alphadoop

Doflamingo

An light-weight monitoring system for Apache Hadoop

TITLE **Kafka/ Zookeeper Monitoring Module built for Flamingo Ecosystem**

DURATION **March 13, 2016 ~ June 8, 2016**

CLIENT **EXEM** PRESENTER **ALPHADOOP**

TEAM ALPHADOOP

YOUNGJAE CHANG [PM]

SEUNGHYO KANG

JARYONG LEE

CONTENTS

PART_01

PROJECT OVERVIEW

_ OBJECTIVE

Collect Performance Metrics,
Visualize it, and
Integrate it with Flamingo.

_ PROBLEM STATEMENT

Monitoring is critical to understand Hadoop Ecosystem.

Flamingo lacks ability to monitor Kafka/Zookeeper rather than nodes.

_ PROBLEM STATEMENT

Is all system working properly?

hadoop

kafka

Doflamingo

Of Course!

Check this out!

_ USEFUL CASES #1

LinkedIn processes 172,000 messages a second. It adds up to 10 billion messages a day. It encounters many engineering problems and they can only be captured via custom built monitoring tools.

_ USEFUL CASES #2

Netflix, as it now runs hundreds of clusters, it became confusing for even experts to understand how system works.

Typical Questions	*Why did my job run slower today than yesterday?*
	Can we expand the cluster to speed up my job?
	What cluster did my job run on?
	How do I get access to task logs?

_ USEFUL CASES #3

Hadoop have been proved to have big business implication, but the ease of maintenance blocks it from being mainstream. Hortonworks built Apache Ambari to solve the problem and give a single point for customers to work with.

PART_02

PROJECT REQUIREMENTS

_ FUNCTIONS

1. **Monitor and Report in Real-time**
2. **Visualize the metrics**
3. **Save metrics into Database**

_ Doflamingo WILL NOT ...

1. **Control configuration**
2. **Alarm users**

_ CONSTRAINTS

1. **Doflamingo Backend**
   - **should work on JVM**
   - **should utilize Maven ecosystem**
   - **should be integrated into Flamingo**
2. **Doflamingo Frontend**
   - **should be built with Sencha ExtJS**
   - **should communicate with WebSocket**

_ External Interfaces: Inputs

1. **Kafka Configuration [JSON]**
   - **Kafka node ip / port**
2. **Zookeeper Configuration [JSON]**
   - **Zookeeper node ip / port**
3. **RRD4J Configuration [JSON]**
   - **Path to RRD4J database**

_ External Interfaces: UI

1. **Overview**
   - **Can View Multiple Charts at Once, in Realtime.**
2. **Timeline**
   - **Can Investigate certain Moment in the History.**

_ SW Quality Attributes

M11 Requirement Compliance

M12 Requirement Traceability

M13 Requirement Change Rate

M21 Fault Density

M22 Bad Fix Rate

M31 Test Coverage

PART_03

SOLUTION

_ TECHNICAL DETAILS

[A] WHAT IS KAFKA?

A high-throughput distributed messaging system

BENEFITS

- Scalable
- High-throughput
- Distributable
- Low response time
- Save on data disk

USED IN

- LinkedIn
- Twitter
- Netflix
- Tumblr
- Foursquare

_ TECHNICAL DETAILS

[B] WHAT IS ZOOKEEPER?

Handles various errors in distributed systems.

Four Features

Using name service to separate loads.
Using distributed lock to handle synchronization error
Error detection and recovery
Configuration management

_ ARCHITECTURE

[A] WHAT IS KAFKA?

A high-throughput distributed messaging system

BENEFITS

- Scalable
- High-throughput
- Distributable
- Low response time
- Save on data disk

USED IN

- LinkedIn
- Twitter
- Netflix
- Tumblr
- Foursquare

_ ARCHITECTURE

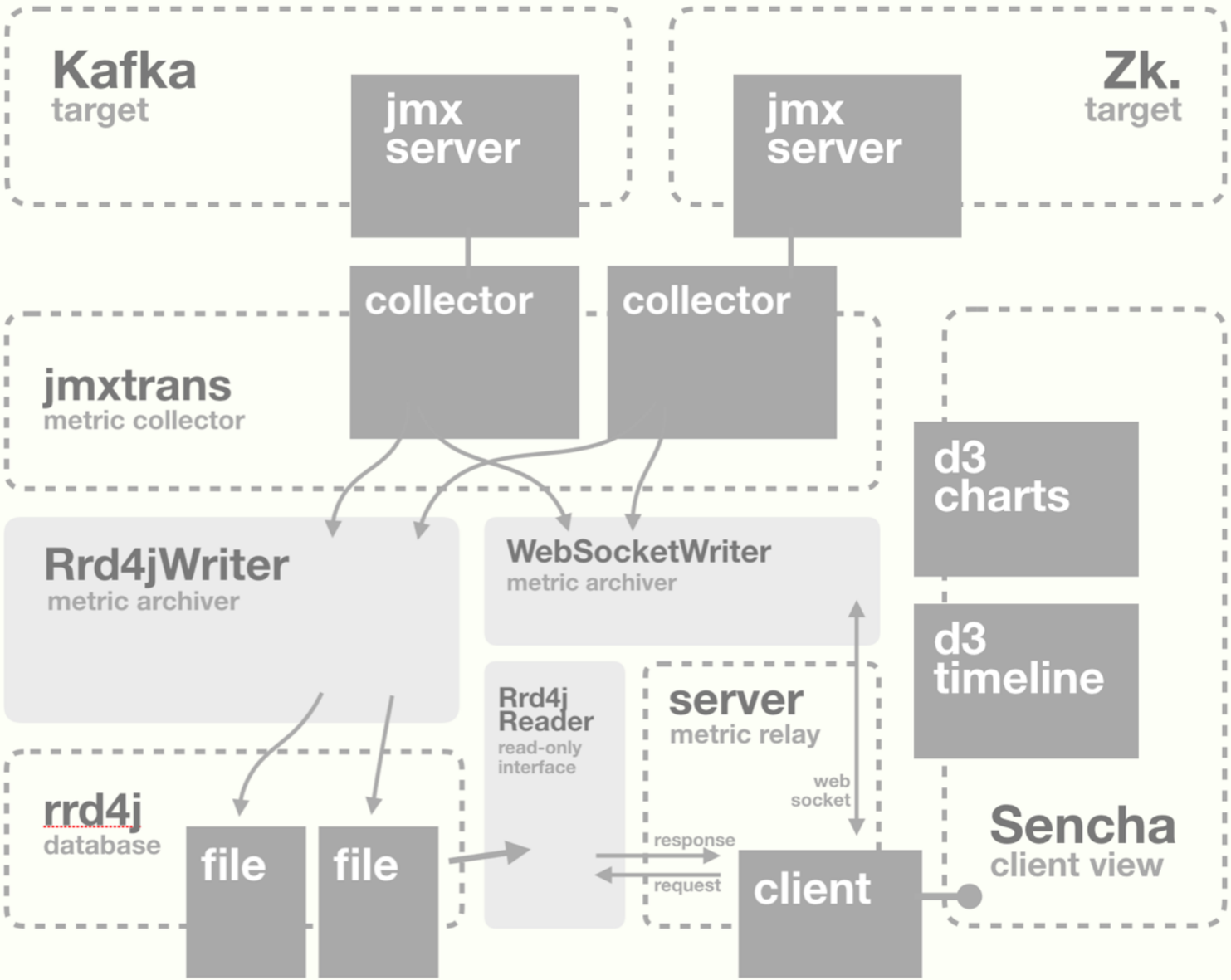
[B] WHAT IS ZOOKEEPER?

