11th International Workshop on Engineering Multi-Agent Systems (AAMAS)

Remote Deployment of a JADE Agent in Docker

Dennis Maecker · Henning Gösling · Oliver Thomas

Department for Smart Enterprise Engineering (Osnabrück, Germany)
German Research Center for Artificial Intelligence

German Research Center for Artificial Intelligence

GAIA-X 4 ROMS

Funded by the European Union
NextGenerationEU

Supported by:
Federal Ministry for Economic Affairs and Climate Action
on the basis of a decision by the German Bundestag

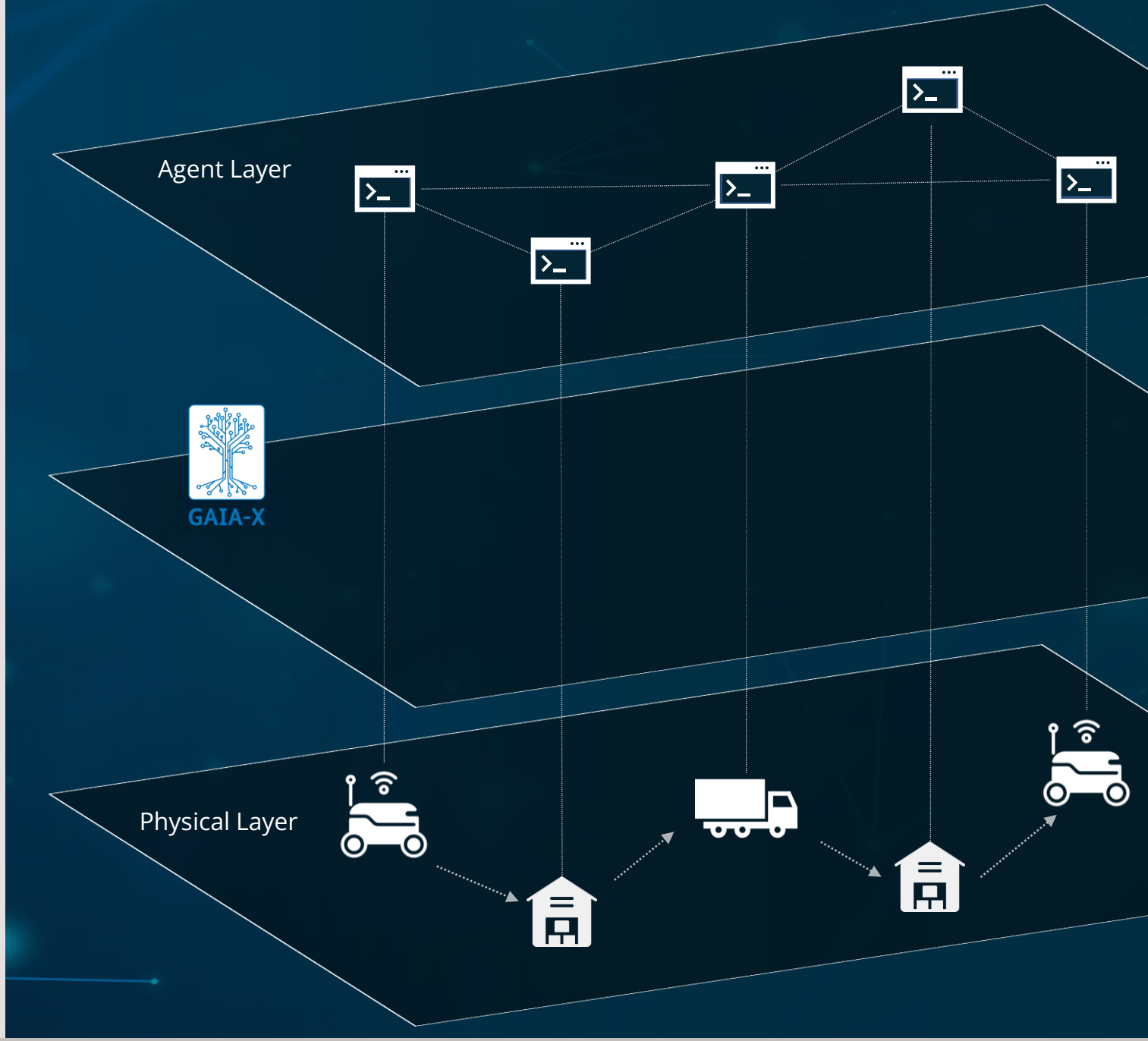
Context

Project GAIA-X 4 ROMS:

