Finance CoLimit

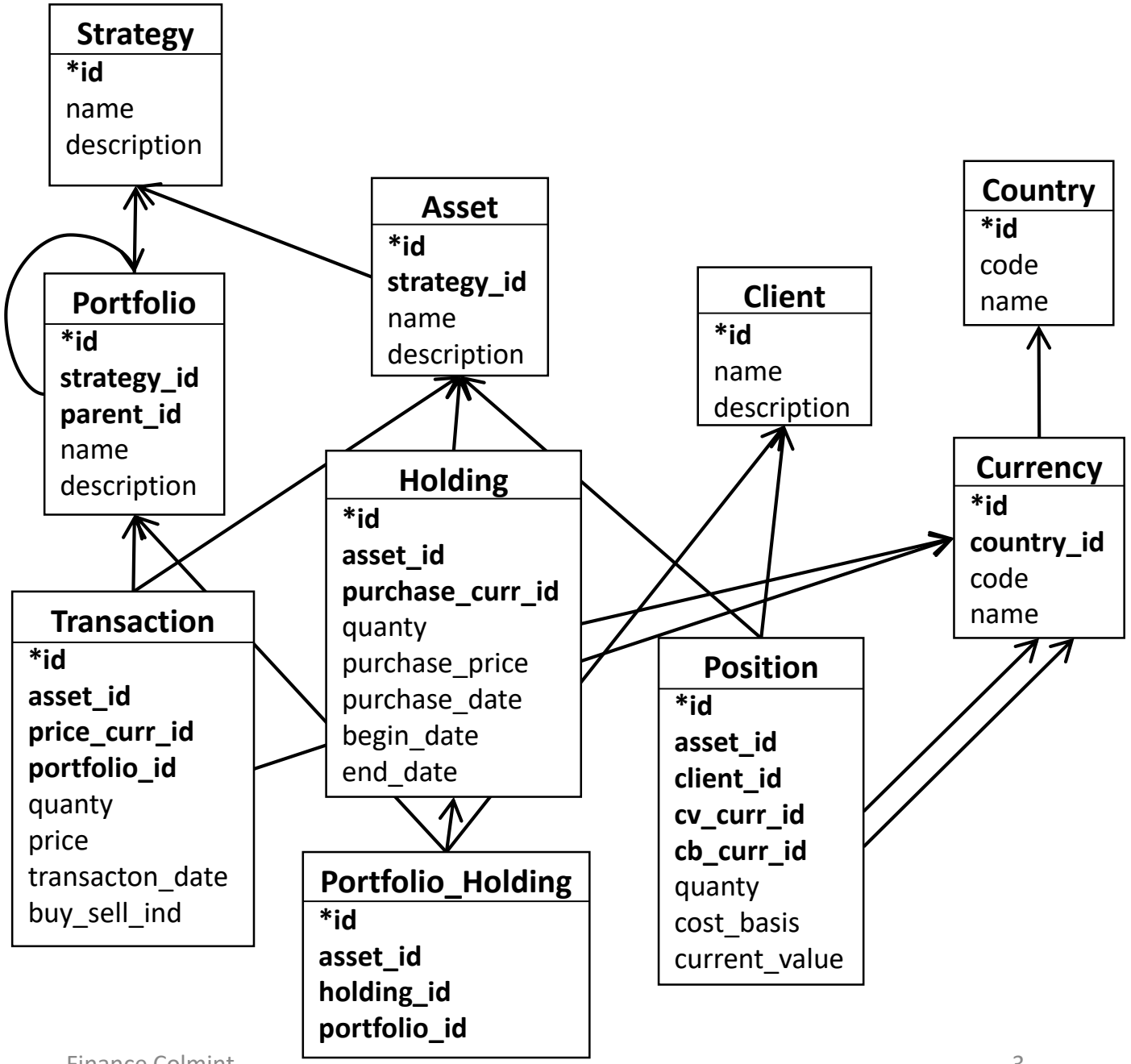
Peter’s take on Jee and Ryan’s AQL file

Project Objectives

- Understand finance concepts and colimit implemetation
  - Create a plan (illustrated in the following slides)
  - Formalize these ideas using AQL
- Spin off proposed model management requirements and tests
  - Generate a list of user requirements based on my experience in formalizing the plan.
  - Generate a list of negative tests based on errors that arise during project execution.
  - Generate a list of positive tests based on successes at various stages in project execution.

Initial Schema

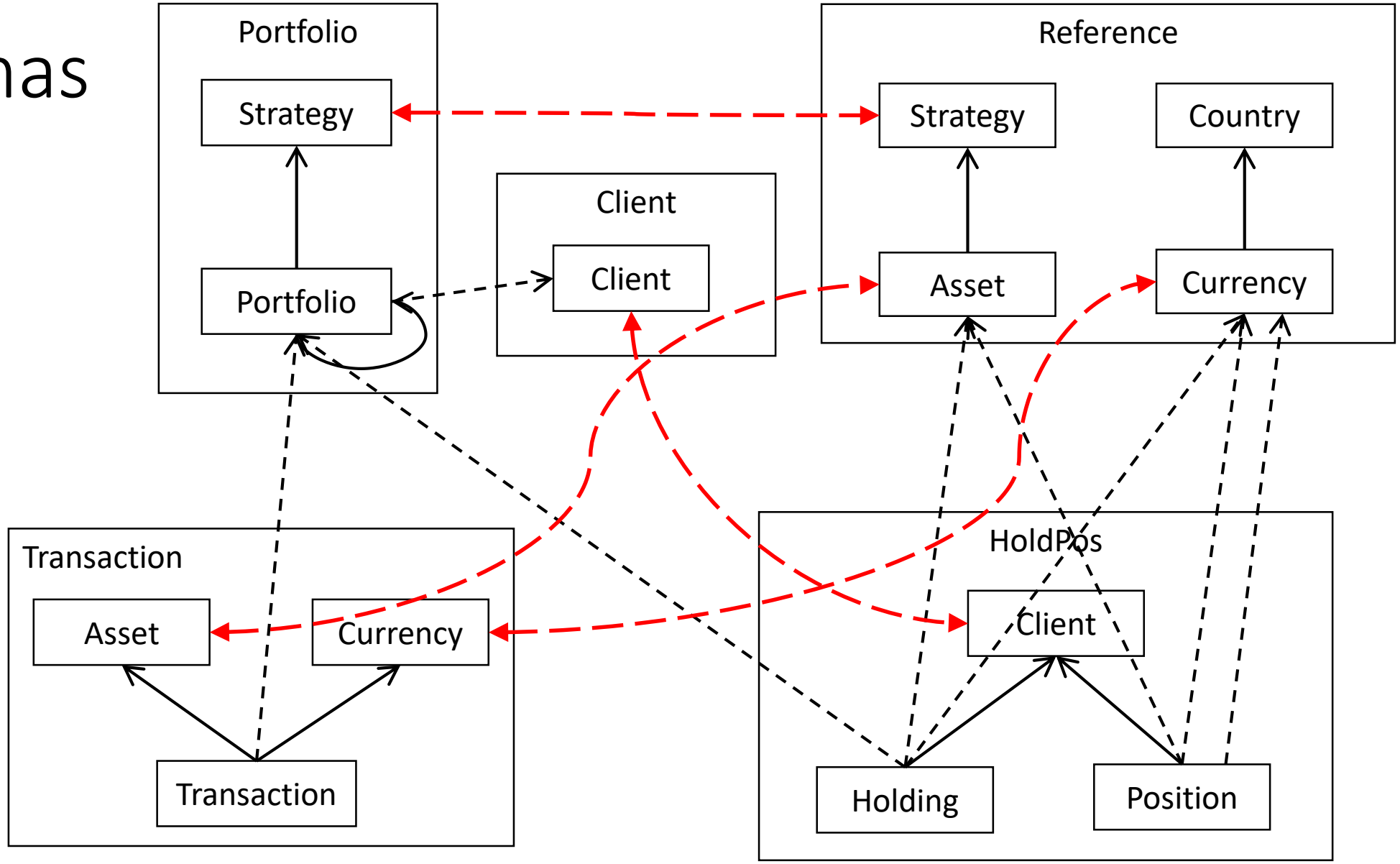
Transcribed from Jee's file:
asset-mgmt-er in drop box.



