Document Number: DSP0262

Version: 1.0.0

Date: 2012/09/21

Work Group Version: 0.7.0

Cloud Auditing Data Federation (CADF) -

Data Format and Interface Definitions Specification

THIS DOCUMENT IS A WORK IN PROGRESS VERSION.

This document is subject to change at any time without further notice.

It expires on: 01/31/2013

Provide any comments through the DMTF Feedback Portal: **http://www.dmtf.org/standards/feedback**

IMPORTANT: This specification is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, this specification may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.

Document Type: DMTF Specification

Document Status: Work In Progress

Document Language: en-US

Notices

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents for uses consistent with this purpose, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party’s reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit:

http://www.dmtf.org/about/policies/disclosures.php.

Contents

Notices 2

Contents 3

Foreword 12

1 Introduction 13

1.1 Document versioning scheme 13

1.2 Cloud auditing data federation use cases 13

1.2.1 Auditing cloud applications independently of provider 13

1.2.2 Auditing hybrid cloud applications 14

1.2.3 Granular use cases 15

2 Terminology, references and definitions 17

2.1 Terminology 17

2.2 Normative references 17

2.3 Document versioning scheme 18

2.4 Definitions 18

3 Specification scope and goals 23

3.1 Scope 23

3.2 Goals 23

3.2.1 Interface definitions 24

3.2.1.1 Interaction model 24

3.2.2 Audit data integrity and security 24

3.2.3 Audit data set sizes and performance 25

3.2.4 Extensibility 25

3.2.4.1 Profiles 25

3.2.5 Use cases and examples 25

3.3 Out of scope 25

3.3.1 Translation 26

3.3.2 Security policies 26

3.3.3 Forensic information 26

3.3.4 Debug information 26

3.3.5 Configuration data 26
3.3.6 Audit event alerting 26
4 CADF Event Model 27
4.1 Basic concepts 27
4.1.1 Resource 27
4.1.2 Actual Event, Event Record, CADF Event Record 27
4.2 Basic model components 27
4.2.1 Conceptual event model 28
4.2.2 CADF Event Type 28
4.2.2.1 CADF Event Type values 29
4.2.3 Reporter chain 29
4.2.3.1 CADF Reporter roles 30
4.2.3.2 Example 30
4.2.3.3 Requirements on intermediate CADF Event Record completeness 31
4.2.4 Additional Model Components 31
4.2.4.1 Measurements and Metrics 31
4.2.5 Resource classification 31
4.3 Examples of mapping typical events to CADF Event Model 32
4.3.1 Use case: "Auditing access to a controlled resource" 32
4.3.1.1 Use case applied to CADF Event Model 32
4.3.2 Use case: "Periodic monitoring resource status" 33
4.3.2.1 Use case applied to CADF Event Model 33
4.3.3 Use case: "Aggregation of resource status into an audit event" 34
4.3.3.1 Use case applied to CADF Event Model 34
4.3.4 Use case: "Auditing compliance of resource monitors" 35
4.3.4.1 Use case applied to CADF Event Model 35
5 Data model and schema conventions 37
5.1 Aliases for domain and namespace URI values 37
5.1.1 Requirements 37
5.2 Namespaces and namespace aliases 37
5.2.1 Requirements 37

5.2.2 Usage example 38

5.3 URI space 38

5.3.1 Requirements 38

5.4 Entity naming conventions 38

5.4.1 Requirements 38

5.4.2 XML naming requirements 38

5.5 Property constraints 38

5.5.1 "Required" constraint: 39

5.6 Format-specific representations 39

5.6.1 Entity type URIs 39

5.6.1.1 Requirements 39

5.6.1.2 Notes 40

5.6.2 Language identification 40

5.6.2.1 Requirements 40

5.6.2.2 Examples 40

5.6.3 Rules for XML and JSON format representation 40

5.6.3.1 Requirements 41

5.6.3.2 Examples 41

6 CADF Entities and data types 43

6.1 Extensibility mechanisms 43

6.1.1 Derivation 43

6.1.2 Attachments 44

6.1.2.1 Attachment notes 44

6.2 Basic data types 44

6.2.1 General requirements 44

6.2.2 boolean 44

6.2.3 integer 44

6.2.4 double 44

6.2.5 string 45

6.2.6 duration 45

6.2.6.1 Lexical representation 45

6.2.7 URI 45

6.2.7.1 Additional URI Requirements 45

6.2.8 Basic type translation to JSON from XML 45

6.3 CADF basic data types 46

6.3.1 Identifier type 46

6.3.1.1 Design considerations 46

6.3.1.2 Requirements 46

6.3.1.3 Lexical representation 47

6.3.1.4 Best practices 47

6.3.1.5 Examples 47

6.3.2 Path type 48

6.3.2.1 Design considerations 48

6.3.2.2 Requirements 48

6.3.2.3 Lexical representation 49

6.3.2.4 Best practices 49

6.3.2.5 Examples 49

6.3.3 Timestamp type 50

6.3.3.1 Design considerations 50

6.3.3.2 Requirements 51

6.3.3.3 Lexical representation 51

6.3.3.4 Examples 52

6.3.3.5 Notes 53

6.4 CADF complex data types 53

6.4.1 Array types 53

6.4.1.1 Serialization example 53

6.4.2 Attachment type 54

6.4.2.1 Design considerations 54

6.4.2.2 Requirements 54

6.4.2.3 Notes 54

6.4.2.4 Properties 54

6.4.2.5 Serialization examples 55

6.4.3 Endpoint type 55
6.4.3.1 Design considerations 55
6.4.3.2 Requirements 55
6.4.3.3 Properties 55
6.4.3.4 Serialization examples 56
6.4.4 Geolocation type 56
6.4.4.1 Design considerations 56
6.4.4.2 Requirements 56
6.4.4.3 Properties 57
6.4.4.4 Property Notes 59
6.4.4.5 Serialization examples 59
6.4.5 Map 62
6.4.5.1 Design considerations 62
6.4.5.2 Requirements 62
6.4.5.3 Properties 62
6.4.5.4 Serialization examples 62
6.4.6 Metric and Measurement types 63
6.4.6.1 Design considerations 63
6.4.6.2 Requirements 63
6.4.6.3 Properties of Metric 64
6.4.6.4 Properties of Measurement 64
6.4.6.5 Serialization examples 65
6.4.7 Reason type 67
6.4.7.1 Design considerations 67
6.4.7.2 Requirements 67
6.4.7.3 Properties 67
6.4.7.4 Examples 67
6.4.7.5 Serialization Examples 68
6.4.8 Reporterstep type 68
6.4.8.1 Design considerations 68
6.4.8.2 Requirements 68

6.4.8.3 Properties 69
6.4.8.4 Serialization examples 70
6.4.9 Resource type 70
6.4.9.1 Design considerations 70
6.4.9.2 Requirements 70
6.4.9.3 Properties 71
6.4.9.4 Serialization Examples 72
6.5 CADF Entities 72
6.5.1 Event type 72
6.5.1.1 Design considerations 72
6.5.1.2 Entity Type URI 73
6.5.1.3 Requirements 73
6.5.1.4 Best practices 73
6.5.1.5 Properties 74
6.5.1.6 Serialization examples 76
6.5.2 Log type 79
6.5.2.1 Design considerations 80
6.5.2.2 Entity Type URI 80
6.5.2.3 Requirements 80
6.5.2.4 Properties 81
6.5.2.5 Serialization examples 82
6.5.3 Report type 83
6.5.3.1 Differences between reports and logs 83
6.5.3.2 Design considerations 83
6.5.3.3 Use cases 83
6.5.3.4 Entity Type URI 83
6.5.3.5 Requirements 83
6.5.3.6 Properties 84
6.5.3.7 Serialization examples 85
7 CADF Resource type derivations 86

7.1 Extended property requirements for resource types 86
7.2 Notes 86
7.3 Extended properties for derived CADF Resource types 86
7.3.1 Account 86
7.3.2 Connection 87
7.3.3 Credential 87
7.3.3.1 Notes 87
7.3.4 Endpoint 88
7.3.5 Node (Network, Compute, Storage) 88
7.3.6 Service 88
7.3.7 User 89
8 CADF Interfaces 90
9 CADF Entity signing 91
10 CADF Profiles 92
10.1 Requirements 92
11 Future Considerations 93
A. CADF Event Model component classification 94
A.0 CADF Resource Taxonomy 94
A.0.1 Model description 94
A.0.2 Notes on mapping to the resource taxonomy 94
A.0.3 Taxonomy URI 94
A.0.4 Requirements 95
A.0.5 Hierarchical resource classification tree 95
A.0.6 Logical resource classification tree 96
A.0.7 Storage subtree classifications 96
A.0.8 Compute subtree classifications 97
A.0.9 Network subtree classifications 98
A.0.10 Service subtree classifications 99
A.0.11 Data (objects) subtree classifications 100
A.0.12 Security (data objects) subtree classifications 101
A.0.12.1 Design considerations 101

A.0.13 Database (data object) subtree classifications 103
A.0.14 Using the resource taxonomy 104
A.1 CADF Action Taxonomy 104
A.1.1 Model description 104
A.1.2 Notes on mapping to the action taxonomy 105
A.1.3 Taxonomy URI 105
A.1.4 Requirements 105
A.1.5 Hierarchical action classification 106
A.1.6 Taxonomy extension 107
A.1.7 Using the action taxonomy 107
A.2 CADF Outcome Taxonomy 108
A.2.1 Design considerations 108
A.2.2 Taxonomy URI 108
A.2.3 Requirements 109
A.2.4 Hierarchical action classification 109
A.2.5 Taxonomy values 109
A.2.6 Requirements 110
A.2.7 Using the outcome taxonomy 110
A.2.8 Considerations when using "unknown" or "pending" values 110
A.3 Treatment of INITIATOR, TARGET, and OBSERVER 111
A.3.1 Overview 111
A.3.2 Treatment of INITIATOR 111
A.3.3 Treatment of TARGET 112
A.3.4 Treatment of OBSERVER 112
A.4 Using the CADF Taxonomies to create CADF Event Records 112
A.4.1 General rules 112
A.4.2 Examples 113
A.4.2.1 Account creation 113
A.4.2.2 User Authentication 114
B. Best practices 115

270 B.0 Treatment of timestamps in CADF Event Records 115

C. Mapping CIMI Events to CADF Event Record 116

D. Mapping CIM Indications to CADF Event Records 117

E. Bibliography (Informative) 118

Change Log 120

271

Foreword

The *Cloud Auditing Data Federation (CADF) Data Format and Interface Specification* (DSP0262) was prepared by the Cloud Auditing Data Federation (CADF) Working Group

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Acknowledgements

The editors wish to acknowledge the following people.

Chairpersons

- David Corlette, NetIQ
- Matthew Rutkowski, IBM

Editors

- Matthew Rutkowski, IBM

Contributors

- Alvin Black, CA Technologies
- Davi Ottenheimer, VMware
- David Corlette, NetIQ
- Hemal Shah, Broadcom
- Il-Sung Lee, Microsoft
- Jacques Durand, Fujitsu
- John Parchem, Microsoft
- Marlin Pohlman, EMC
- Matthew Rutkowski, IBM
- Mike Edwards, IBM
- Monica Martin, Microsoft
- Ola Nordstrom, Citrix Systems
- Rick Cohen, IBM
- Steven Neely, Cisco
- Winston Bumpus, VMware
- Xavier Guerin, France Telecom
- Zhexuan Song, Huawei

1 Introduction

Concerns over cloud provider security remain one of the top inhibitors to adoption of cloud deployment models. Potential consumers of cloud deployments understand and need assurance that the security policies they require on their applications are consistently managed and enforced “in the cloud” as they would be in their enterprise.

A cloud provider’s ability to provide specific audit event, log and report information on a per-tenant and application basis is essential. It is apparent that in order to meet these customer expectations, cloud providers must provide standard mechanisms for their tenant customers to self-manage & self-audit application security that includes information about the provider’s hardware, software and network infrastructure used to run specific tenant applications.

A proven method to address such needs is to develop open standards to enable information sharing. Specifically, this specification provides a data format and interface definitions that support the federation of normative audit event data to and from cloud providers in the form of customized reports and logs. This specification also defines a means to attach domain specific identifiers, event classification values and tags that can be used to dynamically generate customized logs and reports for cloud subscribers or customers.

Adoption of this and other open standards by cloud providers’ management platforms would go far to instill greater trust in “cloud hosted applications” and be a significant step forward in fulfilling the promise of an open cloud marketplace.

1.1 Document versioning scheme

This document will adhere to the versioning scheme defined in clause 6.3 of [DMTF DSP4004].

1.2 Cloud auditing data federation use cases

This section includes the general, high-level use cases that provide the basis for establishing the need for standardized federation of cloud auditing data.

1.2.1 Auditing cloud applications independently of provider

Companies need to audit the compliance of their applications against their corporate or industry requirements and policies while being hosted by cloud providers. Additionally, these applications may run on different cloud deployments or with different providers over their lifecycle. Companies should be able to preserve their investments in the processes and tooling that provides them necessary audit data regardless of cloud deployment model or the provider hosting the application.

In other words, that with open standards for cloud auditing data formats along with open standardized interfaces for interacting with that data companies can more easily compare the costs of hosting their application with various cloud providers without worrying that they will lose their ability to audit their applications or have to factor in the cost of changing auditing processes and tools to adapt to different formats and interfaces.

The following figure shows Company A hosting their application with Cloud Provider A and using auditing processes and tooling that utilize standard interfaces for retrieving standardized auditing data that Cloud Provider A supports.

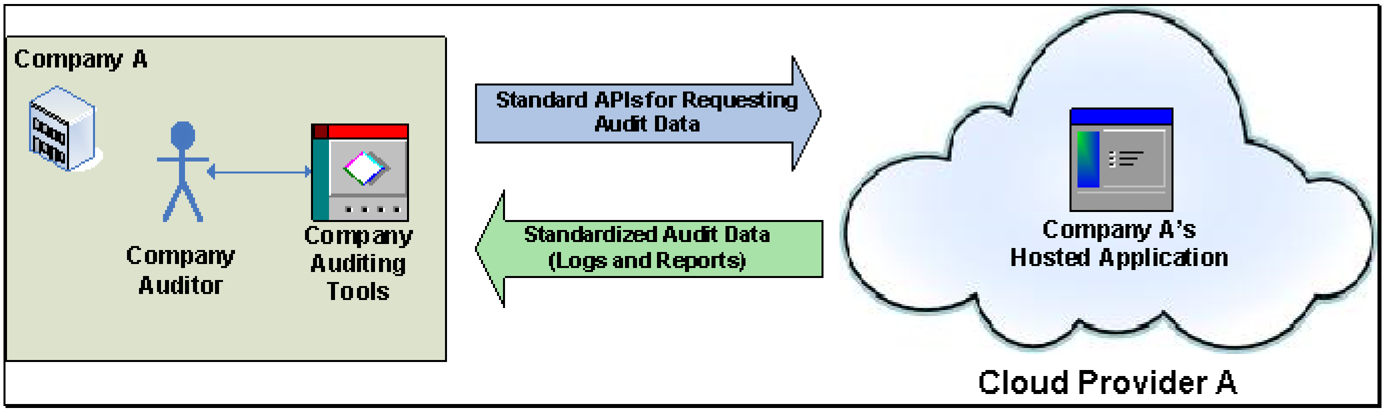

Company A
Company Auditor
Company Auditing Tools
Standard APIsfor Requesting Audit Data
Standardized Audit Data (Logs and Reports)
Company A's Hosted Application
Cloud Provider A

Figure 1 - Company A Hosts Application at Cloud Provider A; Auditing Tools use Open Standards

The following figure shows that Company A decided to move to their hosted application from Cloud Provider A to Cloud Provider B (perhaps to affect cost savings). This change of provider, however, did not affect any changes to Company A's established auditing processes and tooling because both providers supported the same standard audit data format and interfaces.

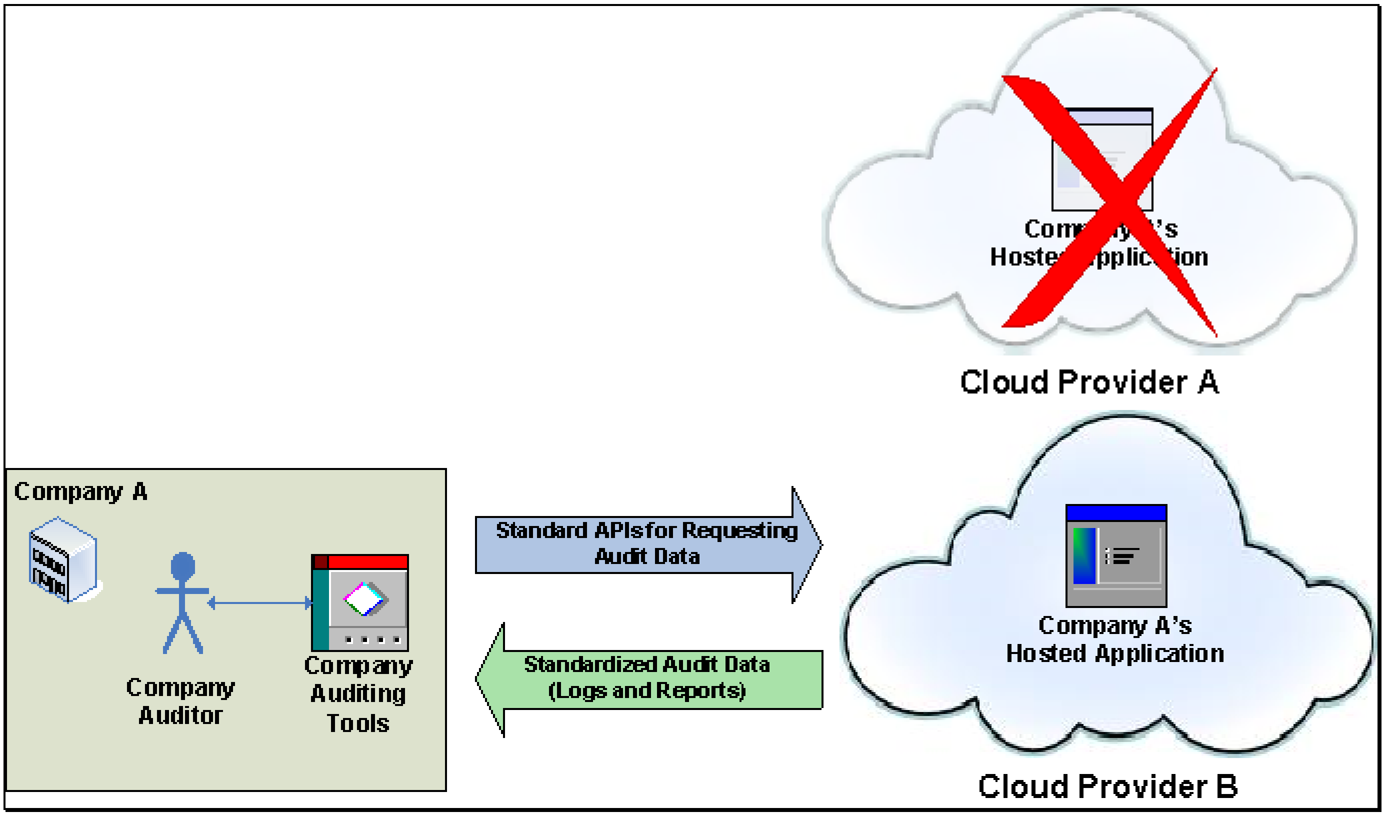

Company A's Hosted Application
Cloud Provider A
Company A
Company Auditor
Company Auditing Tools
Standard APIsfor Requesting Audit Data
Standardized Audit Data (Logs and Reports)
Company A's Hosted Application
Cloud Provider B

Figure 2 - Company A Moves Application from Cloud Provider A to Provider B; Auditing Tools Unchanged

1.2.2 Auditing hybrid cloud applications

Since many cloud providers offer various services and resources, it is easy to understand that companies may wish to compose hybrid applications that span from across multiple traditional and cloud based deployments to take advantage of the best and most cost effective services that meet their needs.

The hybrid application, as a whole needs to be audited regardless of where its composite services and resources are deployed. If each of these deployment environments used an open standards based audit data format with compatible open standard interfaces for management of that data, the company's audit tooling could uniformly access all deployment environments to retrieve audit reports using the same criteria and logs and easily aggregate the data from these independent sources into a single audit trail.

The following figure shows a single company retrieving and aggregating the same standardized audit data from multiple sources using the same standard interfaces. Specifically, these sources include the company's own Operational Support Services (OSS) and Business Support Services (BSS) and externally from two independent cloud providers.

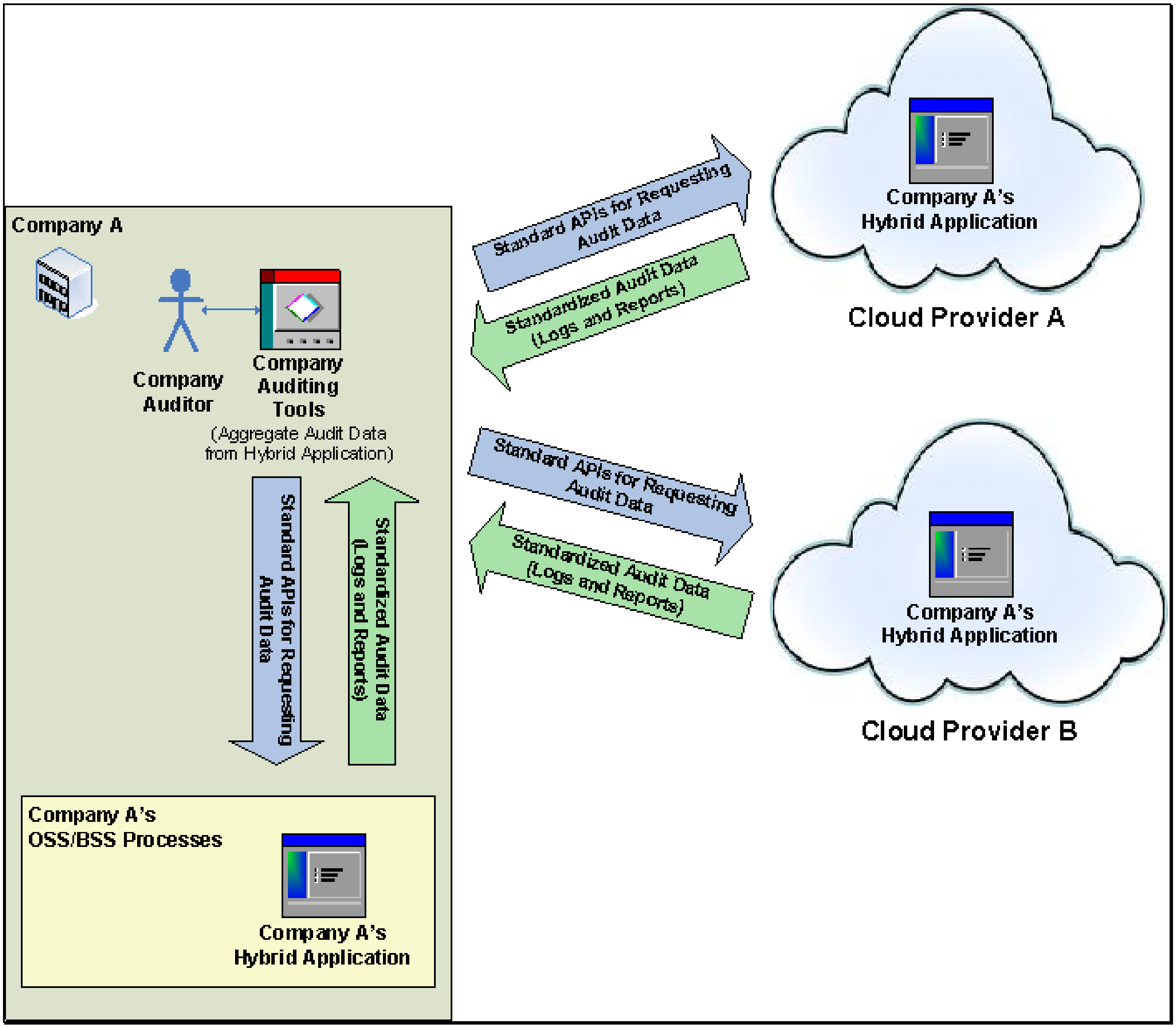

Company A
Company Auditor
Company Auditing Tools
(Aggregate Audit Data from Hybrid Application)
Standard APIs for Requesting Audit Data
Standardized Audit Data (Logs and Reports)
Company A's OSS/BSS Processes
Company A's Hybrid Application
Standard APIs for Requesting Audit Data
Standardized Audit Data (Logs and Reports)
Company A's Hybrid Application
Cloud Provider A
Standard APIs for Requesting Audit Data
Standardized Audit Data (Logs and Reports)
Company A's Hybrid Application
Cloud Provider B

Figure 3 - Company Aggregates Audit Data from Hybrid Cloud Application Across Various Deployments

1.2.3 Granular use cases

Beyond the general use cases, the CADF working group has sought to provide a flexible audit data format suitable for conveying many types of audit and compliance data in the form of events. As a means to ensure that this goal is met, the working group has published DMTF document DSP2028 "*Cloud Auditing*

Data Federation (CADF) Use Case White Paper" which includes discrete use case submissions that were reviewed and considered as non-binding input when developing this specification.

The CADF accepts comments to this white paper in accordance with DMTF processes.

2 Terminology, references and definitions

2.1 Terminology

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

The terms "SHALL" ("required"), "SHALL NOT," "SHOULD" ("recommended"), "SHOULD NOT" ("not recommended"), "MAY," "NEED NOT" ("not required"), "CAN" and "CANNOT" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do not contain normative content. Notes and examples are always informative elements.

2.2 Normative references

The following normative references are used by this specification. The tag value for each reference MAY be used within the document to provide specific attribution for clarity.

Tag	Reference
[DMTF DSP0004]	DMTF Specification DSP0004, *Common Information Model (CIM) Infrastructure*, *Version: 2.7.0*, April 2012, http://dmtf.org/sites/default/files/standards/documents/DSP0004_2.6.0_0.pdf
[DMTF DSP4004]	DMTF Specification DSP 4004*, DMTF Release Process, Version* 2.4.0, 26 January 2011, http://www.dmtf.org/sites/default/files/standards/documents/DSP4004_2.4.0.pdf
[DMTF DSP4009]	DMTF Specification DSP4009, *Process for publishing XML schema, XML 6 documents and XSLT Stylesheets, Version 1.0*, August 2007, http://www.dmtf.org/sites/default/files/standards/documents/DSP4009_1.0.0.pdf.
[IETF RFC 3986]	T.Berners-Lee et al, *Uniform Resource Identifiers (URI): Generic Syntax*, Jan. 2005, http://www.ietf.org/rfc/rfc3986.txt
[IETF RFC 4627]	D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*, July 2006, http://www.ietf.org/rfc/rfc4627.txt
[IANA-ccTLD]	Internet Assigned Numbers Authority (IANA), Root Zone Database, Listing of Internet Corporation for Assigned Names and Numbers ("ICANN") country codes (ccTLDs), http://www.iana.org/domains/root/db/
[ICANN-ccTLD]	ICANN, *Final Implementation Plan for IDN ccTLD Fast Track Process*, 9 April 2012, http://www.icann.org/en/resources/idn/fast-track/idn-cctld-implementation-plan-redline-09apr12-en
[ISO Directi-Pt2]	ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards, http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype
[ISO 8601:2004]	ISO 8601:2004 (E), *Data Elements and Interchange Formats – Information Interchange – Representation of Dates and Times*, 2004, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874
[W3C-XML]	W3C Recommendation, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, November 2008, http://www.w3.org/TR/REC-xml/.
[W3C-Names]	W3C Recommendation, Namespaces in XML 1.0 (Third Edition), December 2009, http://www.w3.org/TR/REC-xml-names/.

