



ORACLE

ORACLE

Oracle Enterprise Data Quality

Integration Essentials

Oracle Product Development

Batch Integration

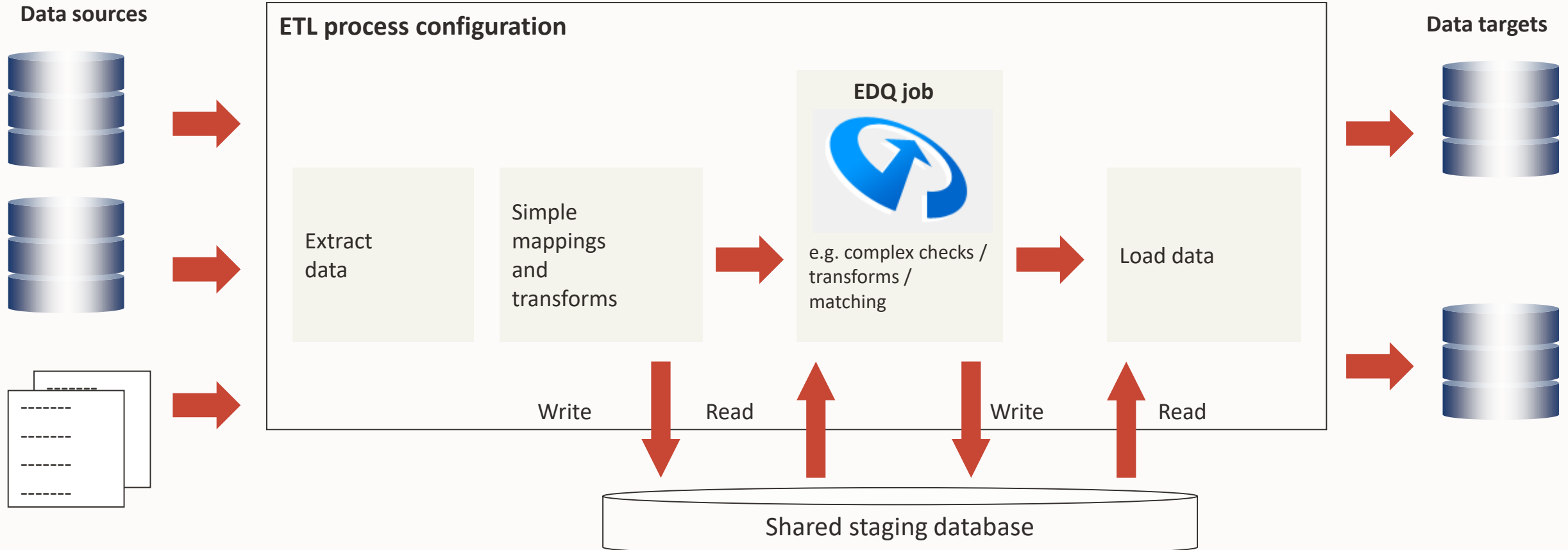
Three models of ETL integration/coexistence with EDQ

