

# UMBRELLA Collaborative Robotics Testbed and IoT Platform

Tim Farnham<sup>†</sup>, Simon Jones<sup>‡</sup>, Adnan Aijaz<sup>†</sup>, Yichao Jin<sup>†</sup>, Ioannis Mavromatis<sup>†</sup>, Usman Raza<sup>†</sup>, Anthony Portelli<sup>†</sup>, Aleksandar Stanoev<sup>†</sup>, Mahesh Sooriyabandara<sup>†</sup>

<sup>†</sup>Bristol Research and Innovation lab. Toshiba Europe Ltd. Bristol, U.K.

<sup>‡</sup>University of Bristol, U.K.

{tim.farnham, adnan.aijaz, yichao.jin, ioannis.mavromatis, usman.raza, anthony.portelli, aleksandar.stanoev, mahesh}@toshiba-bril.com; simon2.jones@bristol.ac.uk

**Abstract**—This paper provides details of the collaborative robotics testbed platform that has been developed within the UMBRELLA project. The testbed is part of a larger open Industrial IoT testbed which is currently being deployed in the UK. The aim of the testbed is to permit flexible experimentation using different end devices (including smart city sensing and robot nodes) to evaluate algorithms or new practical application scenarios. For the collaborative robotics testbed this relates to warehouse robotics, which can move pallets of different sizes and shapes. The testbed also includes simulator facilities for validation of algorithms prior to deployment on the robot nodes. The nodes support a rich set of sensors, actuators, and wireless communication technologies. The software architecture is based on Docker containers and ROS2 DDS middleware for flexible and extensible evolution to support future sensors or network technologies. This will be provided as an open testbed to support research, experimentation and evaluation of swarm robotics and other Industrial IoT use-cases.

**Keywords**—Swarm robotics, testbed, IoT platform

## I. INTRODUCTION

Recently, robotics platforms are becoming openly available to the research community, providing remotely accessible swarm robotics platforms, freely available by anyone. There are a number of testbeds available, such as Robotarium [3] and IRIS [5], that support collaborative robotics experimentation. These existing testbeds are largely used for general swarm algorithm research and evaluation. The Urban Multi Wireless Broadband and IoT Testing for Local Authority and Industrial Applications (UMBRELLA) [2] project will provide access to a vast number of different testbeds, that can be used for various use-cases. One of the main platforms provided is a collaborative robotics testbed used for practical industrial swarm robotics applications, such as warehouse robotics. In this way, the evaluation of performance can be more specific and addresses the needs of the use-case, which are tasks that survey, stock-take, monitor and move objects from place to place within a mock warehouse environment. In addition, the UMBRELLA robot nodes have a very rich set of sensors and actuators, which include cameras and various wireless network technologies.

Due to its unique capabilities, the UMBRELLA collaborative robotics testbed can be used for evaluating versatile use-cases, algorithms and operational scenarios. For instance, collective and coordinated transport of objects in warehousing and logistics applications using swarm intelligence and behavior-based algorithms. By using minimal infrastructure wireless connectivity solutions and formation synthesis techniques, leveraging multi-radio and

multi-sensory information for multi-robot formation control [4], the reliance on expensive fixed infrastructure is minimised. This can greatly reduce the cost of such systems. The testbed supports the evaluation of both relative and absolute localisation techniques through multi-sensory information fusion. Also, performance of wireless mesh protocols under mobility and coverage optimisation techniques can be evaluated in ad-hoc mobile robotic networks through combine robot and radio simulation. This can be extended with 5G-centric use-cases (when private 5G [10] support is included in the future). Other use-cases that can be tested include collaborative surveying and environment mapping, remote operation assisted multi-robot systems and functional safety aspects of cooperative robotics.

The UMBRELLA robotics testbed capability is demonstrated using a “collaborative warehouse storage solution” use-case which exploits multiple robots collectively moving pallets containing objects. The performance can be measured in terms of the time taken to store / retrieve the pallets and the resources, including battery energy used. The pallets can be of various sizes and weights which necessitate the use of different numbers of robot nodes as shown in Fig. 1. Swarm algorithms aim to optimise the efficiency and time taken for moving the pallets by forming cooperative groups that collectively move the pallets. It is also necessary to account for robot energy consumption and battery charge states.



Fig. 1. Vision illustration of the Swarm Testbed for handling pallets with various size and shape.

