

---

# Projet Qualité Logicielle : Modèles de l'environnement

---

Équipe H4314 – 9 février 2014

**Chef de projet**

Jean-Marie COMETS

**Membres de l'équipe**

Franck MPEMBA BONI: responsable qualité

Pierre TURPIN

Samuel CARENSAC

Grégoire CATTAN

Van PHAN HAU

Iler VIRARAGAVANE

# Table des matières

<b>1</b>	<b>Système balance</b>	<b>2</b>
1.1	Système Web . . . . .	2
1.2	Système externe . . . . .	2
1.3	Capteur Balance . . . . .	2
<b>2</b>	<b>Drone</b>	<b>3</b>
2.1	tapisRoulant . . . . .	3
2.2	trappe . . . . .	3
2.3	systemeRonde . . . . .	3
2.4	listeColis . . . . .	4
2.5	balance . . . . .	4
2.6	ListeTimer . . . . .	4
2.7	systemeReceptacle . . . . .	4
<b>3</b>	<b>Réceptacle</b>	<b>5</b>
3.1	PaveNum . . . . .	5
3.2	capteurBadge . . . . .	5
3.3	diode . . . . .	5
3.4	panneauAffichage . . . . .	6
3.5	supportDrone . . . . .	6
3.6	porteLaterale . . . . .	6
3.7	panneauSuperieur . . . . .	6
3.8	capteurRFID . . . . .	6
3.9	systemeDrone . . . . .	6
3.10	systemeClient . . . . .	7
<b>4</b>	<b>Système Ronde</b>	<b>7</b>
4.1	ecranSupervision . . . . .	8
4.2	btnVisualiserDrones . . . . .	8
4.3	btnVisualiserTournees . . . . .	8
4.4	boiteMail . . . . .	8
4.5	interfaceControleClient . . . . .	8
4.6	dronePhysique . . . . .	8
4.7	alarme . . . . .	9
4.8	BD . . . . .	9
<b>5</b>	<b>Système Web</b>	<b>10</b>
5.1	BtnAssignBalance . . . . .	10
5.2	BtnConfigBalance . . . . .	10
5.3	BtnScan . . . . .	11
5.4	BtnVoirCommande . . . . .	11
5.5	BtnValiderCommande . . . . .	11
5.6	InterfaceCatalogue . . . . .	11
5.7	SystèmeBalance . . . . .	11
5.8	ListeCommande . . . . .	11
5.9	PanneauInformation . . . . .	11

# 1 Système balance

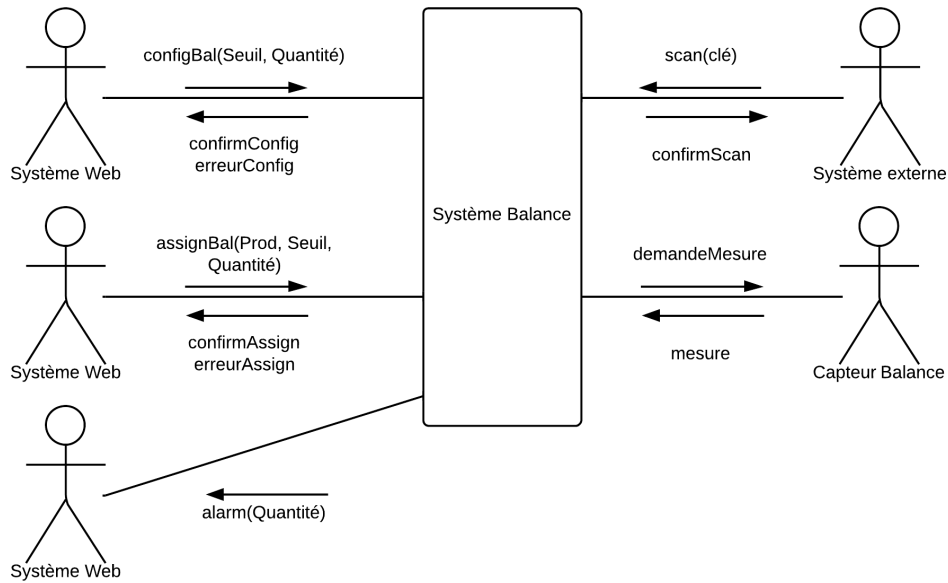


FIGURE 1 – Schéma du modèle de l'environnement "Système balance"

## 1.1 Système Web

Contrôleur principal côté client.

