Dec 2025

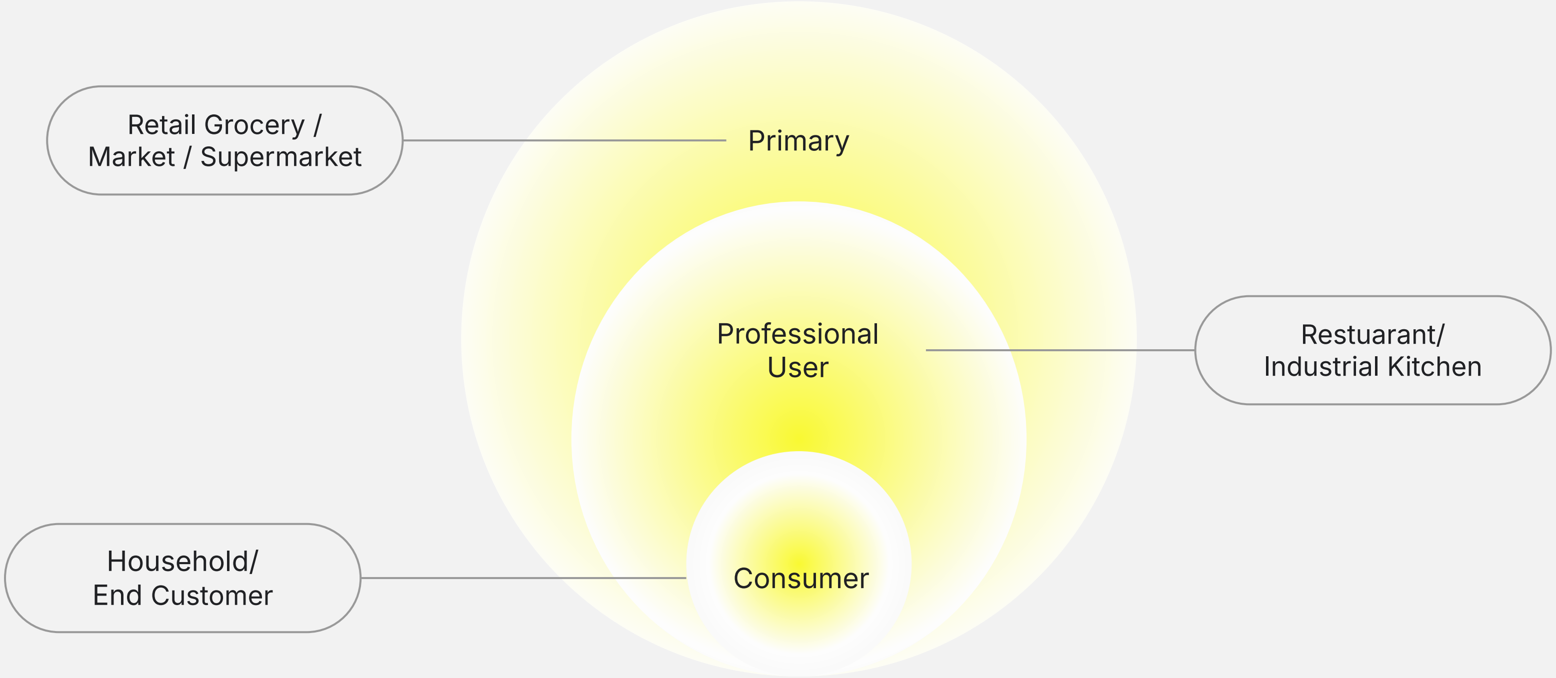
M2M: Meal2Market Autonomy System

Team PY

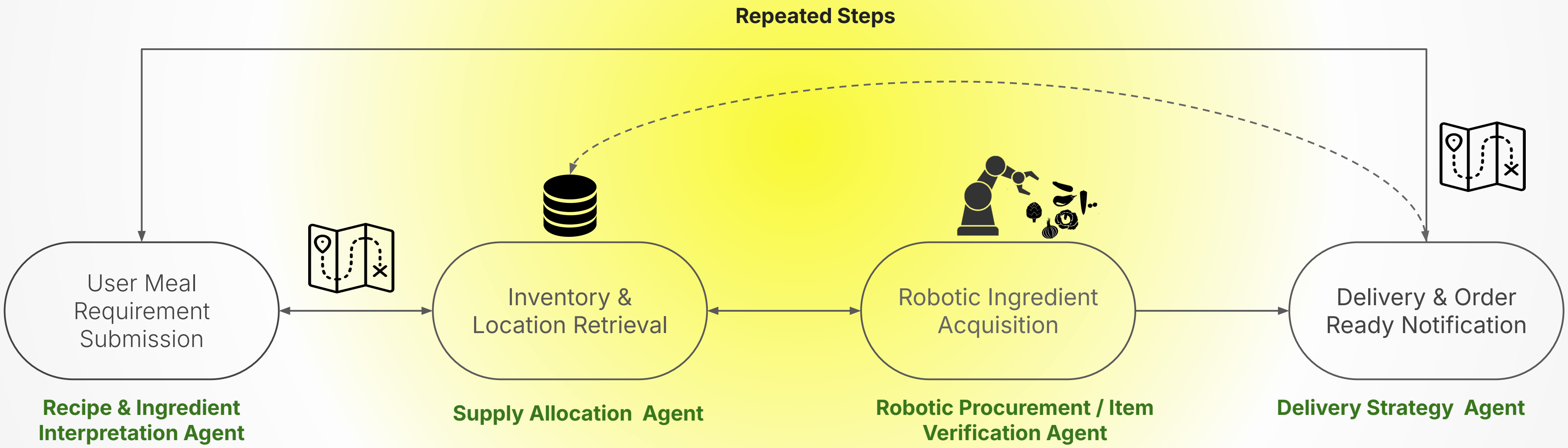
M1461001 Yuyun Wang

M1461002 Priscila Ung

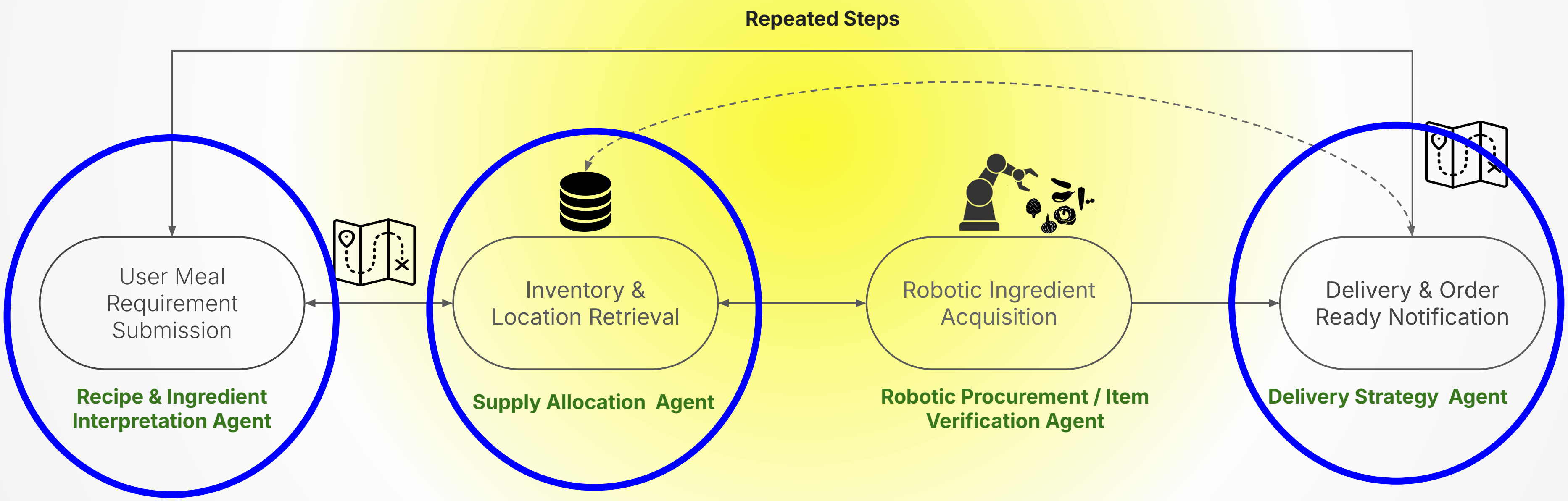
Who are the users?



How we conducted M2M services on multiagent system



How we conducted M2M services on multiagent system



How does M2M make decisions?

When you want to buy something, how do you decide which shop to go to?

Have you ever gone to one grocery store, but still couldn’t find everything you needed?

How does M2M make decisions?