We aim that the UMBRELLA Collaborative Robotics Testbed would serve as a melting pot where researchers and practitioners from different communities including robotics, wireless communication technologies, localisation systems, and IoT platforms, bring their expertise together. Several of these systems are tested in isolation. For example, it is a common practice to benchmark wireless systems for their reliability, latency, coverage and energy consumption and localisation systems for their accuracy, range and position update frequency. UMBRELLA robotics testbed, on the other hand, would enable researchers to know whether the numbers acquired through different benchmarking exercises do

translate into delivering robust applications specifically in warehouse robotics domain. UMBRELLA robotics testbed would serve as a true proving ground for testing Systems-of-Systems (SoS).

The paper introduces the testbed architecture in the section II followed by the IoT platform details and radio protocol stack support. The overall testbed features are compared with prior testbeds in section V.

## II. ARCHITECTURE

The focus of this paper is primarily on the communication and IoT platform aspects of the testbed. Therefore, the main architectural features and interfaces are introduced in this section, followed by the communication interfaces and software architecture.

### A. Communication reference architecture

The testbed communication reference architecture (see Fig.3) is based on the IEC 30141 IoT reference architecture [1], which is adopted by many industry segments for clear separation and securing of domains and interfaces between them. The domains have different responsibilities, and are:

- **Operations and Management:** for overall administration and asset management of the testbed resources, such as adding or removing nodes and handling software, network and node configuration updates.
- **Application Services:** for supporting the user experiment use-cases within the backend, such as storing data and experiment containers and controlling their deployment using the Harbor container registry. Also, Gazebo [6] and radio simulators for off-line experiment evaluation.
- **Access and Interchange:** for permitting secure interaction between external applications and users with the testbed resources through API and identity management, which is based on WSO2 and Keycloak respectively.
- **Sensing and Controlling:** for performing and evaluating swarm experiments that are deployed within containers. We also refer to this as the local robot domain.

The domains are interconnected via different networks, which are:

- **User network:** This network connects entities in the User domain with the Application Services and Access & Interchange system. This is performed over the Internet.
- **Service network:** This network can include both Internet elements and also (private) intranet elements. It supports the administration access to the Operations and Management as well as Application service domains.
- **Access network:** Access networks are wide area networks connecting the IoT gateways with the backend and use a VPN connection over the Internet.
- **Proximity network:** This network connects the robot nodes with each other and to the IoT gateway. It is also used for exchange of ground truth telemetry data. In the physical testbed it is supported over radios, but in the simulator environment is a virtual LAN.

### B. Features

The main features of the testbed architecture are outlined in this section.

#### 1) Dockerised container experimentation

The platform supports a scalable and flexible way of deploying experiments using Docker containers. Some of the key benefits introduced are the: 1) simplicity and faster reconfiguration, 2) isolation and segregation of the applications, 3) security of the nodes, and 4) compatibility and maintainability. A user can containerise an experiment and upload it on the UMBRELLA platform via our portal. This approach ensures that the same experiment can be evaluated on both our physical or simulated nodes, without the need for reconfigurations.

The experiments are deployed on the physical or virtual nodes using a lightweight Kubernetes framework, based on K3S. Such a strategy facilitates both the declarative configuration and automation within our testbed, as well as easier management and remote operation of the supporting infrastructure and resources. Finally, the reason for choosing a lightweight Kubernetes implementation is to minimize the resource utilization required for such a framework and extend the robots battery life. The association of experiment containers with the simulator instances will be performed using different DDS domains / partitions.

#### 2) ROS2 and DDS

The support for ROS2 using DDS enables a fully decentralised Real-Time Publish-Subscribe (RTPS) approach. This is also in accordance with the Industrial IoT Connectivity Framework (IICF) recommendations for real-time control applications [9]. The IoT bus is also supported over the proximity network by WiFi or Bluetooth/IEEE 802.15.4 radio.

#### 3) Gazebo Digital Playground

UMBRELLA Collaborative Robot platform will provide access to online simulation environments, built on Gazebo. These environments will provide a digital replica of the physical setup and are easily reconfigurable via the web portal provided. These playgrounds can be utilized for the experiment validation, prior to the deployment on the arena robots, or for the evaluation of more complex robotic use-cases and configurations. The Gazebo web client interface will be used for visualising the GUI output of the simulator to the users within WebGL compliant browsers (see Fig. 4).

