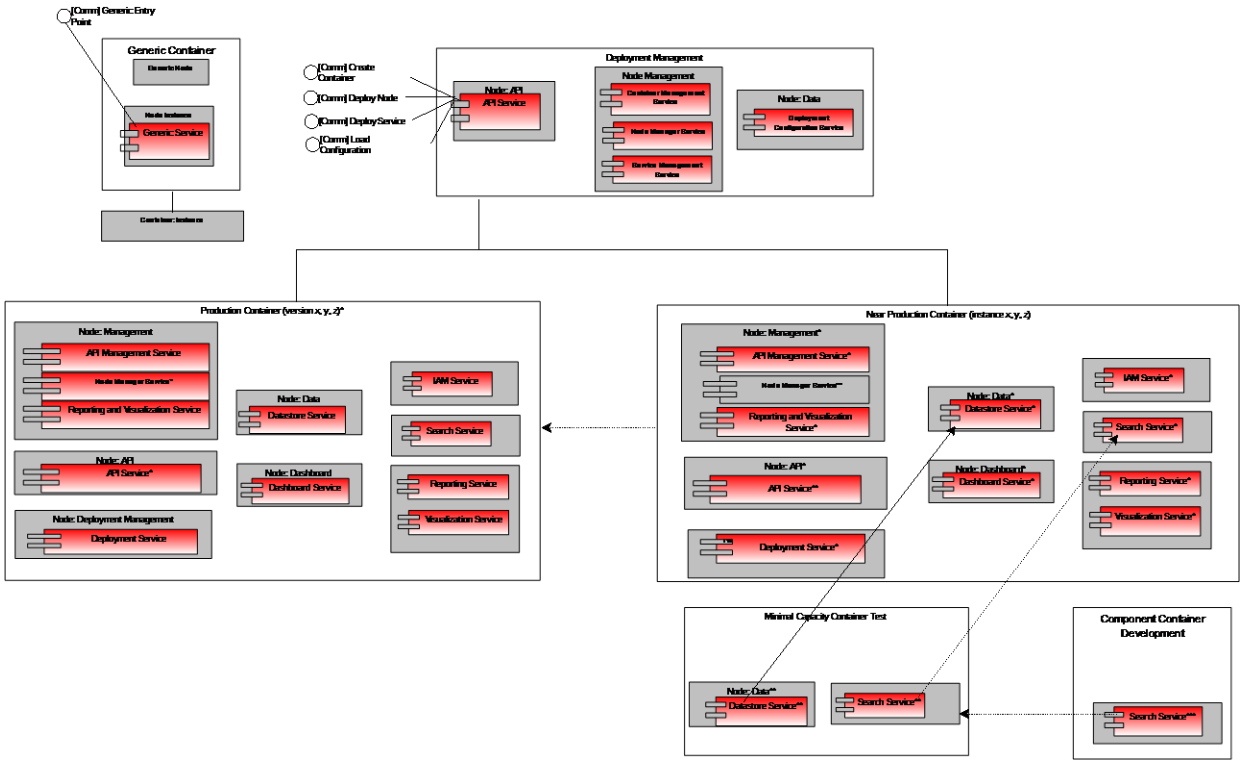
GSA IAE Technical Architecture

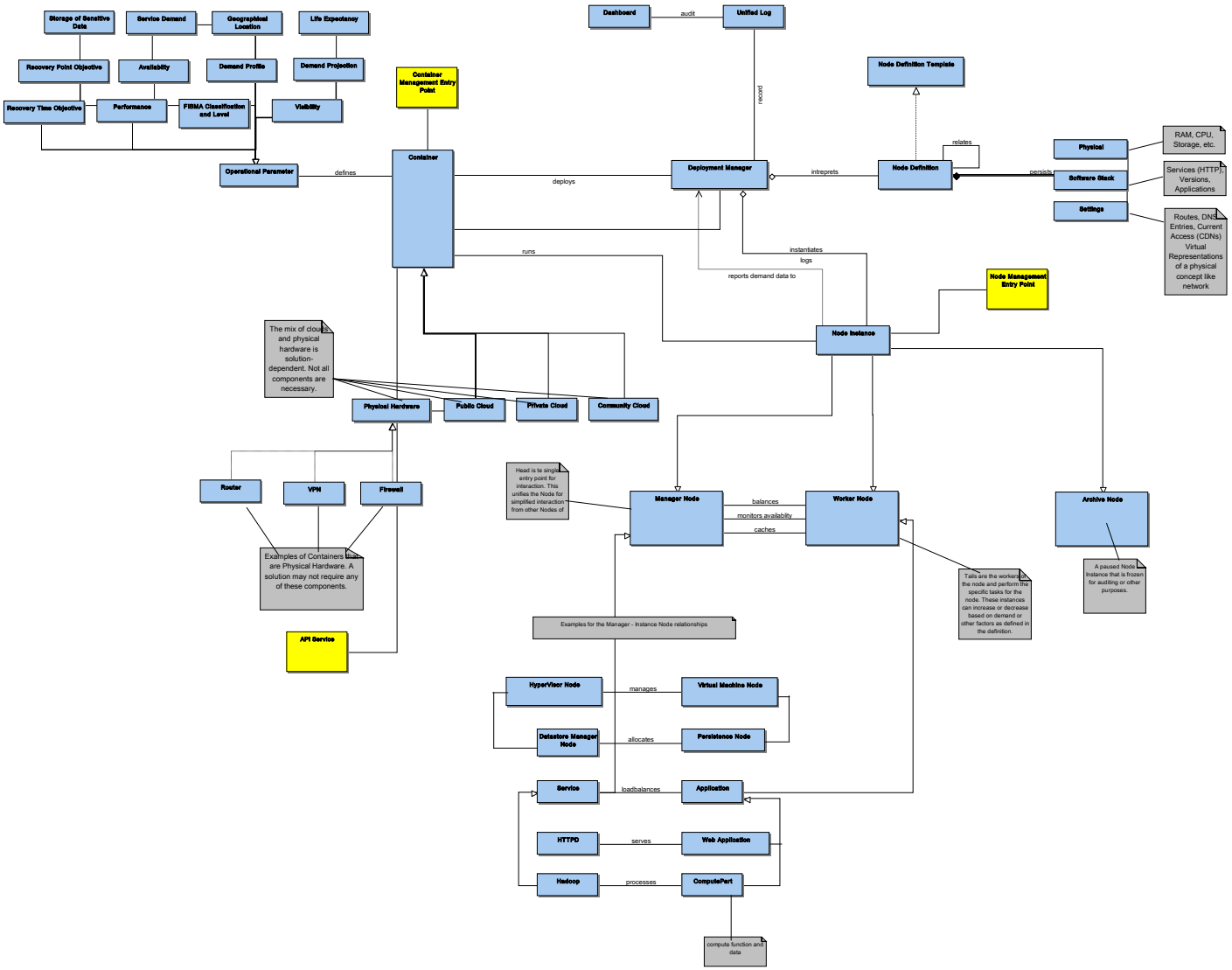
This document provides a point-in-time snapshot of the IAE Conceptual Architecture. It is part of the overall IAE Architecture that describes and defines the To-Be technical state for IAE that includes Business Architecture, Information Architecture and other components. The Conceptual Architecture continues to evolve and this version should not be considered definitive. More information on the current state of the overall architecture is available at interact.gsa.gov.

