Towards a Semantic Architecture for the Internet of Musical Things

Conference Paper · November 2018

DOI: 10.23919/FRUCT.2018.8587917

4 authors:

Luca Turchet
Università degli Studi di Trento

Fabio Viola
INFN - Istituto Nazionale di Fisica Nucleare

György Fazekas
Queen Mary, University of London

Mathieu Barthet
Queen Mary, University of London

Towards a Semantic Architecture for the Internet of Musical Things

Luca Turchet* ✉, Fabio Viola†, György Fazekas* and Mathieu Barthet*
*Queen Mary University of London, London, United Kingdom
Email: {luca.turchet, m.barthet, g.fazekas}@qmul.ac.uk
†University of Bologna, Bologna, Italy
Email: fabio.viola@unibo.it

***Abstract*—The Internet of Musical Things is an emerging research area that relates to the network of Musical Things, which are computing devices embedded in physical objects dedicated to the production and/or reception of musical content. In this paper we propose a semantically-enriched Internet of Musical Things architecture which relies on a semantic audio server and edge computing techniques. Specifically, a SPARQL Event Processing Architecture is employed as an interoperability enabler allowing multiple heterogeneous Musical Things to cooperate, relying on a music-related ontology. We technically validate our architecture by implementing an ecosystem around it, where five Musical Thing prototypes communicate between each other.**

I. Introduction

The recent proliferation of Internet of Things (IoT) technologies [1] is impacting many areas of modern day living. One area that has received considerably little attention from the IoT research community, compared to other domains, is that of music. Lately, some authors have started to investigate how to extend the IoT paradigm to the musical domain, proposing visions for the so-called "Internet of Musical Things (IoMusT)" [2], [3].

Turchet et al. have proposed a definition for the IoMusT from a computer science perspective [4]. According to their vision, the Internet of Musical Things refers to *"the ensemble of interfaces, protocols and representations of music-related information that enable services and applications serving a musical purpose based on interactions between humans and Musical Things or between Musical Things themselves, in physical and/or digital realms. Music-related information refers to data sensed and processed by a Musical Thing, and/or exchanged with a human or with another Musical Thing"*. A Musical Thing was defined as *"a computing device capable of sensing, acquiring, processing, or actuating, and exchanging data serving a musical purpose"*.

Examples of Musical Things are represented by the so-called "smart musical instruments (SMIs)" and "musical haptic wearables (MHWs)". SMIs are a family of musical instruments proposed in [5]. Such musical instruments are characterized by embedded computational intelligence, wireless connectivity, an embedded sound delivery system, and an onboard system for feedback to the player. They were devised to offer direct point-to-point communication between each other and other portable sensor-enabled devices connected to local networks and to the Internet. MHWs are a class of wearable devices for performers and audience members, which encompass haptic stimulation, gesture tracking, and wireless connectivity features [6], [7]. MHWs were devised to enrich musical experiences by leveraging the sense of touch as well as providing new capabilities for creative participation. Their conception was grounded on the findings of research in the field of haptic technologies developed for musical applications and of participatory live music performances.

A key aspect of the IoMusT paradigm is the interoperability among Musical Things. Interoperability involves three levels: network, syntax, and semantics. The network interoperability regards protocols for exchanging information among heterogeneous devices, regardless of the content of the messages (an example of this category is the Wi-Fi protocol). The syntax interoperability level concerns the way messages are structured and encoded (an example is represented by the RDF protocol [8]). The third level conveys the meaning of the exchanged messages [9] (an example of this category is provided by the Web Ontology Language [10]). Interoperability in an IoT scenario can be achieved only if a set of standardized protocols is employed. Indeed in the IoT, differently from the Web, information must be machine readable rather than human readable [11]. Notably, semantic interoperability through standardized protocols is the key for multi-domain applications crossing IoT vertical silos [12]. In this way the context – *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"* [13] – which is the core of the whole IoT application, can be automatically and collaboratively processed and enriched (see e.g., [14]).

To date, interoperability across musical devices in co-located settings has mostly relied on existing communications standards such as Wi-Fi, as well as protocols for wirelessly exchanging musical messages, such as the Open Sound Control (OSC) protocol [15] over the User Datagram Protocol. Semantic technologies have been envisioned as an alternative and more general solution to enable interoperability across heterogeneous Musical Things [4]. Nevertheless, to the best of the authors' knowledge, no effort has been conducted yet to apply semantic technologies to IoMusT scenarios.

In this paper, we propose a semantically-enriched IoMusT architecture supporting interactions between different actors using Musical Things, such as performers and audience members. The architecture is based on a semantic audio server, to which various Musical Things are connected to form an ecosystem of interoperable devices around a wireless local area network. We validate the developed technological infrastructure by applying it to a proof-of-concept scenario where a Musical Thing exchanges information with other Musical Things, by relying on semantic audio [16], [17] and edge computing [18] techniques.

The unique and novel combination of technologies proposed in this paper provides the grounding for and have the capacity to transform live and recorded, local or remote music experiences. By delivering a smart environment in which musical capabilities are augmented, while audiences are brought in closer contact with performers and with each other through mobile devices, wearables, haptics, immersion, extended reality and related concepts and technologies, we may cater for emerging needs in society and the music industry alike.

The remainder of this paper is organized as follows. Section II presents an overview of the related works. Section III describes the developed hardware and software architecture, while Section IV discusses the components of an IoMusT ecosystem that forms around the proposed architecture. Section V presents a basic implementation of an IoMusT ecosystem employing the semantic architecture, which constitutes a technical validation. Section VI discusses the implications and challenges of the proposed architecture.

II. Related works

In this section we review key works on technologies related to the proposed architecture.