#### 4) Robot nodes

The robot nodes (see Fig. 2) have many sensors and radios as well as a lifter actuator for moving pallets either individually or in a collaborative manner. The available sensors consist of:

- 16 IR Laser time-of-flight distance
- 4 Cameras
- 9-DoF IMU
- Temperature, pressure, humidity
- Strain gauges
- Robot health (various voltage, current and temperature readings)

The robot node processor is based on the RockPi™ 4B with GPU support. This permits machine learning and vision application experiment containers to be run on the robot nodes, that can exploit OpenCL.

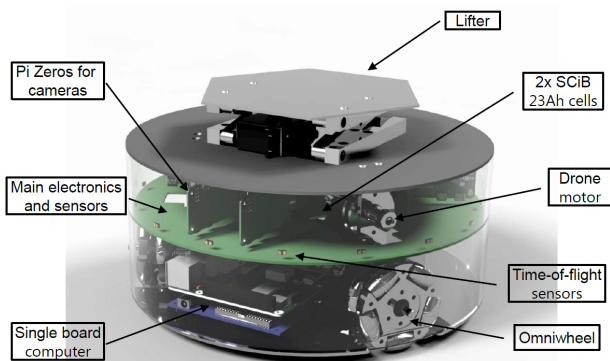


Fig. 2. Robot node prototype.

### 5) Ground truth telemetry

A ground truth telemetry system is provided, based on an Infrared optical tracking solution (Optitrack™) for monitoring and recording the actual movement tracks of robots with millimeter accuracy. Such systems can be used to measure performance of wireless localisation systems and estimate deviation of robots from their expected navigation trajectory.

### 6) Multi-wireless technology including private 5G.

The testbed infrastructure supports multiple wireless technologies including WiFi, UWB (Ultra-Wideband), and two Bluetooth/IEEE 802.15.4 radio modules that can be programmed for experimenting with novel wireless protocols. A testbed user can choose among many radios based on their strengths and needs for the applications. To offer an example, Bluetooth and IEEE 802.15.4 wireless connectivity is provided by Nordic Semiconductor’s NRF52840 chipset. Apart from standard Bluetooth, Thread, and ZigBee stacks, one can flash these chipsets with custom network protocol stacks to provide lower latency and more reliable communication often needed for industrial robotics. Furthermore, Decawave DW1000 UWB chipset can be exploited for accurate time synchronization and localization among the devices.

In addition, a private 5G network will be setup with dedicated equipment (base stations, core network and 5G dongles for robots) and configurations that can offer various customisations with optimised coverage for the collaborative robotics arena and address different requirements such as end-to-end QoS with network slicing. Such multi-wireless collaborative robotics testbed offers a demonstrable use case to evaluate and prove challenges for mission-critical industrial applications, e.g. teleoperation with safety concerns.

### C. Interfaces

The main backend interfaces supported by the testbed are shown in Table 1 and Fig. 3. These interfaces permit the support for testbed administration as well as the testbed users to access the testbed resources through the portal and external applications. Deployment of experiment containers, control of experiments and retrieval of experiment data is performed via the IoT Gateway and IoT Bus, with corresponding IoT bus interfaces shown in Table 2.

