Open Game Data Exchange

v1.0a

Protocol One

Draft for public discussion

Introduction

Open Game Data Exchange is an open specification for sharing information about video games, players, and digital distribution objects. Together with other standards, it is used as a basis for the implementation of open source software for the marketplaces and distribution of video games on various platforms.

For who

The documentation is addressed to technical specialists in the distribution market of computer games and advertising:

- publishers;
- game developers;
- publishing platforms;
- advertising platforms and traffic aggregators, webmasters;

The document contains a description and specifications of an open source and secure protocol for the exchange of game data between independent participants.

Getting started

Motivation

Many developers are familiar with HTML tags and various micro markup formats that are used to add meta information to web pages. There are no specifications and standards for the exchange of meta information about games that would automate the process of adding and updating such data on a large number of platforms.

A number of platforms allow you to update and add such information using special administration panels, but most receive and update such information unsystematically — using e-mail or other means of communication. A separate platform provides an API and mechanisms to obtain such information is generally not public.

Publishers and developers perform routine work on adding and synchronizing information to various platforms, stores, portals and forums dedicated to games. Data consumers use data scraping / grabbing to obtain such information directly from web pages.

Existing schema.org, JSON-LD, RDFa structure to exchange information about the games is well documented, is simple, but can be effectively used for S2S exchange information:

- Focused on micro HTML markup.
- Do not contain all the necessary information
- Not designed for data exchange that requires localization
- Use an approach with a large number of meta information in the properties and assumptions associated with the property type.

This specification and implementation aims to create a simple, understandable and user-friendly format in which companies and people who create, sell and talk about games could automate the exchange of information between their websites and APIs.

Versioning

Open Game data Exchange (Open GDE) is developed incrementally. We use semantic versioning and approach to release new versions of the specification.

The content of each release is the result of the work of the project team located on github. All changes and additions to each version of this specification are publicly discussed by the Open GDE working group.

Roadmap

To work on Open GDE we are planning to attract as many companies and professionals working in the gaming industry as possible. Below is roadmap for the next 4 iterations of Open GDE.

Version	Date	Goals
1.0a	October 2018	Complete a basic description of the data model, entities, and their attributes. Motivational part and description of the use of objects. Description of the event model for receiving notifications. Collection and accounting of feedback from companies in Russia.
1.0b	January 2019	Implementation of schemes for JSON, Protobuf and Domain SDK in PHP, Go, JavaScript, Java. Adding the objects describing billing information. Complete description of Data Hub. Getting the feedback from technology partners in China and the USA.
1.0rc	March 2019	Adding the objects describing the integration and data exchange with SSP, and DMP platforms. Getting the feedback from technology partners. MVP for collecting information in Open GDE format for existing platforms (scrapping).
1.0r	May 2019	Release specification version 1.0.

Data model

Like schema.org and RDF Schema data model of Open GDE is a fairly simple:

- All data consists of a list of different objects organized into a multiple inheritance hierarchy where each type can be a subclass of other types.
- All objects consist of a set of properties, each of them can be a simple data type or another object. Each property has only one type.
- A property can be one or more types combined into a list or dictionary there key is a simple type.
- Simple types supported in Open GDE is string, uint, int, float.

Unlike RDF-like data schemas, Open GDE does not contain a set of abstract or base classes and properties, and does not support multiple inheritance.

The description of the objects includes properties and their values and does not contain additional prefixes and postfixes, does not describe and does not use the properties that describe meta information and/or range of values:

- *title* should be used instead of *gameTitle*
- Properties like itemtype, type, typeof are not used
- No fields describing the base object or base type.

The type hierarchy in Open GDE is simple, the data model is not universal, does not claim to be universal, and cannot be used to describe objects that are not related to the digital distribution of games. We expect this specification to be used primarily in Server-to-Server (S2S) data transfer and not as a substitute for other universal standards such as JSON-LD, Microdata, or Schema. Open GDE is not designed to be used as a basis for adding structural information to HTML pages. In the first edition, we do not aim to enrich the structured data HTML pages for indexing by search engines.

