GETTING STARTED WITH THE MISMO API TOOLKIT - DEFINING RELATIONSHIPS

This article builds on the use case introduced in the GETTING STARTED WITH THE MISMO API TOOLKIT article and illustrates how to define relationships between API objects using the MISMO API Toolkit. The Implementation Guide specifies two ways to define relationships between objects: containment for hierarchical relationships and links for relationships between independent objects. We will demonstrate how to select the appropriate approach and define the schema for linking API objects.

The Use Case

We will create a schema for the following use case:

"As a product owner, I need to enable external clients to retrieve asset information and identify the borrower(s) associated with each asset via a REST API."

Designing the schema

For this exercise, the schema will support assets that are jointly owned by multiple borrowers. This is a many-to-many relationship, because a borrower can own multiple assets and an asset can be owned by multiple borrowers. The following attributes are required for borrowers and assets:

- Borrower – First name, middle name, last name, birth date, and type
- Asset – Identifier, type, and value

Step 1 – Identify the core entities: The YAML files containing the required datapoints can be identified by reviewing the Logical Data Dictionary or searching the contents of the YAML files provided in the API Toolkit. In this case, Name.yaml, Borrower.yaml, and Asset.yaml contain all the necessary datapoints. The following paths apply for these API objects based on the current reference model as defined by the MISMO logical data model:

```
Message.dealSets.deals.party.individual.name
Message.dealSets.deals.party.roles[].borrower
Message.dealSets.deals.assets
```

Step 2 – Determine the approach for associating assets and borrowers: Asset and Party are independent objects. Based on the paths listed above Asset and Party are sibling objects, and the common parent is **Deals**. Therefore, containment cannot be used to define this relationship if the API will comply with the MISMO model. We will follow the linking approach to associate borrowers and assets as we build the initial schema. Then, we will provide samples of alternate approaches that align with MISMO terms, but do not adhere to the MISMO model.

Step 3 – Build the schema based on the MISMO Model and Flattening Rules: The API Toolkit allows the root object for an API to start at any level within the MISMO model hierarchy. So, **Message** and **DealSets** can be eliminated from the hierarchy. There are multiple paths to **Name** in the MISMO model, so the hierarchy will be retained to eliminate ambiguity (Single-Path Rule). **Roles** is plural, so an array will be used to retain cardinality (Cardinality Rule). These flattening rules result in the following paths:

```
deals.party.individual.name
deals.party.roles[].borrower
deals.assets
```

Therefore, the complete schema will be based on the following YAML files: Deal.yaml, Party.yaml, Individual.yaml, Name.yaml, Roles.yaml, Borrower.yaml, and Asset.yaml.

Note: The MISMO API Toolkit adheres to the OpenAPI 3.0 Specification (OAS 3.0) for API definitions, therefore all examples in this article follow OAS 3.0.

```yaml
components:
  schemas:
    Deal:
      type: object
      properties:
        id:
          $ref: "#/components/schemas/MISMOString"
        parties:
          type: array
          items:
            $ref: "#/components/schemas/Party"
        assets:
          type: array
          items:
            $ref: "#/components/schemas/Asset"
      required:
      -parties
```

Deal.yaml

The root object is the common parent **Deal.**

```yaml
    Party:
     type: object
     properties:
        id:
          $ref: "#/components/schemas/MISMOString"
        roles:
          type: array
          items:
            $ref: "#/components/schemas/Role"
        individual:
          $ref: "#/components/schemas/Individual"
        links:
          type: object
```

Party.yaml

The **id** will be specified in the link.

