raddec: Elevating IoT Interoperability Through a Common Radio Decoding Data Format

Jeffrey Dungen
reelyActive
Montréal, Québec, Canada
jeff@reelyactive.com

***Abstract*—Whereas the interoperability of radio-frequency devices at the hardware layer of the IoT is afforded by wireless standards, equivalent standards are largely absent for the digital representation of the data which is propagated to the software and application layers above. The raddec is presented as a common representation of a radio decoding for human-scale RFID technologies such as RAIN RFID and Bluetooth Low Energy, fostering interoperability between the hardware and software layers of the IoT and beyond. Implemented as an open source library, the raddec serves as the core data structure for middleware that enables vendor-agnostic and technology-agnostic software development for any IoT application. Used actively in both industry and research, the raddec continues to evolve to support emerging IoT technologies, products and features. The proven utility of the raddec suggests that a formal standard for the representation of radio decodings would greatly benefit the IoT ecosystem. In the absence of such a standard, developers and researchers are invited to freely use the raddec in their own work and to contribute openly to its continuous evolution.**

***Index Terms*—IoT, interoperability, radio decoding, middleware, RFID, raddec**

I. Introduction

This year marks the 25th anniversary of the MIT Auto-ID Center [1], and the coining of the term "Internet of Things" (IoT) by its co-founder and executive director, Kevin Ashton [2]. In 1999, barcodes were the predominant automatic identification and data capture (AIDC) technology [3], with radio-frequency identification (RFID) technologies on the horizon.

Today, in 2024, RFID technologies have become pervasive in our daily lives. There are currently 44.8 billion RAIN RFID (UHF passive RFID) *things* [4], and 5 billion Bluetooth Low Energy (2.4GHz active RFID) *things* [5] shipping annually. Through economies of scale, and a vibrant ecosystem, wireless technologies such as these advance the realisation of the vision of the IoT where computers can make sense of the physical world—without the need for human-entered data.

However, while standardisation and proliferation of RFID techologies has fostered hardware interoperability such that tags, readers and gateways can be used interchangeably, we at reelyActive have *not* observed convergence towards a data standard at the subsequent level: the interface between hardware and software. Instead, we continue to observe a pattern of vendor-specific and technology-specific data formatting across the growing number of commercial gateways and readers. In other words, the format of radio data over-the-air is standard while the format of its digital decoding is not, to the detriment of interoperability at the software and application layers above.

In 2018, we at reelyActive undertook development of a vendor-and-technology-agnostic digital representation of a radio decoding as a software library, named raddec, to serve as the core data structure for a new version of our open source IoT middleware. In this paper, we will present the motivations and considerations for the development of the raddec data format, examine how radio decodings are represented today, detail the raddec, its novel features and its evolution, and present middleware implementations which provide the foundation for successful applications in both research and industry.

II. Motivations

Like the Internet, the Internet of Things can be envisioned as a framework of layers which, together, simultaneously support a diverse array of applications. Figure 1 illustrates such a layered framework in the context of this research, contrasting the current state of fragmentation with one of fluid interoperability at the software and application layers which, we will argue, can be achieved through a common representation of a radio decoding across technologies and vendors. Our motivations, specifically, are as follows.

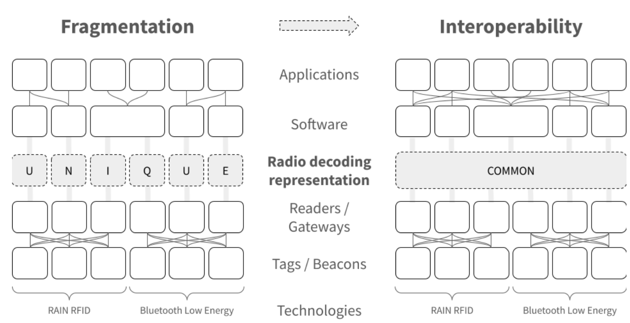

Fig. 1. The IoT modeled as a layered framework.

