Logical Data Models for Cloud Computing Architectures

Augustine (Gus) Samba, *Kent State University*

Describing generic logical data models for two existing cloud computing architectures, the author helps develop a common set of architectural requirements to facilitate traceability between evolving business requirements and cloud architecture implementations.

The cloud computing model for delivering IT services via the Internet enhances collaboration, agility, scalability, and availability for end users and enterprises. This optimized and efficient computing platform is provided through a technology infrastructure that's often virtualized and that lets applications, data storage, processing power, and network resources be easily provisioned, managed, and secured remotely over public or private networks or the Internet.

The cloud model comprises five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured services. It also offers three service models—software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS)—and four deployment models—public, private, community, or hybrid. The cloud architecture must illustrate the platform and software components, middleware, cloud resources and services, and communication protocols. It should also represent cloud operations and management, cloud security, and interactions between the components and with external entities.

Developing efficient cloud architectures isn't a trivial task.[1] A key challenge is ensuring that the architecture provides an effective foundation for evolving cloud service offerings. Otherwise, an efficient architecture could quickly become outdated in a cloud computing environment dominated by next-generation service offerings.

Here, I examine existing cloud computing architectures and present generic logical data models (LDMs), independent of implementations, for two existing architectures: the National Institute of Standard Technology (NIST; www.nist.org) cloud and the Distributed Management Task Force (DMTF) cloud.[2] The data models provide a framework for developing a common set of requirements for cloud architectures.

1520-9202/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

Table 1. Cloud architecture models and their key focus areas.

Cloud architecture models	Key architectural focus areas	Resources
Amazon EC2	Simple storage and relational database services for enterprises	http://aws.amazon.com
Cloud Security Alliance	Enterprise security	https://cloudsecurityalliance.org/research/initiatives/security-guidance/
Cisco	IT data centers and networks	www.cisco.com/en/US/netsol/index.html
Distributed Management Task Force (DMTF)	Working groups developing standards, such as interoperability	http://dmtf.org/
Elastra	IT enterprise management	www.elastra.com
General Services Administration	Federal government cloud service needs	www.gsa.gov/portal/content/159101
IBM	Cloud service management	www.ibm.com/cloud
Juniper	IT data centers and networks	www.juniper.net/in/en/solutions/enterprise/data-center/
National Institute of Standard and Technology (NIST)	Cloud computing standards development	www.nist.gov/itl/cloud/index.cfm
Storage Networking Industry Association	IT cloud storage	www.snia.org
Windows Azure	Microsoft data center applications	www.microsoft.com/windowsazure

Overview of Cloud Computing Architectures

Several organizations, including NIST and other government agencies, have proposed different cloud architectures for large platform enterprises. The architectures scale with traditional large-platform IT solutions. However, they might not effectively scale with rapidly evolving computing requirements of corporations and organizations, such as real-time video communication service offerings.

Cloud architecture requirements for large platforms designed for heterogeneous service types tend to be very different from requirements for application-specific platforms, such as cloud security platforms. Table 1 illustrates several cloud architecture models and their key focus areas. Each area might have one or more unique characteristics or constraints. In general, a given cloud architecture model should overcome the set of unique characteristics.

There are other cloud architecture models. Here, I focus on cloud architectures proposed by two cloud computing standard organizations: DMTF and NIST.

DMTF Reference Architecture

The DMTF cloud architecture consists of three primary actors, or organizations, and a set of interfaces that uniquely describe how the actors interact to meet cloud service objectives. The architecture supports three types of cloud services—SaaS, PaaS, and IaaS—and the following service interactions.[2]

Actors. The *cloud service consumer* represents the end user—that is, an organization or an enterprise that subscribes to one or more cloud services. It interacts with the cloud services (for example, applications and administrative functions for managing cloud storage) via user and programming interfaces. The service consumer isn't expected to know the details of the cloud computing infrastructure. On the managerial side, however, the consumer must negotiate service-level agreements (SLAs) and contract details with the service provider.

The *cloud service provider* delivers services to consumers. The scope of responsibility varies depending on the type of cloud service offering. In the SaaS domain, the cloud service provider should install, manage, and maintain SaaS software in the cloud. In the PaaS domain, the service provider should manage the cloud infrastructure platform for consumer applications. The service provider responsibilities in the IaaS domain include maintaining the storage, database, message queue, or other hosting environments for virtual machines.

