Towards Cloud, Service and Tenant Classification For Cloud Computing

Sebastian Jeuk
Cisco Systems, San Jose, USA
&
Department of Computer Science
University College London
Email: ucabsej@ucl.ac.uk

Jakub Szefer
Yale University
School of Engineering & Applied Science
United States
Email: jakub.szefer@yale.edu

Shi Zhou
Department of Computer Science
University College London
Gower Street, London
United Kingdom
Email: s.zhou@ucl.ac.uk

***Abstract*—One of the major concerns cloud computing platforms face today is the lack of a unique identification of the "who" within the network infrastructure. State-of-the-art technologies (such as VLANs or IP addresses) lack functionality to cope with the highly dynamic and scalable, ever changing and virtualized cloud-enabled data center infrastructures. A shared and limited address space or the loss of identification across boundaries render classification unusable for per-tenant, per-service or per-cloud-provider policies. In this work, we introduce the concept of a classification mechanism that is fine-grained enough to associate tenants, services and cloud providers to their network streams. The Tenant-ID, Service-ID and Cloud-ID is added as a tag to Layer 3 packets throughout the consumer-to-service communication. We argue that the proposed service and tenant isolation concept is generic enough to be applicable across the whole cloud environment, thereby eliminating current limitations and enabling new network functionality.**

***Keywords*-Cloud Computing, "Identity Crisis", classification, multi-tenancy, per-tenant policy, per-service policy, tenant isolation, service isolation**

I. Introduction

Cloud Computing has emerged as a new IT paradigm over the last couple of years. It is clearly distinguished from previous service delivery models by introducing on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services. It is heavily based on virtualization, enabling ubiquitous, convenient and on-demand network access to a shared pool of compute, network and storage resources.

A key issue in Cloud Computing is the context-based classification of service and tenant assets across different boundaries. Classification is used to implement policies that define the "who, what, where, when and how". If we cannot identify and isolate the "who" on the network layer, we cannot define and implement any of the others. Current Layer 2/3/4 attributes are no longer sufficient to uniquely and in a uniform way identify the "who" in a cloud-enabled data center network. A more fine-grained classification is needed to introduce (1) visibility of services and tenants within the network, (2) control, and (3) effective management. The white-paper "Identity-aware Networking" published by the Enterprise Strategy Group (ESG) [1] states the need for identity awareness and classification on the network layer as follows: "large organizations need the network to enable any user to connect securely to applications and services from any network using any device."

In this paper, we answer the following questions: *(1) How can we classify cloud service and tenant assets in a uniform way across boundaries? (2) How can we enable visibility of services, tenants and their data in cloud data centers? (3) How can we translate business needs into network policies?*

Our Contributions:

- We introduce current classification mechanisms used in cloud environments and how they are lacking functionality to perform well in a dynamic and ever-changing environment.
- We develop a new concept that allows classification of cloud providers, their services and tenants across boundaries. We show how to perform classification on the network layer within and outside of a cloud provider.
- We present a design of a cloud enabled data center, which shows how provider, service and tenant classification enables answering the "who, what, where, when and how."

Section 2 identifies the classification crisis and defines the problem space with the help of an example. Section 3 discusses the limitations seen in current technologies and outlines the problematic for cloud environments. Section 4 defines functional limitations due to the lack of fine-grained classification. In Section 5 we introduce the identification scheme, which is followed by a discussion on benefits in Section 6. The Technical realization is outlined in Section 7 followed by a discussion, future work, and conclusion section.

II. Classification challenges in Cloud Computing

Our identity is a key element of our day-to-day lives. It is used to identify us and our assets (e.g., bank accounts, jobs, etc.) and links both closely together.

In Cloud Computing, however, our identity is decoupled from our assets. We refer to assets as components used or generated as part of a cloud service, including but not limited to VMs or traffic streams. Today, these assets cannot be easily associated to their owner's identity.

Identity, in its original sense, is used to describe users' authentication information, associated authorization and accounting details. Here, we extend the meaning of a users identity to its assets.

To better highlight the classification challenges in cloud computing we define an example around the service provider Dropbox (consumer-based cloud storage). A Dropbox consumer is any user who is registered with the Dropbox service. Dropbox chooses an Infrastructure provider like Amazon or Google. We assume that Dropbox runs its services on Amazon. This already shows one of the challenges Cloud Computing introduces. Dropbox is a service provider to its own consumers. At the same time, it is a consumer of the cloud provider Amazon.

We now introduce the concept of a tenant in a service offering. Amazon considers Dropbox a tenant of its infrastructure service. At the same time however, Dropbox offers their services to consumers. These consumers can be seen as tenants of Dropbox. With this example, it becomes clear that a tenant can exist on different levels of the cloud service offering model. To avoid confusion, we will use ”tenant” to describe an end-user of a service provider.

After introducing the concept of tenants across the different service offerings we now look at multi-tenancy. Multi-tenancy is mainly seen in a cloud provider (referred to as CP) such as Amazon or Google. It describes the shared resource utilization within the CPs infrastructure. The CP gains advantages including (1) better resource utilization, (2) rapid elasticity, and (3) on-demand self-service.

