Open Group Snapshot

Open Platform 3.0™

NOTICE

Snapshots are draft documents, which provide a mechanism for The Open Group to disseminate information on its current direction and thinking to an interested audience, in advance of formal publication, with a view to soliciting feedback and comment.

A Snapshot represents the interim results of a technical activity. Although at the time of publication The Open Group intends to progress the activity towards publication of a Preliminary Standard or Open Group Standard, The Open Group is a consensus organization, and makes no commitment regarding publication. Similarly, a Snapshot does not represent any commitment by any member of The Open Group to make any specific products available.

This Snapshot document is valid through January 31, 2017 only.

This document is provided “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Any publication of The Open Group may include technical inaccuracies or typographical errors. Changes may be periodically made to these publications; these changes will be incorporated in new editions of these publications. The Open Group may make improvements and/or changes in the products and/or the programs described in these publications at any time without notice.

Should any viewer of this document respond with information including feedback data, such as questions, comments, suggestions, or the like regarding the content of this document, such information shall be deemed to be non-confidential and The Open Group shall have no obligation of any kind with respect to such information and shall be free to reproduce, use, disclose, and distribute the information to others without limitation. Further, The Open Group shall be free to use any ideas, concepts, know-how, or techniques contained in such information for any purpose whatsoever including but not limited to developing, manufacturing, and marketing products incorporating such information.

If you did not obtain this copy through The Open Group, it may not be the latest version. For your convenience, the latest version of this publication may be downloaded at www.opengroup.org/bookstore.

Open Group Snapshot

Open Platform 3.0™

Document Number: S152

Published by The Open Group, October 2015.
Date of validity extended in June 2016 to January 31, 2017.

Comments relating to the material contained in this document may be submitted to:

The Open Group, Apex Plaza, Forbury Road, Reading, Berkshire, RG1 1AX, United Kingdom

or by electronic mail to:

ogspecs@opengroup.org

Contents

Preface

The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, and establish policies and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Offer a comprehensive set of services to enhance the operational efficiency of consortia
- Operate the industry's premier certification service

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Open Group Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/bookstore.

Readers should note that updates – in the form of Corrigenda – may apply to any publication. This information is published at www.opengroup.org/corrigenda.

This Document

This document is a Snapshot of The Open Group Open Platform 3.0™ Standard. It has been developed and approved by The Open Group.

The Open Platform 3.0 Standard will specify interoperable application platforms that will enable enterprises to gain business benefit from new technologies, including cloud computing, mobile computing, social computing, big data, and the Internet of Things (IoT). This is the second Snapshot of the Open Platform 3.0 Standard. It is a public draft that describes:

- The context in which the standard will apply
- The top-level business capabilities that it will require the platforms to have
- Technical capabilities that instantiate those capabilities
- Some realizations of one of the technical capabilities

- The basic ways in which platforms will interact with each other and with other system components

Chapter 1: Introduction provides an introduction to the Snapshot, indicates what the conformance requirements for the standard will be, and lists the standard terminology to be used in normative text.

Chapter 2: Definitions defines the key terms used in the Snapshot.

Chapter 3: Enterprise Ecosystems presents a reference model, key concepts, and principles for the ongoing evolution of ecosystems within extended enterprises.

Chapter 4: Wider Business Ecosystems presents informative considerations for wider business ecosystems that contain enterprises.

Chapter 5: Platform Capabilities lists the top-level business capabilities that a conforming platform will have or that multiple conforming platforms will have in combination, describes technical capabilities that instantiate the business capabilities, and gives an initial, partial description of how the technical capabilities can be realized.

Chapter 6: Basic Architecture Models describes a number of basic architecture models in which platforms interact with each other and with other system components.

Trademarks

ArchiMate®, DirecNet®, Making Standards Work®, OpenPegasus®, The Open Group®, TOGAF®, UNIX®, and the Open Brand (“X” logo) are registered trademarks and Boundaryless Information Flow™, Build with Integrity Buy with Confidence™, Dependability Through Assuredness™, FACE™, IT4IT™, Open Platform 3.0™, Open Trusted Technology Provider™, UDEF™, and the Open “O” logo and The Open Group Certification logo are trademarks of The Open Group in the United States and other countries.

ISACA® and COBIT® are registered trademarks of the Information Systems Audit and Control Association (ISACA) and the IT Governance Institute.

JavaScript™ is a trademark of Oracle Corporation.

OASIS™ and XACML™ are trademarks of OASIS.

All other brands, company, and product names are used for identification purposes only and may be trademarks that are the sole property of their respective owners.

Acknowledgements

The Open Group gratefully acknowledges the contribution of the following people in the development of this Snapshot:

- Stuart Boardman, KPN (Open Platform 3.0™ Forum Co-Chair)
- Don Brancato, HP
- Michael Brewer, IBM
- Thorbjørn Ellefsen, DIFI
- Kary Främling, Aalto University
- Ed Harrington, Conexiam
- Mike Jerbic, Trusted Systems Consulting
- Seshu Madabhushi, TCS
- Ron Schuldt, Data Harmonizing
- Ken Street, Conexiam
- Dennis Taylor, NASA
- Arnold van Overeem, Capgemini
- Tejpal (TJ) Virdi, the Boeing Company (Open Platform 3.0™ Forum Co-Chair)
- Bob Weisman, Build the Vision
- Stephen Whitlock, the Boeing Company

Referenced Documents

Normative References

Normative references for the Open Platform 3.0 Standard are defined in Section 1.4.

Informative References

These documents do not contain material that must be understood and used to implement the Open Platform 3.0 Standard, but will contribute to the reader's understanding of the standard and its context.

- An Information Architecture Vision: Moving from Data Rich to Information Smart, White Paper, W132, April 2013, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w132.htm.
- ArchiMate® 2.0 – Understanding the Basics, White Paper by Gerben Wierda, W130, February 2013, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w130.htm.
- K. Främling, S. Kubler, A. Buda: Universal Messaging Standards for the IoT from a Lifecycle Management Perspective, IEEE Internet of Things, Vol. 1, Issue 4, 2014, pp. 319-327, DOI: 10.1109/JIOT.2014.2332005.
- Industrial Internet Reference Architecture, the Industrial Internet Consortium; refer to: www.iiconsortium.org/IIRA.htm.
- Internet of Things – Architecture (IoT-A): Deliverables; refer to: www.iot-a.eu/public/public-documents.
- ISO 7498-2:1989: Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture; refer to: www.iso.org/iso/catalogue_detail.htm?csnumber=14256.
- ISO/IEC 10181-2:1996: Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems – Part 2: Authentication Framework; refer to: www.iso.org/iso/catalogue_detail.htm?csnumber=18198.
- NIST Cloud Computing Reference Architecture (CCRA); refer to: www.nist.gov/customcf/get_pdf.cfm?pub_id=909505.
- Open Platform 3.0™, Business Scenario, K130, October 2013, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/k130.htm.
- Real-World Data Mining: Applied Business Analytics and Decision-Making, Dursun Delen, Pearson FT Press, December 2014.

- The Nexus of Forces in Action – Business Use-Cases of Open Platform 3.0™, White Paper, W145, May 2014, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w145.htm.
- The Open Group IT4IT™ Reference Architecture, Version 2.0, Open Group Standard, C155, October 2014, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/c155.htm.
- The Open Platform 3.0™, White Paper, W147, May 2014, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w147.htm.
- [IETF RFCs]: Requests for Comments, Internet Engineering Task Force (IETF); refer to: www.ietf.org/rfc.html.

1 Introduction

1.1 Objective

The Open Platform 3.0 Standard will enable agile, secure, reliable, interoperable, and manageable multiple-technology solutions within and across enterprises.

It will do this by designating a set of common platform services that support the integration and interoperability of cloud computing, mobile computing, social computing, big data analytics, and the Internet of Things (IoT) computing paradigms, technologies, infrastructures, and applications across enterprises.

The standard will:

- Define the naming convention, description, and compositional structure of platform services at an architectural level of abstraction
- Define usage constraints of applicable platform services
- Advance the vision of Boundaryless Information Flow™ by establishing platform services that are standards-based, loosely coupled, accessible, secure, reliable, scalable, and manageable
- Support the emergence and optimization of new functional capabilities, business processes, architectures, and systems design patterns that would be difficult to achieve utilizing any single technology
- Establish guidelines, references, and exemplary practices that assist in the understanding of ecosystems and solutions that are based on Open Platform 3.0 and are not bound by any stakeholder's proprietary implementation approach or technologies
- Support the notation of vertical and horizontal integration by providing recommendations on the use of Open Platform 3.0 platform services for development, test, and production environments

1.2 Overview

The Open Platform 3.0 Standard focuses on new and emerging technology trends and computing paradigms converging with each other and leading to new business models and system designs. These trends currently include:

- Cloud computing
- Mobility
- Social networks and social enterprise

- Big data analytics
- The Internet of Things (networked sensors and controls)

Other technologies may be taken on board as the standard develops.

These convergent forces – united by the growing consumerization of technology and the resulting evolution in user behavior – offer the potential to create new business models and system designs. However, they also pose architectural issues and structural considerations that must be addressed for businesses to benefit.

Platform has traditionally been used to describe a single technology environment for a specific domain. Platform architecture models exist individually for domains such as cloud, mobile computing, social media, data analytics, and the Internet of Things. Such platforms currently tend to be used for creating organization or domain-specific information systems, which signifies that platforms mainly facilitate and create vertical integration. The collection of these individual platforms that conform to the Open Platform 3.0 Standard will in combination provide new capabilities for performing horizontal integration through use of open interoperability standards and best-practices.

The eventual Open Platform 3.0 Standard will state requirements for platforms, including that a platform shall provide the capabilities stated in Chapter 3: Platform Capabilities. It will also state further requirements for how a platform shall provide those capabilities, and for how it shall interface to other platforms so that the combination provides those capabilities.

1.3 Conformance

This is a Snapshot, not an approved standard. Do not specify or claim conformance to it.

Readers are advised to check The Open Group website for any conformance and certification requirements referencing the Open Platform 3.0 Standard.

1.4 Normative References

The documents referenced in this section contain material that must be understood and used to implement the Open Platform 3.0 Standard.

At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

- [ADOPTION]: The Open Group Standards Adoption Criteria; refer to: www.opengroup.org/standardsprocess/standards-adoption-criteria.html.
- [ARCHIMATE]: ArchiMate® 2.1 Specification, Open Group Standard, C13L, December 2013, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/c13l.htm.

- [CLOUD RM]: The Open Group Cloud Ecosystem Reference Model, Open Group Standard, C14I, January 2014, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/c14i.htm.

- [ISO/IEC 2382]: ISO/IEC 2382: 2015. Information technology – Vocabulary; refer to: www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63598.

- [ISO/IEC 11179]: ISO/IEC 11179:2004. Information Technology – Metadata Registries (MDR); refer to: www.iso.org/iso/catalogue_detail.htm?csnumber=35343.

- [NIST CLOUD DEFINITION]: NIST SP 800-145: NIST Definition of Cloud Computing; refer to: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

- [O-DF]: Open Data Format (O-DF), an Open Group Internet of Things (IoT) Standard, C14A, October 2014, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/c14a.htm.

- [O-MI]: Open Messaging Interface (O-MI), an Open Group Internet of Things (IoT) Standard, C14B, October 2014, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/c14b.htm.

- [TOGAF]: TOGAF® Version 9.1, Open Group Standard, G116, December 2011, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/g116.htm.

- [XDSF]: Distributed Security Framework (XDSF), Open Group Guide, G410, December 1994, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/g410.htm.

- [XACML]: eXtensible Access Control Markup Language (XACML), Version 2.0, OASIS, 2005; refer to: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

1.5 Terminology

For the purposes of the Open Platform 3.0 Standard, the following terminology definitions apply:

Can Describes a possible feature or behavior available to the user or application.

May Describes a feature or behavior that is optional. To avoid ambiguity, the opposite of "may" is expressed as "need not", instead of "may not".

Shall Describes a feature or behavior that is a requirement of the standard. To avoid ambiguity, "must" is not used as an alternative to "shall".

Shall not Describes a feature or behavior that is an absolute prohibition of the standard.

Should Describes a feature or behavior that is recommended but not required.

Will Same meaning as "shall"; "shall" is the preferred term.

1.6 Future Directions

This Snapshot builds on the first Snapshot towards creating a complete description, which will be refined over time resulting in the recommended standard.

The Open Platform 3.0 Forum intends to release further snapshots as development of the platform proceeds, prior to the publication of the first standard description of Open Platform 3.0.

These Snapshots are indications of what the platform might be, and invitations for input and comment. The Forum will develop the platform in the light of such input and in accordance with the evolving views of its members. This and succeeding Snapshots are not guaranteed to be consistent with each other or with the eventual standard. They should not be used in procurement specifications by customers or in claims of conformance by vendors.

2 Definitions

For the purposes of this standard, the following terms and definitions apply. Merriam-Webster's Collegiate Dictionary should be referenced for terms not defined in this section. (Note that model elements are defined in Chapter 6.)

2.1 Capability

An ability that an organization, person, or system possesses. [TOGAF]

2.2 Data

A re-interpretable representation of information in a formalized manner suitable for communication, interpretation, or processing. [ISO/IEC 2382]

Data can be processed by humans or by automatic means.

2.3 Data Analytics

The method of logical analysis applied to *data*.

Note: Merriam-Webster defines *analytics* as “the method of logical analysis”.

Analytics facilitates realization of business objectives through reporting of data to analyze trends, creating predictive models for forecasting, and optimizing business processes for enhanced performance. [DURSUN DELEN]

2.4 Data Element

1. A unit of data that is considered in context to be indivisible. [ISO/IEC 2382]
2. A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes. [ISO/IEC 11179]

2.5 Data Element Concept

A concept that can be represented in the form of a data element, described independently of any particular representation. [ISO/IEC 11179]

2.6 Ecosystem

A network of participating entities, each of which plays one or more roles to achieve targeted objectives.

Ecosystem participants are not necessarily aware of all other entities in the ecosystem but will in general affect or be affected by them. An ecosystem is subject to the effects of both internal and external factors.

2.7 Enterprise

A collection of organizations that has a common set of goals. [TOGAF]

For example, an enterprise could be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership.

2.8 Enterprise Ecosystem

An ecosystem maintained by an enterprise, whose participating entities are within or associated with the enterprise, that effectively manages product and service lifecycles, and supports a seamless integration between technical capabilities and business opportunities to meet business needs.

2.9 Identification

The assignment of a name by which an entity can be referenced. The entity may be high level (such as a user) or low level (such as a process or communication channel). [XDSF]

2.10 Interoperability

3. The ability to share information and services.
4. The ability of two or more systems or components to exchange and use information.
5. The ability of systems to provide and receive services from other systems and to use the services so interchanged to enable them to operate effectively together. [TOGAF]

2.11 Platform

A combination of technology infrastructure products and components that provides the prerequisites to host application software. [TOGAF]

2.12 Platform Service

A technical capability required to provide enabling infrastructure that supports the delivery of applications. [TOGAF]

2.13 Principal

An entity whose identity can be authenticated (see ISO/IEC 10181-2:1996). [XDSF]

2.14 Real Time Response

A response to an event with very low latency; usually within seconds of the event occurrence.