Types

The following complex types are used to describe the properties of objects in Open GDE.

Type	Description	Value
DateTime	The DateTime type describes the date (year, month, day) and optional time (hours, minutes).	ISO 8601
URL	Unicode string is the identity of the Internet resource.	RFC 1738
UUID	Unique identity in UUID V4 format	RFC 4122
Array	Contains an unordered list of base types or objects.	
Dictionary	Hash table.	

Localized properties

A large number of Open GDE properties can be represented in different languages. These are mainly text fields describing human-readable text and links to language-specific digital assets.

Any property that allows values of different languages should use the structure on the basis of the dictionary. The key in the dictionary is a string language tag written in lowercase. The value can be an arbitrary base type or an object.

The syntax of language tags was set by IETF BCP 47. BCP stands for 'Best Current Practice ' and is a permanent name for the entire RFC series, whose number changes as they are

updated. The latest RFC describing the syntax of language tags is RFC 5646(Tags for the Identification of Languages), and it is an update to RFC 4646, 3066 and 1766.

Previously, you had to search for subtitles by referring to the code lists of different ISO standards, but today you can find all the subtitles in the IANA Language Subtag Registry.

Most language tags consist of a two-or three-letter sub-tag. It is often followed by a region sub-tag consisting of two letters or three digits. RFC 5646 also allows the use of several more subtitles, if necessary.

Use the shortest possible language tag. We highly recommend using only the primary language tag. A region subtag should only be used in a situation where the same string is represented in multiple versions of the same base language.

All languages supported in an object can be listed in the base languages property of each object or in the languages property of the root object.

```
{
    "name": {
        "ru": "В а ш е  и м я",
        "en": "Your name",
        "zh-ch": "你的名字"
    }
}
```

In the description of the property containing the localizable base type should use the type alias loc<base type>, for example loc<string>.

Countries and regions

Some properties and objects within Open GDE describe values, facts, or lists associated with individual regions and countries. An example of such a property would be the location of the publisher, developer, or game release in a particular territory.

The region tag syntax is set to ISO 3166. You can find all ISO tags describing countries and regions in the ICAN Online Browsing Platform. The text two-letter code (Alpha-2 code) should be used as country tags.

Open GDE uses the “GLOBAL” tag as an exception to ISO 3166 to describe properties that imply regional segmentation but describe information that is the same for all regions.

Currency

The cost of games and goods may vary in different regions. Open the GDE operates with ISO 4217 standard for unique identification of currency and country. The full list of codes can be found on the website currency-iso.org!

Domain Layers

The exchange of information between different independent sources is a complex task with many aspects and details. To simplify the implementation, we have divided Open GDE into several layers: data transmission over the network (1), encryption and protection of transmitted data (2), serialization (3) and the data itself.

Transport layer	1	Describes the protocol for data delivery. It does not matter how and that data is transmitted, it just provides the transmission mechanism itself.
Encryption	2	Describes methods of encryption and protection of transmitted data.
Encoding	3	Describes the methods serialize and compress the data between the endpoints.
Domain	4	Describes the objects, events, and notifications, exchanged between the parties.

The each layer description is part of this specification. We consider the first three layers as the default standard that is implemented in the SDK and Open GDE schemas. In any real data exchange system they can be implemented differently. The last, 4th layer of Domain contains a description of the data and objects that should not change between different implementations of data transfer, without extending the specification and updating its version.

Transport layer

Open GDE use POST requests in the HTTP standard for transferring data between endpoints. All requests and responses must use the correct return codes in the HTTP standard.

- Requests that return content should return HTTP code 200 for all valid requests.
- Requests that do not return data (event notifications, status requests) should return HTTP return code 204.
- Incorrect requests (such as requests with an incorrect signature or checksum, or requests with an incorrect data schema) should return an HTTP return code 400 without content.

- Endpoint should return HTTP return code of 500 without content if has any temporary errors..

While making requests in Open GDE using HTTP, the version is should be passed as a special HTTP header, which makes it easier to choose the correct implementation of objects on the receiving and transmitting side. The version passed is specified as major and minor of the version using semantic versioning. For the transmission version must use the header "*x-opengde-version: 1.0*".

