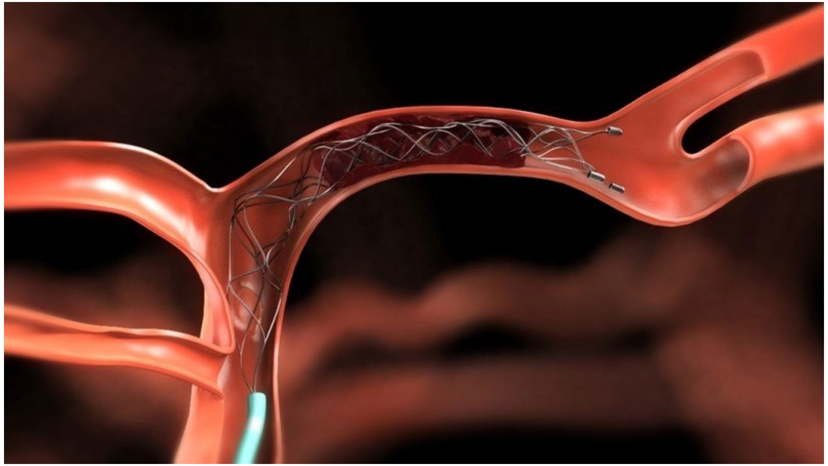
Building Foundation Model-powered Robots with ROS: A Survey

Yifei Cheng
Remedy Robotics
ROSCon 2024

Remedy Robotics

We are developing AI-enhanced robotic systems that perform remote endovascular intervention and also improve its safety and precision.

Our mission is to bring cutting edge endovascular intervention to everyone around the world, as quickly and as safely as possible.



About Me

- Robotics engineer for the last 10 years in logistics, autonomous driving, and medical industries
- Been experimenting with a self-built Annin Robotics AR4 on the side

UNIVERSITY OF WATERLOO

KINDRED
PART OF ocado GROUP

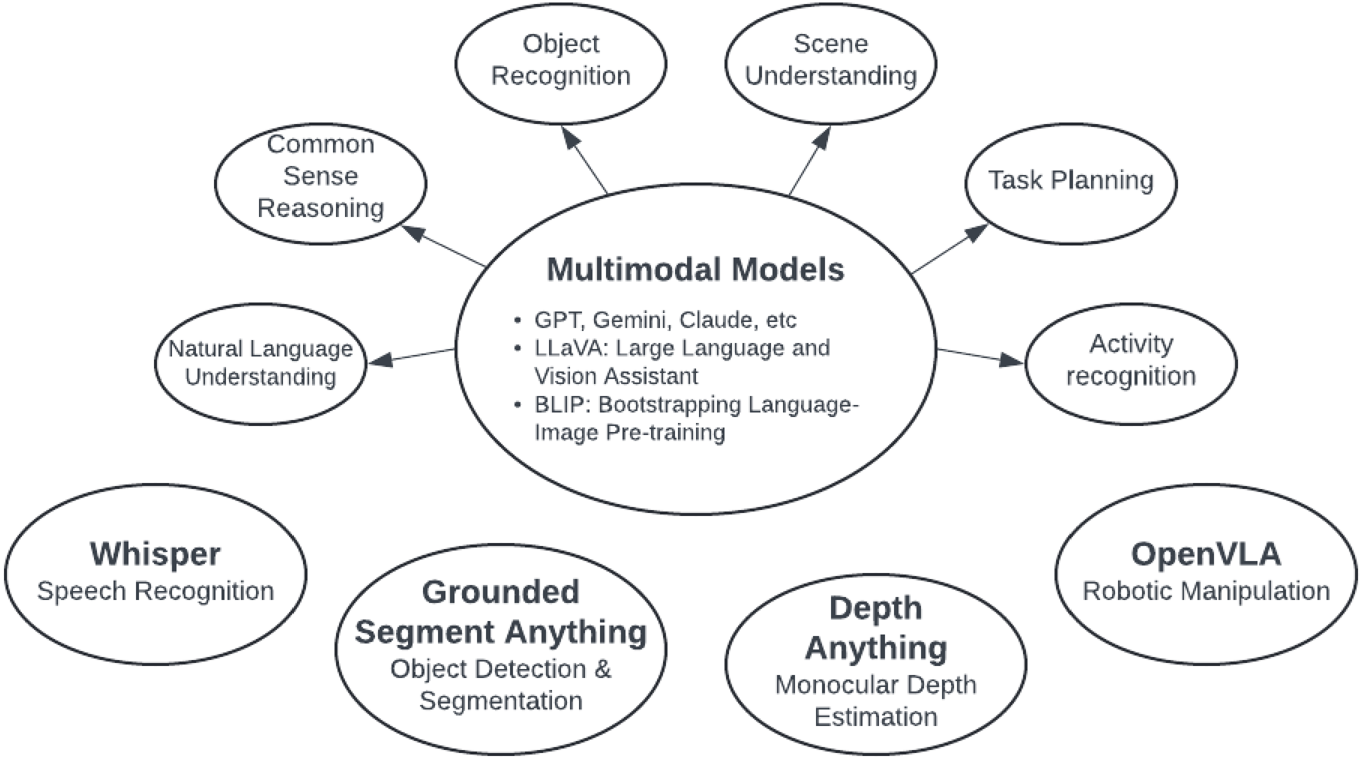
nuro

Remedy Robotics

Disclaimer

- Opinions are my own
- Many new developments in the space, so I apologize for any omissions
- I Intermix these terms: Large Language Models (LLM), Multi-modal Large Language Models (MLLM), and Multi-modal Models

Models and Capabilities For Robotics



Multimodal Models

- GPT, Gemini, Claude, etc
- LLaVA: Large Language and Vision Assistant
- BLIP: Bootstrapping Language-Image Pre-training

Foundation Model Landscape in ROS

Frameworks

- RAI
- ROSA
- ROS-LLM (Huawei)
- ROS-LLM (Auromix)
- ROSGPT
- ROSGPT Vision
- TCC-IRoNL