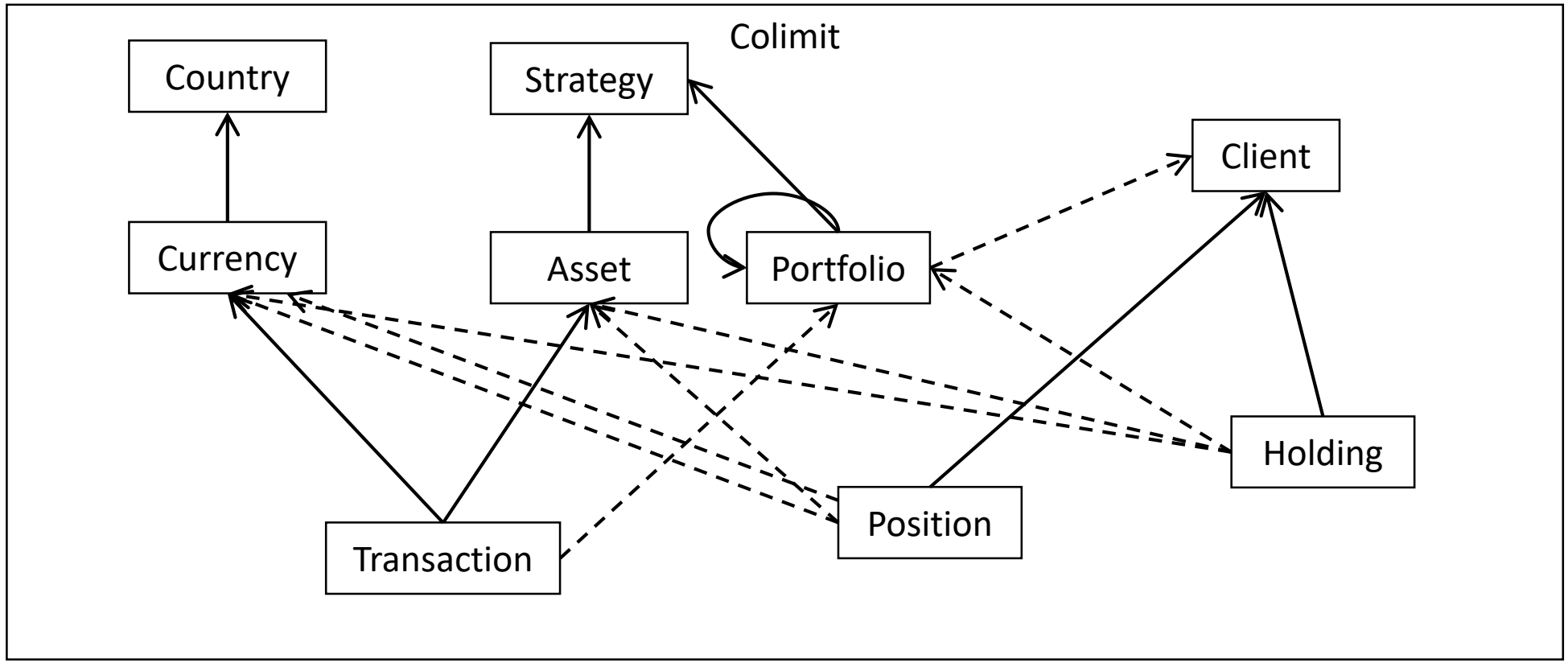
Analysis of Existing Finance Colimit AQL Implementation

- What follows are slides reflecting my attempt to understand the existing Finance Colimit file.
- Slide 4 contains the schemas deconstructed from the initial schema in slide 2.
- Red double headed arrows indicate entities to be merged using a colimit.
- Solid arrows indicate AQL foreign keys
- Dashed arrows indicate foreign keys in the original schema that crossed schema boundaries of the deconstructed schemas.
- Changes implemented using uber flowers indicated in green (arrows/entities added) and red (arrows/entities removed).

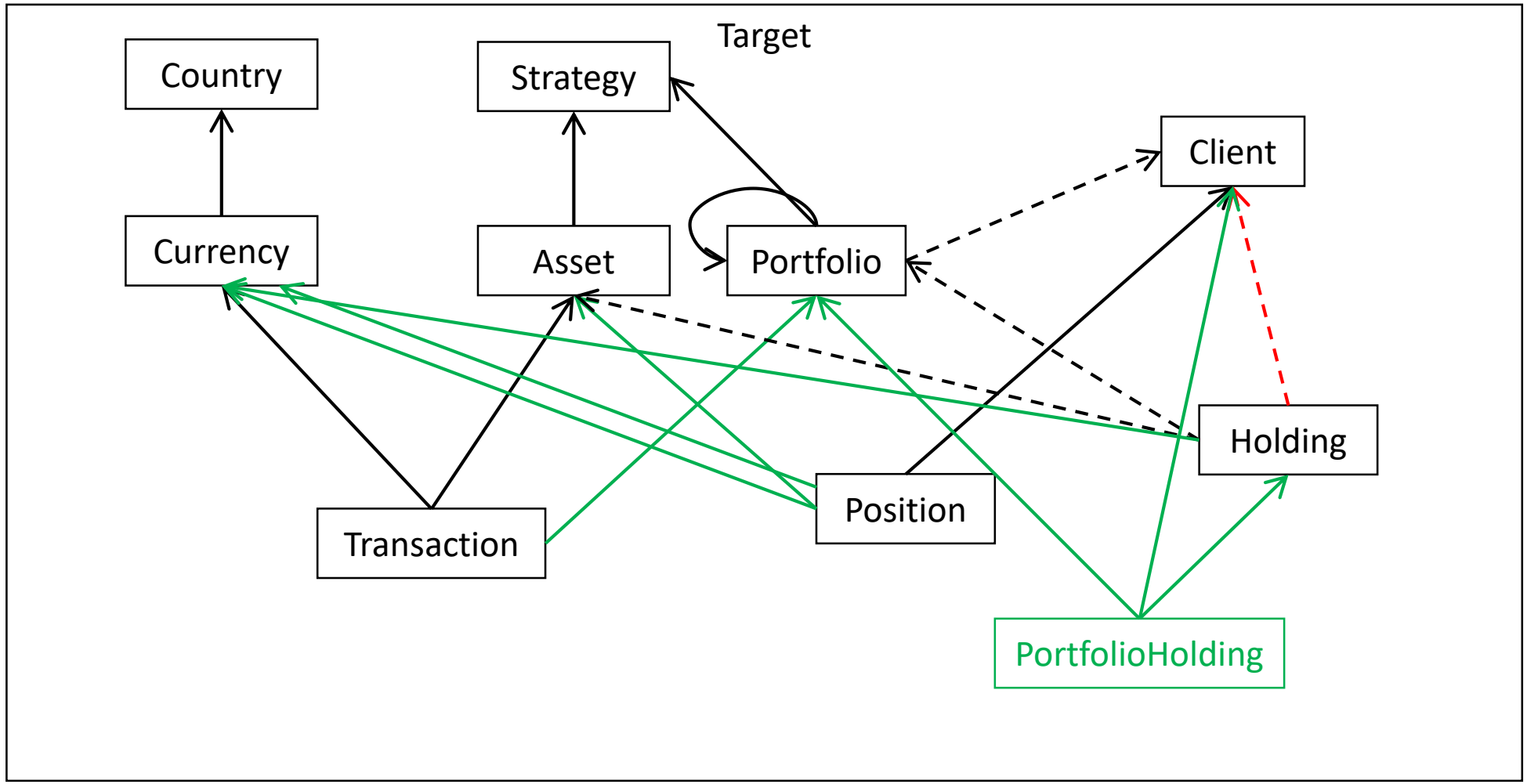
Schemas


Portfolio
Strategy
Portfolio
Client
Client
Reference
Strategy
Country
Asset
Currency
Transaction
Asset
Currency
Transaction
HoldPos
Client
Holding
Position

