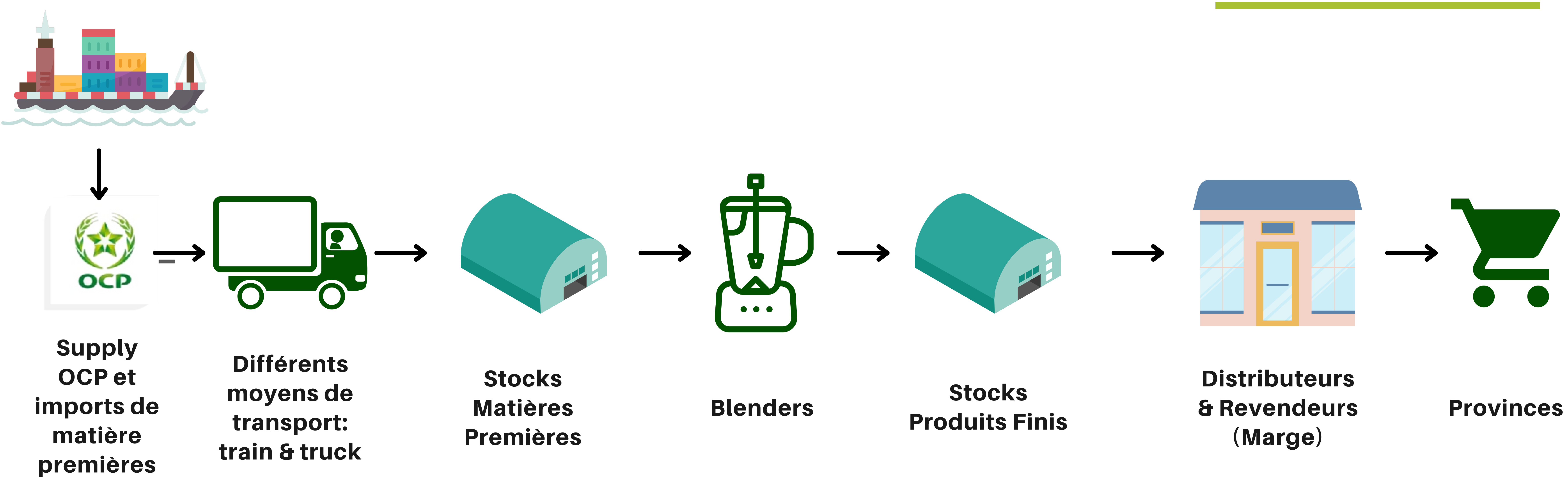
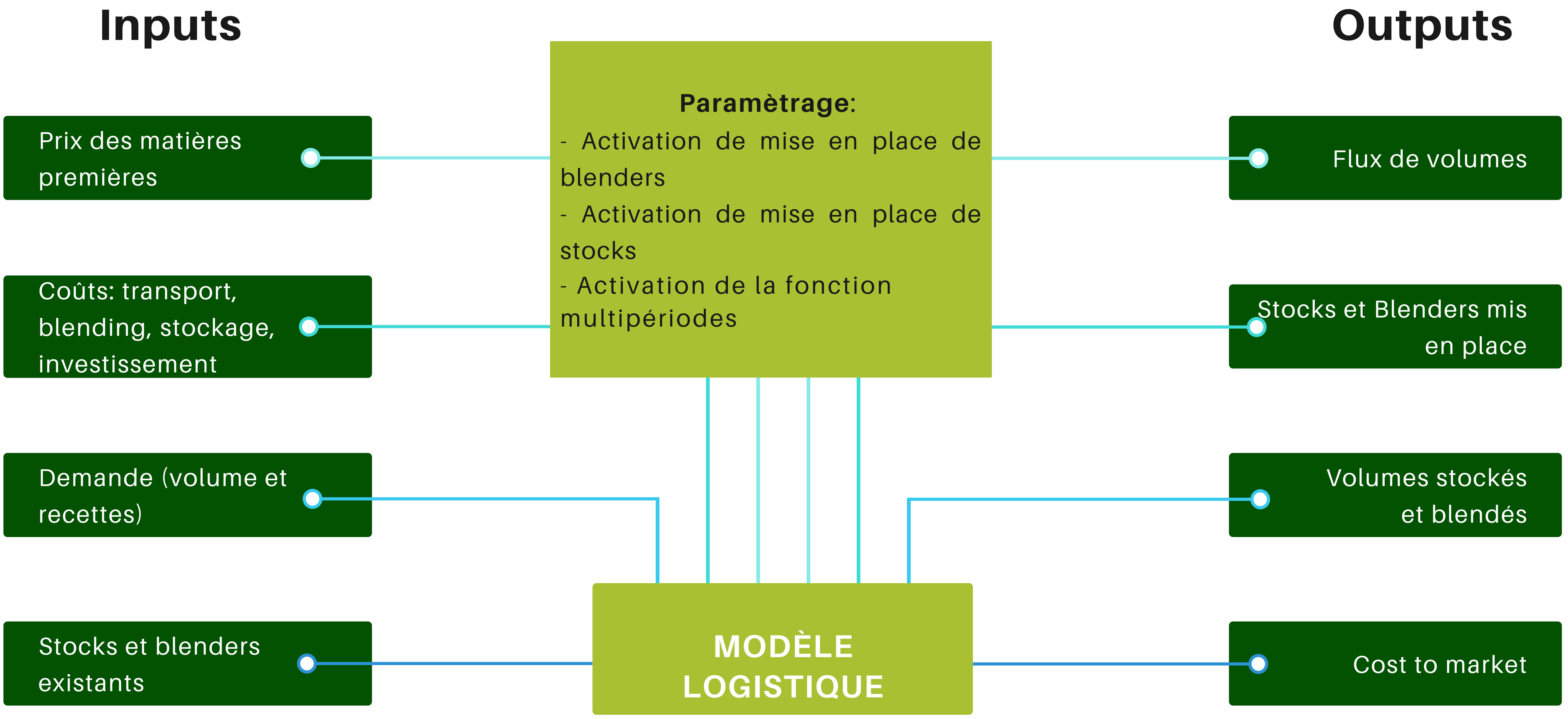
# Modèle logistique Marché local