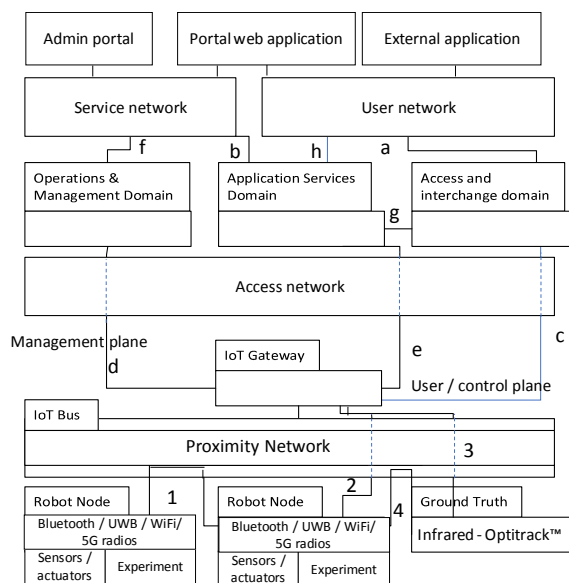


Fig. 3. IoT Reference architecture for Collaborative robotics Testbed

TABLE 1. Testbed Backend Interfaces

User network - Access and interchange	
a.1	API registration / API subscription and monitoring
a.2	API policy management and revocation
a.3	Access token / refresh token issue and verification
a.4	API gateways for access to production APIs
Service network – Application Services	
b.1	Update of asset / inventory related information
b.2	Software / hardware status and versions and location
Access network - Access and interchange	
c.1	Validate access token / violation alerts
c.2	Reporting of usage / performance KPIs
c.3	Identity for Kubernetes authentication provider
Access network - Operations / management	
d.1	Container operations - run / suspend / update security parameters
d.2	Retrieve logs / performance data
d.3	Update OS / FW / SW
d.4	Verify SW / HW inventory (SW versions / memory / CPU info)
d.5	Node Heartbeat status messages
Access network – IoT Gateway	
e.1	Retrieve result and ground truth log files using REST / HTTP
e.2	Issue commands using REST / HTTP
e.3	Ground truth tracking (optitrack™)
e.4	Radio simulator update using REST / HTTP
Service Network – Operations and management	
f.1	View / update infrastructure / asset information
f.2	Handle / action support ticket requests / incidents and alerts
f.3	Map overlay visualisations using WMS / WPS API
Access and Interchange – Application services	
g.1	Application service API invocations
User network – Application services	
h.1	Request video stream data (video camera streams – DASH)
h.2	Digital twin visualisation (gazebo web client)
h.3	Map overlay visualisations using WMS / WPS API

The backend interfaces with the user network support the end-user login and deployment of experiments in the testbed using the microservices in the application services domain. The service network supports the interfaces for operation and management of the testbed and the access network interfaces provide the interaction between the backend domains and the experiments running on the testbed resources (via the IoT gateway). The proximity network supports the interaction between the nodes / experiment containers and the IoT gateway. This proximity network supports the IoT bus, but it is exposed to the experiment containers through serial (USB) ports. This permits the deployment of the customised protocol

stacks within the containers. Hence, the swarm domain collaboration can use customisable protocol stack firmware that is deployed along with the experiment containers. The protocol stacks do not necessarily need to use the standardised IoT bus protocol (DDS), as they run within the same experiment context and can support flexible experimentation using novel radio and robot algorithms.

The experiment containers access the radio devices using a specified serial (USB) interface. The interfaces towards the sensors, actuators and IoT Gateway use the DDS standard with topics defined (see table 2) to permit the collection and subsequent retrieval of data via the IoT gateway. Ground truth and camera data is also passed via the IoT gateway. Optionally, ground truth can also be passed to the experiment containers and a radio simulator container if radio simulation or emulation is being performed.

TABLE 2. IoT Bus Interfaces

-Robot swarm domain collaboration - customised protocol stacks	
1.1	Telemetry (robot nodes – proximity, relative position)
1.2	Formation control
1.3	Status and sensor updates
1.4	Discovery
1.5	Leader election and task assignment etc.
1.6	Sensor data streams
-Robot - IoT gateway	
2.1	Telemetry (robot nodes – proximity, relative position)
2.2	Event stream (raw sensor data for storage / processing)
2.3	Monitoring KPIs (traffic, latency, invocations etc.)
Ground truth telemetry / Cameras – IoT Gateway	
3.1	Ground truth telemetry data stream
3.2	Commands (start / stop / stream data or video)
Ground truth – Experiment (optional)	
4.1	Ground truth telemetry data stream

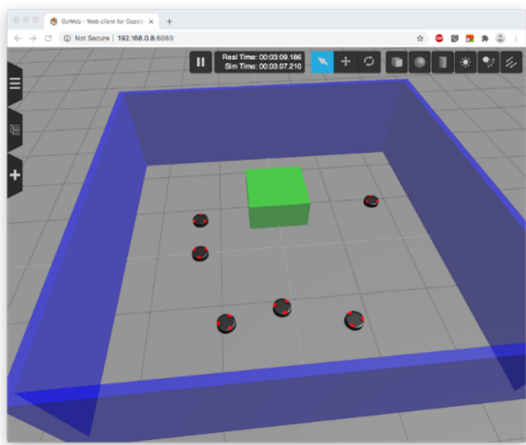
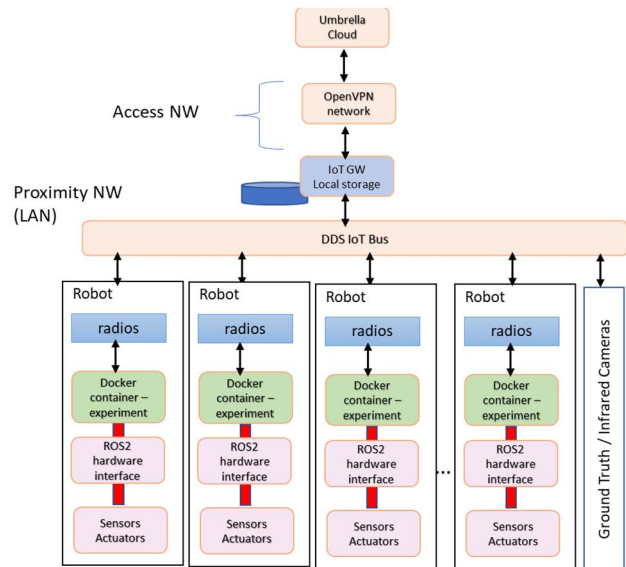