### Acteur du CdU : Utilisateur

- *configBal(Seuil, Quantité)* : Demande de configuration du seuil de quantité minimale *Seuil* et de la quantité de produit à racheter *Quantité*.
- *confirmConfig()* : Confirmation de la configuration réussite du système balance
- *erreurConfig()* : Indication qu'une erreur à eu lieu lors de la configuration du système balance (en général cela se produit lorsque les paramètres ne sont pas valides)
- *assignBal(Produit, Seuil, Quantité)* : Demande d'assignement d'un nouveau produit *Produit* avec une nouvelle configuration en accord avec le produit *Seuil* et *Quantité*.
- *confirmAssign()* : Confirmation de l'assignement réussi du système balance
- *erreurAssign()* : Indication qu'une erreur à eu lieu lors de l'assignement du nouveau produit (en général cela se produit lorsque les paramètres ne sont pas valides)
- *alarm(Quantité)* : Envoie une alarme au système web pour notifier le manque d'un produit.

## 1.2 Système externe

Contrôleur principal côté client.

### Acteur du CdU : Système web

- *scan()* : Demande d'identification de la part d'un contrôleur externe (liaison entre les deux toujours pas établie). Répond par une confirmation si la clé du système balance correspond à celui du contrôleur.
- *confirmScan()* : Confirme l'existence du système balance au système contrôleur.

## 1.3 Capteur Balance

Capteur mesurant le poids.

### Acteur du CdU : Système Balance

- *demandeMesure()* : Demande au capteur de fournir la valeur de la mesure.
- *mesure()* : Le capteur renvoie la valeur de la mesure. La balance la traite en la transformant en grandeur physique (valeur utilisable).

## 2 Drone

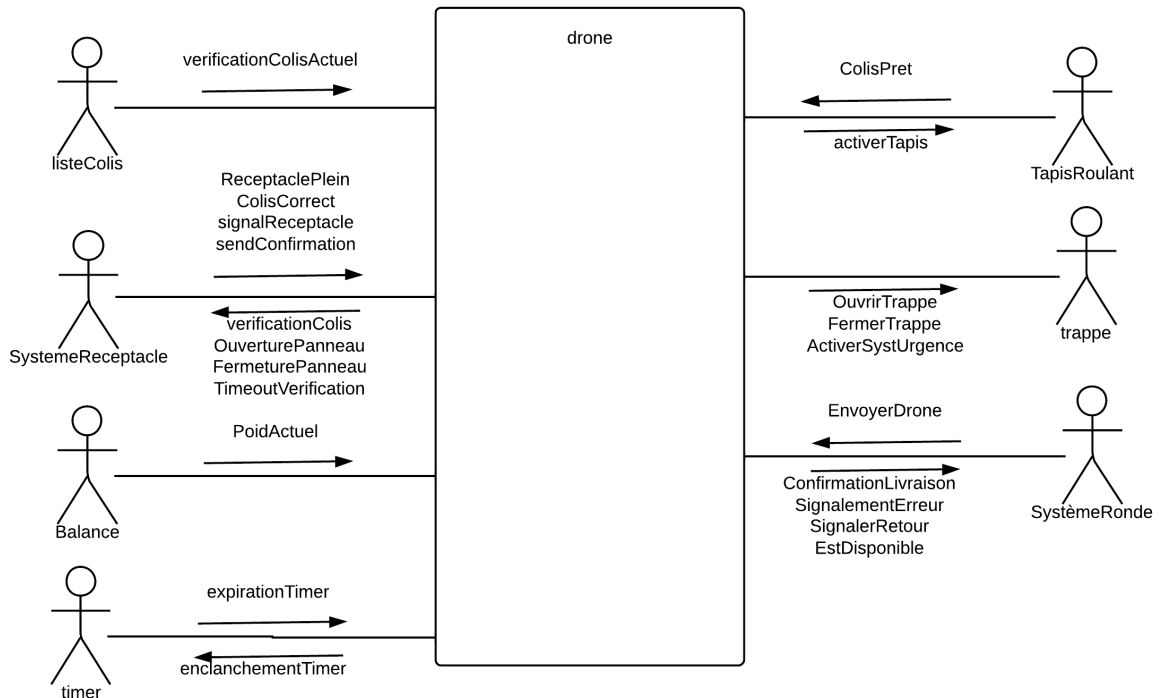


FIGURE 2 – Schéma du modèle de l'environnement "Drone"

### 2.1 tapisRoulant

Tapis roulant permettant de faire circuler les colis à l'intérieur du drone (notamment pour les déposer).

#### Acteur du CdU : Drone

- *colisPret()* : Signale que le prochain colis est en position pour le dépôt.
- *activerTapis(e)* : Si *e* est vrai, active le tapis roulant. Sinon, arrête le tapis roulant.

### 2.2 trappe

Trappe se situant sous le drone permettant de déposer les colis. Une système de sécurité peut se fermer au cas où le système se bloque en position ouverte.

#### Acteur du CdU : Drone

- *OuvrirTrappe()* : Demande du drone à la trappe pour que celle-ci passe en position ouverte.
- *FermerTrappe()* : Demande du drone à la trappe pour que celle-ci passe en position fermée.
- *ActiverSystUrgence()* : Demande du drone de fermer le système d'urgence de la trappe.