A. Interoperability

The Musical Instrument Digital Interface (MIDI) protocol is a well established framework enabling Digital Musical Instruments to exchange musical information but it relies on the serial transmission of data and is limited in resolution. Open Sound Control (OSC) is a more flexible standard to organize and transmit sound control information over networks. However, the syntax for musical control is not provided by default like for MIDI and needs to be designed in a way that is suitable for networked exchange. In terms of synchronisation, the Ableton Link protocol provides a way for DMIs to share tempo information. Further research is needed to extend the interoperable capabilities of SMIs in the context of the Internet of Musical Things [2]. To this end, semantic web technologies and knowledge representation are promising since they are agnostic to the communication protocols which are used and offer reasoning and inference capabilities (see e.g. [17]).

Malloch et al. proposed a 3-layer mapping framework and tools where a semantic layer links gesture to sound semantics (McGill Digital Orchestra mapper tools) [19], [20]. The McGill libmapper tool relies on Open Sound Control (OSC) and provides decentralized resource allocation, discovery, and flexible connectivity letting devices describe themselves and their capabilities. However, it targets the use of Local Area Network (LAN) subnet where support for multicast can be guaranteed [21].

B. Semantic Web

The Semantic Web was introduced by Tim Berners-Lee [22] as a new way of thinking the Web: a novel way to represent data in order to make them machine-understandable. The set of protocols introduced by the so-called "Semantic Web stack" allows one to identify resources in a univocal way (i.e., IRI), to represent data according to a simple formalism (i.e., RDF) and give data a meaning (through RDFS and OWL). This is fundamental for interoperability at syntactic and semantic levels. In this way, data can be automatically processed.

Applying Semantic Web technologies to the IoT is quite natural for three reasons: it allows one i) to face the heterogeneity among entities in the same domain, ii) to tear down the barriers across vertical silos and iii) to easily integrate devices into the Web. Furthermore this topic is timely as demonstrated, for example, by the work reported in [23] where authors propose a new way to make Semantic Web technologies suitable for embedded devices. One of the main challenges in this field consists of the adaptation of Semantic Web technologies in order to make them comply with the requirements of highly constrained IoT scenarios, such as those of the Internet of Musical Things. Among the examples of technologies exploiting semantics in the IoT, it is worth mentioning the works reported in [24] and [25], where authors rely on a semantic middleware providing publish-subscribe functionalities on top of an RDF knowledge base [26] in two different IoT scenarios: electro-mobility and home automation. In the music domain, several applications of the Semantic Web were reported in [17].

C. Semantic Audio and Ontologies

Semantic Audio [16] is an emerging field in the confluence of audio signal processing, machine learning and Semantic Web technologies. In Semantic Audio, these techniques are applied in tandem in order to enable interaction with sound and music in human terms [17]. Semantic Audio involves the application of signal analysis to extract descriptors from digital audio, ranging from signal level features, through perceptual characteristics, to high-level semantic descriptors. Examples include spectral centroid [27] and other low-level spectral and temporal characteristics of a signal that are calculated through closed form expressions (see e.g., [28], [29]). Mid-level representation often correspond to a perceptual quality such as loudness or timbre (see e.g., [30]). These require more complex computation involving for instance auditory models [31]. High-level representations are typically in the center of attention aiming to capture musical, cultural and even subjective aspects of audio and music including musical structure [32], genre [33] and mood [34]. Techniques to extract these features include machine learning as well as knowledge representation and processing techniques. Semantic Audio however does not stop at audio feature extraction, but it encompasses structured, machine processable representations of audio analyses that facilitate automated data processing and knowledge-based reasoning [17]. This makes it analogous to Semantic Web research aiming in part to extract machine processable information from textual web content written for humans. Semantic Audio also includes rich conceptualisation of audio metadata including the composition, production and consumption of audio and music [35]. The field has numerous applications, from navigation in large audio collections through automated tagging [36], through music recommendation [37] to intelligent music production [38] and participatory music performance [39], [40].

An emerging application is in the intersection of Semantic Audio and IoT. The analysis of sensor signals, e.g., transducers embedded in instruments, combined with structured representation of these data enables the combination of heterogeneous devices in smart environments. This is leading to IoMusT that is in the focus of this present work. A requisite for these systems to interoperate are music related ontologies that augment data with semantics. Several ontologies have been proposed for this purpose, including the Music Ontology [41] for describing rich editorial metadata, the Studio Ontology [42] for describing detailed nuances in music production and the application of signal processing devices, and the Audio Features Ontology [43] for describing computational features of audio.

D. Musical Things

1) Smart Instruments: According to the proposal for the family of smart musical instruments formulated by Turchet et al. [5], SMIs are the result of the integration of a variety of technologies that were developed for different purposes [44]. These include networked music performance systems [45], Internet of Things technologies [46], sensor- and actuator-based "augmented instruments" [47] (e.g., [48], [49]), embedded acoustic and electronic instruments [50], [51], as well as methods for sensor fusion [52], audio pattern recognition [53], semantic audio [16], and machine learning [54].

To date, only a handful of SMIs exist. An example of this family of instruments is the Sensus Smart Guitar developed by MIND Music Labs [5]. This is a hollow body guitar augmented with on-board processing, a system of multiple actuators attached to the soundboard, different sensors embedded in various parts of the instrument, and interoperable wireless communication. The sound engine is based on the ELK music operating system (https://www.mindmusiclabs.com/ELK) and affords a large variety of sound effects and sound generators, as well as it is programmable via dedicated apps on desktop PCs, smartphones, and tablets.

Another instance of SMIs is the smart cajón reported in [55]. This instrument consists of a conventional acoustic cajón smartified with sensors, Wi-Fi connectivity, motors for vibro-tactile feedback, the Bela board for low-latency audio and sensors processing [56], which runs a sound engine composed by a sampler and various audio effects. A peculiarity of the embedded intelligence is the use of sensor fusion and semantic audio techniques to estimate the location of the players' hits on the instrument's front and side panels, and to map this information to different sound samples simulating various percussive instruments [57].

