Received 31 July 2024; revised 27 August 2024; accepted 12 September 2024. Date of publication 20 September 2024; date of current version 4 October 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3464696

DSP as a Service: Foundations and Directions

LUCA TURCHET [1] (Senior Member, IEEE), AND SACHA KRSTULOVIĆ [2,3] (Senior Member, IEEE)

[1]Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy
[2]Music Tribe, Dubai, UAE
[3]Understand-AI.today, London Cambridge CB4 2SY, U.K.
CORRESPONDING AUTHOR: L. TURCHET (e-mail: luca.turchet@unitn.it)

This work was supported by in part by the European Union under the Italian National Recovery and Resilience Plan of NextGenerationEU, with the MUR PNRR PRIN 2022 under Grant 2022CZWWKP, and in part by the Music Tribe.

ABSTRACT This article proposes a vision for digital signal processing as a service (DSPaaS), which exploits the Internet of Musical Things' capabilities to dematerialise and enhance music production tasks beyond networked music performance. If all musical devices were connected to the Internet, which new services and value could be created for musicians, and which technical challenges and trade-offs would arise? First, we identify the main components of a DSPaaS system and introduce its building blocks, technological enablers, design trade-offs and network configurations. Then, we segment DSPaaS applications into three categories based on different latency constraints. Subsequently, we describe three illustrative case studies and analyse them under the key performance indicators of latency and reliability. Finally, we discuss the current research challenges, aiming to inform developers' choices during the processes of creating new digital services for musicians, and to facilitate researchers' understanding of the key research directions stemming from the DSPaaS vision.

INDEX TERMS Internet of Musical Things, digital audio processing, embedded audio, cloud, edge computing.

I. INTRODUCTION

NEW TECHNOLOGY capability brings new economic value, in terms of enhanced user experiences and business models. Looking at the software sector, the commoditisation of Internet, cloud and Web technologies has transformed a software-as-a-product industry, where users would purchase a license for a specific software version and manually install it on their personal computers via cumbersome procedures, into a software-as-a-service (SaaS) industry [1], [2], where users purchase direct access to up-to-date software functionality against a subscription fee. For the users, value resides in constant access to the latest version without having to worry about manual upgrades, ubiquity of availability across several computers and new collaborative use cases. For businesses, a constant and more easily forecastable income stream, perceived by consumers as a lower point-in-time payment, results in higher profits. Looking at the media entertainment sector, improvements in Internet bandwidth and media encoding standards have supported the raise of platforms such as Netflix or Spotify, which offer an endless catalog of media contents to their users, viewable across a variety of devices, against a subscription fee. Following up with the Internet of Things [3], the miniaturisation of embedded systems, maturity in wireless connection standards and maturity in human-machine dialogue systems have opened up new convenience in home management. This is monetised through selling a new class of devices, as well as through offering companion subscriptions to an associated range of cloud services. Furthermore, cloud gaming [4] exploits commoditised high Internet bandwidth, bespoke video transmission standards [5] and evolution in cloud computing [6] to provide an endless catalogue of games, direct access to the gaming functionality, ubiquity of access across a variety of devices and collaborative gaming.

Conversely, the audio contents production industry seems late to the table as it still relies to a large extent on manufacturing and selling isolated un-networked hardware devices, as well as on one-off sales of a multiplicity of isolated audio processing software plugins, large audio sample collections and digital audio workstation software

packages, without a clear connection between the hardware side and the software side of this industry. Thus, the present article makes a case for a technological evolution towards *Digital Signal Processing as a Service* (DSPaaS), presented as a key enabler for new usage and new business value in the audio contents production sector.

DSPaaS has roots in the Internet of Musical Things paradigm [7], and can be seen as a subfield of such a paradigm. DSPaaS is defined as the interconnection of musical devices and remote servers to provide new networked services which dematerialise and enhance music production, beyond the most widely studied case of networked music performance. First we recap the background and drivers which justify this framework. Subsequently, we identify its main technical components and their related design trade-offs, e.g., network configurations. Then, we formalise and segment DSPaaS applications into three categories based on different latency constraints. Subsequently, we describe three case studies and analyse them under the key performance indicators of latency and reliability. Finally, we discuss the technical challenges that the field currently faces, with a particular focus on analysing the technical trade-offs at play in this technology. This discussion aims to inform designers' and developers' choices during the process of creating new digital services for musicians and audio content producers, as well as researchers' understanding of the key research directions stemming from the DSPaaS vision.

II. BACKGROUND AND DRIVERS

A. THE INTERNET OF MUSICAL THINGS

The paradigm of the Internet of Musical Things (IoMusT) [7] extends the Internet of Things to the musical domain, and draws upon several lines of music technology research including ubiquitous music [8], semantic audio [9], embedded audio [10], Web audio [11], and networked music performance [12], [13]. The IoMusT refers to the network of musical things, which are musical devices equipped with embedded intelligence and networking capabilities. Trends are clearly emerging in both academia and the industry to propose novel kinds of digital musical instruments or other devices serving a musical purpose [7], which are capable of exchanging information with external systems through Internet protocols via wired or wireless channels. A prominent example is the family of smart musical instruments proposed in [14].

The IoMusT infrastructure enables multidirectional communication, locally and remotely, between musical stakeholders, as well as between musical stakeholders and their devices. This supports the design of a new range of services in a variety of musical practices such as performance, composition, pedagogy or recreational music making, for playing in groups or individually. However, scarce research has been conducted thus far on which services can be built over IoMusT architectures, how to design such services and what the supporting networking architectures should consist of. In particular, the case of networked musical performance is very challenging because it requires very low communication latency, low and constant jitter and low packet losses or signal dropouts [15], [16]. Furthermore, the transmission of audio over Internet protocols is divided between two poles. On the one hand, fully managed networks, built either within the walls of a production studio or concert venue, or by renting private "dark fibers" from Internet service providers to link remote processing sites, combined with standards such as AES67 [17], RAVENNA [18] or Dante [19], allow to build low latency, high fidelity audio services. On the other hand, commodity Internet combined with digital encoding standards such as Opus, MP3 or AAC [20], [21], with audio streams transmitted over real-time protocol (RTP), only allow to build high latency services which suffer from lossy audio quality. Thus, achieving the combination of low latency and high fidelity outside of fully contained hardware and across commodity Internet remains a technological frontier [22]. Although the fifth-generation (5G) of cellular networks aims to reconcile these specifications over a commoditised form of network infrastructure, and advertises itself as a fundamental enabler of the IoMusT paradigm [23], its latency remains significantly higher than that of a fully managed wired infrastructure, and incompatible with low-latency lossless services.

IoMusT research has focused mostly on the case of human-human interactions, e.g., via networked music performance systems [12], [13] which allow geographically displaced musicians to play together [24], [25], [26], [27]. Less attention has been dedicated to the case of human-machine interactions mediated by the network, such as server-based music production services associated with Musical Things over commodity Internet, where the networking choices complement the digital signal processing choices to achieve a viable service.

B. WHY DSP AS A SERVICE?

This subsection lists categories of user and market values which justify the consideration of DSPaaS as a technical field and frame the analysis of related technical networking requirements:

1) LOWERING THE BARRIERS TO MAKING MUSIC

Remote computing coupled with artificial intelligence (AI) [28] is expected to empower users by lowering the barriers to playing and producing music: acquisition of skills such as dexterity, music notation and theory; equipment selection; sound design; mixing and mastering - all quantifiable in terms of effort, time and money costs. E.g., AI-powered mixing and mastering services such as RoEx[1] or Landr[2] save time and effort in the music production process, but as importantly make good mixing and mastering results accessible to musicians who could not otherwise afford the services of a human mixing/mastering professional.

[1]https://www.roexaudio.com/
[2]https://www.landr.com/

Similarly, conditional generation [29] allows, e.g., a musician able to sing and play guitar to obtain an AI-generated drum track to accompany their top line, instead of having to find a human drummer. Such AI-based services do not mean to compete against human professionals, insofar as their price points and quality levels aren't directly comparable. Rather, they act as entry points in the music making process, in terms of sketching something that a musician can select and refine with manual adjustments, and in terms of raising aspiring musicians' awareness of essential music production practices. Such services can run on device, but in most cases the size of the required AI models and involved machine learning processes, or the business models needed to make the service economically viable, require to run these in the cloud.

