

ROS Interface for gRPC model servers

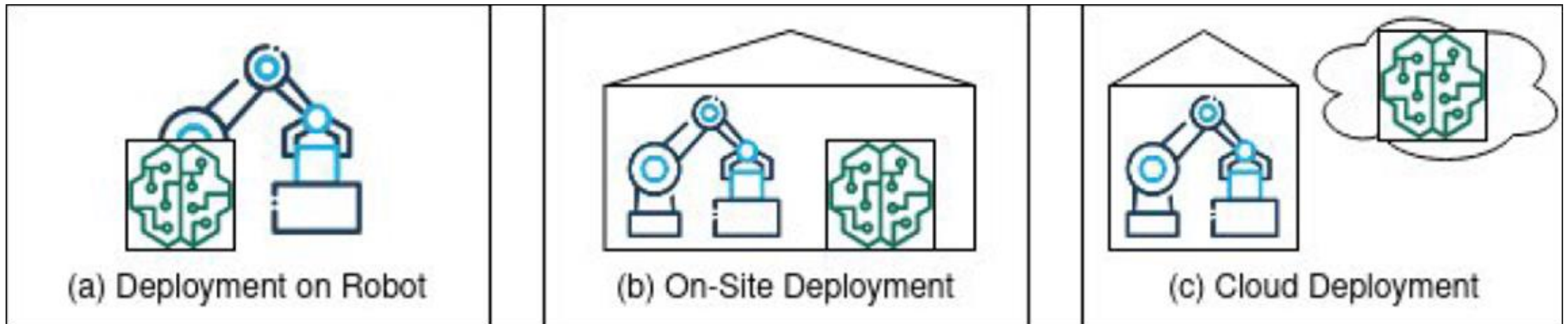
Ragesh Ramachandran (Fraunhofer IPA)
Ansgar Radermacher (CEA)

AI4EU Platform goals

Deployment of models



- The model is deployed directly on the robot, respectively its controller unit.
- The model is deployed centrally on-site.
- The model is deployed in a cloud environment



- ROS currently does not have a standard infrastructure for using ML models
- Every iteration on model required rebuilding the complete workspace
- gRPC model server will remove the ROS dependency on ML libraries
- Leveraging the combined capabilities of ROS and AI4EU platform

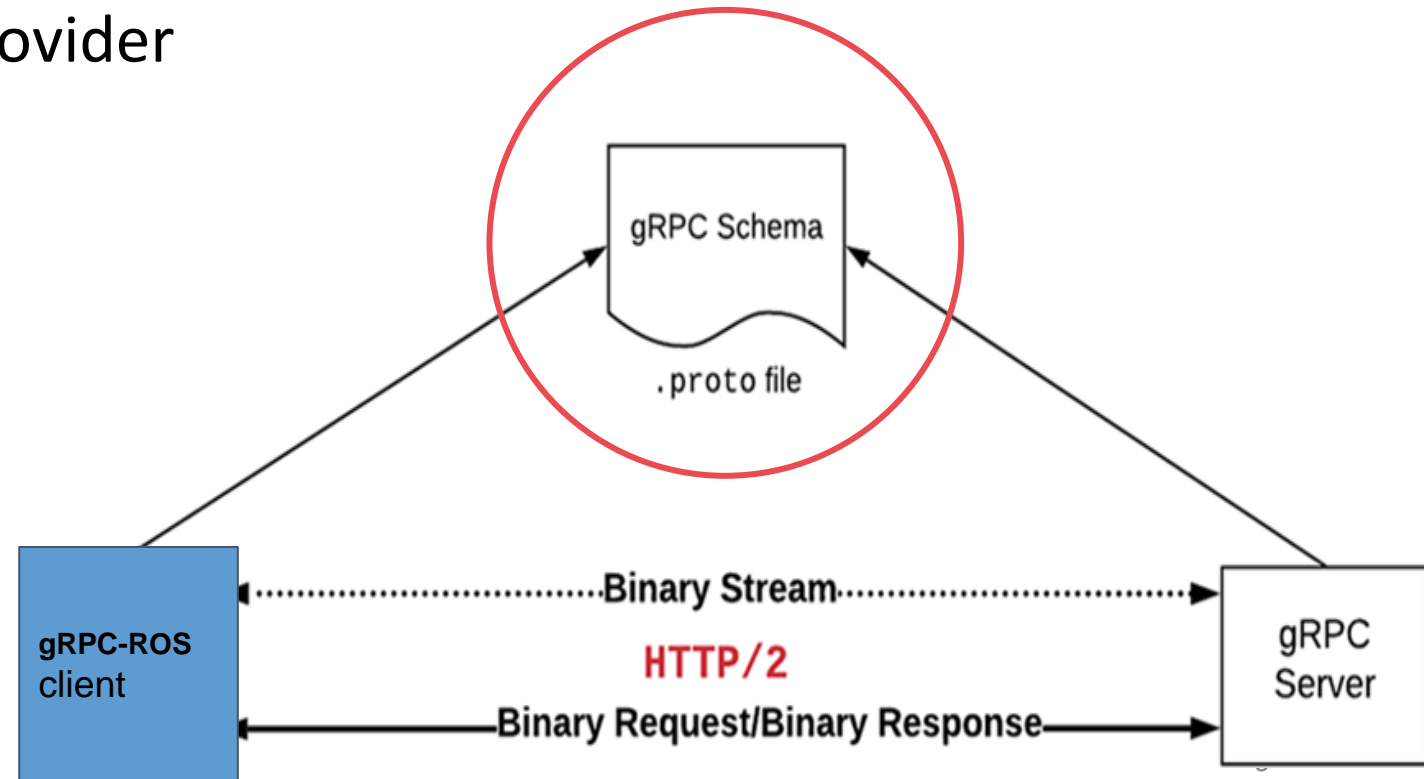
darknet_ros/ETHZurich

```
net = load_net("cfg/tiny-yolo.cfg", "tiny-yolo.weights", 0)
meta = load_meta("cfg/coco.data")
```