Current State

- *Required ingredients*
- *Available suppliers*

M2M Decision Policy

Single-Agent Utility Function

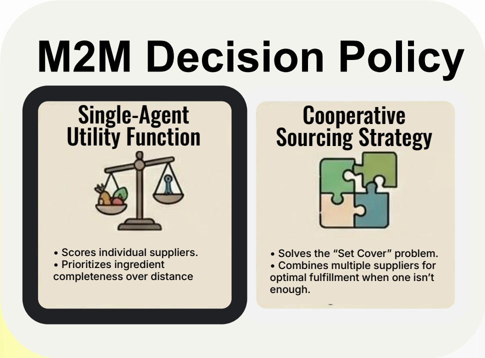
- Scores individual suppliers.
- Prioritizes ingredient completeness over distance

Cooperative Sourcing Strategy

- Solves the “Set Cover” problem.
- Combines multiple suppliers for optimal fulfillment when one isn’t enough.

Final Purchasing Plan

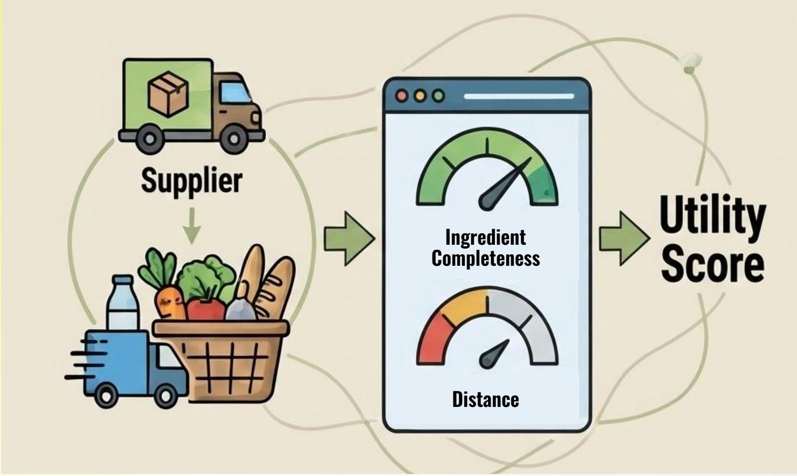
How to know whether a supplier is good or not?



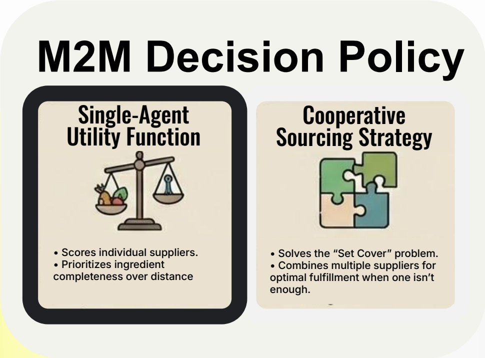
Single-Agent Utility Function

- Who can provide the most ingredients? *(most important)*
- Is the distance reasonable? *(secondary)*

$$U(s_j) = w_1 \cdot \text{Coverage}(s_j) - w_2 \cdot \text{Distance}(s_j)$$



How to know whether a supplier is good or not?



Single-Agent Utility Function

- Who can provide the most ingredients? *(most important)*
- Is the distance reasonable? *(secondary)*

$$U(s_j) = w_1 \cdot \text{Coverage}(s_j) - w_2 \cdot \text{Distance}(s_j)$$

$$\text{Coverage}(s_j) = \sum_{i \in I_{req}} \text{Inv}(s_j, i)$$

$$\text{Inv}(s_j, i) = \begin{cases} 1, & \text{if supplier } s_j \text{ has ingredient } i \\ 0, & \text{otherwise} \end{cases}$$

I_{req}: Set of required ingredients derived from recipe analysis

S: Set of all available suppliers

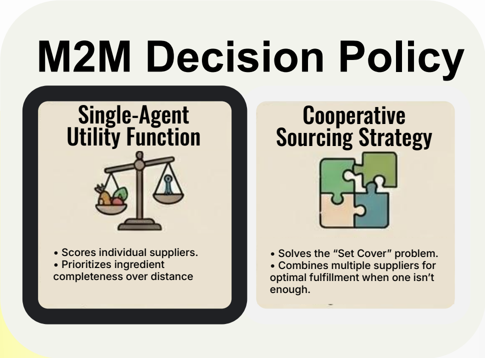
$s_j \in S$: A supplier

L_{s_j}: Geospatial coordinates of supplier s_j

L_{user}: Geospatial coordinates of the user

$w_1 \gg w_2$: Ingredient coverage is prioritized over distance

How to know whether a supplier is good or not?



