
KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Hands-off feature releases with Keptn, OpenFeature, and OpenTelemetry

Michael Beemer
Senior Product Manager
Dynatrace

Johannes Bräuer
Senior Product Manager
Dynatrace

KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Agenda

1. Feature Flagging
2. OpenFeature
3. Keptn
4. Demo
5. Recap

Quick intro to feature flagging

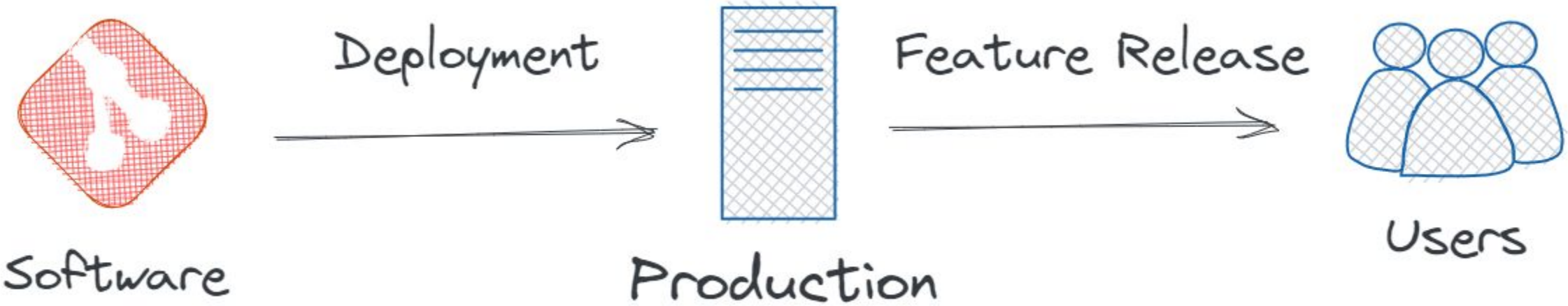
Software technique that enables teams to modify system behavior without changing code.

There are many different categories including:

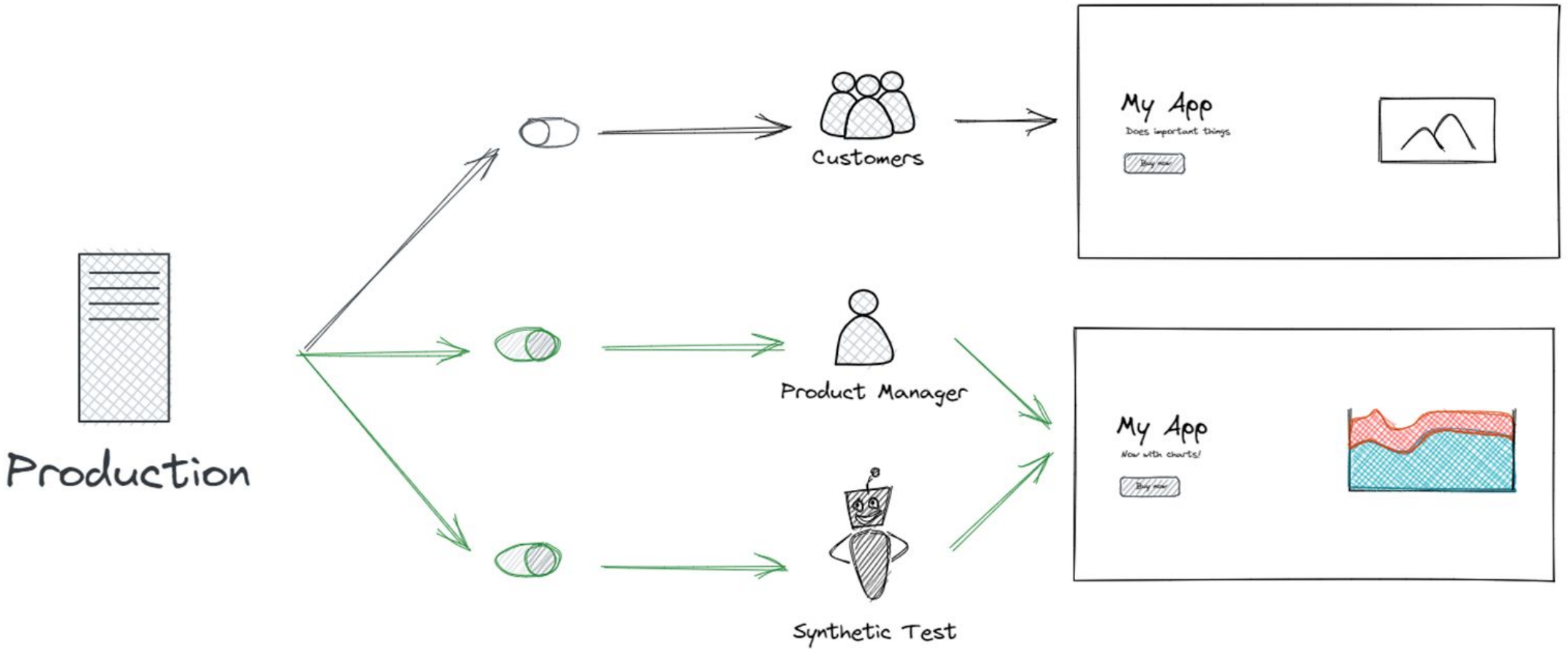
- Release toggles
- Experimental toggles
- Ops toggles
- Permissions toggles

```
function reticulateSplines() {
  if( featureIsEnabled("use-new-SR-algorithm") ) {
    return enhancedSplineReticulation();
  } else {
    return oldFashionedSplineReticulation();
  }
}
```

