Enterprise System Concepts

1 Abstract

The CEISAR mission is to help Enterprise to describe and improve define Enterprise Systems. It implies defining a common **vision** of Enterprise Systems and establishing common **definitions.**
Thanks to this common language we allow not only simplifications and consistency in our white papers, but we also provide a common language between our sponsors who can more easily exchange experiences or compare provider’s offers (approach and tools).
Enterprises require a **global** vision of their System, including Processes, Business Entities, common functions, organization, and IT.
An **Enterprise** is an economic agent which creates economic **Value** under the **same responsibility.** An Enterprise may be composed of Enterprises. An Enterprise System is decomposed into 3 levels:

- The **Core Business System**: represents **what** is the Business independently from who does what
- The **Organization System** represents **how** people and automates execute the Business
- The **IT System** represents what is automated.

Core Business System description requires 3 concepts:
Business Entity
An Entity represents an Object of the real world.
An Entity has an **identifier** which allows recognizing the different instances of the same Entity.
Examples: Person, Customer, Product, Contract, Account, Operation …
Business Process
A Business Process is a sequence of Business Functions triggered by a single Business Event for a unique Business objective and for a main Business Process Customer.
A Business Process is always defined by a **verb** + a **Business Entity** (+ a complement).
Example: Create a Product, Subscribe a Contract, Pay a Bill, …
Business Function
A Function is an action which brings Business "Value" in a Business Process.
Example: check data entry, compute price, send message to Customer.
A Business Function can de decomposed into other Business Functions.
Example: Check validity of input data, compute price, send a message to customer…

Organization System description requires 4 concepts
Organization Entity
Organization Entities define all resources used by the company: Organization Actors, Organization Units, Rights and Duties, location, building…
Organization Process
For each Business Process, may exist different scenarios of assignments of Activities to Actors, because for the same Business Process may exist different organizations. Each scenario is called an Organization Process.
Organization Function
To define with accuracy an Organization Process, Business Functions are not sufficient: we must add Organization Functions like “am I Authorized”, or “send a message to the manager”, or “ask someone to check” …
Activity
An Activity is a set of Functions inside an Organization Process which is **schedulable**, which means assigned to one Organization Actor (human or automate).
Activities can be:

- totally manual (example: check a signature): **manual** Activity
- totally automatic (example: produce Payroll statements): **automatic** Activity
- human but helped by IT (example: any transaction done on a keyboard): **transactional** Activity

IT System description requires 3 concepts
Software Service

A Software Service is a callable piece of software defined by an **Interface** (what the caller must know) and an **implementation** (hidden part which does the work).
When a Function is computerized, it is implemented as a Software Service.
A Software Service may call **other** Software Services which embed part of the complexity.

Class

The Software Services are grouped into Classes
CEISAR suggests designing a System using Object Oriented Approach even if the implementation is done through classical technologies. A **Class** is the logical definition of Objects sharing common Attributes (or Data) and Methods (or Software Services). An Entity is implemented with one or several Classes.

Block

A Block is a set of Blocks or Classes. As for any complex system, like a Town, System simplification comes from hierarchical decomposition into independent **Blocks**.
Ex: Accounting Block, CRM Block, Reference data Block, …
The structure of Blocks + their interfaces is called "**Block Cartography**".
The objective is to build a Block hierarchy which minimizes exchanges between Blocks.

Architecture

Architecture is defined as what can be **shared.**

Business Architecture includes:

- Shared Business Entities
- Shared Business Functions
- Shared Business Process Models
- Shared decomposition of Business Process and Entity Domains.

Organization architecture includes

- Shared Organization Entities: Organization Actors, Organization structure
- Shared Organization Functions like Security and Activity assignment
- Shared decomposition of Organization Process and Activity Domains.

IT Architecture includes

- Shared Block Cartography with shared Interfaces
- Shared Classes
- Shared Software Services
- Shared Data
- Development Architecture: languages and Development tools (Specification, development, test, integration, documentation, …)
- Operation Architecture: common tools and resources to execute software hardware, network, OS, DBMS, middleware, deployment tools

Shared Architecture Governance and Approach

Architecture follows same decomposition than Enterprises.
A Group may define a **Group Architecture** applied to all companies in the Group.
Each company may also define a **Company Architecture** which is based on Group Architecture, and **adds** sharable elements at its level.

2 Objectives

The CEISAR mission is to help Enterprise to describe and improve define Enterprise Systems. It implies defining a common **vision** of Enterprise Systems and establishing common **definitions.**
The CEISAR work is only useful if its white papers are:

- **Simple** enough to be readable my majority of people who could make usage of them, which means the smallest number of Concepts to describe an Enterprise System and a clear definition of each of them.
- Pertinent enough to represent the real **complexity** of Enterprise Systems.

Thanks to this common language we allow not only simplifications and consistency in our white papers, but we also provide a common language between our sponsors who can more easily exchange experiences or compare provider's offers (approach and tools).

2.1 Increasing complexity of Enterprise Systems

The CEISAR only has one objective: how to **simplify** complex Enterprise System to allow easy **evolutions**.
Complexity comes from different trends:

- Everything **increases:** number of Products, Processes, Partners, Countries, …
- **Number** of IT Applications increases not only because business becomes more and more complex, but also because more and more activities can be computerized.
- The overall system must **evolve** at an increasing speed.
- Companies look for ways to adapt Product or Organization (defined as "who does what") **faster** than their competitors, which means offer tools (like Product Factory, or Rule engines) to business professionals so that they can directly update Products and Organization.
- Groups try to **share same data** between different Business Units, like customer information. They do not want to capture data several times and create **bridges** between independent systems. Some Groups try to share not only data but also processes and softwares.
- Extended Enterprise: companies partner more and more. They connect their information System to the **external** world: external distribution networks, customers, prospects are directly linked to the Information System.
- They often keep application software for 20 years, which means maintaining software with different **technology generations.**

2.2 Consequences of complexity

50% of projects are stopped before delivery.
90% of delivered projects are late.
100% of projects fail to deliver all the functionality defined at the outset.

- **Project Difficulty:** many large specific projects have failed because of the complexity linked to :
  - Number of business functions to be delivered, and instability of requirements
  - Number of technical layers on which the application must run
  - Number of coexisting systems
  - Number of intervening parties in the project process
- **Time** : lack of flexibility, and slow evolutions, while time to market is the key point in a competitive world. Introducing new business functions often destabilizes the IS.
- **Money**:
  - **Organization money**: Process discontinuity, inconsistencies, lack of automation means low productivity and specialization of Users who have difficulties to move from one position to another in the Enterprise
  - **IT Money**: Maintenance costs, integration costs, operations costs, training costs. Saving time also means saving money while budget cuts are so frequent
- **User specialization**: the global Organization lacks flexibility because Processes are discontinuous and user interface is inconsistent.

- Lack of **quality** of data
- Difficulty to take advantage of **new technologies:** difficult to take advantage of new IT possibilities without losing the business investment already done when developing applications

2.3 Many good principles have already been defined, but they are based on many concepts

The concept of Enterprise System is widely used. But as many widely used concepts, definitions vary. It generally is considered as including the IT system, but also business parts like definition of Business Concepts.
Good vision come from organizations like ITIL, MDA, OMG, Praxime, Togaf, UML…
But many different concepts are used.
Example: for "**Architecture**" exist many concepts like:

- Business Architecture
- Component Architecture
- Data Architecture
- Deployment Architecture
- Enterprise Architecture
- Functional Architecture
- Hardware Architecture
- Infrastructure
- Integration Architecture
- Logical Architecture
- Middleware Architecture
- Organization architecture
- Platform Architecture
- Portal Architecture
- Process Architecture
- Service Architecture
- SOA
- Software Architecture
- Technical architecture
- Transactional Architecture
- Web Architecture

Example: for "**Objectives**" also exist many concepts like:

- Business Objectives
- Operational Objectives
- Organization Objectives
- Strategic Objectives
- Structural Objectives
- Technical Objectives

Are they all really useful?

2.4 CEISAR proposes a reduced number of concepts

Enterprises require a **global** vision of their System, including Processes, Business Entities, common functions, organization, and IT.
They want that this global vision be based on a little number of consistent concepts.
CEISAR mission is to

- take advantage of good ideas already presented by the different initiatives, methodologies and approach available to-day
- but summarize them with a common and consistent language accessible to everyone

Smallest number of concepts but sufficient to represent complexity of Enterprise Systems:
The first list of Concepts defined in this document was sufficient for describing first Ceisar White Papers on “Simplify Legacy Systems” and “Find Business Entities”. We finally propose to use 10 fundamental concepts which are enough to represent the first white papers. These 10 Concepts are: Business Entity, Business Process, Business Function, Organization Process, Activity, Organization Function, Organization Entity, Software Service, Class, and Block.
For following White Papers CEISAR will reuse these same concepts. If necessary CEISAR will add some concepts: we preferred this approach, rather than defining from existing methodologies all concepts which could be useful one day, because we think that this is the only way to minimize the number of concepts and make Ceisar White papers understandable by anyone.

3 What is an Enterprise?

First questions when we describe Enterprise System: what is an **Enterprise?**

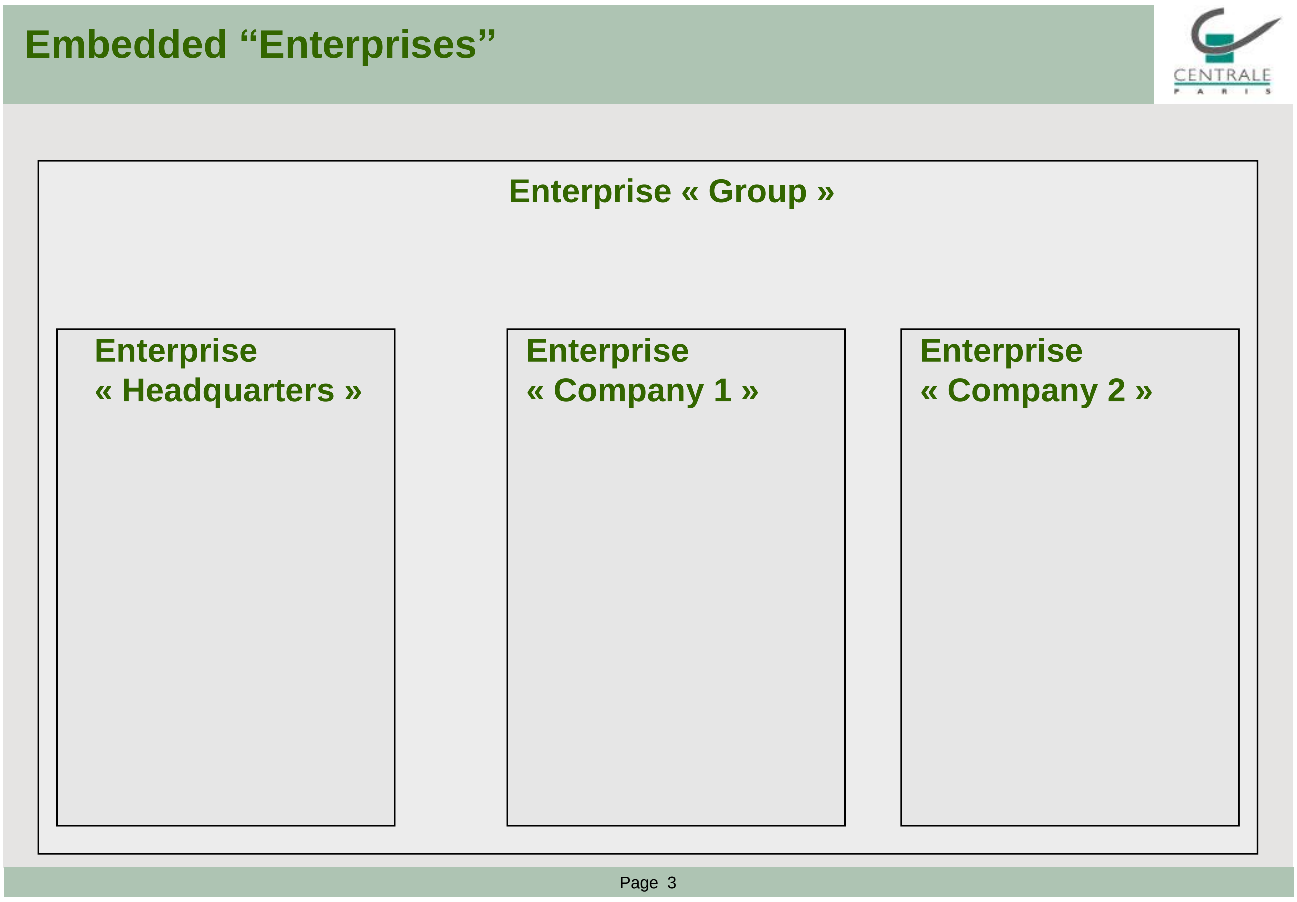
3.1 Enterprise Definition

An Enterprise is an economic agent which creates economic **Value** under the **same responsibility.**.
An Enterprise cannot be reduced to a **Legal** definition. An Enterprise can be a Company, a part of a Company, a network of Companies, an Association, a Governmental Organization, or a Research center.
Its “business” definition must include the **extended** Enterprise. For example

- A Group of companies may represent a real Economic Entity with a unique decision center, without being described as one big Legal Entity.
- When a company sells its products through distribution networks which belong to other Legal entities, we must define Cross Processes and standardize Business Object definitions, identifiers, which means that they represent together a global Enterprise, the Extended Enterprise.

3.2 An Enterprise composed of Enterprises

An Enterprise may be composed of Enterprises.
A Group composed of 2 companies is an Enterprise. Each company is also an Enterprise. And consider that support functions executed by Headquarters are also an “Enterprise”.
Each Company may in turn be decomposed into several smaller Enterprises.



Embedded “Enterprises”

Enterprise « Group »

Enterprise
« Headquarters »

Enterprise
« Company 1 »

Enterprise
« Company 2 »

Page 3

3.3 How to define the scope of an Enterprise ?

An Enterprise is defined by a unique top management and a common objective.
The Enterprise Scope can be defined by a combination of one or several different dimensions.

- By Territory (like Country or Region)
- By Product Line
- By Organization layer (like Front Office Canal, Back Office)
- By Customer Target (like Commercial/personal segments)

The choice is the responsibility of the Enterprise management.
Today trends are:

- decrease of Territory criteria,
- increase of Product Line (because Products become more and more international),
- increase of Organization Layer (because extended Enterprise).

3.3.1 Territory

Enterprise can be specific to each Territory (Country or region).

3.3.2 Product Line

An Enterprise can be specific to each Product Line.
For example if a Bank also sells Insurance products, they can use 2 different Enterprises.
The decomposition is **hierarchical**. Ex: Insurance Products can be divided into Life and Non Life Products, and may require an Enterprise for Life Products and another one for non-life products.

3.3.3 Organization

Enterprise Systems can also be specific to each Organization.
For example may exist one Enterprise System by distribution network, and one Enterprise System for Back Office.
Several Business Process Domains are defined in an Enterprise like:

- Define Strategy
- Design a Product
- Order from Provider
- Produce the Product
- Distribute the Product: sell and deliver
- Deliver Services: maintenance
- Check results and react
- Manage Resources
  - Manage the internal people
  - Manage a distribution Network
  - Manage finances
  - Manage offices, supplies, …

When management of these Domains are clearly separated, it means that exist sub-enterprises.
Example: Domain “Distribute Policy” can be executed by a different Enterprise than “Manage Claims”.