Single-Agent Utility Function

- Who can provide the most ingredients? *(most important)*
- Is the distance reasonable? *(secondary)*

$$U(s_j) = w_1 \cdot \text{Coverage}(s_j) - w_2 \cdot \text{Distance}(s_j)$$

$$\text{Distance}(s_j) = \|L_{s_j} - L_{user}\|$$

I_{req}: Set of required ingredients derived from recipe analysis

S: Set of all available suppliers

$s_j \in S$: A supplier

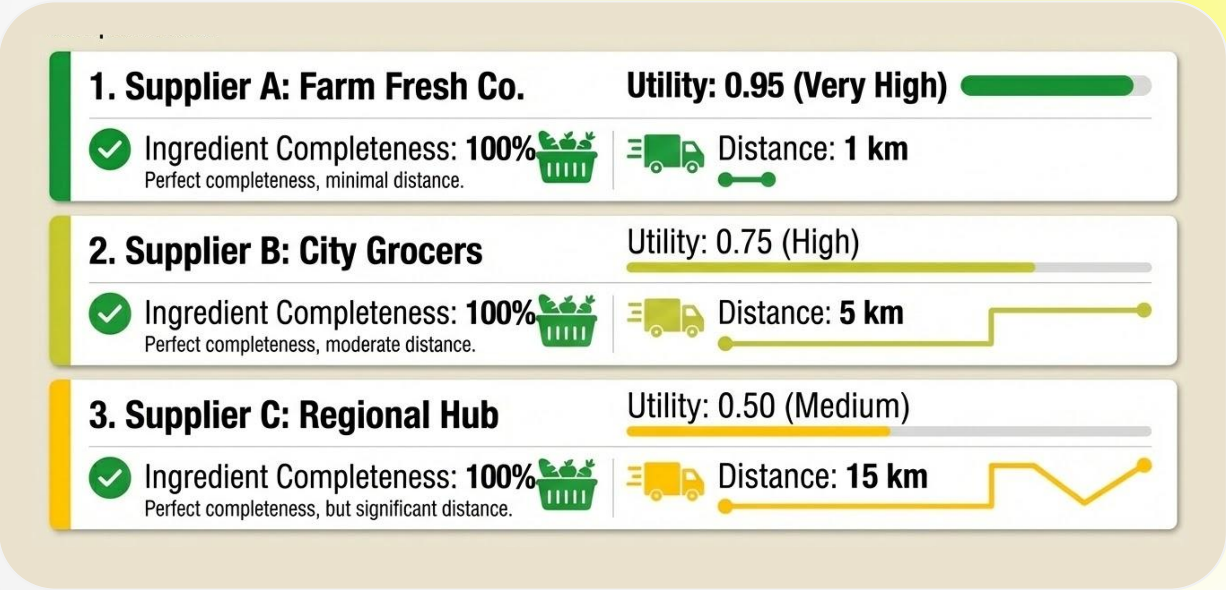
L_{s_j}: Geospatial coordinates of supplier s_j

L_{user}: Geospatial coordinates of the user

$w_1 \gg w_2$: Ingredient coverage is prioritized over distance

How to know whether a supplier is good or not?