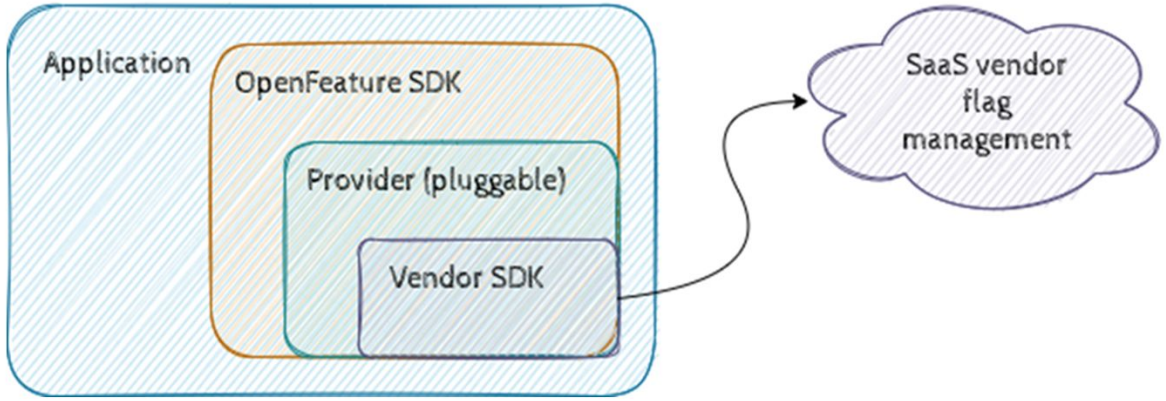
Decouple feature releases from deployments



Safely test in production



- Open standard, vendor-neutral feature flagging specification
- Broad industry and thought-leader support
- SDKs available in many popular languages
- Developer-first, cloud-native implementation



CLOUD NATIVE SANDBOX

Why develop a standard?

Easier language support for vendors

Easier migration between solutions

Common paradigms for ecosystem development

Consistent API for developers

Lower adoption risk

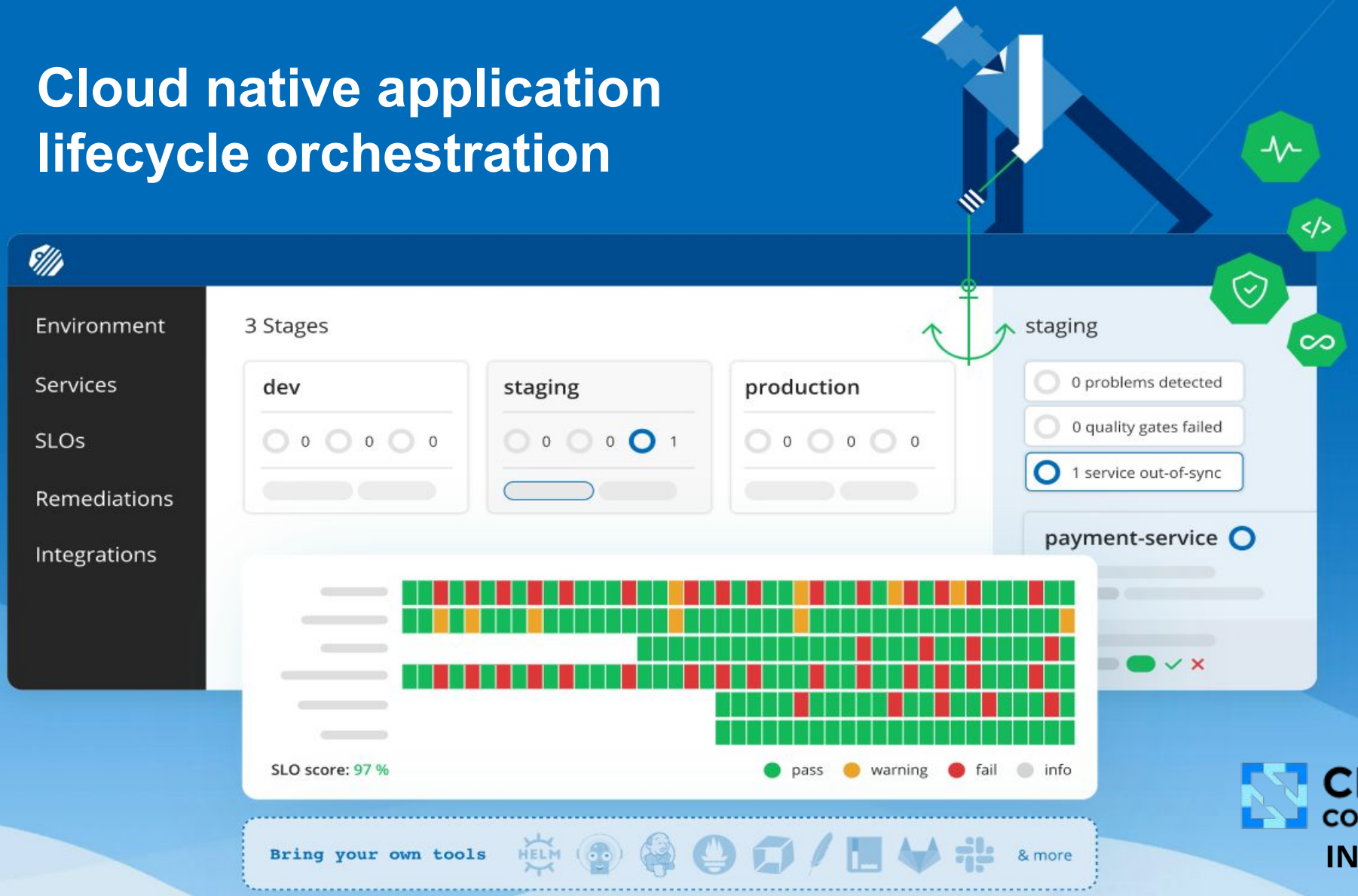
Simplified testing and local development

Conventional semantics for telemetry

keptn

KubeCon CloudNativeCon North America 2022

Cloud native application lifecycle orchestration



CLOUD NATIVE COMPUTING FOUNDATION
INCUBATING PROJECT

What problem does Keptn solve?