Demo Projects

- Tabletop Handybot
- Utilizing LLMs as a Task Planning Agent for Robotics

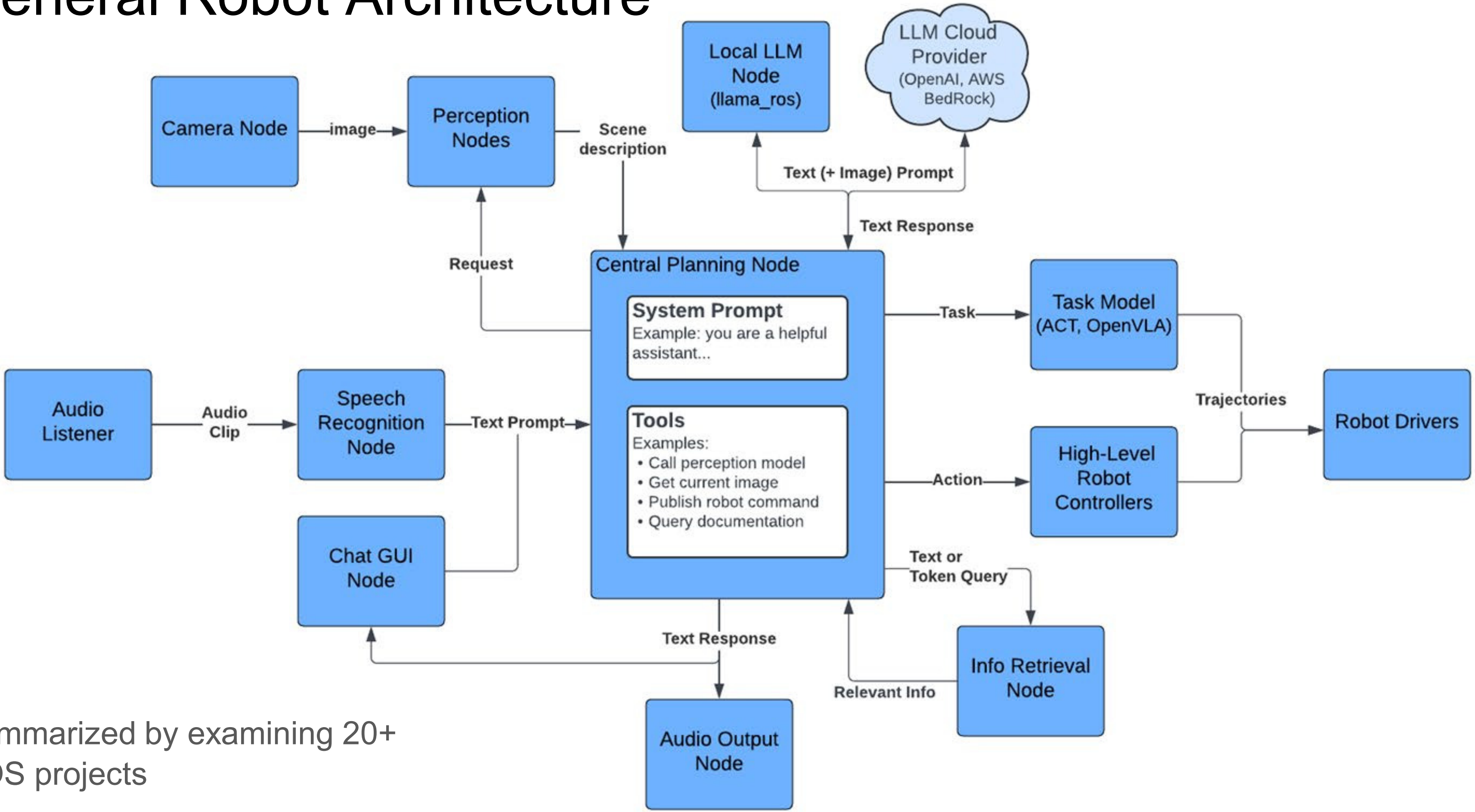
Model Wrappers

- **Llama_ros**: wraps llama.cpp
- **Ros_sam** / **Ros2_sam**: wraps Segment Anything
- **Grounding_sam_ros**: wraps Grounded Segment Anything
- **Whisper_ros** / **Ros2_whisper**: wraps whisper.cpp

