Chapter 4
Fundamentals of Real-time Linked Dataspaces

Keywords Data platform · Linked data · Stream processing · Event processing · Dataspaces · Internet of Things · Incremental data management

4.1 Introduction

Dataspaces can provide an approach to enable data management in smart environments that can help to overcome technical, conceptual, and social/organisational barriers to information sharing. However, there has been limited work on the use of dataspaces within smart environments and the necessary support services for real-time events and data streams. This chapter introduces the Real-time Linked Dataspace (RLD) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and (near) real-time processing capabilities. The RLD has been specifically designed to support the sharing and processing of data between intelligent systems within smart environments. We propose a set of specialised dataspace support services to enable the requirements of loose administrative proximity and semantic integration for event and stream systems. These requirements form the foundation of the techniques and models used to process events and streams within the RLD.

The chapter is structured as follows: event and stream processing for the Internet of Things are discussed in Sect. 4.2, and the fundamentals of RLD are defined in Sect. 4.3 including principles, comparison to existing dataspaces, and the main components of the architecture. Section 4.4 details a principled approach to pay-as-you-go data management and introduces the 5 star pay-as-you-go model for RLD support services. Section 4.5 introduces the support platform for the RLD, Sect. 4.6 discusses its suitability as a data platform for intelligent systems within smart environments by comparison to similar platforms, and a summary is provided in Sect. 4.7.

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_4

4.2 Event and Stream Processing for the Internet of Things

At the end of the twentieth century and in the first decade of the twenty-first century, a recognition emerged among researchers and practitioners that a new class of information processing systems was needed. The need for the event processing paradigm emerged from a range of diverse distributed applications that required on-the-fly and low-latency processing of information items. Example applications include environmental monitoring [124], stock market analysis [125], RFID-based inventories [126], resource management (e.g. energy, water, mobility) in smart environments [4], and security systems such as intrusion detection [127].

Smart environments have emerged in the form of smart cities, smart buildings, smart energy, smart water, and smart mobility, all of which have large quantities of real-time data that must be processed. The Internet of Things (IoT) is producing events and streams that are generating significant quantities of data within smart environments which are driving a new wave of data-driven intelligent systems that can more effectively and efficiently manage resources while also providing enhanced user experience.

The paradigms of event processing and stream processing have evolved through the work of several communities, including active databases [128, 129], reactive middleware [130–132], event-based software engineering [133, 134], and Message-Oriented Middleware [135]. As these paradigms emerged, they created their own communities around data stream management systems [136–139], event processing systems [140, 141], Complex Event Processing [140], and Publish-Subscribe [142, 143].

Cugola and Margara [139] complement this picture to justify a new umbrella paradigm for a set of emerging systems where "timeliness and flow processing are crucial" and call it the Information Flow Processing (IFP) Domain. Figure 4.1 presents an elaboration of Cugola and Margarita's functional model and shows the main functionalities of an IFP engine for single and complex event processing. In event processing, data items that are shared in real-time are called *events*. An event

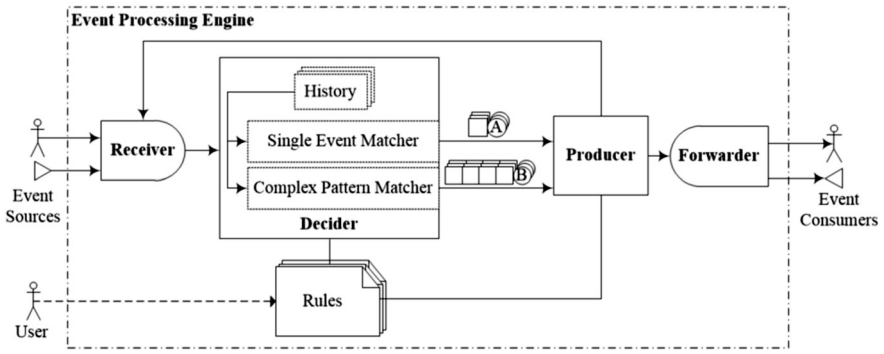

Fig. 4.1 The information flow processing model for single and complex event processing [144]

