2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) | 978-1-6654-5381-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PERCOMWORKSHOPS56833.2023.10150324

Validating Quality of Context in Pervasive Computing Systems: Surf Life Saving Use Case

Kanaka Sai Jagarlamudi, Arkady Zaslavsky, Seng W. Loke, Kevin Lee
School of Information Technology
Deakin University
Geelong, Australia
(kjagarlamudi, arkady.zaslavsky, seng.loke, kevin.lee)@deakin.edu.au

***Abstract*—We present context Cost and Quality computational engine 2.0 (conCQeng 2.0, in short); this system addresses a significant drawback in Quality of Context (QoC) measurement models that lead to QoC-aware selection uncertainties in Context Management Platforms (CMPs). Current QoC measurement models rely on the QoC parameters in context (such as time-stamps) to assess QoC metrics, representing the context's usability for the pervasive computing applications and selecting the better-performing context providers. Nevertheless, such parameters are prone to misrepresentation, limiting the credibility of QoC measurement. In this paper, we propose a QoC validation mechanism through which conCQeng 2.0 determines the genuineness of measured QoC-metrics, further contributing to a credible QoC-aware selection. We motivate the proposal through its significance in the surf life saving use case. Our evaluation demonstrates that conCQeng 2.0 improves the credible QoC acquisition through the selection process - with slight but addressable processing overhead compared to its baseline version that attains higher QoC adequacy than heuristic models.**

***Index Terms*—QoC Validation, QoC-Aware Selection, Context Management Platforms, QoC Measurement**

I. Introduction

Adopting context-aware pervasive solutions for real-world problems has come a long way: they have evolved from a closed-loop system containing limited context sources to provide pervasive applications with seamless context (context describes the situations related to the entities (e.g., places) [1]). Google's traffic analysis [2] is an excellent example of such a solution; it finds the optimal routes for the drivers by dynamically acquiring the traffic information from the vehicles in relevant routes. In addition, context awareness in emergency service-related applications such as patient monitoring [3] and flood monitoring [4] has significantly reduced human errors.

The data providers (services providing context) and the applications (systems using context) in the context-aware pervasive environments are typically known as the *context providers* and *context consumers*. An increasing number of context providers on a global scale led to an inconsistent context space – having a diverse context provider with dynamically changing availability. Hence, *context management platforms* (*CMPs*, for short) emerged to connect context providers with consumers. CMPs provide the context consumers with seamless context access by querying and retrieving the context by selecting the relevant providers, thereby removing the storage and processing overheads on the context consumers.

Context's usability to the consumers depends on a quality dimension called the Quality of Context [5] (QoC, for short); it is assessed based on QoC metrics adequacy, e.g., timeliness and completeness adequacy, representing the context's freshness and information adequacy. CMPs measure and determine the context's QoC level by measuring and aggregating such metrics using QoC measurement models; works in [4, 6] discuss a few of them. However, the CMPs may attain false measurements through these models – as they rely on the QoC parameters (e.g., the time-stamp of context generation) delivered by the context providers. Unfortunately, such parameters are prone to misrepresentation and manipulation – as the context provider owners can manipulate them to their favour.

In this paper, we propose a component called context Cost and Quality computational engine 2.0 (conCQeng 2.0), which extends conCQeng [7], a component proposed to attain QoC measurement and QoC-aware selection in CMPs. The conCQeng 2.0 uses a novel method to obtain *veracity* in context providers, representing the truthfulness level of their QoC metrics. So including the veracity for QoC-aware selection overcomes issues in solely using the current QoC measurement models.

We present conCQeng 2.0 from the perspective of its application to surf life saving use case [8] (discussed in section 2). The conCQeng 2.0 analyses the veracity based on the context provider's quality of service (QoS) and Quality of device (QoD) metrics that affect the QoC metrics required by the consumer. The QoS metrics represent the performance of both cloud resources (context provider's data processing resources) and network resources (communication channel between the CMP and context providers). The Quality of Device (QoD) metrics represent the context provider's sensing features (e.g., coverage of thermal imaging). The context consumer's QoC requirements vary based on the situation. For instance, an emergency-handling application may need higher timeliness and resolution. Using a method based on Multi-Criteria Decision Making [9] through Analytical Hierarchy Process [10], the conCQeng 2.0 estimates the QoC validity and determines the veracity of context providers in fulfilling the QoC requirements.

