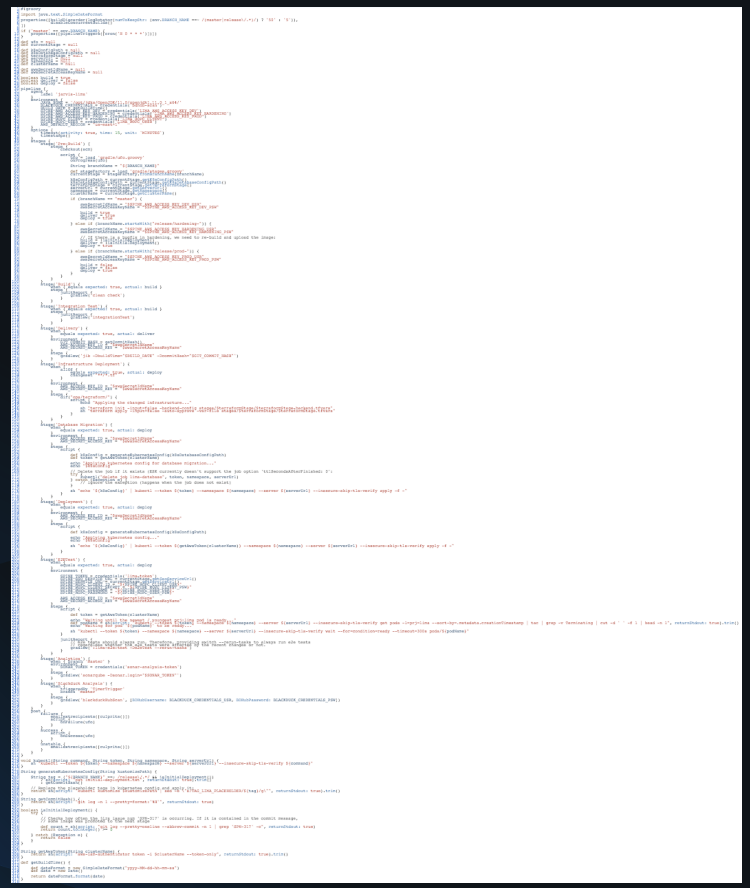
Introducing Keptn

Andreas Grimmer
Technology Strategist
Dynatrace

keptn

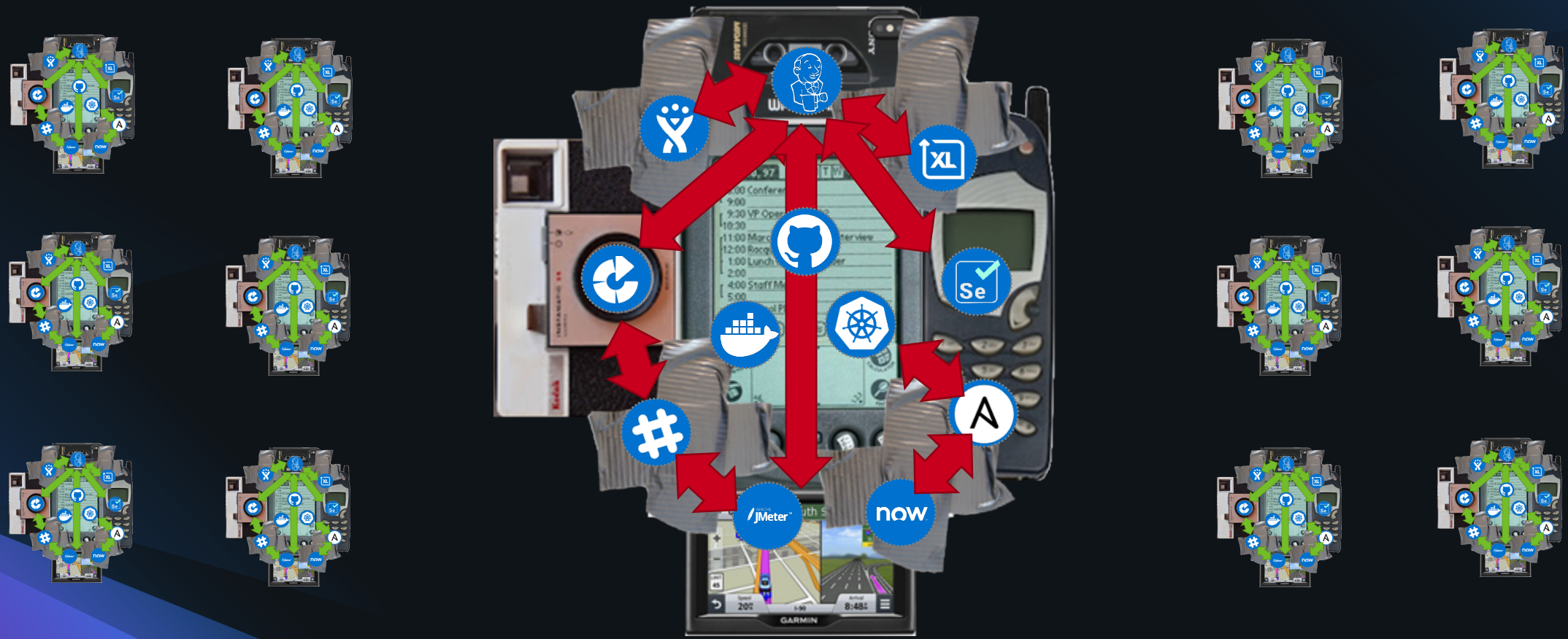
Which problems does Keptn solve?