3 Enterprise Ecosystems

Major advances in technology and processes have created a highly disruptive “Digital Economy”, bringing about both business opportunities and business risks, forcing enterprises to innovate or face the consequences. Many enterprises are not well prepared for the rapid pace of digital disruption that requires continual change as a reality of the digital economy. An enterprise must maintain an agile ecosystem that effectively manages product and service lifecycles, and supports a seamless integration between technical capabilities and business opportunities to meet business needs.

Within such an ecosystem, the Open Platform 3.0 standard enables an agile architecture for the development of enterprise business solutions. These solutions take advantage of a wide range of distributed capabilities utilizing modern technology such as cloud computing, social computing, mobile computing, big data analytics, and embedded systems with sensing and/or actuation capabilities. They have appropriate security policies enabling actionable information to be made available to eligible participants (human, non-human, or both).

3.1 Enterprise Ecosystem Reference Model

Architecture includes not only the structure of the components of a system and their inter-relationships, but also the principles and guidelines governing their design and evolution over time. The Open Platform 3.0 enterprise ecosystem reference model shows the key artifacts related to the evolution of enterprises and their ecosystems over time, rather than to the arrangement of their components. It is shown in the figure (Figure 1).

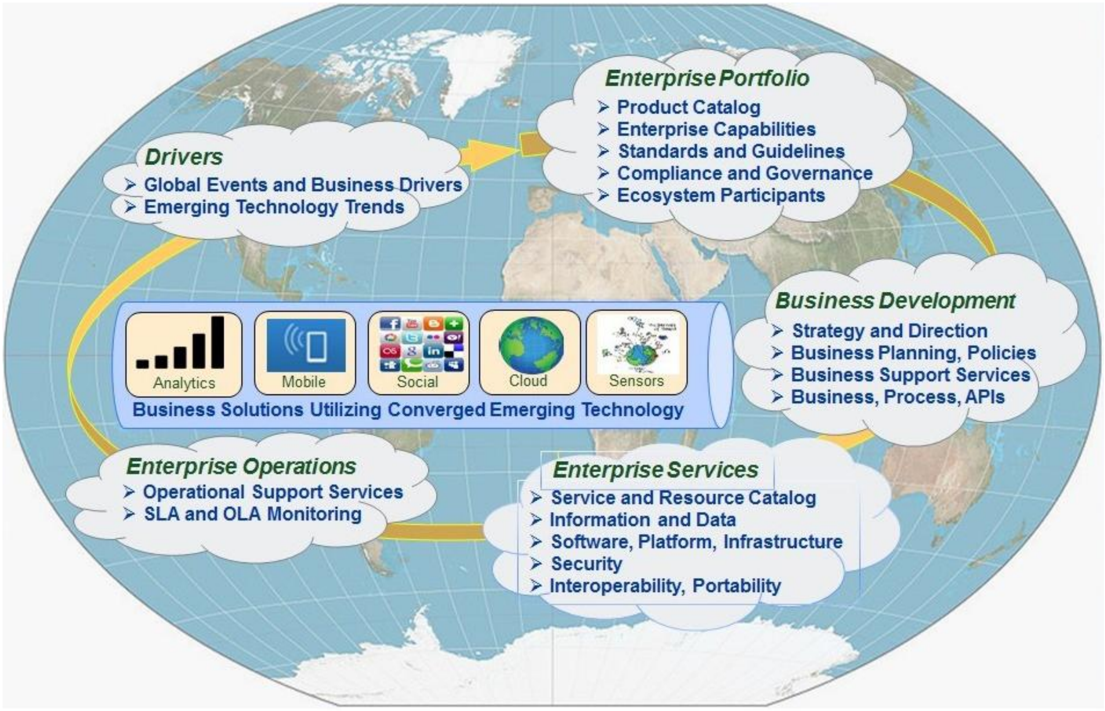

Drivers
Global Events and Business Drivers
Emerging Technology Trends
Enterprise Portfolio
Product Catalog
Enterprise Capabilities
Standards and Guidelines
Compliance and Governance
Ecosystem Participants
Analytics
Mobile
Social
Cloud
Sensors
Business Solutions Utilizing Converged Emerging Technology
Business Development
Strategy and Direction
Business Planning, Policies
Business Support Services
Business, Process, APIs
Enterprise Operations
Operational Support Services
SLA and OLA Monitoring
Enterprise Services
Service and Resource Catalog
Information and Data
Software, Platform, Infrastructure
Security
Interoperability, Portability

Figure 1: The Enterprise Ecosystem Reference Model

Conforming platforms support the ecosystem and enable the enterprise to utilize combination(s) of the emerging technologies.

3.2 Key Concepts

The key concepts of the reference model are outlined below.

3.2.1 Drivers

Enterprises should have structured processes for identifying new opportunities and evaluating emerging technologies that enable them to create innovative business solutions and respond to changes in the ecosystem. They need to deliver and continue to deliver the intended value of these solutions. The processes include assessments of how emerging trends and technology will impact an enterprise's resources, products, services, prioritization, justification for investment decisions, and ability to deliver according to promised performance levels. Other factors affecting this include external events at local and global level, changes in resource availability, disruptive innovation, and changes in business models elsewhere in the ecosystem.

3.2.2 Enterprise Portfolio

Depending on the emerging trends, and their capabilities to realize the opportunities presented by those trends, enterprises apply suitable funding models (e.g., full-scale in-house investment, partnering with the ecosystem participants, etc.) to address gaps in capabilities. Decision-making and governance processes to track opportunities must be applied.

3.2.3 Business Development

Opportunities presented by emerging trends require the right skills, knowledge, and understanding of the risks. Impact assessments help enterprises not only to build and enhance their capabilities, but also to define the right approach for business development. Risk and impact assessments predict where the opportunities are most likely to create unfavorable results, allow enterprises to eliminate barriers to success, and effectively address the gaps in their capabilities.

3.2.4 Enterprise Services

Enterprises should build competence around modular service-based architectures and solutions to enable their agile plug-and-play enterprise ecosystems to tap potential business opportunities effectively, and to mitigate the high risk of investment in emerging technology. The constant evolution of an enterprise’s ecosystem capabilities (especially platform services and practices that simplify the development of business solutions) that can quickly and easily be leveraged, modified, implemented, and tested help enterprises to realize new business opportunities effectively. These proven services dramatically reduce cost and risk, and insulate business solutions from underlying technological complexity.

3.2.5 Enterprise Operations

An enterprise should define an appropriate level of competence for its business and operational ecosystem capabilities. It should have effective mechanisms to mature different sets of skills, tools, and processes, for real-time performance monitoring, and to provide actionable information for a continuous evolution of its capabilities.

3.3 Enterprise Ecosystem Metamodel

The Enterprise Ecosystem Metamodel shown in the figure below provides an effective mechanism to manage dependencies, inter-relationships, business interactions between participating entities, and needed support for traceability of enterprise capabilities across the various entities of an enterprise ecosystem.

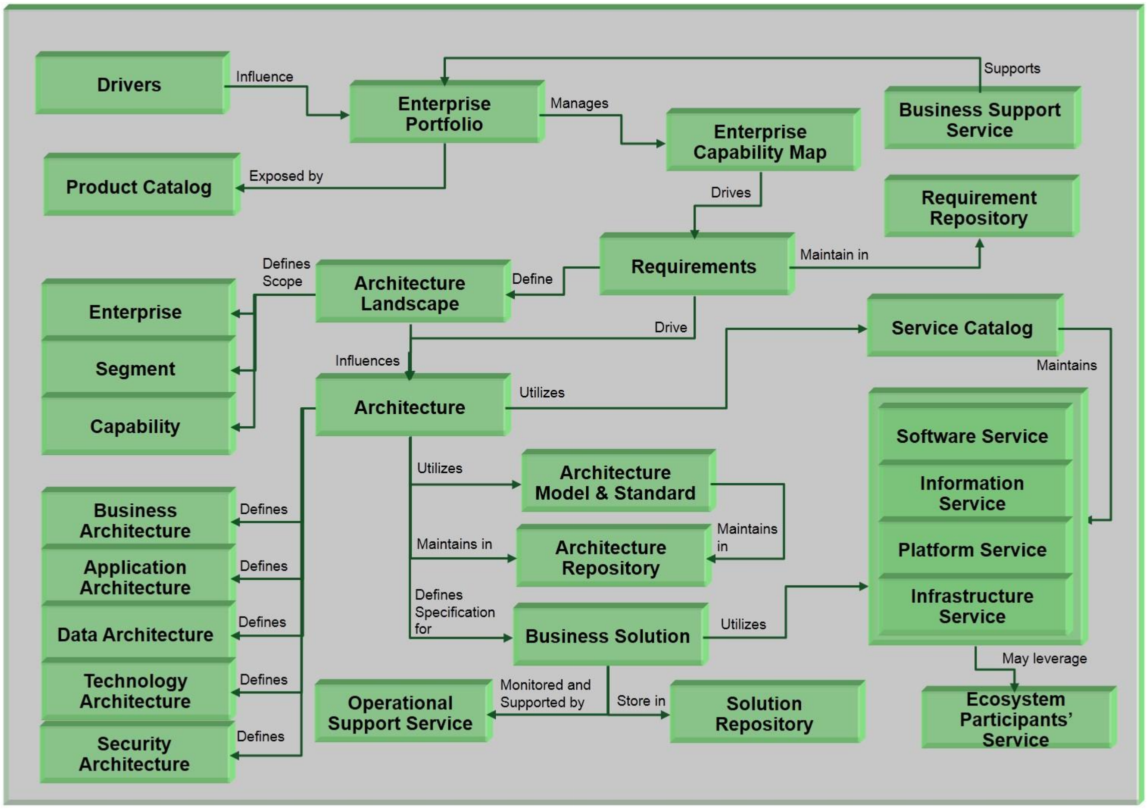


Figure 2: The Enterprise Ecosystem Metamodel

The Enterprise Ecosystem Metamodel extends the TOGAF Metamodel with enterprise-ecosystem-related enabling concepts. The Metamodel provides a common set of semantics that can be utilized by all participants of an enterprise ecosystem. Depending on the scope of business interactions between the participants, there could be many views of the Metamodel. The important feature of the Metamodel is that it must allow the greatest degree of flexibility in utilizing emerging technologies for innovations in order to achieve a higher level of business agility. It also enables enterprises to evolve their capabilities without compromising the integrity of business information that is being shared. To address evolving business needs, any change in relationships between an enterprise and its ecosystem's participating entities (providing support to enterprise capabilities) must be easily assessed and effectively managed.

The assessment of business requirements, maturity of dependent capabilities, and impacting factors such as applicable global business rules and regulations will determine an enterprise's approach to enabling business capabilities of its ecosystem. Such an assessment could provide insights as to whether the enterprise should develop new business capabilities internally or leverage the expertise of new or existing participating entities of its ecosystem. In any case, the approach should ensure that all enabling capabilities of its ecosystem are seamlessly and reliably interoperable from one participating entity to another without adversely impacting the expected Service Level Agreements (SLAs).

3.4 Principles

Principle	Bring together emerging trends and technologies to deliver business value
Statement	The enterprise can develop business solutions using the convergence of relevant emerging trends and technologies.
Rationale	In order to meet business objectives to maximize profit, enterprises should be able to seamlessly bring together emerging technologies into an enterprise ecosystem to develop business solutions.
Implications	(1) An enterprise ecosystem supports a built-in holistic approach that is conducive to emerging trends and technologies. (2) An enterprise ecosystem supports mechanisms that determine what business capabilities are required to deliver business value. (3) An enterprise ecosystem supports progressive evolution of business solutions by infusing emerging trends and technologies, and have or support mechanisms to measure business value performance of an enterprise capabilities. (4) An enterprise ecosystem leverages enterprise business capability maturity assessments to effectively evolve business capabilities.

Principle	Evolution of enterprise frameworks and practices
Statement	New practices and frameworks introduced into the enterprise ecosystem are complementary to and supportive of its existing practices and frameworks.
Rationale	New practices and frameworks should be easily and seamlessly plugged into an enterprise ecosystem.
Implications	(1) The enterprise can tailor new practices and frameworks to achieve an optimal compromise with existing ones. (2) The enterprise has a mechanism to seamlessly represent any specific need to change existing practices and frameworks. (3) The enterprise is aware of and minimizes the impact of new architectures and practices on its existing ones.

Principle	Integration of new business practices
Statement	The enterprise has an effective mechanism to integrate new business innovation guidelines and practices into its ecosystem.
Rationale	This is needed for a pragmatic development of enterprise capabilities for timely seize of business opportunities.

Principle	**Integration of new business practices**
Implications	(1) The enterprise can utilize assessment mechanisms to obtain information to help it evolve existing processes, roles, responsibilities, and architectural skills for its ecosystem. (2) The enterprise has an effective transformation process to evolve its ecosystem and minimize business, technical, and operational constraints.

Principle	**Common platform**
Statement	The enterprise has common, foundational, and extensible capabilities to consistently develop and evolve business solutions.
Rationale	Common and extensible services maximize reusability and minimize duplication. A common foundation enables uniform application of concepts and practices and avoids ambiguities. Core and foundational capabilities are needed to facilitate and support effective communications within the enterprise ecosystem.
Implications	(1) The common foundation has (but is not limited to) capabilities for security, scalability, reliability, and supportability. (2) Where applicable, existing capabilities should be used or evolved.

4 Wider Business Ecosystems (Informative)

An enterprise can act as an entity within the ecosystem of another enterprise, or within a wider business ecosystem that is not maintained by any particular enterprise. In such a wider ecosystem, each enterprise makes services available for use by other enterprises.

An enterprise ecosystem can participate in more than one wider ecosystem. For this reason a wider ecosystem does not have to involve all of the services of each of its enterprises or all parts of their enterprise ecosystems – only those parts that are relevant to the purpose of the wider ecosystem.

The Open Platform 3.0 standard is concerned with scenarios where new and emerging technology trends are in use (e.g., IoT devices providing data used by a data analytics service). The services provided by these technologies may involve multiple providers. An example scenario is described in Appendix B.

4.1 Participation

An enterprise typically belongs to more than one ecosystem, if only because it delivers more than one type of service. This in itself is a reason that an enterprise's strategic relationship to any one ecosystem may change. In some cases an ecosystem may consist exclusively of enterprises that have an agreed common goal. In general, however, each enterprise has its own interests in the ecosystem, which therefore determine the extent and scope of its commitment to the ecosystem as a whole.

In the biological world a common example of an ecosystem is a pond. All of the participants have an interest in the continued health of the ecosystem. Some of them, however, are in a position to leave the pond (think frogs or birds) for another that offers better prospects. If all the frogs leave the pond, the ecosystem will change drastically – but to attempt to predict the result would be to trivialize the way ecosystems work. The ecosystem will figure out how to compensate or it won't and will die or totally transform.

It is the same in a business ecosystem. Every participating enterprise has to be concerned about the health of the whole thing but has its own level of commitment and options. Enterprises are human organizations and are therefore capable of reasoning in advance about these things. That allows them to take into consideration the possible behavior of types of participant, of whose concrete existence they may not be aware.