The following is this paper's organization: section 2 discusses the surf life saving use case to motivate conCQeng 2.0; section 3 provides a literature review; Section 4 discusses the architecture and process flow in conCQeng 2.0; section 5 discusses the veracity measurement model; Section 6 discusses the implementation and evaluation details. Finally, section 7 concludes this paper and outlines future work.

II. Use Case: The Surf Life Saving

Analysing location-wise crowd density in recreational areas improves disaster handling and prevention applications. The surf life savers (personnel that responds to emergencies involving beachgoers) identify catastrophes (e.g., drowning persons or potential shark attacks) through surveillance of crowds [8]. These personnel use jet skis, surveillance helicopters and boats to identify possible catastrophes. However, human surveillance could be inefficient and inaccurate due to human resource costs and errors due to fatigue and adverse weather. Hence, analysing the location-wise crowd density to identify the emergencies using context-aware solutions leads to the effective deployment of the surf life savers from the managerial hub. It allows them to allocate more personnel to surveillance the heavily crowded areas and send the rescue teams to the crisis locations.

As Fig. 1 depicts, the context providers could relay the location-wise crowd density through Mobile crowd sensing from the smartphones of the beachgoers. Furthermore, water-resistant equipment such as smartwatches and surveillance cameras can detect and relay the possible crisis (e.g., drowning) by enduring the harsh water activities on beaches. The context providers deliver this context as raw data (Low-level context (C)); for example, through mobile crowd sensing, the context provider relays the count of people. The CMP further infers such information and sends it, along with incurred price, to surf life savers' managerial hub upon detecting any situations that must be addressed (e.g., overcrowding and confirmed emergencies); such inferred context is known as high-level context.

Different manufacturers build context acquisition devices, and various cloud and network service providers process their context and connect them. Furthermore, diverse context providers (e.g., Google and Apple) may collect and relay their context. Hence, the QoC associated with the context dramatically varies depending on the technical specification of such parties. The QoC can be measured using the current methods (e.g., proposed in [4,6]); however, these methods are heavily dependent on the context providers' inputs. For instance, the CMPs measure contexts' age (a form of timeliness) using the time-stamp, relaying the context generation time issued by the context provider. So, these parameters are vulnerable to manipulation, as the quality measurement outcomes are what the providers intend, potentially leading to ineffective rescue operations. Hence, in this paper, we propose a method to obtain veracity – relaying the trustworthiness of context providers to deliver the QoC parameters accurately – so that we can select the context providers relaying a valid context.

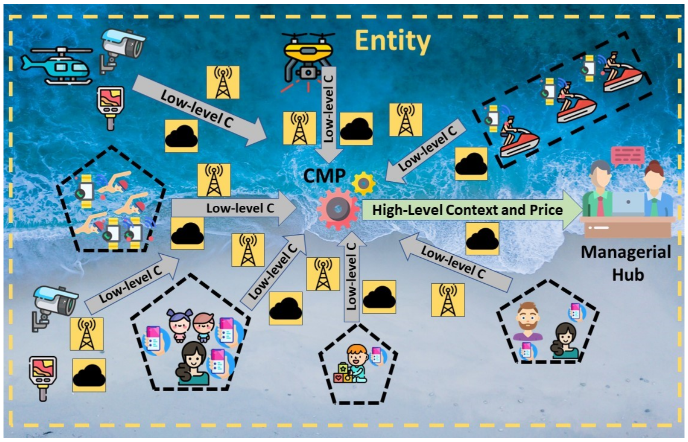

Fig. 1. The context-aware surf life saving use case. The context providers (cluster of mobile phones, smart watches and cameras) relay the low-level context to the CMP (represented using the gear icon), and CMP sends high-level context and price to the managerial hub. Each context provider is powered with cloud and network resources - depicted using the relevant icons.

