

Function Specification Consistency Checker

Ericsson Dynamic Activation

FUNCTION SPECIFICATION

Copyright

© Ericsson AB 2012–2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

Ericsson™ is the trademark or registered trademark of Telefonaktiebolaget LM Ericsson.

All other products or service names mentioned in this document are trademarks of their respective companies.



Contents

1	Introduction	1
1.1	Purpose and Scope	1
1.2	Target Group	1
1.3	Typographic Conventions	1
2	Functional Overview	2
2.1	Consistency Checker Interfaces	2
2.2	Data Analysis Process	3
2.2.1	Data Collection	4
2.2.2	Analysis	7
2.2.3	Post-processing	9
3	Analysis Functions	9
3.1	Pattern Based Analysis	9
3.2	Rule Based Analysis	12
3.2.1	Offline Rule Based Analysis	14
3.2.2	Real Time Rule Based Analysis	16
3.2.3	Two Steps Analysis	17
4	Deployment	18
5	Logical Architecture	19
6	Integration	20
7	Operation and Maintenance	21
8	Security	21
9	Vocabulary	22
	Reference List	25





1 Introduction

This section is an introduction to this document. It contains information about the purpose, scope, and target group for the document. This section also contains explanations of typographic conventions used in this document.

1.1 Purpose and Scope

The purpose of this document is to provide readers a high level understanding of the Consistency Checker application.

This document describes the application functionality as well as information regarding integration, deployment and operation and maintenance of the application.

1.2 Target Group

The target groups for this document are as follows:

- Application Owner
- Application User
- Integrator
- Operation and maintenance personnel

For more information about different target groups, see *Library Overview*, Reference [1].

1.3 Typographic Conventions

Typographic conventions are described in the document *Library Overview*, Reference [1].

2 Functional Overview

This document describes the application functionality as well as information regarding integration, deployment and operation and maintenance of the application.

The Consistency Checker can be used to perform the following use cases:

- **Redundant data source inconsistency analysis** – Audit whether entries in the two data sources are identical, for example, to compare a pair of Home Location Registers (HLRs) in redundant configuration.
- **User data inconsistency analysis** – Audit whether a user is provided with consistent services across data sources, for example, to check whether a user in the Business Support System (BSS) also exists in the HLR and the services are consistent for that user in both data sources.
- **Data updates over time** – Find changed data in a given data source between two different points in time, for example, find out how many new users are introduced between *day1* and *day2*.

2.1 Consistency Checker Interfaces

Figure 1 shows the Consistency Checker interfaces.

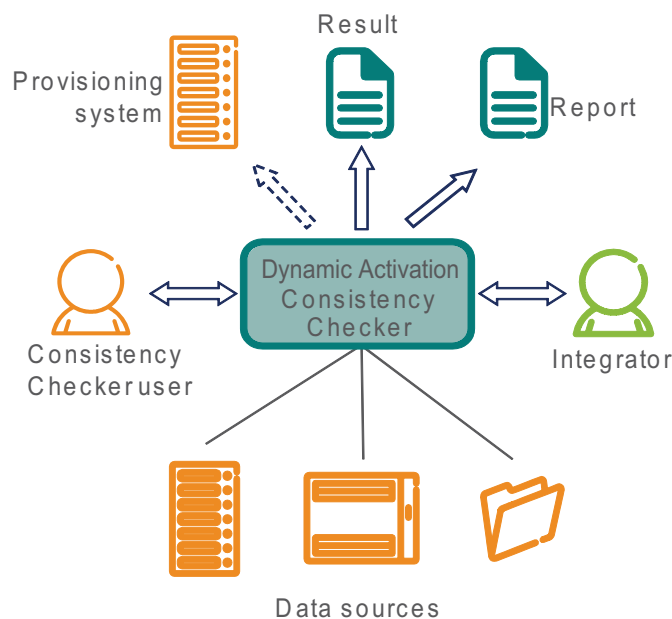


Figure 1 Consistency Checker Interfaces



The Consistency Checker analyzes data from various data sources throughout the operator's network. A data source could be a high performance telecom application/enabler or a legacy enterprise IT solution. A data source can include tens of millions of entries each containing complex information.

Data sources are classified in two categories:

- Online data sources

The instances that are in operation.

- Offline data sources

Data extracted from an online data source, for example a database backup, file output of data, and so on.

The Consistency Checker user defines what, how and when to perform data analysis for consistency and is at the same time the primary receiver of the data analysis outcome, that is, report and result files.

The Consistency Checker user performs tasks via Graphical User Interface (GUI). For more information about the Consistency Checker GUI, see *User Guide for Consistency Checker*, Reference [2].

The Integrator performs integration tasks that require programming competence, such as integrate data sources and develop advance consistency analysis functions. For information about how to produce integration tasks, see *Programmers Guide for Consistency Checker*, Reference [3].

The Provisioning System is an optional actor of the Consistency Checker. It could provide mediation function when fetching data from online data sources or correct inconsistencies based on the analysis result. Behind the scene, the system administrator makes sure that the system is configured and in condition to perform the tasks.

2.2 Data Analysis Process

As shown in Figure 2, a data analysis process can be divided into three phases:

- **Data Collection**

For more information, see Section 2.2.1 on page 4.

- **Analysis**

For more information, see Section 2.2.2 on page 7.

- **Post Processing**

For more information, see Section 2.2.3 on page 9.