2) UBIQUITOUS CREATIVITY AND COLLABORATIVE AUDIO PRODUCTION

Ubiquitous and collaborative document editing has been possible since about 2010 in the space of shared office documents, calendars, lists, etc. Such applications are fundamentally supported by ubiquitous Internet access, cloud/Web technologies and powerful portable devices. When it comes to music, networked interaction with audio resources, material and tools is only starting to emerge as support to the creativity, expression and musical production of contemporary musicians [30], [31]. This phenomenon was particularly amplified by the COVID-19 pandemic [32], [33], [34]. Making music editable anywhere and anytime requires to blur the boundary between hardware and software. Ideally, a musician could start a music project using a hardware console, continue it using a digital audio workstation running on their laptop, then edit it on their smartphone when walking in a park because a creative idea emerged there and then. While ubiquity and interoperability still need progress, collaborative production has received more attention, e.g., with cloud-based applications such as Audiotool.[3]

3) NAVIGATING AN ENDLESS VARIETY OF AUDIO SAMPLES, EFFECTS AND SYNTHESISER SOUNDS

Looking at the consumption of music on a mobile phone, apps facilitate the choice of music and hardware integration, e.g., mp3 decoders on chip facilitate the intended usage by saving battery life and achieving good quality audio. But when it comes to music production, a deeper integration is still lacking. While services such as Splice[4] or Waves[5] offer access to very large collections of audio samples, audio effects and sound synthesisers, which seems more convenient than storing and managing gigabytes worth of samples and software instruments on one's local device, such collections remain hard to navigate, partly because the navigation process lacks an understanding of the user's audio production context. Thus, a better integration between audio production hardware, music production processes and very large collections of audio samples/effects/synthesisers should improve the music producers' experience of searching the sounds that they need to build their musical identity.

4) HARDWARE COMMODITISATION, STANDARDISATION AND LOGISTICS

The audio industry still largely relies on specialised audio processing chips such as digital signal processors (DSPs) and field-programmable gate arrays (FPGAs) to achieve low-latency, high quality audio processing [35]. This specialism comes with the downside of fragmented development practices across a variety of software development kits (SDKs) and programming standards. It also means lower buying volumes, which entail a limited influence on chip designs and a lack of resilience against chip supply shortages. In comparison, global electronics are dominated by personal devices, e.g., mobile phones and tablets, and mass-market computing, e.g., personal computers, cloud computers and gaming consoles, which have led to the commoditisation of CPU, GPU and ARM architectures. DSPaaS proposes that under suitable specifications, running various audio services which are competitive in latency and quality, but in the cloud and on commodity hardware instead of specialised hardware, would benefit the audio industry in terms of better supply chain resilience, better standardisation of development practice, economies of scale and simpler logistics of delivery to users.

III. COMPONENTS OF A DSPAAS SYSTEM

DSPaaS operates as a technology stack which blends audio engineering, digital signal processing, computing and networking components, as illustrated in Figure 1. The more specific design choices for each component are linked with the requirements of specific musical services. For instance, all services rely on an Internet protocol, but the specific choice of protocol may depend on service requirements, e.g., a type of protocol may be reliable at the expense of high latency, whereas another type of protocol may sacrifice audio continuity in aid of low latency. Starting from the input sound, the layers are: audio acquisition, audio encoding, audio data transport, computation, and network topology. These are detailed hereinafter.

A. THE AUDIO ACQUISITION LAYER

This layer transforms physical sound waves into digital samples or the other way around with an analog-to-digital converter (ADC) or digital-to-analog converter (DAC), respectively. The latency triggered by this part of the system varies with the application's sampling rate and audio frame buffer size, but can reach below 2 ms at a high sampling rate and low buffer size [36], [37].

B. THE AUDIO ENCODING LAYER

This layer may be used to encode the digital sound samples in a particular format before sending the audio via the

[3]https://www.audiotool.com/
[4]https://splice.com/
[5]https://www.waves.com/

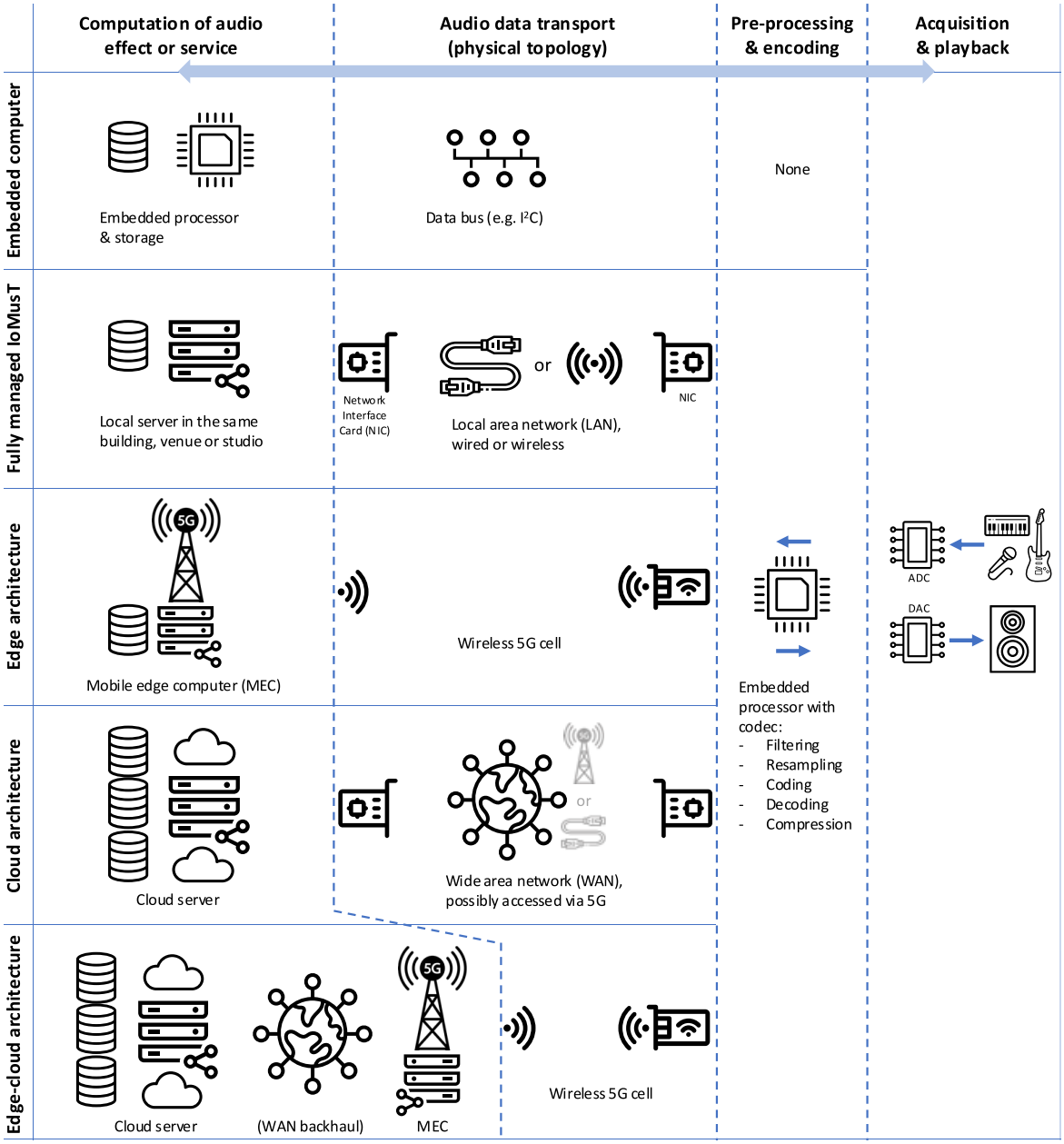

FIGURE 1. Common components of a DSPaaS system: audio acquisition, pre-processing/encoding, audio transport, computing unit.

network. The design of audio codecs is driven by the trade-off between bit-rate, loss of audio information and speed of encoding. Pulse code modulation (PCM) means sending the uncompressed samples. It results in a high bit rate, e.g., 1.411 Mbits/s for mono CD quality with 16 bit samples at 44.1 kHz sampling rate, or 1.536 Mbits/s for high-end 32 bits floating point samples at 48 kHz, but no loss of audio information besides the sampling process. The Free Lossless Audio Codec (FLAC) can reduce the bit rate to roughly 0.8-1.0 Mbps depending on the audio contents, but at the cost of computational and algorithmic latency.[6] Lossy formats such as MP3, AAC or Opus [20], [21] reduce the bit rate even further in a scalable way, typically down to 320-256 kbps for distributed audio files, or 256-128 kbps for streamed audio, even further down to 96 kbps for spoken word, by trading audio losses against bit rate in growing order of perceptual impact, and at the cost of algorithmic latency.[7] While a proportion of the encoding latency depends on the computation platform, e.g., hardware encoders/decoders for the fastest possible speed, it also and mostly depends on the algorithm's settings, e.g., required observation window. For example, Opus entails an algorithmic delay of 26.5 ms by design if using the default application settings, which can be reduced to 5 ms in the

[6]https://xiph.org/flac/
https://openbenchmarking.org/test/pts/encode-flac

[7]https://stsaz.github.io/fmedia/audio-formats/

codec's special restricted low delay mode, but at the cost of reducing the audio quality.