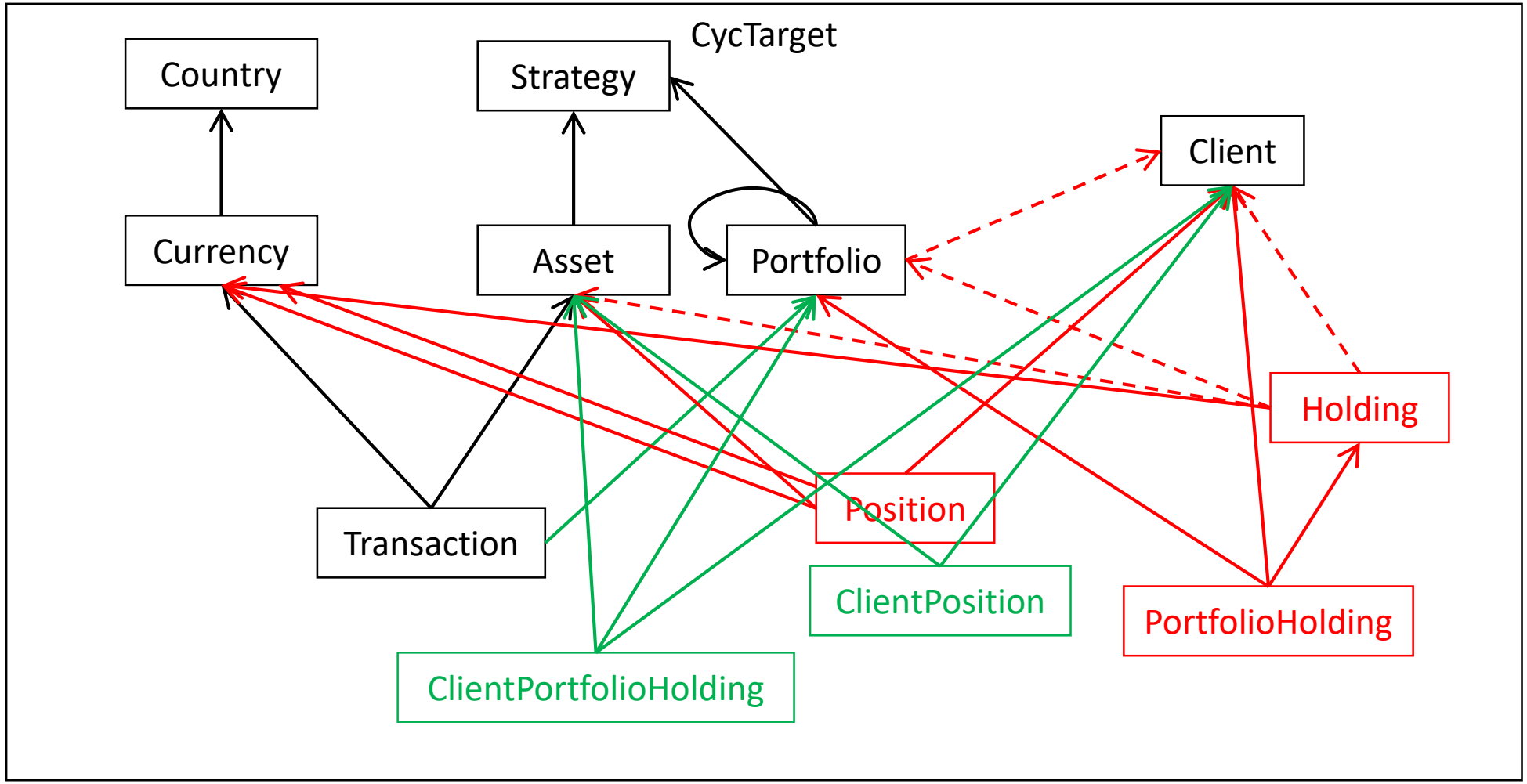
Schema Colimit



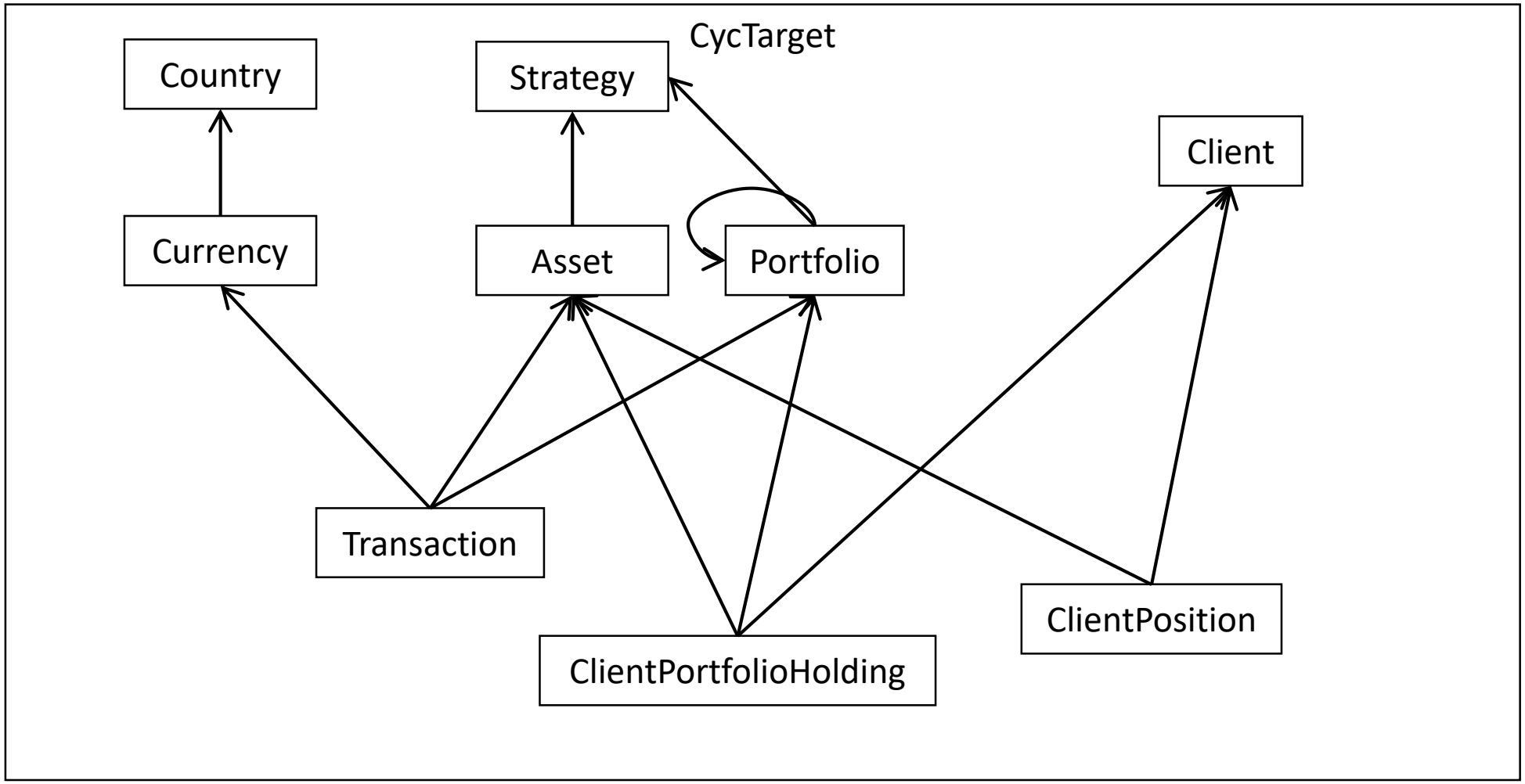
Schema Target: Changes from Colimit (query ETL)



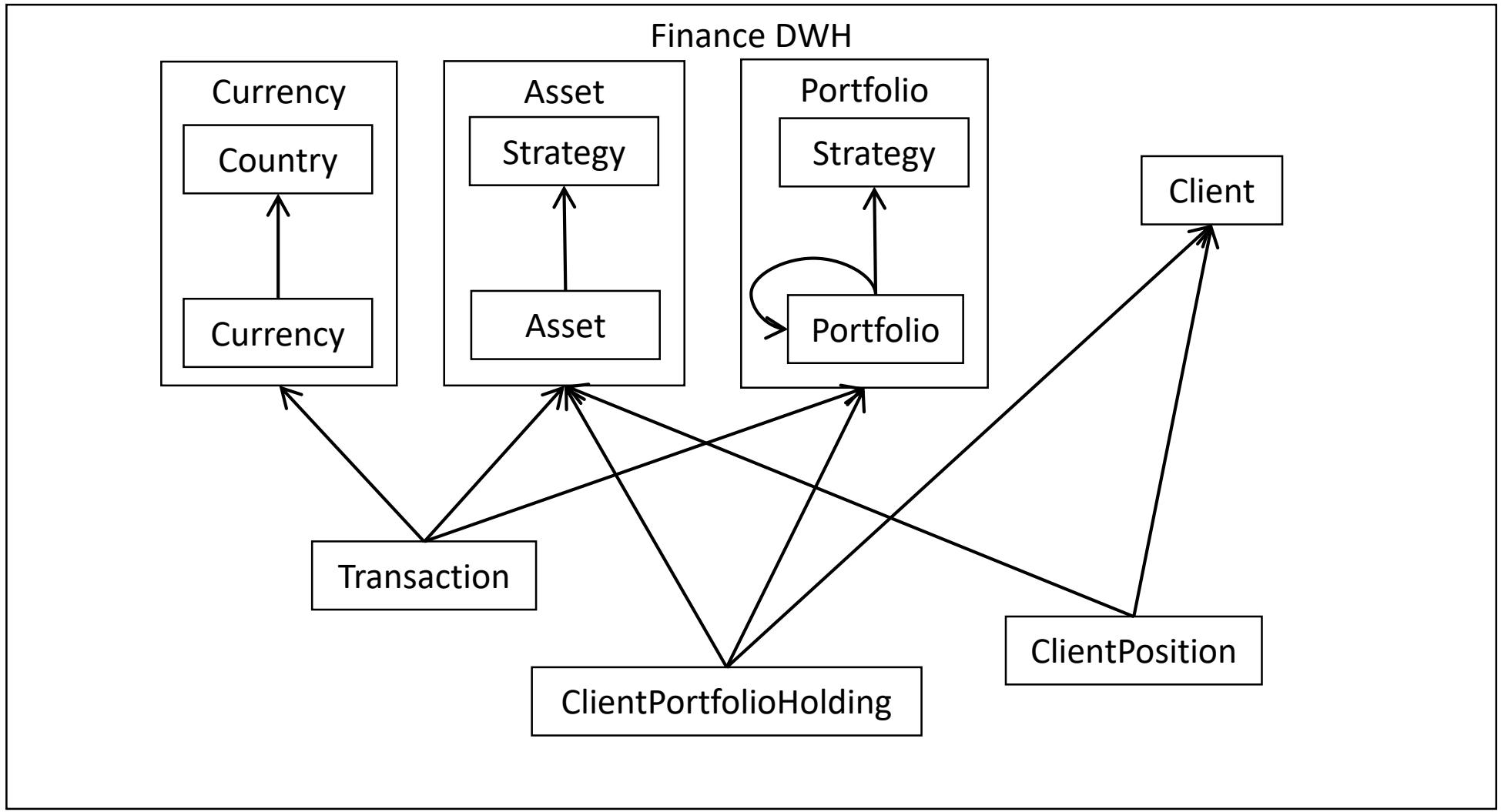
Schema CycTarget: Changes from Target (query ETL_Cyc)



