ORACLE

ORACLE

Oracle Enterprise Data Quality

Integration Essentials

Oracle Product Development

Batch Integration

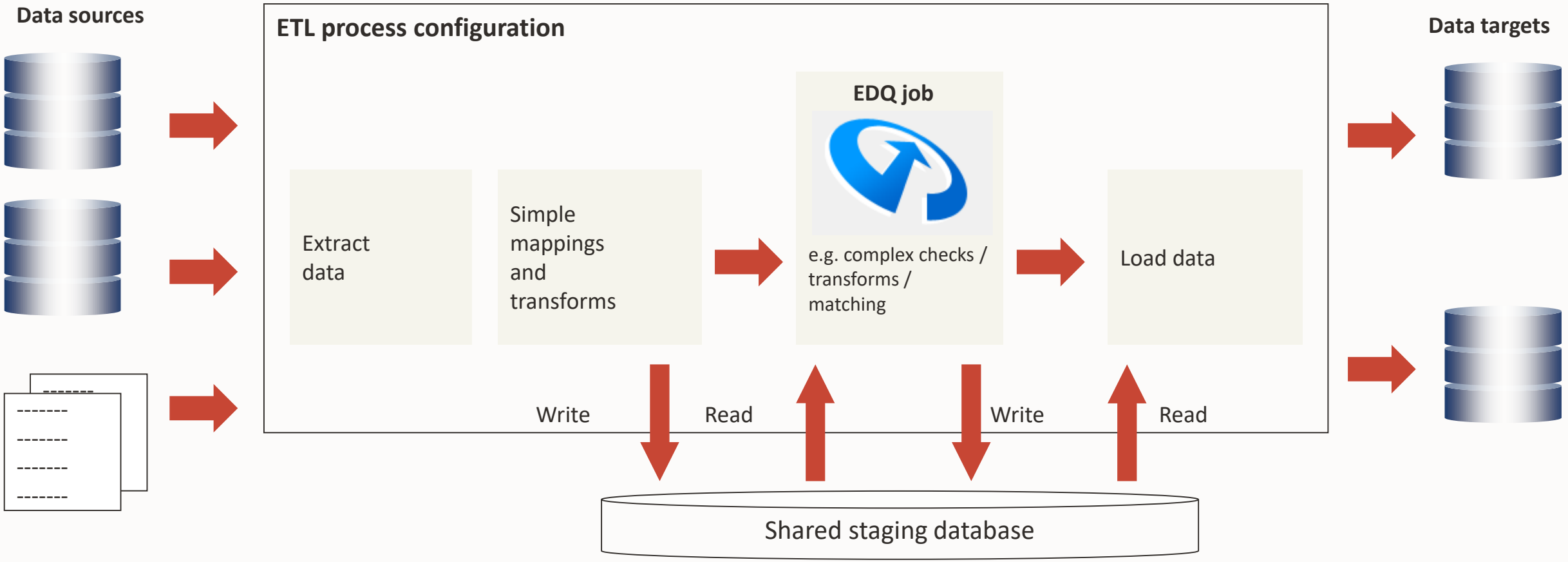
Three models of ETL integration/coexistence with EDQ

1. Batch processing, ETL masters process control
2. Batch processing, EDQ masters process control
3. Transaction processing, ETL masters process control

1 - Batch processing, ETL master

- An EDQ job is modelled into the ETL tool as a single step, called using a command line interface or the REST API
- The job writes its results to a staged data area or files, with shared access
- The job returns information when it has finished
- ETL tool then continues to the next step, which reads the output data written by the job from the shared staged data area
- Oracle Data Integrator has a built-in tool that can call EDQ jobs either synchronously or asynchronously at any point in a data flow

1 - Batch processing, ETL master

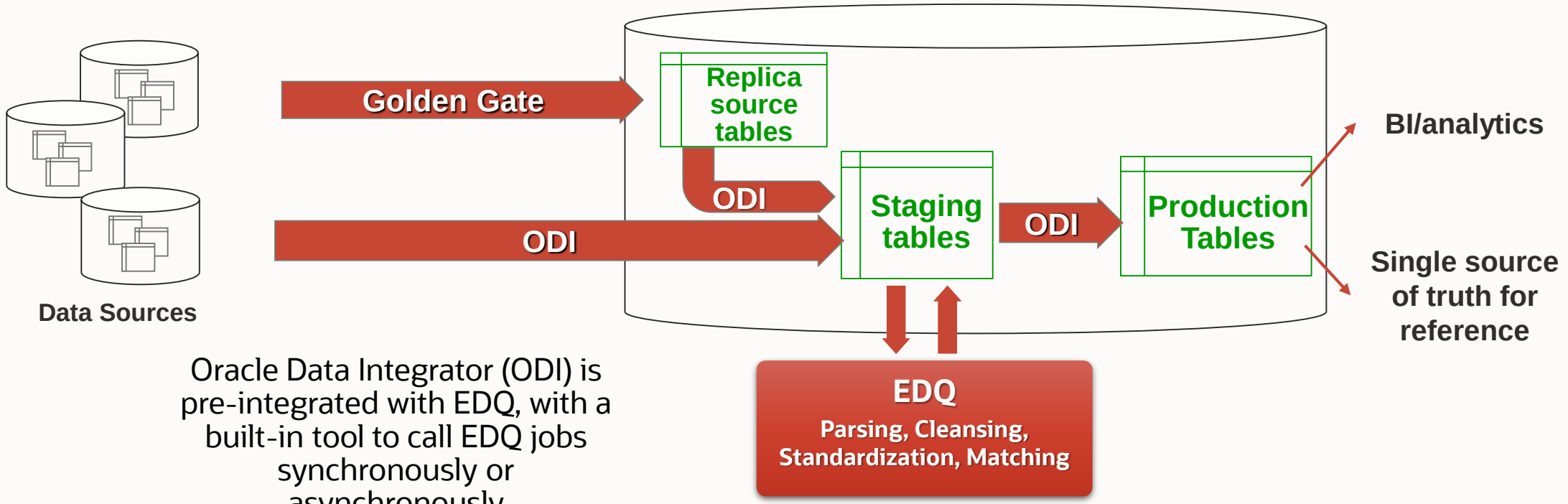


1 - Batch processing, ETL master with Oracle Data Integrator

1. ODI or GoldenGate move data into DW
2. ODI maps data into Staging tables
3. EDQ 'cleanses' data
4. ODI moves from staging into production tables

