SEVENTH FRAMEWORK PROGRAMME

QualiMaster
www.qualiMaster.eu

QualiMaster

A configurable real-time Data Processing Infrastructure mastering autonomous Quality Adaptation

Grant Agreement No. 619525

Deliverable D1.1

Work-package	WP1: Use Cases
Deliverable	D1.1: Initial Use Cases and Requirements
Deliverable Leader	MAX
Quality Assessor	Stefan Burkhard
Estimation of PM spent	7
Dissemination level	Public (PU)
Delivery date in Annex I	30.04.2014
Actual delivery date	30.04.2013
Revisions	12
Status	Final
Keywords:	Use cases, actors, application scenarios

List of Authors

Partner Acronym	Authors
MAX	Steve Hutt
LUH	Mohammad Alrifai, Claudia Niederée
SUH	Holger Eichelberger
SPRING	Holger Arndt, Walter Colliti
TSI	Apostolos Dollas, Ekaterini Ioannou

Table of Contents

Executive summary

Collecting requirements about a system scheduled for realization helps defining the terminology, stabilizing the common vision and detailing the functional and quality requirements. This deliverable reports on the results of the early requirements collection for the QualiMaster project, and in particular the QualiMaster Applications for systemic risk analysis in the financial domain and the underlying QualiMaster infrastructure. We will present and discuss the actors who will interact with the applications and the QualiMaster infrastructure and, in particular, the initial descriptions of individual use cases, i.e., their specific interactions with the QualiMaster components. Furthermore, we provide initial requirements for the data streams to be processed by the QualiMaster infrastructure and the algorithms to be applied in a data analysis pipeline for systemic risk calculation.

1 Introduction

This deliverable summarizes the requirements that have been collected in the QualiMaster project for the two targeted QualiMaster application scenarios and the underlying QualiMaster infrastructure. This document has to be considered as a working document, which - in line with the planning of the work in workpackage WP1 - will be further extended and refined in the upcoming months of the project. It provides the basis for further discussions of the QualiMaster requirements in the project and for identifying further dependencies between the different components of the system. An updated version of the requirements collected in the project will be documented in deliverable D1.2, which is due in month 12 of the project.

For collecting the requirements, a use-case based approach has been chosen, which enables an intuitive and user-centered starting point for better understanding and discussing the system functionality. As a first step in describing the use cases, a set of stakeholders or key actors have been identified, which interact with the system in different ways and in different roles.

In addition to describing the use cases of the two QualiMaster applications from the financial domain, we also compiled a set of system use cases for covering the core functionalities and actors of the QualiMaster infrastructure. This addresses the configuration of processing pipelines, the management of the adaptation (the flexible adaptation support is one of the special features of the QualiMaster infrastructure) and the characteristic aspects of the set up and administration of the overall QualiMaster infrastructure.

The description of the stakeholders and the use cases is complemented by the identification of requirements towards the data and the description of relevant quality requirements for the system. This provides the basis for a more detailed description of non-functional requirements in D1.2. They also provide important input for WP2 and WP4, which are both concerned with quality aspects of the processing pipelines.

This collection and documentation process of the core use cases has already served as a good trigger for discussions about the functionalities, terminologies, and dependencies within the consortium.

1.1 Requirements Collection Approach

Use cases are a popular means for collecting requirements in a user-oriented way. Starting from a set of actors, i.e., a set of persons or other systems that interact with the system under consideration, use cases describe the flow of interaction of those actors with the system. The advantages of use cases are that they are very intuitive and easy to understand due to their textual form. Furthermore, they do not only support the description of the normal flow of interaction (the so called use case scenario), which helps in the identification of required system functionalities. They also foster the description of exceptional cases, which already gives a broader picture of the expected system functionality.

One of the disadvantages of use cases is that they are restricted to functional requirements. Therefore a separate part has been added to this deliverable, which documents non-functional requirements especially with respect to the data and the quality requirements, which have been identified during the discussion of the use cases.

For the documentation of the use cases tables have been used, which are a simplified form of the table-based templates suggested by Cockburn [5] for this purpose. Each table contains

- the use case name and an unique identifier,
- the involved actors,
- the goal of the use case,
- the preconditions for the use case and the postconditions that are established by successful use case execution,

- the scenario description (interaction steps) for successful execution of the use case (typically the actor and the system alternate in their interaction),
- the description of exceptional cases in the interactions and interaction variants,
- the business constraints for the use case (business rules) and
- the processed data or employed system functionality.

The use case identifier is assigned to the use cases for easing cross-referencing between use cases. In particular, referenced use cases in preconditions are intended to be transitive across all references so that indirectly referenced use cases do not need to be listed explicitly.

1.2 Application Scenarios Overview

The QualiMaster project will validate its results in terms of two application scenarios on systemic risk analysis, one focusing on institutional financial clients and one on regulatory bodies. The use cases for the application scenarios will be discussed in more detail in Section 4.

Systemic risk had been identified as a key factor in the stability of financial markets. In its broadest form, it represents scenarios where financial markets, and the exposure of financial institutions to those markets, become strongly correlated or coupled, potentially leading to industry-wide institutional failure. The need to measure and identify predictive signals of systemic risk is one of the most challenging issues facing institutional users and market regulators today. This is particularly so given the velocity of financial markets and the need to rapidly identify and act on hotspots before contagion sets in. It is well understood that today's financial markets are correlated to a degree significantly greater than historically and not simply within asset classes. Due to the Risk-On-Risk-Off nature of today's trading, broad ranges of asset classes have become strongly correlated, so that diversification of risk is more challenging. The analysis will be complemented with taking into account Social Web data for supporting and stabilizing the prediction of systemic risk, which we expect can be used for identifying additional indicators, and for contextualizing risk predictions.

The application scenario targets **institutional financial clients**, such as Hedge funds, Banks or Asset Managers. In this scenario, we aim at pre-trading risk analysis and real-time, real money trading risk analysis. The implementing applications will be integrated with the trading applications of SPRING and enhance them by a multi-variant and multi-market risk analysis approach, which is expected to add important insights about systemic risks, and help avoiding fatal losses of capital under management. As a collateral outcome, this will help stabilizing the capital markets at the roots, i.e., within the trading system of the financial industry.

In the second application scenario for **regulatory bodies**, QualiMaster aims at providing regulatory authorities with an early warning system for impending financial stress. We envisage that the underlying techniques can be integrated into financial exchanges and used to monitor the integrity of financial markets. Multivariate, real time analysis of exchange traded products, allows the exchange more sophisticated circuit breakers, rather than sudden trading time outs. By analyzing trading flow, market imbalance and liquidity indicators, exchanges can then implement incremental mechanisms to modify trading patterns, thereby avoiding sudden market dislocation.

1.3 Components of the QualiMaster Infrastructure

QualiMaster aims at a configurable infrastructure for real-time data stream processing, which adapts itself to the actual needs and the runtime requirements imposed by actual data streams. In this section, we discuss the basic components of the QualiMaster infrastructure in order to introduce the background. Please note that this section does not aim at introducing the overall system architecture of the QualiMaster project, as this the architecture is currently under development and will be described in Deliverable D5.1 (due in month 7).

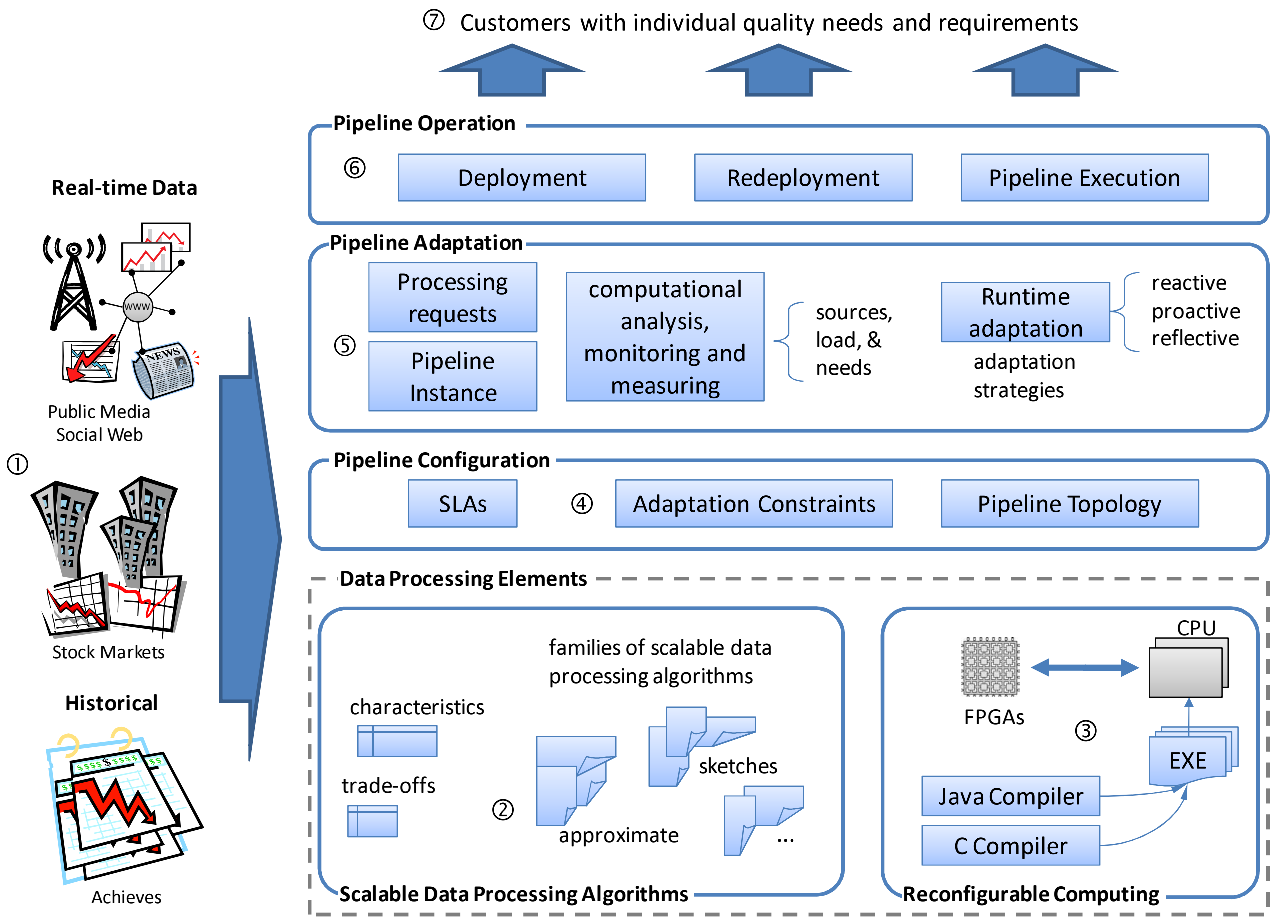

Figure 1: Overview of the QualiMaster Infrastructure

Figure 1 depicts an overview of the QualiMaster infrastructure as also envisioned in the Description of Work (DoW). Basically, this figure was communicated at the Kick-off meeting to all participants and the authors of this document had this figure in mind while collecting and describing the QualiMaster use cases.

The **QualiMaster infrastructure** consists of an environment for adaptively executing data stream pipelines as well as related tools for configuring and managing that environment. We will call the execution environment the **QualiMaster platform**. The term **configuration** refers to the activity of setting up the platform itself and the data analysis to be executed on the platform. The configuration is performed by an expert human being and a proper configuration is a prerequisite for successful and efficient data analysis. Basically, the notion of configuration originates from Software Product Line Engineering (SPLE) [17, 23], a successful approach for systematic software reuse. By applying SPLE techniques, we aim at an efficient and consistent configuration (of a generic "template") of the QualiMaster platform in order to save time, effort and computational resources. In contrast, **adaptive execution** refers to autonomous activities carried out by the QualiMaster platform in order to maintain the actual quality of the data analysis and the efficiency of the use of the physical computing resources. In turn, the adaptive execution relies on the configuration that implicitly defines the boundaries and the validity of the autonomous activities.

Various **data sources** such as stock market data, public social media as well as collected historical archives will serve as the input for the data analysis ①. The actual data analysis will be performed by **data processing algorithms** ②, such as the identification of causality in multivariate times series. One core idea of making the data analysis in QualiMaster adaptable is the notion of **algorithm families**. An algorithm family is a group of algorithms performing the same analysis step, while the individual algorithms differ in their actual execution quality, i.e., requiring different amount of memory or producing more accurate results. Let us consider three algorithms *A*, *B* and

C for the identification of causality. As the algorithms perform the same task, they belong to the processing family for causality identification. However, algorithm *A* produces highly precise results, but requires vast amounts of memory. In contrast, algorithm *B* is imprecise within acceptable boundaries, while it consumes significantly less memory. Switching between these algorithms at runtime allows reacting on changing conditions in the data streams and the execution environment. However, in the described case, algorithm *B* will impact the quality of the result produced. In particular, the notion of data processing families applies naturally to algorithms, which are realized for specialized **reconfigurable hardware** ③, e.g., data flow computing hardware developed by the partner MAX. For example, algorithm *C* is precise, memory and execution time efficient, but requires a certain amount of specialized hardware, i.e., specifically configured FPGA processors (which of course must be present and available as they are an efficient but expensive and, thus, limited resource).

To perform a certain data analysis, data processing families are combined to a **data processing pipeline**. The data processing pipeline is at the heart of the **configuration** ④ of a QualiMaster platform, also specifying the execution hardware, the algorithm families, quality constraints for Service Level Agreements (SLA) and the adaptive behaviour. A data processing pipeline consists of sources, sinks, data processing elements and data flows [4] connecting data **sources** ① with **processing elements**, processing elements among each other and processing elements with data **sinks** ⑦. In QualiMaster, a data processing element of a data processing pipeline is realized through a specific algorithm family, thus enabling variants of the data processing pipeline, i.e., adaptive execution of data processing pipelines ⑤. One challenge for the adaptive execution is the selection (and modification) of the most appropriate algorithm within each processing element / family at runtime, i.e., to determine the actual pipeline instance for execution. This is supported by the analysis of the overall (end-to-end) quality of the pipeline and the impact on the data and processing quality introduced by the (combination of) variable data processing algorithms. In our example above, selecting algorithm *B* might save resources, e.g., in high load situations, but may also imply a reduced quality of the analysis results of the subsequent and, thus, entire pipeline. Finally, pipeline execution and adaptation is supported by the (low-level) QualiMaster platform in terms **pipeline operations**, ⑥ such as starting or stopping a pipeline.

As indicated above, adaptivity also needs knowledge about the underlying execution platform. To illustrate, if the data stream comes from Germany and there is a MAX dataflow system in Greece, it is entirely possible that some pre-processing will be done in Germany and the adaptive pipeline will have its next stage in Greece, but it is unlikely that the data can go back and forth a lot due to the communications overhead. However, if both the data and all resources are available in the same location, the platform may include multiple accesses of some resource in the adaptive pipeline.

1.4 Terminology

In this section, we introduce some further terms we will use throughout this deliverable.

- **Platform instantiation** is the process of turning the configuration into an executable and optimized (version of the implementation of the) QualiMaster platform. Akin to the term “configuration”, also the term “instantiation” (also called “product derivation”) originates from SPLE [17, 23]. Basically, the QualiMaster platform will be realized as generic but possibly not (fully) configured piece of software, which may include more functionality than actually required for executing a certain set of pipelines, e.g., measurement and monitoring mechanisms for a wider set of qualities. Based on the configuration, the process of instantiating the platform will turn the generic QualiMaster platform into a specific instance, e.g., adding, disabling or removing unused monitoring mechanisms. Further, it will take care of the appropriate integration of hardware algorithms and the hardware execution, including the choice for different strategies of realizing a dataflow as indicated in the geographically distributed example above.
- A **quality parameter** [20] is a measurable and quantifiable property of a computational element (also other terms are used in literature, e.g., *quality dimension* or *quality attribute*

depending on the community [10]). A computational element may be a data processing algorithm, a data flow, a data analysis pipeline or a physical compute resource. Examples for quality parameters are numbers of tuples per time unit (data flow or data pipeline level), execution time or memory usage (compute resource level). We will distinguish between primitive quality parameters directly measured by the infrastructure or the hardware (as the aforementioned examples) and derived quality parameters defined by the platform administrator (such as a domain-specific kind of throughput).
Primitive quality parameters will be built into the QualiMaster platform to be measured. However, quality parameters that remain unused in the configuration can be disabled in or removed from the running QualiMaster platform instance during platform instantiation.