2) Musical haptic wearables: To date, scarce research has been conducted on the development and perceptual evaluation of musical haptic wearables for audience members (MHWAs). The work reported in [58] describes an IoMusT architecture connecting a smart cajón and a smart mandolin with four instances of an armband-based MHWA. Such IoMusT architecture relied on a Wi-Fi network and the exchange of OSC messages between the involved Musical Things. Specifically, the onset of hits and strums above an amplitude threshold were extracted in real-time from acoustic signals captured by the microphones of the two instruments. Such information was sent to the connected MHWAs and mapped to a strong and short vibration so that audience members can experience a tactile stimulation mirroring strong hits and strums.

The examples mentioned above show the use of MHWs as receivers. However, these devices may encompass a sensor interface that allows one to deliver messages. For instance, in [6] and [7], the authors report musical haptic wearables for performers (MHWPs), which are equipped with buttons. In one of the experiments reported in [7], two buttons embedded in an armband-based MHWP were used by electronic music performers to send musical directions between them to organize improvisations.

III. IoMusT semantic architecture

From a general perspective, a semantic architecture for the IoMusT should grant a timely and loosely-coupled interaction among entities, sharing semantically-enriched information suitable for automatic computations. A semantic publish-subscribe message-oriented middleware [59], together with a set of agreed ontologies, allows one to realize this vision. A central role in the middleware is played by a message broker, an intermediary software that acts as an interoperability enabler by sharing the content of the knowledge base (represented as an RDF graph) among the entities that simultaneously shape it. The IoMusT semantic architecture relies then on a client/server architecture named SPARQL Event Processing Architecture (SEPA) born as a descendent of the Smart-M3 interoperability platform (described in [60] and [26]) and one or more domain-specific ontologies.

This Section provides details about both the server and client sides of SEPA. The ontology determining the way information is represented (being application-dependent) is introduced in Section V where a prototype of IoMusT ecosystem, based on the semantic architecture, is presented.

A. Server-side

Through Semantic Web data representation model, all the information is encoded as a set of triples (i.e., *subject*, *predicate* and *object*) that form an oriented and labelled graph. Graph stores are then used by Semantic Web applications to store and access information by means of the SPARQL Update and Query languages (graph stores with these capabilities are usually referred to as "SPARQL endpoints"). Semantic technologies are one of the key enabling technologies for the IoT, as mentioned by Goudos et al. in [61], but only if proper ways to meet IoT requirements are set up. To minimize the impact of semantic technologies in this IoT scenario, an enhanced semantic architecture was adopted. A SPARQL Event Processing Architecture [62] implements a content-based publish-subscribe broker [63] on top of a standard SPARQL endpoint. Through SEPA, clients avoid polling for data (i.e., data is dispatched as soon as available). Moreover, SEPA (after an initial message with the state of the subgraph of interest) sends notifications containing only the delta (i.e., added and removed triples) between the last update and the current one. In this way, it is possible to reduce the amount of data transferred over the network as well as the computational effort of every client.

Server-side, SEPA can be considered a SPARQL endpoint enriched with publish/subscribe capabilities. To the best of authors' knowledge, the publish-subscribe paradigm is not natively implemented in any of the existing SPARQL endpoints, therefore SEPA represents an additional layer built on top of them. SEPA exposes the same interface of a SPARQL endpoint (i.e., it is a transparent layer) plus an additional WebSocket interface to support the SPARQL Subscribe Language.

B. Client-side

The general architecture of a SEPA application (as depicted in Fig. 1) encompasses three kinds of client, known in the Smart-M3 terminology as *Knowledge Processors (KP)*. According to the work by Roffia et al. [64], KP can be classified among:

- **Producers.** Their role is to update the content of the knowledge base by adding, removing or modifying information to a given graph. Producers operate through SPARQL Update requests sent over HTTP(S) according to the SPARQL 1.1 protocol (continuous black lines in Fig. 1). In every SEPA application there is at least one producer (while the maximum number of producers is not limited a priori).
- **Consumers.** They perform read-only operation on the knowledge base. Consumer may operate through the request-response paradigm (by issuing a query) or through the publish/subscribe (by issuing a subscription, also known as "persistent query"). Subscriptions allows one to be timely notified about changes in the subgraph of interest. While query requests are issued via HTTP(S) according to the SPARQL 1.1 protocol (dashed lines in Fig. 1), subscriptions exploit the WebSocket protocol (dashed and dotted lines). In both cases, the subgraph of interest is specified by means of the SPARQL 1.1 Query Language.
- **Aggregators.** They are at the same time consumers and producers, since they react to changes in the subgraph of interest by updating it with new knowledge. It is worth noticing that in SEPA applications the presence of aggregators is not mandatory if at least one consumer is present.

Notably, this high-level abstraction of a KP is not limiting, since a complex application may implement and run more than one KP at a time, also playing different roles.

The SEPA platform also provides a way for the fast prototyping of KPs, through the definition of a JSON Semantic Application Profile (JSAP), a JSON file including the whole set of templates for SPARQL Updates and Queries/Subscriptions shaping the information flow of the application as well as the configuration parameters for the reference broker. Then, producers, consumers and aggregators can be quickly set up by loading SPARQL code from the JSAP and implementing the business logic around them. The ensemble of SEPA server and clients can be classified as a publish-subscribe message-oriented middleware [59].

IV. IoMusT ecosystem based on the architecture

An IoMusT ecosystem based on the semantic architecture described in Section III may encompass several different Musical Things playing the roles of producers, consumers and aggregators as shown by Fig. 2. In the following we detail examples of Musical Things covering each of these roles.