C. THE AUDIO DATA TRANSPORT LAYER

This layer refers to the networking standard specifically aimed at transporting audio. It interleaves various standards for low-level network transport protocols, higher level audio-over-Internet protocols and hardware/software infrastructures:

1) THE NETWORK TRANSPORT PROTOCOL

This is the norm by which the encoded samples are encapsulated into Internet packets which contain addressing information and are transmitted through the network's infrastructure. There are many transport protocols, each with their pros and cons, so it is difficult to list them all, but we introduce some of them below by order of complexity. The transmission control protocol over Internet protocol (TCP/IP) includes handshake, acknowledgement, re-transmission and order preservation procedures to guarantee that packets do not get lost and arrive in order, thus meaning continuous audio but at the expense of high latency. Conversely, the user datagram protocol (UDP) relies entirely on the network's performance for transmission: it does not control that the packets reach their destination nor that they arrive in order, thus it entails less procedural latency but at the cost of potential audio stream interruptions if some packets get lost, e.g., due to network congestion. The real-time transport protocol (RTP) is based on UDP and adds ordering information in the packets' header, thus allowing to rebuild the sequential order at the destination, but at the cost of extra wait for out-of-order packets. The secure reliable transport (SRT) protocol [38] stands between TCP/IP and RTP by trimming down the procedural bulk of TCP/IP but keeping some lightweight acknowledgement and re-transmission procedures. SRT is thus saving on execution time, but cannot get past the latency implications of acknowledging receipt, triggering re-transmission or re-ordering the audio chunks. The hyper-text transfer protocol (HTTP) and Apple's HTTP live streaming (HLS) standards [39] can be used to stream media to browsers or HTTP-enabled devices, but with a very high delay which is suitable for one-way broadcasts, e.g., live concerts with several seconds delay, but unsuitable for real-time musical interactions. Alternatively, Web real-time communication (WebRTC) [40] is a set of APIs built around UDP, designed from the ground up for real-time peer-to-peer communications such as teleconferencing, where reliability is handled by networking architecture beyond the protocol, e.g., using external servers to implement session traversal utilities for NAT (STUN). However WebRTC is lossy, as it relies on the Opus codec and doesn't guarantee 100% packet arrival, and its latency is of the order of 100 ms, which is OK for a conversation but not for precise music synchronisation. Looking at all the protocols globally, latency against reliability stands out as an unavoidable trade-off.

2) AUDIO OVER INTERNET PROTOCOL (AOIP) STANDARDS

Standards such as AES67 [17], RAVENNA [18], SMPT2110 [41] or Dante [19], make specific choices about transport protocol, stream management, quality-of-service mechanism, time synchronisation, and device discovery. They achieve hard latency specs by requiring specific routing software and hardware standards across the network, e.g., DiffServ [42] to prioritise audio traffic or audio/video bridging (AVB) [43] to achieve deterministic transmission times. As such, they require fully managed networks, which makes them untractable for general-public DSPaaS offerings over commodity Internet. Thus, DSPaaS is more likely to rely on lower level transport protocols than on established AoIP standards.

D. THE COMPUTATION OF THE AUDIO EFFECT OR SERVICE

Computation is traditionally operated locally on a music production device, e.g., a digital mixing console or a digital audio workstation on a laptop. The main paradigm shift operated by DSPaaS is the displacement of this computation to a remote computing unit, via networked audio transport, to access the benefits outlined in Section II-B. This assumes that remote facilities are able to provide more computational power, i.e., an embedded processor offers less computational power than a laptop, which offers less than a local server, which offers less than a Multi-Access Edge Computing (MEC) server, which offers less than a data centre's cloud server [44], [45]. So the crux of DSPaaS is to balance multiple trade-offs, one of them between the computational power required by the service and the latency and reliability of audio data transport. As such, DSPaaS design requires to combine the understanding of DSP computation, networking components and networked service architecture.

E. NETWORK TOPOLOGY

The topology of the network is a key design decision for DSPaaS, and addresses various factors:

1) NETWORK TRANSMISSION SPEED

is physically bounded by the speed of light in the best case scenario of optical fibers or electromagnetic spectrum. Concretely, London to Sydney represents a surface distance of 17,000 km, which corresponds to 57 ms travel time at the speed of light. For comparison, London to Paris is about 400 km which could be covered in 1.3 ms at best, whereas a 10 ms latency corresponds to 3000 km distance at most between the end-points of an optically transmitted audio stream. The network infrastructure can be fully managed to make sure that each section implements the fastest technology available. However, consumer Internet applications are more likely to traverse a Wi-Fi router, a 5G mobile link or a domestic section of copper cables before reaching the optical fiber, with transmission speeds several orders of magnitude lower than the speed of light.

2) CONGESTION AND TRAFFIC MANAGEMENT

large sections of the network remain a shared medium and are thus subject to congestion. E.g., the reliability of 5G and Wi-Fi diminishes as a function of the number of users tethered to the same tower or router, because there is only a finite number of allowed radio channels, each with a bounded spectrum and finite bandwidth. Network operators mitigate congestion risks by deploying wider and wider bandwidth capacities, but the solution also relies on network traffic management. Here two strategies apply: (a) keep the existing routes but apply traffic shaping to sort out the transmission priorities depending on the type of traffic, e.g., more priority for audio and less priority for emails, and (b) steer certain services across specific routes, e.g., with content delivery networks (CDNs) which cache the delivered data and services closer to their user. Broadcasting studios avoid congestion by extending their fully managed network across remote sites via rented "dark fibers" which offer guaranteed unshared bandwidth. Alternatively, video platforms such as YouTube or TikTok resort to CDNs.

3) FLEXIBLE NETWORK TOPOLOGY

distributed approaches to cloud computing [46] include (i) fog computing, which considers using the "ambient" computing power of network installations such as routers and network nodes; (ii) edge computing, which proposes to locate a proportion of computing power at the edge of wireless networks or CDNs; (iii) cloud computing, which proposes to centralise the bulk of the computation in a remote data centre. Hybrid edge-cloud approaches can also be deployed [46]. The optimisation of service distance, latency, congestion avoidance and thus service quality can be automated with software-defined wide area networking (SD-WAN) [47], [48], [49], where the data plane, i.e., the part of the network which carries user traffic, is managed by an orchestration plane, i.e., a set of applications which provide traffic monitoring, engineering and security, via the control plane, a server which forms an abstract view of the network and steers the data plane's networking components via a set of standard protocols. SD-WAN is mostly deployed for large enterprise networks. It could reduce latency and packet loss for DSPaaS, but for consumer services there might still be a non-SD-WAN last mile to cover, with lesser reliability and higher latency, before reaching an SD-WAN gateway.

IV. DEFINING LATENCY TIERS FOR DSPAAS

After illustrating the degrees of freedom and specific choices available when designing DSPaaS, we define three types of DSPaaS latency tiers and related use cases, depending on the required service response time: asynchronous, reactive, and low-latency DSPaaS. These categories increase in their level of complexity, as a function of the increasingly stringent latency requirement imposed on round-trip communication between the user and the server.

A. ASYNCHRONOUS DSPAAS

In this category, the user can tolerate response times of an order between a few seconds and "next day delivery" if the service's value justifies the wait. Examples are:

1) MUSIC PRODUCTION AS A SERVICE

This example refers to mixing and mastering remotely, either by AI or by human agents crowd-sourced via the cloud [50]. Whereas a regular supplier may take one day or more to create the mixing/mastering of a music project, at a cost mostly reserved to professionals, music production as a service provides quicker delivery due to a larger agent pool, lower costs due to economies of scale, and thus wider accessibility of music production services to hobbyists and consumers, while remaining within asynchronous latency requirements.

2) DEVICE CUSTOMISATION

This example refers to use cases where the parameters of a device get configured after some processing is applied on a server to an uploaded audio sample, and where the device's parameters do not need to change very frequently. For instance, in digital room correction for playback devices, measurements of the impulse response of a room are uploaded and processed by a server, and the room correction parameters, e.g., resonance equalisation or phase control parameters, are sent back to the device to improve the perceived playback quality. In the context of musical performance rather than audio perception, a smart musical instrument's sound engine can be configured by presets recommended and downloaded from a server [51], or its functionality can be flexibly adapted by machine-learning a model in the cloud, e.g., to control stage lights or other peripherals upon recognition of bespoke musical phrases [52]. User value in those cases is to improve the device's performance or to adapt its functionality with algorithms which would be too complicated or too computationally demanding to run locally on the device. They also justify additional revenue on top of device sales.

3) ASSISTANCE TO COMPOSITION, ARRANGEMENT OR LYRICS

A cloud-based AI agent receives partial musical data from an artist and is tasked with completing or conditionally predicting other parts of the music project for which the music composer or producer lacks skill or inspiration. Depending on the application, the data can be audio, e.g., the recording of a melody or a beat, symbolic, e.g., the MIDI encoding of a melody or a beat, or textual, e.g., some lyrics or the textual description of a desired outcome. The delivered result can be some music that complements the input data, e.g., an accompaniment of the melody [53], some form of arrangement, e.g., an orchestral arrangement built around a set of chords, or some complementary lyrical suggestions. The composer or producer can then use the received elements in their composition, or take inspiration from these and

adapt them manually. User value resides in quicker delivery and access to missing skills at an affordable cost to realise a musical idea, but also in providing stimulation to help music composers or lyric writers to get out of writer's block. Latency expectations may vary as a function of the perceived complexity of the task, e.g., a set of chords, a melody or a lyrical stimulus may be expected to come in seconds, whereas next-day delivery of an orchestral arrangement may be acceptable.