III. Literature Review

According to features of advanced CMPs in the survey [11], most CMPs lack standard QoC-aware selection and QoC measurement functions. The works in [4,6,12] discussed the popular QoC measurement models; that rely on QoC parameters in context. Furthermore, works in [3,7,12,13] discuss QoC-aware selection models; that rely on the outcomes from the QoC measurement models (e.g., aggregation of context provider's QoC metrics) to prioritise the providers for selection. Therefore, incorporating the current QoC-aware selection models in CMPs can be less credible in determining the best-performing context provider (Sections 1 and 2 discuss the issues in current QoC measurement models). Motivated by such drawbacks of current QoC-aware selection and the infancy of QoC validation models, in this work, we propose to use QoS and QoD metrics of context providers to validate their QoC outcomes; that assist in QoC-aware selection. The QoS metrics of context providers can be computed by the CMP or obtained from the cloud and network service providers – without relying on context providers' inputs. In the survey in [14], we discussed the types, methods and examples to compute QoS metrics in CMPs. Further, QoD metrics are device features of the context providers; they can be determined based on the context provider's details (e.g., provided on their SLAs). For instance, the coverage of thermal imaging is one of its QoD features [15].

We surveyed the effect of QoS metrics related to context providers' processing (cloud resources handling their data processing) and network (network channel connecting them with the CMP) components on various QoC metrics [14]. Furthermore, the works in [4,7] indicate that similar context providers may produce varying accuracy depending on their sensing mechanisms, which are the QoD features. Hence, we assess the veracity based on the context providers' QoS and QoD outcomes affecting the required QoC metrics.

Multi-criteria decision-making (or MCDM) [9] is a widely employed approach to determine the performance of items - context providers in our case - based on various metrics' (QoS and QoD metrics) and their effect on required criteria (QoC requirements). Therefore, we use MCDM as the base method for our design to determine the context providers' performance. Section 5 discusses the details.

IV. conCQeng 2.0 – Design and Process Flow

Fig. 2 depicts a high-level architecture and process flow among the components in conCQeng 2.0. It is the advanced version of conCQeng [7], designed to deliver a QoC-aligned and cost-effective context to the CMPs through the QoC, cost-aware selection, and their respective measurement. Considering conCQeng 2.0 is an improved version, it is architected by modifications to its previous version.

ConCQeng 2.0 adopts the following sub-components from its previous version: RRP (known as Relative reputation processor), AP (Assurance processor), QMU (QoC measurement unit), and CoCM (cost of context measurement unit). The RRP performs the QoC-aware selection by assessing and maintaining the context providers' performance in delivering adequate QoC. The AP invokes (or recommends the CMP with) the most cost-efficient context provider from the short-listed ones from RRP. AP uses context providers' given costs and penalties for QoC inadequacies (obtained from their service level agreements (SLAs)) to determine cost efficiencies. Further, it also caches those context providers from RRP; so that it can invoke a next cost-efficient one in case of inadequate QoC from the formerly invoked provider, thereby improving chances of completing the context request. The QMU measures the QoC metrics in the context and checks its compliance with the context request, prompting only a worthy context further. Otherwise, it notifies the AP to invoke the next provider. Finally, CoCM measures the final cost of context - by applying penalties concerning QoC inadequacies - and promotes the context to CMP.

We have improved the conCQeng's functionalities in this work by adding VMU (veracity measurement unit) – a component to measure context providers' veracity. VMU assesses the context provider's performance in QoS and QoD metrics that directly affect its QoC metrics; this veracity further assists RRP with the QoC-aware selection. Hence, it improves RRP's credibility in QoC-aware selection, as RRP no longer only relies on outcomes of QMU, which uses traditional QoC measurement models.

The process flow in conCQeng 2.0, starting with QoC and CoC-aware selection to QoC and CoC measurement and validation, occurs as follows. In step 1, the RRP receives the context request(s) (CRs) issued by the context consumers from the CMP. It then performs the QoC-aware selection and finds the context providers (CPs) delivering the adequate QoC for each context request. The RRP relies on two metrics to find such providers: QoC adequacy rate – the rate of QoC adequacy from a context provider to complete the CR, and

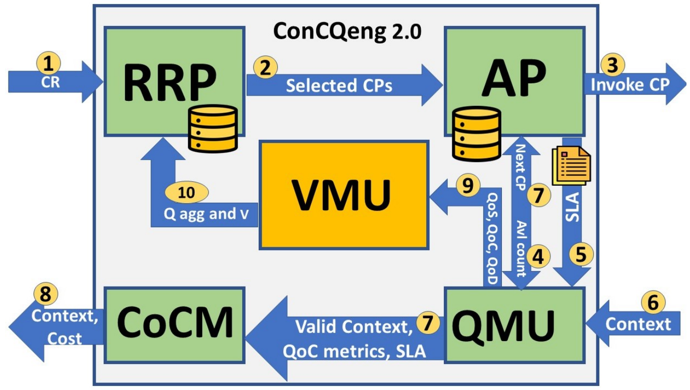

Fig. 2. The conCQeng 2.0 - the architecture and process flow.

