



# ***Algorithme et programmation***

***Mathématiques 2nde  
Module No 23***

***Savoir poser et analyser un problème :  
les méthodes de conception***





# Programme

- **CM#4 Savoir poser et analyser un problème : les méthodes de conception**
- Divers niveaux de réflexion
- Méthode de conception et méthode de réalisation
- Méthode de conception
  - Que faire ?
  - Notion de processus
  - Notion de modèle
  - Modéliser les processus
  - Méthodes, langages et notations.
  - Une méthode : SADT/IDEF
  - Exemples de modélisation de processus.





# Quelques questions

Quel intérêt ont les modèles pour comprendre les organisations ?

Comment représenter les activités de gestion et les activités métier au sein de l'organisation ?

Quelles caractéristiques des activités recenser pour pouvoir construire un nouveau système de Gestion ?

Qu'est-ce que la méthode SADT-IDEF ?





# Deux niveaux de réflexion

- Deux niveaux de réflexion dans la démarche de réalisation des logiciels.
- En premier lieu, approche d'une solution globale en réponse au besoin formulé ou au besoin imaginé : modélisation de l'organisation, identification et description formelle des données et des processus). Cette démarche est dite « de conception » (autrefois dite « d'analyse ») .
- En second lieu, au niveau de chaque unité logique identifiée et délimitée au sein de cette solution globale, démarche d'écriture d'un programme, dite de « de réalisation » (autrefois dite « programmation »).
- A ce dernier niveau on parle plutôt aujourd'hui de **processus d'ingénierie de logiciel** que de méthode de programmation.





# La source

- Le besoin exprimé dans un cahier des charges (voir chapitre 2)





# Deux familles de méthodes

- **Méthodes de conception** pour le premier niveau :
  - Objectif : analyser les processus (BPM); construire le diagramme des flux; construire le dictionnaire des données; identifier les dépendances fonctionnelles; modéliser l'ensemble processus-données; préparer la phase de réalisation en construisant les bases de données et en découpant en unités fonctionnelles (applications), en unités de traitement (jobs, transactions) et en unités de programmation; intégrer et valider les résultats de la phase de réalisation.
  
- **Méthodes de réalisation** pour le second niveau
  - Objectif : analyser de manière détaillée et structurée chaque unité de programmation identifiée; exprimer la solution du problème sous forme algorithmique, indépendamment du langage choisi; supporter la phase d'écriture et de test dans ce langage; définir les règles de maintenance et d'assurance qualité.





# Retour sur la démarche de programmation

***Reprenons les étapes de la démarche suivie dans le chapitre précédent :***

## → LES ETAPES DE CONCEPTION D'UN LOGICIEL

1. Identifier un besoin
2. Analyser le contexte du besoin : processus et données
3. Elaborer une solution globale pour répondre au besoin
4. Choisir un niveau de langage (Symbolique, évolué L3G, L4G)
5. Décomposer la solution en opérations élémentaires jusqu'au niveau où il devient possible d'associer à l'opération une instruction du niveau de langage choisi
6. Rédiger et tester le pseudo-code
7. Choisir le langage
8. Rédiger les programmes
9. Compiler ou interpréter (traduire le langage choisi dans le langage de base de l'ordinateur)
10. Vérifier que les programmes satisfassent bien le besoin exprimé à l'aide d'un jeu d'essai
11. Documenter le programme
12. Livrer le logiciel à l'utilisateur (Formation, Assistance)

Niveau  
conception  
(auquel est  
consacré ce  
chapitre)

Niveau  
réalisation



# → De la conception à la réalisation

→ **Concevoir** : Une approche globale du problème posé :

Quel résultat recherchons-nous ? Quelles informations devons-nous manipuler pour obtenir ce résultat ?

→ Une description précise des données et de leurs propriétés

Quel traitement devons-nous appliquer aux données pour obtenir le résultat convoité ?

→ Une description précise d'un processus en tant qu'enchaînement de traitements élémentaires

→ **Réaliser** : une démarche précise pour construire un bloc logiciel (programme, composant, service) assumant l'un des traitements élémentaires identifiés.

→ Dans ce chapitre nous traitons de la conception de nouveaux **processus** (en vue de leur automatisation).







# Qu'est ce qu'un processus ?

- Une combinaison d'opérations, souvent enchaînées, ...
- ... en vue d'une **finalité**.
- Mettant en œuvre des ressources diverses (**ressources** humaines, ressources matérielles, informations).
- Faisant le plus souvent intervenir plusieurs acteurs selon un axe « **transversal** » par rapport au découpage de l'organisation.
- Génératrice de **résultats** caractérisés par une création de **valeur**.
- Susceptible d'être évaluée en fonction de divers indicateurs de **performance** :
  - Pour le client final : Utilité/coût
  - Pour l'entreprise :  $\frac{\text{Qualité*performances}}{\text{ressources consommées}}$



# → Qu'est ce qu'un processus ?

→ *Un processus est une combinaison d'opérations interdépendantes et récurrentes, faites avec des moyens et selon des règles, en vue d'une finalité.*

→ **Exemples :**

- Mettre au point un nouveau produit
- Accueillir et traiter un client
- Exploiter et entretenir un équipement tout au long de son cycle de vie

→ **Notions clefs :**

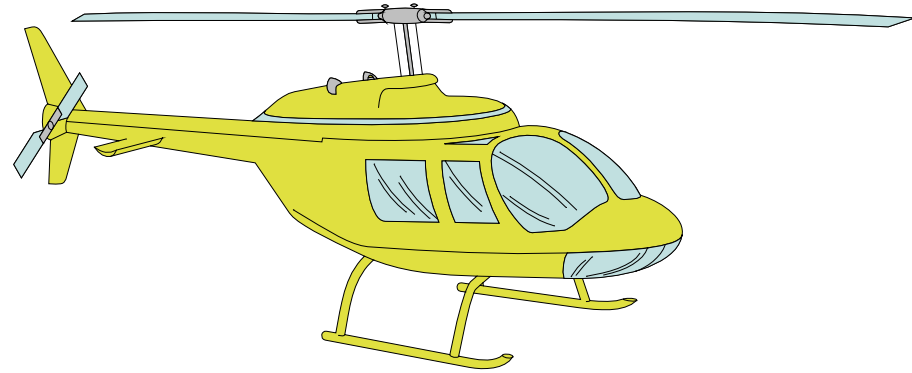
- Finalité (objectifs),
- Activités (Tâches, opérations, travaux, actions),
- Ressources (Acteurs, moyens)
- Contraintes (Contexte, règles, temps)
- *Comment décrire un processus ?*
  - A l'aide d'un modèle.



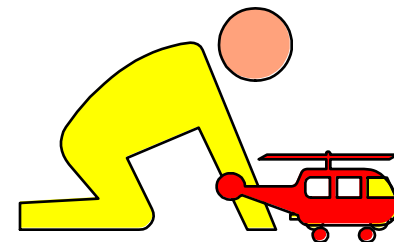


# Modéliser pour concevoir

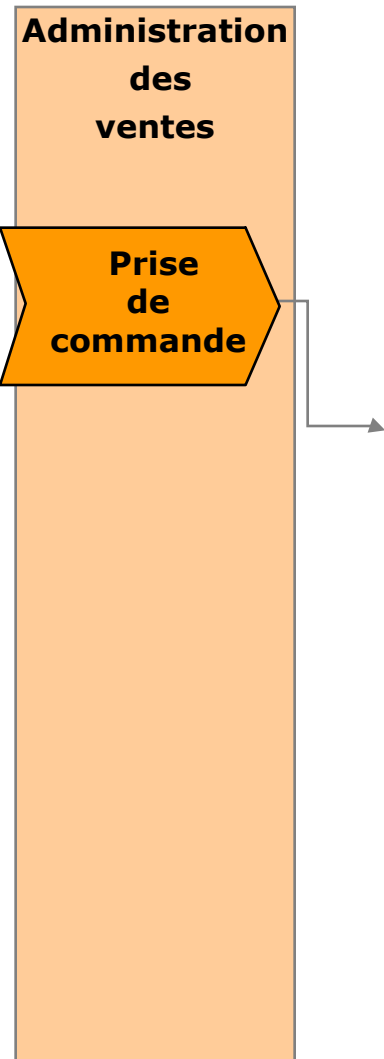
- Un **modèle** est une représentation abstraite des composants et des relations d'un processus au sein d'un système.
- Un modèle est une image de la réalité.
- Un modèle doit être simplificateur, unificateur et prédictif.
- L'objet à modéliser doit être décrit dans sa totalité :
  - Fonctionnant.
  - Evoluant.



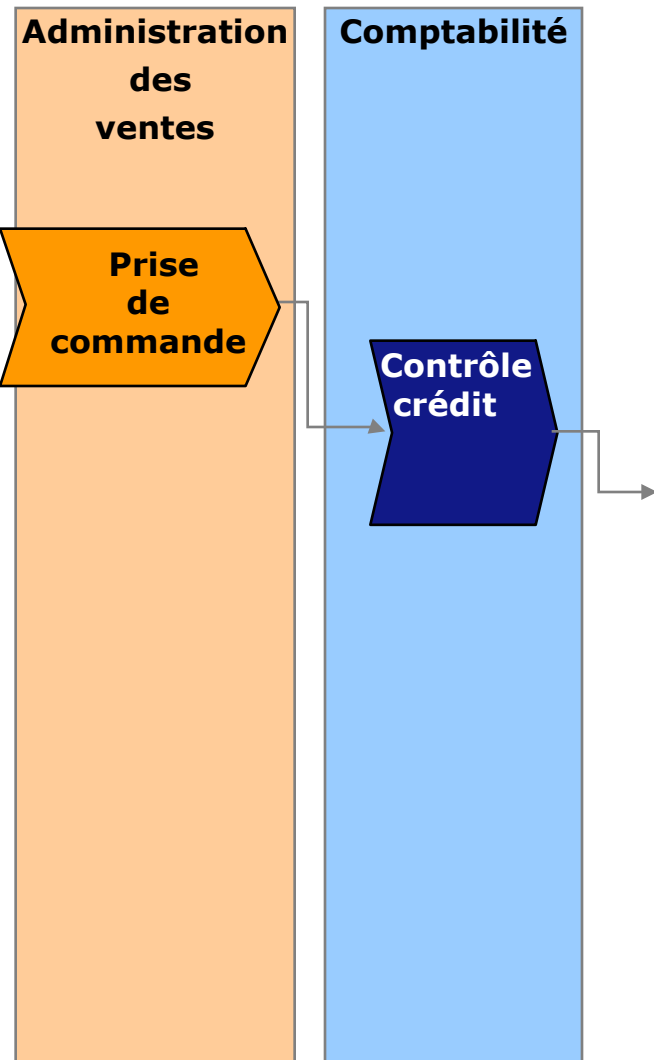
**MODELE = Image de la réalité qui la rend plus accessible**



