Document Number: DSP0262

Version: 1.0.0c

Date: 2014-01-10

Cloud Audit Data Federation (CADF) - Data Format and Interface Definitions Specification

Information for Work-in-Progress version:

IMPORTANT: This document 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, it may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.

It expires on: 2014-03-31

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

Document Type: DMTF Specification

Document Status: Work In Progress

Document Language: en-US

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

Foreword 9
Acknowledgements 9
Introduction 10
Document versioning scheme 10
Cloud auditing data federation use cases 10
Auditing cloud applications independently of provider 10
Auditing hybrid cloud applications 11
Granular use cases 12
1 Scope and goals 13
1.1 Scope 13
1.2 Goals 13
1.2.1 Audit data integrity and security 14
1.2.2 Audit data set sizes and performance 14
1.2.3 Extensibility 14
1.2.4 Use cases and examples 14
1.3 Out of scope 15
1.3.1 Translation 15
1.3.2 Security policies 15
1.3.3 Forensic information 15
1.3.4 Debug information 15
1.3.5 Configuration data 16
1.3.6 Audit event alerting 16
2 Normative references 16
3 Terms and definitions 17
3.1 Interface definitions 21
3.2 Interaction model 22
3.3 Document versioning scheme 22
4 CADF Event Model 22
4.1 Basic concepts 22
4.1.1 Resource 22
4.1.2 Actual Event, Event Record, CADF Event Record 23
4.2 Required model components 23
4.2.1 Basic conceptual event model 24
4.2.2 The OBSERVER perspective 24
4.2.3 Notes 25
4.3 Conditional model components 25
4.3.1 MEASUREMENT 25
4.3.2 REASON 25
4.3.3 Basic conceptual event model with optional components 25
4.4 Optional components 26
4.4.1 Reporters and the Reporter chain 26
4.5 Types of CADF Events 28
4.5.1 Valid EventType values 28
4.5.2 EventType Requirements 29
4.6 Refinement of Event semantics based upon the selected EventType value 29
4.6.1 Resource classification 31
4.7 Mapping typical events to CADF Event Model 31
4.7.1 General approach 31
4.7.2 Use case 1: Auditing access to a controlled resource 32
4.7.3 Use case 2: Periodic monitoring resource status 34
4.7.4 Use case 3: Aggregation of resource status into an audit event 36

4.7.5 Use case 4: Auditing compliance of resource monitors 38
4.7.6 Use case 5: Auditing controlled resource accesses 40
5 Data model and schema conventions 42
5.1 Namespace URIs and alias conventions 42
5.1.1 Namespace URIs 42
5.1.2 Namespace aliases 43
5.2 Namespaces and namespace aliases 43
5.2.1 Requirements 43
5.2.2 XML usage example 44
5.2.3 JSON usage example 44
5.3 Reserved Namespace URIs and aliases for RESOURCES in the CADF Event Model 45
5.4 Entity naming conventions 46
5.4.1 Requirements 46
5.4.2 XML naming requirements 46
5.5 Property constraints 46
5.5.1 "Required" constraint: 46
5.6 Format-specific representations 47
5.6.1 Entity Type URIs 47
5.6.2 Language identification 48
5.6.3 Rules for XML and JSON format representation 49
6 CADF Entities and data types 50
6.1 Extensibility mechanisms 50
6.1.1 Attachments 51
6.1.2 Derivation 51
6.1.3 Tags 52
6.2 Basic data types 52
6.2.1 General requirements 52
6.2.2 boolean 52
6.2.3 integer 52
6.2.4 double 52
6.2.5 string 52
6.2.6 duration 52
6.2.7 URI 53
6.2.8 Basic type translation to JSON from XML 53
6.3 CADF basic data types 53
6.3.1 Identifier type 53
6.3.2 Path type 56
6.3.3 Tag type 59
6.3.4 Timestamp type 60
6.4 Composition of data types in CADF 63
6.4.1 Array Syntax 63
6.4.2 Map type 65
6.5 CADF complex data types 66
6.5.1 Attachment type 67
6.5.2 Credential type 69
6.5.3 Endpoint type 71
6.5.4 Eventset type 72
6.5.5 Geolocation type 75
6.5.6 Host type 83
6.5.7 Metric and measurement types 84
6.5.8 Reason type 88
6.5.9 Reporterstep type 91
6.5.10 Resource type 93
6.5.11 Resultset type 96
6.6 CADF Entities 99
6.6.1 Event (data) type 99

6.6.2 Log type 110
6.6.3 Report type 114
7 CADF Interfaces 117
7.1 CADF Query Interface 117
7.1.1 Design Notes 117
7.1.2 Requirements 117
7.1.3 CADF Query Syntax 117
7.1.4 CADF Query Syntax subset 117
7.1.5 Semantics of path values in filters 119
7.1.6 Limiting query results using Pagination 121
7.1.7 Case sensitivity 124
7.1.8 Examples using the CADF Query Syntax 125
8 CADF entity signing 127
9 CADF profiles 127
9.1 Requirements 127
10 Future considerations 128
ANNEX A CADF Event Model component classification 129
A.1 General use of the reserved classification value "unknown" 129
A.1.1 Requirements 129
A.2 CADF Resource Taxonomy 129
A.2.1 Model description 129
A.2.2 Notes on mapping to the resource taxonomy 129
A.2.3 Taxonomy URI 130
A.2.4 Requirements 130
A.2.5 Hierarchical resource classification tree 130
A.2.6 Logical resource classification tree 131
A.2.7 Storage subtree classifications 132
A.2.8 Compute subtree classifications 133
A.2.9 Network subtree classifications 134
A.2.10 Service subtree classifications 134
A.2.11 Data (objects) subtree classifications 136
A.2.12 Security (data objects) subtree classifications 137
A.2.13 Database (data object) subtree classifications 138
A.2.14 Using the resource taxonomy 139
A.3 CADF Action Taxonomy 140
A.3.1 Model description 140
A.3.2 Notes on mapping to the action taxonomy 140
A.3.3 Taxonomy URI 140
A.3.4 Requirements 140
A.3.5 Hierarchical action classification 141
A.3.6 Taxonomy extension 143
A.3.7 Using the Action Taxonomy 143
A.4 CADF Outcome Taxonomy 143
A.4.1 Design considerations 144
A.4.2 Taxonomy URI 144
A.4.3 Requirements 144
A.4.4 Hierarchical action classification 144
A.4.5 Taxonomy values 145
A.4.6 Requirements 145
A.4.7 Using the Outcome Taxonomy 145
A.4.8 Considerations when using "unknown" or "pending" values for action classification 146
A.5 Treatment of INITIATOR, TARGET, and OBSERVER 146
A.5.1 Overview 146
A.5.2 Treatment of INITIATOR 146

A.5.3 Treatment of TARGET 147
A.5.4 Treatment of OBSERVER 147
A.6 Using the CADF Taxonomies to create CADF Event Records 148
A.6.1 General rules 148
A.6.2 Example: Account creation 148
A.6.3 Example: User authentication 149
ANNEX B Best practices 151
B.1 Treatment of “extra” contextual event data 151
B.1.1 Use case: Debug Information 151
B.2 Treatment of timestamps in CADF Event Records 151
B.2.1 Filling in Timestamps 152
B.2.2 Handling Activities with Duration 153
B.3 Handling Complex Events 153
B.3.1 Resource Context 154
B.3.2 Multi-Target Events 155
B.3.3 Multiple Affected Targets 157
B.3.4 Request-Response Events 157
B.3.5 Action-Reaction Events 158
B.3.6 Correlated Events 159
ANNEX C Mapping DMTF CIM Indications to CADF Event Record 161
C.1 Informative References: 161
ANNEX D Mapping DMTF CIMI Events to CADF Event Records 162
D.1 Recommended mapping rules 162
D.1.1 cadf:event.id 162
D.1.2 cadf:event.eventType 162
D.1.3 cadf:event.eventTime 162
D.1.4 cadf:event.action 162
D.1.5 cadf:event.outcome 163
D.1.6 cadf:event.initiator 163
D.1.7 cadf:event.target 163
D.1.8 cadf:event.severity 163
D.1.9 cadf:event.measurements 163
D.1.10 cadf:event.attachments 164
D.2 Informative References 164
ANNEX E Mapping CADF Query Syntax to XML and JSON 165
E.1 XML mapping examples 165
E.1.1 Sample event data set used for all examples 165
E.1.2 Resource create query 166
E.1.3 Resource creation failure query 168
E.1.4 Reporter time query 168
E.1.5 Time range query 168
E.1.6 Pagination query 168
E.2 JSON mapping examples 169
E.2.1 Resource create query 169
E.2.2 Pagination query 170
ANNEX F Examples of the CADF Query Interface over HTTP 171
F.1.1 Create events query over HTTP 171
ANNEX G (informative) Change log 173
Bibliography 174

Figures

Figure 1 – Hosting application at a cloud provider; tools use open standards 11
Figure 2 – Moving an application from Cloud Provider A to Provider B; tools unchanged 11

Figure 3 – Company aggregates audit data from hybrid cloud application across various deployments ...12
Figure 4 – CADF Event Model: Basic components ...24
Figure 5 – CADF Event Model: Basic and conditional model components ...26
Figure 6 – Example of REPORTERCHAIN construction ...28
Figure 7 – Use case 1: Conceptual mapping ...34
Figure 8 – Use case 2: Conceptual mapping ...36
Figure 9 – Use case 3: Conceptual mapping ...38
Figure 10 – Use case 4: Conceptual mapping ...40
Figure 11 – Use case 5: Conceptual mapping ...42
Figure 12 – Top-level CADF Resource Taxonomy Hierarchy ...132
Figure 13 – CADF Resource Taxonomy - Storage subtree ...133
Figure 14 – CADF Resource Taxonomy - Compute subtree ...133
Figure 15 – CADF Resource Taxonomy - Network subtree ...134
Figure 16 – CADF Resource Taxonomy - Service subtree ...135
Figure 17 – CADF Resource Taxonomy - BSS, OSS, Orchestration subtree ...136
Figure 18 – CADF Resource Taxonomy - Data subtree ...137
Figure 19 – CADF Resource Taxonomy - Security subtree ...138
Figure 20 – CADF Resource Taxonomy - Database subtree ...139
Figure 21 – CADF Action Taxonomy Hierarchy ...143
Figure 22 – CADF Outcome Taxonomy Hierarchy ...145

Tables

Table 1 – Resource definition ...23
Table 2 – Types of events ...23
Table 3 – Required CADF Event Model components ...23
Table 4 – Conditional MEASUREMENT component definition ...25
Table 5 – Conditional REASON component definition ...25
Table 6 – REPORTERCHAIN definition ...26
Table 7 – CADF: Reporter roles ...27
Table 8 – EventType definition ...28
Table 9 – Valid EventType values ...29
Table 10 – Event component semantics for "monitor" type events ...30
Table 11 – Event component semantics for "activity" type events ...30
Table 12 – Event component semantics for "control" type events ...30
Table 13 – General mapping approach using the CADF Event Model ...32
Table 14 – Use case 1: Mapping of actors and elements to the CADF Event Model ...33
Table 15 – Use case 2: Mapping of actors and elements to the CADF Event Model ...35
Table 16 – Use case 3: Mapping of actors and elements to the CADF Event Model ...37
Table 17 – Use case 4: Mapping of actors and elements to the CADF Event Model ...39
Table 18 – Use case 5: Mapping of actors and elements to the CADF Event Model ...41
Table 19 – Namespaces ...43
Table 20 – Basic type translation from XML to JSON ...53
Table 21 – Sample array type property of cadf:attachment type ...63
Table 22 – Sample array type property of cadf:identifier types ...64
Table 23 – Map type properties ...66

Table 24 – CADF Attachment type properties 67
Table 25 – Credential type properties 69
Table 26 – Endpoint type properties 71
Table 27 – Eventset data type properties 74
Table 28 – Geolocation type properties 76
Table 29 – Host type properties 83
Table 30 – Metric type properties 85
Table 31 – Measurement type properties 85
Table 32 – Reason type properties 89
Table 33 – Reporterstep type properties 92
Table 34 – Resource type properties 94
Table 35 – Resultset data type properties 97
Table 36 – Event data type properties 100
Table 37 – Log data type properties 112
Table 38 – Report data type properties 115
Table 39 – CADF Event data type properties to return based upon “detailLevel” and “eventType” 122
Table 40 - Properties to return based upon CADF Type and “detailLevel” 123
Table A–1 – Resource taxonomy’s top-level resource classification names 131
Table A–2 – Resource classification names for the storage classification subtree 132
Table A–3 – Resource classification names for the compute classification subtree 133
Table A–4 – Resource classification names for the network classification subtree 134
Table A–5 – Resource classification names for the service classification subtree 134
Table A–6 – Resource classification names for the “oss” and “bss” classification subtrees 135
Table A–7 – Resource classification names for the data (objects) classification subtree 136
Table A–8 – Resource classification names for the security (objects) classification subtree 137
Table A–9 – Resource classification names for the database (objects) classification subtree 138
Table A–10 – CADF Resource Taxonomy values expressed in relative and absolute URI forms 139
Table A–11 – CADF Action Taxonomy informal grouping color key 141
Table A–12 – CADF Action Taxonomy values 141
Table A–13 – CADF Action Taxonomy values expressed in relative and absolute URI forms 143
Table A–14 – CADF Outcome Taxonomy “root” outcome values 145
Table A–15 – CADF Outcome Taxonomy values expressed in relative and absolute URI forms 146
Table B–1 – CADF Timestamp data type properties 152

Foreword

The *Cloud Audit Data Federation - Data Format and Interface Definitions 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 DMTF acknowledges the following individuals for their contributions to this document:

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

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 and 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.

Document versioning scheme

This document will adhere to the versioning scheme defined in clause 6.3 of DSP0004.

Cloud auditing data federation use cases

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

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 having to factor in the cost of changing auditing processes and tools to adapt to different formats and interfaces.

Figure 1 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.

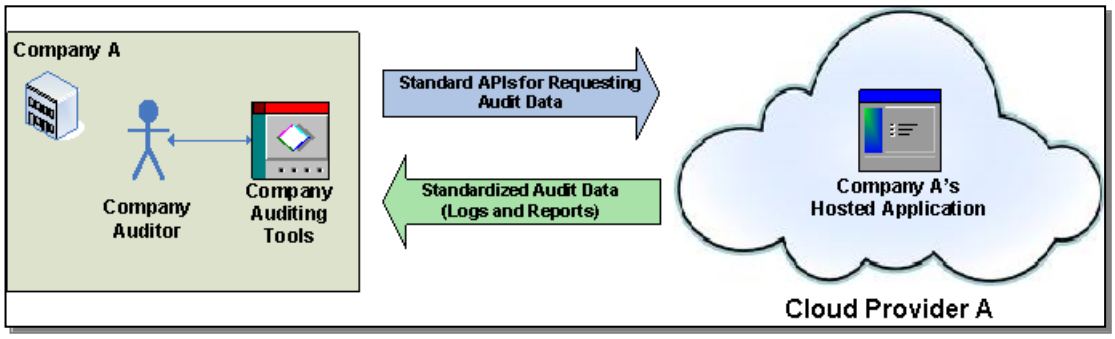


Figure 1 – Hosting application at a cloud provider; tools use open standards

Figure 2 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.

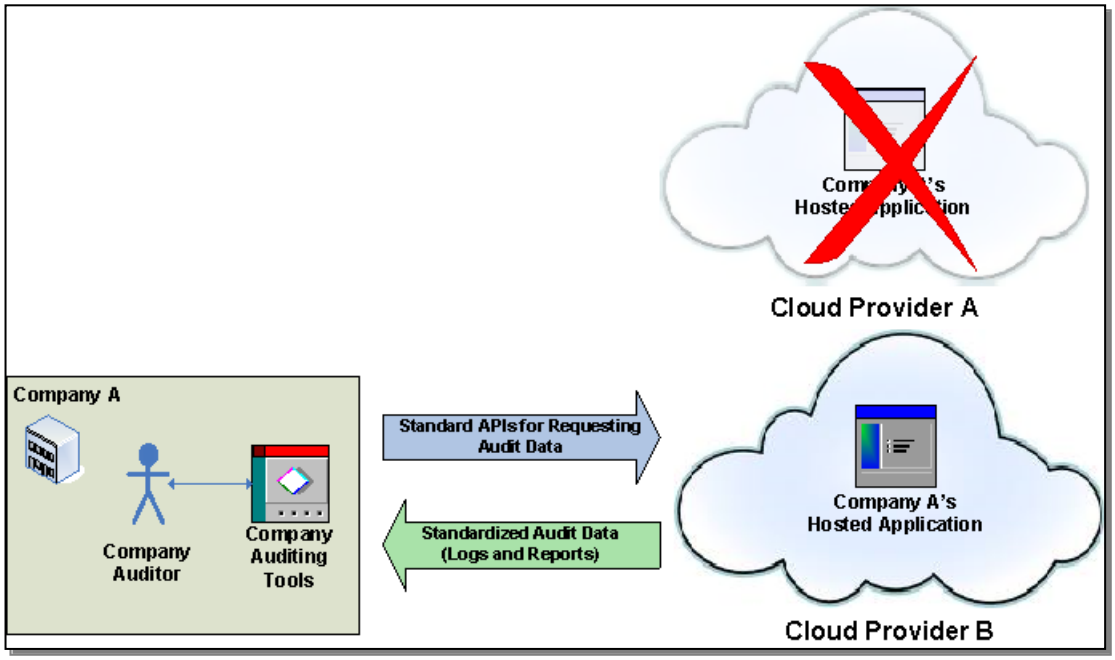


Figure 2 – Moving an application from Cloud Provider A to Provider B; tools unchanged

Auditing hybrid cloud applications

Because 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 by using the same criteria and logs and easily aggregate the data from these independent sources into a single audit trail.

Figure 3 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.

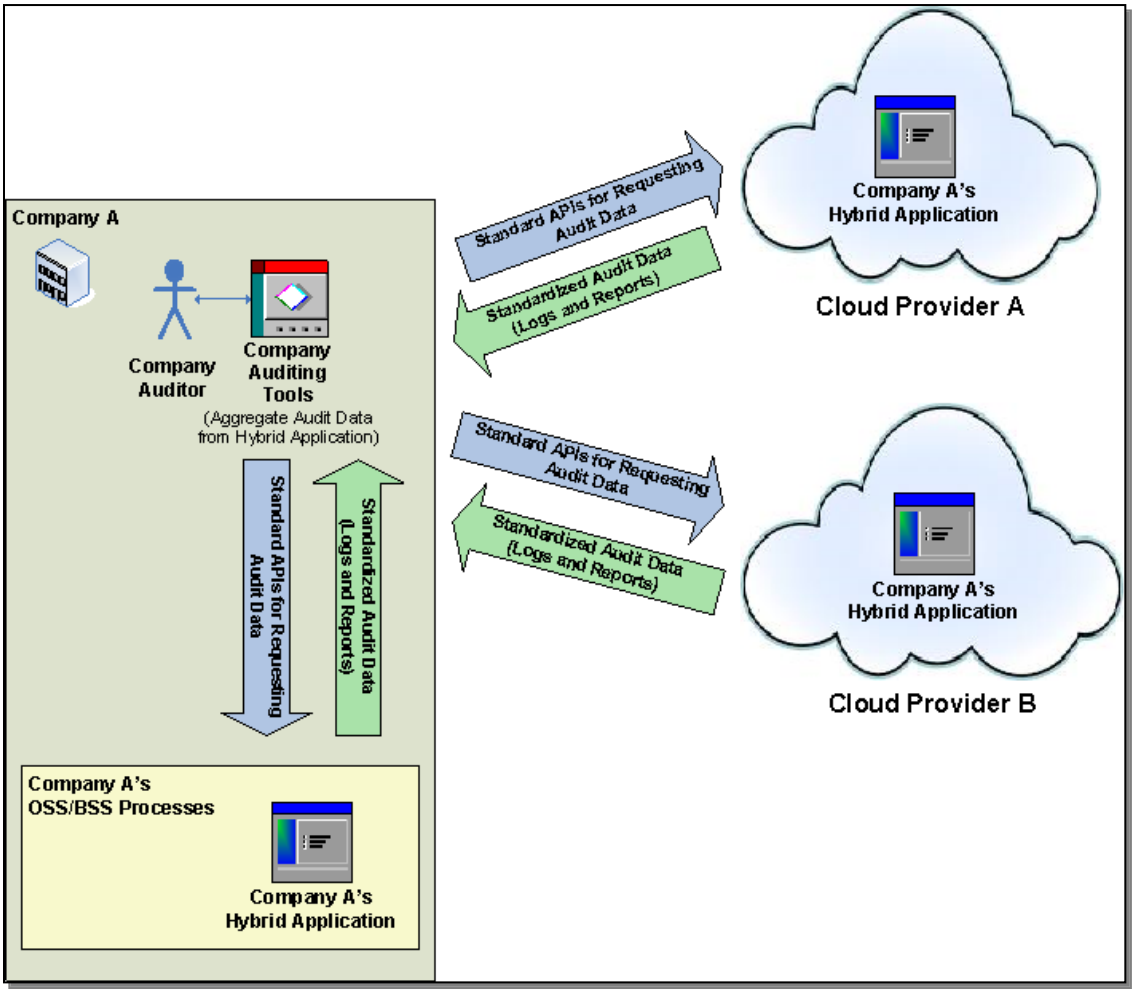


Figure 3 – Company aggregates audit data from hybrid cloud application across various deployments

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. To ensure that this goal is met, the working group has published DMTF document *Cloud Auditing Data Federation (CADF) Use Case White Paper* (DSP2028), 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.

Cloud Audit Data Federation - Data Format and Interface Definitions Specification

1 Scope and goals

1.1 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.

1.2 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 needs to be federated.

1.2.1 Audit data integrity and security

There is a strong need for ensuring the integrity and security of data that is used for auditing purposes. This need is 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.

In addition, this specification will design the data model such that it can easily be signed by various format specific mechanisms.

1.2.2 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 might 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 by consumers.

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.

1.2.3 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 clause 6.1 (Extensibility mechanisms) for approved extension methods.

1.2.3.1 Profiles

Profiles may be developed t at 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.

See clause 9 (CADF profiles) for more information.

1.2.4 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 or reference to use cases and examples to further demonstrate the need for or correct use of this specification’s data format and interface definitions.

1.3 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 clause 6.1 “Extensibility mechanisms” for approved extension methods.

Specific discussions of areas that are "Out of Scope" follow this clause.

1.3.1 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.

1.3.2 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) the audit event data, unauthorized disclosure of event contents, unauthorized submission of events, or unauthorized modification of events that are in transit or stored.

1.3.3 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.

1.3.4 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.

1.3.5 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.

1.3.6 Audit event alerting

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

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

DMTF DSP0004, *CIM Infrastructure Specification 2.6,*
http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

DMTF DSP0223, *Generic Operations 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide 1.1,*
http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

DMTF DSP4004, *DMTF Release Process 2.4,*
http://www.dmtf.org/sites/default/files/standards/documents/DSP4004_2.4.0.pdf

DMTF DSP4009, *Process for publishing XML schema, XML 6 documents and XSLT Stylesheets 1.0*,
http://www.dmtf.org/sites/default/files/standards/documents/DSP4009_1.0.0.pdf.

IANA-ccTL, 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

IETF RFC3986, T.Berners-Lee, et al., *Uniform Resource Identifiers (URI): Generic Syntax*, Jan. 2005,
http://www.ietf.org/rfc/rfc3986.txt

IETF RFC4627, D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*, July 2006,
http://www.ietf.org/rfc/rfc4627.txt

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 (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 Recommendation, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, November 2008,
http://www.w3.org/TR/REC-xml/

W3C Recommendation, *Namespaces in XML 1.0* (Third Edition), December 2009,
http://www.w3.org/TR/REC-xml-names/

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

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/

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/

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/

3 Terms and definitions

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.

This clause 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.

The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document. The following additional terms are used in this document.

Please note that this entire document is considered normative using the rules described above; however, critical requirements are frequently set apart in separate subsections for greater visibility.

3.1
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.”

3.2
Aggregation

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 that preserves and keep the original events accessible.

3.3
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]

3.4

Audit Event

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

3.5

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]

3.6

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]*

3.7

Authorization

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

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

3.8

Compliance Event

A compliance event is any event record that reports activity that is required to show compliance to a policy or requirement that 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.*

3.9

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.

3.10

Correlated Event

Any Event that is associated with some other set of Event s by some relationship, possibly causal. For example, a "throw" event may be associated with a corresponding "catch" event, with the implication that the same resource that was thrown was then caught.

3.11

Event Consumer

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

3.12

Event Provider

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

3.13

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]

3.14

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.

3.15

Event Action

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

3.16

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]**.**

3.17

Event Log

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

3.18

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.

3.19

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.

3.20

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.

3.21

Event Source

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.

3.22

Event Stream

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

Typically an event stream:

3. May be ordered by time.
4. May be bounded by a certain time interval or other criteria (content, space, source), or be open ended and unbounded.

3.23

Event Target

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

3.24

Filtering

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.

3.25

Geolocation

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.

3.26

Georouting

The geographical tracking of an event from its origin through the various resources that participated in the event or the handling an event.

3.27

Log

See definition for Event Log.

3.28

Query

See definition for Event Query.

3.29

Security Event

An 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].

3.30

Security Incident

A 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]

3.31

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.

3.32

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.

3.33

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.

3.34

Summarization

The consolidation of multiple related events into a single event, typically for storage or bandwidth optimization or for other analytical purposes.

3.35

Suppression

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

3.1 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:

3.36

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.

3.37

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).

3.38

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.

3.39

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 Interaction model

This specification's interface definitions are based upon a simple interaction model that describes the 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.

3.3 Document versioning scheme

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

4 CADF Event Model

The CADF Event Model applies semantics to the activities, resources, information and changes within a cloud provider's infrastructure and models these using the concept of an event. Some components of this model are essential (required) in creating a valid record of the event which is able to provide consumers (e.g., auditors, investigators, etc.) the fundamental information they need to perform analysis or assessments. Other components are optional or may be required depending on the type of event (i.e., conditional) and its additional contextual value to these consumers.

This section (4) establishes the semantics and rationale of the parts of a CADF Event Record that are conceptually most significant. Such parts are called here CADF Event Model components. These components will translate into a subset of the CADF Event Record's properties whose actual representation is the CADF Event data type (Section 6). Please note that additional CADF Event data type properties are defined in Section 6 and are not discussed as model components within this section.

This section explains the core concepts and components that comprise the CADF Event Model which enables a straightforward, prescriptive approach to creating CADF Event Records consistently regardless of cloud provider.

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, (Table 1) for the purposes of this specification, we define as follows:

Table 1 – Resource definition

Term	CADF Definition
RESOURCE	An entity or component that has the 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, which 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 (Table 2) to clearly distinguish between the different types of events:

Table 2 – 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 by using the CADF Event Schema. ***Note:*** *The schema of the CADF Event Record is designed so that other event record models, types or formats can be mapped to the CADF Event data type.*

4.2 Required model components

The names and semantics for all required CADF Event Model components are described below in Table 3:

Table 3 – Required CADF Event Model components

Model Component	CADF Definition
OBSERVER	The RESOURCE that generates the CADF Event Record based on its observation (directly or indirectly) of 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, according to the OBSERVER. ***Note:*** *a TARGET (in the CADF Event Model) can represent a plurality of target resources.*
OUTCOME	The result or status of the ACTION against the TARGET, according to the OBSERVER.

4.2.1 Basic conceptual event model

Conceptually, a single RESOURCE called the OBSERVER is responsible for observing the Actual Event and creating the (initial) CADF Event Record based upon its perspective and purpose. The OBSERVER does its best to identify and classify all other required model components (e.g., INITIATOR, TARGET, ACTION, etc.) along with any relevant data.