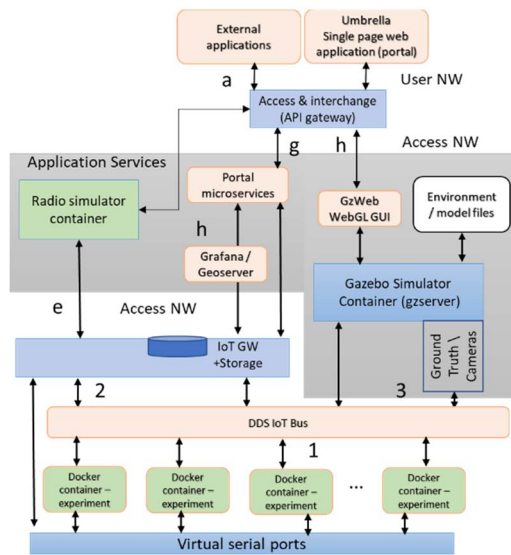
Fig. 4. GzWeb GUI showing robot arena.

#### D. Software architecture

The overall software architecture for the robot testbed is shown in fig. 5. The collaborative robotics arena and simulation environment have a similar approach to permit experiment containers to be evaluated in both the simulated and real arena environments without change. This is an essential step for the validation of experiment algorithms and protocols prior to deployment in the robot arena. It also permits evaluation of experimental algorithms with larger numbers of swarm robot nodes or different and larger deployment environments.



(a)



(b)

Fig. 5. Software architecture of collaborative robotics testbed (with experiment containers shown in green) for (a) arena (b) simulator.

The software architecture permits the scheduling of the experiments and upload of the corresponding experiment containers. It also supports the visualisation of results data in graphs (using Grafana) or spatial map overlays (using Geoserver) in the single page web application. In addition, the results data or ROS2 bags can be downloaded from the portal via the IoT gateway.

### III. IOT PLATFORM

The testbed IoT platform architecture is illustrated in fig. 5 and supports secure testbed operation using different cloud and edge resources. The common interfaces, container deployment and security approach permit flexibility to use both cloud-based simulation and physical robotics arena platforms.

#### A. Access and Interchange

The collaborative robotics testbed IoT platform is primarily based on REST APIs for interactions with the

backend services and applications. The API access is controlled through the access and interchange domain. Prior to invoking APIs an access token is acquired through the WSO2 API manager [8] using the application key and user credentials or Single Sign-On (SSO) identity token. After this the token is embedded into the authorisation header of the API requests and are validated by API gateways. This permits authorisation and monitoring, with associated role-based access control (RBAC). The access tokens are based on either JWT or OAuth2 and uniquely identify the application and users to grant appropriate level of access. In this manner the testbed users can be assigned different levels of service and access to testbed resources. This is particularly important in the case of the collaborative robot testbed to prevent unauthorised access to robot nodes and potential for resource starvation caused by parallel experiments running on simulator instances.

### B. IoT Gateway

The IoT gateway is a key element that provides REST APIs to the backend and DDS RTPS interfaces towards the IoT Bus with robot nodes, gazebo simulator and ground truth telemetry. The DDS domains, partitions and topics are defined to isolate the local robot from the swarm interactions, as well as different users' experiments. This is particularly important in the case of the simulation environments to permit parallel simulation of different experiments.

In addition, when the optional radio simulation containers are deployed the IoT gateway performs the necessary redirecting of the swarm domain and Gazebo simulator ground truth telemetry topics to the corresponding simulator instance using ROS2/DDS topics and a REST API provided by the simulator containers. The radio simulator container emulates the radios and returns a response to determine how the IoT gateway sends the message to experiment container instances. This is enabled by the simulator response containing performance parameters corresponding to recipient nodes. The performance parameters include a success probability and latency, which are used to adjust the delivery of the messages prior to sending to experiment containers via the associated virtual serial ports.