Taking the general case, we can say that, by defining a business ecosystem, we have identified a group of participants (organizations, individuals, things), whose interactions are the scope of the ecosystem. All the participants have an interest in the health of the system but their concept of value within and commitment to the ecosystem and their specific dependencies vary from one participant to the next. Each participant, therefore, has to construct its own view of all these factors (functional and non-functional), the degree of uncertainty associated with them, and its own risk and mitigation strategy in that context.

4.2 Service Catalogs

The service catalog is central to relationships between parties in such an ecosystem. The contractual, on-demand nature of service aggregation requires on-demand contracts, whether by system or human users. One enterprise's catalog will include and/or be dependent on services provided by other enterprises. Each party has its own catalog and the combination of these catalogs is where the whole ecosystem works.

This means that all aspects of the enterprise ecosystem – see Enterprise Ecosystem Reference Model (Section 3.1) – that are relevant to the service under consideration are dependent on parts of the service delivered by other enterprises participating in the business ecosystem.

4.3 Dependencies

A dependency may be indirect. For example, consider three enterprises A, B, and C, where Enterprise A offers a business analytics service, which is used by Enterprise B, and draws data from a data aggregation service provided by Enterprise C. An element of enterprise operations offered by Enterprise A is dependent on one offered by Enterprise C (which may or may not be covered by some form of contract or SLA).

The ability of any enterprise to deliver according to expectations depends on factors such as resource availability and cost, applicable laws and regulations, and the enterprise's strategy, all of which are liable to change. Enterprise A is indirectly dependent on these factors as they affect Enterprise C. The degree of risk associated with this dependency will be determined by the nature of the relationship between the parties, which may be a supplier/consumer contract, a partnership, or simply a usage relationship (e.g., using a public service's API).

Enterprise B is a consumer of Enterprise A's service and is therefore also dependent on the service of Enterprise C. Enterprise B has no direct relationship with Enterprise C but is dependent on the quality and reliability of its service. It is in general advisable to obtain as much information as possible about a consumed service and the conditions affecting its delivery. Where such information is not available, Enterprise B may be able to impose (auditable) contractual obligations on the use of third parties. In any event it has to develop a strategy to determine its risk appetite and deal with a level of uncertainty.

5 Platform Capabilities

This chapter lists the business and technical capabilities that a conforming platform must or can have, and describes some realizations of the technical capabilities.

A capability is *mandatory* if a platform "SHALL" provide it. A capability is *optional* if a platform "CAN" rather than "SHALL" provide it. The documentation of a platform SHALL state clearly which optional capabilities it does provide.

A platform SHALL have a description of the realization of each mandatory capability and of each optional capability that it provides. This description, and any standards or specifications that it references, SHALL be open: it SHALL meet The Open Group Standards Adoption Criteria. [ADOPTION]

Any capability can be realized in a number of ways. How capabilities are realized in different platforms affects interoperability. It is important that realizations use the same standards, and are compatible with each other. This Snapshot does not constrain realizations, but it does require that they be documented so that, even if interoperability cannot be achieved "out of the box", it can at least be programmed.

Note: It is intended that the Open Platform 3.0 standard will include standard realizations. This Snapshot includes a small sample, to illustrate that intention.

5.1 Business Capabilities

These are capabilities that are deemed necessary (though not necessarily sufficient) to the achievement of business objectives.

5.1.1 Support for Services

Service Design

A platform CAN provide facilities for service design.

Service Deployment

A platform MUST support the deployment of services.

Service Operation

A platform MUST support the operation of services.

Discovery by Users

A platform MUST provide a means by which appropriately-authorized users can discover what services it supports.

Discovery by Services

A platform CAN provide a means by which appropriately-authorized other platform or service instances can discover what services it supports.

Service Catalog

A platform CAN provide a service catalog, which describes services that its instances support and/or services that other platform instances support. If a service catalog is not provided, human local knowledge or system intelligence for how to select, size, model, and pay for a service is required.

Many vendors may imagine the service catalog as the entry into Open Platform 3.0 from a business capability perspective.

5.1.2 Service Composition

This is the composition of business services (from existing and supported emerging technologies) that may involve both human and machine interaction, multiple locations, and multiple providers, to create a new business service.

Service Composition

A platform CAN enable appropriately-authorized users to compose services, including by service orchestration and service choreography.

5.1.3 Business Data Analysis

These are capabilities to analyze data so that a user or service can discover and communicate insights for knowledge and decision-making purposes.

Analytics is key to developing a purposeful and trustworthy information-driven means to empower decision-making. It can span structured, semi-structured, and unstructured data.

Guided Data Acquisition

A platform CAN enable a user or a service to discover, link, and synthesize relevant data from diverse sources (existing or new, within or outside of the enterprise, and increasingly on-the-fly) with minimal user guidance moving from data *in situ* to its curation for downstream use.

Acquisition needs could involve diverse types and formats of data. Guided acquisition tends to involve humans and tools to engineer, extract, and establish relevance of features in data.

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

Data Exploration

A platform CAN enable a user or a service to reduce the amount of data to examine so as to focus on evaluating key metrics and measures (as in KPIs) for business reporting or for visualization.

Exploration may involve the use of:

- Descriptive statistics to characterize data distribution and dispersion
- Techniques to identify possible relationships among two or more items
- Visualization for reporting

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

Data Persistence

A platform CAN enable a user or a service to store, retrieve, and manage large volumes of diverse types of data.

This could involve smart storage and retrieval technologies and related aspects, such as capacity planning, transport network, integration, protection, backup, and archival.

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

Velocity Adaptation

A platform CAN handle variability in speeds and feeds (variation in data velocity).

There is a need to manage variances in data velocities at required scale in a timely manner. The historical solution approach to handling velocity was mostly based on separating writes and reads – this may not be adequate in the case of big data.

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

Stream Processing

A platform CAN process large-scale streaming data and provide near real-time response.

This capability enables analysis of data in motion. It is required for processing real-time streaming data or event data where the ratio of event throughput to number of queries is usually high.

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

5.1.4 Business Data Reporting

This is a capability to support business management reporting, accountability, traceability, and governance, and to make business information available for system-of-record support, including information for billing, inventory, and regulatory reporting.

Analytics Reporting

A platform CAN provide reports showing data analytics results.

5.1.5 Data Conveyance

These are capabilities to convey information between devices and services.

Seamless exchange of information across organizational or system boundaries is crucial to The Open Group vision of Boundaryless Information Flow™. It implies the ability to transport information for any end-to-end business scenario.

Information distribution may be viewed as a special case of information exchange.

Different platforms or parts of platforms may be optimized for particular forms of communication. Information needs to be transported transparently, using multiple standard network technologies and protocols. This results in the individual technologies being irrelevant/invisible to the business.

Information to be transported may provide notification of events that have occurred, or actions to be taken. Such notifications need to be given consistently at all interfaces where they are presented.

The following requirements apply to the realization of the capabilities in this section.

- **Data Representation**: The documentation of an interface by which a platform receives or delivers data SHALL describe how data elements are represented within the data.
- **Network-Neutral Data Interface**: A platform SHALL be able to receive data from and deliver data to devices and services in a form that is independent of the networks and protocols used to transport the data.
- **Device-Neutral Data Interface**: A platform SHALL be able to deliver data received from devices to services and receive data from services for delivery to devices in a form that is independent of the devices concerned.

Data Conveyance within a Platform

A platform SHALL be able to convey data between any services that it supports and any devices connected to it.

Data Conveyance with Other Conforming Platforms

A platform SHALL be able to convey data between any service that it supports or device connected to it and any other conforming platform that supports a service or is connected to a device producing or consuming the data.

Data Conveyance with Non-Conforming Platforms

A platform SHOULD be able to convey data between any service that it supports or device connected to it and non-conforming platforms (for example, legacy platforms), as well as with conforming platforms.

Note: This clearly depends on support for the appropriate interfaces by the non-conforming platforms concerned.

5.1.6 Intellectual Property

A platform SHALL have a means of preserving intellectual property.

5.1.7 Governance

Note: Governance capabilities are expected to form an important part of this section ultimately. At this point, no specific governance capabilities are defined.

The following requirements and considerations apply.

- A platform must enable appropriately authorized identities to institute the desired behaviors needed for effectively managing information/knowledge assets, and related business processes across the ecosystem(s) in focus by establishing decision rights, accountabilities, and mechanisms for change.
- Effective governance ensures proper alignment and coordination across parties, compliance, and mitigation of risks. It involves lifecycle processes around a firm's information/data assets.
- Existing governance frameworks like the ISACA COBIT and emerging reference models like The Open Group IT4IT™ standard should be considered as applicable, and conforming platforms should be sensitive to the need to trace and report value to the business and comply with existing and future regulatory demands for managing risk.

5.2 Technical Capabilities

These are capabilities that are defined in technology terms and that are needed for the instantiation (in whole or in part) of business capabilities.

Note: A technical capability must always be driven by one or more business capabilities.

They include capabilities related to system operation, and capabilities that support the maintenance of good operational characteristics in a system of which a platform forms a part, such as safety, security, resilience, quality of service, and quality of data.

5.2.1 Event Handling

Business solutions can require the correlation of events from multiple sources. Also, system event correlation can be critical to assure SLA/OLA compliance, and so to quality of operation.

A possible event model is described in Appendix B.

Event Correlation

A platform CAN have the ability to correlate events seamlessly, real-time, from multiple sources (including cloud computing, mobile devices, social media, big data, and IoT).

5.2.2 Semantic Consistency

This is the ability to enable consistent business and technology semantics. Seamless information exchange relies on integration and interoperability methods that effectively address nomenclature, syntax, format and structure, and semantic mismatches.

In a business or social ecosystem, different participants create and process data in the light of different information models. Meaningful information exchange requires consistent interpretation of the data across different models. This implies a multi-enterprise data governance approach that is much more difficult than the approach available within a single enterprise.

Semantic Analysis

The documentation of a service or device that delivers or receives data may describe the data element concepts that its data elements represent. A platform CAN provide a means by which users or services can determine whether described data element concepts are equivalent or similar.

Data element concepts are often described in terms of semantic models, which can take many different forms and formats. Within an ecosystem, they will likely span a wide spectrum from Entity-Relationship-Attribute (ERA) forms used for traditional data models to the subject-predicate-object Semantic Web triples in Resource Description Framework (RDF) and Web Ontology Language (OWL) forms. Even when different participants use the same modeling technique within an ecosystem, they can produce radically different models.

5.2.3 Resource Management

A platform that has resource management capabilities SHOULD enable them to be applied in accordance with policies set by appropriately-authorized users.

Parallel Processing

A platform CAN have the ability to utilize parallelism.

Parallelism can give significant gains in processing speed, and help speed up the overall process, from ingestion to insight, especially with large data volumes.

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

Load Apportionment

A platform CAN provide a means of apportioning load between computing resources.

Resource reservations and load balancing CAN include migrating load between platform instances.

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

Cluster Deployment

A platform CAN enable a user or system to set up a service infrastructure of hardware clusters to process workloads (including big data workloads).

Note: Many big data projects involve as the first step both set-up and maintenance of an infrastructure cluster. Some platform providers utilize Infrastructure as a Service (IaaS) models to cut down time and effort involved.

Note: This capability is derived from the draft Business Data Lake specification, and is subject to revision as that specification is further developed.

5.2.4 Identity, Entitlement, and Access Management

Identity, entitlement, and access management is the security discipline that enables the right individuals to access the right resources at the right times for the right reasons. This is also applicable for devices, applications, or any other entity that requires access to information or a system.

Principal Identity Determination

A platform CAN enable a service instance to determine reliably the identity of any principal.

This typically relies on authentication of principals.

Principal Role Determination

A platform SHALL enable a service instance to determine reliably the roles (for example, “staff” or “public”) that a principal has.

This typically relies on authentication of principals.

Directory

A platform CAN maintain a directory of users, devices, applications, or any other entity that requires access to information or system, and provide services with information about them.

Note: When a set of conforming platforms is used by an enterprise or in an ecosystem, one of them should have a directory, which can be accessed by the others.

Access Control

A platform SHALL enable immediate, easy, and secure access to services or data to authorized principals, and SHALL deny access to principals that are not authorized.

Authorization

A platform SHALL ensure that authorization is granted in accordance with policies set by appropriately authorized users. Roles and personas CAN be attributes used by the platform in this context.

5.2.5 Security

Note: Security capabilities are expected to form a major part of this section ultimately. At this point, no specific security capabilities are defined.

The following security requirements and considerations apply.

- Security capabilities must support provision of appropriate security and privacy as information passes from (and is possibly transformed by) services, in order to enable appropriate and adequate trust in the resulting information that is utilized in the new business services (and potential business models).
- Further, security is assured through enterprise and service monitoring and event alerting through any one of the cloud computing, mobile devices, social media, big data, or IoT capabilities by asserting that Open Platform 3.0 has an enterprise event correlation capability.
- Security levels and compliance requirements must be able to be defined and implemented consistently and interoperably throughout an ecosystem. This means, in particular, that the representations of roles or other access management-related attributes with common semantics can be mapped across different parts of the ecosystem.

5.2.6 Audit

Audit capabilities contribute to security and governance.

Audit Trail

A platform SHALL have means to capture any activity from a user, device, application, or any other entity to support policy performance reporting, policy compliance, and policy deviation. This will include privileged user account monitoring and privilege creep.

5.2.7 Quality of Service (QoS)

Platform QoS Attributes

A platform SHALL expose standard non-functional and supportability QoS attributes for its own operation.

Service QoS Attributes

A platform SHALL maintain records of the non-functional and supportability QoS attributes that it exposes for its own operation and of those exposed by the services that is supports.

Quality of Service Reports

A platform SHALL provide reports of the QoS attributes that it records.

QoS Exception Events

A platform SHALL be able to create an exception event when there is a variance beyond set limits of any supported attribute. The limits and whether an event is created for any particular breach SHALL be determined by policies set by appropriately-authorized users.

The variance SHOULD be addressed, where possible, without disruption at the service layer, so that the attribute returns to expected policy levels as soon as practicable.

5.2.8 Quality of Data

Quality of Data Attributes

A platform SHALL maintain records of the data quality attributes of the data that it makes available to users and services.

Quality of Data Reports

A platform SHALL provide reports of the quality of data attributes that it records.

5.2.9 Management/Monitoring

Management by Users

A platform SHALL enable appropriately-authorized users to monitor and control the services that it supports.

Management by Services

A platform SHOULD enable appropriately-authorized service instances to monitor and control the services that it supports.

Platform Self-Management

A platform SHOULD monitor and reliably report its state of health, and take action when a report warrants this.

5.2.10 Upgradability

Non-Disruptive Assimilation

A platform SHOULD support non-disruptive assimilation of upgrades and new components by itself and by the services that it supports. If it cannot support non-disruptive assimilation, it SHALL support scheduled assimilation of upgrades and new components.

Note: This enables continual service improvement.

5.2.11 Policy-Driven Platform

Platforms and other system components typically act as policy enforcement points for access control and other policies. This section describes capabilities that support policy-driven operation. The underlying concepts are described in [XACML].

For example, a user, system, or machine selecting a service from a service catalog may select options for time, service level, and model size, and the chosen options are enforced by policy whether the service is executed in-house, outsourced, or multi-sourced. The user has and should expect that the chosen SLA is protected by policy that enforces OLA compliance on outsourced or multi-sourced service delivery, no matter what "depth" the OLA outsource or multi-source has derived.

