

Innovation & Internship: Evaluation and improvements of Open Source Keptn for cloud-native application life-cycle orchestration

- [Presentation to the RES-ESA team and Amadeus DevOps Community](#)
- [What is Keptn? :](#)
 - [Keptn itself :](#)
 - [Declarative multi-stage delivery :](#)
 - [The Shipyard file :](#)
 - [Stage, Sequence and Task :](#)
 - [Tools orchestration](#)
 - [Separation of concern](#)
 - [If it was for OBE ?](#)
 - [Features](#)
 - [Quality Gates Evaluations :](#)
 - [Auto-remediation](#)
- [Objectives of the internship](#)
 - [Technology Watch](#)
 - [Open-source contribution](#)
- [Open-source contribution : Spunk SLI-provider](#)
 - [Our work](#)
 - [The integration :](#)
 - [Technologies used :](#)
 - [Functionalities :](#)
 - [Difficulties :](#)
 - [The contribution Process](#)
- [Project Management](#)

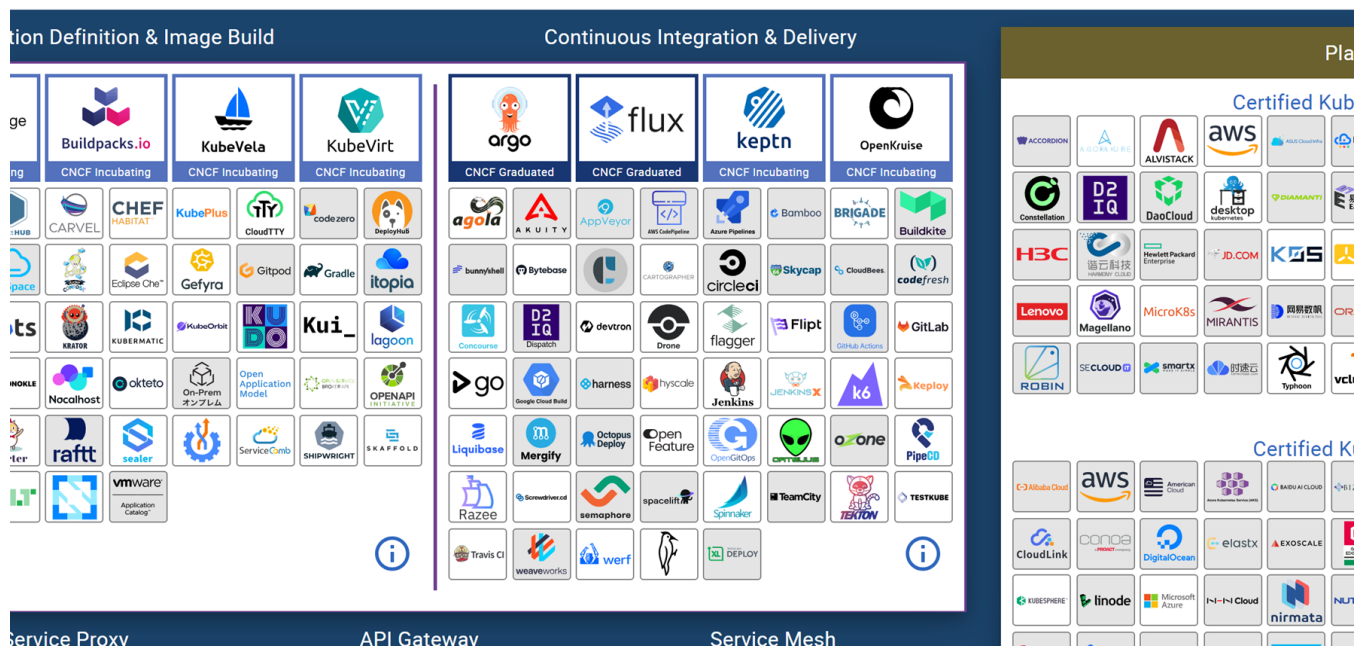
Presentation to the RES-ESA team and Amadeus DevOps Community

