



Éléments d'architecture système

Daniel Krob

CESAMES & IRT SystemX

23 Septembre 2015

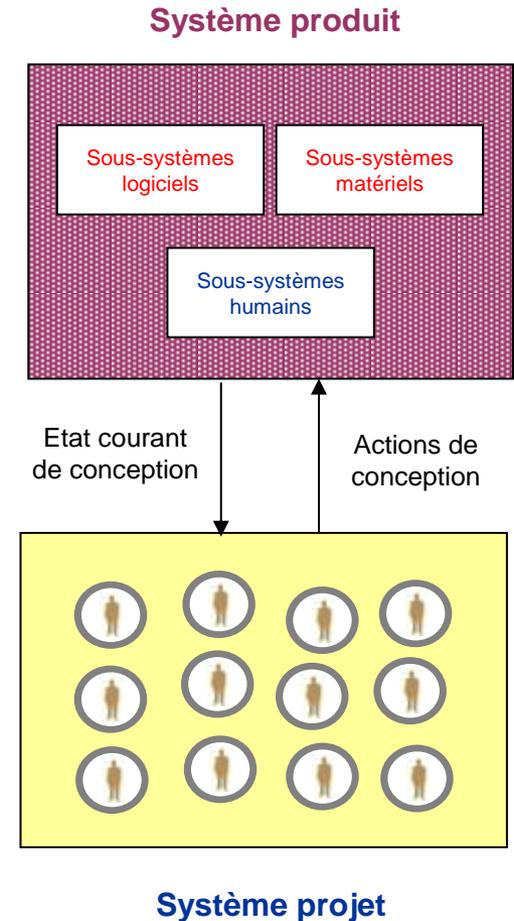


Table des matières

- **1 : Les enjeux de la maîtrise de la complexité**
- **2 : Les fondements de l'architecture système**
- **3 : Les impacts organisationnels de l'approche système**

Quelques catégories de problèmes systémiques classiques auxquels les projets “systèmes” sont confrontés

- **Problèmes d’intégration du produit**
 - **Les composants du système produit ne coopèrent pas correctement ensemble**
 - *Problème type* : un système intégré possède des propriétés émergentes non prévues & plus ou moins fortement indésirables (souvent résultant d’un effet « domino »)
- **Problèmes d’intégration du projet**
 - **Les parties prenantes du système projet ne collaborent pas correctement ensemble**
 - *Problème type* : le système projet oublie la mission du système produit dont il a la charge & chaque entité du projet travaille de manière silotée sans interface réelle avec les autres acteurs du projet



Exemple d'un problème d'intégration

Le cas du premier vol d'Ariane 5 (1/2)

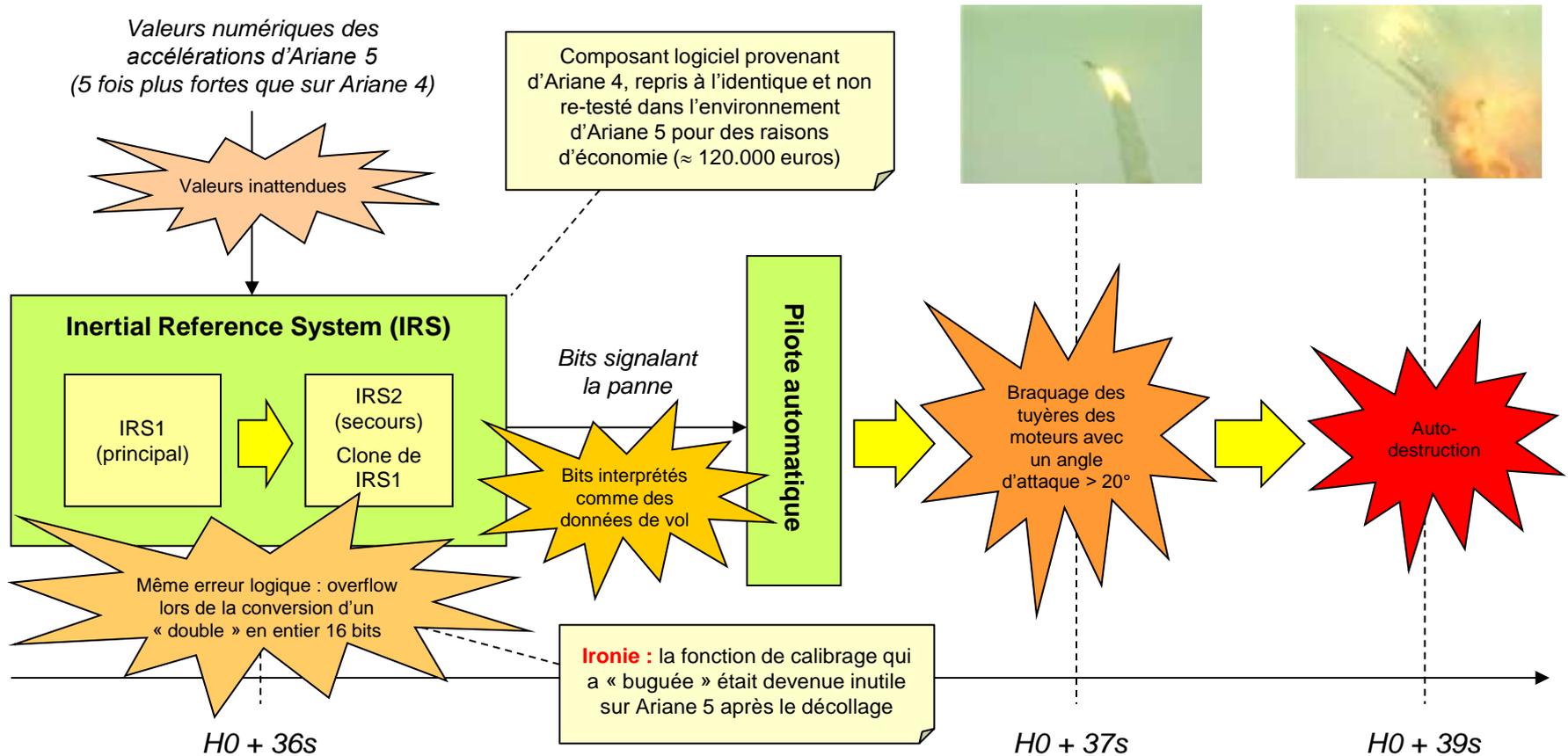


- **L'accident du 4 Juin 1996 :**
 - H0-H0+36 s. : vol parfait d'Ariane 5 jusqu'à la 36-ième seconde (1)
 - H0+36,7 s. : défaillance quasi-simultanée des deux systèmes de guidage inertiel
 - H0+37 s. : mise en route du pilote automatique qui interprète mal les signaux de panne des systèmes de guidage inertiel et corrige brutalement la trajectoire du lanceur (2)
 - H0+39 s. : rupture des boosters déclenchant l'auto-destruction de la fusée (3)
- **Coûts immédiats de l'accident :**
 - Coût direct de 370 millions de dollars (perte de la charge utile)
 - Coûts induits : 1 mois de travail pour récupérer les débris les plus dangereux (comme les réserves de carburant) dans l'eau opaque de la mangrove Guyanaise
- **Importants coûts indirects en terme de retards sur le programme Ariane 5 :**
 - Le second vol d'Ariane 5 n'aura lieu qu'un an après l'accident du vol inaugural
 - Le premier vol commercial d'Ariane 5 aura lieu 3 ans après, soit le 10 Décembre 1999

Exemple d'un problème d'intégration

Le cas du premier vol d'Ariane 5 (2/2)

Source : rapport de la commission Lions, 1996

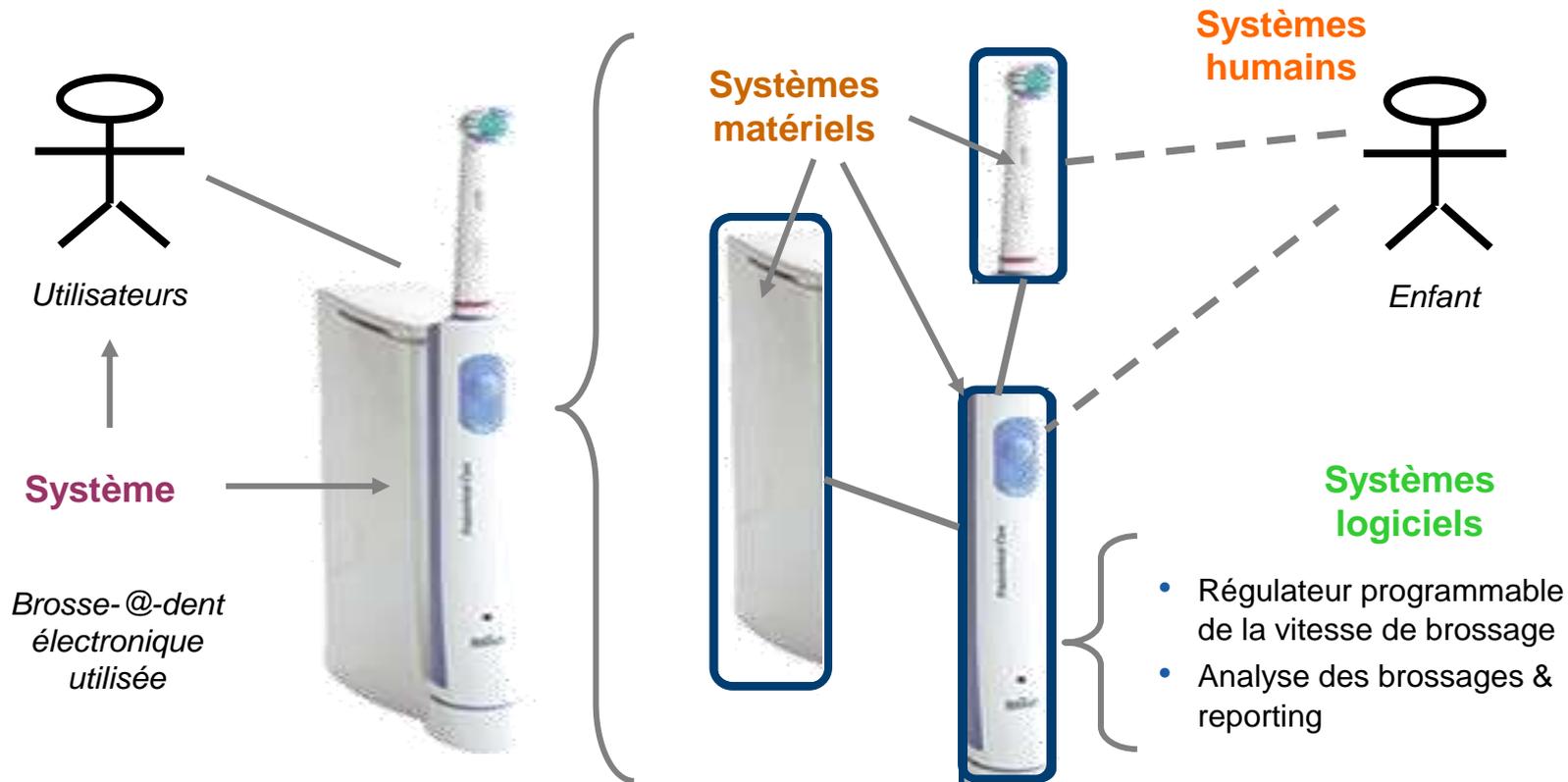


Le système Ariane 5 a été miné par une erreur d'intégration qui le rendait incorrect par construction !