4) QUERY BY PLAYING

A user interrogates an online repository of musical content via the act of playing, e.g., [31]. An AI agent analyzes the recorded audio file and extracts features used to perform a query on the repository to return a corresponding set of music tracks or stems. In this scenario the latency should be minimized as much as possible, but it is not critical for the user to receive the requested content immediately. User value is fast access to very large collections of music tracks or samples stored in the cloud, but also the capability to express a query as a musical template rather than a verbal description. E.g., describing a synthesizer's sound can be difficult for someone who is not familiar with synthesis techniques: "I want a bandpass filtered sawtooth oscillator modulated by a LFO for vibrato" contains lots of knowledge, whereas showing an example of the sound may be easier. Similarly, query by humming [54] is a type of service based on audio queries rather than text queries.

B. REACTIVE DSPAAS

This category consists of services with a response time between tens of milliseconds and a few seconds, where these boundaries are not rigid but are consistent with human gestural reaction times, and tolerable by musicians, producers or engineers for responsive automation tasks. Examples are:

1) REACTIVE MIXING AS A SERVICE

This example relates to the live performance case where a cloud-based AI agent performs real-time mixing of the sounds coming from an ensemble of instruments. A latency of hundreds of milliseconds is consistent with the reaction time of a live sound engineer noticing the necessary changes and operating a mixing console by hand. User value is to assist the engineers by automating particular tasks, e.g., initial presets during a sound check where the musicians do not want to remain on stage for a very long time.

2) BROADCAST PROCESSING

Audio may be transmitted one way between its creator and its consumer, e.g., in the case of live media broadcast or internet radio, where the audience does not need to interact immediately with the audio creator and a latency of the order of a few seconds is still defined as a "live". The choice of audio processing, e.g., different types of mastering depending on the broadcast destination, could be automated with AI. User value here is a better personalisation of the audio rendering to the destination of the broadcast, and remains compatible with some seconds of delay.

3) MUSIC TRANSCRIPTION AS A SERVICE

The musician streams a recording in real-time to a server, which uses music transcription algorithms [55] to convert the audio stream to a symbolic representation, e.g., a MIDI score or some classical sheet music notation. The generated file is then streamed back to a visual display or to a document used by the musician. Just like spoken dictation services can tolerate some delay as long as they keep up with speech velocity, music transcription must happen as the performance goes but can accommodate delays of the order of fractions of a second. Like speech recognition, user value is to make a symbolic transcription of some audio contents readily available for subsequent usage.

4) MUSIC PERFORMANCE OR MUSIC RENDERING AS A SERVICE

A MIDI score or sheet music notation input, potentially complemented by metadata generated by a musical device, is converted to an audio file via a sound synthesis engine, using synthesis algorithms or instrument samples. The generated audio file is then sent back to the musician. This allows musicians to decouple the composition from its rendering and to try several renderings of the same piece, e.g., comparing a pop version against an orchestral version, or accessing and trying a larger range of timbres at later stages of the creation process. It is also a route towards licensing the timbre of specific or rare instruments via a cloud service, akin to licensing a singer's voice for other people's compositions. If considered for live rendering, the service can tolerate a latency commensurate with music streaming services.

5) DISTRIBUTED LIVE CODING

This is a musical practice which brings together music generation and real-time code writing in live performance, and where a server generates the sounds based on the commands sent to it. While live coding usually leverages local networks for co-located musicians, it can also leverage a cloud infrastructure to allow distributed live coding practitioners to perform together across larger distances [56]. In this case, the practitioners learn to cope with the delay between the launch of their written code and the sound generated.

C. LOW-LATENCY DSPAAS

In this category the service must respond with a latency of the order of 10–30 ms or below. This corresponds to the approximate threshold where humans cannot distinguish the delay between an action and its acoustic result, e.g., pressing a key and getting the resulting sound [57]. In the context of digital recording or live performance, the processing must be that fast because delays may accumulate across other components of the processing chain. Besides, in the commercial space, analog processing sets a benchmark

for low latency that digital processing must match or exceed. Thus, high end digital audio interfaces and consoles usually operate at 2 ms latency or below. Examples of services at this latency tier are:

1) LIVE MIXING AS A SERVICE

During a live performance, a remote processor performs the effects, mixing or mastering of the sounds coming from the instruments of an ensemble co-located with their audience. The latency must be as low as possible to be comparable to on-premise processing. There is demand from live sound engineers to be able to do their work remotely from home or from a safer or more comfortable location than the concert hall itself. They could also access, compare and perhaps license several versions of the live processing chain more flexibly.

2) LIVE PROCESSING CHAIN AS A SERVICE

This example is similar to the previous, but is more focused on the performer than on the engineer. A musician transmits their unprocessed performance in real-time to a server, which applies the desired effects chain, e.g., compression, chorus, reverb, etc., and returns the processed signal back to the musician or to the stage loudspeakers (see, e.g., [58]). For the musician, the value is reliable access to the same effects chain everywhere, in order to guarantee the consistency of their performance regardless of its location, or flexible trial of a variety of processing chains. Licensing models can be built around particular types of processing chains and sound textures, similarly to Waves' StudioVerse marketplace[8] but for low-latency live applications.

3) LIVE SYNTHESIS AS A SERVICE OR CLOUD-BASED INSTRUMENTS

A musician plays an instrument which only sends control messages to a sound synthesis engine running on the server. The latter streams back the synthesized sounds to the musician with the same latency as if they were generated by the instrument itself. This is similar to the music rendering service of the reactive latency tier, but with much more stringent latency requirement. User values are live access to a wider variety of timbres than what can be stored locally on a device, enhanced navigation of the available sounds, and the possibility to access or license the timbre of specific or rare instruments.

4) TELEPRESENCE AND METAVERSE

Telepresence and metaverse and metaverse applications require very low latency to maintain a sense of reality. E.g., small delays between synthetic speech and lip syncing may impede the sense of presence. Music rendering in telepresence services must ensure that the visual and acoustic aspects of the actions rendered by the system are perfectly synchronised - e.g., the hitting of a drum, the reverberation related to a virtual space, or more generally a performance as a virtual band [59].

[8]https://www.waves.com/studioverse

TABLE 1. Latency tier versus networking constraints.

Latency tier	Audio codec	Transport protocol	Network topology
Async.	Any codec	TCP/IP-based	Commodity internet
Reactive	Any codec	TCP/IP, SRT or WebRTC	Commodity internet
Low latency	PCM - no time to encode	UDP - no time to retransmit	Fully managed, edge, CDN, SD-WAN

Table 1 introduces the trade-offs that each latency tier imposes on DSPaaS design.

V. CASE STUDY ON THREE DSPAAS ARCHITECTURES

In this section, we analyse prototypes of the asynchronous, reactive and low-latency DSPaaS tiers, instantiated as various IoMusT architectures in wireless access cases. Figure 2 illustrates their main components and the upstream and downstream data flows occurring between the user and the server.

A. ARCHITECTURAL TRADE-OFFS

Centralised cloud architectures for musical services [60], [61] offer a low complexity of design and management, a high computational power, and large amounts of storage. However, they may suffer from high latency related to server distance and network congestion, particularly if the service involves concurrent access by a large number of musical devices.

Edge-based musical services [62] push the computing resources to the edge of network access points. By reducing communication distance and risks of congestion, the latency is reduced. However, distributed musical services suffer from higher management complexity, lower computational power and lower storage capacity compared to a centralised cloud-computing architecture [63].

In edge-cloud computing [46], the features of both centralized and distributed musical services are combined to offer a trade-off between access latency, computational power and storage capacity. This, however, entails more complex decisions on which tasks need to be performed on the embedded device, on the edge server, and on the cloud server. Moreover, the reliability of the service might be affected by the multiplication of device interconnections, themselves sources of packet loss, compared to the purely edge- and cloud-computing cases. Besides, the exact type of communication between the cloud and the edge depends on the application at hand: the transmission can be either of data or of service functions that need to be installed locally.