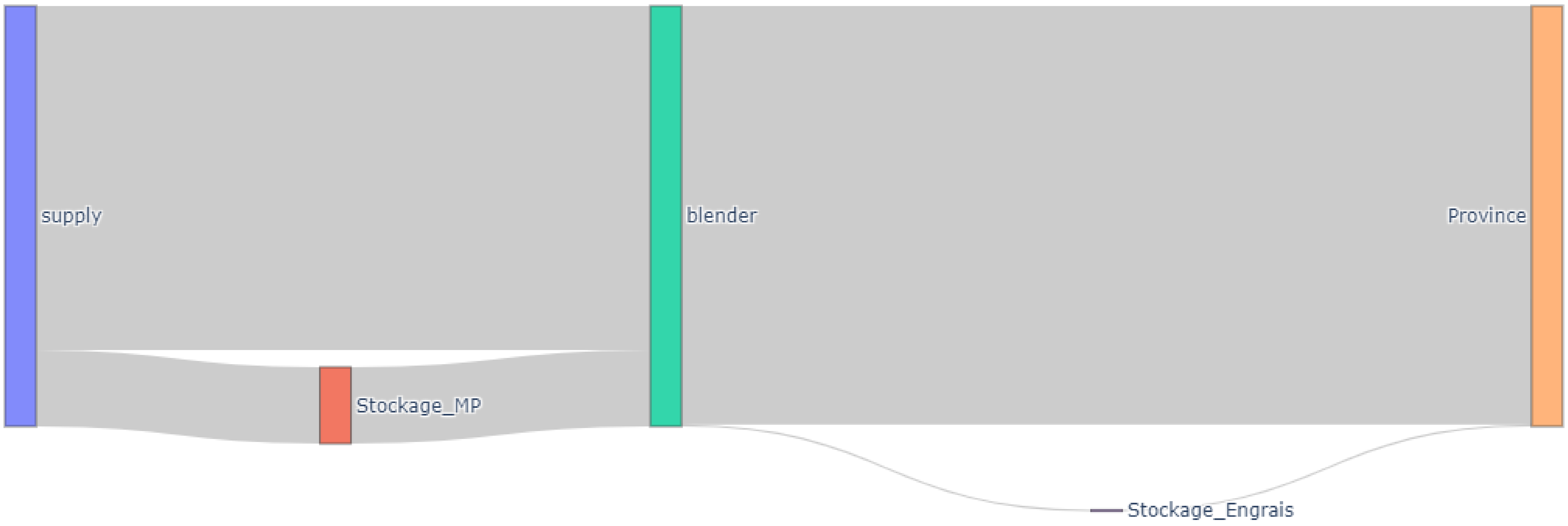
# Chaîne Logistique modélisée



# Structure du modèle actuellement



# Résultats principaux du modèle - Les flux



# Used Data

- The raw materials and their prices (TSP, DAP, MAP, MOP, EM, CAN, DURAMON 28%) are the raw materials that come out of OCP and with which we constitute the fertilizer recipes dedicated to each province and each crop.

