

L'ingénierie des Systèmes : l'approche SysML

Alain FRAISSE

Marc Vernay

09 Octobre 2014 – Le Hive

Introduction :

- Alain Fraisse :

- Chef de projet du déploiement et définition de l'approche Système au sein de la Business unit « Partner ».

alain.fraisse@schneider-electric.com +33 6 87 80 24 04



- Marc Vernay :

- Responsable des équipes Systèmes dans la Line of Business « Distribution Terminale » au sein de la Business unit « Partner ».

Marc.vernay@schneider-electric.com +33 6 30 24 89 85



Sommaire :

- Introduction : De nouveaux Systèmes pour de nouveaux enjeux
 - Exemples
- Une démarche : l'ingénierie des systèmes
- Rappel des Processus majeurs – IEC 15288
- Le Processus technique et le positionnement de SysML
 - Le modèle de données
 - Positionnement de SysML et définition
- Méthodes et SysML
- Intérêt du langage SysML
- Déploiement et recommandations
- Conclusion et Exemples

Introduction : De nouveaux Systèmes pour de nouveaux enjeux

- Nous passons progressivement d'une réponse **Produit** à une réponse **de Solutions/Systèmes de Produits** face aux enjeux applicatifs :

Exemple pour la RT 2012 

« Etre capable de proposer des Solutions pour la maitrise de l'énergie »



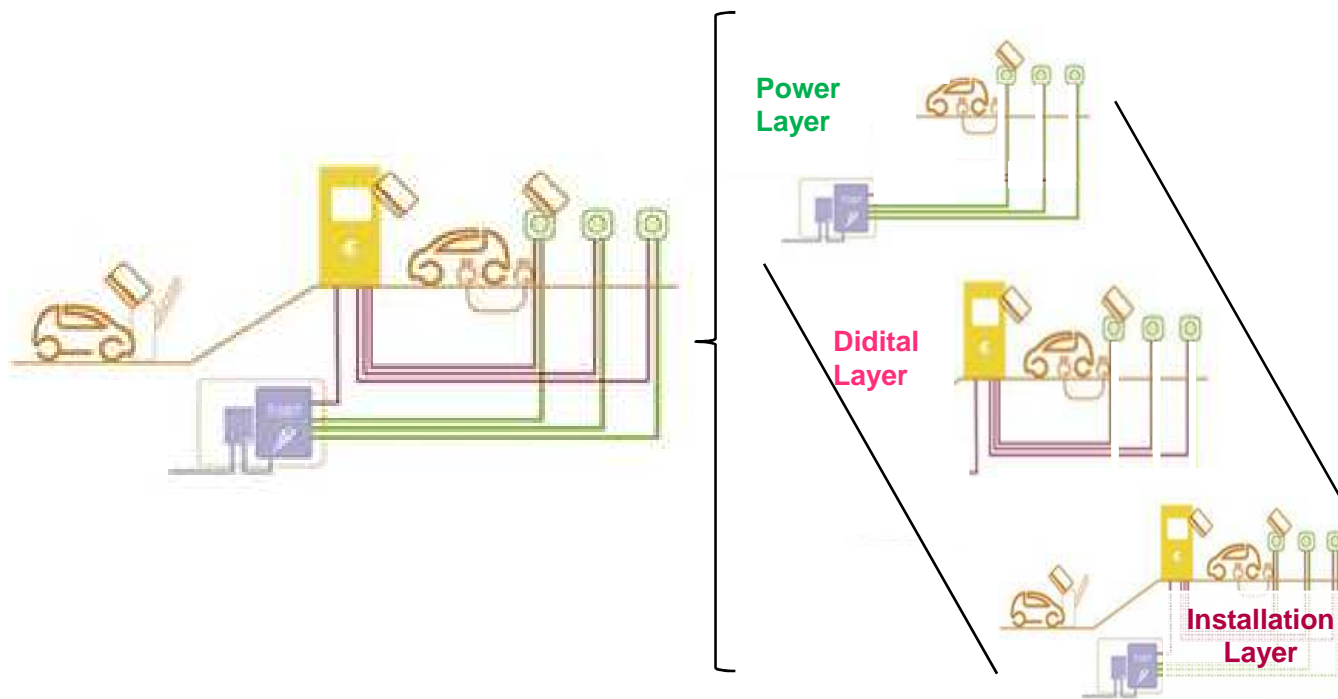
Des questions se posent pour le développement :

- Quels « use cases » ?
- Quelles fonctions systèmes ?
- Comment optimiser l'architecture ?

Introduction : De nouveaux Systèmes pour de nouveaux enjeux

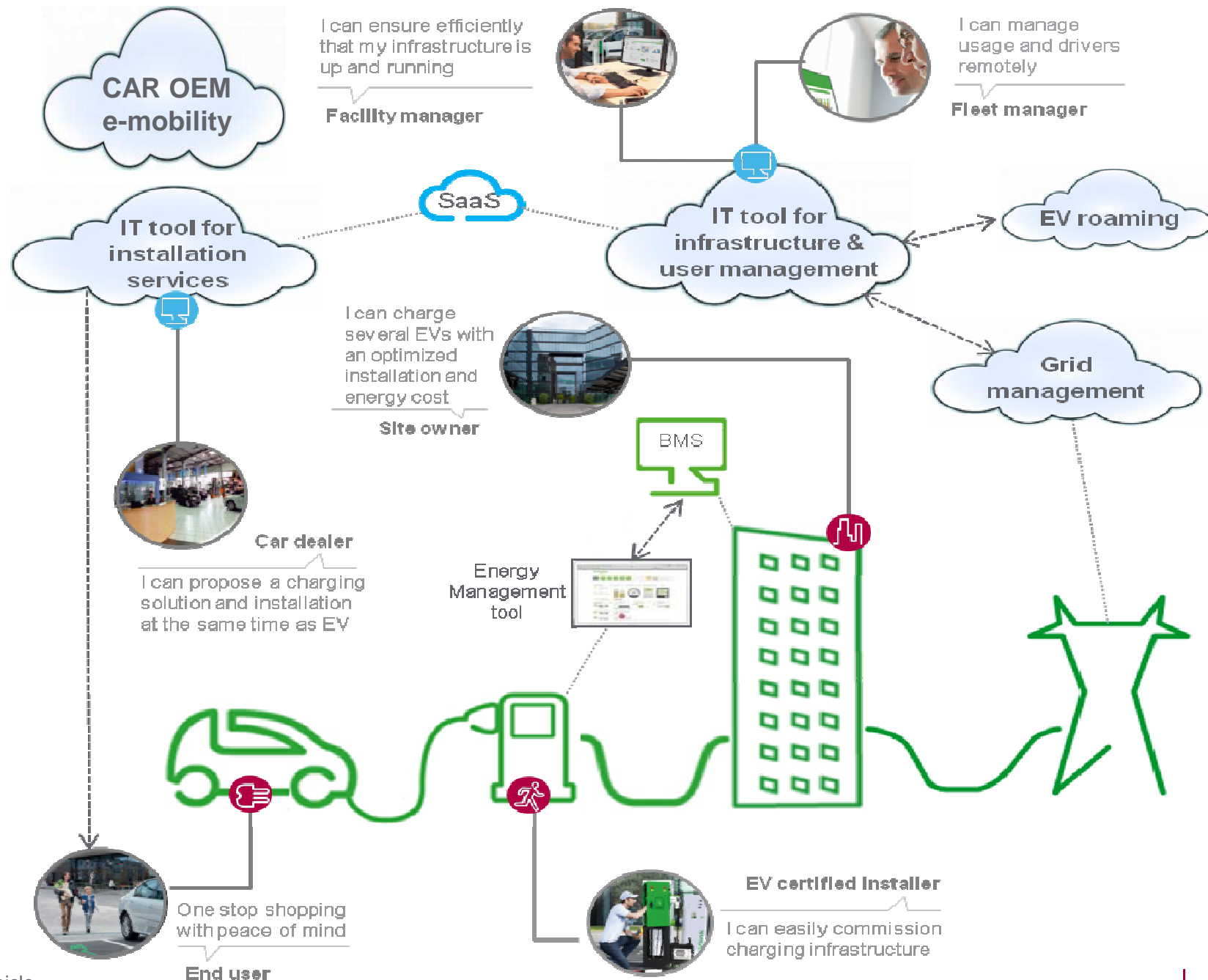
- De nouveaux systèmes émergent et doivent être connectés

Exemple : des solutions de bornes pour le **Véhicule Electrique** :
« Etre capable de proposer des Solutions d'alimentation connectés avec des systèmes existants »



Des questions se posent pour le développement :

- Quels « use cases » ?
- Quelles fonctions systèmes ?
- Comment optimiser les architectures imbriqués ?

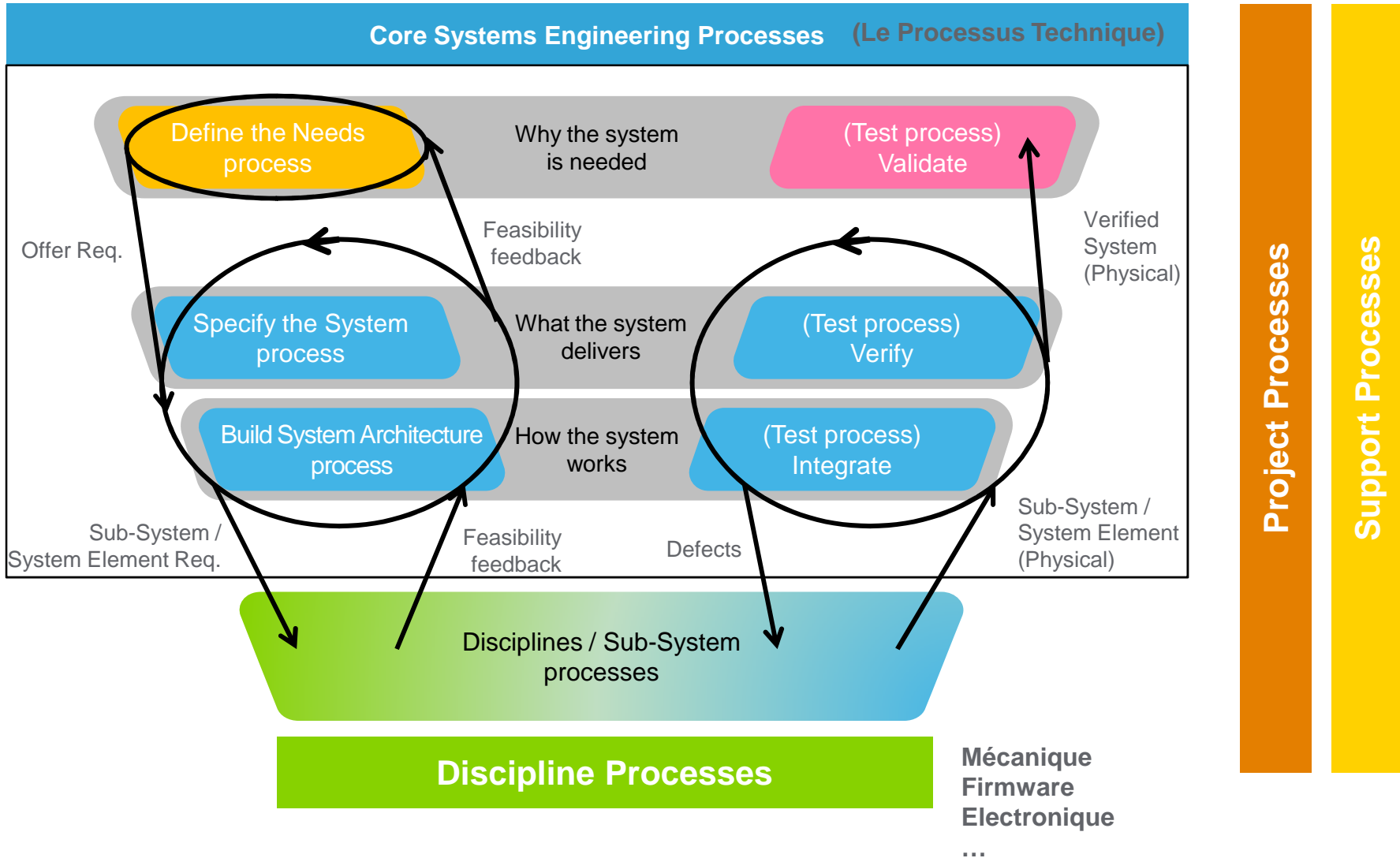


Une démarche : l'ingénierie des systèmes :