Encryption

HTTPS (secure HTTP) should be used to exchange data in the Open GDE format. We strongly recommend using HSTS (HTTP Strict Transport Security) if you are requesting data from a client.

End-to-End encryption (E2EE)

One of the ways to use Open GDE is to transfer data about the audience of games between different companies. Asymmetric e2ee encryption will be used if messages transferred with Data Hub feature.

This version of the document and specification does not select or describe the final encryption algorithm. This part of the specification we plan to modify to version 1.0 r.

Encoding

Representation

JSON (JavaScript Object Notation) is the default format for exchanging data. We chose JSON for its simplicity of human understanding, compactness and readiness for quick integration with any programming language or framework. Other serialization standards (MessagePack, Protobuf, Avro, etc.) can be used to optimize network bandwidth or for other reasons. Open GDE provides out-of-the-box implementations for using these formats and we strongly recommend using them to simplify integration processes between different participants.

- Requests sent using HTTP 1.1 in JSON format must contain the correct mime type specified by the Content-Type header.
- The standard value for JSON Content-Type header is "*application/json*".
- The default value for MessagePack Content-Type header is "*application/x-msgpack*".
- The standard value for Protobuf Content-Type header is "*application/x-protobuf*".

If the content-Type header is not specified in the message, the "*application/json*" format will be used.

Compression

The amount of data transferred with Open GDE can be huge, and endpoints are required to support data compression to reduce the amount and speed of data transfer over the network.

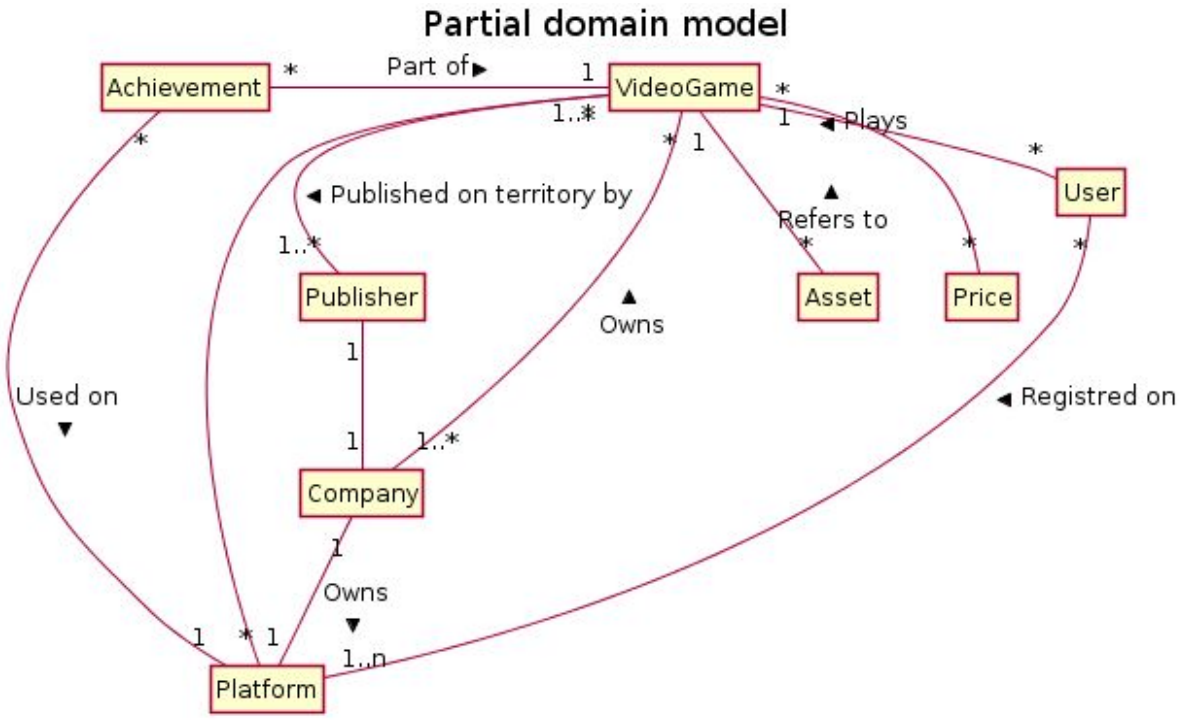
Endpoints that send HTTP 1.1 messages are required to send *gzip-compressed* data by specifying the appropriate value in the HTTP Accept-Encoding header.