Line of Products and Functional Domains are 2 independent dimensions.
For example the distribution domain can be common or not to Line of Products.

3.3.4 Customer target

An Enterprise can be defined for each Customer Target: Personal, Professional, Commercial.
Then each category can be subdivided into smaller. For example, Commercial can be classified as Small or Large companies.

4 What is an Enterprise System?

4.1 The 3 domains of an Enterprise System: Business, Organization and IT.

An Enterprise System describes all Enterprise mechanisms through:

- a Business perspective which defines Products, Customers, and Business Processes
- an Organization perspective: who does what, when and where
- and their translation inside an IT system

Traditionally, for each Business Domain exists a Software Application which together defines:

- **Business Processes**
- How **Actors** manage these Processes
- Related **software**

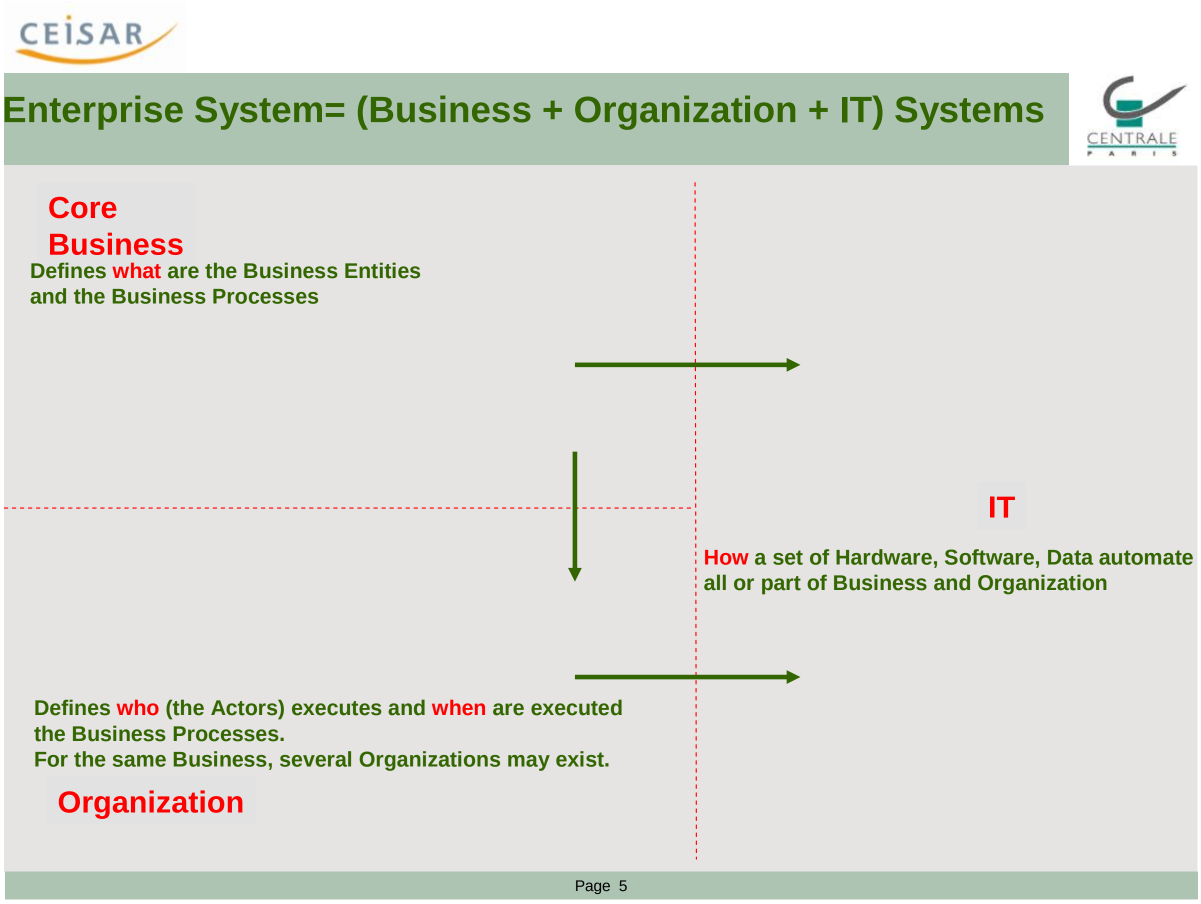
But

- For the same basic Business Process may exist several ways to organize and share work: so we must **split** the **Business** and the **Organization**.
- A Business Process is decomposed into Business Functions. All these Functions do not belong to the same Block of software: for example for the Process “Subscribe a Contract”, the Function “am I authorized” is not in the same Block than the Function “compute price” or the Function “Update Account”. It means that **Business Processes** and **Software** are not strictly parallel.

These remarks mean that an **Enterprise System** is composed of 3 parts:

- The **Business System**: represents what is the Business independently from who does what
- The **Organization System** represents how people and automates execute the Business
- The **IT System** represents what is automated.

Our objective is to describe for these 3 parts the **minimum number of concepts** that we will reuse in our white papers.



Enterprise System= (Business + Organization + IT) Systems

Core Business

Defines what are the Business Entities and the Business Processes

IT

How a set of Hardware, Software, Data automate all or part of Business and Organization

Defines who (the Actors) executes and when are executed the Business Processes.
For the same Business, several Organizations may exist.

Organization

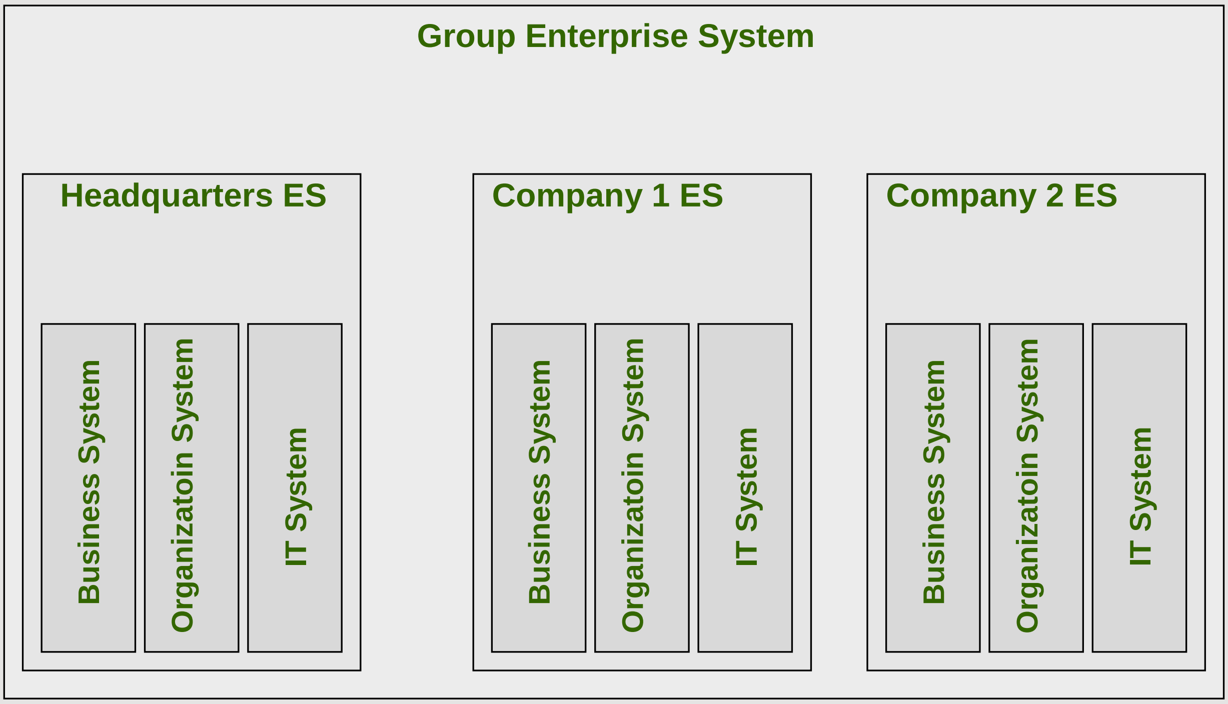
Page 5

4.2 One or several Enterprise Systems in a Group?

As an Enterprise is composed of several Enterprises, an Enterprise System can be composed of several Enterprise Systems. **Enterprise Systems can be embedded.**

We called “Headquarters” the part of the Group which works for Business Activity common to the different Companies of the Group, like Group Reporting, or Group Marketing, or Executive Management. The Headquarters require an Enterprise System as any Company of the Group.

Embedded “Enterprise Systems”



Group Enterprise System

Headquarters ES

Business System

Organizatoin System

IT System

Company 1 ES

Business System

Organizatoin System

IT System

Company 2 ES

Business System

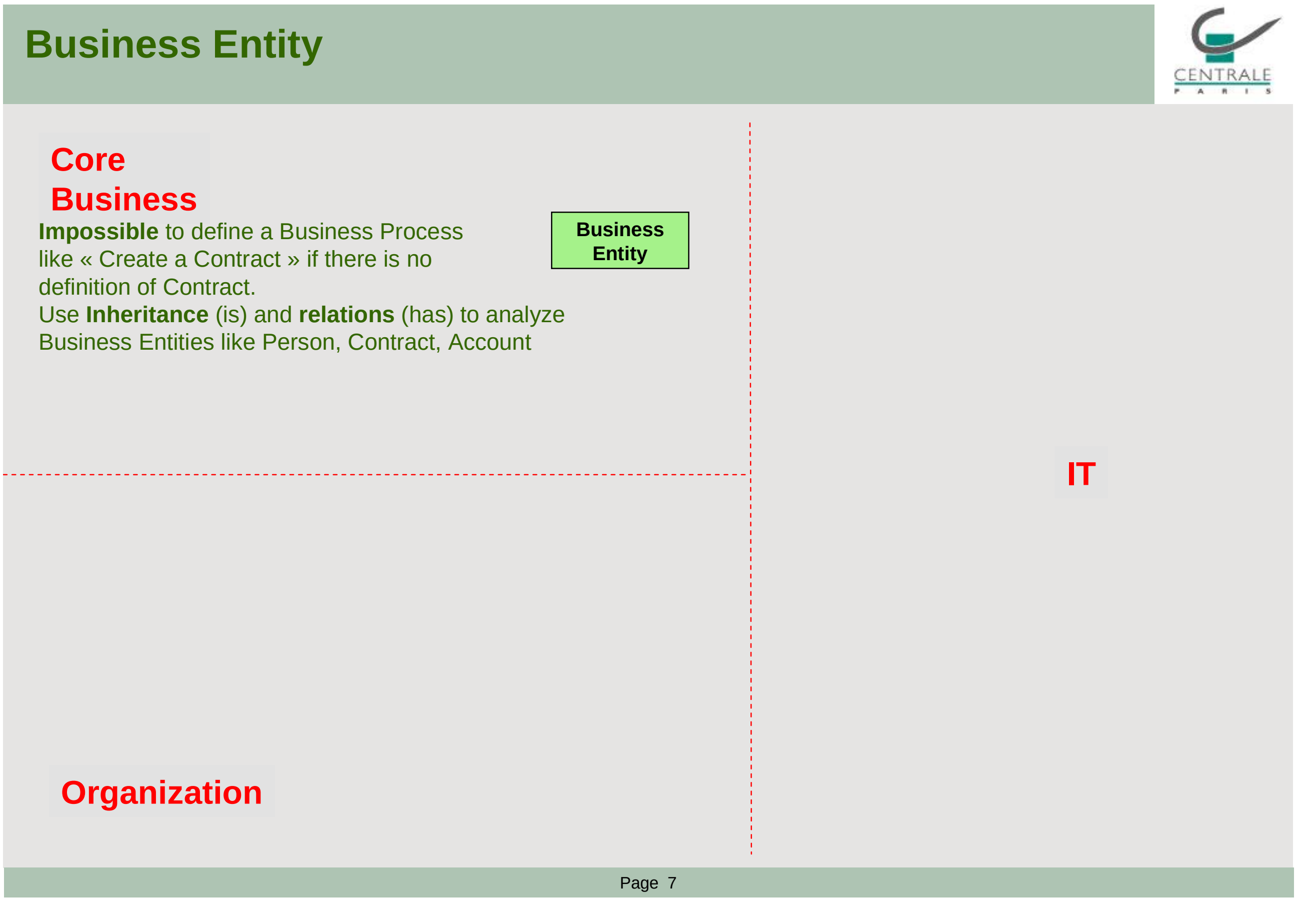
Organizatoin System

IT System

5 Business Entity

It is impossible to define Business Processes like “Create a Person” or “Subscribe a Contract” if we do not know the definition of “Person” or “Contract”.
This is why the first step must be to define the Business Entities.
An Entity represents an Object of the real world.
An Entity has an **identifier** which allows recognizing the different instances of the same Entity.
Examples: Person, Customer, Product, Contract, Account, Operation …

When we give a definition for a Business Entity, we often reuse other Business Entities, which means that there are links between these Entities. Links between entities are Relations or Inheritance.
If an Entity **has** another Entity, or relates to another Entity: this is a **Relation**.
Example: a Person has an Address
If an Entity **is** a specialization of another one, it **inherits** from it.
Example: a Life Insurance Contract inherits from an Insurance Contract which itself inherits from a Contract.
Relation and inheritance are defined thanks to a **Entity Relation Diagram.**
The quality of the Business Entity Relation Diagram which describes Relations and Inheritance, defines the quality of the Language between Business and IT people, and the quality of the software structure: this is why it is so important to design it carefully.



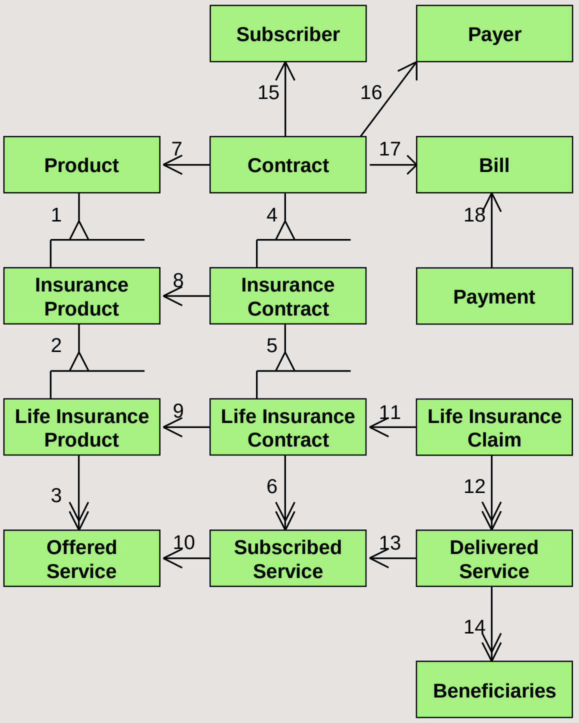
Business Entity

Core Business

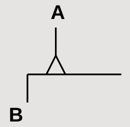
Impossible to define a Business Process like « Create a Contract » if there is no definition of Contract.
Use **Inheritance** (is) and **relations** (has) to analyze Business Entities like Person, Contract, Account

Business Entity links are represented through an Entity Relation diagram which represents inheritance and relation between Entities.