Table 2 compares the features of the various musical service architectures, while the rest of the section describes the key performance indicators (KPIs) of latency and

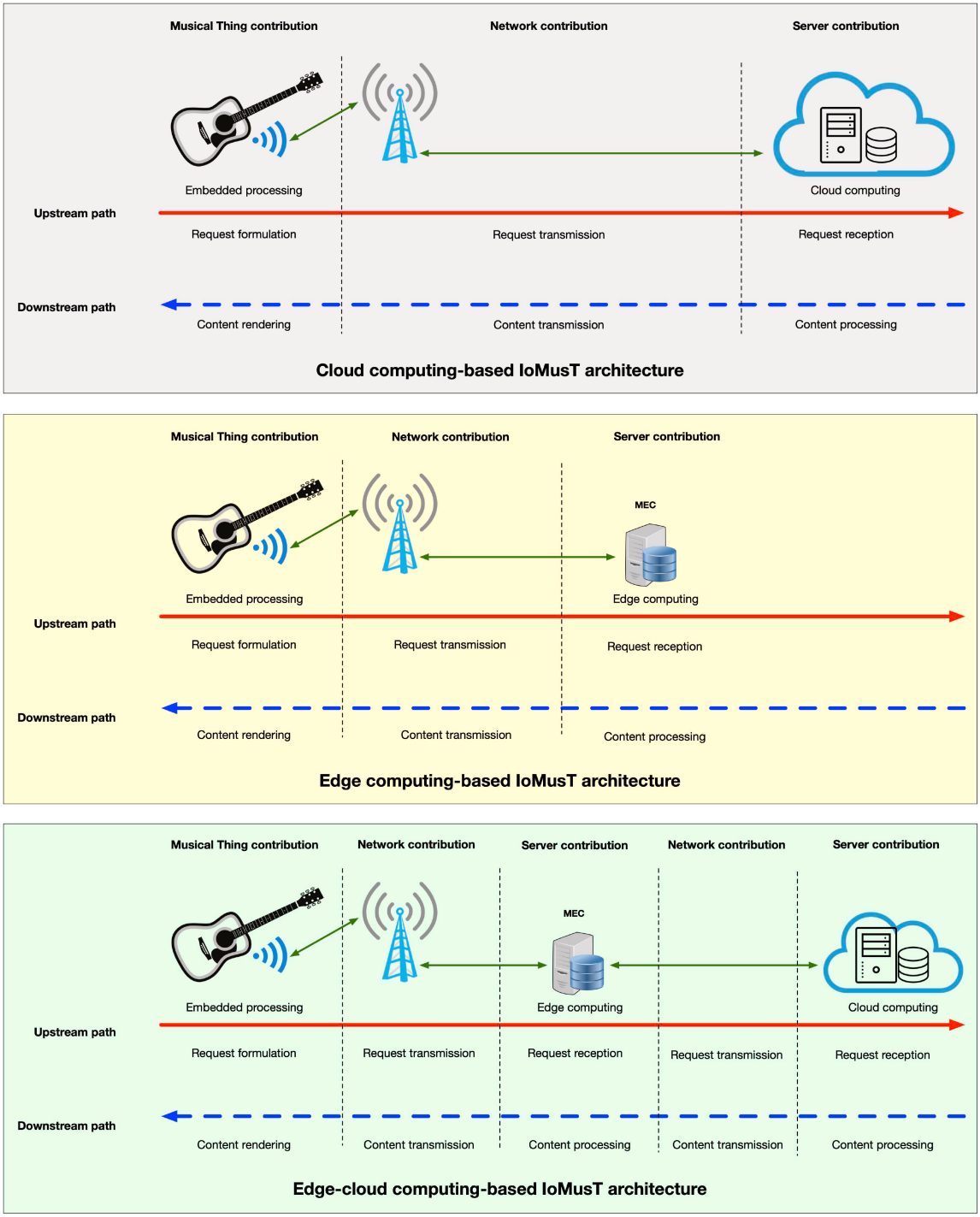

FIGURE 2. Three IoMusT architectures supporting DSPaaS.

reliability for the implemented DSPaaS prototypes. This illustrates how the latency tiers from Section IV can be satisfied in practice.

B. KEY PERFORMANCE INDICATORS: LATENCY AND RELIABILITY

For the cloud computing and edge computing IoMusT architectures, the latency between signal generation in a musical device and service delivery to the musician is defined as:

$$\begin{aligned}\mathcal{L}_{\text{cloud|edge}} = {} & \tau_{\text{request}} + \tau_{\text{uplink}} + \tau_{\text{upstream}} \\ & + \tau_{\text{cloud|MEC rcpt}} + \tau_{\text{cloud|MEC proc}} \\ & + \tau_{\text{downstream}} + \tau_{\text{downlink}} + \tau_{\text{render}} \end{aligned} \quad (1)$$

For the edge-cloud computing-based IoMusT architecture, in the scenarios involving just an initial configuration of the

TABLE 2. Comparison of the features offered by embedded, edge, cloud, edge-cloud architectures for musical services.

Metric / Architecture	Computational power	Available storage	Reliability	Latency	Scalability	Geographical distribution
Embedded	Lowest	Lowest	By design	Fastest	Null	Local
Edge	Medium	Medium	By design (e.g., 5G)	Fast	High	Distributed
Cloud	High	High	Topology-dependent	Slowest	Low	Centralised
Edge-cloud	Highest	Highest	Topology-dependent	Medium	Medium	Hybrid

edge server from the cloud upon the service request:

$$\mathcal{L}_{\text{edge-cloud}} = \mathcal{L}_{\text{edge}} + \tau_{\text{MEC init}} \tag{2}$$

For the edge-cloud computing-based IoMusT architecture, when continuous or discrete interactions occur between the edge and the cloud:

$$\mathcal{L}_{\text{edge-cloud}} = \mathcal{L}_{\text{edge}} + \tau_{\text{upstream}} + \tau_{\text{cloud rcpt}} + \tau_{\text{cloud proc}} + \tau_{\text{downstream}}, \tag{3}$$

where

- τ_{request} is the time taken by the musical device to formulate the service request before passing it to the wireless transmission module, e.g., the time to digitise an analog audio signal and to put it in packets;
- τ_{uplink} includes the wireless transmission module's processing time at both the transmitter and base station sides, plus the transmission time over the wireless link;
- τ_{upstream}, for edge and cloud architectures, is the delay caused by the transmission of the service request from the base station to the server; for edge-cloud architectures this time also includes the delay caused by the transmissions from the MEC server to the cloud server;
- $\tau_{\text{cloud|MEC rcpt}}$ is the time taken by the cloud server or the MEC server to receive and manage the incoming service request; in cloud architectures the request is generated by the musical device, whereas in edge-cloud architectures the request is generated by the MEC;
- $\tau_{\text{MEC init}}$ is the time taken to perform the initial steps where the edge gets configured from the cloud; at steady state, after initialisation, this time is null;
- $\tau_{\text{cloud|MEC proc}}$ is the time taken by the cloud server or the MEC server to process the requested service;
- $\tau_{\text{downstream}}$ for edge and cloud architectures, is the delay caused by the transmission of the service content from the server to the base station; for edge-cloud architectures this time also includes the delay caused by the transmissions from the cloud server to the MEC server;
- τ_{downlink} is the counterpart of τ_{uplink} and includes the wireless transmission's processing time at both the base station and the transmitter sides, plus the transmission time over the wireless link. Note that, due to the different direction of the transmission (downlink vs. uplink), it is likely that $\tau_{\text{downlink}} \neq \tau_{\text{uplink}}$;
- τ_{render} is the time taken by a musical device to render the requested service to the user.

Network reliability is typically defined by the packet error ratio (PER), i.e., the percentage of sent packets which reach the server within the time constraints set by the service, divided by the total number of sent packets. However, in the case of music, the PER should ideally be combined with the sequential distribution of lost audio information over time, because the number of consecutively lost packets has a specific impact on the audio quality perceived by the musician. For example, considering 100 seconds of transmission, 1% of packet loss can describe a single burst of 1000 ms of lost audio, or 100 equally distant 10 ms audio losses, whereas a single 1000 ms burst will be more perceivable than a 10 ms audio error. Besides, packet loss concealment (PLC) methods typically need to operate at zero delay in real-time scenarios in order to avoid the introduction of additional latency [64], [65], [66] and may be less effective to compensate appropriately in the case of long bursts. However, formulating the relation between the PER, the distribution of packet loss over time, the effect of PLC and the perceived audio quality is not trivial [25]. State-of-the-art remains the Perceptual Evaluation of Audio Quality (PEAQ), a standard of the International Telecommunication Union, which was designed as a generic metric of perceived audio quality characterised in terms of psychoacoustic effects rather than in terms of the specific effects of packet loss. Because the research efforts to find a better suited metric [67] are still ongoing, the measurements below will remain limited to the PER only, under the simplifying assumption that the perceived quality will evolve proportionally with the PER.

The end-to-end reliability can be decomposed into multiple reliability components as follows. For the cloud computing and edge computing IoMusT architectures:

$$\mathcal{R}_{\text{cloud|edge}} = p_{\text{uplink}} \cdot p_{\text{upstream}} \cdot p_{\text{downstream}} \cdot p_{\text{downlink}} \tag{4}$$

For the edge-cloud computing architecture, there is no change in reliability incurred by the initial steps where the edge gets configured from the cloud, i.e., $\mathcal{R}_{\text{edge-cloud}} = \mathcal{R}_{\text{edge}}$. On the other hand, for the edge-cloud computing architecture, when interactions occur between the edge and the cloud:

$$\mathcal{R}_{\text{edge-cloud}} = \mathcal{R}_{\text{edge}} \cdot p_{\text{edge-cloud,upstream}} \cdot p_{\text{edge-cloud,downstream}} \tag{5}$$

where

- p_{uplink} and p_{downlink} are the success probabilities of the uplink and downlink transmissions, respectively;
- p_{upstream} and $p_{\text{downstream}}$, for edge-based and cloud-based architectures, are the success probabilities of the packet forwarding between the base station and the server handling the request in the upstream and downstream directions, respectively; for edge-cloud architectures this probability also includes the packet losses caused by the transmissions between the MEC server and the cloud server, i.e., $p_{\text{edge-cloud,upstream}}$ and $p_{\text{edge-cloud,downstream}}$;

Note that we did not consider any packet losses or processing errors in the cloud server, edge server or musical device themselves, i.e., the reliability metric only applies to the network transmission components.