Safe storage for distributed systems

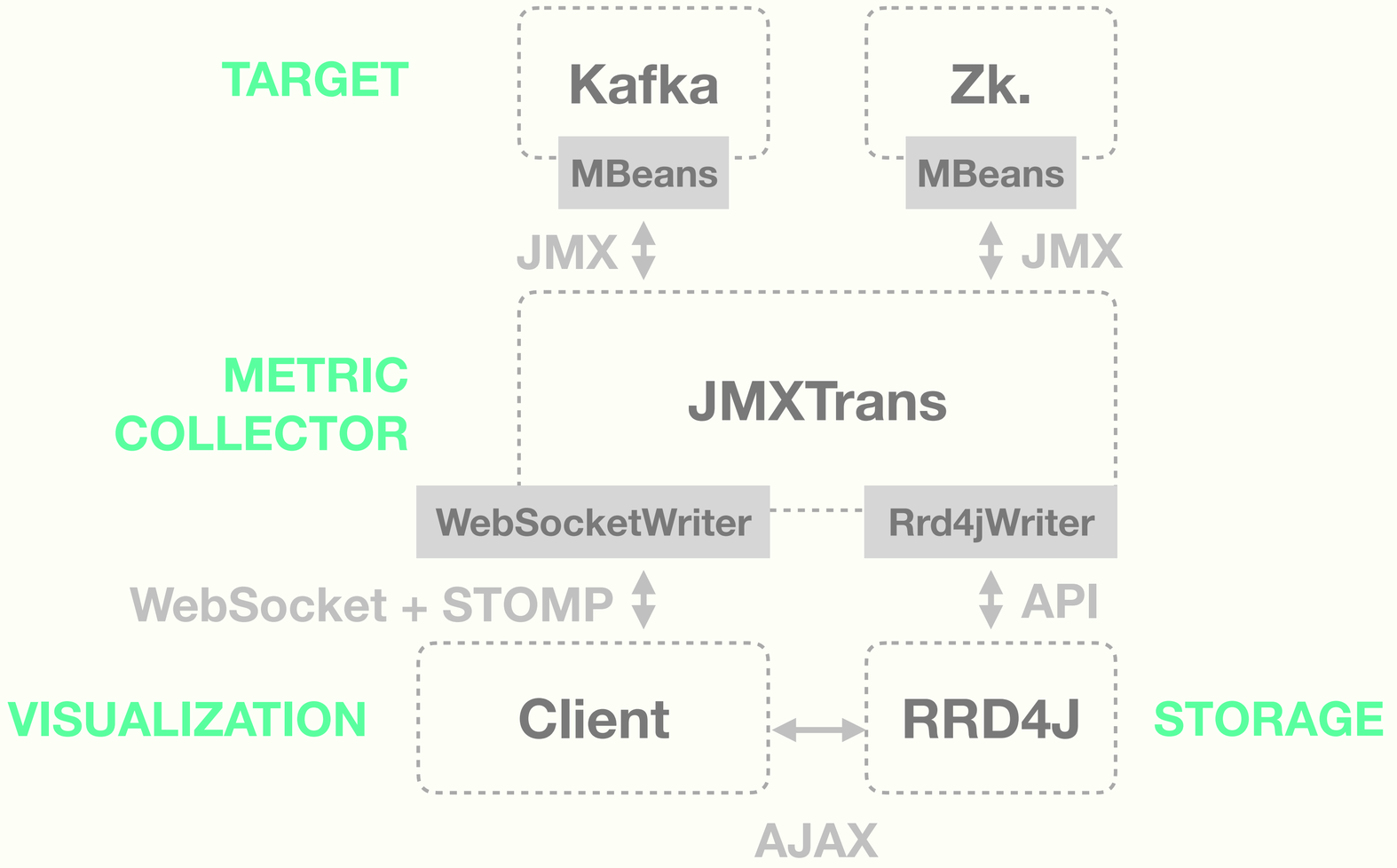
Four Features

Using name service to separate loads.
Using distributed lock to handle synchronization error
Error detection and recovery
Configuration management

_ Solution



_ ARCHITECTURE



_ Solution

Architecture

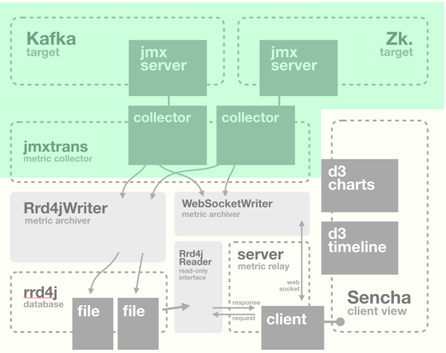
Metric Collection

Metric Storage

Communication

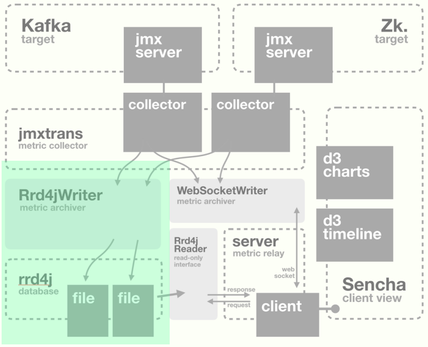
UI Design

_ METRIC COLLECTION



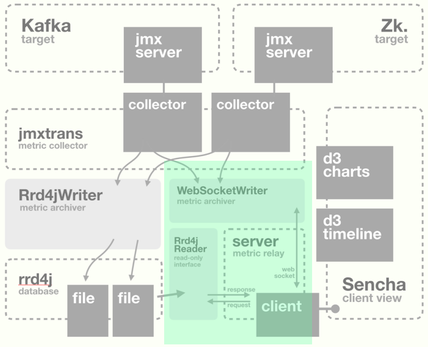
1. **JMX protocol is used to extract metrics from target system.**
2. **JMXTrans schedule collection job every 2 seconds.**
3. **Subprocess calls writer classes.**

_ METRIC STORAGE



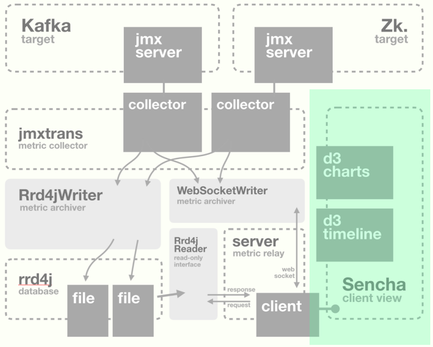
1. **JMXTrans calls Rrd4jWriter.**
2. **RRD4J saves metrics with several predetermined timescale.**
3. **RRD4J data is saved to a file.**

_ COMMUNICATION



1. **JMXTrans calls WebSocketWriter.**
2. **WebSocketWriter broadcasts data to all whom subscribes the topic.**
3. **Past data can be retrieved via AJAX call to RRD4J.**

_ UI DESIGN



1. **Sencha ExtJS is used as main framework.**
2. **SockJS and STOMP.js.**
3. **D3.js is used to draw charts.**

