

# DRONE-2-DRONE COMMUNICATION

## CCAS & ASTA Grand Opening

- Morten Skov Jørgensen, Senior Cyber-Physical Specialist, Alexandra Institute

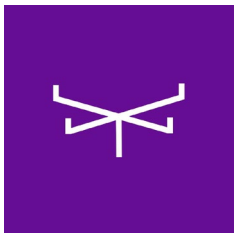
**The Alexandra Institute helps you create innovative solutions based on pioneering IT research.**

**Our mission is to stimulate growth and social prosperity.**



# Secure Drone-to-Drone Communication

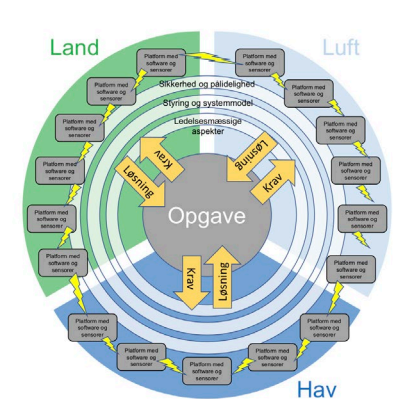
- Rambøll
  - Meili Robots
  - DTU Space
  - Alexandra Institute
- 
- Funded by InnoSec





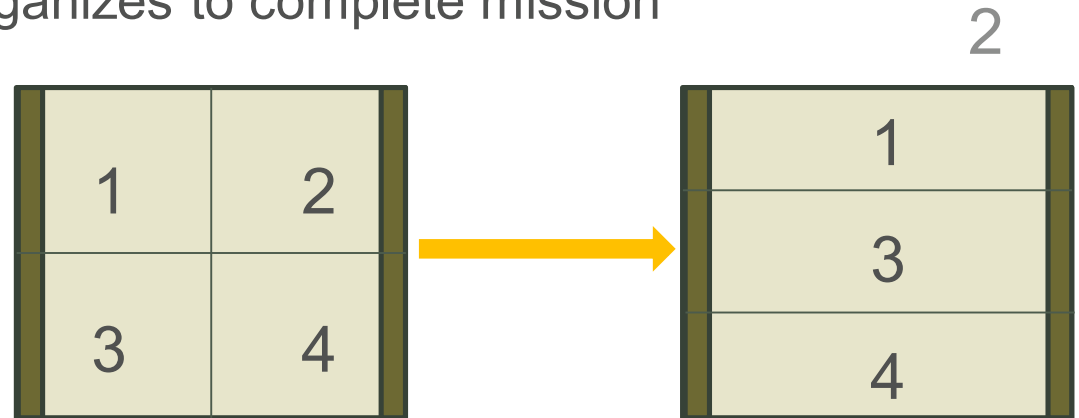
# Description

- Not a catch-all, but a possibility
- Encrypted swarm communication for collective hivemind
- Based on well-known principles from the P2P world
  - Hardened from the software pirate world
  - Catalan independence referendum
  - Wikipedia mirror



# Usage - organization

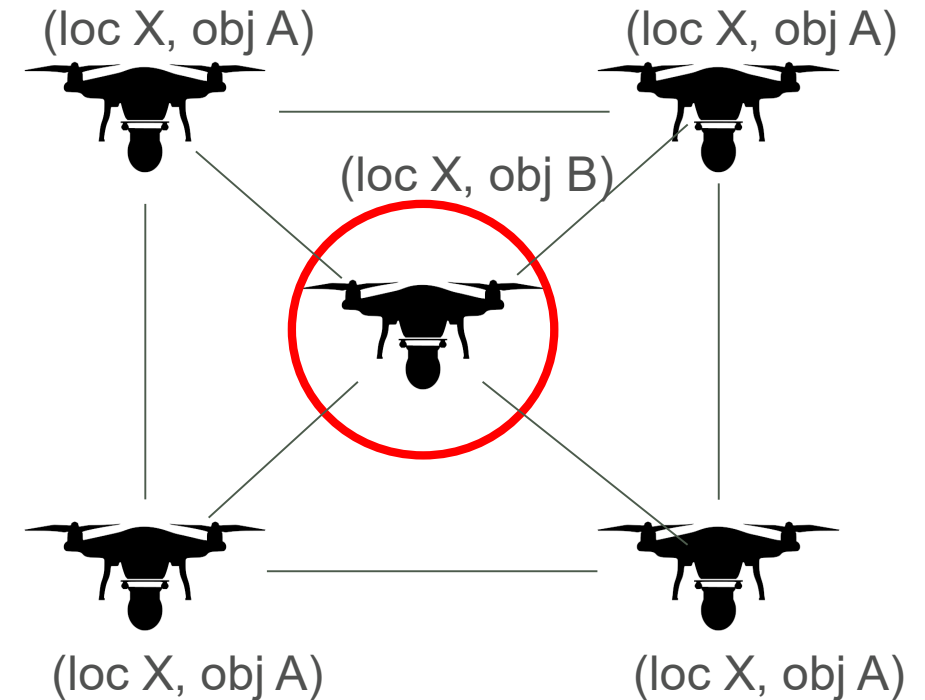
- Enhance resilience in case of break-down
  - If C2 unit loses connection, the mission can continue
  - If an agent breaks down, swarm re-organizes to complete mission



- Utilizes different features of agents for optimizing efficiency
  - If an agent has special hardware, data can be piped to said unit
  - Self-organising based on agent features
  - Minimizes latency for real-time actions

# Usage - context

- All agents know about each other
  - Inverse: unknown units are possible adversaries
  - Bad actors can use almost undetectable units
- Any sensor can be placed in the network
  - RGB camera
  - Thermal
  - Radio
  - Etc