Solution for a smart managed freight fleet for parcel delivery within a self-sovereign GAIA-X data ecosystem

Our contribution: Development of a multi-agent system (MAS) for managing fleet assets

A decentralized approach necessitates interoperable, remotely deployed agents

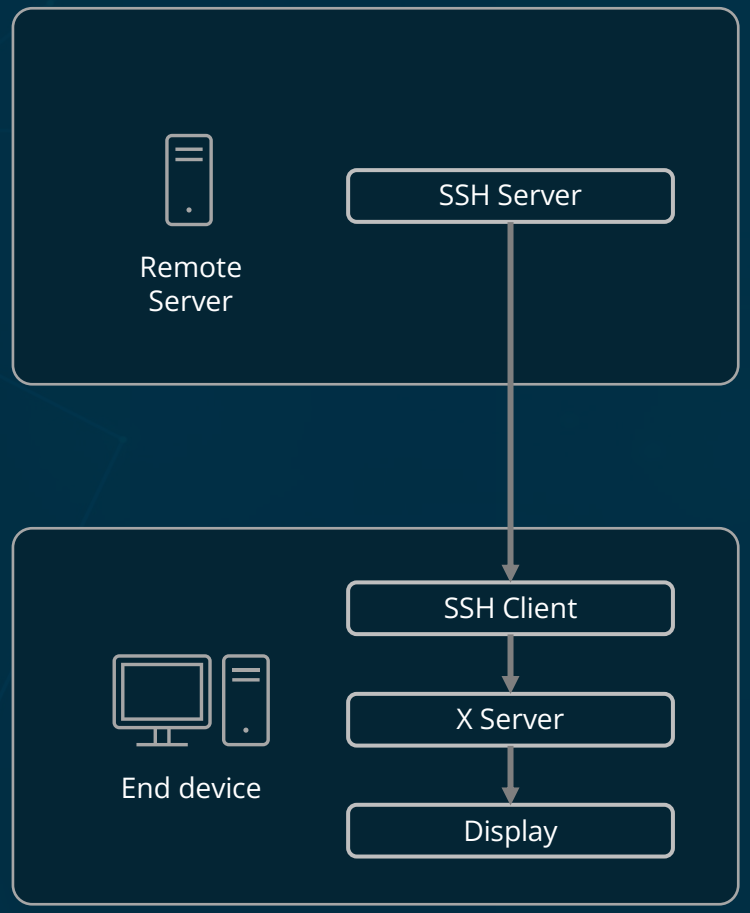


Our Approach

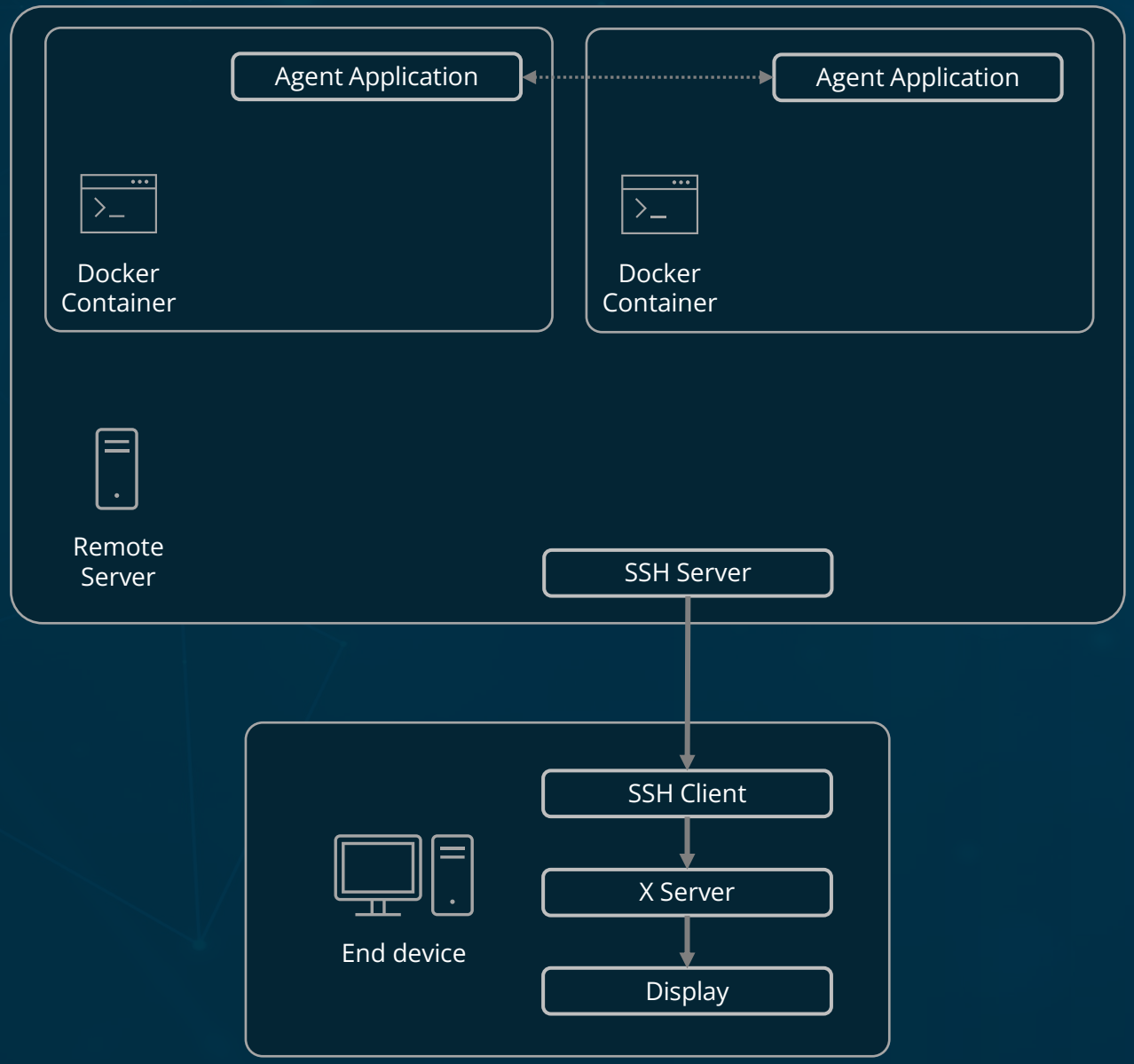
- Java Agent DEvelopment Framework (JADE) for agent development
- Containerization using Docker for facilitated deployment
- Usage of a graphical user interface (GUI) for managing the MAS and the behavior of each agent

Problem: No solution for deploying a JADE-based MAS in Docker remotely while allowing the access of the GUI from an end device.

