On Resilience in Cloud Computing : A survey of techniques across the Cloud Domain

THOMAS WELSH, University of Limerick, Republic of Ireland
ELHADJ BENKHELIFA, Staffordshire University, UK

Cloud infrastructures are highly favoured as a computing delivery model worldwide, creating a strong societal dependence. It is therefore vital to enhance their resilience, providing persistent service delivery under a variety of conditions. Cloud environments are highly complex and continuously evolving. Additionally, the plethora of use-cases ensures requirements for persistent service delivery vary. As a contribution to knowledge, this work surveys resilience techniques for cloud environments. We apply a novel perspective using a layered model of traditional and emerging cloud paradigms. Works are then classified according to the resilinets model. For each layer, the most common techniques with limitations are derived including an actor's strength in influencing resilience in the cloud with each technique. We conclude with some future challenges to the field of resilient cloud computing.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Networks → Cloud computing**; *Network security*; *Network mobility*; • **Computer systems organization → Dependable and fault-tolerant systems and networks**; • **Security and privacy**;

Additional Key Words and Phrases: resilience, cloud, fog, edge, survey

ACM Reference Format:
Thomas Welsh and Elhadj Benkhelifa. 2020. On Resilience in Cloud Computing : A survey of techniques across the Cloud Domain. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, Article 1, 35 pages. https://doi.org/10.1145/3388922

1 SCOPE: CLASSIFYING RESILIENCE FOR THE CLOUD

Resilience, in the context of computer systems and networks, is defined in many ways. Some consider it synonymous with fault-tolerance [84]. Laprie provides two descriptions: "the persistence of dependability when facing changes" and "the persistence of service delivery that can justifiably be trusted, when facing changes" [62]. Sterbenz et. al. provides a similar definition: "the ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation" [112]. Queiroz et. al. suggests: "Resilience is the capacity of critical services to adapt in order to provide their functionalities in cases of undesired events compromising parts of the system." [95].Abdullah et. al. [1] considers a business/organisation perspective : "Resilience refers to the capacity of human beings/system/organization to survive and thrive in the face of adversity...it is a property that is closely associated with the capacity to avoid, contain and mitigate accidents". Or simply, : "the percentage of lost traffic upon failures" [67].These few definitions indicate the numerous factors which can be considered during the development and deployment of a resilient system. This variety is related to the variety of fields in which resilience is applied. As each will have different

Manuscript submitted to ACM

characteristics, numerous measurement methodologies will appear. Additionally, specific use-cases may omit certain characteristics due to their lesser relevance.

In this survey, we adhere to the comprehensive definitions of the Resilinets model by Sterbenz et al. They cover a variety of measurable or desirable characteristics of resilience, grouped into *trustworthiness* and *challenge tolerance*, viewed as internal and external factors respectively. These may be adapted as appropriate in order to take into account novel features of the cloud. We direct the reader to the resilience discipline definitions categorised within [112].

1.1 Defining the Cloud

Cloud computing is a service-driven computing model whereby an end-user will provision and use computing resources from a Cloud Service Provider (CSP) in line with an agreed upon Service Level Agreement (SLA). The service hosted by the CSP could take many forms. Consisting of networking, storage or computational components [76]. Similar to traditional computing environments, cloud environments are multi-layered. The composition differs depending upon the CSP infrastructure, the application's use-case or the particular model used for analysis.

A typical cloud datacentre would consist of the underlying physical infrastructure: servers, storage arrays and networking hardware. Virtualised Infrastructure (VI), a pool of resources: virtual machines (VMs) and/or containers running atop of virtual machine monitors (VMM)s with Virtual Storage (VS) devices and Virtual Networks (VNs). These resources are situated upon the Physical Infrastructure (PI) hardware, connected by Physical Networking (PN). A management layer coordinates physical Resource Management (RM) and the service life cycle. Performance is managed through distributing services using Load Balancing (LB). Services are created and managed using Service Orchestration (SO) and executed using Service Scheduling (SCH). Further service-oriented capabilities such as security are also provided.

The datacentre (DC) architecture is relevant when examining resilience within cloud infrastructure as it is the foundations upon which the cloud service will sit. However the resilience of the DC is not always relevant to the resilience of a service being hosted. For example a cloud service may straddle multiple forms of infrastructure and secondly the user/CSP may have no ability to affect the resilience at this layer, dependent upon the cloud service delivery model employed.

In the NIST definition for cloud computing [76] the prominent service delivery models are defined as a layered architecture: Software-as-a-Servce (SaaS), Platform-as-a-Service(PaaS) and Infrastructure-as-a-Service (IaaS). Responsibilities (for the management / configuration / security etc.) of the service being delivered vary between CSP and the user. This division of responsibility is an important concept within the context of resilience, as the level of control given may determine the user's abilities to affect its resilience. NIST defines a further 3 actors: auditor, broker and carrier.

Due to emerging disciplines and delivery models, matters are complicated further. In addition to those layers discussed above there are layers within the decentralised cloud. Once considered to be an emerging discipline, Cloud computing is now arguably emerged, although is constantly evolving. In tandem with new technologies and use-cases, new forms of cloud computing are developed to accommodate emerging disciplines such as the internet of things (IoT) and big data. These involve distributing the cloud services across devices or network architectures dissimilar to the typical DC only model.

Bilal et. al. explains: "different emerging technologies situated at the edge of the network to provide computational and storage resources to deliver real-time communication with minimum latency" [20]. While Baktir et. al. explain that despite differences, these disciplines all largely attempt to accomplish the same goal and are variations of edge

disciplines. What varies is their use-case, and presumably the underlying technologies in which the new processing occurs [11] .

A summary of these emerging disciplines are below:

- **Fog Computing** - seen first as an extension to the cloud but now as complimentary or independent from it. It involves a hierarchy of services where some processing/storage is executed closer to the edge of the network whilst analytics can occur in the cloud. This can occur in small-scale clouds but also on a variety of different hardware such as base stations, routing hardware, etc. [98] [82] [20] [90]
- **Mobile Cloud Computing (MCC)** - the concept of resource augmentation from a mobile to a remote device in order to maximise resource efficiency and power consumption. Originally intended for centralised cloud DCs, the potential for processing at the edge is now seeing interest [124] [20] [98].
- **Cloudlets** - involve the deployment of small clouds, used to reduce short falls in mobile cloud computing [20] [2] .
- **Mobile Edge Computing (MEC)** - provides cloud services at the edge of cellular networks such as 5G nodes, this increases performance through latency reduction, traffic optimisation and enhanced services e.g. location-driven [98] [72] [73]. [90] [20] [124] .
- **Mist Computing** - pushes data processing services as far as possible to the sensor and actuator devices [93] [119]

These definitions illustrate that decentralised disciplines involve distributing cloud services closer to the edge of the network, where the end-user device, sensor or actuator will be. Within the context of this work, in order to manage the complexity associated with non-standardised and evolving definitions, these cloud disciplines are grouped into 3 layers. This creates a new hierarchy of centralised and decentralised cloud architectures, where services may be positioned in one or more layers. The topmost layer is the centralised cloud infrastructure within a data centre. The middle layer is the fog, where cloud services and data processing can occur during transit to the cloud or in a constrained manner upon devices closer to the application edge. The final layer, mist, is where the sensors, actuators and user devices sit and where minimal processing may occur. This model represents the hierarchical layered cloud family of disciplines, components of these disciplines (i.e. the physical devices, protocols and actors) sit within these layers.

The following points are made considering resilience in emerging cloud disciplines:

- The cloud infrastructure's distinct architecture is relevant to understanding it's own resilience but not always responsible for guaranteeing service resilience. Therefore the relationship between resilience techniques operating in lower-levels and a service on a higher-level should be established.
- Emerging disciplines cause services to be delivered on decentralised cloud infrastructure far away from the DC, sometimes independently from it.
- The chosen service delivery model will affect the ability of the user or CSP to adjust the resilience of the service. Therefore this is a key factor in resilience technique selection. All delivery models can be employed upon all architectures although with greater constraints closer to the edge.

We illustrate in figure 1 the relationship between the centralised and decentralised cloud disciplines and their underlying architectural constituents. The diagram shows that cloud disciplines (coloured) may span one or more architectural layers, potentially encompassing a variety of different hardware configurations in addition to physical and logical architectures.

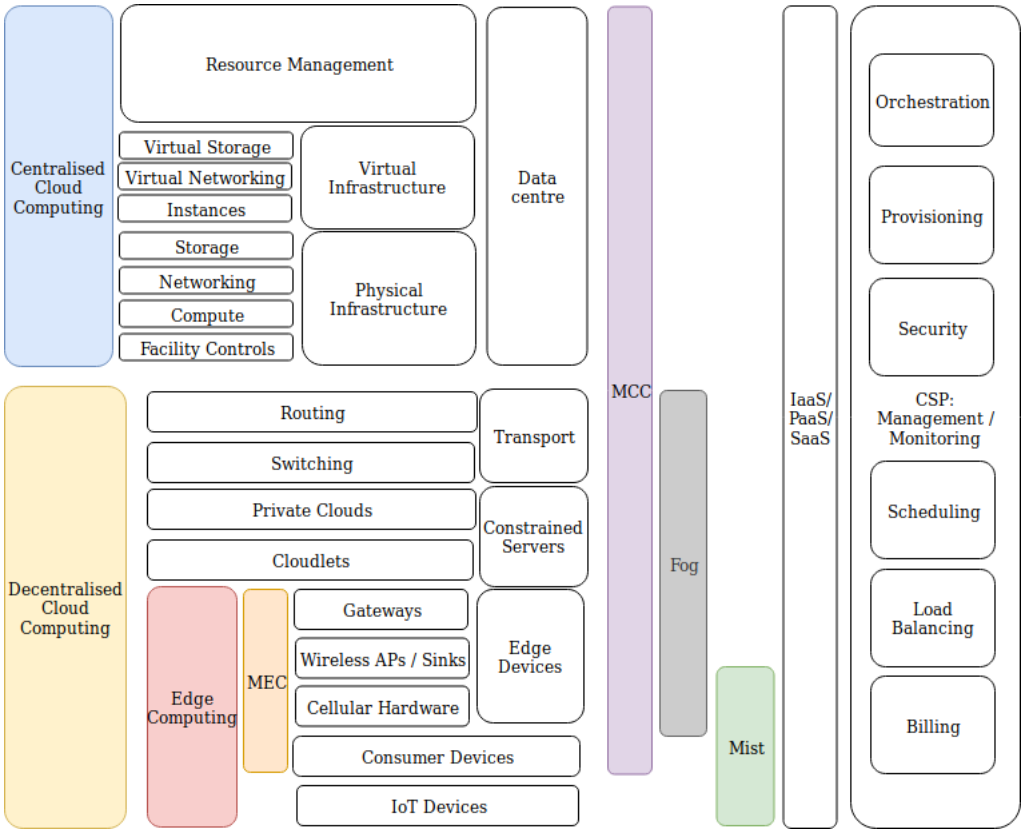

Fig. 1. Decentralised cloud computing model. Illustrating the relationship and overlap between different cloud models and architectural components. MEC=Mobile Edge Computing, MCC=Mobile Cloud Computing

1.2 Related Work in Cloud Resilience

Due to the novelty of this area, surveys in resilience are limited, this work extends our previous survey in [125], which lacks the detailed analysis according to techniques and disciplines presented here, in addition to the further analysis of the decentralised cloud. Moreover, this work compliments our previous survey on intrusion detection for resilient IoT [17], which is out of the scope of this paper.

A few surveys encompass some aspects of this work. Cheraghlou et. al. provides a survey of fault tolerant specific architectures in the cloud [29], Milani et. al. presents a survey of data replication techniques in the cloud [78] and Mistrik et. al. discusses fault tolerant workflow management techniques [91]. Colman-Meixner et.al present the only survey on resilience techniques in cloud computing infrastructure [30]. Employing a layered model, they provide an in depth study involving classification of resilience approaches used. Their primary findings highlight that for cloud systems, replication and checkpointing are the most common techniques for storage, virtualisation and migration from the storage side and multi-layer protection for networking. However, a considerable number of techniques they evaluated are based on non-cloud environments which may be applicable to the cloud. That approach is avoided in this work due to its open-ended nature. Additionally their model fails to account for emerging/decentralised cloud

disciplines. Without this consideration it becomes difficult to understand how integration and interoperability between emerging disciplines will affect the resilience at different layers. In contrast, this review will focus upon resilience techniques which enable a given CSP to deliver services resiliently upon their platform using any delivery model.

As a contribution to knowledge, we propose a model which encompasses the decentralised cloud and its relationship with the centralised cloud. We review only cloud-centric techniques. Definitions from the Resilinets model [112] are then applied to provide a rigid classification of techniques employed throughout literature. Combining these two models, each work can now be classified appropriately in terms of cloud layer, cloud components used and resilience disciplines enabled. This is important to manage the scope and complexity of the survey. This survey classifies work in cloud service resilience according to a number of factors:

- Where in the hierarchy of centralised/decentralised cloud disciplines the work is situated.
- The architectural components according to the model in figure 1 the work applies to.
- The cloud delivery model that the work applies to.
- The resilience disciplines that apply to the work, according to the resilinets model [112]
- The techniques used to accomplish the resilience.

The work will therefore be considered within the following categories which are representative of the centralised/decentralised cloud environment.

(1) **Physical and data-centre resilience** - techniques used to enhance the resilience of the datacentre. Placed here for completeness and to illustrate the underlying resilience
(2) **Virtual Resource Abstraction** - the resilience of virtualised/abstracted resources such as VS, VI, VMMs etc
(3) **Cloud Management** - techniques associated with cloud middleware / management e.g. orchestration of services
(4) **Decentralised Cloud** - resilience techniques for cloud environments which are closer to the network edge.
(5) **Alternative Architectures** - methods using unconventional architectures.

1.3 Paper selection criteria

Papers were selected for inclusion after searching popular databases, namely ACM Library, IEEE Explore, Science Direct and Google scholar. A number of keyword permutations were used which involved "resilience" combined with different cloud computing models and architectures e.g. "resilient PaaS", "Cloud resilience" etc. During this search it quickly transpired that resilience definitions vary considerably from author to author. This is the motivation for leveraging the resilinets model [112] as it provided a consistent definition and classification of these disciplines. This model is also frequently found referenced in other works and therefore provided support to its accuracy. These disciplines then replaced the resilience keyword with variations of cloud computing to procure more works for evaluation.

Scope management is an important issue due to the range of disciplines involved and the complex nature of the cloud. Firstly, papers were omitted which did not focus solely upon cloud environments. While intuitive, the authors in [30] include a number of non-cloud works which could still be applicable. Although relevant to resilience in general it was decided this these papers could skew our analysis and could cause the review to be open ended. The one exception is works which focus upon data-centre resilience where deemed relevant. In a similar manner a number of security-oriented papers were excluded as techniques for firewalls, intrusion detection systems, etc. can largely be applied anywhere but the authors' did not have a cloud resilience specific goal. Finally, works older than 10 years at the time of first selection were omitted. Overall, these works were selected to answer the following: What was the current

state of the art in cloud computing resilience, including what techniques were used, how effective are they and what are their limitations? It was also conducted in order to identify any further gaps in the field to provide situation of the work.

2 STATE-OF-THE-ART IN CLOUD RESILIENCE

This section surveys literature according to the layers defined previously. Each work is analysed according to the techniques used and cloud components applied. A summary table is given for each section.

2.1 Physical and data-centre resilience

DCs are used as a strong argument that the security and resilience of cloud computing is greater than that of traditional paradigms. Largely because cloud infrastructures are typically hosted in DCs with greater facilities than those possible to finance or manage by a single organisation. Highly redundant resources, excellent power resilience, excellent physical security and strong network links to the internet backbone. Therefore these underlying characteristics ensures these environments are resilient by nature. This section discusses a number of works in this layer which attempt to improve this further. Table 1 summarises this literature.