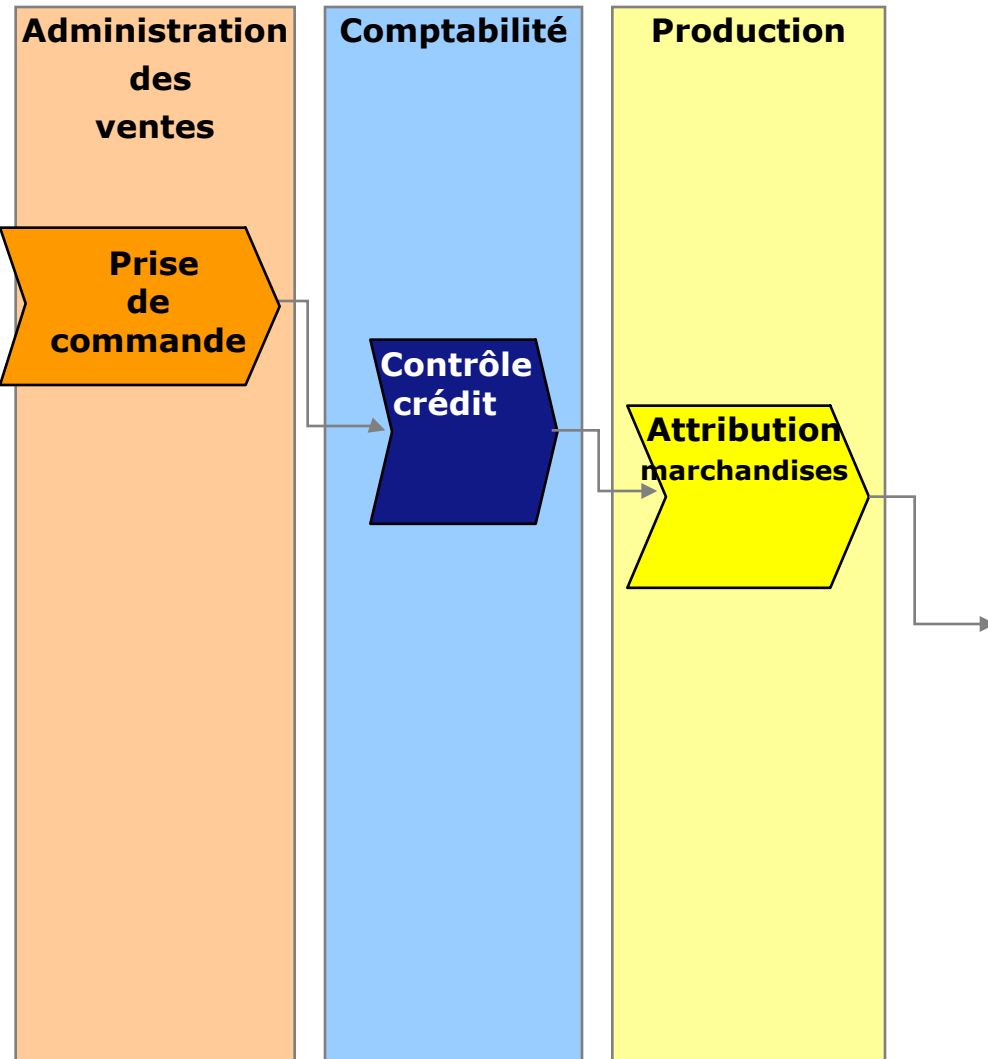
# → Exemple d'un modèle de processus



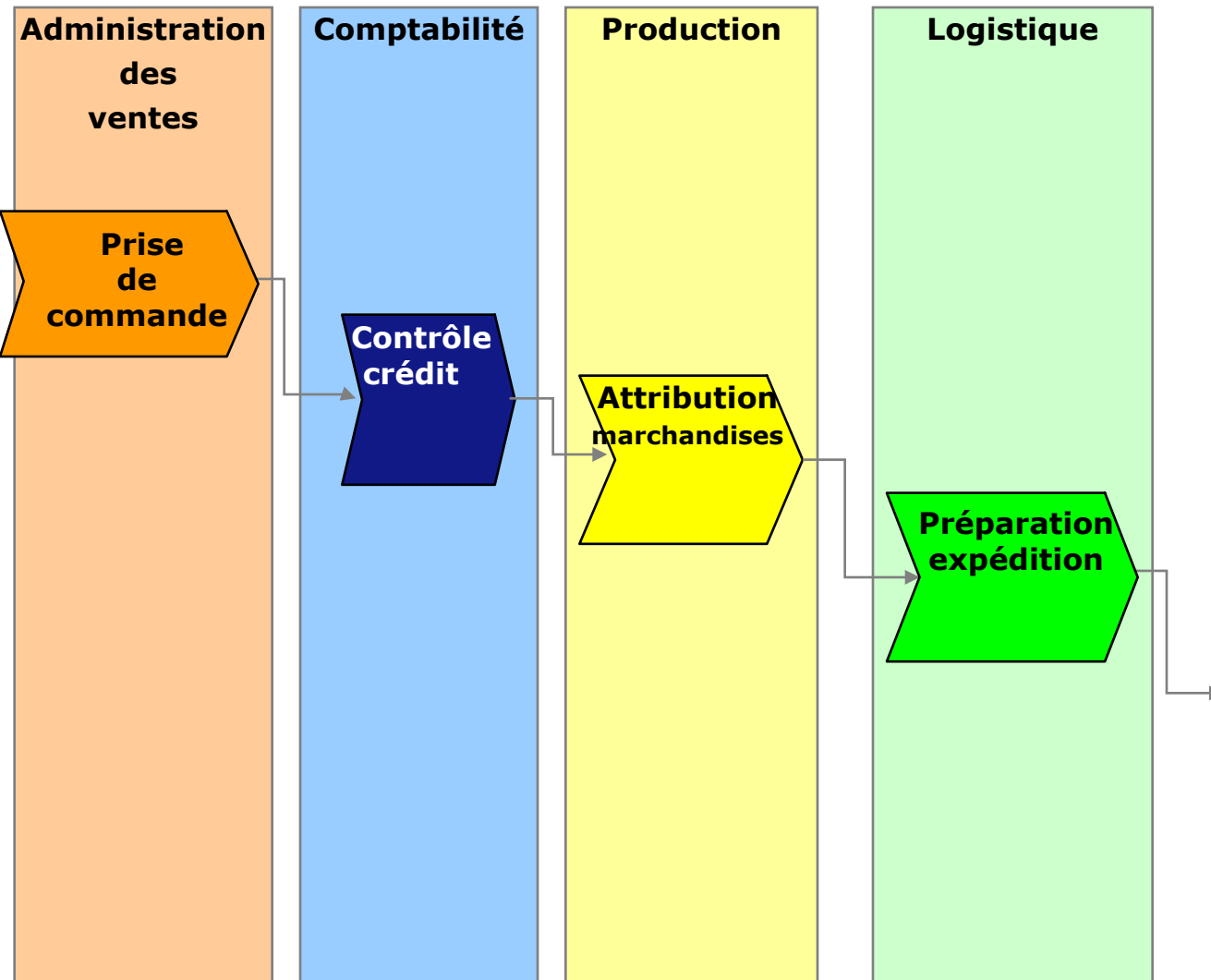
# → Exemple d'un modèle de processus



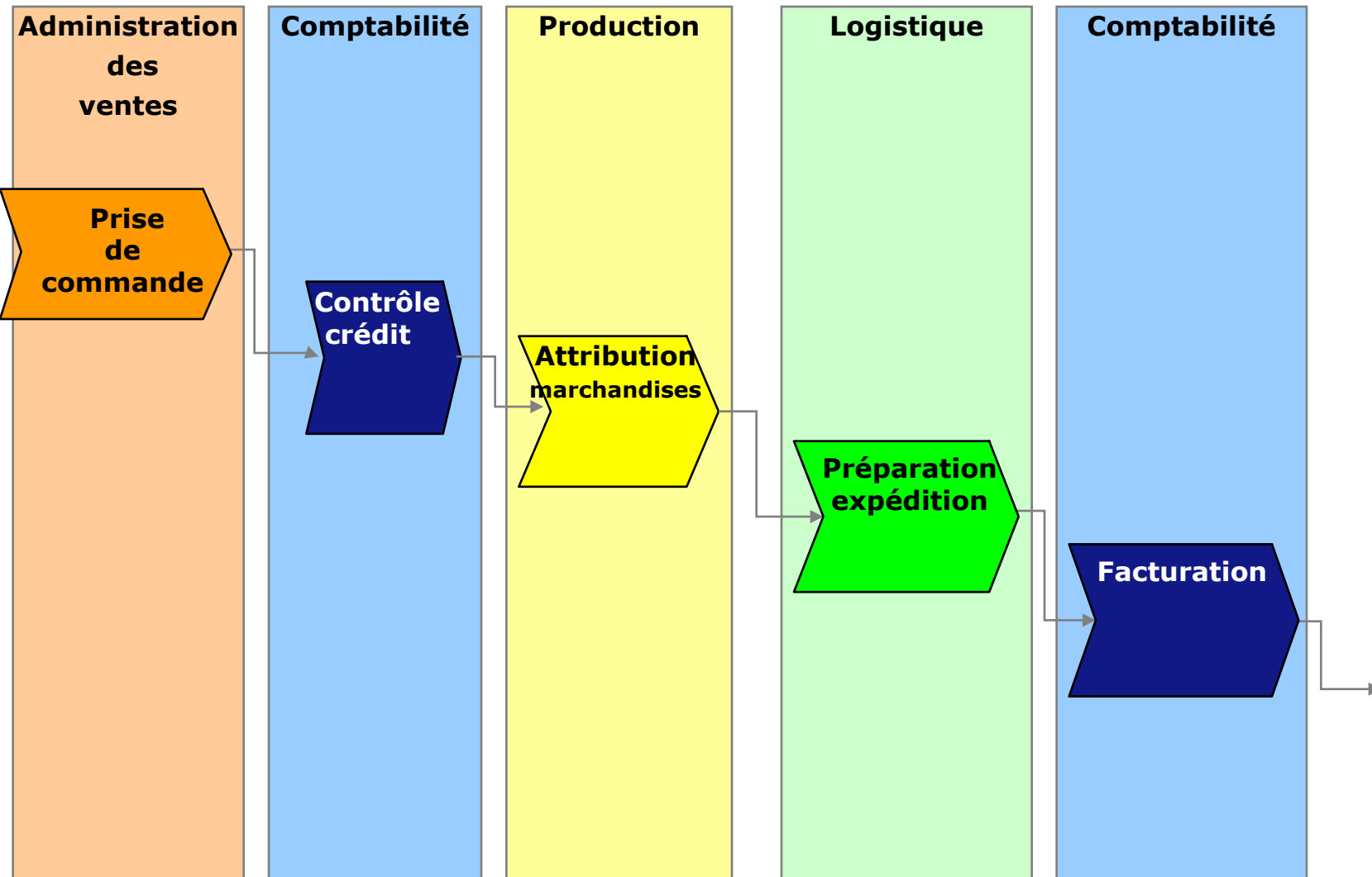
# → Exemple d'un modèle de processus



# → Exemple d'un modèle de processus

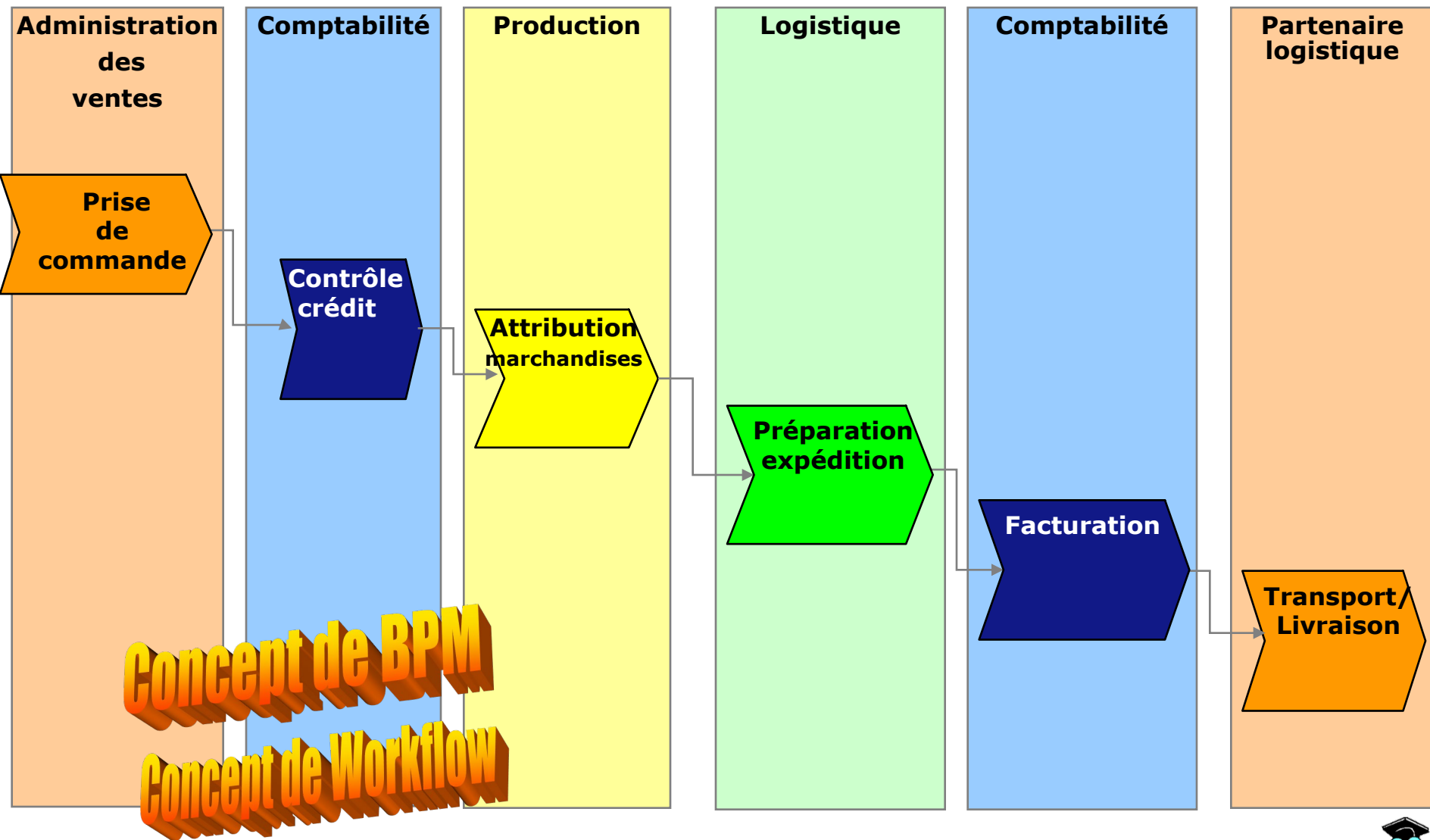


# → Exemple d'un modèle de processus





# → Exemple d'un modèle de processus





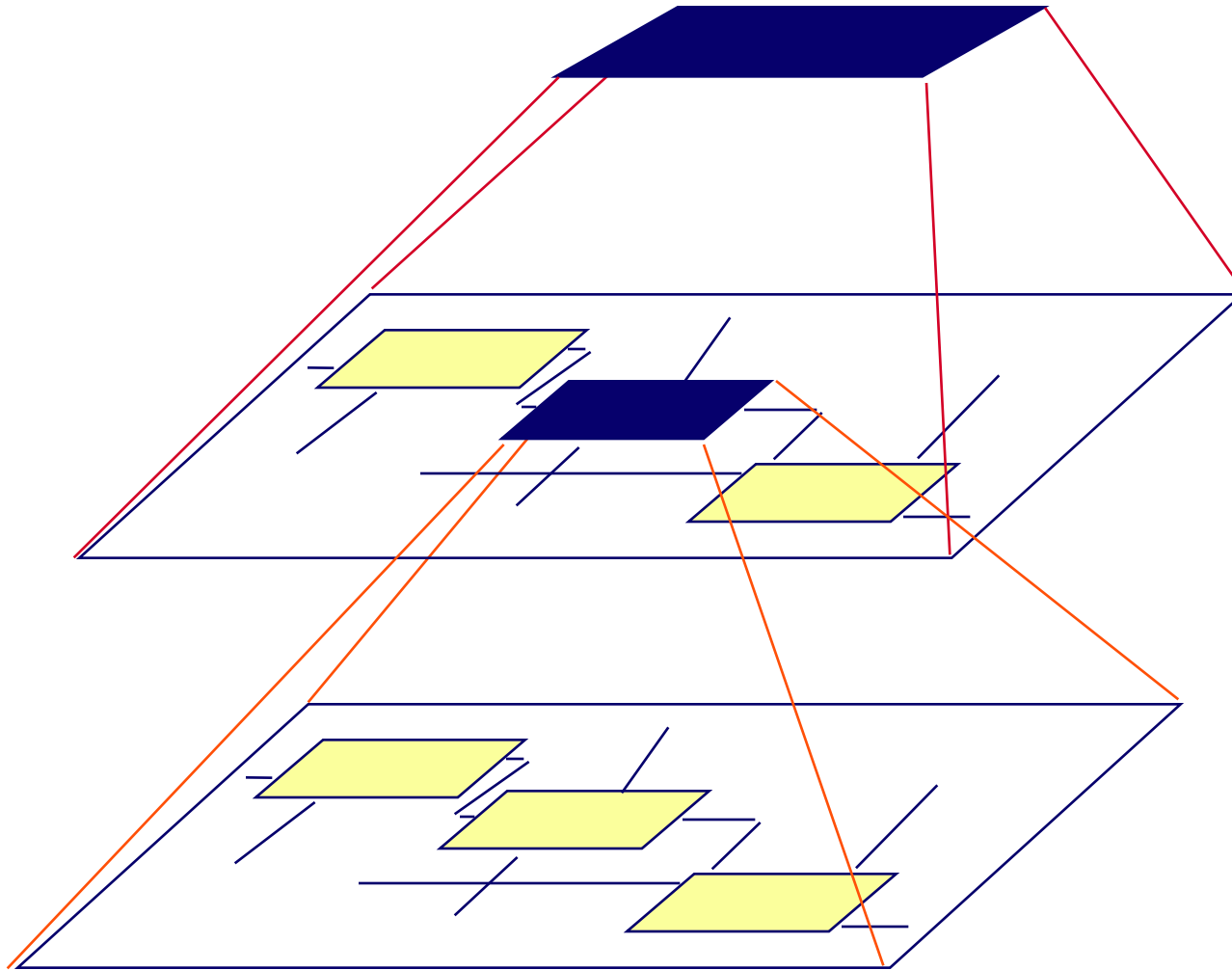
# Quelques définitions

- **Business Process Modelling** : Modélisation de processus de gestion ou de processus métier grâce à la création d'un modèle graphique.
- **Business Process Management** : L'objectif de cette démarche est d'aboutir à une meilleure vue globale de l'ensemble des processus de l'entreprise et de leurs interactions, afin d'être en mesure de les optimiser et, dans la mesure du possible, de les automatiser.
- **Workflow** : On appelle "*WorkFlow*" (littéralement "flux de travail") la modélisation et la gestion informatique de l'ensemble des tâches à accomplir, ainsi que des différents acteurs impliqués, dans la réalisation d'un processus de gestion ou d'un processus métier.





# La hiérarchie des modèles

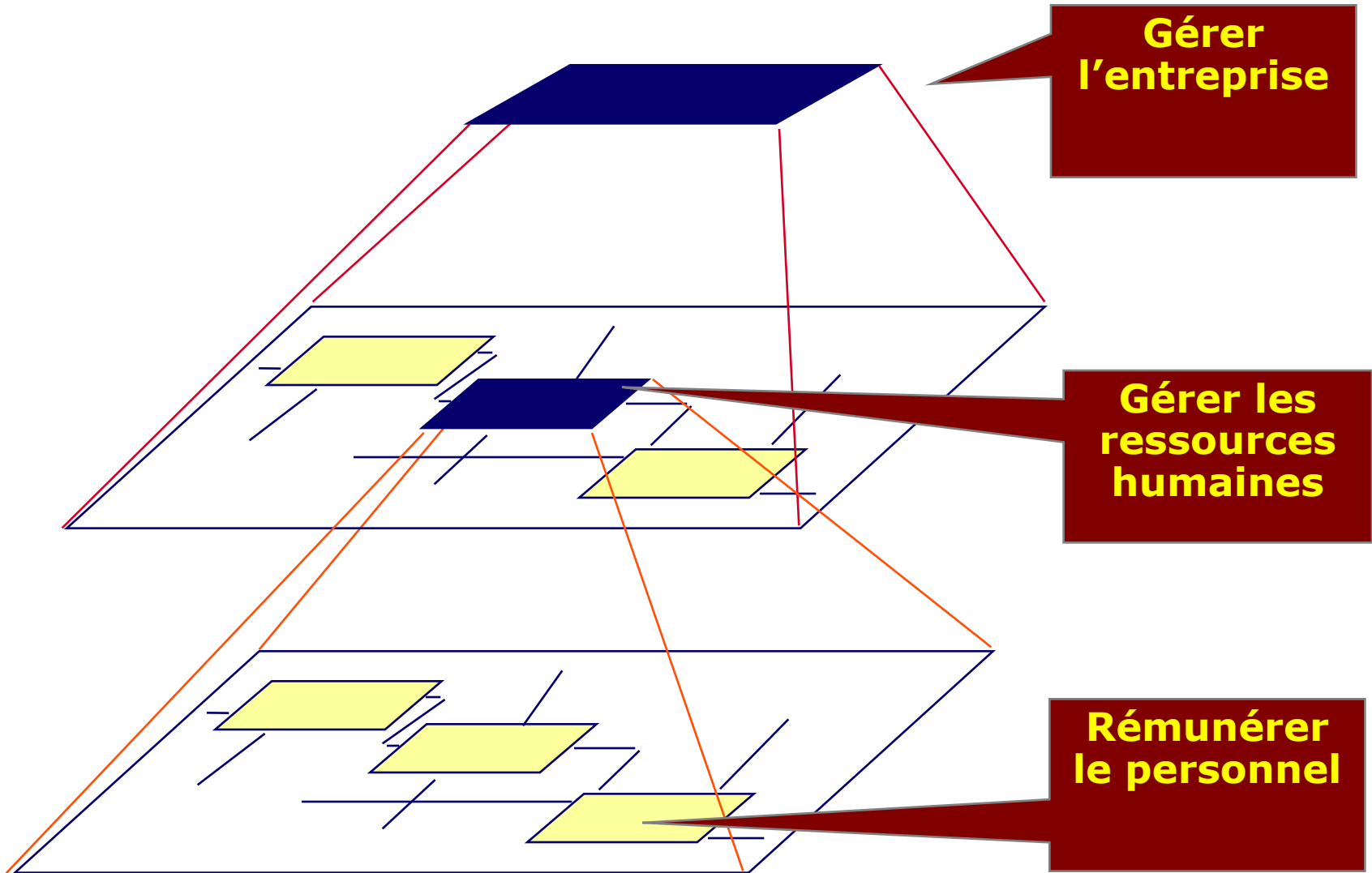


DU GENERAL  
VERS LE  
PARTICULIER  
(Top Down)





# La hiérarchie des modèles



# → Pourquoi modéliser un processus

- Pour analyser et évaluer un processus existant
- Pour concevoir un nouveau processus (ingénierie d'un processus)
- Enchaînement des deux démarches si la connaissance des processus actuels est insuffisante
- Etude de l'écart entre les deux versions (*Gap analysis*)
- Nécessité d'une démarche formalisée de description qui a pour corollaire l'**exigence d' outils méthodologiques pour l'Analyse et la Conception des Processus**



# → Pourquoi une méthode de modélisation des processus ?

## Une « méthode » pour :

- Analyser, comprendre et concevoir les processus mis en œuvre au sein de systèmes complexes;
- Communiquer les résultats de l'analyse et de la conception;
- Assurer la qualité, la précision et la complétude des spécifications;
- Répartir le travail d'équipe et coordonner les efforts;
- Assurer la direction d'un projet, maîtriser sa réalisation et son suivi.

## Méthode, langage ou simple notation ?



# → Les outils méthodologiques

- Des **méthodes** : Une méthode est une démarche structurée permettant de formaliser les étapes du développement d'un SI. Une méthode décrit les processus d'analyse et de conception. Une méthode impose un langage.
- *Merise, SADT sont des méthodes.*
- Des **langages de modélisation** : Un langage est un ensemble formel et normalisé de termes et de règles syntaxiques pour permettre aux acteurs d'un projet de conception d'un SI de communiquer sans aucune ambiguïté. Un langage implique une notation.
- *UML est un langage de modélisation*
- *Attention à ne pas confondre langage de modélisation et langage de programmation*
- Des **notations** : Une notation est un ensemble formel et normalisé de symboles pour permettre aux acteurs d'un projet de conception de représenter les éléments d'un SI sans aucune ambiguïté.
- *BPMN est une notation*





# Qu'est-ce qu'une méthode d'analyse et de conception ?

- « *Processus discipliné qui engendre un ensemble de modèles décrivant les différents aspects d'un système d'information, en utilisant une certaine notation bien définie* » (d'après Booch, 94)
- **Analyse** : Etude du problème
  - Etudier le système existant
  - Comprendre les besoins : diagnostiquer
  - En déduire le niveau conceptuel : donner une vision fonctionnelle du système
- **Conception** : Etude de la solution
  - Proposer de nouvelles solutions organisationnelles pour améliorer le processus
  - Evaluer le 'gap' » (lors du déploiement, il sera nécessaire de combler l'écart entre processus ancien et processus nouveau au moyen de « *gap killing sessions* »
  - Préparer le niveau technique de réalisation







# Les méthodes d'analyse et de conception

<b>Courants</b>	<b>Analyse structurée</b>	<b>Systémique</b>	<b>Objet</b>
<b>Origine</b>	Anglo-Saxone	Européenne	Approche Objet
<b>Modélisation</b>	Double vision d'une décomposition arborescente des processus et des données	Modélisation des données et des traitements	Objet = Classe + Instance + Héritage + Propriété + « Méthode »
<b>Caractéristiques</b>	Enchaînement des tâches Flux de données	Séparation données / traitements	Implémentation Orientée Objet
<b>Exemples</b>	<b>SADT/IDEF</b> DEMARCO (SA) YOURDON (SD)	<b>MERISE</b> AXIAL IDA	Méthodes fondées sur le langage <b>UML</b>  OOA HOOD OOM (Merise + objet)



# → Méthodes, langages et notations courantes

- La méthode **Merise**, initiée en 1977, a été longtemps la référence dans l'espace francophone. Elle est aujourd'hui dépassée, hormis son MCD (Modèle Conceptuel de Données) conforme au modèle entité-association et directement applicable dans l'univers des bases relationnelles de Chen, qui reste très utilisé
- La méthode **SADT**, devenue **IDEF** en tombant dans le domaine public, a été longtemps le No 1 mondial ayant été choisi par le DoD US et imposé aux principales autres forces armées et à leurs fournisseurs.
- L'objectif de normalisation est aujourd'hui moins ambitieux. On définit un cadre normatif au niveau du langage, chacun pouvant sur cette base construire sa méthode.
- Le langage **UML** est le nouveau standard en matière de langage de modélisation.
- La notation **BPMN**, normalisée comme **UML** par l'**OMG**, devient progressivement l'outil de communication entre les utilisateurs et les informaticiens rompus à **UML**.





