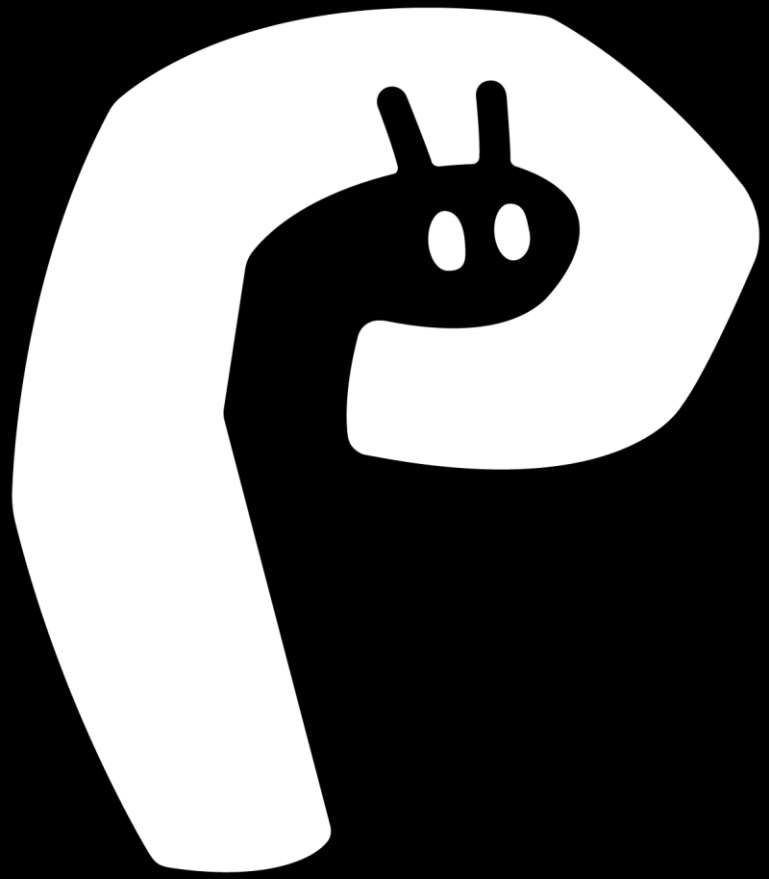
Project “Piper”



Oliver Nocon, SAP
March 4th, 2021

Our customers
distribute 78%
of world's food &
82% medical
devices

Our customers
produce more
than 79% of the
world's chocolate.

77% of the world's transaction
revenue
touches an SAP system

Our customers
produce more than
77% of the
world's beer.

Our customers
manufacture more
than 95,000
automobiles
per day.

Shared coded knowledge – ready to use

Piper “ready-made” pipelines

Shared coded knowledge – ready to use

Piper “ready-made” pipelines

Piper step library

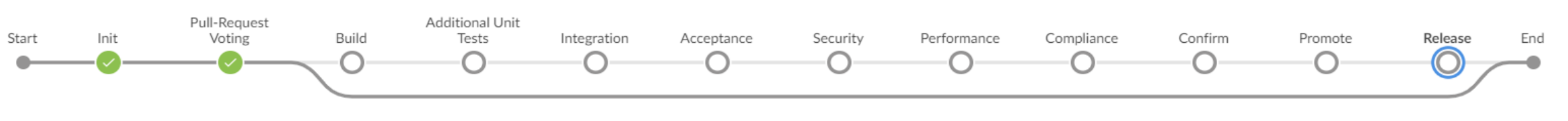
Shared coded knowledge – ready to use

Piper “ready-made” pipelines

Piper step library

Piper common configuration layer

Getting started …



Pipeline script (Jenkins)

```
@Library('piper-lib-os') _

piperPipeline script: this
```

Initial config (Pull-Request voting, other things can be added iteratively)

```
general:
  buildTool: npm
```

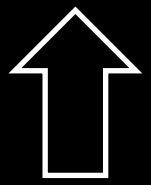
Separation of pipeline logic and configuration

Provided

Pipeline script (currently Jenkins)

```
@Library('piper-lib-os') _

piperPipeline script: this
```



innovate “centrally”

