

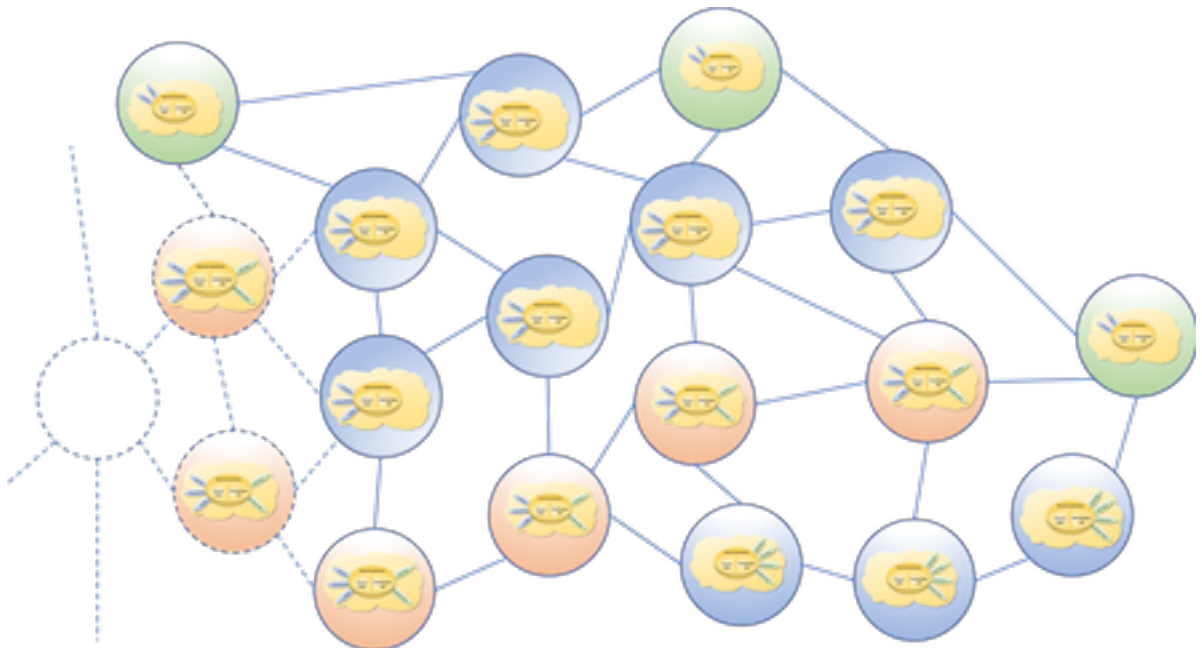
DISTRIBUTED IMMUTABLE DATA OBJECT

(DIDO) **Data Model (DM)**

for
Cryptocurrencies, Blockchains, Distributed Ledgers and Tangles

Version 1.0

June 2020



R. W. "Nick" Stavros
Ian T. Stavros
Bryan Turek
Char Wales
Jackrabbit Consulting

Distributed Immutable Data Object Data Model

Dido-DM

APPROVAL

The original of this document is approved and signed by:
Name: R. W. "Nick" Stavros, Ph.D.
Title: President/CEO Jackrabbit Consulting, Inc.
Date: 31 May 2020



Signature:

REVISION HISTORY

Revision	Date	Description	Author
1.0	1 June 2020	Technical Document for DIDO Project	Jackrabbit Consulting

Distributed Immutable Data Object Data Model

Dido-DM

TABLE OF CONTENTS

Introduction	1
Purpose	1
Scope	1
Overview	1
Package DIDO - Conceptual Model	2
Package DIDO - Conceptual Communities	2
Package DIDO - Conceptual Datatypes Package	3
Enumeration Copyright Grant Type	3
Enumeration Copyright Permission Type	4
Enumeration Copyright Private Use Type	4
Enumeration DIDO Key Type	4
Enumeration Email Address Type	5
Enumeration Email Type	5
Enumeration Multiplicity Type	5
Enumeration Parameter Definition Type	6
Enumeration Parameter Diretion Type	6
Enumeration Release Status Type	6
Enumeration Social Network Type	7
Enumeration Spoken Language Type	7
Enumeration Verdict Type	7
Package DIDO - Conceptual Domain Package	8
Class Allowed Domain Copyright	8
Class Domain	8
Class Domain Board Member	9
Class Domain Chair	9
Class Domain FAQ	10
Class Domain _Glossary	10
Class Domain Governing Board	10
Class Domain Keychain	11
Class Domain Kind	11
Class Domain Member	11
Class Domain Tag	12
Class Smart Contract	12
Package DIDO - Conceptual Ecosphere Package	12
Class Allowed Ecosphere Copyright	13
Class Ecosphere	13
Class Ecosphere Board Member	14
Class Ecosphere Chair	14
Class Ecosphere FAQ	15
Class Ecosphere Glossary	15
Class Ecosphere Governing Board	15
Class Ecosphere Keychain	16
Class Ecosphere Member	16
Class Ecosphere Sponsor	16
Class Ecosphere Tag	17
Class Ecosphere Version Control	17
Class Allowed Ecosys Copyright	18
Class Ecosys Board Member	18
Class Ecosys Chair	18
Class Ecosys FAQ	19
Class Ecosys Glossary	19
Class Ecosys Governing Board	19
Class Ecosys Keychain	20
Class Ecosys Member	20
Class Ecosys Tag	20
Class Ecosystem	21
Package DIDO - Conceptual Resource Package	21

Distributed Immutable Data Object Data Model

Dido-DM

TABLE OF CONTENTS

Class Copyright_Notice	22
Class External Reference	22
Class FAQ Answer	23
Class FAQ Question	23
Class Image Content	23
Class Key_Store	24
Package DIDO - Conceptual Runtime_Environment_Package	24
Class Application Container	25
Class Application_Container_Set	26
Class Cloud Provider Supported_OS	26
Class Cloud_Provider	26
Class Composite Application	27
Class CPU Architecture	27
Class DIDO Platform	28
Class Disk Image Repository	28
Class Node Platform	29
Class Node Profile	29
Class Node_Classification	30
Class Node_Set	30
Class OS Type	31
Class OS_Platform	31
Class Runtime Environment	32
Class Sign On Provider	32
Package DIDO - Conceptual Taxonomy Package	33
Class Glossary Entry	33
Class Tag	33
Package DIDO - Conceptual Testing Package	34
Class Expected Results	34
Class Parameter Definition	35
Class Parameter List	35
Class Test	35
Class Test Case	36
Class Test Executable	36
Class Test Plan	37
Class Test Result	37
Class Test Run	38
Class Test Set	38
Class Test Step	39
Class Test Tag	39
Package DIDO - Conceptual User_Details_Package	40
Class Email	41
Class Phone Number	41
Class Role	41
Class Social Network Connection	42
Class Street Address	42
Class User Details	42
Class User Profile	43
Class User_Role	43
Class Virtual Contact Card	44
Appendix A: Class Context Diagrams	45
Allowed Domain Copyright	45
Allowed Ecosphere Copyright	46
Allowed_Ecosys Copyright	47
Application Container	48
Application Container Set	49
Cloud Provider Supported_OS	50
Cloud_Provider	51
Composite Application	52

Distributed Immutable Data Object Data Model

Dido-DM

TABLE OF CONTENTS

Copyright Notice	53
CPU Architecture	54
DIDO Platform	55
Disk Image Repository	56
Domain Board Member	59
Domain Chair	60
Domain FAQ	61
Domain _Glossary	62
Domain Governing Board	63
Domain Keychain	64
Domain Kind	65
Domain Member	66
Domain Tag	67
Ecosphere	68
Ecosphere Board Member	70
Ecosphere Chair	71
Ecosphere FAQ	72
Ecosphere Glossary	73
Ecosphere Governing Board	74
Ecosphere Keychain	75
Ecosphere Member	76
Ecosphere Package	76
Ecosphere Sponsor	78
Ecosphere Tag	79
Ecosphere Version Control	80
Ecosys Board Member	81
Ecosys Chair	82
Ecosys FAQ	83
Ecosys Glossary	84
Ecosys Governing Board	85
Ecosys Keychain	86
Ecosys Member	87
Ecosys Tag	88
Ecosystem	89
Email	91
Enumeration Types	91
Expected Results	93
External Reference	94
FAQ_Answers	95
FAQ_QUESTION	96
Glossary Entry	97
Image Content	98
Key Store	99
Node Platform	100
Node Profile	101
Node_Classification	102
Node_Set	103
OS Type	104
OS_Platform	105
Parameter Definition	106
Parameter List	107
Phone Number	108
Primitive Types	109
Role	109
Runtime Environment	110
Signon Provider	112
Smart Contract	112
Social Network Connection	113
Street Address	114
Tag	115
Test	115
Test Plan Package	116
Test_Case	117
Test_Executable	118

Distributed Immutable Data Object Data Model
Dido-DM

TABLE OF CONTENTS

Test_Plan	119
Test_Result	120
Test_Run	121
Test_Set	122
Test_Step	123
Test_Tag	124
User Detail	125
User Profiles	127
User Roles	127
Virtual Contact Card	127

Distributed Immutable Data Object Data Model

Dido-DM

TABLE OF FIGURES

Figure 1.	Communities.....	3
Figure 2.	Datatypes Package	Error! Bookmark not defined.
Figure 3.	Testlt.....	Error! Bookmark not defined.
Figure 4.	Copyright Grant Type	3
Figure 5.	Copyright Permission Type	4
Figure 6.	Copyright Private Use Type.....	4
Figure 7.	DIDO Key Type	4
Figure 8.	Email Address Type	5
Figure 9.	Email Type.....	5
Figure 10.	Multiplicity Type.....	5
Figure 11.	Parameter Definition Type.....	6
Figure 12.	Parameter Diretion Type	6
Figure 13.	Release Status Type	6
Figure 14.	Social Network Type	7
Figure 15.	Spoken Language Type	7
Figure 16.	Verdict Type	8
Figure 17.	Domain Package.....	Error! Bookmark not defined.
Figure 18.	Allowed Domain Copyright.....	8
Figure 19.	Domain	9
Figure 20.	Domain Board Member	9
Figure 21.	Domain Chair	9
Figure 22.	Domain FAQ.....	10
Figure 23.	Domain _Glossary.....	10
Figure 24.	Domain Governing Board.....	11
Figure 25.	Domain Keychain	11
Figure 26.	Domain Kind.....	11
Figure 27.	Domain Member.....	12
Figure 28.	Domain Tag.....	12
Figure 29.	Smart Contract	12
Figure 30.	Ecosphere Package	13
Figure 31.	Allowed Ecosphere Copyright	13
Figure 32.	Ecosphere	14
Figure 33.	Ecosphere Board Member	14
Figure 34.	Ecosphere Chair.....	15
Figure 35.	Ecosphere FAQ.....	15
Figure 36.	Ecosphere Glossary	15
Figure 37.	Ecosphere Governing Board.....	16
Figure 38.	Ecosphere Keychain	16
Figure 39.	Ecosphere Member	16
Figure 40.	Ecosphere Sponsor.....	17
Figure 41.	Ecosphere Tag.....	17
Figure 42.	Ecosphere Version Control	17
Figure 43.	Ecosystem Package.....	Error! Bookmark not defined.
Figure 44.	Allowed Ecosys Copyright.....	18
Figure 45.	Ecosys Board Member	18
Figure 46.	Ecosys Chair	18
Figure 47.	Ecosys FAQ	19
Figure 48.	Ecosys Glossary.....	19
Figure 49.	Ecosys Governing Board.....	20
Figure 50.	Ecosys Keychain	20
Figure 51.	Ecosys Member.....	20
Figure 52.	Ecosys Tag.....	20
Figure 53.	Ecosystem.....	21
Figure 54.	Resource Package	22
Figure 55.	Copyright_Notice.....	22
Figure 56.	External Reference.....	22
Figure 57.	FAQ Answer	23

Distributed Immutable Data Object Data Model

Dido-DM

TABLE OF FIGURES

Figure 58.	FAQ Question.....	23
Figure 59.	Image Content.....	23
Figure 60.	Key_Store.....	24
Figure 61.	Runtime_Environment_Package.....	25
Figure 62.	Application Container.....	26
Figure 63.	Application_Container_Set.....	26
Figure 64.	Cloud Provider Supported_OS.....	26
Figure 65.	Cloud_Provider.....	27
Figure 66.	Composite Application.....	27
Figure 67.	CPU Architecture.....	28
Figure 68.	DIDO Platform.....	28
Figure 69.	Disk Image Repository.....	29
Figure 70.	Node Platform.....	29
Figure 71.	Node Profile.....	30
Figure 72.	Node_Classification.....	30
Figure 73.	Node_Set.....	31
Figure 74.	OS Type.....	31
Figure 75.	OS_Platform.....	32
Figure 76.	Runtime Environment.....	32
Figure 77.	Sign On Provider.....	32
Figure 78.	Taxonomy Package.....	33
Figure 79.	Glossary Entry.....	33
Figure 80.	Tag.....	33
Figure 81.	Testing Package.....	34
Figure 82.	Expected Results.....	35
Figure 83.	Parameter Definition.....	35
Figure 84.	Parameter List.....	35
Figure 85.	Test.....	36
Figure 86.	Test Case.....	36
Figure 87.	Test Executable.....	37
Figure 88.	Test Plan.....	37
Figure 89.	Test Result.....	38
Figure 90.	Test Run.....	38
Figure 91.	Test Set.....	39
Figure 92.	Test Step.....	39
Figure 93.	Test Tag.....	40
Figure 94.	User_Details_Package.....	40
Figure 95.	Email.....	41
Figure 96.	Phone Number.....	41
Figure 97.	Role.....	42
Figure 98.	Social Network Connection.....	42
Figure 99.	Street Address.....	42
Figure 100.	User Details.....	43
Figure 101.	User Profile.....	43
Figure 102.	User_Role.....	43
Figure 103.	Virtual Contact Card.....	44
Figure 104.	Allowed Domain Copyright.....	45
Figure 105.	Allowed Ecosphere Copyright.....	46
Figure 106.	Allowed_Ecosys Copyright.....	47
Figure 107.	Application Container.....	48
Figure 108.	Application Container Set.....	49
Figure 109.	Cloud Provider Supported_OS.....	50
Figure 110.	Cloud_Provider.....	51
Figure 111.	Composite Application.....	52
Figure 112.	Copyright Notice.....	53
Figure 113.	CPU Architecture.....	54
Figure 114.	DIDO Platform.....	55

Distributed Immutable Data Object Data Model

Dido-DM

TABLE OF FIGURES

Figure 115.	Disk Image Repository.....	56
Figure 116.	Domain	57
Figure 117.	Domain Board Member.....	59
Figure 118.	Domain Chair.....	60
Figure 119.	Domain FAQ.....	61
Figure 120.	Domain_Glossary	62
Figure 121.	Domain Governing Board	63
Figure 122.	Domain Keychain.....	64
Figure 123.	Domain Kind	65
Figure 124.	Domain Member	66
Figure 125.	Domain Tag	67
Figure 126.	Ecosphere.....	68
Figure 127.	Ecosphere Board Member.....	70
Figure 128.	Ecosphere Chair	71
Figure 129.	Ecosphere FAQ	72
Figure 130.	Ecosphere Glossary	73
Figure 131.	Ecosphere Governing Board	74
Figure 132.	Ecosphere Keychain	75
Figure 133.	Ecosphere Member	76
Figure 134.	Ecosphere Package.....	77
Figure 135.	Ecosphere Sponsor	78
Figure 136.	Ecosphere Tag	79
Figure 137.	Ecosphere Version Control.....	80
Figure 138.	Ecosys Board Member	81
Figure 139.	Ecosys Chair.....	82
Figure 140.	Ecosys FAQ.....	83
Figure 141.	Ecosys Glossary	84
Figure 142.	Ecosys Governing Board	85
Figure 143.	Ecosys Keychain	86
Figure 144.	Ecosys Member	87
Figure 145.	Ecosys Tag	88
Figure 146.	Ecosystem	89
Figure 147.	Email.....	91
Figure 148.	Enumeration Types.....	92
Figure 149.	Expected Results.....	93
Figure 150.	External Reference	94
Figure 151.	FAQ_Answers.....	95
Figure 152.	FAQ_QUESTION.....	96
Figure 153.	Glossary Entry	97
Figure 154.	Image Content	98
Figure 155.	Key Store	99
Figure 156.	Node Platform.....	100
Figure 157.	Node Profile	101
Figure 158.	Node_Classification	102
Figure 159.	Node_Set.....	103
Figure 160.	OS Type.....	104
Figure 161.	OS_Platform	105
Figure 162.	Parameter Definition	106
Figure 163.	Parameter List	107
Figure 164.	Phone Number.....	108
Figure 165.	Primitive Types	109
Figure 166.	Role	110
Figure 167.	Runtime Environment	111
Figure 168.	Signon Provider	112
Figure 169.	Smart Contract.....	113
Figure 170.	Social Network Connection.....	114
Figure 171.	Street Address	114

Distributed Immutable Data Object Data Model
Dido-DM

TABLE OF FIGURES

Figure 172.	Tag.....	115
Figure 173.	Test.....	116
Figure 174.	Test Plan Package.....	117
Figure 175.	Test_Case	118
Figure 176.	Test_Executable	119
Figure 177.	Test_Plan.....	120
Figure 178.	Test_Result.....	121
Figure 179.	Test_Run	122
Figure 180.	Test_Set	123
Figure 181.	Test_Step	124
Figure 182.	Test_Tag.....	125
Figure 183.	User Detail	127
Figure 184.	User Profiles	127
Figure 185.	User_Roles.....	127
Figure 186.	Virtual Contact Card	128

Distributed Immutable Data Object Data Model

Dido-DM

Introduction

Purpose

This document provides class, interface, and enumeration specification. The specification provides the details of the types being modeled within the system

Scope

This document provides a Data Model (DM) for the concepts presented in the Distributed Immutable Data Object Reference Architecture (DIDO-RA) which is presented as a separate document DIDO-RA.

Overview

This document provides a list of all packages with a summary for each. Each package has a section that contains a list of its classes, interfaces and enumeration type, with a summary for each. Class and Interface contains description, summary tables, detailed member descriptions, and relation table.

Distributed Immutable Data Object Data Model

Dido-DM

Package DIDO - Conceptual Model

Name	DIDO - Conceptual Model
Qualified Name	DIDO - Conceptual Model

Package DIDO - Conceptual Communities

The Community Package captures the ideas present in the DIDO Reference Architecture (DIDO-RA) put forth in the Object Management Group's (OMG's) Discussion Paper.

The term Distributed Immutable Data Objects (DIDO) refers to the underlying technologies supporting distributed data and computation across a distributed network of peers using consensus algorithms to maintain integrity and consistency across the network. After the publication of Satoshi Nakamoto's paper Bitcoin: A Peer-to-Peer Electronic Cash System [1] and the exponential growth of other cryptocurrencies, there is a need to understand, in general terms, the underlying DIDO architectures and provide a framework to enable engineering due diligence of DIDOs. DIDOs are not limited to cryptocurrencies, the original blockchain, or distributed ledger technologies. DIDOs are applicable to other non-cryptocurrency domains such as supply chain, registries for birth, deaths etc., and lists of authentication and identification (IA) acceptable certificates including those that have been revoked. The DIDO concepts are captured within a Reference Architecture (RA), intended to represent any architectures relying on distributed networks of peers that store data and allow parallel computation. The DIDO RA is not intended as a physical "must-have" requirements list, but more as a conceptual catalogue of what "can-be".

As illustrated in the figure, the DIDO RA is an idealized, general set of requirements and constraints. Each incarnation of the DIDO software (i.e., cryptocurrency, distributed ledger, etc.) uses the stakeholder requirements as a filter of the DIDO RA and tailors it to the unique needs and desires of its community of stakeholders. For example, the DIDO RA provides standards for logging, yet the logging requirements are driven by DIDO Software Stakeholders who may or may not opt to include it. There is a plethora of DIDO Software incarnations available starting with the original Blockchain Software that drives Bitcoin and moving onto IBM's Distributed Ledger, Ethereum and Iota.

The DIDO Software incarnations are adopted or used by another community of stakeholders that wish to leverage the DIDO Software into a DIDO Network of Nodes to address requirements and needs of a specific domain. For example, one DIDO Network Community wants to provide a cryptocurrency and another public records. The DIDO Software selected by the DIDO Network Stakeholders might be different. In Figure 1, the bi-directional arrows communicate the notion that the DIDO RA not only influences the DIDO Software and Networks, but also that evolution of DIDO software and networks feeds back into subsequent versions of the DIDO RA.

Distributed Immutable Data Object Data Model

Dido-DM

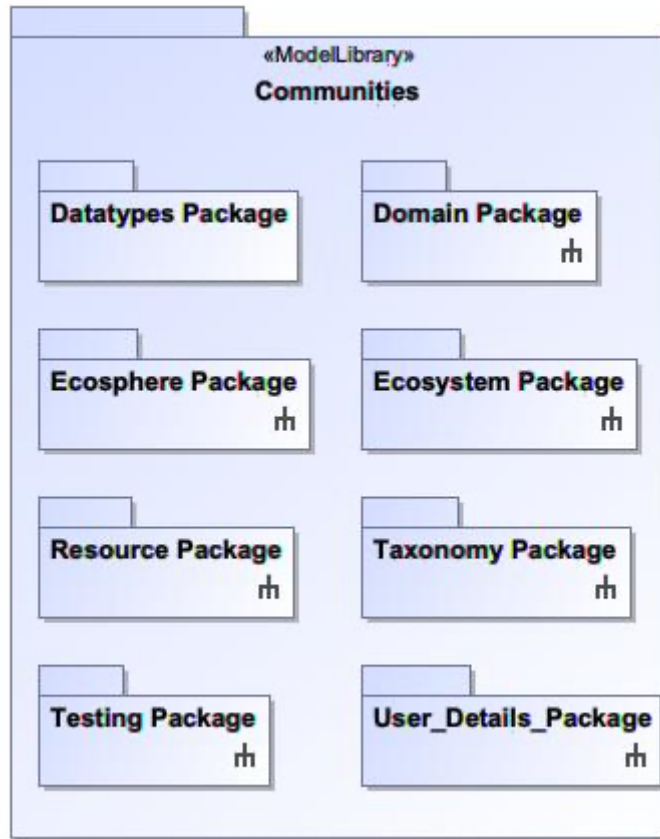


Figure 1. Communities

Name	Communities
Qualified Name	DIDO - Conceptual Communities

Package DIDO - Conceptual Datatypes Package

Name	Datatypes Package
Qualified Name	DIDO - Conceptual Datatypes Package

Base Classifier	
Realized Interface	

Enumeration Copyright Grant Type

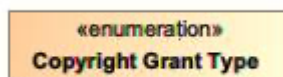


Figure 2. Copyright Grant Type

Name	Copyright Grant Type
-------------	----------------------

Distributed Immutable Data Object Data Model

Dido-DM

Qualified Name	DIDO - Conceptual Datatypes Package::Copyright Grant Type
Visibility	public
Base Classifier	

Enumeration Copyright Permission Type



Figure 3. Copyright Permission Type

Name	Copyright Permission Type
Qualified Name	DIDO - Conceptual Datatypes Package::Copyright Permission Type
Visibility	public
Base Classifier	

Enumeration Copyright Private Use Type



Figure 4. Copyright Private Use Type

Name	Copyright Private Use Type
Qualified Name	DIDO - Conceptual Datatypes Package::Copyright Private Use Type
Visibility	public
Base Classifier	

Enumeration DIDO Key Type



Figure 5. DIDO Key Type

Name	DIDO Key Type
Qualified Name	DIDO - Conceptual Datatypes Package::DIDO Key Type
Visibility	public
Base Classifier	

Distributed Immutable Data Object Data Model

Dido-DM

Enumeration Email Address Type



Figure 6. Email Address Type

Name	Email Address Type
Qualified Name	DIDO - Conceptual Datatypes Package::Email Address Type
Visibility	public
Base Classifier	

Enumeration Email Type



Figure 7. Email Type

Name	Email Type
Qualified Name	DIDO - Conceptual Datatypes Package::Email Type
Visibility	public
Base Classifier	

Enumeration Multiplicity Type

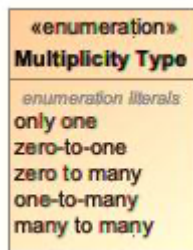


Figure 8. Multiplicity Type

Name	Multiplicity Type
Qualified Name	DIDO - Conceptual Datatypes Package::Multiplicity Type
Visibility	public
Base Classifier	

Distributed Immutable Data Object Data Model

Dido-DM

Enumeration Parameter Definition Type

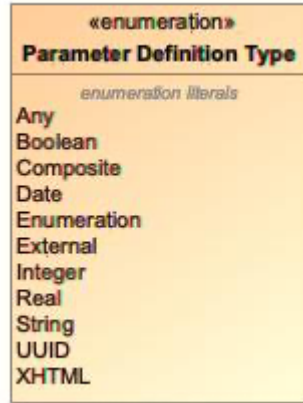


Figure 9. Parameter Definition Type

Name	Parameter Definition Type
Qualified Name	DIDO - Conceptual Datatypes Package::Parameter Definition Type
Visibility	public
Base Classifier	

Enumeration Parameter DIRECTION Type

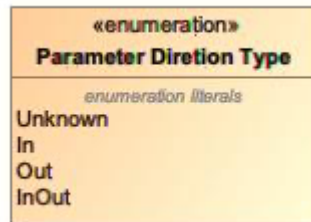


Figure 10. Parameter DIRECTION Type

Name	Parameter DIRECTION Type
Qualified Name	DIDO - Conceptual Datatypes Package::Parameter DIRECTION Type
Visibility	public
Base Classifier	

Enumeration Release Status Type

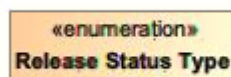


Figure 11. Release Status Type

Distributed Immutable Data Object Data Model

Dido-DM

Name	Release Status Type
Qualified Name	DIDO - Conceptual Datatypes Package::Release Status Type
Visibility	public
Base Classifier	

Enumeration Social Network Type



Figure 12. Social Network Type

Name	Social Network Type
Qualified Name	DIDO - Conceptual Datatypes Package::Social Network Type
Visibility	public
Base Classifier	

Enumeration Spoken Language Type

An Enumeration of the acceptable human languages for the Community of Interest (Col).



Figure 13. Spoken Language Type

Name	Spoken Language Type
Qualified Name	DIDO - Conceptual Datatypes Package::Spoken Language Type
Visibility	public
Base Classifier	

Enumeration Verdict Type

The Verdict Type is an enumeration of the various outcomes from a test.

Distributed Immutable Data Object Data Model

Dido-DM

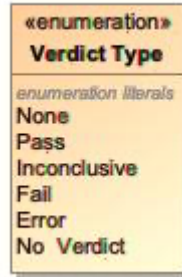


Figure 14. Verdict Type

Name	Verdict Type
Qualified Name	DIDO - Conceptual Datatypes Package::Verdict Type
Visibility	public
Base Classifier	

Package DIDO - Conceptual Domain Package

The Domain Package contains all the Entities and relationship between these entities that are directly related to the Domain COI. The Domain Package has relationships with Entities from other packages such as Ecosystem and testing.

To see the Entity with context of other Entities it links tom follow the link after the diagram.

Class Allowed Domain Copyright

The Allowed Domain Copyright are the subset of copyrights allowed by the Ecosystem and are allowed within the domain. Any other copyrights must be approved by the Ecosystem. This is important because some Copyrights can have a cascading effect on other software that uses the copyright.



Figure 15. Allowed Domain Copyright

See: [Allowed Domain Copyright](#)

Class Domain

The Domain is a DIDO Community of Interest (COI) which addresses a particular topic that is approved by the Ecosystem. The Domain is actually responsible for the entire lifecycle of a DIDO solution from architecture, design, implementation, testing, deployment, maintenance and End-of-Life (EoF). A Domain is responsible for the use of the data distributed on the Node Network (i.e., currency, rewards programs, certificates, etc.).

The Domain may get some budget from the Ecosystem but may also raise money for it day-to-day operations from other sources.

Distributed Immutable Data Object Data Model

Dido-DM

Although a Domain can stand on its own, it is most effective if it is part of an Ecosystem which is part of an Ecosphere.

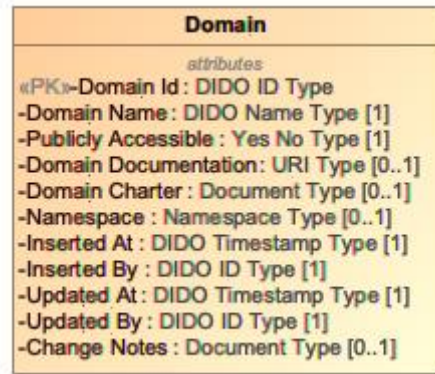


Figure 16. Domain

See: [Domain](#)

Class Domain Board Member

There can be any number of A Domain Board Members. The Chair of the Domain is automatically a member of the board so the minimum number of board members is one. The Domain Board acts as the governing board for the Domain and approve anything including requirements, reviews and release schedules. It also can appoint or remove board members.

A Domain is also responsible for the allocation of resources within the Domain.

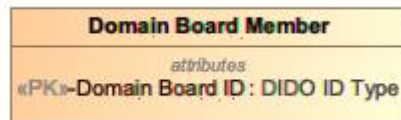


Figure 17. Domain Board Member

See: [Domain Board Member](#)

Class Domain Chair

The Domain Chair is part of the Domain Governing Board and is responsible for the entire Domain. Each Domain sets up its own Charter and Policies and Procedures (P&P) defining the extent and the limits of the power of the Chair. Some Domains can have multiple Chairs. Domain. Domain Chairs are also members of the responsible Ecosystem Board.



Figure 18. Domain Chair

See: [Domain Chair](#)

Distributed Immutable Data Object Data Model

Dido-DM

Class Domain FAQ

The Domain provides as many Frequently Asked Questions (FAQ) and Answers as is deemed necessary to provide information about the Domain. A Domain can have FAQ which are unique to itself, but also can provide FAQ from the hierarchical COIs such as Ecosystem and Ecosphere. New FAQ are created in response to questions provided by the Domains' user community.

FAQs are intended to be a supplement to the Domains Charter and Policies and Procedures (P&P) and not a replacement for either. The FAQ can cover administrative questions or technical questions and be developed as support for User's guides.



Figure 19. Domain FAQ

See: [Domain FAQ](#)

Class Domain _Glossary

Each Domain provides its own glossary which is specific to the Domain. Unless a Glossary term defined by the Domain is different from those already provided by its Ecosystem and Ecosphere, the terms should not be repeated. All glossary items which are not uniquely created for the Domain but yet are important to the Domain, should provide a definition and references for the definition.



Figure 20. Domain _Glossary

See: [Domain _Glossary](#)

Class Domain Governing Board

The Domain Governing Board is responsible for maintaining the Domain Charter and the Policies and Procedures (P&P) that control, regulate and govern the Domain. The Domain Governing Board must operate within the confines of the Ecosystem and Ecosphere they operate within.

The Domain Governing Board can either directly approve the release of new products or they can create P&P delegating the responsibility to other individuals or intelligent agents.

The Domain Governing Board has one or more chairs and a board of directors that are the representatives of the Domain on the Ecosystem Governing Board.

Distributed Immutable Data Object Data Model

Dido-DM



Figure 21. Domain Governing Board

See: [Domain Governing Board](#)

Class Domain Keychain

A Domain has a Keychain that maintains a list of Key Stores used by the Domain to sign documents (files) it publishes. The Domain Keychain can have any number of Key Stores. It does not maintain keys for the Ecosystem or Ecosphere.



Figure 22. Domain Keychain

See: [Domain Keychain](#)

Class Domain Kind

The Domain Kind is a way of classifying the different kinds of nodes that can be created within a Domain. For example, a Node might be classified as a Full Node, Wallet Node or a Permanent Node.

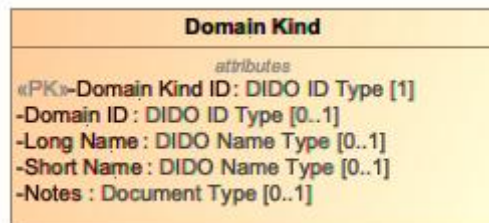


Figure 23. Domain Kind

See: [Domain Kind](#)

Class Domain Member

A Domain Member is any entity (human or intelligent agent) that has access to the Domain. Although a Domain may want to allow anonymous members to participate in the Domain, the concept of an Anonymous Domain Member still needs to be captured. A Domain may grant different privileges to different members. For example, Read, Write, Delete, Comment, etc.



Distributed Immutable Data Object Data Model

Dido-DM

Figure 24. Domain Member

See: [Domain Member](#)

Class Domain Tag

A Domain Tag is a subset of the Ecosystem and Ecosphere tags. If new tags are needed to describe the Domain, they should be officially added to the other two Communities first. This is to ensure a common vocabulary and ultimately ontology that describes the entire space. Tags are a word or simple phrase with a description that describe some aspect of the Domain.

For example: sensor, cryptocurrency, supply chain, transport, warehouse, field, etc.



Figure 25. Domain Tag

See: [Domain Tag](#)

Class Smart Contract

A smart contract is a piece of software that is executed on a Distributed Ledger when a transaction occur. These are associated with a particular Domain and on certain kinds of Nodes.



Figure 26. Smart Contract

See: [Smart Contract](#)

Package DIDO - Conceptual Ecosphere Package

The Ecosphere View comprises the set of all known Ecosystems within the Ecosphere and the interactions of the ecosystems. Outside of a limited set of use cases, a DIDO can not function independently especially when a DIDO is ultimately meant to be part of an enterprise. The description of the DIDO Ecosystem works well when everything is defined as a DIDO using greenfield development. However, greenfield development for an enterprise is probably not going to happen nor should it due to the large amount of legacy data information and processes held outside of the Ecosystem. This external data, referred to as ancillary data, is the Immutable Data Object.

The Ecosphere concept is a way to encapsulate the Ecosystem and external Ancillary data required to make the individual domains within the ecosystem functional. An individual DIDO Domain can access other data sources outside its Domain or for that matter even its Ecosystem.

Distributed Immutable Data Object Data Model

Dido-DM

References

DIDO Reference Architecture, Object Management Group (OMG) Discussion Paper, Stavros, Stavros, Turek, et al., Version 2.0, (2020)

<a View

href="https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1_views:0_stakeholder:4_ecosphere">Ecosphere

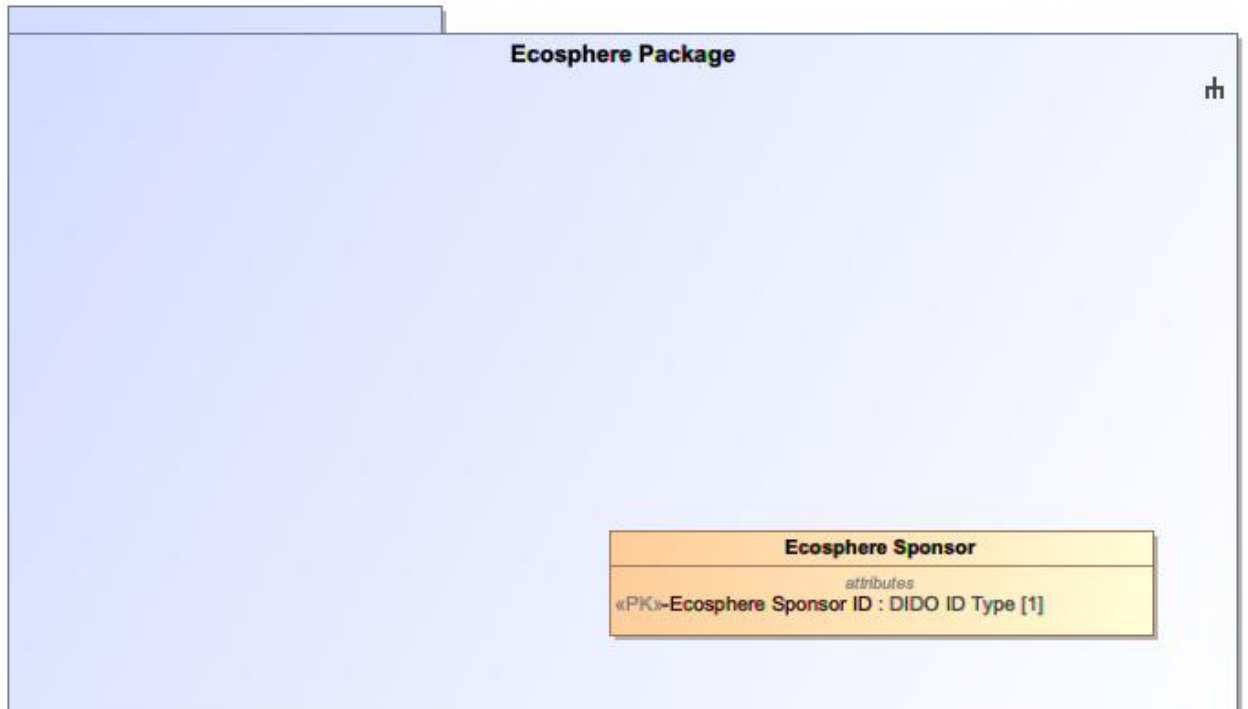


Figure 27. Ecosphere Package

Class Allowed Ecosphere Copyright

The Allowed Ecosphere Copyright are copyrights that are allowed within the Ecosphere. Ecosystems and Domains can tailor restrictions or limit the list of copyrights acceptable from their level downwards (Ecosystems can restrict sub-ecosystems and subordinate Domains.



Figure 28. Allowed Ecosphere Copyright

See: [Allowed Ecosphere Copyright](#)

Class Ecosphere

Ecosphere is responsible for a collection of Domains and Ecosystems associated with a common governance, which crosses over multiple Areas of Interest (AoI) such as military, government, automotive,

Distributed Immutable Data Object Data Model

Dido-DM

finance, etc. Examples of an Ecosphere might be DARPA, EPA, NATO, Education, Medicine. These are all areas which cover a broad range of ecosystems such finance, data storage, regulations, environment, etc. Each of these ecosystems might cover more specific Aol such as water and air quality sensors, medical records, cryptocurrency, bonds, stocks, public records, etc.

An Ecosphere is a formal organization with a Charter, policies and Procedures (P&P) that help direct and govern the organization. Ecosystems that wish to be part of the Ecosphere need to formally apply and follow the rules and regulations set out by the Ecosphere.



Figure 29. Ecosphere

See: [Ecosphere](#)

Class Ecosphere Board Member

There can be any number of a Ecosphere Board Members. The Chair of the Ecosphere is automatically a member of the board so the minimum number of board members is one. The Ecosphere Board acts as the governing board for the Ecosphere and approve anything including how Ecosystems and Domains can modify or extend their specific Charter, Policies and Procedures (P&P), allowable copyrights. etc.

A Ecosphere is also responsible for the allocation of some resources within the Ecosystem and to the Domains that fall within its jurisdiction. For example, the storage of documents, source code, etc.

The Ecosphere is also responsible for the approval of new Ecosystems.



Figure 30. Ecosphere Board Member

See: [Ecosphere Board Member](#)

Class Ecosphere Chair

The Ecosphere Chair is part of the Ecosphere Governing Board and is responsible for the entire Ecosphere. Each Ecosphere sets up its own Charter as well as Policies and Procedures (P&P) defining the extent and the limits of the power of the Chair. Some Ecosphere can have multiple Chairs.

Distributed Immutable Data Object Data Model

Dido-DM



Figure 31. Ecosphere Chair

See: [Ecosphere Chair](#)

Class Ecosphere FAQ

The Ecosphere provides as many Frequently Asked Questions (FAQ) and Answers as is deemed necessary to provide information about the Ecosphere. A Ecosphere can have FAQ which are unique to itself, but also can provide FAQ from the hierarchical COIs such as Ecosystem and Ecosphere. New FAQ are created in response to questions provided by the Ecosphere's user community.

FAQs are intended to be a supplement to the Ecosphere Charter and Policies and Procedures (P&P) and not a replacement for either. The FAQ can cover administrative questions or technical questions and be developed as support for User's guides.



Figure 32. Ecosphere FAQ

See: [Ecosphere FAQ](#)

Class Ecosphere Glossary

Each Ecosphere Glossary provides its own glossary which is specific to the Ecosphere and can be used by the Ecosystem and the Domains it encompass.



Figure 33. Ecosphere Glossary

See: [Ecosphere Glossary](#)

Class Ecosphere Governing Board

The Ecosphere Governing Board is responsible for maintaining the Ecosphere Charter and the Policies and Procedures (P&P) that control, regulate and govern the Ecosphere and subsequent Ecosystems and Domains.

The Ecosphere Governing Board has one or more chairs and a board of directors that are the representatives of the Sponsors on the Ecosphere.

Distributed Immutable Data Object Data Model

Dido-DM



Figure 34. Ecosphere Governing Board

See: [Ecosphere Governing Board](#)

Class Ecosphere Keychain

The Ecosphere Keychain is a collection of key stores used for encryption within the Ecosphere. An Ecosphere can have one or more Ecosphere Keychains. It does not store keys for ecosystems or domains.



Figure 35. Ecosphere Keychain

See: [Ecosphere Keychain](#)

Class Ecosphere Member

A Ecosphere Member is any entity (human or intelligent agent) that has access to the Ecosphere. Although a Ecosphere may want to allow anonymous members to participate in the Ecosphere, the concept of an Anonymous Ecosphere Member still needs to be captured. A Ecosphere may grant different privileges to different members. For example, Read, Write, Delete, Comment, etc.



Figure 36. Ecosphere Member

See: [Ecosphere Board Member](#)

Class Ecosphere Sponsor

Every Ecosphere requires a Sponsor that is responsible for the Ecosphere Community of Interest (COI) and all the sub-COIs (i.e., Ecosystems and Domains). Although the Sponsor may defer to the COI governing Boards for the day-to-day operation of the COIs, the sponsor can have the final word when it comes to goals and scope and make a determination of what is and what is not within the scope of the Ecosphere.

The Sponsor is responsible for paying the costs associated with the running of the Ecosphere including the repository. The Sponsor can veto any Sub-COI as part of the Ecosphere.

Distributed Immutable Data Object Data Model

Dido-DM



Figure 37. Ecosphere Sponsor

See: [Ecosphere Sponsor](#)

Class Ecosphere Tag

An Ecosphere Tag describe some attribute of the Ecosphere. The Ecosphere Tags can be used to help identify the Ecosphere and differentiate it from other Ecospheres. It also sets the stage the Ecosystems and Domains that are part of the Ecosphere.

For example: government, military, financial, Environmental, Education, IOT, IIOT, etc



Figure 38. Ecosphere Tag

See: [Ecosphere Tag](#)

Class Ecosphere Version Control

The Ecosphere Version Control contains the list of Version Control Systems (VCS) allowed in the Ecosphere and Ecosystem or Domains within the Ecosphere. The data contained within tis table is required to access the VCS.

Version control systems (VCSs) are a category of software tools that helps record changes to files by keeping a track of modifications done to the code. Use of Version Control System: A repository: It can be thought as a database of changes. It contains all the edits and historical versions (snapshots) of the project. <https://www.geeksforgeeks.org/version-control-systems/>

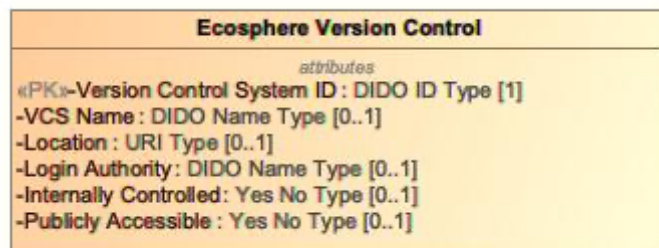


Figure 39. Ecosphere Version Control

See: [Ecosphere Version Control](#)

Distributed Immutable Data Object Data Model

Dido-DM

Class Allowed Ecosys Copyright

The Allowed Ecosystem Copyright are the subset of copyrights allowed by the Ecosphere and are allowed within the ecosystem. Any other copyrights must be approved by the Ecosphere. This is important because some Copyrights can have a cascading effect on other software that uses the copyright.



Figure 40. Allowed Ecosys Copyright

See: [Allowed Ecosys Copyright](#)

Class Ecosys Board Member

There can be any number of A Ecosystem Board Members. The Chair of the Ecosystem is automatically a member of the board so the minimum number of board members is one. The Ecosystem Board acts as the governing board for the Ecosystem and approve anything including how Domains can modify or extend their specific Charter, Policies and Procedures (P&P), allowable copyrights. etc.

A Ecosphere is also responsible for the allocation of resources within the Ecosystem and to the Domains that fall within its jurisdiction. It is also responsible for the approval of new Domains.



Figure 41. Ecosys Board Member

See: [Ecosys Board Member](#)

Class Ecosys Chair

The Ecosystem Chair is part of the Ecosystem Governing Board and is responsible for the entire Ecosystem. Each Ecosystem sets up its own Charter as well as Policies and Procedures (P&P) defining the extent and the limits of the power of the Chair. Some Ecosystems can have multiple Chairs. Ecosystem Chairs are also members of the responsible Ecosphere Board.



Figure 42. Ecosys Chair

See: [Ecosys Chair](#)

Distributed Immutable Data Object Data Model

Dido-DM

Class Ecosys FAQ

The Ecosystem provides as many Frequently Asked Questions (FAQ) and Answers as is deemed necessary to provide information about the Ecosystem. An Ecosystem can have FAQ that are unique to itself, but also can provide FAQ from the hierarchical COIs such as Ecosphere and Domains it is responsible for. New FAQ are created in response to questions provided by the Ecosystem's user community.

FAQs are intended to be a supplement to the Ecosystem Charter and Policies and Procedures (P&P) and not a replacement for the either. The FAQ can cover administrative questions or technical questions and be developed as support for User's guides.