The conceptual diagram in Figure 4 shows basic components of the CADF Event Model and their interactions:

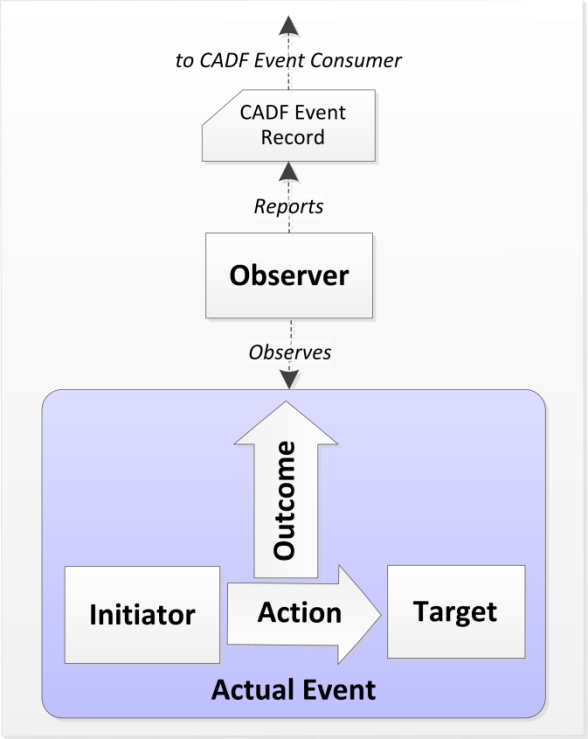


Figure 4 – CADF Event Model: Basic components

4.2.2 The OBSERVER perspective

Many software systems and platforms are constructed as layers which ACTIONS pass through in order to affect some final TARGET resource. It is assumed that OBSERVERS reside in different layers and each produces a CADF Event Record that can be correlated to produce an end-to-end log of all actions as they pass through the layers of a system. This means that each OBSERVER should only report the INITIATOR, TARGET and other data as it "sees" and can classify them from within their own layer since it can rely on other OBSERVERS to do the same.

For example, a user might call an API from a remote system to store some data at a cloud provider. This API request (along with the data) might pass through many layers of a cloud platform before affecting an actual hardware resource (e.g., a block storage device). An OBSERVER within an IaaS (middle) layer may see the authorized "storage" request, but have no direct knowledge of the user that initialed the request at a higher layer. Likewise, it may not know the eventual TARGET is a physical storage device but passes the request to a storage service. Therefore, that OBSERVER should not attempt to claim the INITIATOR was a user nor that the TARGET was some block storage device. Instead, it should only record (identify and classify) the immediate resources that it received or sends the API request from and to (i.e. its apparent INITIATOR and TARGET resources).

Of course, each OBSERVER should preserve and include in the CADF Event Record any relevant data received from the INITIATOR that is significant in fulfilling the API request by the final TARGET and may be useful for an audit.

4.2.3 Notes

In some cases, the OBSERVER, INITIATOR, and TARGET could reference the same resource. The precise interpretation of these components, therefore, will depend somewhat on the type of event being recorded, and the specific activity and resources involved. Please see the mapping examples later in this chapter (see 4.7 “Mapping typical events to CADF Event Model”) which describes such use cases.

4.3 Conditional model components

As previously mentioned, CADF Event Records may contain different information depending on the perspective and of the OBSERVER and its audit purpose. This clause introduces additional CADF Event Model components that may optionally be added or even be required for certain event types this specification defines. These event types and treatment are described later in this chapter within the section titled “Types of CADF Events”.

4.3.1 MEASUREMENT

Measurements are an optional component of the CADF Event Type, but are essential and required for any CADF Event Record that is classified as a **monitor** type event (see section “Types of CADF Events”).

Table 4 – Conditional MEASUREMENT component definition

Model Component	CADF Definition
MEASUREMENT	A component 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).

The MEASUREMENT component is embodied by the CADF Measurement data type which is included in the CADF Event data type. The MEASUREMENT component also includes information (or a reference) to the metric used to record the MEASUREMENT (e.g., unit, calculation method, etc.) which is represented by the CADF Metric data type.

4.3.2 REASON

Reason data is an optional component of the CADF Event Type, but is essential for any CADF Event Record that is classified as a **control** event (see section “Types of CADF Events”).

Table 5 – Conditional REASON component definition

Model Component	CADF Definition
REASON	A component that contains a means to provide additional details and further classify the top-level OUTCOME of the ACTION included in a CADF Event Record.

The REASON component is embodied by the CADF Reason data type which is included in the CADF Event data type.

4.3.3 Basic conceptual event model with optional components

The following diagram shows the CADF Event Model with conditional components added:

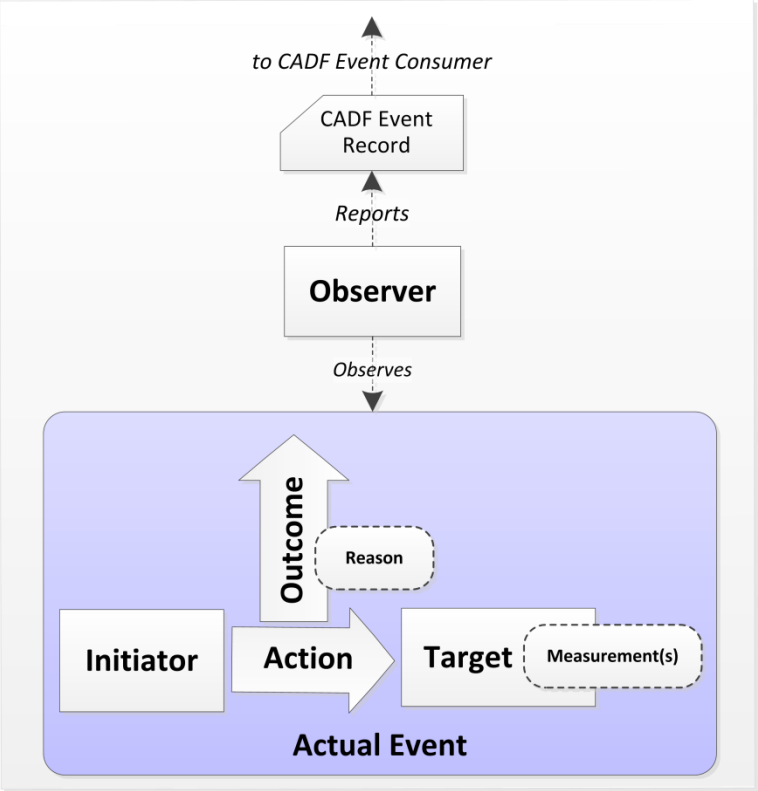


Figure 5 – CADF Event Model: Basic and conditional model components

4.4 Optional components

4.4.1 Reporters and the 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 resources may assist in constructing and surfacing the CADF Event Record before it is presented to the end consumer. In the CADF Event Model we call each of these resources a REPORTER which can each be described, along with their role, within the CADF Event Record as part of a sequential chain (sequence) of REPORTER components called a REPORTERCHAIN.

The following table describes the REPORTER and REPORTERCHAIN as optional components of the CADF Event Model (Table 6):

Table 6 – REPORTERCHAIN definition

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

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

918 #### 4.4.1.1 CADF Reporter roles

919 As described above, many REPORTER components may assist in constructing and surfacing the CADF Event
920 Record before it is presented to the end consumer. In this specification, we will describe requirements based upon
921 REPORTER roles which we define in Table 7.

922 This specification defines the following basic CADF Reporter roles:

923 **Table 7 – CADF: Reporter roles**

Reporter Role	CADF Definition
observer	A REPORTER that fulfills the role of OBSERVER.
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).

924

925 #### 4.4.1.2 Example

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

929 In Figure 6, a flow demonstrating the construction of a CADF Event Record by several “reporters” is shown from left
930 to right:

931 • Reporter A is the OBSERVER of the Actual Event and generates the CADF Event Record from its
932 perspective by recording the required INITIATOR, TARGET, ACTION, and OUTCOME entities and
933 properties. Reporter A then adds itself as the first entry in the REPORTERCHAIN of the CADF Event Record
934 (an optional entry) with REPORTER “`role`” property set to ‘**observer**’ and passes the record to Reporter B.

935 • Reporter B receives the CADF Event Record and modifies the record in order to augment the event's
936 INITIATOR data with more detailed user account information. Reporter B then adds itself as a ‘**modifier**’ (a
937 CADF Reporter Role) to the event record's REPORTERCHAIN after the entry for Reporter A and passes the
938 CADF Event Record to Reporter C.

939 • Reporter C receives the CADF Event Record from Reporter B. Reporter C adds itself as the
940 REPORTERCHAIN after Reporter B's entry indicating it simply acted as a ‘**relay**’ (another CADF Reporter
941 Role) and performed no other modifications to the CADF Event Record. Reporter C passes the CADF Event
942 Record to Reporter D.

943 • Reporter D receives the CADF Event Record from Reporter C. Reporter D "modifies" the event record to add
944 CADF resource categorization information, and then adds itself as the last entry in the REPORTERCHAIN
945 (as the second ‘**modifier**’ CADF Reporter Role entry) prior to presenting the CADF Event Record to the end
946 CADF Event Consumer.

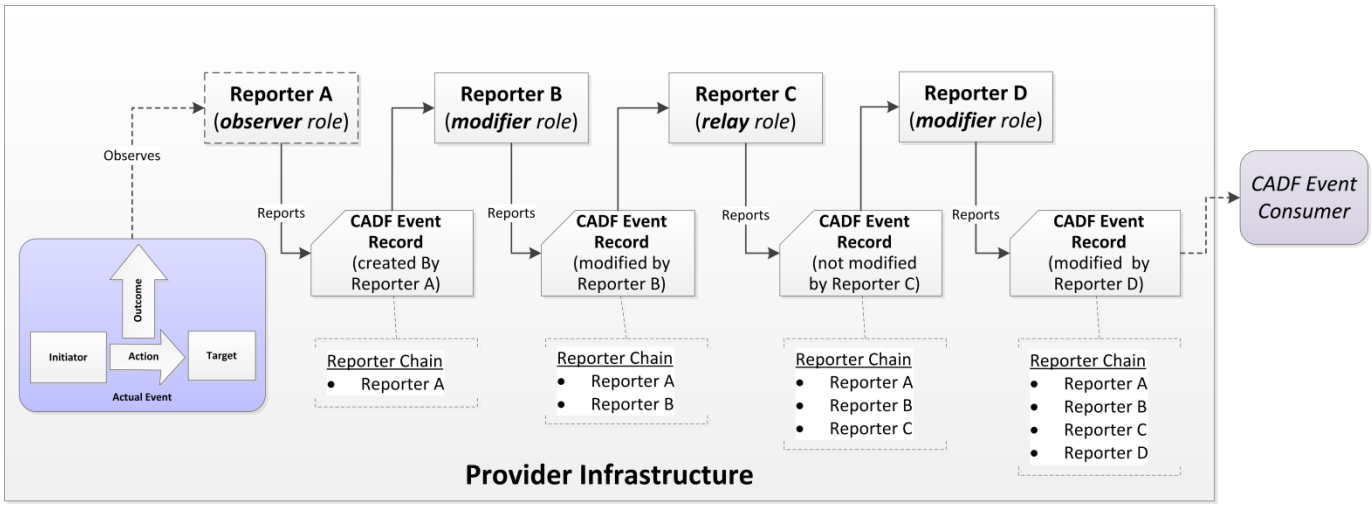


Figure 6 – Example of REPORTERCHAIN construction

4.4.1.3 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, a CADF Event Record could be logged, and later on could be processed again by a new Reporter, thus extending its REPORTERCHAIN.

4.5 Types of CADF Events

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 the concept of an "event type" which affects what data is required as part of the CADF Event Model and included within the CADF Event Record (see the "`eventType`" property of the CADF Event data type).

Within this specification, we will reference the concept of an "event type" using the keyword (term) "EventType" which is defined below.

Table 8 – EventType definition

Term	CADF Definition
EventType	A conceptual top-level classification of the CADF Event Record and its data that is intended to communicate additional or more specific data and requirements. ***Note:*** *Valid values for EventType would appear in the "eventType" property within the CADF Event data type.*

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 types of CADF Events defined in this specification (see Table 9) or one of its profiles (by extension).

4.5.1 Valid EventType values

The RESOURCE that generates the CADF Event Record (see the OBSERVER model component defined below) declares the purpose for creating the audit record, reflecting its distinct perspective, by setting the "`eventType`" property in the CADF Event data type using one of the valid values from the table below.

This specification defines the following valid values for use in the CADF Event data type's "`eventType`" property:

Table 9 – Valid EventType values

EventType Value	CADF Definition
monitor	Characterizes events that provide information about the status of a resource or of its attributes or properties, Such events typically report on measurements or periodic probes on cloud resources, and may produce aggregate data such as statistical or summary metrics..
activity	Characterizes events that provide information about actions having occurred or intended to occur, and initiated by some resource or done against some resource, Such events typically report on regular operations of a Cloud infrastructure or services.
control	Characterizes events that reflect on or provide information about the application of a policy or business rule, or more generally express the outcome of a decision making process. Such events typically report on how these policies or rules manifest in concrete situations such as attempted resource access, evaluation of resource states, notifications, prioritization of tasks, or other automated administrative action.

4.5.2 EventType Requirements

- Although it is envisioned that profiles of this specification could define additional EventType values, these profiles SHOULD NOT override ore redefine the basic semantic meaning assigned to core event fields and event types defined in this specification.
- The creator or producer of a CADF Event Record SHOULD, in general, assume that there is no guarantee that the record consumer has access to any extension profile, and where possible therefore should attempt to map data to entities, properties and values defined in this specification.

Selecting an EventType value

EventType values are more reflective of the general purpose of an event rather than of a precise, unambiguous event category. The same Actual Event could often be recorded or could produce more than one CADF Event of different types – depending on the general interpretation made by one (or more) event OBSERVER(s).

For example, a monitoring device will generally produce events of type "**monitor**". However if the intent is to report on the activity of the device itself as a resource acting on another resource, then an event of type "**activity**" could be generated **as well.** Similarly, raising an alarm about the state of a resource can be seen as a "**control**" event due to the policy rule decision on the critical aspect of this state, yet also involves simple monitoring of this resource (i.e. the collection of state data can be seen as a "**monitor**" event).

Please note, however, that a '**control**' event describes **only** the application of the policy on target resources such as a network connection that is denied by a firewall policy. It may not describe important details about the underlying activity that caused the policy to be evaluated in the first place: these details may be made available in other CADF Event Records (as an '**activity**' type event) and associated with the control event as correlated events.

4.6 Refinement of Event semantics based upon the selected EventType value

Depending on the event type, the generic components of an event (see table 3 in 4.2) will have a refined definition, although still consistent with their general meaning as stated in 4.2. Some of these components may be optional or redundant; others will be preeminent, depending on the event type.

The following tables show how the interpretation of some event components may be extended for each type.

***Note**: some secondary event components not defined in 4.2 but defined in the detailed event model may be involved and are listed below for clarity; their names appear in lower-case characters.*

Refined semantics of Event components for the **monitor** type:

Table 10 – Event component semantics for "monitor" type events

Event Component	Prescription Level	CADF Refined Definition
INITIATOR	Mandatory	The RESOURCE that initiated the "monitoring" action. It must be the same resource as the OBSERVER component.
ACTION	Mandatory	The monitoring action itself. Only the "monitor" value in the ACTION taxonomy applies (see Annex A2).
TARGET	Mandatory	The RESOURCE being monitored.
OUTCOME	Mandatory	An assessment about the monitoring operation itself. All values of the OUTCOME taxonomy apply (Annex A3). *For example, An outcome value of "success" means that the resource data has been successfully collected, "failure" means the data could not be properly reported (failed monitoring).*
MEASUREMENT	Mandatory	The measure resulting from the monitoring.

Refined semantics of Event components for the **activity** type:

Table 11 – Event component semantics for "activity" type events

Event Component	Prescription Level	CADF Refined Definition
INITIATOR	Mandatory	The RESOURCE that initiated the "activity" (the resource author of the ACTION).
ACTION	Mandatory	The operation or action identifying the "activity". All values in the ACTION taxonomy (see Annex A2) are applicable.
TARGET	Mandatory	The RESOURCE that is the target of this "activity".
OUTCOME	Mandatory	The result or status of the "activity", i.e. expressing an assessment about the execution of this activity. All values of the OUTCOME taxonomy apply (Annex A3)
MEASUREMENT	Optional	Some measure associated with the execution of this activity (e.g. for a request action, a response time).

Refined semantics of Event components for the **control** type:

Table 12 – Event component semantics for "control" type events

Event Component	Prescription Level	CADF Refined Definition
INITIATOR	Mandatory	The RESOURCE that performed the "control" decision making or applied the related policy.
ACTION	Mandatory	The decision-making action itself. Only the "evaluate", "allow", "deny" and "notify" values in the ACTION taxonomy apply (see Annex A2).
TARGET	Mandatory	The RESOURCE being the main object of the decision or policy, if any.

OUTCOME	Mandatory	A general assessment about the decision making process itself. Only some values of the OUTCOME taxonomy apply (Annex A3): • "***success***" means that the decision making was successfully completed • "***failure***" means that a decision outcome could not be produced for some reason. • "***pending***" means that the decision process is still in progress, or waiting for more input. However, this taxonomy could be extended with specific values as needed.
REASON	Mandatory	Provides a rationale for why the particular control action was taken, including a reference to the policy that drove the decision.
MEASUREMENT	Optional	Some measure on which the decision outcome was based (e.g. an average response time for a target server, leading to an alarm if beyond a threshold.).

4.6.1 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.

4.7 Mapping typical events to CADF Event Model

This clause 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.

4.7.1 General approach

The following table shows the CADF Event model components and how to obtain the correct classification and type values:

1022 **Table 13 – General mapping approach using the CADF Event Model**

CADF EventType and Model Components	Value selection methodology
EventType	Select a valid EventType value that best describes the primary (audit) purpose the OBSERVER has in reporting the Actual Event (and generating the CADF Event Record). • e.g., "activity" (default), "control" or "monitor"
OBSERVER	Select a classification value from the CADF Resource Taxonomy that best describes the type resource that is observing the actual event and is generating the CADF Event record.
INITIATOR	Select a classification value from the CADF Resource Taxonomy that best describes the type of resource that initiated the actual event from the point of view of the OBSERVER.
ACTION	Select a classification value from the CADF Action Taxonomy that best describes the action the INITIATOR of the actual event is attempting at the time the OBSERVER is generating the CADF Event Record. • e.g., "create", "update", "deploy", "notify", etc.
TARGET	Select a classification value from the CADF Resource Taxonomy that best describes the type of resource that is the target of the actual event's action from the point of view of the OBSERVER.
OUTCOME	Select a classification value from the CADF Outcome Taxonomy that best describes the actions outcome (against the TARGET resource) at the time the OBSERVER is generating the CADF Event Record. • e.g., "success", "failure", "pending", etc.
MEASUREMENT	If the EventType value is "monitor", then this component must be included with a valid Measurement type and associated property values, otherwise (for other EventType values it is optional.
REASON	If the EventType value is "control", then this component must be included with a valid Reason type and associated property values, otherwise (for other EventType values it is optional.

1023

1024 ### 4.7.2 Use case 1: Auditing access to a controlled resource

1025 A cloud provider has a software component that manages identity and access control that we will call an "identity
1026 management service". This service is required, by the provider's security policy, to log all user activities including
1027 "logon" attempts against any servers within the provider's infrastructure.

1028 This example attempts to highlight the following mapping or classification decisions:

1029 • The EventType value is set to "**activity**" since the OBSERVER's purpose is to report on a security activity.

1030 #### 4.7.2.1 Mapping to the CADF Event Model

1031 The following table shows a mapping of the significant actors and elements described in this use case to the
1032 conceptual CADF Event Model:

1033 **Table 14 – Use case 1: Mapping of actors and elements to the CADF Event Model**

CADF EventType and Model Components	Selected classification or type value	Rationale
EventType	activity	Selected because OBSERVER is required to report any user security activity (e.g., a “logon”) as part of its proof that the provider is adhering to its company’s “security” policy.
OBSERVER	service/security/identity	This value from the CADF Resource Taxonomy most closely describes an “Identity Manager Service”.
INITIATOR	data/security/account/user	This value from the CADF Resource Taxonomy most closely describes a “user” attempting to “logon” to a “server” perhaps from some application service or client).
ACTION	authenticate/logon	This value from the CADF Action Taxonomy most closely describes a user “logon” action.
TARGET	compute/node	This value from the CADF Resource Taxonomy most closely describes a target “server” that the “user” is attempting to “logon” to.
OUTCOME	Any valid CADF Outcome Taxonomy value	The OBSERVER would select a value from the CADF Outcome Taxonomy that best describes the result of the action it observed. • e.g., success, failure, pending, etc.
MEASUREMENT	N/A	A MEASUREMENT component is not required for “activity” type events.
REASON	N/A	A REASON component is not required for “activity” type events.

1034 The following figure shows the same mapping applied to the conceptual CADF Event Model:

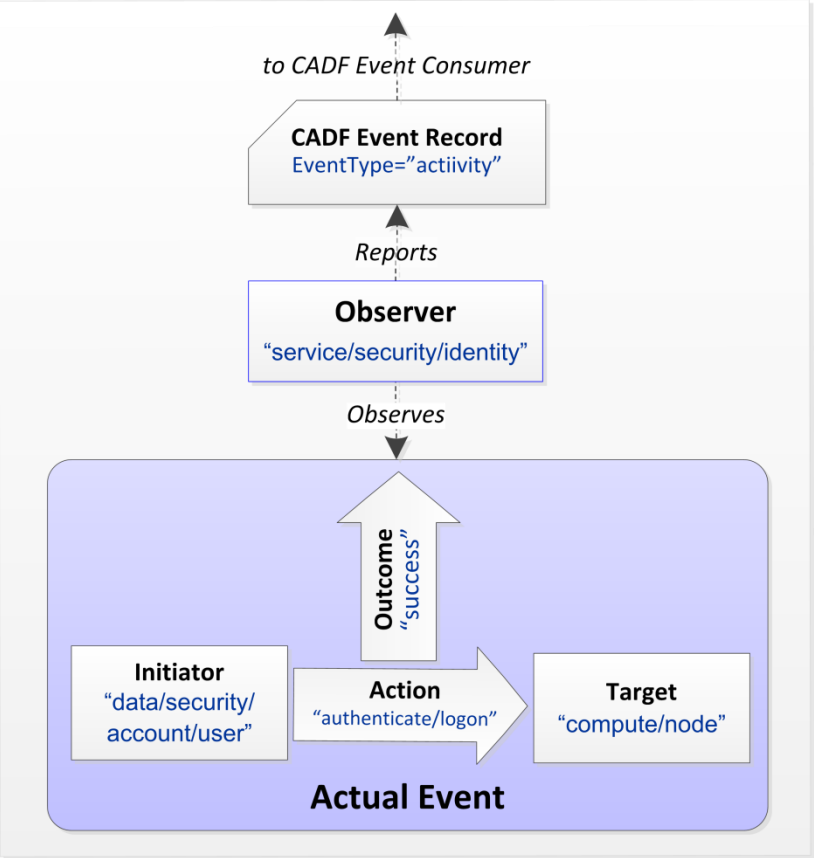


Figure 7 – Use case 1: Conceptual mapping

4.7.3 Use case 2: Periodic monitoring resource status

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 by using the CADF Event Record format.

This example attempts to highlight the following mapping or classification decisions:

- 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.

4.7.3.1 Mapping to the 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:

1048 **Table 15 – Use case 2: Mapping of actors and elements to the CADF Event Model**

CADF EventType and Model Components	Selected classification or type value	Rationale
EventType	monitor	Selected because OBSERVER is required to monitor a server's CPU utilization.
OBSERVER	service/oss/monitoring	This value from the CADF Resource Taxonomy most closely describes a "software monitoring agent".
INITIATOR	service/oss/monitoring	The OBSERVER is also the INITIATOR of this monitoring event.
ACTION	monitor	This value from the CADF Action Taxonomy (or a direct extension of this value) SHALL be used when the EventType value is "monitor".
TARGET	compute/cpu	This value from the CADF Resource Taxonomy most closely describes a server's "cpu".
OUTCOME	success	The OBSERVER successfully obtained and reported a CPU utilization measurement and therefore selected the "success" value from the CADF Outcome Taxonomy.
MEASUREMENT	80%	The MEASUREMENT component is required and the observed 80% CPU utilization is provided as the value.
REASON	N/A	A REASON component is not required for "monitor" type events.

The following figure shows the same mapping applied to the conceptual CADF Event Model:

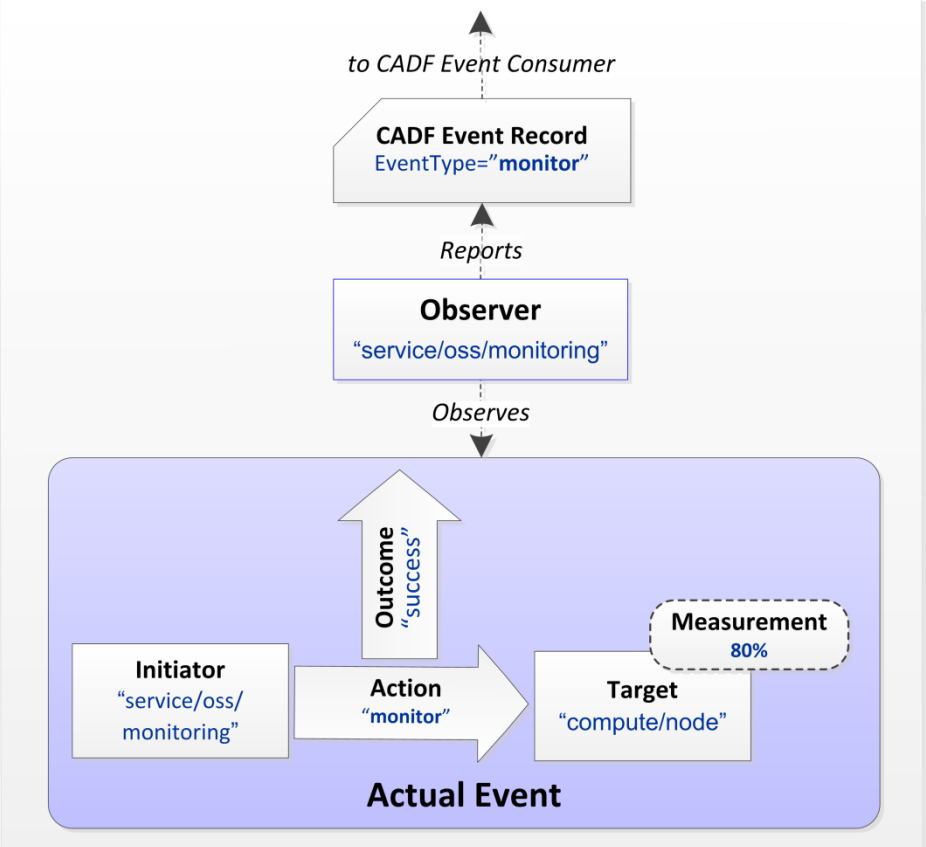


Figure 8 – Use case 2: Conceptual mapping

4.7.4 Use case 3: Aggregation of resource status into an audit event

In this use case, a cloud provider has a "monitoring server" (i.e. a dedicated compute node on the cloud network) 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 that are 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 by using the CADF Event Record format.

This example attempts to highlight the following mapping or classification decisions:

- The EventType value is set to **monitor**.
- The OBSERVER's purpose is to monitor multiple servers' CPU utilization and provide summary events.

