

alphadoop

Doflamingo

An light-weight monitoring system for Apache Hadoop

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

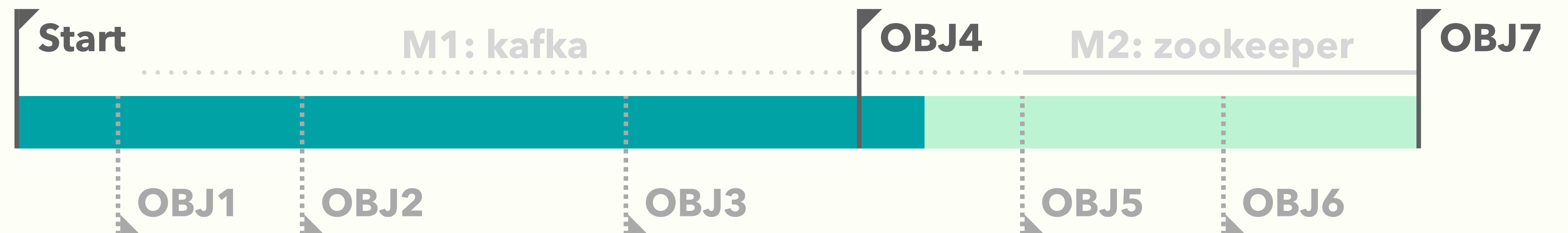
DURATION **March 13, 2016 ~ June 8, 2016**

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

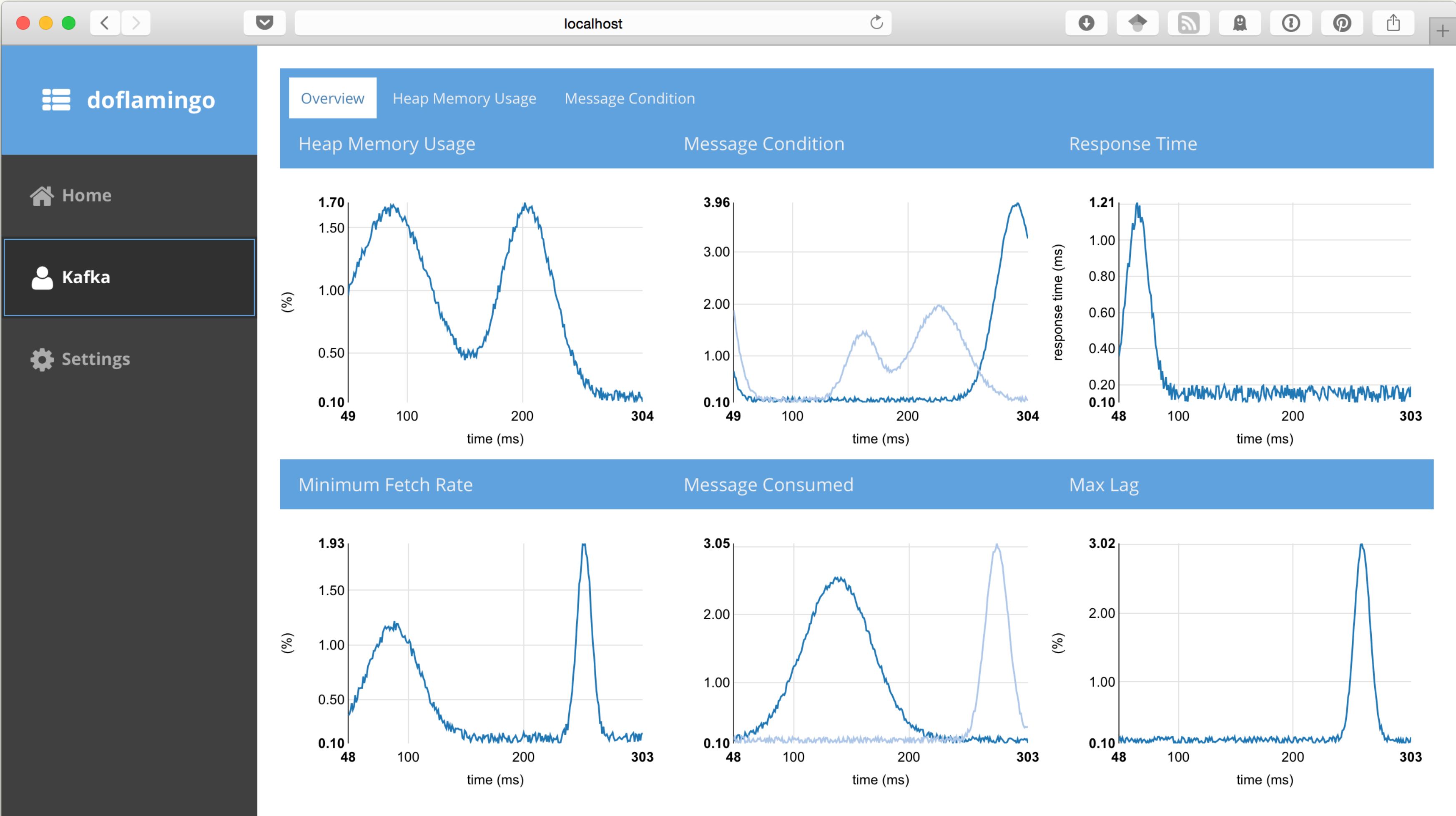
TIMELINE



OBJECTIVES



_ LOOK OF PROGRAM



PART_01

SOFTWARE REQUIREMENTS SPECIFICATION

_ WHAT WE WILL DO

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

_ WHAT WE WILL DO

Is all system working properly?



Doflamingo

Of Course!

Check this out!

_ WHY WE NEED THIS PROJ

1. Hard to understand Hadoop

Distributed system – not intuitive

Unable to track fluctuant mass traffic

Eyes on only the upper level

– run and hope everything goes well

_ WHY WE NEED THIS PROJ

2. The Missing Link of Flamingo

Currently flamingo is able to monitor:

- Resources
- YARN application
- Map Reduce
- Nodes

_ HOW WE DO IT

Learn from other monitoring tools

Plenty of tools exists in the field – Learn from them and try to build up similar metrics

Build it into flamingo platform

There's flamingo's way of monitoring hadoop system. Add a new task into jobscheduler.

_ HOW WE DO IT

AGILE APPROACH

1 SPRINT = 2 WEEKS

TOTAL 5 SPRINTS along the semester

_ REQUIREMENTS

- 1. Built as a part of Flamingo system**
- 2. Monitor and Report in Real-time**
- 3. Utilize JVM ecosystem**
- 4. Visualize the metrics, avoid numbers**
- 5. Save metrics into Database**
- 6. Special caution on log management**

_ OBJECTIVES

O1: Set up an environment for Flamingo

SPRINT 1

O2: Define Kafka measurement metrics, visualization forms

O3: Implement API server which provides collected metrics

SPRINT 2

O4: Implement charts with Sencha

O5: Integrate with Flamingo Ecosystem

SPRINT 3

O6: Define Zookeeper measurement metric, visualization

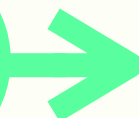
SPRINT 4

O7: Implement a Zookeeper monitoring module on Flamingo

SPRINT 5

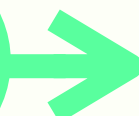
KAFKA MODULE

M1



ZOOKEEPER MODULE

M2

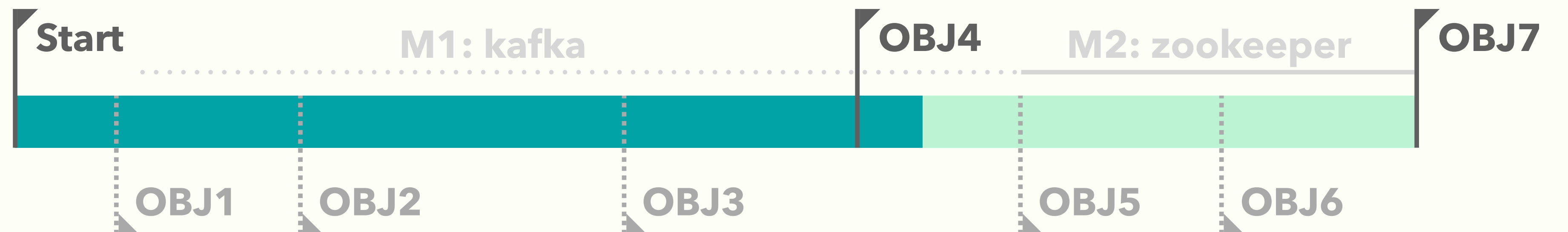


_ HOW FAR WE CAME

TIMELINE



OBJECTIVES

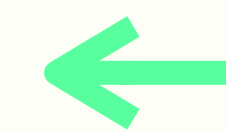


_ WHO WILL DO WHAT

TEAM _ALPHADOOP

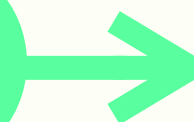
SEUNGHYO

KANG *the hadoop master*



Metric Analysis

RESTful Server

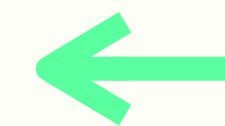


JARYONG

LEE *the spring master*

YOUNGJAE

CHANG *the sencha master*



Visualization

_ WE ARE RESPONSIBLE FOR:

1. Built as a open source software

Fork and request merge into flamingo

License/ Copyrights are same with flamingo

2. Bye-bye after spring semester

A/S are not supported after June 21, 2016

_ WE ONLY HAVE THESE:

LIMITED TIME: 10 WEEKS

No delay accepted – when semester ends,
project should be ended

LIMITED DEVELOPERS: 3 PEOPLE

No one will help us
– no money to hire someone!

