



Version

4.0

» Overview Manual

May 2025

Author Tecnoteca srl

www.tecnoteca.com

ENG

www.cmdbuild.org

No part of this document may be reproduced, in whole or in part, without the express written permission of Tecnoteca s.r.l.

CMDBuild ® uses many great technologies from the open source community: PostgreSQL, Apache, Tomcat, Eclipse, Ext JS, JasperStudio, Enhydra Shark, TWE, OCS Inventory, Liferay, Alfresco, GeoServer, OpenLayers, Quartz, Xeokit.
We are thankful for the great contributions that led to the creation of these products.

CMDBuild ® is a product of Tecnoteca S.r.l. which is responsible of software design and development, it's the official maintainer and has registered the CMDBuild logo.



CMDBuild ® is released under AGPL open source license (<http://www.gnu.org/licenses/agpl-3.0.html>)

CMDBuild ® is a registered trademark of Tecnoteca Srl.

Every time the CMDBuild® logo is used, the official maintainer "Tecnoteca srl" must be mentioned; in addition, there must be a link to the official website:

<http://www.cmdbuild.org>.

CMDBuild ® logo:

- cannot be modified (color, proportion, shape, font) in any way, and cannot be integrated into other logos
- cannot be used as a corporate logo, nor the company that uses it may appear as author / owner / maintainer of the project
- cannot be removed from the application, and in particular from the header at the top of each page

The official website is <http://www.cmdbuild.org>

Contents

1. Introduction.....	5
1.1. The application.....	5
1.2. Official website.....	6
1.3. CMDBuild modules.....	6
1.4. Available manuals.....	6
1.5. Applications based on CMDBuild.....	6
2. An open source CMDB, and so much more.....	8
2.1. What is a CMDB.....	8
2.2. Why use a CMDB.....	9
2.3. Implementation guidelines.....	9
2.4. Open Source.....	10
2.5. Not only a CMDB.....	10
3. CMDBuild philosophy.....	11
3.1. Detachment between "core" and "business logic".....	11
3.2. The logic of metadata.....	12
3.3. The CMDBuild conceptual schema.....	12
3.4. The CMDBuild architecture.....	14
3.5. The CMDBuild Service Bus for the interoperability.....	14
3.6. The configuration mechanisms and the base features of CMDBuild.....	15
3.7. CMDBuild elements.....	16
3.8. High availability configuration.....	16
4. CMDBuild features: main mechanisms.....	18
4.1. Configurability of the data model.....	18
4.2. Editor and workflow engine.....	18
4.3. Editor and report engine.....	19
4.4. Dashboard configuration.....	19
4.5. Configuration of GUIs and custom business logics.....	20
5. CMDBuild features: native features.....	21
5.1. Relationship paradigm.....	21
5.2. Visual graph.....	21
5.3. Documents archive.....	22
5.4. GIS and BIM georeferences.....	22
5.5. Data history.....	22
5.6. Scheduler for automatic tasks.....	23
5.7. User profiling and Multitenant configuration.....	23
5.8. Data Import / Export through files.....	24
5.9. Email management.....	25
5.10. Schedule.....	25
6. CMDBuild features: user interfaces.....	26
6.1. Web desktop GUI for operators.....	26
6.2. Web desktop GUI desktop web for the administrator.....	27
6.3. mobile APP for operators.....	28
6.4. Self-Service portal for final users.....	28
6.5. REST Webservices.....	29
7. The IT Governance verticalization: CMDBuild READY2USE.....	30
7.1. The product.....	30
7.2. Data model.....	30

7.3. The processes and the Self-Service portal.....	31
7.4. Connectors with Automatic Discovery and Monitoring systems.....	32
7.5. The report system.....	33
7.6. The mobile App.....	34
8. The Facility Management verticalization: openMAINT.....	35
8.1. The product.....	35
8.2. Data model.....	36
8.3. Maintenance processes and the Self-Service portal.....	36
8.4. The interaction with GIS and BIM systems.....	38
8.5. The report system.....	39
8.6. The mobile App.....	40
9. Appendix: Glossary.....	41

1. Introduction

1.1. The application

CMDBuild is an open source web environment for the configuration of custom applications for the Asset Management.

On the one hand, it provides native mechanisms for the administrator, implemented in a "core" code which has been kept separated from the business logic, so that the system can be configured with all its features.

On the other hand, it generates dynamically a web interface for the operators, so that they can keep the asset situation under control and always know their composition, detachment, functional relations and how they update, in order to manage their life-cycle in a comprehensive way.

The system administrator can build and extend his/her own CMDB (hence the name of the project), modelling the CMDB according to the company needs; a proper interface allows you to progressively add new classes of items, new attributes and new relations. You can also define filters, "views" and access permissions limited to rows and columns of every class.

Using external visual editors, the administrator can design workflows, import them into CMDBuild and put them at operators' disposal, so that they can execute them according to the configured automatisms.

Similarly, using external visual editors, the administrator can design various reports on CMDB data (printouts, graphs, barcode labels, etc.), import them into the system and put them at operators' disposal.

The administrator can also configure some dashboards made up of charts which immediately show the situation of some indicators in the current system (KPI).

A task manager included in the user interface of the Administration Module allows you to schedule various operations (process starts, e-mail receiving and sending, connector executions) and various controls on the CMDB data (synchronous and asynchronous events). Based on their findings, it sends notifications, starts workflows and executes scripts.

The interoperability with other systems is managed through the CMDBuild BUS Service, called WaterWAY.

The Plugin Manager enables CMDBuild to be a highly extensible and flexible platform, capable of adapting to the specific needs of any organization through centralized and intuitive management of integrations and customizations.

Thanks to the REST API integration with the open source solution Alfresco you will be able to attach documents, pictures, videos and other files. It is possible to define document categories differentiated by class and sets of metadata differentiated by category. Please note that integration with the commercial document management product SharePoint is also available.

There is also a Scheduling, which can be supplied both automatically when filling in a data card and manually. This Scheduling will manage single or recurring deadlines related, for example, to certifications, warranties, contracts with customers and suppliers, administrative procedures, etc.

Moreover, you can use GIS features to georeference and display assets on a geographical map (external map services) and / or on vector maps (local GeoServer and spatial database PostGIS) and BIM features to view 3D models (IFC format).

The system includes a REST webservice, so that CMDBuild users can implement custom interoperability solutions with external systems.

CMDBuild includes a mobile interface (for smartphone and tablet). It is implemented as multi-platform app (iOS, Android) and is able to interact with the CMDDB through the REST webservice (supplied with non-open source license to the users that subscribe the annual Subscription with Tecnoteca).

CMDBuild is an enterprise system: server-side Java, web Ajax GUI, SOA architecture (Service Oriented Architecture), based on webservice and implemented by using the best open source technologies and following the sector standards.

CMDBuild is an ever-evolving system, which has been released for the first time in 2006 and updated several times a year in order to offer more features and to support new technologies.

1.2. Official website

CMDBuild has a dedicated website: www.cmdbuild.org

The website gathers a lot of documents on technical and functional features of the project: brochures, slides, manuals (see next paragraph), testimonials, case histories, newsletters, forums.

1.3. CMDBuild modules

The CMDBuild application includes two main modules:

- the Administration Module for the initial definition and the next changes of the data model and the base configuration (relation classes and typologies, users and authorization, dashboards, upload report and workflows, options and parameters)
- the Management Module, used to manage cards and relations, add attachments, run workflow processes, visualize dashboards and execute reports

The Administration Module is available only to the users with the "administrator" role; the Management Module is used by all the users who view and edit data.

1.4. Available manuals

This manual is for those who need certain first introductory information on CMDBuild and who are interested in knowing the general philosophy of the project.

You can find all the manuals on the official website (www.cmdbuild.org):

- system usage for operators ("User Manual")
- system administration ("Administrator Manual")
- installation and system management ("Technical Manual")
- workflow configuration ("Workflow Manual")
- webservice details and configuration ("Webservice Manual")

1.5. Applications based on CMDBuild

Tecnoteca has used the CMDBuild environment in order to implement two different pre-configured solutions:

- CMDBuild READY2USE, for the management of assets and IT services, oriented to internal IT infrastructures or services for external clients (www.cmdbuildready2use.org) according to the ITIL best practice (Information Technology Infrastructure Library)
- openMAINT, for the inventory management of assets, properties and related maintenance activities (www.openmaint.org)

Both applications are released with open source license, except for certain external components (data sync connectors, Self-Service portal, mobile APP, etc.), that are reserved to the users that subscribe the annual Subscription with Tecnoteca.

2. An open source CMDB, and so much more

2.1. What is a CMDB

A CMDB (Configuration Management Data Base) is a storage and consultation system that handles the asset information of a company.

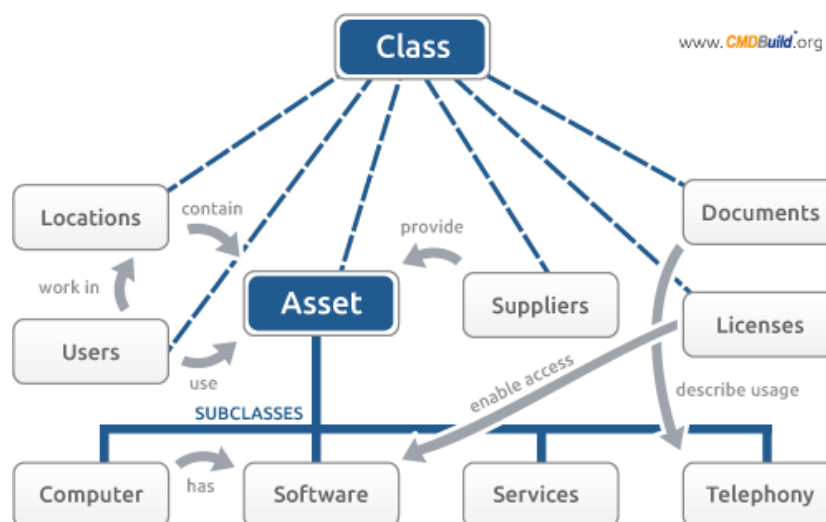
The idea of CMDB was born in an IT environment (it is a fundamental component of ITIL best practice - Information Technology Infrastructure Library) and can be applied in every context where assets must be known, managed and controlled.

It is the official central repository and provides a consistent view of the items that have to be managed.

It's a dynamic system that represents the current situation and knowledge of the inventory of assets and the related connections.

What kind of asset can a CMDB manage?

- information assets: hardware (computers, peripherals, networking, telephone equipment), software (basic, environment, application software), services for users and other internal and external resources
- assets related to properties (buildings, plants, technical devices, furniture)
- assets related to manufacturing plants (factories, plants, machinery)
- other kinds of assets (vehicles, electro-medical, etc.)



Questions the CMDB answers to:

- where is an asset or a CI (configuration item) ?
- who uses it ?
- what does it belong to ?
- what is it made up of ?

- what and where are other similar CIs ?
- what has happened in the CI life-cycle ?
- what other CIs does a possible change influence ?
- which are the activities waiting for my intervention ?
- how much does the CI management cost ?
- what are the assets that are mainly interested at a certain assistance ?
- what's the status of the compilation of CI cards ?
- which are the intervention SLAs following intervention requests on CIs ?
- which are the KPI values that evaluate the performance of my organization ?

2.2. Why use a CMDB

The use of a CMDB allows to control the situation of managed assets, knowing at every time their composition, dislocation and functional relations.

Missing or outdated information means useless costs, redundant operations, delay in the problem solving, hindrance to the company activities.

Keywords of a CMDB are response time and system control.

A CMDB - for any type of Asset Management application - allows to:

- reduce problems in your infrastructure
- solve more rapidly the remaining problems
- solve at the first answer a major problem percentage, without involving too frequently the expert staff
- keep track of every data-change operations
- have a repository for statistical data, useful to check SLA

In other words, costs decrease and service quality improves.

2.3. Implementation guidelines

The introduction of an application for the management of your assets must be adequately prepared in terms of organization and formation in order to reduce the risks of failure or refusal for the tool.