### 2.3 systemeRonde

Système de l'entreprise ronde. Comprend tout ce qui est relatif à l'entrepôt.

#### Acteur du CdU : Système Ronde

- *EnvoyerDrone(itineraire)* : Demande au drone de partir de l'entrepôt et d'effectuer la tournée correspondant à l'itinéraire prévu.
- *ConfirmationLivraison()* : Signal du drone à l'entrepôt pour signaler qu'il a bien effectué sa livraison.
- *SignalementErreur(msg)* : Signal du drone à l'entrepôt pour indiquer qu'une erreur est intervenu lors de la tournée (Drone tombe en panne, réceptacle plein, ...).

- *SignalerRetour()* : Signal du drone à l'entrepôt indiquant que le drone a fini sa tournée et est de retour à l'entrepôt.
- *EstDisponible(e)* : Si e est vrai, signal du drone à l'entrepôt indiquant que le drone est disponible pour effectuer une tournée. Sinon, indique que le drone ne peut pas être choisis pour effectuer une tournée.

## 2.4 listeColis

Liste de tous les colis contenus dans le drone.

### Acteur du CdU : Drone

- *verificationColisActuel(idColis)* : Renvoie les informations du colis en première position.
- *getCurColis()* : Demande du drone pour recevoir les informations du colis se trouvant en position de livraison sur le tapis roulant.

## 2.5 balance

Une balance contenue dans le drone permettant de connaître le poids total des colis actuellement contenus dans le drone.

### Acteur du CdU : Drone

- *PoidActuel(n)* : Signal de la balance vers le drone indiquant le poids total actuel des colis.

## 2.6 ListeTimer

Une liste de timers permettant au drone d'estimer si le réceptacle est tombé en panne.

### Acteur du CdU : Drone

- *expirationTimer(idTimer)* : Indique qu'un timer est arrivé à expiration (ce qui signifie que le réceptacle ne nous a pas répondu).
- *enclenchementTimer(timer)* : Met en place un timer correspondant aux informations passées en paramètres.
- *desactiverTimer(command)* : Désactive le timer correspondant à la commande.

## 2.7 systemeReceptacle

Se référer à la section 3.