Entity Model Example



1 An Insurance Product **is** a Product
2 A Life Insurance Product **is** an Insurance Product
3 A Life Insurance Product **has** Offered Service
4 An Insurance Contract **is** a Contract
5 A Life Insurance Contract **is** an Insurance Contract
6 A Life Insurance Contract **has** Subscribed Services
7 A Contract relates to a Product
8 An Insurance Contract **relates to** an Insurance Product
9 A Life Insurance Contract **relates to** a Life Insurance Product
10 A Subscribed Service **relates to** an Offered Service
11 A Life Insurance Claim **relates to** a Life Insurance Contract
12 A Life Insurance Claim **has** Delivered Services
13 A Delivered Service **relates to** a Subscribed Service
14 A Delivered Service **relates to** a Beneficiary
15 A Contract **has** a Subscriber
16 A Contract **has** a Payer
17 A Contract Has a Bill
18 A Payment **has** a Bill



Means that B inherits from A,
or B **is** a specialisation of A

A ← B Means that B **has** one A, or B relates to one A

A ↞ B Means that B **has** several A, or B **relates to** several A

Page 8

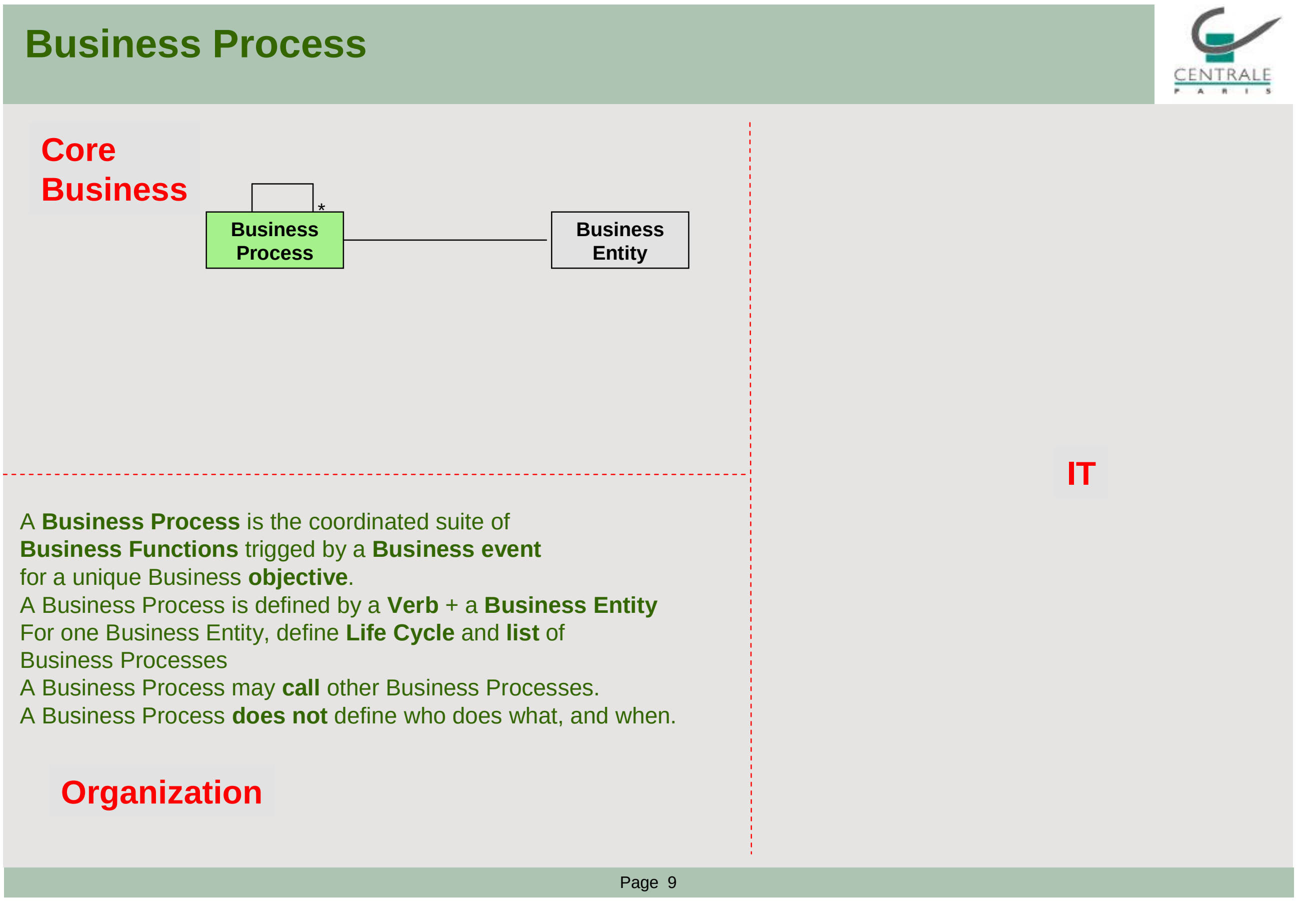
6 Business Process

A Business Process is a sequence of Business Functions triggered by a single Business Event for a unique Business objective and for a main Business Process Customer.

A Business Process is always defined by a **verb** + a **Business Entity** (+ a complement).
Example: Create a Product, Modify a Product, Create a Person, Subscribe a Contract, Modify a Legal Entity, Pay a Bill, …

An easy way to find main processes is to analyze **Life Cycle** of a Business Entity.

A Business Process A may call another Business Process B.
But Business Process B may also be directly executed.
Example: the Business Process A “Welcome a new employee” calls the process “attribute an email address”, or “attribute a computer”, … These Processes may also be directly executed: they are not called sub-processes.



Business Process

Core Business

IT

A **Business Process** is the coordinated suite of **Business Functions** trigged by a **Business event** for a unique Business **objective**.
A Business Process is defined by a **Verb** + a **Business Entity**
For one Business Entity, define **Life Cycle** and **list** of Business Processes
A Business Process may **call** other Business Processes.
A Business Process **does not** define who does what, and when.

Organization

Page 9

7 Business Process Domain and Entity Domain

Several Entities can be grouped into one Entity Domain.
Example: “Person” and “Legal Entity” belong to the Entity Domain “Actor.

Several Business Processes can be grouped into one Business Process Domain.
Example: “Update Accounts by Accounting Entries” and “Produce Accounting Statements” belong the Business Process Domain “Manage Accounts” (we prefer to use “manage accounts” rather than “accounting” to be able to identify by the verb that is a Business Process Domain and not an Entity Domain).
This hierarchy of Business Domain is just useful to summarize the Business of the Company into a small number of large domains.
Different classifications can be built according to needs like statistics, definition of authorizations …

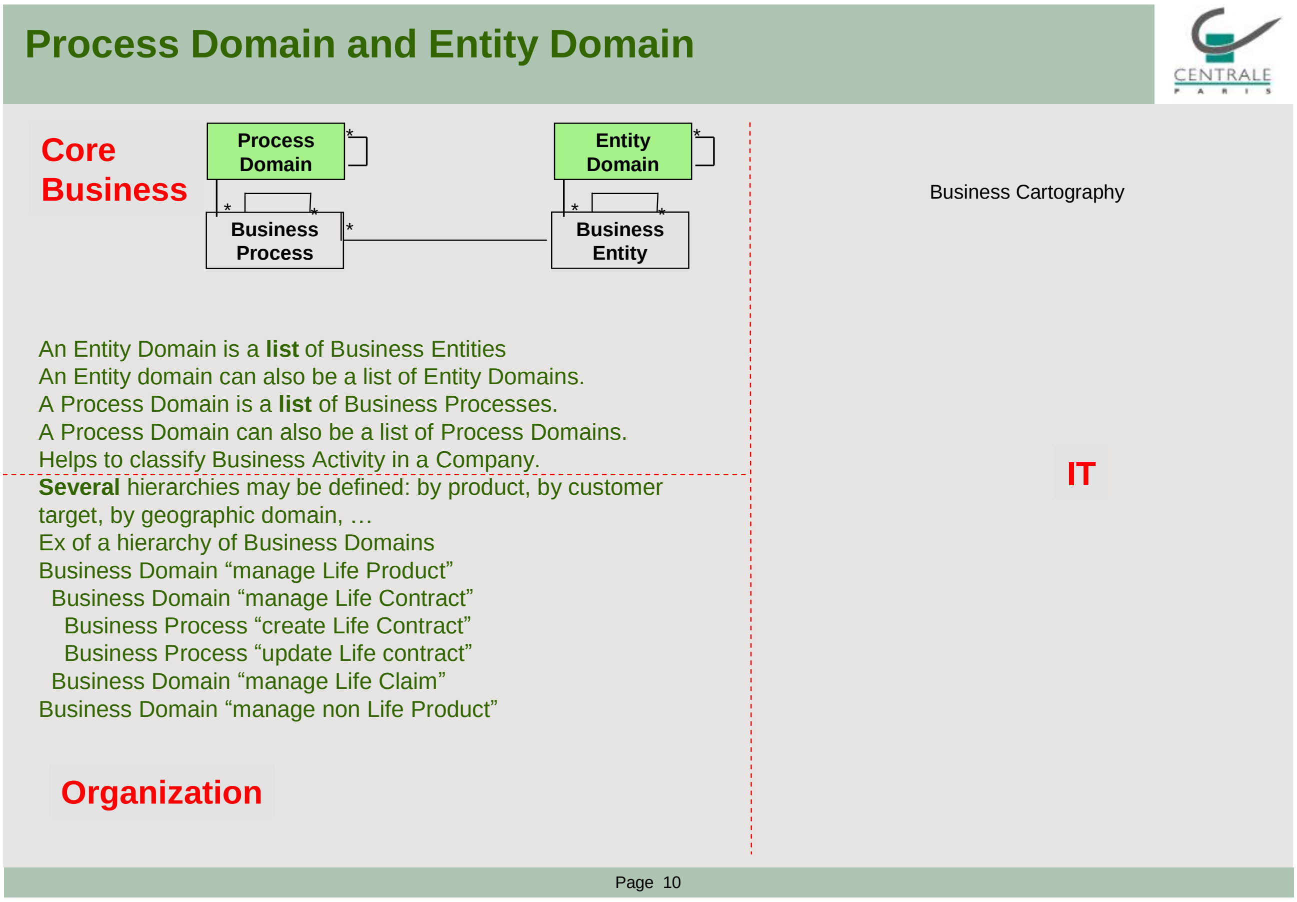
A Domain is a recursive concept, which means that a **Domain can be composed of Domains.**

Example of Business Process Domains:

- Manage Customer relations
- Manage Contracts
- Manage Resources like Employees, or premises.

A classification often used is:

- **Operational** Process Domain: like manage Customers, manage Products, manage Contracts, manage Third Party Accounts
- **Referential** Process Domain: like manage Actors, manage Organization, manage Rights (some prefer to include “manage Products” in this Domain)
- **Support** Process Domain: like manage employees, manage IT system, manage premises
- **Business Intelligence** Process Domain: like manage budget, provide reports



Process Domain and Entity Domain

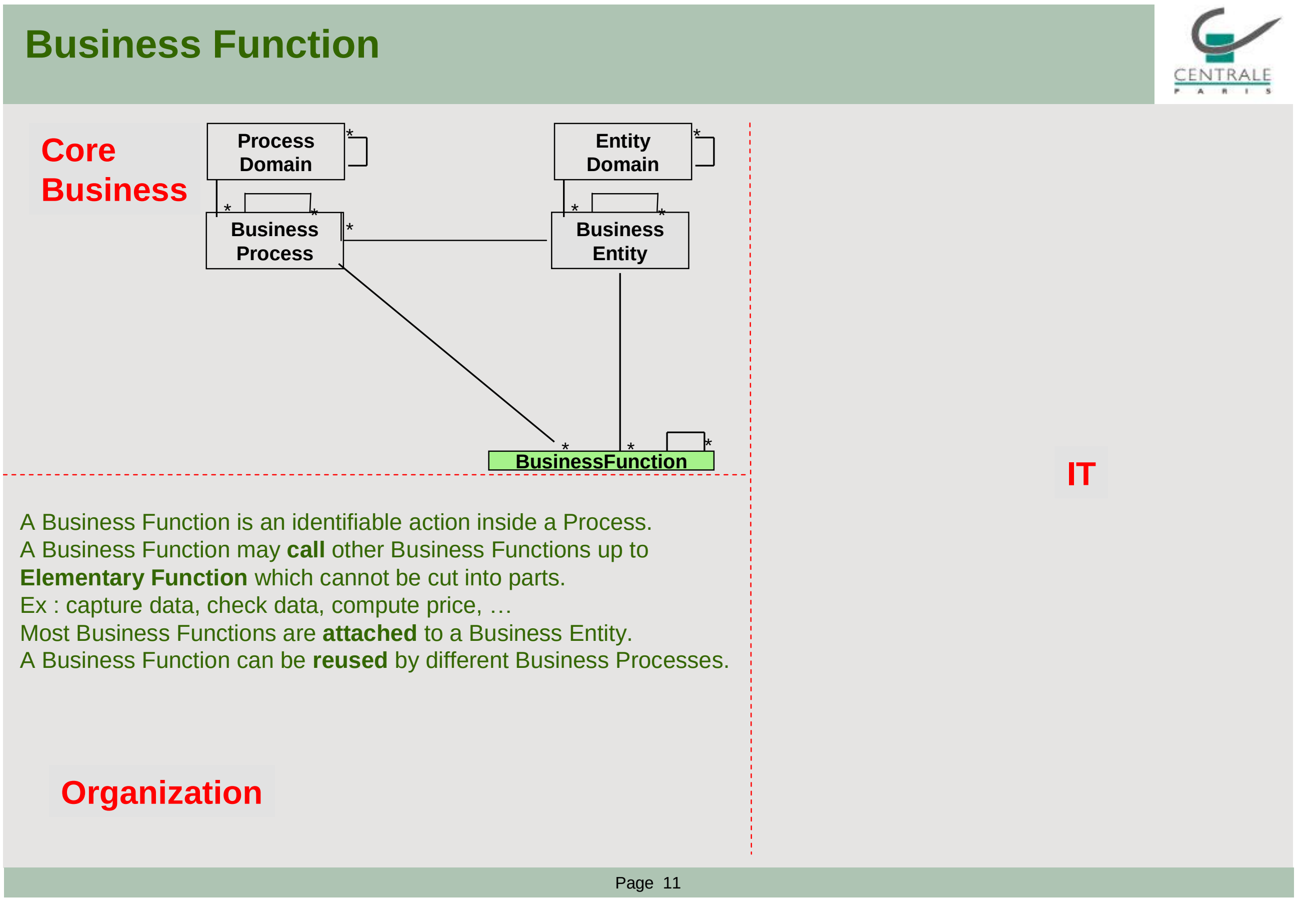
An Entity Domain is a **list** of Business Entities
An Entity domain can also be a list of Entity Domains.
A Process Domain is a **list** of Business Processes.
A Process Domain can also be a list of Process Domains.
Helps to classify Business Activity in a Company.
Several hierarchies may be defined: by product, by customer target, by geographic domain, …
Ex of a hierarchy of Business Domains
Business Domain “manage Life Product”
  Business Domain “manage Life Contract”
    Business Process “create Life Contract”
    Business Process “update Life contract”
  Business Domain “manage Life Claim”
Business Domain “manage non Life Product”

