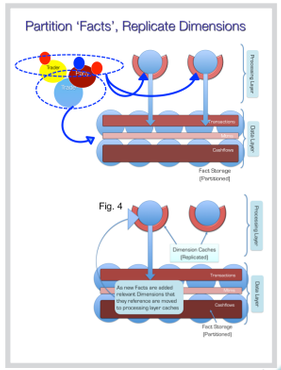
ODC

Join Engine

A write to ODC triggers a distributed, recursive mechanism for tracking reference data connected to the trade being written. This ripples from one cache to another, flagging relations (arcs) that exist in the active data set. Such 'connected' entities are then replicated to the query engines so that joins can be performed efficiently in memory, without needing to replicate all reference data for the purpose of joins (or making multiple network calls.



Notifications

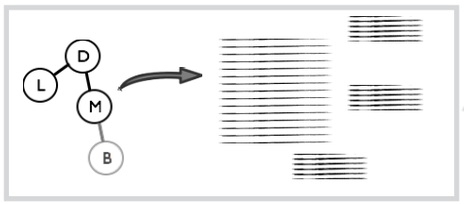
All data in ODC is an event. Clients can listen to data written to the store using this to drive their processing.

Fluent APIs

Queries can be expressed using a fluent Java interface or equivalent http call. The query engine supports advanced features such nested ANDs and ORs as well as ordering of distributed objects, with pagination.

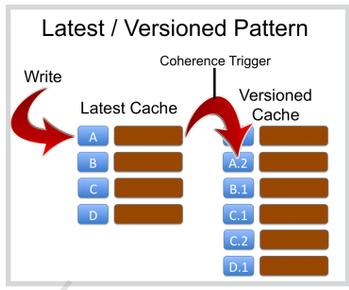
SQL/Relational Projection

Native, distributed support for the major elements of the SQL '99 dialect presented through a dynamically generated schema. You can query all ODC entities from trades through reference data in the familiar way, with data returned as rows an columns rather than graphs.



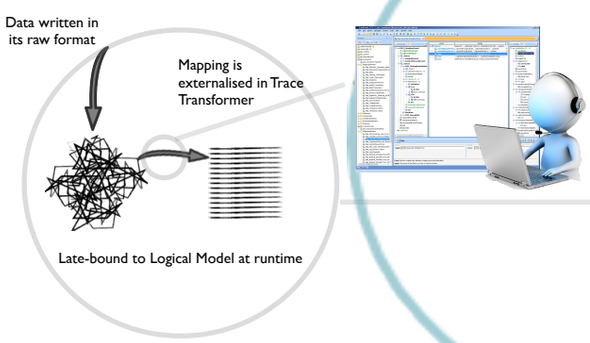
Immutable, Bi-temporal Data with Business Keys

Data is recorded as a stream of immutable entries. Can be accessed via the Latest (using a business key) or Historic views (which use a business key + version).