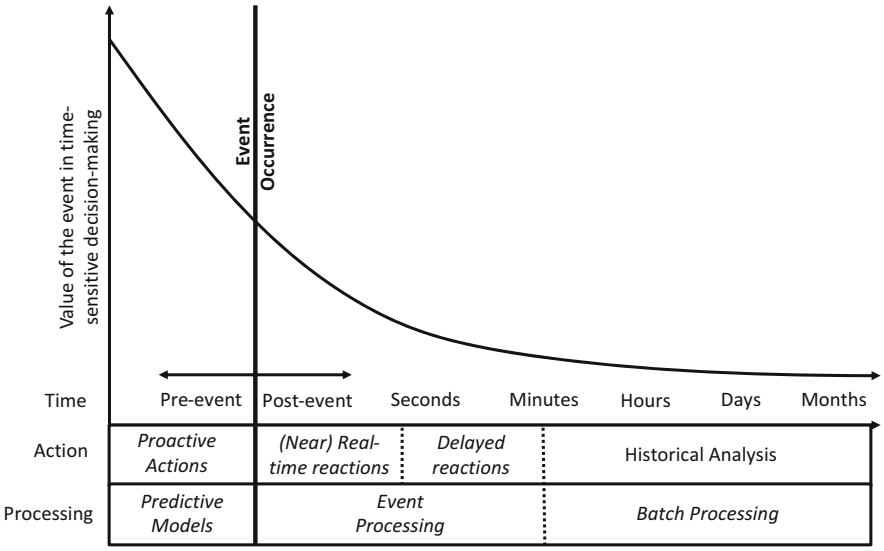

Fig. 4.2 The value of events in time-sensitive decision-making, actions, and processing approaches

can take the form of a sensor reading. Data sources which produce events are called *event producers*. Users and software which are interested in an event, or set of events, are called *event consumers*. An essential part of the event processing paradigm is the matching mechanism between the events and the interests of event consumers. This is similar to the concept of query processing in relational database management systems, where events replace the concept of a data tuple, and subscriptions or rules replace the concept of queries. In a specific family of event processing, called stream processing, queries take the name of continuous queries as they are evaluated continuously against moving data.

4.2.1 Timeliness and Real-time Processing

The concept of timeliness has been expressed in the literature using various terms such as low latency [145, 146], high throughput [146, 147], low delay [148], and real-time processing [3, 149]. All these terms, except real-time processing, can be classified under the umbrella of fast computing. This term means that the system is efficient in processing information items in a way that the ratio Value/Time is maximised as detailed in Fig. 4.2.

Another perspective is that real-time processing includes the notion of executing the information processing task within a time constraint, called a deadline [150]. Timeliness, as described by Cugola and Margara [139], is more similar to

the concept of fast computing. Technically, it can be measured by the related concepts of latency and throughput:

- *Latency* is defined in this book as the total time required to process an information item starting from its arrival to the processing agent until its completion.
- *Throughput* is the number of information items wholly processed within a time unit.

Thus, real-time, whenever used in this book, means low latency, high throughput, and processing as soon as the information items are available. Near-real-time processing, when used, should also be taken to have this meaning. The ability to quickly react to an event is a critical requirement within many real-world situations. Systems that can provide data processing capabilities suitable for real-time information need to be designed in a specific manner, Stonebraker et al. [3] suggest eight requirements for an effective and efficient design:

- *Rule 1—Keep the Data Moving:* Process messages "in-stream" without the need to store the message to perform an operation or sequence of operations.
- *Rule 2—Query Using SQL on Streams:* Support a high-level stream language with built-in extensible stream-oriented primitives and operators.
- *Rule 3—Handle Stream Imperfections:* Built-in mechanisms to provide resiliency against stream "imperfections" including delayed, missing, and out-of-order data which occur in real-world data streams.
- *Rule 4—Generate Predictable Outcomes:* Guarantee predictable and repeatable outcomes.
- *Rule 5—Integrate Stored and Streaming Data:* Ability to efficiently store, access, and modify state information, and combine it with live streaming data.
- *Rule 6—Guarantee Data Safety and Availability:* Provide high availability and integrity of the data maintained.
- *Rule 7—Partition and Scale Applications Automatically:* Ability to distribute processing across multiple computing resources for incremental scalability.
- *Rule 8—Process and Respond Instantaneously:* Highly optimised, minimal-overhead execution engine to deliver a real-time response.

4.3 Fundamentals of Real-time Linked Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in the world. This is in part due to increased adoption of the Internet of Things and the use of sensors for improving data collection and monitoring of smart environments, which enhance different aspects of our daily activities in smart buildings, smart energy, smart cities, and others [1]. To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of knowledge among systems. A data platform can provide a clear framework to support the sharing of data among a group of

intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an "*as-needed*" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to setup data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental "*pay-as-you-go*" fashion with more detailed mappings among the required data sources.