veracity - guarantees the legitimacy of the provider's QoC-delivery rate based on their QoS and QoD metrics. In step 2, the AP receives the selected CPs. It then sorts them in the order of their CoC-efficiencies – from top to bottom; then stores them in a cache and advises (by supplying their id) the CMP to invoke the top provider in step 3. Finally, it transfers the number of cached providers and the invoked provider's SLA (it contains the details of context, QoC, cost and penalties for QoC inadequacies given by the provider) to the QMU in steps 4 and 5.

In step 6, the QMU receives the context from the invoked provider; it measures the QoC and QoS metrics and determines QoD features using the provider's SLA. If the QoC metrics comply with the CR's requirements, it promotes the context for further processing. Otherwise, it advises AP to invoke the next available provider (by checking the availability of the providers based on the information received from AP in step 4). Finally, in step 7, the CoCM computes the context cost – penalizing the provider for degraded QoC based on the penalties given on its SLA. It then promotes the context and final cost details to the CMP in step 8.

Furthermore, in step 9, the QMU transfers the QoC, QoS and QoD metrics to VMU. Using QoS and QoD metrics, VMU computes the provider's veracity. It then promotes the QoC metrics' aggregate and veracity unit to RRP in step 10. Finally, the RRP computes the provider's QoC-performance rate and veracity using these inputs. Our previous work in [7] discusses our method to compute the QoC metrics. Furthermore, computing QoS metrics is out of this paper's scope; our survey in [14] discusses such methods.

V. Computing and Using Veracity

Each component in conCQeng 2.0 incorporates a unique method to perform its designated action. For example, RRP uses a novel QoC-aware selection model called the relative reputation; to select the context providers based on their reputation (assessed as the QoC adequacy rate) to fulfil each unique type of context request. AP uses Assurance to determine the provider's cost-effectiveness based on having a low cost and high penalties for QoC inadequacies as the criteria.

Furthermore, QMU incorporates a QoC measurement model to compute the QoC metrics using the supplied QoC parameters in the context. Using these metrics, the RRP computes the provider's QoC adequacy rate. Finally, CoCM computes the cost of context based on obtained QoC metrics - by applying penalties for any inadequacies. Therefore, the extension – VMU - supplements the RRP with additional metrics for QoC-aware selection and QoC validation to overcome integrity issues using the outcomes from traditional QoC measurement models by QMU.

Our previous work in [7] discusses the details of relative reputation, assurance, QoC and CoC measurement mechanisms, and it also provides an SLA template used by the context providers to negotiate QoC and cost guarantees with the CMP. This section discusses the method for veracity measurement in VMU and its role in QoC-aware selection. We measure veracity based on the context provider's QoS and QoD metrics - that directly affect the QoC metrics - compared to those of other context providers (those invoked earlier and exact time) addressing a similar type of context request in a particular situation (e.g., location, time). For example, video cameras and GPS systems on the shark and swimmer could detect a possible shark attack. Therefore, invoking a video camera results in measuring its veracity with GPS systems. The VMU measures the veracity of each provider using the algorithm below, inspired by multi-criteria decision-making; it considers the QoS and QoD metrics that affect QoC and the significance of the QoC metrics to the context consumer to assess veracity.

Algorithm 1 An algorithm to compute the veracity

Inputs(ConCQeng 2.0 - QMU): QoC$_i, QoS_i, QoD_i$
Inputs(CMP): Ri, QW$_i, SW_i$
Where i = 1....n, i.e.,number of metrics;
For each QoS$_i$
while $i \leqq$n **do**
  if QoS_i needs to be higher **then**
    $RM_i = \frac{QoS_i}{max(QoS_i)}$...........(1)
  else if QoS_i needs to be lower **then**
    $RM_i = \frac{min(QoS_i)}{QoS_i}$...........(2)
find $(Ri\ value\ for\ QoD_s)$
    SM$_i = \frac{Ri}{max(Ri)}$...........(3)
    while $i \leqq$n **do**
      CS$_i = QW_i \times RM_i$...........(4)
      CD$_i = SW_i \times SM_i$...........(5)
Veracity Unit = $\sum_{i=1}^{n}(CS_i + CD_i)$...........(6)

