CollaboRobot - Collaborative Nonverbal Negotiation of a Recipe with a Human and a Robot

CollaboRobot - Niko Pallas, Mark Ringer, Khaireddine Essid

Today's agenda

Motivation & Concept

User Interaction Design

Implementation

Reflection

Demonstration

Questions

Motivation & Concept

Motivation & Concept

A **user and robot** arm **collaboratively** decide what to cook through **non-verbal, physical negotiation**. Rather than the user specifying a complete recipe upfront, the system creates a **back-and-forth dialogue**.

Negotiate

- **Robot proposes ingredients** based on current state & recipe knowledge
- **User accepts/rejects** by physically placing or removing ingredients
- **System adapts** its suggestions **dynamically** until a meal emerges

Nonverbally

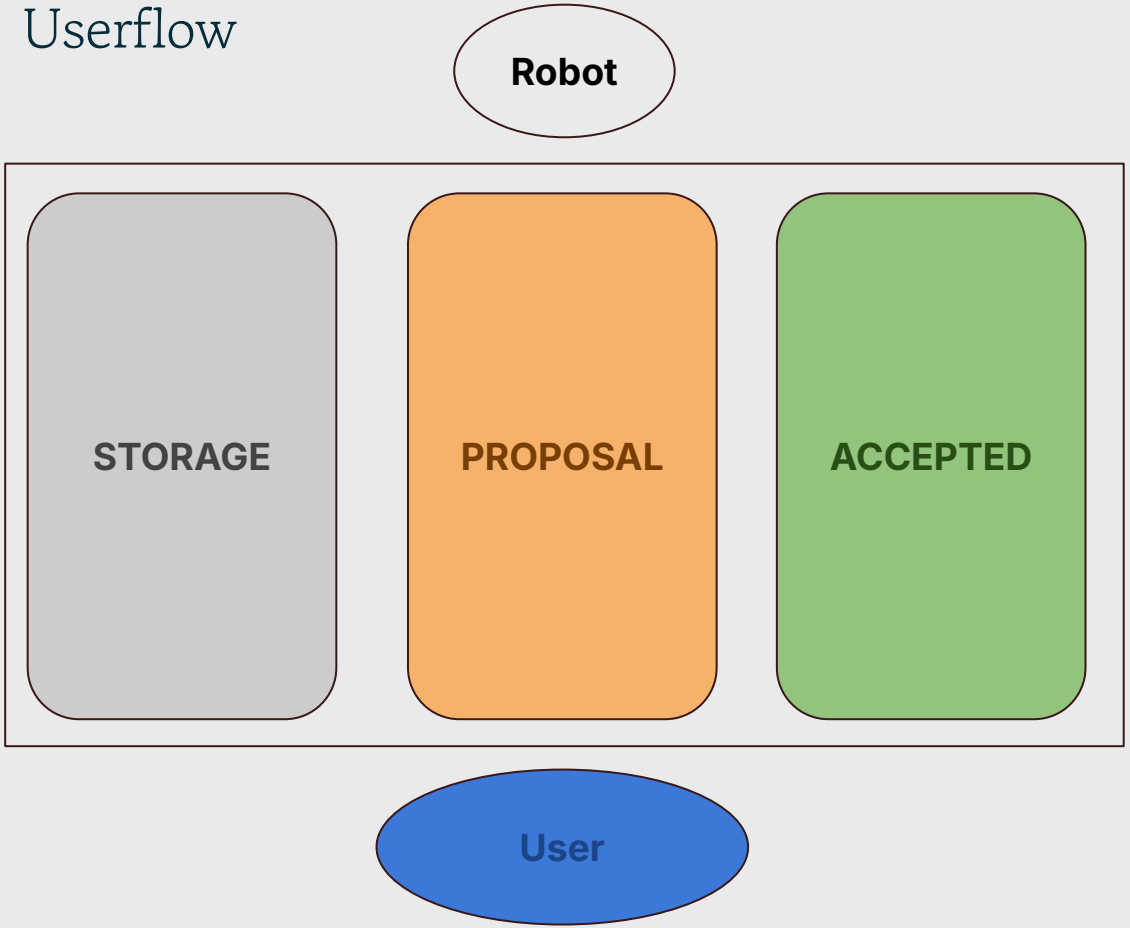
- No voice commands needed - purely object-based
- **ArUco** markers **identify** ingredients
- **Hot-zones** define **intent**
- **Physical placement** resembles the **communication**

Bidirectional influence

- **Neither party fully controls the outcome**
- **Robot** doesn't just assist, it **actively proposes** and shapes the meal
- **User** can **counter-propose**, **reject** or **redirect** at any time

