Aggregated Capability Assessment (AgCA) for CAIQ enabled Cross-cloud Federation

Usama Ahmed, Imran Raza, Omer F. Rana and Syed Asad Hussain

[1]***Abstract*—Cross-Cloud Federation (CCF) enables resource exchange among multiple, heterogeneous Cloud Service Providers (CSPs) to support the composition of services (workflow) hosted by different providers. CCF participation can either be fixed, or the types of services that can be used are limited to reduce potential risk of service failure or secure access. Although many service selection approaches have been proposed in literature for cloud computing, their applicability to CCF (i.e. cloud-to-cloud interaction) has not been adequately investigated. A key component of this cloud-to-cloud paradigm involves assessing the combined capability of contributing participants within a federation and their connectivity. A novel Aggregated Capability Assessment (*AgCA*) approach based on using the Consensus Assessment Initiative Questionnaire from Cloud Security Alliance is proposed for CCF. The proposed mechanism is implemented as a component of a centralized broker to enhance the quality of the selection process for participants within a federation. Our experimental results show that *AgCA* is a useful tool for partner selection in a dynamic, heterogeneous and multilevel cloud federation.**

***Index Terms*—CAIQ, composite services, cross-cloud, federation, capability.**

I. Introduction

CLOUD federation also known as "cloud-of-clouds" provides composite services (realized as a workflow) that involve aggregation of capabilities provided by multiple Cloud Service Providers (CSPs) [1]. Federated services help a CSP to deal with unanticipated changes in resource behavior from one cloud provider, by acquiring the same resource from multiple potential CSPs (often dynamically, i.e. the interaction pattern between CSPs may not be known apriori). In federated clouds, a Service Level Agreement (SLA) is agreed between a user and a single CSP, which may involve use of a leased service(s) from various CSPs within the federation [2]. A permanent federation formed between CSPs having similar infrastructures, well acquainted due to continuous interactions with each other, offers limited benefits when application requirements/ demands change over time [3]. However, a cross-cloud federation [4] offers benefits in terms of scalability and the potential for diversity in service composition. Cloud providers contributing to CCF are not restricted by the resource limitations of their peer CSPs, but they can choose from a pool of resources shared by various peers. Regardless of these benefits, CSPs are reluctant to contribute to CCF, mostly due to the lack of confidence in each other [5].

Significant research exists focusing on using historical information to characterize cloud consumer and cloud provider relationship, such as a focus on conventional cloud computing [6-8], multi-clouds [9-11] and federated clouds [12, 13]. Limited coverage, however, exists for investigating relationships between clouds (i.e. cloud-to-cloud) [14]. Some authors have however identified the need to consider a variety of factors, such as social network ratings [14], behavior [15], pricing [16], etc to characterize this relationship.

Establishing an interaction between CSPs in a federation [17] requires an adaptive model to satisfy the inherent principles of federation of: *i) bi-directionality ii) relationship composition* and iii) *delegation control* [18]. Hence, the federated services require methods of representation, evaluation and dissemination of historical (interaction) information to reflect the hierarchical nature of the federation. Such evaluation must support methods to deal with cloud-to-cloud bi-directional relationship management and delegation control. Moreover, it must enable a cumulative score to be calculated, instead of specifying a score for individual CSPs [5, 18].

An Aggregated Capability Assessment (AgCA) metric is proposed for evaluating the cumulative capability score of a composite service within CCF. The proposed approach makes use of an audit based approach, extending existing efforts that make use of a policy or feedback based mechanism for capability assessment. Each CSP is audited for its security credentials through either a self-certification process or via a third party that has been approved by the Cloud Security Alliance (referred to as level II certification below). This assessment is only undertaken once, unless the services or the security capability of a CSP changes. Each CSP, therefore, has a single assessment, based on the assessment methodology that has been provided by Cloud Security Alliance. For this purpose, CAIQ assessment of CSPs having level-II certification from the CSA [19] has been utilized.

A CSP can either initiate or engage in a federation to offer services. CSPs may therefore have dependencies between them, which can be sequential (i.e. a one-to-one relationship between one CSP and another) or concurrent (i.e. one-to-many relationship, i.e. one CSP interactions concurrently with a number of others). The benefit of engaging in this type of relationship and is evaluated using the proposed *AgCA* approach. *AgCA* is implemented as part of a third-party broker that acts as a facilitator and a delegation controller for CSPs. Experimental evaluation is used to show the effectiveness of the proposed approach, i.e. determine the most effective CSP involved in delivering a service. The key contributions of this work are:

i. to establish a unique cross-cloud federation to deliver a specific case of the inter-cloud computing model and

identify a particular cloud-to-cloud relationship paradigm;

ii. to present a novel mechanism for evaluating accumulated capability value *AgCA* of composite services based on dependency relationships amongst peer CSPs offering these services. This is particularly relevant when a workflow is enacted across several different CSPs, based on specific (specialist) capability made available by each provider.

This paper is further divided into seven sections. Related work is presented in section II. Section III provides an overview of CCF and its uniqueness of relationship paradigm along with an overview of CAIQ from CSA. Section IV describes the research methodology. Section V presents the details of an individual capability assessment method for CSPs at the time they join the CCF. Section VI presents a detailed description of our proposed *AgCA* approach with section VII presenting experiment details and results. Section VIII concludes the discussion listing future research challenges.

II. Related Work

We review current trends in audit and assessment based approaches in conventional cloud computing and emerging trends in inter-cloud computing. A consolidated comparison of the proposed approach with various techniques has been presented in Table I.

A. Conventional cloud computing

The earliest work that refers to CAIQ for peer selection in cloud computing is a framework by Habib *et al.* [20]. The authors apply the notion of security attributes as defined by the Cloud Security Alliance (CSA) CAIQ framework. However, the article lacks a description of using it. In a later attempt [21], the authors elaborate by establishing an evaluation system based on CAIQ. Another framework presented in [22] uses the idea of Third Party Auditor (TPA) to support security auditing of a CSP according to security preferences requested by cloud users. However, the detailed functionality of such a mechanism is not discussed by the authors and they plan to develop multiple algorithms to support their work. In [23] authors have proposed a method of utilizing CAIQ complemented by feedback from users. However, their method of fusion lacks any kind of adaptability for dynamic cloud environments, i.e. when the properties of a CSP or their usage alongside other CSPs changes. In [24], the authors have proposed to utilize Cloud Trust Protocol (CTP) for users to request CAIQ assessment of CSPs in the form of opinions. These opinions are combined with the latest user feedback for the same service.

B. Inter-cloud computing

A reputation-based cooperation and resource sharing scheme for cloud providers is presented in [25]. The reputation score of a CSP is based on recommendations from peer cloud providers. However, this scheme does not address the concern of fake reputation scores from competitor CSPs. Another resource sharing scheme that utilizes Trust Service Providers (TSPs) has been presented in [11]. Their model evaluates the potential compliance of a CSP to its SLA. A framework, Service Operator-aware Trust Scheme (SOTS) [10] serves as a middleware for the discovery of resources in various clouds. SOTS utilizes information entropy theory for developing a broker-based adaptive evaluation approach. A similar work [9] proposes a service brokering scheme as a centralized broker (T-Broker) which utilizes an adaptive method for assessing the capability of a CSP. Their method complements the real-time service/resource monitoring by user feedback. A collusion resilient trust establishment framework is proposed in [14] along with a coalitional game theory based model. This framework enables different CSPs to create multi-cloud communities for resource collaboration. It is however unclear how the overall trust of the entire cluster is evaluated, managed and updated in a dynamic environment of the federation.

A model aimed specifically for cloud federation is presented in [12]. This model declares the previous research to be unfit for federated clouds due to the unique requirements of cloud-to-cloud interaction. However, the article lacks further elaboration of these limitations and still utilizes commonly available mechanisms. A Joint Trust and Risk Model (JRTM) is introduced for federated cloud services [13]. The model is based on the performance history of a CSP. It addresses provider and consumer concerns by relying on a third party provider to collect various data to perform the evaluation, assessing the risk that can be associated with a CSP for a cloud consumer, based on security, privacy and service performance metrics. However, this model is not directly applicable for a cloud federation, although it discusses the service composition in cloud federation.

A coalitional graph game, called “trust-aware cloud federation formation game” is proposed in [26] to support cooperation among cloud providers in a federation. The proposed approach considers a specific case of Map/Reduce programs while considering reputation among the participating cloud providers to achieve maximum profit for their participation. A cloud provider rates another cloud provider based on its direct interaction which is considered as a local rating. However, the proposed mechanism does not take false feedback and other security vulnerabilities in recommendation based trust mechanisms into consideration. Moreover, the authors have not considered the hierarchical nature of service composition within federated clouds.

