QoX: Quality of Service *and* Consumption in the Cloud

Murad Kablan, Eric Keller
University of Colorado, Boulder

Hani Jamjoom
IBM Research

Abstract

Cloud services today are increasingly built using functionality from other running services. In this paper, we question whether legacy Quality of Services (QoS) metrics and enforcement techniques are sufficient as they are producer centric. We argue that, similar to customer rating systems found in banking systems and many sharing economy apps (e.g., Uber and Airbnb), *Quality of Consumption (QoC)* should be introduced to capture different metrics about service consumers. We show how the combination of QoS and QoC, dubbed QoX, can be used by consumers and providers to improve the security and management of their infrastructure. In addition, we demonstrate how sharing information among other consumers and providers increase the value of QoX. To address the main challenge with sharing information, namely sybil attacks and mis-information, we describe how we can leverage cloud providers as vouching authorities to ensure the integrity of information. We explore the motivations, challenges, and potentials to introduce such a framework in the cloud environment.

1 Introduction

In this paper, we revisit quality of service (QoS) abstractions and enforcement techniques in the growing ecosystem of interdependent services within the cloud. Specifically, we observe that while QoS is the subject of much research, it has been primarily provider centric. For example, QoS for video streaming looks at the quality of provider's video and how it is impacted by bandwidth, latency, jitter, etc. This unidirectional view has dominated the design of past QoS abstractions and enforcement techniques.

Thinking of the problem space from a cloud service ecosystem perspective creates new opportunities for improving both service production and consumption. In this new environment, better service production is encouraged. Similarly, better service consumption is rewarded. Finally, poorly implemented services and malicious clients are isolated. Achieving this vision requires questioning the unidirectionality of QoS designs. In particular, we introduce *Quality of Consumption (QoC)* to refer to metrics that can be captured to define attributes of how a consumer is using a service.

The idea of QoC is not new in the real world. Lending (e.g, mortgage and credit cards) are dependent on the ratings of the customers. Customers (credit consumers) who demonstrate consistent repayment of loans have improved FIMCO scores – the lender benefits in having greater confidence in lending money to those with higher credit scores, and the borrower (with good credit scores) benefit in higher future credit limits. In the cloud, consider one service consumer that always keeps its 3rd party software up to date, and has a rigorous testing framework before deploying new functionality. In contrast, a second hasn't updated its 3rd party software in over a year, and has, on multiple occasions in the past, deployed software that has submitted malformed requests to the service's API that led to monitoring alarms to go off. The first will cause less problems for the service provider, and therefore require less resources, and in turn, should receive a benefit. We are proposing that service providers can differentiate based on a metric that captures how a customer is consuming a service.

In most real systems, both QoS and QoC are needed. Recent startups in the *sharing economy* space, like Uber and Airbib, demonstrate how quality of the providers and consumers can be leveraged to enable trust between all parties. We use the term QoX to capture systems that integrate both QoS and QoC. Conceptually, there are then three import components to QoX environments: (C1) providers' QoS are monitored and rated, (C2) consumers' QoC are similarly tracked, and (C3) this information is shared among providers and consumers and is used in provider selection and service differentiation.

In this paper, we show how a similar setup can be

Dimension	Metrics
Performance	Throughput, packet loss probability, response time, jitter
Dependability	Reliability (*e.g.,* maximum number of crashes or interruption), availability (*e.g.,* maximum number the service will be unavailable)
Cost	Prices and rates

Table 1: QoS metrics of Cloud services

Dimension	Metrics
Customer purchase power	Length existing as a user, frequency of orders, amount of purchases
Customer's code efficiency	Version of software customer is running, malformed requests (*e.g.,* Web server error logs)
Customer threat	IDS alerts. Service crash reports

Table 2: QoC metrics of Cloud services

achieved in cloud environments. Specifically, we look at how to extend existing QoS frameworks to support QoC. This is to support C1 and C2 above. Furthermore, we also look at how to use cloud providers as vouching authorities to achieve C3, even in the presence of sybils and liers. We present initial thoughts on the appropriate abstractions and interfaces to address them on a cloud based framework that manages and define the quality of interaction and service from both consumer and provider's perspectives. We explore the motivations, challenges, and potentials to introduce such a framework in the cloud environment.