By managing the project, it is important to:

- adopt guidelines aimed at a gradual and flexible implementation
- choose a detail level proportional to the available organization needs, and to human, financial, information and technology resources (if you have a really extensible and modular system it is better for you to make use of following autonomous extensions of the support data model, rather than overloading the initial schema)
- introduce the application in an organizational system based on well-outlined and formalized responsibilities, procedures and roles

A successful project must consider impacts and changes introduced by the system and must acquire the explicit approval of organization's managers.

2.4. Open Source

Open source is better !

An open source solution allows to:

- avoid (high) licensing fees
- provide the source code in order to reduce the dependency of the supplier
- grant major freedom of use, in different seats and for more various needs
- reuse evolutionary features, developed on demand by other users
- cooperate with a community of users who share their experiences in terms of organization and implementation (data model, processes, etc.)

An open source solution is not free of charge, both in terms of external services and internal costs for the activation, but numerous and authoritative sources recognize their lower total required cost (TCO, Total Cost Ownership).

An open source product is not necessarily a good product, but the trade evolution provides a lot of open source solutions technologically more advanced and better supported than the related proprietary applications

Open source components, integrated or interoperable with CMDBuild, are selected among the most complete, widespread and developed ones.

2.5. Not only a CMDB

Despite of its name, CMDBuild is not only a Modeling Environment for CMDB applications, i.e. limited within the management of a database of items (Configuration Item).

CMDBuild mechanisms allow you to manage Configuration Items and their conditions of use within the entire life cycle: processes, business rules, documents, reports, georeferences, interoperability with other systems, etc.

With reference to the IT service area, as described by ITIL (Information Technology Infrastructure Library) and implemented in the CMDBuild READY2USE verticalization, CMDBuild is not only a CMDB system (Configuration Management Data Base), but rather a CMS system (Configuration Management System).

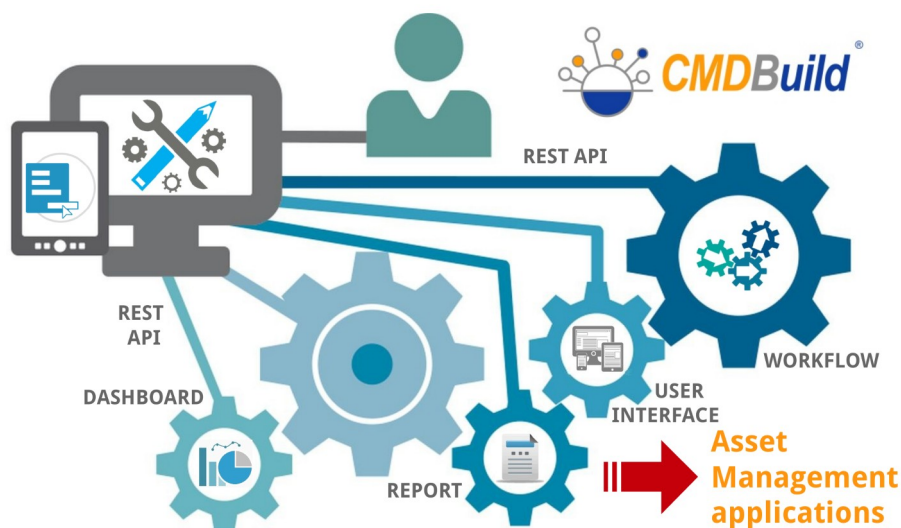
3. CMDBuild philosophy

3.1. Detachment between "core" and "business logic"

The Asset Management of enterprise systems is a complex problem, regardless of the application domain.

Each company / institution has its own organization and peculiarities, which have to be recognized by the management software.

CMDBuild, understood as "engine" where the desired application solution is configured, allows you to adapt the software to the way of working and not viceversa.



Also, it allows you to do that without changing the code of the software application, that would prevent from any possibility of doing the update for the next versions.

How does all this happen ?

It happens by separating the "engine" from the "business logic", i.e. encapsulating the peculiarities of the information entities and of the processes inside a data model that can be configured from a user interface and inside XML indicators related to your processes and reports.

Configurability and flexibility were pursued as main project standard, allowing the system administrator to define the optimal application for his/her operative needs.

A system based on the configurability has its clear advantages:

- it can adapt to your organization and work procedures
- possibility of a gradual carrying out of the system, reducing the organizational impact
- easy response to organizational and technological editing
- more independence for users
- same "core" code (managed "per product") used in different vertical solutions

3.2. The logic of metadata

From a technical point of view, the configurability is based on a wide use of "metadata".

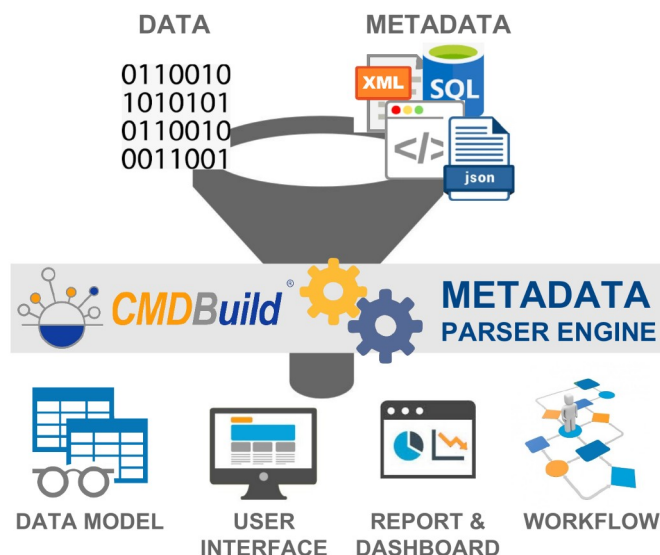
All CMDBuild behaviours are based on metadata configured by the administrator:

- definition of entities and attributes in the data model design, filed in the database
- workflow design and automatisms, created in a visual way, saved as XPD files and filed in the database
- design of reports, created in a visual way, saved as XML files and filed in the database
- dashboard configuration through parameters filed in the database

The CMDBuild "core" code interprets metadata creating also the web interface for operators, who have the possibility of updating the CMDB, start and advance processes, execute reports, view dashboards, etc.

The main advantages of this solution are as follows:

- maintainability
- guaranteed evolutions (version updates)
- economic customizations



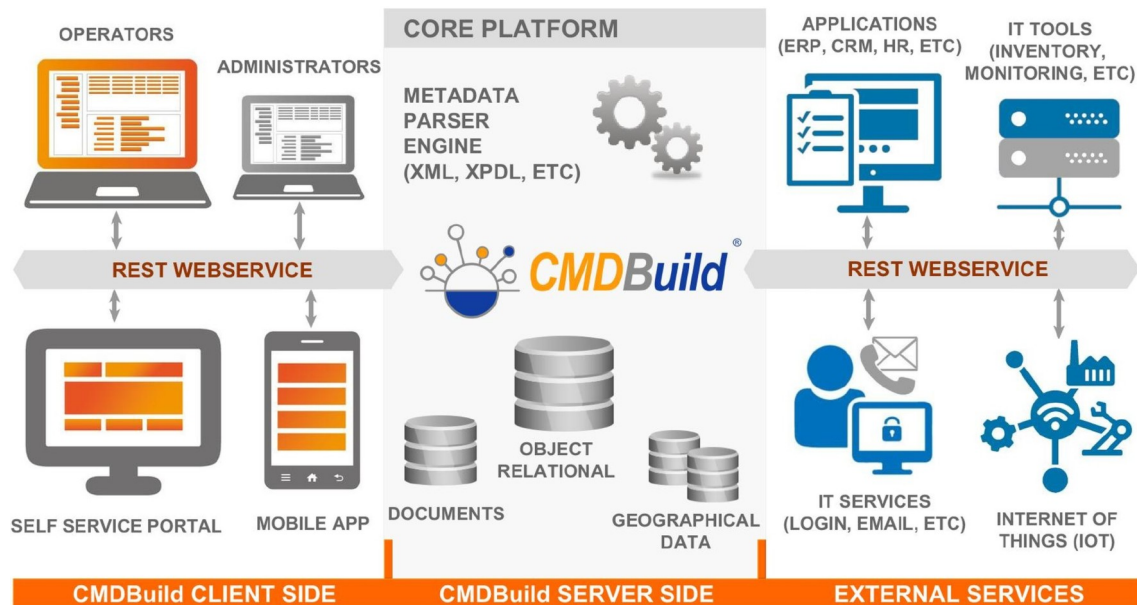
3.3. The CMDBuild conceptual schema

The following schema helps you to better understand the CMDBuild philosophy.

Please note:

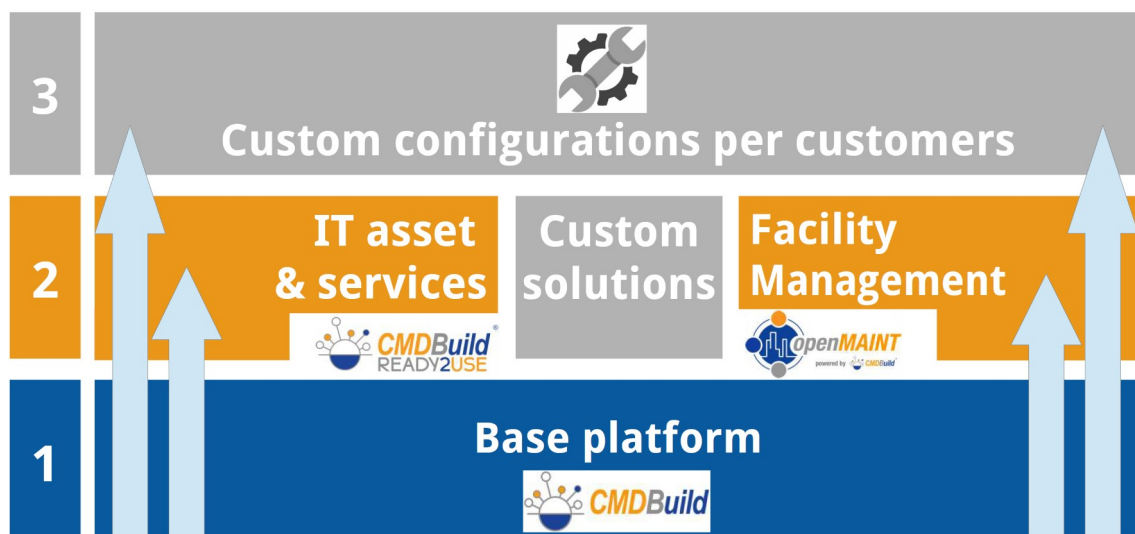
- the CMDBuild "engine" in the middle
- the main relational database (with specialization object oriented mechanisms for the creation of classes inherited from templates) based on PostgreSQL
- the document archive, connected through Alfresco REST APIs
- the geographical archive for the asset georeference, based on PostGIS, GeoServer components

- the REST webservice allows the communication between CMDBuild front-end and back-end, and between CMDBuild and external systems
- the available interfaces: web desktop for operators, web desktop for the administrator, simplified web pages for non technicians, mobile APP to work in remote way



