

ros2ai Next-Gen ROS 2 CLI

ros2ai is a next-generation ROS 2 command line interface extension with OpenAI and [Ollama](#)

Everything Here!!!

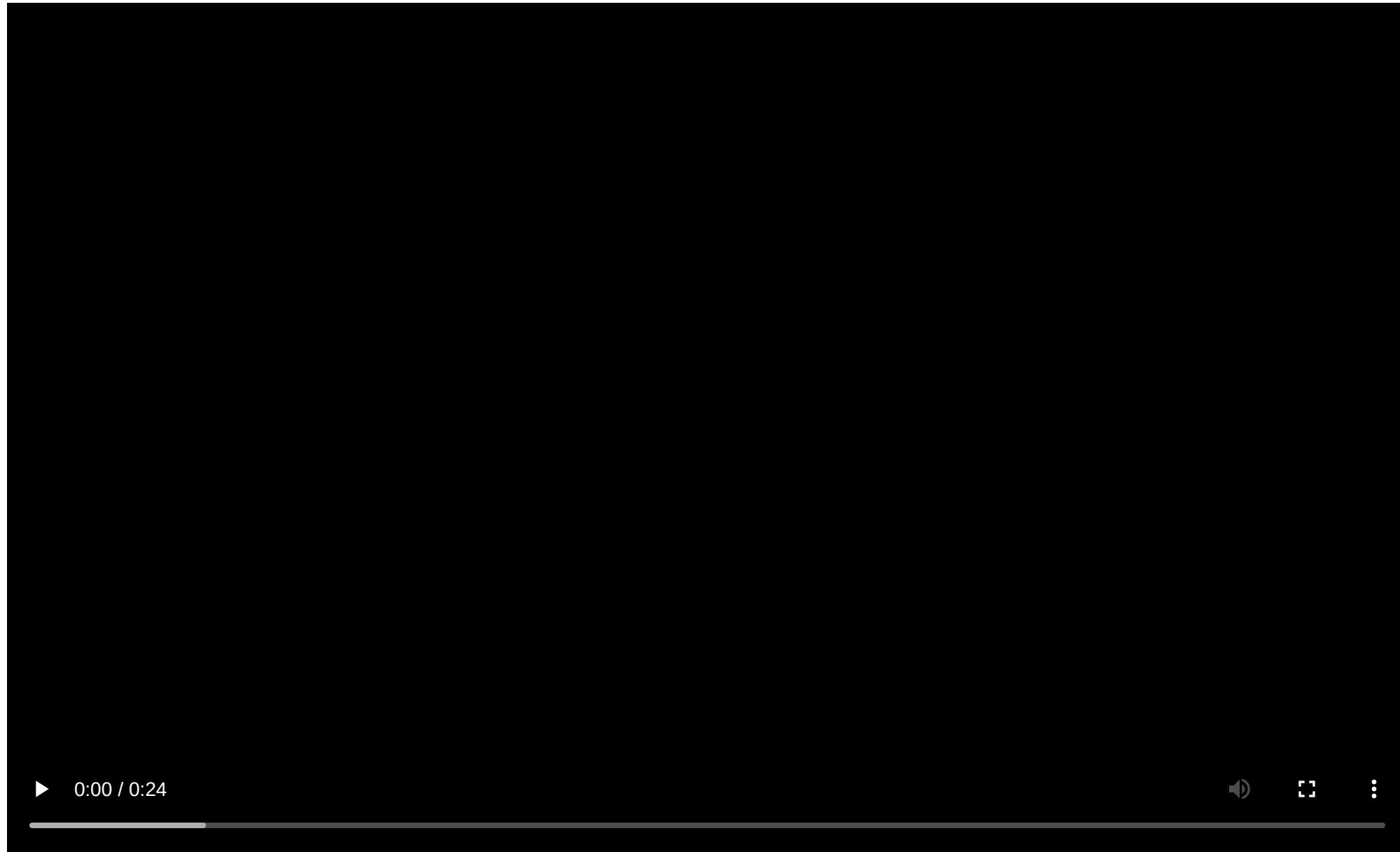
Issues/PRs Welcome







ros2ai Next-Gen ROS 2 CLI



Motivation

- (Originally just for fun 😅)
- Quickly get the answers against questions.
- Multiple Language Support.
- Support beginners and students. (your best trainer)
- Easy to use for everyone.
- Dedicated / AI based Support (your own concierge)
- Bridge / Proxy to [LLM](#)

Background

I want the answers w/o searching...

browsing, clicking and typing many times to get to the information does not work for me. I need **what**, not **where**. all these are small things, but can be easily compiled up to mountain, especially beginners.

I really do not care how to use ros2cli...

many sub-commands, options, and arguments. besides, those could be deprecated or removed time to time... this does not work for me. actually i just want to ask "what parameters are available?", "check the details for the topic /chatter", but before that i need to know how to do that... this is already barrier for beginners.

it would be better for maintenance if someone abstracts this to the user, so that maintainers can just go ahead to remove, change options w/o deprecation or certain soak time...