[WSI-Basic-1.2] WS-I WG Draft, *Basic Profile Version 1.2*, October 2007, http://www.ws-i.org/Profiles/BasicProfile-1_2%28WGAD%29.html.

[XMLSchema0] World Wide Web Consortium (W3C) Recommendation, D. Fallside, P. Walmsley, et al., Editors, *XML Schema Part 0: Primer Second Edition*, 28 October 2004, http://www.w3.org/TR/xmlschema-0/.

[XMLSchema1] World Wide Web Consortium (W3C) Recommendation, H. Thompson, et al., Editors, *XML Schema Part 1: Structures Second Edition*, 28 October 2004, http://www.w3.org/TR/xmlschema-1/

[XMLSchema2] World Wide Web Consortium (W3C) Recommendation, P. Biron, A. Malhotra, Editors, *XML Schema Part 2: Datatypes Second Edition*, 28 October 2004, http://www.w3.org/TR/xmlschema-2/

2.3 Document versioning scheme

This document will adhere to the versioning scheme defined in the W3C's XML Schema Part 2 section 6.3.

2.4 Definitions

This section defines terms for use within the CADF specification. In doing so, this specification may re-use terms from other domains, in some cases extending, modifying, or restricting those definitions.

Actual Event

Anything that happens, or is contemplated as happening [EPTS Glossary]. This definition encompasses events taking place within or outside computing domains, and has nothing to do with any description of the actual event.

In common usage and where the meaning is clear in context, we will sometimes use simply “Event” when discussing “Actual Events.”

Aggregation

Aggregation refers to the combination within a single event of two or more other events (or references to those events). Aggregation is typically a bundling of separate events which preserves and keep the original events accessible.

Audit

A survey of a set of systems to determine if they are complying with stated policy objectives.

Systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which audit criteria are fulfilled. [ISO 14001:2004]

Within the scope of this specification, the definition of "audit" is restricted to the representation, collection, storage and evaluation of CADF Event Records. [ISO 15288:2008]

Audit Event

An audit event is any event record that reports activity that may be used for the purposes of an audit.

Audit Trail

A chronological record that reconstructs and examines the sequence of activities surrounding or leading to a specific operation, procedure, or event in a security relevant transaction from inception to final result. [CNSS4009]

Authentication

A process used to achieve sufficient confidence in the binding between the entity and the presented identity. NOTE: Use of the term authentication in an Identity Management (IdM) context is taken to mean entity authentication. [ITU X.1252]

Authorization

The process of determining, by evaluating applicable access control information, whether a subject is allowed to have the specified types of access to a particular resource. [SAML-Gloss-2.0]

A prescription that a particular behavior shall not be prevented [ISO 15414:2006]

Compliance Event

A compliance event is any event record that reports activity that is required to show compliance to a policy or requirement which are often described by compliance standards.

Note: Security compliance events are specialized compliance events that record activity related to authorization and enforcement of security policies in accessing system resources.

Control Objective

A control objective refers to a compliance related requirement or practice. These control objectives are often described by policies and enforcement proven by compliance audits.

In the context of this specification, control objectives are typically requirements on cloud providers that are expected to supply audit compliance data in the form of event records, logs and reports.

Event Consumer

An entity which needs to process, report on, or otherwise use CADF Event Records.

Event Provider

An entity which is able to produce or deliver CADF Event Records.

Data Federation

Any means in which two or more domains enable sharing and exchange of information, such as audit data, for service or content composition, consumption or delivery and coordination with each other. [Kobielus:2006], [Navajo:2009]

Event

1. An “Actual Event.”
2. An “Event Record.”

In common usage we will use the simpler term “Event” to refer to either “Actual Events” or “Event Records,” with the expectation that the correct definition will be clear in context. In this specification, we attempted to use the more complete term to disambiguate where possible.

Event Action

The action (verb) performed by the event initiator (a resource) against the event target resource or resources.

Event Initiator

The resource that initiated, originated or instigated the event action. Typically, the initiating resource is either a user or service that can be identified or described by the system in which the event occurs [TOG-XDAS1].

Event Log

A persistent collection of event records. In context, this term may be expressed simply as “Log.”

Event Observer

The resource that observed the actual event and generated an event record to describe it. The observer may or may not itself have been the event initiator or event target.

Please note that in the [EPTS Glossary], this resource is referred to as an event source for the event record. In this specification, we avoid use of the term "source" to prevent ambiguity between event observer and event initiator.

Event Query

A request initiated, for example by a consumer to a provider, asking for a particular set of persisted event records that match some selection criteria. The returned set is typically a bounded set, in that it is returned as part of a discrete transaction and returns only the event records that are currently available at the time of the query.

Event Record

A record or object that represents, encodes, or records an event, generally for the purpose of computer processing [EPTS Glossary].

In common usage and where the meaning is clear in context, we will sometimes use simply “Event” when discussing “Event Records”.

The term "CADF Event Record" is used specifically to reference an event record that conforms to the CADF specification.

Event Source

Is a term often used in different ways in other domains, such as the [EPTS Glossary], when modeling events and could lead to ambiguity. Therefore, the CADF specification will prefer the more precise terms “Event Initiator” and “Event Observer” and avoid the use of this term.

Event Stream

A non-persistent, linearly ordered sequence of events [EPTS Glossary].

Typically an event stream:

1. may be ordered by time.
2. may be bounded by a certain time interval or other criteria (content, space, source), or be open ended and unbounded.

Event Target

The resource or resources that were the intended targets of the event action [TOG-XDAS1].

Filtering

Filtering refers to the process of selecting a subset of event records to be returned as the result of a query and is typically performed based upon selection criteria within the query.

Geolocation

Geolocation refers to the identification of the geographical location of a resource or entity related to an event. The identification of the physical location of a resource or player is important from a legal compliance perspective to ensure or audit compliance with the laws of various countries, regions, or logical boundaries which dictate where information must be stored.

Georouting

Geo-routing refers to the geographical tracking of an event from its origin through the various resources which participated in the event or the handling an event.

Log

See definition for "Event Log".

Query

See definition for "Event Query".

Security Event

Identified occurrence of a system, service or network state indicating a possible breach of information security, policy or failure of controls, or a previously unknown situation that may be security relevant. [ISO 27000:2009]

An occurrence in a system that is relevant to the security of the system. See "Security Incident". [RFC 2828]

Security Incident

Single or a series of unwanted or unexpected information security events that have a significant probability of compromising business operations and threatening information security. [ISO 27000:2009]

Selection Criteria

A set of terms that define rules for matching against a set of input records. Records that match the selection criteria are included in the output set; records that do not match are filtered out of the output set.

Sexagesimal

A numeral system with sixty as its base (i.e. base 60). In the context of this specification, geographic coordinates are often expressed as degrees, minutes and seconds which is a base 60 system.

Subscription

A contract that is established between a consumer and a provider that asks the provider to deliver future generated records that match some selection criteria to the consumer. The records can be delivered in real time or on a scheduled basis; individually or in aggregated forms; or according to any other terms in the contract.

Summarization

Summarization refers to the consolidation of multiple related events in to a single event, typically for storage or bandwidth optimization or for other analytical purposes.

Suppression

Suppression refers to the dropping or elimination of event records from an event stream or event log. From an auditing perspective, the entity which drops the event records will typically create a “meta” event record indicating the count and type of event records being dropped.

3 Specification scope and goals

3.1 Scope

This specification includes the definition of an:

- **Audit Data Format** - that includes describing a data model and associated schema definitions for event records, logs and reports that can be formatted for federation and are suitable for audit purposes.
- **Extensible Event Taxonomies –** that are to be used to categorize and classify CADF Event Records and their component resources and properties.

  These CADF taxonomies include:

  - **Resource Taxonomy** - used to classify the event by the logical IT or cloud resources that are related to the event's action. For example, values of this taxonomy could be used to classify the resource that observed the action or the resource that was the (intended) target of the action.
  - **Action Taxonomy** - used to classify the event by the activity that caused it to be generated.
  - **Outcome Taxonomy** - used to describe the outcome of the attempted action of the event.
- **Interface Definitions –** that define the service methods for management and federation of the CADF data model. This includes definitions for event submission, import, export, and query using the specified event record, log and report formats.
  - This includes the specification of any additional data formats needed to support the query and generation of customized logs and reports.

3.2 Goals

The principal goal of this specification is to ensure that similar auditable events, such as a "logon" or "critical resource update" resolve to the same data format with prescriptive data types, entities and properties to facilitate reporting, query, federation and aggregation.

Therefore, where possible this specification will describe rules to achieve event record normalization and will include:

- Prescriptive data format with supporting schema that defines where possible:
  - Required data entities, properties and values
  - Discrete data types
  - Validatable data value formats
  - Valid data values, ranges, enumerations, etc.
- Clear event classification, using taxonomies, of common event resources, actions and outcomes.
  - Encouraging the consolidation of descriptors for similar resources, actions and outcomes from other domain classification systems so that the terms or values they use can be mapped to single, discrete CADF provided values.
- Common cloud resource definitions.
  - Prescriptive data types, properties and permitted values to represent resources that repeatedly appear on auditable events. For example, this specification will define the data schema that can be used to represent an "Account" or a "Database" as an event resource.
- Interfaces and the supporting data model to reference, query and analyze audit event data.

- Recommendations and best practices to assure scalability to accommodate the potentially large volumes of audit data that need to be federated.

3.2.1 Interface definitions

This specification provides interface definitions that can be used to further specify application or service methods for managing audit event records (in support of federation), including:

- **Event Submission**
  - Support message-level submission of one or more events from federated sources (or services) to a cloud provider.
  - Support information about the source that submitted the event in order to provide domain specific context to resources that could be used to additionally classify or augment the event data.
- **Event Import and Export**
  - Support the import and export of logs containing auditable event records with similar contextual information to and from a cloud provider.
  - Support transforms that can be used for converting domain specific values (e.g., identifiers, classification values, etc.) to values that permit federation and conform to this specification (or vice-versa).
- **Event Query**
  - Support for a standard means to query event records that match specific criteria such as date/time ranges, event taxonomy classifications, domain specific identifiers and tags, occurrences of specific resource types, etc.
  - Support filters used for selecting audit event data sets (for example in the form of logs or reports) that clearly match/identify events that contain specific resource types and/or classification values either defined by this specification or associated with specific domains.
- **Event Subscription**
  - Support cloud provider management platforms that wish to support persistent queries that could be used to generate periodic logs and reports.
  - Support data to describe event, report or log generation frequency (with associated filters) and possible storage or transmission destination(s). This includes subscription to real-time event feeds.

3.2.1.1 *Interaction model*

This specification's interface definitions are based upon a simple interaction model that describes need to federate audit data between cloud deployments and cloud consumers or subscribers (e.g., users, corporations, enterprises, etc.). These definitions seek to account for best practices for message-based data federation and security so that they are consumable for development of application or service methods.

3.2.2 Audit data integrity and security

There is a strong need for ensuring the integrity and security of data used for auditing purposes and especially important when federating the data across domains. This specification describes methods for assuring the security and provenance of the audit data.

To address data integrity this specification will describe methods for:

- **Data Chaining** - ensuring that audit data, once placed in the CADF Event Record, is not deleted or modified; that instead data should be appended to the record.

To address data security this specification will describe methods for:

- **Data Signing** - securely signing audit events records, logs and reports

3.2.3 Audit data set sizes and performance

Cloud providers may produce large amounts of auditable data that will need to be federated by this specification. Wherever possible, the specification attempts to ensure that the CADF data formats do not cause unreasonable overhead that may impact performance.

In addition, cloud consumers need to be able to produce customized views (or reports) from the entirety of the audit data available from a cloud deployment. They also need to produce this data in a timely and predictable manner when queried.

This specification intends to define mechanisms to discretely classify, identify and tag audit event data using values from different domains to help enable both goals.

3.2.4 Extensibility

The logical data model is designed to be extensible by format specific profiles while preserving constraints and rules described by this specification. This specification will draw from XML Schema [XML-Schema] as a means to describe the data model.

See section titled "Extensibility Mechanisms" for approved extension methods.

3.2.4.1 *Profiles*

Profiles may be developed that extend this core specification and its schema in order to accommodate particular methods of consumption. Most typically these profiles may define and describe how data from other domains can be mapped, classified, referenced and/or conveyed by this specification's data model and schema.

Please see the section titled "CADF Profiles" for more information.

3.2.5 Use cases and examples

It is a goal of this specification to provide normative and prescriptive data schema and interfaces that allow customers to audit their applications, resources and data within provider infrastructures. This specification may incorporate reference to use cases and examples to further demonstrate the need for or correct use of this specification’s data format and interface definitions.

3.3 Out of scope

It should be noted that modern computing systems report a wide variety of information in many different ways. This standard is focused on the proper exchange of normative auditable events across cloud deployment models and follows a particular interaction model; the format for reporting other types of data is out of scope.

To be more precise:

- This specification does not define standard interfaces to secondary sources of information commonly used to collect event information, such as interfaces to configuration, debugging or bug tracking systems or services, policies, etc.
- This specification does not define data types or entities for secondary sources of information commonly used in conjunction with events or helping the collection of event information, e.g., configuration data or files, bug data, alerts or alarms, policy rules, etc.

This specification does consider the need to express additional event data within the CADF Event Record and defines specific extension mechanisms for accomplishing this. See section titled "Extensibility Mechanisms" for approved extension methods.

Specific discussion of areas that are "Out of Scope" follow this section.

3.3.1 Translation

This specification will not describe translation of other event formats, schema and notation into or out of this standard's. Such translations may be described in external profiles of this specification.

3.3.2 Security policies

This specification will not address any concerns relating to security policies or their enforcement. This includes consideration of policy enforcement or policy decisions (e.g., authentication, authorization of roles, etc.) that permitted an action to be performed that led to the generation of the auditable event.

Neither will this specification address authentication or authorization to access (permissions) to the audit event data, unauthorized disclosure of event contents, unauthorized submission of events, or unauthorized modification of events that are in transit or stored.

3.3.3 Forensic information

The event format defined in this specification contains normative information that supports activities such as forensics (e.g., eDiscovery, etc.), incident management, risk assessment and others; however, this specification does not attempt to address these issues.

The data, interaction and component models described will not describe analytical processes such as the detection of sequences of events, compound events, root causes, security risks, or policy violations. This type of analysis would be done by backend applications and services consuming the security events.

Profiles and extensions of this specifications data schema SHALL NOT define additional schema to include forensic information.

3.3.4 Debug information

This specification does not address the inclusion of fine-grained debug or trace output including stack dumps, variable states, and other debugging style output.

Profiles and extensions of this specifications data schema SHALL NOT define additional schema to include debug or trace data. Although profiles may provide information that can help locate or reference debug data as an external resource.

3.3.5 Configuration data

The configurations of hardware, software and network components at the time of audit are not considered in this specification.

Profiles and extensions of this specifications data schema SHALL NOT define additional schema to include configuration data. Although profiles may provide information that can help locate or reference configuration data as an external resource.

3.3.6 Audit event alerting

The specification will not include any definitions for alert generation, delivery or similar requirements (e.g., user interface display, emailing, notifications, SMS, etc.).

4 CADF Event Model

4.1 Basic concepts

4.1.1 Resource

The CADF event model is intended to describe the interactions between resources that compose a cloud service provider's infrastructure and that may have significance in showing compliance against policies. The term resource, for the purposes of this specification we define as follows:

Terms	CADF Definition
RESOURCE	is an entity or component that has capabilities to provide or consume services or information within the context of a cloud infrastructure.

Resources in general can be used to describe traditional IT components (e.g., servers, network devices, etc.), software components (e.g., platforms, databases, applications, etc.), operational and business data (e.g., accounts, users, etc.) and roles, that can be assigned to persons, that describe the authority to access capabilities.

4.1.2 Actual Event, Event Record, CADF Event Record

The use of the term "event", when used by itself, can be interpreted in different ways. Therefore, this specification will use the following terms to clearly distinguish between the different types of events:

Terms	CADF Definition
Actual Event	Anything that happens, or is contemplated as happening. This definition encompasses events taking place within or outside computing domains, and has nothing to do with any description of the actual event. See full definition for "Actual Event".
Event Record	The significant information about the Actual Event represented as a formatted set of data for preservation. See full definition for "Event Record".
CADF Event Record	An Event Record that describes its event data using the CADF Event Schema. *Note: The schema of the CADF Event Record is designed so that other event record types or formats can be mapped to the CADF Event Record format.*

4.2 Basic model components

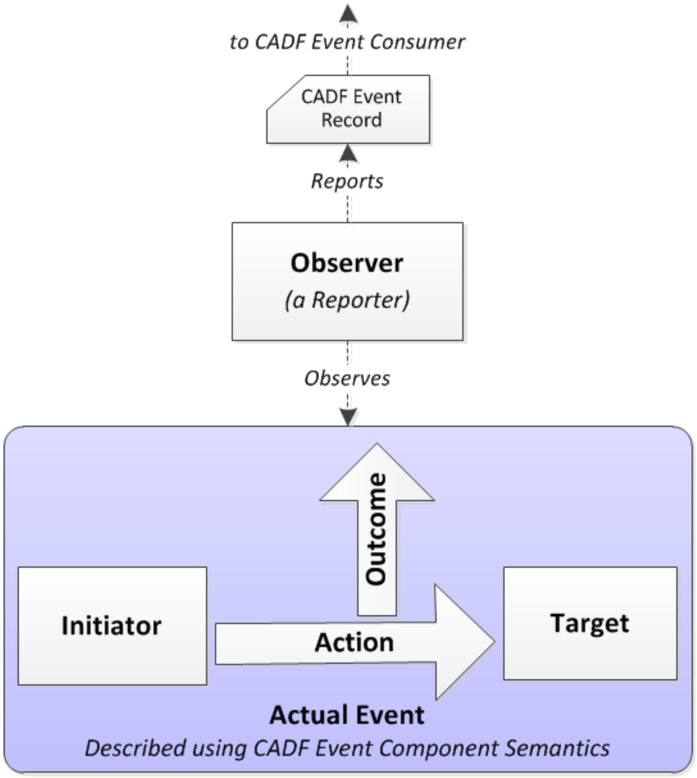
The CADF Event Model applies semantics to the activity and resources relative to the role they play in the actual activity (or event) that occurs within a cloud provider's infrastructure. These semantics are described in the table below as named components of the CADF Event Model.

Model Component	CADF Definition

REPORTER	A RESOURCE that contributes to the CADF Event Record. Note: There may be several REPORTERS that contribute to the CADF Event Record prior to it being presented to the end consumer.
OBSERVER	The first REPORTER that generates the CADF Event Record, either directly or indirectly, from the Actual Event.
INITIATOR	The RESOURCE that initiated, originated or instigated the event's ACTION, according to the OBSERVER.
ACTION	The operation or activity the INITIATOR has performed, attempted to perform or has pending against the event's TARGET, according to the OBSERVER
TARGET	The RESOURCE against which the ACTION of a CADF Event Record was performed, was attempted or is pending. Note: a TARGET can represent a plurality of target resources.
OUTCOME	The result or status of the ACTION of the observed event.

4.2.1 Conceptual event model

The following conceptual diagram shows basic components of the CADF Event Model and their interactions:



4.2.2 CADF Event Type

This specification recognizes that CADF Event Records may be used to communicate audit information to a consumer to fulfill different objectives or purposes. In addition, the CADF Event Model is designed to be extended and profiled to enable the CADF specification to be referenced or used in various audit applications. Therefore, the CADF Event Model describes a CADF Event Type property that is associated to the CADF Event Record. It is intended to be used by the CADF Event consumer to easily interpret the data fields in the CADF Event Record and understand any additional data that may be included in the record specific to that type of event.

Providing a "type" as part of the CADF Event Record is intended to clearly signal to the event consumer how to properly validate the CADF Event Record contents against requirements from the CADF Event Types defined in this specification or one of its profiles (by extension).

These basic event types reflect distinct perspectives of the event OBSERVER component and its purpose in reporting the event.

Event Component	CADF Definition
EVENTTYPE	A top-level classification of the CADF Event Record that is intended to communicates additional or more specific data and requirements.

4.2.2.1 *CADF Event Type values*

This specification defines the following basic CADF Event Type values:

CADF Event Type	CADF Definition
activity	Events that provide information on (attempted) actions against resources which may be subject to operational or business controls and policies.
monitor	Events that provide periodic statistical information or measurements on a resource or one of its attributes or properties. These types of events are often used as supporting information when evaluating compliance to a policy.

4.2.3 Reporter chain

Cloud provider architectures are generally layered in a way such that many Actual Events may occur at the lower layers which are close to the infrastructure components and services. Additionally, operational systems and processes may span many layers of the architecture, each with critical information that would be valuable to associate with audit events.

The CADF Event Model recognizes that many components may assist in constructing and surfacing the CADF Event Record before it is presented to the end consumer. These components can each be viewed as CADF Event Record REPORTERS each serving a specified role in raising the CADF Event Record as part of a sequential chain of REPORTER components.

The CADF Event Model includes a component called a "Reporter Chain" which is defined as follows:

Event Component	CADF Definition
REPORTERCHAIN	A record that includes the sequence of REPORTER components that handled the CADF Event Record.

Note that each CADF Event Record could have more than one REPORTER that handles the record within a provider's infrastructure and each MAY be listed in the REPORTERCHAIN at the discretion of the provider.

4.2.3.1 *CADF Reporter roles*

As described above, many REPORTER components may assist in constructing and surfacing the CADF Event Record before it is presented to the end consumer. In this specification, we will describe requirements based upon REPORTER roles which we define below.

This specification defines the following basic CADF Reporter Roles:

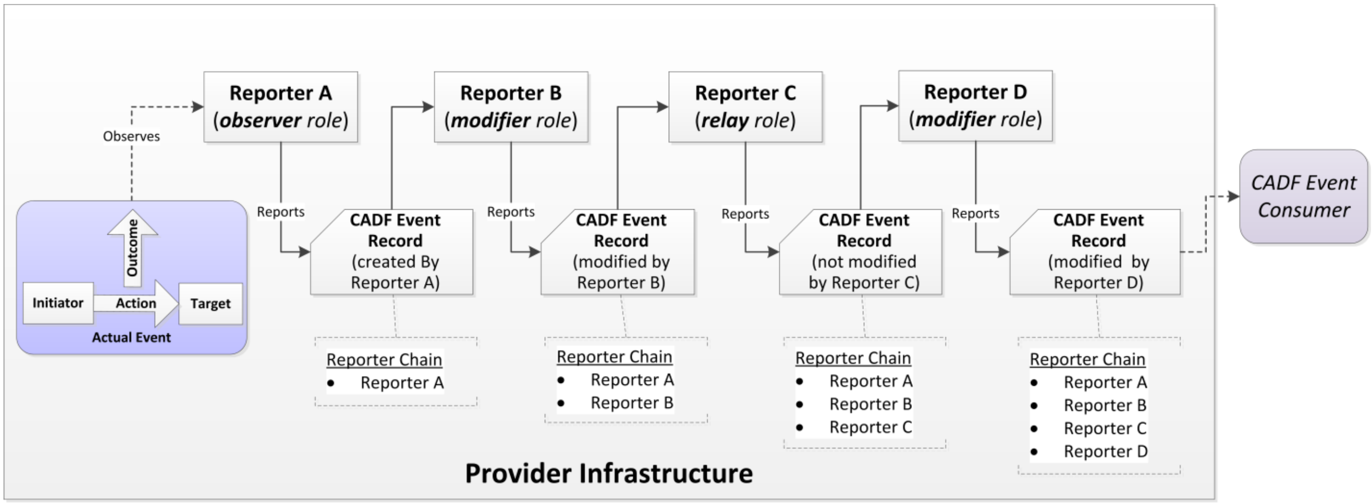
Reporter Role	CADF Definition
observer	A REPORTER that fulfills the role of OBSERVER. • There SHALL be one and only one REPORTER of this type per CADF Event Record.
modifier	A REPORTER that adds, modifies or augments information in the CADF Event Record for the purposes of normalization or federation.
relay	A REPORTER that passes the CADF Event Record to another REPORTER or to end record consumer without modifying the information in the CADF Event Record (with the exception of adding its own REPORTER entry in the REPORTERCHAIN).

4.2.3.2 *Example*

The following example shows a provider infrastructure that has an OBSERVER create a CADF Event Record that gets both modified and relayed by REPORTER components as it is moved across layers of the provider's architecture prior to getting presented to the end consumer of the record.

In the diagram, a flow showing the construction of a CADF Event Record is shown from left to right:

- Reporter A is the OBSERVER of the Actual Event and generates the CADF Event Record from its perspective by recording the required INITIATOR, TARGET, ACTION and OUTCOME entities and properties. Reporter A then adds itself as the first entry in the Reporter Chain of the CADF Event Record (with the CADF Reporter Role "**observer**") and passes the record to Reporter B.
- Reporter B receives the CADF Event Record and modifies it in order to augment the event's INITIATOR data with more detailed user account information. Reporter B then adds itself as a "**modifier**" (a CADF Reporter Role) to the event record's Reporter Chain after the entry for Reporter A and passes the CADF Event Record to Reporter C.
- Reporter C receives the CADF Event Record from Reporter B. Reporter C adds itself as the Reporter Chain after Reporter B's entry indicating it simply acted as a "**relay**" (another CADF Reporter Role) and performed no other modifications to the CADF Event Record. Reporter C passes the CADF Event Record to Reporter D.
- Reporter D receives the CADF Event Record from Reporter C. Reporter D "modifies" the event record to add CADF resource categorization information, and then adds itself as the last entry in the Reporter Chain (as the second "**modifier**" CADF Reporter Role entry) prior to presenting the CADF Event Record to the end CADF Event Consumer.


Reporter A (observer role)
Reporter B (modifier role)
Reporter C (relay role)
Reporter D (modifier role)
Observes
Reports
Outcome
Initiator
Action
Target
Actual Event
CADF Event Record (created By Reporter A)
CADF Event Record (modified by Reporter B)
CADF Event Record (not modified by Reporter C)
CADF Event Record (modified by Reporter D)
Reporter Chain
Reporter A
Reporter B
Reporter C
Reporter D
CADF Event Consumer
Provider Infrastructure

4.2.3.3 *Requirements on intermediate CADF Event Record completeness*

Every reporter SHALL produce a well-formed CADF Event Record. However, there is no indication in the CADF Event Record that the REPORTERCHAIN is closed: in other words, an CADF event record could be logged, and later on could be processed again by a new Reporter, thus extending its REPORTERCHAIN.

