dora-rs

Modern Dataflow Framework for Robotics

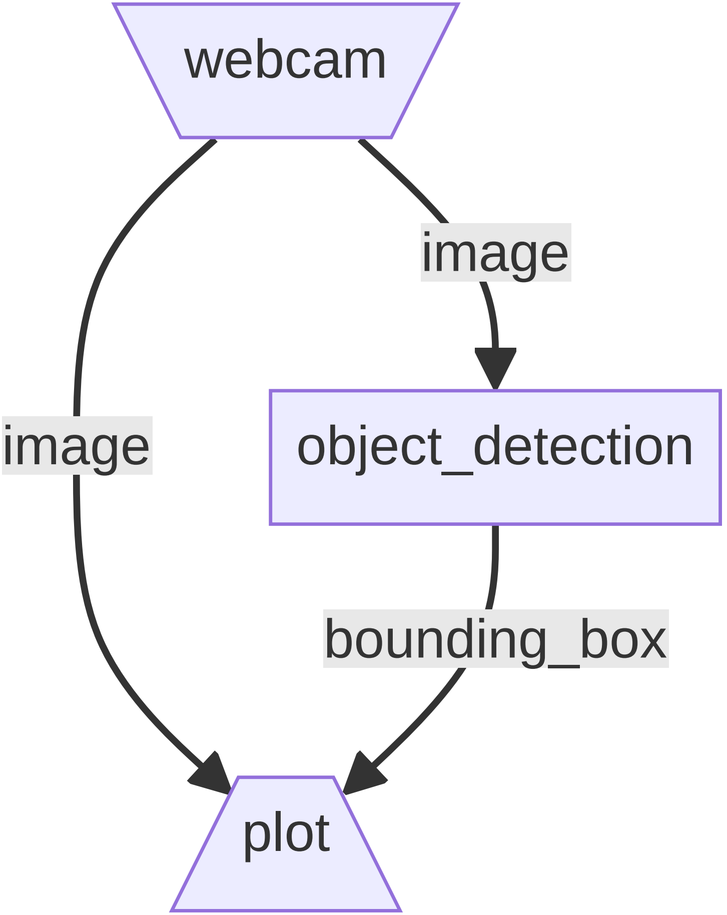
Homepage: dora.carsmos.ai
Repo: github.com/dora-rs/dora

Xavier Tao and Philipp Oppermann

FOSDEM 2024
2024-02-03

Dataflow Frameworks for Robotics

- Model programs as directed graph
  - Nodes represent operations
  - Data is sent along edges
- Advantages of dataflow design
  - Isolation of components
  - Option to use multiple machines
  - Messages can be observed for debugging
- Most popular frameworks: **ROS** and **ROS2**
  - Widely used in research and industry
  - Main languages: C and C++
  - Complex build system



Dora: Motivation

- Make creation of robotic applications fast and simple
- First class support for nodes written in Python and Rust
  - Also supports C and C++
  - Planned: Add support for WebAssembly nodes
- Simple build system
- Easier integration with latest technologies (e.g., AI models)