[Sharepoint - Keptn Presentation](#)

[Sharepoint - Record of the presentation](#)



What is Keptn? :



Keptn itself :



Keptn is a CNCF open-source framework designed to automate the deployment, operation, and monitoring of applications in a cloud-native environment. Keptn aims to simplify the process of delivering and maintaining software applications by providing tools and automation for various stages of the software development lifecycle. Keptn can be seen as a way to implement the principles of Continuous Delivery, DevOps, and Site Reliability Engineering (SRE) by providing automation, monitoring, and self-healing capabilities throughout the application lifecycle. It's particularly well-suited for organizations adopting microservices architectures and deploying applications in containerized environments like Kubernetes.

Key features of Keptn include:

1. **Continuous Delivery Automation:** Keptn automates the deployment process, allowing developers to focus on writing code rather than managing complex deployment pipelines.
2. **Event-Driven Architecture:** Keptn operates based on events triggered by different actions in the application lifecycle. This event-driven approach enables dynamic responses and automation.
3. **Multi-Stage Deployment:** It supports multi-stage deployment, including development, testing, staging, and production. This allows for consistent and reliable application promotion through different environments.
4. **Configuration as Code:** Keptn uses configuration files to define deployment workflows, making it easier to version and manage changes to the deployment process.
5. **Monitoring and Remediation:** Keptn provides built-in monitoring capabilities and the ability to set up automatic remediation actions in case of performance or availability issues.
6. **Integration with Observability Tools:** It integrates with various observability tools, such as Prometheus, Dynatrace, and others, to provide insights into application performance.
7. **Cloud-Native and Kubernetes-Oriented:** Keptn is designed to work well in cloud-native environments, particularly within Kubernetes clusters.
8. **Support for Multiple Programming Languages:** Keptn is not tied to a specific programming language or technology stack, making it versatile and adaptable to various applications.

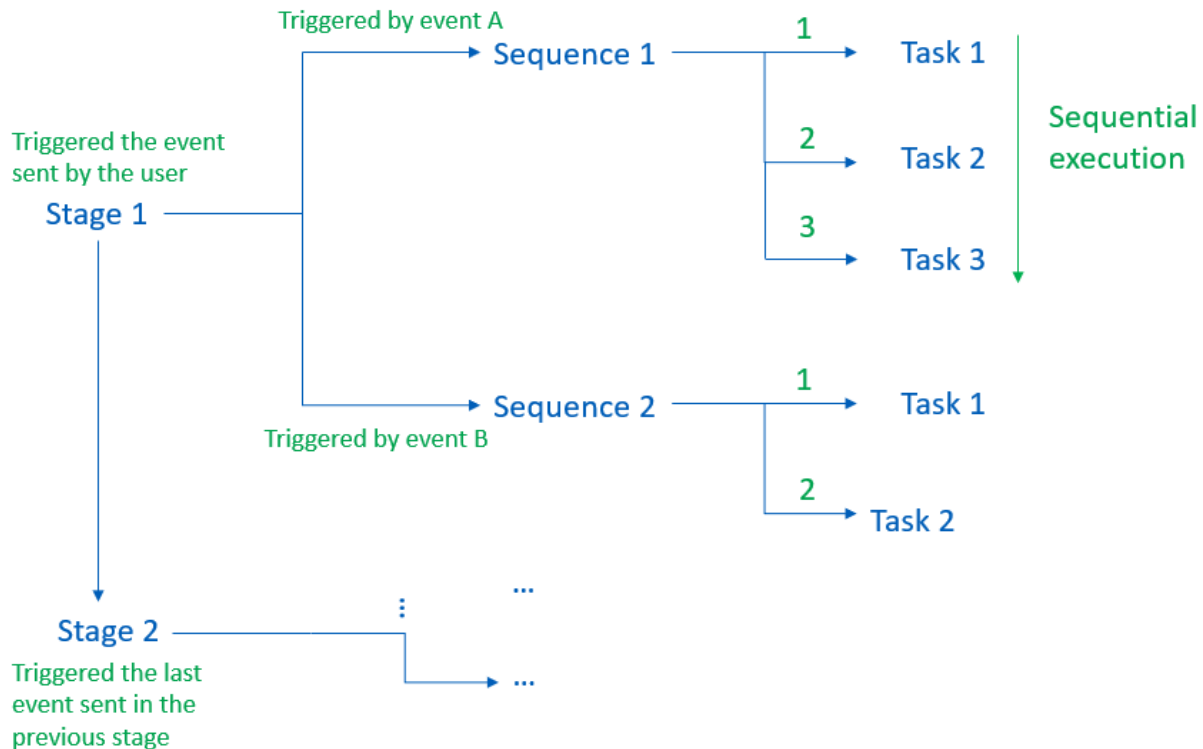
Declarative multi-stage delivery :