4.2.4 Additional Model Components

Different CADF Event Types introduce the need for additional model components which are introduced in this section.

4.2.4.1 *Measurements and Metrics*

Measurements are an optional component of the CADF Event Type, but are essential for any CADF Event Record that is classified as a "**monitor**" type event.

Event Component	CADF Definition
MEASUREMENT	An entity that contains statistical or measurement information for TARGET resources that are being monitored. .The measurement should be based upon a defined metric (a method of measurement).

4.2.4.1.1 *Requirements*

- CADF Event Records that are classified as "**monitor**" type events SHALL contain at least one valid set of MEASUREMENT data.
- Other types of CADF Event Records MAY contain one or more instances of MEASUREMENT data.

4.2.5 Resource classification

One of the key values of the CADF Event Model is that the action and the resources that participated in the Actual Event, in addition to being described in the CADF Event Record, must also be classified using values from CADF defined taxonomies included in this specification. These CADF Taxonomies are designed to be hierarchical and are extensible by profiles of this specification.

Resource classification provides the following benefits:

- Enables consumers to construct action or resource-based queries using CADF defined interfaces to obtain sets of events (typically in the form of logs or reports) that will produce similar results when used against various providers.
- Supports comparison of similar resource types across multiple providers and platforms.

4.3 Examples of mapping typical events to CADF Event Model

This section describes some typical audit event use cases along with examples showing how Actual Event information could be mapped to the CADF Event Model and semantics. These use cases were selected to show how different types of events would be identified and mapped from the perspective of the OBSERVER.

4.3.1 Use case: "Auditing access to a controlled resource"

In this example, a cloud provider has a software component that manages identity and access control that we will call an "identity management service". This service is a subclass of a "security" service (as shown in the CADF Resource Taxonomy) which is required by the provider's security policy to prove *security control compliance* by logging all user "login" actions against all servers within their infrastructure using the CADF Event Record format.

Please note that in this use case:

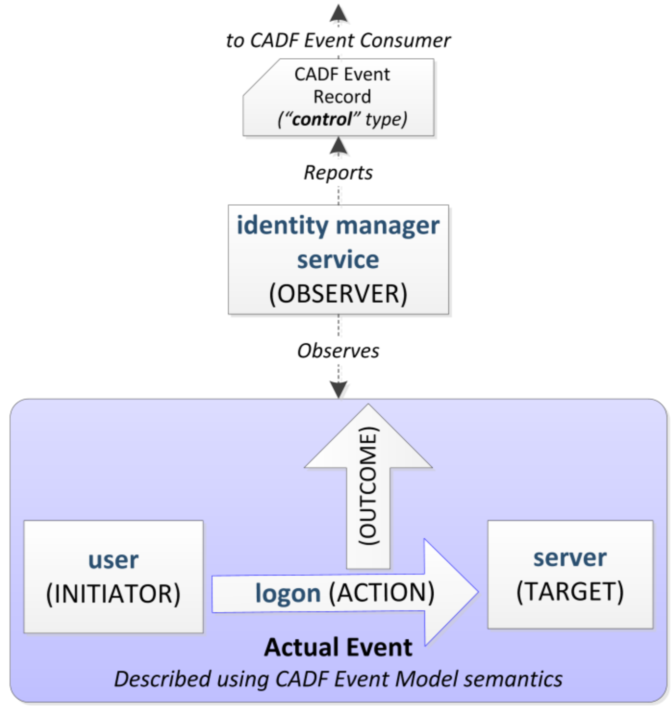
- The EVENTTYPE is "**activity**".
- The OBSERVER's purpose is to report on a security ACTION.

4.3.1.1 *Use case applied to CADF Event Model*

The following table shows a mapping of the significant actors and elements described in this use case to the conceptual CADF Event Model:

OBSERVER	EVENTTYPE	INITIATOR	ACTION	TARGET	OUTCOME	MEASUREMENT
identity management service	**activity** (e.g., a security or access control event)	**user** *(connecting from some client which would be additional data attached to initiator)*	**logon** *(an operation, which is being monitored for security compliance purposes)*	**server** *(a CADF Resource Taxonomy value)*	***Any valid CADF Outcome value*** (e.g., success, failure, etc.)	**N/A** (not required for "**activity**" type events)

The following diagram shows the same mapping from the table, but in graphical format:


to CADF Event Consumer
CADF Event Record ("control" type)
Reports
identity manager service (OBSERVER)
Observes
(OUTCOME)
user (INITIATOR)
logon (ACTION)
server (TARGET)
Actual Event
Described using CADF Event Model semantics

4.3.2 Use case: "Periodic monitoring resource status"

In this example, a cloud provider has software monitoring agents installed on every server that it makes available as an IaaS resource to its customers. These agents are required to provide periodic *informational status* of each server's CPU utilization along with metric data to their operations management software using the CADF Event Record format.

Please note that in this use case:

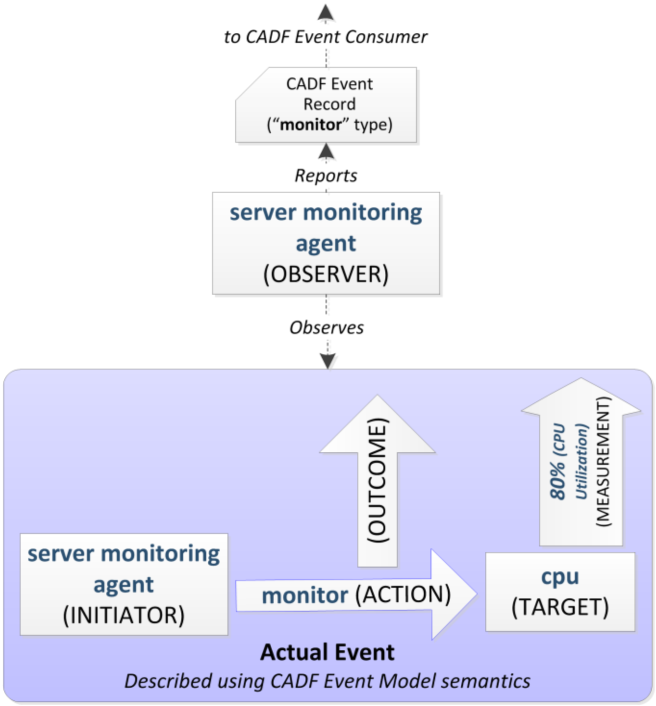
- The TARGET is the resource being monitored
- The INITIATOR is performing the monitoring function and is also the OBSERVER as it reports the event.
- The OBSERVER's purpose is to monitor a server's CPU (classified by the CADF Resource Taxonomy as "cpu"); therefore, the ACTION is set to the "**monitor**" value.

4.3.2.1 *Use case applied to CADF Event Model*

The following table shows a mapping of the significant actors and elements described in this use case to the conceptual CADF Event Model:

OBSERVER	EVENTTYPE	INITIATOR	ACTION	TARGET	OUTCOME	MEASUREMENT
server monitoring agent	**monitor**	**server monitoring agent**	**monitor**	**cpu**	***Any valid CADF Outcome value*** (e.g., success, failure, etc.)	**80%** *(CPU utilization)*

The following diagram shows the same mapping from the table, but in graphical format:


to CADF Event Consumer
CADF Event Record ("monitor" type)
Reports
server monitoring agent (OBSERVER)
Observes
(OUTCOME)
80% (CPU Utilization) (MEASUREMENT)
server monitoring agent (INITIATOR)
monitor (ACTION)
cpu (TARGET)
Actual Event
Described using CADF Event Model semantics

4.3.3 Use case: "Aggregation of resource status into an audit event"

In this example, a cloud provider has a Monitoring Server that collects CPU utilization information from server monitoring agents that are installed on every server that it makes available as an IaaS resource to its customers running application images.

The "monitoring server" summarizes these periodic measurements from the agents, by calculating an average utilization value and then generates a single *informational status* event that it sends to the provider's operations management software using the CADF Event Record format.

4.3.3.1 *Use case applied to CADF Event Model*

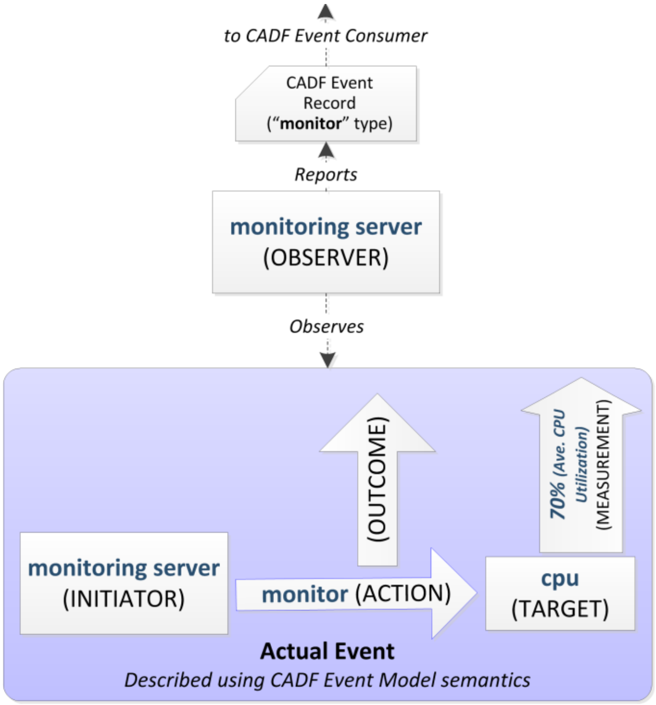
The following table shows a mapping of the significant actors and elements described in this use case to the conceptual CADF Event Model:

Please note that in this use case:

- The EVENTTYPE is "monitor".
- The OBSERVER's purpose is to monitor multiple servers' CPU utilization and provide summary events.

OBSERVER	EVENTTYPE	INITIATOR	ACTION	TARGET	OUTCOME	MEASUREMENT
monitoring server	**monitor**	**monitoring server**	**monitor**	**cpu** *(a set of CPUs from multiple servers)*	***Any valid CADF Outcome value*** (e.g., success, failure, etc.)	**70%** *(Average CPU utilization percentage data for all CPUs)*

The following diagram shows the same mapping from the table, but in graphical format:


to CADF Event Consumer
CADF Event Record ("monitor" type)
Reports
monitoring server (OBSERVER)
Observes
(OUTCOME)
70% (Ave. CPU Utilization) (MEASUREMENT)
monitoring server (INITIATOR)
monitor (ACTION)
cpu (TARGET)
Actual Event
Described using CADF Event Model semantics

4.3.4 Use case: "Auditing compliance of resource monitors"

In this example, a cloud provider has software monitoring agents installed on every server that it makes available as an IaaS resource to its customers. These agents may themselves be considered "controlled resources" within the provider infrastructure and are required by the provider's operational policy to send audit events to show that their activities are in compliance when performing operations (e.g., a "read") against the resources they are monitoring (or observing) using the CADF Event Record format.

Please note that in this use case:

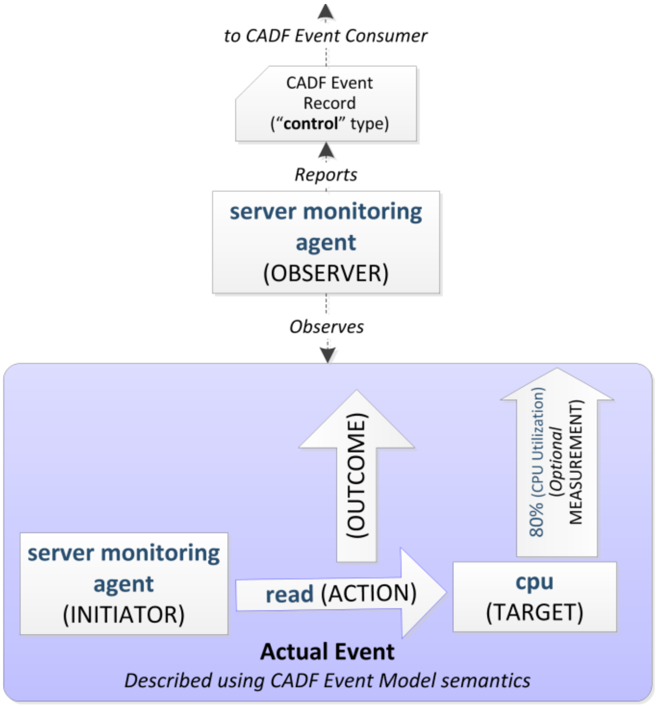
- This event record represents an alternative view of the same ACTUAL EVENT as described in the previous use case (above) titled "Periodic monitoring resource status", but is OBSERVED from a different perspective.
- The EVENTTYPE is "**activity**".
- The OBSERVER's purpose is to report on the "read" ACTION for compliance reasons.
- The MEASUREMENT is an optional property that could be included in the event record.

4.3.4.1 *Use case applied to CADF Event Model*

The following table shows a mapping of the significant actors and elements described in this use case to the conceptual CADF Event Model:

OBSERVER	EVENTTYPE	INITIATOR	ACTION	TARGET	OUTCOME	MEASUREMENT
server monitoring agent	**activity**	**server monitoring agent**	**read**	**cpu**	***Any valid CADF Outcome value*** (e.g., success, failure, etc.)	***Optional Value*** *(e.g. 80%)*

900 The following diagram shows the same mapping from the table, but in graphical format:


to CADF Event Consumer
CADF Event Record ("control" type)
Reports
server monitoring agent (OBSERVER)
Observes
(OUTCOME)
80% (CPU Utilization) (Optional MEASUREMENT)
server monitoring agent (INITIATOR)
read (ACTION)
cpu (TARGET)
Actual Event
Described using CADF Event Model semantics

901

5 Data model and schema conventions

5.1 Aliases for domain and namespace URI values

This specification will support domain-specific entity or property values to uniquely identify or tag events, reference classification systems, taxonomies, schemas and for other purposes.

In this specification, universal identification of these types of values will be done via attribution using domain and instance specific URI values which ensure that when data is federated there is no ambiguity as to which domain has defined the data.

In order to improve processing performance and reduce data size for storage and transmission of event data, the definition of domain and namespace URI "aliases" will be supported for use in property values.

5.1.1 Requirements

- Any alias name for a domain or namespace URI value that is defined within this specification SHALL be considered reserved for the sole use by this specification.
- Extensions or profiles of this specification SHALL NOT mask or redefine any alias name (or its corresponding URI value) which is defined in this specification
- Alias names SHALL be unique within the scope of any CADF Entity.
    - An alias name MAY be defined within a top-level CADF Entity. This permits the alias to be referenced repeatedly within that entity's scope.
- Any alias reference that is used within the scope of a CADF Entity SHALL not be disassociated from its alias definition.

5.2 Namespaces and namespace aliases

The following table lists the namespaces that are used in this specification along with their referenced specifications. One of the types of aliases described above would be a namespace alias that can be used as a prefix for a URI. The choice of any namespace prefix is arbitrary and not semantically significant.

Alias	Namespace	Specification
cadf	http://schemas.dmtf.org/cloud/audit/1.0/	The CADF Namespace. It is used to represents this specification
xs	http://www.w3.org/2001/XMLSchema	XML Schema

5.2.1 Requirements

- The CADF Namespace alias for this specification's schema SHALL be the value "cadf" (i.e. only the lowercased characters within the quotes).
    - The CADF Namespace alias SHALL be used for XML namespace prefixes.
- The CADF Namespace SHALL appear in the target namespace for the XML schema that represents the definitions and requirements of this specification.
- The namespace for the data schema defined in this specification is consistent with DMTF specification DSP4009 and SHALL be the following value:
    - http://schemas.dmtf.org/cloud/audit/1.0/

5.2.2 Usage example

The following example shows the proper use of this specification's namespace for XML schema:

```
<xs:schema
   xmlns="http://schemas.dmtf.org/cloud/audit/1.0/"
   targetNamespace="http://schemas.dmtf.org/cloud/audit/1.0/"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified">
```

5.3 URI space

5.3.1 Requirements

- CADF Event Model consumers SHALL NOT make assumptions about the layout of the URIs or the structures of any URI used in this specification, extensions or profiles,

5.4 Entity naming conventions

5.4.1 Requirements

All schema names (e.g., entity, data type, element, property, operation, parameter, etc.) defined by this specification, or defined via an extension, SHALL adhere to the following rules:

- Entity names SHALL be treated as case sensitive
- Entity names SHALL only use the following set of characters:
    - Upper case ASCII (U+0041 through U+005A)
    - Lower case ASCII (U+061 through U+007A)
    - Digits (U+0030 through U+0039)
    - Underscore (U+005F)
- The first character of an Entity Name SHALL NOT begin with the following set of characters:
    - Digits (U+0030 through U+0039)

5.4.2 XML naming requirements

In order to avoid naming collisions with other XML data schemas the following requirements are specified:

- All elements in this specification's XML Schema SHALL be qualified by a namespace, as per [XMLSchema0], to avoid collisions with other data schemas that may be encapsulated within this specification's schema
- All extensions and profiles of this specification that define additional properties (represented as XML attributes) to CADF defined entities (represented as XML elements) SHALL be qualified by the namespace that defines the additional properties. This is intended to avoid collisions for common attribute names and any conflicts with CADF defined property names.

5.5 Property constraints

Each entity (e.g., element or property) described in this schema is augmented by a set of constraints that further qualify the entity being defined.

5.5.1 "Required" constraint:

The schema definition tables include a "required" column that indicates whether the associated data type, entity or property (and its corresponding feature or value) is required. Possible values are:

- **Yes** - indicates that the specified entity or property is required and SHALL be present.
- **No** - indicates that the specified entity or property is optional and MAY be present.
- **Dependent** - indicates the specific entity or property SHALL or MAY be required depending upon some condition described by the property. For example, a format dependency may be described on a per-entity or per-property basis when serializing in XML or JSON.

5.6 Format-specific representations

This specification is written to be neutral to transmission format since format profiles of this specification are permitted. However, this specification acknowledges that both XML, as the normative format for federation, and JSON, as a popular format used by cloud providers, need special consideration in this specification. This section attempts to provide requirements and guidance for expressing this specification's entities, data types and properties in either XML or JSON.

5.6.1 Entity type URIs

The specification supports serialization of top-level entity instances (or approved extensions of them) with the following conventions:

5.6.1.1 *Requirements*

XML serialization:

Any top-level entity, when serialized as an XML element with name equal to the Entity name, MAY include the property "typeURI" with the defined "Entity Type URI" value for the entity being serialized. For example:

```
<Entity typeURI="xs:anyURI" simpleproperty="value">
    ...
</Entity>
```

JSON serialization:

Any top-level entity, when serialized as a JSON object SHALL include a "typeURI" property with the defined "Entity Type URI" value as defined for the CADF Entity being serialized. For example:

If an entity is expressed by itself it would appear as follows:

```
{
     "typeURI": "URI string",
     "simpleproperty": "value",
     ...
}
```

or as follows if the entity is itself a named property of another data type:

```
{
    "<Entity's propertyname>": {
     "typeURI": "URI string",
     "simpleproperty": "value",
     ...
    }
}
```

5.6.1.2 *Notes*

Please note that although the "typeURI" property may be included in XML serializations for CADF Entities, it is not recommended or necessary to identify the Entity schema type since it is implicit from the element name and XML schema and therefore not recommended.

5.6.2 Language identification

This specification may include optional descriptive or informational elements that contain human-readable text (data). In order for processors to correctly select such elements against a specified set of desired language(s), attributing normative language values to such elements is important. The presence of this property will assist in the creation of views optimized for the language of the end consumer of an event, report or log.

5.6.2.1 *Requirements*

When language identification is indicated:

- for language identification in XML, XML elements that provide human readable, text based information as their value data SHALL use the W3C special attribute (property) “xml:lang” to specify the language where necessary. [W3C-XML]
- for language identification in JSON, JSON structures that provide human readable, text based information SHALL include the CADF defined property "lang" with permitted values as specified by [W3C-XML].

5.6.2.2 *Examples*

XML serialization:

Language identification in XML SHALL be accomplished with the use of the "xml:lang" attribute:

```
<Element xml:lang="en">
      ...
</Element>
```

JSON serialization:

Language identification for JSON objects SHALL be accomplished with the use of the "lang" property:

```
object: {
      "lang": "en",
      ...
}
```

5.6.3 Rules for XML and JSON format representation

This section describes how the CADF Entities, data types and properties defined in this specification would be translated to XML and JSON formats.

5.6.3.1 *Requirements*

The following rules SHALL be applied when representing CADF Entities, data types and properties in XML:

- Any CADF Entity, and any of its extensions or derivations, SHALL be expressed as an XML element where the XML element name is the same as the entity's name.
- Any property defined as a CADF complex data type, and any of its extensions or derivations, SHALL be expressed as an XML element where the XML element name is the same as the property name defined for that data type and its composite properties follow the same expression rules recursively (and are expressed as attributes or nested elements).
- Any property defined as a basic data type or CADF basic type and its corresponding value SHALL be expressed as an XML attribute-value where the XML attribute's name is the same as the property name defined for that data type and the XML attribute's value SHALL conform to the defined values for that property and XML schema data type.
- Any property defined as a CADF Entity or CADF complex data type, and any of its extensions or derivations, that does not have any properties that are CADF complex data types SHOULD be expressed as a self-closing XML element.

The following rules SHALL be applied when representing CADF Entities, data types and properties in JSON:

- Any CADF Entity, and any of its extensions or derivations, SHALL be expressed as a JSON object.
- Any CADF Entity, and any of its extensions or derivations, SHALL have a JSON name-value pair where the JSON pair's name (string) SHALL be "typeURI" and pair's value is the specified "Entity Type URI" for that CADF Entity.
    - Note that this requirement is also explained in the section titled "Entity Type URIs" above.
- Any CADF complex data type, and any of its extensions or derivations, SHALL be expressed as a JSON object where the JSON object's name is the same as the property name defined for that data type.
- Any basic data type or CADF basic type and its corresponding value SHALL be expressed as a JSON name-value pair where the JSON pair's name (string) is the same as the property name defined for that data type and pair's value SHALL conform to the defined values for that property and its schema type.

5.6.3.2 *Examples*

If a CADF Entity and its basic and complex properties are defined as follows:

Entity Name	***Entity1***		
Property Name	**Property Type**	**Required**	**Description**
simple1	xs:string	Yes	A required property of the basic XML "string" type.
simple2	cadf:Identifier	No	An optional property of the CADF basic "identifier" type.
complex1	<namespace>:<ComplexTypeA>	Yes	A required complex type (see table below).

and whose complex type is defined as follows:

Complex Type Name	***ComplexTypeA***		
Property Name	**Property Type**	**Required**	**Description**
simpleA	xs:string	Yes	A required property for the sample complex type. Whose value is another basic XML "string" type.

would have the following format serializations:

XML serialization:

Showing the preferred serialization using a self-closing XML element:

```
<Entity1 simple1="some string" simple2="myscheme://mydomain/id/1234">
     <complex1 simpleA="another string"/>
</Entity>
```

JSON serialization:

Showing the preferred serialization using an JSON object name for the CADF Entity:

```
{
    "typeURI": "Entity1's specified Entity Type URI value",
    "simple1": "some string",
    "simple2": "myscheme://mydomain/id/1234",
    "complex1": {
        "simpleA": "another string"
    }
}
```

6 CADF Entities and data types

This section defines the CADF entities and data types that are necessary to ensure providers produce CADF specified event data in a normative fashion so that it can be properly aggregated, federated and searched to produce consistent logs and reports. These CADF data types will be referenced by the CADF data schema.

6.1 Extensibility mechanisms

This section describes extensibility mechanisms that can be applied to both to CADF Entities and CADF complex data types.

In this specification, CADF entities (and in some cases CADF complex data types) represent classes of resources that may vary significantly from one cloud environment to the other, yet are expected to share a same set of core properties for cross-domain comparison when auditing. In order to accommodate these considerations, this CADF data model provides ways to extend or augment these resources. The approach allows for associating additional data to entity or complex type instances, while providing enough meta-level description so that interoperability and profiling are possible.

Two extensibility mechanisms are used in the CADF data model, as indicated for each CADF Entity or complex data type:

- Derivation
- Attachments

6.1.1 Derivation

A CADF Entity (and in some cases CADF complex data types) will allow for additional user-defined properties. In other words, a new derived entity or data type can be defined, that contains additional properties in addition to the core properties defined in the original CADF Entity or data type. Such derived types are typically described as part of a specific profile of the CADF model. Several derivations may be defined for the same base CADF Entity, yet any processing or query that is possible over a base CADF Entity and its instances will also apply to its derivations.

To this end, derived entities and types also must derive their type name from the name of the base CADF Entity or type they derive from. This means that any CADF Entity or complex data type that is derivable contains a “typeURI” property which identifies the base CADF Entity type and any derived type would be identify itself within the same property by adding an additional segment name to the base type's "typeURI” property.

As for entities, the existence of a "typeURI" property in a CADF complex data type indicates that this complex type is derivable.

For example, a cloud provider may decide to derive different resource types from the complex CADF Resource type defined in this model in order to match different types of resources in its environment.

The typeURI value for the derived provider Resource type may extend the typeURI value as specified for the base CADF Resource type (i.e., "http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/").

Derived entities or data types will typically be associated with an XML schema extended from the original, yet the instances of such derived entities must validate against the original schema.

6.1.2 Attachments

Another way to extend a CADF Entity or complex data type is to associate attachments to it. An attachment is a container for data or “content” that may follow any structure – from an atomic type to a complex hierarchy. However, it is desirable for processing and interoperability, that the type – or structure - of the content be identified by a simple value. To this end the attachment also contains a “content type”, i.e. a URI that identifies the kind of content. When XML is used for the content, the value of the content type MUST always be associated with a unique XML schema that the content must validate against.

The data type used to implement Attachments for CADF entities is described in section "Attachment Type".

6.1.2.1 *Attachment notes*

Attachments are intended to be used for inclusion of domain-specific, informative, or descriptive information. Information in attachments should NOT be critical to a basic understanding of the Event Record – indeed, any and all attachments should be considered optional and the generator should assume that downstream consumers may drop any and all attachments to save space.

Attachments may be generated and attached by the original CADF Event OBSERVER or by any downstream REPORTER. For example, an access control mechanism may report that it allowed access to a resource based on an opaque SAML token, then a downstream Reporter may reverse-lookup that token, resolve it to the identity of a person, and attach that identity to the Event Record.

Attachments may also contain state information about a resource – e.g. a list of attributes about that resource at the time the event occurred. This information can be highly useful for understanding the context in which the activity took place, but again the attachment must be considered optional, and in general such state information should be limited to highly-relevant pieces of data to avoid inflated events and logs that become unprocessable.

6.2 Basic data types

This section describes basic data types for typing property values when specifying data schema within this document. In general, these data types are not specific to CADF, but each may have specific constraints or requirements that are necessary when representing CADF data.

6.2.1 General requirements

- The simple data types defined below SHOULD be used wherever possible by extensions and profiles of this specification.
- Any constraints on the specific ranges allowed for any particular property SHOULD be specified by that property's definition.