Solution

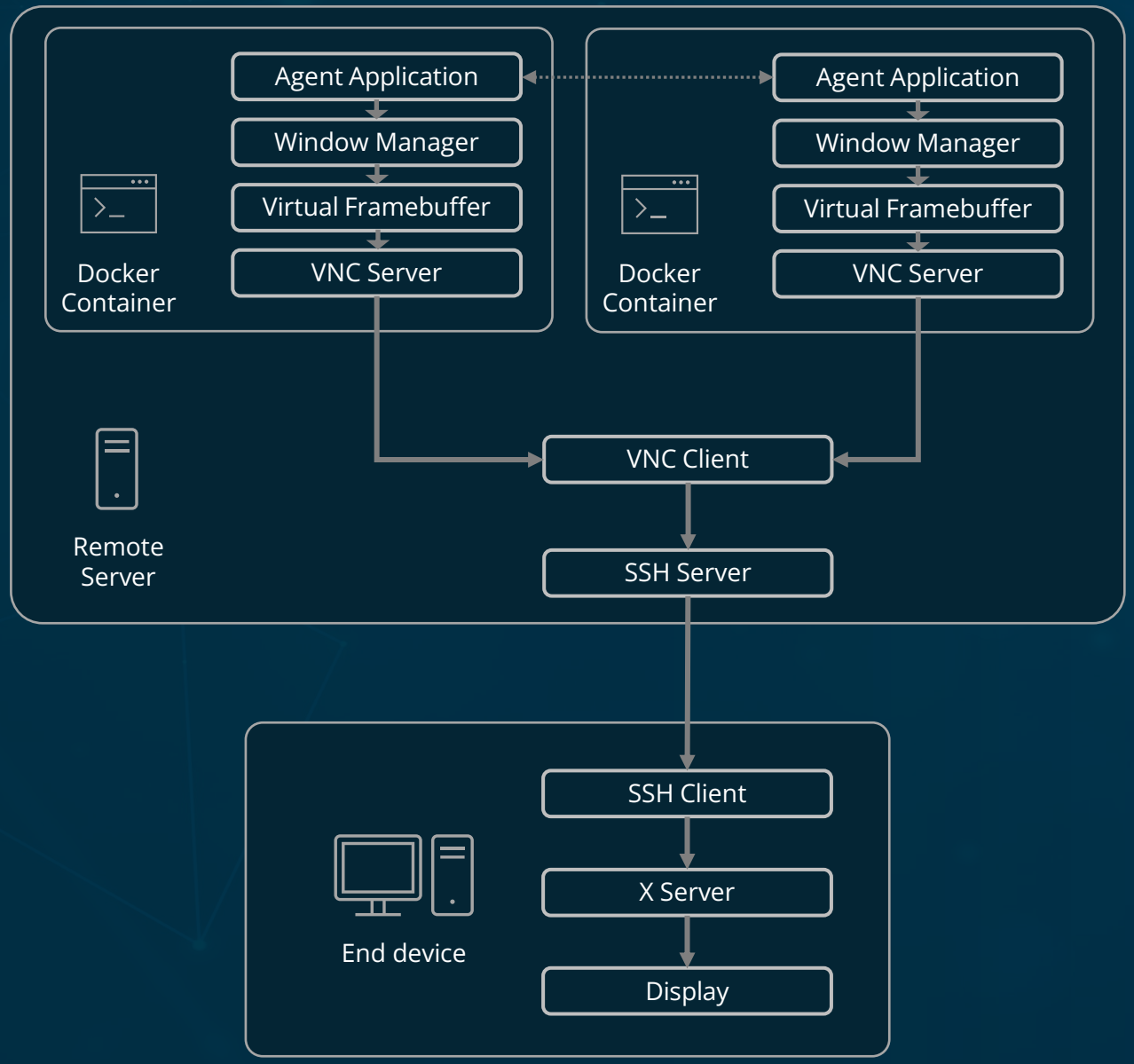


Forwarding of graphical data from the remote server to the end device via SSH



Solution

Forwarding of graphical data from the remote server to the end device via SSH

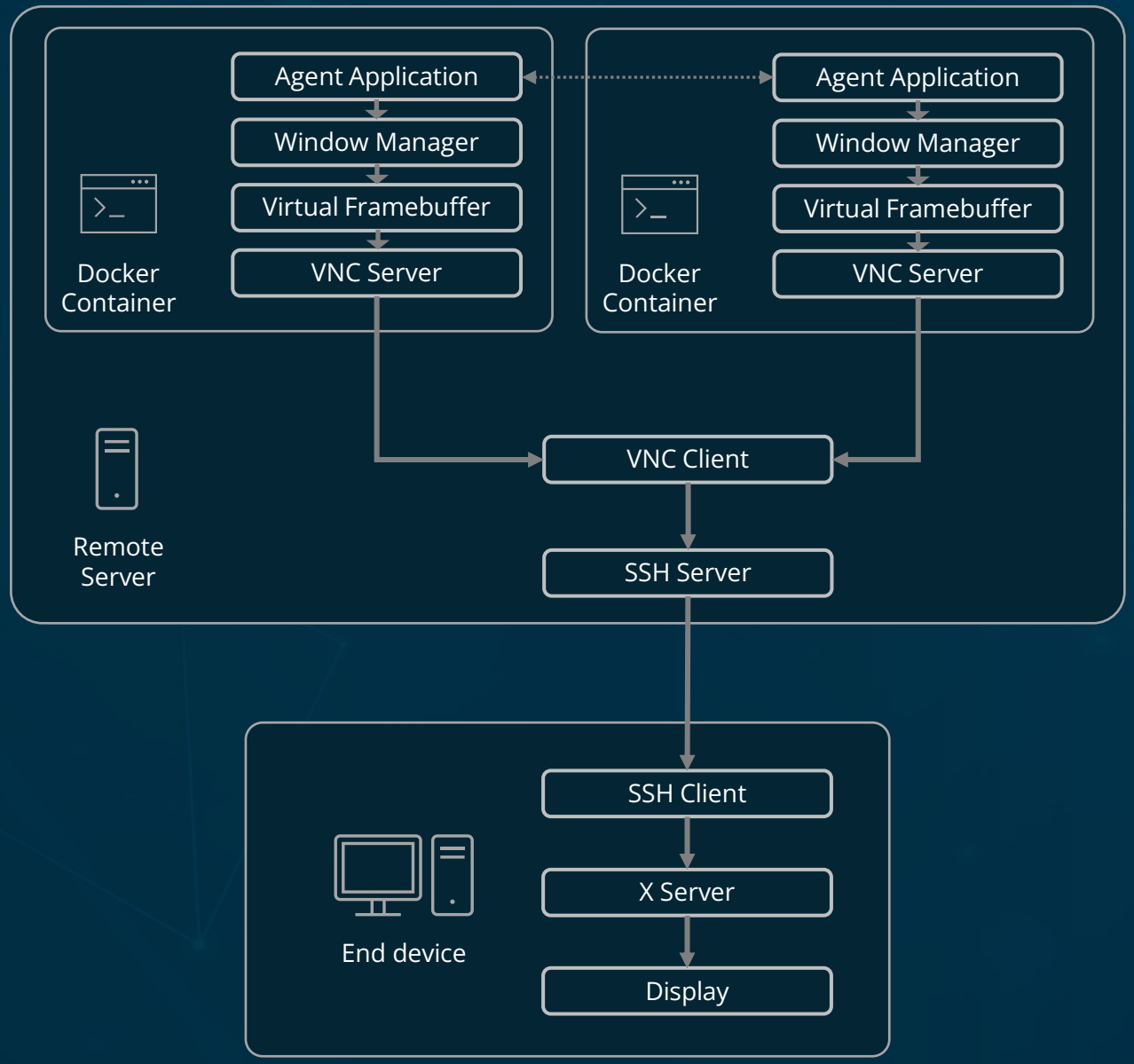


Solution

Forwarding of graphical data from the remote server to the end device via SSH

Setup of a graphical environment within the Docker containers

Installing a VNC server in the Docker container enabled accessing the agent GUI by a VNC client on the remote server