C. DSPAAS PROTOTYPES ACROSS VARIOUS ARCHITECTURES

1) ASYNCHRONOUS DSPAAS - FEW SHOTS LEARNING FOR PATTERN RECOGNITION

The prototype consists of a MIDI keyboard connected to a Raspberry Pi, which runs the Elk Audio operating system [37], enhanced with a HifiBerry audio shield. A VST3 audio plugin and a smartphone-based interface, specifically created for this prototype, allow musicians to record up to 10 MIDI patterns. The recorded patterns are sent via Wi-Fi, using the TCP/IP protocol, to a remote Linux server equipped with one Nvidia RTX 4090 GPU. Upon reception of the patterns, the server immediately performs a training procedure to learn a pattern recognition model, using the algorithm described in [52]. The produced model is sent back and gets automatically installed in the instrument.

This cloud-based architecture, illustrated in Figure 2 (top), was tested between a house in the city of Verona (Italy) and a server in the city of Trento (Italy), about 100 km apart, leveraging commodity Internet. The size of the streamed MIDI patterns was about 5 MB while the model generated by the server had a size of 21 MB. Table 3 shows the latency and reliability values measured over 5 sessions. In each session 4 patterns were utilized. The main contributions to latency were due to the transmissions of the content over the WAN and to the computations on the cloud, and the use of the TCP/IP protocol over a reliable network connection ensured the absence of packet losses.

2) REACTIVE DSPAAS - MUSIC RENDERING

This edge-based prototype, illustrated in Fig. 2 (middle), leverages a 5G testbed deployed indoors at the ZTE Innovation & Research Center in the city of L'Aquila (Italy). A musician records a music score on a MIDI-enabled guitar, sends it to a server, and the instrument receives an audio file back for reproduction. The guitar is equipped with a Fishman Triple Play Connect audio-to-MIDI converter, connected via a USB cable to a Raspberry Pi enhanced with a HifiBerry audio shield. This musical device runs the Elk Audio operating system [37] to host a headless VST3 audio plugin, itself controlled by a smartphone app to start/stop the recording and trigger the transmission of the resulting MIDI score to the remote server. Because the Rapsberry Pi was not natively equipped with a 5G module, it was connected via ethernet cable with a customer premise equipment (CPE) 5G module to communicate wirelessly with the base station. The average available bandwidth was 1000 Mbit/s in downlink and 270 Mbit/s in uplink. On the server side, a MEC is connected to the base station via a 1 m long optical fiber cable. Upon reception of the MIDI score, the MEC creates a mono audio file, sampled at 44.1 KHz and 16 bits in the.wav format, by running the Fluidsynth synthesizer with the General MIDI soundfont bank and the fast render flag enabled. Upon return of the resulting wav file to the smart guitar, the VST3 plugin plays it back to a loudspeaker via the HifiBerry shield. For the file transfer from the guitar to the server and vice versa, the secure copy protocol is utilized, which is based on TCP/IP.

Table 3 shows the latency and reliability values measured over 3 sessions, with a MIDI score size of 14 KB and a resulting audio file size of 12.7 MB. The total latency for the service is 1101.3 ms. Such a latency is not fit for live performance but it is suitable to render the composed music piece across a range of guitar tones and pick the best sounding one, in a way that is more flexible and more convenient than first writing the MIDI score then applying the various tones. As such, this application aims to improve the fluidity of the musician's creativity when it comes to seeking the best match between a set of chords and a tone.

3) LOW-LATENCY DSPAAS - FLEXIBLE REAL-TIME EFFECTS

This prototype implements the edge-cloud architecture illustrated in Figure 2 (bottom), where a smart electric guitar transmits its unprocessed string signals to a MEC server via 5G. The server applies a selected effect and returns the processed signal to the instrument. The processed sound is output directly by the guitar to the amplifier, thus bypassing the need for extra effect pedals. The conducted test leverages the same 5G testbed as used for the edge-based architecture described in Section V-C2, with the difference that the guitar and the edge-side MEC station are both connected with Elk Live devices [37] to handle the low-latency audio transmission. The guitar provides audio input to the first Elk Live device, which produces the audio packets that feed into the 5G CPE router for wireless upstream transmission to the MEC, via a base station located 3 m away. The MEC is connected via a short ethernet cable to the second Elk device, which runs a Python-based application to process the received audio signal with a selected audio effect. The effect can be flexibly changed by the user: a request for a particular effect, e.g., reverb, overdrive or any other choice, is sent to the MEC, which obtains the corresponding VST3 plugin from a server connected via a 10 m optical fiber, and installs it into the second Elk Live device. Thus, the edge device

TABLE 3. Mean latency and reliability metrics for each component of the implemented DSPaaS prototypes.

Latency	Asynchronous	Reactive	Low latency
τ_{request}	≈0.1 ms	≈0.1ms	1.83 ms
τ_{uplink}	101.4 ms	22.3 ms	4.44 ms
τ_{upstream}	12.2 ms	0.1 ms	≈0 ms
$\tau_{\text{MEC init}}$	N/A	N/A	≈1 ms
$\tau_{\text{cloud\|MEC rcpt}}$	124 ms	≈0 ms	≈0 ms
$\tau_{\text{cloud\|MEC proc}}$	3 min 37 s	763 ms	2.66 ms
$\tau_{\text{downstream}}$	12.9 ms	0.1 ms	≈0 ms
τ_{downlink}	287 ms	315.7 ms	4.43 ms
τ_{render}	1.4 ms	≈0 ms	8.49 ms
$\mathcal{L}$	3 min 37.549 s	1 s 101.3 ms	21.85 ms +1ms MEC init.

Reliability	Async.	Reactive	Low lat.
p_{uplink}	1	1	0.994
p_{upstream}	1	1	1
$p_{\text{downstream}}$	1	1	1
p_{downlink}	1	1	0.995
$\mathcal{R}$	1	1	0.989

is occupied only by the requested effect(s). After applying the effect, the second Elk Live device returns the processed signal back to the first Elk Live device, itself connected to the guitar amplifier.

Table 3 reports the latency and reliability values measured with Elk Live's system tools. Five measurement sessions were conducted, in each of which the guitarist played for 5 minutes. The latency of about 22 ms is above the 10 ms requirement: the electric guitarist reported that they could notice the latency but were able to tolerate it while playing. On the other hand, the lost packets had a significant impact on the perceived audio quality, despite the application of an auto-regressive packet loss concealment method [64].

VI. FUTURE RESEARCH DIRECTIONS

Researchers interested in DSPaaS should consider the following components for their research programmes:

4) OPTIMISATION OF AUDIO TRANSPORT SETUPS

Network congestion triggers packet loss and increased jitter, which in turn requires a larger buffer for correction. In other terms, network performance directly translates into latency and audio quality. Our tests over a state-of-the-art 5G infrastructure (3GPP R15), illustrated in Table 3, suggest that 5G remains too slow and too prone to errors to fully fit the requirements of lowest latency live music DSPaaS in the wireless case. The implementation of a dedicated 5G slice could be required, i.e., a portion of the radio network and 5G core with bandwidth and functions reserved for musical interactions [68]. More generally, a wider range of combinations of CDN, SD-WAN, use of a MEC and optimisation of the last mile between the WAN and the consumer remain to be researched and evaluated.

5) JOINT OPTIMISATION OF COMPUTE CAPABILITIES AND NETWORKED ARCHITECTURE

Table 3 indicates that the second largest source of latency is the computation of DSP algorithms. Indeed, DSPaaS operates on a three-pronged trade-off: audio quality versus network performance versus choice of computing architecture. Optimising data transfer and compute capability jointly isn't a new problem, e.g., the Map-Reduce approach arose about 20 years ago [69]. However, specialisation to audio, where the tolerance on latency is much stricter than Web services, remains a research topic. Audio computation is an evolving field; e.g., machine learning optimisation techniques such as neural network distillation, quantization, and pruning [70] can yield smaller and faster models, but may impact service performance with more classification errors or more signal processing distortions. MECs can be accessed with less latency but have much less computation and storage capacities than their centralised cloud counterparts. Computation could be distributed, e.g., between the embedded system of a musical device, a MEC station and a cloud server, but the guidelines on how to divide the computation may be service-dependent. So the trade-off between available computation and available locations remains to be jointly and specifically optimised for DSPaaS applications, and its operating point further improved.

6) IMPROVEMENT OF PACKET LOSS CONCEALMENT (PLC)

The third trade-off, illustrated in Section III, is latency versus predictability of packet arrival, e.g., TCP/IP is reliable but has high latency, while UDP has low latency but lets packet loss happen. PLC reframes this trade-off by balancing network reliability against audio reconstruction performance. Although PLC is well studied for speech transmission, e.g., [71], producing a perceptually neutral compensation of network losses for musical signals, particularly in the presence of large error bursts, remains an open research topic. Deep learning techniques [65] offer a promising avenue, but they need to deliver predictions from past samples at low latency on the processor embedded at the receiving end, which calls for research on the trade-off between model size and reconstruction performance.

