Simulation of Smart Factory Communication with ROS

Joel Mattila

Instructors: Juuso Autiosalo & Riku Ala-Laurinaho

8.4.2021

Aalto-yliopisto
Aalto-universitetet
Aalto University

Motivation

- **Machinaide project is building virtual factory where machines communicate with each other**
- **Research questions:**
  - How different machines and entities in smart factory should communicate?
  - What is the most effective way for communication?
- **GOAL:**
  - Suggestion for best communication model in smart factory

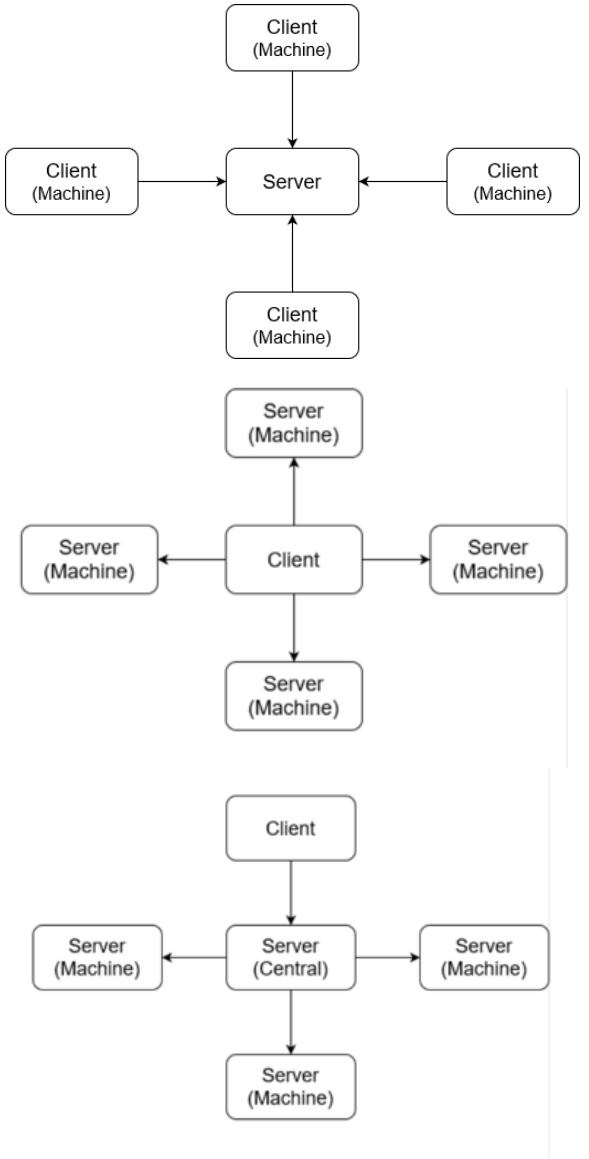
Agenda

- Background: Communication methods
- Research method: Simulation with ROS
- Next steps: Implementing communication methods with ROS

Communication Methods 1/3

Client-Server

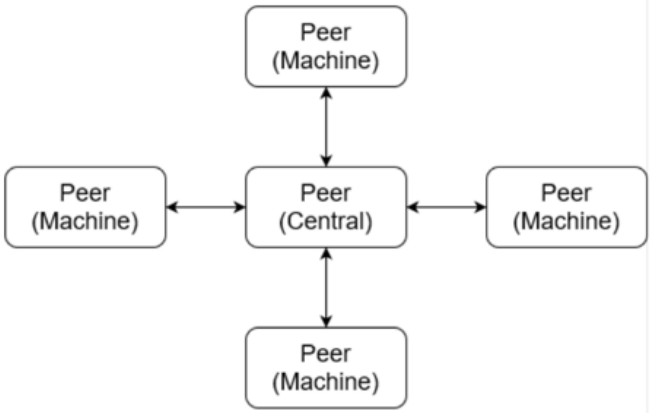
- **Dedicated client and server**
- **Processing can be done either in server or client**
  - Usually in server
- **Communication from client to server**
- **Server side can be made from multiple different components that are not required like database and data processing**
  - Divided to different tiers depending how many layers in server there are

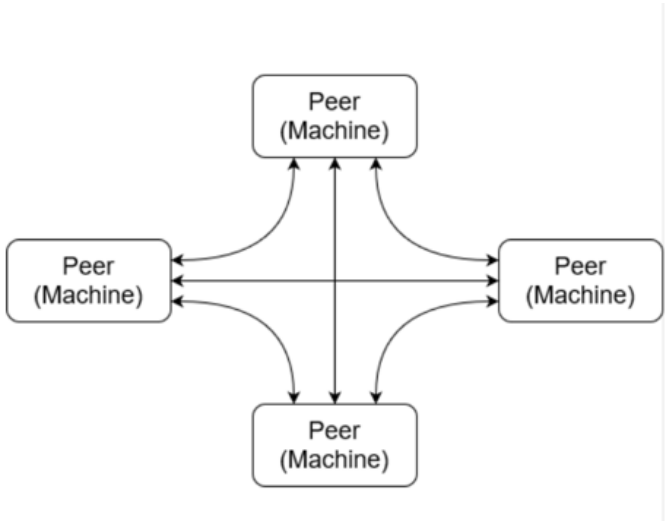


Communication Methods 2/3

Peer-to-Peer

- **All nodes act as clients and servers**
- **Adding new nodes adds processing power**
- **Pure or hybrid**
  - Hybrid combines some elements of Client-Server communication with Peer-to-Peer
- **Communication can go trough multiple nodes**