Producers. Producers represent the source of information in a SEPA application. In this ecosystem at least one producer must be present and this role can be played by devices belonging to different categories. As an example, producers may be Musical Things such as SMIs, MHWPAs, MHWPs, or smartphones with musical apps, which publish audio features calculated on board. Notably, these calculations are particularly relevant to the edge computing paradigm as instead to leave the centralized server compute features from the signals generated by the devices, these are computed by the Musical Things themselves.

Consumers. Multiple and heterogeneous consumers can co-exist in the ecosystem. Such Musical Things may be for instance SMIs, which may modify some of the parameters of their sound engine according to the information read from SEPA. Other examples are stage equipment (e.g., lighting systems, screens displaying visuals, smoke machines, etc.), wearables such as smart glasses, virtual reality headsets, and MHWPAs, or smartphones and tablets. All these Musical Things change their behavior in response to the information to which they have subscribed from SEPA.

Aggregators. As we said earlier, aggregators are not mandatory in this ecosystem. When present, aggregators could be for instance SMIs, MHWPAs, MHWPs, smartphones, or laptops.

V. Technical validation: implementation of an IoMusT ecosystem

To validate our architecture we implemented an ecosystem around it, where five prototypes of Musical Things communicated between each other relying on semantic audio [16] and edge computing [18] techniques. This section reports a description of the hardware and software components of the developed system.

A. Ecosystem components

The ecosystem (see Fig. 3) comprised the following components:

i) Musical Thing prototypes. We developed five prototypes that encompass key features of Musical Things, namely an embedded platform and wireless connectivity. Specifically, as a platform we used the Bela board for low-latency audio and sensors processing [56] (either in its normal version and in the pocket version called "Bela-mini"). Wireless connectivity was enabled via the NETGEAR A6100-100PES Wi-Fi USB dongle attached to the Bela board (which supports the IEEE 802.11ac Wi-Fi standard). Power supply was provided by a powerbank.

One prototype was given the role of producer, three that of consumer, and one that of aggregator. Differently from the aggregator, the producer and the three consumers were configured to generate sounds. For both producer and consumers,

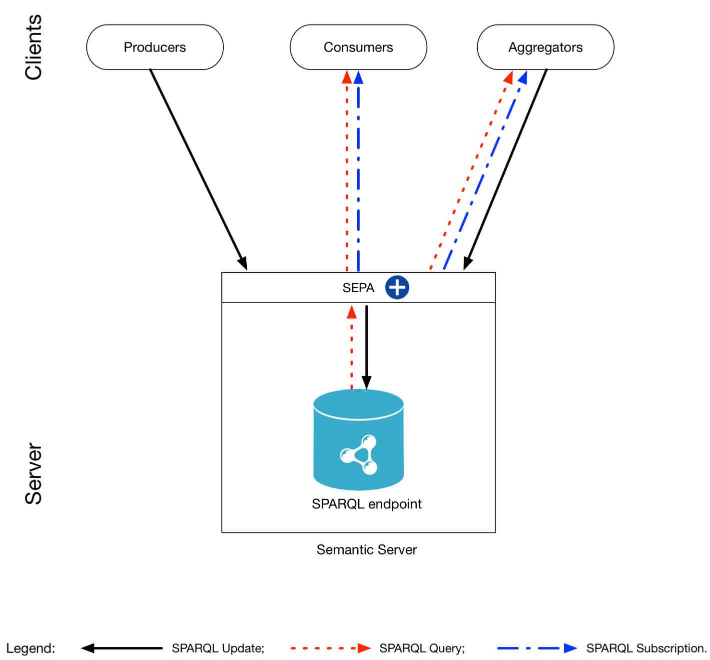

Fig. 1. The semantic IoMusT architecture, based on a client/server architecture of a SEPA application.

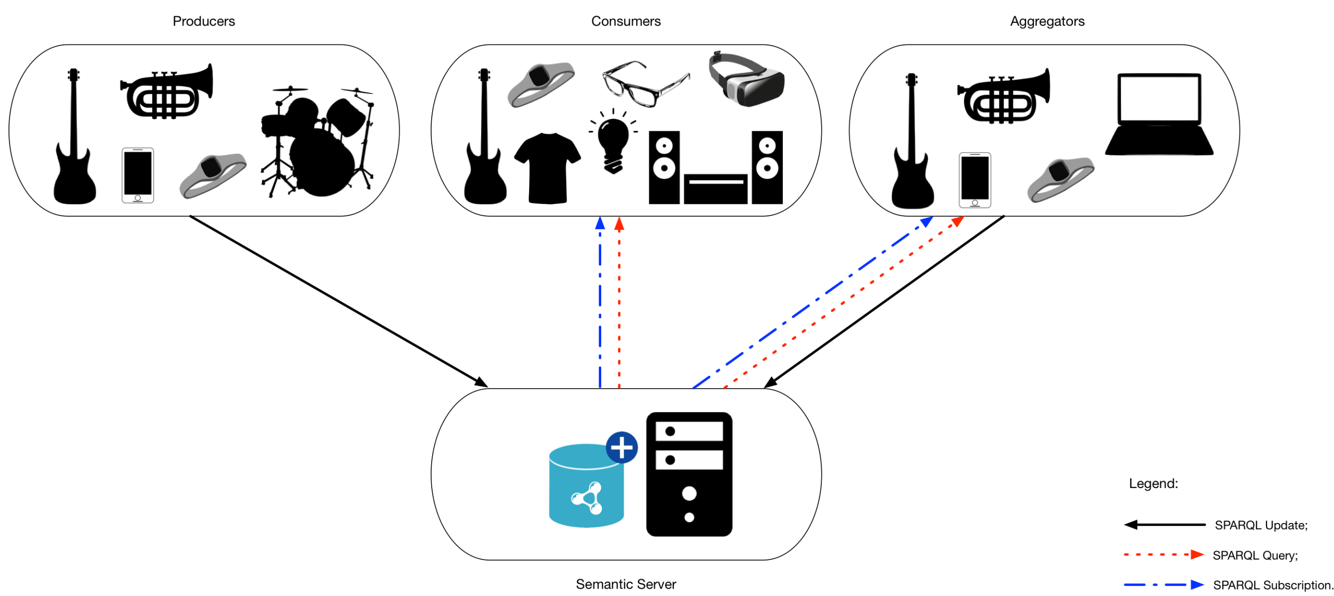

Fig. 2. Schematic representation of an example of ecosystem based on the proposed semantic IoMusT architecture. Different types of Musical Things play different roles of producer, consumer, or aggregator.