Automation pipelines are hard to maintain

“I am constantly reacting to ‘Pipeline Broken – please fix!’“

ERT Confidence at every turn™

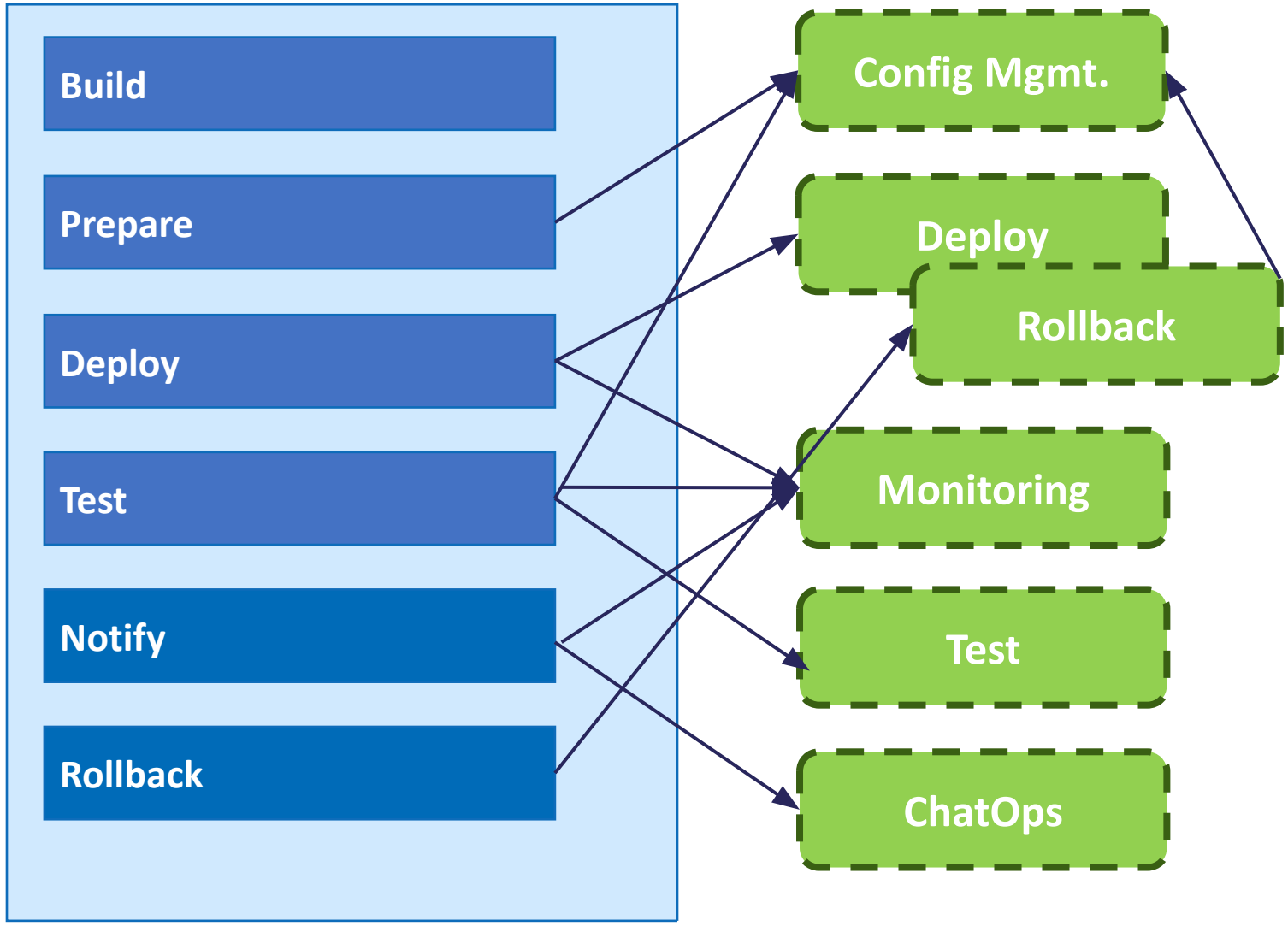
Christian Heckelmann
Senior DevOps Engineer