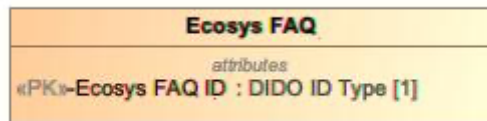


Figure 43. Ecosys FAQ

See: [Ecosys FAQ](#)

Class Ecosys Glossary

Each Ecosystem provides its own glossary which is specific to the Ecosystem. Unless a Glossary term defined by the Ecosystem is different from those already provided by its Ecosphere, the terms should not be repeated. All glossary items which are not uniquely created for the Ecosystem but yet are important to the Ecosystem, should provide a definition and references for the definition.



Figure 44. Ecosys Glossary

See: [Ecosys Glossary](#)

Class Ecosys Governing Board

The Ecosystem Governing Board is responsible for maintaining the Ecosystem Charter and the Policies and Procedures (P&P) that control, regulate and govern the Ecosystem. The Ecosystem Governing Board must operate within the confines of the Ecosphere they operate within.

The Ecosystem Governing Board has one or more chairs and a board of directors that are the representatives of the Ecosystem on the Ecosphere Governing Board.



Distributed Immutable Data Object Data Model

Dido-DM

Figure 45. Ecosys Governing Board

See: [Ecosys Governing Board](#)

Class Ecosys Keychain

An Ecosystem has a Keychain that maintains a list of Key Stores used by the Ecosystem to sign documents (files) it publishes. The Ecosystem Keychain has any number of Key Store. It does not maintain keys for the Domain or Ecosphere.



Figure 46. Ecosys Keychain

See: [Ecosys Keychain](#)

Class Ecosys Member

A Ecosystem Member is any entity (human or intelligent agent) that has access to the Ecosystem. Although a Ecosystem may want to allow anonymous members to participate in the Ecosystem, the concept of an Anonymous Ecosystem Member still needs to be captured. A Ecosystem may grant different privileges to different members. For example, Read, Write, Delete, Comment, etc.



Figure 47. Ecosys Member

See: [Ecosys Member](#)

Class Ecosys Tag

An Ecosystem Tag is a subset of the Ecosphere tags. If new tags are needed to describe the Ecosystem, they should be officially added to the Ecosphere Community first. This is to ensure a common vocabulary and ultimately ontology that describes the entire space. Tags are a word or simple phrase with a description describing some aspect of the Ecosystem.

For example: air, water, asset, commodity, currency etc.



Figure 48. Ecosys Tag

Distributed Immutable Data Object Data Model

Dido-DM

See: [Ecosys Tag](#)

Class Ecosystem

Ecosystem is responsible for a collection of Domains associated with a particular Area of Interest (AoI) such as green groceries, interest rate swaps, a particular tank, class of automobiles, etc.). An Ecosystem is part of a larger Community of Interest (COI) called an Ecosphere (i.e., US Government, World Health Organization, NATO, etc.) An Ecosystem operates within the Charter of its governing Ecosphere and can extend or refine the Ecosphere's Charter, Policy and Procedures, but it can not have conflicts with it.

The Ecosystem may get some budget from the Ecosphere but may also raise money for its day-to-day operations from other sources.

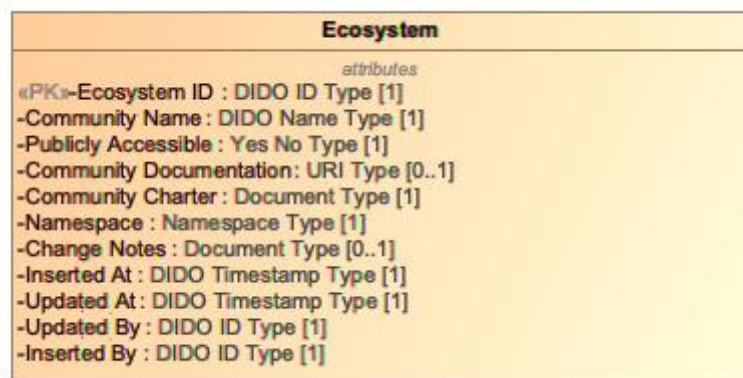
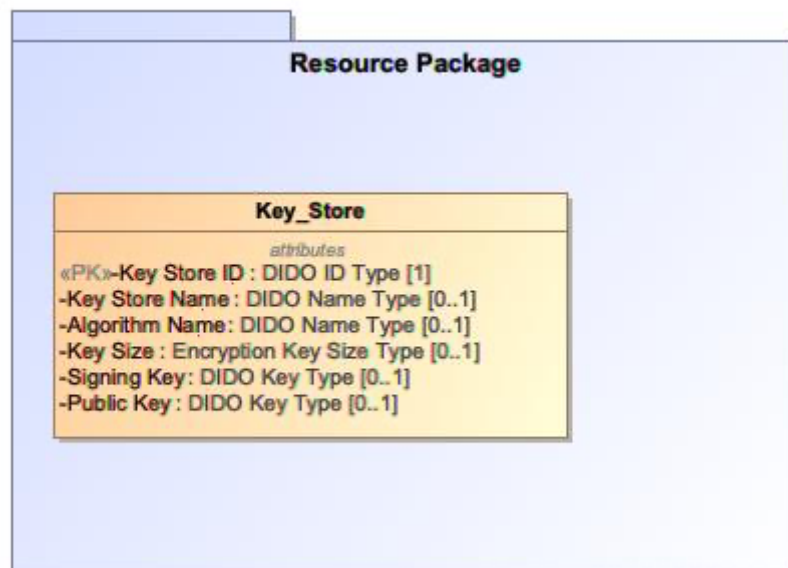


Figure 49. Ecosystem

See: [Ecosystem](#)

Package DIDO - Conceptual Resource Package



Distributed Immutable Data Object Data Model

Dido-DM

Figure 50. Resource Package

Name	Resource Package
Qualified Name	DIDO - Conceptual Resource Package

Class Copyright_Notice

The Copyright Notice describes the known copyrights describe for DIDOs. This s not a static list and can be added to but is provided as a reference for the Copyright. It is not intended to be a replacement for the authoritative source for the copyright but to provide meta-data that describes the copyright,

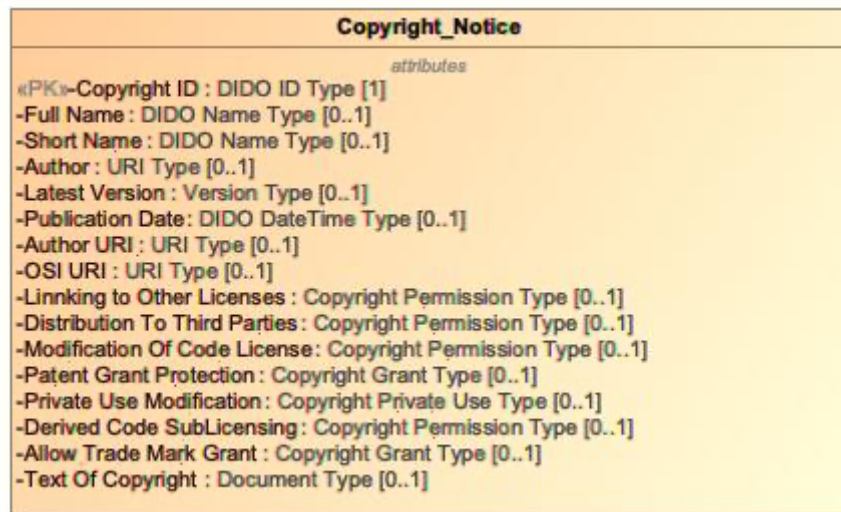


Figure 51. Copyright_Notice

See: [Copyright Notice](#)

Class External Reference

An External Reference provides a place to describe citations that are external the Ecosphere, Ecosystem or Domain.

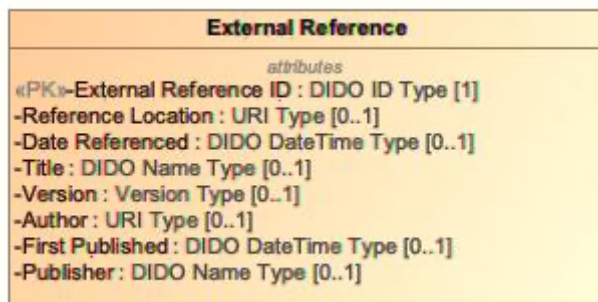


Figure 52. External Reference

See: [External Reference](#)

Distributed Immutable Data Object Data Model

Dido-DM

Class FAQ Answer

The Frequently Asked Question (FAQ) Answer is the text representing an answer to one or more Community of Interest (Col) questions.

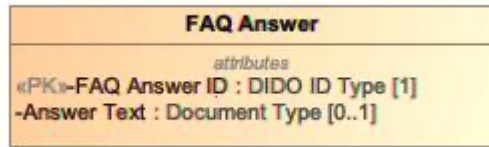


Figure 53. FAQ Answer

See: [FAQ Answers](#)

Class FAQ Question

The Frequently Asked Question (FAQ) is the text representing a question asked by a Community of Interest (Col). A single question might have zero to many Answers.

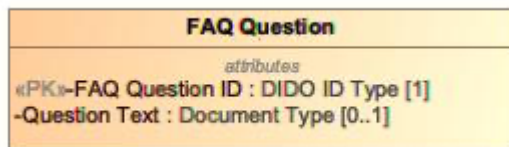


Figure 54. FAQ Question

See: [FAQ Answers](#)

Class Image Content

The image Content represents common images used within the DIDO Reference Implementation (DIDO-RI). The image can be stored locally or be a reference to an external source.

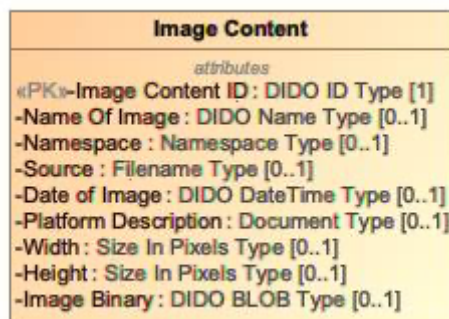


Figure 55. Image Content

See: [Image Content](#)

Distributed Immutable Data Object Data Model

Dido-DM

Class Key_Store

The Key Store is a repository for encryption public and private (i.e., signing) keys and the name of the associated encryption algorithm.



Figure 56. Key_Store

See: [Key Store](#)

Package DIDO - Conceptual Runtime_Environment_Package

Distributed Immutable Data Object Data Model

Dido-DM

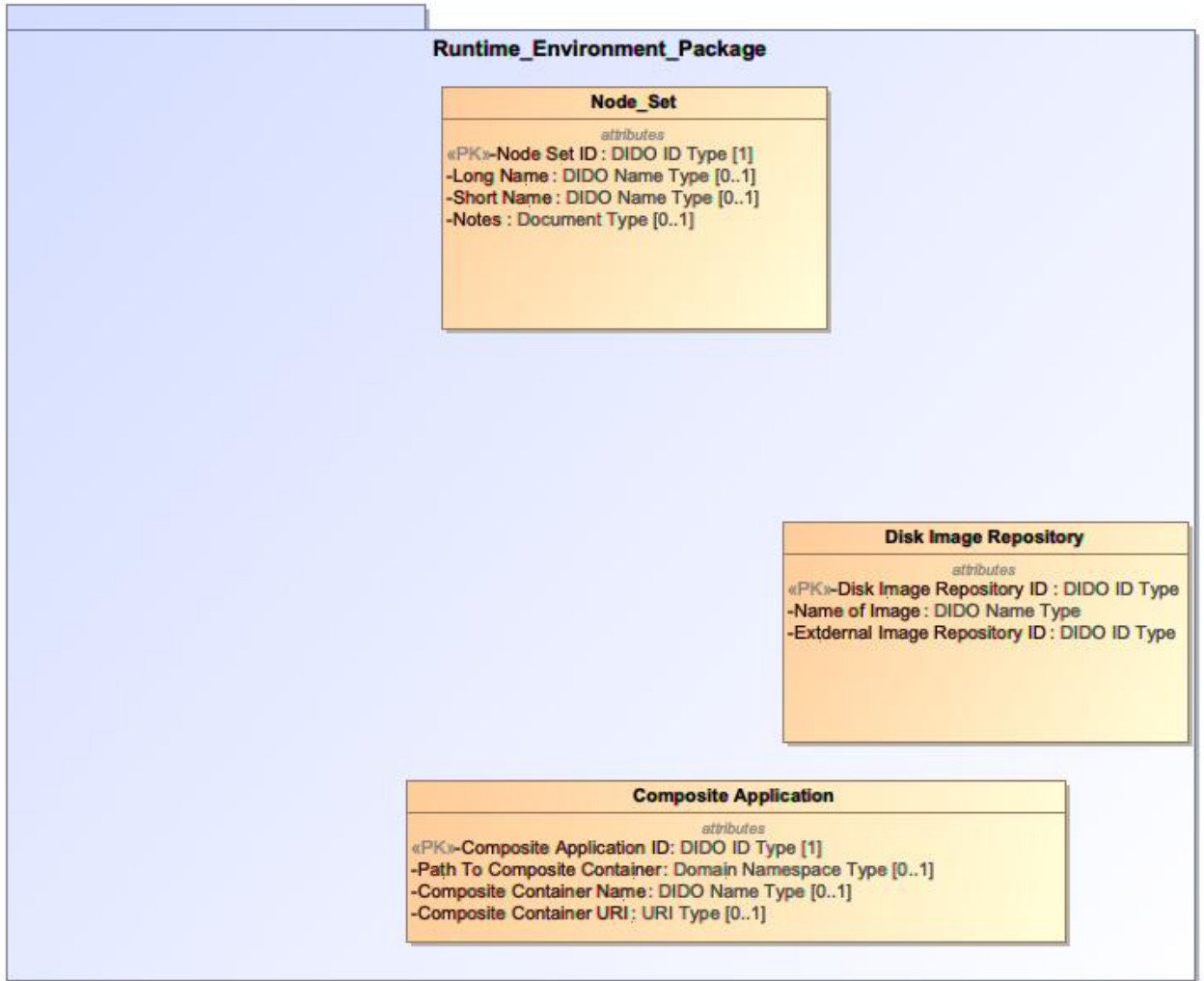


Figure 57. Runtime_Environment_Package

Name	Runtime_Environment_Package
Qualified Name	DIDO - Conceptual Runtime_Environment_Package

Class Application Container

The Application Container describes a container used to encapsulate an application. The Application Container has a Name, a URI where the containers are stored and a Application specific path.

Containers are a technology for packaging the (compiled) code for an application along with the dependencies it needs at run time. Each container that you run is repeatable; the standardization from having dependencies included means that you get the same behavior wherever you run it.

Containers decouple applications from underlying host infrastructure. This makes deployment easier in different cloud or OS environments. <https://kubernetes.io/docs/concepts/containers/>

Distributed Immutable Data Object Data Model

Dido-DM

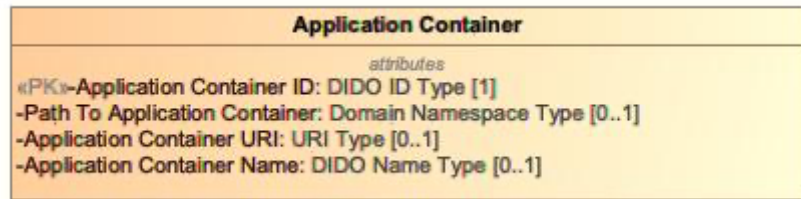


Figure 58. Application Container

See: [Application Container](#)

Class Application_Container_Set

The Application Container set is a way of making a collection of containers that can be readily reused.

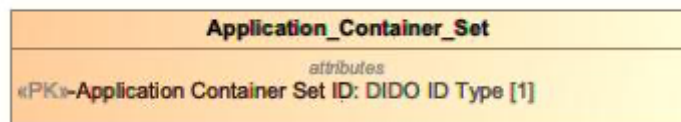


Figure 59. Application_Container_Set

See: [Application Container Set](#)

Class Cloud Provider Supported_OS

The Cloud Provider Supported OS defines Operating System (OS) platforms that are available for a particular Cloud Provider. An entry can be enabled or disabled to allow for the construction and deprecation of OSs.

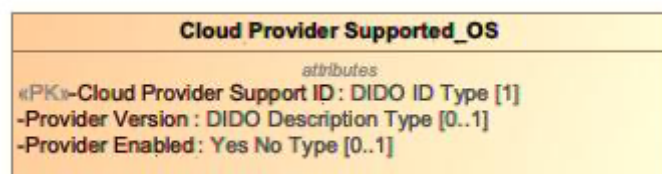


Figure 60. Cloud Provider Supported_OS

See: [Cloud Provider Supported_OS](#)

Class Cloud_Provider

The Cloud Provider provides information where the virtual DIDO Test Environment will be hosted. Examples, would be Amazon Web Services (AWS), Google Cloud, Microsoft Azure.

Distributed Immutable Data Object Data Model

Dido-DM

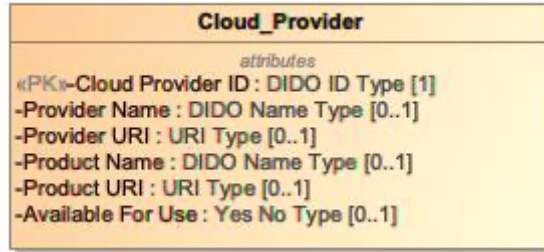


Figure 61. Cloud_Provider

See: [Cloud_Provider](#)

Class Composite Application

Composite applications are applications built from a combination of multiple existing functions using business sources of information. Composite applications are software asset collections assembled to provide business capability. These assets are generally artifacts deployed independently enabling composition and leveraging of specific platform capabilities.

Using a composite application can relieve a user from switching between applications. It provides ready access to multiple applications at the same place, with the additional advantage of manually adding and removing features. Composite applications can be compared with mashups. However, composite applications use business sources of information while mashups use Web-based, mostly free resources. See: <https://www.techopedia.com/definition/169/composite-applications>

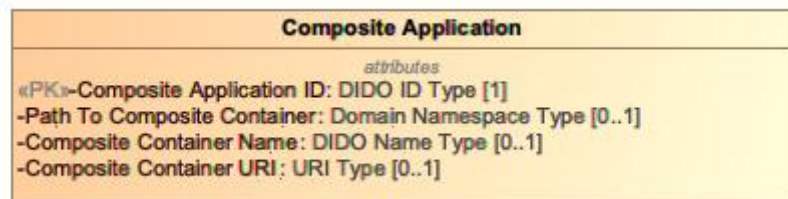


Figure 62. Composite Application

See: [Composite Application](#)

Class CPU Architecture

The CPU Architecture provides a list of Central Processing Unit (CPU) architectures.

For Example:

Intel 32-bit (x86)

The central processing unit (CPU) or processor, is the unit which performs most of the processing inside a computer. It processes all instructions received by software running on the PC and by other hardware components, and acts as a powerful calculator. See: <https://www.techopedia.com/definition/2851/central-processing-unit-cpu>

Distributed Immutable Data Object Data Model

Dido-DM



Figure 63. CPU Architecture

See: [CPU Architecture](#)

Class DIDO Platform

The DIDO Platform describes the various DIDO platforms that are available. For example, Ethereum, Iota, Hyperledger, etc. All Node Platforms described within the Ecosphere, Ecosystem or Domain must be associated with at least one Node Platform.

Any particular DIDO Platform can have multiple entries that reflect the version of the platform (i.e., latest, V1.12.3b, etc).

Not all entries in the DIDO Platforms are available for use. This may be because they are "in work" and not ready for use, or because the particular DIDO Platform is deprecated.

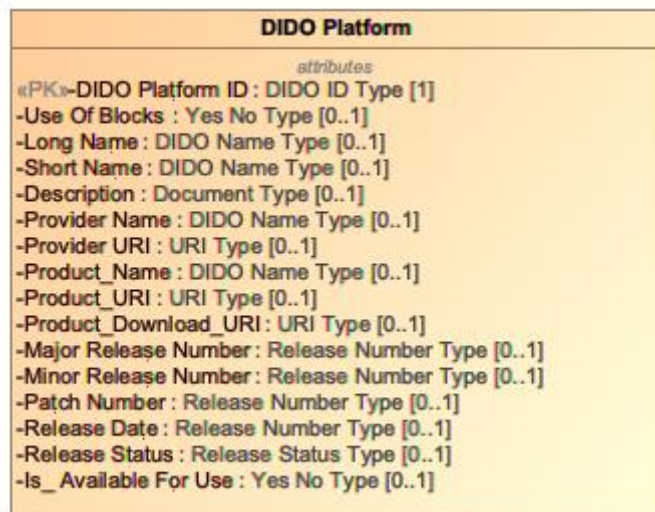


Figure 64. DIDO Platform

See: [DIDO Platform](#)

Class Disk Image Repository

The Disk Image Repository is a catalog of Disk Images available for use with the DIDO Community.

A disk image is a single file or storage device that holds a replica of all data on a storage medium or device, such as a hard drive, tape drive, CD, DVD, floppy disk or key drive. A disk image is usually created through a sector-by-sector replication of the original - or source - storage medium, including the structure (directories and folders) and contents (files).

Distributed Immutable Data Object Data Model

Dido-DM

Disk image is a noun and should be distinguished from disk cloning, which is a verb that describes the process of copying disk contents to another storage medium or image file.

See: <https://www.techopedia.com/definition/12705/disk-image>

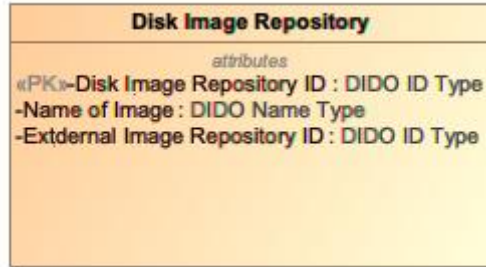


Figure 65. Disk Image Repository

See: [Disk Image Repository](#)

Class Node Platform

The Node Platform describes the two main components that describe a Node: a DIDO Platform and a Operating System Platform. It also allows for the nodes to have a classification indicating what kind of node it is. For example, a full node, a wallet, a permanent node, etc. Each of these kinds of nodes would have the same DIDO Platform and OS Platform.



Figure 66. Node Platform

See: [Node Platform](#)

Class Node Profile

A Node Profile represents the definition of a collection of nodes. The Node profile specifies the Number of processors required for the node and the amount of memory, storage associated with the node. It also specifies the number of nodes in the profile. This allows this Profile to be reused in testing and to represent conceptual aggregation of Nodes.

Distributed Immutable Data Object Data Model

Dido-DM

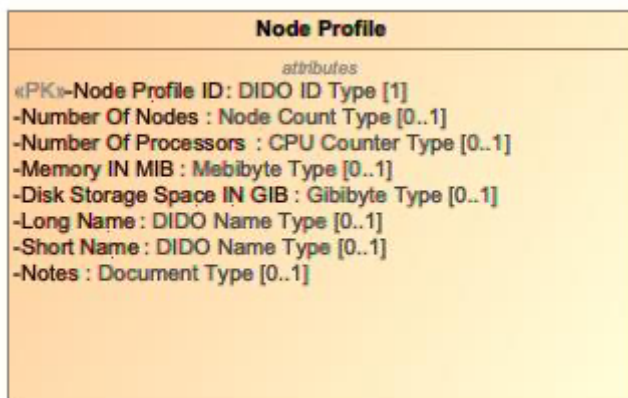


Figure 67. Node Profile

See: [Node Profile](#)

Class Node_Classification

The classifications are generalizations; each Domain may define more or fewer types of nodes with different roles. For example, Bitcoin really defines only two kinds of Nodes: Full Node and Lightweight Nodes. Full Nodes have the entire copy of the Ledger and can create Transactions on their own. Lightweight Nodes must work with a Full Node in order to synchronize the current “true” value of the data. Bitcoin also has a Mining Node which is a Full Node used to validate that a block of transactions is “true”. Nodes are generally classified by the level of engagement with the DIDO, however, any domain can define or modify the definitions for node types.

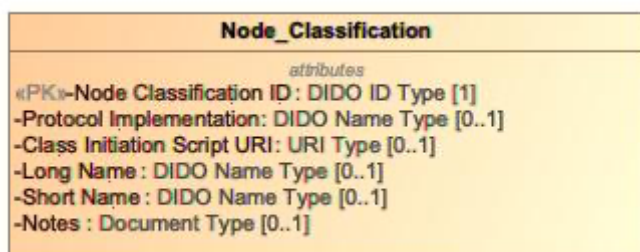


Figure 68. Node_Classification

See: [Node_Classification](#)

Class Node_Set

A Node Set provides a way to organize collections of Node Profiles and Runtime Environments into reusable components that can be referred to by name. This allows for repeatability of the Node Sets in virtualized testing and can cover regression testing and support Assurance Test Cases.

Distributed Immutable Data Object Data Model

Dido-DM

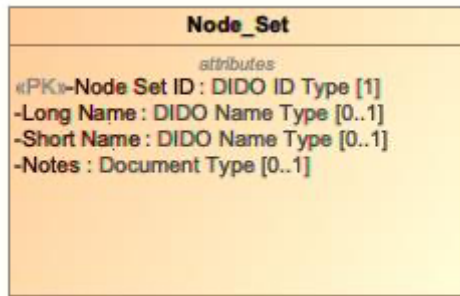


Figure 69. Node_Set

See: [Node_Set](#)

Class OS Type

The Operating System (OS) Type provides a way to describe an Operating System that can be used by Application Containers, Cloud Provider Supported OSs, in Test Executables. It is possible that any particular OS Type could be used by any or all of the Application Containers, Cloud Provider Supported OSs, or the Test Executables.

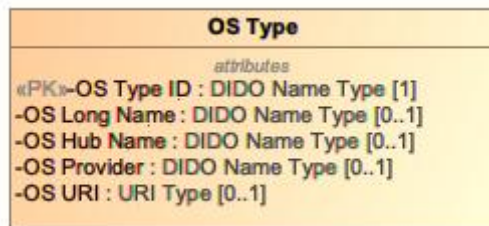


Figure 70. OS Type

See: [OS Type](#)

Class OS_Platform

The Operating System (OS) Platform describes all the information required to describe a particular version of an Operating System and is associated with various Cloud Providers and DIDO Nodes.

The description includes long and short names, release information, descriptions of the Operating System and of the provider of the OS. It also includes information on the license for the Operating System.

Distributed Immutable Data Object Data Model

Dido-DM

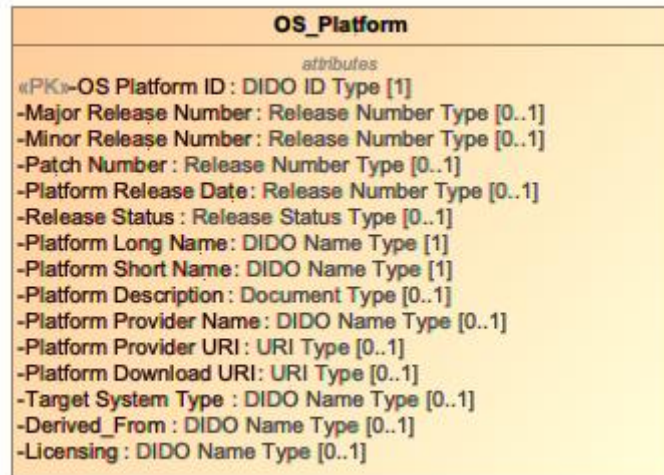


Figure 71. OS_Platform

See: [OS_Platform](#)

Class Runtime Environment

The Runtime Environment is used by Domains and Test Sets. It describes the requirements needed an application. The Environment is not a simple OS and DIDO Platform. It is the DIDO Runtime Environment which spans multiple nodes.

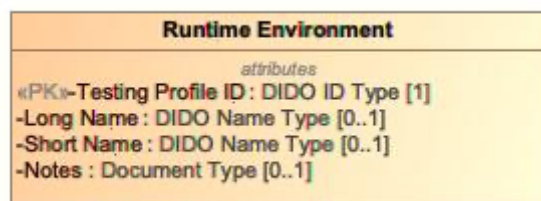


Figure 72. Runtime Environment

See: [Runtime Environment](#)

Class Sign On Provider

The Sign On Provider is the name of the organization that performs the user's authentication. This might be local or it might be an organization such as Github, Amazon, Google, LinkedIn, etc.



Figure 73. Sign On Provider

Distributed Immutable Data Object Data Model

Dido-DM

See: [Signon Provider](#)

Package DIDO - Conceptual Taxonomy Package



Figure 74. Taxonomy Package

Name	Taxonomy Package
Qualified Name	DIDO - Conceptual Taxonomy Package

Class Glossary Entry

A Glossary Entry describes definition within a Glossary. should provide a term, definition, a relevant abbreviation or acronym and a references for the definition.

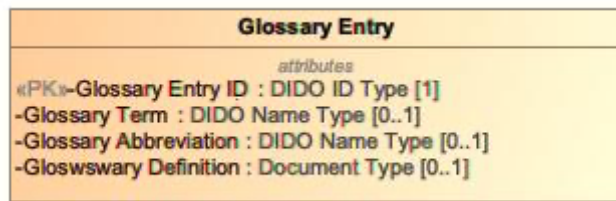


Figure 75. Glossary Entry

See [Glossary Entry](#)

Class Tag

A Tag is a word or small expression used to help provide attribution to objects within the DIDO. An example of an objects are Ecosphere, Ecosystem, Domain or Test. An Example of tags are: crypto-currency, supply chain, IOT, etc.



Figure 76. Tag

See: [Tag](#)

Distributed Immutable Data Object Data Model
Dido-DM

Package DIDO - Conceptual Testing Package

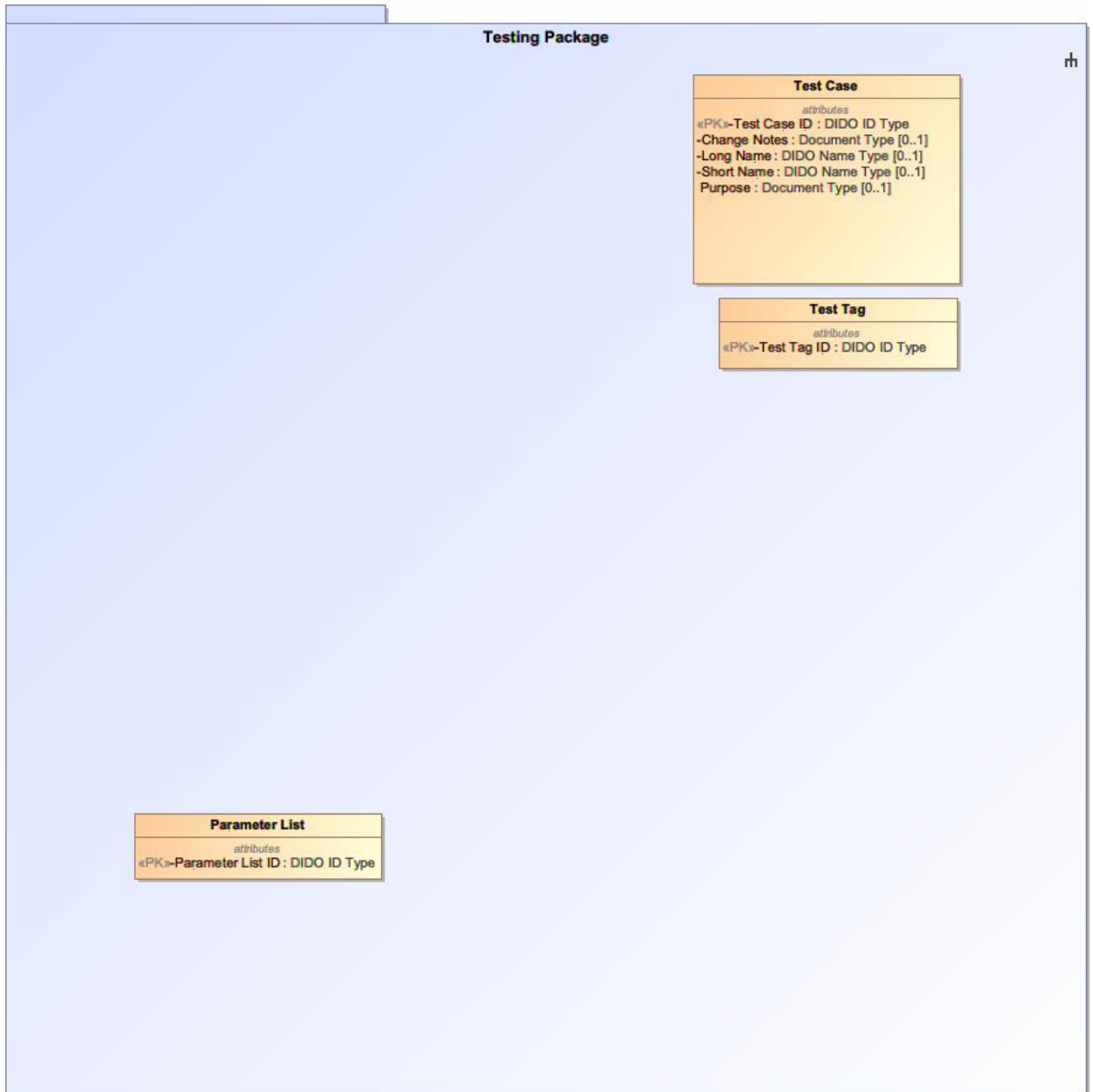


Figure 77. Testing Package

Name	Testing Package
Qualified Name	DIDO - Conceptual Testing Package

Class Expected Results

The Expected Results are the the data that is expected to be generated by a successful Test Step or Test Case. Test Results are compared against Test Results generated during the execution of Test Steps or Test Cases.

Distributed Immutable Data Object Data Model

Dido-DM



Figure 78. Expected Results

See: [Expected Results](#)

Class Parameter Definition

The Parameter Definition describes any particular parameter within a parameter list. The definition includes information on if the parameter is required or optional, is multiplicity (cardinality), the direction of the parameter (in, out, inout) and its default value.

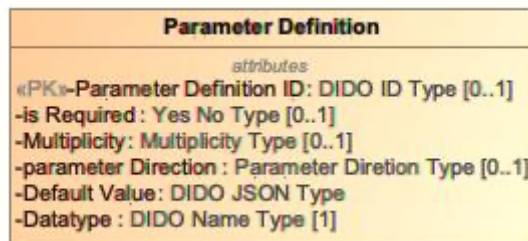


Figure 79. Parameter Definition

See: [Parameter Definition](#)

Class Parameter List

The Parameter List is a collection of Parameters passed into Test Sets, Test Steps and Test Executables.



Figure 80. Parameter List

See: [Parameter List](#)

Class Test

A Test is comprised of Test Plans and Test Results and is annotated with a series of Tags that can be used for searching. Every Test is placed must have a Purpose describing the purpose of the test, a Long and Short Name, and have change notes describing any changes made to the Test.

Distributed Immutable Data Object Data Model

Dido-DM

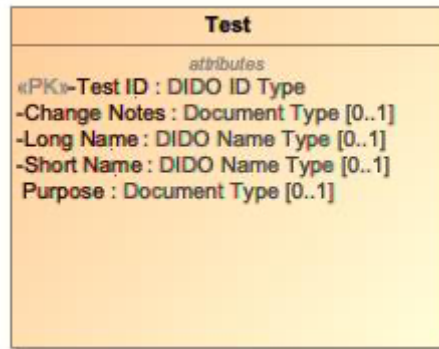


Figure 81. Test

See: [Test](#)

Class Test Case

A Test Case describes a particular goal for a Test. For example, a Test Case might be with minimal number of Nodes, homogeneous Nodes, Heterogeneous Nodes, Disadvantaged Intermittent network Links (DIL), with percent storage and without. All of the Test Cases are intended to fulfill the purpose of a particular Test Set.

The Test Case has a Short and Long Names, a purpose and notes about any changes made to the Test Case.

A Test Case can be "stand-alone" with its own Test Executable or it can be further refined into a set of Test Steps, each with their own executable.

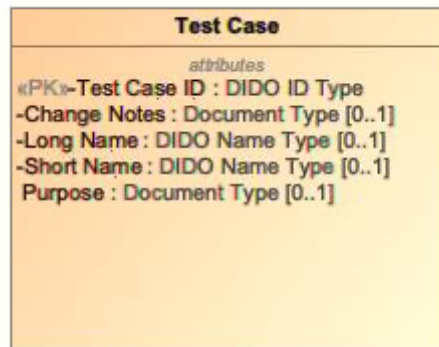


Figure 82. Test Case

See: [Test Case](#)

Class Test Executable

7.4.7 Class TestExecutable

Test Executable is similar to Test Step, for use by steps that contain low-level procedural (code/script) step definitions

(intended for automation code).

See: <https://www.omg.org/spec/TestIF/1.0/PDF>

Distributed Immutable Data Object Data Model

Dido-DM

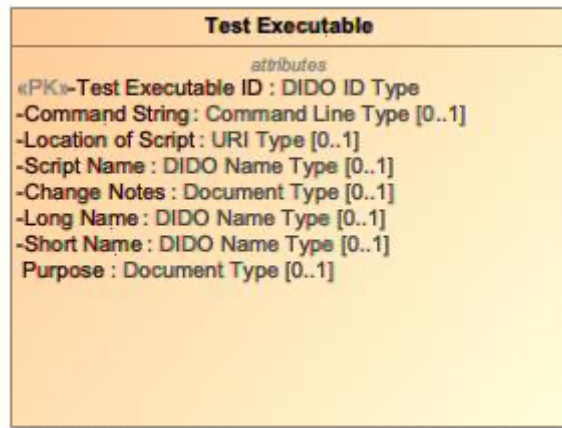


Figure 83. Test Executable

See: [Test Executable](#)

Class Test Plan

A Test Plan lays out the methodology used to verify and validate a Test. There can be one to many Test Plans associated with a Test. The Test Plan has a Short and a Long names, series of descriptive tags, and has a purpose describing goals of the Test Plan. It uses a series of Test Sets describing how the Test Plan is to be executed.

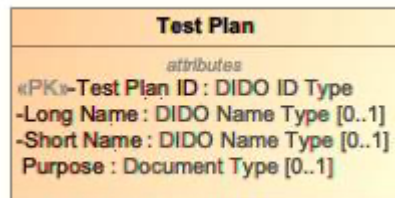


Figure 84. Test Plan

See: [Test Plan](#)

Class Test Result

The Test Results represents the results of running a Test, a Test Case or a Test Step. The Test Results contain a Verdict on the success of the Test, The Test Initiation and CompletionTimes, a text string that has path e.g., (URI) to where the Results are found and a test specific file name for the results.

Distributed Immutable Data Object Data Model

Dido-DM

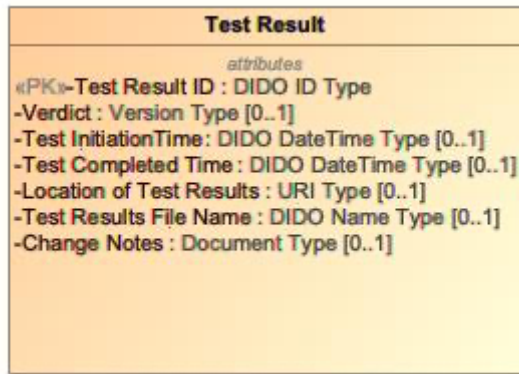


Figure 85. Test Result

See: [Test_Result](#)

Class Test Run

A Test Run captures the data associated with any particular run of a Test Executable. The Test Run is then aggregated into a Test Result. There is a one-many relationship between Test Results and Test Run.

The Test Run captures the details such as start and stop times of the executable.

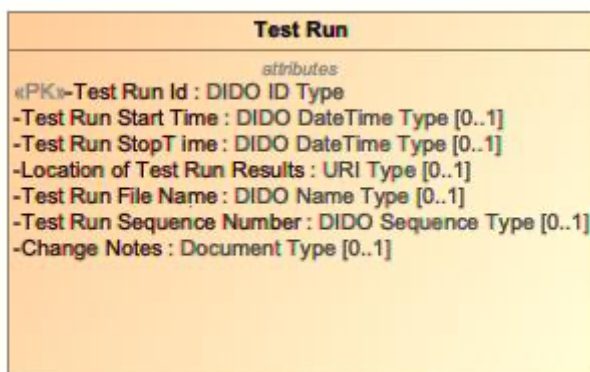


Figure 86. Test Run

See: [Test_Run](#)

Class Test Set

Test Purpose provides a textual description of the reason for a test. This field may be left blank.

See: 7.4.15.1.1 testPurpose of OMG TestIF spec: <https://www.omg.org/spec/TestIF/1.0/PDF>

Distributed Immutable Data Object Data Model

Dido-DM

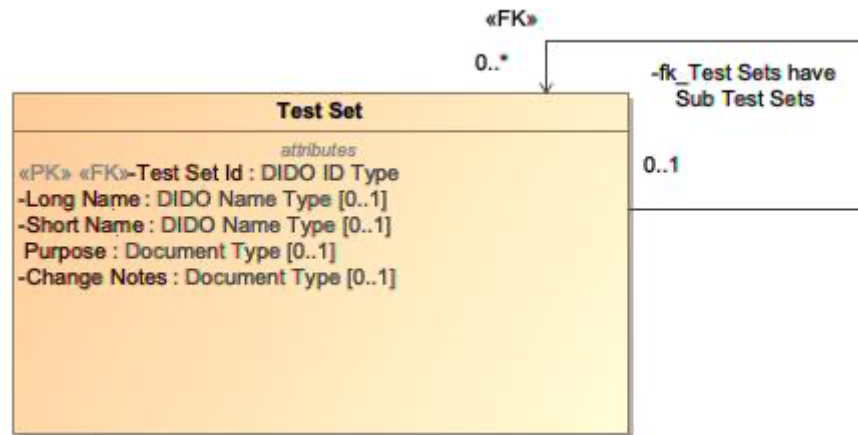


Figure 87. Test Set

See: [Test_Set](#)

Class Test Step

A Test Step represents the lowest level of a Test. Each Step step is Orchestrated by a Test Case and can itself be a collection of Test Steps. The Test Step has a Short and a Long names, and has a purpose describing goals of the Test Step.

The Test Step is associated with one or more Test Executables and corresponding Test Results.

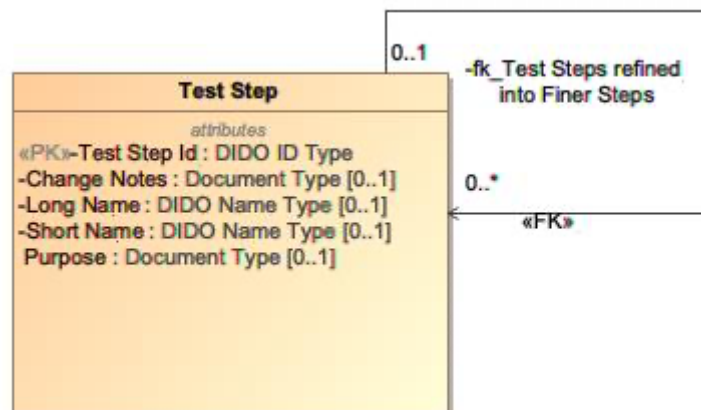


Figure 88. Test Step

See: [Test_Step](#)

Class Test Tag

An Test Tag is a subset of the Domain tags. If new tags are needed to describe the Domain, they should be officially added to the Domain Community first. This is to ensure a common vocabulary and ultimately ontology that describes the entire space. Tags are a word or simple phrase with a description describing some aspect of the Test.

For example: Real-Time, regression test, stress test, boundary conditions, vulnerabilities etc.

Distributed Immutable Data Object Data Model

Dido-DM



Figure 89. Test Tag

See: [Test Tag](#)

Package DIDO - Conceptual

User_Details_Package

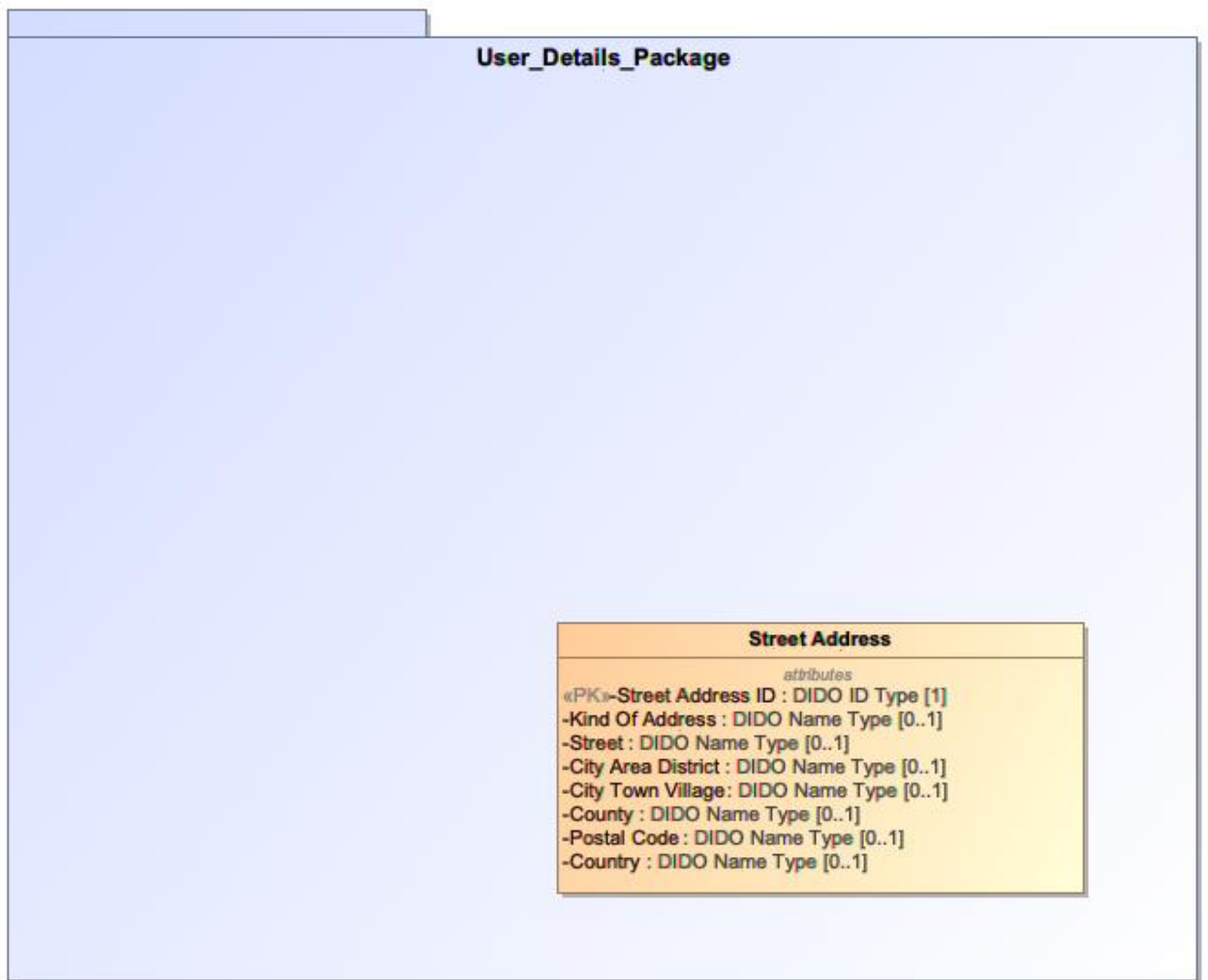


Figure 90. User_Details_Package

Name	User_Details_Package
Qualified Name	DIDO - Conceptual User_Details_Package

Distributed Immutable Data Object Data Model

Dido-DM

Class Email

The Email is text containing an email address. The Email entries also describe the Kind Of Email such as home, office, etc. A way to link emails together is to provide a Primary email address. The primary email can be the same or different from the email described in the entry.

