

# MAAS-Hub

Scalable homogenous architecture for multi  
agent systems

Moazzam Abdullah Khan

Daiem Nadir Ali

Nour Soufi

Segun Ajibola

# Assumptions

- Unlimited resources
  - All orders can be completed
  - No need to prioritize orders to maximize profit
- Release time
  - Orders are released with individual specified delays
- No deadlines
  - Orders don't have a deadline attached
  - No need to prioritize orders to maximize profit
- Fixed processing time

# Challenge

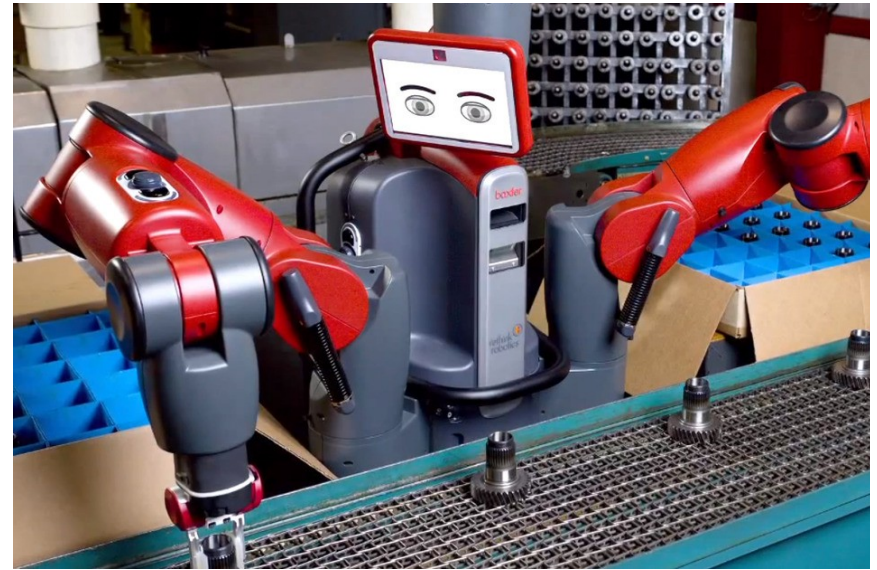
- Customer places order
  - Orders require multiple steps to complete
- Agents have to complete the order
  - Agents have specializations
    - Transporting
      - Can move
    - Painting
      - Can paint
    - Fastening
      - Can fasten two objects together
- Deliver finished order to customer

# Assumptions

- Unlimited resources
  - All orders can be completed
  - No need to prioritize orders to maximize profit
- Release time
  - Orders are released with individual specified delays
- No deadlines
  - Orders don't have a deadline attached
  - No need to prioritize orders to maximize profit
- Fixed processing time