Additionally, the DMTF architecture expects the service provider to meter the cloud service usage by determining who uses the cloud and to

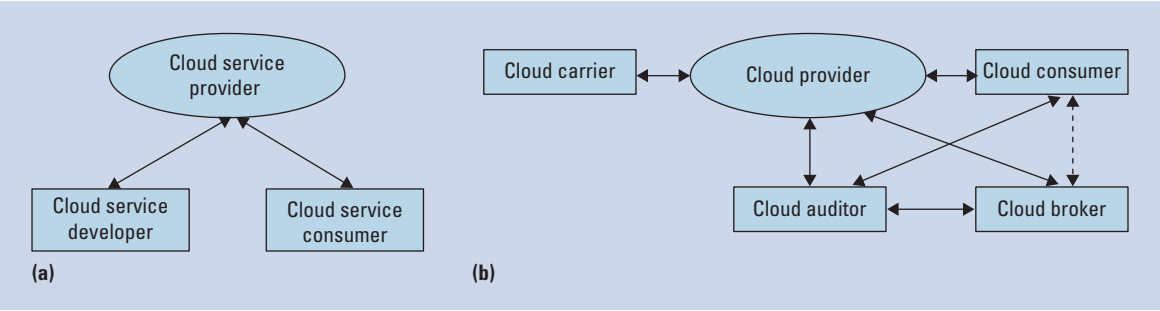

Figure 1. High-level interactions between actors in the (a) Distributed Management Task Force (DMTF) cloud and (b) National Institute of Standards and Technology (NIST) cloud. The solid lines indicate required interactions and the dashed line indicates an optional interaction.

what extent; provisioning consumer applications in the cloud; monitoring performance, security, and usage billing; and managing the SLA.

The *cloud service developer* designs and implements service components, creating services using DMTF service templates, which are then provisioned or installed in the cloud. The service provider then validates and customizes the services based on the requirements.

Provider interface and data artifacts. The DMTF reference architecture includes a provider interface layer, which specifies how the cloud service developer and consumer interact with the cloud service provider. This architecture differentiates between service endpoints that accept (and respond to) messages over a protocol based on some message exchange pattern and the data elements and operations that an interface can support.

DMTF profiles. These profiles represent extensions of the provider interfaces and artifacts, or combinations of them. The profiles provide the means to handle special cases, such as those of interest to a security manager or a contract billing administrator.

Figure 1a illustrates high-level interactions between the actors to provide the cloud services in the DMTF framework.

The NIST Cloud Reference Architecture

The NIST cloud reference architecture consists of five major actors or organizations and three types of cloud services. Each actor performs a set of assigned tasks and interacts with other actors to provide, maintain, and manage cloud services. The *cloud consumer* represents an individual or organization that can request services from one or more *cloud providers* using a *cloud broker* as an intermediary. The architecture supports three types of consumers consistent with the three types of service:

- SaaS consumers request applications;
- PaaS consumers develop, test, deploy, and manage applications hosted in the cloud; and
- IaaS consumers install, manage, and monitor services.

The cloud broker represents an individual or organization that manages the use, performance, and secured delivery of cloud services to consumers. It negotiates relationships between cloud providers and consumers and can customize services by aggregating multiple services based on consumer requirements.

A *cloud carrier* represents an organization that provides an access network to the cloud infrastructure for hardware and storage devices. It also ensures consistent SLAs for cloud service offerings.

The cloud provider represents an organization responsible for making a service available to cloud consumers. A cloud provider sets up SLAs with cloud carriers and cloud consumers. Additional functions include

- deploying the cloud infrastructure in one of four models: private, public, community or hybrid;
- managing the cloud infrastructure—that is, coordinating and managing virtual machines, virtual data storage, hypervisors, hardware resources, and service applications, and supporting network management tools;
- managing cloud services—that is, supporting external interfaces and client applications to help manage accounts, customers, contracts, inventory, and SLAs, and supporting

provisioning, metering, migration, portability, and security; and
- managing the network, which entails fault, configuration, accounting, performance, and security management.

Finally, the *cloud auditor* represents an individual or organization that performs independent assessments of the cloud provider's services, information systems operations, performance, and security on behalf of the cloud consumer.

Figure 1b illustrates high-level interactions between the major actors in the NIST framework for providing cloud services. Communication between the cloud consumer and cloud broker is optional. The cloud carrier interacts primarily with the cloud provider, while the other associations could be classified as one-to-many.