The **links** will identify the assets the party owns.

```yaml
    Role:
      type: object
      properties:
        borrower:
          $ref: "#/components/schemas/Borrower"
```

Role.yaml

```yaml
    Borrower:
      type: object
      properties:
        borrowerBirthDate:
          $ref: "#/components/schemas/MISMODate"
        borrowerClassificationType:
          allOf:
          - $ref: "#/components/schemas/MISMOEnum"
```

```
      type: string
      enum:
      - Primary
      - Secondary
```

Borrower.yaml

```
  Individual:
    type: object
    properties:
      name:
        $ref: "#/components/schemas/Name"
```

Individual.yaml

```
  Name:
    type: object
    properties:
      firstName:
        $ref: "#/components/schemas/MISMOString"
      lastName:
        $ref: "#/components/schemas/MISMOString"
      middleName:
        $ref: "#/components/schemas/MISMOString
```

Name.yaml

```
Asset:

   type: object
   properties:
     id:
       $ref: "#/components/schemas/MISMOString"
     assetCashOrMarketValueAmount:
       $ref: " #/components/schemas/MISMOAmount"
     assetType:
       allOf:
       - $ref: "#/components/schemas/MISMOEnum"
       - type: string
         enum:
         - Annuity
         - CheckingAccount
         - MoneyMarketFund
         - MutualFund
         - SavingsAccount
         - Stock
     links:
       type: object
```

Asset.yaml

The **id** will be specified in the link.

Enum values are reduced to simplify the example.

The **links** will identify the party that owns the asset.

Sample JSON Document – Based on the MISMO Model

In this sample, **borrower** and **asset** appear at the same level to align with the MISMO model. Both parties are linked to asset 11189. For illustrative purposes, both **Parties** and **Assets** contain **links** to specify the many-to-many relationship. However, implementors can determine where to define **links** based on the use case to eliminate any undesired redundancy.

```
{
  "deal":
  {
   "parties": [
   {
      "id": "123456",
      "roles": [
      {
         "borrower":
         {
            "borrowerBirthDate": "1979-01-01",
            "borrowerClassificationType": "Primary"
         }
      }],
      "individual":
      {
         "name":
         {
            "firstName": "John",
            "lastName": "Cox",
            "middleName": "Henry"
         }
      },
      "links": {
          "self": "http://example.org/borrowers/123456",
          "ownsAsset1": "http://example.org/asset/11189",
          "ownsAsset2": "http://example.org/asset/22278"
      }
   },
   {
      "id": "234567"
      "roles": [
      {
         "borrower":
         {
            "borrowerBirthDate": "1980-11-11",
            "borrowerClassificationType": "Secondary"
         }
      }],
      "individual":
      {
         "name":
         {
            "firstName": "Jane",
            "lastName": "Cox"
         }
      },
      "links": {
          "self": "http://example.org/borrowers/234567",
          "ownsAsset1": "http://example.org/asset/11189",
          "ownsAsset2": "http://example.org/asset/33378"
      }
   }],
   "assets": [
      {
        "id": "11189",
        "assetType": "SavingsAccount",
        "assetCashOrMarketValueAmount": 50000,
        "links": {
          "self": "http://example.org/asset/11189",
          "owner1": "http://example.org/party/123456",
          "owner2": "http://example.org/party/234567"
       }
      },
      {
        "id": "22278",
        "assetType": "MoneyMarketFund",
        "assetCashOrMarketValueAmount": 1500,
        "links": {
            "self": "http://example.org/asset/22278",
            "owner1": "http://example.org/party/123456"
```

The format for **links** is based on the JSON:API specification.

The key for the **link** property is the appropriate arcrole as defined in the "Arcroles" section of the LDD.

```
        }
      },
      {
        "id": "33378",
        "assetType": "CertificateOfDepositTimeDeposit",
        "assetCashOrMarketValueAmount": 20000,
        "links": {
            "self": "http://example.org/asset/33378",
            "owner1": "http://example.org/party/234567"
        }
      }
    ]
  }
 }
}
```

Sample JSON Document – Leveraging MISMO Terms and Links

If a flatter structure is desired, the schema can leverage the MISMO vocabulary (MISMO terms and definitions) and eliminate the MISMO hierarchy. This produces a simpler schema, but also reduces interoperability. So, implementors need to evaluate the tradeoffs and determine the required level of interoperability.