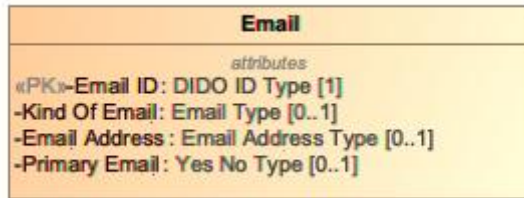


Figure 91. Email

See: [Email](#)

Class Phone Number

The Phone Number captures a primary and secondary phone number which is a sequence of digits assigned to a fixed-line telephone subscriber station connected to a telephone line or to a wireless electronic telephony device, such as a radio telephone or a mobile telephone, or to other devices for data transmission via the public switched telephone network (PSTN) or other public and private networks. A telephone number serves as an address for switching telephone calls using a system of destination code routing. https://en.wikipedia.org/wiki/Telephone_number

There are different kinds of Phone Numbers: Work, Home, Fax, etc.

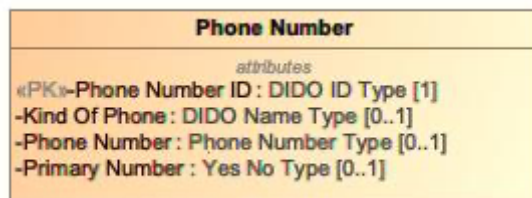
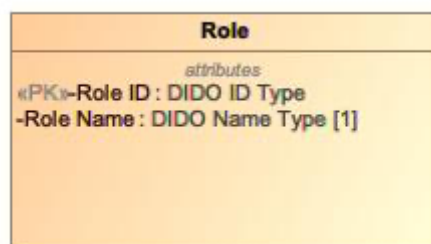


Figure 92. Phone Number

See: [Phone Number](#)

Class Role

Rle contains a list of Role Names that are recognized by the Community of Interest (Col).



Distributed Immutable Data Object Data Model

Dido-DM

Figure 93. Role

See: [Role](#)

Class Social Network Connection

The Social Network Connection allows the COI users to access other well established Social Networks such as LinkedIn, Github or Google.



Figure 94. Social Network Connection

See: [Social Network Connection](#)

Class Street Address

The Street Address contains the information needed to describe an actual physical address. For example, street name, city, State/Province, Country and postal code. The Kind of Address indicates the address, such as home or office.

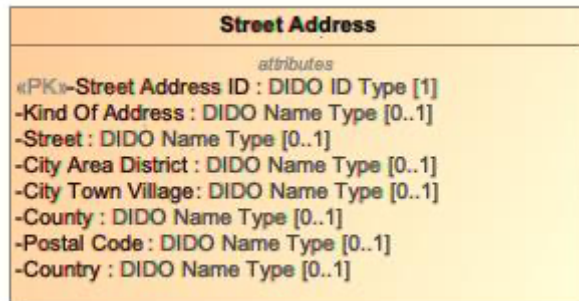


Figure 95. Street Address

See: [Street Address](#)

Class User Details

The User Details describes a User within a Community of Interest (COI) such as Ecosphere, Ecosystem or Domain. User can be actual people and also intelligent agents that play a role within the COI. The User Details help provide Social Network Connections, Virtual Contact Card and User roles.

Distributed Immutable Data Object Data Model

Dido-DM

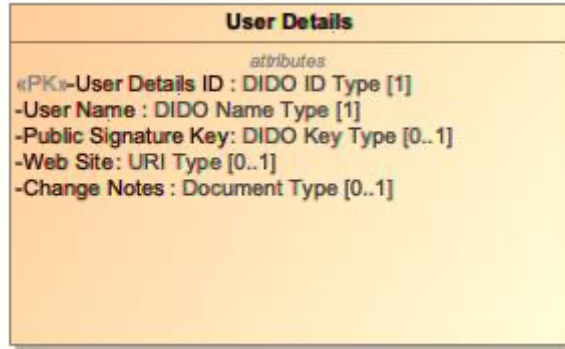


Figure 96. User Details

See: [User Detail](#)

Class User Profile

The User Profile associates the User Details (i.e., User) with a Sign On Provider identity. This allows the User to use the Sign On Provider credentials to access the COI.



Figure 97. User Profile

See: [User Profiles](#)

Class User_Role

User Role represents a collection of named Roles associated with any particular User. In order to perform certain operations, the User must be assigned a role that permits those operations.



Figure 98. User_Role

See: [User Roles](#)

Distributed Immutable Data Object Data Model

Dido-DM

Class Virtual Contact Card

The Virtual Contact Card (vCard) represents the information defined by IEF RFC 2526 for White Page listings. It captures information about an individual especially for exchanging a variety of information about an individual (e.g., formatted and structured name and delivery addresses, email address, multiple telephone numbers, photograph, logo, audio clips, etc.). See: <https://www.ietf.org/rfc/rfc2426.txt>

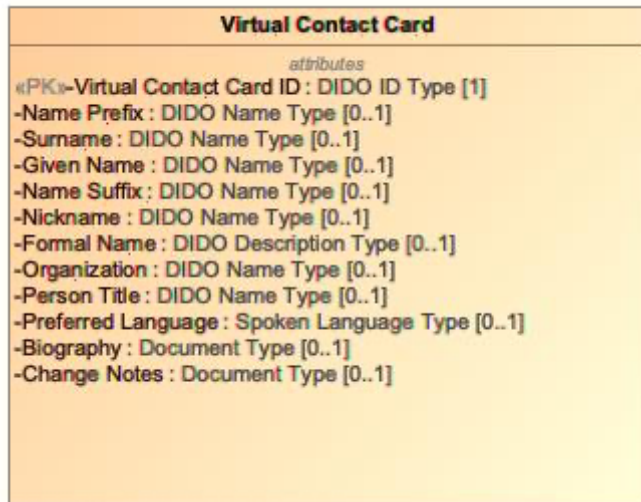


Figure 99. Virtual Contact Card

See: [Virtual Contact Card](#)

Appendix A: Class Context Diagrams

This appendix goes beyond just showing the Entity and its attributes. It provides the Entity in a class Diagram with the context of all the other Entities it links to.

To get a description of the Entity, follow the link back to the Entity Description page (usually the first link after the diagram).

Allowed Domain Copyright

The Allowed Domain Copyright are the subset of copyrights allowed by the Ecosystem and are allowed within the domain. Any other copyrights must be approved by the Ecosystem. This is important because some Copyrights can have a cascading effect on other software that uses the copyright.

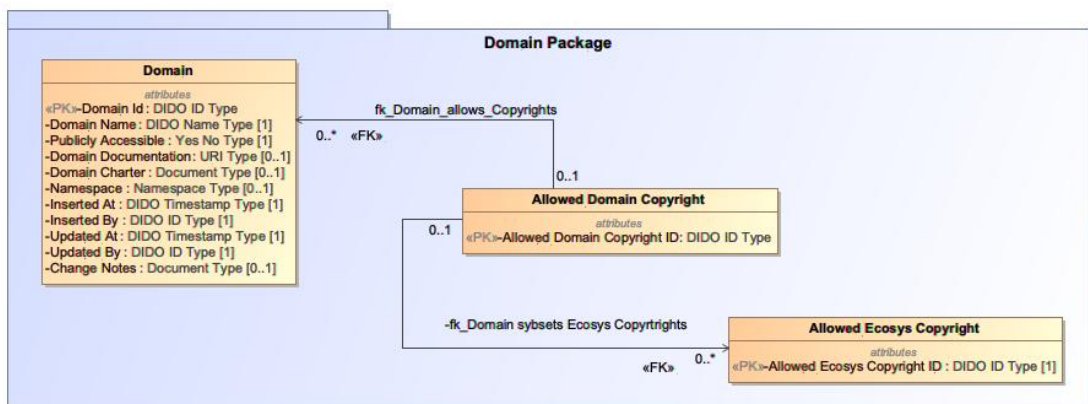


Figure 100. Allowed Domain Copyright

See:

- [Class Domain](#)
- [Class Allowed Domain Copyright](#)
- [Class Allowed Ecosys Copyright](#)

Distributed Immutable Data Object Data Model

Dido-DM

Allowed Ecosphere Copyright

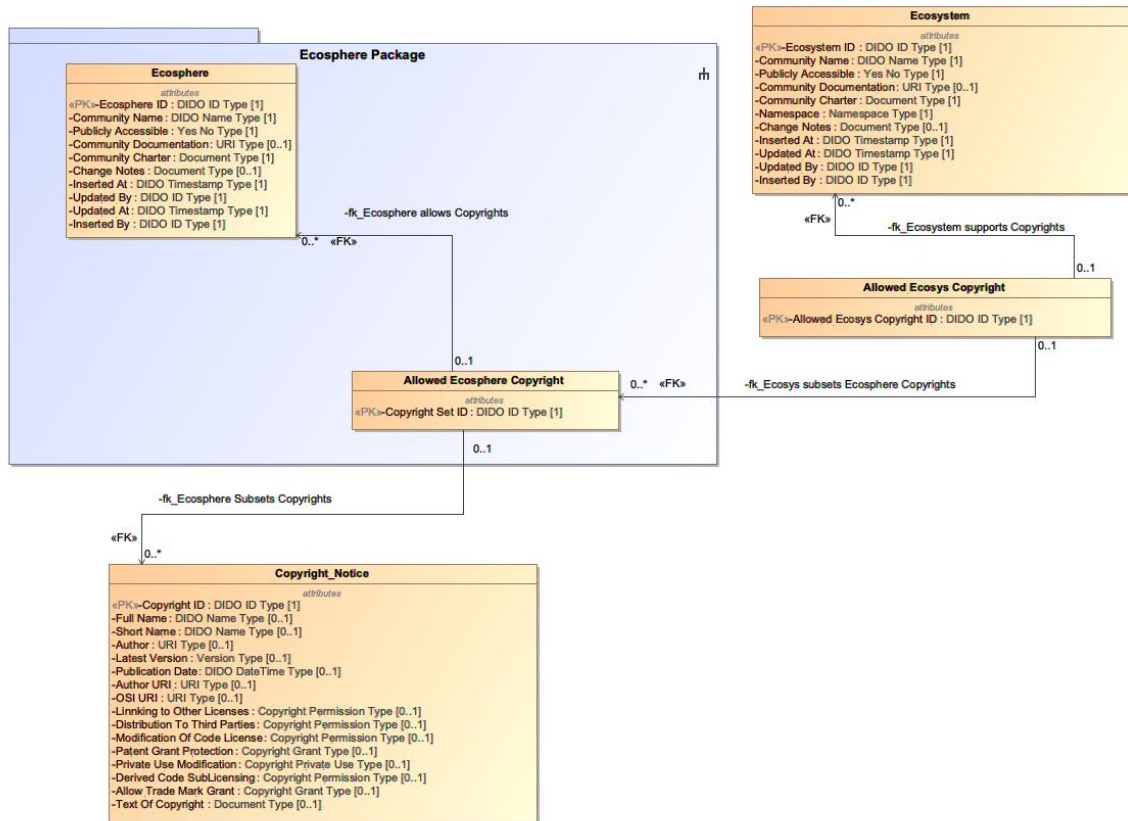


Figure 101. Allowed Ecosphere Copyright

See:

- [Class Allowed Ecosphere Copyright](#)
- [Class Ecosphere](#)
- [Class Ecosystem](#)
- [Class Allowed Ecosys Copyright](#)
- [Class Copyright_Notice](#)

Distributed Immutable Data Object Data Model

Dido-DM

Allowed_Ecosys Copyright

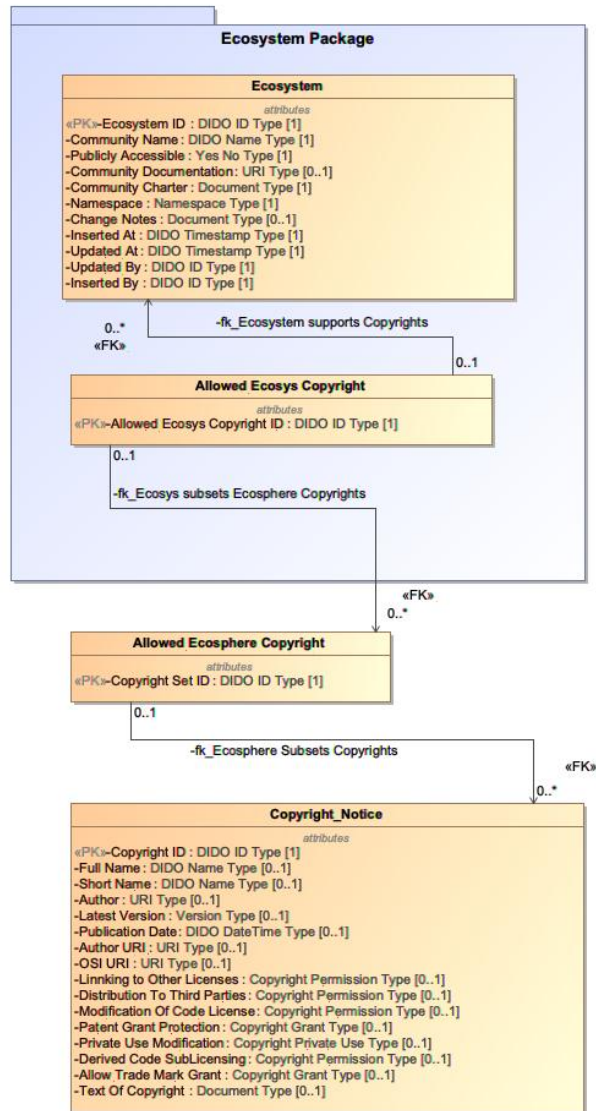


Figure 102. Allowed_Ecosys Copyright

- [Class Allowed Ecosys Copyright](#)
- [Class Ecosystem](#)
- [Class Allowed Ecosphere Copyright](#)
- [Class Copyright Notice](#)

Distributed Immutable Data Object Data Model

Dido-DM

Application Container

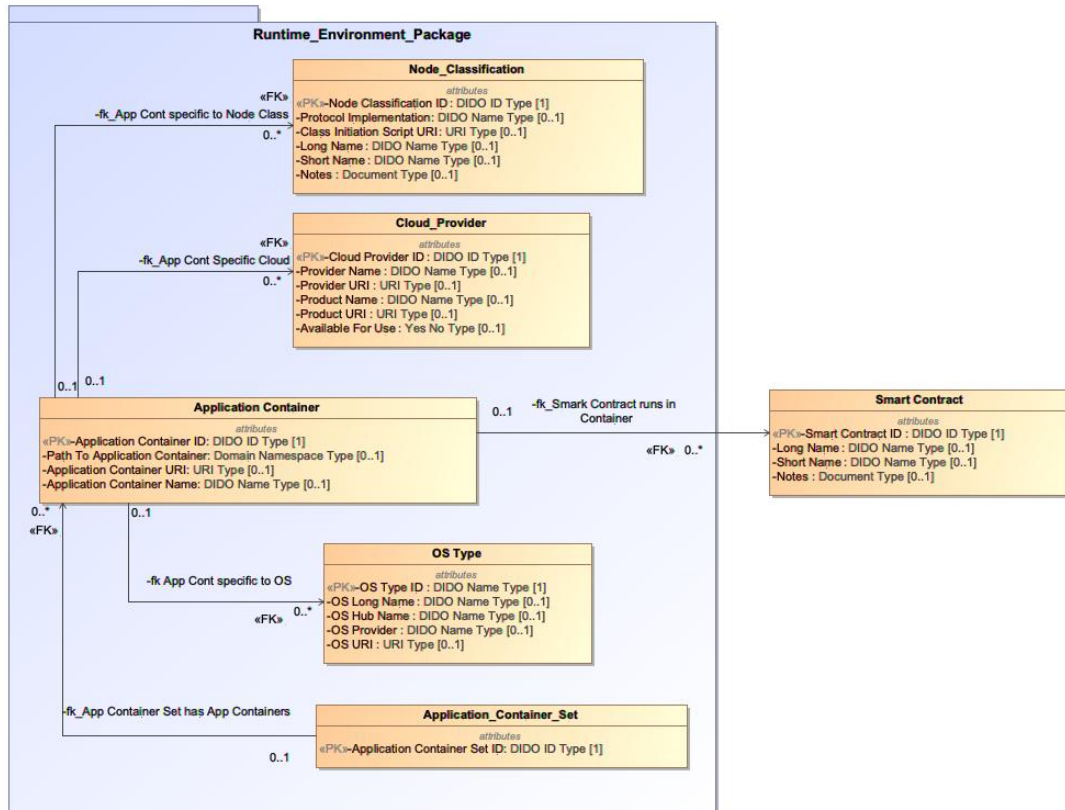


Figure 103. Application Container

See:

- [Class Application Container](#)
- [Class Node Classification](#)
- [Class Cloud Provider](#)
- [Class Smart Contract](#)
- [Class OS Type](#)
- [Class Application Container Set](#)

Distributed Immutable Data Object Data Model

Dido-DM

Application Container Set

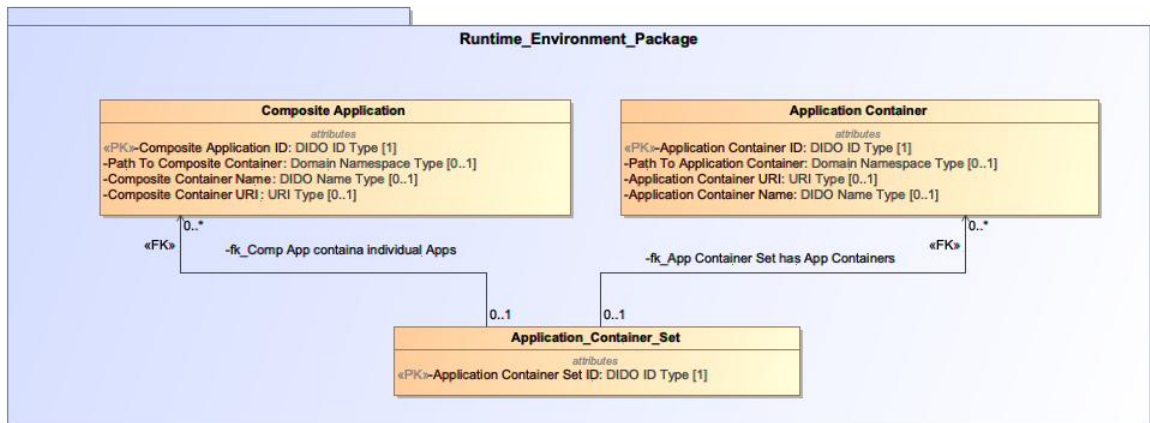


Figure 104. Application Container Set

See:

- [Class Application Container Set](#)
- [Class Application Container](#)
- [Class Application Container Set](#)

Distributed Immutable Data Object Data Model

Dido-DM

Cloud Provider Supported_OS

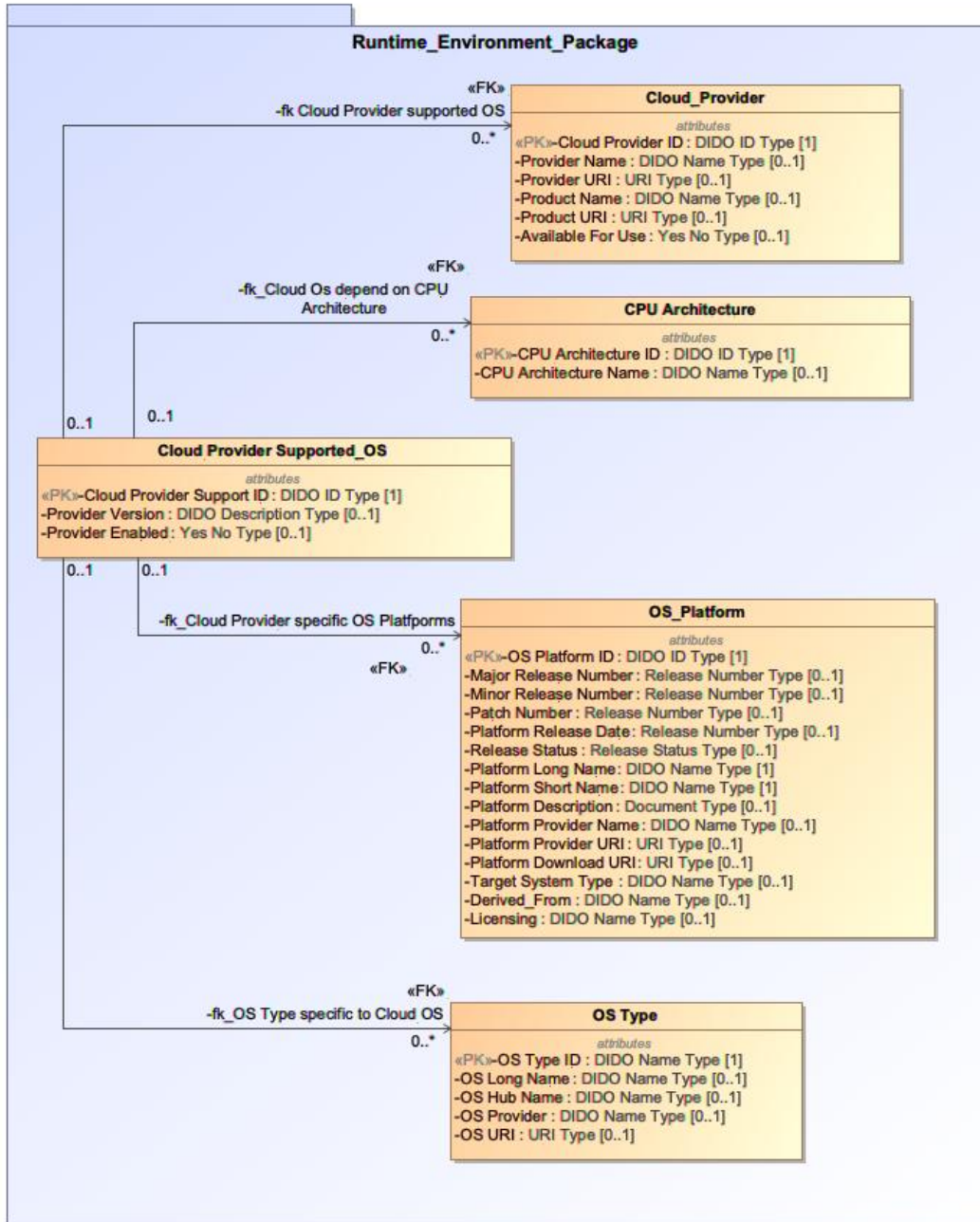


Figure 105. Cloud Provider Supported_OS

See:

- [Class Cloud Provider Supported_OS](#)
- [Class Cloud Provider](#)
- [Class CPU Architecture](#)
- [Class OS Platform](#)
- [Class OS Type](#)

Distributed Immutable Data Object Data Model

Dido-DM

Cloud_Provider

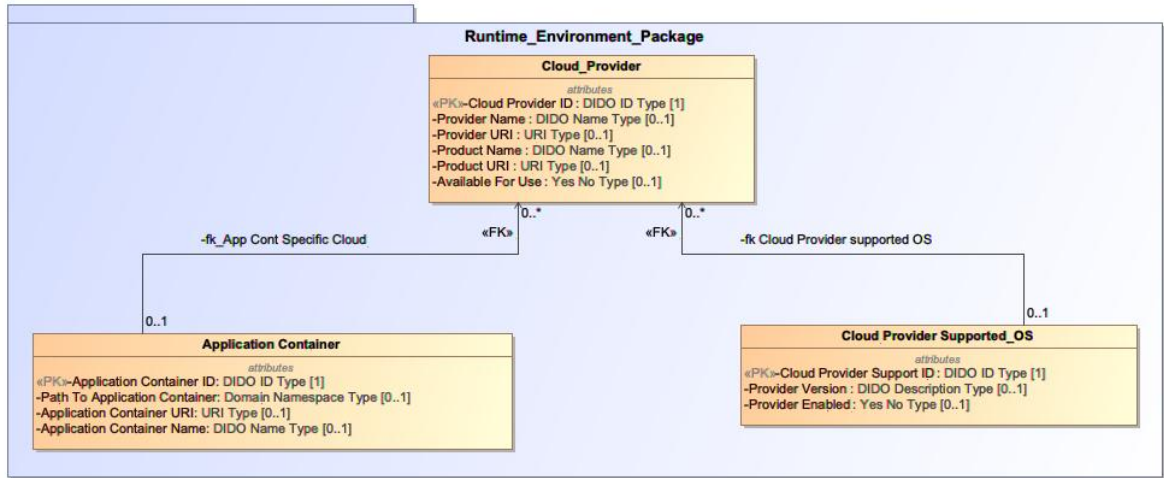


Figure 106. Cloud_Provider

See:

- [Class Cloud_Provider](#)
- [Class Application Container](#)
- [Class Cloud Provider Supported_OS](#)

Distributed Immutable Data Object Data Model

Dido-DM

Composite Application

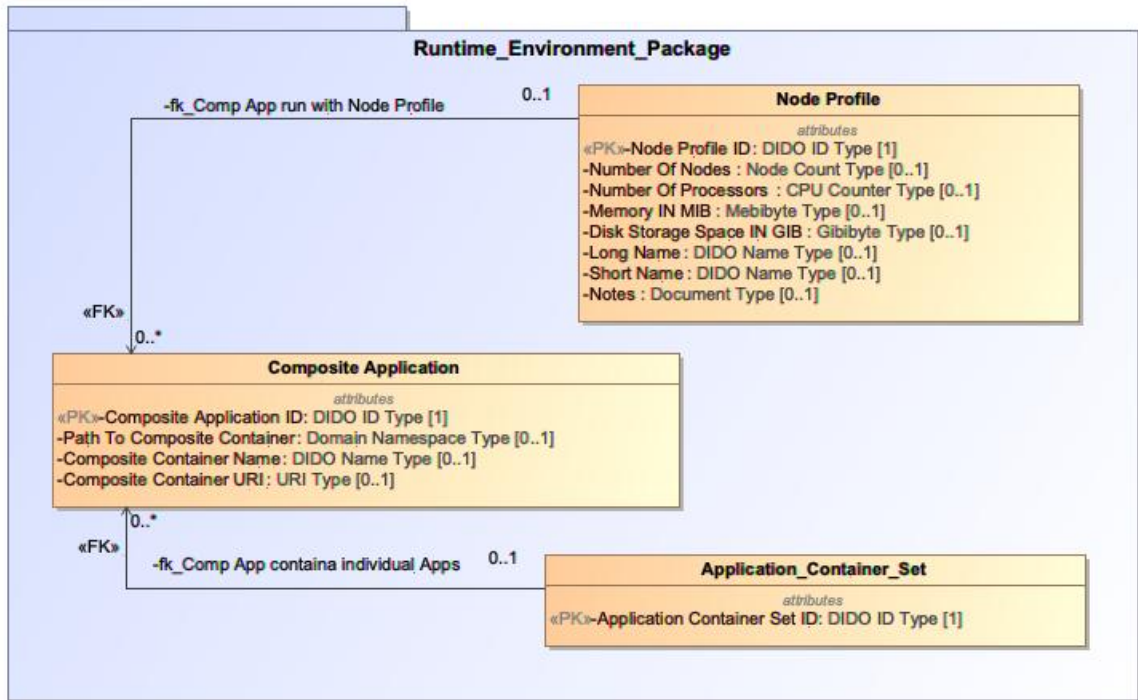


Figure 107. Composite Application

See:

- [Class Composite Application](#)
- [Class Node Profile](#)
- [Class Application_Container_Set](#)

Distributed Immutable Data Object Data Model

Dido-DM

Copyright Notice

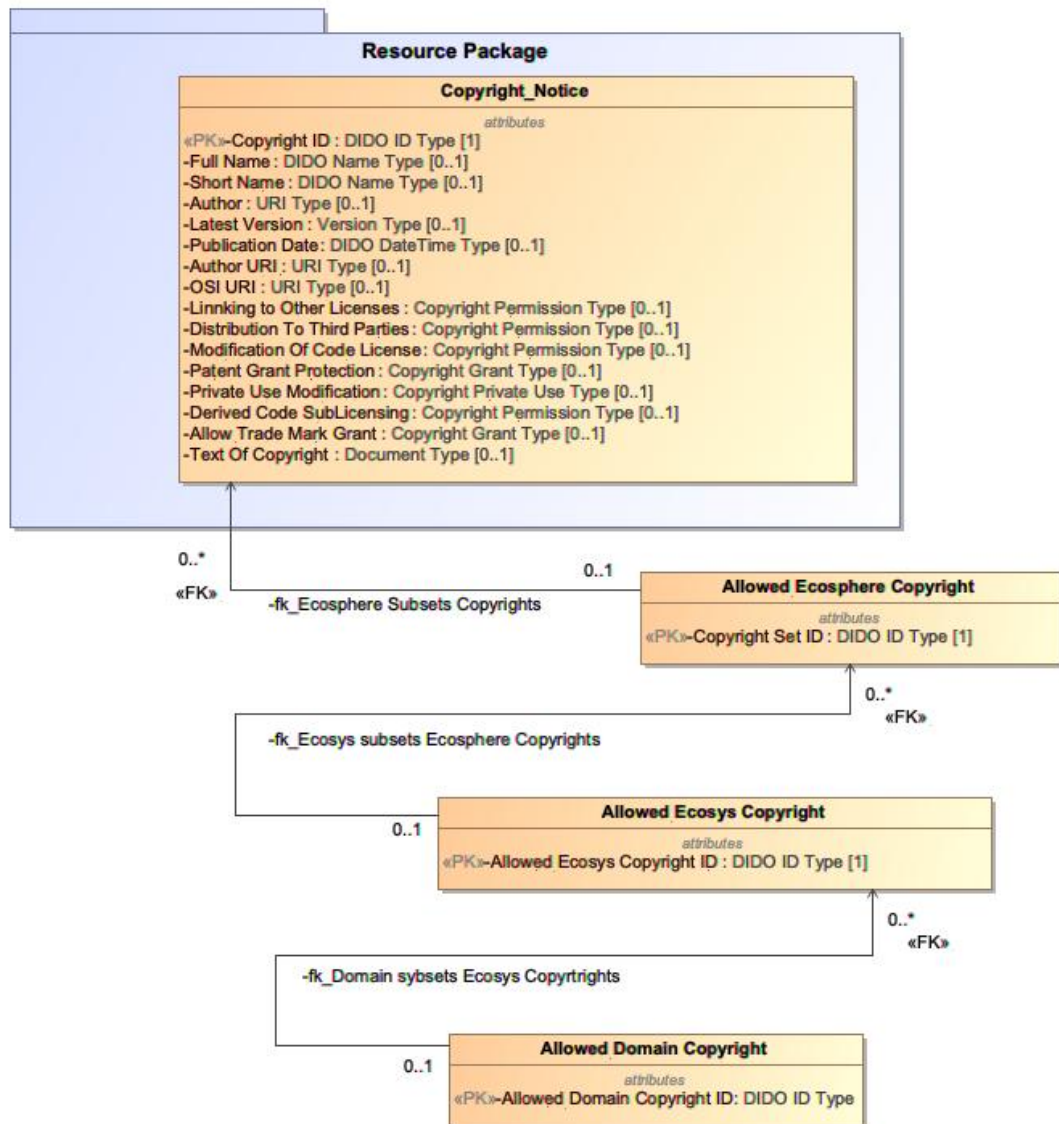


Figure 108. Copyright Notice

See:

- [Class Copyright Notice](#)
- [Class Allowed Ecosphere Copyright](#)
- [Class Allowed Ecosys Copyright](#)
- [Class Allowed Domain Copyright](#)

Distributed Immutable Data Object Data Model

Dido-DM

CPU Architecture

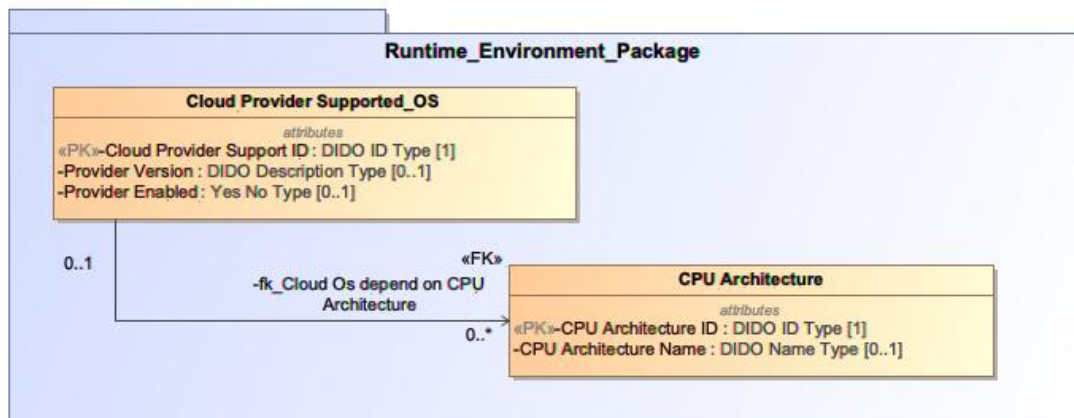


Figure 109. CPU Architecture

See:

- [Class CPU Architecture](#)
- [Class Cloud Provider Supported_OS](#)

Distributed Immutable Data Object Data Model

Dido-DM

DIDO Platform

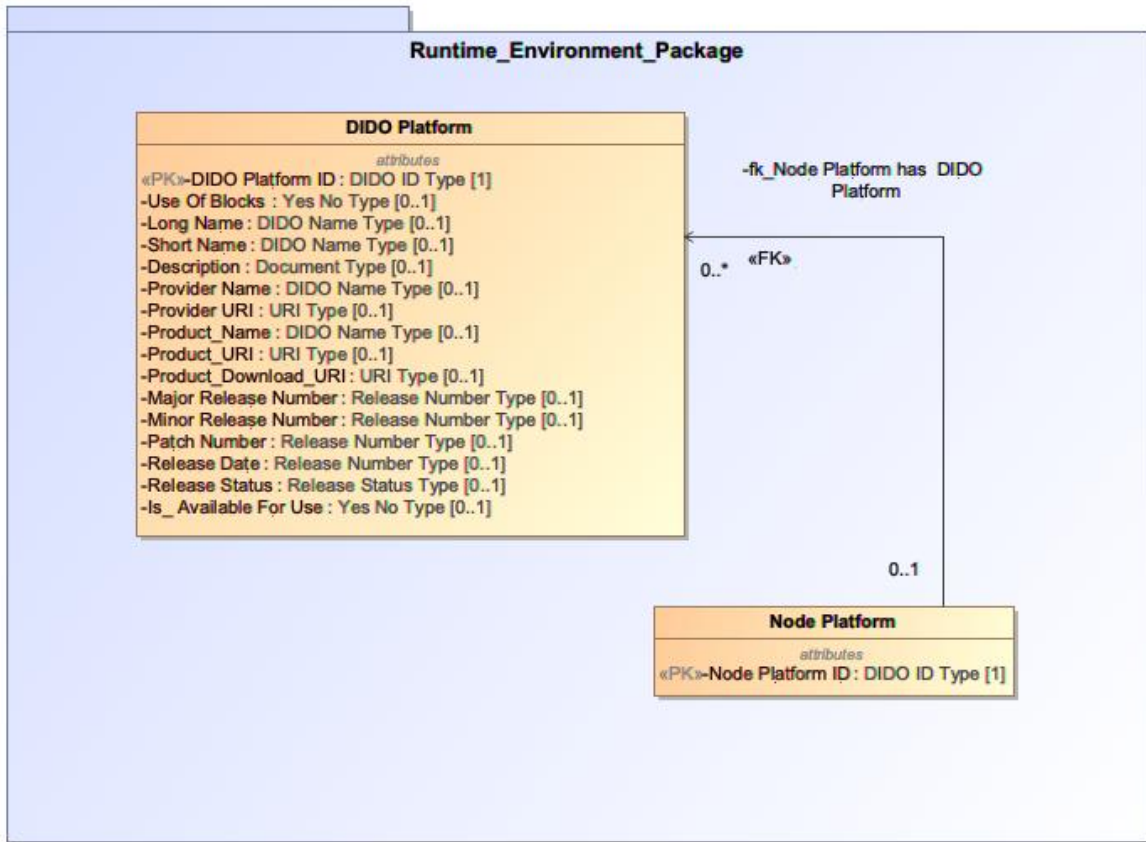


Figure 110. DIDO Platform

See:

- [Class DIDO Platform](#)
- [Class Node Platform](#)

Distributed Immutable Data Object Data Model

Dido-DM

Disk Image Repository

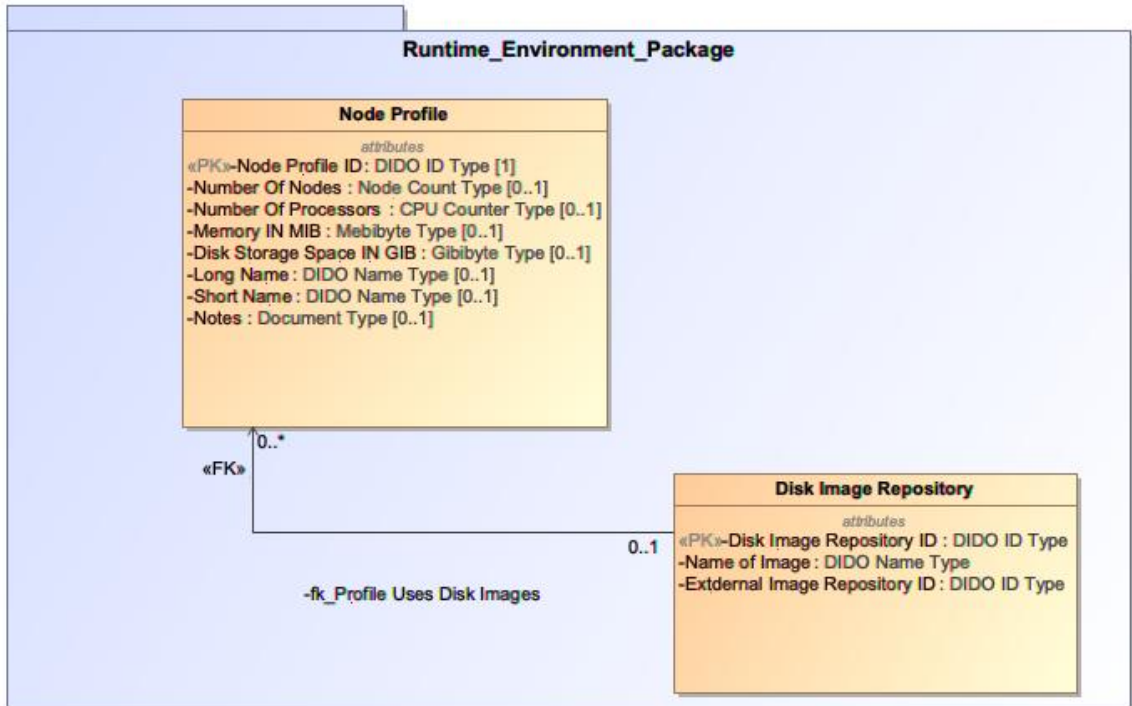


Figure 111. Disk Image Repository

See:

- [Class Disk Image Repository](#)
- [Class Node Profile](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain

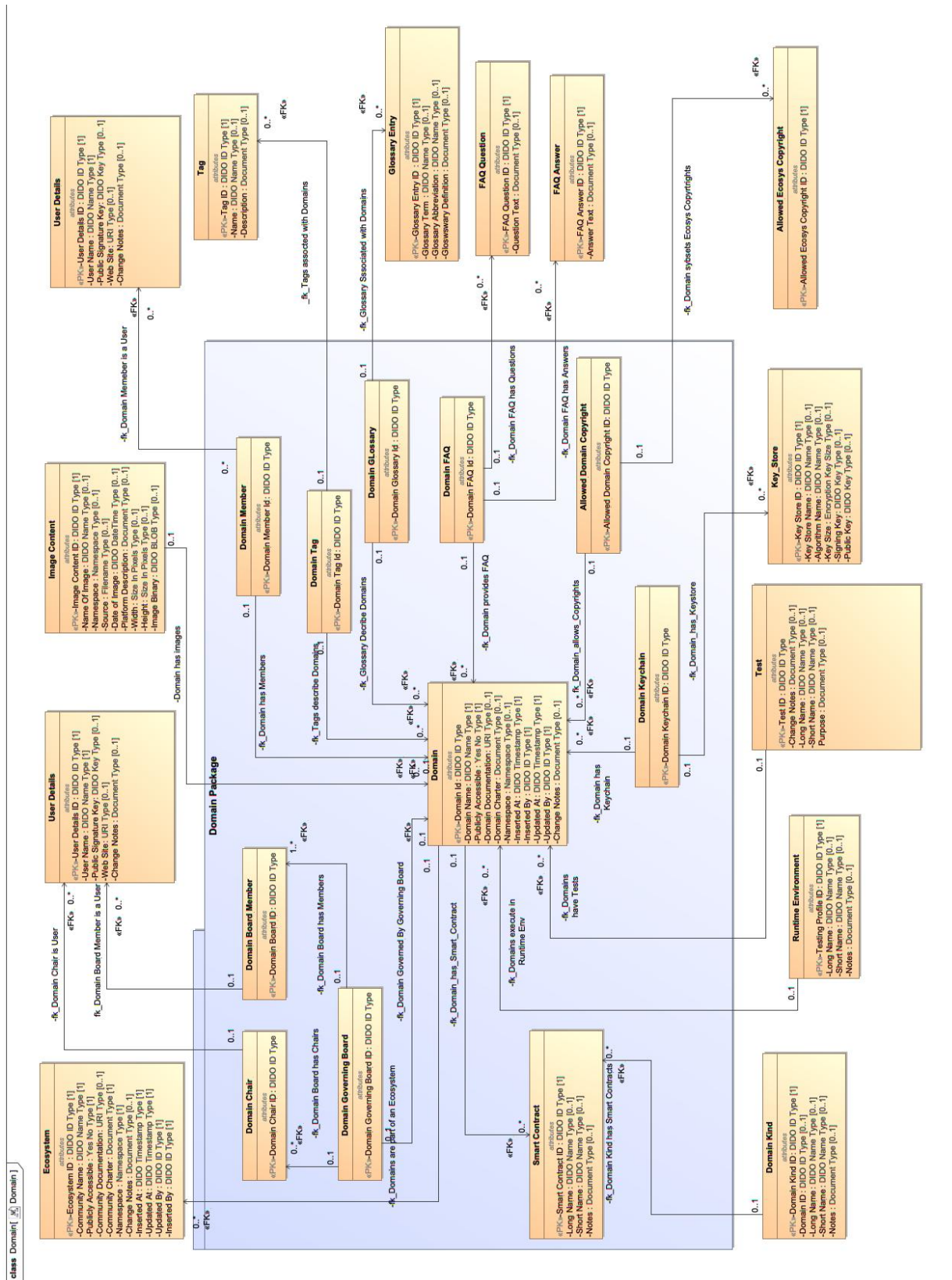


Figure 112. Domain

See:

- [Class Domain](#)
- [Class Allowed Domain Copyright](#)

Distributed Immutable Data Object Data Model

Dido-DM

- [Class Domain Board Member](#)
- [Class Domain Chair](#)
- [Class Domain FAQ](#)
- [Class Domain Glossary](#)
- [Class Domain Governing Board](#)
- [Class Domain Keychain](#)
- [Class Domain Kind](#)
- [Class Domain Member](#)
- [Class Domain Tag](#)
- [Class Smart Contract](#)
- [Class Ecosystem](#)
- [Class User Details](#)
- [Class Image Content](#)
- [Class Tag](#)
- [Class Glossary Entry](#)
- [Class FAQ Question](#)
- [Class FAQ Answer](#)
- [Class Allowed Ecosys Copyright](#)
- [Class Key Store](#)
- [Class Test](#)
- [Class Runtime Environment](#)
- [Class Domain Kind](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain Board Member