Note: [XACML] defines a framework for policy-driven access control, but its definitions of PAP, PDP, and PIP are framed in general terms, and are to be understood here as potentially applying to other kinds of policy. (Policy Retrieval Point (PRP) appears to be a recent introduction. It is not in the XACML 2.0 standard, and is not used here.)

Policy Administration

A platform CAN act as a Policy Administration Point (PAP).

Policy Decision

A platform CAN act as a Policy Decision Point (PDP).

Policy Information

A platform CAN act as a Policy Information Point (PIP).

5.3 Capability Realizations

This section describes specific realizations of the capabilities described earlier in this chapter.

A platform NEED NOT implement the realizations described here for the capabilities that it has, except where the realization is explicitly stated to be mandatory. The documentation of a platform SHALL state which realizations it does implement, and SHALL describe all of its realizations (including those not described here) so that interoperable systems can be developed independently.

Note: Only four realizations, and only two capabilities, are described in this Snapshot. This illustrates the concept of capability realization. It is intended that the Open Platform 3.0 standard will describe more realizations.

5.3.1 Device Data Exchange

This section describes realizations of the Device Data Exchange capability for Data Conveyance (Section 5.1.5).

Realization 1

The platform functions as the Internet Object User Platform of the Internet Object Model (Section 6.7.1) and communicates with the Internet Object using the Open Messaging Interface [O-MI] over HTTP, with payload formatted in accordance with the Open Data Format [O-DF].

Realization 2

The platform functions as the Managed Object User Platform of the Managed Object Model (Section 6.7.2) and communicates with the Object Manager Platform using the Open Messaging Interface [O-MI] over HTTP, with payload formatted in accordance with the Open Data Format [O-DF].

Realization 3

The platform functions as the Object Manager Platform of the Managed Object Model (Section 6.7.2) and communicates with the Managed Object using the Open Messaging Interface [O-MI] over HTTP, with payload formatted in accordance with the Open Data Format [O-DF].

5.3.2 Access Control

This section describes a realization of the Access Control capability of Identity, Entitlement, and Access Management (Section 5.2.4). This implementation is mandatory.

The platform acts as a Policy Enforcement Point (PEP) as defined in [XACML].

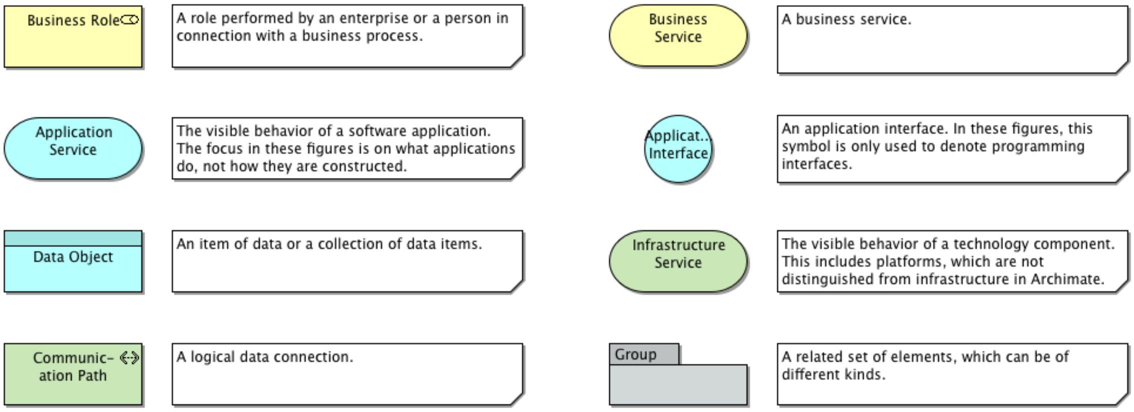
6 Basic Architecture Models

This chapter describes a number of basic architecture models in which platforms interact with each other and with other system components. They may be used in the eventual standard as the basis for stating normative requirements on conforming platforms.

6.1 Notation

Many of the figures in this chapter are expressed in the ArchiMate modeling language. The ArchiMate specification [ARCHIMATE] is an Open Group Standard for an open and independent modeling language for Enterprise Architecture. There is a good explanation of it in the ArchiMate® 2.0 – Understanding the Basics White Paper.

The figures in this chapter use a small subset of the ArchiMate symbols. They are shown and explained below (in Figure 3).



A role performed by an enterprise or a person in connection with a business process.

A business service.

The visible behavior of a software application. The focus in these figures is on what applications do, not how they are constructed.

An application interface. In these figures, this symbol is only used to denote programming interfaces.

An item of data or a collection of data items.

The visible behavior of a technology component. This includes platforms, which are not distinguished from infrastructure in Archimate.

A logical data connection.

A related set of elements, which can be of different kinds.

Figure 3: ArchiMate Notation

6.2 Applications

6.2.1 Application Model

The distinction between applications and systems software is one of the oldest in computing. Applications form part of the Technical Reference Model (TRM) defined in [TOGAF]. A modified form of this model, including a business layer and a more broadly scoped infrastructure layer, is shown in the figure (Figure 4).

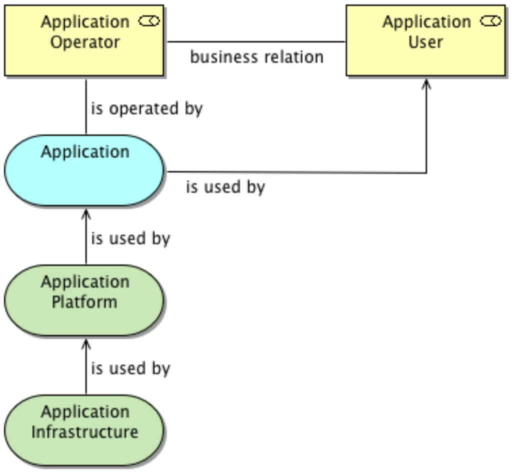


Figure 4: The Application Model

An *application* (as defined in [TOGAF]) is a deployed and operational IT system that supports business functions and services; for example, a payroll. Applications use data and are supported by multiple technology components but are distinct from the technology components that support the application.

The *operator* of an application is an enterprise or person that manages it and makes it available for use.

The *application users* are people who use the application. They often have a business relationship with the operator; for example, they may be employed by the operator. They include people that use the application in order to manage or maintain it, as well as people who use the application for business purposes (end users). Users are treated individually; different users can access different information, and may have rights to access different facilities.

The *application platform* (again as defined in [TOGAF]) is the collection of technology components of hardware and software that provide the services used to support applications.

The *application infrastructure* includes communications infrastructure, processing, storage, and possibly other infrastructure providing basic IT capabilities, but not part of the application platform. The application platform uses the infrastructure and makes its services available to the application. (Many definitions of *infrastructure* include what is defined here as the application platform. In this Snapshot, the platform is considered to be separate from the infrastructure, not a part of it.)

6.2.2 Web Application Model

With the advent of the World-Wide Web, a particular form of the basic model emerged, in which the user accesses the application across the World-Wide Web, using a web browser. This is shown in the next figure (Figure 5).

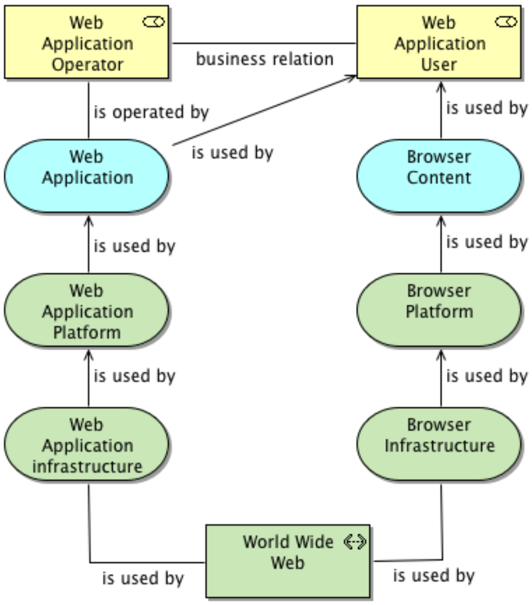


Figure 5: The Web Application Model

The *web application operator*, *web application user*, *web application*, *web application platform*, and *web application infrastructure* can be considered as the application operator, application user, application, application platform, and application infrastructure of the Application Model. The *web application infrastructure* includes communications infrastructure to access the World-Wide Web.

The web application user interacts with the web application using *browser content* that consists of HTML with scripts (e.g., written in Javascript) and applications (known as *applets*) downloaded from the server.

The *browser platform* is the client-side web platform. It supports the browser content and enables the user to interact with the web application through that content. It uses the *browser infrastructure*.

The *browser infrastructure* consists of communications and other infrastructure on the client that is used by the browser platform, including to access the World-Wide Web.

As use of the World-Wide Web became more sophisticated, the Web Service Model emerged. It is described in the next section.

6.3 Web Services

6.3.1 Web Service Model

The essence of the web service model is that a web service exposes an API on the World-Wide Web, for use by software programs. This is illustrated in the figure (Figure 6).

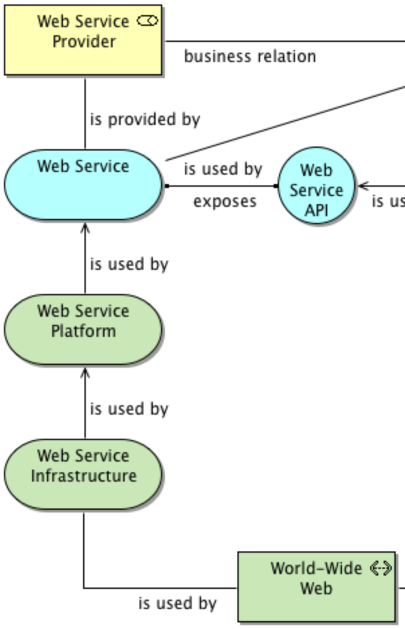


Figure 6: The Web Service Model

A *web service* is a software application or component that is web-accessible, which signifies that it provides a network-accessible service interface based on the Hypertext Transfer Protocol (HTTP).

A *web service API* is an interface by which another system interacts with a web service using the World-Wide Web. There are two styles of web service API that are comprehended by this Snapshot: SOAP and REST.

The *web service provider* is an enterprise or person that makes the web service available to the web service consumer. It is the *operator* of the web service. The provider often has a business relationship with the consumer, and may offer the service under commercial terms. When the service is offered free-of-charge, this may be under terms of a license that the consumer is deemed to accept.

The *web service consumer* is a person or enterprise that uses the service and in most cases has a contractual relationship with the provider. (This relationship may be implicit rather than being stated in a written contract.) Generally, each consumer is treated separately by the service, and different consumers may have different rights to use facilities and access information.

The *web service consumer program* is a program that interacts with the web service and is used by the consumer.

The *web service platform* supports the web service and enables it to use the underlying web service infrastructure.

The *web service infrastructure* provides the processing, storage, and other infrastructure capabilities that the web service needs, including access to the World-Wide Web.

The *web service consumer platform* supports the web service consumer program and enables it to use the underlying web service consumer infrastructure.

The *web service consumer infrastructure* provides the processing, storage, and other infrastructure capabilities that the web service consumer program needs, including access to the World-Wide Web.

6.3.2 Web Service Application Model

A frequently-encountered pattern is that of an application that exposes capabilities through an API as a web service. This is illustrated in the figure (Figure 7).

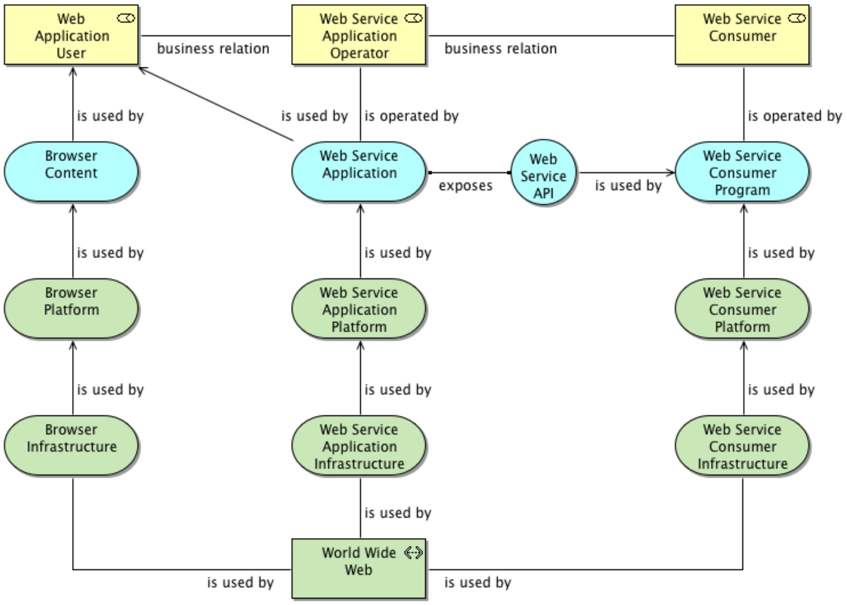


Figure 7: The Web Service Application Model

This model is a combination of the web application model and the web service model.

A *web service application* is both a web application and a web service. A *web service application operator* is an enterprise or person that manages it and makes it available for use; it is both a web application operator and a web service operator. The *web service application platform* is its supporting platform, and is both a web application platform and a web service platform. The *web service application infrastructure* is its underlying infrastructure, and is both a web application infrastructure and web service infrastructure.

The web application user, browser content, browser platform, and browser infrastructure are as in the web application model.

The web service consumer, web service consumer program, web service consumer platform, and web service consumer infrastructure are as in the web service model.

6.4 Cloud Computing

The essential concept of cloud computing is that IT resources are made available, within an environment that enables them to be used, via a communications network, as a service.

The authoritative definition by the US National Institute of Standards and Technology (NIST) says that:

"*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*" [NIST CLOUD DEFINITION]

This cloud model is composed of five essential characteristics, three service models, and four deployment models. The five essential characteristics are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. The three service models are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The four deployment models are private cloud, community cloud, public cloud, and hybrid cloud.

The Open Group Cloud Ecosystem Reference Model [CLOUD RM] defines the major actors in cloud computing and their relationships and a minimum set of architecture building blocks. It incorporates the essential NIST definitions, and adds a fourth service model: Business Process as a Service (BPaaS). The figure (Figure 8) shows the business roles and cloud services of the Cloud Ecosystem Reference Model without the architecture building blocks, to present cloud computing in a comparable way to other technologies in this Snapshot.

