STREAMLINE. FAST REACTIVE ANALYTICS

D7.3 – Annual Report, Quality Assurance and Evaluation Period 1

Project Number	688191
Project Acronym	STREAMLINE
Nature	R: Report
Dissemination Level	Public
Work Package	WP7
Due Delivery Date	May 2017
Actual Delivery Date	December 2017
Lead Beneficiary	SICS
Authors	Björn Hovstadius (SICS)

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 688191.

Executive Summary

The STREAMLINE work plan is divided in three phases corresponding to each year of the project. The reporting for STREAMLINE is divided into two periods; M1-M18 and M19-M36 (end of project).

This document is to describe the work carried out during the first period, M1-M18, of the project. It details work in each Work Package and its results.

Following a brief introduction to this deliverable, Chapter 2 presents the impact on selected KPIs.

Chapter 3 details the work done to assure Deliverable quality.

Chapter 4 provide details for each Work Package work during the first period.

Chapter 5 explain the changes in the project and how it affects the work done in each work package.

Table of Contents

List of Figures

1 Introduction

The project's vision is to make data analytics of streaming data and batch data more accessible for a broader base of user than data science and data analytics experts. This is important, as analyzing large amounts of data is becoming a competitive must for European companies, large or small. Access to tools that enable this is key to building the necessary competence supply needed to stay competitive. The project is designed to both invest in the necessary research and also include European use cases. The use cases are there to both to validate results but also inject industrial requirements to the research work.

The project is built on the European project Apache Flink that is a stream processing framework that is gaining momentum and popularity. STREAMLINE will propose its results to Apache Flink to enhance the platform to support both data at rest and in motion, make it more robust and easier to use. STREAMLINE is working with the Apache Flink community and its processes to include the results in the core of Apache Flink. This process is called FLIP or Flink Improvement Process. All code generated is available to the community on GITHUB. The research partners are DKFI, SICS and SZTAKI.

The project started with four use cases from the European companies Rovio, Internet Memory Research, Portugal Telecom (now Altice) and NMusic, all described in detail in the Description of Work. They all have in common that they have massive amounts of streaming data that they need to analyze and also make decisions from, such as real-time recommendations. During the first year NMusic did some very good work together with SZTAKI on a recommender for their online music service. They also investigated the context awareness approach put forward by IMR. Unfortunately due to business reasons NMusic decided to shut down its business and had to withdraw from the project at M12.

The remaining beneficiaries were able to rearrange parts of the project as detailed below. STREAMLINE does not foresee any impact on the results and ability to reach its objectives other than the missing use case and that certain reports delivered late.

This document describes in more detail the results and work done during the first period.

2 Overview of results

All Work Packages performed according to plan and contributed to the impact of STREAMLINE during the first period. Each Work Package work and results are detailed in the subsequent chapter.

On a high level the following KPIs are relevant to measure the success of STREAMLINE

- Community growth
- Papers published

- Contributions to Apache Flink

2.1 Community growth

One of the key goals of STREAMLINE is to make Data Analytics of streaming and batch data more accessible. STREAMLINE has chosen to make its contributions through working with the Apache Flink platform and its community. By strengthening this European Data Analytics platform we create a strong foundation for Europe in this important segment. A way to measure impact is to look at the number of people involved in the Apache Flink Open Source development.

*Over the first period the number of contributor grew almost **300%** or form 80 to 313 contributors*[1].

2.2 Publications

The scientific work in the technical Work Packages is measured by publications and presentations at high-level conferences.

*The academic partners in STREAMLINE published **16** major papers during the first period.*

- Boden, C., Spina, A., Rabl, T., & Markl, V. (2017, May). Benchmarking Data Flow Systems for Scalable Machine Learning. In Proceedings of the 4th Algorithms and Systems on MapReduce and Beyond (p. 5). ACM.
- Rabl, T., & Jacobsen, H. A. (2017, May). Query Centric Partitioning and Allocation for Partially Replicated Database Systems. In Proceedings of the 2017 ACM International Conference on Management of Data (pp. 315-330). ACM.
- Grulich, P., Rabl, T., Markl, V., Sidló, C. I., & Benczúr, A. A. (2017). STREAMLINE-Streamlined Analysis of Data at Rest and Data in Motion. In EDBT/ICDT Workshops.
- Traub, J., Steenbergen, N., Grulich, P., Rabl, T., & Markl, V. (2017). I2: Interactive Real-Time Visualization for Streaming Data. In EDBT (pp. 526-529).
- Benczur, A. A., Palovics, R., Balassi, M., Markl, V., Rabl, T., Soto, J., ... & Haridi, S. (2016). Towards Streamlined Big Data Analytics. ERCIM NEWS, (107), 31-32.
- Rabl, T., Traub, J., Katsifodimos, A., & Markl, V. (2016). Apache Flink in current research. it-Information Technology, 58(4), 157-165.
- Jonas Traub, Sebastian Breß, Tilmann Rabl, Asterios Katsifodimos, and Volker Markl. Optimized On-Demand Data Streaming from Sensor Nodes. was accepted for publication at this year's ACM Symposium on Cloud Computing (SoCC) 2017 conference.
- Cao, P., Gowda, B., Lakshmi, S., Narasimhadevara, C., Nguyen, P., Poelman, J., ... & Rabl, T. (2016, September). From BigBench to TPCx-BB: Standardization of a Big Data Benchmark. In Technology Conference on Performance Evaluation and Benchmarking (pp. 24-44). Springer, Cham.
- Carbone, P., Traub, J., Katsifodimos, A., Haridi, S., & Markl, V. (2016, October). Cutty: Aggregate sharing for user-defined windows. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 1201-1210). ACM.

[1] https://github.com/apache/flink

- Kunft, A., Alexandrov, A., Katsifodimos, A., & Markl, V. (2016, June). Bridging the gap: towards optimization across linear and relational algebra. In Proceedings of the 3rd ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond (p. 1). ACM.
- Alexandrov, A., Salzmann, A., Krastev, G., Katsifodimos, A., & Markl, V. (2016, June). Emma in action: Declarative dataflows for scalable data analysis. In Proceedings of the 2016 International Conference on Management of Data (pp. 2073-2076). ACM
- Paris Carbone, Stephan Ewen, Gyula Fóra, Seif Haridi, Stefan Richter, Kostas Tzoumas:State Management in Apache Flink®: Consistent Stateful Distributed Stream Processing. PVLDB 10(12): 1718-1729 (2017)
- Pálovics, Kelen, Benczúr: RecSys 2017 Tutorial: Open Source Tools for Online Learning Recommenders. https://recsys.acm.org/recsys17/tutorials/#content-tab-1-4-tab
- Frigó, Pálovics, Kelen, Kocsis and Benczúr: RecSys 2017 poster: Alpenglow: Open Source Recommender Framework with Time-aware Learning and Evaluation. http://ceur-ws.org/Vol-1905/
- Frigó, Pálovics, Kelen, Kocsis and Benczúr: RecTemp 2017 - workshop on reasoning on temporal aspects in user modeling in conjunction with RecSys 2017: Online ranking prediction in non-stationary environments https://sites.google.com/edu.haifa.ac.il/tempreasoninginrs/program?authuser=0
- Zvara,Szabó, Hermann and Benczur: Workshop on Autonomic Systems for Big Data Analytics (ASBDA 2017) in conjunction with the 2017 IEEE International Conference on Cloud and Autonomic Computing (ICCAC). Tracing Distributed Data Stream Processing Systems

STREAMLINE was presented at the following conferences

- SIGMOD
- VLDB
- SOCC
- ICDE
- EDBT – best demo award for the l^2 paper[2]
- CIKM
- RECSYS

2.3 Contributions to Apache Flink

There are a number of ways to contribute to the Apache Flink platform. When making suggestions for more impactful additions or changes a process called FLIP[3], or Flink Improvement Process, is used.