The VMU uses the inputs from QMU and the CMP (from the context request) to measure the veracity unit for each context response from a provider; the aggregate of such veracity units forms the veracity of a provider. The QoC_i and QoS_i represent the context provider's i^{th} QoC and QoS metrics. QoD_i represents the context provider's sensing feature (e.g., coverage of thermal camera). The i ranges from 1 to n (the number of metrics) for these quality dimensions. Furthermore, Ri represents the relevance indexes, a suitability level of each sensing feature (QoD_i) to address the context consumer's requirements. For example, an image-processing video camera possesses the Ri of 8 or 9 (Ri ranges from 1-9), indicating that the sensing feature is highly suitable for confirming emergencies in a particular location – as it offers a high resolution. Furthermore, QW_i, and SW_i represent the weights (the significance -level) of the QoS_i and QoD_i metrics to be valid. CMP determines this weight through Analytical Hierarchy Process (AHP) [10], using context consumer-defined input representing the importance of each QoC_i in the range of 1 to 9. CMP uses such input to determine the QW_i and SW_i of each QoS and QoD metric that affects the QoC_i. For example, CMP determines the QW_i of response time based on the importance of timeliness; SW_i of coverage based on the importance of accuracy.

The VMU normalise the QoS and QoD metrics using equations (1), (2) and (3). Equation (1) and (2) obtains RM_i, a normalised value of QoS_i; (1) normalise the metrics requiring to be higher for an adequate performance (e.g., availability) by comparing it with the maximum value from other providers; (2) normalise the metrics requiring to be lower for an adequate performance (e.g., delay). Using (3) normalises the QoD metrics, obtaining SM_i by comparing provider's Ri with the Ri of the most suitable QoD feature in the existing providers. Finally, equations (4) and (5) obtain the CS_i and CD_i, providing the alignment of the QoS and QoD outcomes to their importance to the context consumer. Finally, we determine the veracity unit using (6).

The conCQeng 2.0's RRP maintains each context provider's veracity units' average to form veracity. Therefore, it selects the context provider if this veracity exceeds the average veracity of all providers suitable to complete the context request while the veracity also exceeds the threshold value. The RRP obtains such threshold values from the QMU, computed by having all the relevant QoS and QoD metrics to a bare minimum to meet QoC compliance.

VI. Implementation and Evaluation

The setup to evaluate conCQeng 2.0 contains the four following components. (i) conCQeng 2.0, an instance running on Google App Engine. (ii) CoaaS - a CMP [16]. (iii) Data servers that collect real-world crowd density data from the APIs, then convert it to resemble context and deliver it to conCQeng 2.0. (iv) A Web application, developed using React.JS, to produce context requests and SLAs of context providers to conCQeng 2.0; it also visualises conCQeng 2.0's outcomes (context, QoC, QoS and Cost) for evaluation. Fig. 3 depicts the outcome visualisation page from the application, representing the top providers to address the context request, proportions of QoC metrics in the overall QoC achieved, and penalties applied for QoC metric inadequacies. The application also allows users with advanced features such as defining the context requests for their use cases, saving and viewing the QoC, QoS and final cost (the cost upon applying penalties for the QoC inadequacies); due to space constraints, we could not depict more features form this outcome visualisation page.