Intégration



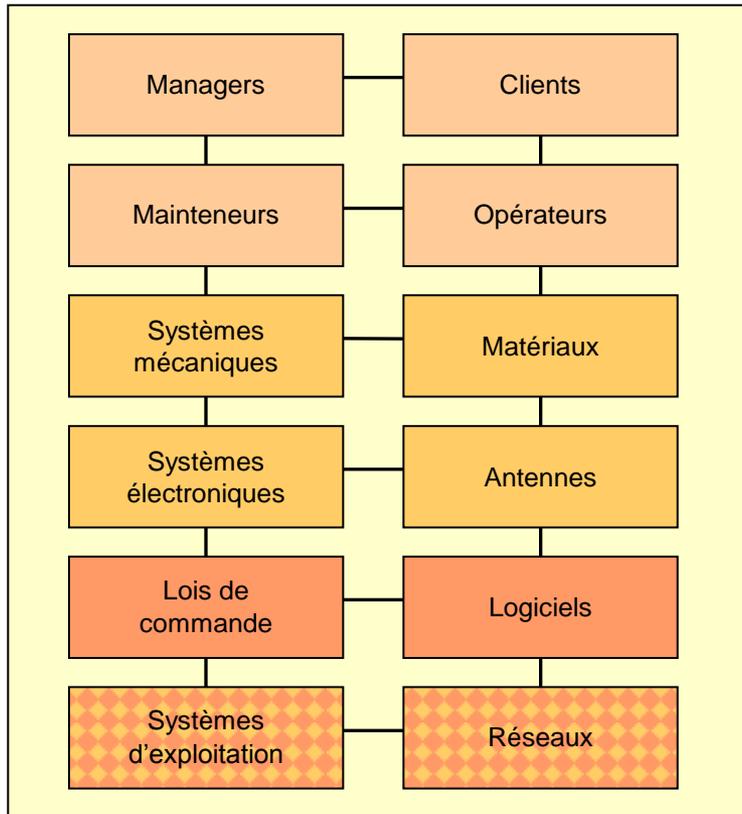
L'intégration engendre toujours de l'**émergence** qu'il faut donc maîtriser



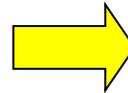
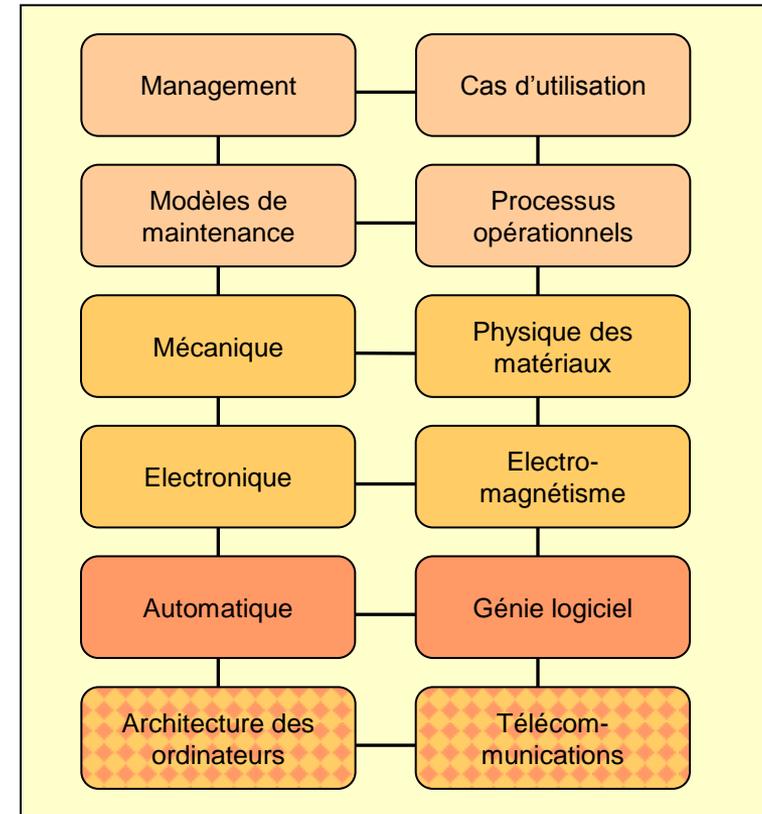
L'intégration est le mécanisme qui permet de **construire** un système **à partir d'autres systèmes** (matériels, logiciels & humains) en les organisant de manière à ce qu'il puisse accomplir – au sein d'un environnement donné – ses missions

La complexité réside dans l'intégration

Systeme à intégrer

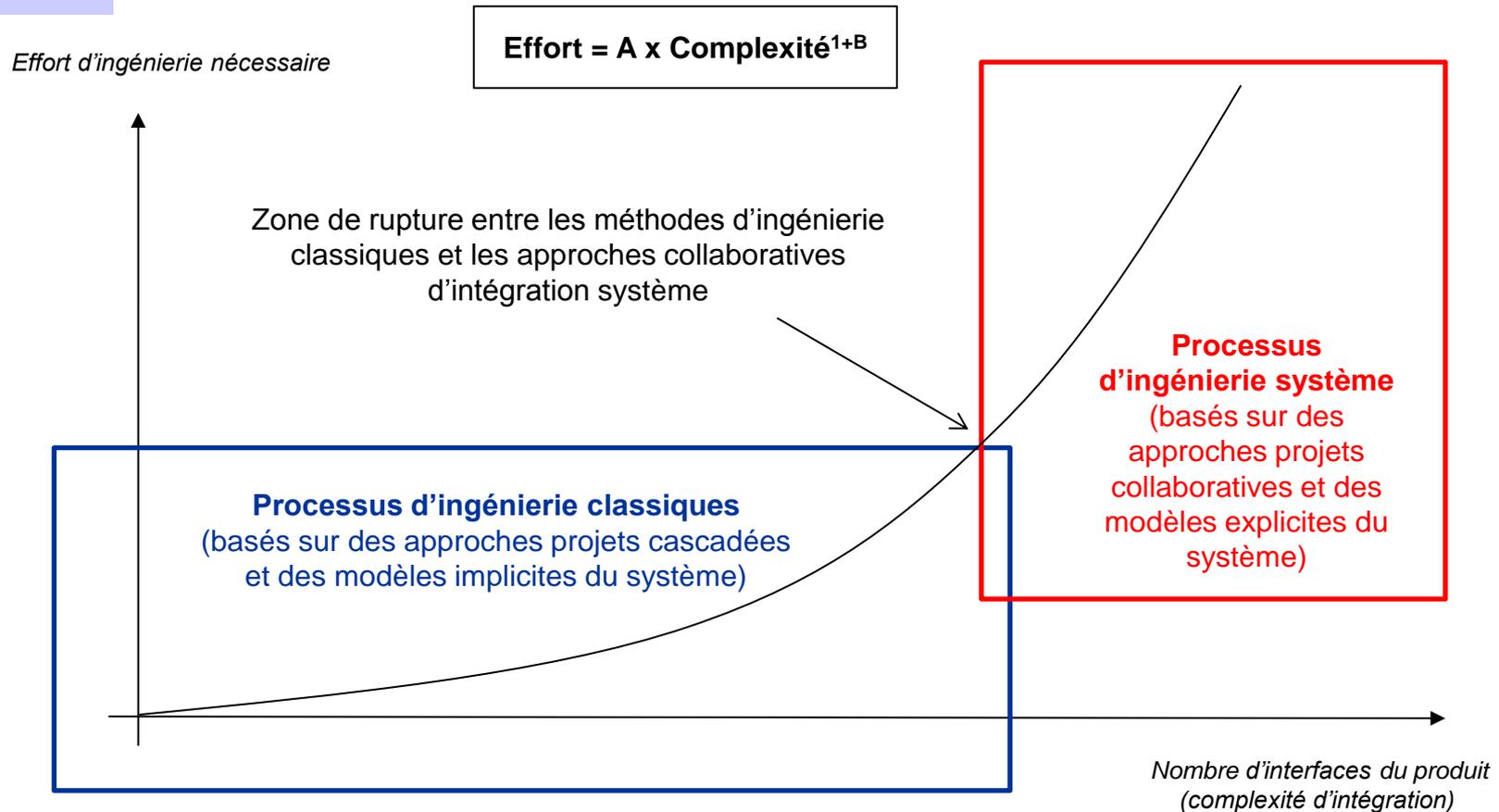


Ingénieries & modèles à intégrer



Maîtriser un système « complexe », c'est maîtriser l'intégration des systèmes hétérogènes qui le composent et l'intégration des ingénieries & des modèles associés

