

Smart Domain Architecture Pattern

Re-engineering DDD

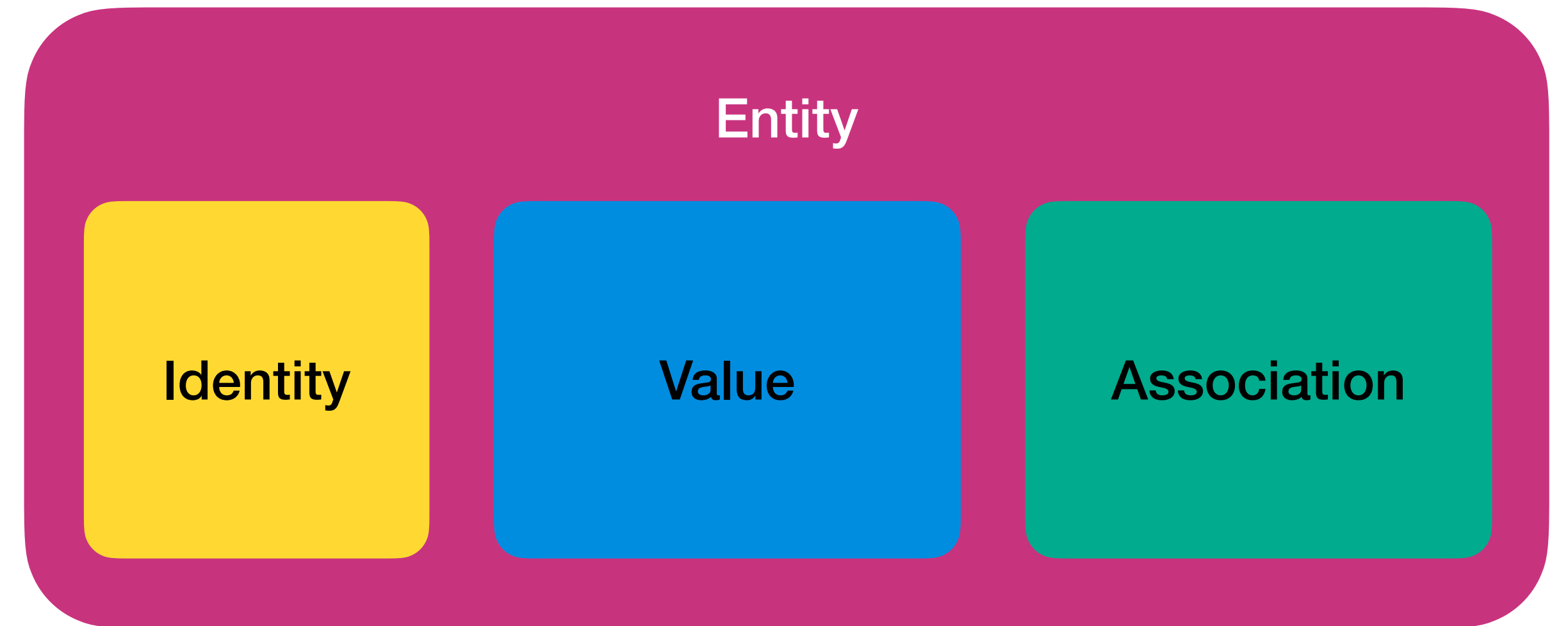
Xu Hao

Smart Domain

- Object-Oriented Approach (No service)
- Model as fully connected object graph (Not disconnected aggregation)
- Directly map to RESTful API(with HATEOAS)
- Act as an abstraction layer to hide implementation details
- Consistent between conceptual model/model/api