### 3 Réceptacle

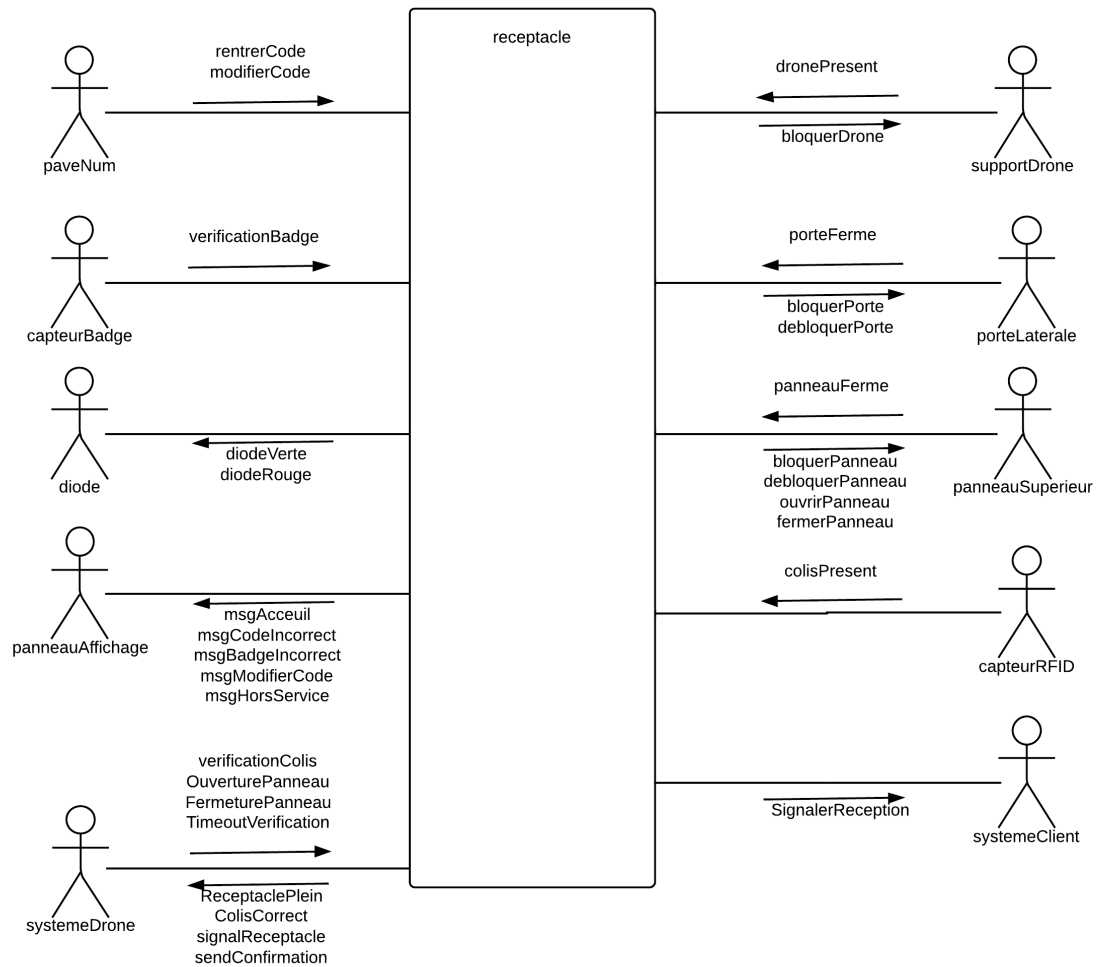


FIGURE 3 – Schéma du modèle de l'environnement "Réceptacle"

#### 3.1 PaveNum

Pavé numérique type standard, composé de chiffres (valeur allant de 0 à 9), d'un bouton "valider", et d'un bouton "annuler".

##### Acteur du CdU : Utilisateur

- *rentrerCode(n)* : Le code *n* est vérifié.
- *modifierCode(n)* : Le code *n* est fixé en tant que nouveau code.

#### 3.2 capteurBadge

Capteur de badge pour détecter la présence du badge de l'utilisateur (ou celui du technicien).

##### Acteur du CdU : Utilisateur

- *verificationBadge(b)* : La valeur du badge est vérifiée.

#### 3.3 diode

Diode pouvant prendre une couleur verte ou rouge pendant un temps donné.

##### Acteur du CdU : Utilisateur

- *diodeVerte()* : Demande à la diode de devenir verte pour 3 secondes.
- *diodeRouge()* : Demande à la diode de devenir rouge pour 3 secondes.

### 3.4 panneauAffichage

Panneau d'affichage pour l'affichage d'informations générales sur l'état du système.

#### Acteur du CdU : Utilisateur

- *msgAccueil()* : Demande d'affichage du message d'accueil.
- *msgCodeIncorrect(e)* : Si *e* est vrai, demande d'affichage d'un message indiquant que le code rentré est incorrect. Sinon, demande du réceptacle d'effacer le message.
- *msgBadgeIncorrect(e)* : Similaire à *msgCodeIncorrect(e)*.
- *msgModifierCode(e)* : Similaire à *msgCodeIncorrect(e)*. L'effacement est suivi d'une demande d'affichage d'un message demandant la fermeture.
- *msgHorsService(e)* : Similaire à *msgCodeIncorrect(e)*.

### 3.5 supportDrone

Drone

#### Acteur du CdU :

- *dronePresent()* : Le support indique que le drone est présent.
- *bloquerDrone(e)* : Si *e* est vrai, une demande est envoyée au support pour bloquer le drone. Sinon, une demande est envoyée au support pour relâcher le drone.

### 3.6 porteLaterale

Porte latérale du réceptacle, utilisée par le client pour récupérer les colis. Ne peut être ouverte que manuellement.

#### Acteur du CdU : Utilisateur

- *porteFerme()* : Signale au réceptacle que la porte vient d'être fermée.
- *bloquerPorte()* : Demande du réceptacle de bloquer la porte latérale.
- *dbloquerPorte()* : Demande du réceptacle de débloquent la porte latérale.

### 3.7 panneauSuperieur

Panneau supérieur du réceptacle, utilisé par le drone pour déposer les colis. Peut être ouverte électroniquement ou manuellement.

#### Acteur du CdU : Drone

- *panneauFerme()* : Signale au réceptacle que le panneau supérieur vient d'être fermé.
- *bloquerPanneau()* : Demande du réceptacle de bloquer le panneau supérieur.
- *debloquerPanneau()* : Demande du réceptacle de débloquent le panneau supérieur.
- *ouvrirPanneau()* : Demande du réceptacle d'ouvrir du panneau.
- *fermerPanneau()* : Demande du réceptacle de fermer le panneau supérieur.

### 3.8 capteurRFID

Un capteur RFID permettant de vérifier la présence d'un colis à l'intérieur du réceptacle.

#### Acteur du CdU : aucun

- *colisPresent()* : Le capteur indique s'il a détecté la présence d'un colis à l'intérieur du réceptacle.

### 3.9 systemeDrone

Correspond au système du drone en liaison avec le réceptacle.