However, from a tenant's perspective, multi-tenancy is not always beneficial. A tenant of Dropbox might worry that the used infrastructure is shared with other tenants (may that be a tenant of Dropbox or Amazon itself). Multi-tenancy and its impact to tenants heavily relies on the service model used. Amazon, offering Infrastructure-as-a-Service (IaaS), uses multi-tenancy to share physical servers among many different tenants.

Here, the isolation between tenants relies on the hypervisor and the network infrastructure. Dropbox, offering Software-as-a-Service (SaaS), uses multi-tenancy on the application layer to provide access to the SaaS App to many consumers.

Within the SaaS APP, the isolation between tenants is based on user accounts within the application. DataBase isolation is often used to provide multi-tenancy in Platform-as-a-Service (PaaS) as seen in salesforce Apps.

III. Existing Technologies and Their Classification Limitations

A. *State-Of-The-Art Network IDs*

Network IDs have been used for decades in network computing to identify network assets, may it be for (1) forwarding elements or (2) end hosts. Cloud providers leverage VLANs, IP addresses, port numbers and other network, compute and storage-specific resource identifiers to address elements. For many years, the existing identifiers have been sufficient to uniquely address entities.

At first glance, it may seem that these IDs are sufficient to uniformly link identities to their assets in a multi-tenancy environment as well. This initial perception however, fails to take into account that most of the IDs have been defined for different purposes. VLANs and IPs are not designed to address entities used by many different tenants at the same time (as it is the case in a virtualized multi-tenant environment).

The address space of legacy IDs is too small to cover large cloud deployments with hundreds or thousands of tenants and services. Recent attempts to solve this (by introducing overlay solutions such as VxLANs [2]) however, fail to address the second, more important, problem. Siloed IDs identify assets that belong to (or are used by) a tenant but cannot be easily associated with the tenant.

Siloed IDs cause problems for (1) managing tenants and their assets, (2) identifying exact resource usage across network, storage and compute elements, (3) behavior discovery among tenants for eCommerce purposes, and (4) applying tenant specific policies to enable business needs.

With the introduction of Cloud Computing, network IDs have dominated discussions on how to adapt them to the virtualized, dynamic and agile infrastructure of a cloud environment. In legacy networks, matching traffic against their VLAN, IP or port affiliation often has been granular enough to accurately define business related policies. However, with Cloud Computing, simply relying on VLANs, IP addresses or port numbers as classification mechanisms, is not sufficient anymore. State-of-the-art network identifiers are now shared among many different tenants.

B. *Uniform Resource Identifier (URI)*

Based on a string of characters, a Uniform Resource Identifier (URI) [3] allows the identification of a web resource. They are distinguished by locators (Unified Resource Locator or URL) and names (Unified Resource Name or URN). The URN defines an entity's identity while the URL provides means for finding it. URIs are heavily used to share a web service within a cloud provider among many tenants. As part of HTTP, URIs are located on Layer 7 of the OSI model. Hence, classifying traffic based on URIs requires deep-packet-inspection. This typically results in high latency for the tenant. We argue that URIs are not sufficient to provide a uniform classification to enables service and tenant visibility within a network environment.

C. *HTTP - Virtual Hosting*

Based on HTTP 1.1 [4] Virtual Hosting [5] allows web-servers to host more than one website. The multiplexing can either happen on an IP or URL-name basis. This technique is seen as one of the first resource sharing technologies allowing multiple tenants to host their websites on one physical server. Here, each tenants website is either identified via a unique IP or a unique name that translates to a single IP address. Without this unique identification it would be impossible to enable this multiplexing of websites. If the multiplexing is done using different URL-names, identification of services on Layer 3 of the OSI model is impossible without performing further packet inspection.

D. *HTTPS - TLS and Server Name Indication*

Server Name Indication is used within TLS [6] based connections such as seen with HTTPS [7]. The destination URL is transmitted as part of the three-way handshake. It allows a web-server to host multiple HTTPS based website offering a certificate per side.

The unique identification of multiple HTTPS based websites hosted on the same physical server is done on a certificate basis. This approach allows the client to validate the certificate received for a specific secure website. Website specific certificates enable the end-user to trust the connection and therefore the service offered by the website.

This technology is not feasible for non-HTTPS applications. It is therefore not generic enough to be used as a classification method for services and tenants in a Cloud environment.

E. *IP Address Sharing Solutions*

IEFT RFC 6269 [8] introduces several new technologies that tackle the issue of address sharing among many consumers. This RFC introduces the solutions in regards to broadband access. However, they are also relevant to Cloud Computing and hence are briefly introduced in this section. The introduced proposals extend the address space by adding port information. RFC 6269 covers (1) Carrier Grade NAT, (2) Large Scale NAT, (3) Dual-Stack Lite, (4) NAT 64, (5) Address+Port proposals and (6) Stateless Address Mapping. For further details on each technologies refer to references .

F. *Security Group Tag*

The term “Security Group Tag” (SGT) [9] is defined as part of Cisco’s Identity product suite. SGT enables the meaningful translations of business terms into network policies. As the term suggests it classifies systems or users based on their context including (1) user role, (2) device, (3) location or (4) access method. This classification can then be used by firewalls, routers and switches to make forwarding or blocking decisions in a Data Center.