A. Complementarity of Applications

Complementary IoT applications enabled by RFID technologies broadly include identifying, counting, locating, tracking and sensing *things*. Different applications require different data, and so, to maximise the utility of any IoT deployment—*which is often based initially on the merits a single application or business case*—it is therefore imperative for the radio decoding data to embody all the properties necessary to support the broadest possible range of anticipated applications.

B. Pervasiveness of Infrastructure

In 2013, we published a paper entitled "Towards a Simple, Versatile, Distributed Low-Power Wireless M2M Infrastructure" [6] addressing the challenge of deploying hardware infrastructure at a scale necessary to realise a ubiquitous IoT. Fortunately, that infrastructure challenge has since transformed into an opportunity: WiFi access points (APs), which are widely deployed in public, commercial and industrial spaces, have increasing capability to double as IoT gateways for BLE and/or IEEE 802.15.4 low-rate wireless personal area networks, for example [7] and [8]. In other words, where there is WiFi, there can be IoT. To maximise the utility of this infrastructure for the IoT, the radio decoding data it provides should be represented in a common format.

C. Coexistence of RFID Technologies

In 2017, we published a paper entitled "Towards collective hyperlocal contextual awareness among heterogeneous RFID systems" [11] presenting the emerging opportunities of complementary technologies, co-located at a human scale of about 10 metres, such as RAIN RFID and BLE. The continued proliferation of these technologies, as described in the introduction, reinforces such opportunities today. To maximise the IoT potential of coexisting technologies, the data format in which the decoding of their radio transmissions is encoded should be sufficiently common so as to enable applications for which those technologies can be used interchangeably to the extent possible.

D. Scale of Data

It is not uncommon for a BLE sensor to transmit one packet (or more!) every second. Multiply that by the number of receiving gateways in range to obtain the number of radio decodings per second. And then multiply that by the number of bytes used to represent a radio decoding, and so on... The scale of the radio decoding data from billions and billions of *things*, including transport, processing and storage, has the potential to overwhelm existing networks, computers and databases. A compact, efficient representation of a radio decoding has the potential to significantly optimise IoT operations for scaleability.

E. Scarcity of Research

A literature review suggests that there is scarce research on interoperability at the level of the radio decoding, with research instead focused on adjacent interoperability topics such as IoT data transport [9] and IPv6 using RFID [10]. This work is intended to motivate further research in this domain, which merits attention for the aforementioned reasons.

III. Anatomy of a Radio Decoding

A radio decoding may be defined as the digital representation of a received radio transmission. Figure 2 illustrates the basic elements of a radio decoding, with respect to both active and passive technologies, which we analyse in further detail next, in order to identify common properties.

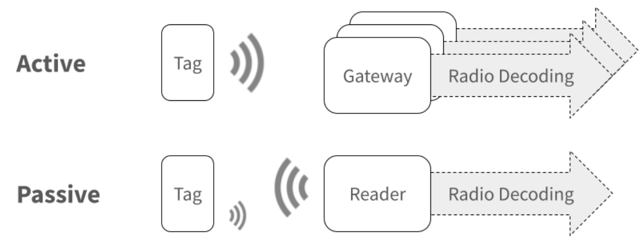

Fig. 2. Radio decoding comparing active and passive technologies.

A. Active Radio Decoding

Active technologies, such as BLE, support the spontaneous transmission of radio packets by devices using energy that is self-sourced, often from a battery. Any peer device in range which is listening on the same channel may receive and decode these radio packets. As a result, there may be multiple receivers, and hence radio decodings, of a single radio transmission.

Additionally, devices may retransmit radio packets periodically and/or on different radio channels (ex: BLE specifies 3 advertising channels) for redundancy, resulting in additional radio decodings.

In the case of BLE, an advertising packet, which a device may spontaneously transmit, includes a 48-bit advertiser address (which identifies the transmitting device) and an optional payload.

In addition to these properties, any receiving device can typically contribute the following radio decoding metadata:

- Receiver identifier
- RSSI (received signal strength indicator)
- Timestamp (provided it has a real-time clock)

Specialised receiving devices may contribute additional metadata, for example to support angle of arrival and/or departure (AoA/AoD) determination based on the Bluetooth Asset Tracking Profile [12], or distance estimation based on the draft Bluetooth Channel Sounding specification [13], which may be used to estimate the location of the transmitting device.

