FAIR Internet of Things Data: Enabling Process Optimization at Munich Airport

Michael Freund[1][0000−0003−1601−9331], Julian Rott[2][0000−0002−9882−6737], Rene Dorsch[1][0000−0001−6857−7314], and Andreas Harth[1,3][0000−0002−0702−510X]

[1] Fraunhofer Institute for Integrated Circuits IIS, Nürnberg, Germany
firstname.lastname@iis.fraunhofer.de
[2] Munich Airport, Munich, Germany
[3] Friedrich-Alexander-Universität Erlangen-Nürnberg, Nürnberg, Germany

Abstract. We present how data collected from Internet of Things (IoT) devices adhering to the FAIR data principles forms the foundation for data analytics applications at Munich Airport. We describe how the collected IoT data is annotated, how our APIs are structured, present two data analytics applications currently in use to analyze FAIR IoT data for process optimization, and share lessons learned.

Keywords: FAIR data principles · Internet of Things · Solid.

1 Introduction

The Internet of Things (IoT) has become an integral part of enterprise process monitoring systems, enabling improved transparency of operational processes through the collection of additional data. In general, IoT devices operate continuously and generate large amounts of heterogeneous data [1]. For instance, our IoT system at Munich Airport has been in operation for about 9 months and is installed at a baggage check-in counter. The IoT system consists of three RGB stereo depth IoT cameras and processes approximately 2,700 pieces of baggage on a typical day, resulting in approximately 8,100 timestamps, 8,100 RGB images and 8,100 depth images.

Managing, integrating, and using the raw data produced by our IoT system presents challenges, especially data fusion complexities [5], as the images and timestamps require additional provenance data and need to be integrated with existing datasets. Furthermore, there are difficulties in efficiently finding images and provenance data associated with a particular piece of baggage and in securely accessing the found data segments with access control. Challenges related to Findability, Accessibility, Interoperability, and Reusability (FAIR) are prevalent not only in enterprise IoT environments but also across various domains. As a potential solution, the use of FAIR data principles [6] has proven to be effective.

2 Approach

In our setup at Munich Airport, we use an IoT architecture introduced in previous work [2]. A key component of the architecture is the use of Solid [4]. Solid

Listing 1. RDF data describing an image and provenance data.

```
1  @prefix sosa: <http://www.w3.org/ns/sosa/>.
2  @prefix prov: <http://www.w3.org/ns/prov#>.
3  @prefix dc: <http://purl.org/dc/elements/1.1/>.
4  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
5  @prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
6  @prefix : <http://www.munich-airport.de/ontology/fmo#>.
7  @base <http://www.munich-airport.de/iotdata> .
8
9  <#image1> a prov:Entity;
10   dc:format "image/png"; dc:creator <#cam37>;
11   prov:generatedAtTime "2023-12-14T11:54:01"^^xsd:dateTime;
12   dc:source
         <http://munich-airport.de/cam37/20240114/150340.png>.
13
14 <#cam37> a sosa:Sensor;
15   sosa:observers <#baggageTop>;
16   rdfs:label "Camera 37";
17   sosa:isHostedBy <platform1>.
18
19 <#baggageTop> a sosa:ObservableProperty;
20   rdfs:label "Top view of transported baggage.".
21
22 <#imageProcessingAgent> a prov:Agent;
23   rdfs:label "Image Processing Agent".
24
25 <#imageProcessingActivity> a prov:Activity;
26   prov:wasAssociatedWith <#imageProcessingAgent>;
27   prov:used <#image1>.
28
29 <#processedData20231214115401876282> a prov:Entity;
30   prov:wasGeneratedBy <#imageProcessingActivity>;
31   :baggageHeight "0.1"^^xsd:decimal; :color "FF0000".
```

is a specification enabling secure and decentralized data storage built on the Linked Data Platform (LDP). The use of Solid forms the basis for ensuring that our IoT data adheres to FAIR principles. A snippet of the stored RDF data can be seen in Listing 1 and is used as a reference in the following.

Findability To ensure findability, we assigned each collected image a unique identifier in the form of a URI. We added RDF descriptions to the image data, such as the data format and creator of the image (line 10), the timestamp when the image was generated (line 11), and the URI of the image resource (line 12). Both the RDF metadata file and the images are stored in Solid and linked to the camera that created the image, which in turn is linked to the entire IoT platform. This allows us to use link following to discover and find images.