Security Group Tags are limited to the local Data Center. Hence, the grouping or classification is lost as soon as traffic leaves the network boundaries. In addition, the classification is based on context groups (e.g., departments) and is therfore not granular enough to be used on a per-tenant basis.

G. *IDs and Their Boundaries*

With the diversity of IDs, both legacy and cloud specific, it is important to understand their boundaries. This section summarizes previous ones by compiling a list of IDs, their positioning on the OSI model, the address space and their significance boundaries (refer to Table I). One or many of the shown IDs are needed to establish Layer 2 to Layer 4 connectivity to services.

IV. Classification Related Functionality Limitations

A. *Within the Cloud*

*1) **Accurate Billing**:* How are tenants billed if a CP cannot track their usage of resources within the cloud environment? The utilization of the network is currently not reflecting each tenants behavior. Hence, a CP trying to bill a tenant according to their usage of network devices such as (1) switches, (2) firewalls or (3) load-balancers is nearly impossible. Amazon defines their EC2 pricing scheme [10] on compute and storage resources used. Billing its services according to their usage of network resources is not possible as a unique link between the service and its traffic is not given. Here, this means that there is no way to determine what network resources the Amazon service Dropbox leverages to send and receive traffic.

*2) **Converting Business Needs to Security Policies**:* Security is a hot topic for every company. In a cloud environment, security is even more critical as all resources are shared amongst many different tenant’s (often even between competitors). Currently, CPs are lacking the right tools and functionality to define specific security policies for tenants or service assets. This is mainly caused by the disconnect of tenant or service identities and their assets within the network. In our example, security issues arise if Dropbox as an IaaS customer of Amazon wants to apply per-tenant security policies. Dropbox currently has no means to distinguish its tenants on a network or transport layer. Hence, applying inline security policies per-tenant is not possible.

*3) **Per-tenant QoS**:* Classification for Quality of Service is based on either IntServ or DiffServ [11]. Both methods introduce arbitrary characteristics to distinguish traffic flows. This classification is often only based on the type of traffic but not its tenant or service affiliation.

ID	OSI Model Layer	Address Space	Boundary
VLAN	Data Link	12 bit	Valid within Layer 2 domain. Not carried across layer 3. Not unique per tenant nor service.
MAC	Data Link	48 bit	Valid within layer 2 segment. Unique but configurable in virtualized environment.
Security Group Tag	Data Link Layer or independent protocol	16 bit	Cisco proprietary, within Data Center
VxLAN	Layer 2.5	24 bit	Within data center.
IPv4	Network	32 bit	Non RFC-1918 addresses globally valid. Depending on translation configured. Not unique per service.
IPv6	Network	128 bit	Different address types with both local or global significance. Not unique per service.
TCP/UDP port	Transport	16 bit	Transmitted as part of TCP/UDP header. Seen end-to-end between sender and receiver. Not unique per service.
Service User account ID	Application	protocol dependent	Encrypted in transit. Only seen on application layer

Table I: **Layer 2 - 7 IDs:** In a typical network many different identifiers are used to separate and identify traffic. In a cloud environment those IDs are reused multiple times to cope with virtualized layers. This table introduces the different IDs seen in a cloud environment. We show their address space and their boundaries.

4) ***Resource Orchestration Complexity****:* Cloud management is getting more and more critical in orchestrating services and tenants within minimal time frames. Orchestrating a whole service within minutes requires the handling of many different resources and spinning up of assets. Often, management tools define network, compute and storage resources without a uniform link between the new tenant or service and the needed resources. This can be critical for both Amazon as the CP but also for Dropbox as the service running on top of Amazons IaaS infrastructure.

5) ***Service Stacking****:* In our example we introduced Amazon as the underlying IaaS Service Provider. Dropbox, as a tenant of Amazon, is a SaaS provider to its own customers. This two layer service delivery approach is a typical scenario for CPs and virtual Service Providers leveraging the CPs infrastructure to offer SaaS or PaaS services. We refer to this phenomenon in cloud environments as “Service Stacking”.

This and the unique identification of services is seen as a challenge in Cloud Computing for tenant & service identification. Running a SaaS over a PaaS over a IaaS service obfuscates services in the network. Both SaaS and PaaS services appear as the underlying IaaS service on the lower layers of the OSI model. In our example, Dropbox is obfuscated by Amazons underlying IaaS environment. Without the use of deep-packet inspection and the use of Layer 7 application data, services cannot be distinguished within the lower layers (2-4).

6) ***Network, Compute and Storage Resource Abstraction****:* Resource abstraction from physical hardware is one of the underlying concepts that enables auto-scaling and proactive resource adjustment. Abstracting physical hardware enables virtualization and brings several advantages to a cloud architecture. However, the network, compute and storage resource abstraction also causes problems in usage identification and traffic isolation.

7) ***Virtual Service Providers****:* Our previous example introduced Dropbox as a service provider offering consumer storage capabilities and therefore can be referred to as a Virtual Service Provider (VSP). A VSP relies on a CP to provide the network infrastructure, Data Center facilities, power and everything related to the physical implementation.