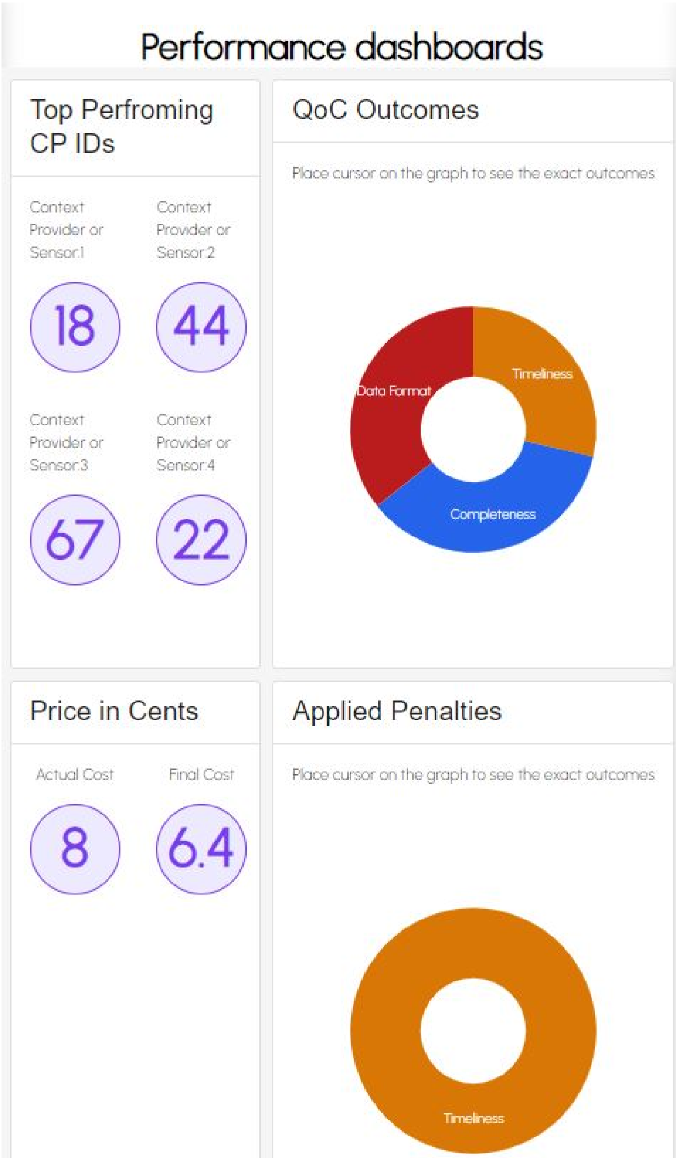

Fig. 3. Web application's page that visualises the outcomes from conCQeng 2.0, including the top performing providers, QoC outcomes, price of context - before and after applying penalties, and the QoC metrics for which the penalties are applied.

We have used the crowd density data related to a busy street in Melbourne, collected from the Melbourne city pedestrian counting system [17], to evaluate conCQeng 2.0. The data resides on an individual CSV file for each server; each data set includes the crowd density identified at each hour for a week. Upon invocation, the data server obtains a crowd density value from the CSV file - providing a different value per invocation; it then converts the data values to context - by adding QoC parameters (e.g., context generation time-stamps) and delivers it to the conCQeng 2.0.

The evaluation consists of two objectives. The first objective is to assess the proposed method's credibility to determine the invalid QoC metrics (occurs due to misrepresented QoC parameters) in different circumstances: increasing and periodic errors. These are situations with increasing and periodic mismatches of QoS (e.g., delays and response times) – while QoC parameters align with the computed QoC - potentially leading to an invalid QoC measurement. The second one is to determine the performance efficiency of conCQeng 2.0.

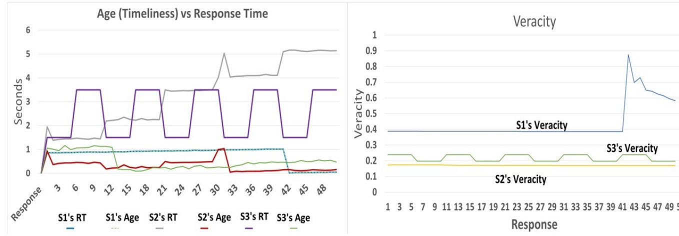

Fig. 4. The graphs on the left and right-hand sides depict the age and response times (RT) exhibited by three servers (S1, S2 and S3) and their veracities assessed by conCQeng 2.0.

We attained the first objective: We used three data servers (S1, S2 and S3), with S1 delivering correct context generation time-stamps while the rest delivered the incorrect ones. However, servers S2 and S3 exhibit increasing and periodically varying response times. All three servers' ages (a QoC metric for timeliness) and response times are visualised in the left-hand side graph of Fig. 4. ConCQeng 2.0 computes the age as the difference between the current and context generation time-stamps. It determines the response time based on the time between the data servers receiving the request to getting back the context response. In a real-world deployment, conCQeng 2.0 obtains such information from the network service providers (by making contracts to disclose the message information).