Within the dataspace paradigm, there has been limited work on addressing the requirements of real-time processing of events and streams, and research into relevant support services. The Real-time Linked Dataspace (RLD) has been created as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support large-scale distributed heterogeneous collection of streams, events, and data sources [4]. This work builds on past efforts to use dataspaces in Building Data Management [62], Energy Data Management [100], and System of Systems [85]. The goal is to support a principled approach to incremental real-time data management based on a set of support services with tiered levels of support, to provide a unified entity-centric query framework over real-time and historical data streams in a smart environment.

This section details the foundations of the Real-time Linked Dataspace approach and describes how its architectural components meet the key requirements identified for real-time information processing (as identified by Stonebraker et al. [3]) and data platform for smart environments (as identified in Chap. 2).

4.3.1 Foundations

The Knowledge Value Ecosystem (KVE) framework (see Chap. 2) helps us to understand the challenge with knowledge flows at different levels within data ecosystems. At the knowledge exchange level, a data platform for a smart environment would need to overcome administrative and semantic barriers. Acknowledging that the core challenges within dataspaces are lack of administrative proximity, and loose semantic integration, the question becomes: how can we better tailor the principles of dataspaces, to real-time data processing? To answer this question, we look no further than the literature of the event processing paradigm itself. A core principle in event processing is decoupling, which refers to the lack of explicit agreements in order to increase scalability as defined by Eugster et al. [142]. Three

main dimensions have been recognised in the event processing literature with respect to decoupling:

- *Space Decoupling*, which means that event producers and consumers do not hold identifiers (e.g. IP addresses) of each other.
- *Time Decoupling*, which means that event producers and consumers do not have to be active at the same time.
- *Synchronisation Decoupling*, which means that event producers and consumers do not block each other when exchanging events.

These dimensions can be classified within the "administrative proximity" aspect of data management from Franklin et al. [2]. The decoupled nature of event-based systems reduces their administrative proximity. However, in terms of semantic integration, event-based systems currently require tight semantic integration. Hasan and Curry [151] identified forms of semantic integration based on Carlile's framework within event-based systems:

- *Syntactic Coupling:* The amount of agreement among participants in the event processing environment on the sharing and establishment of a common low-level syntax. This view has been established by Shannon and Weaver [38] in their communication theory, where syntax has the form of zeros and ones. They claim that once such a syntax is shared, accurate communication can be ensured, and the task becomes that of information processing rather than communication.
- *Semantic Coupling:* The amount of agreement among participants in the event processing environment on the mappings among symbols used in event messages and the meanings to which they refer.

Bringing both of these views together helps us to understand the challenge with knowledge flows within data ecosystems. Here we build on the concept of decoupling to meet the principles of dataspaces by Halevy et al. [78], as illustrated in Fig. 4.3. Here we can see that within an RLD the main administrative issues are around space, time, and synchronisation of interacting systems. While semantic integration is centred on syntactic and semantic concerns, this requires a Real-Time Linked Dataspace to support an event processing paradigm that supports many formats of data, does not depend on schema agreement, and supports a best-effort approximate and pay-as-you-go approach.

4.3.2 Definition and Principles

A Real-time Linked Dataspace is a specialised dataspace that manages and processes the large-scale distributed heterogeneous collection of streams, events, and data sources [4]. It manages the sources without presuming a pre-existing semantic integration among them, uses linked data and knowledge graphs to coordinate the dataspace, and operates under a 5 star model for "pay-as-you-go" data management (see Sect. 4.4).

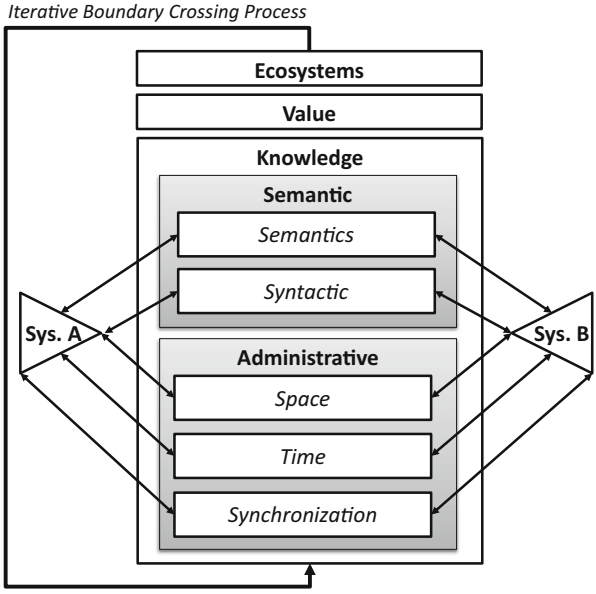

Fig. 4.3 Dimensions of decoupling for knowledge flows between event-based systems based on the KVE framework