This concept originated in the telecommunication industry where a Internet Service Provider (ISP) would rent out its infrastructure to a VSP. In the telecommunication world the identification of the tenants and its assets is standardized by using a 10-digit unique number called mobile subscription identification number (MSIN). That number, being part of the international mobile subscriber identity (IMSI), allows tracking a mobile phone uniquely within both the ISPs and the VSPs environment.

However, in a Cloud Environment this concept is not available. The VSP much less the CP are able to uniquely track the behavior of a tenant or service and their related assets. This increases the problematic of identifying services both within and outside of a cloud infrastructure.

In a remote organizations network (e.g., Cisco Systems) Dropbox traffic is identified as part of Amazon without inspecting each packet more thoroughly. This lack of tenant/service isolation introduces challenges for organizations remote to the CP. Here, Cisco Systems would not be able to identify traffic from Dropbox without looking at each packets Layer 7 information. This deep packet inspection causes high latency as its performed in software and is therefore not viable on a large scale.

*8) **Complex Compute Infrastructures**:* Compute infrastructures can be highly complex with many different layers of configuration. Identifying services or tenants and their assets in those environments is critical to (1) simplify management, (2) enable resource tracking on compute infrastructures, and (3) ease troubleshooting tenant or service related issues. Based on Cisco Systems Unified Computing System (UCS) we demonstrate challenges seen with disconnected service or tenant identities and related assets. Dropbox serves as the service in this scenario. A service, especially IaaS services, typically have (1) VMs, (2) virtual Routers, (3) virtual service appliances, and (4) sometimes whole blade servers dedicated to their compute infrastructure. The resources needed or used depends on the complexity and the utilization of the service. We assume that Dropbox has VMs on a shared blade server within the UCS system. Those VMs, deployed during the initialization of the service, often have random and arbitrary names.

*B. **Outside The Cloud***

In addition to challenges a CP is facing, Internet Service Providers and remote network operators currently also lack service or tenant awareness and classification within the network layer. As a result an ISP cannot define fine-grained routing decisions as often demanded to reflect cloud service needs. Today, it is not possible to forward certain services across links with better characteristics for a particular cloud service. In our example that could mean that Dropbox is send across links that are also used for low latency voice and video traffic. Hence, both Dropbox and any other service on that link are affected by higher latencies and therefore decreased user experience.

A network operator, e.g. a remote organization such as Cisco System, currently has limited means to filter particular services from one CP without filtering all services. Amazon is hosting a service that is violating Cisco Systems security policies. Cisco needs to be able to block this particular service while allowing other services such as Dropbox.

We can clearly see that without a unique identification and classification of tenants or services and their assets, major business requirements are not easily met.

V. The Identification Scheme

This paper introduces three IDs, the (1) Cloud-ID, (2) Service-ID, and (3) Tenant-ID. Each ID is used to classify traffic according to its affiliation with cloud providers, services and tenants. We propose a 16 byte IPv4 or IPv6 header increase with 4 bytes being reserved for the cloud provider, 6 bytes for its services and 6 bytes for the consumers.

*A. **Cloud-ID***

The C-ID is intended to be used to uniquely identify a cloud provider globally. Similar to the domain name registration, we propose that multiple third parties will allocate IDs to cloud providers and make such information public. We propose a 4-byte (32-bit) number for the C-ID. It consists of two parts. (1) The first part is a 10-bits registry number, which encodes the registry service that assigns C-IDs to cloud providers in its region. (2) The second part is a 22-bit cloud number, which identifies a cloud provider.

Registry Location	ID Value (Numeric - 10 bit)
USA	0001
GB	0002
DE	0003
AUS	0004

Table II: **Registry Location ID:** The first 10 bit of the Cloud-ID are used to identify the location of the registry service used to register a CP in a certain location. One CP can have multiple ”Registry Location IDs” to reflect its location.

The registry service is responsible for assuring uniqueness, security in terms of Provider authentication and transparency of easy lookup of C-ID values. Suppose a registry service maintains a database of cloud providers that are geographically located within its service area, then the registry number can provide useful information on the region where a cloud is physically based. To demonstrate the concept we identify several registries and corresponding C-IDs.

Cloud Provider	ID Value (Numeric - 22 bit)
Amazon	1310010
Google	1310020
Microsoft	1310030
Nirvanix	1310040

Table III: **CP-ID:** The cloud provider ID needs to be unique per Cloud provider. It is assigned per provider and unique across different registry locations.

After defining the registry values for the first part of the C-ID we provide an example of several cloud providers identified within one location. For the purpose of this paper we selected registry location USA with the registry value of 001. The cloud provider part of the C-ID consists of 32 bits with more than 4 billion possible IDs. The cloud provider part of the C-ID is chosen and managed by the corresponding registry authority. The C-ID, as a combination of the registry and the cloud provider ID is represented as follows (based on the examples shown above).

Cloud-ID	ID Value (Numeric - 48 bit)
Amazon	0001.1310010
Google	0001.1310020
Microsoft	0001.1310030
Nirvanix	0001.1310040

Table IV: **Overall Cloud-ID:** The Cloud-ID is defined as a combination of the registry location ID and the actual CP ID.

A cloud provider can register with multiple registry services depending on different locations of its data centers. Ideally the cloud provider number is kept the same whereas the registry number changes. For example, Amazon can have two C-IDs, 0001.1310010 and 0002.1310010, reflecting its cloud representation in the US and in Great Britain respectively.