PART_02

BACKGROUND RESEARCH

_ SAY HELLO TO MONITORING

Seeing is believing

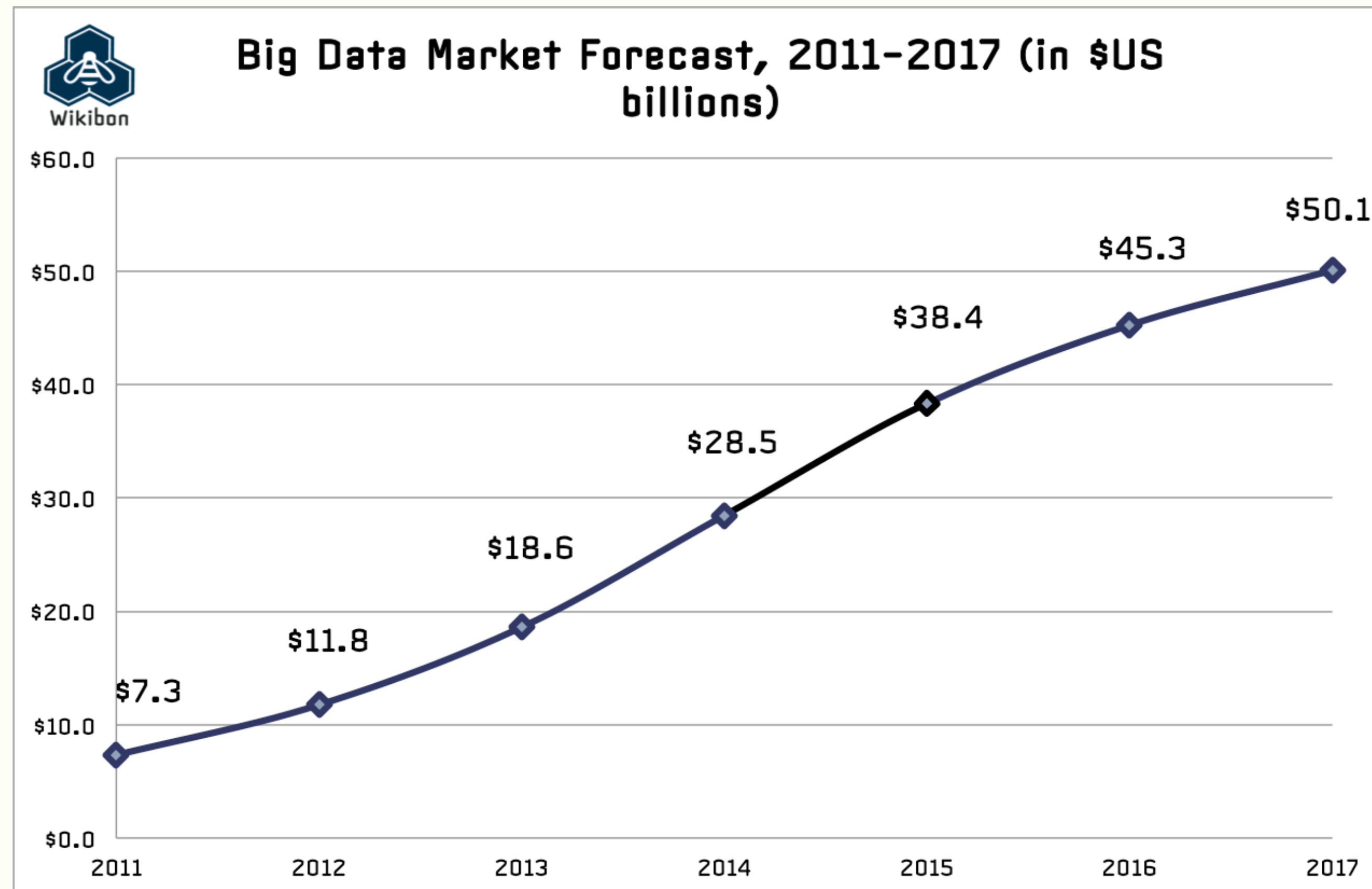
Software is intangible; so, where can we find it?

Bigdata: the buzz needs money

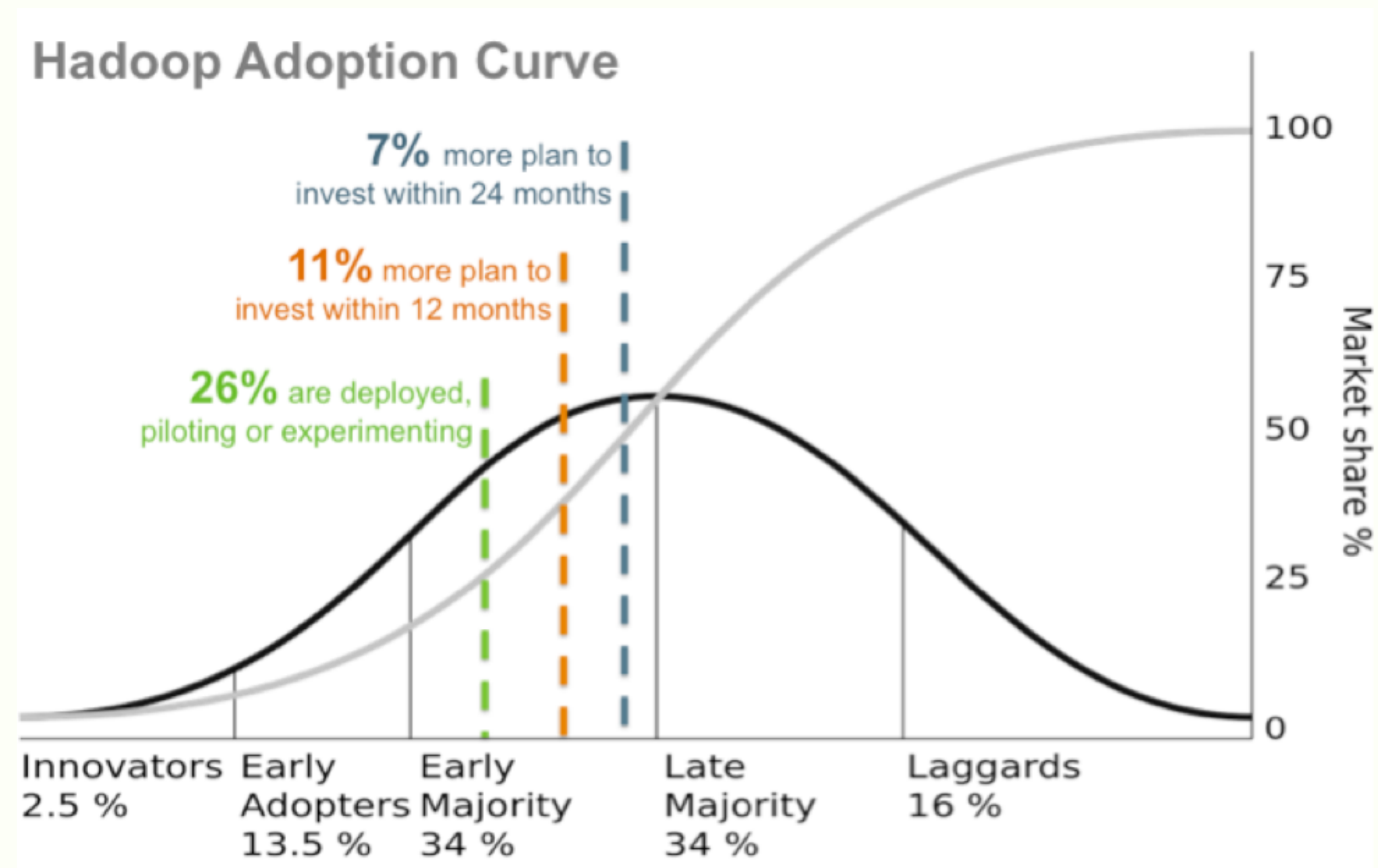
Hadoop is a money-eater:

10+ nodes, consulting, (expensive) engineers

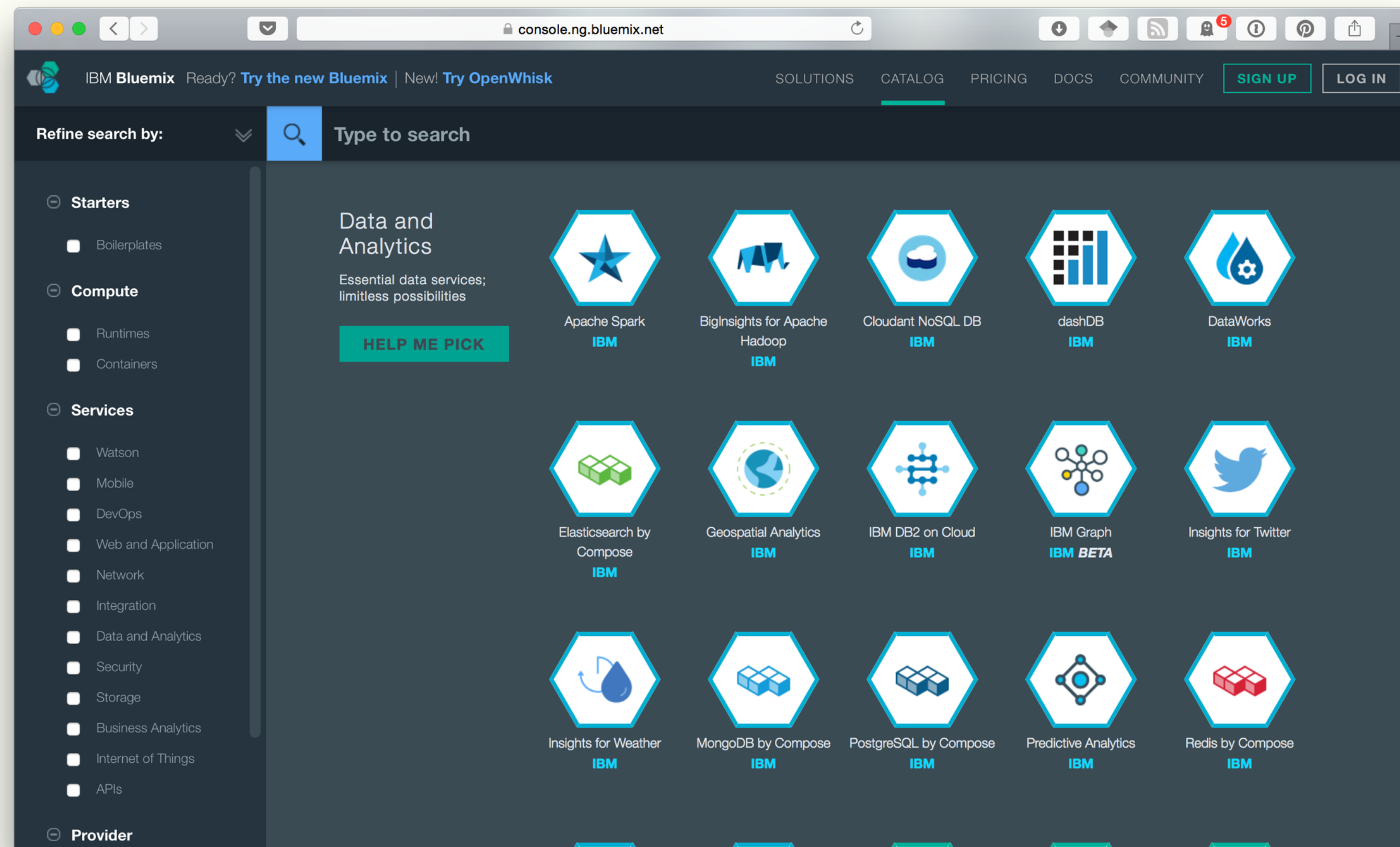
_ SAY HELLO TO MONITORING



_ SAY HELLO TO MONITORING



_ FUTURE OF FLAMINGO



Opensource
IBM
Bluemix

_ THE EFFECT OF OUR WORK

The ultimate control tower

Flamingo now monitors not only nodes, but also modules that compose pipeline.