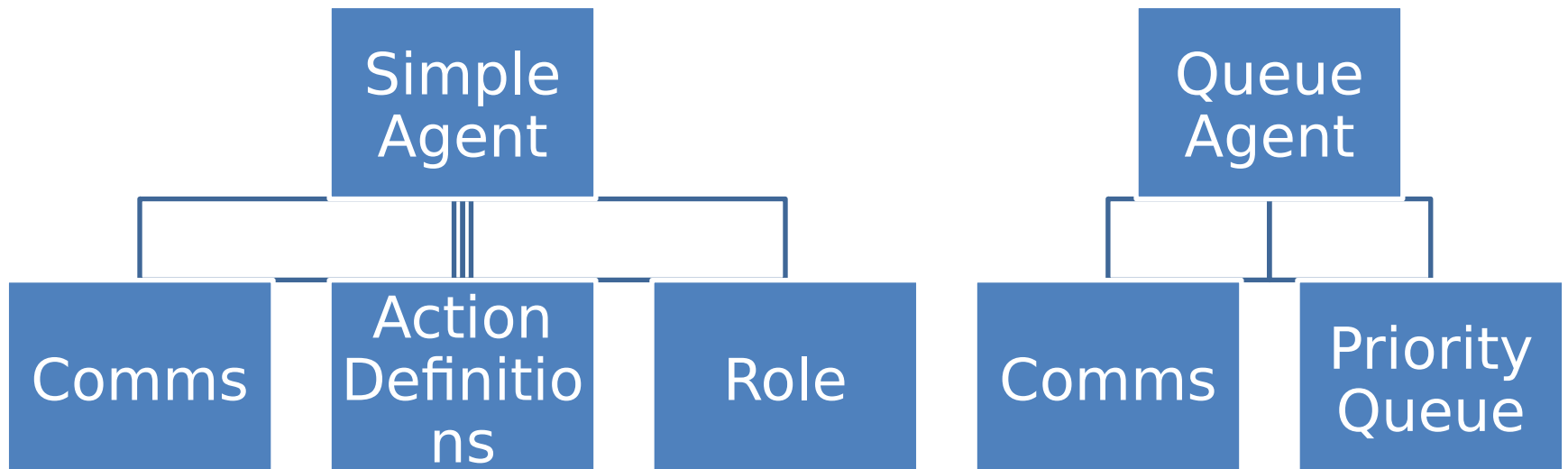
# Our Motivations

- Keep agents
  - Homogenous
  - Versatile
  - Indefinitely scalable

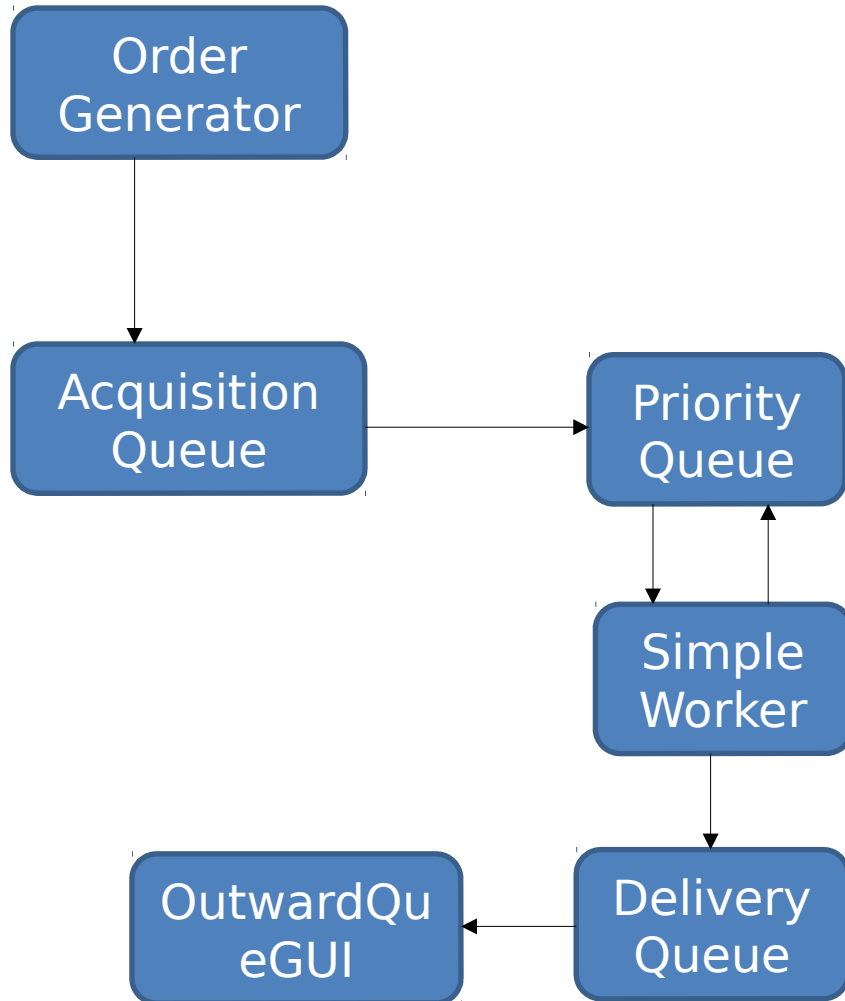


Baxter Robot: A versatile robot worker capable of doing complex tasks

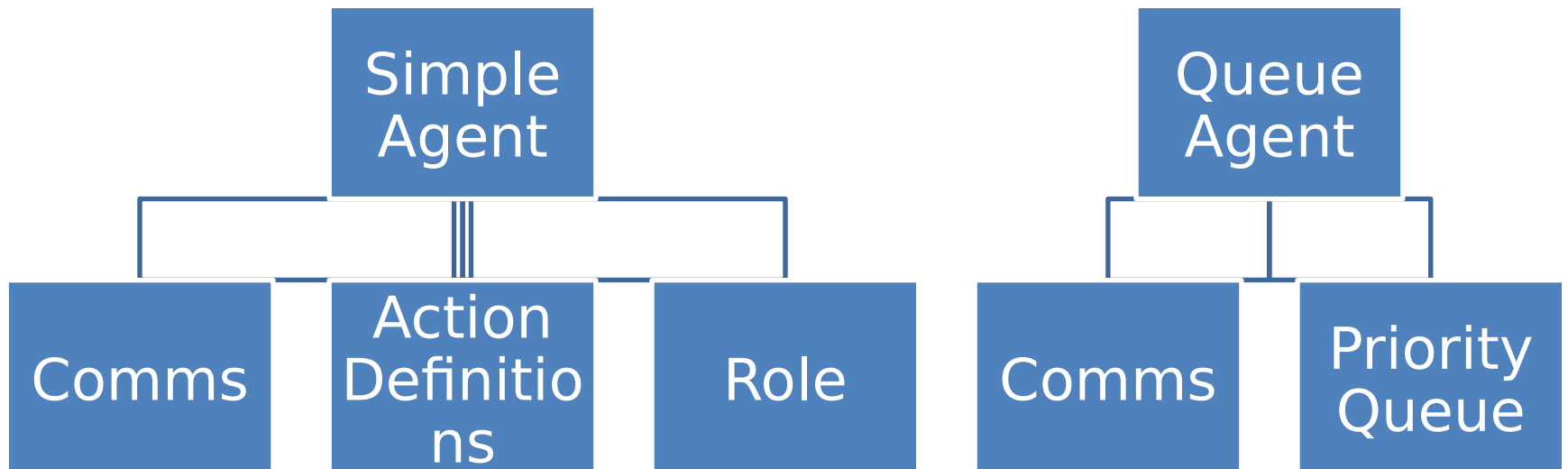
# Structure



# Communication Diagram

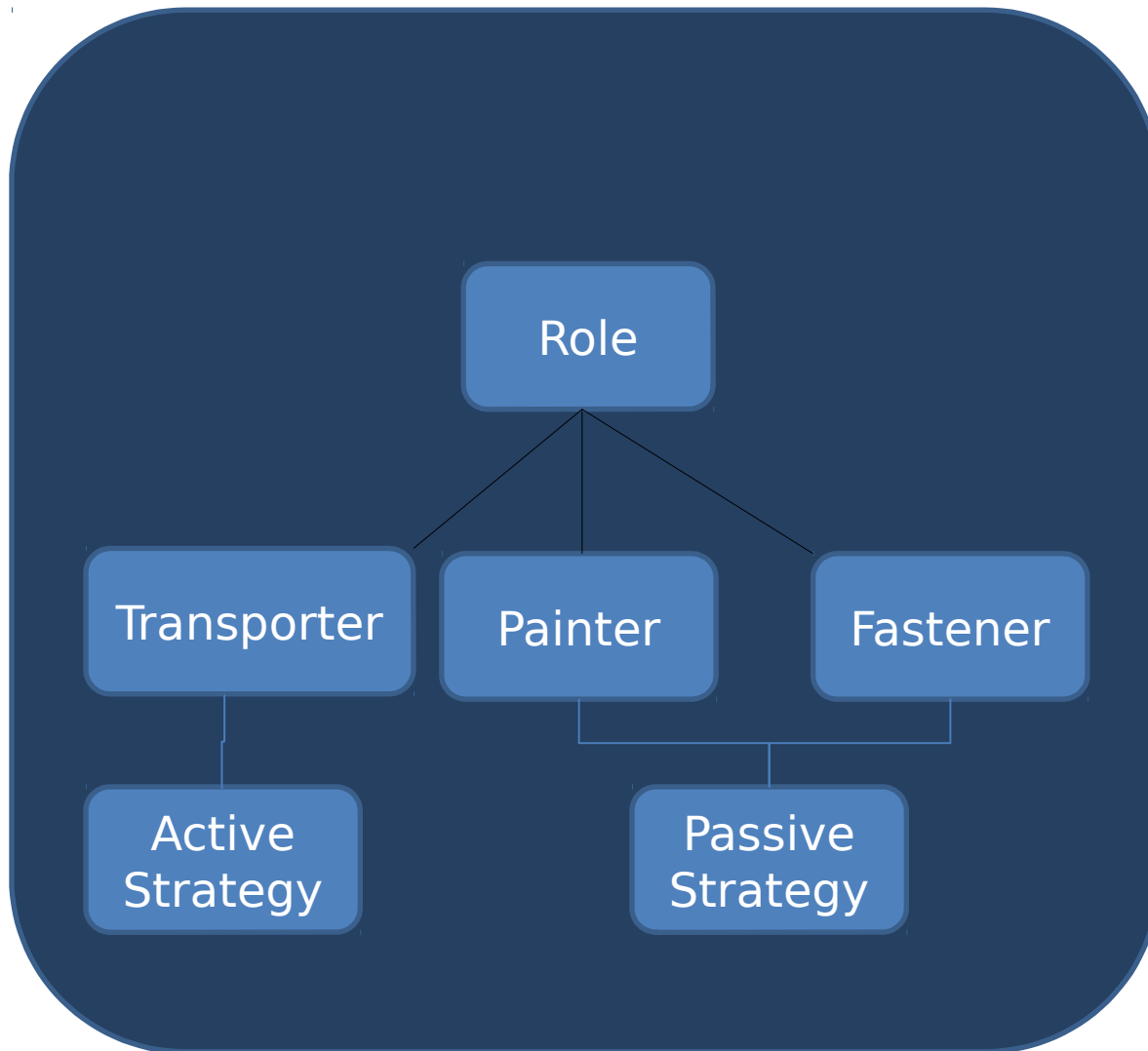