La barrière de la complexité



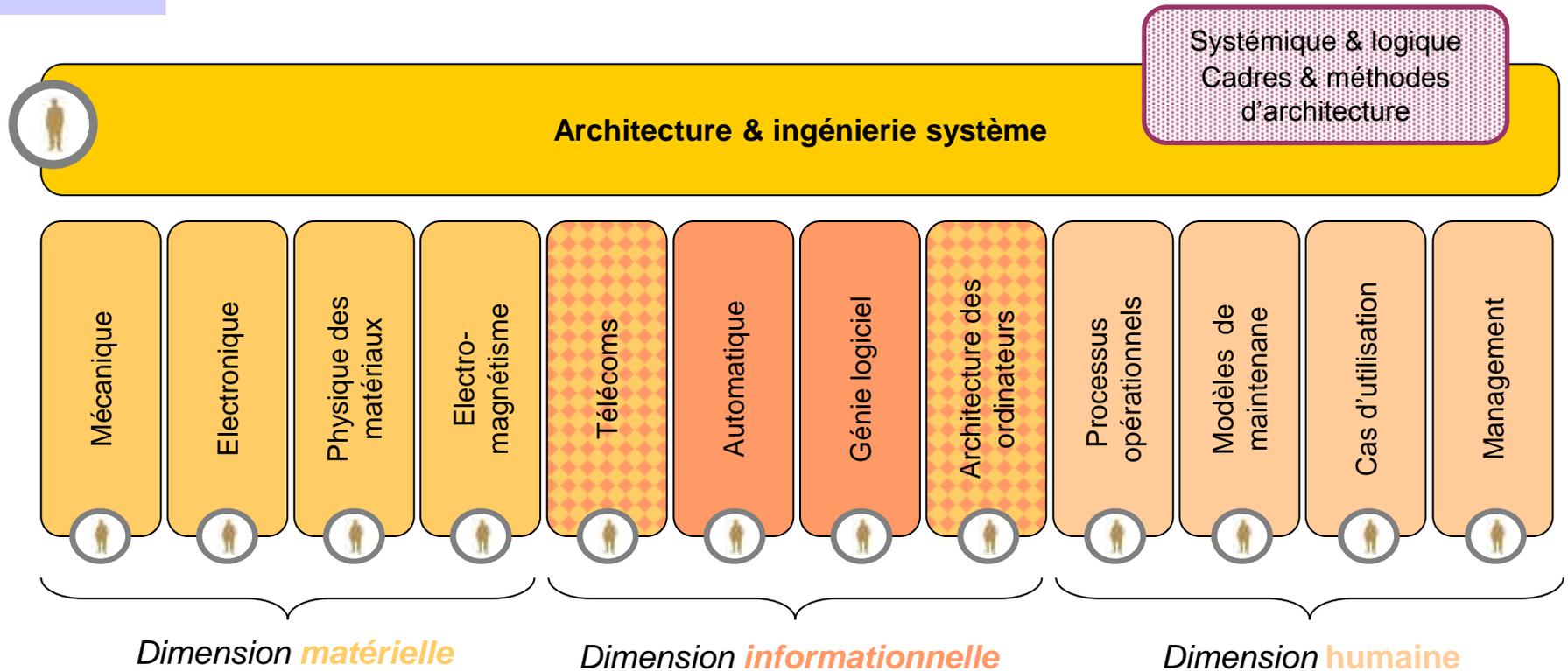
Quand la **complexité d'intégration d'un système augmente**, la capacité à **gérer efficacement les interfaces produit & les interactions projet** devient cruciale : ces nouveaux défis nécessitent donc d'utiliser de **nouvelles méthodes**



Table des matières

- 1 : Les enjeux de la maîtrise de la complexité
- 2 : Les fondements de l'architecture système
- 3 : Les impacts organisationnels de l'approche système

L'approche système comme discipline intégratrice

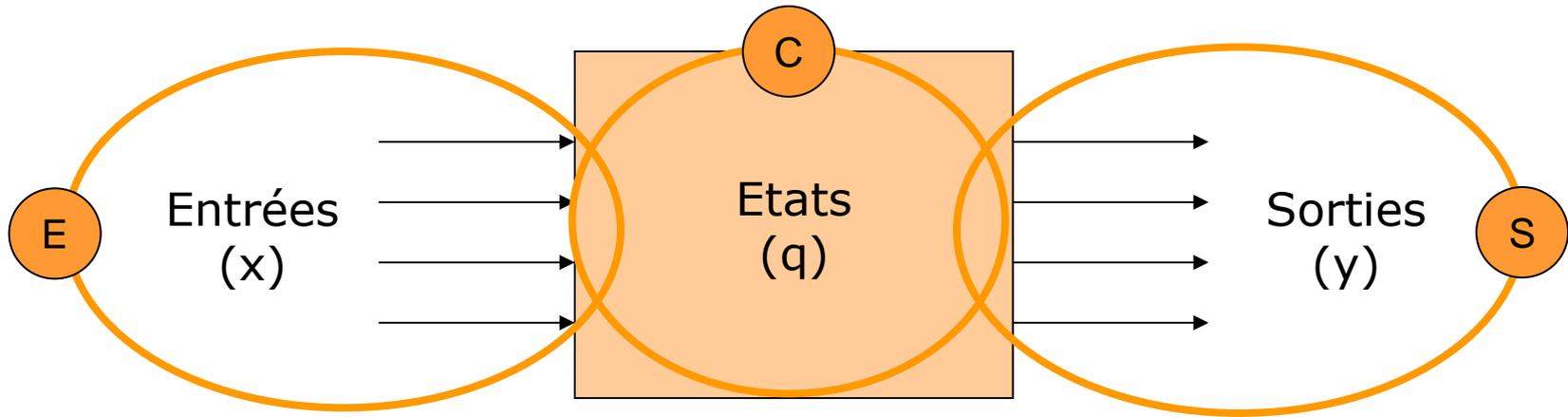


L'**architecture & l'ingénierie système** sont des disciplines **intégratrices** qui permettent de penser dans une même **vision d'ensemble** & en **subsidiarité** avec les **disciplines & les ingénieries existantes** tous les **composants matériels & logiciels** (dimension technique) et tous les **facteurs humains** (dimension humaine) d'un système

La notion de système



Considérer un objet comme un système est donc un **choix de modélisation**, pas une caractéristique intrinsèque de l'objet



Pré-conditions (besoins & contraintes pour le système considéré), données, ressources, décisions, etc.

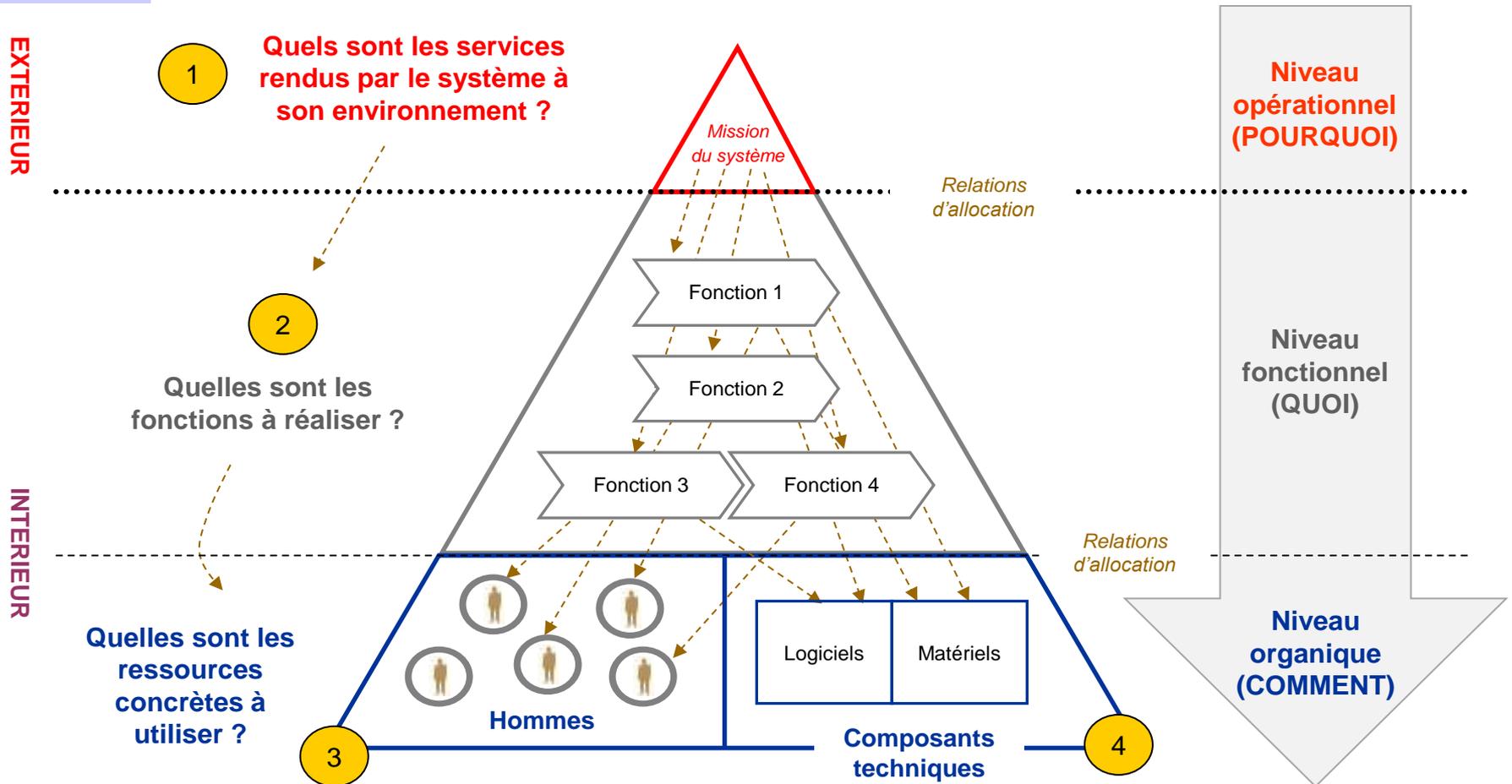
Traitements internes

Post-conditions (besoins & contraintes pour d'autres systèmes), données, livrables, actions, etc.