An active radio decoding therefore includes identification of the transmitting and receiving devices, an optional transmitted payload, as well as decoding metadata provided by the receiving device.

B. Passive Radio Decoding

Passive technologies, such as RAIN RFID, rely on energy provided by a reader, in the form of radio waves, to tags which backscatter data from their memory banks as radio packets which can then be decoded by the reader.

A RAIN RFID tag (also called endpoint) includes a tag identifier (TID) which is a unique serial number, and an electronic product code (EPC) which is a unique tracking identifier. An EPC may be assigned to more than one tag, and therefore the TID is the only truly unique identifier of a radio decoding of a specific tag. The tag may also include relevant information in additional memory banks.

In addition to data read from the tag memory banks, any reader can typically contribute the following radio decoding metadata:

- Reader identifier
- Antenna identifier (where multiple antennas are supported)
- RSSI (received signal strength indicator)
- Number of reads (within a specific time window)
- Timestamp (provided it has a real-time clock)

Specialised readers may contribute additional metadata, such as a 2D or 3D position estimation, which may be achieved using antenna arrays, for example [14] and [15].

A passive radio decoding therefore includes identification of the tag and reader, optional additional memory bank data, as well as decoding metadata provided by the reader.

C. *Common Properties*

Based on the preceding analysis, active and passive radio decodings do indeed share a common set of properties which include transmitter and receiver identifiers, and standard decoding metadata such as RSSI and the number of decodings within a given period of time. The radio decodings may include payload or memory bank data, as well as additional decoding metadata which is typically at the service of real-time location or positioning algorithms.

In the next section we will examine existing implementations of radio decoding data representations before presenting our own raddec implementation in the section following.

IV. Radio Decoding Implementations

In order to propagate received wireless data, gateways and readers must represent radio decodings in some digital format. In the case of RAIN RFID, there are two vendor-agnostic specifications, however in the case of BLE, to the best of our knowledge, there is no such specification, but rather only vendor-specific implementations.

A. *LLRP*

The Low Level Reader Protocol (LLRP), maintained by GS1, provides a common interface for controlling RFID readers, with the laudable promise that "if middleware or application software uses the LLRP interface, portability will be increased" [16].

Specifically, LLRP represents radio decodings as a binary RO_ACCESS_REPORT which includes a TagReportData parameter which optionally includes equivalents to many of the common properties identified in the previous section, specifically:

- EPCData
- AntennaID
- PeakRSSI
- TagSeenCount
- LastSeenTimestampUTC

Nonetheless, as LLRP is technology-specific, it cannot be considered a viable candidate for a common radio decoding format.

B. *RCI*

The RAIN Communication Interface (RCI), maintained by the RAIN Alliance, also provides a common interface for controlling readers [17].

Specifically, RCI represents radio decodings as JSON in a Tag Spot Report, which optionally includes equivalents to many of the common properties identified in the previous section, taking, for example, the following form:

```
{
  "Report": "TagEvent",
  "Scheme": "SGTIN",
  "EPC": ":3003:4567:89AB...",
  "Ant": 1,
  "RSSI": -99,
  "InvCnt": 123,
  "TimeStamp": 1343392496789

}
```

Like LLRP, RCI is also technology-specific, and, moreover, we are not aware of a single commercial reader which currently supports the protocol. As such, it is not a viable candidate for a common radio decoding format.

C. *Vendor-Specific Implementations*

In the absence of a shared standard or even a technology-specific standard, vendor-specific implementations of a radio decoding data format become commonplace. Table I lists a variety of such implementations and their key characteristics.

TABLE I
Vendor-Specific Radio Decoding Implementations

Vendor	Protocol name	Technology	Encoding
HPE Aruba	Telemetry WebSocket	BLE, Serial[a]	Protobuf
HPE Aruba	IoT Operations	BLE, Serial[a]	Protobuf
CSL	Low Latency Alert	RAIN RFID	JSON
Impinj	IoT Device Interface	RAIN RFID	JSON
Huawei	Transparent	BLE	Binary
Minew	JSON-Long	BLE	JSON
Minew	Binary-Long	BLE	Binary
RF Controls	RFC OS WebSocket	RAIN RFID	JSON

[a]Supports EnOcean Alliance and proprietary wireless protocols.