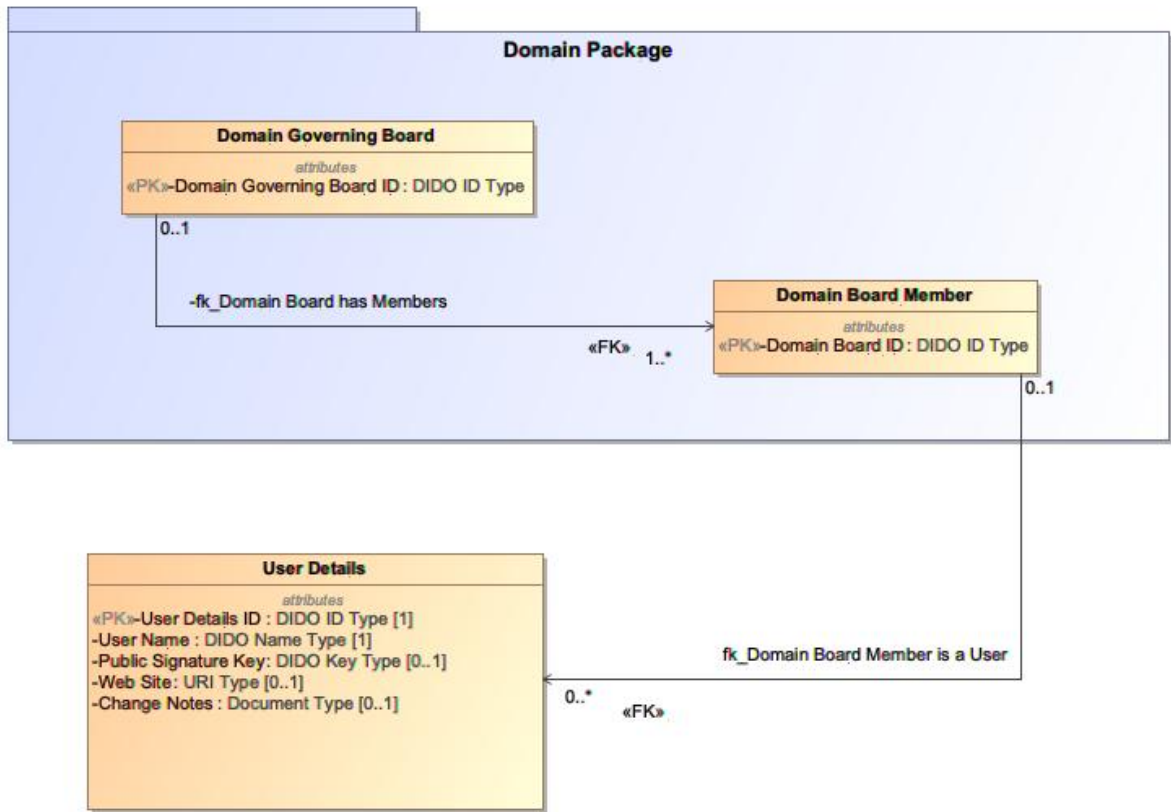


Figure 113. Domain Board Member

See:

- [Class Domain Board Member](#)
- [Class Domain Governing Board](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain Chair

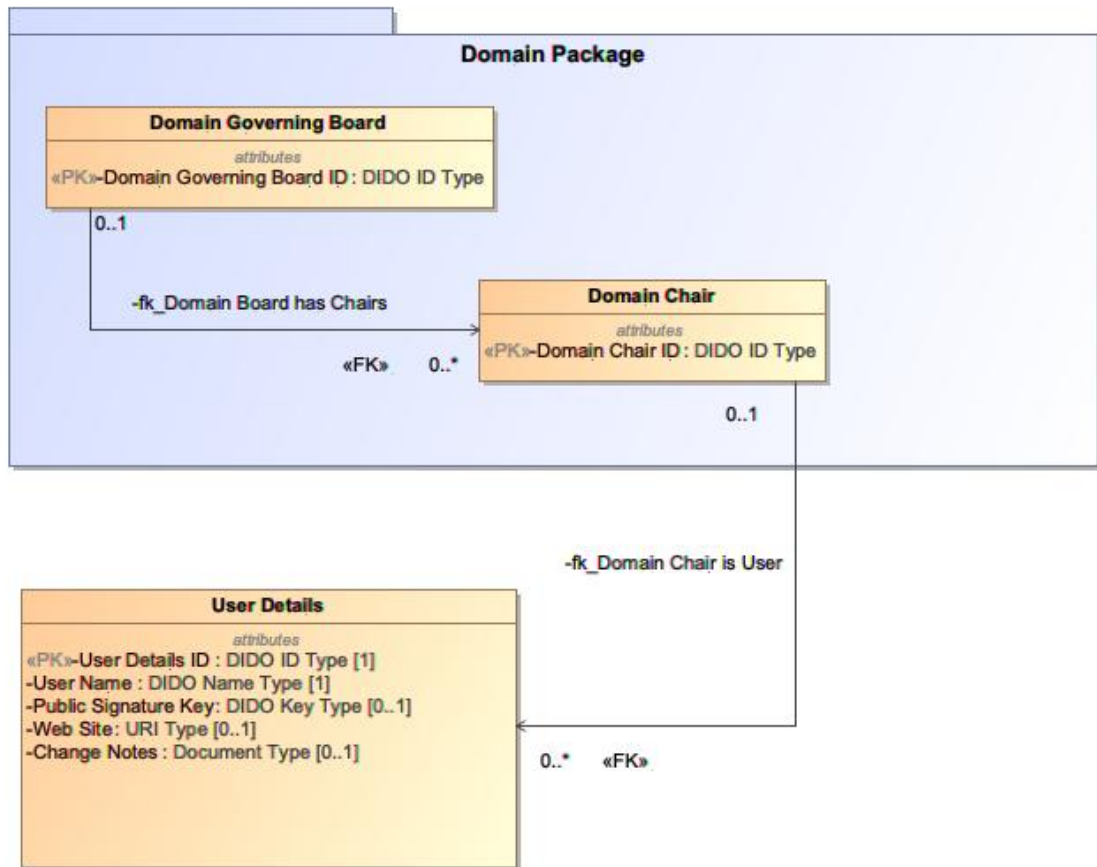


Figure 114. Domain Chair

See:

- [Class Domain Chair](#)
- [Class Domain Governing Board](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain FAQ

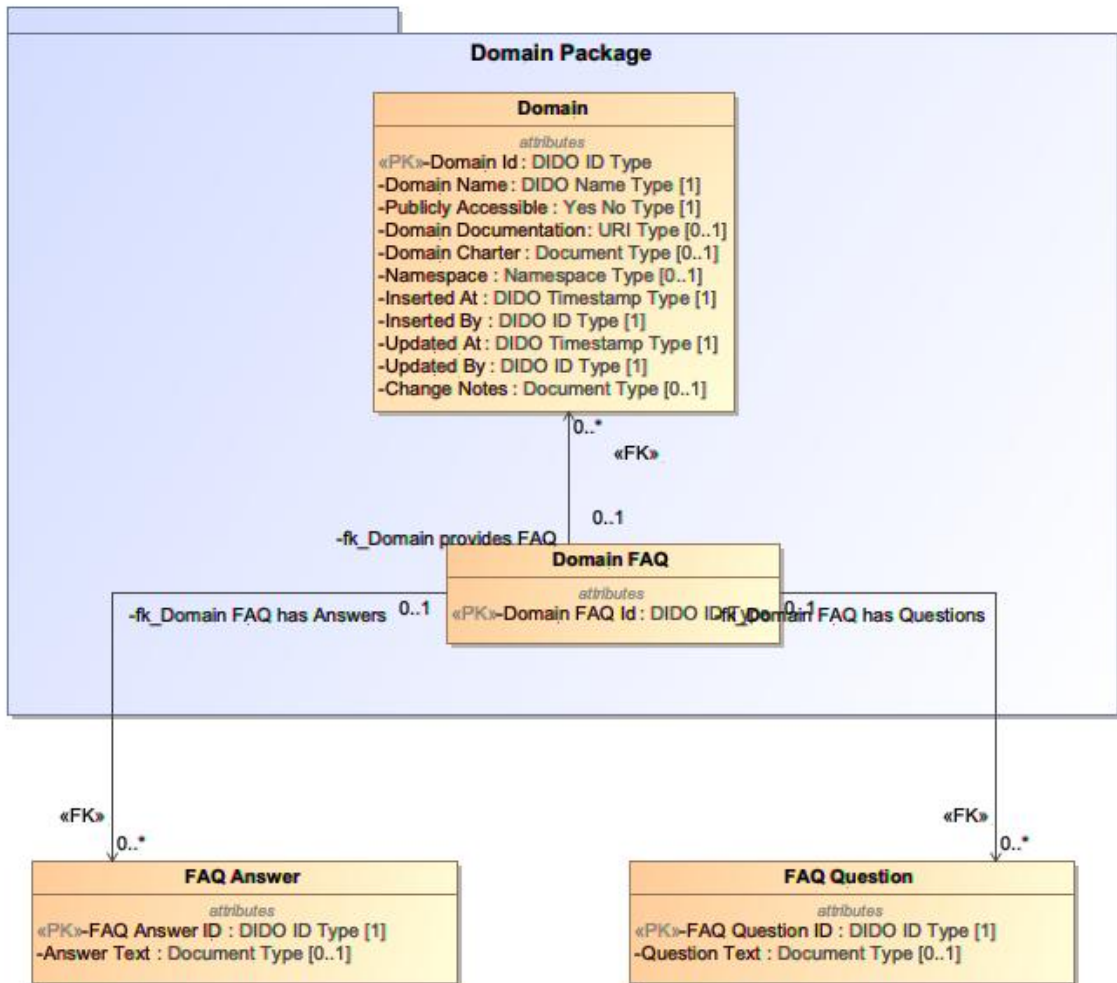


Figure 115. Domain FAQ

Ee:

- [Class Domain FAQ](#)
- [Class Domain](#)
- [Class FAQ Question](#)
- [Class FAQ Answer](#)

Distributed Immutable Data Object Data Model Dido-DM

Domain_Glossary

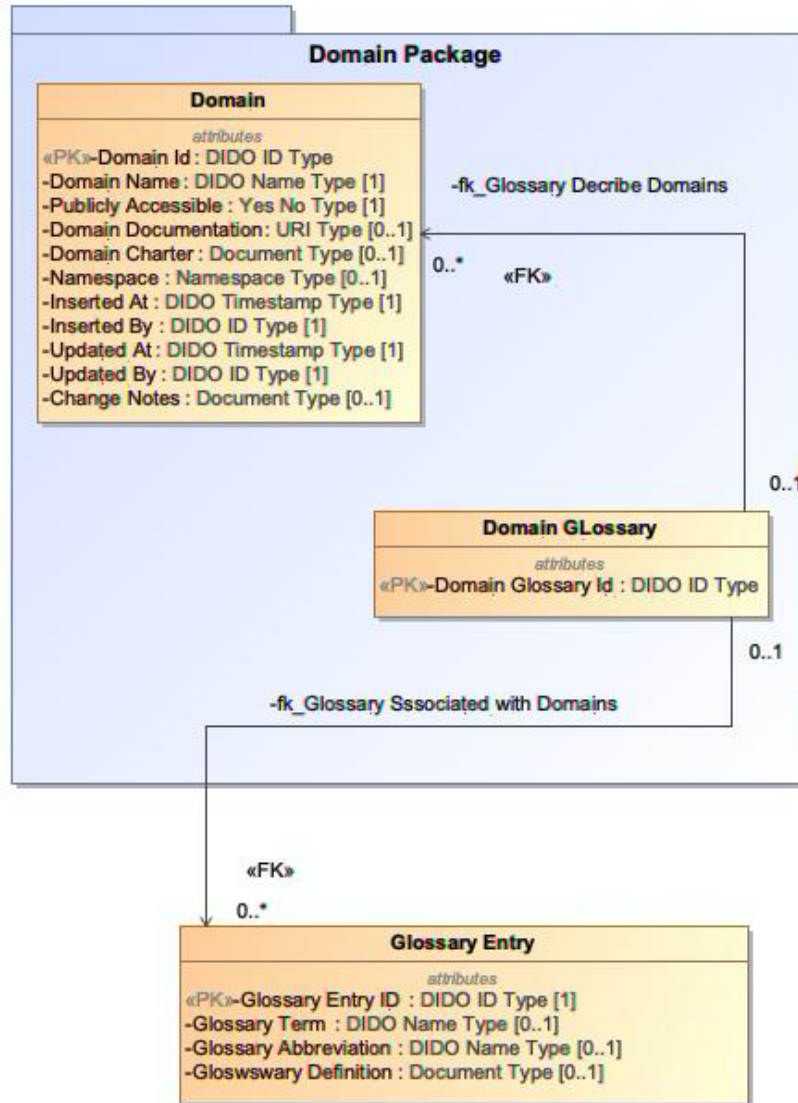


Figure 116. Domain_Glossary

See:

- [Class Domain_Glossary](#)
- [Class Domain](#)
- [Class Glossary Entry](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain Governing Board

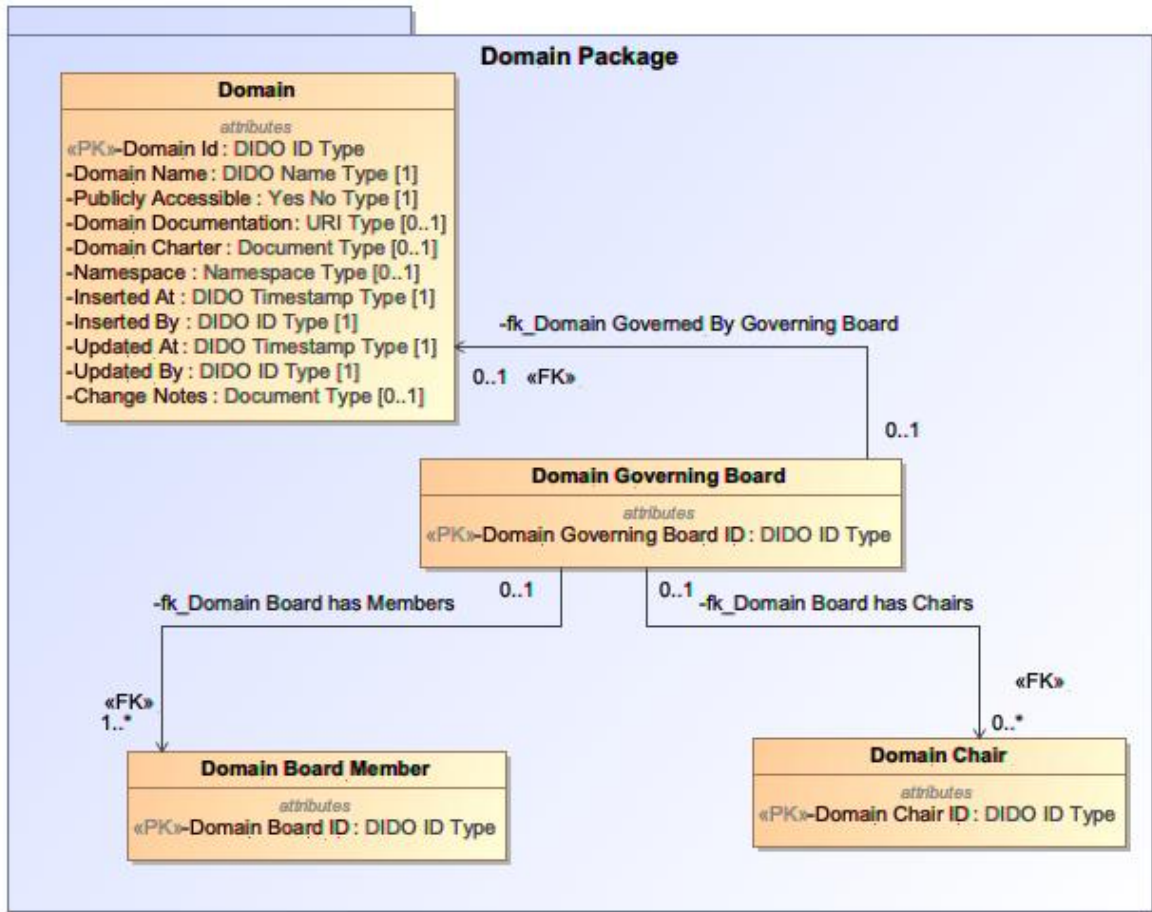


Figure 117. Domain Governing Board

See:

- [Class Domain Governing Board](#)
- [Class Domain](#)
- [Class Domain Chair](#)
- [Class Domain Board Member](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain Keychain

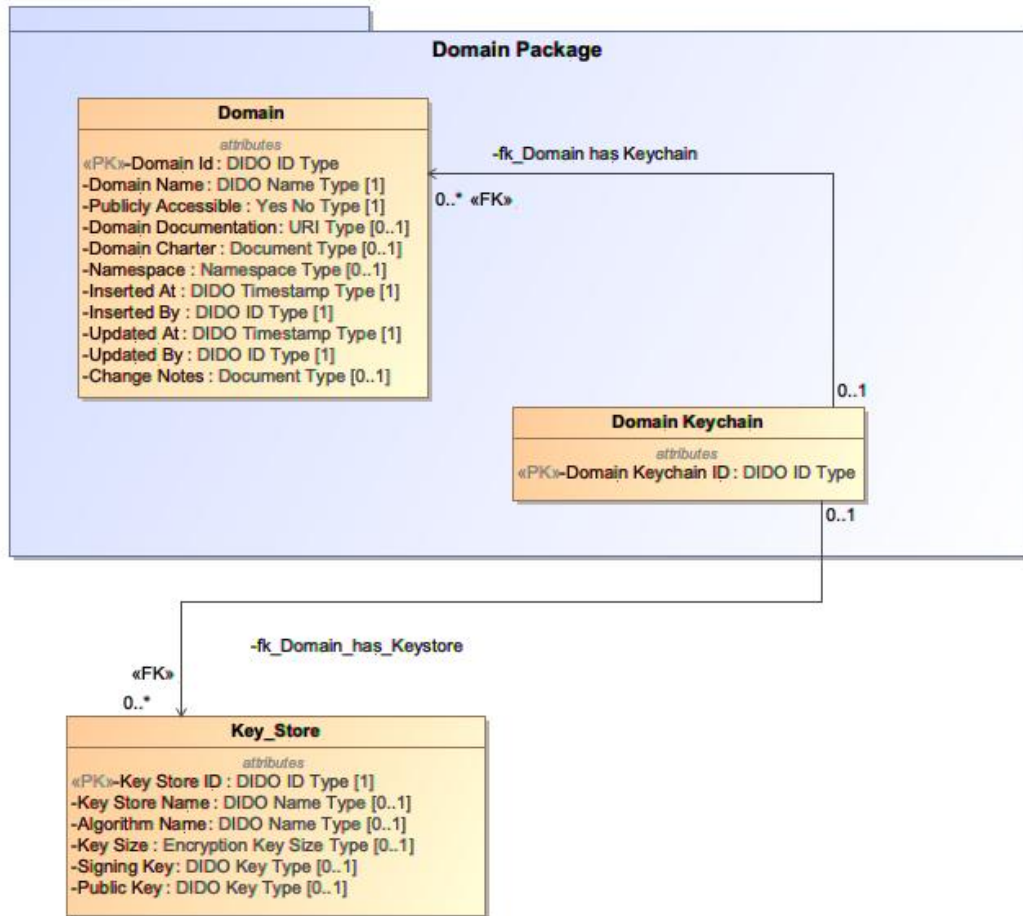


Figure 118. Domain Keychain

See:

- [Class Domain Keychain](#)
- [Class Domain](#)
- [Class Key_Store](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain Kind

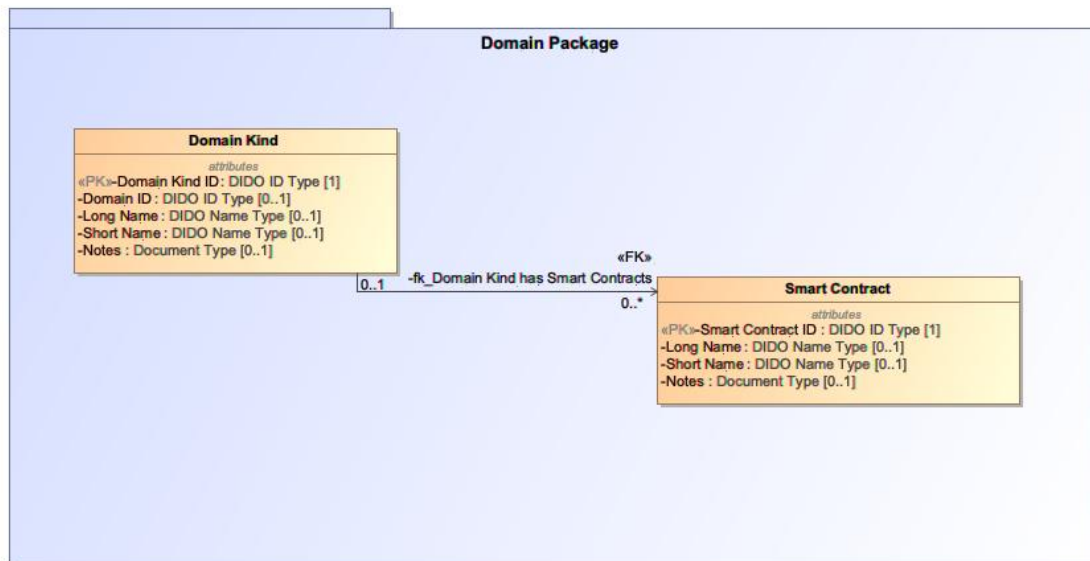


Figure 119. Domain Kind

- [Class Domain Kind](#)
- [Class Smart Contract](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain Member

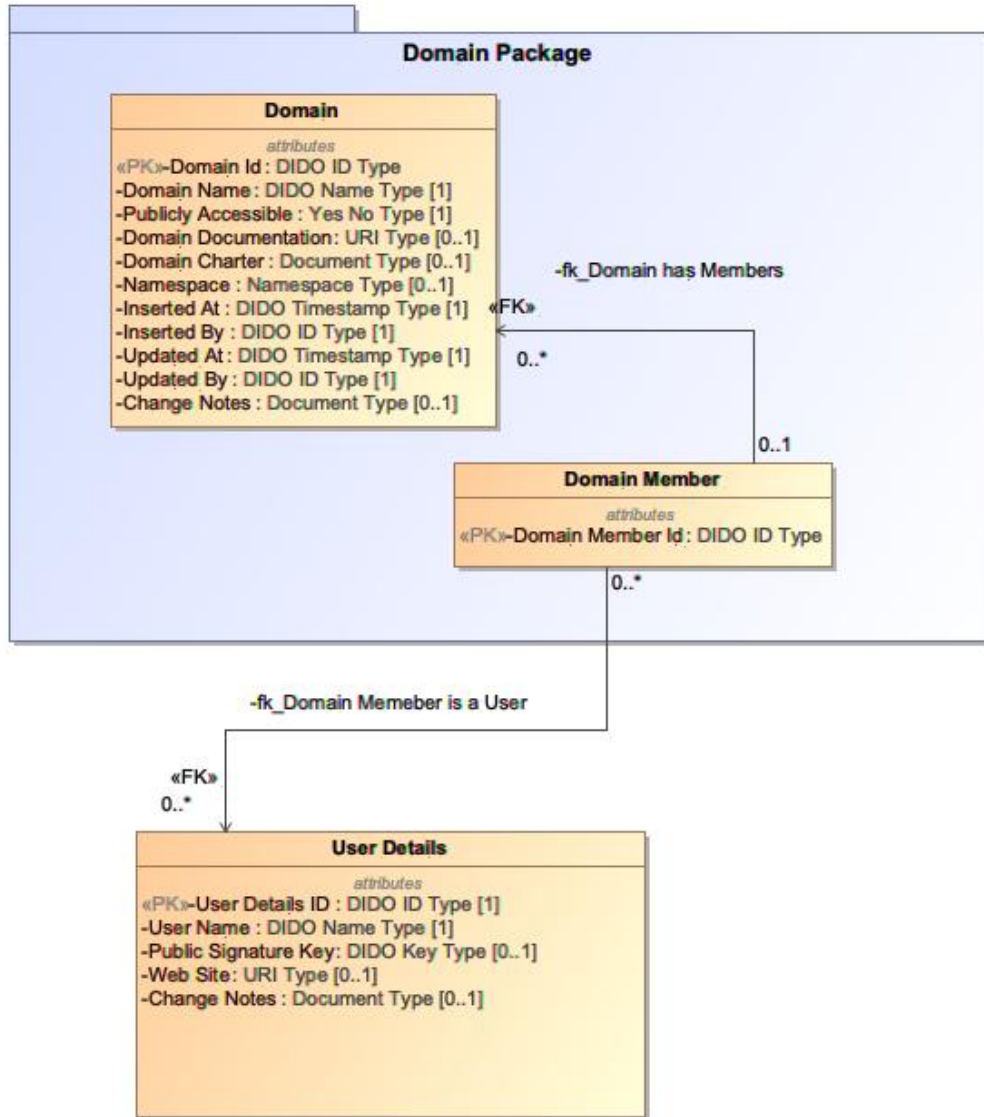


Figure 120. Domain Member

- [Class Domain Member](#)
- [Class Domain](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model

Dido-DM

Domain Tag

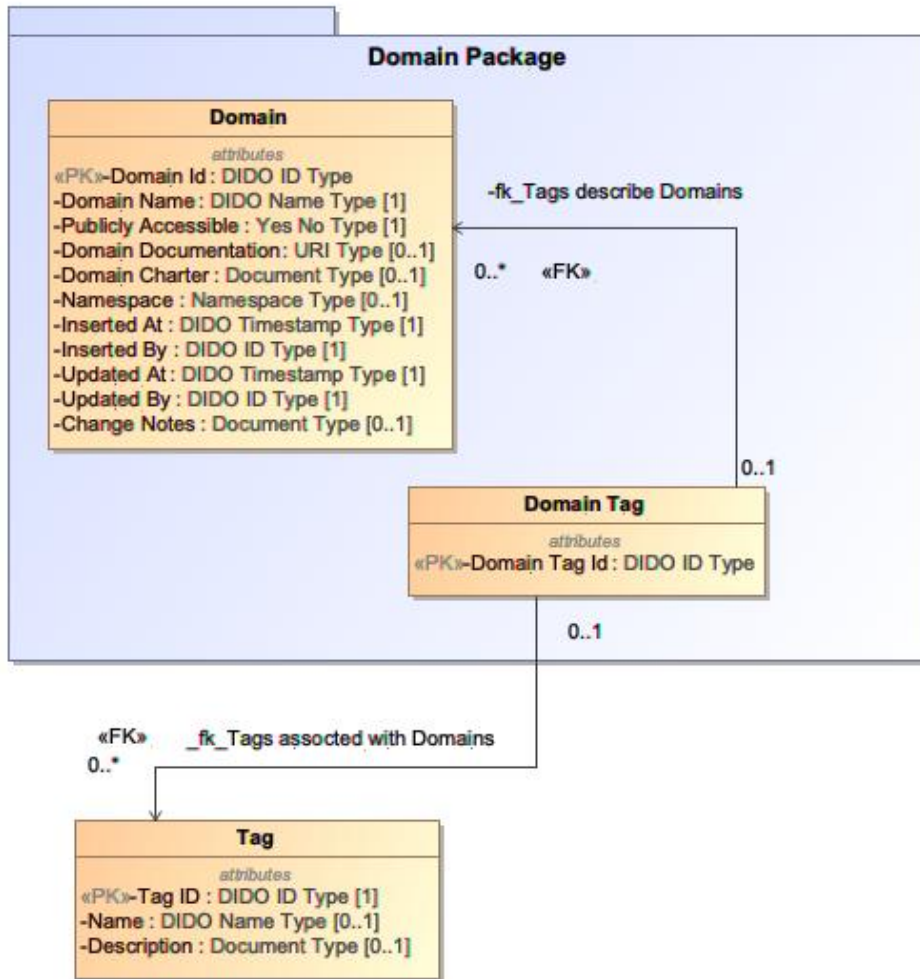


Figure 121. Domain Tag

- [Class Domain Tag](#)
- [Class Domain](#)
- [Class Tag](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosphere

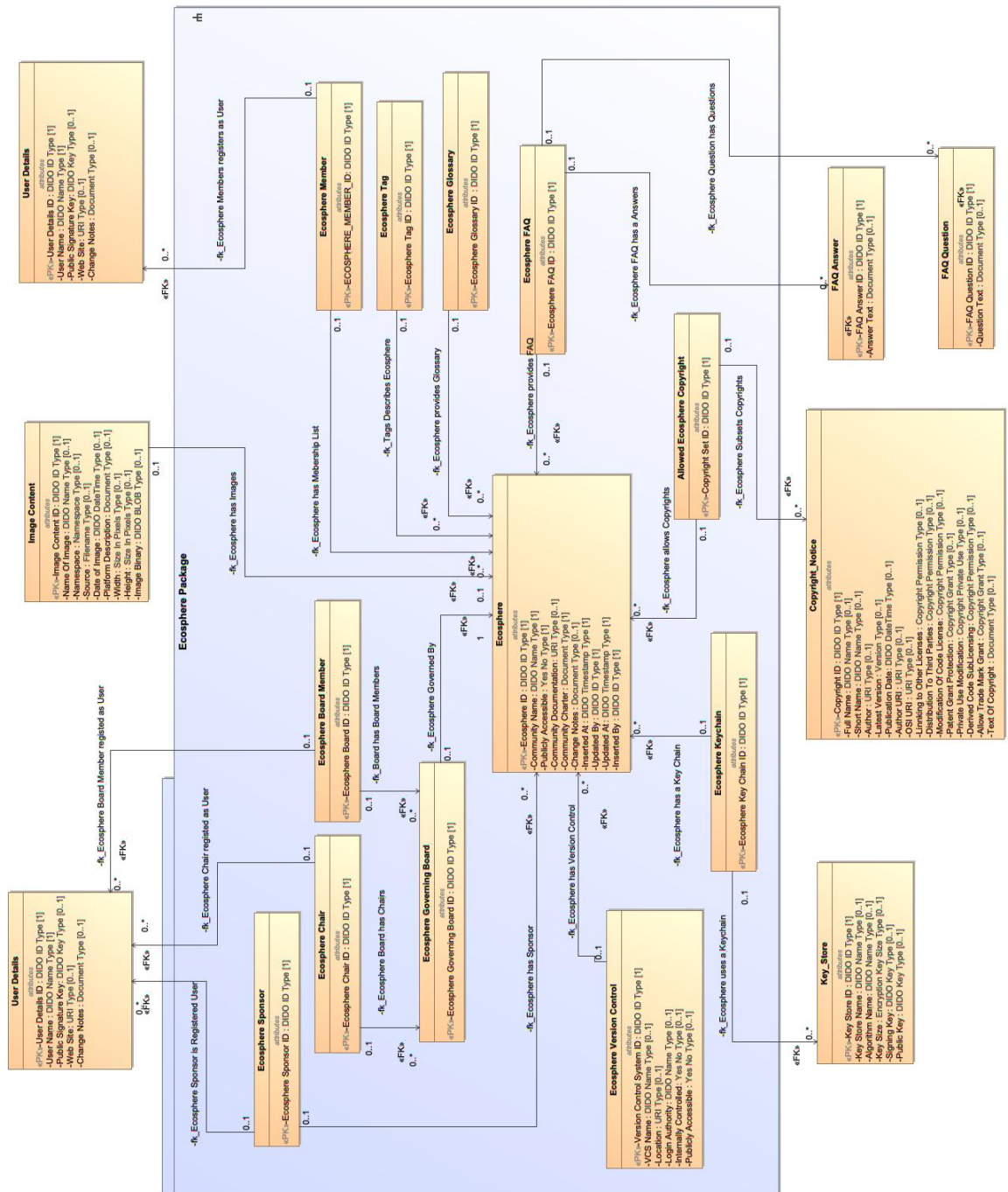


Figure 122. Ecosphere

See:

- [Class Ecosphere](#)
- [Class Allowed Ecosphere Copyright](#)
- [Class Ecosphere Board Member](#)
- [Class Ecosphere Chair](#)
- [Class Ecosphere FAQ](#)
- [Class Ecosphere Glossary](#)
- [Class Ecosphere Governing Board](#)
- [Class Ecosphere Keychain](#)

Distributed Immutable Data Object Data Model

Dido-DM

- [Class Ecosphere Member](#)
- [Class Ecosphere Sponsor](#)
- [Class Ecosphere Tag](#)
- [Class Ecosphere Version Control](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosphere Board Member

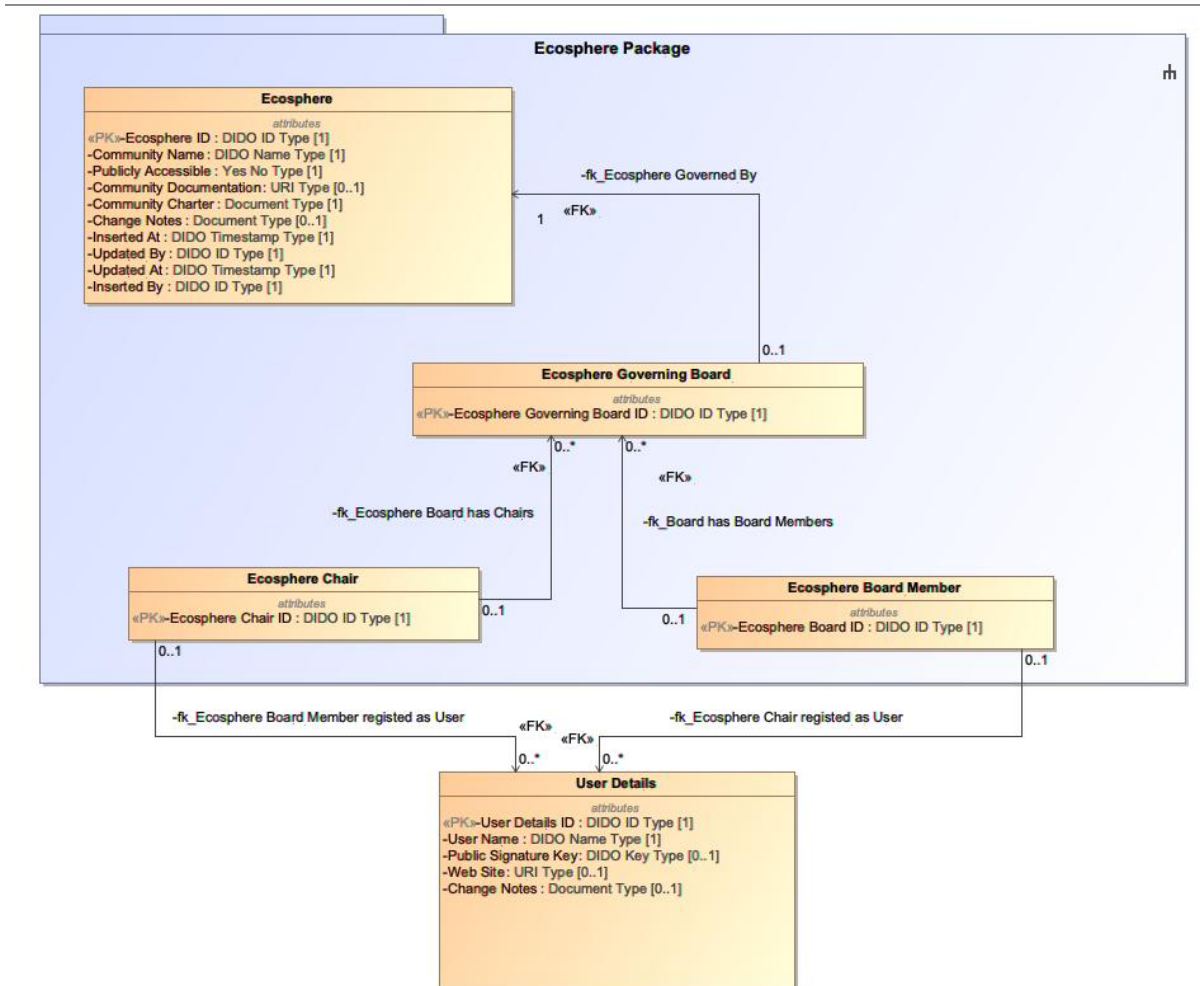


Figure 123. Ecosphere Board Member

See:

- [Class Ecosphere Governing Board](#)
- [Class Ecosphere Board Member](#)
- [Class Ecosphere](#)
- [Class Ecosphere Chair](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosphere Chair

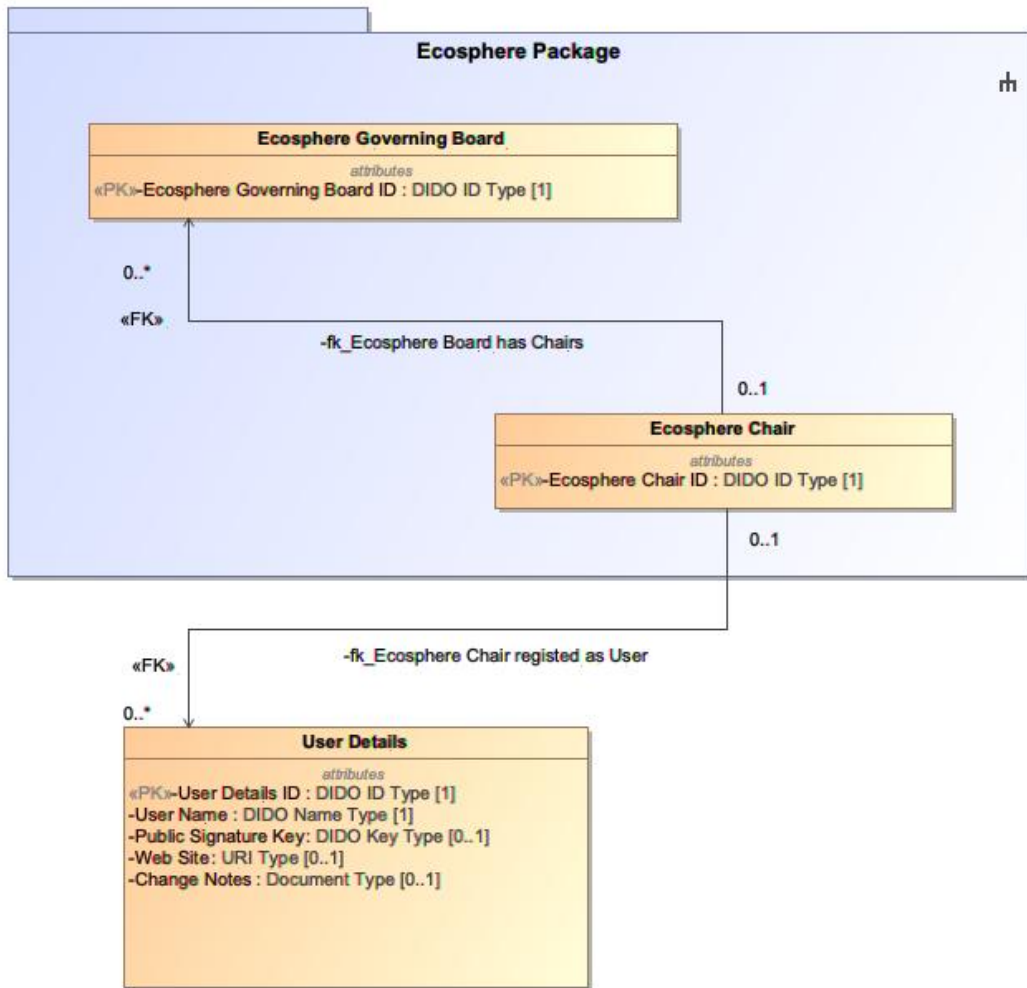


Figure 124. Ecosphere Chair

See:

- [Class Ecosphere Chair](#)
- [Class Ecosphere Governing Board](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosphere FAQ

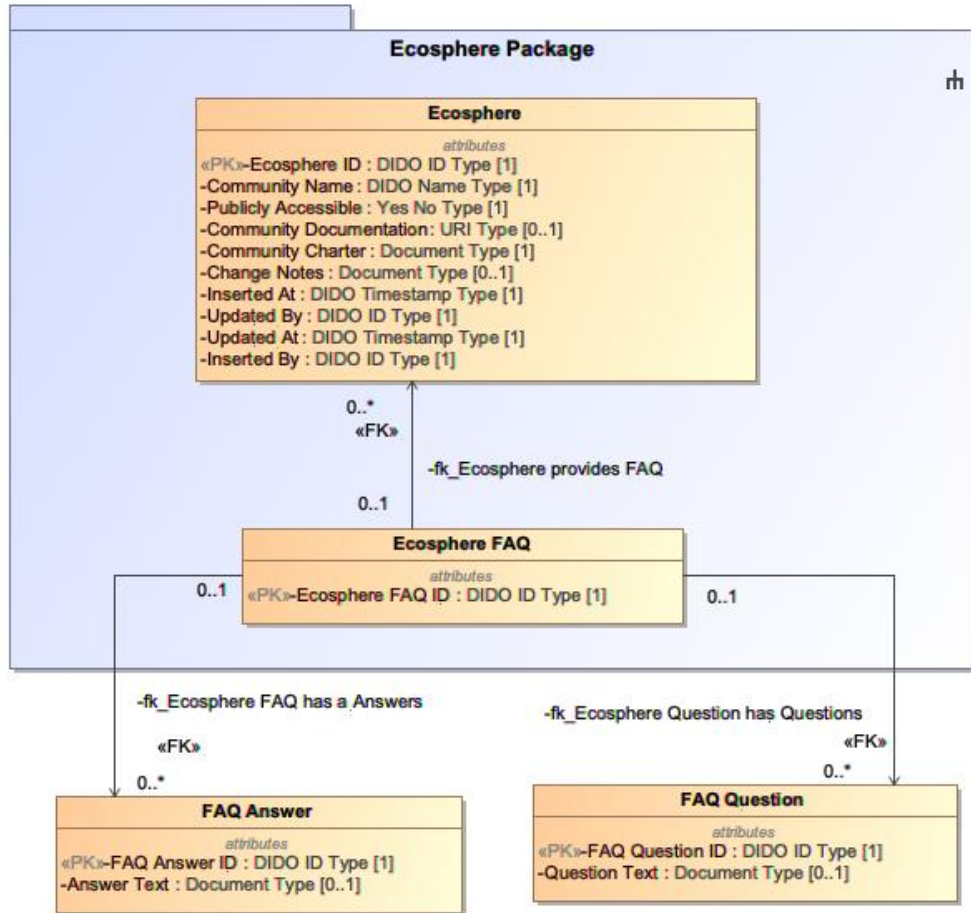


Figure 125. Ecosphere FAQ

See:

- [Class Ecosphere FAQ](#)
- [Class Ecosphere](#)
- [Class FAQ Question](#)
- [Class FAQ Answer](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosphere Glossary

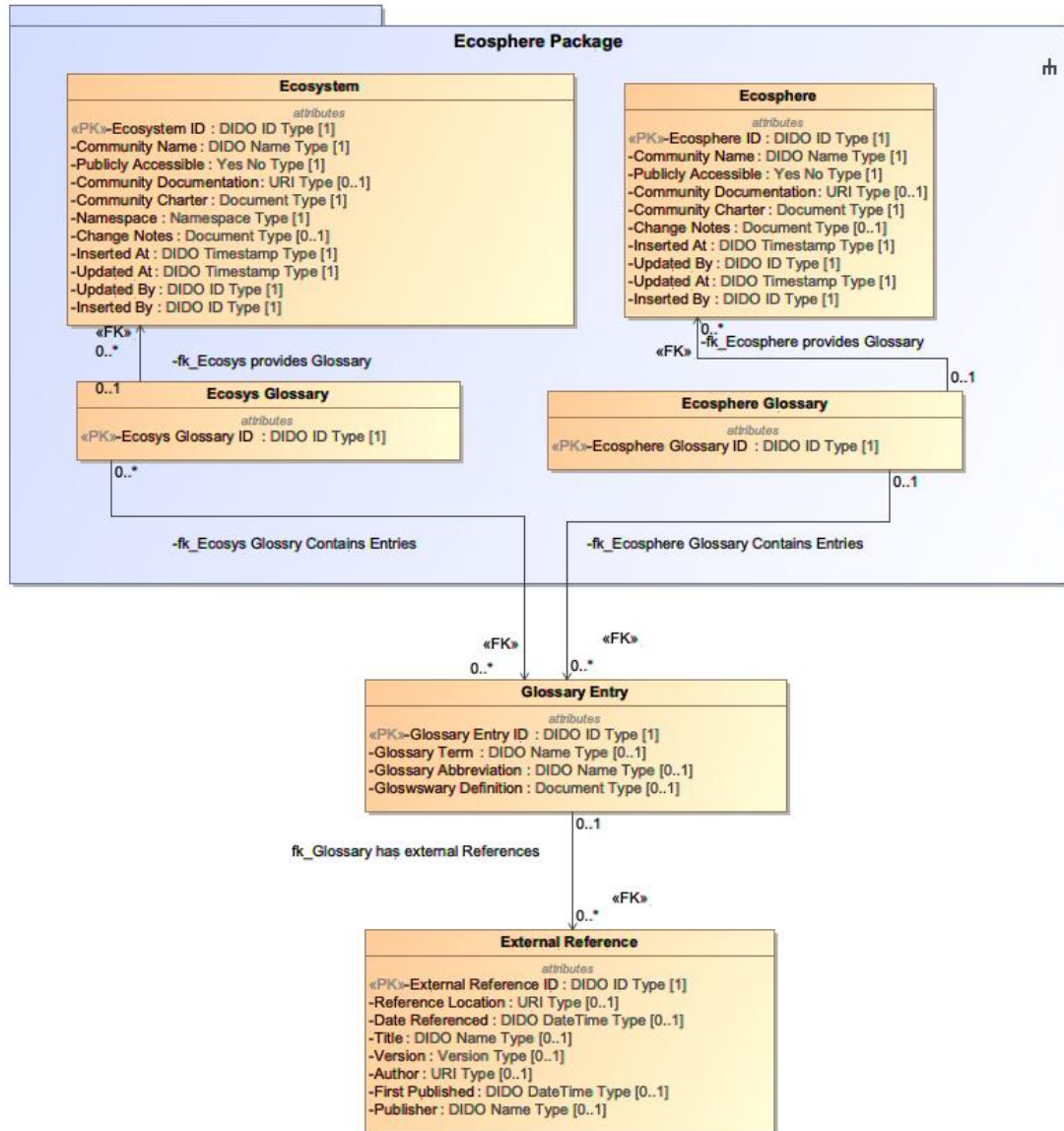


Figure 126. Ecosphere Glossary

See:

- [Class Ecosphere Glossary](#)
- [Class Ecosphere](#)
- [Class Ecosphere Glossary](#)
- [Class Ecosys Glossary](#)
- [Class Glossary Entry](#)
- [Class External Reference](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosphere Governing Board

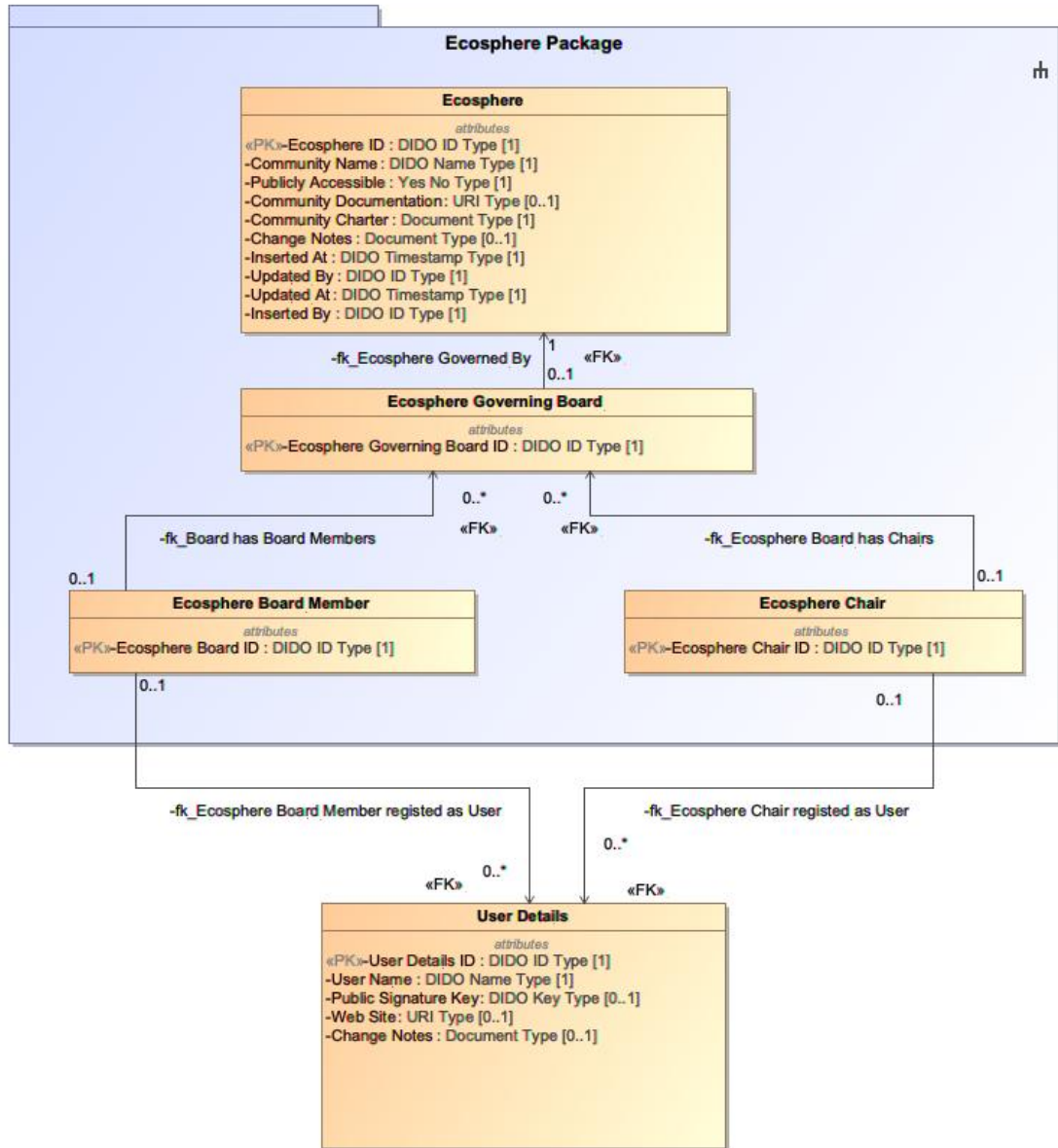


Figure 127. Ecosphere Governing Board

See:

- [Class Ecosphere Governing Board](#)
- [Class Ecosphere](#)
- [Class Ecosphere Board Member](#)
- [Class Ecosphere Chair](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model

Dido-DM

Ecosphere Keychain

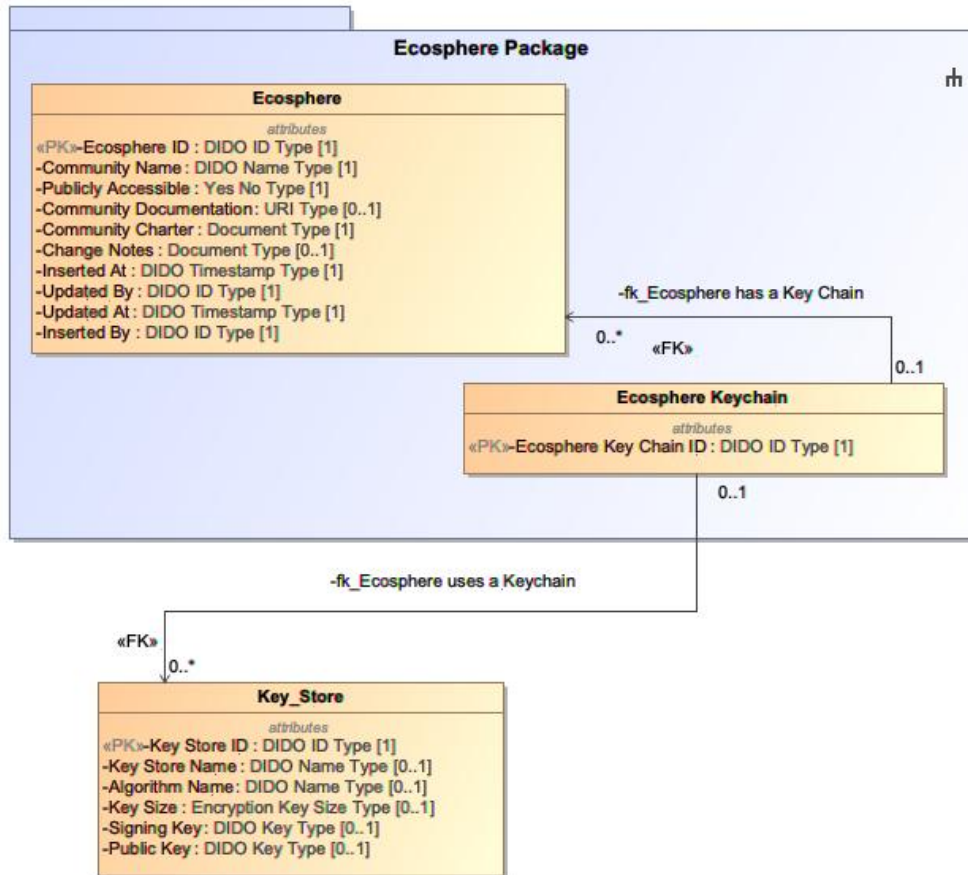


Figure 128. Ecosphere Keychain

See:

- [Class Ecosphere Keychain](#)
- [Class Ecosphere](#)
- [Class Key_Store](#)

Distributed Immutable Data Object Data Model

Dido-DM

Ecosphere Member

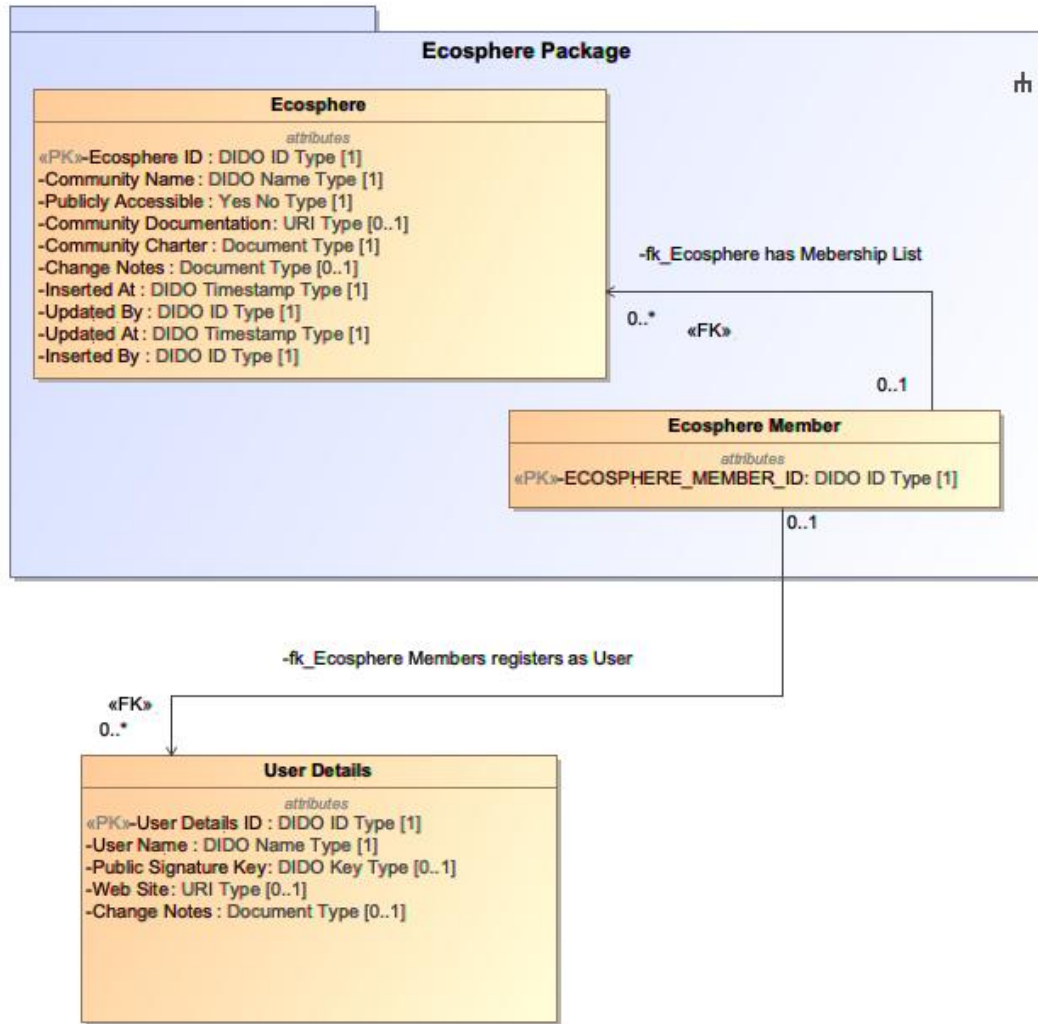


Figure 129. Ecosphere Member

See:

- [Class Ecosphere Member](#)
- [Class Ecosphere](#)
- [Class User Details](#)

Ecosphere Package

The Ecosphere Package contains the entities and the relationship between the entities that relate to an Ecosphere. The Ecosphere does not stand alone and is related to other packages with the Distributed Immutable Data Object (DIDO) Community of Interest (Col).

Distributed Immutable Data Object Data Model

Dido-DM



Figure 130. Ecosphere Package

Distributed Immutable Data Object Data Model

Dido-DM

Ecosphere Sponsor

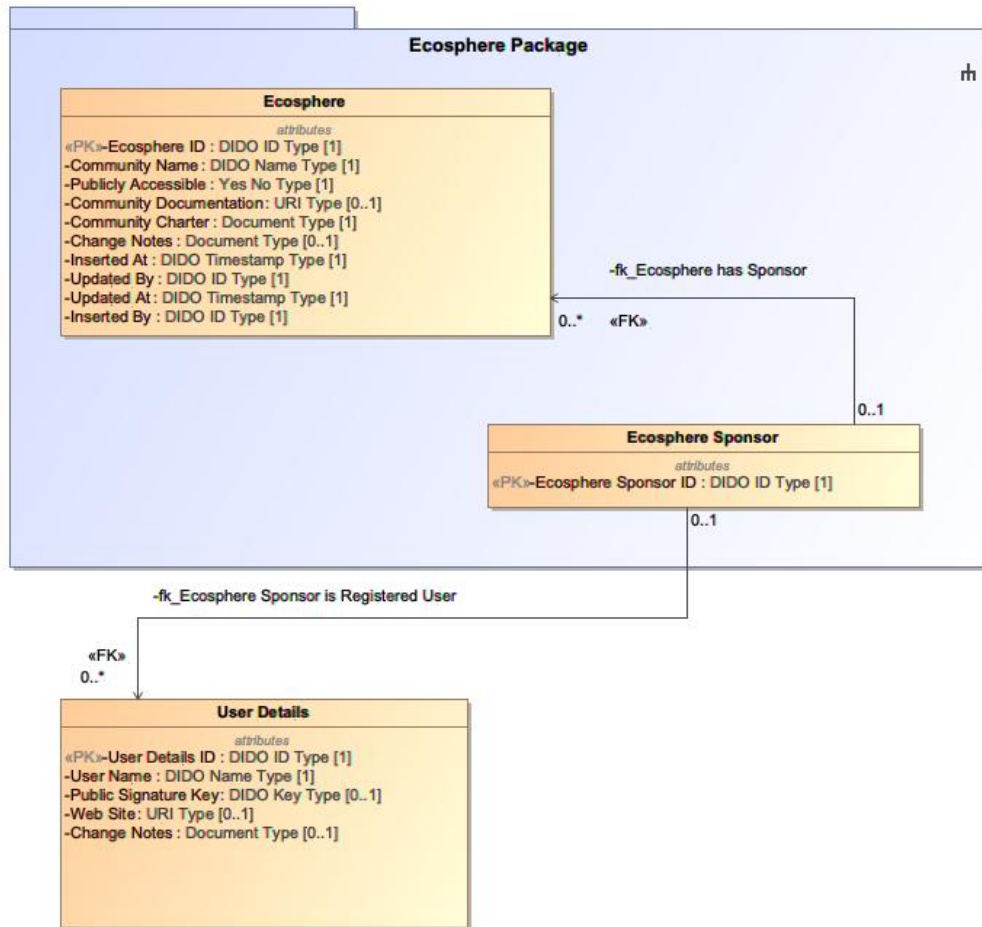


Figure 131. Ecosphere Sponsor

See:

- [Class Ecosphere Sponsor](#)
- [Class Ecosphere](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model

Dido-DM

Ecosphere Tag

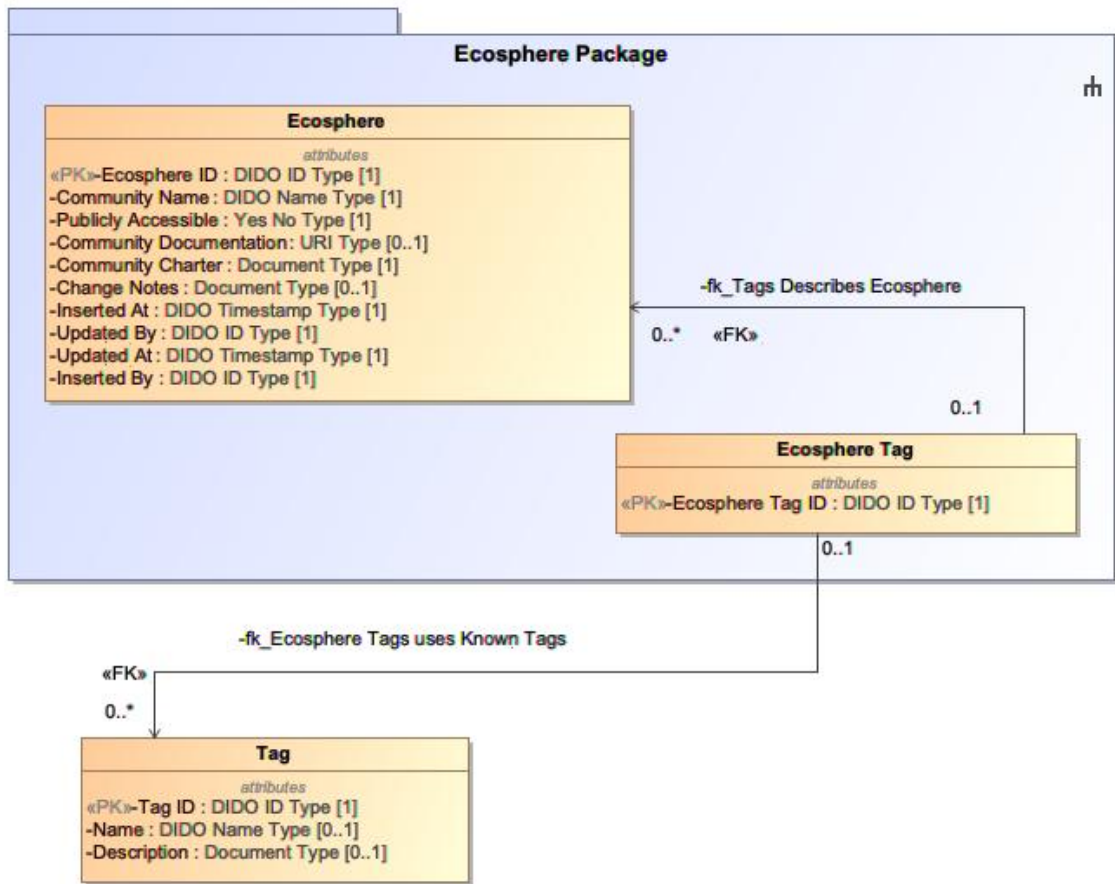


Figure 132. Ecosphere Tag

See:

- [Class Ecosphere Tag](#)
- [Class Ecosphere](#)
- [Class Tag](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosphere Version Control

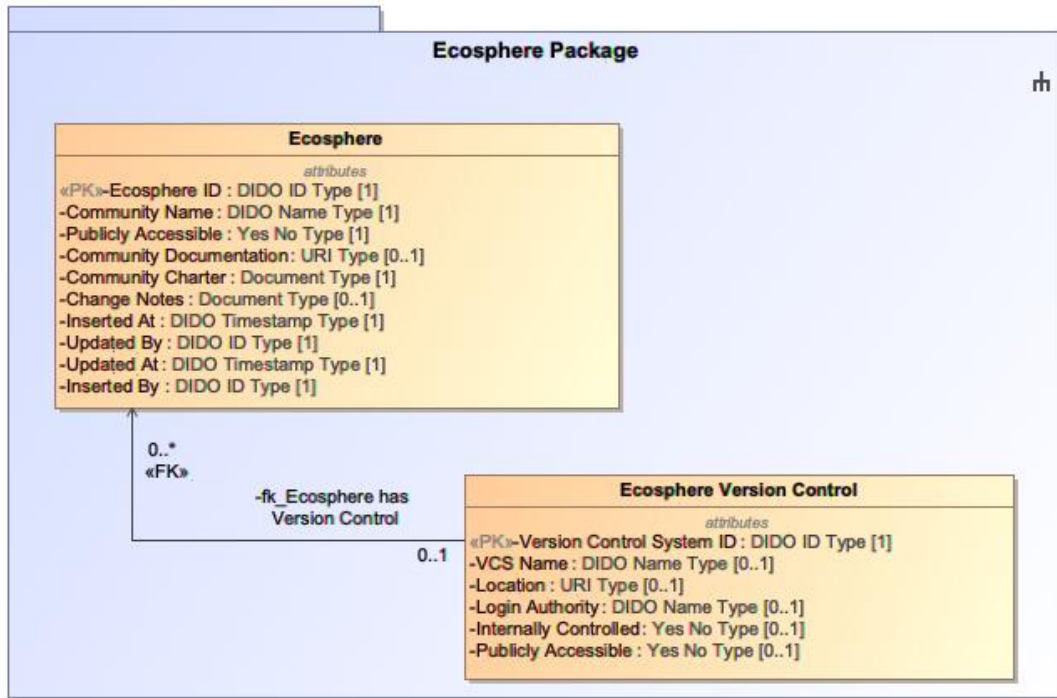


Figure 133. Ecosphere Version Control

See:

- [Class Ecosphere Version Control](#)
- [Class Ecosphere](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosys Board Member

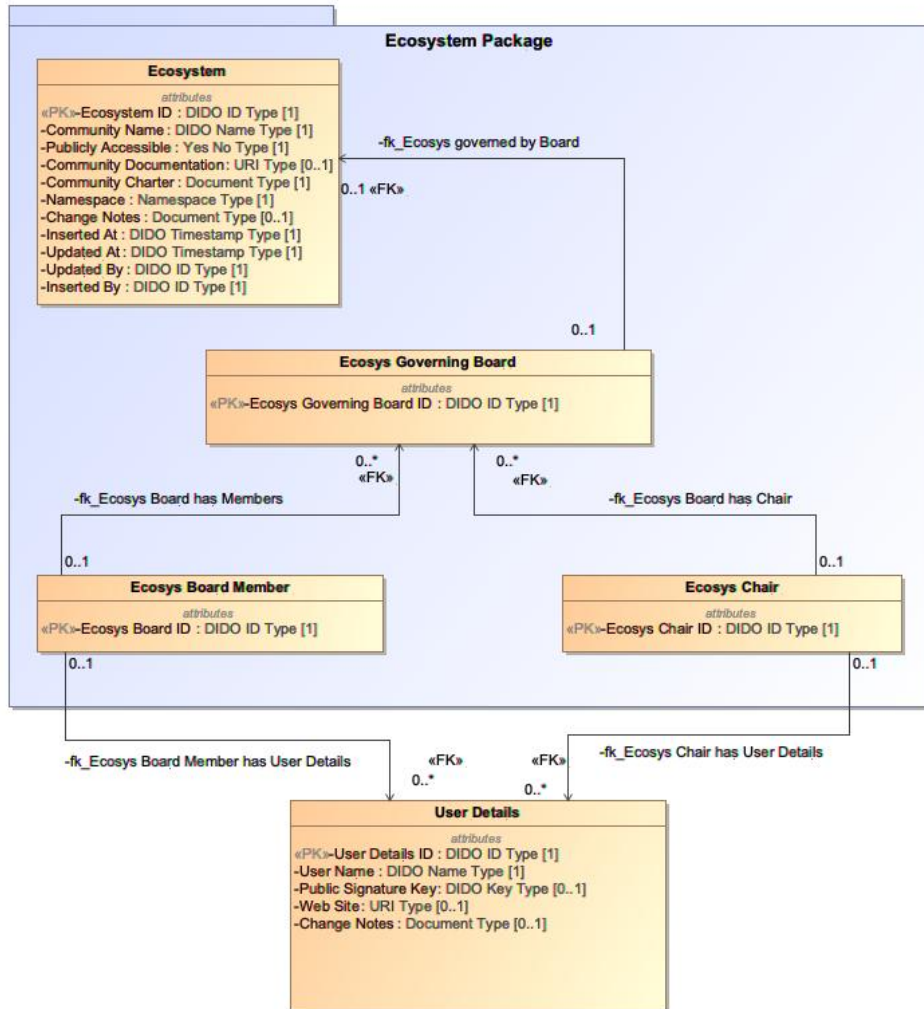


Figure 134. Ecosys Board Member

See:

- [Class Ecosys Board Member](#)
- [Class Ecosystem](#)
- [Class Ecosys Board Member](#)
- [Class Ecosys Chair](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosys Chair

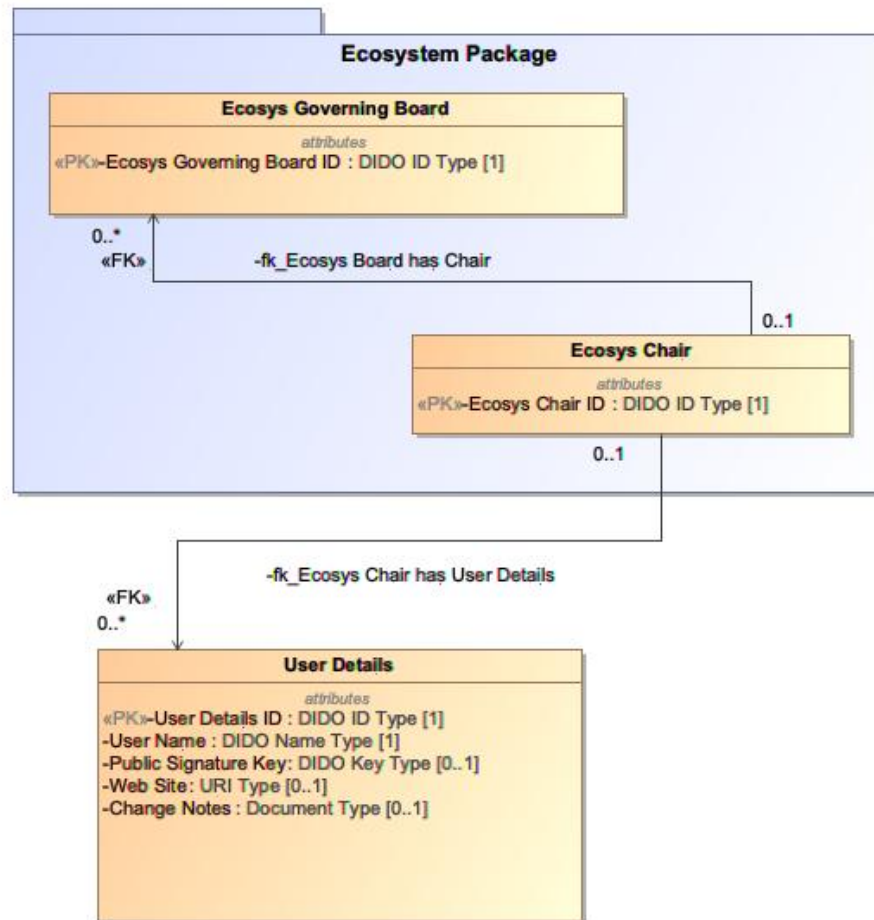


Figure 135. Ecosys Chair

See:

- [Class Ecosys Chair](#)
- [Class Ecosys Governing Board](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model

Dido-DM

Ecosys FAQ

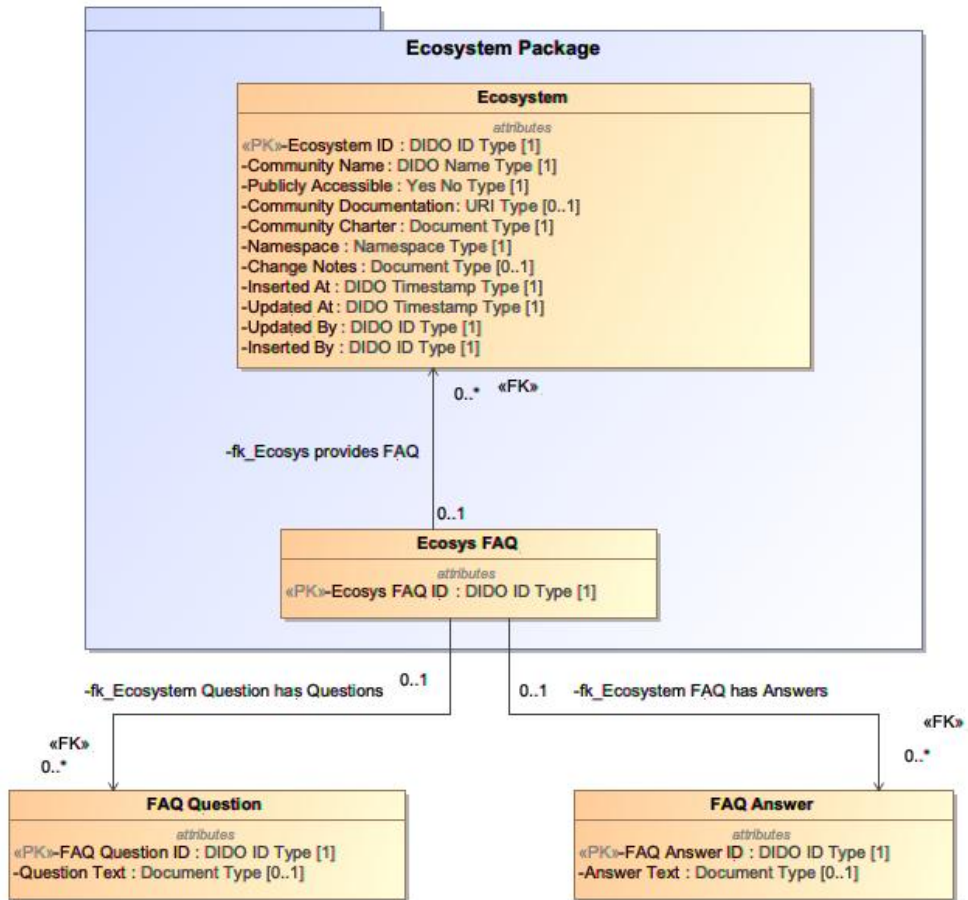


Figure 136. Ecosys FAQ

See:

- [Class Ecosys FAQ](#)
- [Class Ecosystem](#)
- [Class FAQ Question](#)
- [Class FAQ Answer](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosys Glossary

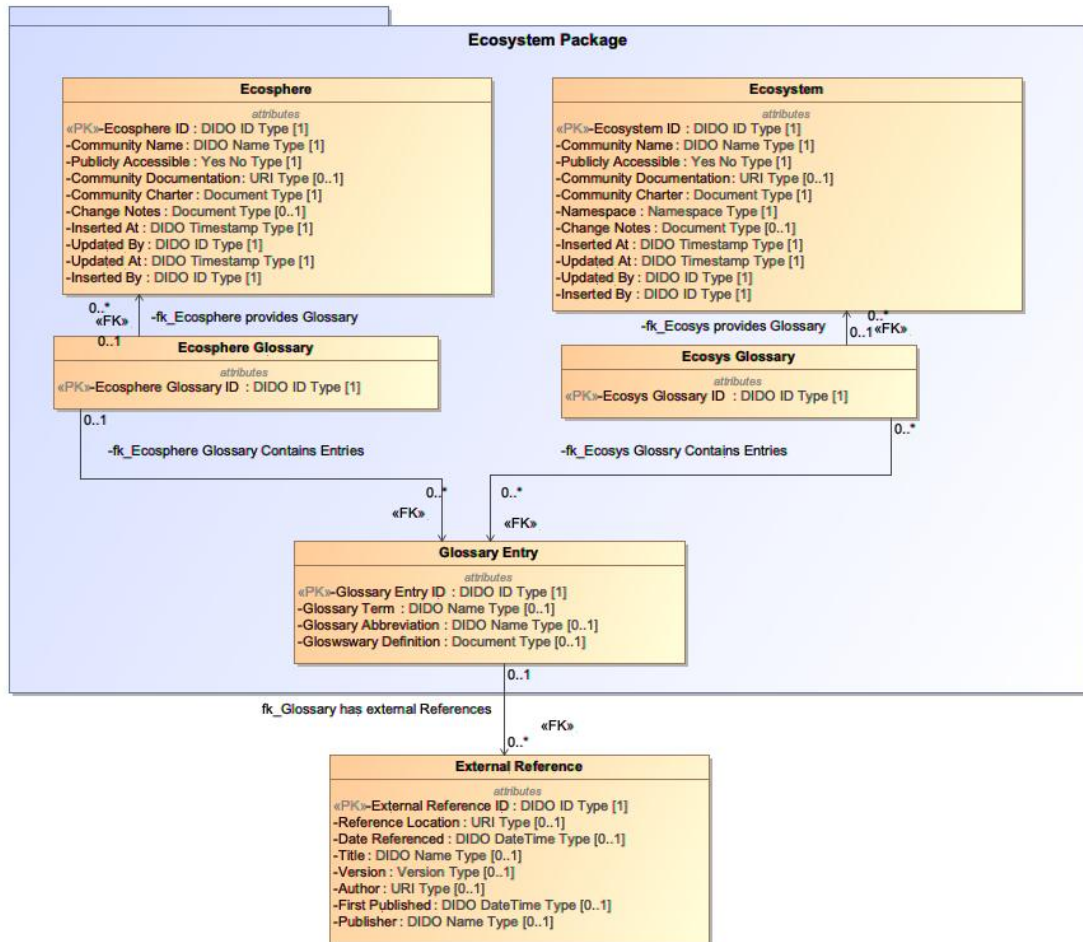


Figure 137. Ecosys Glossary

See:

- [Class Ecosys Glossary](#)
- [Class Ecosystem](#)
- [Class Ecosphere](#)
- [Class Ecosys Glossary](#)
- [Class Ecosphere Glossary](#)
- [Class Glossary Entry](#)
- [Class External Reference](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosys Governing Board

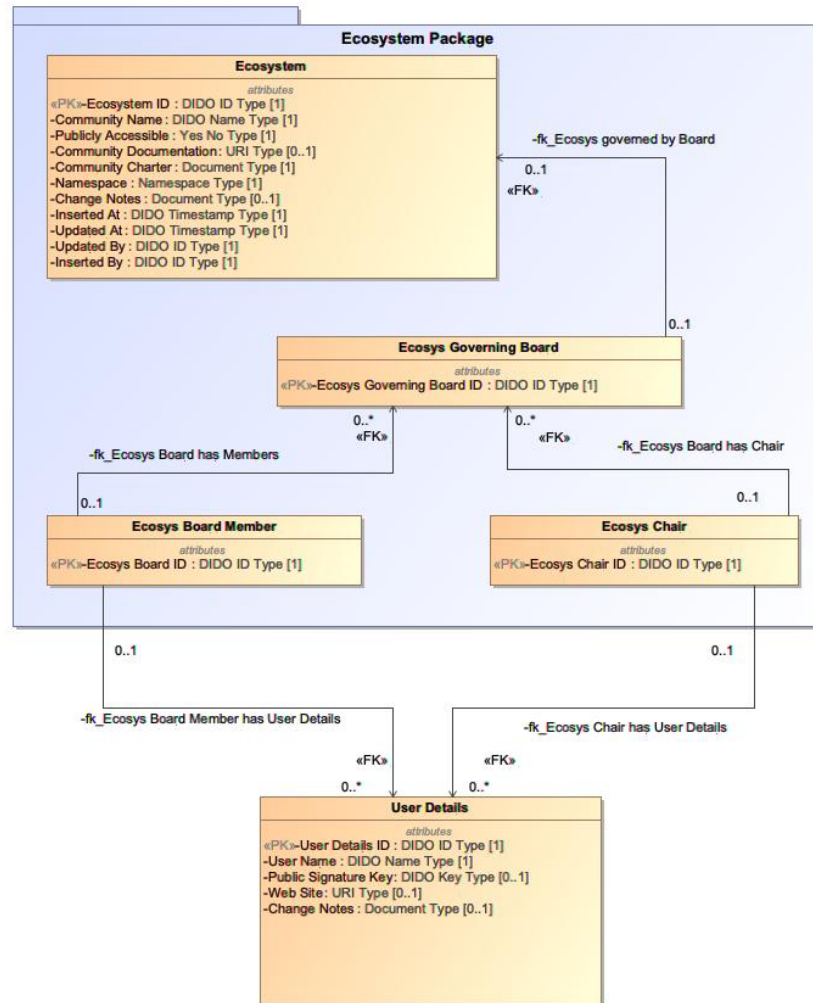


Figure 138. Ecosys Governing Board

See:

- [Class Ecosys Governing Board](#)
- [Class Ecosystem](#)
- [Class Ecosys Board Member](#)
- [Class Ecosys Chair](#)
- [Class User Details](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosys Keychain

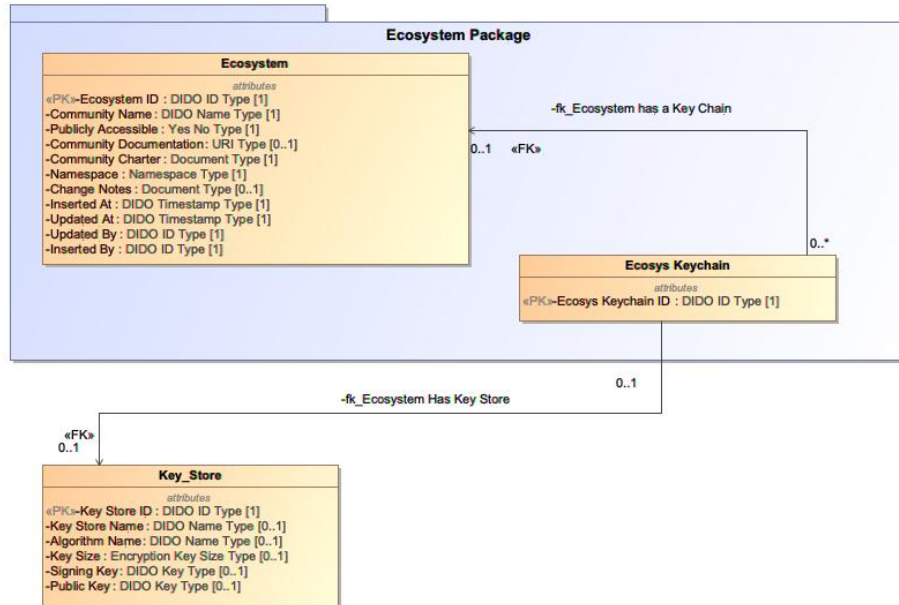


Figure 139. Ecosys Keychain

See:

- [Class Ecosys Keychain](#)
- [Class Ecosystem](#)
- [Class Key_Store](#)

Distributed Immutable Data Object Data Model

Dido-DM

Ecosys Member

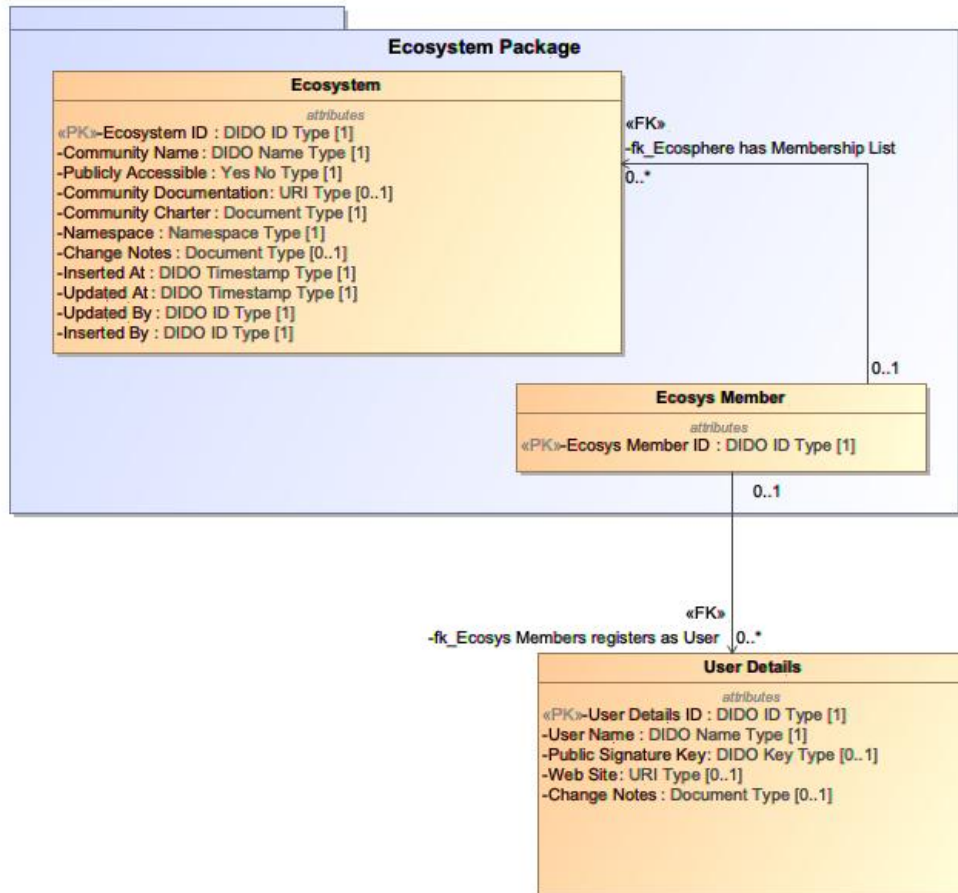


Figure 140. Ecosys Member

See:

- [Class Ecosys Member](#)
- [Class Ecosystem](#)
- [Package DIDO - Conceptual User_Details_Package](#)

Distributed Immutable Data Object Data Model

Dido-DM

Ecosys Tag

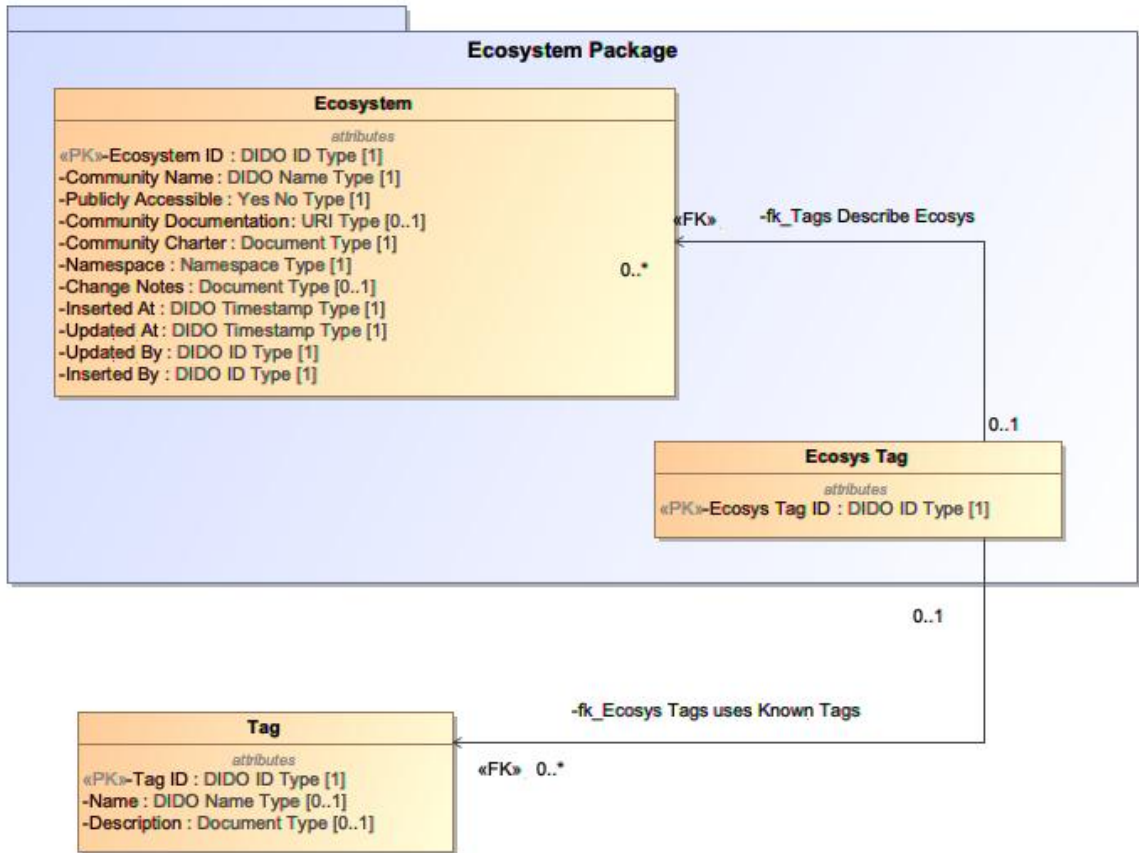


Figure 141. Ecosys Tag

See:

- [Class Ecosys Tag](#)
- [Class Ecosystem](#)
- [Class Tag](#)