Schema CycTarget



Proposed DWH



Bus Matrix

	Currency	Asset	Portfolio	Client
Transaction	X	X	X	
Client Portfolio Holding	X	X	X	X
Client Position	X	X		X

Refactoring Exercise

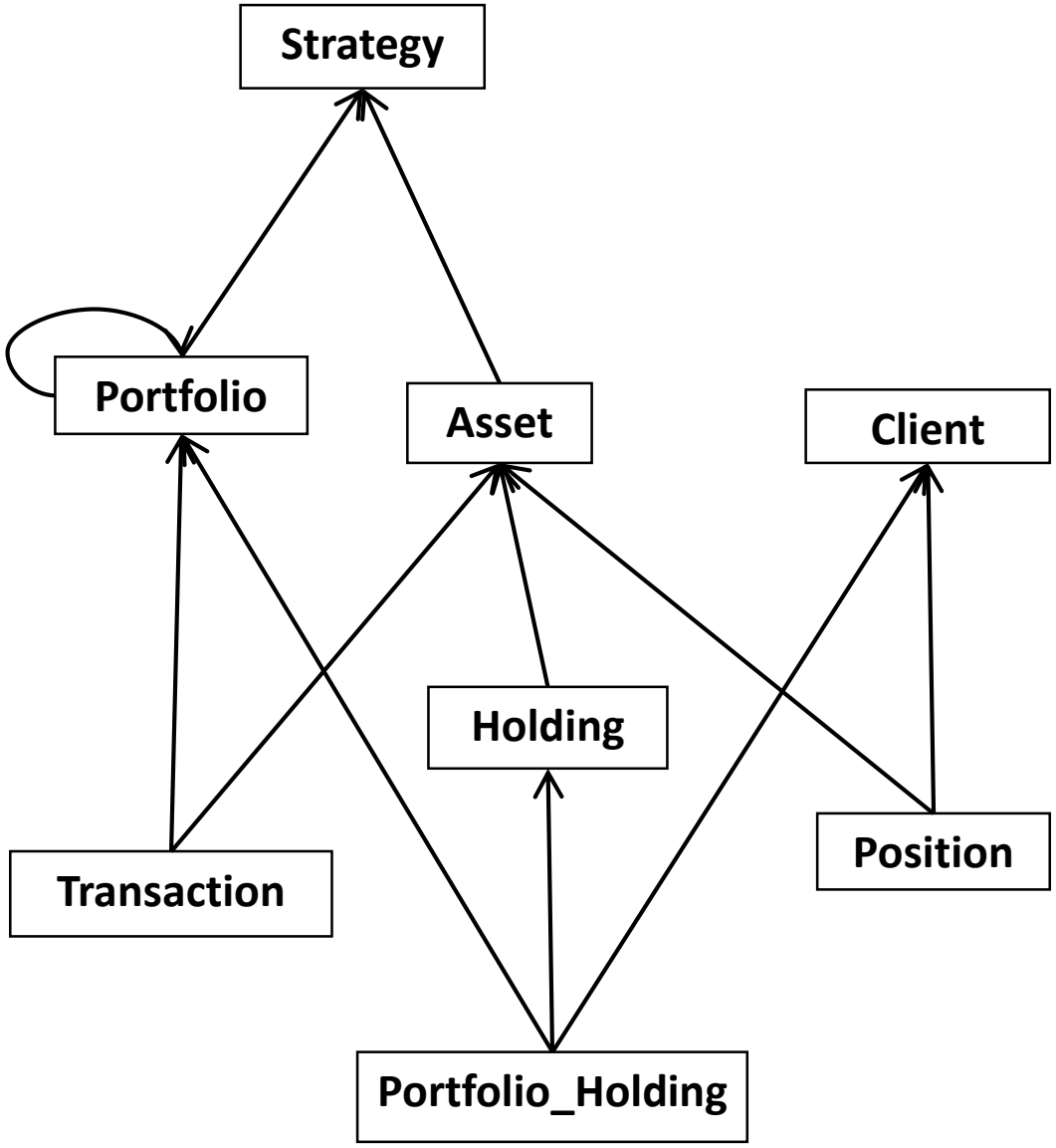
- Given an understanding of the concepts obtained from an analysis of the existing model and conversations with Jee consider an alternative starting schema consistent with slides 9 and 10.
- Consider how to redefine a set of source schema's that are more plausible as sources in an integration project.
- Conclusions from this analysis can be found in slide 20.
- Note: My goal is to create a more plausible set of source schema's while maintaining a faithful connection to finance concepts.

❖**Request to Jee:** Please review and let me know if I have departed from such a faithful connection.

Initial Schema Less:

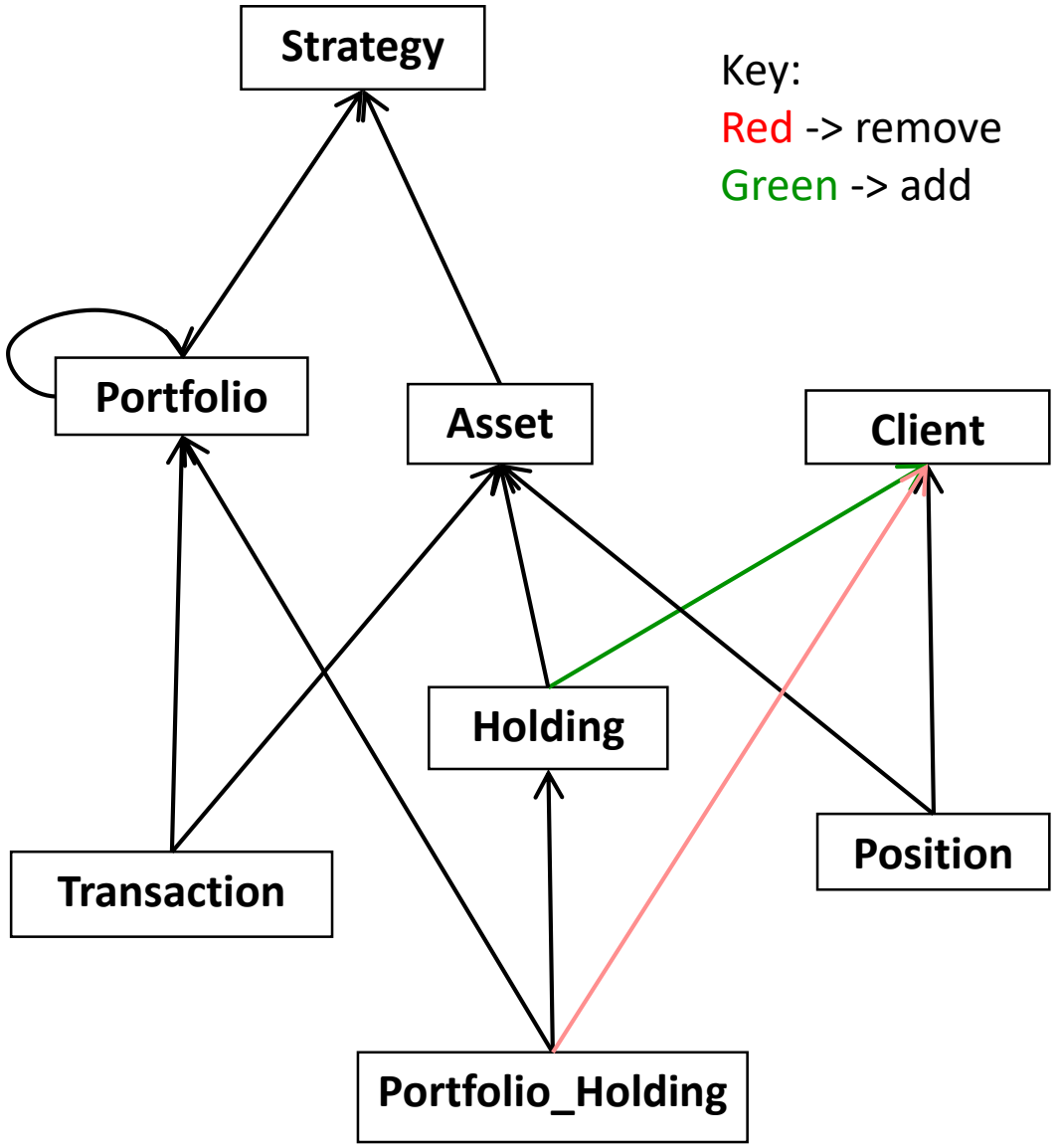
Attributes

Currency



Refactoring #1

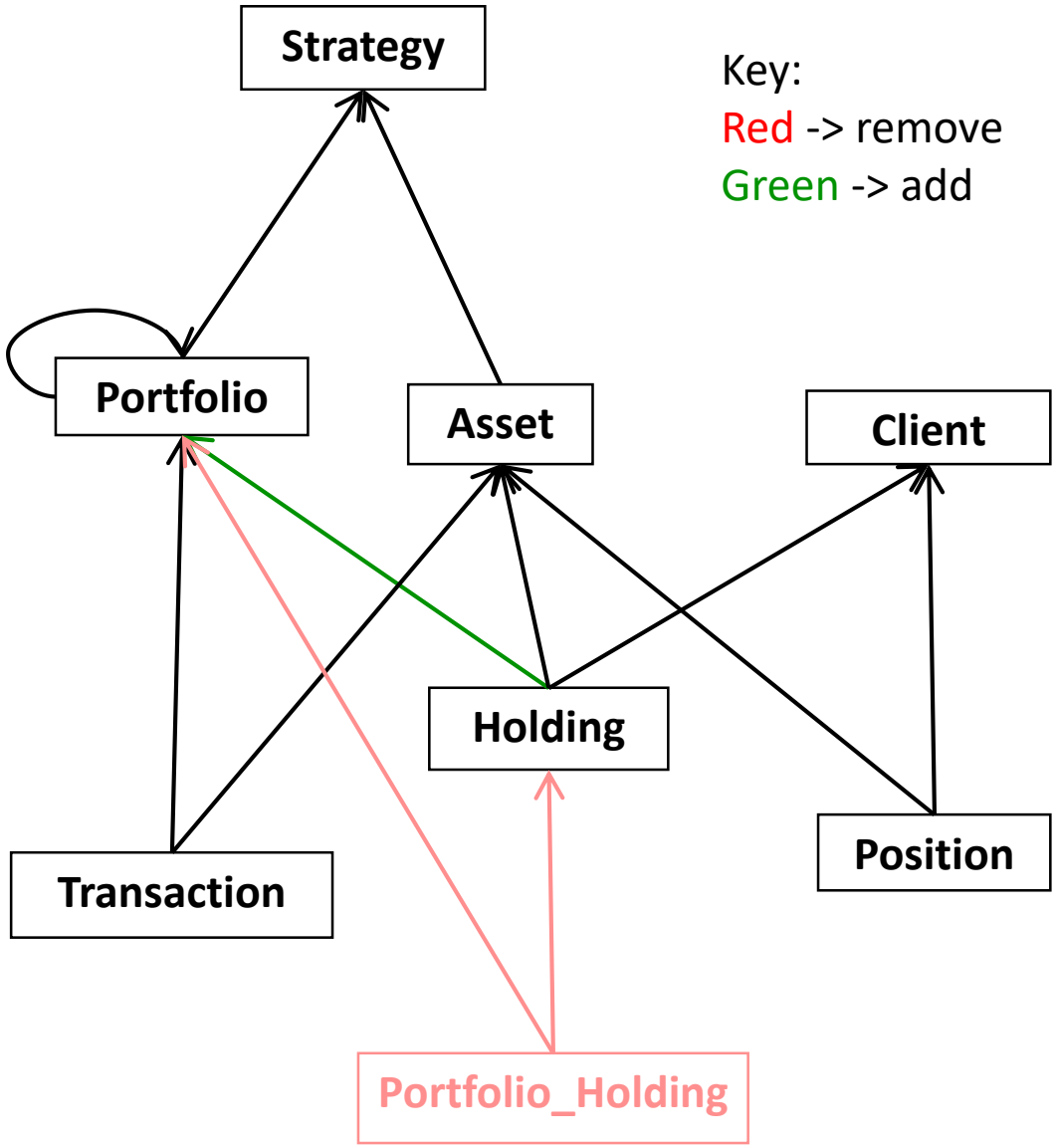
Rational:
A holding can only be owned by a single client. This implies a direct functional dependency from holding to client and so the foreign key from portfolio_holding to client is not needed.



Refactoring #2

Rational:
I am less sure about this one. I don't really know what a portfolio is. If it is something that groups holdings so that they can be collectively managed using a strategy than this is correct. The question is: Can a holding be in more than one portfolio? If not, an association table (portfolio_holding) provides too much flexibility.

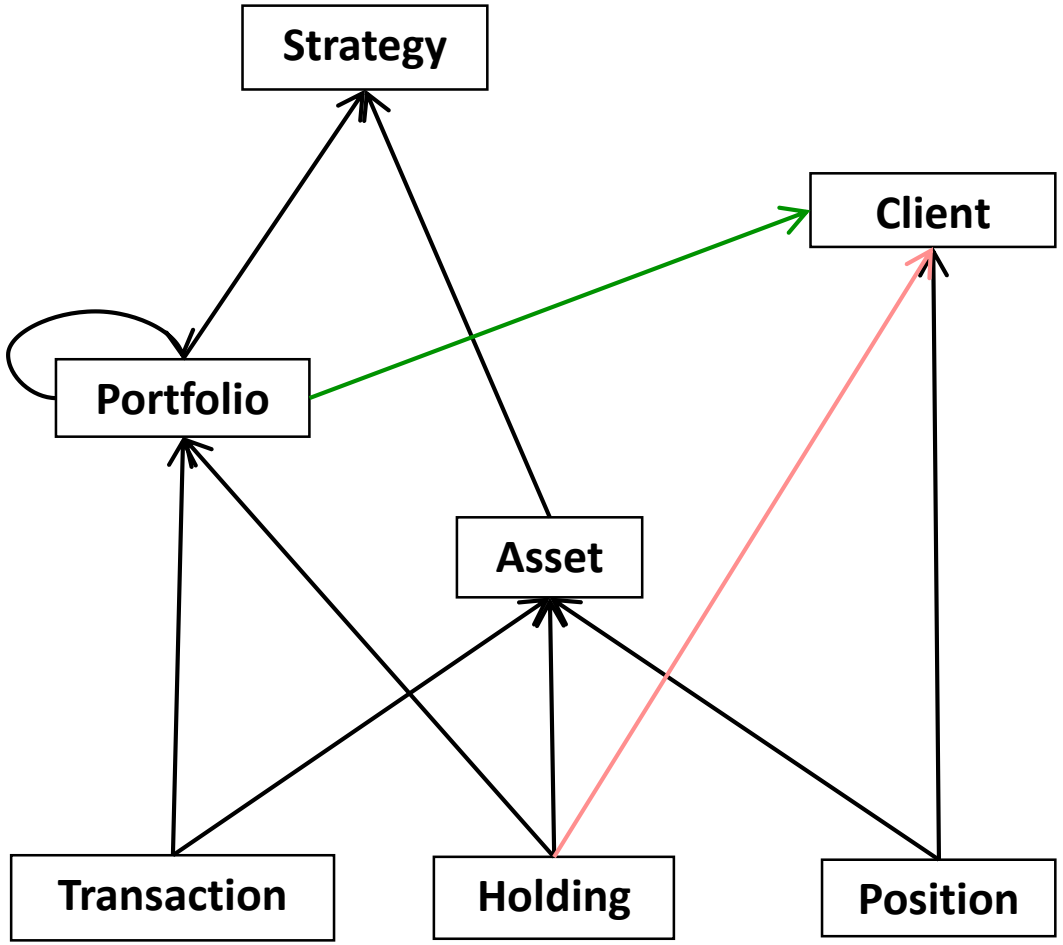
Given the hierarchical structure of portfolio I am wondering if the notion of a portfolio might be overloaded. This might explain the model, but I would be more comfortable pulling apart the different subtypes and thinking about how they are related to other entities. Jee mentioned that words are frequently overloaded in finance.