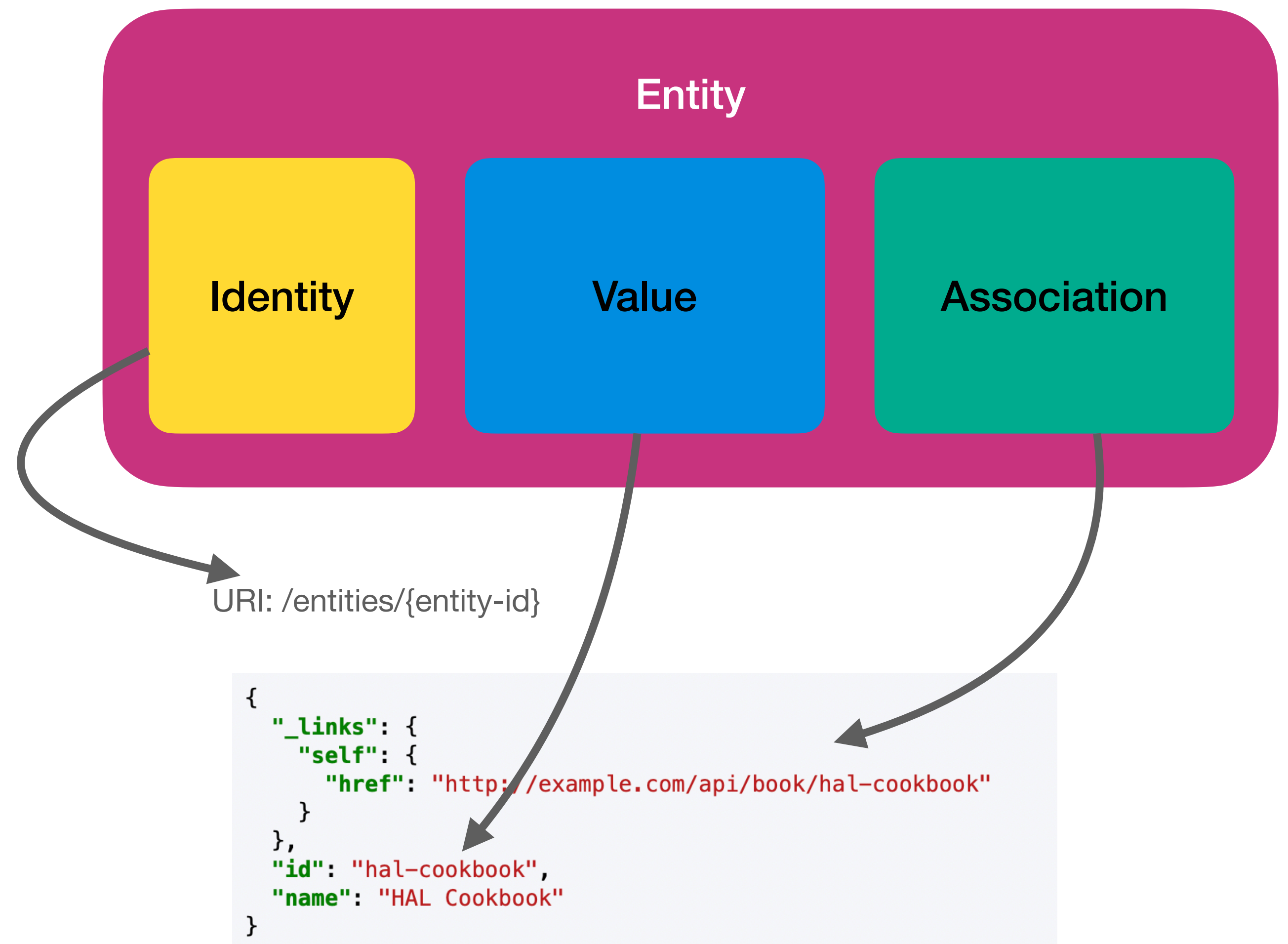
Smart Domain

- Entity is not about value, but about identity and association



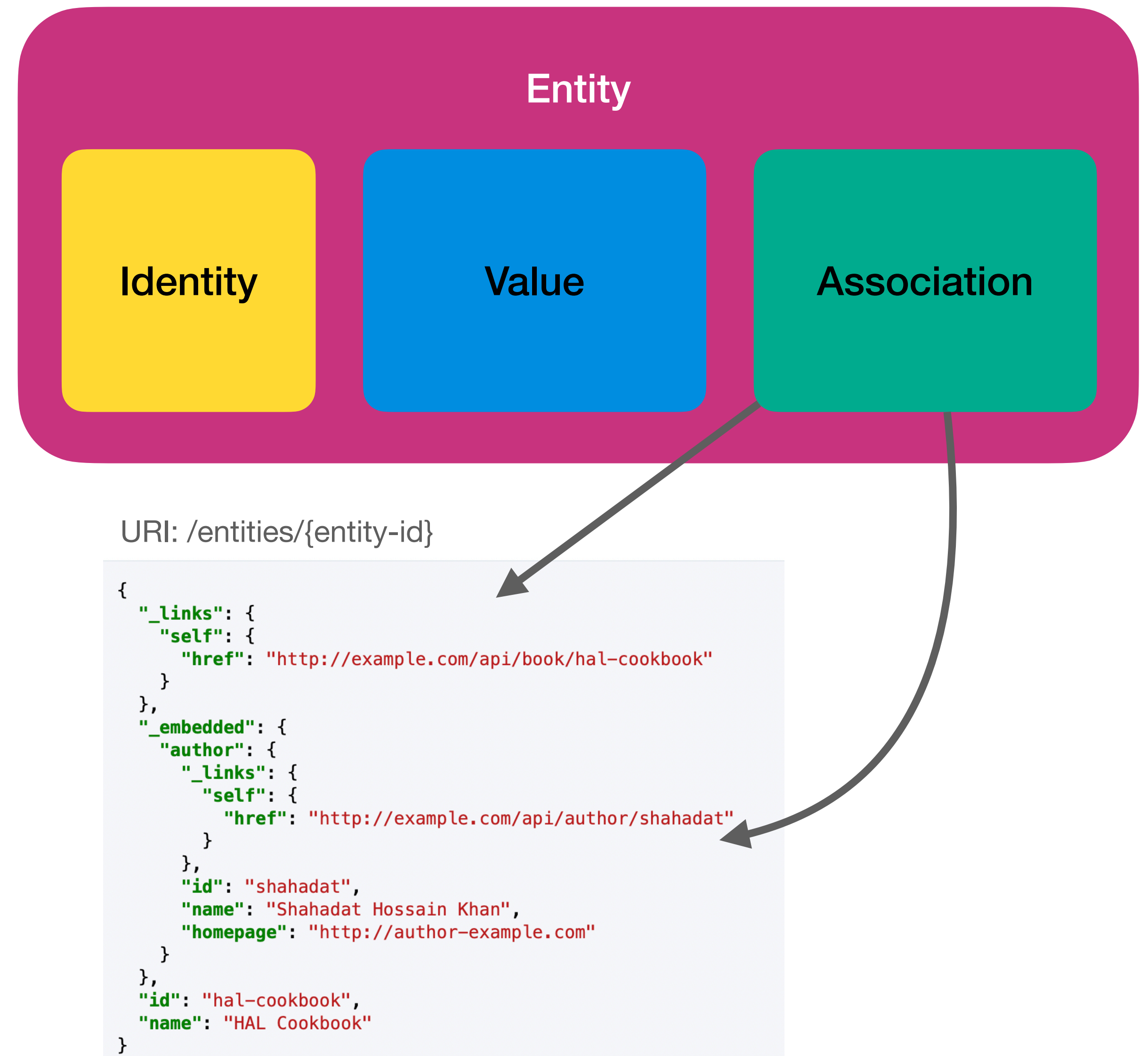
Smart Domain

- Entity is not about value, but about identity and association
- Can be mapped easily as a lossless representation model



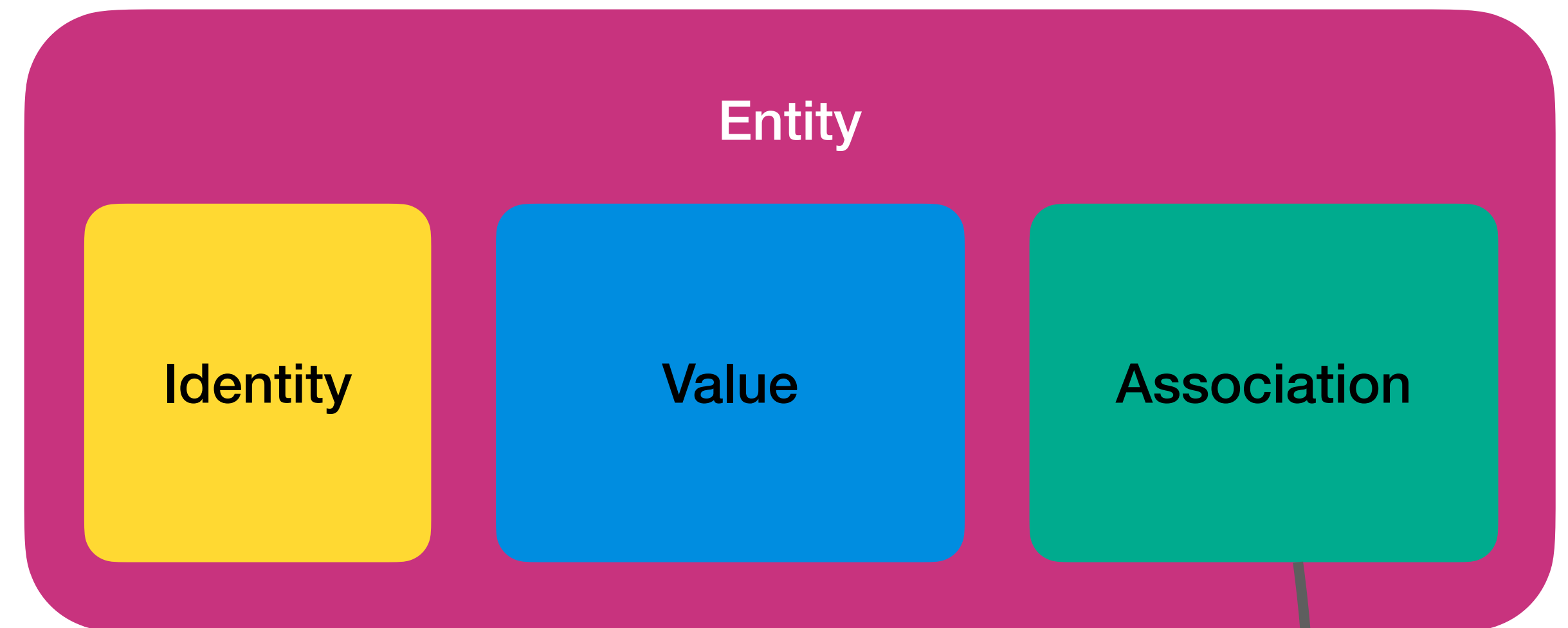
Smart Domain

- Entity is not about value, but about identity and association
- Can be mapped easily as a lossless representation model
- Association is an abstraction mechanism for connections



