KubeCon | CloudNativeCon

North America 2018

Enhancing Kubernetes

A Journey Through the KEP Process

Who are these two?

STEPHEN AUGUSTUS

Works at Red Hat as a Specialist Solution Architect on the OpenShift Tiger Team

Co-chair of SIG-Release, SIG-Azure, Sub-project lead for SIG-PM

Super power: Turning technical debt into action

JAICE SINGER DuMARS

Works at Google as the Cloud Native Open Source Strategy Program Manager

Co-chair of SIG-Architecture, Sub-project lead for SIG-PM, former Release Team Lead, emeritus chair of SIG-Release, SIG-Azure, on the Kubernetes Code of Conduct Committee

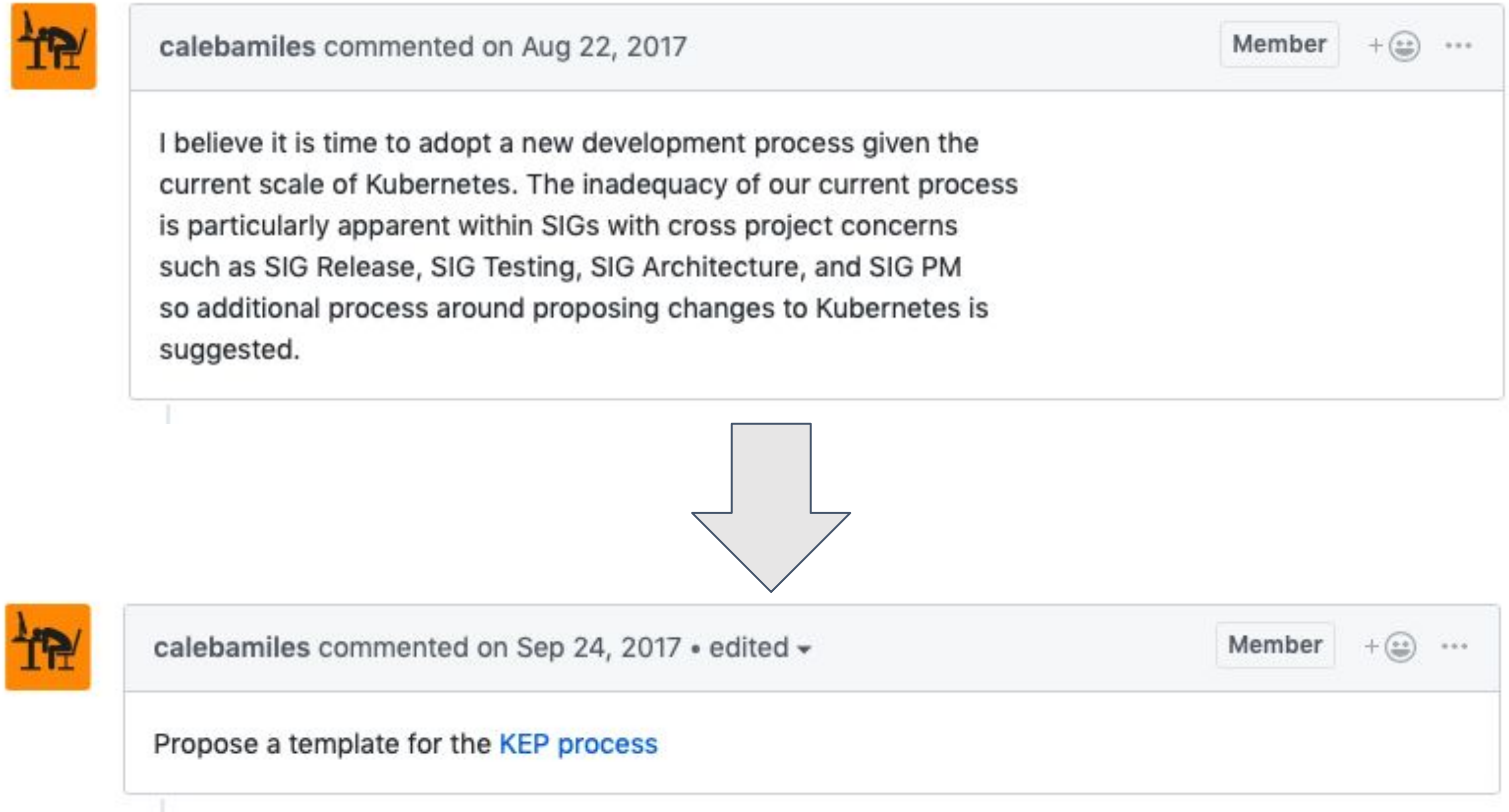
Super power: surviving lightning strikes

KEP History

Propose adopting the Rust RFC process #967

Merged **bgrant0607** merged 7 commits into `kubernetes:master` from `calebamiles:propose-adopting-rust-rfc-process` on Sep 26, 2017