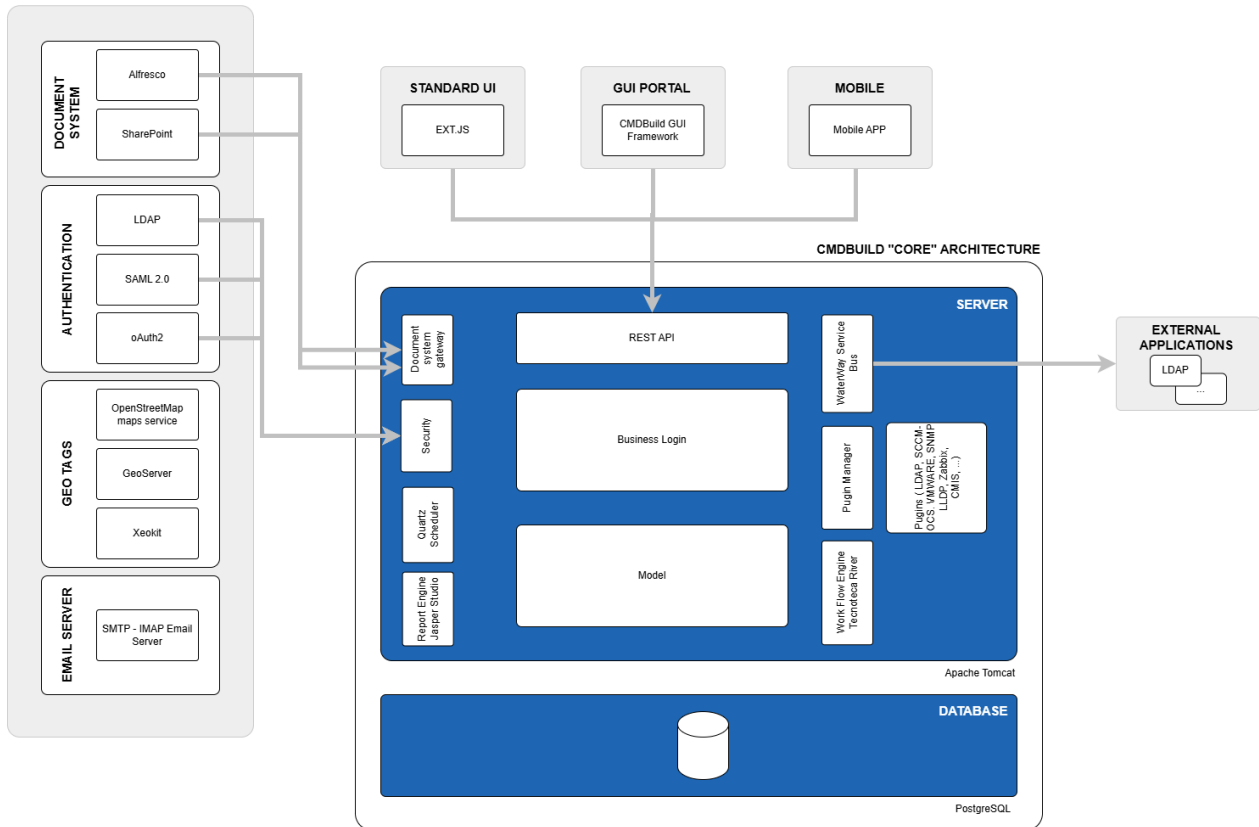
The system can be represented with a three-level structure as shown in the figure below:

- level 1: CMDBuild basic platform
- level 2: preconfigured vertical application solutions (CMDBuild READY2USE for IT and openMAINT for Facility) or custom solutions in order to cover various management areas
- level 3: custom configurations for every customer



3.4. The CMDBuild architecture

The following schema helps you to better understand the CMDBuild architecture, in terms of components and interaction ways.



Please note:

- above, the available clients
- in the middle, in blue, the components of the central system ("core") in two levels: business logic and database
- on the left, some external services used by CMDBuild in order to carry on features which are covered by available solutions (suggested by Tecnoteca or available in the prior infrastructure)
- on the right, external applications

3.5. The CMDBuild Service Bus for the interoperability

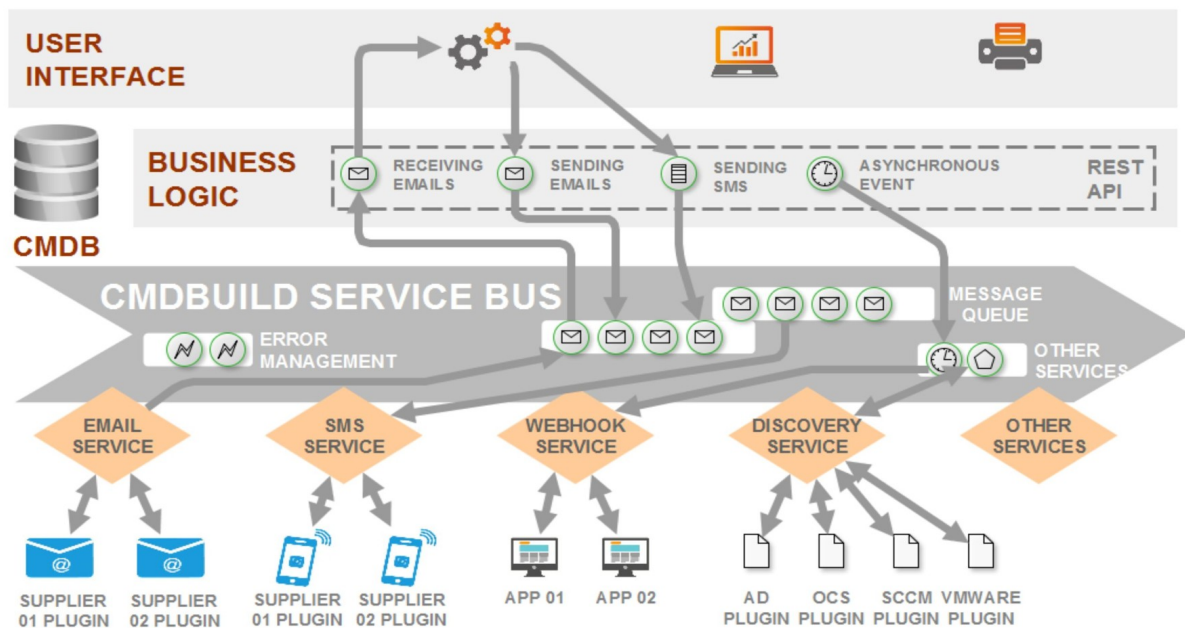
CMDBuild is a web enterprise system, implemented in SOA architecture (Service Oriented Architecture) and designed to grant interoperability solutions.

In the most efficient implementations, an architecture like this is supported by a BUS Service (or ESB, Enterprise Service Bus), a software infrastructure (middleware) for the interoperability among applications and systems through standard services for coordination, security, messaging, routing and transformations.

The CMDBuild Service Bus includes:

- management of data flows and messages (receipt, routing, elaboration, sending)
- distributed management of the charge related to some batch processes
- input/output modules with features of coordination among data sending / receiving / sync processes
- error and notification management
- “webhook” features for the subscription of a notification service in case of certain events in the CMDB
- supporting the “chat” service for the communication among CMDBuild users

The CMDBuild Service BUS is designed with a plugin architecture, where standard basic components available for everybody, standard advanced components supplied with the subscription and custom components can coexist.



3.6. The configuration mechanisms and the base features of CMDBuild

Different configuration mechanisms and basic functionalities are available in CMDBuild to obtain the best application based on each specific context of use.

They are divided in:

- main configuration mechanisms :
 - data model design
 - workflow engine and editor
 - report engine and editor
 - dashboard configuration
 - custom GUI logic configuration
- native base features :

- "relationship" paradigm
- document archive
- data historicization
- user profiling and security
- email management (input and output)
- visual graph for impact analysis
- GIS and BIM georeferences
- scheduler for automatic tasks
- import/export from GUI and batch
- scheduling management
- service bus
- plugin manager
- usage options :
 - different authentication protocols
 - multitenant
 - cluster configuration
 - preconfigured appliance
 - SaaS service

Both the main configuration mechanisms and the native base features of the platform are described in more detail in the following three chapters.

3.7. CMDBuild elements

CMDBuild uses only high tech open source components, working in the solid and mature Java enterprise environment:

CMDBuild is an enterprise system based on open standards:

- Service Oriented Architecture, organized in elements and services that cooperate with external applications through webservice
- Ajax user interface (ExtJS libraries) that grants an intuitive use of the application, interaction with ergonomics, fast response from the system
- server components developed with Java Enterprise environment, which is solid, scalable, widely used for the development of web enterprise applications
- PostgreSQL database: the most advanced, solid, safe and complete open source database

3.8. High availability configuration

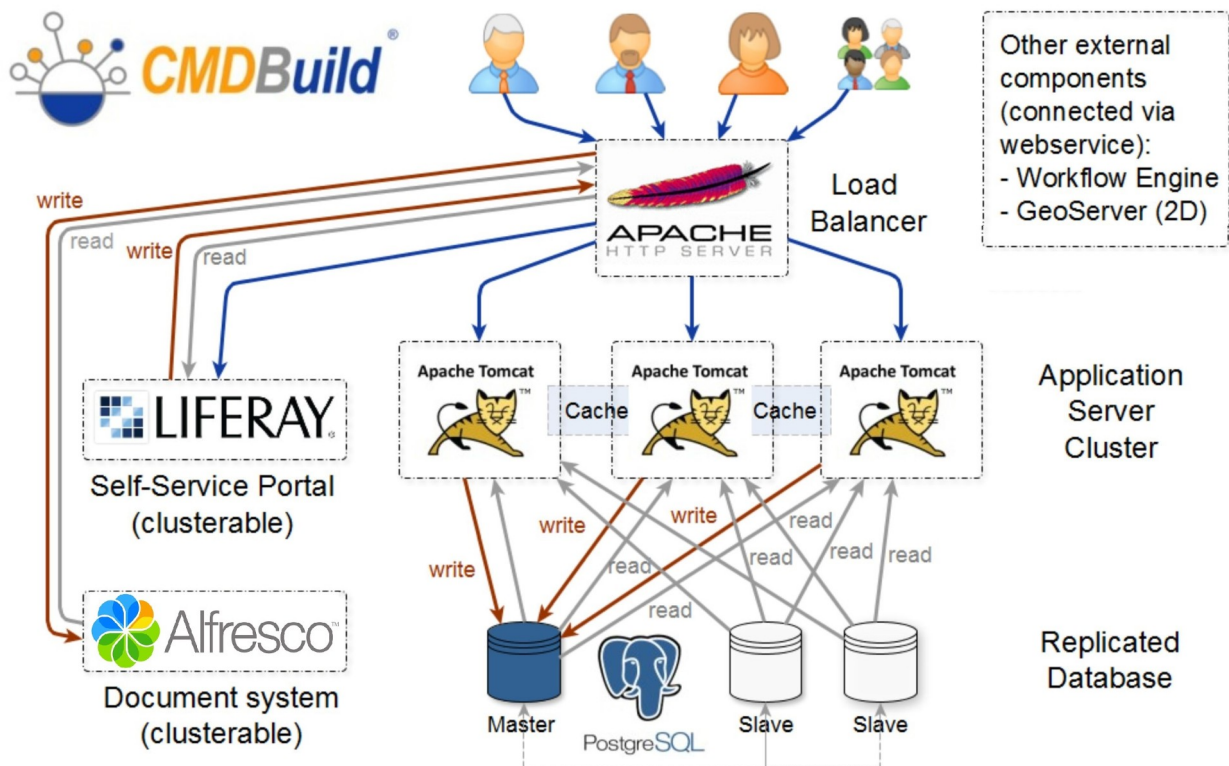
Based on a service-oriented architecture, CMDBuild is able to share its own services and components on separate servers: "core" application, workflow engine, filing module, Self Service portal, data redundancy, GeoServer, etc.

CMDBuild implements the idea of clustering and at the same time keeps several instances working on different Tomcat systems located on different servers.

This guarantees the global functioning of the system also in case a Tomcat instance breaks down (business continuity).

On the other side it allows to distribute the work load on various servers in case of several concurrent accesses, making the system "scalable", without limitations - except for the number of the available server resources

The following schema describes a cluster configuration.



4. CMDBuild features: main mechanisms

4.1. Configurability of the data model

CMDBuild has a sophisticated mechanism that allows users to configure or edit the data model of their application instance.

Thanks to CMDBuild, you are not only able to define some additional attributes, but you will be able to create an entire custom database from scratch, in terms of:

- "classes", i.e. database tables
- class "attributes", i.e. table columns (for every type, included lists with closed values, single or multiple, foreign keys, files, formulas) and related layout (talking about rows and columns)
- "domains", i.e. relation types, even with any specific additional attributes
- "lookup", i.e. lists with closed values that can be associated to a class attribute
- "inheritance", i.e. the possibility of specializing a class in "subclasses" (even with more levels), by adding specific attributes and domains and sharing the superclass ones
- "views" based on filters or SQL queries
- user interface behaviours that are specific for the class: data validation, automatisms at data update, contextual menus, online help, etc.