Opening up new possibility

The gathered metrics can be used for further optimization or anomaly detection feature.

_ 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

[A] WHAT IS KAFKA?

Kafka consists of producer, broker, and consumer, managed by **Zookeeper**

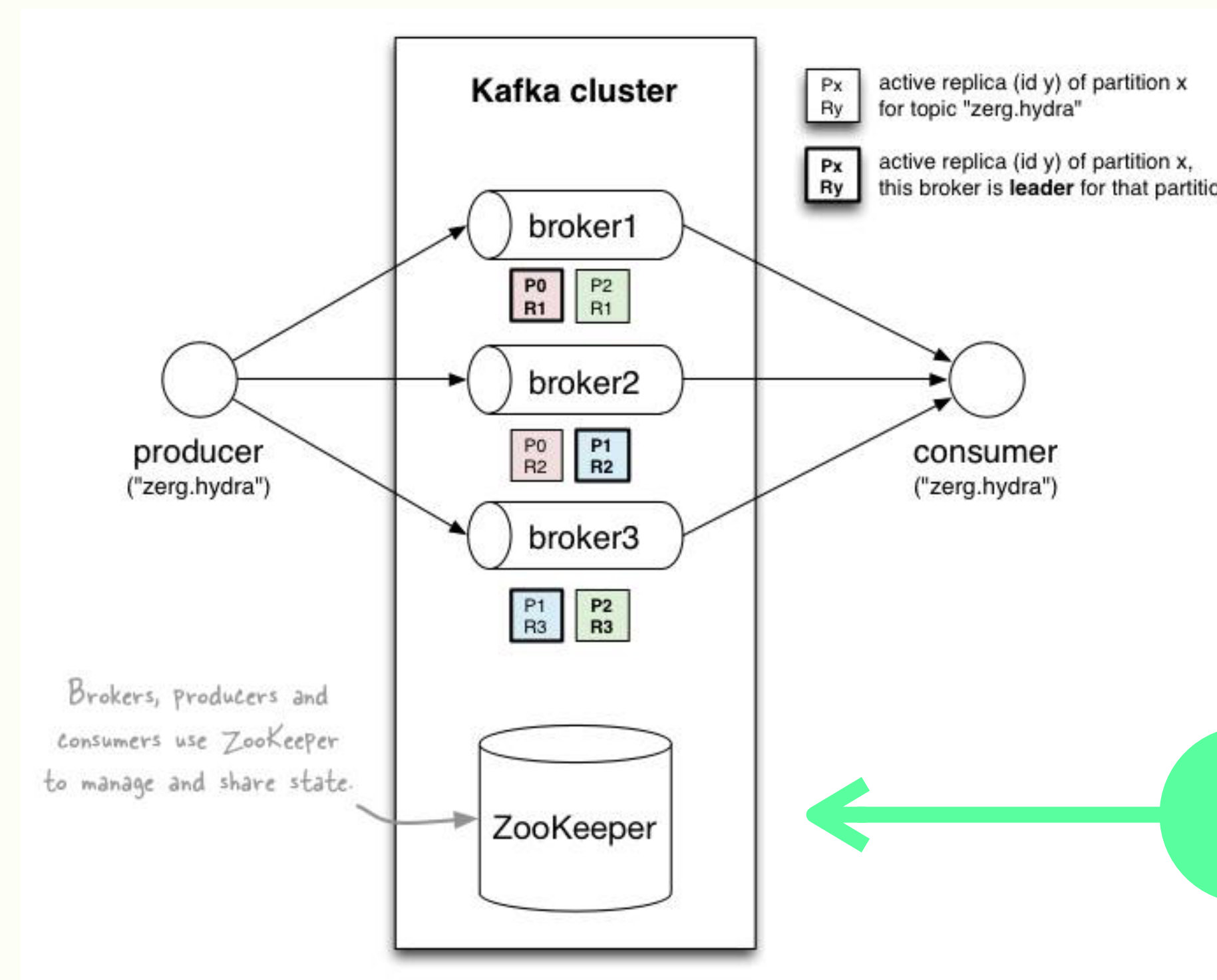
Producers send system messages to **brokers**

Brokers process them **distributively**

Consumers store the messages to their **disks**

_ TECHNICAL DETAILS

[A] WHAT IS KAFKA?



producer

generate message that fall into certain topics

consumer

subscribe specific topics & process the message

broker

stacks up logs based on topic

_ TECHNICAL DETAILS

[A] WHY KAFKA?

Store the messages in the **DISK**, not in the cache.

Consumers can rewind back to old data and re-consume them since they are in the disk for a certain period of time.

PULL model, not push model

consumer pull messages from broker without exceeding their limit; no drop occurs unlike producer-push model