7) ENHANCING NETWORK AWARENESS AND INTELLIGENCE

Intelligent networks [72], [73], where artificial intelligence techniques are used in software-defined networking, e.g., for traffic prediction and consequent network adaptation, is a key technology to solve the complexity and trade-offs implied by DSPaaS, particularly in the low latency case. Research is necessary to specilaise this technology to DSPaaS applications.

8) EVALUATING NEW BUSINESS MODELS

DSPaaS is relatively new, so its economic viability, pro-duct/market fit, cost balancing and business strategy [72], [74] need to be more fully tried and validated. In particular, the technico-commercial agreements between musical device manufacturers and Internet service providers remain to be clarified.

VII. CONCLUSION

This article proposed a vision for DSPaaS, which exploits and generalises the IoMusT paradigm to define novel services for musicians. First, we analysed the IoMusT roots of the DSPaaS concept, then its driving values which consist in lowering the barriers to making music, supporting ubiquitous creativity, supporting collaborative audio contents production, accessing larger collections of sound samples, effects and synthesisers, and commoditising digital audio production setups. After describing the components of a DSPaaS system in terms of audio acquisition/playback, encoding, transport and effect computation, we proposed to segment DSPaaS applications in three categories defined by different latency constraints and related music production applications. To illustrate these categories, we implemented three prototypes for a case study where we identified and measured the KPIs to be considered for the evaluation of DSPaaS solutions, and identified the technical challenges and trade-offs at play between network latency/reliability, networking architecture and computational power. Finally, we proposed a set of directions for research, along the lines of network latency and reliability, choice of architecture, packet loss concealment methods, network awareness/intelligence and viability of business models. This informs designers and developers willing to create new digital services for musicians and audio content producers, as well as researchers seeking to structure new research directions in the field of digital music production. Thus, we expect that the conceptualisation of DSPaaS delivered by this article will lead to radically new and valuable types of interactions between musicians and musical content in a large variety of situations, such as playing alone or in group, learning, composing, recording and performing music, either professionally or recreationally.

ACKNOWLEDGMENT

The authors are grateful to Daniele Ciambrone and Domenico Puntillo, from company ZTE, and to Nishal Silva, from University of Trento, for their technical support. Sacha Krstulović is grateful to Miguel Pereira, now at Forvia, and Aldo Ricci, now at Merging Technologies, for early work and discussions on DSPaaS while at Music Tribe.

REFERENCES

[1] W. Sun, X. Zhang, C. J. Guo, P. Sun, and H. Su, "Software as a service: Configuration and customization perspectives," in *Proc. IEEE Congr. Services Part II (Services-2)*, 2008, pp. 18–25.
[2] B. Waters, "Software as a service: A look at the customer benefits," *J. Digit. Asset Manag.*, vol. 1, no. 1, pp. 32–39, 2005.
[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.
[4] W. Cai, M. Chen, and V. C. M. Leung, "Toward gaming as a service," *IEEE Internet Comput.*, vol. 18, no. 3, pp. 12–18, May/Jun. 2014.
[5] W. Cai et al., "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620, 2016.
[6] C.-H. Hong, I. Spence, and D. Nikolopoulos, "GPU virtualization and scheduling methods: A comprehensive survey," *ACM Comput. Surveys*, vol. 50, no. 3, pp. 1–37, 2017.
[7] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet, "Internet of Musical Things: Vision and challenges," *IEEE Access*, vol. 6, pp. 61994–62017, 2018.
[8] D. Keller and V. Lazzarini, "Ecologically grounded creative practices in ubiquitous music," *Org. Sound*, vol. 22, no. 1, pp. 61–72, 2017.
[9] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, and P. Herrera, "Semantic audio content-based music recommendation and visualization based on user preference examples," *Inf. Process. Manag.*, vol. 49, no. 1, pp. 13–33, 2013.
[10] E. Meneses, J. Wang, S. Freire, and M. M. Wanderley, "A comparison of open-source Linux frameworks for an augmented musical instrument implementation," in *Proc. Conf. New Interfaces Musical Expression*, 2019, pp. 222–227.
[11] B. Matuszewski, "A Web-based framework for distributed music system research and creation," *J. Audio Eng. Soc.*, vol. 68, no. 10, pp. 717–726, 2020.
[12] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.
[13] L. Gabrielli and S. Squartini, *Wireless Networked Music Performance*. Singapore: Springer, 2016.
[14] L. Turchet, "Smart musical instruments: Vision, design principles, and future directions," *IEEE Access*, vol. 7, pp. 8944–8963, 2019.
[15] R. Hupke, J. Dürre, N. Werner, and J. Peissig, "Latency and quality-of-experience analysis of a networked music performance framework for realistic interaction," in *Proc. Audio Eng. Soc. Conv.*, 2022, pp. 11–20.
[16] L. Turchet and P. Casari, "Latency and reliability analysis of a 5G-enabled Internet of Musical Things system," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 1228–1240, Jan. 2024.
[17] "AES standard for audio applications of networks—High-performance streaming audio-over-IP interoperability," 2018. [Online]. Available: https://www.aes.org/publications/standards/search.cfm?docID=96
[18] A. Hildebrand, "RAVENNA & AES67," 2014. [Online]. Available: https://ravenna-network.com/wp-content/uploads/2020/02/RAVENNA_AES67_V1.0.pdf
[19] "TI 317 dante audio networking 1.1," 2016. [Online]. Available: https://www.dbaudio.com/assets/products/downloads/ti/dbaudio-technical-information-ti317-1.1-en.pdf
[20] K. Brandenburg, "MP3 and AAC explained," in *Proc. AES 17th Int. Conf. High Qual. Audio Coding*, Sep. 1999, pp. 1–12.
[21] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-quality, low-delay music coding in the opus codec," in *Proc. 135th Conv. Audio Eng. Soc.*, Oct. 2013, pp. 1–10.
[22] X. Jiang et al., "Low-latency networking: Where latency lurks and how to tame it," *Proc. IEEE*, vol. 107, no. 2, pp. 280–306, Feb. 2019.
[23] L. Vignati et al., "Is music in the air? Evaluating 4G and 5G support for the Internet of Musical Things," *IEEE Access*, vol. 12, pp. 38081–38101, 2024.
[24] A. Carôt, M. Dohler, S. Saunders, F. Sardis, R. Cornock, and N. Uniyal, "The world's first interactive 5G music concert: Professional quality networked music over a commodity network infrastructure," in *Proc. Sound Music Comput. Conf.*, Jun. 2020, pp. 407–412.

[25] J. Dürre et al., "In-depth latency and reliability analysis of a networked music performance over public 5G infrastructure," in *Proc. Audio Eng. Soc. Conv.*, vol. 153, 2022, pp. 1–11.

[26] F. Cheli and S. Giordano, "Service parameters identification for adaptive networked music performance," in *Proc. Global Inf. Infrastruct. Netw. Symp.*, 2022, pp. 94–98.

[27] L. Turchet, C. Rinaldi, C. Centofanti, L. Vignati, and C. Rottondi, "5G-enabled Internet of Musical Things architectures for remote immersive musical practices," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 4691–4709, 2024.

[28] E. R. Miranda, *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*. Cham, Switzerland: Springer Nat., 2021.

[29] D. Makris, G. Zixun, M. A. Kaliakatsos-Papakostas, and D. Herremans, "Conditional drums generation using compound word representations," in *Proc. EvoMUSART*, 2022, pp. 1–13.

[30] J. Martinez-Avila, C. Greenhalgh, A. Hazzard, S. Benford, and A. Chamberlain, "Encumbered interaction: A study of musicians preparing to perform," in *Proc. Conf. Human Factors Comput. Syst.*, 2019, pp. 1–13.

[31] L. Turchet, J. Pauwels, C. Fischione, and G. Fazekas, "Cloud-smart musical instrument interactions: Querying a large music collection with a smart guitar," *ACM Trans. Internet Things*, vol. 1, no. 3, pp. 1–29, 2020.

[32] L. K. Fink, L. A. Warrenburg, C. Howlin, W. M. Randall, N. C. Hansen, and M. Wald-Fuhrmann, "Viral tunes: Changes in musical behaviours and interest in coronamusic predict socio-emotional coping during COVID-19 lockdown," *Humanities Social Sci. Commun.*, vol. 8, no. 1, pp. 1–11, 2021.

[33] A. Daubney and M. Fautley, "Editorial research: Music education in a time of pandemic," *Brit. J. Music Educ.*, vol. 37, no. 2, pp. 107–114, 2020.

[34] K. E. Onderdijk, F. Acar, and E. Van Dyck, "Impact of lockdown measures on joint music making: Playing online and physically together," *Front. Psychol.*, vol. 12, May 2021, Art. no. 642713.

[35] M. Popoff, R. Michon, T. Risset, Y. Orlarey, and S. Letz, "Towards an FPGA-based compilation flow for ultra-low latency audio signal processing," in *Proc. Sound Music Comput. Conf.*, 2022, pp. 1–8.

[36] A. McPherson and V. Zappi, "An environment for submillisecond-latency audio and sensor processing on BeagleBone black," in *Audio Eng. Soc. Conv.*, vol. 138, 2015, pp. 1–9.