User Interaction Design

Userflow



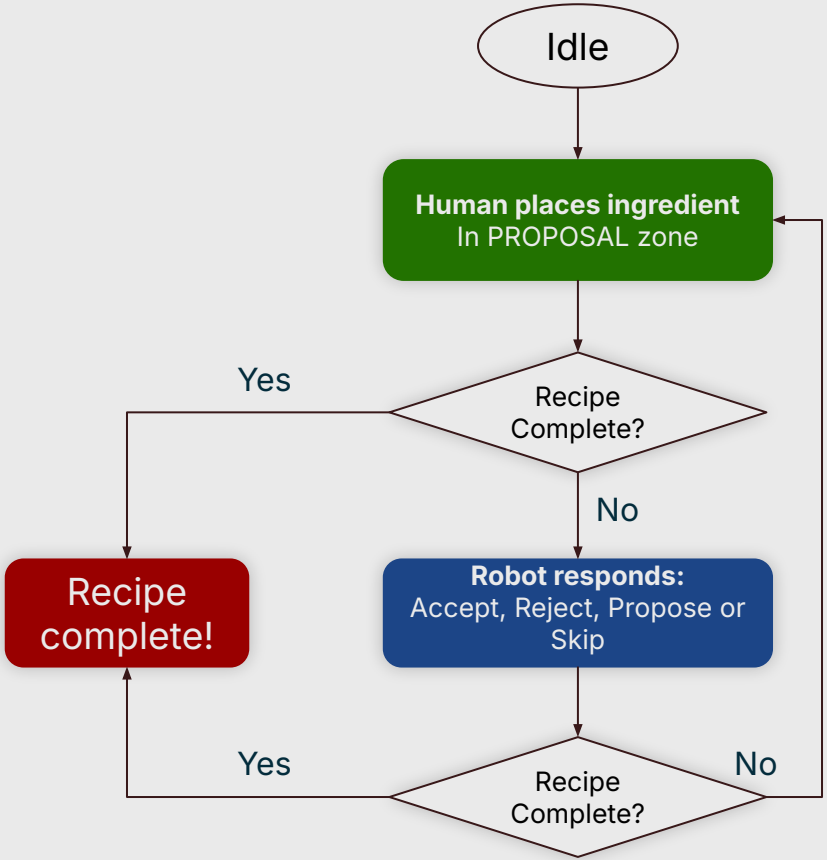
Example:

- → User: *tomato* in **PROPOSAL**
  {A: [], P: [“tomato”]}
- → Robot: *tomato* in **ACCEPTED**
  {A: [“tomato”], P: []}
- → Robot: *pasta* in **PROPOSAL**
  {A: [“tomato”], P: [“pasta”]}
- → User: *pasta* in **ACCEPTED**
  {A: [“tomato”, “pasta”], P: []}
- → Robot: *garlic* in **PROPOSAL**
  {A: [“tomato”, “pasta”], P: [“garlic”]}
- → User: *garlic* in **STORAGE**
  {A: [“tomato”, “pasta”], P:[]}
- → Robot: *basil* in **PROPOSAL**
  {A: [“tomato”, “pasta”], P:[“basil”]}
- → User: *basil* in **ACCEPTED**
  {A: [“tomato”, “pasta”, “basil”], P:[]}
- → User: *DONE* in **PROPOSAL**
  {A: [“tomato”, “pasta”, “basil”], P:[“Skip”]}

Result: Tomato Basil Pasta 🍝

Userflow

Userflow



Initiation
- → Human starts by placing first ingredient
- → Signals intent to begin cooking
- → Robot activates negotiation mode

Negotiation Loop
- → PROPOSAL zone = shared decision space

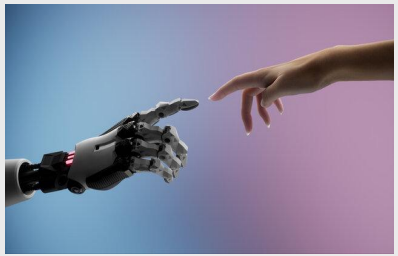
Turn-Taking
- → Flexible: either party can propose, accept and reject
- → Robot uses recipe DB for suggestions
- → Robot rejects ingredients that don't fit

Termination
- → Robot signals with a gesture when the recipe is complete

Implementation

System Architecture Overview

Vision **Thinking and Control** **Motion**



From real World

To real World

System Architecture Overview

Vision Thinking and Control Motion

From real World

To real World

State Detection

Detect ArUco Tags

- Detect Tags with OpenCV
- Return IDs with positions as dict

Caching positions

- Marker Positions saved in memory
- If marker can’t be detected, old position is used