Smart Domain

- Entity is not about value, but about identity and association
- Can be mapped easily as a lossless representation model
- Association is an abstraction mechanism for connections
- Association is an abstraction mechanism for lifecycle

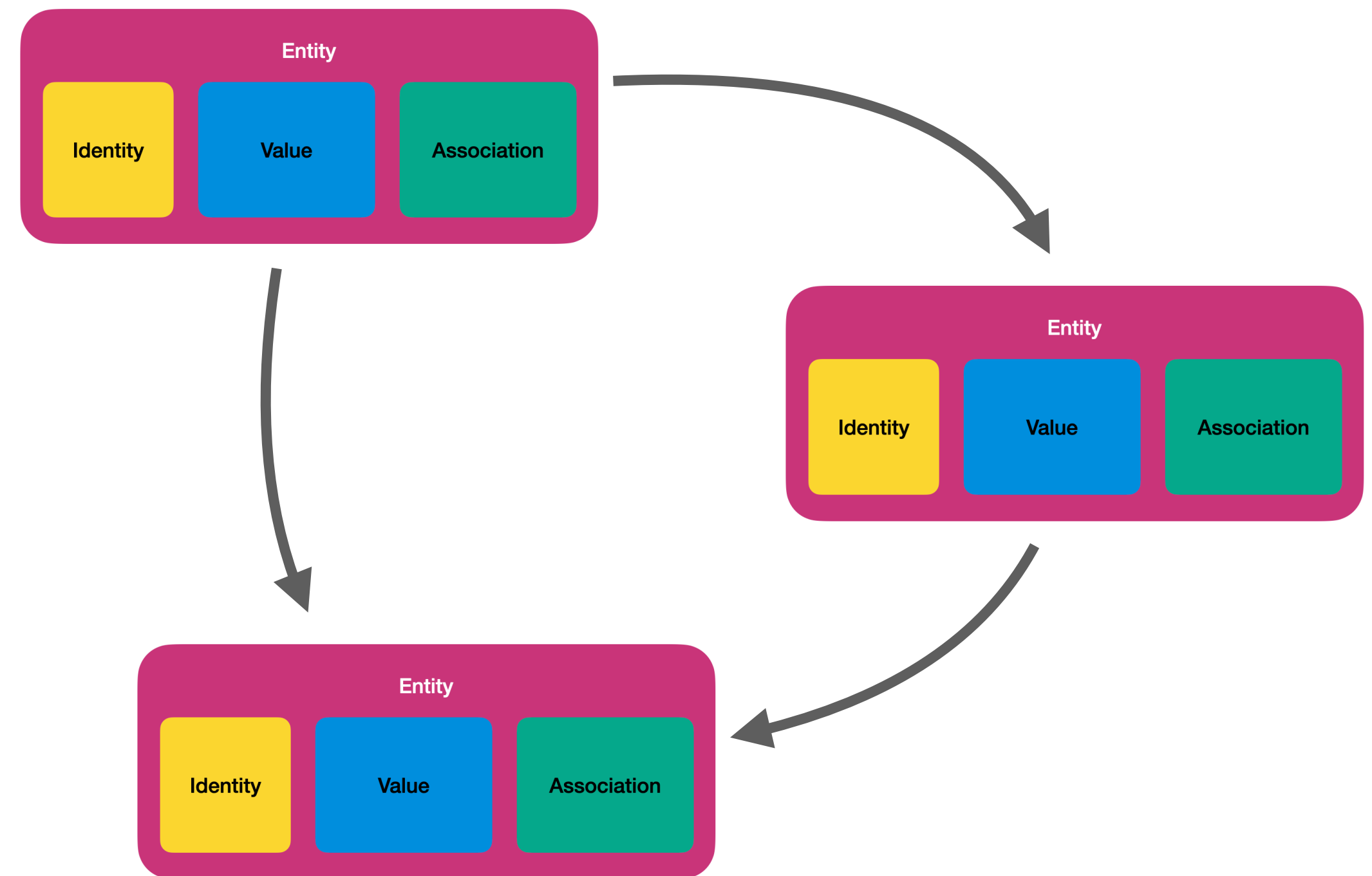


URI: /entities/{entity-id}

```
{
  "_links": {
    "self": {
      "href": "http://example.com/api/book/hal-cookbook"
    },
    "next": {
      "href": "http://example.com/api/book/hal-case-study"
    },
    "prev": {
      "href": "http://example.com/api/book/json-and-beyond"
    },
    "first": {
      "href": "http://example.com/api/book/catalog"
    },
    "last": {
      "href": "http://example.com/api/book/upcoming-books"
    }
  }
},
```