Summary
Background
Deep cuts
Thoughts
Realization
Silver-lining

_ 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

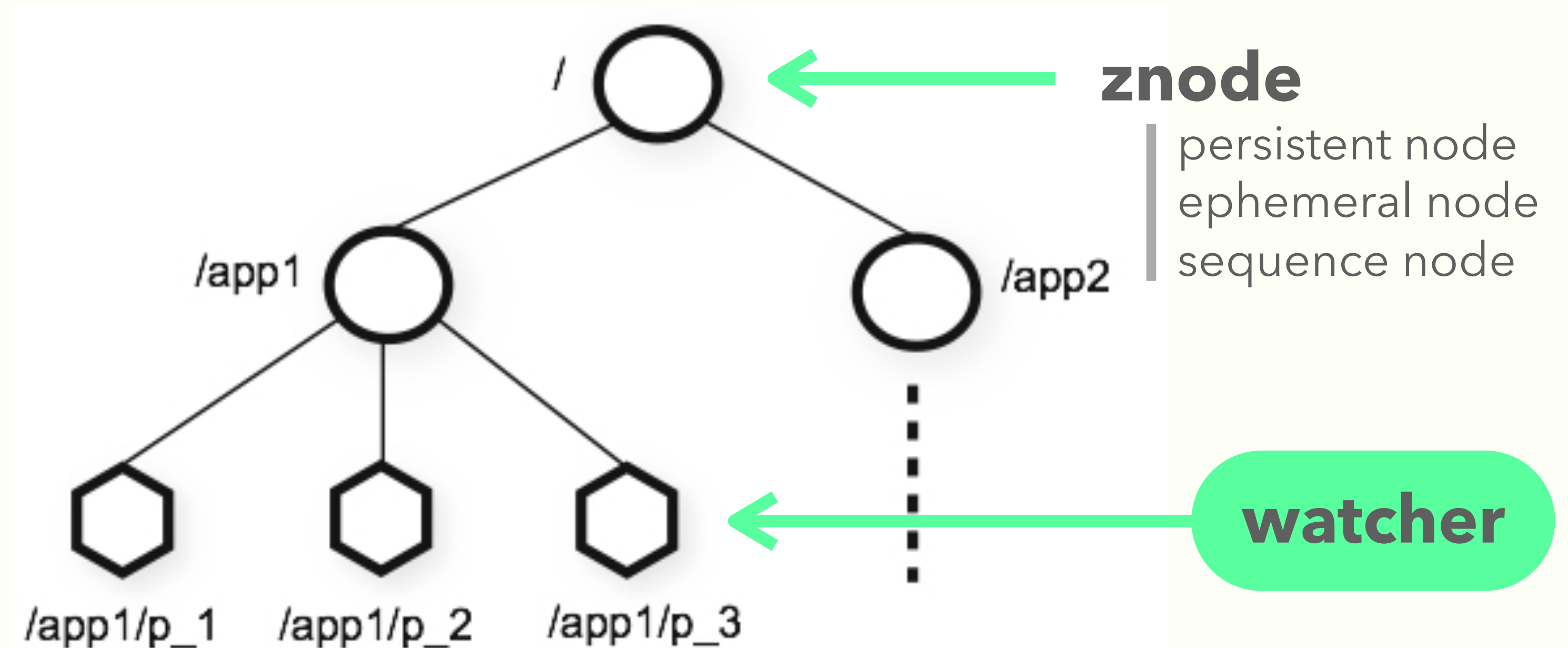
Error detection and recovery

Configuration management

Summary
Background
Deep cuts
Thoughts
Realization
Silver-lining

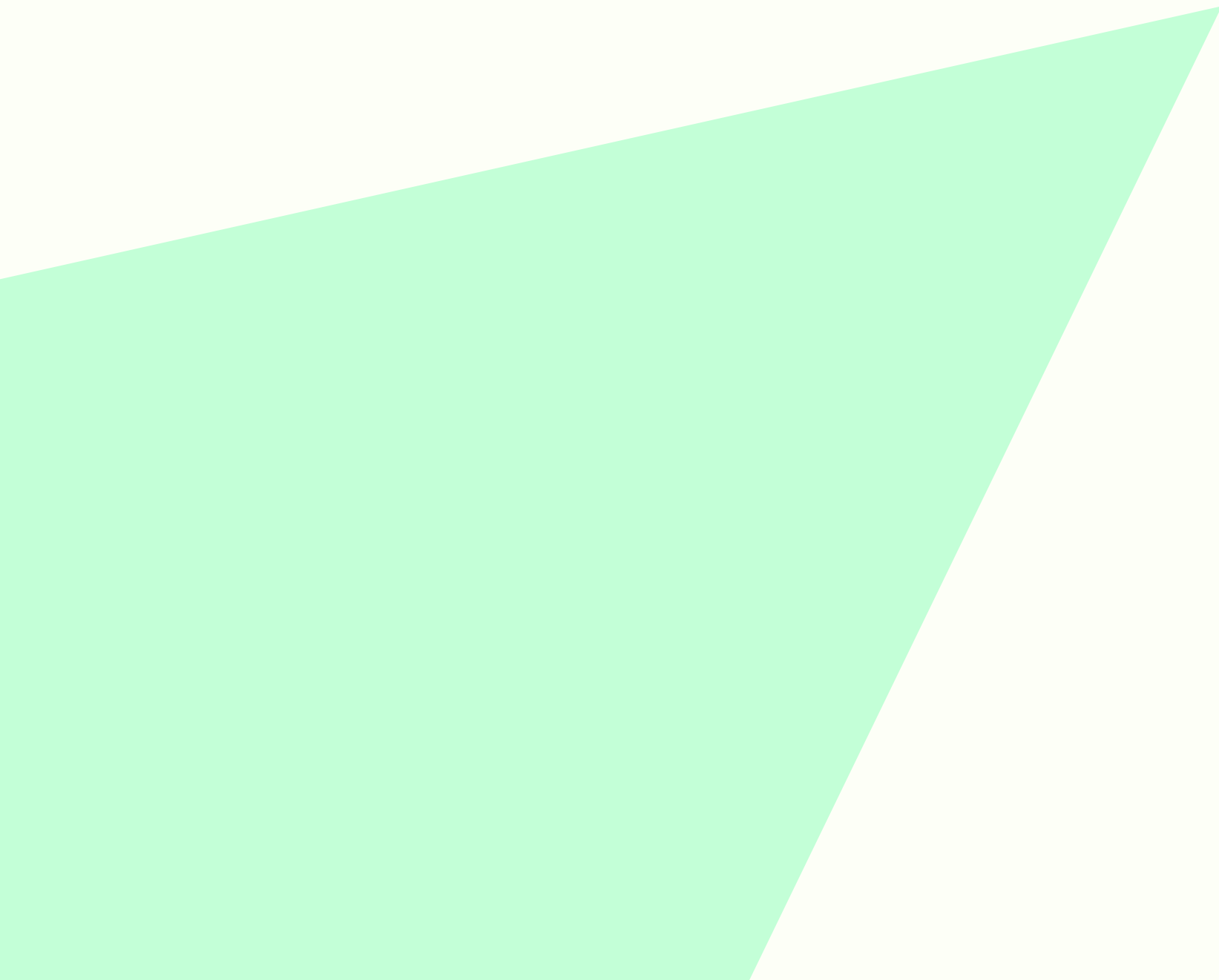
_ TECHNICAL DETAILS

[B] WHAT IS ZOOKEEPER?



PART_03

USER SCENARIO DESIGN



Overview

Users

Problems

Solutions

Novelty

Scenario

Schedule

_ QUESTION

PHASE #1

What is a monitoring?

PHASE #2

Why do we monitor?

Overview

Users

Problems

Solutions

Novelty

Scenario

Schedule

_ TWO NEEDS

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

*To find out
the cause of
abnormal
behavior*

Overview

Users

Problems

Solutions

Novelty

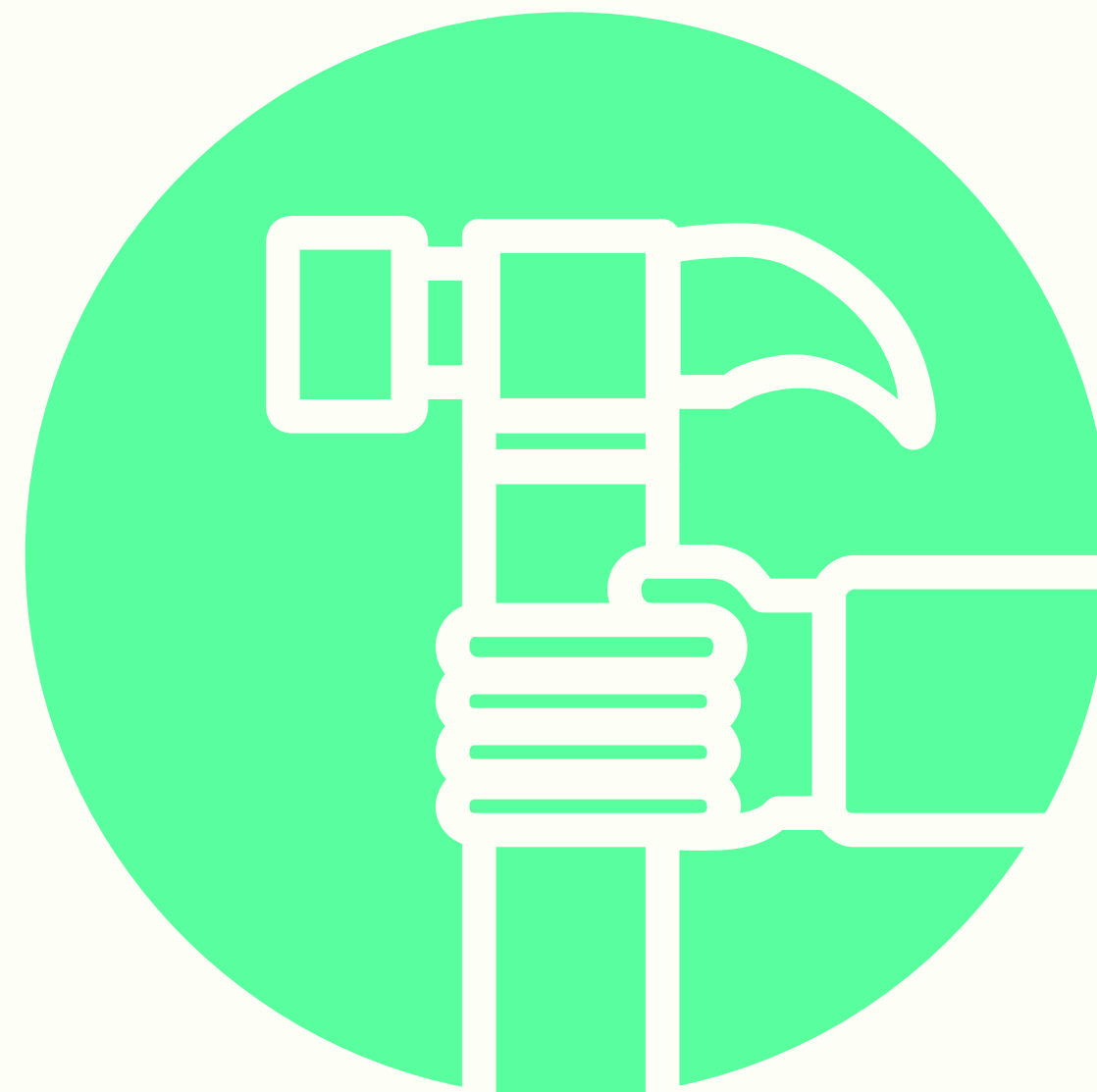
Scenario

Schedule

_ TWO USERS

USER #1

Administrator



USER #2

Engineer



Overview

Users

Problems

Solutions

Novelty

Scenario

Schedule

_ USERS

USER #1

Administrator

- A. Hope everything stays normal
- B. Determine whether to put more resources or not
- C. Usually maintains a volume of system
- D. Focus on real-time data

USER #2

Engineer

- A. Fix the problem
- B. Find out the cause of the problem by traveling the past data
- C. Deeper understanding on whole system
- D. Focus on specific events

Overview

Users

Problems

Solutions

Novelty

Scenario

Schedule

_ DIFFERENT REQUIREMENTS

USER #1

Administrator

- A. Visualize constantly changing statistics of sys.
- B. At a glance view of metrics
- C. Real-time update without user intervention

USER #2

Engineer

- A. Visualize abrupt events
- B. Can travel back to the past to find the cause of event
- C. Detailed analysis on changing variables during specific timeframe