The RLD adapts the dataspace principles as set out by Halevy et al. [78] to describe the specific requirements within a real-time dataspace setting:

- A Real-time Linked Dataspace must deal with many different formats of streams and events.
- A Real-time Linked Dataspace does not subsume the stream and event processing engines; they still provide individual access via their native interfaces.
- Queries in the Real-time Linked Dataspace are provided on a best-effort and approximate basis.
- The Real-time Linked Dataspace must provide pathways to improve the integration among the data sources, including streams and events, in a pay-as-you-go fashion.

In order to enable these principles to support real-time data processing, we propose a set of specialised dataspace support services to enable the requirements of loose administrative proximity and semantic integration for event and stream systems. Loose coupling of event processing systems on the semantic dimension reflects a low cost to define and maintain rules concerning the use of terms, and a low cost to building and agreeing on the event semantic model. This requirement forms the foundation of the techniques and models used to process events and streams within the Real-time Linked Dataspace.

Table 4.1 Comparison of DBMS, Dataspace, and a Real-time Linked Dataspace

	DBMS	Dataspace	Real-time Linked Dataspace
Data management requirements			
Model	Relational	All	All
Formats	Homogeneous	Heterogeneous	Heterogeneous
Schema	Schema first, data later	Data first, schema later or never	Data first, schema later or never
Control	Complete	Partial	Partial
Leadership	Top-down	Top-down/Bottom-up	Distributed
Query	Exact	Approximate	Approximate
Integration	Upfront	Incremental	Incremental
Architecture	Centralised	Decentralised	Distributed
Data processing	None	None	Real-time and batch processing
Real-time requirements			
Rule 1: Keep the data moving	No	Possible	Yes
Rule 2: High-level stream language	No	Possible	Yes
Rule 3: Handle stream imperfections	Difficult	Possible	Yes
Rule 4: Predictable outcome	Difficult	Possible	Possible
Rule 5: High availability	Possible	Possible	Possible
Rule 6: Stored and streamed data	No	Possible	Yes
Rule 7: Distribution and scalability	Possible	Possible	Possible
Rule 8: Instantaneous response	Possible	Possible	Yes

4.3.3 Comparison

While the initial vision of dataspaces encompassed the notion of support for data streams, the details of how to specifically handle streams within a dataspace were not covered in depth. The RLD goes beyond a traditional dataspace approach [2] by supporting the management of entities within the dataspace as first-class citizens along with data sources, and it extends the dataspace support platform with real-time processing and querying capabilities for streams and events as detailed in Table 4.1.

4.3.4 Architecture

The RLD contains all the relevant information within a data ecosystem, including things, sensors, and data sources and has the responsibility for managing the relationships among these participants. Figure 4.4 illustrates the architecture of the RLD with the following main concepts:

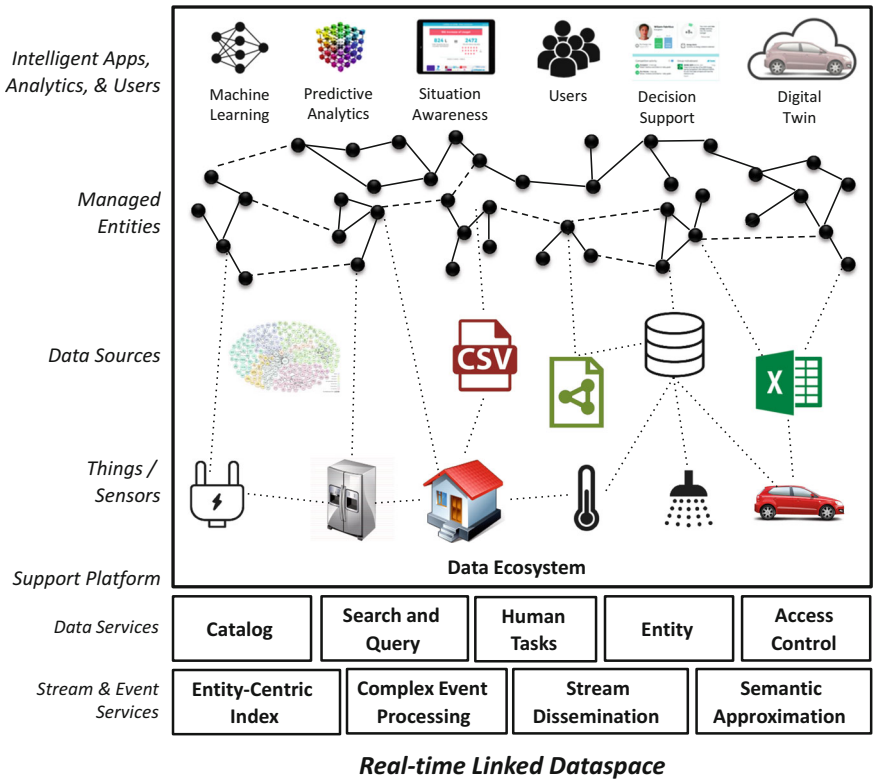

Fig. 4.4 Real-time Linked Dataspace architecture

- *Support Platform:* Responsible for providing the functionalities and services essential for managing the dataspace. Support services are grouped into data services and stream and event services.
- *Things/Sensors:* Produce real-time data streams that need to be processed and managed. Things in a smart environment range from connected devices, energy, and water sensors, to connected cars and manufacturing equipment.
- *Data Sources:* Available in a wide variety of formats and accessible through different systems interfaces. Example data sources include building management systems, energy and water management systems, passenger information systems, financial data, weather, and (linked) open datasets.
- *Managed Entities:* Actively managed entities (e.g. people, equipment, buildings) within the data ecosystem, including their relationship to participating things, data sources, and other entities in the RLD.
- *Intelligent Applications, Analytics, and Users:* Interact with the RLD and leverage its data and services to provide data analytics, decision support tools, user interfaces, and data visualisations. Applications/Users can query the RLD in an entity-centric manner, while users can be enlisted in the curation of the data and entities via the Human Task service.

4.4 A Principled Approach to Pay-As-You-Go Data Management

Within the RLD, the pay-as-you-go approach to data management is complemented with a principled tiered approach to the design of support services where an increase in the level of active data management has a corresponding increase in the associated effort [4]. This tiered approach to data management provides flexibility by reducing the initial effort and barriers to joining the dataspace. The tiers for the RLD are a specialisation of the 5 stars scheme defined by Tim Berners-Lee for publishing open data on the web [43].

4.4.1 TBL's 5 Star Data

The W3C Linking Open Data (LOD) project started in 2007 and began publishing datasets under open licenses and following the linked data principles. To encourage people to publish linked data, the inventor of the web and the initiator of the linked data paradigm, Tim Berners-Lee, proposed a 5 star rating system [41]. The rating system (see Fig. 4.5) helps data publishers to evaluate how much their datasets conform to the linked data principles. The first star is to make data available on the web, with each additional star corresponding to increased reusability and interoperability of the published data as more of the principles of linked data are followed.

Fig. 4.5 Tim Berners-Lee's 5 star rating system [41]

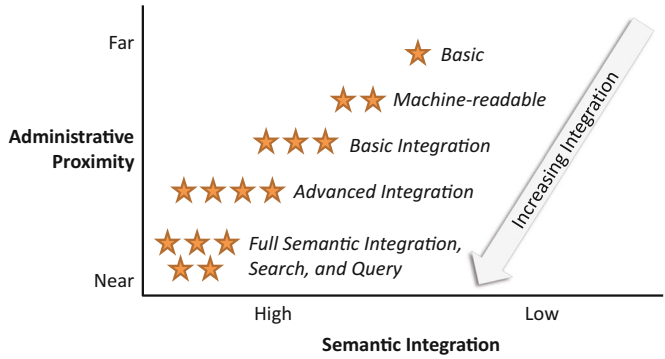

Fig. 4.6 The 5 star pay-as-you-go model of a Real-time Linked Dataspace

4.4.2 5 Star Pay-As-You-Go Model for Dataspace Services

In contrast to the classical one-time integration of datasets that causes a significant upfront overhead, the RLD adopts a principled pay-as-you-go paradigm for supporting an incremental approach to data management. At the foundation of the approach is the principle that the publisher of the data is responsible for paying the cost of joining the dataspace. This pragmatic decision allows the RLD to grow and enhance gradually with participants joining or leaving the dataspace at any time. The next principle is that data is managed following a tiered approach, where an increase in the level of active data management has a corresponding increase in associated costs.