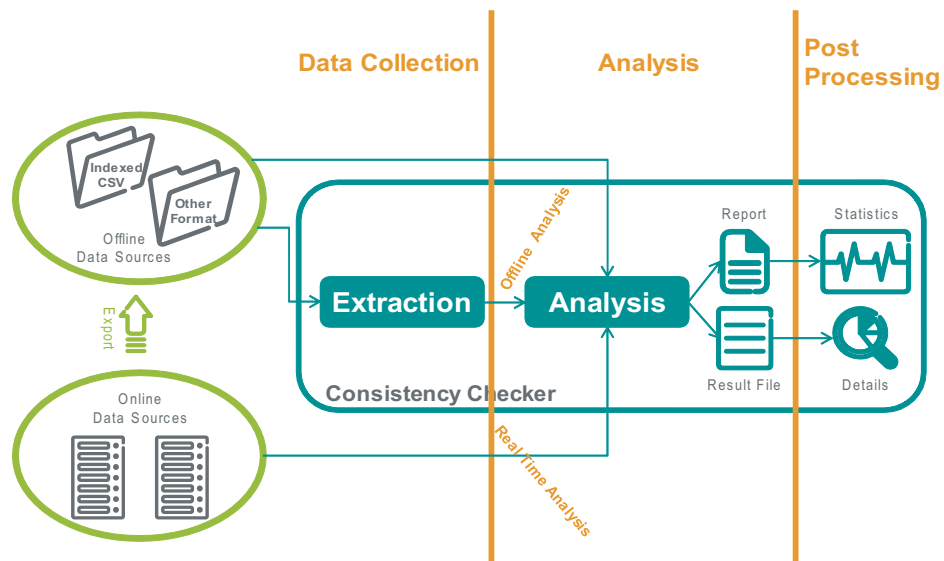


Figure 2 Overview of the Analysis Process

Analysis tasks are grouped into two categories based on the data source type:

- Online data source analysis (real time analysis)
- Offline data source analysis (offline analysis)

The Consistency Checker comes with two main features

- The Offline Consistency Checker feature provides the functions for offline analysis.
- The Real Time Consistency Checker feature provides analysis functions based on online data sources.

2.2.1 Data Collection

The goal for data collection is to prepare comparable data for analysis.

Offline analysis is performed on dump files stored in the dump store, see Figure 3. Data collection takes place prior to Analysis. Extraction and analysis can be handled as two separate processes. This makes it possible to use the same extracted data for multiple analyses.

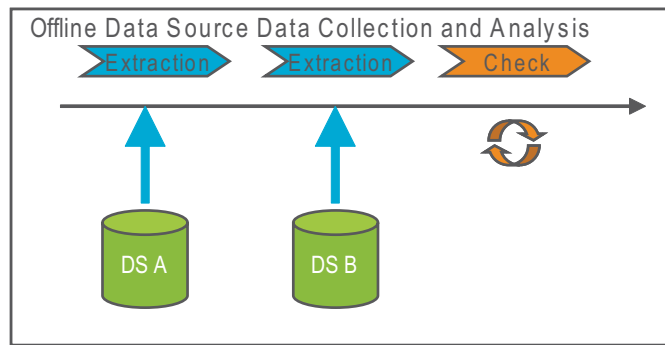


Figure 3 Offline Data Source Data Collection and Analysis

Offline analysis is based on Comma Separate Values (CSV) format files and sorted based on an identifier; such files are called dump files.

In order to prepare the dump files, the data should be exported from the online data sources. If the data sources do not provide means to perform the export, this must be solved during the Consistency Checker integration phase.

If the files exported from the online data sources are in CSV format and indexed based on an identifier, they can be used directly for analysis.

If the files exported from the online data sources are of other formats, for example, binary or LDIF, the extraction function in the Consistency Checker is used to perform data conversion and indexing. For more information about the extraction function, see Section 2.2.1.1 on page 6.

The logic to handle the conversion and indexing is called Extraction Handler. Extraction Handlers need to be developed and integrated in the Consistency Checker prior to deployment of the application.

It is necessary to consider data segmenting in the collection phase, as there is a possibility that one data source will contain all subscribers in the network, while the other data source will contain only a sub-set of the subscribers. The analysis result of these two data sources will naturally include unintended findings. In order to ensure credible analysis result, the dump files for an analysis shall include compatible segment of entries.

Online analysis is based on data retrieved from the online data sources in real time, thus, data collection takes place in conjunction with Analysis, see Figure 4. The Consistency Checker fetches one pair of entries at a time and compares the data immediately. This process is repeated until all the entry pairs included in the task are checked.

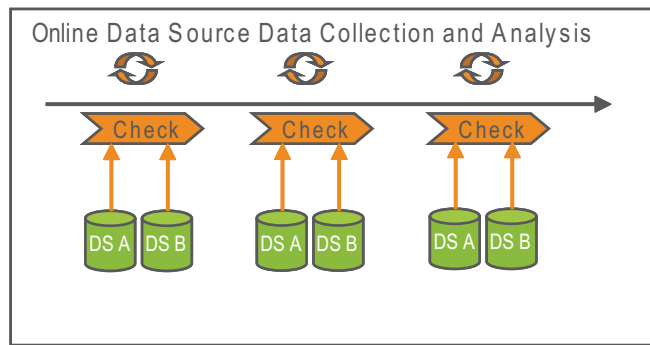


Figure 4 Online Data Source Data Collection and Analysis

For online data sources, the analysis function fetches data directly from the data sources. The logic to handle data collection is called Resource Adapter. Resource Adapters are produced in the integration phase. In order to reduce the Resource Adapter complexity, it might be necessary to fetch data via a mediation system.

2.2.1.1

Extraction

The extraction process is shown in Figure 5.

