RE4DY

MANUFACTURING DATA NETWORKS

RE4DY TOOLKIT

Name of the Tool	Meta Repository Demonstrator
Tool Owner	Industry Commons Foundation
Version	1.0
Date	Nov 2025
Version	V1.0

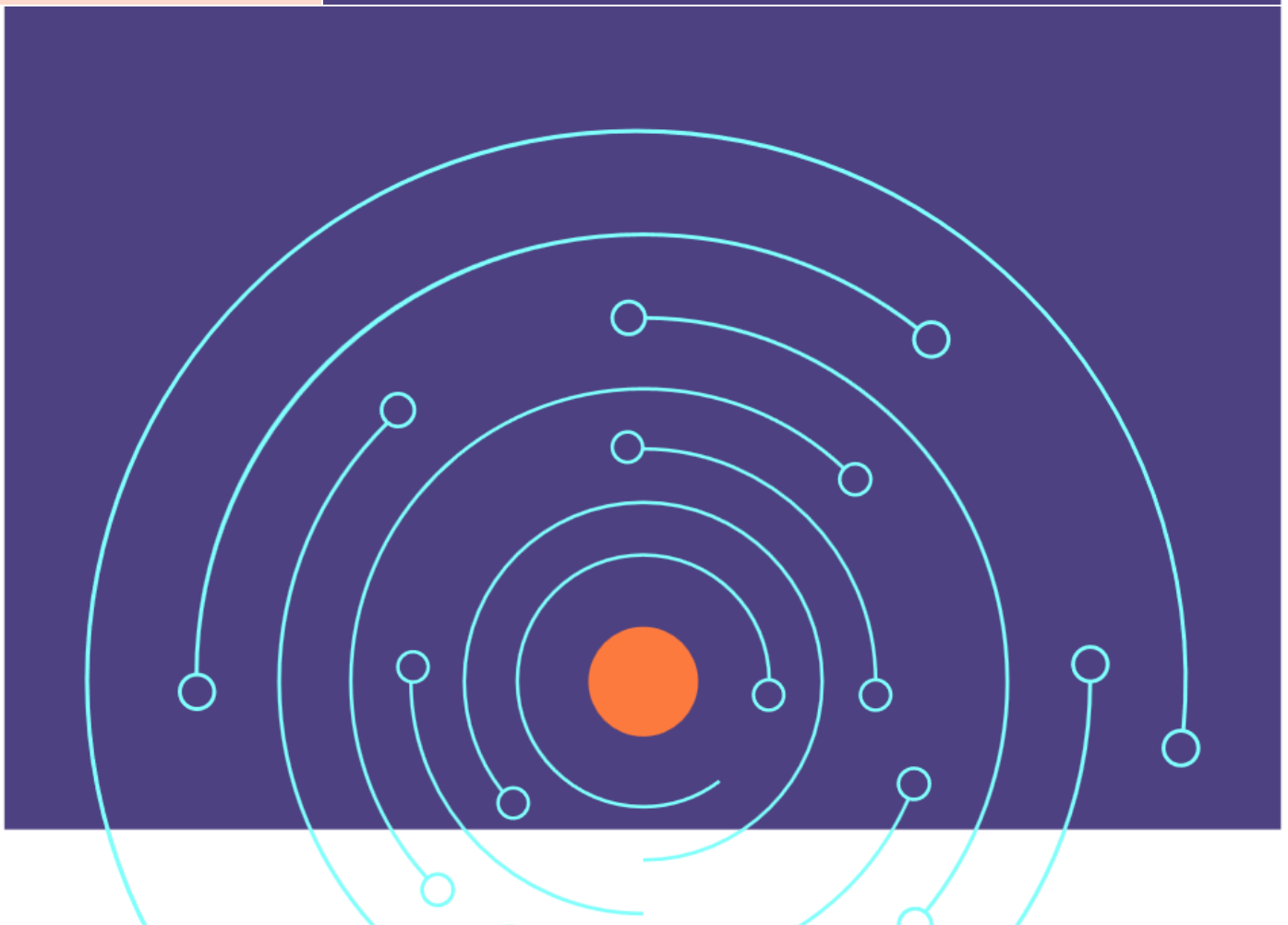


Table of contents

1. Component Description

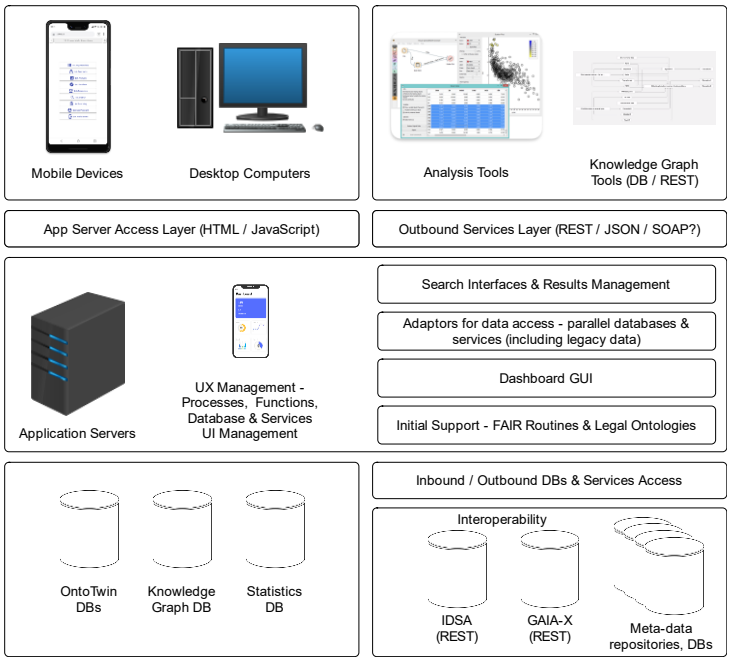
The Meta Repository Demonstrator (MRD) is a management infrastructure for related data and meta-data for objects, knowledge resources, interconnectivity resources and relationships. MRD allows the management of a large type of information resources of different types and contents that can be extended over time to meet future challenges. The MRD can be used to demonstrate a simple Digital Twin management system during the RE4DY project.

2. Input

As a Digital Twin, the resources during the demonstration are manually added (option if needed - import of existing data) and linked to other relevant objects in the installation. Beyond industry-specific information types, data can be entered in the form of media, external links to web-based data, and external databases and services in specific cases. External data storage (measurement and time-series data) is an example of an external service.

3. Output

End-user web-based GUI for desktop systems (optionally, if time allows, also mobile device GUI). Access to data via REST calls and direct database access calls is optional. The end-user environment can act as an interface to other externally connected systems. Report and export functionalities.



4. Information Flow

MRD can be regarded as a resource hub architecture designed to effectively support large-scale P2P implementation, storing and managing a multitude of data and information that can be linked freely between themselves and with data stored in external (also legacy) data.

5. Internal Architecture

The system is separated into three main functionalities: database manager, application server and interconnected end-user devices of different types.

6. API

Beyond the end-user GUI devices of different types, the main APIs for access are REST and, in relevant cases, direct-controlled DB access or SOAP/XML.

7. Implementation Technology

- End-user systems: webpage-based GUI for desktop and mobile devices, native applications for Android and iOS (mobile phones and tablets).
- REST access (JSON)

- Application server: Omnis Studio application server (omnis.net) for managing end-user devices and internal/external database and services access.
- Database server: PostgreSQL database server. Used for both core database systems and for supplementary/supporting databases such as Knowledge Graph (pgGraph), Vector (LLM) database (pgVector) and statistical support/time series database systems.
-

8. Comments

Omnis Studio was chosen as a development tool for application server environments and end-user systems due to its effective pilot development processes. The application server layer can later be developed in other development environments using the now-developed system as a specification example.