Distributed Immutable Data Object Data Model Dido-DM

Ecosystem

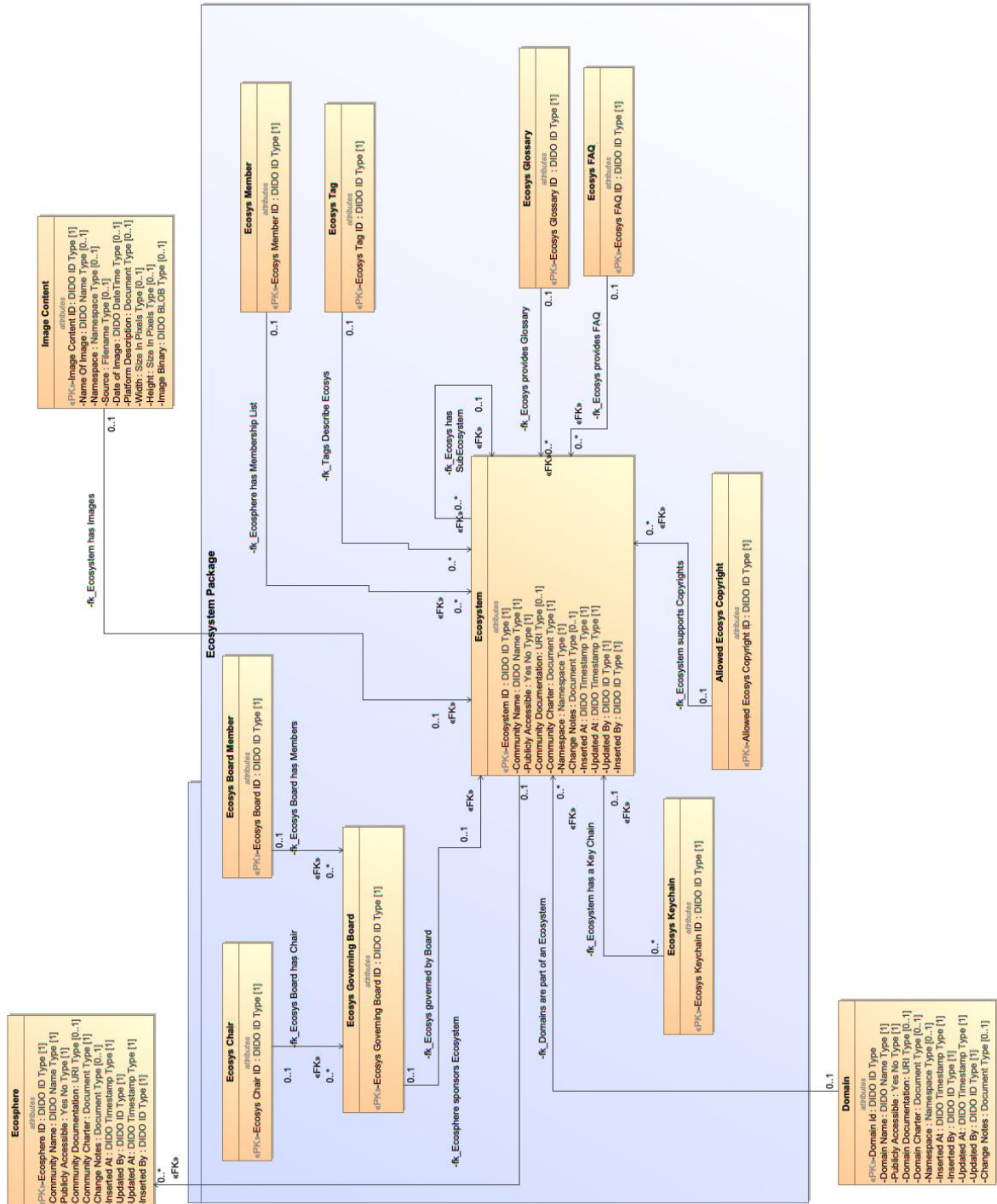


Figure 142. Ecosystem

See:

- [Class Ecosystem](#)
- [Class Allowed Ecosys Copyright](#)
- [Class Ecosys Board Member](#)
- [Class Ecosys Chair](#)
- [Class Ecosys FAQ](#)
- [Class Ecosys Glossary](#)

Distributed Immutable Data Object Data Model

Dido-DM

- [Class Ecosys Governing Board](#)
- [Class Ecosys Keychain](#)
- [Class Ecosys Member](#)
- [Class Ecosys Tag](#)

Distributed Immutable Data Object Data Model

Dido-DM

Email

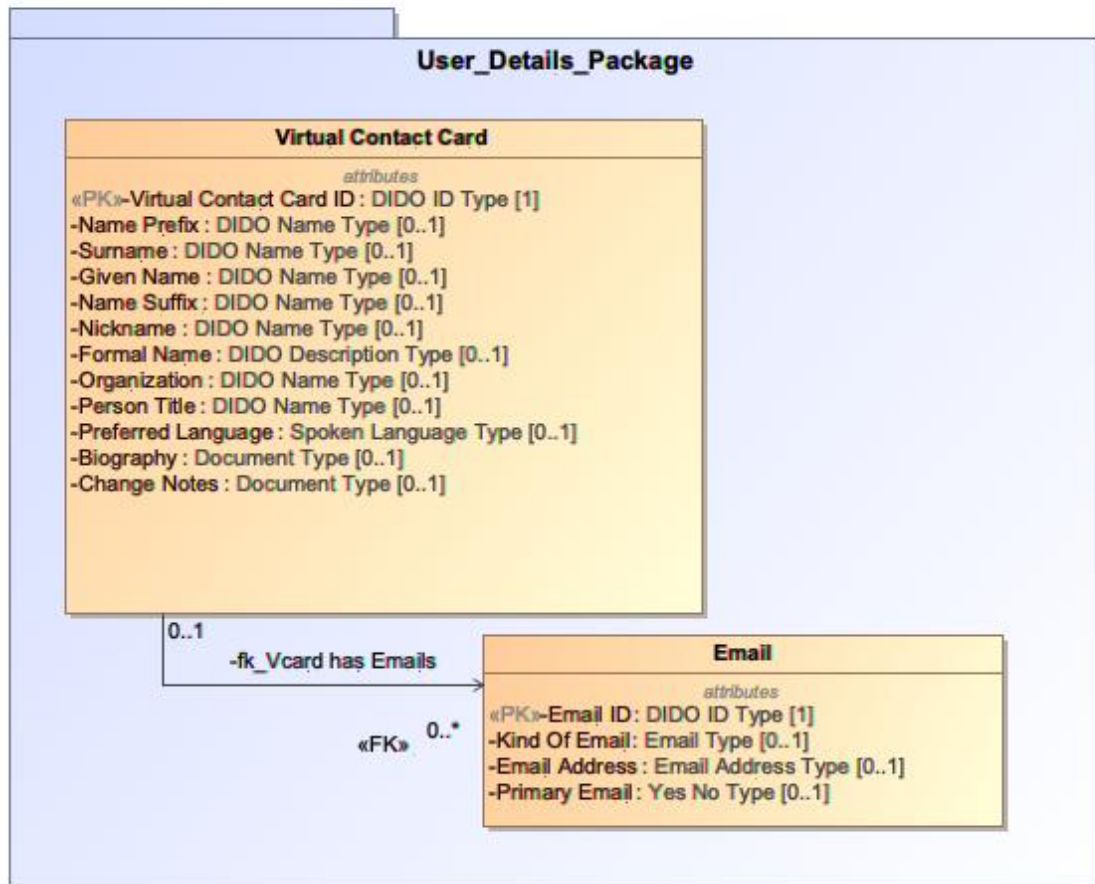


Figure 143. Email

See:

- [Class Email](#)
- [Class Virtual Contact Card](#)

Enumeration Types

Distributed Immutable Data Object Data Model Dido-DM

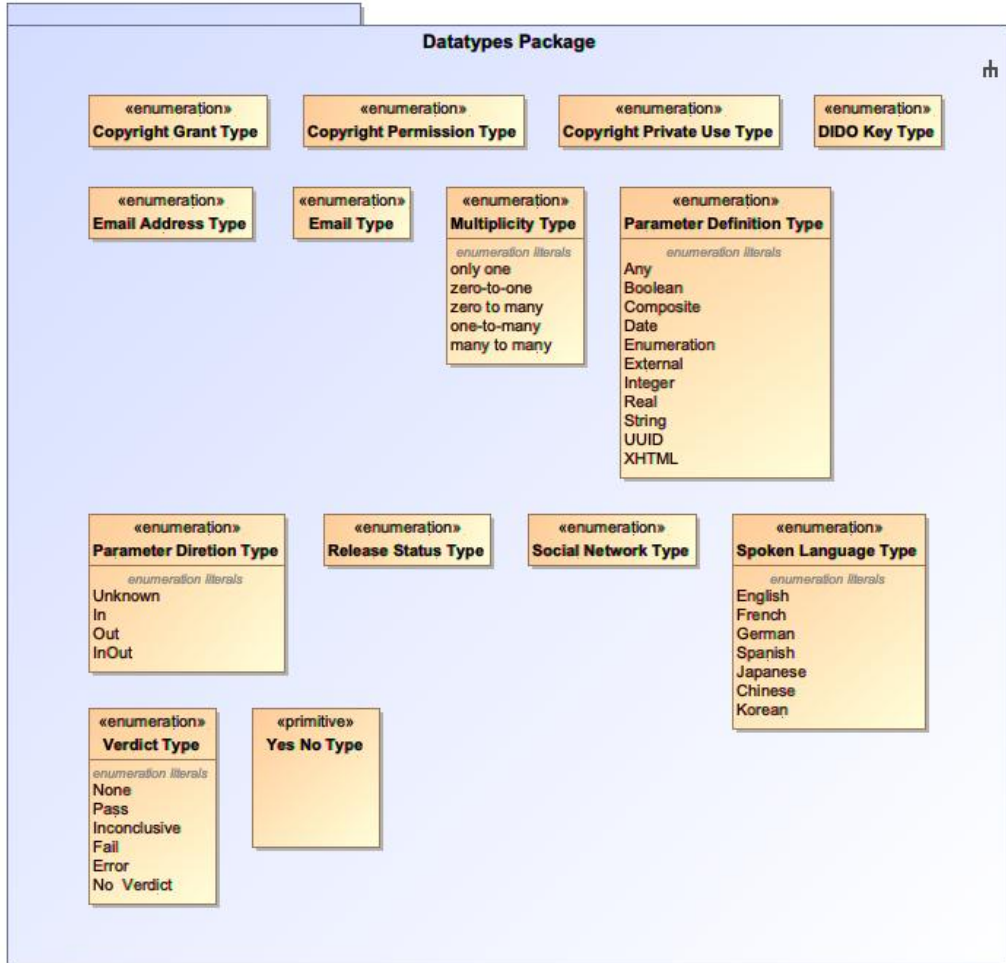


Figure 144. Enumeration Types

Distributed Immutable Data Object Data Model Dido-DM

Expected Results

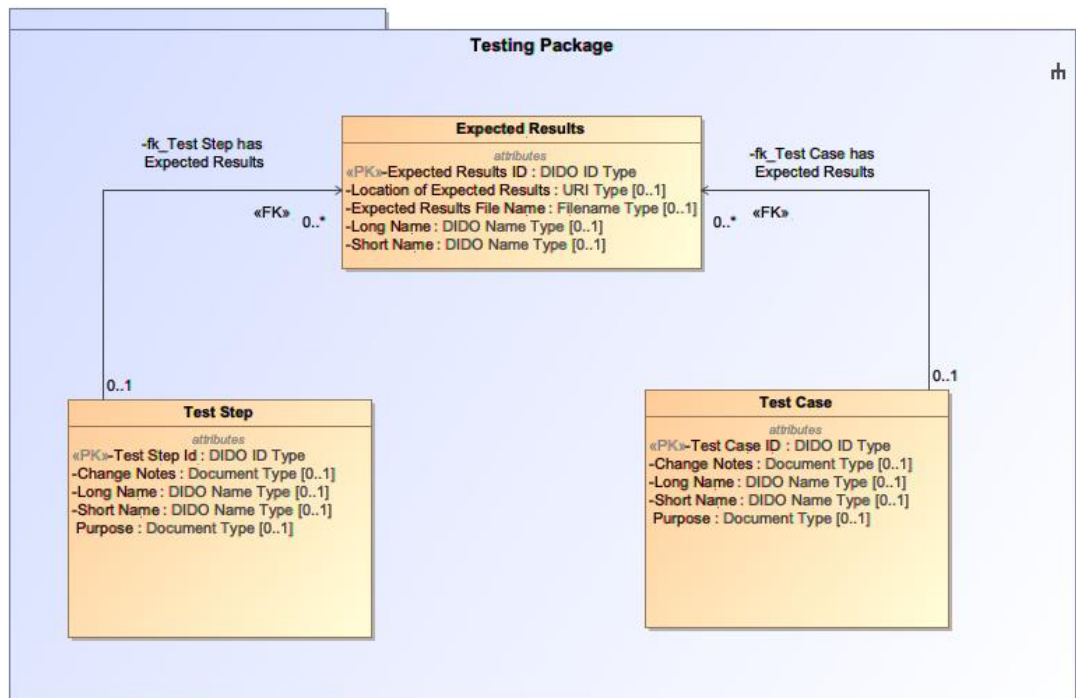


Figure 145. Expected Results

See:

- [Class Expected Results](#)
- [Class Test Step](#)
- [Class Test Case](#)

Distributed Immutable Data Object Data Model

Dido-DM

External Reference

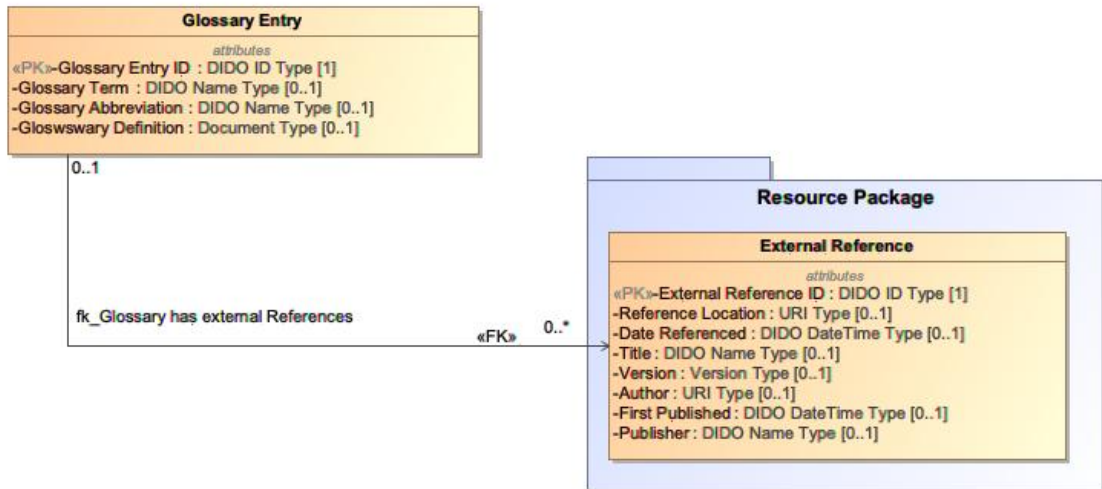


Figure 146. External Reference

See:

- [Class External Reference](#)
- [Class Glossary Entry](#)

Distributed Immutable Data Object Data Model Dido-DM

FAQ_Answers

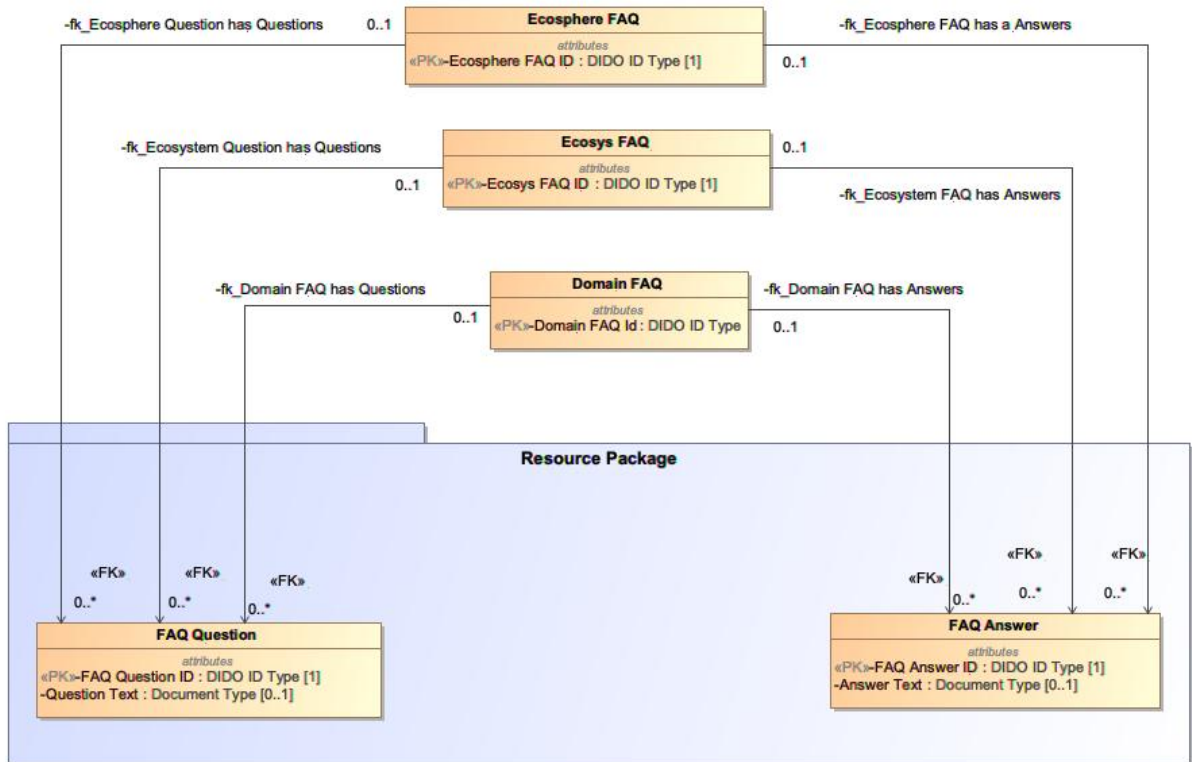


Figure 147. FAQ_Answers

See:

- [Class FAQ Answer](#)
- [Class FAQ Question](#)
- [Class Ecosphere FAQ](#)
- [Class Ecosys FAQ](#)
- [Class Domain FAQ](#)

Distributed Immutable Data Object Data Model

Dido-DM

FAQ_QUESTION

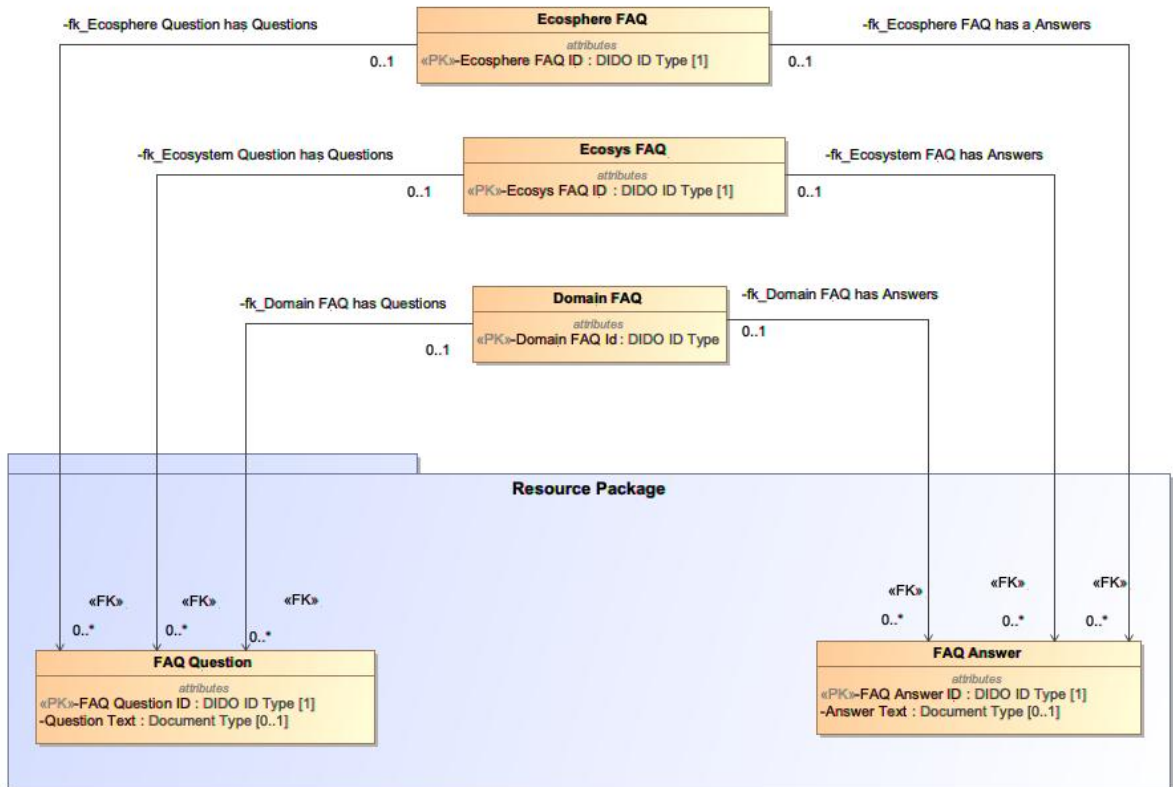


Figure 148. FAQ_QUESTION

See:

- [Class FAQ Question](#)
- [Class FAQ Answer](#)
- [Class Ecosphere FAQ](#)
- [Class Ecosys FAQ](#)
- [Class Domain FAQ](#)

Distributed Immutable Data Object Data Model Dido-DM

Glossary Entry

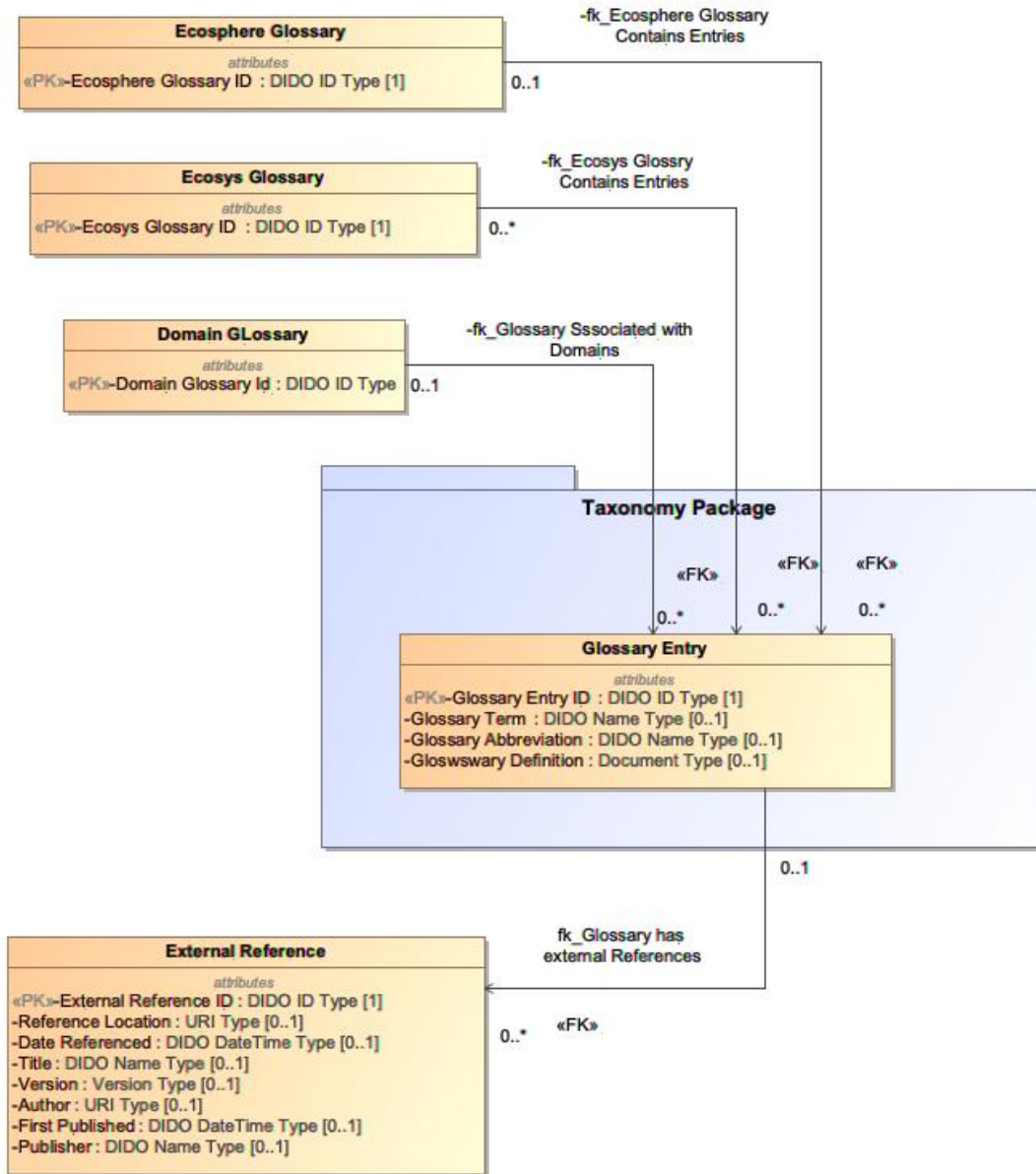


Figure 149. Glossary Entry

See:

- [Class Glossary Entry](#)
- [Class Ecosphere Glossary](#)
- [Class Ecosys Glossary](#)
- [Class Domain Glossary](#)
- [Class External Reference](#)

Distributed Immutable Data Object Data Model

Dido-DM

Image Content

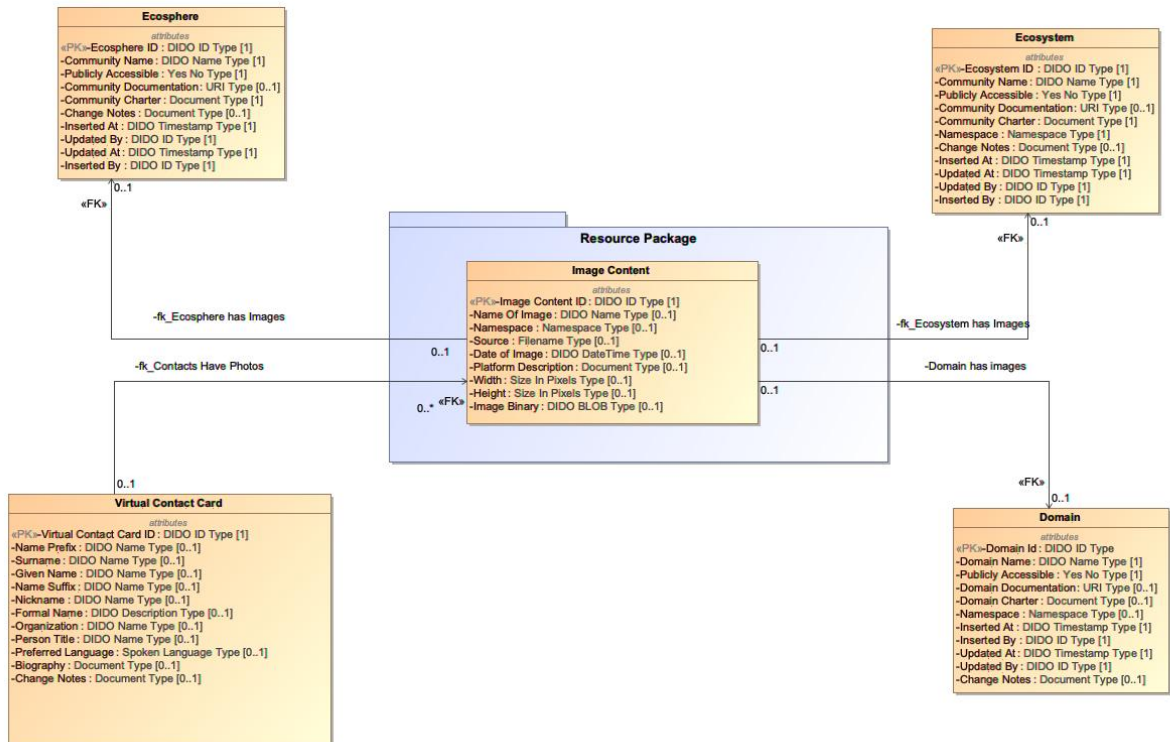


Figure 150. Image Content

See:

- [Class Image Content](#)
- [Class Ecosphere](#)
- [Class Ecosystem](#)
- [Class Domain](#)
- [Class Virtual Contact Card](#)

Distributed Immutable Data Object Data Model

Dido-DM

Key Store

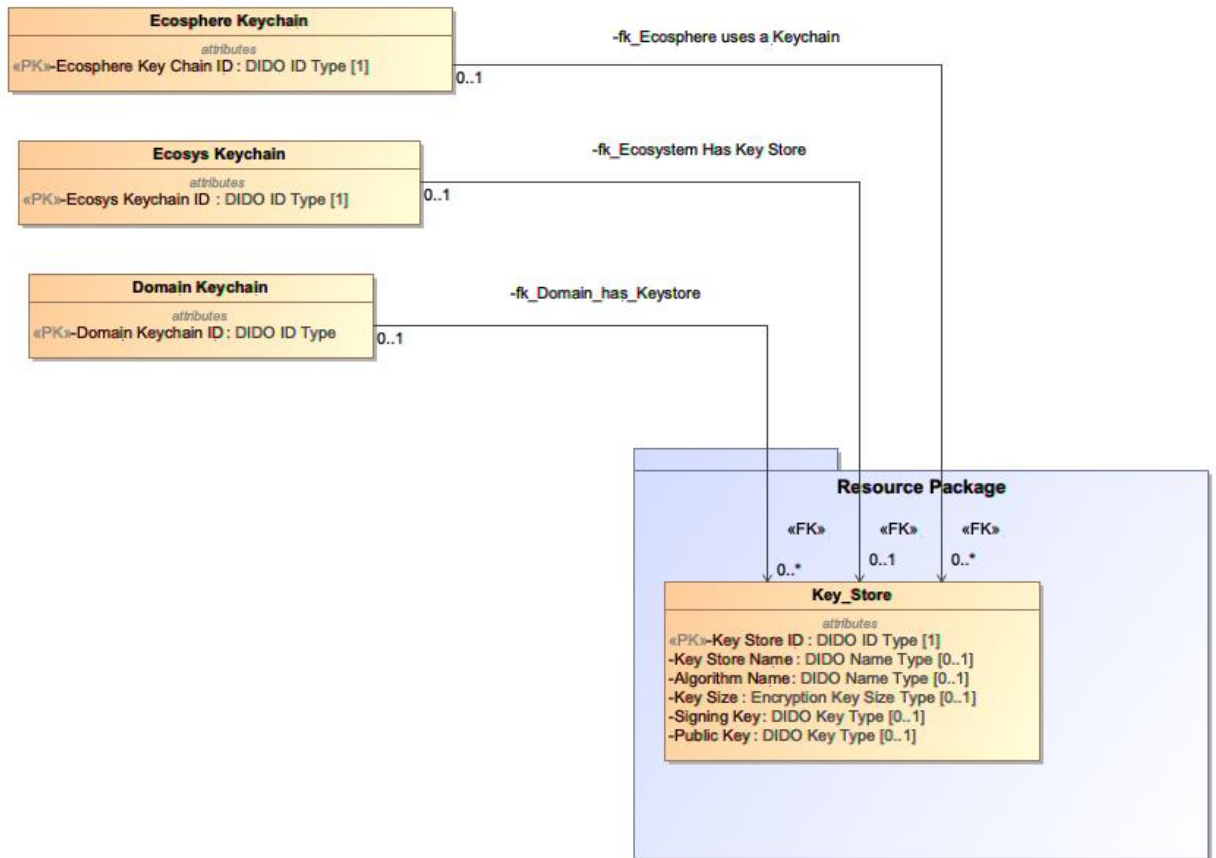


Figure 151. Key Store

See:

- [Class Key_Store](#)
- [Class Ecosphere Keychain](#)
- [Class Ecosys Keychain](#)
- [Class Domain Keychain](#)

Distributed Immutable Data Object Data Model

Dido-DM

Node Platform

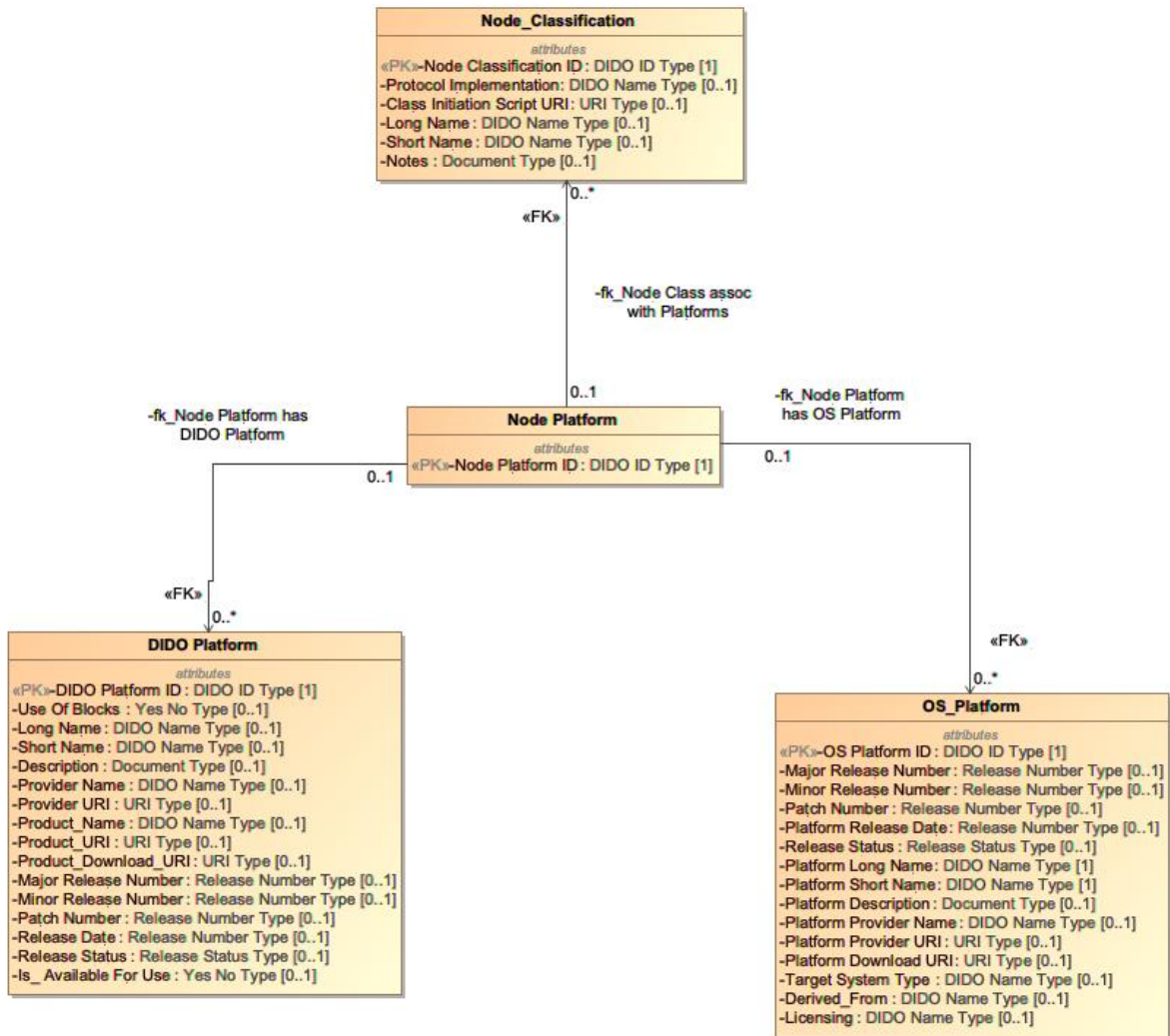


Figure 152. Node Platform

See:

- [Class Node Platform](#)
- [Class Node Classification](#)
- [Class DIDO Platform](#)
- [Class OS Platform](#)

Distributed Immutable Data Object Data Model

Dido-DM

Node Profile

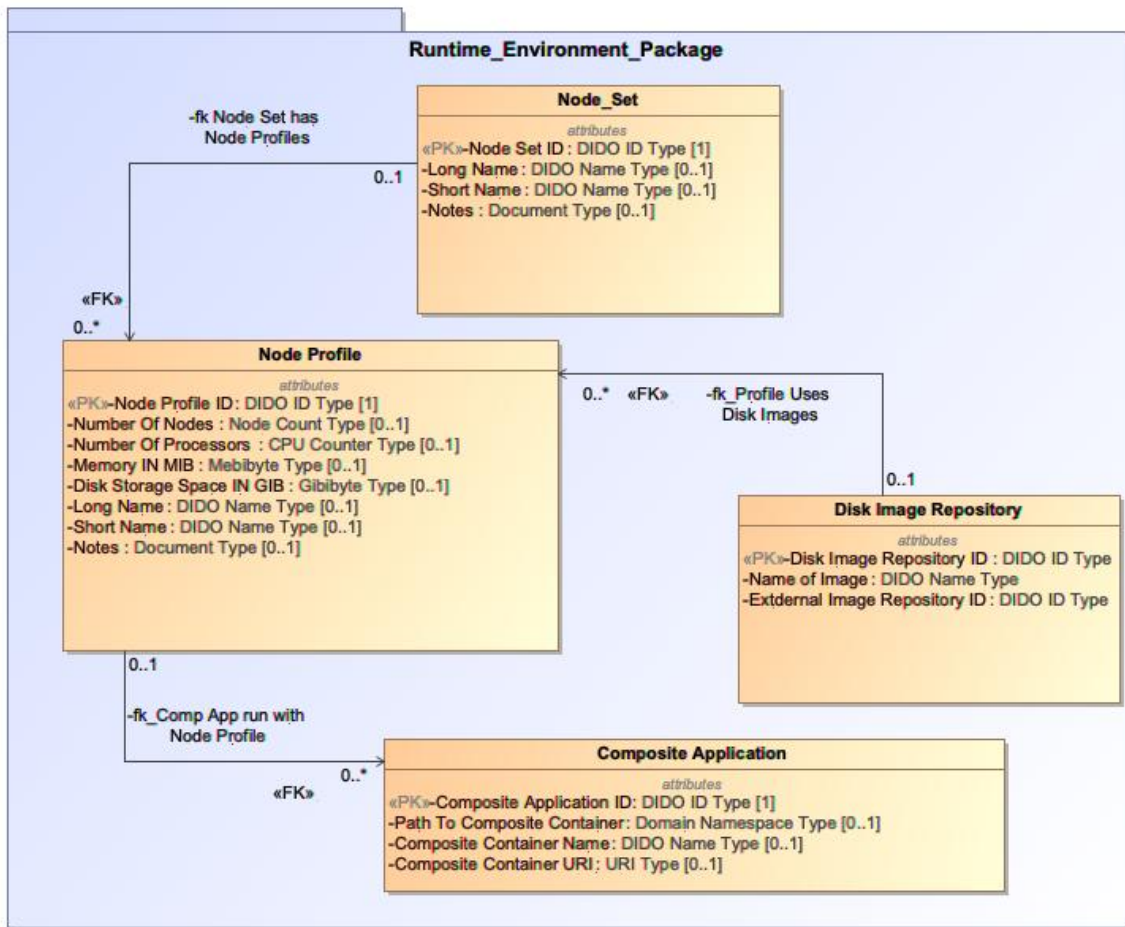


Figure 153. Node Profile

See:

- [Class Node Profile](#)
- [Class Node_Set](#)
- [Class Disk Image Repository](#)
- [Class Composite Application](#)

Distributed Immutable Data Object Data Model

Dido-DM

Node_Classification

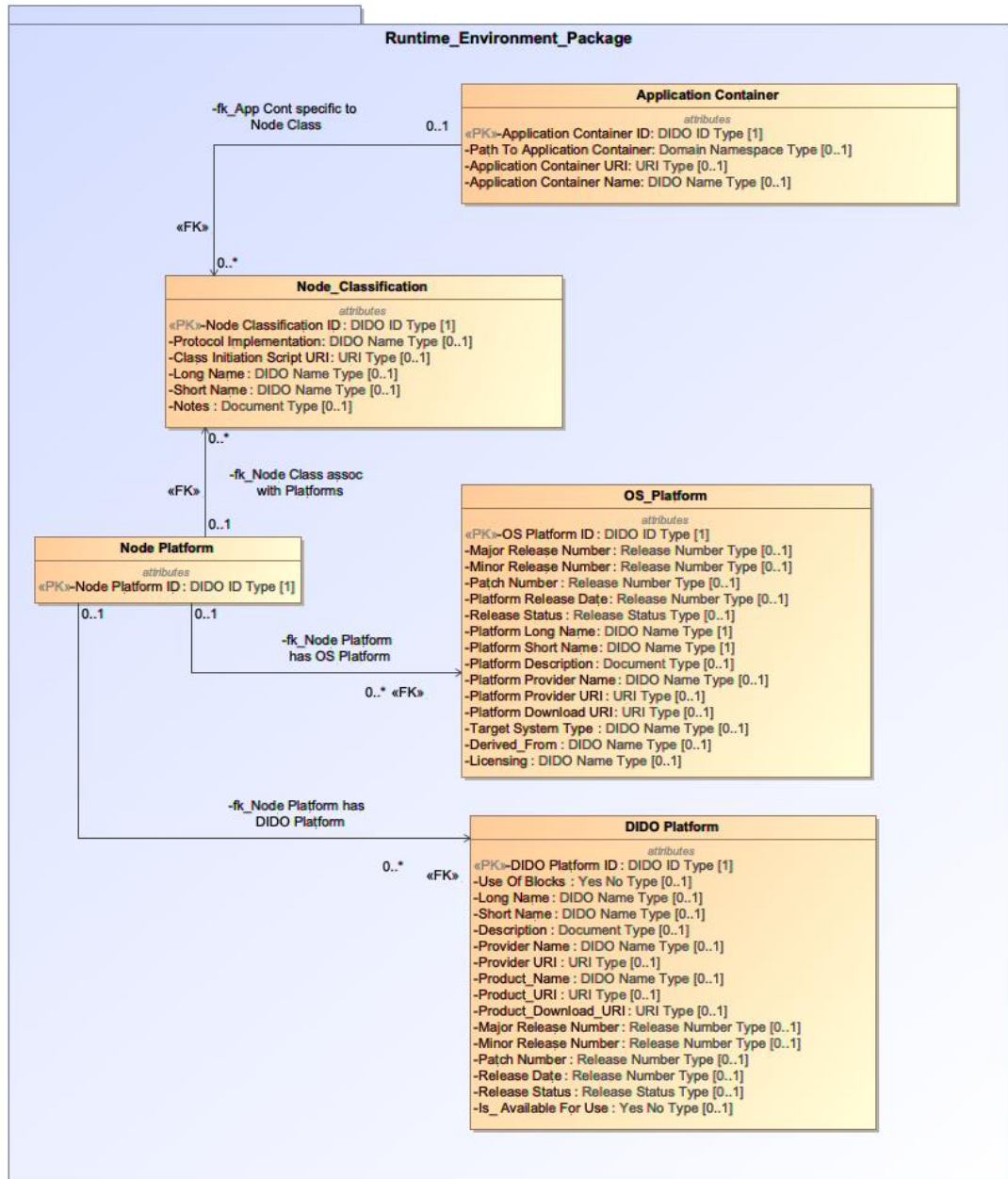


Figure 154. Node_Classification

See:

- [Class Node_Classification](#)
- [Class Application Container](#)
- [Class Node Platform](#)
- [Class OS_Platform](#)
- [Class DIDO Platform](#)

Distributed Immutable Data Object Data Model

Dido-DM

Node_Set

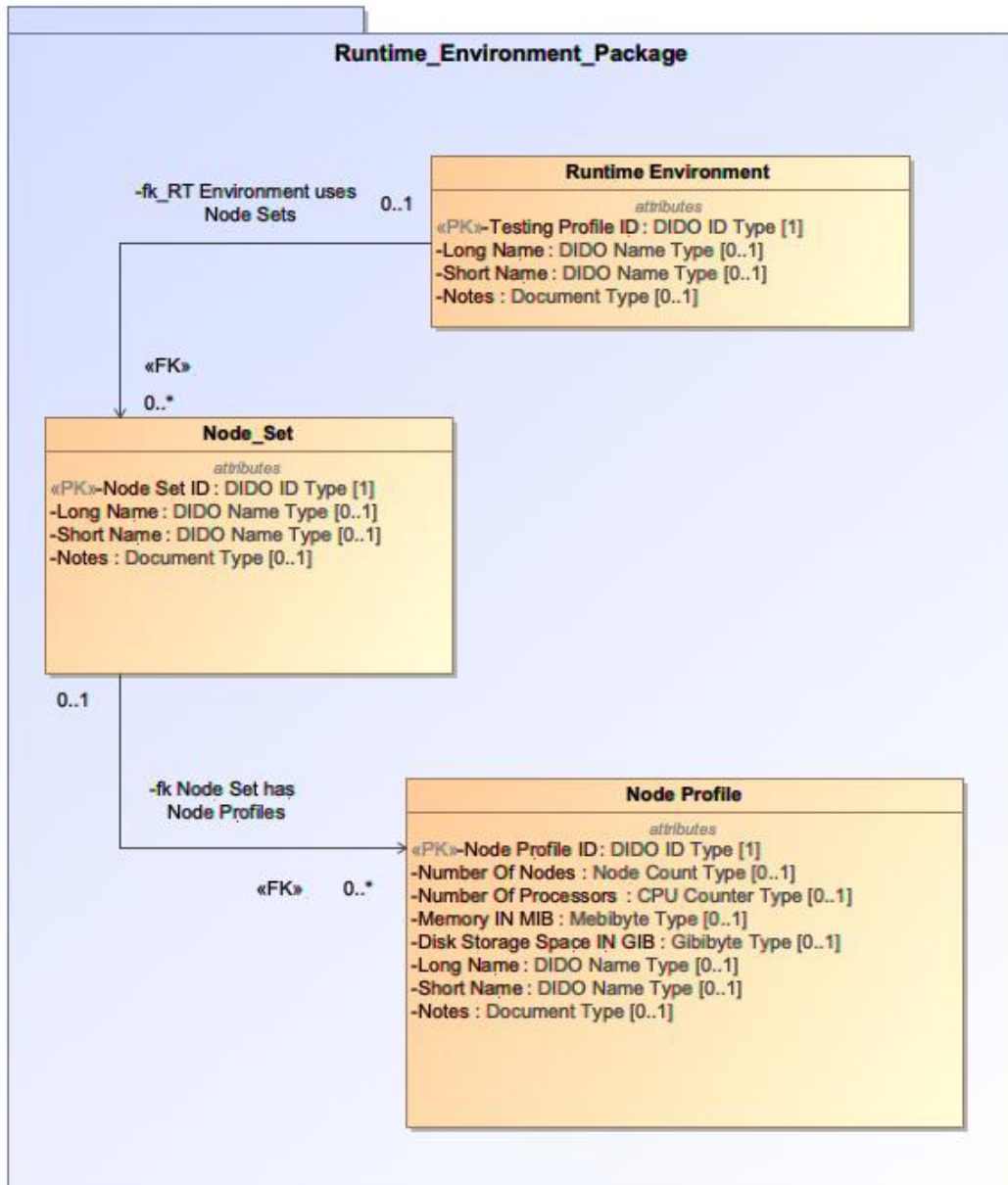


Figure 155. Node_Set

See:

- [Class Node_Set](#)
- [Class Runtime Environment](#)
- [Class Node Profile](#)

Distributed Immutable Data Object Data Model Dido-DM

OS Type

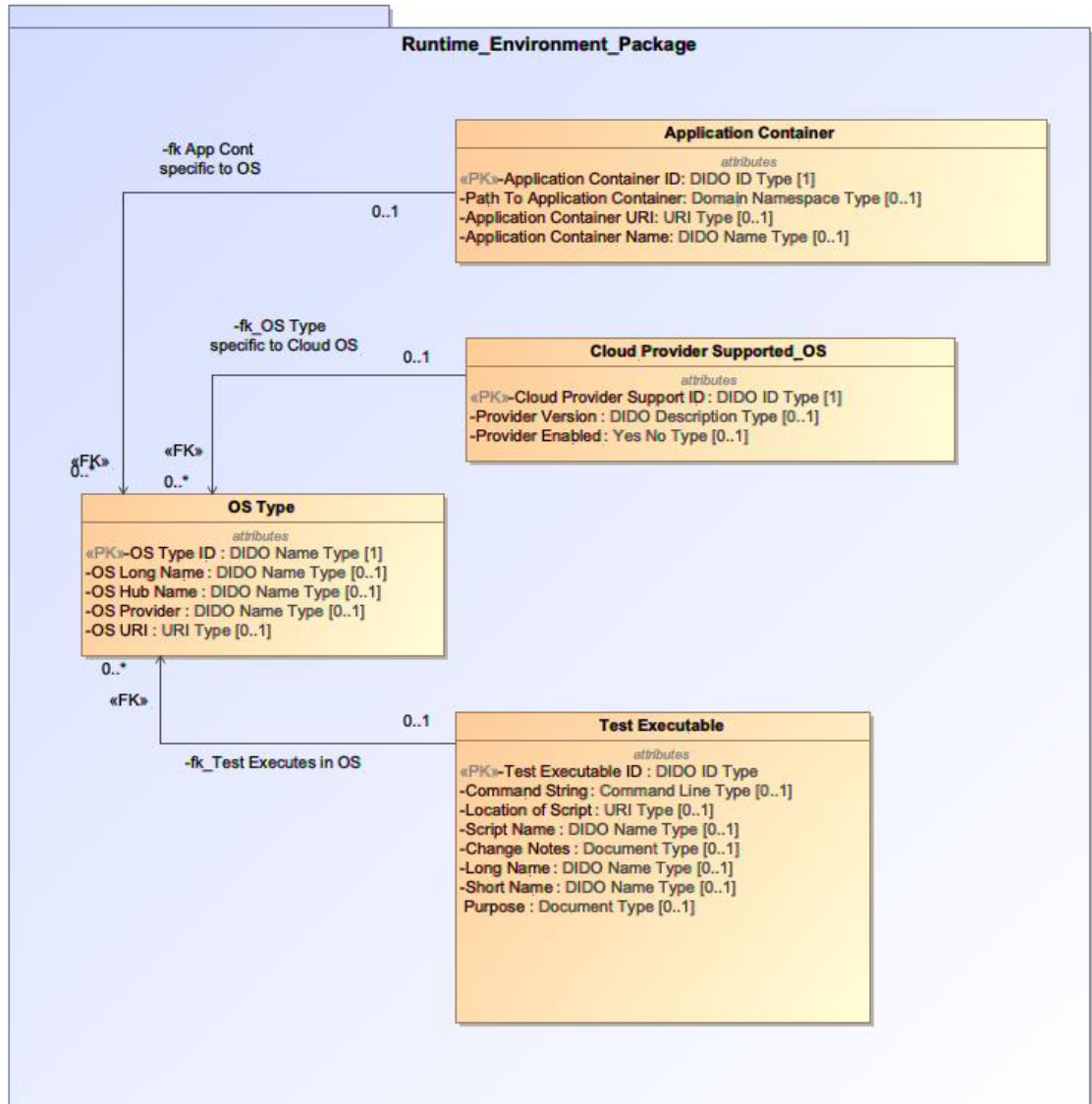


Figure 156. OS Type

See:

- [Class OS Type](#)
- [Class Application Container](#)
- [Class Cloud Provider Supported OS](#)
- [Class Test Executable](#)

Distributed Immutable Data Object Data Model

Dido-DM

OS_Platform

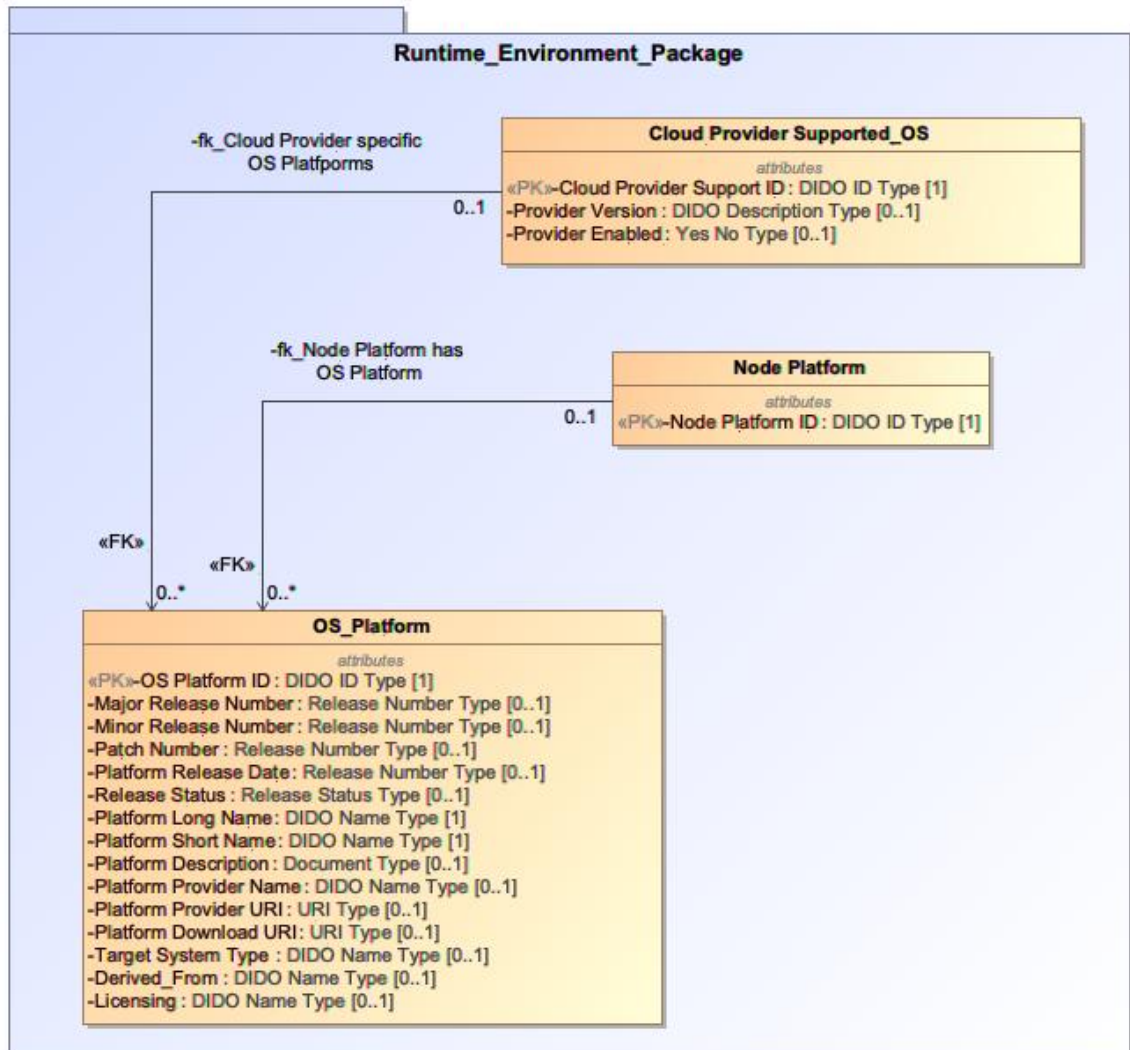


Figure 157. OS_Platform

See:

- [Class OS Platform](#)
- [Class Cloud Provider Supported_OS](#)
- [Class Node Platform](#)

Distributed Immutable Data Object Data Model

Dido-DM

Parameter Definition

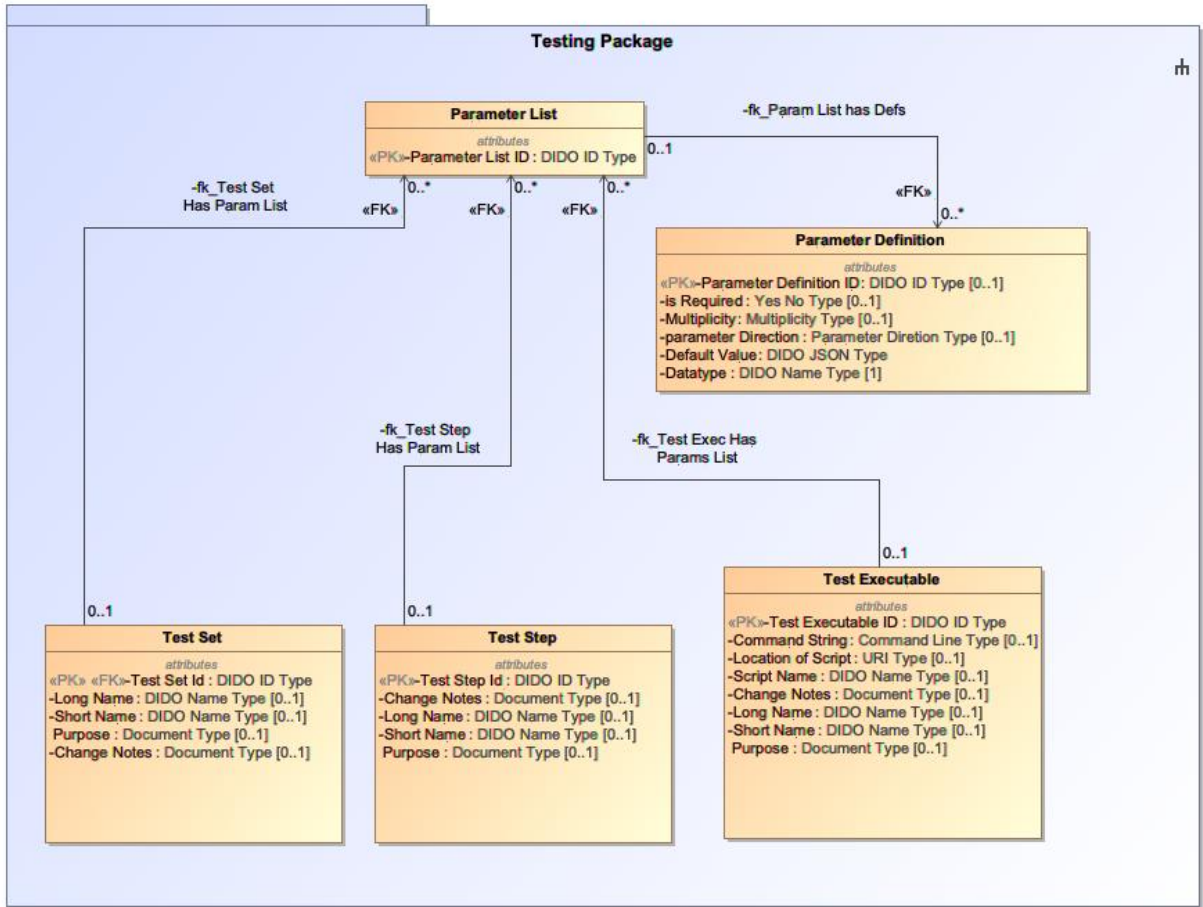


Figure 158. Parameter Definition

See:

- [Class Parameter List](#)
- [Class Parameter Definition](#)
- [Class Test Executable](#)
- [Class Test Step](#)

Distributed Immutable Data Object Data Model Dido-DM

Parameter List

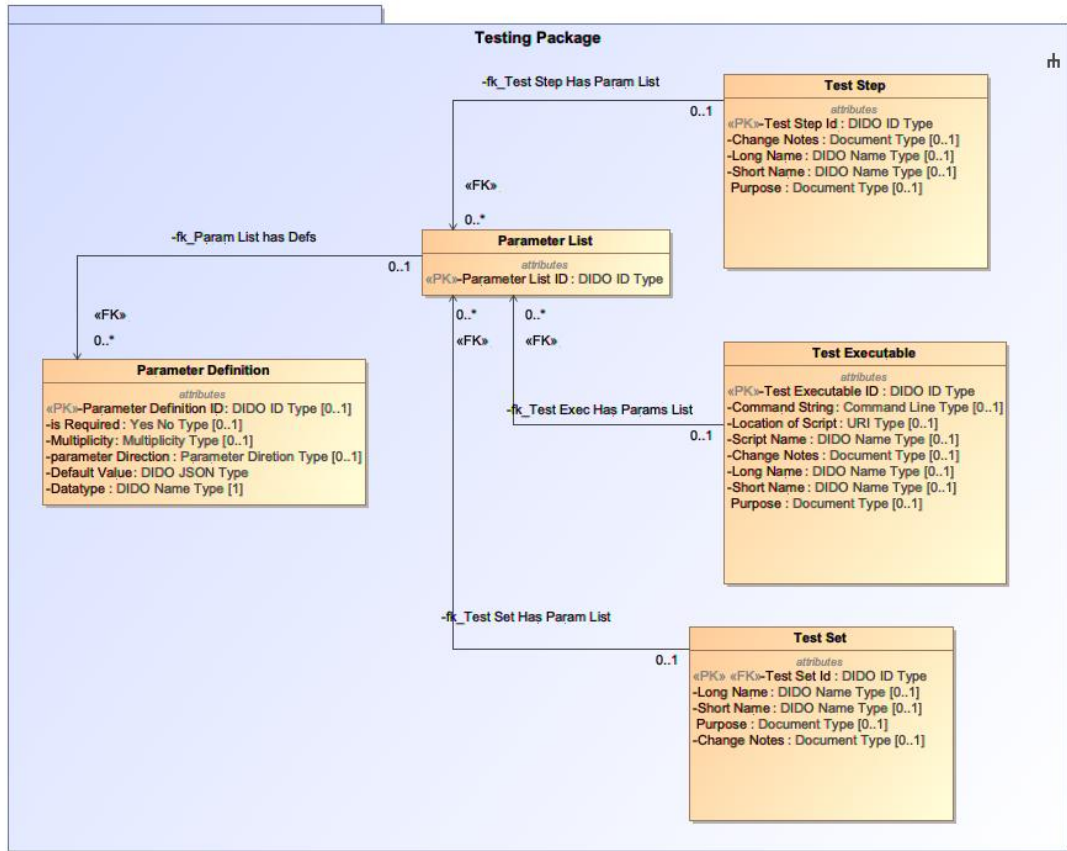


Figure 159. Parameter List

See:

- [Class Parameter List](#)
- [Class Parameter Definition](#)
- [Class Test Step](#)
- [Class Test Executable](#)
- [Class Test Set](#)