In our experience, even vendor-specific implementations across products or software generations may not be interoperable, and therefore none are viable candidates for a common radio decoding format either.

V. raddec

We now present the raddec as a technology-agnostic, vendor-agnostic and application-agnostic representation of a radio decoding, implemented as a MIT-licensed open source library, written in JavaScript for Node.js [18].

The raddec supports both a JSON and binary representation, with the library providing methods to convert between either format. As an example, the decoding of an Apple iBeacon BLE advertising packet by a single receiver would be represented in JSON as follows:

```
{
  "transmitterId": "a441e1beac07",
  "transmitterIdType": 3,
  "rssiSignature": [{
      "receiverId": "0123456789ab",
      "receiverIdType": 2,
      "rssi": -99,
      "numberOfDecodings": 3
  }],
  "packets": [ "402407acbee141a40201061aff4c000215..." ],
  "timestamp": 1711934625678
}
```

with a binary equivalent (in hexadecimal) as follows:

```
10004503a441e1beac07011c03020123456789abf0018e \\
9741578ef10126402407acbee141a40201061aff4c0002 \\
1500112233445566778899aabbccddeeff01234567fced
```

For comparison, the BLE advertising packet is 38 bytes long, the binary raddec is 69 bytes long and the JSON representation is 271 bytes long (with whitespace removed).

A. *raddec Properties*

The raddec is comprised of a number of required and optional properties. Table II lists these properties and indicates how each is represented in both JSON and binary. The rssiSignature property of the raddec, which contains at least one element, is detailed in Table III. An online developer cheatsheet [19] outlines the raddec properties in greater detail.

TABLE II
REQUIRED AND OPTIONAL RADDEC PROPERTIES

Property name	JSON Type	Binary Type	Req?
transmitterId	String (hex)	Number	Yes
transmitterIdType	Number	8-bit number	Yes
rssiSignature	Array of Object	See Table III	Yes
timestamp	Number	32-bit number	No
packets	Array of String	n x raw	No
events	Array of Number	8-bit flag	No
position	Array of Number	3 x 64-bit float	No

TABLE III
RSSISIGNATURE ELEMENT PROPERTIES

Property name	JSON Type	Binary Type	Req?
receiverId	String (hex)	Number	Yes
receiverIdType	Number	8-bit number	Yes
rssi	Number	8-bit number	Yes
numberOfDecodings	Number	8-bit number	Yes
receiverAntenna	Number	8-bit number	No
aoa	Array of Number	2 x 8-bits	No

B. *raddec Identifier Types*

In addition to the transmitter and receiver identifiers themselves, an identifier type property is included for each. This property serves to distinguish devices with identical identifiers of different types (ex: 96-bit TID vs. 96-bit EPC). It can also be used by software to simplify the filtering of raddec instances by class or type. Table IV enumerates the raddec identifier types implemented to date.

TABLE IV
ENUMERATION OF RADDEC IDENTIFIER TYPES

Index	Type	Used by
0	Unknown	n/a
1	EUI-64	RAIN RFID readers, reelyActive, ...
2	EUI-48	BLE, WiFi
3	Random 48-bit	BLE (non-public addresses)
4	96-bit TID	RAIN RFID
5	96-bit EPC	RAIN RFID
6	128-bit UUID	RF Controls, ...
7	32-bit EURID	EnOcean Alliance
8-255	n/a	Reserved for future use

C. *raddec Event Types*

The optional events property is an index list of event types, which are listed in Table V. This property, which has no direct equivalent among the vendor-specific implementations examined in this research, can be used for event-driven radio decoding representation. In other words, rather than representing every radio decoding with a raddec, instead, a raddec may only be generated when a specific event occurs, for example if a new packet is decoded, as illustrated in Figure 3. It can also be used to simplify the filtering of raddec instances by events of interest.