B. Service-ID

The S-ID is proposed to identify cloud-based services from remote entities. It is managed and controlled by the cloud provider and therefore does not require a public registry service. Uniqueness is critical only within the cloud providers environment making the usage of the S-ID more straight forward. Similar to the C-ID, we propose a more sophisticated format for the S-ID than just a plain number. Having a pre-defined numbering scheme has the advantage of incorporating information into the ID.

The Service ID is defined as a 6-byte numeric string. For the purpose of the S-ID, we propose several sub-IDs defining the following information.

Sub-ID	Bits
Data Center within CP location	8 bit value
Service	32 bit value
Option	8 bit value

Table V: **Service-ID sub-IDs:** The service ID consists of three sub-IDs defining the Data Center location, the Service itself and optional values relevant to the service.

The above values are used to demonstrate the concept of the three sub-IDs identified for the S-ID. The Service is identified by a 15-bit number. Another 4 bits are reserved for further information or to extend the 15-bit service number if needed.

To show how the Service sub-IDs can be built, we provide some examples. The first table shows three example locations within the United States. These IDs add granularity per Data Center within a certain country location.

Data Center within location	ID Value (Numeric - 8 bit)
San Francisco	001
Boston	002
New York	003

Table VI: **Data Center Location:** The location of the Data Center used for the particular service is encoded using a 8 bit numerical value. One Service can have multiple locations.

Below we define the service value. For the purpose of this paper, we leverage Amazon as an cloud provider. The values are selected randomly.

Service	ID Value (numeric - 32 bit)
Dropbox	25352
S3	23948
Salesforce	00617

Table VII: **Service value:** The Service-ID incorporates a 32-bit numerical value that is used as the unique identifier for the Service itself.

A cloud provider internal database is necessary to manage service values and assure their uniqueness. After defining both sub-IDs we can now show the full S-ID. In combination with the C-ID the S-ID and its sub-IDs allow specifying the location on a per-Data Center basis within a certain country.

For the purpose of this paper we assume that the optional 8-bit value is left blank, and is therefore showing a 000. The table below shows S-IDs for three different services hosted by the Amazon Cloud.

Final Service-ID	ID Value (numeric - 6 byte)
Dropbox	001.2535200000.000
S3	002.2394800000.000
Salesforce	001.0061700000.000

Table VIII: **Overall Service-ID:** The Service-ID consists of three sub-IDs defining its location, a unique identifier and options relevant to the service. The option field is defined within a CP to support meta data for a particular service.

C. Tenant-ID

We proposed a Tenant-ID on Layer 2 to allow the unique classification of tenant specific traffic within VLANs and Layer 2 segments. To leverage this classification for Layer 3 mechanisms we port the Layer 2 ID to Layer 3 and add further metadata. The Tenant-ID on Layer 2 uses a 15-bit address space. We propose to extend this address space to 48-bits on Layer 3. The extended address space allows for even more granular classification based on metadata describing the tenant within a cloud environment. The overall structure of the Layer 3 Tenant-ID is shown below:

Sub-ID	Bits
Layer 2 Tenant-ID	16 bit value
Metadata	32 bit value

Table IX: **Tenant-ID sub-IDs:** The Layer 3 Tenant-ID is composed of two sub-IDs. The first part describes the Layer 2 equivalent that is used to classify tenants within a LAN segment. The second part is an 32-bit optional field that allows the Cloud Provider to add metadata to the ID.

The metadata is defined by the Cloud Provider and is therefore not restricted. To classify traffic per tenant the whole Layer 3 Tenant-ID is used. We argue that the meta data added by the Cloud Provider has only local significance. Outside of the Cloud Providers network the whole 48 bit are considered to be the identifier for a certain Tenant. External networks are not able to decode meta data incorporated into the ID. Adding meta data to the Tenant-ID on Layer 3 enables the Cloud Provider with further classification capabilities.

D. *Scheme Characteristics*

1) ***Hierarchical Relation Between IDs****:* The identification scheme is based on a hierarchical approach. It is used to show the relationship between different IDs. Each ID has a direct relationship with another ID, may it be as a child or as a parent.

A cloud provider only receives a single global ID. This ID is used as the root for all following Service-IDs and Tenant-IDs. One Cloud-ID can have one-to-many Service-IDs.

On the next Layer, each Service-ID is seen as the root for the Tenant-IDs. Multiple Tenant-IDs are linked to one Service-ID. As a tenant can leverage the offerings of multiple services the relationship between Tenant-IDs and Service-IDs can be many-to-many.

2) ***Identity Registration and Assignment****:* The registration of the IDs is a critical component of the overall scheme. The following two bullet points outline the Cloud-ID registry service and how the Cloud Provider is assigning local IDs.

- **Global Registration for Cloud-ID**: We propose an approach that is borrowed from registering domain names. Third party authorities are responsible for assuring uniqueness in providing domain names on a per-top-level domain basis. For the purpose of the C-ID registration a 3rd party registry is needed to distribute IDs and to assure their uniqueness. We argue that a simple registration per cloud provider on a secured website is sufficient to distribute C-IDs.
- **Local Assignment for Service-ID**: The S-ID and the T-ID on the other hand are solely significant within the cloud providers environment. Hence, they can be managed by the managed and maintained by the cloud management systems. It is the combination of the above three cloud identities that ensures a globally unique identification of a cloud computing entity (in addition to IP address and port number).