_ UI DESIGN: TWO NEEDS

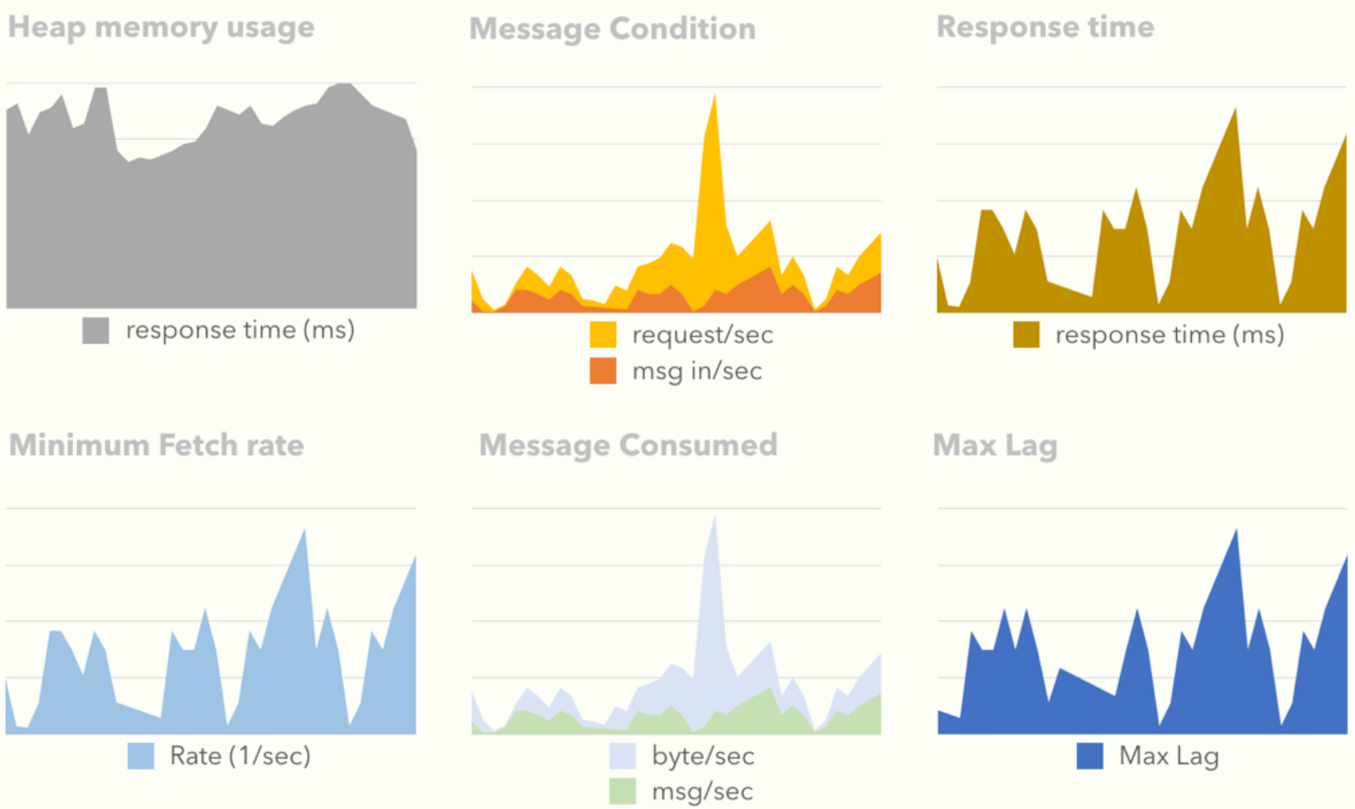
*To ensure the **normal operation** of the system*