There are currently 6 accepted FLIPs and STREAMLINE has contributed one. This is FLIP-16: Loop Fault Tolerance. This FLIP has major impact on the use of Apache Flink in enterprise applications.

[2] http://edbticdt2017.unive.it/?awards#EDBTbestdemo
[3] https://cwiki.apache.org/confluence/display/FLINK/Flink+Improvement+Proposals

In face the company Data Artisans that is responsible for a commercial release of Apache Flink advertise this as a major feature[4].

Nine proposed FLIPs are under discussion. STREAMLINE is a contributor to 2.

FLIP-15 Redesign Iterations (Scoping, Flow Control and Termination)
FLIP-17 Side Inputs for DataStream API pending

3 Deliverables and quality assurance first period

In order for timely delivery and document quality the following work procedure has been followed during the first period:

What	When	Responsible
First Draft	4 weeks before due date	Lead partner
Comments first Draft	3 weeks before due date	Reviewer
Second Draft	2 weeks before due date	Lead partner
Comments second Draft	1 week before due date	Reviewer
Final incorporating comments	2 days before due date	Lead partner
Quality check and review		Coordinator
Delivery	Due date	Coordinator

During M1-M18 the following deliverables were due.

Deliverable	Deliverable Title	WP	Lead	Reviewer
D7.1	Project Plan Period 1	WP7	SICS	NMusic
D6.1	Dissemination Roadmap & Project Website	WP6	NMusic	SICS
D7.2	Ethics Management Plan	WP7	PT	SICS
D5.1	Design and Implementation v1	WP5	PT	ROVIA
D1.1	Combined Data at Rest and Data in Motion Analysis Platform v1	WP1	DFKI	SICS, SZTAKI
D2.1	Flink Real Time Stream Mining Library v1	WP2	SZTAKI	DFKI, SICS
D3.1	Flink deployment software	WP3	SICS	DFKI, SZTAKI
D4.1	Target events types per industry and benchmark per KPI	WP4	IMR	PT
D5.2	Field trials and Evaluation v1	WP5	ROVIO	NMusic
D6.2	Status report on dissemination activities Period 1	WP6	PT	SICS
D7.3	Annual Report, Quality Assurance and	WP7	SICS	ALL

[4] www.data-artisans.com, “Stateful stream processing:Enabling real-time applications for the modern enterprise”

	Evaluation Period 1			
D7.6	Ethics audit 1	WP7	SICS	ALL

The work in each deliverable is described in detail in the following chapter.

4 Explanation of the work carried per WP

In the following, we explain the work carried out individually in each work package and highlight results and next steps as well as any deviations from the initial project plan.

4.1 Work Package 1

4.1.1 Contributors:

Beneficiary	**Effort in PMs**
DFKI (lead)	15
SICS	9.72
SZTAKI	4.40

Work Package 1 is designed to address the Objectives I and II of the project.

4.1.2 Deliverables:

The work is reported in D1.1.

Objective	Results	Next steps	Any deviation from plan
Fault Tolerance: Managing Stateful Streaming	Stateful streaming in unified batch and streaming environment (FLIP-16, FLINK-3257)	Merge changes proposed in FLIP-16 to Apache Flink	
Unified Batch-Stream-Processing System	-Side Input Flink Improvement Plan (FLIP-17) that eventually has lead the JIRA issue (FLINK-6131). -Dynamic Tables that	-Implement or provide support for implementation of FLINK-6131. -Continue further development of Dynamic Tables	Initially the focus was on introduction Side Inputs as a viable option for performing hybrid computation. However, use cases proposed by the

	allow conversion of streaming data into batch and vice versa. -Redesign of the stream iteration model of Flink (FLIP15)	concept -Propose progress tracking and window iteration model	industrial partners and Flink community showed an interest in integration of hybrid processing and SQL. As a result, the concept of Dynamic Tables were introduced.
Incremental Computation	Cutty Framework		
Streaming SQL Joins	We propose a solution, called Joins in Streaming SQL, to enable rich SQL query processing with stream processing, and integrate it into Flink. This work has not yet started.	- Survey on Join Techniques for Streams and Materialized Views - Application to Changelogs on a Stateful Stream Processor - Implementation and Integration in Streamline Flink Evaluation	

Detailed explanation of the work done for each objective is explained below.

4.1.2.1 T1.2 Fault Tolerance: Managing Stateful Streaming

At the core of Apache Flink lies a unique snapshotting algorithm, as the central mechanism for managing consistent state and (batch and stream) application re-configuration. The overall effort, led by members of the *Streamline* project, has evolved from a snapshotting mechanism used for fault tolerance into an ecosystem of stateful stream operations: from re-scaling to consistent queryable state and version control. The snapshotting mechanism has also been extended recently by the same *Streamline* project members to address stream iterations in cyclic application graphs (see FLIP-16 and associated pull-request [3]). Finally, a definite scientific paper of the overall effort work has been recently published at the Proceedings of the VLDB Endowment (PVLDB) [1] and presented at the 43rd International Conference on Very Large Databases (VLDB 17) [4], a prestigious yearly conference that holds the highest rank (A*) in database research.

Publications and Documents:

1. Paris Carbone, Stephan Ewen, Gyula Fóra, Seif Haridi, Stefan Richter, Kostas Tzoumas:State Management in Apache Flink: Consistent Stateful Distributed Stream Processing. Industrial Track. PVLDB 10(12): 1718-1729 (2017)
2. https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=66853132

3. https://cwiki.apache.org/confluence/display/FLINK/FLIP-16%3A+Loop+Fault+Tolerance
4. https://github.com/apache/flink/pull/1668
5. https://issues.apache.org/jira/browse/FLINK-6131
6. https://www.slideshare.net/ParisCarbone/state-management-in-apache-flink-consistent-stateful-distributed-stream-processing

4.1.2.2 T1.3 Unified Batch-Streaming System

Our approach to unify the batch and data stream processing models is two-fold:
(I) We propose binary or n-ary graph operations that practically enable the interplay of data-at-rest and data-in-motion in the same application, namely *side inputs*.
(II) We are enriching the DataStream execution model with the functionality of bulk and stale-synchronous iterative processing, the most central computational paradigm of the DataSet model that allows multiple passes of the data.

These two approaches combined can yield a complete unification of DataSet and DataStream types in a single application graph which can term the separation of the two models obsolete. We further analyze the two approaches below:

Side-inputs: To enable join between static and streaming data, the concept of *Side Inputs* [3] was introduced. Side inputs are static datasets (typically stored on HDFS) that are treated as a DataStream by Flink. Upon the creation of a side input object, Flink loads the side input into the state of the operator. Once loaded, streaming operators have access to the side input. Side input allows us to perform the following tasks:

- Join stream with static data
- Join stream with slowly evolving data
- Evolving or static Filter/Enriching
- Window-Based Joins

The Flink Improvement Proposal (FLIP-17) is a collaboration between Flink PMC members and STREAMLINE contributors and explains the implementation details of side inputs. A complementary feature to side-inputs that is exposed through Flink's relational model is the concept of *dynamic tables* [4] that has been publicly proposed and discussed. This API abstraction allows for a conceptual mapping of streaming data into static data in the means of an evolving view of a statically-viewed table.