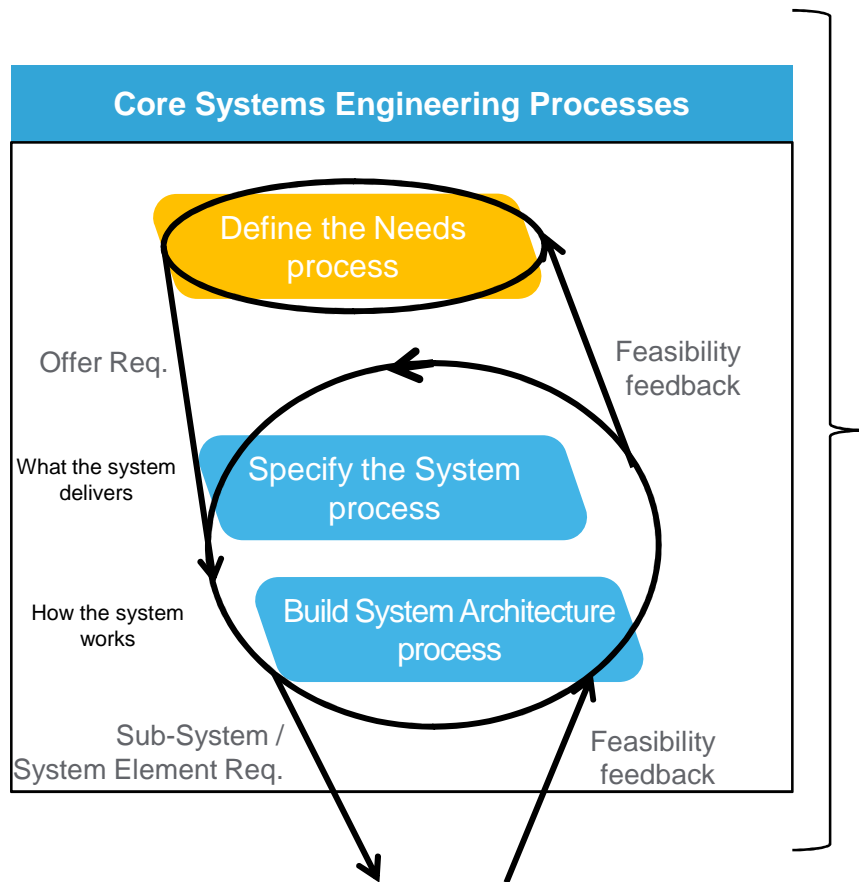
- L'ingénierie des Systèmes est une approche **interdisciplinaire** pour guider et gérer le développement de **systèmes Complexes** .
- Cette approche **collaborative** doit permettre **l'émergence de fonctionnalités** qui sont impossibles à obtenir en considérant chaque produit individuellement.



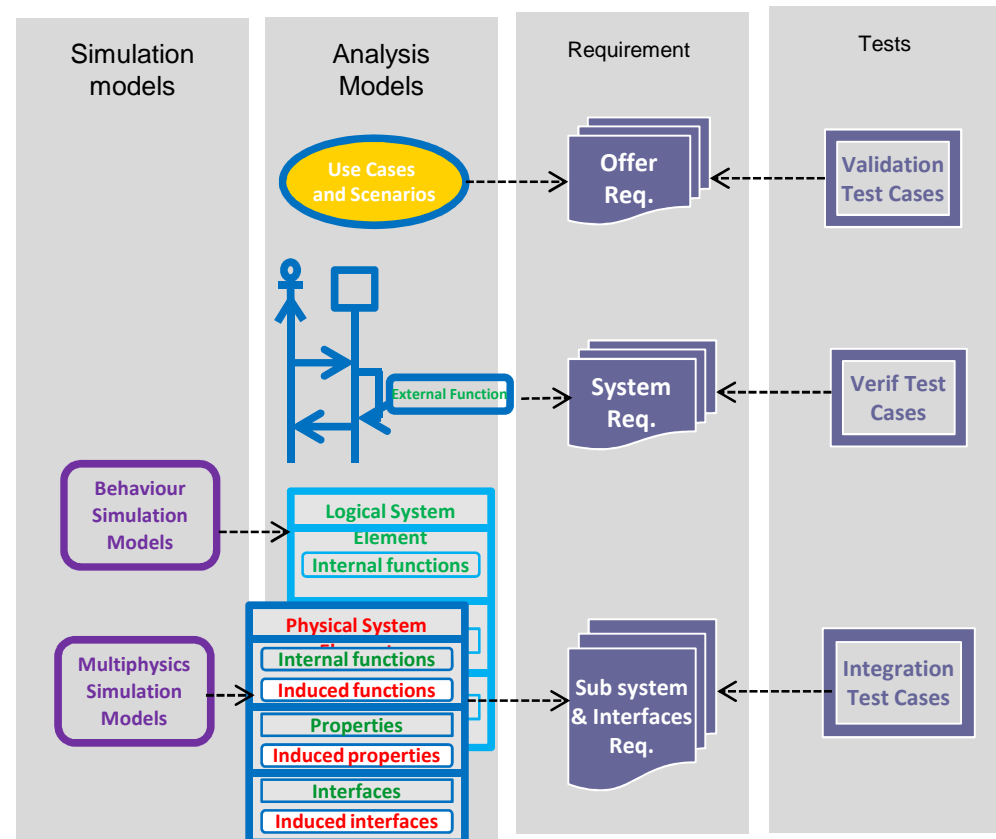
Rappel des Processus majeurs – IEC 15288



Le Processus Technique et le positionnement de SysML



Modèle de traçabilité Projet

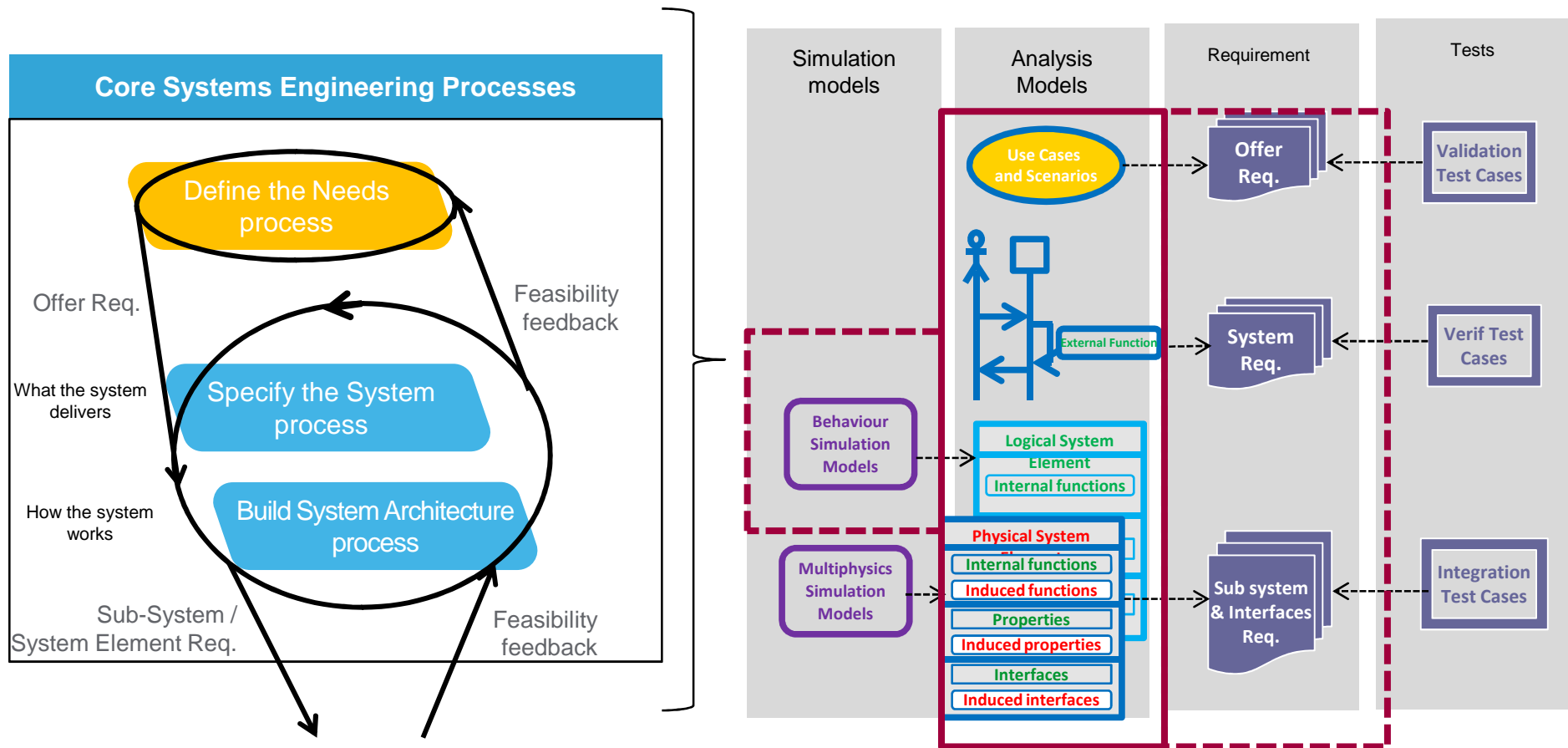


According to  EIRTS

Le Processus Technique et le positionnement de SysML

Modèle de traçabilité Projet

According to  EIRIS

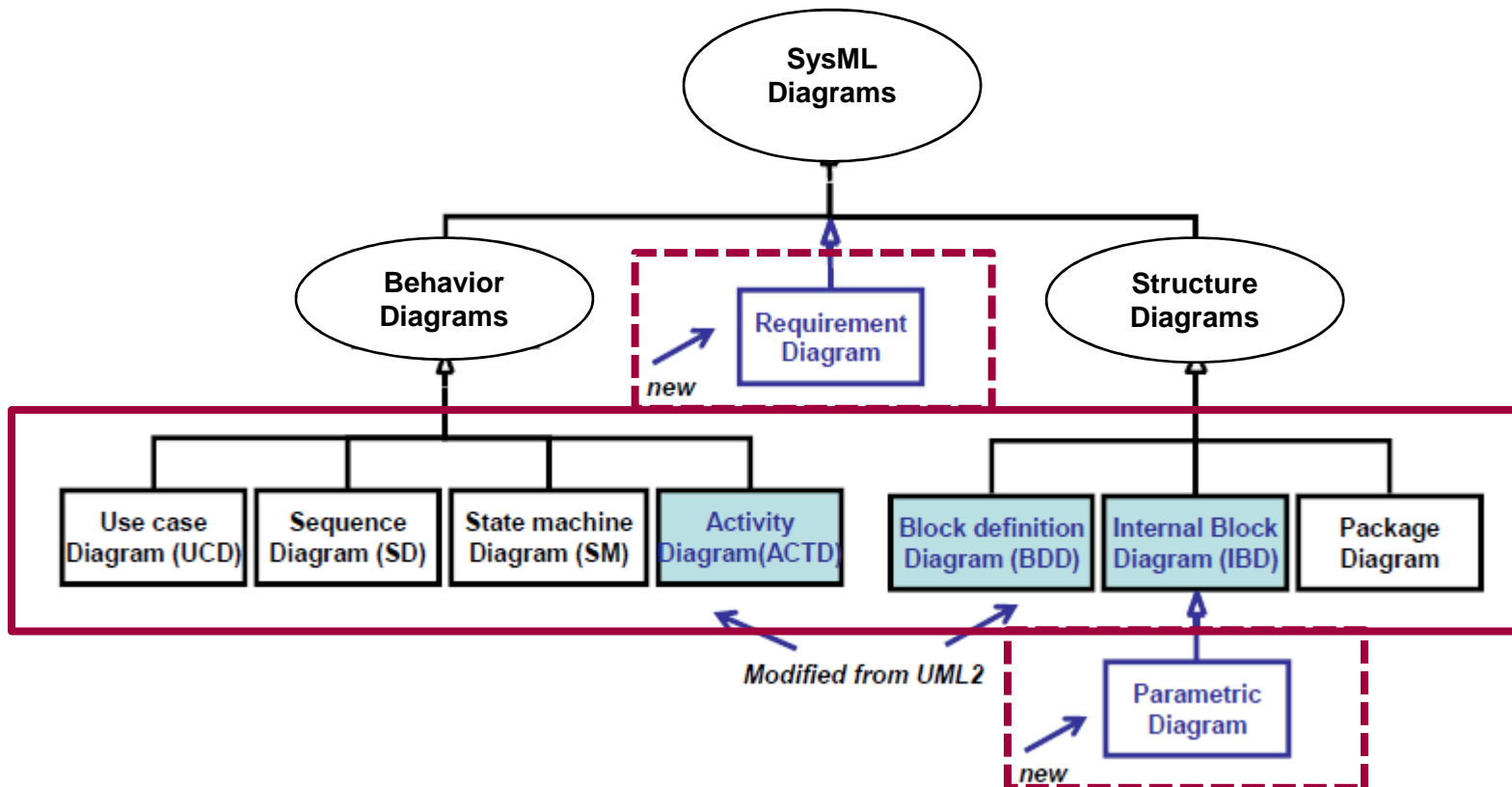


SysML :

