

Lire un diagramme des exigences

Requirements diagram

Objectif du diagramme

Une exigence permet de spécifier une capacité ou une contrainte qui doit être satisfaite par le système. Elle peut spécifier une

fonction que le système réalise ou une condition de performance, de fiabilité, de sécurité, etc ...

Les exigences servent à établir un **contrat entre le client et les réalisateurs d'un système**

Types d'entités utilisés

Les exigences: Qui peuvent être soit fonctionnelles, soit techniques. Elles

contiennent :

- Un nom

<<Requirement>>
Cafetière

<<requirement>>
Cafetière

Types d'entités utilisés

Les exigences: Qui peuvent être soit fonctionnelles, soit techniques. Elles

contiennent :
- Un nom

<<Requirement>>
Cafetière

- Une description de l'exigence
- Eventuellement une priorité, une source, un risque

Text = * La cafetière
doit fournir du café
chaud pendant une
journée*

ID = *001*

<<requirement>>

Cafetière

Text = * La cafetière
doit fournir du café
chaud pendant une 1
journée

ID = *001*

On décompose la « **fonction principale** » en plusieurs **fonctions secondaires** avec leurs exigences

Puis en **sous fonctions**

fonction principale

Fonctions secondaires

<<requirement>>
Alimenter

Text = *Alimenter en
café et en eau
facilement*
ID = *002*
Source = *Marketing*
Risk = *moyen*
Status = *validé*
Priorité = *haute*

<<requirement>>
Cafetière

Text = * La cafetière
doit fournir du café
chaud pendant une 1
journée ID = *001*

<<requirement>>
Chauffer

<<requirement>>
Conserver

Text = Conserver
le café chaud
pendant une
journée *
ID= *004*

<<requirement>>
Informer

Text = *Informer
lorsque la cafetière
est entartrée ID =

003

Puis en sous fonctions

<<requirement>>
Alimenter en eau

ID *002.006*

<<requirement>>
Alimenter café

ID *002.005*

<<requirement>>
Durée de conservation

Texte = *12 heures

« Bloc »
Thermos

Types d'entités utilisés

Les exigences: Qui peuvent être soit fonctionnelles, soit techniques. Elles contiennent :

- Un nom

<<Requirement>>
Cafetière

- Une description de l'exigence
- Eventuellement une priorité, une source, un risque

Text = * La cafetière doit fournir du café chaud pendant une journée*
ID = *001*

Les notes stéréotypées (commentaires particuliers) de type : -Problem (problème) : Signale un problème pas encore résolu

<<problem>>
Il n'est pas décidé si on fournit un filtre pour l'eau

- Rationale (Justification) : Justifie un choix réalisé

<<rationale>>
Il n'est pas nécessaire de garder le café chaud pendant la nuit

Cas d'utilisation bloc : Voir fiche cas d'utilisation, diagramme de définition des blocs

On décompose la « fonction principale » en plusieurs fonctions secondaires avec leurs exigences

<<requirement>>

Alimenter

Text = *Alimenter en
café et en eau
facilement*

ID = *002*

Source = *Marketing*

Risk = *moyen*

Status = *validé*

Priorité = *haute*

<<requirement>>

Cafetière

Text = * La cafetière
doit fournir du café
chaud pendant une 1
journée ID = *001*

<<requirement>>

Chauffer

<<requirement>>

Conserver

Text = Conserver
le café chaud
pendant une
journée *
ID= *004*

<<requirement>>

Informer

Text = *Informer
lorsque la cafetière
est entartrée ID =

003

Puis en sous fonctions

<<requirement>>

Alimenter en eau

ID *002.006*

<<requirement>>

Alimenter café

ID *002.005*

<<requirement>>

Durée de conservation

Texte = *12 heures

« Bloc »