2800 projects

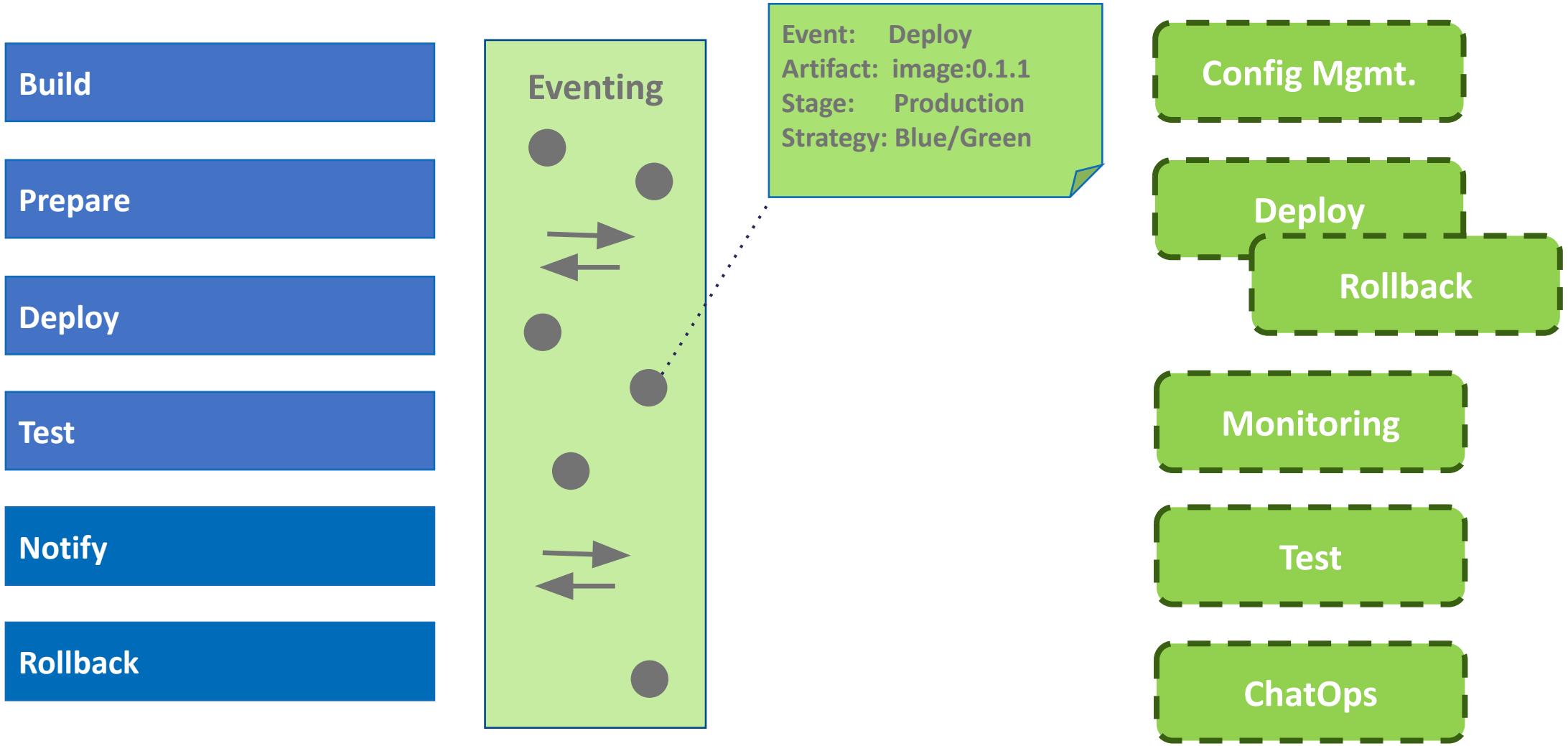
966 CI/CDs

```
image: gitcloud-cr.ert.com/efs/testing/docker/jmeter:latest
variables:
    GIT_STRATEGY: none
    QA_TARGET_REF: $PACKAGE_VERSION
before_script:
    - QA_TARGET_REF=v${PACKAGE_VERSION%.*}
script:
    - set -x
    - echo download QA branch $QA_TARGET_REF
    - curl -sg -G -o qa.zip -d "private_token=$GITLAB_TOKEN" h
    - unzip -o -q qa.zip && rm qa.zip
    - find . -maxdepth 1 -type d -name $QA_PROJECT_NAME-$QA_TA
    - bash -x qa/test
tags:
    - docker
    - linux
except:
    - tags
```

Remove hard dependencies and integrations



Remove hard dependencies and integrations



Which events to generate → Process Definition

Who consumes events → Tool Definition

Keptn Shipyard and Uniform - *Process versus Tooling*

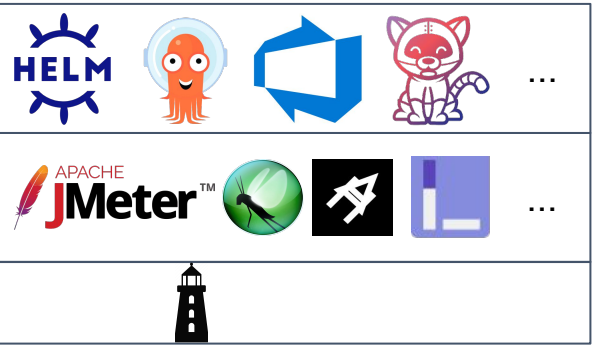
Process & environment definition

Transparent to tooling

Separation done via CloudEvents

Shipyard | *Uniform*

```
spec:
  stages:
    - name: "dev"
      sequences:
        - name: "delivery"
          tasks:
            - name: "deployment"
              properties:
                deploymentstrategy: "direct"
            - name: "test"
              properties:
                teststrategy: "functional"
            - name: "evaluation"
            - name: "release"
    - name: "staging"
      sequences:
        - name: "delivery"
          triggeredOn:
            - event: "dev.delivery.finished"
          tasks:
            - name: "deployment"
              properties:
                deploymentstrategy: "blue_green_service"
            - name: "test"
              properties:
                teststrategy: "performance"
            - name: "evaluation"
            - name: "release"
```



Keptn brings cloud native automation to your pipeline

Reduce your pipeline's **complexity** by letting keptn orchestrate **declarative, data-driven delivery and automation**