- **Quality characteristics** describe how a set of quality parameters behave for a computational element over time in a certain setting, e.g., a financial data processing algorithm under high load. Typically, the quality parameters cannot be considered constant, so that mathematical or statistical means must be applied to capture quality characteristics appropriately. Quality characteristics are defined for individual algorithms, propagate to algorithm families (through the selection of an algorithm) and, ultimately, to the end-to-end quality characteristics of a data processing pipeline.
- A **quality constraint** is a logical expression involving and restricting quality parameters in order to define the validity of the actual execution, e.g., that the throughput of a certain data processing element (implying the underlying selected data processing algorithm) shall not be below a given number. Quality constraints will formalize quality (also known as non-functional) requirements collected, e.g., for a specific pipeline or an organization running the QualiMaster infrastructure. Specific categories of quality constraints deal with the cost of execution, the adaptation or the pipeline itself. On pipeline level, quality constraints will in particular define the SLAs of a data processing pipeline for both, source (pipeline input) and sink (pipeline output) side, i.e., the SLAs negotiated with the customer. Quality constraints bound the adaptation space, i.e., the violation of a quality constraints must be avoided (although they may have to be tolerable for a short period of time until the result of an adaptation can be enacted) and may be used as triggers for reflective adaptation or in order to indicate exceptional situations.

1.5 Structure of the Deliverable

The rest of this deliverable is structured into six sections. Section 2 introduces stakeholders or key actors interacting with the QualiMaster application and the different parts of the QualiMaster infrastructure. Section 3 is dedicated to the two QualiMaster application scenarios and describes the respective use cases. Section 4 provides an initial collection of non-functional requirements with respect to data, algorithms derived from the application scenarios and the QualiMaster platform. Sections 5 contains the system use cases for the QualiMaster infrastructure in terms of three subsections, each focussing on one of the key actors for the infrastructure: the Pipeline Designer, the Adaptation Manager, and the Platform Administrator, respectively. Finally, Section 6 presents some conclusions from this early requirements collection process and outlines the next steps in WP1.

2 Key Actors of the QualiMaster Infrastructure

An actor represents a group of users, who interact with a system and who have a similar view on a system. Actors may be persons, but also companies, organizations or even computer systems [5]. Actually, actors are stakeholders of the system, but not all (groups of) stakeholders are required to interact with the system. Further, a real person may take the role of different actors, e.g., depending on the organization structure of a company running the QualiMaster infrastructure. In this section, we will define the key actors of the QualiMaster infrastructure in terms of two distinct groups, namely application users in Section 2.1 and infrastructure users in Section 2.2. In Section 3 and 5, we will describe the use cases according to these groups of actors, respectively.

2.1 Application Users

The application users are the actors that interact with the financial applications to be built on top of the QualiMaster Infrastructure. They access the system with a task at hand (e.g. risk analysis of a certain market player) and use the QualiMaster applications to perform this task.

Application users do not need to know how the underlying QualiMaster platform is configured or developed. The respective financial application should, however, support some flexibility regarding the data analysis such as the selection of the market players to analyze, the time span to be considered in the analysis, etc. This functionality should also be supported through the graphical user interface of the application.

The most important actors for QualiMaster in the group of the Application users are:

- Hedge Fund Manager
- Investment Company
- Investment Bank
- Regulator

We will detail these actors in the table below:

Application Users	
Actor	**Hedge Fund Manager**
	A Hedge Fund Manager oversees and makes decisions about the investments in a hedge fund. To be successful, a hedge fund manager must consider how to gain a competitive advantage, a clearly defined investment strategy, adequate capitalization, a marketing and sales plan and a risk management strategy. QualiMaster strives to provide the Hedge Fund Manager with a tool to achieve this competitive advantage mainly in terms of systemic risk management. This can be used for portfolio optimization or risk management and hedging.
Actor	**Investment Company**
	An Investment Company is a corporation or trust engaged in the business of investing the pooled capital of investors in financial securities. This is most often done either through a closed-end mutual fund or an open-end mutual fund. The open-end fund must be willing to buy back shares from investors every business day. Exchange-traded funds (or "ETFs" for short) are open-end funds or unit investment trusts that trade on an exchange. Open-end funds are most common, but exchange-traded funds have been gaining in popularity. Closed-end funds generally issue shares to the public only once, when they are created through an initial public offering. Their shares are then listed for trading on a stock exchange. Investors who do no longer wish to invest in the fund cannot sell their shares back to the

	fund (as they can with an open-end fund). Instead, they must sell their shares to another investor in the market.
Actor	**Investment Bank**
	Management of enterprise wide risk across a wide range of asset types has become a major regulatory requirement in addition to being a prerequisite for effective capital allocation. The need to include a network of potential exposure outside of the investment bank has recently become recognised, in the sense that internal liquidity is no longer a sufficient indicator of financial stress. By providing an Investment Bank with a tool that identifies in real time the co-dependencies and avenues of contagion between major market participants, they will be better able to manage such exposures.
Actor	**Regulator**
	Historically, Regulators have been retrospective in their analysis of major systemic risk events. Increasingly, Regulators have access to real time exchange data and are looking to leverage this data to provide more timely, and ultimately proactive management of the financial system. The QualiMaster project will provide Regulators with a unique opportunity to view in real time a systemic risk network identify the sources and sinks of risk and view contagion through network topology.

2.2 Infrastructure Users

In contrast to application users, infrastructure users directly interact with the platform in order to define data analysis pipelines, the adaptation space of individual pipelines or to administer a platform (including its initial setup). To perform their tasks, infrastructure users utilize specific tools and, thus, have specific requirements towards the QualiMaster infrastructure.

The following types of infrastructure users have so far been identified in the QualiMaster project:

- Pipeline Designer
- Adaptation Manager
- Platform Administrator

We will detail these actors in the table below:

Infrastructure Users	
Actor	**Pipeline Designer**
	The Pipeline Designer defines the structure of data processing pipelines for performing specific analysis tasks. In particular, a Pipeline Designer identifies the data sources, the data sinks, the data processing elements (families) to be used in a pipeline and the data flow among the processing elements. The task of the pipeline designer may also include the selection of adequate visualizations for the pipeline processing results.
Actor	**Adaptation Manager**
	The Adaptation Manager defines and specifies the adaptive behaviour of the system. This includes defining the quality characteristics of the different data processing elements, the methods for measuring them and defining methods for estimating the end-to-end quality of pipelines. In addition, the Adaptation Manager also has to define a set of rules on the pipeline level for reactive and

	proactive adaptation as well as prediction mechanisms for quality parameter for proactive adaptation. Furthermore, the Adaptation Manager monitors and analyzes the execution of adaptation rules and reflects the results of the analysis by adjusting these rules when required to further optimize the adaptations (reflective adaptation).
Actor	**Platform Administrator**
	The Platform Administrator sets up, installs and maintains the QualiMaster infrastructure. This includes the administration of the physical computing resources, the algorithm and algorithm families pool, the reconfigurable hardware units (such as Data Flow Engine boards) as well as the storage of data. In addition, the Platform Administrator is also of in charge of monitoring the pipeline operation and of starting and stopping pipelines and, thus, taking the responsibility of the physical compute resources.

3 Application Domain & Business Use Cases

In this section, we discuss the use cases of the QualiMaster applications. In Section 3.1 we provide an overview on the applications. In Section 3.2, we detail the application use cases for systemic risk assessment for institutional financial clients and in Section 3.3 the application use cases for regulatory bodies. These use cases and the envisioned data analysis pipeline resulted from intensive discussions between the partners MAX and SPRING.

3.1 QualiMaster Applications

Five specific application domains have been identified, three of them belonging to the business domain “Risk assessment for institutional financial clients“ and two of them belonging to the domain “Systemic Risk Analysis for Regulatory Bodies“. The five application use cases are illustrated in Figure 2.

Basically, financial data, news and social media data ① will be considered as input streams (details will be given in Section 4). The instantiated QualiMaster infrastructure ② with specific data processing pipelines for the financial applications will process the input streams. The output of the processing will be prepared for the actual application by data analytics and visualization techniques ③. Finally, the five applications ④ will present the analysis results to the financial end users, i.e., the application user actors introduced in Section 2. We will describe the use cases of these five applications in the remainder of this section. As indicated above, the applications can be assigned to the two financial business domains ⑤.

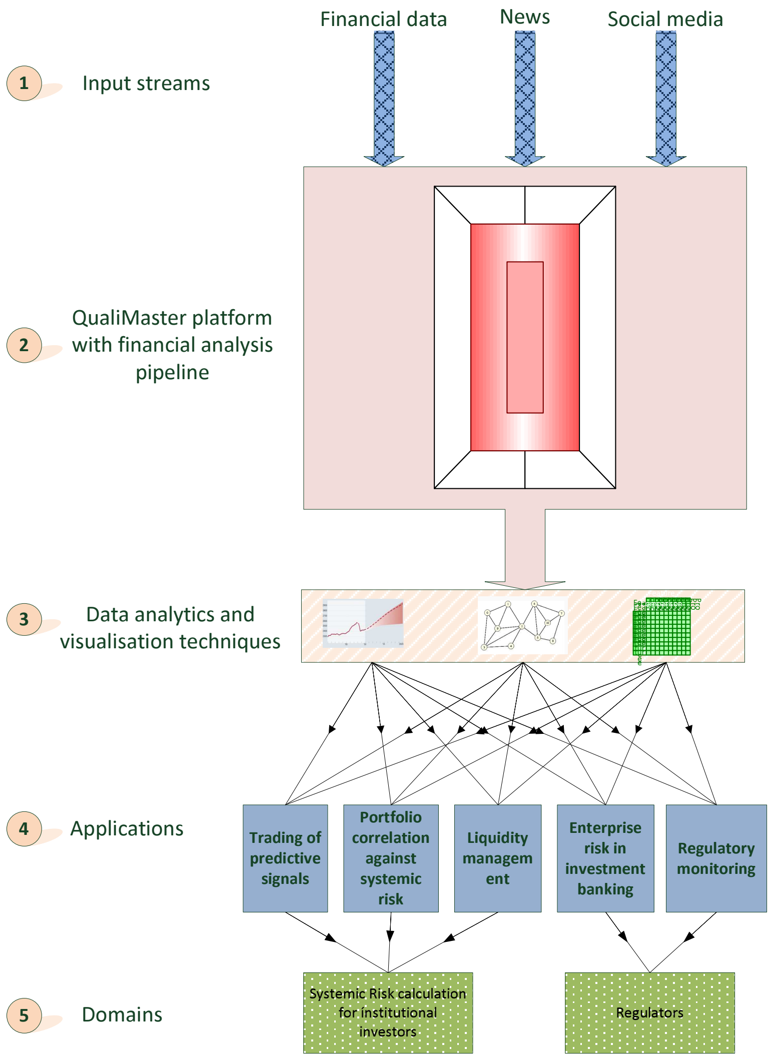

Figure 2: Information flow between QualiMaster infrastructure and applications

3.2 Application Use Cases for Systemic Risk Assessment for Institutional Financial Clients

SPRING is very active in the Hedge Fund industry, and has identified three main application use cases that will benefit from QualiMaster support and advanced risk analysis. In the following paragraphs, we first provide some background on the underlying business domain (Section 3.2.1), namely on the hedge fund industry and describe then the use cases for the three applications in the business domain of systemic risk assessment for institutional financial clients (Section 3.2.3).

3.2.1 The Hedge Fund Industry

A hedge fund is a pooled investment vehicle administered by a professional management firm, and often structured as a limited partnership, limited liability company, or similar vehicle. Many hedge fund investment strategies aim to achieve a positive return on investment regardless of whether markets are rising or falling ("absolute return"). Some hedge funds have several billion dollars of Assets Under Management (AUM). As of June 2013, the estimated size of the global hedge fund industry was US$ 2.4 trillion. As of February 2011, 61% of worldwide investment in hedge funds comes from institutional sources. In June 2011, the hedge funds with the greatest AUM was Bridgewater Associates (US$ 58.9 billion), Man Group (US$ 39.2 billion), Paulson & Co. (US$ 35.1 billion), Brevan Howard (US $31 billion), and Och-Ziff (US$ 29.4 billion). Bridgewater Associates, had $70 billion under management as of 1 March 2012.

Hedge funds employ a wide range of trading strategies but classifying them is difficult due to the rapidity with which they change and evolve. However, hedge fund strategies are generally said to fall into four main categories: global macro, directional, event-driven, and relative value (arbitrage). These four categories are distinguished by investment style and each have their own risk and return characteristics.

Because investments in hedge funds can add diversification to investment portfolios, investors may use them as a tool to reduce their overall portfolio risk exposures. Managers of hedge funds use particular trading strategies and instruments with the specific aim of reducing market risks to produce risk-adjusted returns, which are consistent with investors' desired level of risk. Hedge funds ideally produce returns relatively uncorrelated with market indices.

The total capital invested globally in hedge funds increased to a record level for the fourth consecutive quarter in Q2 2013, according to the latest HFR Global Hedge Fund Industry Report [30]. The total hedge fund capital increased by a net total of US$ 40 bn in 2Q13 to a record US$ 2.41 trn. The total number of hedge funds increased to over 10,000 funds for the first time since 2006. Positive capital inflows occurred across all fund sizes, with firms below US$ 500 m in AUM experiencing combined inflows of approximately US$ 2.4 bn. The industry's largest firms, those in excess of US$ 5 bn in AUM, experienced net inflows of US$ 6.1 bn, while firms between US$ 1 bn and US$ 5 bn experienced inflows of US$ 5.8 bn [22].

3.2.2 Use Cases for Institutional Financial Clients

In this section, we describe the use cases for institutional financial clients, namely for

- UC-TOPS1, UC-TOPS2: Application Trading of Predictive Signals
- UC-PCASR1: Application Portfolio Correlation against Systemic Risk
- UC-LM1, UC-LM2: Application liquidity management

3.2.2.1 Trading of Predictive Signals

In this setting, the actor (Hedge Fund Manager, Investment Company) is looking for new investment opportunities. From his own market analysis, he/she identifies one or more market players that seem to have good trading opportunities. We will call this specific application *Trading Of Predictive Signals Application* (TOPS App). In the first use case (UC-TOPS1), the actor checks co-dependencies against existing portfolio members. In the second use case (UC-TOPS2), the actor checks co-dependencies against all markets.

Use Case Identifier	**UC-TOPS1**
Use Case Name	Application Trading of predictive signals against existing portfolio
Actor	Hedge Fund Manager, Investment Company
Goal	Assist investment decision makers in selecting market players for new investments, taking into account the systemic risk
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	The actor is able to make decisions for modifying his portfolio members
Scenario Sequence	1. Actor starts the QualiMaster TOPS App, enters his login credentials and becomes an application user. 2. Depending on the role of the user, the TOPS App allows the user to select more or less detailed market segments and output visualizations. 3. The user selects ‘New market player’ and ‘existing portfolio member’, chooses analysis mode ‘Compare players’ visualization mode ‘Dependency table’. 4. The TOPS App through the QualiMaster infrastructure with running data analysis pipeline tells the user whether the new market player(s) have strong dependencies to the market players in which they have already invested in. This is based on correlation analysis of time series including real-time and historical data. 5. Based on this information, the user decides to invest in the new market player or not.
Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the TOPS App 3a Based on his role (senior or junior investment manager), the user is able to access/select more intermediate steps of analytics and can select specific pipeline paths for analysis. He can see on which input streams the application Trading of predictive signals is mainly based on. 3b Depth and complexity of data selection, filtering, data representation, and visualisation can be increased and decreased. For example, one additional factor in measuring risk, using predictive signals could be a statistical filtering process to find social media inputs that have proven a good track record in predicting correlation. 3c The user is able to select/de-select each input stream. This may be done by a user-triggered adaptation of the data analysis pipeline. For example, he deselects the impact of social media on the application trading of predictive signals.
Business Rules	
Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure

	d. Output visualizations, e.g. comparison and table visualization
Use Case Identifier	**UC-TOPS2**
Use Case Name	Application trading of predictive signals against all markets
Actor	Hedge Fund Manager, Investment Company
Goal	Assist investment decision makers in selecting market players for new investments, taking into account the systemic risk
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	The actor is able to make decisions for modifying his portfolio members
Scenario Sequence	1. Actor starts the TOPS App, enters his login credentials and becomes an application user. 2. Depending on the role of the user, the TOPS App allows the user to select more or less detailed market segments and output visualizations. 3. The user selects 'New market player' and 'all markets', chooses analysis mode 'Compare players' visualization mode 'Dependency table'. 4. The TOPS App through the QualiMaster infrastructure with running data analysis pipeline tells the user which dependencies the new market player(s) has against all market segments. The user can now check whether there are dependencies to market segments in which the investor would not like to invest in (for example emerging markets). This is based on correlation analysis of time series including real-time and historical data. 5. Based on this information, the user decides to invest in the new market player or not.
Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the TOPS App 3a Based on his role (senior or junior investment manager), the user is able to access/select more intermediate steps of analytics and can select specific pipeline paths for analysis. He can see on which input streams the application Trading of predictive signals is mainly based on. 3b Depth and complexity of data selection, filtering, data representation, and visualisation can be increased and decreased. For example, one additional factor in measuring risk, using predictive signals could be a statistical filtering process to find social media inputs that have proven a good track record in predicting correlation. 3c The user is able to select/de-select each input stream. This may be done by a user-triggered adaptation of the data analysis pipeline. For example, he deselects the impact of social media on the application trading of predictive signals.
Business Rules	

Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure d. Output visualizations, in particular comparison and table visualization

3.2.2.2 Portfolio Correlation Against Systemic Risk

The actor (Hedge Fund Manager, Investment Company) wants to check if his already existing portfolio is diversified in terms of systemic risk or not. The given (successful) use case scenario sequence assumes that the actor wants his portfolio to be as diversified as possible. We will call the application *Portfolio Correlation Against Systemic Risk Application* (PCASR App).