3) ***Flexibility to Enable Service-Stacking****:* Service stacking is an important characteristic in Cloud environments. More and more services run on top of other services, which run on IaaS providers. We want to assure that classification happens for each layer of Services stacked on top of each other. Hence, a method is required that enables adding multiple Service-IDs to the Cloud information. By using IPv6 and extension headers we easily can stack multiple Service-IDs. Therefore the Service-ID part of the IPv6 Cloud extension header can contain one or many Service IDs. This approach allows us to define very granular policies.

4) ***End User Control Over Classification****:* As part of the new Cloud classification scheme we suggest that the end user should be able to request certain IDs. This provides the most control to the Tenant, while still providing the benefits to other stakeholders (e.g., cloud provider or network operators).

Having said that, giving the end-user full control over the classification limits the benefits to those stakeholders dramatically. We therefore propose that within a cloud provider all IDs are used by default (even if the end-user requested no classification). The end user only controls if those IDs are stripped of when the packets leave the cloud provider. This way, the cloud provider can leverage the classification while the user keeps control over the identification outside the cloud provider.

VI. Technical Realization of Cloud-, Service- & Tenant-ID

We defined three novel identities. The first to uniquely identify cloud providers globally. The second to uniquely distinguish services locally in a given cloud. The two together form a unique global identity for a cloud service. Optionally, a tenant identity can be added to allow traffic separation per cloud tenant.

We propose to incorporate the cloud identities in the header of a Layer 3 packet. Both the IPv4 and the IPv6 protocols provide ways to carry additional information in packet header(s). Our proposal strictly follows the Layer 3 protocol specifications to ensure that it is fully compatible with the existing network systems. This not only makes our proposal feasible and practical, but also reduces the implementation cost to a minimum.

We focus our implementation example on IPv6 as the soon-to-be replacement for IPv4. IPv6 is more flexible, as it supports extension headers, and therefore, can be adjusted as needed. For backward compatibility we introduce a sample implementation for IPv4. This implementation is limited to a single C-ID, S-ID and T-ID, and therefore, does not support future-proof service stacking.

The implementations for both the IPv6 and IPv4 protocols are examples and not seen as the only viable implementation solution.

A. *Classification on Layer 2*

For cloud internal tenant traffic identification we published [12] a conceptual idea introducing a Tenant-ID on Layer 2. This identity is incorporated into a modified 802.1q header. The ID is 15 bits in size and has pure cloud provider internal significance.

The T-ID on Layer 2 is used in combination with VLAN-IDs. This way a tenant can be uniquely identified while allowing the segmentation of traffic on a per tenant basis. We argue that the Layer 2 T-ID is more than just a VLAN-ID extension as multiple VLANs can be associated to one Tenant. This allows sub-segmentation of traffic per T-ID.

This approach allows us to maintain the hierarchical ID scheme proposed in this paper. One Tenant can be associated to multiple VLANs but not vice versa. Hence, we assure that traffic within a Cloud Provider is fully segregated between tenants.

The Tenant-ID can exist on both Layer 2 and Layer 3, either for traffic within a cloud environment or for traffic leaving the cloud providers network, respectively.

B. *Cloud Header Fields*

The following header fields are defined to enable Cloud classification.

- The **Copied field** (CP, 1 bit, value=1) specifies the need to copy the option fields in all fragments.
- The **Class field** (2 bits, value=0) defines a general option category. Included for future expansion.
- The **Number** (5 bits, value=10) field specifies the ID number "10" of the proposed CLOUD option fields.
- The **Length** (8 bits, value=64) field defines the total number of bits of the subsequent data fields.
- We propose a **Cloud flag** (3 bit) field that allows the client to indicate interest in CLOUD identities. The most significant bit identifies the C-ID, the next bit is used to request the S-ID, and the least significant bit is used to set the T-ID. Here, we propose that the client controls what IDs are added to a packet. Therefore the provider is passive and only fulfills the request of the client.

  The most significant bit indicates whether the C-ID is present in the optional data fields. The next two bits trigger the S-ID and the T-ID respectively.
- The **Data fields** contain all three CLOUD identities.

We propose the following implementation for both IPv6 and IPv4.

C. *IPv6*

The IPv6 [13] does not allow the addition of optional fields to the basic header, but it has been designed with built-in flexibility to add extension header where necessary. This makes the incorporation of the cloud identities into IPv6 packets straightforward - we just add a new extension header.

The IPv6 characteristic of allowing extension headers also enables one of our proposed features. The service stacking in cloud environments is as of today mainly limited to services and tenants. However, in the future that might change whereby services are running on top of services with multiple Service-IDs stacked on top of each other. Extension headers are the ideal way to support a higher degree of service stacking in the future. Our proposed extension header is illustrated in Fig. 2. The IPv6 header allows cascading multiple extension headers. This means that our proposal will not interfere with the functionality of the IPv6 protocol and it will not influence the upper-layer protocols. The IPv6 header allows cascading multiple extension headers. This means that our proposal is not interfering with the functionality of the protocol nor does it influence the upper-layer protocols.