→ reduce teams’ cognitive load

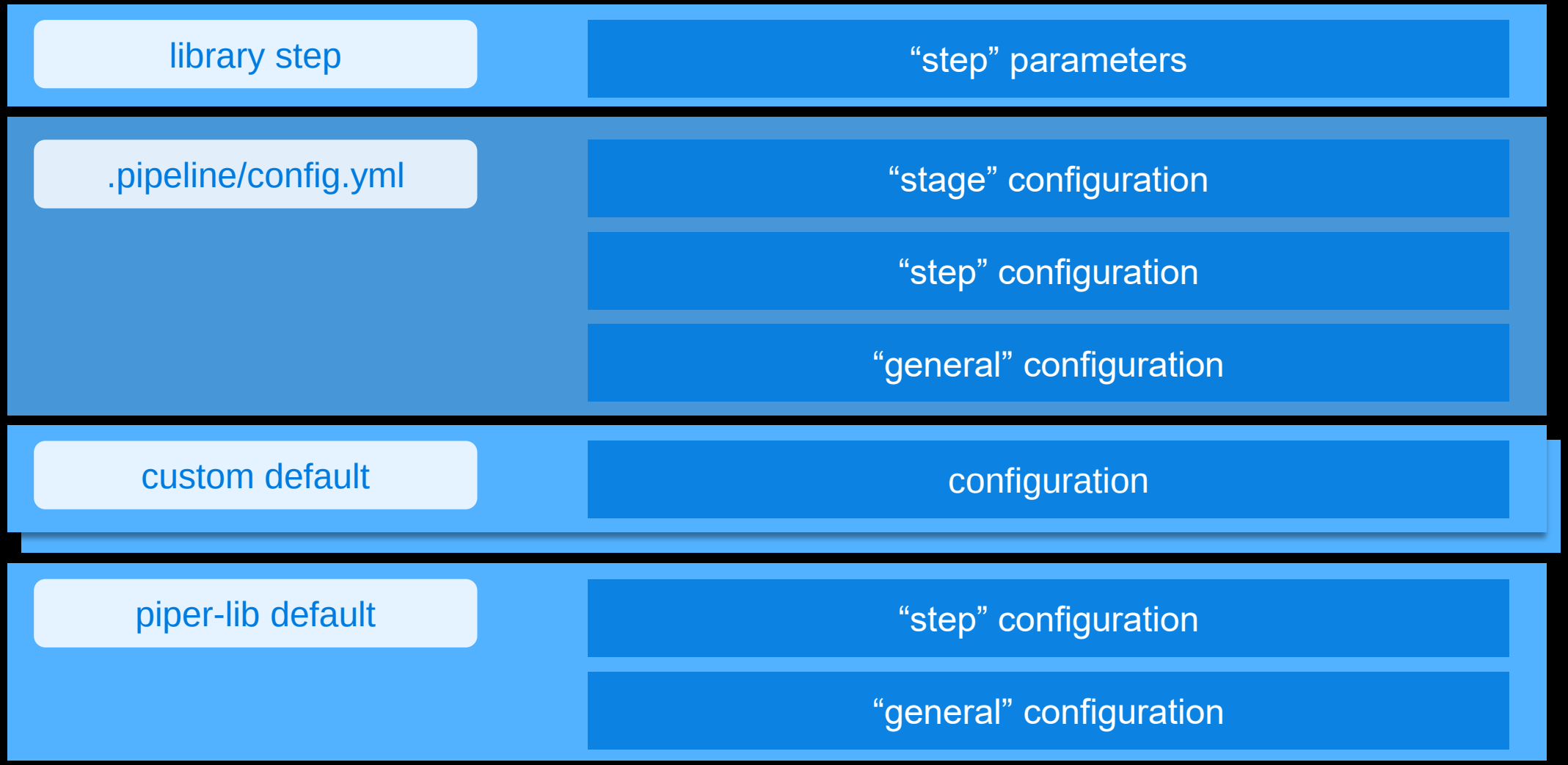
Team-owned

Configuration (universal)

```
general:
  buildTool: npm
stages:
  Security:
    verbose: true
steps:
  artifactPrepareVersion:
    shortCommitId: true
    ...
```