Pipeline for CD

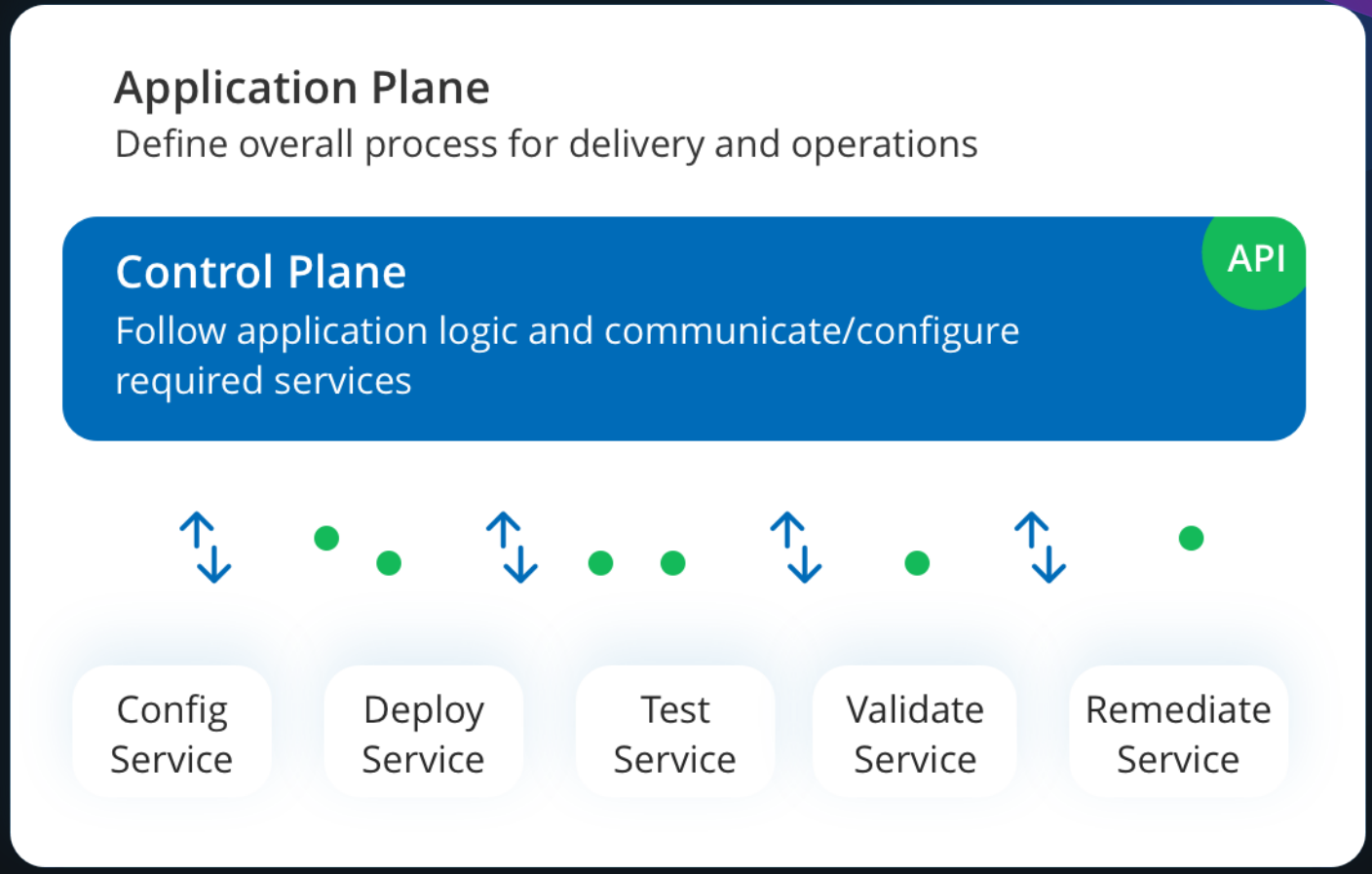
- >350 LOC
- Mixed information about
  - Process (deploy, test, evaluate, …)
  - Target platform (k8s, …)
  - Environments (dev, hardening, …)
  - Tools (Terraform, Helm, hey, …)