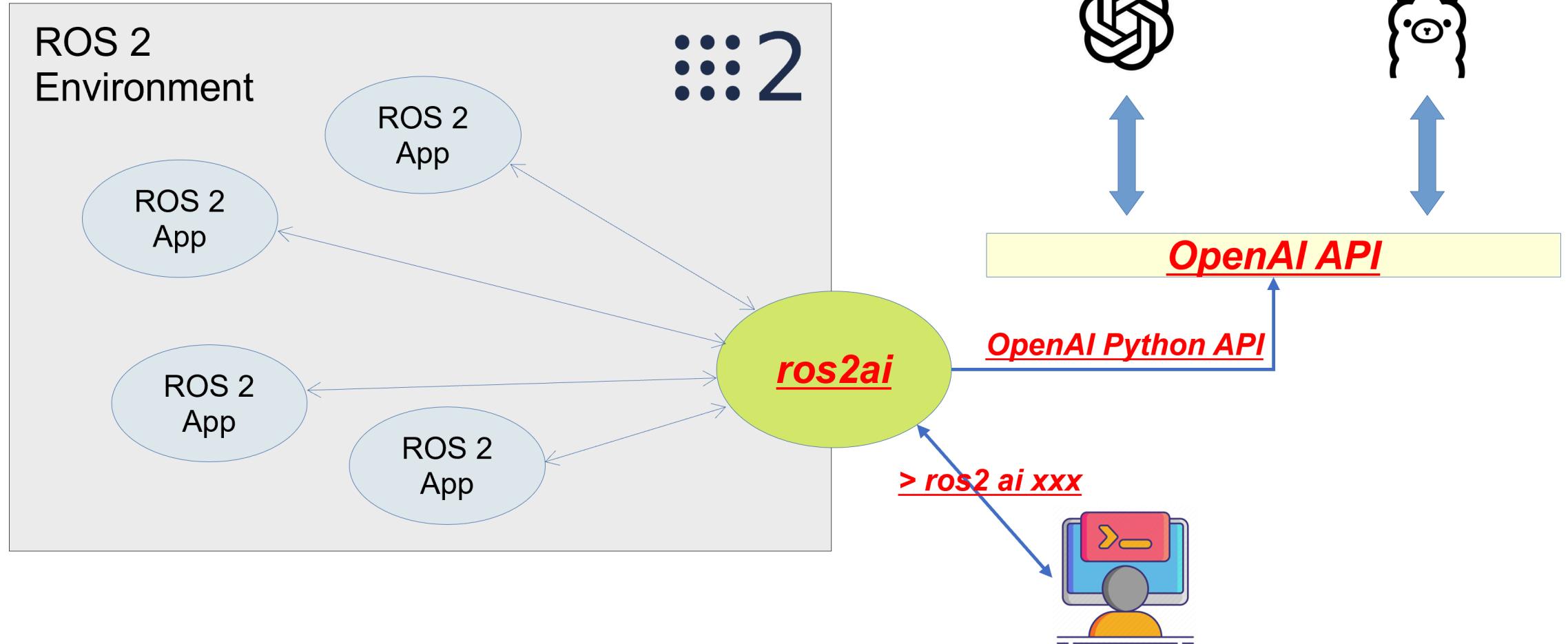
Multi-Language Documentation

OSS projects tend to add multi-language support documentation under mainline doc. this is really likely, and they do. but the question is ***Does it really work?***... IMO, it does (will) not. eventually those docs will be outdated, not maintained, not scalable, not precise and once that happens these docs would be garbage.

i believe that is exactly where AI comes to play. instead of paying the resources to non-scalable things, focus on the mainline with precise information which is available for any languages.

Design

- ***SIMPLE***. one of the original motivation, it has to be simple as much as possible. That is said that it would be even better to remove all sub-commands, just `ros2 ai <whatever your request>` if that is possible.



- `status` command is to check if `ros2ai` is configured properly.
- `query` command is to ask any questions related to ROS 2. This is a single-shot completion, no session is supported currently. OpenAI system role attribution is set to default, but can be reconfigurable.
- `exec` command is that AI executes the appropriate command based on the requests. OpenAI system role attribution is set to default, but can be reconfigurable.

Future Plans

Model Comparison

- Experimental performance and accuracy comparison over different LLM models.

API Parameter Adjustment

- for more stable AI behavior.
- performance and accuracy comparison.
- latency for user experience.

Context-Awareness

currently, `ros2ai` only supports single-shot completion API, that means we can not rely on the previous questions or answers to make further requests. e.g) "Is /chatter topic available?", "Subscribe it!". This ***it*** should be recognized in the same context or session. If this command is issued, prompt should be initiated to keep the session alive until the entire session is closed by the user.

Daemonized Agent

`ros2ai` can instantiate the agent daemon process as ROS 2 service, so that even other ROS 2 application running in the network can use OpenAI API via this proxy. those ROS 2 application can be agnostic from LLM specific APIs but ROS 2 service interface.

Audio Interface Support

It would be really helpful to iterate via audio system instead of typing anymore. So the user can just talk to the `ros2ai` agent and then `ros2ai` answers any questions and executes the corresponding operations as user wants to do.

Function Calling

for more user friendly experience, user should not be aware of `ros2ai` sub-command at all such as query, execute. this is actually against the design policy for `ros2ai`.

we could take advantage of **Function Calling** to conceal these sub-commands, and internally categorize the request based on the user input.

Fine-tuning

- ROS 2 general fine-tuning
 - this requires scaled training dataset for ROS 2, agnostic from user environment. could be distro specific. so that AI can response more precisely based on questions and requests.
- User specified environment tuning
 - this dataset should be uploaded to help more user specific problems and questions. if this is doable, `ros2ai` agent can be the 1st customer support for anyone?