a small loudspeaker was used for sound delivery. The sound engine was coded in libpd, a porting of the Pure Data computer music environment into a library for embedded systems [65].

Producer. The producer's sound engine consisted of a generator of synthesized notes (by means of a basic sinusoidal oscillator), whose density was randomized in the range of [1, 200] notes per second. The parameters of each generated note were randomized as follows: the frequency ranged among the frequencies of the A major scale across three octaves; the duration ranged between 10 and 150 ms; the amplitude ranged between 0.01 and 1. Following the tenets of edge computing, we did not stream to the semantic server the produced audio signal or the flow of numbers characterizing the random behavior of the notes generator. Instead, we streamed the average of the four parameters (density, frequency, duration, and amplitude) computed every 5 seconds. Such computations were performed by a libpd patch. The data were streamed through SPARQL Update requests by a python script, which received data from the libpd sound engine via OSC messages and mapped the requests according to the Audio Features Ontology [43].

Aggregator. The aggregator did not produce any sound but served two other purposes: the first was to analyze the information related to the producer, sent by the semantic server; the second was to deliver to the server the results of the performed analysis. Specifically, we adopted an analysis based on fuzzy logic [66]: the 81 possible combinations (resulting from dividing into 3 parts the range of each of the 4 parameters), were randomly grouped into 4 subsets of 20, 20, 20, and 21 quadruplets. The quadruplets belonging to each subset were then associated to one of the 4 possible statuses: "A major", "E major", "F# minor", "silence". Such statuses were then sent back to the semantic server that dispatched it to the three consumers.

Consumers. Consumers were given the role of accompaniment of the melody played by the producer. Their sound engine was configured to produce one of the following chords: A major, E major, F# minor. These chords were selected to achieve a sense of consonance with the played melody (according to the tenets of the classic harmony theory [67]). These chords were rendered by a bank of sinusoidal oscillators. Thanks to a python script communicating via OSC messages to the libpd sound engine, each chord was played and stopped according to the notifications issued by the SEPA server. Each of the three consumers was assigned to one of the statuses "A major", "E major", "F# minor". When the status "silence" was issued by the SEPA server then no chord was played and only the melody of the producer was played.

ii) Audio Features Ontology. To identify the proper ontology for this application scenario, we focused on the context of our application. The main entities in the context were four high-level audio features, i.e., average amplitude, average frequency, average duration and average density. Therefore, the ability to map this context to a set of RDF triples led to the selection of the Audio Features Ontology reported in [43]. The ontology was then exploited by the client to represent or interpret information. The ontology was extended to define a new class for the inferred status and a new object property linking instances of the current performance with the instance of the status.

iii) Semantic Server. SEPA ran on a Dell Alienware 17 R2 laptop supporting the IEEE 802.11ac Wi-Fi standard and running Ubuntu Linux 17.10. The version 0.8.4 of the Java implementation of SEPA was used. To enhance the performance of the application and meet the requirements of the IoMusT domain, the semantic server only hosted the current state of the context. Then, from a semantic point of view, the context of this use case encompassed the following entities: 1) the current performance; 2) the last high-level audio features extracted by the producer; 3) The most recent state inferred by our aggregator KP.

iv) Network. All devices were connected using the Wi-Fi router TP-Link TL-WR902AC, which features the IEEE 802.11ac standard over the 5GHz band. Following the recommendations reported in [68] to optimize the components of a Wi-Fi system for live performance scenarios to reduce latency and increase throughput, the router was configured in access point mode, security was disabled, and only the IEEE 802.11ac standard was supported.

VI. Discussion and Conclusion

The Semantic server proposed in this paper hosts and shares the current context of an Internet of Musical Things application. The whole architecture leverages web standards, as the knowledge base is a set of RDF triples expressed according to an OWL ontology (i.e., the Audio Features Ontology [43]). Moreover, the server provides a standard interface compliant with the SPARQL 1.1 protocol and enhanced with a WebSocket interface. This opens the way for a fully interoperable scenario where 1) multiple agents can seamlessly join/leave the ecosystem, 2) multiple applications exploiting the same context can co-exist, possibly enriching the context with new information. The set of KPs realizing the business logic of the application was developed using the Python programming language ensuring easy portability across different platforms.

The system presented in Section V also provides an example of how artificial intelligence combined to semantic web can support music making. This is achieved here by conferring Musical Things the ability to produce a simple musical accompaniment based on audio-related attributes. This could be of interest for new interfaces for musical expression and computationally creative music systems (see e.g., [69]).

Important advantages of our system and this particular realization using seamntic web standards include *i)* the openness of the proposed architecture, *ii)* the separation of the logical data model from implementation details. This makes the data models, representations and their relation to musical concepts, events or acctions reusable and improves interoperability overall. The graph-based conceptualization of RDF data representation also lends itself to representing complex musical metadata more easily compared to tree-based structures such as XML or pure JSON (Note that this problem can be addressed using JSON-LD: http://json-ld.org/) or protocols that only support ad-hoc semantics such as OSC. The benefits of this is apparent in how SEPA, originally conceived for very different IoT applications, was easily able to serve as an arbitrator in a musical performance environment.