Thermos

<<problem>>

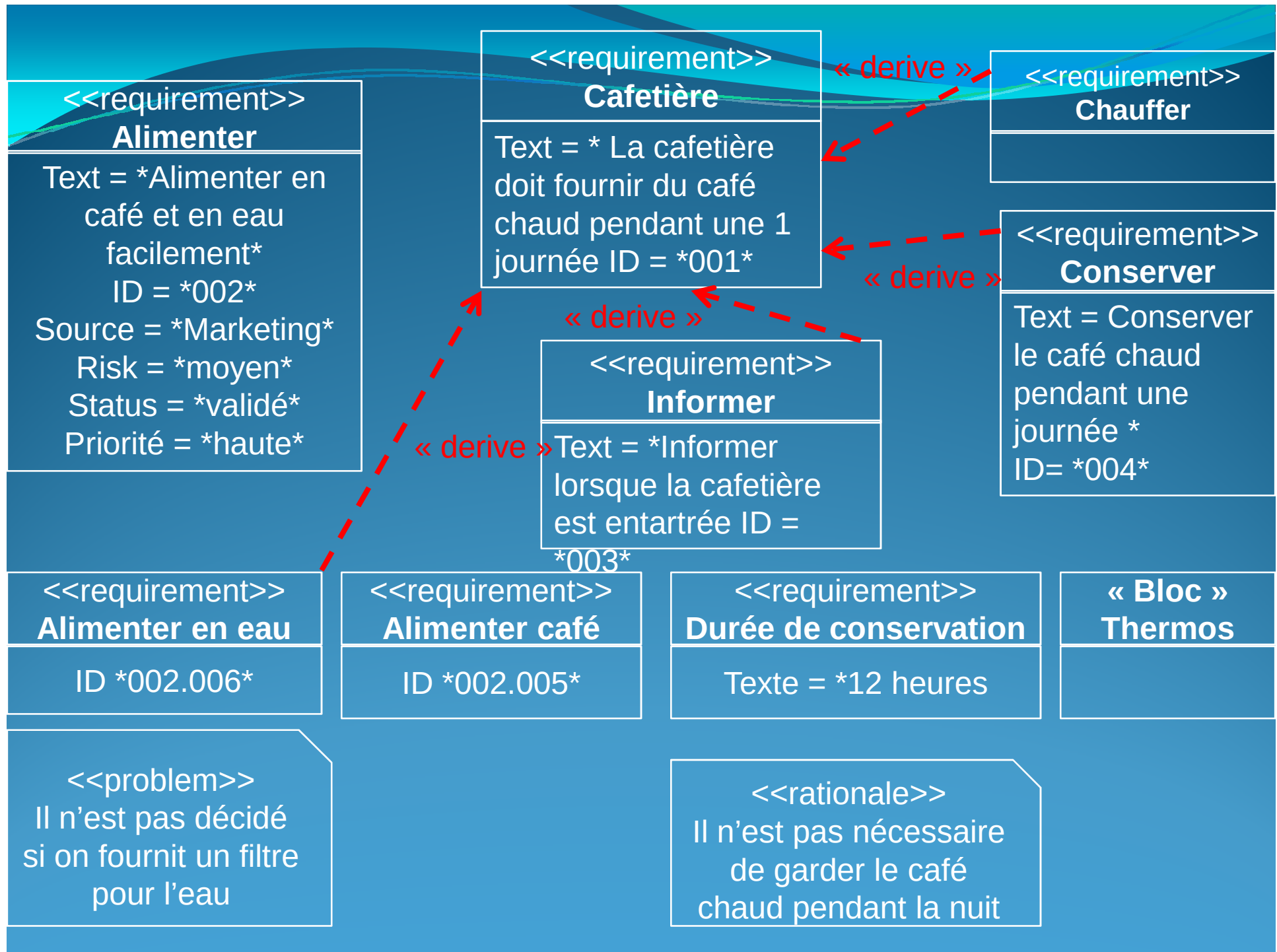
Il n'est pas décidé
si on fournit un filtre
pour l'eau

<<rationale>>

Il n'est pas nécessaire
de garder le café
chaud pendant la nuit

Types d'association (liens de dépendance)

Derive : Relie des exigences de sous systèmes à une exigence système
(ou sous système de niveau inférieur)



Types d'association (liens de dépendance)

Derive : Relie des exigences de sous systèmes à une exigence système

(ou sous système de niveau inférieur)

Contenance : Décompose une exigence en plusieurs exigences unitaires

<<requirement>>

Alimenter

Text = *Alimenter en
café et en eau
facilement*

ID = *002*

Source = *Marketing*

Risk = *moyen*

Status = *validé*

Priorité = *haute*



<<requirement>>

Cafetière

Text = * La cafetière
doit fournir du café
chaud pendant une 1
journée ID = *001*

<<requirement>>

Informer

Text = *Informer
lorsque la cafetière
est entartrée ID =

003

<<requirement>>

Chauffer

<<requirement>>

Conserver

Text = Conserver
le café chaud
pendant une
journée *
ID= *004*

<<requirement>>

Alimenter en eau

ID *002.006*

<<requirement>>

Alimenter café

ID *002.005*

<<requirement>>

Durée de conservation

Texte = *12 heures

« Bloc »

Thermos

<<problem>>

Il n'est pas décidé
si on fournit un filtre
pour l'eau

<<rationale>>

Il n'est pas nécessaire
de garder le café
chaud pendant la nuit

Types d'association (liens de dépendance)