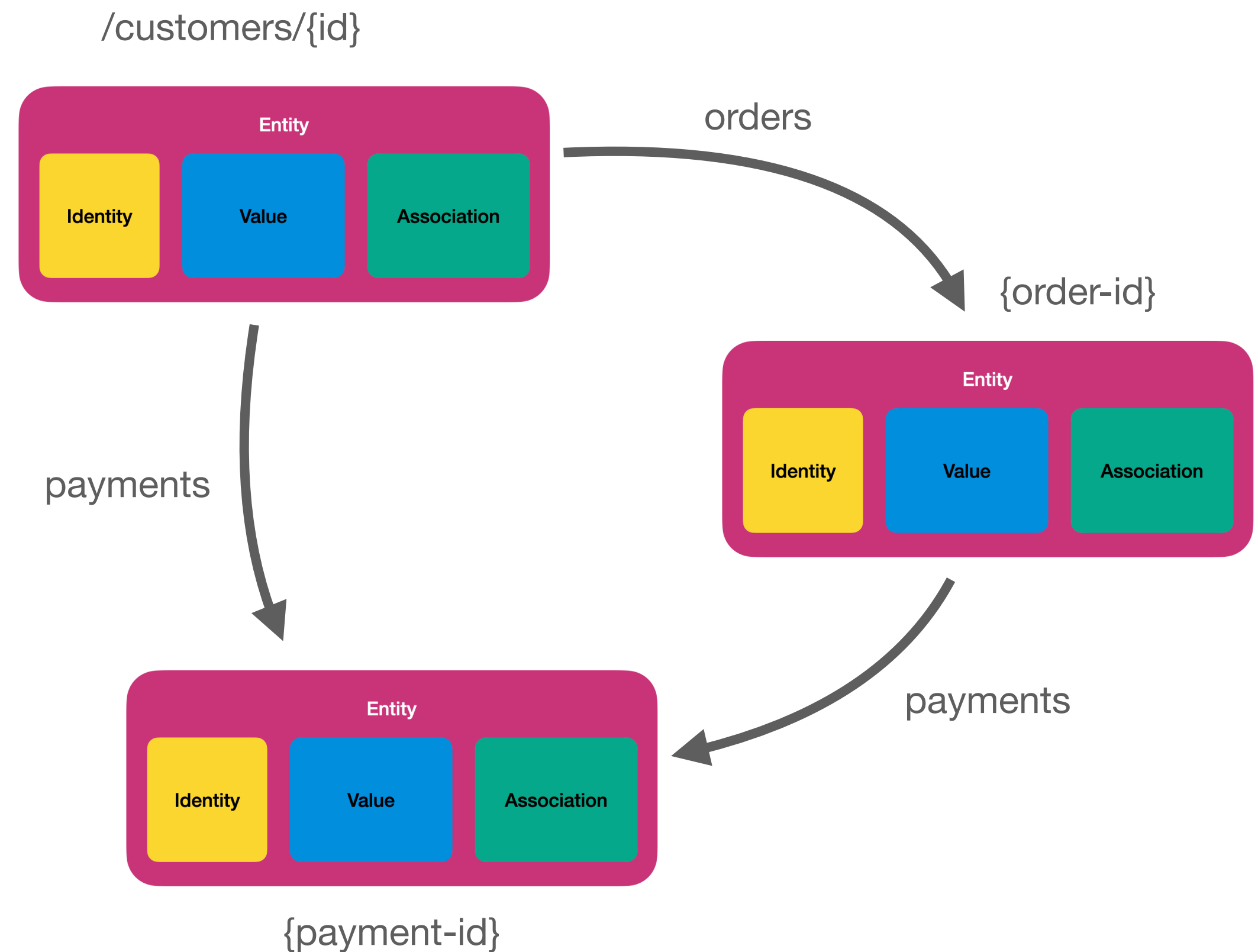
Smart Domain

- Smart domain provides an abstraction for API



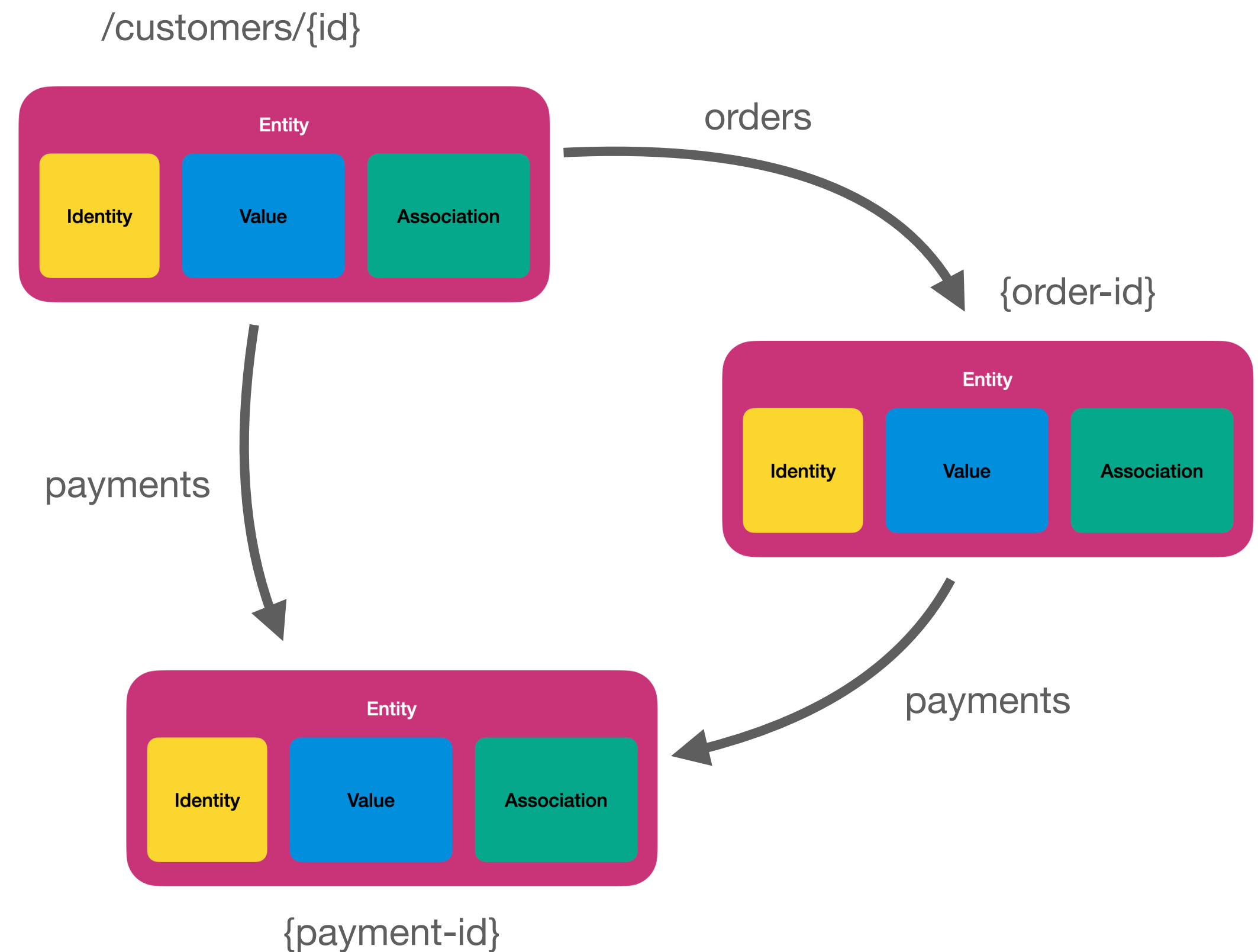
Smart Domain

- Smart domain provides an abstraction for API: navigation of the graph
 - Root entity as Root Resource(JAX-RS)
 - Association as Sub-Resource
 - Connected Entity as Sub-Resource of association