Un **système** est caractérisé par un **double comportement entrée / sortie et interne** qui lui permet en permanence – i.e. tout au long du **temps (t)** – d'une part de transformer des **entrées (x)** en **sorties (y)** selon la nature de ses **états internes (q)** et d'autre part de faire évoluer ses **états internes (q)** sous l'action de ses **entrées (x)**

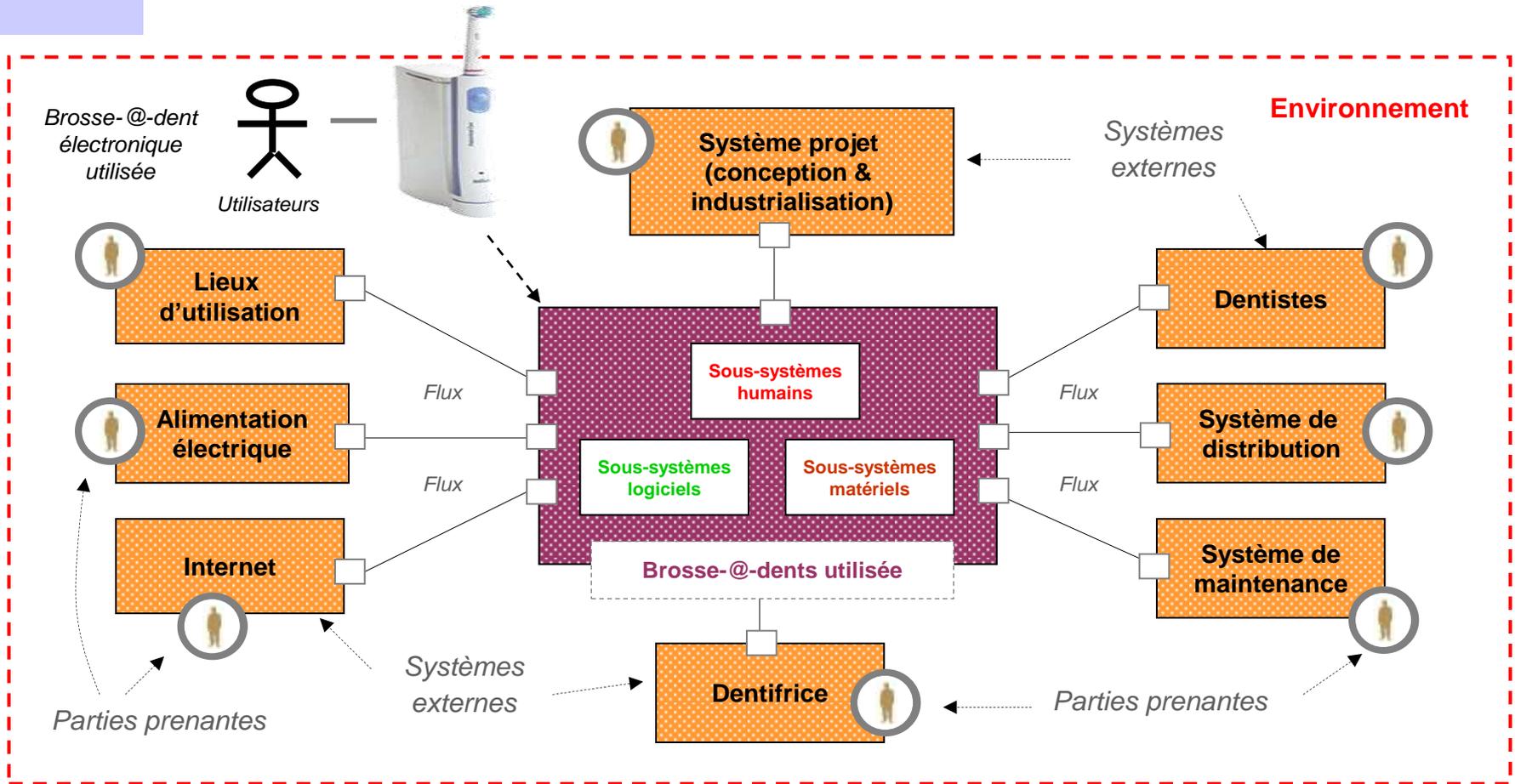
Note : les systèmes « ingénierés » peuvent se classer en trois grandes catégories en fonction des lois qui régissent les comportements qui les définissent : **systèmes matériels** (lois physiques), **systèmes logiciels** (lois logiques) et **systèmes humains** (lois du comportement humain)

Les trois grandes visions architecturales permettant d'analyser un système



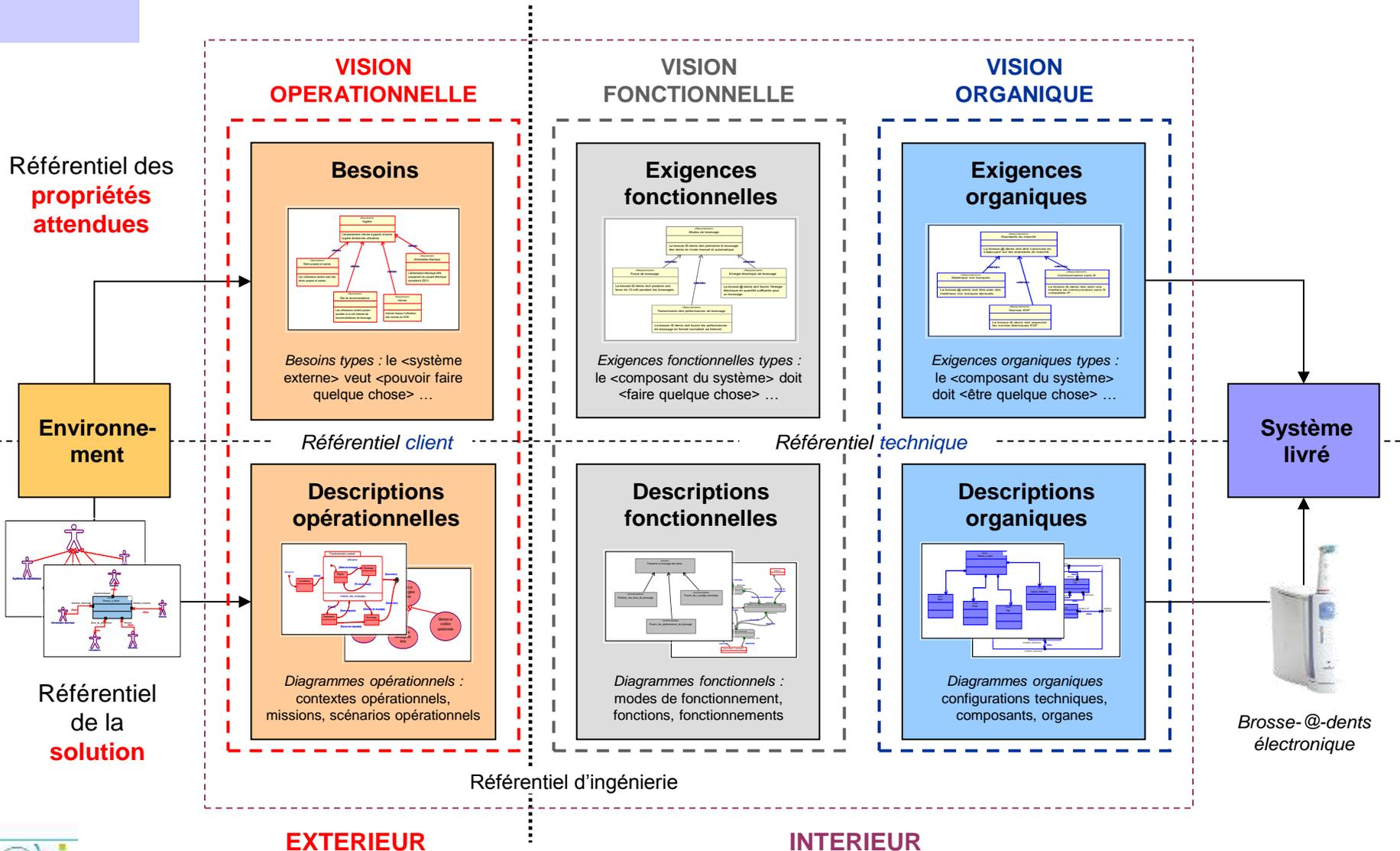
L'architecture système repose sur un changement de paradigme de conception : le passage d'une logique « **Bottom-Up** » à une logique « **Top-Down** »

Le point de départ de toute analyse architecturale

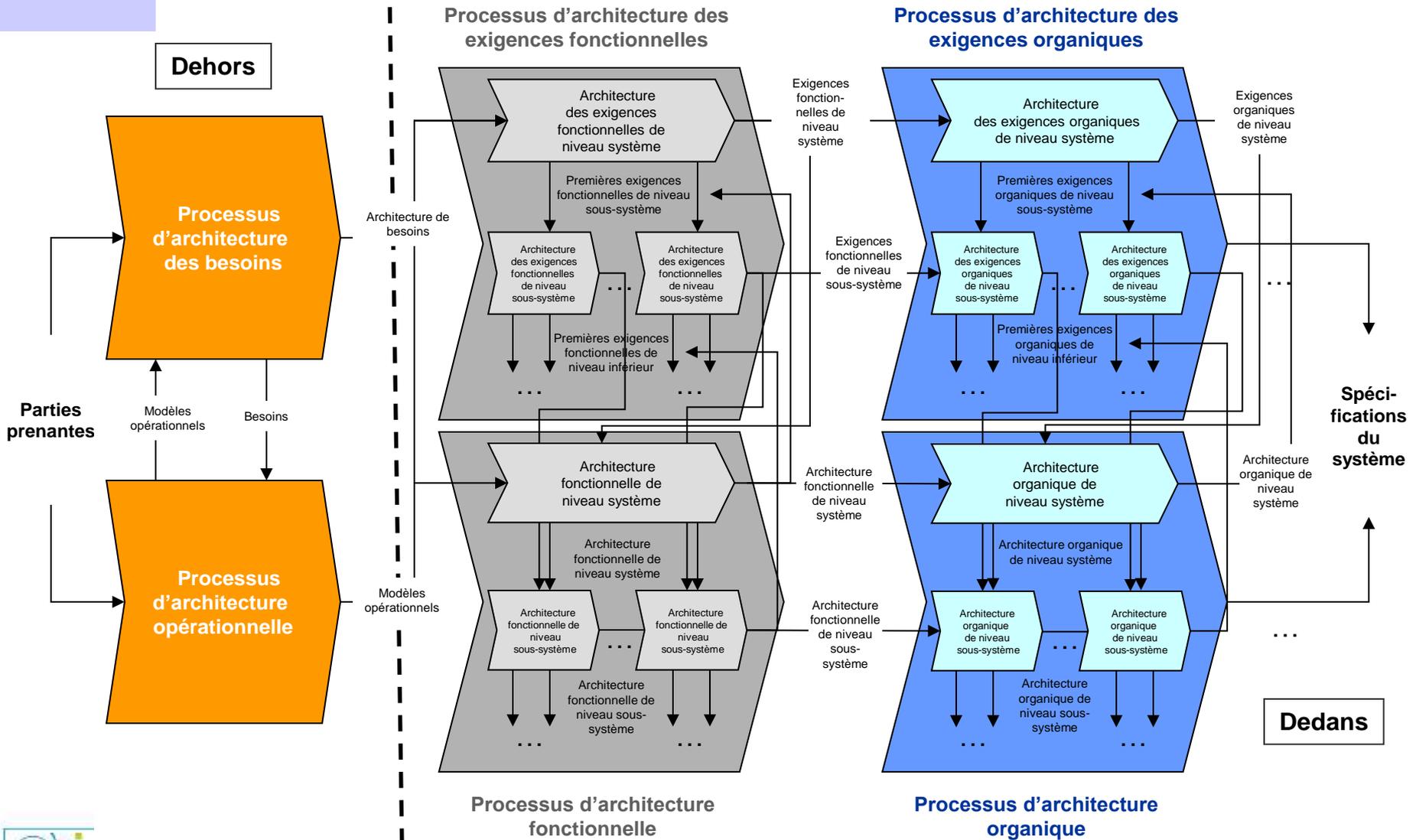


L'**extérieur** d'un **système** est l'ensemble des **systèmes externes** de son environnement qui ont une influence sur le système. Une **partie prenante** est un acteur humain qui incarne un système externe, i.e. qui est légitime pour représenter le système externe considéré.