- The upstream unit costs are the costs per ton between the source or the Supply (port/factory) and the province (blender / Raw material storage)

- Unit downstream costs are the costs per ton between the province (blender / MP storage / fertilizer storage) and the province (blender / MP storage / fertilizer storage / farmer).

- Costs of blending and storage and out-of-stock.

- Existing blenders and Smart blenders with their locations and capacities

- Existing storage with their locations and capacities.

- The capabilities of the sources that the model must respect.

- The overall volume of demand by province and crop that we have restructured to be faithful to reality which is a bag sent by a blender to a (province, culture) must contain the right percentages of raw materials which gives us almost 589 fertilizers (combination (province, crop)).

# Proposed Methodology

For data size concerns, we will start with a reduced size input set: with 5 provinces and some blenders and storages.

We will make a model in several stages:

**Step 1: Fix storages and blenders and focus on minimizing overall logistics costs, for one period (8 points)**

**Step 2: Add decision variables for investment in storage and blenders (4 points)**

**Step 3: Add Multiperiod option (2 points)**

**Bonus, Step 4:**

- **Ensure mutualization of storage for both raw materials and finished products (2 points)**
- **Scale up to the whole input with 35 provinces and as well blenders and storages (2 points)**
- **Explore multi-objective optimization (2 points)**

We will provide both datasets: small size and full size

# Model Parameters

Marge distributeur: 6% | the percentage of profit for the distributors

Marge revendeur: 8%  | the percentage of profit for the re-sellers

Cout_de_rupture: 10^5  | associated cost with the unsatisfied demand

Cout_de_stockage_Engrais: 10  | Cost of stocking one unit of raw materials

Cout_de_stockage_MP: 10 | Cost of stocking one unit of final product

Capacité_stock_actif: 3500 | Capacity of new storages

Capacité_blender_actiif: 12000 | Capacity of new blenders

Capacité_smart_blender_actiif: 3000| Capacity of new smart blenders

Stockage_min: 100 | Min limit of flow to open a new storage invest in a storage

Blender_min: 500 | Min limit of flow to open a new storage invest in a blender

Cout_blender: 2600000 | Investment cost for a single blender

Cout_sblender: 380000 | Investment cost for a single smart blender

Cout_stockage_building: 10 | Investment cost for building a single storage

Cout_blending: 120 | Cost of blending in a regular blender per unit

Cout_smart_blending: 120 | Cost of blending in a smart blender per unit

# Model description - Proposed Decision variables

**For s in supply, st in Storage_MP, m in MP, b in blenders, t in periods, se in Storage_Fertilizer and e in fertilizers.**

**Continuous variables:**

·**Flow Supply à Storage MP** (**Flow of raw material from supply to a warehouse**):
Flow_Supply_StorageMP[s, st,m,t]

·**Flow Storage MP à Blender and Flow Storage MP à Smart Blender** (**Flow of raw material from a warehouse to blenders**) :
Flow_StorageMP_Blender[st,b,m,t] and Flow_StorageMP_SBlender[st,b,m,t]

·**Flow Supply à Blender and Flow Supply à Smart Blender** (**Flow of raw material from supply to a warehouse**):
Flow_Supply_Blender[s,b,m,t] and Flow_Supply_SBlender[s,b,m,t]

·**Flow Blender à Storage Fertilizer** (**Flow of final product from blenders to warehouse**) : Flow_Blender_StorageFertlizer[b,se,e,t]

·**Flow Storage Fertilizer à Province** (**Flow of final product from warehouse to province**) :
Flow_StorageFertlizer_Province[se,e,t]

·**Flow Blender à Province and Flow Smart Blender à Province** (**Flow of final product from blenders to province**) **:**
Flow_Blender_Province[b,e,t] and Flow_SBlender_Province[b,e,t]

·**Reliquat_Demand** (**The amount of unsatisfied demand**): Reliquat_Demand[e,t]