# Structure



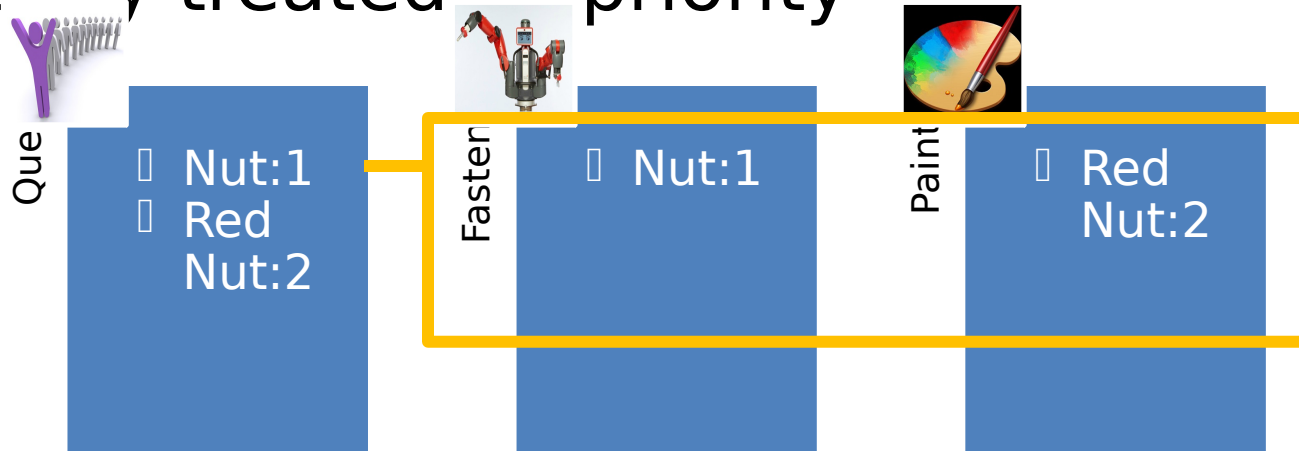


# Simple Worker



# Command and Control

- Every agent registers on yellow pages
- Queue holds information on agent inventories
- Objects have a number representing overall progress (step number) which is currently treated as priority