Accessibility Using Solid allows us to use HTTP as a means to securely access our IoT data, and allows us to use access control lists to restrict access to unauthorized users. By separating the RDF metadata files from the images, we ensure that the metadata remains available even if the image data is no longer accessible. Using HTTP not only provides easy access to the data, but also ensures that the required ports are not blocked by typical firewall configurations.

Interoperability Metadata is stored using RDF in combination with standard and open vocabularies such as Dublin Core, SOSA/SSN, and PROV-O (lines 10, 15, 26) to describe the data and ensure interoperability. In addition, we use a custom ontology developed for Munich Airport specific entities using the LOT methodology [3] (line 31).

Reusability We organized the IoT data according to the airport's established internal data structures to promote data reuse. In addition, we annotated the image data with provenance information to track the data processing steps and algorithms involved (line 22), thereby clarifying and documenting how the data was generated.

Solid API The HTTP API of the Solid storage is based on the hierarchical structure of IoT devices. For example, the data collected by `cam37` is stored in an LDP container for the camera, which contains LDP containers for each day the system has been running, each containing image data named by its creation timestamp (line 12). In addition, each date-specific LDP container has its own metadata file containing RDF data, similar to Listing 1.

3 Applications

The FAIR-compliant images and metadata stored in the Solid storage form the basis for applications. The applications are designed to consume the FAIR data, extract additional information useful for process improvement, and generate new FAIR data. Currently, we have implemented and deployed two such applications:

Size and Color Feature Extraction The first application queries the metadata of the images stored in Solid to identify groups of images that correspond to a single piece of baggage, and retrieves all identified images via HTTP. The application then uses the group of images to create a 3D point cloud of the baggage. The 3D point cloud is then used to estimate the dimensions of the baggage and to extract the four most dominant colors. The extracted information, along with metadata such as provenance data, is added back to the Solid storage as RDF triples using HTTP PATCH requests.

Baggage type classification The second application also queries the metadata of the images stored in Solid and retrieves the color image taken from the top perspective via HTTP. The application uses a pre-trained image classification algorithm to perform inference on the retrieved image to identify different types of baggage, such as suitcases or backpacks. The classified category, along with the associated probability score and provenance data, is added as RDF triples to the Solid storage using HTTP PATCH requests.

The information stored in Solid is then linked and integrated with existing RDF process data. The integrated process information enables optimization of operations, such as evaluating lead times for handling specific types of baggage or creating loading plans based on estimated baggage dimensions.

4 Lessons Learned

During the implementation of FAIR principles for IoT data at Munich Airport, we, the responsible project team, gained several insights:

- The initial process of making IoT data FAIR-compliant is more complex than traditional methods. But the FAIR approach offers long-term benefits, especially in application development and data integration.
- Provenance data can be central to troubleshooting and understanding nested processing steps because data aggregation steps and analysis algorithms used are documented along with the actual data.
- Using Solid for data storage supports FAIR compliance through the restrictions of the Solid protocol, such as the use of HTTP and RDF, and provides an actively maintained, mature server software implementation.
- Combining HTTP with access control lists simplifies data access for everyday users by enabling data retrieval using standard web browsers.

5 Conclusion and Future Work

In this paper, we detailed our approach to making IoT data FAIR at Munich Airport. We presented two deployed applications that consume and produce FAIR data, and shared lessons learned during the implementation of the project. Looking ahead, we plan to deploy more IoT devices and develop additional analytics applications.

Acknowledgements This work was funded by the Bayerisches Verbundforschungsprogramm (BayVFP) des Freistaates Bayern through the KIWI project (grant no. DIK0318/03).

References

1. Cai, H., et al.: IoT-based big data storage systems in cloud computing: perspectives and challenges. IEEE Internet of Things Journal **4**(1), 75–87 (2016)
2. Freund, M., et al.: WoT2Pod: An Architecture enabling an Edge-to-Cloud Continuum. International Conference on Internet of Things (2023)
3. Poveda-Villalón, M., et al.: LOT: An industrial oriented ontology engineering framework. Engineering Applications of Artificial Intelligence (2022)
4. Sambra, A.V., et al.: Solid: a platform for decentralized social applications based on linked data. MIT CSAIL & Qatar Computing Research Institute, Tech. Rep. (2016)
5. Wang, L.: Heterogeneous data and big data analytics. Automatic Control and Information Sciences **3**(1), 8–15 (2017)
6. Wilkinson, M.D., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific data **3**(1), 1–9 (2016)