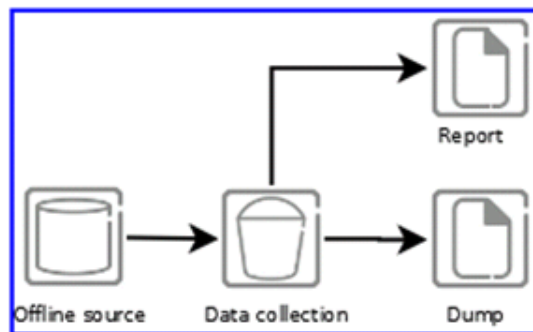


Figure 5 Extraction

Data extraction is a pre-processing step of an analysis order that extracts data from one or more data sources of the same type and generates one dump file that aggregates the entries from all the data sources, as shown in Figure 6.

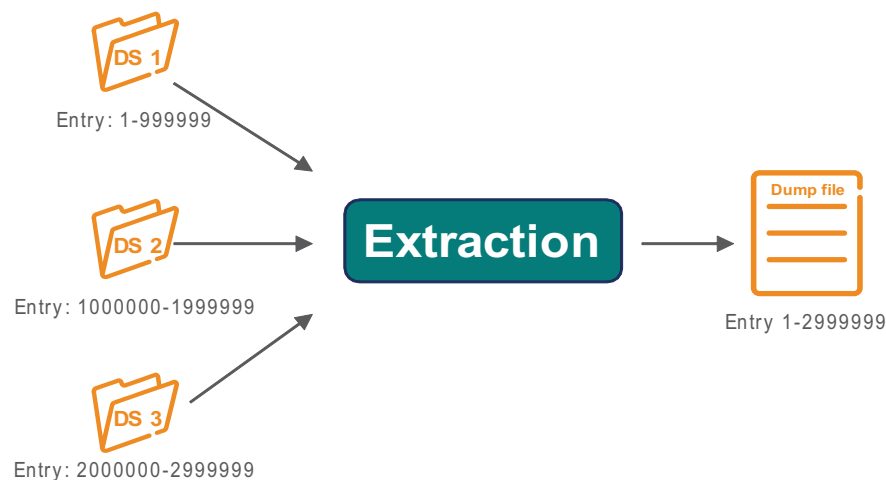


Figure 6 Offline Data Source Aggregation

When performing extraction from a data source, the extracted data is stored in a temporary file. Upon successful completion, this file is then moved to the dump store. If an error occurs during the extraction, the extraction process is terminated and the status of the order is set to failed. This procedure ensures that a dump file reflects the complete content of the data source(s). As such dump files will be used for consistency check, the quality of input data is secured.

Another mechanism to secure dump file quality is time stamp control. For recurrent analysis order, time stamp for each extraction is recorded. Before executing a new extraction, the last run time stamp is used to compare with the time stamp of the data source file. If the time stamp of the data source file is older or equal than the last run, the extraction is canceled. This procedure prevents extraction of outdated data source.

When deleting an analysis order, the order itself is removed from the Consistency Checker. However, the extracted dump files are kept in the dump store, which makes it possible to use and reuse the files for one or more data analysis orders.

The dump files can be deleted manually in the Consistency Checker file system.

Data sources with time stamp are listed in the report, so that the user can validate whether the extraction performed correctly or not.

For more information, see *System Administrators Guide for Consistency Checker*, Reference [4].

2.2.2

Analysis

The main task of Analysis is to identify inconsistency between entry pairs in the two data sources. The outcome is a report and a result file.

Consistency Checker comes with two methods for data analysis, that is, pattern based analysis and rule based analysis. Pattern based analysis applies for offline data sources for this release while rule based analysis is applicable for both offline data sources and online data sources.

The pattern based analysis method automatically identifies data comparison pattern between the two data sources and performs analysis based on the pattern, see Figure 7.

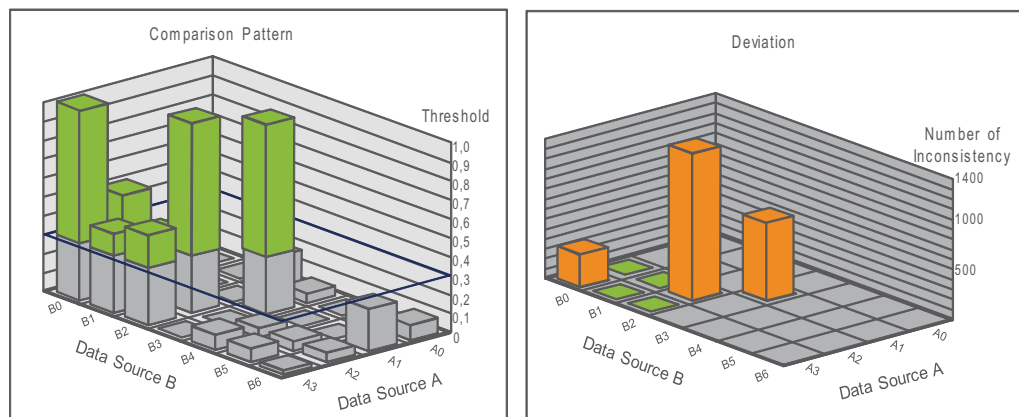


Figure 7 Pattern Based Analysis

In the pattern detection phase, the Consistency Checker goes through the entries pair by pair to identify correlation between any two given attributes in the entries. Correlation is defined by a number of implemented algorithms. When the correlation value is higher than the predefined threshold after reading all the data values, the two attributes are considered comparable. All comparable attributes together form the comparison pattern. In the example in B0-A0, B0-A1, B0-A3, B1-A0, ..., B6-A3 are compared for all the entries pairs. The threshold is defined at 0,3. The comparison pattern is then B0-A3, B1-A3, B2-A3, B1-B2, B2-A2, B3-A1.