2 Defining Quality

In this section, we introduce the term Quality of Consumption (QoC) as a counterpart to the Quality of Service (QoS) metrics. Both can be readily measured (latency, bandwidth, etc.) and can be specified as part of SLAs. We define QoS and QoC. We also describe the needed components to enable QoX in cloud environments.

Quality of Service (QoS). QoS can be defined as a measurable level of service delivered to its users'satisfaction. QoS of cloud services can be characterized across multiple dimensions, each having a set of metrics (Table 1). For example, the quality of a database service, such as ClearDB[2], can be measured by its dependability (availability) and performance (response time for SQL request). While the quality of an ad service such as Adobe Ad[1] can be measured by its rate.

Quality of Consumption (QoC). QoC captures how well users are consuming a service. It can be used by service providers to assist in admission control decisions or when providing service differentiation. QoC is a way to recognize that service consumers are not equal. To some extent, QoC monitoring already exists (e.g., intrusion detection and prevention systems). These are point solutions. The problem is that there is no abstraction or framework for cloud service providers to collect various QoC metrics and then translate them to suitable actions. Similar to QoS, QoC can be characterized across multiple dimensions, each having a set of metrics (Table 2).

QoX = QoS + QoC. We use the abbreviation QoX to capture the combination of QoS and QoC. As illustrated in Figure 1, through measurement systems and other system logs, consumers and providers capture quality metrics about each other (*e.g.,* via an IDS or other resource monitors), captured in the box labeled information about service provider(s) or consumer(s) – indicating it is available, not necessarily stored. Collectively, this information is interpreted as QoX, and can be used, either directly or indirectly, in managing the infrastructures operation, both at run-time and during initialization. We discuss the implementation and integration of initial proof-of-concept prototype in Section 5.

3 Democratization and Sharing of QoX

In sharing information, consumers and providers can gain the benefit of others' experiences. This can be useful, for example, when choosing a service provider, or knowing that certain consumers are likely to pose security threats. Even more, this information can be used for self-feedback akin to sentiment analysis widely used in corporations (*e.g.,* monitor Twitter feeds to observe whether there is any positive or negative chatter affecting its brand [10, 7]).

Illustrated in Figure 2 is an ecosystem of consumers and providers, all interacting with one another (forming a system of engagement [13]), and exchanging the information. The information about a service (or consumer), previously illustrated in Figure 1, is shared with a logical service labeled information exchange. In the remainder of this section, we discuss the two main types of information. The first summarizes information about the interaction as a whole; the second is a record of a specific interaction. We address the challenges of dealing with lying and sybils in Section 4.

3.1 Summary of Engagement

The challenge in simply exchanging the information about a provider (or consumer) is there is no clear, standardized way to compare quantifiable metrics across

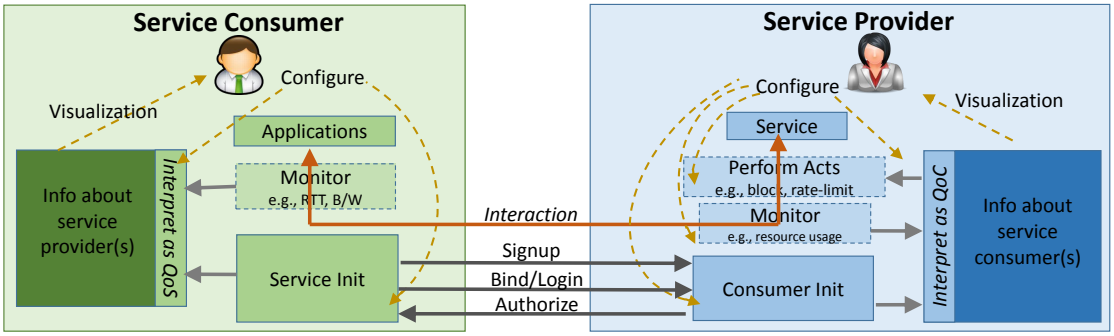

Figure 1: Interpretation of measured information as quality of service or quality of consumption.

providers and consumers. Even simple and as well-defined metrics such as latency can be subjective (*e.g.,* due to network proximity). Instead, we are inspired by review systems found in web sites such as Yelp (for restaurants) and Amazon (for products). In this case, consumers and providers share a scalar, subjective rating of quality coupled with a (machine generated) text based review.