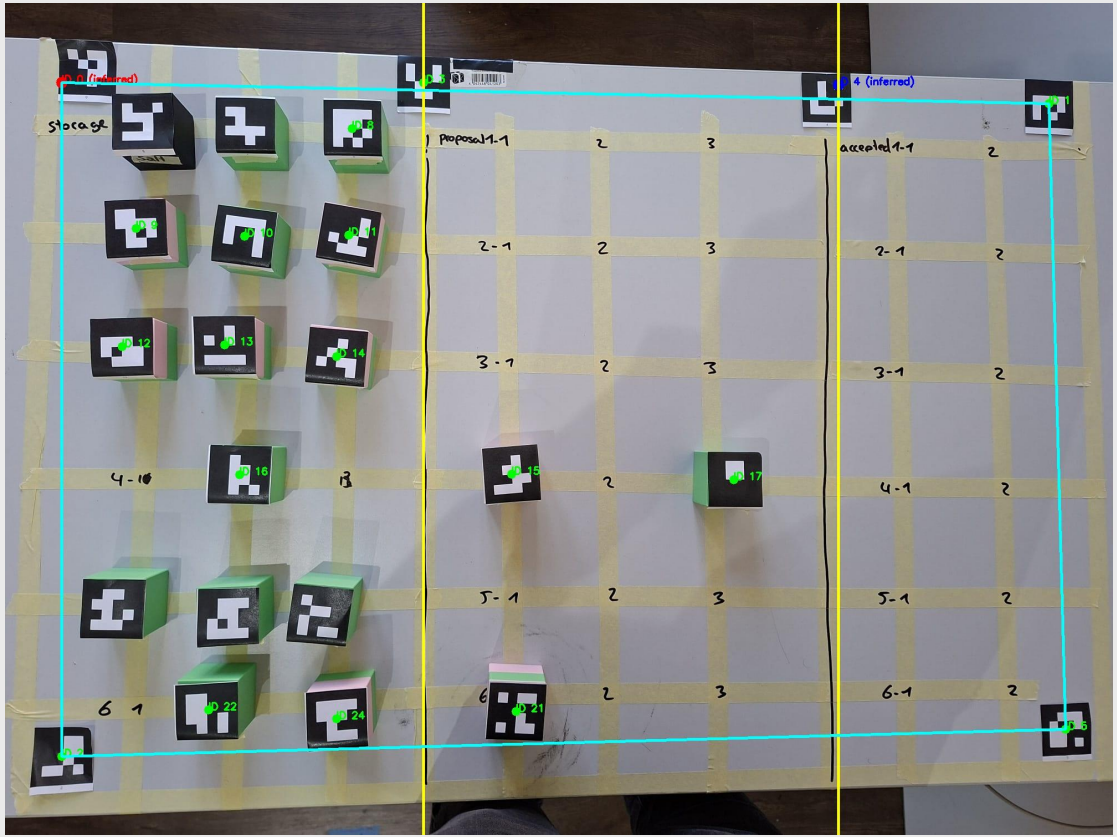
Assigning to Zones

- Markers 0-5 are used to defined the zones
- Markers 6-24 are each assigned to storage, proposed or accepted

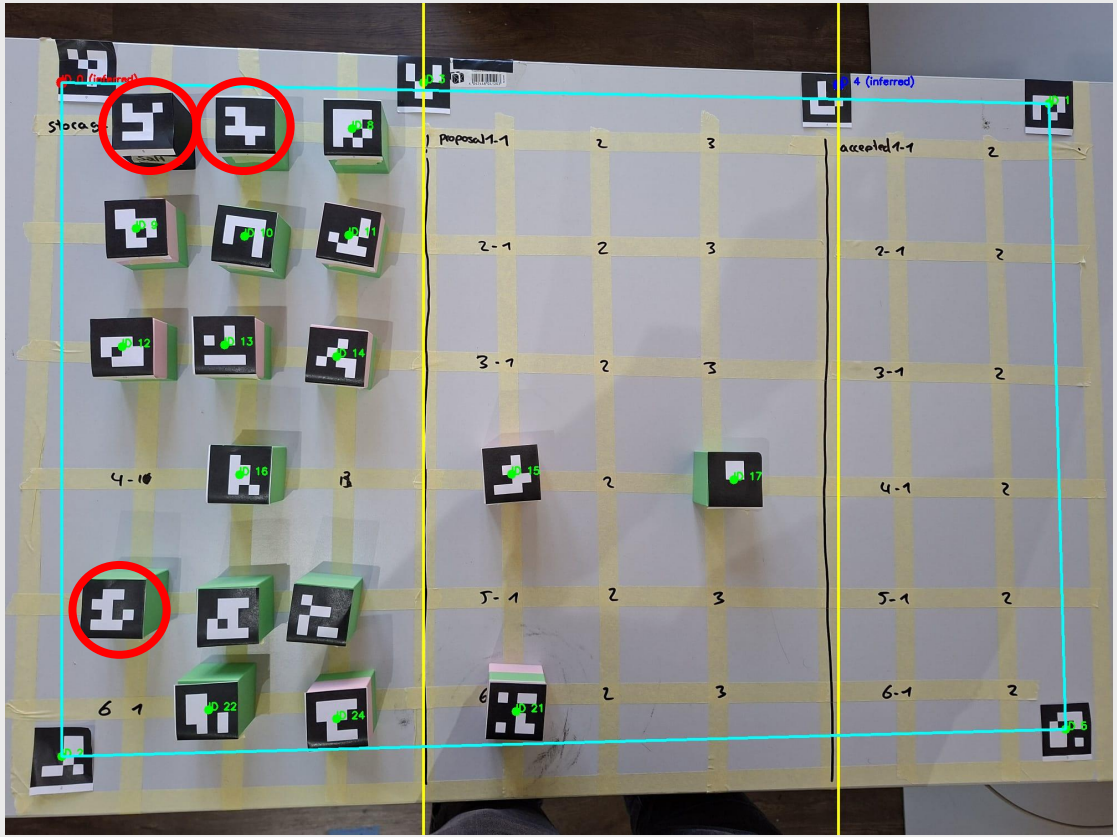
Constraints

- Due to Resolution issues, markers are not reliably detected
- Mocking positions via Command Line Interface
- Improved Visual detection could be integrated seamlessly