Solution

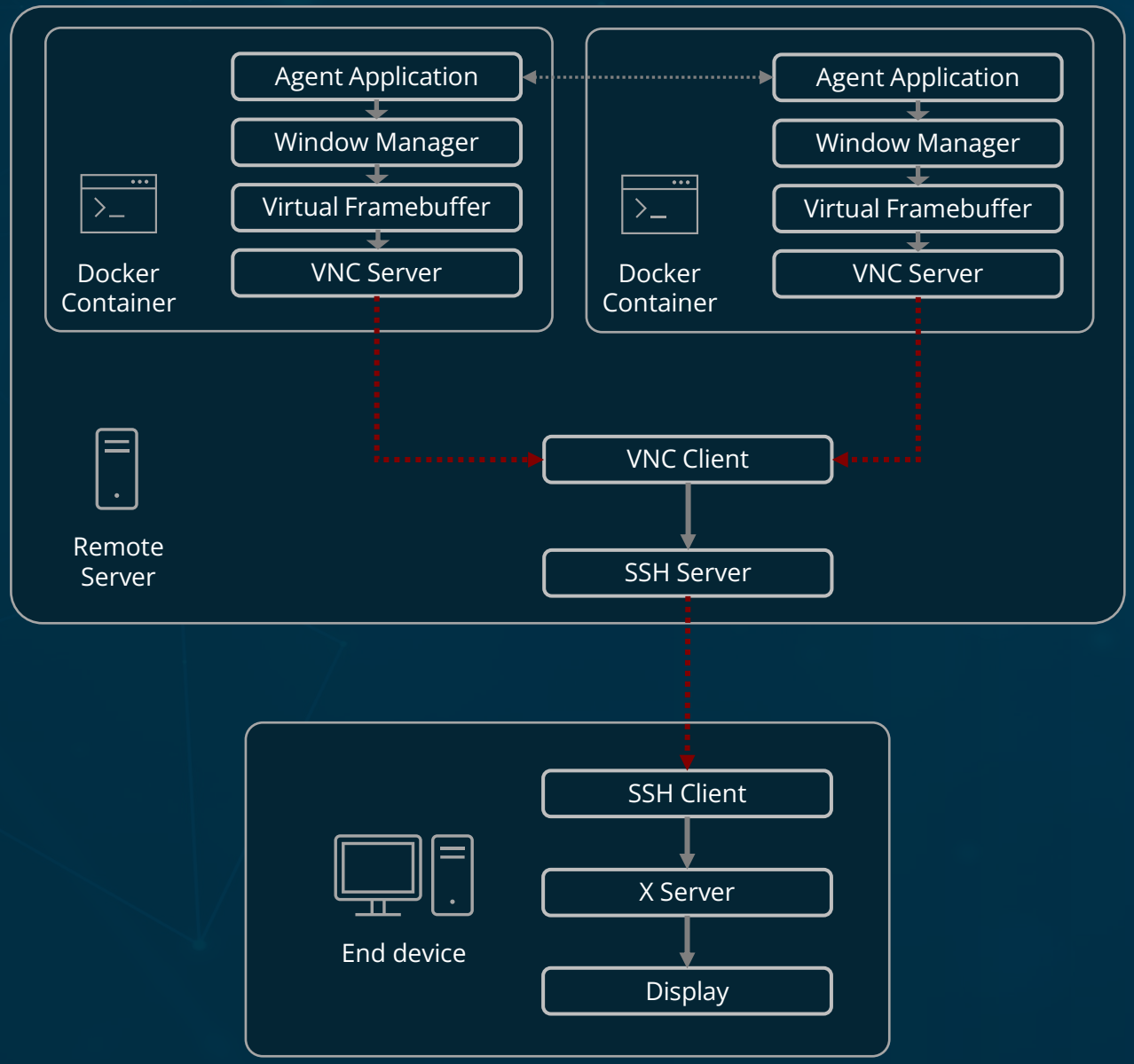
Forwarding of graphical data from the remote server to the end device via SSH

Setup of a graphical environment within the Docker containers

Installing a VNC server in the Docker container enabled accessing the agent GUI by a VNC client on the remote server

Advantages:

- Only one data connection (SSH)
- Agent applications stay active when SSH or VNC connections are disconnected