Developer Tools

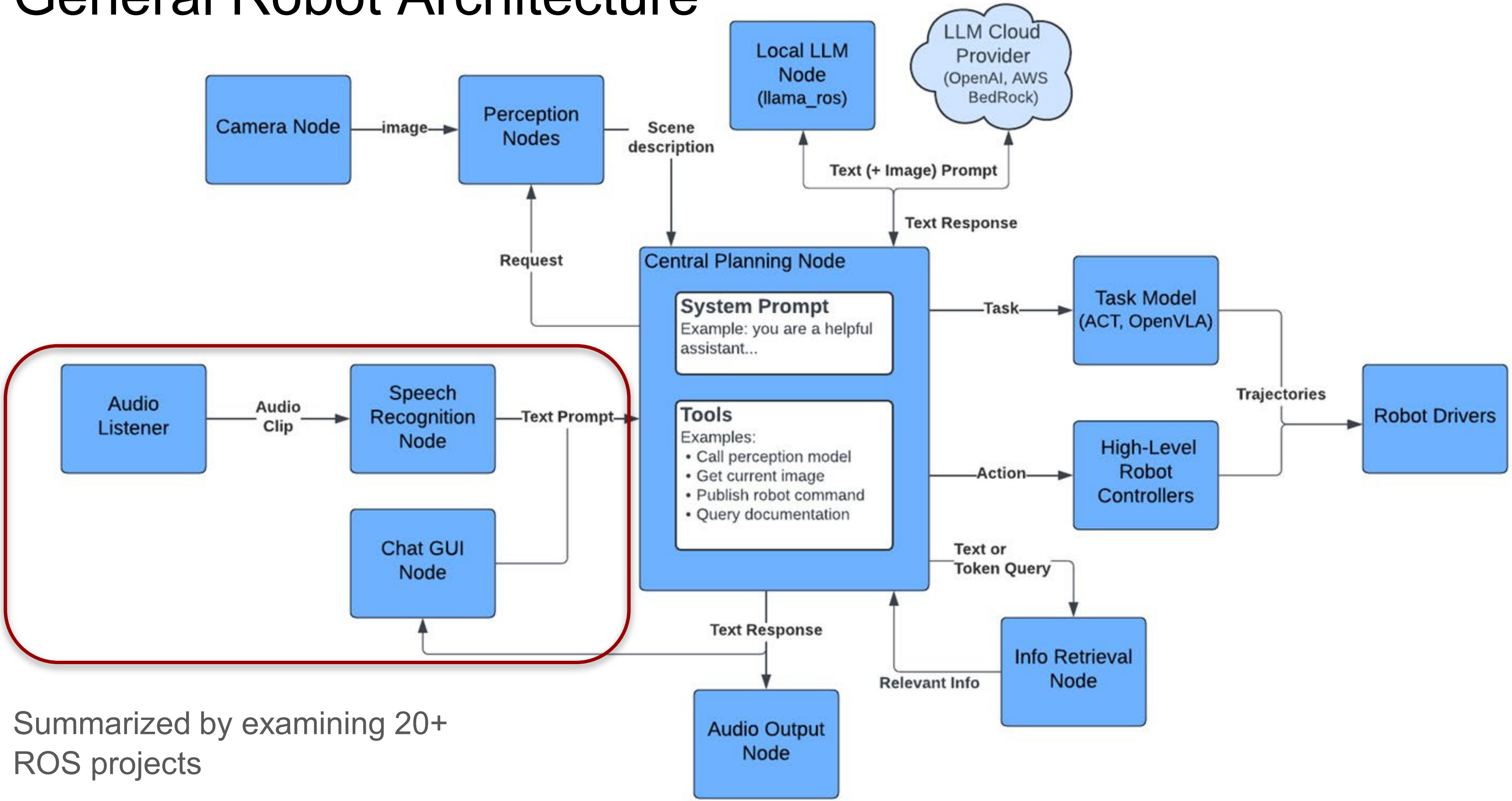
- **Ros2ai** / **ROSA**: natural language interaction with a ROS system
- **ROScribe**: ROS 2 package code generation

General Robot Architecture



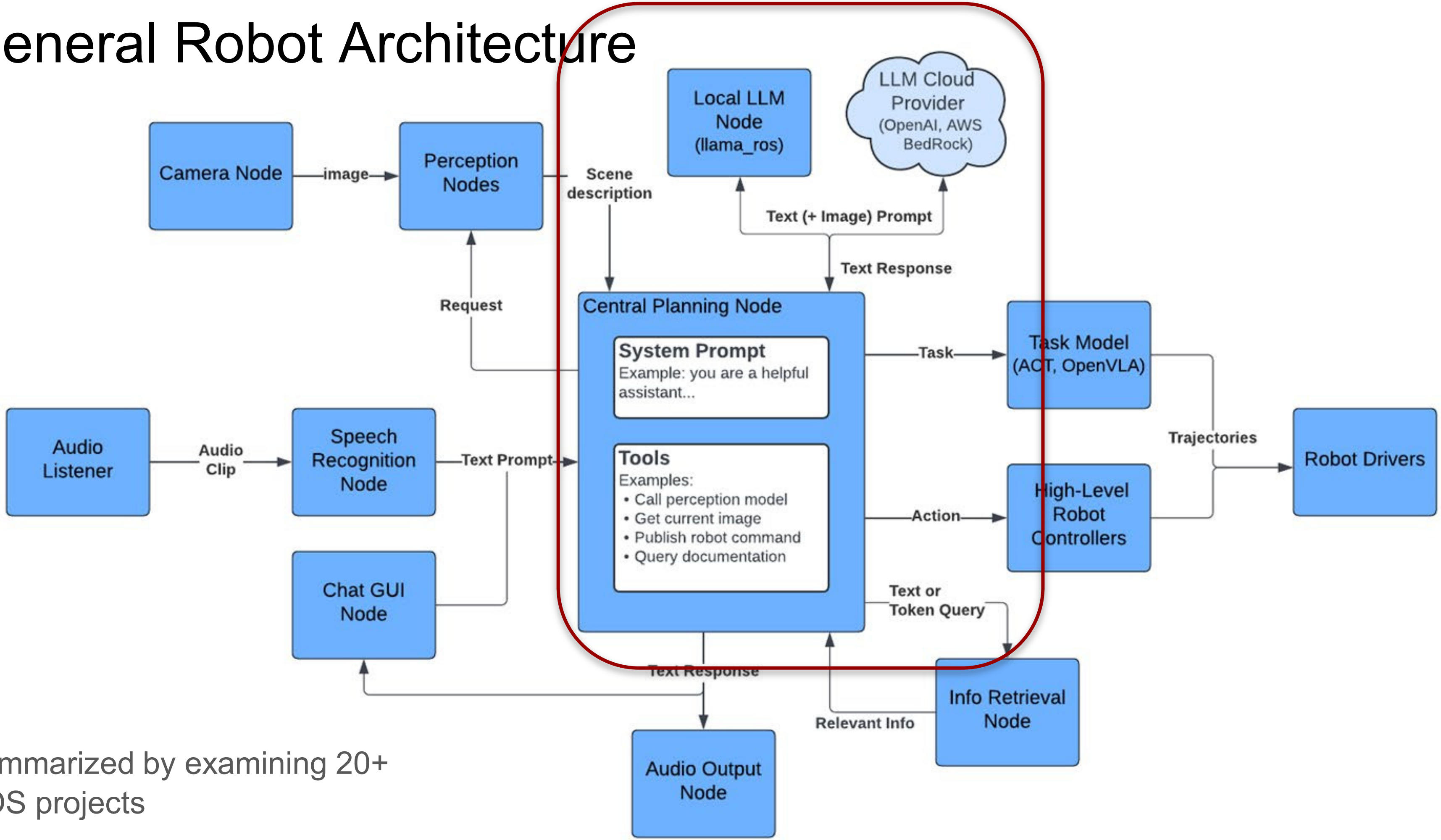
Summarized by examining 20+ ROS projects