*To find out the cause of **abnormal behavior***

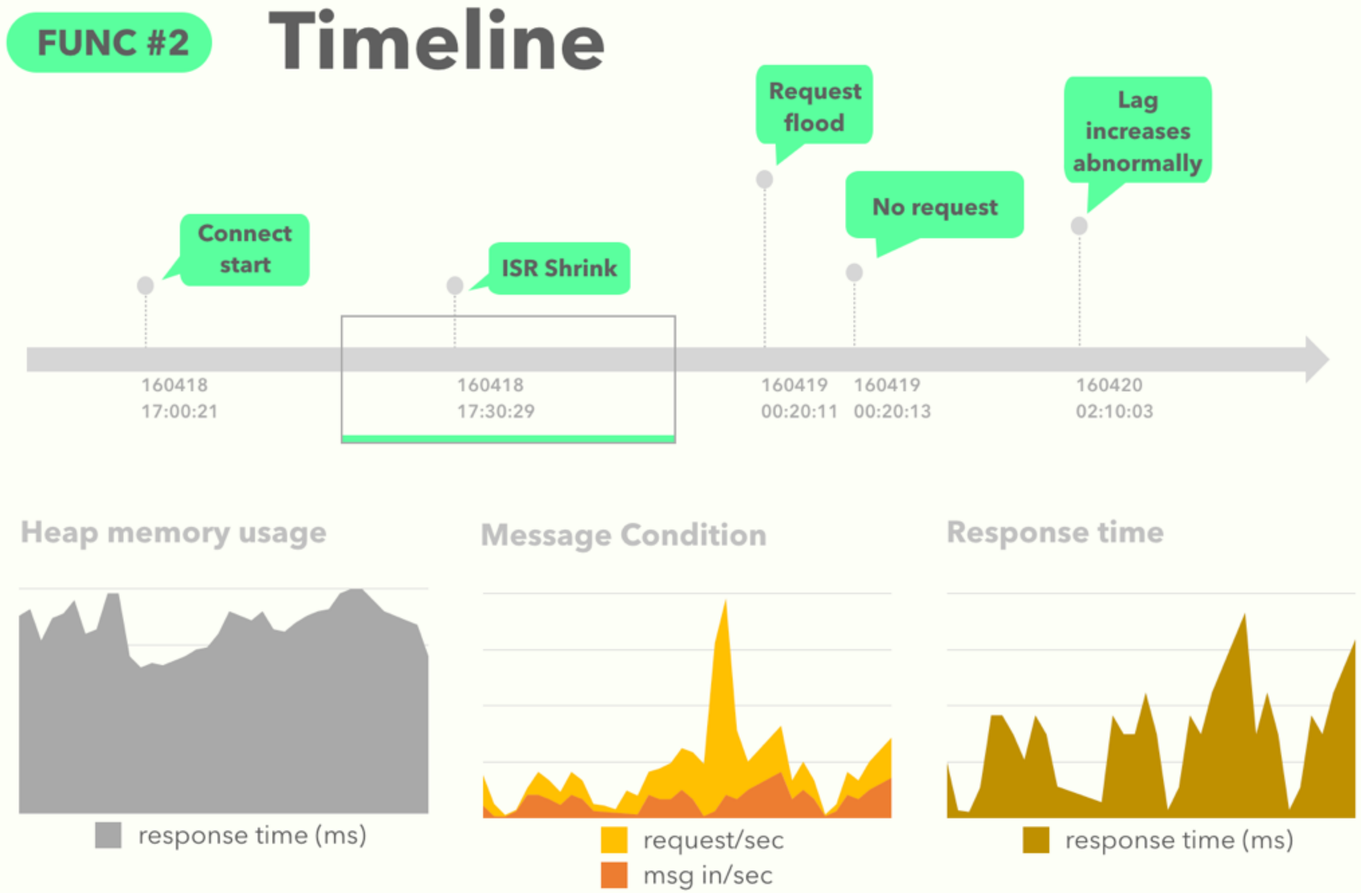
_ UI DESIGN: PAGE #1

FUNC #1

Overview



_ UI DESIGN: PAGE #2



PART_04

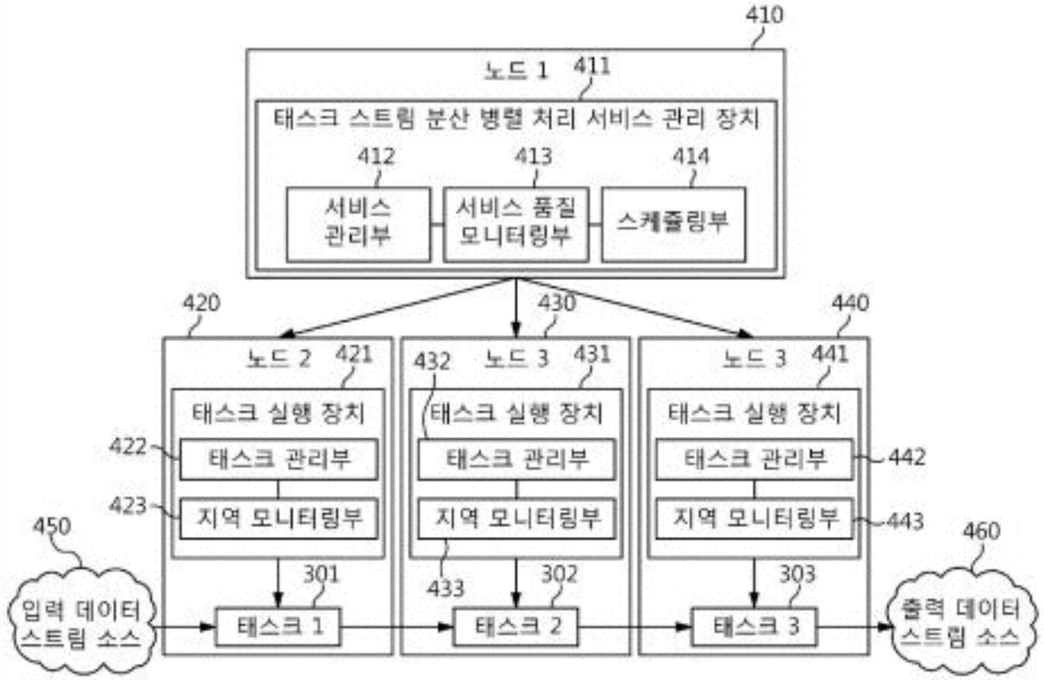
NOVELTY

_ PATENT RESEARCH

APPARATUS AND METHOD FOR MANAGING DATA STREAM DISTRIBUTED PARALLEL PROCESSING SERVICE

KR 2013-0095910 A

ETRI
Assignee



_ PATENT RESEARCH

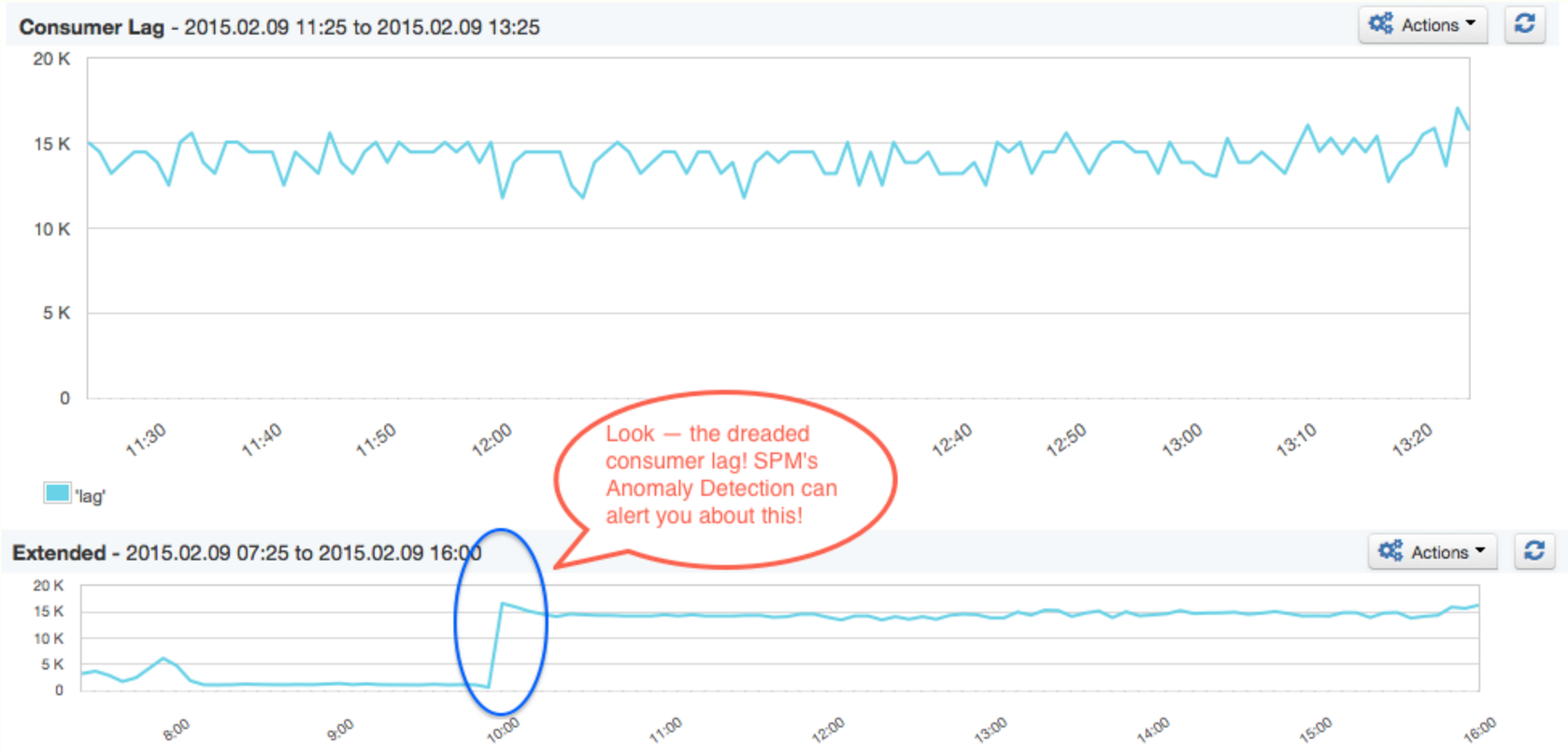
APPARATUS AND METHOD FOR ANALYZING BOTTLENECKS IN DATA DISTRIBUTED PROCESSING SYSTEM

KR 2015-0050689 A

SAMSUNG ELECTRONICS SEOUL NATIONAL UNIV.

Assignee

_ Sematext SPM



SPM KAFKA: CONSUMER LAG

_ Novelty

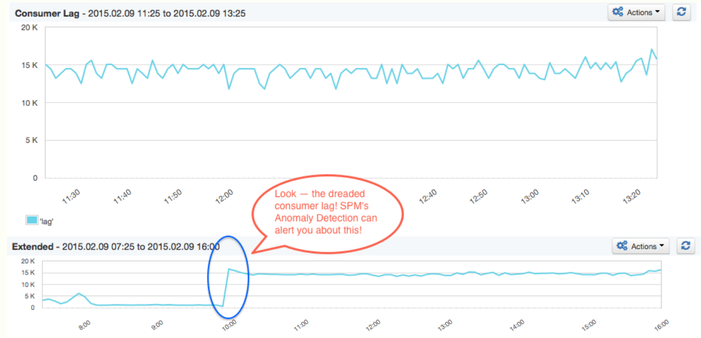
Patent Research

Sematext SPM

Kafka Offset Monitor

Comparison

_ Sematext SPM



1. **SPM alert user when abnormal event occurs via anomaly detection.**
2. **Provide abundant set of metrics: ~100 metrics are now being supported.**
3. **Integrated with Log Analyzer.**