Référentiels d'architecture & visions architecturales



Panorama du processus d'architecture système



Les livrables du processus d'architecture système (1/2)


SAFRAN
 Messier-Bugatti-Dowty
EABS2 Research and Development Project

DOCUMENT TITLE: Pages number: 92

EABS2

Mémoire de formation architecture système

Enclosures: none	Date of distribution: 01/06/12 DISTRIBUTION: - MB: SE/TSR - Safran Université: Y. DE SAINTIGNON - CESAMES: D. KROB, G. MORGANTI
------------------	--

1	Julien THIBAUT	06/12	-	-	-	-	-	-	-

Is.	Name	Date	Visa	Name	Date	Visa	Name	Date	Visa
Compiled by			Checked by			Approved by			
							Reference DR40883		

© MESSIER-BUGATTI-DOWTY 2012
This document is Messier-Bugatti-Dowty proprietary.
 It shall not be reproduced or transferred without Messier-Bugatti-Dowty authorisation.


SAFRAN
 Messier-Bugatti-Dowty

DR40883-01
 Page 3

Table des matières

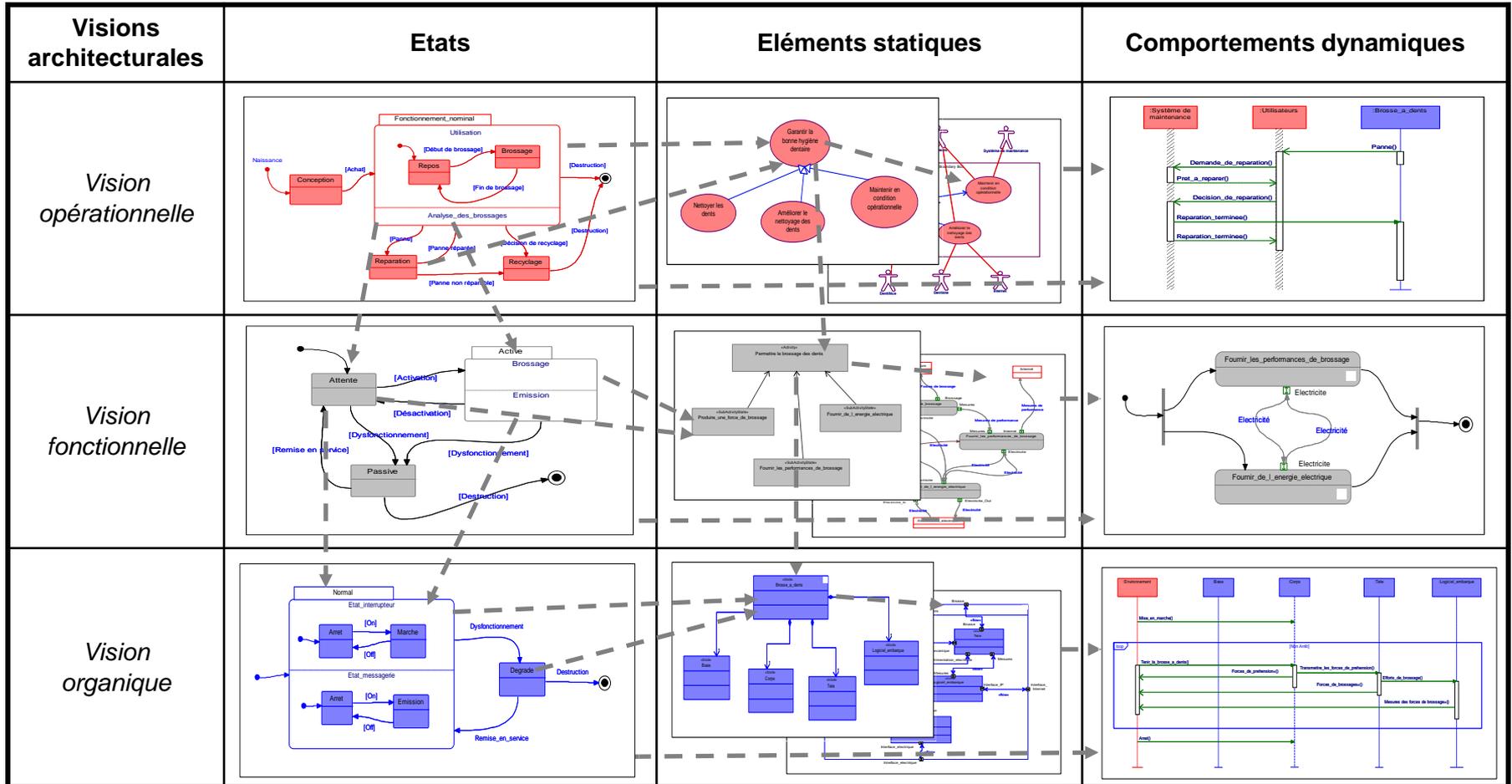
1	Présentation du projet d'architecture système	4
1.1	Introduction	4
1.1.1	Le projet EABS2	4
1.1.2	Objectifs dans le cadre de la formation	4
1.1.3	L'organisation du projet EABS2 au sein du groupe SAFRAN	5
1.1.4	Rappel du modèle architectural de la formation	7
1.2	Description de l'environnement du système existant.....	8
1.2.1	Réingénierie des besoins.....	8
1.2.2	Vision opérationnelle.....	12
1.2.3	Vision fonctionnelle.....	19
1.2.4	Vision organique.....	27
1.3	Objectifs et périmètre du projet d'architecture système.....	42
1.4	Organisation du projet d'architecture système	47
1.4.1	Plateau.....	47
1.4.2	Support.....	47
1.4.3	Intégration.....	47
1.4.4	Equipements.....	47
1.4.5	Éléments de macro-planning.....	49
2	Démarche d'architecture système.....	51
2.1	Recueil des besoins.....	51
2.2	Définition de l'architecture opérationnelle.....	59
2.2.1	Carte de l'environnement.....	59
2.2.2	Cycle de vie.....	59
2.2.3	Contextes opérationnels.....	59
2.2.4	Cas d'utilisation.....	80
2.2.5	Scenari opérationnels.....	81
2.3	Définition de l'architecture fonctionnelle et des exigences fonctionnelles.....	85
2.3.1	Evolution des fonctions du système.....	85
2.3.2	Fonctionnement du système.....	87
2.3.3	Exigences fonctionnelles.....	89
2.4	Définition de l'architecture organique et des exigences organiques.....	71
2.4.1	Comparaison entre l'existant hydraulique et la cible électrique.....	71
2.4.2	Comparaison entre les options de la cible.....	71
2.4.3	Architecture organique de l'actionneur.....	79
2.4.4	Architecture organique du calculateur.....	80
2.5	Synthèse par rapport au cadre d'architecture de la formation.....	84
3	Analyses dysfonctionnelles.....	85
3.1	Prise en compte de la sûreté de fonctionnement au niveau opérationnel et fonctionnel.....	85
3.2	Prise en compte de la sûreté de fonctionnement au niveau organique.....	85
3.2.1	Pertes de freinage.....	85
3.2.2	Freinage intempestif.....	88
3.2.3	Freinage résiduel.....	89
3.2.4	Perte de protection anti-dérapiage.....	90
3.2.5	Résumé.....	90
3.3	Analyse de l'indépendance des cas de défaillance.....	91
4	Conclusion : retour d'expérience sur le déroulement du projet d'architecture système.....	92

Livrable type : un dossier d'architecture système

Les livrables du processus d'architecture système (2/2)



Ne pas confondre architecture système et langage de modélisation !



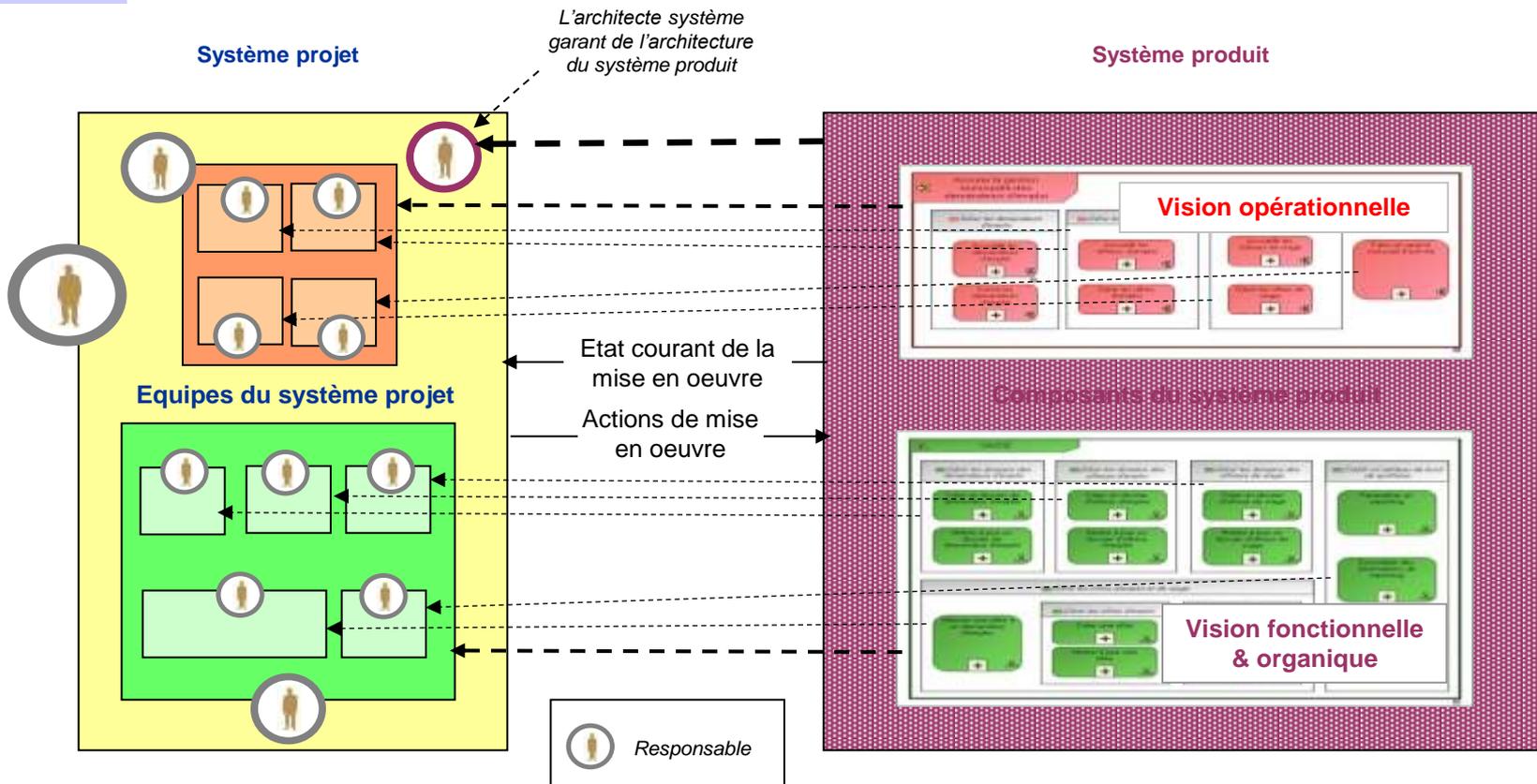
Livrable type : un modèle d'architecture système (décrit ici en SysML)