Mohamed discusses the area comprehensively, without the cloud context. He reiterates the inherent resilience of data centres and evaluates routing protocols, load balancing and graph analysis techniques to providing enhanced resilience [81].

Lou et. al. consider Cloud network survivability and sustainability within the context of energy aware solutions: Energy aware Backup Protection (EABP) [71]. They argue that as the requirement for energy efficient services increases, survivability and resilience should not be ignored. They present a new model which allows multiple links to share one backup, drastically reducing capacity requirements with only a small increase in energy consumption. It is not certain whether the system will maintain resilience in the case that the number of links failed exceeds the capacity of the backup. Goścień and Walkowiak also consider survivability [44]. They undertake a study to investigate the physical placement of DCs along optical fibre links from a topology and demographic-economical perspective. They illustrate that placement policies have a strong impact upon survivability. Applying a monte-carlo tree search for resource allocation has been shown to optimise this process [3].

Conversely, Couto et. al. illustrate that survivability techniques including placement and redundancy can have grave negative effects upon the latency of applications in highly survivable situations (up to 80% degradation) [33]. Which is intuitive yet relevant. They also discuss the design for clouds which must be resilient in the face of disasters [110], in contrast to many surveyed works. Their motivation is that most SLAs do not cover disaster resilience, (e.g. hurricanes). A key component for their resiliency is geographical distribution and fail-over systems for activation during the event of a failure. They present a methodology for developing disaster resilient networks and also a VM placement algorithm [32].

Zhang et. al. propose resource orchestration as a technique to enable survivability in optical networks through minimising datacentre provisioning [130]. Additionally they aggregate backup "k-node" links for multiple users to improve surivivability in a resource optimal manner upon disaster [64].

Chandna et. al. present a survivability solution in optical DC networks using Software Defined Networking (SDN) [27]. They illustrate the strength of these techniques and highlight that most future methods are likely to employ SDN. However whilst geo-distribution improves resiliency it does not guarantee it. They highlight the necessity for VM placement algorithms as a high priority for guaranteeing resilience for cloud SLAs. This work highlights the need to

understand the differing network layers when considering resilience in the cloud. Therefore an important question is: how effectively can resilient virtual networks be designed without information about its lower layers?

Table 1. Data and Physical Layer Resilience

Method	Model	Components	Disciplines	Limitations
Back-up links [71]	Phy	PI	SRV	High Cost
Optimal geo-distribution of datacentres [110]	Phy	DC's	DT	High cost
Server placement for VM backups [32]	IaaS	PI; DC	DT	Physical layer only
Optimal routing using load balancing and graph analysis [81]	Phy	PI; DC;	R,FT	Non-cloud
Topology and demographic/economic focused dc placement [44]	Phy	PI; DC;	SRV	Only available to some CSP
Monte-carlo tree search for resource provisioning [3]	Phy	PI; DC; RM	SRV	Only available to some CSP
Datacentre provision for resource optimisation [130]	Phy	PI; DC; RM	SRV	High Cost and complexity
User aggregated link sharing [64]	Phy,	PI; DC; RM	SRV	Limited compartmentalisation
Simulating survivability scenarios [33]	Phy	PI; DC	SRV	Simulation only
Leverage SDN for survivability [27]	Phy	PI; DC; VN; RM	SRV	Centralised management

PI=Physical Infrastructure, DC=Data Centres, RM=Resource Management, VN=Virtual Networking, SRV=Survivability, FT=Fault Tolerance, R=Resilience, DT=Disruption Tolerance

2.2 Virtual Resource Abstraction

The works reviewed in the following sub-sections describe techniques which are used to provide resilience to a particular VR such as VS, instances, VNs or the VMMs such as hypervisors.

2.2.1 Storage. The most simplistic method of ensuring resiliency of data is replication. When a failure happens, copies are accessed or migrated. This principle is the basis of many storage resiliency techniques such as the commonly deployed Redundant Array Inexpensive Disks (RAID). Storage, however, is expensive so techniques to reduce the cost of redundancy are sought after. This is not just the cost of the hardware but also methods ensuring consistency. There is a current drive to move the majority of an organisation's storage to cloud infrastructures, due to their inherent storage resilience upon mass redundancy of resources and global remote access. Therefore as the standard cloud could be considered a resilient storage mechanism, the techniques discussed will focus on more extreme scenarios or optimisation techniques. Table 2 summarises this literature.

Jaiswal et. al. present Resilient Storage Cloud Map (RSCMap) [53]. It's goal is optimising the design of resilient cloud storage via disaster recovery planning. It permits an appropriate replication function to be selected according to the data type, needs and cost available. This is applicable to a variety of use-cases. The efficacy of relying on disaster recovery for resilience is questionable due to the number of situations (e.g. data leakage, safety-critical systems) where resilience is required prior to a system fault.

Westmark provides another technique focusing upon resilience to disasters (disruption tolerance) [126]. They propose Rapid Data Evacuation (RDE) in which a-priori knowledge of an imminent disaster permits a heuristic to determine the external network links with the least delay. This enables safe and quick data migration from the cloud. Similarly, Yao et.

al. determine the shortest window necessary for backups in order to optimise the cost of replication in case of a disaster scenario [129].

An alternative to needing to evacuate data is to ensure it is replicated across diverse locations in the first instance. Gonzalez et. al. present a network overlay to optimise data management across multiple data-centre sites [43]. Bessani et. al. develop a software library called DEPSKY which utilises consumer cloud storage solutions from multiple providers to provide diverse replication for object storage [19]. Whereas Matos et. al. give a different perspective providing the resilience from the client through enhancing the security of the client machine [75].

Methods of optimising the replication, typically for cost reasons can be frequently found. Calis and Koyluoglu focus upon mitigating blocks (groups of storage nodes) which fail simultaneously [24]. Using Block Failure Resilient codewords, which facilitate the recovery of a block from neighbour blocks utilising load balancing. Nachiappan et. al. conducts a survey on both coding and replication techniques [83]. They suggest that a hybrid method of both is the only cost effective way of guaranteeing storage resilience in the cloud.

Liu and Shen argue that the majority of resilient storage techniques in the cloud are effective at either correlated or non-correlated failures [68]. They propose a multiple failure resilient replication scheme (MRR) accommodating both forms. They use a nonlinear integer programming approach to accommodate multiple objectives. Which are a reduction in network latency to optimise consistency and the optimal number of replications and storage upon inexpensive media. Whilst still maintaining high-availability.

Qu and Xiong attempt to optimise the quantity of needed replications for a specific use-case which is unknown quantity of web requests [94]. Their method of replication across all nodes is in contrast to the typical micro-services architecture and raises questions of scalability and compartmentalisation. Wang et. al. focuses simply on ensuring the integrity of data in cloud infrastructures [123]. Their technique is unconcerned with whether the data is altered maliciously or through hardware fault. A block-based storage mechanism enables the location of data corruption to be identified to permit fast recovery. Finally, Yanez-Sierra take a holistic approach by attempting to provide resilience through modelling the entire data workflow, from client to cloud [128]. Their component-based approach allows modularity and features to be applied and examined at each stage.

2.2.2 Instance and VMM Techniques. Within this section, application layer resilience is referred to as examining an individual instance such as a VM or container and not considering the platform of multiple applications. Table 3 provides a summary.

Nguyen et. al. argue that network and system fault-tolerance is well covered by conventional architectures and therefore they focus upon application resilience [85]. They propose a test bed facilitating error detection and recovery of cloud applications. They explain that errors may be detected long after their cause was executed and that tracing the exact cause can be difficult due to multiple software layers and component dependencies. They cite the complexity of scalable software as a cause of these issues. In order to accommodate management of errors appropriately, they deploy an error ranking system according to severity. This enables selection of an appropriate action such as to ignore, restart or revert. The software takes advantage of virtualisation within cloud environments to accomplish this. The implementation details are scarce but it is important to note that this method does take into account generalised application faults so would detect malicious and non-malicious faults although it is not explicitly stated as a goal of the software.

VI is a common solution to issues such as intrusion due to the ability to revert the system in question to a clean state and migrate them to other nodes. Lombardi et. al. present a system for enhanced resilience of Windows VMs,

Table 2. Storage Resilience Techniques

Method	Model	Components	Disciplines	
Cost-based disaster recovery planning, replication [53]	IaaS	VS	A, DT	Disaster-recovery not always beneficial to service delivery
Code-based recovery [24]	IaaS	VS	A, FT	Cost resulting from groups of nodes
Rapid data evacuation (RDE) prior to disaster event using a heuristic to select least delay paths [126]	IaaS/	VS	DT	A-pirori disaster knowledge, bandwidth requirements
Network overlay on diverse storage across multi DCs [43]	PaaS	VS, VN, RM, SO	FT	High replication cost
Replication covering correlated and non-correlated failures [68]	IaaS	VS	FT	High replication cost
Storage across diverse cloud-of- clouds [19]	IaaS	VS, RM	FT, SEC	High replication cost
Provide security to the client side of cloud storage services [75]	Client	CD	SEC	Client side only
A survey and hybrid technique between erasure coding and replication [83]	IaaS	VS, RM	FT	High replication cost
Efficient replication for unknown query rate [94]	IaaS	VS, RM	FT	High replication cost
Homomorphic token and erasure data [123]	IaaS	VS	FT, I	Data loss with loss of key
Component based workflow RM from client to cloud [128]	PaaS	VS, RM	FT, SEC	Complex modelling process
Multi-dc backup disaster routing, shortest window for disaster RB [129]	IaaS	DC; VS, RM	DT	A-priori disaster knowledge, High replication cost

VS=Virtual Storage, VN=Virtual Networking, RM=Resource Management, SO=Service Orchestration, CD=Client Devices, DC=Data Centres, A=Availability, DT=Disruption Tolerance, FT=Fault Tolerance, SEC=Security, I=Integrity

recovering from malicious and non-malicious faults [69]. The solution employs VM introspection and anomaly-based integrity verification to detect intrusions or errors arising. A reactive solution using the VI is then employed. However, it could be argued that repetition of malicious actions could allow a malicious user to plot the state changes taken and thus understand the intent of the IDS. Likewise during the evaluation of the system, the authors claim that despite a noticeable increase in resource use when implementing the introspection system, the attacker would not be able to detect the use of the system. Again, this is not a sufficient means of disguising the use of the system as comparative analysis would result in wide variations in performance against a system that did not employ the introspection.

Reizer and Kapitza also employ VI for a proactive recovery (periodic node refreshing) system [96], primarily for intrusion relation failures. They explain that proactive recovery reduces support for recovery from genuine faults and also decreases system availability time. In their solution a domain is replicated across numerous, isolated guest domains where all network activity is proxied via a remote server. Diversity between the guest and primary domains ensures attacks will not affect the replica domains. This will hold true as long as the service is developed to be deterministic. Although unable to protect against a physical fault, this system will prevent against malicious and non-malicious faults. However there are large cost and time implications due to the additional resources required.

Jhawar and Vincenzo propose a similar method named Remus [54]. It uses replication leveraging VI to provide a high degree of fault tolerance. The system periodically snapshots the host's state, storing a backup in memory. This ensures prompt availability. It is possible that anomalous data would cause both systems to crash, whereas the diversity in the previous work clearly protects against this. VM replication techniques are found often such as in [101] and [34]. Egwutuoha et. al [36] and Tchana et. al. [115] examine resilience at the process-level. They provide replication and check-pointing of individual processes in order to recover from faults. These techniques could be considered more resource optimal than checkpointing an entire VM whilst also more relevant to container-based architectures. However the security implications of changing the integrity of a running VM are concerning.

Binun et.al focus upon the resilience of the VMM [21]. They present a novel self-stabilising hypervisor for increased robustness against malicious faults. A Stability Manager examines the VMM and its VMs for any misbehaviour, resetting the VM, software or physical machine when a subversion is detected. Whilst the system does provide an adequate method of resisting intrusions it is uncertain whether it is possible to recover from a persistent threat. Without adequate constraints, it might be easy for an attacker to perform a DoS upon the machine through corruption, requiring a constant reboot. There are further performance issues, as with the integrity check, requiring the entire system to freeze.

Kanter and Taylor present a hypervisor which uses compiler and run-time techniques to increase diversity within an application, making attacks more costly [57]. Their combination of techniques prevents all memory addresses within the application being known to an attacker a-priori, however it does require the application's source code. This extreme case of diversity across the cloud infrastructure enables high resilience. The method is used in the deployment of an OS named *Bear* consisting of a minimal kernel and a VMM, where the kernel and all other components, including device drivers, are periodically refreshed with new, diverse replacements. A point of note is that it does not attempt to detect intrusions, and operates without any further information. This is a useful characteristic which mitigates detection-related issues. They mention that performance is degraded due to the additional processes, the compile time decrease at 5% typically and sometimes up to 16%. However the benefits of having a different set of binaries for every individual host in the cloud outweighs the performance hits.

Xu and Huang focus upon VMM resilience. [127]. Where execution of each VMM is replicated across a another. This provides resilience against hardware layer faults. Essentially providing redundancy for hardware, the system is successful with minimal overhead. However, the authors mention that the replication mechanism is not self-protecting

2.2.3 Virtual Networking. Providing resilience within the networking layers of a cloud architecture is the focus of a number of works seen in literature. Resilience within networks may often be considered in terms of its survivability, distributed information systems being the focus of the term survivability [126]. As networking operates on a variety of different layers, the resilience may again differ depending on the layer in question. Clouds may be distributed across multiple geo-locations and therefore require resilience on the physical layer. They also employ considerable quantities of virtualisation and with the advent of SDN networking in the upper application layers can become complex. Table 4 summarises the following literature.

Bui et. al. investigate two methods for ensuring resilience in virtual networks [22]. They consider network resilience from the perspective of both the PN providers and VN operators and in particular, the mapping between these two layers. Their resilience models involve handover to another DC during primary failure, with the resilience techniques involving source routing to the secondary DC. Two tests were conducted where data centres where uniformly distributed and locally paired. The results showed that the VN routing performed slightly better than the PN routing during uniformed location distribution whilst during the paired locations there was very little difference.

Table 3. Instance and VMM Resilience Techniques

Method	Model	Components	Disciplines	Limitations
Snapshots with error ranking[85]	IaaS	VI	FT	High replication cost and storage, state-management
Snapshots with VM introspection and anomaly detection[69]	IaaS	VI	FT	High replication cost and storage, state-management, windows only
Proactive recovery [96]	IaaS	VI	FT	High replication cost and storage, state-management, no physical fault protection
Periodic snapshots and recovery [54]	IaaS	VI	FT	High replication cost and storage, state-management
Process level replication, checkpointing [36]	PaaS/IaaS	VI, RM	FT	Medium replication cost and storage, state-management, VM integrity
Process checkpointing [115]	PaaS/IaaS	VI, RM	FT	Medium replication cost and storage, state-management, VM integrity
VI replication [101]	IaaS	VI	FT	High replication cost and storage, state-management
VI replication [34]	IaaS	VI, RM	FT	High replication cost and storage, state-management
Replication, checkpointing of HV using introspection [21]	IaaS	PI, VI	RB, FT	Liable for anomaly subversion
Compiler and runtime software diversity [57]	IaaS	PI, VI, RM	SRV	Requires application source-code
VMM replication / mirrors with minimal overhead [127]	IaaS	PI, VI, RM	FT	High Replication cost

VI=Virtual Infrastructure, RM=Resource Management, PI=Physical Infrastructure, FT=Fault Tolerance, RB=Robustness, SRV=Survivability

Barla et. al. consider the performance of network resilience methods [13] with the motivation of SLAs guaranteeing various quality of IT services but not end-to-end communications. VN is cited as a solution to this issue. The study simulates two scenarios, in the first the VN is responsible for the resilience and in the second, the PN operator is responsible for the resilience. The results indicated that the VN always outperformed the PN. Complexity of the design is mentioned for consideration. The authors expand this further in [14] and [15] where they providing models for developing the VNs. These were shown to outperform previous approaches by finding a resilient solution in every case, drastically reducing communication delays.Barla et. al. next examine resilience in VNs using redundant back-up links (refereed to as shared protection)[12], similar to the approach employed in [71] for physical networks. They use redundant resources shared amongst multiple VNs, accomplished through appropriate information exchange between the PN operator and VN operator. As before, the VNs outperformed the PNs with the addition of cost saving benefits through optimised set-up. This work highlights the effectiveness of providing high-level resilience.