Use Case Identifier	**UC-PCASR1**
Use Case Name	Application portfolio correlation against systemic risk
Actor	Hedge Fund Manager, Investment Company
Goal	Assist investment decision makers in enhancing the diversification (against systemic risk) of their already existing investments
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	The actor is able to make decisions for modifying his portfolio members
Scenario Sequence	1. Actor starts the QualiMaster PCASR App, enters his login credentials and becomes an application user. 2. Depending on the role of the user, the PCASR App enables the user to select more or less detailed market segments and output visualizations. 3. The user selects all market players he/she has in his portfolio. In case a market player of his portfolio is not provided by the underlying QualiMaster data analysis pipeline, he/she selects a market index that represents his market player best. For analysis mode, he selects 'Compare members'. For visualization, he selects 'Cluster representation'. 4. The PCASR App provides a Cluster visualization and a corresponding numerical table. The user can now see how strong the co-dependency from each of his portfolios' market players against each other is. This is based on correlation analysis of time series including real-time and historical data. 5. In case of strong clustering and/or recognized dependency loops, the user can reduce position sizes or even close positions.
Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the PCASR App 3a Based on his role (senior or junior investment manager), the user is able to access/select more intermediate steps of analytics and can select specific pipeline paths for analysis. He can see on which input streams the portfolio correlation against systemic risk is mainly based on.

	3b Depth and complexity of data selection, filtering, data representation, and visualisation can be increased and decreased. For example, one additional factor in measuring portfolio correlation might be a co-dependency module that measures and visualizes market players that are often mentioned together in social media. 3c The user is able to select/de-select each input stream. This may be done by a user-triggered adaptation of the data analysis pipeline. For example, he deselects the impact of social media on the portfolio correlation against systemic risk analysis. 4a The PCASR App does not report significant co-dependencies. 5a Subsequent to 4a the user does not need to act.
Business Rules	
Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure d. Output visualizations, in particular cluster and table visualizations

3.2.2.3 Liquidity Management

The investor (Hedge Fund Manager, Investment Company) wants to check the systemic risk of the whole market and market segments of interest to modify his liquidity management. For example, in the case of high overall systemic risk, the investor will reduce market player position sizes and hold more liquidity. We will call the related application the *Liquidity Management Application* (LM App). In the first use case (UC-LM1), the actor checks general systemic risk on markets while in the second use case (UC-LM2), the actor checks systemic risk on market players he has invested in.

Use Case Identifier	**UC-LM1**
Use Case Name	General Systemic Risk Assessment for Markets
Actor	Hedge Fund Manager, Investment Company
Goal	Assist investment decision makers in modifying the liquidity of the investment pool with respect to the systemic risk
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	The system provides risk information so that the actor is able to make decisions for modifying his portfolio liquidity.
Scenario Sequence	1. Actor starts the QualiMaster LM App, enters his login credentials and becomes a user of the application 2. Depending on the role of the user, the LM App presents selection options for more or less detailed market segments and output visualizations 3. The user selects 'common major markets', chooses analysis mode 'Compare against portfolio' and chooses 'Current systemic risk', 'Tendency' and 'Prognosis' for output. Tendency and prognosis are based on the user selection of the time horizon, e.g., this moment, the last five minutes, the last hour,

	the last day, the last week, etc. 4. The LM App tells the user through the results of the data analysis pipeline running on the QualiMaster infrastructure, that the current systemic risk (based on real-time co-dependency data) is low, but the previous tendency was raising, also the prognosis says, there is a relevant chance, that risk will raise more. 5. Based on this information, the user decides to reduce the size of investment in general, raising the amount of available liquidity from 20% to 30%.
Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the LM App 3a Based on the role of the user (Senior or junior investment manager), the user is able to access/select more intermediate steps of analytics and can select specific pipeline paths for analysis. For example, he can see, on which input streams the risk analysis is mainly based on. The risk analysis of German blue chips is based on 80% price information, 15% social media streams and 5% News streams. 3b The user is able to select/de-select each input stream. This may be done by a user-triggered adaptation of the data analysis pipeline. For example, he/she deselects the impact of social media on the risk analysis. 4a The LM App does not report significant risks. 5a Subsequent to 4a the user does not need to take new decisions.
Business Rules	
Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure d. Output visualizations

Use Case Identifier	**UC-LM2**
Use Case Name	Specific Systemic Risk Assessment for Individual Market Players
Actor	Hedge Fund Manager, Investment Company
Goal	Assist investment decision makers in modifying the liquidity of the investment pool with respect to the systemic risk
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	The system provides risk information so that the actor is able to make decisions for modifying his portfolio liquidity.
Scenario Sequence	1. Actor starts the QualiMaster LM App, enters his login credentials and becomes a user of the application 2. Depending on the role of the user, the LM App presents

	selection options for more or less detailed market segments and output visualizations 3. The user selects market segment 'German Blue Chips', selects Market players 'BMW', 'Volkswagen' and Currency 'EURUSD' (The user has invested in the German automobile industry and he knows that the profits of those market players are strongly dependent of exports to North America. So he includes the currency rate in the risk analysis). 4. The LM App tells the user through the results of the data analysis pipeline running on the QualiMaster infrastructure that the systemic risk in the German main market is stable (using real-time co-dependency calculations). However, the risk for the automobile market player slightly raised based on a comparison of the current risk and historical risk data. The system also shows that there is a prognosis of raising currency rate risk. 5. As this may affect the automotive market players, the user decides to reduce the position size for those market players.
Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the LM App 3a Based on the role of the user (Senior or junior investment manager), the user is able to access/select more intermediate steps of analytics and can select specific pipeline paths for analysis. 3b The user is able to select/de-select each input stream. This may be done by a user-triggered adaptation of the data analysis pipeline. For example, he deselects the impact of social media on the risk analysis. 4a The LM App does not report significant risks. 5a Subsequent to 4a the user does not need to take new decisions.
Business Rules	
Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure d. Output visualizations

3.2.3 Application Use Cases for Systemic Risk Assessment for Regulatory Bodies

Since the credit crisis the environment for risk management within the Investment Banking (IB) sector, especially with regard to the regulatory response, has changed dramatically towards greater regulatory oversight together with significantly increased internal changes for improving risk management. Post Dodd-Frank in the US and European Market Infrastructure Regulation (EMIR) in the EU, IBs have been asked to provide the results of extensive stress tests designed to allow the regulators to assess the financial health of individual IBs. These stress tests can take many forms, from simple what-if scenarios (simple in definition but generally resource intensive to produce) to more complex tail risk analytics such as Value at Risk (VaR) or Comprehensive Risk Measure (CRM). Regulators use these stress tests to understand and specify capital requirements for IBs, which may apply bank wide or to specific trading activities.

The requirements for stress testing were certainly present, though in a reduced form, prior to the credit crisis. The difference today is not just the severity of the tests and their application, but more importantly the way in which cross-sector stress data is analysed. Prior to the credit crisis the focus was almost exclusively on the health of individual institutions. Little attempt was made to understand the nature of systemic risk transfer amongst IBs or to apply such understanding to the capital requirements of individual IBs. With the identification of Systemically Important Financial Institutions, regulators have acknowledged the importance of systemic risk and the need to reflect this in capital requirements.

Stress testing is an important component of identifying systemic risk in that it identifies the most vulnerable institutions. However, since stress tests generally take many months to complete, the information can rapidly become outdated as market events overtake prior results. Furthermore the nature of stress contagion or alternatively, the topology of systemic risk networks, cannot be inferred without further information. Therefore it is important both for regulators, and for IBs themselves, to have access to the current state of the financial network in order to guide decision making during periods of stress.

The use cases below identify how QualiMaster will enable both regulators and IBs to proactively manage and rapidly respond to stress events in the financial markets, providing a valuable tool for reducing both impact of events as they occur and insight into the nature of systemic risk to inform policy making.

3.2.3.1 Enterprise Risk in Investment Banking

The actor (Investment Bank) wishes to understand better the interdependencies in the market and protect against contagion. We will call the realizing application the Enterprise Risk in Investment Banking Application (ERIB App). In the first use case (UC-ERIB1), the actor checks co-dependencies against existing portfolio members. In the second use case (UC-ERIB2), the actor checks co-dependencies against all markets.

Use Case Identifier	**UC-ERIB1**
Use Case Name	Checking co-dependencies against existing portfolio members
Actor	Investment Bank
Goal	Assist in capital allocation and systemic risk management
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	The actor is able to make decisions for capital allocation and take measures to minimize systemic risk in the bank.
Scenario Sequence	1. The actor starts the QualiMaster ERIB App, enters his login credentials and becomes an application user. 2. Depending on the role of the user, the ERIB App enables the user to select more or less detailed market segments and output visualizations. 3. The user selects ‘New market player’ and ‘existing portfolio member’, chooses analysis mode ‘Compare players’ visualization mode ‘Dependency table’. 4. The ERIB App through the QualiMaster infrastructure with running data analysis pipeline tells the user whether the new market player(s) have strong dependencies to the MPs in which they have already invested in. 5. Based on this information, the user may modify assets/liabilities and take protection.

Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the ERIB App 3a The user may choose to focus on a specific market participant in order to understand their dependency network. Having understood the degree to which the market participant is exposed to contagion may decide to reduce exposure. 3b The user may monitor the systemic risk network topology via, for example, centrality metrics, to understand in real time the health of the financial network. 3c The user may choose to take the systemic risk network and overlay the investment banks exposures by market participant in order to derive an investment bank specific systemic risk network.
Business Rules	
Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure d. Output visualizations, in particular comparison and table visualization

Use Case Identifier	**UC-ERIB2**
Use Case Name	Check co-dependencies against all markets
Actor	Investment Bank
Goal	Assist in capital allocation and systemic risk management
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	The actor is able to make decisions for capital allocation and take measures to minimize systemic risk in the bank.
Scenario Sequence	1. The actor starts the QualiMaster ERIB App, enters his login credentials and becomes an application user. 2. Depending on the role of the user, the ERIB App enables the user to select more or less detailed market segments and output visualizations. 3. The user selects ‘New market player’ and ‘all markets’, chooses analysis mode ‘Compare players’ visualization mode ‘Dependency table’. 4. The ERIB App through the QualiMaster infrastructure with running data analysis pipeline tells the user which dependencies the new market player(s) has against all market segments. The user can now check whether there are dependencies to market segments in which the investor would not like to invest in (for example emerging markets). 5. Based on this information, the user decides to invest in the new market player or not.

Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the ERIB App 3a The user may choose to focus on a specific market participant in order to understand their dependency network. Having understood the degree to which the market participant is exposed to contagion may decide to reduce exposure. 3b The user may monitor the systemic risk network topology via, for example, centrality metrics, to understand in real time the health of the financial network. 3c The user may choose to take the systemic risk network and overlay the investment banks exposures by market participant in order to derive an investment bank specific systemic risk network.
Business Rules	
Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure d. Output visualizations, in particular comparison and table visualization

3.2.3.2 *Regulatory Monitoring*

Regulators have access to real time market data from sources such as exchanges. Due to the volume of data, they currently do not have a comprehensive view of the state of the financial market. QualiMaster will provide such a view in real time that will enable them to identify key drivers of contagion and focus any preemptive response. The related application allows the actor to monitor the systemic risk network for possible signs of increasing dependency or contagion. We will call this application the *Regulatory Monitoring Application* (RM App).

Use Case Identifier	**UC-RM**
Use Case Name	Regulatory Monitoring
Actor	Regulator
Goal	Provide regulator with real time comprehensive monitoring of the state of the financial network.
Precondition	QualiMaster platform is running (UC-PA12) and the QualiMaster data analysis pipeline is started (UC-PA8).
Postcondition	Actor is able better regulate the financial markets.
Scenario Sequence	1. The actor starts the QualiMaster RM App, enters his login credentials and becomes a user of the application. 2. Depending on the role of the user, the RM App allows the user to select more or less detailed market segments and output visualizations. 3. The user selects 'market participant' (node) to obtain detailed breakdown of information flow. 4. The RM App through the results of the configured data processing pipeline running on the QualiMaster infrastructure tells the user the market dependencies of the selected market

	participant. 5. Based on this information, the user decides to investigate the liquidity position of the market participant (via external sources). 6. The user monitors the centrality metric of systemic risk network. 7. The RM App displays the real time centrality metrics through the QualiMaster infrastructure. 8. The user responds to severe changes in centrality metrics as a signal for heightened monitoring.
Extensions	1a Login is not permitted due to invalid credentials and the actor is informed by the RM App 3a The user specifies pre-dependency analysis filters to input data in order to focus on specific asset types or regional segments. To perform efficient processing, this leads to a user-triggered adaptation of the data analysis pipeline. 3b The user utilizes social data to enhance market sentiment insight and capture breaking news. This may be done by a user-triggered adaptation of the data analysis pipeline.
Business Rules	
Data/Functions	a. User information, login and authentication functions b. Market segment information c. User triggers into the QualiMaster infrastructure d. Output visualizations, in particular centrality metric of systemic risk

Additional remarks

Some use cases indicate that the user provides feedback back to the QualiMaster infrastructure, which leads to a **user-triggered adaptation** of the data processing. However, this affects the costs of data processing. So two user models should be considered:

1) The user utilizes the QualiMaster infrastructure and the running pipeline ‘as it is’, without the possibility of pipeline modification. This would enable lower usage costs.
2) The user can send triggers to adapt the pipeline processing at runtime. This will result in higher and dynamically changing costs for using the system. Please note, that such an adaptation enables changes to a running data processing pipeline within boundaries given by the pipeline design in terms of structure and quality constraints. Such triggers do not imply the ability to modify or specify a new pipeline. This is a task of the Pipeline Designer.

4 Data, Algorithm and Quality Requirements

In this section, we discuss the requirements that arise from the data and the algorithms that will be used by the QualiMaster applications as introduced in Section 3.1. First, in Section 4.1 we discuss the requirements collected so far for the data (sources). Then, in Section 4.2 we present the requirements for the algorithms to be applied. Finally, in Section 4.3 we will discuss initial quality tradeoffs and derive quality requirements for the QualiMaster platform as a basis for the adaptive execution of pipelines. All sections start with a discussion on the background and conclude with initial requirements. The requirements will be marked with a unique identifier for further and future reference and will be given in a simplified form of controlled natural language, which is frequently used in requirements engineering to avoid ambiguities (e.g. [2, 6, 11]).

4.1 Data and Data Stream Requirements

The QualiMaster infrastructure must support various data sources with different characters and specifications. The requirements originating from these data sources are discussed in the following paragraphs.

4.1.1 Types of Data Sources

As introduced in Section 3.1, the QualiMaster applications will be built on two types of data sources. Data from the **financial domain** is the main source. With respect to this, QualiMaster must handle data from foreign exchange rates (currently around 140 currency pairs), futures on indices and commodities, bond markets, stocks from worldwide exchanges and market indices. Additional sources for financial data that might be also incorporated in QualiMaster are international interest rates. The financial data is provided by SPRING through a specific API to the project partners.