#### Acteur du CdU : Drone

- *verificationColis(idDestinataire)* : Signal du drone demandant la vérification de la correspondance entre un colis et le réceptacle.
- *ouverturePanneau()* : Signal du drone demandant l'ouverture du panneau supérieur du réceptacle.
- *fermeturePanneau()* : Signal du drone demandant la fermeture du panneau supérieur du réceptacle.

- *timeoutVerification()* : Signal du drone indiquant que celui-ci n'a pas reçu la confirmation de l'action demandée dans le temps voulu.
- *receptaclePlein()* : Signal du réceptacle indiquant au drone que le réceptacle est plein.
- *colisCorrect(e)* : Si *e* est vrai, signale que le colis correspond bien à ce réceptacle. Sinon, signale que le colis ne correspond pas à ce réceptacle.
- *signalReceptacle()* : Signal du réceptacle pour indiquer au drone la position du réceptacle pour que celui-ci se pose.
- *sendConfirmation(msg)* : Signal du receptacle pour indiquer qu'une action a bien été réalisée. Cette action est indiqué par le message *msg* (sous la forme d'une énumération).

### 3.10 systemeClient

#### Acteur du CdU : Système web

- *signalerReception()* : Signal du réceptacle au système client pour indiquer qu'il y a un colis à récupérer dans le réceptacle.

## 4 Système Ronde

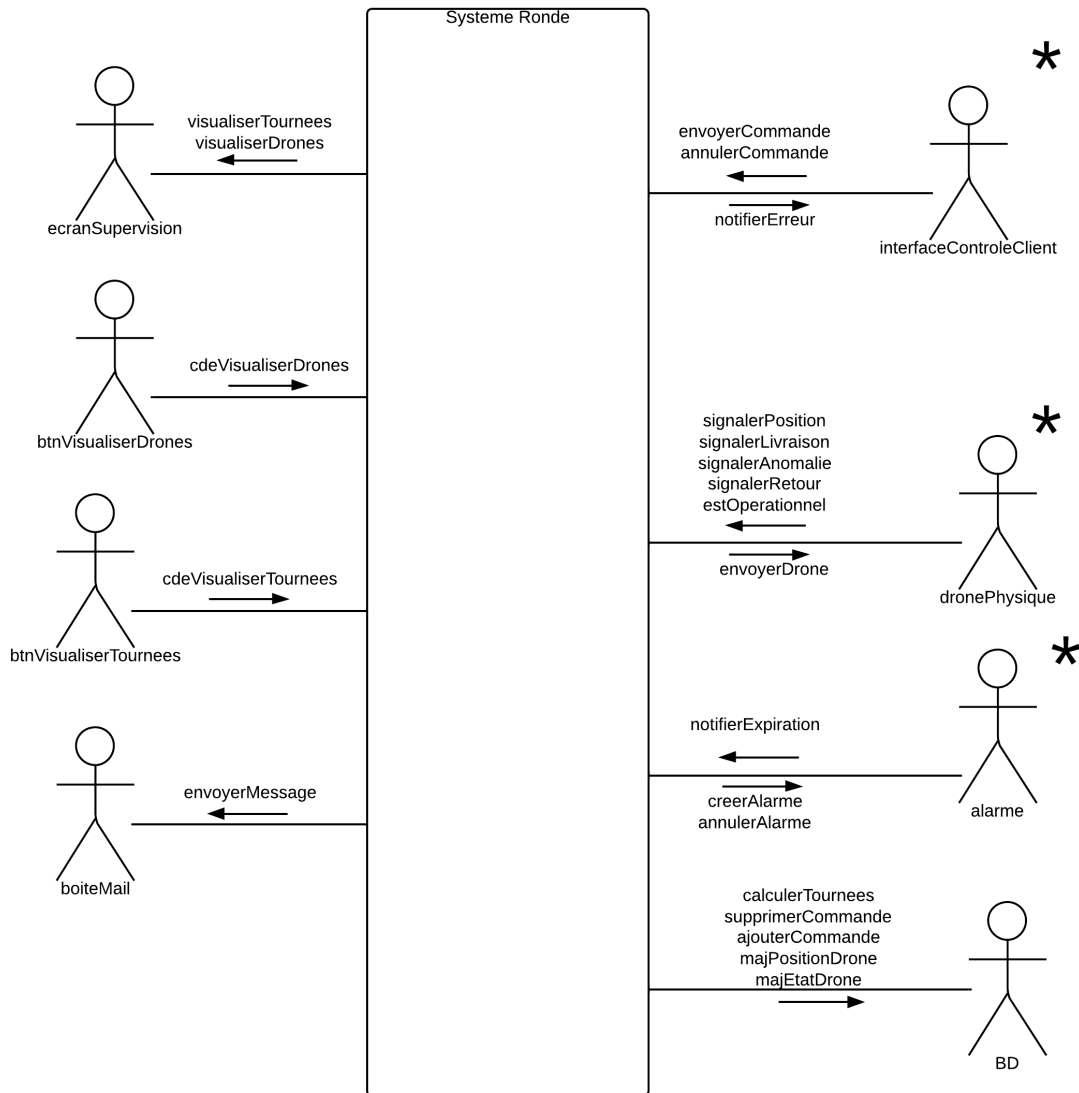


FIGURE 4 – Schéma du modèle de l'environnement "Système Ronde"



## 4.1 ecranSupervision

Un écran d'affichage où l'on peut visualiser les tournées, leur avancement, et la position des drones qui leur sont rattachées.