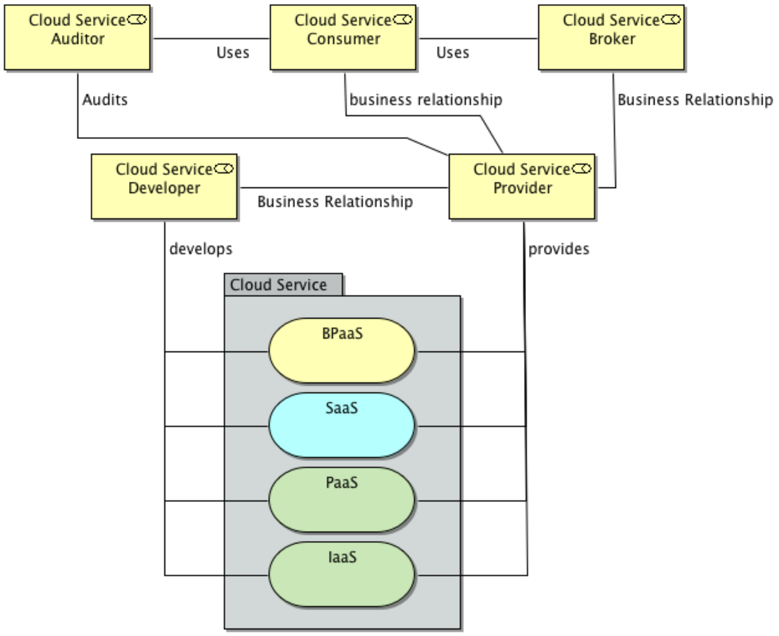


Figure 8: Cloud Business Roles and Services

The cloud service may be BPaaS, SaaS, PaaS, or IaaS. If it is SaaS, PaaS, or IaaS, its use follows a model that is a combination of the basic application and cloud computing models. These three models are described in the following sections. If it is BPaaS, it is realized by a business process that may use applications, and the basic application model or one of the same three combination models applies to each such use.

6.4.1 Cloud Software as a Service Model

In the SaaS service model, the cloud service is realized as a combination of application, platform, and infrastructure, as illustrated in the figure (Figure 9).

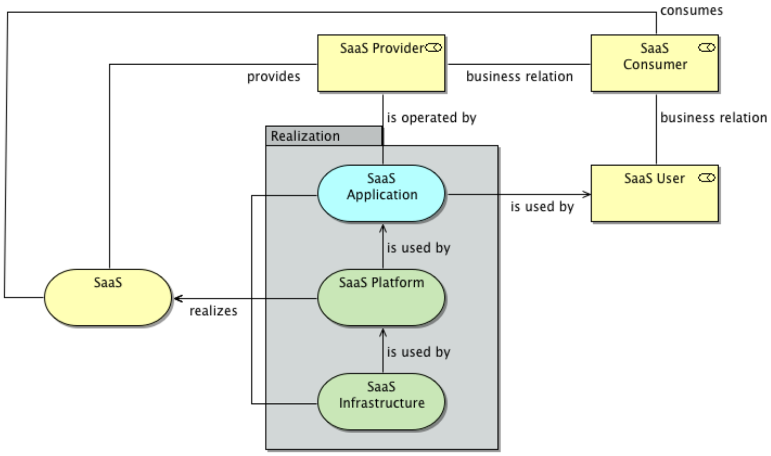


Figure 9: The Cloud Software as a Service Model

The *SaaS provider* and *SaaS consumer* are the cloud service provider and cloud service consumer of the SaaS cloud service.

The *SaaS provider*, *SaaS user*, *SaaS application*, *SaaS platform*, and *SaaS infrastructure* form a copy of the Application Model, with the SaaS provider as the operator of the SaaS application.

The SaaS consumer has a business relation with (often is the employer of) the SaaS users.

6.4.2 Cloud Platform as a Service Model

In the PaaS service model, the cloud service is realized as a combination of platform and infrastructure, as illustrated in the figure (Figure 10).

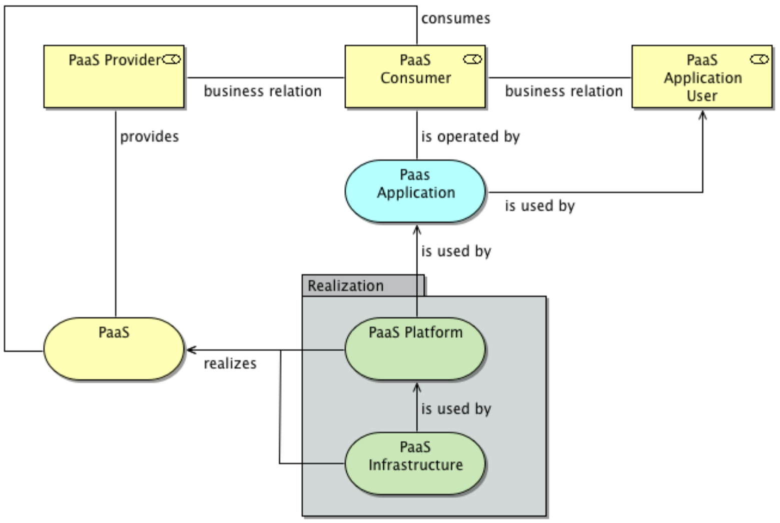


Figure 10: The Cloud Platform as a Service Model

The *PaaS provider* and *PaaS consumer* are the cloud service provider and cloud service consumer of the PaaS cloud service.

The *PaaS consumer*, *PaaS application user*, *PaaS application*, *PaaS platform*, and *PaaS infrastructure* form a copy of the Application Model, with the PaaS consumer as the operator of the PaaS application.

The PaaS consumer uses the *PaaS service* to support the PaaS application, and has a business relation with the PaaS application users.

6.4.3 Cloud Infrastructure as a Service Model

In the IaaS service model, the cloud service is realized as infrastructure, as illustrated in the figure (Figure 11).

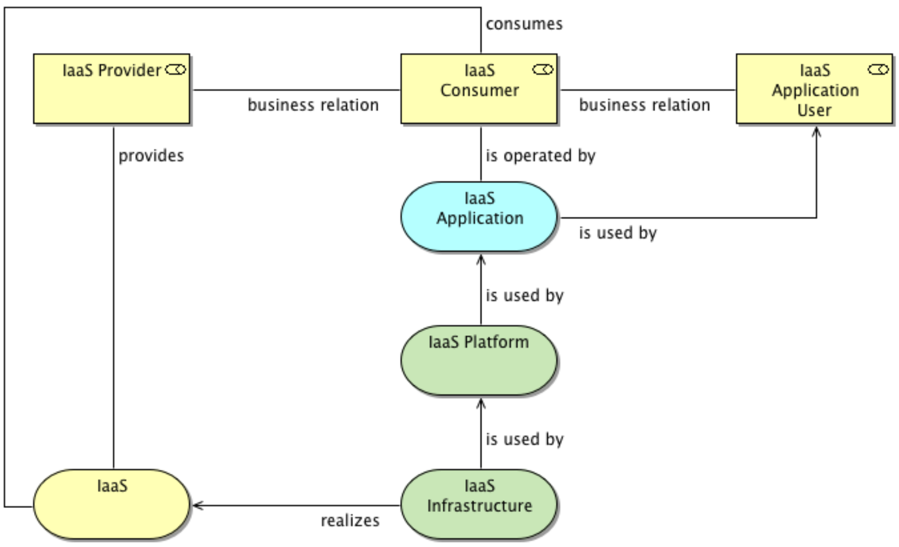


Figure 11: The Cloud Infrastructure as a Service Model

The *IaaS provider* and *IaaS consumer* are the cloud service provider and cloud service consumer of the IaaS cloud service.

The *IaaS consumer*, *IaaS application user*, *IaaS application*, *IaaS platform*, and *IaaS infrastructure* form a copy of the Application Model, with the IaaS consumer as the operator of the IaaS application.

The IaaS consumer uses the IaaS service to support the IaaS platform, which in turn supports the *IaaS application*, and has a business relation with the IaaS application users.

6.5 Mobile Computing

6.5.1 Mobile Computing Model

The mobile computing phenomenon is based on the use of portable computing devices that can connect to the Internet wherever the user is. This can be achieved in various ways, including:

- The use of cellular telephone networks as networks in the Internet
- The ability to connect a device to WiFi networks that are part of the Internet
- The ability to connect a portable computer to access points on different fixed networks that are part of the Internet

(The Internet is a collection of connected networks that can include cellular telephone networks, WiFi networks, fixed Ethernet networks, and other networks of various kinds.)

Examples of mobile devices include cellphones, tablets, and laptop computers.

The basic mobile computing model is illustrated in the figure (Figure 12). It is essentially the same as the Application Model with the addition of a connection to the World-Wide Web, and with the understanding that the user of the mobile device and apps is also their operator.

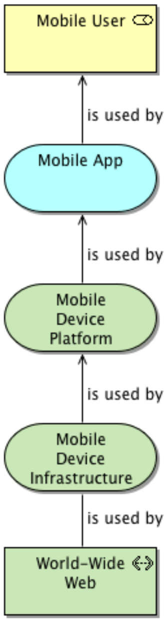


Figure 12: The Mobile Computing Model

The application programs that run on mobile devices are known as *apps*. "App" is not just an abbreviated form of "application". An app generally does not use a large amount of computing resource (since mobile devices often do not have large amounts of computing resource) and uses information that is stored on a server rather than holding that information in local storage long-term.

A *mobile device platform* supports the application programs on the device and enables them to use the capabilities of the device infrastructure. Examples are Android and iOS.

Mobile device infrastructure typically includes some means of obtaining user input, some form of user display, one or more processors, local storage, and connectivity to the Internet.

6.5.2 App Store Model

Apps are usually downloaded to mobile devices from app stores. An *app store* is a web resource from which users can acquire and download apps. There are a number of app stores on the web. The original examples were operated by mobile device platform vendors. There are now also app stores operated by other enterprises.

The app store model captures the business and technical environment for app store use, as shown in the figure (Figure 13).

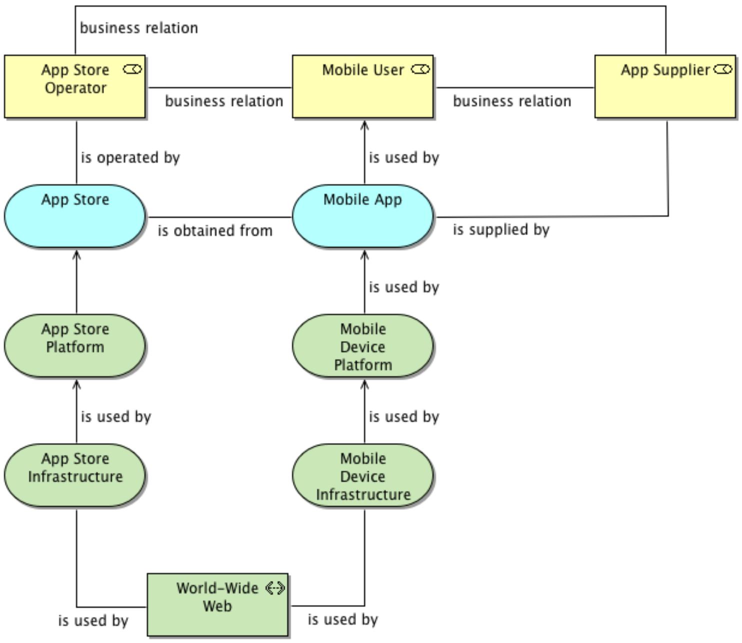


Figure 13: The App Store Model

The mobile user, mobile app, mobile device platform, and mobile device infrastructure are as in the Mobile Computing Model. This is an extension of that model.

The user of a mobile device can download an app from an app store. The user often has a business relation with the *app store operator*. For example, the operator may be a mobile device platform vendor, and the user may have purchased or licensed the platform. In any case, the operator typically identifies each user and keeps track of their downloads.

The *app supplier* is an enterprise or person that makes the app available for acquisition in the app store. The user often purchases or licenses the app from the supplier, rather than from the app store operator.

The app store operator generally has a business relation with the supplier. The operator may, for example, charge a commission on sales of the app through the store. The operator may also impose quality or security standards on apps made available in the store.

6.5.3 Mobile Device Management Model

Mobile Device Management (MDM) is management of mobile devices deployed across mobile operators, service providers, and enterprises. MDM functionality typically includes over-the-air distribution of applications, data, and configuration settings for all types of mobile devices, including mobile phones, smartphones, tablet computers, ruggedized mobile computers, mobile printers, mobile point-of-sale devices, etc. This applies to company-owned and employee-owned devices across the enterprise and to mobile devices owned by consumers.

The Mobile Device Management Model is shown in the figure (Figure 14).

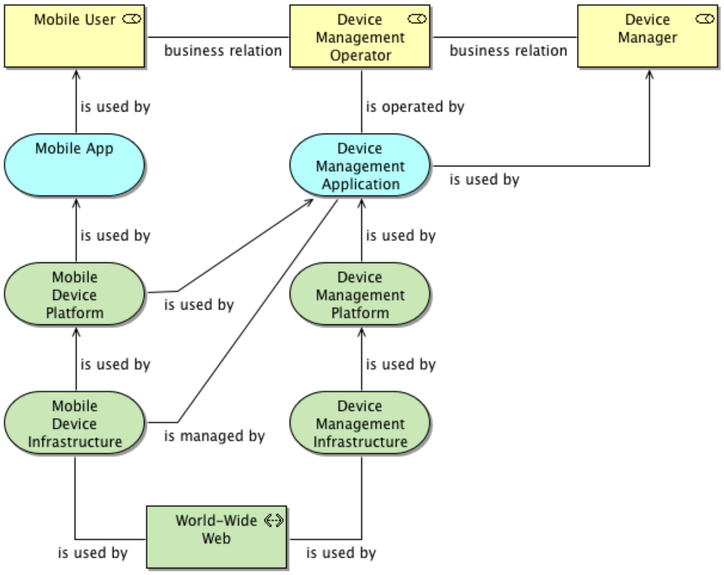


Figure 14: The Mobile Device Management Model

The mobile user, mobile app, mobile device platform, and mobile device infrastructure are as in the Mobile Computing Model. This is an extension of that model.

The remainder of the MDM model is a copy of the Application Model. The *device management operator*, *device manager*, *device management application*, *device management platform*, and *device management infrastructure* can be considered as the application operator, application user, application, application platform, and application infrastructure of the application model.

The device management operator is an enterprise that manages a set of devices. A common scenario is that these are devices used – and in some cases owned – by its employees. The device manager is also often an employee of that enterprise, with responsibility for managing the devices, perhaps a member of the IT department. In any case, there is some kind of business relation between the device management operator and the mobile user, and between the device management operator and the device manager.

The device manager uses the device management application to manage the devices. This application is operated by the device management operator. It is able to use the mobile device platforms on the managed devices to manage the infrastructure. It communicates with these platforms across the World-Wide Web, and has access to this through the device management platform and the underlying device management infrastructure.

6.5.4 Mobile Application Management Model

Mobile Application Management (MAM) is the management of apps on mobile devices deployed across mobile operators, service providers, and enterprises, including provisioning and configuration of access control.

MAM differs from MDM. It focuses on application management, while MDM manages the device at lower levels, including firmware and configuration settings.

The mobile application management model is shown in the figure (Figure 15).