Data in DW is 'Fit for Purpose'

- BI/Analytics
- Source of truth

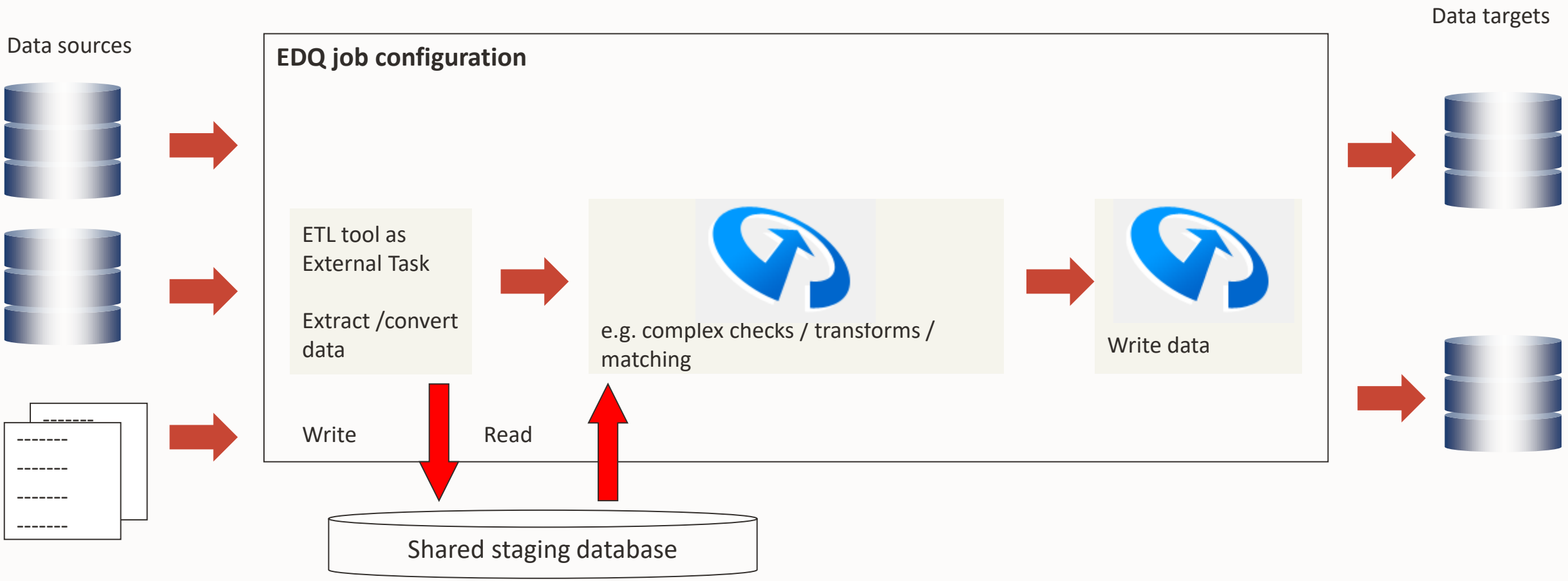


Oracle Data Integrator (ODI) is pre-integrated with EDQ, with a built-in tool to call EDQ jobs synchronously or asynchronously

2 - Batch processing, EDQ master

- An EDQ job includes all required processing
- External Tasks in the job are used for any callouts, for example to ETL
- Where required, shared staging is used
- Jobs will commonly use externalized options so that the files/tables to process, and those to write, can be specified using command line or REST API options/overrides, or a stored Run Profile
- Most commonly used where EDQ performs ‘most of’ the ETL, with occasional callouts to other tools, for example for legacy systems (mainframes etc.)