### Acteur du CdU : Superviseur

- *visualiserDrone()* : Visualise la position des drones sur l'écran, ainsi que leur état (marche, alerte).  
CdU concerné : *Superviser ronde*
- *visualiserTournée()* : Visualise les tournées et leur avancement sur l'écran  
CdU concerné : *Superviser ronde*

## 4.2 btnVisualiserDrones

Un bouton de l'interface qui permet d'afficher les drones sur *ecranSupervision*.

### Acteur du CdU : Superviseur

- *cdeVisualiserDrones()* : Le bouton a été actionné.  
CdU concerné : *Superviser ronde*

## 4.3 btnVisualiserTournées

Un bouton de l'interface qui permet d'afficher les tournées sur *ecranSupervision*.

### Acteur du CdU : Superviseur

- *cdeVisualiserTournées()* : Le bouton a été actionné.  
CdU concerné : *Superviser ronde*

## 4.4 boiteMail

Une application qui gère l'affichage ainsi que le traitement des messages.

### Acteur du CdU : Superviseur

- *envoyerMessage(idDrone, Message)* : Le système envoie le message *Message* venant du drone *idDrone* à la *boiteMail* (quitter entrepôt, en tournée, livraison impossible, alerte, ...).  
CdU concerné : *Superviseur ronde*

## 4.5 interfaceControleClient

L'interface de l'application permettant de réceptionner les messages d'un client.

### Acteur du CdU : Client

- *envoyerCommande(Commande)* : Envoi de la commande *Commande* au système ronde.  
CdU concerné : *Livrer une tournée*
- *annulerCommande(Commande)* : Demande l'annulation de la commande *Commande*.  
CdU concerné : *Livrer une tournée*
- *notifieErreur(Message)* : Préviens le client que sa demande ne peut être satisfaite (absence de produit demandé, annulation survenue trop tard, ...), en spécifiant le message *Message*.  
CdU concerné : *Livrer une tournée*

## 4.6 dronePhysique

Un drone du système ronde

### Acteur du CdU : Drone

- *signalerPosition(Position)* : Envoie les coordonnées GPS *Position* du drone au système.  
CdU concerné : *Livrer une tournée*
- *signalerLivraison(Commande)* : Signale au système que la commande *Commande* a bien été livrée.  
CdU concerné : *Déposer une commande*
- *signalerAnomalie(Anomalie)* : Signale au système une anomalie (panne, attaque, impossibilité de livrer une commande, ...). Une anomalie *Anomalie* peut être déclenchée manuellement si elle est constatée par un technicien du système.  
CdU concerné : *Livrer une tournée*

- *signalerRetour()* : Le drone signale au système qu'il est bien revenu.  
CdU concerné : *Livrer une tournée*
- *estOperationnel()* : Le drone signale au système qu'il est disponible (à l'entrepôt et en bon état).  
CdU concerné : *Livrer une tournée*
- *envoyerDrone(Tournee)* : Le système demande au drone de commencer la tournée *Tournee*.  
CdU concerné : *Livrer une tournée*

## 4.7 alarme

Une alarme associée à une tournée, qui prévient le système si la tournée risque de ne pas être livrée à temps.

### Acteur du CdU : Alarme

- *notifierExpiration(Tournee)* : Signale au système qu'un élément de la tournée *Tournee* risque de ne pas être livré à temps si l'on attend plus longtemps.  
CdU concerné : *Livrer une tournée*
- *creerAlarme(Tournee, Temps)* : Le système associe une alarme à la tournée *Tournee* et demande son expiration dans *Temps* heures.  
CdU concerné : *Livrer une tournée*
- *annulerAlarme()* : Détruit l'alarme avant qu'elle n'expire.  
CdU concerné : *Livrer une tournée*

## 4.8 BD

Base de données contenant toutes les commandes en cours de livraison (ou en attente) ainsi que leur tournée correspondante.