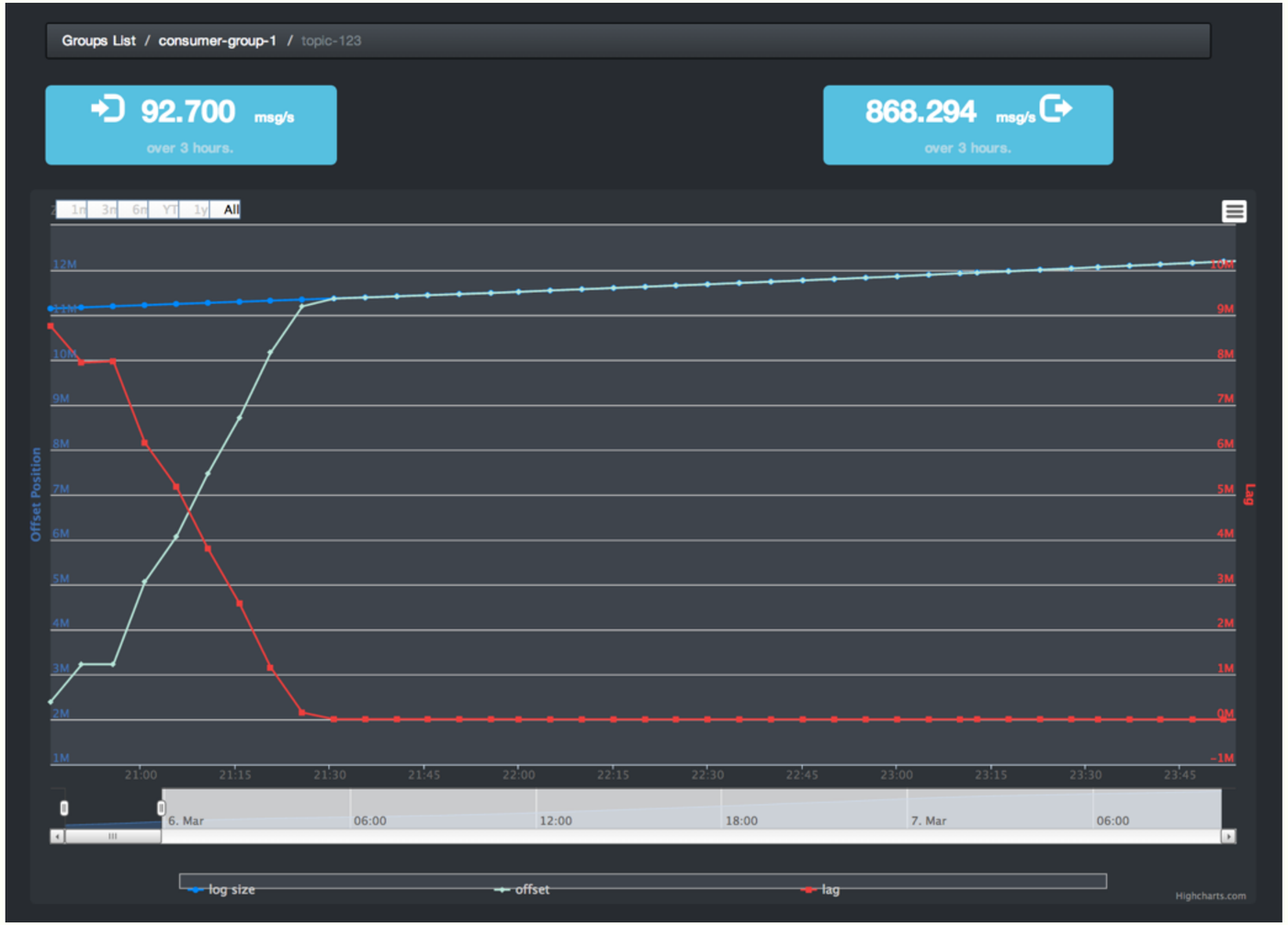
Patent Research

Sematext SPM

Kafka Offset Monitor

Comparison

_ Kafka Offset Monitor



_ Novelty

Patent Research

Sematext SPM

Kafka Offset Monitor

Comparison

_ Kafka Offset Monitor

1. **Concentrate on single metric: Offset Position of each topic.**
2. **The program also shows configuration of nodes participating in Kafka.**
3. **Built with python.**

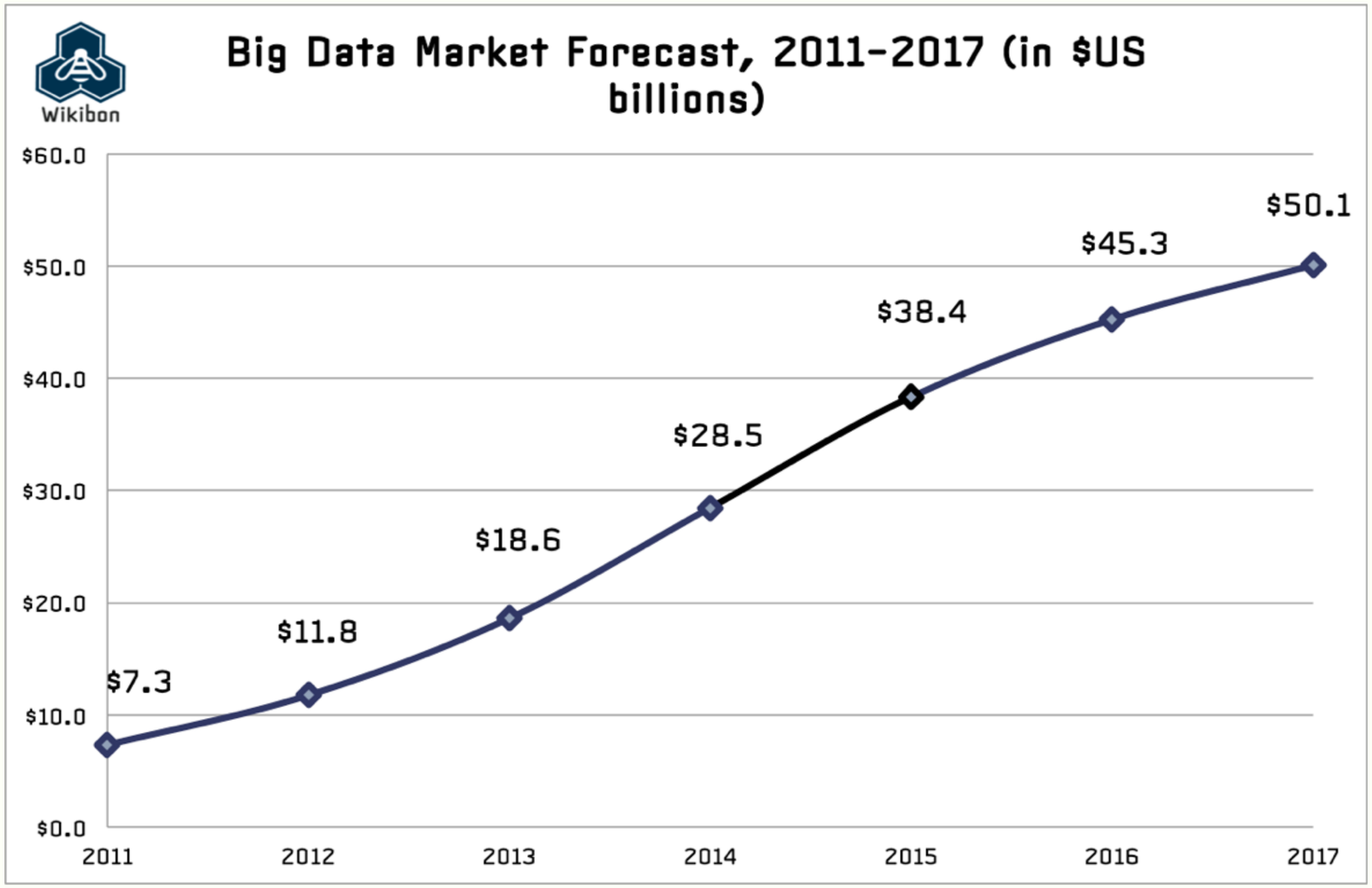
_ COMPARISON

Features	SPM Kafka	Kafka Offset Monitor	Doflamingo
Communicate with WebSocket?	×	×	O
Can view past trends?	×	×	O
Work with Flamingo?	×	×	O
Open Source?	×	O	O