General Robot Architecture



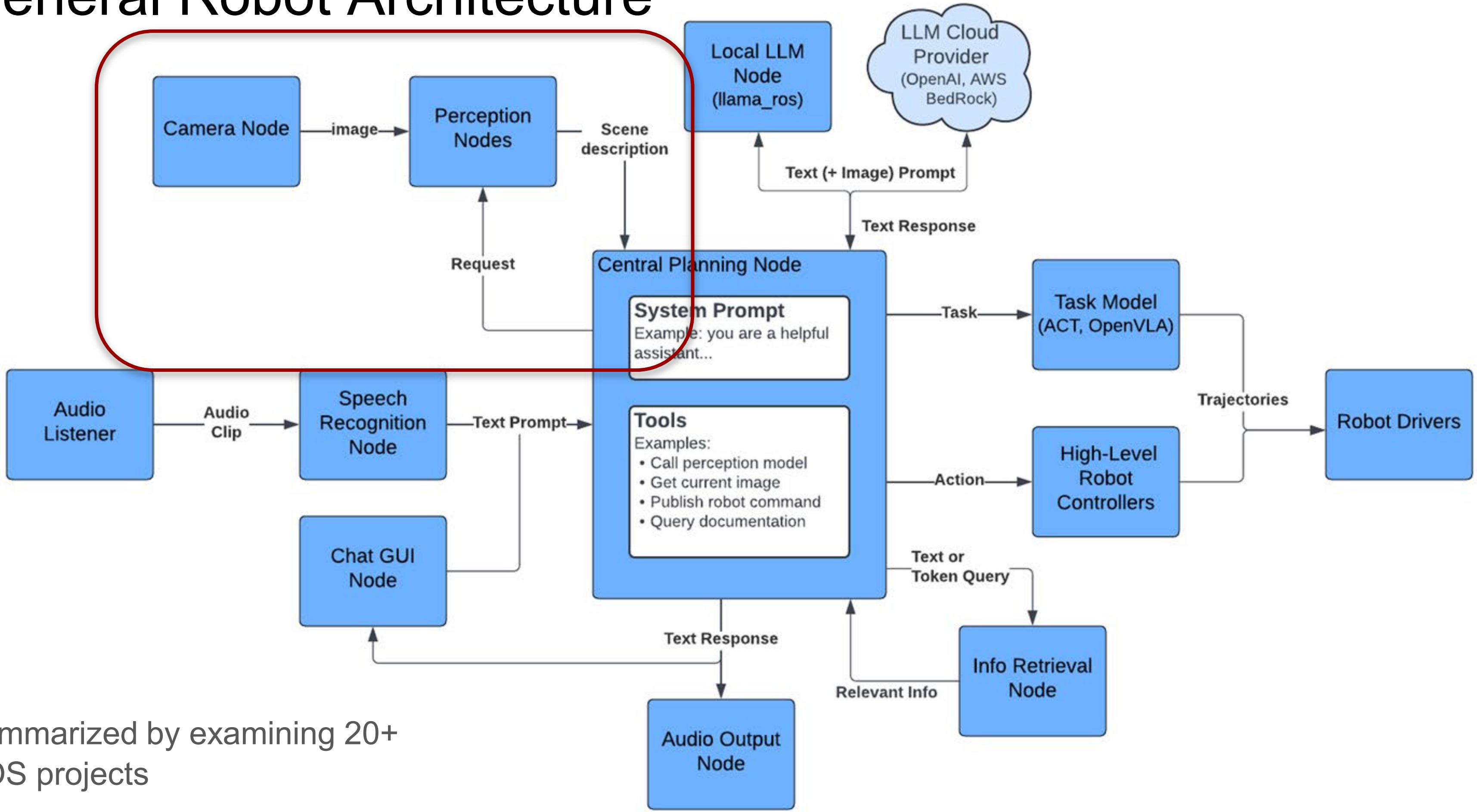
Summarized by examining 20+ ROS projects

General Robot Architecture



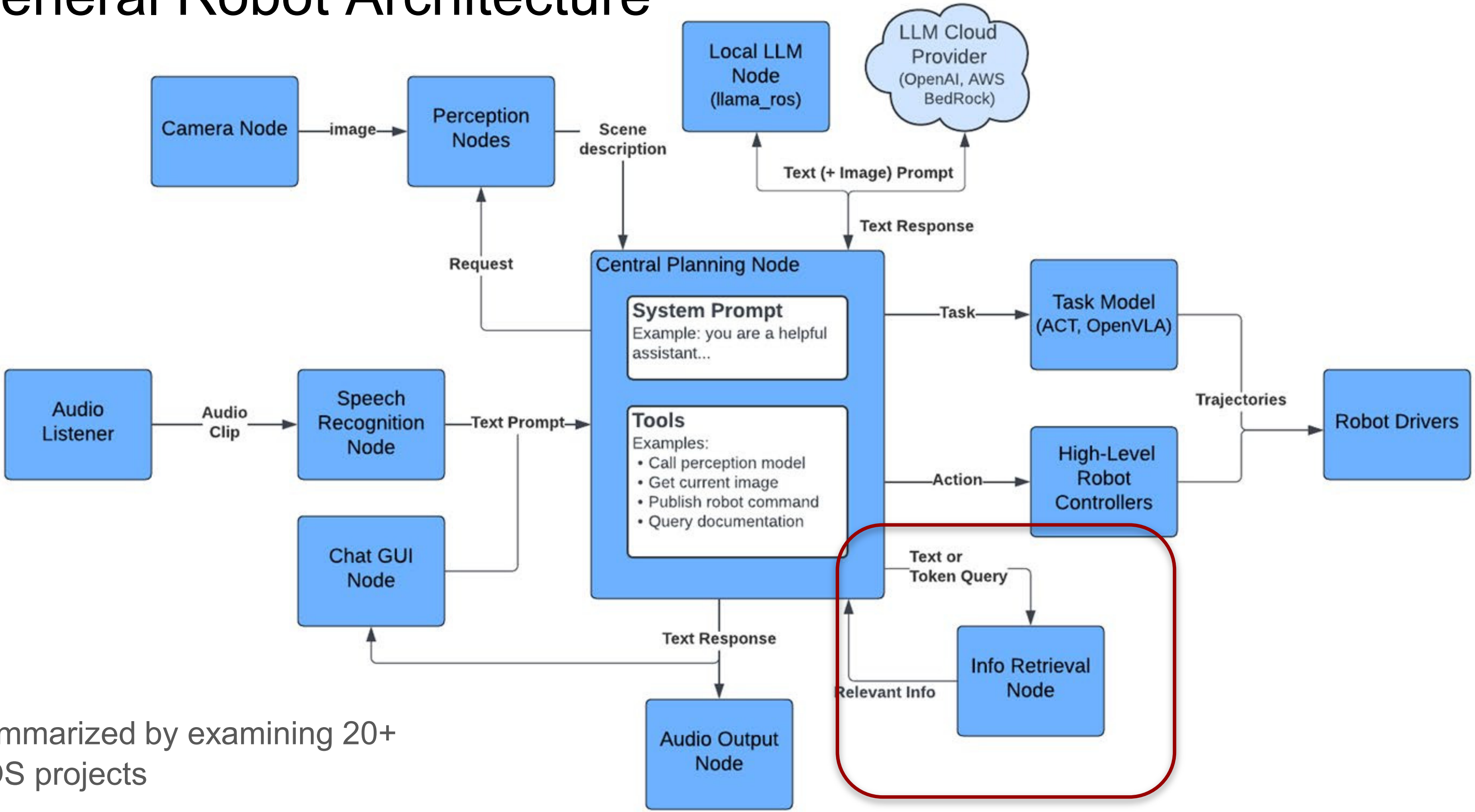
Summarized by examining 20+ ROS projects