The entities, modelled through the Administration Module, can be provided through custom menus according to the role.

The complete configurability of the data model allows you to:

- adopt a policy to gradually create the CMDB, developing its data structure through following steps and reducing the risks a complex project can involve
- solve autonomously the need of managing new kind of items, without recurring to expensive software changes.
- choose a detail level proportional to the organization needs, human resources, financial, information and technology, without weighing down the first schema

4.2. Editor and workflow engine

One important added value of CMDBuild is the possibility of defining processes for operators to execute the management activities and grant at the same time:

- a certainty about the correct updating of the CMDB, with which the workflow system is completely integrated and correlated
- a standard interface for users
- a complete tracking of the activities carried out, used to facilitate the control and get periodic statistics on the service (SLA check)



As the data model, also the workflow definition system can be configured both in terms of information and management and automatisms flow chart.

So CMDBuild doesn't supply standard processes implemented in a static way, but a generic configuration system used according to the specific needs of every single organization.

In particular every workflow is described in the system in terms of:

- persistent data structures (specific classes and related attributes and domains)
- roles authorized to perform each interactive step of the workflow
- activities sequence (interactive steps and automatisms), with conditional flows
- information and widgets to be displayed or to be filled by user on each step
- tasks performed when the flow advances (start sub-processes, CMDB update, send e-mail, report generation, interaction with external application, etc.)

Workflows are designed using the open source external visual editor (Together Workflow Editor), and then imported into CMDBuild (standard XPD files) in order to be executed with the Together Workflow Server engine.

In the IT environment, these basic mechanisms allow the configuration of all processes provided by ITIL "best practices", included Incident Management, Change Management, Request Fulfillment, Service Catalog, etc. In the Facility Management environment, all maintenance processes can be configured.

4.3. Editor and report engine

An efficient management requires various reports in order to consult and control data registered in the system: summary, analytic, statistical with various charts, etc.

Simple table reports can be created by the user interface of CMDBuild by printing class and view content in PDF or CSV.

In order to design more sophisticated reports you would use the external visual editor JasperStudio from the open source suite JasperReports, which allows you to manage intuitively layout, formatting, grouping, sorting, running total, subreport, images, chart, etc.

As with workflows, also for reports the visual editor produces an XML descriptor, which is imported to CMDBuild and associated to the related report.

The internal engine JasperReports will interpret the XML file and execute the related report.

In this way you can configure reports for the production of tables and lists, letters and documents, statistics with various charts, labels with Barcode / QRcode, etc.

Also in this case the solution adopted by CMDBuild requires a complete separation between "core" code and metadata, allowing all users to extend and customize their report system.

Through proper training courses, the System Administrator can acquire the necessary skills for the implementation of custom reports and solve quickly new problems notified by the company business.



4.4. Dashboard configuration

It is important for the company management to have the possibility of consulting the trend of main indicators (KPI - Key Performance Indicator), which are achieved by analysing the information (data cards and processes) filed in the CMDB.



For this purpose you are able to define "dashboard" pages, each dedicated to different aspects (assets situation, service desk performance, income statement, etc.), and each containing one or more representative charts of the various interesting parameters.

The dashboard configuration requires you to:

- create the page of the new dashboard from the Administration Module
- insert one or more charts in the page through:
 - define the data source in the shape of database function
 - define the chart features (pie, bar, line, gauge) through a proper wizard of the Administration Module
 - define the layout (talking about rows and columns) by moving each chart (drag and drop) into the desired position inside the dashboard
- insert into the page some text-elements or data grids coming from functions, classes or views, with the same range of layouts above mentioned

As for the reporting, also the dashboard configuration can be executed by the company's technicians after having acquired the knowledge about the CMDb basic structure.

4.5. Configuration of GUIs and custom business logics

CMDBuild makes available further mechanisms to get user interfaces and custom business logics.

Simultaneously with the data model design, you can configure custom behaviours when editing the related forms: validating criteria for attributes, possibility of hiding or making not editable an attribute according to the inserted data, possibility of enhancing an attribute according to the value of other attributes.



Thanks to the features of the framework Sencha ExtJs (used also for the implementation of the CMDBuild GUI), you can realize in javascript custom web pages you can open from the application menu and subject to the standard security mechanisms.

It is possible to program in javascript custom widgets, which operators can use in the form of popup windows during the update of cards and processes.

It is possible to configure from the Administration Module contextual menus, which will quicken the execution of the commands useful in the current context, will be available during the visualization and navigation among data, will have their own user interface realized with the same modalities of the custom pages and will be implemented through javascript code.

And, finally, it is possible to configure from the Administration Module triggers that will be processed at the occurring of the different events associated with the data management forms (visualization, pre-charging, post-confirmation, etc.) and will be written through javascript code.

5. CMDBuild features: native features

5.1. Relationship paradigm

A necessary condition for a CMDB (Configuration Management Database) is the possibility of describing all managed Configuration Item (CI) and all related connections.

The connections among CIs are as important as the describing information of the CIs, since they allow to know the mutual dependencies and analyze the possible effects in case of changes, breakdowns, etc.

When modelling the database, CMDBuild allows you to create all necessary domains (types of relations), for both cardinality 1:N, 1:1 and N:M.

You can also configure specific attributes of a domain (e.g. the role for which an operator is responsible for a CI, the color of a cable that connects two CIs, etc.).

When using CMDBuild, you will be naturally led to register the relations among CIs and consult them and navigate intuitively.

The interface of Management Module allows the operators to:

- open the list of the already defined relations
- create new relations, explicitly by selecting the domain and the CIs to be linked, or implicitly optimizing a reference attribute (foreign key in CMDBuild)
- edit or delete a relation
- navigate on the connected card, as if following a link
- open the relations in a graphic way (see following paragraph)



5.2. Visual graph

The Relation Chart is a tool for facilitating the analysis of the relations among cards, allowing you to:

- examine them in a visual way, recognizing the items through the class icons and with the help of the numerous available features
- assess intuitively and consciously the effects among cards in order to:
 - decide in advance if and how you want to intervene
 - identify the reason why something got wrong after the intervention



The viewer works in 3D in order to grant an optimal result also when displaying complex charts.

The Relation Graph allows you to:

- move in the graph (pan, zoom, rotation, refresh)
- select a chart and navigate in 3D following its relations
- apply filters to analyze homogeneous relation subsets
- have available the complete data of cards and of the analyzed relations

5.3. Documents archive

The documents are an important informative component related to each card (Configuration Item) filed in the CMDB: pictures (locations, items, plants, etc.), videos, technical documentation (manuals, brochures, certifications, etc.), administrative documentation (bills, invoices, contracts, etc.), set of forms, error screenshots, etc.

CMDBuild allows operators to upload and attach files to every card with no number or type limit.

The system admin can configure, for each class, diversified lists of allowed document categories (folders of documents), by setting up, for each category, different numbers and types of metadata (attributes) that can describe each document.

From every card you can also open the list of attached documents and download the desired ones or open them in the browser.

CMDBuild suggests as embedded document repository the DMS Alfresco system, which is market leader among the open source solutions.



5.4. GIS and BIM georeferences

Keeping update the location (georeference) of items is important especially during the Asset Management activities.

CMDBuild includes three possible solutions:

- georeference of items on maps (e.g. buildings, data lines or assets located outside the buildings, like street cabinets, traffic lights, cameras in restricted traffic areas, etc.)
- items georeference (plants, technical items, furniture, fire extinguishers, etc.) on 2D planimetries of the floors of each building
- items georeference within 3D models



In these cases the features to view, query and change the item position are available from web browsers, without installing any plug-in.

In the first case, CMDBuild uses external map services, among which OpenStreetMap.

In the second case, you can upload vector maps into the system (which are usually produced by the program AutoCAD) used as background files, while the item position is archived in the same database PostgreSQL, by using the spatial extension PostGIS.

In the third case, CMDBuild supports the BIM paradigm (Building Information Modeling), which is a conceptual solution with the aim of managing the building design procedure, from the conception of the building to its construction, use and maintenance, and finally to its demolition, if any.

CMDBuild is able to sync automatically the chosen information from a BIM repository by using the open standard format IFC (Industry Foundation Classes) and display in a realistic way interiors and assets thanks to a 3D interactive viewer, integrated in the user interface.

5.5. Data history

In order to assure the complete traceability of operations, CMDBuild historicizes the edited cards and relations.



This allows you to control the complete situation of a portion of the CMDB in a certain date (for example for analyzing the contest in which a breakdown occurred), date back to the operator who did an irregular edit on data, calculate an inventory in a certain previous data, etc.

The filing of each "historic" card includes all attributes of this card, its initial and final dates and the username of the operator who did the change.

Operators can also consult the list of all changes done on a data card, where for each version the edited attributes are highlighted.

You can also create views and reports that open or print a situation (inventory, list of open tickets, etc.) at any previous date.

A visual browse mode will be implemented: you will be able to consult the historic data of the CMDB (in read-only mode), by selecting the chosen date through a specific time slider.

From a technical point of view, this process is implemented through historic derived tables (thanks to the native mechanism of the database PostgreSQL) from the main table and fed over time through triggers at each change of a card or of a relation.

5.6. Scheduler for automatic tasks

CMDBuild allows you to configure automatic, synchronous and asynchronous operations executed in background by a scheduler integrated in the system.



In particular, the scheduler allows you to:

- control the e-mail inbox, register the e-mails in the system together with any attachments, notify their incoming and start a workflow
- send an e-mail with default deadline, with a possible attached report
- start an operation for exporting or importing CSV or Excel (XLS, XLSX) files according to the configuration in the related import / export templates
- start an operation for import and sync of data from relational database according to the configuration in the related import / export templates
- start an operation that imports plans (DWG files) or BIM models (IFC files) according to the configuration of the related import / export templates
- start a workflow, by specifying its type and parameters

For all above mentioned operations you can find a visual configuration wizard in the Administration Module..

The control panel of the Scheduler (Task Manager), available in the Administration Module, shows the configured tasks with their related status and allows their management (start, suspension, removal, configuration).

The feature is based on the use of the open source tool Quartz Job Scheduler, integrated in CMDBuild.

5.7. User profiling and Multitenant configuration

It is important for an application of an enterprise company to have a solid and flexible data protection system.

CMDBuild manages access permissions in a consistent and granular way:



- in a consistent way because it is independent from the request channel, be it the desktop web interface or the mobile APP or other applications based on the REST webservices (Self-Service portal, connectors, etc).
- in a granular way, because the permissions can be defined on classes (also restricted to rows or columns), processes, views, search filters, reports, dashboards

Permissions are assigned to user groups (roles) and every user can belong to one or more roles.

The user's login can be done through:

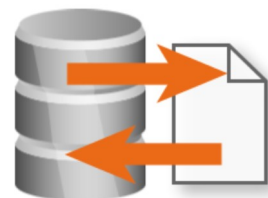
- local control on the CMDBuild database (with possible password control on the LDAP)
- Header Authentication
- SSO (Single Sign On) authentication on LDAP through the open source tool CAS or on SAML 2.0 or on ADFS 4.0

In case of multiple companies' use, or use among Departments or independent seats within a company, it is possible to configure CMDBuild in Multitenant mode, allowing each Tenant to work on an isolated CMDB subdivision. You can configure both a complete division (completely separate databases) and partial divisions (some information in common with all users and other separated). The list of usable Tenants can be defined from an applicable class of CMDBuild (seats, companies, customers, etc.) or from a database custom function, where you can implement complex visibility rules.

5.8. Data Import / Export through files

In addition to the possibility of automatically reading / writing data in the CMDB through webservice, there is also a sophisticated import / export system through data source in the most common formats:

- "CSV", "XLS" and "XLSX" files (which can also be exported)
- relational databases
- DWG files (2D planimetries)
- IFC files (3D BIM models)



In case of importing, you can work in "merge" mode, by editing the already available strings (recognized through a unique key), inserting the new ones and deleting the missing ones (logic cancellation or change of the application status).

The import / export operation can be done both in interactive mode from user interface (through the contextual menu) and in batch mode (through the above described scheduler).