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

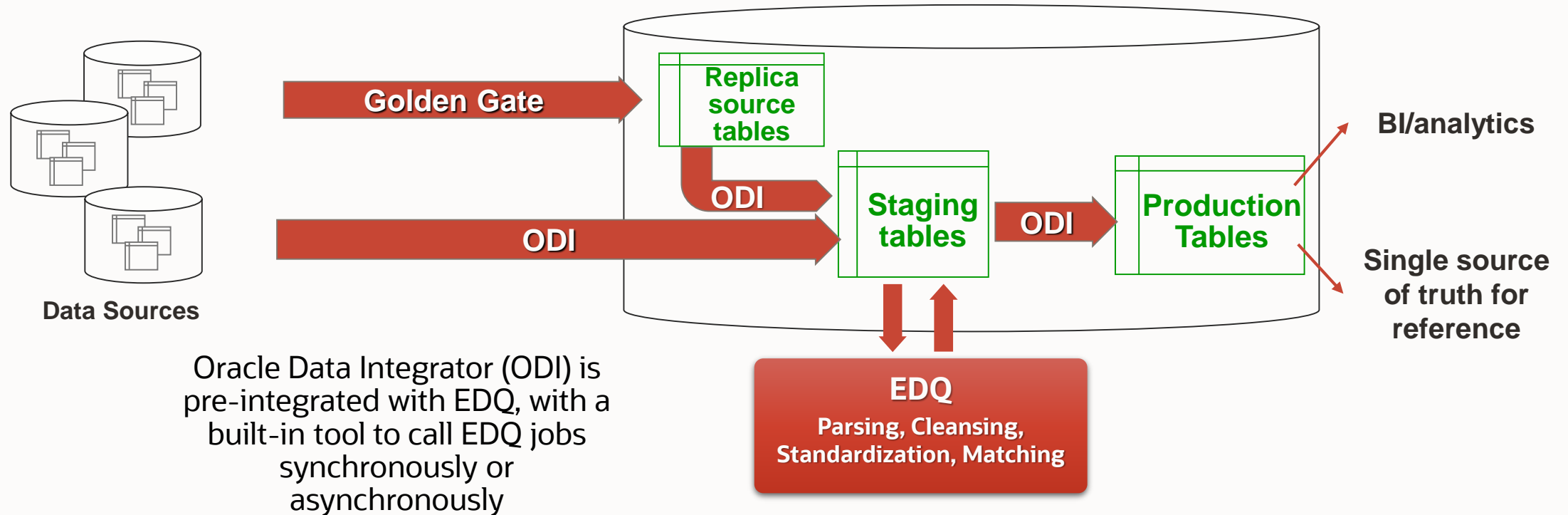


1 - Batch processing, ETL master with Oracle Data Integrator

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

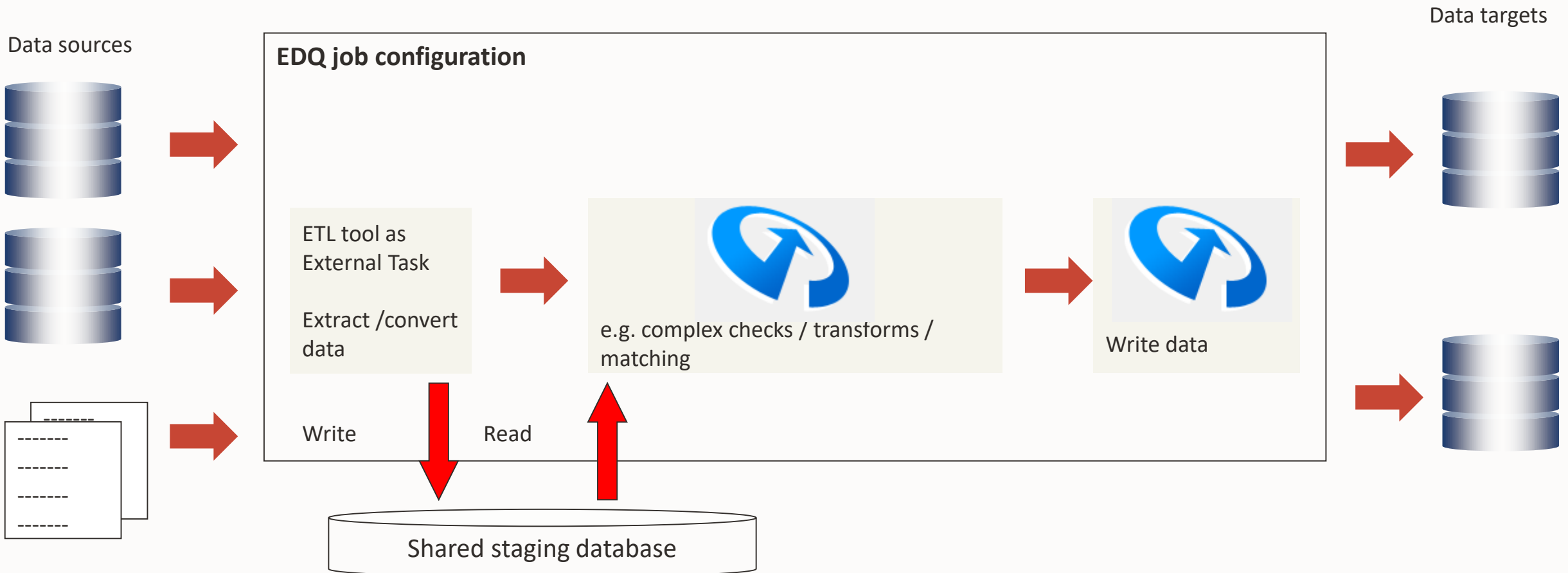


Oracle Data Integrator (ODI) is pre-integrated with EDQ, with a built-in tool to call EDQ jobs synchronously or asynchronously