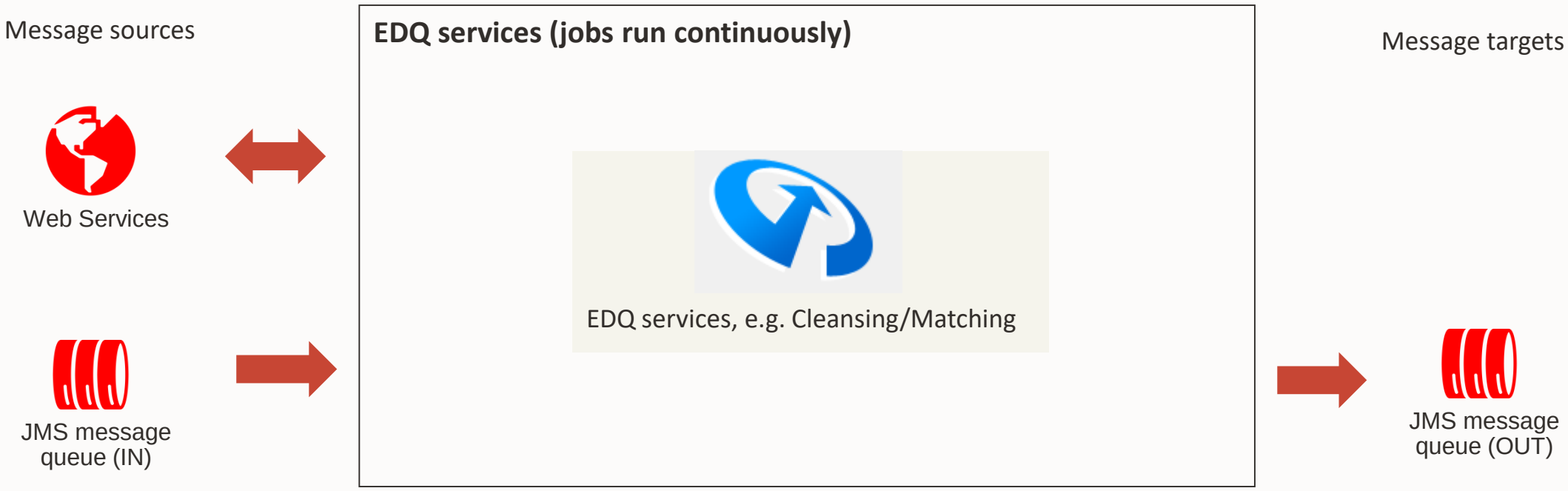
2 - Batch processing, EDQ master



3 - Transaction processing, ETL master

- EDQ jobs may be modelled using real-time architecture (Web Services or JMS) and run continuously to provide DQ services
- Jobs are normally set up to run whenever the EDQ server(s) are running
- Processes normally run continuously
- EDQ's real-time interface is preferable for small batches of records (<1000) as well as record streams, as it avoids startup costs for batch jobs

3 - Transaction processing, EDQ called over Real-Time interface



Real-Time Integration (SOA)

The Need for Real-time DQ Services

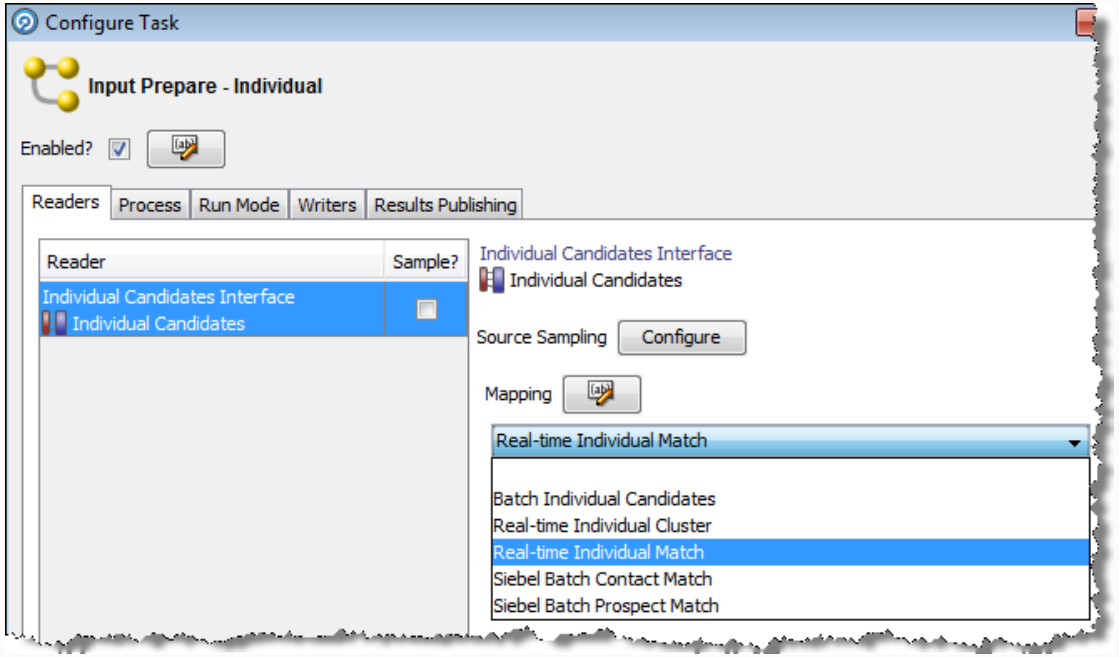
- **Most data quality initiatives begin with a project to improve existing information in batch**

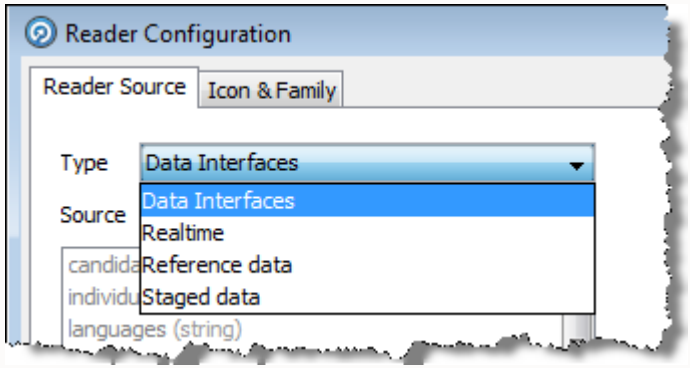
But

- **There is no sense in cleaning the lake if the upstream factory continues to pollute the river…**
- **EDQ delivers real-time DQ services from the same rules & configuration used in the batch processes**

Moving from Batch to Real-time

- **All EDQ processing is independent of the physical source of the data – e.g. database, file, Web Service, JMS message etc.**
- **Jobs can use the same processes, bound to either real-time or batch sources and targets, using Data Interface mappings**
- **The mapping can be overridden at runtime**
- **Or, just change Readers and Writers**