All interactions are also recorded as REST API log traces and ROS2 bags by the IoT gateway for later retrieval by the users. These can be downloaded for post processing and experiment evaluation.

### C. Operations and Management

The operations and management domain permits testbed administrators to provision, configure and monitor the nodes and software. This domain is separated in two different logical entities, i.e., the robot plane and the backend plane.

The robot plane, as mentioned above, is supported by a lightweight Kubernetes implementation. It is administered by a Master node, operating on a server next to the arena within the same proximity network. This Kubernetes plane is responsible for the deployment of the different experiments as well as the control plane components of the platform. Such components are the kube-apiserver that provides remote access from the Application Services and Access and Interchange domains to the end-nodes via REST APIs interactions, the kube-scheduler responsible for scheduling the experiments on the nodes, and the kube-controller-

manager that handles the different processes, node and endpoint controllers.

The backend plane, based on a highly-available native Kubernetes cluster and being part of the entire UMBRELLA platform, is responsible for the monitoring, control, and configuration of the supporting infrastructure. With what regards the Collaborative Robot Testbed it can interact with the nodes via MQTT and REST APIs for the monitoring, the radio and sensor configuration, and for the exchange of results data. It is also supported by highly available microservices that are responsible for the user management (Keycloak), the container storage and deployment (Harbor Private Docker Registry), and various high availability time series and relational databases for storing the experimental results and support the key system components. What is more, our system is integrated with Ansible [11] and CmdBuild [7] configuration management frameworks for handling the software and hardware updating. The above infrastructure provides the advantage that new robots can be easily added, orchestrated and automatically configured, while the existing ones can be maintained and updated frequently to ensure the security of the platform.

### D. Experiment containers

Experiment containers use Docker to encapsulate the experiment code for deployment to either the cloud environment (for simulation) or the robot nodes. The deployment is controlled using the lightweight K3S cluster distributed processing framework. This permits a separation of resources and central management and monitoring through the cluster master. The containers have interfaces to support DDS / ROS2 interactions with sensors and actuators and other robots as well as the IoT gateway.

Each robot node operates with a daemon service that is responsible for connecting to the various sensors and wireless interfaces on the robot nodes. The daemon handles the monitoring and re-flashing of radio devices, allowing the experiments to use customised protocol firmware for node-node collaboration and interfaces with the user experiments (containers). This interaction with the user containers is via the topic information exposed from the daemon using ZeroMQ and a publish-subscribe scheme. Using such a configuration, the user later can exploit the Bluetooth/IEEE 802.15.4, UWB or WiFi radios on the robots to experiment with real-time protocols and their enhancements. In this case the custom protocols can be simulated in the optional radio simulator containers. The radio simulators are deployed in the same manner as the other experiment containers and can exploit raytracing, machine learning or other techniques to estimate the communication performance between each robot node.

Experiment containers are first uploaded by the user to the Umbrella platform using the portal, provided as a single page web application and experiment microservice (MS). The portal also permits the selection of multiple configurations, that may consist of different experiment containers or binary files. Each configuration can select a subset of robot nodes that will be involved in the experiment. The experiment can then be scheduled and will run for the specified duration and permit return of results, ROS2 bag and console/log output data. Video streams from arena cameras can also be viewed in the portal during the experiment for remote observation and recording.

### E. Simulator

The Gazebo robot simulator is provided with robot and arena environment models in order to evaluate experiments prior to arena deployment. These include full physics-based models at different resolutions, with sensors such as ranging, IMU and cameras. The experiment containers interact directly with the simulator instance in the experiment domain / partition in order to access the ROS2 topics relating to the simulated sensors and actuators. The configuration optionally supports simulation of radio devices by using the virtual serial ports together with redirection via the IoT gateway. These serial port messages for the local container ports are relayed to the corresponding radio simulator container instance rather than directly between experiment containers. An example sequence to deploy experiment containers and retrieve data from a simulator instance is shown in fig. 6. In this example firstly the robot simulator, radio simulator and experiment containers are deployed. Then during the experiment, the robot domain interactions permit the simulated local robot interactions with sensors and actuators.

The swarm domain interactions are facilitated by intercepting virtual serial (USB) port messages from the experiment containers and ground truth data from the simulator. The serial port data is passed via the IoT gateway using a serial over IoT bus encapsulation method and relayed to the corresponding radio simulator container instance using the simulator REST APIs. The radio simulator then determines the estimated radio performance for each simulated robot, embeds this information and returns the message to the specified experiment containers in the response. The ground truth data sent from the robot simulator is updated at a periodic refresh rate to provide synchronisation between the radio and robot simulators.