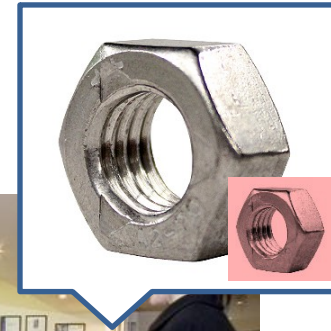
# Flow



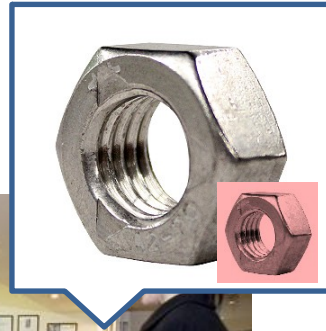
# Flow



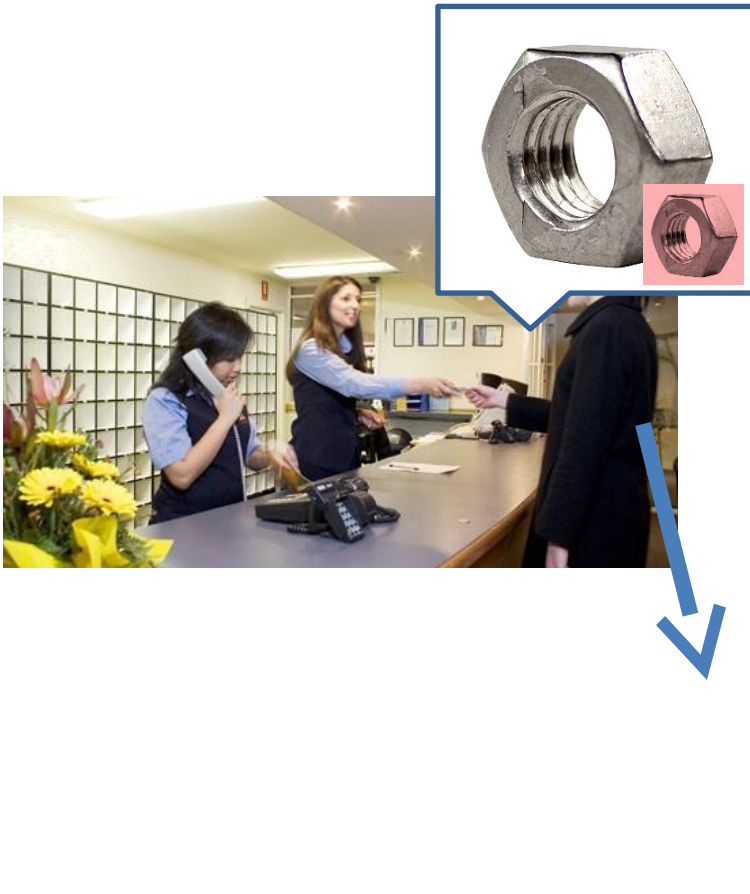
# Flow



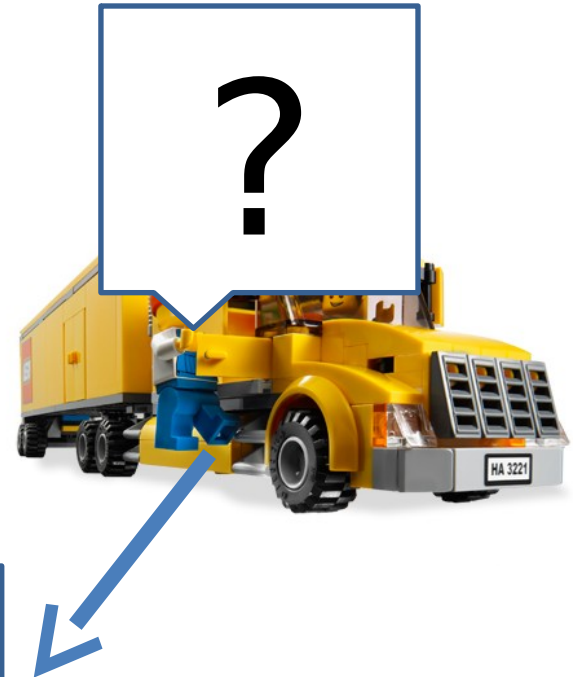
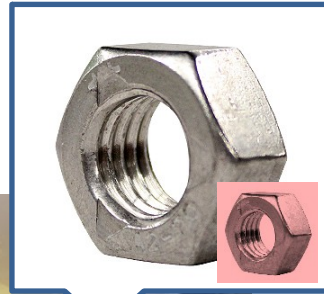
# Flow



# Flow

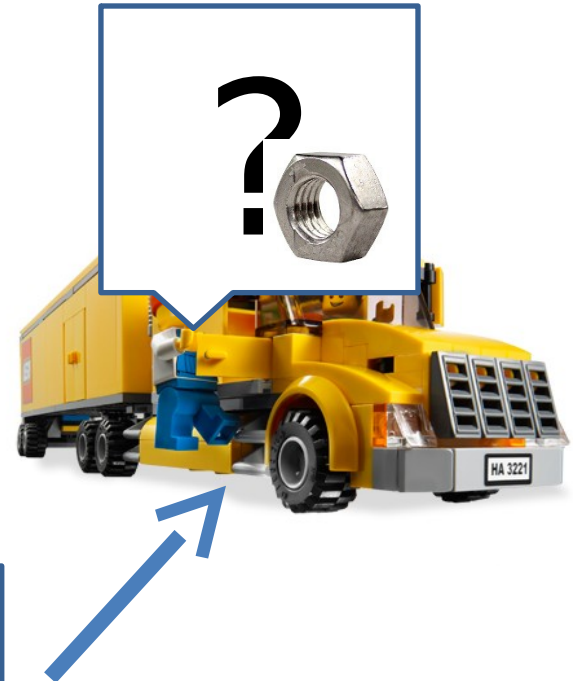
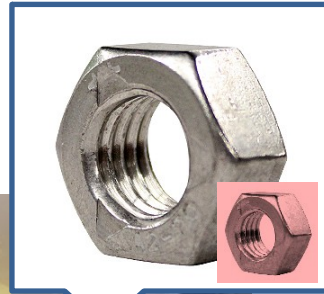


# Flow

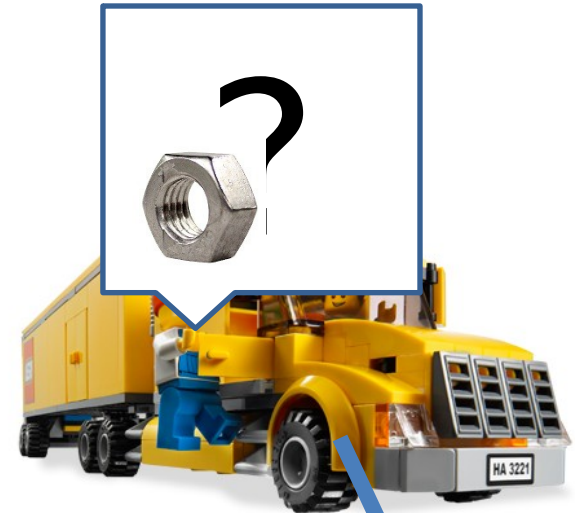
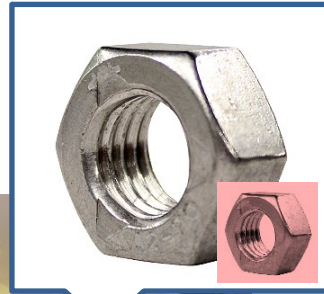