# Notre choix

- Fondé sur des considérations didactiques
- **Merise** est trop ancienne. Nous n'utiliserons ultérieurement que son MCD.
- **UML** est trop complexe (13 diagrammes différents) et trop ciblé sur les besoins des informaticiens.
- **BPMN** n'est qu'une notation : pratique et d'avenir, mais elle ne nous donne pas de guide méthodologique pour analyser et concevoir.
- **SADT** (*Structured Analysis and Design Techniques*), devenue IDEF, correspond exactement à notre objectif de disposer d'une démarche structurée d'analyse et de conception.
- 737 000 articles sur SADT/IDEF référencés par Google sur Internet, avec des milliers de diagrammes.
- Au delà de la méthode d'analyse et de conception, SADT consitue un excellent outil de "remue-méninges« (*brain storming*) applicable en de nombreuses situations





- Méthode S.A.D.T. (***System Analysis and Design Techniques***) de SofTech, aujourd'hui dans le domaine public et normalisée (DoD) sous le nom de IDEF (***Integration Definition for Function Modeling***).
  - Identifier les activités.
  - Mettre en évidence les échanges :
    - Ce qui entre dans chaque activité,
    - Ce qui sort de chaque activité.
  - Recenser les contraintes (associées à la notion d'évènement) :
    - Conditions de déclenchement,
    - Tops de synchronisation.
  - Recenser les ressources mises en œuvre.





# Les sept concepts fondamentaux de la méthode

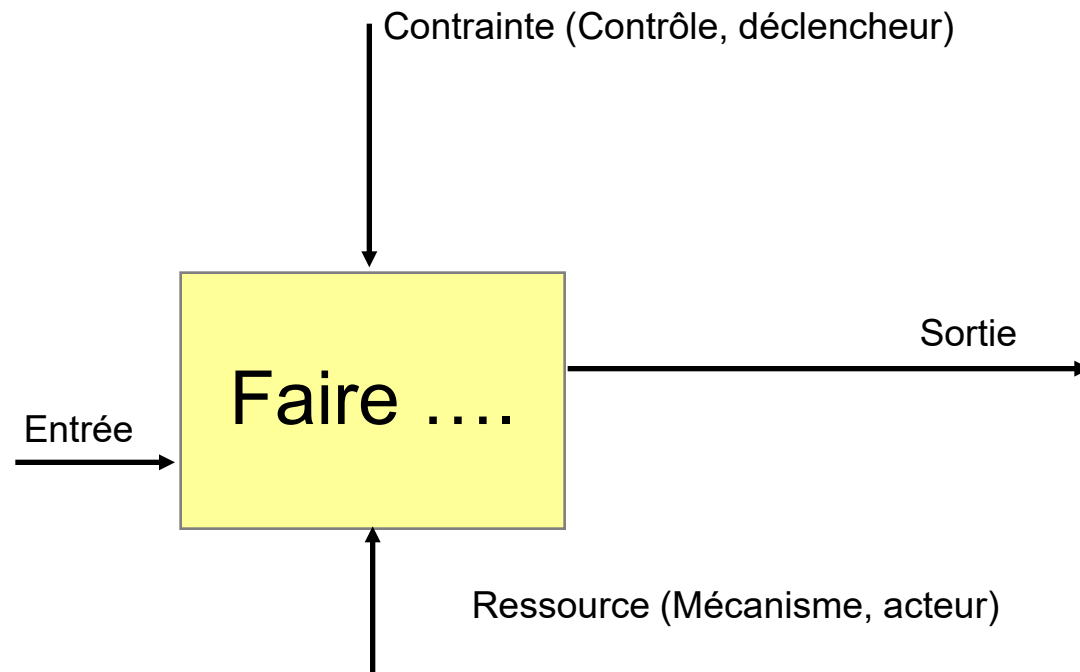
- La méthode aborde un processus en construisant un **Modèle** de ce processus.
- L'analyse de tout processus est menée de manière descendante, modulaire, hiérarchique et structurée.
- La méthode différencie autant que possible le **Modèle Fonctionnel** (étude du problème) et le **Modèle de Conception** (Description d'une solution à ce problème).
- La méthode modélise à la fois les « objets » (documents, produits, individus) et les activités (réalisées par les individus, les organisations, les machines, les logiciels, ..).
- Le langage de la méthode est graphique.
- La méthode favorise un travail d'équipe discipliné et coordonné.
- La méthode oblige à consigner sous forme écrite les choix importants.





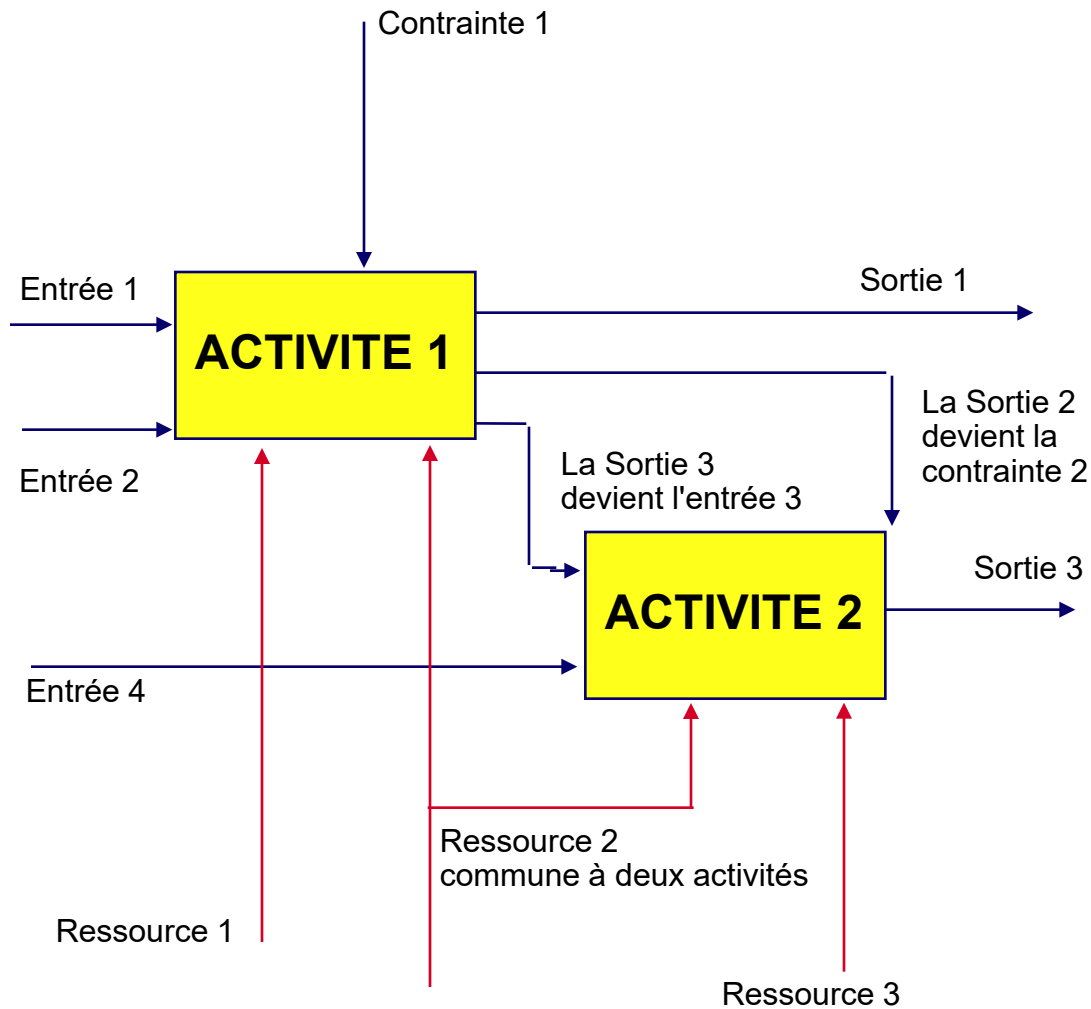
# Des boîtes et des flèches

- Les activités sont étiquetées par des verbes, commentés si besoin est.
- Les données sont étiquetées par des noms, qualifiés si besoin est.





# Des boîtes et des flèches



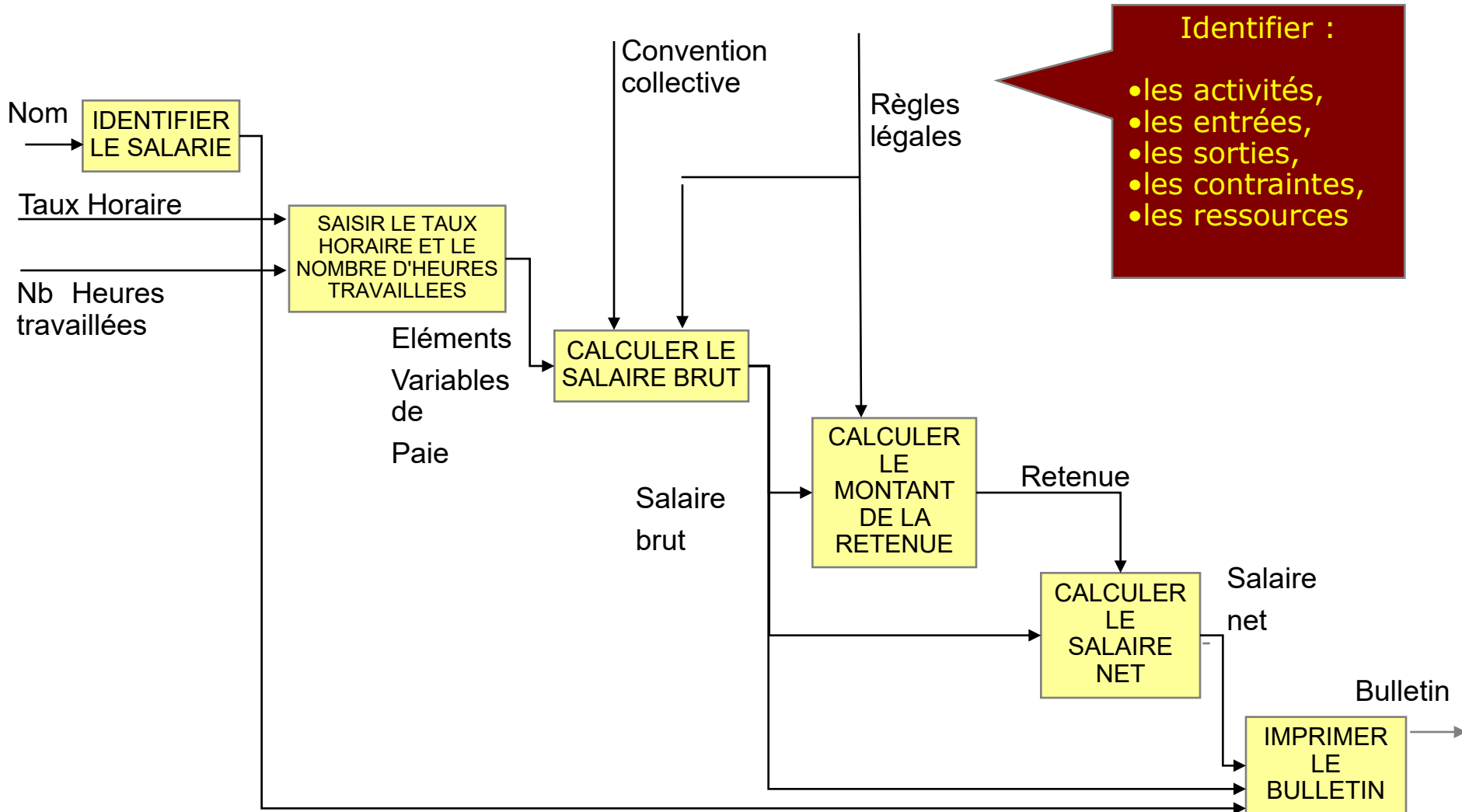
- Identifier les **activités**.
- Mettre en évidence les échanges:
  - Ce qui entre dans chaque activité (**input, entrée**),
  - Ce qui sort de chaque activité (**output, sortie, résultat**).
- Recenser les **contraintes** (certaines associées à la notion d'évènement):
  - Conditions de déclenchement,
  - Tops de synchronisation.
- Recenser les **ressources** mises en œuvre.



# Rappel : notre programme exemple

## Identifier :

- les activités,
- les entrées,
- les sorties,
- les contraintes,
- les ressources