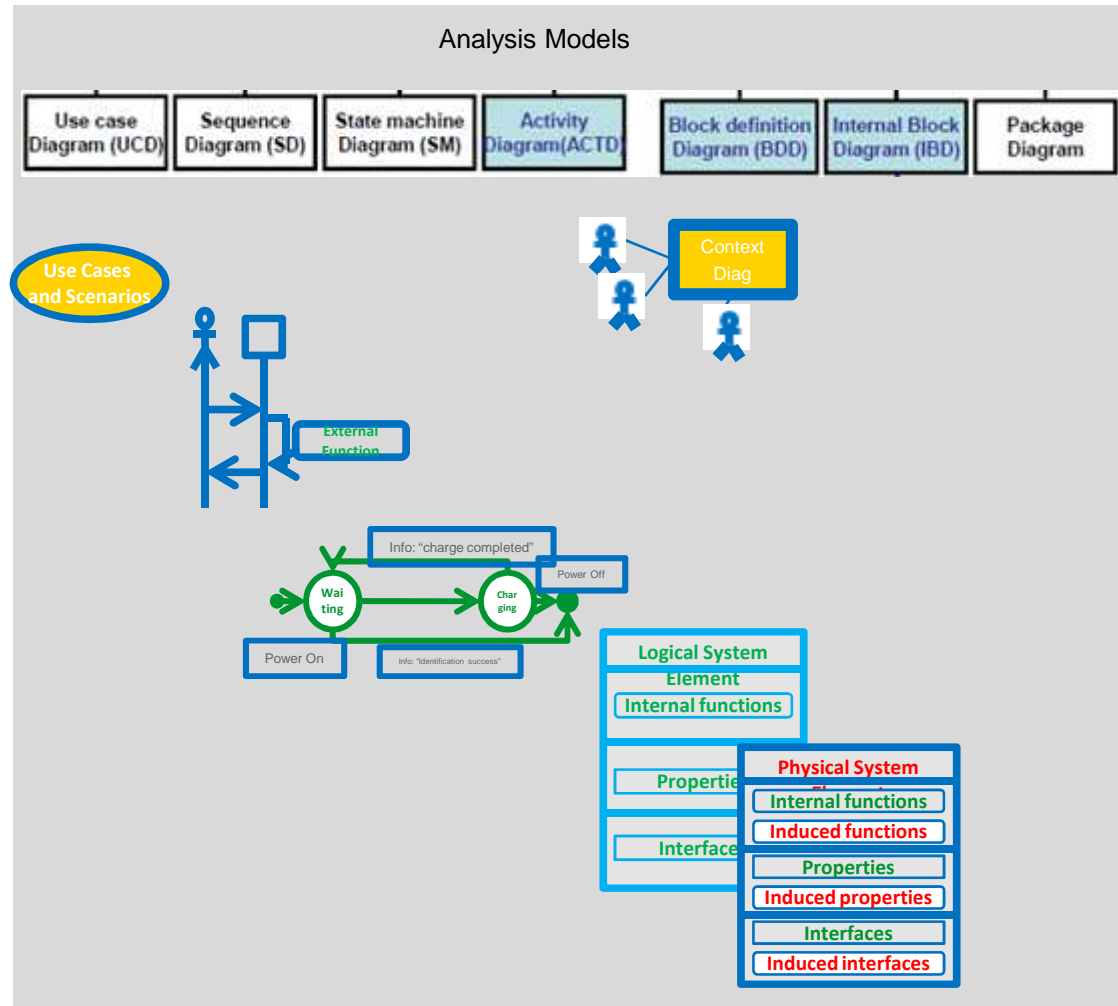
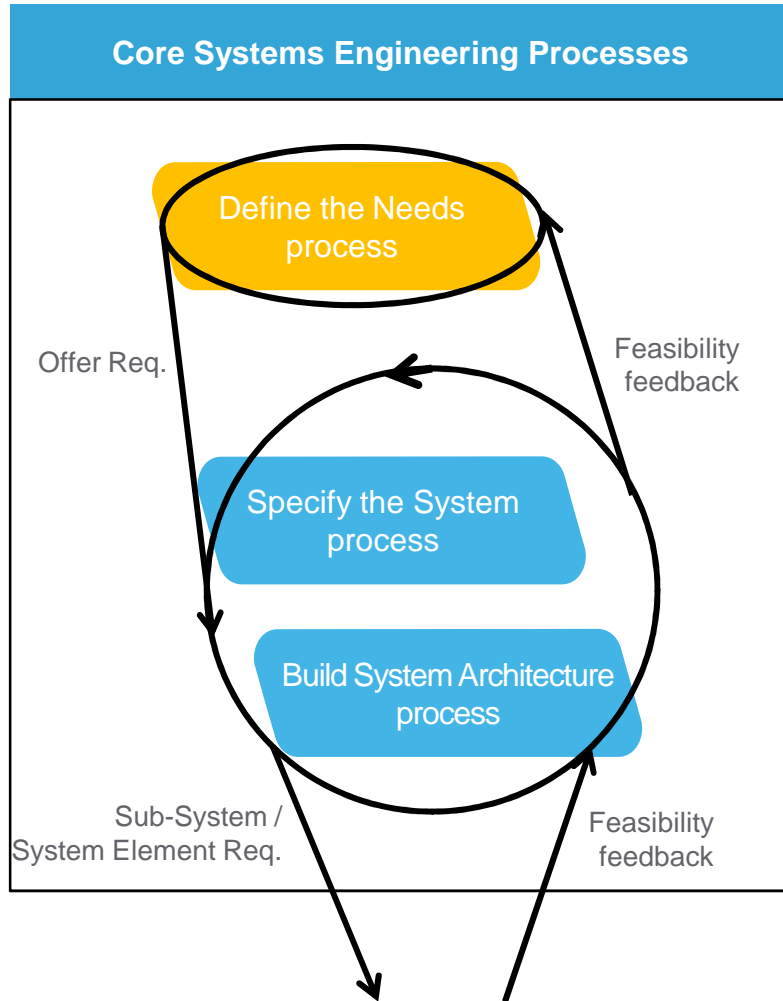
- Définition de SysML <http://www.omgsysml.org/> (OMG : Object Management Group)

The OMG systems Modeling Language (**OMG SysML™**) is a general-purpose **graphical modeling language for specifying, analyzing, designing, and verifying complex systems** that may include hardware, software, information, personnel, procedures, and facilities. In particular, **the language provides graphical representations with a semantic foundation** for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models.

SysML V 1.3 : OMG adoption june 2012



Methodes et SysML :



Cas d'utilisation Diagramme de séquence Machines d'états Architectures fonctionnelles
 Analyse fonctionnelle Diagramme de contexte Architectures Physiques

Intérêt du langage SysML :

- Les avantages :

- Avoir un **langage unifié** pour la communauté des ingénieurs systèmes dans l'entreprise et à l'extérieur de l'entreprise. 
- Aider à la construction d' **architectures modulaires** (Innovation).
- Permettre l'analyse du **besoin** et la construction formelle d'**architectures fonctionnelles et physiques (Technologiques)**.

- L'apport des outils associés :

- Réaliser des **bibliothèques** réutilisables
- Formaliser et **maintenir les évolutions** tout au long du cycle de vie du Système.
- Garantir une **traçabilité** du besoin jusqu'à l'architecture du système dans un environnement descriptif cohérent.

- Les limites :

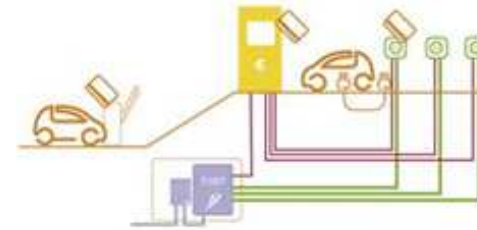
- Le langage peut être parfois difficilement compréhensible par des fonctions périphériques (Marketing) –

Déploiement et Recommandations :

- **Déploiement chez Schneider de SysML :**
 - Le déploiement se réalise progressivement depuis 3 ans sur des projets ciblés.
 - Pour cela il est nécessaire de **construire une proposition claire articulant le processus , les méthodes , et le langage** (illustré par le slide 12).
 - Nous déployons un **cursus de formation en 3 niveaux de maturité** (comprendre, s'engager, agir) sur l'ingénierie systèmes incluant SysML .
- **Recommandation :**
 - Inscrire le langage dans une démarche :
d'abord faire l'apprentissage du processus et méthodes associées de l'ingénierie système avant d'appliquer SysML.

Conclusion :

- La construction de systèmes performants et simple vue des utilisateurs est un enjeu majeur.

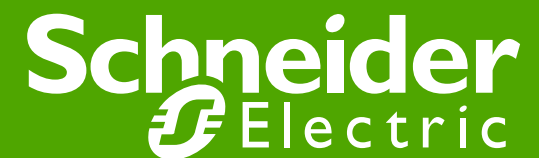


- L'approche de l'ingénierie Système structure et organise le « comment réaliser et maintenir ces systèmes simples et performants ».
- Le langage SysML s'inscrit dans cette démarche.

Quelques Exemples en Annexe :

Annexe : Mise en oeuvre de SysML

Partner Business Unit

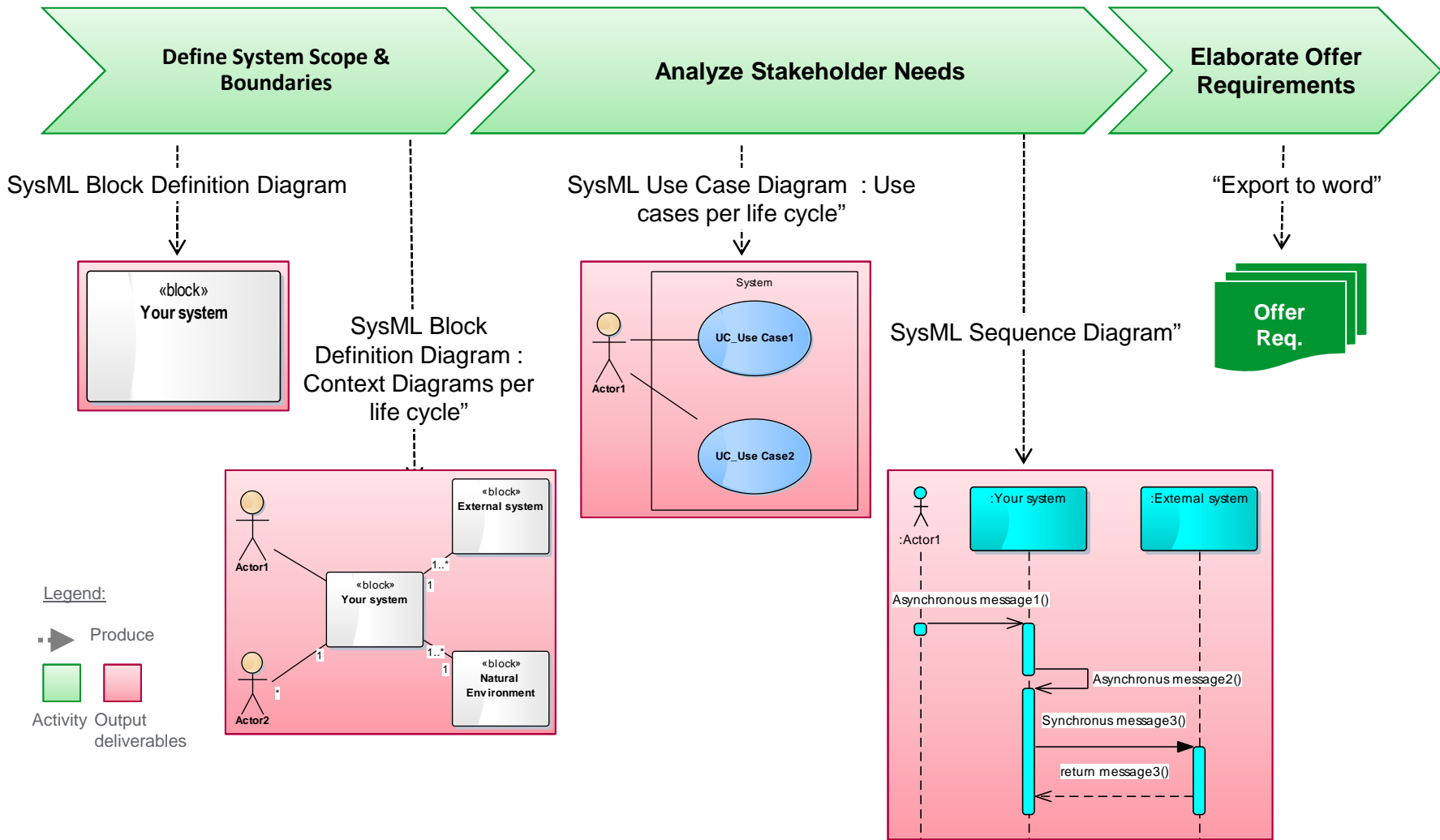


Mise en oeuvre de SysML

- Le langage de modélisation SysML est utilisé pour:
 - Collecter le besoin des parties prenantes (Use Cases, Sequence Diagrams)
 - Spécifier le système (Sequence Diagrams)
 - Définir l'architecture fonctionnelle (Activity Diagrams)
 - Définir l'architecture logique (Block Definition Diagram, Internal Block Diagram)
 - Définir l'architecture Technologique (Block Definition Diagram, Internal Block Diagram)
- Les diapositives suivantes illustrent l'utilisation de ces diagrammes.