Overview

Users

Problems

Solutions

Novelty

Scenario

Schedule

_ EXTERNAL INTERFACE

FUNC #1

Overview

A. Dashboard

B. Configuration

FUNC #2

Timeline

A. Event Timeline

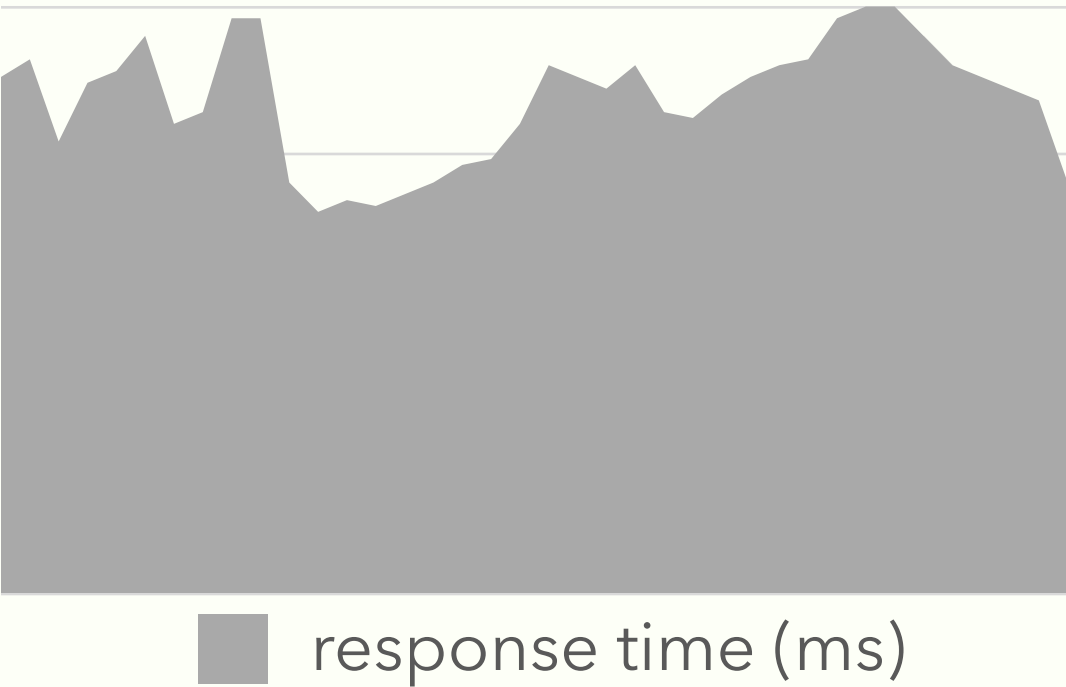
B. Timemachine

- Overview
- Users
- Problems
- Solutions**
- Novelty
- Scenario
- Schedule

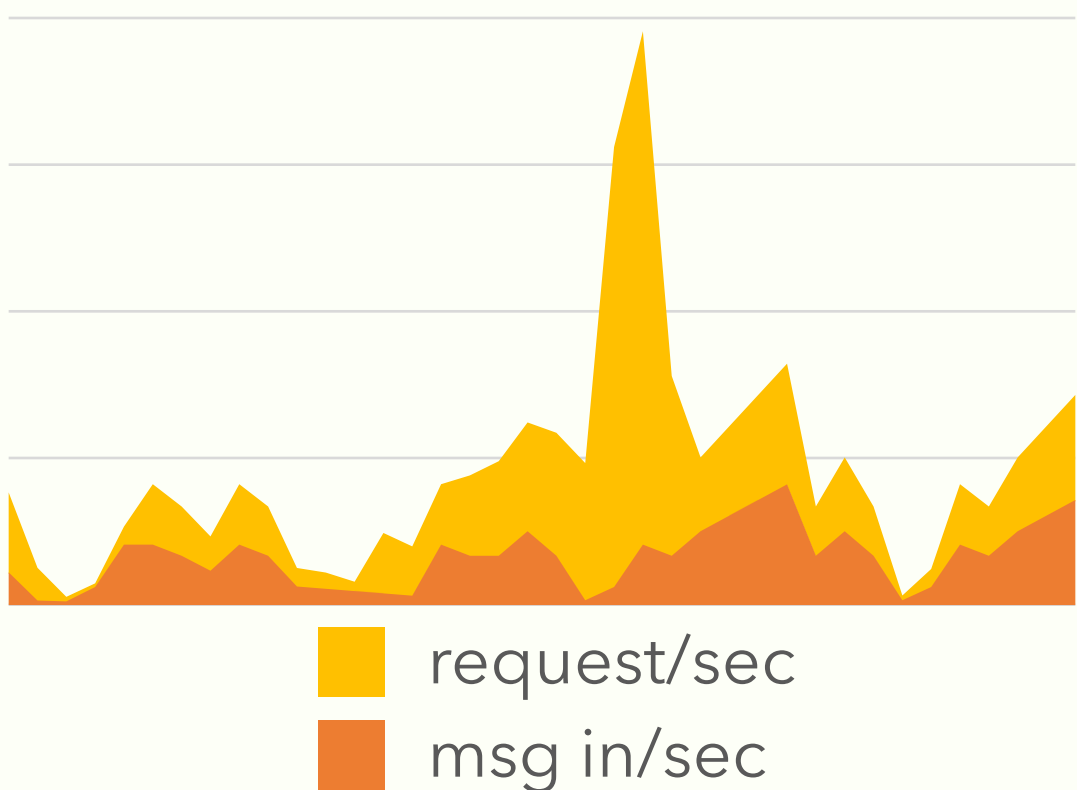
FUNC #1

Overview

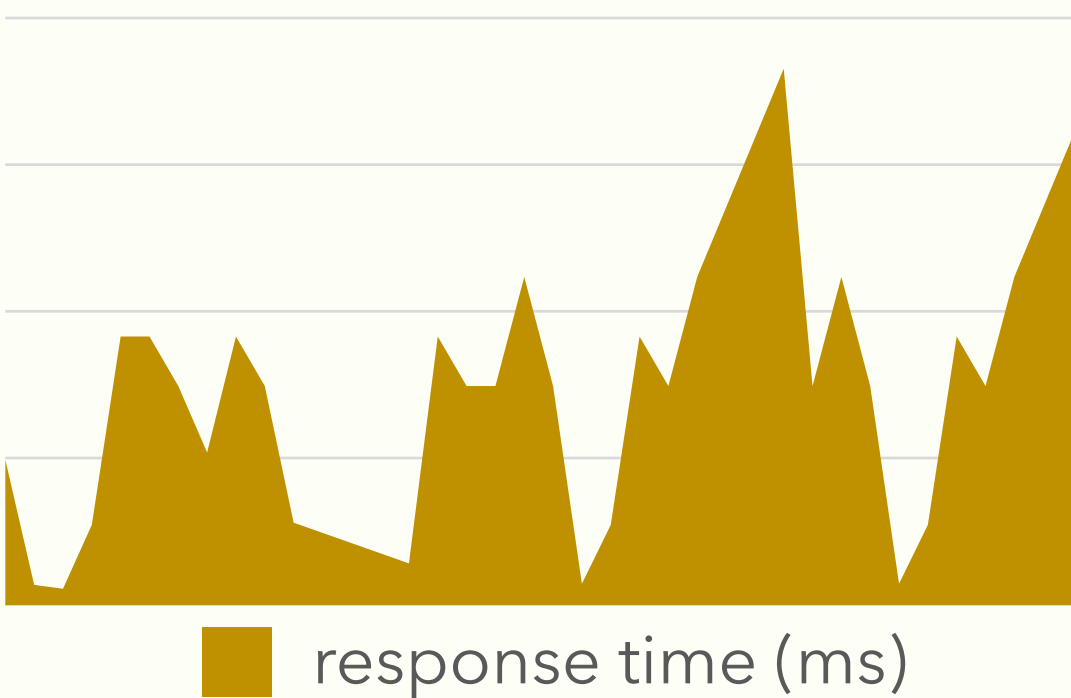