3.1.1 Scalar, Subjective Rating

The summarization rating is scalar (as opposed to just good or bad) as there are many factors that go into overall quality, and subjective to account for the wide variety of metrics and needs of various providers or consumers. This rating represents the current view, with the weighting of history versus recent experiences left to the rater. A key challenge is determining the rating. As illustrated in Figure 2, we envision that each consumer and provider will have software which interprets the information about a consumer or provider to ultimately determine the rating . The administrator configures how this interpreter behaves. This is a long term challenge—creating a language to enable administrators on either side to integrate with the measurement systems, being able to specify expectations, and how each component impacts the overall rating. For initial exploration, we can provide thresholds for specific metrics, or simply let the human administrators provide a rating.

Having the shared information can strengthen the confidence in those actions, but also opens some additional avenues for the specific case of a given consumer and provider having not had interaction before. On the consumer side, the shared information can support the consumer in making a decision about which service provider to use. On the service provider side, the shared information can support the provider in offering incentives to use a service or introducing restrictions in use.

3.1.2 Text Review to Help Interpret the Rating

A coarse-grained rating has the benefit that it can be used in any context. The downside, however, is that it hides potentially useful information (*e.g.,* information about why a particular reviewer gave the rating they did) – making it a challenge to interpret the rating.

One approach to getting around this is to use sub-categories, though this has notable downside of determining the sub-categories requires forecasting every single criteria that might be used—an impossible task. In practice, a small set of sub-categories is useful (*e.g.,* Home Depot has quality and value in addition to overall rating, to help separate all metrics related to the product and metrics related to the cost), but having too many will inevitably require standardization.

Instead, we propose including machine-generated text based reviews to go along with the rating. This will allow each reviewer the means to specify why they gave the rating, and each user of a review to find whether the rating reflects its needs. Clearly, text-based reviews are helpful for human adminstrators if they want to view the ratings, but we believe text reviews can also help guide automated systems as well.

Through systems (such as Elastic Search) which extract structure from unstructured data and provide analysis, we believe that it will be possible to extract the commonalities among reviews – that is, automatically creating sub-categories that are relevant for that particular provider or consumer. Amazon does this for product reviews as they highlight common comments.

Creating these machine-generated text reviews is not a significant challenge as they can be built out of specific log text, and based on the administrators configuration of the QoX interpreter. Configuring the infrastructure to search for specific information will require some administrative effort initially, to understand, generally, what the commonalities are and then building that into the configuration for how to use the ratings as they are adjusted over time.

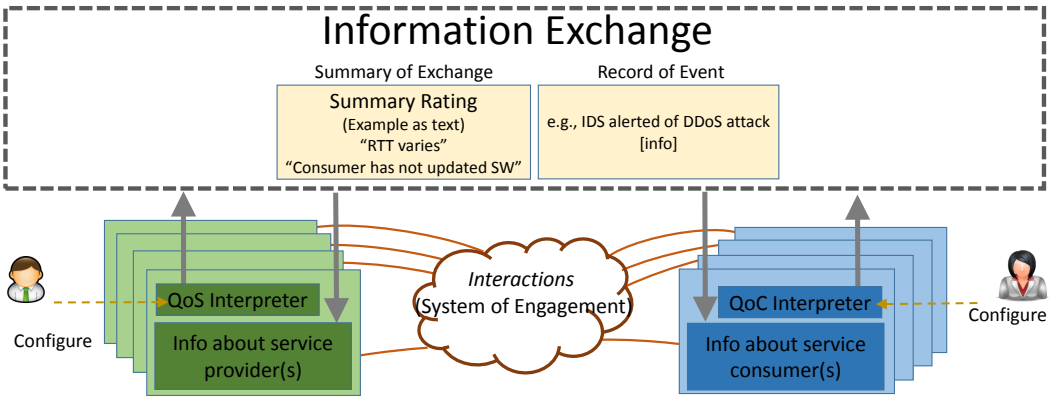

Figure 2: Illustration of information exchange

3.1.3 Personalization without Connections

In other information exchange systems, there have been proposals to leverage social structure to provide more relevant information. We believe the same is true for cloud services, except, of course, there is no notion of friends among service providers or consumers. Previous attempts [9] have extracted such a 'social' structure by way of interactions. This, however, requires a fairly connected graph. Whereas, the consumer-provider graph is likely to be mostly a bi-partite graph.