On the downside, Semantic Web technologies are often criticized for being too verbose [70], [71] (i.e., messages are often too long, resulting in higher requirements in terms of bandwidth and computational power). Potentially, this is conflicting with the typical requirements of IoT applications (such as timeliness, scalability, and ability to run on constrained devices to name a few) [72], and especially IoMusT applications. Therefore, further investigation is needed to assess the scalability of the scenario to better define future directions. Furthermore, Semantic Web technologies have a steep learning curve [73], that may represent an obstacle to the newcomers willing to write an IoMusT application.

In this paper we validated our architecture in use, by implementing an ecosystem around it, where basic prototypes of Musical Things were involved without involving any form of interactive control by human actors. However, the true power of our semantic IoMusT architecture is the interconnection of a diverse network of real Musical Things (such as SMIs or MHWs), during real-world applications such as live concerts. For instance, one can envision a multisensory concert where one or more SMIs control MHWAs, smoke machines, stage lights, or even parameters of other SMIs' sound engine.

The use case implemented in this study focused on a co-located settings where all devices were connected across a Wi-Fi-based wireless local area network. Nevertheless, the

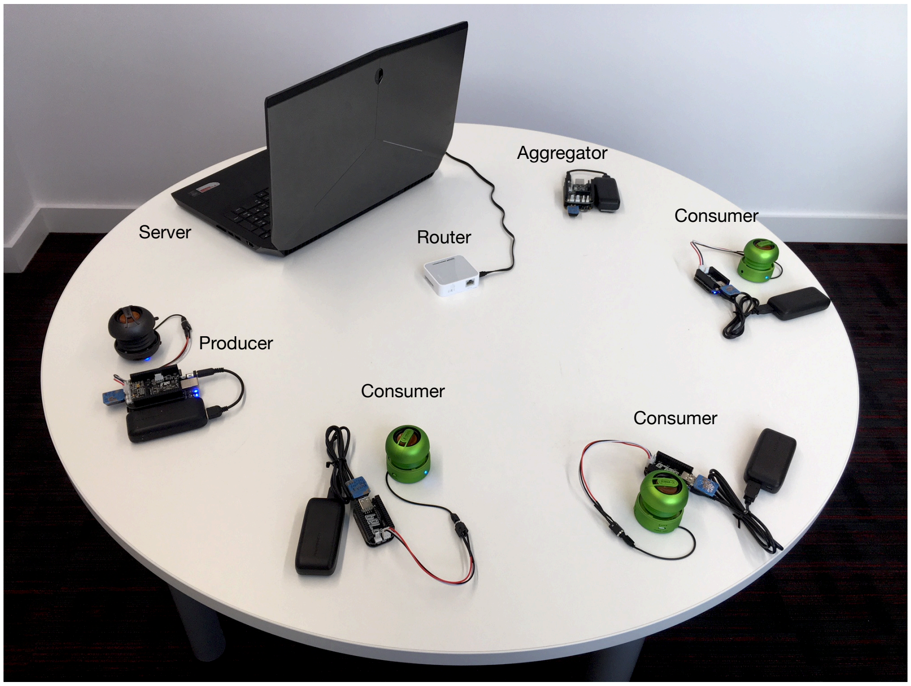

Fig. 3. The developed IoMusT ecosystem.

architecture can be extended to support remote interactions between Musical Things across a wide area network.

Acknowledgment

Luca Turchet acknowledges support from the EU H2020 Marie Curie Individual fellowship (No. 749561), and Mathieu Barthet and George Fazekas also acknowledge support from the EU H2020 Audio Commons Initiative grant (No. 688382).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010. [Online]. Available: https://doi.org/10.1016/j.comnet.2010.05.010

[2] L. Turchet, C. Fischione, and M. Barthet, "Towards the Internet of Musical Things," in *Proceedings of the Sound and Music Computing Conference*, 2017, pp. 13–20.

[3] D. Keller and V. Lazzarini, "Ecologically grounded creative practices in ubiquitous music," *Organised Sound*, vol. 22, no. 1, pp. 61–72, 2017.

[4] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet, "Internet of Musical Things: Vision and Challenges," *IEEE Access*, vol. 6, pp. 61 994–62 017, 2018.

[5] L. Turchet, "Smart Musical Instruments: vision, design principles, and future directions," *IEEE Access*, 2018.

[6] L. Turchet and M. Barthet, "Envisioning Smart Musical Haptic Wearables to Enhance Performers' Creative Communication," in *Proceedings of International Symposium on Computer Music Multidisciplinary Research*, 2017, pp. 538–549.

[7] ——, "Co-design of Musical Haptic Wearables for electronic music performer's communication," *IEEE Transactions on Human-Machine Systems*, 2018 (submitted).

[8] O. Lassila and R. Swick, "Resource description framework (rdf) model and syntax specification," 1998. [Online]. Available: http://www.w3.org/TR/REC-rdf-syntax/

[9] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "Role of middleware for internet of things: A study," *International Journal of Computer Science and Engineering Survey*, vol. 2, no. 3, pp. 94–105, 2011.

[10] D. McGuinness and F. Van Harmelen, "Owl web ontology language overview," 2004. [Online]. Available: https://www.w3.org/TR/owl-features/

[11] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," in *Architecting the internet of things*. Springer, 2011, pp. 1–24.

[12] P. Desai, A. Sheth, and P. Anantharam, "Semantic gateway as a service architecture for iot interoperability," in *IEEE International Conference on Mobile Services*. IEEE, 2015, pp. 313–319.

[13] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *International symposium on handheld and ubiquitous computing*. Springer, 1999, pp. 304–307.