6.2.2 boolean

A value as defined by xs:boolean per [XMLSchema2], with the exception that the only allowable values are either "true" or "false". The value is case sensitive.

6.2.3 integer

A value as defined by xs:integer per [XMLSchema2].

6.2.4 double

A value as defined by xs:double per [XMLSchema2].

6.2.5 string

A value as defined by xs:string per [XMLSchema2].

6.2.6 duration

A value as defined by xs:duration per [XMLSchema2].

6.2.6.1 *Lexical representation*

```
'-'? 'P' n 'Y' n  'M' n 'D' 'T' n 'H'  n 'M' n 'S'
```

- Where "n" represents numeric values:

```
[0-9]+
```

- Where the 'n' value for S (seconds) permits numeric values in fractions of a second:

```
[0-9]+(\.[0-9]+)?
```

- A preceding '-' (minus) sign is permitted to indicate a negative duration.

6.2.7 URI

Note that the base format and syntax of properties of type “URI” are defined by RFC 3986 [IETF RFC 3986]. The CADF provides some additional requirements on URIs types below.

6.2.7.1 *Additional URI Requirements*

The following additional constraints SHALL apply to URI typed data in this specification, extensions or profiles:

- URIs that are intended to be identifiers SHALL not be relative URIs unless a valid alias is defined in the containing entity (e.g., a URI defined in a CADF Log could be used as a valid alias when composing a CADF Identifier in place of a absolute URI).
- Relative URIs SHALL NOT start with a "/", otherwise the URI is assumed to be absolute and no URI processing (to determine the full path) will be performed.

6.2.8 Basic type translation to JSON from XML

This specification references basic data types as they are defined by XML schema. The following table shows how these basic data types would translate from XML to JSON:

XML type	JSON type
xs:boolean	*boolean*
xs:integer	*number*
xs:double	*number*
xs:string	*string*
xs:anyURI	*string*
xs:duration	*string*

6.3 CADF basic data types

This section defines basic CADF data types. These types may be used when defining complex CADF data types and entities.

6.3.1 Identifier type

This data type is defined to normatively describe identifiers as part of the CADF Event Record.

6.3.1.1 *Design considerations*

In order to effectively audit any form of compliance, it is essential to clearly identify the precise resources and actors that are performing activities and represent them in event records.

In addition, any identity must be composed such that is reasonably guaranteed to be "globally unique" so that, when CADF Event Records are aggregated from multiple sources, identities do not "collide" and result in an audit logs or reports where it is not clear which resource or actor actually performed the action and in where (e.g., provider domain).

Since CADF Logs and Reports may contain many CADF Event Records each with multiple identifiers, it is desirable that the identifier format permit composition to prevent duplication of commonly repeated components.

6.3.1.2 *Requirements*

This specification defines an Identifier type that is based upon the Uniform Resource Identifier Reference (URI) as specified in IETF RFC 3986. Any value that represents a CADF Identifier type in this specification, its extensions or profiles SHALL adhere to the following requirements:

Type name

Name	**Identifier**

Syntax requirements

- CADF Identifiers SHALL adhere to the URI Syntax as defined by in IETF RFC 3986 with additional requirements listed below.
  - For convenience, the syntax components from IETF RFC 3986 are as follows:

```
scheme ":"  hier-part [ "?" query ] [ "#" fragment ]
```

  - and the hierarchical component (or "hier-part") is defined as follows:

```
hier-part = "//" authority path-abempty
              / path-absolute
              / path-rootless
              / path-empty
```

- CADF Identifiers that SHALL include a valid "authority" as defined by IETF RFC 3986 as part of the URI.
  - This means that the "authority" component SHALL be present and SHALL NOT be empty.
  - By corollary this also means that the "path-abempty" component SHALL NOT be permitted as an option.

  - The value of the "authority" SHOULD be provided by registry that can guarantee the uniqueness of the value.
- CADF Identifiers SHALL be composed only of characters from the US-ASCII coded character set and SHALL only use unreserved characters
  - This means that characters from other character sets SHALL be encoded into the US-ASCII character set as described by IETF RFC 3986.

6.3.1.3 *Lexical representation*

- The following is the required Lexical representation of the CADF Identifier type described using IETF RFC 3986 components as above:

```
[ scheme ":" ]  hier-part [ "?" query ] [ "#" fragment ]
```

  - where the hierarchical component (or "hier-part") SHALL be as follows:

```
hier-part = "//" authority
              / path-absolute
              / path-rootless
              / path-empty
```

Please note that the CADF identifier data type is compatible with the xs:anyURI data type described by XMLSchema2.

6.3.1.4 *Best practices*

- When CADF Identifier values include a protocol schemes (such as "http"), it SHOULD NOT be assumed that this represents a resource that can be accessed by the identifier value.
- CADF Identifier “authority” names SHOULD be the same for resources managed by the same provider domain (i.e. the same management domain) and SHOULD NOT change frequently.

6.3.1.5 *Examples*

Example 1: "CADF Identifier using an absolute URI"

In this example, the CADF Identifier is composed as an **absolute** URI that includes the optional scheme component (i.e. "http"), the cloud provider's registered domain name and followed by a hierarchical path that describes an instance (e.g., "4321") of an application server (e.g., "appserver") within the provider's infrastructure.

```
http://publiccloud.com/datacenter1/appserver/4321
```

Example 2: "CADF Identifier using a relative reference URI"

This example represents the same resource as in Example 1 above; however, the CADF Identifier is composed as a **relative reference** URI (i.e. it has no scheme).

```
//publiccloud.com/datacenter1/appserver/4321
```

Example 3: "Provider-specified scheme"

In this example, the CADF Identifier is composed as an **absolute** URI that is further classified by provider specified scheme (e.g., "myscheme"). This scheme is followed by the cloud provider's domain name of the cloud provider followed and followed by a hierarchical path that identifies a unique user managed by the provider.

```
myscheme://mycloud.com/account/1234/user/5678
```

6.3.2 Path type

This section describes how to represent values from CADF Taxonomies when used by properties that classify CADF Event Records as path values from hierarchical taxonomies.

6.3.2.1 *Design considerations*

This specification includes CADF classification taxonomies that are designed to identify, request and collect CADF Event Records from a provider that may be relevant to proving compliance against various compliance frameworks.

The values within these classification taxonomies are designed as hierarchical trees where nodes defined at greater levels representing a more granular classification. Individual nodes (or values) with the tree can be identified by its unique path constructed by combining the node values from the root node of the tree to its node value along with any intermediate node values traversed.

The design of this type needs to represent these classification values as paths in a way that is compatible with popular path traversal and search mechanisms such as XPath and XQuery yet be simple enough to support other, non-XML tooling.

6.3.2.2 *Requirements*

The CADF Path uses URI references to identify CADF Taxonomy values with certain URI Syntax components given the specific additional requirements listed below.

Any value that represents a CADF Path type in this specification, its extensions or profiles SHALL adhere to the following requirements:

Type name

Name	**Path**

Syntax requirements

- CADF Path values SHALL adhere to the URI Syntax as defined by in IETF RFC 3986 with additional requirements listed below.
  - For convenience, the syntax components from IETF RFC 3986 are as follows:

    ```
    scheme ":"  hier-part [ "?" query ] [ "#" fragment ]
    ```

  - and the hierarchical component (or "hier-part") is defined as follows:

    ```
    hier-part = "//" authority
                 / path-absolute
                 / path-rootless
                 / path-empty
    ```

  - where the "path-rootless" component is defined as follows:

    ```
    path-rootless = segment-nz *( "/" segment )
    ```

- CADF Paths SHALL NOT contain the query component of the URI Syntax.

- CADF Paths SHALL NOT contain the optional fragment component of the URI Syntax.
- CADF Paths SHALL contain at least one valid non-zero length path segment (as defined by IETF RFC 3986 path component named "segment-nz").
    - This means that the URI Syntax component "path-rootless" SHALL contain at least one valid "segment-nz" value.
    - This means that the URI Syntax component "path-empty" SHALL NOT be permitted.
    - By corollary, this means "empty", "blank" or zero-length values SHALL NOT be permitted.
- if (1) the "selected-node-value" is a direct child node of the "root-node-value" AND the (2) "root-node-value" for a specific taxonomy is understood or established based upon the context where it is being used then the "selected-node-value" MAY appear by itself.

Absolute path requirements

- Absolute CADF Paths SHALL have the URI Syntax "scheme" component value set to the following value:

```
cadf
```

- Absolute CADF Paths SHALL begin with the URI Syntax "authority" and "path-absolute" components set to the following value:

```
//schemas.dmtf.org/cloud/audit/1.0/taxonomy/
```

Relative path requirements

- Relative CADF Paths MAY be permitted by properties in this specification where the property clearly specifies it MAY be used and also declares that CADF Path's "scheme", "authority" and "path-absolute" are assumed.
- Relative CADF Paths MAY include the optional URI Syntax scheme value (i.e. the value "cadf") along with a ":" (or colon) character.

6.3.2.3 *Lexical representation*

- The following is the required Lexical representation that SHALL be used for CADF Path type values:

```
[ "cadf:" ] [ "//schemas.dmtf.org/cloud/audit/1.0/taxonomy/" ] path-rootless
```

    - where the "path-rootless" component is defined as follows:

```
path-rootless = segment-nz *( "/" segment )
```

6.3.2.4 *Best practices*

- Audit logs and reports often contain large numbers of event records; therefore It is encouraged, wherever possible, to use the shortest length **Relative Path** form of the CADF Path possible for the document or context where the CADF Event Record is being used.

6.3.2.5 *Examples*

Example 1: "Relative path representation for the CADF Outcome Taxonomy"

In this example, the event's outcome was a "Failure". Since the property "code" clearly establishes the value as coming from the CADF Outcome Taxonomy and the node for "failure" is a direct child node of the outcome taxonomy root node, we may express the value using a **Relative Path**.

```
<Event
```

```
     ...
     outcome="failure"
     ...
/>
```

Example 2: "Relative path representation for the CADF Resource Taxonomy"

In this example, a CADF Event Record that contains a TARGET resource, in this case a database resource, that is categorized using the CADF Resource Taxonomy using a **Relative Path** representation within the CADF Path type for the "typeURI" property:

```
<Event
     ...
     <target typeURI="storage/database"/>
     ...
/>
```

Please note this **Relative Path** representation is the preferred format and is encouraged over **Absolute Path** representation wherever possible.

Here is the same example, but it explicitly includes the optional scheme prefix for CADF Taxonomies:

```
<Event
     ...
     <target typeURI="cadf:storage/database"/>
     ...
/>
```

Example 3: "Absolute path representation for the CADF Resource Taxonomy"

This example is the same as Example 2 (above), but instead expresses the "typeURI" as an **Absolute Path** representation within a CADF Path type:

```
<Event
     ...
     <target
        typeURI="cadf://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/storage
        ...
     />
     ...
/>
```

Please note that although **Absolute Path** representation is permitted, it is considered redundant from being used within the scope of a CADF Event Record. Therefore **Absolute Path** representation is not recommended when a **Relative Path** representation is possible.

6.3.3 Timestamp type

This data type is defined to normatively describe timestamps as part of the CADF Event Record.

6.3.3.1 *Design considerations*

Proper representation of date and time is critical in order to reliably compose a complete audit trail (activity stream) from multiple federated sources. The format used to assign date and time to (or timestamp) auditable event actions must be unambiguous in proving compliance relative to geographic and regional considerations. Therefore, a primary requirement on the format is that it must retain reference to the local time where any auditable action occurred.

Additionally, it is known that timestamp values will be routinely used to create composite audit reports and logs (or views) from disparate audit event sources accumulated using federation techniques. This places further requirements that any timestamp format need to be concise and easily comparable regardless of the event's source.

6.3.3.2 *Requirements*

This specification defines a Timestamp type that is based upon the xs:dateTime as per [XMLSchema2]. Any entity (or property) value that represents a Timestamp type in this specification, its extensions or profiles SHALL adhere to the following requirements:

Type name

Name	**Timestamp**

Syntax requirements

- The dateTime portion of Timestamp typed values SHALL adhere to the Lexical representation as per [XMLSchema2]; section 3.2.1.7 "Lexical representation".
  - *Lexical representation:*

```
yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)
```

- The Time Zone Designator (TZD) portion of the Timestamp typed values SHALL adhere to the Lexical representation as per [XMLSchema2]; section 3.2.7.3 "Timezones" and SHALL always be expressed as a UTC offset.
  - *Lexical representation:*

```
('+' | '-') hh ':' mm
```

- The character 'Z' for Time Zone Designator (TZD) SHALL NOT be used. If a Timestamp typed value indicates an event action that actually occurred in a region where the local time UTC offset is actually zero (or 'Zulu' time), a following fully qualified TZD SHALL be used.
  - *Example:*

```
('+' | '-') 00:00
```

- If the time in UTC is known, but the offset to local time is unknown, the TZD SHALL be represented with an offset of "-00:00". This differs semantically from an offset "+00:00", which implies an actual UTC time zone designation.
  - Note: This requirement aligns with the representation described in [RFC 3339]
- Any constraints on the specific ranges allowed for any particular property SHALL be specified by that property's definition.

6.3.3.3 *Lexical representation*

The following is the required Lexical representation of the Timestamp type used in this specification; all Timestamp typed values SHALL be formatted accordingly:

```
yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)('+' | '-') hh ':' mm
```

Please note again that the UTC offset is always required (not optional) and the use of the character 'Z' (or 'Zulu' time) as an abbreviation for UTC offset +00:00 or -00:00 is NOT permitted.

6.3.3.4 *Examples*

Example 1: "New York City, United States during Eastern Standard Time (EST) or UTC-05:00"

During the period when Eastern Standard Time (EST) is in effect, the UTC offset for New York City would be UTC minus five hours or UTC-05:00. An example of a valid Timestamp typed value for NYC during EST would be:

```
2012-02-25T09:00:00-05:00
```

This above timestamp represents the date February 25th, 2012 at 9:00 AM (EST) local time in New York City.

Example 2: "New York City, United States during Eastern Daylight Time (EDT) or UTC-04:00"

During the period when Eastern Daylight (savings) Time (EDT) is observed, the UTC offset for New York City would be UTC minus four hours or UTC-04:00. An example of a valid Timestamp typed value for NYC during EDT would be:

```
2012-03-22T13:00:00-04:00
```

This above timestamp represents the date March 22nd, 2012 at 1:00 PM (EDT) local time in New York City.

Example 3: "Dublin, Ireland during Greenwich Mean Time (GMT) or UTC+00:00"

During the period when Standard Time is observed, the UTC offset for Dublin is zero or UTC minus zero hours or UTC-00:00. An example of a valid Timestamp typed value for Dublin when GMT time is observed would be:

```
2012-03-17T22:00:00+00:00
```

This above timestamp represents the date March 17th, 2012 at 10:00 PM (GMT) local time in Dublin.

Example 4: "Dublin, Ireland during Irish Standard Time (IST) or UTC+01:00"

During the period when Irish Standard Time (also called "summer time") is observed, the UTC offset for Dublin is UTC plus one hour or UTC+01:00. An example of a valid Timestamp typed value for Dublin during IST would be:

```
2012-04-14T22:00:00+01:00
```

This above timestamp represents the date April 14th, 2012 at 10:00 PM (IST) local time in Dublin.

Example 5: "Beijing, China; China Standard Time (CST) or UTC+08:00"

The UTC offset for Beijing, China, which does not observe daylight savings time, is UTC plus eight hours or UTC+08:00. An example of a valid Timestamp typed value for Beijing would be:

```
2012-06-28T08:00:00+08:00
```

This above timestamp represents the date June 28th, 2012 at 8:00 AM (CST) local time in Beijing.

6.3.3.5 *Notes*

This specification seeks to provide a discrete format (or profile) of the xs:dateTime type, as per [XMLSchema2], that resolves any ambiguity for auditing purposes. The xs:dateTime type itself is based upon ISO 8601:2004(E). [ISO 8601:2004], and can easily be mapped to from applications that use the following format specifications:

- ISO 8601:2004(E). [ISO 8601:2004]:
  - Section 4, "Date and time representations".
  - Specifically the representation of UTC time in section 4.2.5.2 "Local time and the difference from UTC".
- DMTF CIM Infrastructure Specifications [DMTF DSP0004]:
  - Specifically, section 5.2.4 "Datetime Type", which also references the ISO 8601:2004 format.

6.4 CADF complex data types

This section defines the complex CADF data types. CADF complex data types differ from CADF entities in that they are always intended to be used as types for (complex) properties of CADF entities or other complex types. Unlike entities, they are not supposed to be accessed independently: the CADF interfaces assumes these complex types are always accessed in the context of the parent entities that contain them.

6.4.1 Array types

Properties that are arrays of a simple type, are defined using the notation “propertyType[]”, where "propertyType" is the data type name for each item of the array.

6.4.1.1 *Serialization example*

The following table shows a sample array property as it would be specified for a data type in this specification. For this example, this property is defined as an array of the CADF Attachment type:

Property Name	Type	Required	Description
attachments	cadf:Attachment[]	No	An optional array of type CADF Attachment.

The serialization of the array for this complex type would appear as follows:

XML example

```
<Entity>
    ...
    <attachments>
        <attachment contentType="xs:anyURI">
            <content>"xs:any"</content>
        </attachment>
        <attachment contentType="xs:anyURI">
            <content>"xs:any"</content>
        </attachment>
        ...
    </attachments>
</Entity>
```

JSON example

```
{
     ...,
     "attachments":
     [
         {
             "content": "xs:any",
             "contentType": "xs:anyURI"
         },
         {
             "content": "xs:any",
             "contentType": "xs:anyURI"
         }
     ]
}
```

6.4.2 Attachment type

6.4.2.1 *Design considerations*

The attachment type is used as one means to add domain-specific information to a CADF entity. Please see additional discussion on its use in the section titled "Extensibility Mechanisms".

6.4.2.2 *Requirements*

Any entity value that represents a CADF Attachment type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- The properties "contentType" and "content" SHALL have values that are consistent with each other.
    - This means that the "content" property's value SHALL be a valid value as described by the domain specification identified by the "contentType" value.
- The property "contentType" SHALL NOT have an "empty", "blank" or zero-length value.
- The property "content" SHALL NOT have an "empty", "blank" or zero-length value.
- Binary content types SHOULD be encoded as Base64 strings for inclusion under the “content" property".

6.4.2.3 *Notes*

- Any publicly-defined or custom content type may be included in an Attachment type as long the "typeURI" property value is valid and identifies the data in the "content" attribute.
    - For example, an attachment that includes a standard MIME types (such as “application/pdf”) can be included by extension of the "typeURI" set to "http://www.iana.org/assignments/media-types/application/pdf".

6.4.2.4 *Properties*

The following table describes the properties for the CADF Attachment type.

Name	**Attachment**		
Property	**Type**	**Required**	**Description**
typeURI	xs:anyURI	Yes	The URI that identifies the type of data contained in the "content" property.

content	xs:any	Yes	A container that contains any type of data (as defined by the contentType property).
name	xs:string	No	An optional name that can be used to provide an identifying name for the content.

6.4.2.5 *Serialization examples*

XML example

```
<Event id="myscheme://mydomain/id/1234">
    ...
    <attachments contentType="scheme://contenttype" name="foo">
        <content>
           ...
        </content>
    </attachments>
</Event>
```

JSON example

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "id": "myscheme://mydomain/id/1234",
    ...,
    "attachments":{
         "contentType": "scheme://contenttype",
         "name": "foo",
         "content": {
           ...
        }
    }
}
```

6.4.3 Endpoint type

6.4.3.1 *Design considerations*

The endpoint type is used to provide information about a resource's location on a network.

6.4.3.2 *Requirements*

Any entity value that represents a CADF Endpoint type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- If the "port" property is used, its value SHALL be consistent with the "address" property and its URI scheme (i.e., its domain-specific protocol scheme).

6.4.3.3 *Properties*

The following table describes the properties for the CADF Endpoint type.

Name	**Endpoint**		
Property	**Type**	**Required**	**Description**

address	xs:anyURI	Yes	The network address of the endpoint. For IP based addresses, this may be inclusive of port
port	xs:string	No	An optional property to provide the port value separate from the address property.

6.4.3.4 *Serialization examples*

XML example

```
<Event>
    ...
    <target
      id="myscheme://mydomain/network/node/9999"
      name="network-node-9999"
      address="http://mydomain/mypath/server-0001/">
      ...
    </target>
</Event>
```

JSON example

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "target": {
        "id": "myscheme://mydomain/resource/id/0001",
        "name": "server_0001",
        "ref": "http://mydomain/mypath/server-0001/",
        ...,
        "geolocation": {
            "city": "Austin",
            "state": "TX",
            "regionICANN": "US"
        }
    }
}
```

6.4.4 Geolocation type

6.4.4.1 *Design considerations*

Geolocation information, which reveals a resource’s physical location, is obtained using tracking technologies such as global positioning system (GPS) devices, or IP geolocation using databases that map IP addresses to geographic locations. Geolocation information is widely used in context-sensitive content delivery, enforcing location-based access restrictions on services, and fraud detection and prevention.

Due to the intense concerns about security and privacy, countries and regions introduced various legislation and regulation. To determine whether or not an event is compliant sometimes is dependent on the geolocation of the event. Therefore, it is crucial to report geolocation information unambiguously in an audit trail.

6.4.4.2 *Requirements*

Any entity value that represents a CADF Geolocation type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- Geolocation typed data SHALL contain at least one valid property and associated value.
- Geolocation typed data SHALL NOT be used to represent virtual or logical locations (e.g., network zone).
- For each geolocation data instance, the properties SHALL be consistent. That is, all properties SHALL consistently represent the same geographic location and SHALL NOT provide conflicting value data.
  - For example, when latitude, longitude and region are supplied as properties, the latitude and longitude coordinate values should resolve to the same geographic location as described by the region property's value.
- ICANN's implementation plan states "Upper and lower case characters are considered to be syntactically and semantically identical"; therefore, the "regionICANN" property's values MAY be either upper or lower case.

6.4.4.3 *Properties*

The following table defines the properties for the geolocation type. Geolocation must be agnostic to the methods and sources of information that are used to calculate positions.

One resource may contain zero or more geolocation instances.

Name	**Geolocation**		
Property	**Type**	**Required**	**Description**
id	xs:anyURI	No	Optional identifier for a geolocation.

latitude	xs:string	No	Indicate the latitude of a geolocation. Geolocation MAY be provided in a pair of latitude and longitude. Latitude values adhere to the format based on ISO 6709:2008 Annex H.2.1 – H.2.3. [ISO-6709-2008]
			Latitude on or north of the equator shall be designated using a plus sign (+), or no sign. Latitude south of the equator shall be designated using a minus sign (−).
			The first two digits of the latitude string shall represent degrees. Subsequent digits shall represent minutes, seconds or decimal fractions according to the following convention in which the decimal mark indicates the transition from the sexagesimal system to the decimal system:
			Degrees and decimal degrees:
			`DD.DD`
			Degrees, minutes and decimal minutes:
			`DDMM.MMM`
			Degrees, minutes, seconds and decimal seconds:
			`DDMMSS.SS`
			Leading zeros shall be inserted for a degree value less than 10, and zeros shall be embedded in proper positions when minutes or seconds are less than 10.
			For example, the latitude of Sunnyvale, California, United States is:
			`+37.37 or +372207.90`

longitude	xs:string	No	Indicate the longitude of a geolocation. Geolocation MAY be provided in a pair of latitude and longitude. Longitude values adhere to the format based on ISO 6709:2008 Annex H.3.1 – H.3.3. [ISO-6709-2008]
			Longitude on or east of the prime meridian shall be designated using a plus sign (+), or no sign. Longitude west of the prime meridian shall be designated using a minus sign (−)
			The first three digits of the longitude string shall represent degrees. Subsequent digits shall represent minutes, seconds or decimal fractions, according to the following convention in which the decimal mark indicates the transition from the sexagesimal system to the decimal system:
			Degrees and decimal degrees:
			`DDD.DD`
			Degrees, minutes and decimal minutes:
			`DDDMM.MMM`
			Degrees, minutes, seconds and decimal seconds:
			`DDDMMSS.SS`
			Leading zeros shall be inserted for degree values less than 100, and zeros shall be embedded in proper positions when minutes or seconds are less than 10.
			For example, the longitude of Sunnyvale, California, United States is:
			`122.04 or -1220210.20`

elevation	xs:double	No	Indicates the elevation of a geolocation in meters. Elevation at or above the sea level shall be designated using a plus sign (+), or no sign. Elevation below the sea level shall be designated using a minus sign (−).
accuracy	xs:double	No	Indicates the accuracy of a geolocation in meters. Geolocation expresses the resource location to a reasonable degree of accuracy.
city	xs:string	No	Indicate the city of a geolocation.
state	xs:string	No	Indicate the state/province of a geolocation
regionICANN	xs:string	No	Indicate a region (e.g., a country, a sovereign state, a dependent territory or a special area of geographical interest) of a geolocation. Region SHOULD match ICANN country code top level domain (ccTLD) naming convention [IANA-ccTLD] Geolocation MAY be able to resolve to region expressed as country code using the syntax provided by Domain Name System Security Extensions (DNSSEC) or using reverse geocoding services. Note: ICANN country codes (i.e. ccTLD values) MAY be expressed in upper or lower case, they are viewed as semantically equivalent.
annotations	cadf:map	No	Indicate user-defined geolocation information (e.g., building name, room number). The same “key” SHALL NOT be used more than once within a "annotation" property.

6.4.4.4 *Property Notes*

To avoid ambiguity, a geolocation could select one of the following two combinations as the essential properties, along with other supplementary properties.

- Latitude and longitude
- City, state, and region

6.4.4.5 *Serialization examples*

XML examples

The following describes several examples of the serialization of a geolocation in XML.