Page 10

8 Business Function

A Function is an action which brings Business "Value" in a Business Process.
Example: check data entry, compute price, send message to Customer.
A Business function can de decomposed into other Business Functions.
Do not make confusion with Organization Functions (see later).
The decomposition of a Business Process into Functions does not define who will do the job: we will define it later in “Organization”.
“Check Contract Data” is not a Business Process, but a Function which is part of the Business Process “Subscribe the Contract”, whose Business event could be “a Customer asks to Subscribe”.
The most difficult task is to identify the **common Functions**: each Business Line is specific, and all the talent of the Architect is to convince that it is possible to respect each specificity with common functions!
When we break down a Business Process into Functions, we must be guided by

- the fact that one or **several Actors** will participate in this Business Processes
- the fact that some Functions will not be **immediate**



Business Function

Core Business

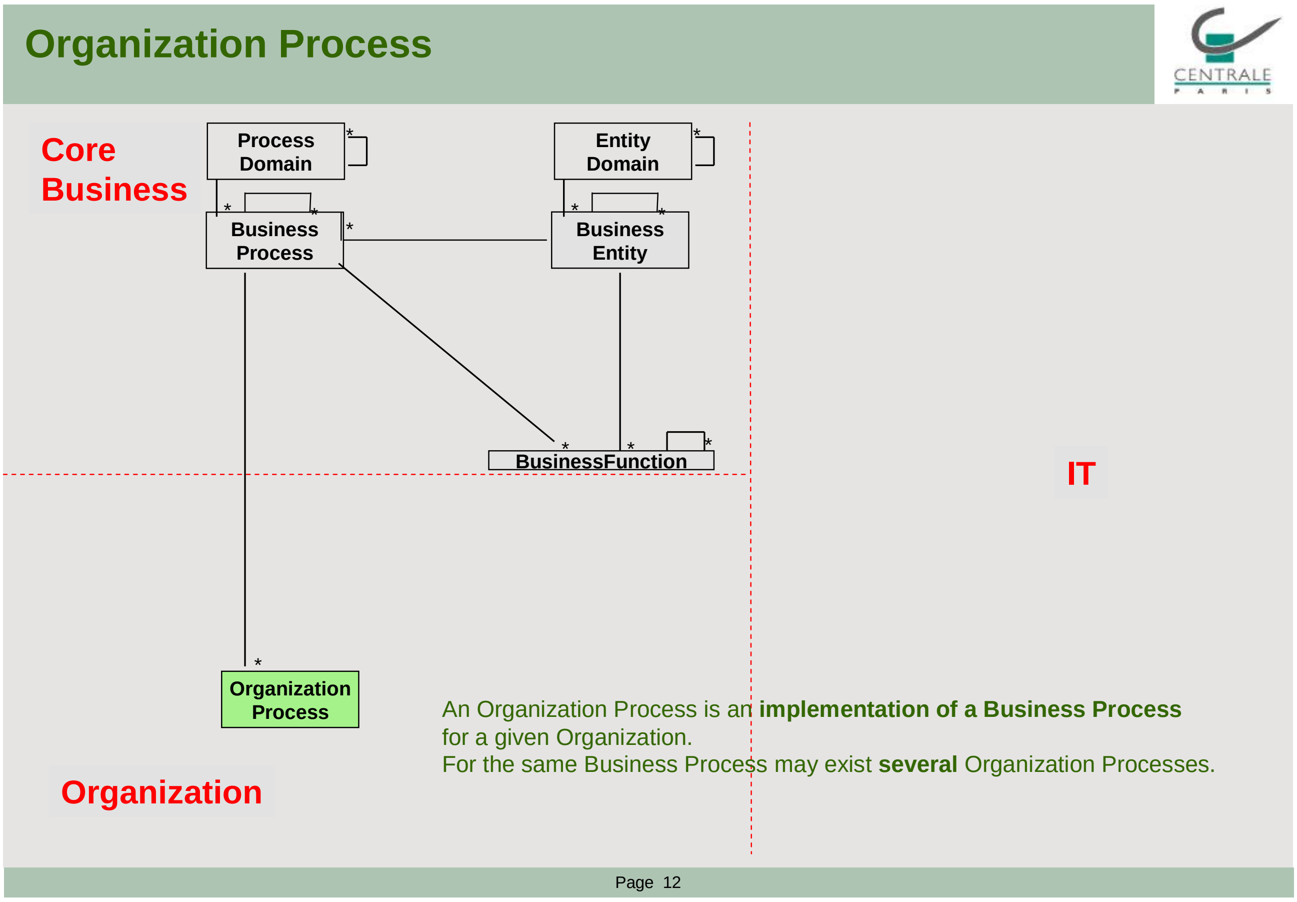
A Business Function is an identifiable action inside a Process.
A Business Function may **call** other Business Functions up to **Elementary Function** which cannot be cut into parts.
Ex : capture data, check data, compute price, …
Most Business Functions are **attached** to a Business Entity.
A Business Function can be **reused** by different Business Processes.

Organization

Page 11

9 Organization Process

Now we must define how a company is organized to accomplish its Business.
For the same Business defined by the Business Processes, may exist many different organizations.
Companies often change their Organization; they do not often change their Core Business.
For each Business Process, may exist different scenarios of assignments of Activities to Actors, because for the same Business Process may exist different organizations.
Each scenario is called an Organization Process.
Definition of Organization Process changes with the internal organization of the company, while the Business Process is relatively stable.
Different Organization Units of the same Group may share the same Business Processes and apply different Organization Processes.



Organization Process

An Organization Process is an **implementation of a Business Process** for a given Organization.
For the same Business Process may exist **several** Organization Processes.

Page 12

Remark: how to simplify description of Organization Processes?

Business Processes are stable, while Organization Processes move with the Organization.
The simplest system is based on:

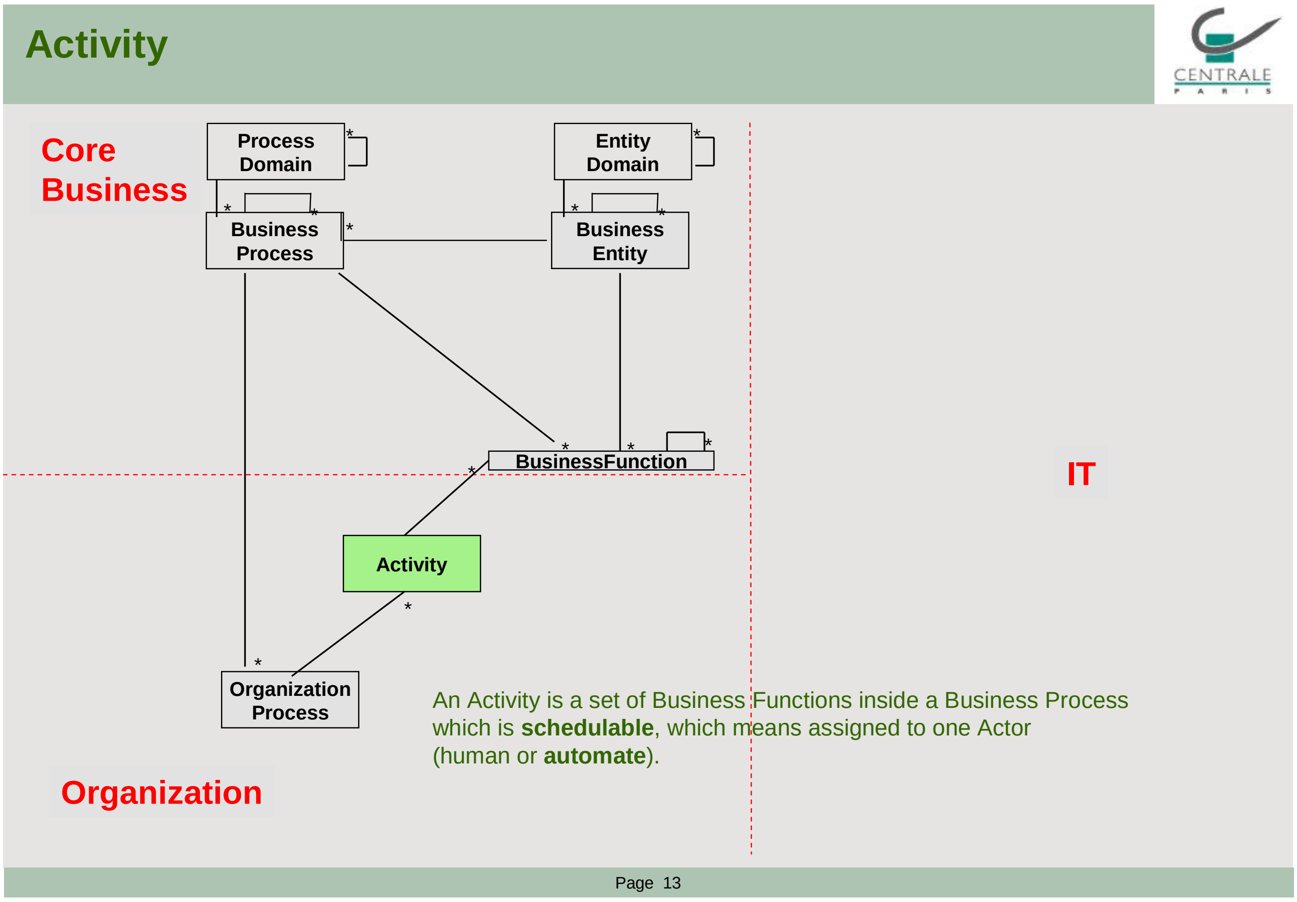
- Develop **software only** for **Business Process**
- Use a **parameterized** system to implement **Organization Processes**
    - Each Activity defined in the Business Process must own parameters like: Functional Domain, geographic scope, level of risk (like amount)
    - Each Organization Actor has Rights and Duties which define what the Actor can do and should do. For each Functional Domain define “yes or no”, “territory”, “risk level”
    - When a new Activity must start, the system check if the current Actor can do it: if yes, it goes on (several Activities for the same Task); if not, then the system looks for another Actor who can do it, and warns him in its “to do list”.

10 Activity

A Process can be executed by different Actors.
Generally there is not one Actor by Business Function, because some Business Functions are always executed together: for example, type a date, control date validity and correct the date if necessary, are 3 functions, but there is no chance that they are distributed to different Actors.
This is why Business Functions are grouped by **Activity**, with a different granularity.
An Activity is a set of Functions inside an Organization Process which is **schedulable**, which means assigned to one Organization Actor (human or automate).
Activities can be:

- totally manual (example: check a signature): **manual** Activity
- totally automatic (example: produce Payroll statements): **automatic** Activity
- human but helped by IT (example: any transaction done on a keyboard): **transactional** Activity; the workstation Desktop allows to launch transactional Activities.

For each Activity, to help the Workflow engine to find next Actor, are defined some parameters like: Functional Domain, geographic scope, amount, …



Activity

An Activity is a set of Business Functions inside a Business Process which is **schedulable**, which means assigned to one Actor (human or **automate**).

Page 13

At execution time, Organization Actors execute the Activities.

Remark: a **Task** is the set of Functions really executed at the same time by the same Actor.

- A Task may include one Activity.
- A Task may include several successive Activities of the same Organization Process if the Actor has convenient Rights and Duties.
- A Task may include part of an Activity, if the Actor decides to suspend the Activity for example because:
  - the Actor has not all the information, and prefers to suspend the Process to come back later, or

- he wants another Actor to interfere on the Process

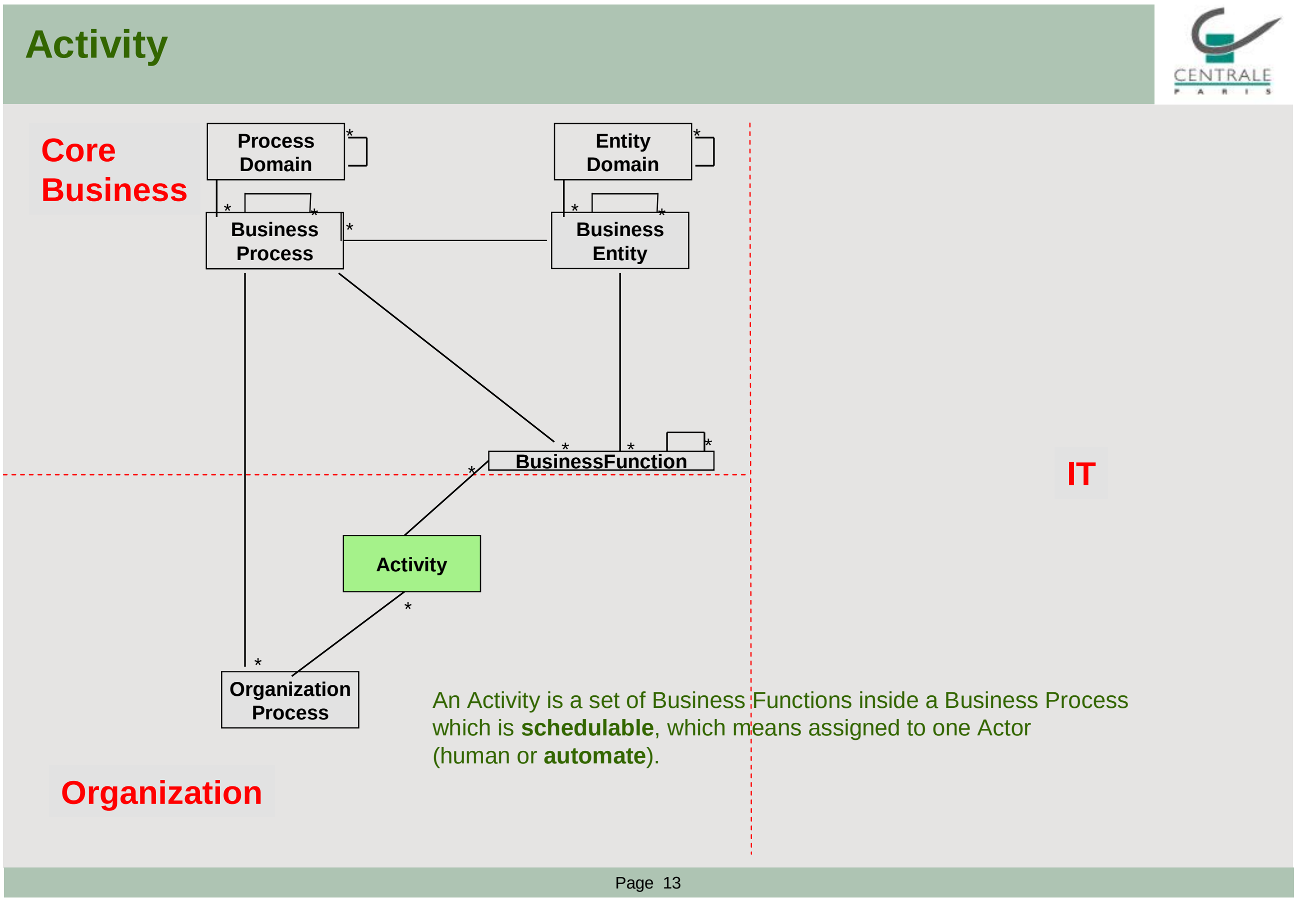
When you design a new System, do not think “Task”, just think “Activity”. Activities will be assigned to Actors thanks to Right and Duty rules defined by the Organization System Administrators. The Tasks will only be the result of assignment a posteriori.