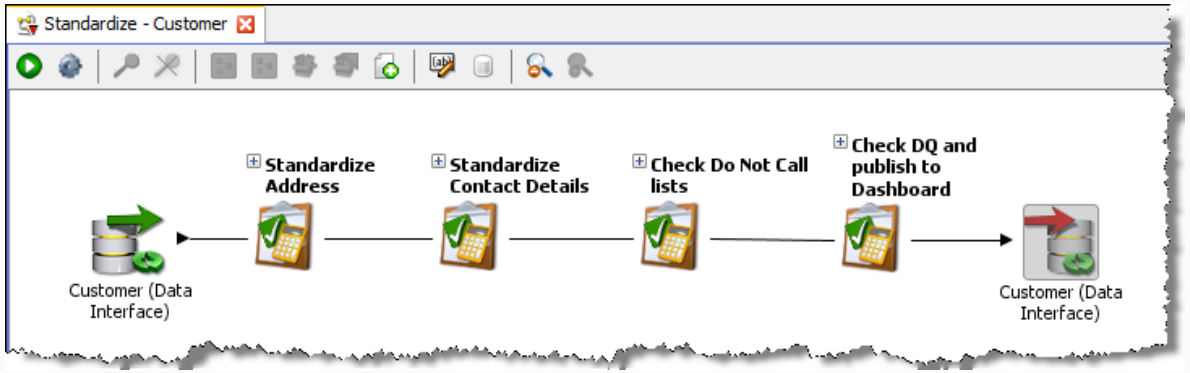
Real-time validation & standardization

- **Ensure new or changed data meets quality standards by validating & standardizing against your business rules:**

New or changed record passed to EDQ

Standardized & validated record returned

High quality data guaranteed and value of information asset is preserved



As your business evolves and your validation & standardization rules change, there is no change to the integration or the web service. New rules are simply configured in EDQ.

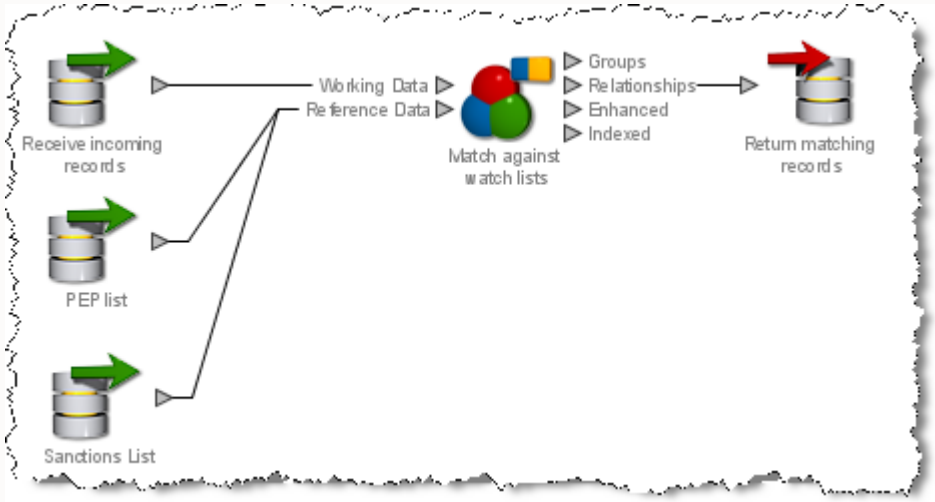
Real-time linking/enhancement

- **Check new records for matches to reference data, e.g. Watchlists:**

User adds a new record

Definite and possible matches are returned

User chooses how to update system, e.g. add, merge, or link



EDQ checks the data against regularly updated snapshots of the reference data, all of which are pre-structured for optimal performance.

Real-time duplicate prevention

- **Protect systems from duplicate records using EDQ Web Services:**

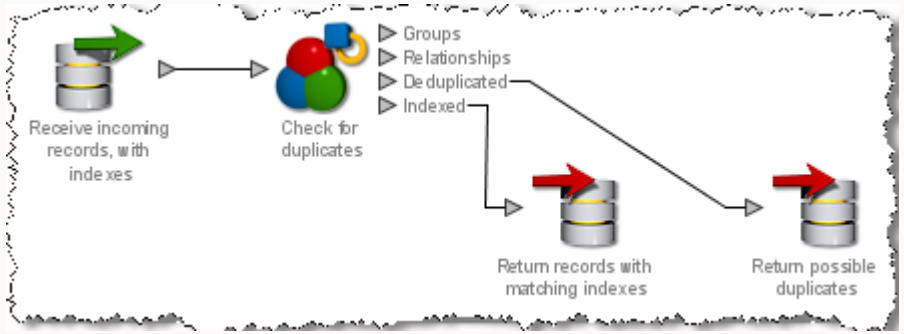
User adds a new record

EDQ returns key values

Candidate list for matching

Definite & possible matches are returned

User chooses how to update system, e.g. add, merge, or link



EDQ does not hold a copy of the data. Records are passed back and forth. This avoids complex data replication & synchronization issues.