Single-Agent Utility Function



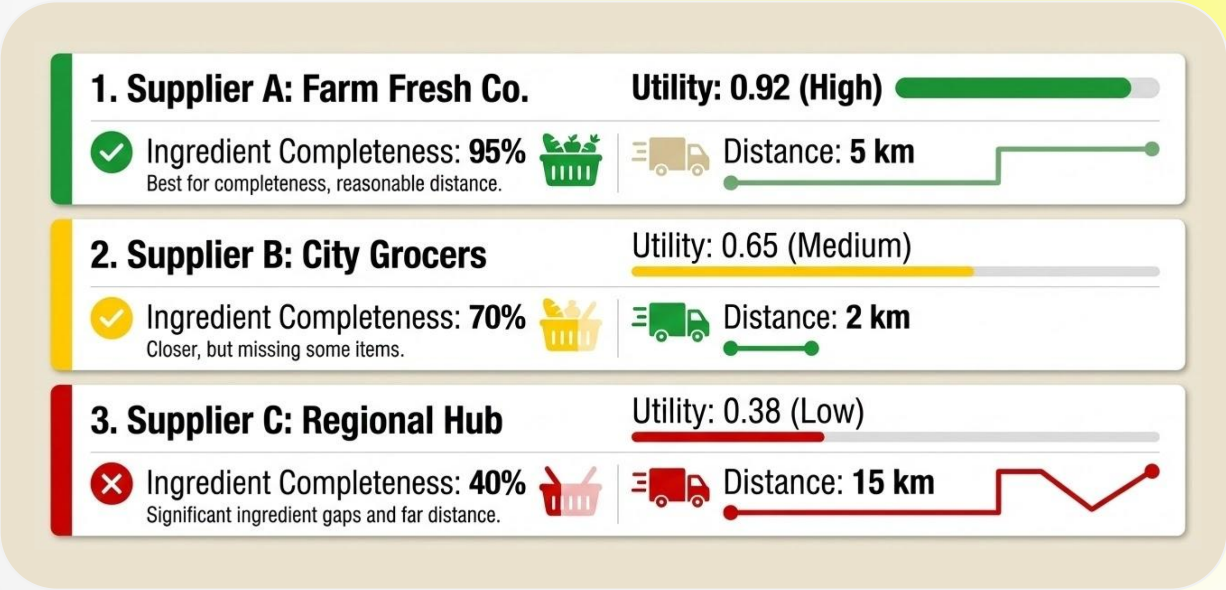
Decision Policy Checks

$$\mathrm{Inv}(s^*_{partial}) \supseteq I_{req}$$

YES → stop, output plan
NO → some ingredients are missing

How to know whether a supplier is good or not?

Single-Agent Utility Function



Decision Policy Checks

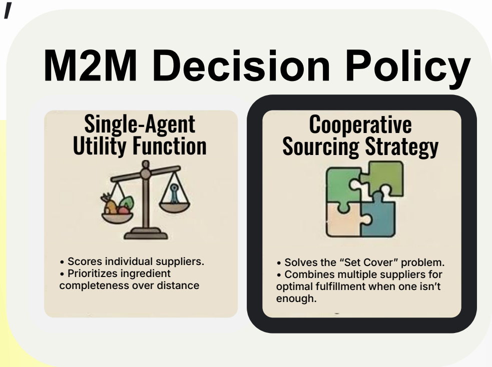
$$\mathrm{Inv}(s^*_{partial}) \supseteq I_{req}$$

YES → stop, output plan

NO → some ingredients are missing

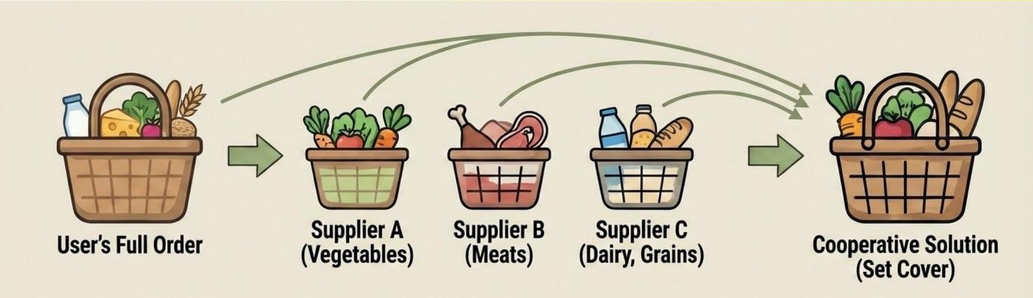
How to solve this problem?

When NO single supplier has all ingredients, how to select multiple suppliers?



Cooperative Sourcing Strategy

- The nearest supplier → may have incomplete stock 😟
- The supplier with the most complete stock → may be too far away 😰
- Cheap but lacking key ingredients → Order failed 😭



Select a combination of suppliers that:

- completes all missing items
- with the lowest delivery cost

$$\text{Find } S^* \subseteq S \text{ such that all } I_{req} \text{ are covered}$$

When NO single supplier has all ingredients, how to select multiple suppliers?

M2M Decision Policy

Single-Agent Utility Function

- Scores individual suppliers.
- Prioritizes ingredient completeness over distance

Cooperative Sourcing Strategy

- Solves the “Set Cover” problem.
- Combines multiple suppliers for optimal fulfillment when one isn’t enough.