State Detection



State Detection



System Architecture Overview

Vision

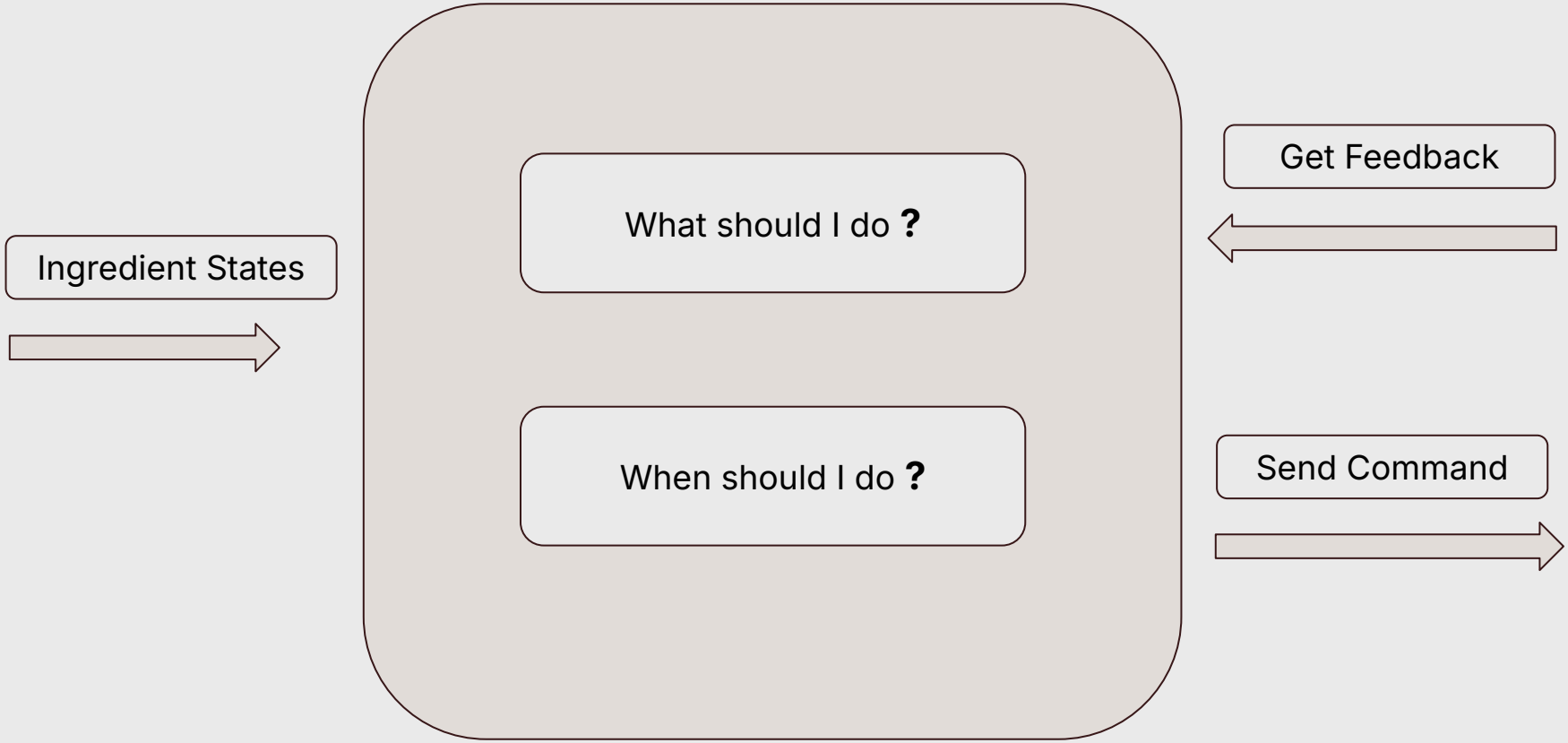
Thinking and Control

Motion

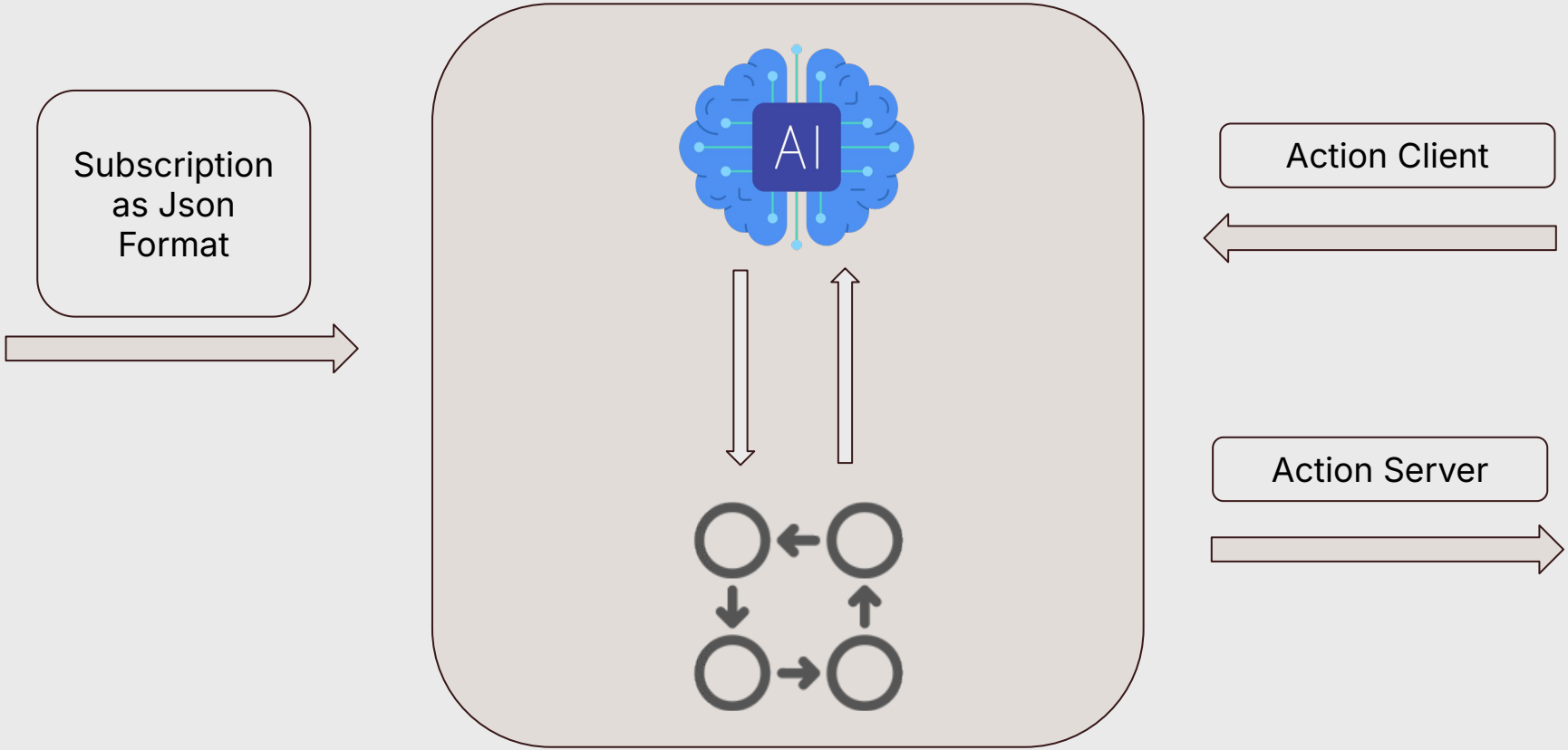
From real World

To real World

System Architecture Thinking and Control

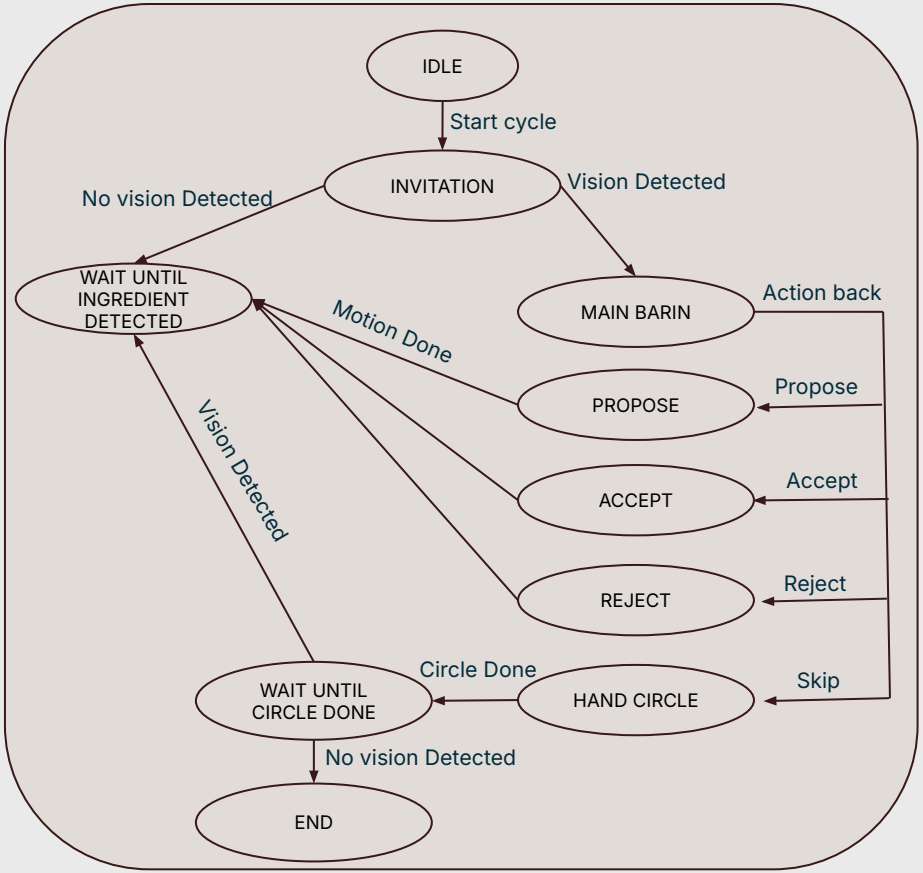


System Architecture Thinking and Control



State Machine & Technical Implementation