11 Organization Function

To define with accuracy an Organization Process, Business Functions are not sufficient: we must add Organization Functions like “am I Authorized”, or “send a message to the manager”, or “ask someone to check” …

To help identify Organization Functions from Business Functions, you can use an easy rule: Organization Functions do not exist if the Business Process is executed in one task by the same Actor with full rights.

It gives a good model to improve efficiency of an Organization System: “Increase Productivity” means decrease number of Activities and decrease number of Business Functions.



Activity

Core Business

Process Domain

Entity Domain

Business Process

Business Entity

BusinessFunction

IT

Activity

Organization Process

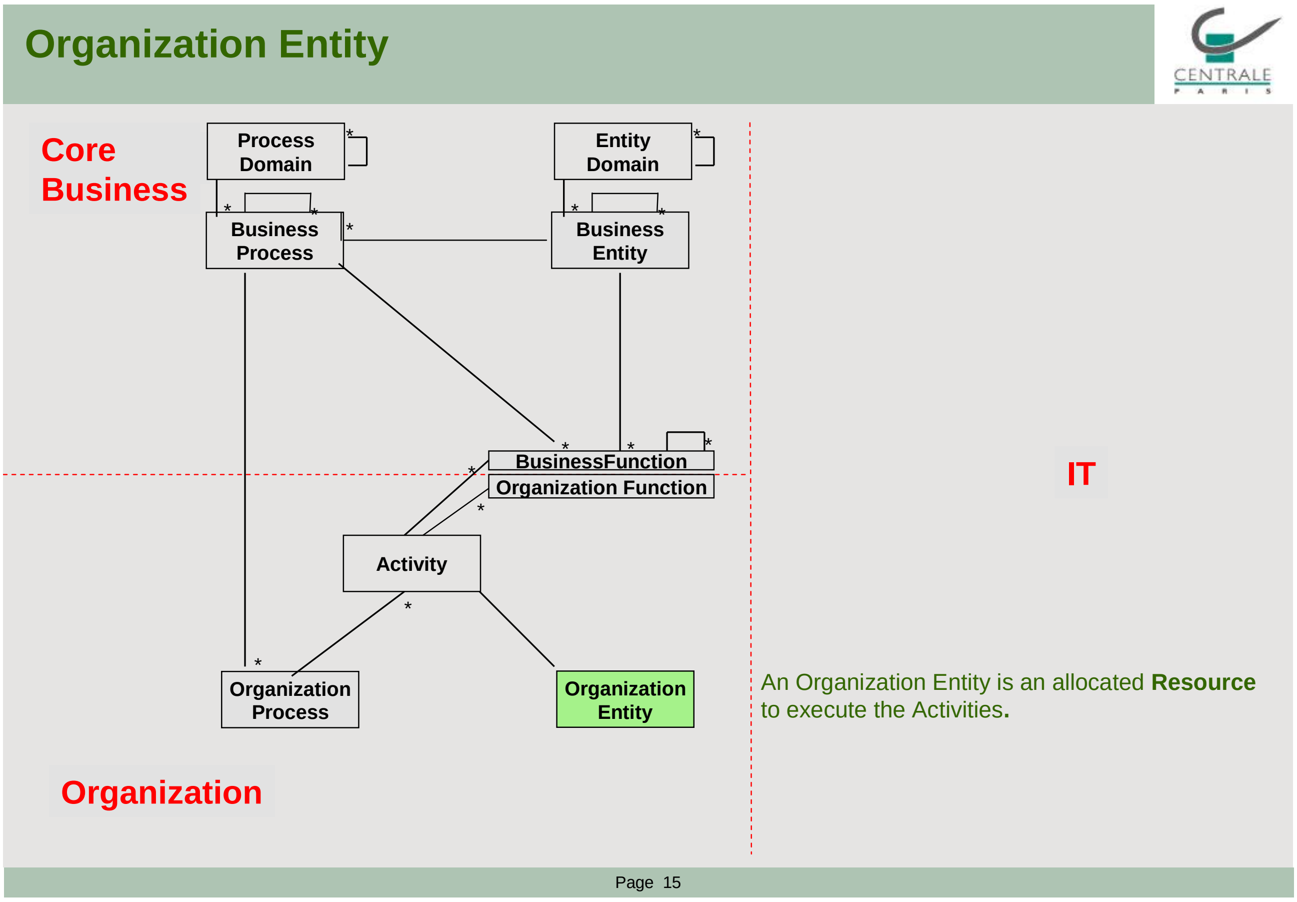
An Activity is a set of Business Functions inside a Business Process which is **schedulable**, which means assigned to one Actor (human or **automate**).

Organization

Page 13

12 Organization Entity

Business describes **what** the company must do.
Organization describes **how** it internally organizes itself to operate Business activity.
Organization Entities define all resources used by the company: Organization Actors, Organization Units, Rights and Duties, location, building…



Organization Entity

An Organization Entity is an allocated **Resource** to execute the Activities**.**

Page 15

12.1 Organization, Organization Unit and Position

Organization can be described not only for the company but also for partners when the Information system requires to know it.
An **Organization Unit** is a node of a hierarchical structure.
An Organization Unit can be the father of several Organization Units.
An Organization Unit can own Positions which is the smallest Organization Unit.
The Organization Unit describes the category of the Organization Unit (department, division, branch, …), the position for the leader of the Organization Unit, the parent Organization Unit, the sub Organization Unit (s), …
An **Organization** is a hierarchy of Organization Units which all depend from the same Organization Unit.
Example:

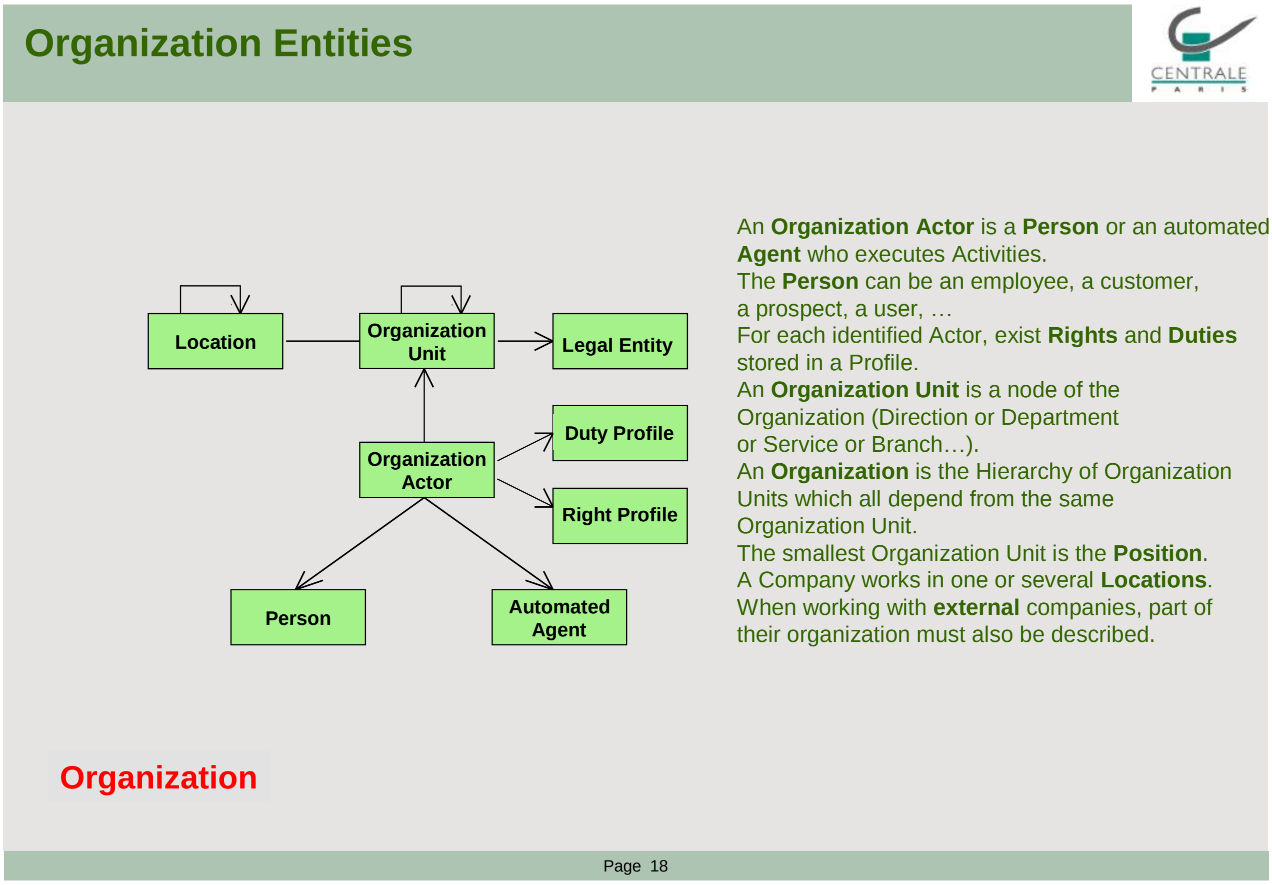
- a region composed of the regional headquarters plus regional branches is an organization,
- the whole company composed of the different regions is also an organization

A Position is the smallest Organization Unit.
Example: “director of the Branch X”, or “assistant of the director of the Branch X”

12.2 Actor

An Actor is a Person or an Automate.
A **Business Actor** is involved in the Business System: Customer, Prospect, Distributor, Provider, …
An **Organization Actor** is a Person or an Automate who executes Activities.
An Organization Actor occupies a Position.
All Organization Actors who use the IT System are IT Users.
As majority of Organization Actors are also IT Users, the 2 concepts are often merged.
Outsourced Activities or Partner Organization include Organization Actors if they use the same IT System
For each identified Actor exist

- **Rights** which define what he **can** do: used to control **responsibilities**
- **Duties** which define what he **must** do: used to share **workload**



Organization Entities

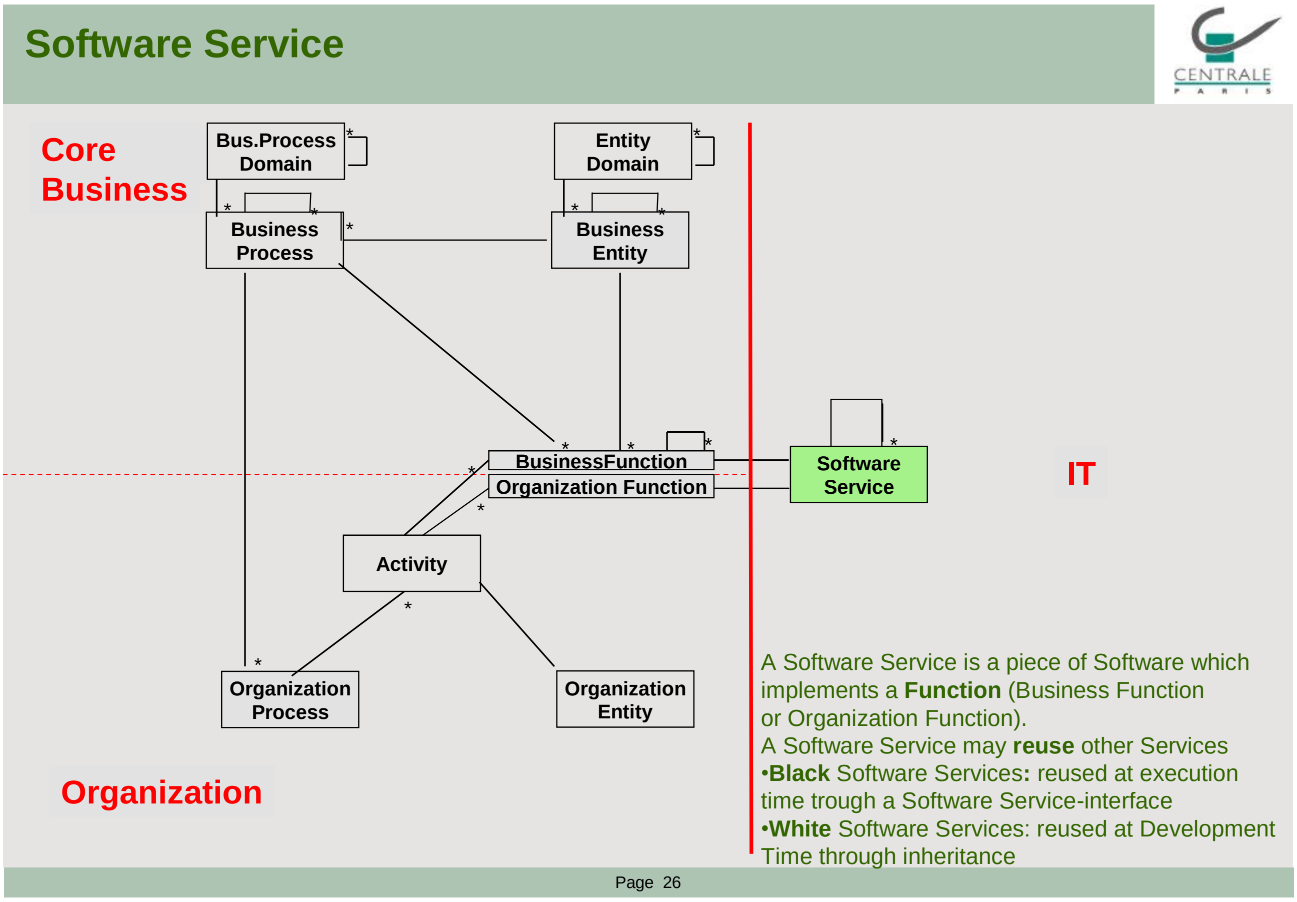
An **Organization Actor** is a **Person** or an automated **Agent** who executes Activities.
The **Person** can be an employee, a customer, a prospect, a user, …
For each identified Actor, exist **Rights** and **Duties** stored in a Profile.
An **Organization Unit** is a node of the Organization (Direction or Department or Service or Branch…).
An **Organization** is the Hierarchy of Organization Units which all depend from the same Organization Unit.
The smallest Organization Unit is the **Position**.
A Company works in one or several **Locations**.
When working with **external** companies, part of their organization must also be described.