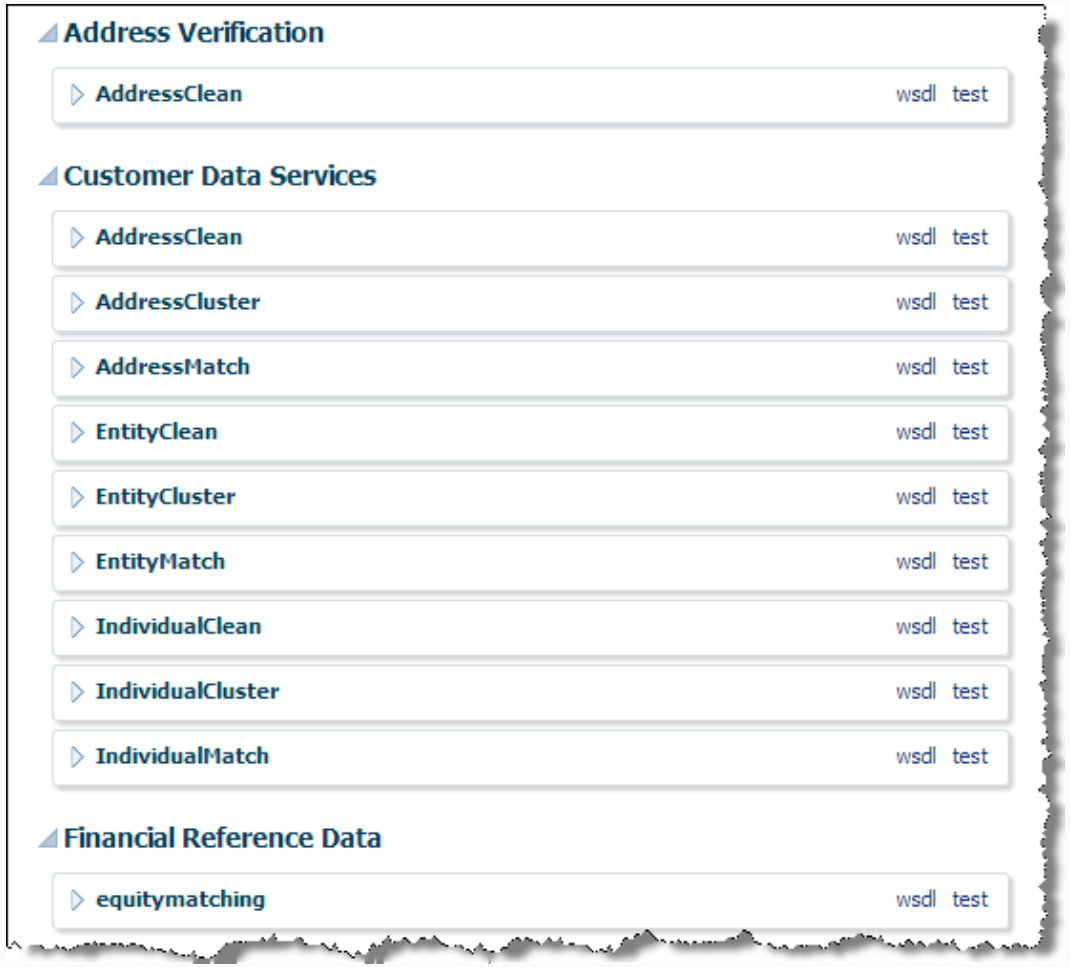
The calling application manages storage of all data, including the key values provided by EDQ which are used for match candidate selection (any record that shares any key value).

Benefits of EDQ Real-Time

- **Protects information assets from errors**
- **Uses the same rules & configuration as batch**
  - Minimizes configuration effort
  - Rules are consistently applied regardless of the source of the data
- **New rules are simply configured in EDQ**
  - No re-work of the integration required
- **High-quality information becomes the norm**

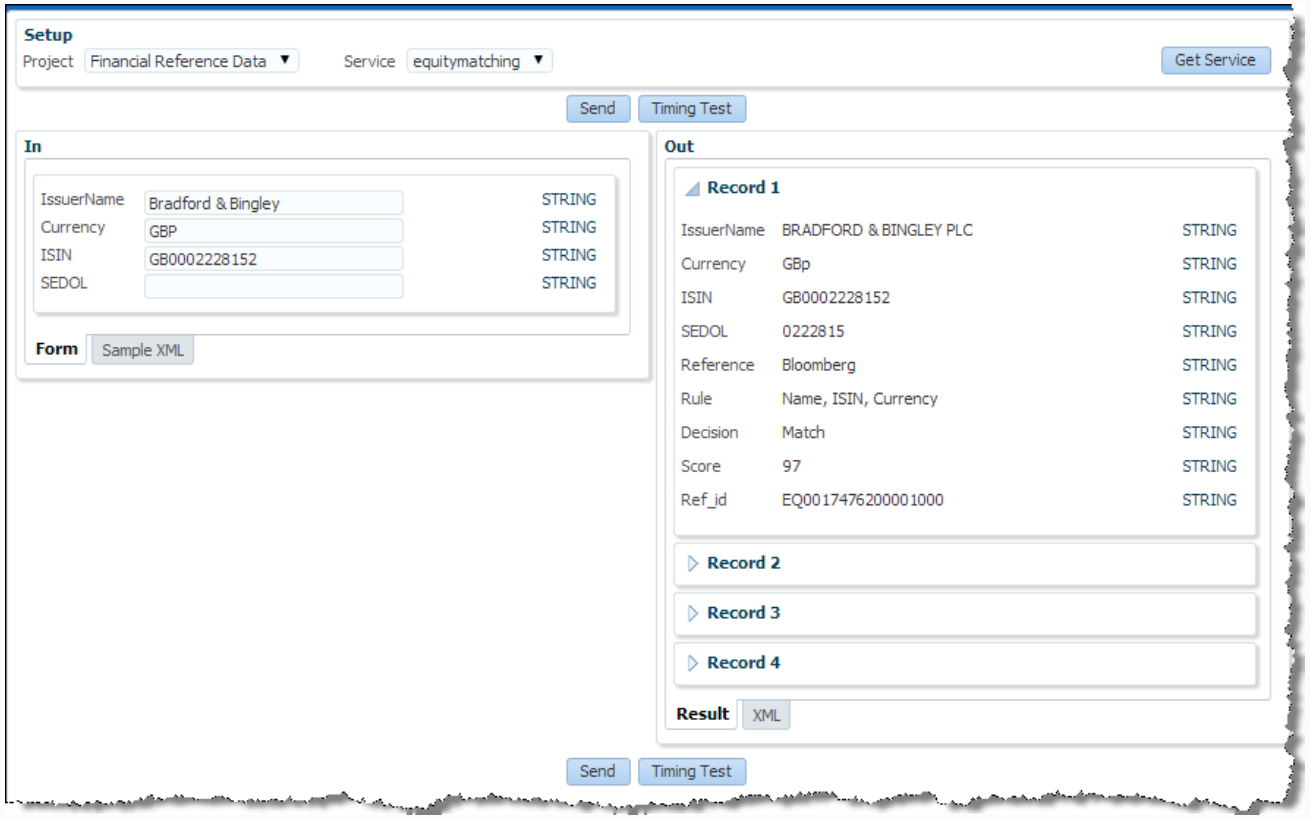
Using Web Services

- **Real time providers and consumers can be defined in the GUI as Web Services**
- **Web Services (and their WSDL files) are generated and kept up-to-date with any modifications**
- **Launchpad provides access to full list of Web Services on a server**
- **Both SOAP/XML and REST/JSON interfaces are generated so either may be used**