“Collecter le besoin des parties prenantes ”

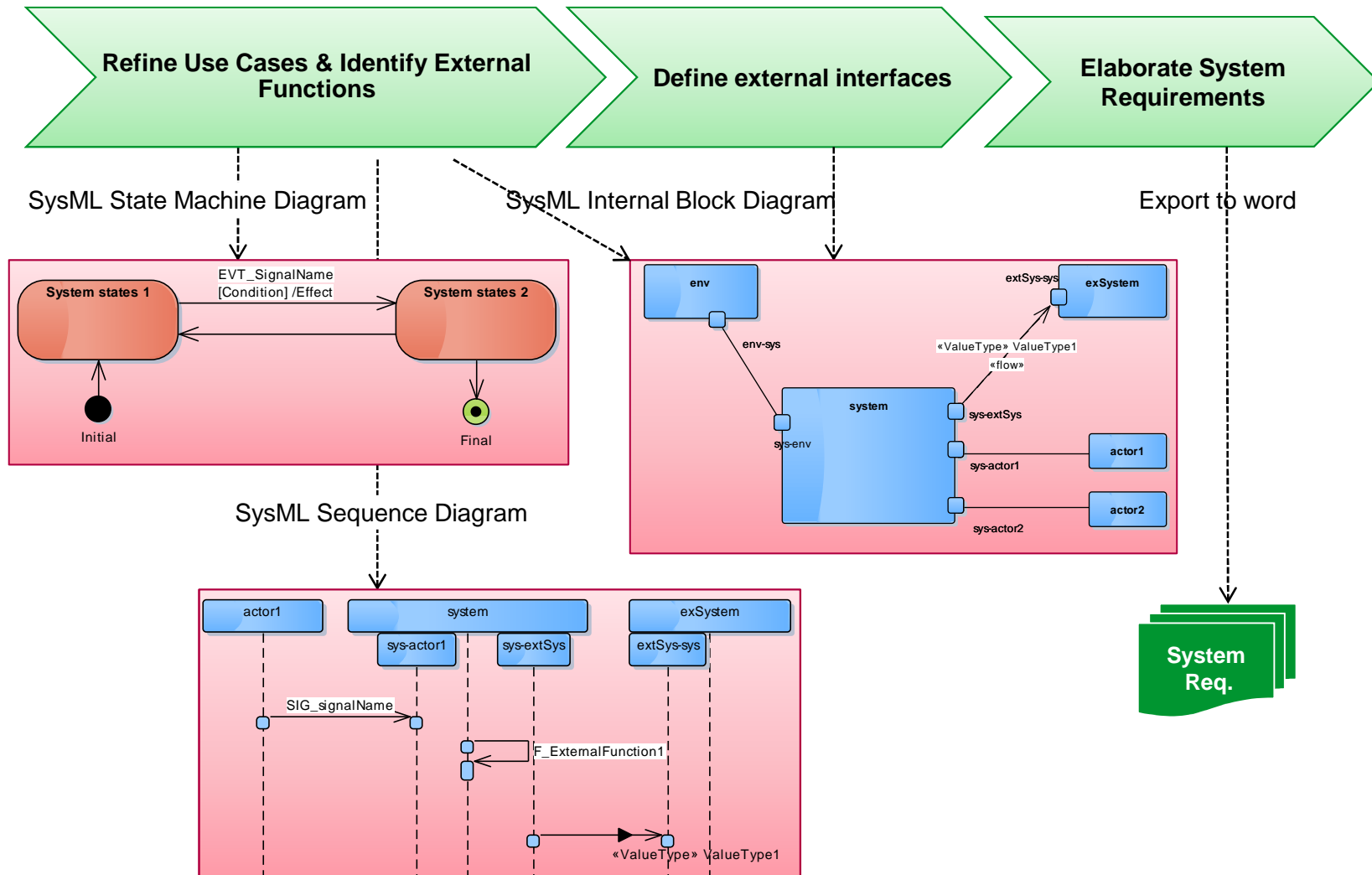
Output Define the Needs : Synthesis



“Spécifier le système ”

Output

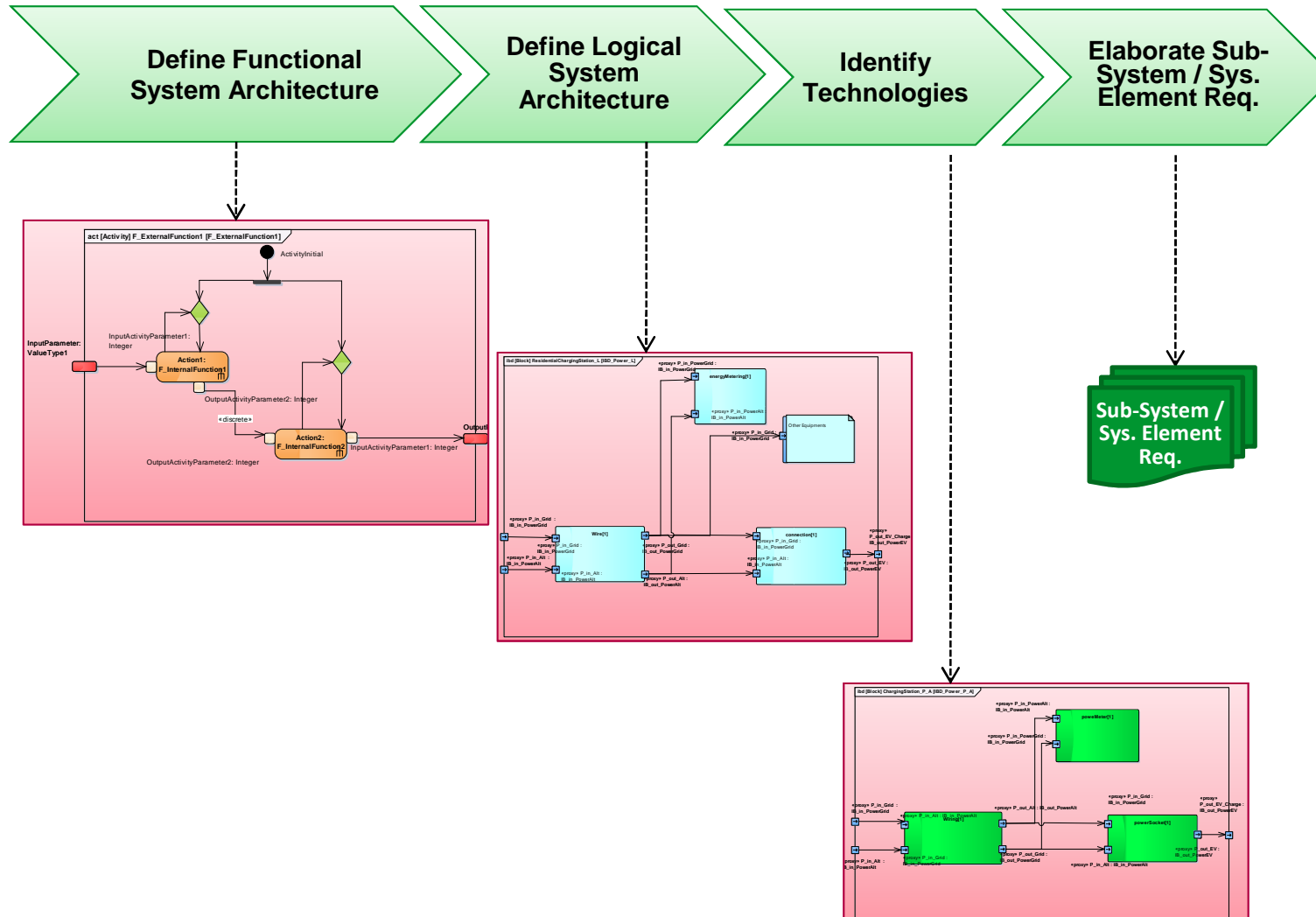
Specify the System : Synthesis



“Définir l’architecture fonctionnelle ”
“Définir l’architecture logique »
« Définir l’architecture Technologique »

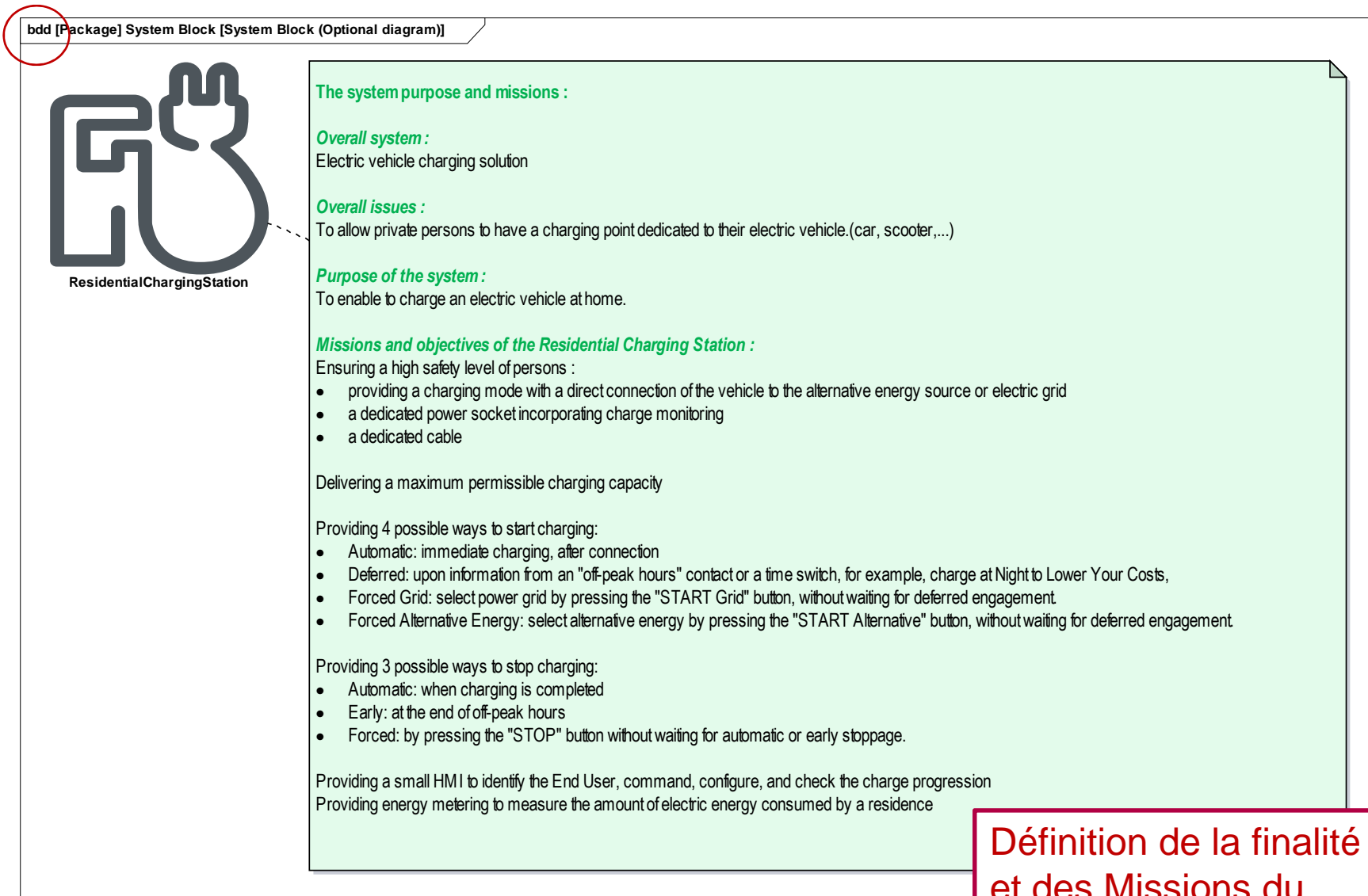
What

Build System Architecture



“Exemple du Véhicule Electrique”