Refactoring #3

Key:
Red -> remove
Green -> add

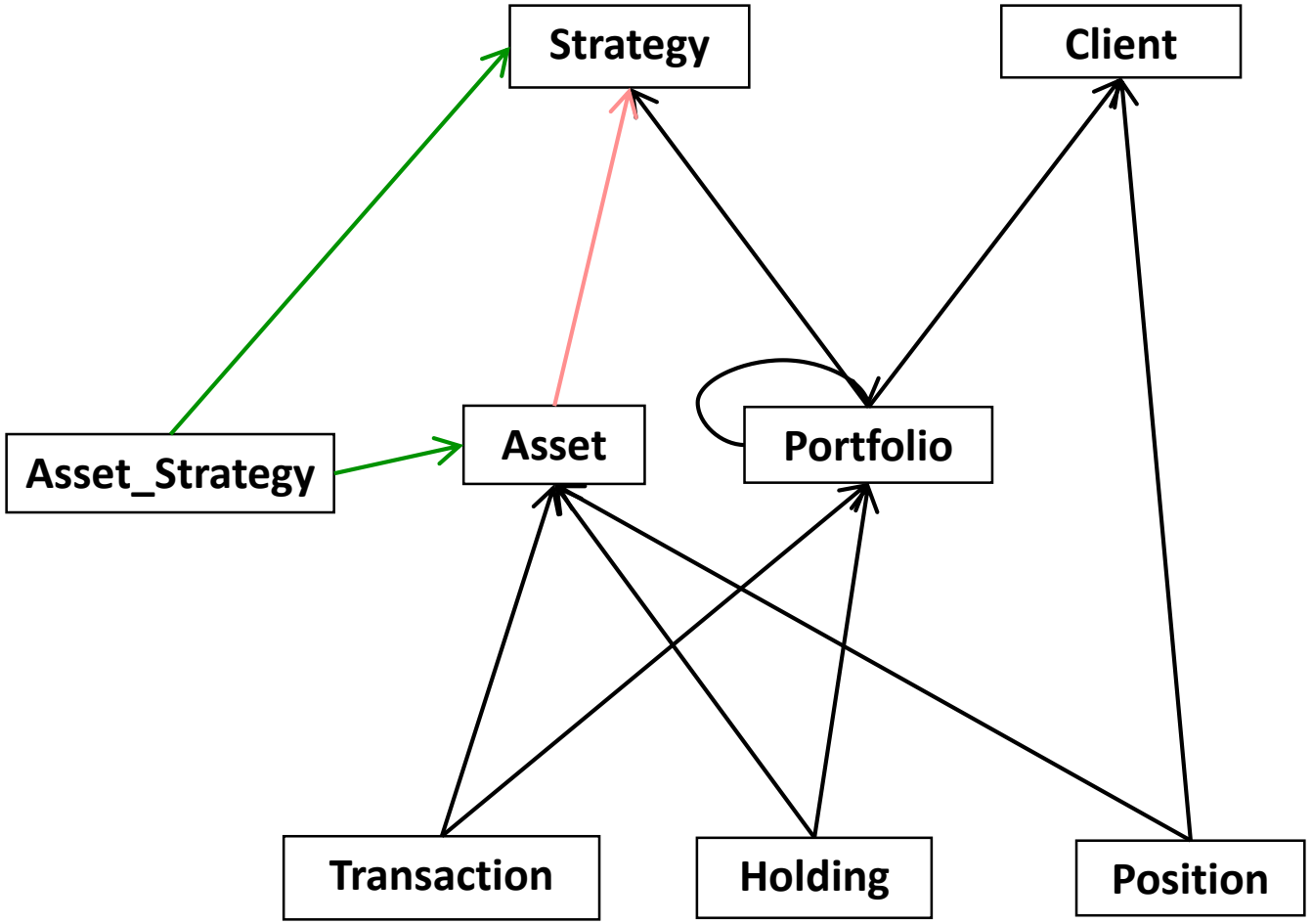
Rational:
If refactoring #2 is correct then it seems to me a portfolio is owned by a client and so that makes lots of things simpler. The foreign key holding -> client is just the path holding -> portfolio -> client and so can be omitted.



Refactoring #4

Key:
Red -> remove
Green -> add

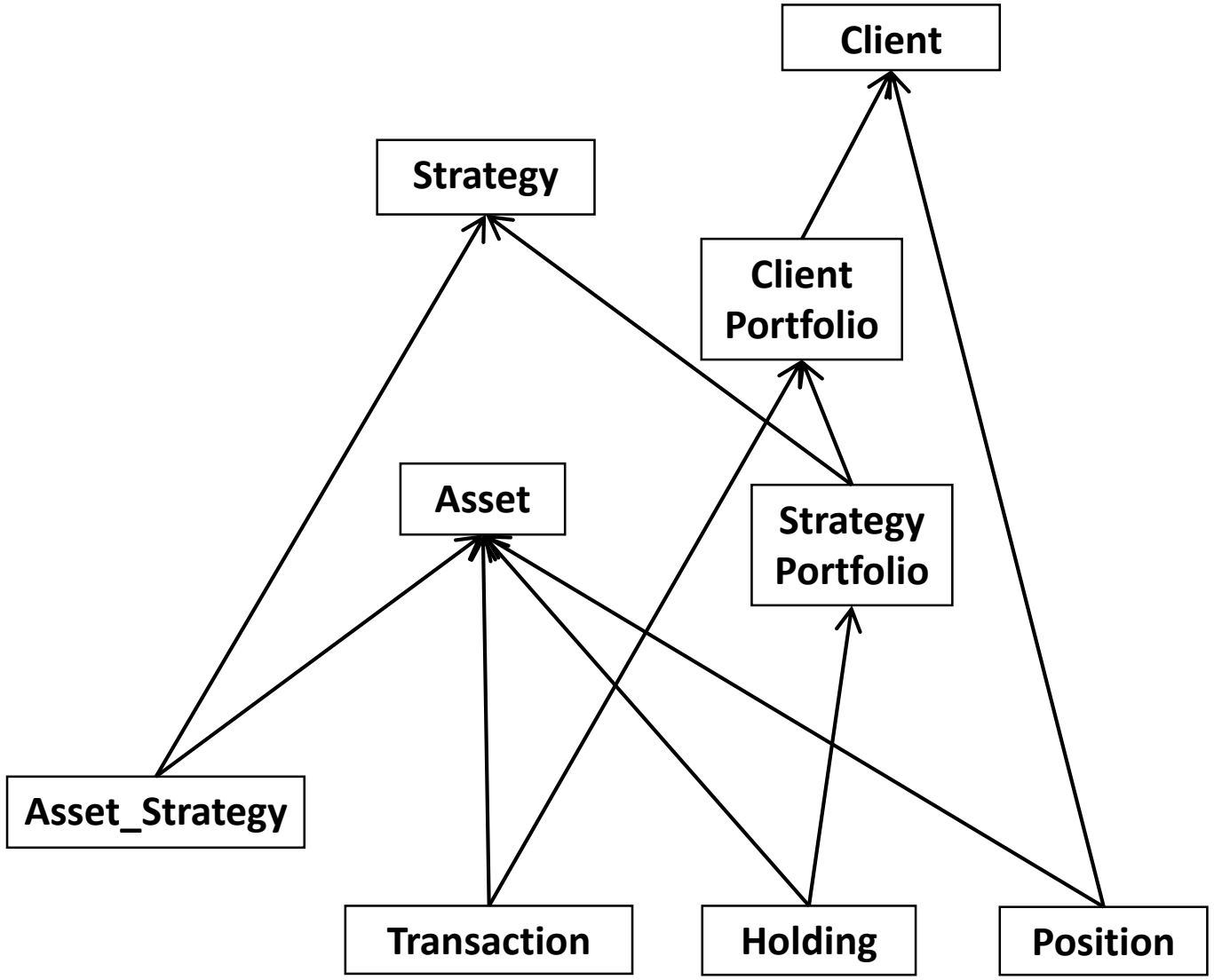
Rational:
An asset being a typing (i.e. reference) entity it could be associated with any number of different strategies.



Refactoring #5

Now I am going to go way out on a limb. Based on my relationship with my financial advisor this is the way I see it. I have an overall portfolio (client portfolio) that is a group of different sub-portfolios, each of which may have a somewhat different strategy and tax implications. For example, Roth IRA, Traditional IRA, 401K, Non tax exempt investment portfolio… Over time I have had to merge portfolios due to a change of employer.

I think this schema could be used to define more realistic integration scenarios.