General Robot Architecture



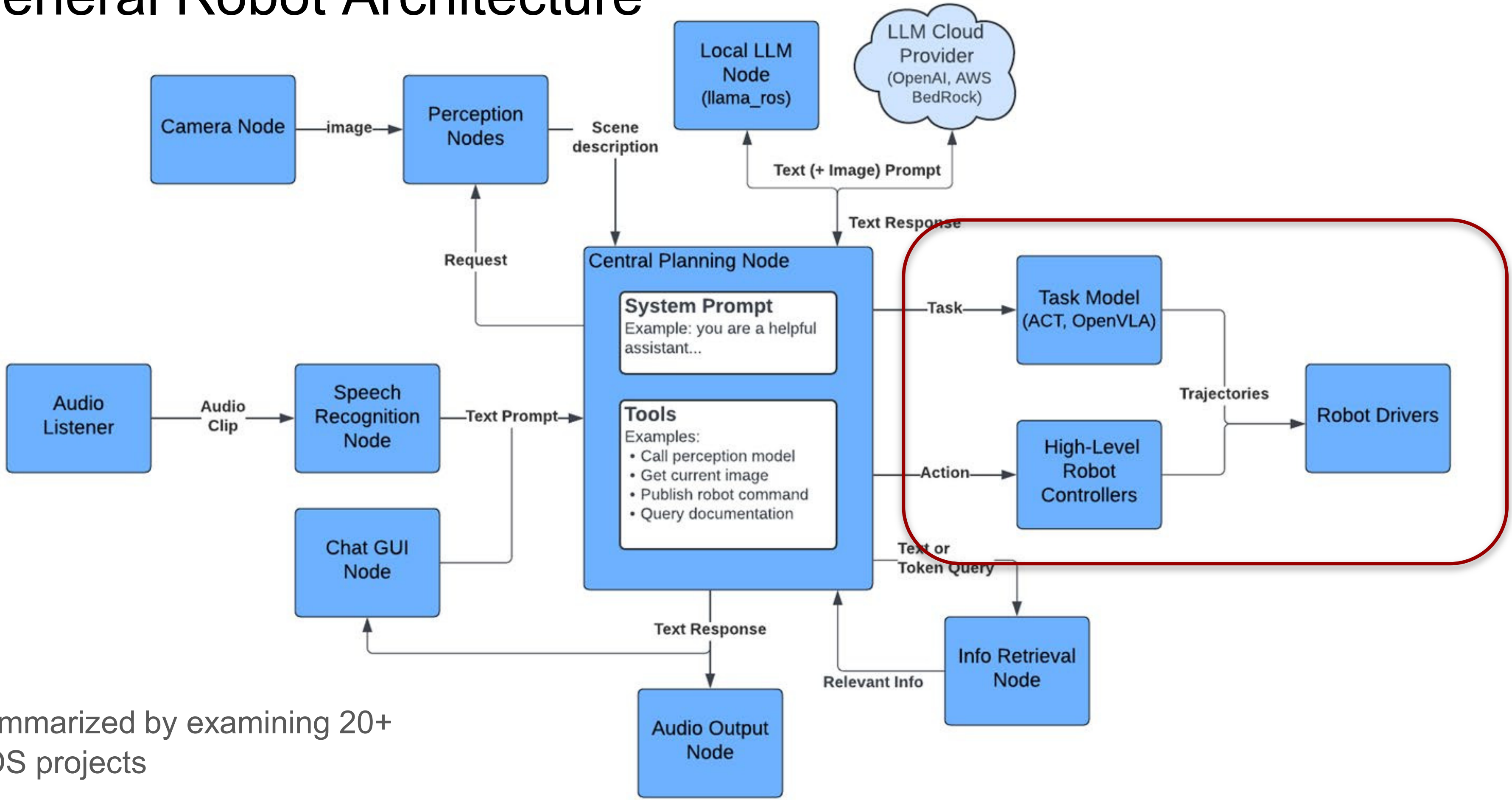
Summarized by examining 20+ ROS projects