Receiving hosts are required to support *gzip* and respond to requests by passing an HTTP Content-Encoding header.

Domain

The exchange of information within Open GDE is built on structured objects that describe the various objects in the ecosystem of publishing and selling games and the processes associated with it. Most of the objects in Open GDE are intended to create a universal and simple methods for the automated exchange of information.

The UML diagram below shows the basic Open GDE objects and their relationship.



Common properties

Open GDE objects can contain optional properties that are specific to any ecosystem object.

Property	Type	Description

languages	array[string]	BCP 47 compliant list of language codes that are used in localized fields.
sourceDomain	string	The ID of the company that first added the object to the Open GDE ecosystem.

Object Genre

Object describes a uniform string and the ID of the game genre.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem.
title	loc<string>	Genre title

Object Platform

The object describes the platform the game is published and distributed. Each platform has a set of properties: a unique identifier, name, site, logo in different views, etc.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem.
title	string	Platform title (common for all languages?)
url	URL	The url of the platform website.
logo	array[Image]	List of logos and platform visual identities.

Object Site

The object describing the site. Should be used for links to external sites, such as the press, localization groups, forums and communities.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem.
name	string	Site name
url	Url	Canonical site URL
icon	Image	Site logo or icon

Object Company

The object describing the company. This is the base object describing publishers, developers, vendors and any representatives of services.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
name	string	Company name

Object Publisher

The object describes the publisher of the game. A game can have one or more publishers. Publishers can be distributed to different territories: specific countries or regions. The global publisher or publisher in the territory may change over time.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
company	Company	Reference to the object of the company associated with the publisher
territory	array[string]	Region identifier according to ISO 3166 (or GLOBAL)
startDate	DateTime	The date the publisher started publishing the game
endDate	DateTime	Date by which the publisher has finished publishing the game or NULL, if it publishes up to the present moment

Object Price

The object describes the cost and the regional currency.

Property	Type	Description
price	float	Price accurate to two decimal places
currency	string	Currency code in accordance with ISO 4217

Object Image

The object describes graphical assets and their properties. It contains not only a url to the image, but also the aspect ratio of the image and some meta information like the presence of a transparent layer, the dominant color, palette, gamma. Additional data will help to choose the right context for placement on web pages and can be used to optimize marketing materials, including in the mode of automatic AB-tests.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
url	URL	Image web address
height	int	Image size in pixels vertically
width	int	Image size in pixels horizontally
aspectRatio	float	The ratio of the width to the height of an image in format: ratio:1. For example, 5:4 -> 1.25: 1, ratio = 1.25
gamma	string	Dark / light
transparent	bool	Presence of a transparent layer
dominantColor	hex	Dominant image color
palette	array[hex]	A palette of 9 primary colors of an image
tags	array[string]	Tags that describe what is pictured

Object Video

Object describing video assets. Video materials are usually placed on one of the public video hosting such as youtube, vimeo, youku. Video is characterized by duration, aspect ratio and quality.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
url	URL	Video web address
platform	string	For example, "youtube"
ratio	float	The ratio of the width to the height of a video in format: ratio:1. For example, 5:4 -> 1.25: 1, ratio = 1.25
quality	array[int]	Image quality options available: 144, 240, 360, 480,

		720, 1080, 1440, 2160 (p), etc.
length	int	Video duration in seconds
uploadDate	date	Video upload date
thumbnail	array[Image]	Thumbnail picture for video
tags	array[string]	Tags that describe what's on the video
stream	bool	True if video is game stream

Object AvailableOn

The object describes the list of platforms there the game is available. The game can be available on a large number of sites. On each of the sites in the game page has its own unique address.