The Shipyard file :

```
1  apiVersion: "spec.keptn.sh/0.2.2"
2  kind: "Shipyard"
3  metadata:
4    name: "shipyard-bird-app"
5  spec:
6    stages:
7      - name: "qcp"
8      sequences:
9        - name: "delivery"
10      tasks:
11        - name: "deployment"
12        - name: "test"
```

The Shipyard file is a configuration file used in the Keptn framework. Keptn uses the Shipyard file to define the high-level workflow and stages of an application's deployment and delivery process. It's a critical component that outlines the sequence of actions and events that should occur during the software development lifecycle.

Stage, Sequence and Task :



1. **Stages:** Stages represent different phases of the software delivery process, such as "dev," "staging," and "production." Stages are usually triggered by the last event of a previous stage. The first stage is triggered when the user decide to start his first delivery sequence.
2. **Sequences:** Sequences define the order of actions, events, and deployments that should take place within each stage. For example, a sequence might involve deploying a new version of the application, running tests, and promoting the application to the next stage. A particular sequence is triggered by a particular event specified in the shipyard file.
3. **Tasks:** Tasks are specific actions or operations that should be executed during a sequence. This could include deployments, running tests, updating configurations, etc. The tasks of sequence are executed sequentially in the order they are specified in the shipyard file.

Tools orchestration

All the tasks defined in the shipyard can be done by, almost, any tool. One aim of Keptn is **allow users to bring their everyday tools** with them. Through keptn services you can connect your tool with Keptn. Those keptn-services will be the **interface** between Keptn and the tool you are using.

Separation of concern

One big advantage of Keptn, and also one of its most important characteristic/feature, is that Keptn **separates the process defined by SREs from the actual tooling defined by DevOps engineers and the information about the artifacts**. In other words, in one hand you have your shipyard file - which will just describe your workflow, with all the tasks to perform - and in the other hand, through keptn services you will decide what to do actually in tasks defined in the shipyard - and which tool should perform that task.

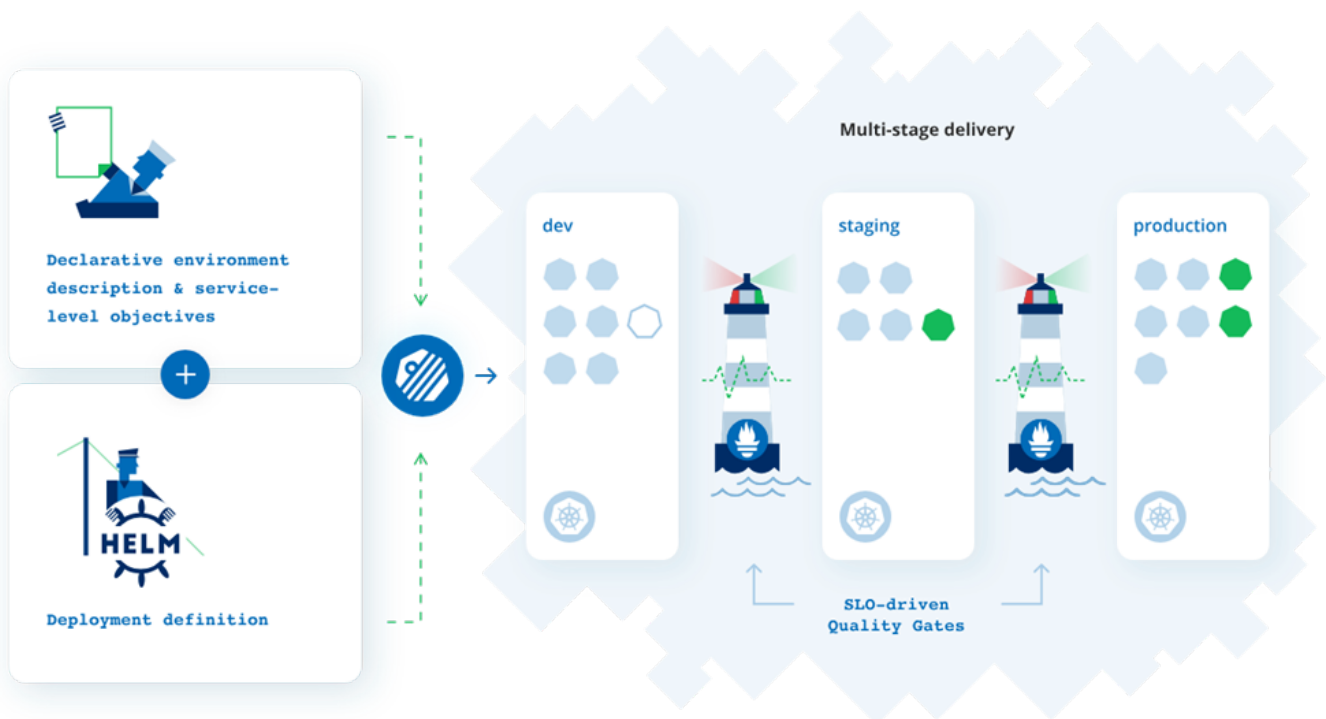
Thus, this separation of concern :

- make Keptn less coupled with the tools your are using
- create a more reusability of your delivery pipeline : **you can share the same shipyard across different projects**

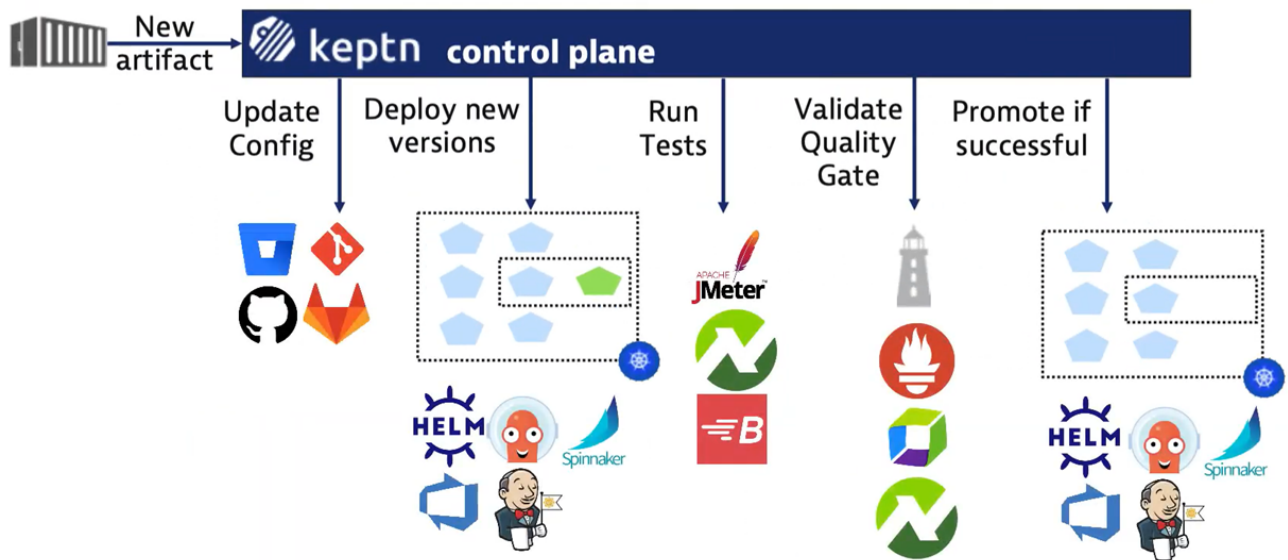
If it was for OBE ?

Features

Quality Gates Evaluations :

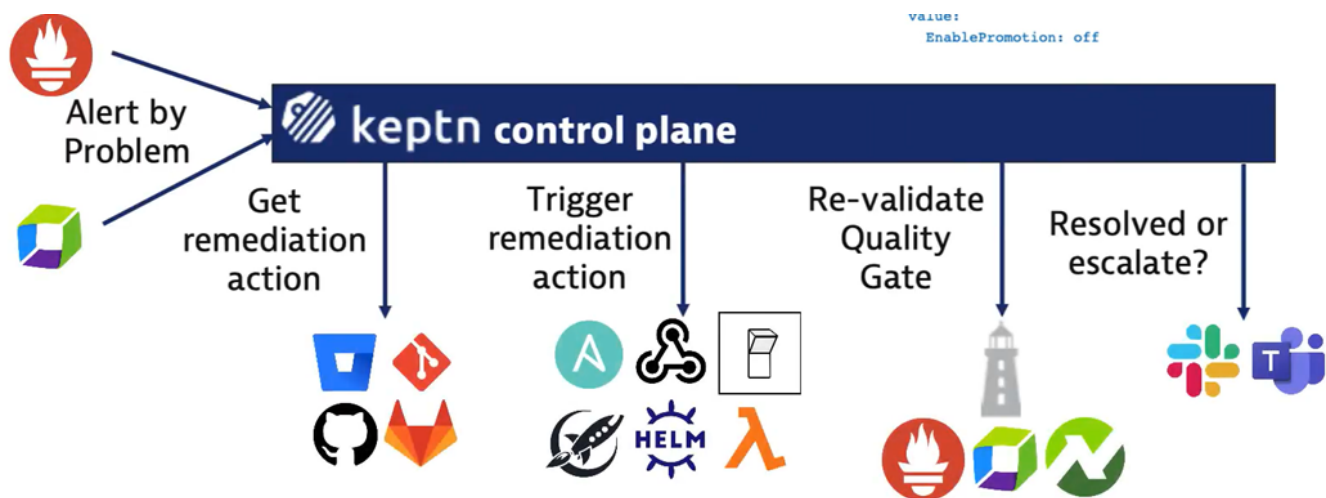


Quality Gates, in the context of software development and continuous delivery, refer to checkpoints or criteria that an application needs to meet before progressing from one stage of the development or deployment pipeline to the next. Quality Gates are designed to ensure that the application deployed meets certain quality standards, thereby reducing the risk of issues reaching production environments.



In order to perform these evaluations, keptn needs two files that are the `sli.yaml` and the `slo.yaml` files. SLIs (Service Level Indicators) define the indicators to measure in order to be able to evaluate our application. SLOs (Service Level Objectives) define the objectives that those indicators have to reach. When an evaluation is triggered by the user, the values of the indicators are calculated and compared to the objectives in order to calculate a score. If the score is high enough, our application can progress to the next stage.

Auto-remediation



```

1  apiVersion: spec.keptn.sh/0.1.4
2  kind: Remediation
3  metadata:
4    name: selobe-wind-remediation
5
6  spec:
7    remediations:
8      - problemType: response_time
9        actionsOnOpen:
10         - name: Rollback to previous version using helm
11           action: rollback
12           value:
13             stableVersion: "selobe-wind:1.0.0"

```

Often new versions of stable applications are deployed. These new versions can pass the tests and evaluations all the way to the production environment but some problems can still occur there. Keptn allows you to monitor your application with your monitoring tool. The slo.yaml and sli.yaml files are used by the monitoring tools in order to set alerts concerning the application. Whenever a is detected by these tools, a remediation process can be automatically started by keptn to perform actions that could solve the issue. This process is called auto-remediation and is possible thanks to the remediation.yaml file that tells keptn what to do when receiving a certain type of problem. An evaluation is performed after the remediation actions have been performed in order to be sure that the problem is solved.

Objectives of the internship

Technology Watch

The first Objective was to explore and stay in touch with the innovations in term of Continuous Delivery by evaluating Keptn.

Open-source contribution

The second one was to contribute to the open-source code of Keptn in order to make Amadeus a little more present in the ecosystem of Open-Source.

Open-source contribution : Splunk SLI-provider

[Splunk SLI-provider repository](#)

Our work

keptn Quality Gates

Lighthouse service

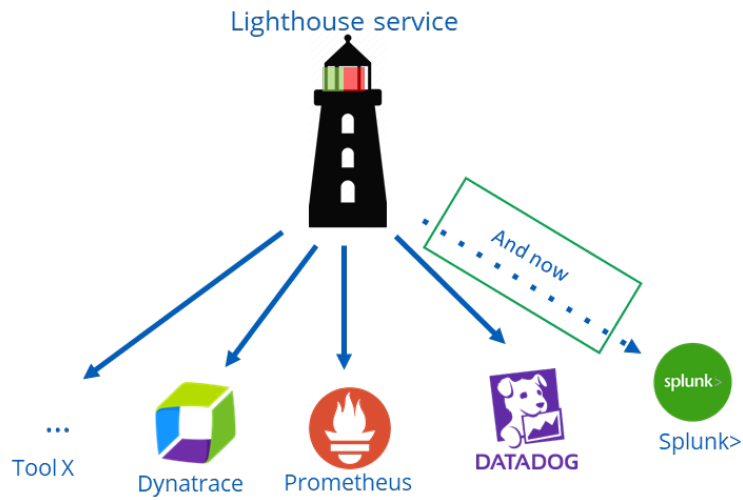


Splunk Service 



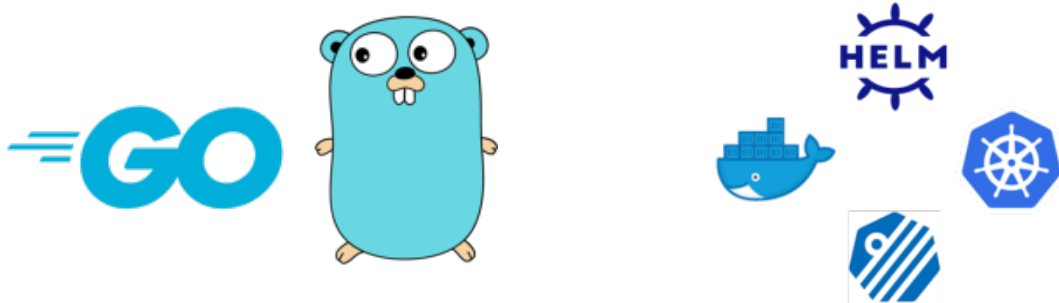
The integration :

keptn allows you to bring the tool you want if there is a Keptn-service allowing you to link the tool to Keptn. There were no link between Keptn and Splunk. A keptn-service allowing keptn to use Splunk as the source of the Service Level Indicators and as a monitoring tool needed to be created. **So we created the [Splunk SLI-provider](#).**



Quality Gates

Technologies used :



Functionalities :

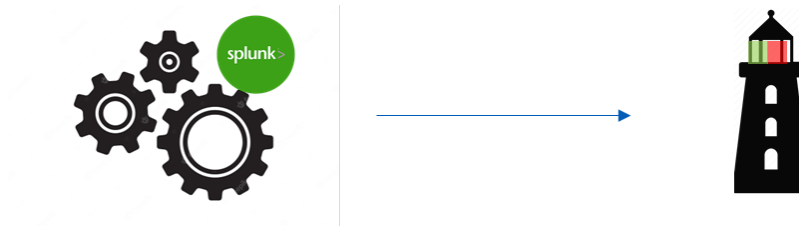
1. The Splunk SLI-provider subscribes to two events :
 - a. `configure.monitoring.triggered` event that is the event sent by Keptn when the user of Keptn specifies the monitoring tool to use for the microservice to be deployed.
 - b. `getsls.triggered` event that is the event sent by Keptn when an evaluation is performed and that values of the indicators need to be calculated.



`Configure.monitoring.triggered` event

`getSLI.triggered` event

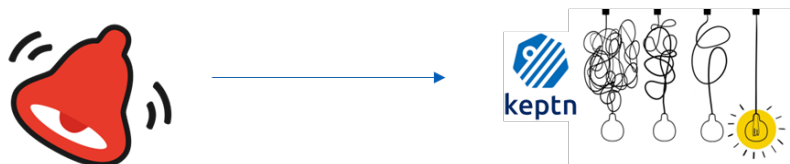
1. When a `configure.monitoring.triggered` event is received, the Splunk SLI-provider fetches the `sli` and `slo` files and sets Splunk alerts in the Splunk instance by using the the Splunk searches of the indicators to get the metrics and by using the negation of the pass condition in the objectives as the triggering condition.



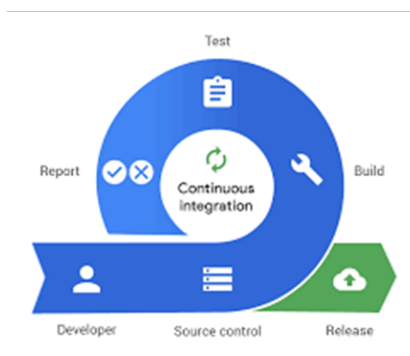
1. When a `getsli.triggered` event is received, the Splunk SLI-provider fetches the `sli` file, executes the splunk searches corresponding to the indicators and sends the results to `keptn` for the evaluation.



1. Whenever the `keptn` alerts created in the Splunk instance are triggered, the Splunk SLI-provider sends a `remediation.triggered` event containing the information about the triggered alert to `keptn`. The auto-remediation process is then started.

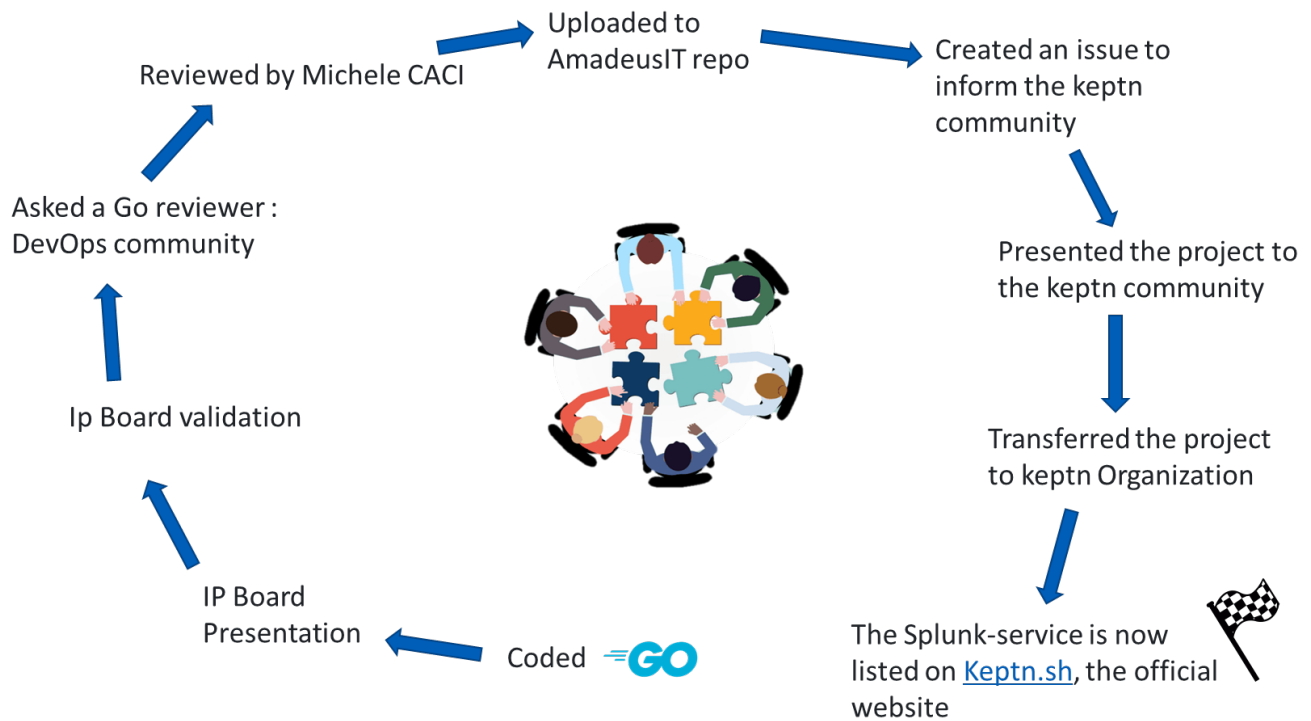


Difficulties :



1. Poor documentation about the code of the Keptn services
2. Learning GO from scratch
3. Setting up the Continuous Integration
4. Dealing with the Splunk SLI-provider security issues

The contribution Process



The code of the Splunk SLI-provider repository is located here in the [Splunk SLI-provider repository](#).

In the page [Integrations](#) of [keptn](#) you can find the among the SLI providers.

Here is the Artifacthub page of the Splunk SLI-provider : <https://artifacthub.io/packages/keptn/keptn-integrations/splunk-sli-provider>

Project Management

We worked with GitHub and its project view to create our backlog and manage the project in an agile way.

Keptn internship

Backlog

By priority

By size

New View

Filter by keyword or by field

Discard

See

New

Backlog

Ready

In progress

In review

Done 27

add item

+ Add item

+ Add item

+ Add item

+ Add item

+ Add item

Draft

service to remove useless configuration or add better ones

Medium

Medium

Draft

As a Keptn contributor I want to write an end-to-end tests for the service to be able to run an entire and real test

High

Large

Draft

As a keptn contributor I want to update/create github action workflows to cover unit test/integration tests

Medium

Medium

Draft

As an intern I want Amadeus open source community allow us to contribute to keptn open source with the Amadeus label

Medium

Medium

Draft

As a keptn contributor, I need to write some unit tests for the remediation part so that the CI is improved

High

Medium

Draft

As an intern I want to understand how Bid works within Amadeus so that I can see how Keptn can be used

Medium

Large

Draft

As an intern I want to understand what is the Forge so that I can see how Keptn and the splunk SLI-provider can be integrated within it

Medium

Large

Draft

As a keptn contributor I want to learn how to contribute to keptn open source code

Urgent

Large

Draft

As an intern I need a keptn presentation that's available at all times.

Urgent

Large

Draft

As a Keptn contributor I want to write a roadmap for the splunk-service to give a clear documentation of our service

High

Large

Draft

As a keptn contributor, I need to address Michele Caci's issues on github so that my golang code will be improved

High

Medium

Draft

As an intern I need to write my internship report so that I can submit it to my school

High

Large