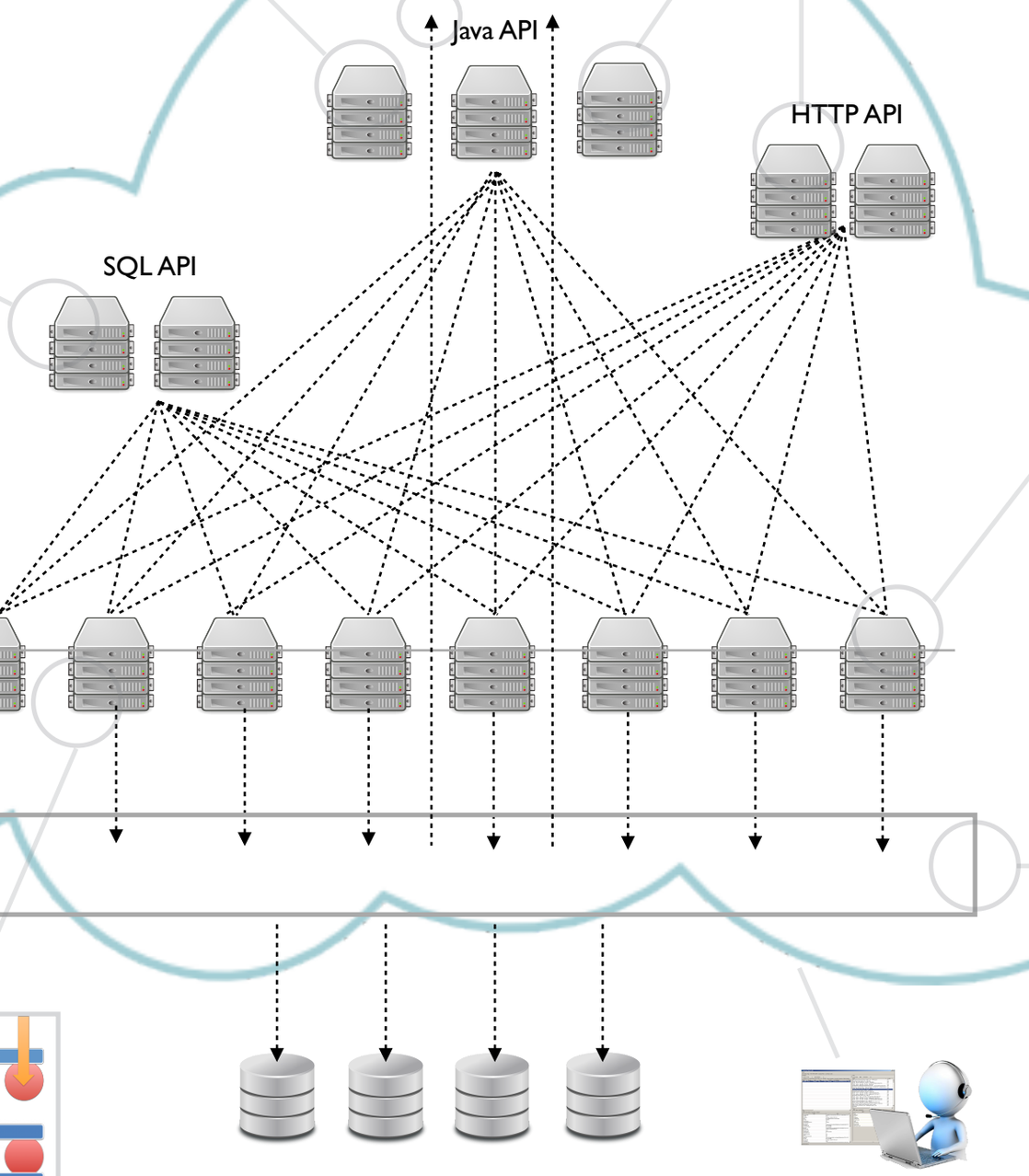
Data Held in Raw Form

All data is held the way it came in. It is not transformed before it is persisted. This prevents issues that result from 'taking a view' on inbound data. Instead the view is taken at runtime using the Late-Bound-Schema



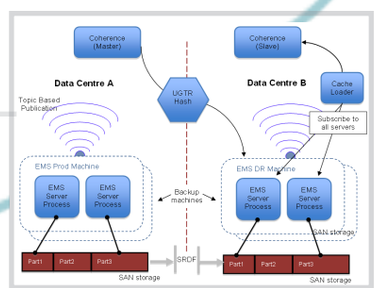
Late-Bound Schema

ODC holds data strictly in the form it was brought in. The view presented to users, which matches the bank's Logical Model, is bound at runtime (i.e. not persisted in the mapped form). The translation process is externalised using a tool called Trace Transformer, which is operated wholly and entirely by the Analysis team. The releases of changes to these mappings is not tied to software releases.



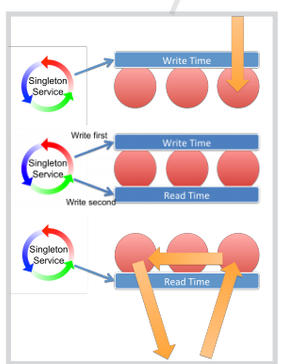
Messaging as a System of Record

Topic based messaging is used to provide a scalable, routable persistence stream. All data written to ODC is recorded through a write to one of the EMS servers via the Persistence Topic. These events can be reinterpreted as necessary for relational persistence, DR or even by clients.



Cluster-Consistent Time

A proprietary mechanism for providing reliable snapshot time using a distributed clock. Read time and write time are separately authored. Write time > read time always. The wall clock time between ticks may vary.



Database Array

An array of relational databases provide historic reporting for ad hoc users through direct (DAL) access or a reporting tool like Spotfire. These run asynchronously to the NoSQL store. The database array can be any Oracle database, even one that is owned by a you!

Data Quality Management

A team of data analysts monitor ODC data looking for places where data does not join up from the different sources, for example trades having invalid books, counterparties etc.