Pipeline Code Duplication:

	ada	config-service	hub-api	hubfront	hub-manager	ipim	lima-autoprov	lima-processing	signup-service
ada	-								
config-service	192	-							
hub-api	86	145	-						
hubfront	78	124	93	-					
hub-manager	98	151	210	113	-				
ipim	437	186	85	77	97	-			
lima-autoprov	179	552	132	115	144	173	-		
lima-processing	203	334	90	86	103	195	310	-	
signup-service	145	436	105	84	109	140	380	269	-
token-exchange	170	487	122	101	126	165	429	291	501

```
image: gitcloud-cr.ert.com/efs/testing/docker/jmeter:latest
variables:
    GIT_STRATEGY: none
    QA_TARGET_REF: $PACKAGE_VERSION
before_script:
    - QA_TARGET_REF=v${PACKAGE_VERSION%.*}
script:
    - set -x
    - echo download QA branch $QA_TARGET_REF
    - curl -sg -G -o qa.zip -d "private_token=$GITLAB_TOKEN" h
    - unzip -o -q qa.zip && rm qa.zip
    - find . -maxdepth 1 -type d -name $QA_PROJECT_NAME-$QA_TA
    - bash -x qa/testrun/perf-test.sh 'cleanup' 'qa'
tags:
    - docker
    - linux
except:
    - tags
```

```
spec:
  stages:
  - name: dev ···
  - name: staging ···
  - name: production
    sequences:
    - name: delivery
      triggeredOn:
      - event: staging.delivery.finished
      tasks:
      - name: monaco
      - name: deployment
        properties:
          deploymentstrategy: blue_green_service
      - name: test
        properties:
          teststrategy: performance
      - name: evaluation
      - name: release
    - name: rollback
      triggeredOn:
        - event: production.delivery.finished
          selector:
            match:
              result: "fail"
      tasks:
        - name: rollback
```