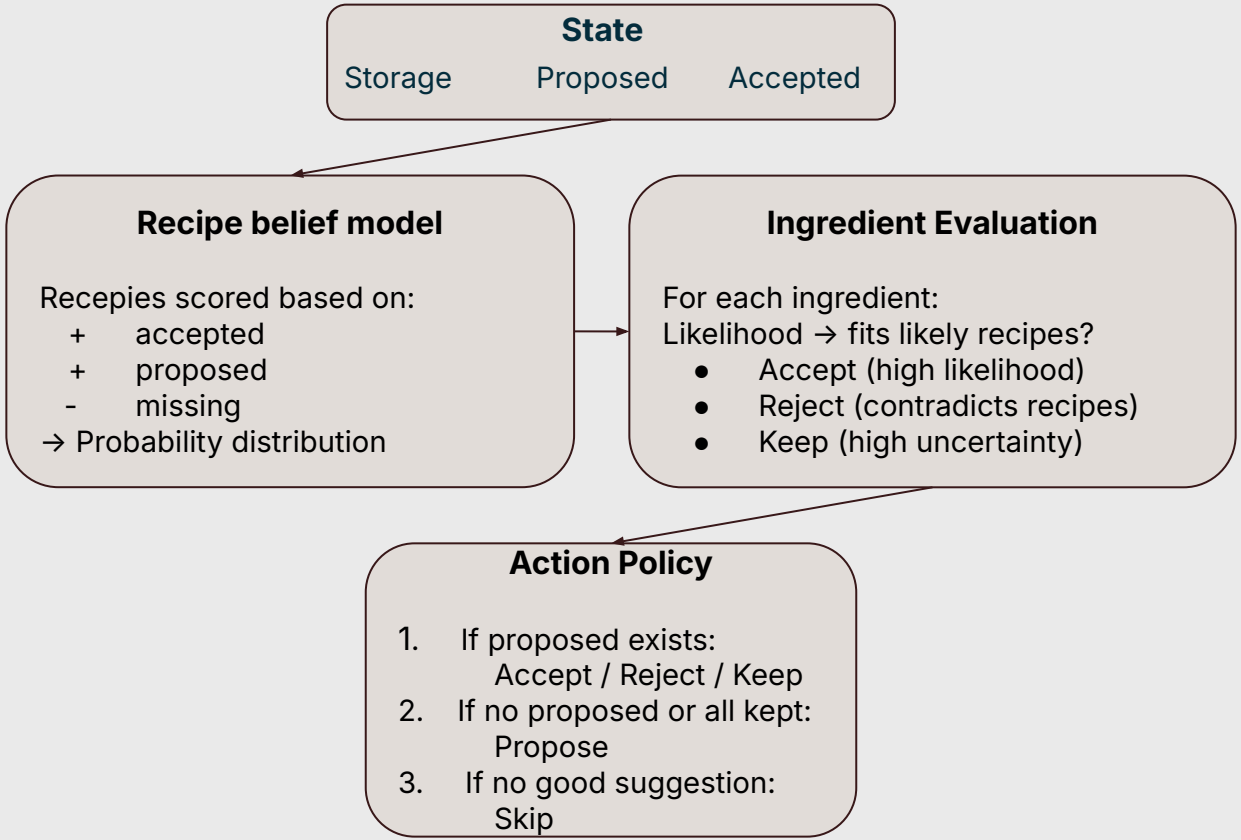
Subscription as Json Format



Get Feedback

Send Command

Recipe Model

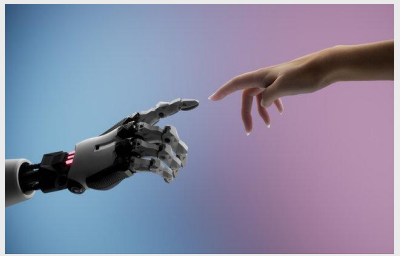


System Architecture Overview

Vision

Thinking and Control

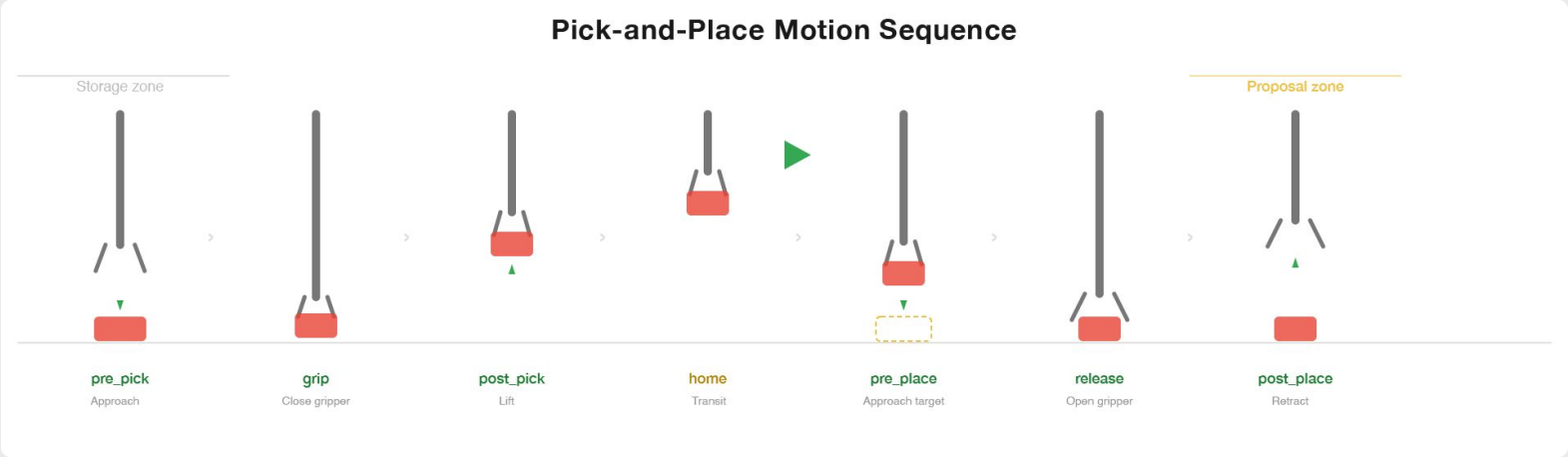
Motion



From real World

To real World

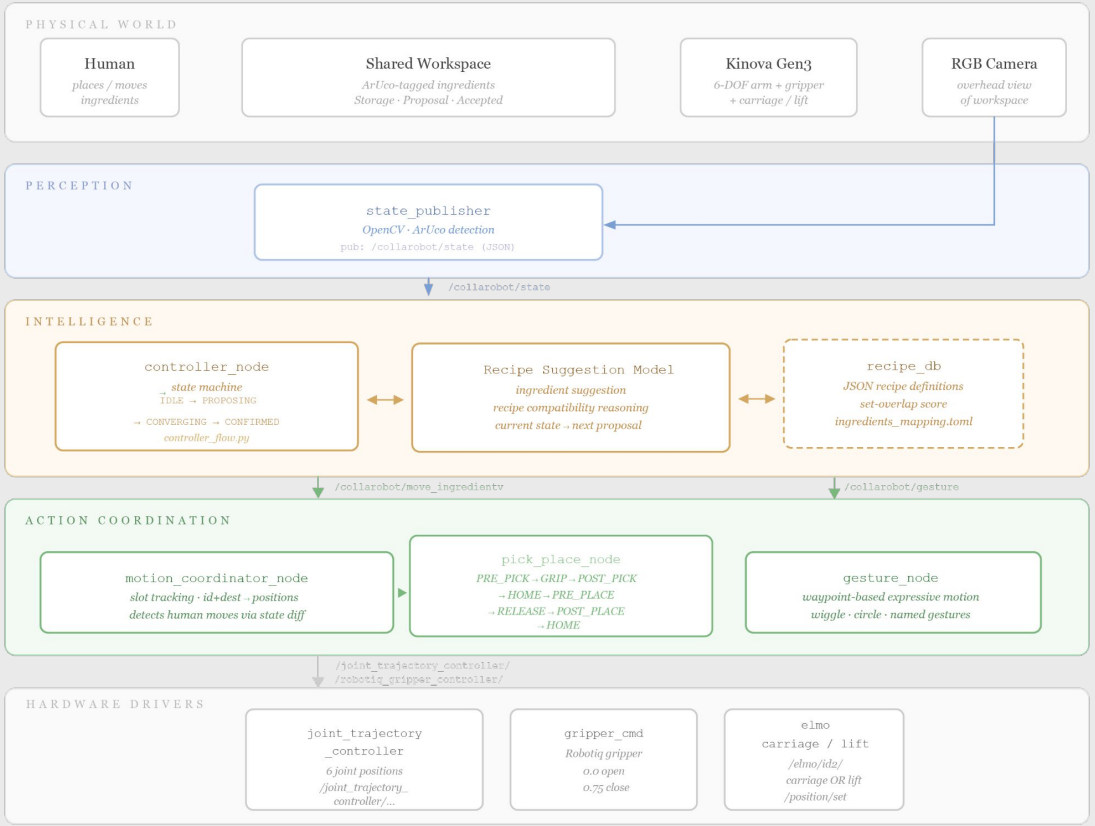
Motion - Architecture



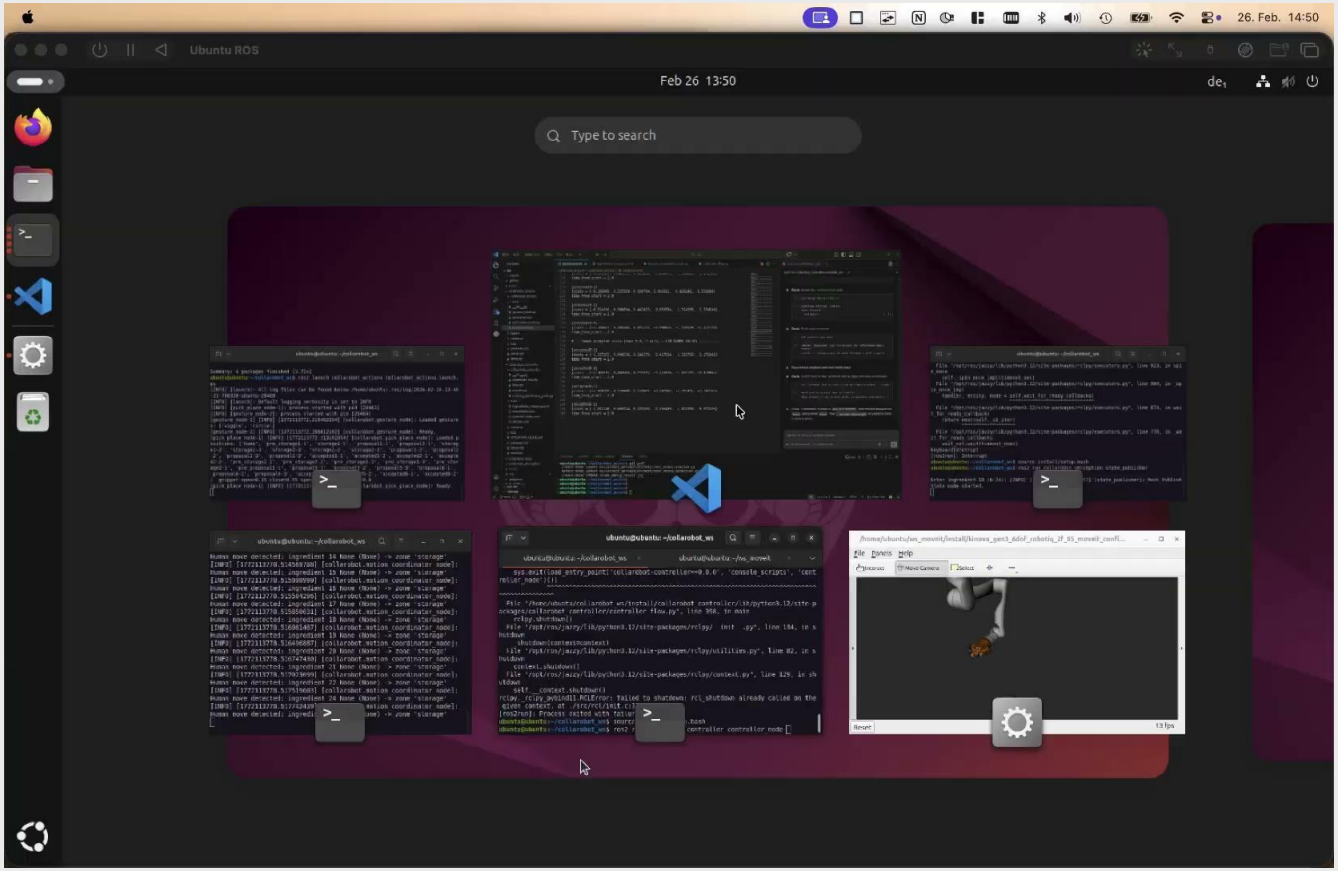
positions.toml

```toml
gripper_open = 0.35
gripper_close = 0.55
gripper_speed = 1.0
gripper_max_effort = 50.0
lift_position = 0.48
carriage_position = 12.550000190734863

[home]
joints = [-0.344518, 1.396093, 1.669235, 3.110204, -1.427716, -1.557667]
time_from_start = 6.0
lift = 0.48
carriage = 12.550000190734863

[pre_storage1-1]
joints = [0.542313, 1.107508, 0.910720, 2.273508, -1.003185, -1.769362]
time_from_start = 4.0

[storage1-1]
joints = [0.641163, 1.034272, 0.789634, 2.319210, -0.978098, -1.827030]
time_from_start = 4.0

[proposal1-1]
joints = [-0.197909, 0.856364, 0.942797, 2.993955, -0.929272, -1.558572]
time_from_start = 4.0

[proposal2-1]
joints = [-0.198009, 0.727259, 0.999784, 2.966011, -0.820248, -1.558980]
time_from_start = 4.0
```