Organization

Page 18

13 Software Service

A Software Service is a callable piece of software defined by an **Interface** (what the caller must know) and an **implementation** (hidden part which does the work).
When a Function is computerized, it is implemented as a Software Service.
Some Software Services only manage access to data they are called **Data Service**. As Data should only be accessible through Software Services **we do not make the distinction between sharing Services and sharing Data.**
A Software Service may call **other** Software Services which embed part of the complexity.
If **common** Functions (Business and Organization) have been identified, then deducted Software Services will be shared, and called **reusable Software Services**.
Business Process logic must be implemented.
Organization Process Logic must also be implemented.
It is recommended to split these 2 logics to limit change of software when Organization changes.
Software Services can be implemented by **Business Rule Engine** to accelerate the Rule updating process and to externalize the Rule modification (algorithm and/or parameters).
To accelerate the Organization Process modifications, use a **Workflow Engine**.



Software Service

A Software Service is a piece of Software which implements a **Function** (Business Function or Organization Function).
A Software Service may **reuse** other Services
•**Black** Software Services**:** reused at execution time trough a Software Service-interface
•**White** Software Services: reused at Development Time through inheritance

Page 26

14 Class

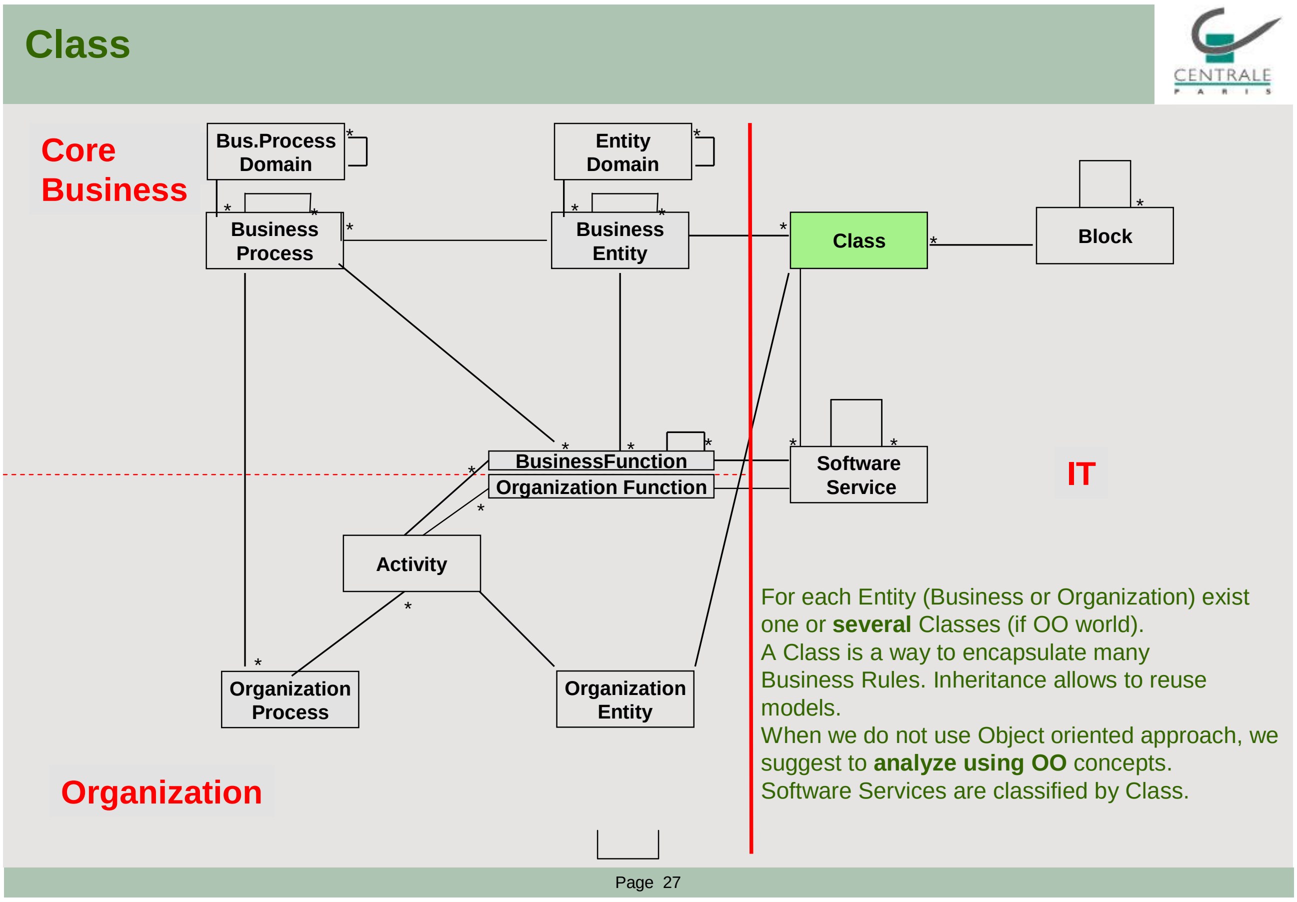
The Software Services are grouped into Modules or Classes.

A **Module** is a logical list of Software Services; some call it by other words like “library”. They do not include Data description.

In the Object Oriented world, a **Class** is the logical definition of Objects sharing common Attributes (or Data) and Methods (or Software Services).

An easy way to define Classes is to start creating a Class for each Business Entity. But as we will explain later on, the translation of Business Entities into Classes can be a little trickier.

We suggest designing a System using Object Oriented Approach even if the implementation is done through classical technologies. This is why we will only use the concept Class in our White Papers.



Class

For each Entity (Business or Organization) exist one or **several** Classes (if OO world).

A Class is a way to encapsulate many Business Rules. Inheritance allows to reuse models.

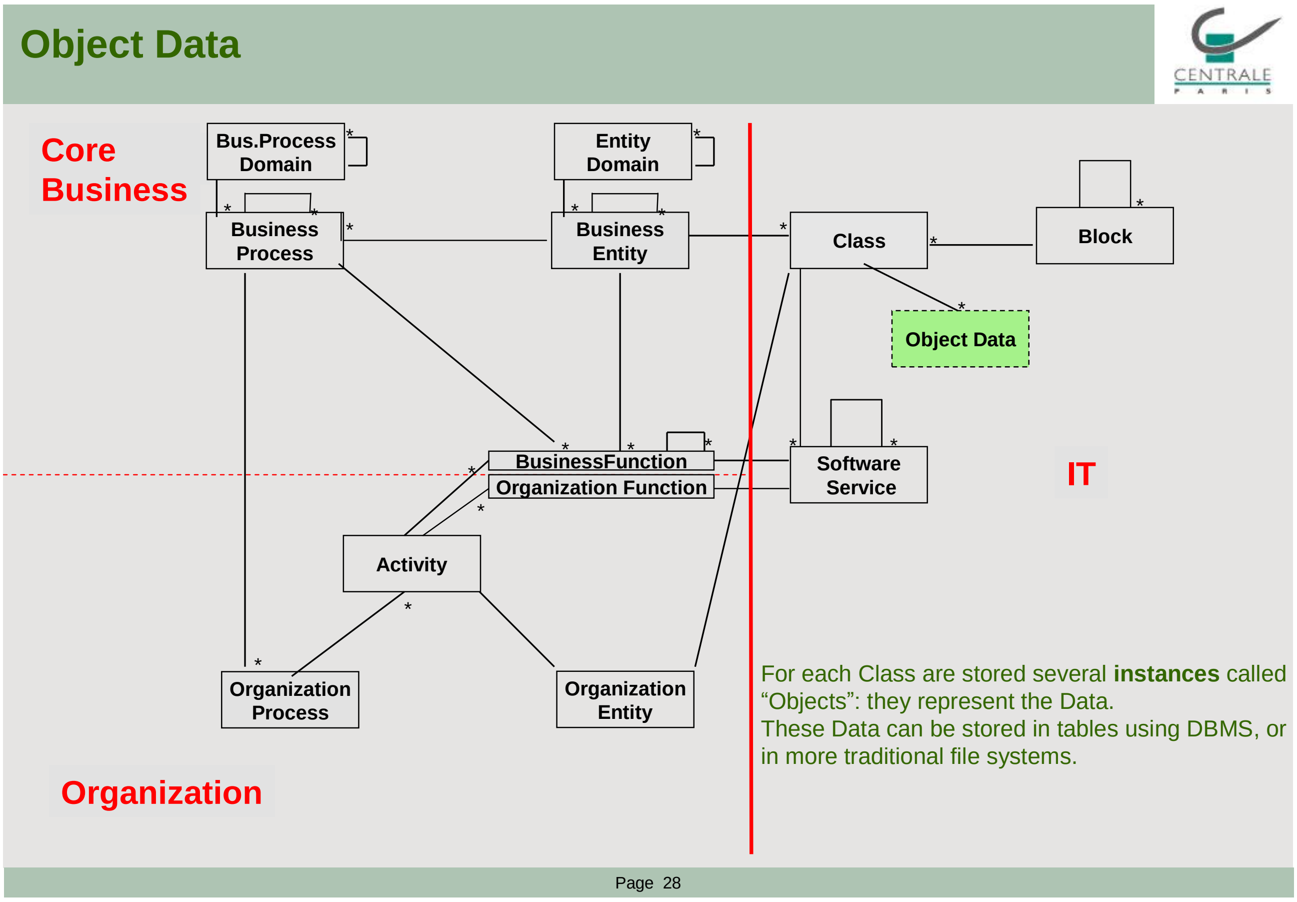
When we do not use Object oriented approach, we suggest to **analyze using OO** concepts.

Software Services are classified by Class.

Page 27

15 Object Data

For each Class exist several instances called Objects.
Example: for the Class “Person”, exist several Objects “Mr Smith”, “Miss Stevenson” …
The Objects are stored using different tools: classical files, relational data bases or Object Oriented data base.
First common Software Services to be implemented are Data Services to Objects. They protect software from data model evolutions.
Who is responsible for data update is described in the organization System (with Rights and Duties).
Localization of Data on servers and replication rules are described in the Operation System.



Object Data

For each Class are stored several **instances** called “Objects”: they represent the Data.
These Data can be stored in tables using DBMS, or in more traditional file systems.

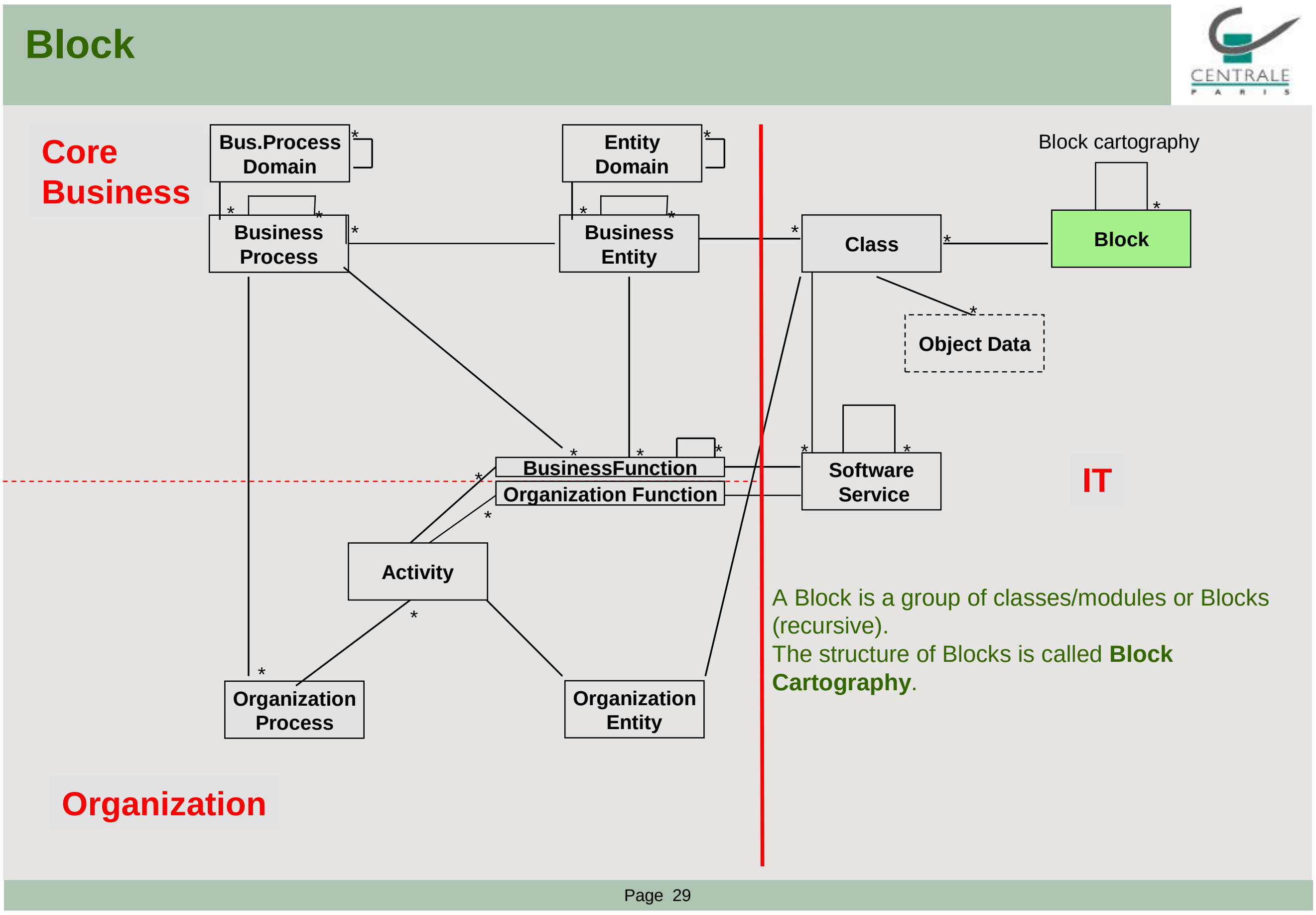
Page 28

16 Block

It is difficult to manage thousands or Modules or Classes. A Block is a set of Blocks or Classes.
As for any complex system, like a Town, System simplification comes from hierarchical decomposition into independent **Blocks**.
Ex: Accounting Block, CRM Block, Reference data Block, …
The structure of Blocks + their interfaces is called "**Block Cartography**".
It is also possible to "type" the Blocks by level as it is done in some organization (like Zone, Area, …).
Exchanges between Blocks use well defined **Interfaces**, accessible through different technologies (EAI, middleware, …). The objective is to build a Block hierarchy which minimizes exchanges between Blocks.

But updating each Block remains a difficult task which must follow the n steps internal software modification process.
To simplify updating of each block :

- **Decrease** the quantity of private software inside each Block by reusing available **components**
- **Externalize** parameters and Business Rules which often change so that updates are directly done by Professionals and not IT people.



