QROWD - Because Big Data Integration is Humanly Possible

Innovation Action

Grant agreement no.: 732194

D6.2 – Integrated processing of data-in-motion and data-at-rest

Due Date	30 Nov 2018
Actual Delivery Date	30 Nov 2018
Document Author/s	Semih Yumuşak (AI4BD) Daniel Hladky (AI4BD)
Version	0.4
Dissemination level	PU
Status	Final
Document approved by	Luis-Daniel Ibáñez

Project co-funded under Horizon 2020 Research and Innovation EU programme

TABLE OF CONTENT

History

Version	Date	Reason	Revised by
0.0	16.10.2018	Table of Contents	Semih Yumuşak (AI4BD)
0.1	26.10.2018	First draft	Semih Yumuşak (AI4BD)
0.2	02.11.2018	Ready for Internal Review	Semih Yumuşak (AI4BD)
0.3	16.11.2018	Internal review	Daniel Hladky (AI4BD)
0.4	26.11.2018	Finalized after review	Luis-Daniel Ibáñez (SOTON) Semih Yumuşak (AI4BD)

EXECUTIVE SUMMARY

The D6.2 document addresses the integrated processing solution for data-in-motion and data-at-rest, which is designed for data science users to process both streaming and static data sources together in a coordinated manner. The integrated processing capabilities demonstrated in this document are designed to meet the requirements for real-time analytics, stream processing, high performance computing, and static data source integrations. The demonstration provides information about the streaming data management, static data management, and integrated processing capabilities of the initial QROWD platform.

The document draws on experiences made in D4.4, D4.2 as well on the initial architecture presented in D8.1 and serves as a foundation for WP5, WP6 and WP7. The overall aim is to support the use cases from WP1 (Advanced road information) and WP2 (Intelligent urban transportation).

The document explains on a technical level both methods and illustrates the use cases with real examples.

1 INTRODUCTION

Real-time analytics for streaming data sources can be provided by many tools, whereas static data analysis has a broader range of tools. However, integrated processing of streaming and static data sources is a requirement for the analytics solutions which use both data sources at the same time. In QROWD, in order integrate big data streams with the human-in-the-loop processes and historical data sources, we need to have an innovative solution to be able to provide real-time analytics while enriching historical databases to extract useful information. The main goal of this deliverable is to define our innovative architectural solution to have an integrated processing for data-at-rest and data-in-motion, at the same time. The proposed D6.2 is inline with the D8.1 architecture and fully compliant with the BDE and FIWARE frameworks.

In this demonstration document, the integrated processing capabilities of the initial QROWD platform are described. This task consists of two processing functionalities (1- Data-in-motion/Streaming data, 2- Data-at-rest/Static Data) and an integration engine to process them all together. In order to implement the integration capabilities, Enterprise Integration Patterns [1] enhanced with various data source integrations were used, on top of the Apache Nifi data flow software. In Section 2, the data processing capabilities of the architectural solution are explained and a prototype implementation is demonstrated.

2 TECHNICAL DESCRIPTION

The integrated process engine acts as an intermediary streaming platform between different data sources. Each streaming process can be pipelined with a static data source, which brings the engine the capability of processing data of both types in the same flowh.

The working principle of the integrated process engine is illustrated in Figure 1.