Heap memory usage



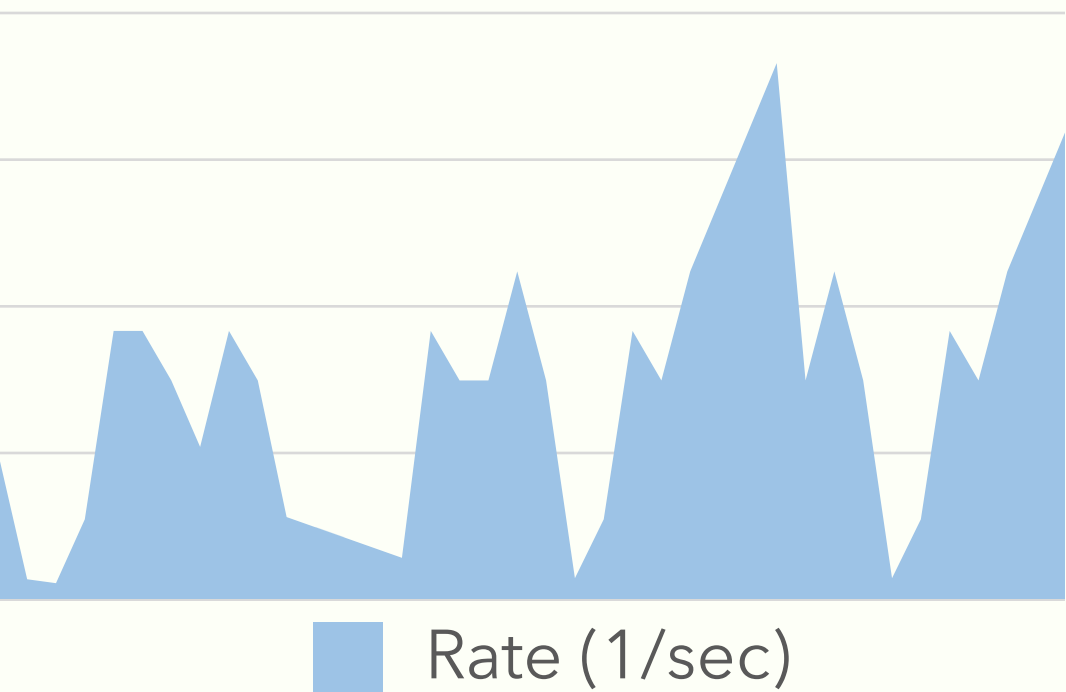
Message Condition



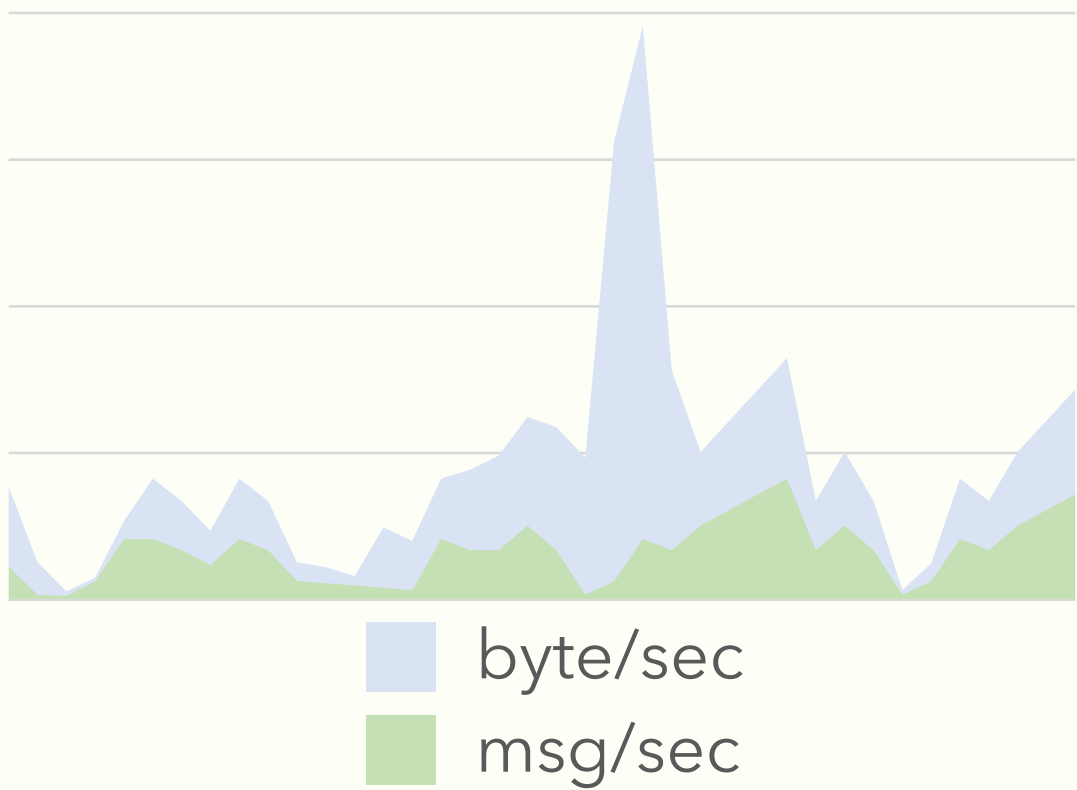
Response time



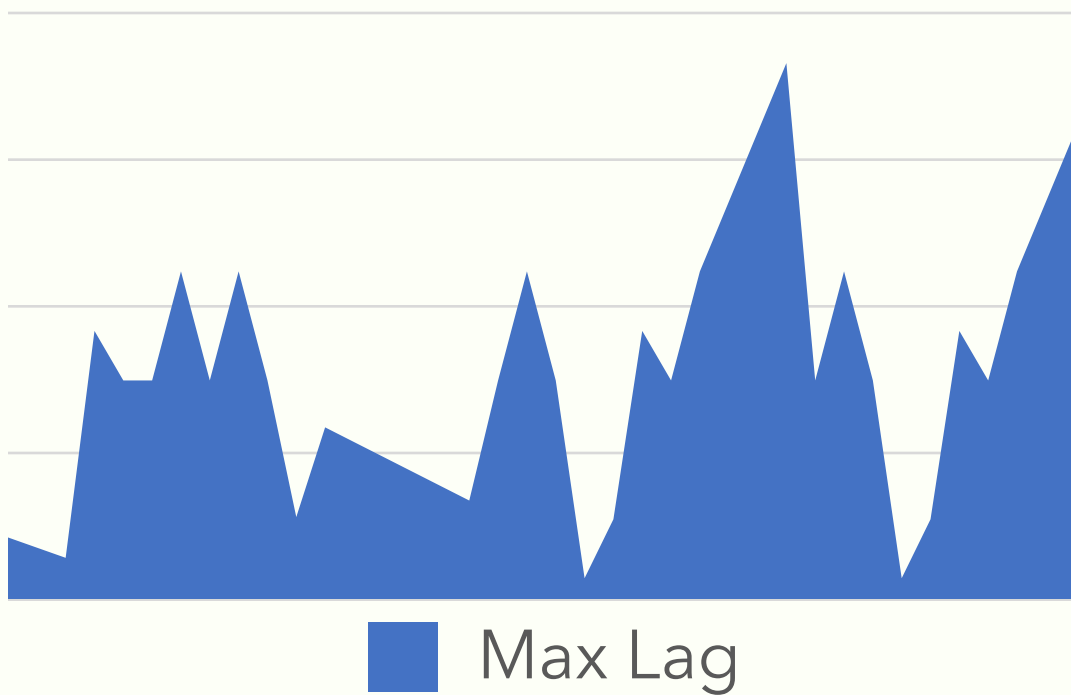
Minimum Fetch rate



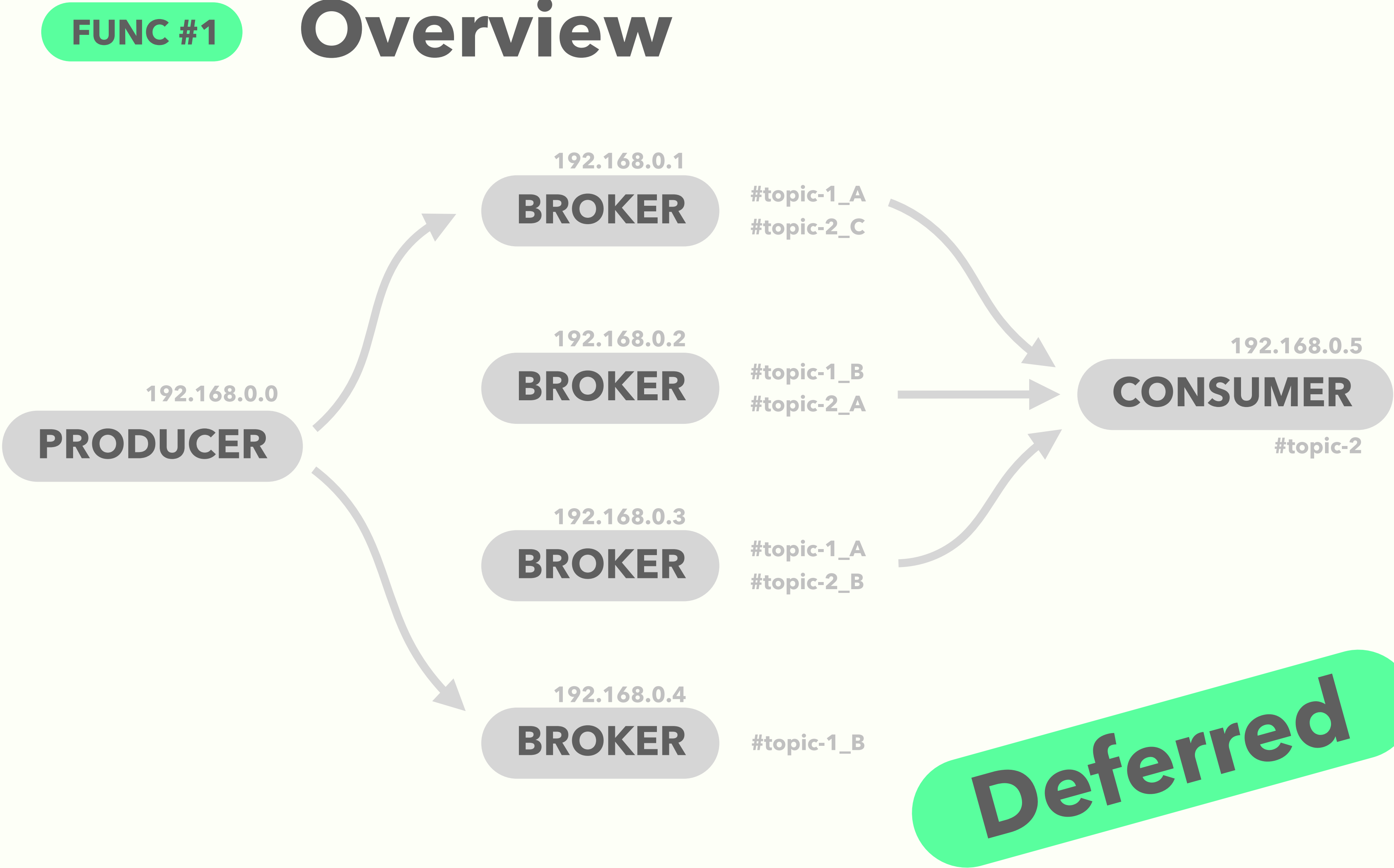
Message Consumed



Max Lag



- Overview
- Users
- Problems
- Solutions**
- Novelty
- Scenario
- Schedule



Overview

Users

Problems