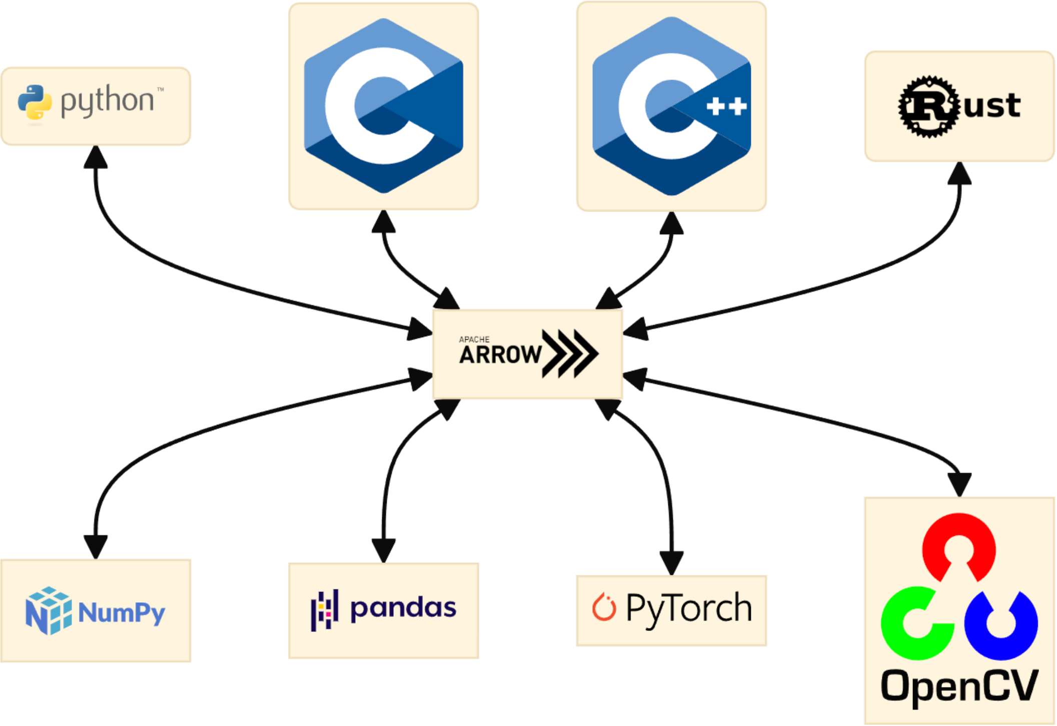
Dora: Design

- Dora dataflows consists of multiple nodes
    - Each node is a separate process → isolation, fairness, flexibility
- Nodes communicate through messages
    - Each node defines a set of inputs and outputs
    - YAML declaration file maps inputs to outputs of other nodes:

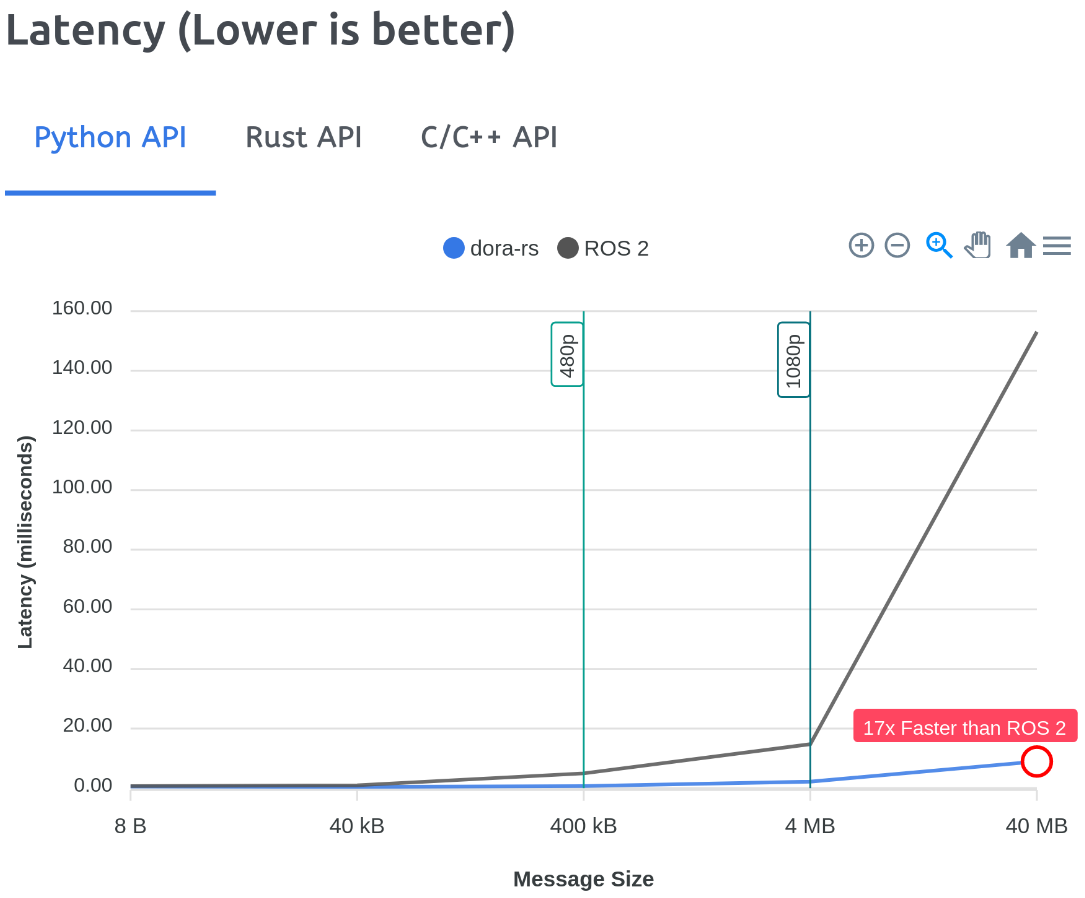
```yaml
nodes:
- id: node_1
  custom:
    outputs:
      - some_output
- id: node_2
  custom:
    inputs:
      foo: node_1/some_output
```