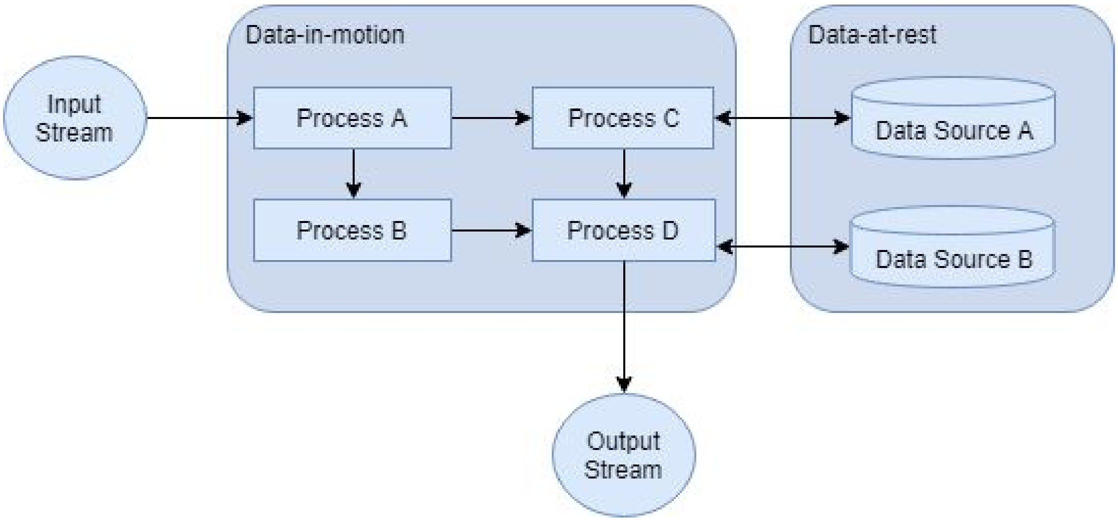


Figure 1: Architectural view of integrated processing of data-in-motion and data-at-rest

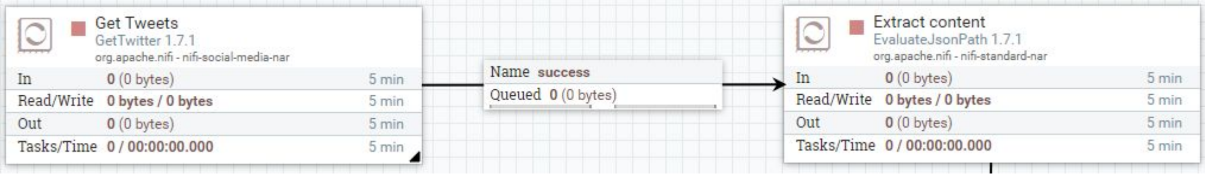
In the following sections, the streaming and static data management capabilities, and the integration concept is explained more in detail with real illustrations. Finally, an Natural Language Processing (NLP) case-study for integrated processing is presented. The NLP use case has a close correlation to the D4.4 harvesting multilingual data.

2.1 Streaming Data Management (Data-in-motion):

Streaming data management consists of 3 main components: Get, Process, and Put. The streaming flow can be initialized from following sources:

- Web API: (Web Socket, HTTP GET, TCP, UDP, Google Cloud Pub/Sub, Kinesis, RSS, Twitter)
- Message Queues (Kafka, AMQP, MQTT, SQS, Fiware Orion)
- E-mail Read/Send (POP3, IMAP, SMTP)

In Figure 2, integrating an external streaming source to process is demonstrated by using GetTwitter processor, a data stream that collects tweets.



Get Tweets
GetTwitter 1.7.1
org.apache.nifi - nifi-social-media-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

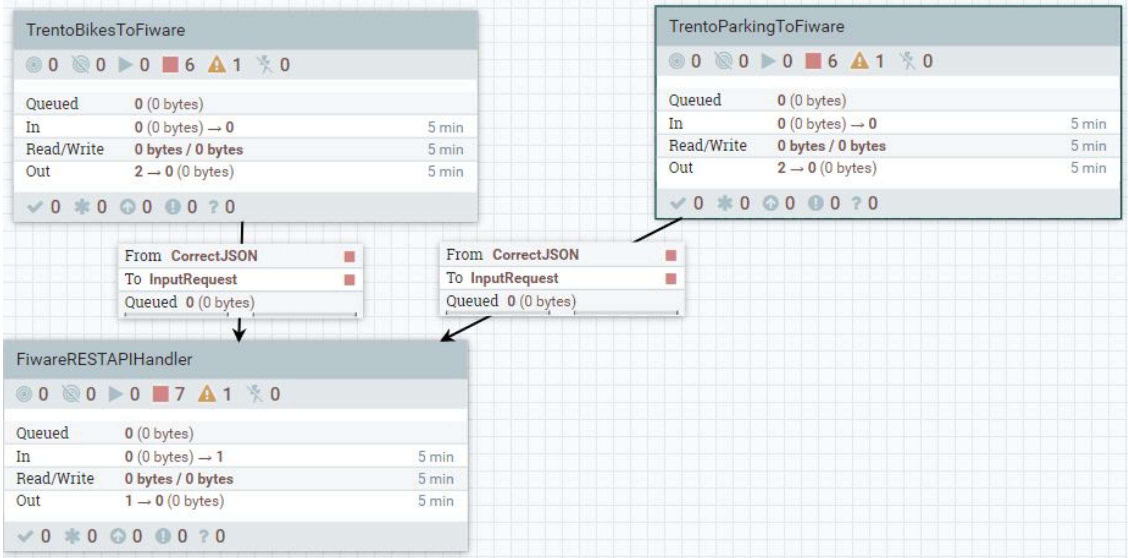
Name success
Queued 0 (0 bytes)