Harter et. al. [47] confirms the benefits of shared protection mechanisms, with cost savings of 10-20%. This model includes heuristics to make the algorithm highly scalable.

Bui et. al. consider the problem of mapping VNs to PNs under varying time constraints [23]. Their solution enables the selection of appropriate PN and (consequently) DC resources for resilient re-routing of networks under varying time constraints. As an improvement upon schemes which allow bandwidth sharing through various backup links, this system allows backup links to be reprogrammed as needs change over time. Their results show that the benefits of reconfiguration are only applicable if the standard working paths are also reconfigured. Also they are only applicable if the majority of traffic "does not have Quality of Service (QoS) requirements that prohibit path reconfiguration", which could be an issue when considering the variety of use cases for cloud traffic.

Secci and Murugesan provide discussion regarding the current cloud network architectures [105]. They stress the need for resilient clouds as without resilience their services are sub-optimal or even useless, due to the service-oriented nature. They explain that conventional cloud RM is considered "dumb" due to over provisioning of resources and in particular, inefficient methods of bandwidth utilisation. They argue that this has caused high centralisation in geo-distributed clouds, which contributes to high risk of failure and therefore, low resilience. As services must be distributed across multiple cloud services in order for them to be resilient, this conventional bandwidth utilisation is at odds with this requirement and therefore the authors suggest that further decentralisation is necessary. To provide further decentralisation, appropriate overlay networks must be employed to ensure network paths diversity and DC end points in order to ensure the necessary resilience. The authors conclude by noting that resilience isn't just a requirement of highly dependable services but necessary in order to fulfil the fundamental cloud SLA In order to provide this, the current architecture must change.

Harter et. al. provide a comprehensive discussion regarding the comparing the resilience of different layers and a novel consideration of the business-oriented responsibilities of each cloud delivery scenario [15]. They provide a method, determined through simulation, to determine the most favourable layer to provide resilience depending on the use-case. They conduct a similar study, investigating which layer is the most effective in terms of cost and fault tolerance to provide the resilience [48]. This time considering PN, VN and overlay networks. The give a framework for ease of selection.

Osanaiye et. al. focus upon one particular attack type/ resilience problem [87]. They provide a survey and framework of DDoS mitigation techniques in the cloud. Whilst effective as a traffic tolerant technique, the lack of coverage for diverse failure types leaves this form of work behind the others.

2.3 Cloud Management

This section reviews literature applying techniques which are operated upon via the cloud-management layer. Tables 5 and 6 provide a summary.

As redundancy is a key component of resilience, task placement can influence its efficacy. Cartlidge and Sriram present an analysis of the effect of different scheduling algorithms on resiliency [25]. This should be considered as resiliency of IaaS. They evaluated random, packed (FILO) and clustered VM allocation, showing that packed was the least resilient, which is obvious when considering single point of failure. Additionally, their results illustrated a clear link between hardware redundancy and resiliency, although the pack scheduling algorithm was not always consistent. When adjusting the DC architecture, they concluded that the pack was sensitive to infrastructure types. Although this work is intuitive, it is interesting when understanding if resilience designed at high layers can overcome the shortcomings of poor resiliency at lower layers.

Gao et. al present an energy aware scheduling algorithm for cloud resilience [40]. Their framework allows reliability when faced with soft-errors, often a consequence of varying voltage levels. It has a performance increase of up to 50%

Table 4. Virtual Networking Resilience Techniques

Method	Model	Components	Disciplines	Limitations
Mapping between virtual and physical, data centre hand over [22]	Phy/IaaS	PI, VI	SRV	Uniform DC distribution with minimal performance increase
Optimal overlay networks [13]	Phy/Iaas/Paas	VN vs PN	SRV, FT	High VNet design complexity
Leverage VN for layer selection[14]	Phy/Iaas/Paas	VN vs PN	SRV, FT	Complex high-level management
A framework for layer selection [15]	Phy/Iaas/Paas	VN vs PN	SRV, FT	Complex high-level management
Shared protection with back up links [12]	Phy/Iaas/	VN, PN	SRV	Lack of compartmentalisation
Scalable heuristic-driven shared protection [47]	Phy/Iaas/	VN, PN	SRV	Lack of compartmentalisation
Shared protection with rerouting [23]	Phy/Iaas/	VN, PN	SRV	Lack of compartmentalisation
Overlay networks creating path diversity [105]	IaaS/PaaS	VN	SRV	High Complexity and centralised management
Selection of physical, virtual or hybrid networking [15]	Phy/IaaS	VN, PN	R, FT	High-level simulation
Cost vs RB across different layers [48]	Phy+	VN, PN	R	High-level and constrained simulation
Cloud DDoS survey and mitigation framework [87]	Phy+	VN, RM	TT	Restricted failure types

PI=Physical Infrastructure, VI=Virtual Infrastructure, VN=Virtual Networking, PN=Physical Networking, RM=Resource Management, SRV=Survivability, FT=Fault Tolerance, R=Resilience, TT=Traffic Tolerance, RB=Robustness

achieved through a hybrid method. Firstly conducting an assessment of static reliability requirements which then leads into dynamic analysis which can occur at run-time. Their implemented system also considers financial data, which is an often overlooked, yet principal component of service-oriented cloud systems.

Liang and Lee consider resiliency when developing PaaS clouds [65]. They take a robustness approach which allows varied and unexpected program input. Their work appears to be concerned with reliability through analysing sub-component effects upon the application. A SO approach is applied to minimise failures through accurate selection or replacement of individual components, maintaining a low failure rate.

Verissimo et. al present a novel paradigm, *cloud-of-clouds*, with their system : TCloud [120]. They argue that DC distribution is not enough to provide resilience of applications within a cloud, as the security aspects of federated clouds are not addressed. To provide resilience, the authors argued that a user must be able to combine clouds from multiple providers providing high diversity. Additionally open architectures are necessary to prevent proprietary vendor lock in and security features from the lower layers up. Their system accommodates these requirements through providing multiple solutions in order to "build layers of progressively more trusted components and middleware systems", which allows layers on the top layer to be trusted due to trusted lower layers. A flaw in this system is that trust in a lower layer cannot always be guaranteed. The authors suggest an intrusion detection system as an example of lower layer security, which are regularly circumvented to allow malicious traffic obfuscation. The system includes modularised components to provide middleware to enable the paradigm, providing a secure and resilient PaaS.

Sharma et. al present the development of a resilient PaaS leveraging state management, known as ReLo. [106]. Resilient state-management enables a session to persist during application down time. Within their system. agents reside within an application. If the agent goes down, a handler agent redirects the session to another application. Essentially the system employs redundancy via a middleware management solution. The single point of failure with the handler agent is a questionable choice. The authors mention that memory constraints and router time outs having a negative effect upon the resilience.

Scholler et. al. describe their method of deploying vNFC (chains of virtual instances, providing networking services) [103] resiliently using their Tenant Infrastructure Management Software (TIMS) upon OpenStack. Different components within the network service have different resiliency requirements, (e.g. scalability, redundancy). The service describes the requirement and TIMS manages it appropriately. Both resource and network requirements are given through Application Layer Traffic Optimisation (e.g. maximum delay between two components. The system is dependent upon OpenStack's availability zone feature, which groups pools resources which allows critical components of one service to be grouped in different failure locations. The authors identified a number Openstack shortcomings such as low resource information within the pools.

Klein et. al. discuss the brownout programming paradigm [60], proposed in order to provide enhanced robustness within cloud services. It attempts to mitigate the requirement to provision large amounts of replicated instances during traffic increase. This should prevent service run-time failures such as flash crowds. A brownout program will downgrade a user experience, such as with enhanced features, in order to prevent excessive use of the system. They extend this by combining with load balancing as the combination currently creates conflict [61]. They propose two novel algorithms and a production ready load balancer. Their results indicate strong performance compared to alternative solutions.

Torres and Holvoet examine service composition architectures [116]. Their decentralised system relies upon two distinct agents, the first monitors the network for appropriate, available subtasks to compose a service with. The second evaluates available resources within the system. These two agents continuously and dynamically assess the current status of the service enabling a rapid response to faults. Each agent delegates work to lower-level agents, which would appear to be biologically inspired by ant processes. Empirical evaluation indicated that performance was better than the common, reactive approach, with lower composition times of between 4 % and 25 %. The authors note that the system suffers from high communication costs.

Minzhe and Prabir investigate diverse replica software in [45] where the configuration of the OS in which the service is built upon is varied across the service. The authors present a game theoretical approach to the problem.

Mihailescu et. al. consider the mapping of components of a service, to hardware resources [77]. Their algorithm is more optimal than global shuffling algorithms, and will eventually converge on a stable global configuration, as long as application requirements remain consistent. The system improves resilience from hardware faults and network errors through understanding component inter-dependencies. It models them as graphs where a divisions equates to VM migrations. The system is dynamic, allowing an end-user to select the required resiliency. Further work might consider the effect of cost upon this feature.

Carvalho et. al. take a biologically inspired approach to cloud [26]. They employ a multi-layered method with a focus on distributed service management. The authors focus on mission continuity and survivability during attacks. The work focuses upon the application of bio-inspired methods of self-organisation and self-management, as well as distributed coordination, to the service discovery and orchestration processes in the cloud. The system layers consist of firstly detecting the damage through distributed sensors, optimised resource management and then response/immunisation to the threat.

Louati et. al focus upon state-less applications, which are easier to provide resilience to than state-full [70]. Their solution uses checkpointing and application restart combined with a back-end built upon Distributed Hash Tables (DHT) for resilient decentralised storage. Nicolae and Cappello also considers checkpointing/restart of applications to mitigate failures [86]. This time for High Performance Computing (HPC) applications using efficient Virtual Disk Image (VDI) snapshots.

Villarreal-Vasquez et. al. argue that the typical replication techniques employed for cloud resilience increase the attack surface of an application and thus are detrimental [122]. They propose a solution which uses Moving Target Defence (MTD). Migrating instances once an anomaly has been detected and provide self-reconfiguration to return to base-line state.

Frîncu applie Genetic Algorithms (GA) to component scheduling optimisation [39]. They consider high availability web applications within the constraint of cost. They propose two distinct algorithms: the first optimises the maximum number of components upon each node within the cost. The second is sub-optimal, finding the minimum required so that the application is still available given that all but one node fails. Antony et. al. also investigate scheduling to optimise resource usage, this time for bandwidth consumption [8]. They provide a heuristic which optimises data locality to reduce the job completion time and provider fault tolerance to the Balance Reduce (BAR) algorithm.

Liao and Cheng propose a resilient scheduling method which involves servers retrieving batches of jobs and then processing them according to a specific weighting to mitigate the effect of a malicious fault [66]. Zheng et. al. take a similar approach [132]. They rank a component's value and orchestrate a service so that a fault will with have a reduced or no impact upon operation. Similarly, Ferdousi et. al. take a similar approach [38]. Again applying ranking, with a greater focus upon the placement of content as opposed to the components themselves.

Al-Ayyoub et. al. provides a framework which leverages mixed integer linear programming to consider multiple objectives to optimise cost-effective resilience across all levels of cloud infrastructure [5] .

Imran et. al. developed A middleware which uses watchdogs, checkpointing and journaling [51]. It is used to create, backup and store replicas of application to provide fault tolerance. Whereas Zhao et.al provide a replica oriented middleware yet with comparatively considerable resource optimisation[131]. whilst only for replications of network protocols (sockets, web protocols etc.) it is an interesting approach. Although the integrity and confidentially of the application is in question.

2.4 Decentralised Cloud Resilience

While the previous sections discussed centralised cloud architectures residing in data centres, this section presents decentralised cloud models such as fog and edge computing. Table 7 summarises this literature.

Due to the constrained nature of IoE devices, data processing, storage and representation must be provided by a third party platform, typically the cloud. However, the high latency, non-deterministic wireless mediums and high volume of data make this relationship difficult. Fog computing is the medium in which psuedo-cloud services, mostly temporary data processing, are provided closer to the edge of the network. Processing data in this form has a greater requirement for resilience due to device mobility, open wireless mediums, constrained device resources, heterogeneous device types, cyber-physical systems and hostile environmental conditions.

Service orchestration (SO) is an important process to conduct securely in fog computing. In order to optimise constrained device resources, only the minimum amount of nodes necessary will be provisioned for an end-user. This requires service requirements to be broadcast for a network which provides a number of security issues, particularly confidentiality. Viejo and Sánchez use Ciphertext-Policy Attribute-Based Encryption (CP-ABE), whereby nodes will have

Table 5. Cloud Management Resilience Techniques

Method	Model	Components	Disciplines	Limitations
Comparison of task scheduling algorithms [25]	IaaS	RM, SCH	RB	VM only
Energy aware scheduling for software FT, [40].	IaaS	SCH	FT, DT	Power-related faults only, resource intensive modelling
Monitor and optimal selection of hosts [65]	PaaS	SO	RB	In-depth application analysis
TCloud, a modular middleware and layered multi-cloud solution [120]	PaaS/IaaS	DC, RM	SEC, RB	Strong interoperability issues, high cost, contentious reliance upon trust
Resilient state management using a remote handler [106]	PaaS	VI	FT	Single point of failure
Chains of virtual network instances [103]	IaaS?	VN, VI, SO	SRV	Centralised management
Brownout experience downgrade [60]	PaaS/IaaS	VI,RM	RB, TT	Single replica only, experience reduction
Brownout experience downgrade with LB [61]	PaaS/IaaS	VI,RM, LB	RB, TT	Single replica only, experience reduction
SO using multi-agent monitoring and feedback [116]	PaaS/IaaS	VI,RM, SO,	FT	High communication cost
SO using diverse OS configurations [45]	PaaS/IaaS	VI, SO,	FT / SRV	Conceptual only
Service to hardware dependency modelling, migration and FT monitoring [77]	PaaS/IaaS	VI, VN, RM	FT / SRV	Static application requirements
SO using self-organisation and self-management [26]	PaaS/IaaS	RM, VI, SO	SRV	High complexity and resource intensive monitoring

RM=Resource Management, SCH=Scheduling, SO=Service Orchestration, DC=Data centres, VI=Virtual Infrastructure, VN=Virtual Networking, LB=Load Balancing,RB=Robustness, FT=Fault Tolerance, DT=Disruption Tolerance, SEC=Security, SRV=Survivability, TT=Traffic Tolerance,

keys corresponding only to the attributes they are allowed to process [121]. Their network is structured hierarchically so that nodes pass messages to those it can control further down the tree. The nodes will require a generalised key. For example, a message containing "temperature" will also need a "weather" key to process it. These messages form policies such as "temperature, zone 1" which are then encrypted separately and transmitted. If any messages can be decrypted by a node it means that further nodes in the hierarchy can also be decrypted so the service discovery can continue. Once the service has been orchestrated between the required nodes, the client and nodes exchange keys to communicate securely. Chejerla et. al. instead chose to develop a scheduling algorithm that uses a game-theoretic and Bayesian approach to mitigates against attack in real time, for Cyber Physical Systems (CPS) [28].

