

# SmartFridge

## 1. Vision produit (Smart Fridge x Drive)

### But

- Ne jamais oublier d'acheter un produit du quotidien.
- Automatiser la liste de courses sans interaction humaine.
- Gagner du temps et enlever la charge mentale ("faut que je pense au lait").

### Comment ça marche

1. Le frigo surveille son contenu en temps réel (caméra interne + IA de vision).
2. Quand un produit disparaît ou passe en "niveau bas", l'info remonte.
3. L'appli alimente automatiquement le panier de courses du drive (ex : E.Leclerc Drive).
4. L'utilisateur n'a plus qu'à valider ou modifier avant commande.

### Valeur client directe

- Plus de rupture de stock chez toi.
- Plus besoin de faire une liste à la main.
- Suivi de ta consommation (tu vois ce qui part vite → budget, diète, gaspillage).
- Alerte péremption potentielle (optionnel).

### Positionnement

→ Frigo connecté milieu/haut de gamme, orienté confort et automatisation des courses familiales.

Formule à garder pour le jury :

| "On ne vend pas juste un frigo, on vend un assistant d'achats alimentaires."

---

## 2. Fonctionnalités majeures

### 2.1. Caméra interne + IA de reconnaissance visuelle

- Caméras intégrées à l'intérieur du frigo.

- L'IA détecte les produits (ex : "bouteille de lait demi-écrémé 1L").
- L'IA suit la quantité ("pack yaourt x8 → reste 1").
- L'IA détecte le retrait produit : si tu prends le dernier yaourt et tu le sors, le système comprend que c'est vidé, pas juste "temporairement sorti 10 secondes".

Points critiques réalistes :

- Les restes maison / Tupperware non identifiés → pas de réassort auto dessus.
- Les produits "non Drive-compatible" (plat cuisiné maison) → ignorés dans le panier automatique.

Traduction simple : on ne réapprovisionne automatiquement QUE ce qui est achetable chez le drive.

---

## 2.2. Gestion du seuil de réapprovisionnement

- Chaque produit a un seuil ("si reste < 1 brique de lait → ajoute au panier Drive").
- Les seuils peuvent être personnalisés par l'utilisateur.
- Le système apprend la routine du foyer au fil du temps (fréquence à laquelle un article disparaît).

C'est là que tu vends le côté intelligent :

■ "Le frigo apprend comment TU consommes."

---

## 2.3. Synchronisation panier Drive

- Connexion au Drive (premier scénario : E.Leclerc).
- Dès qu'un produit passe sous son seuil, il est automatiquement ajouté au panier "Prochaines courses".
- L'utilisateur ouvre l'app → le panier est prêt.
- En un clic : validation + choix du créneau de retrait.

Donc le gain n'est pas juste "liste de courses", c'est "commande quasi prête sans réfléchir".

---

## 2.4. Historique / traçabilité

- Journal horodaté du type :
  - "17/10 18:22 – Beurre ajouté (détection rupture)."

- Sert à :
  - comprendre pourquoi le frigo a rajouté un truc,
  - corriger si l'IA s'est trompée.

Utile pour la confiance utilisateur (sinon ils vont dire "il commande tout seul sans me demander ?!").

---

## 2.5. Mode manuel override

- L'utilisateur peut blacklister un produit.
  - Exemple : "Stop le Coca, même si le frigo croit qu'on n'en a plus, tu n'en remets PAS tout seul."
- Période de pause produit ("Dry January" → pas de bière même si stock = 0).
- Option pour dire "je passe moins de beurre en ce moment → ne réapprovisionne pas automatiquement".

But : montrer qu'on laisse le contrôle humain final. Très important pour adoption.

---

## 2.6. Optionnel (roadmap) : gestion des dates de péremption

- Scan de la DLUO/DLC soit via vision, soit via scan code-barres manuel.
- Notification : "Attention, jambon périmé demain."
- Argument marketing : réduction gaspillage alimentaire = économie + image "responsable".

À garder pour la fin de soutenance en mode "phase 2 du produit".

---

# 3. Architecture technique (version courte et compréhensible pour le jury)

## 3.1. Côté frigo (IoT embarqué)

- Caméra(s) HD grand angle capables d'opérer à basse T°, gérer la condensation.
- Petit module de calcul dans le frigo (edge AI). Exemple réaliste : NVIDIA Jetson Nano / Raspberry Pi 5.
- Le module fait la détection d'objets en local.