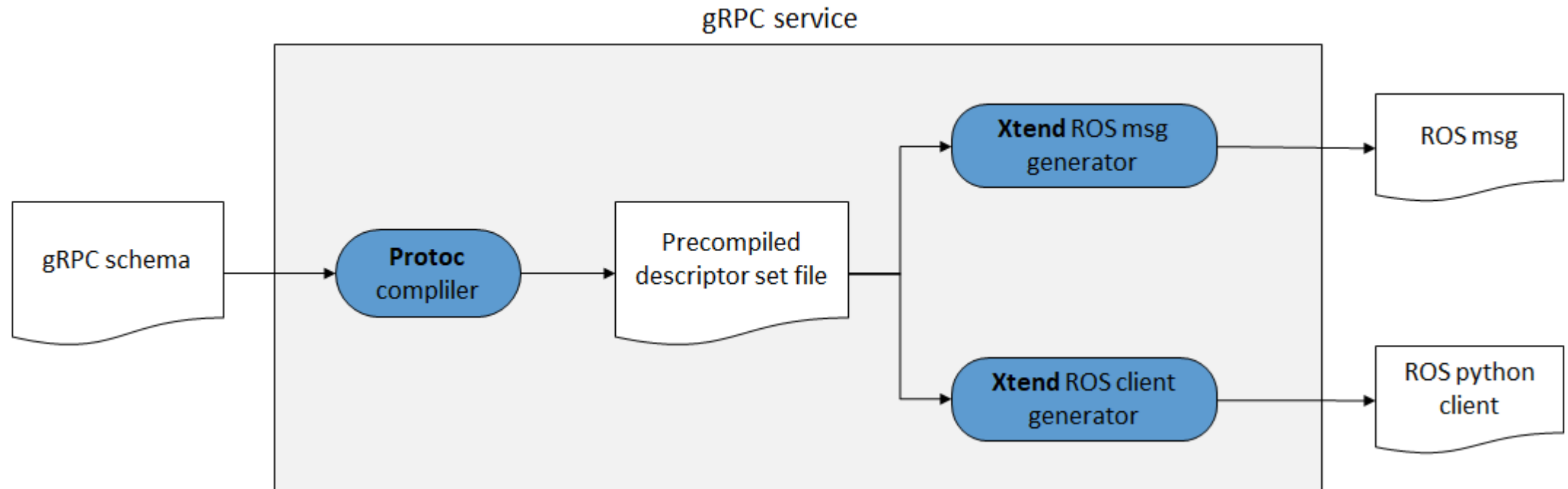
1. Develop a standard framework in robotics for using ML models with ROS
2. Develop an interface from ROS to gRPC model servers from AcumosAI