The tiered approach to data management provides flexibility by reducing the initial cost and barriers to joining the dataspace. The tiers are described using a specialisation of the 5 star scheme defined by Tim Berners-Lee. The original star scheme has been extended to consider the level of integration of the data sources with the support services of a dataspace. At the minimum level, a data source needs to be made available with a dataspace. Over time, the level of integration with the support services can be improved in an incremental manner on an as-needed basis. The more the investment made to integrate with the support services, the better is the integration achievable in the dataspace. The 5 star pay-as-you-go model for the RLD is illustrated in Fig. 4.6. The five tiers are:

1 Star **Basic:** The participant is published in the dataspace with limited or no integration with support services.

2 Stars **Machine-readable:** The participant is publishing data in a machine-readable format. This enables services to provide a minimal level of support with basic functionality (e.g. browsing) where available basic interfaces are exposed.

3 Stars **Basic integration:** The use of a non-proprietary format enables support services to provide essential services at the data-item/entity level with support for simple functionality (e.g. keyword search).

4 Stars **Advanced integration:** The participant is integrated with most support service features (e.g. structured queries) with an awareness of its relationships to other participants with basic support for federation.

5 Stars **Full semantic integration, search, and query:** The participant is fully integrated into the support services (e.g. question answering) and linked to relevant participants. It plays its full role in the global view of the dataspace.

4.5 Support Platform

The RLD-Support Platform (RLD-SP) provides a set of core services to support intelligent application developers and data scientists with a base functionality when working with sources in the RLD. Each of the services in the RLD-SP has been designed to follow the pay-as-you-go paradigm to support varying levels of service offerings to the participants in the dataspace. Two categories of support services have been developed, one targeting core data management and the other focuses on support for streams and events. This section details these services and their tiered levels of support as detailed in Table 4.2.

4.5.1 Data Services

The RLD-SP provides a set of enhanced data support services to enable all participants in the dataspace to get setup and running with a low overhead. These support services are built on the core support services defined by Franklin et al. and extended to follow the entity-centric data management approach of knowledge graphs [2]. The services have been designed to include support for linked data and follow the 5 star pay-as-you-go model. Examples of data services include the catalog, entity management, search and query, and data service discovery. Part II of this book further explores these data services.

4.5.2 Stream and Event Processing Services

The RLD supports real-time data processing with support services that follow the data management philosophy of dataspaces. The RLD-SP provides support services to handle the processing of streaming and event data tackling issues including entity-centric queries, complex event processing, stream dissemination, and semantic approximation. The goal of these services is to support participants in the RLD to get setup and running with a low overhead for administrative setup costs (e.g. establishing data agreements, service selection, and composition). Part III of this book further explores the stream and event support services.

Table 4.2 The 5 star pay-as-you-go scheme for the Real-time Linked Dataspace support services [4]

Pay-as-you-go star rating	Data format	Catalog	Access control	Search and query	Entity	Human tasks	Entity-centric index	Complex event processing	Stream dissemination	Semantic approximation
* Basic	Any format (e.g. PDF)	Registry of datasets and streams	None	Browsing	None	None	None	None	None	None
** Machine-readable	Machine-readable structured data (e.g. Excel)	Non-machine-readable metadata document (e.g. PDF)	Coarse-grained (Dataset level)	Keyword search	Entities identifiers in documentation	Schema mapping	Basic processing	Single stream	None	Semantic matching
*** Basic integration	Non-proprietary format (e.g. CSV, JSON, XML)	Machine-readable metadata Equivalence among schema concepts	Fine-grained (Entity-level) Secure query service	Structure search	Source level (siloed)	Entity mapping	Historical views of streams	Multi-service composition	Point-to-point	Thematic matching
**** Advanced integration	Uses the first two principles of linked data	Relations among schemas (dataspace level)	Data anonymisation	Structured queries	Canonical identifiers and entity mappings across sources	Entity enrichment	Stream enrichment with context and entity data	Quality of service aware service composition	Wireless broadcast	Entity-centric matching
***** Full semantic integration, search, and query	Follows all publishing principles of linked data	Full semantic mappings	Usage control	Schema-agnostic question answering	Knowledge graphs semantically link entities to related entities, data, and streams	Data quality improvement	Entity-centric real-time query	Context-aware	Complex patterns	Context-aware