There is a complete error handling, shown in the user interface if working in interactive mode and sent per e-mail. Once the errors have been corrected, the file can be recharged without having to change it, since the already imported strings will be ignored.

In order to manage the quite complex operations in a simplified way, the system is based on the configuration of specific "templates". Each template will provide diversified access permissions and will include all information useful for the operation automation: operation type, attributes mapping (even multiple, for the management of the sync of more classes), sync mode (unique key and deleting mode), etc.

5.9. Email management

CMDBuild uses the e-mails both as inner tool for notifications to operators and as communication tool and as external event, in view of which you can generate automatic operations.

Notifications can be generated:

- manually from the workflow engine through the interactive management widget or automatically through API
- automatically from the scheduler after receiving e-mails on specific accounts (notification to the operator in question) or as output of import / export operations
- from custom code (external programs, scripts, trigger, etc.) through API



The CMDBuild user interface allows you to:

- consult through a specific TAB from the user interface the e-mails produced within CMDBuild or coming from external accounts, in both cases registered in the system in the form of information related to cards or processes
- send new e-mails from the same TAB in free form or using accounts and templates preconfigured by the system administrator

In the end the scheduler allows you to start processes when receiving e-mails on pre-defined IMAP accounts (this feature is useful to open a maintenance process when receiving an alarm or a signal).

5.10. Schedule

This function helps operators to remind and manage single or recurring deadlines related, for example, to certifications, warranties, contracts with customers and suppliers, administrative or staff-related procedures, etc.

Deadlines can be generated automatically from the system when inserting new data cards containing data attributes marked as "deadline", but also can be inserted manually by operators directly into the Scheduling file.



In the first case the System Administrator can define generation criteria (start, end, number of occurrences, frequency type and value, notification mode, etc.) and link them to any Data-type attribute of any class. The user can view and/or edit these criteria when inserting a data card of that class. The resulting deadlines will be then registered into the Scheduling file.

The tasks that are due are notified via e-mail, with the required notice, to the specified user or group.

Having the necessary permissions, in the Scheduling file you will be able to:

- view the deadlines list, with the standard search, filter and print function
- view / edit the complete data card of a deadline, with the information related to category, date, description, interested group / user, status, etc.
- close or cancel a deadline intervening on the status and on the date of event
- access the "Calendar" view and view the deadlines in the type of interface
- view the received notifications

6. CMDBuild features: user interfaces

6.1. Web desktop GUI for operators

The interface for operators, called Data Management Module, allows you to process all daily Asset Management operations.

According to each role, this means:

- consult and update data of assets (items and hierarchy depending on the applicable domain) and of related entities (company, employees, buildings, customers, suppliers, orders, etc.)
- execute your activities according to the company processes helped by collaborative workflows specifically projected
- execute the control activities by opening dashboards and printing reports specifically configured



The main features of the data management module include:

- card management and "views" on data cards:
 - searches through set criteria or through filters previously saved
 - inserting, creation, change, cloning of cards
 - master view - detail
 - consultation, creation and change of relations among cards
 - open relation graph
 - consultation of the changes of cards
 - card attachments upload, by compiling the metadata
 - execution of commands in the contextual menu, that include the standard ones for the massive modification and data import / export (in case of configured templates)
 - consultation of assets on cartography, vector maps, 3D models
 - exportation and printing of cards, choosing columns and row filter
 - printing of current card with relations
- process management:
 - consultation and research
 - start and advance using the provided widgets
- reports and dashboards:
 - report printing (tabular, with graphs, documents, etc.)
 - consultation of graphs configured in the system and control of related KPI
- other features:
 - change of group / change of tenant
 - user preferences
 - password change
 - notifications
 - schedule

6.2. Web desktop GUI desktop web for the administrator

The administrator's interface, called Administration Module, allows to process all configuration operations of the system.

The main features in Administration Module include:

- summary dashboard for the CMDDB functioning and contents
- configuration of the CMDDB data model and of certain behaviours of the related user interface:
 - creation of new classes, i.e. new object typologies
 - creation and change of (common and geographical) attributes in a class and possible definition of presentation layout
 - definition of behaviours on the user interface: categories of attachments, data organisation, attributes grouping in fieldsets, form trigger for the implementation of business logics, contextual menus, widgets
- workflow configuration related to the collaborative CMDBuild processes:
 - definition of persistence information in workflows (classes of a particular type)
 - import of the flow externally designed with TWE tool (XPDL file)
 - definition of the behaviours on the user interface, which are similar to the class ones
- creation of domains, i.e. types of relations and any related attributes among couples of classes
- creation of tablelists (lookup) to manage attributes with closed values
- definition of filters and views:
 - creation of predefined filters used during the data reference on the data cards
 - creation of views, both through filters on simple and multitable data and SQL queries
- definition of navigation trees (used in the GIS and in the Relation Graph)
- creation and change of dashboards, configured from user interface
- report creation and change, by importing the XML layout designed externally with the JasperStudio tool
- definition of custom menus for different users' groups
- configuration of custom elements for the user interface: custom pages and components
- definition of the way to manage attachments: document categories and document models (groups of metadata)
- safety management:
 - creation of users and groups of users (roles)
 - definition of permissions on classes and processes (also restricted to subsets of rows and columns) and on other items managed in the system (views, search filters, reports, dashboards)
- e-mail management (accounts and templates)
- template configuration for import / export (CSV / XLS, XLSX, DB tables, DWG, IFC)
- configuration of the Task Manager and of the operations to execute in the background
- configuration of Bus Service features and components (WaterWAY)
- configuration of the timetable feeding modes
- configurations of GIS and BIM
- localization into several languages
- system management
- definition of the system parameters



6.3. mobile APP for operators

A lot of operations executed during the management of the asset lifecycle occurs in the field (movement of items, problem solving, inventory, etc.). It is useful to keep track of them as they are performed, in order to avoid delays and errors.

The mobile APP of CMDBuild has been realized with the aim of executing the main features of the Data Management Module through tablets and smartphones.



The mobile APP allows you to:

- use the navigation menu
- consult and update cards, relations and attachments
- do searches and apply filters
- consult and process workflow with the main widgets
- print reports
- set the configuration parameters and consult logs

You can also apply additional features that are not available in the desktop interface:

- taking pictures and filing them in the CMDB as attachments
- automatically reading Barcodes and Qrcodes when opening the related card

The mobile CMDBuild can be used with Android and iOS devices.

In order to get the APP you should subscribe the maintenance service.

6.4. Self-Service portal for final users

The standard CMDBuild user interface is addressed to technicians who use it regularly.

It is often necessary to make available CMDBuild features for non-technical users who need to process some basic operations on simplified interfaces hosted by intranet portals.



The typical operations these users are interested in can include the signal of requests, the control of the solving iter, the consult of FAQs, the print of reports, etc.

CMDBuild provides a component called GUI Framework for the creation of simplified interfaces aimed to non-technical users.

The CMDBuild GUI Framework has the following features:

- it can be activated in portals based on different technologies (it is developed in javascript / JQuery environment)
- it allows an (almost) unlimited freedom when projecting the graphic layout, defined through an XML descriptor and with the possibility of intervening on the CSS
- it grants a quick configuration thanks to predefined functions for the interaction with CMDBuild and thanks to native graphic solutions (forms, grids, upload buttons and other widgets)
- it interacts with CMDBuild core through the REST webservice

The CMDBuild GUI Framework has been used for the verticalization of CMDBuild READY2USE and openMAINT for the implementation of Self Service portals used for the interaction among final users and IT operators.

6.5. REST Webservices

Besides user interfaces (GUI), CMDBuild provides REST webservices, i.e. programmatic interfaces that allow to interact with the system (read and write data, execute operations, etc.) through the programming code.

The interfaces themselves are used also within CMDBuild for the interaction among its various components.

When used externally, they allow to implement interoperability solutions with external applications and systems interested in exchanging data and notifications with CMDBuild.

Some possible scenarios:

- automatically accepting in CMDBuild any updates of data whose main manager (master) is an external specialized system, for example
 - an application for the asset management
 - an application for the staff's management
 - an ERP system for receiving purchase orders of new assets
 - tools for managing infrastructure (control systems, inventory systems, etc.)
- receiving automatic notifications of malfunctions from monitoring systems
- interacting with IOT systems (Internet of Things) like sensors, data collection centers, etc.

Webservices are used from the following external CMDBuild components:

- the CMDBuild GUI Framework, which is useful for creating simplified interfaces for non-technical users
- the mobile APP, which can be used to work with CMDBuild by smartphones and tablets (available if subscribing the maintenance service)

Every CMDBuild user can also use the same webservices to implement custom interoperability solutions.



7. The IT Governance verticalization: CMDBuild READY2USE

7.1. The product

CMDBuild REASY2USE is an application for the IT Governance management, ready to be used in the production environment and resulting as configuration of the CMDBuild basic environment.



CMDBuild READY2USE was born as a result of the experience of numerous installations realized all over the world; it follows the ITIL best practice and implements every functionality necessary for medium and large organizations:

- it manages the configuration of hardware and software items, client and infrastructure area items, physical servers and virtual ones, services and dependencies
- it includes a self-service portal to access the service catalogue, report malfunctions and send new requests
- it implements the Incident Management, Request Fulfillment, Change Management, Problem Management, Asset Management workflows
- it includes sync connectors with Active Directory (staff), OCS Inventory (automatic client discovery), VMware VCenter (virtual host data) and SNMP / LLDP (network topology) and it interfaces with the monitoring tool Zabbix
- it includes already configured reports and dashboards
- it uses the mobile APP

As you can see, CMDBuild READY2USE is not only a CMDB application, but it completely covers also the aspects regarding the management processes of the IT service (Service Desk).

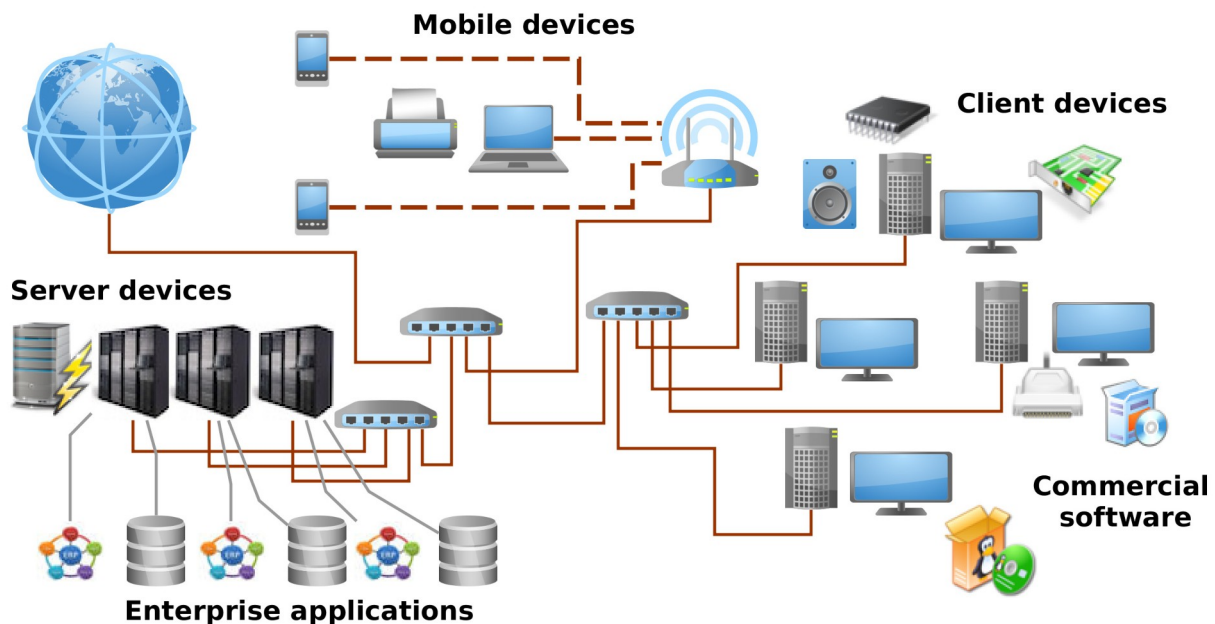
CMDBuild READY2USE keeps the possibility of reusing CMDBuild base mechanisms to customize or expand the suggested standard configurations.