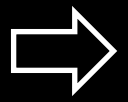
Stage extension

Common configuration layer



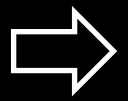
Example: Automatic versioning

```
general:
  buildTool: maven
steps:
  artifactPrepareVersion:
    shortCommitId: true
    ...
```



1.2.3-20210101010203_218a743

```
general:
  buildTool: npm
steps:
  artifactPrepareVersion:
    versioningType: library
    ...
```



1.2.3

https://www.project-piper.io/steps/artifactPrepareVersion/

Project “Piper” evolution

Jenkins

CD pipeline “ready-mades”

CD step library

Common configuration layer

Project “Piper” evolution

Jenkins	GitHub Actions	Tekton	…

CD pipeline “ready-mades”

CD step library

Common configuration layer

“Agnostic” Continuous Delivery

```
general:
  buildTool: npm
  vaultPath: my/demo/path
  vaultNamespace: my/demo/namespace
  vaultServerUrl: https://vault.demo.com
  # relevant for Jenkins only:
  vaultAppRoleTokenCredentialsId: vaultToken
  vaultAppRoleSecretTokenCredentialsId: vaultSecret
steps:
  artifactPrepareVersion:
    shortCommitId: true
```

Configure once – run anywhere …

Jenkins library

```
@Library(['piper-lib-os']) _

node {
    stage('Checkout & Version') {
        deleteDir()
        checkout scm
        artifactPrepareVersion script: this
    }
}
```

Shell

```
$ export PIPER_vaultAppRoleID=appRole
$ export PIPER_vaultAppRoleSecretID=appRoleSecret
$ git clone ...
$ piper artifactPrepareVersion
```

GitHub actions

```
name: CI
on:
  push:
    branches: [ master ]

env:
  PIPER_vaultAppRoleID: ${{ secrets.VAULTAPPROLEID }}
  PIPER_vaultAppRoleSecretID: ${{ secrets.VAULTAPPROLESECRETID }}

jobs:
  init:
    runs-on: [ self-hosted ]
    steps:
      - name: checkout
        uses: actions/checkout@v2
      - name: setVersion
        uses: SAP/project-piper-action@master
        with:
          piper-version: master
          command: artifactPrepareVersion
```

Tekton

to come …

What’s next?

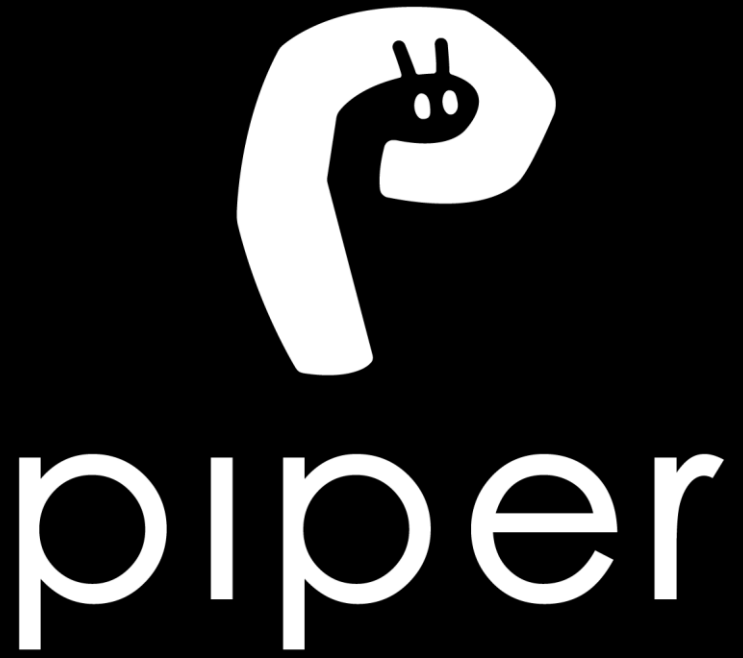
Agnostic ready-made pipelines (orchestrator-independent)

- Allow centrally provided pipeline flow changes
- Allow custom stage extensions (extend / replace)

Requirement: no mandatory team actions to consume an enhancement!

Possible solutions:

- Jenkins-X 2.0 DSL (prototype available)
- New proprietary DSL → not our favorite
- Consider as “not a use-case”, “agnostic” library sufficient? → we don’t think so …
- New CDF interop project?
- …



https://www.project-piper.io