Rios et. al explain that modelling fog networks in a hierarchical manner, with a singular provider, is oversimplified and detrimental to it's security [97]. They should instead be considered as a federated architecture with numerous service providers within different trust domains. The authors propose an architecture (SMOG) to provide resilience in fog networks. It consists of number of baseline characteristics such as *secure interconnection*, *authentication and authorisation*, *protection of virtualised environments* and *situational awareness*. They list enhanced characteristics as *trust*

Table 6. Cloud Management Resilience Techniques II

Method	Model	Components	Disciplines	Limitations
Checkpoint-restart of stateless applications upon decentralised DHT [70]	IaaS	VR, SO, RM	FT, A	Stateless only
Self-reconfiguring moving target defence [122]	IaaS	VI, RM,	FT	High Complexity and replication cost
Optimised checkpoint-restart for HPC using VI [86]	IaaS	VR, RM	FT	High Replication cost
GA based SCH, homogeneous spread of components across all nodes [39]	IaaS	VI, SCH	A, FT	High initial resource cost and web app only
Balance Reduce for data locality and job time reduction [8]	IaaS	VI, SCH	FT	Constrained use-cases
Batches of jobs weighted to prevent malicious faults [66]	IaaS	VI, SCH	FT, SEC	Pull and then process can cause synchronisation errors
Component quality ranking and service construct to reduce faults [132]	IaaS	VI, SO,	FT	High complexity
Risk minimisation using content ranking and placement [38]	IaaS	VR, PR, DC	DT	High complexity
Mixed integer-linear programming [5]	IaaS	VR, PR, SCH	A	Conceptual online
Replica-oriented middleware [51]	IaaS/PaaS	VI, RM	FT	Complexity and resource cost
Multi-component middleware to enforce policies, determined by anomaly detection [107]	Phy/IaaS	RM, SCH,PM	FT, SEC	Complexity and resource cost
Networking protocol (Sockets, web etc) replication [131]	IaaS/PaaS	VI, VN, RM	FT	High complexity and resource cost, Web sockets only

VR=Virtual Resources, SO=Service Orchestration, RM=Resource Management, FT=Fault Tolerance, A=Availability, VI=Virtual Infrastructure, SCH=Scheduling, PR=Physical Resources, DC=Data centres, SCH=Scheduling, RM=Resource Management, Physical Networking, FT=Fault Tolerance, SEC=Security, VN=Virtual Networking

services, *distributed decision making*, *privacy capabilities* and *digital evidence management*. They explain that these base line requirements are largely missing from literature and are necessary to ensure a secure and resilience fog.

Edge nodes are without doubt a point of failure in any decentralised cloud network. Le et. al. give a solution to partial failures in MEC (e.g. connectivity loss) between the edge nodes [63]. Their architecture is again hierarchical, with mobile nodes storing local back up data dispersed amongst them. If partial failure with the edge nodes occurs, the devices switch to a P2P model, processing data collaboratively. This is an alternative mobile computing model and the results show good time reduction performance when the task is disrupted across the nodes. However the power consumption is likely to be highly variable according to the difference between nodes and therefore it the suitability will not be universal.

Modarresi and Sterbenz consdier Fog Computing as a solution for resilient IoT/edge computing in [80]. They argue that the uncertainty surrounding resource, link and bandwidth availability ensures that typical edge computing is not resilient for IoT processing. For example, too many clients can overload resources and thus cause a denial-of-service. They argue that the introduction of fog nodes between the edge and the cloud creates greater autonomy within the network. If a connection is lost between the edge and the cloud, the fog maintains this network and increases the *survivability* of the ecosystem. Further to this, they suggest that the diversity of standards, protocols and network links, which caus fog computing to be quite complex, is actually beneficial to its resilience due to the increase in variety. They

also indicate that through fog reducing traffic further in the core and distribution network, its implementation provides *traffic tolerance*. Finally, *disruption tolerance* is enhanced through a reduction in latency permitting applications to be processed quicker and thus any disruption has less impact. The authors support these statements with numerous simulations inclusive of the fog environment.

Hussein et. al. provide a mobile edge computing solution which applies Software Defined Networking(SDN) to 5G provide resilient processing to Vehicle Area Networks (VANETS) [50]. Safety concerns are paramount in vehicles and as such so is the resilience of VANETs. Their proposed solutions provides enhanced security through an additional security layer using SDN. As opposed to a traditional centralised SDN approach or a traditional distributed VANET approach, they present a hybrid method. A centralised 5G base station is used to manage SDN security functions distributed across a number of roadside controllers. This approach illustrates a strong example of custom networking hierarchy technologies being supported at the edge for specific use-cases and resilience requirements.

Modarresi deploys SDN again in tandem with fog for resilience in [79]. This time fog nodes are used to detect anomalies in network traffic and notify the SDN controller. This can make security-focused decisions about what traffic to drop or restrict. A strongly illustrative example of the application of fog for greater network resilience although does not help to strengthen resilience of the fog nodes themselves.

Bensen et. al takes a middleware approach to provide continued operation of critical events from IoT devices when their connection to the cloud fails [18]. Their system contains two components, the first periodically probes different paths to the cloud, detecting possible faults or failures. The second provides multicast message dissemination according to information received from the first component. They again use SDN to provide this information and use it create "resilient overlays". This middleware approach enables varied support for IoT devices as the middleware works seamlessly.

Kahla et. al. provide a solution to low trust in IoT environments [56]. They leverage moving target defence to migrate targeted or subverted virtual instances to another host fog machine. It is not clear how this would prevent a number of different attacks or heal the instance once it had migrated although the autonomic aspect of integrity verification is commendable.

Eisele et. al. state that resilience is necessary to consider in edge environments due to both resource and network uncertainty [37]. Whilst security is important due to the resource constrained nature of edge devices preventing virtualisation providing adequate isolation. They propose a novel programming paradigm: RIAPS (Resilient Information Architecture Platform for Smart Grid) which provides a platform for distributed applications to be deployed resiliently. The platform provides a diverse number of different services and managers (such as for security, persistence, fault management etc.). Whilst the platform appears to be complex and thus has an increased attack surface, given the number of required components, it illustrates the notion of an underlying platform providing resilience to higher levels.

Arval et. al. use bayesian belief networks to mine dependencies between replicated edge nodes. Their solution uses past server performance from logs and temporal dependencies to highlight the probability of when failures may occur concurrently. Although their current solution is theoretical it shows strong optimisation through replica reduction [9].

Neto et. al. tackles a somewhat different resilience problem [10] . Where Fog enabled service does not suffer a fault but an outage related to the CSPs SLA. They focus on Amazon's Spot Instances which are transient servers acquired by the user when the maximum they wish to pay (bid) is greater than the value of the instance. Due to the nature of this acquisition the continued operation of these servers cannot be guaranteed. Therefore in this fog platform the failure results from the unavailable CSP back-end. To mitigate this, they propose an agent-based case-based-reasoning solution which aims to predict the survival time of an instance. This enables checkpoints to be made in order to resume the work

in case of application fault. Their solution could be modified for application processing closer to the edge, although the resource requirements for checkpoints must be considered.

Ozeer et. al have a similar focus on recording and reverting to application states [88]. They take an uncoordinated approach, recording application events with a corresponding recovery timer. Expiration indicates lack of synchronisation with the physical world and cant thus be ignored. Event details are logged in a global and failure-free storage system to permit recovery to any node from a central location. This centralised storage suffers from central point of failure. The authors present a competent yet complex solution consisting to enable system fault tolerance. The question of how failures are to be handled in the system handling the failures is still open.

Khalifa et. al. move away from a traditional cloud architecture, improving the resilience of Hybrid Mobile Clouds [59]. Mobile clouds require greater resilience than a static system due to the dynamic network characteristics. The proposed architecture is interesting due to its flexibility in running on a diverse devices, essentially ignoring the underlying hardware. The resilience requirements are aided through a resource prediction mechanism and an early failure detection mechanism to facilitate handover of vital services. The system proves successful, although performance is still dependent upon the quantity of fixed nodes within the cloud, making the system not purely mobile. However, overall it exhibits a good example of how cloud systems can be built upon non-deterministic environments.

2.5 Alternative Architectures

Some work will choose to encourage a conventionally different cloud architecture in order to provide increased resilience. Table 8 summarises this literature.

An alternative to the infrastructure layer, Suciu et. al. present SlapOS [113]. They choose to provide a purely distributed cloud architecture where single point of failure is remedied through distributed the cloud resources over multiple PCs within homes, as opposed to within DC. Whilst this might bring forth bandwidth, capacity, and latency issues; the benefits of reducing single point of failure are considerable. Particular for safety-critical events such as during disasters.

Courteaud et. al consider further resilience of SlapOS [31]. They refer to the concept of community cloud, whereby the cloud is collaboratively built from personal devices. The main current issues are summarised as: 1) Migrating from commodity cloud to resilient, secure and dependable clouds 2) Promoting diverse and open ecosystems 3) Building a coherent, modular and reusable architecture. They also consider the leader selection problem (the process of selecting the next master node after loss of the current). Further issues include: implementation an accurate failure detection methods, and methods of replicating the master database prior to handover to another master node. The authors not that conventional delivery models of (IaaS/PaaS/SaaS) become obsolete. Finally, the authors explain that an implementation of hierarchical masters (such as with DNS) will be implemented for increased resilience. Whilst the architecture and delivery model is certainly interesting there are issues directly relating to resilience concerning master node hierarchies which undoubtedly cause problems. A decentralised system such as this is not as resilient as one which is purely distributed.

Garlick also considers community cloud based resilience [41]. They promote the model as an enhancer for organisational resilience. As with SlapOS, the author highlights the ownership and location of current cloud models being unsuitable for providing resilience. The authors note that for natural disasters centralised disaster recovery is too late and excessive. They argue that disaster recovery must be conducted at the community level. The breakdown of communication networks is cited as a key issue, where the more effective communication was developed by the decentralised communities. The author explains that community cloud models enable the benefits of public cloud

Table 7. Decentralised Cloud Resilience Techniques

Method	Model	Components	Disciplines	Limitations
SO using Attribute Based Encryption [121]	PaaS/SaaS	EDGE, RM, SO	C, I	Resource intensive cryptography and hierarchichal network structure
Multi-component federated fog architecture [97]	IaaS/Paas	RM, EDGE, CONST, TRAN	SEC	Conceptual model for resilience, high complexity
Hierarchical data replication and service downgrade using p2p networking [63]	Paas/SaaS	CONS, EDGE, CONST	A, DT	Questionable energy consumption per task
Fog computing for resilience [80]	IaaS	FOG	SRV, DT, TT	High Resource cost and governance issues
Decentralised SDN for 5G VANETS [50]	IaaS	CONS, IOT, TRAN, EDGE	DT, SRV	Centralised management
Fog-enabled anomaly detection for SDN [79]	IaaS	TRAN	SRV, SEC	Centralised management
SDN middleware for critical events using rerouting andbackup links [18]	PaaS/SaaS	EDGE, TRAN	SRV, FT,	Single point of failure
Anomaly detection and moving target defence [56]	IaaS/PaaS	VI, CONST, RM	I, FT	Questionable resource cost
Watchdog-based multi-layer programming architecture [37]	PaaS	VI, VN, RM	FT	High Complexity
Dependency mining for replica prediction and optimisation [9]	IaaS/PaaS	VI, RM	FT	Theoretical
Agent-based spot-instance survival reasoning [10]	IaaS/PaaS	VI, RM	DT	Specific to one cloud provider
Uncoordinated application checkpointing and replication [88]	IaaS/PaaS	VI, RM	FT, SRV	Single point of failure
Resource-predicting Hybrid mobile cloud [59]	IaaS	PH	DT	Dependent upon fixed nodes
Game theoretic with bayesian approach to SCH during real time attack [28]	IaaS	SCH, VI, EDGE	A, SEC	Attack-specific

C=Confidentiality, I=Integrity, SEC=Security, A=Availability, DT=Disruption Tolerance, SRV=Survivability, TT=Traffic Tolerance, RM=Resource Management, SO=Service Orchestration, CONS=Constrained Devices, TRAN=Transportation, EDGE=Edge Devices, VI=Virtual Infrastructure, FOG=Fog Computing, IoT=Internet of Things devices, VN=Virtual Networking, PH=Physical Hardware, SCH=Scheduling, EDGE=Edge Devices, FT=Fault Tolerance, DT=Disruption Tolerance, A=Availability, SEC=Security

offerings with greater control. Issues surrounding community clouds, such as malicious users, are said to be mitigated through user vetting, a process which may not always be practical or effective. Sathiaseelan et. al. presents a similar discussion is [100]. Where similar they highlight that commodity hardware based community clouds have considerable advantages over resilience due to the highly decentralised nature.

Sterbenz and Kulkarni preset DefCloud [111] which attempts to provide greater resilience through increasing diversity and redundancy within all layers of the cloud architecture. Highly flexible, it allows resilience to be adjusted in a "service-aware manner". This might be argued to be similar in concept to the usability vs security trade-off. Such a feature is likely necessary for a cloud platform which accommodates a wide spectrum of use-cases. The authors argue that the first point in designing the infrastructure, is the removal of monoculture as it enables malware and attacks

to propagate effectively through only needing to attack one type of hardware architecture or software application. This concept is then applied to all layers of the cloud. Firstly is the *infrastructure layer diversity*. This is encompasses *data-centre diversity* and *cloud diversity*. They consider DC diversity considered sub-trees of features where similar trees are not selected in tandem in order to maximise diversity. For example similar trees will utilise the same network vendor hardware or operating systems. Whilst diversity provides resilience against security related failures, it does not protect against failures from direct physical DC attacks, e.g. natural disasters or military attacks (such as an EMP). To mitigate these issues, the architecture applies cloud diversity through distributing the cloud over multiple geo-locations, using varying ISPs. After the infrastructure layer, the DefCloud then assures resilience through *Process-level Program diversity*, where diversity focuses upon distribution via space and time. Spatial diversity is concerned with distribution of different software versions. Temporal diversity is concerned with varying application configurations over time. Application diversity ensures binaries are diverse e.g. an attack on one application binary will not apply to another. Whilst this has consequences on the current state of 0-day exploits, it complicates the software development process. DefCloud undoubtedly covers resilience in the cloud through adaptions of the conventional architecture, the system lacks real implementation or simulation and thus its resilience is yet to be determined. For one, the complexity of the system is clearly greater which increases the attack surface.

Keromytis uses similar diversity in their MEERKATS system [58]. It is a fully novel architecture for a security mission critical cloud. The system constantly evolves across all aspects, reducing monoculture and increasing diversity. One component of the system, DREME [16], is concerned with execution diversity of replicas and provides a framework for I/O redirection.

IBM present a somewhat novel architecture name SCE+ [99] which is built from the ground up to be highly resilient. The authors make the distinction between typical cloud architectures employed by Amazon and Google by explaining that they are constructed from "redundant, inexpensive, expendable building blocks" whereas the IBM SCE+ employs "high-end building blocks with significant internal redundancy and an established track record of very high MTBF for every element." It would appear that the contrast is in SCE+ employing mature and extremely resilient fewer components with conventional architectures employing many less mature components and relying upon replication/redundancy. The architecture applies resiliency to differing cloud layers. The physical layer is designed so as to avoid single point of failure through division of resources and replication in separate geo-locations with a backup dark-fibre link. Software resilience is then considered from multiple aspects. Components are deployed in redundant pairs and constant "health-checks" are in place to monitor correct functioning. In addition, redundancy of data and regular backups ensures resiliency within the data layer. The authors explain that standardisation of hardware within the system components aids the resilience, however this is contentious, as diversity within hardware is surely a necessity for resiliency. They cite virtualisation as an enabling factor of the resilience, however this is typically a component of cloud infrastructures anyway and therefore offers the environment no additional advantage. Overall the architecture offers a variety of additional components for resilience although some are questionable such as the physical distance between components as well as the added complexity within the system.