Alternative Schema

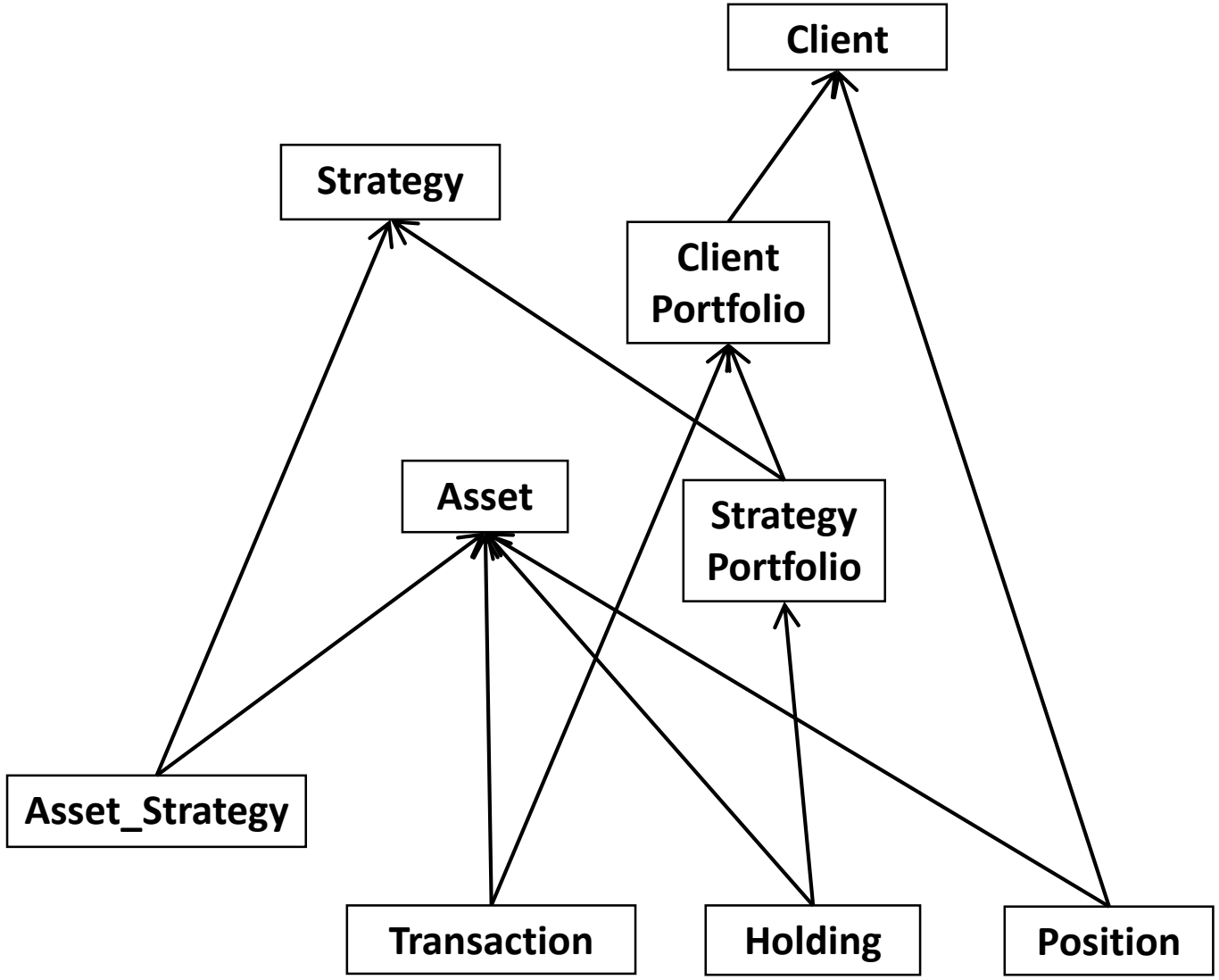
This schema has some interesting implications from a data warehousing point of view. You end up with an interesting mix is facts and dimensions.

Facts:

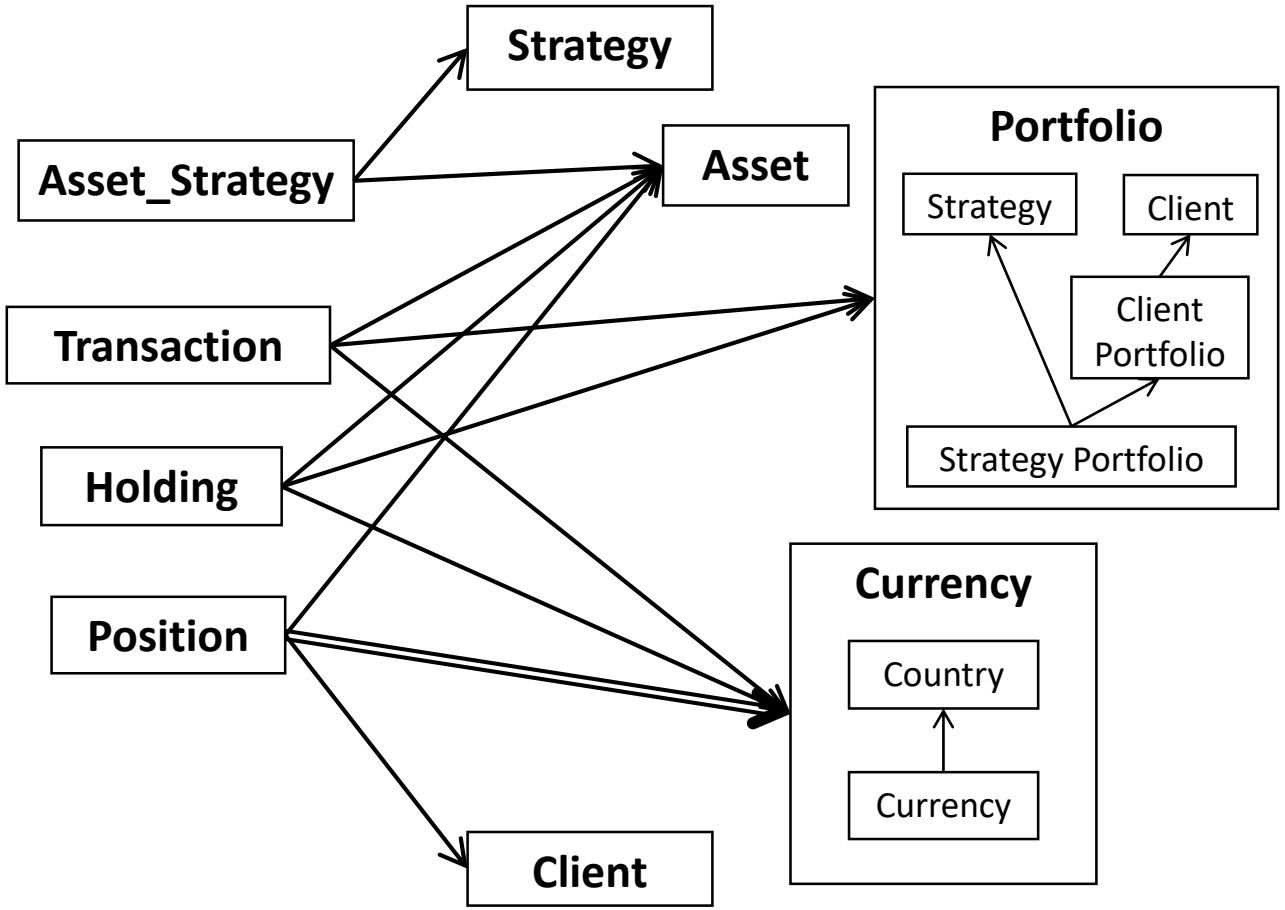
1. Asset_Strategy
2. Transaction
3. Holding
4. Position

Dimensions

1. Asset
2. Strategy
3. Portfolio (includes both portfolios, strategy and client)
4. Client



Data Warehouse (Alternative Schema)



Conforming portfolio across holding and transaction will provide insight in how labeled nulls work.

Bus Matrix (Alternative Schema)

	Strategy	Asset	Portfolio	Client	Currency
Asset Strategy	X	X	X		
Transaction		X	X	X	X
Holding	X	X	X	X	X
Position		X	X	X	X

One could consider adding a references from holding to client and strategy for performance reasons. This is related to how optimizers implement star joins using bitmap indexes. Also one could conform both position and asset strategy to make use of the portfolio dimension at the cost of creating a lot of labeled nulls. These two alternatives are indicated with the gray X's. It might be interesting to illustrate these alternatives in AQL.

BTW notice the similarity of the above bus matrix to the one from the existing AQL file.