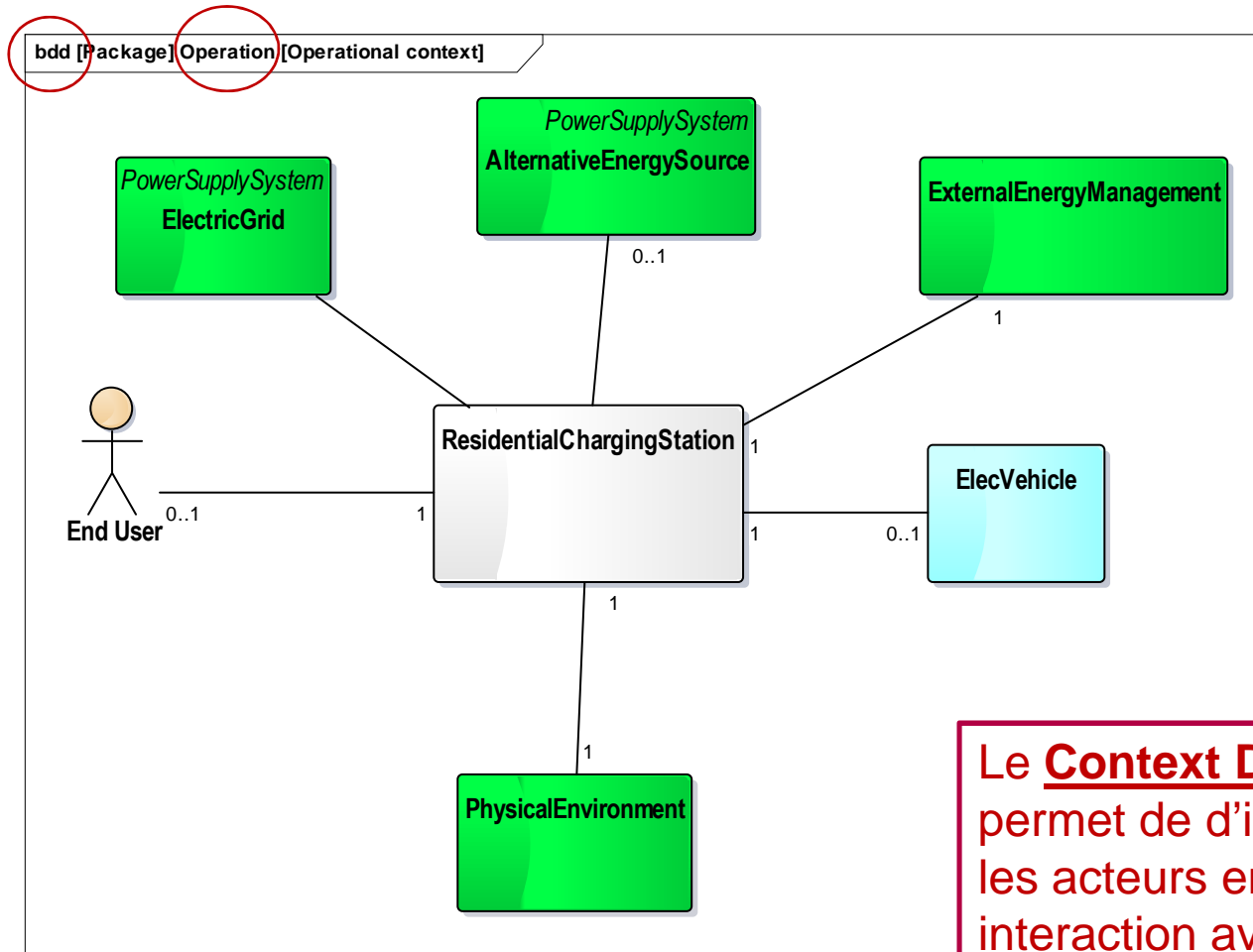
Collecter le besoin des parties prenantes :



Définition de la finalité
et des Missions du
Système à développer.

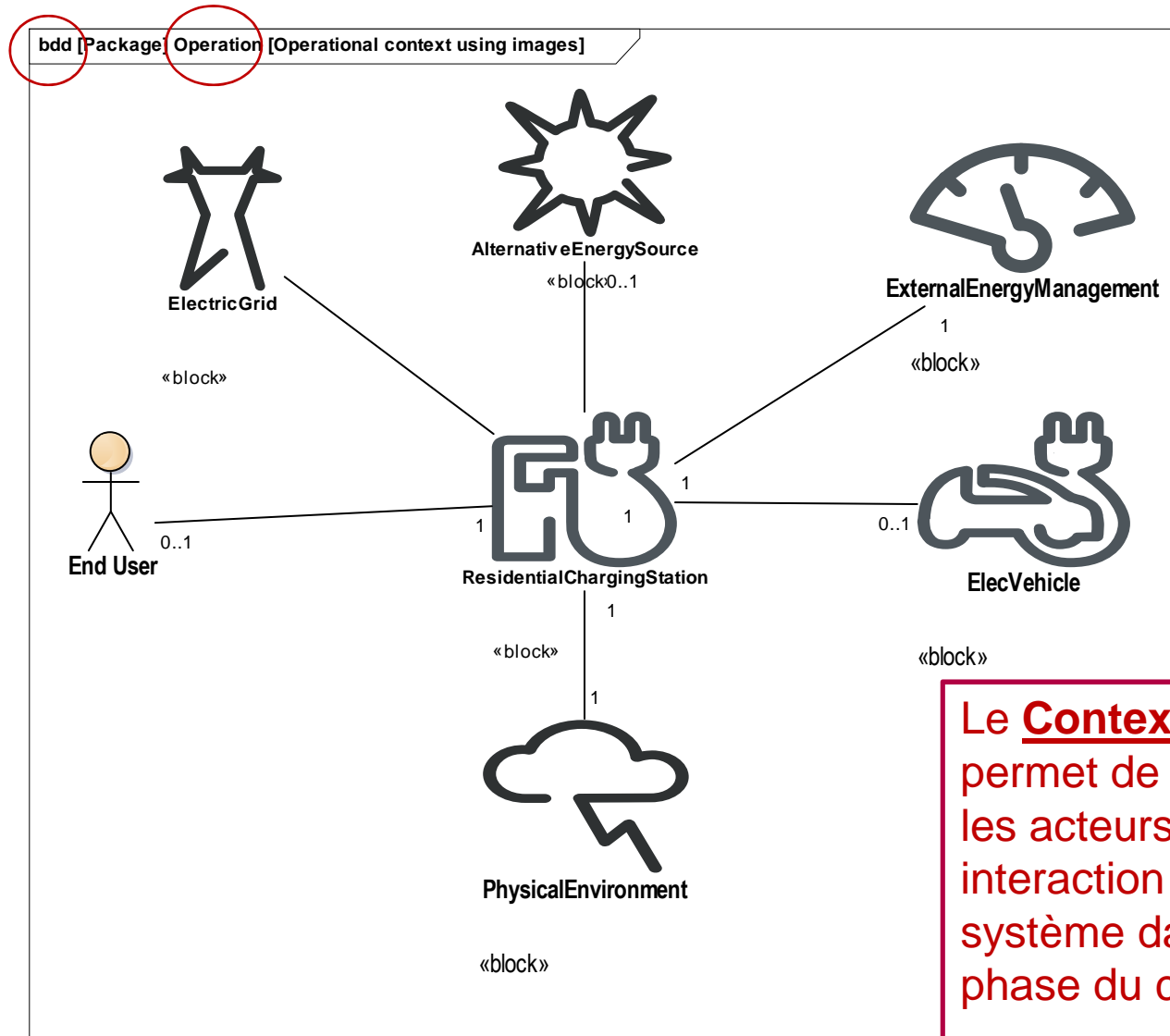
“Collecter le besoin des parties prenantes ”

Collecter le besoin des parties prenantes :



Le **Context Diagram** permet de d'identifier les acteurs en interaction avec le système dans chaque phase du cycle de vie.

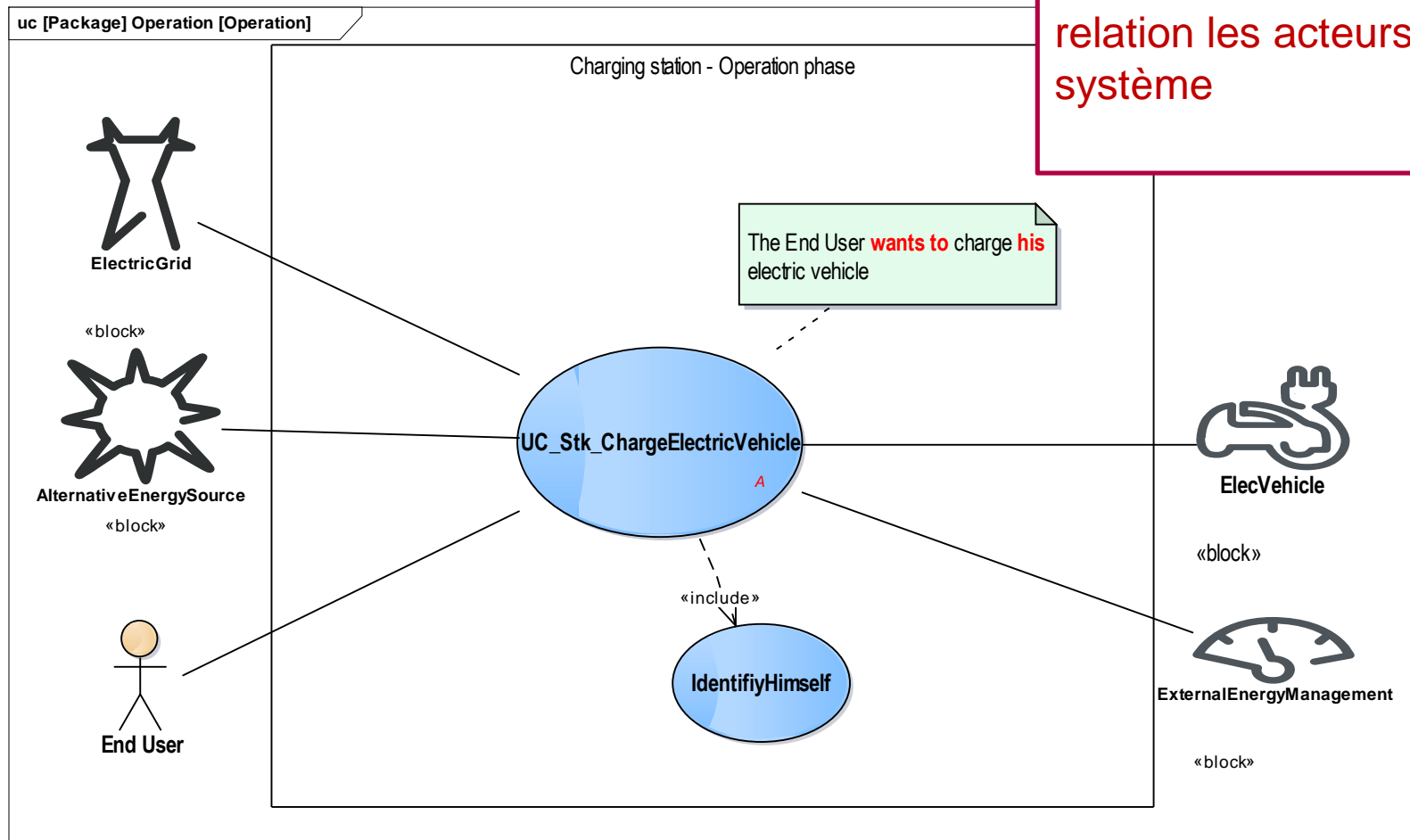
Collecter le besoin des parties prenantes :



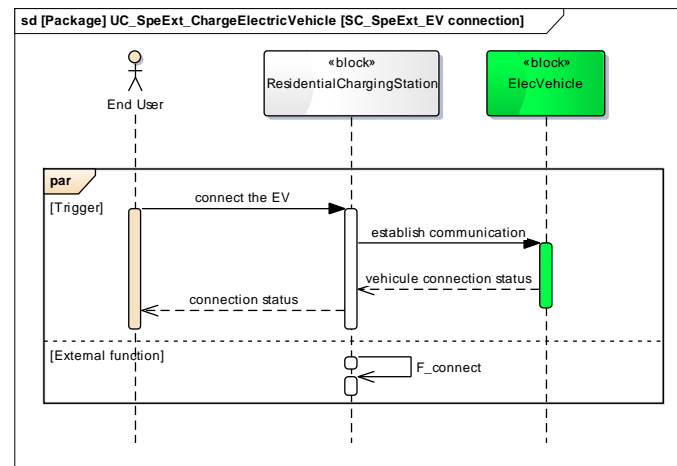
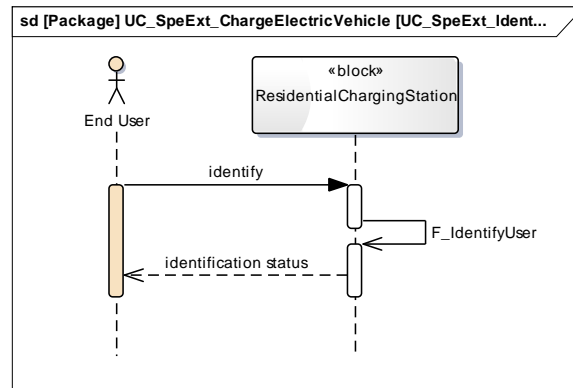
Le Context Diagram permet de d'identifier les acteurs en interaction avec le système dans chaque phase du cycle de vie.