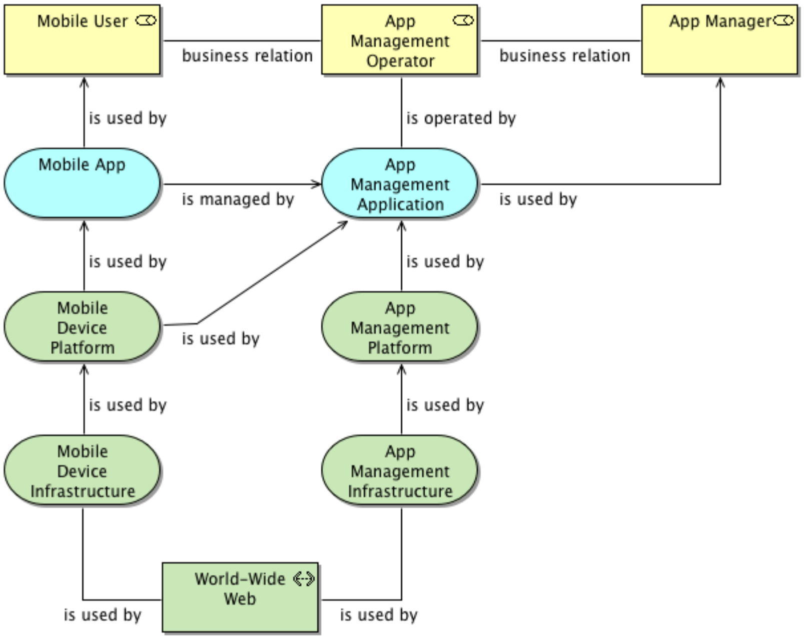


Figure 15: The Mobile Application Management Model

The mobile user, mobile app, mobile device platform, and mobile device infrastructure are as in the Mobile Computing Model. This is an extension of that model.

The remainder of the MAM model is a copy of the Application Model. The *app management operator*, *app manager*, *app management application*, *app management platform*, and *app management infrastructure* can be considered as the application operator, application user, application, application platform, and application infrastructure of the application model.

The app management operator is an enterprise that manages a set of mobile apps. A common scenario is that these are apps used by its employees. The app manager is also often an employee of that enterprise, with responsibility for managing the apps, perhaps a member of the IT department. In any case, there is some kind of business relation between the app management operator and the mobile user, and between the app management operator and the app manager.

The app manager uses the app management application to manage the devices. This application is operated by the app management operator. It is able to use the mobile device platforms on the managed devices to manage the apps that run on them. It communicates with these platforms across the World-Wide Web, and has access to this through the app management platform and the underlying app management infrastructure.

6.5.5 Mobile Connected Device Model

Mobile devices often have local connections to other devices. This is an important feature of mobile computing.

Locally connected devices can include:

- Other, similar, mobile devices
- Peripherals, such as headsets or printers
- Fixed devices connected to the Internet (for example, payment terminals)
- Sensors and controls

Local connections can use a number of media, including:

- USB cables
- Bluetooth
- Near Field Communication (NFC)

The mobile connected device model is shown in the figure (Figure 16).

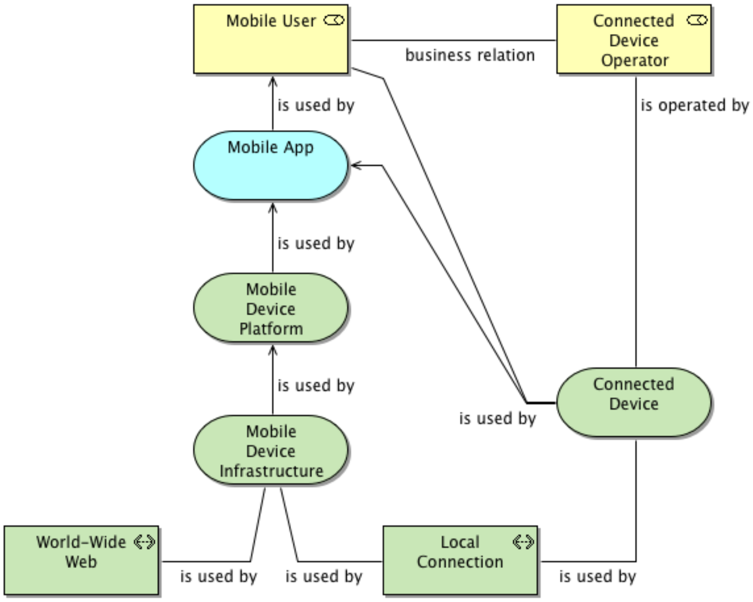


Figure 16: The Mobile Connected Device Model

The mobile user, mobile app, mobile device platform, and mobile device infrastructure are as in the Mobile Computing Model. This is an extension of that model.

A *connected device* is a device connected to a mobile computing system by some form of local connection. The *connected device operator* is the person or enterprise responsible for operating the connected device. This may be the mobile user. There is generally a business relation between the mobile user and the connected device operator, covering use of the connected device. (This is the trivial "same entity" business relation if the connected device operator is the mobile user.)

The *mobile device platform* and *mobile device infrastructure* enable apps and the user to use connected devices by communicating with them across the local connections.

6.6 Social Computing

There are no models defined specifically for social computing, but social computing solutions often include instances of the Web Application Model and the Web Service Model.

6.6.1 Social Networks

Social Computing refers here to computing related to or using social media. *Social media* is an application of Internet and web technology that provides a means of interactions among people in which they create, share, and exchange information and ideas in virtual communities and networks.

A social media system is a *social network*. Such a system is essentially a web application, as shown in the figure (Figure 17).

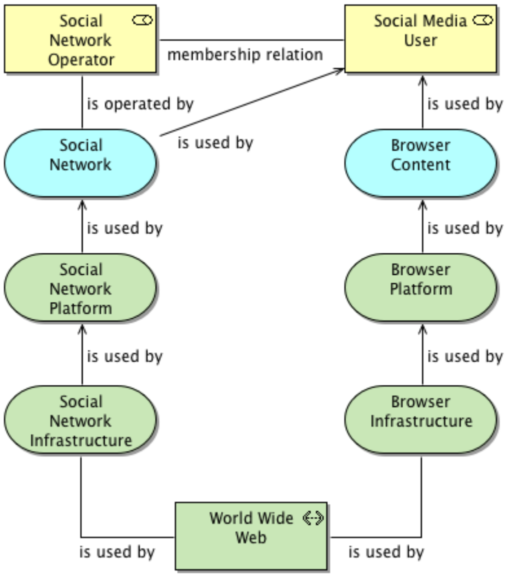


Figure 17: Social Network

The *social network* is a web application, the *social network operator* is the web application operator, the *social network platform* is the web application platform, and the *social network infrastructure* is the web application infrastructure, as in the Web Application Model. The *social media user* is the web application user. Users are treated individually; different users can access different information, and may have rights to access different facilities. Membership is often free-of-charge, but charges may apply, perhaps with different charges for different levels of membership.

6.6.2 Social Web Service

Social networks can often be accessed through APIs. Many social network operators expose APIs and have developed apps that use them.

These APIs are of two different kinds. One kind is used by the members of the social network, and enables them to use it more effectively than through a web browser. The other kind provides information about use of the social network, and is used by other enterprises, often for marketing purposes. The business model of many social networks is that their members use them free-of-charge, and other enterprises pay for information about that use.

A social network that exposes APIs is essentially a web service, as shown in the figure (Figure 18).

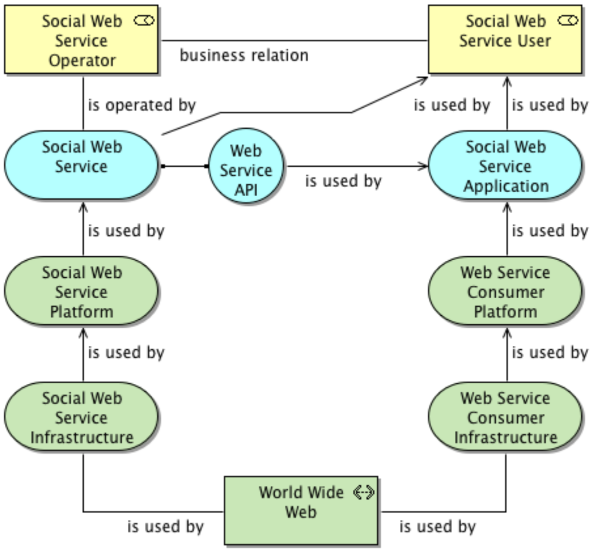


Figure 18: Social Web Service

The social network is the social web service, with operator, platform, and infrastructure as in the Web Service Model. The *social web service user* is a web service consumer, and the *social web service application* is the web service consumer system.

6.7 The Internet of Things

There is no commonly agreed definition of the *Internet of Things*. The term is used here to mean "the collection of uniquely identifiable objects embedded in or accessible by Internet hosts".

A "uniquely identifiable object" can be, for example:

- A sensor, such as a temperature sensor (thermometer)
- A control; for example, to control a valve in a heating system
- A combination of sensor and control (for example, a thermostat)

- An object identifier, such as an RFID tag or a barcode

It is estimated that there will be over 25 billion such devices by 2020, generating data at a rate that is one or more orders of magnitude higher than the rate at which people generate data today.

For the purposes of this Snapshot, the things of interest are those that are or can be connected to the Internet, either directly or indirectly. These alternatives are covered by two basic models: the Internet Object Model for direct connection, and the Managed Object Model for indirect connection.

6.7.1 Internet Object Model

The Internet object model is related to the Internet of Things and also to big data analytics. It captures the situation where a device is directly connected to the Internet. Some such devices can be sources of data for big data analytics.

In the Managed Object Model, discussed next, an identifiable object, which can be a sensor, a control, or a passive object (such as an item with a barcode) is connected by a local network to a system that manages it. This system may in turn be connected to the Internet, enabling other systems on the Internet to access the object indirectly. In the Internet connected device model, by contrast, the object is connected to the Internet directly. This rules out the possibility of it being a passive object, since it must be able to manage the Internet connection. It can, however, be a sensor or a control (or both). If the object is a sensor, then it can be a data source.

The Internet object model is shown in the figure (Figure 19).

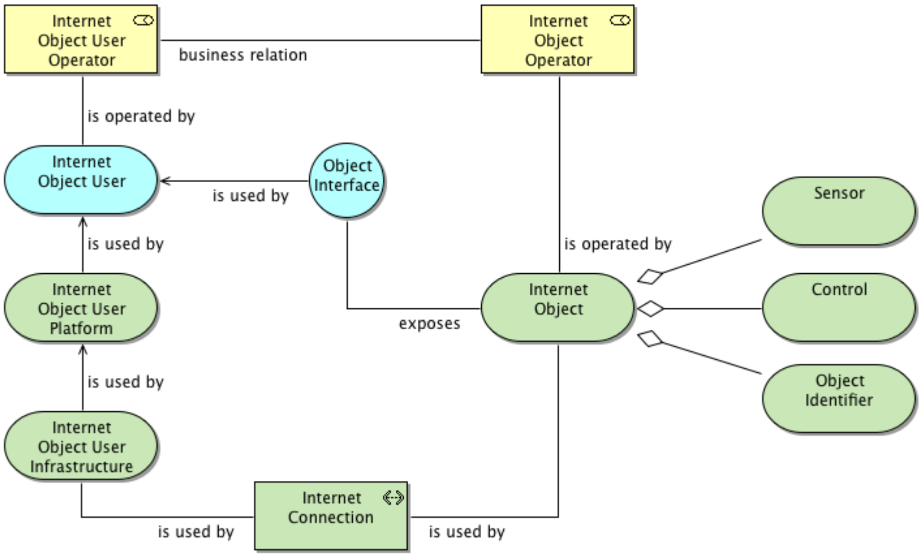


Figure 19: The Internet Object Model

An *Internet object* is a uniquely identifiable object that is connected to the Internet and exposes an interface – the *object interface* – that enables other systems connected to the Internet to interact with it. It is operated by a person or enterprise: the *Internet object operator*.

The object interface can be a web service API, using HTTP over the Internet protocols, or can be some other kind of interface that uses the Internet protocols, such as a set of defined IP messages.

The *Internet object user* is a program that interacts with the object. The person or enterprise that operates it is the *Internet object user operator*. There is generally some kind of business relationship between the Internet object user operator and the Internet object operator that covers the use of the object. (This includes the trivial business relationship where the two entities are the same.)

The *Internet object user* is supported by the *Internet object user platform* and *Internet object user infrastructure*. The Internet object user infrastructure provides access to the Internet for interaction with the object.

6.7.2 Managed Object Model

The more common scenario today is of an object that is connected indirectly, as in the managed object model, shown in the figure (Figure 20).

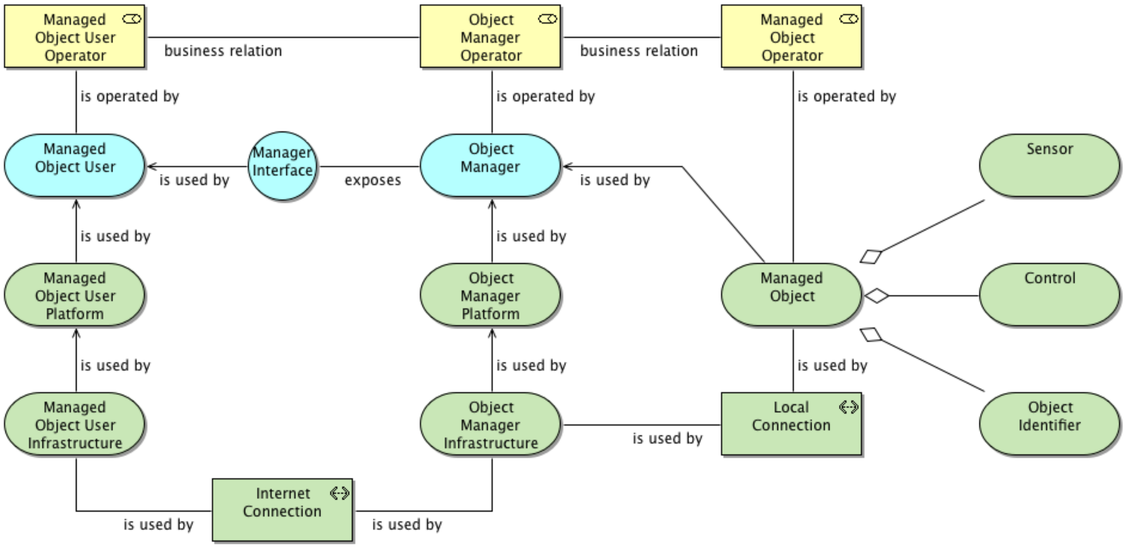


Figure 20: The Managed Object Model

A *managed object* is a uniquely identified object that is connected by a local connection to a system that manages it and is connected to the Internet.

The managed object is operated by a person or enterprise that is the *managed object operator*. The system that manages it is operated by a person or enterprise that is the *object manager operator*. These two entities are often the same. If they are not, then there is generally a business relation between them covering management of the object.

The local connection media can be, for example:

- A dedicated device inter-connection network such as a vehicle Controller Area Network (CAN)

- Radio, in particular for NFC and RFID
- Optical; for example, for barcodes

The connection can be permanent (as in a CAN) or transient (as with NFC).

The system that manages the object includes a program, the *object manager*, that reads from or gives instructions to the object, and in some cases may be able to adjust or configure it. This program is supported by the *object manager platform* which enables it to use the capabilities of the *object manager infrastructure*. These include connectivity to the local connection and to the Internet.

The *manager interface* is exposed by the object manager to enable remote systems to interact with it and with the objects that it manages. This interface can be a web service API, using HTTP over the Internet protocols, or can be some other kind of interface that uses the Internet protocols, such as a set of defined IP messages.