Table des matières

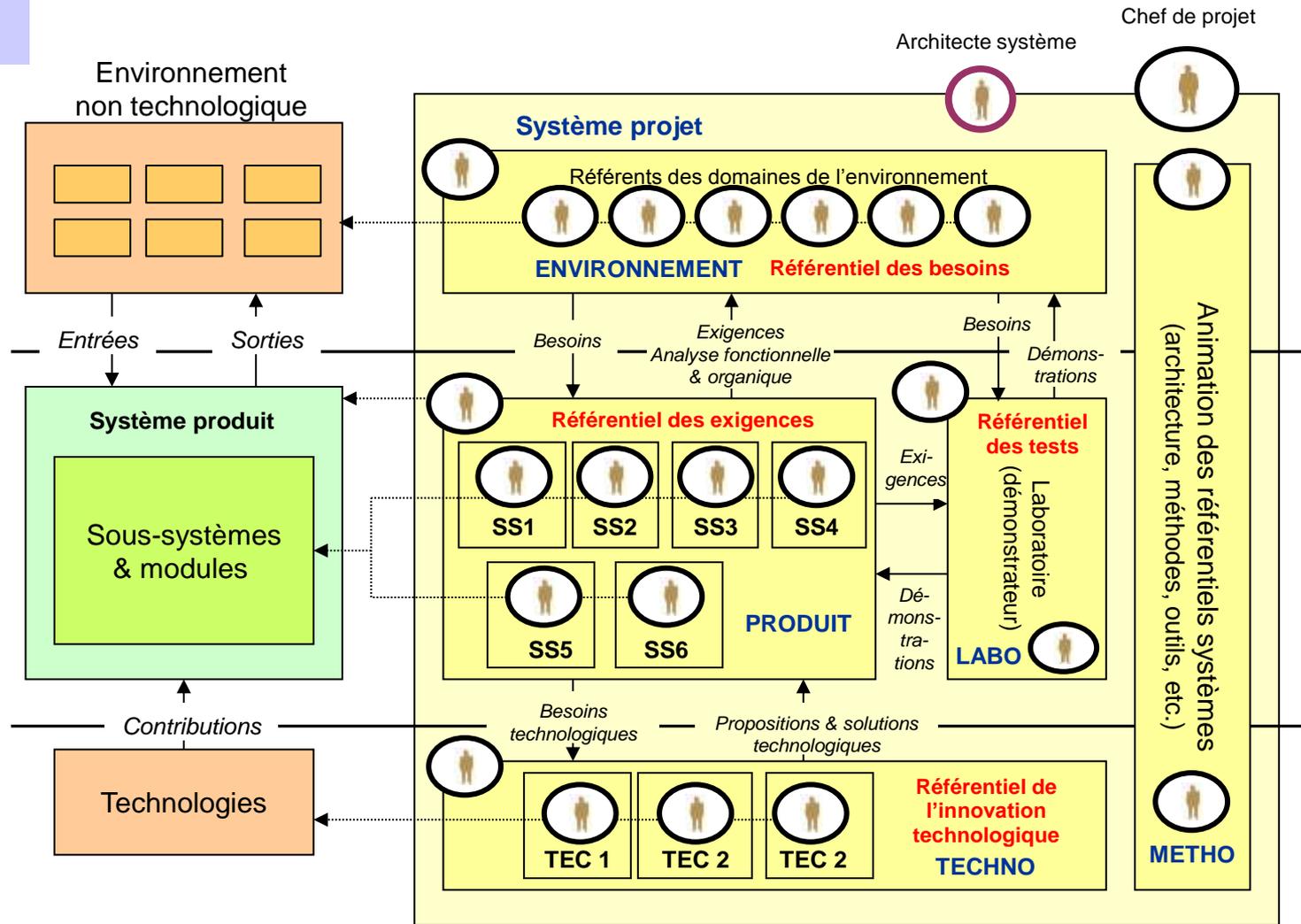
- **1 : Les enjeux de la maîtrise de la complexité**
- **2 : Les fondements de l'architecture système**
- **3 : Les impacts organisationnels de l'approche système**

Règle clef 1 : chaque composant architectural du système produit doit être pris en charge par le système projet



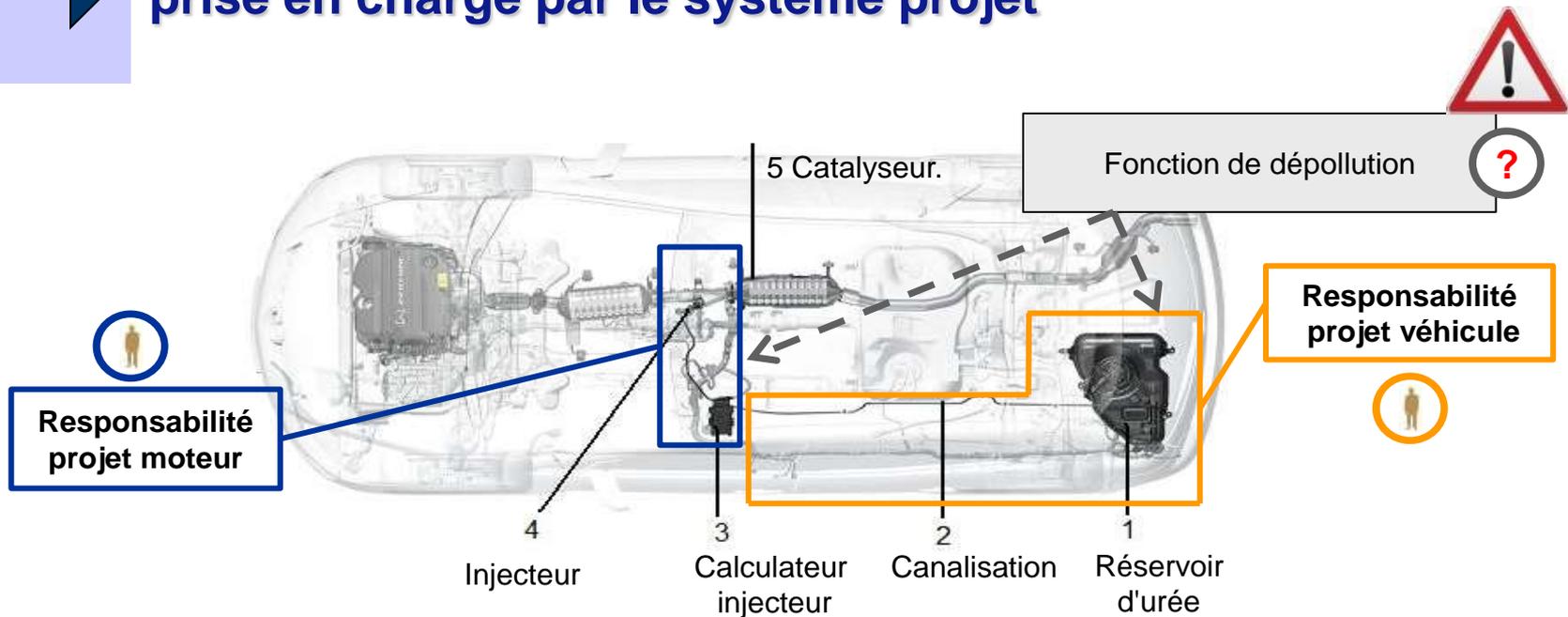
Pour avoir un fonctionnement optimal du système projet, tout **composant architectural** du **système produit** doit se projeter dans l'**architecture** du **système projet** et être pris en charge par un **responsable unique**