In the pattern based analysis phase every entry pair is audited against the detected pattern. Deviation from the pattern is considered inconsistent and recorded in the report and result file. In the example in Figure 7, there are inconsistencies between B0-A3, B2-A2 and B3-A1.

This method also provides self-adaptive function, that is, a recurrent pattern based order can adopt the pattern changes over time. The longer a pattern has been stabilized, the longer it takes to change the pattern. Thus, it can require many analysis before a changed pattern takes effect.

This method is available for analysis of offline data sources in this release.

The rule based analysis method is based on a specification where the user specifies the rules for comparison. In order to make the analysis possible, data model for each data sources types are defined and embedded into the Consistency Checker during the integration phase. The data models are then used by users to define analysis specification.



The rule based analysis method is the original analysis method of the Consistency Checker and is available for analysis of both offline data sources and online data sources. Specification is configured via the GUI. For more information, see *User Guide for Consistency Checker*, Reference [2].

2.2.3 Post-processing

Post-processing provides statistics for recurrent analysis tasks to present the inconsistent status over a period of time. It also provides drill down function to present the actual inconsistent data and value in the entry pairs for rule based analysis.

3 Analysis Functions

This section describes the functional details with consideration to data source type and the desired analysis method.

3.1 Pattern Based Analysis

Pattern based analysis is only applicable for offline data sources in this release.

Pattern based analysis can be used to perform redundant data source analysis, user data inconsistency analysis as well as data update analysis.

Figure 8 illustrates the general work flow for pattern based analysis.

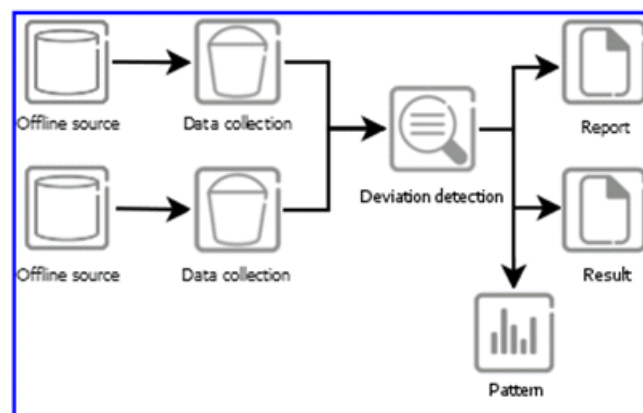


Figure 8 Pattern Based Analysis Work Flow



The prerequisites for performing pattern based analysis are dumps files in CSV format indexed by comparable identifier and listed in same order. Such dump files can either be exported from the online data sources directly or produced with the help of the extraction function. For detail about the extraction function, see Section 2.2.1.1 on page 6.

A pattern based analysis task is defined as an analysis order. An analysis order contains the data sources to be analyzed and the execution schedule. An analysis order can be a single event to be executed immediately or in a later stage it can also be a recurrent activity starting from a given day and time. An example is shown in Figure 9.

Add Analysis Order

Analysis Details

Name: myAnalysis ✓ OK

Description: Find inconsistencies between data source A and B.

Schedule: Once ▼ 10/06/2016 16:36:00 ✕ 📅

Data Sources

Data Source A	Data Source B
Type: <input checked="" type="radio"/> Dump file <input type="radio"/> Extraction <input type="radio"/> Online	Type: <input checked="" type="radio"/> Dump file <input type="radio"/> Extraction <input type="radio"/> Online
Source: dsA.csv ...	Source: dsB.csv ...

Analysis Type

Type: Pattern based analysis ▼

Your analysis will find inconsistencies based on identified data relation pattern

Save **Cancel**

Figure 9 Pattern Based Analysis Order

The pattern based analysis function identifies the comparison pattern when executing a one time order or the first instance of a recurrent order. The



detected pattern is shown in the **Definition** tab when expanding the pattern based analysis order in the launch page, see Figure 10. This pattern is then used for analysis.

Pattern			
Attribute in Data Source A	Attribute in Data Source B	Distinctness Level	Identifier
Surname	Mail	99.99%	false
FamilyName	Mail	99.99%	false
PhoneNumber	PublicID	99.99%	false
PrivateID	PrivateID	100%	true
PrivateID	SIPURI	100%	false
Enabled	PrivateID	99.99%	false
Enabled	SIPURI	99.99%	false

Figure 10 Data Relation Pattern

Additional pattern algorithms can be introduced as product customization. For more information, please contact your local Ericsson representative.

The result of an order is presented under the **Report** tab, as shown in Figure 11.

PatternBased

Report

Definition

Statistics

Reports

	Start time	Status
▼	10/06/2016 12:42	SUCCESS

Report 10/06/2016 12:42

Details

Unique posts in A

3

Unique posts in B

4

Mismatching po...

24

Analyzed posts

99998

Posts in A

100001

Posts in B

100002

Input source A

/var/cc/cc_home/var/dump/dsA.csv

Input source B

/var/cc/cc_home/var/dump/dsB.csv

Execution time

11076ms

Result files:

PatternBased_2016-10-06_10.42.25_Result.zip

PatternBased_2016-10-06_10.42.25_Report.xlsx

Deviations per Rule

Attribute in Data Source A	Attribute in Data Source B	Deviations
Surname	Mail	4
FamilyName	Mail	8
PhoneNumber	PublicID	6
PrivateID	PrivateID	0
PrivateID	SIPURI	0
Enabled	PrivateID	6
Enabled	SIPURI	6

Figure 11 Report of An Pattern Based Analysis Order

Result files are stored in the Consistency Checker file system, and can be downloaded through the links in Figure 11.