We aimed to prove conCQeng's 2.0 credibility based on the veracity it assigns to these servers. Therefore, we invoked these servers repeatedly (50 times) for the context request related to "crowd density" with 1.1 seconds as the required age, i.e., the server must generate and deliver the context within 1.1 seconds after receiving the request. As a result, the S1's response time closely matches the context age (metric for timeliness), indicating that the provided context generation time-stamp is correctly represented. On the other hand, S2 and S3's response times are more excessive than their ages, indicating misrepresented context generation time-stamps.

The graph on the right-hand side of Fig. 4 indicates the veracity assigned by conCQeng 2.0 to the servers. The results are obtained by having Ri (the QoD metric's compatibility to produce the correct context) of 0.7 for S1, and 0.3 for S2 and S3; same weights (QW_i, and SW_i) for QoS and QoD metrics. The results depict that conCQeng maintained higher veracity for S1. Therefore, it indicates that relying on conCQeng 2.0 aids in selecting a credible context provider in QoC-aware selection. Besides, the veracity alignment of S3 with its response time indicates conCQeng 2.0's accuracy in assessing the veracity.

Next, we evaluated the conCQeng's 2.0 performance effi-

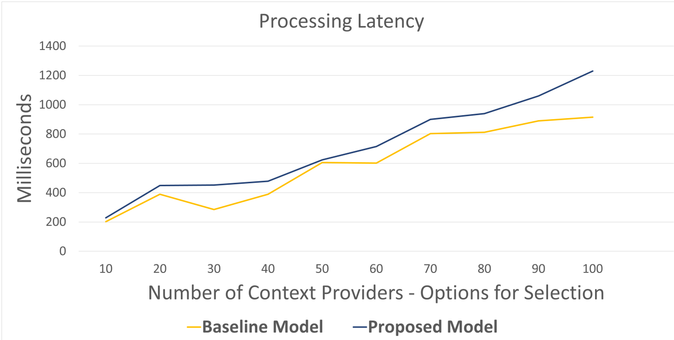

Fig. 5. The process latency exhibited by the baseline (conCQeng) and the proposed (conCQeng 2.0) models.

ciency by increasing the selection overhead, i.e., increasing the number of potential options in context providers for the selection. Considering the overheads (e.g., cost) of the numerous servers in the cloud, we have performed this experiment in a local environment, in a system with 16 GB RAM and 16 core i7 CPU. Using the web application, we simulated the context providers by defining their SLAs in conCQeng 2.0 and its previous version in [7] ("baseline model"). We aim to find both systems' performance efficiency for the increased selection options - from 10 to 100 providers to complete the context request used to evaluate the first objective. The conCQeng 2.0 relies on two metrics between the providers: veracity and relative reputation, unlike the baseline version that uses only the RR. Hence, as depicted in Fig. 5, the processing latency of the proposed model (conCQeng 2.0) is slightly higher than the baseline model; it occurs due to the additional processing components (VMU). However, the proposed model rules out the credibility issues from the context providers related to QoC, thereby addressing a significant challenge. Furthermore, the processing latency in this model can be reduced by employing more cloud resources, which will be a part of our future work.

VII. Conclusion

Satisfying QoC requirements is essential for any CMP to deliver high usability for its consumers (e.g., pervasive applications). However, incorporating the current QoC-aware context provider selection models and QoC measurement models may select less credible providers - those relaying false QoC metrics. Therefore, this paper proposed a component named conCQeng 2.0 that performs QoC validation along with QoC-aware selection and QoC-measurement. This component incorporates a novel QoC validation method that forms veracity on context providers in delivering correct parameters in context for QoC metric assessment. Hence, relying on the veracity and the QoC delivery rate of context providers (obtained from QoC measurement) leads to selecting credible context providers delivering an adequate QoC.

We have motivated the conCQeng 2.0 to introduce context-aware solutions to surveillance and detect emergencies in the surf life saving. Furthermore, our evaluation proves that conCQeng 2.0 obtains valid QoC outcomes for the CMP. Our future work includes improving the performance efficiency of conCQeng 2.0 through effective resource allocation and expanding it to other popular CMPs such as FIWARE [18].