The information within this document was generated on April 4th, 2013.

Consolidated Deployment Diagram



Hosting Domain Diagram



Hosting Domain Diagram (Interpretations)

Business Class Modeler: Topic: Hosting Domain Diagram

The domain diagram below provides an identification of the nodes that comprise the design. The Domain model provides a conceptual overview of the hosting services identifying the “things” that answer the question “what are the hosting services?”.

API Service

An instance of a Service designed to provide business functionality.

Application

Example of a type of Worker Node - A specific instance of a software stack, i.e. LAMP stack.

For each Application:

- we will find a Service it is associated with
- we may find one Worker Node it is associated with

Archive Node

A node that is frozen for auditing or other purposes

Availablity

A container characteristic that measures the requirement for the system availability (available to the end user) of the container.

For each Availablity:

- we will find a Performance it is associated with

Community Cloud

A semi-private cloud that is organized and managed to support the specific needs of a particular domain. An example might be the DISA cloud or the AWS GovCloud.

For each Community Cloud:

- we may find one Container it is associated with

ComputePart

Example of a Worker Node - A generic instance of a Node that runs a specific set of services.

For each ComputePart:

- we will find a Hadoop it is associated with

Container

Describes the business attributes that are required by a particular environment. A container is a template that, when combined with a Node Definition creates a run-time environment called a Node Instance.

For each Container:

- we will find a Community Cloud it is associated with
- we will find a Container Management Entry Point it is associated with
- we will find a Deployment Manager it is associated with
- we will find a Node Instance it runs
- we will find an Operational Parameter it is associated with
- we will find a Private Cloud it is associated with
- we will find a Public Cloud it is associated with

Container Management Entry Point

The location that the Deployment Manager Service is accessed by the Clients.

For each Container Management Entry Point:
- we will find a Container it is associated with

Dashboard

A user interface to display important performance metrics and other characteristics of the system to those with permissions

For each Dashboard:
- we will find a Unified Log it audit

Datastore Manager Node

Example of a Manager Node - A storage node, NOT a database, that can be used across nodes for typical physical storage needs, i.e. Filesystem

For each Datastore Manager Node:
- we will find a Persistence Node it allocates

Demand Profile

A container characteristic that identified the requirement for short and long term demand of the container. Demand may be specified over a 24 hour cycle, weekly, monthly, quarterly, or through an annual cycle. Demand defines the maximum capacity that must be supported.

Demand Projection

How demand will grow over the long term.

Deployment Manager

The single point within the IAE network in charge of managing and instantiating all nodes, including managing auto-scaling and other load management controls.

For each Deployment Manager:
- we will find a Container it deploys
- we will find a Node Instance it is associated with
- we will find a Unified Log it is associated with

Firewall

A physical network component providing layer 4 or 7 filtering.

FISMA Classification and Level

A container characteristic that measures the requirement for the FISMA classification - Low, Moderate, High - of the container.

Geographical Location

Restrictions on geographical location

Hadoop

Example of a Manager Node - Just a specific example of things that can be implemented as nodes as opposed to just VMs

For each Hadoop:

- we will find a ComputePart it processes

HTTPD

Example of a Manager Node - HTTPD would be considered an entry point for a series of services used in a Node.

For each HTTPD:

- we will find a Web Application it serves

HyperVisor Node

Example of a Manager Node - HyperVisor is for a specific instance of a set of VM Nodes

For each HyperVisor Node:

- we will find a Virtual Machine Node it manages

Life Expectancy

A typical length of time that an environment based on this container template would last

Manager Node

A part of a node used for interaction between other nodes or clients.

For each Manager Node:

- we will find a Worker Node it balances
- we will find a Worker Node it caches
- we will find a Worker Node it monitors availablity

Node Definition

The pre-defined configurations for a specified node. Nodes are one or more services that the system supports.

For each Node Definition:

- we will find a Node Definition it relates

Node Definition Template

An example or preconfigured node definition that can be used to quickly create a complete node definition.

Node Instance

An instance of one of the node definitions managed by the deployment manager

For each Node Instance:

- we will find a Container it is associated with
- we may find one Deployment Manager it reports demand data to
- we will find a Node Management Entry Point it is associated with

Node Management Entry Point

For each Node Management Entry Point:

- we will find a Node Instance it is associated with

Operational Parameter

Characteristics of a Container

For each Operational Parameter:

- we will find a Container it defines
- we will find a Performance it is associated with
- we will find a Recovery Time Objective it is associated with

Performance

A container characteristic that measures the requirement for the performance - response time, latency, (RTO) of the container.

For each Performance:

- we will find an Availablity it is associated with
- we will find an Operational Parameter it is associated with

Persistence Node

Example of a Worker Node - A specific instance of a node which provides the Database service

For each Persistence Node:

- we will find a Datastore Manager Node it is associated with

Physical

Required configurations that persist for all instances of the node. The physical configuration would relate to physical attributes of a solution. Location, Routing, Hardware, etc.

Physical Hardware

Physical environments, perhaps virtualized but not implemented as a cloud infrastructure.

Private Cloud

A part of the GSA Network used to provide compute and storage capacity

For each Private Cloud:

- we will find a Container it is associated with

Public Cloud

A third-party service used to provide compute, storage and other services

For each Public Cloud:

- we will find a Container it is associated with

Recovery Point Objective

A container characteristic that measures the requirement for the Recovery Point Objective (RTO) of the container.

For each Recovery Point Objective:

- we will find a Recovery Time Objective it is associated with

Recovery Time Objective

A container characteristic that measures the requirement for the Recovery Time Objective (RTO) of the container.

For each Recovery Time Objective:

- we will find an Operational Parameter it is associated with
- we will find a Recovery Point Objective it is associated with
- we will find a Visibility it is associated with

Router

A physical network device that provides layer 3 routing of packets

Service

A business or technical function that is encapsulated as a single logical unit. The concept of service is defined within the IAE API Services Technical Architecture.