Collecter le besoin des parties prenantes :

Les **Use Cases** permettent de définir le besoin en mettant en relation les acteurs et le système



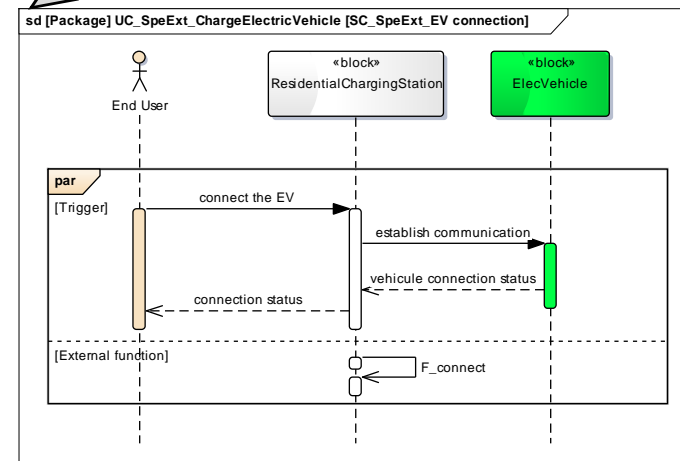
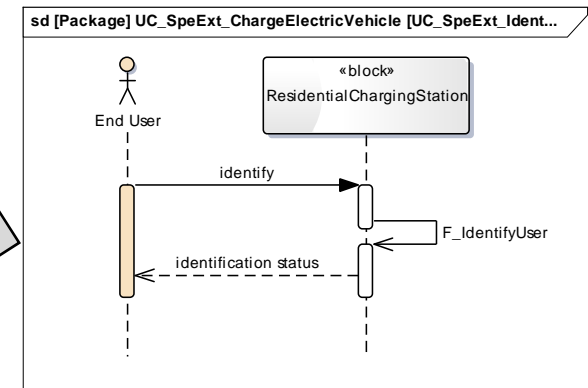
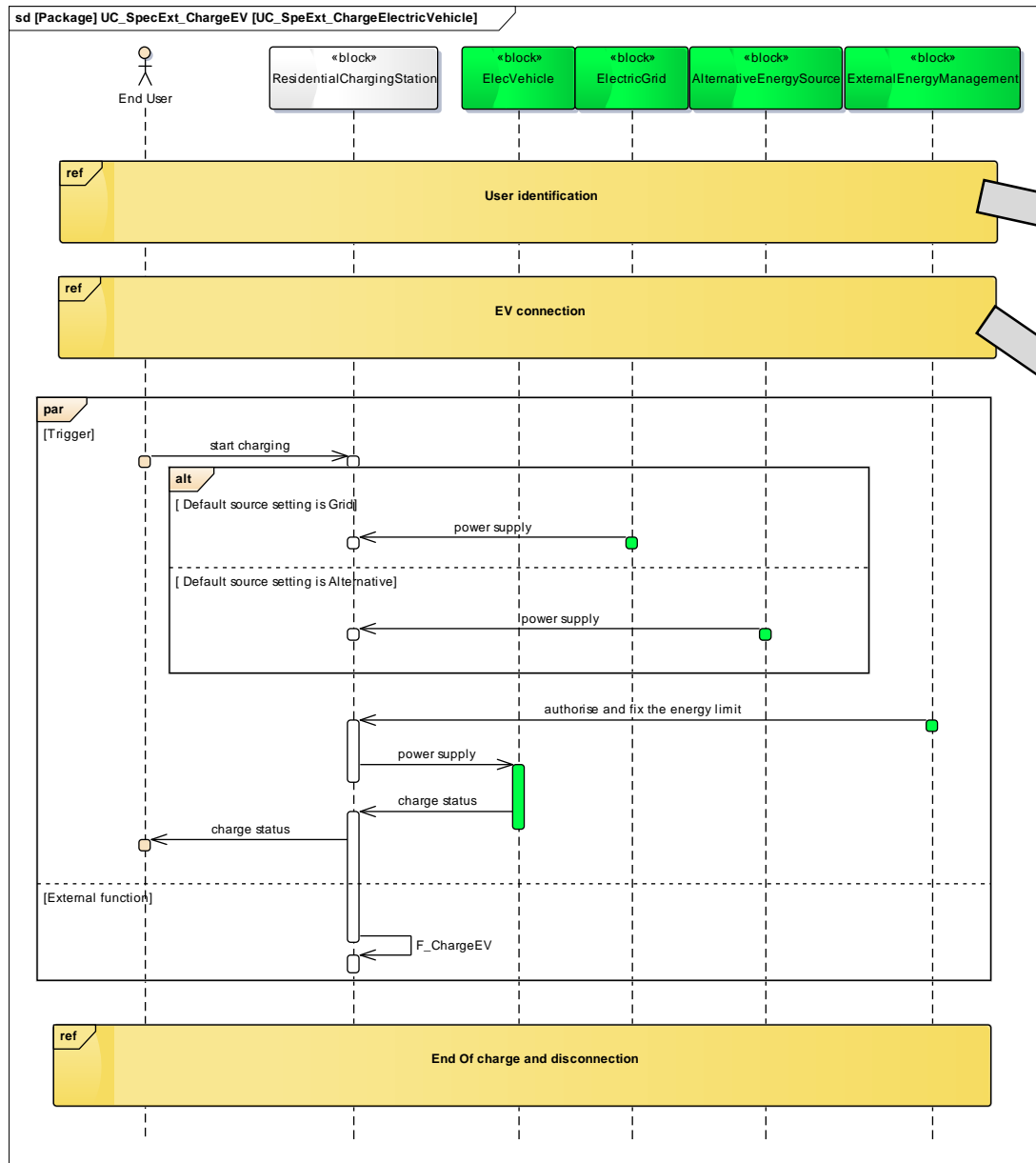
Collecter le besoin des parties prenantes :



Les **Sequence Diagrams** raffinent l'expression de besoin en mettant en évidence les interactions entre le système et les acteurs, pour chaque **Use Case**.

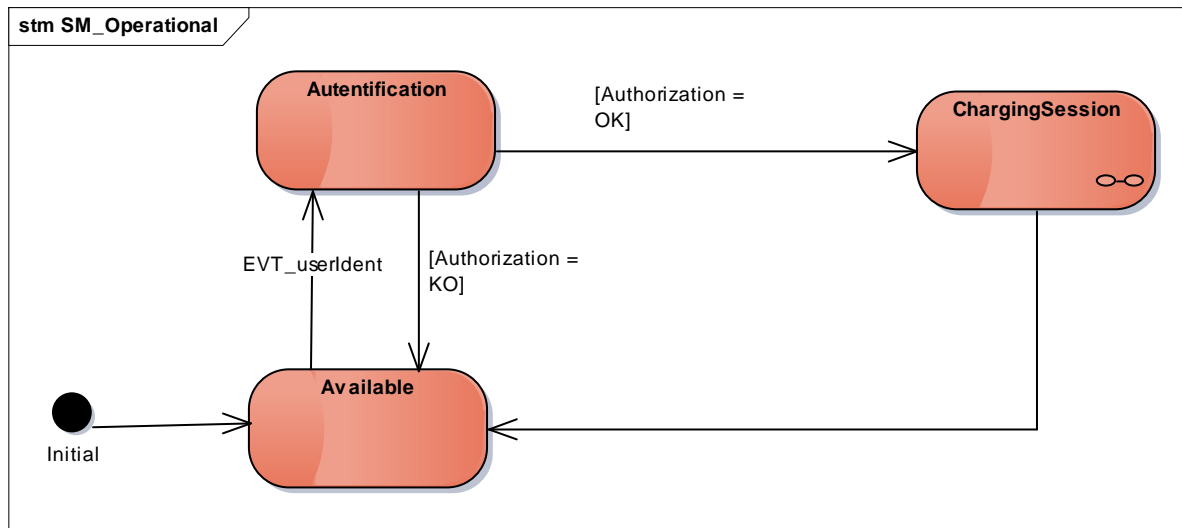
“Spécifier le système ”

Spécifier le système :



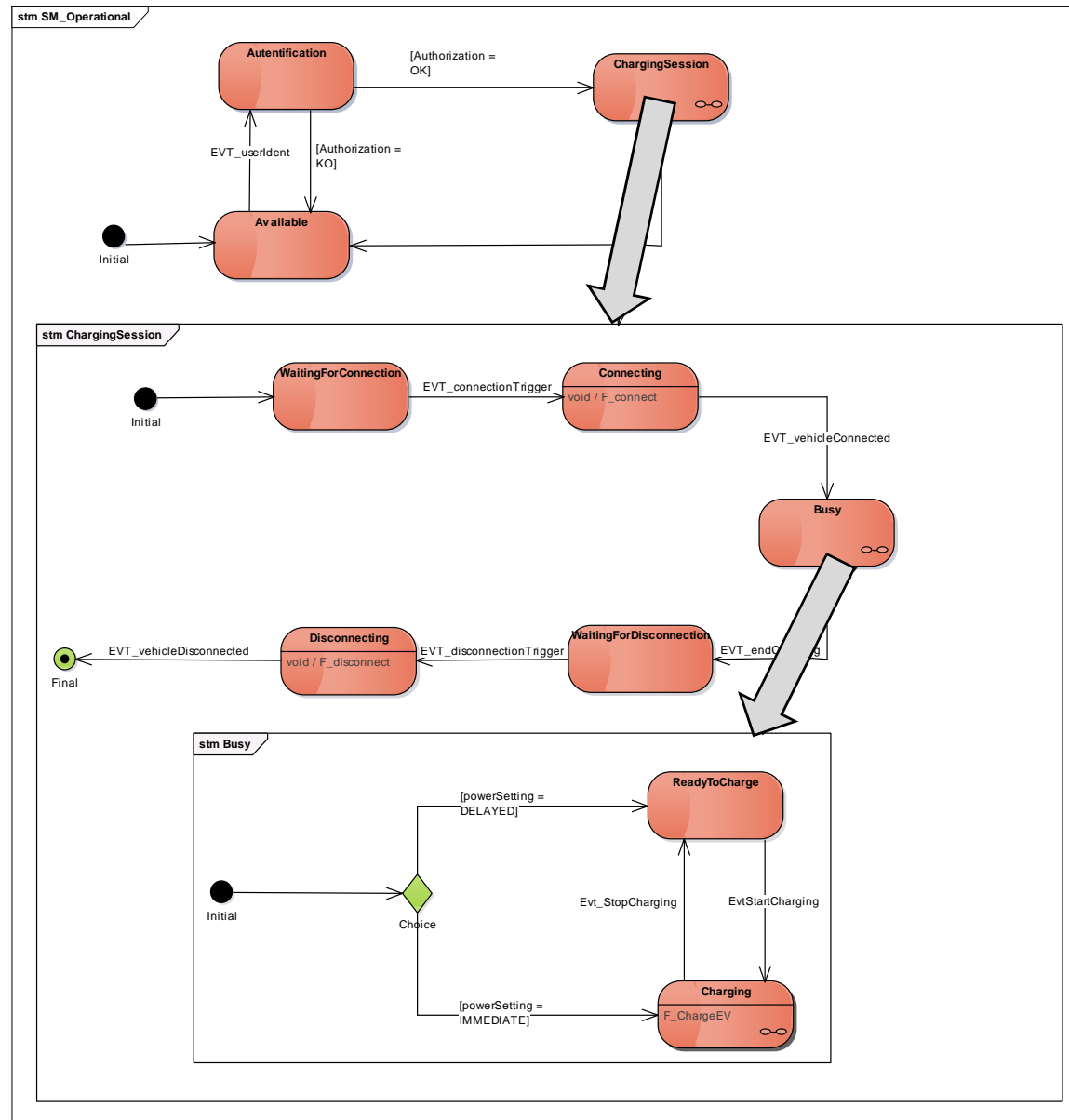
Les **Sequence Diagrams** sont détaillés: les interactions entre les acteurs et le système sont transformées en signaux et les Fonctions Externes sont identifiées.

Spécifier le système :

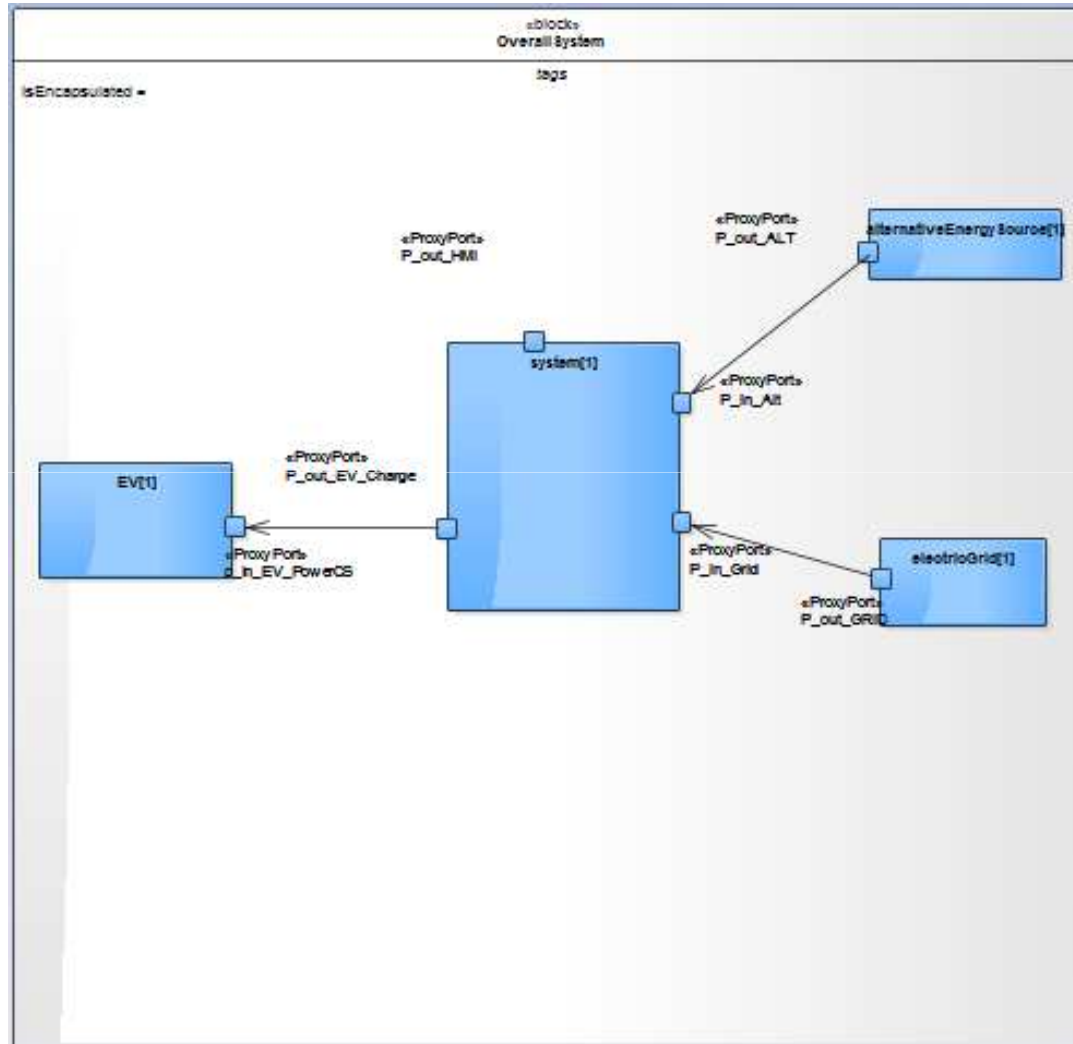


Les **State Diagram** permettent de décrire les états du système. Cela permet la description des évènements déclencheurs et d'allouer ensuite les fonctions externes par état.