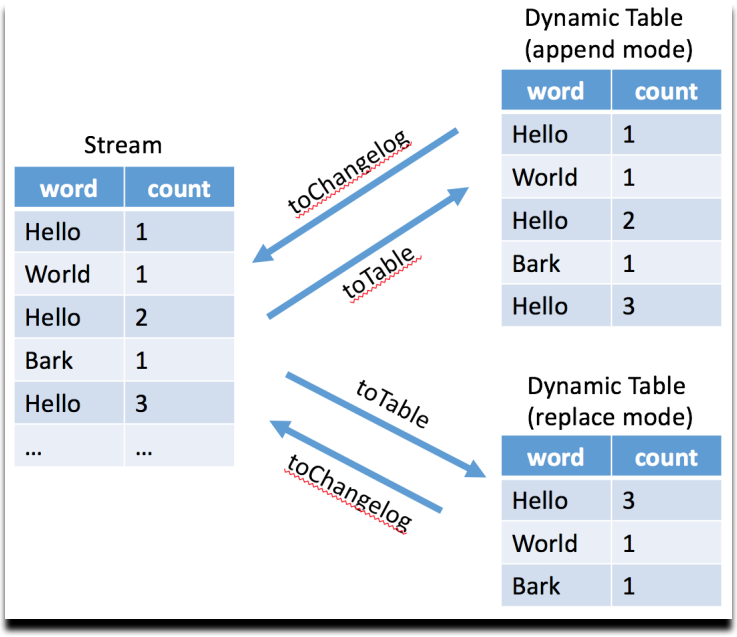

Figure 1 - Dynamic Table

Stream Iterations: A vast spectrum of DataSet applications rely on bulk synchronous iterative capabilities (e.g., graph processing and ML). Important algorithms such as Page Rank, Connected Component, Stochastic Gradient Descent etc. rely on bulk synchronization primitives and a static data representation. Currently, bulk iterative synchronization is not supported natively by the DataStream programming model and its execution primitives, which so far rely on asynchronous event-based feedback channels. We strive to enable iterative processing since we foresee it as an eminent need for a complete fusion of the DataSet model on the streaming system. So far, we have publically proposed and implemented the first steps towards that direction, namely *scopes*, *termination* estimation and *iterative flow control* (summarized in FLIP15 [5]) which are planned to be merged to an upcoming Flink Release once prioritized dependencies are accepted first. Currently, we are implementing and evaluating a full prototype of the solution (including runtime and API changes) which enables arbitrarily nested bulk synchronous iterations on stream windows through decentralized progress tracking. In essence, this will allow stream windows to encapsulate iterative DataSet capabilities in the long run and thus, materialize the vision of a unified batch and streaming framework. We plan to publish and propose our research results and source code implementation in an upcoming FLIP which will build on FLIP15.

Publications and Documents:

1. Grulich, P., Rabl, T., Markl, V., Sidló, C. I., & Benczúr, A. A. STREAMLINE-Streamlined Analysis of Data at Rest and Data in Motion. In EDBT/ICDT Workshops 2017
2. Benczur, A. A., Palovics, R., Balassi, M., Markl, V., Rabl, T., Soto, J., ... & Haridi, S. (2016). Towards Streamlined Big Data Analytics. ERCIM NEWS, (107), 31-32
3. https://cwiki.apache.org/confluence/display/FLINK/FLIP-17+Side+Inputs+for+DataStream+API
4. https://docs.google.com/document/d/1qVVt_16kdaZQ8RTfA_f4konQPW4tnl8THw6rzGUdaqU
5. https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=66853132

4.1.2.3 T1.4 Incremental Computing

Stream *windows* are, in essence, a user-defined operation that enables unbounded data streams to map into bounded data sets for grouped aggregations. To aggregate the data in user-defined windows every single data item in a window has to be processed. However, in typical streaming scenarios, windows are sliding and thus, the same set of items may appear in multiple windows, leading to overlapping computation. In some cases, redundant processing can be potentially avoided, depending on the nature of the defined sliding windows. Cutty [1] is a state-of-the-art framework that minimizes redundant computation on sliding windows through the use of incrementalization. Cutty includes a discretizer and aggregator component that automatically detects parts of computation that can be shared across multiple windows. Moreover, Cutty works with dynamic windows (dynamically changing the range and slide). Our paper that was published and presented [2] at the 25th ACM International Conference on Information and Knowledge Management (CIKM) showcase important results that build on Flink's DataStream programming model for efficient window aggregation.

Publications:

1. Carbone, P., Traub, J., Katsifodimos, A., Haridi, S., & Markl, V. (2016, October). Cutty: Aggregate sharing for user-defined windows. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 1201-1210). ACM.
2. https://www.slideshare.net/ParisCarbone/aggregate-sharing-for-userdefine-data-stream-windows

4.1.2.4 T1.5 Processing Joins in Streamline through View Maintenance with Changelogs

In this task, we propose a solution, called Joins in Streaming SQL, to enable rich SQL query processing with stream processing, and integrate it into Flink. SQL is the most popular language for data processing, and is widely used in industry. Therefore, proposing new ways to combine the results from SQL queries processing with those of stream processing will benefit a large class of users. The work on this task is ongoing. Below is our plan for integrating streaming SQL join into the Streamline API:

- Survey of existing techniques to combine relational tables with streams (e.g., Continuous Query Language), as well as streams with materialized views. Study their definitions of semantics for combining streams and relations.
- Definition of the semantics of SQL join queries on streams, especially with the concept of time.
- Integration of this API into Streamline Flink.
- Evaluation of the proposed API with relevant benchmarks (e.g., TPCH).
- Comparison with existing state-of-the-art (e.g., Continuous Query Language)

4.2 Work Package 2

4.2.1 Contributors:

Beneficiary	Effort in PMs
DFKI	3.0

ALB	1.28
SZTAKI (lead)	22.00
NMusic	1.27
IMR	1.80

Work Package 2 is designed to address Objective II of the project.

In this work package the combined data at rest and data in motion-processing platform was defined, designed and implemented as extension to the current version of Apache Flink. In this WP we built upon the next generation stream processing engine Apache Flink to provide a machine learning library serving the direct needs of the application partner business cases. We implemented recent approaches for approximate data structures for data streams to aid longer term storage, delayed messages in sliding windows and stream joins. We developed decentralized collaborative filtering methods to adapt the underlying distributed topology to the data locality present in event streams, based on our Parameter Server implementation.

The main achievements are summarized as the following pull requests and internal software libraries.