7.2. Data model

The data model includes the following macro-areas:

- Organization: Group Companies, Employees, Organization Units
- Locations: Buildings, Floors, Rooms
- Suppliers: Data, Contact Personnel, Purchase Orders
- Customers: Data, Contact Personnel
- Services: Business Service, Technical Service, Service Contract, KPI
- Other Configuration Items
 - Client: Desktop, Notebook, Mobile, Printer, Monitor
 - Infrastructure: Physical Server, Virtual Server, Cluster, Storage, Network Equipment
 - Telephone: Telephone Exchange, Telephone equipment, SIM

- Software: Software Catalogue, Client Software, Infrastructure Software, Database, Application, Component, License
- Networks, Network interfaces and IP Addresses



7.3. The processes and the Self-Service portal

ITIL Best Practices, generally adopted as support for the IT service management by mature organizations, use a process-oriented approach.

With the basic CMDBuild mechanisms and, in particular, using the workflow editor and engine, the following most used processes have been implemented in the CMDBuild READY2USE solution in "ITIL compliant" mode:

- Incident Management, for the management of malfunctions
- Request Fulfillment, for the request of information and services
- Change Management, for the control and tracking of any changes to the infrastructure
- Problem Management, to solve the problems highlighted by Incident
- Asset Management, for the management of the asset lifecycle

For the support of Service Desk processes a Self Service Portal has been implemented. It provides non-IT users with a simplified interface to interact with the Service Desk staff and to process the following main activities:

- signal of malfunctions by starting the Incident Management workflow
- consult of the IT Service Catalogue, send a new request or ask for information, by starting the Request Fulfillment workflow
- control of the advancement of your own requests
- consult of FAQs in order to limit the reports
- execution of reports, if enabled

- consultation of the profile information and signal of any corrections
- consultation of the asset information and signal of any corrections
- consultation of the news posted by the IT Staff and visualization of useful links

The Self Service Portal is supplied with non-open source license to the users that subscribe the annual Subscription with Tecnoteca.

7.4. Connectors with Automatic Discovery and Monitoring systems

The management of IT Services by large and medium-sized institutions and companies is necessarily carried out through more specialized information systems, which should cooperate within the management of their activities.

In particular, when the information system is the data owner (master), which is the one designed to manage a certain type of data and when they are repeated in CMDBuild, it is important to have automatic tools to verify the alignment of these information updates and eventually gather any changes.

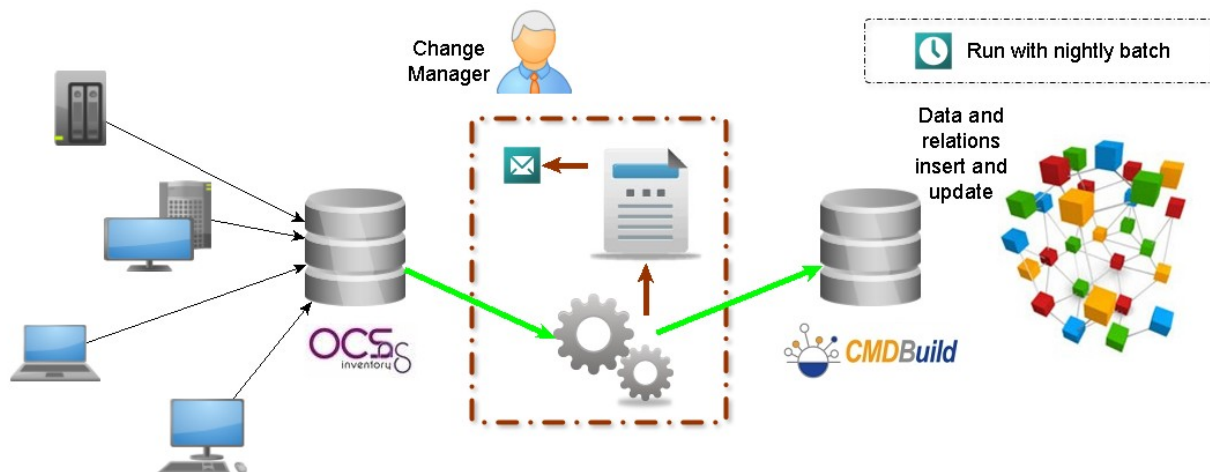
CMDBuild READY2USE includes the sync connectors with:

- Active Directory, for the control of data related to the employees file
- OCS Inventory or Microsoft SCCM, for the control of data related to the used physical computers
- VMware VCenter, for the control of data related to the used virtual servers
- Amazon AWS, for the control of data related to the used virtual servers
- SNMP / LLDP connectors, to control data related to the network structure

All connectors share some common features:

- they are based on the waterWAY framework
- they are executed every night according to the batch modality
- they require that the external system can be contacted by CMDBuild through a direct access to the database or through webservice or API
- they execute the sync of non-updated information in the CMDB
- they produce a summary reporting the changes and then they send it per email to a technician

There is also another connector with the monitoring open source tool Zabbix which gathers and registers in CMDBuild the malfunctioning events, notifies them to the responsible technicians and starts an Incident Management process.



The connectors described in this paragraph are supplied with non-open source license only to the users that subscribe the annual Subscription with Tecnoteca.

7.5. The report system

With the basic CMDBuild mechanisms and, in particular, using the report editor and engine, several standard reports and dashboards have been implemented in the CMDBuild READY2USE solution.

Using the same mechanisms, users can easily create new reports customized according to their organization and needs.

Among the available reports and dashboards there are:

- Inventory of jobs (also at previous dates)
- Dashboard with the asset situation according to the availability / status / type / age
- Location list with assets
- MTBF - Average time between malfunctions
- Control report of Incident Management processes (open processes, average closing times, advancement times, tickets active at a certain date, tickets active during a date interval)
- Dashboards of Incident Management processes according to the status / category / average closing time / open-closed
- Reports of Request Fulfilment processes (open processes, average closing times, advancement times)
- Report for the quality control of the information filed in the CMDB (data infrequently updated, incompleteness of information classified as not important)
- Dashboards of Request Fulfilment processes according to the status / category / average closing time / open-closed
- Dashboards of active requests for the Service Desk
- Dashboards of Service Level Agreement analysis
- Rack schema
- Barcode print

- Reports for the control of the sync connectors (LDAP, OCS Inventory, VMware, LLDP)

7.6. The mobile App

Also for the IT Services management a lot of operations executed during the management of the asset lifecycle occurs in the field (delivery / withdrawal of computers and devices, inventory, etc.). It is useful to keep track of them as they are performed, avoiding delays and oversights.

The mobile APP made available by CMDBuild for smartphone and tablet allows to:

- consult and update the data cards of IT assets
- register the activities processed within the expected IT processes
- print reports
- send to the server pictures of the IT asset in process
- read a Barcode or a QRcode and receive the complete card of the IT asset from the server

8. The Facility Management verticalization: openMAINT

8.1. The product

openMAINT is an application dedicated to the Property & Facility Management, ready to be used in the production environment and resulting as configuration of the CMDBuild basic environment



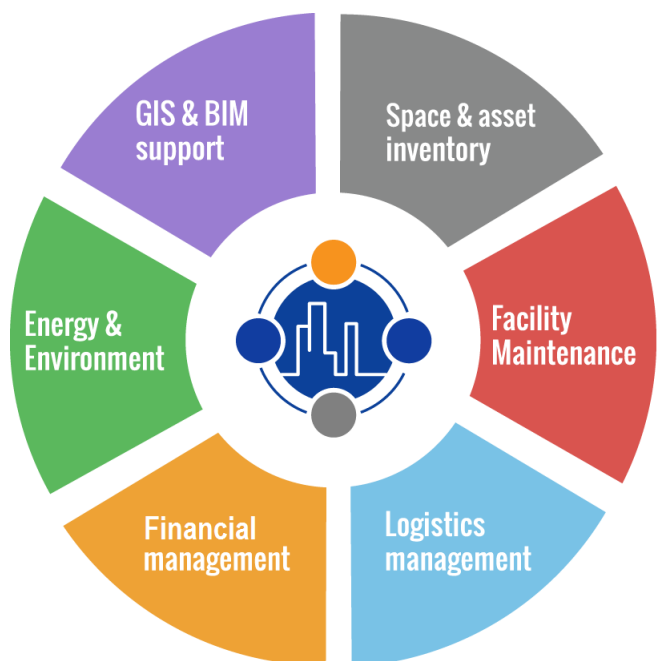
openMAINT is an application for the management of mobile assets, industrial properties and infrastructure (road networks, technological networks, etc.), plants and technical devices (electric panels, heating and cooling, light fixture, fire extinguishers, safety systems, etc.), and the related logistical, economical and maintenance activities (scheduled and breakdown ones).

It is then an application that helps you know, organize and keep update the data related to the property, supporting the managing authority's decision and operative activities.

openMAINT is a CMMS product (Computerized Maintenance Management System) derived from specific and extensive experience in the field, following the rules and best practice (nota bene for Italian users: please refer to the Clause UNI 10951 Sistemi Informativi per la Gestione della Manutenzione dei Patrimoni Immobiliari).

The functional areas that organize openMAINT include:

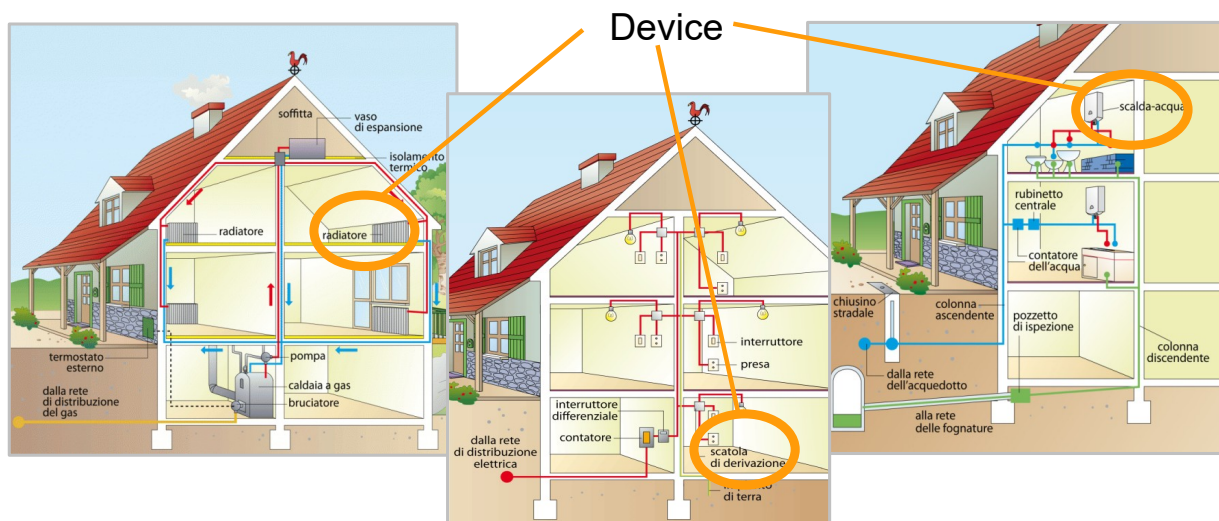
- inventory of property assets and mobile assets, dedicated to the complete knowledge of maintenance items and their technical and functional features, included the related document archive
- scheduled and breakdown maintenance management using the Maintenance Manual, a knowledge base integrated in openMAINT and customizable according to the needs of each organization
- logistic management, for the management of warehouse records and items and for the movement of materials
- financial management, for the management of budget, suppliers, maintenance costs, contracts and of the purchasing documents
- energy and environment, for the registration and analysis of consumption data
- GIS & BIM support, for the asset georeference and visualization on geographical maps, on 2D GIS planimetries and on 3D BIM models (which can be sync with the design tools through standard IFC files)



8.2. Data model

The data model includes the following macro-areas:

- Organization: group companies, employees, organization units
- Suppliers: data, contact personnel
- Customers: data, contact personnel
- Locations: buildings, floors, rooms, units, greenery and external infrastructures
- Other Configuration Items (maintenance items):
 - Structural elements
 - Systems: security, electrical, air-conditioning, sanitary, industrial production, etc.)
 - Technical items, related to the above-mentioned systems
 - Furniture
- Maintenance management: maintenance teams, preventive maintenance (maintenance manual, maintenance plan with checklist, maintenance calendar, work orders, SLA), maintenance on breakdown (receipt of notifications, resolution process, SLA)
- Economic management: lists and price lists, ledger entries, budget, contracts, purchase orders for suppliers, millesimal tables and allocations of lease costs
- Logistics: records of spare parts and consumable, warehouse movements
- Energy: counters, consumptions

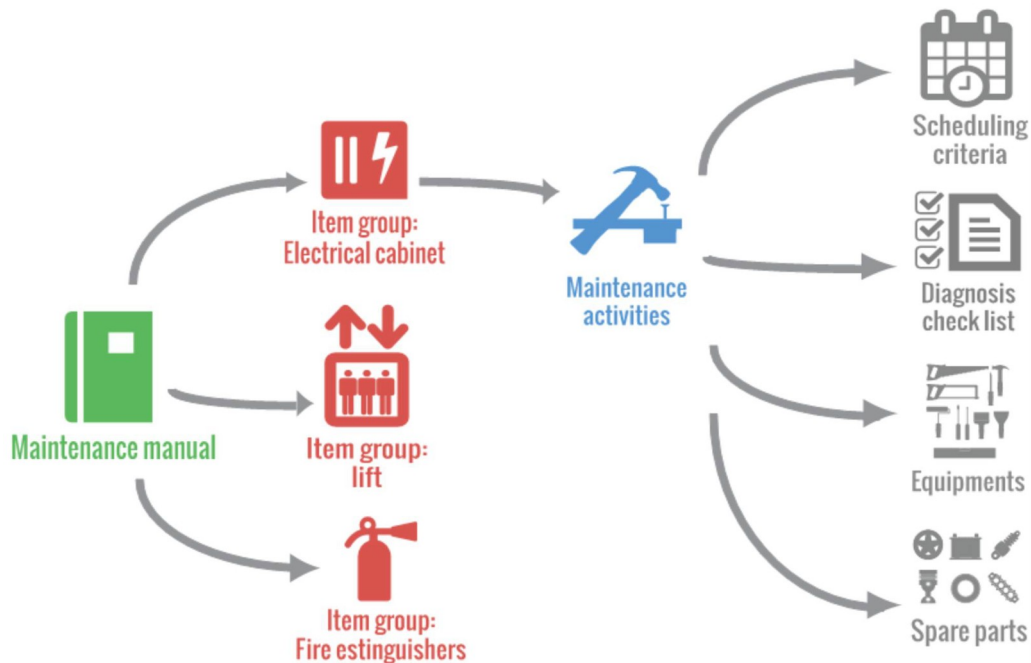


8.3. Maintenance processes and the Self-Service portal

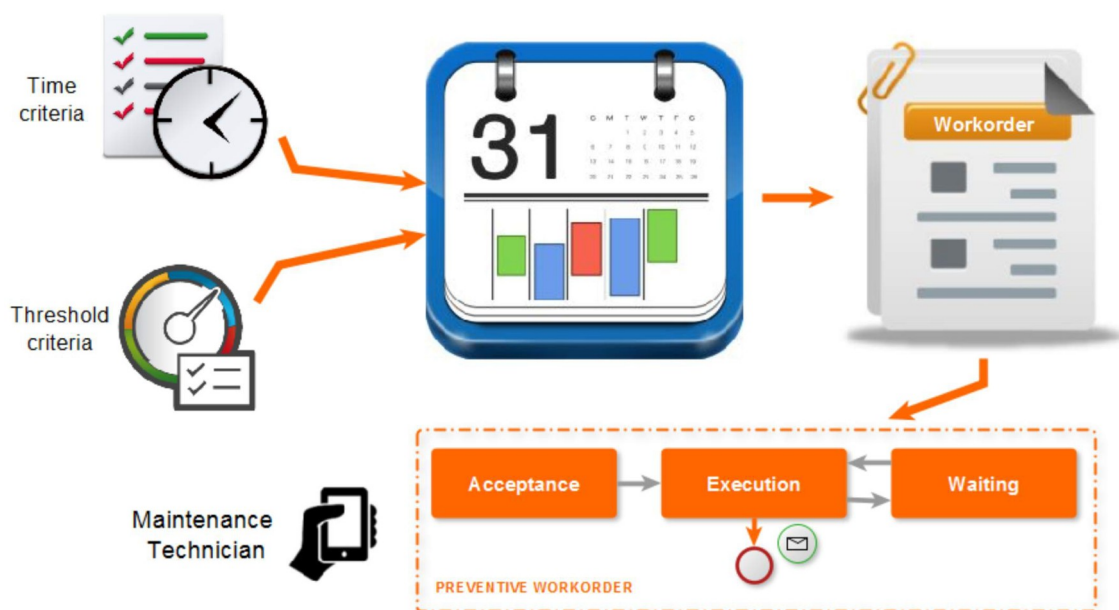
One of the main purposes of openMAINT is to lead the operators during the maintenance management.

The maintenance concerns the inventory items registered in the system and has to be done following the managing authority's organization (intervention team, external suppliers, resources, schedule, etc.) and the technical features of the item (control frequency, type of intervention, equipment, repair parts, etc.).

In order to support these needs, openMAINT will create in the system a knowledge base to follow during the maintenance activities, called Maintenance Manual that includes information on frequencies and check-lists, expert workteams, possible breakdowns, spare parts and equipment, etc.

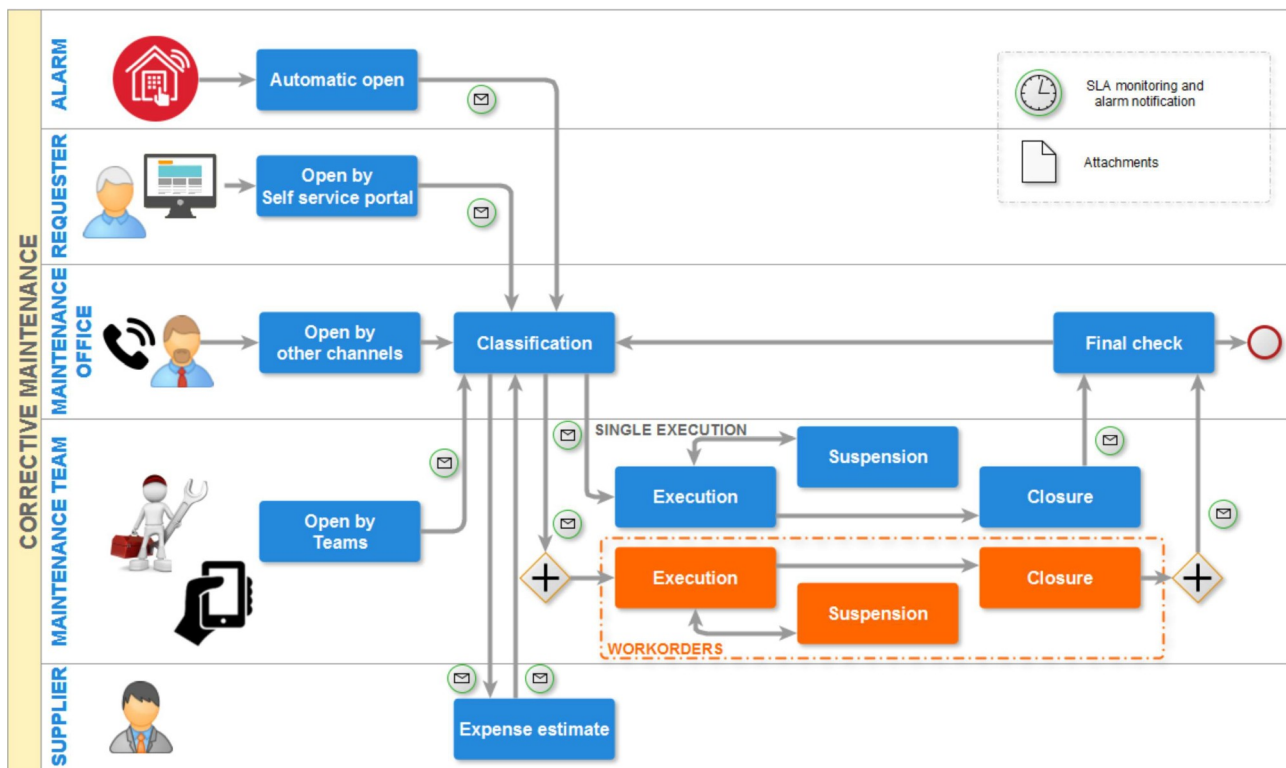


On the base of the information registered in the system within the Maintenance Manual and in particular according to the expected frequencies for the various kinds of activities, the Calendar for the Preventive Maintenance is generated automatically and also the relative Workorders.



The Breakdown Maintenance is managed through a collaborative process, made of:

- a main flow including the opening of the breakdown from alarm systems or users or operators from the Maintenance Office (if coming from other channels), its classification and approval, the possible request for estimate to one or more suppliers, the choice of supplier and the generation of one or more Workorders in order to solve it
- a secondary execution flow of Workorders from teams of internal or external maintenance technicians, with the possibility of specifying duration and materials used in order to get from the system a summary (pre-invoice) of the related costs



For the support of the Breakdown Maintenance process a Self Service Portal has been implemented. It provides non-IT users with a simplified interface to interact with the Service Managers and to process the following main activities:

- signal of malfunctions by starting the Breakdown Maintenance workflow
- control of the advancement of your own requests
- consult updated news about the distribution of technical services
- execution of reports, if enabled
- consultation of the profile information and signal of any corrections

The Self Service Portal is supplied with non-open source license to the users that subscribe the annual Subscription with Tecnoteca.

8.4. The interaction with GIS and BIM systems

The openMAINT application uses the base mechanisms of CMDBuild in order to manage the georeference of items on geographical maps (real estates, other kinds of infrastructure) and on 2D vector maps (plants, technical items, furniture, etc.).

The georeference on geographical maps uses the online open source OpenStreetMap service.

For the georeference on 2D vector maps, if some conditions are met, openMAINT is able to import files produced in a semiautomatic mode thanks to design tools like Autodesk AutoCAD. It is also able to correlate the design elements to its cards, guaranteeing the automatic navigation from the planimetry to the information card of the asset and viceversa.

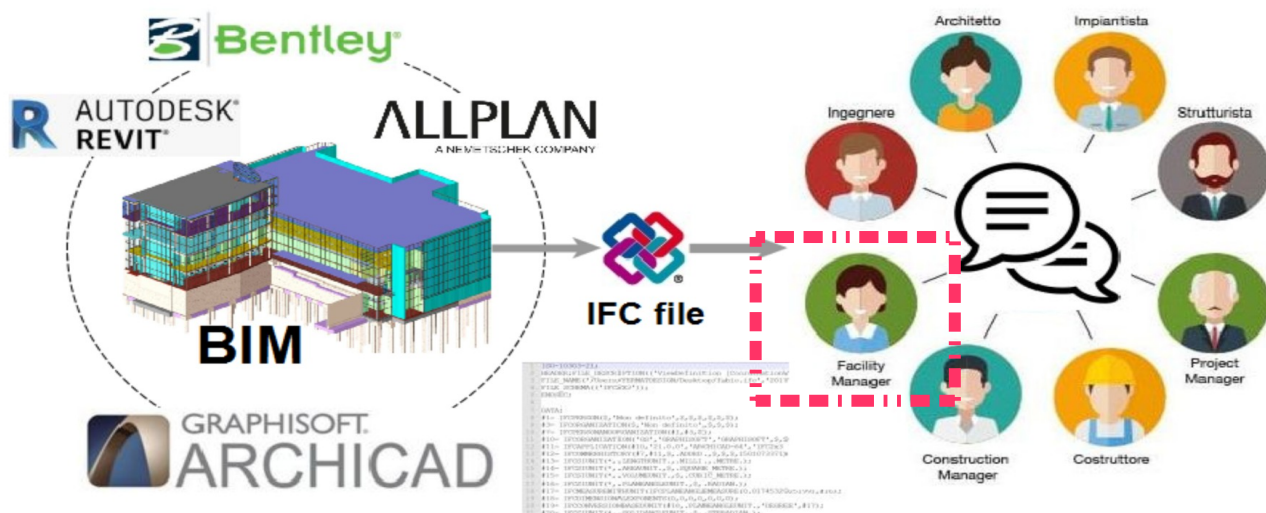
openMAINT supports also the BIM paradigm (Building Information Modeling), which is a conceptual solution with the aim of managing the building design procedure, from the conception of the building to its construction, use and maintenance, and finally to its demolition, if any, and is able to:

- automatically sync the information managed with the 3D building design software using your database, thanks to the open standard format IFC (Industry Foundation Classes)
- represent in a realistic way interiors and assets thanks to a 3D interactive viewer, integrated in the user interface

BIM methodologies have been suggested for a long time because they provide an environment for data sharing that grants every involved company elements like transparency, traceability and completeness of information.