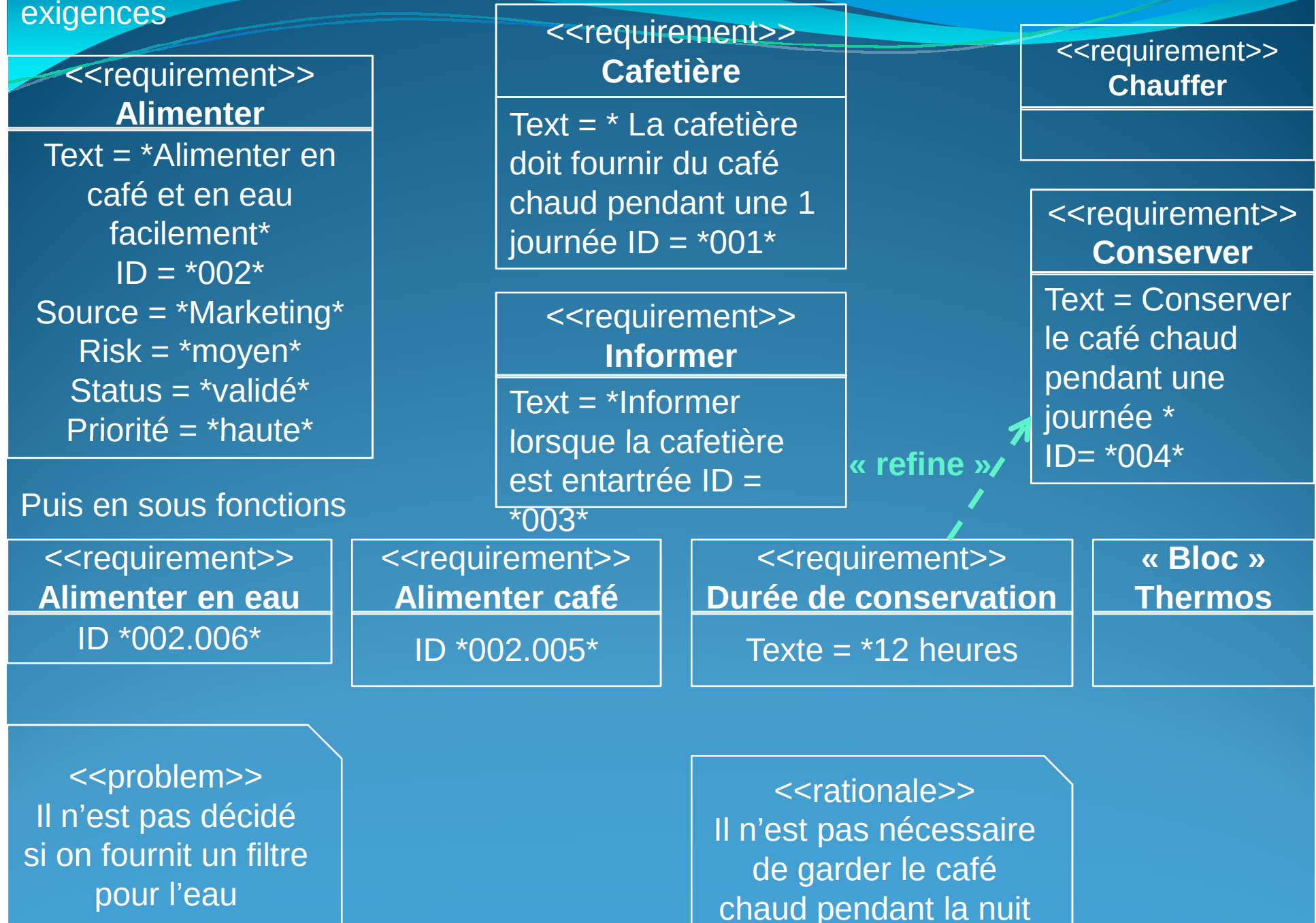
Derive : Relie des exigences de sous systèmes à une exigence système

(ou sous système de niveau inférieur)

Contenance : Décompose une exigence en plusieurs exigences unitaires

Refine (raffinement) : Donne des précisions sur l'exigence

On décompose la « fonction principale » en plusieurs fonctions secondaires avec leurs exigences



Types d'association (liens de dépendance)

Derive : Relie des exigences de sous systèmes à une exigence système

(ou sous système de niveau inférieur)

Contenance : Décompose une exigence en plusieurs exigences unitaires

Refine (raffinement) : Donne des précisions sur l'exigence

Satisfy : (par exemple un bloc) Permet de satisfaire une exigence

On décompose la « fonction principale » en plusieurs fonctions secondaires avec leurs exigences

<<requirement>>

Alimenter

Text = *Alimenter en
café et en eau
facilement*

ID = *002*

Source = *Marketing*

Risk = *moyen*

Status = *validé*

Priorité = *haute*

<<requirement>>

Cafetière

Text = * La cafetière
doit fournir du café
chaud pendant une 1
journée ID = *001*