Dora: Zero Copy

- Send messages via **shared memory** on the same machine
- Messages use Apache Arrow data format



Python Performance



Latency (Lower is better)

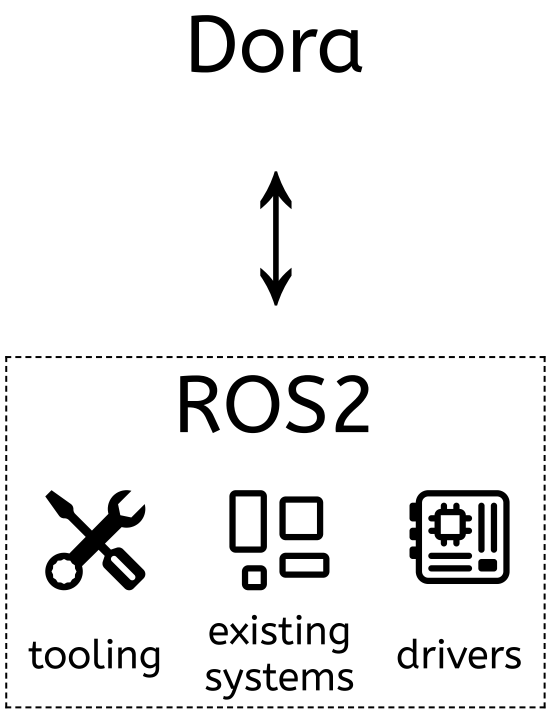
17x Faster than ROS 2

Dora: ROS2 Bridge

- Allows gradual migration of existing ROS2 applications
- Makes it possible to use ROS2 tooling with Dora