Property	Type	Description
url	Url	Link to the game on platform's website
platform	Platform	Link to the platform object
price	array[Price]	List of prices on a specific platform

Object ReleaseDate

The object describes the information about the list of regions and the release date of the game in these regions. Regional division is typical for a large number of MMO games. In the case of a global release, the territory attribute can be empty or contain a single value — global.

***UNDONE**: apparently this object is superfluous, and the information about the release must be associated with a platform or publisher.*

Property	Type	Description
date	DateTime	Date of release
region	array[string]	Region identifier according to ISO 3166 (or GLOBAL)

Object Achievement

The object describes the achievements and tasks associated with the gameplay progress, style of play, finding the secrets, collectibles, etc. For a number of platforms, getting achievements unlocks new content, such as bonus maps, drawings, character skins.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
internalId	string	Internal ID of the achievement on a specific platform
index	uint16	Number in the list of achievements
name	loc<string>	Achievement name
description	loc<string>	Achievement description
score	uint16	An abstract value associated with the value or difficulty of obtaining achievement
secret	bool	Flag that determines whether the Description of achievements can be shown if it is not received
icon	Image	Achievements icon when it is not obtained
iconReceived	Image	Achievements icon when it is obtained
image	Image	Large image for achievements

Example object in JSON format:

```
{
  "uid": "8d13b6b7-f52f-41d1-b573-6bd56ec7a750",
  "internalId": "too_risky_man",
  "index": 1,
  "name": {
    "en": "Risky Man",
    "ru": "Рисковый парень"
  },
  "description": {
    "en": "Be risky all the game",
    "ru": "Рисковать на протяжении всей игры"
  },
  "score": 10,
  "secret": false,
  "icon": {
    url: "https://www.gstatic.com/webp/gallery3/1.png"
```

```
  },
  "iconReceived": {
    url: "https://www.gstatic.com/webp/gallery3/1.png"
  },
  "image": null
}
```

Object GainedAchievement

The object describes the received achievements (date and time or conditions of their receipt, etc.).

Property	Type	Description
achievement	Achievement	Link to the achievement object
createdAt	Date	Date and time of obtaining the achievement

Object VideoGame

The Central object in the open gde ecosystem, which describes information about the game and all related processes and objects.

UNDONE: Properties for

- DLC
- pre-order
- discounts
- age certificates

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
title	loc<string>	Game title
about	loc<string>	Short text about game
description	loc<string>	Long game description
slogan	loc<string>	Game slogan
releaseDate	array[ReleaseDate]	Links to ReleaseDate objects
language	array[string]	ISO 639-1 code

owners	array[Company]	Links to Company objects of companies that owns the game
followers	uint	The aggregated count of the game followers across of all platforms.
tags	array[string]	The list of unique tags with which the game is associated. For example 3d, Indie, Adventure
category	array[string]	The list of categories with which the game is associated. For example Multi-player, Single-player, Co-op, Full controller support
price	array[Price]	List of prices for the game in local currencies
subCategory	array[string]	The list of additional category tags.
genre	array[Genre]	The list of game genres
platforms	array[Platform]	The list of available platforms.
developer	Company	The developer company object.
publisher	array[Publisher]	List of publisher objects.
musicBy	Company	Associated music company object.
contentRating	array[ContentRating]	https://en.wikipedia.org/wiki/Video_game_content_rating_system
currentRating	float	The game rating.
maxRating	float	Max available rating value.
awards	array[Award]	The list of game award objects.
reviews	array[Review]	The list of game press review objects.
trailer	Video	The game official trailer.
website	Url	The game website URL.
availableOn	array[AvailableOn]	List of associated release platforms.
videos	array[Video]	List of associated video.
images	array[Image]	List of game visual identity and assets including screenshots.

achievements	array[Achievements]	Game achievement list.
sysRequirements	dict<string, SystemRequirements>	Hash table with system requirement objects. Could contains default, min and max keys.
license	URL	Link to game license
privacyPolicy	URL	Link to privacy policy
translations	dict<string, Company>	The list of translation companies working on the game. The key is the language code.