For each Service:

- we will find an Application it loadbalances

Service Demand

The level of usage, number of users, API calls and other measures

Settings

Required configurations that persist for all instances of the node. The conceptual configuration would relate the conceptual “high-level” setings required for hosting or deployment.

Software Stack

Required configurations that persist for all instances of the node. The logical configuration would relate to the version and instance of the software being used in the solution.

Storage of Sensitive Data

Whether PII or other information may be stored or processed within the environment

Unified Log

A log for the whole domain retaining important information such as the number of logins, errors, number of users and performance metrics

For each Unified Log:

- we will find a Dashboard it is associated with
- we will find a Deployment Manager it record

Notes:

- state, usages, protocol, application logs may be used to develop quants that relate running systems to costs, etc.

Virtual Machine Node

Example of a Worker Node - A virtual machine node.

For each Virtual Machine Node:

- we will find a HyperVisor Node it is associated with

Visibility

A container characteristic that measures the requirement for the the sensitivty of the data stored, accessed and transformed by the container.

For each Visibility:

- we will find a Recovery Time Objective it is associated with

VPN

A physical network device that provides secure remote access.

Web Application

Example of a Worker Node - A specific instance of a generic dedicated server implementation - A JBOSS application.

For each Web Application:

- we will find an HTTPD it is associated with

Worker Node

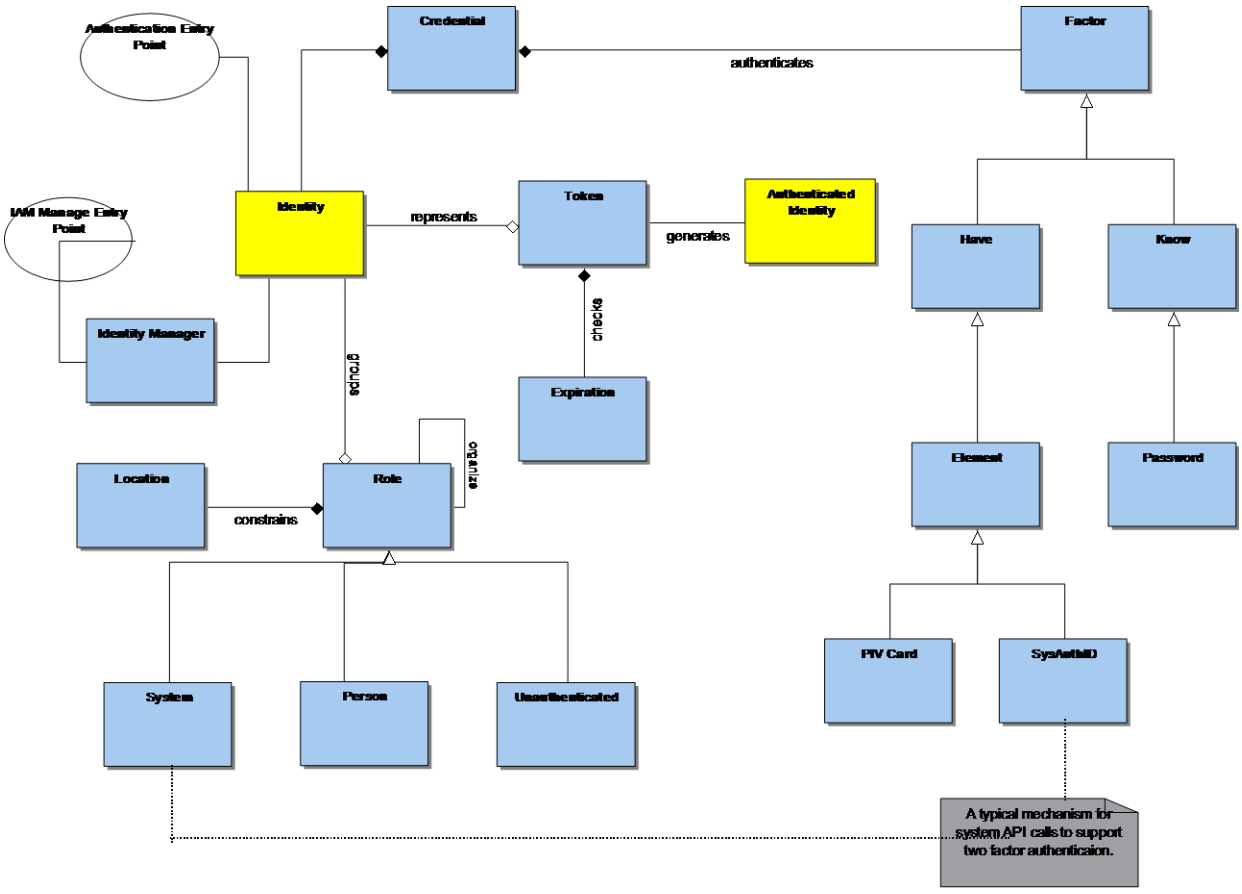
A group within a node that perform the specific tasks for the node

For each Worker Node:

- we will find an Application it is associated with
- we will find a Manager Node it is associated with

- we will find a Manager Node it is associated with
- we will find a Manager Node it is associated with

IAM Domain Diagram



IAM Domain Diagram (Interpretations)

Business Class Modeler: Topic: IAM Domain Diagram

Created: 4/4/2014 1:07:13 PM
Changed: 4/4/2014 1:37:44 PM by Local User:PamelaAMiller (Owner of local repository)
Version: 0
Status: Draft

Authenticated Identity

Type: Concrete
For each Authenticated Identity:

- we will find a Token it generates

Authentication Entry Point

The location that the service is accessed by the Clients.

Type: Concrete
For each Authentication Entry Point:

- we will find an Identity it is associated with

Credential

The combination of the identity and factor being authentication.

Type: Concrete
It includes two constituent Business Classes:

- Factor
  A part of a credential that is unique to the requester seeking authentication. A valid factor in a two-factor authentication system is some combination of Have and Know.
- Identity
  The unique representation of the requestor for identification that is grouped by 1 or more Roles

Each Credential can perform:

- Send Credential
  The combination of the identity and factor that is used for authentication.
- Check Credential

Element

(a kind of Have)

A generic instance of a second factor “Have”

Type: Concrete

Expiration

A timeout for the token. If a token is expired it is invalid. The configuration of timeouts can be constrained by role.

Type: Concrete
It is a constituent of:

- Token
  The representation of the authentication transaction. Valid by a constrained Expiration. This is

checked by services to determine if identity has been confirmed and authenticated. This is proved by a valid token from IA&M. The individual service will determine access.