## Exemple #1

# La visite chez le médecin





Exercice





# Exemple visite chez le médecin

Prendre  
conscience  
de son état

**A1**

Evaluer  
la gravité et  
l'urgence

**A2**

Prendre  
Rendez-vous

**A3**

Consulter

**A4**

S'approvisionner  
en médicaments

**A5**

Suivre les  
prescriptions

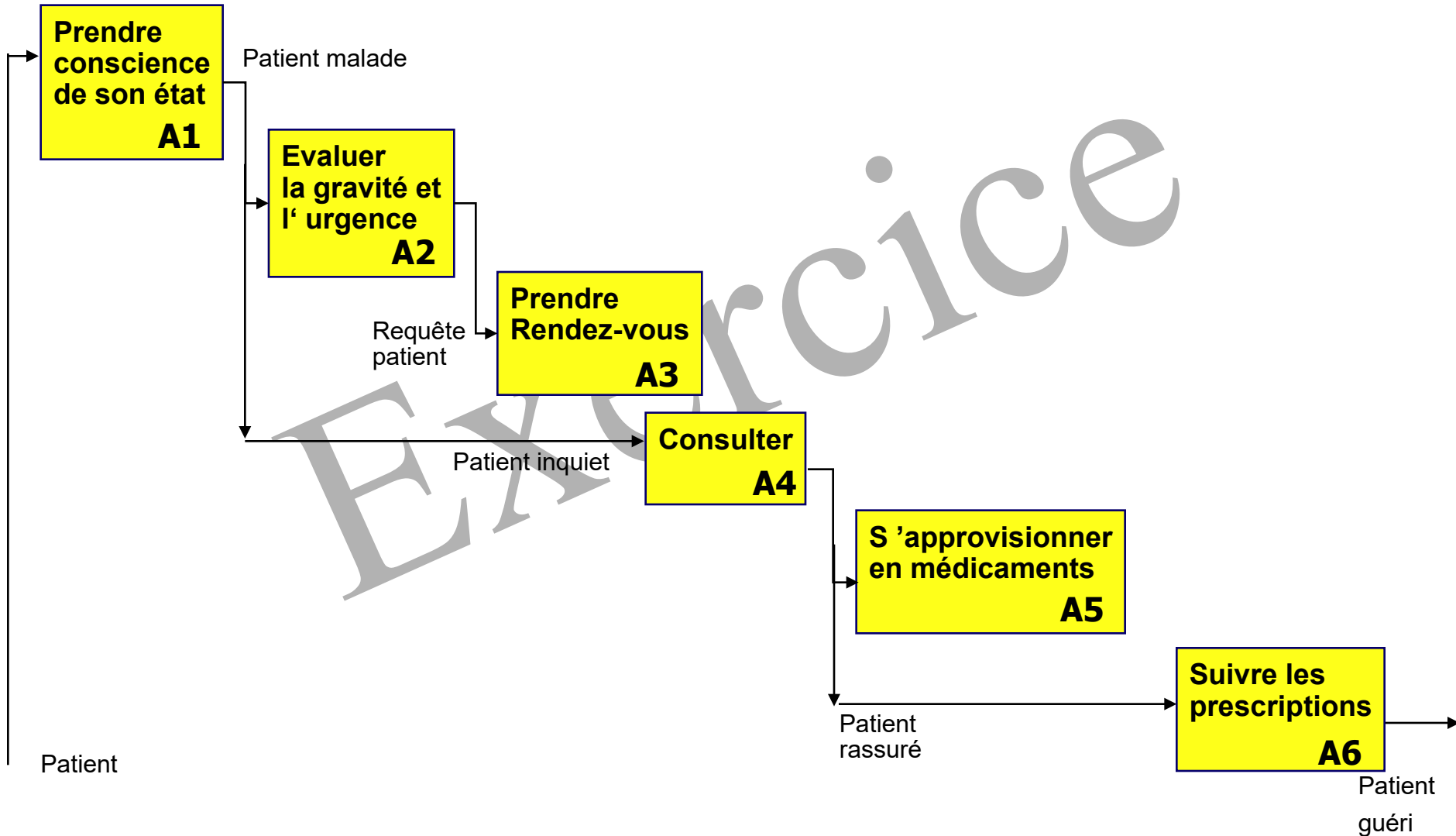
**A6**

EXERCICE





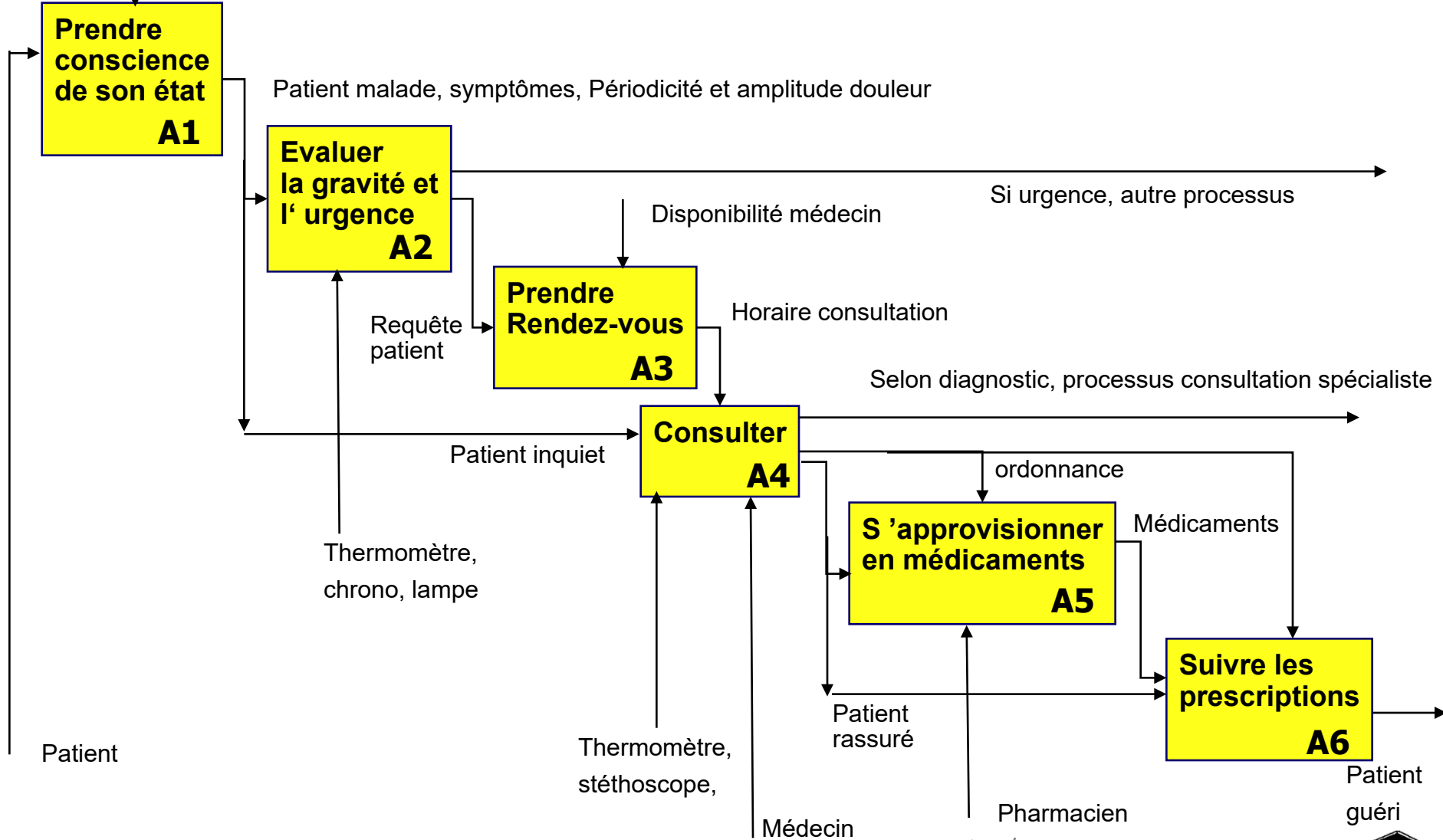
# Exemple visite chez le médecin





# Exemple visite chez le médecin

Symptômes, Périodicité et amplitude douleur





# Exemple #2

## Petit projet de développement logiciel





# Exercice





# Appliquer la méthode sur le processus de réalisation d'un logiciel

Formaliser  
les besoins  
exprimés

**A1**

Etablir les  
spécifications  
détaillées

**A2**

Ecrire et  
tester le  
code

**A3**

Intégrer

**A4**

Valider par  
rapport aux  
spécifications

**A5**

Mettre en  
exploitation

**A6**

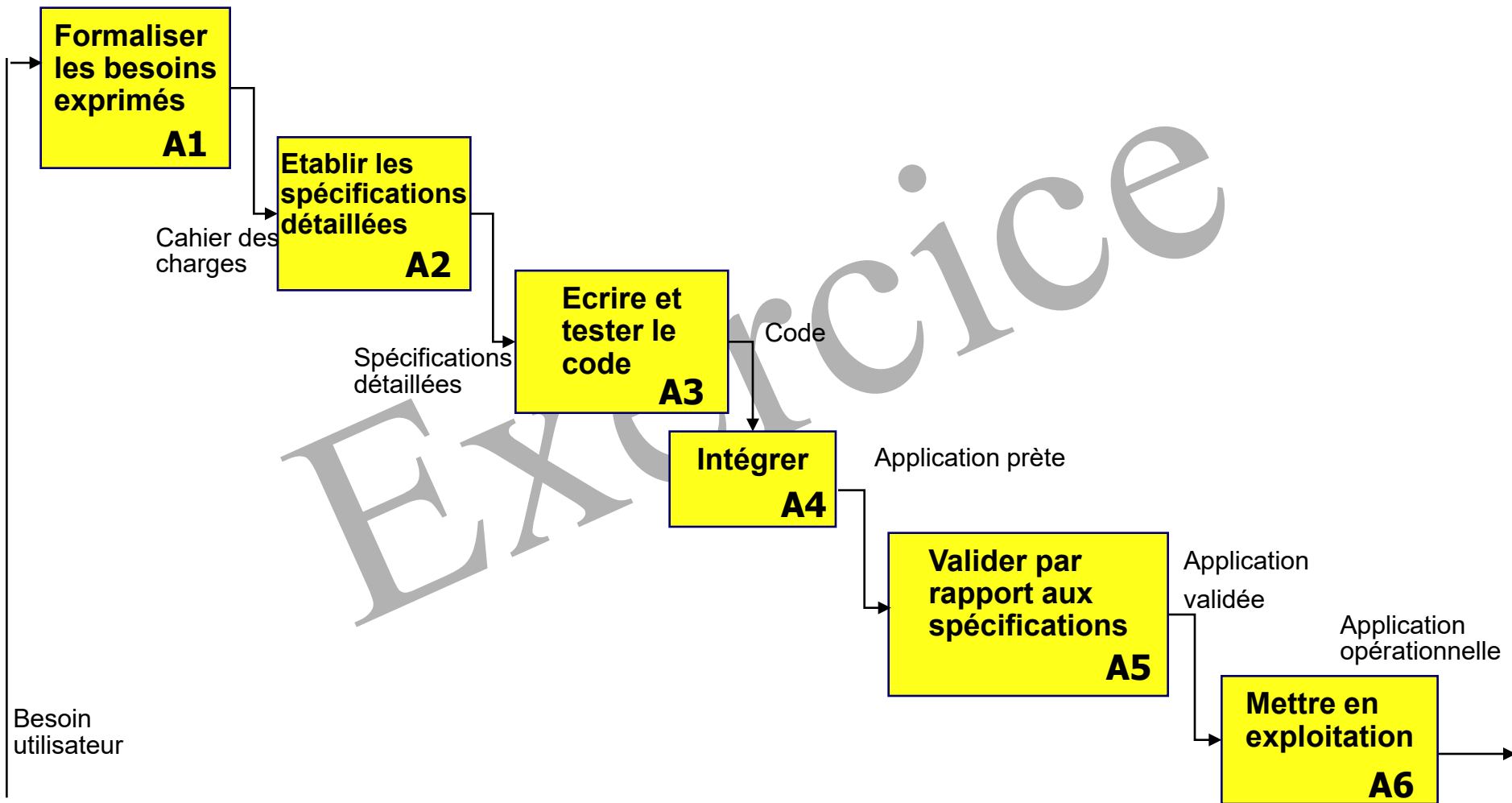
EXERCICE





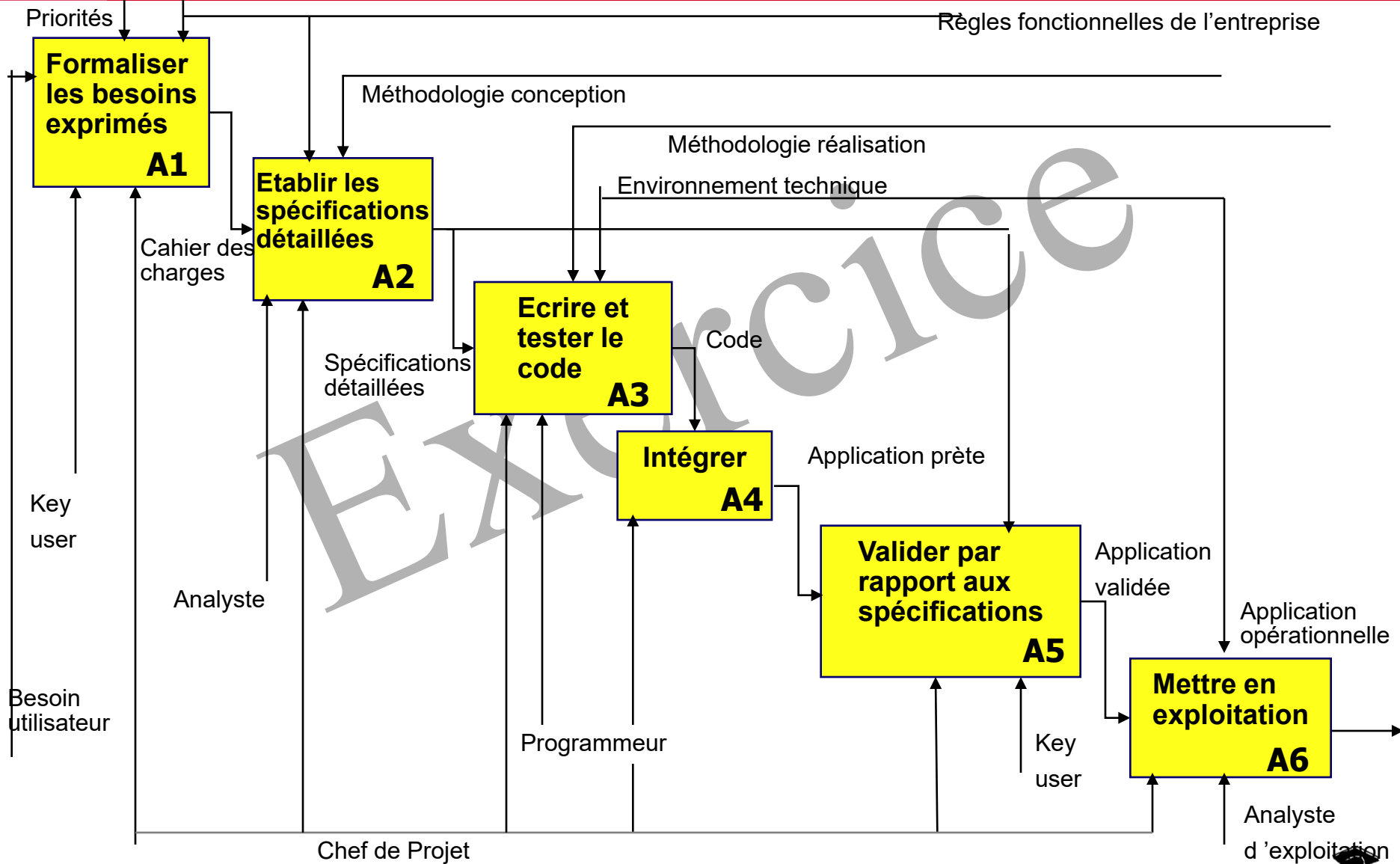


# Appliquer la méthode sur le processus de réalisation d'un logiciel





# Appliquer la méthode sur le processus de réalisation d'un logiciel





- Comme nous l'avons dit plus haut, SADT/IDEF vous est présenté à titre d'exemple.
- C'est une méthode ancienne mais qui reste un excellent outil de « remue-méninges »
- Les informaticiens utilisent des langages de modélisation complexes comme UML (*Unified Modeling Language*).
- Pour exprimer leurs besoins, ils proposent aux utilisateurs des notations plus simples comme BPMN (*Business Process Modelling Notation*)
- Mais l'usage d'une notation ne fournit pas la démarche associée à une méthodologie
- Vous pouvez appliquer la lettre (la forme) de BPMN en gardant l'esprit (le fond) de SADT





## Etude de cas

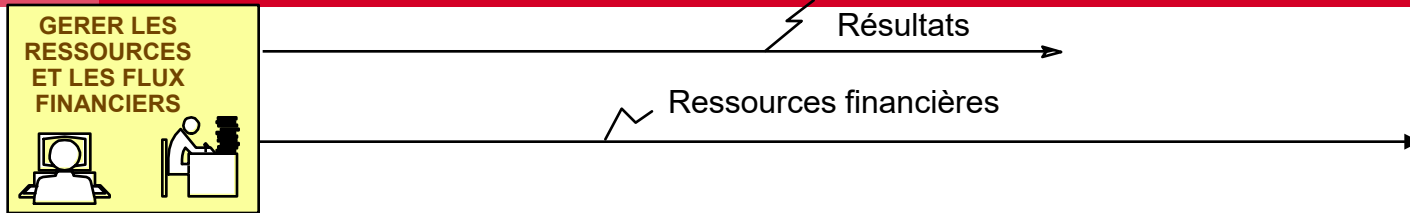
# L'entreprise et son système logistique