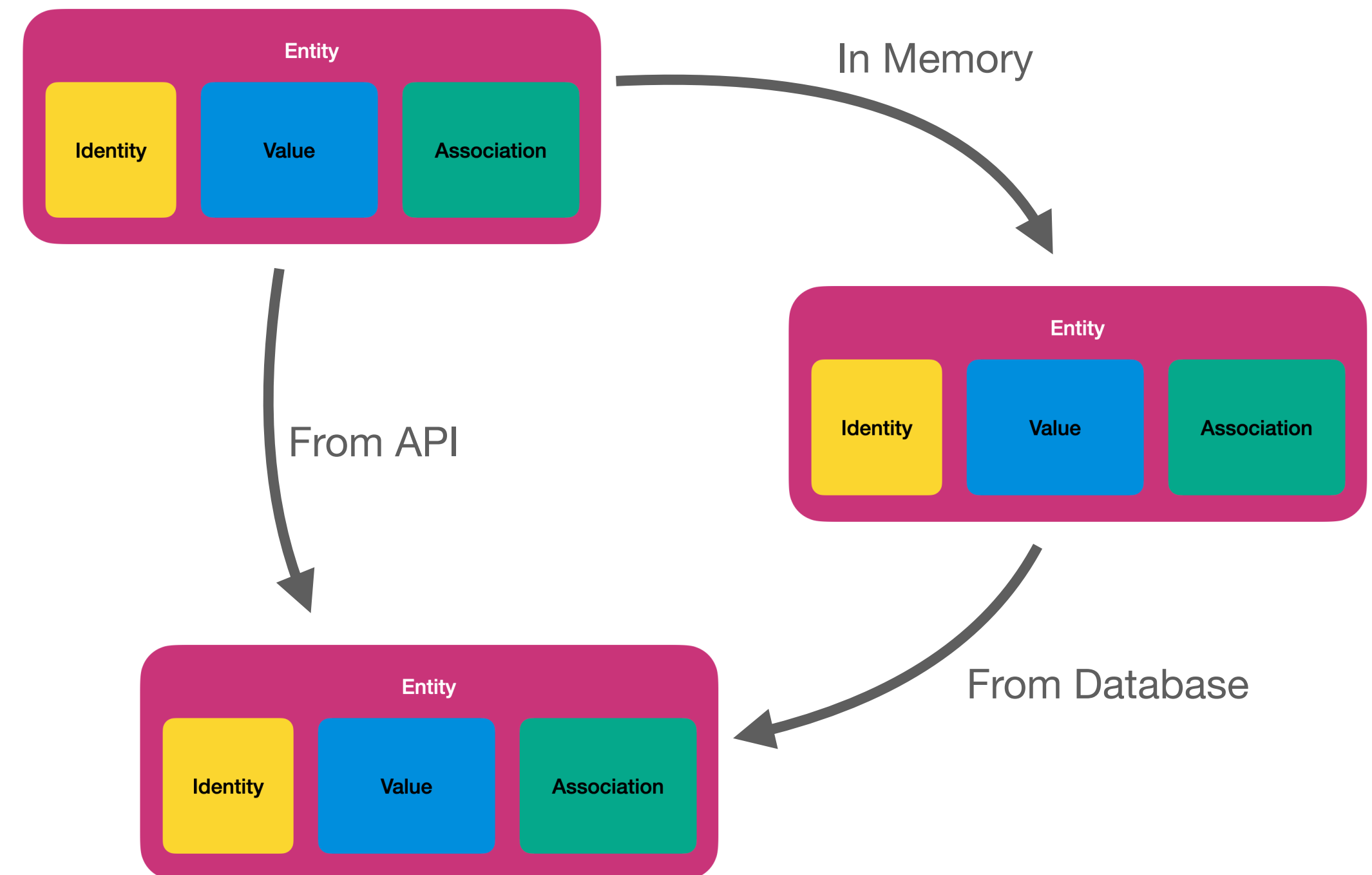
Smart Domain

- Smart domain provides an abstraction for API: navigation of the graph
- Smart domain provides an abstraction for lifecycle



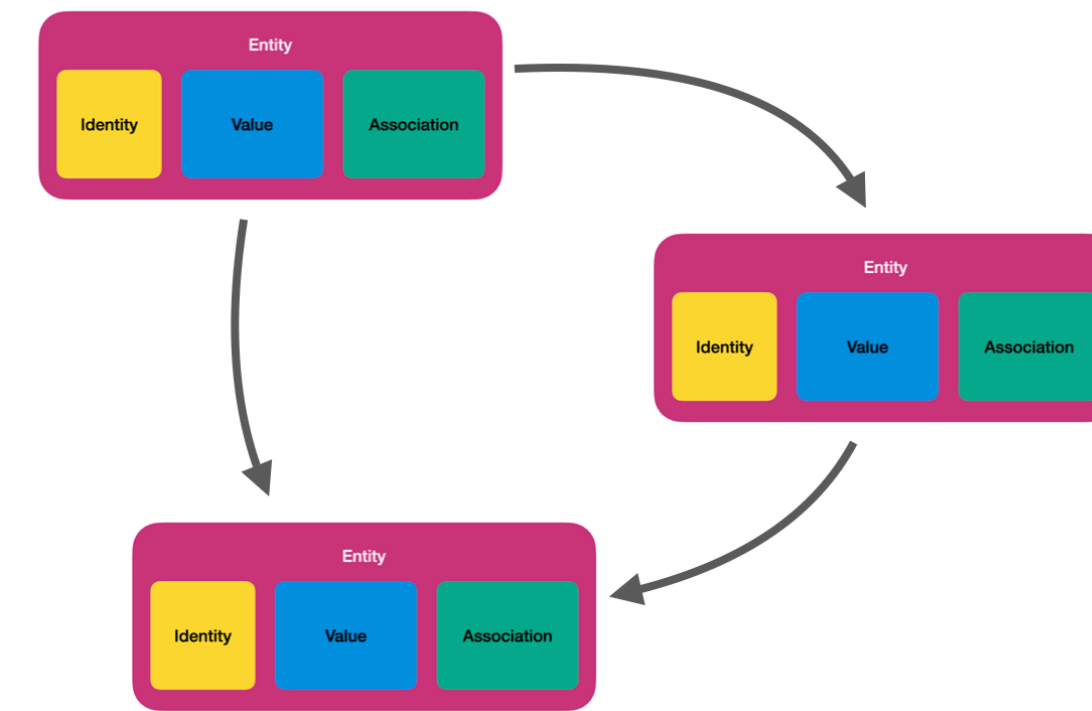
Smart Domain

- Smart domain provides an abstraction for API: navigation of the graph
- Smart domain provides an abstraction for lifecycle
 - Different implementations of association
 - Aggregation is **an implementation detail**



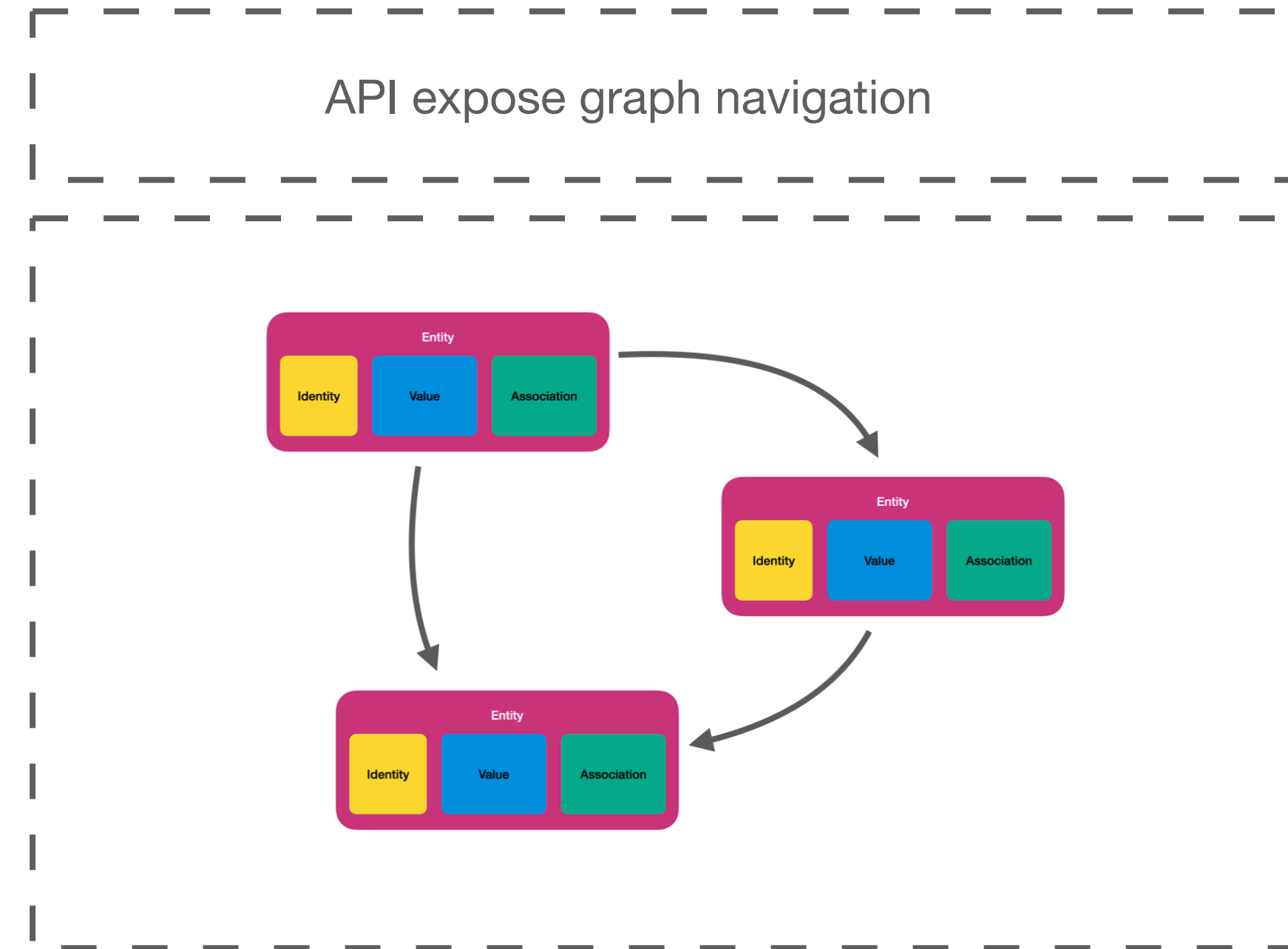
Smart Domain

- Two layers architecture
 - Domain layers: define the entities and the associations between entities



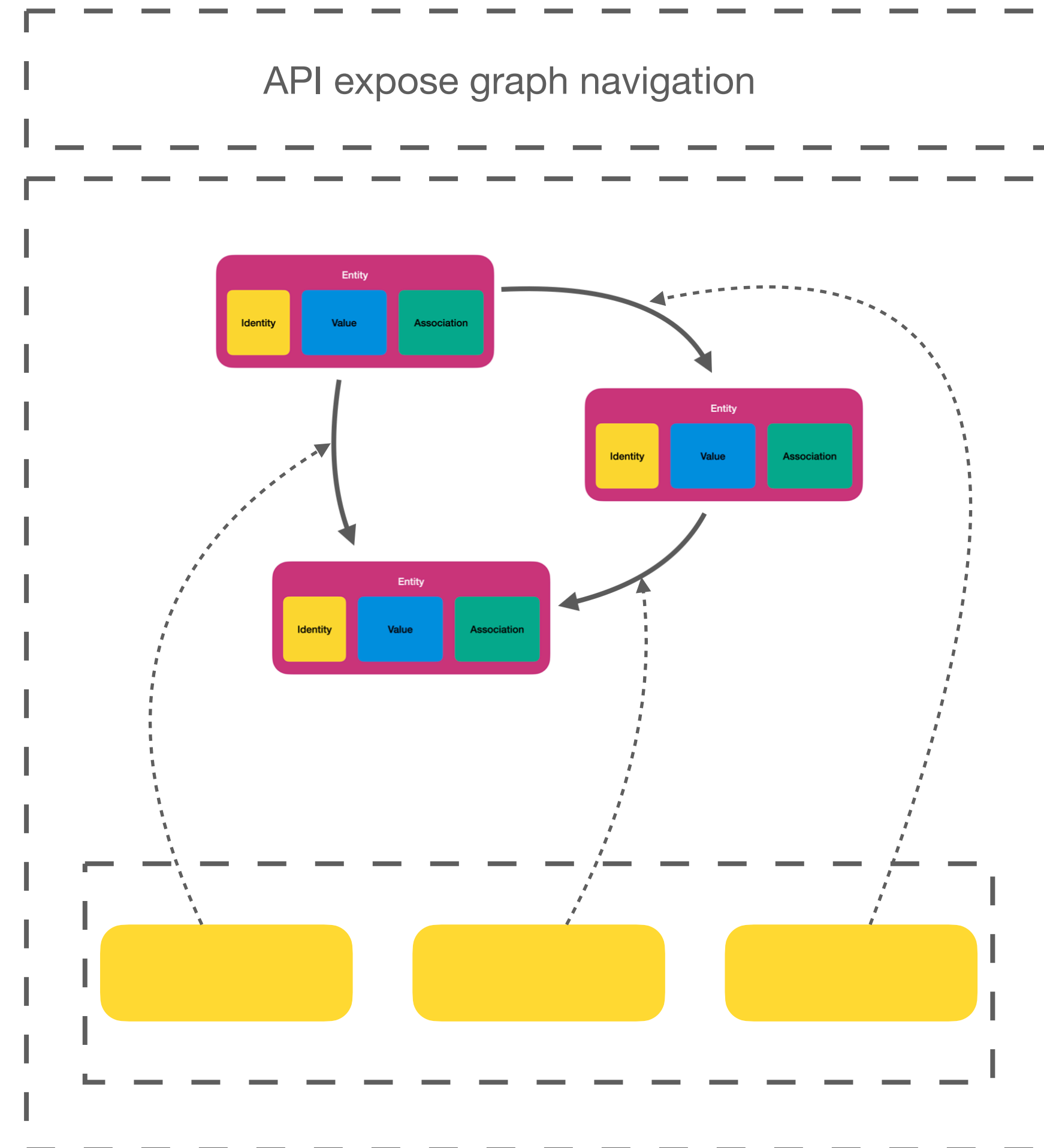
Smart Domain

- Two layers architecture
 - Domain layers: define the entities and the associations between entities
 - API layers: provides API based on the graph navigation defined in domain layer