As we focus our research and prototype development on IPv6, we do not provide further details on the example implementation for IPv4 networks.

VII. Benefits

A. *Remote Cloud Service Identification*

The main advantage in adding global identities to Layer 3 information is seen by tagging service specific traffic flows. The proposed solution is versatile enough to support any kind of services that are based on Layer 3 information. Adding a third identification mechanism to the already existing IP addresses and corresponding port numbers adds granularity in matching and policing traffic.

B. *Identity-Based Security*

The proposed solution introduces identities into Layer 3 packet headers that allow fine-grained policy matching according to cloud service affiliations of traffic flows. This advantage can be leveraged to define security rules on a per-service basis.

C. *Transparency and Backwards Compatibility*

The identities introduced on Layer 3 are transparent to both upper and lower level protocols. This characteristic allows a step-by-step introduction to existing networks with an easy to maintain backwards compatibility.

D. *Cloud Service Location Awareness*

Based on the proposed ID structure the identification mechanism enables location awareness to traffic tagged for a specific service. This is seen as an advantage for remote entities to determine where data/information is accessed and stored in the Cloud. In addition, service traffic to countries in violation of certain data privacy regulations can be blocked based on the location part of the ID.

E. *Cloud-ID/Service-ID Spoofing Resistant*

The C-ID and the S-ID are added dynamically to traffic flows. As with IP address on Layer 3 C-IDs and S-IDs pose the risk of being manipulated by rogue entities between the sender and the receiver. With the introduction of the C-ID registry service and the cross-checking of IDs on the providers site it can be verified if the ID has been tempered with. Packets with tempered IDs could be marked malformed on the cloud providers end terminating the traffic flow completely.

VIII. Discussion

A. *Possible Downsides*

*1) **Performance Impact**:* Increasing the size of the IPv6 packet (due to extension header) might decrease the performance of forwarding packets inside the cloud provider and the enterprise. The forwarding across Internet Service Provider networks should not be affected unless the ISP bases the forwarding decisions on the Cloud- or Service-ID. Another performance drop might be seen by processing packets by policies trying to match against particular services. Here, we argue that the benefit of matching the traffic for security, forwarding and billing purposes outranks a slight performance drop.

B. *Compatibility*

The framework proposed in this paper is fully compatible with protocols within and outside the cloud provider network environment. Leveraging the elastic option field or an additional extension header in IPv4 or IPv6 protocols respectively eliminates the need for hardware or software alteration. With the added benefit of full compatibility and transparency, deploying the identities is voluntary. Leveraging the proposed framework enables certain functions, and therefore enriches a cloud provider's portfolio. The benefit increases proportionally with the adoption of the identities. Those who do not adopt the solution will operate as normal, keeping the interoperability with C-ID/S-ID/T-ID enabled systems.

IX. Related Work

The Enterprise Strategy Group (ESG), on behalf of Extreme Networks, published a white paper discussing the needs of identity-awareness at the network layer. They argue that identity at the network is required (1) to meet new business requirements, (2) legacy networks cannot meet business needs, and (3) enterprises require identity-aware networks.

ESG research suggest that the business, regulatory compliance, and security ROI delivered by an identity-aware network eventually triggers the replacement of legacy networks. The research underdone by ESG is closely aligned to this paper's research topic. Their white paper only outlines the problem space and the need for identity-aware networks; they do not introduce a solution to this problem.

As part of this research, we recently published a conceptual paper introducing the T-ID on Layer 2. This ID has local relevance to a cloud provider and is crucial in identifying tenant assets within the cloud network environment. Refer to reference [12] for further details.

Khoudali et al. [14] and Benzidane et al. [15] propose a Frame Tag header. Added on Layer 3 at the beginning of the payload, it identifies tenants and the services used. By using md5 hash values, the overall packet size increases by 32 bytes. We argue that the proposals by Khoudali et al. [14] and Benzidane et al. [15] have several limitations.

- The frame-tag for inter-VM traffic inspection is added on Layer 3. Inter-VM traffic in a cloud data center, however, is predominantly happening on Layer 2. Devices in between the VMs are therefore ignoring information above Layer 2. Forcing traffic across Layer 3 domains dramatically increases the bandwidth requirements, and therefore, renders the solution unpractical.
- The approach both papers suggest requires an alteration in switching hardware and software, making it incompatible with existing devices.
- The proposed md5 hash values increase the overall packet size by at least 32 bytes. In comparison, a typical IPv4 header is between 20 to 60 bytes large. Adding 32 bytes dramatically increases the processing overhead, causing delays in a delay sensitive network area.

We can conclude that Khoudalis et al. [14] and Benzidanes et al. [15] research point out the need for a unique tenant and service identities in a cloud environment. However, the proposed solution has several limitations that make it unpractical and not feasible for a cloud environment. Therefore, we have no doubt that their proposed solution will not solve the problem.

Content-Centric Networking (CCN) [16] [17] [18] is closely aligned to our research topic. CCNs approach replaces the "Where" information in a packet with "What" information. This way not the Host itself is identified per packet but the content or data residing on the host. One of the major problems in Content-Centric Networking is the need for a very large routing table that cannot be handled by current routing protocols in the Internet.