The second QualiMaster data source is **Web data**. This is a collection of contributions from individual sources on the Web (e.g., posts about products, stocks, companies, real state, jobs) as well as experts from the financial domain (e.g., posts about recent expert analysis, studies, job reports, predictions) in micro-blogging systems such as Twitter. In particular, LUH is collecting Twitter data from the public sample stream via its REST API [28]. The API returns a random sample of 1% of all public tweets. LUH is using several parallel streams to increase the amount of collected tweets. In order to increase the amount of data related to the financial domain, additional focused streams will be collected using the public filter stream API [28]. For example:

- Streams filtered by a static set of financial terms using relevant and general terms such as “financial market”, “stocks”, “banks”, “Dow Jones” etc, that are time independent. The filter terms will be defined in the configuration and may need manual reconfiguration over time or adjustments by user triggers.
- Streams filtered by a dynamic set of financial terms using a set of terms that are related to current events. The terms shall be extracted from current news, and updated continuously.
- Streams of news agencies and users that are identified as “experts” in the financial domain.

While the financial data is typically composed by numbers, the Web data is composed by more complex data types. For instance, a tweet from Twitter contains the actual text and a lot of metadata [29]. The metadata contains fields related to the tweet such as creation time, hashtags used, URLs included in the tweet and whether tweet is a retweet or not and the geo coordinates or location of the user when sending the tweet. In addition, the metadata includes fields related to the user, such as number of followers, friends, tweets, location, time zone and many more.

Furthermore, QualiMaster will collect and analyze information from online news that can have an impact on the financial market, such as political news (e.g., new elections, armed conflicts), science and technology (e.g., trends in technology that may affect some industrial sectors), and news on natural disasters (e.g., major earth quakes, tsunamis etc).

While the financial data is typically structured and composed by numbers, the Web data is less structured, heterogeneous and composed by more complex data types. In addition to the more complex format, Web data can have noise, incorrect values, may be biased and the meaning or the actual interpretation is typically uncertain (since the messages are expressed in evolving natural languages). QualiMaster must be able to handle the different data formats of the incorporated data sources.

4.1.2 Real-time vs. Static Sources

In addition to the real-time data, the QualiMaster infrastructure will incorporate static data sources. This is historical financial data, collected for up to 20 years, depending on the availability of market players (available through SPRING). The historical data of QualiMaster has the same format and characteristics as the real-time data and covers most market players. QualiMaster needs to be able to batch process the past data to provide historical time series of the systemic risk. The collection of Twitter data from the public sample stream by LUH started already in February 2013. There have been 3.7 billion tweets (~2.22 TByte) collected till end of March 2014, which can be used as historical data In order to increase the amount of data related to the financial domain, additional focused streams shall be collected. This includes the Web data stream examples descried in Section 4.1.1.Data Stream Characteristics

We will detail the characteristics of the data streams in terms of three dimensions, the data load, the message rate and the volume.

Data Load (i.e., number of streams)

QualiMaster must be able to handle various data streams arriving at the same time. For instance, there will be more than 100 pairs of foreign exchange rates, around 500 virtual streams for the futures, indices and bonds from one source. Information coming from international stock exchanges will provide some thousand stocks, which implies the same number of virtual streams.

Rate of Messages (i.e., messages per second arriving at QualiMaster)

In average market situations, an individual stock causes around 400 messages per second (see, for example, the Microsoft stock at NASDAQ with estimated 500.000 trade counts per day). According to the experience of the industrial partners in QualiMaster, the number of messages may grow by a factor of around 20 in times of larger market movements, news or other impacts.

In addition to the messages coming from the financial data, QualiMaster must be able to handle the messages coming from the Web data, including News articles and tweets. The News articles are around 400 per day. The tweets collected by LUH via Twitter's public sample stream (1% of all tweets) have an average rate of 300 per second. Obviously, this can increase the total rate of tweets which, as listed in the official statement of 2011, has an average rate of 4600 per second.

Volume of Messages (i.e., messages per second arriving at QualiMaster)

A larger exchange of about 2000-3000 stocks (e.g., NASDAQ) produces about 4 million messages per second. This results in a volume of about 10 GByte data per day, which QualiMaster must be able to handle. In addition, there will be around 5 GByte tweets that are collected and provided to QualiMaster by LUH via Twitter's public sample [28]. This corresponds to 1-3% of all tweets (this is the union of all sampled tweets that are collected from three parallel streams, each receiving 1% of all public tweets).

4.1.3 Initial Requirements

From the data stream background introduced by the sections above, we summarize the following requirements in the style of controlled natural language (possible with additional information in natural language).

- REQ-DS1: The QualiMaster infrastructure must support multiple real-time data sources, possible with each source having a different type of data. *Information:* In particular, this includes structured financial stock market data and social Web data.

- REQ-DS2: The QualiMaster infrastructure must support filtering of data according to criteria. *Information:* Filtering is at least defined through the configuration and may be influenced by user triggers.
- REQ-DS3: The QualiMaster infrastructure must support the integration of historical data sources and data processing. *Information:* This also includes queries over historical data.
- REQ-DS4: The QualiMaster infrastructure must support at least 400 stock market messages per second per market player under normal load. *Information:* We plan for 1.500 market players, i.e., 600.000 stock market messages per second.
- REQ-DS5: The QualiMaster infrastructure must support growth rates up to factor 20 over normal load for stock market streams.
- REQ-DS6: The QualiMaster infrastructure must support processing at least 1% of all public tweets from Twitter.
- REQ-DS7: The QualiMaster infrastructure must support growth rates up to 10% of all public tweets from Twitter.
- REQ-DS8: The QualiMaster must support processing online public news and financial blogs.

4.2 Requirements for Supported Algorithms

QualiMaster will implement algorithms designed to identify co-dependency and causality in multivariate time series. Co-dependency describes the degree to which time series tend to move together, typically captured by correlation. Co-dependency measures are symmetric. Causality describes the degree to which a time series is influenced by the prior behavior of another time series [14]. There are a number of different approaches to Causality, most of which utilize the concept of Information or Entropy. Since we are interested in contagion effects we will focus primarily on causality, although codependence will be useful as a way to quickly identify possible causal relationships.

The co-dependency or causality relationships between major market participants will be inferred from steaming market and social data, suitably filtered and normalized. Note that data of different market participants will arrive asynchronously. Classical time series analysis assumes that a multivariate vector of values may be sampled at regular time points. Where possible, QualiMaster will adapt such algorithms to allow incremental update of metrics suitable for asynchronous data.

A systemic risk network [8] quantifies the linkage between major market participants and provides a framework for identifying instability in financial markets. Major changes in the topology of the network (e.g., a sudden increase in the risk of contagion [1]) can signal ongoing financial stress. For co-dependency measures undirected, weighted networks will describe the degree of risk clustering. For causal measures directed, weighted networks will describe the major sources and flow of information.

The computation of information flow is in general significantly more expensive than correlation. Therefore, an initial correlation analysis will be done for a broad range of market participants to reduce dimension to subset of interest prior to the causality analysis.

We now give an overview of the main codependence and causality algorithms to be incorporated in QualiMaster.

4.2.1 Correlation

Pearson's correlation is a measure of the linear dependence between two variables. In the case of two time series X_t and Y_t, it is a measure of the extent to which a movement of X happens simultaneously with a proportional movement of Y. The correlation is a value between -1 and 1. If this value is 1, respectively -1, then a movement of X in one direction happens at the same time as a proportional movement of Y in the same, respectively the opposite direction. At the other extreme, a correlation of 0 means that there is no linear relation between the movements of X and

Y. Note the importance of the word linear here since independent variables will have correlations of 0, but the reverse is not true.

The classical approach to statistical correlation estimation assumes that the time series X_t and Y_t update synchronously. This is not the case in general for financial time series so that a feasible method to correlation in the financial domain needs synchronise the time series. The choice of the method is important since a poor choice can lead to biased results depending on the relative frequency of data points in the time series. Different approaches to synchronisation will be tested including interpolation (e.g., based on the last value) and Fourier correlation (see [21] for a survey).

An alternative to the classical correlation statistics first introduced in [13] will also be implemented. Here, an asynchronous estimator for the correlation of times series is defined based on the degree of inter-event overlap. The algorithm admits an inline version based on a state machine representation.

Pros

Correlations can be computed quickly compared to other methods. Computing correlations is an industry standard technique that is widely known and well accepted. Further, correlations have been used efficiently in more advanced studies such as Correlation Networks (see [19] for an example, or [3] for a survey).

Cons

Pearson's correlation only exposes linear relations between time series and may fail to properly represent non-linear co-dependencies in time series data. In particular, Pearson's correlation is not invariant under monotonic transformations of the marginal distributions. Alternatives to Pearson's correlation, such as Spearson's or the more general class of Rank Correlations mitigate this somewhat. More importantly, correlation metrics do not infer directionality or cause-and-effect relationship, which is a key feature of a systemic risk network.

4.2.2 Granger Causality

Given two processes X_t and Y_t, it is said that *Y* Granger-causes (G-causes) *X* if a model that incorporates past values of X_t as well as Y_t is better at predicting X_t than the same model, but without the past values of Y_t [12]. More precisely, consider a model

$$X_t = \sum_{i=1..n} A_{X,i} X_{t-i} + \sum_{i=1..n} A_{Y,i}\ Y_{t-i} + E_t \qquad (1)$$

In formula 1, the coefficients $A_{X,i}$ and $A_{Y,i}$ are computed using linear regression, and the term E_t represents the error between the model and the actual observed values. Then compare the standard deviation of E_t with the standard deviation of the error term of the same model without the terms in Y_{t-i}. If the standard deviation of the former is lower than that of the latter, then *Y* is said to G-cause *X*. Statistical significance tests can be applied to decide if the causality should be accepted or rejected as not significant. A survey with emphasis on computational aspects is available in [18].

Pros

G-causality is a well established method with a long history of applications in economy and other subjects. One obvious advantage of G-causality is that it creates a direction: G-causality can be used to show that some processes drive the behaviour of other processes. However, this is not in general a one-way 'causality': Given two processes *X* and *Y*, one can find that *X* G-causes *Y*, but also that *Y* G-causes *X*.

In some work, G-causality has been extended, for example to adapt it to point processes [16]. A marked point process is the natural representation of the asynchronous market and social data streams.

Cons

Akin to correlation, G-causality only exposes linear relations. There are also extensions of G-causality to take non-linear features into account (see for example [9]), but these approaches are less well understood. Further, G-causality is computationally more expensive than the calculation of correlations.

4.2.3 Transfer Entropy

Information entropy, or simply entropy for short, was defined by Shannon [27] and is a measure of the uncertainty of a random variable or equivalently the average number of bits described by the random variable. The higher the entropy, the more uncertain it is or the more information is obtained on average by sampling the random variable. More formally, given a random variable *X*, its entropy is the average number of bits necessary to represent an outcome of *X*. The precise formula is

$$H_X = -\sum_x p(x) log_2 p(x) \tag{2}$$

In formula 2, H_X denotes the entropy of *X* and *p(x)* is the probability of a particular outcome *x* of *X*.

Given two time series X_t and Y_t, one can ask the question "How much information is encoded in X_t if we assume that we already know past values X_s and Y_s for $s<t$?" Another way of asking this question is "How many bits on average are necessary to encode an outcome of X_t if we assume that we have already encoded past values of X_t and Y_t". One example where the answer to the above question is 0 could be if X_t is always equal to Y_{t-1}. In general, this value will not be 0.

Assume that both, *X* and *Y,* are Markov processes, then the transfer entropy [15] from Y to X is defined by:

$$T_{Y\to X} = \sum_{x,y} p(x_t, x_{t-1}, y_{t-1}) log_2 (\frac{p(x_t | x_{t-1}, y_{t-1})}{p(x_t | x_{t-1})}) \tag{3}$$

Formula 3 can be extended to look at a set of past values of *X*, not just at x_{t-1}. One can also increase the number of past values for *Y*. The computation of transfer entropy requires estimating the joint distribution of (x_t, x_{t-1}, y_{t-1}) which may be achieved through binning or kernel estimation, amongst other methods [14].

Pros

Information entropy is not symmetric, i.e., $T_{Y->X}$ must not necessarily be equal to $T_{X->Y.}$ It can be used to distinguish between driving and responding elements [25]. For that purpose, transfer entropy is useful when creating directed networks or market players and seeking for the market players that are driving market movements. In that respect, transfer entropy is similar to G-causality. Its advantage over G-causality is that it is sensitive to non-linear signal properties [26].

Cons

Transfer entropy is more computationally intensive than correlation. It also requires substantially more data than G-causality to provide statistically significant results. Different techniques have been suggested to improve performance [14].

4.2.4 Initial Requirements

From the algorithmic background introduced by the sections above, we summarize the following requirements in the style of controlled natural language (possible with additional information in natural language).

- REQ-ALG1: The instantiated QualiMaster infrastructure for systemic risk analysis must support the calculation correlation for synchronized time series data streams. *Information:* At least interpolation and Fourier correlation will be considered.
- REQ-ALG2: The instantiated QualiMaster infrastructure for systemic risk analysis must support the calculation of correlation networks based on time series data streams.

- REQ-ALG3: The instantiated QualiMaster infrastructure for systemic risk analysis must support the calculation of Granger causalities on point processes of time series data streams.
- REQ-ALG4: The instantiated QualiMaster infrastructure for systemic risk analysis must support the calculation of transfer entropy for time series data streams. *Information:* Techniques for improving the performance will be applied.

Alternative ways of implementing the algorithms, e.g., those mentioned in REQ-ALG1, will form the basis for respective algorithm families. Furthermore, the algorithms will be considered for implementation in reconfigurable hardware in workpackage 3, i.e., depending on the approach for translating algorithms to hardware, appropriate algorithms will be chosen for demonstration and experimentation.

4.3 Platform Quality Requirements

Based on the sections before, we present now general quality requirements for the QualiMaster platform. In Section 4.3.1, we start with a discussion of quality dimensions. In Section 4.3.2, we summarize initial requirements drawn from the discussion of the quality dimensions.

4.3.1 Quality Dimensions

In this section, we discuss basic quality dimensions such as timeliness, coverage, accuracy, efficiency (performance) and resource consumption in the context of the applications and the QualiMaster infrastructure.

Timeliness

As described in Section 4.1, QualiMaster must enable the processing of real-time data streams to produce up-to-date analysis results in addition to historical data (REQ-DS6). This requires the capability to deal with high velocity data streams such as the financial tick data from the stock markets (REQ-DS1). By combining both, real-time and historical data, QualiMaster will be able to produce trend analysis and predictions of future development in the financial market. Real-time processing of large volume data with high velocity is typically computationally expensive (in terms of CPU and memory consumption). Therefore, means must be provided to express preferences in terms of timeliness vs. computational cost.

Coverage

The coverage of the data sources to be processed in QualiMaster shall be maximized to produce a comprehensive market analysis. However, there is a trade-off between the volume of data being processed and the computational cost. Processing more data will probably consume more computational resources. On the other hand, there is a trade-off between the volume of the data to be processed and efficiency, within a given set of resources consumption constraints (more data require more time to be processed). Here, user triggers from the applications may enable the user to influence the calculation and to express preferences in terms of coverage vs. performance.

Accuracy

The accuracy of prediction models vary depending on several factors such as the characteristics of the data and the underlying statistical and machine-learning models. Using more historical data and more accurate models typically is associated with some additional computational cost in terms of resources consumption and computation time. While increasing the accuracy of the applied model is desired, it is expected that the timeliness and performance of the analysis can be negatively affected. Therefore, the QualiMaster infrastructure must allow for balancing between these quality parameters.

Computational Performance

To achieve timeliness but also to satisfy the user, it is required to minimize the overall computation time of the analysis tasks. This may be achieved by parallelizing algorithmic tasks as well as in software-based algorithms as well as on reconfigurable computing (e.g., using multiple DFE boards in parallel). Further, workpackage 3 will analyze the required algorithms for translating them into hardware and to eliminate potential bottlenecks of software-based execution (as this was done by MAX in several application settings in the past). Generally, there is a trade-off between the efficiency and the timeliness, coverage and accuracy properties, which should be taken into account when applying adaptation tasks.

Resources Consumption

To enable further pipelines, value added computing or elastic resources, it is required to minimize the overall consumption of computational resources as much as possible. Given a set of constraints on the computational resources (CPU and memory) the system should be able to maximize the quality of the produced analysis and/or satisfy domain or user defined quality requirements (in terms of accuracy, coverage and timeliness) as much as possible.

4.3.2 Initial Requirements

In this section, we summarize the discussion above in terms of initial (dedicated) quality requirements, which give a first indication for the tradeoffs to be specified and handled by the adaptivity in QualiMaster. Preferences among the individual quality dimensions will be specified in terms of pipeline or adaptation constraints as well as adaptation rules (see use cases in Section 5).