Geolocation: Sunnyvale, CA, United States

XML example 1: "latitude and longitude"

```
<geolocation
    latitude="+37.37"
    longitude="-122.04"
/>
```

XML example 2: "latitude, longitude, and elevation**"**

```
<geolocation
    latitude="+372207.90"
    longitude="-1220210.20"
    elevation="10"
/>
```

XML example 3: "latitude, longitude, and accuracy"

```
<geolocation
    latitude="N372207.90"
    longitude="W1220210.20"
    accuracy="100"
/>
```

XML example 4: "city, state and region"

```
<geolocation
    city="Sunnyvale"
    state="CA"
    regionICANN="US"
/>
```

XML example 5: "city, state, region, and user specific information"

```
<geolocation
    city="Sunnyvale"
    state="CA"
    regionICANN="us"
    <annotations>
        <item key=”building” value="B2"/>
        <item key=”room” value="201"/>
    </annotations>
</geolocation>
```

XML example 6: Geolocation referenced by a CADF Event

The following example shows a Geolocation definition being referenced from a TARGET resource within a CADF Event Record that is defined within the same CADF Log.

```
<Log>
    ...
    <geolocations>
      <geolocation
        geolocationId="muid://location.org/XYZ"
        unit="GB"
        name="Storage Capacity in Gigabytes"/>
      ...
    </geolocations>
    ...
    <events>
      <Event>
        ...
        <target
            id="myscheme://mydomain/resource/id/0001"
            typeURI="cadf://.../resource/..."
            name="server_0001"
            ref="http://mydomain/mypath/server_0001/"
            ...
            geolocationId="muid://location.org/XYZ"/>
        ...
      </Event>
    </events>
</Log>
```

JSON examples

JSON example 1: "latitude and longitude"

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "target": {
        ...,
        "geolocation": {
            "latitude": "+37.37",
            "longitude": "-122.04"
        }
    }
}
```

JSON example 2: "latitude, longitude, and elevation**"**

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "target": {
        ...,
        "geolocation": {
            "latitude": "+372207.90",
            "longitude": "-1220210.20",
            "elevation": "10"
        }
    }
}
```

JSON example 3: "latitude, longitude, and accuracy"

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "target": {
        ...,
        "geolocation": {
            "latitude": "N372207.90",
            "longitude": "W1220210.20",
            "accuracy": "100"
        }
    }
}
```

JSON example 4: "city, state and region"

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "target": {
        ...,
        "geolocation": {
            "city": "Sunnyvale",
            "state": "CA",
            "regionICANN": "US"
        }
    }
}
```

JSON example 5: "city, state, region, and user specific information"

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "target": {
        ...,
        "geolocation": {
            "city": "Sunnyvale",
            "state": "CA",
            "regionICANN": "us",
            "annotations": [
                {
                   "key": ”building”,
                   "value": "B2"
                },
                {
                   "key": ”room”,
                   "value": "201"
                }
            ]
        }
    }
}
```

JSON example 6: Geolocation referenced by a CADF Event

The following example shows a Geolocation definition being referenced from a TARGET resource within a CADF Event Record that is defined within the same CADF Log.

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/log",
    ...,
    "geolocations": [
        {
           "geolocationId": "muid://location.org/XYZ",
           "unit": "GB",
           "name": "Storage Capacity in Gigabytes"
        },
       ...
     ],
     ...
     "events":[
        {
           "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
           ...,
           "target": {
               "id": "myscheme://mydomain/resource/id/0001",
               "typeURI": "cadf://.../resource/...",
               "name": "server_0001",
               "ref": "http://mydomain/mypath/server_0001/",
               ...,
               "geolocationId": "muid://location.org/XYZ"
        }
     ]
}
```

6.4.5 Map

6.4.5.1 *Design considerations*

A list of key/value pairs with the additional constraints listed in the Requirements section below.

6.4.5.2 *Requirements*

Any entity value that represents an CADF Map type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- The same "key" property value SHALL NOT be used more than once within the same Map instance.
- The "key" property's value SHALL be treated as case-sensitive.

6.4.5.3 *Properties*

The following table describes the properties for the Map type defined by this specification:

Name	**Map**		
Property	**Type**	**Required**	**Description**
key	xs:string	Yes	The unique name that describes to the "value" property.
value	xs:string	Yes	Contains the data that corresponds to the "name" property.

6.4.5.4 *Serialization examples*

The serialization of a CADF Map complex type would appear as follows:

XML example

```
<Entity>
```

```
    ...
    <"map's property name">
        <item key="key 1" value="value 1">
        <item key="key 2" value="value 2">
        ...
    </"map's property name">
</Entity>
```

JSON example

```
{
    ...,
    "map's property name":
    [
        {
            "key": "key 1",
            "value": "value 1"
        },
        {
            "key": "key 2",
            "value": "value 2"
        }
    ]
}
```

6.4.6 Metric and Measurement types

This specification includes the consideration of auditable events generated to show operational compliance to measurable values. This section defines the following metric related types:

6.4.6.1 *Design considerations*

Cloud provider infrastructures are composed of resources that often need to share common metrics (e.g., storage sizes for volumes, processor speeds, etc.). These metrics are often tracked or monitored by other components perhaps to relate them to some external requirement or agreement (e.g., a Service License Agreement or SLA).

The Metric data type describes the rules and processes for measuring some activity or resource, resulting in the generation of some values (captured by the Measurement type). A set of metric instances may be associated with an Event Log, and referred to by individual events.

The Measurement type is intended to hold the values generated by the application of a metric in a particular context (e.g. for a resource or during an activity). The CADF Event Record includes a property that is capable of holding measurements represented by this type.

Additionally, it is often desirable to indicate the resource that actually provided or computed the value, as part of a measurement, if it is not provided by some other part of the event record.

6.4.6.2 *Requirements*

Any entity value that represents an CADF Metric or Measurement type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- Metric typed data SHALL provide "name" and "unit" properties with consistent values.
- Measurement typed data SHALL provide "metric" and "result" properties with consistent values.

- Measurement typed data SHALL contain either a valid "metric" property or a valid "metricId" property, but SHALL NOT contain both properties.

6.4.6.3 *Properties of Metric*

The following table describes the properties for the Metric type defined by this specification:

Name	**Metric**		
Property	**Type**	**Required**	**Description**
metricId	cadf:Identifier	Yes	The identifier for the metric. Metric data is designed so that it can be described once, for example in the context of a CADF Log, and referenced by the multiple CADF Event (records) the log contains..
unit	xs:string	Yes	The metrics unit (e.g. "msec.", "Hz", "GB", etc.)
name	xs:string	No	A descriptive name for metric (e.g. "Response Time in Milliseconds", "Storage Capacity in Gigabytes", etc.)
annotations	cadf:Map	No	Indicate user-defined metric information. The same “key” SHALL NOT be used more than once within a "annotation" property.

6.4.6.4 *Properties of Measurement*

The following table describes the properties for the Measurement type defined by this specification:

Name	**Measurement**		
Property	**Type**	**Required**	**Description**
result	xs:any	Yes	The quantitative or qualitative result of a measurement from applying the associated metric. The measure value could be boolean, integer, double, a scalar value (e.g. from an enumeration), or a more complex value.
metric	cadf:Metric	Dependent (see description)	The property describes the metric used in generating the measurement result. **Dependent Requirements** • This property SHALL be required if the "metricId" property is not used.
metricId	cadf:Identifier	Dependent (see description)	This property identifies a CADF Metric by reference and whose definition exists outside the event record itself (e.g., within the same CADF Log or Report). Note: This property can be used instead of the "metric" property to reference a valid Metric definition, which is already defined outside the Measurement itself, by its identifier (e.g., a CADF Metric already defined within a CADF Log which also contains the CADF Event with a CADF Measurement which is making the reference).

			Dependent Requirements
			• This property SHALL be required if the "metric" property is not used.
calculatedBy	cadf:Resource	No	An optional description of the resource that calculated the measurement (if it is not the same resource described by the INITIATOR already provided in the same CADF Event Record).

6.4.6.5 *Serialization examples*

XML examples

The following describes several examples of the serialization of CADF Measurements and Metrics in XML.

XML example 1: Using the "metric" property

The following XML format example shows how a CADF Measurement, within a CADF Event inside of a CADF Log, would reference a CADF Metric definition defined within the context of the same CADF Log using the metric's identifier.

```
<Event
  ...
  <measurements>
     <measurement result="10>
          <metric
            metricId="muid://metric.org/1234"
            unit="GB"
            name="Storage Capacity in Gigabytes"/>
     </measurement>
  </measurements>
</Event>
```

XML example 2: Using the "metricId" property

The following XML format example shows how a CADF Measurement, within a CADF Event inside of a CADF Log, would reference a CADF Metric definition defined within the context of the same CADF Log using the metric's identifier.

```
<Log>
     <metrics>
        <metric
          metricId="muid://metric.org/1234"
          unit="GB"
          name="Storage Capacity in Gigabytes"/>
        ...
     </metrics>
     ...
     <events>
        <Event
          ...
          <measurements>
               <measurement result="10
               metricId="muid://metric.org/1234"/>
          </measurements>
          ...
        </Event>
     </events>
```

```
</Log>
```

JSON examples

The following describes several examples of the serialization of CADF Measurements and Metrics in JSON.

JSON example 1: Using the "metric" property

The following JSON format example shows how a CADF Measurement, within a CADF Event inside of a CADF Log, would reference a CADF Metric definition defined within the context of the same CADF Log using the metric's identifier.

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "measurements": [
        {
          "metricId": "muid://metric.org/1234",
          "unit": "GB",
          "name": "Storage Capacity in Gigabytes"
        }
    ],
    ...
}
```

JSON example 2: Using the "metricId" property

The following JSON format example shows how a CADF Measurement, within a CADF Event inside of a CADF Log, would reference a CADF Metric definition defined within the context of the same CADF Log using the metric's identifier.

```
{
   "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/log",
   ...,
   "metrics": [
      {
         "metricId": "muid://metric.org/1234",
         "unit": "GB",
         "name": "Storage Capacity in Gigabytes"
      }
   ],
   ...,
   "events":[
      {
         "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
         ...,
         "measurements": [
            {
               "result": "10",
               "metricId": "muid://metric.org/1234"
            }
         ],
         ...
      }
   ]
}
```

6.4.7 Reason type

This data type is defined to describe the outcome of an Actual Event, along with related information, as part of the CADF Event Record.

6.4.7.1 *Design considerations*

There should be a consistent means to classify the top-level outcome of any action using the CADF Outcome Taxonomy along with any domain specific information, reasons or codes that enable further diagnostics within a specific provider's infrastructure.

6.4.7.2 *Requirements*

Any entity value that represents a CADF Reason type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- The "reasonType" and "reasonCode" properties' values SHALL be consistent with each other.
  - This means that the "reasonCode" value SHALL be a valid value as described by the domain specification identified by the "reasonType" value.
- The property "reasonType" SHALL NOT have an "empty", "blank" or zero-length value.
- The property "reasonCode" SHALL NOT have an "empty", "blank" or zero-length value.
- If the resource that calculated the measurement is different than the resource being recorded as the INITIATOR then the "calculatedBy" property SHOULD be provided.

6.4.7.3 *Properties*

The following table describes the properties for the Reason type defined by this specification:

Name	**Reason**		
Property	**Type**	**Required**	**Description**
reasonType	xs:anyURI	Yes	The domain URI which defines the "reasonCode" property's value. See examples below.
reasonCode	xs:string	Yes	An optional detailed result code as described by the domain identified in the "reasonType" property. Note: The “reasonCode” should in general indicate what type of policy was violated for its associated domain.

6.4.7.4 *Examples*

The "reasonCode" property is domain-specific and although CADF recommends the use of standard published “reasons” for events, it is recognized that many vendors have developed their own sets of event codes. The only constraint placed on such event code sets is that a reference can be constructed to them using the reasonType URI field.

One excellent canonical source for event reason codes is the HTTP Status Codes, which are defined by the URI (http://www.iana.org/assignments/http-status-codes/http-status-codes.xml). Although the HTTP Status Code definitions are somewhat specific to HTTP operations, in most cases they can be applied to many common INITIATOR-TARGET interactions equally well.

For example, any request to access a resource for which proper authorization has not been provided can result in a “401” reasonCode which corresponds to “Unauthorized.”

Similarly, The Open Group defines a series of codes in XDAS to represent various reasons for activity outcomes, defined by the URI (http://www.opengroup.org/bookstore/catalog/p441.htm). As an example, an attempt to use a resource that could not be completed due to hardware failure could be reported using reasonCode “0x00000401” which corresponds to “XDAS_OUT_HARDWARE_FAILURE.”

6.4.7.5 *Serialization Examples*

XML example

```
<Event>
    ...
    <reason
       reasonType="http://www.iana.org/assignments/http-status-codes/http-
          status-codes.xml"
       reasonCode="408"/>
    ...
</Event>
```

JSON example

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "reason": {
      "reasonType": "http://www.iana.org/assignments/http-status-
      codes/http-status-codes.xml",
      "reasonCode": "408"
    },
    ...
}
```

6.4.8 Reporterstep type

This type represents a step in the REPORTERCHAIN which captures information about a REPORTER and the action it performed on the CADF Event Record it is contained within.

6.4.8.1 *Design considerations*

The "Reporterstep" data type should capture information about systems (resources) that have a role in creating, modifying or relaying the CADF Event Record during its lifecycle.

The intent of "Reporterstep" data when included within a REPORTERCHAIN is to support forensic auditing of the sources of event data and the systems which subsequently handle that data for the purposes of verification, validation, and troubleshooting (i.e. these sources of event data are CADF REPORTERS).

Please note that any timestamp value that appears in the "reportTime" property, as filled in from any one REPORTER's perspective, might not be accurate with respect to any other REPORTER's "reportTime" value (e.g., perhaps due to local clock differences).

6.4.8.2 *Requirements*

Any entity value that represents a CADF Reporterstep type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- Each REPORTER that handles (i.e., creates, observes, modifies or relays) a CADF Event Record SHOULD add a Reporterstep entry to the REPORTERCHAIN, especially if the REPORTER modifies the CADF Event Record in any way.

- The REPORTER, when adding a Reporterstep entry to a CADF Event Record, SHOULD append it at the end (after) all other existing entries in the REPORTERCHAIN.
- ReportStep typed data SHALL contain either a valid "reporter" property or a valid "reporterId" property, but SHALL NOT contain both properties.

6.4.8.3 *Properties*

The following table describes the properties for the Reporterstep type defined by this specification:

Name	**Reporterstep**		
Property	**Type**	**Required**	**Description**
reporter	cadf:Resource	Dependent (see description)	This property defines the resource that acted as a REPORTER on a CADF Event Record. **Dependent Requirements** • This property SHALL be required when the "reporterId" property is not used.
reporterId	cadf:Identifier	Dependent (see description)	This property identifies a resource that acted as a REPORTER on a CADF Event Record by reference. and whose definition exists outside the event record itself (e.g., within the same CADF Log or Report). Note: This property can be used instead of the "reporter" property if the ReportStep is contained within a CADF Event that is in the same CADF Log or Report that also contains a valid CADF Resource definition for the resource being referenced as the REPORTER. **Dependent Requirements** • This property SHALL be required when the "reporter" property is not used.
role	xs:string	Yes	The role the REPORTER performed on the CADF Event Record (e.g., an "observer", "modifier" or "relay" role). The valid set of values is defined in the section "Reporter Roles".
reporterTime	cadf:Timestamp	Yes	The time a REPORTER adds its Reporterstep entry into the REPORTERCHAIN (which follows completion of any updates to or handling of the corresponding CADF Event Record).
attachments	cadf:Attachment[]	No	An optional array of additional data containing information about the reporter or any action it performed that affected the CADF Event Record contents.

6.4.8.4 *Serialization examples*

XML example

```
<Event
    ...
    <reportchain>
       <reporterstep
           role="observer"
           reporterTime="2012-03-22T13:00:00-04:00">
           <reporter id="myscheme://mydomain/resource/monitor/id/0002"/>
           ...
       </reporterstep>
    </reportchain>
</Event>
```

JSON example

```
"Event": {
    ...,
    "reporterchain": [
        {
           "role": "observer",
           "reporterTime": "2012-03-22T13:00:00-04:00",
           "reporter": {
             "id": "myscheme://mydomain/resource/monitor/id/0002"
           }
         },
         ...
     ]
}
```

6.4.9 Resource type

This data type is provided as the means to describe any resource that participated in an Actual Event (e.g., INITIATOR, TARGET or REPORTER) as part of a CADF Event Record.

6.4.9.1 *Design considerations*

There should be a consistent means to identify, classify and track resources and their usage within a provider's infrastructure; it is fundamental consideration for auditing. Therefore, we introduce a CADF base resource data type which will enable these goals, but also permit extended resource descriptions for specific profiles of this specification.

6.4.9.2 *Requirements*

Any entity value that represents an CADF Resource type in this specification, its extensions or profiles SHALL adhere to the following requirements.

- Any profile or extension of this specification that defines additional resource types that derive from CADF Resource type and can be included in or referenced by a CADF Event Record SHALL extend the CADF Resource Type.
  - This means that extensions or profiles of this specification that derive resource types from the CADF resource type SHALL provide valid "typeURI" values for these derived types that extend from the URI values specified by the CADF Resource Taxonomy.

- Any profile or extension of this specification that extends any CADF defined Resource type, including any derived types, SHALL NOT override or change any properties already defined by this specification.
- All CADF Resource typed data, including all derived types, SHALL be classified using the CADF Resource Taxonomy or extensions of it using the "typeURI" property.
    - Relative path representation of CADF Resource Taxonomy values SHOULD be used in the "typeURI" property of CADF Resource typed data when possible.
- Any CADF Resource typed data that includes CADF Geolocation data SHALL have either valid "geolocation" property or a valid "geolocationId" property, but SHALL NOT contain both properties.

6.4.9.3 *Properties*

The following table describes the properties for the Resource Type defined by this specification:

Name	**Resource**		
Property	**Type**	**Required**	**Description**
id	cadf:Identifier	Yes	The identifier for the resource.
typeURI	cadf:Path	Yes	The classification (i.e., type) of the resource using the CADF Resource Taxonomy.
name	xs:string	No	The optional local name for the resource (not necessarily unique).
ref	xs:anyURI	No	An optional navigatable reference to the resource. Note: This is not necessarily a publicly accessible reference; but may be navigatable in a private or secured context.
domain	xs:string	No	The optional name of the domain that qualifies the name of the resource (e.g., a path name, a container name, etc.).
geolocation	cadf:Geolocation	Dependent (see description)	This optional property describes the geographic location of the resource using a CADF Geolocation data type. **Dependent Requirements** • This property SHALL be required if the "geolocationId" property is not used.
geolocationId	cadf:Identifier	Dependent (see description)	This optional property identifies a CADF Geolocation by reference and whose definition exists outside the event record itself (e.g., within the same CADF Log or Report level). Note: This property can be used instead of the "geolocation" property to reference a valid CADF Geolocation definition, which is already defined outside the resource itself, by its identifier (e.g. a CADF Geolocation already defined at the CADF Log or Report level which also contains the CADF Resource definition). **Dependent Requirements** • This property SHALL be required if the "geolocation" property is not used.
attachments	cadf:Attachment[]	No	An optional array of extended or domain-specific information about the resource or its context.

6.4.9.4 *Serialization Examples*

XML example

```
<Event>
    ...
    <target
      id="myscheme://mydomain/resource/id/0001"
      name="server_0001"
      ref="http://mydomain/mypath/server-0001/">
      ...
      <geolocation city="Austin" state="TX" regionICANN="US"/>
    </target>
</Event>
```

JSON example

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
    ...,
    "target": {
        "id": "myscheme://mydomain/resource/id/0001",
        "name": "server_0001",
        "ref": "http://mydomain/mypath/server-0001/",
        ...,
        "geolocation": {
            "city": "Austin",
            "state": "TX",
            "regionICANN": "US"
        }
    }
}
```

6.5 CADF Entities

This section defines CADF Entities, as inspired from Entity-Relationship (ER) modeling, that represent complex CADF data types that also represent significant resources that can be referenced, modeled and have relationships that can be referenced through unique identifiers.

Note: As a corollary, this specification makes the distinction that CADF complex data types should only be referenced within the scope of CADF Entities and other CADF complex data types.

6.5.1 Event type

This entity represents the CADF Event Record.

6.5.1.1 *Design considerations*

The design of the event schema is intended to address the following requirements:

- The event schema should be able to represent any auditable event. This includes consideration of events that support compliance reporting and monitoring of:
  - operational and business processes, applications and services running in cloud deployments.
  - cloud services and software usage including monitoring of Service License Agreements (SLAs) and Software License Management (SLM) in the cloud.
- The event schema should be able to preserve other or domain specific event record formats.

- The event schema should support cross-event correlation.

6.5.1.2 *Entity Type URI*

The following entity type URI value is used to identify the CADF Event data type:

Entity	Entity Type URI
Event	http://schemas.dmtf.org/cloud/audit/1.0/event

6.5.1.3 *Requirements*

Any value that represents a CADF Event type in this specification, its extensions or profiles SHALL adhere to the following requirements:

- CADF Event Records SHALL contain either a valid "initiator" property or a valid "initiatorId" property, but SHALL NOT contain both properties.
- CADF Event Records SHALL contain either a valid "target" property or a valid "targetId" property, but SHALL NOT contain both properties.
- **Action property requirements:**
  - The "action" property SHALL include a valid value from the CADF Action Taxonomy or an extension thereof.
  - The "action" property's value SHALL NOT be an empty string.
  - The "action" property's value SHOULD represent the perspective of the OBSERVER (see the Basic Model Components section).
- **Outcome property requirements**:
  - The "outcome" property SHALL include a valid value from the CADF Outcome Taxonomy or an extension thereof.
  - The "outcome" property's value SHALL NOT be an empty string.
  - The "outcome" property's value SHOULD represent the perspective of the OBSERVER (see the Basic Model Components section).
- **Initiator property requirements:**
  - The "initiator" property SHALL include a valid resource classification value from the CADF Resource Taxonomy or an extension thereof.
  - The "initiator" property's value SHALL NOT be an empty string.
  - The "initiator" property's value SHOULD represent the perspective of the OBSERVER (see the Basic Model Components section).
- **Target property requirements:**
  - The "target" property SHALL include a valid resource classification value from the CADF Resource Taxonomy or an extension thereof.
  - The "initiator" property's value SHALL NOT be an empty string.
  - The "initiator" property's value SHOULD represent the perspective of the OBSERVER (see the Basic Model Components section).

6.5.1.4 *Best practices*

- Note: A array of CADF Event Records may appear as part of a CADF Log or CADF Report. These CADF Entities provide the facility to fully describe resources, metrics and other attachments (once) as part of array properties so that CADF Event Records may reference these log-level definitions without having to describe them repeatedly in each event where they may appear.

- CADF Event Records that appear within a CADF Log or CADF Report SHOULD reference log-level resource, metric, geolocation and attachment definitions when possible (e.g., for properties such as "initiator", "target" or "reporter" as part of the "reporterchain"). For example, a CADF Event Record inside of a CADF Log could have a TARGET resource that is referenced using the "targetId" property and whose full definition is listed in the "resources" array property of the CADF Log type.

6.5.1.5 *Properties*

The following table describes the properties for the Event Type defined by this specification:

Name	**Event**		
Property	**Type**	**Required**	**Description**
typeURI	cadf:Path	Dependent (See description)	This property has the dependent requirements that are described in the "Entity Type URIs" section of this specification. Additional requirements are listed below. **Dependent Requirements** • If the "typeURI" property is included on this entity then the value SHALL be the Entity Type URI specified for the CADF Event type. **Format Dependent Requirements** • If XML format is used, the "typeURI" property MAY be used. • JSON format is used: the "typeURI" property SHALL be used.
id	cadf:Identifier	Yes	The unique identifier of the CADF Event Record.
eventType	xs:string	Yes	The CADF Event Type. See the section titled "CADF Event Type values" for valid values.
eventTime	cadf:Timestamp	Yes	The OBSERVER's best estimate as to the time the Actual Event occurred or began (note that this may differ significantly from the time at which the OBSERVER is processing the Event Record).
action	cadf:Path	Yes	This property represents the event's ACTION. See Basic Model Components for details. Please see the CADF Action Taxonomy for valid values and requirements.
outcome	cadf:Path	Yes	A valid classification value from the CADF Outcome Taxonomy.
initiator	cadf:Resource	Dependent (see description)	This property represents the event's INITIATOR. See Basic Model Components for details.. **Dependent Requirements**

			• This property SHALL be required if the "initiatorId" property is not used.
initiatorId	cadf:Identifier	Dependent (see description)	This property identifies the event's INITIATOR resource by reference. Note: This property can be used instead of the "initiator" property if the CADF Event data is contained within the same CADF Log or Report that also contains a valid CADF Resource definition for the resource being referenced as the INITIATOR. **Dependent Requirements** • This property SHALL be required if the "initiator" property is not used. • If this property is used, its value SHALL reference a valid CADF Resource definition (e.g., at CADF Log level).
target	cadf:Resource	Dependent (see description)	This property represents the TARGET. See Basic Model Components for details. **Dependent Requirements** • This property SHALL be required if the "targetId" property is not used.
targetId	cadf:Identifier	Dependent (see description)	This property identifies the event's TARGET by reference. Note: This property can be used instead of the "target" property if the CADF Event data is contained within the same CADF Log or Report that also contains a valid resource definition for the resource being referenced as the TARGET. **Dependent Requirements** • This property SHALL be required if the "target" property is not used. • If this property is used, its value SHALL reference a valid CADF Resource definition (e.g., at CADF Log level).
reason	cadf:Reason	No	This property contains an optional, domain-specific reason code and related information which provides an additional level of detail to the outcome value.

severity	xs:string	No	This property describes domain-relative severity assigned to the event by the OBSERVER. This property's value is non-normative, but is the recommended place where such information should be placed. **Note**: This property's value may only have meaning within the usually limited domain understood by the OBSERVER and does not represent any form of enterprise risk. This property's value may be used by event consumers that understand the OBSERVER's domain and need to prioritize events it reported. Note: Profiles of this specification may define specific severity values that could be used in this property.
measurements	cadf:Measurement[]	Dependent (see description)	This property represents any measurement (values) associated with the event, resulting from the application of some metrics. **Dependent Requirements** • This property SHALL be present if the "eventType" property's value is "**monitor**". • This property MAY be present if the "eventType" property's value is "**activity**".
attachments	cadf:Attachment[]	No	An optional array of extended or domain-specific information about the event or its context.
reporterchain	cadf:Reporterstep[]	Yes	An array of Reporterstep typed data that contains information about the sequenced handling of or change to the associated CADF Event Record by any REPORTER. See discussion of the Reporter Chain component of the CADF Event Model.

6.5.1.6 *Serialization examples*

XML examples

The following example shows the CADF Event Record using the dependent properties "initiator" and "target" which fully describes these resources within the record itself.

```
<Event
    id="myscheme://mydomain/event/id/1234"
    eventType="activity"
    eventTime="2012-03-22T13:00:00-04:00"
    action="create"
    outcome="success">
    <initiator id="..." typeURI="..."/>
    <target id="..." typeURI="..."/>
    ...
    <reporterchain>
        <reporterstep
            role="observer"
            reporterTime="2012-08-22T23:00:00-02:00">
            <reporter id="..."/>
        </reporterstep>
        ...
    </reporterchain>
</Event>
```

The following example shows the CADF Event Record using the dependent properties "initiatorId" and "targetId" (instead of the "initiator" and "target" properties) which reference CADF resources which are fully defined within the same CADF Log that also contains the CADF Event record itself.

```
<Log>
    ...
    <resources>
        <resource id="muid://location.org/resource/0001" typeURI="..."/>
        <resource id="muid://location.org/resource/0099" typeURI="..."/>
        <resource id="muid://location.org/resource/0321" typeURI="..."/>
        ...
    </resources>
    <events>
        <Event id="myscheme://mydomain/event/id/1234"
            eventType="activity"
            eventTime="2012-03-22T13:00:00-04:00"
            action="create"
            outcome="success"
            initiatorId="muid://location.org/resource/0001"
            targetId="muid://location.org/target/0099">
            ...
            <reporterchain>
                <reporterstep
                  role="observer"
                  reporterTime="2012-08-22T23:00:00-02:00">
                  <reporter id="muid://location.org/resource/0321"/>
                </reporterstep>
                ...
            </reporterchain>
        </Event>
        ...
     </events>
</Log>
```

JSON examples

The following example shows the CADF Event Record using the dependent properties "initiator" and "target" which fully describes these resources within the record itself.

