

SPEEDD – **Scalable Proactive Event-Driven Decision-making**



|  |
| --- |
| D6.1 |
| Architecture Design |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Acronym | SPEEDD | | |
| Project Title | Scalable Proactive Event-Driven Decision-making | | |
| Project Number | 619435 | | |
| Workpackage | #WP6 | Scalability and System Integration | |
| Lead Beneficiary | IBM | | |
| Editor | Alexander Kofman | | IBM |
| Contributors | Inna Skarbovsky | | IBM |
|  | Fabiana Fournier | | IBM |
|  | TBD | | TBD |
| Reviewers | Alexander Artikis | | NCSR |
|  | TBD | | TBD |
| Dissemination Level | Public | | |
| Contractual Delivery Date | 31/09/2014 | | |
| Actual Delivery Date | 31/09/2014 | | |
| Version | V1.0 | | |

Abstract

This report presents TBD

Document History

|  |  |  |
| --- | --- | --- |
| Version | Date | Comments |
| V0.1 | tbd | First draft |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[Abstract 2](#_Toc398570087)

[Document History 3](#_Toc398570088)

[Table of Contents 4](#_Toc398570089)

[List of Tables 5](#_Toc398570090)

[List of Figures 5](#_Toc398570091)

[Acronyms 6](#_Toc398570092)

[1. Introduction 7](#_Toc398570093)

[2. System Requirements 7](#_Toc398570094)

[3. Conceptual Architecture 7](#_Toc398570095)

[4. Component Architecture 7](#_Toc398570096)

[5. SPEEDD Runtime Architecture 7](#_Toc398570097)

[5.1. Complex Event Processor 7](#_Toc398570098)

[5.2. Decision Management 7](#_Toc398570099)

[5.3. Dashboard application 7](#_Toc398570100)

[5.4. Event/Data Providers 8](#_Toc398570101)

[5.5. Action Consumption – Actuators/Connectors 8](#_Toc398570102)

[6. Summary 8](#_Toc398570103)

[7. Appendix A – Technology Evaluation 8](#_Toc398570104)

[7.1. Stream Processing – requirements and evaluation criteria 8](#_Toc398570105)

[7.2. Storm 8](#_Toc398570106)

[7.3. Akka 8](#_Toc398570107)

[7.4. Spark Streaming 8](#_Toc398570108)

[7.5. Messaging Platform 8](#_Toc398570109)

List of Tables

**No table of figures entries found.**

List of Figures

**No table of figures entries found.**

Acronyms

|  |  |
| --- | --- |
| Acronym | Explanation |
| BCM | Business Collaboration Module |
| CEP | Complex Event Processing |
| EPA | Event Processing Agent |
| EPM | Event Processing Module |
| EPN | Event Processing Network |
| GE | Generic Enabler |
| FInest | Future Internet enabled optimisation of transport and logistics networks |
| JMS | Java Message Service |
| JSON | JavaScript Object Notation |
| IoT | Internet of Things |
| TCP | Transport Chain Plan |
| TEP | Transport Execution Plan |
| TPM | Transport Planning Module |
| WP | Work Package |

# Introduction

Describe the goal and the structure of this document.

# System Requirements

This section lays out the main requirements for SPEEDD prototype. The requirements are based on the requirements documents provided by the use cases.

# Conceptual Architecture

This section provides a high-level overview of SPEEDD prototype. The goal is to introduce the main concepts, high-level components and information flow without getting into implementation and technological details.

# Component Architecture

Describe the event-driven architecture paradigm and introduce the functional components. Refine the component view to include technology stack. Define integration mechanisms, APIs and data formats.

# SPEEDD Runtime Architecture

Describe in details the architecture of the runtime part. Provide an general overview of the big picture (architectural diagram). In subsections describe the architecture of the components.

## Event Bus

Describe our implementation of the event bus using KAFKA technology.

## Event/Data Providers

Explain how we’re going to stream the data into the system, for each use case.

## Action Consumption – Actuators/Connectors

Describe who and how would consume action messages issued by the system. Per use case.

## Complex Event Processor

Describe the architecture of PROTON, including implementation on STORM.

## Decision Management

Architecture of the decision management component(s), approaches, issues, etc. Possibly discuss separately the design for every use case.

## Dashboard application

Describe the architecture of the dashboard application.

# Build-Time Architecture

Describe the build-time path and the architecture in details (based on the conceptual view presented earlier).

## Event Pattern Mining

Describe how machine learning approach is used to extract complex event patterns from annotated historic data.

## Authoring of CEP Rules

Describe the process, challenges, and approach to translation of the CEP patterns discovered in using machine learning into Proton EPN definition.

## Decision Management – the Offline Part

Should see if this is relevant. If it is, describe how decision management component is configured or adjusted based on the exploration of the historic data.

# Integration – APIs and Data Formats

Describe the integration mechanisms between different components and between the system and the outside world. List and describe the APIs and data formats in use.

# Non-Functional Aspects

## Scalability

Explain why is the proposed architecture is scalable. Describe how the system will scale up and out to match the load.

## Fault Tolerance

Explain what types of failures the system is designed to stand. Describe the designed behavior of the system in case of such failures.

## Testability

Describe the approach to testing the system. Address the functional testing as well as performance testing approach as designed.

# Summary

TBD - Summarize

# Appendix A – Technology Evaluation

This could be one or more appendix parts. Here we’ll explain the approach, criteria, and the final choice of the technology stack that was made.

## Stream Processing – requirements and evaluation criteria

## Storm

## Akka

## Spark Streaming

## Choice of the Messaging Platform