This permits experiments, including the swarm interaction, to be validated prior to the deployment on the arena testbed. It also permits simulation of much larger deployments in more complex environments. If there is no radio simulation the serial port redirection to the IoT gateway provides a perfect channel between the experiment containers rather than simulating the radio performance.

### IV. RADIO PROTOCOL STACKS

Customised radio protocol stacks are supported on the testbed platform by abstracting the radio interfaces as serial (USB) ports. An upload facility is provided to flash the radio hardware with the customised firmware. Protocol stacks deployed in the firmware or within the experiment containers can then use the serial ports to communicate with other experiment containers running on other robots. A reference mapping is also provided between Bluetooth mesh or WiFi and DDS to facilitate a DDS multicast swarm domain communication facility. This can be deployed as an IoT gateway within the experiment container to be accessible from ROS.

### V. COMPARISON OF MULTI-ROBOT TESTBEDS

This section compares the key features and capabilities of prominent multi-robot testbeds. Table 3 indicates the main features of Robotarium [3] and IRIS [5] compared with the UMBRELLA collaborative robot testbed. In particular, the UMBRELLA testbed provides an online simulation framework including radio simulation, which the others do not

support. It also has more capable and larger robots with more processing, radio and communication stack options. For instance, the GPU capabilities of the RockPi™ permits real-time video processing on the robots which can be used for localisation and object or neighbouring robot tracking. In addition, the omni-wheel approach enables the robots to move in any direction and robot orientation. This allows a more flexible and complete approach to evaluating collaborative multi-robot systems, including the wireless communication, localisation and scalability aspects. Apart from Robotarium and IRIS, several testbeds, precisely surveyed in [12], have been built with their major focus on wireless networking research. UMBRELLA sets itself apart from these due to its ability to provide both simulation and real-world testbed features of robot and radio communication as part of a single suite. This enables a more flexible and complete evaluation of swarm systems.

The customisation of radio protocol stacks is provided using experiment containers together with radio firmware re-flashing support in order to permit evaluation of novel protocols stacks in conjunction with swarm robotics algorithms. To the best of our knowledge this capability is not available in any prior collaborative robot testbed infrastructure. Moreover, synergist operation of networking, localisation, and robotic swarms for reproducing practical warehouse robotics situations is a unique feature not supported by existing testbeds.

### VI. CONCLUSIONS

This paper provides an overview of the UMBRELLA collaborative robotics testbed and IoT platform in support of collaborative multi-robot experimentation. The testbed is unique in that it supports an example practical warehouse use-case, which can be used to evaluate the benefits of swarm robotics approaches. The testbed also provides a simulator framework for validation and evaluating larger and more complex deployments, including the radio communication and protocol stack aspects. Users can deploy experiments and radio simulators using a flexible container-based approach. Support for custom radio firmware deployment is also provided to evaluate novel radio protocols, such as based on Bluetooth mesh or UWB. The testbed will be provided as an open experimentation platform for researchers and industrial evaluators of swarm technologies.

### ACKNOWLEDGMENT

The Urban Multi Wireless Broadband and IoT Testing for Local Authority and Industrial Applications (UMBRELLA) testbed platform aims to create a Centre of Excellence for ‘open’ research and innovation into future Industrial Internet of Things (IIOT) linking key academic, research and economic locations. The project is funded in conjunction with South Gloucestershire Council by the West of England Local Enterprise Partnership through the Local Growth Fund, administered by the West of England Combined Authority.

The authors acknowledge the valuable contributions of the UMBRELLA team in designing and developing the UMBRELLA testbed. In particular, Eliot Glasspole, Elliot Hogg, Ingram Weeks, Theo Spyridopoulos, Woon Hau Chin, Ben Holden, Pietro Carnelli, Thomas Berton, Michael Baddeley, Nan Jiang, Maruthi Chintam, Sandeep Shivaraju, Mamadur Nagaraj, Albert Pinto, Abhishek Somashekar, Shanthala Nagaraja, Sandeep Maranna and Varada Narendra.