TABLE V
ENUMERATION OF RADDEC EVENT TYPES

Index	Type	Description
0	APPEARANCE	New transmitter decoded
1	DISPLACEMENT	rssiSignature order changed
2	PACKETS	packets changed
3	KEEPALIVE	Periodic update
4	DISAPPEARANCE	No longer decoded
5-7	n/a	Reserved for future use

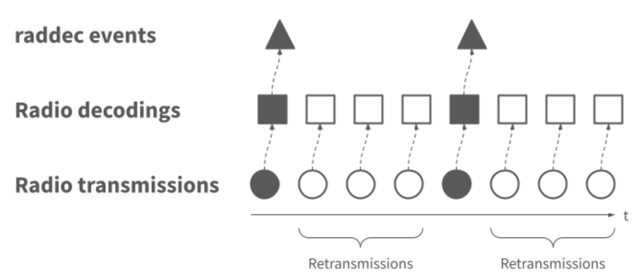

Fig. 3. Event-driven radio decoding representation.

D. *raddec Library Methods*

The raddec library provides methods to convert between JSON and binary representations, as well as to merge together raddec representations from the same transmitter. For a complete listing of the code and methods, see [18].

In the case of a raddec merge with common receivers, the numberOfDecodings property serves the additional purpose of facilitating the calculation of the RSSI for each receiver as a weighted average.

VI. Middleware Implementations

IoT middleware may be defined as software residing between the hardware and software layers of the IoT, providing services and functionalities which facilitate the exchange of data between the two. At reelyActive, we have, since 2018, adopted the raddec as a core data structure for the open source middleware modules we develop and maintain, namely the barnowl family of modules and the Pareto Anywhere IoT middleware suite.

A. barnowl-x Modules

The barnowl family of middleware modules [20] provides a unidirectional interface between hardware capable of relaying radio decodings (in some specific format) and software consuming radio decodings (in the raddec format). A barnowl instance can support any combination of barnowl-x instances, each of which provides an interface to consume radio decodings in a specific format. Figure 4 illustrates this hierarchy, and Table VI lists the modules implemented to date.

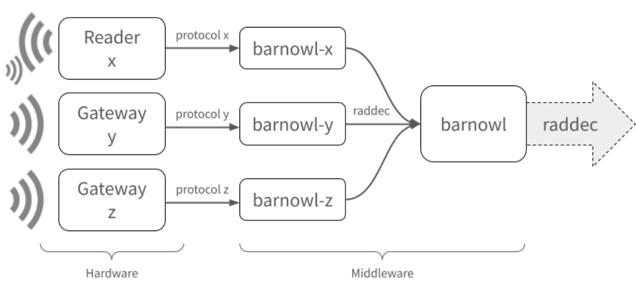

Fig. 4. Vendor-and-technology-agnostic middleware using barnowl modules.

TABLE VI
Alphabetical list of barnowl-x modules implemented to date

Module	Hardware	Technologies
barnowl-aruba	HPE Aruba APs	BLE, EnOcean
barnowl-chafon	Chafon readers	RAIN RFID
barnowl-csl	CS463 reader	RAIN RFID
barnowl-enocean	EnOcean USB dongle	EnOcean
barnowl-hci	Bluetooth HCI devices	BLE
barnowl-huawei	Huawei APs	BLE
barnowl-impinj	Impinj R7x0	RAIN RFID
barnowl-laird	IG60-BL654	BLE
barnowl-llrp	LLRP readers	RAIN RFID
barnowl-minew	Minew gateways	BLE
barnowl-reel	reelyActive reelceivers	BLE, Proprietary
barnowl-rfcontrols	RFC OS	RAIN RFID
barnowl-tcpdump	WiFi adapters	WiFi

B. Pareto Anywhere

Pareto Anywhere [21] is an IoT middleware suite which includes a commonly used set of barnowl-x modules and maintains a machine-readable representation of "who/what is where/how" in the form of hyperlocal context [22]. It also includes the advlib libraries [23] which decode payload data (from the packets property of the raddec) into JSON with a standard set of properties, further contributing to vendor-agnostic IoT interoperability [24].