Testing Web Services

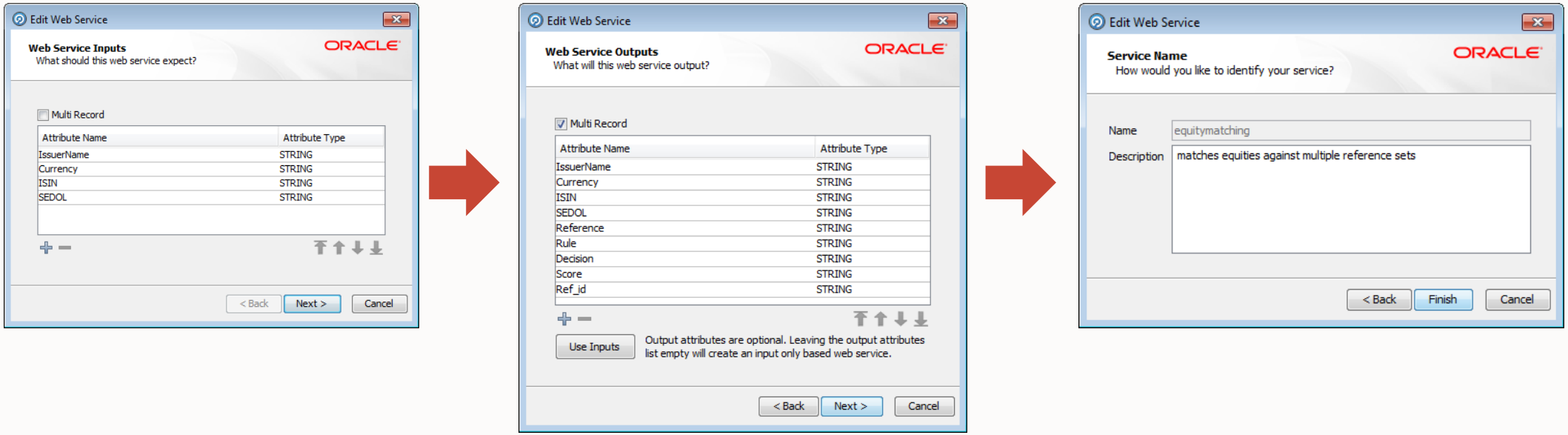
- **EDQ has a built-in UI for testing Web Services**



Web Service Generation

EDQ provides a fully GUI-controlled DQ Web Service management environment

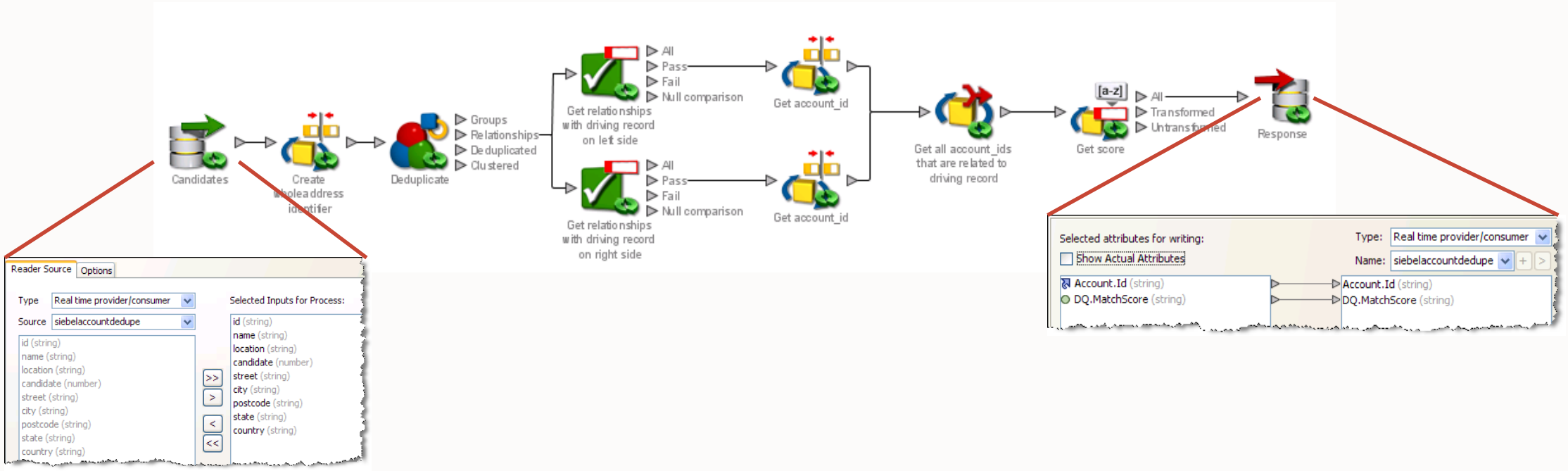
- Create and manage Web Services in EDQ Director:



Note: Multi-record support for Web Services where a single inbound or outbound message may contain many records, e.g. match candidates, and matches