Table I: Summary of Recent Literature

Literature	Cloud Model	Architecture	Bi-directionality	Composite Trust	Delegation control	Sources	Representation	Evaluation	Disseminations
[20]	S	C	×	×	×	P	H	SSt	✓
[21]	S	C	×	×	×	P	H	SSt	✓
[22]	S	C	×	×	×	P	H	×	×
[25]	M	C	×	×	×	R	H	SAd	×
[9]	M	C	×	×	×	R/E	×	SAd	×
[11]	M	D	×	×	×	P/R/E	×	×	✓
[10]	M	C	×	×	×	R/E	×	SAd	✓
[14]	M	D	×	×	×	R/E	×	SAd	✓
[23]	S	C	×	×	×	P/R	H	Sst	×
[24]	S	C	×	×	×	P/R	H	SSt	✓
[12]	F	C	×	×	×	P/R	×	SSt	×
[13]	F	C	×	×	×	E	H	SAd	×
[26]	F	D	×	×	×	R	H	SSt	×
[27]	F	P2P	×	×	×	R	H	SSt	×
Proposed	F	C	✓	✓	✓	E	H	HAd	✓

Legend

✓ = Present, × = Not Present, **-/NA** = Not Applicable, **SC** = Single cloud, **M** = Multiple clouds, **F** = Federated cloud, **C** = Centralized management, **D** = Decentralized management, **P2P** = Peer-to-peer, **P** = Policy, **R** = Recommendation, **E** = Evidence, **H=** Homogenous representation, **SAd** = Simple Adaptive, **SSt** = Simple Static, **HAd** = Hierarchical Adaptive, **HSt** = Hierarchical Static

Authors in [27] have proposed a lightweight algorithm based on ratings of cloud providers based on prior interactions. The proposed mechanism is an extension to the Trust Network Analysis with Subjective Logic (TNA-SL) algorithm [28] for cloud environments. The proposed algorithm, however, utilizes recommendation and feedback ratings in the federation. This approach has an inherent risk of being susceptible to malicious feedback and collusion attacks. The authors have not considered service composition or service delivery to accumulate feedback.

Existing literature therefore does not fully deal with the dynamic and hierarchical focus of cloud-to-cloud interaction, and support for integrating services across these clouds.

III. Background and Context

A. Cross-cloud Federation

Cross-cloud federation [4] involves the dynamic sharing of resources among CSPs. Service composition among heterogeneous participants of the federation occurs in layers [5, 29]. In a cross-cloud federation, a request triggered by a home CSP (from a specific consumer) to lease a resource is called a transaction. All other exchanges for resource that originate as a part of this transaction are its sub-transactions. Generally, a transaction usually passes through multiple stages, starting from resource discovery, matchmaking and eventually establishing a relationship between the CSPs [4]. The basic properties of a cross-cloud federation can be summarized as follows.

- *Highly dynamic structure* – relationships between CSPs are short-term and frequently updated. The underlying structure of the federation is therefore highly dynamic.
- *Heterogeneity in service providers* – CSPs joining the federation may have different infrastructures along with a contract language (which may include a number of different quality of service metrics) and security requirements. This results in the scalability and diversity of service composition.

B. Composite service and relationship formation

Considering service as a method of representing, performing and delivering a specific task, a composite service within a federation is integration of various sub-services or service components or resources from various providers. The basic idea of a composite service in the cross-cloud federation can be elaborated by considering a typical scenario illustrated in Figure 1 with four CSPs. A home CSP is providing various SLA based services to consumers (i.e. Individual / End User and Enterprise) while other CSPs are depicted as Foreign CSPs. Each CSP owns a set of distinguished virtualized resources. The Home CSP has one set of additional resources leased from a foreign *CSP-1*. Another set of resources is leased from foreign *CSP-2*, which in turn has leased a part of these resources from *CSP-3* thus supporting resource exchange between CSPs.

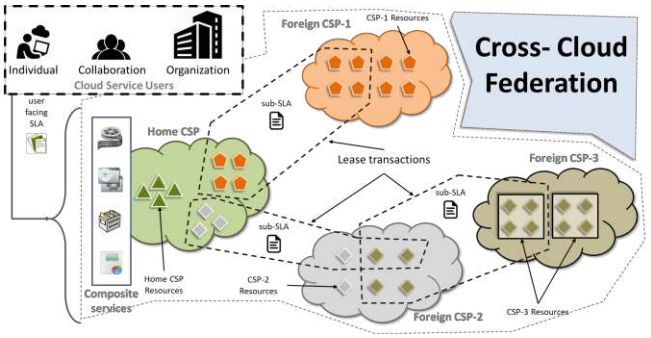

Figure 1: A typical cross-cloud federation scenario

All CSPs joining the federation may have different types of computational infrastructure and support monitoring of different performance metrics. A CSP could be a service provider with a large and sustained user community, or a new market entrant with limited service delivery experience. Each relationship within the federation is governed by rules and agreements, which must be a subset of the contract signed between the home CSP and the consumer [4, 5]. In such a scenario, a home CSP is entirely responsible for service delivery to the customer. The entire mechanism has been depicted in Figure 2 showing the consumer-to-cloud and cloud-to-cloud relationship as two distinct paradigms.

The cloud-to-cloud relationship requires establishing confidence between CSPs of the federation. The performance of a federated service is reflective of the behavior of all its sub-providers including the home CSP. A contract violation by a foreign CSP is going to have a cascading influence on the performance of home CSP [29], ultimately deteriorating its relationship with the customer. Therefore, the capability and competence gathered at the home CSP must be reflective of the entire chain of CSPs involved in the service composition. Any change in the capability of a cloud must be reflected to the upper levels. This global capability assessment of the composite service certifies the home CSP to entrust confidence to the user.

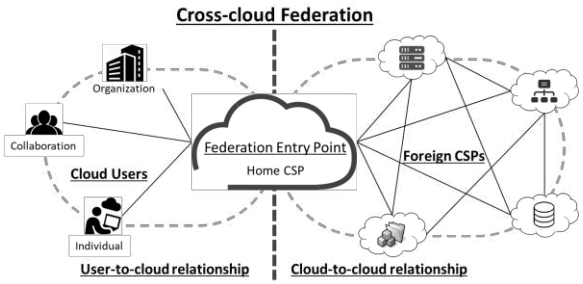

Figure 2 An abstract illustration of relationship breakdown in CCF

C. Capability assessment fundamentals

In the context of distributed and multi-agent systems, the concept of a CSP's assessment is mostly bound to "performance", "security" and "privacy" parameters. Capability assessment mechanisms utilize various indicator values of these parameters collected from various sources based on human behavior, perception and interaction experiences with the system. In a generalized perspective, such information sources can be classified into three categories based on *i) Recommendations*, either direct or transitive, provided to a potential user by others based on their own experience *ii) Verification of contract* signed between the user and the provider to estimate the level of variation from the defined thresholds in policy and *iii) Attribute assessment* to verify the capabilities and competencies of cloud providers.

An efficient peer selection mechanism must rely on more than one source of information for precise decision making [30]. However, selection based on policy verification is not suitable for the federation with heterogeneity in its participants' infrastructures, services, and contracting languages. Moreover, in a cross-cloud federation, the only concern is the cloud-to-cloud relationship establishment, and hence user feedback is not a feasible option. This work utilizes an attribute assessment based selection mechanism so that participating CSPs can be assessed over a commonly defined feature space. This approach tends to be optimistic for CSPs that are new entrants in the cloud market to compete with more mature service providers.

Attribute assessment can be performed either by the service provider itself, cloud auditor, accreditor or cloud broker, etc. The Cloud Security Alliance (CSA) has proposed a detailed and transparent attribute assessment mechanism called the Consensus Assessment Initiative Questionnaire (CAIQ). This CAIQ is a part of the "Security, Trust & Assurance Registry (STAR)" program [19].

D. Consensus Assessment Initiative Questionnaire (CAIQ)

CSA STAR is a three-level program with a free publicly accessible STAR registry. At the first level, it allows CSPs to publish assessments of their security capabilities, in a "Consensus Assessments Initiative Questionnaire (CAIQ)". At level two, an independent third party audit is made for CAIQ attestation and certification of the cloud provider. A mechanism for continuous monitoring based certification at level three is currently under development [19]. CAIQ offers a method to assess the competencies and capabilities of CSPs for different attributes i.e., compliance, governance, security, etc.

CAIQ can be used by cloud providers to outline their security capabilities to customers, publicly or privately, in a standardized method based on Cloud Control Matrix (CCM), which categorizes by the control groups referred hereby as 'capability groups', and then maps to major compliance and regulatory standards [19]. Despite heterogeneity in infrastructures, this standard method of demonstrating capabilities allows a client or a user to analyse, compare, or combine information from multiple CSPs over a homogenous space.

The outcome of CAIQ assessment aims to support clients in making informed decisions before adopting/using CSPs when there are no transaction ratings available (i.e. for new market entrants) or there is a likelihood of false ratings or biased feedback (federation). Afterward, the relationships can be viewed or monitored during actual service enactment to monitor particular QoS measures. Using the information and conversational assertions from the CAIQ, an organization can build a robust Request for Proposal (RFP) and verify that the answers given by the vendor(s) during the RFP review are valid and comparable.

IV. Methodology

The proposed approach makes use of data from CSPs that have attained a level-II certification i.e. the quantitative assessment of CSPs is endorsed by CSA certified third party auditors. CAIQ from the Cloud Security Alliance has been used as an information source in this research and is freely available at the STAR repository in the form of a spreadsheet. This spreadsheet contains a set of 295 assertions that a CSP (or an auditor) answers as either yes, no or not applicable. These assertions are categorized into 133 control groups referred hereby as "capability groups" and 16 control domains referred hereby as "capability domains" grouped by their relevance as in CCM and are shown in Table II. It is mandatory for a CSP to furnish this information once, and subsequently whenever there is a change in the status of its capability.

To carry out the proposed research, we assume that each CSP provides a single service and hence the term CSP is used interchangeably for a service offered by that CSP. Moreover, a peer CSP opting for a service can be interested in all or selective capabilities. Hence, the capability of a CSP can be an aggregated effect of all capability groups in the former case or a selective set of groups in the latter, based on either the type of resource i.e. storage, processing, etc., or the type of business objective i.e. Datacenter Security, Encryption & Key Management, etc. For example, a CSP may be interested in only outsourcing Identity and Access Management (IAM) to another CSP. In such a case, only 40 assertions related to the IAM domain of the offering CSP need to be evaluated as prescribed in the nomenclature of CCM/CAIQ and shown in Table II.