Pareto Anywhere includes web applications, which consume raddec data, to facilitate the visualisation and interpretation of hyperlocal context. For this purpose, we also develop and maintain beaver.js [25] as an open source client-side module which retrieves and manages raddec data in browser memory.

VII. Evolution and Adoption

Over five years have elapsed since the raddec software library was first created and openly published on GitHub in 2018. During that time, the raddec has evolved to support new technologies and features, and has been adopted in industry and research for countless applications.

A. Evolution of the raddec

The initial version of the raddec was designed to support BLE and reelyActive's own proprietary active RFID technology, with a view to support additional active and passive radio technologies. Support for RAIN RFID and EnOcean Alliance (active sub-GHz) technologies, as well as support for location/positioning decoding metadata, was successfully added to both the JSON and binary representation of the raddec without breaking changes. In other words, software developed using the initial version of the raddec remains compatible with the current version today.

B. Adoption in Industry

Given that the raddec library, barnowl-x modules and Pareto Anywhere middleware are freely-available open source code, it is impossible to know the extent to which they are adopted and used in practice. Nonetheless, we are aware of applications for asset and personnel tracking, environmental sensing, anonymous occupancy estimation and interaction detection by developers, integrators and end users around the world. Users often share with us that they selected our raddec-based middleware after experiencing firsthand the limitations of vendor-specific implementations.

A deployment of our raddec-based middleware has been in continuous operation at our office since 2020, collecting data from reelyActive BLE reelceivers, HPE Aruba APs, Minew gateways, Raspberry Pis (via HCI) and, more recently, RAIN RFID readers, all using a single data format. The raddec data is used for real-time dashboards [26] and for storage and analytics using the Elastic Stack which we have documented extensively [27].

C. Adoption in Research

A deployment of our raddec-based middleware has been in continuous operation at the GreenUXLab [28] of ESG-UQAM since 2019, collecting data from reelyActive BLE reelceivers and Owl-in-One gateways, and more recently, RF Controls smart antennas which support 3D location of RAIN RFID tags. Serving as an experimental retail store, the lab facilitates research in the tracking of items, assets and personnel, as well as environmental and occupancy sensing, allowing the side-by-side comparison, and combination, of active and passive technologies, using a single data format.

Our raddec-based middleware has also been used by several groups of researchers for the automated capture of peer-to-peer interactions in early childhood education [29] [30]. Among the advantages presented, the raddec affords the researchers the ability to change gateways without having to change their application software. In the case of one research group, this is especially valuable as the gateways used in their previous studies have reached their end-of-life and will be replaced to support ongoing studies.

VIII. Conclusion

In this paper we presented the raddec as a common representation of a radio decoding for human-scale RFID technologies such as RAIN RFID and Bluetooth Low Energy: a novel means to foster interoperability between the hardware and software layers of the IoT. Implemented as an open source library, the raddec serves as the core data structure for middleware that enables vendor-agnostic and technology-agnostic software development for any IoT application. Used actively in both industry and research, the raddec continues to evolve to support emerging IoT technologies and products, and their features. The proven utility of the raddec suggests that a formal standard for the representation of radio decodings, which is arguably overdue, would greatly benefit the IoT ecosystem. In the absence of such a standard, we invite anyone to use the raddec and to contribute to its continuous evolution.

References

[1] "MIT AUTO-ID LABORATORY", MIT. [Online]. Available: https://autoid.mit.edu/

[2] K. Ashton. "That Internet of Things Thing", RFID Journal, USA. [Online]. Available: https://www.rfidjournal.com/that-internet-of-things-thing

[3] "AIM 1990", AIM Global. [Online]. Available: https://www.aimglobal.org/aim-1990/

[4] "Shipment of RAIN RFID Tag Chips Surged to 44.8 Billion in 2023", RFID Journal. [Online]. Available: https://www.rfidjournal.com/shipment-of-rain-rfid-tag-chips-surged-to-44-8-billion-in-2023

[5] "2024 Market Update", Bluetooth SIG. [Online]. Available: https://www.bluetooth.com/2024-market-update/

[6] J. Dungen, T. Antonescu, and P-O. Genest, "Towards a Simple, Versatile, Distributed Low-Power Wireless M2M Infrastructure," *2013 IEEE 38th Conf. on Local Computer Networks (LCN)*, Sydney, Australia, 2013, pp.890-895, 21-24.