References

[1] A. K. Dey, "Understanding and Using Context." Personal and ubiquitous computing, vol. 5, pp. 4-7, Feb 2001.
[2] "Google Maps Platform - Location and Mapping Solutions." https://mapsplatform.google.com/ (accessed: Nov. 07, 2022).
[3] K.-M. Kouamé, H. Mcheick, and H. Ajami, "Adaptive Mechanism Model for the Prevention of SLA Violation in the Context of COPD Patient Monitoring." Symmetry, vol. 12, no. 9, p. 1575, 2020, doi: 10.3390/sym12091575.
[4] A. Manzoor, H. L. Truong, and S. Dustdar, "Quality of context: Models and applications for context-aware systems in pervasive environments," Knowl. Eng. Rev., vol. 29, no. 2, pp. 154–170, 2014, doi: 10.1017/S0269888914000034.
[5] T. Buchholz, A. Küpper, and M. Schiffers, "Quality of Context: What it is and why we need it," Proc. 10th Work. OpenView Univ. Assoc. OVUA'03, pp. 1–14, 2003, doi: 10.1.1.147.565.
[6] J. B. Filho and N. Agoulmine, "A Quality-Aware Approach for Resolving Context Conflicts in Context-Aware Systems," 2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing, 2011, pp. 229-236, doi: 10.1109/EUC.2011.9.
[7] K. S. Jagarlamudi, A. Zaslavsky, S. W. Loke, A. Hassani, and A. Medvedev, "Quality and Cost Aware Service Selection in IoT-Context Management Platforms," 2021 IEEE International Conferences on Internet of Things (iThings), Dec. 2021, pp. 89-98, doi: 10.1109/ithings-greencom-cpscom-smartdata-cybermatics53846.2021.00028.
[8] A. Walters, "Conducting Information Superiority Research in an Unclassified Surrogate Defence Environment ." Defence Science and Technology Group, Sep. 2018. [Online]. Available: https://www.dst.defence.gov.au/publication/conducting-information-superiority-research-unclassified-surrogate-defence-environment
[9] A. Ishizaka and P. Nemery, Multi-criteria decision analysis : methods and software. Chichester, West Sussex, United Kingdom: Wiley, 2013.
[10] T. Saaty, "A scaling method for priorities in hierarchical structures", Journal of Mathematical Psychology, vol. 15, no. 3, pp. 234-281, 1977.
[11] X. Li, M. Eckert, J. F. Martinez, and G. Rubio, "Context aware middleware architectures: Survey and challenges," Sensors (Switzerland), vol. 15, no. 8, pp. 20570–20607, 2015, doi: 10.3390/s150820570.
[12] P. Marie, T. Desprats, S. Chabridon, M. Sibilla, and C. Taconet, "From Ambient Sensing to IoT-based Context Computing: An Open Framework for End to End QoC Management," Sensors, vol. 15, no. 6, pp. 14180–14206, Jun. 2015, doi: 10.3390/s150614180.
[13] E. Eryilmaz, F. Trollmann, and S. Albayrak, "Quality-aware Service Selection Approach for Adaptive Context Recognition in IoT," Proceedings of the 9th International Conference on the Internet of Things, Oct. 2019, doi: 10.1145/3365871.3365874.
[14] K. S. Jagarlamudi, A. Zaslavsky, S. W. Loke, A. Hassani, and A. Medvedev, "Requirements, Limitations and Recommendations for Enabling End-to-End Quality of Context-Awareness in IoT Middleware," Sensors, vol. 22, no. 4, p. 1632, Feb. 2022, doi: 10.3390/s22041632.
[15] M. Kristoffersen, J. Dueholm, R. Gade, and T. Moeslund, "Pedestrian Counting with Occlusion Handling Using Stereo Thermal Cameras," Sensors, vol. 16, no. 1, p. 62, Jan. 2016, doi: 10.3390/s16010062.
[16] A. Hassani et al., "Context-as-a-Service Platform: Exchange and Share Context in an IoT Ecosystem," 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Mar. 2018, doi: 10.1109/percomw.2018.8480240.
[17] "City of Melbourne - Pedestrian Counting System," www.pedestrian.melbourne.vic.gov.au. http://www.pedestrian.melbourne.vic.gov.au/ (accessed Nov. 28, 2022).
[18] "The Open Source platform for our smart digital future," FIWARE. https://www.fiware.org/ (accessed Nov. 28, 2022).