Table II: CAIQ nomenclature

ID	Domains (16)	Groups (133)	Asserts. (295)
AIS	Application & Interface Security	4	9
AAC	Audit Assurance & Compliance	3	13
BCR	Business Continuity Management & Operational Resilience	11	22
CCC	Change Control & Configuration Management	5	10
DSI	Data Security & Information Lifecycle Management	7	17
DCS	Datacenter Security	9	11
EKM	Encryption & Key Management	4	14
GRM	Governance and Risk Management	11	22
HRS	Human Resources	11	24
IAM	Identity & Access Management	13	40
IVS	Infrastructure & Virtualization Security	13	33
IPY	Interoperability & Portability	5	8
MOS	Mobile Security	20	29
SEF	Security Incident Management, E-Discovery, & Cloud Forensics	5	13
STA	Supply Chain Management, Transparency, and Accountability	9	20
TVM	Threat and Vulnerability Management	3	10

The proposed approach has been implemented as a part of a CCF broker. The CAIQ information is fed to the broker via its control interface. The information contained in the spreadsheet is then parsed for extracting and storing answers to each assertion. After the parsing is complete, the auditor's opinion regarding a CSP is represented using the three quantitative scalars i.e., belief (λ), disbelief (γ) and uncertainty (φ) regarding the CSP's capability (C).

Table III: Operators, representation and types

Operators	Representation	Type
p	Positive answer	
q	Negative answer	
un	unanswered	Direct
na	Not applicable	
ε	Initial expectation	
N	Total Applicable	
ρ	Average Positiveness	
δ	Average Negativeness	
$\mathbb{O}$	Opinion	Indirect
ζ	confidence	
λ	belief	
γ	disbelief	
φ	uncertainty	
C	Individual capability	
D	Dependency score	Decision
$\hat{C}$	Aggregated capability	

The proposed approach uses 5 direct, 8 indirect and 3 decision operators as defined in Table III. The stored answer to each assertion is used to evaluate the values for belief (λ), disbelief (γ) and uncertainty (φ) regarding individual capability domains. Each positive answer *'p'* to an assertion means the presence of an attribute and is counted towards an increase in the belief by increasing the average positiveness of the respective capability domain. Whereas, a negative answer *'q'* counts towards the disbelief on that CSP capability by adding to the average negativeness of the domain. Any assertion left unanswered *'un'* is counted towards an increase in uncertainty. The derived opinions are afterward stored in the repository for further evaluation to derive Individual capability *('C')* as and when required. Details of deriving indirect operators from these direct operators have been mentioned in section V. The dynamic nature of federation has been fully captured by the proposed *AgCA* approach by evaluating Aggregated Capability ($\hat{C}$) as a function of individual capability '*C'* and dependency score *'D'* evaluated on the basis of connectivity between CSPs.

V. Modeling Individual Capability assessment

This section presents the details for capability assessment of a CSP when it joins the federation for the first time. It is a two-step process i.e. representation and evaluation as elaborated below. The outcome of individual capability assessment is a numeric score representing confidence in a CSP and its offered services. For the proposed *AgCA* approach, the aggregated effect of all capability groups has been taken into account. However, the same approach is equally applicable in the case of context-dependent capability assessment and can be extended as work proposed in [31].

The proposed approach models the auditor's opinions regarding a CSP as an extension to subjective belief (beta distribution and Dempster-Shafer belief theory [32]) contrary to the Bayesian models. In Bayesian models, assessment is a subjective probability value such that the anticipated outcome is based on currently available evidence along with the prior subjective knowledge. The consideration of the prior knowledge allows the user to integrate their dispositional attribution in the model e.g. in case of recommendation or feedback based models. However, in case of an attribute assessment regarding a CSP in CCF, knowledge regarding a CSP's behavior before joining the federation is not much helpful. Instead, there should be some other mechanism to evaluate the CSP capability based on the amount of current evidence available regarding that CSP i.e. the total number of CAIQ declarations that are applicable in any case.

The proposed extension approach allows the elements of "Subjective Probability" based evidence space to be combined with "subjective logic" based opinion space for those CSPs that have attained level-II STAR certification. Being a level-II certified CSP means that the CAIQ assessment of that CSP is audited, verified and endorsed by trusted third party auditors on the CSA panel and has no involvement from the end-user. Hence, beta distribution holds between the CSP and the auditor with the beta density function being indexed by the two parameters α and β. Given pr as the probability of occurrence of events, the beta (α, β) distribution for binary events ($\varepsilon = 0.5$) can be expressed using a gamma function Γ as:

$$f(pr;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} pr^{\alpha-1}(1-pr)^{\beta-1}$$

$$\text{given } 0 \le pr \le 1, \alpha > 0, \beta > 0 \qquad (1)$$

$$\text{with } pr \neq 0 \text{ if } \alpha < 1 \text{ and } pr \neq 1 \text{ if } \beta < 1$$

$$\text{Expected value} = E(pr) = \int_0^1 pr.f(pr;\alpha,\beta).dpr$$

$$= \int_0^1 pr.\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}.pr^{\alpha-1}(1-pr)^{\beta-1}.dpr \qquad (2)$$

$$= \frac{\alpha}{\alpha+\beta}$$

For non-binary events (when ε=[0,1]),

$$\alpha = p + p_0 = p + 2\varepsilon$$
$$\beta = q + q_0 = q + 2(1-\varepsilon) \quad (3)$$

Table IV: Individual capability representation of five CSPs

	N	*p*	*q*	*un*	*λ*	*γ*	*ϕ*	*C*
S	264	234	30	0	0.8864	0.1136	0	0.8864
A	295	200	95	0	0.6780	0.3220	0	0.6780
B	295	200	30	65	0.8679	0.1302	0.0019	0.8698
C	255	100	100	55	0.4989	0.4989	0.0022	0.5011
Q	219	175	19	25	0.9010	0.0978	0.0012	0.9132

Given p is the total number of positive declarations, p_0 is the prior knowledge for expectation of p, q is the number of negative declarations and q_0 is the prior knowledge for expectation of q. The dispositional structure of CAIQ [33] with $N = (p + q)$ as the total number of declarations that are applicable as prior evidence, when taken into account gives the following results.

$$p_0 = 2 \cdot \varepsilon \cdot (1 - \frac{p+q}{N}), \quad q_0 = 2 \cdot (1-\varepsilon) \cdot (1 - \frac{p+q}{N}) \quad (4)$$

The expected value of individual capability can thus be given as

$$E(pr) = \frac{p + 2\varepsilon(1 - \frac{p+q}{N})}{p+q+2\varepsilon(1-\frac{p+q}{N}) + 2(1-\varepsilon)(1-\frac{p+q}{N})} \quad (5)$$

$$= \frac{pN(p+q)}{[(p+q)N + 2(N-(p+q)](p+q)} + \left(\frac{2(N-(p+q)) + N(p+q)}{(p+q)N + 2(N-(p+q))} - \frac{N(p+q)}{(p+q)N + 2(N-(p+q))}\right).\varepsilon \quad (6)$$

Based on the parameters belief λ, disbelief γ, uncertainty φ, and initial expectation ε. The overall assessment of CSP's capability is defined as $C(\lambda, \gamma, \varphi, \varepsilon) = \lambda + \varphi.\varepsilon$ [32]. Mapping $\lambda + \varphi.\varepsilon$ to $E(pr)$ and thus λ and φ can be finally given as follows.

$$\begin{cases} \lambda = \frac{p}{p+q} . \frac{N.(p+q)}{(p+q).N + 2.(N-(p+q)]} \\ \varphi = 1 - \frac{(p+q).N}{(p+q).N + 2.(N-(p+q))} \end{cases}$$

$$\text{where } \frac{p}{p+q} = \rho, \frac{N.(p+q)}{(p+q).N + 2.(N-(p+q)]} = \zeta \quad (7)$$

$$\text{and } \gamma = 1 - \rho = \frac{q}{p+q}$$

Therefore, for all ε < 1, the extended opinion space for CAIQ based assessment is summarized as follows.

$$\lambda = \rho \cdot \zeta, \quad \gamma = \delta \cdot \zeta, \quad \varphi = 1 - \zeta \quad (8)$$

$$\text{where } \rho = \frac{p}{p+q}, \eta = \frac{q}{p+q} \text{ and}$$
$$\zeta = \frac{N \cdot (p+q)}{2 \cdot (N-p-q) + N \cdot (p+q)} \quad (9)$$

In the equation (9) ρ is the average positiveness of a capability group and δ is the average negativeness of the group. Both ρ and δ are calculated based on p and q for each group. A group is said to have a zero assessment score when $p + q = 0$. Confidence, ζ, is calculated based on *N, p and q,* given *N= (p+q+un) - na*. Based on the above mechanism, the individual capability *C* of a CSP is the average opinion of all groups selected for any given transaction.

To elaborate the concept, assume CSPs *S, A, B, C and Q* having *N*, *p, q* and *un* as specified in Table IV. A three dimensional graphical illustration of these capability parameters for individual CSPs is presented in Figure 3. Belief is represented on X-axis, disbelief on Y-axis and uncertainty on Z-axis. Among these representative CSPs, *Q* is top rated as having the maximum capability score. However, when comparing all CSPs on a precise level, taking into account the detailed belief system, *A* is considered the best suited as having the maximum belief and no uncertainty in it assessment. The concepts of comparison are further elaborated in section (V.B.3).