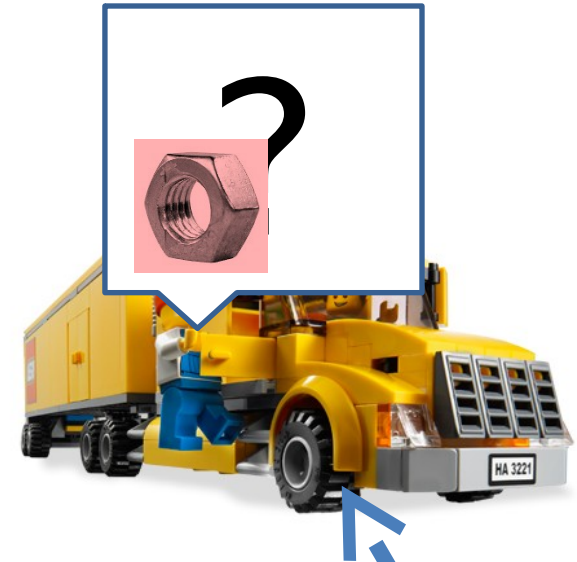
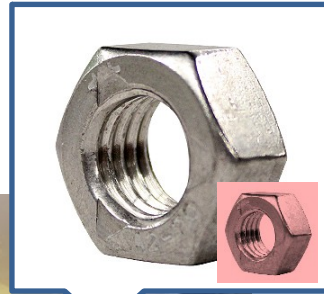
# Flow



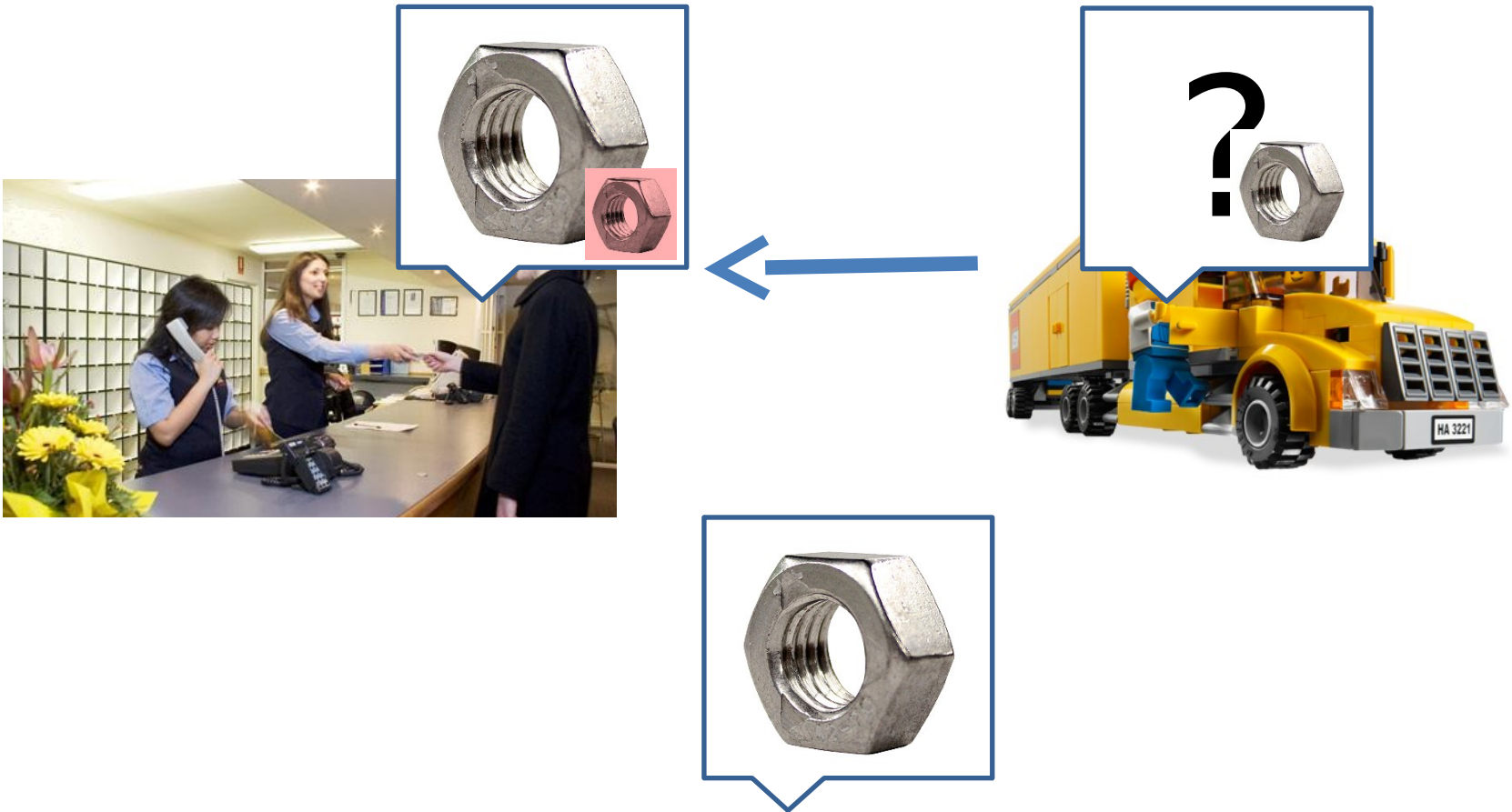
# Flow



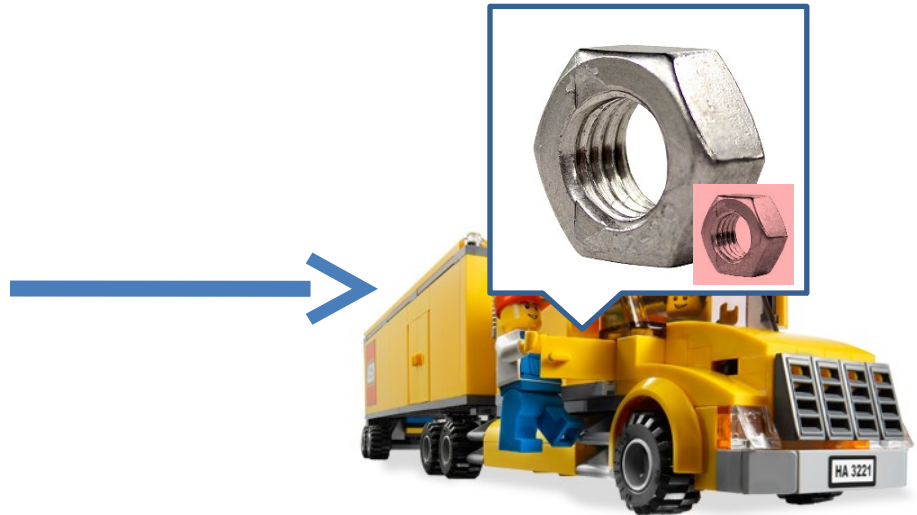
# Flow



# Flow



# Flow

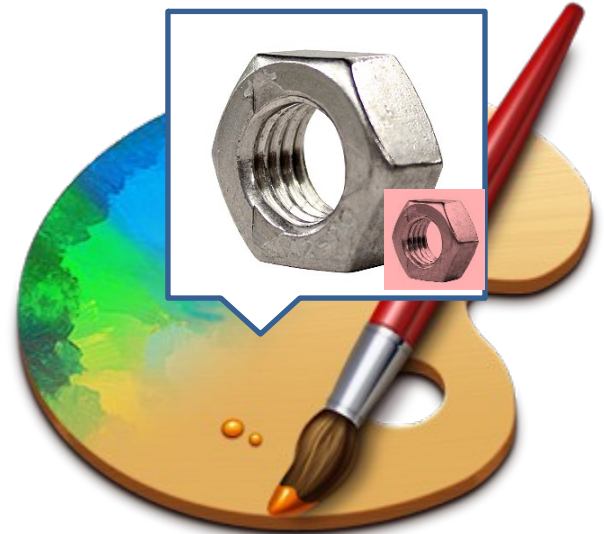




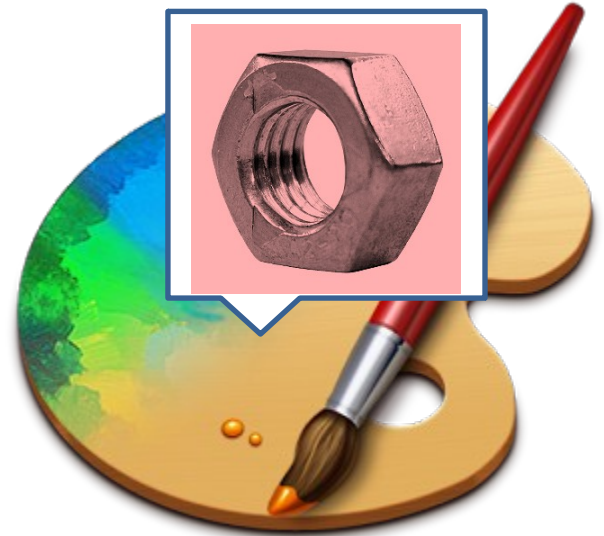
# Flow



# Flow

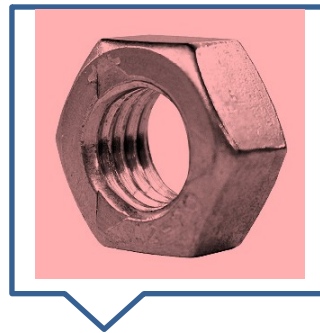
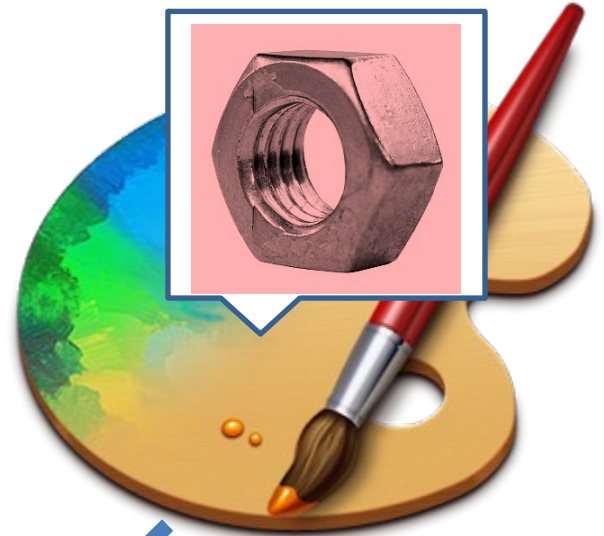


# Flow

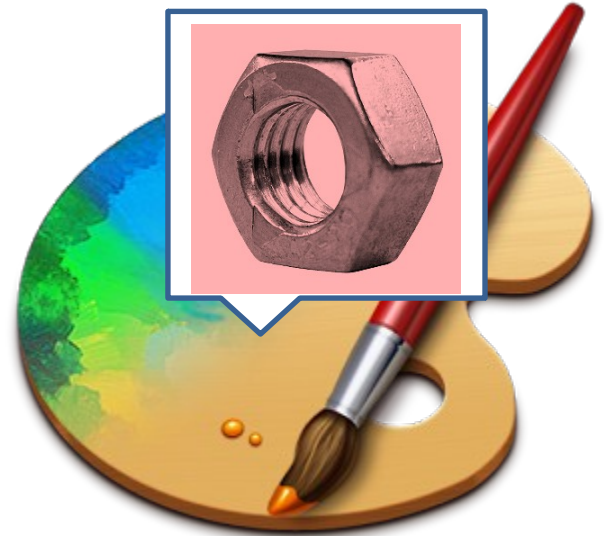
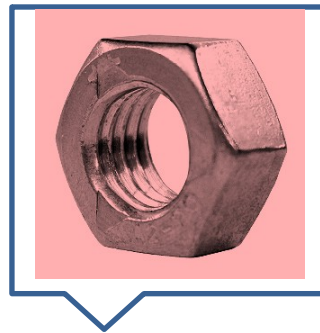




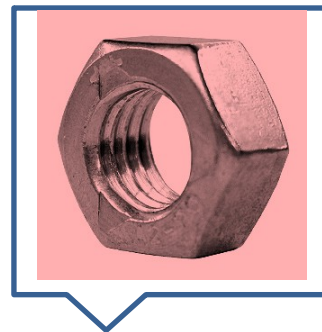
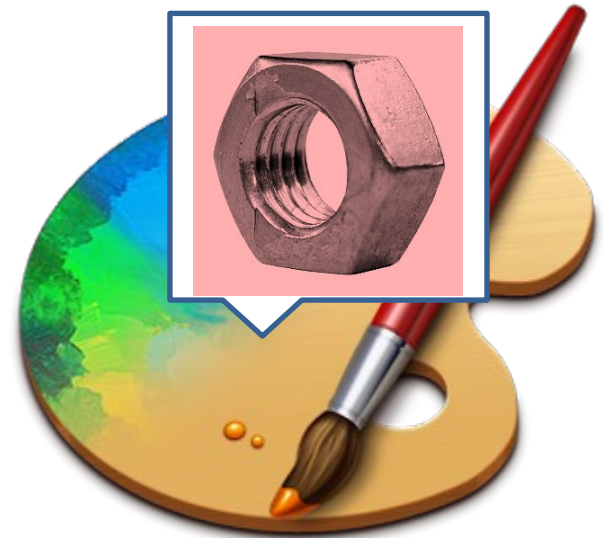
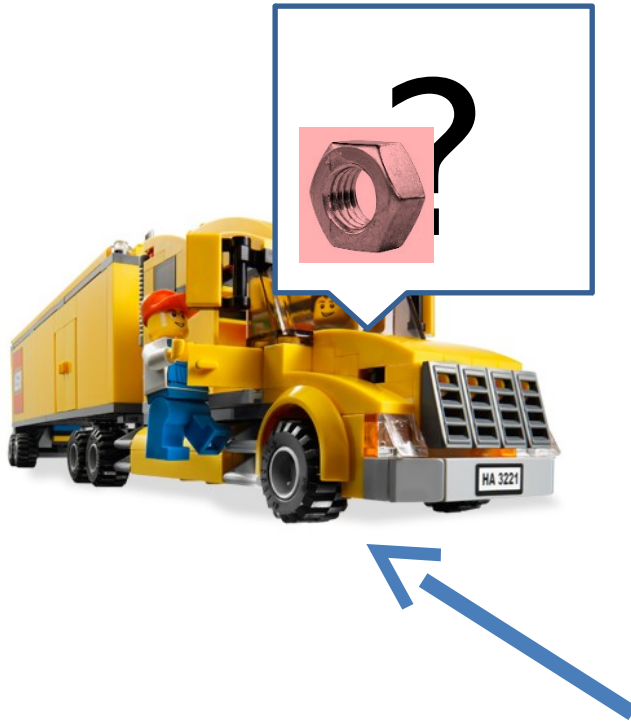
# Flow