Factor

A part of a credential that is unique to the requester seeking authentication. A valid factor in a two-factor authentication system is some combination of Have and Know.

Type: Concrete
It is a constituent of:

- Credential
  The combination of the identity and factor being authentication.

Have

(a kind of Factor)

An authentication factor which relies on the users possession of a particular, physical item that is not part of their body, e.g., a PIV card.

Type: Concrete

IAM Manage Entry Point

The location that the service is accessed by the user or administrator.

Type: Concrete
For each IAM Manage Entry Point:

- we will find an Identity Manager it is associated with

Identity

The unique representation of the requestor for identification that is grouped by 1 or more Roles

Type: Concrete
It is a constituent of:

- Credential
  The combination of the identity and factor being authentication.
- Role
  The hierarchy of roles as defined by business hierarchy not defined by system usage roles.

  I&AM is not a federated access to resource system. It is an augmented credentialing system. A global (external List to system) Role is augmented to identity for system authentication. No resource Access Control is defined for I&AM.

- Token
  The representation of the authentication transaction. Valid by a constrained Expiration. This is checked by services to determine if identity has been confirmed and authenticated. This is proved by a valid token from IA&M. The individual service will determine access.

For each Identity:

- we will find an Authentication Entry Point it is associated with
- we will find an Identity Manager it is associated with

Identity Manager

Type: Concrete
For each Identity Manager:

- we will find an IAM Manage Entry Point it is associated with
- we will find an Identity it is associated with

Know

(a kind of Factor)

An authentication factor which relies on a piece of knowledge the user has, e.g., a password

Type: Concrete

Location

A geographical constraint on role. The role may constrained only to allow authentication from specific geographical locations.

Type: Concrete
It is a constituent of:

- Role
  The hierarchy of roles as defined by business hierarchy not defined by system usage roles.

  I&AM is not a federated access to resource system. It is an augmented credentialing system. A global (external List to system) Role is augmented to identity for system authentication. No resource Access Control is defined for I&AM.

Notes:

- <<new Note>>
- Location object is used in sever places. Resolve the definition or make separate notes on the objects to make specific examples.

Password

(a kind of Know)

A secret phrase known only to the user.

Type: Concrete

Person

(a kind of Role)

A specific type of Role that represents a human/user. System and Person are the types of roles that can be augmented and must be managed by I&AM

Type: Concrete
For each Person (inherited from Role):

- we may find several Roles it organize
  This forms the hierarcy of roles

PIV Card

(a kind of Element)

An example of a second factor “Have” for a person to authenticate

Type: Concrete

Role

The hierarchy of roles as defined by business hierarchy not defined by system usage roles.

I&AM is not a federated access to resource system. It is an augmented credentialing system. A global (external List to system) Role is augmented to identity for system authentication. No resource Access Control is defined for I&AM.

Type: Concrete
It includes two constituent Business Classes:

- Identity
  The unique representation of the requestor for identification that is grouped by 1 or more Roles
- Location
  A geographical constraint on role. The role may constrained only to allow authentication from specific geographical locations.

It is a constituent of:

- Acess Map

For each Role:

- we may find several Roles it organize
  This forms the hierarcy of roles

SysAuthID

(a kind of Element)

A second factor “Have” for a system to authenticate

Type: Concrete

System

(a kind of Role)

A specific type of Role that represents a technical non-human. API access.

Type: Concrete
For each System (inherited from Role):

- we may find several Roles it organize
  This forms the hierarcy of roles

Token

The representation of the authentication transaction. Valid by a constrained Expiration. This is checked by services to determine if identity has been confirmed and authenticated. This is proved by a valid token from IA&M. The individual service will determine access.

Type: Concrete
It includes two constituent Business Classes:

- Expiration
  A timeout for the token. If a token is expired it is invalid. The configuration of timeouts can be constrained by role.

- Identity
    The unique representation of the requestor for identification that is grouped by 1 or more Roles

For each Token:

- we will find an Authenticated Identity it is associated with

Each Token can perform:

- Send Token
    The representation of the authentication transaction. Valid by a constrained Expiration. This is checked by services to determine if identity has been confirmed and authenticated. This is proved by a valid token from IA&M. The individual service will determine access.
- Check Token
- Check Access
- Send Response
- Check Expiration
- Generate Token

Unauthenticated

(a kind of Role)

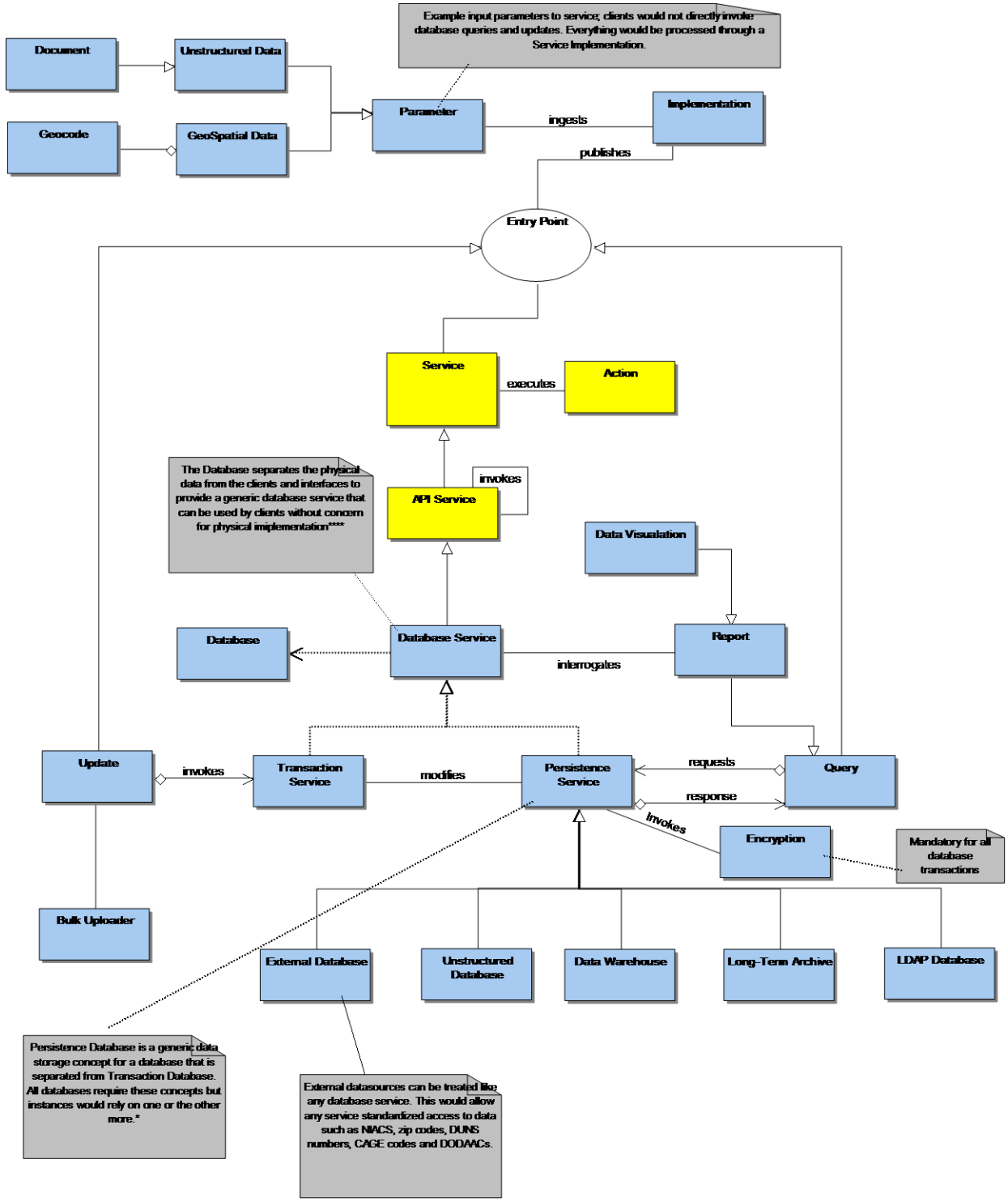
An unauthenticated or non-role assigned user (sytem or person) of a resource, An un-authenticated access can be allowed which is essentially an empty role