Impact on Model provider

- Our solution begins from the **gRPC schema** provided by the model provider.
- No additional resources requirement from model provider
- **Zero impact** on the model provider



Integration of package generator to AI4EU pipeline



About the generator

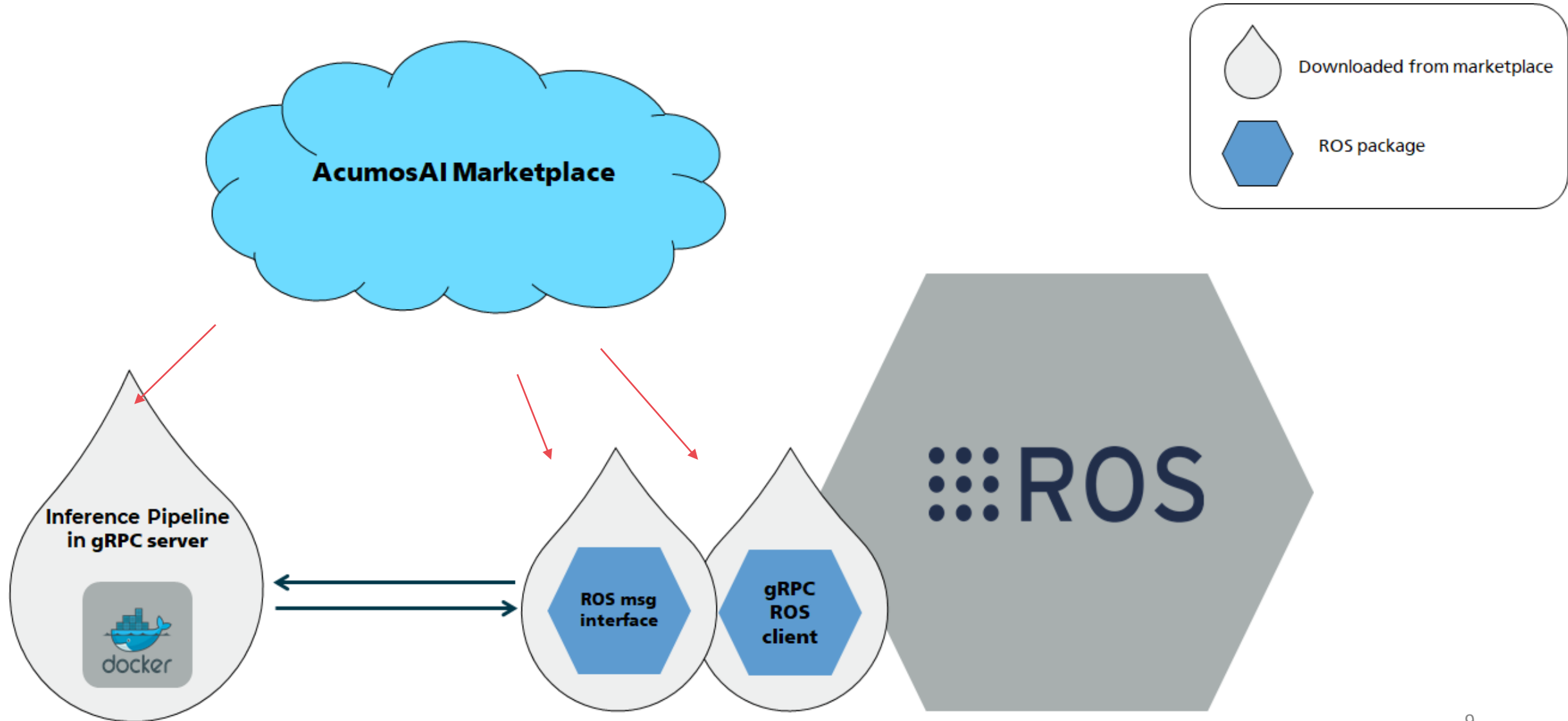
- **Input:** a precompiled descriptor set file
(useful? proto file, automate descriptor set creation via protoc)
- **Outputs** (see next slide)
 1. A corresponding ROS2 message package including build files
 2. A ROS2 gateway in python that forwards data in both directions