Instead, we can perform personalization by leveraging techniques used by recommendation engines. In recommendation systems, the goal is to predict a choice (*e.g.,* what movie to watch) based on others with like characteristics (*e.g.,* who have watched similar movies). That is, they find connections between unconnected users using machine learning techniques (*e.g.,* PredictionIO [5]).

Our goal is not to make a recommendation, but to highlight reviews that are particularly relevant. The challenge is identifying relevant features. This may include examining consumers who use a similar set of services, or who uses a similar set of API calls for a given service. This is an area for future research.

3.2 Exchanging Specific Records

Ultimately, exchanging summary of exchange information will capture a scalar rating of different metrics about consumers and providers. While ratings, even detailed ratings, have value in making coarse decisions, some information that an individual consumer or provider records would help others if shared. This is specifically evident in securing an infrastructure. At a high level, consider the possibilities if each service provider share their intrusion detection system (IDS) logs and alerts. Now, all providers could get the benefit of a 'global IDS', allowing them to protect themselves even before seeing an attack and even if they didn't have their own ability to detect a given attack.

As a specific example, in 2014, hackers exploited a bug in the Amazon EC2 API to gain access to other tenants' accounts and then flood other servers with UDP packets to cause a denial of service [6, 11]. Providers sharing alerts of the attack, can greatly reduce the damage to services by warning others of potential threats, allowing them to, for example, block that particular tenant. Note that in this example, while it may not prevent the attack from successfully denying service for the first tenant attacked, it generally helped others (making it valuable), and there are a larger set of situations where everybody benefits – specifically in detecting, and comparing, reconnaissance efforts of potential attackers.

4 Cloud Provider as Vouching Authority

A key challenge arises when dealing with any sharing is ensuring the validity of information. As we move toward an entirely cloud based infrastructure—both IaaS and PaaS—we believe that an opportunity to overcome these challenges becomes possible, where the cloud provider serves as a vouching authority. We elaborate on two ways by which validity of information can be compromised, and the role of the cloud provider in each.

4.1 Fake Identities

In Section 3, we proposed that each consumer or provider can rate the other party for which they have had an interaction with. In an ideal world, quality is computed by equally weighting everyone's opinion (e.g., via average rating). In reality, reviewers can create fake identities, also know as the *sybil attack* [8]. For example, if a service provider were able to create a substantial number of fake entities that then rate its service as high quality, that service provider's rating will be unnaturally high.

The inverse is possible, where a competitor that wants to negatively influence another provider's rating. There are a number of prevention techniques [12, 15] that have been designed for decentralized (peer-to-peer) systems. These will not work here as each relies on a trust graph (*e.g.,* a social network graph) in determining the likelihood that an entity is real and limiting the influence the collection of sybils can have. Such graph does not exist here.

Fortunately, these are not necessary for cloud based interactions as we have a central authority: the cloud provider. The cloud provider can vouch for the identity is real and unique. It is much more challenging to create fake accounts where identity verification is required, such as requiring a credit card (as Amazon does). While it may be possible for a small-scale attack by creating a few accounts (with a few credit cards), overall this prevention mechanism is sufficient if the cloud provider can vouch [15]

4.2 Providing Mis-information

While having the cloud provider vouch for an identity prevents sybil attacks, real parties can provide false information. Here, we propose that the cloud provider can, to some degree, vouch for the validity of information based on the visibility the cloud provider has.

Did the reviewer actually interact with the reviewee? Generally, we can rely on the crowd to deal with any lying, as a single rating will be noise in the overall rating. For example, if everyone is giving a service or consumer 5 stars except for its main rival, then not only will that rating have little ability to influence the overall score, but it might stand out and hurt the party giving that low rating. However, it is still desirable to restrict the ability to rate parties that one has not interacted with, to prevent collusion or compromises in some accounts. An IaaS provider has visibility into the network infrastructure, so can, for example, see whether the tenants exchanged at least a certain number of packets. An PaaS provider, such as running CloudFoundry [3], brokers the interconnection between service provider and consumer, so has the ability to indicate if a connection was actually made.