<<requirement>>

Chauffer

<<requirement>>

Conserver

Text = Conserver
le café chaud
pendant une
journée *
ID= *004*

<<requirement>>

Informer

Text = *Informer
lorsque la cafetière
est entartrée ID =

003

Puis en sous fonctions

<<requirement>>

Alimenter en eau

ID *002.006*

<<requirement>>

Alimenter café

ID *002.005*

<<requirement>>

Durée de conservation

Texte = *12 heures

« Bloc »

Thermos

« satisfy »

<<problem>>

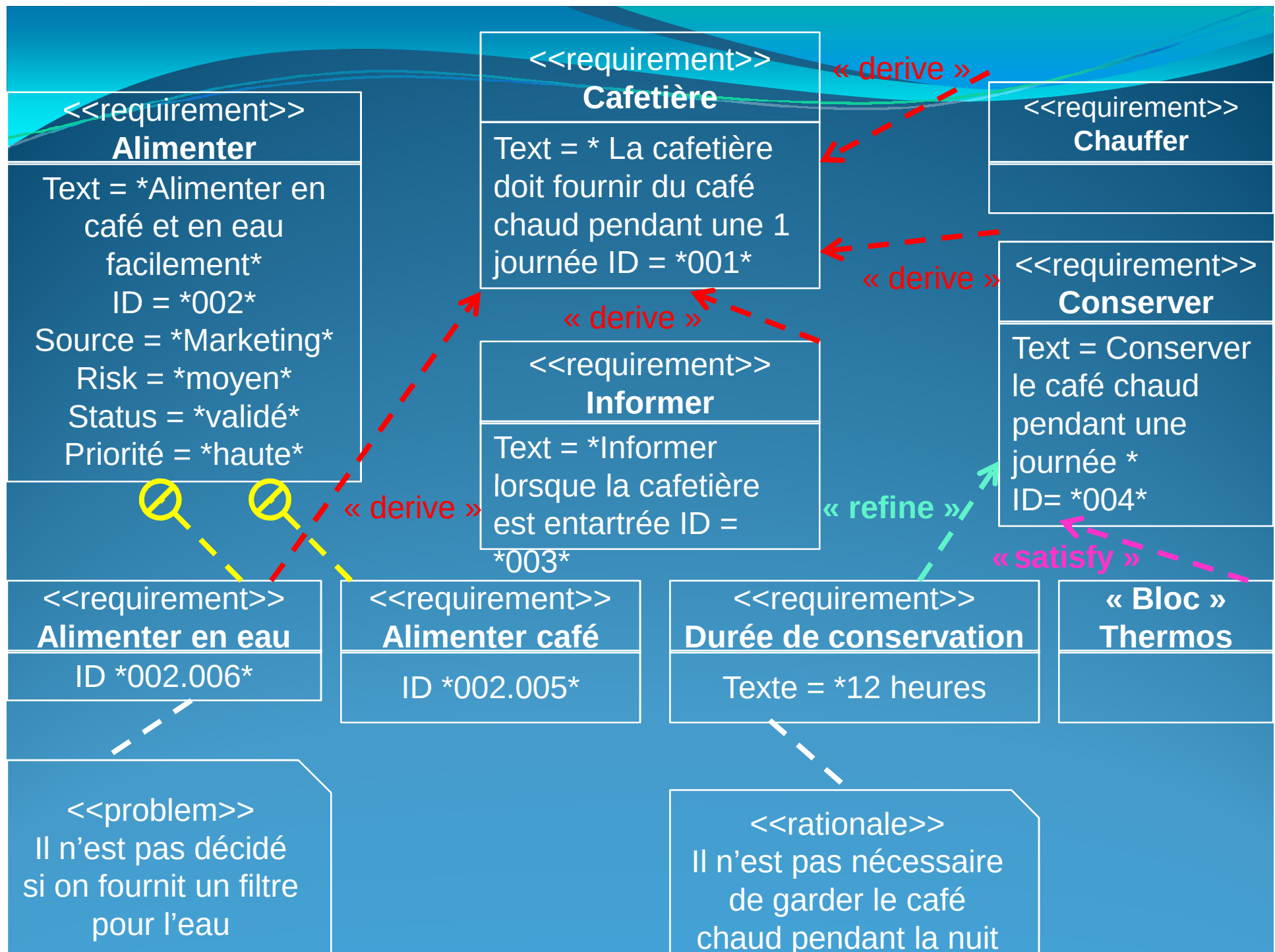
Il n'est pas décidé
si on fournit un filtre
pour l'eau

<<rationale>>

Il n'est pas nécessaire
de garder le café
chaud pendant la nuit

Types d'association (liens de dépendance)

Verify : (par exemple un block « text case ») permet de vérifier et valider une exigence



FIN

On décompose la « fonction principale » en plusieurs fonctions secondaires avec leurs exigences



