JYU. Since 1863.

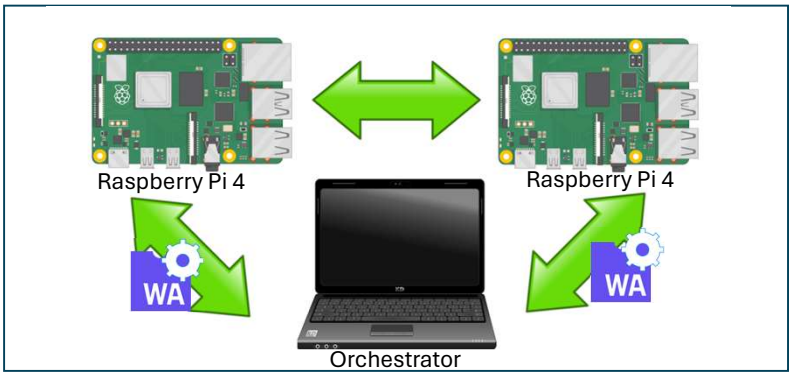
1. FACULTY OF INFORMATION TECHNOLOGY, UNIVERSITY OF JYVÄSKYLÄ
2. FACULTY OF COMPUTING, TAMPERE UNIVERSITY

Pyry Kotilainen[1], Viljami Järvinen[1], Teemu Autto[1], Lakshan Rathnayaka[2], Tommi Mikkonen[1]

Demonstrating Liquid Software in IoT Using Webassembly

ABSTRACT

In this paper we introduce a demonstration of our prototype orchestration system utilising WebAssembly to achieve isomorphism for a liquid software IoT system. The demonstration hardware consists of two Raspberry Pi IoT devices and a computer acting as the orchestrator. The audience can interact with the orchestrator through a web interface to deploy different software configurations to the devices, and observe the deployment process as well as the deployed application in action.



The demo system consisting of A laptop acting as an Orchestrator for 2 Raspberry Pi 4 devices with the Supervisor

DEMONSTRATION SYSTEM

The Orchestrator serves a web GUI through which users can change software deployment on devices and observe the current deployment, changes on it, as well as telemetry from the devices and results of the deployed distributed application. The GUI offers following functionality:

- Configuring a new deployment for the devices.
- Visually follow deployment progress and state.
- Triggering deployed functions if applicable.
- Visual feedback of results of each function.

WASMIOT LIQUID SOFTWARE

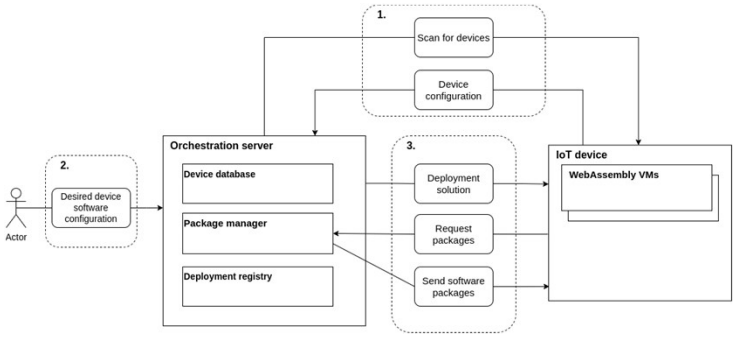
The system consists of an orchestration server or orchestrator and a variable number of heterogeneous node devices in the same local area network. An actor (user or another system that interacts with our system) can control the system through the orchestrator.

Aside from communication and application logic, the orchestration server consists of three components:

Device database contains the hardware configurations of the various devices, and it is populated by listening to mDNS messages and requesting information from the associated devices.

Deployment registry contains all executed deployments by the orchestrator. With each deployment listing the devices involved and the services they provide in it.

Package manager maintains a database of all available WebAssembly software modules that can be sent to the devices. It is also capable of resolving dependencies to provide a complete list of required modules for a given module to run.



Overview of the Wasmiot software architecture

REPOS AND MORE INFORMATION

LiquidAI project: wasmiot-orchestrator, https://github.com/LiquidAI-project/wasmiot-orchestrator
LiquidAI project: wasmiot-supervisor, https://github.com/LiquidAI-project/wasmiot-supervisor