[14] F. Viola, A. Stolfi, A. Milo, M. Ceriani, M. Barthet, and G. Fazekas, "Playsound.space: enhancing a live performance tool with semantic recommendations," in *Proc. 1st SAAM Workshop (in press)*. ACM, 2018.

[15] M. Wright, A. Freed, and A. Momeni, "Opensound control: State of the art 2003," in *Proceedings of the Conference on New Interfaces for Musical Expression*, 2003, pp. 153–160.

[16] G. Fazekas and T. Wilmering, "Semantic Web and Semantic Audio Technologies," *Tutorial presented at the 132nd Convention of the Audio Engineering Society, Budapest, Hungary*, 2012.

[17] G. Fazekas, Y. Raimond, K. Jakobson, and M. Sandler, "An overview of Semantic Web activities in the OMRAS2 Project," *Journal of New Music Research special issue on Music Informatics and the OMRAS2 Project*, vol. 39, no. 4, pp. 295–311, 2011.

[18] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[19] J. Malloch, S. Sinclair, and M. M. Wanderley, "A network-based framework for collaborative development and performance of digital musical instruments," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2007, pp. 401–425.

[20] J. Malloch, S. Sinclair, and M. Wanderley, "Libmapper: (a library for connecting things)," in *Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 3087–3090.

[21] ——, "Distributed tools for interactive design of heterogeneous signal

networks," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5683–5707, 2015.

[22] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.

[23] V. Charpenay, S. Käbisch, and H. Kosch, "μrdf store: Towards extending the semantic web to embedded devices," in *European Semantic Web Conference*. Springer, 2017, pp. 76–80.

[24] A. D'elia, F. Viola, F. Montori, M. Di Felice, L. Bedogni, L. Bononi, A. Borghetti, P. Azzoni, P. Bellavista, D. Tarchi *et al.*, "Impact of interdisciplinary research on planning, running, and managing electromobility as a smart grid extension," *Access, IEEE*, vol. 3, pp. 2281–2305, 2015.

[25] A. D'Elia, L. Perilli, F. Viola, L. Roffia, F. Antoniazzi, R. Canegallo, and T. S. Cinotti, "A self-powered wsan for energy efficient heat distribution," in *Proceedings of the IEEE Sensors Applications Symposium*, 2016, pp. 1–6.

[26] A. D'Elia, F. Viola, L. Roffia, P. Azzoni, and T. S. Cinotti, "Enabling interoperability in the internet of things: A OSGi semantic information broker implementation," *International Journal on Semantic Web and Information Systems*, vol. 13, no. 1, pp. 147–167, 2017.

[27] E. Schubert, J. Wolfe, and A. Tarnopolsky, "Spectral centroid and timbre in complex, multiple instrumental textures," in *Proc. of the 8th International Conference on Music Perception and Cognition*, 2004.

[28] M. Barthet, P. Depalle, R. Kronland-Martinet, and S. Ystad, "Acoustical correlates of timbre and expressiveness in clarinet performance," *Music Perception: An Interdisciplinary Journal*, vol. 28, no. 2, pp. 135–154, 2010.

[29] ——, "Analysis-by-synthesis of timbre, timing, and dynamics in expressive clarinet performance," *Music Perception: An Interdisciplinary Journal*, vol. 28, no. 3, pp. 265–278, 2011.

[30] M. Barthet, R. Kronland-Martinet, and S. Ystad, "Consistency of timbre patterns in expressive music performance," in *9th International Conference on Digital Audio Effects*, 2006, pp. 19–25.

[31] A. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *Trans. Audio, Speech and Lang. Proc.*, vol. 16, no. 2, pp. 255–266, 2008.

[32] M. Tian, G. Fazekas, D. A. A. Black, and M. Sandler, "On the use of the tempogram to describe audio content and its application to music structural segmentation," in *Proc. of the 40th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[33] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proc. of the 17th International Society for Music Information Retrieval (ISMIR-16) conference*, 2016, pp. 805–811.

[34] M. Barthet, G. Fazekas, and M. Sandler, *Music Emotion Recognition: From Content- to Context-Based Models*, From Sounds to Music and Emotions ed., ser. Lecture Notes in Computer Science, M. Aramaki, M. Barthet, R. Kronland-Martinet, and S. Ystad, Eds. Springer-Verlag, Heidelberg, Germany., 2013, vol. 7900.

[35] G. Fazekas and M. Sandler, "Knowledge representation issues in audio-related metadata model design," in *Proc. of the 133rd Convention of the Audio Engineering Society*, 2012.

[36] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[37] A. Allik, M. Mora-Mcginity, G. Fazekas, and M. Sandler, "Musicweb: Music discovery with open linked semantic metadata," in *Proc. 15th International Semantic Web Conference (ISWC), Posters and Demonstrations Track*, 2016.

[38] R. Stables, S. Enderby, B. De Man, G. Fazekas, and J. D. Reiss, "Safe: A System for Extraction and Retrieval of Semantic Audio Descriptors," in *15th International Society of Music Information Retrieval (ISMIR) Conference*, 2014.

[39] G. Fazekas, M. Barthet, and M. B. Sandler, *Novel Methods in Facilitating Audience and Performer Interaction Using the Mood Conductor Framework*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2013, vol. 8905, pp. 122–147.

[40] Y. Wu, L. Zhang, N. Bryan-Kinns, and M. Barthet, "Open symphony: Creative participation for audiences of live music performances," *IEEE MultiMedia*, vol. 24, no. 1, pp. 48–62, 2017.

[41] Y. Raimond, S. Abdallah, M. Sandler, and F. Giasson, "The music ontology," in *Proceedings of International Society for Music Information Retrieval Conference*, 2007.