*Comprendre le fonctionnement de l'entreprise en deux diagrammes*



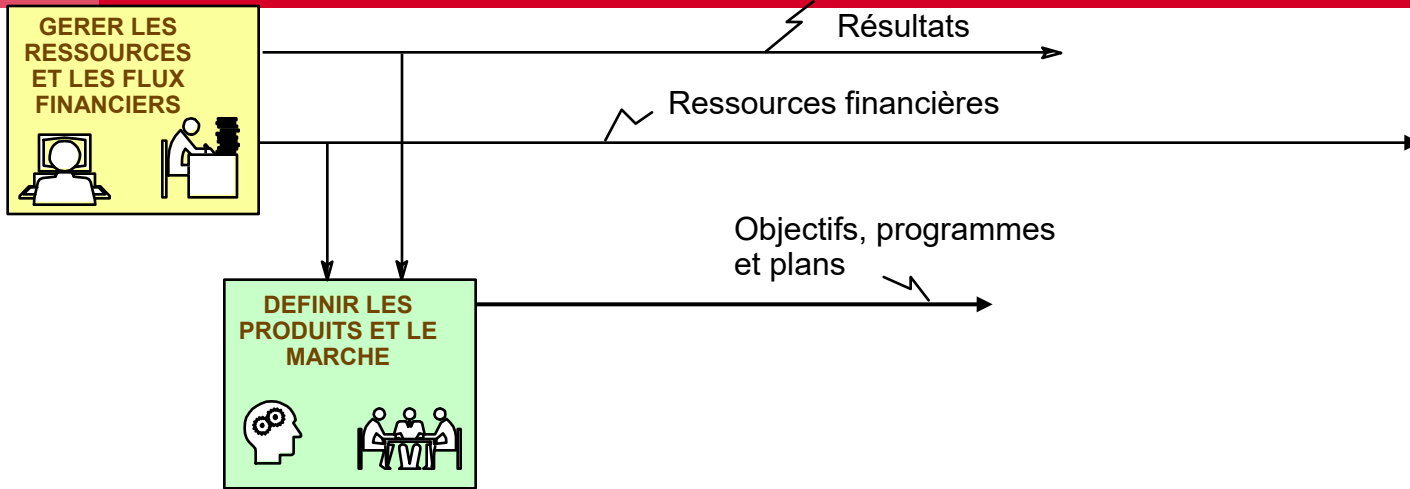


# Les systèmes de gestion de l'entreprise



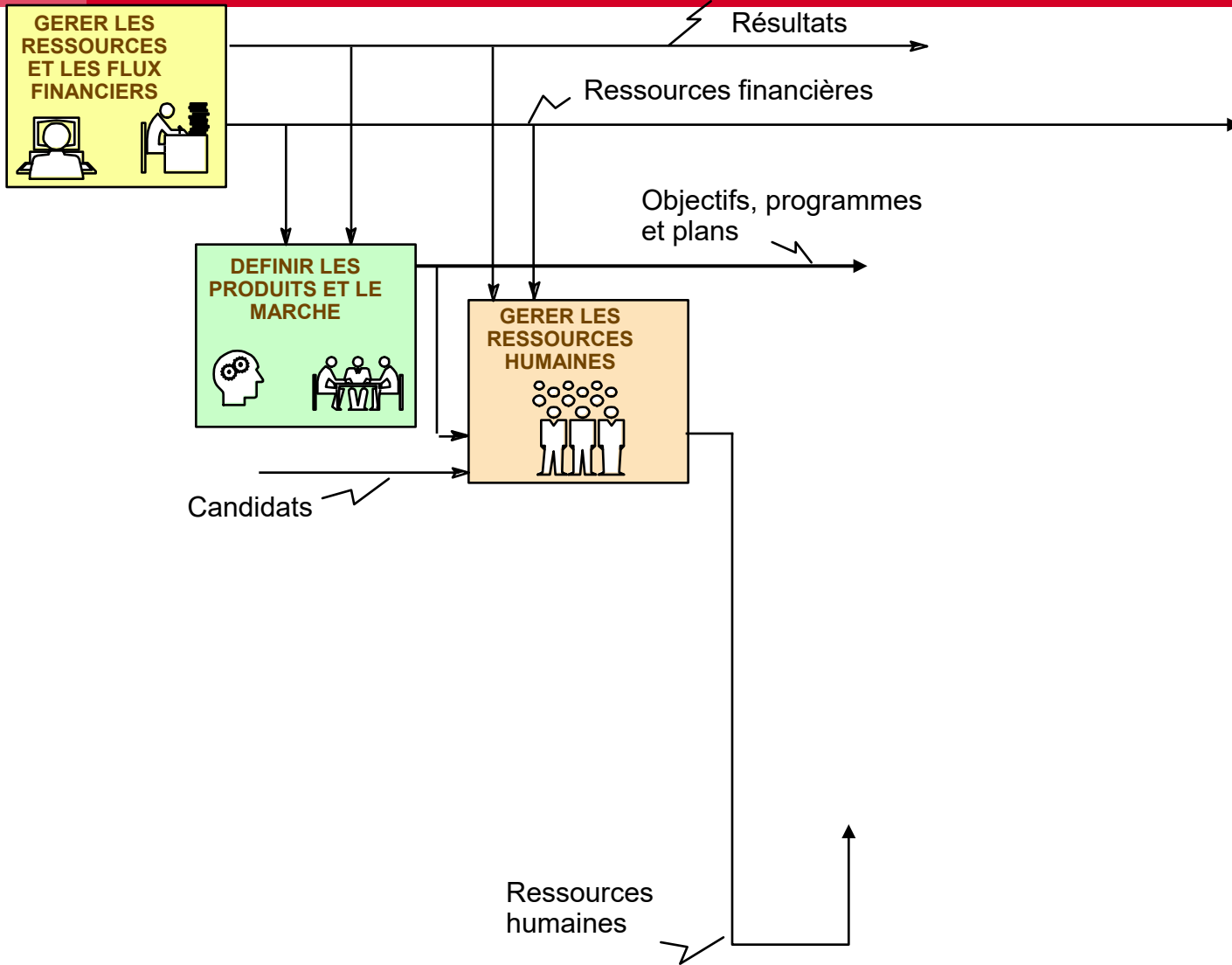


# Les systèmes de gestion de l'entreprise



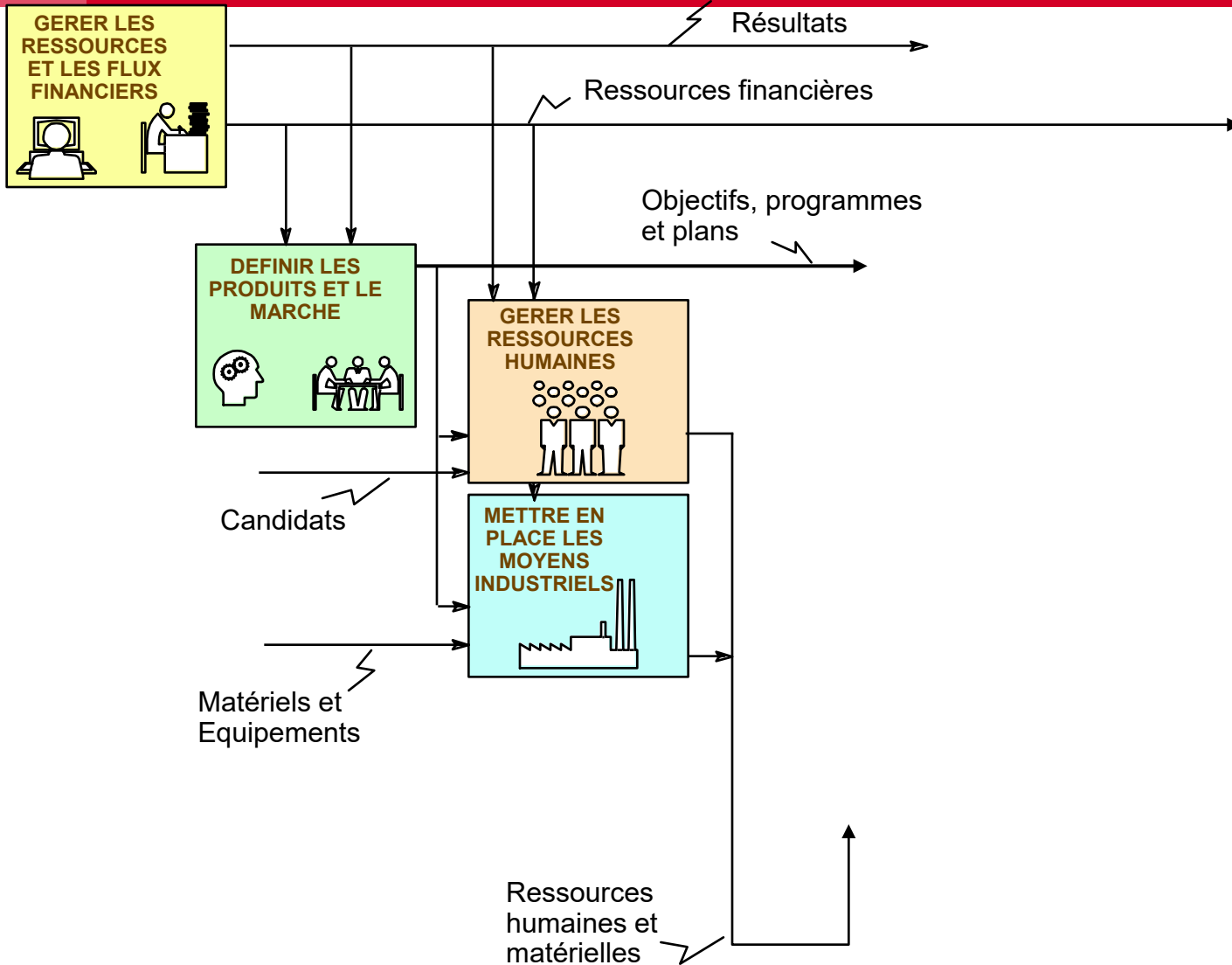


# Les systèmes de gestion de l'entreprise





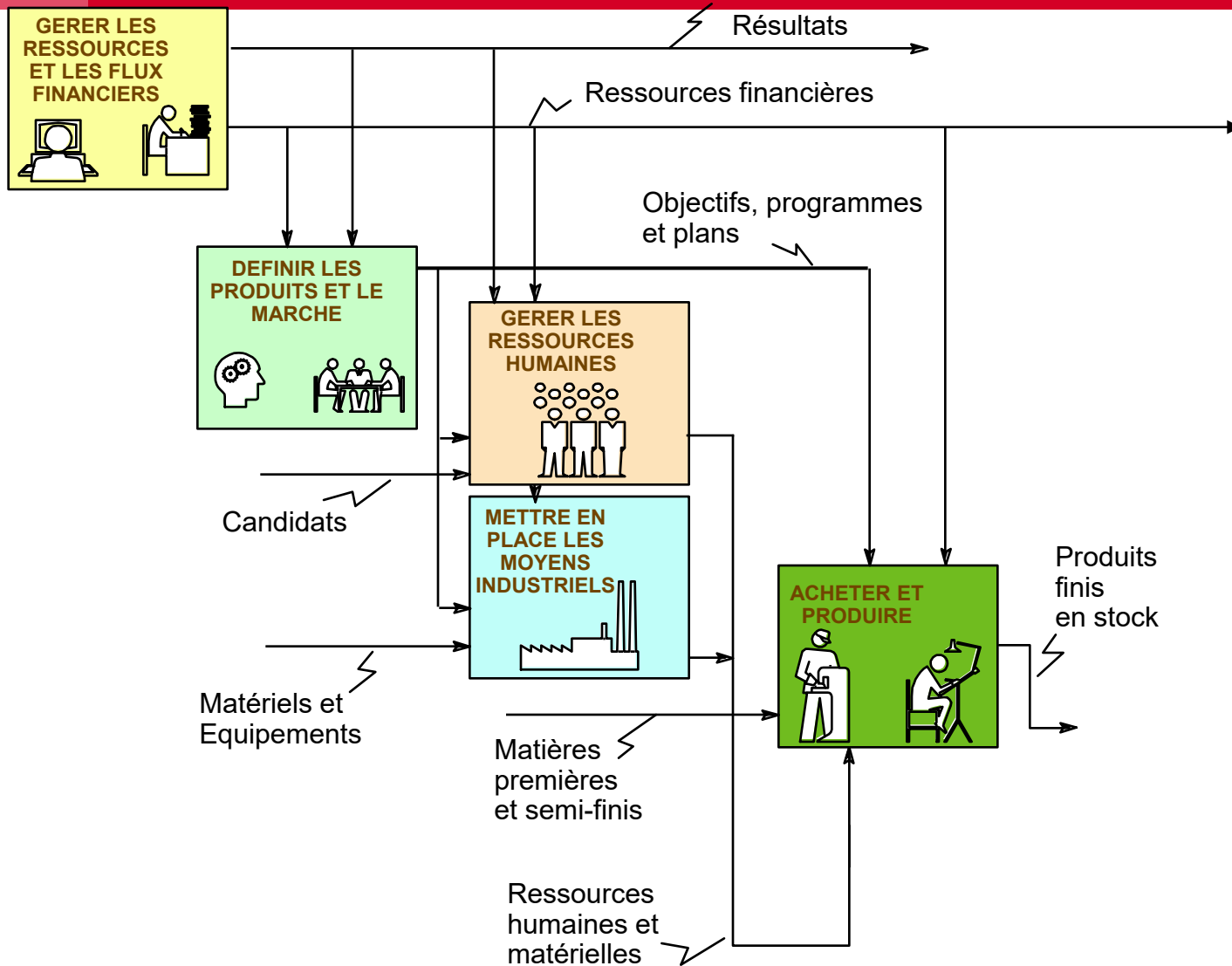
# Les systèmes de gestion de l'entreprise





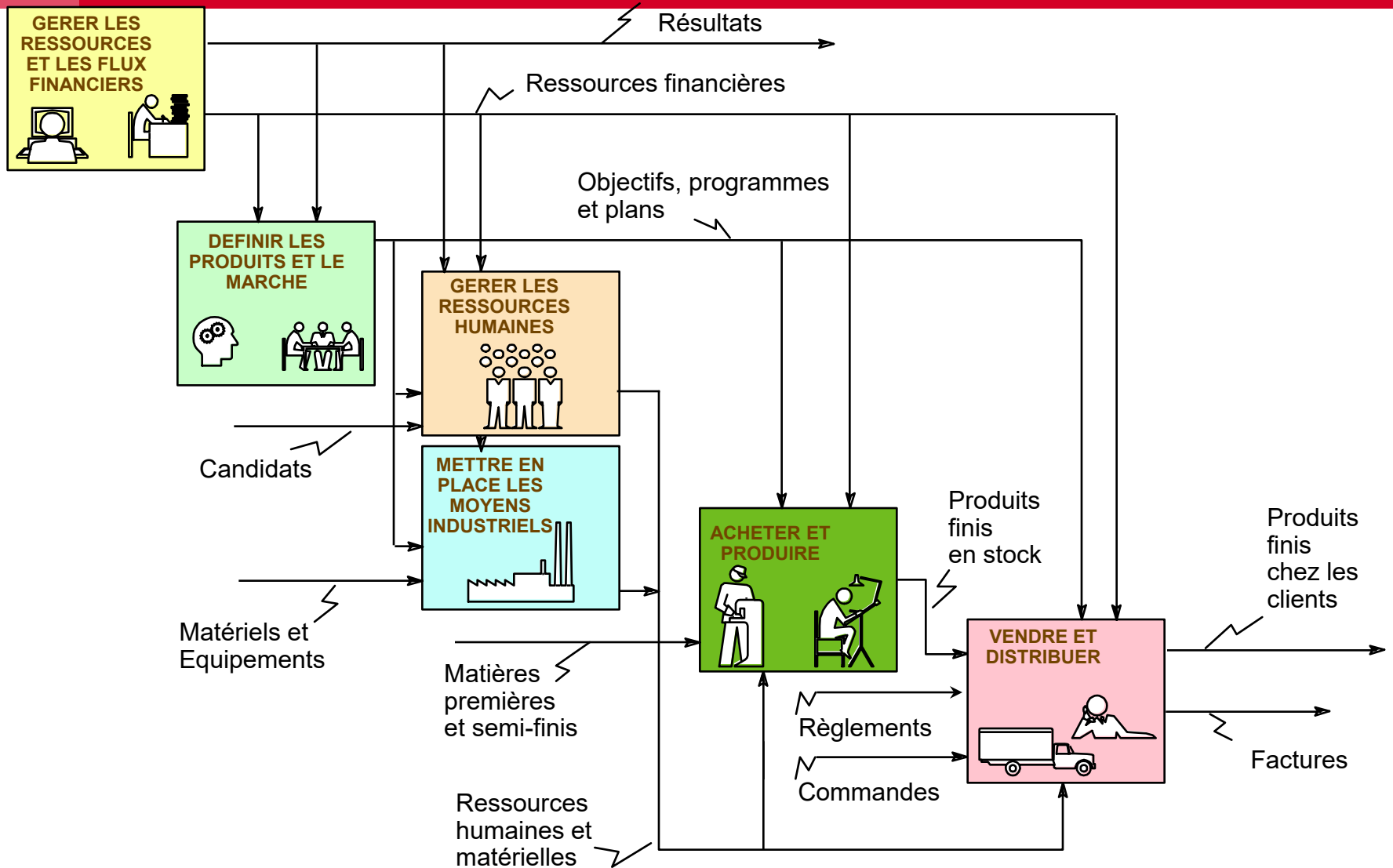


# Les systèmes de gestion de l'entreprise



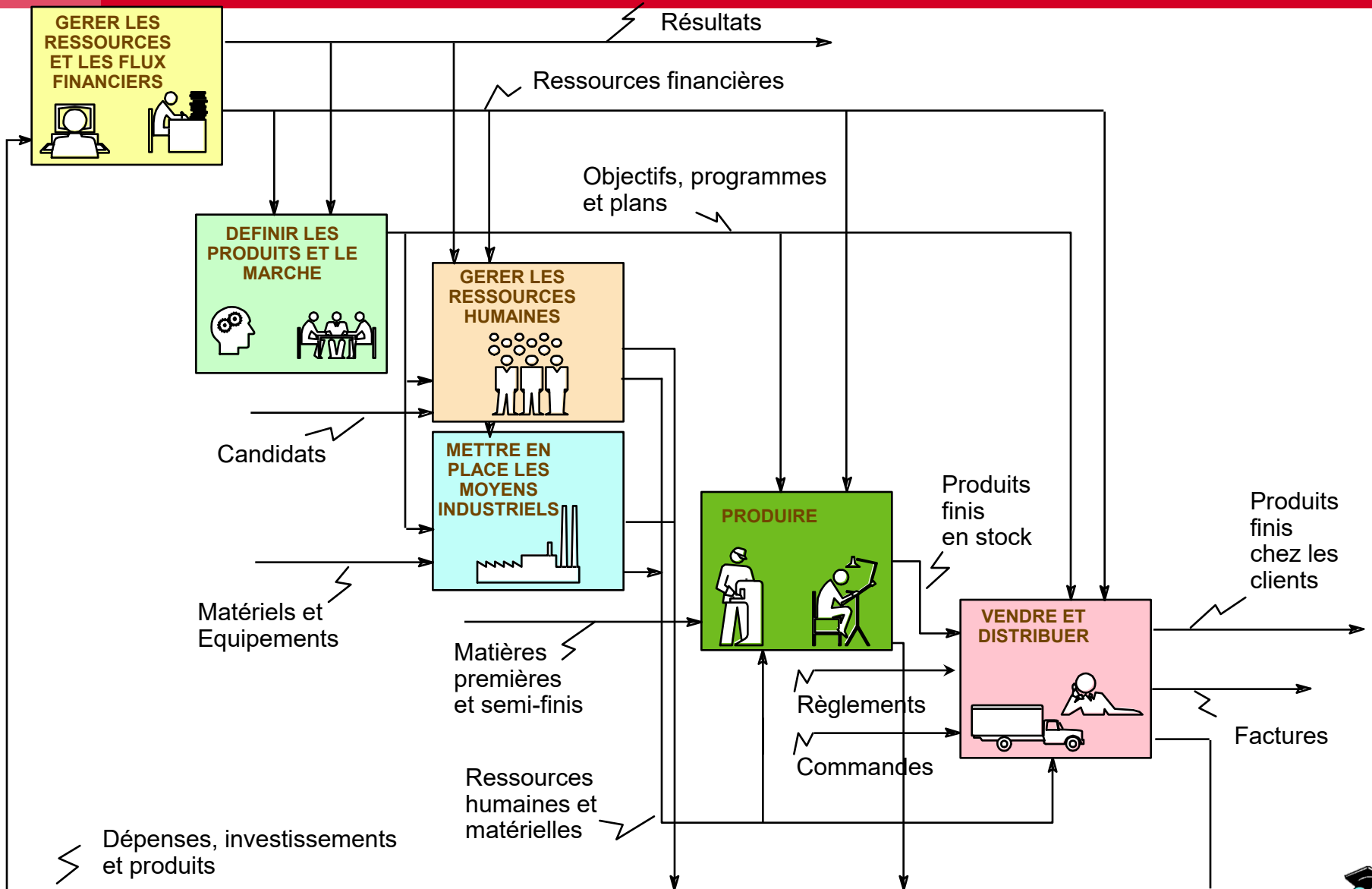


# Les systèmes de gestion de l'entreprise





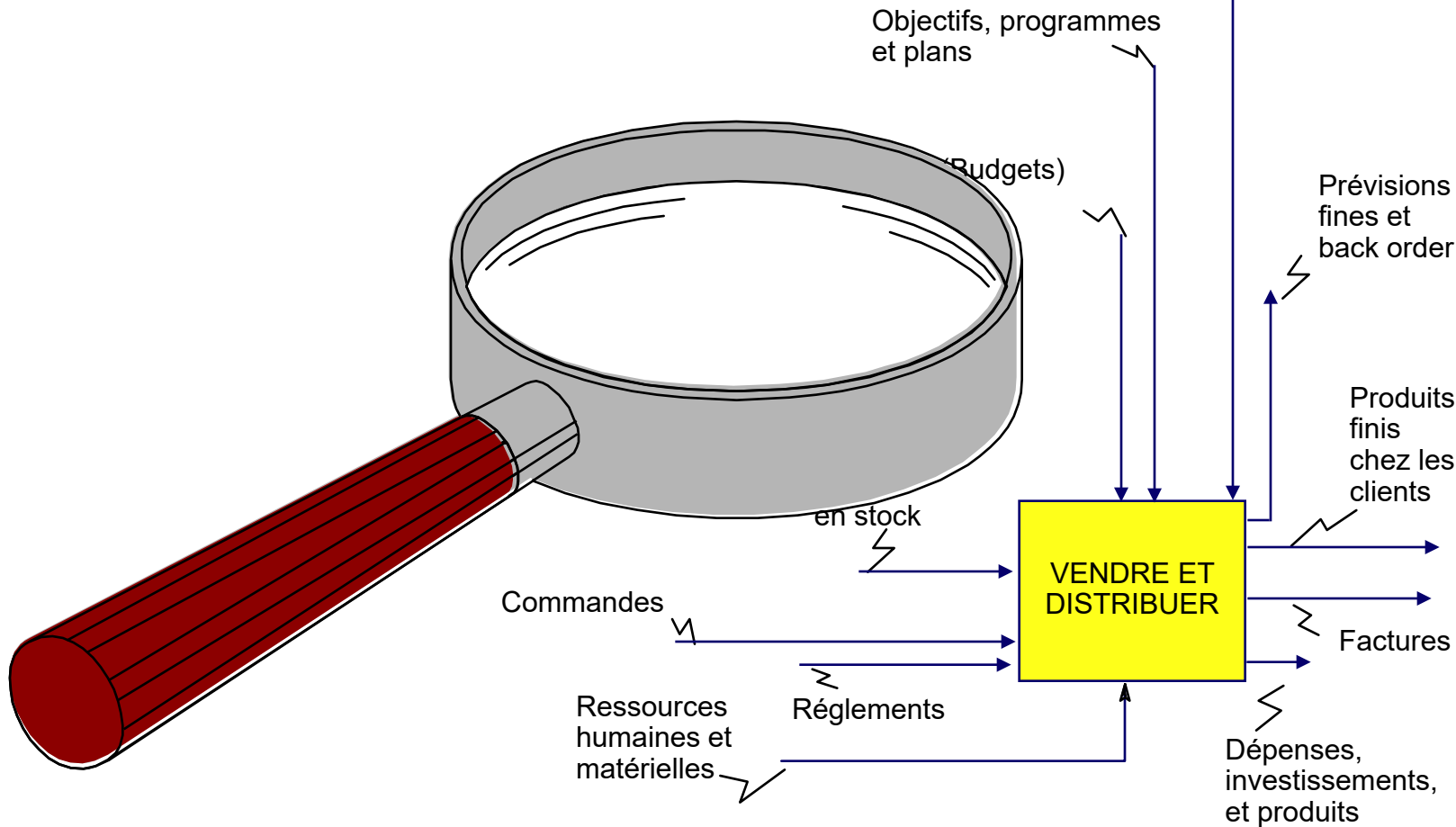
# Les systèmes de gestion de l'entreprise

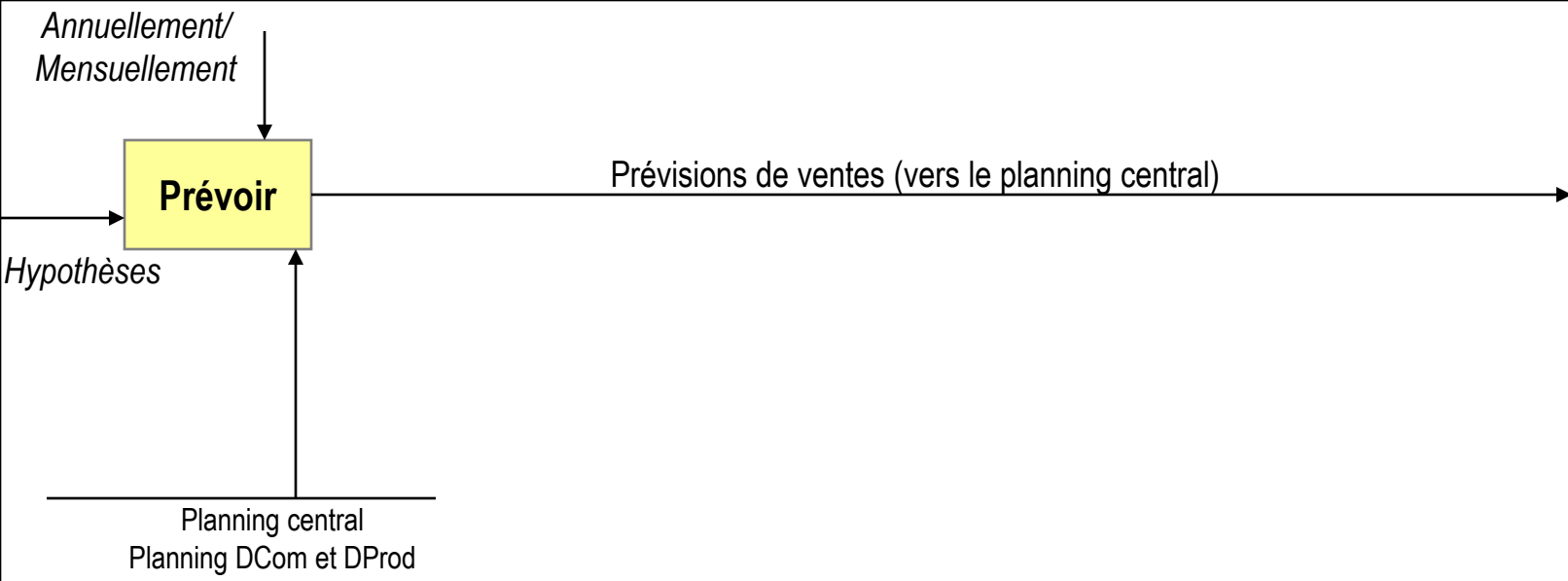




# Zoom sur le S.G. Ventes-Distribution

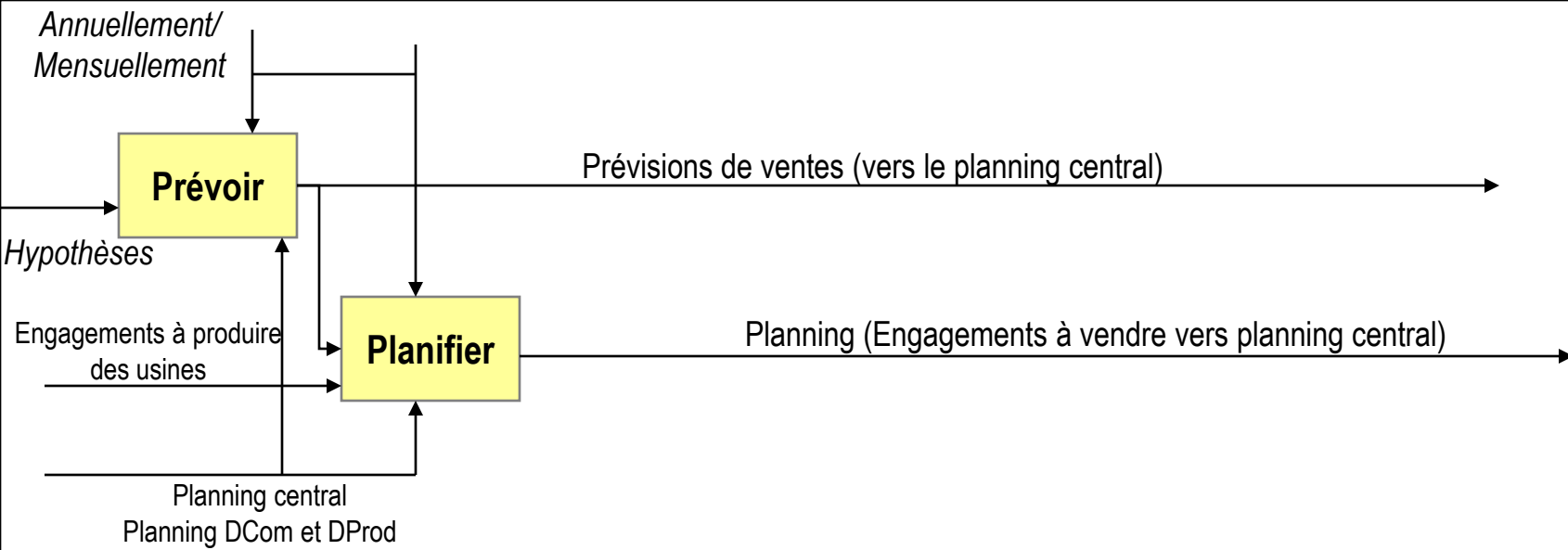
Contexte économique  
Marché  
Concurrence