Table of pull requests:

ref	**feature**	**working repo (s)**	**JIRA**	**Pull Request**									
[1]	iALS recommender	https://github.com/gaborhermann/flink/tree/ials	https://issues.apache.org/jira/browse/FLINK-4613	https://github.com/apache/flink/pull/2576 [2]	Distributed SGD recommender		https://issues.apache.org/jira/browse/FLINK-4961	[3]	Ranking prediction, R	https://github.com/proto-n/flink/tree/ranking-rec-eval	https://issues.apache.org/jira/browse/FLINK-4712 https://issues.apache.org/jira/	[4]	Train batch, predict streaming	https://github.com/apache/flin

									ecommendationmetrics		browse/FLINK-4713			om/streamline-eu/ML-Pipelines/tree/streamml

Table of internal software libraries:

[5]	Parameter Server	https://github.com/gaborhermann/flink-parameter-server Design description: https://github.com/streamline-eu/WP2-MidtermReport/blob/master/Parameter%2BServer%2Bdesign%2B2.doc
[6]	Zeppelin notebooks for system comparison including batch and online combination	https://github.com/rpalovics/recsys-2017-online-learning-tutorial/
[7]	XGBoost connector	Forked from original source, now local code https://github.com/streamline-eu/xgboost-jvm-packages Use case examples https://github.com/streamline-eu/xgboost-application
[8]	Sketches in windows	https://github.com/streamline-eu/BloomFilter
[4]	Passive Aggressive classifier	https://github.com/streamline-eu/PassiveAggressiveClassifier

4.2.2 Deliverables:

D 2.1	**Flink Real Time Stream Mining Library V1** Version V1 the Flink Real Time Stream Mining Library with evaluation measurements over use case partner data. Basic classification, regression and recommendation methods for combined batch and stream machine learning based on linear models and stochastic gradient descent, also involving low memory synopses for sublinear storage of long-term updateable model components. Baseline measures defined for WP2.	M12
D 2.2	**Flink Real Time Stream Mining Library V2** Version V2 the Flink Real Time Stream Mining Library with evaluation measurements over use case partner data. Advanced methods, depending on use cases, potentially including gradient boosted trees, kernel methods, implicit and explicit ALS and tensor factorization, differential privacy and peer-to-peer recommenders.	M24

Objective	Results	Next steps	Any deviation

			from plan
To provide a data analysis framework for data in motion and data at rest ready to be used by data scientists. (Task 2.1)	[4] Train batch, predict streaming library [1] Distributed stochastic gradient recommender, with streaming API, conjoint batch and online algorithm. [6] Large scale evaluation and testing of conjoint batch and online recommenders. [3] Utility library.	Conduct large scale comparison with recommender tasks, batch, online, conjoint, both in Flink and Apache Spark. Organize a data challenge (ECML/PKDD or like) based on Portugal Telecom set-top box recommendation.	We completely abandoned the Flink batch API due to lack of support in the community. We implement batch algorithms with the streaming API as well.
To define and build different window operation semantics. (Task 2.2)	[8] sketches implemented in sliding windows	Conduct a use case analysis for social media trend real time visualization, in collaboration with WP4.	
To develop a machine learning library for mining streaming data. (Task 2.3)	[5] Parameter server for conjoint batch and online ML implementation, in streaming API. Parameter Server based methods including recommenders [3] and the Passive Aggressive classifier [9]. [7] XGBoost connector.	Implement more algorithms with Parameter Server. Provide Parameter Server pull request. Improve the parameter server with integrating looping API developments in WP1.	Passive Aggressive algorithm as a new requirement added.

4.2.3 Detailed description of D2.1 status regarding the Work Plan and current D2.2 status.

Version V1 the Flink Real Time Stream Mining Library with evaluation measurements over use case partner data.

- We evaluated recommender systems for NMusic in D2.1.
- We showed ongoing work for set-top box recommendation evaluation in D2.1.
- We evaluated text classification methods and developed the Passive Aggressive classifier for Internet Memory, to be reported in D2.2.

Basic classification, regression and recommendation methods for combined batch and stream machine learning based on linear models and stochastic gradient descent.

- Pull requests for iALS, DSGD recommenders [1,2].

- Utility library [3] pull request.
- Train batch, predict streaming [4].

Low memory synopses for sublinear storage of long-term updateable model components.

- Postponed to D2.2 due to increased needs for recommender systems and no direct use case need for sketches.

Baseline measures defined for WP2.

- Recommendation quality reported in D2.1

D2.2 Flink Real Time Stream Mining Library V2

Version V2 the Flink Real Time Stream Mining Library with evaluation measurements over use case partner data.

- Parameter server implementation [5] as the prime main achievement, currently in public github working repository. For design preliminary document, see

https://github.com/streamline-eu/WP2-MidtermReport/blob/master/Parameter%2BServer%2Bdesign%2B2.doc

- New measurements with improved algorithms, preliminary report on public data in [6].
- Intensive measurements for IMR and Altice Labs. Preliminary documents in Streamline github,

https://github.com/streamline-eu/WP2-MidtermReport/blob/master/IMR-TextClassification.doc

And

https://github.com/streamline-eu/WP2-MidtermReport/blob/master/SettopBoxRecommendationExperimentReport-RAW.mht

These describe the experiments in detail and will appear in polished form in D2.2 and D5.4.

Advanced methods, *depending on use cases, potentially including* gradient boosted trees, kernel methods, implicit and explicit ALS and tensor factorization, differential privacy and peer-to-peer recommenders.

- Boosted trees: We started our own decision tree implementation. Due to batch API issues, we abandoned this development direction. Our new connector to XGBoost will be described in D2.2, which completely replaces the stale code with poor design and outdated interfaces (Flink 0.10) that do not work with recent Flink versions (1.3+), https://github.com/dmlc/xgboost/tree/master/jvm-packages/xgboost4j-flink
- We updated Flink version to latest. The implementation will be described in D2.2.
- At present, we have the methods [1,2,3,5,6,7,9], all of with are planned to be or already pull requests by M24. We rely on two former team member committers to pull into Flink main, Marton Balassi (Cloudera) and Gyula Fóra (King.com), both of them are active Flink developers as part of their present affiliation.
- New elements compared to the description are the Passive Aggressive classifier [9] and other recommenders e.g. nearest neighbor that will be reported in D2.2.
- Instead of special topics in recommender systems, we have work in progress in two areas:

- deep learning evaluated for recommenders and Flink implementations are considered for use;
- a text trend analysis use case is worked out to demonstrate sketches.

4.3 Work Package 3

4.3.1 Contributors:

Beneficiary	**Effort in PMs**
SICS (lead)	14.42
DFKI	7.0
ALB	1.28
SZTAKI	7.5
NMusic	0.92
IMR	0.82

Work Package 3 is designed to address the Objectives I and III of the project. The Work Package focuses on practical usability of STREAMLINE Flink, addressing the "everyday needs" of Flink application developers and data scientists who may lack the in-depth knowledge and/or time of Flink deployment, configuration, operation, optimization, and interaction with the environment such as HDFS and Kafka streaming services. Furthermore, the Work Package aims at raising the abstraction level of programming in Flink by capitalizing on fundamental contributions to Flink being done in the scope of STREAMLINE WP1. Finally, the projects' achievements and contributions to Flink are being evaluated both from the technical perspective (Flink scalability and performance), as well as usability for human developers.

4.3.2 Deliverables:

D3.1 Flink deployment software (M12)

Objective	Results	Next steps	Any deviation from plan
Flink as a service	Flink deployment software based on Karamel/Chef for common types of cloud, cluster and	Flink deployment on Apache Yarn (resource manager in Hadoop) needs rework such that	

	virtualized environments. Flink is integrated as a service in Hops - an improved distribution of Apache Hadoop.	Flink uses Yarn-managed resources (containers) for all types of its processes. Also, the Flink implementation of SSL-based security will be evaluated for practical use in Hops/Hadoop.	
High Level Declarative Language	Extended SQL on Dynamic Tables. Integration of SystemML with Apache Flink which is currently halted due to limitations on Apache Flink. (FLINK-1730, SYSTEMML-637)	The work on designing the extended SQL is still ongoing. We have made proposals to address Flink's limitation for integration of SystemML and Flink	
Interactive Processing	I^2		
Holistic Evaluation and Benchmarking	Survey and Benchmarking Tool	Design benchmarking tools for hybrid workloads	

Detailed explanation of the work done for each objective is explained below.