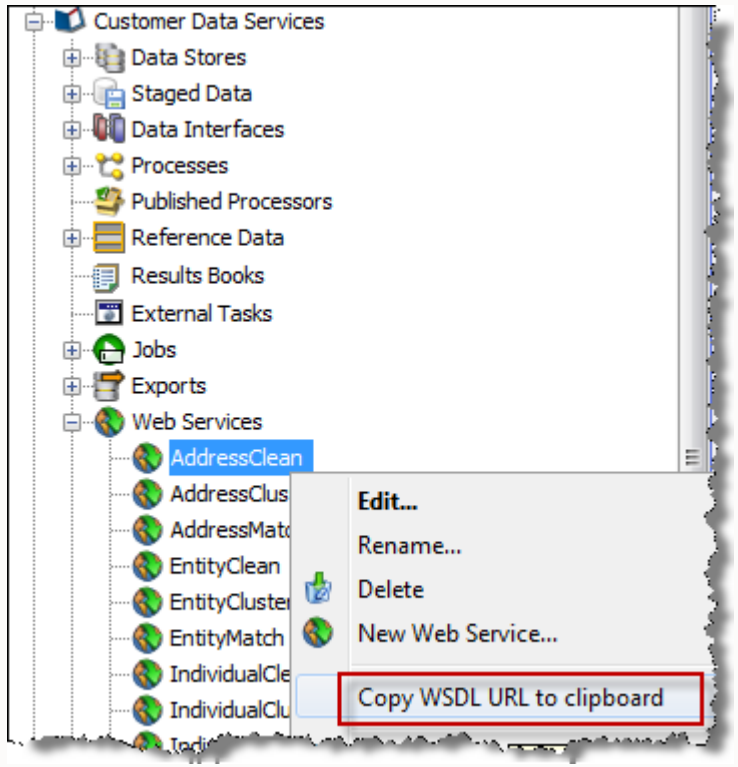
Web Service Generation

- Map the Readers and Writers in a process to the Web Service, which is now a configured real-time provider and consumer of records
- The same mappings can also be defined in a job, or at runtime



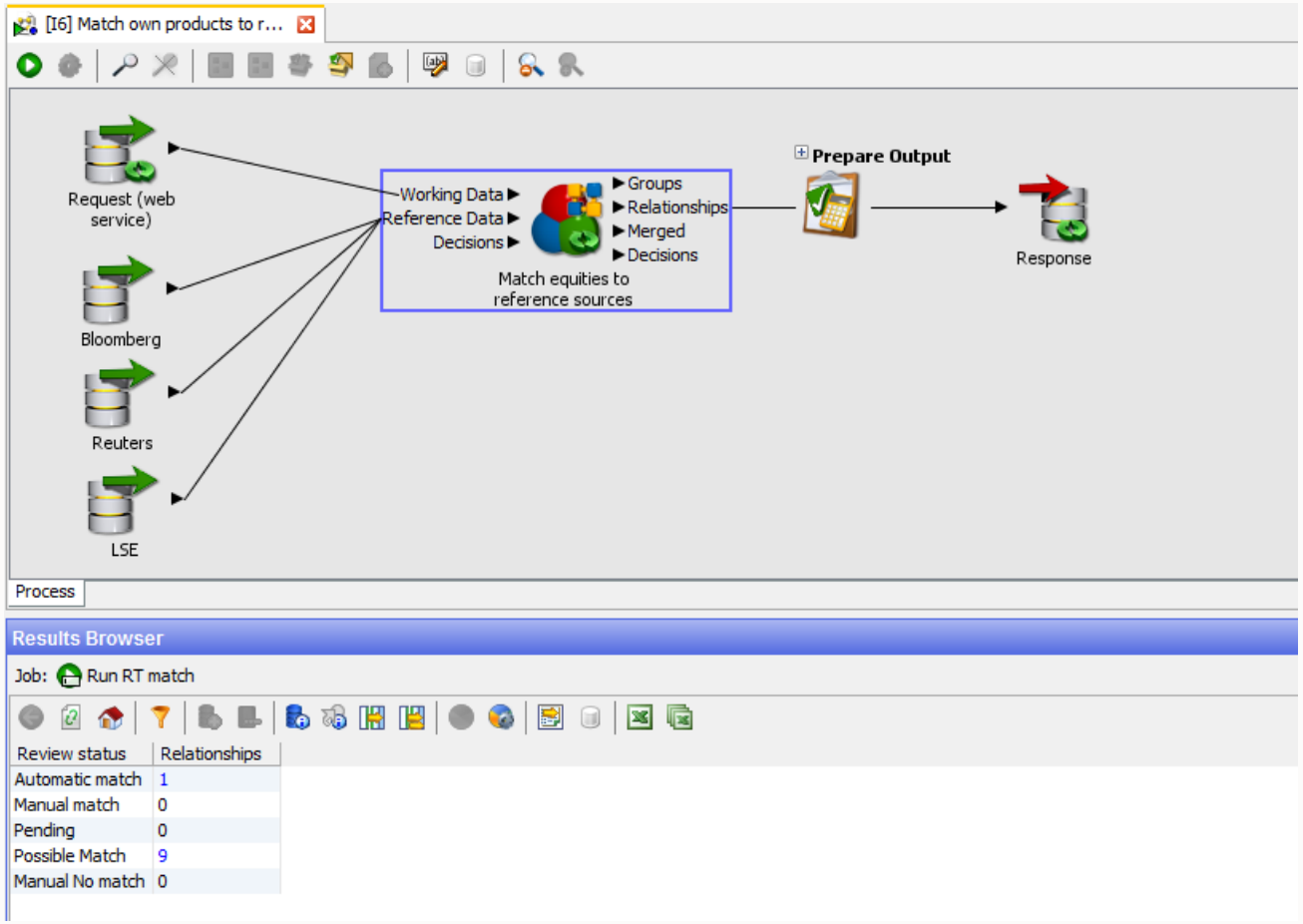
Web Service Generation

- To integrate, copy the URL of the generated WSDL file to the clipboard (SOAP/XML), or use the Launchpad to see all Web Services on the server and the generated REST documentation