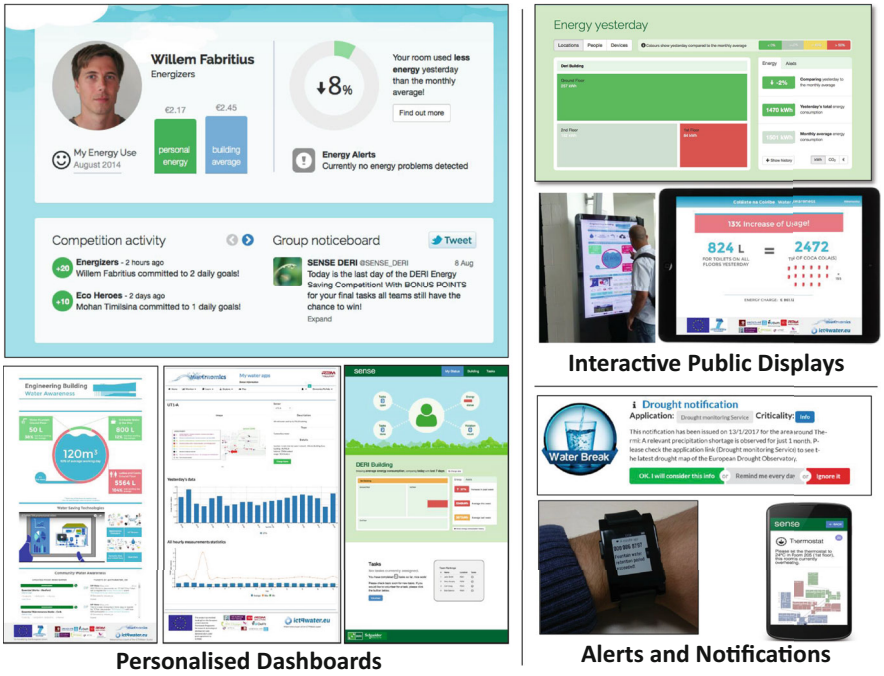

Fig. 4.7 Intelligent systems and applications built using the Real-time Linked Dataspace [16]

4.6 Suitability as a Data Platform for Intelligent Systems Within IoT-Based Smart Environments

The RLD has been used as a data platform to support the development of intelligent systems and applications within a range of smart environments including smart home, school, office building, university, and airport [16]. Within these environments, a data platform needs to support a wide range of end-users with different interests and priorities from corporate managers looking for data to improve the performance of their business to software engineers developing applications for the smart environment (see Fig. 4.7). Intelligent systems and applications developed using the RLD are discussed in detail in part IV of this book. In this section, we examine the suitability of the RLD as a data platform for a smart environment.

4.6.1 Common Data Platform Requirements

The goal of the RLD is to support a principled approach to incremental real-time data management based on a set of services with tiered levels of support within a smart

environment. The FAIR principles (Findability, Accessibility, Interoperability, and Reusability) are designed to enable good data management to support knowledge discovery and innovation, and the subsequent data and knowledge integration and reuse [61]. The principles are described further in Chap. 2. Within the context of this work, the principles can serve as a high-level guide for the design of a data platform to support knowledge sharing within a smart environment.

Section 2.4 identified a set of common data management requirements for a data platform. The common data platform requirements are [4]:

- *Standard Data Syntax, Semantics, and Linkage:* Facilitate integration and sharing, ideally with open standards and non-proprietary approaches.
- *Single-Point Data Discoverability and Accessibility:* Allow the organisation and access to datasets and metadata through a single location.
- *Incremental Data Management:* Enable a low barrier to entry and a pay-as-you-go paradigm to minimise costs.
- *Secure Access Control*: Support data access rights to preserve the security of data and privacy of users in the smart environment. Access control is needed at both the level of the data source and at the level of the data item (i.e. entity-value).
- *Real-time Data Processing:* Including ingestion, aggregation, and pattern detection within event streams originating from sensors and things in the smart environment.
- *Unified Querying of Real-time Data and Historical Data:* Provide applications and end-users with a holistic queryable state of the smart environment at a latency suitable for user interaction.
- *Entity Management:* The storage, linkage, curation, and retrieval of entity data, such as users, zones, and locations.
- *Event Enrichment:* Enhancement of sensor/things streams with contextual data (e.g. entities) to make the stream data more encapsulated and useful in downstream processing.

These requirements can be used to survey the capabilities of existing approaches for data platforms within a smart environment and to highlight the main contribution of the RLD.