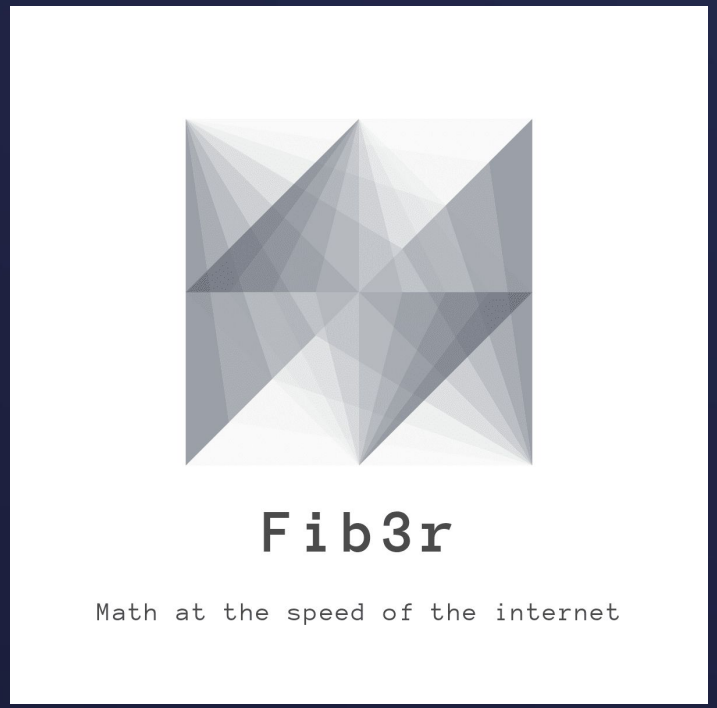
90% less automation code

Separation of process & tool

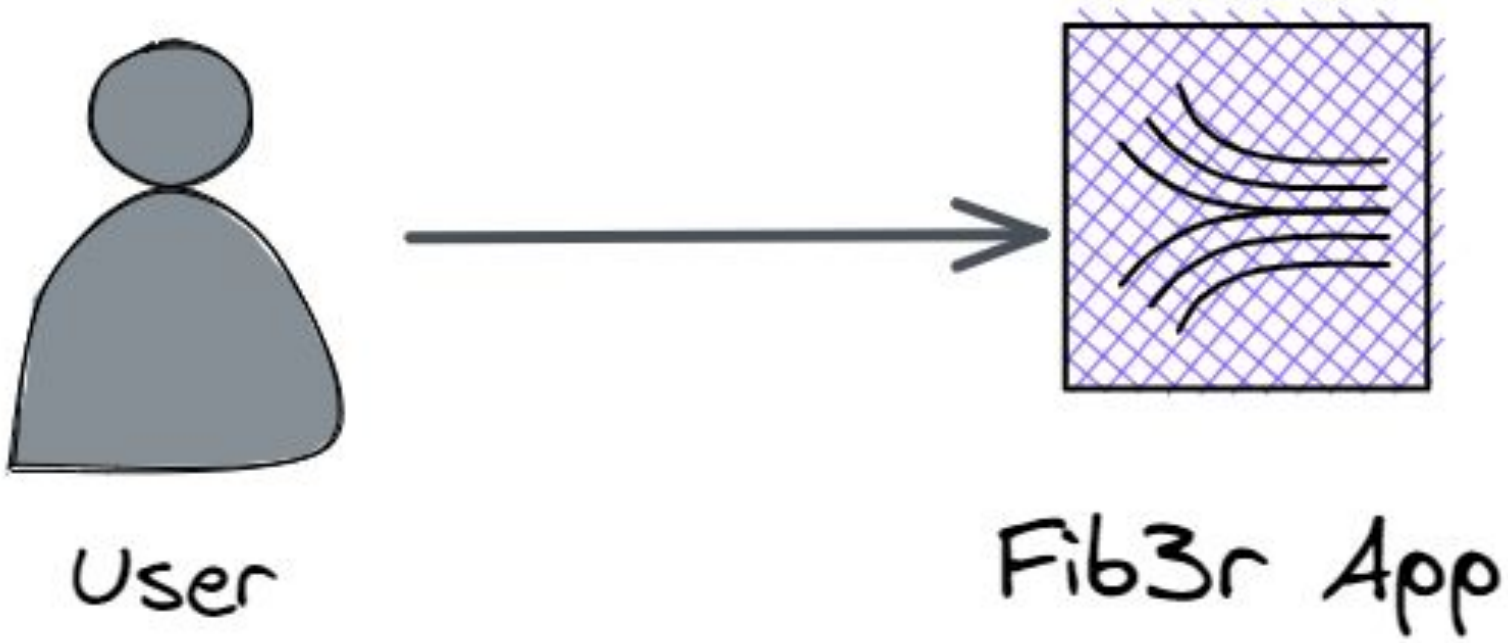
SLOs built-in

Connects with your tools

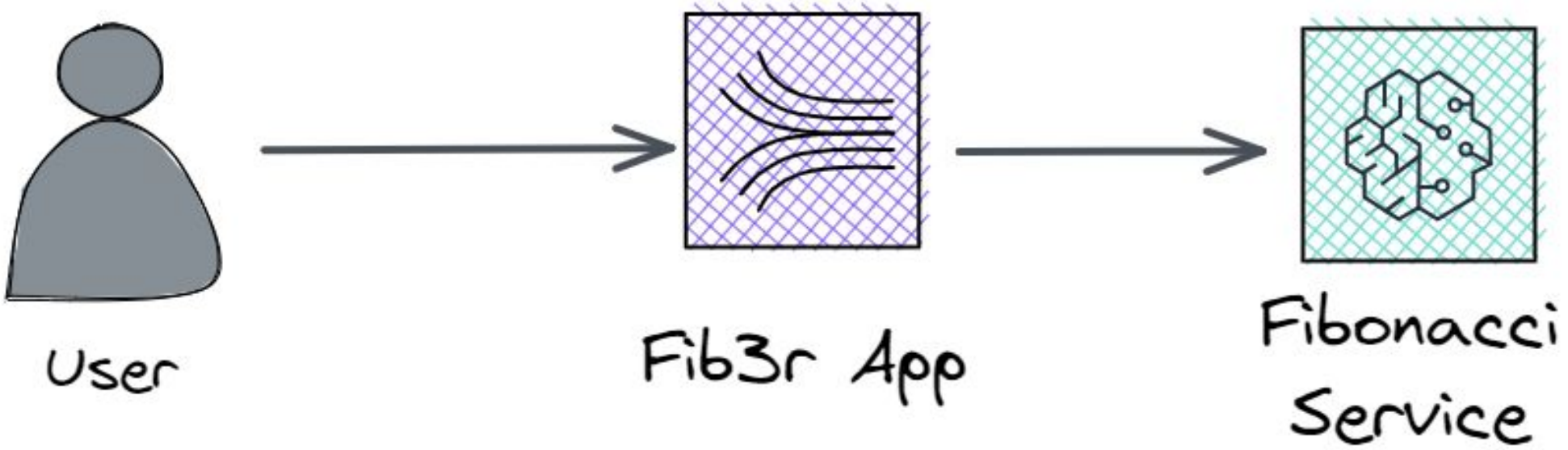
The **Fibonacci as a Service (FaaS)** industry is rapidly growing. To keep pace, the team at Fib3r must **re-architect** their system to handle the demand.



Fib3r's current architecture

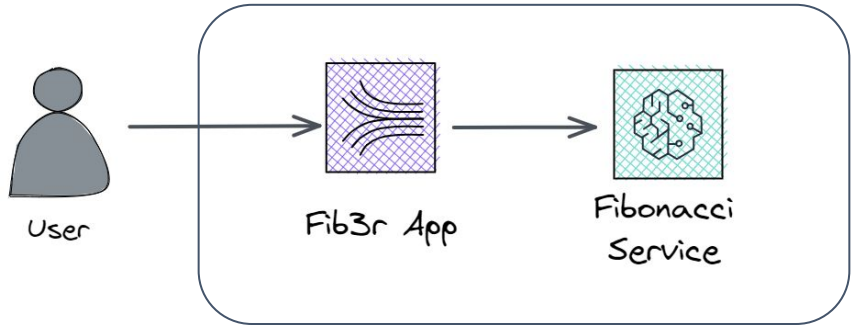


Fib3r’s next gen architecture



The plan

- Create a new Fibonacci microservice
  - Put the microservice behind a feature flag
  - Enable the feature only for automated tests
- Deploy and validate in a staging environment
- Deploy and validate in production
- Enable the new feature for all users

