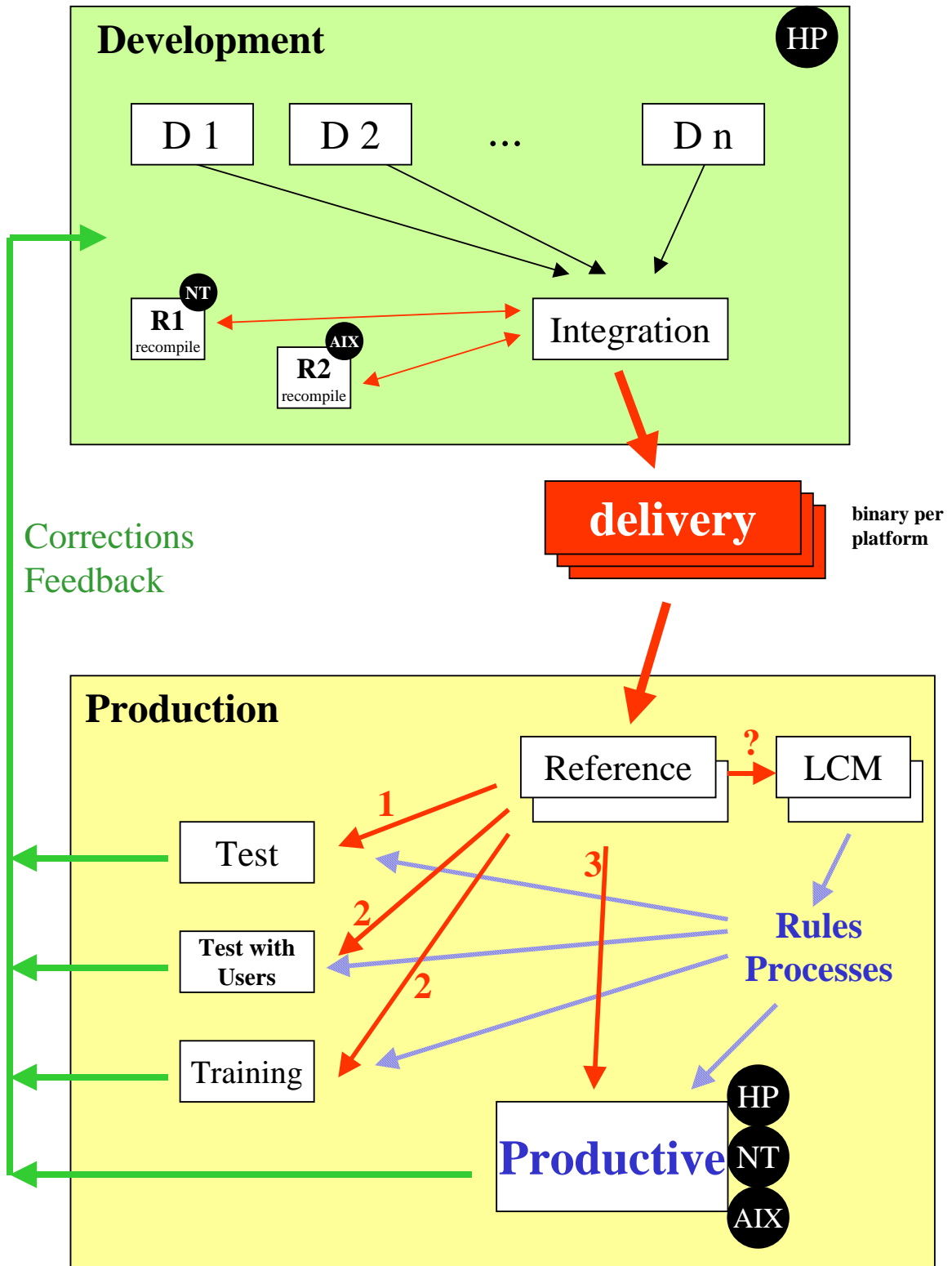
The following table shows a typical set of Teamcenter® Enterprise instances in a company maintaining their own customization. This schema includes the recommended rollout path.

<sup>13</sup>) Used for query optimization only.

<sup>14</sup>) Used exclusively to grant uniqueness (over several tables and Oracle instances).

Some instances may be shared for several topics (like “test with users” and “training”), while they are not used simultaneously. But the responsibility areas should not overlap. **In general developers should be kept off from productions systems.**

Option: To maintain the same set of rules and processes all the instances under production control should have a master instance for this topic (here named “LCM”). And any respective changes are done ONLY in this instance and then exported to the other instances. The export/import process is not very simple.



## 8 Platform specifics

### 8.1 AIX

#### 8.1.1 Shared Libraries

The behavior of “shared libraries” has some specifics in AIX:

- **"Asynchronous IO device" must be turned on in SMIT**; otherwise shared libraries cannot be loaded. This in general is already required to get running the Oracle installation. But on non-DB hosts you need to verify respectively to set it properly.
- **Changing a shared library** (install a standard Teamcenter® Enterprise patch or a new version of your custom servers shared library) is no problem, if you boot the server after modifying a shared library.

**If you do not boot, you probably will be surprised, that still the old library is executed, even so this old library does no more exist anywhere on your system.**

This is due to the fact that the library is still in memory. The respective file is just its swap out image. As long as the shared memory is big enough no reload occurs and you really will execute it. Normally you should not be able to overwrite the library file while it is in memory. Nevertheless sometime you may do it.

- To clean out the memory (either to be able to overwrite the library file, and essentially for loading late the new version) you need two commands (executable under root only):  
**“slibclean”**: unloads any not actually active library from shared memory.  
**“genkld”**: lists all the shared memory content including the library names. This is to verify that all Teamcenter® Enterprise libraries have been unloaded.  
If after “slibclean” the program “genkld” still shows Teamcenter® Enterprise libraries loaded (in case of very abnormally died services) you need unconditionally to boot the system.
- Custom server libraries in **customization environments** should not have read permit to “others”. This indicates to the system, that the library is a “private shared library” and unloaded when you stop your service. This means it works like a shared library so that you may work with it and test it, but it becomes not really shared.
- On AIX cluster systems (i.e. SP line) the default Teamcenter® Enterprise IP port is in the IBM reserved range for Cluster communication. IBM requests any non-root application port to be higher than 20'000.

### 8.2 HP-UX 10 and later

Loader reactions on trial to load missing symbols on HP-UX are very misleading. The visual error message says **“not enough memory”**. This problem can occur after either re-linking a standard server due to a patch installation, or after recompilation of a custom server. In reality this often means that loader could not satisfy the load request for all required symbols.

- **Cause of “Missing symbols”**: either at least one of the required libraries is not know to the loader (path, library path), or a function or API method called in the custom server is misspelled, thus cannot be found in any of the libraries
- **Detection of “Missing symbols”**: Start MUX and DSP normally, then start the server (service) manually by typing its program name on the command line. In this case the system will tell you, which symbol could not be loaded.