Spécifier le système :



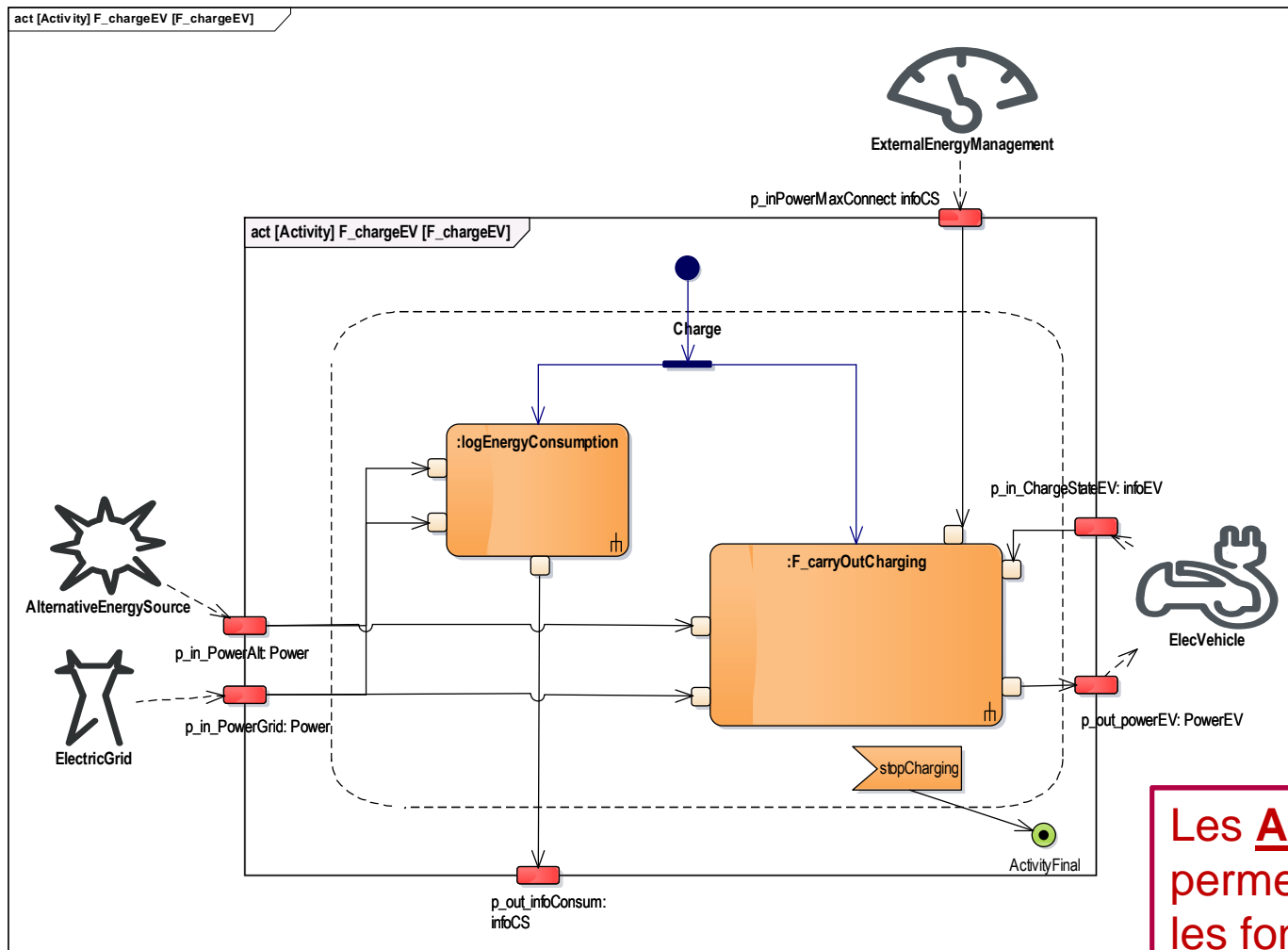
Spécifier le système :



Identification des interfaces Externes

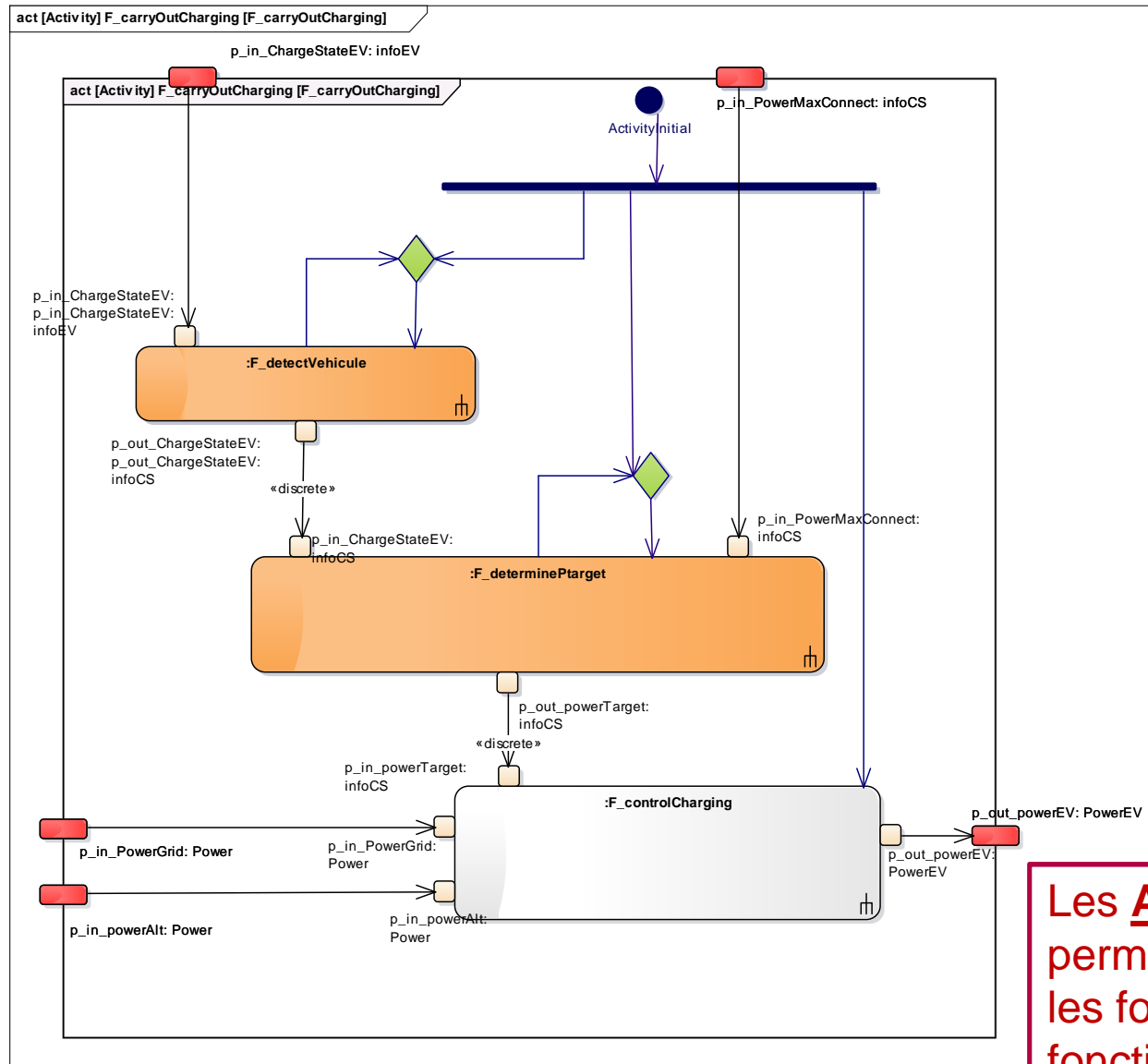
“Définir l’architecture fonctionnelle ”
“Définir l’architecture logique »
« Définir l’architecture Technologique »

Définir l'architecture fonctionnelle :



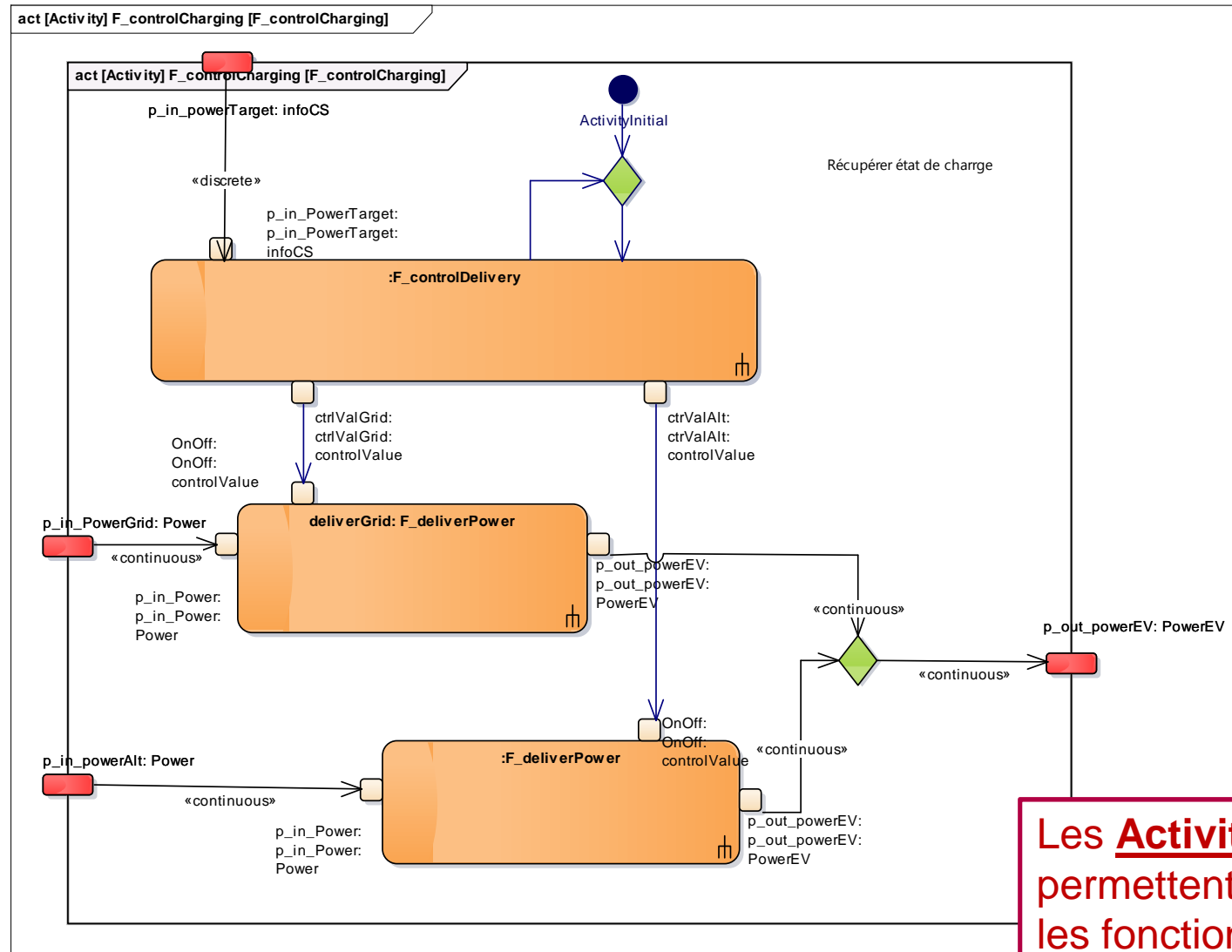
Les **Activity Diagrams** permettent de décomposer les fonctions externes en fonctions internes.