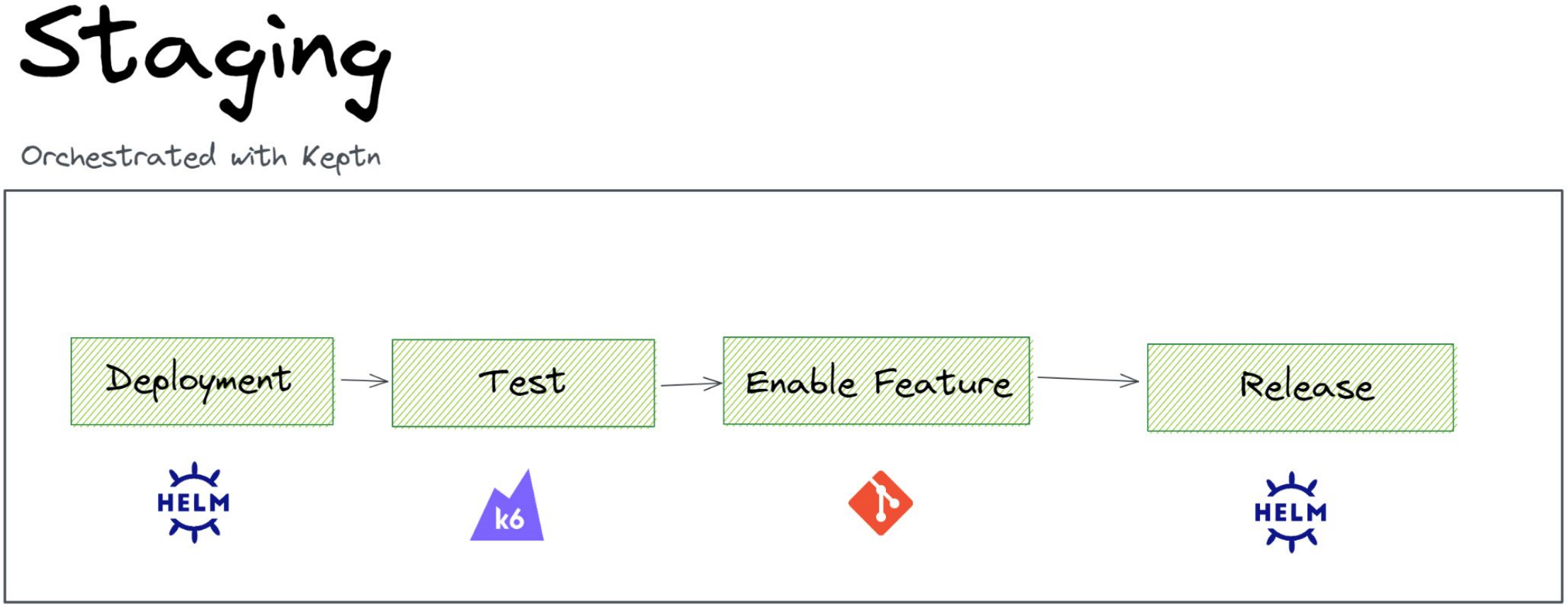


in production

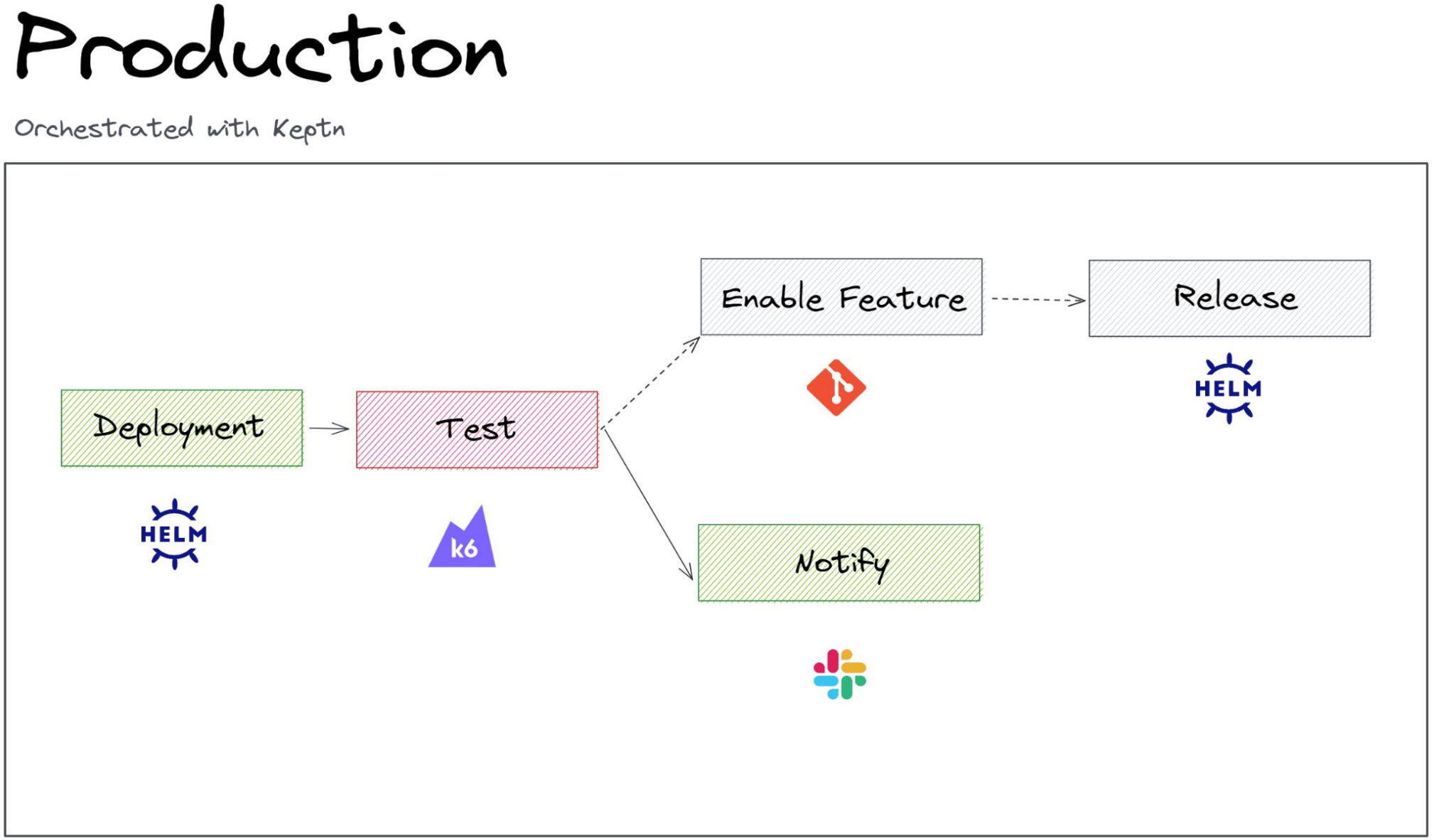
```
async calculateFibonacci(num: number) {
  const useRemoteFibService = await this.client.getBooleanValue(
    'use-remote-fib-service',
    false,
    { userAgent }
  );

  if (useRemoteFibService) {
    return callRemoteFibService(num);
  }

  return callLocalFibService(num);
}
```


DEMO TIME
meme-arsenal.ru

Recap - Staging looks good!