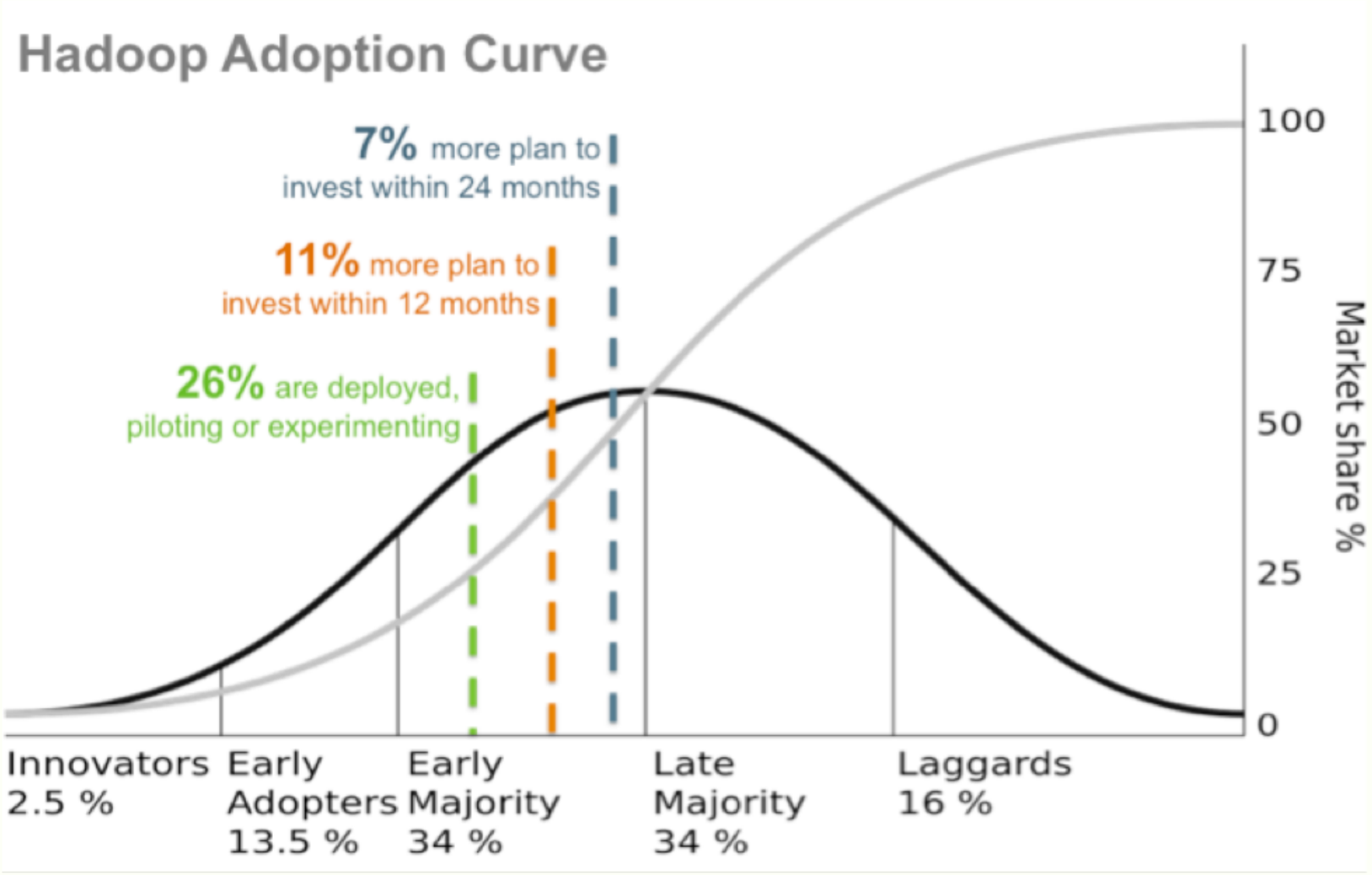
PART_05

CONTRIBUTIONS

_ TRENDS: $$ WITH BIG DATA



_ TRENDS: $$ WITH BIG DATA



_ OBSTACLES

**“ The biggest obstacle we’re running into is ”
not knowing what’s possible.**

Praveen Kankariya, the founder of Impetus Technologies

_ POSITIONING

SINGLE POINT APPROACH

Flamingo

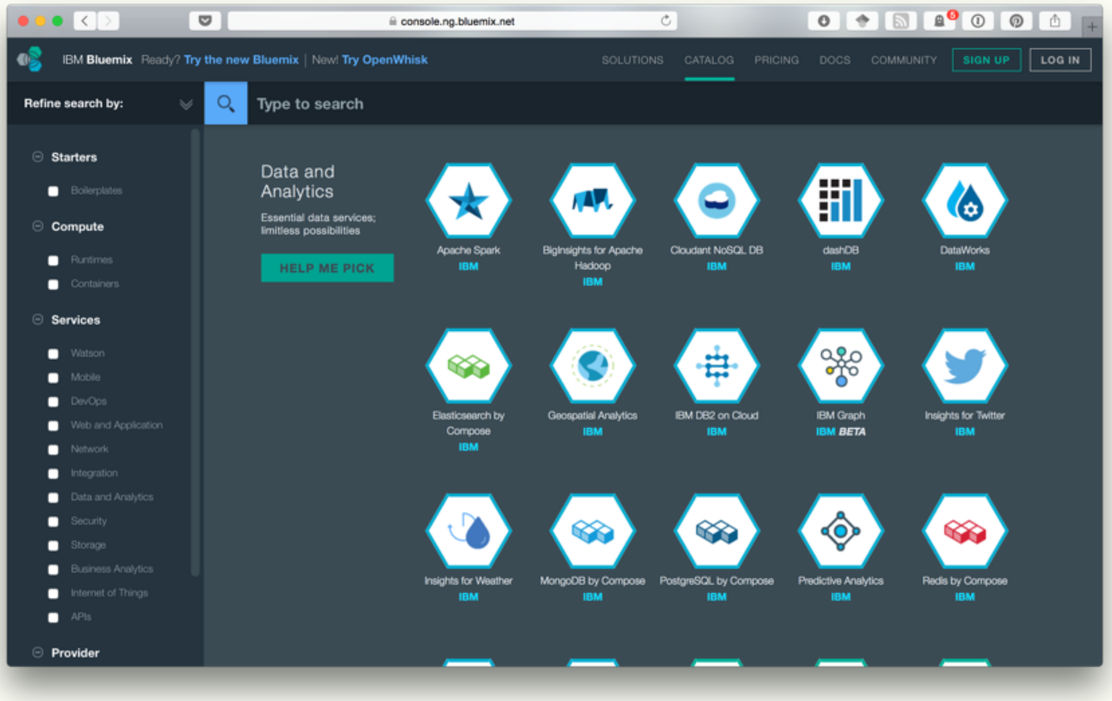
EXPERIMENT
PLATFORM

_ POSITIONING

Even a simple monitoring tool may be a great indicator to tell what can be done and what can’t be done.

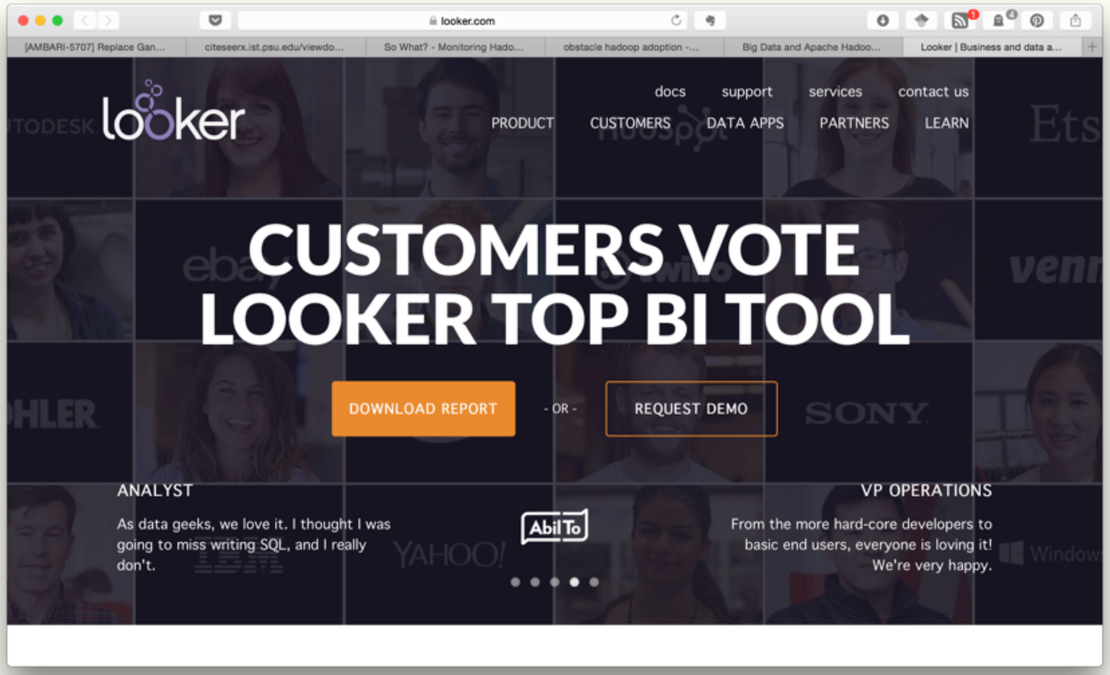
_ FUTURE

“Software as a Service”



_ FUTURE

“Extreme Abstraction”



PART_06

PROJECT MANAGEMENT

_ TEAM

TEAM _ ALPHADOOP

SEUNGHYO KANG *the hadoop master*

← **Metric Analysis**

RESTful Server →

JARYONG LEE *the spring master*

YOUNGJAE CHANG *the sencha master*

← **Visualization**

_ METHODOLOGY

AGILE APPROACH

1 SPRINT = 2 WEEKS

TOTAL 5 SPRINTS along the semester

_ OBJECTIVES

O1: Set up an environment for Flamingo

O2: Define Kafka measurement metrics, visualization forms

O3: Implement API server which provides collected metrics

O4: Implement charts with Sencha

O5: Integrate with Flamingo Ecosystem

O6: Define Zookeeper measurement metric, visualization

O7: Implement a Zookeeper monitoring module on Flamingo

SPRINT 1

SPRINT 2

SPRINT 3

SPRINT 4

SPRINT 5

KAFKA MODULE **M1**

ZOOKEEPER MODULE **M2**

_ OBJECTIVES

Objectives	Spaces
O1: Set up an environment for Flamingo	O
O2: Define Kafka measurement metrics, visualization forms	O
O3: Implement API server which provides collected metrics	O
O4: Implement charts with Sencha	O
O5: Integrate with Flamingo Ecosystem	X
O6: Define Zookeeper measurement metric, visualization	O
O7: Implement a Zookeeper monitoring module on Flamingo	X