Object SystemRequirements

The object contains the hardware configuration requirements for the game launch.

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
OS	string	Identity of OS
CPU	string	Required CPU
RAM	string	Requires RAM
videoCard	string	Required video card
minResolution	string	Required minimum resolution
maxResolution	string	Max available resolution
DirectX	string	Required DirectX version
HDD	string	Required free space on hard drive
Internet	string	Required Internet speed

Object User

The object describes the basic user with a list of available platform binding information.

***UNDONE**: It is not decided how to describe meta information related to the platforms and / or the amount of data that can be requested from the platforms. How to split and merge users. This object is a draft.*

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
lastLoginAt	Date	Last login date and time
createdAt	Date	User account creation date and time
updateAt	Date	Last user data update date and time
nickname	string	User nickname
avatar	array[Image]	User avatar list

Object Award

This object describes the received awards(publications, festivals, platforms or exhibitions).

Property	Type	Description
company	Company	The object of company issuing award
title	loc<string>	Award title
description	loc<string>	Short description
image	Image	Award image

Object Review

The object describes the game reviews.

Property	Type	Description
site	Site	The object describes the site of press
title	loc<string>	Review title
description	loc<string>	Short description
author	string	Review author name
url	Url	Url to review
currentRating	float	The current rating.
maxRating	float	The highest possible value of rating

Object DLC

The object describes additional content for the game.

***UNDONE**: this object is not described completely in version 1.0a. We plan to complete itto version 1.0r.*

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
title	loc<string>	DLC title.
description	loc<string>	DLC short description.

Object Digital Key

The object describes a digital key. The activation of akey allows to issue a digital license to own the game, DLC or digital goods.

***UNDONE**: this object is a draft. We plan to complete it to version 1.0b.*

Property	Type	Description
uid	UUID	Unique identity in Open GDE ecosystem
assetId	UUID	Unique identifier of the game / DLC / digital good in the Open GDE ecosystem
streamUrl	URL	The address of the stream where the key was generated and where the activation information should be sent.
tags	array[string]	List of tags associated with the game
createDate	DateTime	Key generation date and time
restrictions	array[string]	List of available regions activation

Object Owning License

The object describes a digital license for a game, DLC, or digital goods owned by a user on a particular platform.

***UNDONE**: this object is a draft. We plan to complete it to version 1.0b.*

Property	Type	Description

uid	UUID	Unique identity in Open GDE ecosystem
assetId	UUID	Unique identifier of the game / DLC / digital good in the Open GDE ecosystem
userId	UUID	Unique identifier of the license owner.
createDate	DateTime	Date and time of license activation.

Events

Open GDE uses an asynchronous mechanism to notify you any changes with objects. Such events are used to automate the back office, update information about individual objects, change statuses.

Event describe the processes of CUD (create, update, delete). Read property alerts are not used.

- Creating new objects.
- The delete of objects.
- Change of properties.
    - also add objects to properties represented by lists or hash tables.
    - also delete values or remove objects from properties represented by lists or hashtables.

A separate event subclass describes the registration or generation of new unique object identifiers (UUIDs) there a single data hub or company acts as the principal that first created such an object or identifier. Such a member is described by the sourceDomain property.

The HTTP 1.1 and the approaches described in the Domain Layer of this document are used to send events.

In most cases, the action that causes the notification is the result of the user's purchase of the game, change of data by the developer/publisher or platform/store.

As part of the Open GDE project, there are basic implementations in the main programming language for both the objects and the service for receiving alerts.

Subscription

A endpoint that wants to receive notifications about changes in information within the Open GDE must first register its RSA key on the distribution side and explicitly specify a list of alerts and objects that it is interested in changing.

The current version of the document does not contain a detailed description of the subscription and publication of changes and should be finalized later.