Pourquoi c'est important :

→ pas besoin d'envoyer de vidéo brute dans le cloud → respect vie privée, moins de bande passante.

→ argument RGPD qu'on ressort en boucle.

- Connexion Wi-Fi pour synchroniser l'état du frigo avec l'extérieur.

## 3.2. Cloud

- Le frigo n'envoie pas des images.
- Il envoie uniquement des métadonnées du type :
  - "Produit : Lait 1L Marque Repère, Quantité estimée : 0"
  - "Statut : RUPTURE"
- Le cloud met à jour l'inventaire virtuel du frigo pour cet utilisateur.

En gros : dans le cloud, on sait ce qu'il te manque en temps réel.

## 3.3. Application mobile

- Affiche l'inventaire actuel du frigo.
- Affiche la liste des articles "à acheter bientôt".
- Montre le panier Drive en cours.
- Permet d'enlever / modifier avant validation commande.

Ça rassure les gens qui flippent d'un truc 100 % automatique :

→ "Tu restes le dernier mot."

## 3.4. Intégration partenaire Drive (E.Leclerc en premier)

- On mappe ce que l'IA voit ("lait demi-écrémé 1L Marque Repère") vers une vraie référence vendable en Drive (EAN / SKU).
- Le service ajoute automatiquement cette référence dans le panier Drive de l'utilisateur.
- L'app te propose le panier final prêt à valider.

Point sensible (donc à dire au tuteur, ça montre qu'on n'est pas naïfs) :

→ Il nous faut l'accès au catalogue produit Drive (références exactes, marques, formats).

C'est une contrainte business réelle : partenariat / API.

---

## 4. Ressources humaines (personas pour `groupname_skills` )

Tu mets UNE PAGE PDF par rôle. Je te les écris comme ça :

### 1. Computer Vision / IA Engineer

- Mission : entraîner et améliorer le modèle de détection d'objets et estimer les quantités.
- Compétences : vision par ordinateur, classification d'images, détection d'objets temps réel, optimisation edge.

### 2. Embedded Systems / IoT Engineer

- Mission : intégrer caméra + module IA dans le frigo, gérer le Wi-Fi, la robustesse dans un environnement froid/humide.
- Compétences : électronique embarquée, firmware bas niveau, Linux embarqué, sécurité IoT.

### 3. Mobile / Backend Developer

- Mission : développer l'app mobile (inventaire, panier Drive) + backend cloud + synchro Drive.
- Compétences : React Native / API REST / gestion auth / stockage sécurisé utilisateur.

### 4. Business & Partenariats Retail

- Mission : négocier l'accès catalogue Drive, aligner les références produits, convaincre un partenaire pilote (Leclerc, Carrefour...).
- Compétences : négociation B2B, connaissance retail, stratégie go-to-market.

### 5. Chef de Projet / Product Owner

- Mission : tenir le planning, le budget, la gestion des risques, la com interne/externe, et coordonner tous les autres.
- Compétences : gestion de projet, Gantt, reporting, arbitrage priorités.

### 6. Sécurité / Privacy Officer (même temps partiel au début)

- Mission : RGPD, politique de données, pas de fuite d'images, conformité si on envoie de la donnée utilisateur vers des partenaires.
- Compétences : cybersécurité IoT, chiffrement, conformité CNIL/UE, rédaction de politique de confidentialité claire.

→ Ces 6 rôles couvrent tech + business + conformité.

→ Tu balances ça comme `smartfridge_skills.pdf` .

---

## 5. Risques majeurs (pour `groupname_risks` )

### Risque #1 : Mauvaise reconnaissance produit

- Problème : l'IA confond "lait entier bio Carrefour" avec "lait demi-écrémé Marque Repère".
- Impact : mauvais articles commandés → l'utilisateur perd confiance direct.  
Impact : Élevé.
- Probabilité : Moyenne au début.
- Mitigation : apprentissage supervisé (l'utilisateur peut corriger "c'est pas ça"), amélioration du dataset avant déploiement large.
- Impact planning : rallonger les tests utilisateurs avant industrialisation.