```
{
     "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
     "id": "myscheme://mydomain/event/id/1234",
     "eventType": "activity",
     "eventTime": "2012-03-22T13:00:00-04:00",
     "action": "create",
     "outcome": "success",
     "initiator": {
           "id": "...",
           "typeURI": "..."
     },
     "target": {
           "id": "...",
           "typeURI": "..."
     },
     ...,
     "reporterchain": [
         {
             "role": "observer",
             "reporterTime": "2012-08-22T23:00:00-02:00",
             "reporter": {
                 "id": "..."
             }
         },
         ...
     ]
}
```

The following example shows the CADF Event Record using the dependent properties "initiatorId" and "targetId" (instead of the "initiator" and "target" properties) which reference CADF resources which are fully defined within the same CADF Log that also contains the CADF Event record itself.

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/log",
    ...,
    "resources": [
        {
           "id": "muid://location.org/resource/0001",
           "typeURI": "...",
           ...
        },
        {
           "id": "muid://location.org/resource/0099",
           "typeURI": "...",
           ...
        },
        {
           "id": "muid://location.org/resource/0321",
           "typeURI": "...",
           ...
        },
        ...
    ],
    "events": [
        {
            "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
            "id": "myscheme://mydomain/event/id/1234",
            "eventType": "activity",
            "eventTime": "2012-03-22T13:00:00-04:00",
            "action": "create",
            "outcome": "success",
            "initiatorId": "muid://location.org/resource/0001",
            "targetId": "muid://location.org/target/0099",
            ...,
            "reporterchain": [
                {
                   "role": "observer",
                   "reporterTime": "2012-08-22T23:00:00-02:00",
                   "reporter": {
                     "id": "muid://location.org/target/0321"
                   }
                }
             ]
         },
         ...
     ]
}
```

6.5.2 Log type

The log schema is intended to contain one or more event elements that are compiled together by a system component for storage and/or submission to another application for the purposes of compilation, backup and event analysis. The report format is suitable for federation and composition with other logs of the same schema.

The interaction model described in this specification provides interfaces and filters for the query of auditable event data whose result set defined by the report schema.

6.5.2.1 *Design considerations*

The design of the log schema is intended to address the following Design considerations:

- The log should contain a unique identifiable reference and information about the resource (e.g., an application or service) that compiled the event data within the log.
- The log should be able to provide declarations that provide short-form values that can used to replace repeated, long-form entity and property values (such as namespaces and identifiers) that permit condensed reports for transmission / federation.
- The log may be assigned a time period that defines time boundaries (a begin date/time, and end date/time) for all events of interest for this log. In other words, all events of interest over this time period are supposed to be present in the log.
- The log should permit the ability to contain signed and/or encrypted event or informational data.

6.5.2.2 *Entity Type URI*

The following entity type URI value is used to identify the CADF Log data type:

Entity	Entity Type URI
Log	http://schemas.dmtf.org/cloud/audit/1.0/log

6.5.2.3 *Requirements*

Any value that represents a CADF Log type in this specification, its extensions or profiles SHALL adhere to the following requirements:

- CADF Event Records that appear in a CADF Log SHOULD only have "eventTime" property values (timestamps) that are equal to or greater than the "beginTime" property value.
- CADF Event Records that appear in a CADF Log SHOULD only have "eventTime" property values (timestamps) that are equal to or less than the "endTime" property value.
- All recurring instances of a same complex type or entity within a CADF Report (e.g. CADF Resource, CADF Event, CADF Metric, etc.) SHALL have a unique identifier (cadf:Identifier) within the report.

6.5.2.4 *Properties*

The following properties are supported by the CADF Log type:

Name	**Log**		
Property	**Type**	**Required**	**Description**
typeURI	cadf:Path	Dependent (See description)	This property has the dependent requirements that are described in the "Entity Type URIs" section of this specification. Additional requirements are listed below. **Dependent Requirements** - If the "typeURI" property is included on this entity then the value SHALL be the Entity Type URI specified for the CADF Log type. **Format Dependent Requirements** - If XML format is used, the "typeURI" property MAY be used. - JSON format is used: the "typeURI" property SHALL be used.
id	cadf:Identifier	No	The identifier for this CADF Log (instance).
generatorId	cadf:Identifier	Yes	The identifier of the actual resource that generated the log.
logTime	cadf:Timestamp	Yes	The time the log was last updated. This time may be used to represent the time the log creation is complete and ready for subsequent consumption (e.g., federation, processing or archival). See discussion of "future considerations" for more information on this topic.
beginTime	cadf:Timestamp	No	The beginning time for the time period of event records within the log. Event records that appear in the log should only have event times (timestamps) that are equal to or greater than this time.
endTime	cadf:Timestamp	No	The end time for the time period of event records within the log. Event records that appear in the log should only have event times (timestamps) that are equal to or less than this time.
description	xs:string	No	An optional description of the log or its contents.
resources	cadf:Resource[]	No	An optional array of CADF Resources that may be referenced by multiple CADF Event Records within the log (i.e. the events would refer to a resource by its ID).
geolocations	cadf:Geolocation[]	No	An optional array of CADF Geolocations that may be referenced by multiple CADF resources that appear within CADF Event Records within the log (i.e. the resources refer to a geolocation by its ID, as part of a resource typed property, such as a TARGET or INITIAITOR).
metrics	cadf:Metric[]	No	An optional array of CADF Metrics that may be referenced by

			multiple CADF Events Records within the log (i.e. the events would refer to a metric by its ID, as part of its measurement property).
events	cadf:Event[]	Yes	An array of CADF Event (records) that are the primary compositional entity of the CADF Log. Note: In the case that the log was created, but no events occurred during the log period, the events property should be present but the array should contain no elements (i.e. be an "empty" array of events).
attachments	cadf:Attachment[]	No	An optional array of extended or domain-specific information about the log or its context.

6.5.2.5 *Serialization examples*

XML example

```
<Log
    id="myscheme://mydomain/log/id/log_1234"
    logTime="2012-03-22T13:00:00-04:00"
    ...
    <events>
        <Event id="myscheme://mydomain/event/id/AAA">
          ...
        </Event>
        <Event id="myscheme://mydomain/event/id/BBB">
          ...
        </Event>
        ...
    </events>
</Log>
```

JSON example

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/log",
    "id": "myscheme://mydomain/log/id/log_1234",
    "logTime": "2012-03-22T13:00:00-04:00",
    ...,
    "events": [
       {
         "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
         "id": "myscheme://mydomain/event/id/AAA",
         ...
       },
       {
         "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
         "id": "myscheme://mydomain/event/id/BBB",
         ...
       },
       ...
    ]
}
```

6.5.3 Report type

The report is intended to contain one or more event records that are compiled together in response to request for auditable data that fulfills a discrete query. The report format is suitable for federation and composition with other reports of the same schema.

The interaction model described in this specification provides interfaces and filters for the query of auditable event data whose result set is defined by the report schema.

6.5.3.1 *Differences between reports and logs*

CADF acknowledges that, especially in auditing domains, reports and logs are distinct named entities with different functional purposes. In this draft, the CADF Logs and Report data types may look very similar however, future draft revisions will evolve these types to be significantly different.

Fundamentally, logs are intended to a more compact, simpler container for federating events with some basic information about log identity and construction. Reports are intended to be more robust containers that contain information such as attestations of contents (e.g. events, etc.), linkage to compliance frameworks and controls and query data used to generate the report data.

Please note that we expect profiles of this specification to convey their specific "Report" information via extensions of these data types (and remain compatible with CADF interfaces) by extending these types. For example, an SSAE16 report could be attached to a this CADF entity and signed along with other information and provided to a cloud consumer.

6.5.3.2 *Design considerations*

The design of the report schema is intended to address the following Design considerations:

- The report may contain a reference to or the actual query used to generate the report.
- The report may provide declarations that permit aliasing of URIs and Paths that may be repeated referenced by entities contained within the report.

6.5.3.3 *Use cases*

The following are exemplary use cases for reports in the context of this specification:

- Report "privileged access" events that reflect actions against a resource performed by users who have a privileged role such as an administrator, manager or security officer.
- Report all events related to a specific cloud application or service that occurred between a specific date-time interval.
- Report all events that have been classified as being applicable to a specified security compliance standard.

6.5.3.4 *Entity Type URI*

The following entity type URI value is used to identify the CADF Report data type:

Entity	Entity Type URI
Report	http://schemas.dmtf.org/cloud/audit/1.0/report

6.5.3.5 *Requirements*

Any value that represents a CADF Report type in this specification, its extensions or profiles SHALL adhere to the following requirements:

- CADF Event Records that appear in a CADF Report SHOULD only have "eventTime" property values (timestamps) that are equal to or greater than the "beginTime" property value.
- CADF Event Records that appear in a CADF Report SHOULD only have "eventTime" property values (timestamps) that are equal to or less than the "endTime" property value.
- All recurring instances of a same complex type or entity within a CADF Report (e.g. CADF Resource, CADF Event, CADF Metric, etc.) SHALL have a unique identifier (cadf:Identifier) within the report.

6.5.3.6 *Properties*

The following properties are supported by the CADF Report Data Type

Name	**Report**		
Property	**Type**	**Required**	**Description**
typeURI	cadf:Path	Dependent (See description)	This property has the dependent requirements that are described in the "Entity Type URIs" section of this specification. Additional requirements are listed below. **Dependent Requirements** • If the "typeURI" property is included on this entity then the value SHALL be the Entity Type URI specified for the CADF Report type. **Format Dependent Requirements** • If XML format is used, the "typeURI" property MAY be used. • JSON format is used: the "typeURI" property SHALL be used.
id	cadf:Identifier	No	The identifier for this CADF Report (instance).
reportTime	cadf:Timestamp	Yes	The time the report was last updated. This time may be used to represent the time the report creation is complete and ready for subsequent consumption (e.g., federation, processing or archival). See discussion of "future considerations" for more information on this topic.
beginTime	cadf:Timestamp	No	The beginning time for the time period of event records within the report. Event records that appear in the report should only have event times (timestamps) that are equal to or greater than this time.
endTime	cadf:Timestamp	No	The end time for the time period of event records within the report. Event records that appear in the report should only have event times (timestamps) that are equal to or less than this time.
description	xs:string	No	An optional description of the report or its contents.
resources	cadf:Resource[]	No	An optional array of CADF Resources that may be referenced by multiple CADF Event Records within the report (i.e. the events would refer to a resource by its ID).

geolocations	cadf:Geolocation[]	No	An optional array of CADF Geolocations that may be referenced by multiple CADF resources that appear within CADF Event Records within the report (i.e. the resources refer to a geolocation by its ID, as part of a resource typed property, such as a TARGET or INITIAITOR).
metrics	cadf:Metric[]	No	An optional array of CADF Metrics that may be referenced by multiple CADF Events Records within the report (i.e. the events would refer to a metric by its ID, as part of its measurement property).
logIds	cadf:Identifier[]	Dependent	The references to the CADF Log(s) that contains the CADF Event Records that are the primary compositional entity of the CADF Report.
logs	cadf:Log[]	Dependent	The CADF Log(s) that contains the CADF Event Records that are the primary compositional entity of the CADF Report.
attachments	cadf:Attachment[]	No	An optional array of extended or domain-specific report information or additional context information.

6.5.3.7 *Serialization examples*

XML example

```
<Report
    id="myscheme://mydomain/report/id/report_889"
    reportTime="2012-08-31T18:00:00-02:00"
    ...
    <logs>
        <Log id="myscheme://mydomain/log/id/XXX">
           ...
        </Log>
    </logs>
</Report>
```

JSON example