Block

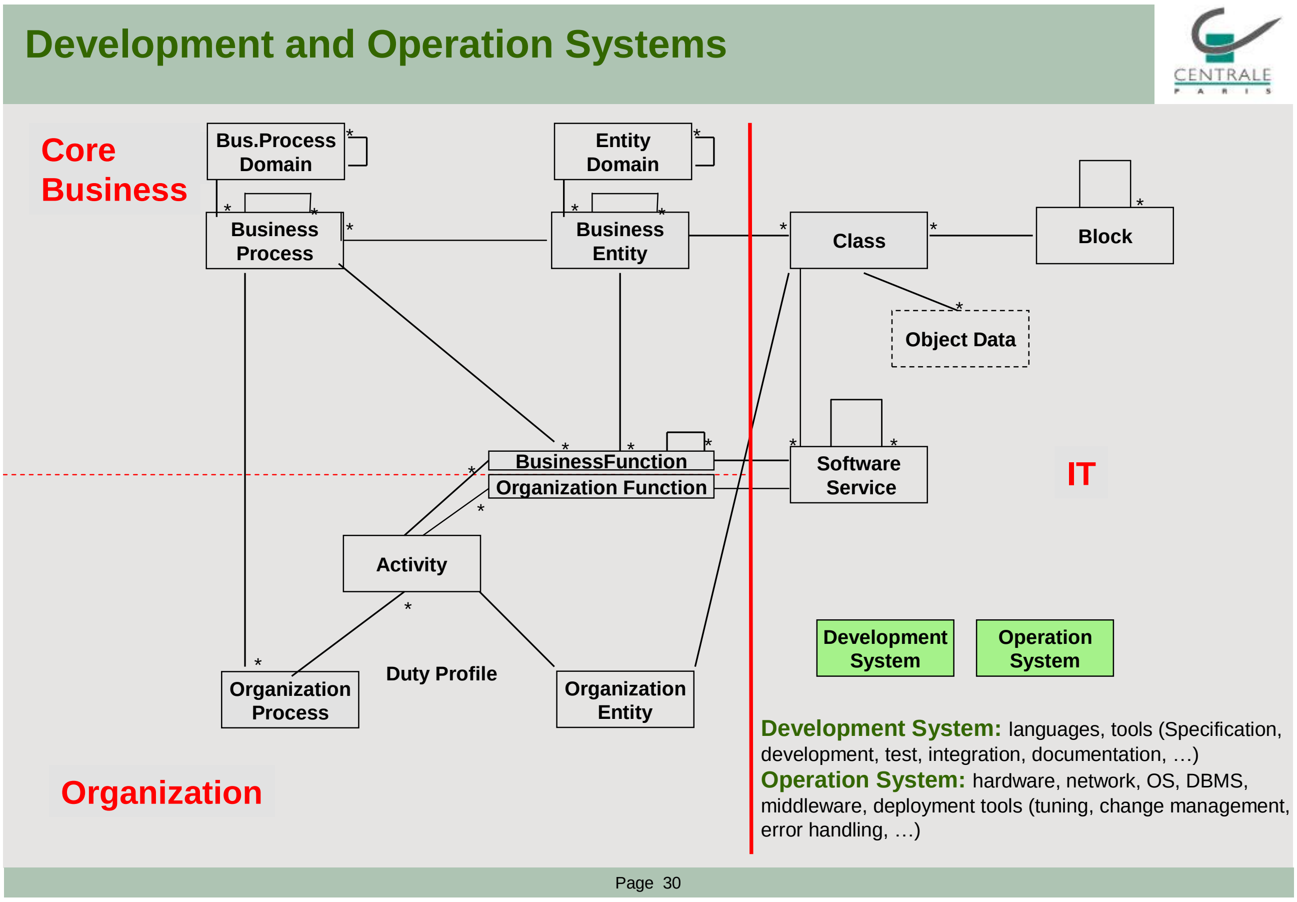
A Block is a group of classes/modules or Blocks (recursive).
The structure of Blocks is called **Block Cartography**.

Page 29

17 Development and Operation Environments

Development Environment is the set of tools offered to people who participate in the development processes.

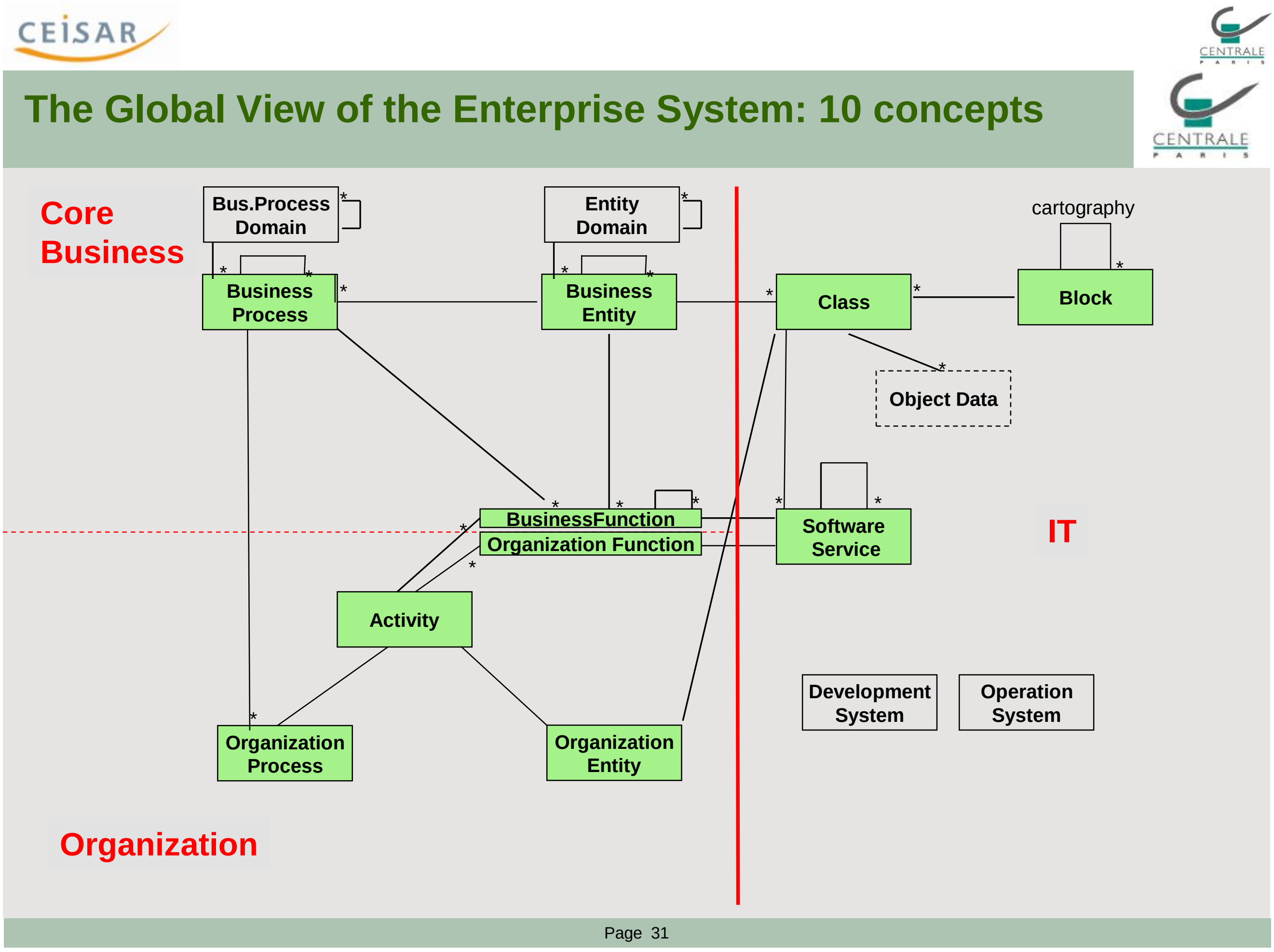
Operation Environment is the set of hardware, tools, network who participate in the operation processes.



Development and Operation Systems

Development System: languages, tools (Specification, development, test, integration, documentation, …)
Operation System: hardware, network, OS, DBMS, middleware, deployment tools (tuning, change management, error handling, …)

Page 30



The Global View of the Enterprise System: 10 concepts

Core Business

Bus.Process Domain

Entity Domain

cartography

Business Process

Business Entity

Class

Block

Object Data

BusinessFunction

Organization Function

Software Service

IT

Activity

Development System

Operation System

Organization Process

Organization Entity

Organization

Page 31

18 Component

A Component is a piece of software with clean Interfaces which allow different softwares to make use of it. The talent of the Software Developer is to design Components useful for many calling software: this is **reusability**.

A Component can have different granularities:

- A reusable **Software Service** can be a Component
- A **Class** which includes reusable Software Services can also be a Component
- A **Block** which includes several classes or Blocks can also be a Component.

When a Component is a Software Service we define Black Software Services and White Services.

Black Services are reused at execution time by a direct “call” or via a middleware. A Black service has an Interface and an Implementation (“Black” comes from “Black Box”: you just have to know interface to use it: implementation is hidden).

White Services are reused at development time by inheritance or typing. A Class from which we inherit to create a new Class is also a reused Component. But it is reused at development time by mechanisms like Inheritance or Typing, and cannot be solved by a middleware: we call it “**White Component**” because the Developer must understand its content to make good usage of it. These White components explain why there is no exact parallelism between developed software and executed software.

To compute the **reuse rate** of a piece of software, consider both Black components and White components.

Interface exists at each level:

- **Software Service Interface** is the interface of the IT Service
- **Class Interface** is the set of IT Service interfaces for **Public** Services offered by the Class
- **Block Interface** is the set of IT Service Interfaces for **Public** Services offered by the Block.
- We use “**Public**” to make a distinction with **Private** Software Services which are not known outside the Class/Module or Block.

GAINS	Block Cartography	Reuse Classes and Software Services
Visibility and understanding of the IS	+++	+
Time/cost to integrate a new application	+++	+
Time/cost to modify an existing application	+	+++
User Interface consistency		+++
Business Processes flexibility	+	+++
Quality of informations (no duplication, data consistency)	++	++

19 What is Enterprise Architecture?

19.1 Enterprise Architecture = what is shared

Architecture has several definitions:

- An Architecture is the **structure** of the System: decomposition of a complex System into smaller components.
- An Architecture also describes **approach and tools** to reach a good structure

To obtain the best Architecture, it is not sufficient to "**componentize**" a System, it is much more efficient if Components are **reused** by several Systems: from Entity definitions or Business Process models, to Software Services or Development System.

This is the case in the Car industry: reusing parts to offer a new car model means:

- Easier design and decrease time to market
- More reliability
- Lower costs

Reuse rate becomes the key competitive advantage.

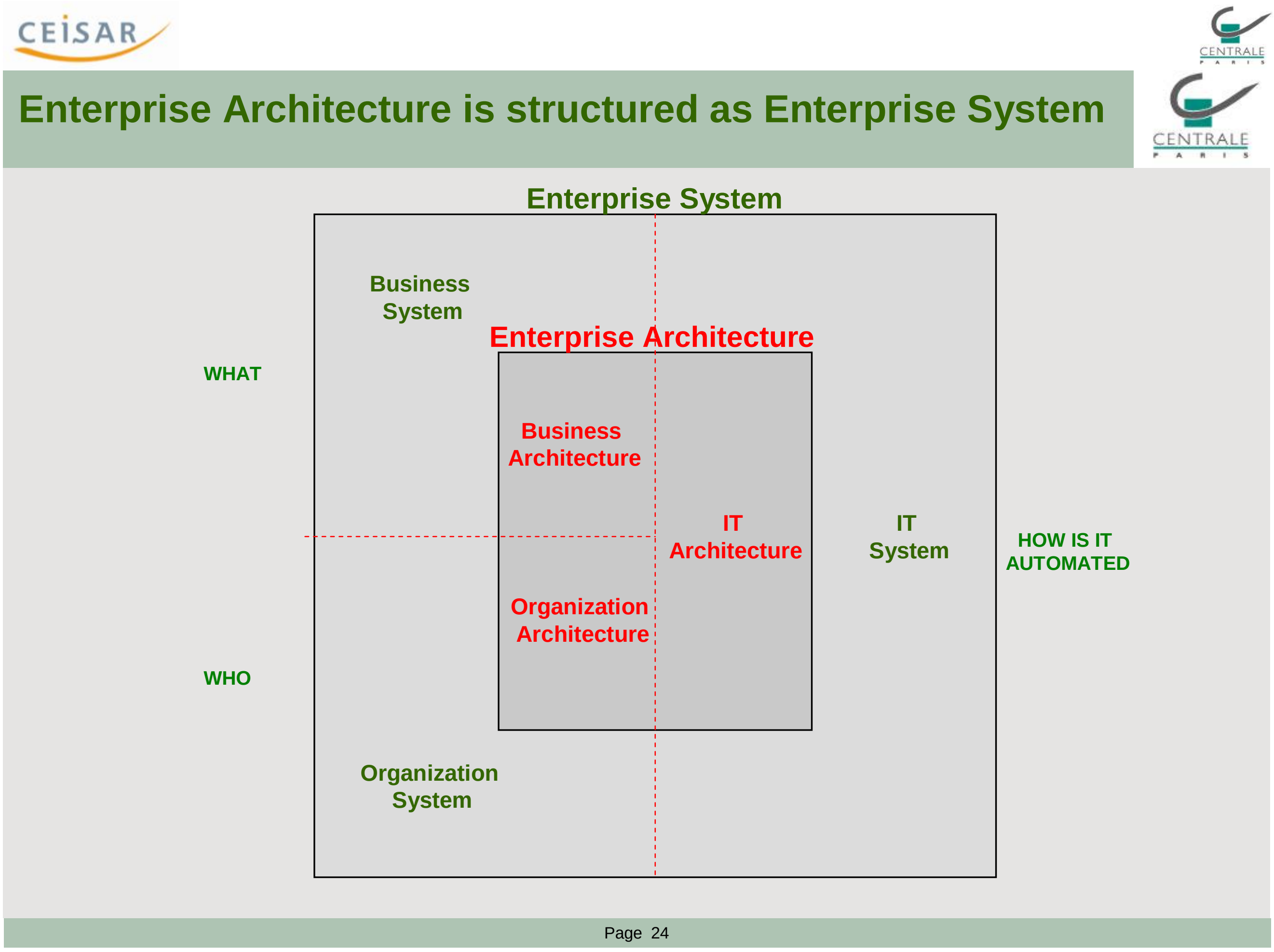
This is why CEISAR proposes a more ambitious definition of Architecture: "**Architecture is what can be shared**". By giving this definition, Architecture includes together "how to build an Architectured System with reused Components" and the deliverable "the System Architecture itself with high reuse rate".

Reusing same Components automatically creates the good structure.

Enterprise Architecture is **not limited to technical layers**; it includes many domains like:

- Shared Business Entity definitions
- Shared Process models
- Shared Functions
- Shared Data
- Shared Block cartography with shared Interfaces
- Shared Software Services and shared Classes
- Shared IS Governance
- Shared Approach
- Development Architecture : shared tools and resources to analyze business requirements and develop software
- Operation Architecture : common tools and resources to execute software

19.2 (Business + Organization + IT) Architectures



Enterprise Architecture is structured as Enterprise System

Enterprise System

Business System

Enterprise Architecture

WHAT

Business Architecture

IT Architecture

IT System

HOW IS IT AUTOMATED

Organization Architecture

WHO

Organization System

Page 24

Enterprise Architecture can be structured as is structured the Enterprise System : Business, Organization and IT Architectures.

Business Architecture includes:

- Shared Business Entities
- Shared Business Functions
- Shared Business Process Models
- Shared decomposition of Business Process and Entity Domains.