4.7.4.1 Mapping to the 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:

1063 **Table 16 – Use case 3: Mapping of actors and elements to the CADF Event Model**

CADF EventType and Model Components	Selected classification or type value	Rationale
EventType	monitor	Selected because OBSERVER is required to monitor a server's CPU utilization.
OBSERVER	compute/node	This value from the CADF Resource Taxonomy most closely describes a "server".
INITIATOR	compute/node	The OBSERVER is also the INITIATOR of this monitoring event.
ACTION	monitor	This value from the CADF Action Taxonomy (or a direct extension of this value) SHALL be used when the EventType value is "monitor".
TARGET	compute/cpu	This value from the CADF Resource Taxonomy most closely describes a set of CPUs from multiple servers.
OUTCOME	success	The OBSERVER successfully obtained and reported a CPU utilization measurement and therefore selected the "success" value from the CADF Outcome Taxonomy.
MEASUREMENT	70%	The MEASUREMENT component is required and the observed 70% CPU utilization percentage (average) is provided as the value.
REASON	N/A	A REASON component is not required for "monitor" type events.

1064 The following figure shows the same mapping applied to the conceptual CADF Event Model:

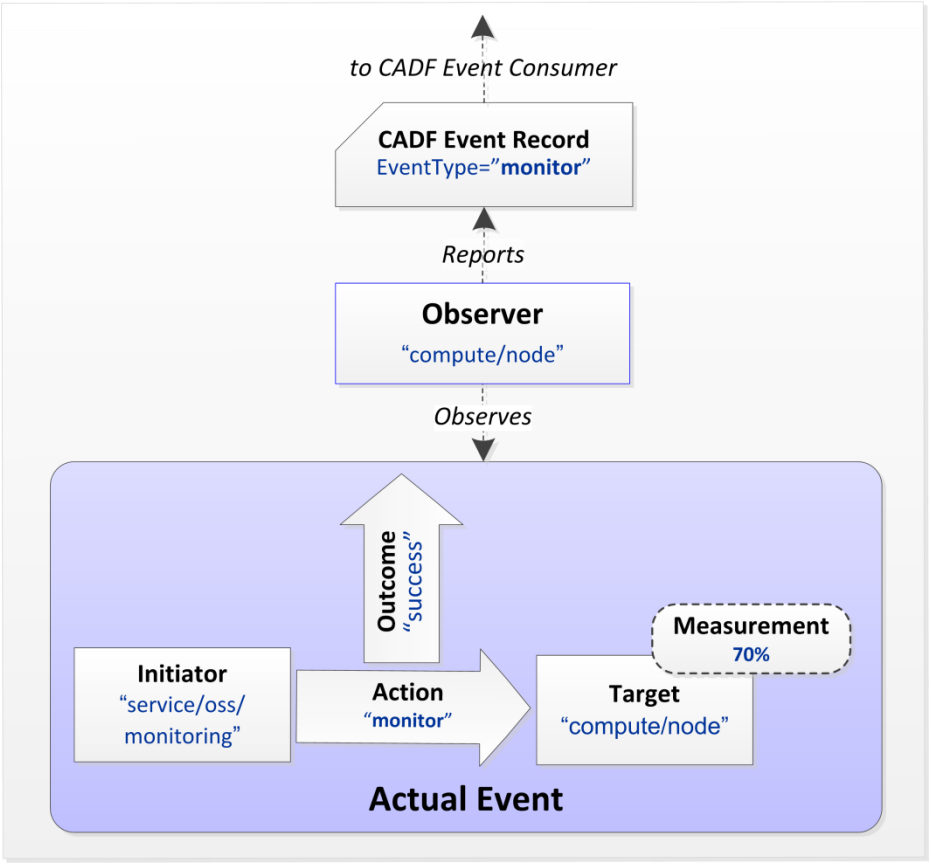


Figure 9 – Use case 3: Conceptual mapping

4.7.5 Use case 4: Auditing compliance of resource monitors

In this use case, 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) by using the CADF Event Record format.

This example attempts to highlight the following mapping or classification decisions:

- This event record represents an alternative view of the same ACTUAL EVENT as described in Example 2 (Periodic monitoring resource status), but is observed from a different perspective.
- The EventType is set to **activity**.
- The OBSERVER's purpose is to report on the "read" ACTION for compliance reasons.
- The MEASUREMENT represents an optional component that could be included in the event record.

4.7.5.1 Mapping to the 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 (Table 17):

1081 **Table 17 – Use case 4: Mapping of actors and elements to the CADF Event Model**

CADF EventType and Model Components	Selected classification or type value	Rationale
EventType	activity	Selected because OBSERVER is reporting on the low-level "read" activity it is performing against a server's CPU.
OBSERVER	service/oss/monitoring	This value from the CADF Resource Taxonomy most closely describes a "resource monitor".
INITIATOR	service/oss/monitoring	The OBSERVER is also the INITIATOR of this monitoring event.
ACTION	read	This value from the CADF Action Taxonomy reflects an audit of a "read" action against the TARGET resource.
TARGET	compute/cpu	This value from the CADF Resource Taxonomy most closely describes a set of CPUs from multiple servers.
OUTCOME	success	The INITIATOR successfully "read" the CPU utilization from the target server and therefore selected the "success" value from the CADF Outcome Taxonomy.
MEASUREMENT	80%	The MEASUREMENT component is OPTIONAL since this is an "activity" EventType. However, since the "read" activity obtained a CPU utilization measurement, the OBSERVER chose to include this on the CADF Event Record.
REASON	N/A	A REASON component is not required for "activity" type events.

1082 The following figure shows the same mapping applied to the conceptual CADF Event Model:

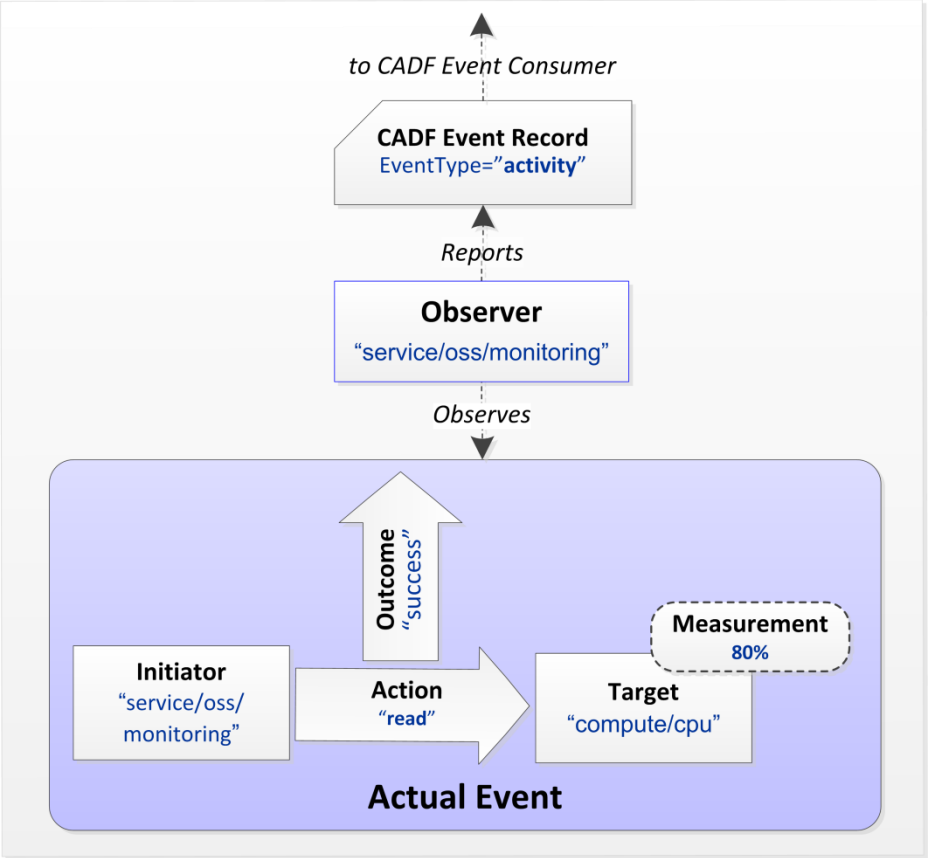


Figure 10 – Use case 4: Conceptual mapping

4.7.6 Use case 5: Auditing controlled resource accesses

In this use case, a user attempts to perform an unauthorized access of a document (a controlled resource) residing in a cloud provider's storage infrastructure. The failed access request was made using an HTTP interface exported as part of the provider's cloud storage service which is designed to return IANA HTTP status codes in the response message. In this example, a "401" "`reasonCode`" value, which corresponds to "Unauthorized" is returned when the provider's authorization system determines the user does not have access to the document they requested.

This example attempts to highlight the following mapping or classification decisions:

- The event record represents a specific view of an ACTUAL EVENT as observed from a resource that is reporting on an access control decision from its perspective for compliance audits.
- The EventType is set to **control**.
- The OBSERVER's purpose is to report on the "`deny`" ACTION for compliance reasons (in this case, the denial of access to the controlled resource).
    - Note: that other OBSERVERS of the same ACTUAL EVENT may generate other CADF Event Records that describe the activity of reading the document (i.e., an "`eventType`" value of "`activity`" and an ACTION value of "`read`"). CADF Event Records that represent different perspectives (or observations) of the same ACTUAL event should be correlatable by consumers when examining the set of event records produced by the event record provider.
- The REASON represents a mandatory component for control-type events that would be included in this type of event record.

4.7.6.1 Mapping to the 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 (Table 18):

Table 18 – Use case 5: Mapping of actors and elements to the CADF Event Model

CADF EventType and Model Components	Selected classification or type value	Rationale
EventType	control	Selected because OBSERVER is reporting on the control action made by a security authorization service.
OBSERVER	service/security/authorization	This value from the CADF Resource Taxonomy most closely describes a service that is observing the authorization decision on the TARGET resource. In this case, it is the same service that is the INITIATOR of the "denial" ACTION.
INITIATOR	service/security/authorization	The INITIATOR is the authorization service, as defined in the security subtree of the CADF Resource Taxonomy.
ACTION	deny	This value from the CADF Action Taxonomy reflects an audit of a "deny" action against the TARGET resource. That is, the authorization service is actively denying a user access to a controlled document.
TARGET	data/file	This value from the CADF Resource Taxonomy most describes a generic file-based document that the user is trying to access.
OUTCOME	success	The INITIATOR successfully "denied" access to the controlled TARGET document. Therefore the "success" value was selected from the CADF Outcome Taxonomy.
MEASUREMENT	N/A	The MEASUREMENT component is OPTIONAL since this is a "control" EventType.
REASON	401	A REASON component is required for "control" type events. In this case, an IANA code "401", meaning "Unauthorized", appears in the value of the `reasonCode` property. The "`reasonType`" property would be set to the IANA standard's registry "http://www.iana.org/assignments/http-status-codes/http-status-codes.xml".

The following figure shows the same mapping applied to the conceptual CADF Event Model:

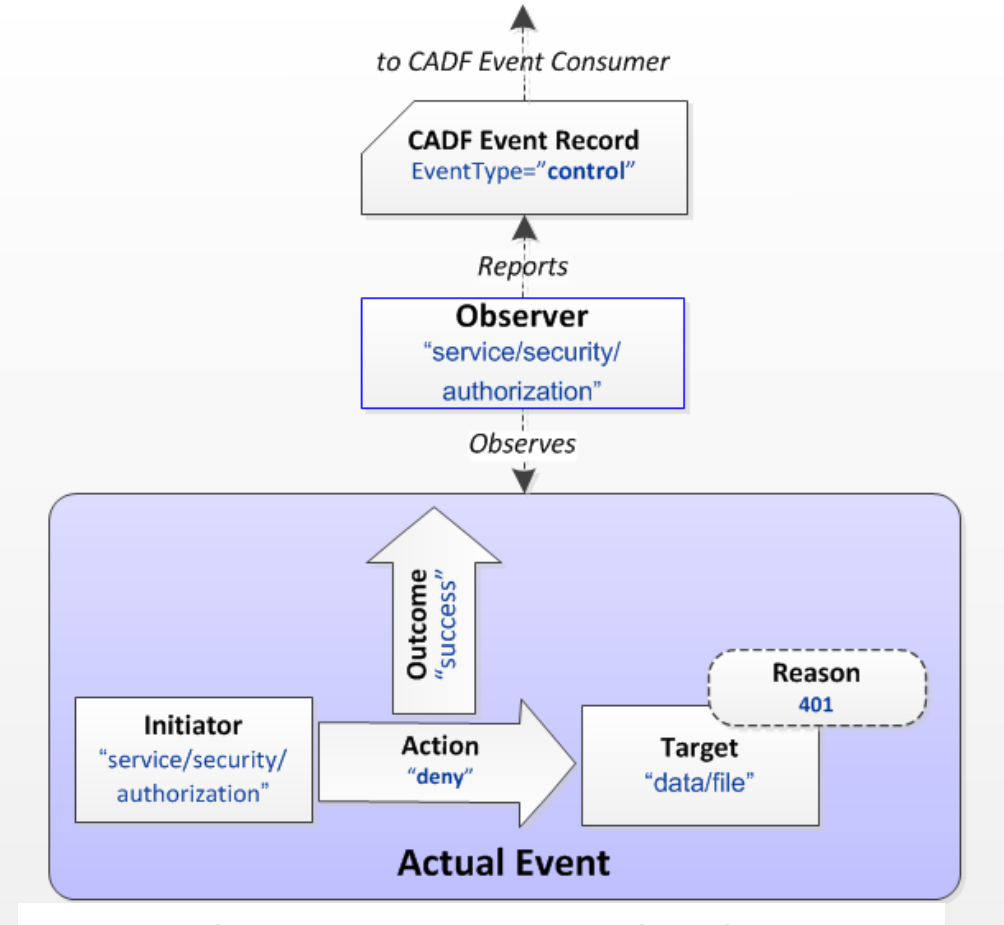


Figure 11 – Use case 5: Conceptual mapping

5 Data model and schema conventions

5.1 Namespace URIs and alias conventions

CADF data is designed to be federated and merged from various sources, as well as extended via profiles. Therefore, this specification must produce data (e.g., events, logs and reports) that provides clear identification of each domain (schema) that may have defined a data entity, type, property or property value to CADF data consumers. This consideration includes the definition of values that are used to uniquely identify resources, provide classifications, reference CADF and external schemas, etc.

5.1.1 Namespace URIs

Namespace URIs are used throughout this specification to uniquely identify the CADF specification domain when defining CADF Event Model components, CADF Entities, CADF properties, CADF classification values and other values.

5.1.1.1 Requirements

- Any Namespace URI 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 Namespace URI that is defined in this specification.
- CADF data consumers SHALL NOT make assumptions about the layout or network accessibility of the URIs or the structures of any URI used in this specification, extensions, or profiles.
    - For example, just because a URI uses the "http" protocol scheme prefix to identify some data schema (e.g., "`http://mystandard.org/schema`") or a server resource (e.g., "`http://mycompany.com/myserver`"), it does not imply that these can actually be dereferenced as URLs.

5.1.2 Namespace aliases

The use of Namespace URIs within events, logs and reports achieves clear identification of data, it can also lead to repetition, increased data sizes and reduced readability. 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 this specification.

5.1.2.1 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 Namespace alias that 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

Table 19 lists the namespaces (i.e., URIs) and namespace aliases 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.

Table 19 – Namespaces

Alias	Namespace	Specification
cadf	http://schemas.dmtf.org/cloud/audit/1.0/	The CADF Namespace and CADF Namespace alias used to represent this specification (by version).
xs	http://www.w3.org/2001/XMLSchema	XML Schema

5.2.1 Requirements

- The CADF Namespace and Namespace alias SHALL be reserved for use by this specification.
- The CADF 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/
```

- The CADF Namespace alias for this specification's schema SHALL be the value "`cadf`" (i.e., only the lowercased characters within the quotes):

```
cadf
```

- The CADF Namespace SHALL be used as the target namespace for any schema (e.g., XML, JSON, etc.) that represents the definitions and requirements of this specification.
- The CADF Namespace alias "`cadf`" SHOULD be used to represent the CADF Namespace as a prefix wherever possible. For example:

```
cadf:<data entity, type, property or value>
```

- Profiles of this specification MAY define additional namespaces and aliases to reference themselves within CADF documents and schema.

5.2.2 XML usage example

The following example shows the proper use of this specification's namespace within an XML schema definition (XSD) document which would declare CADF schema elements and attributes.

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

Then following example shows how the CADF schema would be referenced within an XML instance document that references the CADF XML Schema Definition (XSD):

```
<?xml version="1.0" encoding="UTF-8"?>
   <cadf:log
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="..."
      xmlns:cadf="http://schemas.dmtf.org/cloud/audit/1.0/">
      ...
   </cadf:log>
</xml>
```

***Note**: All CADF elements are qualified properly within the XML document instance.*

5.2.3 JSON usage example

As of the authoring of this specification, there is no standardized way to express namespaces in JSON documents. This specification provides a property named "typeURI" for all top-level CADF Entities (i.e., CADF Event, Log and Report) which can be used by interpreters of JSON or other data formats (e.g., YAML, etc.) to recognize a set of CADF data:

```
{
   "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
   ...
},
```

The above example would indicate all the other properties and values within the same structure are to be interpreted as a CADF Event type as defined by the CADF version 1.0 specification (schema).

5.2.3.1 Notes

The recently published W3C JSON-LD 1.0 candidate recommendation is one potential standard that shows promise for declaring identifiers and types (i.e., a data schema) for JSON documents.

The following example is non-normative; however, it shows how the CADF schema's namespace could be declared using JSON-LD 1.0 to establish a target namespace for all properties in the JSON data it is associated with (unless otherwise aliased or prefixed (using a full Internationalized Resource Identifiers or IRIs) :

```
"@context": {
   "@vocab": "http://schemas.dmtf.org/cloud/audit/1.0/",
   ...
},
```

The above JSON-LD declaration could be used within the context of a document to setup the "base" vocabulary for the CADF schema (i.e., the CADF namespace) prior to introducing a CADF Entity (e.g. a CADF Event, Log or Report). The context could also be used to create the "`cadf`" schema namespace alias:

```
"@context": {
   ...
   "cadf": "http://schemas.dmtf.org/cloud/audit/1.0/",
   ...
},
```

5.3 Reserved Namespace URIs and aliases for RESOURCES in the CADF Event Model

In some cases, the same actual RESOURCE may fulfill more than one of the roles of the CADF Event Model (i.e., INITIATOR, TARGET or OBSERVER). It is not efficient to require the same RESOURCE to be defined multiple times within the scope of the same CADF Event Record if not necessary.

The following Namespace URIs are reserved for use within this specification:

Namespace URI	Description
http://schemas.dmtf.org/cloud/audit/1.0/event/initiator	This value MAY be used, by specified properties, as a value to reference the resource defined by the "`initiator`" or "`initiatorId`" property (i.e., its value) within the same CADF Event data type.
http://schemas.dmtf.org/cloud/audit/1.0/event/target	This value MAY be used, by specified properties, as a value to reference the resource defined by the "`target`" or "`targetId`" property (i.e., its value) within the same CADF Event data type.

http://schemas.dmtf.org/cloud/audit/1.0/event/observer	This value MAY be used, by specified properties, as a value to reference the resource defined by the "`observer`" or "`observerId`" property (i.e., its value) within the same CADF Event data type.

The following Namespace aliases are reserved for use within this specification:

Alias	(alias for) Namespace URI
initiator	http://schemas.dmtf.org/cloud/audit/1.0/event/initiator
target	http://schemas.dmtf.org/cloud/audit/1.0/event/target
observer	http://schemas.dmtf.org/cloud/audit/1.0/event/observer

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:
  - Uppercase ASCII (U+0041 through U+005A)
  - Lowercase 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 requirement 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:

"Required" Constraint Value	Description
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 formats.

5.6 Format-specific representations

This specification is written to be neutral to transmission format because format profiles of this specification are permitted. The intent is that this specification describes the CADF Data Model in a way that allows formats to be authored such that they can easily (and losslessly) be translated form one format to another. However, this specification acknowledges that both XML and JSON are popular formats used by cloud providers and deserve special consideration in this specification.

This clause specifically 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 (see section 7), 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 (see section 7), 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

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 because 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 clause describes how the CADF Entities, data types, and properties defined in this specification would be translated to XML [W3C XML] and JSON [RFC 4627] 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 clause 5.6.1 ("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 proper serialization using a self-closing XML element:

```
<entity1 simple1="some string" simple2="myscheme://mydomain/id/1234">
   <complex1 simpleA="another string"/>
</entity1>
```

JSON serialization:

Showing the proper serialization using a JSON object name for the CADF Entity:

```
{
   "typeURI": "entity1's specified Type URI value",
   "simple1": "some string",
   "simple2": "myscheme://mydomain/id/1234",
   "complex1": {
      "simpleA": "another string"
   }
}
```

6 CADF Entities and data types

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

In this specification, CADF Entities (and in some cases complex CADF 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. 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.

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

- Attachments
- Derivation

- Tags

6.1.1 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.

The data type used to implement Attachments for CADF entities is described in clause 6.4 ("Attachment type").

6.1.1.1 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 CADF 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, and then a downstream Reporter may reverse-lookup that token, resolve it to the identity of a person, and "attach" a custom identity record to the CADF 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.

6.1.2 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 properties in addition to the core properties that are defined in the original CADF Entity or data type (also referenced here "base entity" or "base 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 from which they derive. This means that any CADF Entity or complex data type that is derivable contains a "`typeURI`" property that identifies the base CADF Entity type and any derived type would 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`" property value for the derived provider Resource type may extend the URI value as specified for the base CADF Resource Taxonomy URI (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.

6.1.3 Tags

Tags provide a powerful mechanism for adding domain-specific identifiers and classifications to CADF Event Records which can be referenced by the CADF Query Interface. This allows customers to construct custom reports or views on the event data held by a provider for a specific domain of interest. A CADF Event Record can have multiple Tags that enable cross-domain analysis.

- For example, CADF Tags added to CADF Event Records could help link “events of interest” to customers using well-defined security compliance standards or frameworks (e.g. ISO 27001, PCI DSS, SSAE16, ISACA COBIT, etc.). CADF Tag syntax can be used to identify the frameworks (and their versions) and also include specific numbered control values defined within these frameworks and then associated to the appropriate event records.

The data type used to implement Tags for CADF entities is described in clause 6.3.3 (“Tag type”).

6.2 Basic data types

Basic data types are typically simple (single) values and are not composed of or contain other (standalone) data types and are typically well-understood by most programming languages.

This clause 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. The basic data types we recognize in CADF schema are defined in other specifications that we normatively reference in this section.

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 and SHALL be lowercase.

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 a preceding '-' (minus) sign is permitted to indicate a negative duration.
- 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]+)?
```

6.2.7 URI

The base format and syntax of properties of type “URI” are defined by RFC3986. However, the CADF URI type includes some additional requirements described within this clause.

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 an 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. Table 20 shows how these basic data types would translate from XML to JSON:

Table 20 – Basic type translation 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 clause defines basic CADF data types. These types may be used when defining complex CADF data types and entities. CADF basic data types, much like Basic data types defined in section 6.2, are represented by simple (single) values and are derived from other specifications that we normatively reference in this section. However, these types are different in that this specification provides additional semantic meaning and/or changes in internal format or syntax.

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

Because 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 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Identifier data type:

Type Name	identifier
Type Qualified Name	cadf:identifier
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/identifier

6.3.1.3 Requirements

This specification defines an Identifier type that is based upon the Uniform Resource Identifier Reference (URI) as specified in RFC3986. Any value that represents a CADF Identifier type in this specification, its extensions, or profiles SHALL adhere to the requirements listed in this section:

General requirements

- CADF Identifier type values SHALL be created to be Universally Unique Identifiers (UUIDs) so that when CADF data (e.g., CADF Event Records, Logs, Reports, Resources, Metrics, etc.) are federated it will be uniquely identifiable to the source (e.g., cloud provider, service, etc.) that created them.

Syntax requirements

- CADF Identifiers SHALL adhere to the URI Syntax as defined by in RFC3986 with any exceptions listed in this requirements section.
- CADF Identifiers SHALL NOT have empty paths as allowed by the ABNF grammar of RFC3986.
    - by corollary, CADF Identifiers SHALL end with one or more valid path segments (as defined by RFC3986) in order to assure they are valid UUIDs.
- **Character Encoding:**
    - 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 RFC3986.
- **Namespacing**:
    - CADF Identifiers MAY be constructed using namespace prefixes (i.e., aliases), as defined in in RFC3986, to substitute for portions of an absolute URI.
    - If a namespace is used on a CADF Identifier, the namespace definition SHALL be defined within the same scoping document as the CADF Identifier (e.g. a CADF Log or Report) which references the namespace.

- Aliases defined as part of the CADF standard (see sections 5.2 and 5.3) do not need to be defined when referenced within any CADF Identifier.

6.3.1.4 Lexical representation

- The following syntax is the required Lexical representation of the CADF Identifier type described using RFC3986 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
```

Note: *The CADF identifier data type is compatible with the xs:anyURI data type described by XMLSchema2.*

6.3.1.5 Best practices

- When CADF Identifier values include a protocol scheme (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.
- CADF Identifiers MAY use a namespace prefix to substitute for the scheme, domain and portions of the hierarchical path as long as the identifier is able to reference or resolve the namespace definition which includes the scheme, domain and portions of the hierarchical path that it replaces.
    - For example, within a CADF Log a namespace definition could be defined at the beginning of the log at top-level and any CADF Event Records (or other CADF entities that use CADF Identifiers) that appear within that same CADF Log could use that namespace instead of using the full representation wherever it was needed.

6.3.1.6 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: "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
```

Example 3: "Provider-specified scheme using a UUID"

In this example, the CADF Identifier is composed as namespace alias plus a UUID that is meaningful within the cloud provider that is identified by the namespace.

```
mynamespacealias:9e929943-6903-50ad-af9e-90b68bf8ec59
```

6.3.2 Path type

This clause describes how to represent values that are elements of hierarchies. This construct is used for example when providing values from CADF Taxonomies that classify components of the CADF Event Model within CADF Event Records as path values.

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 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Path data type:

Type Name	path
Type Qualified Name	cadf:path
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/path

6.3.2.3 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:

Syntax requirements

- CADF Path values SHALL adhere to the URI Syntax as defined by in RFC3986 with additional requirements listed below. For convenience, the syntax components from RFC3986 are as follows:

```
scheme ":"  hier-part
```

  - 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 RFC3986 URI Syntax so that they remain extensible.
- CADF Paths SHALL NOT contain the optional fragment component of the RFC3986 URI Syntax so that they remain extensible.
- CADF Paths SHALL contain at least one valid non-zero length path segment (as defined by RFC3986 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.

Absolute path requirements

- Absolute CADF Paths that reference values from this specification SHALL begin with the URI Syntax "authority" and "path-absolute" components set to the following value:

```
http://schemas.dmtf.org/cloud/audit/1.0/
```

- As an alternative, absolute CADF Paths that reference values from this specification MAY use the URI Syntax "scheme" component value (i.e., the CADF Namespace alias) set to the following value:

```
cadf
```

***Note**: Section 5.2 "Namespaces and namespace aliases" defines the CADF specification reserved URI and alias that is shown above.*

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.
  - For example, the "action" property of a CADF Event must always be a value from the CADF Action Taxonomy (or an extension thereof); therefore, a relative path value from that taxonomy MAY be used since the CADF Action Taxonomy URI is assumed to prefix the relative path value provided.
  - For example, the "outcome" property of a CADF Event must always be a value from the CADF Outcome Taxonomy (or an extension thereof); therefore, a relative path value from that taxonomy MAY be used since the CADF Outcome Taxonomy URI is assumed to prefix the relative path value provided.
  - For example, the "typeURI" property of a CADF Resource must always be a value from the CADF Resource Taxonomy (or an extension of it); therefore, a relative path value from that taxonomy MAY

be used since the CADF Resource Taxonomy URI is assumed to prefix the relative path value provided.