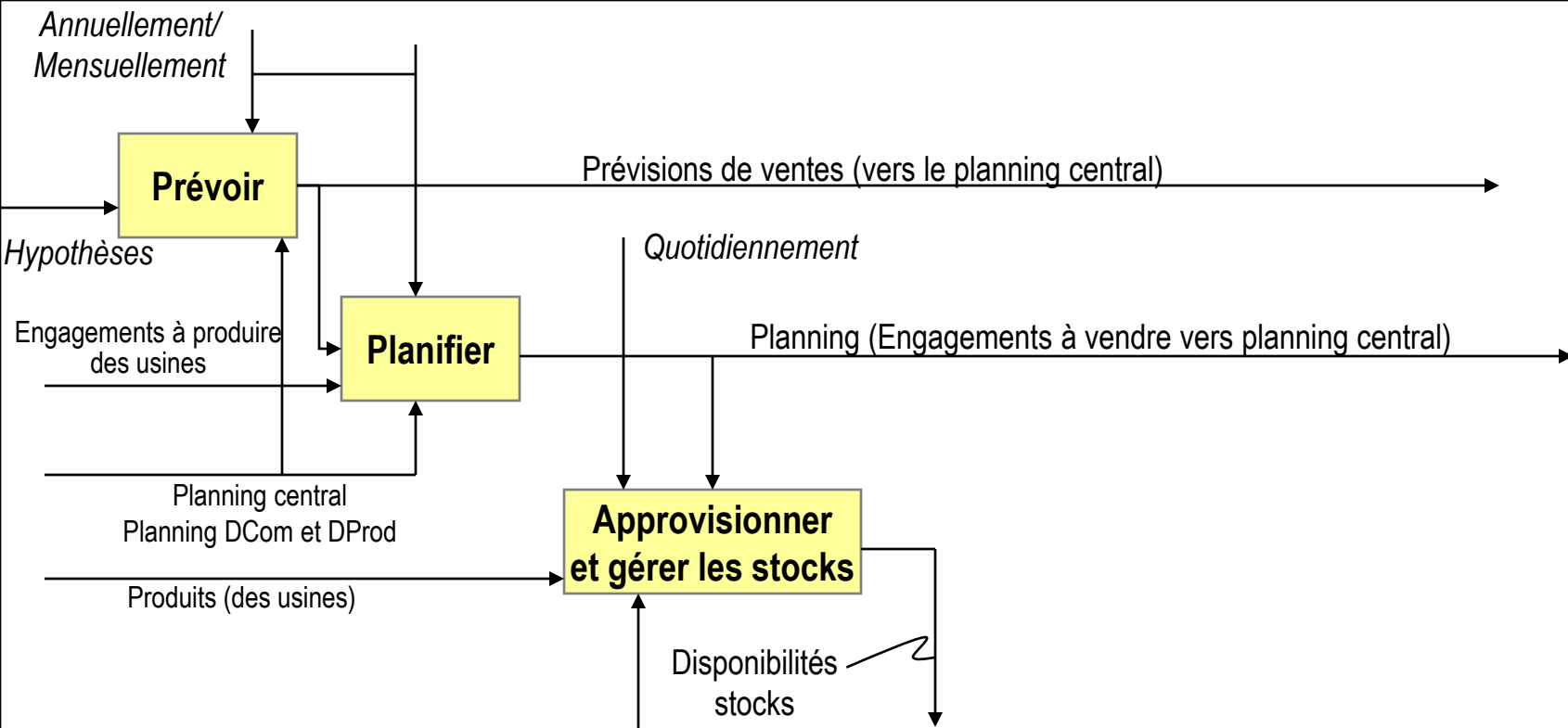
## **Le processus linéaire**





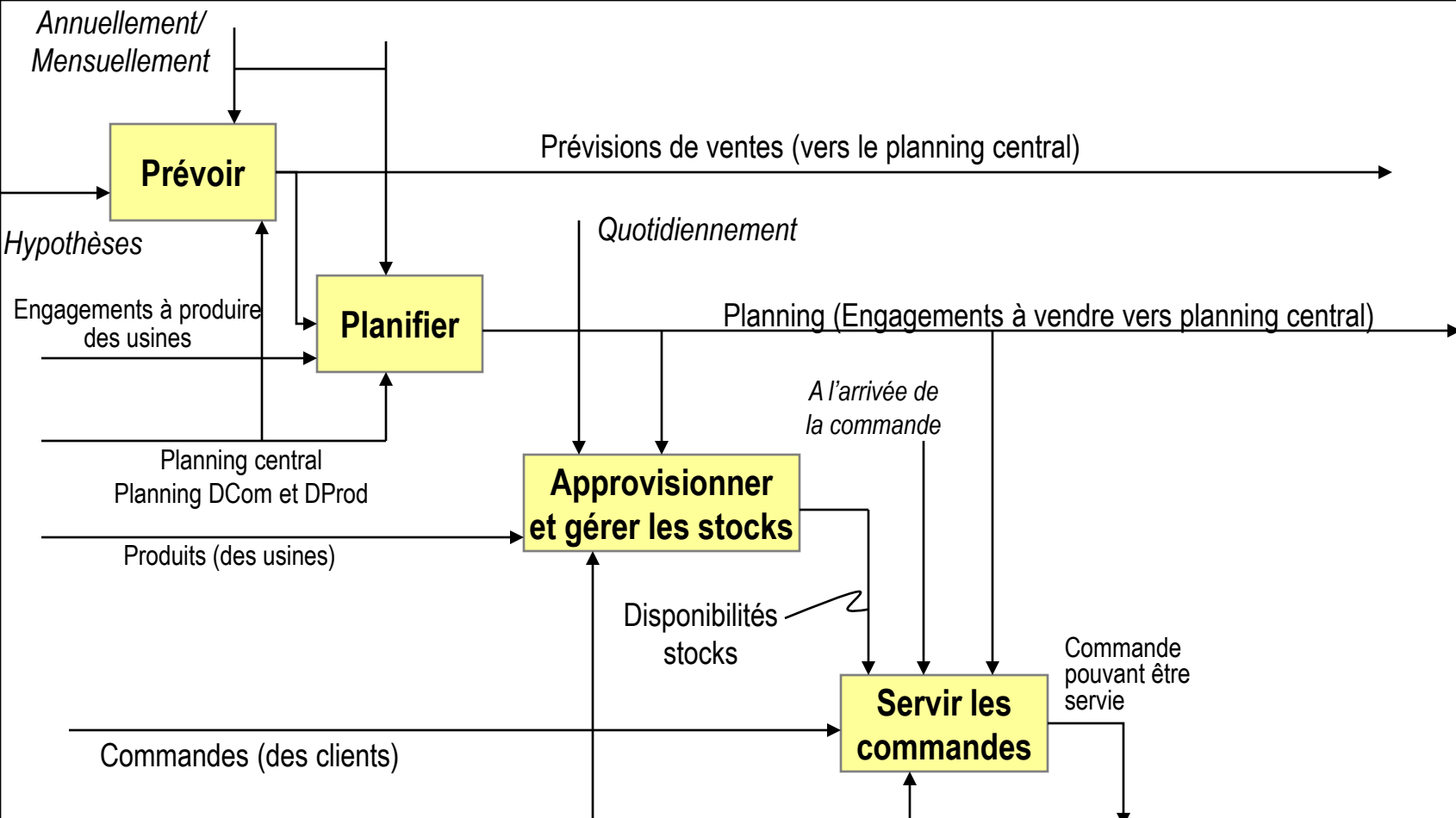
## **Le processus linéaire**





# Le processus linéaire



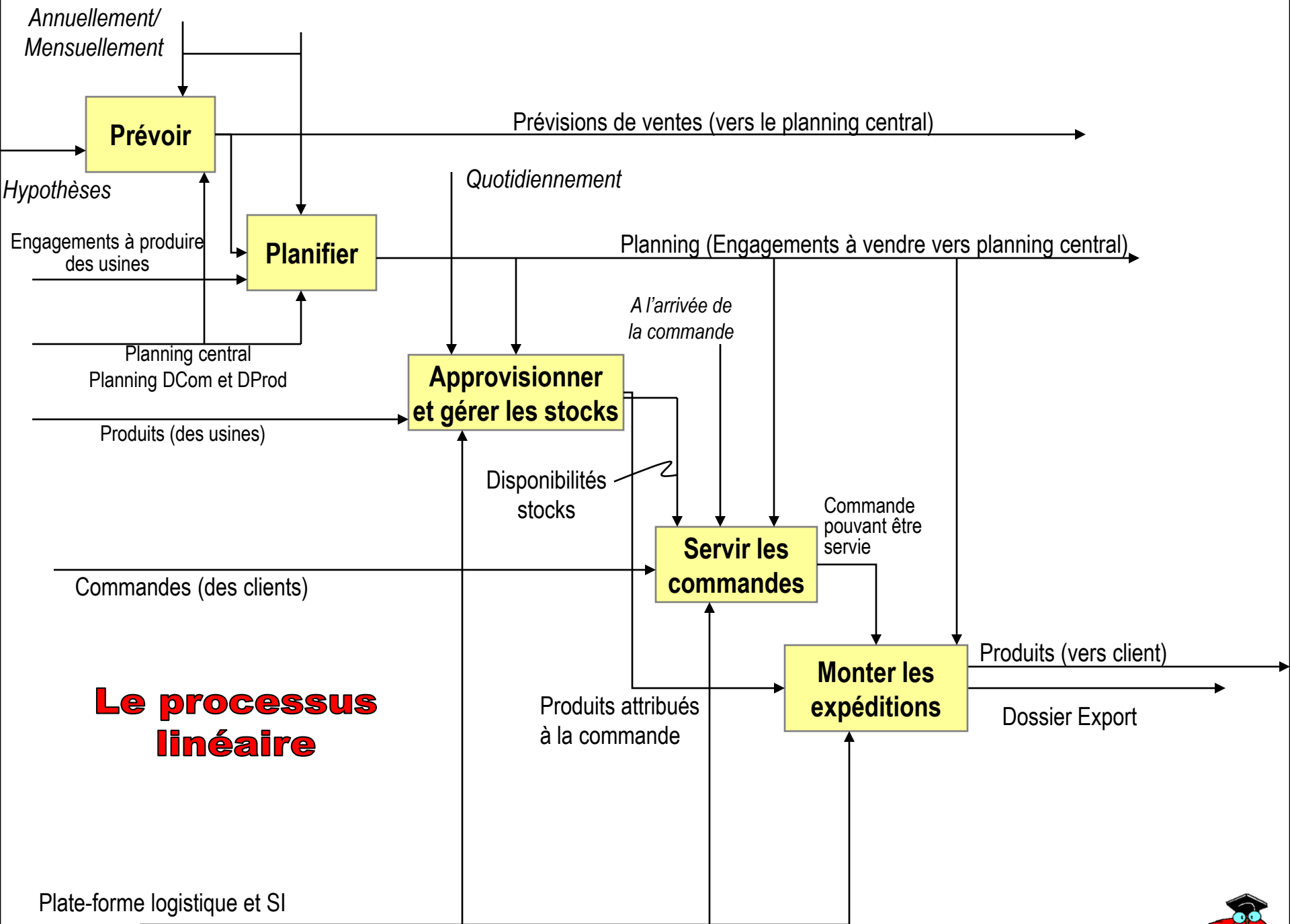


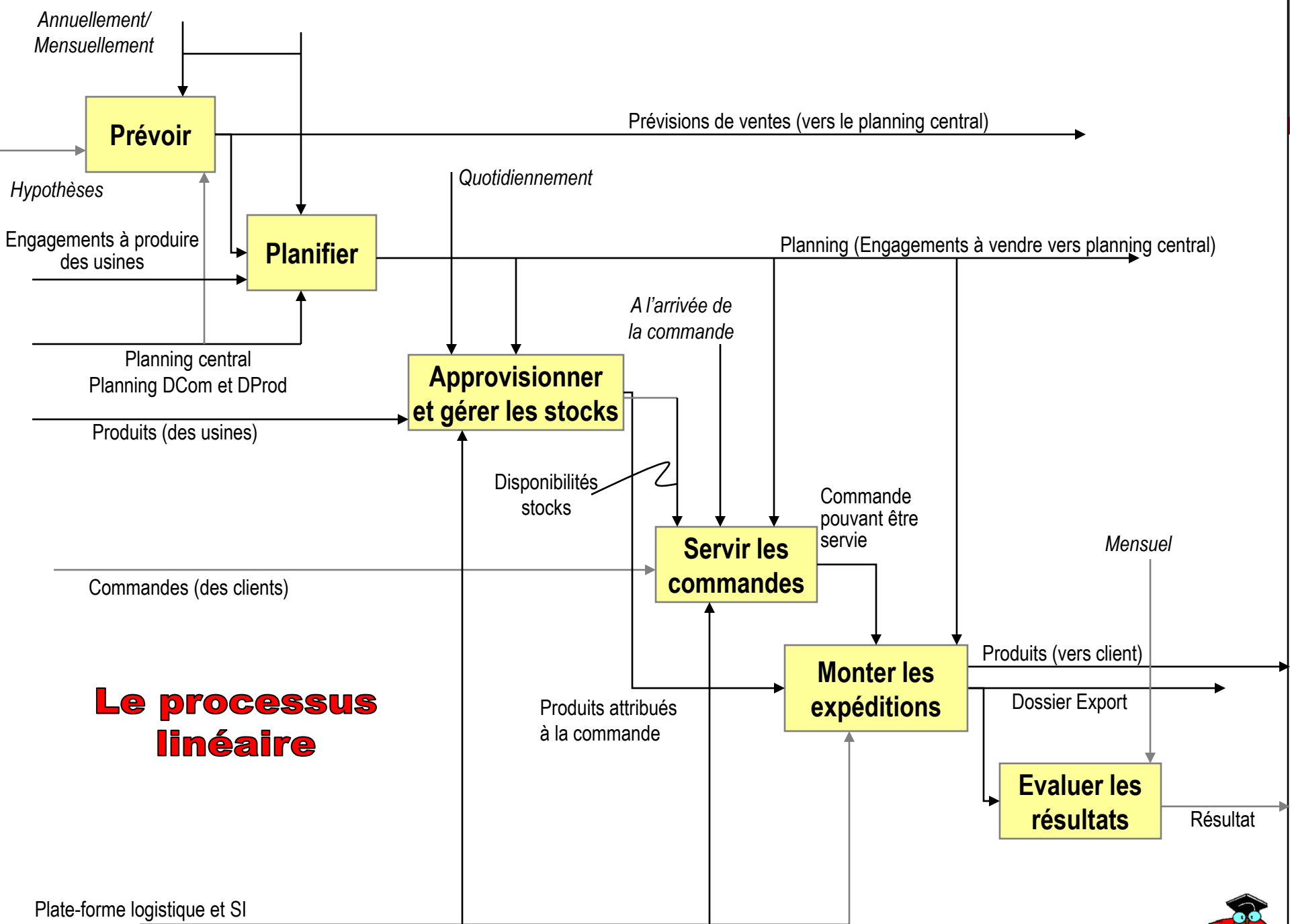
## Le processus linéaire

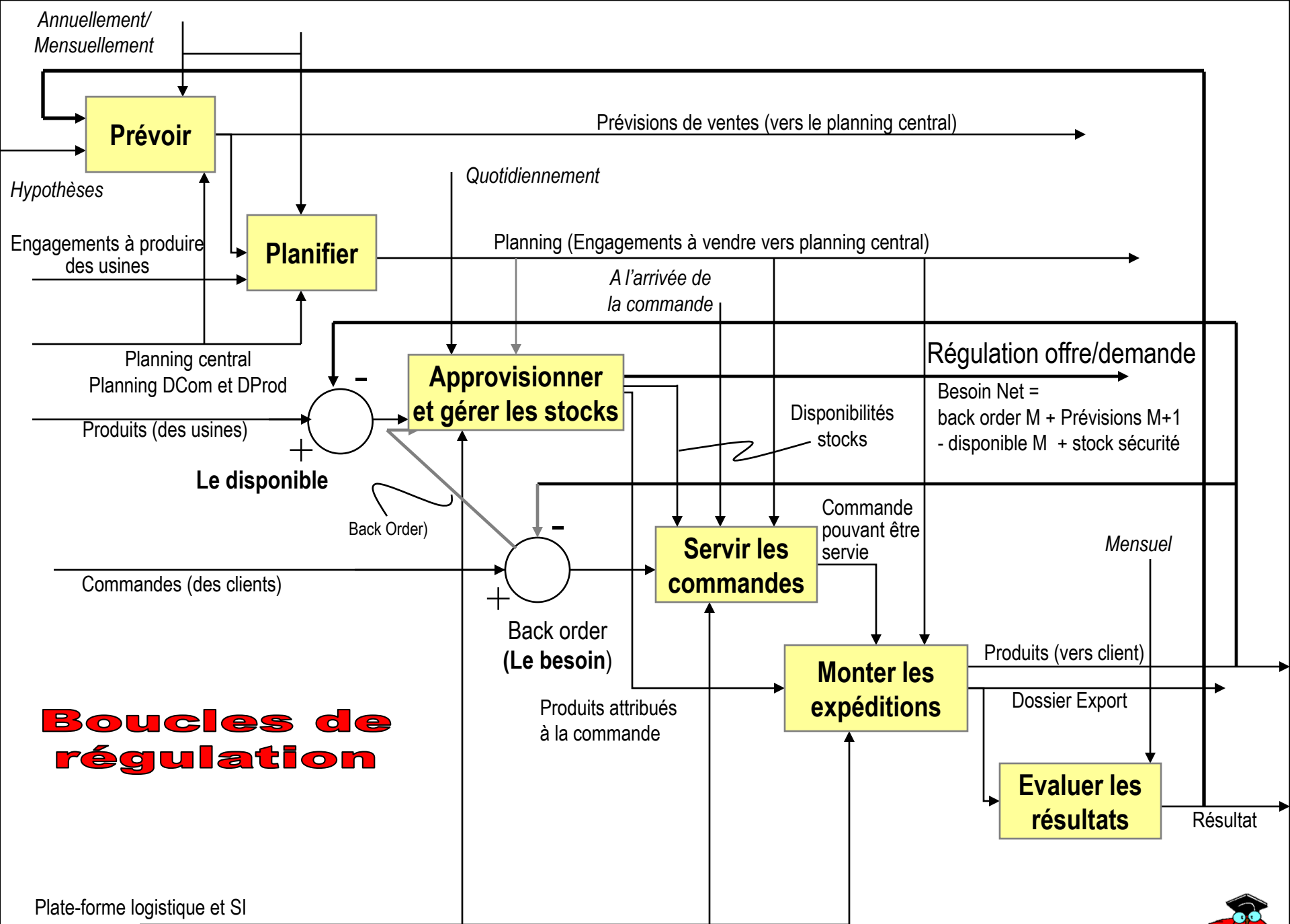
Plate-forme logistique et SI

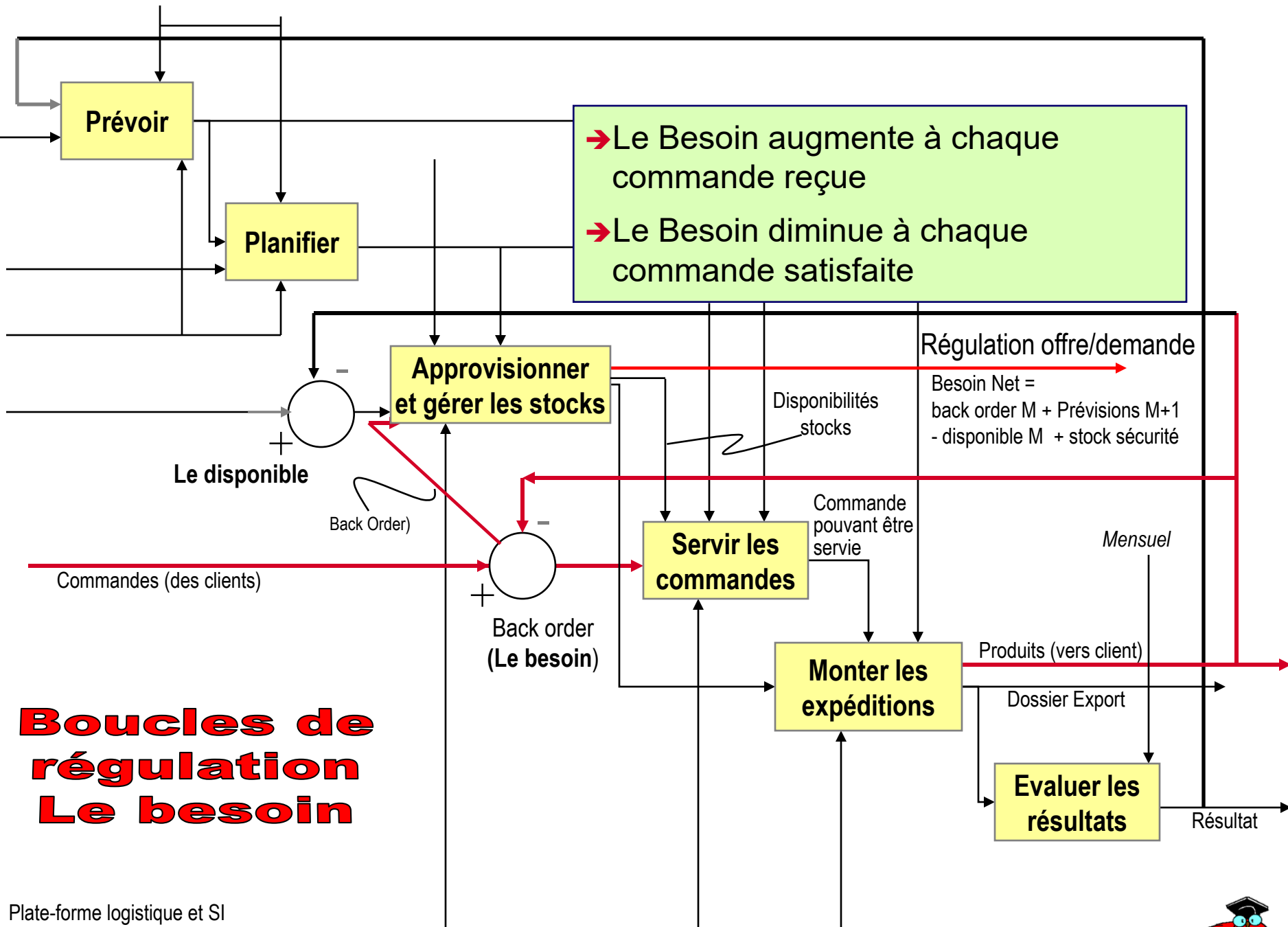


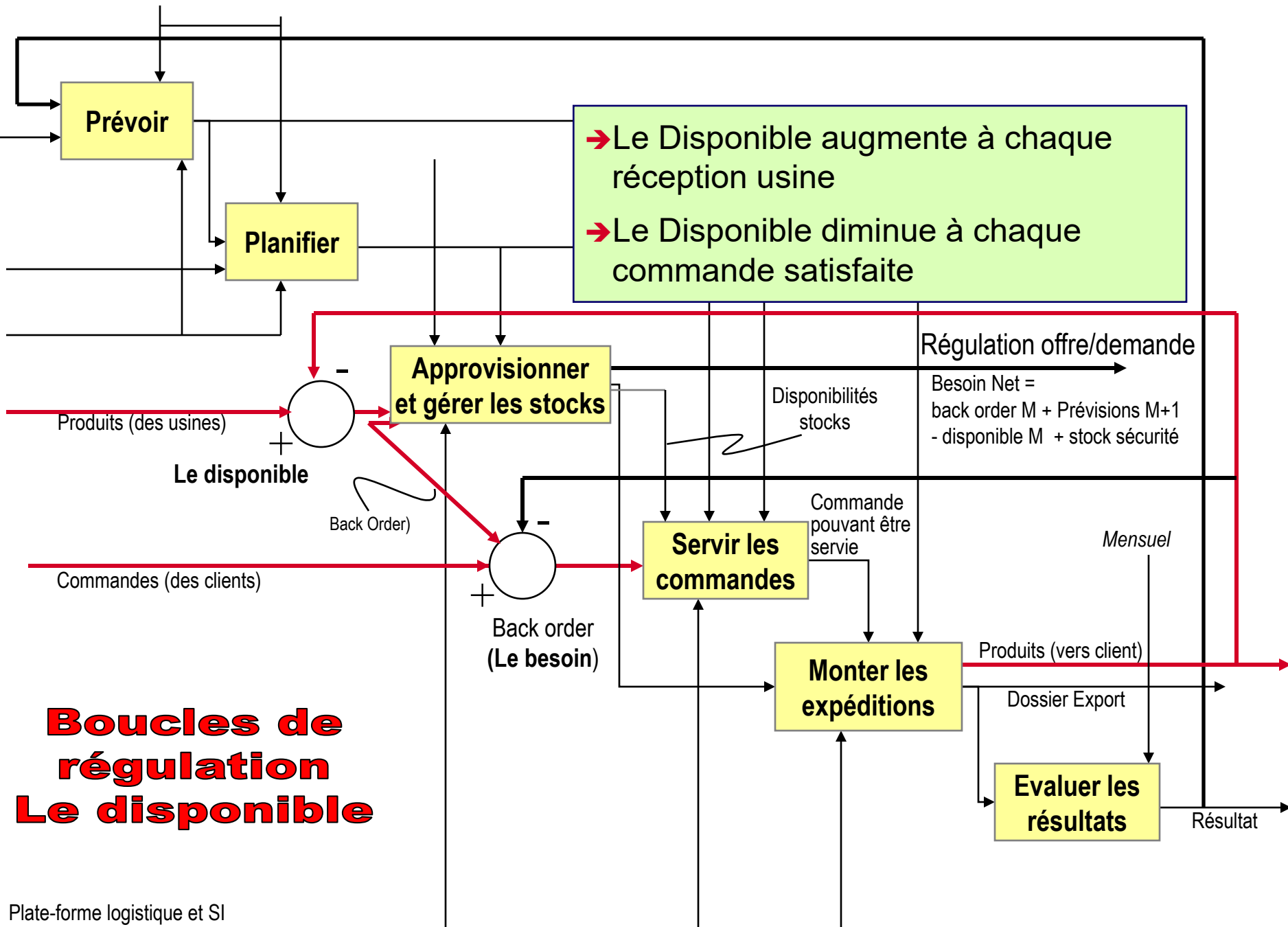




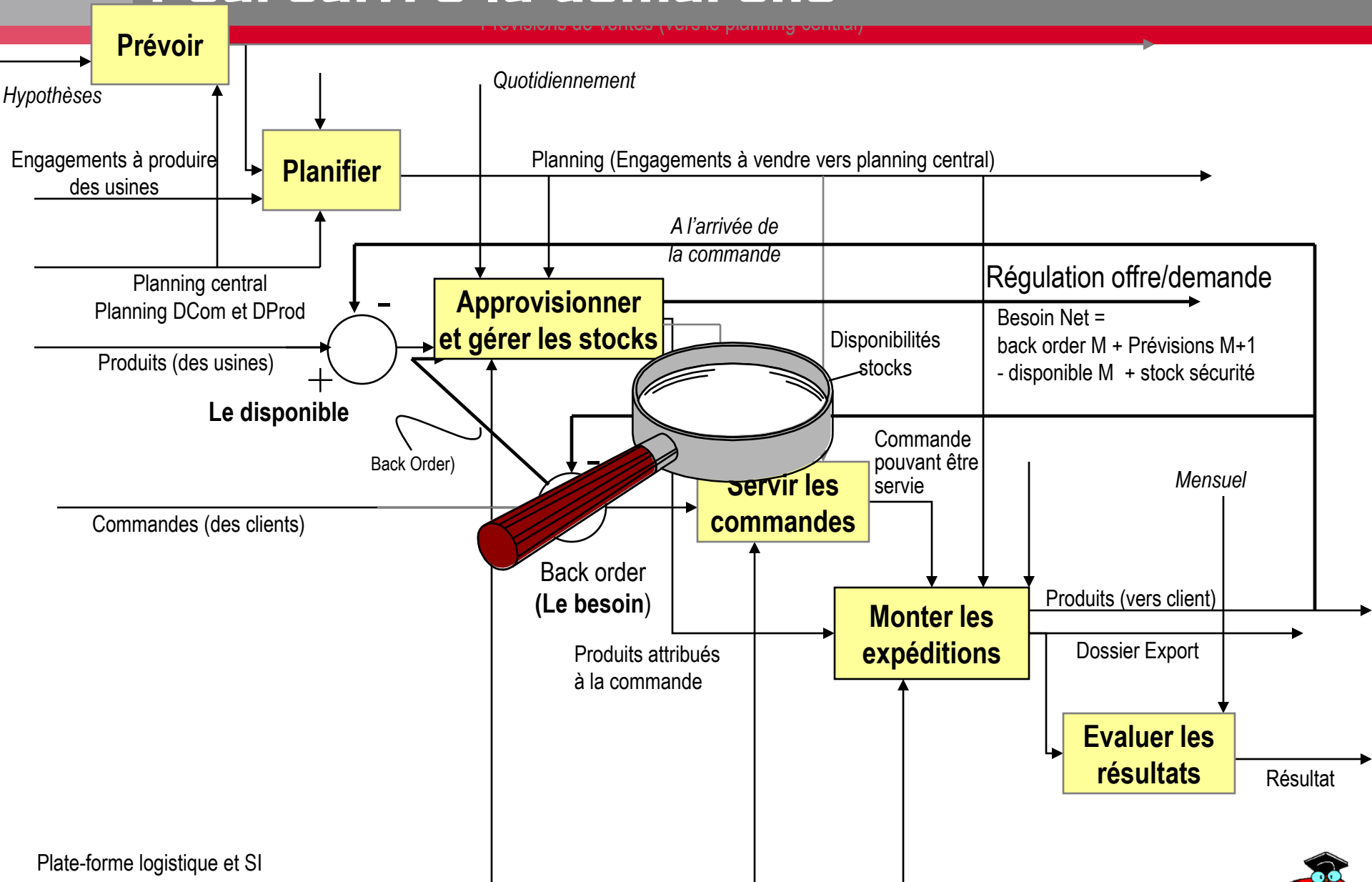




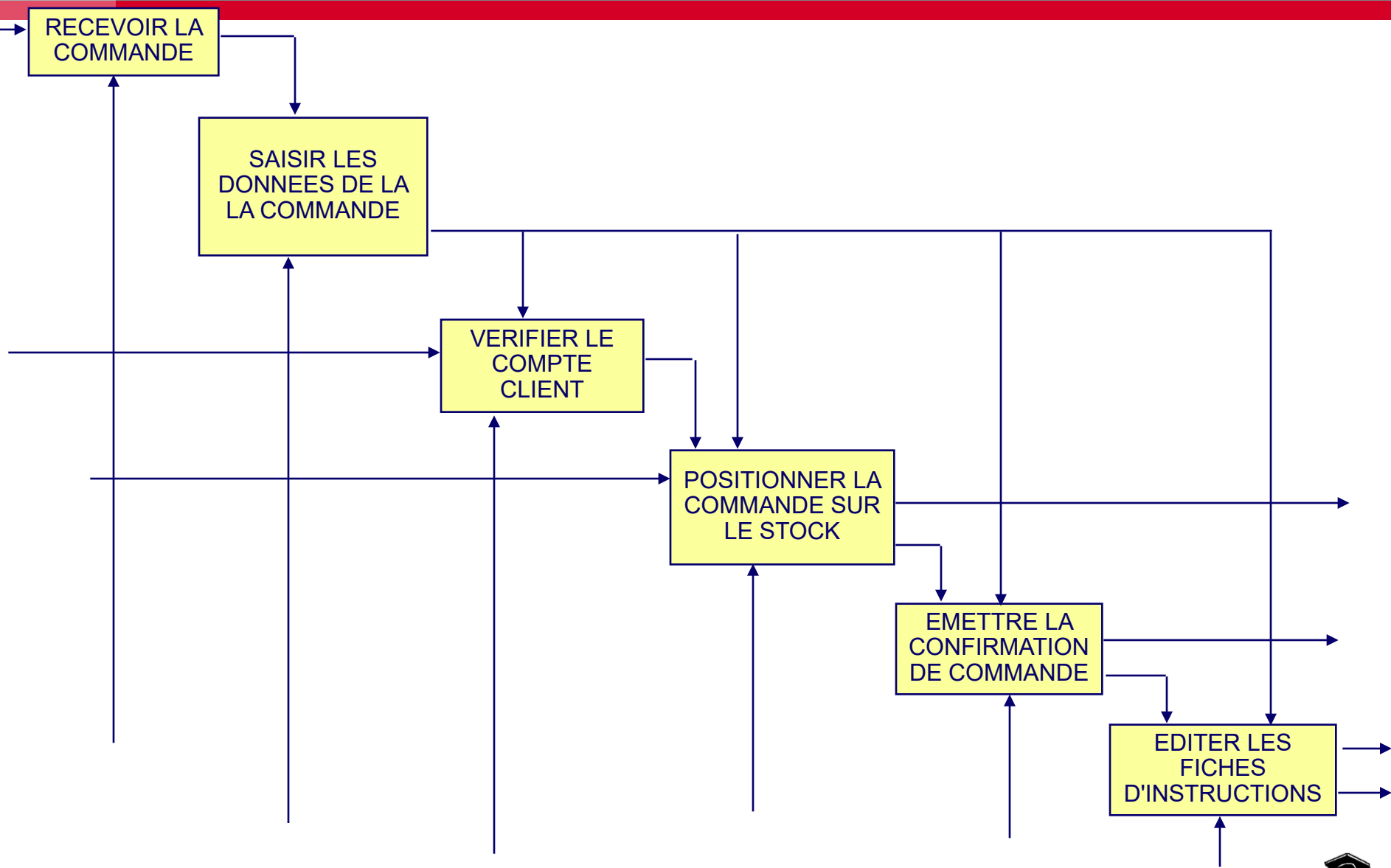




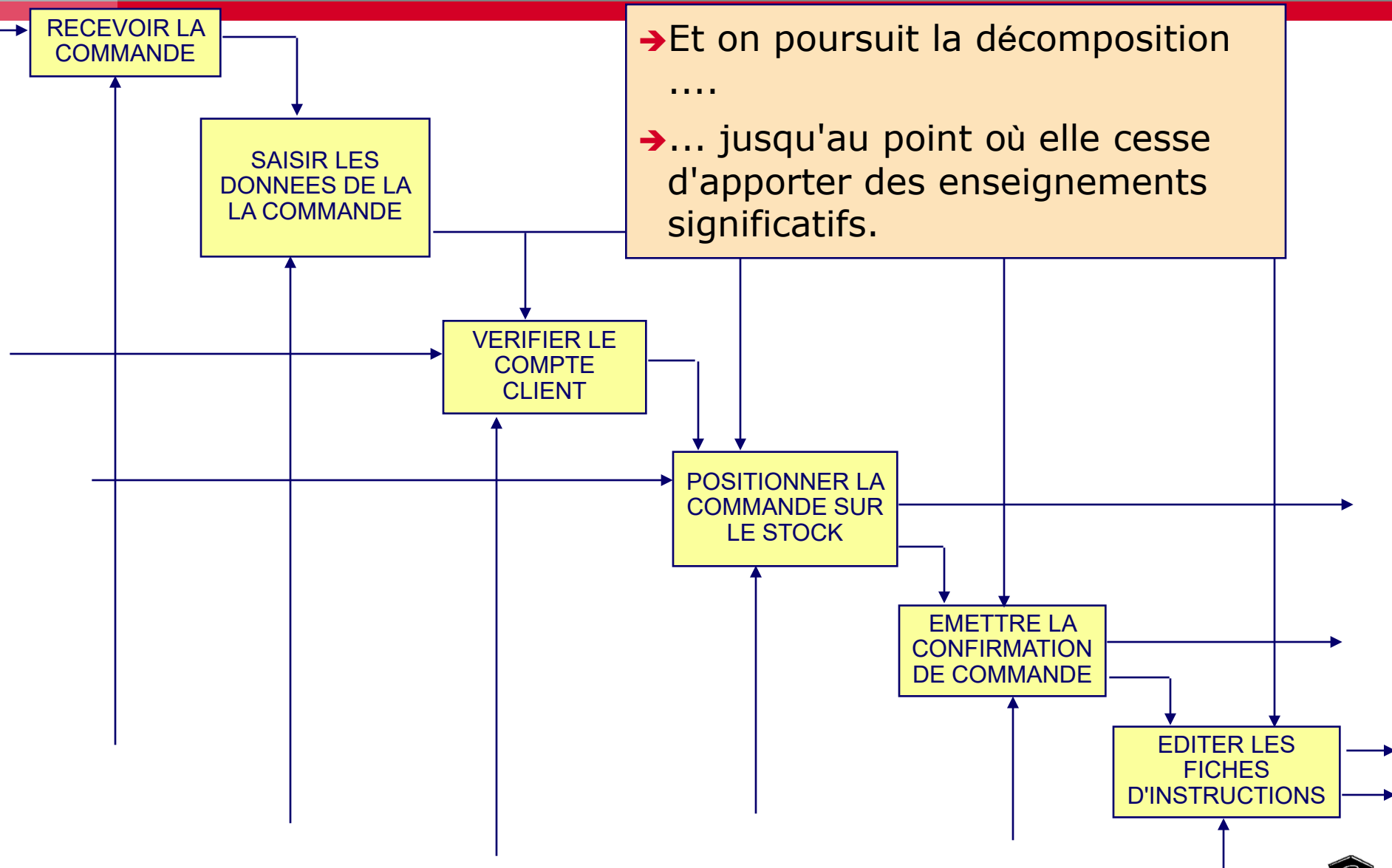
# → Poursuivre la démarche



# → Poursuivre la démarche



# → Poursuivre la démarche

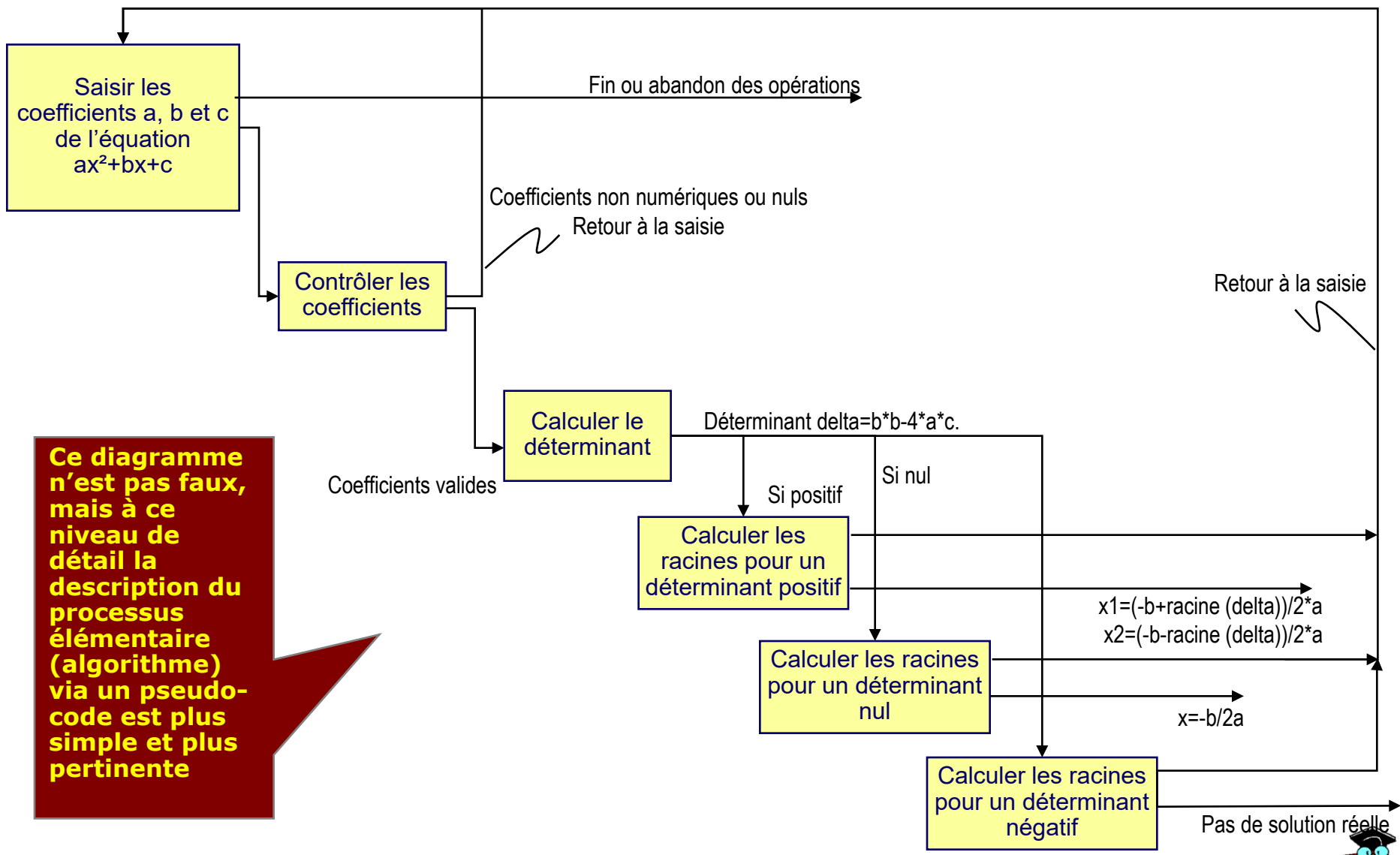


→ Et on poursuit la décomposition  
....  
→ ... jusqu'au point où elle cesse d'apporter des enseignements significatifs.





# → Savoir s'arrêter



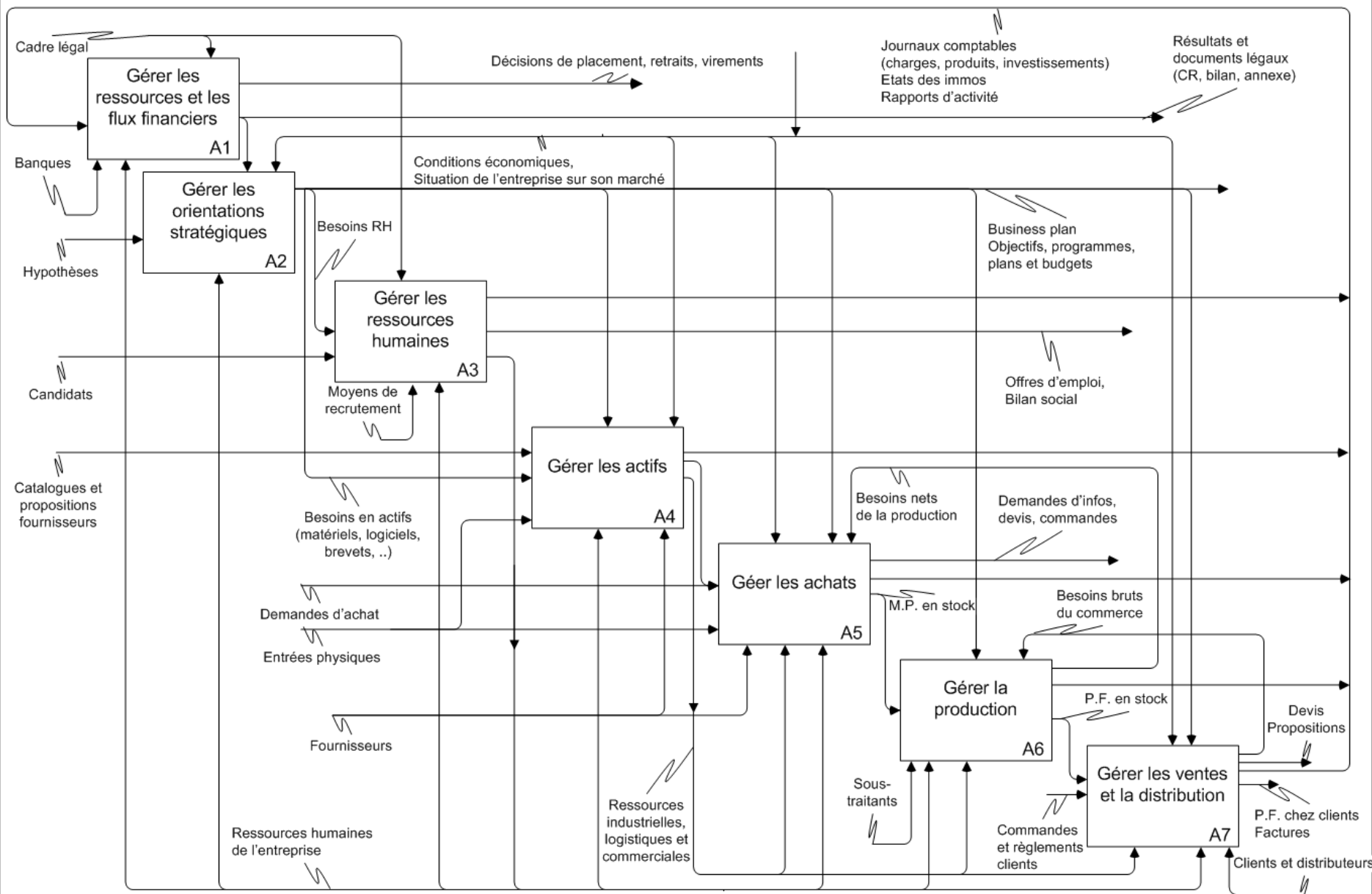
**Ce diagramme n'est pas faux, mais à ce niveau de détail la description du processus élémentaire (algorithme) via un pseudo-code est plus simple et plus pertinente**





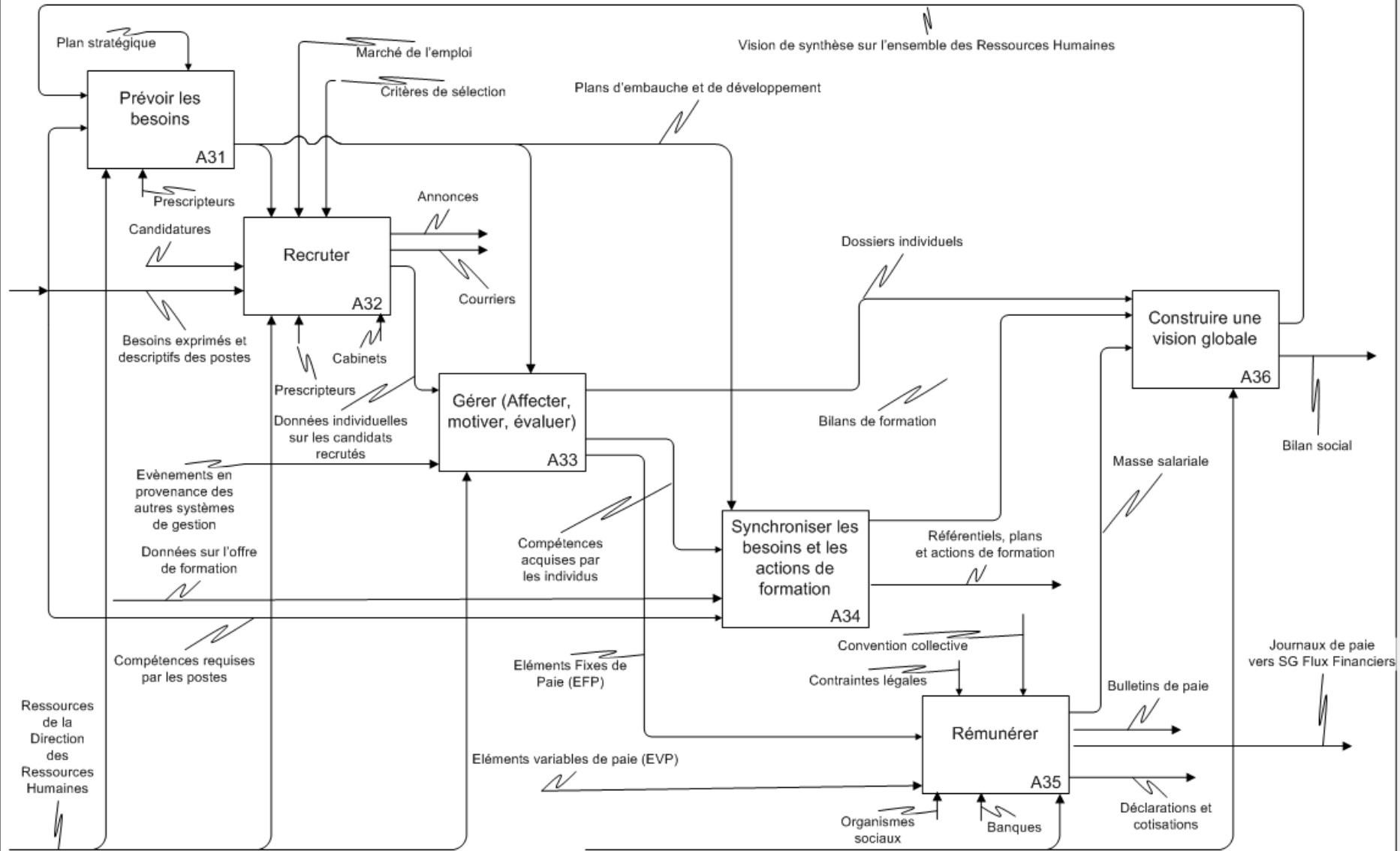
## Exemples de diagrammes SADT/IDEF réalisés avec un EDI professionnel





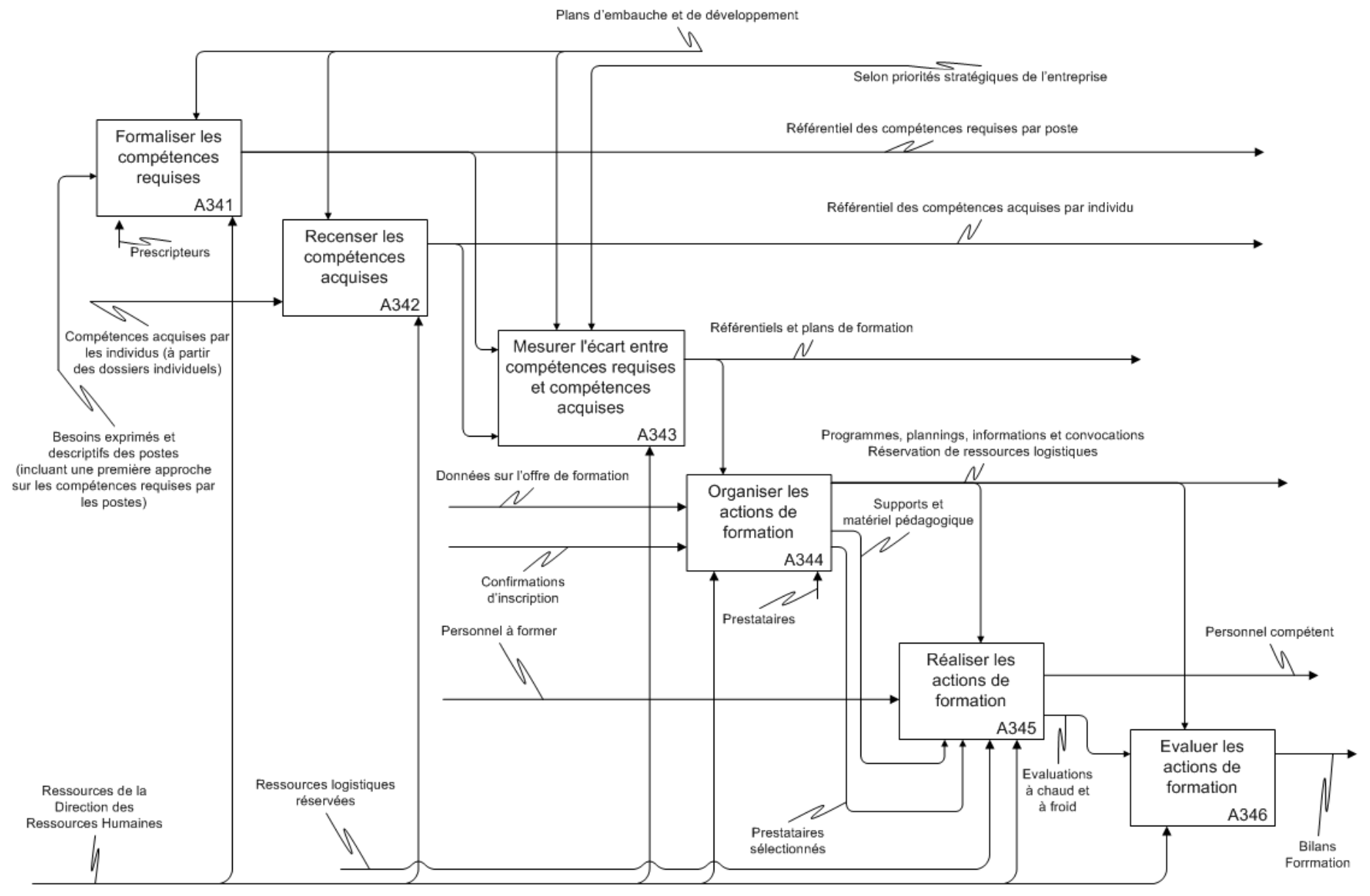
NOEUD : A0 TITRE : GERER L'ENTREPRISE INDUSTRIELLE N° :





**A3 GERER LES RESSOURCES HUMAINES**





A34