```
{
    "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/report",
    "id": "myscheme://mydomain/report/id/report_889",
    "reportTime": "2012-08-31T18:00:00-02:00",
    ...,
    "logs": [
       {
          "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/log",
          "id": "myscheme://mydomain/log/id/XXX",
          ...
       },
    ]
}
```

7 CADF Resource type derivations

The following complex types are derived from the CADF Resource complex data type. This means that these types essentially extend the base CADF Resource type by defining additional "Extended Properties" that can be required for inclusion in the base CADF Resource type.

7.1 Extended property requirements for resource types

Any CADF Resource types that is included in a CADF Event Record (e.g., INITIATOR, TARGET, REPORTER, etc.) and is classified by the CADF Resource Taxonomy as one of the derived types listed below (i.e., by its "typeURI" property):

- CADF Resource typed data SHALL include the (extended) "properties" listed for the derived type they are classified by based upon the value provided in the "typeURI" property of the CADF Resource type as specified below.
- Any (extended) "properties" that are included in a derived CADF Resource type SHALL have valid values.

7.2 Notes

The CADF acknowledges that additional derived resource types with "extended properties" may be identified for inclusion in future drafts of this specification. This draft includes an initial set of CADF defined derived resource types which address audit use cases the working group has had time to address at the time of this draft's authoring.

7.3 Extended properties for derived CADF Resource types

This section lists the derived types of the CADF Resource data type, as classified by CADF Resource Taxonomy URI values, along with the "extended properties" the CADF has identified as necessary for normative audit purposes.

7.3.1 Account

Any CADF Resource data type that is classified by the CADF Resource Taxonomy as an "account" SHALL have the following additional properties:

Derivation Name	**Account**		
typeURI	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/data/security/account		
Property	**Type**	**Required**	**Description**
effectiveAccountId	cadf:Identifier	No	The identifier for the effective account whose credentials were actually used to evaluate access to a resource (e.g., superuser or administrator account using a "sudo" command).
effectiveAccountName	xs:string	No	The optional name of the effective account whose credentials were actually used to evaluate access to a resource (e.g. superuser or administrator).
accountCredentials	cadf:Credential	Yes	Identifies/describes the source and its authorizations for performing the event action.

7.3.2 Connection

Any CADF Resource data type that is classified by the CADF Resource Taxonomy as an "connection" SHALL have the following additional properties:

Derivation Name	**Connection**		
typeURI	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/network/connection		
Property	**Type**	**Required**	**Description**
protocol	xs:string	Yes	The protocol schema used to interpret the address. For example: http, ftp, etc.
source	cadf:Endpoint	Yes	The endpoint for that describes the starting point for a network data stream.
destination	cadf:Endpoint	Yes	The endpoint for that describes the ending point for a network data stream.

7.3.3 Credential

This type, which derives from the CADF Resource type, provides a means to describe various credentials along with any information about the authority that is responsible for maintaining them.

Any CADF Resource data type that is classified by the CADF Resource Taxonomy as a "credential" SHALL have the following additional properties:

Derivation Name	**Credential**		
typeURI	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/data/security/credential		
Property	**Type**	**Required**	**Description**
type	xs:anyURI	No	Type of credential. TBD (e.g., auth. token, identity token, etc.)
authority	xs:anyURI	No	Identifies the trusted authority (a service) that understands and can verify the credential.
assertions	cadf:Map	Yes	Optional list of opaque or non-opaque assertions or attributes that belong to the credential.

7.3.3.1 *Notes*

This resource type is intended to describe various credentials that are used to evaluate access control decisions when accessing resources. This data type is intended to allow representation of any credentials at any granularity by allowing any assertion to be included in the "assertions" property. Examples of credential data that may be represented by this data type include:

- Simple userid-password credentials or basic authentication information.
- Various opaque and non-opaque token formats and profile information (e.g. OAuth (1.0, 2.0), SAML 2.0, JSON Web Token (JWT), etc.).
- Certificates and other “trust” indication information.
- Other types by enabling assertion based description of other credential formats.

7.3.4 Endpoint

Support top-level field that can represent a physical or logical address or location on a network. These extended properties encourage the inclusion of a network address, such as an IP address and perhaps a port number (if applicable). The base CADF Resource type's existing properties can be used to hold other descriptive endpoint information, such as a Host Name or DNS Name.

Any CADF Resource data type that is classified by the CADF Resource Taxonomy as an "endpoint" SHALL have the following additional properties:

Derivation Name	**Endpoint**		
typeURI	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/network/endpoint		
Property	**Type**	**Required**	**Description**
address	xs:anyURI	Yes	The network address of the endpoint.
port	xs:string	No	For IP based addresses, this would be inclusive of port.

7.3.5 Node (Network, Compute, Storage)

Any CADF Resource data type that is classified by the CADF Resource Taxonomy as a "node" SHALL have the following additional properties:

Derivation Name	**Node**		
typeURI	**Network**	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/network/node	
	Compute	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/compute/node	
	Storage	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/storage/node	
Property	**Type**	**Required**	**Description**
endpoint	cadf:Endpoint	No	The endpoint used to access (or perform operations on) the node if it addressable on a network. If the node is disconnected from the network or has not been allocated an address, this property MAY be omitted.

7.3.6 Service

Any CADF Resource data type that is classified by the CADF Resource Taxonomy as a "service" SHALL have the following additional properties:

Derivation Name	**Service**		
typeURI	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/service		
Property	**Type**	**Required**	**Description**
endpoint	cadf:Endpoint	Yes	The service endpoint used to access (or perform operations on)

			the service.
role	xs:string	No	The role (e.g. operational, business, security, etc.) the service fulfills in the provider infrastructure.
credentials	cadf:Credential	No	Describes any authorizations the service may have.

7.3.7 User

Any CADF Resource data type that is classified by the CADF Resource Taxonomy as a "user" SHALL have the following additional properties:

Derivation Name	**User**		
typeURI	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/data/security/account**/**user		
Property	**Type**	**Required**	**Description**
attributes	cadf:Map	No	User (identity) attributes.

8 CADF Interfaces

This draft version of the CADF specification will not define CADF interfaces; these will be developed in subsequent public drafts.

9 CADF Entity signing

This draft version of the CADF specification will not address entity signing, specifically the signing of the CADF Event Record, Event Log and Event Report. This topic will be developed in subsequent public drafts.

10 CADF Profiles

This draft version of the CADF specification will not address profiling of the specification in detail. This topic will be developed in subsequent public drafts. However, the CADF WG has already identified several requirements for profiles of this specification that are listed below.

10.1 Requirements

The following requirements SHALL be followed when creating profiles of this specification:

- Profiles SHOULD seek to extend the data schema from this specification whenever possible.
- Profiles SHALL follow all guidelines and requirements when extending CADF Entities, Data types and their properties as defined or listed in this specification.
- Profiles MAY define additional namespaces or domain identifiers.
    - Profiles that define additional domain identifiers or namespaces SHALL follow the requirements described in this specification.
- Profiles MAY define additional entities data types and properties when extension of existing CADF Entities, data types and properties is not possible.
    - Profiles that define additional data schema elements SHALL ensure they adhere to and are compatible with the approved Extensibility Mechanisms described in this specification.
- **Format Profiles** MAY be developed to describe data representation and exchange formats other than XML or JSON. *Note, that this approach may be desirable to reduce the size of audit data within deployments when not being federated.*
    - However, the XML format SHALL be considered as the normative exchange format for federation between cloud providers.
    - Non-XML format profiles SHALL provide deterministic translations and lossless (data) to/from the core XML data schema described by this specification.
- XML-based format profiles that extend this specification's XML data schema SHALL be validatable against this specification's XML data schema definition.

11 Future Considerations

The CADF will potentially consider the following items in future version drafts of this specification's event, data and interface models:

- Support for mapping to multiple domain specific compliance frameworks.
  - The WG has already discussed potential support for domain specific identifiers and tags that enable domain specific identification that may be supported by query interfaces.
  - Such mechanisms would help link CADF Event Records to well-defined security compliance standards and frameworks such as ISO 27001, PCI DSS, SSAE16 (formerly SAS70), ISACA COBIT, etc.
- Support for summarization of sets of like events into a single CADF Event Record.
- Support for aggregation of sets of like events into a single CADF Event Record.
- Support for correlating related events using the CADF Event Record.
- Support for secure signing of CADF Events, Logs and Report entities.
- Support for multiple TARGETS on CADF Event Records.
- Support for declaring relationships between CADF Resource Types on the same event
  - This consideration would also permit attaching additional CADF Resource Types (data) to the CADF Event Record that represent cloud resources that have significant relationships to the CADF Event Record's INITIATOR, TARGET or REPORTER.
- For this draft, the concept of a CADF Log or CADF Report are entities that are perhaps created as a response to a consumer query against the provider at a point in time. This concept views audit logs and reports as "immutable"; in future drafts, we will address use cases that perhaps specify how to have "mutable" CADF Logs and Reports.

A. CADF Event Model component classification

This CADF Event Record is designed to support a means to classify the primary components the CADF Event Model using the extensible taxonomies defined in this appendix.

These values are intended to be used by the query interfaces defined in this specification to construct meaningful views for CADF Event Record consumers from the complete set of provider audit data available in the form of logs and reports.

This section describes the action taxonomy that is used to classify the type of activity that is described in an event record.

A.0 CADF Resource Taxonomy

This section describes the CADF logical resource taxonomy used as a basis to classify types of resources that may be significant when auditing cloud provider infrastructures. These represent values that are to be used in the "typeURI" property for the CADF Resource data type.

A.0.1 Model description

This taxonomy is intended to provide a logical naming model for resources that will be encountered when auditing cloud deployments. It is not intended to be an object type inheritance model. It is designed to provide the basis for a domain extensible, path-based mechanism to name resources that appear in audit events in order to enable normative classification and query of events data.

The CADF Logical Resource Taxonomy's hierarchical design and node names have been derived from research into traditional compliance frameworks and evolving cloud architecture and platform management standards.

Resource names are also chosen to be meaningful to IT auditors seeking to create human readable queries on resources of "like" items as typically seen in audit frameworks. Where similar names were found, for essentially the same type of resource (or data object) by definition, the CADF agreed to resolve to a single name that could be normalized to.

A.0.2 Notes on mapping to the resource taxonomy

In some cases when classifying resources on CADF Event Records:

- A given resource might be mappable to more than one CADF Resource Taxonomy node.

- A provider’s infrastructure architecture and implementation may affect how events are mapped and cause similar events to be mapped differently across providers.

- A provider’s choices on taxonomic assignment may not map exactly to a consumer’s use of those resources.

Despite such ambiguities, classification of resources is critical to support cross-domain analysis in the vast majority of cases. When querying for CADF events, providers and consumers may need to take this into consideration, and ensure that the query is sufficiently broad to cover alternate choices. CADF seeks to engage with other standards organizations that provide compliance frameworks and standards to develop profiles that will provide more discrete guidance on how to classify provider resources.

A.0.3 Taxonomy URI

The following URI value is used to identify the CADF Logical Resource Taxonomy:

Taxonomy	Taxonomy URI
resource	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/

A.0.4 Requirements

The following are requirements on the use of the CADF Resource Taxonomy:

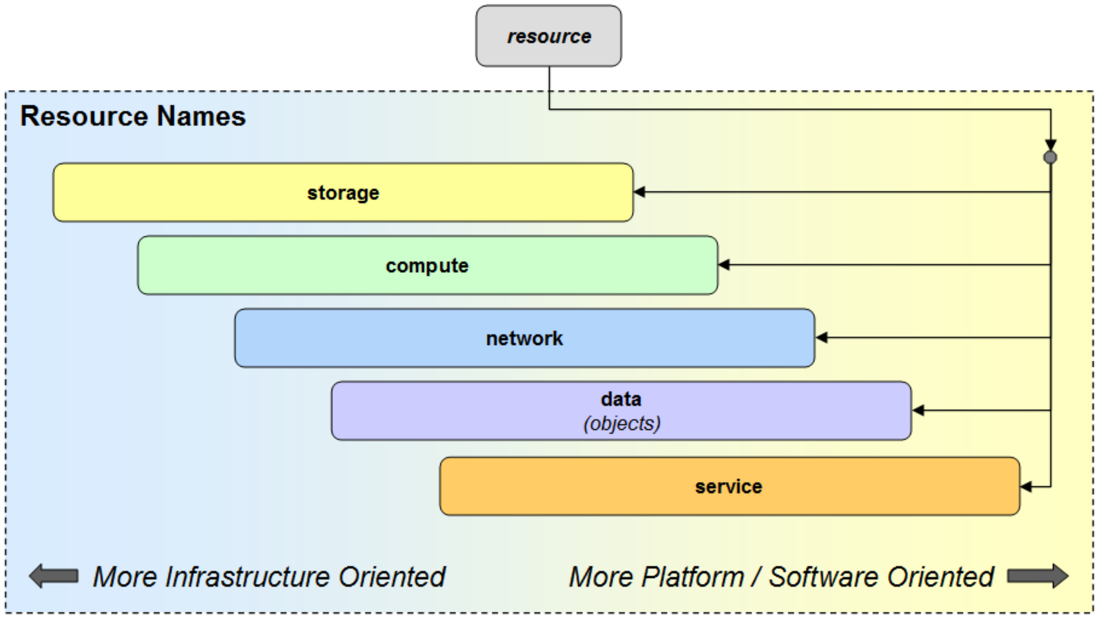
- CADF Resource typed data SHALL be classified using the CADF Resource Taxonomy, specifically as a value of its "typeURI" property.
  - Absolute path representation for CADF Resource Taxonomy values MAY be used anytime a value from this taxonomy is required.
  - Relative path representation for CADF Resource Taxonomy values SHOULD be used for the "typeURI" property value of the CADF Resource type since the base URI for the CADF Resource Taxonomy MAY be assumed for that property by context.

A.0.5 Hierarchical resource classification tree

The CADF Resource Taxonomy describes resources that are commonly used in cloud and enterprise infrastructures. This list was developed based on surveys of existing cloud architectures, deployments, and implementations. The Resource Taxonomy, however, is fully intended to be extensible by profiles that may define additional resource nodes as child nodes to the ones specified below. When doing so, however, vendors and cloud providers should be aware that this places an additional burden on the consumer to correctly comprehend the new node type, and should be careful to extend the existing tree from the most granular node that closely matches the functions of any newly-defined resource types. This approach will provide consumers with a baseline understanding of the function of the new resource type.

In all resource node diagrams that follow, any node that is outlined in a dashed style is meant to show a possible (example) extension to an already-specified CADF Resource Taxonomy node. CADF-specified nodes are shown in a solid outline style.

The following diagram shows the top-level taxonomies that are children of the CADF Resource Taxonomy as nodes. These top-level resource taxonomies include storage, compute, network, service and data.



The diagram attempts to convey that resources that may be defined under these top-level nodes may represent resources some providers may consider more "infrastructure oriented" and offer as via an IaaS service model, whereas other providers may offer resources that they instead consider to be more "platform oriented" and offer via PaaS or SaaS service models.

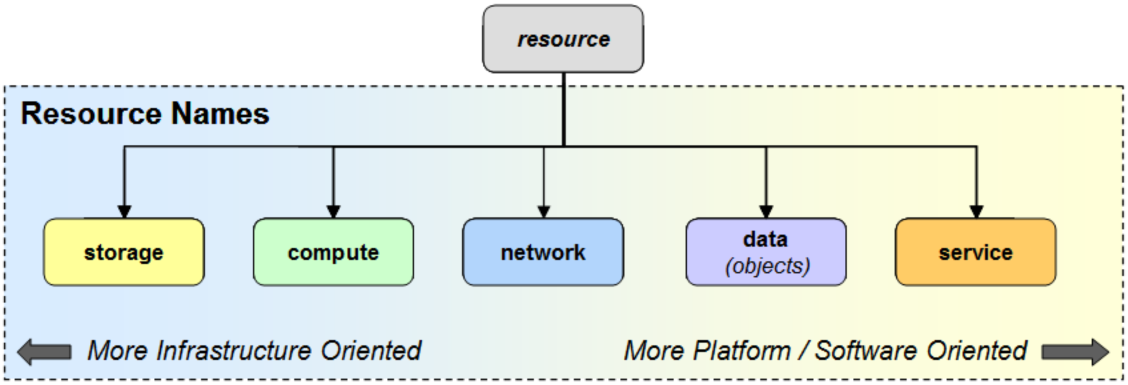
A.0.6 Logical resource classification tree

The resource taxonomy is designed to be a hierarchical tree with a fixed set of top-level nodes that are designed to be sufficient to classify any infrastructure or platform oriented resource that could be audited from a cloud deployment.

The names and descriptions for the top-level resource classifications for the "resource" taxonomy are described below:

Name	Description
storage	Logical constructs that represent storage containers
compute	Logical resources that are used to perform logical operations or calculations on data
network	Logical resources that interconnect computer systems, terminals, and other equipment allowing information to be exchanged.
service	Logical sets of operations, packaged into a single entity, that provide access to and management of cloud resources (for a given domain).
data	Logical named sets of information (objectified data) that are referenced and managed by services.

The following diagram shows these same top-level resource classifications as child nodes under the "resource" taxonomy's classification tree:



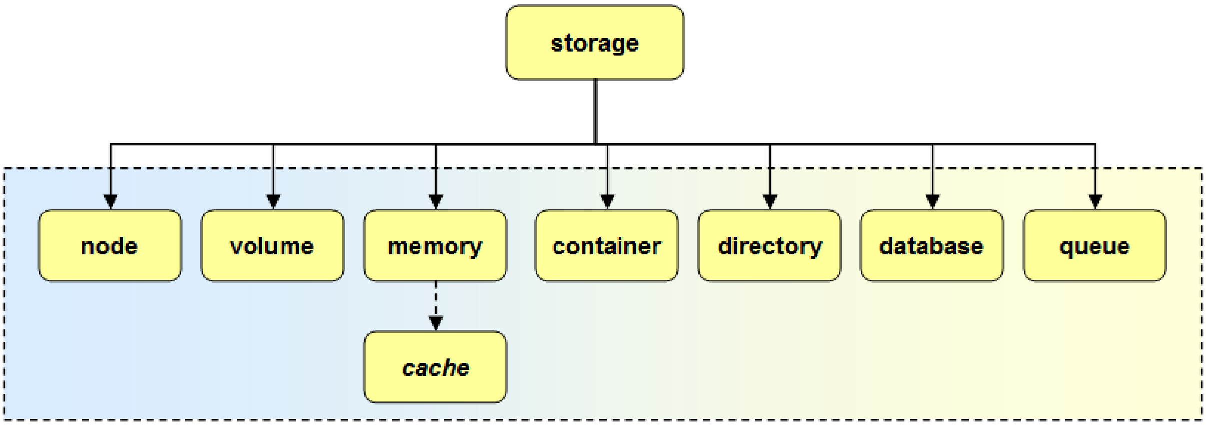
A.0.7 Storage subtree classifications

The names and descriptions for resource classifications that are children of the "storage" subtree are described below:

Name	Description
node	Logical resource that contains the necessary processing components to store data.
volume	Logical unit of persistent data storage that is may or may not be physically removable from the computer or storage system.
memory	Logical unit of data storage that is used for dynamically processing data.

container	Logical unit of storage where data objects are deposited and organized for persistent storage.
directory	Logical storage used to organize records about resources (e.g., files, subscribers, etc.) along with their locations and other metadata. Typically, these records are organized in a hierarchical structure.
database	Logical storage used to organize data to a model (schema) that reflects relevant aspects of a specific real-world application.
queue	Logical storage of a list of data awaiting processing.

The following diagram shows these same storage-oriented resource classifications as child nodes under the "storage" subtree:

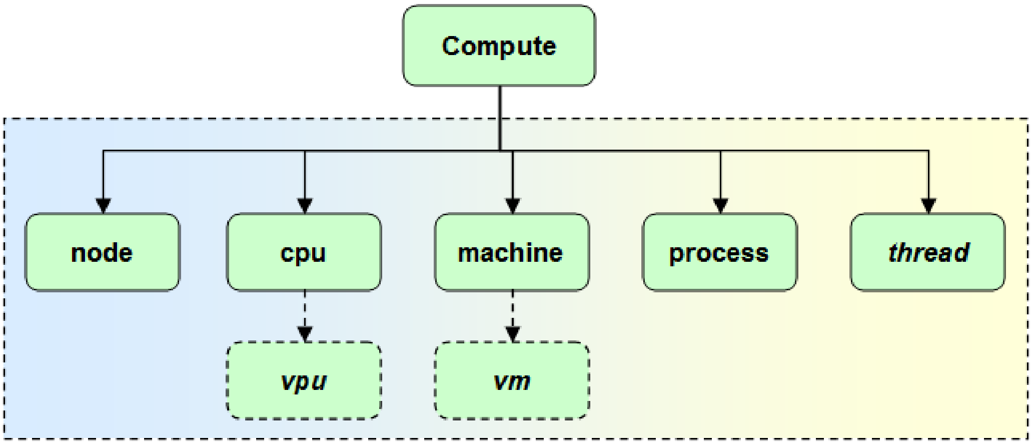


A.0.8 Compute subtree classifications

The names and descriptions for resource classifications that are children of the "compute" subtree are described below:

Name	Description
node	Logical resource that contains the necessary processing components to execute a workload.
cpu	Logical resource that represents a unit processing power that can consume a workload.
machine	Logical resource that encapsulates both CPU and Memory.
process	An instance of a granular workload, such as an application or service, that is being executed.
thread	A separable function of a running process that shares its virtual address space and system resources.

The following diagram shows these same compute-oriented resource classifications as child nodes under the "compute" subtree:



2018

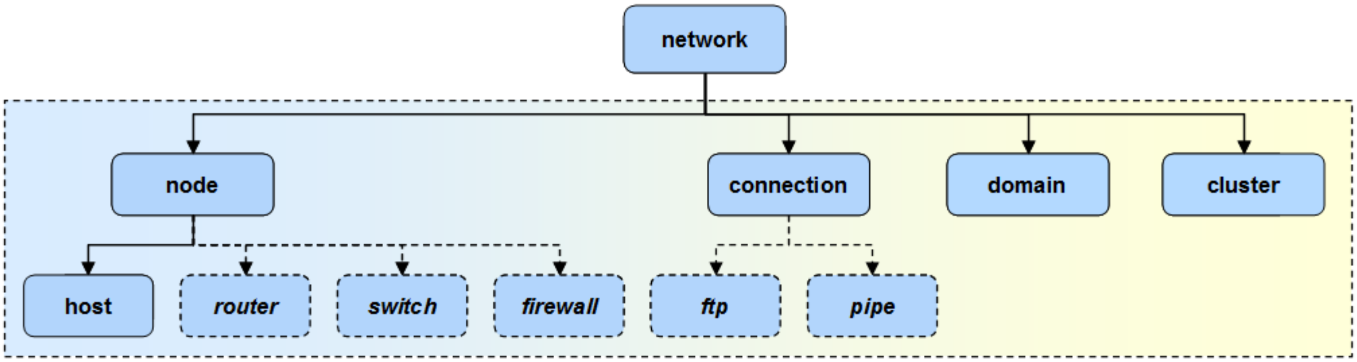
2019 A.0.9 Network subtree classifications

2020 The names and descriptions for resource classifications that are children of the "network" subtree are
2021 described below:

Name	Description
node	A logical resource that can be networked and provide services on data from network connections. A node may export zero or more endpoints (zero implies it is has not been provisioned).
host	A network node that can perform operations or calculations on data. Note: Network "nodes" should not attempt to describe details of compute or storage functions; specific compute and storage nodes exist that better suit this purpose).
connection	A single network interaction involving two or more endpoints (sources and destinations).
domain	Represents a logical grouping of networked resources
cluster	Represents a logical combination of tightly coupled, network resources.

2022 **Note**: In this model, an **endpoint** is defined as data type that contains the address or location information
2023 for a network node or service on a network (without details of the underlying service, interfaces or
2024 protocols).

2025 The following diagram shows these same network-oriented resource classifications as child nodes under
2026 the "network" subtree:



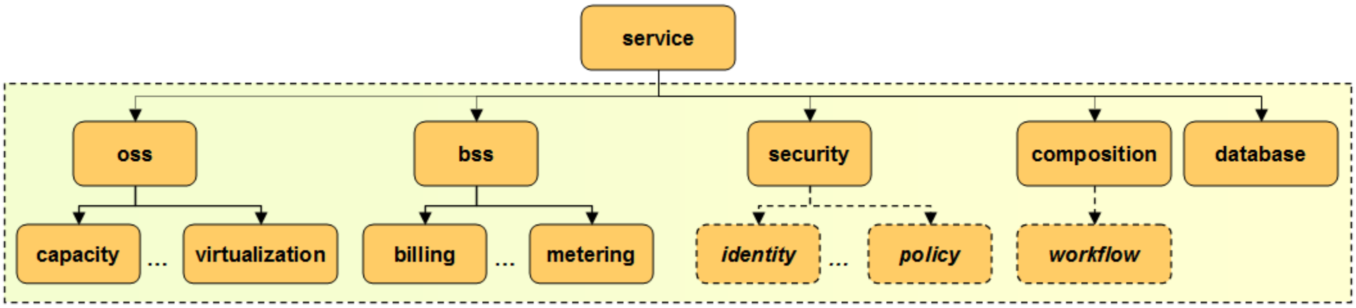
2027

A.0.10 Service subtree classifications

The names and descriptions for resource classifications that are children of the "service" subtree are described below:

Name	Descriptive Name	Description
oss	**Operational Support Services (OSS)**	The logical classification grouping for services that are identified to support operations including communication, control, analysis, etc.
bss	**Business Support Services (BSS)**	The logical classification grouping for services that are identified to support business activities.
security	**Security Services** *(or Sec-as-a-Service)*	The logical classification grouping for security services including Identity Mgmt., Policy Mgmt., Authentication, Authorization, Access Mgmt., etc. (a.k.a. "Security-as-a-Service")
composition	**Composition Services**	The logical classification grouping for services that supports the compositing of independent services into a new service offering
database	**Database Services** *(or DB-as-a-Service)*	Database services that permits substitutability to various provider implementations.

The following diagram shows these same network-oriented resource classifications as child nodes under the "service" subtree:

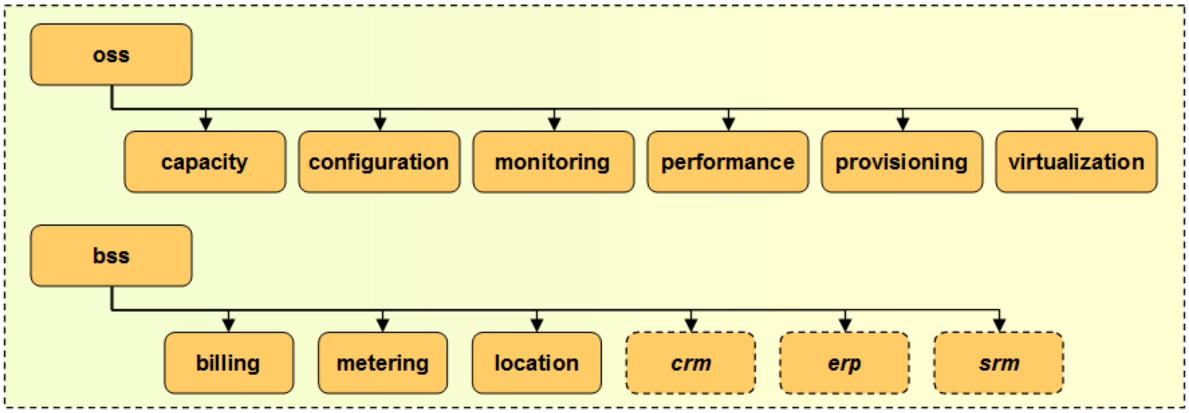


The names and descriptions for resource classifications that are children of the "oss" and "bss" subtrees are described below:

Name	Description
capacity	Operational services that ensure that the resource capacity allocated to an application (including compute, storage and networking resources) matches its current utilization.
configuration	Operational services that manage and monitor configuration changes on applications to avoid incompatibilities that can result in reduced performance or compliance failures.
monitoring	Operational services that monitor for ensure the availability of services and that they are provided in accordance with terms of Service License Agreements (SLAs)...
virtualization	Operational services that manage virtualization of compute, storage and network infrastructure.
location	Business services to manage the location, physical or virtual, of cloud based resources as well as clients (e.g., mobile devices).
billing	Business services to manage different types of charges for cloud based resources relevant to a given customer.

metering	Business Services to manage the measurement of cloud based resources (e.g., utilization, transactions, performance, etc.), often to determine how to bill for service usage.
crm	*Customer Relationship Mgmt. (CRM) Services*
erp	*Enterprise Risk Mgmt. (ERM) Services*
srm	*Service Request Mgmt. (SRM) Services*

The following diagram shows the Operational (OSS) and Business (BSS) Support Services subtree:



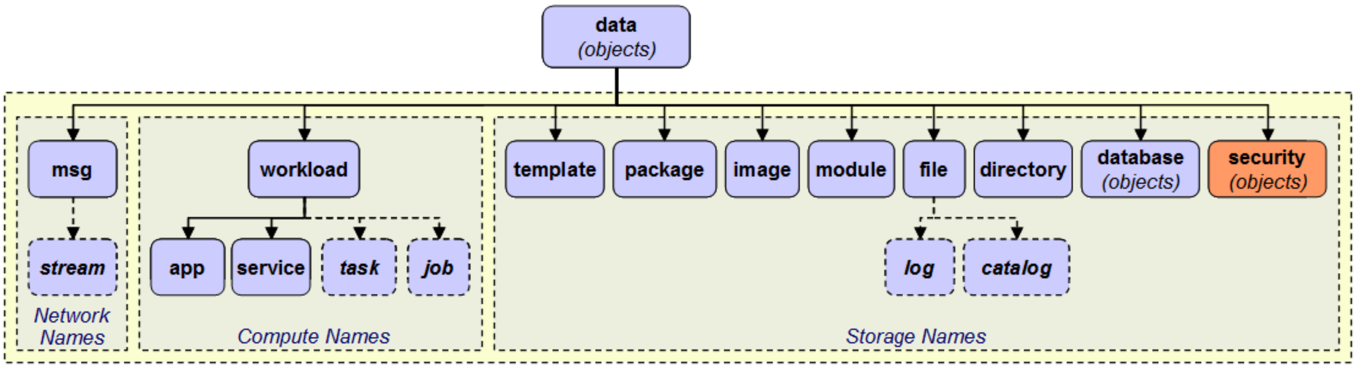
A.0.11 Data (objects) subtree classifications

The names and descriptions for resource classifications that are children of the "data" (objects) subtree are described below:

Name	Description
message	A block of information that is transmitted over a connection between networked endpoints
message/stream	A continuous message or series of messages between networked endpoints
workload	A set of data that represents the amount of work that *computational nodes* can consume at a given time
workload/app	A workload that performs a wide range of operations, some may be exported as services
workload/service	A workload that perform a single or a few specialized operations. Please see Service subtree classifications when describing specific services in events apart from generic management as compute workloads.
workload/task	*An example of a possible workload type. A workload that performs a granular, short-lived function.*
workload/job	*An example of a possible workload type. A workload that can be scheduled for processing.*
file/catalog	*An example of a possible file type*. A file used to register data items, information or metadata about them and perhaps provide links to them.
template	A logical representation of data that determines or serves as a pattern or model for representing or creating other resources.
package	A wrapped collection files and data, along with metadata, meaningful to the processing domain that will utilize it

image	A readily usable or processable set of data that can be easily transferred between processing domains.
module	A portion of a program typically aligned with a specific functional set.
file	A logical block of data for storing information, which is available to computer programs
file/log	*An example of a possible file type*. A file that used to record events from automated computer programs. Typically used to provide an audit trail that can be used to understand the activity of a system and to diagnose problems.
directory	The parent classification for all directory related data objects.
database (objects)	The parent classification for all database related data objects. Please see the section titled "Database (data objects) subtree classifications" that shows the full set of database-related classifications.
security (objects)	The parent classification for all security related data objects. Please see the section titled "Security (data objects) subtree classifications" that shows the full set of security-related classifications.

The following diagram shows these same security-oriented resource classifications as child nodes under the "data" (objects) subtree:



A.0.12 Security (data objects) subtree classifications

The following CADF Resource Taxonomy classification nodes represent commonly expressed security data objects. The CADF Resource Taxonomy attempts to represent such security related information so that it can be consistently associated as resource data on CADF Event Records where applicable.

A.0.12.1 Design considerations

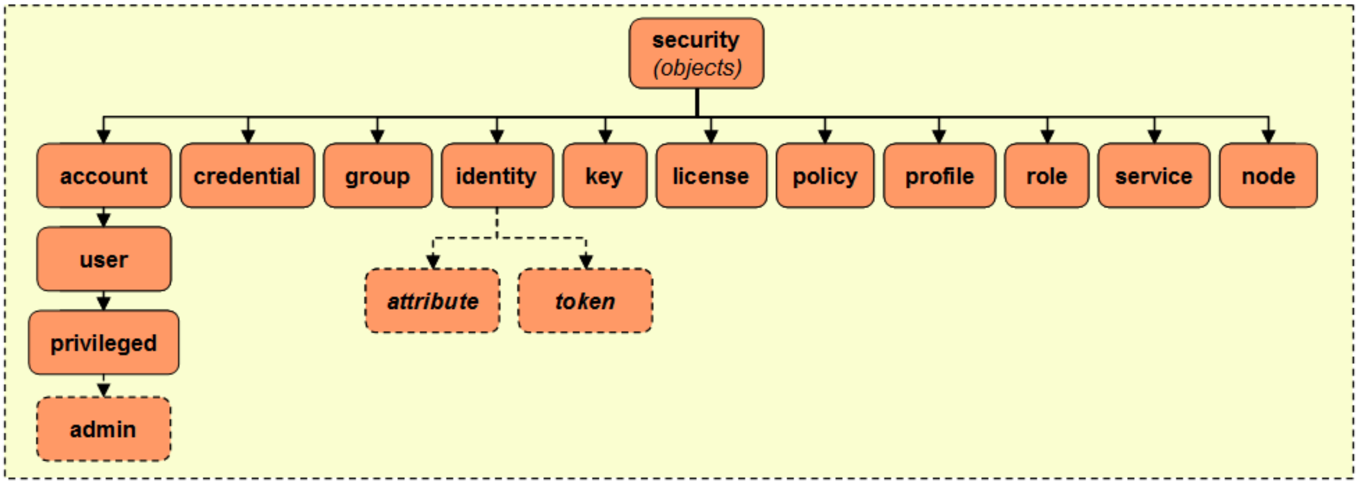
Regardless of compliance domain, a major aspect of compliance for the auditor is to verify policies that govern access to resources can be proven. It is important that representation of security information be consistent across provider deployments for auditing purposes

For example, in IT systems, users or services can attempt operations on cloud resources (as INITIATORS of ACTIONS on TARGET resources) by presenting their authorization credentials. The user or services credentials, along with other context specific information, may contribute to the evaluation of security policies (and rules) to determine if access should be granted.

The names and descriptions for resource classifications that are children of the "security" (objects) subtree are described below:

Name	Description
account	Represents a business agreement for providing regular services between a provider and consumer. (SAML Glossary)
credential	Represents security data that is transferred to establish a claimed identity. [SAML Gloss]
group	Represents named groups of users or roles can be assigned to that carries access rights or entitlements its members inherit..
identity	Represents the essence of an entity (e.g., a user or service) and may describe the entity's characteristics and properties.
key	A secret token used to protect data typically through signing or encryption. The key (or its public variant) can be provided to one or more parties that enable access to the protected data
license	Represents an authorization or permission to do something on, or with, somebody else's resources.
policy	Represents security data that contains rules and procedures that regulates resources within a system.
profile	Represents security data that defines extended rules, constraints or properties that apply to particular domains
role	Represents named jobs or functions users may be assigned. A role may carry access rights and entitlements that users inherit from being assigned to that role.
service	Represents a service acting with some (perceived) credential or authority to perform some action against another resource.
node	Represents a network node (e.g. router, server, etc.) acting with some (perceived) credential or authority to perform some action against another resource. This would be used if limited information is known to the event's observer (e.g. perhaps only an endpoint address is known).
account/user	Represents a user with an account who has the ability to use cloud resources or applications.
account/user/privileged	A user that has been assigned privileged access to (manage) resources. (Covers notion of an "administrator" and other named roles that carry special entitlements).

The following diagram shows these same security-oriented resource classifications as child nodes under the "security" (objects) subtree:

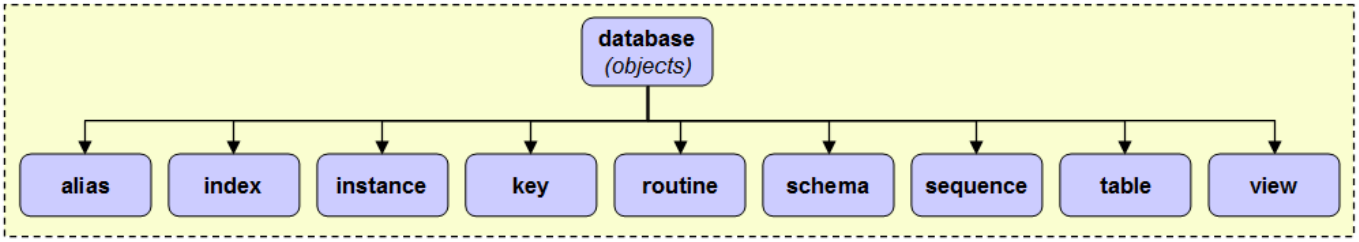


A.0.13 Database (data object) subtree classifications

The names and descriptions for resource classifications that are children of the "database" (objects) subtree are described below:

Name	Description
alias	An alias is an alternative name for an object such as a table, a view or another alias. It can be used to reference an object wherever that object can be referenced directly.
catalog	A set of tables containing information about objects in the database such as its tables, views, indexes, packages, and constraints.
constraints	Restrictions or rules associated with tables used for enforcing access controls.
index	A set of pointers that are logically ordered by the values of one or more keys. They are typically used to improve performance and ensure key uniqueness.
instance	A logical representation of the structures, memory and storage used to realize a database, its objects and data.
key	A property used to identify data stored in a database table. Typically, each table has a primary key which uniquely identifies records.
routine	An executable database object that perform operations on other database objects.
schema	A collection of named objects that are grouped logically. A schema is also a name qualifier; it provides a way to use the same natural name for several objects, and to prevent ambiguous references to those objects.
sequence	A stored object that simply generates a sequence of numbers in a monotonically ascending (or descending) order. Sequences provide a way to have the database manager automatically generate unique keys and to coordinate keys across multiple rows and tables.
table	A logical structure made up of columns and rows. At the intersection of every column and row is a specific data item called a value. There is no inherent order of the rows within a table.
trigger	Describes a set of actions that are performed in response to an operation on a specified table.
view	An alternative way of looking at the data in one or more tables.

The following diagram shows these same database-oriented resource classifications as child nodes under the "database" (objects) subtree:



A.0.14 Using the resource taxonomy

Any resource classification value MAY be represented as path segments that build upon the base Resource Taxonomy URI. However, within the context of the CADF Event Record, specifically the "typeURI" property of the CADF Resource type, the CADF Resource Taxonomy URI is assumed to be the base URI. Therefore, use of a relative URI can be viewed as equivalent to the absolute form and SHOULD be used when supplying classification values for CADF Resource types properties for compactness.

The following table includes examples of valid CADF Resource Taxonomy values as expressed in their relative and absolute URI forms:

Relative URI Form *(Preferred)*	**Equivalent Fully Qualified URI Form**
storage	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/storage
compute	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/compute
network	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/network
data	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/data
service	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/service
storage/memory/cache	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/storage/memory/cache
compute/machine	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/compute/machine
network/connection/ftp	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/network/connection/ftp
data/workload/app	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/data/workload/app
service/database/table	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/service/database/table

A.1 CADF Action Taxonomy

This section describes the action taxonomy that is used to classify the type of activity that is described in an event record. These represent values that are to be used for the "action" property for the CADF Event type.