- Relative CADF Paths MAY include the optional URI Syntax scheme value (i.e., the value "cadf") along with a ":" (colon) character.

6.3.2.4 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/" ] path-
rootless
```

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

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

6.3.2.5 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.

Note: *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.*

6.3.2.6 Examples

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

In this example, the event's outcome was a "failure". Since the CADF Outcome Taxonomy value for “failure” will appear in the CADF Event property "`outcome`" the context is clearly established; therefore, we are allowed to express the value using a **Relative Path** (and omit the CADF Outcome Taxonomy’s URI path “`http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/outcome/`” when providing the value).

```
<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, specifically 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 (omitting the CADF Resource Taxonomy’s URI path “`http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/`” scheme and root path):

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

```
/>
```

***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 the CADF specification:

```
<event
   ...
   <target typeURI="cadf:taxonomomy/resource/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/database"
   ... />
   ...
/>
```

6.3.3 Tag type

A “Tag” is a label that can be added to a CADF Event Record to qualify or categorize an event. Whereas taxonomies defined in this specification are used to categorize event by the components of the event (See CADF Event Model) according to a predefined classification hierarchies (e.g. the ACTION component, as represented by the “`action`” property of a CADF Event), a “Tag” allows for orthogonal categories to also be associated with the event. For example, a Tag name “PCI-DSS” could be used to label all events related to this security area of concern regardless of their event types, resources involved or assigned taxonomy values.

Tags provide an extensibility mechanism enabling domain-specific views on event data. This specification does not define particular tags, but allows users or profiles of this CADF specification to define sets of tags that match their domain of interest.

6.3.3.1 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Tag data type:

Type Name	tag
Type Qualified Name	cadf:tag
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/tag

6.3.3.2 Requirements

Any value that represents a CADF Tag type in this specification SHALL adhere to the following requirements:

Syntax requirements

The CADF Tag uses URI references with the specific additional requirements listed below. Although a CADF Tag is represented as a single URI value, different parts of a Tag may be distinguished as follows:

- The **Tag namespace** (optional): if a Tag has a namespace, its URI value SHALL be an absolute URI. The URI "authority" and "path-absolute" components (see Path type) up to the path segment before last, represent the namespace. For example, in the Tag (below), the "//GRC20.gov/cloud/security" portion is the Tag namespace:

```
//GRC20.gov/cloud/security/pci-dss
```

- The **Tag name** (required): the Tag name is the last segment of the URI. In the above example, "pci-dss" is the Tag name.
- The **Tag value** (optional): if a Tag has a value, it will be represented by a query parameter named "value". For example, the following Tag named "auditplan" has the value "audit101":

```
//GRC20.gov/cloud/auditplan?value=audit101
```

- If a Tag does not have a namespace, then it SHALL be represented as a relative URI with a single segment (the tag name) in the URI path.
- CADF Tags SHALL NOT contain the optional fragment component of the URI Syntax

6.3.4 Timestamp type

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

6.3.4.1 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 (or timestamp) to 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.

Please see Annex 10B.2 "Treatment of timestamps in CADF Event Records" for discussion of how timestamps are used within the CADF Event Model.

6.3.4.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Timestamp data type:

Type Name	timestamp
Type Qualified Name	cadf:timestamp
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/timestamp

6.3.4.3 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:

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 that this requirement aligns with the representation described in RFC3339.
- Any constraints on the specific ranges allowed for any particular property SHALL be specified by that property's definition.

6.3.4.4 Lexical representation

The following example shows 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
```

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.

6.3.4.5 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 (saving) 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 saving 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.

6.3.4.6 Notes

Relation to existing standard dateTime types

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) and can easily be mapped to or 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 [DSP0004]:
  - Specifically, clause 5.2.4 "Datetime Type", which also references the ISO 8601:2004 format.

Duration or time interval notes

The Timestamp type and its syntax does not allow for any representation of duration or time intervals. Please see Annex 10B.2.2 Handling Activities with Duration.

6.4 Composition of data types in CADF

This clause defines how CADF Entities or data types can be composed into predefined patterns typically seen in programming languages.

6.4.1 Array Syntax

Properties that are arrays of some data 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 examples

Please note that in the following examples the name of the array element is explicitly set by the definition of that property. For the XML examples, the name of the child elements is implicitly set to the name of the contained data type (lowercased). For JSON, which natively supports arrays, a child element name is not necessary.

6.4.1.1.1 Example 1: Array of cadf:attachment type

This example shows sample a property "attachments" that is an array property of the CADF Attachment data type as it might appear in a CADF complex data type definition or CADF Entity definition such as the CADF Event data type:

Table 21 – Sample array type property of cadf:attachment type

Property Name	Type	Required	Description
attachments	cadf:attachment[]	No	A sample array of type CADF Attachment.

The serialization of the array for the "attachments" property 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"
      }
   ]
}
```

6.4.1.1.2 Example 2: Array of cadf:identifier type

The following example shows sample array properties as they would be specified for data types in this specification. For this example, we define one property as an array of the CADF Identifier simple type, and another property as an array of the CADF Attachment complex type:

Table 22 – Sample array type property of cadf:identifier types

Property Name	Type	Required	Description
Ids	cadf:identifier[]	No	A sample array of type CADF Identifier

The serialization of the array for the “ids” property would appear as follows:

XML example

```
<entity>
   ...
   <ids>
      <identifier>http://pcloud.com/dc1/appsrv/4321</identifier>
      <identifier>http://pcloud.com/dc1/dbsrc/1234</identifier>
```

```
      ...
   </ids>
</entity>
```

JSON example

```
{
   ...,
   "ids": [
      "http://pcloud.com/dc1/appsrv/4321",
      "http://pcloud.com/dc1/dbsrc/1234"
   ]
}
```

6.4.2 Map type

This clause introduces a CADF data type used to compose (map) one recognized CADF Entity or data type value to another.

6.4.2.1 Design considerations

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

6.4.2.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Map data type:

Type Name	map
Type Qualified Name	cadf:map
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/map

6.4.2.3 Requirements

Any entity value that represents a 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.
- The Map consists of a number of entries that SHALL each have the property name "`item`" when required by format.

6.4.2.4 Properties

Table 23 describes the properties for the CADF Map type.

1717 **Table 23 – Map type properties**

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

1718 #### 6.4.2.5 Serialization examples

1719 The serialization of a CADF Map complex type (of a simple string typed value) would appear as follows:

1720 **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>
```

1721

1722 **JSON example**

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

1723 ## 6.5 CADF complex data types

1724 This clause defines the complex CADF data types. CADF complex types are composed of or contain other (basic or
1725 complex) data types and collectively we have attached additional semantic meaning to.

1726 CADF complex data types differ from CADF entities in that they are always intended to be used as types for
1727 (complex) properties of CADF entities or other complex types. Unlike entities, they are not supposed to be

accessed independently: the CADF interfaces assume these complex types are always accessed in the context of the parent entities that contain them.

6.5.1 Attachment type

6.5.1.1 Design considerations

The CADF Attachment type is used as one means to add domain-specific information to certain CADF entities or data types. Please see additional discussion on its use in clause 6.1 (Extensibility mechanisms).

6.5.1.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Attachment data type:

Type Name	attachment
Type Qualified Name	cadf:attachment
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/attachment

6.5.1.3 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.
- When the "`content`" property's value contains binary data, the data SHOULD be encoded in Base64.
- When the "`content`" property's value contains XML data, the value of the "`contentType`" SHOULD always be associated with a unique XML schema to which that the content must validate.

6.5.1.4 Properties

Table 24 describes the properties for the CADF Attachment type.

Table 24 – CADF Attachment type properties

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.

6.5.1.5 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".

6.5.1.6 Serialization examples

XML example

```
<event id="myscheme://mydomain/id/1234">
   ...
   <attachments>
      <attachment contentType="scheme://mycontenttype" name="foo">
         <content>
            ...
         </content>
      </attachment>
      ...
   </attachments>
</event>
```

JSON example

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

6.5.2 Credential type

6.5.2.1 Design considerations

This type provides a means to describe various credentials along with any information about the authority that is responsible for maintaining them. This is intended to be associated with a CADF Resource's identity and reflects any authorizations or identity assertions the resource may use to gain access to other resources.

6.5.2.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Credential data type:

Type Name	credential
Type Qualified Name	cadf:credential
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/credential

6.5.2.3 Requirements

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

- Valid Credential typed data SHALL contain at least one valid identify token.
- The "`token`" property SHALL contain the primary identity token, credential or assertion value which was used to represent the INITIATOR's access credentials at the time an authorized access (i.e. ACTION) to the TARGET resource(s) was observed (by the OBSERVER resource).
- Additional, relevant secondary identity token, credential or other assertion values MAY be added to the "`assertions`" property.

6.5.2.4 Properties

Table 25 describes the properties for the CADF Credential type.

Table 25 – Credential type properties

Type Name	**credential**		
Property	**Type**	**Required**	**Description**
type	xs:anyURI	No	Type of credential. (e.g., auth. token, identity token, etc.) ***Note**: Profiles of this specification MAY define URIs for their credential types.*
token	xs:any	Yes	The primary opaque or non-opaque identity or security token (e.g. an opaque or obfuscated user ID, opaque security token string, or security certificate). ***Note**: the "assertions" property allows for any number of additional or associated credentials to be included for the same identity.*
authority	xs:anyURI	No	Identifies the trusted authority (a service) that understands and can verify the credential.
assertions	cadf:map	No	Optional list of additional opaque or non-opaque assertions or attributes that belong to the credential (see Notes below).

6.5.2.5 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 type of identity assertion to be included in either the primary "`token`" property or within the "`assertions`" property map.

Examples of credential data that may be represented in this data type include:

- Simple "userid-password" credentials or basic authentication information
- 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
- User roles, job credentials or responsibilities, physical characteristics, etc.
- other types by enabling assertion based description of other credential formats

6.5.2.6 Serialization examples

XML example

```
<event action=”authenticate”>
   ...
   <initiator id="joe.user@tenant1.com"
      typeURI=”data/security/account/user" />
         ...
         <credential type=”https://mycloud.com/v2/token”
            token=”myuuid:1ef0-abdf-xxxx-xxxx”/>
   </initiator>
</event>
```

JSON example

```
{
   "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
   “action”: “authenticate”,
   ...,
   "initiator": {
      "id": "joe.user@tenant1.com",
      “typeURI”: ”data/security/account/user",
      ...,
      "credential": {
         "type": "https://mycloud.com/v2/token",
         "token": "myuuid:1ef0-abdf-xxxx-xxxx"
      }
   }
```

```
}
```

6.5.3 Endpoint type

6.5.3.1 Design considerations

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

6.5.3.2 Type name and URI

The following type name, qualified name and URI values are used to identify the CADF Endpoint data type:

Type Name	endpoint
Type Qualified Name	cadf:endpoint
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/endpoint

6.5.3.3 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 "`url`" property and its URI scheme (i.e., its domain-specific protocol scheme).

6.5.3.4 Properties

Table 26 describes the properties for the CADF Endpoint type.

Table 26 – Endpoint type properties

Type Name	**Endpoint**		
Property	**Type**	**Required**	**Description**
url	xs:anyURI	Yes	The network address of the endpoint. For IP-based addresses. ***Note**: the IP address value may include the port number as part of the syntax as an alternative to separating it out into the optional attribute provided below.*
name	xs:string	No	An optional property to provide a logical name for the endpoint.
port	xs:string	No	An optional property to provide the port value separate from the address property. ***Note**: This property is intended to facilitate a consistent means to query resource information on a specific port.*

6.5.3.5 Serialization examples

XML example

```
<event>
   ...
   <target
   id="myscheme://mydomain/network/node/9999"
```

```
   name="network-node-9999"
      <addresses>
         <endpoint
            name=”public”
            url="http://mydomain/mypath/server-0001/" />
         ...
      </addresses>
      ...
   </target>
</event>
```

JSON example

```
{
   "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
   ...,
   "target": {
      "id": "myscheme://mydomain/resource/id/0001",
      "name": "server_0001",
      "addresses": [
         {
            “name”: “public”,
            ”url”: “http://mydomain/mypath/server-0001/”
         },
         ...
      ],
      ...
   }
}
```

6.5.4 Eventset type

The Eventset type’s schema is intended to contain one or more event elements within a simple structure along with relevant metadata, such as associated resources, metrics, attachments, etc. The format is designed for data federation and sharing use cases, or as a base structure upon which more refined structures may be defined by profile.

6.5.4.1 Design considerations

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

- The Eventset type 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 Eventset type may be assigned a time period that defines time boundaries (a begin date/time, and end date/time) for all events included in the set.

6.5.4.2 Type name and URI

The following type name, qualified name and URI values are used to identify the CADF Eventset data type:

Type Name	eventset
Type Qualified Name	cadf:eventset
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/eventset

6.5.4.3 Requirements

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

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

6.5.4.4 Properties

Table 37 describes the properties for the CADF Eventset type:

Table 27 – Eventset data type properties

Type Name	eventset		
Property	**Type**	**Required**	**Description**
beginTime	cadf:timestamp	No	The beginning time for the time period of event records within the Eventset. Event records that appear in the Eventset 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 Eventset. Event records that appear in the Eventset should only have event times (timestamps) that are equal to or less than this time.
resources	cadf:resource[]	No	An optional array of CADF Resources that may be referenced by multiple CADF Event Records within the Eventset (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 Eventset (i.e., the resources refer to a geolocation by its ID, as part of a resource typed property, such as a TARGET or INITIATOR).
metrics	cadf:metric[]	No	An optional array of CADF Metrics that may be referenced by multiple CADF Events Records within the Eventset (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 Eventset. ***Note:*** *In the case that the Eventset data represents a time period (as designated by the 'beginTime' and 'endTime' period) when no event records were captured (i.e., an empty set), the 'events' property should be present but the array should contain no elements (i.e., be an "empty" array of events).*

6.5.4.5 Serialization examples

XML example

```
<eventset
   beginTime="2012-03-22T13:00:00-04:00"
   endTime="2012-03-29T13:00:00-04:00”
   ...
   <events>
      <event id="myscheme://mydomain/event/id/AAA">
         ...
      </event>
      <event id="myscheme://mydomain/event/id/BBB">
         ...
      </event>
```

```
      ...
   </events>
</eventset>
```

JSON example

```
{
   "beginTime": "2012-03-22T13:00:00-04:00",
   "endTime": "2012-03-29T13:00:00-04:00",
   ...,
   "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",
         ...
      },
      ...
   ]
}
```

6.5.5 Geolocation type

6.5.5.1 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 an event is compliant sometimes depends on the geolocation of the event. Therefore, it is crucial to report geolocation information unambiguously in an audit trail.

6.5.5.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Geolocation data type:

Type Name	geolocation
Type Qualified Name	cadf:geolocation
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/geolocation

6.5.5.3 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 all supplied as properties describing the same geolocation, the "`latitude`" and "`longitude`" properties' 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.

6.5.5.4 Properties

Table 28 defines the properties for the CADF Geolocation type.

Table 28 – Geolocation type properties

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

latitude	xs:string	No	Indicates 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:

Type Name	geolocation		
Property	**Type**	**Required**	**Description**

			`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	Indicates 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 (−).

Type Name	geolocation		
Property	**Type**	**Required**	**Description**
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	Indicates the city of a geolocation.
state	xs:string	No	Indicates the state/province of a geolocation
regionICANN	xs:string	No	Indicates a region (e.g., a country, a sovereign state, a dependent territory or a special area of geographical interest) of a geolocation. The value used to indicate the region SHOULD match the 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 lowercase; they are viewed as semantically equivalent.*
annotations	cadf:map	No	Indicates user-defined geolocation information (e.g., building name, room number). The same “key” SHALL NOT be used more than once within an "annotation" property.

6.5.5.5 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

6.5.5.6 Serialization examples

XML examples

The following several examples show 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="myuuid://location.org/XYZ"
         unit="GB"
         name="Storage Capacity in Gigabytes"/>
      ...
   </geolocations>
   ...
```

```
   <events>
      <event>
         ...
         <target id="myscheme://mydomain/resource/id/0001"
            typeURI="cadf://.../taxonomy/resource/..."
            name="server_0001"
            ref="http://mydomain/mypath/server_0001/"
            ...
            geolocationId="myuuid://location.org/XYZ"/>
         ...
      </event>
   </events>
</log>
```

1884 **JSON examples**

1885 **JSON example 1:** "latitude and longitude"

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

1886 **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"
      }
   }
}
```

1887 **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"
      }
   }
}
```

1888 **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"
      }
   }
}
```

1889 **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"
         }
      ]
   }
}
```

1890 **JSON example 6:** Geolocation referenced by a CADF Event

1891 The following example shows a Geolocation definition being referenced from a TARGET resource within a CADF
1892 Event Record that is defined within the same CADF Log.

```
{
   "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/log",
   ...,
   "geolocations": [
      {
         "geolocationId": "myuuid://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": "myuuid://location.org/XYZ"
         }
      }
   ]
```

```
}
```

6.5.6 Host type

6.5.6.1 Design considerations

Most resources that are referenced in an IT or cloud infrastructure are conceptually “hosted on” or “hosted by” other resources. For example, “applications” are hosted on “web servers” or “users” may be hosted on a “network connected device” or a “terminal”. In addition, networked resources are “hosted” by some device attached to some network.

The host resource often provides context or location information for the resource it is hosting at the time the Actual Event was observed and recorded (e.g., an IP address, software agent, platform, etc.). Providing a means to record host information with a CADF Event Record is valuable for audit purposes since compliance policies and rules are often based on such information.

6.5.6.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Host data type:

Type Name	host
Type Qualified Name	cadf:host
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/host

6.5.6.3 Requirements

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

- Host typed data SHALL contain at least one valid property and associated value.

6.5.6.4 Properties

Table 29 describes the properties for the CADF Host type.

Table 29 – Host type properties

Type Name	**host**		
Property	**Type**	**Required**	**Description**
id	cadf:identifier	No	The optional identifier of the host RESOURCE. ***Note**: This SHOULD be the “id” for a CADF Resource if known.*
address	xs:anyURI	No	The optional address of the host RESOURCE.
agent	xs:string	No	The optional agent (name) of the host RESOURCE.
platform	xs:string	No	The optional platform of the host RESOURCE.

6.5.6.5 Serialization examples

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

XML example

```
<host id=”myuuid:1234-5678-90abc-defg-0000”
   address="10.0.2.15"
   agent="Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:18.0)"
   platform="Linux version 3.5.0-23-generic (gcc version 4.6.3
   (Ubuntu/Linaro 4.6.3-1ubuntu5) ) #35~precise1-Ubuntu SMP Fri Jan 25
   17:15:33 UTC 2013"
/>
```

JSON example

```
{
   “id”: ”myuuid:1234-5678-90abc-defg-0000”,
   “address”: "10.0.2.15",
   “agent”: "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:18.0)",
   “platform”: "Linux version 3.5.0-23-generic (gcc version 4.6.3
   (Ubuntu/Linaro 4.6.3-1ubuntu5) ) #35~precise1-Ubuntu SMP Fri Jan 25
   17:15:33 UTC 2013"
}
```

6.5.7 Metric and measurement types

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

6.5.7.1 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.

6.5.7.2 Type names and URIs

The following type name, qualified name and URI are used to identify the CADF Metric data type:

Type Name	metric
Type Qualified Name	cadf:metric
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/metric

The following type name, qualified name and URI are used to identify the CADF Measurement data type:

Type Name	measurement
Type Qualified Name	cadf:measurement
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/measurement

6.5.7.3 Requirements

Any entity value that represents a 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.

6.5.7.4 Properties of Metric type

Table 30 describes the properties for the Metric type.

Table 30 – Metric type properties

Type 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	User-defined metric information. The same “key” SHALL NOT be used more than once within a "annotation" property.

6.5.7.5 Properties of Measurement type

Table 31 describes the properties for the Measurement type.

Table 31 – Measurement type properties

Type 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 property 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 that 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).

6.5.7.6 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="myuuid://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="myuuid://metric.org/1234"
         unit="GB" name="Storage Capacity in Gigabytes"/>
      ...
   </metrics>
   ...
   <events>
      <event
      ...
      <measurements>
         <measurement result="10 metricId="myuuid://metric.org/1234"/>
      </measurements>
      ...
      </event>
   </events>
</log>
```

JSON examples

The following several examples show 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": "myuuid://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": "myuuid://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": "myuuid://metric.org/1234"
            }
         ],
         ...
      }
   ]
}
```

6.5.8 Reason type

This data type is defined to further describe and provide additional information relevant to the OUTCOME of an Actual Event, as part of the CADF Event Record.

6.5.8.1 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.

6.5.8.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Reason data type:

Type Name	reason
Type Qualified Name	cadf:reason
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/reason

6.5.8.3 Requirements

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

- If the CADF Reason type is provided within a CADF Event Record, it SHALL contain either a "`reasonCode`" or a "`policyId`" property, or both. Furthermore,
  - if a "`reasonCode`" property value is provided, a valid "`reasonType`" property value SHALL also be provided,
  - if a "`policyId`" property value is provided, a valid "`policyType`" property value SHALL also be provided.
- 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`", if provided, SHALL NOT have an "empty", "blank", or zero-length value.
- The property "`reasonCode`", if provided, SHALL NOT have an "empty", "blank", or zero-length value.

6.5.8.4 Properties

Table 32 describes the properties for the Reason type.

Table 32 – Reason type properties

Type Name	**reason**		
Property	**Type**	**Required**	**Description**
reasonType	xs:anyURI	No	The domain URI that defines the "reasonCode" property's value. See examples below.
reasonCode	xs:string	No	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.*
policyType	xs:anyURI	No	The domain URI that defines the "policyId" property's value. See examples below.
policyId	xs:string	No	An optional identifier that indicates which policy or algorithm was applied in order to achieve the described OUTCOME.

6.5.8.5 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" property value, 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."

Similarly, the "policyId" property is entirely domain-specific and may represent anything from a firewall rule to an authentication policy to a virus signature. Since in many cases policies may be custom-defined within the application, the "policyType" URI may point to the unique source instance within which the policies are defined. These properties will commonly be used for 'control'-type CADF Event Records, but may also appear in other types of events.

6.5.8.6 Serialization examples

XML example

```
<event>
   ...
   <reason
      reasonType="http://www.iana.org/assignments/http-status-codes/http-
      status-codes.xml"
      reasonCode="408"
      policyType="http://schemas.xmlsoap.org/ws/2002/12/policy"
      policyId="http://10.0.3.4/firewall-ruleset/rule0012"/>
   ...
</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",
      "policyType": "http://schemas.xmlsoap.org/ws/2002/12/policy",
      "policyId": "http://10.0.3.4/firewall-ruleset/rule0012"
```

```
    },
    ...
}
```

6.5.9 Reporterstep type

This type represents a step in the REPORTERCHAIN that captures information about any notable REPORTER (in addition to the OBSERVER) that modified or relayed the CADF Event Record and any details regarding any modification it performed on the CADF Event Record it is contained within.

6.5.9.1 Design considerations

- The Reporterstep data type should capture information about the resources that have had a role in modifying, or relaying the CADF Event Record during its lifecycle after having been created by the OBSERVER.
- The intent of Reporterstep data, when included within a REPORTERCHAIN, is to support forensic auditing of the sources of event data and the systems that subsequently handle that data for the purposes of verification, validation, and troubleshooting (i.e., these sources of event data are CADF REPORTERS).
- The timestamp value that appears in the "`reporterTime`" property, as filled in from any one REPORTER's perspective, might not be accurate with respect to any other REPORTER's "`reporterTime`" value (e.g., perhaps due to local clock differences).

6.5.9.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Reporterstep data type:

Type Name	reporterstep
Type Qualified Name	cadf:reporterstep
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/reporterstep

6.5.9.3 Requirements

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

- Any REPORTER that observes a CADF Event Record MAY be recorded as part of a Reporterstep entry in the CADF Event type's "`reporterchain`" property with its "`role`" property set to the value "observer".
    - Any Reporterstep entry with a "`role`" value of "`observer`" SHALL be the first entry in the "`reporterchain`" and there SHALL only be one entry with this value.
    - If a Reporterstep entry has the "`role`" value equal to "observer", then the REPORTER referenced in this entry SHALL be the same resource (i.e., have the same CADF Identifier) as the resource referenced as the OBSERVER resource in the same CADF Event Record.
- Any REPORTER that modifies the CADF Event Record in any way SHOULD be added as a part of a Reporterstep entry in the CADF Event type's "`reporterchain`" property with its "`role`" property set to the value "modifier".
- Any REPORTER that relays or transmits the CADF Event Record (without modifying it) in any way MAY be added as a part of a Reporterstep entry in the CADF Event type's "`reporterchain`" property with its "`role`" property set to the value "relay".
    - 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 CADF Event type's "`reporterchain`" property.

- A Reporterstep entry SHALL contain either a valid "reporter" property or a valid "reporterId" property, but SHALL NOT contain both properties.

Additional Requirements for the "reporterTime" property

- If the "role" property has a value of "observer" and the "reporterTime" property is not present, then the "reporterTime" property's value MAY be assumed to be the same as the "eventTime" property's value provided within the same the CADF Event Record.
- If the "role" property has a value other than "observer" (i.e., "modifier" or "relay") and the "reporterTime" property is not present, then the "reporterTime" property's value MAY be assumed to be the same time as (or the granular equivalent to) the "reporterTime" property value of the previous Reporterstep entry listed within the REPORTERCHAIN of the same CADF Event Record.

6.5.9.4 Properties

Table 33 describes the properties for the Reporterstep type.

Table 33 – Reporterstep type properties

Type Name	reporterstep		
Property	**Type**	**Required**	**Description**
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 clause "Reporter Roles".
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 to reference a valid CADF Resource definition, which is already defined and can be referenced by its identifier (e.g., a CADF Resource already defined within the same CADF Event record or at the CADF Log or Report level that also contains the referencing CADF Event record).* ***Note:*** *Aliases for resources already defined within the same CADF Event record MAY be used as valid values for this property (see Section 5.3 "Reserved Namespace URIs and aliases for RESOURCES in the CADF Event Model").* **Dependent Requirements** • This property SHALL be required when the "reporter" property is not used.
reporterTime	cadf:timestamp	No	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.

6.5.9.5 Serialization examples

XML example

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

JSON example

```
{
   "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
   ...,
   "reporterchain": [
      {
         "role": "observer",
         "reporterTime": "2012-03-22T13:00:00-04:00",
         "reporter": {
            "id": "myscheme://mydomain/resource/monitor/id/0002"
         }
      },
      ...
   ]
}
```

6.5.10 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.

6.5.10.1 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 that will enable these goals, but also permit extended resource descriptions for specific profiles of this specification.

6.5.10.2 Type name and URI

The following type name, qualified name and URI are used to identify the CADF Resource data type:

Type Name	resource
Type Qualified Name	cadf:resource
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/resource

6.5.10.3 Requirements

Any entity value that represents a 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.

6.5.10.4 Properties

Table 34 describes the properties for the CADF Resource type.

Table 34 – Resource type properties

Type 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).
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.).

credential	cadf:credential	No	The optional security credentials associated with the resource's identity.
addresses	cadf:endpoint[]	No	The optional descriptive addresses (including URLs) of the resource.
host	cadf:host	No	The optional information about the (network) host of the resource.
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 that 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.