Components & System Requirements



Bringing it all together



Learnings

Our Learnings & Future Work

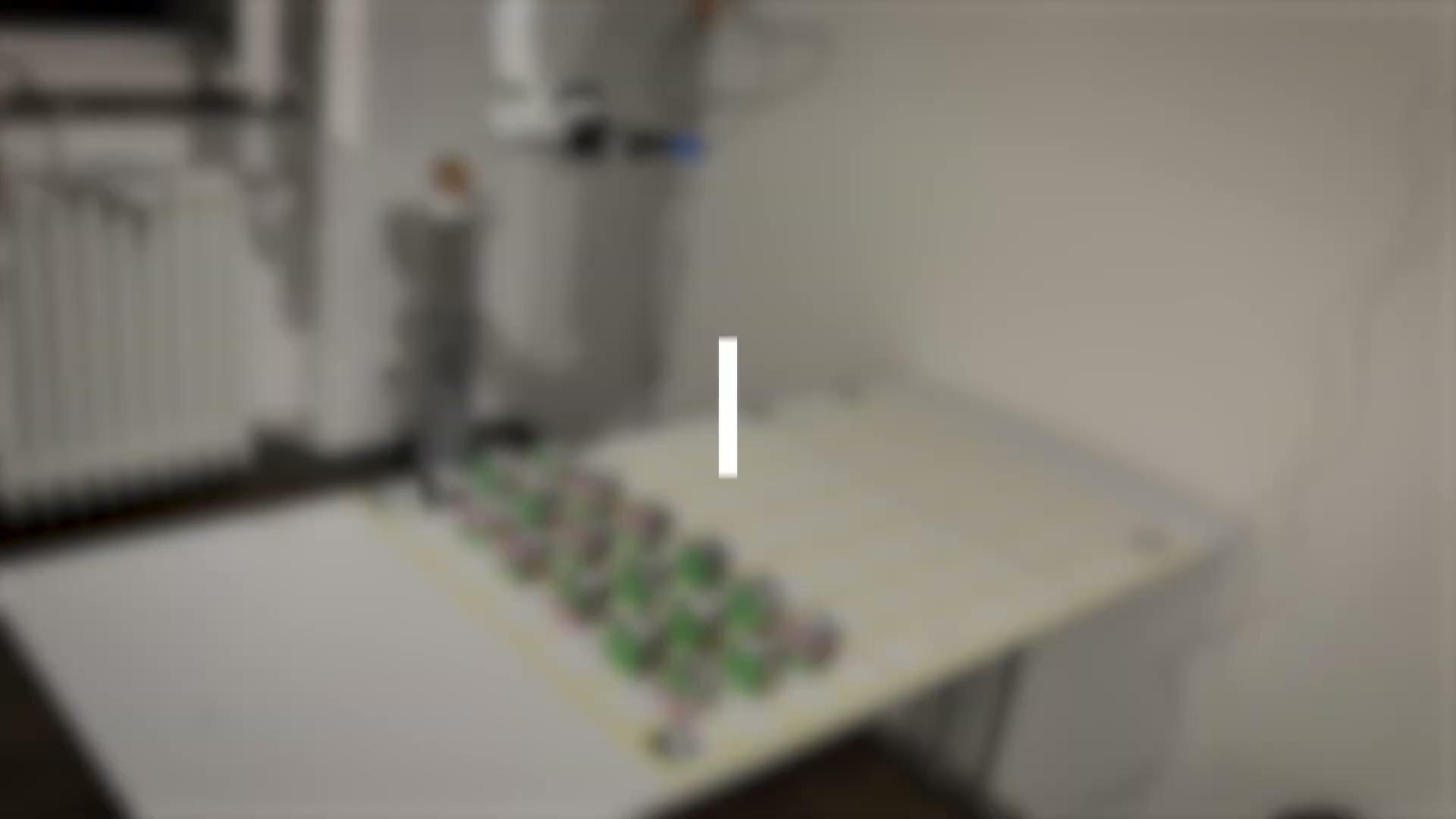
Future Work

- Improved Ingredient detection and classification (YOLO or similar)
- RAG based recipe evaluation / LLM inclusion
- GUI/ App → To preferences (i.e. calories) & display recipe
- Human detection to operate dynamically
- Different personalities of robot encoded via movement qualities

Our Learnings

- Physical constraints
- Robot is more restricted in movement then thought
- Simplest possible solution (fixed positions)
- Grabbing cubes not as easy as it seams

Demonstration



Questions