·**Stock MP** (**Stock of raw material and final product at the end of a perod**):
Stock_MP[st,m,t] and Stock_Fertlizer[se,e,t]

**Binary variables:**

·**Active Blender and Active Smart blender** (**Investment in new blenders and smart blenders per period**):
Active_Blender[b,t] and Active_SBlender[b,t]

·**Active downstream storage and Active upstream storage** (**investment in downstream and upstream warehouses per period**) :
Active_Downstream_Storage[st,t] and Active_Upstream_Storage[se,t]

·**Storage investment variable**(**mutualization of investment in new downstream and upstream storages**):
Storage_investement[st,t]

**Storage Cost** (**Investment once a time in upstream and downstream storages**):
Storage_Cost[st]

·**Blender and Smart Blender investment variables**(**Investment in new blenders and smart blenders only one time**) :
Blender_Cost[b] and SBlender_Cost[b]

# Model description - Objective function

**We want to minimize the sum of :**

1. The transportation costs
2. Blending costs
3. Storage costs
4. Investment costs whether it is for blenders or warehouses

**While**:

1. Ensuring demand statisfaction (Reliquat_demand variable= demand - supply  must be equal to 0)
2. Respecting Blending and Stock capacities
3. Respecting conservation constraints

# Model description - Constraints

## Global constraints

1- Possibility of activation of new blenders (M is a big constant of activation of the constraints) :

sum(b in blender and t in period) [Active_Blender[b][t] + Active_SBlender[b][t] < M*New_Blenders

2- Possibility of activation of new storages:

3- Mutualization of storages between raw materials and fertilizers (Bonus)

4- Possibility to activate storages only if the flow is greater than a minimum limit Stockage_min and the existing capacity

5- Possibility to activate blenders only if the flow is greater than a minimum limit Blender_min and the existing capacity

6- Only pay one time when opening a new blender:

7- Only pay one time when opening a new storage:

8- Unsatisfied demand:

Reliquat_Demand[e,t]+ sum[b in blender, e in Fertilizer ] (Flow_Blender_Province[b,e,t] + Flow_SBlender_Province[b,e,t])= *Demand*[e,t]

# Model description - Constraints

## Constraints of respecting existing capacities

**9- Storage capacity:**

**Sum[s in supply,m in MP]( Flow_Supply_StorageMP[s, st,m,t]) +sum[b in blender, e in Fertilizer] (Flow_Blender_StorageFertlizer[b,se,e,t] + Flow_SBlender_StorageFertlizer[b,se,e,t] ) ≤ *Storage_Capacity*[*st*] + *Capacity_stock_actif* ∗ Active_Downstream_Storage[se,t]+ *Capacity_stock_actif* ∗ Active_Upstream_Storage[se,t]**

**10- Blender capacity:**

**11- smart blender capacity:**

**12- Capacity of the supply:**

# Model description - Constraints

## Constraints of conservation

**13- Product flow through Stock of raw materials**

**Stock_MP[st,m,t0]=sum[ s in supply] (Flow_Supply_StorageMP[s, st,m,t0]) - sum[b in blender] (Flow_StorageMP_Blender[st,b,m,t0] + Flow_StorageMP_SBlender[st,b,m,t0]) ; for all st in storageMP and m in MP**

**Stock_MP[st,m,t]= Stock_MP[st,m,t-1]sum[ s in supply] (Flow_Supply_StorageMP[s, st,m,t]) - sum[b in blender] (Flow_StorageMP_Blender[st,b,m,t] + Flow_StorageMP_SBlender[st,b,m,t]) ; for all st in storageMP and m in MP and t>t0**

**14- Product flows through Stock of fertilizers**

**15- Flows going through Blenders**

# Model Outputs

We will have as an output all decisions variables (flows, stocks, investments variables—blenders and warehouses--, Reliquat_Demand, etc.), they will be represented in form of tables in order to have meaningful data which can be easier to analyze. (Data exported as an excel file)

- Cost to supply
- Flow from supply to blenders
- Flow from supply to storage MP
- Flow storage MP to blender
- Cost into blenders
- Flow from blender to storage
- Flow from blender to province
- Flow from storage to province
- Cost to province
- Stock of raw materials
- Stock of final products
- Reliquat_Demand
- Activation of blenders and smart blenders
- Activation of upstream and downstream warehouses

In addition, teams will have to prepare a 15min presentation and deliver a structured notebook.