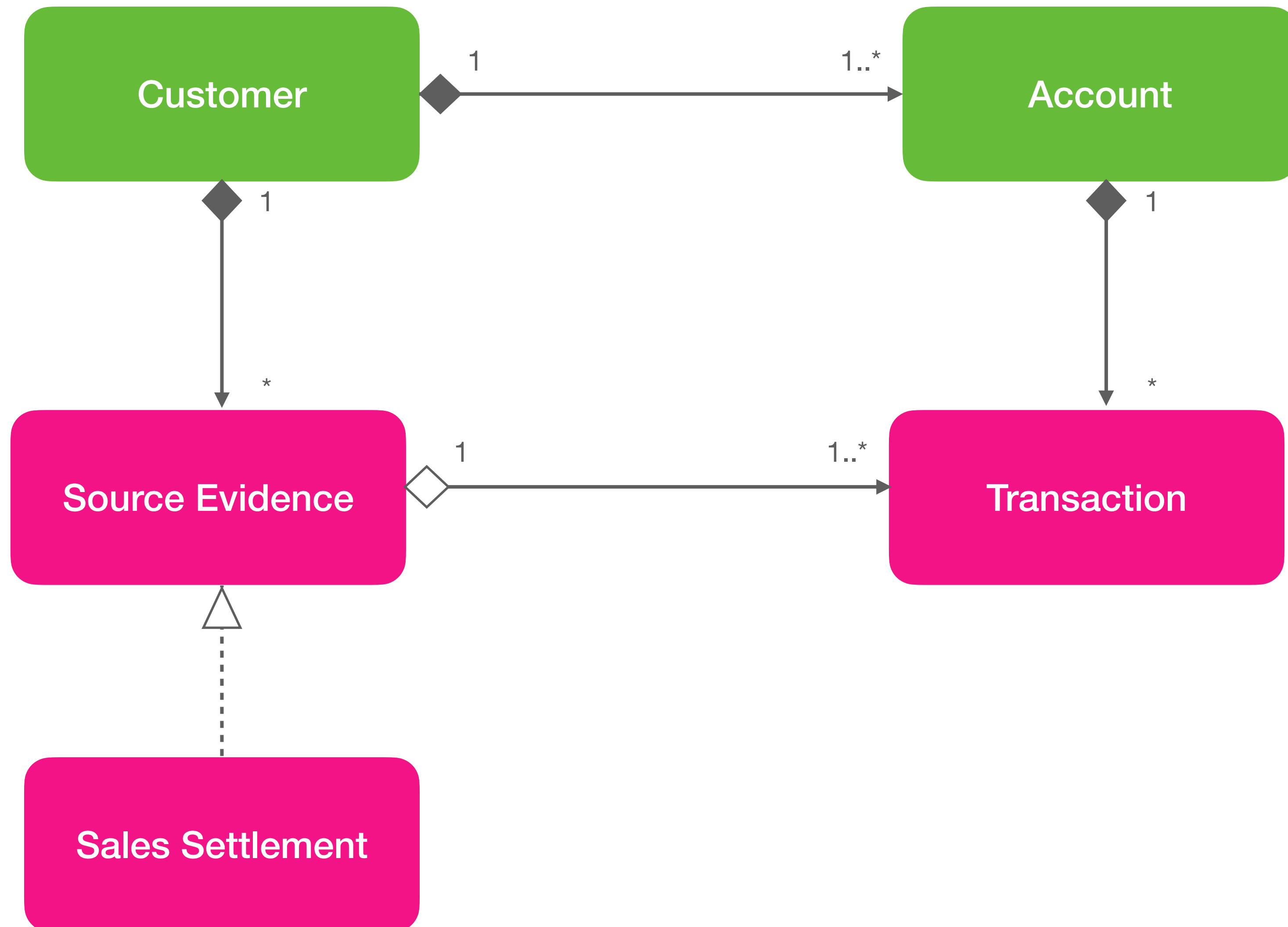


Smart Domain

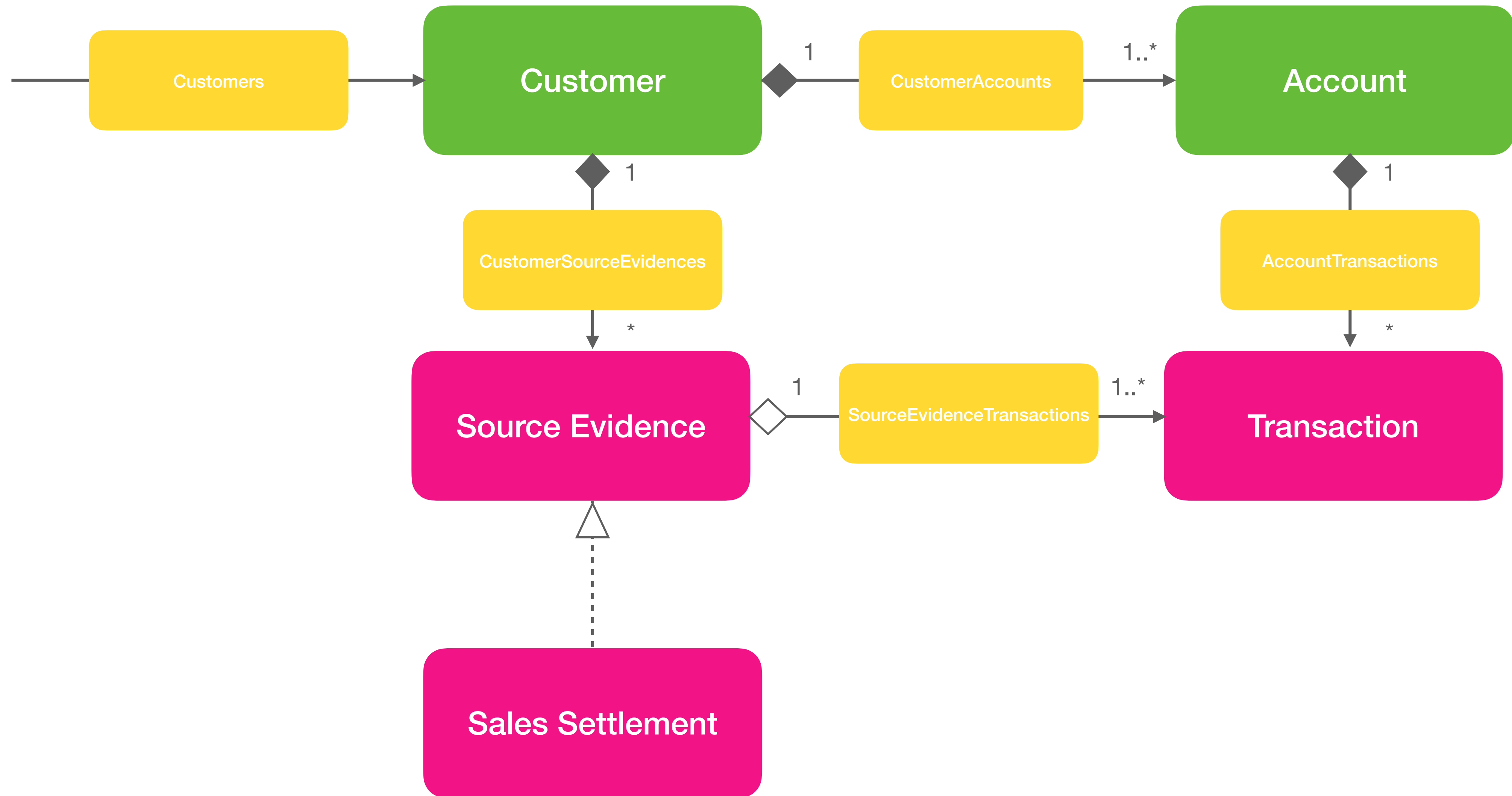
- Two layers architecture
 - Domain layers: define the entities and the associations between entities
 - Integration layers: provides implementation of associations
 - API layers: provides API based on the graph navigation defined in domain layer