Metric Analysis

Sprint#1~2 is for research, environment setup				M11. Requirement compliance			M12. Requirement traceability	
Collection step	Version	Date	Inspection time (min.)	UCR	ICP	ICT	(해당 단계) 요구사항 수	설계/코딩에 반영된 요구사항 수
SPRINT#3	v1a	4/27	-	2	0	0	3	1
	v1b	5/8	-	2	0	0	3	3
SPRINT#4	v2a	5/9	20	1	0	0	3	1
	v2b	5/16	20	1	1	1	3	3
SPRINT#5	v2a	5/23	20	0	0	0	3	1
	v2b	6/7	20	0	0	0	3	3

Kafka, ~~Zookeepr~~ JMX

Requirements clearly understood

Metric Analysis

				M13. Requirement change rate		M31. Test coverage	
Collection step	Version	Date	Inspection time (min.)	(이전 단계) Baseline 요구사항 수	변경된 요구사항 수	(해당 단계) 요구사항 수	요구사항 대비 테스트 통과 수
SPRINT#3	v1a	4/27	-	3	0	2	2
	v1b	5/8	-	2	1	3	2
SPRINT#4	v2a	5/9	20	3	0	3	1
	v2b	5/16	20	3	0	3	3
SPRINT#5	v2a	5/23	20	3	0	2	1
	v2b	6/7	20	3	0	2	2

Kafka, ~~Zookeeper~~

Metric Analysis

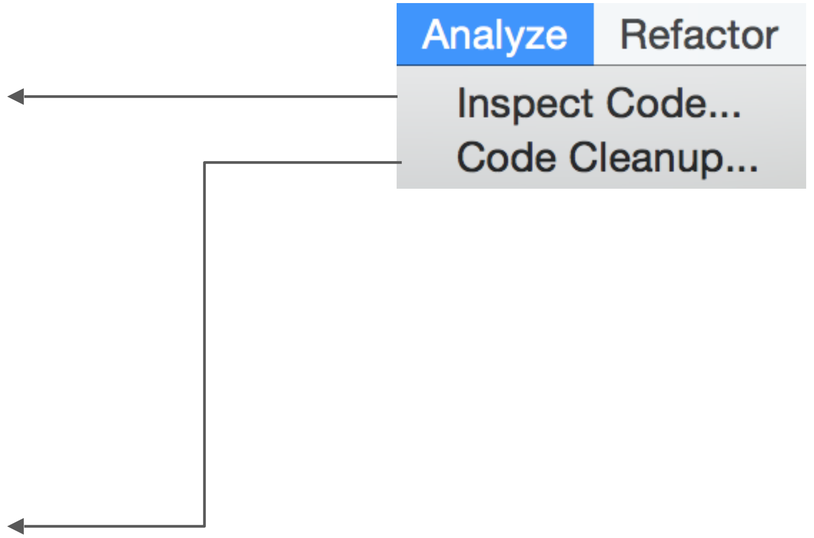
Not working code

				M21. Fault density		M22. Bad fix rate	
Collection step	Version	Date	Inspection time (min.)	결함 수	결함제거노력 (hour)	전체 결함 수	Side-effect 발생 수
SPRINT#3	v1a	4/27	-	0	0	0	0
	v1b	5/8	-	0	0		0
SPRINT#4	v2a	5/9	20	0	0	0	0
	v2b	5/16	20	0	0		0
SPRINT#5	v2a	5/23	20	0	0	0	0
	v2b	6/7	20	0	0		0

Metric Analysis

About 10,000 inspection points
➔ Because of extra library
(Ext.js, d3.js etc)

Only Performed Code Cleanup
➔ Reduced to 8,651



Requirements	Specified		Done
Built as a part of Flamingo system	-		O
Monitor and Report in Real-time	Implement Websocket writer		O
	Connect Websocket writer to JMX	Kafka	O
		Zookeeper	X
Utilize JVM ecosystem	-		O
Visualize the metrics, avoid numbers	Using d3.js, show metrics with graphs		O
Save metrics into Database	Implement RRD4j		O
	Connect RRD4j writer to JMX	Kafka	O
		Zookeeper	X
Special caution on log management	Timeline		O