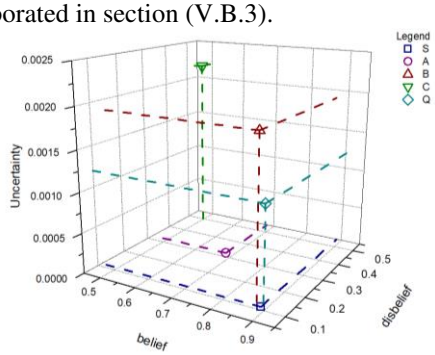

Figure 3 Representation of individual capability of CSPs

VI. AGGREGATED CAPABILITY ASSESSMENT (*AGCA*)

This section describes the proposed Aggregated Capability Assessment (*AgCA*) approach for CAIQ enabled cross-cloud federation. *AgCA* evaluates the cumulative capability of a composite service based on a service dependency model as described in sections as follows.

A. Representation as service dependency model

In the conventional cloud computing model, the credibility of a CSP solely depends on the behavior of its services and attributes of its underlying system. In cloud federation, upholding the same level of credibility is challenging as different cloud providers concurrently strive to deliver a service to the end user. As this service is dependent on multiple providers, arranged in a specific hierarchy, its behaviour must reflect the capabilities of this chain of providers and their complex dependency relation [5, 18]. This dependency relation when taken into account assures that peer selection decisions are authentic and adaptive to the dynamics of the federation.

Keeping in view a simple example as depicted in Figure 4, a service *S* has its dependency on resources from CSP A, which is further dependent on *B* and *C*. Both *B* and *C* are equally dependent on *Q*. Here *S* is the root node and is directly dependent on *A* and indirectly dependent on many others including the leaf node *Q*. The typical scenario of CCF does not allow forming loops structures, since a CSP can neither lease a resource nor form a relationship with itself. However, in the case of circular invocation, it can be unfolded by cloning itself *n* times [34].

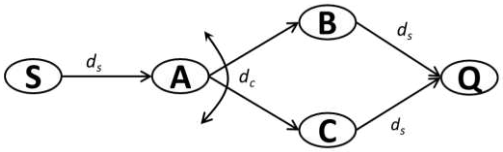

Figure 4: A composite service with sequential and concurrent relations

1) Dependency relations in composite services

In federated services, the dependency between service components may result from a relationship formation due to the sharing of resources with each other at any level of the cloud service model. Such a dependency can be defined as follows.

Definition 1: *In composite services, the fact that the credibility of a service is dependent on the credibility of its basic provider and all its succeeding sub-providers is termed as relationship dependency. Given the composite service S characterized by the occurrence of binary relationship events with V i.e. a set of sub-providers of S, and f being a subjective probability function as defined in the equation* (1)*, we get*

$$f(S) = f(S \rightarrow \forall V) \tag{10}$$

From (10) it can be observed that the probability of occurrence of S can be evaluated by its relationship to all members of V. This dependency may be direct, or otherwise transitive forming a chain of dependency in any given transaction. For example, if a CSP A offering a service S_A to the consumer leases a resource from CSP B, we say that "A depends on B" represented as $A \rightarrow B$. Similarly if *CSP B* leases a resource from *CSP C* for the same service, it is said that *"A depends on B and B depends on C"* thus forming a chain of transitive dependency. Keeping in view the above statements, two types of dependency relations corresponding to the resource federation are identified and are elaborated as follows.

- *Sequential dependency:* A service S_A at SaaS layer is offered from a *CSP A* to the customer *C*. S_A is dependent on another service *B* from *CSP B*. This is termed as a sequential dependency and is denoted as $d_s(A \rightarrow B)$
- *Concurrent dependency:* A service S_A at SaaS layer is offered from *CSP A* to the customer *C*. Service S_A is using additional service *B* from *CSP B* and *C* from CSP_C. This dependency of S_A on both *B* and *C* at the same time is termed as concurrent dependency and is denoted as $d_c(S \rightarrow B,C)$

2) Service Dependency Graph

A *Service Dependency Graph (SDG)* represents the dependency structure of composite service with the following definitions.

Definition 2. *An SDG is a directed graph DG = (V, E, R) with V being a finite set of vertices, E being the finite set of directed edges and R being the set of dependency relation i.e. sequential and concurrent. In SDG, each vertex v ∈ V is a service and e ∈ E is a directed edge given ∀e = (v_1, v_2), where v_1, $v_2 \in V$ and v_1 is the requestor and v_2 is the granter vertex and v_1 is the predecessor of v_2 and v_2 is the successor of v_1.*

Definition 3. *In SDG a service $v_1 \in V$ is said to be dependent on service $v_2 \in V$ given $e = (v_1, v_2) \in E$ or $p = (v_1, v_2) \in P$, given P is a directed path in SDG with v_1 being the start vertex and v_2 is the ending vertex thus forming a transitive dependency, i.e. if $v_1 \rightarrow v_2$, $v_2 \rightarrow v_3$ then $v_1 \rightarrow v_3$.*

Algorithm: Create subgraph(s) from *Service Dependency Graph*

```
Input: A SDG
Output: All possible edges within the graph, primary subgraph, internal
        subgraph(s), root node, leaf node(s), internal node(s)
1  Begin:
2      Let the service dependency root be S and leaf be Q
3      Create stacks all_sub_graph, pri_sub_graph, internal_sub_graph
4      Push S into pri_sub_graph and mark as visited
5      for each successor u of S do
6          Push u into pri_sub_graph
7      Push pri_sub_graph into all_sub_graph
8      for each internal node i of internal nodes do
9          Push i into internal_sub_graph
10         for each successor j of i do
11             Push j into internal_sub_graph
12     Push internal_sub_graph into all_sub_graph
13     return all_sub_graphs, root_node, leaf_node, internal_nodes
14 end:
```

Definition 4. In *SDG*, *the entry vertex without any predecessors is the root of an SDG and the composite service delivered to the end user whereas the leaf is the exit vertex without any successors. A SDG may have at least one or more leaves.*

Based on these definitions, a representation of composite service can be given as

$$SDG = (V, E_p, E_s, R_p, R_s) \tag{11}$$

where

- $V = \{v_i \mid v_i = root/leaf\}$. In a *SDG*, there is only one root but one or more leaves.
- $E_p = \{E_{pi}\}$ and E_{pi} is a set of predecessors of v_i, i.e. $E_{pi} = \{p_{i,j} \mid p_{i,j}$ and $v_i \in V$ and $p_{i,j} \rightarrow v_i\}$
- $E_s = \{E_{si}\}$ and E_{si} is a set of successors of v_i, i.e. $E_{si} = \{s_{ij} \mid v_i$ and $s_{i,j} \in V$ and $v_i \rightarrow s_{i,j}\}$
- R_p represents a set of dependency relations between E_p and V, which includes $d_s(v_i \rightarrow s_i)$ and $d_c(v_i \rightarrow s_i,s_j)$.
- R_s represents a set of dependency relations between V and E_s, which includes $d_s(v_i \rightarrow s_i)$ and $d_c(v_i \rightarrow s_i,s_j)$.

B. Capability assessment of composite services

The proposed approach evaluates the global capability of S as a factor of capabilities from all its successors. At the start of this process, given node(s) V, the entire *SDG* is processed from the root S to the terminal Q to get E_p *and* E_s for V and marks V for obtaining its dependency relationship as depicted in Algorithm. This dependency relationship is composed of its relation with a hierarchy of its successors. Following the same traversal, the *SDG* terminal is finally processed. For Q, $E_p = \{\phi\}$s as having no further dependencies, it is established that $D(Q) = C(Q)$, where D is the dependency score i.e. the capability of a provider depending on it successors and C is the individual capability of a given CSP derived from (8).

Further to this, the global capability assessment requires visiting the chain of successors of Q i.e. $E_s = \{B, C, A, S\}$ such that this traversal is representative of the dependency relation i.e. sequential or concurrent, between all nodes. Since both B and C have individual single dependency on Q, their respective dependency score must reflect the sequential dependency evaluated by the sequential dependency operator '□ '. As node

A is dependent on both *B* and *C* simultaneously, its respective dependency score must reflect the concurrent dependency on both *B* and *C* evaluated by the concurrent dependency operator '⊗'. The overall dependency score of *S* is the sequential dependency score for *S*→*A* relationship. As each dependency represented by either sequential or concurrent structures can be computed, the global aggregated capability value $\hat{C}$ for the *SDG* in Figure 4 can be finally obtained. This entire traversal is presented in (12).

$$\begin{aligned}
&Step\ 1: \quad D(Q) = C(Q)\\
&Step\ 2: \quad D(B) = C(B) \oslash D(Q)\\
&Step\ 3: \quad D(C) = C(C) \oslash D(Q)\\
&Step\ 4: \begin{cases} D(AC)=C(A) \oslash D(C) \\ D(AB)=C(A) \oslash D(B) \\ D(A)=T'(AB) \otimes D(AC) \end{cases}\\
&Step\ 5: \quad \hat{C}_{SDG} = D(S) = C(S) \oslash D(A)
\end{aligned} \tag{12}$$

The consolidated results computed from the traversal of *SDG* are presented in Table V and illustrated in Figure 5(a). The results presented in each row of Table V refer to each step of (12). The final row gives the aggregated capability $\hat{C}$ for service *S*. The results presented in Table V are further elaborated with the underlying details of capability assessment for both sequential and concurrent dependency in the following sections.

1) Sequential dependency operation

According to the subjective probability theory, forming an opinion about an object that depends on another object must be the result of combining their individual opinions in such a way that new opinion reflects the truth of both opinions simultaneously [35]. The dependency score of a sequential structure such as A→*B* or A→C etc. is proposed to be evaluated as follows.

$$D(x) = C(x) \oslash D(y) \tag{13}$$

Where '*Ø*' is the sequential dependency evaluation operator. Given $C(x) = C(\lambda_x, \gamma_x, \varphi_x, \varepsilon_x)$, and $C(y) = C(\lambda_y, \gamma_y, \varphi_y, \varepsilon_y)$, then $D(x) = D(\lambda_{x\varnothing y}, \gamma_{x\varnothing y}, \varphi_{x\varnothing y}, \varepsilon_{x\varnothing y})$ and is computed using this operator as follows.

$$\begin{aligned}
\lambda_{x \oslash y} &= \lambda_x \lambda_y + \frac{(1-\varepsilon_x)\varepsilon_y \lambda_x \varphi_y + \varepsilon_x (1-\varepsilon_y)\varphi_x \lambda_y}{1-\varepsilon_x \varepsilon_y}\\
\gamma_{x \oslash y} &= \gamma_x + \gamma_y - \gamma_x \gamma_y\\
\varphi_{x \oslash y} &= \varphi_x \varphi_y + \frac{(1-\varepsilon_x)\lambda_x \varphi_y + (1-\varepsilon_x)\varphi_x \lambda_y}{1-\varepsilon_x \varepsilon_y}\\
\text{where } \varepsilon_{x \oslash y} &= \varepsilon_x \varepsilon_y
\end{aligned} \tag{14}$$

Considering the CSPs *B, C and Q* from *Table IV*, and their dependency structure as depicted in *Figure 4*, the dependency score computed by (13) and (14) are given in *Table VI* and illustrated in Figure 5 (b).

Table VI: Sequential dependency score for CSPs

Node(s)	*λ'*	*γ'*	*φ'*	*ε'*	*D*
B'=d(B→Q)	0.7833	0.2153	0.0014	0.9801	0.7847
C'=d(C→Q)	0.4508	0.5479	0.0013	0.9801	0.4521
$A_{B'}$=*d(A→B')*	0.5314	0.468	0.0006	0.9703	0.5320
$A_{C'}$=*d(A→C')*	0.3059	0.6935	0.0006	0.9703	0.3065

2) Concurrent dependency operation

Dependency score evaluation of concurrent structures can be viewed as an abstraction of combining opinions for a single object in such a way that reflects both opinions in a fair and equal way. Given '*k*' as a cumulative uncertainty operator, the proposed method to evaluate the dependency score for such concurrent structures is as follows.

$$\begin{aligned}
&\lambda_{x,y} = \frac{(\lambda_x \varphi_y + \lambda_y \varphi_x)}{k}, \quad \gamma_{x,y} = \frac{(\gamma_x \varphi_y + \gamma_y \varphi_x)}{k} \text{ and}\\
&\varphi_{x,y} = \frac{\varphi_x \varphi_y}{k} \text{ with}\\
&\varepsilon_{x,y} = \frac{\varepsilon_y \varphi_x + \varepsilon_x \varphi_y - (\varepsilon_x + \varepsilon_y)\varphi_x \varphi_y}{\varphi_x + \varphi_y - 2\varphi_x \varphi_y}\\
&\text{where } k \neq 0 \text{ and } k = \varphi_x + \varphi_y - \varphi_x \varphi_y,\\
&\text{and for } k = 0
\end{aligned} \tag{15}$$

$$\lambda^{x,y} = \frac{\varpi \lambda_x + \lambda_y}{\varpi + 1}, \ \gamma_{x,y} = \frac{\varpi \gamma_x + \gamma_y}{\varpi + 1} \text{ and } \varphi_{x,y} = 0 \tag{16}$$

$$\text{given } \varepsilon_{x,y} = \frac{\varpi \varepsilon_x + \varepsilon_y}{\varpi + 1}$$

Where $\varpi = \varphi_y / \varphi_x$ is known as the relative reliability between two beliefs λ_x and λ_y. As illustrated in *Figure 4*, CSP *A* is dependent on both *B* and C forming two distinct dependencies as A→*B* and A→C which can be derived from (13) as *D(AB)* and *D(AC)*. However, the dependency score of *A* must reflect both its dependencies on *B* and *C* in a fair and equal way. Considering capability parameters for *A, B* and *C* in *Table IV* and their dependency as depicted in *Figure 4*, the dependency score computed by (15) or (16) are given in *Table VII* and illustrated in Figure 5(c).

Table VII: concurrent dependency score for CSPs

Node(s)	*k*	λ'	γ'	φ'	ε'	*D*
A'=d(A→B',C')	0.0012	0.41878	0.58092	0.0003	0.9703	0.41907

Table V Aggregated capability for service S as evaluated by AgCA

	Node	Predecessors	Successors	Dependency Resolution	λ'	γ'	φ'	ε'	*D(λ,γ,φ,ε)*
1	Q	*B*,C	None	Q' = Q	0.901	0.0978	0.0012	0.99	0.9022
2	*B*	A	Q	$B'=d_s(B \to Q)$	0.7833	0.2153	0.0014	0.9801	0.7847
3	C	A	Q	$C'=d_s(C \to Q)$	0.4508	0.5479	0.0013	0.9801	0.4521
4	A	S	*B*, C	$A'=d_c(A \to B', C')$	0.4188	0.5809	0.0003	0.9703	0.4191
5	S	None	A	$S'=d_s(S \to A')$	0.3713	0.6285	0.0002	0.9606	0.3715

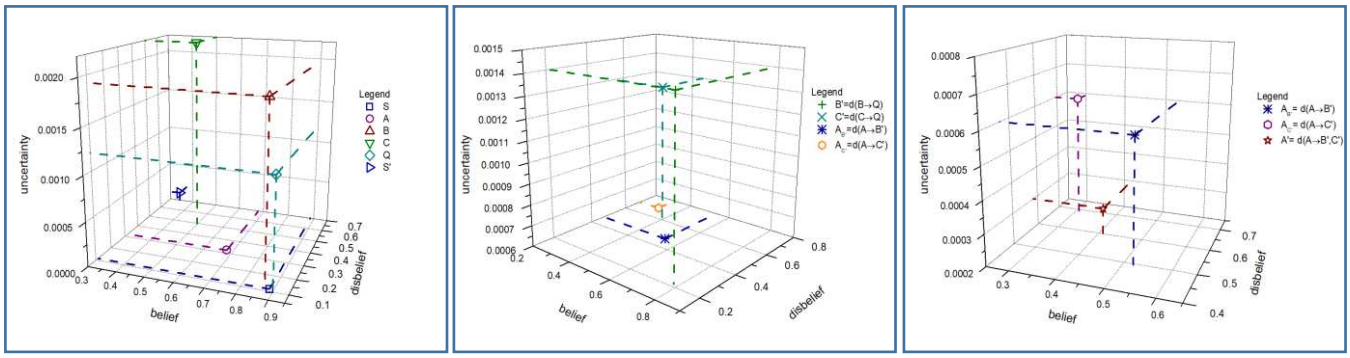

Figure 5: (a) Aggregated capability for service S (b) singular dependency (c) concurrent dependency

3) *Result significance and selection criteria*

The proposed *AgCA* approach aims to deliver a representative view of the credibility of composite services delivered to a consumer by CCF. The significance of the obtained results becomes apparent when used as a CSP selection criteria before engaging in a transaction. When comparing multiple *SDG*s, it is recommended to make a fine-grained comparison regarding their global assessment scores. Considering a case when two *SDG*s have the same aggregated competence score, the one with the greater belief gets a preference over the other. If both have the same belief values, the selection criteria should be based on their disbelief values with a lower disbelief as more preferable than the other one. Given an SDG_x with $\hat{C}_x = (\lambda_x, \gamma_x, \varphi_x, \varepsilon_x)$ and SDG_y with $\hat{C}_y = (\lambda_y, \gamma_y, \varphi_y, \varepsilon_y)$, they are comparable in the following cases:

Case 1: If $|\lambda'(SDG_1) - \lambda'(SDG_2)| < e_1$ and $|\gamma'(SDG1) - \gamma'(SDG_2)| < e_2$, SDG_1 and SDG_2 are equivalent, if and only if $0 < e_1$ and $e_2 << 1$ given e_1 and e_2 are thresholds as specified by service requester. For example, considering service level threshold $0 < e_1$ and $e_2 < 0.002$, two SDGs, SDG_1 having D= (0.7215, 0.21785, 0, 0.99) and SDG_2 having D=(0.7215, 0.2775, 0.001, 0.99) are considered equivalent and comparable for their capability level.

Case 2: If $\lambda'_x - \lambda'_y > 0$, SDG_x is more preferable.

Case 3: If $\lambda'_x - \lambda'_y = 0$ and $\gamma'_x - \gamma_y' < 0$, SDG_y is more preferable.

VII. Experimental Evaluation

This section presents the experimental validation of the proposed approach by discussing the implementation details from system setup and evaluation perspectives. The proposed approach, where applicable, is compared to two different approaches namely Simple Capability Aggregation (SCA) and Numerical Capability Accumulation (NCA). Both approaches are based on CAIQ based CSP selection methods [21, 22, 36] in conventional cloud computing extended to support composite services within the federation. The SCA approach is a simple averaging method and NCA is the dependency graph based aggregation of capability scores of all CSPs involved in service delivery.

A. *CAIQ information*

The proposed approach has been validated against a set of CAIQ self-assessment reports of various CSPs published at the CSA STAR registry [29]. Many other authors have also made use of this data set, such as S.M. Habib *et al.* [20, 21], S. Rizvi *et al.* [22, 23] and Algamdi *et al.* [36]. In this research, we have used the data published in CAIQ v3.0.1 format from a total of thirty CSPs, including Acer Cyber Center Services, Amazon, GitHub, Google, IBM, SAP and Salesforce, etc. The CSA STAR registry enables a standards-based, community wide perspective on cloud security offerings, enabling end users to "accelerate their due diligence and leading to higher quality procurement experiences"[2]. The registry uses several industry standards, such as Cloud Controls Matrix (CCM), CAIQ (as used in this work), and the Cloud Audit and Control Trust Protocol. The CSA has worked with many international certification agencies, e.g. ENISA (European Security Agency) and the Chinese CEPREI, ensuring that the outcome has a wider applicability across many different international markets. The associated CSA STAR Watch initiative provides a tool in a database structure to monitor and assess public and private cloud providers.

B. *System Setup*

The proposed broker is implemented in CPython and executes as a service within a Linux based system. SQLite serves as a repository to store all data related to this system including CSP details, capability scores and transaction details, etc. Provision of service and signing SLA with the end user or the consumer is not in the scope of the broker and is the responsibility of the individual CSP. Several experiments have been performed to validate the notion of aggregated capability and to analyze the suitability of *AgCA* for CCF. These experiments have been performed with the following aims.

[2] https://cloudsecurityalliance.org/media/press-releases/csa-star-registry-surpasses-100-entries-new-csa-star-watch-tool-now-in-open-beta/

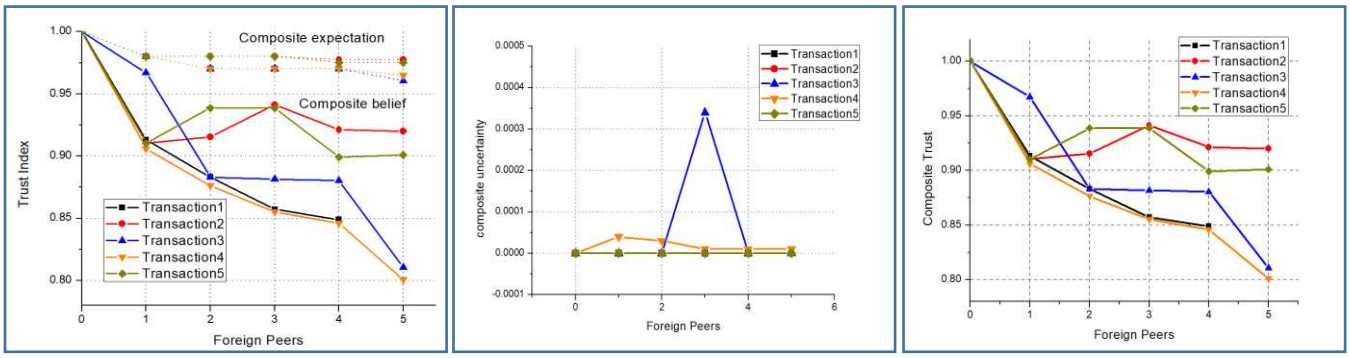

Figure 6: (a) Composite belief and expectation (b) composite disbelief and (c) variation in uncertainty for experiment 1

- To identify generic trends in capabilities as CSPs collaborate on demand to form composite services.
- To identify the necessity to consider dependency structures when evaluating aggregated capability for services composed of more than two CSPs.
- How to identify the best possible arrangement of CSPs with maximum utilization and longstanding relation?
- What is the most preferable dependency structure (d_s, d_c or random) to consider for CSPs with uncertainty in their beliefs?

A service S_x is delivered by a CSP x having capability score $E(S)$. To maintain its user-facing SLA for this service, the CSP requires additional resources/services from other CSP(s). To follow the scope of these experiments, SLA is considered to be comprised of only the threshold for the capability of the service S i.e. $E'(S) \geq threshold$. The nature of resources/services exchanged among these peers is out of the scope of these experiments. Consider the service S_x composed of n components namely S_1, S_2, S_3, S_4 … S_n, such that each component can be delivered by any CSP as they have a similar level of Quality of Service (QoS) and other parameters except their capability. These foreign CSPs interconnect their infrastructures in a random formation to support service delivery to home CSP. Each formation termed as "transaction" randomly consists of sequential and concurrent dependencies as defined in section VI.B. The best possible transaction is then selected to offload the respective service components to continue service delivery to the user.

1) Experiment 1

We have considered a service S from a CSP having capability $C(S) = 0.99$. A threshold limit of $C_{min}(x) = 0.9$ and $D(S) = 0.8$ is enforced in all iterations. A total of 5 candidate foreign CSPs with qualifying $C(x) > C_{min}(x)$ participate in delivering the composite service S such that they are randomly arranged in a way that $D(S) \geq 0.8$.

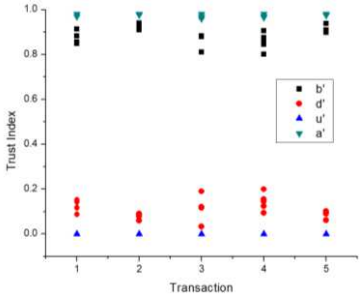

Figure 7 Trend for all capability parameters for 5 CSPs in random

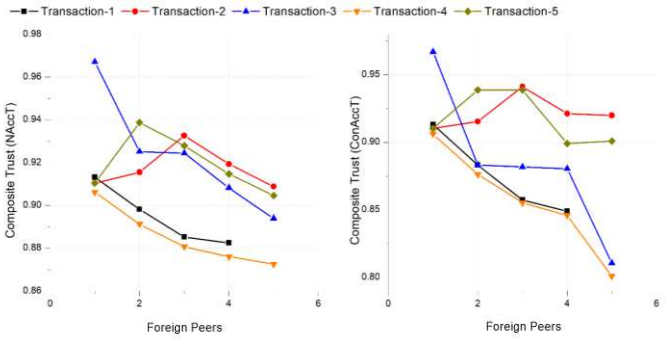

Figure 8: Comparison of NCA and AgCA approaches in experiment-1

The initial values of capability parameters λ, γ, φ and C for each CSP is iterated with an optimistic initial expectation of 0.99 for each CSP. The overall contribution of this experiment is twofold *(i)* to depict the method of identifying an optimal combination of CSPs feasible for any composite service formation, *(ii)* to establish the necessity of taking the dependency structure into account when evaluating capability for composite services.

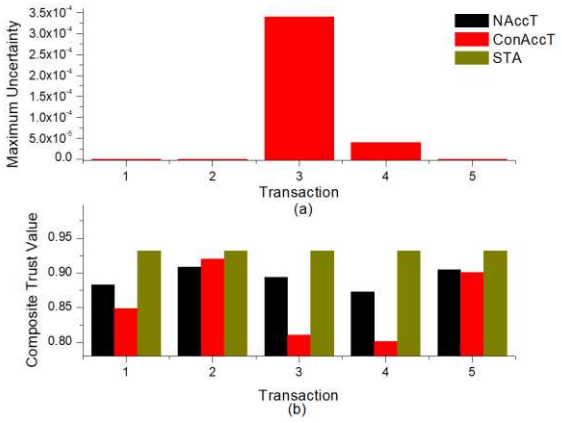

Figure 9: Comparison of NCA, AgCA and SCA approach in case of (a) maximum uncertainty and (b) aggregated capability for experiment 1

Figure 7 shows the overall trend for all capability parameters from CSPs involved in transactions. The arrangement of CSPs

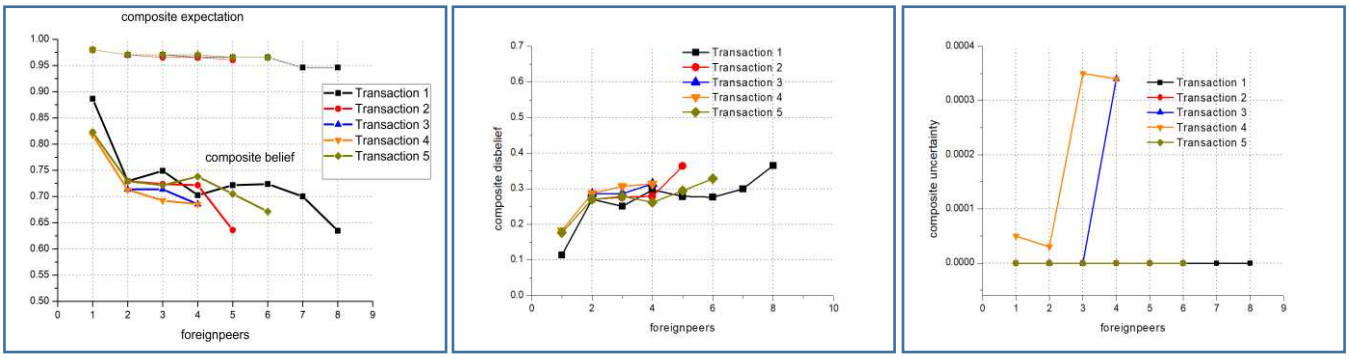

Figure 11: (a) Composite belief and expectation (b) composite disbelief and (c) variation in uncertainty for experiment 2

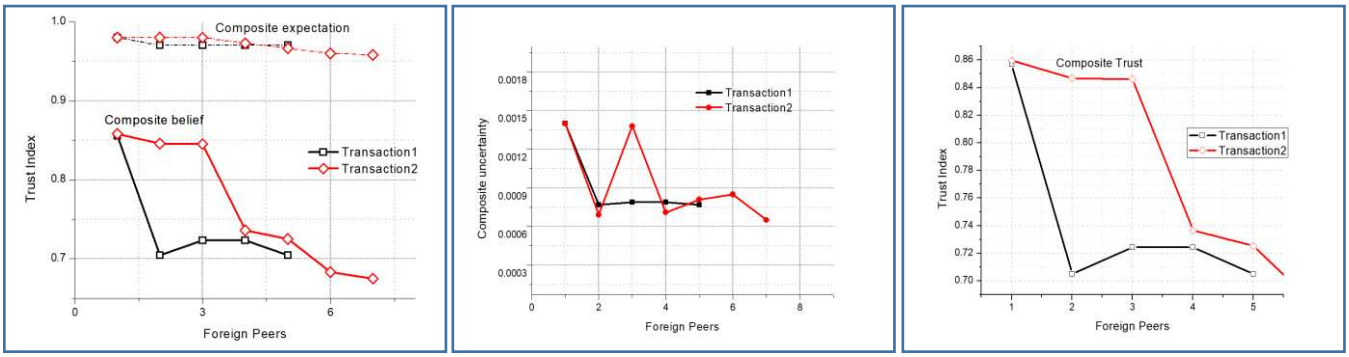

Figure 10: (a) Composite expectation, belief (b) uncertainty and (c) aggregated competence for experiment 3

in a transaction having less scattered capability parameters is the optimal formation, i.e. transaction 2. The closer the values are to each other, the better the chances for a transaction to continue scaling by adding more CSPs.

Figure 6 (a) shows the trend for composite belief and expectation. Figure 6 (b) depicts the variation in uncertainty for all given formations. Figure 6 (c) depicts the aggregated capability for service *S* composed of 5 CSPs combined in random formations. The most preferable formation is the one with maximum capability score i.e. transaction 2. Although all transactions involved the same CSPs, there is a lot of variation in capability scores for all arrangements. This non-linearity is due to our proposed dependency model without which a linear model would have only depicted the same result in any case. The same has been depicted in Figure 8 comparing the results for *AgCA* approach with NCA. Since the NCA approach has used a single numerical value in evaluating capability, therefore the results are less random than the *AgCA* approach. Figure 9 depicts a comparison between *AgCA*, NCA and SCA approaches based on aggregated capability value and maximum uncertainty faced in any transaction.

Figure 9 (a) shows that SCA based capability is always the same for every transaction due to the presence of the same CSPs. Hence this method of obtaining service capability as the average of all CSPs' capability scores is not feasible for dynamic environments like CCF. Moreover, as depicted in Figure 9 (b), the *AgCA* approach has successfully identified the service compositions that have uncertainty in their capability by making them the least possible of all choices. This, however, is not the case with NCA and SCA approach as they lack any consideration for manipulating capability under uncertainty. In both *AgCA* and NCA cases transaction 2 is the best possible option for federating service S.

2) Experiment 2

Experiment 1 is repeated with a threshold $T_{min}(x)$ such that $0.9 \geq C(x) \geq 0.8$ and $D(S) \geq 0.7$ is enforced in all iterations. A total of 8 candidate foreign CSPs are selected with qualifying $C(x) \geq C_{min}(x)$ participate in delivering the composite service *S* such that they are randomly arranged in a way that $D(S) \geq 0.7$. Figure 10 depicts the composite belief, expectation, disbelief and uncertainty in the case of 8 CSPs randomly arranged for composite service formation. Transaction 1 is the most preferable formation among all given formations. The proposed *AgCA* approach has successfully emphasized the benefit of taking part in CCF from the perspective of capability. Since the other two approaches, NCA and SCA does not directly deal with uncertainty, they are unable to highlight such benefits.

3) Experiment 3

For this experiment, we have considered two CSPs providing service S_1 and S_2 with uncertainty in their capability scores. Both CSPs lease resources from the same set of foreign peers, but in different random formations. As is evident from Figure 11, there is a decline in the composite belief of both peers, however, the uncertainty of both services have eventually declined over the period of time. This indicates a potential benefit for CSPs to join the CCF.

4) Experiment 4

To evaluate the effect of the sequential dependency structure on the aggregated capability assessment, a service *S* with maximum capability value is considered *(i.e. b=0.99, d=0.01 and u=0)*.

Figure 12: A chain of singular dependency

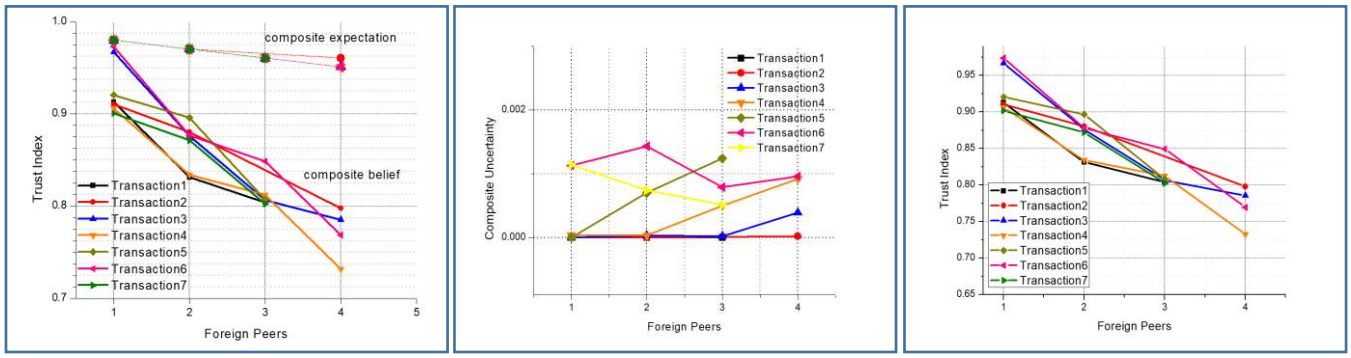

Figure 13: (a) composite belief and expectation (b) uncertainty (c) aggregated capability for sequential dependency federation

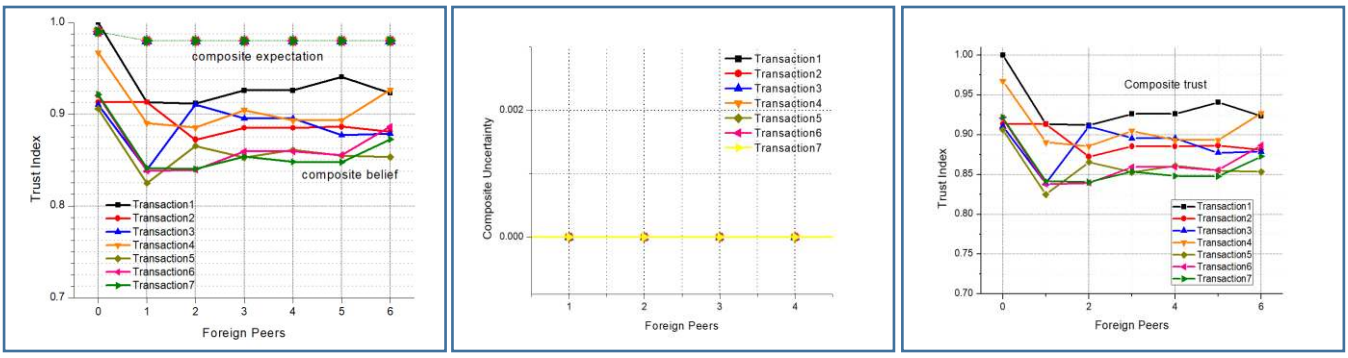

Figure 14: (a) composite belief and expectation (b) uncertainty (c) aggregated capability for concurrent dependency federation

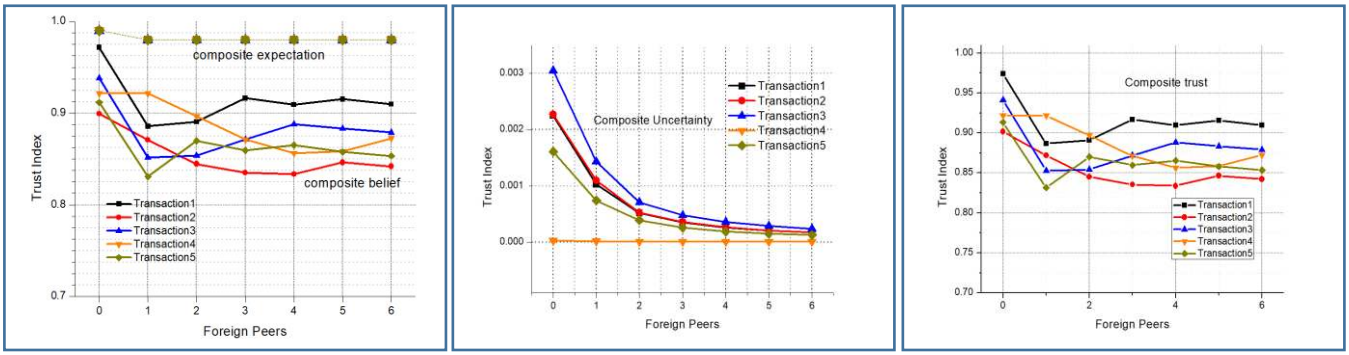

Figure 15: (a) composite belief and expectation (b) uncertainty (c)) aggregated capability for experiment 6

The service *S* and its corresponding foreign CSPs are bound to lease a resource from only one other CSP such that a chain of sequential dependency is formed as shown in Figure 12. A total of 5 foreign CSPs qualify with $C(x) \geq 0.9$ such as to maintain $D(S) \geq 0.8$. An overall observation keeping in view Figure 13 is that sequential dependency results in a sudden decrease of cumulative capability and an increase in uncertainty even when there is no uncertainty in the individual capability of CSPs.