3.2 Rule Based Analysis

Rule based analysis is based on a specification defined by the user. A specification is configured based on two pre-integrated data models. Data models are developed and integrated with the Consistency Checker by integrator. For detail of how to integrate new data models see Section 6 on page 20.

User defines one or more comparison rules in a specification by selecting attribute pairs from the two data models and assigns each with a comparison method. For more information on how to configure specification, see *User Guide for Consistency Checker*, Reference [2].

An example of a specification is shown in Figure 12.

Model A (dsA)	Model B (dsB)	Rule	Identifier	
PrivateID	PrivateID	Equal	true	

Figure 12 Specification

Common used rule specifications can be saved as templates. User can define any number of templates. One template can be used by any number of analysis orders. Existing templates can be deleted from the GUI. The template is then removed from the Consistency Checker. However, the removal will not affect any analysis order that uses this template.

The available comparison rules are:

- Equal



To check if two attributes are identical (case sensitive).

- `Equal ignore case`

To check if two attributes are identical (case insensitive).

- `Not Equal`

To check if two attributes are not identical (case sensitive).

- `Equal Ending`

To check if two attributes have identical endings. For example, the `MSIDSN` in data source A is stored in format with country code and regional code as 0046315234562, and in data source B in format 5234562.

- `A contains B`

To check if the attribute value in data source A contains the attribute value in data source B.

- `B contains A`

To check if the attribute value in data source B contains the attribute value in data source A.

- `Conditional Mapping`

To compare two attributes of different information type. For example, `STATE=CONNECTED` is compatible to `Subscriber_status=0`.

- `Match`

To check if two attributes match a certain pattern. The syntax is: `<matcher A>=<matcher B>`.

The following example shows that to fulfill the rule, the attribute value from data source A must contain at least one character, and the attribute value from data source B starts with ABC.

```
EXIST=MATCH(^ABC.*$)
```

Additional comparison rules can be developed and integrated into the Consistency Checker. For more information, see *Programmers Guide for Consistency Checker*, Reference [3].

Conditional rules are used to determine if a **Rule** shall be run or not by applying a pre-condition of the rule. The following conditional rules are available:

- **Exists** – `EXISTS`, to check if an attribute exists.
- **Not exists** – `!EXISTS`, to check if an attribute does not exist
- **Match** – `/<Regular expression>/`, to check if an attribute matches a regular expression.

It is possible to use the logical operators **AND** and **OR** in the conditional rules for comparison rules.

The conditional rules can be entered in the conditions field as shown in Figure 13.

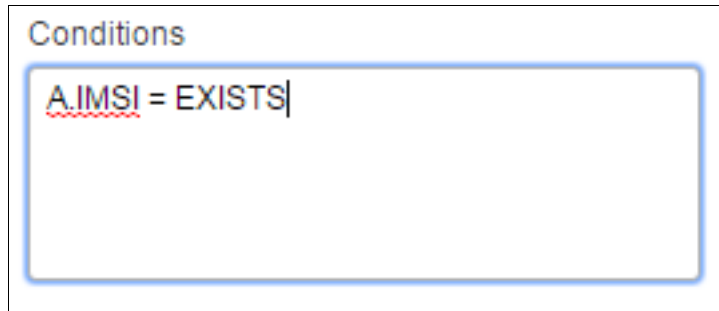


Figure 13 Rule Condition

For more information on comparison rules and rule conditions, refer to *User Guide for Consistency Checker*, Reference [2].

3.2.1 Offline Rule Based Analysis

Offline rule based analysis can be used to perform user data inconsistency analysis as well as data update analysis.

Figure 14 illustrates the general work flow for offline rule based analysis.

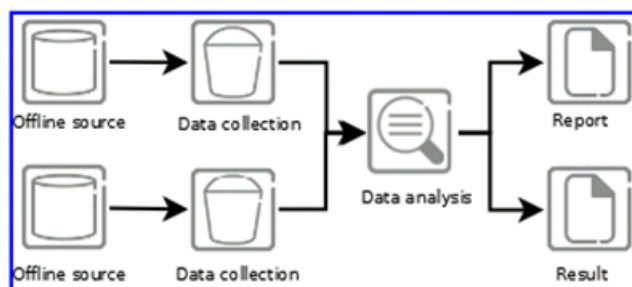


Figure 14 Offline Rule Based Analysis

The prerequisite is the same as for rule based analysis. Dump files in CSV format must be available and sorted in the right order based on compatible identity. Such dump files can either be exported from the online data sources directly or produced with the help of the extraction function. For more information about the extraction function, see Section 2.2.1.1 on page 6.

An offline rule based analysis task is defined as an analysis order. This order contains the data sources to be analyzed, the specification and the execution schedule. An order can be a single event to be executed immediately or in a later stage. It can also be a recurrent activity starting from a given day and time. An example is shown in Figure 15.



Definition

Details		Data Source A		Data Source B	
Name	DailyDemo	Type	Dump source	Type	Dump source
Type	Rule based analysis order	Data source	dsA.csv	Data source	dsB.csv
Occurrence	Daily 07:58:17				

Rule specification

	Attribute in Data Source A	Attribute in Data Source B	Rule	Identifier
>	PrivateID	PrivateID	Equal	true
▼	PhoneNumber	PublicID	Equal ending	false

Rule details

Rule	Equal ending			
Parameter	9			
>	PrivateID	SIPURI	B contains A	false

Figure 15 Example of Offline Rule Based Analysis

For recurrent analysis order, time stamp for each analysis is recorded. Before executing a new analysis, the last run time stamp is used to compare with the time stamp of the dump files. If the time stamp of any of the dump files holds a time stamp older than that of the last run, the analysis is canceled. This procedure prevents usage of outdated dump files.