4.3.2.1 T3.1 Platform as a Service

Flink is available now a as service in Hops - an improved "Hadoop for the humans" distribution of Apache Hadoop. Hops can be deployed just with a few mouse clicks on common cloud-, bare-metal cluster and virtualized environments (AWS/GCE/OpenStack/Vagrant). In this Task, Flink deployment scripts for Karamel/Chef were developed. Karamel is an orchestration engine for time- and resource-efficient, parallel provisioning of computing resources and installation and configuration of software services. Karamel/Chef is used for installation of all Hops services in a uniform fashion. The new STREAMLINE Karamel/Chef installation scripts for Flink are, to the best of our knowledge, the only solution for automated Flink deployment.

Hops provides other services, in particular - an scalable version of HDFS for batch data, and Kafka service for streaming data, all accessible through an easy-to-use Hops user interface called Hopsworks. Hopsworks allows to define projects with their data sets and streaming data services, and share those upon need in a multi-tenant environment in a controlled manner. Project data can be visualized using e.g. Apache Zeppelin.

Furthermore, Hops provides services to simplify programming of Flink applications accessing Kafka streaming services, taking over the tedious tasks of dealing with security, data schemes, and integration with the Hopsworks environment.

Publications:

1. Salman Niazi, Mahmoud Ismail, Mikael Ronström, Steffen Grohsschmiedt, Seif Haridi, Jim Dowling. HopsFS: Scaling Hierarchical File System Metadata Using NewSQL Databases, USENIX FAST 2017.
2. Salman Niazi, Mahmoud Ismail, Mikael Ronström, Seif Haridi, Jim Dowling. Scaling HDFS to more than 1 million operations per second with HopsFS, IEEE Scale Prize Winner, May 2017
3. Mahmoud Ismail, Ermias Gebremeskel, Theofilos Kakantousis, Gautier Berthou, Jim Dowling. Hopsworks: Improving User Experience and Development on Hadoop with Scalable, Strongly Consistent Metadata, ICDCS Demo, 2017
4. HopsFS - the fastest hierarchical distributed filesystem. IEEE Scale Prize Final, Madrid, Spain. 15 May 2017
5. Hopsworks Demo, ICDCS, Atlanta, USA, 5 June 2017

4.3.2.2 T3.2 A High-Level Declarative Language for Machine Learning at Rest and in Motion

An extension of SQL language was designed to allow querying over both batch and streaming data. It uses the Dynamic Table that allows "conversion" of Streams in to Tables and vice-versa. The hybrid query language designed based on the stream-batch join mechanism discussed in Work Package 1. Table below shows the list of different types of joins

Processing Time	**"SELECT * FROM R, S WHERE R.A = S.A AND JOINED_TIME(R.proctime, S.proctime)"** → no buffering, immediate execution
Processing Time Batching	**"SELECT * FROM R, S WHERE R.A = S.A AND JOINED_TIME(R.proctime, S.proctime)" & Consistent Trigger** → Black/white buffering, execution if checkpoint is completed
Event Time	**"SELECT * FROM R, S WHERE R.A = S.A"** → Buffering sorted by timestamp, filtering by watermark
Event Time Batching	**"SELECT * FROM R, S WHERE R.A = S.A" & Consistent Trigger** → Buffering sorted by timestamp, filtering by watermark, black/white buffering

We also attempted to integrate SystemML and Apache Flink. SystemML is an Apache project allows user to write scripts in R language which are then executed on scalable data processing platforms. SystemML performs several optimizations that are specifically suited for machine learning workloads. Currently, SystemML support Hadoop M/R and Apache Spark as its processing engine. We integrated Apache Flink into SystemML (SYSTEMML-637). The submitted pull request (https://github.com/apache/systemml/pull/119), however, was rejected because of

performance issues. After investigation, the source of issue was as the inability of Apache Flink to cache intermediate results. FLINK-1730 addresses the very same issue. We are currently assessing the feasibility solving the Flink issue.

Publications and Documents:

1. Kunft, A., Alexandrov, A., Katsifodimos, A., & Markl, V. (2016, June). Bridging the gap: towards optimization across linear and relational algebra. In Proceedings of the 3rd ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond (p. 1). ACM.
2. Alexandrov, A., Salzmann, A., Krastev, G., Katsifodimos, A., & Markl, V. (2016, June). Emma in action: Declarative dataflows for scalable data analysis. In Proceedings of the 2016 International Conference on Management of Data (pp. 2073-2076). ACM
3. https://issues.apache.org/jira/browse/FLINK-1730
4. https://issues.apache.org/jira/browse/SYSTEMML-637
5. https://github.com/apache/systemml/pull/119
6. https://systemml.apache.org/

4.3.2.3 T3.3 Interactive processing

We developed I², an interactive development environment that allows real time visualization of streaming data. I² coordinates the running cluster applications and corresponding visualizations such that only the currently depicted data points are processed and transferred.

Publications and Documents:

1. Traub et. al. Optimized On-Demand Data Streaming from Sensor Nodes, SoCC 2017
2. Traub et. al. I²: Interactive Real-Time Visualization for Streaming Data, EDBT 2017
3. I²: Interactive real-time visualization for streaming data with Apache Flink and Apache Zeppelin, Flink Forward 2017

Repository:

- https://github.com/streamline-eu/i2

4.3.2.4 T3.4 Holistic Evaluation and Benchmarking

To evaluate the human latency, we designed a survey that requires the participants to compare the implementation of two hybrid (batch and stream) use cases in Flink and Streamline APIs. The survey was performed in conjunction with Streamline Hackathon in Munich (19th, 20th August 2017). For both use cases, only less than 10% of the participants deemed Flink APIs more readable. The majority of participants (80-90%) favors Streamline implementation of the use cases. More specifically, they agreed that Streamline API manages to hide the cumbersome detail of loading and preprocessing the static datasets.

To establish a baseline for evaluating the efficiency of the Streamline hybrid engine, we first designed a benchmarking tool that measures the throughput and latency for aggregation and windowed join under different configurations for Storm, Spark, and Flink.

We designed another benchmarking suite for evaluating the efficiency of running large machine learning workloads on distributed dataflow systems. We benchmarked both Apache Flink and Apache Spark for both supervised and unsupervised machine learning tasks. The results show that

while both systems perform comparable in most of the cases, Flink outperforms Spark when the dataset does not fit in memory and Spark outperforms Fink when the dataset can completely resides in memory.

We have proposed the designed benchmarking tool to SPEC Research group and the TPC, for integration with current standards.

Publications and Documents:

1. Karimov et. al. . Stream Processing Performance in Online Game Scenarios *(Paper under review)*
2. Rabl, T. et. al. . Analysis of TPC-DS - the First Standard Benchmark for SQL-Based Big Data Systems, SOCC 2017, *to appear*
3. Rabl, T. (2016, September). From BigBench to TPCx-BB: Standardization of a Big Data Benchmark. In Technology Conference on Performance Evaluation and Benchmarking.
4. Efficiently executing R Dataframes on Flink, at Flink Forward 2017.
5. PEEL: A Framework for benchmarking distributed systems and algorithms

Repository:

- https://github.com/streamline-eu/Streamline-Survey-2017
- https://github.com/streamline-eu/StreamBenchmarks

4.4 Work Package 4

4.4.1 Contributors:

Beneficiary	**Effort in PMs**
ALB	2.87
NMusic	2.94
IMR (lead)	12.67
ROVIO	0.25

Work Package 4 is designed to address the Objectives I and II of the project.