Exemple type : l'architecture d'un système projet



Exemple de découpage organisationnel d'un projet de R&D système

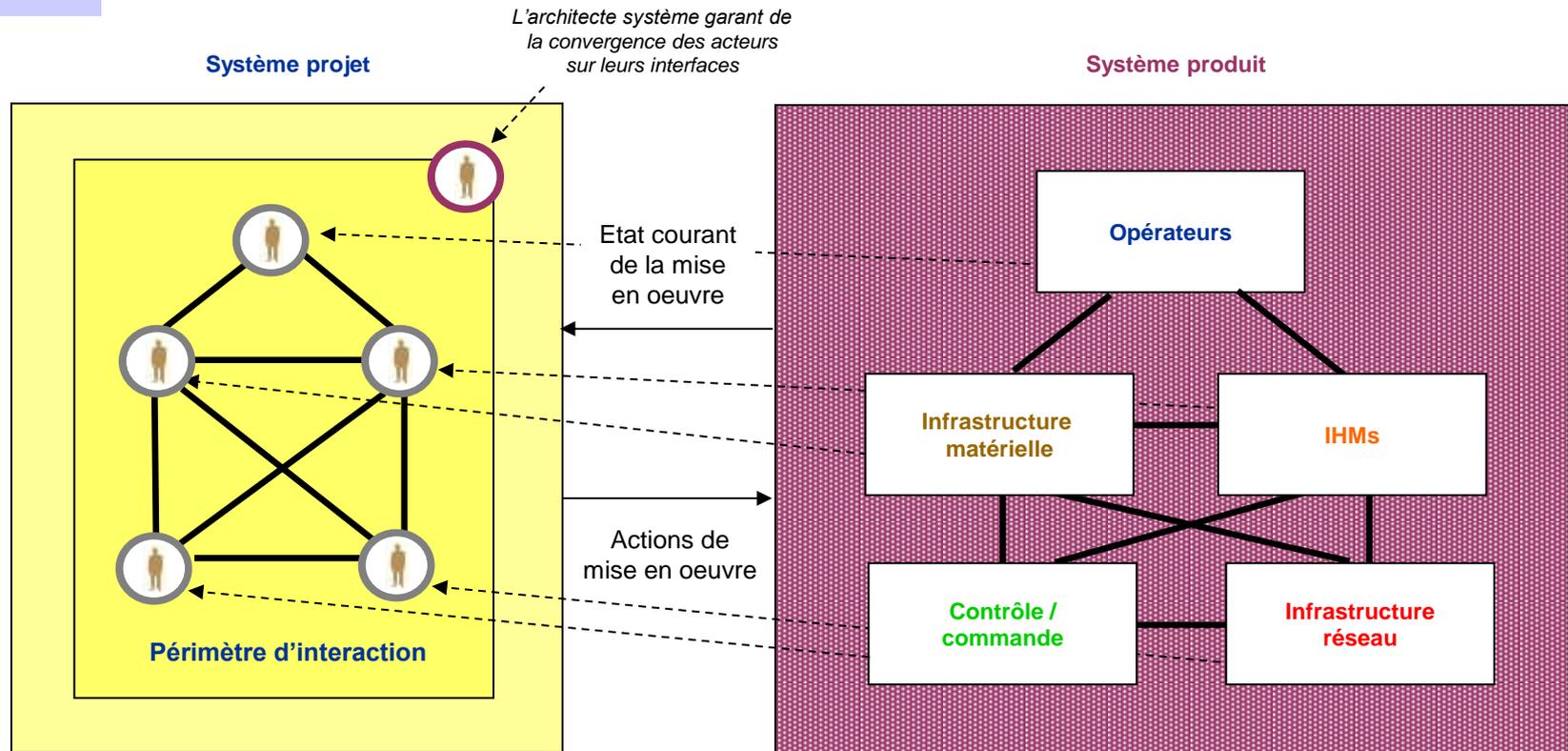
Contre-exemple type : une fonction produit transverse non prise en charge par le système projet



Un constructeur automobile cherche à implémenter une **technologie de dépollution** par injection d'urée dans les gaz d'échappement. La fonction de dépollution est arbitrairement découpée en deux sous-systèmes de responsabilité projet véhicule et moteur sans qu'un rôle de responsable de la fonction transverse ait été défini. Bilan à 5 ans : un **dépassement de 5 M€** et un système non endurant sur le cycle de vie

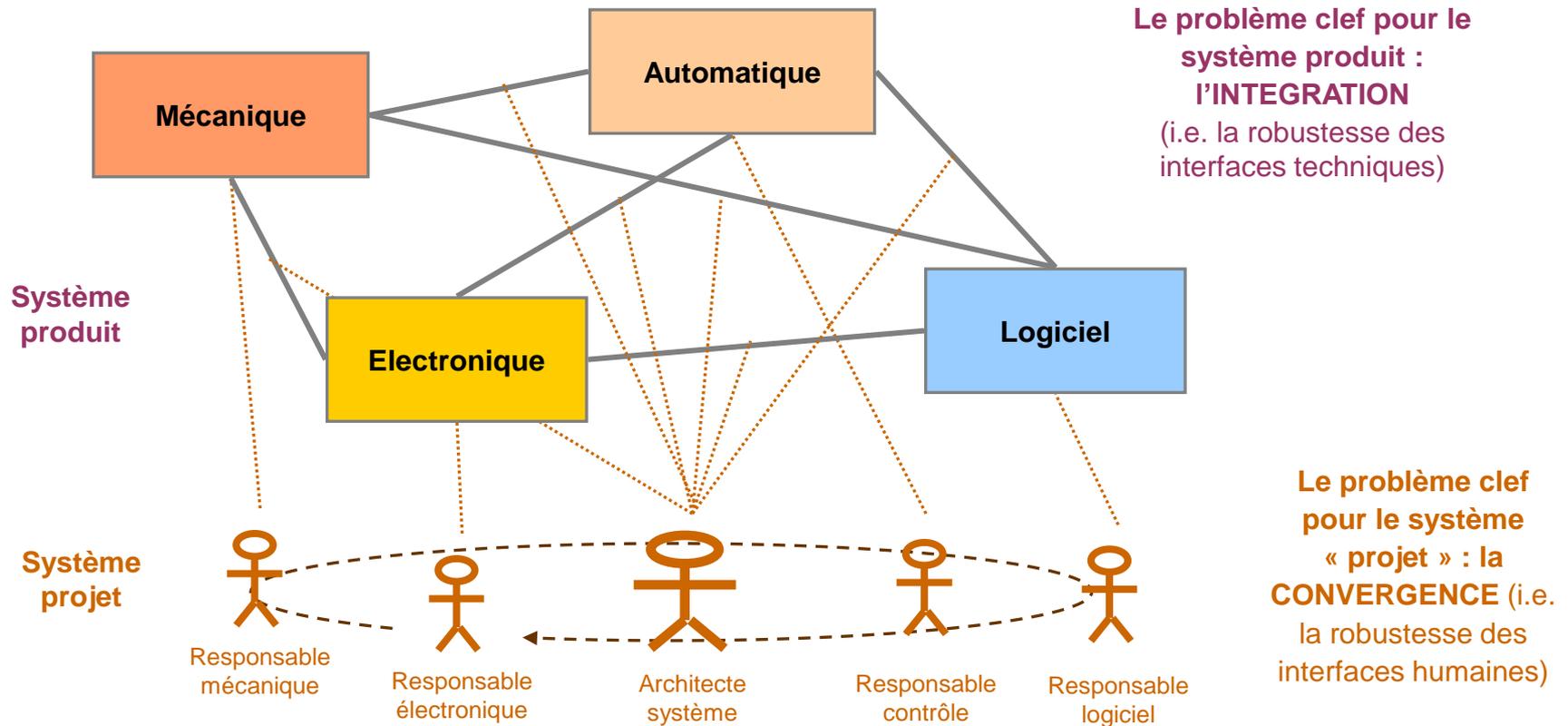
Conclusion : l'absence de **responsable** de la **fonction transverse de dépollution** n'a pas permis **l'intégration correcte** des deux sous-systèmes

Règle clef 2 : la maîtrise des interfaces techniques passe toujours par la maîtrise des interfaces projets



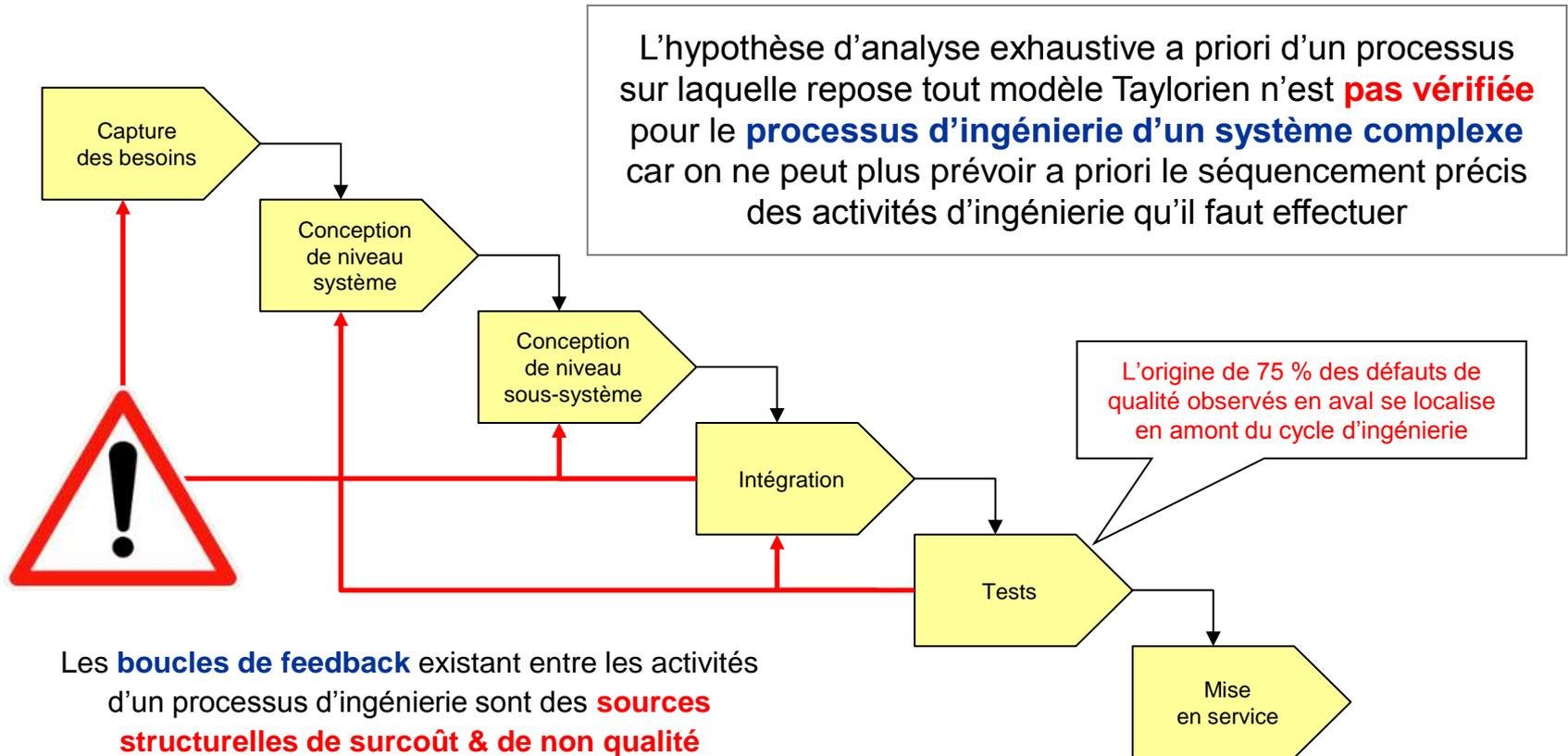
Toute **interface** de l'architecture opérationnelle, fonctionnelle ou organique du **système produit** se projette à l'identique dans une **interface humaine du système projet**, ce qui induit des logiques de **convergence d'acteurs** qui doivent toujours être pilotées en tant que telle par des architectes-facilitateurs au sein du **système projet**