### Acteur du CdU : Ronde

- *calculerTournee()* : Recalcule les tournées (et éventuellement crée les nouvelles).  
CdU concerné : *Livrer une tournée*
- *ajouterCommande(Commande)* : Ajoute la commande *Commande* dans la *BD*.  
CdU concerné : *Livrer une tournée*
- *supprimerComande(Commande)* : supprimer des commandes dans la *BD*.  
CdU concerné : *Livrer une tournée*
- *majPositionDrone(idDrone, Position)* : Met à jour la position GPS à *Position*, pour le drone identifié par *idDrone*, dans la *BD*.  
CdU concerné : *Livrer une tournée*
- *majEtatDrone(idDrone, Anomalie)* : Met à jour l'état, pour le drone identifié par *idDrone*, à l'état *Anomalie*.  
CdU concerné : *Livrer une tournée*

## 5 Système Web

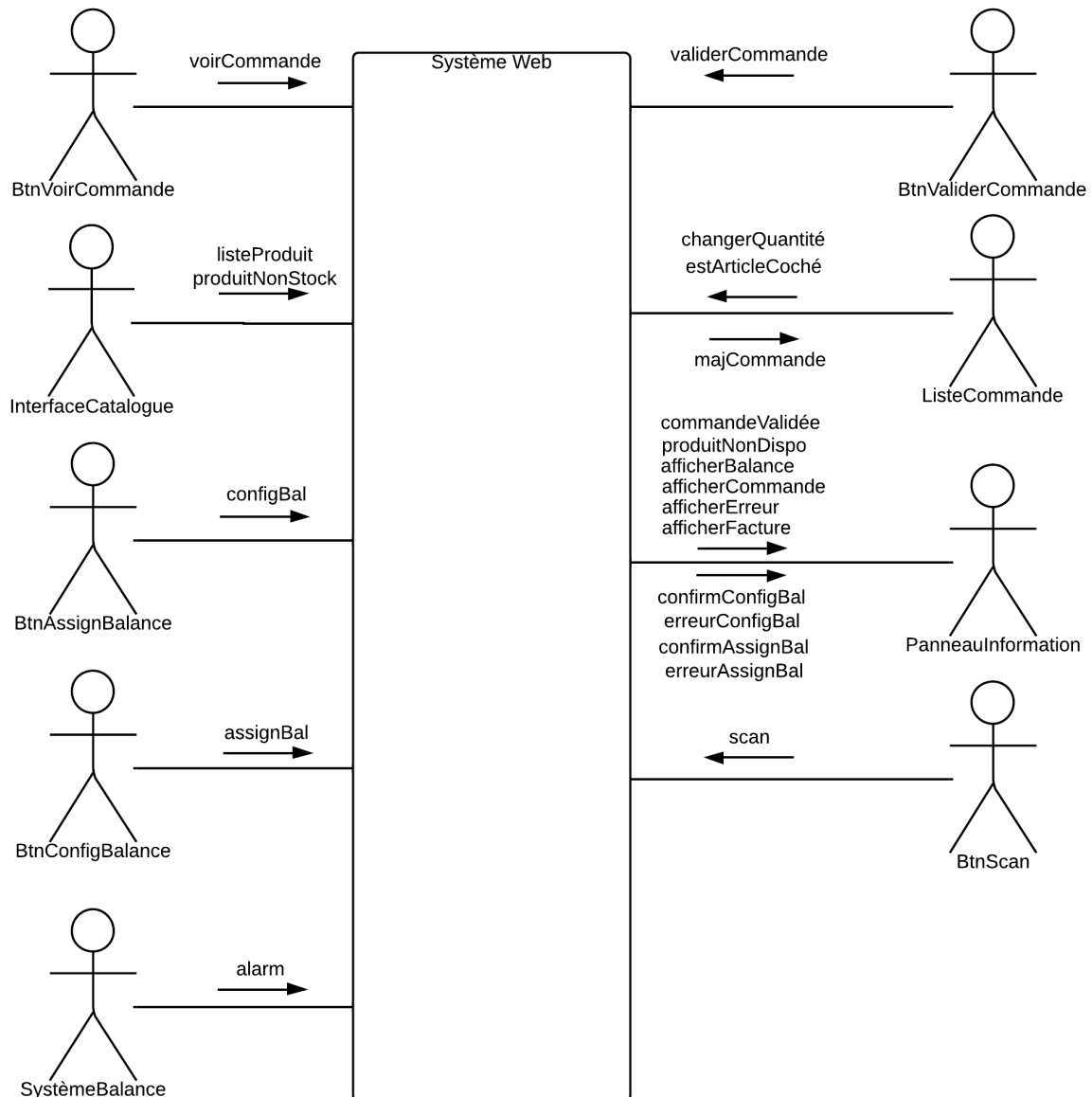


FIGURE 5 – Schéma du modèle de l'environnement "Système Web"

### 5.1 BtnAssignBalance

Un bouton pour assigner un produit à une balance existante.