4.4.2 Deliverables:

D4.1

Objective	Results	Next steps	Any deviation from plan

Have business experts (marketing, BI) identify relevant types of events potentially impactful for their business KPI	Various event source types evaluated and tested. Unsatisfactory results in most cases. KPIs, architecture, services identified for NMusic and Altice Labs. NMusic: identified and scraped 40 sites of interest, collected business stats.	Focus on web scraping	The main conclusion is that a business activity is affected by a very specific type of events that can mostly be found on specialized web sites that need to be scraped.
Scale source identification and processing using stream classification and machine learning to enable critical mass of events time series to be created	Beginning of the project: the scraping process were semi-supervised. Now: fully automatic extraction, relies on ML methods (some explanations below), and application domain heuristics.	Automatic extraction has been fully deployed for the eCommerce application domain. Classifiers and specific heuristics for events remain to be designed, tested and implemented.	Scalable event extraction is now possible. The scraping technology has been indirectly demonstrated with the Bomerce app.
Calculate a method to measure Context Impact per event type for each business case to determine where they are useful to augment predictive power of current models	We devised a methodology to match events and business times series. Preliminary evaluation of off-the-shelf algorithms (TS decomposition, and rule mining).	Evaluation of real use case	Although seductive in theory, the methodology appears quite speculative, as it requires a very dense collection of relevant events, and qualified business descriptors (i.e. including geolocation). These conditions were far from being guaranteed, hence the choice to re-orient the WP.

4.4.2.1 Task 4.1 Event Type Selection via Business Intelligence

We investigated a list of source types beyond Web crawling: iCal documents, APIs, Wikipedia logs. For each we implemented an extractor as an extension of our crawler and initialized a database of events. We then chose to focus of web scraping for several reasons that all relate to the inability of the other sources to meet the needs of our partners use cases: (i) general-purpose event sources are not focused enough, making interesting events for a specific use case extremely rare, (ii) several sources (other than Web scraping) are either not reliable enough, or incurs an additional cost (event delivery services), (iii) it appears that most of the events of interest can only be found on specialized web sources, some not open.

Characteristics of the investigated source types are reported in D4.1.

4.4.2.2 Task 4.2 Large scale Events sourcing

At the beginning of the project, the scraping methodology relied on an annotation mechanism. This method is the one reported in D4.1. We recently achieved a fully automatic crawl and extraction technology, which relies on a combination of ML algorithms and domain specific heuristics.

- *ML algorithms.* Recall that (as explained in D4.1) we aim at finding two categories of pages: *catalog pages* refers to a list a items, and *item pages* fully describe a specific item. In the context of event sites (WP4), the catalog page refers to a list of events, the details of which are exposed in item pages; in the context of eCommerce sites (W5), we deal with list of products, and product offers. The semi-supervised approach has led us to constitute a large DB (approx. 2,5 M) of wrappers for either catalog and item pages. We used this DB to learn a classification model and detect, at crawl time, whether a crawl page belongs to one of these categories.
- *Domain specific heuristics*. In order to produce a wrapper for an item page, we developed a set of heuristics that currently focus on the eCommerce domain. They are based on micro data exploitation, on-line analysis of the page structure, and matching of the data common to the catalog and item page. In short, one can find in the catalog page a summary of each referred item (for instance the name and the price of a product) and this allows to detect the location of the common features in the item page.

This technology, now fully operational for the eCommerce use case (as demonstrated during the demo) can be transposed to events detection and extraction. This requires the production of a training base, in order to automatically detect event sites ate crawl discovery time, and investigations on event-specific heuristics.

4.4.2.3 Task 4.3 Contextual Impact Measurement

We worked during the first phase of the project on the design mathematical methods to identify remaining variations and to measure impact of events series on these variations (as required by the DOW). These methods were intended to be developed during the second phase of the project and are not presented in D4.1. In short, the method idea is to decompose times series of user activities in order to detect “outliers”, i.e., peaks that cannot be related to some regular feature of the TS (trend or seasonality). These peaks are candidate to match an event, and the repeated

occurrence of (peaks, event time) would suggest a rule stating that such event types influence the user behavior.

We studied the candidate for the two main components of the method, selected the STL algorithm for Time Series Decomposition, and apriori for the association rule mining component. A prototype in Python has been implemented and tested on public business reports (retail sales in the US).

4.4.3 Refactoring of W4: explanations and new objectives

Key findings from the first phase.

First, it is very hard to collect enough events to find a significant subset that can match with a user activity, given that this activity takes place in a specific location, at a specific time, and might be influenced by a bunch of factors, most of which falls out of the scope of a contextualization engine. During the use case study, we found that sources apt at providing dense coverage of relevant events are mostly highly specialized web sites, the list of which has to be supplied by the use case partners. Even though we developed during the first phase a scalable scraping technology, implementing a full crawling and extraction system dedicated to the discovery and scraping of event-related sites would have required considerable efforts. A second important issue is that matching events and user activity requires full access to the latter, including important features such as geolocation. This raises privacy issues, and restricts the deployment of a full-fledged architecture.

In the context of Streamline, after the departure of NMusic, the only remaining use case (Altice labs) were highly affected by those two issues. First, users descriptions did not feature the geolocation, which made uncertain the determination of event sources that could affect them, second Altice Labs did not want to disclose use activities. We considered that, in this situation, we would not be able to build with enough confidence in the viability of the approach, in the scope of the project, a contextualization system.

New approach.

New plan: use contextual information to support recommendation algorithms. Contextual here means no longer events but textual description of the recommended items, either part of the items description, or collected from external sources.

Recommendation can either use contextual information in conjunction with user profiles and activities, or rely only on contextual information to solve the cold start problem when a new user or new item enters the system

Two use cases internal to the project:

1. TV recommendation, with contextual information being TV programs, and possibly comments and reviews obtained from social media. (Altice Labs)
2. Product recommendation, based on product description, and user comments extracted from web pages (IMR)

In both cases, the goal is to process textual data attached to items to infer similarity and boost the recommendation engine.

Yize Li, Jiazhong Nie, Yi Zhang, Bingqing Wang, Baoshi Yan, and Fuliang Weng. 2010. Contextual recommendation based on text mining. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 692-700.

4.5 Work package 5

4.5.1 Contributors:

Beneficiary	**Effort in PMs**
DFKI	4.0
ALB	8.31
SZTAKI	4.0
NMusic	11.92
IMR	13.61
ROVIO (lead)	9.56

Work Package 5 is the work done in the industrial use cases that both drive the requirements of the technical Work Packages I-III as well as provide validation.

4.5.2 Deliverables:

D5.1 and D5.2

Objective	Results	Next steps	Any deviation from plan
Requirements and Design of industrial applications	Software development specification of use cases prototypes (v1) from three partners as reported in D5.1.	Design of use cases for pilot (v2) and production platform (v3).	Rovio does not follow the release cycle of having prototype, pilot and production deployments in successive years. Instead Rovio

			implements multiple smaller use cases where some of them are already in production.
Data and Corporate Systems Integration	Apache Flink deployed in the data centers of each partner and integrated to data flows.	Implement enhancements to deployment and monitoring. Integration of more data sources, e.g. in Rovio some games are not yet sending their client analytics events to new Kafka endpoint. Better utilization of STREAMLINE deliverables from academic partners.	Rovio uses AWS instead of on-premise deployment; Rovio STREAMLINE features used mostly from Apache Flink main branch.
Use Case Implementation	Prototype use cases implemented (v1) and details reported in D5.1	Implementation of pilot (v2) and production (v3) use cases.	Rovio release cycle.
Validation and Evaluation	Prototype use cases were released and their performance reported in D5.2.	Release and performance analysis of pilot (v2) and production (v3) use cases.	Rovio release cycle.