Extract content
EvaluateJsonPath 1.7.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Figure 2: Get data from a stream

In Figure 3, feeding the Fiware Orion Context broker from a static data flow is demonstrated, a feature we used in the Urban mobility dashboard described in D2.3



TrentoBikesToFiware

Queued	0 (0 bytes)	
In	0 (0 bytes) → 0	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	2 → 0 (0 bytes)	5 min

TrentoParkingToFiware

Queued	0 (0 bytes)	
In	0 (0 bytes) → 0	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	2 → 0 (0 bytes)	5 min

From CorrectJSON
To InputRequest
Queued 0 (0 bytes)

From CorrectJSON
To InputRequest
Queued 0 (0 bytes)

FiwareRESTAPIHandler

Queued	0 (0 bytes)	
In	0 (0 bytes) → 1	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	1 → 0 (0 bytes)	5 min

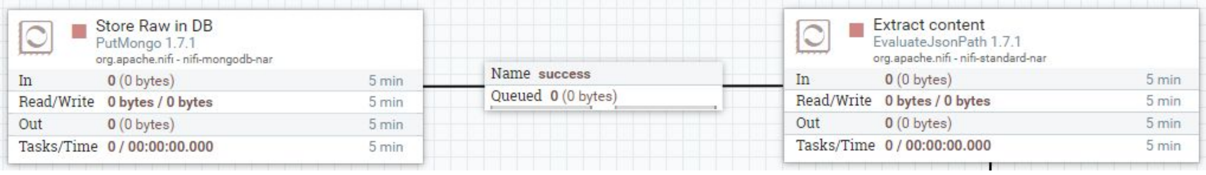
Figure 3: Put data to an external streaming data source (Get static bike and parking information from a static data source and push it to Fiware Orion Context Broker)

2.2 Static Data Management (Data-at-rest)

Static data management consists of 3 main components: Get, Process, and Put. The static flow can be initialized from following sources:

- Static File Reading (CSV, Text, JSON, HDFS, FTP, SFTP, Azure Blob)
- Database (Mongo, RethingDB, InfluxDB, HBase, DynamoDB, Cassandra

As shown in Figure 4, a static data source can be used as a source or as a destination.



Store Raw in DB
PutMongo 1.7.1
org.apache.nifi - nifi-mongodb-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name success
Queued 0 (0 bytes)