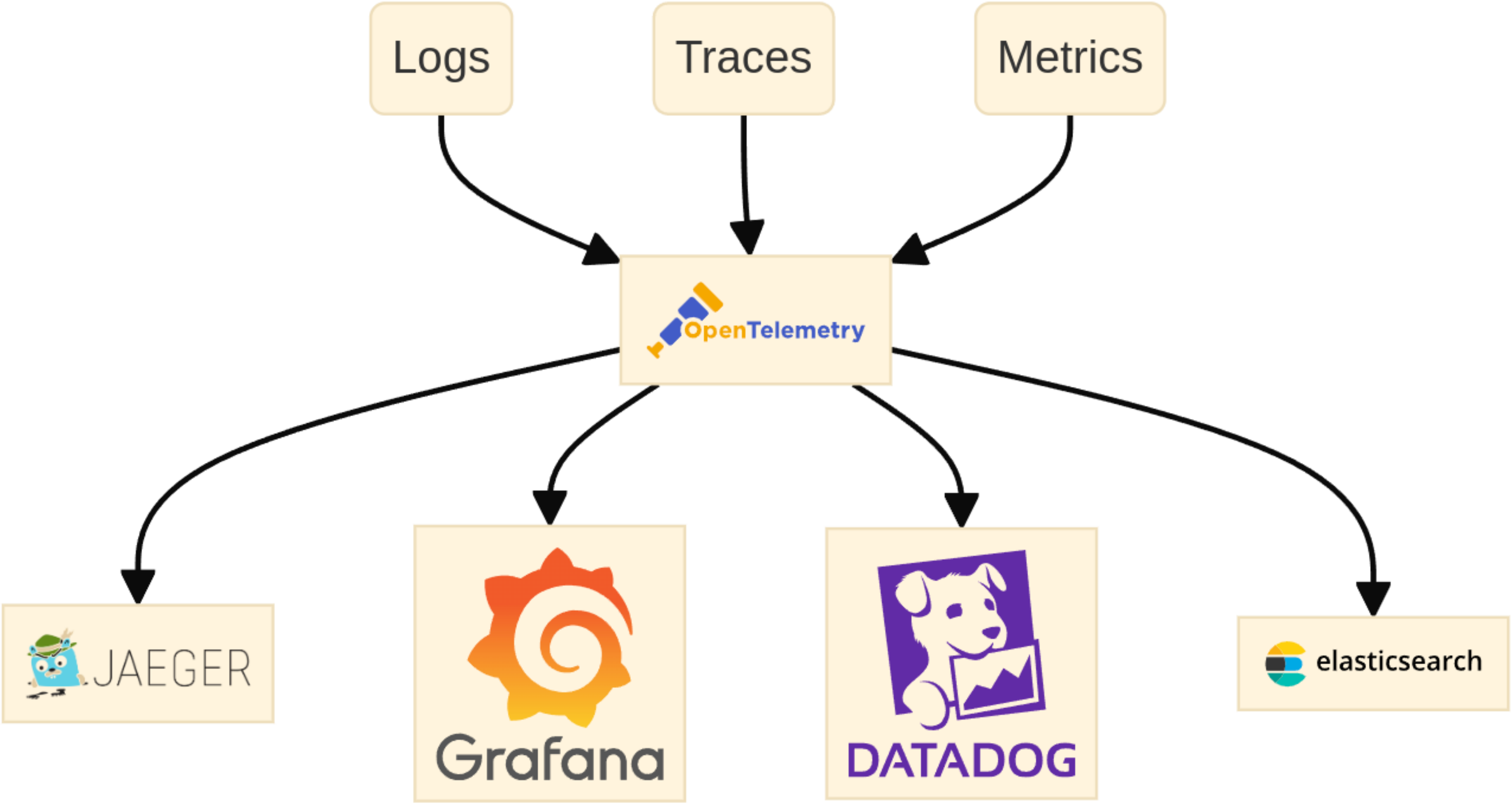
Implementation (in progress)

- Interface via DDS middleware
  - no need integrate with complex ROS2 build system
- Parse ROS2 message files
  - Autogenerate Rust and C++ bindings
- Automatic type conversions between:
  - ROS2 message types
  - Arrow data format
  - Rust types



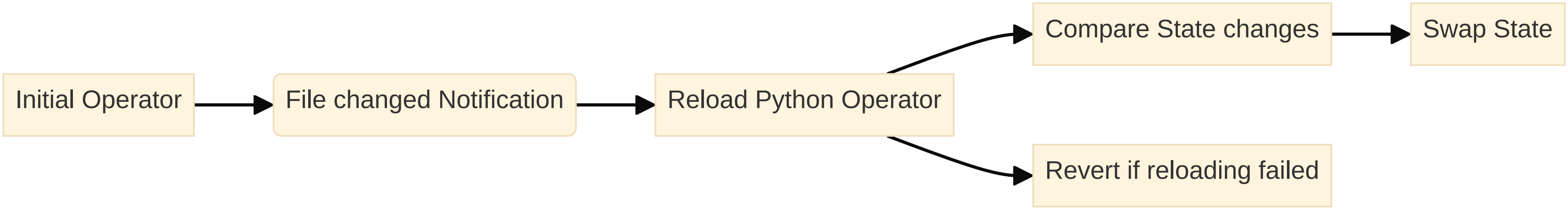
Dora: Opentelemetry

- Uses Opentelemetry for logs, tracing and metrics



Dora: Hot Reloading for Python

- Enables code change in real time and keep current states intact.
- Removes the need to reset robots at each iteration step.
- Try out code generative AI in real time.