- REQ-Q1: The QualiMaster infrastructure must support timeliness in the processing of real-time data streams in order to produce up-to-date analysis results.
- REQ-Q2: The QualiMaster infrastructure must support means to customize the coverage of the data sources to produce a comprehensive market analysis. *Information:* Preferences of coverage vs. performance may be given in terms of user triggers (see also REQ-DS2).
- REQ-Q3: The QualiMaster infrastructure must support means to specify the accuracy of the performed calculation.
- REQ-Q4: The QualiMaster infrastructure must support dynamic means to exploit mechanisms to maximize computational performance. *Information:* The flexible integration of hardware-based computing will enable the adaptivity to dynamically exploit the benefits of reconfigurable computing where applicable.
- REQ-Q5: The QualiMaster infrastructure must provide means to measure and optimize its resource usage.

5 System Requirements and Use Cases

In this section, we describe the identified system use cases from the point of view of an infrastructure use. Those use cases are structured along the three types of infrastructure users, which have been identified as actors in the context of the QualiMaster Infrastructure. Conceptually, we will name a configuration tool for each of the actors, e.g., a Pipeline Configuration Tool for the Pipeline Designer. Thus, an actor implicitly has access to "his/her" tool so that we do not detail user management aspects such as logging in. However, an implementation may realize the conceptually separated tools also in terms of one integrated configuration tool, which then might require some form of user management. Further, we will use the term repository for mechanisms storing and retrieving configuration data for such as the Pipeline Repository for all information directly related to pipelines, e.g., the data flow.

5.1 Pipeline Designer

As defined in Section 2.2, the Pipeline Designer creates a data processing pipeline by combining data processing elements. For this purpose, the following three use cases have been identified for the Pipeline Designer:

- UC-PD1: Define new pipeline
- UC-PD2: Modify Pipeline definition
- UC-PD3: Delete Pipeline definition

Some elements of the pipeline may be implemented in terms of dedicated hardware, e.g., a FPGA-based processor, such as a Maxeler dataflow supercomputer. These elements will be possible to be included as needed in the processing but need configuration by the Platform Administrator (see UC-PA5, UC-PA6, UC-PA11).

5.1.1 Use Case: Define New Pipeline

This use case enables the pipeline designer to define a new pipeline based on underlying technical configuration parts using the QualiMaster Pipeline Configuration Tool (PC tool). The definition of a pipeline includes the validation of syntax, semantics and feasibility of the pipeline and finally storing the new pipeline configuration.

Use Case Identifier	**UC-PD1**
Use Case Name	Define new pipeline
Actor	Pipeline Designer
Goal	Define a data stream analysis pipeline based on underlying configuration information, existing data processing elements and static quality validation.
Precondition	QualiMaster platform is configured (UC-PA10, optionally UC-PA11), data sources and sinks are configured (UC-PA7), quality characteristics are defined (UC-AM1)
Postcondition	New pipeline is validated and successfully stored
Scenario Sequence	1. The Pipeline Designer starts the QualiMaster PC tool and selects the definition of a new pipeline. 2. The PC tool shows an editor to enter the new pipeline. Available parts of a pipeline in particular sources, sinks and progressing elements (algorithm families) are shown to simplify the creation of a pipeline. 3. The Pipeline Designer enters the pipeline by selecting processing elements, defining the data flow between sources

	and data processing elements, among data processing elements, and, finally, from data processing elements to sinks. 4. The PC tool checks the syntactic and semantic validity of the pipeline (as far as possible interleaved with step 3). This includes, whether successors of processing elements can be linked, whether sources are connected and paths to sinks are present. 5. The Pipeline Designer adds quality constraints, such as SLAs for sources and sinks, constraints on the output quality of the processing elements or constraints on the connecting data flows. 6. The PC tool checks the syntactic and semantic validity of the quality constraints (as far as possible interleaved with step 5). 7. The Pipeline Designer initiates a static analysis of the feasibility of the pipeline, i.e., whether quality constraints can be met and whether the infrastructure is basically feasible for executing the pipeline (e.g., based on the actual or the maximum available resources). 8. The PC tool performs the static analysis of the end-to-end pipeline quality and whether the underlying pipeline infrastructure is basically capable of executing the configured pipeline. 9. The Pipeline Designer stores the configured pipeline (using a symbolic name for the pipeline design) into the pipeline repository of the infrastructure. 10. The PC tool acknowledges the successfully stored pipeline.
Extensions	3a Optional: The Pipeline Designer may access the configuration of other pipelines in order to reuse existing parts. 4a In case of syntactic or semantic errors in the constraint syntax, the PC tool displays appropriate messages in human readable form and highlights the involved elements. The use case continues at step 3. 6a In case of syntactic or semantic errors in the constraint syntax, the PC tool displays appropriate messages in human readable form and highlights the involved elements. The use case continues at step 5. 8a In case of an infeasible infrastructure, the PC tool indicates missing resources and suggests the increase of resources. The use case continues at step 3, 5 or 7. 8b In case that overall quality constraints cannot be fulfilled, the PC tool highlights critical parts or critical data flows. The use case continues at step 3, 5 or 7. 10a In case of a syntactically, semantically or not validated pipeline, the PC tool informs the Pipeline Designer about the actual status and stores the draft pipeline for further configuration. 10b In case of a physical storage error, the PC tool informs the Pipeline Designer about the failed pipeline repository action.
Business Rules	• Invalid pipelines cannot be executed on the QualiMaster

	platform. • Pipeline configurations which exceed the actual resources of the platform and the underlying hardware cannot be executed.
Data/Functions	a. Access to sources and sinks b. Access to quality parameter c. Syntactic pipeline analysis d. Semantic pipeline analysis e. Static pipeline quality analysis f. Access the pipeline repository g. Structural display (or visualization) of pipelines including editor

Additional remarks:

Currently, we see the technical information about the pipeline in the responsibility of the pipeline administrator (at the moment the technical information about sources and sinks such as IP addresses, credentials, the adapter etc) in order to separate concerns between pipeline design and its technical realization.

Actually, the PC Tool may be a graphical or a textual tool. WP4 envisions a textual Domain Specific Language (DSL) in a syntax-driven content-assisted editor, possibly with a side-by-side visualization of the pipeline. Thus, syntactic pipeline analysis will be done by the DSL infrastructure based on the grammar of the DSL. Semantic pipeline analysis includes type checking and the translation of the pipeline description into IVML, the INDENICA variability modeling language [7, 24]. The resulting IVML model then used for instantiating the QualiMaster platform / the pipeline utilizing the SPLE tooling provided by SUH.

In practice, steps 3-6 of the use case scenario may happen interleaved and in an incremental fashion. In the use case scenario, we provided these in a conceptually separated form. In fact, closing the PC tool without storing the pipeline will lead to a warning.

5.1.2 Use Case: Modify Pipeline Definition

This use case enables the pipeline designer to modify an existing pipeline definition based on underlying technical configuration parts using the PC tool. Akin to the definition of a new pipeline (UC-PD1), the modification of a pipeline definition requires the validation of syntax, semantics and feasibility of the pipeline and, finally, storing the new pipeline configuration.

Use Case Identifier	**UC-PD2**
Use Case Name	Modify pipeline definition
Actor	Pipeline Designer
Goal	Modify the definition of an existing data stream analysis pipeline based on underlying configuration information, existing data processing elements and static quality validation.
Precondition	QualiMaster platform is configured (UC-PA10, optionally UC-PA11), data sources and sinks are configured (UC-PA7), quality characteristics are defined (UC-AM1)
Postcondition	Existing pipeline definition is modified, validated and successfully stored
Scenario Sequence	1. The Pipeline Designer starts the QualiMaster PC tool, selects the modification of an existing pipeline and specifies which of

	the pipelines stored in the pipeline repository shall be modified. 2. The PC tool shows an editor displaying the pipeline selected in Step 1. Available parts of a pipeline in particular sources, sinks and progressing elements (algorithm families) are shown to simplify the modification of the pipeline. 3. The Pipeline Designer modifies the pipeline adding, changing or removing processing elements and by (re)defining the data flow between sources, sinks and data processing elements. 4. The PC tool checks the syntactic and semantic validity of the pipeline (as far as possible interleaved with step 3). This includes, whether successors of processing elements can be linked, whether sources are connected and paths to sinks are present. 5. The Pipeline Designer adds, removes or changes quality constraints such as SLAs for sources and sinks, constraints on the output quality of the processing elements or constraints on the connecting data flows. 6. The PC tool checks the syntactic and semantic validity of the quality constraints (as far as possible interleaved with step 5). 7. The Pipeline Designer initiates a static analysis of the feasibility of the pipeline, i.e., whether quality constraints can be met and whether the infrastructure is basically feasible for executing the pipeline (e.g., based on the actual or the maximum available resources). 8. The PC tool performs the static analysis of the end-to-end pipeline quality and whether the underlying pipeline infrastructure is basically capable of executing the configured pipeline. 9. The Pipeline Designer stores the modified pipeline into the pipeline repository of the infrastructure. 10. The PC tool asks the Pipeline designer whether the existing pipeline definition shall be overwritten. 11. The Pipeline Designer acknowledges that the existing pipeline shall be overwritten. 12. The PC tool acknowledges the successfully stored pipeline.
Extensions	1a No pipeline definitions are available for modification, i.e., the PC tool will not show pipelines for selection and the scenario ends at step 1. 3a Optional: The Pipeline Designer may access the configuration of other pipelines in order to reuse existing parts. 4a In case of syntactic or semantic errors in the constraint syntax, the PC tool displays appropriate messages in human readable form and highlights the involved elements. The use case continues at step 3. 6a In case of syntactic or semantic errors in the constraint syntax, the PC tool displays appropriate messages in human readable form and highlights the involved elements. The use case continues at step 5.

	8a In case of an infeasible infrastructure, the PC tool indicates missing resources and suggests the increase of resources. The use case continues at step 3, 5 or 7. 8b In case that overall quality constraints cannot be fulfilled, the PC tool highlights critical parts or critical data flows. The use case continues at step 3, 5 or 7 11a The Pipeline Designer does not acknowledge that the existing pipeline shall be overwritten. In this case, the scenario may continue at step 3, 5, 7 or 9. 12a In case of a syntactically, semantically or not validated pipeline, the PC tool informs the Pipeline Designer about the actual status and stores the pipeline for further configuration. 12b In case of a physical storage error, the PC tool informs the Pipeline Designer about the failed pipeline repository action.
Business Rules	• Invalid pipelines cannot be executed on the QualiMaster platform. • Pipeline configurations which exceed the actual resources of the platform and the underlying hardware cannot be executed.
Data/Functions	a. Access to sources and sinks b. Access to quality parameter c. Syntactic pipeline analysis d. Semantic pipeline analysis e. Static pipeline quality analysis f. Read and write access to the pipeline repository g. Access to the pipeline repository h. Structural display (or visualization) of pipelines including editor

Additional remarks:

In practice, steps 3-6 of the use case scenario may happen interleaved and in an incremental fashion. In the use case scenario, we provided these in a conceptually separated form. In fact, closing the PC tool without storing the pipeline will lead to a warning. Further, step 9 may also be used for storing the modified pipeline as a new pipeline. Basically, this would correspond to the definition of a new pipeline (UC-PD1).

5.1.3 Use Case: Delete Pipeline Definition

This use case enables the pipeline designer to delete an existing pipeline. Please note that this step just deletes the definition of the pipeline so that it cannot be (re)instantiated and schedules the instantiated artefacts of the pipeline for (eventual) physical deletion by the Platform Administrator. Further, stopping a running pipeline is a task of the Pipeline Administrator.

Use Case Identifier	**UC-PD3**
Use Case Name	Delete pipeline definition
Actor	Pipeline Designer
Goal	Delete an existing data stream analysis pipeline.
Precondition	QualiMaster platform is configured (UC-PA10, optionally UC-PA11),

	data sources and sinks are configured (UC-PA7), quality characteristics are defined (UC-AM1)
Postcondition	Existing pipeline definition is deleted from the pipeline repository
Scenario Sequence	1. The Pipeline Designer starts the QualiMaster PC tool, selects the option *deletion of an existing pipeline definition* and specifies which of the pipelines stored in the pipeline repository shall be deleted. 2. The PC tool asks the Pipeline Designer whether the pipeline selected in Step 1 shall be deleted. 3. The Pipeline Designer acknowledges the deletion of the pipeline selected in Step 1. 4. The PC tool deletes the successful deletion of the pipeline selected in Step 1.
Extensions	1a. No pipeline definitions are available for deletion, i.e., the PC tool will not show pipelines for selection and the scenario ends at step 1. 3a The Pipeline Designer does not acknowledge that the existing pipeline shall be deleted. In this case, the scenario continues at the pipeline selection of step 1. 4a. In case of an access error, the PC tool informs the Pipeline Designer about the failed deletion.
Business Rules	• Actual pipeline execution requires responsibility about physical compute resources and is, thus, a responsibility of the Platform Administrator.
Data/Functions	a. Access to the pipeline repository b. Structural display (or visualization) of pipelines

5.2 *Adaptation Manager*

As introduced in Section 2.2, the role of the adaptation manager is to define and specify the adaptive behaviour of the QualiMaster infrastructure. In particular, the adaptation manager prepares the running QualiMaster platform to act upon the adaptation needs of the system, by small- and large-scale changes, e.g., the tuning a threshold to reduce or increase sensitivity (small scale adaptation), or change of the pipeline altogether or the number of nodes for the processing (large scale adaptation). Given that the real-time response of the QualiMaster platform might need to be smaller than the amount of time needed for large scale pipeline reconfiguration in the presence of special-purpose hardware (e.g., full reconfiguration of the MAX system might take several seconds), the Adaptation Manager will need to assess (in conjunction with the Platform Administrator) the desirability of the adaptation process vis-a-vis the real-time system requirements, as well as optimising the scheduling of workloads onto the reconfigurable hardware to minimise the amount of time spent reconfiguring.

The tasks of the Adaptation Manager are detailed through the following use cases:

- UC-AM1: Define quality characteristics of processing elements
- UC-AM2: Define pipeline quality parameters
- UC-AM3: Define reactive adaptation rules
- UC-AM4: Define proactive adaptation rules

- UC-AM5: Monitor execution of adaptation rules (i.e., reflective adaptation)

Some use cases may overlap with the Pipeline Designer (from a domain perspective) or the Platform Administrator (from a resource perspective), in particular UC-AM3 and UC-AM4. However, depending on the actual organization structure, the role of the Adaptation Manager may also jointly be filled by a Pipeline Designer or a Platform Administrator.

We will call the related tool implementing these use cases the QualiMaster *Adaptation Manager Tool* (AM tool).

5.2.1 Use Case: Define Quality Characteristics of Processing Elements

This use case describes how quality parameters and quality characteristics for processing elements are defined. Thereby, the quality parameters to be measured for a processing element (and how the parameters can be measured) as well as relating the quality characteristics (behavior over time determined by analysis or lab measurements of the individual algorithms) to the processing elements are specified. This is the basis for deriving quality characteristics for entire pipelines. We will detail the notion of quality characteristics in future deliverables (in particular D1.2 and D2.1).