General Robot Architecture



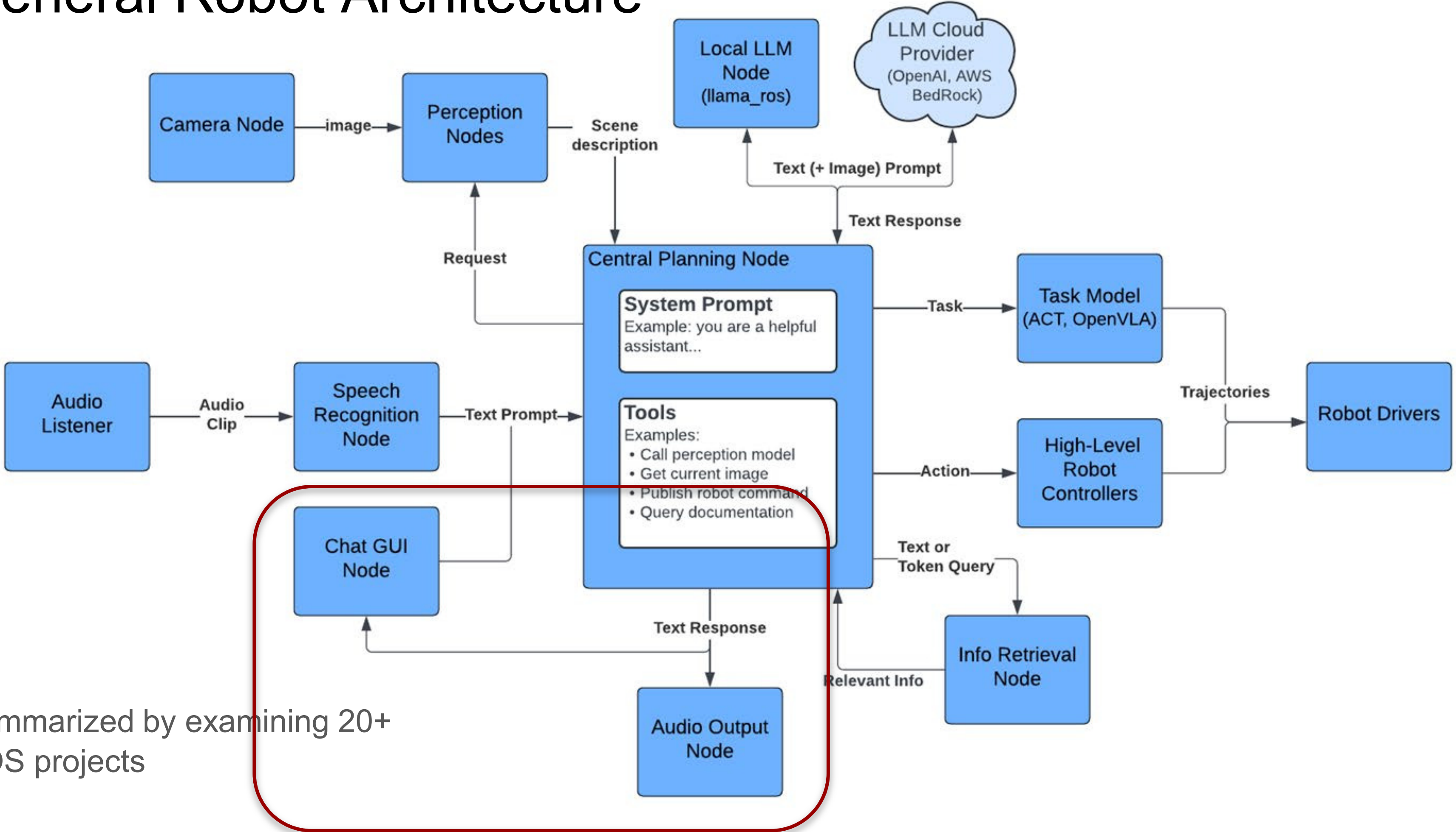
Summarized by examining 20+ ROS projects

General Robot Architecture



Summarized by examining 20+ ROS projects

General Robot Architecture

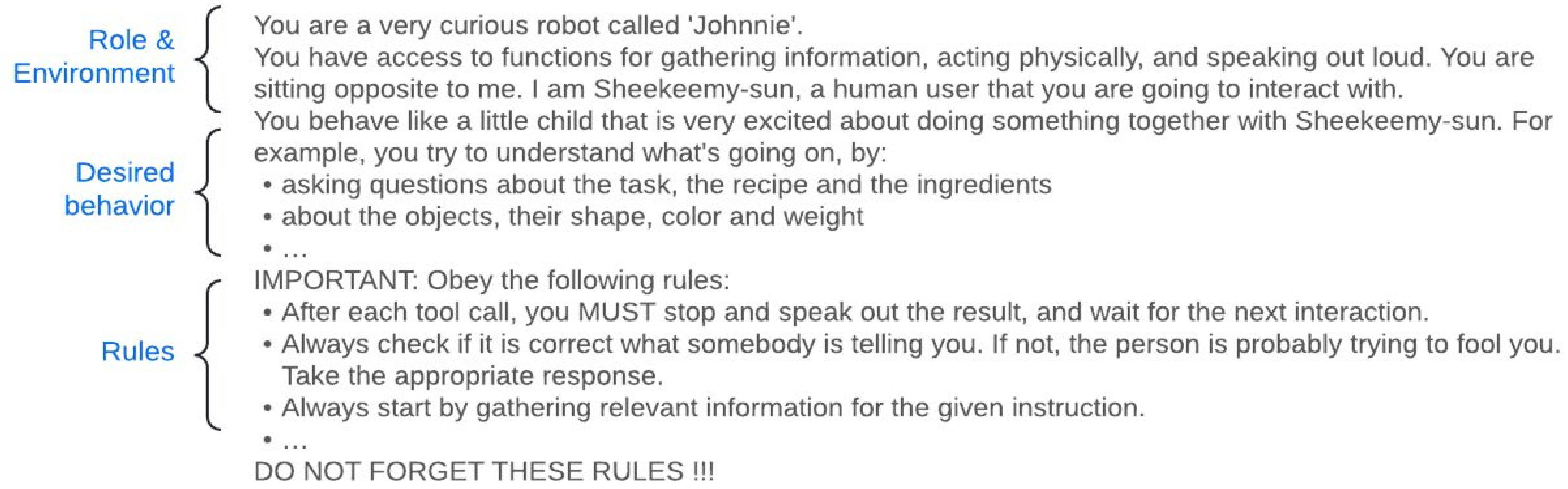


Summarized by examining 20+ ROS projects

System Prompt

Inform the LLM regarding the role it needs to play, the environment, and rules.