Implementation choice

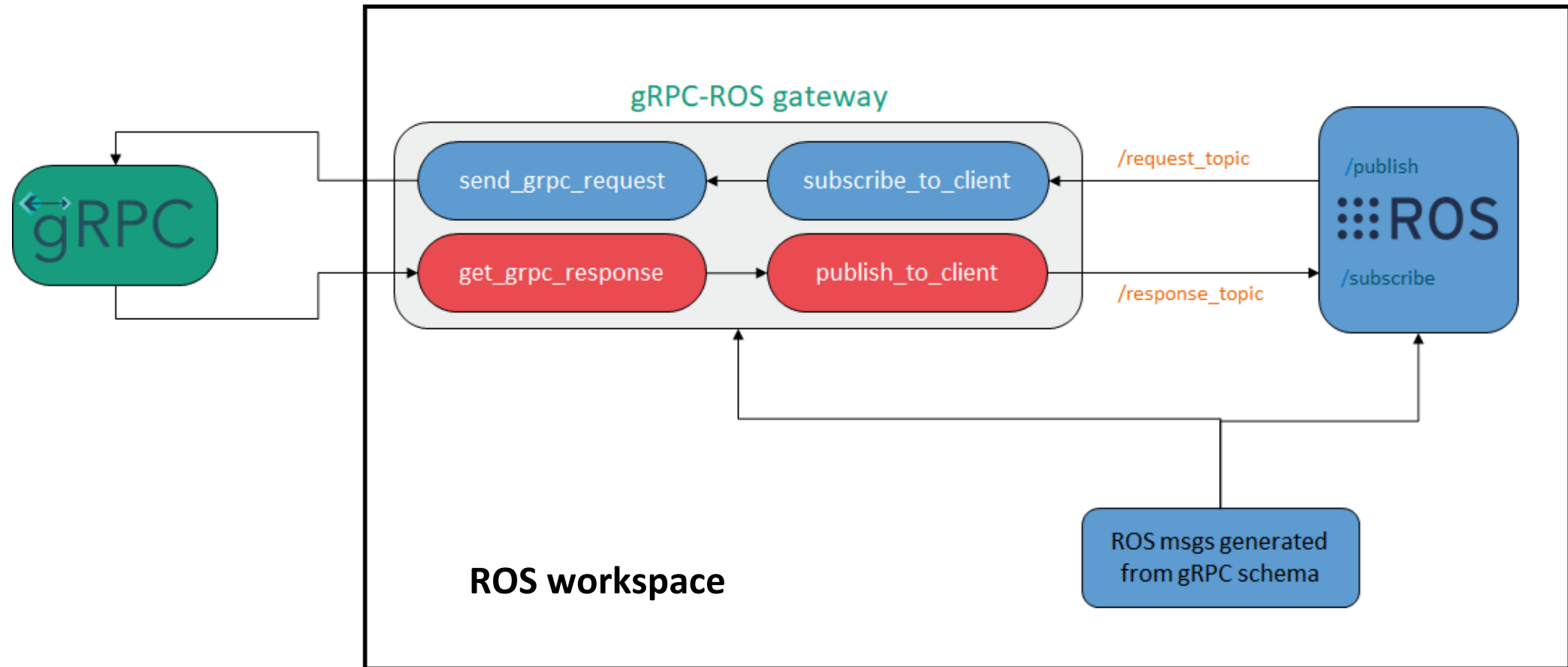
written in Java/xtend (<https://www.eclipse.org/xtend/>)

Text templates facilitate text creation

The deployment end user



How the interface works



- Package the generator to a gRPC service
- Integrate into AcumosAI pipeline
- Support multiple RPC services per proto file
- Extend support to auto generation of ROS1 (required?) and ROS2 C++ gateway

Priorities?

Thank you

Feedback?