Le rôle fondamental de l'architecte système



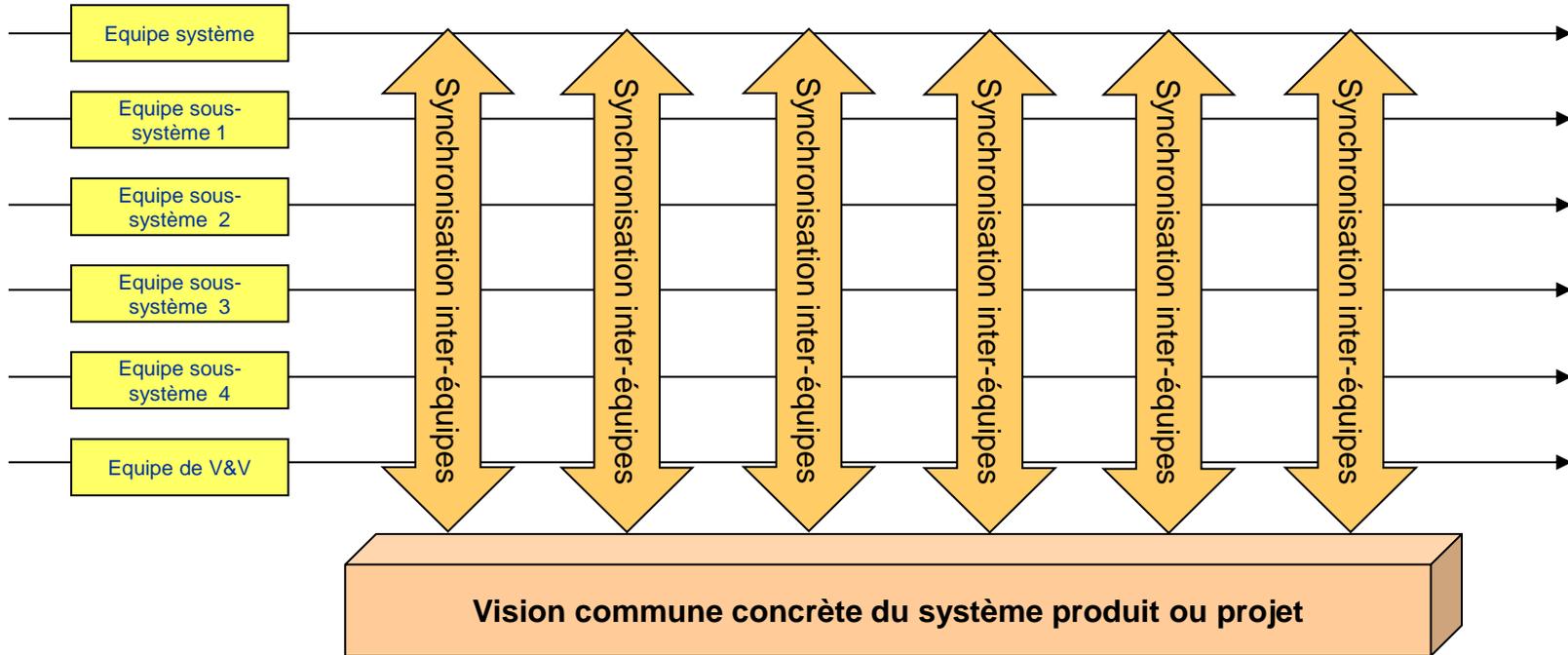
L'architecte système est à la fois le responsable de l'intégration des **systèmes techniques** (i.e. des interfaces techniques) et des **systèmes humains** (i.e. il doit faire converger les parties prenantes d'un système technique sur une vision unique)

Le modèle de la cascade n'est en effet pas adapté à l'ingénierie des systèmes intégrés hétérogènes complexes !



Le modèle de la cascade est inefficace pour les systèmes intégrés complexes en raison du trop grand nombre de **boucles de feedbacks** existants entre les activités d'ingénierie

Le modèle de l'architecture collaborative



Le **modèle d'architecture collaborative** repose fondamentalement sur la **capacité à synchroniser en permanence** les parties prenantes d'un projet système autour d'une **vision commune concrète** du **système produit** (qui peut être incarnée par un modèle systémique, une maquette numérique, etc, ...) ou du **système projet** (typiquement un planning)

Exemple de supports de synchronisation (1/2)

Définition partagée d'une architecture d'exigences

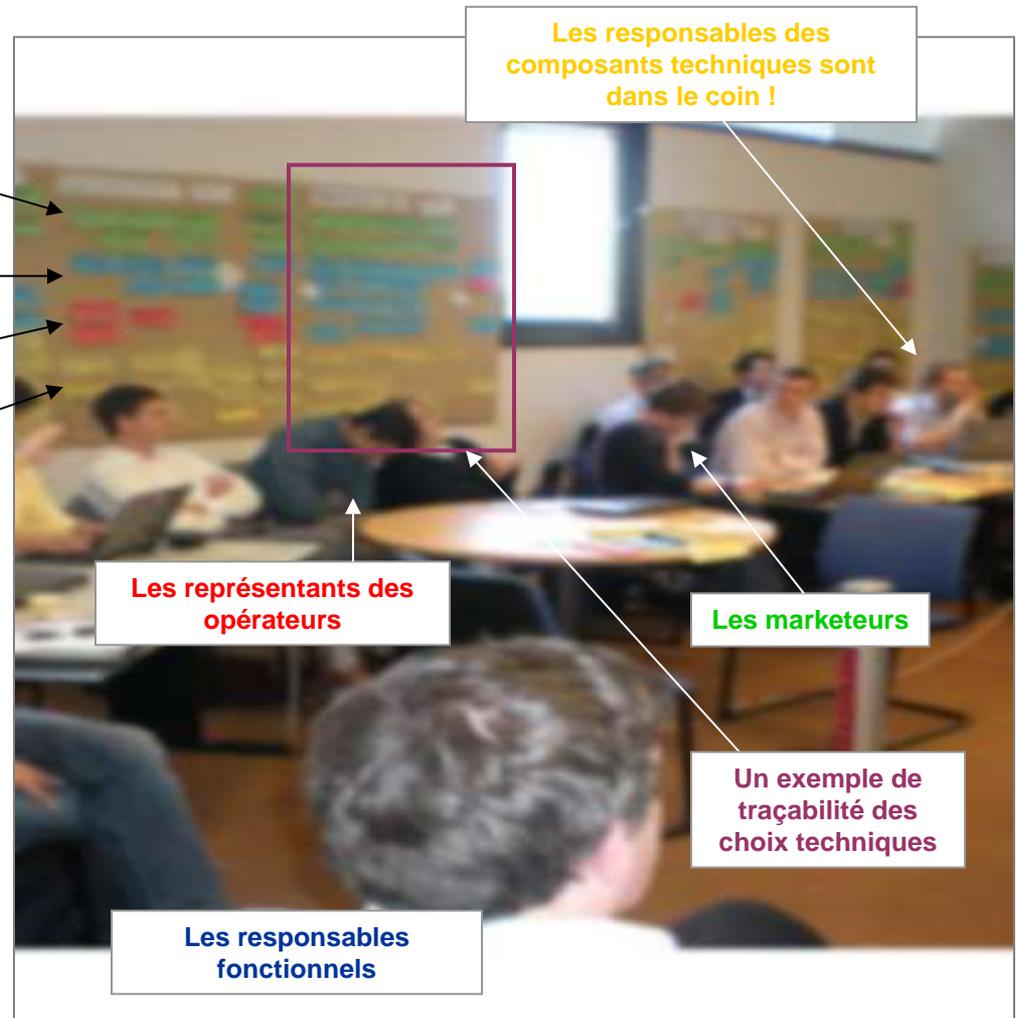
Niveau Vert : les besoins externes

Niveau Bleu : les fonctions

Niveau Rouge : les opérateurs humains

Niveau Jaune : les organes techniques

La première étape de l'atelier a consisté à identifier en mode collaboratif avec une **méthode top-down** tous les **impacts d'un ensemble de besoins** le long d'une architecture système structurée ici en 4 "vues" (clients, fonctions, opérateurs humains, organes techniques)

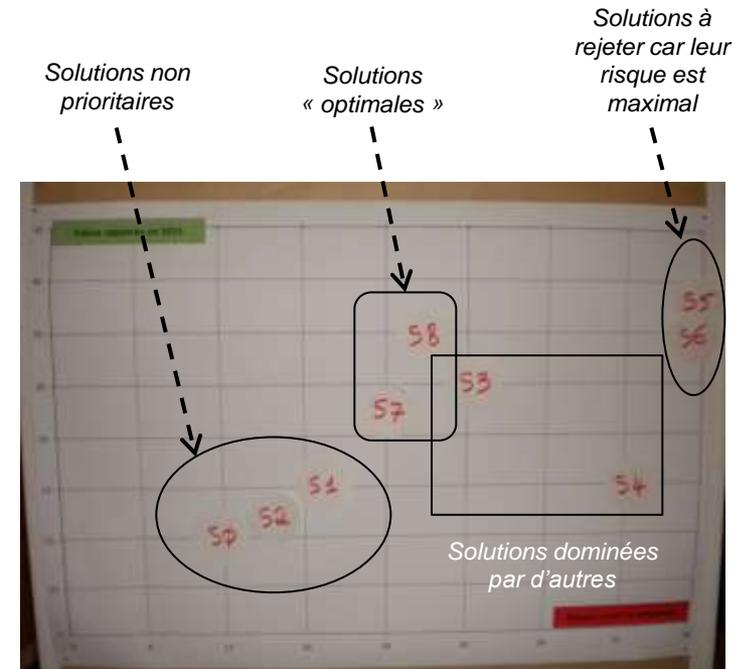


Exemple de supports de synchronisation (2/2)

Priorisation partagée de choix architecturaux



Evaluation collective de chacune des solutions organiques à étudier par rapport à des critères d'impacts positifs et négatifs



Evaluation des solutions organiques proposées et identification de « clusters » de solutions

Un **atelier collaboratif de priorisation** permet de sécuriser les choix à effectuer entre différentes **solutions fonctionnelles ou organiques** en impliquant l'ensemble des parties prenantes dans le processus de décision



Éléments d'architecture système

FIN