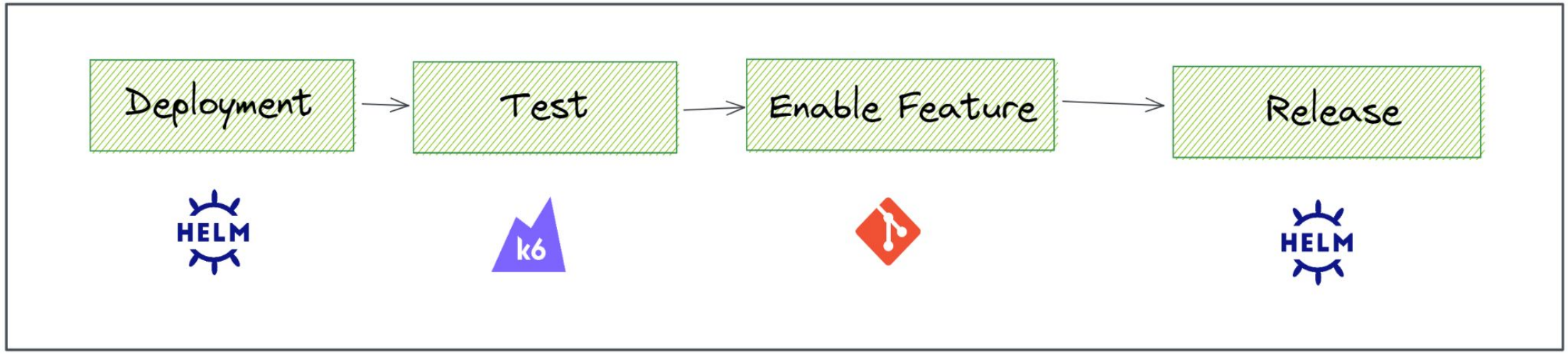
Recap - But there is an issue in prod



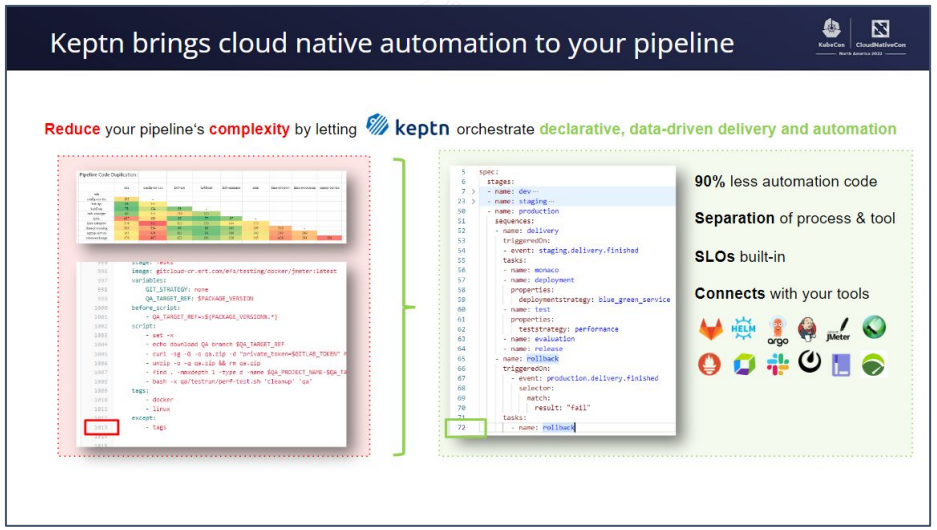
Recap - Issue resolved, new feature enabled

Production

Orchestrated with Keptn

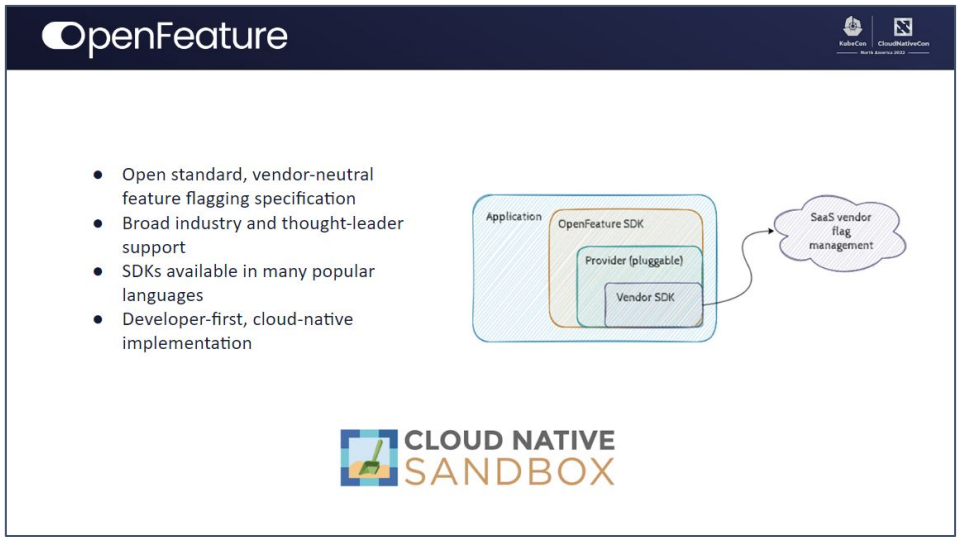


Takeaways



keptn

enables orchestrations that separates processes from tools



OpenFeature

enables standardized, vendor agnostic feature flagging

Find out more at...

KubeCon CloudNativeCon North America 2022

Keptn

https://keptn.sh/

@keptnProject

OpenFeature

https://openfeature.dev/

@OpenFeature

KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Let us know your thoughts!

Please scan the QR Code above to leave feedback on this session

Demo recording