The *managed object user* is a program that interacts with the object manager and the objects that it manages. The person or enterprise that operates it is the *managed object user operator*. There is generally some kind of business relationship between the managed object user operator and the object manager operator that covers the use of the manager and managed objects. (This includes the trivial business relationship where the two entities are the same.)

The managed object user is supported by the *managed object user platform* and *managed object user infrastructure*. The managed object user infrastructure provides access to the Internet for interaction with the manager and managed objects.

If the manager interface is a web service API then the model is an extension of the Web Service Model, with the managed object user operator, managed object user, managed object user platform, managed object user infrastructure, object manager operator, object manager, object manager platform, and object manager infrastructure as, respectively, the web service provider, web service, web service platform, web service infrastructure, web service consumer, web service consumer program, web service consumer platform, and web service consumer infrastructure of that model.

The system that manages the object can be a mobile device, with the manager interface a web service API. In this case, this model is also an extension of the Mobile Connected Device Model, with the object manager operator, object manager, object manager platform, object manager infrastructure, managed object operator, and managed object as, respectively, the mobile user, mobile app, mobile device platform, mobile device infrastructure, connected device operator, and connected device of that model.

6.8 Big Data

The term *big data* refers to data that is so large that it is difficult to work with using IT systems available today. There is a growing body of analysis, visualization, and distributed processing software that enables people to extract useful information from such data.

A big data application obtains data from one or more sources and analyzes it.

Data sources can include:

- Corporate data in SQL databases
- Data in SQL or NoSQL databases that are cloud services
- Data provided by social networks
- Data provided by sensors or object identifiers in the Internet of Things

An item of data generally has an owner. The operator of the big data application has a business relationship with the owner that covers use of the data.

The application may include visualization functionality to enable effective presentation of the results of its analysis to users.

The application may be a web service that makes the results of its analysis available to other applications or apps through APIs.

Performance and quality of service of the application are crucial. Achieving them is a key challenge for big data analysis.

To be able to process large amounts of data, and to achieve performance and quality of service, the application may use the platform to marshal and configure underlying processing, storage, and network resources.

Objects in the Internet of Things can be generators of data; the Internet Object Model and the Managed Object Model are big data models as well as being Internet of Things models. There are two other basic models for big data: the Database Source Model, and the Data API Source Model.

6.8.1 Internet Object Model for Big Data

When an Internet object generates data and exposes it through its object interface, the Internet object user may process large volumes of data, and be a big data application. The Internet object operator is often the data owner.

A large volume of data may be generated by a single object, or a single Internet object user may interface to a large number of Internet objects that together generate a large aggregate volume of data.

6.8.2 Managed Object Model for Big Data

Similarly, when a managed object generates data and the object manager exposes it through its manager interface, the managed object user may process large volumes of data, and be a big data application. The managed object operator or the object manager operator is often the data owner.

A large volume of data may be generated by a single object, or by multiple objects managed by a single object manager, or a single managed object user may interface to a large number of object managers that together generate a large aggregate volume of data.

6.8.3 Database Source Model

The database source model applies when the big data application accesses data in a database via a programmatic interface, such as the ODBC interface. This model is shown in the figure (Figure 21).

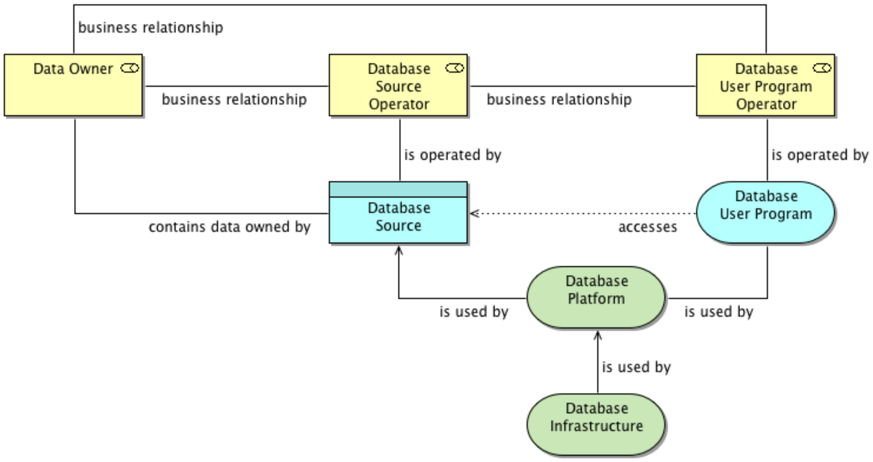


Figure 21: The Database Source Model

The *database user program* is a big data application. The *database source* is the database that holds the data, plus the data that it holds.

The *database source operator* is the person or enterprise that operates the database.

The *database user program operator* is the person or enterprise that operates the database user program. This program may have users, such as data scientists, that use the program to process and analyze the data. Alternatively (or in addition), the database user program may expose the data, possibly in processed form, to other programs, acting as a data API source in the Data API Source Model.

A *data owner* is a person or enterprise that owns some of the data. There are generally business relationships between the data owner, the database source operator, and the database user program operator, covering storage and use of the data.

The *database platform* supports the database user program and the database source, and provides the capabilities by which the database user program accesses the data. The underlying *database infrastructure* includes the media on which the data is stored.

The media on which the data is stored may be directly attached to the processor on which the database user program runs, or may be attached to another processor that is connected by a network. In this case, the database infrastructure includes both processors and the network, and the database platform manages the data transfer between the processors.

6.8.4 Data API Source Model

The data API source model applies when the big data application accesses data that is exposed by another application via an API. This model is shown in the figure (Figure 22).

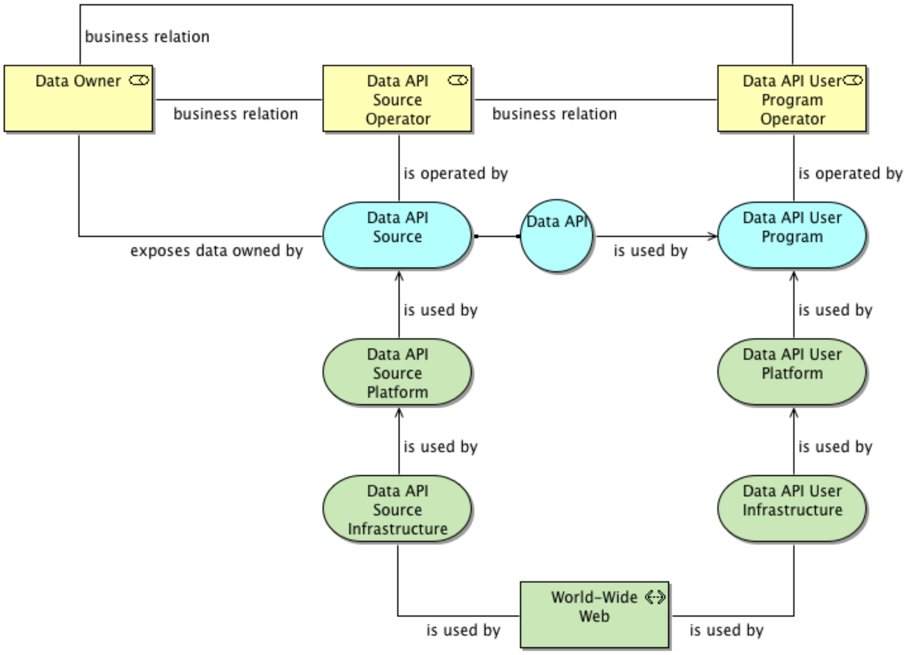


Figure 22: The Data API Source Model

The *data API user program* is a big data application. The *data API source* is an application program that exposes the data via an API, the *Data API*.

The data API source operator is the person or enterprise that operates the data API source.

The data API user program operator is the person or enterprise that operates the data API user program. This program may have users, such as data scientists, that use the program to process and analyze the data. Alternatively (or in addition), the data API user program may expose the data, possibly in processed form, to other programs, acting as a data API source.

A data owner is a person or enterprise that owns some of the data. There are generally business relationships between the data owner, the data API source operator, and the data API user program operator, covering storage and use of the data.

A common application of this model is one in which the data API source is a social web service, as in the Social Web Service use of the web service model. In this case, the data API source operator and the data owner are the social web service operator, the data API is the web service API, the data API user program operator is the social web service user, and the data API user program is the social web service application.

The *data API source platform* and *data API source infrastructure* support the data API source. The *data API user platform* and *data API user infrastructure* support the data API user program, and enable it to use the data API to access the data.

6.9 Composition

Composition is a technique that:

- Enables rapid assembly, configuration, and operation of functional and cost-effective solutions by IT-aware business users
- Enables development and integration of solutions by IT specialists

It can be used in providing the Multi-Party Services Orchestration capability.

6.9.1 The Composition Model

The composition model is shown in the figure (Figure 23).

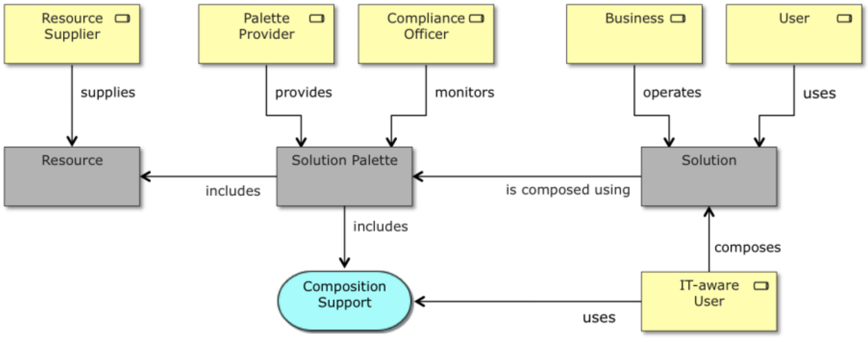


Figure 23: The Composition Model

A *solution* is composed by *IT-aware users*, using a *solution palette*. This palette is created by *IT specialists*. It contains *resources* that are composed to make the solution, and *composition-support* applications that facilitate the composition process. It may just be essentially a single solution that can be configured, or may be a complex system in which the business can assemble and configure a variety of components.

The users of the resulting solution may include the IT-aware users that created it and also other *users*, not necessarily IT-aware, and not necessarily employed by the business. (They may, for example, be its customers, or employees of its business partners.) The users rely on the quality of services performed by and information provided by the solution, and the business is responsible for achieving and maintaining the required levels of quality.

The organization that employs the IT specialists to create the solution palette (or a single IT specialist, acting as a one-person enterprise) is the *palette provider*. The business operates the resources. The palette provider manages their use, performance, and delivery, and negotiates relationships between the business and their suppliers. The business and the palette provider may be the business and IT departments of the same enterprise, or the palette provider may be a separate enterprise acting as resource *broker*. A “cloud broker” is a particular case of this.

The *compliance officer* is a person acting for the business to ensure that it conforms to regulation and meets its self-imposed standards of operation. Where the palette provider is a separate organization, the business retains the responsibility for conformance to regulation and policy. To meet this responsibility, the compliance officer monitors compliance parameters of the solution kit and of the solutions, including rights to access services and information.

Resources used in solutions can include cloud services (BPaaS, SaaS, PaaS, and IaaS), application components, platform components, and infrastructure components.

A Glossary

Actor

A person, organization, or system that has a role that initiates or interacts with activities; for example, a sales representative who travels to visit customers. [TOGAF]

Actors may be internal or external to an organization. In the automotive industry, an original equipment manufacturer would be considered an actor by an automotive dealership that interacts with its supply chain activities.

App

An application program that runs on mobile devices.

“App” is not just an abbreviated form of “application”. An app generally does not use a large amount of computing resource (since mobile devices often do not have large amounts of computing resource) and uses information that is stored on a server rather than holding that information in local storage long-term.

App Store

A web resource from which users can acquire and download apps.

There are a number of app stores on the web. The original examples were operated by mobile device platform vendors. There are now also app stores operated by other enterprises.

Application

A deployed and operational IT system that supports business functions and services; for example, a payroll. [TOGAF]

Applications use data and are supported by multiple technology components but are distinct from the technology components that support the application.

Application Platform

The collection of technology components of hardware and software that provide the services used to support applications. [TOGAF]

Application Programming Interface (API)

An interface that specifies how some software components should interact with each other. (Refer to http://en.wikipedia.org/wiki/API.)

Authentication

Verification of claimed identity (see ISO 7498-2:1989). [XDSF]

Authorization

The granting of rights, which includes the granting of access based on access rights (see ISO 7498-2:1989). [XDSF]

Big Data

A collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.

The challenges include capture, curation, storage, search, sharing, transfer, analysis, and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found. (Refer to http://en.wikipedia.org/wiki/Big_data.)

Business Service

A service that supports business capabilities through an explicitly defined interface and is explicitly governed by an organization. [TOGAF]

The defined interface is not necessarily technology-based.

Business User

A person using a resource for business purposes.

Capability

An ability that an organization, person, or system possesses. [TOGAF]

Capabilities are typically expressed in general and high-level terms and typically require a combination of organization, people, processes, and technology to achieve. For example, marketing, customer contact, or outbound telemarketing.

Cloud Computing

A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. [NIST CLOUD DEFINITION]

This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Compliance Officer

A person acting for a business enterprise to ensure that it conforms to regulation and meets its self-imposed standards of operation.

Consumer

A person or organization that maintains a business relationship with, and uses the service from, a provider; e.g., Cloud Consumer as defined in the NIST CCRA.

A service is a resource, and its consumer is the resource operator.

Continual Service Improvement (CSI)

An IT Infrastructure Library (ITIL) process that uses methods from quality management in order to learn from past successes and failures continually to improve the effectiveness and efficiency of services and processes.

Distributed Computing

The decomposition of a computing problem across a network to individual computers and nodes for processing.

Ecosystem

A network of participating entities, each of which plays one or more roles to achieve targeted objectives.

Ecosystem participants are not necessarily aware of all other entities in the ecosystem but will in general affect or be affected by them. An ecosystem is subject to the effects of both internal and external factors.

Enterprise

A collection of organizations that has a common set of goals. [TOGAF]

For example, an enterprise could be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership.

First Platform

The application platform provided by traditional computer operating systems as described in the TOGAF Technical Reference Model (TRM). [TOGAF]

HTTP

The Hypertext Transfer Protocol defined by the IETF in RFC 2616 or in compatible RFCs that supersede RFC 2616. [IETF RFCs]

HTTPS

HTTP layered over the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol.

Infrastructure as a Service (IaaS)

The cloud service model in which the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. [NIST CLOUD DEFINITION]

The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

Infrastructure Component

A component of the technology infrastructure that is not part of a platform but may be used by a platform.

Infrastructure Service

An externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment. [ARCHIMATE]

In this Snapshot, platform services are distinguished from infrastructure services.

Insight

A meaningful pattern of data.

Internet of Things

The collection of uniquely identifiable objects embedded in or accessible by Internet hosts.

The term is generally used to mean the connection of autonomous sensors and controls to the Internet, so that enterprises can monitor the activities and operation of people, machines, buildings, engineering structures, and the natural environment and, in some cases, control that operation.

Internet Protocol

The essential communications protocol of the Internet.

The version in common use currently is version 4 (IPv4). This is gradually being replaced by version 6 (IPv6). This is of critical importance for scalability requirements (especially with the growth of the Internet of Things).