Cooperative Sourcing Strategy

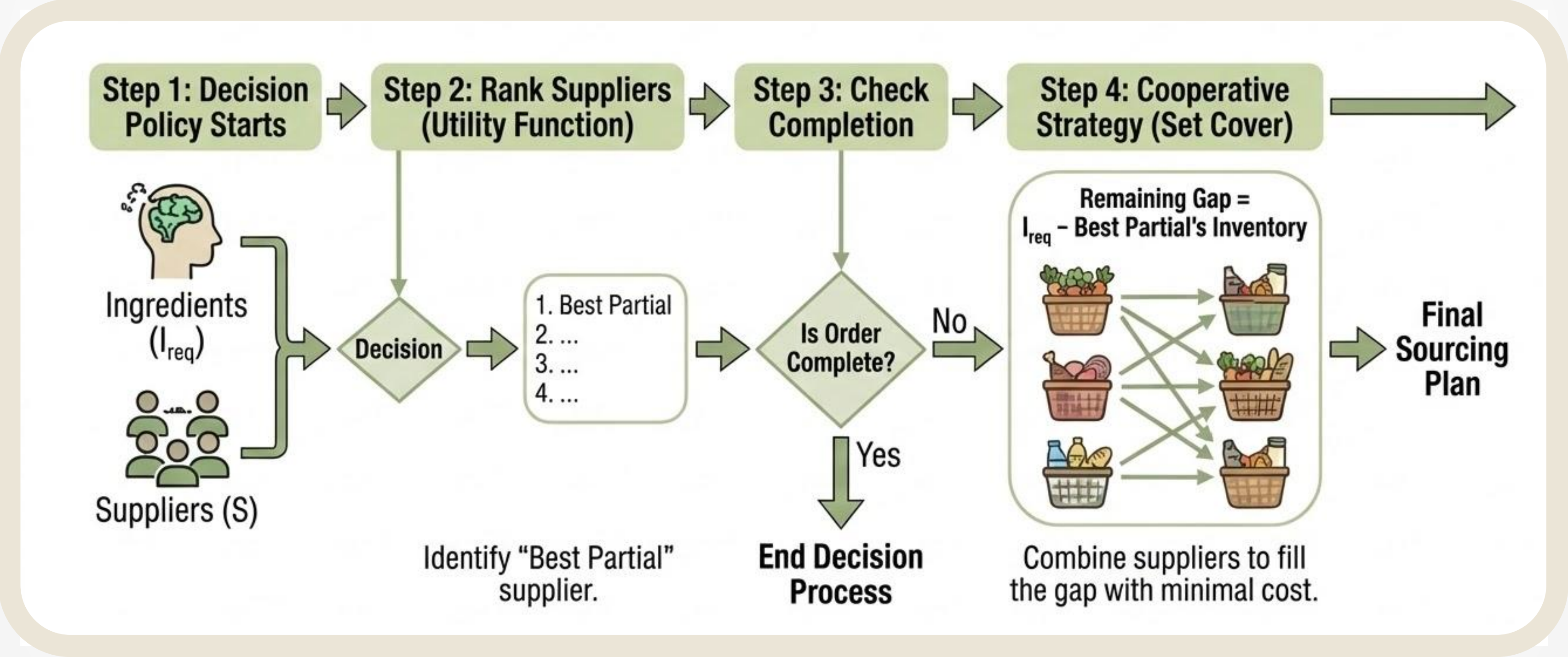
Coverage Constraint:

Every ingredient is supplied by at least one supplier.

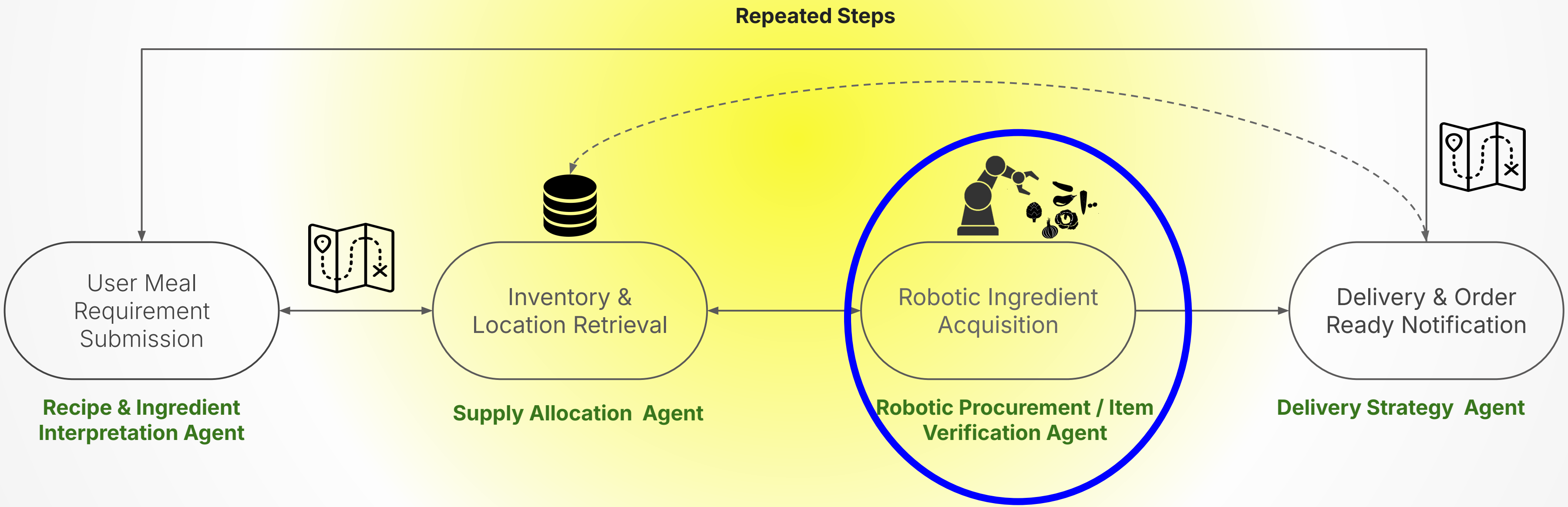
$$\forall i \in I_{req}, \quad \sum_{j \in S} x_{ij} \geq 1$$