Définir l'architecture fonctionnelle :



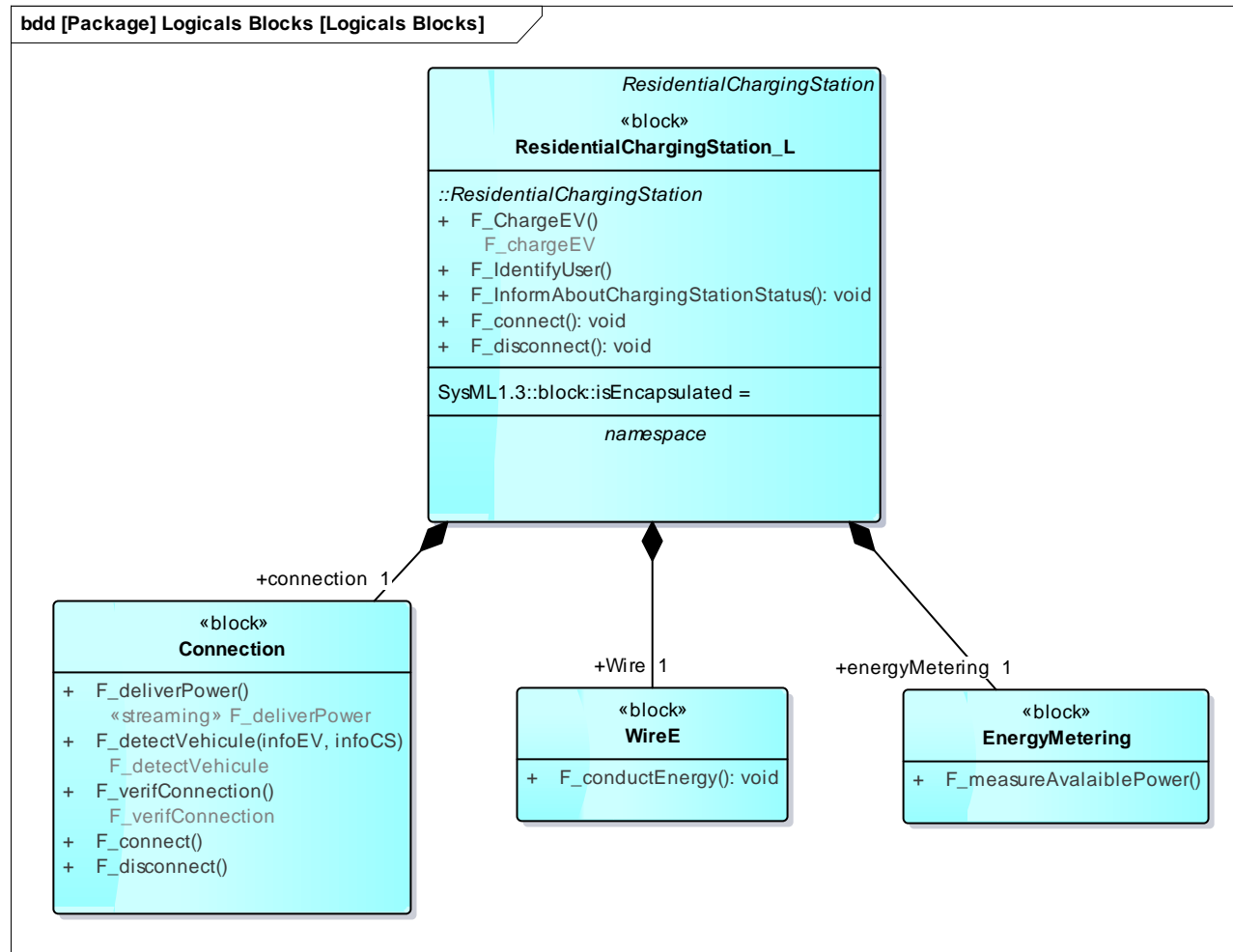
Les **Activity Diagrams** permettent de décomposer les fonctions externes en fonctions internes.

Définir l'architecture fonctionnelle :



Les **Activity Diagrams** permettent de décomposer les fonctions externes en fonctions internes.

Définir l'architecture logique :

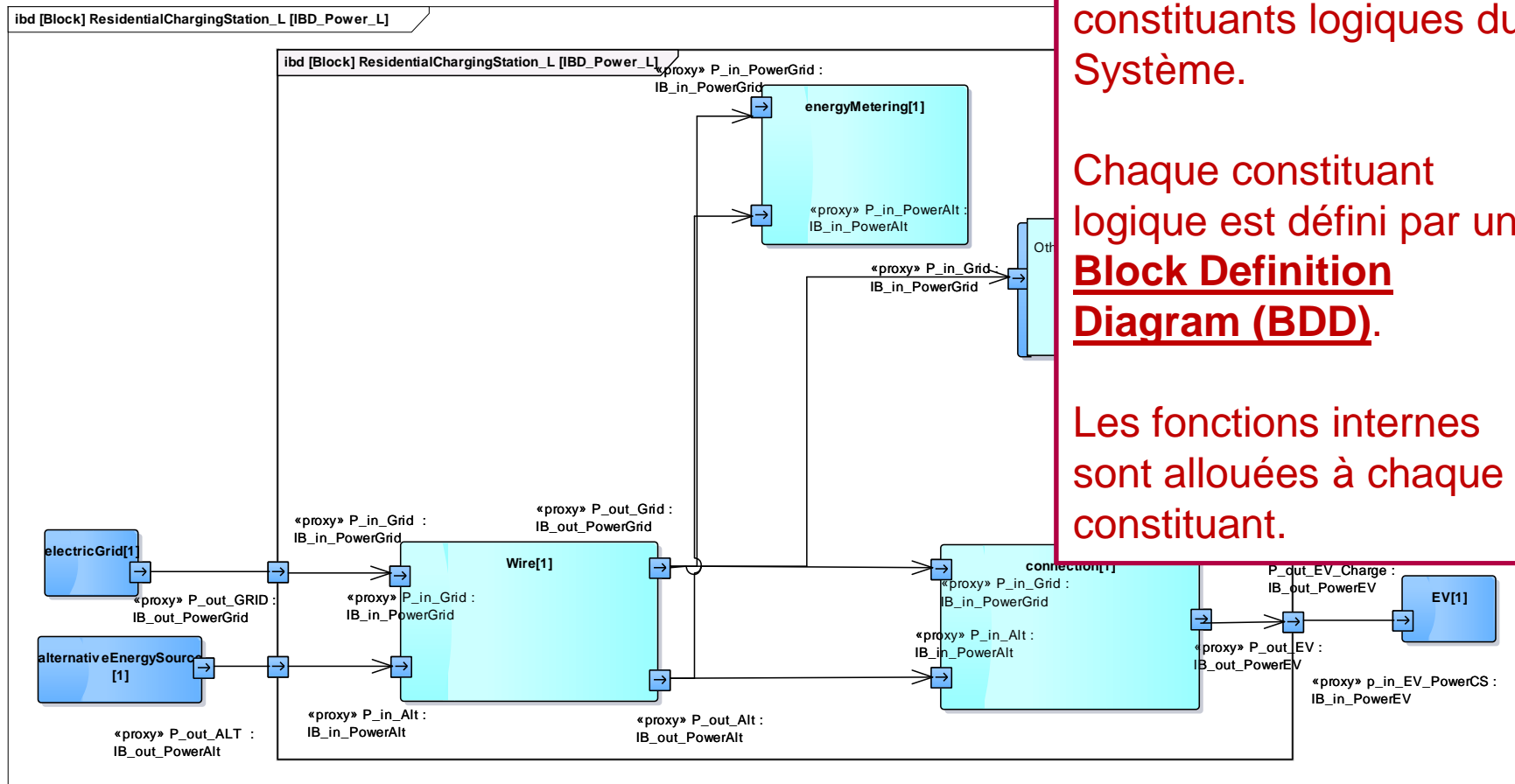


Définir l'architecture logique :

Les **Internal Block Diagrams (IBD)** permettent d'identifier les interfaces entre les composants logiques du Système.

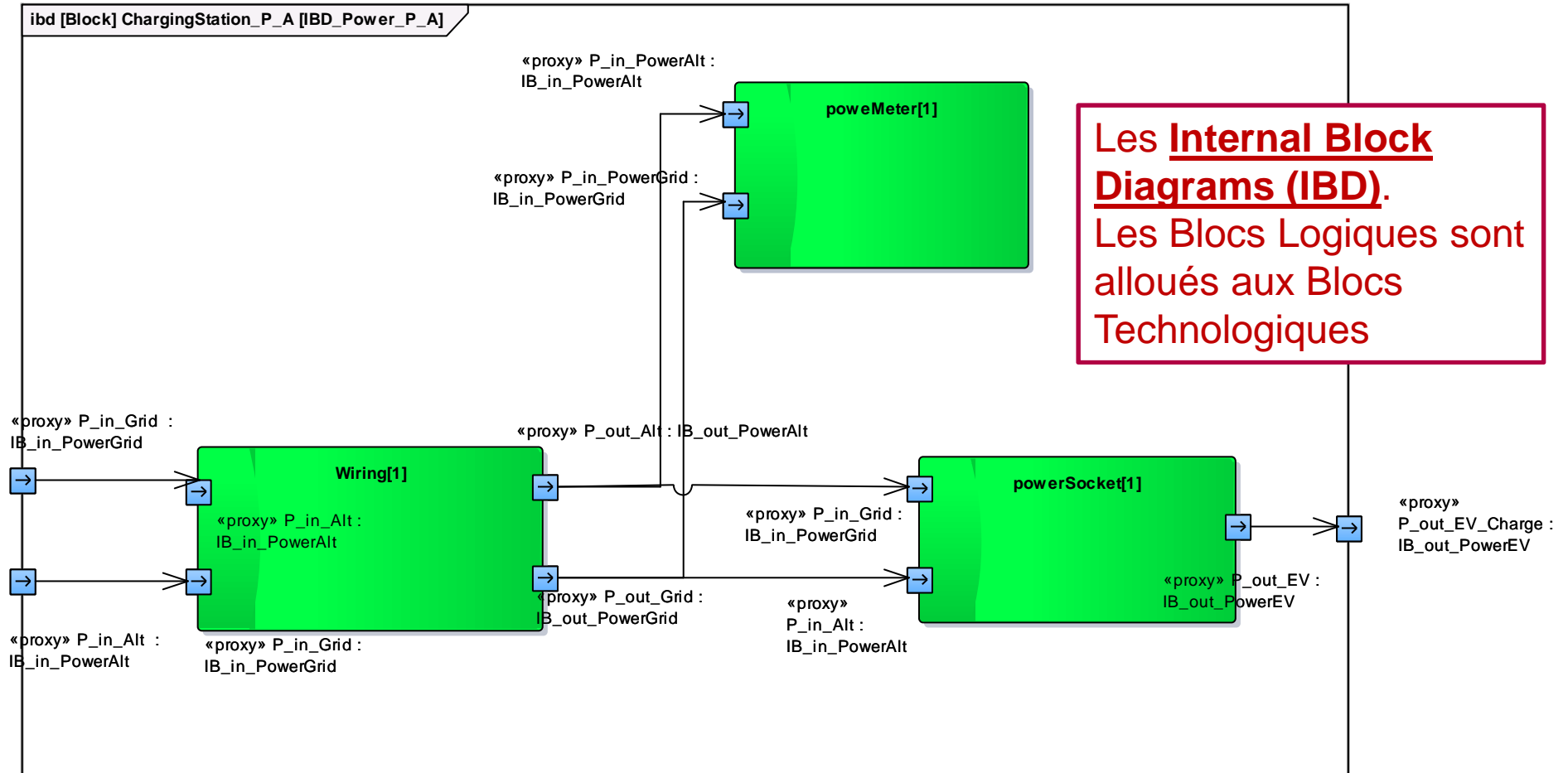
Chaque constituant logique est défini par un **Block Definition Diagram (BDD)**.

Les fonctions internes sont allouées à chaque constituant.




Définir l'architecture Technologique :

ibd [Block] ChargingStation_P_A [IBD_Power_P_A]



QUESTIONS ?

Schneider
 Electric

Bibliographie :

- **SysML par l'exemple** , Un langage de modélisation pour systèmes complexes
- [Pascal Roques \(Auteur\) - Guide \(broché\). Paru en 05/2013](#)

- Sites web de Pascal Roques

- **A Practical Guide to SysML, Second Edition: The Systems Modeling Language (The MK/OMG Press) Paperback – October 16, 2007**
- by [Sanford Friedenthal \(Author\), Alan Moore \(Author\), Rick Steiner \(Author\)](#)