Distributed Immutable Data Object Data Model

Dido-DM

Phone Number

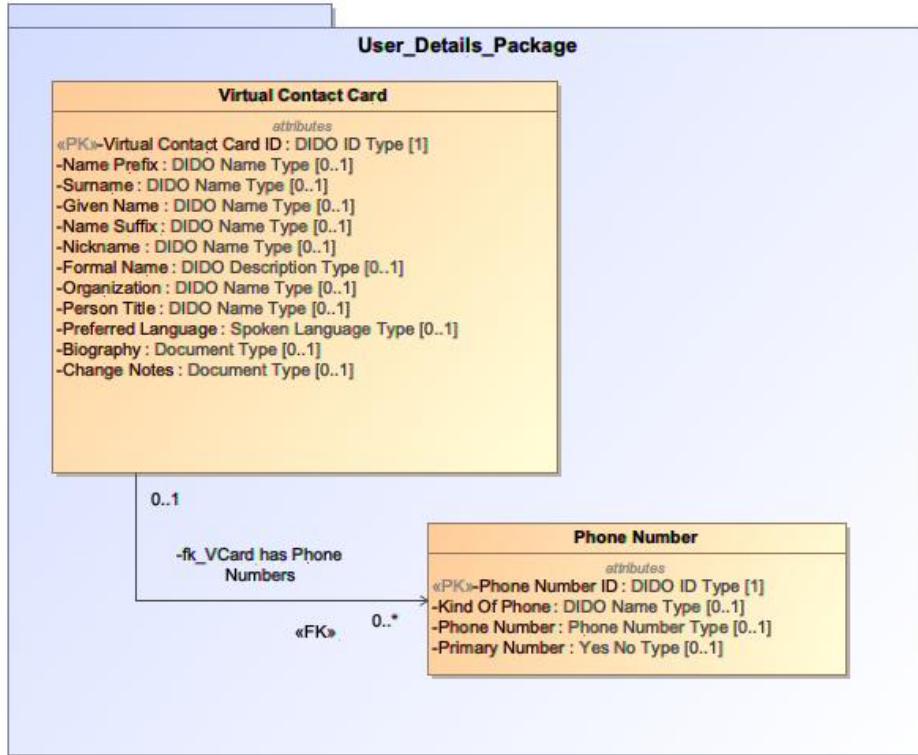


Figure 160. Phone Number

See:
•

Distributed Immutable Data Object Data Model Dido-DM

Primitive Types

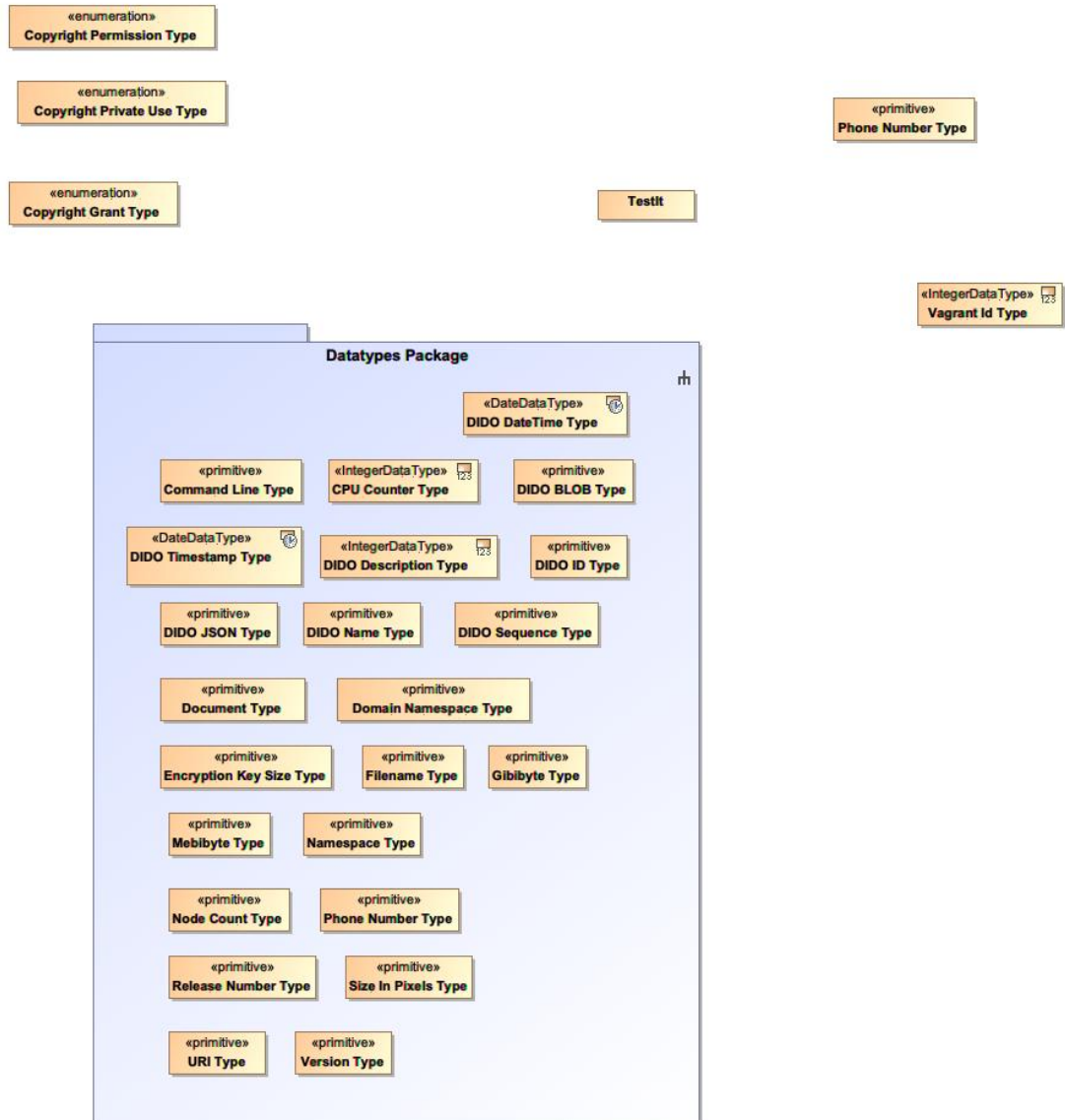


Figure 161. Primitive Types

Role

Distributed Immutable Data Object Data Model Dido-DM

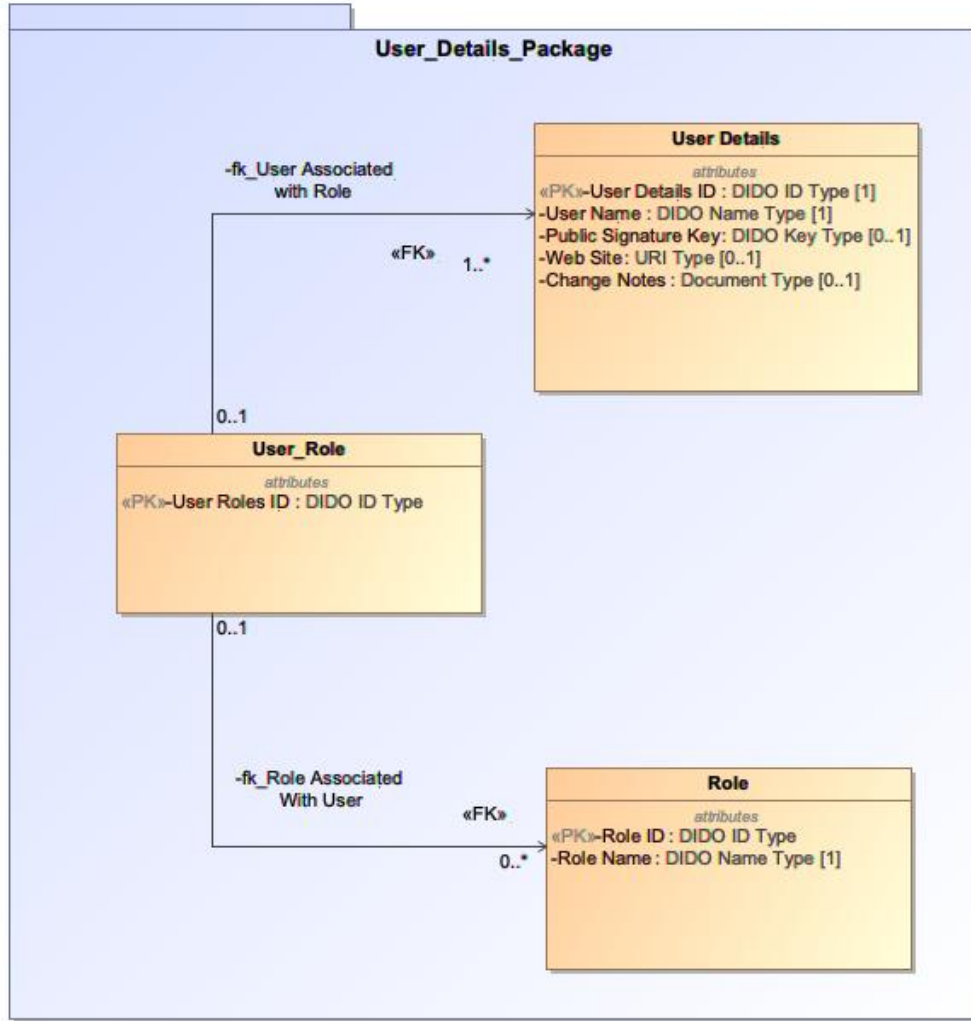


Figure 162. Role

Runtime Environment

Distributed Immutable Data Object Data Model

Dido-DM

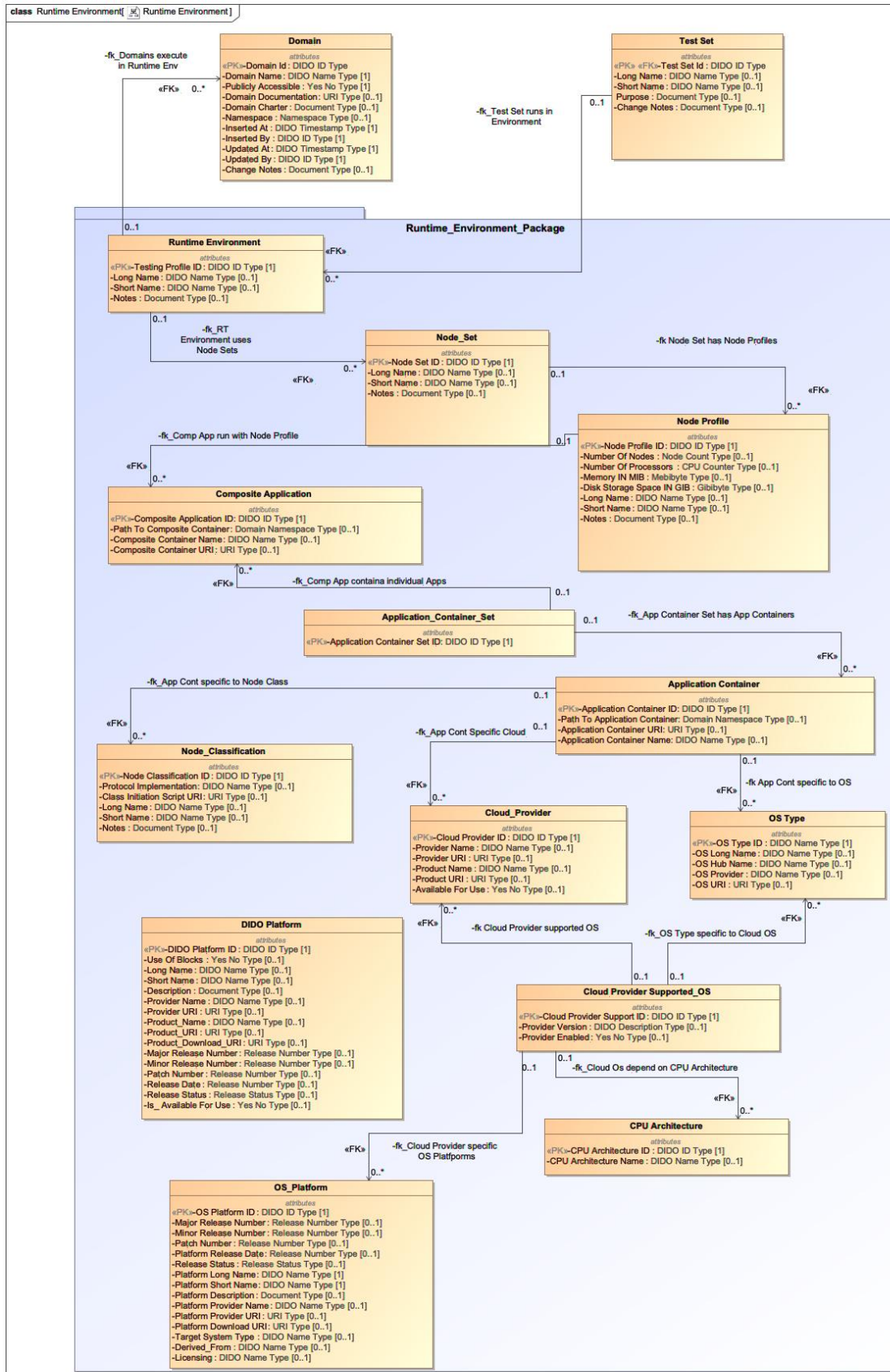


Figure 163. Runtime Environment

Distributed Immutable Data Object Data Model Dido-DM

Signon Provider

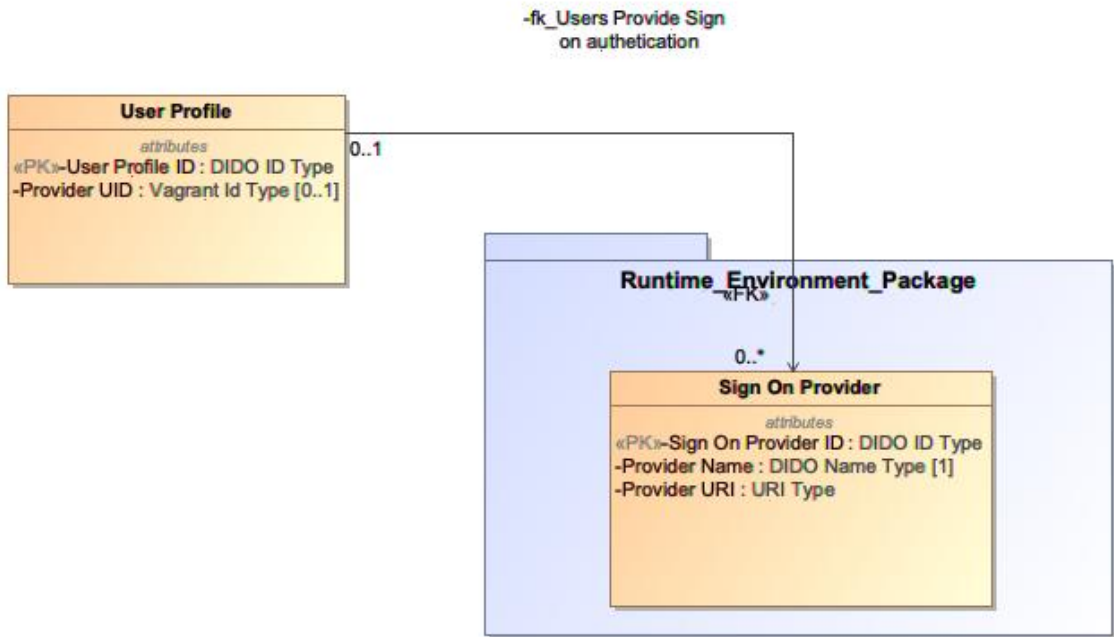


Figure 164. Signon Provider

Smart Contract

Distributed Immutable Data Object Data Model Dido-DM

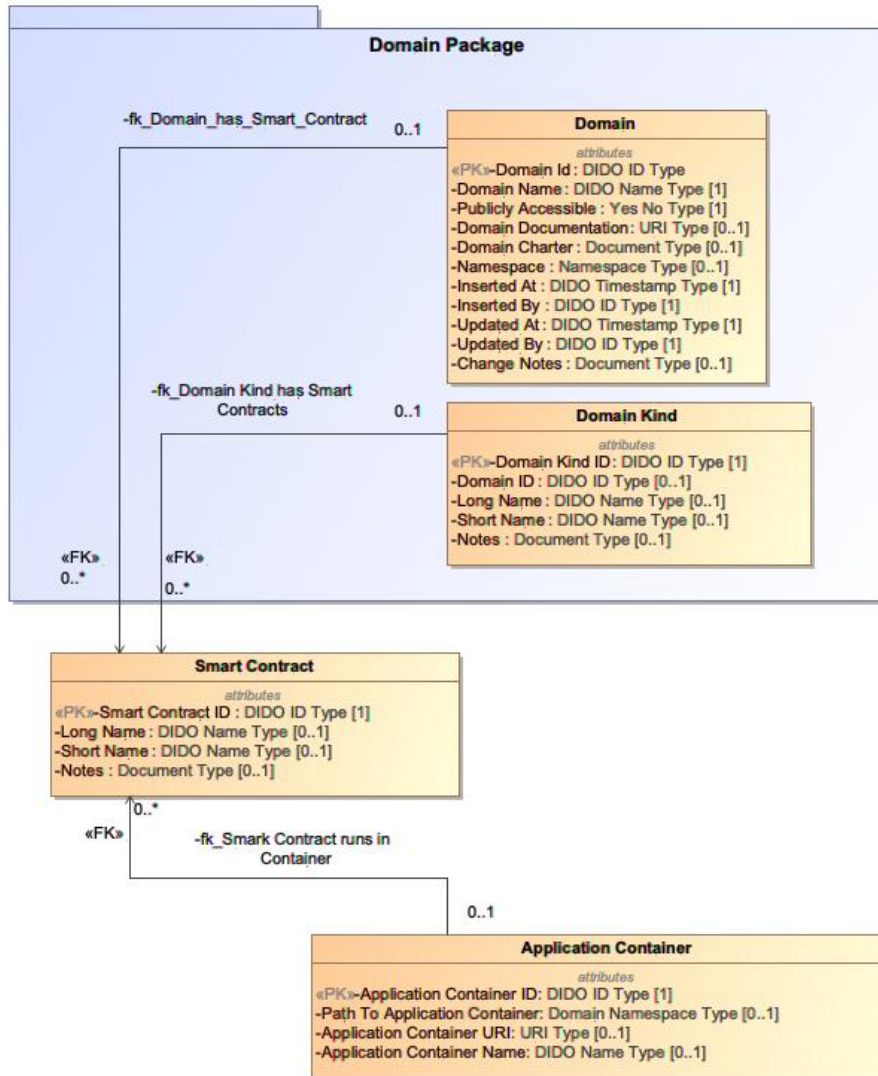


Figure 165. Smart Contract

Social Network Connection

Distributed Immutable Data Object Data Model Dido-DM

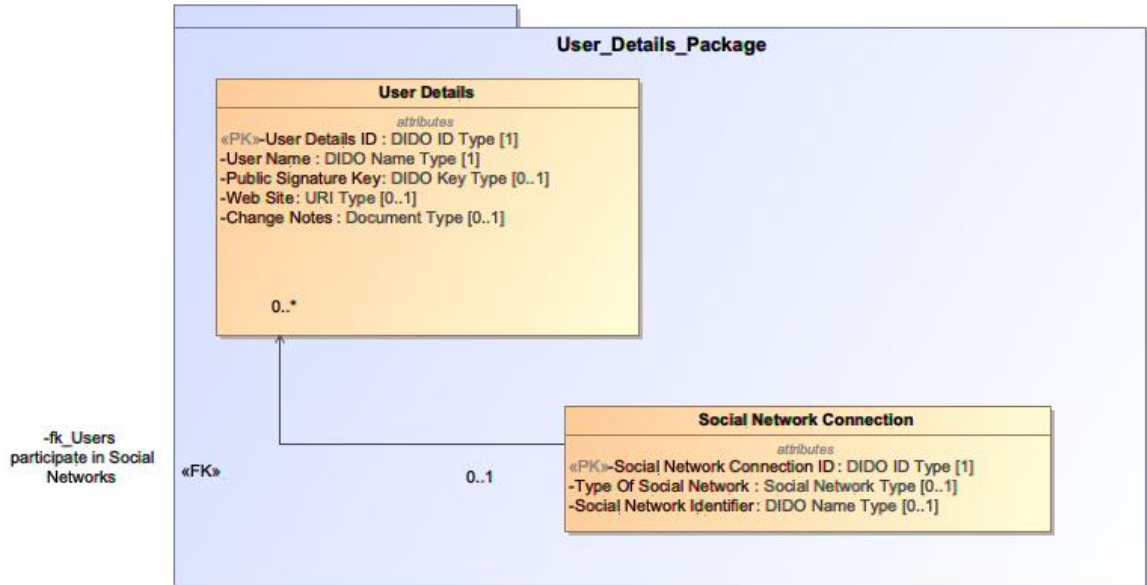


Figure 166. Social Network Connection

Street Address

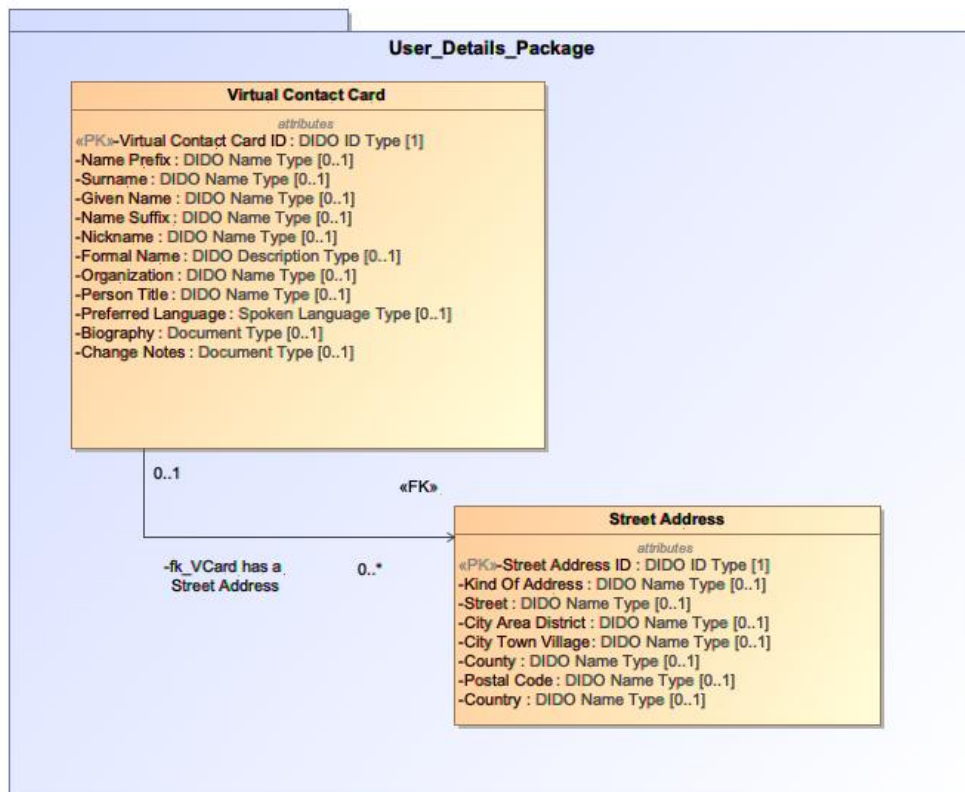


Figure 167. Street Address

Distributed Immutable Data Object Data Model Dido-DM

Tag

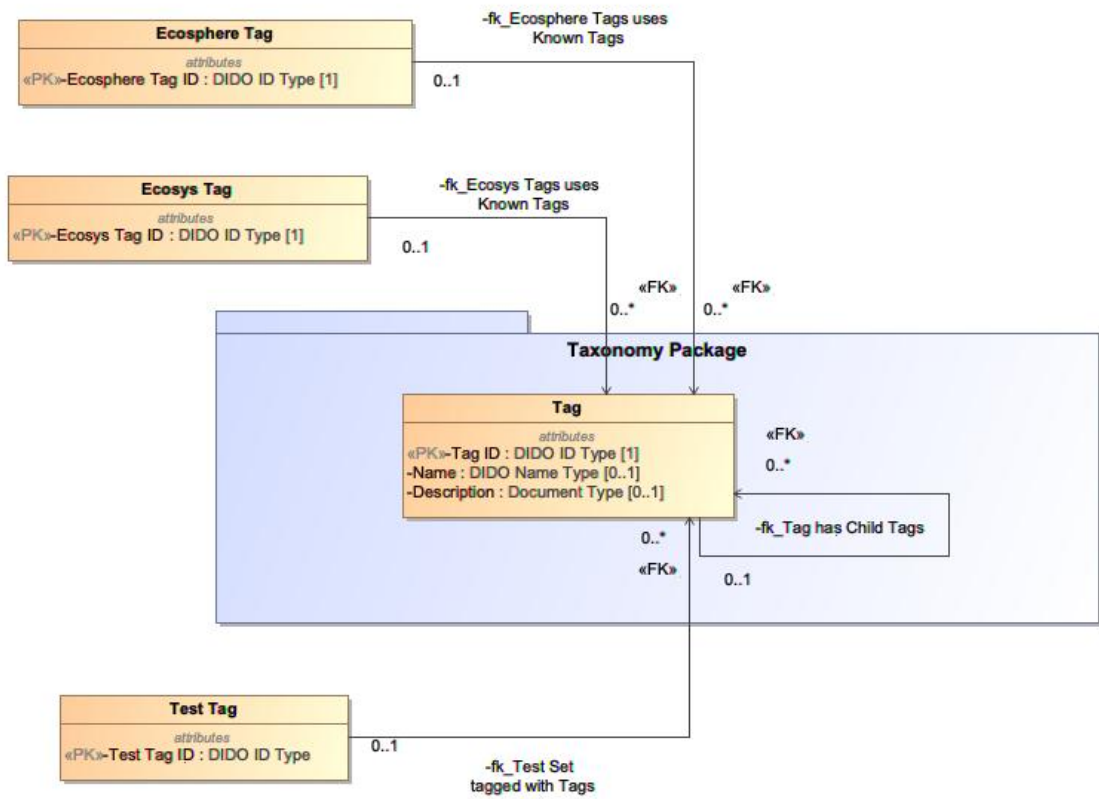


Figure 168. Tag

Test

Distributed Immutable Data Object Data Model Dido-DM

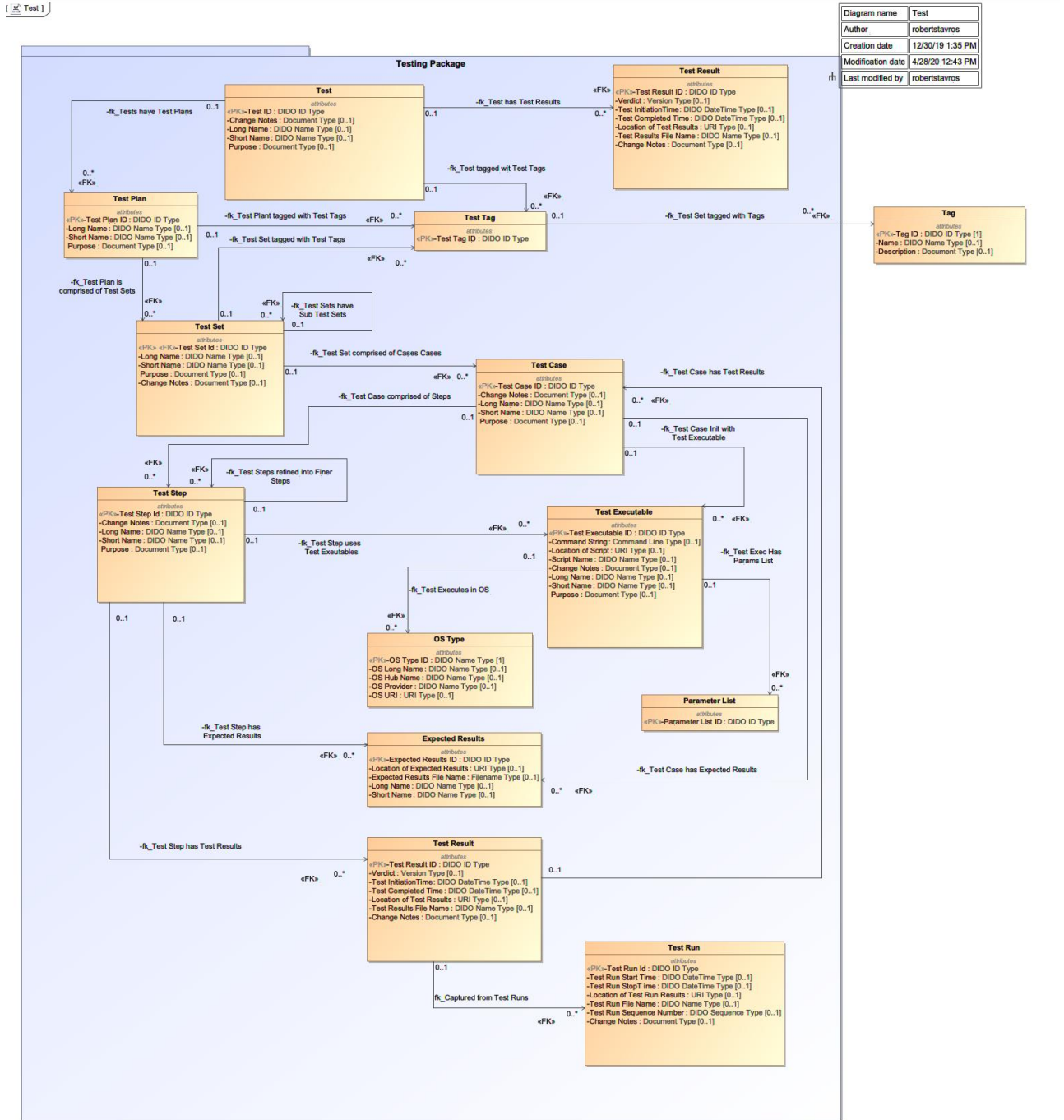


Figure 169. Test

Test Plan Package

Distributed Immutable Data Object Data Model

Dido-DM

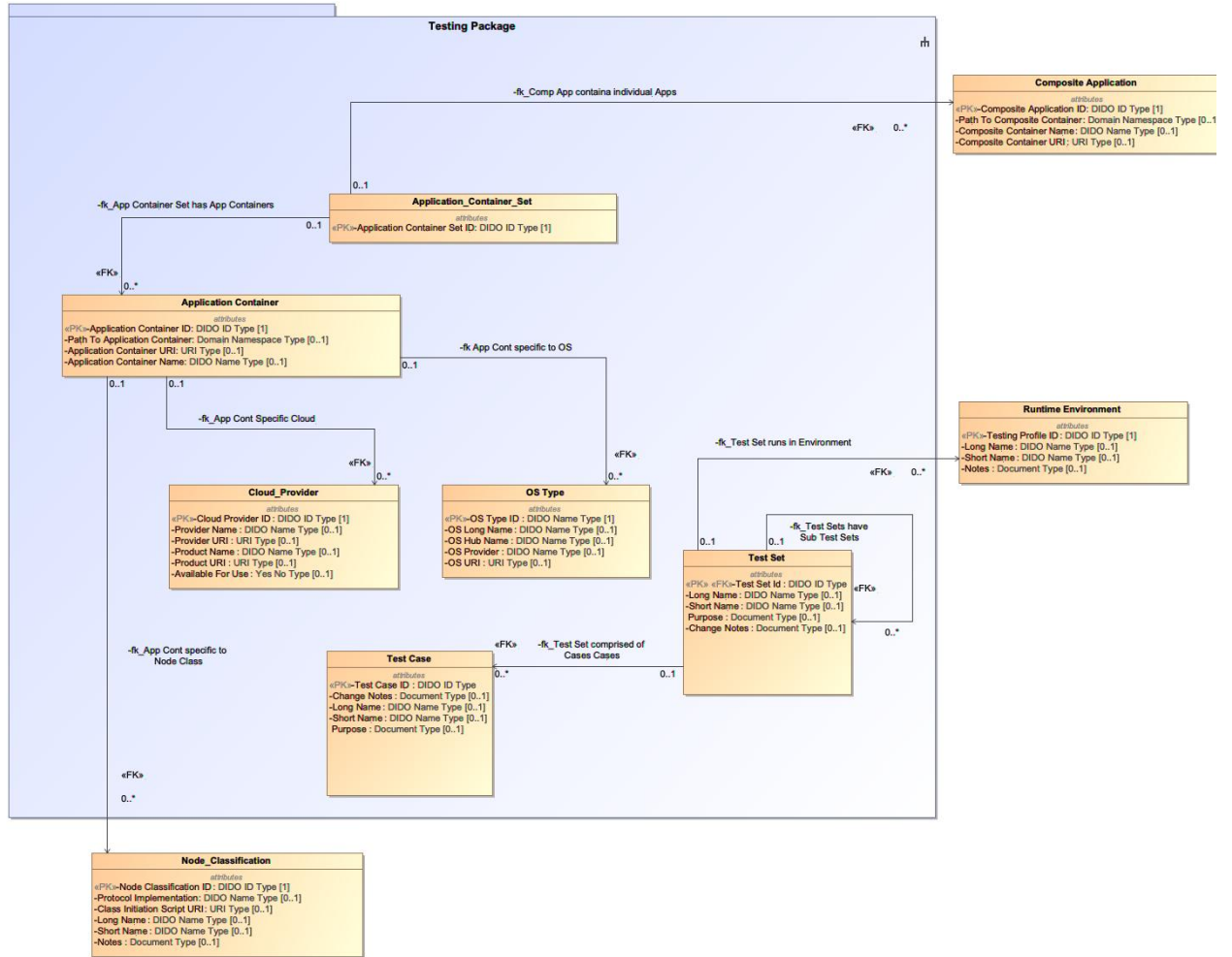


Figure 170. Test Plan Package

Test_Case

Distributed Immutable Data Object Data Model Dido-DM

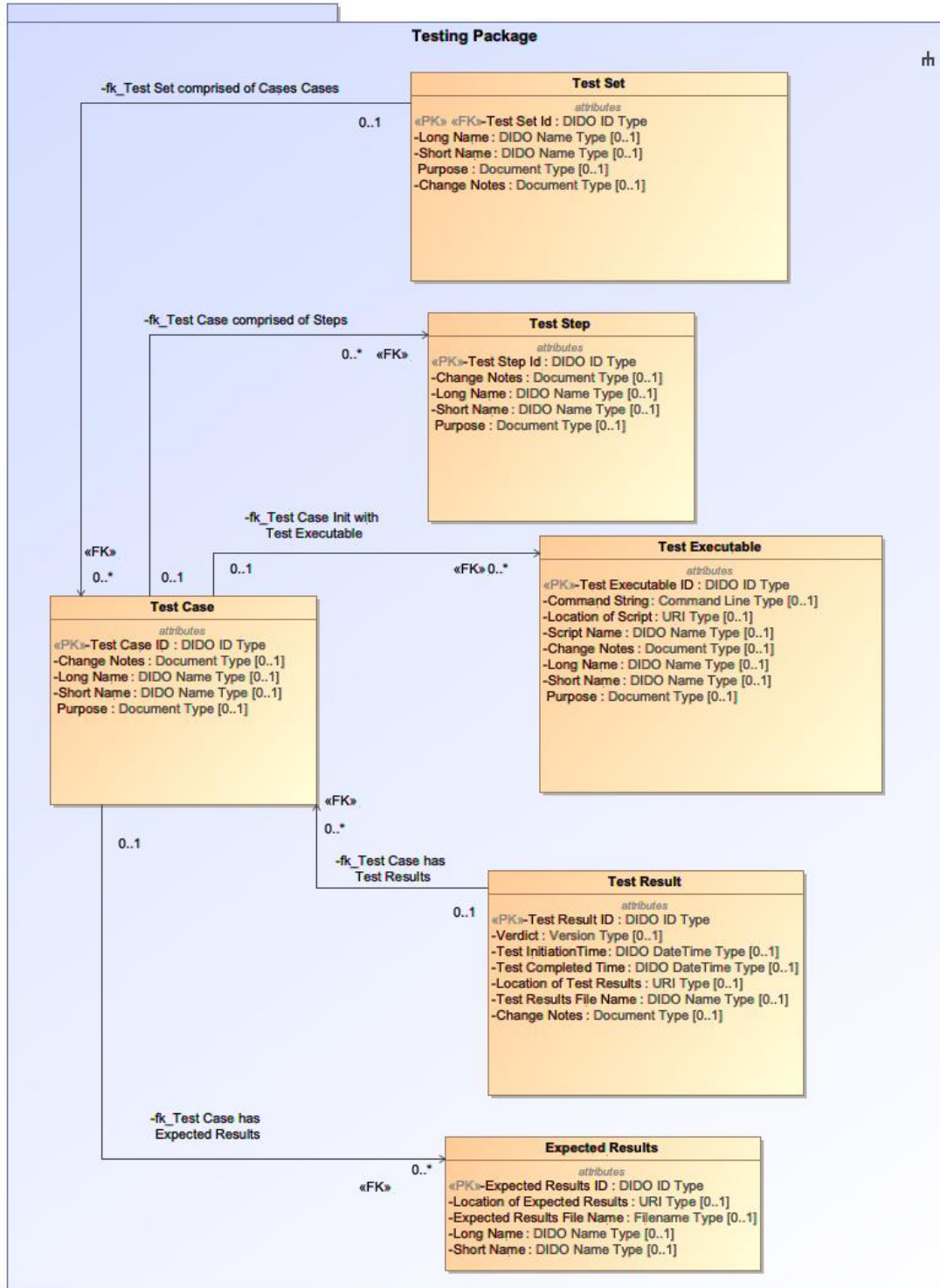


Figure 171. Test_Case

Test_Executable

Distributed Immutable Data Object Data Model Dido-DM

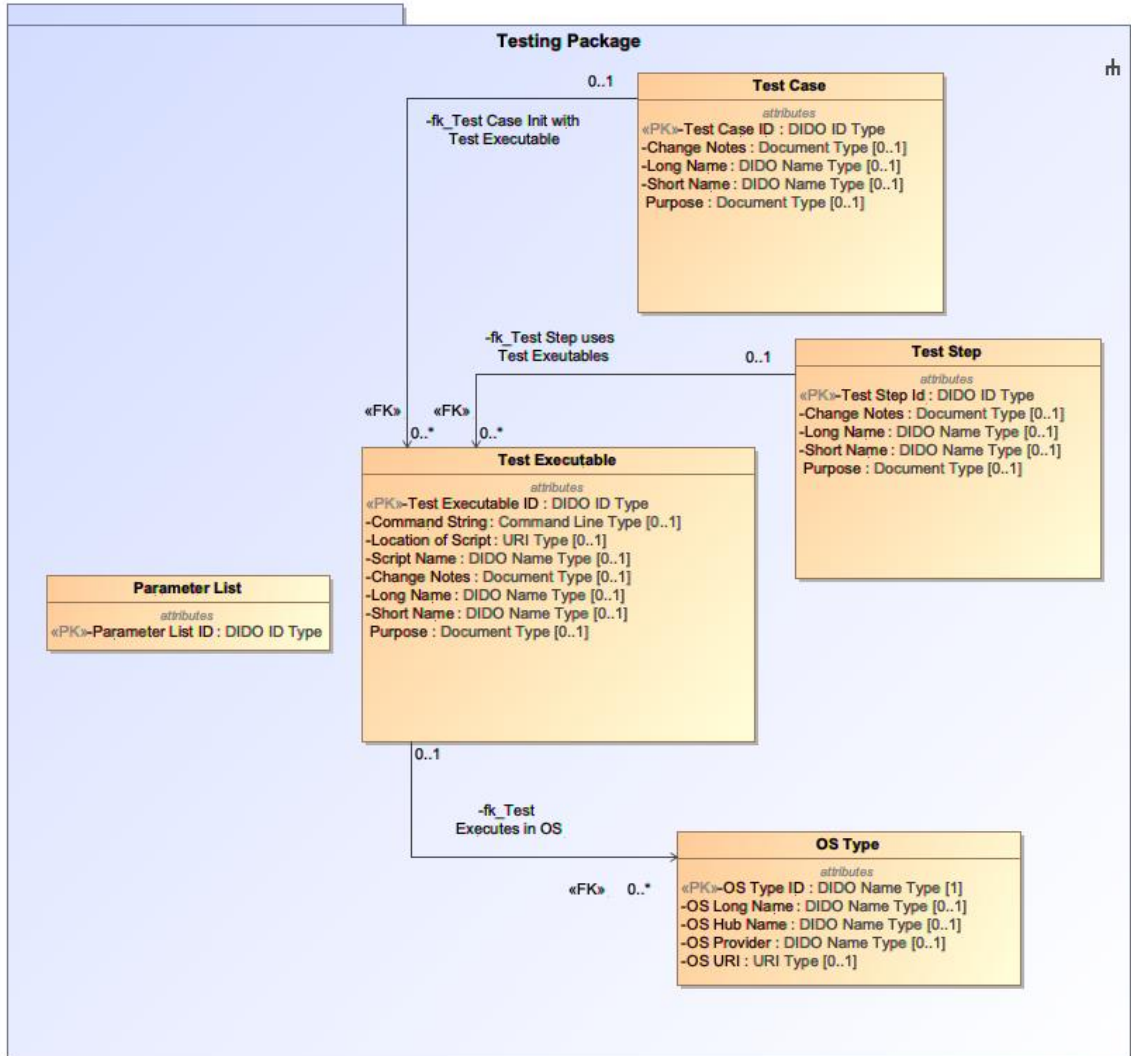


Figure 172. Test_Executable

Test_Plan

Distributed Immutable Data Object Data Model Dido-DM

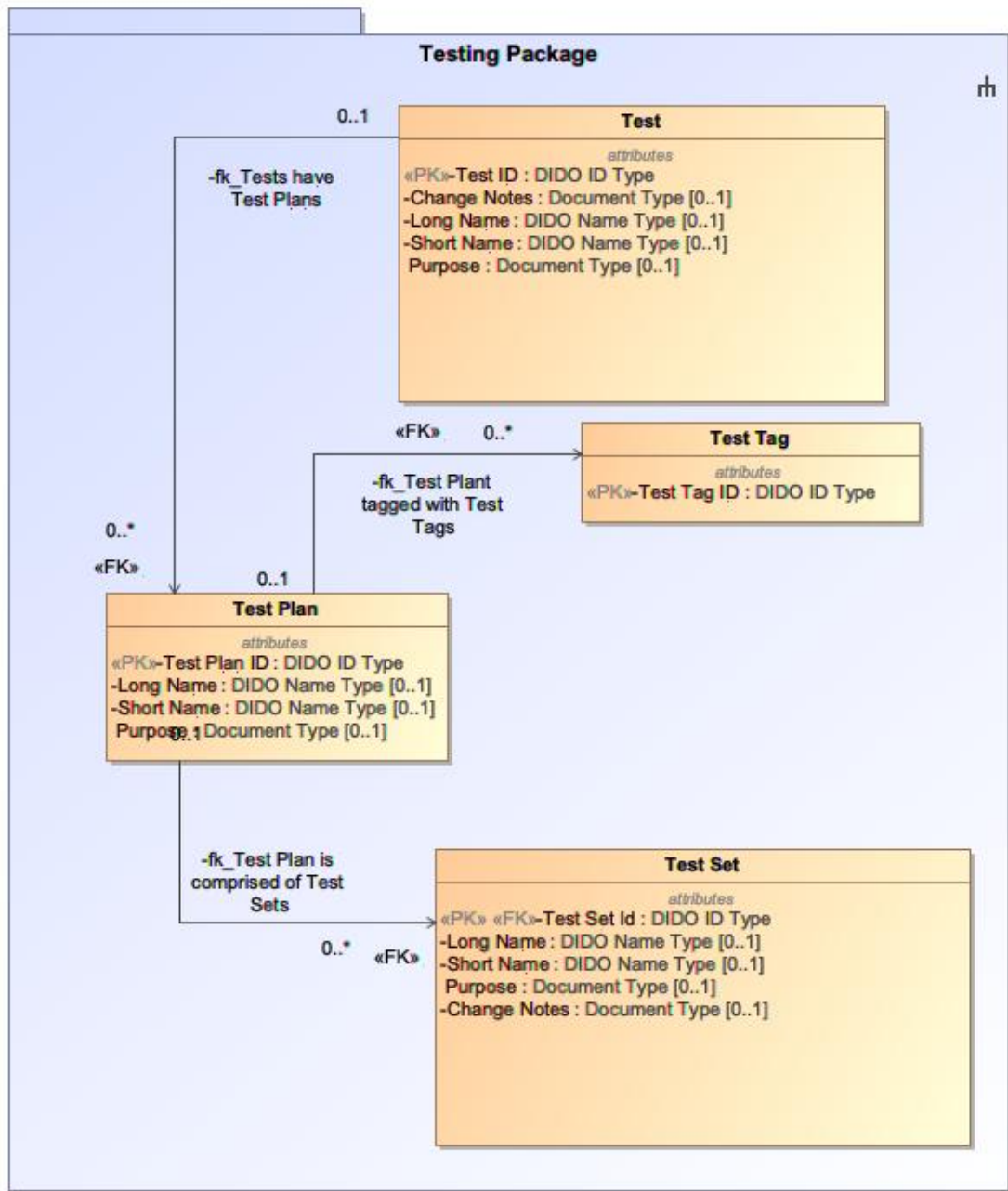


Figure 173. Test_Plan

Test_Result

Distributed Immutable Data Object Data Model Dido-DM

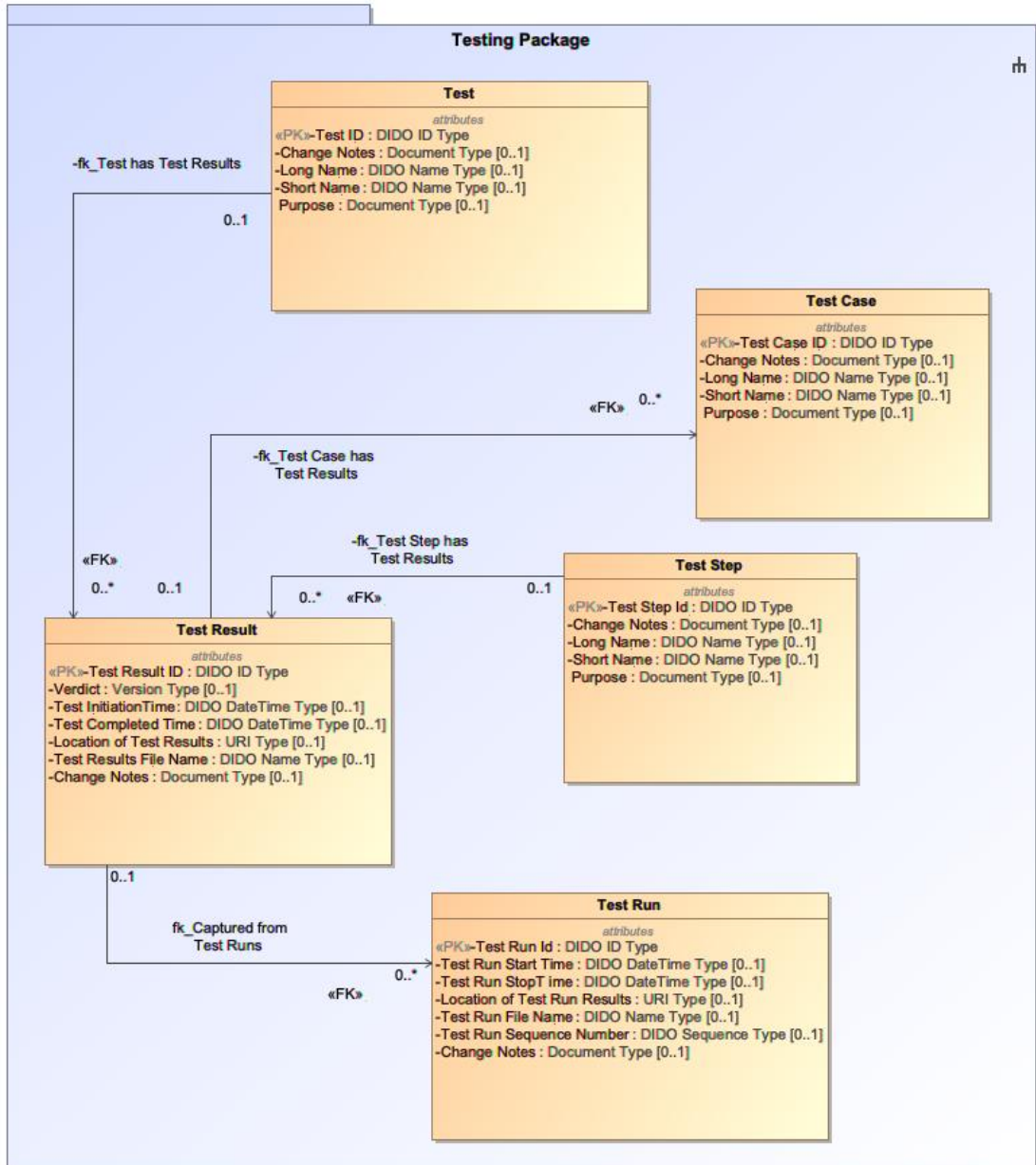


Figure 174. Test_Result

Test_Run

Distributed Immutable Data Object Data Model Dido-DM

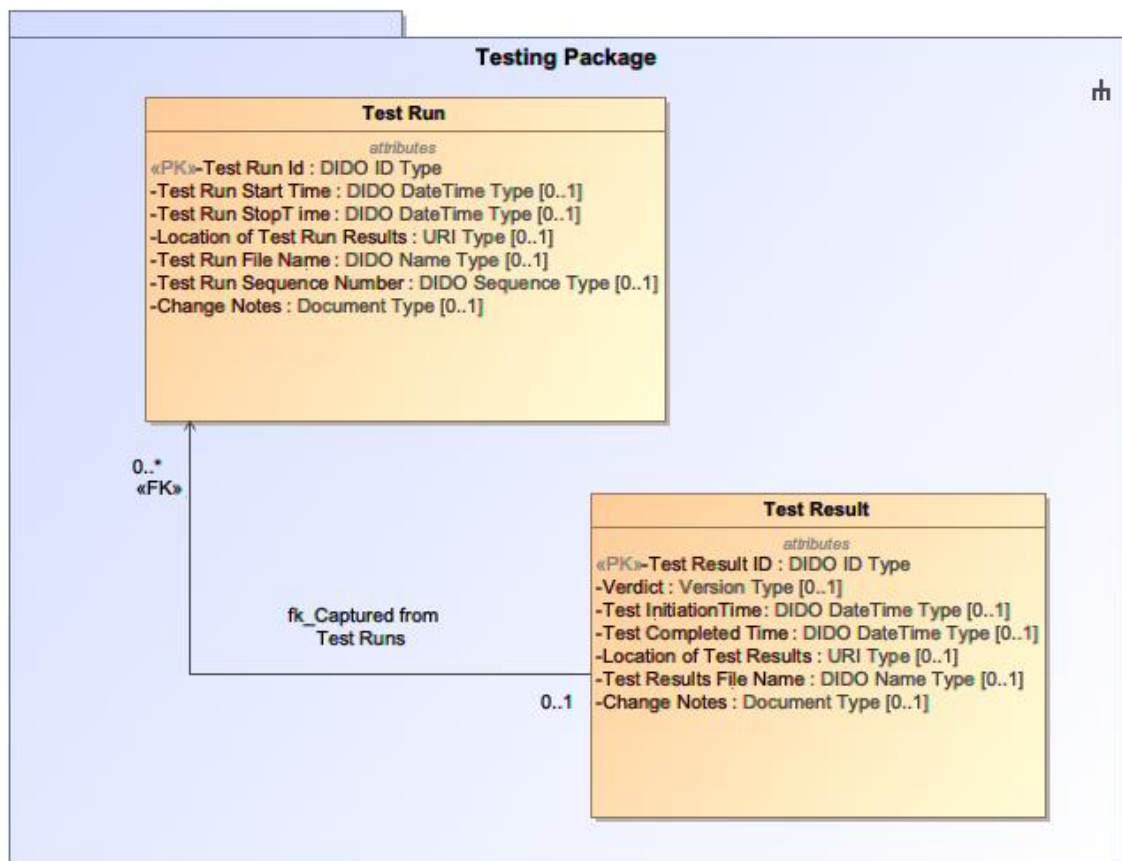


Figure 175. Test_Run

Test_Set

Distributed Immutable Data Object Data Model Dido-DM

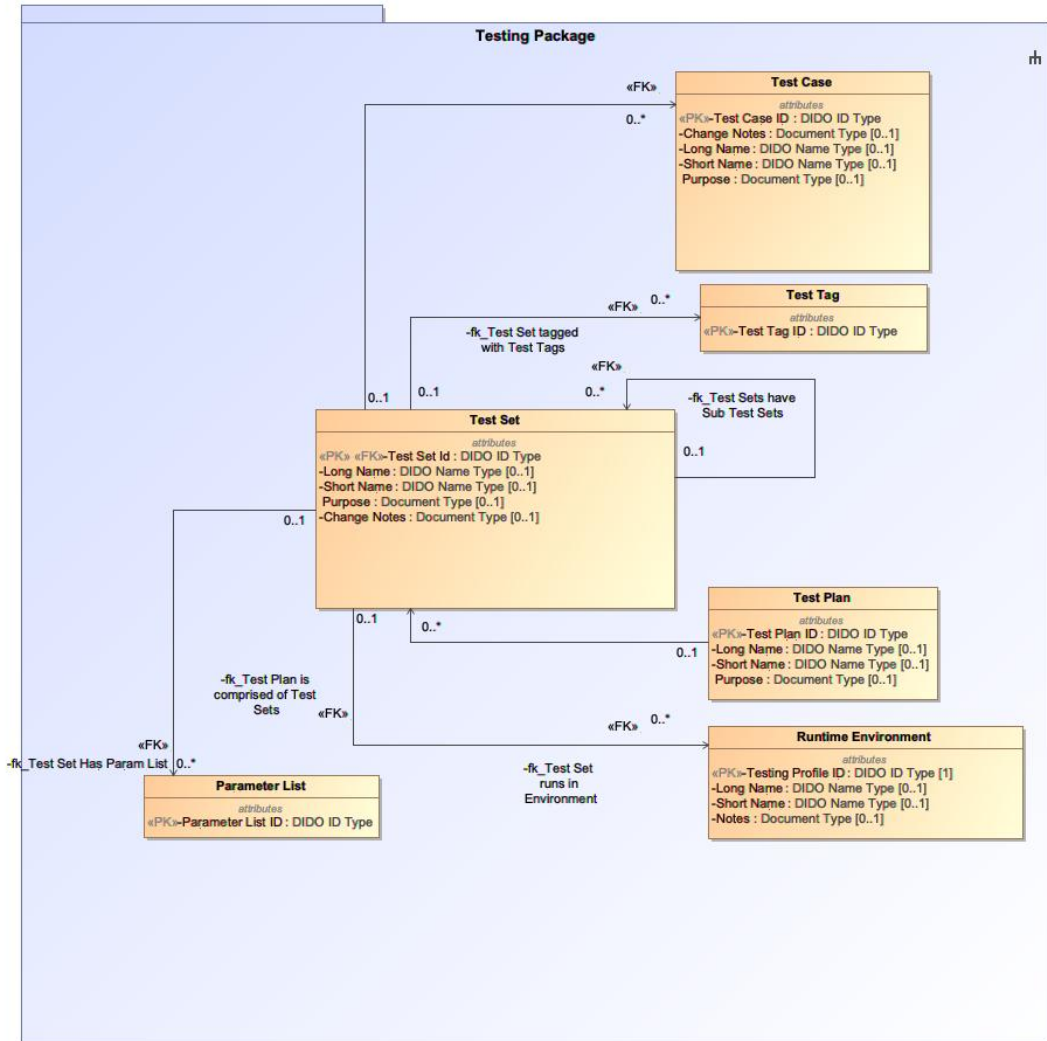


Figure 176. Test_Set

Test_Step

Distributed Immutable Data Object Data Model Dido-DM

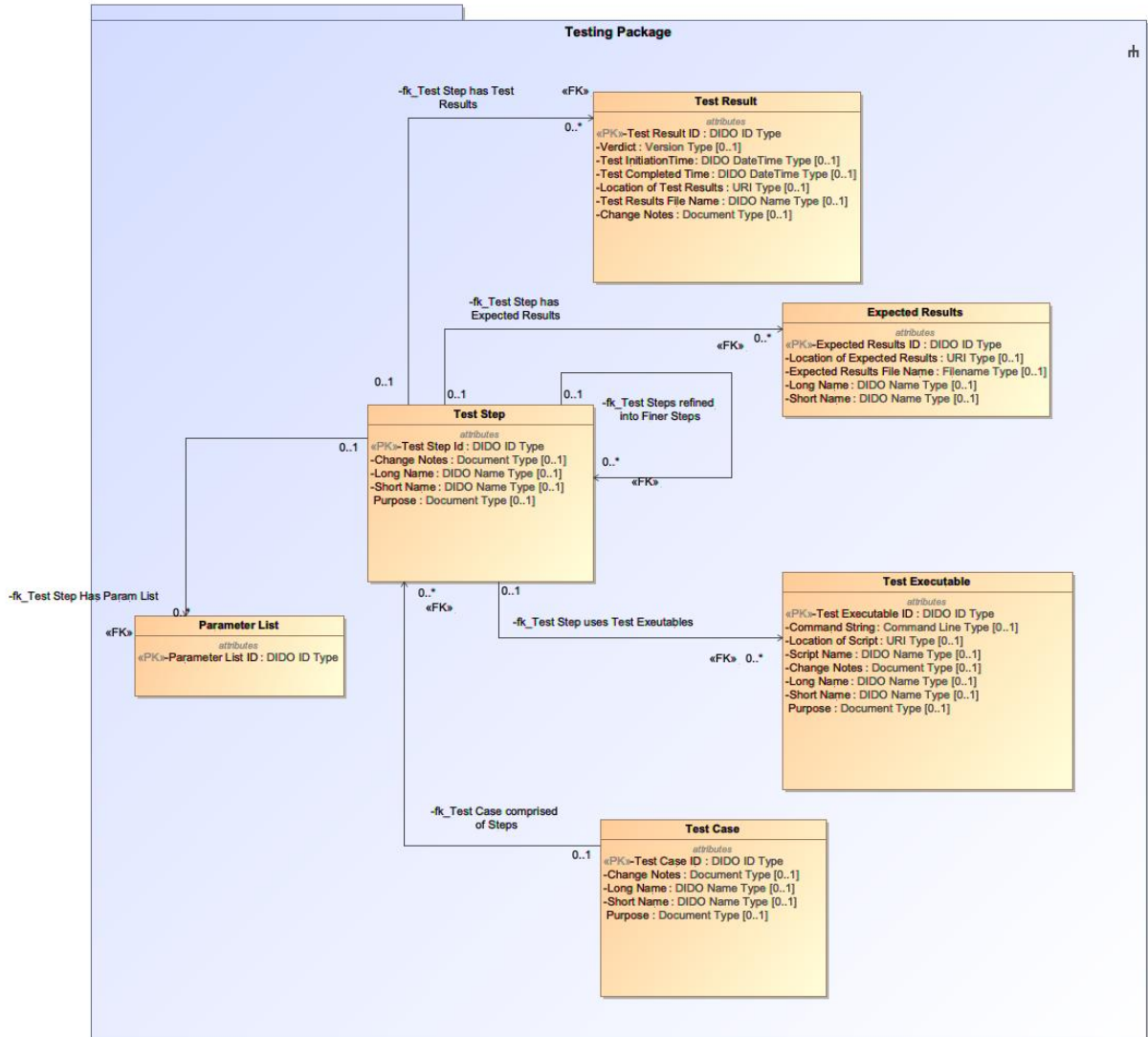


Figure 177. Test_Step

Test_Tag

Distributed Immutable Data Object Data Model Dido-DM

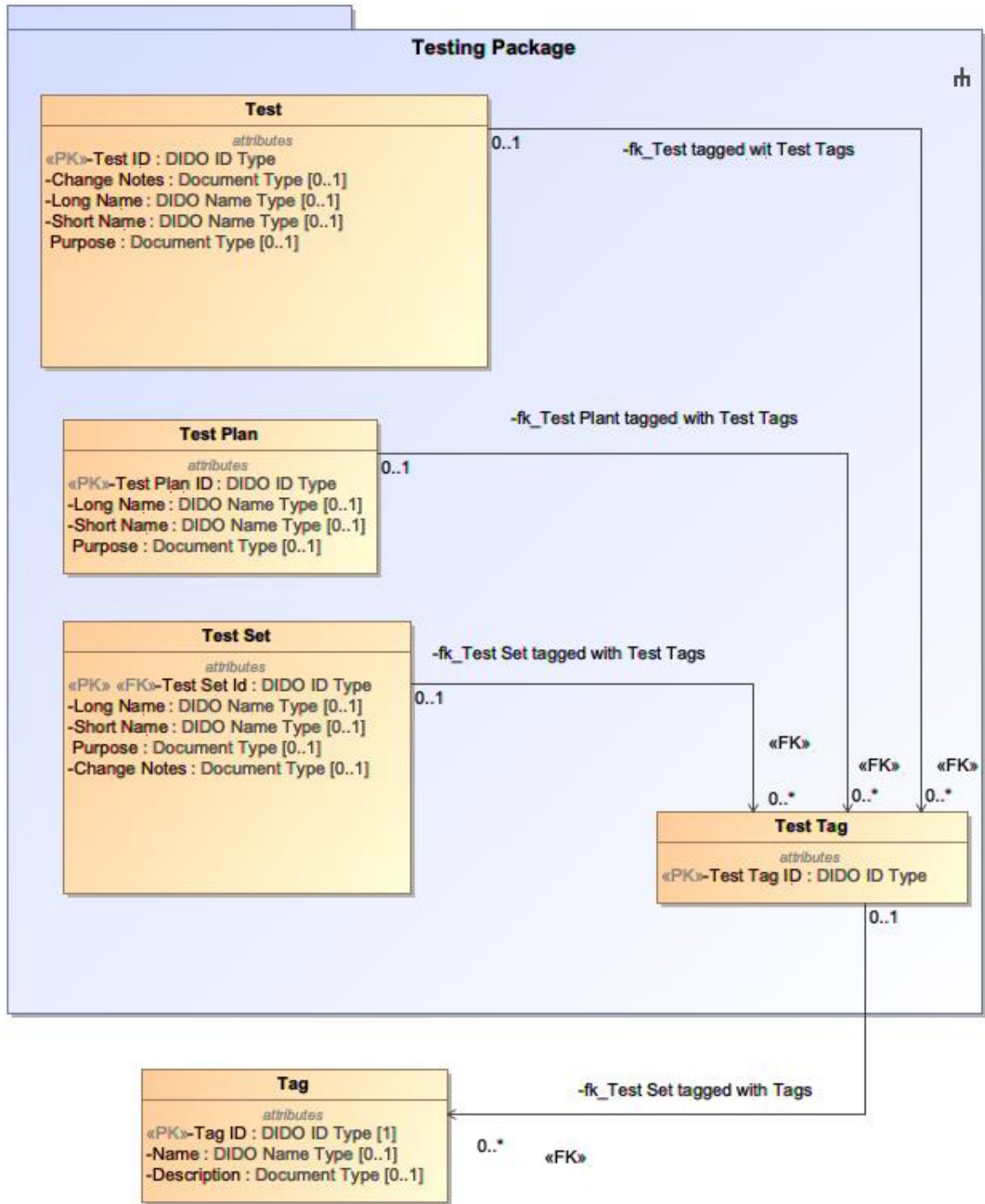
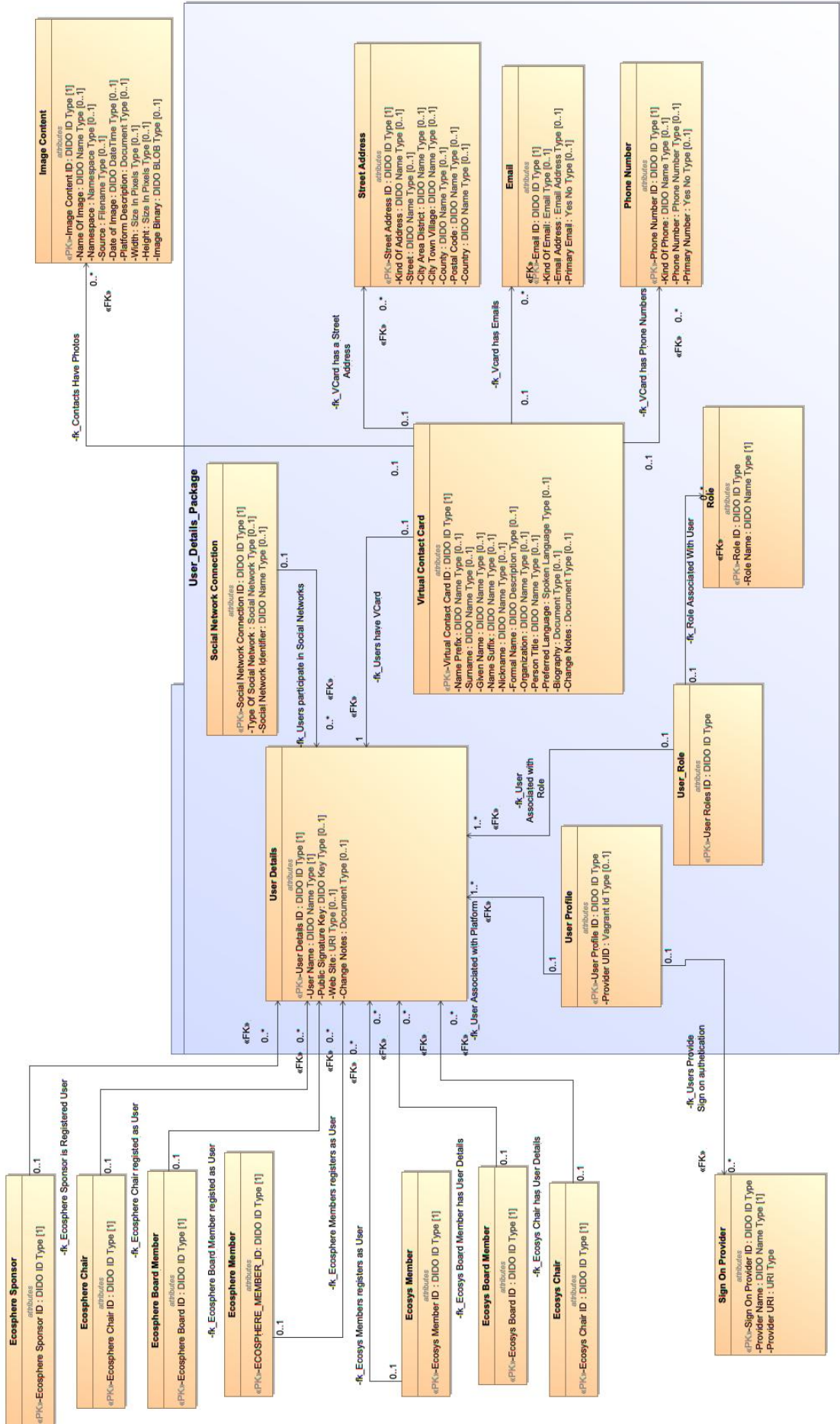


Figure 178. Test_Tag

User Detail

Distributed Immutable Data Object Data Model

Dido-DM



Distributed Immutable Data Object Data Model

Dido-DM

Figure 179. User Detail

User Profiles

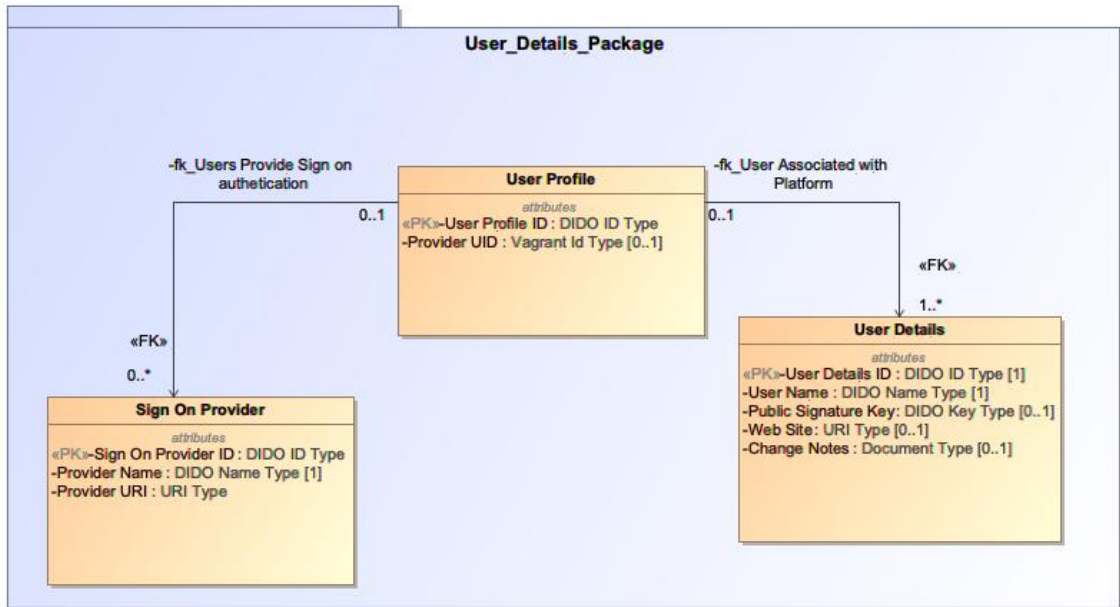


Figure 180. User Profiles

User_Roles

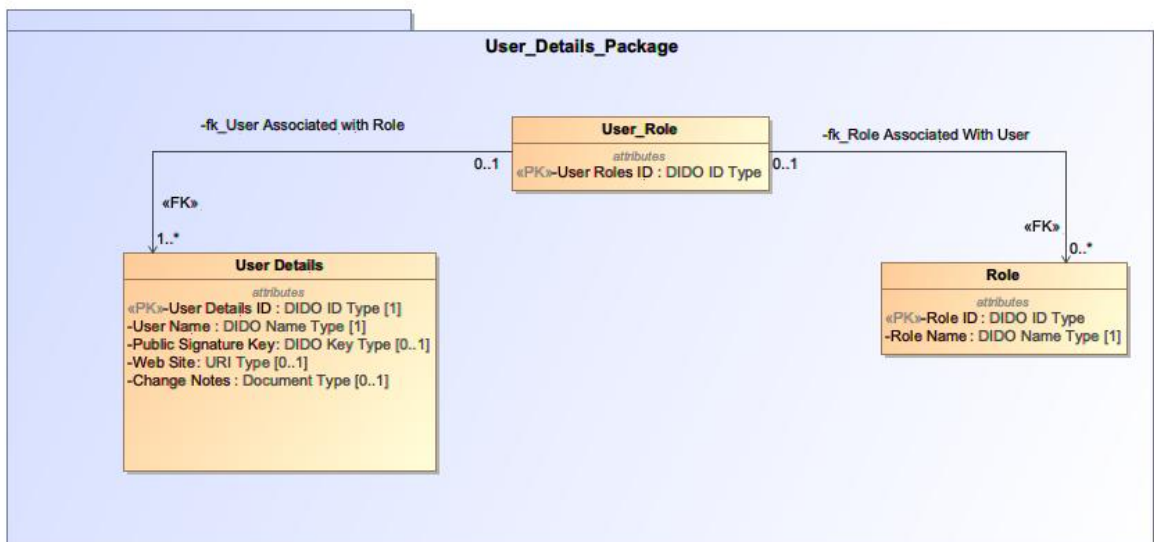


Figure 181. User_Roles

Virtual Contact Card

Distributed Immutable Data Object Data Model

Dido-DM

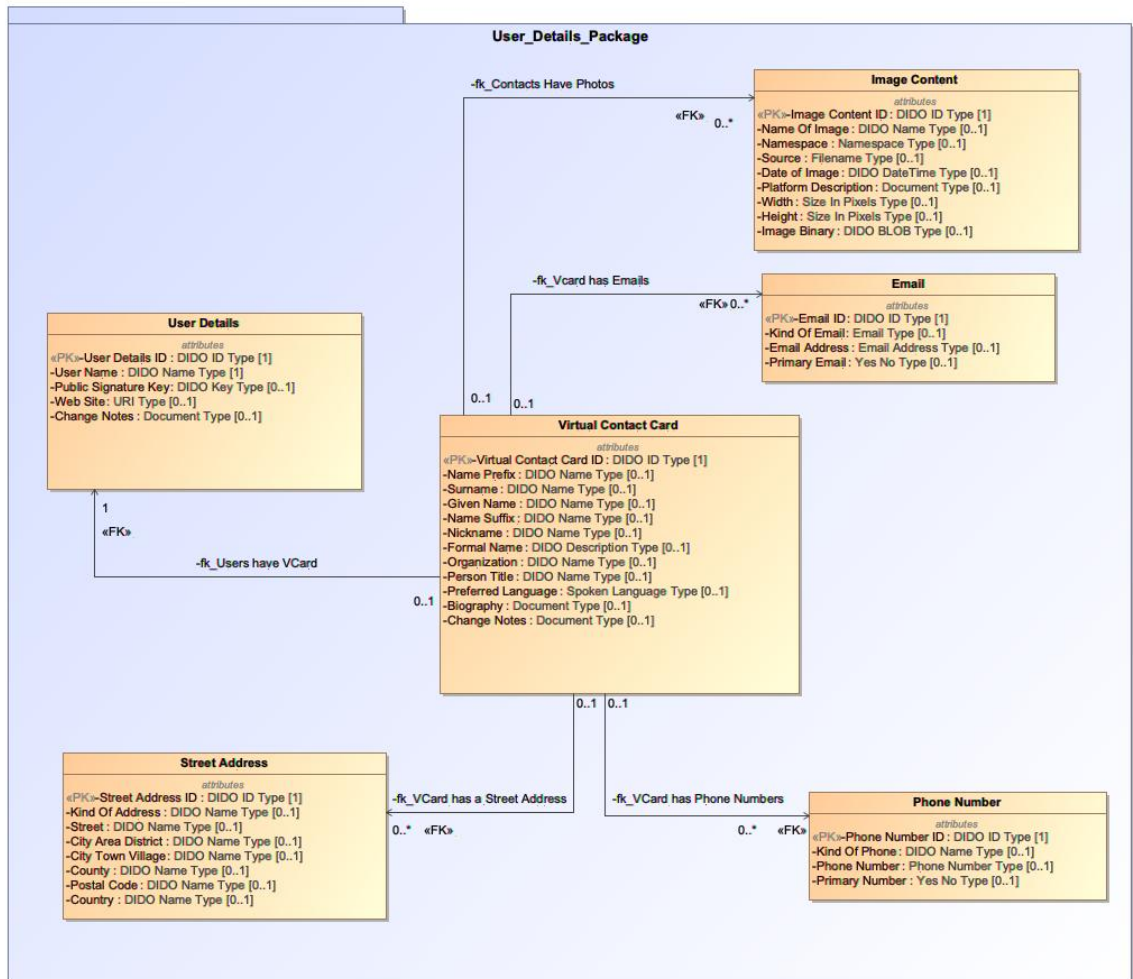


Figure 182. Virtual Contact Card