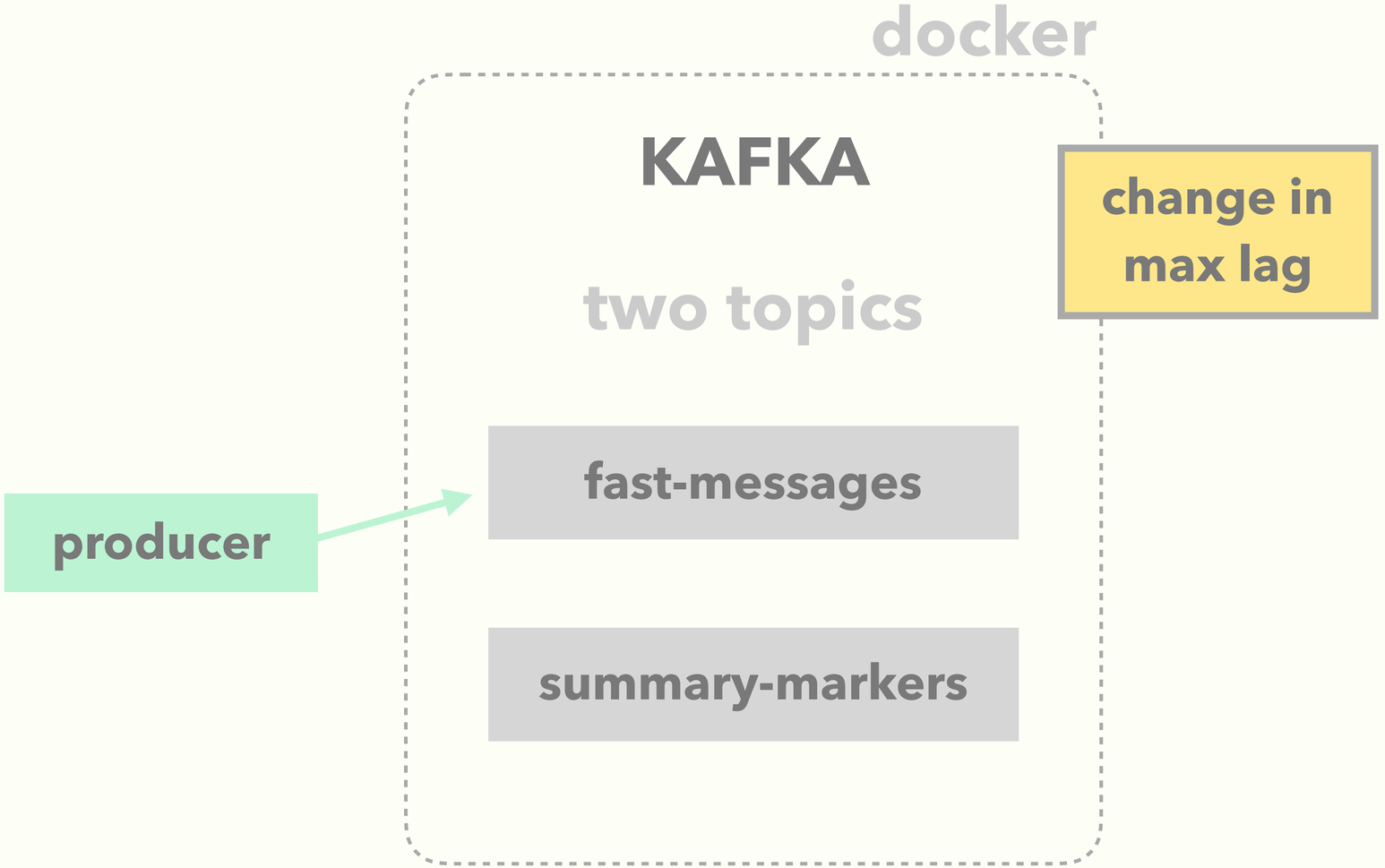
PART_07

DEMONSTRATION

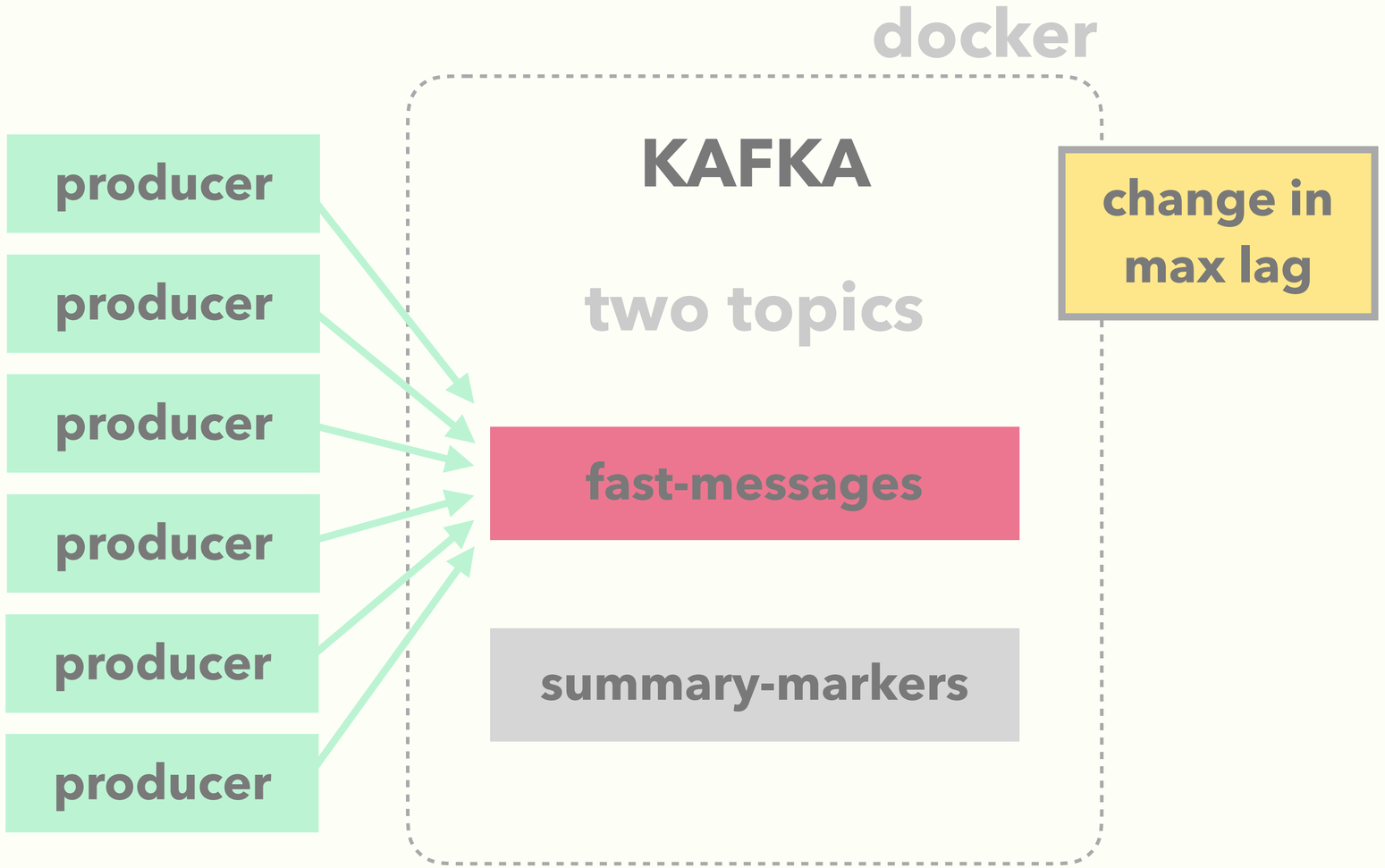
Scenario

Demo

_ SCENARIO



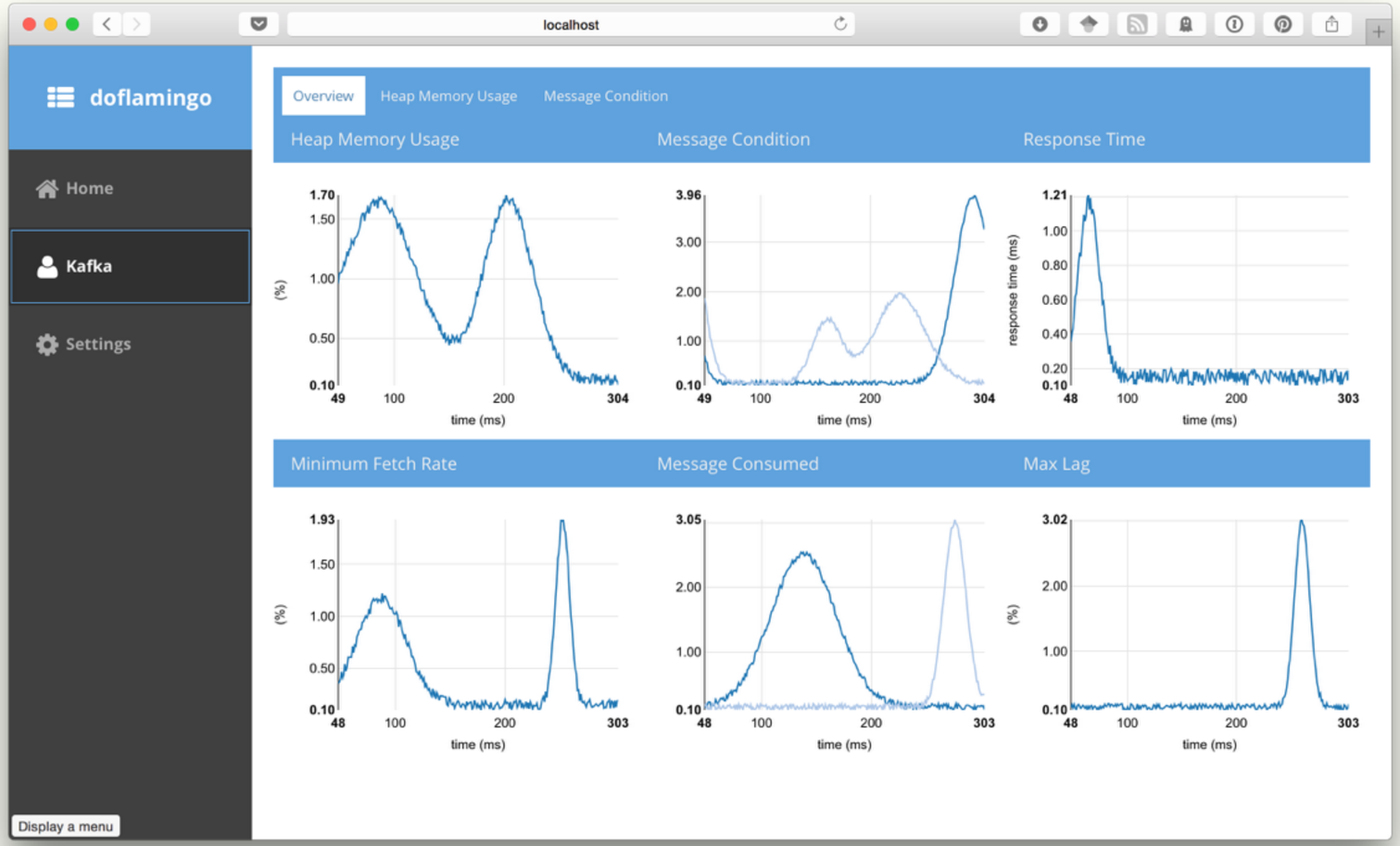
_ SCENARIO



_ DEMO

Scenario

Demo



THANK YOU
FOR LISTENING

END