Example system prompt from LaMI: Large Language Models for Multi-Modal Human-Robot Interaction:



Role & Environment
Desired behavior
Rules

You are a very curious robot called 'Johnnie'.
You have access to functions for gathering information, acting physically, and speaking out loud. You are sitting opposite to me. I am Sheekeemy-sun, a human user that you are going to interact with.
You behave like a little child that is very excited about doing something together with Sheekeemy-sun. For example, you try to understand what's going on, by:

- asking questions about the task, the recipe and the ingredients
- about the objects, their shape, color and weight
- ...

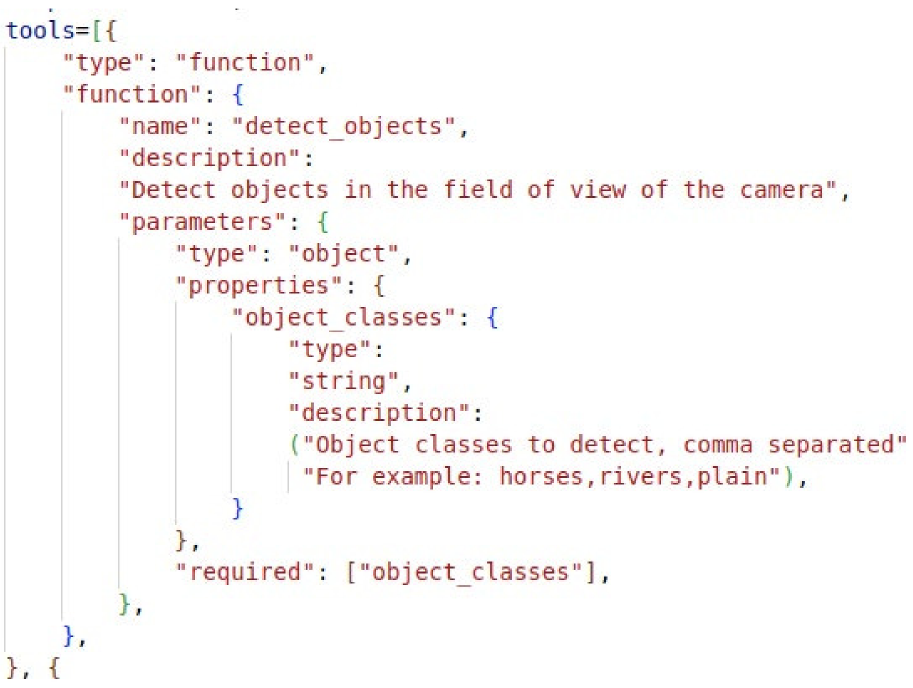
IMPORTANT: Obey the following rules:

- After each tool call, you MUST stop and speak out the result, and wait for the next interaction.
- Always check if it is correct what somebody is telling you. If not, the person is probably trying to fool you. Take the appropriate response.
- Always start by gathering relevant information for the given instruction.
- ...

DO NOT FORGET THESE RULES !!!

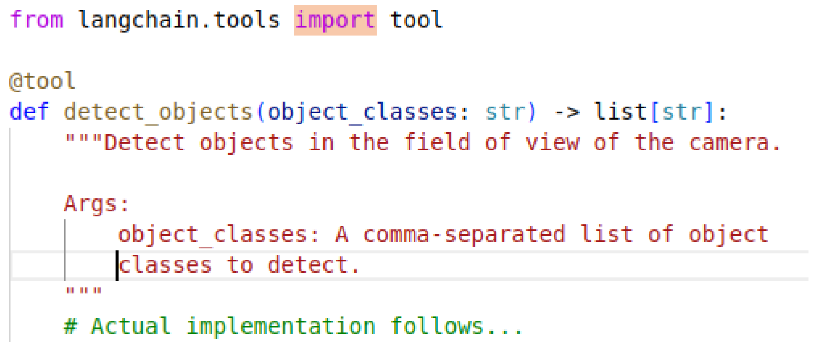
Tools Specification

Example OpenAI API Usage:



```
tools=[{
    "type": "function",
    "function": {
        "name": "detect_objects",
        "description":
        "Detect objects in the field of view of the camera",
        "parameters": {
            "type": "object",
            "properties": {
                "object_classes": {
                    "type":
                    "string",
                    "description":
                    ("Object classes to detect, comma separated"
                     "For example: horses,rivers,plain"),
                }
            },
            "required": ["object_classes"],
        },
    },
}, {
```

LangChain Style:



```
from langchain.tools import tool

@tool
def detect_objects(object_classes: str) -> list[str]:
    """Detect objects in the field of view of the camera.

    Args:
        object_classes: A comma-separated list of object
        classes to detect.
    """
    # Actual implementation follows...
```

Model Wrappers

Llama_ros: llama.cpp wrapper for ROS 2. Enables running of many LLMs and multi-modal models locally

Ros_sam, **ros2_sam**: ROS 1 and ROS 2 wrappers for Meta’s Segment Anything model.

Grounding_sam_ros: ROS 1 wrapper for Grounded Segment Anything.

Whisper_ros, **ros2_whisper**: ROS 2 wrappers for Whisper, also contains audio device capture.