A report is provided per analysis instance base, that is, one report for immediate analysis order and several reports for a recurrent analysis order, see Figure 16.

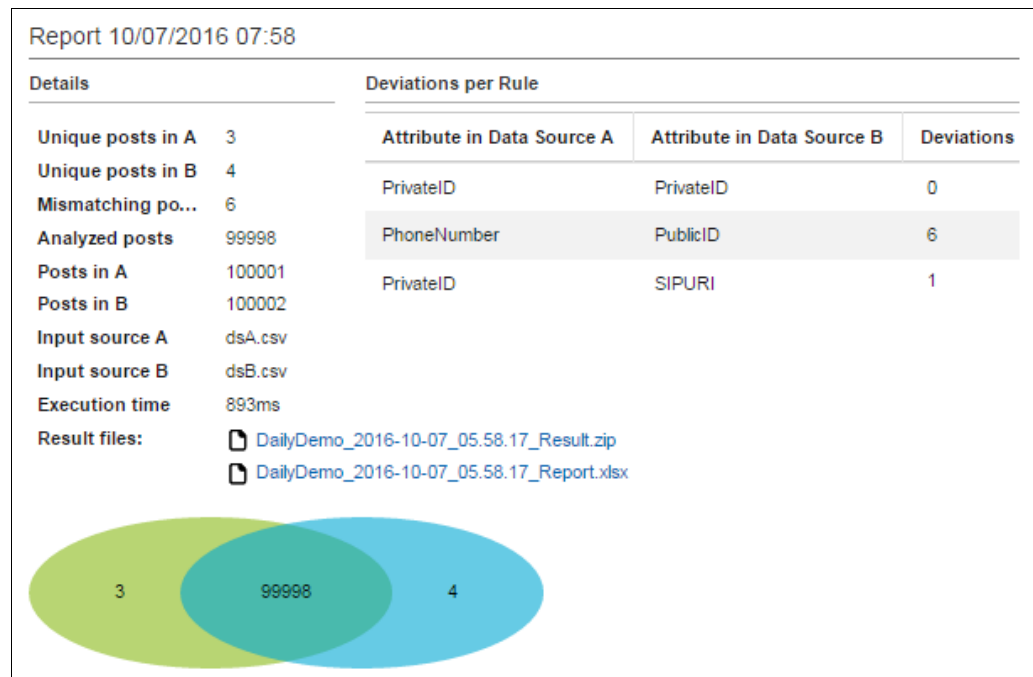


Figure 16 Report of an Offline Rule Based Analysis

Statistics are provided for recurrent orders. The result files are stored in the Consistency Checker file system.

3.2.2 Real Time Rule Based Analysis

The Real Time Consistency Checker analyses data is retrieved directly from online data sources in real time.

This function can be used to perform all three use cases, that is, redundant data source inconsistency analysis, user provisioning and service provisioning inconsistency analysis.

The Real Time Consistency Checker provides one end-to-end process that covers the following:

- Real time data retrieval from online data sources per entry pair base.
- Comparison of the retrieved data according to the specification per entry pair base.
- Recording in result file for inconsistent entry per pair base.

Report is generated when all entry pairs defined in the order are compared.

During the analysis process, the Real Time Consistency Checker does not generate dump files as the Offline Consistency Checker. Consistent entries are discarded while inconsistent entries are recorded in the result file.



Two analysis types are provided specifically for the Real Time Consistency Checker. These two analysis types are activated only when the deployed system is entitled to the Real Time Consistency Checker feature. They are as follows:

- **Real time analysis**, see Figure 17.

User specifies the identity of the entries to be compared. The identities are expressed in the form of ranges. The available identity type is number range. The number ranges are used to retrieve entries from the two data sources. Thus, it is crucial that the number ranges specified are the identity of entries in both data models. Other identity types can be integrated in the back-end and will appear in the drop down.

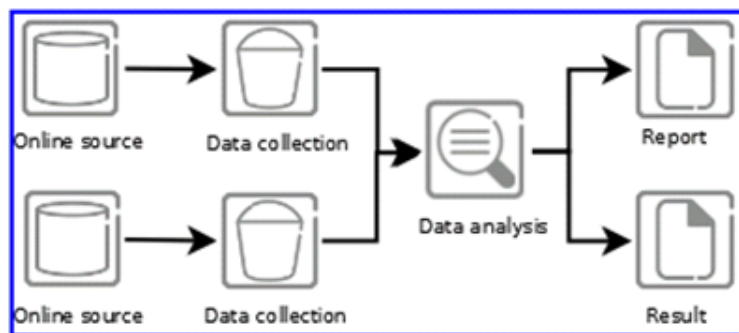


Figure 17 Real Time Analysis

- **Real time analysis based on a previous analysis**, see Figure 18.

User specifies which existing analysis order or analysis instance report is the base for this new order.