Interoperability

1. The ability to share information and services.
2. The ability of two or more systems or components to exchange and use information.
3. The ability of systems to provide and receive services from other systems and to use the services so interchanged to enable them to operate effectively together. [TOGAF]

IP

The Internet Protocol defined by the IETF in RFC 791 or in compatible RFCs that supersede RFC 791. [IETF RFCs]

KPI

Key Performance Indicator.

Mobile Computing

Computing carried out using portable computing devices that can be connected to servers via the Internet.

Mobile computing can be based on numerous communications technologies (e.g., Wi-Fi, Mobile Telephony, etc.).

Mobile Device

Synonym for Personal Computing Device.

Near Field Communication (NFC)

A set of standards for smartphones and similar devices to establish radio communication with each other by touching them together or bringing them into proximity, usually no more than a few inches. (Refer to http://en.wikipedia.org/wiki/Near_Field_Communications.)

Network

A communication medium between two or more devices.

The systems connected to a network that input data from the network or output data to the network for conveyance to other systems are called nodes. A particular network, of fundamental importance, is the Internet. A special instance in the context of the Open Platform 3.0 Standard is a Social Network.

Open Standards

Standards made available to the general public that are developed (or approved) and maintained via a collaborative and consensus-driven process.

Open standards facilitate interoperability and data exchange among different products or services and are intended for widespread adoption. (Refer to http://en.wikipedia.org/wiki/Open_standard for ITU-T definition and others.)

PAP

Policy Administration Point. [XACML]

PDP

Policy Decision Point. [XACML]

PEP

Policy Enforcement Point. [XACML]

Personal Computing Device

One of any of a large class of mobile technology devices including (but not limited to) mobile phones, smartphones, tablet computers, ruggedized mobile computers, mobile printers, mobile point-of-sale devices, etc.

PIP

Policy Information Point. [XACML]

Platform

A combination of technology infrastructure products and components that provides the prerequisites to host application software. [TOGAF]

Platform as a Service (PaaS)

The cloud service model in which the capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. [NIST CLOUD DEFINITION]

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

Platform Component

A technology component that is part of a platform.

Platform Service

A technical capability required to provide enabling infrastructure that supports the delivery of applications. [TOGAF]

A service performed by a platform.

Product Lifecycle Management (PLM)

An activity whose main goal is the management of all the business processes and associated data.

Data is generated by events and actions of various lifecycle actors (both human and software systems) and it is distributed along the product's lifecycle phases such as Beginning Of Life (BOL), Middle Of Life (MOL), and End Of Life (EOL).

Provider

A person or organization that is the entity responsible for making a service available to interested parties; e.g., Cloud Provider as defined in the NIST CCRA.

A provider acquires and manages the infrastructure required for providing the services, runs the capabilities that provides the services, and makes arrangement to deliver the services to the consumers through appropriate access. A service is a resource, and its provider is the resource supplier.

Quantum Lifecycle Management (QLM)

A major leap beyond PLM, in order to make possible the management of products on the Internet of Things.

Common, open, and trustworthy information exchange standards for QLM will enable the closing of information loops, allowing information to be shared across the whole spectrum of lifecycle including products, human, food and beverage, pharmaceutical, healthcare, supply chain and logistics, and data governance, among many.

Radio Frequency Identification (RFID)

The wireless non-contact use of radio-frequency electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. (Refer to http://en.wikipedia.org/wiki/Radio-frequency_identification.)

Role

The responsibility for performing specific behavior, to which an actor can be assigned. [ARCHIMATE]

Second Platform

The combination of the web service platform, the consumer platform, and the web browser.

Sensor

A device that sends to IT systems information obtained from its environment.

Social Computing

Computing related to or using social media.

Social Media

An application of Internet and web technology that provides a means of interactions among people in which they create, share, and exchange information and ideas in virtual communities and networks. (Refer to http://en.wikipedia.org/wiki/Social_Media.)

Social Network

A structure made up of a set of social actors (such as individuals or organizations) and a set of the dyadic ties between these actors.

In the context of the Open Platform 3.0 Standard it is those actors utilizing a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of actor-generated content. A social network depends on mobile and web-based technologies to create highly interactive platforms through which individuals and communities share, co-create, discuss, and modify user-generated content.

Software as a Service (SaaS)

The cloud service model in which the capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. [NIST CLOUD DEFINITION]

The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

SSL

The Secure Sockets Layer protocol defined by the IETF in RFC 6101 or in compatible RFCs that supersede RFC 6101. [IETF RFCs]

Supplier

An enterprise or person that supplies a resource to an operator.

The supplier of a resource usually has a business relation with its operator. For example, the operator may purchase or lease the resource from the supplier.

Systems Manager

A person responsible for the configuration and operation of IT systems.

Systems managers include service managers and data stewards.

TCP/IP

The Internet Transmission Control Protocol and the Internet Protocol defined by the IETF in RFC 793 and RFC 791, respectively. [IETF RFCs]

TCP

The Transmission Control Protocol defined by the IETF in RFC 793 or in compatible RFCs that supersede RFC 793. [IETF RFCs]

TLS

The Transport Layer Security protocol defined by the IETF in RFC 5246, as modified by RFC 6176, or in compatible RFCs that supersede RFC 5246 and RFC 6176. [IETF RFCs]

TRM

Technical Reference Model as defined in the TOGAF Standard. [TOGAF]

User

A person using a resource.

A user may be a business user acting on behalf of, and possibly an employee of, the operator of the resource. Business users include: business managers and users wanting to design processes, data scientists, and business technologists. A user may be acting on behalf of, and possibly an employee of, a business partner of the operator of the resource. A user may be a member of the general public who is a customer of the operator of the resource. A user may be a member of the general public using the resource but not as a customer of its operator. A user may use non-functional interfaces to configure or provision the application or application service, as well as using its functional interface.

Web Browser

A software application for retrieving, presenting, and traversing information resources on the World-Wide Web. (Refer to http://en.wikipedia.org/wiki/Web_Browser.)

In this Snapshot, a web browser is considered not as an application, but as part of the platform.

Web Service

A software application or component that is web-accessible, which signifies that it provides a network-accessible service interface based on the Hypertext Transfer Protocol (HTTP).

Web Service API

An interface by which another system interacts with a web service using the World-Wide Web.

Web Service Description Language (WSDL)

The web service description language defined by the World-Wide Web Consortium.

World-Wide Web

A system of interlinked hypertext documents accessed via the Internet. (Refer to http://en.wikipedia.org/wiki/World-Wide_Web.)

There is a single World-Wide Web, which is a common global resource.

World-Wide Web Consortium (W3C)

The main international standards organization for the World-Wide Web. (Refer to http://en.wikipedia.org/wiki/W3C.)

XACML

The eXtensible Access Control Mark-Up Language defined by OASIS.

B Example of a Wider Business Ecosystem

Use-case 15 in the Nexus of Forces in Action White Paper [Nexus] provides a good example of a wider business ecosystem. It concerns smart charging of electric vehicles. The participants include:

- Bulk energy generators
- (National) grid operators (transmission operators)
- Local supply network operator (distribution service operator or DSO)
- Energy suppliers (i.e., the parties who have a contractual relationship with the energy consumer – these may also be bulk generators)
- Consumers (vehicle users)
- Local government

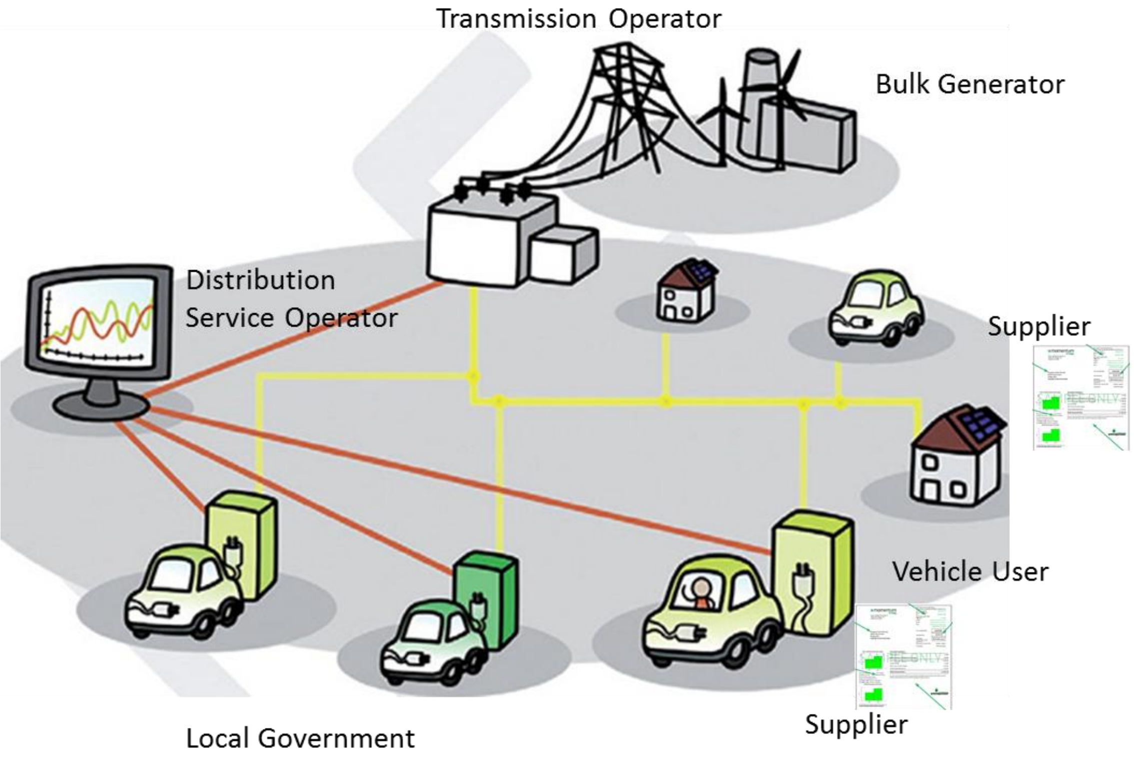


Figure 24: Electric Vehicle Smart Charging Ecosystem

A Vehicle User can charge a vehicle at a public charging station or via a private power supply (home or office). This use-case is only concerned with the former situation but could be extended to cover both scenarios.

A local controller (a device – part of the Internet of Things) controls one or more charging stations. The Energy Supplier informs the vehicle (and possibly the Vehicle User) via the local controller how much capacity is available to it. If the battery is nearly full the vehicle can inform the local controller that it needs less capacity and this capacity can then be made available to other vehicles at other charging stations. Standard protocols are defined, which can be used to support end-to-end communication.

The Charging Operator determines the capacity to be made available on the basis of information provided by the Distribution Service Operator (DSO) (maximum allowable capacity at that time), possibly combined with commercial information (e.g., current spot prices, predicted trends, flexibility agreements with vehicle-owners/customers where applicable). The DSO has predicted available capacity on the basis of currently predicted weather conditions and long-term usage patterns in the relevant area. The DSO is able to adapt to unexpected changes in real time and restrict or increase the locally available capacity.

We can regard the purpose of this ecosystem as providing a sustainable infrastructure for the use and support of electric vehicles. If we speak of the health of the system as a whole, this is how we will measure it. Each participant has to a greater or lesser extent an interest in the health of the system. Nonetheless the individual participants have varying measures of value and varying degrees of commitment to or dependence on the system as a whole.

C Event Model

This appendix describes a possible event model.

Human and system users require that event handling is transparent, assuring service solutions maintain their expected assurance, security, and privacy levels. A failure of a Service-Level Agreement (SLA) or an Operational-Level Agreement (OLA) creates an event, or policy non-compliance. Events are correlated to provide human or system operators with the ability to discern a root cause to failure. In general, there are four enterprise vectors for event correlations that manage enterprise properties and/or facilities:

- Security vectors (one or more Security and Information Event Managers (SIEM)
- One or more IT operations managers (ITOM)
- One or more Data Center Information Managers (DCIM)
- One or more Integrated Work Management Systems (IWMS)

These four systems may contain details for any event that might occur over an enterprise in order to support SLA/OLA compliance of services sponsored by Open Platform 3.0. The intersection or union of these event vectors may contain SLA/OLA event details for the assets, systems, and/or things that caused a service disruption.

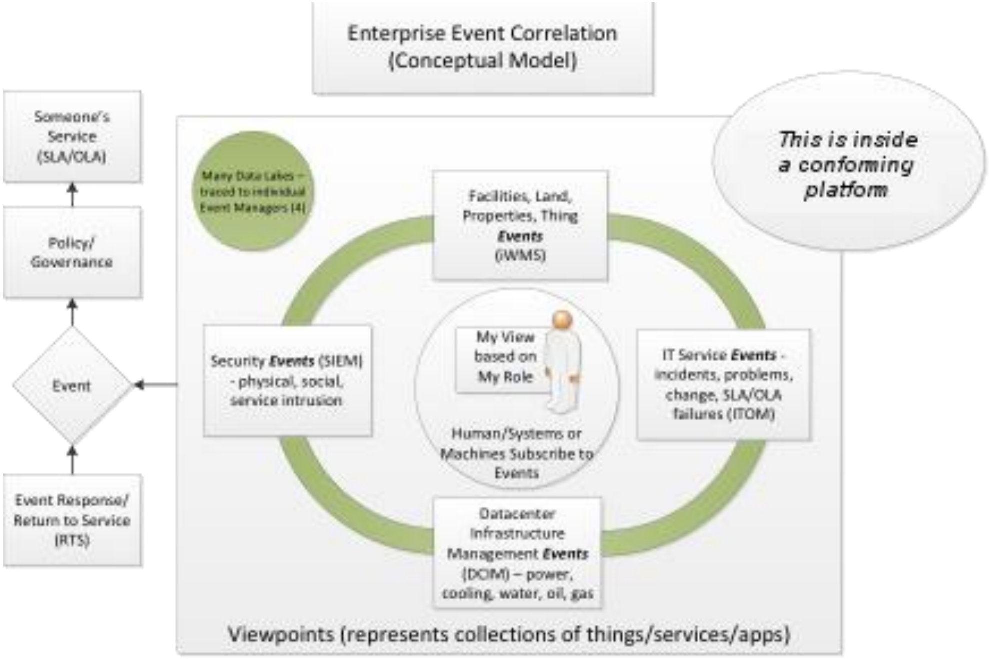

Enterprise Event Correlation (Conceptual Model)
Someone's Service (SLA/OLA)
Policy/ Governance
Event
Event Response/ Return to Service (RTS)
Many Data Lakes – traced to individual Event Managers (4)
Facilities, Land, Properties, Thing Events (IWMS)
This is inside a conforming platform
Security Events (SIEM) - physical, social, service intrusion
My View based on My Role
Human/Systems or Machines Subscribe to Events
IT Service Events - incidents, problems, change, SLA/OLA failures (ITOM)
Datacenter Infrastructure Management Events (DCIM) – power, cooling, water, oil, gas
Viewpoints (represents collections of things/services/apps)

Figure 25: Event Model

Transparent event flow should support the union of cloud computing, mobile computing, social computing, big data, and IoT events that support a singular user or system service. Enterprise event-correlation is critical to assure SLA/OLA compliance – demand from a user, system, or machine.