Initial Operator

File changed Notification

Reload Python Operator

Compare State changes

Swap State

Revert if reloading failed

Demo: Voice controlled Robot

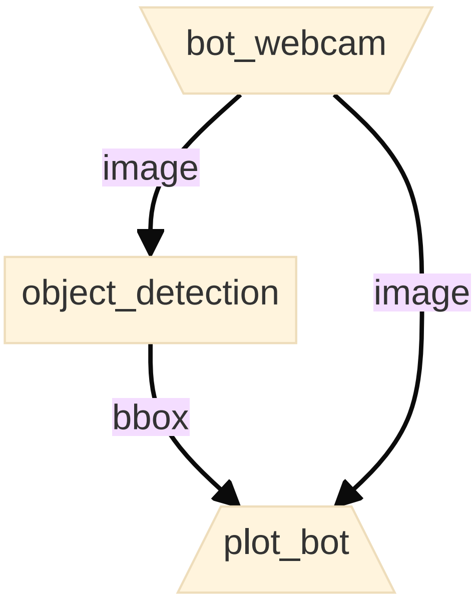
Setup:

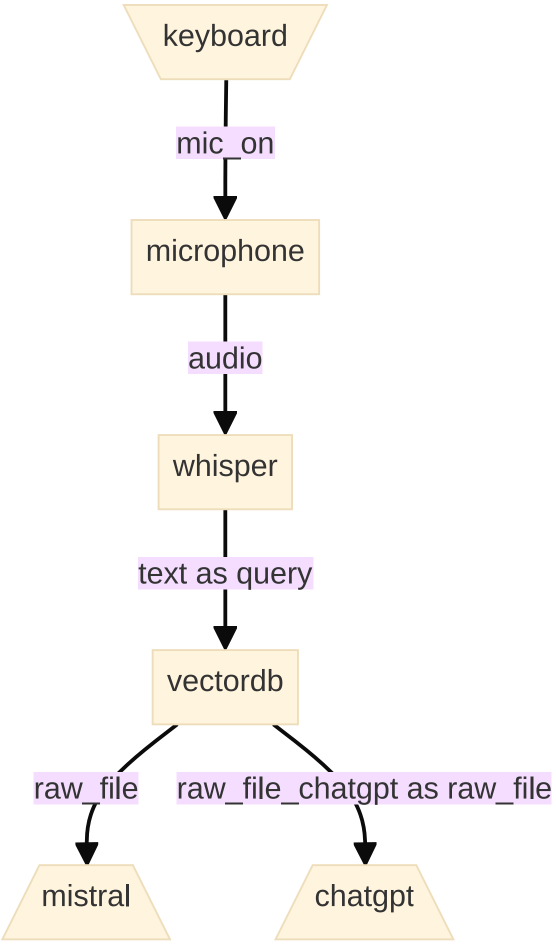
- Robot controlled via an SDK
- Microphone
- A Whisper node to convert Speech to Text
- A LLM to convert "text to code". Either Mistral or GPT4.
- 2 Webcams: One on the robot and one outside of the robot

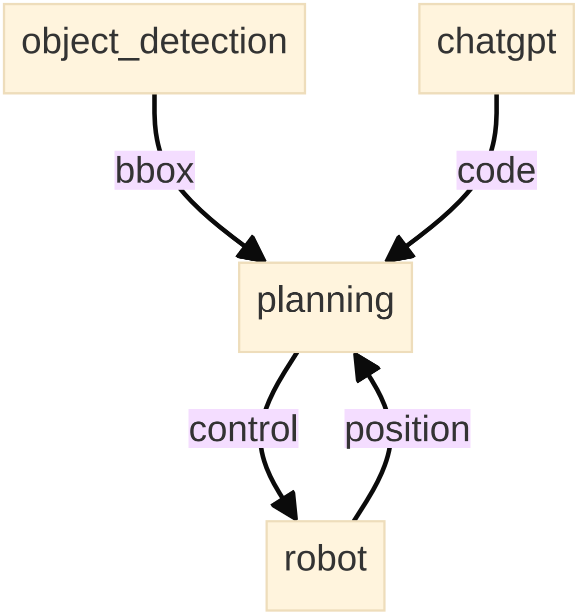
Additional nodes:

- Yolov8 to detect object in an image
- VectorDB to detect which source code to modify