Quote: “Pipelines seem to be becoming our new future un-managable legacy code!“



What is Keptn?

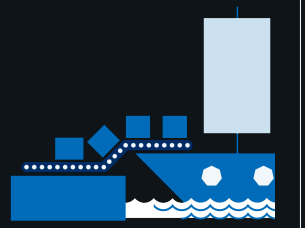
Keptn is an **event-based control plane** for **continuous delivery** and **automated operations** for cloud-native applications.



Core concepts

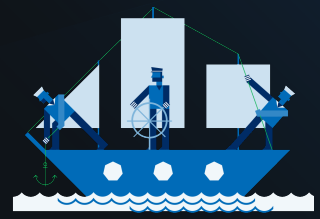
Declarative Approach for Delivery and Operation Automation

- Share definitions across any number of microservices w/o individual pipelines and scripts



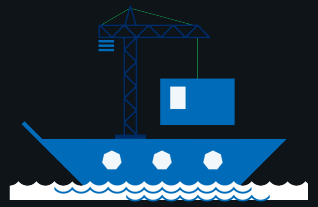
Separation of concerns

- Processes defined by SREs
- Tooling defined by DevOps
- Artifacts defined by Devs



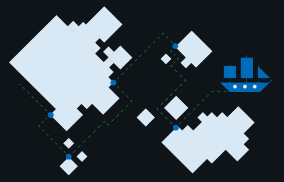
Event-based automation for extensibility

- CloudEvents for all delivery and operations steps
- Simple and fast integration by registering to those events



Built-in Observability

- Built-in tracing capabilities for all deployments and operations flows
- Visualization in the Bridge



Declarative delivery and operations: Shipyard

- Shipyard specifies stages and what to do

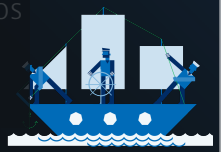
```
---
version: 0.2.0
kind: Shipyard
metadata:
  name: shipyard-example
spec:
  stages:
    - name: "hardening"
      workflows:
        - name: artifact-delivery
          tasks:
            - update:
            - deployment:
                strategy: blue_green
            - test:
                kind: performance
            - evaluation:
            - release:
```

Declarative Approach for Delivery and Operation Automation

- Share definitions across any number of microservices w/o individual pipelines and scripts

Separation of concerns

- Processes defined by SREs
- Tooling defined by DevOps
- Artifacts defined by Devs



Event-based automation for extensibility

- CloudEvents for all delivery and operations steps
- Simple and fast integration by registering to those events

Built-in Observability

- Built-in tracing capabilities for all deployments and operations flows
- Visualization in the Bridge

Declarative tooling: Uniform

- Uniform declares tools, which register to tasks specified in the Shipyard

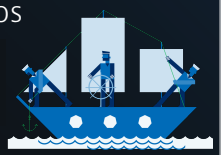
```
---
version: 0.1.0
kind: Uniform
metadata:
  name: uniform-example
spec:
  services:
    - name: deployment-service
      image: keptn/argo-service:1.0.0
      events:
        - sh.keptn.deployment.triggered

    - name: test-service
      image: keptn/jmeter-service:1.6.0
      events:
        - sh.keptn.test.triggered
```

Declarative Approach for Delivery and Operation Automation

- Share definitions across any number of microservices w/o individual pipelines and scripts

Separation of concerns

- Processes defined by SREs
- Tooling defined by DevOps
- Artifacts defined by Devs



Event-based automation for extensibility

- CloudEvents for all delivery and operations steps
- Simple and fast integration by registering to those events

Built-in Observability

- Built-in tracing capabilities for all deployments and operations flows
- Visualization in the Bridge