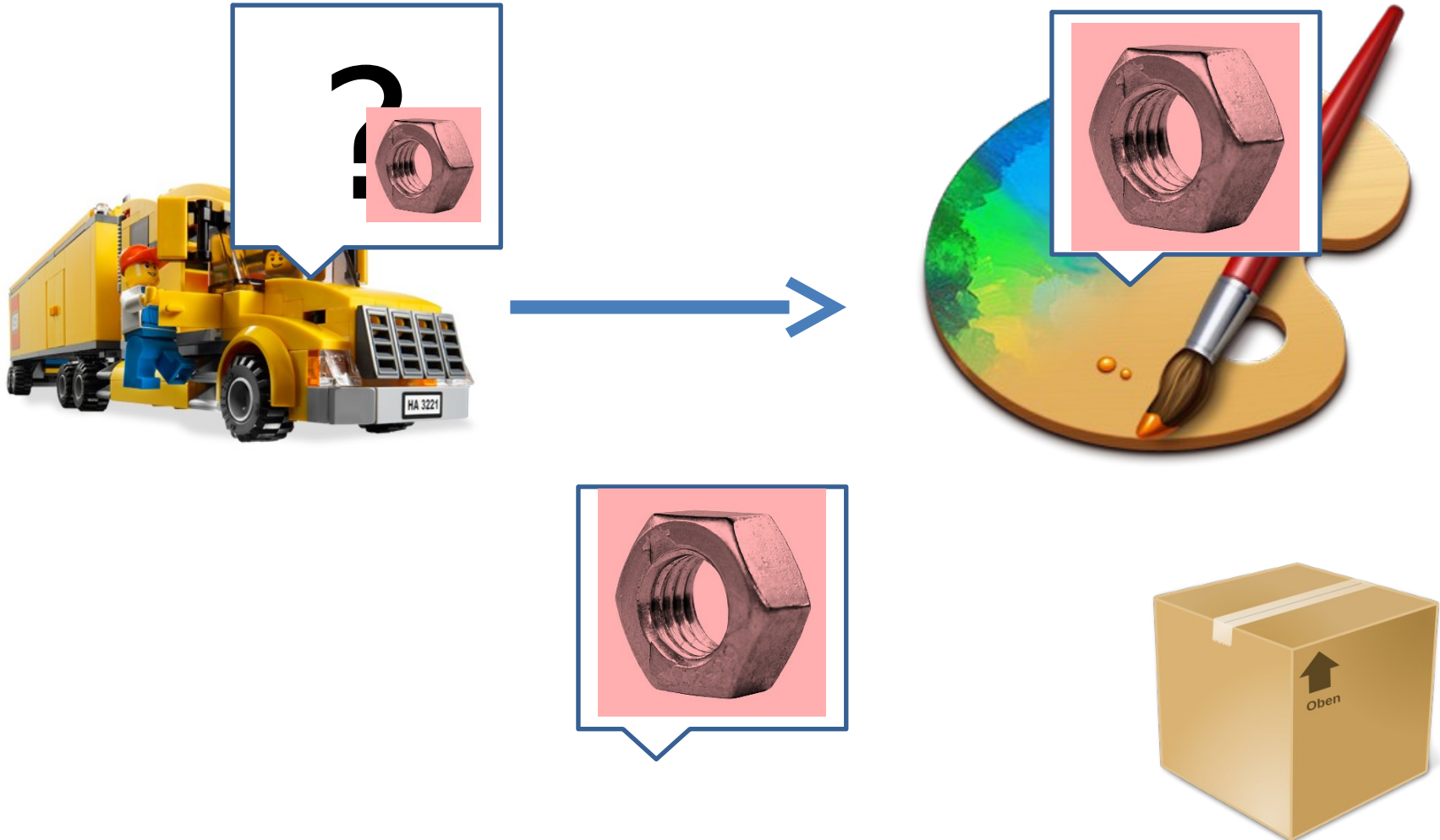
# Flow



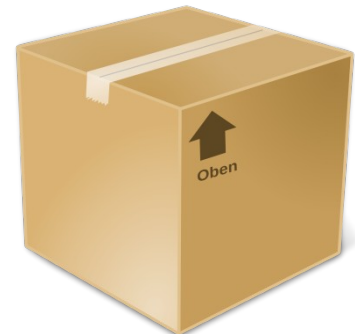
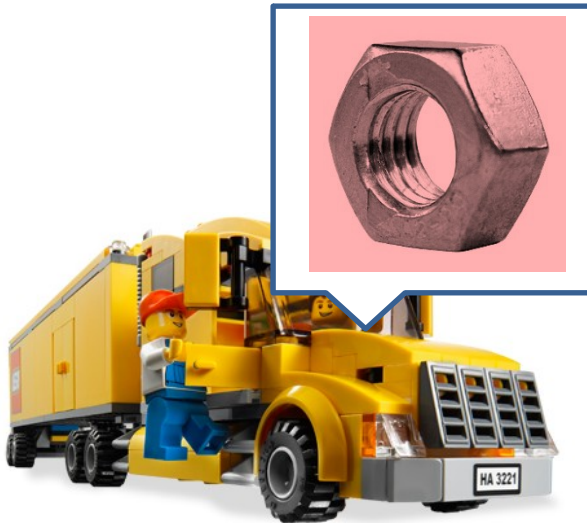
# Flow



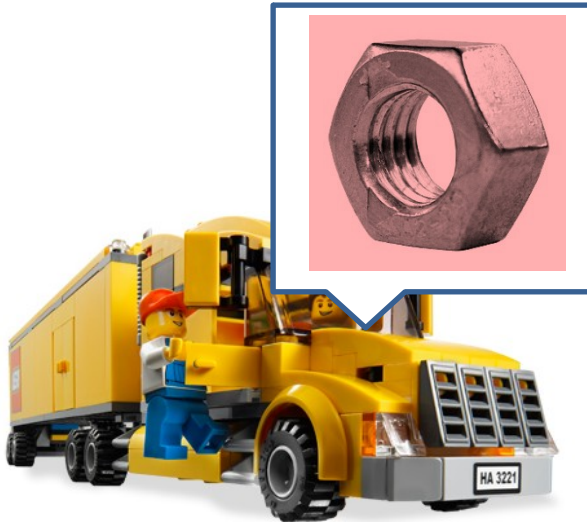
# Flow



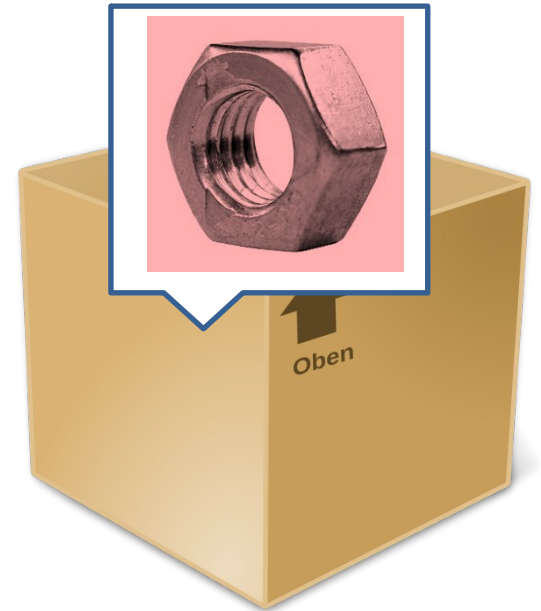
# Flow



# Flow



# Flow



# Peculiarities

- **Advantages**

- Yellow pages based communication
  - Centralized but robust
- Micro contracts
  - Contract look ahead only 1 step
    - Little bandwidth overhead
- Priority based implementation
  - Can be changed to target other priorities e.g deadline

- **Disadvantages**

- New Type of order will take considerable effort to be included in the system.
- Number of transporters define the bottle neck of the system.



# Implementation

- JADE issues
  - Minor implementation issues
- Tools used
  - Ticker behaviors
  - RE Initiators/Responder
  - Priority Queues
  - Yellow pages
  - Finite State Machines

# Lessons Learned

- Scalable architecture
- Can be used ideally in any scenario
  - Computational complexity restricts to relatively high end devices
    - ARM based devices ideal
    - Not suitable for microcontrollers
- Deadlines can be accommodated by changing priority number