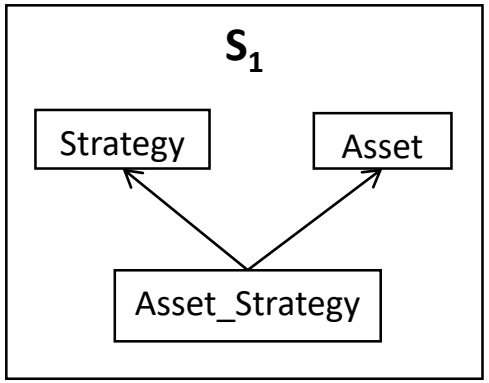
*	Currency	Asset	Portfolio	Client
Transaction	X	X	X	
Client Portfolio Holding	X	X	X	X
Client Position	X	X		X

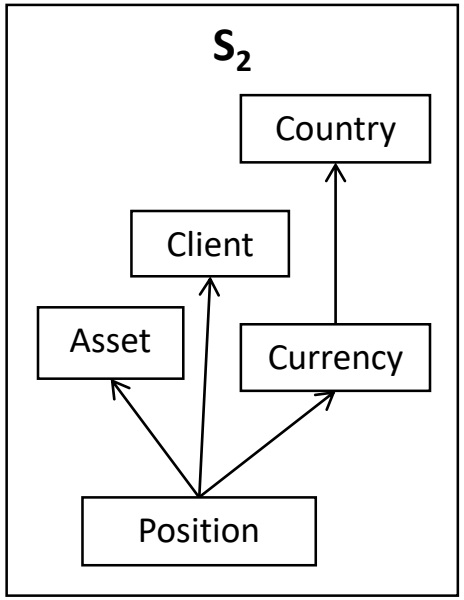
* Copied from slide 10.

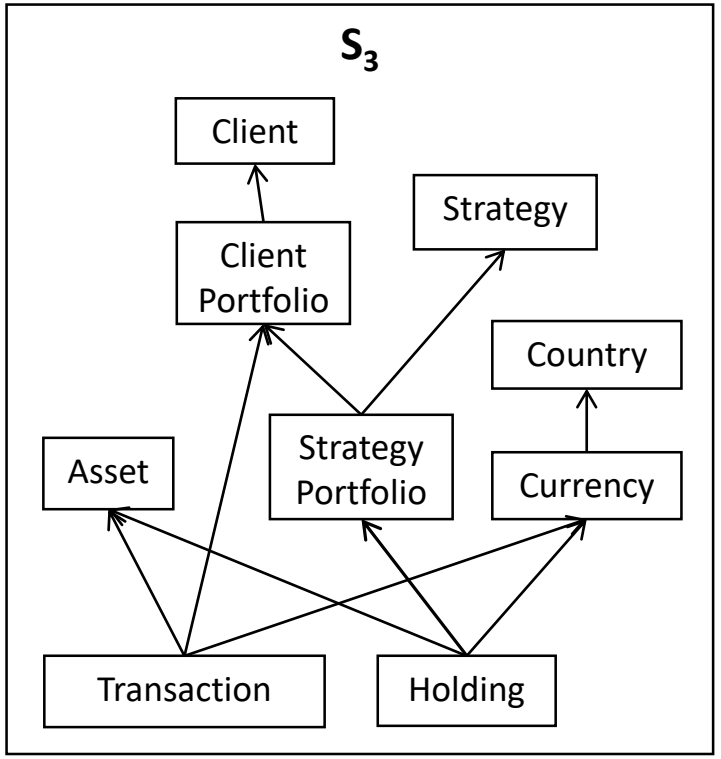
Some Thoughts on the Alternative Schema

- The original Finance Colimit seems like an artificial deconstruction of entities.
- There is a logical breakdown for defining different sources by fact table for the alternative schema.
- The integration problem then becomes one of constructing conforming dimensions from the versions of the dimensions that come from each source.
- In the following two slides I describe two different integration scenarios based on slide 17

Integration Scenario #1: Source Schemas



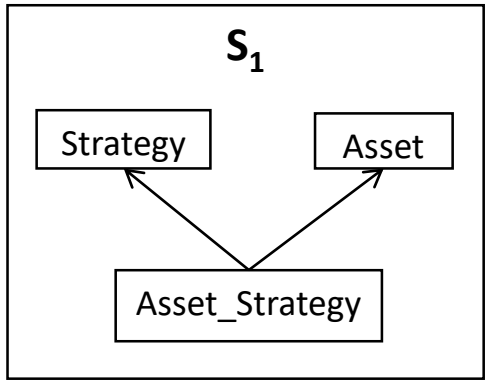


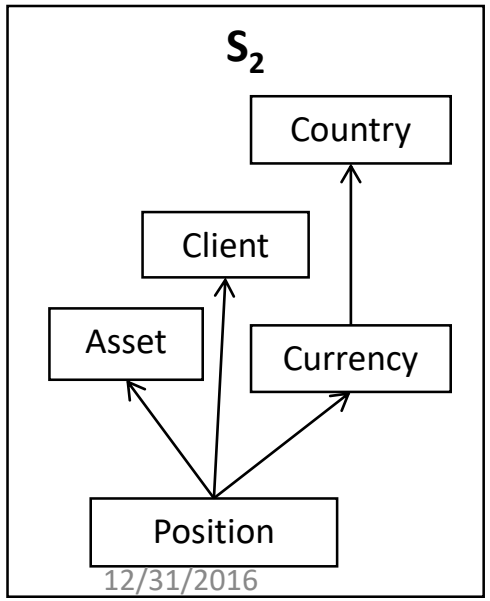


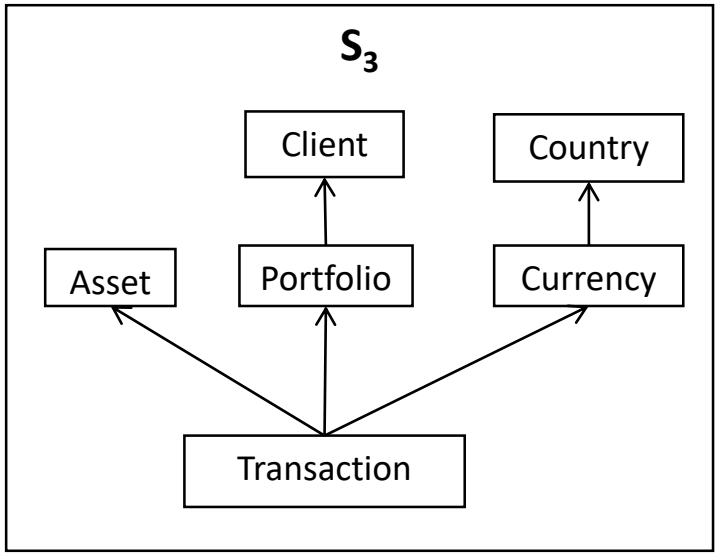
Integration Scenario #1

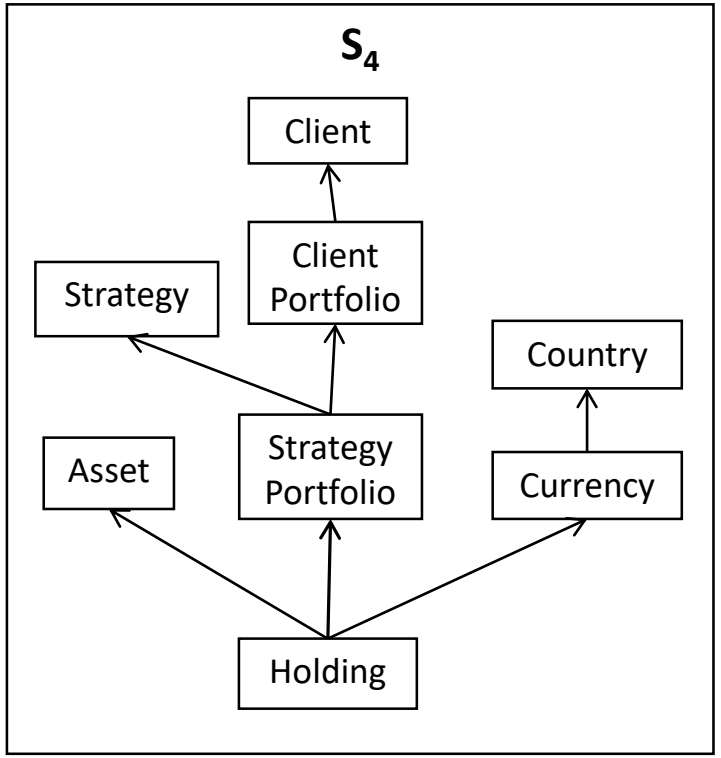
- Source S_1 – Asset Strategy fact table comes with a reference list of assets and strategies. The fact table defines how one distributes investments across the assets in the strategy.
- Source S_2 – Position fact together with currency, asset and client reference entities.
- Source S_3 – Transaction and holding facts come from the same source referencing currency and asset. In addition they both reference a standardized model for portfolio, strategy and client.
- Integration would require merging entities across the sources to define conforming dimensions.
  - Standardizing the currency models between S_2 and S_3.
  - Standardizing Clients across S_2 and S_3.
  - Standardizing Strategies across S_1 and S_3.
  - Standardizing Assets across all three sources.

Integration Scenario #2: Source Schemas









Integration Scenario #2

- Sources S_1 and S_2 – Same as scenario #1.
- Sources S_3 and S_4 – Are a deconstruction of S_3 of scenario #1.
- Integration would require merging entities across the sources to define conforming dimensions.
  - Scenario #2 would inherit all of the implementations of scenario #1.
  - In addition scenario #2 would require the reconciliation of the two models for portfolio in S_3 and S_4.