Use Case Identifier	**UC-AM1**
Use Case Name	Define quality parameters of processing elements
Actor	Adaptation Manager
Goal	Define quality and adaptation parameters of data processing elements and methods for measuring them
Precondition	Data processing algorithms are added to the infrastructure (UC-PA3 or UC-PA4) and platform quality parameters are defined (UC-PA1 or UC-PA2).
Postcondition	The description of data processing elements is augmented with quality characteristics that can be taken into account in the quality-driven adaptation process (and the definition of adaptation rules).
Scenario Sequence	1. The Adaptation Manager starts the QualiMaster AM tool. 2. The AM tool displays the list of existing data processing elements. 3. The Adaptation Manager selects one of the data processing elements. 4. The AM tool views the description and properties of the selected data processing element. 5. The Adaptation Manager defines/modifies the quality characteristics of the selected processing element providing their description and metrics, e.g., based on measuring the processing element in certain settings. 6. The AM tool validates the provided characteristics, in particular, whether the underlying quality parameters can actually be determined by the QualiMaster infrastructure at runtime. The Adaptation Management tool saves the changed/defined quality characteristics. 7. The Adaptation Manager defines how individual quality parameters can be measured in terms of low-level monitoring functionalities provided by the QualiMaster platform. Further, derived (calculated) quality parameter calculated can be specified based on already defined quality parameters, in

	particular those determined by low-level measurements. 8. The AM tool validates the provided measurement methods, e.g., whether actual implementations of the methods are provided by the QualiMaster platform (through its configuration). 9. The Adaptation Manager requests saving the definitions. 10. The AM tool saves the changes and definitions.
Extensions	2b Data processing elements displayed as groups of families/clusters that share the same functionality. 5a Quality characteristics are defined for a whole family of processing elements and apply for all members of the family. 6a If quality characteristics for already known quality parameters are missing, the AM tool issues a warning that adaptation may not be able to take these parameters into account. 6b If quality characteristics for unknown quality parameters are specified, the AM tool issues a warning that the underlying quality parameters must be specified or measured. 7a If no measurement methods can be defined, the Adaptation Manager may specify constant values, such as quality levels for each individual member of the family. 8a Akin to 6a and 6b 10a If saving the information fails, the AM tool informs the Adaptation Manager by an error message.
Business Rules	• Quality parameters must either be measured at runtime or defined by the Adaptation manager • If measurement of quality parameters is not defined, the adaptation may ignore the related quality parameters. • If quality characteristics for quality parameters are not defined, the adaptation may ignore these quality parameters.
Data/Functions	a. Access to the data processing elements repository (including metadata). b. Formatting of data processing element information such as listing or grouping. c. Access to the description of derived quality measurement methods. d. Access to the configuration of the QualiMaster platform.

5.2.2 Use Case: Define Pipeline Quality Characteristics

This use case complements UC-AM1 (Section 5.2.15.1.2) by defining how to derive the quality characteristics of entire data analysis pipelines based on the characteristics of the constituting processing elements. Basically, we aim at end-to-end quality characteristics, but also quality characteristics for pipeline parts might be needed. Thus, we refer in this use case to both options using the term “(end-to-end) quality characteristics”, but we will detail this use case based on further discussions in future deliverables (in particular D1.2 and D4.1).

Use Case Identifier	UC-AM2
Use Case Name	Define pipeline quality characteristics
Actor	Adaptation Manager
Goal	Define methods for measuring/estimating the (end-to-end) quality characteristics (including costs) of a pipeline by propagating quality, error and cost of individual data processing elements through the pipeline.
Precondition	Pipelines are defined (UC-PD1, UC-PD2) and the quality characteristics of the individual data processing elements are available (UC-AM1)
Postcondition	The end-to-end quality and cost of the pipeline can be measured/estimated
Scenario Sequence	1. The Adaptation Manager starts the QualiMaster AM tool. 2. The AM tool displays the list of existing pipelines 3. The Adaptation Manager selects one pipeline. 4. The AM tool displays the structure of the processing elements of the selected pipeline. 5. The Adaptation Manager defines or modifies the methods for measuring/estimating the (end-to-end) quality characteristics (including costs) of the pipeline depending on the quality characteristics of the individual processing elements. 6. The AM tool validates the methods, e.g., whether the required quality characteristics of the individual data processing elements are available to calculate the (end-to-end) pipeline characteristics. 7. The Adaptation Manager requests saving the definitions. 8. The AM tool saves the changes
Extensions	4a The AM tool highlights those processing elements without complete specification of quality characteristics. 6a The AM tool detects inconsistencies, e.g., missing individual quality characteristics of the processing elements and warns the Adaptation Manager. The use case continues at step 5. 6b The AM tool identifies missing measurement methods for quality parameters and warns the Adaptation Manager accordingly. 8a The AM tool cannot store the related elements and informs the Adaptation Manager about the error. The use case continues at step 5.
Business Rules	• Only consistently specified (end-to-end) quality characteristics can be propagated and determined.
Data/Functions	a. Access to the Pipeline Repository. b. Access to the data Processing Elements Repository. c. Structural display (or visualization) of pipelines. d. Access to the software artefact repository to store/modify

	implemented quality measurement methods so that the infrastructure instantiation process can integrate them.

5.2.3 Use Case: Define Reactive Adaptation Rules

In this use case, the specification of the actual behavior of the reactive adaptation is described. Reactive adaptation focuses on the detection of certain triggers and to quickly perform adaptation changes without extensive planning or predictions of quality characteristics. The specification of predictive adaptation rules will be described as an extended use case in Section 5.2.4.

The specification of the reactive adaptation is based on the quality parameters and characteristics introduced by UC-AM1 and UC-AM2. As mentioned above, the Adaptation Manager role may also be filled by the Platform Designer or the Platform Administrator, each from his/her specific (domain vs. resource) view. The aim of this use case is to specify the adaptive behaviour and the boundaries of the adaptation space rather than implementing it in a fixed way. This enables adjusting, modifying or evolving the adaptation behaviour without digging into its actual implementation (which will be derived as one step during the platform instantiation, see UC-PA14). Please note that we use the generic term “adaptation rule” to denote the elements of a (declarative) adaptivity specification. We will detail this use case as well as the notion of adaptation rules in future deliverables (in particular D1.2 and D4.1)

Use Case Identifier	**UC-AM3**
Use Case Name	Define reactive adaptation rules
Actor	Adaptation Manager
Goal	Define quality-driven adaptation rules to be implemented by the adaptation module at run-time on the pipeline level
Precondition	Processing pipelines exist (UC-PD1, UC-PD2) and methods for measuring/estimating their (end-to-end) quality characteristics are defined (UC-AM2)
Postcondition	Define quality-driven adaptation rules to be considered and executed by the adaptation module at run-time on the pipeline level.
Scenario Sequence	1. The Adaptation Manager starts the QualiMaster AM tool. 2. The AM tool displays the existing pipelines. 3. The Adaptation Manager selects the pipeline for the definition of adaptation rules. 4. The AM tool displays the actual adaptation rules for the selected pipeline. 5. The Adaptation Manager defines (or modifies) a set of quality parameters (such as data load, velocity or resources consumption, quality impact on the pipeline) to be monitored. 6. The AM tool validates the parameters, thresholds and tradeoffs against the available measurement methods for quality parameters. 7. The Adaptation Manager defines (or modifies) the set of reactive adaptation rules (including thresholds and tradeoffs for the quality parameters) to be executed by the adaptation module. 8. The AM tool validates the rules against the parameters specified in step 5 as well as the rules for potential

	inconsistencies. 9. The Adaptation Manager requests saving the definitions. 10. The AM tool saves the changes
Extensions	6a Validation fails as underspecified quality parameters shall be used. An error message is displayed to the Adaptation Manager and the use case continues at step 5. 8a Validation fails as underspecified quality parameters shall be used. An error message is displayed to the Adaptation Manager and the use case continues at step 5. 8b Validation fails as inconsistencies or cyclic dependencies have been specified in the reactive rules and the use case continues at step 5. 10a Saving the changes fails for some reasons so that the AM tool informs the Adaptation Manager in terms of an error message. The use case continues at step 5.
Business Rules	• Inconsistent or invalid adaptation rules shall not be turned into an implementation or considered at runtime.
Data/Functions	a. Access to the Pipeline Repository b. Access to the Processing Elements Repository c. Adaptivity rule validation

5.2.4 Use Case: Define Proactive Adaptation Rules

This use case is actually an extension to the definition of proactive adaptation rules (UC-AM3), as in addition to the adaptation rules also mechanisms to predict quality parameters and characteristics must be specified. In order to keep the use cases readable, we did not describe these extensions within UC-AM3, but provide an extended description in this section.

Use Case Identifier	**UC-AM4**
Use Case Name	Define proactive adaptation rules
Actor	Adaptation Manager
Goal	Define quality-driven adaptation rules to be implemented by the adaptation module at run-time on the pipeline level
Precondition	Processing pipelines exist (UC-PD1, UC-PD2) and methods for measuring/estimating their (end-to-end) quality characteristics are defined (UC-AM2)
Postcondition	Define quality-driven adaptation rules to be considered and executed by the adaptation module at run-time on the pipeline level.
Scenario Sequence	1. The Adaptation Manager starts the QualiMaster AM tool. 2. The AM tool displays the existing pipelines. 3. The Adaptation Manager selects the pipeline for the definition of adaptation rules. 4. The AM tool displays the actual adaptation rules for the selected pipeline. 5. The Adaptation Manager defines (or modifies) a set of quality

	parameters (such as data load, velocity or resources consumption, quality impact on the pipeline) to be monitored and their thresholds and tradeoffs that should trigger a reactive adaptation. 6. The AM tool validates the parameters, thresholds and tradeoffs against the available measurement methods for quality parameters. 7. The Adaptation Manager defines methods for predicting individual quality parameters and characteristics, e.g., in terms of software components. 8. The AM tool validates that the data provided by the prediction methods fits to the related quality parameter or characteristic. 9. The Adaptation Manager defines (or modifies) the set of proactive adaptation rules to be executed by the adaptation module in response to the triggers (such as adjusting the filtering and sampling of data or switching to an alternative execution path of the pipeline, etc.) 10. The AM tool validates the rules against the parameters specified in step 5 as well as the rules for potential inconsistencies. 11. The Adaptation Manager requests saving the definitions. 12. The AM tool saves the changes.
Extensions	6a Validation fails as underspecified quality parameters shall be used. An error message is displayed to the Adaptation Manager. The use case continues at step 5. 8a Validation fails as for example the data types of the prediction mechanism and the quality parameters do not match or cannot be converted. The use case continues at step 5. 10a Validation fails as underspecified quality parameters shall be used. An error message is displayed to the Adaptation Manager. The use case continues at step 5. 10b Validation fails as inconsistencies or cyclic dependencies have been specified in the predictive rules or the plan derivation. The use case continues at step 5. 12a Saving the changes fails for some reasons so that the AM tool informs the Adaptation Manager in terms of an error message. The use case continues at step 5.
Business Rules	• Inconsistent or invalid adaptation rules or specifications for plan derivation shall not be turned into an implementation or considered at runtime. • Inconsistent prediction mechanisms shall not lead to runtime failures.
Data/Functions	a. Access to the Pipeline Repository b. Access to the Processing Elements Repository c. Adaptivity rule validation d. Prediction component validation

	e. Predictive plan validation f. Access to the software artefact repository to store/modify prediction components so that the infrastructure instantiation process can integrate them.

5.2.5 Use Case: Monitor Execution of Adaptation Rules

This use case aims at the identification of improvement potential or problems caused by the adaptation of the pipeline execution (reflective adaptation with human-in-the-loop). Therefore, the QualiMaster infrastructure will provide detailed logs on the executed adaptation actions and the AM tool supports the Adaptation Manager in reviewing and analyzing the log results.

Use Case Identifier	**UC-AM5**
Use Case Name	Monitor execution of adaptation rules
Actor	Adaptation Manager
Goal	The adaptation manager monitors the execution of the adaptation rules and their impact to identify any needs for adjustments.
Precondition	Adaptation rules are defined (UC-AM3, UC-AM4) and logs of execution history of adaptations are provided by the QualiMaster infrastructure.
Postcondition	Needs for adaptation adjustments or improvements are identified
Scenario Sequence	1. The Adaptation Manager starts the QualiMaster AM tool 2. The AM tool displays the existing pipelines. 3. The Adaptation Manager selects the pipeline to monitor the execution of the adaptation for. 4. The AM tool retrieves the adaptation execution logs from the QualiMaster infrastructure and performs reflective analysis on the actual log, historical log information and past analyses. 5. The Adaptation Manager reviews the logs and the analysis results and identifies needs for modifications/extensions of the reactive or proactive adaptation rules and marks them for modification (continuing at UC-AM3 step 2 or UC-AM4 step 2, respectively).
Extensions	4a Retrieving the execution log fails for some reason. The AM tool informs the Adaptation manager accordingly. The use case continues at step 2. 4b The reflective analysis does not identify any issues or improvement potential. Then the Adaptation Manager may continue with a manual analysis. 5a The Adaptation Manger does not identify any needs for changes. Then the use case ends at step 5.
Business Rules	
Data/Functions	a. Access to Adaptation Execution Logs (including a sufficient level of details for quality parameters, characteristics, predictions, plans etc) through the QualiMaster infrastructure. b. Reflective adaptation analysis

5.3 Platform Administrator

The Platform Administrator, as introduced in Section 2.2, will setup, install and maintain the QualiMaster infrastructure. In particular, the platform administrator will be responsible for the configuration, and (if needed) reconfiguration of hardware used for the execution of the data processing pipelines, including the configuration of special-purpose hardware. This task entails knowledge of the availability of resources as well as quantitative performance aspects (in collaboration with the Adaptation Manager) in order to properly configure the QualiMaster platform. The related tool supporting the Platform Administrator will be called the QualiMaster *Platform Administration* tool (PA tool). In the remainder of this section, the related use cases are described:

- UC-PA1: Define platform quality parameter (as basis for quality constraints)
- UC-PA2: Modify platform quality parameter
- UC-PA3: Add data processing algorithm (including related visualization for new data)
- UC-PA4: Modify data processing algorithm (including related visualization)
- UC-PA5: Add hardware-based algorithm
- UC-PA6: Modify hardware-based algorithm
- UC-PA7: Configure data sources and sinks
- UC-PA8: Start pipeline
- UC-PA9: Stop pipeline
- UC-PA10: Configure QualiMaster platform for software-based execution
- UC-PA11: Configure QualiMaster platform for hardware-based execution
- UC-PA12: Start platform
- UC-PA13: Stop platform
- UC-PA14: Instantiate platform

As the tasks of the Platform Administrator may affect running pipelines, in most cases an explicit approval of the ultimate execution is required.

5.3.1 Use Case: Define Platform Quality Parameters

The QualiMaster platform will be able to monitor low-level quality parameters, such as data stream related measurements (e.g., tuples/throughput per second, etc.) or resource consumption (e.g., execution time or memory consumption). This use case is about defining the quality parameters (considering that implementing them in some cases may require manual modifications of the QualiMaster platform). However, measuring certain quality parameters may influence the performance of the execution, in particular, if unnecessary measurements are performed. Therefore, the infrastructure instantiation process (UC-PA14) will take care of those quality parameters actually used in the definition of processing families (UC-PD1, UC-PD-2) or pipelines (UC-PD1, UC-PD-2) and disable or even eliminate unused measurements.

Use Case Identifier	**UC-PA1**
Use Case Name	Define platform quality parameters
Actor	Platform Administrator
Goal	Define the (low-level) quality parameters for the QualiMaster infrastructure and the methods to measure them in the platform.
Precondition	QualiMaster platform is configured (UC-PA10, optionally UC-PA11)
Postcondition	The quality parameters are defined and can be used for the

	specification of quality constraints by the Adaptation Manager.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the list of existing quality parameters. 2. The PA tool displays the existing quality parameters. 3. The Platform Administrator defines a new quality parameter in terms of its metadata and its implementation (either as a component or by modification of the QualiMaster infrastructure). 4. The PA tool validates the input and saves the changes
Extensions	2a Quality parameters are categorized in groups/clusters that share the same functionality. 3a No new quality parameter is required. The use case stops here. 4a Validation fails and the platform administrator is informed about the related reason. The use case continues at step 3.
Business Rules	
Data/Functions	a. Access to the (metadata) of the platform quality parameters. b. Display of quality parameters as list or as groups c. Access to the software artefact repository to store/modify monitoring components so that the infrastructure instantiation process can integrate and the infrastructure can execute them.

5.3.2 Use Case: Modify Platform Quality Parameters

In addition to the definition of (new) platform quality parameters (UC-PA1), platform parameters may need to be modified, e.g., disabled or a more recent implementation shall be configured. However, due to possible references from the pipeline families, the pipelines and the adaptation specification, the use case does not support the deletion of quality parameters rather than disabling them. Changes to the quality parameters may require a review of the referring elements and, thus, explicit approval.

Use Case Identifier	**UC-PA2**
Use Case Name	Modify platform quality parameters
Actor	Platform Administrator
Goal	Define the (low-level) quality parameters for the QualiMaster infrastructure and the methods to measure them in the platform.
Precondition	Platform quality parameters are defined (UC-PA1)
Postcondition	The quality parameters are defined and can be used for the specification of quality constraints.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the list of existing quality parameters. 2. The PA tool displays the existing quality parameters. 3. The Platform Administrator changes the metadata of a quality parameter, in particular in terms of enabling / disabling individual quality parameters or changing the underlying implementation. Disabling a quality parameter requires review

	of the quality characteristics (UC-AM1) or the processing pipeline definitions (UC-PD2). 4. The PA tool validates the input and saves the changes. 5. The Platform Administrator approves the new algorithm(s). 6. The PA tool acknowledges the approval.
Extensions	2a Quality parameters are categorized in groups/clusters that share the same functionality. 3a No new quality parameter is required. The use case stops here. 4a Validation fails and the platform administrator is informed about the related reason. The use case continues at step 3. 7a No approval happens so that the new algorithms will not be considered for actual execution until explicit approval. Then the use case ends here.
Business Rules	• Quality parameters shall not be deleted, just disabled and thus shall not be available to the upper level layers. • Modifications of existing quality parameters require explicit approval as running pipelines may be affected.
Data/Functions	a. Access to the (metadata) of the platform quality parameters. b. Display of quality parameters as a list or as groups. c. Access to the software artefact repository to store/modify monitoring components so that the infrastructure instantiation process can integrate and the infrastructure can execute them. d. Approval mechanism.