x_{ij}: whether supplier j supplies ingredient i (1 = yes, 0 = no)

Integrated Process Flow



How we conducted M2M services on multiagent system



What Does the Robot Need?

Goal space

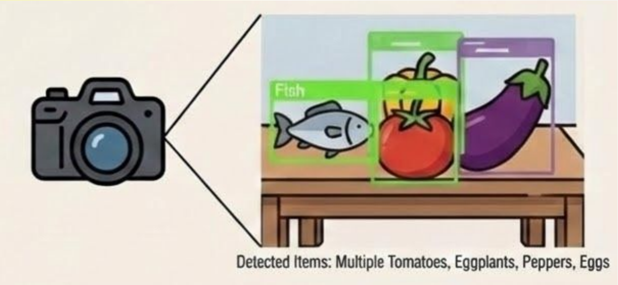
$$g_t = \{(o_k, n_k)\}_{k=1}^{K}$$

- g_t: Goal set at time t
- o_k: Semantic category of the k-th goal item (e.g., tomato, egg, milk)
- n_k: Required quantity of that item (e.g., 2 units, 1 box)
- K: Number of goal item categories

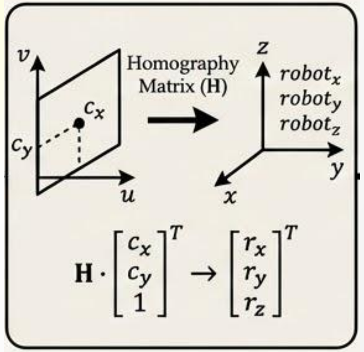
What Does the Robot See?

Observation space

$$O_t = \{o_{t,i}\}_{i=1}^{N} = \{(c_i, b_i, p_i, g_i)\}_{i=1}^{N}$$

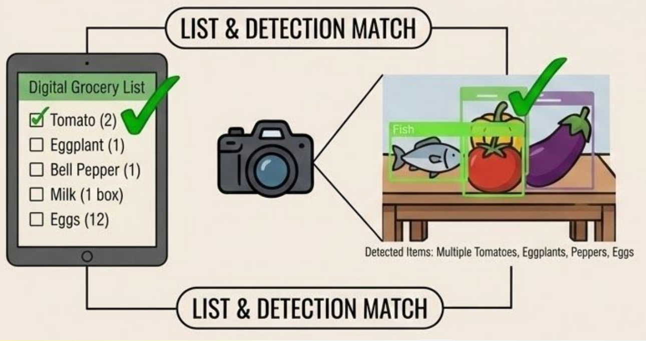


- c_i: Detected category
- b_i: bounding box
- p_i: 3D position
- g_i: graspability (Probability or score)



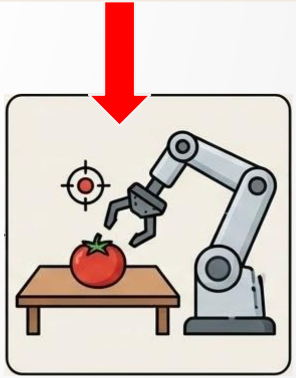
Can the Robot Act on This Item?