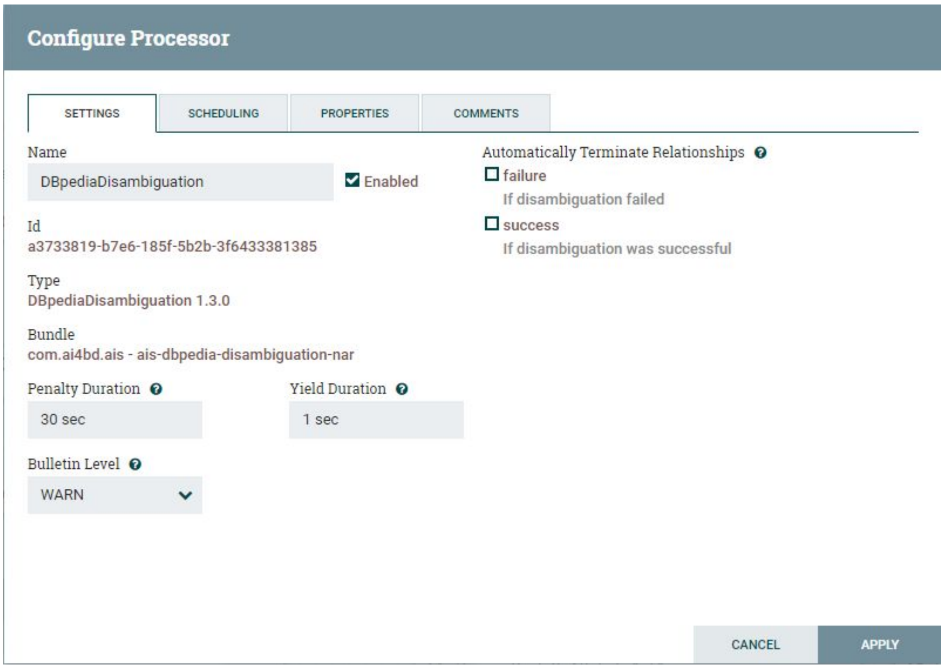
Extract content
EvaluateJsonPath 1.7.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Figure 4: Static data source integration

2.3 Integrated Processing Environment Descriptions

Each data source and processor comes with a configuration screen as shown in Figure 5. In this configuration screen, the settings, scheduling, and properties of a processor can be adjusted. Whereas every processor has built in properties, each processor can have its unique properties to be used inside of the processing functionality.



Configure Processor

Name
DBpediaDisambiguation ☑ Enabled

Id
a3733819-b7e6-185f-5b2b-3f6433381385

Type
DBpediaDisambiguation 1.3.0

Bundle
com.ai4bd.ais - ais-dbpedia-disambiguation-nar

Penalty Duration
30 sec

Yield Duration
1 sec

Bulletin Level
WARN

Automatically Terminate Relationships
□ failure
If disambiguation failed
□ success
If disambiguation was successful

Figure 5: Configuring a processor

2.4 Case-Study: Integrated NLP Processing of Data-in-Motion and Data-at-rest

As a case-study, AI4BD NLP tools were integrated into the initial platform for Data-in-Motion processing. Figure 6 shows the main text processing flow for an English text, which takes a text input and process it as a stream. This processing flow performs the following respectively after getting a text input:

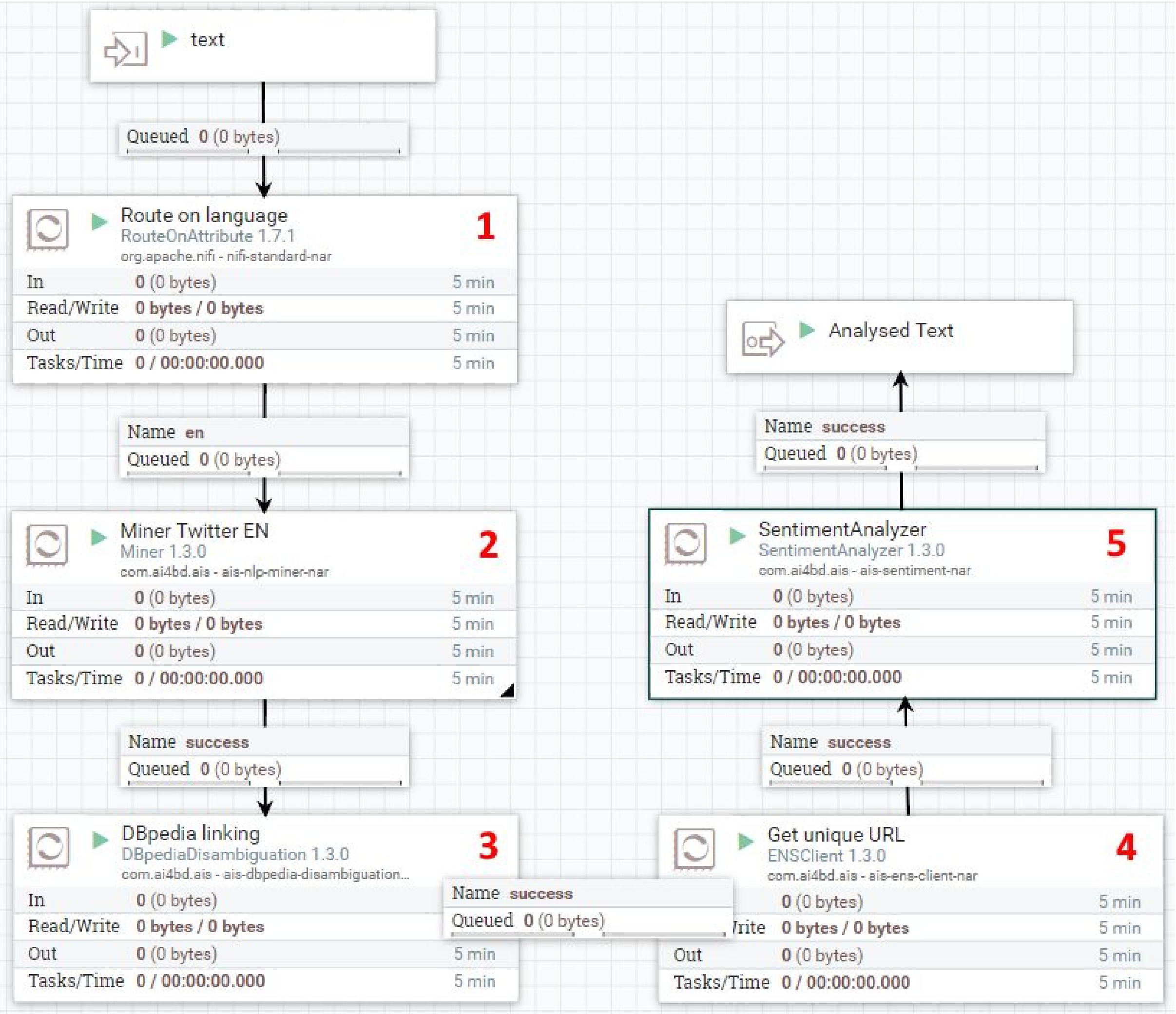
1- Perform language inference and route to English flow if English language detected. (Enabling a multilingual support for routing different languages to different processing tasks)

2- Apply Named Entity Recognition by using AI4BD Miner application

3- Connect to DBpedia Static Knowledge base for entity linking

4- If no entity can be linked from DBPedia, use AI4BD ENS service to create a unique URI

5- Perform a sentiment analysis on the text



text

Queued 0 (0 bytes)

Route on language
RouteOnAttribute 1.7.1
org.apache.nifi - nifi-standard-nar
1

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name en
Queued 0 (0 bytes)

Miner Twitter EN
Miner 1.3.0
com.ai4bd.ais - ais-nlp-miner-nar
2

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name success
Queued 0 (0 bytes)

DBpedia linking
DBpediaDisambiguation 1.3.0
com.ai4bd.ais - ais-dbpedia-disambiguation...
3

In	0 (0 bytes)	
Read/Write	0 bytes / 0 bytes	
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name success
Queued 0 (0 bytes)

Get unique URL
ENSClient 1.3.0
com.ai4bd.ais - ais-ens-client-nar
4

	0 (0 bytes)	5 min
/rite	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name success
Queued 0 (0 bytes)

SentimentAnalyzer
SentimentAnalyzer 1.3.0
com.ai4bd.ais - ais-sentiment-nar
5

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name success
Queued 0 (0 bytes)

Analysed Text

Figure 6: Integrated text processing flow

While processing the stream, the integrated processing engine performs a query to a Static (Data-at-rest) source (DBpedia Knowledge Base) to ask for an entity URI. Within this simplified NLP data flow, both Data-in-motion and Data-at-rest are processed processed in an integrated manner for a text processing task.

As explained more in the previous chapters, the initial platform comes with a processing functionality for many sources which are classified as data-in-motion and data-at-rest.

References

[1] Hohpe, G., & Woolf, B. (2004). *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional.