WHITEPAPER

Observability Pattern

CAELIUS CONSULTING

MuleSoft

www.caeliusconsulting.com

As businesses continue to embrace digital transformation, MuleSoft integration has become increasingly important. MuleSoft's Anypoint Platform offers a comprehensive suite of tools for API development, integration architecture, and enterprise integration, making it a top choice for organizations seeking robust data and cloud integration solutions. However, with the growing complexity of modern applications and the need for real-time monitoring and troubleshooting, observability has become a critical requirement for successful integration projects. In this whitepaper, we will explore the concept of observability in MuleSoft integration and how it can be achieved using the Anypoint Platform. We will delve into the key features of the platform, including API lifecycle management and API management platform, and show how they can be leveraged to improve the overall observability of integration solutions. Ultimately, this whitepaper aims to provide insights and best practices for organizations looking to optimize their MuleSoft integration projects through better observability.

The best way to correctly analyze the performance of a system is by measuring it. Observability is a comprehensive set of capabilities around logging, monitoring, analytics, troubleshooting, and measurement of systems.

MuleSoft, as an integration platform, plays a critical role in enabling observability. MuleSoft provides observability features, such as the ability to monitor APIs and services, view logs and metrics, and track errors and performance issues. By leveraging MuleSoft's capabilities and implementing observability patterns, teams can build and maintain resilient, high-performing systems that meet the needs of their users.

Below are some the observability patterns used :-

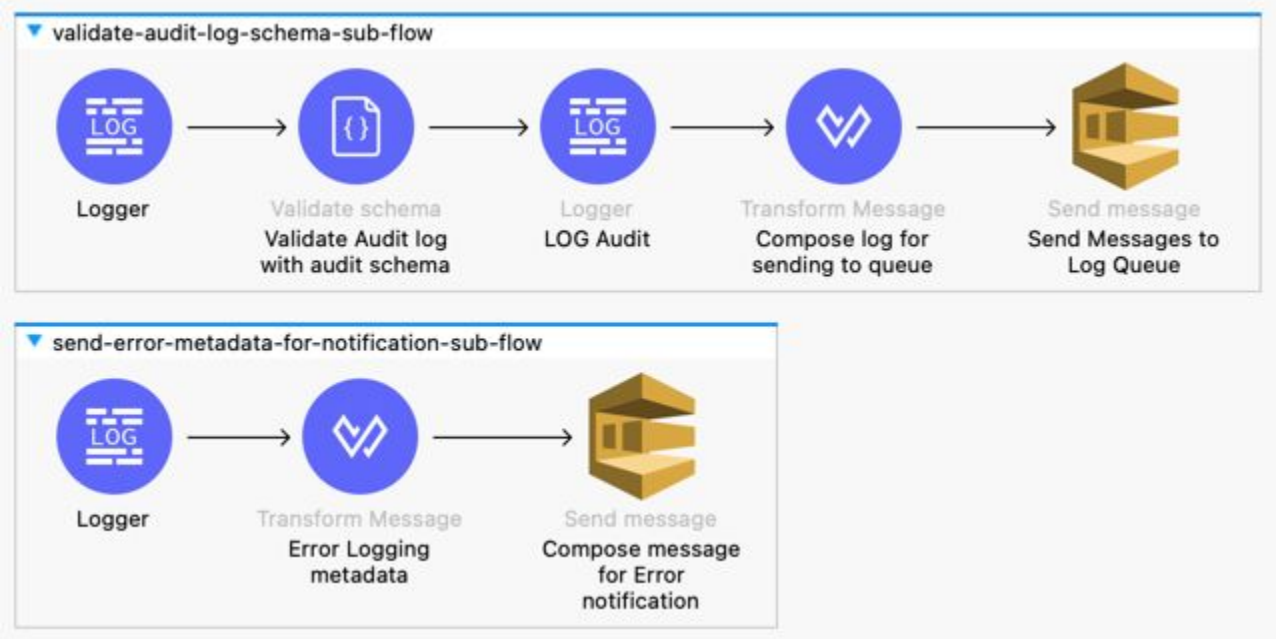
Audit Logging

This is used for documenting activity within your applications or systems whenever a new event is triggered. It records the occurrence of an event, the timestamp of the event, the responsible users and impacted systems. A series of audit logs is called audit trail. They are highly specific, give enhanced details of the events and their description. Audit logs can be used to track activities happening in a system and also help in finding any potential security breaches. In MuleSoft, audit logging can be leveraged in the application by using an audit structure which can be modified depending on the use case and can further be pushed to a database or some log analytic tools to get


```
{
 "transactionId" :
"b9e0a2dd-a2a9-48cc-8fbc-d50e70e8b8cf" ,
 "auditReferenceId" :
"194b9516-0d91-46b7-92d3-21b55beb3adc" ,
 "currentTimestamp" : "2023-02-26T07:16:01.325419Z" ,
 "sourceSystem" : "Salesforce",
 "targetSystem" : "MuleSoft",
 "description" : "",
 "status" : "Success",
 "otherAttributes" : [
   {
     "key": "orderNumber",
     "value": 2342
   },
   {
     "key": "eventId",
     "value": "23hjd-35asf"
   }
 ],
 "errorPayload" : "",
 "errorDescription" : ""
}
```

The above-mentioned Audit logging pattern can be managed in MuleSoft by creating a common logging and exception handling plugin. In our case, we have leveraged Amazon SQS to send logs and errors, which will further can send it to any log analytic platform such as elastic search, new-relic, splunk.

Below is a snapshot of how plugin looks like



An ingest pipeline from SQS to log analytic platform can be built whether in MuleSoft itself, any serverless app or any other platform. For error notification, the necessary attributes can be sent a queue, on further reading of the message, a decision can be taken whether to send a notification or retry the message.

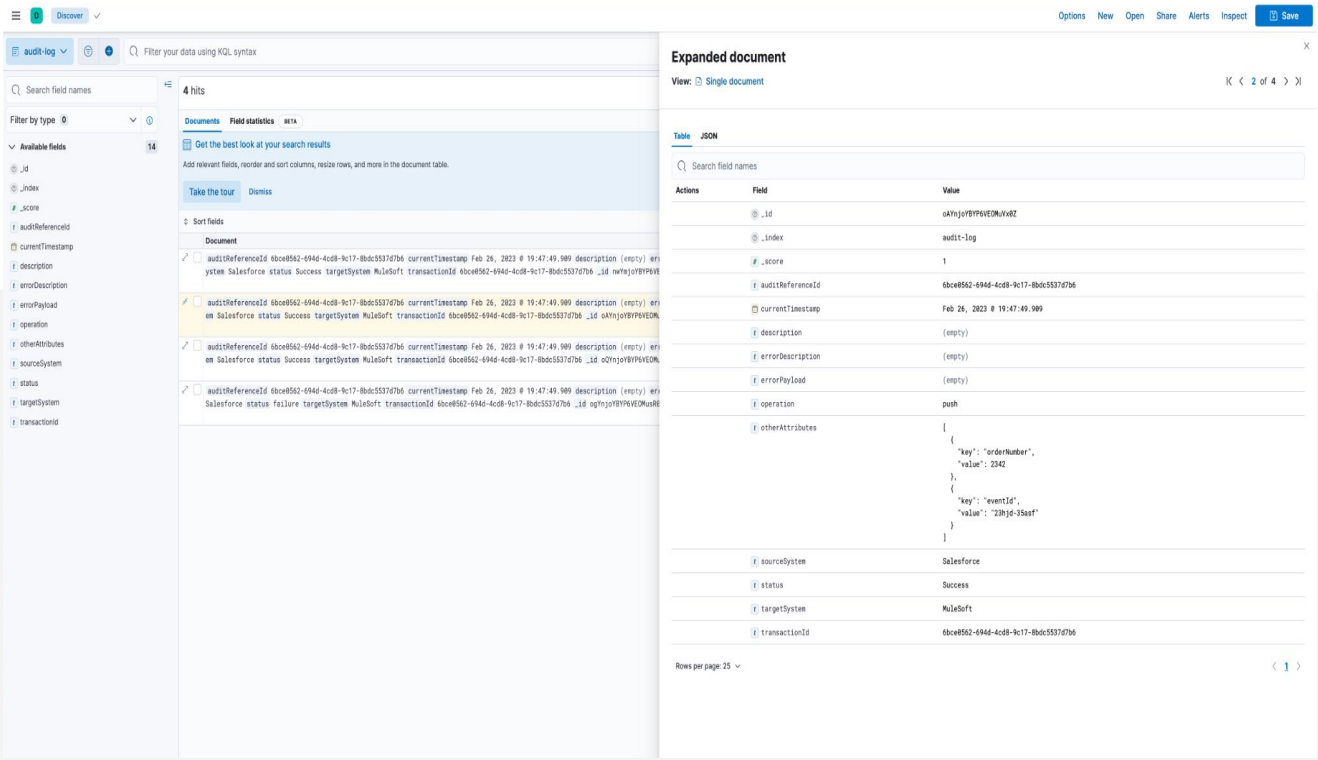
Some of the advantages of audit logging are :-

When pushed to any log analytic tools such as elk, splunk we can perform, aggregate and make dashboards with the data.

Troubleshooting system issues

Recommending new security and audit procedures

Ensuring compliance with industry regulations



Application Metrics

In order to measure the performance of your system, using application metrics is certainly the best way to do it. Application metrics comprise of certain core parameters to measure if a system is running in an optimal state.

CPU and Memory utilization

If CPU and memory usage in your system, your system will suffer to perform well. It is a very important metric to measure whether a system has enough memory capacity or enough CPU usage to work in an efficient way.

Error Rates

Rate of transactions failing over a given period of time. For good measurement system, accuracy errors should be within 4% and precision errors should be within 9%

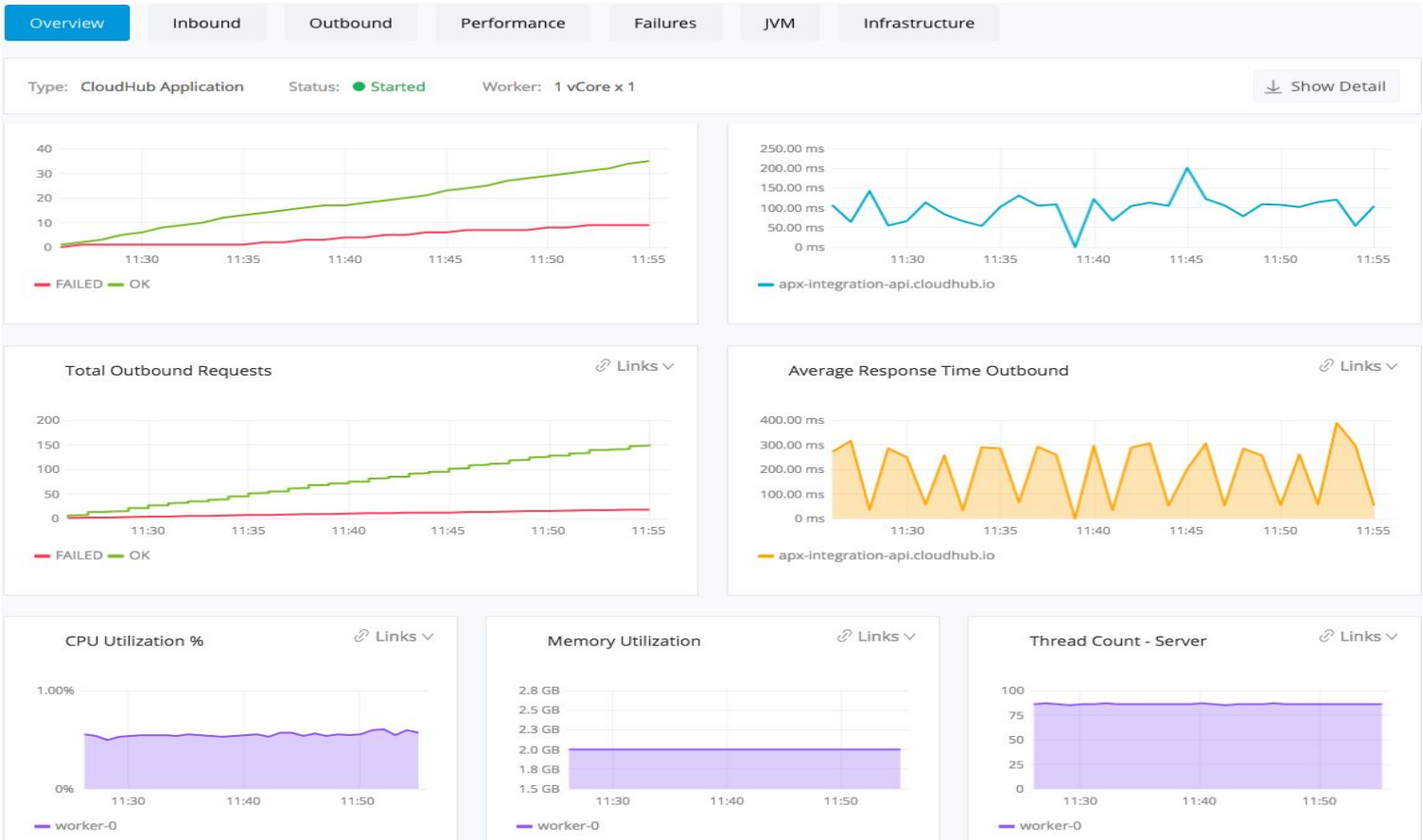
Average Response time

The average response time of a given transaction and whether it is in line with the SLA

JVM Metrics

Some of the JVM metrics include garbage collection count, Thread count, heap count etc.

Anypoint Platform helps in providing enhanced application metrics. In addition to that, tools like Prometheus, Grafana and New Relic can also be used to measure application metrics. Below dashboards are good examples of application metrics provided by anypoint platform.

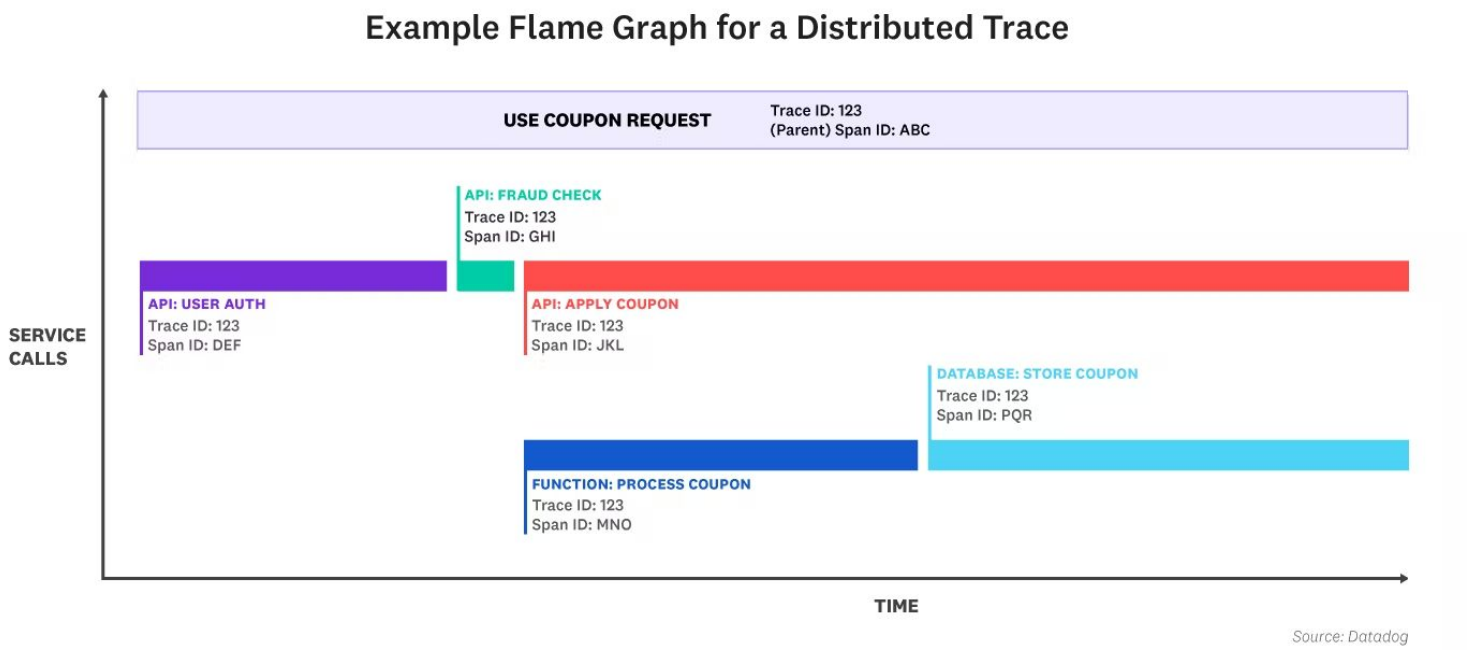


Average Response Time Grouped by Endpoint Outbound

Metric	Avg	Min	Max	Total	Count
/tmp	93.97 ms	0 ms	15.45 s	1.64 min	1050
/endpointA	1.10 s	0 ms	23.83 s	20.00 min	1093
/api/deck/new/	805.05 ms	0 ms	24.39 s	14.67 min	1093
/api/breeds/list/all	793.99 ms	0 ms	20.76 s	14.46 min	1093

Distributed Tracing

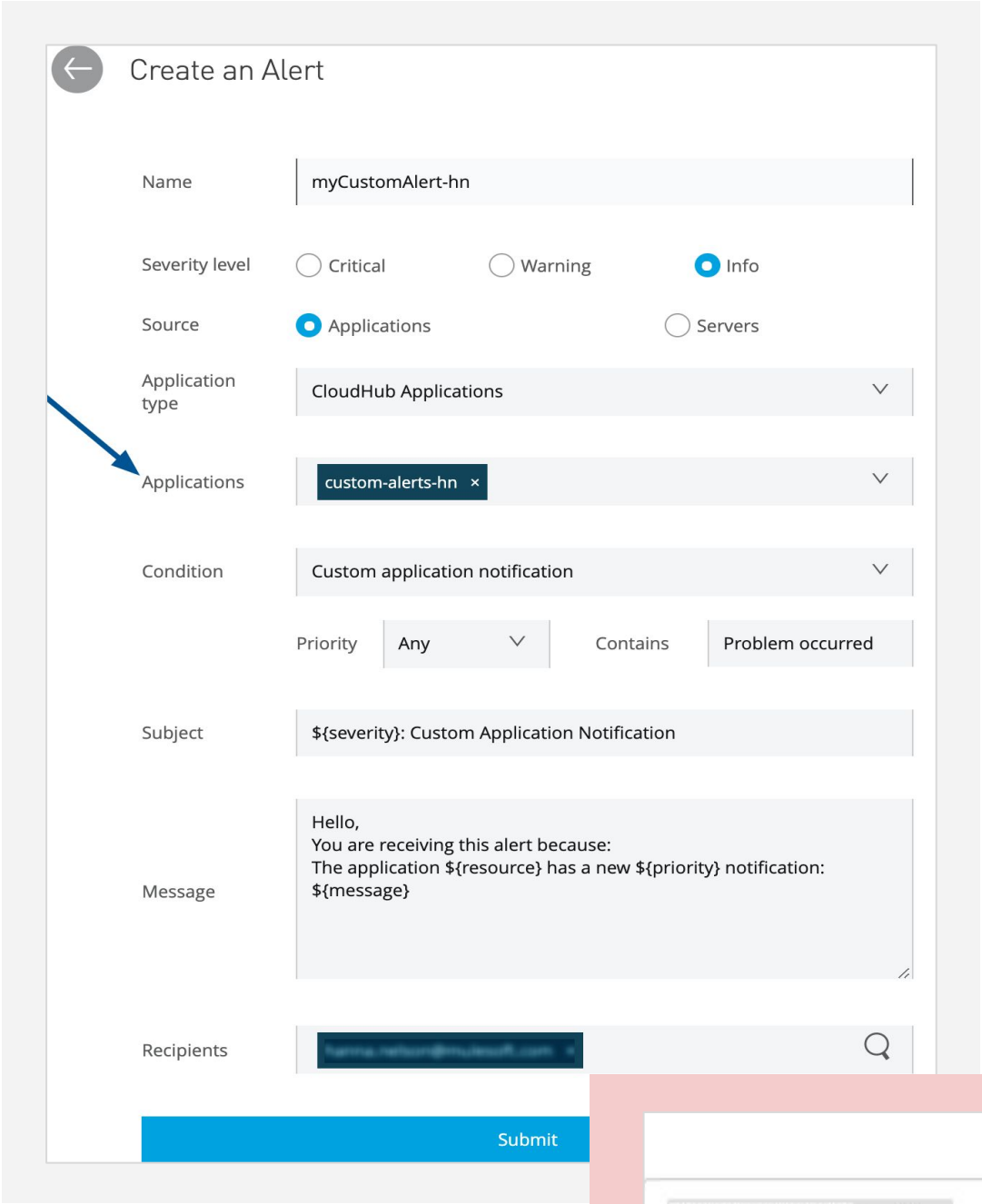
This method is used to track application requests as they flow from frontend devices to backend devices or between different api layers to databases or different target systems. End to End distributed tracing is triggered the moment a request is received from frontend. A parent time span is created to measure the time taken for full transaction to complete along with child time span to measure time taken at individual api or component level to measure if any latency is there at given point in the system. Below is an example showcasing distributed tracing.