[7] "Access Points", HPE Aruba Networking. [Online]. Available: https://www.arubanetworks.com/products/wireless/access-points/

[8] "WLAN", Huawei Enterprise. [Online]. Available: https://e.huawei.com/en/products/wlan

[9] N. Nikolov, "Research of the Communication Protocols between the IoT Embedded System and the Cloud Structure," *2018 IEEE XXVII International Scientific Conference Electronics - ET*, Sozopol, Bulgaria, 2018, pp.1-4.

[10] I. Farris, S. Pizzi, M. Merenda, A. Molinaro, R. Carotenuto and A. Iera, "6lo-RFID: A Framework for Full Integration of Smart UHF RFID Tags into the Internet of Things," in *IEEE Network*, vol. 31, no. 5, pp. 66-73, 2017.

[11] J. Dungen and J. P. Neto, "Towards collective hyperlocal contextual awareness among heterogeneous RFID systems", *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, San Francisco, CA, USA, 2017, pp. 1-6.

[12] "Asset Tracking Profile", Bluetooth SIG. [Online]. Available: https://www.bluetooth.com/specifications/atp-1-0/

[13] "Channel Sounding", Bluetooth SIG. [Online]. Available: https://www.bluetooth.com/specifications/specs/channel-sounding-cr-pr/

[14] "Smart Antennas", RF Controls LLC. [Online]. Available: https://rf-controls.com/smart-antennas/

[15] "Impinj RAIN RFID Gateways", Impinj. [Online]. Available: https://www.impinj.com/products/readers/impinj-gateways

[16] "Low Level Reader Protocol", GS1. [Online]. Available: https://www.gs1.org/standards/epc-rfid/llrp/1-1-0

[17] "RAIN Communication Interface", RAIN Alliance. [Online]. Available: https://rainrfid.org/what-is-rain-rfid/readers/rain-communication-interface-rci/

[18] "reelyactive/raddec", GitHub. [Online]. Available: https://github.com/reelyactive/raddec

[19] "Cheatsheet", reelyActive Developers. [Online]. Available: https://reelyactive.github.io/diy/cheatsheet/

[20] "reelyactive/barnowl", GitHub. [Online]. Available: https://github.com/reelyactive/barnowl

[21] "reelyactive/pareto-anywhere", GitHub. [Online]. Available: https://github.com/reelyactive/pareto-anywhere

[22] J. Dungen and P-O. Genest, "Hyperlocal Context to Facilitate an Internet of Things Understanding of the World", *2013 International Workshop on Identification, Information and Knowledge in the Internet of Things (IIKI)*, Beijing, China, 2013, pp. 1-8.

[23] "reelyactive/advlib", GitHub. [Online]. Available: https://github.com/reelyactive/advlib

[24] M.I. Jameel and J. Dungen, "Low-power wireless advertising software library for distributed M2M and contextual IoT", *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, Italy, 2015, pp. 597-602.

[25] "reelyactive/beaver", GitHub. [Online]. Available: https://github.com/reelyactive/beaver

[26] "reelyactive/pareto-anywhere-apps", GitHub. [Online]. Available: https://github.com/reelyactive/pareto-anywhere-apps

[27] "reelyActive Kibana integration", reelyActive Developers. [Online]. Available: https://reelyactive.github.io/diy/kibana/

[28] "Accueil - GreenUXLab", UQAM. [Online]. Available: https://greenuxlab.uqam.ca/

[29] T. J. Foster, L. Justice, H. Gonzalez Villasanti, D. Irvin and D. Messinger, "Classroom Sensing Tools: Revolutionizing Classroom-Based Research in the 21st Century", *Topics in Early Childhood Special Education*, 2024, 0(0).

[30] T. J. Foster, et al. "Peer-Mediated Intervention for Socially Isolated Preschoolers: An Early-Stage Feasibility Study.", *Language, Speech, and Hearing Services in Schools 55.1*, 2024, pp. 166-180.