Type: Concrete

For each Unauthenticated (inherited from Role):

- we may find several Roles it organize
    This forms the hierarcy of roles

Datastore Domain Diagram



Datastore Domain Diagram (Interpretations)

Business Class Modeler: Topic: Datastore Domain Diagram

Created: 4/4/2014 1:07:13 PM
Changed: 4/4/2014 1:37:56 PM by Local User:PamelaAMiller (Owner of local repository)
Version: 0
Status: Draft

Action

One of the capabilities of a service

Type: Concrete
For each Action:
- we will find a Service it is associated with

API Service

(a kind of Service)

An instance of a Service designed to provide business functionality.

Type: Concrete
For each API Service:
- we will find an API Service it invokes

For each API Service (inherited from Service):
- we will find an Action it executes

Each API Service can perform (inherited from Service):
- Request
- Response

Bulk Uploader

Bulk load of existing data to support environment creation or migration.

Type: Concrete
For each Bulk Uploader:
- we will find an Update it is associated with

Data Visualation

(a kind of Report)

A specific instance of a report with particular capabilities related to graphical representation of data

Type: Concrete
For each Data Visualation (inherited from Report):
- we will find a Database Service it interrogates

Data Warehouse

(a kind of Persistence Service)

A specialized example of a persistence database organized to support reporting

Type: Concrete

For each Data Warehouse (inherited from Persistence Service):

- we may find one Database Service it is associated with
- we will find an Encryption it Invokes
- we may find one Query it requests
- we may find one Query it response
- we will find a Transaction Service it is associated with

Database

A database is an implementation of a database service.

Type: Concrete
For each Database:

- we will find a Database Service it is associated with

Database Service

(a kind of API Service)

A generic data storage service

Type: Concrete
For each Database Service:

- we may find one Database it is associated with
- we will find a Persistence Service it is associated with
- we will find a Report it is associated with
- we will find a Transaction Service it is associated with

For each Database Service (inherited from API Service):

- we will find an API Service it invokes

Each Database Service can perform (inherited from Service):

- Request
- Response

Document

(a kind of Datum)

Part of a request for information from a data store

Type: Concrete
For each Document:

- we may find one Unstructured Data it is associated with

Encryption

A specific instance of an action that may be performed as part of the data service. The capability to encrypt data at rest is a mandatory requirement.

Type: Concrete
For each Encryption:

- we will find a Persistence Service it is associated with

Entry Point

The location that the service is accessed by the Clients. Possible entry points include IP and port combinations, URLs, email addresses or drop boxes

Type: Concrete
For each Entry Point:
- we will find an Implementation it publishes

Notes:
- Entry point class is used in other diagrams.

External Database

(a kind of Persistence Service)

An example of a persistence store

Type: Concrete
For each External Database (inherited from Persistence Service):
- we may find one Database Service it is associated with
- we will find an Encryption it Invokes
- we may find one Query it requests
- we may find one Query it response
- we will find a Transaction Service it is associated with

Geocode

Part of a request for information from a data store

Type: Concrete
It is a constituent of:
- GeoSpatial Data
  Part of a request for information from a data store

GeoSpatial Data

(a kind of Parameter)

Part of a request for information from a data store

Type: Concrete
It includes one constituent Business Class:
- Geocode
  Part of a request for information from a data store

For each GeoSpatial Data (inherited from Parameter):
- we will find an Implementation it ingests

Implementation

A generic protocol implementation, describing the available implementations of the API

Type: Concrete
For each Implementation:
- we will find an Entry Point it is associated with
- we will find a Parameter it is associated with

LDAP Database

(a kind of Persistence Service)

An example of a persistence store that is optimized for directory information.

Type: Concrete

For each LDAP Database (inherited from Persistence Service):

- we may find one Database Service it is associated with
- we will find an Encryption it Invokes
- we may find one Query it requests
- we may find one Query it response
- we will find a Transaction Service it is associated with

Long-Term Archive

(a kind of Persistence Service)

A specialized example of a persistence database organized for long-term storage of data

Type: Concrete
For each Long-Term Archive (inherited from Persistence Service):

- we may find one Database Service it is associated with
- we will find an Encryption it Invokes
- we may find one Query it requests
- we may find one Query it response
- we will find a Transaction Service it is associated with

Parameter

Part of a request for information from a data store

Type: Concrete
For each Parameter:

- we will find an Implementation it ingests

Persistence Service

A data store optimized and separate from the transactiona data store

Type: Concrete
For each Persistence Service:

- we may find one Database Service it is associated with
- we will find an Encryption it Invokes
- we may find one Query it requests
- we may find one Query it response
- we will find a Transaction Service it is associated with

Query

(a kind of Entry Point)

A query facilitates the retrieval of data from the data store. A request by a user to provide matches from the index that match the information provided.

Type: Concrete
It includes one constituent Business Class:

- Query Context Constraint
  Determines which domain to query. It allows for complex queries for an entity across legacy apps or services and can provide context for analysis, ordering of results.