Logical Data Models

Here, I describe LDMs for the DMTF and NIST cloud service offerings using entity-relationship diagrams (E-RD).[3–5] To keep the discussions simple, I use simplified, standard notations to identify relationships between entities. In particular, lines (connectors) depict relationships between two connecting entities, and the end points indicate the cardinality. In this simplified ER-D modeling, rectangular blocks illustrate entity types. An entity type is classified as a dependent entity if its attributes are based on its relationship with a connecting entity. The cloud service provider domain is represented in each figure as shaded entities. Entities in other domains are illustrated with bold line edges.

Cloud models for large enterprises, including public clouds, typically generate a vast amount of data items. It can require multiple tools to effectively analyze the data items, and subject matter experts might need to provide a coherent view of the analysis.

The LDM describes information that must be retained to meet the business requirements for operating and managing cloud service instances. The model illustrates the logical interactions and relationships between the key entities that define a given reference architecture.

The basic philosophy in designing the LDMs is to focus primarily on the cloud service requirements rather than the solutions. So, each model is independent of the technology or service implementation. A complete LDM describes entities, attributes, and relationships and provides definitions and a detailed documentation of the cloud reference architecture. An entity represents a distinct cloud service objective and encapsulates a set of key feature offerings for a given cloud service objective. The relationship defines the association between entities, and the attribute specifies a property of the entity.

An LDM for the DMTF Cloud Architecture

The ER-D in Figure 2 represents a high-level illustration of the LDM for the DMTF across three domains: cloud service provider, cloud service consumer, and cloud service developer. The three domains reflect the three major actors in the DMTF reference architecture.

The Consumer Information entity specifies the attributes of consumer organizations (such as corporations or government agencies) that request on-demand cloud services and resources. Additional attributes might be required, depending on the implementation. We can define the consumer ID attributes based on the list of activated cloud services for the consumer.

The type of connecter between the Consumer Information entity and the Cloud Computing Active Services entity indicates the following architectural features:

- The cloud service provider can customize one or more cloud services for each consumer organization. The service provider maintains the profiles, SLAs, interfaces, access controls, and cost structure for the customized service.
- The cloud service provider can activate or deactivate the customized services for each consumer organization.
- The cloud service provider maintains a unique consumer ID for each cloud service consumer.

The cloud service provider activates customized services based on the cloud consumer organizations' needs. The end users of the Cloud Service Consumer entity subscribe to the active cloud services. The type of connecter between the Cloud Computing Active Services entity and the Customized Service entity indicates the following relationships:

- the service provider can offer multiple versions of services deployed by the cloud service developer;