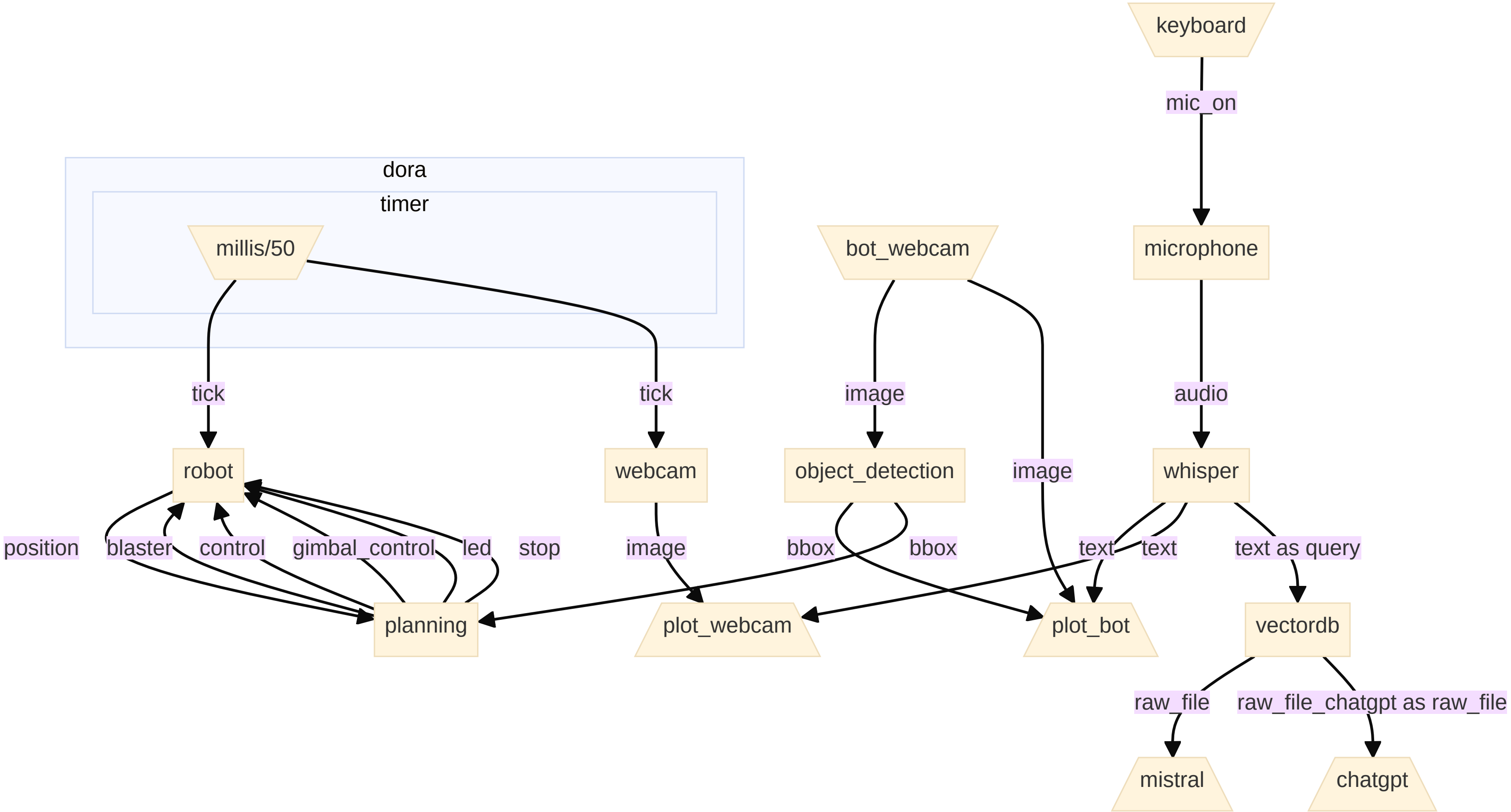
Voice controlled Robot Graph







Quick Robomaster demo



Feature Roadmap

Current

- Rust, Python, C/C++ API
- Zero Copy & Arrow
- Opentelemetry
- Hot-reloading (Python)

Planned

- Data Log & Replay & Visualization
- Remote machine
- Elastic Resources
- Dynamic Dataflow

Hoped

- Fleet Management
- Time Constraints
- Deadline
- Fault tolerance
- Redundancy

Thanks for listening 😃

Still in active development → we love contributions!

- github.com/dora-rs/dora Stars 326
- dora.carsmos.ai