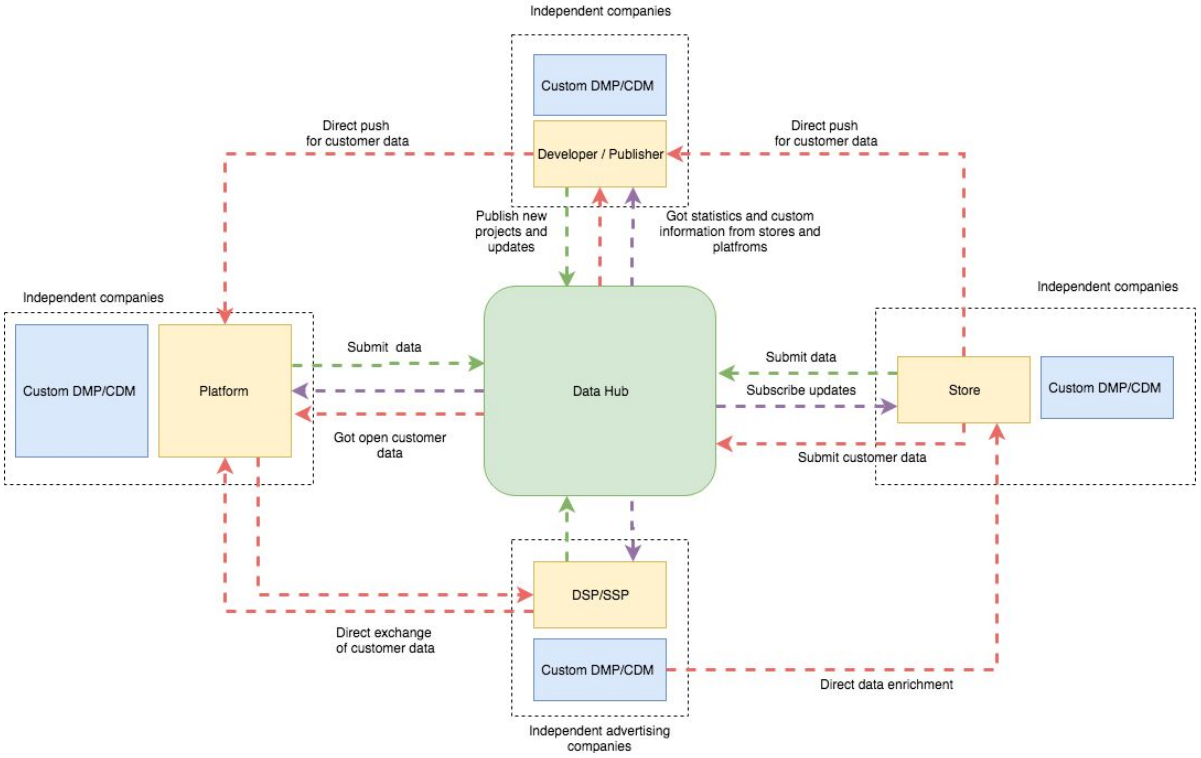
Delivery guarantee

The Internet is not a reliable means of delivery and receiving data.Some messages may be lost or late delivered. Send events implied the availability of services re notification as long as the receiving party does not confirm receipt.

Open GDE uses HTTP response codes to indicate a successful or unsuccessful request.

Data Hub

The Open GDE Data Hub is a separate non-profit organization and ecosystem member whose deployment and operation is handled by the Protocol One team.



Data Hub responsible for

- Generate and/or register new unique object identifiers (UUIDs).
- Registration and distribution of notifications and events among members of the network.
- Help to exchange and validate data in Open GDE ecosystem between multiple participants.

- Enrich the data about the end users without disclosing personal information and information identifying the user on the target platforms.
- Backup storage of UUID and Open GDE data/schemas.

The key goal of Data Hub is to act as an independent intermediary for creation and secure storage of information common to all participants of the ecosystem.

Disclaimer

This document does not constitute an offer to invest in or implement any other type of material support for the Protocol One or Open Game Data Exchange project. The main purpose of this document is a detailed and comprehensive description of the Open GDE specification, on the basis of which the digital distribution platforms of PC video games, developers and publishers of video games can be integrated.

Statements and other information declarations contained in this document should not be interpreted as direct representations or promises except in cases where it expressly States.