A.1.1 Model description

The CADF Action Taxonomy is intended to normalize the set of all possible verbs that could be used to describe activity into a commonly recognized enumerated taxonomy. The goal is to provide a simple set of values that consumers can query on to get exactly the events of interest, rather than having to guess what a particular implementation might have used. The CADF event should form a familiar subject-verb-object tuple, with the 'verb' part being drawn from the Action Taxonomy.

The CADF enumerated actions are drawn from common usage and should be familiar to anyone, although it is recognized that in some cases CADF has preferred a more generic term rather than a term of art used in a particular context. For example, CADF has selected 'update' to represent updates/changes/modifications to any particular resource based on common usage in databases and simplified 'CRUD' terminology, rather than the word 'modify' which is used in other scenarios but is a synonym.

Not all actions can be taken against all targets – there is an explicit mapping between the type of resource that is the primary target of the event and the set of possible actions that can be. The corollary is that the type of action being described dictates the set of possible primary target resources, and in some cases the combination of action and primary target can further imply additional context that should be described.

A.1.2 Notes on mapping to the action taxonomy

In some cases when classifying an event's action for CADF Event Records:

- A given action might be mappable to more than one CADF Action Taxonomy value.
- A provider's infrastructure architecture and implementation may affect how events are mapped and cause similar events to be mapped differently across providers.
- A provider's choices on taxonomic assignment may not map exactly to a consumer's use of those resources.

Despite such ambiguities, classification of actions is critical to support cross-domain analysis in the vast majority of cases. When querying for CADF events, providers and consumers may need to take this into consideration, and ensure that the query is sufficiently broad to cover alternate choices. CADF seeks to engage with other standards organizations that provide compliance frameworks and standards to develop profiles that will provide more discrete guidance on how to classify provider resources.

A.1.3 Taxonomy URI

The following URI value is used to identify the CADF Action Taxonomy:

Taxonomy	Taxonomy URI
action	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/action/

A.1.4 Requirements

The following are requirements on the use of the CADF Action Taxonomy:

- This action value "monitor", or a valid extension of this value, SHALL be used for all CADF Event Records classified as type "**monitor**".
- CADF Event Records SHOULD contain a valid ACTION value from the CADF Action Taxonomy or a valid extension or profile of it where the selected value logically corresponds to the TARGET resource type using the resource mapping tables below.

A.1.5 Hierarchical action classification

The CADF Action Taxonomy is designed to be a hierarchy (much like the CADF Resource Taxonomy) whose "root" values defined in this specification can be extended to accommodate action values (or names) that are domain specific.

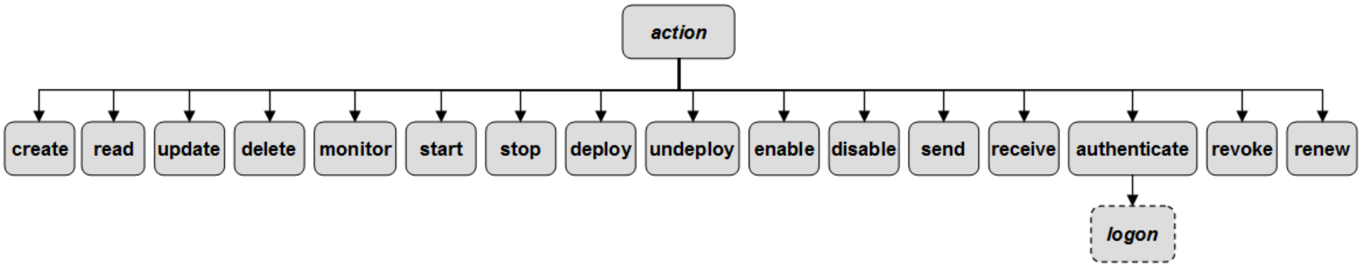
In designing the taxonomy, the CADF has acknowledged the widely accepted use of "CRUD" operations (i.e., "Create", "Read", "Update" and "Delete") used in cloud management platforms. These action values are supported for all classifying an action taken on any TARGET resource as classified by the CADF Resource Taxonomy. Additionally, the CADF Event Model describes "**monitor**" type events in which the TARGET is the subject of a monitoring action; therefore, a special action value "monitor" is specified for events so classified. For this draft, the CADF has included other values that also appear as "root" values of the CADF Action Taxonomy based upon a small, agreed upon set of use cases; however, the CADF intends to evaluate a much wider set of use cases for future draft revisions and anticipates that this taxonomy will expand to include more "root" values.

The following table lists the CADF Action Taxonomy's values along with their definitions:

Value	Description
create	The target resource described in the event was created (or an attempt was made to do so) by the initiator resource.
read	Data was read from the target resource by the initiating resource (or an attempt was made to do so).
update	One or more of the target resource's properties were modified or changed by the initiator resource.
delete	The target resource described in the event was deleted (or an attempt was made to do so) by the initiator resource.
monitor	The target resource is the subject of a monitoring action from the initiating resource.
start	The target resource is being made functional by the initiator resource and able to perform or execute operations.
stop	The initiator resource is causing the target resource to no longer be functional or able to perform or execute operations.
deploy	The target resource is being positioned or made available for use by the initiator resource, but not yet started.
undeploy	The initiator resource is causing the target resource to no longer be positioned or available for use.
enable	The target resource [that has been started[is being changed by the initiator resource to allow or permit some set of functions.
disable	The initiator resource is causing the target resource [that has been started] to disallow or block some set of functions.
send	The initiator resource is transmitting a message or data to the target resource. Note that this a separate action from that of "creating" the message.
receive	The initiator resource is receiving a message or data from the target resource. Note that this is a separate action from any action the receiver performs based upon the content of the message or with the data.

authenticate	A security request used to establish the an initiator’s identity and/or credentials to the target resource against a trusted authority.
revoke	A security request from the initiator resource to remove entitlements or privileges from a resource’s identity and/or credentials sent to the target resource (an authority).
renew	A security request from the initiator resource to renew a resource’s identity, credentials or related attributes or privileges sent to the target resource (an authority).
authenticate/login	An example extension of the authenticate action. Logon is a specialized authentication action, typically used to establish a resource’s identity or credentials for the resource to be authorized to perform subsequent actions. Note that "logon" is sometimes generalized to include the entire process used to capture a user's credentials (e.g. user ID and password); however, this action refers to only the discrete step used to actually authenticate those credentials.

The following diagram shows these same CADF Action Taxonomy values as a hierarchical taxonomy that demonstrate how they extend form the base Action Taxonomy URI defined above:



A.1.6 Taxonomy extension

The CADF Action Taxonomy can be extended to add more granular or domain-specific values. It is recommended that these domain-specific extensions should be done via CADF profiles that clearly define these extended action names, and specify the fully-qualified URI that identifies domain-specific profile to the CADF Event consumer.

A.1.7 Using the action taxonomy

Any action classification value MAY be represented as path segments that build upon the base Action Taxonomy URI. However, within the context of the CADF Event Record, specifically when used as value for the "action" property of the CADF Event Type, the CADF Action Taxonomy URI can be assumed to be the base URI. Therefore, use of a relative URI in this property can be viewed as equivalent to the absolute form and SHOULD be used when filling out a CADF Event Record for compactness.

The following table includes examples of valid CADF Action Taxonomy values as expressed in their relative and absolute URI forms:

Relative URI Form *(Preferred)*	**Equivalent Fully Qualified URI Form**
create	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/action/create
update	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/action/update

monitor	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/action/monitor
deploy	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/action/deploy
authenticate	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/action/authenticate

A.2 CADF Outcome Taxonomy

The Outcome Taxonomy defines the normative set of valid event result (or outcome) values that are required by certain data schema elements in this specification. These represent values that are to be used for the "outcome" property for the CADF Event type.

A.2.1 Design considerations

General considerations

This version of the outcome taxonomy is designed to support the following Design considerations which have been derived from use cases the CADF examined in DSP2028 "*Cloud Auditing Data Federation (CADF) Use Case White Paper*".

- Every "activity" event that represents a deliberate action (see CADF Action Taxonomy), and as opposed to a state indication) should have some form of outcome classification which describes the outcome and/or result of that attempted action.
- Outcome classification should roughly categorize events into very high level groups conforming to common understanding of normal outcomes (e.g., “it worked”, “it failed”, “don't know”, etc.)
  - This supports simplified queries for commonly-asked questions like “show me all failed logins.”
  - Classifications should be derived from high-level compliance reporting requirements that ask for events with specific outcomes.
  - In addition to determinate outcomes, the classification must account for scenarios where the outcome is unknown, or where the outcome is not yet known (e.g., for long running transactions).
- Each classification should be assigned a text value (or label) that is human readable.

Operational considerations

In general, “operational” queries are designed to determine whether a system is functioning properly, and outcomes for events with operational significance should usually indicate whether the action was successful or not. If the attempted action failed, this will usually indicate some sort of system problem, and the related “reason” should indicate the broad class of why the action failed.

Security and compliance considerations

By contrast, security or compliance related queries will typically be designed to determine whether people are conforming to one or more security or compliance policies, and hence outcomes will typically indicate how the event action was resolved against those policies relative to the perspective of the OBSERVER).

A.2.2 Taxonomy URI

The following URI value is used to identify the CADF Outcome Taxonomy:

Taxonomy	Taxonomy URI
outcome	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/outcome/

A.2.3 Requirements

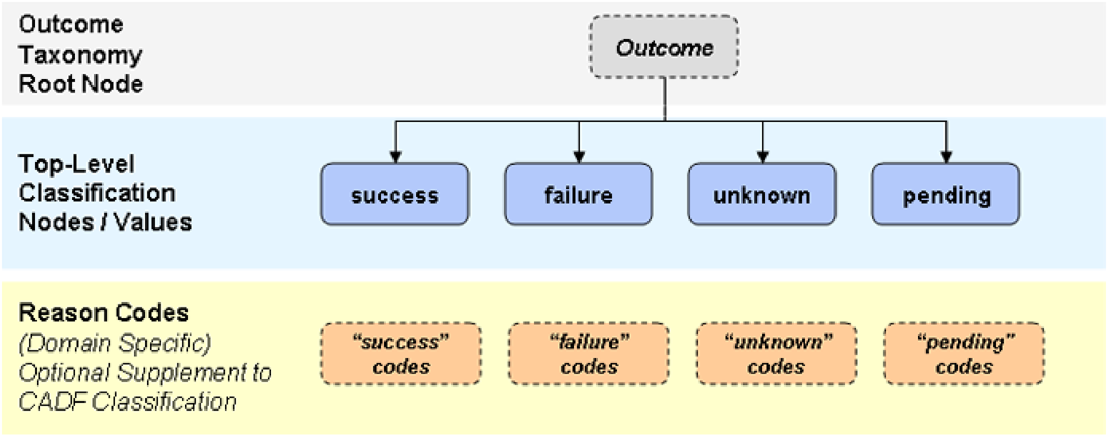
The following are requirements on the use of the CADF Outcome Taxonomy:

- Profiles or extensions of this specification SHALL NOT define any additional top-level nodes for the CADF Outcome Taxonomy. This means that sibling values to "success", "failure", "unknown" or "pending" SHALL NOT be permitted.
- Profiles or extensions of this specification MAY define new outcome values that extend from the values already defined by this specification (by extending their names with additional path segments).

A.2.4 Hierarchical action classification

The CADF Outcome Taxonomy is designed to be a hierarchy (much like the CADF Resource Taxonomy) whose "root" values defined in this specification can be extended to accommodate outcome values (or names) that are domain specific. In addition to the base outcome value, an optional domain-specific "reasonCode" can be provided as a separate property to augment the value from the CADF Outcome Taxonomy.

The following diagram shows that the CADF Outcome Taxonomy as a hierarchical model:


Outcome Taxonomy Root Node
Outcome
Top-Level Classification Nodes / Values
success
failure
unknown
pending
Reason Codes
(Domain Specific) Optional Supplement to CADF Classification
"success" codes
"failure" codes
"unknown" codes
"pending" codes

A.2.5 Taxonomy values

The CADF Outcome Taxonomy provides the following "root" outcome values that SHALL be used for any extensions or profiles of this specification. They are:

Value	Description
success	The attempted action completed successfully with the expected results.
failure	The attempted action failed due to some form of operational system failure or because the action was denied, blocked or refused in some way.
unknown	The outcome of the attempted action is unknown and it is not expected that it will ever be known.
pending	The outcome of the attempted action is unknown, but it is expected that it will be known at some point in the future.

	A future event correlated with the current event may provide additional detail.

A.2.6 Requirements

The following are requirements on the use of the CADF Outcome Taxonomy:

- Extensions or profiles of this specification SHALL NOT define new "root" values for the CADF Outcome Taxonomy.
- Extensions or profiles of this specification MAY define new outcome values that extend from the "root" values of the CADF Outcome Taxonomy defined in this specification.

A.2.7 Using the outcome taxonomy

Any outcome classification value MAY be represented as path segments that build upon the base Action Taxonomy URI. However, within the context of the CADF Event Record, specifically when used as value for the "outcome" property of the CADF Event Type, the CADF Outcome Taxonomy URI can be assumed to be the base URI. Therefore, use of a relative URI in this property can be viewed as equivalent to the absolute form and SHOULD be used when filling out a CADF Event Record for compactness.

The following table includes examples of valid CADF Outcome Taxonomy values as expressed in their relative and absolute URI forms:

Relative URI Form *(Preferred)*	**Equivalent Fully Qualified URI Form**
success	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/outcome/success
failure	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/outcome/failure
unknown	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/outcome/failure
pending	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/outcome/pending

A.2.8 Considerations when using "unknown" or "pending" values

- An OUTCOME that is set to the value of “unknown” is expected to never have a known outcome value by the OBSERVER.
    - As an example, this might occur if some data is sent to a third-party via an unreliable protocol such as UDP – the sender has no expectation that it will ever know if the data was received correctly.
- By contrast, a “pending” OUTCOME value indicates that the OBSERVER has detected an ongoing activity and is waiting for the final results to come in.
    - An example might be a long-running database transaction or similar activity. In general the rationale for issuing such an event is to notify consumers as soon as possible (or at the correct point in the time-ordered stream of events) that the activity is taking place. Since the outcome is also important, however, it is anticipated that the OBSERVER will usually follow this type of event with a nearly identical event that includes the final outcome; this follow-up event could be linked to the original “pending” event(s) by some type of correlation identifier.

A.3 Treatment of INITIATOR, TARGET, and OBSERVER

A.3.1 Overview

As explained in the CADF Event Model, the CADF Event Record, includes the description of top-level component resources. These resources include the INITIATOR, TARGET and OBSERVER, along with any other REPORTERS that contribute to the record. Orthogonal to this model is the CADF concept of a "resource", which refers to some cloud (or IT) resource that can be described relative to the provider's environment.

In the CADF Event Record, the INITIATOR, TARGET, and OBSERVER are just named roles that a given CADF Resource takes on with respect to the described activity (i.e., or ACTION) of the event record. In some events a single CADF Resource may appear as the INITIATOR, in others as the TARGET, and in others perhaps an OBSERVER or REPORTER.

A.3.2 Treatment of INITIATOR

The INITIATOR as described in a CADF Event entity reflects the resource that caused the described event activity to take place. Ultimately this is almost always an actual physical person, but note that in most circumstances the visibility of the OBSERVER will likely not extend out to the point where that person is uniquely identifiable. For example, an administrator may configure a service to perform some task; in this case the service will likely act as the INITIATOR in an event. Or a user may be issued a SAML token that is then accepted for access to a resource - the access grantor may only see the token and never know the identity or even the user account of the user.

Naturally, then, the CADF Event Record's INITIATOR would be described as resources that can take action along with descriptive information about those resources (such as tokens or credentials) that could ultimately be used to resolve their unique identity within the provider. If such resolution is not performed by the original OBSERVER but by a downstream REPORTER, the downstream REPORTER can attach the resolved resource to the CADF Event Record.

Not all CADF Resources therefore can act as INITIATORS - it would not make much sense, for example, for a "File" resource to be listed as the INITIATOR. In fact, INITIATORS in most cases are acting as security principals in the context of the event, and as such will generally be resources located under the 'data/security' branch of the CADF Resource Taxonomy. However, in some cases, INITIATORS may be services that are acting using some authorization and be found under the 'service' branch of the CADF Resource Taxonomy. Still in other cases, INITIATORS may be network nodes under the 'network/node' branch of the CADF Resource Taxonomy.

Please note that If developers of this specification do not find the precise resources needed to describe the environment, the CADF Resource Taxonomy can be extended by profile if necessary to provide domain-specific values (names).

Examples of valid INITIATOR resources include: '

- data/security/identity
- data/security/account/user
- service
- network/node/host

As a best practice, developers are therefore encouraged to use the resources available under the three identified CADF Resource Taxonomy branches:

- data/security
- network/node
- service

A.3.3 Treatment of TARGET

Any CADF Resource can appear as the TARGET within a CADF Event Record, since conceivably any resource that we describe could be affected by enterprise IT activity. As such CADF places no constraints on which CADF Resources can take on the role of TARGET.

A.3.4 Treatment of OBSERVER

The OBSERVER describes the resource that detected the activity and caused a CADF Event Record to be generated while filling out the record with data based upon its perspective. Like the INITIATOR, therefore, the set of resource capable of reporting an observation may be limited to resources capable of actually observing and creating records such as running applications or services. Such services are typically located under the '/service' branch of the CADF Resource Taxonomy, and as before the list can be extended by profile as necessary.

Examples of valid OBSERVER resources include:

- service/oss/monitoring
- service/oss/configuration
- service/security/policy
- service/security/authentication

As a best practice, developers are therefore encouraged to use the resources available under the following CADF Resource Taxonomy branches:

- service

A.4 Using the CADF Taxonomies to create CADF Event Records

This section provides some general rules, along with examples, for using the CADF defined taxonomies when classifying components of the CADF Event Model while constructing proper CADF Event Records.

A.4.1 General rules

The general algorithm that is followed to create a CADF Event Record is:

1. Identify the OBSERVER that detects the activity and reports it and find the resource type name from the CADF Resource Taxonomy that best describes it.

2. Identify the primary purpose of the OBSERVER and its perspective and ask "what is the OBSERVER's purpose and what domain resource objects does have direct knowledge of?".

   - For example, a low-level file-system driver, acting as an OBSERVER, would not know that a particular file contains account information; conversely an account management application should not be reporting low-level file activity.

3. Based on the OBSERVER’s perspective, ask "what was the resource that attempted the activity?". This resource would be the INITIATOR of the event.

   a. Work down the CADF Resource Taxonomy tree to find the most granular name that best describes the INITIATOR resource.

4. Based on the OBSERVER’s perspective, what was the primary resource that was the intended TARGET resource of the activity (whether the action was successful or not)?

   a. Work down the CADF Resource Taxonomy tree to find the most granular name that best describes the TARGET resource.

5. Based on the OBSERVER’s perspective, select the most appropriate available ACTION from the CADF Action Taxonomy that describes the attempted activity.

   a. Work down the CADF Action Taxonomy tree to find the most granular value that best describes the ACTION. Attempt to use an ACTION value that the CADF recommends for use with the selected TARGET resource.

6. Based on the OBSERVER’s perspective, select the most appropriate result or OUTCOME of the attempted ACTION from the CADF Outcome Taxonomy.

   a. Work down the CADF Outcome Taxonomy to select the OUTCOME value that reflects the result the OBSERVER can directly attest it observed at the time the event record is being created.

A.4.2 Examples

A.4.2.1 Account creation

An consumer account administrator logs in to a cloud's account management service and successfully creates a new user account.

1. Identify the OBSERVER that detects the activity and reports it and find the resource type name from the CADF Resource Taxonomy that best describes it.

   The OBSERVER was the account management service as it processes the account addition. Using the CADF Resource Taxonomy, the value "**service/security/account**" could be a valid extended classification for an account management service.

2. Identify the primary purpose of the OBSERVER and its perspective and ask "what is the OBSERVER's purpose and what domain resource objects does have direct knowledge of?".

   The purpose of the account management service, as the OBSERVER, is to report activities on the customer account. Therefore, the event type would be "activity".

3. Based on the OBSERVER’s perspective, ask "what was the resource that attempted the activity?". This resource would be the INITIATOR of the event.

   The INITIATOR of the activity, using the resource taxonomy, would be the "administrator" of the consumer account (e.g., "**data/security/account/user/admin"**).

4. Based on the OBSERVER’s perspective, what was the primary resource that was the intended TARGET resource of the activity (whether the action was successful or not)?

   The TARGET of the activity, using the CADF Resource Taxonomy, would be the customer "account" which is affected by the activity (e.g., "**data/security/account"**).

5. Based on the OBSERVER’s perspective, select the most appropriate available ACTION from the CADF Action Taxonomy that describes the attempted activity.

The observed ACTION taken on the customer account, using the CADF Action Taxonomy, would be "**create**" .

6. Based on the OBSERVER's perspective, select the most appropriate result or OUTCOME of the attempted ACTION from the CADF Outcome Taxonomy.

   The observed OUTCOME of the activity, using the CADF Outcome Taxonomy, would be "**success**".

A.4.2.2 User Authentication

A user successfully logs in to a CRM service using their assigned account.

1. Identify the OBSERVER that detects the activity and reports it and find the resource type name from the CADF Resource Taxonomy that best describes it.

   The OBSERVER was the CRM service that accepted the authentication request and reports the activity (e.g., "**service/bss/crm**").

2. Identify the primary purpose of the OBSERVER and its perspective and ask "what is the OBSERVER's purpose and what domain resource objects does have direct knowledge of?".

   The purpose of the CRM service, as the OBSERVER, is to report any user activities taken against it (including authentication). Therefore, the event type would be "activity".

3. Based on the OBSERVER's perspective, ask "what was the resource that attempted the activity?". This resource would be the INITIATOR of the event.

   The INITIATOR of the activity, using the resource taxonomy, would be the "user" of the consumer account (e.g., "**data/security/account/user"**).

4. Based on the OBSERVER's perspective, what was the primary resource that was the intended TARGET resource of the activity (whether the action was successful or not)?

   The TARGET of the activity, using the CADF Resource Taxonomy, would be the CRM service itself (e.g., "**service/bss/crm"**).

5. Based on the OBSERVER's perspective, select the most appropriate available ACTION from the CADF Action Taxonomy that describes the attempted activity.

   The observed ACTION taken on the customer account, using the CADF Action Taxonomy, would be "**authenticate**" .

6. Based on the OBSERVER's perspective, select the most appropriate result or OUTCOME of the attempted ACTION from the CADF Outcome Taxonomy.

   The observed OUTCOME of the activity, using the CADF Outcome Taxonomy, would be "**success**" .

B. Best practices

B.0 Treatment of timestamps in CADF Event Records

CADF Event Records seek to represent time so that consumers can make intelligent decisions about how each event, within the same activity domain, relates to each other temporally. For example, events captured within an enterprise whose employees access cloud services should be comparable temporally with events at the cloud provider. This task can be surprisingly difficult given that there is no guarantee that any given source of event data has a clock that is in any way synchronized with any other system's clock, not to mention when one has to deal with multiple timezones and timezone representations.

In order to remove ambiguity, timestamps in CADF Event Records should be recorded in local time, meaning the 24-hour clock time for the local time zone, with explicit reference to the UTC timezone offset (see the definition for the data type). This allows for common use cases such as "after hours" analysis of access to local systems, as well as absolute comparison with events from other systems across the globe. To prescribe this concept, the CADF has defined its own Timestamp data type which is used throughout its data model and schema.

The CADF Event Record has several entities and complex data types where a CADF Timestamp type value appears as a property. The following table shows all such CADF Timestamp typed properties along with their parent entity and a description of their intended use.

CADF Timestamp Properties		
Parent Entity Name	**Property Name**	**Property Description**
Log	logTime	The time the log was last updated. This time may be used to represent the time the log creation is complete and ready for subsequent consumption (e.g., federation, processing or archival).
Log	beginTime	The beginning time for the time period of event records within the log.
Log	endTime	The ending time for the time period of event records within the log.
Report	reportTime	The time the report was last updated. This time may be used to represent the time the report creation is complete and ready for subsequent consumption (e.g., federation, processing or archival).
Report	beginTime	The beginning time for the time period of event records within the report.
Report	endTime	The ending time for the time period of event records within the report.
Event	eventTime	The OBSERVER's best estimate as to the time the Actual Event occurred or began (note that this may differ significantly from the time at which the OBSERVER is processing the CADF Event Record).
Reporterstep	reporterTime	The time a REPORTER adds its Reporterstep entry into the REPORTERCHAIN (which follows completion of any updates to or handling of the corresponding CADF Event Record).

C. Mapping CIMI Events to CADF Event Record

in future draft revisions of this specification, the CADF will develop a section to describe how to map CIMI Events to CADF Event Records.

D. Mapping CIM Indications to CADF Event Records

in future draft revisions of this specification, the CADF will develop a section to describe how to map CIM Indications to CADF Event Records.

E. Bibliography (Informative)

This clause lists references that are helpful for the application of this guide.

Tag	Reference
[Navajo:2009]	Miguel Montarelo Navajo et al. "Draft Report of the Task Force on Interdisciplinary Research Activities applicable to the Future internet", A Draft Report of the DG INFSO Task Force of the European Commission on the Future Internet Content focusing on FOT Federated, Open and Trusted Platforms), European Commission 2009. p.p. 3-5., June 2009, http://www.future-internet.eu/fileadmin/documents/reports/FI-content/Report_on_the_Future_Internet_Content_v4.1.pdf
[Kobielus:2006]	Kobielus, James, Title: "New Federation Frontiers In IP Network Services", Source: Business Communications Review, v36 n8 p37(6), ISSN**:** 0162-3885, August 2006, http://direct.bl.uk/bld/PlaceOrder.do?UIN=194282677&ETOC=RN&from=searchengine
[CNSS4009]	CNSS Instruction No. 4009, Committee on National Security Systems (CNSS), *National Information Assurance (IA)*. 26 April 2010, http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf
[DMTF DSP2028]	DMTF White Paper DSP2028, *Cloud Auditing Data Federation (CADF) Use Case White Paper*, *Version: 1.0.0a*, 26 June 2012, http://dmtf.org/sites/default/files/standards/documents/DSP2028_1.0.0a.pdf
[EPTS Glossary]	Event Processing Technical Society (EPTS), David Luckham, Roy Schulte, et al. Editors, *Event Processing Glossary - Version 2.0*, July 2008, http://www.complexevents.com/wp-content/uploads/2011/08/EPTS_Event_Processing_Glossary_v2.pdf
[IBM-SQL-2012]	IBM DB2 10.1 for Linux, UNIX, and Windows; SQL Reference Volume 1, SC27-3885-00, © Copyright IBM Corporation 2012. http://public.dhe.ibm.com/ps/products/db2/info/vr101/pdf/en_US/DB2SQLRefVol1-db2s1e1010.pdf
[ISO 6709:2008]	ISO 6709:2008, TC 211 Geographic Information/Geomatics, Standard representation of geographic point location by coordinates, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53539
[ISO 9075-2011]	ISO/IEC JTC 1/SC 32/WG 3, ISO/IEC 9075-1:2011(E), "Information technology - Database languages - SQL - Part 1: Framework (SQL/Framework)", 2011-07-18, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=53681
[ISO 14001:2004]	ISO 14001:2004, *Environmental Management Systems -- Requirements with Guidance for Use*, http://www.iso.org/iso/catalogue_detail?csnumber=31807
[ISO 15288:2008]	ISO/IEC 15288:2008, System and Software Engineering – System life cycle processes, http://www.iso.org/iso/catalogue_detail?csnumber=43564
[ISO 15414:2006]	ISO/IEC 15414:2008, Information technology – Open distributed processing – Reference model – Enterprise language, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43767
[ISO 27000:2009]	ISO/IEC 27000:2009, *Information Technology -- Security Techniques -- Information Security Management Systems -- Overview and vocabulary*, http://www.iso.org/iso/catalogue_detail?csnumber=41933
[ITU X.1252]	Recommendation ITU-T X.1252, *Baseline identity management terms and definitions*, International Telecommunication Union – Technical Communication Standardization Sector (ITU-T), April 2010. http://www.itu.int/rec/T-REC-X.1252-201004-I/
[NIST-SP800-145]	P. Mell, T. Grance, *The NIST Definition of Cloud Computing SP800-145 (Draft)*. National Institute of Standards and Technology (NIST) - Computer Security Division – Computer Security Resource Center (CSRC), January 2011. http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf.
[OpenXDAS]	OpenXDAS, a SourceForge open source implementation of The Open Group's XDAS Version 1 Standard, http://openxdas.sourceforge.net/.
[RFC 2828]	IETF RFC 2828, *Internet Security Glossary,* May 2000, http://www.ietf.org/rfc/rfc2828.txt.
[RFC 3339]	IETF RFC 3339 (Proposed Standard), *Date and Time on the Internet: Timestamps*, July 2002, http://www.ietf.org/rfc/rfc3339.txt
[RFC 4949]	IETF RFC 4949, *Internet Security Glossary, Version 2*, August 2009, http://www.ietf.org/rfc/rfc4949.txt.

[SAML-Gloss-2.0] OASIS Standard, *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf.

[TOG-XDAS1] The Open Group, Distributed Audit Services (XDAS) Project, *Distributed Audit Service (XDAS) – Preliminary Specification*, http://www.opengroup.org/bookstore/catalog/p441.htm.

Change Log

Version	Date	Description
1.0.0	2012-09-21	Matt Rutkowski (IBM): Final editor draft candidate. for WIP public review.