Verification

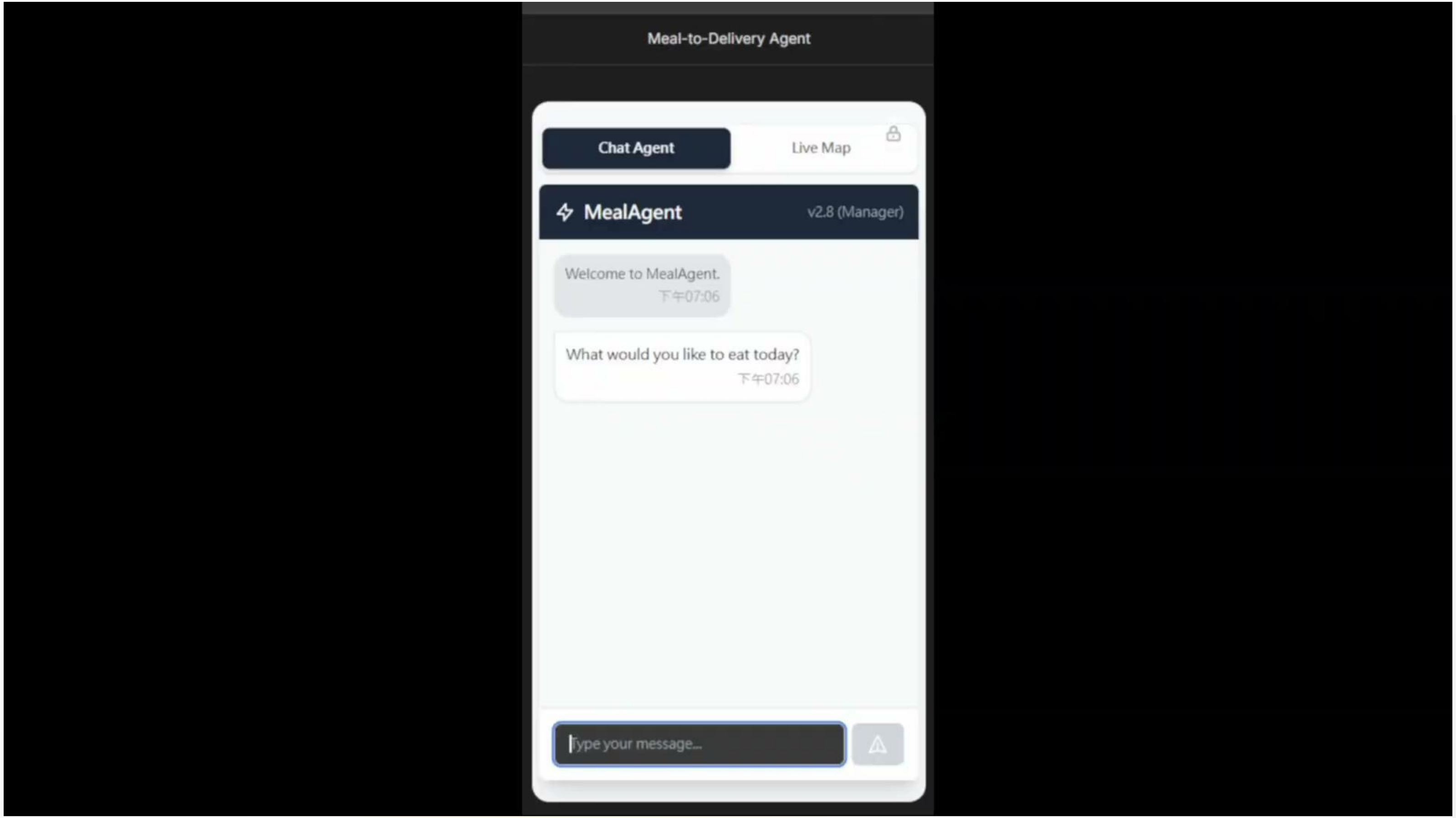


$$V_t(o_k) = \mathbf{1}[\exists_{i\in\{1,\cdots,N\}}\, s.t.\, c_i = o_k \wedge g_i \geq \tau_g]$$

- $V_t(o_k)$: At time t, whether the target category o_k is 'verified as actionable'
- $\mathbf{1}[\cdot]$: Indicator function (condition true → 1, otherwise → 0)
- $\exists_{i\in}\{1,\cdots,N\}$: There exists at least one perceived object instance
- $c_i = o_k$: The perceived object category matches the target category
- $g_i \geq \tau_g$: The object's graspability is greater than or equal to the threshold τ_g



Demo


Meal-to-Delivery Agent
Chat Agent
Live Map
MealAgent
v2.8 (Manager)
Welcome to MealAgent.
下午07:06
What would you like to eat today?
下午07:06
Type your message...

M2M app