Hariri et. al. present an architecture based on biologically inspired processes which allows tunable redundancy at multiple cloud levels, known as BioRAC [46]. One layer of the architecture involves division of components into "cells" which allows dynamic real-time configuration and combine together to form an "organism" which then is then applied to a particular goal. In an additional layer, the system provides high levels of diversity through varying execution and finally it provides intelligent algorithms for collaborative threat alert and detection. Although lacking an implementation, the architecture is interesting in providing a system designed with resilience from the ground up with

novel components, as opposed to those adapted on top of conventional systems. However the system complexity due its multiple layers has an adverse effect upon its resilience.

Table 8. Alternative Architecture Resilience

Method	Model	Components	Disciplines	Limitations
SlapOS - cloud distributed across homes [113]	IaaS	VR, RM, CONS	R	Low/non-deterministic redundancy
Modular and highly diverse SlapOS [31]	IaaS	VR, RM, CONS	R	Low/non-deterministic redundancy
Diversity using community clouds [41]	IaaS	VR, RM, CONST	DT, SEC	Low redundancy, uncertain governance
DefCloud using strong diversity across all layers [111]	IaaS	VR, PR, RM	SRV, FT,	High complexity
MEERKATS constant evolving diversity [58]	IaaS	VR, RM	SRV, FT,	Resource intensive
DREME replica execution diversity [16]	IaaS	VR, RM	SRV, FT,	Resource intensive
SCE+ using mass redundancy [99]	IaaS	VR, RM	FT	Resource intensive
BioRac, cell based diversity and redundancy [46]	IaaS	VR, RM	SRV, FT,	Resource intensive, High Complexity
SDN for resilient industrial IoT [100]	IaaS	CONS, CONST	R	Centralised management, high complexity

VR=Virtual Resources, RM=Resource Management, CONST=Constrained Devices, CONS=Consumer Devices, PI=Physical Infrastructure, R=Resilience, DT=Disruption Tolerance, SEC=Security, SRV=Survivability, FT=Fault Tolerance

2.6 Evaluation and Models

As with the resilience disciplines, measurement of cloud resilience could follow traditional performance-based resilience metrics such as Mean Time Between Failures (MTBF) and Mean Time Taken to Repair (MTTR) and the corresponding availability which is easily calculated from the two. However these metrics could be considered primitive at best [30] considering the complexity of these environments. The resilinets model [112] provides a method of determining which resilience features are available through binary selection of distinct features (e.g. the network provides confidentiality or it does not). Other non-cloud specific resilience metrics also suffice, such as graph metrics. Graph metrics are noted for their ease of comparing distinct architectures as they examine the structural characteristics of a network. Alenazi and Sterbenz evaluate a number of graph metrics for resilience are [6] and [7]. These include elementary metrics such as the quantity of nodes, node connectivity (the average number of connected nodes to each other node), node centrality (the most important nodes) etc. They also include those specifically for network resilience through removal of links and nodes e.g. network criticality and effective graph resistance. All of the above are arguably strongest when examining a distinct service as opposed to the entire cloud environment. Some of the works surveyed according to layer perform some form simulation of a model in order to evaluate resilience within the context of that particular use-case, the following works focus upon more generalised models for resilience.

Jabbar states that resilience is more difficult to measure than traditional security metrics due to the need to evaluate how effectively the service is still being delivered [52]. They propose that resilience should be measured as a state space considered in terms of degradation. Where a service is more resilience if it contains more states in which it stays operational and not severely degraded. Such a high-level approach may be applicable to complex environments.

Ghosh et. al. provides a model for resiliency based on stochastic reward nets [42]. The work is interesting in that the metrics for resiliency focus upon evaluating how effectively the job is scheduled through Quality of Service metrics. Those given are the rate of rejected jobs, and the delay in VM provision. Following from the definition of resiliency: "quantification of service delivery during changes", the authors evaluate changes as fluctuations in job arrival rate and the quantity of physical machines. Their results showed a faster provision rate was more resilient. Als that removal of a hot physical machine has an adverse effect upon resiliency, whereas removal of a cold one has a minimal effect.

Ju et. al. evaluate the resilience of OpenStack [55]. They develop a novel fault injection framework for both the architecture and its services. They uncovered 23 different bugs which developed into faults in the system. Highlighting the lack of effective resilience considerations within the stock cloud management software.

Tu and Xu present a resilience model built for a typical IaaS cloud, using Eucalyptus [118]. They explain that resilience and robustness are strongly connected in complex systems, where both properties describe the system's ability to react to disturbances but vary in how they do so. Considering the cloud as a multi-component, hierarchical system, the model evaluates the component interacting and interdependency upon resource consumption. Resilience is modelled by the strength of interactions between the components, where the strength is the percentage needed to consume from another component. A disturbance within the system results in a large queue, exhausting resources causing the services to fail. The authors describe system wide resilience as the quantity of processes which fail due to the inability to consume. They note that this system does not take into account factors which may influence the interaction strength such as one to many and many to one resource consumption interactions. It is mentioned that resiliency is accomplished through redundancy, which has an adverse affect upon cost. To fit in line with the author's model, they explain that increased redundancy, weakens the requirements for resource consumption links between individual components. Whilst redundancy is a key component of resiliency, it is not the only method, and poor implementations can even reduce resiliency under certain circumstances. The authors then attempt to understand more about this effect, examining of replication algorithms with modularisation of a cloud system. Their results show that as size increases, modulation is more important to prevent duplicate replication updates. However they also mention that poor modularisation implementation can create a single point of failure and thus become an enabling factor for poor resilience.

Scholler et. al. present an archtiectural model which enables insight into the security implications of cloud architectures [104] [49]. Their motivation is that current cloud services do not accommodate security and resiliency for critical infrastructures. Their model distinguishes between the different roles, (such as the physical provider, service developer and service user) as well as the different infrastructures (the physical and virtual) to assess the given requirements against the system. It promotes greater logging for audit purposes, as well as increased transparency between the physical and virtual layers, in order to increase trust between the users. Arguably, many issues within current cloud architectures ensure their unsuitability for a wide range of critical infrastructure services.

Sousa et. al. conduct an evaluation of Quality of Resilience evaluation criteria within the cloud, in order to activate appropriate proactive resilience measures [109]. They propose to use multiple criteria to evaluate the resilience, partly due to the wide variety of requirements associated with resilience and also because many proactive mechanisms require further information. The authors implement proactive resilience systems using multiple criteria for the cloud, MeTH [108] and TOPSIS [117]. The results showed that both methods improved the resilience of protocols which were unable to detect cloud layer faults but MeTH provided the greater performance in both fault and non-fault scenarios.

A classification of types of resilience metrics found within cloud computing is described below:

- **Binary feature** based metrics are those relating to the resilinets model such as confidentiality which either exist in the service of cloud or do not.
- **State-based** are those which examine the degradation of service to determine when resilience has failed.
- **Performance-oriented** metrics are the traditional type such as MTTR or QoS which typically involve examining one distinct service.
- **Graph-based** metrics examine issues in topologies such as network criticality.
- **Multi-criteria** metrics aggregate and summarise a variety of metrics into one to take into account very complex systems.

3 STATE-OF-THE-ART ANALYSIS

After reviewing the work in the previous section, an analysis of the state of the art of resilience in cloud literature is given in this section. It summarises the techniques used at each layer and the limitations of these techniques within the context of resilience in cloud environments. The complexity of the cloud environment is reflected in the multitude of characteristics and methods involved which enable resilience in cloud systems.

3.1 Techniques and Disciplines - Discussion and Limitations

To reiterate, the work in the previous section was grouped depending upon the layer of the cloud architecture it focuses upon. Some of the techniques employed may be seen across different layers but in different forms. Diversity and redundancy are two characteristics which are necessary attributes for a resilient system, albeit inherently costly. Both may be seen throughout the literature of cloud resilience in differing forms. The other techniques are autonomic, enabling dynamic adaptation to persist in service delivery.

The review in the previous section examined literature on cloud resilience for technique, architectural component applied, resilience disciplines used and the cloud layer in which the work is situated. Figure 2 classifies the techniques used to achieve resilience in the cloud into 3 separate categories: redundancy, diversity and autonomic management. Many of these techniques require no description, such as redundancy and diversity in hardware. However the autonomic techniques may be considerably more complex and will provide for new avenues of research. For example the management techniques to schedule and orchestrate services, and provision hardware are popular research topics although not always for resilience. Whilst software diversity techniques also see many different methods from dynamically altering protocols in transit to altering execution path diversity. Table 9 summarises the techniques and their limiations at each layer in the cloud.

Data-centre and physical resilience techniques leverage redundancy and diversity in both network links and data centre distribution, with survivability a notable discipline which is intuitive given the resilience upon networking at this layer. Most of these works tend to focus on resilience in inter-datacentre optical networks and most strongly in the placement and provisioning of datacentres. The latest studies push towards software defined techniques and evaluation/optimisation with upper-layer techniques. This point illustrates the weakness in attempting resilience at this layer due to the centralisation of management. Focusing resilience upon these layers is not feasible for the mast majority of cloud users/providers. Only those that manage at the datacentre or optical fibre link layer can affect this resilience.

Storage resilience techniques largely rely upon fault tolerance, optimising replication to both reduce the cost and optimise the processes involved. This is likely due to the most prevalent issue occurring in storage resilience is the failure of physical mediums. These can involve low level techniques such as erasure coding or topology/policy based

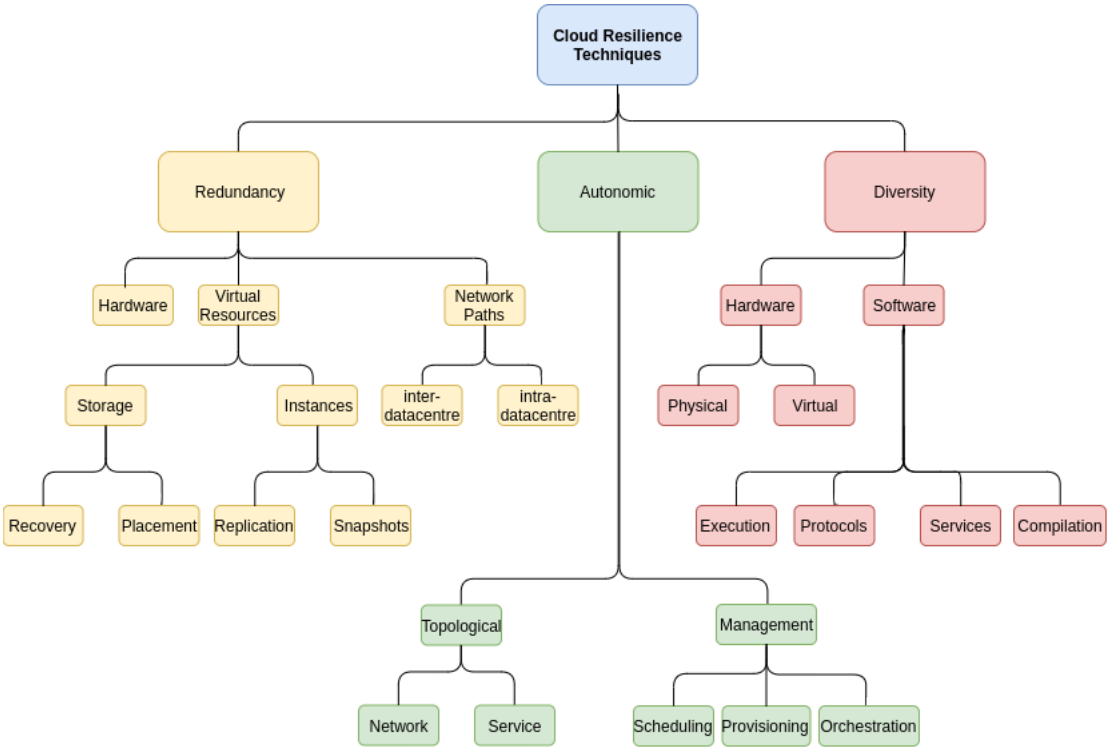

Fig. 2. Classification of techniques used in cloud resilience

placement methods. Due to the wealth of data involved in storage, techniques involving the CIA triad are also prevalent. Arguably data storage is a point of weaknesses in environments due to its cost and high target for theft. Therefore a better method of resilience might be to prevent its long term storage at all, where possible. These techniques, although attempt to optimise storage usage are still strongly resource intensive.

Atomic instance resilience techniques, as with storage, mostly focus upon fault tolerance techniques enabling process and system level snapshots, and replication. Again, as with storage, these techniques are very resource intensive and highlight the weakness in statefull applications operating in these environments. On a slightly higher level, hypervisor resilience techniques can again be seen using replication techniques although the diversity in compiler and runtimes is certainly an interesting and effective method if deployed correctly and with managed complexity.

Virtual networking techniques also tend to focus on survivability with some fault tolerance and traffic tolerance also. This is again an intuitive finding given the network-oriented nature of these work. Overall the techniques seen will often use some intelligence to optimise the way virtual overlay networks select links or route traffic upon the lower level physical networks, with some resource optimisation occurring to aggregate traffic on links with similar requirements and diversity selected to maximise survivability. Overall physical-aware virtual techniques show strong performance results. Virtual networks have been used for resiliency for a considerable time and it could be argued that many elementary networking techniques (e.g. VLANS) are virtual abstractions of underlying resources leveraged to increase resilience. The current drive towards software-defined-networking [102] is therefore a strong contender

and proponent for using virtual networking to provide resilience. However software-defined environments have considerable complexity and often found in centralised form therefore must be managed to enable its resilience.

Cloud management resilience techniques have the largest body of works. These tend to focus on the management of virtual resources (mostly virtual instances) to enable a wide variety of disciplines such as fault tolerance and robustness but also a number of security disciplines too. This shift in focus from the other layers (which tend to focus on one isolated discipline or cloud component) is likely due to the more holistic perspective at this layer. Component monitoring is a prominent method here, with the orchestration and scheduling of services also highly ranked. The majority of these techniques are accomplished through a novel middleware, with interoperability a crucial enabling factor. Overall this layer provides a clear advantage to conduct resilience and is arguably essential in either monitoring or execution. Some limitations of techniques at this layer include the lack of ability to affect lower level resilience, the huge complexity involved and the contrasting objectives between different applications.

Decentralised cloud resilience studies are spread across a number of disciplines. Survivability and fault tolerance are present as before but a number of security disciplines can be seen in addition to disruption and traffic tolerance. These techniques often focus on networking such as routing or middleware. However the data-driven nature of these environments, coupled with the inherently low security drives data security methods. Techniques such as ABE and anomaly intrusion detection are low-resource alternatives to traditional security solutions, designed to operate in hostile environments. Redundancy is still prevalent despite the lack of resources through the use of instance checkpointing, although the efficacy of this is questionable. What is missing from these disciplines are those disciplines which provide strong network resilience due to high node churn which may become an issue in contested and highly mobile smart environments in the future.

Alternative architectures present some variety on those previously mentioned. Diversity techniques are strongly represented, from topology based geo-spatial techniques on consumer hardware, execution level techniques or simply including diversity into every aspect of the architecture. The lack of redundancy based techniques over diversity could be argued as an attempt to move away from the costly methods which are undoubtedly have a negative effect. The primary limitations of these works is that most of these architectures are experimental yet evidence that drastically diverging from the traditional cloud architecture in order to provider stronger resilience yet meeting the functional requirements is possible and effective.