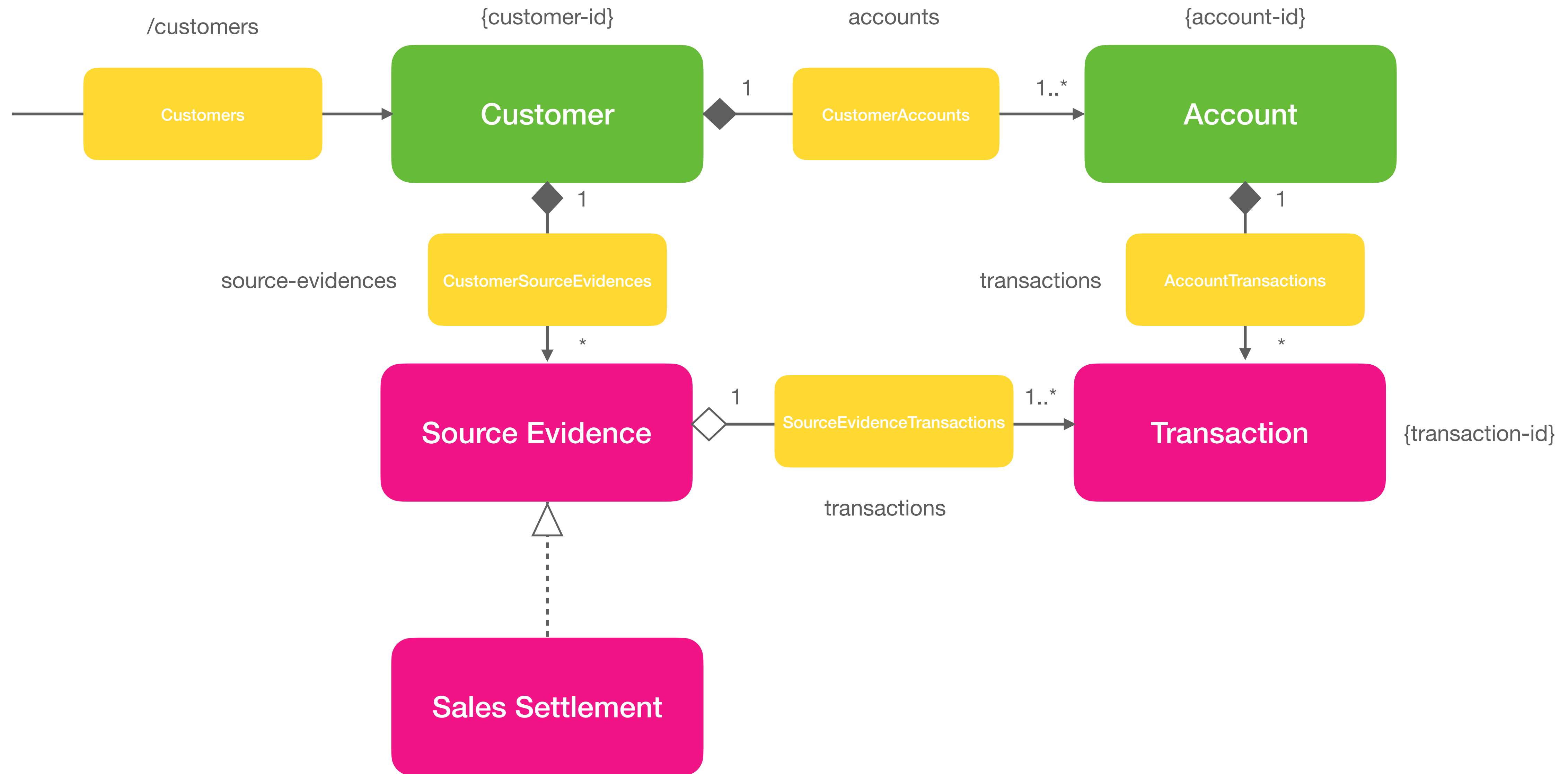
Accounting Domain Model



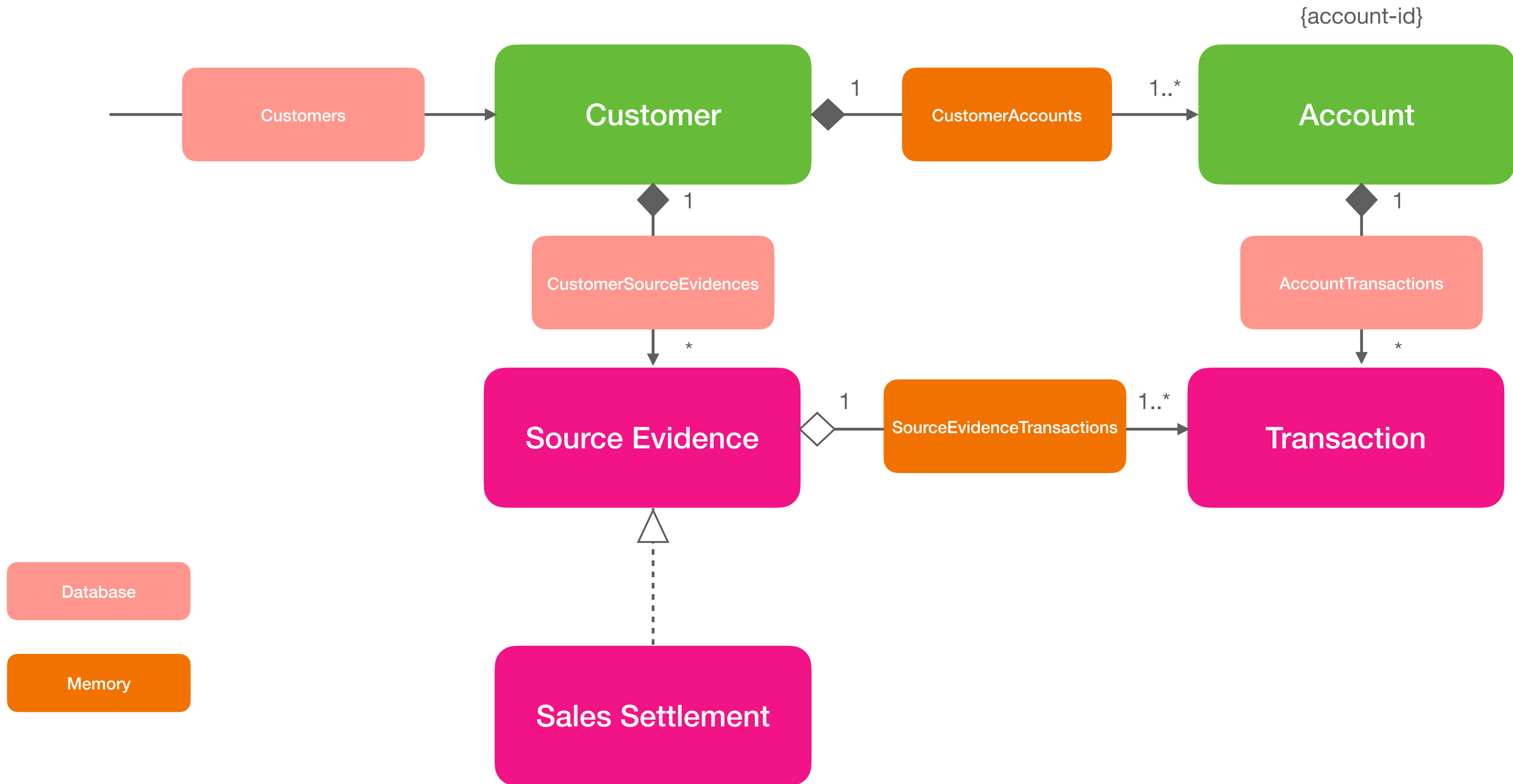
Accounting Domain Model - Association



Accounting Domain Model - API



Accounting Domain Model - Lifecycle



Demo project

- <https://github.com/Re-engineering-Domain-Driven-Design/Accounting>