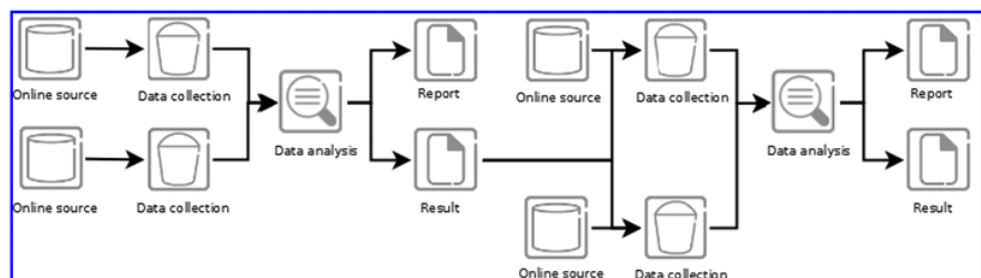


Figure 18 Real Time Analysis Based on a Previous Analysis

3.2.3

Two Steps Analysis

Two steps analysis refers to the practice that combines the Offline Consistency Checker and the Real Time Consistency Checker functionality is shown in Figure 19. This practice is only available for rule based analysis in the current release.

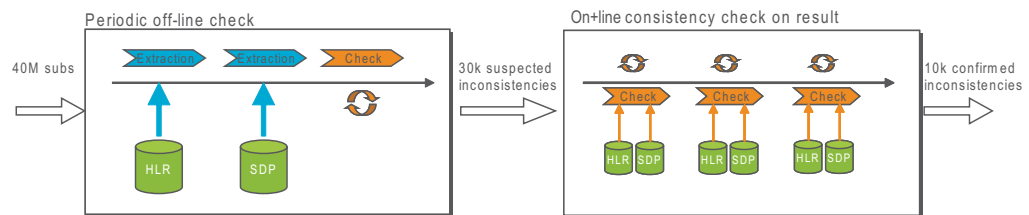


Figure 19 Two Steps Analysis

The first step is to perform an offline data analysis on the entire subscriber base. This analysis pinpoints that minor amount of subscribers have inconsistent data in the data dumps. One reason for the found inconsistencies could be that the data dumps were generated at different points in time. Another reason could be that a data dump takes a lot of time to generate itself, as well as there is a big difference in time between the oldest and the newest item in the same dump file.

The second step of the data analysis confirms the real inconsistencies. The second data analysis is done only on the suspected inconsistencies and is based on real time data analysis. This step removes the dependency of the dump files timing. The result shows that only a factor of the suspected subscribers actually was inconsistent in the network. With the two steps approach the impact on the network is minimal. For an example user case, refer to *User Guide for Consistency Checker*, Reference [2].

It is also possible to perform ad-hoc real time analysis. If the report of an analysis instance shows unusual high inconsistency rate between two data sources, a one-time analysis order based on the report can be issued to reconfirm the inconsistency status.

4 Deployment

The Consistency Checker is a software application that can be deployed on any JEE 7 compliant application server. There is no specific requirement on hardware or operating system. It can be deployed on a personal computer or on a server. Operating system and application server selections are up to each deployment, thus, not supplied with the Consistency Checker. Hardware selection is depending on capacity and throughput demand.

Porting Guide for Consistency Checker, Reference [5], provides general information on how to deploy and maintain the Consistency Checker.

The Consistency Checker is verified on Glassfish Open Source Edition. For installation details, see *Installation Instruction for Consistency Checker on Glassfish Server Open Source Edition*, Reference [6].



Application characteristics and hardware selection information are described in the Characteristics Description. For details, please contact your local Ericsson representatives.

5 Logical Architecture

The logical architecture of the Consistency Checker is shown in Figure 20.

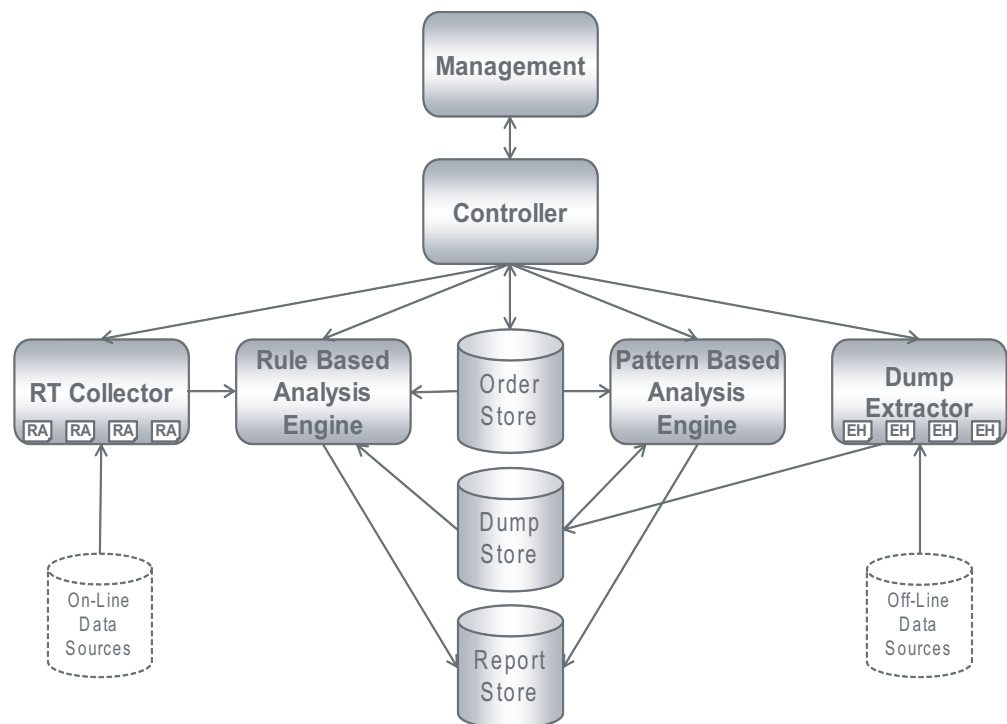


Figure 20 Logical Architecture

Management: The Management unit is the system interaction front end. This is realized as the GUI.