# Technology

- Golang
- Libp2p
  - Open Source
  - Support for multiple languages
  - Maintained by IPFS community
  - Used by Microsoft, Cloudflare, Brave
- ROS
- Three.js
- Unity



# Technology

- Libp2p
  - TCP-based
  - Private/public key
  - Pubsub
  - Gossip Protocol
- Three.js
  - Debugging
- Unity
  - Visualization
  - Simulation

```
func ini var agentType model.AgentType ) *model.Agent {
    var agentsMux = &sync.Mutex{}
    agent var reorgMux = &sync.Mutex{} sSim, isCtrl)
    if * agentType AgentType
        agentType = model.ControllerAgent
        log.Println("This agent is started as a controller")
        missionaryId = &agent.ID
        SetController(&agent)
        MyMission = model.GetMission()
    } else {
        log.Println("This agent is started as a context unit")
        agentType = model.ContextAgent
        log.Printf("agent.Nick* is starting at position: %v \n", agent.Position)
    }

    log.Println("Init communication with agentType: ", agentType)
    comm.InitD2DCommunication(agentType)

    log.Println("Start registration on path: " + discoveryPath)
    //comm.InitRegistration(discoveryPath)
    comm.InitCommunicationType(discoveryPath, comm.DiscoveryMessageType)

    log.Println("Start state on path: " + statePath)
    comm.InitCommunicationType(statePath, comm.StateMessageType)

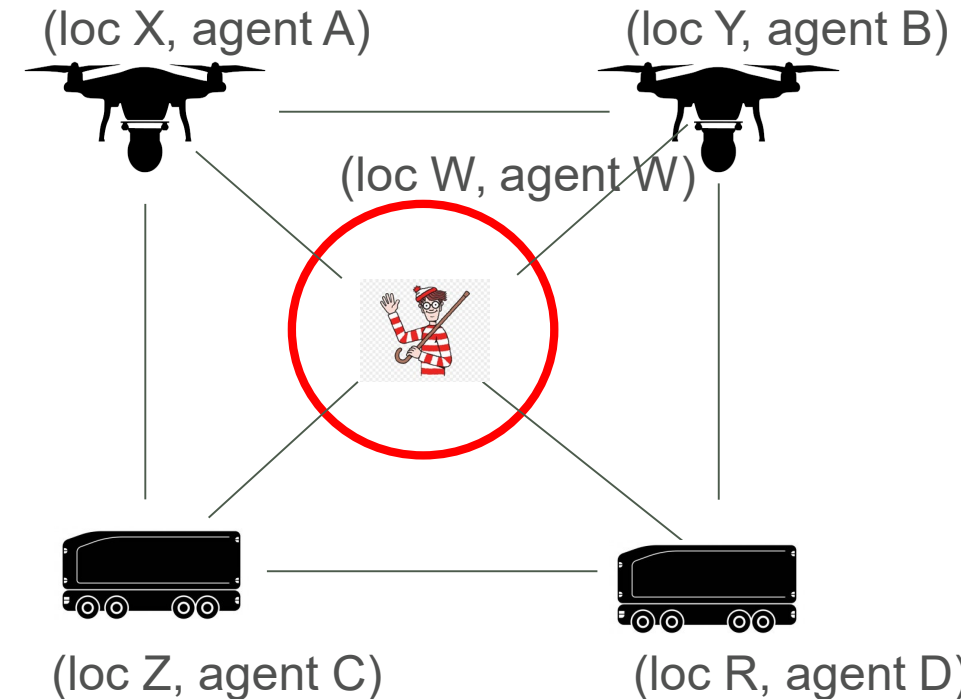
    if *UseViz {
        log.Println("This agent sends visualization data")
        comm.InitVisualizationMessages(false)
    }
    agent.ID = comm.SelfId.Pretty()
    return &agent
}

func startDiscoveryWork() {
    go func() {
        log.Println("Waiting to find companions")
        for {
            msg := <-comm.DiscoveryChannel
            agentId := &msg.DiscoveryContent.ID
            //log.Println(msg)
            agentsMux.Lock()
            ok := agents[*agentId]
            agentsMux.Unlock()
            if ok {
                //agent already known
            } else {
                if msg.SenderType == model.ControllerAgent {
                    if agentType == model.ContextAgent && !HasCtrl {
                        //TODO: check that this is a controller, we trust
                        log.Println("Found a controller - handing over mission privileges")
                        ControllerDiscoveryChannel <- &msg.DiscoveryContent
                        comm.InitCommunicationType(MySelf.ID, comm.MissionMessageType)
                        missionaryId = &msg.DiscoveryContent.ID
                    } else if agentType == model.ControllerAgent {
                        log.Println("Weird, another controller in swarm...we should panic!")
                        log.Println(msg)
                    }
                }
            }

            log.Printf("Found a buddy with nick: %s - adding to list \n", msg.DiscoveryContent.Nick)
            //check to see if the agent was believed to be dead
            if _, ok := lostAgents[*agentId]; ok {
                //we thought it was dead - remove it from watchlist
                reorgMux.Lock()
                delete(lostAgents, *agentId)
                reorgMux.Unlock()
            }
        }
    }
}
```

# Example – Where's Waldo

- Find Waldo in urban environment
- Modality: flying, driving
- Human Detection ML in swarm
  - REST endpoint for modularity
- Flow:
  - Agent B breaks down
  - Swarm agrees
  - Reorganization handled by chosen agent
  - New specific mission sent to all agents





# Theme day – 24/11-20

- The ASTA facility at DTU
- Physical and virtual demo of protocol
- Talks and discussions about the technology

# Thank you for your time

- Morten Skov Jørgensen
- Senior Cyber-Physical Specialist



Morten.skov@Alexandra.dk



+45 21 13 57 07



ALEXANDRA  
INSTITUTE