

2.3.4.8 Object Data Taxonomy

[Return to Data Taxonomy](#)

Overview

[Return to Top](#)

In programming, an [Object](#) represents a real-world object such as a car, an employee or a task. Ultimately, as the real-world Object is defined in terms of software, it is translated to a [Data Structure](#). The Object Data (see Figure 1) is categorized into Characteristic Data, Definition Data, Field Data and Operation Data.

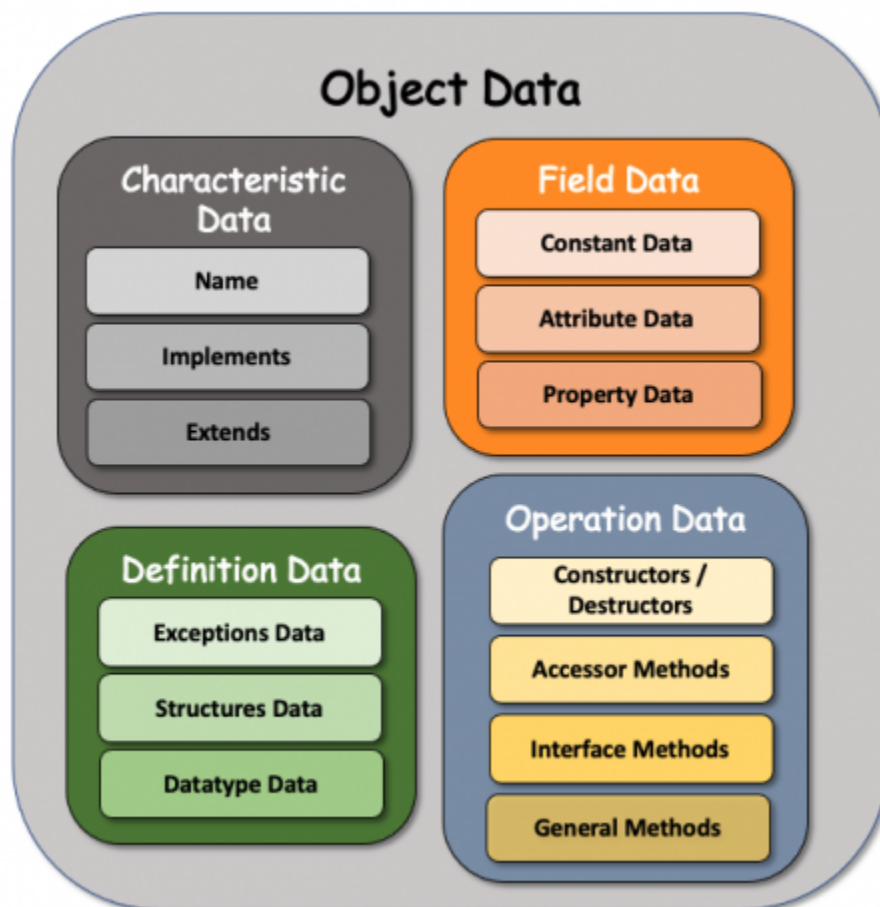


Figure 1: Components of an Object

DIDO Specifics

[Return to Top](#)

Many [DIDO Platforms](#) refer to Object Data as **Smart Contracts**. These are similar too, but not the same as classes in [Object-Oriented Programming \(OOP\)](#) languages such as C++, Java, C#, Python, etc. In many ways, the differences are lexical in nature (i.e., **Smart Contract** versus **Contract**, **method** versus **function**, etc. However, there are some important differences which have to do with the distributed nature of DIDOs and the immutability of the field data (i.e., **view** and **pure** designators on **methods**).

The following discussion has to do with the more generic concept of **Object Data** but does include discussions of how it relates to DIDO Platforms in general and [Ethereum's Solidity](#) more specifically.

- [2.3.4.8.1 Characteristic Data](#)
- [2.3.4.8.2 Definition Data](#)
- [2.3.4.8.3 Field Data](#)
- [2.3.4.8.4 Operation Data](#)

From:
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1.2_views:3_taxonomic:4_data_tax:08_objects:start&rev=1644114056

Last update: **2022/02/05 21:20**