Controller: The Controller is the controlling unit that manages the persisted artifacts in the storage units. It is also responsible for order scheduling.

Rule Based Analysis Engine: The Rule Based Analysis Engine evaluates data according to the rules defined in the analysis specification.

Pattern Based Analysis Engine: The Pattern Based Analysis Engine identifies patterns of the data sources and detects inconsistent data based on the pattern.



Extraction Handler (EH): An Extraction Handler is the logic that interprets offline data sources of a specific type. Extraction Handlers are not part of the standard Consistency Checker and have to be developed by integrator.

Dump Extractor: The Dump Extractor provides utility services for the hosted Extraction Handlers.

Resource Adaptor (RA): A Resource Adaptor is the logic that interworks with online data sources of a specific type. Resource Adaptors are not part of the standard Consistency Checker. Any Resource Adaptor that follows the JEE connector architecture (JCA) standard can be used by the Consistency Checker.

Real Time (RT) Collector: The Real Time Collector provides utility services for the hosted Resource Adaptors.

Order Store: The Order Store is where the orders are persisted.

Dump Store: The Dump Store is where the dump files are persisted.

Report Store: The Report Store is where the reports and result files are persisted.

6 Integration

The Consistency Checker is delivered as a tool box. In order for it to work in a certain operator's environment, customization of the Consistency Checker must be performed prior to deployment on the target environment. Detail instructions for customization are described in the *Programmers Guide for Consistency Checker*, Reference [3].

The typical customization tasks are:

- Development of Extraction Handler

Development of Extraction Handler is applicable for offline data sources.

- Development of Resource Adaptor

Development of Resource Adaptor is applicable for online data sources where JCA compliant adaptor is not available.

- Development of new comparison rules

Development of new comparison rules is only applicable for the rule based analysis method. The Consistency Checker comes with a number



of pre-defined comparison rules as described in Section 3.2 on page 11. If these rules are not adequate for a specific tasks, new rules can be developed by the integration.

- Additional post processing function

Customized post processing function can be added outside of the Consistency Checker. Consistency Checker provides a notification API that can be used to send notification to external program. Some examples of post processing are advance reporting and correction of inconsistent data.

- Customized Reporting

The report and result files are in `xml` format and can be used as input to produce customer specific reports.

Ericsson maintains an inventory of integrations performed by Ericsson organizations for re-use purpose. This ensures the cost effectiveness of any future customization work.

Product customization service is also available for customer requirements that cannot be satisfied with the above customization. For more information, please contact the local Ericsson representative.

7 Operation and Maintenance

The operation and maintenance methods and procedures are depending on the environment on which the Consistency Checker is deployed.

System Administrators Guide for Consistency Checker, Reference [4], is provided for Glassfish Open Source Edition. Similar document for the target environment shall be produced when porting the Consistency Checker.

8 Security

System Administrators Guide for Consistency Checker, Reference [4], is provided for Glassfish Open Source Edition. Similar document for the target environment shall be produced when porting the Consistency Checker.



The default role for CC is `ccuser`. It is possible to add new roles.

A user can be mapped to one or more roles. Each user has a unit user name and password.

9 Vocabulary

Analysis: Analysis in the Consistency Checker context, refers to the process that compares data according to specified rules and generates/stores report and result file in the report store.

Analysis Instance: An analysis instance is an executed analysis order.

Analysis Order: An analysis order is a description of an analysis task. The Consistency Checker user defines, in an analysis order, which data sources are to be compared by which specification and when.

Conditional Rule: Pre-condition of a rule. A comparison rule will only be evaluated if all of its Conditional rules have passed.

Data Model: A data model represents a collection of metadata and can reflect the complete metadata or a subset of the metadata.

Data Source: Data source refers to the entity that contains the original data to be analyzed.

Pattern Based Analysis: The analysis method that identifies automatically comparable data pattern between the two data sources and performs data comparison based on the pattern.

Extraction: Data Extraction refers to the pre-process that collect data for analysis. It converts data from offline data sources into the Consistency Checker internal format and stores the result as a dump file in the dump store.

Identifier rule: A rule that defines if a record from a Data Source A is comparable with a record from a Data Source B.

Online Data Source: Data sources that are in operation.

Offline Data Source: Data extracted from an online data source, that is a database backup, file output of data, and so on.

Report: A report is a summary of an extraction instance or an analysis instance.



Result File: A result file is the record of inconsistency details of an analysis instance.

Specification: An analysis specification (sometimes referred to as specification) is a description of how to compare two data sources. The Consistency Checker user defines, in a specification, which data or data pairs to compare and their respective comparison rule.

Rule Based Analysis: The analysis method that performs analysis based on manually defined comparison rules, that is analysis specification.





Reference List

Ericsson Documents

- [1] *Library Overview*, 18/1553-CSH 109 628 Uen
- [2] *User Guide for Consistency Checker* , 24/1553-CSH 109 628 Uen
- [3] *Programmers Guide for Consistency Checker*, 25/1553-CSH 109 628 Uen
- [4] *System Administrators Guide for Consistency Checker*, 5/1543-CSH 109 628 Uen
- [5] *Porting Guide for Consistency Checker*, 2/006 92-CSH 109 628 Uen
- [6] *Installation Instruction for Consistency Checker on Glassfish Server Open Source Edition*, 5/1531-CSH 109 628 Uen