#### Acteur du CdU : Utilisateur

- *assignBal(bal, art, seuil, qte)* : Le bouton a été activé pour la balance *bal* avec l'article *art* et comme configuration le couple *(seuil, qte)*, respectivement le seuil et la quantité.  
CdU concerné : *Configurer les balances*

### 5.2 BtnConfigBalance

Un bouton pour configurer une balance

#### Acteur du CdU : Utilisateur

- *configBal(bal, seuil, qte)* : Le bouton a été activé pour la balance *bal* avec la configuration *(seuil, qte)*.  
CdU concerné : *Configurer les balances*

### 5.3 BtnScan

Un bouton pour identifier les balances associées au client

#### Acteur du CdU : Utilisateur

- *scan()* : Demande d'identification de la part d'un contrôleur externe (liaison entre les deux toujours pas établie). Répond par une confirmation si la clé du système balance correspond à celui du contrôleur.  
CdU concerné : *Configurer les balances*

### 5.4 BtnVoirCommande

Un bouton pour afficher la liste des commandes courante associée à l'utilisateur.

#### Acteur du CdU : Utilisateur

- *voirCommande()* : Le bouton à été actionné.  
CdU concerné : *Valider une commande*

### 5.5 BtnValiderCommande

Un bouton pour valider la liste des commandes courante associée à l'utilisateur.

#### Acteur du CdU : Utilisateur

- *validerCommande()* : Le bouton à été actionné.  
CdU concerné : *Valider une commande*

### 5.6 InterfaceCatalogue

Interface client pour ajouter un produit à la liste des commandes de l'utilisateur.

#### Acteur du CdU : Utilisateur, Controleur

- *listeArticle()* : La liste des produits est envoyée au client.  
CdU concerné : *MAJ d'une commande*
- *produitNonStock(art)* : L'article *art* n'est plus en stock.  
CdU concerné : *MAJ d'une commande*

### 5.7 SystèmeBalance

L'interface balance qui notifie un manque en quantité de l'article.

#### Acteur du CdU : Balance, Controleur

- *alarm(art, qte)* : Une quantité *qte* est demandée pour l'article *art* associé à la balance afin d'être ajoutée à la commande.  
CdU concerné : *MAJ d'une commande*

### 5.8 ListeCommande

La liste des commandes de l'utilisateur divisée en articles, avec chacun ses caractéristiques.

#### Acteur du CdU : Utilisateur, Controleur

- *estArticleCoch(art)* : Un article *art* de la liste des commande a été coché ou décoché.  
CdU concerné : *Valider une commande*
- *changerQuantit(art, qte)* : Demande de modification de la quantité de l'article *art* par la quantité *qte*.  
CdU concerné : *Valider une commande*
- *majCommande()* : Demande du *SystmeWeb* de mettre à jour la liste des commandes de l'utilisateur.  
CdU concerné : *Valider une ommande, MAJ d'une commande*

### 5.9 PanneauInformation

Un écran pour afficher les différentes informations sur l'état du système (du texte, des lumières, des graphiques, des boutons, ...).

#### Acteur du CdU : Utilisateur, Controleur

- *afficherErreur(err)* : Demande du *SystmeWeb* d'afficher l'erreur *err* rencontrée.  
CdU concerné : *Valider une ommande, MAJ d'une commande*
- *produitNonDispo(art,b)* : Si *b* est vrai, le *SystmeWeb* demande l'affichage d'un texte annonçant que l'article *art* n'est pas disponible en stock. Sinon, le *SystmeWeb* n'affiche rien.  
CdU concerné : *MAJ d'une commande*
- *commandeValide(b)* : Si *b*, le *SystmeWeb* demande l'affichage d'un texte annonçant que la commande a été validée. Sinon, le *SystmeWeb* n'affiche rien.  
CdU concerné : *Valider une commande*
- *afficherBalance(bal)* : Demande du *SystmeWeb* d'afficher la balance *bal* sélectionnée depuis la liste des balances de l'utilisateur.  
CdU concerné : *Configurer les balances*
- *afficherCommande(cmd)* : Demande du *SystmeWeb* d'afficher la liste des commandes *cmd* de l'utilisateur.  
CdU concerné : *Valider une commande*
- *afficherFacture(cmd,prixTotal)* : Demande du *SystmeWeb* d'afficher la facture relative à la commande *cmd* avec le prix total *prixTotal* liée à la commande.  
CdU concerné : *Valider une commande*
- *confirmAssignBal()* : L'assignement du produit a bien eu lieu.  
CdU concerné : *Configurer les balances*
- *erreurAssignBal()* : Une erreur a eu lieu pendant l'assignement de la balance.  
CdU concerné : *Configurer les balances*
- *confirmConfigBal()* : La configuration a bien eu lieu.  
CdU concerné : *Configurer les balances*
- *erreurConfigBal()* : Une erreur a eu lieu durant la configuration.  
CdU concerné : *Configurer les balances*