Solution

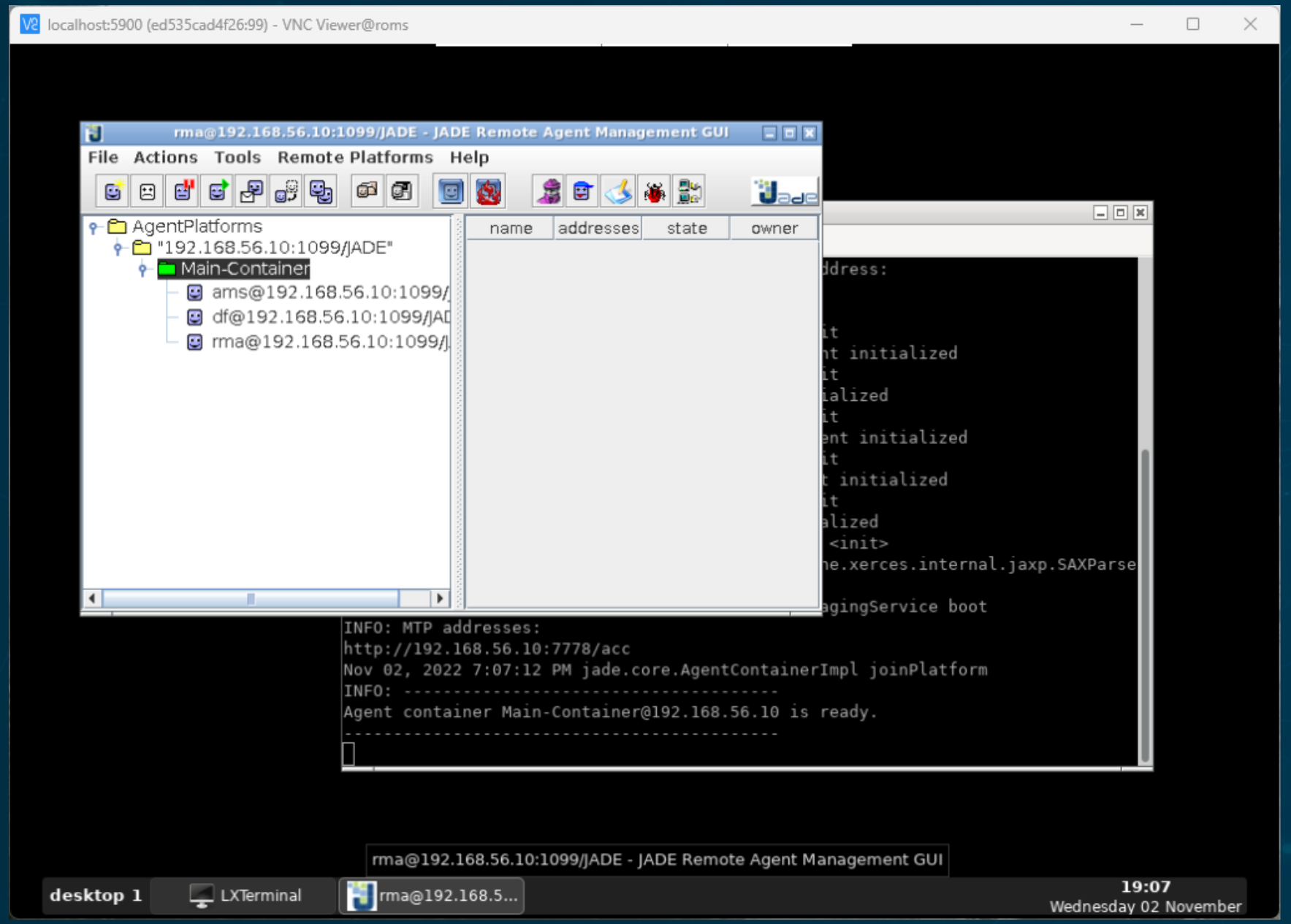
Forwarding of graphical data from the remote server to the end device via SSH

Setup of a graphical environment within the Docker containers

Installing a VNC server in the Docker container enabled accessing the agent GUI by a VNC client on the remote server

Advantages:

- Only one data connection (SSH)
- Agent applications stay active when SSH or VNC connections are disconnected


localhost:5900 (ed535cad4f26:99) - VNC Viewer@roms
rma@192.168.56.10:1099/JADE - JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help
AgentPlatforms
"192.168.56.10:1099/JADE"
Main-Container
ams@192.168.56.10:1099/
df@192.168.56.10:1099/JA
rma@192.168.56.10:1099/
name addresses state owner
INFO: MTP addresses:
http://192.168.56.10:7778/acc
Nov 02, 2022 7:07:12 PM jade.core.AgentContainerImpl joinPlatform
INFO: ----------
Agent container Main-Container@192.168.56.10 is ready.
rma@192.168.56.10:1099/JADE - JADE Remote Agent Management GUI
desktop 1
LXTerminal
rma@192.168.5...
19:07
Wednesday 02 November