The following sample illustrates this approach by merging the data points from Borrower, Name, and Party into a single object. Borrowers and Assets are both at the same level. This supports the many-to-many relationship between borrowers and assets.

Party, Roles, Individual and **Name** are eliminated from the hierarchy.

```
{
  "deal":
    {
      "borrowers": [
        {
            "id": "123456",
            "firstName": "John",
            "lastName": "Cox",
            "middleName": "Henry",
            "borrowerBirthDate": "1979-01-01",
            "borrowerClassificationType": "Primary",
            "links": {
              "self": "http://example.org/borrowers/123456",
              "ownsAsset1": "http://example.org/asset/11189",
              "ownsAsset2": "http://example.org/asset/22278"
            }
        },
        {
            "id": "234567",
            "firstName": "Jane",
            "lastName": "Cox",
            "borrowerBirthDate": "1980-01-01",
            "borrowerClassificationType": "Secondary",
            "links": {
              "self": "http://example.org/borrowers/234567",
              "ownsAsset1": "http://example.org/asset/11189",
              "ownsAsset2": "http://example.org/asset/33378"
            }
        }
      ],
    "assets": [
      {
          "id": "11189",
          "assetType": "SavingsAccount",
          "assetCashOrMarketValueAmount": 50000
          "links": {
           "self": "http://example.org/asset/11189",
           "owner1": "http://example.org/party/123456",
           "owner2": "http://example.org/party/234567"
        }
      },
      {
          "id": "22278",
          "assetType": "MoneyMarketFund",
          "assetCashOrMarketValueAmount": 1500,
          "links": {
            "self": "http://example.org/asset/22278",
```

```
            "owner1": "http://example.org/party/123456"
          }
        },
        {
            "id": "33378",
            "assetType": "CertificateOfDepositTimeDeposit",
            "assetCashOrMarketValueAmount": 20000
            "links": {
              "self": "http://example.org/asset/33378",
              "owner1": "http://example.org/party/234567"
          }
        }
      ]
    }
```

Sample JSON Document – Leveraging MISMO Terms and Containment

Containment can also be used to represent the relationship between borrowers and assets if the API will not align to the MISMO model. However, this produces duplication as illustrated in the example below. Implementation teams need to assess the options for representing relationships and determine the best approach based on their use case and interoperability requirements.

```
{
  "deal":
    {
      "borrowers": [
        {
          "id": "123456",
          "firstName": "John",
           "lastName": "Cox",
           "middleName": "Henry",
           "borrowerBirthDate": "1979-01-01",
           "borrowerClassificationType": "Primary",
           "assets": [
             {
               "id": "11189",
               "assetType": "SavingsAccount",
               "assetCashOrMarketValueAmount": 50000
             },
             {
               "id": "22278",
               "assetType": "MoneyMarketFund",
               "assetCashOrMarketValueAmount": 1500
             },
           ]
        },
        {
          "id": "234567",
          "firstName": "Jane",
           "lastName": "Cox",
           "borrowerBirthDate": "1980-01-01",
           "borrowerClassificationType": "Secondary",
           "assets": [
             {
               "id": "11189",
               "assetType": "SavingsAccount",
               "assetCashOrMarketValueAmount": 50000
             },
             {
               "id": "33378",
               "assetType": "CertificateOfDepositTimeDeposit",
               "assetCashOrMarketValueAmount": 20000
             }
           ]
        }
      ]
  }
```

Asset 11189 is duplicated.

The API Toolkit provides a fast, efficient way to develop MISMO aligned APIs resulting in shorter time-to-market and improved interoperability. The examples above can be expanded to suit any use case your entity might have. If you have any questions or comments, the API Community of Practice meets the 1st and 3rd Thursday of each month. We would love to hear from you.

Note: MISMO's API Toolkit is available to its members free of charge. Contact us at info@mismo.org to learn about the benefits of MISMO membership.