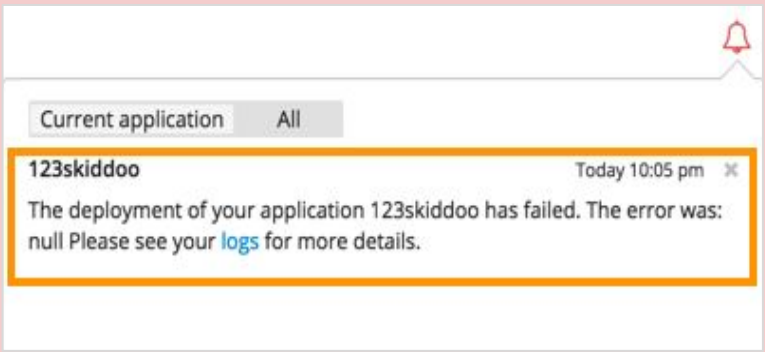


Exception Tracking

This method is used to track any runtime or deployment exceptions that occur and can send notification to appropriate shareholders. The notification can be sent via email or to any internal communication medium like slack, teams etc. The exception can consist of details such as application name,transaction id, the api layer(if any), target system involved, status code, error description and the error payload (if any).

Anypoint platform supports alerting mechanisms for raising alerts on the basis of status code, CPU or memory limits and many more. For custom exception tracking. Combination of queues and SNS topics can be used to raise email, slack or even phone notifications.





Conclusion

Observability is a crucial aspect of modern software systems, and observability patterns provide developers and operations teams with a way to build and maintain these systems. By implementing observability patterns such as logging, tracing, metrics, distributed tracing, and synthetic monitoring, teams can better understand the behavior of their systems and improve their reliability, availability, and performance. These patterns are especially important in complex, distributed computing environments, where issues can be challenging to diagnose and resolve.

CAELIUS CONSULTING

Reach us at:

+1 972 370 3073

info@caeliusconsulting.com

US

8501 Wade Blvd. Suite 250.
Frisco, TX 75034

Canada

6-2557 Dougall Avenue,
Suite 200, Windsor, ON
N8X 1T5

Singapore

Fifth Avenue, #03-08,
Guthrie House, Singapore -
268802

India

627 - A,, Bestech Business
Towers, Sector-66,
Chandigarh - 160066

www.caeliusconsulting.com