4.6.2 Related Work

The CityPulse [65] project provides a distributed system for semantic discovery, data analytics, and interpretation of large-scale and near-real-time Internet of Things data and social media data streams [14]. In addition to providing unified views of the data, CityPulse also provides data analytics modules that perform intelligent data aggregation, event detection, quality assessment, contextual filtering, and decision

support. CityPulse supports open standards for semantics, real-time stream processing, and entity management. However, no support exists for single-point data access, a pay-as-you-go data management paradigm, unified views over real-time and historical data, security, and event streams enrichment.

The OpenIoT [66] platform enables the semantic interoperability of IoT services in the cloud through the use of the W3C Semantic Sensor Networks (SSN) ontology [152], which provides a common standards-based model for representing physical and virtual sensors. OpenIoT provides middleware for uniform access to IoT data and support for the development and deployment of IoT applications. OpenIoT supports open standards for semantics, real-time stream processing, security, and entity management. However, it lacks support for single-point data access, a pay-as-you-go data management paradigm, unified views over live and historical data, and event streams enrichment.

The SmartSantander project developed the City Data and Analytics Platform (CiDAP) [33], a centralised platform to access data generated from multiple heterogeneous sensors installed in a city. The platform can deal with historical data and near real-time information in an architecture like Lambda. CiDAP provides limited support for data management beyond the low-level sensor streams and pushes these concerns to the application-level. The result is applications duplicating common data management functionalities. SmartSantander follows open standards for semantics, single-point data access, security, real-time stream processing, and partial unified queries over streams and datasets. However, it lacks support for an incremental data management paradigm, entity management, or event streams enrichment.

The Spitfire [64] project uses semantic technologies to provide a uniform way to search, interpret, and transform sensor data. Spitfire works towards a Semantic Web of Things, by providing abstractions for things, basic services for search and annotation, as well as by integrating sensors and things into the LOD cloud. Spitfire adopts semantic web standards for describing data, partial secure access control, entity management, and event enrichment. It does not support single-point access for data, incremental data management, real-time data processing, or unified queries for real-time and legacy data.

ThingStore [153] provides a "marketplace" for IoT applications development with the ability to deploy and host them. The platform provides support for event detection, service discovery, an Event Query Language, together with event notification and management. The architecture of ThingStore is a computation hub to connect things, software, and end-users. ThingStore supports secure and real-time data processing. However, it lacks support for open standards to describe data, single-point access for data, entity management and event enrichment, incremental data management, and unified queries for real-time and legacy data.

From the analysis in Table 4.3, we note that existing data platforms support semantic descriptions of data according to open standards such as semantic web and linked data. However, they lack an incremental data management paradigm and do

Table 4.3 Comparison of related frameworks to common data platform requirements [4]

Requirements	City pulse [65]	Open IOT [66]	SmartSantander [33]	Spitfire [64]	ThingStore [153]	Real-time Linked Dataspace
Standard data syntax, semantics, and linkage	Yes	Yes	Partial	Yes	No	Yes
Single-point data discoverability and accessibility	No	No	Partial	No	No	Yes
Incremental data management	No	No	No	No	No	Yes
Secure access control	No	Yes	Yes	Partial	Partial	Yes
Real-time data processing	Yes	Yes	Yes	No	Yes	Yes
Unified querying of real-time data and historical data	No	No	Partial	No	No	Yes
Entity management	Partial	Yes	No	Yes	No	Yes
Event enrichment	No	No	No	Partial	No	Yes

not support a single access point to discover and access datasets. Most data platforms address the real-time processing of data but do not provide unified access to it along with historical data. Half of the data platforms provide some support for entity management. However, streams are not typically enriched with contextual data. Based on the analysis of the requirements identified we could see there is a clear need for an incremental pay-as-you-go data management, a single point of data/stream access, support for entity-centric views of real-time and historical data, and streams enrichment for better entity-centric and contextual data retrieval.

4.7 Summary

Real-time Linked Dataspaces (RLD) enable data ecosystems for intelligent systems within Internet of Things-based smart environments by providing a principled approach to the incremental management of stream events that can reduce the technical and conceptual barriers to information sharing. This chapter introduces the Real-time Linked Dataspace that combines the pay-as-you-go paradigm of dataspaces and linked data with real-time search and query capabilities. The chapter details the fundamentals of the RLD and its basis in the stream and event processing,

and dataspace communities. The design of the RLD is detailed, including the main components of the architecture, the 5 star model for pay-as-you-go support services including services for streams, events, and data. Finally, a high-level overview of the suitability of the RLD for a data platform is provided with a comparison to related platforms.