5.3.3 Use Case: Add Data Processing Algorithm

As a basis for defining quality characteristics (UC-AM1), data processing algorithms must be known to the QualiMaster platform. In this use case, the Platform Administrator is enabled to incorporate the algorithms in terms of their implementation. Currently, the specification of the quality characteristics (UC-AM1) is conceptually separated from the introduction of a data processing algorithm (this use case). However, further work may lead to an integrated approach. One example is to specify (default) quality characteristics with the algorithm, e.g., in terms of source code annotation or a manifest file. This may for example simplify the specification and maintenance of quality characteristics in UC-AM1 and even of adding new data processing algorithms. Further, data processing algorithms may require related data visualization algorithms to be instantiated into the applications in case of new kind of data. Thus, a relation between processing algorithms and visualizations is made in this use case.

Use Case Identifier	**UC-PA3**
Use Case Name	Add data processing algorithm
Actor	Platform Administrator
Goal	Include a new data processing algorithm in the QualiMaster infrastructure.
Precondition	The QualiMaster platform is configured (UC-PA10, optionally UC-PA11).

Postcondition	The new data processing algorithm is incorporated and can be used as part of a data processing family.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the data processing algorithm view. 2. The PA tool displays the already specified data processing algorithms. 3. The Platform Administrator selects that a new algorithm shall be added. 4. The PA tool enables the Platform Administrator to define the metadata for the algorithm. 5. The Platform Administrator provides the metadata, such as the containing algorithm family, the name of the algorithm or its inputs and outputs (or “unstructured”) and specifies the actual implementation (e.g., JAR) of the specific algorithm, possibly including a reference to an existing visualization algorithm or a new visualization algorithm (in the JAR). 6. The PA tool validates the input. 7. The Platform Administrator requests storing the new algorithm. 8. The PA tool saves the metadata, places the implementation into the processing algorithm repository and request for approving the changes. 9. The Platform Administrator approves the new algorithm(s). 10. The PA tool acknowledges the approval.
Extensions	4a The PA tool takes over the metadata from source code annotations. 6a Validation fails, e.g., as the implementation is not accessible, not compliant to the algorithm family or metadata is missing. Then the Platform Administrator is informed and the use case continues at step 4. 8a Saving the metadata or the implementation fails. Then the use case continues at step 4. 8b No QualiMaster platform is running so that no approval is needed. The use case stops here. 10a No approval happens so that the new algorithms will not be considered for actual execution until explicit approval. Then the use case ends here.
Business Rules	• Only valid data processing algorithms shall be made available to the upper level layers of the QualiMaster infrastructure. • New data processing algorithms become only available to the platform if actually needed and due to explicit approval of the Platform Administrator to ensure consistency of the execution.
Data/Functions	a. Access to the Data Processing Algorithm Repository. This implies access to the software artefact repository. b. Algorithm approval mechanism.

5.3.4 Use Case: Modify Data Processing Algorithm

In addition to the definition of (new) data processing algorithms (UC-PA3 in Section 5.3.3), processing algorithms may need to be modified, e.g., disabled or a more recent implementation shall be used. However, due to possible references from the pipeline families, the use case does not support the deletion of processing algorithms. Instead, it enables disabling a processing algorithm so that it is not available anymore, but requires a review of the referring elements as not an algorithm family may be empty.

Use Case Identifier	**UC-PA4**
Use Case Name	Modify data processing algorithm
Actor	Platform Administrator
Goal	Modify an existing data processing algorithm in the QualiMaster infrastructure.
Precondition	The QualiMaster platform is configured (UC-PD10, optionally UP-PD11) and data processing algorithms are specified (UC-PA3).
Postcondition	The modified data processing algorithm is incorporated and can be used as part of a data processing family.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the data processing algorithm view. 2. The PA tool displays the already specified data processing algorithms. 3. The Platform Administrator selects the algorithm that shall be modified. 4. The PA tool enables the Platform Administrator to define the metadata for the algorithm. 5. The Platform Administrator provides the changed metadata such as whether the algorithm is disabled, its name or its updated implementation (e.g., JAR). 6. The PA tool validates the input. 7. The Platform Administrator requests storing the changed algorithm. 8. The PA tool saves the metadata and places the implementation into the processing algorithm repository and requests for explicit approval. 9. The Platform Administrator approves the changed algorithm(s). 10. The PA tool acknowledges the approval.
Extensions	4a The PA tool takes over the metadata from source code annotations of the provided implementation. 6a Validation fails, e.g., as the implementation is not accessible or metadata is missing. Then the Platform Administrator is informed accordingly and the use case continues at step 4. 8a Saving the metadata or the implementation fails. Then the use case continues at step 4. 8b No QualiMaster platform is running so that no approval is needed. The use case stops here. 9a No approval happens so that the changed algorithms will not be considered for actual execution until explicit approval. Then the use case ends here.
Business Rules	• Only valid data processing algorithms shall be made available

	to the upper level layers of the QualiMaster infrastructure. • New data processing algorithms become only available to the platform if actually needed and due to explicit approval of the Platform Administrator to ensure consistency of the execution.
Data/Functions	a. Access to the Data Processing Algorithm Repository. This implies access to the Software Artefact Repository. b. Algorithm approval mechanism.

5.3.5 Use Case: Add Hardware-based Data Processing Algorithm

Hardware-based data processing algorithms can transparently be used in algorithm families. However, in order to avoid unnecessary overhead, arbitrary switches between software-based and hardware-based algorithms shall be avoided. Further, the platform instantiation process (UC-PA14) may assemble multiple (alternative) hardware-based algorithms for the execution on one reconfigurable hardware unit, e.g., a data flow engine. Please note, that in contrast to software-based algorithms, hardware-based algorithms have a clearly determined performance behavior if once laid out for a specific reconfigurable hardware. Also here, a integration with the quality characteristics (UC-AM1), e.g., in terms of source code annotations or manifest may be considered in future. Further, data processing algorithms may come with related data visualization algorithms to be instantiated into the applications. Thus, a relation between processing algorithms and visualizations is made in this use case. Please note that the instantiation of the composition of hardware-based algorithms and their actual layout towards a specific hardware unit is considered during the platform instantiation process.

Use Case Identifier	**UC-PA5**
Use Case Name	Add hardware-based data processing algorithm
Actor	Platform Administrator
Goal	Include a new hardware-based data processing algorithm in the QualiMaster infrastructure.
Precondition	The QualiMaster platform is configured (UC-PD10, optionally UC-PD11).
Postcondition	The new data processing algorithm is incorporated and can be used as part of a data processing family.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the data processing algorithm view. 2. The PA tool displays the already specified data processing algorithms. 3. The Platform Administrator selects that a new hardware-based algorithm shall be added. 4. The PA tool enables the Platform Administrator to define the metadata for the algorithm. 5. The Platform Administrator provides the metadata such as its name, the containing family or its inputs and outputs (or “unstructured”) and specifies the actual implementation (e.g., JAR) of the specific algorithm, possibly including a reference to an existing visualization algorithm or a new visualization algorithm (in the JAR).

	6. The PA tool validates the input. 7. The Platform Administrator requests storing the new algorithm. 8. The PA tool saves the metadata, places the implementation into the processing algorithm repository and requests for explicit approval. 9. The Platform Administrator approves the changed algorithm(s). 10. The PA tool acknowledges the approval.
Extensions	4a The PA tool takes over the metadata from source code annotations. 6a Validation fails, e.g., as the implementation is not accessible, it does not comply with the family or (hardware-based) metadata is missing. Then the use case continues at step 4. 8a Saving the metadata or the implementation fails. Then the Platform Administrator is informed and the use case continues at step 4. 8b No QualiMaster platform is running so that no approval is needed. The use case stops here. 9a No approval happens so that the changed algorithms will not be considered for actual execution until explicit approval. Then the use case ends here.
Business Rules	• Only valid data processing algorithms shall be made available to the upper level layers of the QualiMaster infrastructure. • New data processing algorithms become only available to the platform if actually needed and due to explicit approval of the Platform Administrator to ensure consistency of the execution. • Once a hardware-based algorithm is laid out for a certain reconfigurable hardware, it can be executed only on that hardware. In turn, the quality performance parameters are then known for that specific hardware.
Data/Functions	a. Access to the Data Processing Algorithm Repository. This implies access to the Software Artefact Repository. b. Algorithm approval mechanism.

5.3.6 Use Case: Modify Hardware-based Data Processing Algorithm

Akin to UC-PA4, also hardware-based processing algorithms may be modified or disabled (but actually not deleted). Please note that the instantiation of the composition of hardware-based algorithms and their actual layout towards a specific hardware unit is considered by the platform instantiation process (UC-PA14).

Use Case Identifier	**UC-PA6**
Use Case Name	Modify hardware-based data processing algorithm
Actor	Platform Administrator
Goal	Modify an existing hardware-based data processing algorithm in the QualiMaster infrastructure.

Precondition	The QualiMaster platform is configured (UC-PD10, optionally UC-PD11) and hardware-based data processing algorithms are specified (UC-PA5).
Postcondition	The modified hardware-based data processing algorithm is incorporated and can be used as part of a data processing family.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the data processing algorithm view. 2. The PA tool displays the already specified data processing algorithms. 3. The Platform Administrator selects the hardware-based algorithm that shall be modified. 4. The PA tool enables the Platform Administrator to define the metadata for the hardware-based algorithm. 5. The Platform Administrator provides the changed (hardware-specific) metadata such as whether the algorithm is disabled, its name or its updated implementation (e.g., JAR). 6. The PA tool validates the input. 7. The Platform Administrator requests storing the changed algorithm. 8. The PA tool saves the metadata, places the implementation into the processing algorithm repository and asks for explicit approval. 9. The Platform Administrator approves the changed algorithm(s). 10. The PA tool acknowledges the approval.
Extensions	4a The PA tool takes over the metadata from source code annotations of the provided implementation. 6a Validation fails, e.g., as the implementation is not accessible or metadata is missing. Then the use case continues at step 4. 8a Saving the metadata or the implementation fails. Then the use case continues at step 4. 8b No QualiMaster platform is running so that no approval is needed. The use case stops here. 9a No approval happens so that the changed algorithms will not be considered for actual execution until explicit approval. Then the use case ends here.
Business Rules	• Only valid data processing algorithms shall be made available to the upper level layers of the QualiMaster infrastructure. • New data processing algorithms become only available to the platform if actually needed and due to explicit approval of the Platform Administrator to ensure consistency of the execution. • Once a hardware-based algorithm is laid out for a certain reconfigurable hardware, it can be executed only on that hardware. In turn, the quality performance parameters are then known for that specific hardware.
Data/Functions	a. Access to the Data Processing Algorithm Repository. This

	implies access to the Software Artefact Repository. b. Algorithm approval mechanism.

5.3.7 Use Case: Configure Pipeline Sources and Sinks

Conceptually, we separate the definition of data analysis pipelines (on domain level) and the technical specification of pipeline sources and sinks (on platform / resource level). In particular, sources and sinks may need credentials, imply network access restrictions or require an adapter implementation to bring the data from an arbitrary source into the QualiMaster data stream processing platform. Technically, sources and sinks may be considered as an (abstract or draft) pipeline specification, which is then refined by the Pipeline Designer use cases (UC-PD1 or UC-PD2).

Use Case Identifier	**UC-PA7**
Use Case Name	Configure pipeline sources and sinks
Actor	Platform Administrator
Goal	Define the technical information about data sources and sinks such as IP addresses, credentials, adapters, etc.
Precondition	
Postcondition	Draft pipeline source and sink definition has been created and successfully stored.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the pipeline source and sink definitions. 2. The PA tool displays the configured sources and sinks. 3. The Platform Administrator selects that a new source shall be entered. 4. The PA tool displays the input for the metadata. 5. The Platform Administrator now enters the metadata, e.g., the internet address, the credentials, the adaptor implementation realizing the integration of the physical source/sink with the QualiMaster infrastructure (e.g., in terms of a JAR file) or the data structure of the associated data stream (or “unstructured”). 6. The PA tool validates the provided information. 7. The Platform Administrator requires saving the information entered above. 8. The PA tool acknowledges that the source / sink has been successfully stored and requests for explicit approval. 9. The Platform Administrator approves the changed source(s) or sink(s). 10. The PA tool acknowledges the approval.
Extensions	3a The Platform Administrator selects that a new sink shall be entered. 3b The Platform Administrator selects the existing source that shall be modified. 3c The Platform Administrator selects the existing sink that shall

	be modified. 4a If step 3b or 3c was executed before, the already configured metadata for the selected data source or sink is displayed for editing. 5a In case of sinks (step 3a or 3c) the Platform Administrator may enter the network addresses for which access shall be granted (or permitted). 6a Validation fails, e.g., due to missing required information or as the source or sink cannot be accessed through the network. Then the Platform Administrator is notified and the use case continues at step 5. 8a Saving the information fails for some reason and the Platform Administrator is notified by the PA tool accordingly. Then the use case continues at step 5. 8b No QualiMaster platform is running so that no approval is needed. The use case stops here. 9a No approval happens so that the changed algorithms will not be considered for actual execution until explicit approval. Then the use case ends here.
Business Rules	• Access limitations to data sources or sinks may apply, e.g., in case of licensed (financial) data or to separate customer groups. • Invalid source or sink information shall not be made available to a running QualiMaster platform.
Data/Functions	a. Access to the Pipeline Repository. b. Source and sink validation including network access. c. Access to the software artefact repository to store/modify adapters so that the infrastructure instantiation process can integrate them.

5.3.8 Use Case: Start Pipeline

When a pipeline is completely specified ranging from required quality characteristics over algorithms, algorithm families, adaptation up to the pipeline data flow and the infrastructure instantiation process (UC-PA14) has derived the related software artifacts, the pipeline is ready for execution on the QualiMaster infrastructure. Akin to the acknowledgement of changes, this is currently considered as an explicit task, as compute resources will be allocated and existing pipelines may (potentially) be affected.

Use Case Identifier	**UC-PA8**
Use Case Name	Start pipeline
Actor	Platform Administrator
Goal	Start a pipeline that has been fully configured, stored and instantiated for execution with the QualiMaster platform.
Precondition	The platform is running (UC-PA12) and pipelines are configured (UC-PD1 or UC-PD2).
Postcondition	The pipeline is deployed, started and being executed adaptively.

Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the pipelines. 2. The PA tool displays the configured pipelines. 3. The Platform Administrator selects the pipeline he/she wants to start. 4. The PA tool verifies that it can start the selected pipeline (using the quality impact analysis also accessible to the Platform Designer as described in Section 5.1) and asks the Platform Administrator whether the selected platform shall actually be started now. 5. The Platform Administrator approves the start of the pipeline. 6. The PA tool starts the pipeline on the QualiMaster platform and acknowledges that the pipeline was successfully started.
Extensions	4a The PA tool cannot verify that the selected pipeline can be started, e.g., as the execution quality of running pipelines would be affected. The Platform Administrator is informed and the use case continues at step 2. 5a The Platform Administrator does not approve the action and so the use case stops here. 6a The deployment or physical execution fails for some reason. Then the Platform Administrator is notified accordingly.
Business Rules	• Only valid, instantiated and executable pipelines can be started. • The start of pipelines shall not affect the execution of already running pipelines.
Data/Functions	a. Pipeline quality analysis b. Access to the Pipeline Repository c. Access to the lower level QualiMaster infrastructure, e.g., deployment and start of pipelines

Additional remarks:

In an implementation of the QualiMaster infrastructure, separate tasks for deploying and undeploying a pipeline in addition to starting and stopping might be needed. However, this actually depends on the capabilities of the quality impact analysis, which could determine the most appropriate subset of machines to run a pipeline on and, thus, would make an explicit (un)deployment superfluous. We will describe related use cases, if necessary in the refined version of this document (D1.2).