3.2 Resilience Techniques and Actor influence

It is also necessary to understand which actors can influence resilience at each layer. Therefore figure 3 illustrates the different techniques applied by actors at each layer. As the resilience techniques discovered focus mostly upon the resilience of the service as it is composed, the 3 most relevant actors were chosen. The CSP, the user and also the cloud broker. The cloud auditor and cloud carrier (as defined by NIST [76]) were excluded due to their lesser influence upon service composition. The rationale for each actor's influence at each layer is defined in table 10, ranked from 0 (no influence) to 3 (definite influence). The stronger user influence with VR can be strongly seen. Additionally, the more even distribution of influence at decentralised vs centralised cloud layers highlights a shift in responsibility.

3.3 Research Gaps

After an analysis of the previous work, a number of gaps may be seen in literature which may be addressed in future work. This work can span multiple levels but specifically concerns resilience in cloud environments which are disparate from the traditional cloud architectures.

Table 9. Summary of cloud resilience techniques and their limitations

Layer	Techniques Summary	Limitations
Physical Layer and Data-centre	Redundant network links and data centres	Costly redundancy
	Diversity through data-centre and server placement	Not suitable, variable for majority of cloud users
	Autonomic management	Centralisation and complexity
Storage Resilience	Resource replication optimisation	State-management complexity. Costly and could even be eliminated in certain use-cases
Atomic instance	Replication with some optimisation e.g. check pointing	State-management complexity. Costly and could even be eliminated in certain use-cases
Virtual Networking	Intelligent link selection and lower-level mapping	Requires accurate information exchange, complexity
	Traffic aggregation	Reduces compartmentalisation and therefore incident isolation
	Autonomic management	Centralisation and complexity
Cloud Management	Service orchestration	Complexity, multiple conflicting goals (SLAs)
	Task scheduling	Complexity, multiple conflicting goals (SLAs)
	Component monitoring	Resource intensive
	Centralised security management	Complexity, Resource intensive,
	Instance redundancy: replication, check-pointing etc.	Costly
Decentralised Architectures	Data-driven security methods	Cryptography heavy and resource intensive
	Some redundancy	questionable resource usage
	Decentralised autonomic management	Complexity
Alternative Architectures	Strong diversity techniques	experimental nature suffering from interoperability, "vendor" lock-in

Table 10. Rationale for Actor Influence upon Cloud Resilience Techniques

Layer	CSP	Broker	User
DC and Physical	3 – Has definite control over the selection of Dcs, links and cloud management.	1 – minimal selection capacity	1 – Minimal selection capacity
Storage and atomic instance	3 – ultimate control over replication techniques chosen	1- minimal technique selection	2 – Some control over data and software
VNetworking	3 – ultimate control over traffic, links and management	1- minimal technique selection	2- some control over virtual overlays
Cloud Management	3 – ultimate control over all aspects of cloud management	2 – some capacity for orchestration etc.	0 – no capacity
Decentralised	2 – cloud roles and responsibilities become dynamic	2 – some capacity to select providers	2 – influence with providers, geospatial and own devices
Alternative	3 – has ultimate control over management although often autonomic	2 – some capacity to select providers	2 – stronger influence with user-driven models

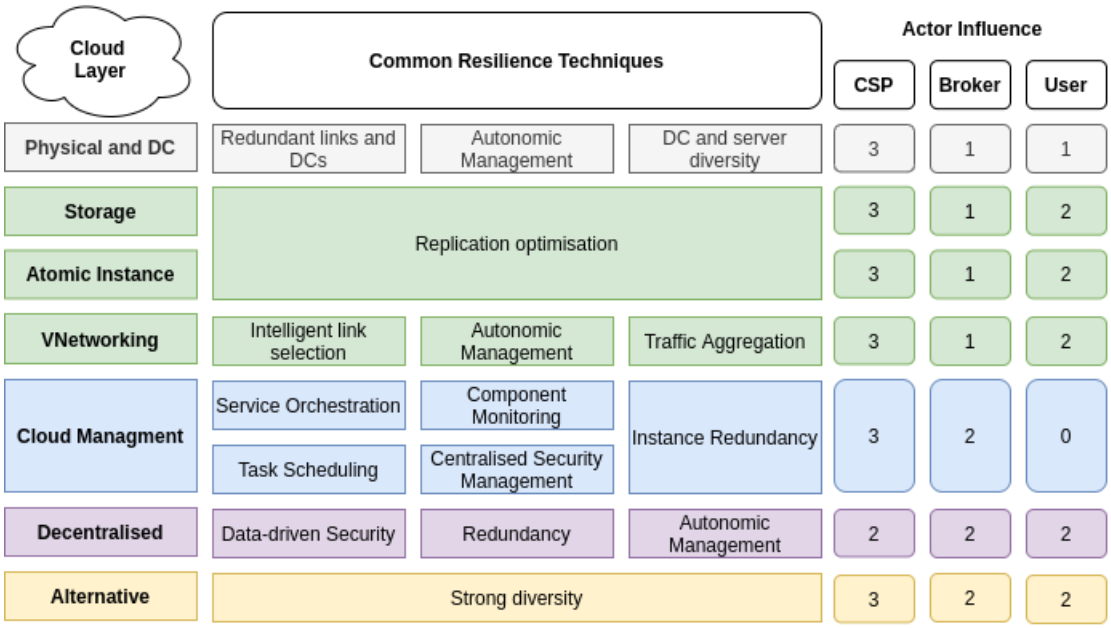

Fig. 3. Cloud resilience techniques used at each layer with corresponding actor influence upon resilience. 0=No influence, 1=Mild Influence, 2=Moderate Influence, 3=Definite Influence

3.3.1 Focus on specific layers. A key factor which is deemed relevant to the growing field of cloud which is one only investigated in an isolated context, is how does the effect of resilience upon one layer, affect the resilience of another layer? For example, if resilience is enabled by a user in the platform/service oriented layers but the underlying physical layer has low resilience, to what extent is it still possible to enable increase resilience in this manner? Such a topic is highly relevant to the way in which cloud architectures are evolving to more mobile and less deterministic networks and away from highly deterministic data centre environment. This has been touched on in some works such as the mapping from VNETS to physical networks [22] but there does not exist models or metrics to determine it for whole delivery platforms, i.e. a resilient PaaS on an non-resilient infrastructure. In addition this touches on the efforts of the layered resilience model which was the focus of the state of the art survey.

In terms of physical layer resilience, the exact effect upon resilience in the cloud with different levels of diverse hardware has seen minimal work. Therefore future research directions in this area could see the exact effect of diversification of hardware resources upon the resilience of a system be investigated. Barriers to this research mostly involve cost and time; as the necessary hardware, proprietary licenses and practical work involved in evaluating these scenarios ensures it is difficult to implement. However simulations may enabled a realisation of evaluating this approach.

3.3.2 Constraints and adaptive resilience. The ability to dynamically adjust within constrained environments is another area not touched upon sufficiently. Whilst some work within engineering has focused upon applying dynamic algorithms to graph analysis and optimisation, little work has been conducted which leverages this for the cloud. Again this has particular relevance for mobile environments due to the constrained resources available for optimisation and the more dynamic environment. Autonomic optimisation in WSNs is not a new concept [74]. Portocarrero et. al. conduct a

systematic review [92] and a number of optimisation and routing techniques [35]. Autonomic self-* characteristics are employed in certain types of networking but further work can involve an evaluation and comparison of different algorithms for both traditional and mobile cloud environments. Another area which could be expanded upon is the theoretical nature of enabling resilience within the context of various constraints. This has particular relevance to cloud SLA but also to constrained models. In short it concerns the analysis of requirements to enable the degree of resilience for the service. As resilience can be considered a scale (i.e. with state-based metrics) as opposed to a binary value, such a model could aid the construction of a service within its given constraints across all cloud service models.

3.3.3 Emerging Cloud Paradigms. Although not resilience specific, techniques are continuously emerging which attempt to further optimise cloud processes. These are worthy of consideration given their potential effect upon resilience although studies are largely lacking in literature. For example, data centre dissagregation is one such technique which involves managing cloud DCs in a resource-centric manner. This is in contrast to traditional server-centric DCs where physical resources (e.g. compute and memory) are stored on a single server. Through dissagregation, similar resources can be be physically decoupled and mounted together in same-resource blades or even racks [114]. This is envisioned to vastly optimise resource management, Pages et. al. illustrate a 50% increase in virtual instance capacity in optically-connected intra-DCs[89]. Not only limited to the centralised cloud, Ajibola et.al. illustrate a reduction in 50% of required fog nodes for specific tasks [4]. Strong performance increases across a variety of paradigms enhances the possibility of uptake. However few works can be seen which highlight how these techniques affect resilience. At first glance, homogenous resources spatially grouped together increases the likelihood of low availability of a specific resource and thus is not resilient. Conversely, enhanced and dynamic resource optimisation may enable dynamic fault tolerance. There are considerable variation in effects upon resilience with this new paradigm which should therefore be studied prior to its implementation.

3.3.4 Decentralised Cloud. Arguably, the requirement for resilience at the decentralised cloud layer (i.e. close to the edge) is greater than the centralised due to data-centre hardware being resilient by nature. Such constrained environments have less ability to fall back upon redundancy and cryptographic methods in order to provide their resilience and generally operate in a hostile environment. They might also employ a variety of diversity related techniques due to disparate hardware involved. The foundation of IoT networks of WSNs and MANETS which have seen bodies of literature [133] [17] attempting to optimise resilient communication, security etc. It could therefore be argued that the entire focus of these disciplines is in delivering a resilient platform given the hostile environment in which they operate. However the decentralised disciplines defined previously consist of more than simply IoT networks. There is now an entire ecosystem where continuously evolving use-cases demand rich data processing at any and all layers from the IoT device, to the transportation/Fog layer back to the centralised cloud. These networks are heterogeneous and non-deterministic which further complicate matters. Traffic will traverse multiple governance domains, operate on a diverse plethora of hardware/software configurations and requirements for performance and resilience will change in fractions of a second according to external and internal requirements.

3.4 Challenges for Resilience in Cloud Computing

A final meta-analysis of the results of this survey highlights challenges for resilience in cloud computing which we envision will drive new avenues of research. Characteristics which create challenges to cloud resilience are discussed below:

(1) **Use-case Diversity** - While cloud environments are inherently employed to provide resources for diverse use-cases, resilience techniques tend to be developed for specific use-cases. This highlighted the need for cloud environments to provide *adaptive resilience* according to the need. Integrating a plethora of techniques and selecting the most appropriate is thus an ongoing challenge for the current and emerging cloud.
(2) **Uncertain and dynamic governance and responsibility** - Traditional cloud delivery models (SaaS/PaaS/IaaS) define clear responsibility boundaries between the CSP and the user. They can assist in determining which actor can affect resilience at which layer. However, in decentralised cloud disciplines, particularly those with node mobility (e.g. MEC and fog computing) these actors can dynamically change according to physical boundaries and network requirements. Ensuring the capacity to both understand and monitor who has responsibility for resilience extemporaneously is crucial to providing resilience in decentralised clouds.
(3) **Evolving cloud paradigms** - Summarising a key concern during this survey is the manner in which cloud computing, as a concept, is continuously in flux. Driven by both changing use-cases and continuous strives for optimisation, the deployment of new and emerging cloud paradigms poses a challenge to service resilience. Where the resilience of new techniques should be considered during their development and not post-deployment.

4 CONCLUSION

To conclude this work, we provided a contribution to knowledge through a comprehensive review and analysis of literature which focuses upon providing resilience across the entire cloud computing consortium. The analysis was structured according to a novel methodology using specific layers within the cloud architecture to accommodate the complexity of these environments. This provided a greater insight into the techniques employed and which were lacking at each layer. We highlighted a number of gaps in literature which focused mostly on the greater need for resilience at decentralised layers and the edge. Firstly we note that almost no works consider the resilience of both centralised and decentralised cloud architectures in tandem. Applications for resilience which vary according to the underlying requirements and can be distributed across multiple disciplines are essential due to the increasing and wide ranging use-cases for these areas. We also note that many cloud resilience techniques rely on the costly method of redundancy. The "seemingly unlimited" resources available in centralised cloud environments will be a key driver of these techniques. However for decentralised cloud disciplines this is less applicable due to their resource constrained nature. One solution is to move to stateless applications where storage redundancy is not needed. Autonomic techniques for managing decentralisation are highlighted as a strong candidate for resilience in constrained environments. Finally understanding and considering the dynamic boundaries of responsibilities for resilience in emerging and decentralised cloud is vital.

ACKNOWLEDGMENTS

This work was supported, in part, by Science Foundation Ireland grants 16/RC/3918 and 13/RC/2094.

REFERENCES

[1] N.A.S. Abdullah, N.L. Md Noor, and E.N.M. Ibrahim. 2013. Resilient organization: Modelling the capacity for resilience. In *Research and Innovation in Information Systems (ICRIIS), 2013 International Conference on*. 319–324. https://doi.org/10.1109/ICRIIS.2013.6716729
[2] Yuan Ai, Mugen Peng, and Kecheng Zhang. 2018. Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks* 4, 2 (2018), 77 – 86. https://doi.org/10.1016/j.dcan.2017.07.001
[3] M. Aibin and K. Walkowiak. 2018. Monte Carlo Tree Search for Cross-Stratum Optimization of Survivable Inter-Data Center Elastic Optical Network. In *2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM)*. 1–7. https://doi.org/10.1109/RNDM.2018.8489841
[4] Opeyemi O Ajibola, Taisir EH El-Gorashi, and Jaafar MH Elmirghani. 2019. Disaggregation for Improved Efficiency in Fog Computing Era. In *2019 21st International Conference on Transparent Optical Networks (ICTON)*. IEEE, 1–7.

[5] Mahmoud Al-Ayyoub, Muneera Al-Quraan, Yaser Jararweh, Elhadj Benkhelifa, and Salim Hariri. 2018. Resilient service provisioning in cloud based data centers. *Future Generation Computer Systems* 86 (2018), 765 – 774. https://doi.org/10.1016/j.future.2017.07.005

[6] M.J.F. Alenazi and J.P.G. Sterbenz. 2015. Comprehensive comparison and accuracy of graph metrics in predicting network resilience. In *Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the*. 157–164. https://doi.org/10.1109/DRCN.2015.7149007

[7] M.J.F. Alenazi and J.P.G. Sterbenz. 2015. Evaluation and improvement of network resilience against attacks using graph spectral metrics. In *Resilience Week (RWS), 2015*. 1–6. https://doi.org/10.1109/RWEEK.2015.7287447

[8] S. Antony, S. Antony, A. S. A. Beegom, and M. S. Rajasree. 2012. Task Scheduling Algorithm with Fault Tolerance for Cloud. In *2012 International Conference on Computing Sciences*. 180–182. https://doi.org/10.1109/ICCS.2012.71

[9] A. Aral and I. Brandic. 2018. Dependency Mining for Service Resilience at the Edge. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. 228–242. https://doi.org/10.1109/SEC.2018.00024

[10] J. P. Araujo Neto, D. M. Pianto, and C. G. Ralha. 2018. An Agent-Based Fog Computing Architecture for Resilience on Amazon EC2 Spot Instances. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. 360–365. https://doi.org/10.1109/BRACIS.2018.00069

[11] A. C. Baktir, A. Ozgovde, and C. Ersoy. 2017. How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases, and Future Directions. *IEEE Communications Surveys Tutorials* 19, 4 (Fourthquarter 2017), 2359–2391. https://doi.org/10.1109/COMST.2017.2717482

[12] I.B. Barla, K. Hoffmann, M. Hoffmann, D.A. Schupke, and G. Carle. 2013. Shared protection in virtual networks. In *Communications Workshops (ICC), 2013 IEEE International Conference on*. 240–245. https://doi.org/10.1109/ICCW.2013.6649236

[13] I.B. Barla, D.A. Schupke, and G. Carle. 2012. Delay Performance of Resilient Cloud Services over Networks. In *Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on*. 512–517. https://doi.org/10.1109/ISPA.2012.75

[14] I.B. Barla, D.A. Schupke, M. Hoffmann, and G. Carle. 2013. Optimal design of virtual networks for resilient cloud services. In *Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the*. 218–225.