X. Future Work

We plan to extend our proposal by developing a prototyping. The prototype will be based Cisco products to assure the most realistic results in a prototype environment. The prototype will be used to evaluate potential performance impacts, show its feasibility and usability in a non-simulated network.

To extend the classification proposal on Layer 2 of the OSI model, we suggest to incorporate "Layer 2 extension headers." They could be used in a similar way to IPv6 extension headers, allowing the optional and stacked addition of Cloud classification information on Layer 2.

Currently, all IDs are visible to all stakeholders within end-to-end connections. We argue that this is not secure enough and might raise privacy concerns. We plan to investigate how certain IDs can be hidden to the different parties in a Cloud environment (i.e., hiding the tenant ID to the underlying IaaS provider).

XI. Conclusion

With the introduction of Cloud Computing, organizations need the network to enable full visibility and transparency of who is using what, where, when and how. Mechanisms are required to translate business needs to network layer policies for accurate billing, security aspects or complex resource orchestration. With the lack of adequate and granular classifications, current technology limitations are magnified and new cloud specific functionalities are restricted.

Previous research has shown that fine-grained classification and unique identification has been an issue for a long time and is getting more and more important with the introduction of virtualization. Multiple layers of virtualized hardware resources, service stacking and on-demand elastic services highlight issues with current technologies such as address space limitations.

In this paper, we introduce a new classification scheme to identify Cloud Providers, their services and tenants both within and outside of a cloud environment. The identifiers, overall size of 16 bytes, are added as an extension header in IPv6 or as part of the option field in IPv4. The key characteristic of the new classification approach can be summarized as follows:

- validity across boundaries (both Layer 2 and Layer 3)
- enablement of new functions such as accurate billing or per service/tenant pair security policies
- visibility and transparency on a per-service/per-tenant basis within a cloud environment

In conclusion, we state that fine-grained classification is key to enable end-to-end service and tenant isolation. The proposed concept is generic enough to be used for other highly-demanded functions within a cloud environment while eliminating limitations of legacy technologies.

References

[1] J. Oltsik, "Identity-Aware Networking," Enterprise Strategy Group, Tech. Rep., 11 2010.

[2] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "Vxlan: A framework for overlaying virtualized layer 2 networks over layer 3 networks," United States, 2014, see also http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-08.

[3] T. Berners-Lee, "RFC 2396: Uniform Resource Identifiers (URI)," MIT, Tech. Rep., 1998. [Online]. Available: http://www.rfc-archive.org/getrfc.php?rfc=2396

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616 (Draft Standard), Internet Engineering Task Force, Jun. 1999, updated by RFCs 2817, 5785, 6266, 6585. [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt

[5] R. Khare and S. Lawrence, "Upgrading to TLS Within HTTP/1.1," RFC 2817 (Proposed Standard), Internet Engineering Task Force, May 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2817.txt

[6] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright, "Transport Layer Security (TLS) Extensions," RFC 3546 (Proposed Standard), Tech. Rep. 3546, June 2003, obsoleted by RFC 4366. [Online]. Available: http://www.ietf.org/rfc/rfc3546.txt

[7] E. Rescorla, "HTTP Over TLS," RFC 2818 (Informational), Internet Engineering Task Force, May 2000, updated by RFC 5785. [Online]. Available: http://www.ietf.org/rfc/rfc2818.txt

[8] M. Ford, M. Boucadair, A. Durand, P. Levis, and P. Roberts, "Issues with IP Address Sharing," RFC 6269 (Informational), Internet Engineering Task Force, Jun. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6269.txt

[9] D. Miller, "Advanced Security Group Tags: The Detailed Walk Through," Cisco Systems Inc., Tech. Rep., 7 2013.

[10] (2014) Amazon ec2 pricing. [Online]. Available: http://aws.amazon.com/ec2/pricing/

[11] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474 (Proposed Standard), Internet Engineering Task Force, Dec. 1998, updated by RFCs 3168, 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2474.txt

[12] S. Jeuk and S. Zhou, "Tenant-id: Tagging tenant assets in cloud environments," in *Proceedings of 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Computer Society, 2013.

[13] S. Deering and R. Hinden, "Internet protocol, version 6 (ipv6) specification," United States, 1998.

[14] S. Khoudali, K. Benzidane, and A. Sekkaki, "Inter-vm packet inspection in cloud computing," in *Communications, Computers and Applications (MIC-CCA), 2012 Mosharaka International Conference on*, 2012, pp. 84–89.

[15] K. Benzidane, S. Khoudali, and A. Sekkaki, "Autonomous agent-based inspection for inter-vm traffic in a cloud environment," in *Internet Technology And Secured Transactions, 2012 International Conferece For*, 2012, pp. 656–661.

[16] S. Hong, M. wuk Jang, and B.-J. Lee, "Ccn networking architecture for mobile applications," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, 2013, pp. 609–612.

[17] G. Rossini and D. Rossi, "A dive into the caching performance of content centric networking," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2012 IEEE 17th International Workshop on*, 2012, pp. 105–109.

[18] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.