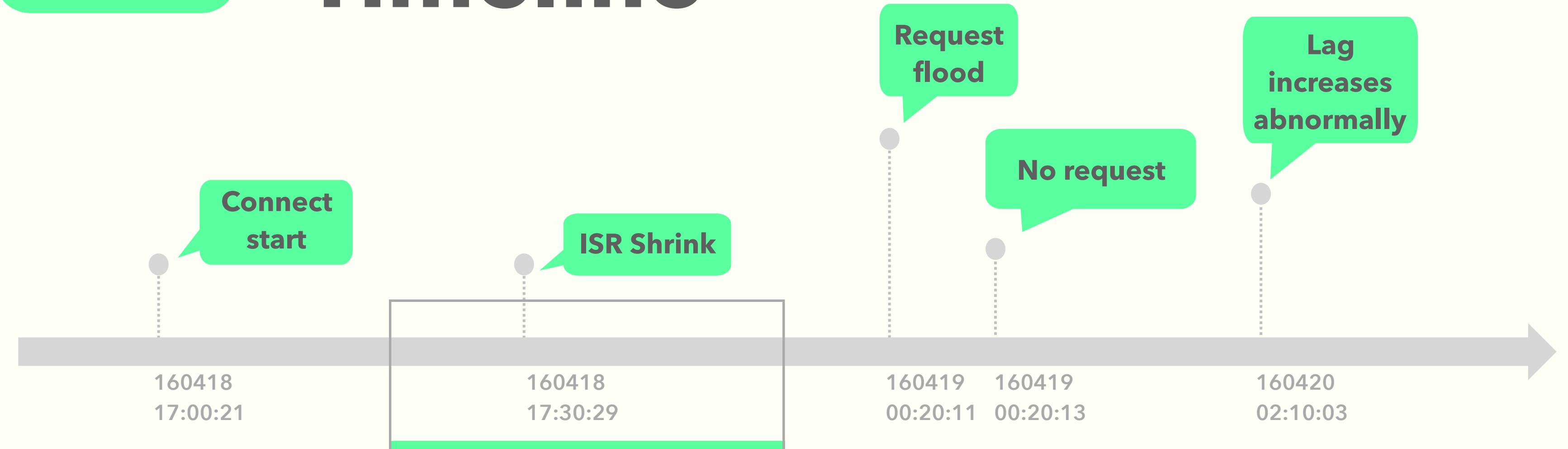
Solutions

Novelty

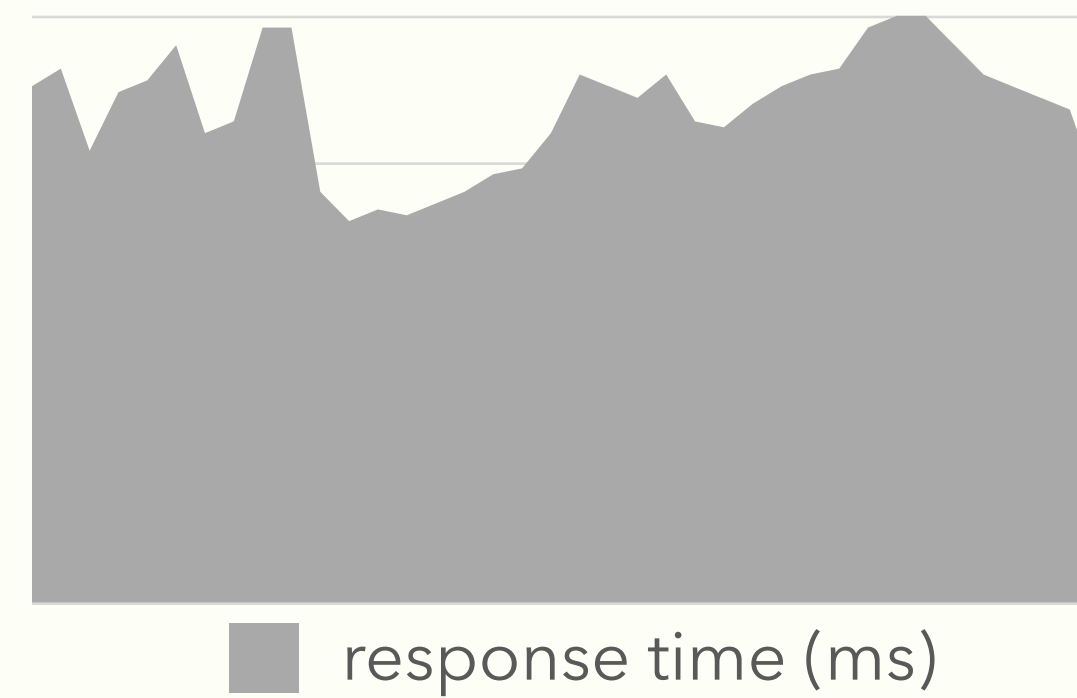
Scenario

Schedule

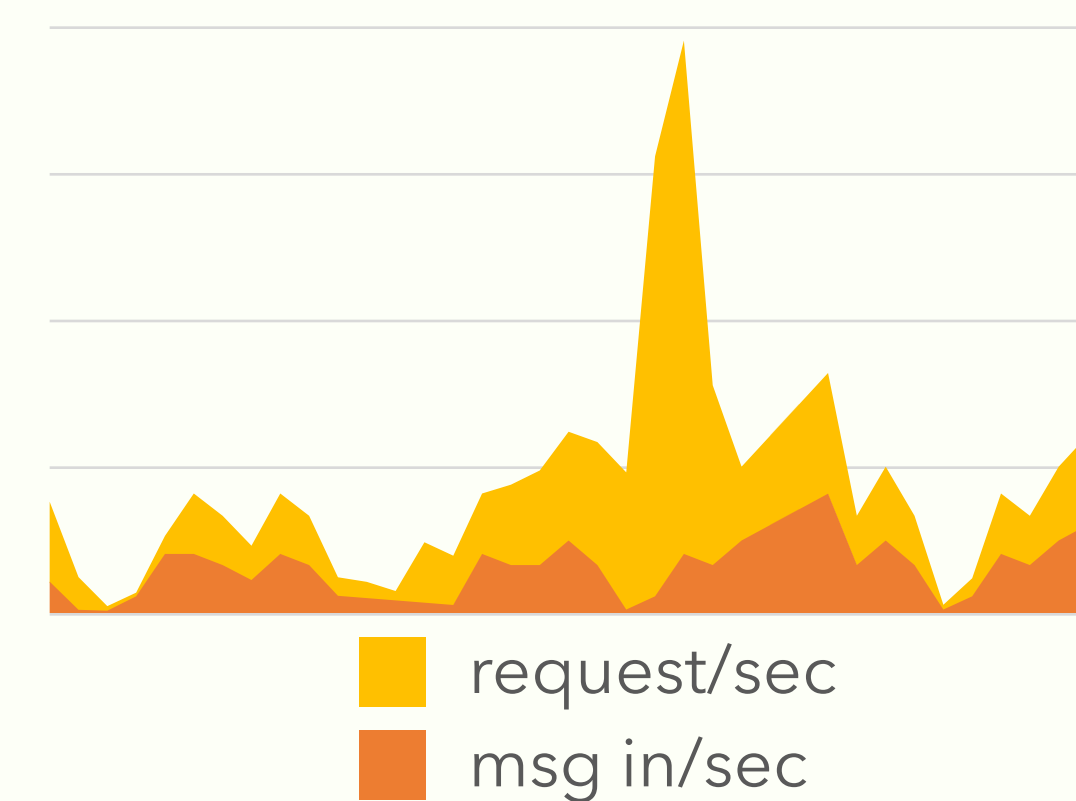
FUNC #2 Timeline



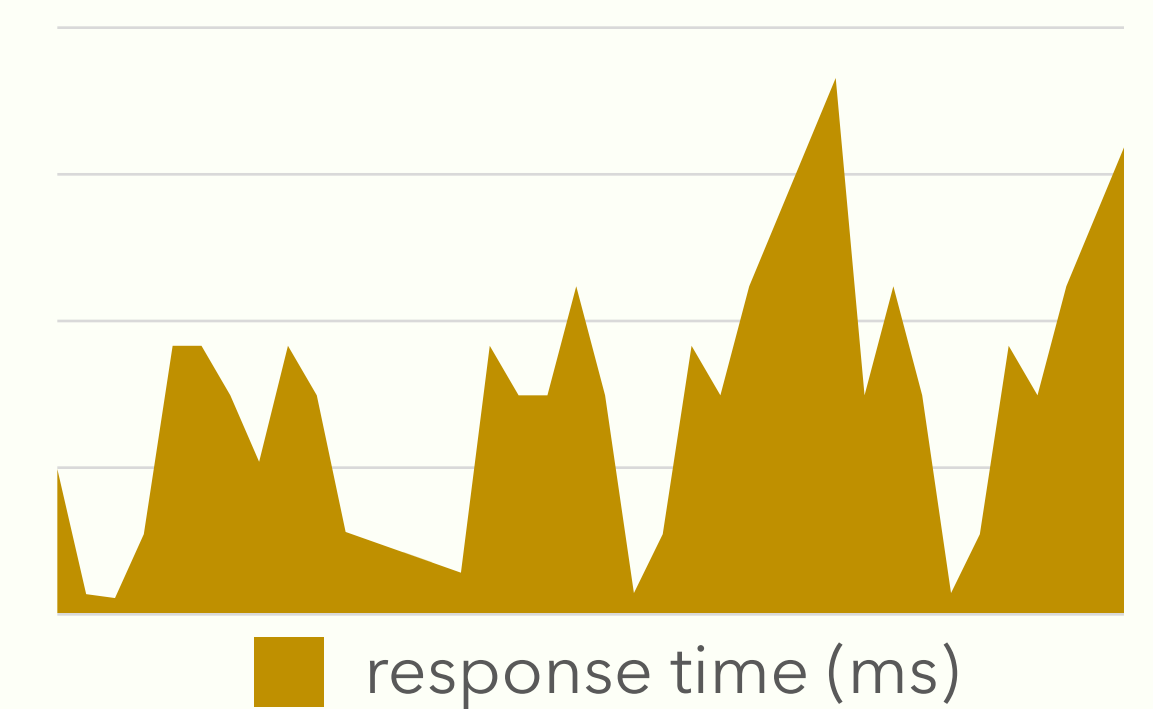
Heap memory usage



Message Condition



Response time



_ Report

40

Overview

Users

Problems

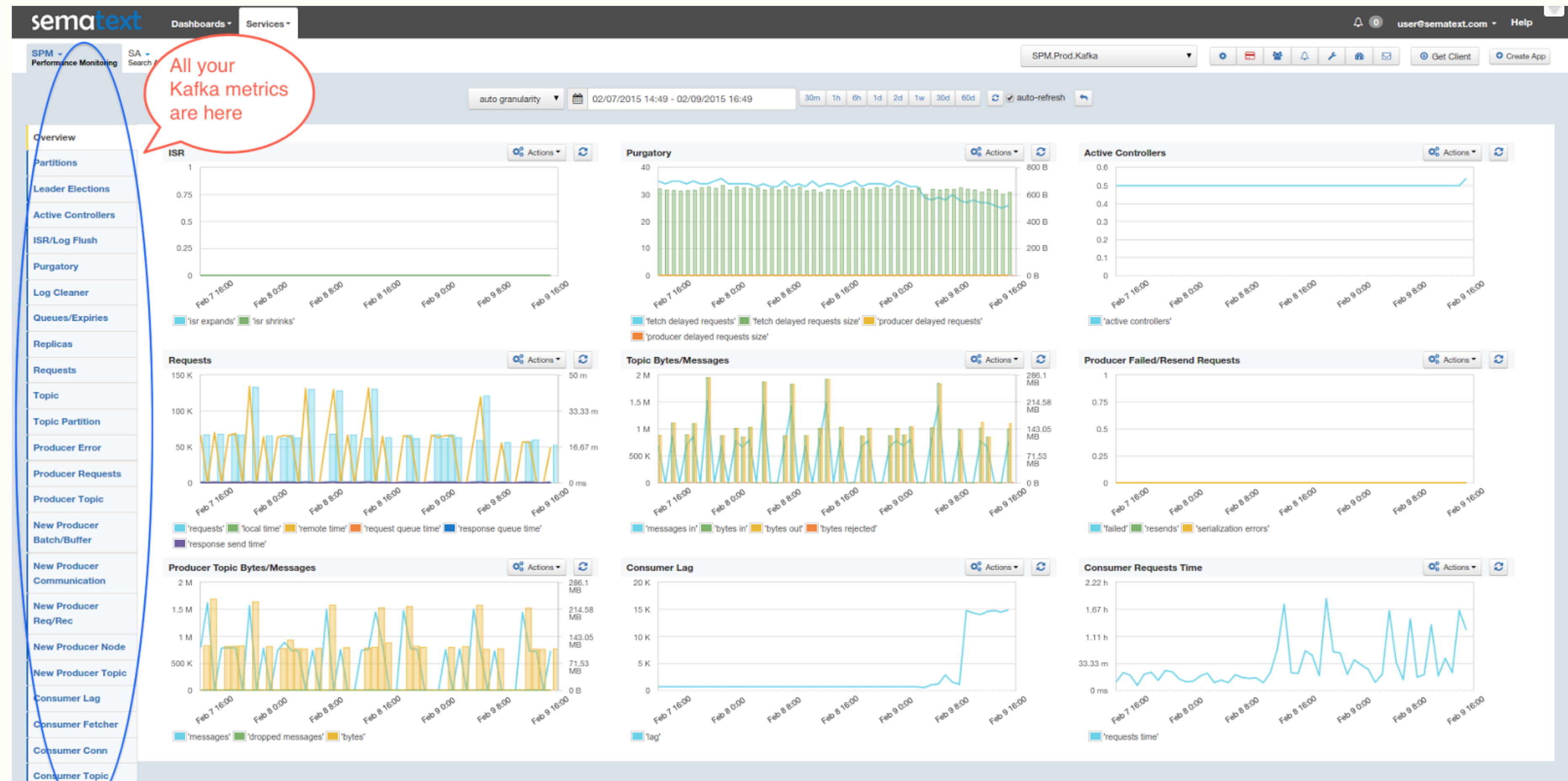
Solutions

Novelty

Scenario

Schedule

WHAT'S NEW?