6.5.10.5 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"
      }
   }
}
```

6.5.11 Resultset type

The Resultset type's schema is intended to contain one or more event elements that are compiled together by a system component in response to a query by a consumer.

Conceptually, a "set" of results is a temporary dataset, possibly filtered, that is extracted from an event repository in response to some query. Although a set is not considered to be immutable, in general consumers will expect that identical queries will always return identical results from the same provider, with the caveat that additional new data might be present (but no data will have disappeared).

6.5.11.1 Design considerations

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

- The Resultset type should contain the data needed to allow providers of large query result sets to present the data in multiple "pages" that can be navigated by the data's consumer.
- The Resultset should contain the information provided as part of the query that was used to compile and produce the result data such as the query filter and detail level requested.

6.5.11.2 Type name and URI

The following type name, qualified name and URI values are used to identify the CADF Resultset data type:

Type Name	resultset
Type Qualified Name	cadf:resultset
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/resultset

6.5.11.3 Requirements

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

- In the case that the query used to produce the Resultset contains no event records (i.e., an empty set), the 'eventSet' property SHOULD still be present with valid properties; specifically, the 'events' property should be present but the array should contain no elements (i.e., be an "empty" array of events).
- The `detailLevel` property's value SHOULD NOT be higher than that requested by the consumer (as part of a CADF Query), but it can be lower – in other words, the provider can provide less detail, but not more than was asked for.

6.5.11.4 Properties

The following table describes the properties for the CADF Resultset:

Table 35 – Resultset data type properties

Type Name	set		
Property	**Type**	**Required**	**Description**
filter	xs:string	No	Contains the filter specification provided by the requester (on a query) that was used to produce the resultset and allows the consumer to reconstruct how the set was generated.
count	xs:integer	No	Lists the total number of CADF Event Records included in this resultset.
nextPage	xs:anyURI	No	In some cases, a resultset will be broken up into multiple pages to restrict the size of a single page. This property will provide a pointer to the next page in the sequence. See section 7.1.6 “Limiting query results”. *Note: If a resultset is paginated, providers are **strongly encouraged** to include this property.*
prevPage	xs:anyURI	No	In some cases, a resultset will be broken up into multiple pages to restrict the size of a single page. This property will provide a pointer to the previous page in the sequence. See section 7.1.6 “Limiting query results”. *Note: If a resultset is paginated, providers are **strongly encouraged** to include this property.*
firstPage	xs:anyURI	No	In some cases, a resultset will be broken up into multiple pages to restrict the size of a single page. This property will provide a pointer to the first page in the sequence. See section 7.1.6 “Limiting query results”.
lastPage	xs:anyURI	No	In some cases, a resultset will be broken up into multiple pages to restrict the size of a single page. This property will provide a pointer to the last page in the sequence. See section 7.1.6 “Limiting query results”.
detailLevel	xs:integer	No	CADF Event Records stored in a resultset can be stored with various levels of detail, as defined in Section 7.1.6.2 “Specifying level of detail for results”. This parameter contains one of the following: • ‘1’: indicates a resultset that contains CADF Event Records with only the most important event details. • ‘2’: indicates a resultset that contains CADF Event Records with a mid-level of detail. • ‘3’: Indicates a resultset that contains CADF Event Records with all known details. If this option is not present, the consumer may not make assumptions about which event details are present/absent and will have to examine the data directly.
eventSet	cadf:eventset	Yes	The set of events described by the CADF Resultset.

6.5.11.5 Serialization examples

XML example

```
<resultset
```

```
  count="2"
  nextPage="http://<addr>/events/event?filter=eventTime>="2012-05-
22T00:00:00-02:00"&limit=2&offset=3"
  firstPage="http://<addr>/events/event?filter=eventTime>="2012-05-
22T00:00:00-02:00"&limit=2&offset=1"
  lastPage="http://<addr>/events/event?filter=eventTime>="2012-05-
22T00:00:00-02:00"&limit=2&offset=3"
   ...
   <eventSet>
     <events>
        <event id="myscheme://mydomain/event/id/AAA">
         ...
        </event>
        <event id="myscheme://mydomain/event/id/BBB">
         ...
        </event>
        ...
     </events>
   </eventSet>
</set>
```

2128 **JSON example**

```
{
  "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/resultset",
  "count"=2,
  "nextPage="http://<addr>/events/event?filter=eventTime>="2012-05-
22T00:00:00-02:00"&limit=2&offset=2",
  "firstPage"="http://<addr>/events/event?filter=eventTime>="2012-05-
22T00:00:00-02:00"&limit=2&offset=1",
  "lastPage"="http://<addr>/events/event?filter=eventTime>="2012-05-
22T00:00:00-02:00"&limit=2&offset=3",
  "eventSet": {
     "events": [
       {
          "id": "myscheme://mydomain/event/id/1234"
          ...
       },
       {
          "id": "myscheme://mydomain/event/id/3333"
          ...
       },
     ]
```

```
    },
}
```

6.6 CADF Entities

This clause defines CADF Entities, as inspired from Entity-Relationship (ER) modeling, which 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.*

6.6.1 Event (data) type

This entity represents the CADF Event Record.

6.6.1.1 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.

6.6.1.2 Type name and URI

The following type name, qualified name and URI values are used to identify the CADF Event data type:

Type Name	event
Type Qualified Name	cadf:event
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/event

6.6.1.3 Requirements

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

- The CADF Event data type SHALL contain either a valid "`initiator`" property or a valid "`initiatorId`" property, but SHALL NOT contain both properties.
- The CADF Event data type SHALL contain either a valid "`target`" property or a valid "`targetId`" property, but SHALL NOT contain both properties.
- The CADF Event data type SHALL contain either a valid "`observer`" property or a valid "`observerId`" 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 SHOULD represent the perspective of the OBSERVER (see clause "Required model components").

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 SHOULD represent the perspective of the OBSERVER (see clause "Required model components").

Initiator, target and observer property requirements:

The "initiator", "target" and "observer" properties' "typeURI" property each:

- SHALL include a valid resource classification value from the CADF Resource Taxonomy or an extension thereof.
- SHOULD represent the perspective of the OBSERVER (see clause "Required model components").

6.6.1.4 Properties

Table 36 describes the properties for the CADF Event type.

Table 36 – Event data type properties

Type 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 clause 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. • If 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 classification of the type of event. • This property SHALL contain a valid value from the list of valid EventType values as specified in Section 4.5.1 "Valid EventType values" or be a valid value from an official profile of this specification). **Note**: The "eventType" property's value affects the requirements (prescription level) for other properties within the CADF Event data type.
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).

Type Name	event		
Property	**Type**	**Required**	**Description**
action	cadf:path	Yes	This property represents the event's ACTION. See Basic Model Components for details. 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.* ***Note:*** *Aliases for resources already defined within the same CADF Event record MAY be used as valid values for this property (see Section 5.3 "Reserved Namespace URIs and aliases for RESOURCES in the CADF Event Model").* **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.

Type Name	event		
Property	**Type**	**Required**	**Description**
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.* ***Note:*** *Aliases for resources already defined within the same CADF Event record MAY be used as valid values for this property (see Section 5.3 "Reserved Namespace URIs and aliases for RESOURCES in the CADF Event Model").* **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).
observer	cadf:resource	Dependent (see description)	This property represents the OBSERVER. See Basic model components for details. **Dependent Requirements** • This property SHALL be required if the "observerId" property is not used.
observerId	cadf:identifier	Dependent (see description)	This property identifies the event's OBSERVER by reference. ***Note:*** *This property can be used instead of the "observer" 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 OBSERVER.* ***Note:*** *Aliases for resources already defined within the same CADF Event record MAY be used as valid values for this property (see Section 5.3 "Reserved Namespace URIs and aliases for RESOURCES in the CADF Event Model").* **Dependent Requirements** • This property SHALL be required if the "observer" property is not used. If this property is used, its value SHALL reference a valid CADF Resource definition (e.g., at CADF Log level).
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**

Type Name	event		
Property	**Type**	**Required**	**Description**
			• This property SHALL be required if the "eventType" property has a value of "monitor"; otherwise, this property is optional.
reason	cadf:reason	Dependent (see description)	This property contains domain-specific reason code and policy data that provides an additional level of detail to the outcome value. **Dependent Requirements** • This property SHALL be required if the "eventType" property has a value of "control"; otherwise, this property is optional.
name	xs:string	No	This optional property represents a descriptive name for the event. This property SHALL NOT be used in place of the required CADF Event property "id".
severity	xs:string	No	This optional 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.*
duration	cadf:duration	No	This optional property describes the duration of activity for long-running activities. It is typically used in the second of a pair of events marking the start and end of such activity. ***Note:*** *See* Annex B.2.2 for best practices on usage.
tags	cadf:tag[]	No	An optional array of Tags that MAY be used to further qualify or categorize the CADF Event Record. ***Note****: Tags enable the querying of domain-specific views on a provider's event data.*
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.

6.6.1.5 Serialization examples

XML examples

The following example shows the CADF Event Record using the in-line properties "`initiator`", "`target`" and "`observer`", 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="myuuid://location.org/resource/0001" typeURI="..."/>
      <target id="myuuid://location.org/resource/0099" typeURI="..."/>
      <observer id=”myuuid://location.org/resource/0321” typeURI=”...”/>
      <reporterchain>
         <reporterstep
            role="observer"
            reporterTime="2012-08-22T23:00:00-02:00">
               <reporter id="myuuid://location.org/resource/0321"/>
         </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 that are fully defined within the same CADF Log that also contains the CADF Event Record itself.

```
<log>
   ...
   <resources>
      <resource id="myuuid://location.org/resource/0001" typeURI="..."/>
      <resource id="myuuid://location.org/resource/0099" typeURI="..."/>
      <resource id="myuuid://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="myuuid://location.org/resource/0001"
         targetId="myuuid://location.org/resource/0099"
         observerId=”myuuid://location.org/resource/0321”
         <reporterchain>
            <reporterstep role="observer"
              reporterTime="2012-08-22T23:00:00-02:00">
                 <reporter id="myuuid://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": "myuuid://location.org/resource/0001",
```

```
      "typeURI": "..."
   },
   "target": {
      "id": "myuuid://location.org/resource/0099",
      "typeURI": "..."
   },
   "observer": {
      "id": "myuuid://location.org/resource/0321",
      "typeURI": "..."
   },
   "reporterchain": [
      {
         "role": "observer",
         "reporterTime": "2012-08-22T23:00:00-02:00",
         "reporterId”: "..."
      },
      ...
   ]
}
```

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 that are fully defined within the same CADF Log that also contains the referencing CADF Event Record itself.

```
{
   "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/log",
   ...,
   "resources": [
      {
         "id": "myuuid://location.org/resource/0001",
         "typeURI": "...",
         ...
      },
      {
         "id": "myuuid://location.org/resource/0099",
         "typeURI": "...",
         ...
      },
      {
         "id": "myuuid://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": "myuuid://location.org/resource/0001",
         "targetId": "myuuid://location.org/target/0099",
         "observerId": "myuuid://location.org/target/0099",
         "reporterchain": [
            {
               "role": "observer",
               "reporterTime": "2012-08-22T23:00:00-02:00",
               "reporter": {
               "id": "myuuid://location.org/target/0321"
            }
         ]
      }
   ]
},
...
]
}
```

6.6.1.6 Best practices

CADF Logs and CADF Reports provide a facility to fully describe resources, metrics, geolocations and attachments globally (once) so that CADF Event Records also included in the same log or report may reference these definitions by their respective identifiers (i.e., UUIDs) and not have to describe them repeatedly within each in each event record.

- CADF Event Records that appear within a CADF Log or CADF Report SHOULD reference, by identifier, log-level or report-level definitions (e.g. resource, metric, geolocation, attachment, etc.) when possible.
- 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. This example's resource referencing technique (by identifier) can also be used for INITIATORS and REPORTERS.

6.6.1.7 Providing resource taxonomy synonyms for event resources

This section describes a mechanism that can be used to provide alternate values for resource taxonomy classification values.

Objective

Define syntax for use with the CADF Tag type allowing the declaration of additional or alternative resource classifications for those that are part of the normative CADF Resource Taxonomy. These alternative classifications could be then associated with the top-level resources defined on a CADF Event (i.e. as defined by its `initiator`, `target` or `observer` properties) and used to provide a means to query CADF Event Records when the resource may have secondary or tertiary classifications other than the primary one provided in the event's "`typeURI`" property.

In these cases, such alternative taxonomy values are specified as extensions in the form of particular tag items of the tags array.

Syntax and semantics

This specification reserves the following URI (i.e. the CADF Taxonomy Synonym URI) and its alias that may be used when creating CADF Tag values to be placed in the CADF Event's "`tag`" property:

CADF Taxonomy Synonym URI	
URI	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/synonym/
URI alias	cadf:taxonomy/synonym

The alternative taxonomy classification is done using the following CADF Tag conventions:

CADF Tag Component	Definition
namespace	The URI (or its alias) for a CADF Taxonomy Synonym as defined above: • http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/synonym/
name	The name of the CADF Event attribute given alternative classification. • e.g. initiator, target or observer
value	The taxonomy value starting with the taxonomy root (resource), • e.g. resource/storage/database

Example

For example, assume that a CADF Event instance has a "`typeURI`" property with the value:

```
http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/resource/data/database
```

The following CADF Tag with component property "name" equal to the keyword "target" defines an alternative taxonomy value for the "target" property defined within the same the CADF Event record.

```
http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/synonym/target?value=
resource/storage/database
```

One or more alternative resource CADF Resource Taxonomy tags may be added as tag extensions (i.e., using the "tags" property) to a CADF Event record.

The resulting CADF Event Record would look something like the following (in JSON format pseudo-code) where a "storage/database" classification can be used as a synonym for the "data/database" classification supplied on the "target" resource's TypeURI property:

```
{
   "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": { ... },
   "target": {
      "id": "myuuid://location.org/resource/0099",
      "typeURI": "data/database"
   },
   "observer": { ... },
   "tags": [
      {
      "http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/synonym/target?value=
      resource/storage/database"
      }
   ]
}
```

6.6.2 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 log format is suitable for federation and composition with other logs of the same schema.

Conceptually, a "log" is an "immutable" entity that is provided as part of a defined auditing process. The CADF acknowledges that the concept of and uses for "logs" may be different within different domains. Therefore, this specification provides this base type which SHALL be used by profiles (e.g. domain-specific extensions) of this specification.

- Please see the clause titled "Differences between reports and logs" in the subsequent section for further discussion.

6.6.2.1 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.

6.6.2.2 Type name and URI

The following type name, qualified name and URI values are used to identify the CADF Log data type:

Type Name	log
Type Qualified Name	cadf:log
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/log

6.6.2.3 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 Log (e.g., CADF Resource, CADF Event, CADF Metric, etc.) SHALL have a unique identifier (cadf:identifier) within the report.

6.6.2.4 Properties

Table 37 describes the properties for the CADF Log type:

Table 37 – Log data type properties

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 clause of this specification. Additional requirements are listed below. **Dependent Requirements** • If the "typeURI" property is included on this entity, 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. • If 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 INITIATOR).
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.

2258 #### 6.6.2.5 Serialization examples

2259 **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>
```

2260 **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",
         ...
      },
      ...
   ]
}
```

6.6.2.6 Notes

- The CADF Log can be viewed as a modelable extension of the CADF Eventset; however, for this version of the CADF specification, the CADF Log duplicates definitions for several of the properties that are also defined in the CADF Eventset.

6.6.3 Report type

The report is intended to contain one or more event records that are compiled with other auditing information in response to some step within an auditing process. Please note that this specification version does not describe how CADF Reports are created, but provides it for domain-specific extension via profiles of this specification.

6.6.3.1 Differences between reports and logs

Fundamentally, logs are intended to a compact, simple 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.

CADF acknowledges that, in this core specification, the CADF Log and Report data types may look very similar. However, in auditing domains and within compliance frameworks, reports and logs are distinct entities with different functional purposes. Therefore, having distinctly separate types for logs and reports enables profiles of this specification to extend either as they see fit.

Note: It is expected that profiles of this specification to convey their specific log and report information via extensions of these the CADF Log and Report types in order to remain compatible with CADF Interfaces (i.e. by using CADF extension mechanisms). For example, an SSAE16 report could be attached to a CADF Entity and signed along with other information and provided to a cloud consumer.

6.6.3.2 Design considerations

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

- The report may contain either 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 repeatedly referenced by entities contained within the report.

6.6.3.3 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.

6.6.3.4 Type name and URI

The following type name, qualified name and URI values are used to identify the CADF Report data type:

Type Name	report
Type Qualified Name	cadf:report
Type URI	http://schemas.dmtf.org/cloud/audit/1.0/report

6.6.3.5 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.

6.6.3.6 Properties

Table 38 describes the properties of the CADF Report type:

Table 38 – Report data type properties

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 clause of this specification. Additional requirements are listed below.
			Dependent Requirements
			If the "typeURI" property is included on this entity, 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. • If 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 INITIATOR).
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.

6.6.3.7 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 Interfaces

7.1 CADF Query Interface

This clause defines the CADF Query Interface. As CADF is primarily concerned with the representation of IT activity in CADF Event Records, the CADF Query Interface is focused on flexibly requesting sets of those records from providers and returning them to audit event consumers. CADF event providers must implement a compatible mechanism to respond to these requests and return accurate result sets.

7.1.1 Design Notes

Please note that the CADF Query Interface is designed to work with the DMTF CIMI Model or any RESTful HTTP-based protocol concept using a "`filter`" query parameter.

- Examples of how the CADF Query Interface and Syntax can be used, with results rendered in either XML or JSON data formats, are shown in Annex E.
- Examples of how the CADF Query Interface and Syntax can be used, when implemented using an HTTP protocol, are shown in Annex F.

7.1.2 Requirements

The CADF Query Interface is an optional component of the CADF Specification. Implementers of the CADF Query Interface SHALL be called CADF Query Providers and they SHALL adhere to the following requirements:

- CADF Query Providers SHALL construct a result set that represents the full set of Event Records selected by the CADF Query Interface by expressing each matched event with a CADF Event Record using the CADF Resultset data type or an extension thereof.
- Each CADF Event Record in a result set SHALL be constructed according to this specification and using one of the formats described in this specification or by a profile of this specification (see Section 5.6).
- Each CADF Event Record in a result set SHALL be a valid CADF Event entity (see Section 6.6) or valid extension thereof.
- All CADF Event Records within the same result set SHALL be constructed using the same format.
    - For example, if JSON is used for one CADF Event Record, then all Event Records in the results set would be expressed in JSON. Providers are encouraged to use protocol mechanisms (such as HTTP-Accept) to negotiate acceptable formats with consumers.
- All CADF Entities SHOULD maintain referential integrity to CADF-defined entities and data types.
    - For example, all use of CADF Identifiers that identify CADF Resource-typed data within a result set should properly reference valid CADF Resource data defined elsewhere within that data set or that can be provided by some other mechanism (such as independent queries, caching, etc.).

7.1.3 CADF Query Syntax

This section describes how a filter parameter expression can be constructed to create queries using path-based expressions that reference the properties and structure of the CADF Event Record. This syntax is derived from and is compatible with both the XPath 1.0 or XPath 2.0 specifications (see bibliography for references); however, this specification does not require knowledge of either of these specifications and the CADF Query Syntax is fully explained in this section.

7.1.4 CADF Query Syntax subset

Retrieval of stored events from a provider is controlled via an optional filter parameter that is appended to a query. The $filter parameter takes the following form:

```
?filter=expression
```

Where "expression" represents a mathematical expression denoting how the top-level attributes of the resources within the collection shall be filtered. The expression is defined by the following EBNF grammar:

```
Filter       ::= Term |
                 '(' , Filter , ')or(' , Filter , ')' |
                 '(' , Filter , ')and(' , Filter , ')'
Term         ::= PropertyPath , Op , Value
PropertyPath ::= [ ComplexProp , '/' ] , SimpleProp[1]
ComplexProp  ::= ? any non-basic data type CADF property, i.e. that has
sub-properties ? |
                 ? ArrayProp with only non-basic data type elements ?
SimpleProp   ::= ? any CADF property with a basic data type ? | ?
ArrayProp with only basic data type elements ?
ArrayProp[2]    ::= Property , '[' , Index , ']'
Index        ::= '*'  |  Integer

Op           ::= '<' | '<=' | '=' | '>=' | '>' | '!='

Value        ::= "'" TypedValue "'" | '"' TypedValue '"'

TypedValue   ::= NumValue | DateValue | StringValue |
                 BoolValue | PathValue

PathValue    ::= ExactPath | PathComp | SplitPath
ExactPath    ::= ? Any CADF Path value (see Section 6.3.2) ?
SplitPath    ::= PathComp , '//' , PathComp
PathComp     ::= PathSeg [ '/' , PathSeg ] [ '*' ]
PathSeg     ::= ? Any single segment of a path corresponding to
'segment-nz' as part of a CADF Path value (see Section 6.3.2) ?
NumValue     ::= [ '-' ] Integer[3] [ '.' Integer ]
DateValue    ::= ? as defined by XML Schema ?
StrValue     ::= ? normal character string[4] ?
BoolValue    ::= 'true' | 'false'
Integer      ::= ? normal integers ?
```

[1] Here XPath syntax and this syntax diverge slightly – in XML/XPath, simple properties (e.g. attributes) would be addressed using the '@attr' syntax, but this causes a conflict with JSON representation which does not distinguish between elements and attributes in the same way. This scheme is normalized to treat all paths as simple hierarchical lists of property names which can be followed down through corresponding XML element/attribute names to match against values or through JSON properties in a similar fashion.

[2] In JSON, arrays are native objects that can be referenced by index. In XML, however, there is no native array and each element in a list will have its own element name (e.g., "reporterStep" or "item"). In XML, this construct

should be interpreted to mean “select the Nth (or all, if ‘*’ is used) element in the set of children.” This interpretation has the side effect that the child element names (such as “`reporterStep`” property) would not appear in the path.

[3] If a `NumValue` is between -1 and 1, a leading zero should be provided before the decimal point.

[4] If a `StrValue` is surrounded by double-quotes, only single-quotes may be used inside the `StrValue`, and vice-versa.

***Note**: When CADF Queries are placed in URIs/URLs, they must be URI-encoded according to RFC3968, which includes replacing spaces with ‘+’ and percent-encoding special characters.*

The choice of which operator (including '`and`' and '`or`') is limited based on the type of the value and attribute. The following describes the allowable logical and relational operators:

```
'or', 'and'                          : Boolean value/attribute, whole terms
'<', '<=', '=', '>=', ">', '!='      : Integer and date value/attribute
'=', '!='                            : String value/attribute
```

Consumers may include multiple filters within a single URI. Provider shall treat multiple filters as a series of ‘`and`’ expressions where an entry of the collection shall only be included in the response message if it satisfies all of the filter expressions specified.

When a “`filter`” is used, the collection's "`count`" attribute would contain the number of resources matching the filter expression.

7.1.5 Semantics of path values in filters

7.1.5.1 Property paths

The use of a “PropertyPath” portion (value) in a query filter shall comply with the following syntactic and semantic rules:

The path is constructed of property names indicating a containment hierarchy of related CADF entities and their included properties, and resolves to an actual value of the last property mentioned. Example:

```
/events/event?filter=target/geolocation/city='Denver'
```

In the above filter expression, “`target/geolocation`” represents the “`geolocation`” property within the “`target`” property within any CADF Event record. Similarly, “`city`” is the name of a property of the Geolocation entity identified by the “`geolocation`” property.

7.1.5.1.1 Additional Considerations

In cases where the event record uses the “`targetId`” property (of type cadf:identifier) to reference a target defined elsewhere instead of “`target`” property, then the “PropertyPath” expression SHALL still use “`target`” and the query service SHALL automatically de-reference into the cadf:resource entity wherever it was stored (effectively replacing the “`targetid`” by the actual Resource definition). This automatic dereferencing SHALL occur whenever a property with a data type of cadf:identifier is encountered while evaluating such a filter.

7.1.5.2 Arrays in a property path

When the PropertyPath value includes property names of a CADF Array type, the array notation [ ] must be used to indicate either the index of a specific item in the array, or to indicate all possible items in the array (using the wildcard ‘*’). Example:

```
/events/event?filter=tags[*]='//GRC20.gov/cloud/security/pci-dss'
```

In the above expression, any event record in the log that has "tag" property which has a value of "//GRC20.gov/cloud/security/pci-dss" will be selected and returned.

When the "PropertyPath" value includes property names of array type, it usually resolves to several possible values for the last property mentioned in the path. Example:

```
/events/event?filter=reporterchain[*]/reporterTime='2012-08-
24T23:00:00-02:00'
```

In the above expression, "reporterchain" is a property for which the type is an array of Reporterstep objects. The "reporterTime" property is then a property defined on the Reporterstep type. More generally, the path is constructed as if each item inside an array node was also a potential node in the path hierarchy. A path node that is an item inside an array is always indicated using the [ ] notation.

***Note**: In XML representation only, the property "reporterStep" is not used in the path above – it is just an item in the array which can be addressed by the index.*

When a path expression resolves to several possible values – e.g. as above if a single event has several Reporterstep objects in the "reporterchain" array, each with a different "reporterTime" value then the relational expression where this path is used will evaluate to "true" if at least one of the values satisfies the relational expression. In the above example, the filter will evaluate to "true" if at least one of the "reporterTime" values is equal to "2012-08-24T23:00:00-02:00".

7.1.5.3 Value paths

In contrast with "property" paths that are equivalent to a property symbol in the query syntax, value paths are "path values" (i.e., "PathValue" in the EBNF above), that appear always between "" (double quotes) or '' (single quotes), and are to be used as values for properties of type cadf:path. These paths typically reflect values that appear in the CADF Resource Taxonomy. For example:

```
/events/event?filter=target/typeURI='service/oss/virtualization'
```

In the above case, the value "target/typeURI" is a property path and "service/oss/virtualization" is a CADF Resource Taxonomy path. Any event that has a target RESOURCE categorized as a "service/oss/virtualization" taxonomy node SHALL be selected.

When the path value is ending with "*" (asterisk), then the path value represents a pattern where the wildcard "*" character may be substituted with any sub-path that is valid after the first part of the path. Example:

```
/events/event?filter=target/typeURI='service/oss/*'
```

In the above case, any event shall be selected that has its TARGET resource categorized as a "service/oss" taxonomy node or any node under the "service/oss" taxonomy path.

When the path value contains "//" then the path value represents a pattern where the characters "//" can be replaced with any sub-path that is valid for the context. Example:

```
/events/event?filter=target/typeURI='taxonomy/resource//database'
```

In the above case, any event shall be selected that has its target Resource categorized as an "database" taxonomy node regardless of which taxonomy sub-tree under "taxonomy/resource" (i.e. the alias for the CADF Resource Taxonomy) the "database" node belongs to (since the path segment value "database" may appear at several places in the CADF Resource Taxonomy).

7.1.6 Limiting query results using Pagination

Sometimes a provider (or server), which has large amounts of audit data needs to limit the size of returned event data to a consumer. This can be accomplished via the techniques described in this clause.

7.1.6.1 Pagination query parameters

When retrieving event records as a collection using the CADF Query Interface, consumers may include query parameters to constrain the number of entities of the collection that are returned. While the previous clause discussed how to perform a filtering on the data within the collection, this clause uses ordinal position within the collection to limit the size of the result set.

This specification defines two query parameters that, when used, shall indicate the first and last ordinal positions of the entities within the collection that are returned. The query parameters shall be of the form:

```
?limit=number
?offset=number
```

7.1.6.1.1 Additional Considerations

Where "`limit`" attribute's value indicates the (1-based positive integer) maximum number of entries in the collection to return and "`offset`" attribute's value indicates the (1-based positive integer) ordinal position of the number of entries in the collection to skip. Consumers are not required to use both at the same time. When "`limit`" is specified but "`offset`" is not, then the implied value for "`offset`" SHALL be the ordinal position of the first entity in the collection. Conversely, when "`offset`" is specified but "`limit`" is not, the value of "`limit`" is defined by the implementation.

Note: *the CADF Query Provider's endpoint (server) is not required to honor the client specified "`limit`" value; however, it SHOULD attempt to limit the number of entries returned to within the requested input parameter or a number less than that requested.*

If any part of the range as expressed by "`offset`" and "`limit`" is outside of the bounds of the collection then just the resources (if any) in the collection that are contained within that range shall be returned. A fault SHALL NOT be generated if any part, or all, of the expressed range is outside the bounds of the collection.

When either "`limit`" or "`offset`" are specified, and a filter expression (as defined above) is also specified, then the filter expression SHALL be performed first and then the ordinal constraints of "`limit`" and "`offset`" shall be applied.

7.1.6.1.2 Paginated results

The CADF Resultset schema is specified to return query results and is designed to support pagination. Partial result sets returned by a query that includes offset or limit as above must necessarily indicate the portion of the total result set that is included. These properties include:

Property	Description
count	Lists the total number of CADF Event Records included in a resultset.
nextPage	Provides a pointer to the next page in the result set's sequence.
prevPage	Provides a pointer to the previous page in the result set's sequence.
firstPage	This property will provide a pointer to the first page in the sequence.
lastPage	This property will provide a pointer to the last page in the sequence.

An example of pagination in use can be found in Annex E.

7.1.6.2 Specifying level of detail for results

The CADF Query Interface supports a "`detailLevel`" parameter that may be included in CADF Query Interface implementations to limit the set of properties returned for each event that appears in a result.

Parameter Name	Description
detailLevel	This parameter MAY be used on implementations of the CADF Query Interfaces to will limit the properties returned for each event that appears in the result set from a successful invocation of (or call to) the interface. Note: If this parameter is not present on an invocation, the CADF Query Provider MAY default this property's value to one ('1').

7.1.6.2.1 Allowed entity and data type property values by level of detail

The following table describes the valid values for the "`detailLevel`" parameter along with the CADF Event data type properties that SHALL be returned when that value is requested on a CADF Query Interface:

Table 39 – CADF Event data type properties to return based upon "detailLevel" and "eventType"

"detailLevel" value	**Value of the CADF Event's "eventType" property**	**CADF Event data type properties to include on results:**
1	activity, control, or monitor	• typeURI • id • eventType • eventTime • action • outcome • initiator, or initiatorId • target, or targetId • observer, or observerId • severity
1	monitor	• measurements
1	control	• reason
2	activity, control, or monitor	• *All properties of a detailLevel value '1' query* • reporterchain • tags
3	activity, control, or monitor	• *All properties of a detailLevel value '2' query* • measurements • reason • duration • attachments • any extended properties (by profiles of this specification)

2464 Some of the top-level properties returned on CADF queries are also complex types of their own. In these cases, the
2465 following properties of these types SHALL be included (when available) for the following `detailLevel` values:

2466 **Table 40 - Properties to return based upon CADF Type and "detailLevel"**

CADF Data Type	"detailLevel" value	Properties to include on results:
cadf:geolocation	1	• id
	2	• *All properties of a detailLevel value '1' query* • latitude • longitude • elevation • accuracy • city • state • regionICANN • any extended properties (by profiles of this specification)
	3	• *All properties of a detailLevel value '2' query* • annotations • *any extended properties (by profiles of this specification)*
cadf:reporterstep	1	• None (no level 1 properties)
	2	• role • reporter, or reporterId • reporterTime (when distinct from eventTime of the Event type)
	3	• *All properties of a detailLevel value '2' query* • attachments • *any extended properties (by profiles of this specification)*
cadf:resource	1	• id • typeURI • host
	2	• *All properties of a detailLevel value '1' query* • name • domain • credential • addresses • geolocation, or geolocationId
	3	• *All properties of a detailLevel value '2' query* • attachments

		• *any extended properties (by profiles of this specification)*

7.1.6.2.2 Detail-restricted results

In order to indicate the level of detail provided to the consumer in response to a query, the CADF Resultset schema includes a 'detailLevel' property.

Parameter	Description
detailLevel	This property includes the levels of detail (value) used by the provider when compiling CADF Event Record data included in the CADF Resultset.

Profiles that define a new type of result set should extend from CADF Log or define an equivalent mechanism.

An example of detailLevel usage can be found in Annex E.

7.1.6.3 Additional "detailLevel" parameter requirements

- CADF Event Records MAY contain properties that are optional. CADF Query Providers SHOULD return all optional properties that it is able to return when requested by the consumer. However, they SHALL NOT add properties to the results that do not have values (i.e. properties with empty or non-existent values SHALL NOT be returned)
    - For example, if a cadf:geolocation does not have a valid value for its optional "`elevation`" property, the geolocation returned SHALL NOT contain the property "`elevation`" in the result (i.e. the result would not contain `elevation=""` or `elevation=NULL`, etc.).

7.1.7 Case sensitivity

In any large-scale, distributed system that federates data from multiple providers, case sensitivity becomes a concern. Some systems are natively case-sensitive and others are not.

This raises questions when querying a federated data store that contains some data where case is important, and some data where it is not, rather complex.

Queries can either default to being case-sensitive or not:

- Case-sensitive queries may "miss" matches against resources that should be matched, if the source systems are case-insensitive but retain case in their event records (or they modify the case of the event data).
- Case-insensitive queries may have extra matches against resources that should not have been matched, e.g. that are resources distinct from the original query target.

By default, the CADF query is case insensitive and is implicit in all the other examples. An optional boolean parameter named "`casesensitive`" MAY used to explicitly set the desired case sensitivity of a given search. If the value "true" is set for this parameter, then providers SHOULD treat the search as "case-sensitive"; otherwise, if "false" is set then the provider SHOULD treat the search as "case-insensitive" (the default).

An example, of a case-sensitive query syntax for any events that contains the value "`Florida`" in the state property of any contained CADF Geolocation would appear as follows:

```
/events/event?filter=geolocation[*]/state='Florida'&casesensitive='true'
```

The CADF query API defaults to case-insensitive queries to ensure that as much data is returned as possible, which the user can then refine, or they can re-issue the query with the "`casesensitive`" parameter set to value `'true'` to force case matching. This approach is intended to ensure that data consumers can find what they're looking for even if the source system does something unexpected, although further tuning may be necessary once the data set is retrieved.

7.1.7.1 Event generation recommendations

CADF recommends the following best practices for all systems that generate events:

- If the source system (OBSERVER) is case-sensitive, then case should be retained for all events generated by the source system.
- If the source system is case-insensitive, then the source system should consistently normalize case for all generated events, regardless of what the actual input was.
- Downstream reporters should not modify the case of the data they receive and pass along.

Whether strings are uppercased or lowercased, camelcased, or some other variant may vary depending on consumer expectations - in Windows, for example, users may expect usernames to be lowercased but domain names to be uppercased by default. The purpose is not to make sure everything looks the same (e.g. everything lowercase), but to provide predictability and readability.

7.1.8 Examples using the CADF Query Syntax

The following examples show how the CADF Query syntax can be expressed as a filter string on a RESTful interface. Please note that specific format examples are included in 10ANNEX E.

7.1.8.1 Resource create query

This example shows how to construct a simple query.

When a provider is presented the following filter string, they SHOULD all CADF event records that have their "action" attribute value set to 'create' from the CADF Action Taxonomy:

```
/events/event?filter=action='create'
```

7.1.8.2 Resource creation failure query

This example shows how to construct a basic compound query.

When a provider is presented the following filter string, they SHOULD return all CADF event records that have their "`action`" property value set to '`create`' from the CADF Action Taxonomy and also have their "`outcome`" property value set to '`failure`' from the CADF Outcome Taxonomy:

```
/events/event?filter=((action='create')and(outcome='failure'))
```

Note: Any compound query is allowed as long as it conforms to the query syntax subset.

7.1.8.3 Reporter time query

To search for an event by its "`reporterTime`" attribute the following query returns the last event.

```
/events/event?filter=reporterchain[*]/reporterTime>='2012-08-
24T23:00:00-02:00'
```

The expression "reporterchain/reporterTime" is a property path that resolves to possibly several "`reporterTime`" items within a single event record, as there are several "cadf:reporterstep" type items in an event record's "`reporterchain`" property. The above expression will select any event that has at least one "`reporterstep`" with a date/time value later or equal to the value: '2012-08-24T23:00:00-02:00'.

7.1.8.4 Time window query

To search for events that occurred on or after the date '2012-07-22', the following query would return the last two events:

```
/events/event?filter=eventTime>='2012-07-22T00:00:00-02:00'
```

Complex time queries can be used to search for events within a specific time period. The follow query searches for events that occurred between the dates '2012-07-22' and '2012-07-23' (inclusive):

```
/events/event?filter=((eventTime>='2012-07-22T00:00:00-
02:00')and(eventTime<='2012-07-23T00:00:00-02:00'))
```

7.1.8.5 Taxonomy value query

To search for all events with a target resource of type equal to the CADF Resource Taxonomy value of "`resource/service/oss/virtualization`", the following query would be used:

```
/events/event?filter=target/typeURI='service/oss/virtualization'
```

To search for all events with a target resource of type equal or under the taxonomy value of "`resource/service/oss`", the wildcard "*" will indicate a path ending of any length, possibly nil:

```
/events/event?filter=target/typeURI='service/oss/*'
```

To search for all events with a target resource of type ending with "`security/profile`" yet under "`resource`", the contraction "`//`" indicates a sub-path of any length possibly empty:

```
/events/event?filter=target/typeURI='taxonomy/resource//security/profile'
```

To search for all events with a target resource of type ending with "`database`" or any type under "`database`":

```
/events/event?filter=target/typeURI='taxonomy/resource//database/*'
```

7.1.8.6 Example query using the "detailLevel" parameter

The "`detailLevel`" parameter is used to limit the size and granularity of returned events matching a specific query. A "`detailLevel`" parameter value of "1", all the attributes of the matched events are included, however contained tags, such as "`querystep`" are not returned.

For example, the following query searches for all events with "`action`" property values equal to '`create`' and specifies that all included tags such as the "`reporterchain`" property must be included.

```
/events/event?filter=action='create'&detailLevel=2
```

A similar query can be executed to include all attachments by adjusting the "`detailLevel`" parameter value accordingly.

```
/events/event?filter=action='create'&detailLevel=3
```

7.1.8.7 Result type

The default format, unless otherwise specified, of a query result type is a "`resultset`". This is implicit in all the previous examples. For example, the '`create`' search example MAY be more explicit by specifying the "`resultset`" result type as follows:

```
/events/event?filter=action='create'&resulttype=resultset
```

Vendors are free to specify additional result types as they see fit. If additional results types are specified they must be explicitly referenced directly in the query via the "`resulttype`" parameter.

Future versions of this document may specify additional result types.

8 CADF entity signing

This version of the CADF specification does not address entity signing, specifically the signing of the CADF Event, Log and Report entities. This topic may be developed in subsequent versions. It should be noted that the CADF Event, Log and Report entities were designed in a way to support (sequential) signing using the REPORTERCHAIN event component.

9 CADF profiles

Domain-specific profiles of this specification are encouraged (preferably by directly working with the DMTF CADF Working Group).

This version of the CADF specification does not provide specific guidance on how to create a profile. This topic may be developed in subsequent versions. However, the CADF WG has already identified requirements that SHALL be followed when creating profiles of this specification which are listed below.

9.1 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.
  - If a format profile is intended to be "federateable", then it SHOULD be designed to allow for lossless exchange of data when translating to other federateable formats.
- XML-based format profiles that extend this specification's XML data schema SHALL be validatable against this specification's XML data schema definition.

10 Future considerations

The CADF working group will potentially consider the following items in future versions of this specification:

- 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 **secure signing** of CADF Events, Logs and Reports.
- Additional annexes that discuss mapping of event records from other domains to the CADF standard.
- Support for indicating precision (granularity) of a CADF Timestamp.
- Provide guidance on use of metric standards for use in the CADF Metric data type (and subsequent reference within a CADF Measurement type).

ANNEX 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 annex.

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

A.1 General use of the reserved classification value "unknown"

It is acknowledged that resources that generate auditable event records will attempt to record or log an actual event even in the case where not all information is available due to perhaps some error or abnormal circumstance. In these cases, the reserved classification value of "`unknown`" is defined within each CADF Taxonomy.

A.1.1 Requirements

In terms of the CADF Event Model:

- In the case when an OBERVER (or downstream REPORTER) of an actual event is unable to identify and classify a RESOURCE, ACTION or OUTCOME (using any other valid value) at the time it generates or modifies the CADF Event Record, the reserved classification value of "`unknown`" MAY be used.

A.2 CADF Resource Taxonomy

This clause 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.2.1 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 which enables normative classification and query of events data by resource.

The logical CADF 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.2.2 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.
- An OBSERVER may have difficulty classifying one or more resources when creating the event record. In these cases, the CADF Resource Taxonomy value of "unknown" may be used as a last resort.

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 about how to classify provider resources.

A.2.3 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.2.4 Requirements

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

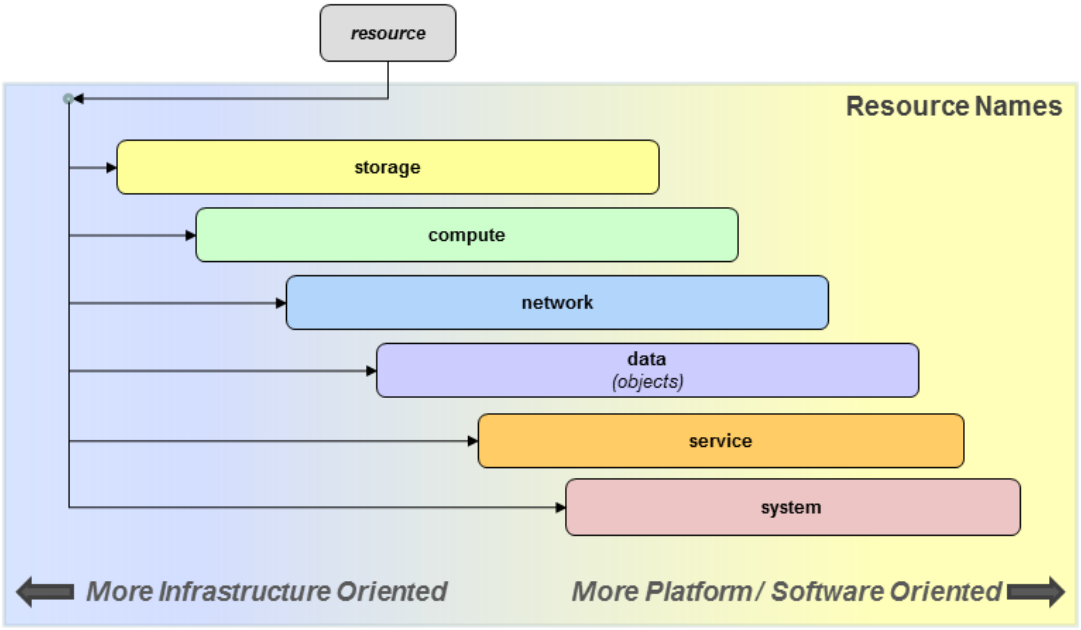
- CADF Resource typed data SHALL be classified using the CADF Resource Taxonomy, specifically as a value of its "`typeURI`" property.
    - Absolute path representation for CADF Resource Taxonomy values MAY be used anytime a value from this taxonomy is required.
    - Relative path representation for 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.
- The values of "`NULL`", an empty string or zero-length string are not valid values and SHALL NOT be used. Please
    - Please see the description of the CADF Resource Taxonomy value of "`unknown`" in the tables below for a description as to when it may be used.

A.2.5 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. Therefore, vendors and providers of CADF audit data should be careful to provide classification values that 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 above diagram attempts to convey that resources that may be named under these top-level nodes can represent resources some providers may consider more "infrastructure oriented" and offer as via an IaaS service model, whereas other providers may consider more "platform oriented" and offer them via PaaS or SaaS service models.

A.2.6 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.

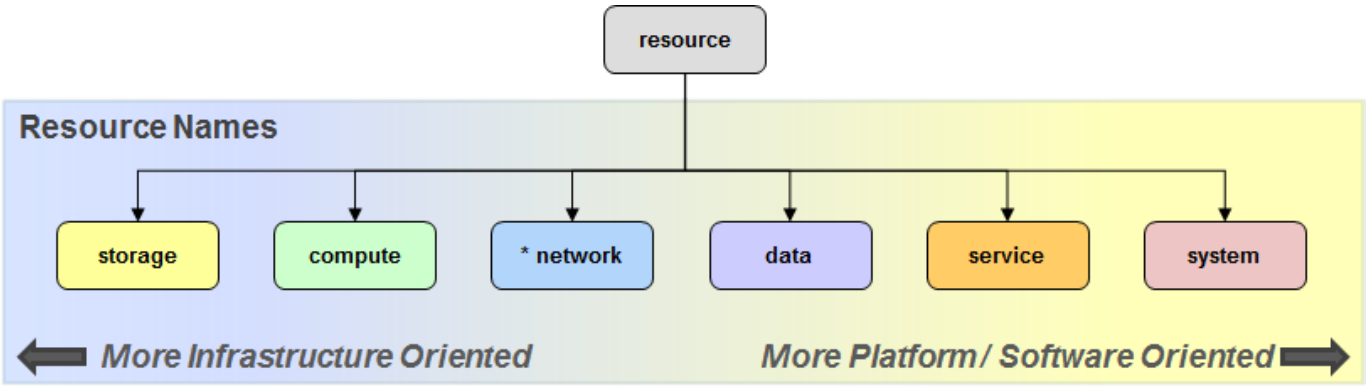
The names and descriptions for the top-level resource classifications for the "resource" taxonomy are described in Table A–1:

Table A–1 – Resource taxonomy's top-level resource classification names

Name	Description
storage	Logical resources 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.
data	Logical named sets of information (objectified data) that are referenced and managed by services.
service	Logical set of operations, packaged into a single entity, that provides access to and management of cloud resources (for a given domain).
system	Logical resources that are a combination of several other [cloud] resources that operate as a functional whole, this combination being manageable (created, operated, audited, etc.) as a unit i.e. offering some operations that could activate lower-level operations over each of the sub-resources.

Name	Description
unknown	Indicates that the OBSERVER of the event is not, to the best of its ability, able to classify a resource that contributed to the actual event it is reporting on using any other valid resource taxonomy value. For example, an OBSERVER may report an event where it is able to classify the TARGET resource, but is not able to classify the resource that was the INITIATOR of the event's action. ***Note**: This value SHOULD only be used as a last resort, and when using another classification value from the CADF Resource Taxonomy is not possible.*

2688 The following diagram shows these same top-level resource classifications as child nodes under the "resource"
2689 node of the CADF Resource Taxonomy's classification tree:



2690

2691 **Figure 12 – Top-level CADF Resource Taxonomy Hierarchy**

2692 ### A.2.7 Storage subtree classifications

2693 The names and descriptions for resource classifications that are children of the "storage" subtree are described in
2694 Table A–2:

2695 **Table A–2 – Resource classification names for the storage classification subtree**

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 waiting to be processed.

2696 The following diagram shows these same storage-oriented resource classifications as child nodes under the
2697 "storage" subtree:

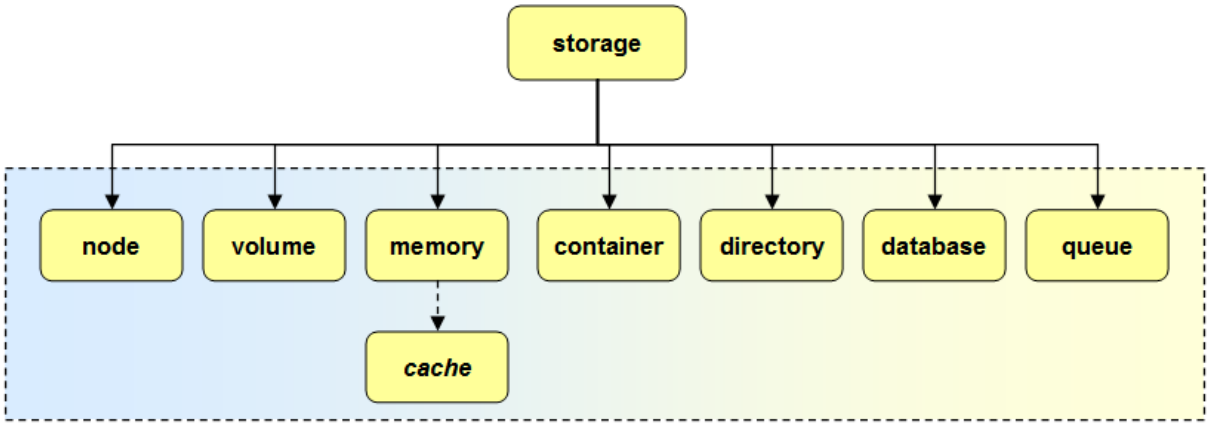


Figure 13 – CADF Resource Taxonomy - Storage subtree

A.2.8 Compute subtree classifications

The names and descriptions for resource classifications that are children of the "compute" subtree are described in Table A–3:

Table A–3 – Resource classification names for the compute classification subtree

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:

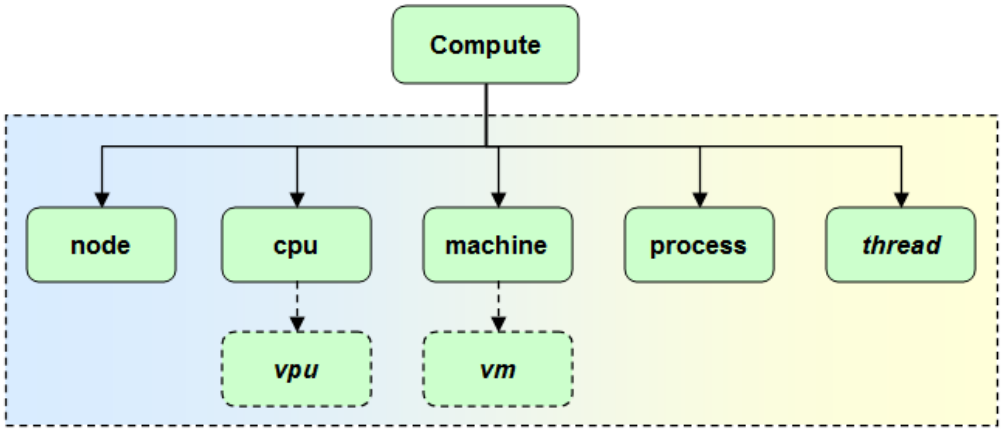


Figure 14 – CADF Resource Taxonomy - Compute subtree

A.2.9 Network subtree classifications

The names and descriptions for resource classifications that are children of the "network" subtree are described in Table A–4:

Table A–4 – Resource classification names for the network classification subtree

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.

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

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

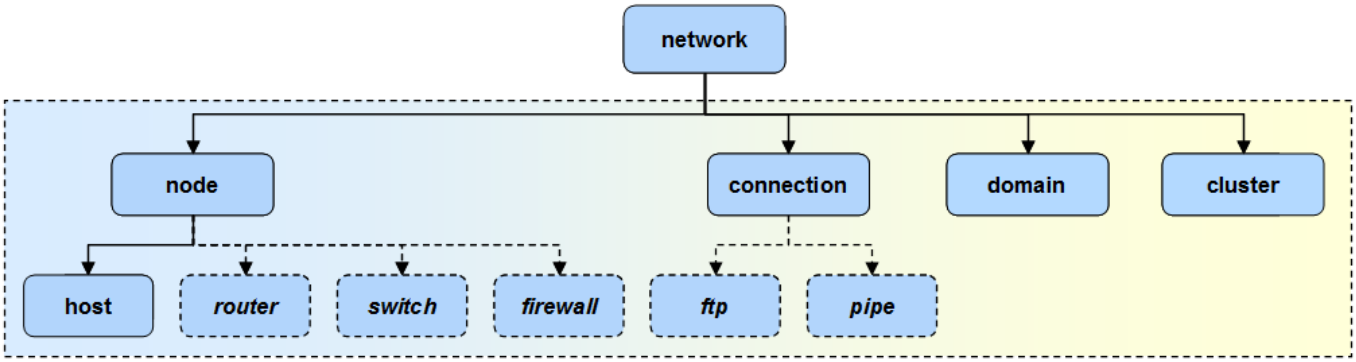


Figure 15 – CADF Resource Taxonomy - Network subtree

A.2.10 Service subtree classifications

The names and descriptions for resource classifications that are children of the "service" subtree are described in Table A–5:

Table A–5 – Resource classification names for the service classification subtree

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")

Name	Descriptive Name	Description
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:

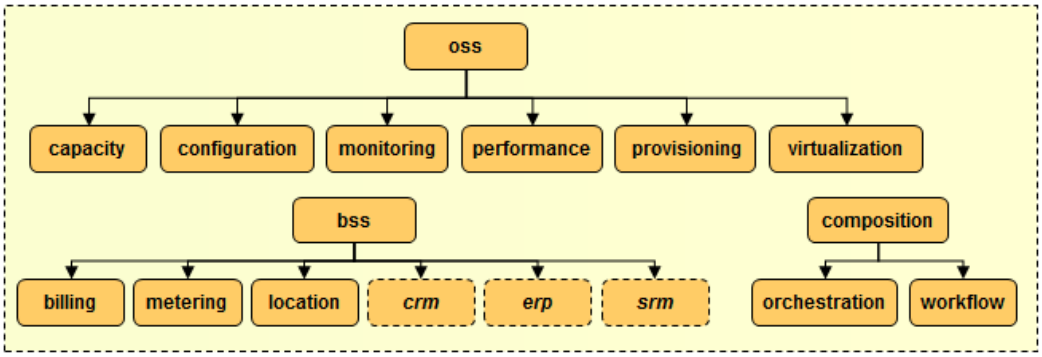


Figure 16 – CADF Resource Taxonomy - Service subtree

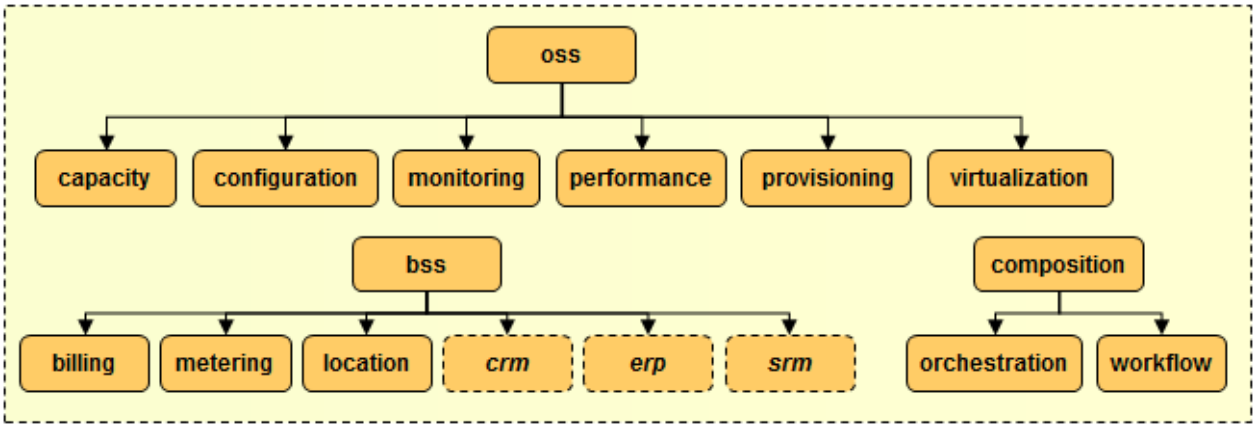
The names and descriptions for resource classifications that are children of the "oss" and "bss" subtrees are described in Table A–6:

Table A–6 – Resource classification names for the “oss” and “bss” classification subtrees

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.
logging	Operational services that capture or record information and identifying data about actions that occur in a system. This includes data that could be or contribute to auditable event records,
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.
orchestration	Composition services that automate the management of complex applications, services, platforms and/or infrastructures to align them to fulfill business and service agreements and operational policies.
workflow	Composition services that sequence connected steps that support management of a document (e.g., transaction, order, service template, etc.) through a complex system of applications, services, platforms and/or infrastructures.
crm	*Customer Relationship Mgmt. (CRM) Services (example extension of the “bss” classification)*
erp	*Enterprise Risk Mgmt. (ERM) Services (example extension of the “bss” classification)*

Name	Description
srm	*Service Request Mgmt. (SRM) Services (example extension of the "bss" classification)*

2726 The following diagram shows the Composition, Operational (OSS) and Business (BSS) Support Services subtree:



2727

2728 **Figure 17 – CADF Resource Taxonomy - BSS, OSS, Orchestration subtree**

2729 ### A.2.11 Data (objects) subtree classifications

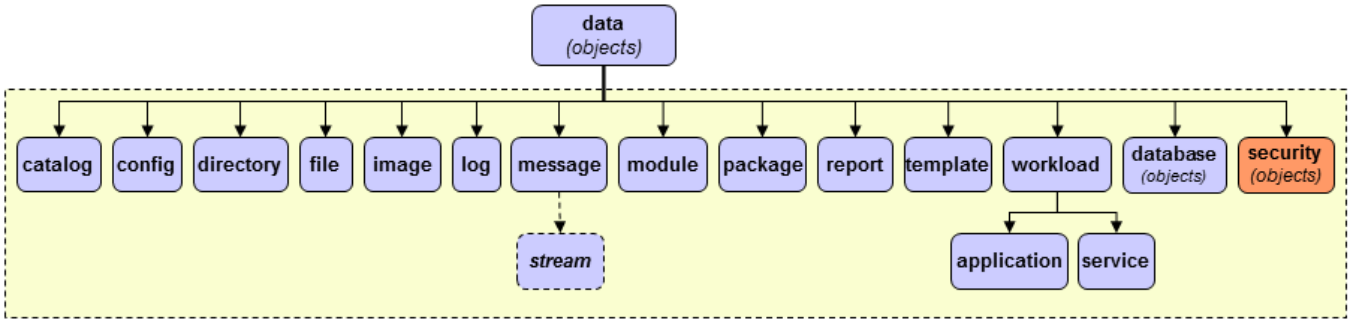
2730 The names and descriptions for resource classifications that are children of the "data" (objects) subtree are
2731 described in Table A–7:

2732 **Table A–7 – Resource classification names for the data (objects) classification subtree**

Name	Description
catalog	A data resource used to register resources along with information or metadata about them and perhaps provide links to them.
config	A data resource that contains information such as settings and parameters that could be used for configuring a resource (or parts of it).
directory	The parent classification for all directory related data objects.
file	A logical block of data for storing information, which is available to computer programs
image	A readily usable or processable set of data that can be easily transferred between processing domains.
log	A data resource 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.
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
module	A portion of a program typically aligned with a specific functional set.
package	A wrapped collection files and data, along with metadata, meaningful to the processing domain that will utilize it
report	A data resource that contains one or more event records that are compiled with other auditing information in response to some step within an auditing process.
template	A data resource that serves as a pattern, stencil or gauge for instantiating a new resource or set of resources. For example, a template that describes the topology and relationships of an application's services and its network to a cloud provider for deployment and management.
workload	A set of data that represents the amount of work that computational nodes can consume at a given time
workload/application	A workload that performs a wide range of operations, some may be exported as services

Name	Description
workload/service	A workload that perform a single or a few specialized operations. See Service subtree classifications when describing specific services in events apart from generic management as compute workloads.
database (objects)	The parent classification for all database related data objects. See the clause titled "Database (data object) subtree classifications", which shows the full set of database-related classifications.
security (objects)	The parent classification for all security related data objects. See the clause titled "Security (data objects) subtree classifications", which shows the full set of security-related classifications.

2733 The following diagram shows these same security-oriented resource classifications as child nodes under the "data"
2734 (objects) subtree:



2735

2736 **Figure 18 – CADF Resource Taxonomy - Data subtree**

2737 ### A.2.12 Security (data objects) subtree classifications

2738 The following CADF Resource Taxonomy classification nodes represent commonly expressed security data objects.
2739 The CADF Resource Taxonomy attempts to represent such security related information so that it can be
2740 consistently associated as resource data on CADF Event Records where applicable.

2741 **Design considerations**

2742 Regardless of compliance domain, a major aspect of compliance for the auditor is to verify policies that govern
2743 access to resources can be proven. It is important that representation of security information be consistent across
2744 provider deployments for auditing purposes

2745 For example, in IT systems, users or services can attempt operations on cloud resources (as INITIATORS of
2746 ACTIONS on TARGET resources) by presenting their authorization credentials. The user or services credentials,
2747 along with other context specific information, may contribute to the evaluation of security policies (and rules) to
2748 determine if access should be granted.

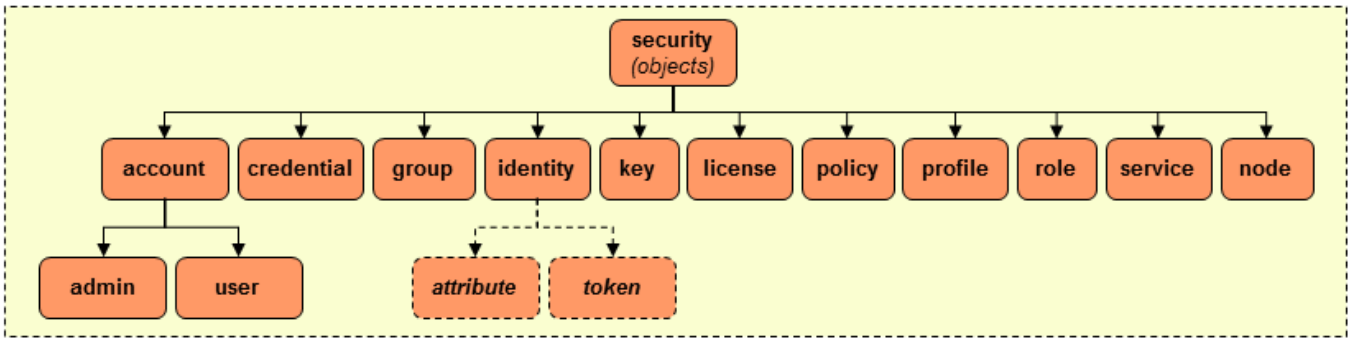
2749 The names and descriptions for resource classifications that are children of the "security" (objects) subtree are
2750 described in Table A–8:

2751 **Table A–8 – Resource classification names for the security (objects) classification subtree**

Name	Description
account	Represents a business agreement for providing regular services between a provider and consumer.
account/user	An account representing a person assigned access to use cloud resources or applications.
account/admin	An account representing a person assigned administrative access to resources.
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.

Name	Description
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).

2752 The following diagram shows these same security-oriented resource classifications as child nodes under the
2753 "security" (objects) subtree:



2754

2755 A.2.13 Database (data object) subtree classifications

2756 **Figure 19 – CADF Resource Taxonomy - Security subtree**

2757 described in Table A–9:

2758 **Table A–9 – Resource classification names for the database (objects) classification subtree**

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 that uniquely identifies records.

Name	Description
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:

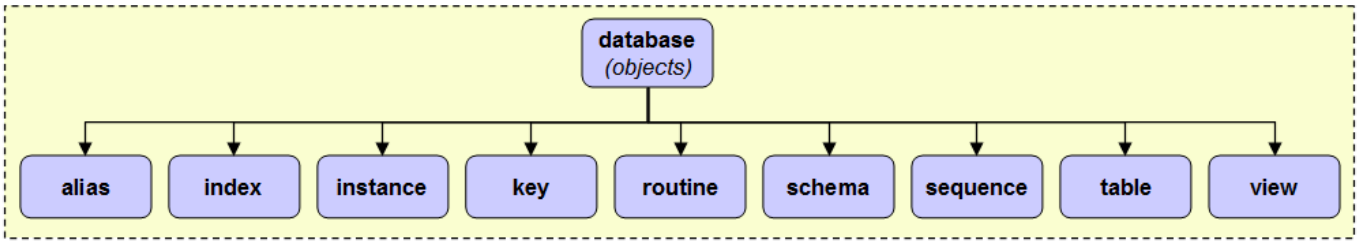


Figure 20 – CADF Resource Taxonomy - Database subtree

A.2.14 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.

Table A–10 includes examples of valid CADF Resource Taxonomy values as expressed in their relative and absolute URI forms:

Table A–10 – CADF Resource Taxonomy values expressed in 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.3 CADF Action Taxonomy

This clause 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.3.1 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 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.3.2 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 about how to classify provider resources.

A.3.3 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.3.4 Requirements

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

- 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.
- The action value "`monitor`", or a valid extension of this value, SHALL be used for all CADF Event Records classified as type **monitor**.

- If the CADF Event Record's property "`eventType`" is set to type **control**, then the same event's "`action`" property value SHALL be one of "`allow`", "`deny`", "`evaluate`", "`notify`" from the CADF Action Taxonomy (or a value that is a valid extension of one of these).

A.3.5 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. The taxonomy values are loosely tied to the base event types as defined by the CADF Event model.

In designing the taxonomy for **activity** type events, the CADF has acknowledged the widely accepted use of "CRUD" operations (i.e., "`create`", "`read`", "`update`" and "`delete`") as typical action values used in cloud management platforms and similar IT domains. These action values are supported for classifying actions taken on any TARGET resource as classified by the CADF Resource Taxonomy. 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.

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.

The taxonomy values for **control** type events are similarly focused on the specific activities involved in policy decisions, including "`allow`," "`deny`," "`evaluate`," and "`notify`." Generally these control type events would be correlated with related action type events that describe the underlying activities that caused the policy to be applied.

The following color key indicates how actions in the taxonomy (as displayed in the tables below) may pertain to certain logical management and operational categories:

Table A–11 – CADF Action Taxonomy informal grouping color key

Color	Informal Classification Grouping
Lt. blue	General resource management (i.e. CRUD operations)
Blue	Monitoring
Green	Workload and data management
Purple	Messaging actions
Orange	Security – Identity
Yellow	Security – Policy / Access Control

Table A–12 lists the CADF Action Taxonomy's values along with their definitions:

Table A–12 – CADF Action Taxonomy values

Informal Grouping	Value	Description
General Resource Mgmt.	**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.

Informal Grouping	Value	Description
Monitoring	**monitor**	The target resource is the subject of a monitoring action from the initiating resource.
Workload and Data Mgmt.	**backup**	The target resource described in the event is being persisted to storage without regard to environment, context or state at the time of storage.
	capture	The target resource described in the event is being persisted to storage along with relevant environment and state information (e.g. program settings, network state, memory/cache, etc.). Conceptually, a “snapshot” of the resource is being captured at a moment in time.
	configure	The target resource described in the event is being set-up to enable it to run on a particular environment or for a particular application or use.
	deploy	The target resource is being positioned or made available for use by the initiator resource, but not yet started.
	disable	The initiator resource is causing the target resource [that has been started] to disallow or block some set of functions.
	enable	The target resource (that has been started) is being changed by the initiator resource to allow or permit some set of functions.
	restore	The initiator is requesting the target resource (or some portion of it) be restored from persistent storage.
	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.
	undeploy	The initiator resource is causing the target resource to no longer be positioned or available for use.
Messaging	**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.
	send	The initiator resource is transmitting a message or data to the target resource. **Note**: this is a separate action from that of "creating" the message.
Security - Identity	**authenticate**	A security request used to establish an initiator’s identity and/or credentials to the target resource against a trusted 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.
	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).
	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).
Security – Policy, Access Control	**allow**	Indicates that the initiating resource has allowed access to the target resource.
	deny	Indicates that the initiating resource has denied access to the target resource.
	evaluate	The evaluation or application of a policy, rule, or algorithm to a set of inputs.
	notify	Indicates that the initiating resource has sent a notification based on some policy or algorithm application – perhaps it has generated an alert to indicate a system problem.
	unknown	Indicates that the OBSERVER of the event is not, to the best of its ability, able to classify the exact action for the actual event it is reporting using any other valid action taxonomy value.

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

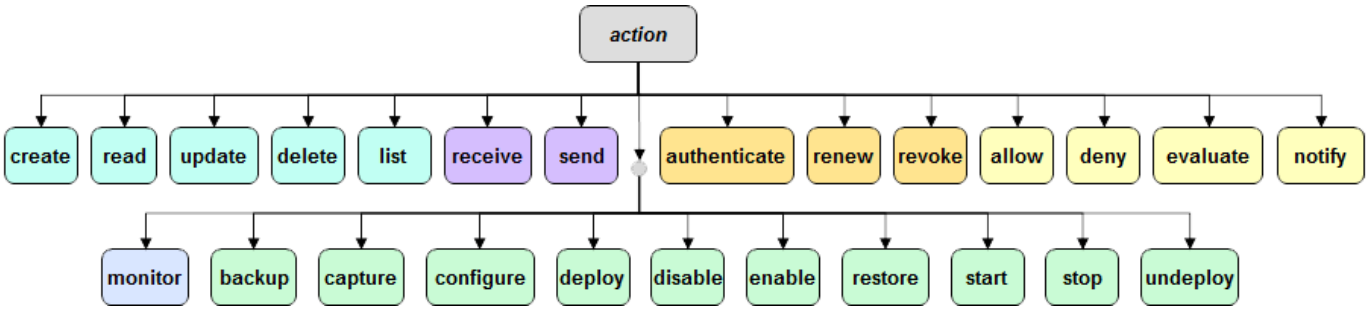


Figure 21 – CADF Action Taxonomy Hierarchy

A.3.6 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.3.7 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 data 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.

Table A–13 includes examples of valid CADF Action Taxonomy values as expressed in their relative and absolute URI forms:

Table A–13 – CADF Action Taxonomy values expressed in 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.4 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.4.1 Design considerations

General considerations

This version of the outcome taxonomy is designed to support the following Design considerations that have been derived from use cases the CADF examined in DSP2028.

- 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 that 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.4.2 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.4.3 Requirements

The following requirements are for 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.4.4 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:

A.4.5 Taxonomy values

The CADF Outcome Taxonomy provides the following "root" outcome values that SHALL be used for any

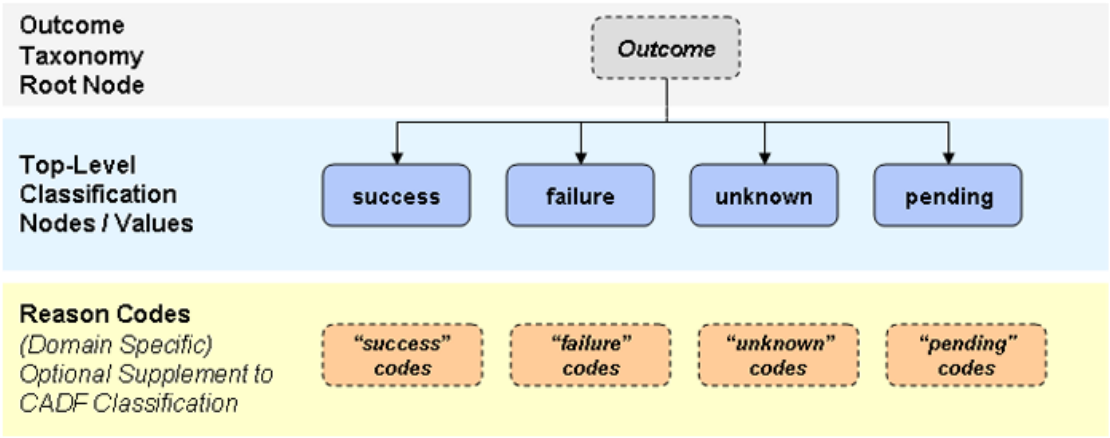


Figure 22 – CADF Outcome Taxonomy Hierarchy

extensions or profiles of this specification. They are shown in Table A–14:

Table A–14 – CADF Outcome Taxonomy "root" outcome values

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. • ***Note***: A different (future) event correlated with the current event may provide additional detail.

A.4.6 Requirements

The following requirements are for 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.4.7 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 data 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:

Table A–15 – CADF Outcome Taxonomy values expressed in 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/unknown
pending	http://schemas.dmtf.org/cloud/audit/1.0/taxonomy/outcome/pending

A.4.8 Considerations when using "unknown" or "pending" values for action classification

- 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. Because 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.5 Treatment of INITIATOR, TARGET, and OBSERVER

A.5.1 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.5.2 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 with 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.

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.5.3 Treatment of TARGET

Any CADF Resource can appear as the TARGET within a CADF Event Record, because 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.5.4 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.6 Using the CADF Taxonomies to create CADF Event Records

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

A.6.1 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 of what domain resource objects does it have direct knowledge?".
   – 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.
   – 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)?
   – 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.
   – 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.
   – 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.6.2 Example: Account creation

A 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 of what domain resource objects does it have direct knowledge?"
   - 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., the CADF Resource Taxonomy value "`data/security/account/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" that 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.6.3 Example: 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 of what domain resource objects does it have direct knowledge?”.
   - 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`".

ANNEX B
Best practices

B.1 Treatment of “extra” contextual event data

As with any pre-defined schema that assigns semantic meaning to given pieces of data, there are inevitable use cases that generate data that does not quite fit into the pre-defined CADF Event Schema. To ensure continued support for such use cases, CADF has defined several Extensibility mechanisms that allow the inclusion of that additional data, plus support for profiles that can more formally define extended schema elements and values.

This section describes some common, known use cases that are out of scope for the core CADF specification and Event Schema, but can be used to describe how such data could be handled.

B.1.1 Use case: Debug Information

In general, it is not best practice to include debug information (such as stack traces and variable state reporting) within audit event records and therefore it was listed as “out of scope” for this specification.

However, it is noted that in some contexts, “debug” type events are extremely common across many types of applications and services and are often intermixed with normal events in logs. The defining characteristic of a debug event is that it generally indicates a fault in software and includes information about the specific point in the code that experienced an issue, such as a stack trace.

In order to include such information within a CADF Event Record, the generator of the debug information could use the Attachments extension mechanism and include any necessary data. It should be noted, however, that downstream consumers may choose to strip off event attachments, so interpretation of the basic event should not be predicated on the attachment(s).

B.2 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 other events temporally. For example, events captured within an enterprise that has employees that 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 the potential complications of multiple time zones and time zone 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 time zone 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.

3101 **Table B–1 – CADF Timestamp data type properties**

CADF Timestamp Properties		
Parent Entity Name	**Property Name**	**Property Description**
CADF 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).
CADF Log	beginTime	The beginning time for the time period of event records within the log.
CADF Log	endTime	The ending time for the time period of event records within the log.
CADF 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).
CADF Report	beginTime	The beginning time for the time period of event records within the report.
CADF Report	endTime	The ending time for the time period of event records within the report.
CADF Event	eventTime	The OBSERVER's best estimate as to the time the Actual Event occurred or began. (Note that this time may differ significantly from the time at which the OBSERVER is processing the CADF Event Record).
CADF 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).

3102 ### B.2.1 Filling in Timestamps

3103 Within a single event, multiple timestamps may be present. These different timestamps serve different purposes,
3104 and should be filled in by the Reporters based on the intended use of that field:

3105 • The "`eventTime`" property field in the base CADF Event data type represents the OBSERVER's best guess
3106 as to the time that the observed activity actually occurred. In cases where the OBSERVER is also the
3107 INITIATOR this should be relatively simple, but in more complex cases the actual time of occurrence might be
3108 significantly removed from the time of observation.

3109 ○ For discrete, point-in-time observations generally speaking the "`eventTime`" field should reflect the
3110 current time according to the OBSERVER's local clock, and is the only required time field.

3111 ○ For complex activities that have some duration, if the OBSERVER can determine the true starting
3112 time of the activity and insert that time into the CADF Event's "`eventTime`" property, that is
3113 desirable. In this case the "`eventTime`" property may differ significantly from the
3114 "`reporterTime`" of the OBSERVER, hence both fields should be provided.

3115 • For CADF Reports and Logs, the service that assembles the output can either determine the "`beginTime`"
3116 and "`endTime`" property values based on the events within the output set, or based on the query (see section
3117 titled "CADF Query Interface").

3118 ○ If the query specifies a specific time range or starting/ending point in time, then either or both the
3119 beginTime and endTime can be filled in with that timestamp, even if there are no events that actually
3120 took place at that time.

3121 – For example, if the requester asked for events between 1:00 PM and 2:00 PM, but only two
3122 events took place at 1:33 PM and 1:35 PM, the CADF Report or Log could still indicate a
3123 "`beginTime`" property value of 1:00 PM and an "`endTime`" property value of 2:00 PM.

- If the query does NOT specify a beginning or ending time to search, the CADF Report or Log can fill in that value with the earliest (for the "`beginTime`" property) or latest (for the "`endTime`" property) timestamp present in the set of output events.
- In no case should any event within the CADF Report or Log have an "`eventTime`" property value outside the range specified by the properties "`beginTime`" or "`endTime`" of the CADF Report or Log.

B.2.2 Handling Activities with Duration

Many activities that are represented in event records in IT systems are discrete, point-in-time actions that are for all intents and purposes instantaneous and are recorded as such. Even in cases where the activity actually takes some period of time, the time period is often brief enough that the only relevant timestamp for consumers is the time the activity started. As such, CADF Event Records contain a top-level eventTime that is defined as the time at which the described activity began.

In some cases, however, described activities do in fact take some lengthy period of time, and further some consumers may be very interested not only in when the activity began, but also when it ended or how long it took. Examples include activity such as long-running queries or backups, login sessions, and so on. In this scenario, OBSERVERS have several options:

- The OBSERVER can delay generating the event record until the activity has completed, and then fill in relevant information about activity duration and issue the event record. This approach has the major drawback that if a consumer queries for recent activity during the time period in which the OBSERVER is holding on to the record, the consumer will not be aware of the activity. This approach is therefore only recommended for activity of relatively short duration and where the acknowledged completion of the activity is virtually guaranteed.
- The OBSERVER could generate an event record to describe the start of the activity, store it someplace, and then go modify the event record when the activity is completed to add relevant information about the activity duration. This approach however is heavily implementation-dependent, violates several important assumptions about event immutability, and is not recommended for any implementation.
- The OBSERVER can generate an event record to describe the start of a long-running activity, and then generate a second event to mark the end of that activity. This is the approach recommended by the CADF WG for most lengthy activities, and is described below.
- The recommended approach involves the OBSERVER issues a matched pair of begin/end events to mark the start and end of the described activity. CADF Event Records include a number of features to support this:
  - The start event should describe the activity as usual, with the eventTime field recording the start time and all other properties set as usual, except for OUTCOME. The OUTCOME property should be set to "pending" per the definition of the taxonomy. A tag should be set with a correlation ID so that the event pair can be associated.
  - The end event should be a near-duplicate of the start event, except that OUTCOME should be resolved to the actual outcome of the activity, and the 'duration' property should be set. The start and end events should be correlated by use of a correlation ID as described in Annex B.3.4.

B.3 Handling Complex Events

There are many scenarios where the representation of an actual event or a set of events in terms of CADF event record(s) is not straightforward:

- An event describes a target, but the context of that target is important: for example, a file is deleted but consumers need to know which directory and host the file was located on.

- A single actual event may by definition affect more than one resource: for example, when a user account is added to a group, both the user account and the group are affected.
- A single action may cause many nearly identical actual events: for example, if a set of files are deleted from a directory.
- A single action may cause many related actual events: for example, a complex system is deleted.
- An event may represent some form of request, which should be associated with its corresponding response(s): for example a database read request may result in multiple result sets.
- An action may trigger a reaction: for example, an attempted connection from one host to another may trigger a firewall block.
- A set of events may be modeled or summarized as a single event: for example, a complex sequence of authentication, authorization, and session creation events may be treated as a single access request.

This section will set forth some best practices for handling such complex scenarios. These best practices are not prescriptive and are subject to the perspective of the observer and the expectations of the consumer of audit events

B.3.1 Resource Context

In most scenarios, the context within which a resource lives is very important for determining the relevance and impact of a particular event. The directory within which a file resides, which host those resources live on, the container for a particular user account – a security team might make a very different decision on how to handle an event if they know that the account 'juser1' resides in the 'executive_team' container versus the 'external contractor' container. The basic CADF Event Record includes an entity to describe the singular target resources affected by the actual event – how should this additional context be included?

As a best practice, consider using the Attachment entity (as opposed to a user-defined extension attribute). to include this context data. However it must be decided whether to use the per-resource 'attachments' property (as defined on the Target resource of an Event) or the 'attachments' property of the Event itself. As a general rule:

- If the context information is really dependent on the resource itself and not contingent to the event, use the resource 'attachments' property. For example, if the resource is part of a container resource – e.g. a catalog to which the resource item belongs – then this container resource may be represented or referred to in an attachment of the contained resource.
- If the context information is really contingent to the event and is not associated with the event resource (target of initiator) in a permanent or stable way, then the 'attachments' property of the event should be used. For example, if the resource is a file being transferred from one directory to the other, then the origin and destination directories can be seen as contextual to the event itself and attached to the event instead of being attached to the target resource (the transferred file).

Any type of context may be included – additional resources, measurements, geolocations , and so forth – that will help consumers understand the event more fully.

- If you plan to use the CADF schema to describe the attached context data, use the appropriate CADF type URI as the attachment 'typeURI'
- Use a descriptive name to describe how the attached context data relates to the parent resource as the attachment 'name' property. The name should ideally be a commonly-understood keyword and/or map to existing specifications, such as DMTF CIM.

XML example

```
<event id="myscheme://mydomain/id/1234">
   ...
   <target id="..." typeURI="..."/>
```

```
   ...
   <attachments>
      <attachment contentType="
      http://schemas.dmtf.org/cloud/audit/1.0/resource" name="hostedOn">
         <content>
            <resource id="myscheme://mydomain/resource/id/0001"
                      typeURI="network/node/host"
                      name="server_0001"
                      ref="http://mydomain/mypath/server-0001"/>
         </content>
      </attachment>
   </attachments>
</event>
```

In the above example, the target resource of an event is hosted on the host described by the attachment.

B.3.2 Multi-Target Events

Another class of events will always affect more than one resource even if the activity is described at the most granular level. An example includes adding a user account to a group – both the user account and the group are affected, and the event cannot be decomposed into two independent parts. In this scenario, deciding whether to set the user account or the group as the target of the event is purely a matter of choice, and will affect the consumer's understanding of the activity plus the ability to query for relevant activity. For example, if the implementer chooses to set the user account as the target, consumers wishing to know who was added to a particular group will find it difficult to query for that information; the opposite choice will make it difficult to query for a particular user's group membership history.

To resolve this dilemma, **multiple** CADF event records may be generated that describe the activity from each perspective: for the example given, one event would set the user account as the target resource and the group information would be included as context (event attachment); a second event would set the group as the target resource and include the user information as context (event attachment).

To ensure that these events are properly understood as different viewpoints on the same actual event, each event should be tagged with an identical **correlation identifier** (see B.3.6) so that the events can be associated.

Consumers may of course choose to combine these multiple events into one record for storage, and a profile of this specification may prescribe a particular method for generating tag names and correlation identifiers, but for general-purpose implementations this best practice will ensure maximal comprehension.

XML example

```
<!-- Event 1 -->
<event id="myscheme://mydomain/id/1234"  action=”associate“>
   ...
   <target id="myscheme://mydomain/resource/id/0001"
      name="user01" typeURI="data/security/account/user" />
   <attachments>
      <attachment
```

```
      contentType="http://schemas.dmtf.org/cloud/audit/1.0/resource"
      name="parent">
         <content>
            <resource id="myscheme://mydomain/resource/id/0002"
            name="group01"
            typeURI="data/security/group"/>
         </content>
      </attachment>
   </attachments>
   <tags>
      <tag>//myobserver/correlationID?value=1234</tag>
   </tags>
</event>