5) Experiment 5

To evaluate the effect of the concurrent dependency structure on the aggregated capability assessment, a service *S* is considered to be offered from random CSPs with different initial capability score. Only the corresponding home CSP of service *S* can lease resources from other CSPs such that a concurrent dependency is formed as shown in Figure 16.

A total of 6 foreign CSPs qualify with $E(x) \geq 0.9$ such as to maintain $E'(S) \geq 0.8$. An overall observation considering Figure 14 is that concurrent dependency results in a gradual decrease of capability and no uncertainty is introduced in a relationship.

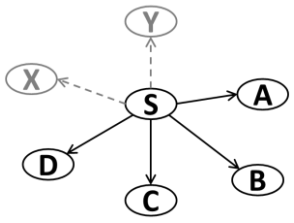

Figure 16: A mesh of concurrent dependency

6) Experiment 6

Experiment 5 is repeated by utilizing a service *S* from different CSPs with uncertainty in their capability scores. Each of these CSPs interact with 6 foreign peers to evaluate the effect of concurrent dependency structure on the aggregated capability assessment. An overall observation is that concurrent dependency results in a gradual decrease of capability scores and also a gradual decrease in uncertainty is observed in the relationship as shown in Figure 15.

VIII. CONCLUSION

This paper has presented *AgCA*, a novel aggregated capability assessment approach for composite services offered within CCF. We suggest that the capability assessment of a CCF is unique due to the hierarchical resource exchanges between multiple heterogeneous CSPs to deliver on-demand composite services. Capability scores for such composite services within CCF are proposed to be evaluated based on the individual capability of all participating CSPs and their dependency relation. The individual capability of each CSP in a federation is evaluated based on its CAIQ assessment as endorsed by CSA. The aggregated capability of any composite service is afterwards evaluated as a function of individual capabilities and the dependency relation of its participant CSPs. This approach is implemented as a component of a centralized broker to increase the reliability and enforcement of peer selection decisions. Numerous experiments have shown that capability values reflected by *AgCA* are adaptive to the service composition structure. Future work includes integrating a competence based approach in this model to enhance the quality of decisions.

REFERENCES

[1] N. Grozev and R. Buyya, "Inter-Cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience,* vol. 44, pp. 369-390, Mar 2014.

[2] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *International Conference on Algorithms and Architectures for Parallel Processing*, 2010, pp. 13-31.

[3] V. Massimo, B. Ivona, and T. Francesco, *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*. Hershey, PA, USA: IGI Global, 2012.

[4] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 337-345.

[5] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, "Thunder in the Clouds: Security challenges and solutions for federated Clouds," presented at the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12), 2012.

[6] R. Chen, J. Guo, D.-C. Wang, J. J. Tsai, H. Al-Hamadi, and I. You, "Trust-based service management for mobile cloud iot systems," *IEEE transactions on network and service management,* vol. 16, pp. 246-263, 2018.

[7] P. Zhang, Y. Kong, and M. Zhou, "A domain partition-based trust model for unreliable clouds," *IEEE Transactions on Information Forensics and Security,* vol. 13, pp. 2167-2178, 2018.

[8] P. Zhang, M. Zhou, and Y. Kong, "A double-blind anonymous evaluation-based trust model in cloud computing environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 2019.

[9] X. Li, H. Ma, F. Zhou, and W. Yao, "T-broker: A trust-aware service brokering scheme for multiple cloud collaborative services," *IEEE Transactions on Information Forensics and Security,* vol. 10, pp. 1402-1415, Jul 2015.

[10] X. Li, H. Ma, F. Zhou, and X. Gui, "Service operator-aware trust scheme for resource matchmaking across multiple clouds," *IEEE transactions on parallel and distributed systems,* vol. 26, pp. 1419-1429, May 2015.

[11] W. Fan and H. Perros, "A novel trust management framework for multi-cloud environments based on trust service providers," *Knowledge-Based Systems,* vol. 70, pp. 392-406, Nov 2014.

[12] A. Kanwal, R. Masood, and M. A. Shibli, "Evaluation and establishment of trust in cloud federation," in *8th International Conference on Ubiquitous Information Management and Communication*, 2014, p. 12.

[13] E. Cayirci, A. Garaga, A. Santana, and Y. Roudier, "A cloud adoption risk assessment model," in *IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC '14)* 2014, pp. 908-913.

[14] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Towards Trustworthy Multi-Cloud Services Communities: A Trust-based Hedonic Coalitional Game," *IEEE Transactions on Services Computing,* vol. 11, pp. 184-201, Jan 2018.

[15] S. Pang, Q. Gao, T. Liu, H. He, G. Xu, and K. Liang, "A behavior based trustworthy service composition discovery approach in cloud environment," *IEEE Access,* vol. 7, pp. 56492-56503, 2019.

[16] Q. Wu, M. Zhou, Q. Zhu, and Y. Xia, "VCG auction-based dynamic pricing for multigranularity service composition," *IEEE Transactions on Automation Science and Engineering,* vol. 15, pp. 796-805, 2017.

[17] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, "Thunder in the Clouds: Security challenges and solutions for federated Clouds," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 2012, pp. 113-120.

[18] U. Ahmed, I. Raza, and S. A. Hussain, "Trust Evaluation in Cross-Cloud Federation: Survey and Requirement Analysis," *ACM Computing Surveys (CSUR),* vol. 52, p. 19, 2019.

[19] cloudsa, "CSA Security, Trust & Assurance Registry (STAR) - Cloud Security Alliance," 2018.

[20] S. M. Habib, V. Varadharajan, and M. Muhlhauser, "A trust-aware framework for evaluating security controls of service providers in cloud marketplaces," in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '13)* 2013, pp. 459-468.

[21] S. M. Habib, S. Ries, M. Mühlhäuser, and P. Varikkattu, "Towards a trust management system for cloud computing marketplaces: using CAIQ as a trust information source," *Security and Communication Networks,* vol. 7, pp. 2185-2200, Nov 2014.

[22] S. Rizvi, K. Karpinski, B. Kelly, and T. Walker, "Utilizing Third Party Auditing to Manage Trust in the Cloud," *Procedia Computer Science,* vol. 61, pp. 191-197, Jan 2015.

[23] S. Rizvi, J. Ryoo, Y. Liu, D. Zazworsky, and A. Cappeta, "A centralized trust model approach for cloud computing," in *23rd Wireless and Optical Communication Conference (WOCC),* , 2014, pp. 1-6.

[24] A. Algamdi, F. Coenen, and A. Lisitsa, "A trust evaluation method based on the distributed Cloud Trust Protocol (CTP) and opinion sharing," *provider,* vol. 5, p. 18.

[25] A. Celestini, A. L. Lafuente, P. Mayer, S. Sebastio, and F. Tiezzi, "Reputation-based cooperation in the clouds," in *IFIP International Conference on Trust Management*, 2014, pp. 213-220.

[26] L. Mashayekhy, M. M. Nejad, and D. Grosu, "A Trust-Aware Mechanism for Cloud Federation Formation," *IEEE Transactions on Cloud Computing,* 2019.

[27] H. Kurdi, A. Alfaries, A. Al-Anazi, S. Alkharji, M. Addegaither, L. Altoaimy*, et al.*, "A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments," *The Journal of Supercomputing,* vol. 75, pp. 3534-3554, 2019.

[28] A. Jøsang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *Proceedings of the 29th Australasian Computer Science Conference-Volume 48*, 2006, pp. 85-94.

[29] A. Al Falasi, M. A. Serhani, and R. Dssouli, "A model for multi-levels SLA monitoring in federated cloud environment," in *10th International Conference on Ubiquitous Intelligence and Computing and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC '13)*, 2013, pp. 363-370.

[30] J. Huang and D. M. Nicol, "Trust mechanisms for cloud computing," *Journal of Cloud Computing: Advances, Systems and Applications,* vol. 2, p. 9, Dec 2013.

[31] U. Ahmed, I. Petri, O. Rana, I. Raza, and S. A. Hussain, "Risk-based Service Selection in Federated Clouds," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 77-82.

[32] A. Jøsang, "The right type of trust for distributed systems," in *workshop on New security paradigms*, 1996, pp. 119-131.

[33] S. Ries, "Trust in ubiquitous computing," Technische Universität, 2009.

[34] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for Web services selection with end-to-end QoS constraints," *ACM Transactions on the Web (TWEB),* vol. 1, pp. 6-es, 2007.

[35] A. Jøsang and D. McAnally, "Multiplication and comultiplication of beliefs," *International Journal of Approximate Reasoning,* vol. 38, pp. 19-51, 2005.

[36] A. Algamdi, F. Coenen, and A. Lisitsa, "A trust evaluation method based on the distributed Cloud Trust Protocol (CTP) and opinion sharing," presented at the International Conference on Computer Applications & Technology, (ICCAT'17), 2017.