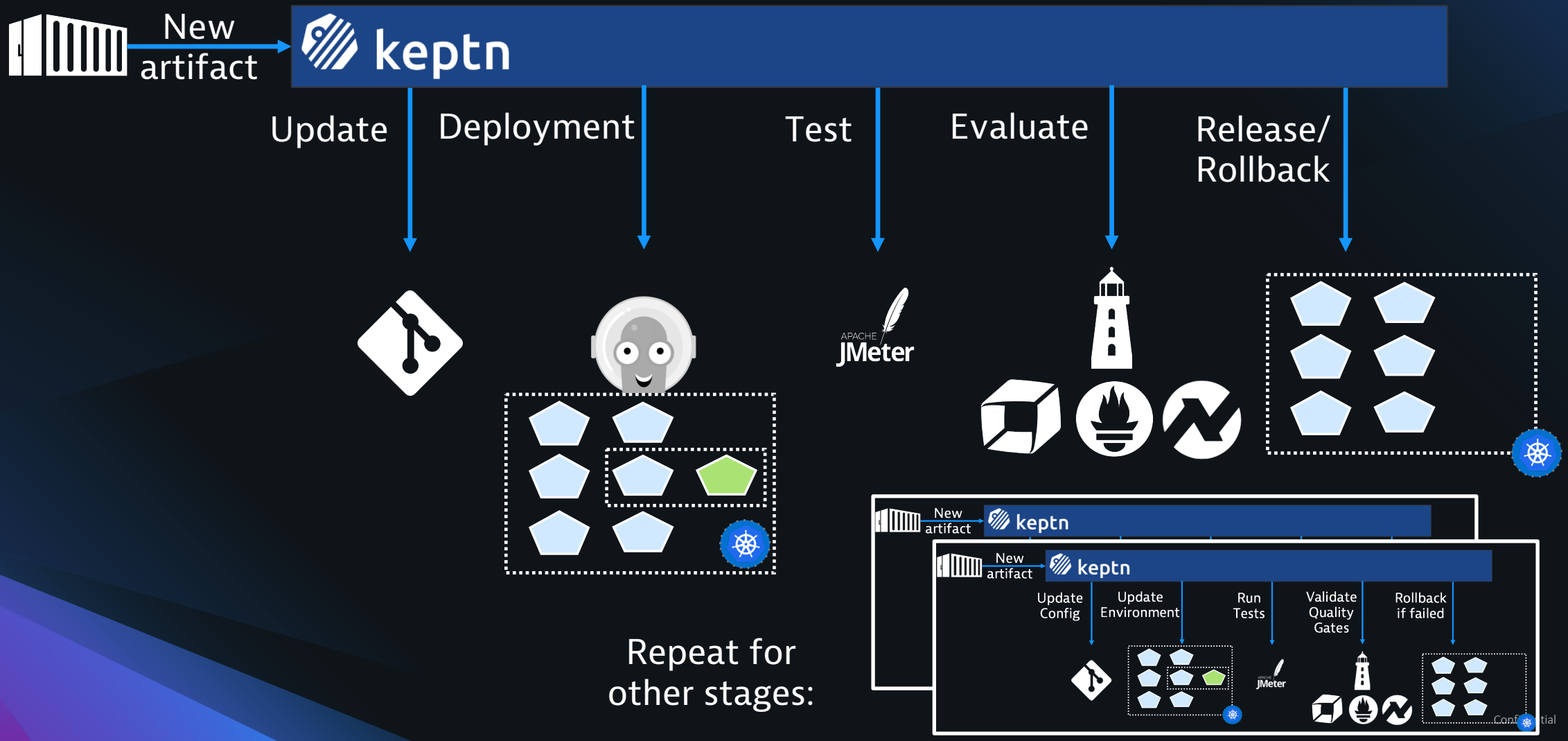
Interopability using Events

- Based on Cloudevents: https://github.com/cloudevents/spec
- Each task is represented by a tripple of events:

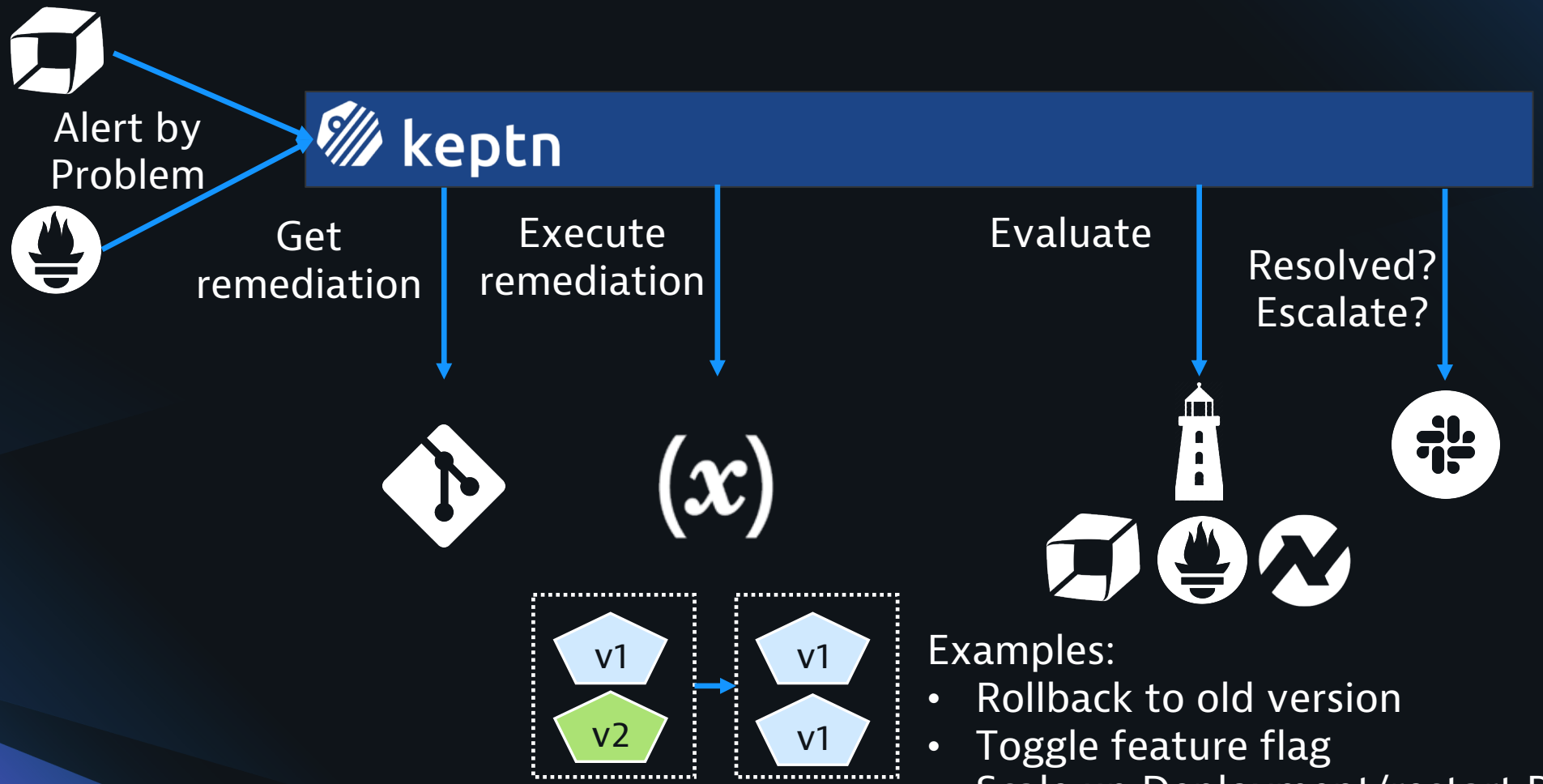
```
{
  "type": "sh.keptn.event.deployment.triggered",
  "id": "f2b878d3-03c0-4e8f-bc3f-454bc1b3d79d",
  "source": "keptn",
  "specversion": "1.0",
  "time": "2020-03-05T07:02:15.64489Z",
  ...
  "data": {
    "entity": { "project": "prj", "service": "svc",
                "stage":"hardening" }
    "deployment": {
      "strategy": "blue_green”
      “gitURL”: “https://github.com/prj/svc”
    }
  }
}
```

```
{
  "type": "sh.keptn.event.deployment.started",
  "source": ”argo-service",
  "time": "2020-03-05T07:02:16.64489Z",
  ...
  "data": {
    "deployment": {
      ”deployingVersion": "473ed93 "
    }
  }
}
{
  "type": "sh.keptn.event.deployment.finished",
  "source": ”argo-service",
  ...
  "data": {
    "deployment": {
      “status” : “Unknown|Errored|Succceeded”
      “result” : “Pass|Failed”
      "deploymentURIPublic":"https://mysvc.com/"
      "deploymentURILocal":"http://mysvc.cluster.local/"
    }
  }
}
```

Continuous Delivery Use Case by Example



Automatic Remediation Use Case by Example



Examples:

- Rollback to old version
- Toggle feature flag
- Scale up Deployment/restart Pods
- Clear disk
- YOUR manual operation tasks

Our expectations from this SIG

- Standardize a set of events for common tasks in Continuous Delivery and Automatic Remediation
  - For example events for deployment, test, evaluation, release, rollback, problem, remediation, etc.

Q & A