[15] I. B. Barla Harter, D. A. Schupke, M. Hoffmann, and G. Carle. 2015. Optimal design of resilient virtual networks [Invited]. *IEEE/OSA Journal of Optical Communications and Networking* 7, 2 (February 2015), A218–A234. https://doi.org/10.1364/JOCN.7.00A218

[16] A. Benameur, N.S. Evans, and M.C. Elder. 2013. Cloud resiliency and security via diversified replica execution and monitoring. In *Resilient Control Systems (ISRCS), 2013 6th International Symposium on*. 150–155. https://doi.org/10.1109/ISRCS.2013.6623768

[17] E. Benkhelifa, T. Welsh, and W. Hamouda. 2018. A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems. *IEEE Communications Surveys Tutorials* 20, 4 (Fourthquarter 2018), 3496–3509. https://doi.org/10.1109/COMST.2018.2844742

[18] K. E. Benson, G. Wang, N. Venkatasubramanian, and Y. Kim. 2018. Ride: A Resilient IoT Data Exchange Middleware Leveraging SDN and Edge Cloud Resources. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 72–83. https://doi.org/10.1109/IoTDI.2018.00017

[19] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando André, and Paulo Sousa. 2013. DepSky: Dependable and Secure Storage in a Cloud-of-Clouds. *Trans. Storage* 9, 4, Article 12 (Nov. 2013), 33 pages. https://doi.org/10.1145/2535929

[20] Kashif Bilal, Osman Khalid, Aiman Erbad, and Samee U. Khan. 2018. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Computer Networks* 130 (2018), 94 – 120. https://doi.org/10.1016/j.comnet.2017.10.002

[21] A. Binun, M. Bloch, S. Dolev, M.R. Kahil, B. Menuhin, R. Yagel, T. Coupaye, M. Lacoste, and A. Wailly. 2014. Self-Stabilizing Virtual Machine Hypervisor Architecture for Resilient Cloud. In *Services (SERVICES), 2014 IEEE World Congress on*. 200–207. https://doi.org/10.1109/SERVICES.2014.44

[22] Minh Bui, B. Jaumard, and C. Develder. 2013. Anycast end-to-end resilience for cloud services over virtual optical networks. In *Transparent Optical Networks (ICTON), 2013 15th International Conference on*. 1–7. https://doi.org/10.1109/ICTON.2013.6603032

[23] Minh Bui, Ting Wang, B. Jaumard, D. Medhi, and C. Develder. 2014. Time-varying resilient virtual network mapping for multi-location cloud data centers. In *Transparent Optical Networks (ICTON), 2014 16th International Conference on*. 1–8. https://doi.org/10.1109/ICTON.2014.6876287

[24] Gokhan Calis and Onur Ozan Koyluoglu. 2014. Repairable Block Failure Resilient Codes. *CoRR* abs/1406.7264 (2014).

[25] John Cartlidge and Ilango Sriram. 2011. Modelling Resilience in Cloud-Scale Data Centres. *CoRR* abs/1106.5457 (2011). http://arxiv.org/abs/1106.5457

[26] Marco Carvalho, Dipankar Dasgupta, Michael Grimaila, and Carlos Perez. 2011. Mission resilience in cloud computing: A biologically inspired approach. In *6th International Conference on Information Warfare and Security*. 42–52.

[27] Sonali Chandna, Nabil Naas, and Hussein Mouftah. 2019. Software Defined Survivable Optical Interconnect for data centers. *Optical Switching and Networking* 31 (2019), 86 – 99. https://doi.org/10.1016/j.osn.2018.10.001

[28] Brijesh Kashyap Chejerla and Sanjay. K. Madria. 2017. QoS guaranteeing robust scheduling in attack resilient cloud integrated cyber physical system. *Future Generation Computer Systems* 75 (2017), 145 – 157. https://doi.org/10.1016/j.future.2017.02.034

[29] Mehdi Nazari Cheraghlou, Ahmad Khadem-Zadeh, and Majid Haghparast. 2016. A survey of fault tolerance architecture in cloud computing. *Journal of Network and Computer Applications* 61 (2016), 81 – 92. https://doi.org/10.1016/j.jnca.2015.10.004

[30] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee. 2016. A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications. *IEEE Communications Surveys Tutorials* 18, 3 (thirdquarter 2016), 2244–2281. https://doi.org/10.1109/COMST.2016.2531104

[31] R. Courteaud, Yingjie Xu, and C. Cerin. 2012. Practical solutions for resilience in SlapOS. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. 488–495. https://doi.org/10.1109/CloudCom.2012.6427511

[32] Rodrigo S. Couto, Stefano Secci, Miguel Elias M. Campista, and Luís Henrique M.K. Costa. 2015. Server placement with shared backups for disaster-resilient clouds. *Computer Networks* 93 (2015), 423 – 434. https://doi.org/10.1016/j.comnet.2015.09.039 Cloud Networking and Communications

II.

[33] R. S. Couto, S. Secci, M. E. M. Campista, and L. H. M. K. Costa. 2014. Latency versus survivability in geo-distributed data center design. In *2014 IEEE Global Communications Conference*. 1102–1107. https://doi.org/10.1109/GLOCOM.2014.7036956

[34] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, and Andrew Warfield. 2008. Remus: High Availability via Asynchronous Virtual Machine Replication. In *5th USENIX Symposium on Networked Systems Design and Implementation (NSDI 08)*. USENIX Association, San Francisco, CA. https://www.usenix.org/conference/nsdi-08/remus-high-availability-asynchronous-virtual-machine-replication

[35] Miguel Franklin de Castro, Levi Bayde Ribeiro, and Camila Helena Souza Oliveira. 2012. An autonomic bio-inspired algorithm for wireless sensor network self-organization and efficient routing. *Journal of Network and Computer Applications* 35, 6 (2012), 2003 – 2015. https://doi.org/10.1016/j.jnca.2012.07.023

[36] I. P. Egwutuoha, S. Chen, D. Levy, and B. Selic. 2012. A Fault Tolerance Framework for High Performance Computing in Cloud. In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. 709–710. https://doi.org/10.1109/CCGrid.2012.80

[37] S. Eisele, I. Mardari, A. Dubey, and G. Karsai. 2017. RIAPS: Resilient Information Architecture Platform for Decentralized Smart Systems. In *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)*. 125–132. https://doi.org/10.1109/ISORC.2017.22

[38] S. Ferdousi, F. Dikhiyik, M. F. Habib, and B. Mukherjee. 2013. Disaster-aware data-center and content placement in cloud networks. In *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. 1–3. https://doi.org/10.1109/ANTS.2013.6802881

[39] Marc Eduard Frîncu. 2014. Scheduling highly available applications on cloud environments. *Future Generation Computer Systems* 32 (2014), 138 – 153. https://doi.org/10.1016/j.future.2012.05.017 Special Section: The Management of Cloud Systems, Special Section: Cyber-Physical Society and Special Section: Special Issue on Exploiting Semantic Technologies with Particularization on Linked Data over Grid and Cloud Architectures.

[40] Yue Gao, S.K. Gupta, Yanzhi Wang, and M. Pedram. 2014. An energy-aware fault tolerant scheduling framework for soft error resilient cloud computing systems. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*. 1–6. https://doi.org/10.7873/DATE.2014.107

[41] G. Garlick. 2011. Improving Resilience with Community Cloud Computing. In *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*. 650–655. https://doi.org/10.1109/ARES.2011.100

[42] Rahul Ghosh, Francesco Longo, Vijay K. Naik, and Kishor S. Trivedi. 2010. Quantifying Resiliency of IaaS Cloud. In *Proceedings of the 2010 29th IEEE Symposium on Reliable Distributed Systems (SRDS '10)*. IEEE Computer Society, Washington, DC, USA, 343–347. https://doi.org/10.1109/SRDS.2010.49

[43] J.L. Gonzalez, Jesus Carretero Perez, Victor J. Sosa-Sosa, Luis M. Sanchez, and Borja Bergua. 2015. SkyCDS: A resilient content delivery service based on diversified cloud storage. *Simulation Modelling Practice and Theory* 54 (2015), 64 – 85. https://doi.org/10.1016/j.simpat.2015.03.006

[44] Róża Goścień and Krzysztof Walkowiak. 2017. Modeling and optimization of data center location and routing and spectrum allocation in survivable elastic optical networks. *Optical Switching and Networking* 23 (2017), 129 – 143. https://doi.org/10.1016/j.osn.2016.06.004 Design and modeling of Resilient optical networks RNDM 2015.

[45] Minzhe Guo and Prabir Bhattacharya. 2014. Diverse Virtual Replicas for Improving Intrusion Tolerance in Cloud. In *Proceedings of the 9th Annual Cyber and Information Security Research Conference (CISR '14)*. ACM, New York, NY, USA, 41–44. https://doi.org/10.1145/2602087.2602116

[46] Salim Hariri, Mohamed Eltoweissy, and Youssif Al-Nashif. 2011. BioRAC: Biologically Inspired Resilient Autonomic Cloud. In *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '11)*. ACM, New York, NY, USA, Article 80, 1 pages. https://doi.org/10.1145/2179298.2179389

[47] I.B.B. Harter, M. Hoffmann, D.A. Schupke, and G. Carle. 2014. Scalable resilient virtual network design algorithms for cloud services. In *Reliable Networks Design and Modeling (RNDM), 2014 6th International Workshop on*. 123–130. https://doi.org/10.1109/RNDM.2014.7014941

[48] I. B. B. Harter, D. A. Schupke, M. Hoffmann, and G. Carle. 2014. Network virtualization for disaster resilience of cloud services. *IEEE Communications Magazine* 52, 12 (December 2014), 88–95. https://doi.org/10.1109/MCOM.2014.6979957

[49] T. Hecht, P. Smith, and M. Scholler. 2014. Critical services in the cloud: Understanding security and resilience risks. In *Reliable Networks Design and Modeling (RNDM), 2014 6th International Workshop on*. 131–137. https://doi.org/10.1109/RNDM.2014.7014942

[50] A. Hussein, I. H. Elhajj, A. Chehab, and A. Kayssi. 2017. SDN VANETs in 5G: An architecture for resilient security services. In *2017 Fourth International Conference on Software Defined Systems (SDS)*. 67–74. https://doi.org/10.1109/SDS.2017.7939143

[51] A. Imran, A. U. Gias, R. Rahman, A. Seal, T. Rahman, F. Ishraque, and K. Sakib. 2014. Cloud-Niagara: A high availability and low overhead fault tolerance middleware for the cloud. In *16th Int'l Conf. Computer and Information Technology*. 271–276. https://doi.org/10.1109/ICCITechn.2014.6997344

[52] Abdul Jabbar. 2010. *A Framework to Quantify Network Resilience and Survivability*. Ph.D. Dissertation. Lawrence, KS, USA. Advisor(s) Sterbenz, James P.G. AAI3417968.

[53] V. Jaiswal, A. Sen, and A. Verma. 2014. Integrated Resiliency Planning in Storage Clouds. *Network and Service Management, IEEE Transactions on* 11, 1 (March 2014), 3–14. https://doi.org/10.1109/TNSM.2013.120713.120349

[54] Ravi Jhawar and Vincenzo Piuri. 2013. Fault tolerance and resilience in cloud computing environments. *Computer and Information Security Handbook*, (2013), 125–141.

[55] Xiaoen Ju, Livio Soares, Kang G. Shin, Kyung Dong Ryu, and Dilma Da Silva. 2013. On Fault Resilience of OpenStack. In *Proceedings of the 4th Annual Symposium on Cloud Computing (SOCC '13)*. ACM, New York, NY, USA, Article 2, 16 pages. https://doi.org/10.1145/2523616.2523622

[56] M. Kahla, M. Azab, and A. Mansour. 2018. Secure, Resilient, and Self-Configuring Fog Architecture for Untrustworthy IoT Environments. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. 49–54. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00018

[57] M. Kanter and S. Taylor. 2013. Diversity in cloud systems through runtime and compile-time relocation. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*. 396–402. https://doi.org/10.1109/THS.2013.6699037

[58] A.D. Keromytis, R. Geambasu, S. Sethumadhavan, S.J. Stolfo, Junfeng Yang, A. Benameur, M. Dacier, M. Elder, D. Kienzle, and A. Stavrou. 2012. The MEERKATS Cloud Security Architecture. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*. 446–450. https://doi.org/10.1109/ICDCSW.2012.42

[59] A. Khalifa, M. Azab, and M. Eltoweissy. 2014. Resilient hybrid Mobile Ad-hoc Cloud over collaborating heterogeneous nodes. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*. 134–143.

[60] Klein et al. 2014. Brownout: Building More Robust Cloud Applications.

[61] C. Klein et al. 2014. Improving Cloud Service Resilience Using Brownout-Aware Load-Balancing. In *Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium on*. 31–40. https://doi.org/10.1109/SRDS.2014.14

[62] J.-C. Laprie. 2005. Resilience for the Scalability of Dependability. In *Network Computing and Applications, Fourth IEEE International Symposium on*. 5–6. https://doi.org/10.1109/NCA.2005.44

[63] M. Le, Z. Song, Y. Kwon, and E. Tilevich. 2017. Reliable and efficient mobile edge computing in highly dynamic and volatile environments. In *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. 113–120. https://doi.org/10.1109/FMEC.2017.7946417

[64] X. Li, T. Gao, L. Zhang, Y. Tang, Y. Zhang, and S. Huang. 2018. Survivable K-Node (Edge) Content Connected Virtual Optical Network (KC-VON) Embedding Over Elastic Optical Data Center Networks. *IEEE Access* 6 (2018), 38780–38793. https://doi.org/10.1109/ACCESS.2018.2852814

[65] Qianhui Liang and Bu-Sung Lee. 2011. Delivering High Resilience in Designing Platform-as-a-Service Clouds. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. 676–683. https://doi.org/10.1109/CLOUD.2011.72

[66] Hsien-Chun Liao and Chien-Fu Cheng. 2014. A Malicious-Resilient Protocol for Consistent Scheduling Problem in the Cloud Computing Environment. *Comput. J.* 58, 2 (04 2014), 315–330. https://doi.org/10.1093/comjnl/bxu028 arXiv:http://oup.prod.sis.lan/comjnl/article-pdf/58/2/315/1037973/bxu028.pdf

[67] Guanglei Liu and Chuanyi Ji. 2009. Scalability of Network-Failure Resilience: Analysis Using Multi-Layer Probabilistic Graphical Models. *Networking, IEEE/ACM Transactions on* 17, 1 (Feb 2009), 319–331. https://doi.org/10.1109/TNET.2008.925944

[68] J. Liu and H. Shen. 2016. A Low-Cost Multi-failure Resilient Replication Scheme for High Data Availability in Cloud Storage. In *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*. 242–251. https://doi.org/10.1109/HiPC.2016.036

[69] F. Lombardi, R. Di Pietro, and C. Soriente. 2010. CReW: Cloud Resilience for Windows Guests through Monitored Virtualization. In *Reliable Distributed Systems, 2010 29th IEEE Symposium on*. 338–342. https://doi.org/10.1109/SRDS.2010.48

[70] Thouraya Louati, Heithem Abbes, and Christophe Cérin. 2018. LXCloudFT: Towards high availability, fault tolerant Cloud system based Linux Containers. *J. Parallel and Distrib. Comput.* 122 (2018), 51 – 69. https://doi.org/10.1016/j.jpdc.2018.07.015

[71] Bing Luo and W. Liu. 2011. The Sustainability and Survivabiltiy Network Design for Next Generation Cloud Networking. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*. 555–560. https://doi.org/10.1109/DASC.2011.103

[72] P. Mach and Z. Becvar. 2017. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys Tutorials* 19, 3 (thirdquarter 2017), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

[73] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. 2017. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys Tutorials* 19, 4 (Fourthquarter 2017), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201

[74] David Marsh, Richard Tynan, Donal O'Kane, and Gregory M. P. O'Hare. 2004. Autonomic wireless sensor networks. *Engineering Applications of Artificial Intelligence* 17, 7 (2004), 741 – 748. https://doi.org/10.1016/j.engappai.2004.08.038 Autonomic Computing Systems.