Caleb Miles - Technical Program Manager, Google



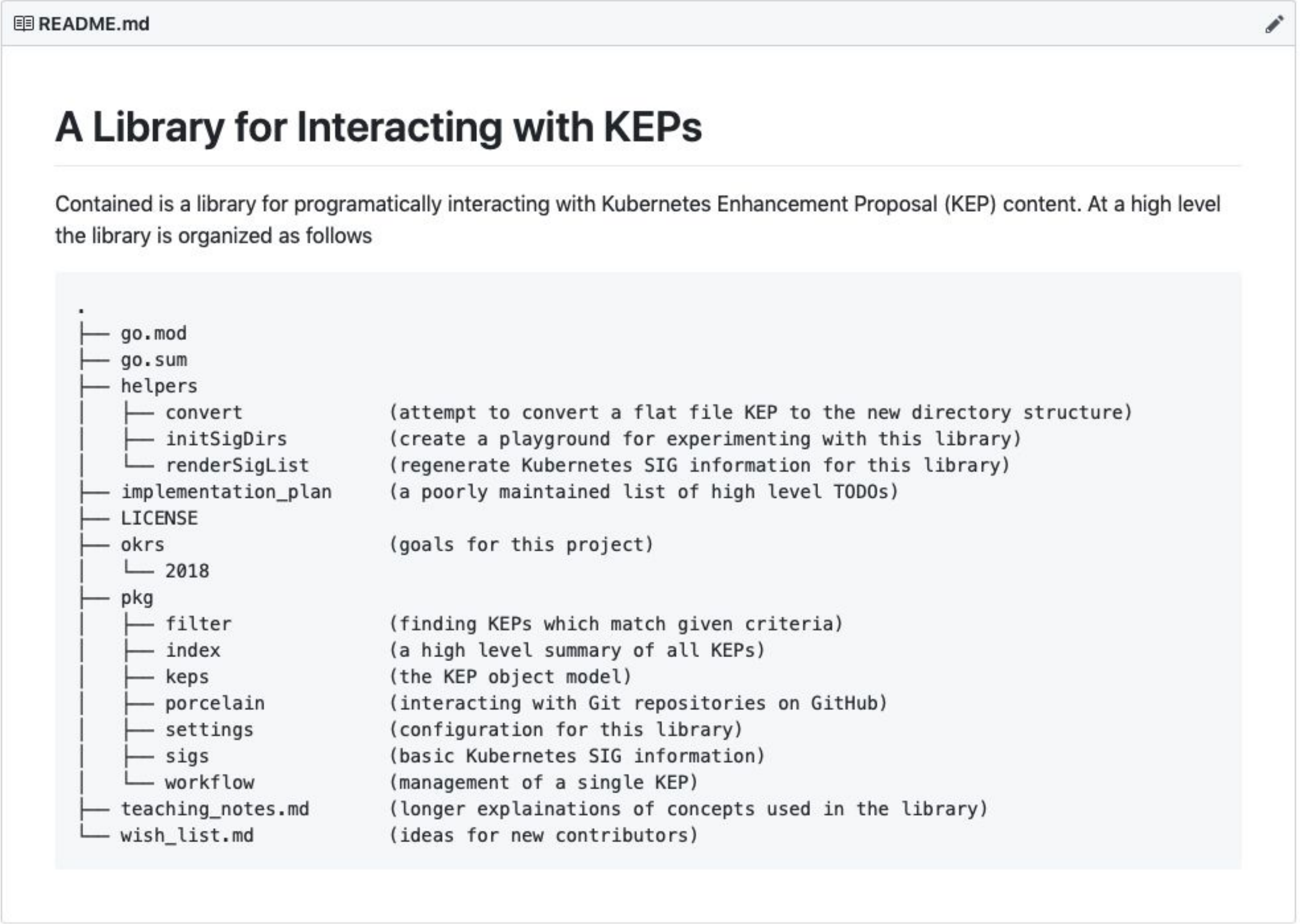
calebamiles commented on Aug 22, 2017

I believe it is time to adopt a new development process given the current scale of Kubernetes. The inadequacy of our current process is particularly apparent within SIGs with cross project concerns such as SIG Release, SIG Testing, SIG Architecture, and SIG PM so additional process around proposing changes to Kubernetes is suggested.

calebamiles commented on Sep 24, 2017 • edited

Propose a template for the KEP process

KEP History



README.md

A Library for Interacting with KEPs

Contained is a library for programatically interacting with Kubernetes Enhancement Proposal (KEP) content. At a high level the library is organized as follows

```
.
├── go.mod
├── go.sum
├── helpers
│   ├── convert             (attempt to convert a flat file KEP to the new directory structure)
│   ├── initSigDirs         (create a playground for experimenting with this library)
│   └── renderSigList       (regenerate Kubernetes SIG information for this library)
├── implementation_plan     (a poorly maintained list of high level TODOs)
├── LICENSE
├── okrs                    (goals for this project)
│   └── 2018
├── pkg
│   ├── filter              (finding KEPs which match given criteria)
│   ├── index               (a high level summary of all KEPs)
│   ├── keps                (the KEP object model)
│   ├── porcelain           (interacting with Git repositories on GitHub)
│   ├── settings            (configuration for this library)
│   ├── sigs                (basic Kubernetes SIG information)
│   └── workflow            (management of a single KEP)
├── teaching_notes.md       (longer explainations of concepts used in the library)
└── wish_list.md            (ideas for new contributors)
```

A strong foundation for a cli and other automation in progress!

Why KEPs?

Proposals came in many different shapes, sizes, and forms

- Proposals were inconsistent
- Lifecycle management is necessary
- Alleviate duplication of effort especially in the release process

if it works for other projects like Rust and Python...

Bringing Consistency Forward

photo: Sandip Dey

- Easier to review
- Lifecycle can be tracked in metadata, keeping everything in source control
- Single artifact can serve multiple purposes

When do you need a KEP?

"If an enhancement would be described in either written or verbal communication to anyone besides the KEP author or developer then consider creating a KEP."

Some things may *seem* easy to describe, until you have to actually do it.

Anatomy of a KEP

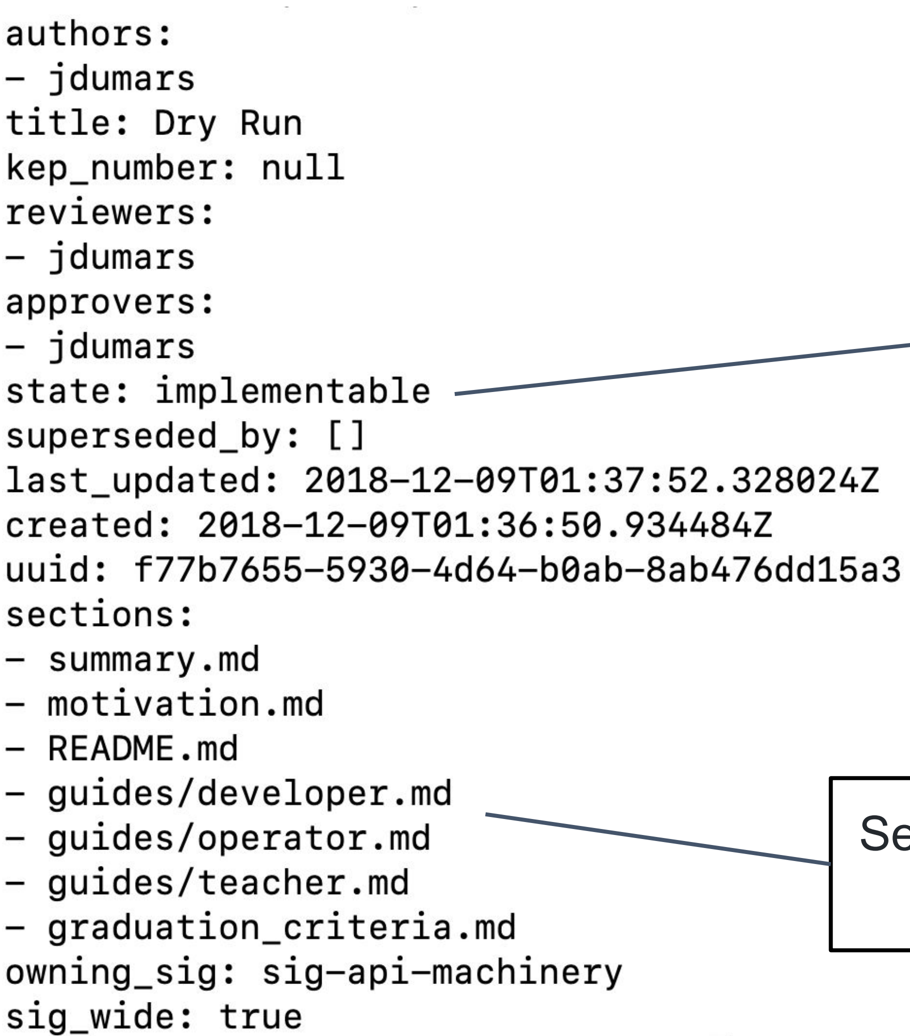
METADATA

MOTIVATION

SUMMARY

ARTIFACTS

Metadata



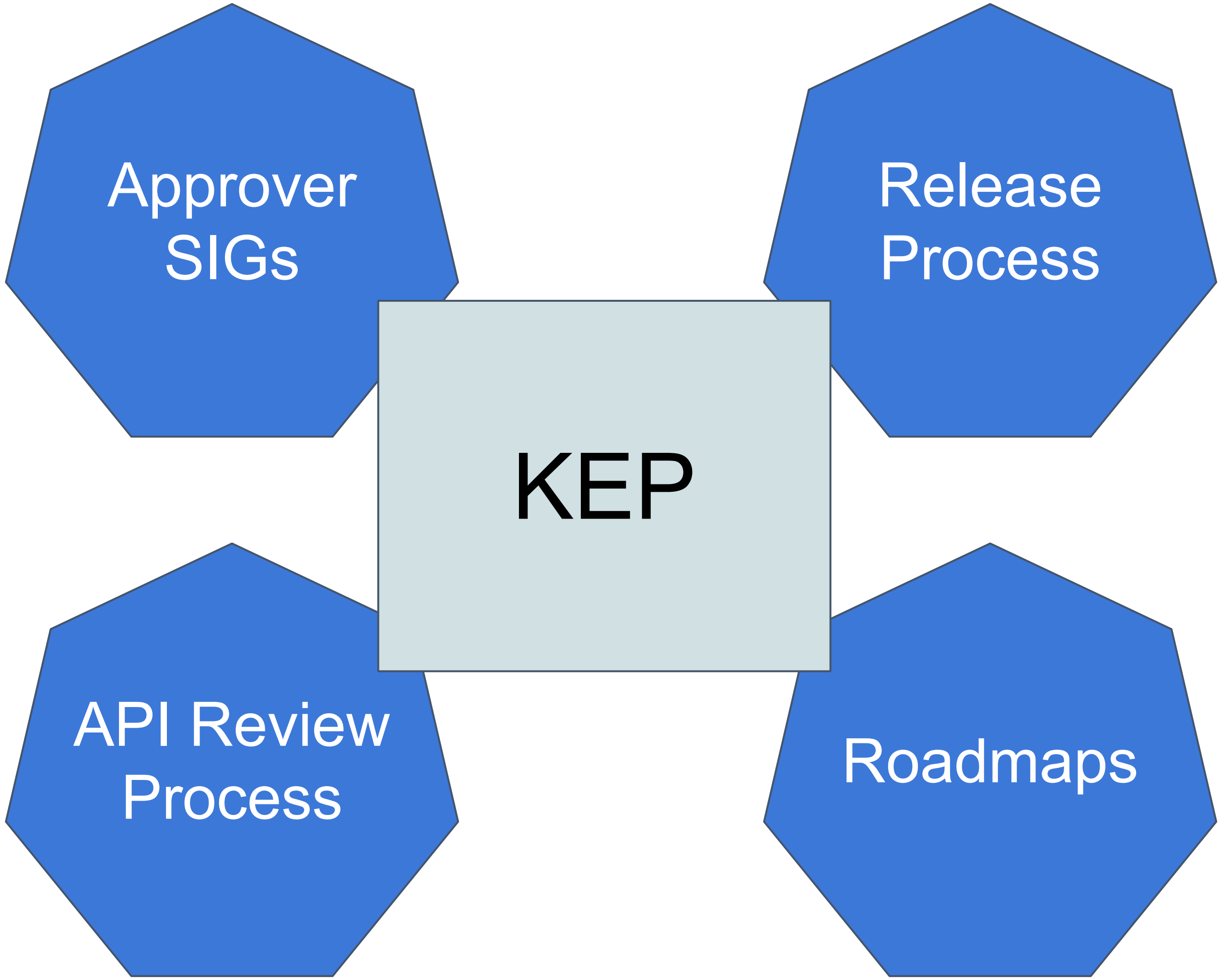
```
authors:
- jdumars
title: Dry Run
kep_number: null
reviewers:
- jdumars
approvers:
- jdumars
state: implementable
superseded_by: []
last_updated: 2018-12-09T01:37:52.328024Z
created: 2018-12-09T01:36:50.934484Z
uuid: f77b7655-5930-4d64-b0ab-8ab476dd15a3
sections:
- summary.md
- motivation.md
- README.md
- guides/developer.md
- guides/operator.md
- guides/teacher.md
- graduation_criteria.md
owning_sig: sig-api-machinery
sig_wide: true
```

Must be one of **provisional**, **implementable**, **implemented**, **deferred**, **rejected**, **withdrawn**, or **replaced**.

Sections added as the KEP state changes and moves toward implementation

KEPs Serve Many Audiences

Hey! It's a lighthouse!



KEP

Approver SIGs

Release Process

API Review Process

Roadmaps

Motivation

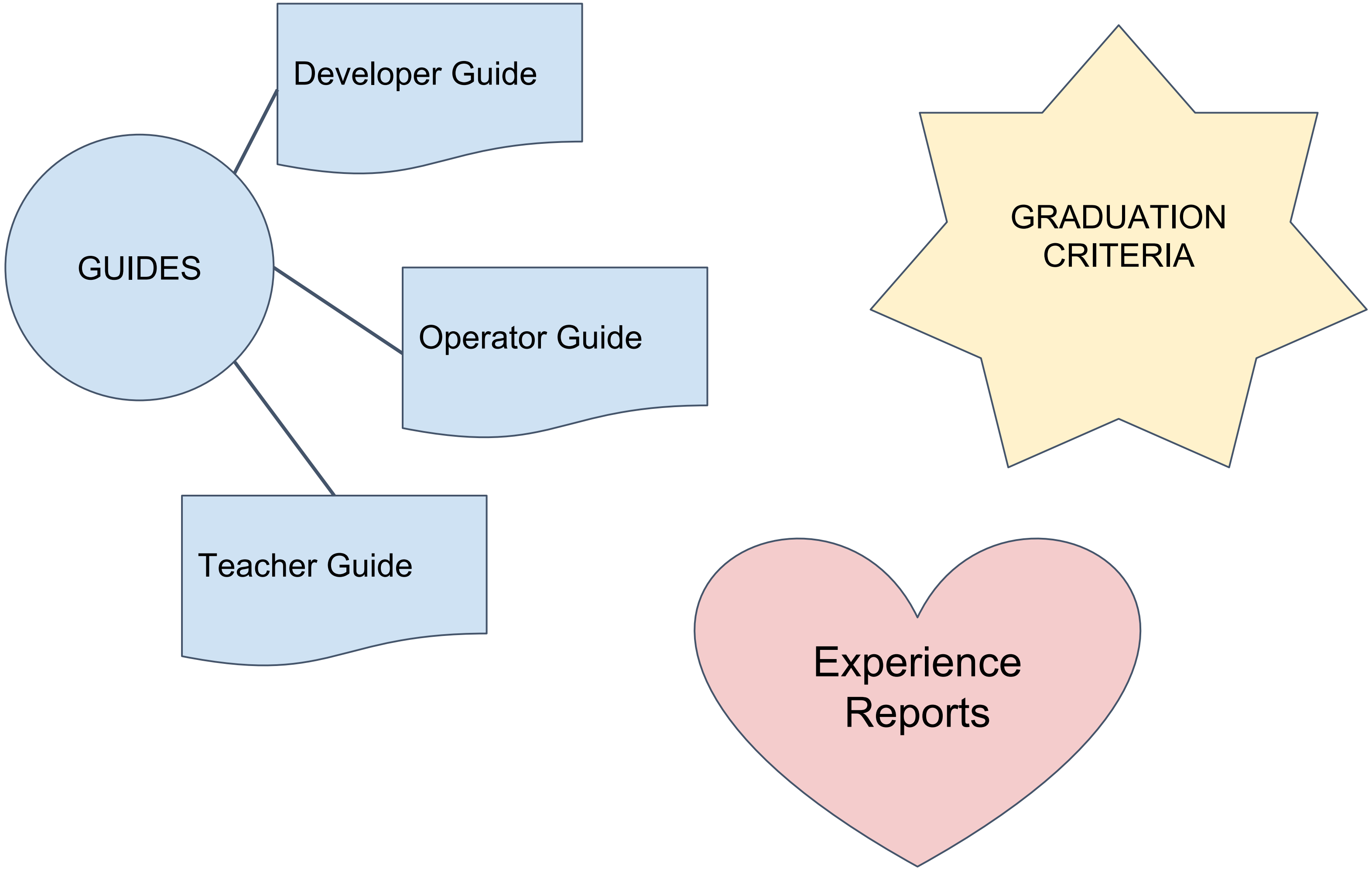
What is the **need**?

- What is the problem being solved?

- Describe the significance of the problem well enough that everyone can understand why we should spend time solving it and maintaining a solution

The motivation helps reviewers and participants decide if they want to prioritize this work

Artifacts


GUIDES

Developer Guide

Operator Guide

Teacher Guide

GRADUATION CRITERIA

Experience Reports

Random boats for no reason.

KEP States

- **`draft`**:
  - The KEP has been authored but not reviewed

- **`provisional`**:
  - Proposed and actively being defined. The owning SIG has accepted that this is work that needs to be done.

- **`implementable`**:
  - The approvers have approved this KEP for implementation.

Other KEP End States

- *`deferred`*:
  - The KEP is proposed but not actively being worked on.
- *`rejected`*:
  - The approvers and authors have decided that this KEP is not moving forward. The KEP is kept around as a historical document.
- *`withdrawn`*:
  - The KEP has been withdrawn by the authors.
- *`replaced`*:
  - The KEP has been replaced by a new KEP. The **`superseded-by`** metadata value should point to the new KEP.

KEP Lifecycle

INIT

KEP AUTHOR ACTION

- Generates templates
- Creates initial metadata
- Gives a chance to see if the cli is set up properly

KEP Lifecycle

PROPOSE

KEP AUTHOR ACTION

KEP STATE: DRAFT

- Submitter has populated the templates
- Motivation is clear
- Preps for SIG reviewers

KEP Lifecycle

ACCEPT

SIG APPROVER ACTION

KEP STATE: PROVISIONAL

- SIG has reviewed and commits to shepherding it towards implementation
- Not tied to a release yet
- Not ready to have PRs associated yet

KEP Lifecycle

PLAN

KEP AUTHOR ACTION

KEP STATE: PROVISIONAL

- KEP author(s) iterate on submitted documentation, getting it ready to be worked by the SIG
- May be a lengthy period as artifacts are developed as early as possible
- May feed into API review process next

KEP Lifecycle

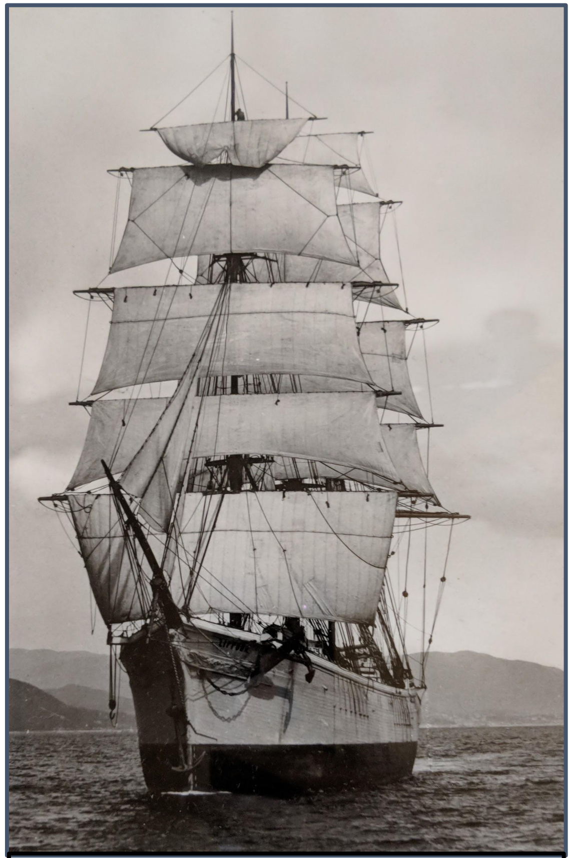
APPROVE

SIG APPROVER ACTION

KEP STATE: IMPLEMENTABLE

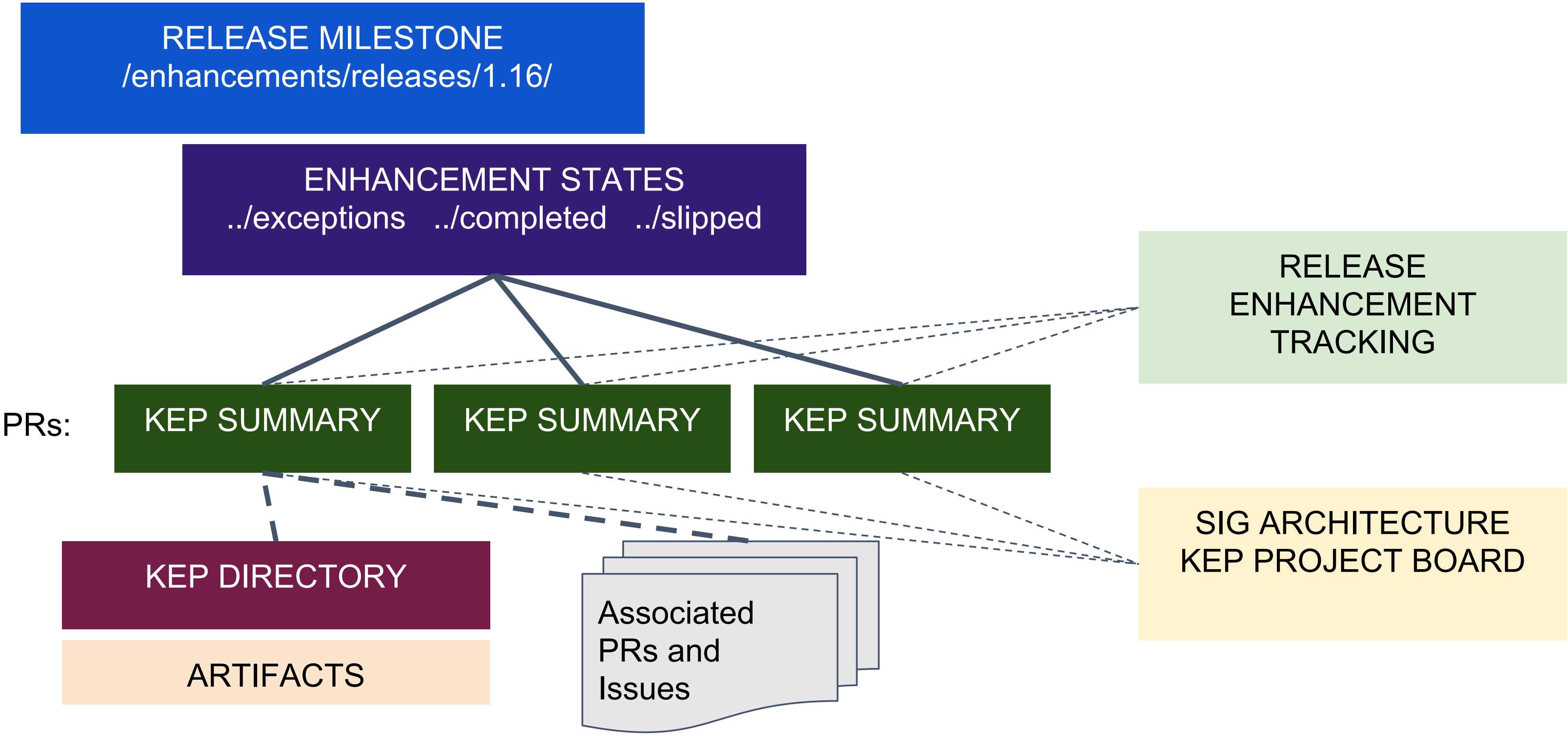
- KEP reviewer/approver is satisfied that the contents of the KEP are in alignment with the SIG, such that coding and other activities may proceed. It is now a trusted document of the SIG.
- API reviews should be complete

Future States



Been too long since we've seen a ship. I know this is about the future, and that is an old ship, sorry.

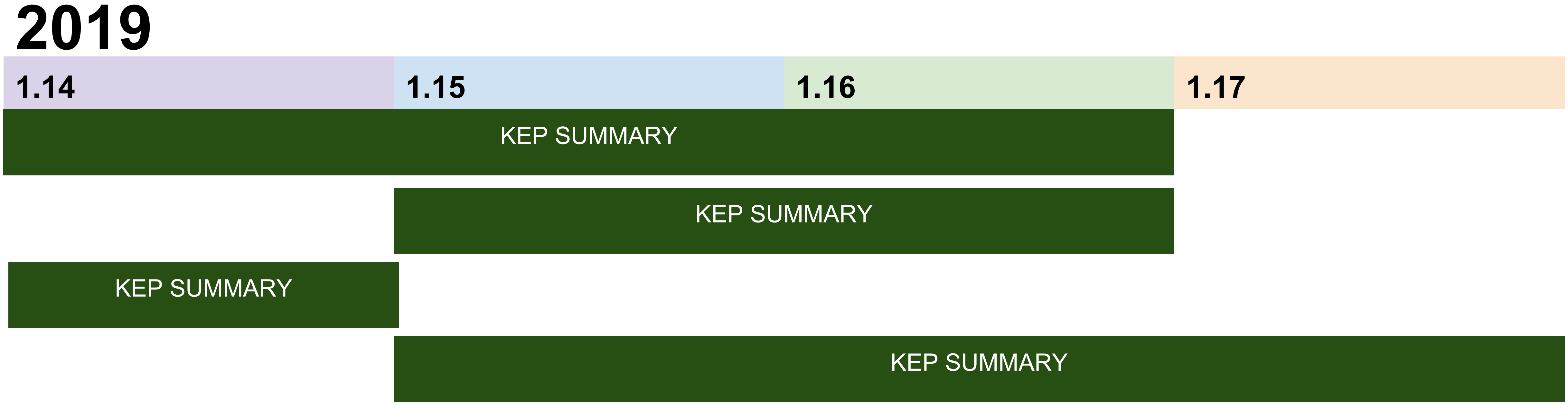
Release Enhancements Management



RELEASE MILESTONE
/enhancements/releases/1.16/
ENHANCEMENT STATES
../exceptions ../completed ../slipped
RELEASE ENHANCEMENT TRACKING
PRs:
KEP SUMMARY
KEP SUMMARY
KEP SUMMARY
KEP DIRECTORY
ARTIFACTS
Associated PRs and Issues
SIG ARCHITECTURE KEP PROJECT BOARD

Future States

The Elusive Roadmap...



2019

1.14

1.15

1.16

1.17

KEP SUMMARY

KEP SUMMARY

KEP SUMMARY

KEP SUMMARY

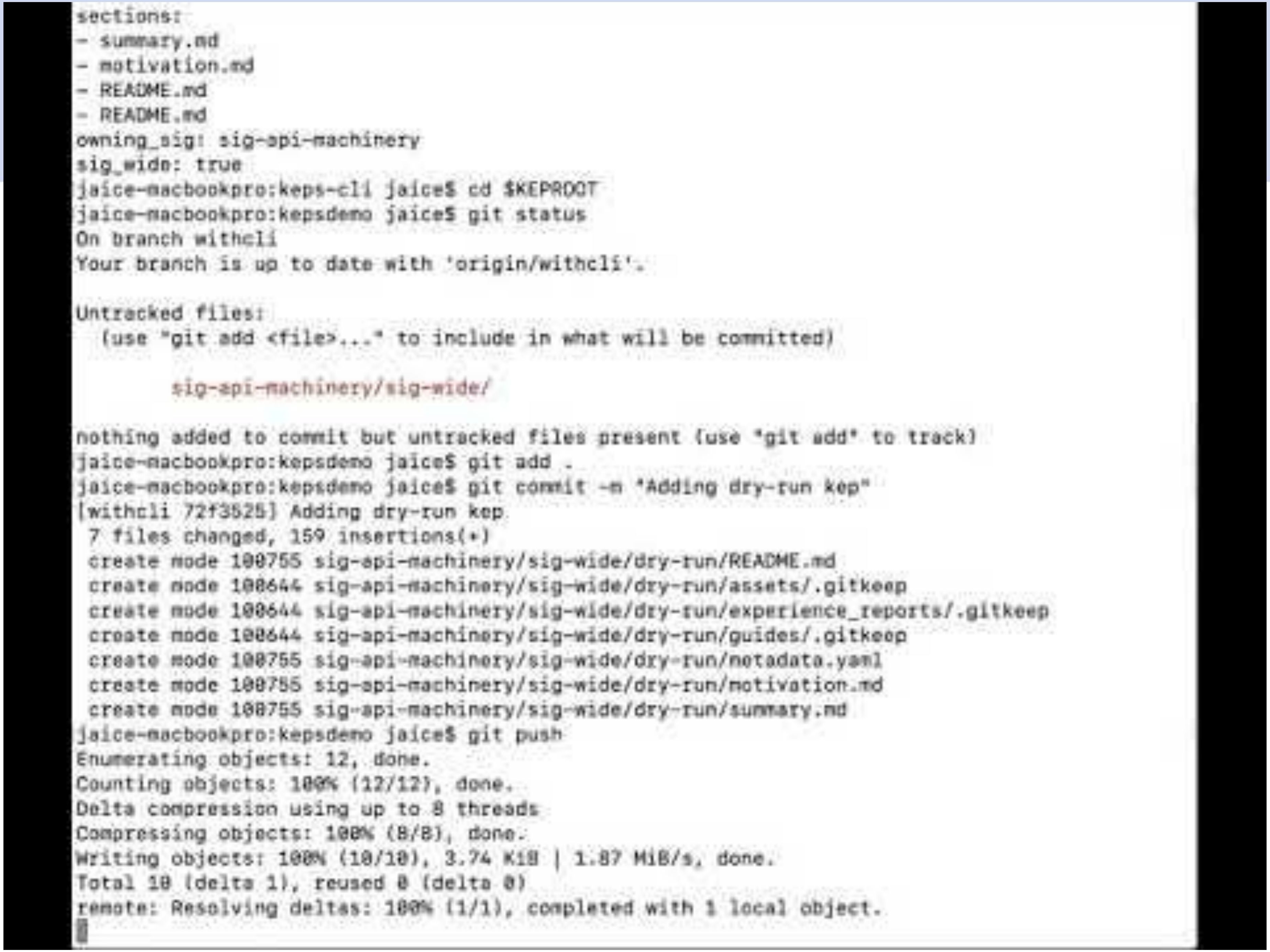
The better planned the KEPs, the clearer our roadmap as a project!

AMA With Caleb Miles

Caleb:

Twitter:	Don’t Have One
Slack:	calebamiles
GitHub:	calebamiles
email:	calebmiles@google.com

DEMO



```
sections:
- summary.md
- motivation.md
- README.md
- README.md
owning_sig: sig-api-machinery
sig_wide: true
jaice-macbookpro:keps-cli jaice$ cd $KEPROOT
jaice-macbookpro:kepsdemo jaice$ git status
On branch withcli
Your branch is up to date with 'origin/withcli'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        sig-api-machinery/sig-wide/

nothing added to commit but untracked files present (use "git add" to track)
jaice-macbookpro:kepsdemo jaice$ git add .
jaice-macbookpro:kepsdemo jaice$ git commit -m "Adding dry-run kep"
[withcli 72f3525] Adding dry-run kep
 7 files changed, 159 insertions(+)
 create mode 100755 sig-api-machinery/sig-wide/dry-run/README.md
 create mode 100644 sig-api-machinery/sig-wide/dry-run/assets/.gitkeep
 create mode 100644 sig-api-machinery/sig-wide/dry-run/experience_reports/.gitkeep
 create mode 100644 sig-api-machinery/sig-wide/dry-run/guides/.gitkeep
 create mode 100755 sig-api-machinery/sig-wide/dry-run/metadata.yaml
 create mode 100755 sig-api-machinery/sig-wide/dry-run/motivation.md
 create mode 100755 sig-api-machinery/sig-wide/dry-run/summary.md
jaice-macbookpro:kepsdemo jaice$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 3.74 KiB | 1.87 MiB/s, done.
Total 10 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
```

All the things....

Stephen:

Twitter:	@stephenaugustus
Slack:	justaugustus
GitHub:	justaugustus
email:	stephen@agst.us

Jaice:

Twitter:	@jaydumars
Slack:	jdumars
GitHub:	jdumars
email:	jdumars@gmail.com

Questions?

One last boat.