<!-- Event 2 -->
<event id="myscheme://mydomain/id/1235" action="associate">
   ...
   <target id="myscheme://mydomain/resource/id/0002"
      name="group01"
      typeURI="data/security/group" />
   <attachments>
      <attachment
      contentType="http://schemas.dmtf.org/cloud/audit/1.0/event/resource"
      name="member">
         <content>
            <resource id="myscheme://mydomain/resource/id/0001"
            name="user01"
            typeURI="data/security/account/user"/>
         </content>
      </attachment>
   </attachments>
   <tags>
      <tag>//myobserver/correlationID?value=1234</tag>
   </tags>
</event>
```

Notes: *In the above example, the contextual information in each event is represented as an attachment of the event itself and not of its target resource. Although these two resources (user and group) are now tightly associated, this association is considered here as a property of the activity reflected by the event (adding the new user account to the group) more than an intrinsic property of the resource itself.*

This user account could later be removed from the group, and associated with another group. In that case it is more obvious that the "group" data should not be associated with the user resource (and vice versa): an event log may indeed decide to describe user resources and group resources in a "reusable" way at log level and have events only refer to these using their "targetId" property. In such a case, it is clearer that the contextual information should be attached to the event rather than to the target.

B.3.3 Multiple Affected Targets

In this scenario, a single user or service action impacts multiple targets, but the action is decomposable into multiple events. A typical example here would be the deletion of all files in a subdirectory – from a user perspective, this is one action; but from the system perspective there is a chain of multiple individual deletes.

Introducing a complex multi-target construct such as an array of file references as attachment to the “subdirectory” target resource or as attachment to the event itself would negatively affect a user’s ability to query such events. The best practice in this area is to issue an individual CADF Event Record for each system level action that affects a singular target. As with the intrinsically multi-target event, best practice is to use a correlation identifier as a tag to tie the individual events together so that the consumer can optionally understand them as one transaction:

XML example

```
<!-- Event 1 -->
<event id="myscheme://mydomain/id/1234" action=”delete” >
   ...
   <target id="myscheme://mydomain/resource/id/0001"
      name="file01.txt" typeURI="data/file" />
   <tags>
      <tag>//myobserver/correlationID?value=1234</tag>
   </tags>
</event>

<!-- Event 2 -->
<event id="myscheme://mydomain/id/1235" action=”delete” >
   ...
   <target id="myscheme://mydomain/resource/id/0002"
      name="file02.txt" typeURI="data/file" />
   <tags>
      <tag>//myobserver/correlationID?value=1234</tag>
   </tags>
</event>
```

***Note**: This concept applies equally well to actions over complex targets with multiple unlike resources, for example the deletion of a cloud system consisting of a host, network, and storage.*

B.3.4 Request-Response Events

A common paradigm in computing is the request/response paradigm, where one resource requests some service from another resource. In some cases this activity can be treated atomically – one is unlikely to decompose a filesystem delete into separate requests and responses to/from the filesystem driver, for example – but in other cases with loosely-coupled asynchronous APIs and long-running transactions activity might be better modeled as paired request/response events.

Treatment of this type of activity is similar to the multiple-target events listed above, with multiple events related by a correlation identifier tag. In this case, however, the actions will be different between the two events: here is a send/receive example:

XML example

```
<!-- Event 1 -->
<event id="myscheme://mydomain/id/101" action=”send”
   initiatorId=”myscheme://mydomain/myself”>
   ...
   <target id="myscheme://mydomain/resource/id/0001"
      typeURI="service/oss/provisioning" />
   <tags>
      <tag>//myobserver/correlationID?value=1234</tag>
   </tags>
</event>

<!-- Event 2 -->
<event id="myscheme://mydomain/id/102"
   action=”receive”
   initiatorId=”providerscheme://pdomain/providerXYZ”>
   ...
   <target id="myscheme://mydomain/resource/id/0001"
      typeURI="service/oss/provisioning" />
   <tags>
   <tag>//myobserver/correlationID?value=1234</tag>
   </tags>
</event>
```

***Note**: In the example shown above, the observer is the system making the request; the system receiving the request may generate its own pair of related events to describe the same activity.*

It is relatively easy for a single observer to tie related events together with a correlation identifier, but only in rare cases is it simple to correlate the events generated by the requestor with the requestee – only a very few APIs explicitly call for passing session identifiers between the two parties.

As a best practice, requestors and requestees should annotate generated CADF Event Records with as much state information as they can to describe the session – for example, a web service could record the source IP and port of an inbound request. This could allow a consumer to connect the requestor event (which hopefully records the same or similar information) with the requestee event.

B.3.5 Action-Reaction Events

This paradigm is similar to the request-response paradigm, but the initiating resource is not directly making a request of the system that reacts. An example would be one host attempting to connect to another host, which is then subsequently blocked by a third party, perhaps a firewall.

In this case, the resource that blocks the activity will likely generate a ‘control’ type event to describe the connection that it blocked. The ‘control’ event, however, describes only the resource making the control decision and the characteristics of the activity that was blocked, it does not necessarily describe the activity that triggered the policy decision in the first place. Sometimes this information can be gleaned from other observers in the environment, but in simple cases the control resource may also issue an ‘activity’ event in addition to the ‘control’ event, and relate the two using a correlation identifier:

3277 **XML example**

```
<!-- Event 1 -->
<event id="myscheme://mydomain/id/101" eventType="activity"
   action="connect">
      <initiator id="myscheme://mydomain/resource/id/0001"
         typeURI="network/node/host" name="host01" />
      <target id="myscheme://mydomain/resource/id/0002"
         typeURI="network/node/host" name="host02" />
      <tags>
         <tag>//myobserver/correlationID?value=1234</tag>
      </tags>
</event>

<!-- Event 2 -->
<event id="myscheme://mydomain/id/102" eventType="control" action="deny">
   <initiator id="myscheme://mydomain/resource/id/0003"
      typeURI="network/node/firewall" name="fw01" />
   <target id="myscheme://mydomain/resource/id/0004"
      typeURI="network/connection" name="10.0.0.2:1234-192.168.4.3:8080"
      />
   <tags>
      <tag>//myobserver/correlationID?value=1234</tag>
   </tags>
</event>
```

3278 ### B.3.6 Correlated Events

3279 Any set of events could be loosely correlated to describe a relationship between them. This may involve events
3280 from one or more observers, or may involve correlation internal to the observer, or performed by a third-party
3281 system. Third-party tools such as Security Information and Event Managers (SIEM) may issue synthetic events
3282 which describe or summarize the activity that is believed to be indicated by the set of related events. In this
3283 scenario, the various raw events that are tied together by the correlation event may involve different event types,
3284 actions, and resources.

3285 One way to correlate events is to introduce explicit **correlation identifiers** in forms of tags. A correlation identifier is
3286 domain-specific to the observer generating the CADF Event Records, and should be namespaced accordingly. A
3287 descriptive name for the tag that includes the string ‘correlation’ somewhere in the tag name may help consumers to
3288 interpret it effectively, although in many cases a particular tag is known to act as a correlation ID, e.g. the instance
3289 ID of a business process will correlate all events generated by the process engine for this process instance.

3290 Multiple events with identical tags the name of which is known to indicate a “correlation” tag,’ may generally be
3291 interpreted as belonging to a single related activity.

3292 **Examples:**

```
<event id="myscheme://mydomain/id/1111">
```

```
   ...
   <tags>
      <tag>//myobserver/correlationID?value=1234</tag>
      <tag>//businessProcessXYZ/instanceID?value=1111</tag>
   </tags>
</event>
```

Another more explicit correlation means is by using attachments.

The suggested implementation uses a simple list that refers to a set of correlated CADF Event Records by reference. Such a list of event IDs or references may be attached (see CADF Attachment type) to an event, indicating that this event is correlated with all the referred events.

XML example

```
<event id="myscheme://mydomain/id/1111">
   ...
   <attachments>
      <attachment
         contentType="http://schemas.dmtf.org/cloud/audit/1.0/identifier"
         name="correlatedEvent1">
            <content>myscheme://mydomain/event/id/1234</content>
      </attachment>
      <attachment
         contentType="cadf:identifier"
         name="correlatedEvent2">
            <content>myscheme://mydomain/event/id/5678</content>
      </attachment>
   </attachments>
</event>
```

In this example, the described event is related to the several events listed in the attachment; those events are defined elsewhere, perhaps within the same or in another CADF Log or Report.

ANNEX C
Mapping DMTF CIM Indications to CADF Event Record

This section provides guidance on how DMTF's CIM standard's event type named "CIM_Indication" would, in general, map to a CADF audit event record.

The event type associated with CADF event records communicates audit information.

The record of a particular type is an indication of a specific event. This concept is conceptually related to an abstract class:CIM_Indication in the Common Information Model. CIM_Indication is an abstract class from which a CADF event is derived. CADF events are modeled as CIM indications to leverage key features described in CIM and supported in the industry.

As described in CIM Indication, DSP1054, an Indication is a "communication and record of the detection of an event of interest." The Indication may be an aspect of or the event itself. Indications are defined in a profile where CIM_Indication properties are found. In general, an instance of an indication type derives from CIM_Indication.

Similar to CADF event types, many Indications may be associated with an event. An Indication logically relates to the REPORTER that observes or initiates an event action on a resource. The key elements defined in the CIM_Indication abstract class relate to that of a CADF event type. For example, elements of the abstract CIM_Indication class relate to basic CADF event type properties such as '`eventTime`', '`initiator`', '`initiatorId`', and '`severity`'.

The construction of Indications and its relationship to CADF are not described here. The purpose of identifying this relationship is to promote consistency between the CIM and CADF concepts rather the mechanics used to implement them.

C.1 Informative References:

- CIM Indication Schema (.xsd) in CIM 2.3.5 (final):
  - http://dmtf.org/sites/default/files/cim/cim_schema_v2350/cim_schema_2.35.0Final-XSDClasses.zip
- DSP1054 Indication Profile 1.2.1:
  - http://dmtf.org/sites/default/files/standards/documents/DSP1054_1.2.1.pdf

The DSP0227 WS-MAN CIM Binding Specification provides several examples and scenarios where Indication instances and events are used. For example, a management client receives specific indications from a device being managed.

A service may internally create CIM indication-related instances when the service accepts a subscription using the Subscribe message from a Web services client.

- http://dmtf.org/sites/default/files/standards/documents/DSP0227_1.2.0.pdf

ANNEX D
Mapping DMTF CIMI Events to CADF Event Records

This section provides guidance on how DMTF's CIMI standard's event type would, in general, map to a CADF audit event record.

CIMI events are generated during operations of an IaaS provider that complies with Cloud Infrastructure Management Interface (CIMI, [….]).CIMI events may have audit relevance and need to be translated into CADF Event Records, A CIMI provider will typically keep a record of CIMI events concerning a CIMI resource, in an EventLog resource associated with this CIMI resource. The translation into a CADF Event may require using information from both the CIMI event and the CIMI EventLog resource.

***NOTE**: The mapping defined here only defines foundational rules that any event mapping from CIMI to CADF are expected to follow. However in many cases, these rules are not sufficient and should or may be complemented by additional rules that are left for users to agree upon (e.g. via a mapping profile). When the mapping rules below are insufficient to handle the mapping of a particular item and opportunities exist for user-defined additional rules, this will be indicated as an "extensibility" point.*

The following notation is used:

```
<specification prefix> ":" <object> "." <attribute> [ "."
<subattribute> ]
```

For example, "cadf:event.id" means: the "`id`" property attribute of a CADF Event record.

D.1 Recommended mapping rules

The recommended mapping rules to generate a CADF Event Record (by attribute) from a CIMI Event are:

D.1.1 cadf:event.id

Here the mapping does not recommend a particular ID scheme. The CIMI event URI may just be imported as the CADF Event's "`id`" property, or the latter may be left for the migration function to generate.

D.1.2 cadf:event.eventType

There are four predefined values for CIMI:Event.type, which map to the following "cadf:event.eventType" property:

- CIMI:Event.type = "state" → cadf:event.eventType = "monitor"
- CIMI:Event.type = "alarm" → cadf:event.eventType = "control"
- CIMI:Event.type = "model" → cadf:event.eventType = "activity"
- CIMI:Event.type = "access" → cadf:event.eventType = "activity"

D.1.3 cadf:event.eventTime

CIMI:Event.timestamp → cadf:event.eventTime

D.1.4 cadf:event.action

For CIMI "model" events (modifications to the CIMI resource model), the "cadf:event.action" value will result from a map of the "CIMI:Event.content.change" value. In particular, the CRUD values map to similar CRUD values of the CADF Action Taxonomy (create/read/update/delete)

For CIMI "access" events (access requests to the CIMI resource model), the “cadf.event:action” value will result from a map of the "CIMI:Event.content.operation" value.

NOTE: “alarm” and “status” CIMI events map respectively to “control” and “monitor” events in CADF. Consequently their action value in CADF is already determined as there is only one possible value in the CADF Action Taxonomy for these types.

D.1.5 cadf:event.outcome

- CIMI:Event:outcome = "Pending" → cadf:event.outcome = "pending"
- CIMI:Event:outcome = "Unknown" → cadf:event.outcome = “unknown"
- CIMI:Event:outcome = "Success" → cadf:event.outcome = "success"
- CIMI:Event:outcome = "Failure" → cadf:event.outcome = "failure"
- CIMI:Event:outcome = "Status" → cadf:event.outcome = "success"
    - and will map to an cadf:event:event.type = "monitor".
- CIMI:Event:outcome = "Warning" → cadf:event:outcome = "success"
    - and the event should also contain an cadf:event.severity element, of value to be agreed on.

D.1.6 cadf:event.initiator

This mapping will depend on the CIMI event type:

- If CIMI:Event.type = "access" → cadf:event.initiator = CIMI:Event.content.initiator
- If CIMI:Event.type = "model" → the initiator is not assumed to be part of the CIMI event, but can be traced by correlating with the "access" event causing that model change.
    - This is a mapping extensibility point.
- If CIMI:Event.type = "alarm" → the cadf:event.initiator might not be identified unless recorded in the content.detail. This is a mapping extensibility point.
- If CIMI:Event.type = "monitor" → the cadf:event.initiator might not be identified from the CIMI event. If unknown, it should be set to “nil” value.

D.1.7 cadf:event.target

This attribute maps to CIMI:Event.content.resource, which should be similar to the resource reference in CIMI:EventLog.targetResource.

D.1.8 cadf:event.severity

Must reflect the CIMI:Event.severity value (if any).

- This is a mapping extensibility point.

D.1.9 cadf:event.measurements

Must be present when mapping “state” CIMI events (CIMI:Event.type = "state").Its value must reflect the content of CIMI:Event.content.state.

D.1.10 cadf:event.attachments

Map from CIMI:Event.content. Even if some items of CIMI:Event.content can be extracted and mapped individually thanks to some standardized structure (depending on CIMI:Event.type), the overall CIMI:Event.content value is mapped as an attachment in the CADF Event record.

If the CIMI detailed content of an event (“content.detail” attribute) needs be preserved in CADF, then the whole CIMI:event.content should become an attachment in CADF.

D.2 Informative References

- DSP0263 - Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP Specification, Version 1.0.1, 30 Oct 2012:
    - http://dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.1.pdf

ANNEX E
Mapping CADF Query Syntax to XML and JSON

This section provides examples and guidance on how the CADF Query Syntax can be mapped to both JSON and XML formats.

E.1 XML mapping examples

Using the same conceptual event records and resources as shown for the XML mapping examples, this section shows how several sample queries (using the CADF Query Syntax) would yield the results in JSON format.

E.1.1 Sample event data set used for all examples

The following is a conceptual event log rendered in a CADF XML format which will be used as an event source to illustrate the subsequent queries. It also contains a listing of CADF resource definitions that are referenced within the event records.

Conceptual resultset (e.g. CADF Log derivation) containing a list of resources and event records

```
<resources>
    <resource id="myuuid://location.org/resource/01" typeURI="..."
              geolocationId="myuuid://location.org/loc/NYC"/>
    <resource id="myuuid://location.org/resource/09" typeURI="..."
              geolocationId="myuuid://location.org/loc/WDC"/>
    <resource id="myuuid://location.org/resource/21" typeURI="..."
              geolocationId="myuuid://location.org/loc/BOS"/>
</resources>

<-- Notice resources in these examples only use IDs, in real system
these would be
    defined elsewhere -->

<events>
    <event id="myscheme://mydomain/event/id/1234"
        eventType="activity"
        eventTime="2012-06-22T13:00:00-04:00"
        action="create"
        outcome="success"
        initiatorId="myuuid://location.org/resource/01"
        targetId="myuuid://location.org/resource/09"
        observerId=="myuuid://location.org/resource/21"
        <reporterchain>
            <reporterstep
                role="observer"
```

```
                reporterTime="2012-06-22T23:00:00-02:00">
                <reporterId="myuuid://location.org/resource/21"/>
            </reporterstep>
        </reporterchain>
    </event>
    <event id="myscheme://mydomain/event/id/5678"
        eventType="activity"
        eventTime="2012-07-23T13:00:00-04:00"
        action="delete"
        outcome="failure"
        initiatorId="myuuid://location.org/resource/01"
        targetId="myuuid://location.org/resource/09"
        observerId="myuuid://location.org/resource/0321"
        <reporterchain>
            <reporterstep
                role="observer"
                reporterTime="2012-07-23T23:00:00-02:00">
                <reporterId="myuuid://location.org/resource/21"/>
            </reporterstep>
        </reporterchain>
    </event>
    <event id="myscheme://mydomain/event/id/3333"
        eventType="activity"
        eventTime="2012-08-24T13:00:00-04:00"
        action="create"
        outcome="failure"
        initiatorId="myuuid://location.org/resource/01"
        targetId="myuuid://location.org/resource/09">
        <reporterchain>
            <reporterstep
                role="observer"
                reporterTime="2012-08-24T23:00:00-02:00">
                <reporterId="myuuid://location.org/resource/21"/>
            </reporterstep>
        </reporterchain>
    </event>
</events>
```

E.1.2 Resource create query

To search the logged events for create actions the following query is used:

```
/events/event?filter=action='create'
```

This specific query defines as search against all CADF Event records nested in the "events" list, defined within a (conceptual) "log". When executed against the log described in the previous section the following query will output the event IDs "1234" and "3333" in no particular order as shown below.

***Note** that the "paging" element is empty. This is because the endpoint (server) determines that pagination is unnecessary for two elements.*

```
<resultset count="2" detailLevel="1">
  ...
  <eventSet>
    ...
    <events>
      <event id="myscheme://mydomain/event/id/1234"
        eventType="activity"
        eventTime="2012-07-22T13:00:00-04:00"
        action="create"
        outcome="success"
        initiatorId="myuuid://location.org/resource/01"
        targetId="myuuid://location.org/resource/09"
        observerId="myuuid://location.org/resource/0321"
        <reporterchain>
          <reporterstep role="observer"
            reporterTime="2012-07-22T23:00:00-02:00">
            <reporterId="myuuid://location.org/resource/21"/>
          </reporterstep>
        </reporterchain>
     </event>
     <event id="myscheme://mydomain/event/id/3333"
        eventType="activity"
        eventTime="2012-08-24T13:00:00-04:00"
        action="create"
        outcome="failure"
        initiatorId="myuuid://location.org/resource/01"
        targetId="myuuid://location.org/resource/0099"
        observerId="myuuid://location.org/resource/21"
      <reporterchain>
         <reporterstep role="observer"
            reporterTime="2012-08-24T23:00:00-02:00">
            <reporterId="myuuid://location.org/resource/21"/>
          </reporterstep>
```

```
        </reporterchain>
      </event>
    </events>
  </eventSet>
</resultset>
```

E.1.3 Resource creation failure query

It is possible to construct more compound queries. The following query will output only the last event.

```
/events/event?filter=((action='create')and(outcome='failure'))
```

Any query is allowed as long as it conforms to the query syntax subset.

E.1.4 Reporter time query

To search for an event by its "`reporterTime`" attribute the following query returns the last event.

```
/events/event?filter=reporterchain/reporterstep/reporterTime='2012-08-
24T23:00:00-02:00'
```

E.1.5 Time range query

To search for events that occurred on or after the date '2012-07-22' the following query returns the last two events.

```
/events/event?filter=eventTime>='2012-07-22T00:00:00-02:00'
```

Complex time queries can be used to search for events within a specific time period. The follow query searches for events that occurred between the start of '2012-07-22' and not after '2012-07-23'.

```
/events/event?filter=((eventTime>='2012-07-22T00:00:00-
02:00')and(eventTime<='2012-07-23T00:00:00-02:00'))
```

To search for an event by its "`reporterTime`" attribute the following query returns the last event.

```
/events/event?filter=reporterchain/reporterstep/reporterTime='2012-08-
24T23:00:00-02:00'
```

E.1.6 Pagination query

A query that returns a large number of results may be paginated.

Query:

```
/events/event?filter=eventTime>="2012-05-22T00:00:00-02:00"&limit=2
```

Result:

```
<resultset count="2" detailLevel="1"
  nextPage="http://<addr>/events/event?filter=eventTime>='2012-05-
```

```
22T00:00:00-02:00’&limit=2&offset=2"
  firstPage="http://<addr>/events/event?filter=eventTime>=’2012-05-
22T00:00:00-02:00’&limit=2&offset=1"
  lastPage="http://<addr>/events/event?filter=eventTime>=’2012-05-
22T00:00:00-02:00’&limit=2&offset=3">
  ...
  <eventSet>
    ...
    <events>
      <event id="myscheme://mydomain/event/id/1234" … />
      <event id="myscheme://mydomain/event/id/5678" … />
    </events>
  </eventSet>
</resultset>
```

Note: The ”nextPage”, ”firstPage” and ”lastPage” properties’ values contain URLs that can be used to navigate the complete result set.

E.2 JSON mapping examples

Using the same conceptual event records and resources as shown for the XML mapping examples, this section shows how several sample queries (using the CADF Query Syntax) would yield the results in JSON format.

Please note that the query syntax and filter are the same irrespective of the requested result format (i.e. XML or JSON).

E.2.1 Resource create query

The same query is issued as when the caller expects an XML response:

```
/events/event?filter=action=’create’
```

The query will return the following JSON (abbreviated for readability):

```
{
  "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/resultset",
  "count"=2,
  "detailLevel"=1,
  ...,
  "eventSet": {
     ...,
     "events": [
     {
        "id": "myscheme://mydomain/event/id/1234",
        ...
```

```
    },
    {
       "id": "myscheme://mydomain/event/id/3333",
       ...
    },
    ]
  }
}
```

E.2.2 Pagination query

Using the same paginated query as above:

Query:

```
/events/event?filter=eventTime>="2012-05-22T00:00:00-02:00"&limit=2
```

Results:

```
{
  "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/resultset",
  "count"=2,
  "detailLevel"=1,
  "nextPage"="http://<addr>/events/event?filter=eventTime>=’2012-05-
22T00:00:00-02:00’&limit=2&offset=2",
  "firstPage"="http://<addr>/events/event?filter=eventTime>=’2012-05-
22T00:00:00-02:00’&limit=2&offset=1",
  "lastPage"="http://<addr>/events/event?filter=eventTime>=’2012-05-
22T00:00:00-02:00’&limit=2&offset=3",
  ...,
  "eventSet": {
    ...,
    "events": [
    {
       "id": "myscheme://mydomain/event/id/1234",
       ...
    },
    {
       "id": "myscheme://mydomain/event/id/3333",
       ...
    },
    ],
  }
}
```

ANNEX F
Examples of the CADF Query Interface over HTTP

This section provides examples and guidance on how the can be executed over a REST based HTTP interface using ‘curl’.

F.1.1 Create events query over HTTP

The following curl query searches for ‘create’ events. In this example, no authentication is enabled on the server.

```
curl -v -H "Accept: application/xml" \
     -X GET "http://example.host/events/event?$filter=action=’create’"
```

The HTTP request generated by curl has the following form.

```
GET /events/event?filter=action=’create’ HTTP/1.1
Host: example.host
Accept: application/xml
```

The HTTP response from the server is as follows.

```
HTTP/1.1 200 OK
Date: Fri, 10 May 2013 15:53:47 GMT
Server: Apache/2.2.22 (Ubuntu)
Last-Modified: Mon, 14 Apr 2008 07:11:15 GMT
Accept-Ranges: bytes
Content-Length: 681
Connection: close
Content-Type: application/xml

<resultset count="2" detailLevel="1">
  <eventSet>
    <events>
      <event id="myscheme://mydomain/event/id/1234"
        eventType="activity"
        eventTime="2012-06-22T13:00:00-04:00"
        action="create"
        outcome="success"
        initiatorId="myuuid://location.org/resource/01"
        targetId="myuuid://location.org/target/09"
        observerId=”myuuid://location.org/resource/0321”
        <reporterchain>
           ...
        </reporterchain>
```

```
      </event>
      <event id="myscheme://mydomain/event/id/3333"
        eventType="activity"
        eventTime="2012-08-24T13:00:00-04:00"
        action="create"
        outcome="failure"
        initiatorId="myuuid://location.org/resource/01"
        targetId="myuuid://location.org/target/09"
        observerId=”myuuid://location.org/resource/0321”
        <reporterchain>
           ...
        </reporterchain>
      </event>
    </events>
  </eventSet>
</resultset>
```

Note: *In the above example, the ‘detaillevel’ parameter was not specified and defaulted to “1”. Thus the full properties of the ‘reporterchain’ are not included. Another query specifying a query level value set to “2” or ”3” could be used to request the details of the reporterchain for either of the events.*

3467 ANNEX G
3468 **(informative)**
3469
3470 # Change log

Version	Date	Description
1.0.0c	2014-01-10	Matt Rutkowski (IBM): Editor draft candidate for WIP3 draft public review.

3471

Bibliography

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, 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

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 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

DMTF DSP0263, *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol,* http://dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.1.pdf

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 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, 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/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, *Environmental Management Systems -- Requirements with Guidance for Use*, http://www.iso.org/iso/catalogue_detail?csnumber=31807

ISO/IEC 15288:2008, System and Software Engineering – System life cycle processes, http://www.iso.org/iso/catalogue_detail?csnumber=43564

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/IEC 27000:2009, *Information Technology -- Security Techniques -- Information Security Management Systems -- Overview and vocabulary*, http://www.iso.org/iso/catalogue_detail?csnumber=41933

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/

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, a SourceForge open source implementation of The Open Group's XDAS Version 1 Standard, http://openxdas.sourceforge.net/.

3514 IETF RFC2828, *Internet Security Glossary,* May 2000, http://www.ietf.org/rfc/rfc2828.txt.

3515 IETF RFC3339 (Proposed Standard), *Date and Time on the Internet: Timestamps*, July 2002,
3516 http://www.ietf.org/rfc/rfc3339.txt

3517 IETF RFC4949, *Internet Security Glossary, Version 2*, August 2009, http://www.ietf.org/rfc/rfc4949.txt.

3518 OASIS Standard, *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005.
3519 http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf.

3520 The Open Group, Distributed Audit Services (XDAS) Project, *Distributed Audit Service (XDAS) – Preliminary*
3521 *Specification*, http://www.opengroup.org/bookstore/catalog/p441.htm.

3522 The Open Group, Service-Oriented Cloud Computing Infrastructure (SOCCI) Framework Technical Standard,
3523 http://www.opengroup.org/soa/source-book/socci/

3524 World Wide Web Consortium (W3C) Recommendation, J. Clark and Steve DeRose. *XML Path Language (XPath)*
3525 *Version 1.0*, 16 November 1999, http://www.w3.org/TR/xpath/

3526 World Wide Web Consortium (W3C) Recommendation, A. Berglund, et al., *XML Path Language (XPath) Version*
3527 *2.0*, 14 December 2010, http://www.w3.org/TR/xpath20/

3528 World Wide Web Consortium (W3C) Candidate Recommendation, “A JSON-based Serialization for Linked Data”
3529 JSON-LD 1.0, 10 September 2013, http://www.w3.org/TR/json-ld/