[75] David R. Matos, Miguel L. Pardal, Georg Carle, and Miguel Correia. 2018. RockFS: Cloud-backed File System Resilience to Client-Side Attacks. In *Proceedings of the 19th International Middleware Conference (Middleware '18)*. ACM, New York, NY, USA, 107–119. https://doi.org/10.1145/3274808.3274817

[76] Peter M. Mell and Timothy Grance. 2011. *SP 800-145. The NIST Definition of Cloud Computing*. Technical Report. Gaithersburg, MD, United States.

[77] Madalin Mihailescu et al. 2011. Enhancing Application Robustness in Cloud Data Centers. In *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research (CASCON '11)*. IBM Corp., Riverton, NJ, USA, 133–147.

[78] Bahareh Alami Milani and Nima Jafari Navimipour. 2016. A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions. *Journal of Network and Computer Applications* 64 (2016), 229 – 238. https://doi.org/10.1016/j.jnca.2016.02.005

[79] A. Modarresi, S. Gangadhar, and J. P. G. Sterbenz. 2017. A framework for improving network resilience using SDN and fog nodes. In *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*. 1–7. https://doi.org/10.1109/RNDM.2017.8093036

[80] A. Modarresi and J. P. G. Sterbenz. 2017. Toward resilient networks with fog computing. In *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*. 1–7. https://doi.org/10.1109/RNDM.2017.8093032

[81] Yehia H Khalil Mohamed. 2011. Data center resilience assessment: storage, networking and security.

[82] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos. 2018. A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. *IEEE Communications Surveys Tutorials* 20, 1 (Firstquarter 2018), 416–464. https://doi.org/10.1109/COMST.2017.2771153

[83] Rekha Nachiappan, Bahman Javadi, Rodrigo N. Calheiros, and Kenan M. Matawie. 2017. Cloud storage reliability for Big Data applications: A state of the art survey. *Journal of Network and Computer Applications* 97 (2017), 35 – 47. https://doi.org/10.1016/j.jnca.2017.08.011

[84] W. Najjar and J.-L. Gaudiot. 1990. Network resilience: a measure of network fault tolerance. *Computers, IEEE Transactions on* 39, 2 (Feb 1990), 174–181. https://doi.org/10.1109/12.45203

[85] Toan Nguyen, J.-A. Desideri, and L. Trifan. 2012. Applications resilience on clouds. In *High Performance Computing and Simulation (HPCS), 2012 International Conference on*. 60–66. https://doi.org/10.1109/HPCSim.2012.6266891

[86] B. Nicolae and F. Cappello. 2011. BlobCR: Efficient checkpoint-restart for HPC applications on IaaS clouds using virtual disk image snapshots. In *SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–12. https://doi.org/10.1145/2063384.2063429

[87] Opeyemi Osanaiye, Kim-Kwang Raymond Choo, and Mqhele Dlodlo. 2016. Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. *Journal of Network and Computer Applications* 67 (2016), 147 – 165. https://doi.org/10.1016/j.jnca.2016.01.001

[88] Umar Ozeer, Xavier Etchevers, Loïc Letondeur, François-Gaël Ottogalli, Gwen Salaün, and Jean-Marc Vincent. 2018. Resilience of Stateful IoT Applications in a Dynamic Fog Environment. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*. ACM, New York, NY, USA, 332–341. https://doi.org/10.1145/3286978.3287007

[89] Albert Pages, Rubén Serrano, Jordi Perelló, and Salvatore Spadaro. 2017. On the benefits of resource disaggregation for virtual data centre provisioning in optical data centres. *Computer Communications* 107 (2017), 60–74.

[90] J. Pan and J. McElhannon. 2018. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet of Things Journal* 5, 1 (Feb 2018), 439–449. https://doi.org/10.1109/JIOT.2017.2767608

[91] Deepak Poola, Mohsen Amini Salehi, Kotagiri Ramamohanarao, and Rajkumar Buyya. 2017. Chapter 15 - A Taxonomy and Survey of Fault-Tolerant Workflow Management Systems in Cloud and Distributed Computing Environments. In *Software Architecture for Big Data and the Cloud*, Ivan Mistrik, Rami Bahsoon, Nour Ali, Maritta Heisel, and Bruce Maxim (Eds.). Morgan Kaufmann, Boston, 285 – 320. https://doi.org/10.1016/B978-0-12-805467-3.00015-6

[92] Jesús MT Portocarrero, Flávia C Delicato, Paulo F Pires, Nadia Gámez, Lidia Fuentes, David Ludovino, and Paulo Ferreira. 2014. Autonomic wireless sensor networks: a systematic literature review. *Journal of Sensors* 2014 (2014).

[93] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis. 2015. The Benefits of Self-Awareness and Attention in Fog and Mist Computing. *Computer* 48, 7 (July 2015), 37–45. https://doi.org/10.1109/MC.2015.207

[94] Y. Qu and N. Xiong. 2012. RFH: A Resilient, Fault-Tolerant and High-Efficient Replication Algorithm for Distributed Cloud Storage. In *2012 41st International Conference on Parallel Processing*. 520–529. https://doi.org/10.1109/ICPP.2012.3

[95] C. Queiroz, S.K. Garg, and Z. Tari. 2013. A probabilistic model for quantifying the resilience of networked systems. *IBM Journal of Research and Development* 57, 5 (Sept 2013), 3:1–3:9. https://doi.org/10.1147/JRD.2013.2259433

[96] H.P. Reiser and R. Kapitza. 2007. Hypervisor-Based Efficient Proactive Recovery. In *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*. 83–92. https://doi.org/10.1109/SRDS.2007.25

[97] R. Rios, R. Roman, J. A. Onieva, and J. Lopez. 2017. From SMOG to Fog: A security perspective. In *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. 56–61. https://doi.org/10.1109/FMEC.2017.7946408

[98] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. 2018. Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems* 78 (2018), 680 – 698. https://doi.org/10.1016/j.future.2016.11.009

[99] V. Salapura, R. Harper, and M. Viswanathan. 2013. Resilient cloud computing. *IBM Journal of Research and Development* 57, 5 (Sept 2013), 10:1–10:12. https://doi.org/10.1147/JRD.2013.2266972

[100] Arjuna Sathiaseelan, Mennan Selimi, Carlos Molina, Adisorn Lertsinsrubtavee, Leandro Navarro, Felix Freitag, Fernando Ramos, and Roger Baig. 2017. Towards Decentralised Resilient Community Clouds. In *Proceedings of the 2Nd Workshop on Middleware for Edge Clouds & Cloudlets (MECC '17)*. ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3152360.3152363

[101] Daniel J. Scales, Mike Nelson, and Ganesh Venkitachalam. 2010. The Design of a Practical System for Fault-tolerant Virtual Machines. *SIGOPS Oper. Syst. Rev.* 44, 4 (Dec. 2010), 30–39. https://doi.org/10.1145/1899928.1899932

[102] Sibylle Schaller and Dave Hood. 2017. Software defined networking architecture standardization. *Computer Standards Interfaces* 54 (2017), 197 – 202. https://doi.org/10.1016/j.csi.2017.01.005 SI: Standardization SDNNFV.

[103] M. Scholler et al. 2013. Resilient deployment of virtual network functions. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013 5th International Congress on*. 208–214. https://doi.org/10.1109/ICUMT.2013.6798428

[104] M. Scholler, R. Bless, F. Pallas, J. Horneber, and P. Smith. 2013. An Architectural Model for Deploying Critical Infrastructure Services in the Cloud. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, Vol. 1. 458–466. https://doi.org/10.1109/CloudCom.2013.67

[105] S. Secci and S. Murugesan. 2014. Cloud Networks: Enhancing Performance and Resiliency. *Computer* 47, 10 (Oct 2014), 82–85. https://doi.org/10.1109/MC.2014.277

[106] Vibhu Saujanya Sharma and Aravindan Santharam. 2013. Implementing a Resilient Application Architecture for State Management on a PaaS Cloud. In *Proceedings of the 2013 IEEE International Conference on Cloud Computing Technology and Science - Volume 01 (CLOUDCOM '13)*. IEEE Computer Society, Washington, DC, USA, 142–147.

[107] Noor-ul-hassan Shirazi, Steven Simpson, Simon Oechsner, Andreas Mauthe, and David Hutchison. 2015. A framework for resilience management in the cloud. *e & i Elektrotechnik und Informationstechnik* 132, 2 (01 Mar 2015), 122–132. https://doi.org/10.1007/s00502-015-0290-9

[108] Bruno Sousa, Kostas Pentikousis, and Marilia Curado. 2014. MeTHODICAL: Towards the next generation of multihomed applications. *Computer Networks* 65 (2014), 21 – 40.

[109] B. Sousa, K. Pentikousis, and M. Curado. 2014. Optimizing quality of resilience in the cloud. In *Global Communications Conference (GLOBECOM), 2014 IEEE*. 1133–1138. https://doi.org/10.1109/GLOCOM.2014.7036961

[110] R. Souza Couto, S. Secci, M. Mitre Campista, and L.M. Kosmalski Costa. 2014. Network design requirements for disaster resilience in IaaS clouds. *Communications Magazine, IEEE* 52, 10 (October 2014), 52–58. https://doi.org/10.1109/MCOM.2014.6917402

[111] J.P.G. Sterbenz and P. Kulkarni. 2013. Diverse Infrastructure and Architecture for Datacenter and Cloud Resilience. In *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*. 1–7. https://doi.org/10.1109/ICCCN.2013.6614125

[112] James P. G. Sterbenz, David Hutchison, Egemen K. Çetinkaya, Abdul Jabbar, Justin P. Rohrer, Marcus Schöller, and Paul Smith. 2010. Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines. *Comput. Netw.* 54, 8 (June 2010), 1245–1265.

[113] G. Suciu, C. Cernat, G. Todoran, V. Suciu, V. Poenaru, T. Militaru, and S. Halunga. 2012. A solution for implementing resilience in open source Cloud platforms. In *Communications (COMM), 2012 9th International Conference on*. 335–338. https://doi.org/10.1109/ICComm.2012.6262565

[114] J. Suzuki, Y. Hidaka, J. Higuchi, Y. Hayashi, M. Kan, and T. Yoshikawa. 2016. Disaggregation and Sharing of I/O Devices in Cloud Data Centers. *IEEE Trans. Comput.* 65, 10 (Oct 2016), 3013–3026. https://doi.org/10.1109/TC.2015.2513759

[115] A. Tchana, L. Broto, and D. Hagimont. 2012. Approaches to cloud computing fault tolerance. In *2012 International Conference on Computer, Information and Telecommunication Systems (CITS)*. 1–6. https://doi.org/10.1109/CITS.2012.6220386

[116] M.H.C. Torres and T. Holvoet. 2014. Self-Adaptive Resilient Service Composition. In *Cloud and Autonomic Computing (ICCAC), 2014 International Conference on*. 141–150. https://doi.org/10.1109/ICCAC.2014.33

[117] Phuoc Nguyen Tran and Nadia Boukhatem. 2008. The Distance to the Ideal Alternative (DiA) Algorithm for Interface Selection in Heterogeneous Wireless Networks. In *Proceedings of the 6th ACM International Symposium on Mobility Management and Wireless Access (MobiWac '08)*. ACM, New York, NY, USA, 61–68.

[118] Manghui Tu and Dianxiang Xu. 2013. System resilience modeling and enhancement for the cloud. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*. 1021–1025. https://doi.org/10.1109/ICCNC.2013.6504231

[119] D. Vasconcelos, V. Severino, J. Neuman, R. Andrade, and M. Maia. 2018. Bio-Inspired Model for Data Distribution in Fog and Mist Computing. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 02. 777–782. https://doi.org/10.1109/COMPSAC.2018.10336

[120] P. Verissimo, A. Bessani, and M. Pasin. 2012. The TClouds architecture: Open and resilient cloud-of-clouds computing. In *Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on*. 1–6. https://doi.org/10.1109/DSNW.2012.6264686

[121] Alexandre Viejo and David Sánchez. 2019. Secure and privacy-preserving orchestration and delivery of fog-enabled IoT services. *Ad Hoc Networks* 82 (2019), 113 – 125. https://doi.org/10.1016/j.adhoc.2018.08.002

[122] M. Villarreal-Vasquez, B. Bhargava, P. Angin, N. Ahmed, D. Goodwin, K. Brin, and J. Kobes. 2017. An MTD-Based Self-Adaptive Resilience Approach for Cloud Systems. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. 723–726. https://doi.org/10.1109/CLOUD.2017.101

[123] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou. 2012. Toward Secure and Dependable Storage Services in Cloud Computing. *IEEE Transactions on Services Computing* 5, 2 (April 2012), 220–232. https://doi.org/10.1109/TSC.2011.24

[124] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang. 2017. A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications. *IEEE Access* 5 (2017), 6757–6779. https://doi.org/10.1109/ACCESS.2017.2685434

[125] T. Welsh and E. Benkhelifa. 2017. Perspectives on Resilience in Cloud Computing: Review and Trends. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. 696–703. https://doi.org/10.1109/AICCSA.2017.221

[126] V.R. Westmark. 2004. A definition for information system survivability. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*. 10 pp.–. https://doi.org/10.1109/HICSS.2004.1265710

[127] Xin Xu and H.H. Huang. 2015. DualVisor: Redundant Hypervisor Execution for Achieving Hardware Error Resilience in Datacenters. In *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*. 485–494. https://doi.org/10.1109/CCGrid.2015.30

[128] J. Yanez-Sierra, A. Diaz-Perez, V. Sosa-Sosa, and J. L. Gonzalez. 2015. Towards Secure and Dependable Cloud Storage Based on User-Defined Workflows. In *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*. 405–410. https://doi.org/10.1109/CSCloud.2015.28

[129] J. Yao, P. Lu, and Z. Zhu. 2014. Minimizing disaster backup window for geo-distributed multi-datacenter cloud systems. In *2014 IEEE International Conference on Communications (ICC)*. 3631–3635. https://doi.org/10.1109/ICC.2014.6883885

[130] Q. Zhang, Q. She, Y. Zhu, X. Wang, P. Palacharla, and M. Sekiya. 2013. Survivable resource orchestration for optically interconnected data center networks. In *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*. 1–3. https://doi.org/10.1049/cp.2013.1291

[131] W. Zhao, P. M. Melliar-Smith, and L. E. Moser. 2010. Fault Tolerance Middleware for Cloud Computing. In *2010 IEEE 3rd International Conference on Cloud Computing*. 67–74. https://doi.org/10.1109/CLOUD.2010.26

[132] Z. Zheng, T. C. Zhou, M. R. Lyu, and I. King. 2010. FTCloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications. In *2010 IEEE 21st International Symposium on Software Reliability Engineering*. 398–407. https://doi.org/10.1109/ISSRE.2010.28

[133] Yun Zhou, Yuguang Fang, and Yanchao Zhang. 2008. Securing wireless sensor networks: a survey. *IEEE Communications Surveys & Tutorials* 10, 3 (2008), 6–28.