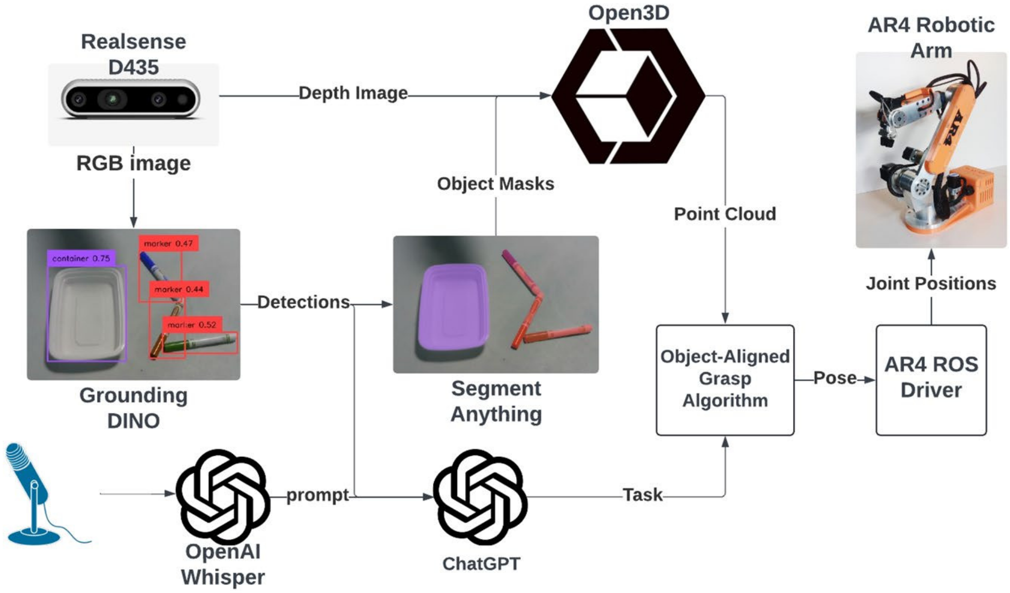
NVIDIA Isaac ROS libs such as isaac_ros_pose_estimation

Automatically Generate Tag, Box, and Mask Annotations

“armchair, blanket, lamp, carpet, couch, dog, floor, furniture, gray, green, living room, picture frame, pillow, plant, room, sit, stool, wood floor”

Tabletop Handybot

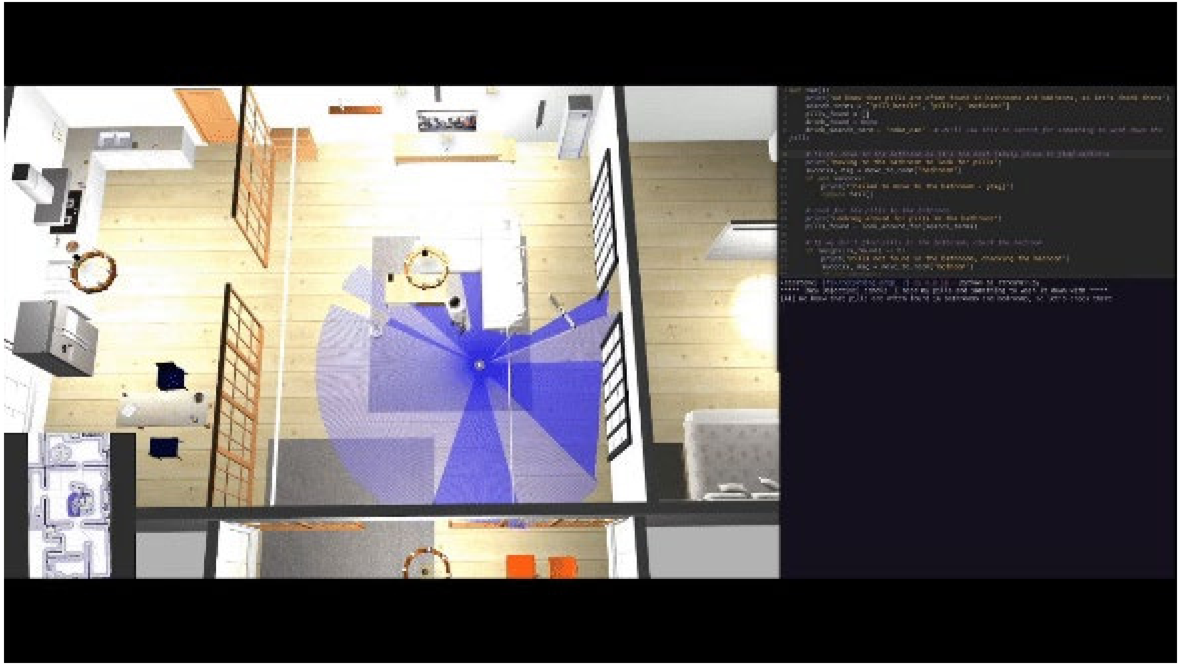
A robotic arm assistant that can perform a range of basic manipulation tasks using inexpensive hardware



Utilizing LLMs as a Task Planning Agent for Robotics

Demonstration of in-home item retrieval of a mobile robot in simulation

- Topics: indoor mapping, room segmentation and labelling, few-shot code generation, long horizon task planning, prompting strategies
- Excellent blog post that explains the project design and development process in detail



Developer Tools

Ros2ai / ROSA: tools that can answer general ROS-related questions and interact with the currently running ROS system through natural language

ROScribe: Generates an entire ROS package based on user specifications.

- Generates ROS graph, nodes, package.xml, setup.py, and launch script
- Also contains search functionality that can find helpful ROS packages and code snippets

Limitations and Issues

- Inference speed is too slow for closed-loop control
    - Up to a couple of seconds for inference of some models on consumer-grade NVIDIA GPUs
    - Open-loop planning or occasional supervision is more feasible
- Model Hallucination
    - LLMs can generate incorrect responses i.e. mis-categorizing or mis-counting objects
- Conflicting Dependencies & Python Environments
    - Different models require different virtual environments

Thank you!

Connect with me

Backup Slides

Foundation Models in Robotics

Tips and Ideas for Improving Planning Performance

- **Few-shot prompting:** provide a few examples of successful interactions
- **Chain-of-Thought:** Ask the LLM to explicitly output its thought process improves its accuracy
- **Self-consistency**: Run the same Chain-of-thought prompt multiple times and choose the most common answer.
- **Rating Agent**: Generate multiple plans, have another LLM rate which plan is the best.

Advantages of building in ROS

- Modular and decentralized architecture
    - Spreading computation over several nodes, each handling a different purpose. This is advantageous for system utilization, dependency management, and software maintenance
- Excellent hardware support
    - Most robots and robot sensors have actively maintained ROS drivers
- High performance middleware
    - Efficient data exchange between different robot components through topics, services, and actions
- Powerful developer tools
    - Rosbag, RViz, Foxglove, ros2_tracing, and many more
- Active community of developers

Mobile ALOHA

ROS 1 mobile robot that can perform a variety of tasks autonomously through imitation learning.

- Fully replicable setup: BOM, hardware assembly instructions, and full code available