Overview

Users

Problems

Solutions

Novelty

Scenario

Schedule

_ WHAT'S NEW?

Clear division of monitoring task

Further implication to BM

Overview

Users

Problems

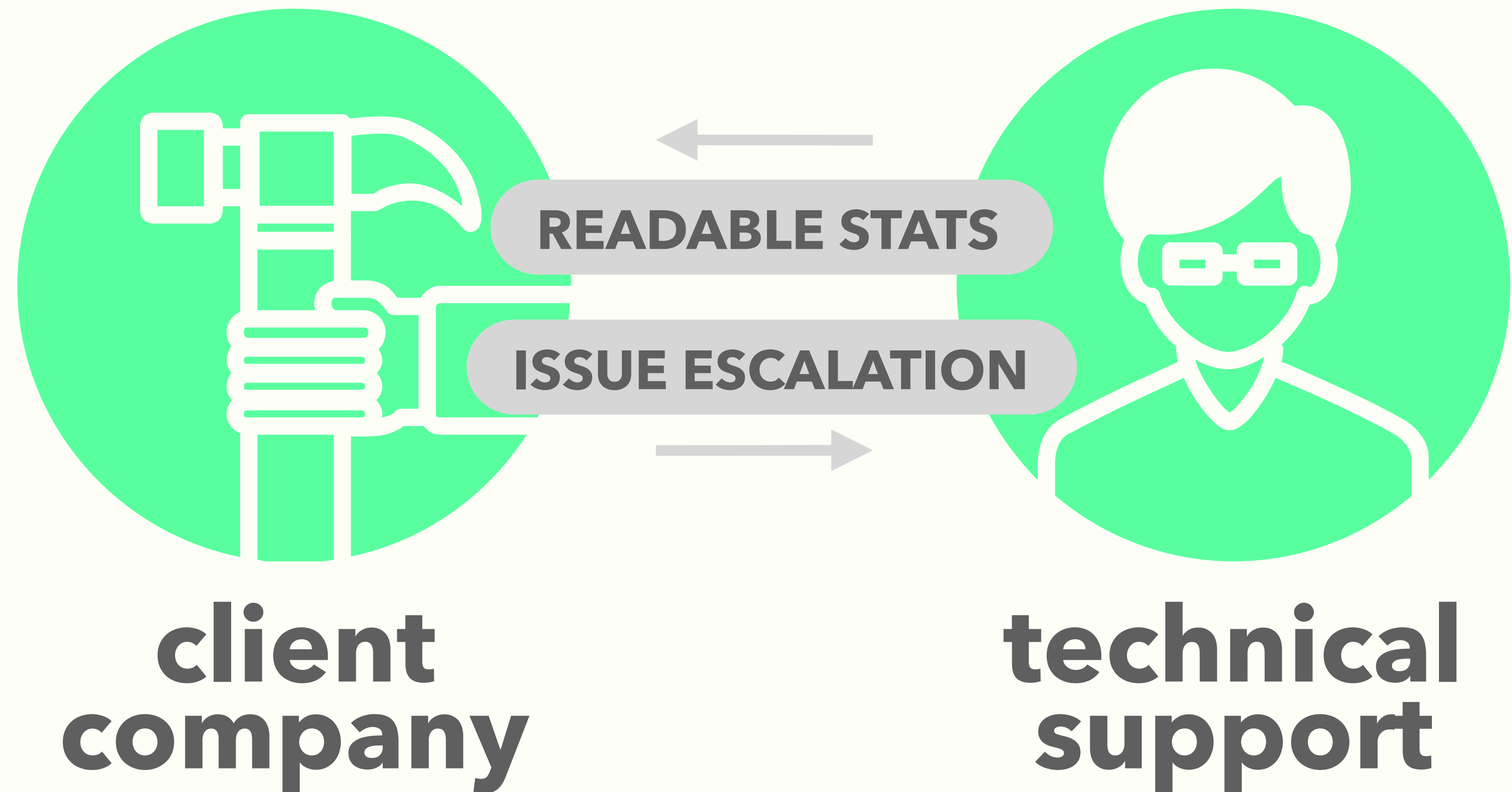
Solutions

Novelty

Scenario

Schedule

_ USER SCENARIO



PROGRESS_UPDATE

SPRINT #5

PROGRESS

_ SPRINT #4

US#1 : As a developer, I can easily plug-in MBean for visualization

- Build General MBean Client Factory (Youngjae Chang)
- Find appropriate D3 chart design for charts (Jaryong Lee)
- Study websocket structure (Seunghyo Kang)
- Design Database schema for saving metric history (Youngjae Chang)
- Define API interface for data communication & update (Jaryong Lee)
- Design websocket communication structure (Seunghyo Kang)

US#2 : As a user, I can monitor Kafka Ecosystem

- Plug-In Kafka MBeans into Interfaces (Youngjae Chang)
- Place charts to fit designated Kafka monitoring module (Jaryong Lee)

Progress

Architecture

Obstacles

_ SPRINT #5

US#1: As an operator, I can look through metric history

- Save metrics with multiple timespans
- Can extract specific metrics for designated timespan

US#2: As an user, I can use the tool with minimum configuration

- JSON interface with default values set by the industry
- Maintain functionality under the high load

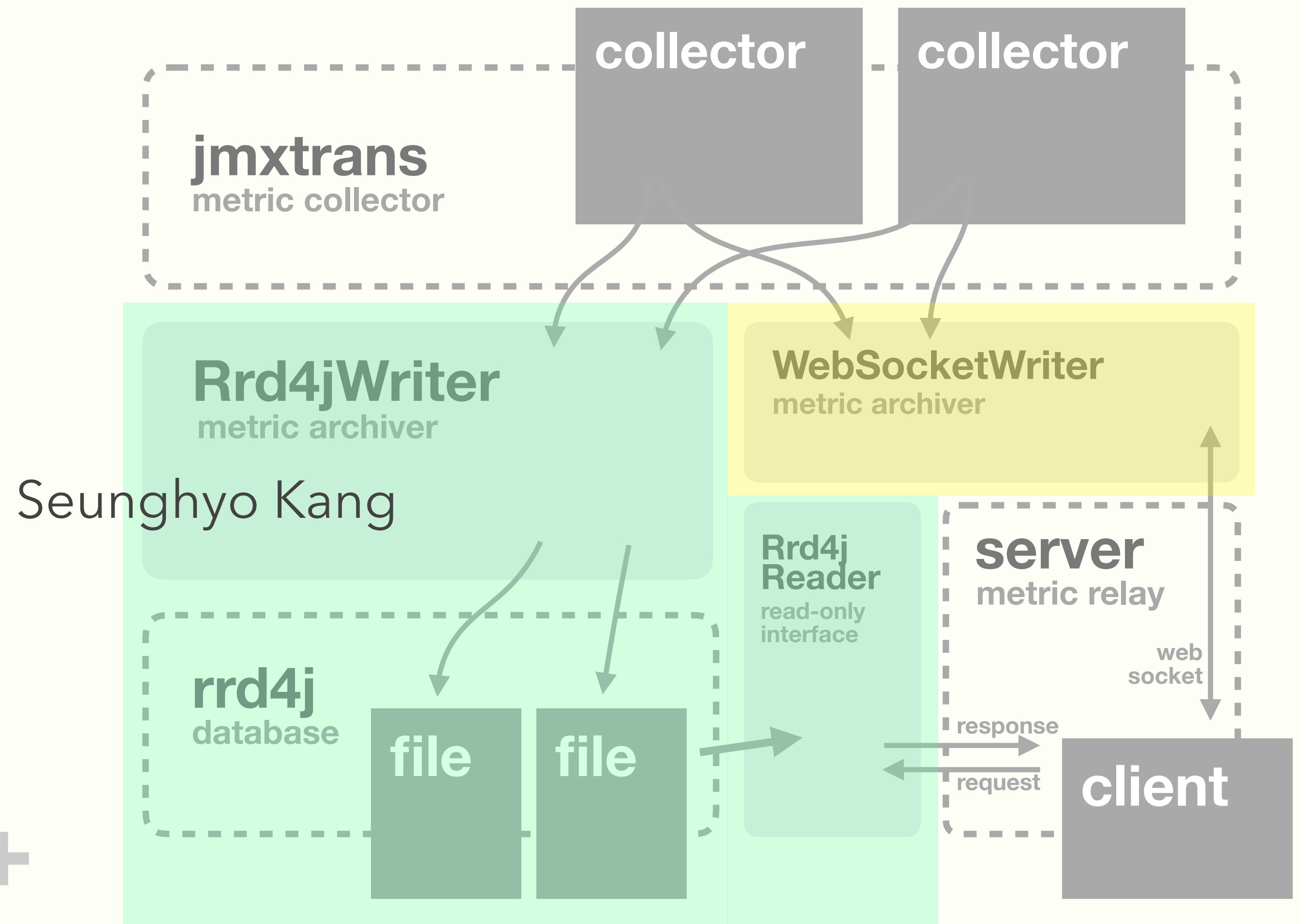
_ Report

46

Progress
Architecture

Obstacles

_ Current Focus



**Refactoring +
Documentation +**

_ WEB SOCKET

Pre-formatted zooms

To increase performance and decrease storage usage, RRD necessitates pre-formatted frequencies in data storage.

UI level interpretation of this restriction?



END