[37] L. Turchet and C. Fischione, "Elk audio OS: An open source operating system for the Internet of Musical Things," *ACM Trans. Internet Things*, vol. 2, no. 2, pp. 1–18, 2021.

[38] "Technical overview: SRT secure reliable transport," 2018. [Online]. Available: https://www.haivision.com/resources/white-paper/

[39] J. G. Min and Y. Lee, "High-quality HTTP live streaming system for limited communication bandwidth," in *Proc. Int. SoC Design Conf. (ISOCC)*, 2020, pp. 113–114.

[40] B. García, F. Gortázar, M. Gallego, and A. Hines, "Assessment of QoE for video and audio in WebRTC applications using full-reference models," *Electronics*, vol. 9, no. 3, p. 462, 2020.

[41] *Professional Media Over Managed IP Networks: PCM Digital Audio*, SMPTE standard ST 2110-30:2017, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8167392

[42] C. Wang, K. Long, J. Yang, and S. Cheng, "An effective feedback control mechanism for DiffServ architecture," *J. Comput. Sci. Technol.*, vol. 17, no. 4, pp. 420–431, 2002.

[43] A. Holzinger and A. Hildebrand, "Realtime linear audio distribution over networks: A comparison of layer 2 and 3 solutions using the example of ethernet AVB and Ravenna," in *Proc. 44th Int. Conf. Audio Netw. Audio Eng. Soc. Conf.*, 2011, pp. 1–9.

[44] S. Guan and A. Boukerche, "Intelligent edge-based service provisioning using smart cloudlets, fog and mobile edges," *IEEE Netw.*, vol. 36, no. 2, pp. 139–145, Mar./Apr. 2022.

[45] H. Chen, T. D. Todd, D. Zhao, and G. Karakostas, "Wireless and service allocation for mobile computation offloading with task deadlines," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5054–5068, May 2024.

[46] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surveys*, vol. 52, no. 6, pp. 1–36, 2019.

[47] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[48] A. Ryhans et al., *CISCO SD-WAN Cloud Scale Architecture*, CISCO, San Jose, CA, USA, 2019.

[49] S. Troia, L. M. Moreira Zorello, and G. Maier, "SD-WAN: How the control of the network can be shifted from core to edge," in *Proc. Int. Conf. Opt. Netw. Design Model.*, 2021, pp. 1–3.

[50] B. De Man, R. Stables, and J. D. Reiss, *Intelligent Music Production*. New York, NY, USA: Routledge, 2019.

[51] L. Turchet and P. Bouquet, "Smart musical instruments preset sharing: An ontology-based data access approach," in *Proc. IEEE World Forum Internet Things*, 2021, pp. 1–6.

[52] N. Silva and L. Turchet, "Real-time pattern recognition of symbolic monophonic music," in *Proc. Int. Audio Mostly Conf.*, 2024, pp. 308–317.

[53] A.-M. Gioti, "Artificial intelligence for music composition," in *Handbook of Artificial Intelligence for Music*. Cham, Switzerland: Springer, 2021, pp. 53–73.

[54] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo, "The MUSART testbed for query-by-humming evaluation," *Comput. Music J.*, vol. 28, no. 2, pp. 34–48, 2004.

[55] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 20–30, Jan. 2019.

[56] A. D. de Carvalho Jr., S. W. Lee, and G. Essl, "SuperCopair: Collaborative live coding on supercollider through the cloud," in *Proc. Int. Conf. Live Coding*, 2015, pp. 1–9.

[57] R. H. Jack, T. Stockman, and A. McPherson, "Effect of latency on performer interaction and subjective quality assessment of a digital musical instrument," in *Proc. Audio Mostly Conf.*, 2016, pp. 116–123.

[58] M. Buffa et al., "Web audio modules 2.0: An open Web audio plugin standard," in *Proc. Compan. Web Conf.*, 2022, pp. 364–369.

[59] L. Turchet, "Musical metaverse: Vision, opportunities, and challenges," *Pers. Ubiquitous Comput.*, vol. 27, pp. 1–17, Oct. 2023.

[60] A. D. de Carvalho Jr., M. Queiroz, and G. Essl, "Computer music through the cloud: Evaluating a cloud service for collaborative computer music applications," in *Proc. Int. Conf. Comput. Music*, 2015, pp. 226–233.

[61] N. Antonopoulos and L. Gillam, *Cloud Computing*. Cham, Switzerland: Springer, 2010.

[62] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[63] Z. Xu, W. Liang, M. Jia, M. Huang, and G. Mao, "Task offloading with network function requirements in a mobile edge-cloud network," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2672–2685, Nov. 2019.

[64] M. Fink and U. Zölzer, "Low-delay error concealment with low computational overhead for audio over IP applications," in *Proc. Int. Conf. Digit. Audio Effects*, 2014, pp. 309–316.

[65] A. I. Mezza, M. Amerena, A. Bernardini, and A. Sarti, "Hybrid packet loss concealment for real-time networked music applications," *IEEE Open J. Signal Process.*, vol. 5, pp. 266–273, 2024.

[66] M. Sacchetto, Y. Huang, A. Bianco, and C. Rottondi, "Using autoregressive models for real-time packet loss concealment in networked music performance applications," in *Proc. Int. Conf. Audio Mostly*, 2022, pp. 203–210.

[67] A. F. Khalifeh, A.-K. Al-Tamimi, and K. A. Darabkh, "Perceptual evaluation of audio quality under lossy networks," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, 2017, pp. 939–943.

[68] L. Turchet and P. Casari, "On the impact of 5G slicing on an Internet of Musical Things system," *IEEE Internet Things J.*, early access, Jul. 2, 2024, doi: 10.1109/JIOT.2024.3422287.

[69] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. 6th Symp. Oper. Syst. Design Implement.*, 2004, pp. 137–149.

[70] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, Oct. 2021.

[71] L. Diener, S. Sootla, S. Branets, A. Saabas, R. Aichner, and R. Cutler, "INTERSPEECH 2022 audio deep packet loss concealment challenge," in *Proc. Interspeech*, 2022, pp. 1–4.

[72] L. Mohjazi, B. Selim, M. Tatipamula, and M. A. Imran, "The journey toward 6G: A digital and societal revolution in the making," *IEEE Internet Things Mag.*, vol. 7, no. 2, pp. 119–128, Mar. 2024.
[73] S. Faezi and A. Shirmarz, "A comprehensive survey on machine learning using in software defined networks (SDN)," *Human-Centric Intell. Syst.*, vol. 3, pp. 312–343, Jun. 2023.
[74] G. S. Jahromi and S. Ghazinoory, "How to use bits for beats: The future strategies of music companies for using industry 4.0 technologies in their value chain," *Inf. Syst. e-Bus. Manag.*, vol. 21, pp. 1–21, Jun. 2023.

LUCA TURCHET (Senior Member, IEEE) received the master's degree (summa cum laude) in computer science from the University of Verona in 2006, the degrees in classical guitar and in composition from the Music Conservatory of Verona in 2007 and 2009, respectively, the Ph.D. degree in media technology from Aalborg University Copenhagen in 2013, and the degree in electronic music from the Royal College of Music of Stockholm in 2015. He is an Associate Professor with the Department of Information Engineering and Computer Science, University of Trento, Italy. He is Co-Founder of the Music-Tech Company Elk. He is the Coordinator of the European Project MUSMET funded by the European Innovation Council. His scientific, artistic, and entrepreneurial research has been supported by numerous grants from different funding agencies, including the European Commission, the European Institute of Innovation and Technology, the European Space Agency, the Italian Ministry of Foreign Affairs, and the Danish Research Council. He serves as an Associate Editor for IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, IEEE ACCESS, and the *Journal of the Audio Engineering Society*, and has been a Guest Editor for the *IEEE Communications Magazine*, the *Personal and Ubiquitous Computing*, the *Journal of the Audio Engineering Society*, *Frontiers in VR*, and *Digital Creativity*. He is the Chair of the IEEE Emerging Technology Initiative on the Internet of Sounds and the Founding President of the Internet of Sounds Research Network.

SACHA KRSTULOVIĆ (Senior Member, IEEE) received the Ph.D. degree from Lausanne's Swiss Polytechnic Federal Institute (EPFL) in 2001. He is an Independent Consultant of Artificial Intelligence for Audio Processing. Until May 2023, he was the Head of AI research at Music Tribe, a multimillion dollar audio equipment manufacturer. His first Post-doc with INRIA was on speaker recognition, and his second post-doc on expressive speech synthesis. From 2007 to 2011, he researched, developed, and promoted parametric speech synthesis as a Researcher with Toshiba Research Europe Ltd. Then, he joined Nuance's Advanced Speech Group as a Senior Research Engineer to research and develop voicemail-to-text transcription. In 2012, he became a Lead Research Engineer then the Director of Innovation with Audio Analytic Ltd., a startup, where he developed novel sound event recognition technology, and which was acquired by Meta in 2022. He is passionate about using his expertise in audio processing and artificial intelligence to introduce digital transformation in the audio industry.

Open Access funding provided by 'Università degli Studi di Trento' within the CRUI CARE Agreement