Conclusions and Outlook

Successfull deployment of a JADE-based MAS on any remote server/device possible

Docker facilitates the compatibility with different systems

JADE extension “MTP HTTP“ enables the formation of a MAS distributed over several servers

Possible integration into a GAIA-X ecosystem via HTTP endpoints

Transferable to other frameworks under consideration (SPADE, ROS)

References

1. Anderson, C.: Docker [software engineering]. IEEE Software 32(3), 102–c3 (2015)

2. Bellifemine, F., Caire, G., Trucco, T., Rimassa, G., Mungenast, R.: Jade administrator's guide. TILab (2003)

3. Bellifemine, F., Poggi, A., Rimassa, G.: JADE - A FIPA-compliant agent framework, pp. 97–108. The Practical Application Company Ltd. (1999)

4. Greenwood, D., Bellifemine, F.L., Caire, G.: Developing Multi-Agent Systems with JADE. WILEY (2007)

5. Heinbach, C., Gösling, H., Meier, P., Thomas, O.: Smart managed freight fleet: Ein automatisiertes und vernetztes flottenmanagement in einem föderierten datenökosystem. HMD Praxis der Wirtschaftsinformatik (2022)

6. murer: Virtual x and vnc server docker image with openbox. https://github.com/murer/docker-xvfb-x11vnc-openbox (2020)

7. Tatham, S.: PuTTY User Manual (2022), https://upload.wikimedia.org/wikipedia/commons/b/b7/PuTTY_User_Manual.pdf

8. Ughetti, M., Trucco, T., Gotta, D.: Development of agent-based, peer-to-peer mobile applications on ANDROID with JADE. In: The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. IEEE (2008)

Code

```
1: Manifest-Version: 1.0
2: Main-Class: smartTransport.Container
```

(a) Content of the `MANIFEST.MF` file.

```
1: FROM openjdk:18-jdk-slim
2: COPY ./out/artifacts/SmartTransport/
   /home/
3: RUN apt-get update && apt-get -y install
   iproute2 x11-apps libxi6 libxtst6
   libxrender1 lib32z1 xvfb openbox tint2
   x11vnc xterm lxterminal
4: ADD config /opt/config
5: RUN chmod +x
   /opt/config/docker-entrypoint.sh
   /opt/config/openbox/autostart
6: RUN rm -rf /etc/xdg/openbox && cp -R
   /opt/config/openbox /etc/xdg/openbox &&
   (rm -rf /etc/xdg/tint2 || true) && cp -R
   /opt/config/tint2 /etc/xdg/tint2
7: CMD "/opt/config/docker-entrypoint.sh"
```

(b) `Dockerfile` for containerization of the multi-agent system intended to be deployed on a remote server.

```
1: #!/bin/bash -xe
2: xvfb-run -s "$DISPLAY" -s '-screen 0
   1024x700x24 -ac' openbox-session
```

(c) `docker-entrypoint.sh` file.

```
1: version: "2.4"
2: networks:
3:  agent_network:
4:  driver: bridge
5:  ipam:
6:   driver: default
7:   config:
8:   - subnet: 192.168.56.0/24
9:   gateway: 192.168.56.1
10: services:
11:  container:
12:   image: dfki/agents
13:   networks:
14:    agent_network:
15:     ipv4_address: 192.168.56.10
16:     ports:
17:     - 1099:1099
18:     - 7778:7778
19:     - 5900:5900
20:   working_dir: /home
21:   environment:
22:    DISPLAY: :99
23:  agent1:
24:   image: dfki/agents
25:   networks:
26:    agent_network:
27:     ipv4_address: 192.168.56.11
28:     ports:
29:     - 1100:1099
30:     - 7779:7778
31:     - 5901:5900
32:   working_dir: /home
33:   environment:
34:    DISPLAY: :99
```

(d) `docker-compose.yml`-file used for launching one JADE main-container and one JADE agent container in dedicated Docker containers.