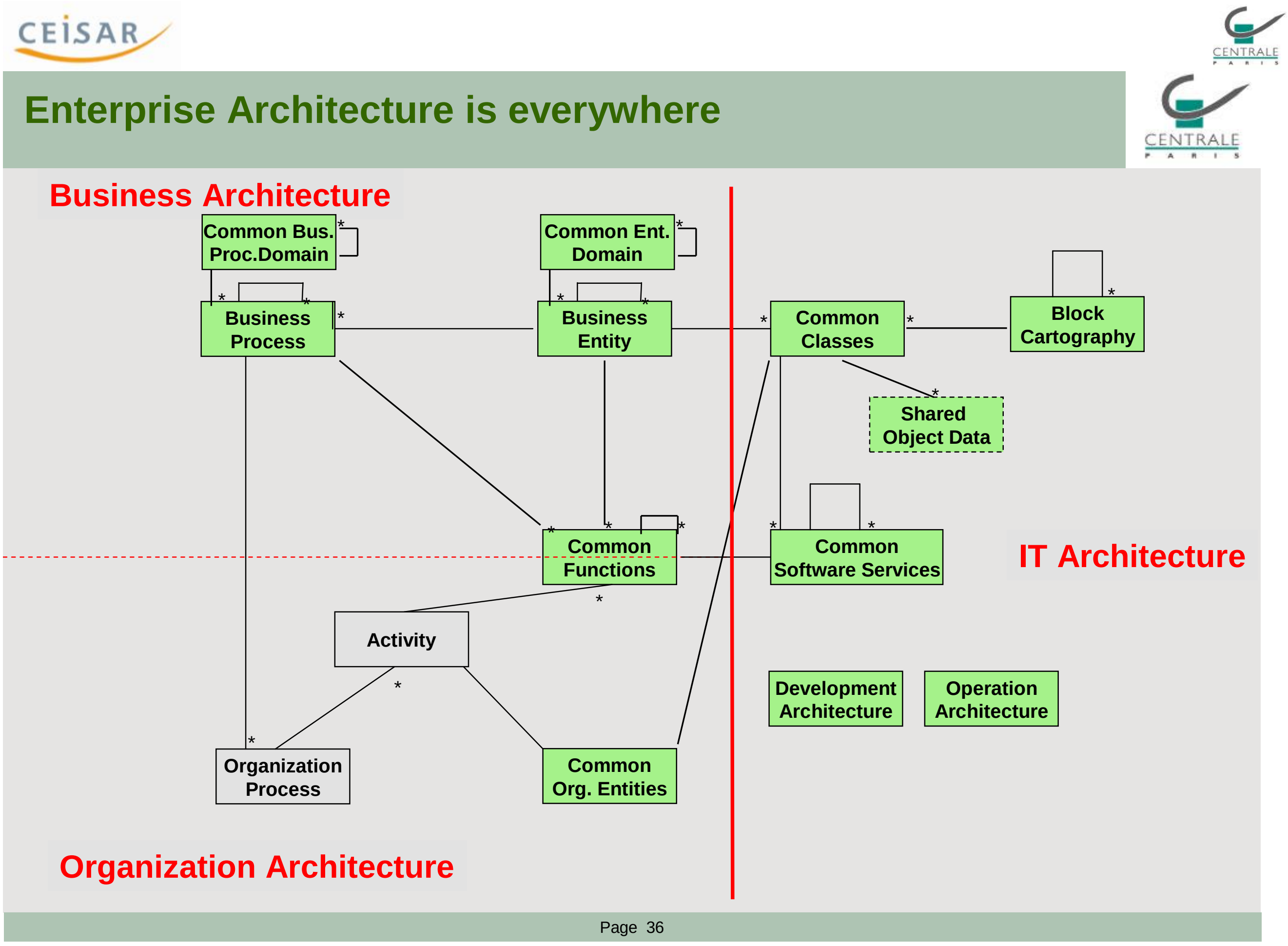
Organization architecture includes

- Shared Organization Entities: Organization Actors, Organization structure
- Shared Organization Functions like Security and Activity assignment
- Shared decomposition of Organization Process and Activity Domains.

IT Architecture includes

- Shared Block Cartography with shared Interfaces
- Shared Classes
- Shared Software Services
- Shared Data
- Development Architecture: languages and Development tools (Specification, development, test, integration, documentation, …)
- Operation Architecture: common tools and resources to execute software hardware, network, OS, DBMS, middleware, deployment tools

Shared Architecture Governance and Approach



Enterprise Architecture is everywhere

Business Architecture

Common Bus. Proc.Domain

Common Ent. Domain

Business Process

Business Entity

Common Classes

Block Cartography

Shared Object Data

Common Functions

Common Software Services

IT Architecture

Activity

Development Architecture

Operation Architecture

Organization Process

Common Org. Entities

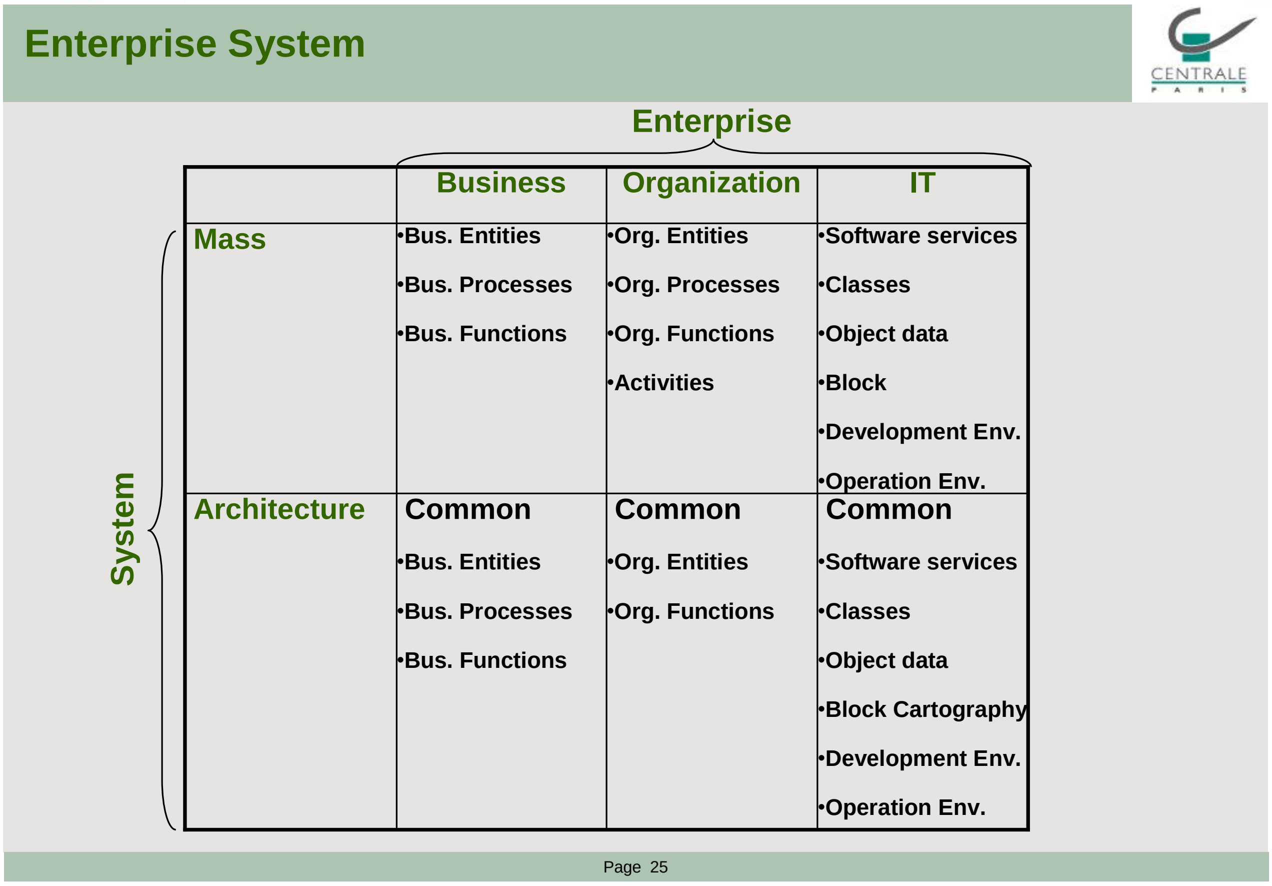
Organization Architecture

Page 36

To summarize, we define

- Enterprise System = Business System + Organization System + IT System
- Enterprise Architecture = Business Architecture + Organization Architecture + IT Architecture, defined as what is sharable.

(We preferred not to use “Information System” to avoid long discussions on what should or should no be included in its scope).



Enterprise System

Enterprise

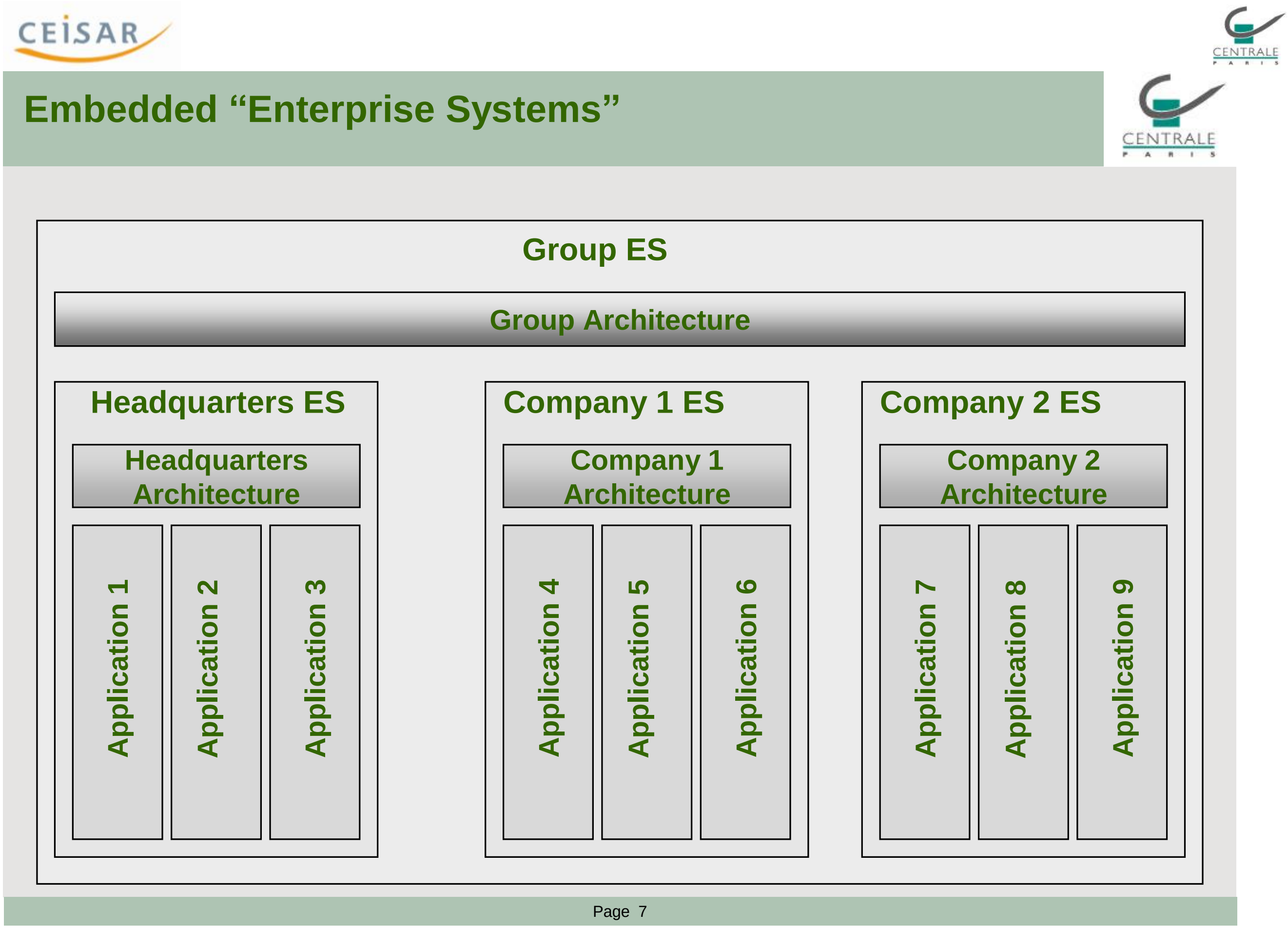
System

	Business	Organization	IT
Mass	•Bus. Entities •Bus. Processes •Bus. Functions	•Org. Entities •Org. Processes •Org. Functions •Activities	•Software services •Classes •Object data •Block •Development Env. •Operation Env.
Architecture	Common •Bus. Entities •Bus. Processes •Bus. Functions	Common •Org. Entities •Org. Functions	Common •Software services •Classes •Object data •Block Cartography •Development Env. •Operation Env.

Page 25

19.3 Embedded Architectures

Architecture follows same decomposition than Enterprises.
A Group may define a Group Architecture applied to all companies in the Group.
Each company may also define a Company Architecture which is based on Group Architecture, and **adds** sharable elements at its level.



Embedded “Enterprise Systems”

Group ES

Group Architecture

Headquarters ES

Headquarters Architecture

Application 1

Application 2

Application 3

Company 1 ES

Company 1 Architecture

Application 4

Application 5

Application 6

Company 2 ES

Company 2 Architecture

Application 7

Application 8

Application 9

Page 7

20 How Business Objectives are translated into the Enterprise System?

Business Objectives are split into 2 categories: Operational Objectives and Structural Objectives.

20.1 Operational Objectives are translated into Processes

Decompose these Objectives as a **hierarchy** whose leaves are **Processes** which indirectly benefit from Architecture.
Define the perimeter for each objective with the 4 dimensions (Line of Product, Customer Target, Territory, and Organization)
Define a verb for each objective and a quantified indicator.

Examples:

- Reduce **time to market** for new products from 4 months to 2 months means optimizing the Development Processes
    - Accelerate process “design a product”, which means
        - Formalize what is a product
        - Create a repository of Product components
        - Provide a tool to assemble products
        - Provide a tool to simulate product
    - Improve the Process “implement a product”
    - Facilitate the Process “train Users”
- Improve **productivity** by 20% means optimizing the main Processes of this unit
    - Part of the work must be done by external people
        - Offer Organization Processes to customers to directly execute some Business Processes

Some changes have consequences on Business System and/or Organization System.
For example, if an Enterprise chooses to work with a new distribution network:

- If same Products, same Customer Segments: then the Business System does not change
- If it is necessary to adapt Products and reach new Customer segments, then the Business System and the Organization System must change.

20.2 Structural Objectives are translated into Architecture

They cannot be decomposed into Processes, and are solved by Architecture
Examples:

- “More Group synergy” means more Architecture to share customers, products, organizations, …
- “Improve security level”, means a common security function
- “Improve employee capacity to execute many tasks”, means common UI and navigation functions
- “Simplify Enterprise System”, means limit number of Development Environments and Operation Environments

For each Objective, define which part of the Architecture must be built, like Security Components, Operation Architecture, …