Can the information be trusted enough to act on immediately? Some information would cause immediate, automated action and as such lying can have a negative impact. Consider one service provider detecting the start of some attack (*e.g.,* a DoS attack), sharing this information with another provider will help them, say block that user. If they are able to lie about it, they can cause another service provider to block a consumer unnecessarily. So, the receiver of the information needs to ensure the information is accurate.

Again, the visibility the cloud provider has can be leveraged to validate certain information. Of course, this is a greater challenge to deal with than simply determining if two parties interacted. The challenge lies in the variety of information that can be shared. Each will have different characteristics which will serve as evidence (*e.g.,* if one tenant wishes to share that another tenant performed a port scan, then the cloud provider needs enough evidence to verify that occurred). For this, we envision the tenant pre-specifying *evidence patterns*, which will specify the evidence that the tenant would like the cloud provider to collect. We envision measurable information such as bursts of traffic, crashes, specific IP address did indeed send something to another IP address, and not performing deep packet inspection, but this is an area for future investigation.

5 Conclusion and Future Work

We presented initial thoughts on including quality of consumption and quality of service in cloud-based services. We discussed the major challenges that arise when representing and controlling the interactions among service providers and consumers. With the proliferation of specialized services and a growing number of applications, we need to go beyond simply measuring and reacting, but to share the information with other consumers or providers. We also discussed how to leverage the (IaaS or PaaS) cloud provider as a vouching authority to deal with sybils and lying.

As future work, we plan to investigate and research a number of challenges: 1) integrating QoX framework with existing systems and applications to demonstrate the feasibility of such integration. As an initial proof-of-concept prototype, we prototyped the outbound logic for a QoX interpreter (as illustrated in Figure 2) using the Snort Intrusion Detection System [14] (configuring consumer ratings to go down for each monitored event seen) and the HAProxy load balancer [4] to use the rating to differentiate the consumer and direct requests of good consumers to a different set of servers than bad consumers. 2) a general specification language for the interpreter, 3) exploration into the cloud provider verifying records of events, 4) exploration of service or consumer features to enable automated personalization of results, 5) use of processing systems to extract structure from unstructured text reviews and automatically be able to use those reviews.

References

[1] Adobe Marketing Cloud. http://www.adobe.com/marketing-cloud.html.

[2] ClearDB. https://www.cleardb.com/.

[3] Cloud Foundry. http://docs.cloudfoundry.org.

[4] HAProxy. http://www.haproxy.org.

[5] PredictionIO. http://prediction.io.

[6] Amazon. Possible Insecure Elasticsearch Configuration. http://aws.amazon.com/security/security-bulletins/possible-insecure-elasticsearch-configuration/, May 2014.

[7] R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proc. AAAI Conference on Weblogs and Social Media*, 2010.

[8] J. R. Douceur. The sybil attack. In *The First International Workshop on Peer-to-Peer Systems*, IPTPS '01, London, UK, 2002.

[9] G. Frazier, Q. Duong, M. Wellman, and E. Petersen. Incentivizing responsible networking via introduction-based routing. *Trust and Trustworthy Computing*, 6740, 2011.

[10] D. Henschen. 10 tips: Tap consumer sentiment on social networks. http://www.informationweek.com/software/information-management/10-tips-tap-consumer-sentiment-on-social\-networks/d/d-id/1105234, July 2012.

[11] R. Millman. Hackers target Elasticsearch to set up DDoS botnet on AWS. http://www.cloudpro.co.uk/cloud-essentials/cloud-security/4353/hackers-target-elasticsearch-to-set-up\-ddos-botnet-on-aws, Aug. 2014.

[12] A. Mislove, A. Post, P. Druschel, and P. K. Gummadi. Ostra: Leveraging trust to thwart unwanted communication. In *Proc. of USENIX NSDI*, San Francisco, CA, Apr. 2008.

[13] G. Moore. Systems of Engagement and The Future of Enterprise IT: A Sea Change in Enterprise IT. http://www.aiim.org/futurehistory, 2010.

[14] M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In *Proc. of USENIX LISA*, Nov. 1999.

[15] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 267–278, New York, NY, USA, 2006. ACM.