2.3.4.3.2 Ancillary Data

return to Data Taxonomy

Ancillary data is adjunct information used in the formulation of transactions. It differs from transaction data because it is not used to define the operation described within a transaction. Probably the best way to explain ancillary data is by using examples.

Examples of Ancillary Data

Supporting Data

> Data needs to be exchanged between two individuals that describe the exchange of one kind of currency with another (e.g., US dollars to EU euros or Bitcoins). The transaction can simply be something like this:

```
TRANSFER 500 USD to 1 Bitcoin in MyAccount
```

> The transaction seems legitimate and can probably be verified easily enough, or can it? In order to verify the transaction, the exchange rate used also needs to be provided. It might also require the timestamp of when the exchange rate was calculated and even where the exchange rate was obtained from. This “extra” information used to verify the transaction is called ancillary data. It ultimately needs to be coded into the transaction. A classic example of why ancillary data needs to be captured is to avoid salami slicing.

Business Process Management

> Data needs to be exchanged between two different business entities. Each business entity has its own business process describing the steps needed on its side in order to approve a transaction. The data required to approve the transaction is *transformational* in nature rather than *transactional*.

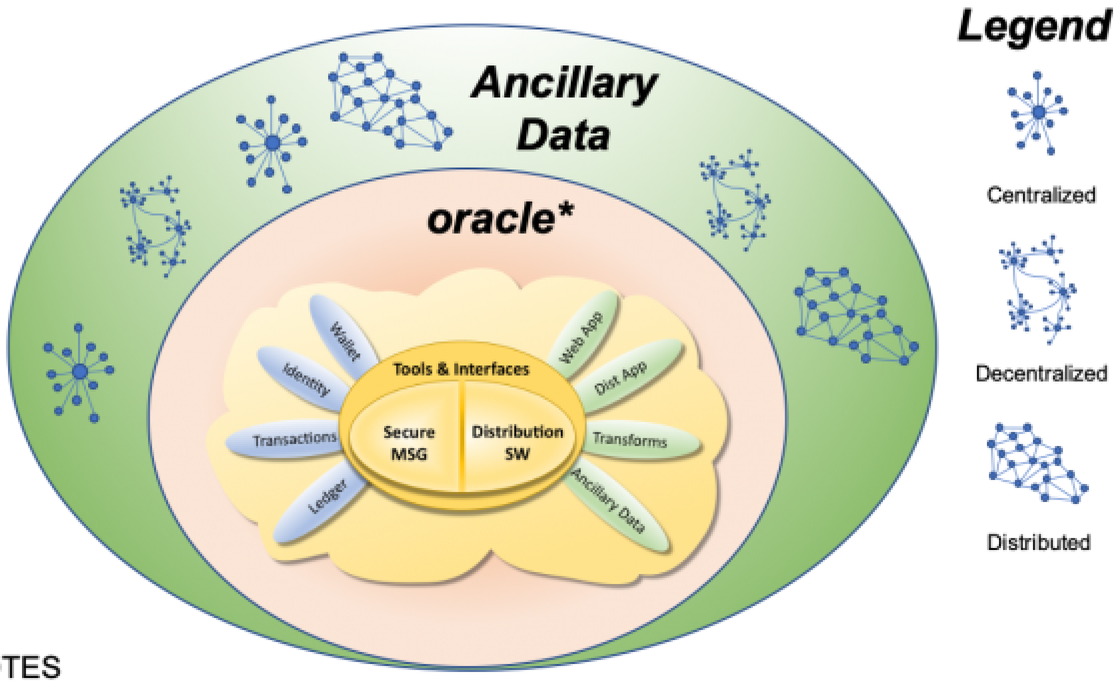
```
TRANSFER 200 AMZN Shares from MyOrg to YourOrg
```

> At a transactional level, all that is needed is verification that MyOrg has the shares, and YourOrg exists in order to prevent “double spend” fraud, i.e., spending the same money more than once. However, many organizations has business processes in place that require signatures from various people that must be in a prescribed sequence (i.e., Director signs, VP signs, Comptroller signs). To verify the transaction, the signatory data needs to be provided and confirmed that it is valid. This

can occur outside the transaction and be transformational in nature. Transformational data can be undone, deleted, or modified until the business process commits to the transaction.

DIDO Implementation of Ancillary Data

Some DIDO implementations may provide direct support of ancillary data within the DIDO or they may provide access to external ancillary data through oracles. However, if ancillary data is not also implemented within the DIDO, but implemented externally with and accessed through the use of oracles, many of the benefits of the distributed architecture are lost because access to the ancillary data can become a bottleneck.



NOTES

* An oracle, in the context of blockchains and smart contracts, is an agent that finds and verifies real-world occurrences and submits this information to a blockchain to be used by smart contracts. https://blockchainhub.net/blockchain-oracles/

Standards

Technical Standards

- None at this time

de facto Standards

- None at this time

Tools

- None at this time

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:03_digi_taxonomy:2_ancillary

Last update: **2022/05/27 19:53**