### Risque #2 : Dépendance à un seul partenaire Drive

- Problème : si Leclerc ferme son API ou refuse l'accès, votre killer feature saute.
- Impact : Énorme sur le modèle économique.
- Probabilité : Moyenne.
- Mitigation : dès le départ, concevoir l'intégration comme un microservice multi-retailers (Leclerc / Carrefour / Intermarché).
- Impact budget : plus de dev backend + plus de négociation commerciale.

Tu dois dire ça en soutenance. Ça fait "on comprend le risque business, pas juste la techno".

### Risque #3 : Vie privée / RGPD

- Problème : perception utilisateur : "y'a une caméra chez moi qui bosse pour un supermarché".
- Impact : Adoption marché. Très Élevé.
- Probabilité : Élevée (les gens sont méfiants).
- Mitigation :
  - traitement IA en local (edge AI),
  - aucune image brute envoyée au cloud,
  - communication claire : "Le frigo ne stocke pas ta vie, il compte des yaourts."

- Impact communication : il faut créer du matériel de communication rassurant. Ça doit apparaître dans `smartfridge_communication2.pdf`.
- 

## 6. Planning projet (base de Gantt pour `groupname_gantt.xlsx`)

### Phase 1 — Proof of Concept (PoC)

- Étude marché / concurrents.
  - Définition fonctionnelle (comment on détecte, quand on ajoute au panier).
  - Analyse faisabilité technique (caméra interne, IA embarquée, Wi-Fi).
  - Premier contact retail pour accès catalogue Drive.
- Livrable interne : "Proof of Concept".

### Phase 2 — Prototype / Pilote

- Construction proto frigo modifié avec caméra + module IA.
  - Développement backend + appli mobile (inventaire en live + panier auto).
  - Intégration API Drive test (ajout d'articles auto).
  - Tests chez quelques familles pilotes.
- Livrable interne : "Prototype / Pilot (appli en bêta)".

### Phase 3 — Industrialisation / Commercialisation

- Intégration propre dans un frigo de série (industrialisation hardware).
  - Verrouillage légal / RGPD / conformité CE.
  - Négociation commerciale avec les enseignes.
  - Préparation de la communication marketing : "Le frigo qui pense pour toi."
- Livrable interne : "Industrialisation / Commercialisation".

Important pour le Gantt :

- Mets aussi vos vraies tâches d'équipe (budget, risques, com interne/externe).
  - Montre que le Gantt évolue (ils insistent sur l'itératif).
  - Montre qui est responsable de quoi.
-

## 7. Stack techno (ce que vous dites en soutenance pour montrer que c'est crédible)

C'est ça qu'il faut retenir par cœur :

### 1. Dans le frigo

- Caméra HD interne.
- Module IA embarqué (Jetson Nano / Raspberry Pi 5).
- Reconnaissance d'objets en local avec un modèle type YOLO optimisé (PyTorch → TensorRT).
- Petit service Python (FastAPI ou Flask) qui maintient l'état "inventaire actuel".

→ Ça reste en local, donc pas de fuite d'images.

### 2. Cloud backend

- Backend FastAPI (Python).
- Base PostgreSQL.
- Stockage des inventaires, des seuils, des historiques d'ajout panier.
- Génération automatique d'un panier Drive à jour.

### 3. Microservice d'intégration Drive

- Service dédié qui parle à l'API Drive Leclerc (et demain Carrefour / Intermarché).
- Traduit "lait vu dans le frigo" → "référence vendable Drive".
- Ajoute au panier Drive de l'utilisateur.
- Tech : Node.js ou Python, peu importe, mais isolé pour limiter le risque business.

### 4. Application mobile

- React Native (iOS + Android).
- Fonctionnalités :
  - Voir l'inventaire du frigo.
  - Voir ce qui manque.
  - Voir le panier Drive auto-généré.
  - Valider/refuser les ajouts.
  - Gérer les seuils et blacklister des produits.



## 5. Sécurité / RGPD

- HTTPS obligatoire.
- Authentification par token (JWT).
- Pas d'upload d'images brutes vers le cloud.
- Les données appartiennent à l'utilisateur.
- Journal d'action horodaté ("on a ajouté le lait à telle heure parce qu'il était à 0").

Tu dis ça et on te prend au sérieux, pas pour un touriste.