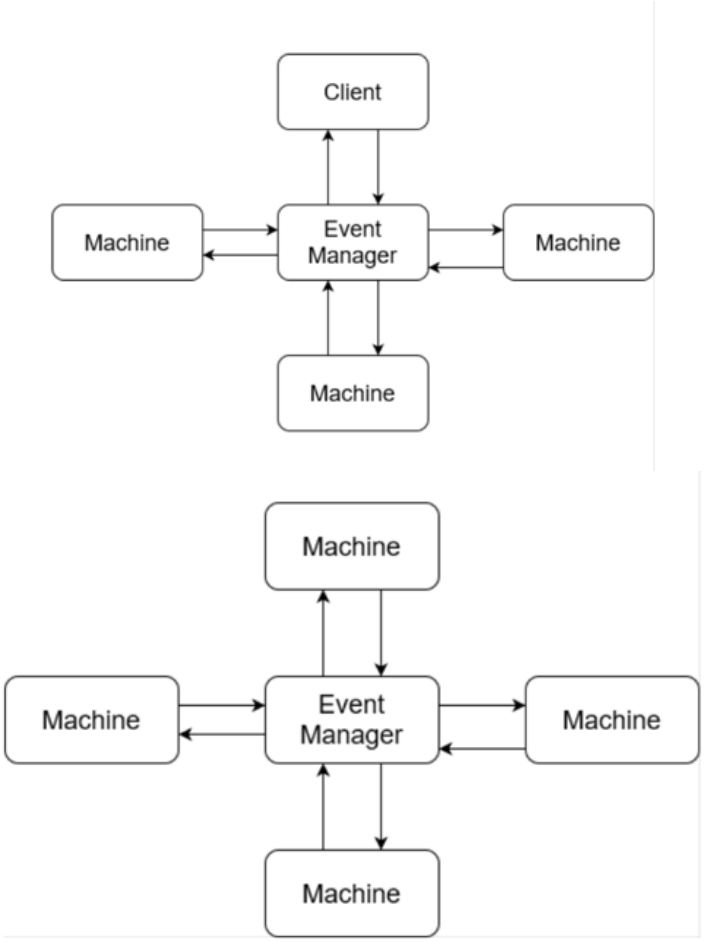




Communication Methods 3/3

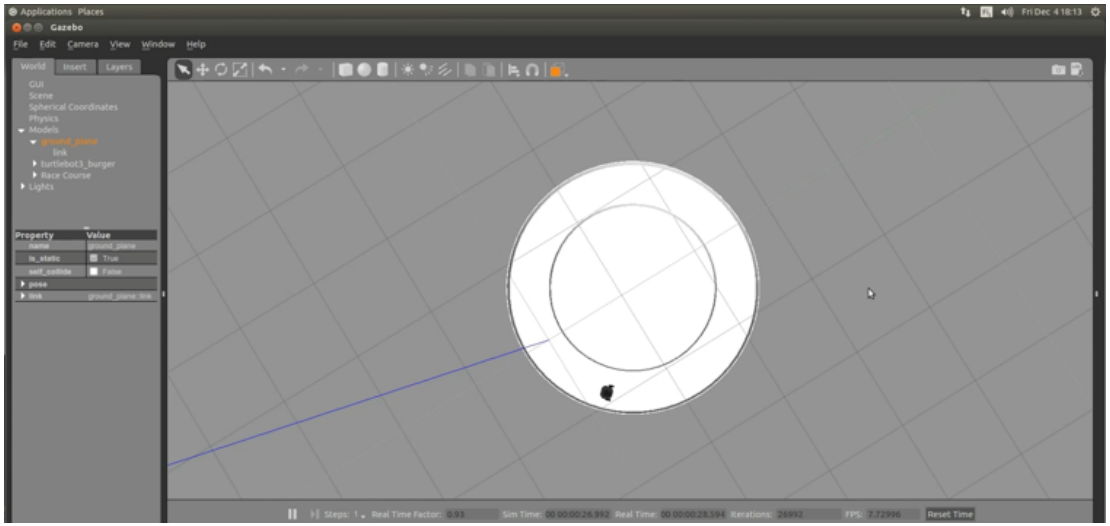
Publish-Subscribe

- **Publisher, Subscriber and Event manager**
- **Asynchronous**
- **Publishers and Subscribers anonymous to each other**
- **Subscribers select from event manager what type of message they want**
  - Topic based, Content based or hybrid
- **Confirming arrival of message needs to implemented as it is not default operation in Publish-Subscribe model**



What is ROS? And why use it?

- **Robot Operating System**
- **Framework that includes different libraries and tools for creating robotic software**
  - Allows implementing communication between nodes
  - Allows adding sensors to them
- **Easy to create and share ROS packages**
- **Open source**
  - Possibility make changes if needed
- **Both Client-Server and Publisher-Subscriber models can be implemented**
- **Lots of different modules**
- **Gazebo**
  - Simulation with sensor data etc.



Next Steps

- **Creating OPC UA servers/nodes for machines**
- **Implementing communication methods between objects**
  - Client-Server
  - Publish-Subscribe
  - Peer-to-Peer
  - Client-Server & Publish-Subscribe combination
- **Running tests and comparing bandwidth usage and how easy to implement and manage communication**
- **Using meta-level digital twins as tool for communication**