Detailed explanation of the work done for each objective is explained below.

4.5.2.1 T5.1 Requirements and Design

This task consisted of requirement analysis and design of the industrial use cases. The purpose of this task was to clarify the requirements for the academic partners and produce the software design specification for T5.2 and T5.3.

Altice Labs

- Identification of requirements for ALB use cases, with user roles together with a description of each of the three use cases - real-time analytics and prediction, real-time profiling and real-time recommendation -, detailed through Epics and User Stories;
- Design and implementation specifications (prototype - v1), starting with the global architecture of the system, identifying planned technologies and the relationships between actors.

- Ongoing at M18 (to be reported in D5.3 - M21)
    - Previously identified requirements are being consolidated, namely detailed analytics and profiling as well as TV recommendation requirements;
    - Global Architecture processing steps updates and new processes based on Apache Flink and Apache Zeppelin is being introduced and its implementation will be subsequently described, namely:
        - Data processing (E2E data processing and data models),
        - Client profiling
        - Analytics visualization
    - Recommender engines design and implementation will be presented and described
        - Hybrid Recommendation Engine - Content and Collaborative Filtering
        - Flink based Collaborative Filtering Recommendation Engine (in articulation with SZTAKI work in WP2)

Rovio

- Requirement analysis and design of use cases:
    - Real-time profiling and KPIs by streaming data to 3rd party analytics provider (v1, continues with other vendor in v2 and v3)
    - Recommendation system for streaming gaming service (v1)
    - Custom real-time aggregation stream to Grafana dashboard (v1-v2)
    - Real-time feature extraction for churn prediction (v2)
    - Data backup to S3 (v2)
    - User journey analysis using Gelly connected components (v2)
- Requirement analysis and design of features required for running Apache Flink in Rovio architecture such as:
    - Kafka upgrade
    - Data collection
    - Deployment
    - Orchestration
    - Monitoring

IMR

- Identification of requirements for IMR use cases;
    - Crawl
    - Extraction
    - Product classification
    - Notification
- Model of the dataflow - focus on the product classification use case.
    - Identification of components
    - Choice of the classification algorithm
- Ongoing at M18 (to be reported in D5.3 - M21)
    - Architecture based on Apache Flink
    - KPIs
    - Collaboration with Sztaki on the distributed classification algorithm

4.5.2.2 T5.2 Data and Corporate Systems Integration

Altice Labs

- Identification and description of the data streams and the data collection framework, followed by envisaged contextual data (description in D5.1), namely:
    - Activity Logs, EPG, External data
- Ongoing at M18 (to be reported in D5.3 - M21)
    - Existing data sources updates and new data sources will be introduced and described, namely:
        - Activity Logs (v2)
        - Internal and external contextual data - EPG, Channels, Catalog

Rovio

- Implementation of generic platform features defined in T5.1
    - Setting up new Kafka cluster (v1)
    - New data collection endpoint (v1)
    - Ability to configure the data collection endpoint for clients in server side (v1)
    - Integration to GitHub and TeamCity for code deployment (v1)
    - Nagios monitoring (v1)
    - Azkaban workflow manager integration (v2)
    - Support for persistent states by enabling external checkpoints and savepoints (v2)

IMR

- Setup of a small cluster (5 machines) supporting the new architecture (evaluation purpose)
- Implementation of a crawler/kafka connector
- Ongoing at M18 (to be reported in D5.3 - M21)
    - Integration of the Flink implementation of the classifier
    - Test of crawler/kafka/flink/HBase integration

4.5.2.3 T5.3 Use Case Implementation

Altice Labs

- Implementation of the use case (v1) defined in T5.1 and T5.2
- Ongoing at M18 (to be reported in D5.3 - M21)
    - Implementation of use case (v2)as defined in T5.1 and T5.2

Rovio

- Implementation of the use cases defined in T5.1

IMR

- Integration of the dataflow operators in Flink

4.5.2.4 T.5.4 Validation and Evaluation

Altice Labs

- Field trial evaluation of IPTV Recommender use case (v1), namely (as described in D5.2)
    - Identification of overall KPIs describing current system situation and proposed initial baseline and target measures
    - Experimental Setup (Laboratory) to evaluate tentative technologies, namely

- Data ingestion: Message Broker Systems - SAPO Broker and Apache Kafka
- Deployment, configuration and automation - Ansible and Chef

Rovio

- During first year (v1) Rovio was running 3rd party system integration use case by integrating one game in technical soft launch to 3rd party analytics vendor. The reliability of the solution was measured by service up-time percentage which was on acceptable level. The business effect was measured with comparing dashboard visits of real-time dashboard with batch based dashboard. Dashboard visits were lower in real-time dashboards mainly because batch based dashboard had more information: people preferred richness of dashboard over timeliness. Recommendation system use case was discontinued because the core STREAMLINE team was moved from Hatch business unit to Rovio Games and streaming service projected decided to use 3rd party for recommendations. These results have been reported in D5.2.
- During second year (v2) Rovio has run custom aggregation stream and real-time feature extraction for churn use cases in production for all priority games and services. Graph analysis and data backup to s3 use cases were only prototyped. Results will be reported in D5.4.

IMR

- We run in production a stable Hadoop/MapReduce implementation of the workflow which suffers from high latency (> 1 day). The goal is to achieve the same quality with Flink, running the PassiveAggressive classifier implemented by Sztaki, and lowering the latency to a few hours (2+ hours targeted)
- Investigation of priority queues using kafka/Flink connectors to manage high priority items.

4.6 Work package 6

4.6.1 Contributors:

Beneficiary	**Effort in PMs**
SICS (new lead)	0.8
DFKI	2.0
ALB	1.68
NMusic (lead)	1.02
IMR	1.69

NMusic started this work by supplying STREAMLINE with internal templates and resources necessary for internal use. An initial web presence has been created. Due to NMusic's departure it has not been maintained to the high standard STREAMLINE is aiming for.

Corrective measures have been made and work is now in progress to update the web. The work to actively engage in social networks has not been as successful as planned. Our twitter account only has 22 followers. A plan is being put in place to improve this substantially. This is also true for LinkedIn.

The project has maintained a web presence at www. h2020-streamline-project.eu for public information and dissemination. The activity is

The project has been active in social media via Twitter as the handle is @h2020streamline.

The work in the research Work Packages is showing good results with papers presented at important conferences, such as EDBT and VLDB.

The community and SMEs have been included by participation in meetups and other fora.
Published papers are either referenced above or will be reported in D6.2.

Concertation with other EU initiatives has started. STREAMLINE has participated in an initial meeting with the project PROTEUS. We presented to other projects at a workshop at the EDBT/ICDT conference (http://edbticdt2017.unive.it/?workshops). The consortium has also established links to the Big Data Value Association as well as other potential users of the results.

Further details in the deliverable D6.2 are available.

4.6.2 Deliverables:

D6.1, D6.2

4.7 Work package 7

4.7.1 Contributors:

Beneficiary	**Effort in PMs**
SICS (lead)	7.61
ALB	0.38
NMusic	0.39
IMR	1.13

Work Package 7 is project administration. It also includes the management of ethics issues.