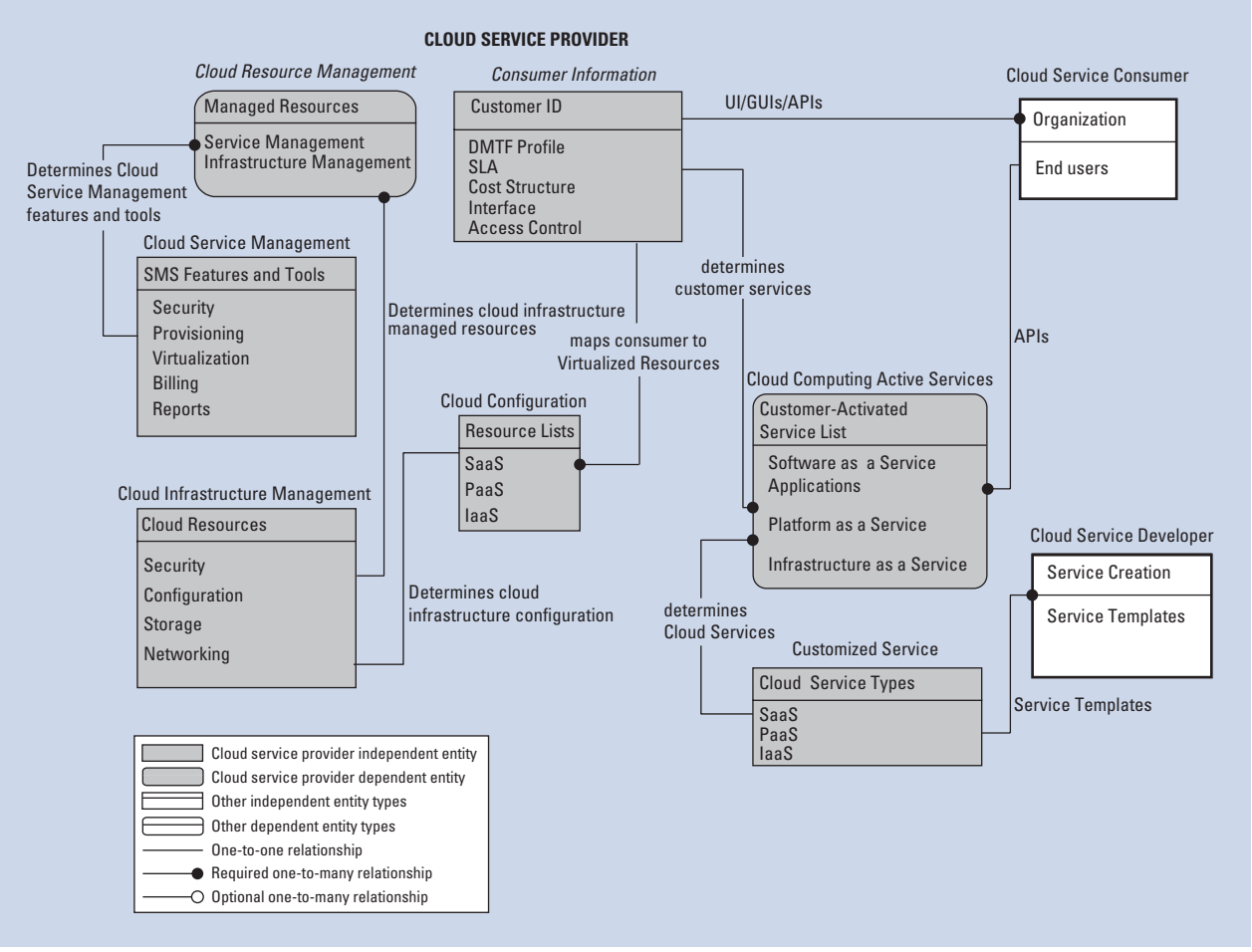

Figure 2. A logical data model for the DMTF cloud reference architecture.

- active services can be SaaS, PaaS, or IaaS; and
- modifying the cloud service offering generates a new customer-activated service list.

The Customized Service entity describes the complete set of services deployed in the cloud. The service provider can customize the services, though some of the services might not be activated. Multiple cloud service developers can create services for the cloud provider; the cloud service developer makes the service available to the cloud service provider using the provider's service templates.

The Cloud Configuration entity identifies the set of resources assigned to each cloud service consumer, while the Cloud Resource Management entity specifies the cloud resources that the service provider manages. Services and infrastructure are two key attributes that the cloud service provider must manage. The cloud service provider might require additional attributes, depending on the cloud implementation. The Cloud Resource Management entity interacts with both the Cloud Service Management entity and the Cloud Infrastructure Management entity. The former specifies the service management system features and tools for security, provisioning, virtualization, billing, audits, and reports. The latter entity specifies the tools and features needed to maintain optimal operations of the cloud infrastructure, including fault management; configuration management; performance management; and storage, networking, and security management.

An LDM for the NIST Cloud Reference Architecture

The ER-D in Figure 3 represents a high-level illustration of the NIST LDM across five domains: the cloud service provider, consumer, auditor, broker, and carrier.

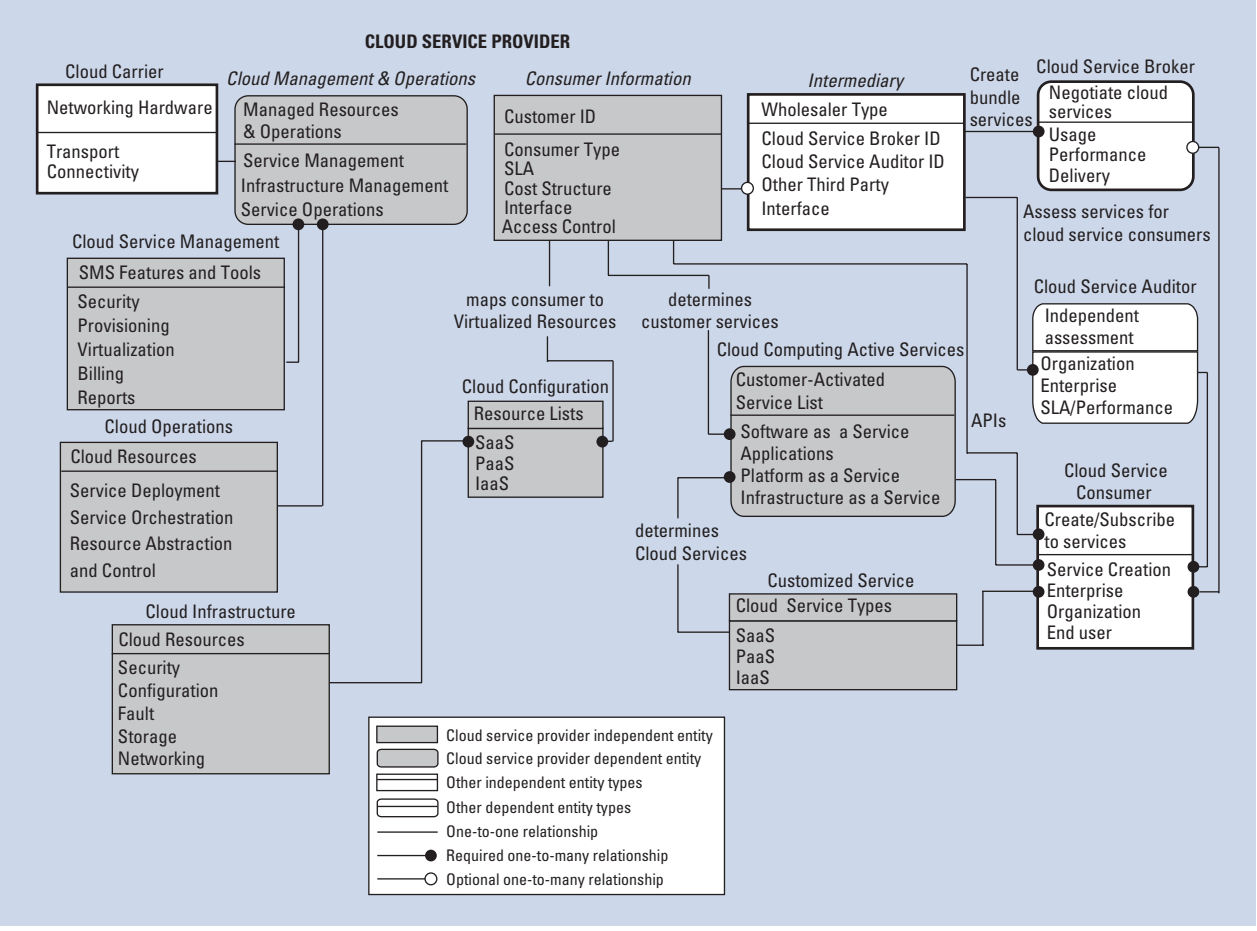

Figure 3. A logical data model for the NIST cloud reference architecture.

The cloud service provider domain describes information that the NIST cloud computing environment must retain to provide cloud services. The description includes the entity types and their internal and external relationships. The Cloud Service Broker, Auditor, Consumer, and Carrier entities are shown as dependent on designated entities in the cloud service provider domain.

The Consumer Information entity specifies the attributes associated with cloud consumer entities (for example, cloud service broker or auditor) and consumer organizations (for example, a person, corporation, or government agency) that request on-demand cloud services and resources. As with the DMTF model, different implementations might require additional attributes. The cloud service provider maintains Consumer ID attributes according to the consumer type and list of activated cloud services.

At the request of a cloud consumer, the Cloud Broker entity interacts with the cloud provider via the Intermediary entity to create bundle services or to enhance existing cloud services. The cloud broker is perceived by the cloud service provider as an Intermediary and, in some cases, as a wholesaler type depending on the contractual agreement between the cloud provider and cloud broker.

At the request of a cloud consumer, the Cloud Auditor entity interacts with the cloud provider via the Intermediary entity to perform independent assessments of the operation and security of the cloud provider's service implementations.

The cloud auditor is perceived by the cloud service provider as an intermediary for the cloud service consumer.

The connector between the Consumer Information and the Cloud Computing Active Services entities has the same relationships as in the DMTF model.

The cloud service provider activates customized services according to the cloud consumer organizations' needs. The end users of the Cloud

Service Consumer entity subscribe to the active cloud services. Again, the connecter between the Cloud Computing Active Services entity and the Customized Service entity has the same relationships as in the DMTF model.

The Customized Service entity is also the same as in the DTMF model, except that the consumer organizations can provision services on the cloud, and the service provider can subsequently customize the provisioned services. The Cloud Configuration entity is the same as in the DMTF. The Cloud Management and Operations entity is similar to the Resources Management entity in the DMTF, except that the former also interacts with a Cloud Operations entity, which specifies the attributes associated with the operational activities of cloud service providers. The operational activities can be classified into four categories: service deployment, service orchestration, resource abstraction, and control. The service management activities can also be classified into five categories: security, provisioning, virtualization, billing and reports.

Finally, unlike in DMTF, the NIST LDM includes a Cloud Carrier entity, which specifies the attributes associated with the networking and transport of the cloud services within the cloud infrastructure and to (or from) the Cloud Service Consumer entity. The Cloud Carrier entity interacts with the Cloud Management and Operations entity to provide connectivity and to transport cloud services from the cloud provider to the cloud consumer.

In general, the complete LDM should also include documentation that illustrates the use cases and describes the cloud service offerings. The documentation supplements the ER-Ds and the associated narratives I've discussed.

Analysis of the LDMs

The LDMs describe cloud entity interactions at several levels: cloud service management, cloud resource management, cloud service development, and cloud service provisioning. These relationships let service providers make objective decisions about evolving business requirements and cloud architecture implementations. Service providers can customize the logical models to meet the needs of company policies while maintaining the core entity relationships.

The LDMs illustrate a set of entities and relationships that are common to both the DMTF and NIST architectures. The models thus provide a framework for developing a common set of requirements for cloud computing architectures. The common requirements include the management of cloud service offerings as SaaS, PaaS, and IaaS and the management of cloud resources. Both models provide the initial framework for traceability between evolving business requirements and cloud architecture implementations.

The models also indicate key differences in the operations paradigm and the cloud interactions with external entities. The DMTF logical data model illustrates how to provide interactions between the cloud service provider and both the cloud service developers and consumers. The DMTF cloud service offerings are designed around the service provider, developer, and consumer; they don't address the cloud operations scenarios in detail.

The NIST LDM, on the other hand, shows the interactions between the cloud service provider and the cloud service consumer, which includes the developer, organization, and end users (similar to DMTF). It also shows interactions between the cloud service provider and entities for the cloud service broker, cloud service auditor, and cloud carriers, as well as interactions between the cloud service consumer and both the cloud service broker and cloud service auditor.

You could conceptually combine the two LDMs to derive a hybrid LDM. Alternatively, you could modify the NIST LDM, which is more detailed, to meet a service provider's specific needs. Figure 4 shows a hybrid that's a modified version of the NIST LDM. Specifically, the Cloud Service Broker, Cloud Service Auditor, and Intermediary entities are designated as optional entities, because they're not an integral part of the DMTF LDM.

Future work includes extending the high-level ER-D abstractions to detailed LDMs using UML class diagrams.[6] The hybrid LDM for the DMTF and NIST architectures will be extended to include other cloud architectures.

Service providers can use LDMs to make objective decisions about evolving business requirements and cloud

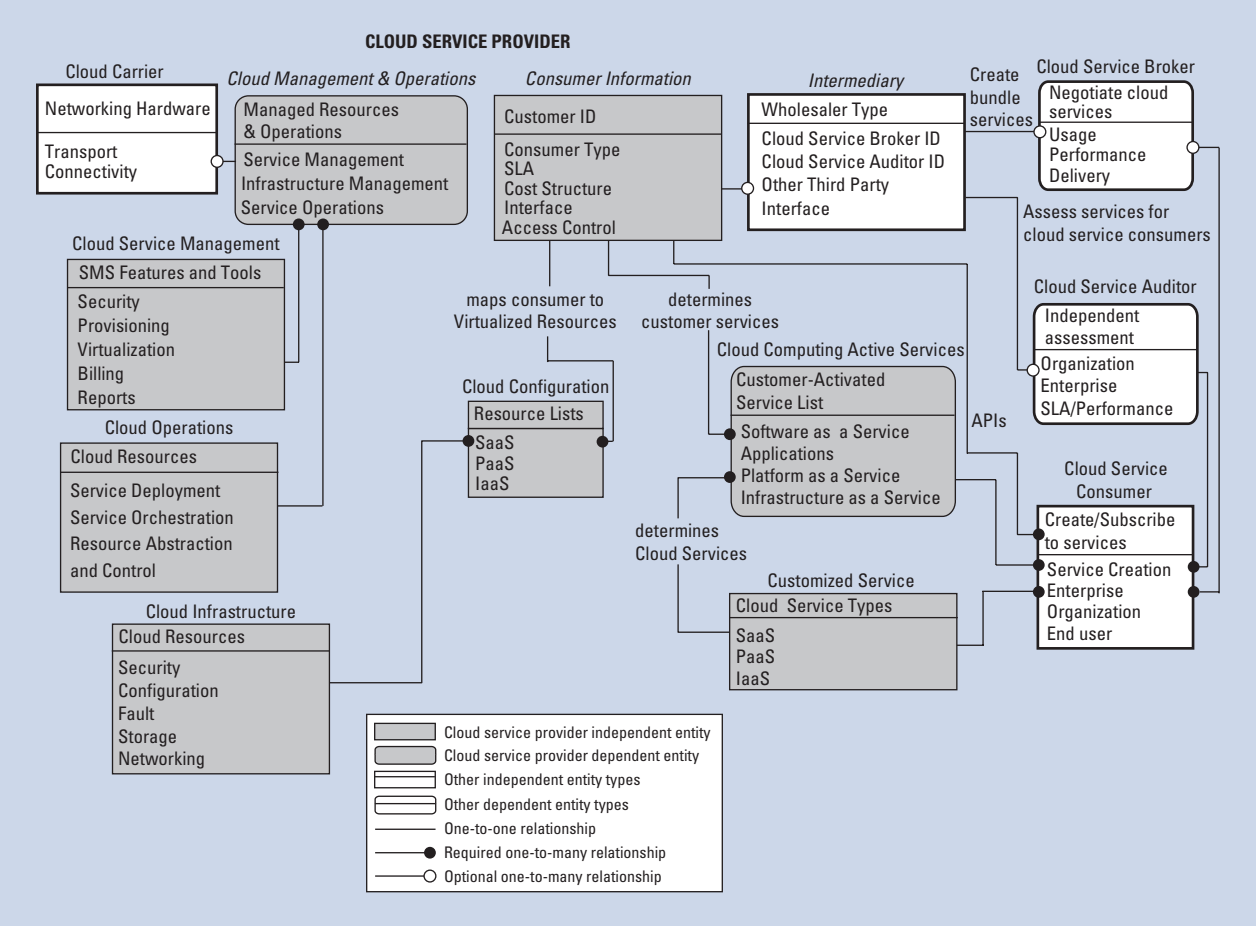

Figure 4. A hybrid logical data model for NIST and DMTF cloud reference architectures.

architecture implementations, and they can customize the LDMs to their specific needs. Thus, they can use the framework as a foundation for developing hybrid cloud architectures. IT

References

1. A. Samba and I. Bojanova, "Analysis of Cloud Computing Delivery Architecture Models," *Proc. IEEE Workshops of Int'l Conf. on Advanced Information Networking and Applications* (WAINA 11), IEEE Press, 2011, pp. 453–458.
2. "Interoperable Clouds," white paper, DMTF, Nov. 2009; www.dmtf.org/sites/default/files/standards/documents/DSP-IS0101_1.0.0.pdf.
3. P. Shoval, R. Danoch, and M. Balabam, "Hierarchical Entity-Relationship Diagrams: The Model, Method of Creation and Experimental Evaluation," *Requirements Eng.*, vol. 9, no. 4, 2004, pp. 217–228.
4. P.P. Chen, "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Trans. Database Systems*, vol. 1, no. 1, 1976, pp. 1–36.
5. L.G. Jimenez, "REERM: Re-Enhancing the Entity Relationship Model," *Data & Knowledge Eng.*, vol. 58, no. 3 2006, pp. 410–435.
6. OMG Unified Modeling Language (UML) Infrastructure, Version 2.4, Jan. 2011; www.omg.org/spec/UML/2.4/Infrastructure.

Augustine (Gus) Samba *is an associate professor at Kent State University, where he holds faculty appointments in the College of Technology and the School of Digital Sciences. He heads the Computer Engineering Technology programs in the College of Technology. His research interests include 4G mobile networks, network management, and cloud computing architectures. Samba received his PhD in computer science from The University of Liverpool, UK. He holds two US patents, and has one patent pending for multimedia network traffic management and control in 4G wireless networks. Contact him at asamba@kent.edu.*