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.)**

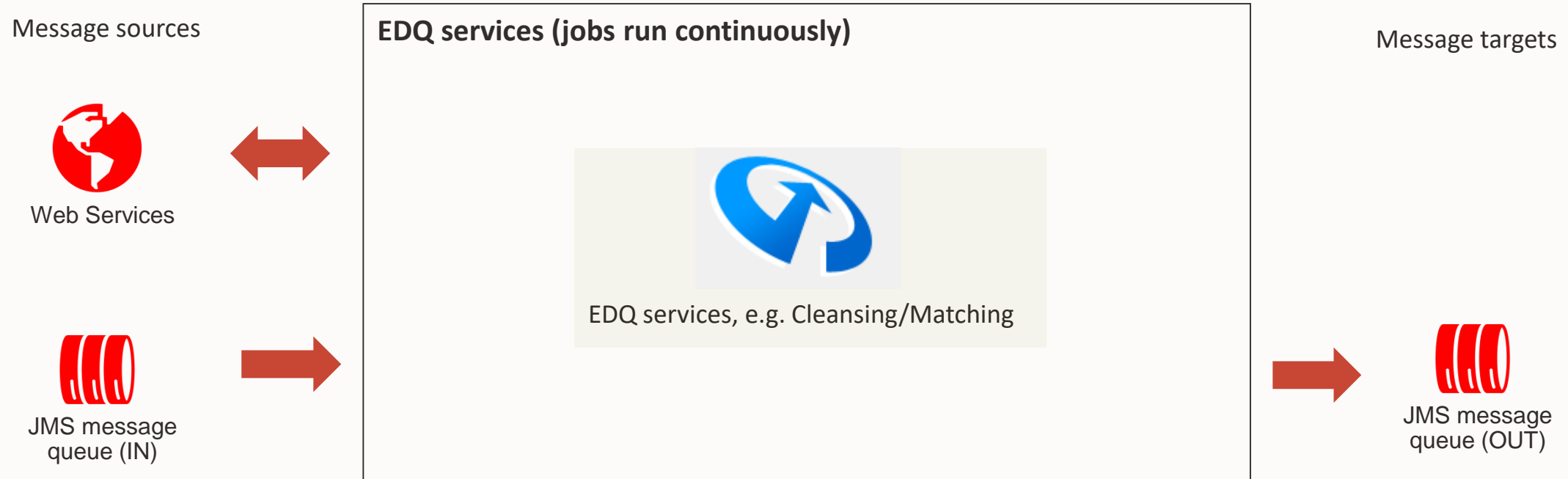
2 - Batch processing, EDQ master



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**

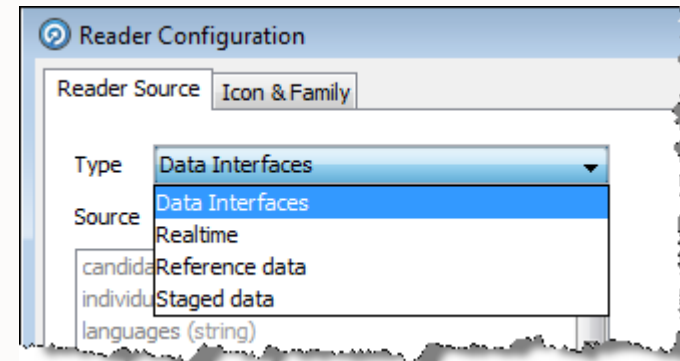
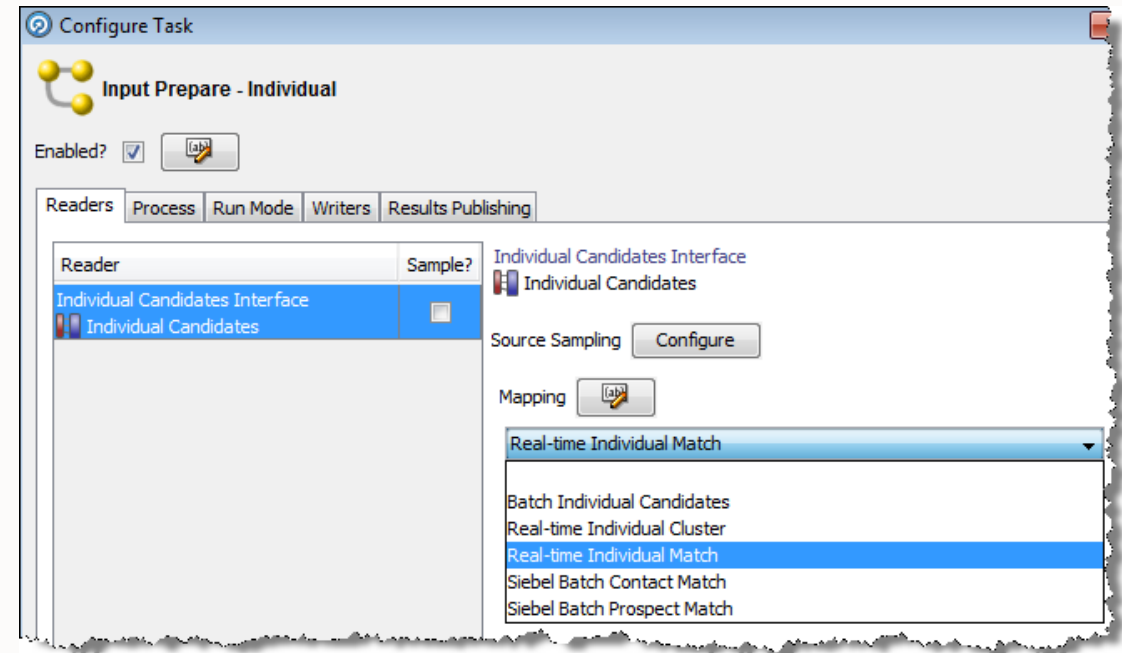
But

- **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**



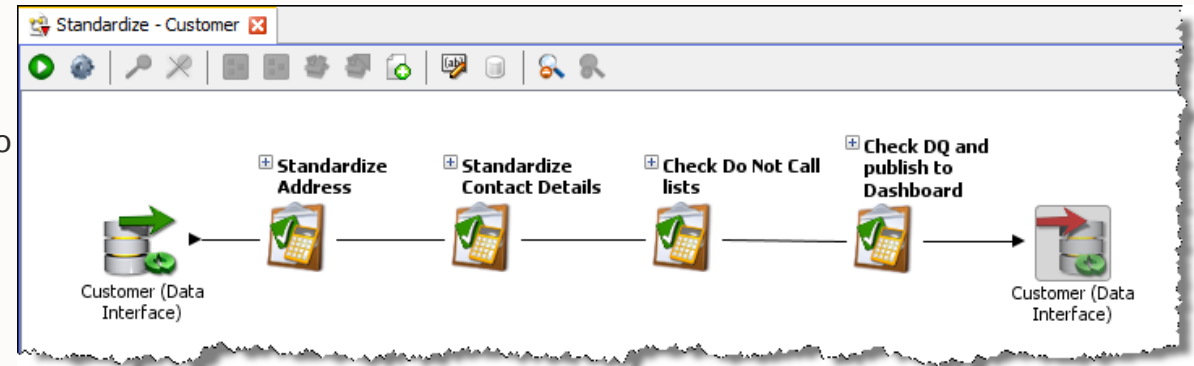
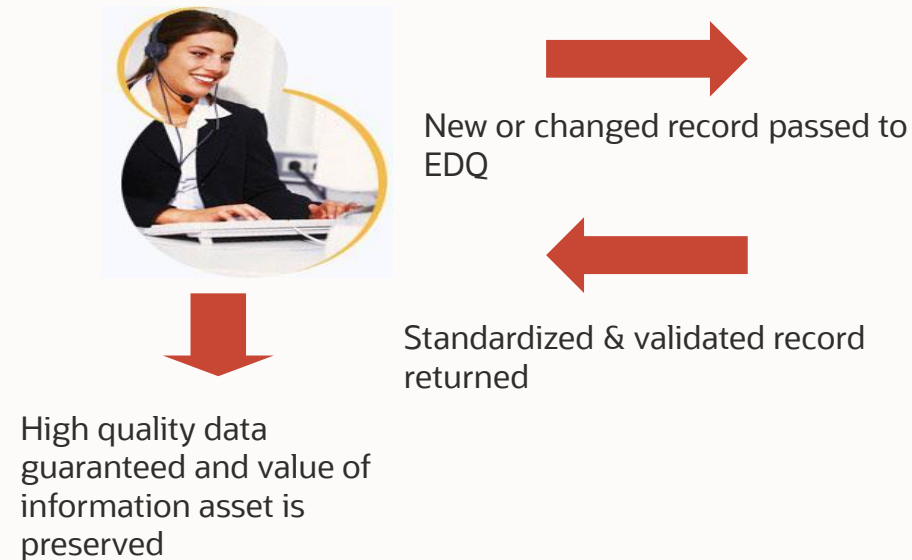
Moving from Batch to Real-time

- 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:**



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:

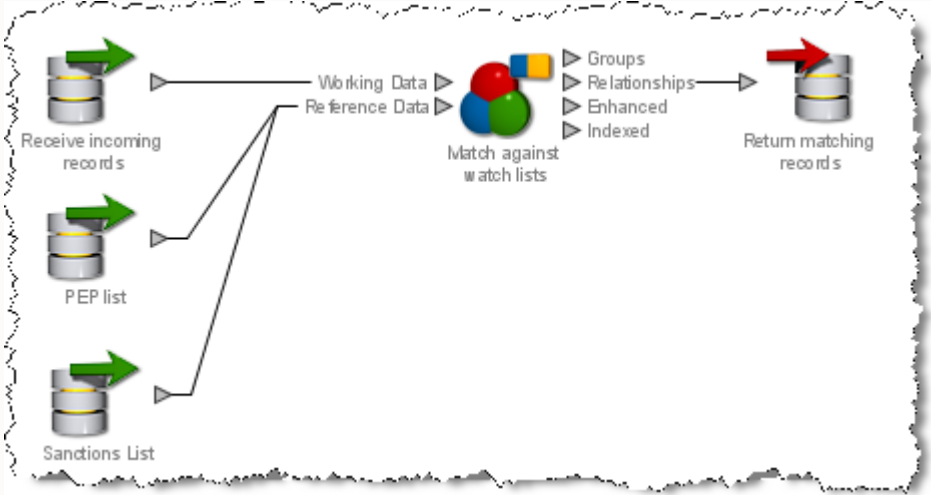


→ User adds a new record



↓ Definite and possible matches are returned

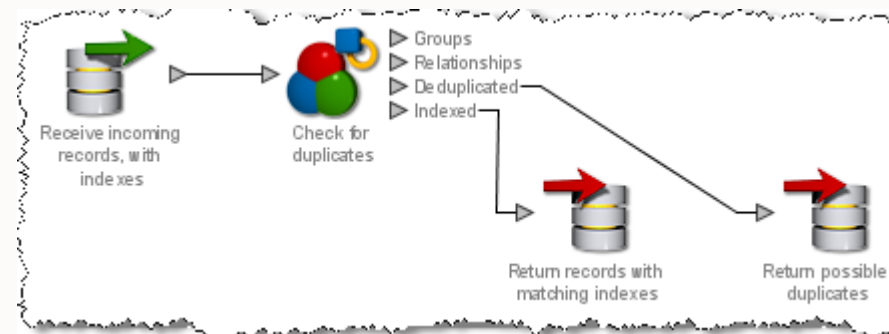
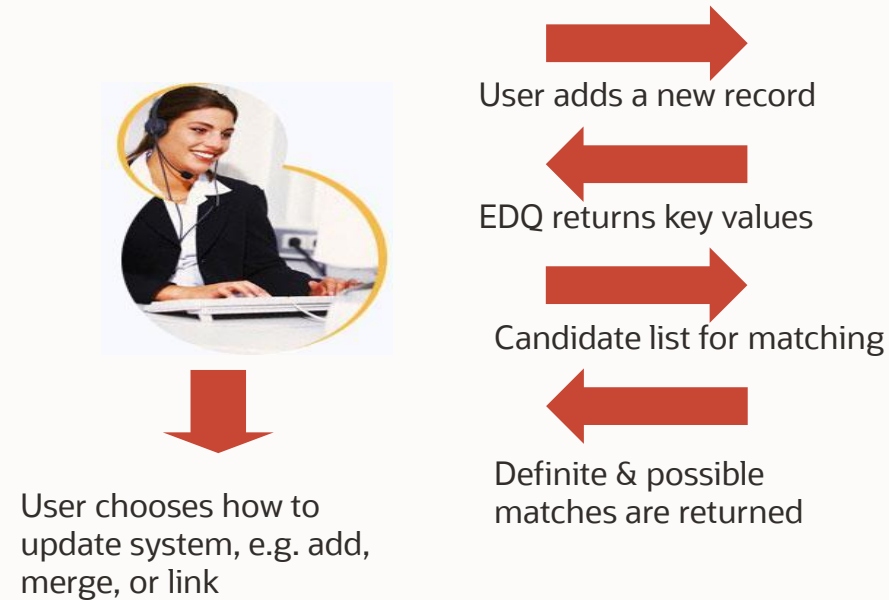
User chooses how to update system, e.g. add, merge, or link



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:**



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

Category	Service Name	Action
Address Verification	AddressClean	wsdl test
Customer Data Services	AddressClean	wsdl test
	AddressCluster	wsdl test
	AddressMatch	wsdl test
	EntityClean	wsdl test
	EntityCluster	wsdl test
	EntityMatch	wsdl test
	IndividualClean	wsdl test
	IndividualCluster	wsdl test
IndividualMatch	wsdl test	
Financial Reference Data	equitymatching	wsdl test

Testing Web Services

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

The screenshot displays the EDQ web service testing interface. At the top, the 'Setup' section includes a 'Project' dropdown set to 'Financial Reference Data' and a 'Service' dropdown set to 'equitymatching'. A 'Get Service' button is located to the right. Below this, there are 'Send' and 'Timing Test' buttons.

The 'In' section contains a form with the following fields:

IssuerName	Bradford & Bingley	STRING
Currency	GBP	STRING
ISIN	GB0002228152	STRING
SEDOL		STRING

Below the form is a 'Form' section with a 'Sample XML' button.

The 'Out' section displays the results of the test, showing four records. The first record is expanded:

IssuerName	BRADFORD & BINGLEY PLC	STRING
Currency	GBP	STRING
ISIN	GB0002228152	STRING
SEDOL	0222815	STRING
Reference	Bloomberg	STRING
Rule	Name, ISIN, Currency	STRING
Decision	Match	STRING
Score	97	STRING
Ref_id	EQ0017476200001000	STRING

Below the first record are three collapsed records labeled 'Record 2', 'Record 3', and 'Record 4'. At the bottom of the 'Out' section, there is a 'Result' section with an 'XML' button. Finally, there are 'Send' and 'Timing Test' buttons at the very bottom of the interface.

Web Service Generation

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

- Create and manage Web Services in EDQ Director:

The image shows three sequential screenshots of the 'Edit Web Service' GUI in Oracle EDQ Director, connected by red arrows. The first screenshot shows the 'Web Service Inputs' tab with a table of input attributes. The second screenshot shows the 'Web Service Outputs' tab with a table of output attributes and a 'Use Inputs' button. The third screenshot shows the 'Service Name' tab with fields for Name and Description.

Web Service Inputs
What should this web service expect?

Attribute Name	Attribute Type
IssuerName	STRING
Currency	STRING
ISIN	STRING
SEDOL	STRING

Web Service Outputs
What will this web service output?

Attribute Name	Attribute Type
IssuerName	STRING
Currency	STRING
ISIN	STRING
SEDOL	STRING
Reference	STRING
Rule	STRING
Decision	STRING
Score	STRING
Ref_id	STRING

Service Name
How would you like to identify your service?

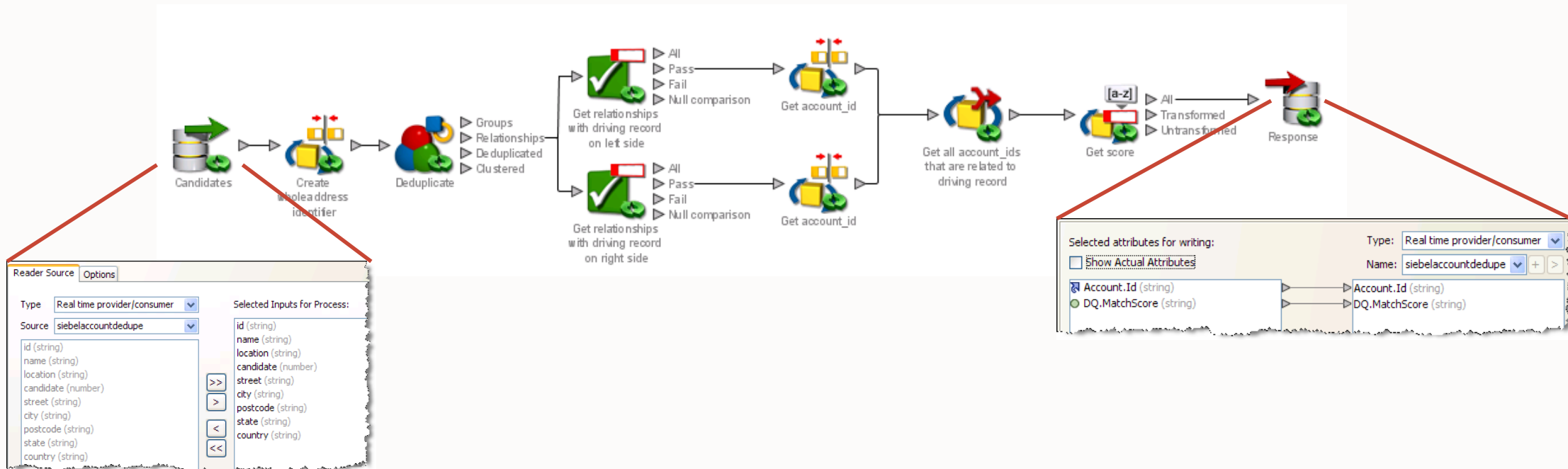
Name: equitymatching

Description: matches equities against multiple reference sets

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

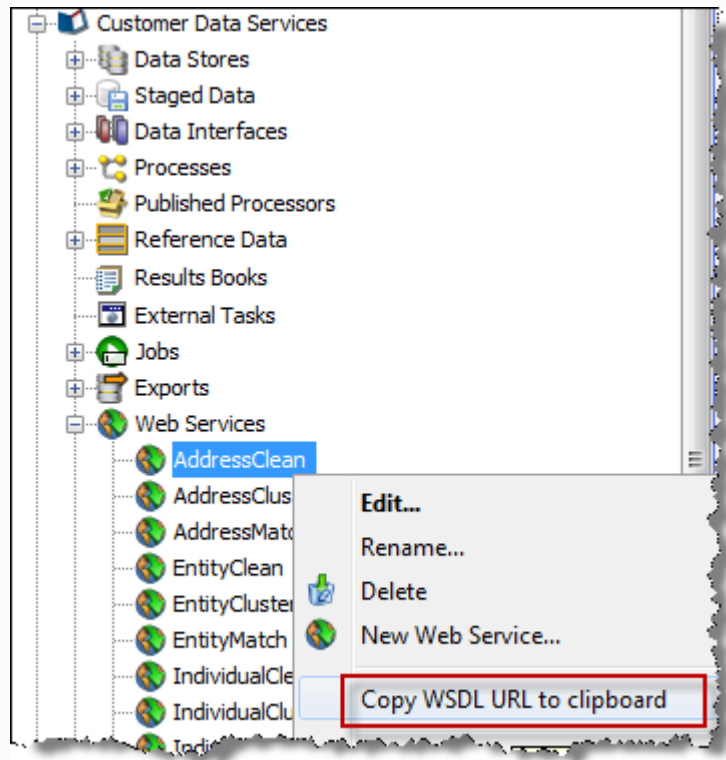
Web Service Generation

- 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



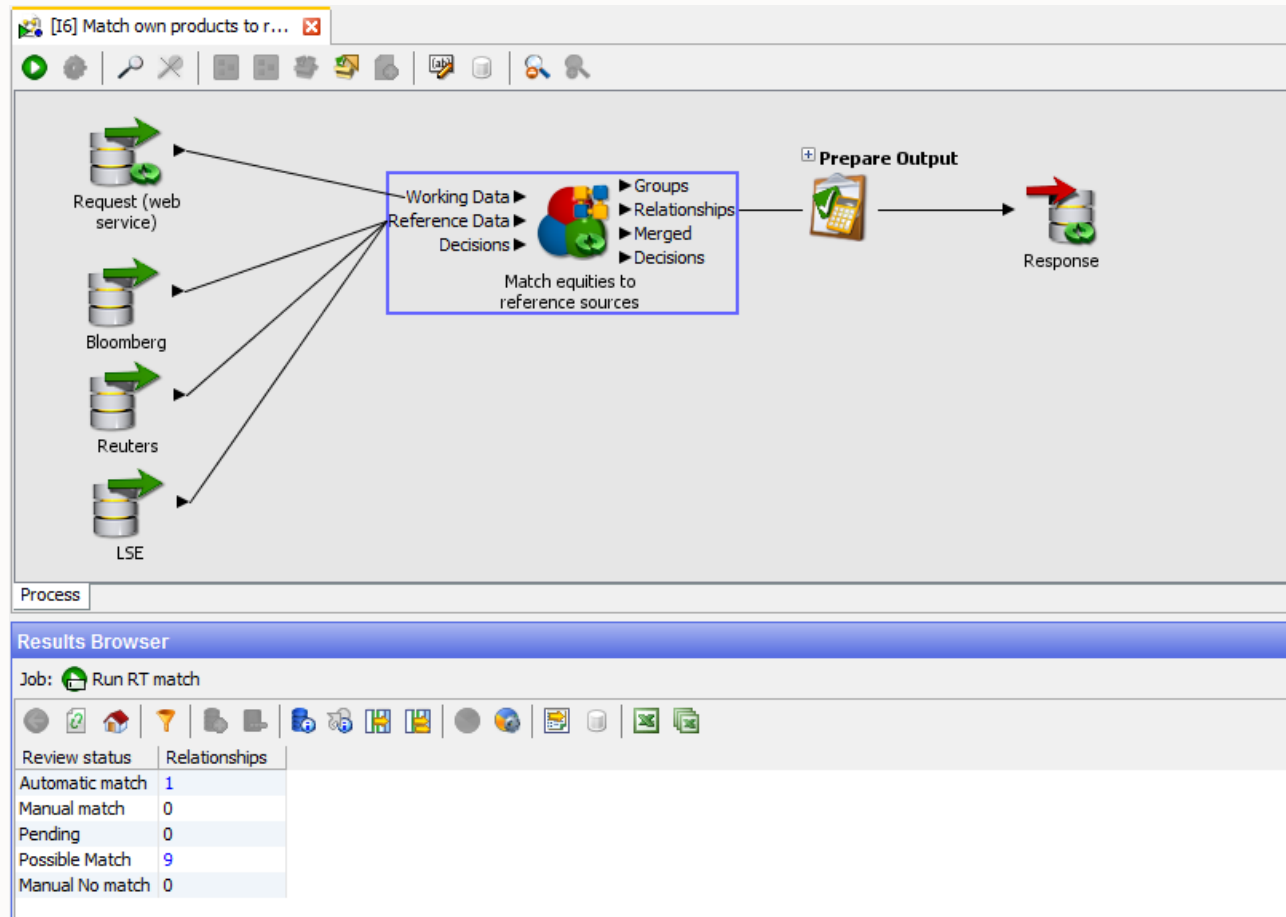
Web Service Generation

- 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

Edit Web Service

Web Service Inputs
What should this web service expect?

Multi Record

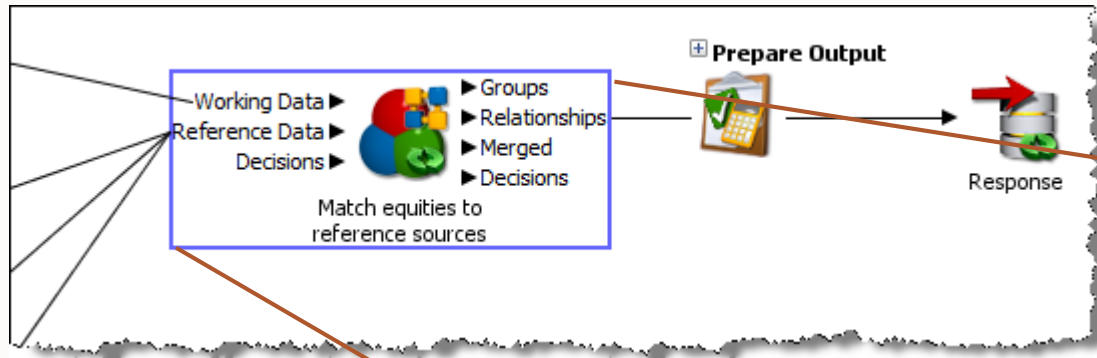
Attribute Name	Attribute Type
IssuerName	STRING
Currency	STRING
ISIN	STRING
SEDOL	STRING
ShortCode	STRING



```
<!--  
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS  
  -->  
<!-- wsdl file generated 04-Jul-2014 12:34 -->  
<wSDL:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns:  
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.datanor  
  <wSDL:types>  
    <xs:schema elementFormDefault="qualified" targetNamespace="http://www.datan  
      <xs:element name="request">  
        <xs:complexType>  
          <xs:sequence>  
            <xs:element minOccurs="0" name="IssuerName" type="xs:string"/>  
            <xs:element minOccurs="0" name="Currency" type="xs:string"/>  
            <xs:element minOccurs="0" name="ISIN" type="xs:string"/>  
            <xs:element minOccurs="0" name="SEDOL" type="xs:string"/>  
            <xs:element minOccurs="0" name="ShortCode" type="xs:string"/>  
          </xs:sequence>  
        </xs:complexType>  
      </xs:element>  
    </xs:schema>  
  </wSDL:types>  
</wSDL:definitions>
```

Web Service Logic

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



The screenshot shows the 'Match' configuration window with the following details:

- Match Rule Groups: All Rules
- Rule Name: Name, SEDOL
- Priority Score: 89
- Decision: REVIEW

Pri...	Name	Decision
<input checked="" type="checkbox"/>	100 Name, ISIN, Currency, SEDOL	MATCH
<input checked="" type="checkbox"/>	99 Name s/w, ISIN, Currency, SEDOL	MATCH
<input checked="" type="checkbox"/>	98 Name prob, ISIN, Currency, SEDOL	MATCH
<input checked="" type="checkbox"/>	97 Name, ISIN, Currency	MATCH
<input checked="" type="checkbox"/>	95 Name, ISIN, SEDOL	MATCH
<input checked="" type="checkbox"/>	94 Name s/w, ISIN, SEDOL	MATCH
<input checked="" type="checkbox"/>	93 Name prob, ISIN, SEDOL	MATCH
<input checked="" type="checkbox"/>	91 Name s/w, ISIN, Currency	MATCH
<input checked="" type="checkbox"/>	90 ISIN, Currency, SEDOL	MATCH
<input checked="" type="checkbox"/>	89 Name, ISIN	MATCH
<input checked="" type="checkbox"/>	89 Name, SEDOL	REVIEW
<input checked="" type="checkbox"/>	88 Name s/w, ISIN	MATCH
<input checked="" type="checkbox"/>	88 Name s/w, SEDOL	MATCH
<input checked="" type="checkbox"/>	86 ISIN, SEDOL	MATCH

Comparison Configuration:

Comparison	Value
ISIN ED	*
NAME ALL KEY WORDS	80-100 (80)
NAME WORD MATCH	*
NAME S/W	*
NAME FIRST TOKEN FUZZY	*
CURRENCY ED	*
SEDOL ED	Exact (0)

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.