The coordinator has arranged and managed 6 face-to-face project meetings and 2 Skype project meetings as well as regular project calls every 2-3 weeks. Two meetings of the PMC has taken place to decide on the amendments executed.

- Stockholm, December 2015. Technical Kick-off
- Luxemburg January 2016. Project kick-off with PO
- Berlin, January 2016. Project meeting incl. PMC and PTC
- Stockholm, June 2016. Project meeting incl. PMC and PTC
- Berlin. September 2015. Project meeting incl. PMC and PTC (In conjunction with FlinkForward)
- Paris November 2016. Project meeting incl. PMC and PTC
- Skype, January 2017. Project meeting incl. PMC and PTC
- Skype, April 2017. Project meeting incl. PMC and PTC

More than normal time was also used on two amendments. The first amendment was due to the fact that beneficiary Portugal Telecom (PT) changed name to MEO. That should have been easy but in the middle of that work the legal entity and name also changed to Altice Labs (ALB), prompting the need for a full-blown amendment and approval process. This requested change happened before the formal start of the project. During the process this was not entered correctly in the system. We tried to change this in the second amendment but failed. This has resulted in the fact that it still formally looks like PT was doing work M1-M9 and then ALB M9-M18 when in fact ALB should be entered as M1-M18 and PT never did enter the project.

The second amendment was to terminate NMusic and rearrange the work among the beneficiaries to ensure all objectives can be met. This took longer than expected and the coordinator failed to anticipate but the work burden and also the timing of this in the middle of the summer which let to late deliveries of the reporting deliverables.

Ethics was flagged for the STREAMLINE project since the term “human latency” was used to describe normal testing of the software platform. We have not yet conducted any testing with users other than internally in the use cases. For the coming hackathons and other test we are committed and serious about personal privacy and we will adhere strictly to all privacy laws as well as making sure we have informed consent. STREAMLINE also use data from the use cases. If the data is used inside the beneficiary owning the date it is made sure that the use follow the user agreements regarding use of the data. Data that have been used at the research partners have been anonymized first. This process is described in D7.2.

4.7.2 Deliverables:

D7.1, D7.2, D7.3, D7.6.

5 Amended work during the first period

During the first period partner NMusic withdrew from the project due to business reasons. NMusic provided a use case for the project. Until its termination M12 NMusic carried out all work and contributed to the relevant deliverables. After NMusic left all tasks were re-assigned with no

impact on the project and its end results. The impact of the partner leaving will be one less use case. However, the project still includes three strong and relevant use cases that will provide good evidence that the technology is relevant and useful and will achieve its impact goals.

The withdrawal of NMusic prompted the need to reassign work and prepare an amendment to the Grant Agreement. Due to many circumstances, mostly due to the fact that the Project Officer left the project and the lack of a new Project Officer during an extended period this amendment was delayed. It was finally accepted on June 27 2017 as AMD-688191-10. This has not impacted the work carried out according to the Grant Agreement but is delayed the deliverables D6.2, D7.3 and D7.6 all of which are only reports and not critical to the success of the project.

On the request of the then current Program Officer the consortium assessed the impact of NMusic leaving and this chapter outlines the changed work for achieving the same or increased impact of the Streamline project. As part of this assessment we have also went through the GA very carefully looking at what might have changed since the proposal was written two years ago. We wanted to make sure we change what had not survived after the proposal was written be it because of external developments or research findings during the first year.

The consortium's opinion was that in general the Grant Agreement is in good shape and aligned with the objectives set out in the proposal. However, as always, we saw the need for updates, additions due to requirements found and redistribution of effort of the work assigned to NMusic.

This chapter describes the impact of NMusic leaving and the changed work during the first period and the coming period.

5.1 Changes to the work plan

A similar use case would have been good but the candidates available to the consortium were not able to commit to this work on such short notice.

What we instead did was to transfer the tasks assigned to NMusic that are important for the delivery of the Streamline project. The work on specific tasks for the NMusic use case (WP5) was ended. The relevant work carried out by NMusic have been reported in the deliverables D5.1 and D5.2. The rest of the resources assigned to NMusic were redistributed for work to ensure the delivery of the work left by NMusic.

5.1.1 WP transfers and reorganization

- WP1
  - No impact
- WP2
  - The recommendation system work was transferred to IMR and the new WP4. Work was transferred to SZAKI to be able to deliver on WP2 with the new organization of WP4.
- WP3
  - We transferred the work from NMusic to the other partners for evaluation and KPI measurements by strengthening WP5.
- WP4

  - The work on implementation of contextualization that was to be performed by NMusic was not transferred due to the research findings the first year. A report was written on the results from the preliminary work on contextualization. WP4 will instead be used to support the research needed for the industry requirements on text recommendation systems supported by a strong use case by IMR.
- WP5
  - Work was transferred to the other use case partners in the evaluation and the KPI measurements. The NMusic specific work was removed.
- WP6
  - Dissemination was picked up by the coordinator SICS.
- WP7
  - SICS used the work transferred from NMusic for rework and added administrative burden.

5.2 STREAMLINE vs Apache Flink

Clarifications on the impact of the research and Apache Flink vs Flink Streamline.

The research done in Streamline aims to have industrial impact and one of the ways is by strengthening the European analytics platform Apache Flink. As it is an Open Source project our results will be offered to the community in the with the aim to make it available in the main Apache Flink branch. The Streamline consortium is working closely with the Flink community to facilitate this process.

Parallel to this “STREAMLINE” and “Flink Streamline” is the vision for the **research** related to a unified system for batch and streaming. This will be delivered as Streamline Flink as a prototype platform where we, as the goal is stated in the proposal, will achieve a TRL of 7. The use case partners will use this platform to verify the unified streaming and batch system.

The Description of Work part B is ambiguous regarding this division of production environment and research platform and an attempt will be made to clarify this in the coming RP2.

5.3 Added work during first period

The proposal was based on strong interaction between the academic partners and the use case partners. This is implemented as three “design cycles” where requirements are fed back into research and reported in the deliverables D5.1, D5.3 and D5.5. During the first year as documented in D5.1 and D5.2 this has been a very fruitful was to make sure the research is relevant to industry and agile to consider new observations, development and requirements. This work has led us to identify the following new or extended work items in the GA that address the original promises.

- There is a strong need in industry use cases for improved support for streaming SQL.
- The use cases have been very good to pinpoint shortcomings in the initial scope for the streaming ML algorithms.

- The need from industry for improved and evolved recommendation system is stronger than anticipated. Especially mining text.
- The platform needs to be hardened to move from research and to able to support a production environment. This impacts robustness and support for multi-tenancy.

This evaluation of opportunities lead to the following additions being approved in the amendment AMD-688191-10. This work is scheduled for the second period of the project, RP2.

- WP1 have added a task and deliverable to operators to cover streaming SQL.
- WP2 will expand to meet the demand for new algorithms.
- WP3 have added a task and deliverable to add YARN support and integration with Hops version of Hadoop.
- WP4 will change focus on recommendation from text mining in RP2.

6 Conclusion

STREAMLINE went through an eventful year with four Project Officers, one partner changing legal status and one partner leaving the consortium. Despite this STREAMLINE was able to deliver on its promises of impact to the European Data Economy and see the eco system around Apache Flink grow as STREAMLINE contributions to the platform made it more attractive to enterprise use. The use case partners in STREAMLINE also benefited from using the technology developed in STREAMLINE. During the next period, many of the components will come together and provide an even stronger platform for Data Analytics on stream and batch data in one single system.