For each Query:

- we may find one Persistence Service it is associated with
- we may find one Persistence Service it is associated with

For each Query (inherited from Entry Point):

- we will find an Implementation it publishes

Report

(a kind of Query)

An end user report

Type: Concrete
For each Report:
- we will find a Database Service it interrogates

For each Report (inherited from Query):
- we may find one Persistence Service it is associated with
- we may find one Persistence Service it is associated with

Service

(a kind of Entry Point, Manager Node)

A business or technical function that is encapsulated as a single logical unit. The concept of service is defined within the IAE API Services Technical Architecture.

Type: Concrete
For each Service:
- we will find an Action it executes

For each Service (inherited from Entry Point):
- we will find an Implementation it publishes

Each Service can perform:
- Request
- Response

Transaction Service

A data store optimized to support transactional activities

Type: Concrete
For each Transaction Service:
- we may find one Database Service it is associated with
- we will find a Persistence Service it modifies
- we may find one Update it is associated with

Unstructured Data

(a kind of Parameter)

Part of a request for information from a data store

Type: Concrete
For each Unstructured Data:
- we will find a Document it is associated with

For each Unstructured Data (inherited from Parameter):
- we will find an Implementation it ingests

Unstructured Database

(a kind of Persistence Service)

An example of a persistence store that is optimized for unstructured data such as documents and images.

Type: Concrete
For each Unstructured Database (inherited from Persistence Service):

- we may find one Database Service it is associated with
- we will find an Encryption it Invokes
- we may find one Query it requests
- we may find one Query it response
- we will find a Transaction Service it is associated with

Update

(a kind of Entry Point)

An interaction with a database that creates new or modifies existing data

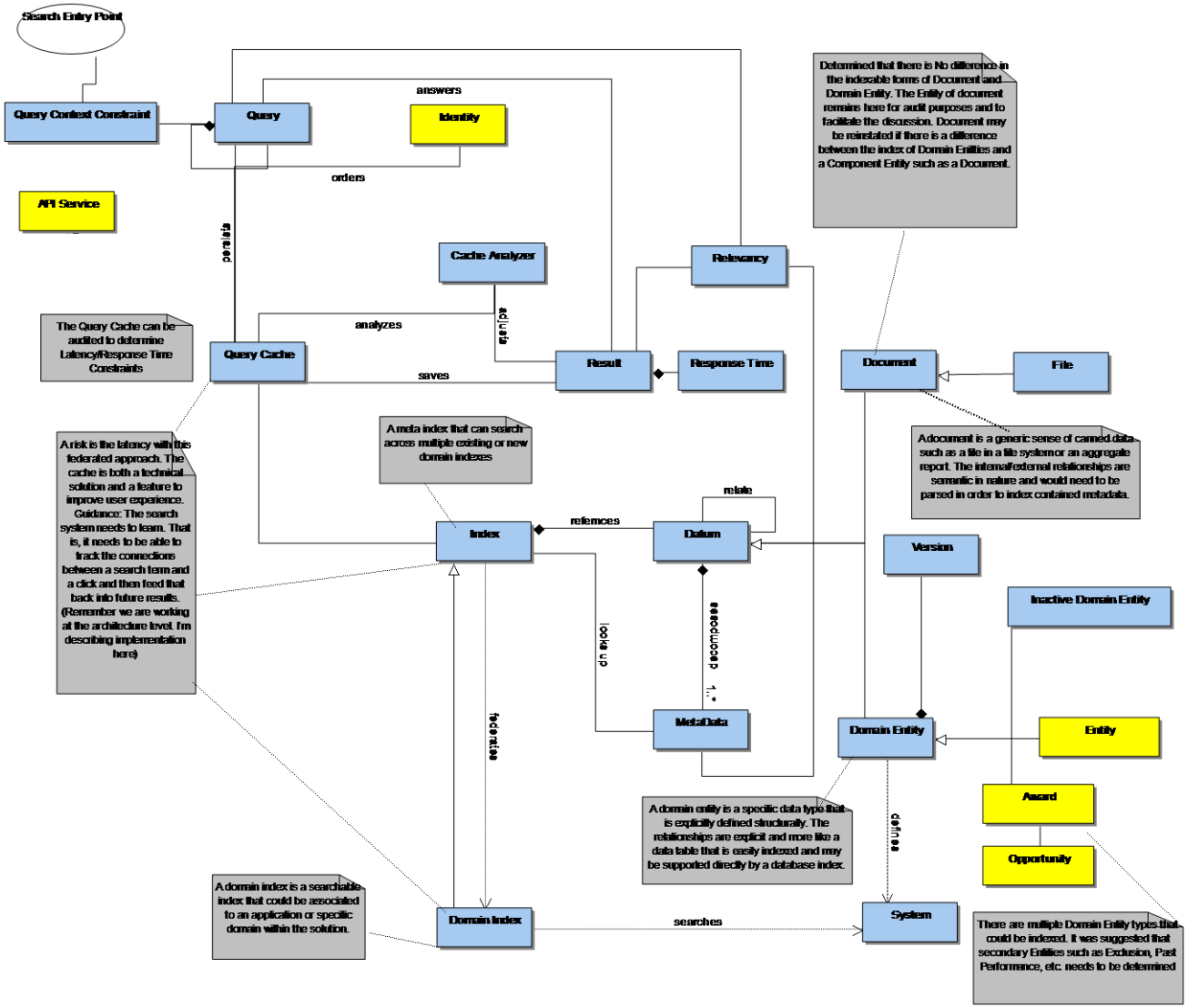
Type: Concrete
For each Update:

- we will find a Bulk Uploader it is associated with
- we may find one Transaction Service it invokes

For each Update (inherited from Entry Point):

- we will find an Implementation it publishes

Search Domain Diagram



Search Domain Diagram (Interpretations)

Business Class Modeler: Topic: Search Domain Diagram

The domain diagram provides an identification of the Entities that comprise this component. The Domain model provides a conceptual overview of search identifying the “things” that answer the “what is search?”.

Notes:
- This model used in TA document is more uptodate than this

API Service

An instance of a Service designed to provide business functionality.