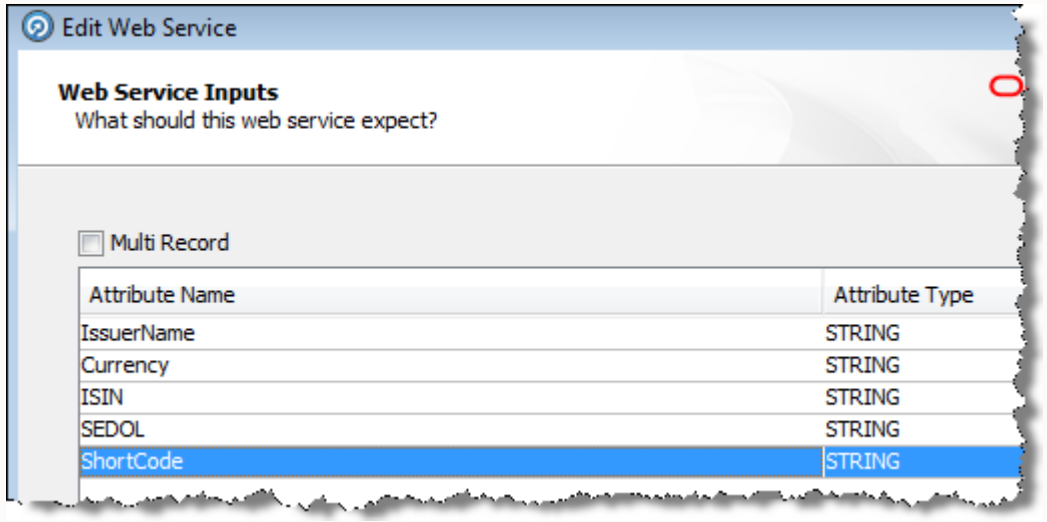
‘Running’ a Web Service

- Processes may be launched externally (e.g. on 3rd party application startup) and configured to write results periodically:



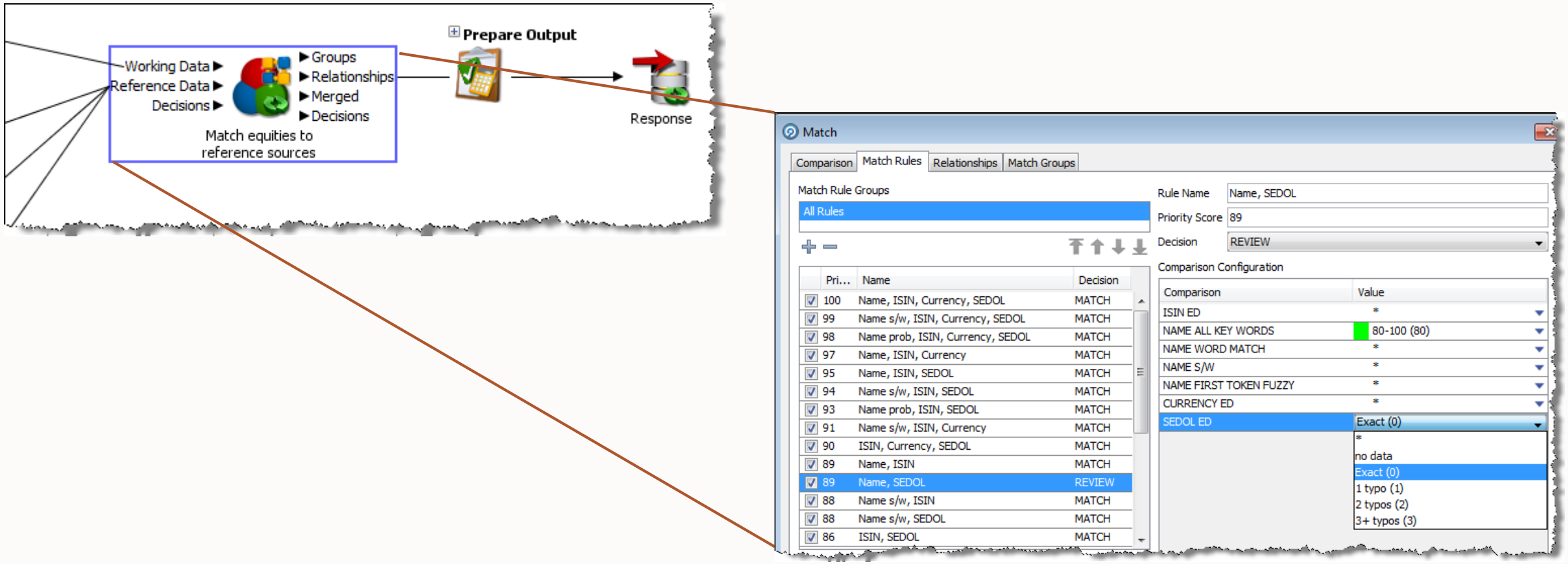
Editing interfaces

- To change the interface, e.g. to use a new field in matching, edit the Web Service in EDQ
- The WSDL and REST API will be updated automatically



Web Service Logic

- To change the logic used by a Web Service, change the process in EDQ:



Using JMS

- **Real time providers and consumer interfaces are defined using XML files on the EDQ server**
- **Process Readers and Writers are then wired up to these interfaces using the GUI (as with Web Services)**
- **JMS allows connectivity to nearly all Middleware and Message Queueing technologies**

Our mission is to help people
see data in new ways, discover
insights, unlock endless possibilities.