In particular, BIM methodologies allow to automatically:

- acquire and update from a central BIM repository the geometric information and the technical and maintenance records of assets, avoiding the activity duplication
- return to the central BIM repository the information gathered during the operative activities (breakdown resolutions and related SLA, defects, consumptions, etc.)



8.5. The report system

With the basic CMDBuild mechanisms and, in particular, using the report editor and engine, several standard reports and dashboards have been implemented in the openMAINT solution.

Using the same mechanisms, users can easily create new reports customized according to their organization and needs.

Among the available reports and dashboards there are:

- Inventory of properties
- Features of glazed areas
- MTBF - Average time between malfunctions
- Dashboard with the properties and assets situation
- Print of requests for breakdown maintenance (per status / building)
- Print of single Workorder
- Print of Workorders (per status / building)
- Dashboard with the situation of requests for maintenance and Workorders
- Print of warehouse summary (synthetic / analytic)
- Print of warehouse movements (type / item)
- Print of barcode labels
- Print of budget and final balance
- Print of contract list
- Print of pro-forma invoice

8.6. The mobile App

Also for the Facility Management, a lot of operations executed occurs in the field (preventive maintenance, breakdown maintenance, movement, inventory, etc.). It is useful to keep track of them as they are performed, avoiding delays and oversights.

The mobile APP made available by CMDBuild for smartphone and tablet allows to:

- consult and update the data cards of technical and plant engineering assets
- register the activities processed within the expected maintenance and breakdown processes
- print reports
- send to the server pictures of the asset in process
- read a Barcode or a QRcode and receive the complete card of the asset from the server

9. APPENDIX: GLOSSARY

ATTACHMENT

An attachment is a file associated to a card.

The management of the attachments supports the versioning of those files that have been uploaded a few times, with automatic numbering.

See also: Card

ACTIVITY

Activity: workflow step.

An activity can be an interaction with the operator (interactive) or a script that processes operations via API (automatic).

A process instance is a single process that has been activated automatically by the application or manually by an operator.

See also: Process

ATTRIBUTE

The term refers to an attribute of a CMDBuild class (for example in "supplier" class the attributes can be: name, address, phone number, etc.).

CMDBuild allows you to create new attributes (in classes and domains) or edit existing ones.

In the database, every attribute is related to a column in the table which implements the associated class and corresponds, in the Data Management Module, to a data entry field of the specific card for the class management.

See also: Class, Domain, Report, Superclass, Attribute Type

BIM

Method with the aim to support the whole life cycle of a building: from its construction, use and maintenance, to its demolition, if any.

The BIM method (Building Information Modeling) is supported by several IT programs that can interact through an open format for data exchange, called IFC (Industry Foundation Classes).

CMDBuild includes a connector to sync some CI information (technical or maintenance records) and an interactive viewer for the 3D model of the building represented by the IFC file.

See also: CI, GIS

CI

We define CI (Configuration Item) each item that provides IT service to users and has a sufficient detail level for its technical management.

In CMDBuild, the term is applied to a generic context of Asset Management extending the concept usually used in the management of IT infrastructure.

CI examples include: server, workstation, software, plant, electric panel, fire extinguisher, furniture, etc.

See also: Configuration, ITIL

CLASS

A Class is a complex data type having a set of attributes that describe that kind of data.

A Class models an object that has to be managed in the CMDB, such as a company, a building, an asset, a service, etc.

CMDBuild allows the administrator - with the Schema Module - to define new classes or delete / edit existing ones.

A class is represented in the database with a table automatically generated when defining the class and corresponds - in the Data Management Module - to a card for the consultation and update of the cards expected in the model.

See also: Card, Attribute

CMDB

ITIL best practice (Information Technology Infrastructure Library), which has become a "standard de facto" and a non-proprietary system for services management, has introduced the term CMDB referred to the Configuration Item database.

CMDBuild extends the concept of CMDB applying it to a generic Asset Management context.

See also: Database, ITIL

CONFIGURATION

The configuration management process is designed to keep updated and available to other processes the IT items (Configuration Items) information, their relations and their history.

Even if it known as one of the main processes within the ITIL Best Practice, the same concept is used in CMDBuild for generic contexts of Asset Management.

See also: CI, ITIL

DASHBOARD

In CMDBuild, a dashboard corresponds to an application page including one or more different graphic representations, in this way you can immediately hold in evidence some key parameters (KPI) related to management aspects of the Asset Management service.

See also: Report

DATABASE

The term refers to a structured collection of information hosted on a server, as well as utility software that handle this information for tasks such as initialization, allocation, optimization, backup, etc.

CMDBuild relies on PostgreSQL, the most powerful, reliable, professional and open source database, and uses its advanced features and object-oriented structure.

The Asset Management database, implemented on the basis of the CMDBuild logics and philosophy, is also indicated as CMDB (Configuration Management Data Base).

DOMAIN

A domain is a relation between two classes.

A domain has a name, two descriptions (direct and inverse), classes codes, cardinality and attributes.

The system administrator, using the Administration Module, is able to define new domains or delete / edit existing ones.

It is possible to define custom attributes for each domain.

See also: Class, Relation

DATA FILTER

A data filter is a restriction of the list of those elements contained in a class, obtained by specifying boolean conditions (equal, not equal, contains, begins with, etc.) on those possible values that can be accepted by every class attribute.

Data filters can be defined and used exceptionally, otherwise they can be stored by the operator and then recalled, or configured by the Administrator and made available by operators.

See also: Class, View

GIS

A GIS is a system able to produce, manage and analyze spatial data by associating geographic elements to one or more alphanumeric descriptions.

GIS functionalities in CMDBuild allow you to create geometric attributes (in addition to standard attributes) that represent, on plans / maps, markers position (assets), polylines (transmission lines) and polygons (floors, rooms, etc.).

See also: BIM

GUI FRAMEWORK

It is a framework provided by CMDBuild to simplify the implementation of external custom user interfaces and to grant a simplified access to non-technical users. They can be issued onto any webportals and can be used with CMDBuild through the standard REST webservice.

The CMDBuild GUI Framework is based on javascript JQuery libraries.

See also: Mobile, Webservice

ITIL

"Best practices" system that established a "standard de facto"; it is a non-proprietary system for the management of IT services, following a process-oriented schema (Information Technology Infrastructure Library).

ITIL processes include: Service Support, Change Management and the Configuration Management.

For each process, ITIL handles description, basic components, criteria and tools for quality management, roles and responsibilities of the resources involved, integration points with other processes (to avoid duplications and inefficiencies).

CMDBuild uses some ITIL concepts and applies them to a generic context of Asset Management.

See also: Configuration

LOOKUP

The term "Lookup" refers to a pair of values (Code, Description) set by the administrator in the Administration Module.

These values are used to bind the user's choice (at the form filling time) to one of the preset values (also called multiple choice or picklist).

With the Administration Module it is possible to define new "LookUp" tables according to organization needs.

See also: Attribute type

MOBILE

It is a user interface for mobile tools (smartphones and tablets).

It is implemented as multi-platform app (iOS, Android) and can be used with the CMDB through the REST webservice.

See also: Webservice

PROCESS

The term process (or workflow) refers to a sequence of steps that realize an action.

For each process (type of process) a new process instance will be started when users have to carry out an action on assets according to a procedure implemented as workflow.

A process is activated by starting a new process (filling related form) and ends when the last workflow step is executed.

The workflows managed in CMDBuild are described in the standard markup language XPDL (XML Process Definition Language), ruled by the WPMC (WorkFlow Management Coalition).

See also: Workflow step

RELATION

A relation is a link between two CMDBuild cards or, in other words, an instance of a given domain.

A relation is defined by a pair of unique card identifiers, a domain and attributes (if any).

CMDBuild allows users, through the Management Module, to define new relations among the cards stored in the CMDB.

See also: Class, Domain

REPORT

The term refers to a document (PDF or CSV) containing information extracted from one or more classes and related domains.

The reports can be configured in the Administration Module importing in XML format the description of the layout designed with the visual editor JasperReports. They can be provided to operators in the application menu.

CMDBuild users can print reports using the Management Module, which will result as printouts, charts, documents, labels, etc.

See also: Class, Domain, Database

CARDS

The term "card" refers to an element stored in a class (corresponding to the record of a table in the database).

A card is defined by a set of values, i.e. the attributes defined for its class.

CMDBuild users, through the Management Module, are able to store new cards and update / delete existing ones.

Card information is stored in the database and, more exactly, in the table/columns created for that class (Administration Module).

See also: Class, Attribute

SUPERCLASS

A superclass is an abstract class used as template to define attributes shared between subclasses. From the abstract class, or from abstract class hierarchies, you can derive real classes that contain data and include both shared attributes (specified in the superclass) and specific subclass attributes, besides the relations on the superclass domains and on specific domains.

For example, you can define the superclass "Company" with some basic attributes (VAT number, Business name, Phone number, etc.) and the derived subclasses "Customers" and "Suppliers", each one with both generic attributes of the superclass and its own attributes and relations.

See also: Class, Attribute

TENANT

A "tenant", in CMDBuild, is a part of the CMDDB reserved to users belonging to a suborganization of the CMDBuild instance (a Group Society, a Seat, a Division, etc.).

Working in "multitenant" mode, every user works only on data of his/her suborganization and, in case, on common data: "tenants".

The list of usable Tenants can be defined from an applicable class of CMDBuild (seats, companies, customers, etc.) or from a database custom function, where you can implement complex visibility rules.

ATTRIBUTE TYPE

Each attribute has a data type that represents attribute information and management.

The type of attribute and its management modes are defined in the Administration Module.

CMDBuild manages the following attribute types: "Boolean", "Date", "Decimal", "Double", "File", "Formula", "Inet" (IP address), "Integer", "Link", "LookUp" (lists set in "Settings" / "LookUp"), "Multi-value LookUp", "Reference" (foreign key), "String", "Text", "TimeStamp".

See also: Attribute

VIEW

A view includes cards defined with logic criteria of filters applied to one or more CMDDB classes.

In particular, a view can be defined in CMDBuild by applying a filter to a class (so it will contain a reduced set of the same rows) or specifying an SQL function which extracts attributes from one or more related classes.

The first view type maintains all functionalities available for a class, the second one allows the sole display and search with fast filter.

See also: Class, Filter

WEBSERVICE

A webservice is an interface that describes a collection of methods, available over a network and working using XML messages.

With webservices, an application allows other information and applications to interact with its methods.

CMDBuild includes REST webservices, which are provided to external applications to read or write data on CMDDB or process operations.

WIDGET

A widget is a component of a GUI that improves user interaction with the application.

CMDBuild uses widgets (presented as "buttons") that can be placed on cards or processes. The buttons open popup windows that allow you to consult or insert data or process other operations.

CMDBuild includes some standards widgets to process the most common operations, but it also supplies the specifications to implement other custom widgets.