For each API Service:
- we will find an API Service it invokes

Award

For each Award:
- we will find an Inactive Domain Entity it is associated with
- we will find an Opportunity it is associated with

Cache Analyzer

The cache analyzer uses the the user response to search results to identify which responses are most appropriate

For each Cache Analyzer:
- we will find a Query Cache it analyzes
- we will find a Result it adjusts

Datum

A business entity that is indexed and searchable

For each Datum:
- we will find a Datum it relate

Document

Part of a request for information from a data store

Domain Entity

Structured data that is part of a datum

Domain Index

An index created and made available by a legacy system

Entity

An organization that can be an awardee

File

A particular Document type

Identity

The unique representation of the requestor for identification that is grouped by 1 or more Roles

For each Identity:

- we will find a Query Cache it orders

Inactive Domain Entity

An example of a business entity that exists, but is no longer relevant to search

For each Inactive Domain Entity:

- we will find an Award it is associated with

Index

A data store organized to allow matching against user provided queries.

For each Index:

- we will find a MetaData it looks up
- we will find a Query Cache it is associated with

MetaData

Data extracted from the Document and Domain Entities that identifies the attributes and attribute values for a Datum.

For each MetaData:

- we will find an Index it is associated with
- we will find a Relevancy it is associated with

Opportunity

For each Opportunity:

- we will find an Award it is associated with

Query

A query facilitates the retrieval of data from the data store. A request by a user to provide matches from the index that match the information provided.

For each Query:

- we will find a Query Cache it is associated with
- we will find a Relevancy it is associated with
- we will find a Result it is associated with

Query Cache

A meta index that can provide responses to search from multiple existing domain indexes

For each Query Cache:

- we will find a Cache Analyzer it is associated with
- we will find an Identity it is associated with
- we will find an Index it is associated with
- we will find a Query it persists
- we will find a Result it saves

Query Context Constraint

Determines which domain to query. It allows for complex queries for an entity across legacy apps or services and can provide context for analysis, ordering of results.

For each Query Context Constraint:

- we will find a Search Entry Point it is associated with

Relevancy

Relates a query and response and tracks the usefulness (a synonym of relevancy) to improve future response sets

For each Relevancy:

- we will find a MetaData it is associated with
- we will find a Query it is associated with
- we will find a Result it is associated with

Response Time

The time taken to produce a response to a query

Result

For each Result:

- we will find a Cache Analyzer it is associated with
- we will find a Query it answers
- we will find a Query Cache it is associated with
- we will find a Relevancy it is associated with

Search Entry Point

For each Search Entry Point:

- we will find a Query Context Constraint it is associated with

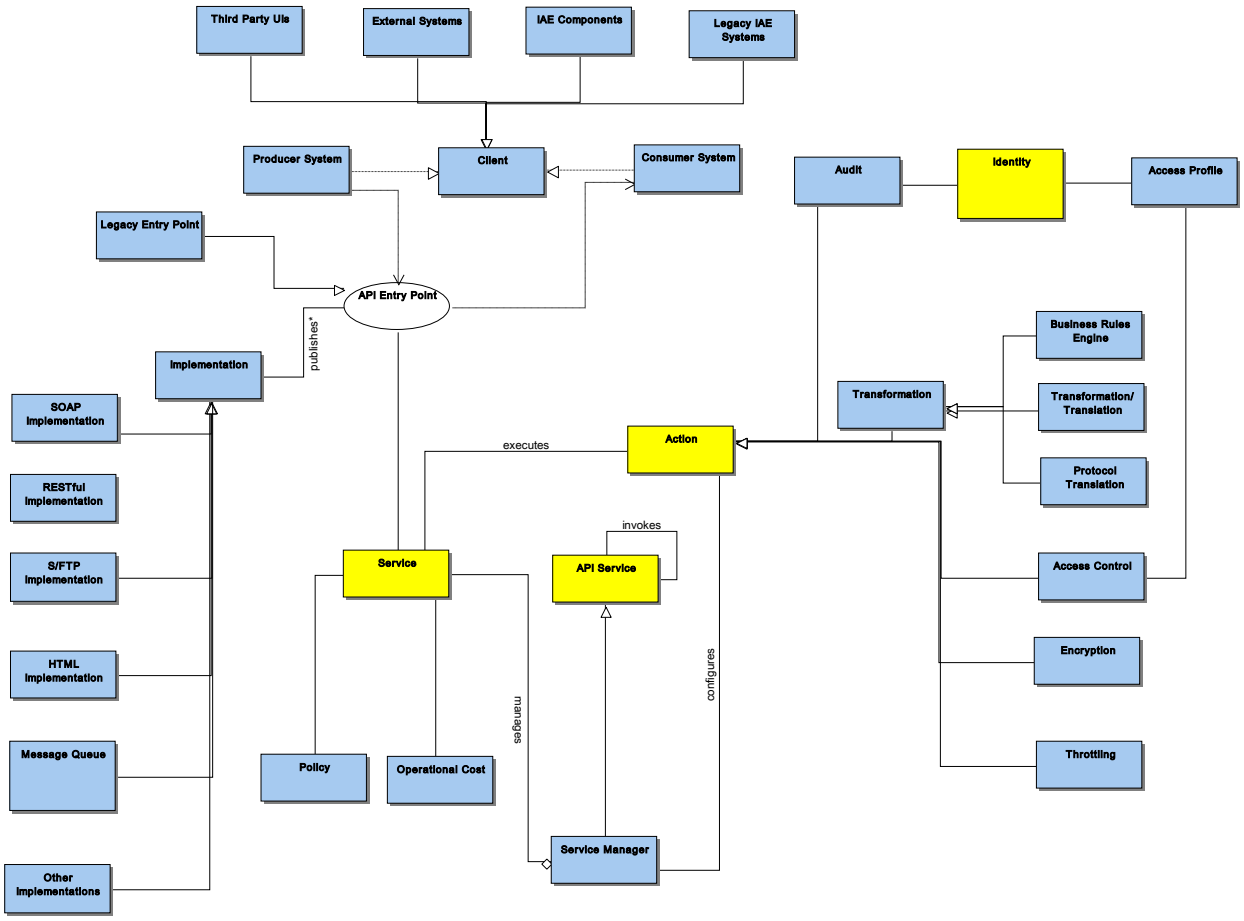
System

A specific type of Role that represents a technical non-human. API access.

Version

Many items with the domain pass through multiple revisions, for example a GWAC solicitation package goes through multiple versions. Version allows the documents to be tracked through its history ensuring that only the current document is presented (or a specific version if requested).

API Management Service



API Management Service (Interpretations)

Business Class Modeler: Topic: API Management Service

Created: 4/4/2014 1:07:13 PM
Changed: 4/4/2014 1:38:25 PM by Local User:PamelaAMiller (Owner of local repository)
Version: 0
Status: Draft

Access Control

A function that ensures that Services are available based on pre-approval

Type: Concrete
For each Access Control:

- we will find an Access Profile it is associated with
- we may find one Action it is associated with

Access Profile

Detail of which Services a particular Account may access such as throttling and time windowing.