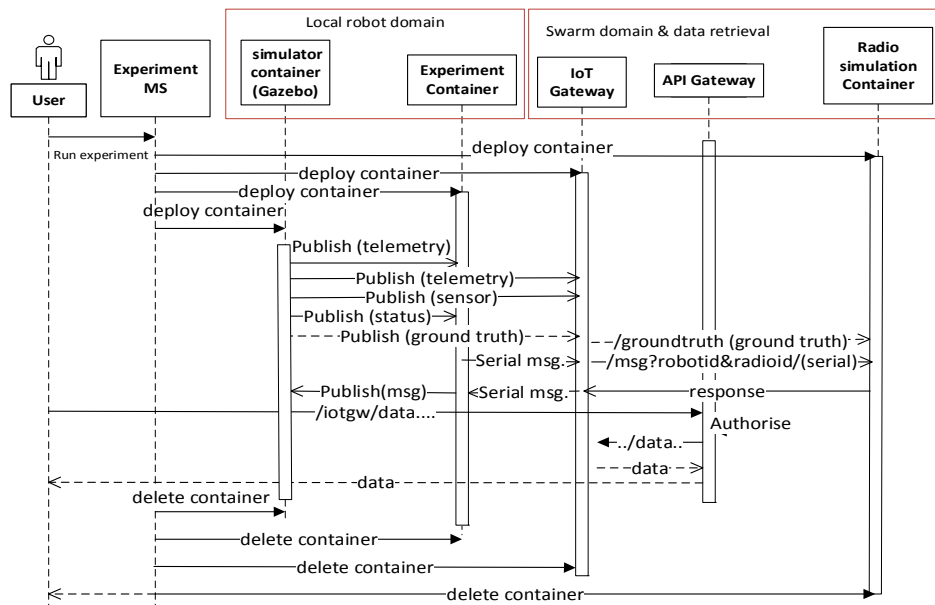


Fig. 6. Example sequence for simulation experiment and data retrieval.

TABLE 3. Testbed Comparison Summary

Testbed	Simulator	Number of nodes	Robot size	Node cost	Communication & Management Stack	Online	Actuation	GPU
Robotarium [3]	MATLAB & python offline + online with virtual robots	20	11cm x 10cm x 7cm Arena 3mx2m	\$50	WiFi	Yes	2xWheel movement	No
IRIS [5]	No simulator	<10	Ø 9.5cm	\$ 350	ROMANO (MQTT-SN)	No	2xWheel movement	No
UMBRELLA	Gazebo robot and custom radio simulation models	10-20	Ø 25cm Arena 5mx5m	\$1000	Bluetooth 5.0, WiFi (802.11ac), (5G), DDS and support for custom protocol stacks	Yes	3xOmni-wheel movement + lifter	RockPi with 4 onboard cameras

## REFERENCES

- [1] International standard for “Internet of Things (IoT) — Reference Architecture”; ISO/IEC 30141:2018
- [2] Umbrella project introduction, weblink, <https://www.westofengland-ca.gov.uk/wp-content/uploads/2019/11/LGF-BF-Umbrella-Oct19.pdf>.
- [3] Daniel Pickem, Paul Glotfelter, Li Wang, Mark Mote, Aaron Ames, Eric Feron, and Magnus Egerstedt; “The Robotarium: A remotely accessible swarm robotics research testbed”; 2017 *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, May 29 - June 3, 2017
- [4] X. Teh, A. Aijaz, A. Portelli, and S. Jones, “Communications-based Formation Control of Mobile Robots: Modeling, Analysis and Performance Evaluation,” *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Alicante, Spain, Nov., 2020.
- [5] J. A. Tran *et al.*, “Intelligent Robotic IoT System (IRIS) Testbed,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [6] Gazebo robot simulator website - <http://gazebo-sim.org/>
- [7] Cmdbuild asset management tools website - <https://www.cmdbuild.org/en/homepage>
- [8] WSO2 API Manager website - <https://wso2.com/api-management/>  
WSO2 API Manager website - <https://wso2.com/api-management/>
- [9] “The Industrial Internet of Things Volume G5: Connectivity Framework”; IIC: PUB:G5:V1.01:PB:20180228; <http://www.iiconsortium.org/IICF.htm>
- [10] A. Aijaz, “Private 5G: The Future of Industrial Wireless.” accepted for *IEEE Industrial Electronics Magazine*, preprint *arXiv:2006.01820* (2020). <https://arxiv.org/abs/2006.01820>
- [11] Ansible configuration management; <https://github.com/ansible/ansible>
- [12] A.-S. Tonneau, N. Mitton, and J. Vandaele, “How to choose an experimentation platform for wireless sensor networks? A survey on static and mobile wireless sensor network experimentation facilities,” *Ad Hoc Networks*, vol. 30, pp. 115–127, 2015.