[42] G. Fazekas and M. Sandler, "The studio ontology framework," in *Proc. of the 12th International Society for Music Information Retrieval (ISMIR'11) conference*, 2011, pp. 24–28.

[43] A. Allik, G. Fazekas, and M. Sandler, "An ontology for audio features," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2016, pp. 73–79.

[44] L. Turchet, "Some reflections on the relation between augmented and smart musical instruments," in *Proceedings of Audio Mostly Conference*, 2018.

[45] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.

[46] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014.

[47] E. Miranda and M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc, 2006, vol. 21.

[48] A. McPherson, "Buttons, handles, and keys: Advances in continuous-control keyboard instruments," *Computer Music Journal*, vol. 39, no. 2, pp. 28–46, 2015. [Online]. Available: https://doi.org/10.1162/COMJ_a_00297

[49] D. Overholt, E. Berdahl, and R. Hamilton, "Advancements in actuated musical instruments," *Organised Sound*, vol. 16, no. 02, pp. 154–165, 2011.

[50] D. MacConnell, S. Trail, G. Tzanetakis, P. Driessen, W. Page, and N. Wellington, "Reconfigurable autonomous novel guitar effects (RANGE)," in *Proceedings of the International Conference on Sound and Music Computing*, 2013.

[51] E. Berdahl, "How to make embedded acoustic instruments," in *Proceedings of the Conference on New Interfaces for Musical Expression*, 2014, pp. 140–143.

[52] L. Pardue, C. Harte, and A. McPherson, "A low-cost real-time tracking system for violin," *Journal of New Music Research*, vol. 44, no. 4, pp. 305–323, 2015.

[53] R. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," *Journal of New Music Research*, vol. 32, no. 2, pp. 153–163, 2003.

[54] R. Fiebrink and B. Caramiaux, "The machine learning algorithm as creative musical tool," in *Oxford Handbook of Algorithmic Music*, R. Dean and A. McLean, Eds. Oxford University Press, 2016.

[55] L. Turchet, A. McPherson, and M. Barthet, "Co-design of a Smart Cajón," *Journal of the Audio Engineering Society*, vol. 66, no. 4, pp. 220–230, 2018. [Online]. Available: https://doi.org/10.17743/jaes.2018.0007

[56] A. P. McPherson, R. H. Jack, and G. Moro, "Action-sound latency: Are our tools fast enough?" in *Proceedings of the Conference on New Interfaces for Musical Expression*, 2016.

[57] L. Turchet, A. McPherson, and M. Barthet, "Real-time hit classification in a Smart Cajón," *Frontiers in ICT*, vol. 5, no. 16, 2018.

[58] L. Turchet and M. Barthet, "An Internet of Musical Things architecture for performers-audience tactile interactions," in *Proceedings of the Digital Music Research Network Workshop*, 2017.

[59] M. Albano, L. Ferreira, L. Pinho, and A. Alkhawaja, "Message-oriented middleware for smart grids," *Computer Standards & Interfaces*, vol. 38, pp. 133–143, 2015.

[60] F. Viola, A. D'Elia, D. Korzun, I. Galov, A. Kashevnik, and S. Balandin, "The M3 architecture for smart spaces: Overview of semantic information broker implementations," in *Proceedings of the Conference of Open Innovations Association (FRUCT)*. IEEE, 2016, pp. 264–272.

[61] S. K. Goudos, P. I. Dallas, S. Chatziefthymiou, and S. Kyriazakos, "A survey of iot key enabling and future technologies: 5g, mobile iot, semantic web and applications," *Wireless Personal Communications*, vol. 97, no. 2, pp. 1645–1675, 2017.

[62] L. Roffia, P. Azzoni, C. Aguzzi, F. Viola, F. Antoniazzi, and T. Salmon Cinotti, "Dynamic Linked Data: A SPARQL Event Processing Architecture," *Future Internet*, vol. 10, no. 4, p. 36, 2018.

[63] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM computing surveys*, vol. 35, no. 2, pp. 114–131, 2003.

[64] L. Roffia, F. Morandi, J. Kiljander, A. DElia, F. Vergari, F. Viola, L. Bononi, and T. S. Cinotti, "A semantic publish-subscribe architecture for the internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1274–1296, 2016.

[65] P. Brinkmann, P. Kirn, R. Lawler, C. McCormick, M. Roth, and H.-C. Steiner, "Embedding pure data with libpd," in *Proceedings of the Pure Data Convention*, vol. 291, 2011.

[66] P. Hájek, *Metamathematics of fuzzy logic*. Springer Science & Business Media, 2013, vol. 4.

[67] W. Piston, *Harmony*. WW Norton, 1948.

[68] T. Mitchell, S. Madgwick, S. Rankine, G. Hilton, A. Freed, and A. Nix, "Making the most of wi-fi: Optimisations for robust wireless live music performance," in *Proceedings of the Conference on New Interfaces for Musical Expression*, 2014, pp. 251–256.

[69] C. Goddard, M. Barthet, and G. Wiggins, "Assessing musical similarity for computational music creativity," *Journal of the Audio Engineering Society*, 2018.

[70] S. Calbimonte, J.P.and Sarni, J. Eberle, and K. Aberer, "XGSN: An Open-source Semantic Sensing Middleware for the Web of Things," in *Proceedings of the International Workshop on the Foundations, Technologies and Applications of the Geospatial Web*, 2014, pp. 51–66.

[71] F. Scioscia and M. Ruta, "Building a semantic web of things: issues and perspectives in information compression," in *IEEE International Conference on Semantic Computing*. IEEE, 2009, pp. 589–594.

[72] M. Lytras, *Progressive Concepts for Semantic Web Evolution: Applications and Developments: Applications and Developments*. IGI Global, 2010.

[73] L. Yu, *Introduction to the semantic web and semantic web services*. Chapman and Hall/CRC, 2007.