Type: Concrete
For each Access Profile:

- we will find an Access Control it is associated with
- we will find an Identity it is associated with

Action

One of the capabilities of a service

Type: Concrete
For each Action:

- we will find an Access Control it is associated with
- we will find an Audit it is associated with
- we will find an Encryption it is associated with
- we will find a Service it is associated with
- we will find a Service Manager it is associated with
- we will find a Throttling it is associated with
- we will find a Transformation it is associated with

API Entry Point

The location that the service is accessed by the Clients. Possible entry points include IP and port combinations, URLs, email addresses or drop boxes

Type: Concrete
For each API Entry Point:

- we will find an Implementation it publishes*
- we will find a Service it is associated with

API Service

(a kind of Service)

An instance of a Service designed to provide business functionality.

Type: Concrete
For each API Service:
- we will find an API Service it invokes
- we will find a Service Manager it is associated with

For each API Service (inherited from Service):
- we will find an Action it executes
- we will find an API Entry Point it is associated with
- we will find an Operational Cost it is associated with
- we will find a Policy it is associated with
- we may find one Service Manager it is associated with

Each API Service can perform (inherited from Service):
- Request
- Response

Audit

A function that supports the analysis of usage of the Service

Type: Concrete
For each Audit:
- we may find one Action it is associated with
- we will find an Identity it is associated with

Business Rules Engine

(a kind of Transformation)

A function that applies business rules that guide the transformation of data within an API body

Type: Concrete
For each Business Rules Engine (inherited from Transformation):
- we may find one Action it is associated with

Client

A generic system

Type: Concrete

Consumer System

(a kind of Client)

A cusumer is a Client system that initiates an API request and consumes the output from the Producer System

Type: Concrete

Encryption

A specific instance of an action that may be performed as part of the data service. The capability to encrypt data at rest is a mandatory requirement.

Type: Concrete
For each Encryption:
- we may find one Action it is associated with

External Systems

(a kind of Client)

Any system operated by organizations other than IAE that access IAE services.

Type: Concrete

HTML Implementation

(a kind of Implementation)

A HTML implementation of the Service

Type: Concrete
For each HTML Implementation (inherited from Implementation):

- we will find an API Entry Point it is associated with
- we will find a Message Queue it is associated with

IAE Components

(a kind of Client)

Components of the To-Be architecture

Type: Concrete

Identity

The unique representation of the requestor for identification that is grouped by 1 or more Roles

Type: Concrete
It is a constituent of:

- Credential
  The combination of the identity and factor being authentication.
- Role
  The hierarchy of roles as defined by business hierarchy not defined by system usage roles.

  I&AM is not a federated access to resource system. It is an augmented credentialing system. A global (external List to system) Role is augmented to identity for system authentication. No resource Access Control is defined for I&AM.

- Token
  The representation of the authentication transaction. Valid by a constrained Expiration. This is checked by services to determine if identity has been confirmed and authenticated. This is proved by a valid token from IA&M. The individual service will determine access.

For each Identity:

- we will find an Access Profile it is associated with
- we will find an Audit it is associated with

Implementation

A generic protocol implementation, describing the available implementations of the API

Type: Concrete
For each Implementation:

- we will find an API Entry Point it is associated with
- we will find a Message Queue it is associated with

Legacy Entry Point

(a kind of API Entry Point, Entry Point)

An entry point separate from the IAE API component that implements an API interface that is also implemented within the API component

Type: Concrete
For each Legacy Entry Point (inherited from API Entry Point):
- we will find an Implementation it publishes*
- we will find a Service it is associated with

Legacy IAE Systems

(a kind of Client)

Legacy IAE systems that have been modified to use the IAE API Services

Type: Concrete

Message Queue

Type: Concrete
For each Message Queue:
- we will find an Implementation it is associated with

Operational Cost

A function that allows tracking and reporting of costs associated with Service usage

Type: Concrete
For each Operational Cost:
- we will find a Service it is associated with

Other Implementations

(a kind of Implementation)

A placeholder for other implementations of the Servces

Type: Concrete
For each Other Implementations (inherited from Implementation):
- we will find an API Entry Point it is associated with
- we will find a Message Queue it is associated with

Policy

Governs contracts between producer and consumer.

Type: Concrete
For each Policy:
- we will find a Service it is associated with

Producer System

(a kind of Client)

A producer is a Client system that responds to a particular API request.

Type: Concrete

Protocol Translation

(a kind of Transformation)

A generic function that supports the translation of protocols between Clients.

Type: Concrete
For each Protocol Translation (inherited from Transformation):

- we may find one Action it is associated with

RESTful Implementation

A RESTful implementation of the Service

Type: Concrete

S/FTP Implementation

(a kind of Implementation)

An FTP/SFTP implementation of the Service

Type: Concrete
For each S/FTP Implementation (inherited from Implementation):

- we will find an API Entry Point it is associated with
- we will find a Message Queue it is associated with

Service

(a kind of Entry Point, Manager Node)

A business or technical function that is encapsulated as a single logical unit. The concept of service is defined within the IAE API Services Technical Architecture.

Type: Concrete
For each Service:

- we will find an Action it executes
- we will find an API Entry Point it is associated with
- we will find an Operational Cost it is associated with
- we will find a Policy it is associated with
- we may find one Service Manager it is associated with

Each Service can perform:

- Request
- Response

Service Manager

A generic function that supports the management of API services

Type: Concrete
For each Service Manager:

- we will find an Action it configures
- we may find one API Service it is associated with
- we will find a Service it manages

SOAP Implementation

(a kind of Implementation)

A SOAP implementation of the Service

Type: Concrete
For each SOAP Implementation (inherited from Implementation):

- we will find an API Entry Point it is associated with
- we will find a Message Queue it is associated with

Third Party UIs

(a kind of Client)

Systems operated by non-IAE organizations that access IAE API Services to support the implementation of user interfaces that replace the standard IAE user interfaces.

Type: Concrete

Throttling

A function that defines usage rules and allows for the controlled degradation of service. Examples of throttling rules include “20,000 API calls per month”, “Maximum response is 1MB”, “Access is only available outside peak hours

Type: Concrete
For each Throttling:

- we may find one Action it is associated with

Transformation

A generic function that converts the content of requests

Type: Concrete
For each Transformation:

- we may find one Action it is associated with

Transformation/Translation

(a kind of Transformation)

A generic function that allows the translation of the content of an API body. For example the translation from ASCII to utf-8.

Type: Concrete
For each Transformation/Translation (inherited from Transformation):

- we may find one Action it is associated with