SYNCHRONISER LES BESOINS ET LES ACTIONS DE FORMATION



# → Intérêt du modèle de processus

- Grâce au Modèle des Processus, l'organisateur peut :
  - Identifier les processus opérationnels pour édifier un référentiel capable de servir de base à une opération de "reengineering",
  - Mettre en évidence les redondances, les activités inutiles, les goulots d'étranglement,
  - Proposer de nouvelles solutions d'organisation à l'aide du même formalisme pour les faire valider par les utilisateurs.
  
- Grâce au modèle des Processus, l'informaticien peut :
  - Bâtir l'architecture générale des traitements des applications qu'il doit mettre en place,
  - Avoir une idée de la nature de ces traitements (transactionnels, différés) en fonction des contraintes qui s'exercent,
  - Avoir une idée de l'architecture du réseau en fonction de la cartographie des échanges et de la répartition des postes de travail concernés par chaque activité.



# → Quelles réponses à nos questions ?

- Quel intérêt ont les modèles pour comprendre les organisations ?
- La modélisation à des niveaux divers permet de formaliser le fonctionnement d'un processus, de recenser et décrire les informations manipulées par ces processus, d'identifier les dysfonctionnements et de trouver des solutions pour optimiser l'organisation.



# → Quelles réponses à nos questions ?

- Comment représenter les activités de gestion et les activités métier au sein de l'organisation ?
- La modélisation des processus permet de représenter les caractéristiques clés des activités de l'entreprise.





# → Quelles réponses à nos questions ?

- Quelles caractéristiques des activités recenser et décrire pour pouvoir construire un nouveau système de Gestion ?
- Finalité (objectifs);
- Nature (Tâches, travaux, actions);
- Entrées;
- Ressources (Acteurs, moyens);
- Contraintes (Contexte, règles de gestion, temps, top de déclenchement, de synchronisation);
- Alternatives de décision, mécanismes d'échange et de transaction.



# → Quelles réponses à nos questions ?

- Qu'est-ce que la méthode SADT-IDEF ?
- Des modèles pour décrire le présent et des modèles pour décrire le futur en vue d'identifier les écarts et de concevoir les meilleurs pratiques pour franchir le fossé.
- La possibilité de poursuivre l'analyse selon une démarche descendante (*Top-down*) jusqu'à un niveau de détail apte à servir de référence à la rédaction du pseudo-code.
- Un travail d'auteur où l'analyste décompose un problème dont le périmètre a été identifié, à partir d'un point de départ précis, en construisant un modèle graphique et en interagissant avec les utilisateurs concernés.