5.3.9 Use Case: Stop Pipeline

This is the counterpart use case of UC-PA8, i.e., to explicitly stop a running pipeline.

Use Case Identifier	**UC-PA9**
Use Case Name	Stop pipeline
Actor	Platform Administrator
Goal	Stop a pipeline that has been already started in QualiMaster platform.
Precondition	QualiMaster platform is running (#UC-PD12) and pipelines are

	configured (UC-PD1 or UC-PD2)
Postcondition	The selected pipeline is stopped.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the pipelines. 2. The PA tool displays the configured pipelines. 3. The Platform Administrator selects the pipeline he/she wants to stop. 4. The PA tool verifies that the pipeline can be stopped without affecting other pipelines and asks the Platform Administrator for final approval. 5. The Platform Administrator approves the action. 6. The PA tool stops the selected pipeline through the QualiMaster platform and acknowledges the Platform Administrator about successfully stopping the selected pipeline.
Extensions	4a The PA tool cannot verify the selected pipeline can be stopped, e.g., as the execution quality of running pipelines would be affected. Then the Platform Administrator is informed and the use case continues at step 2. 5a The Platform Administrator does not approve the action and so the use case stops here. 6a Stopping physical execution or undeployment fails for some reason. Then the Platform Administrator is notified accordingly.
Business Rules	• Only running pipelines may be stopped, but stopping a pipeline shall not affect other running pipelines.
Data/Functions	a. Pipeline quality analysis b. Access to the Pipeline Repository c. Access to the lower level QualiMaster infrastructure, e.g., stopping and undeploying pipelines

Additional remarks:

In an implementation of the QualiMaster infrastructure, separate tasks for deploying and undeploying a pipeline in addition to starting and stopping might be needed. However, this actually depends on the capabilities of the quality impact analysis, which could determine the most appropriate subset of machines to run a pipeline on and, thus, would make an explicit (un)deployment superfluous. We will describe related use cases, if necessary in the refined version of this document (D1.2).

5.3.10 Use Case: Configure QualiMaster Platform for Software-based Execution

This step is required for bootstrapping the QualiMaster Platform for a certain execution environment or when new hardware becomes available. This includes the configuration of the underlying standard hardware, e.g., their physical resources or the numbers of threads to be used for pipeline execution. Please note that we describe the configuration of hardware-based execution in UC-PA10.

Use Case Identifier	**UC-PA10**
Use Case Name	Configure QualiMaster Platform for Software-based Execution
Actor	Platform Administrator
Goal	Configure the standard hardware for software-based execution of data flow pipelines.
Precondition	Appropriate hardware is available
Postcondition	The hardware is configured and the QualiMaster platform is configured for software-based execution.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the hardware view. 2. The PA tool displays the configured hardware. 3. The Platform Administrator selects to add new servers. 4. The PA tool allows entering information about the hardware. 5. The Platform Administrator enters the information, e.g., its physical resources, its network identification or the number of threads to be used for data processing pipeline execution. 6. The PA tool validates the input. 7. The Platform Administrator requires storing the information. 8. The PA tool acknowledges the successful storage, starts the platform instantiation process and finally asks for approval. 9. The Platform Administrator approves the changes.
Extensions	3a The Platform Administrator selects to change or delete information on an individual server. 4a In case of 3a, the PA tool displays the already configured information. 6a The validation fails due to missing or inconsistent data. Then the Platform Administrator is informed and the use case continues at step 3. 8a Storage of the data fails and the administrator is notified accordingly. The use case continues at step 3. 8b Platform instantiation process fails and the administrator is notified accordingly. The use case continues at step 3. 8c Actually no instance of the QualiMaster platform is running (bootstrapping) so that an approval is not needed, but the administrator is informed on how to install and start the platform (UC-PD12). 9a The Platform administrator does not approve the changes so that they do not become effective immediately. Then the use case stops here. 9b In case of 8c, 9b is not executed.
Business Rules	• Only consistently configured instances of the QualiMaster platform may be used for pipeline execution. • In case of modifying hardware for a running QualiMaster

	platform an explicit approval is required, e.g., to ensure that the new hardware is actually switched on.
Data/Functions	a. Access to hardware information repository b. Access to already running QualiMaster platforms c. Validation of hardware information

5.3.11 Use Case: Configure QualiMaster Platform for Hardware-based Execution

This task is required for bootstrapping the QualiMaster Platform for a certain execution environment or when new reconfigurable hardware becomes available. This includes the configuration of the type and the amount of reconfigurable hardware units (e.g., MAX Data Flow Engines) and how to access them (e.g., through a host computer). While software-based execution is required, UC-PA10 is optional depending on whether reconfigurable hardware is available.

Use Case Identifier	**UC-PA11**
Use Case Name	Configure QualiMaster Platform for Hardware-based Execution
Actor	Platform Administrator
Goal	Configure the standard hardware for software-based execution of data flow pipelines.
Precondition	Supported reconfigurable hardware is available
Postcondition	The reconfigurable hardware is specified and the QualiMaster Platform is configured accordingly.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the hardware view. 2. The PA tool displays the configured hardware. 3. The Platform Administrator selects to add new reconfigurable hardware. 4. The PA tool allows entering information about the reconfigurable hardware. 5. The Platform Administrator enters the information, e.g., it's the number of types of Data Flow Engine boards as well as the host computer used for accessing and controlling the Data Flow Engines. 6. The PA tool validates the input. 7. The Platform Administrator requires storing the information. 8. The PA tool acknowledges the successful storage, starts the platform instantiation process and finally asks for approval. 9. The Platform Administrator approves the changes.
Extensions	3a The Platform Administrator selects to change or delete information on a specific Data Flow Engine board or an entire engine cluster through its host computer. 4a In case of 3a, the PA tool displays the already configured information. 6a The validation fails due to missing or inconsistent data. Then the Platform Administrator is informed and the use case

	continues at step 5. 8a Storage of the data fails and the administrator is notified accordingly. The use case continues at step 5. 8b Platform instantiation process fails and the administrator is notified accordingly. The use case continues at step 5. 8c Actually no instance of the QualiMaster platform is running (bootstrapping) so that an approval is not needed, but the administrator is informed on how to install and start the platform (UC-PA12). 9a The Platform administrator does not approve the changes so that they do not become effective immediately. Then the use case stops here. 9b In case of 8c, 9b is not executed.
Business Rules	• Hardware-based execution requires a configuration for software-based execution (UC-PD10) • Only consistently configured instances of the QualiMaster platform may be used for pipeline execution. • In case of modifying reconfigurable hardware for a running QualiMaster platform an explicit approval is required, e.g., to ensure that the new hardware is actually switched on.
Data/Functions	a. Access to hardware information repository b. Access to already running QualiMaster platforms c. Validation of reconfigurable hardware information

5.3.12 Use Case: Start QualiMaster Platform

This is the low-level task to start up a configured QualiMaster Platform during bootstrapping or after maintenance. Currently, we expect that therefore a low-level (shell) command must be issued on one of the servers.

Use Case Identifier	**UC-PA12**
Use Case Name	Stop QualiMaster Platform
Actor	Platform Administrator
Goal	Start a QualiMaster Platform.
Precondition	The QualiMaster Platform is configured (UC-PA9, optionally UC-PA10) and installed.
Postcondition	The QualiMaster Platform is started
Scenario Sequence	1. The Platform Administrator accesses the coordinator server and issues the QualiMaster platform startup command. 2. The startup command displays that the QualiMaster platform has successfully been started.
Extensions	2a Errors prevent startup and the Platform Administrator is notified accordingly.
Business Rules	• Only consistently configured instances of the QualiMaster

	platform can be started.
Data/Functions	a. Startup script

5.3.13 Use Case: Stop QualiMaster Platform

This is the low-level task to stop a configured QualiMaster Platform, e.g., during a maintenance interval. Currently, we expect that this task can be performed through the QualiMaster PA tool.

Use Case Identifier	**UC-PA13**
Use Case Name	Stop QualiMaster Platform
Actor	Platform Administrator
Goal	Stop the QualiMaster Platform.
Precondition	The QualiMaster Platform is running (UC-PA12).
Postcondition	The QualiMaster Platform is stopped
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the platforms view. 2. The PA tool displays the running platforms. 3. The Platform Administrator selects the platform to be stopped and issues the stop command. 4. The PA tool asks for explicit approval. 5. The Platform Administrator approves the stop command. 6. The selected platform is stopped and the Platform Administrator is acknowledged about successfully stopping the selected platform.
Extensions	4a At least one running pipeline prevents stopping the platform. The Platform Administrator is informed and the use case continues at step 2. 5a The Platform Administrator does not approve the stop command. In this case, the use case stops here. 6a The QualiMaster platform cannot be stopped due to technical reasons. In the extreme case, the Platform Administrator must enter the control server and stop the platform using a low-level command (akin to UC-PA10).
Business Rules	• Stopping the platform requires that no pipelines are running, i.e., stopping a QualiMaster platform may lead to the forced stop of pipelines only in very extreme cases.
Data/Functions	a. Access to the platform functionality

5.3.14 Use Case: Instantiate Platform

This use case aims at turning the generic QualiMaster platform into an instantiated one. This includes the instantiation of missing or changed artifacts based on the configuration. The Platform Administrator is responsible for this task as in particular the instantiation of hardware layouts may consume compute resources (on specific build servers).

Use Case Identifier	**UC-PA14**
Use Case Name	Instantiate Platform
Actor	Platform Administrator
Goal	Turn the generic QualiMaster platform into an instantiated one based on the configuration (models)
Precondition	The required configuration information is provided, at least the hardware information (UC-PA10, UC-PA11). Further artefacts can only be created if configuration information is available through the appropriate infrastructure use cases.
Postcondition	The artefacts described by the configuration are instantiated. If a complete configuration is provided, the platform and the configured pipelines are ready for installation / deployment / execution.
Scenario Sequence	1. The Platform Administrator starts the QualiMaster PA tool and selects the instantiation view. 2. The PA tool displays the status of the configuration. 3. The Platform Administrator selects ‘instantiate platform’. 4. The PA tool instantiates the platform.
Extensions	2a Important configuration information is missing so that the platform cannot be instantiated, i.e., the configuration is inconsistent The Platform Administrator is informed and the use case stops. 4a The Platform instantiation fails for technical reasons. Information about the failure is displayed and the use case stops here.
Business Rules	• Only consistently configured platforms shall be executed.
Data/Functions	b. Access to all repositories containing configuration information or implementation components. c. Access to the software artefact repository to store the generic QualiMaster platform as a source and to produce the instantiated platform.

6 Conclusions and Outlook

In this deliverable, we described the results of the early requirements collection for the QualiMaster applications and the QualiMaster infrastructure. We have determined and detailed the actors that will interact with QualiMaster. In particular, we identified two groups of actors, namely application users and infrastructure users, which are actually a subset of the stakeholders interested in the QualiMaster project. We then detailed the user-centric view on the QualiMaster application infrastructure for the financial domain in terms of use cases for institutional financial clients and regulators (detailing the business domains / application scenarios in the DoW). Subsequently, we discussed the requirements for the data streams to be processed and the (initial set of) algorithm families to be provided by an instantiated QualiMaster platform for the financial domain. Finally, we described the use cases for the three infrastructure users, namely, the Pipeline Designer, the Adaptation Manager and the Platform Administrator.

The collection of the use cases and requirements as well as the documentation process has already served as a good trigger for discussions about the functionalities, terminologies, and dependencies within the consortium. As scheduled in the DoW, this initial set of use cases and requirements will be further studied in the following months, which will lead to an extended and revised version of this deliverable (D1.2). In particular, to detail the actual data analysis pipeline to be used for the validation, its adaptivity space and the related adaptivity requirements. The preparation of D1.2 will foster the work on the QualiMaster infrastructure (D5.1 due in month 7) and the method deliverables of workpackages 2, 3 and 4 (D2.1, D3.1 and D4.1 due in month 12). This collection of the use cases and requirements will also guide the work in QualiMaster until the completion of the project.

References

[1] D. Acemoglu, A. Ozdaglar, and A. Tahbaz-Salehi. Systemic risk and stability in financial networks. *NBER Working Paper*, (18727), January 2013.

[2] D. M. Berry, E. Kamsties, and M. M. Krieger. From contract drafting to software specification: Linguistic sources of ambiguity—a handbook. Technical report, 2003.

[3] J. Birch. A short history of correlation networks research, 2012. http://www.fna.fi/blog/2012/07/06/a-short-history-of-correlation-networks-research/.

[4] S. Chakravarthy and Q. Jiang. *Stream Data Processing: A Quality of Service Perspective Modeling, Scheduling, Load Shedding, and Complex Event Processing*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[5] A. Cockburn. *Writing Effective Use Cases*. Addison Wesley, 2000.

[6] D. de Almeida Ferreira and A. R. da Silva. A controlled natural language approach for integrating requirements and model-driven engineering. In *International Conference on Software Engineering Advances (ICSEA '09)*, pages 518–523, Sept 2009.

[7] H. Eichelberger and K. Schmid. A Systematic Analysis of Textual Variability Modeling Languages. In *International Software Product Line Conference (SPLC '13)*, pages 12–21, 2013.

[8] European Central Bank. Recent advances in modeling systemic risk using network analysis. Technical report, January 2010.

[9] W. A. Freiwald, P. Valdes, J. Bosch, R. Biscay, J. C. Jimenez, L. M. Rodriguez, V. Rodriguez, A. K. Kreiter, and W. Singer. Testing non-linearity and directedness of interactions between neural groups in the macaque inferotemporal cortex. *J Neurosci Methods*, 94:105–119, 1999.

[10] S. Frølung and J. Koistinen. QML: A Language for Quality of Service Specification. Technical Report HPL98-10, HP Labs, HP Software Technologies Laboratory, February 1998.

[11] V. Gervasi and D. Zowghi. Reasoning about inconsistencies in natural language requirements. *ACM Trans. Softw. Eng. Methodol.*, 14(3):277–330, July 2005.

[12] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37:424—438, 1969.

[13] T. Hayashi and N. Yoshida. Estimating correlations with nonsynchronous observations in continuous diffusion models. Technical report, 2006. Working paper.

[14] K. Hlavackova-Schindler, M. Palus, M. Vejmelka, and J. Bhattacharya. Causality detection based on information-theoretic approaches to time series analysis. *Physics Reports*, 441:1–46, 2007.

[15] A. Kaiser and T. Schreiber. Information transfer in continuous processes. 110:43—62, 2002.

[16] S. Kim, D. Putrino, S. Ghosh, and E. N. Brown. A Granger Causality Measure for Point Process Models of Ensemble Neural Spiking Activity. *PCBI*, March 2011. DOI: 10.1371/journal.pcbi.1001110.

[17] F. van der Linden, K. Schmid, and E. Rommes. *Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering*. Springer, 2007.

[18] Y. Liu and M. T. Bahadori. A survey on granger causality: A computational view, October 2012. http://www-bcf.usc.edu/ liu32/granger.pdf.

[19] R.N. Mantegna. Hierarchical structure in financial markets. *Eur. Phys. J. B*, 11:193–197, 1999.

[20] C. Marchetti, B. Pernici, and P. Plebani. A Quality Model for Multichannel Adaptive Information Systems. In *International World Wide Web Conference (WWW '04)*, pages 48–54.

[21] V. Mattiussi, M. Tumminello, G. Iori, and R. N. Mantegna. Comparing correlation matrix estimators via Kullback-Leibler divergence. Technical report. http://ssrn.com/abstract=1966714.

[22] Joseph G. Nicholas. HFR Global Hedge Fund Industry Report, 2013. https://www.hedgefundresearch.com/?fuse=products-irglo.

[23] K. Pohl, G. Böckle, and F. van der Linden. *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer, 2005.

[24] INDENICA project consortium. Open Variability Modeling Approach for Service Ecosystems, 2012. http://www.indenica.eu [validated: February 2014].

[25] T. Schreiber. Measuring information transfer. *Phys. Rev. Lett.*, 85, 2000.

[26] A. Seth. Granger causality, 2007. http://www.scholarpedia.org/article/Granger_causality.

[27] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379—423, 1948.

[28] Twitter. Public apis, 2012. https://dev.twitter.com/docs/streaming-apis/streams/public.

[29] Twitter. Tweets - field guide, 2013. https://dev.twitter.com/docs/platform-objects/tweets.

[30] D. A. Vaughan. Selected definitions of "hedge fund", May 2003. Comments for the U.S. Securities and Exchange Commission Roundtable on Hedge Funds, https://www.sec.gov/spotlight/hedgefunds/hedge-vaughn.htm.