Welcome, new Beam contributor!

First Things First

Beam belongs to the Apache Software Foundation and operates under Apache license 2.0

What does that mean to you as a contributor?

The Apache License is intended to be a permissive, and business-friendly license.
Contributors to any Apache project *sign an Individual Contributor License Agreement (ICLA)* that allows your contributions to be consistent with the Apache License.

All Apache Beam contributors Adhere to the Apache Code of Conduct, please take time to read it.

It will be also helpful to introduce yourself to "The Apache Way" to get familiar with the customs and the etiquette of the ASF. You will help you to understand how decision making works, how voting process goes etc. It is very important!

First Things First

Learn more about history, governance and structure of Apache Beam

Following slides will give you more in depth introduction to the Apache Beam, but don’t worry it is not long!

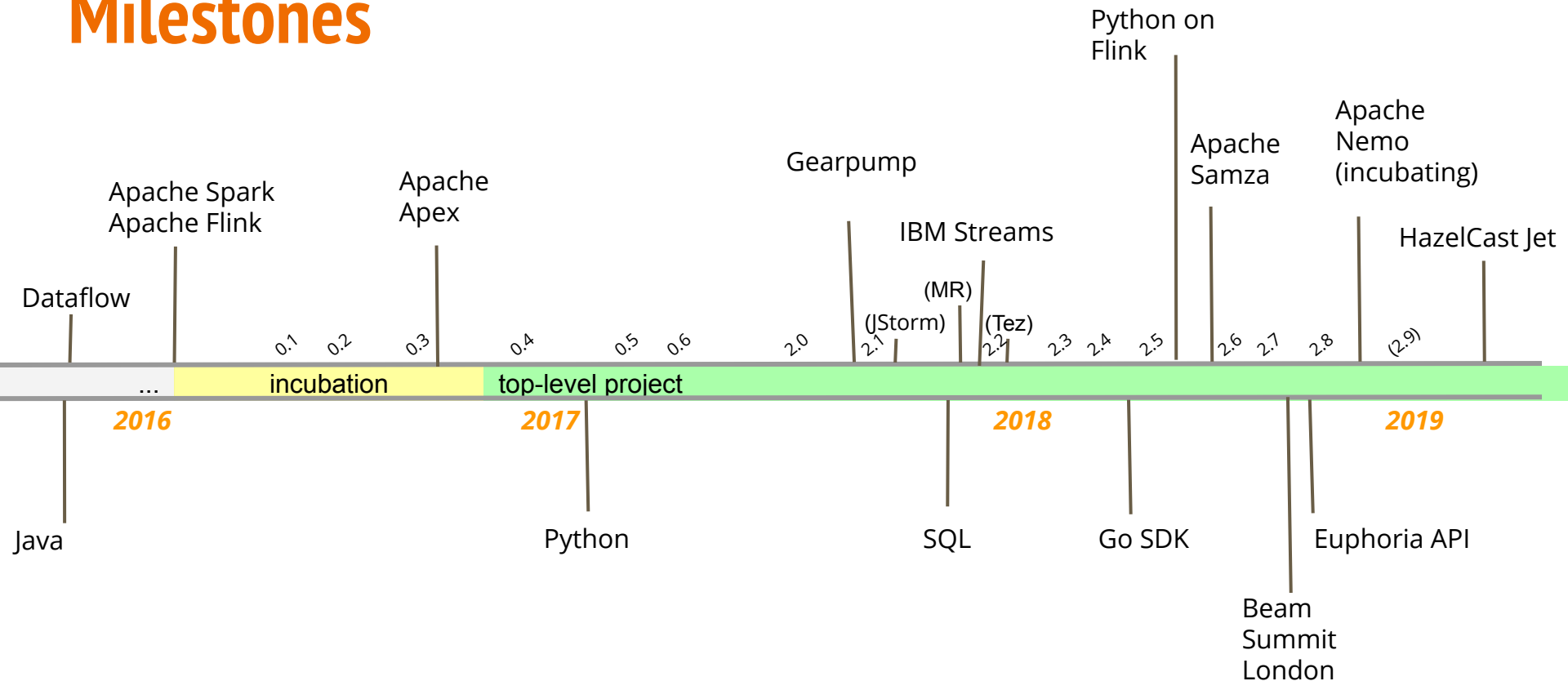
What is Apache Beam?

“Provide unified programming model for both batch and streaming data processing.”

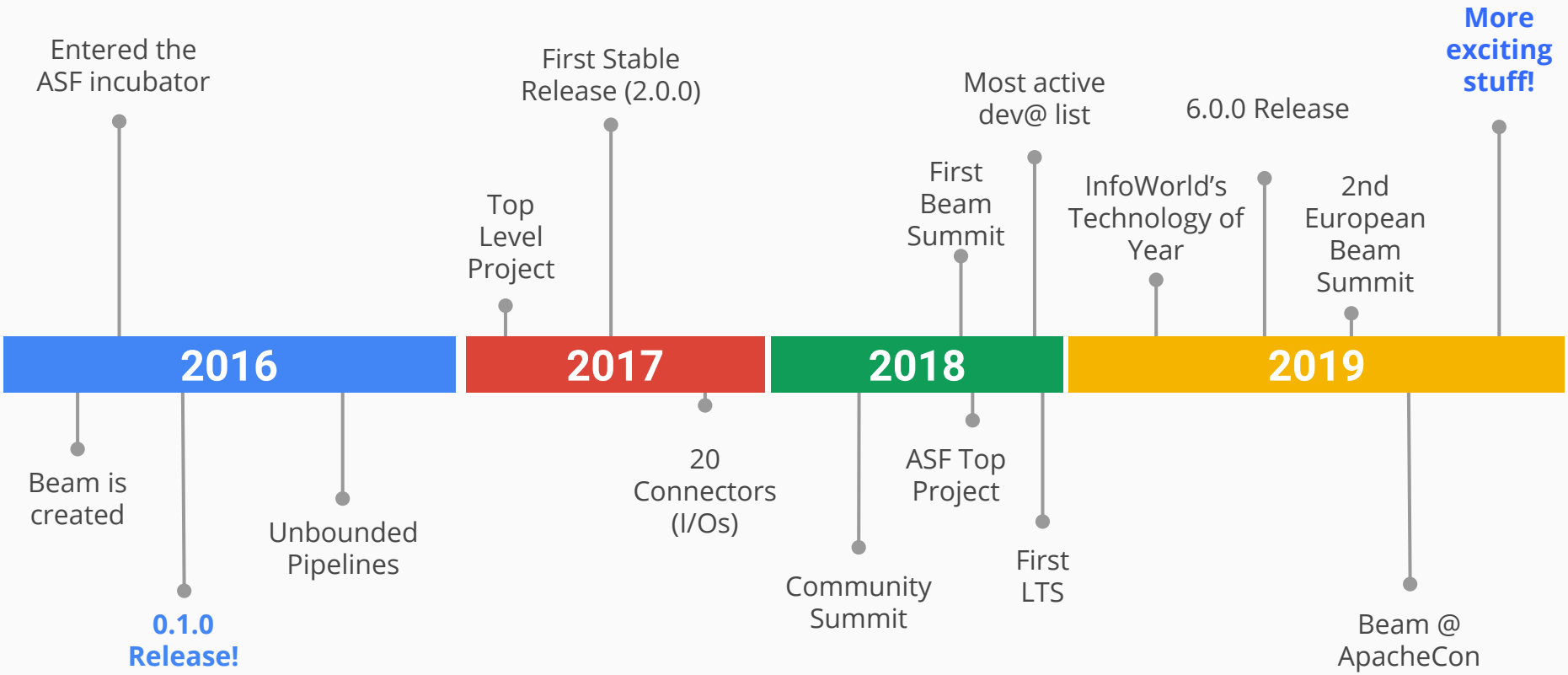
By enabling

- *efficient execution across diverse distributed execution engines*
- *providing extensibility points for connecting to different technologies and user communities*

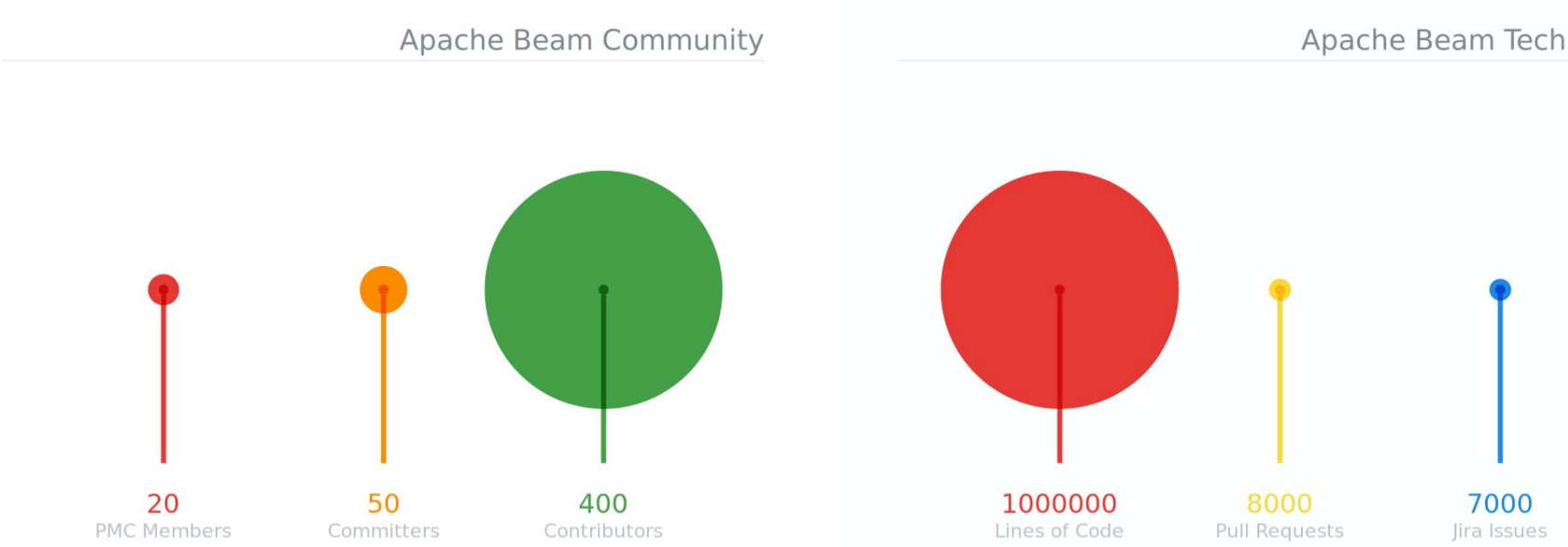
Milestones


Python on Flink
Apache Samza
Apache Nemo (incubating)
HazelCast Jet
Gearpump
IBM Streams
Apache Spark
Apache Flink
Apache Apex
Dataflow
(MR)
(JStorm)
(Tez)
0.1
0.2
0.3
0.4
0.5
0.6
2.0
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8
(2.9)
...
incubation
top-level project
2016
2017
2018
2019
Java
Python
SQL
Go SDK
Beam Summit London
Euphoria API

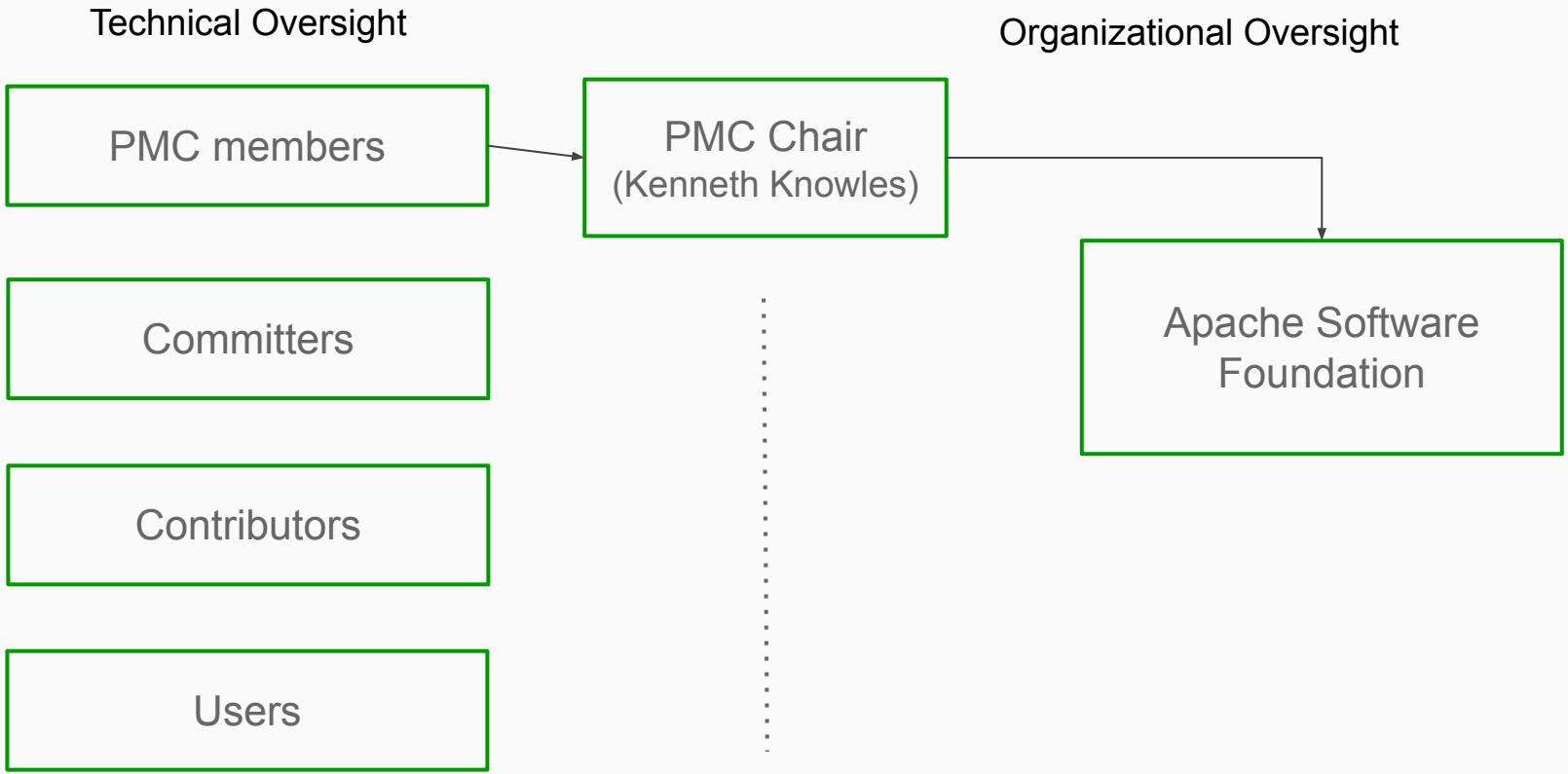
Timeline


Entered the ASF incubator
First Stable Release (2.0.0)
Most active dev@ list
More exciting stuff!
6.0.0 Release
Top Level Project
First Beam Summit
InfoWorld's Technology of Year
2nd European Beam Summit
2016
2017
2018
2019
Beam is created
0.1.0 Release!
Unbounded Pipelines
20 Connectors (I/Os)
Community Summit
ASF Top Project
First LTS
Beam @ ApacheCon

As of March 2019...



Org chart of Apache Beam



First Things First

I love the project, where do I start?

Start by subscribing to the mailing lists. Beam has different mailing lists depending on what kind of support you need.

1. For user support and questions: user-subscribe@beam.apache.org
2. To participate in development discussions: dev-subscribe@beam.apache.org

Once you confirm the subscription, you may even want to send an introductory email to the community. Let people get to know you!

Later on you might need to use other mailing lists, such as commits@ and builds@, but we will keep things short here

There is also a slack channel, but the email list is crucial!

First Things First

Have questions? Ask in the mailing lists!

Asking the right questions is very important for our project! It helps us to understand what issues our users and contributors are facing and which parts of the product are unclear. It gives us an opportunity to improve and help future contributors to get onboarded quicker.

As you browse and search, you will see the way other people write emails. Follow the leading examples.

Send your comments to the appropriate mailing list. *Well-directed conversations are more productive and easier to find later*. Choose appropriate Subject, include all the referring points.

As you will notice, some of the emails have subject that start with [VOTE], [DISCUSS], [UPDATE]: those are to let people know what is the email about before they even open it.

Structuring clear emails is a skill.

Start contributing

You found an area where you could contribute?

Remember, Beam accepts all kind of contributions, not just code!

Beam, as many other ASF projects uses Jira to track issues.

Send an email in the dev@beam.apache.org saying that you would like to be added as a contributor. Provide your Jira username. Here is an *example*

After that you will be able to create and assign tickets to yourself or other contributors

In the next slide you will find more about Jira for starters

Jira

Beam Jira Beginner's Guide

https://cwiki.apache.org/confluence/display/BEAM/Beam+Jira+Beginner%27s+Guide

Daniel Oliveira wrote this guide to help you navigate Jira. It will be your best friend to walk you through the process of joining Jira, navigating it, etc :)

- P0 **Blocker**: get paged, stop whatever you planned on doing, work late to fix
- P1 **Critical**: continually update everyone on status and shouldn't sit around unassigned
- P2 **Major**: most things here; they are important but not an interrupt - if unsure, use Major and someone from the community will review the details
- P3 **Minor**: nice-to-have things, may not make it onto regular planning and roadmapping but still useful if someone wants to do it
- P4 **Trivial**: a special category for typos or minor cleanups that might be good for newcomers just to get started with the project, build system, etc

Jira

Beam Jira Beginner's Guide

https://cwiki.apache.org/confluence/display/BEAM/Beam+Jira+Beginner%27s+Guide

Some other highlights on tags:

- **starter** - means that it is a good task to get started contributing to Beam.
- **triaged** - means that an experienced member of the community has reviewed the component, priority, issue type, and that someone who understands the issue has been contacted. ***Do not add this label to new issues or no one will look at it.***
- **sickbay** - means that the Jira is about a test that was disabled because it was broken, and it should be fixed and re-enabled
- **flake** - means the Jira is about a test that is flaky so it harms everyone's productivity, a non-sickbayed flake must always be Critical priority or higher

Beam Components

By learning about Beam components, you will have a better idea where your skills may benefit the project the most

Beam is a complex project, with many moving parts. Many of them have subtle connections, and some amount of code debt (like any project).

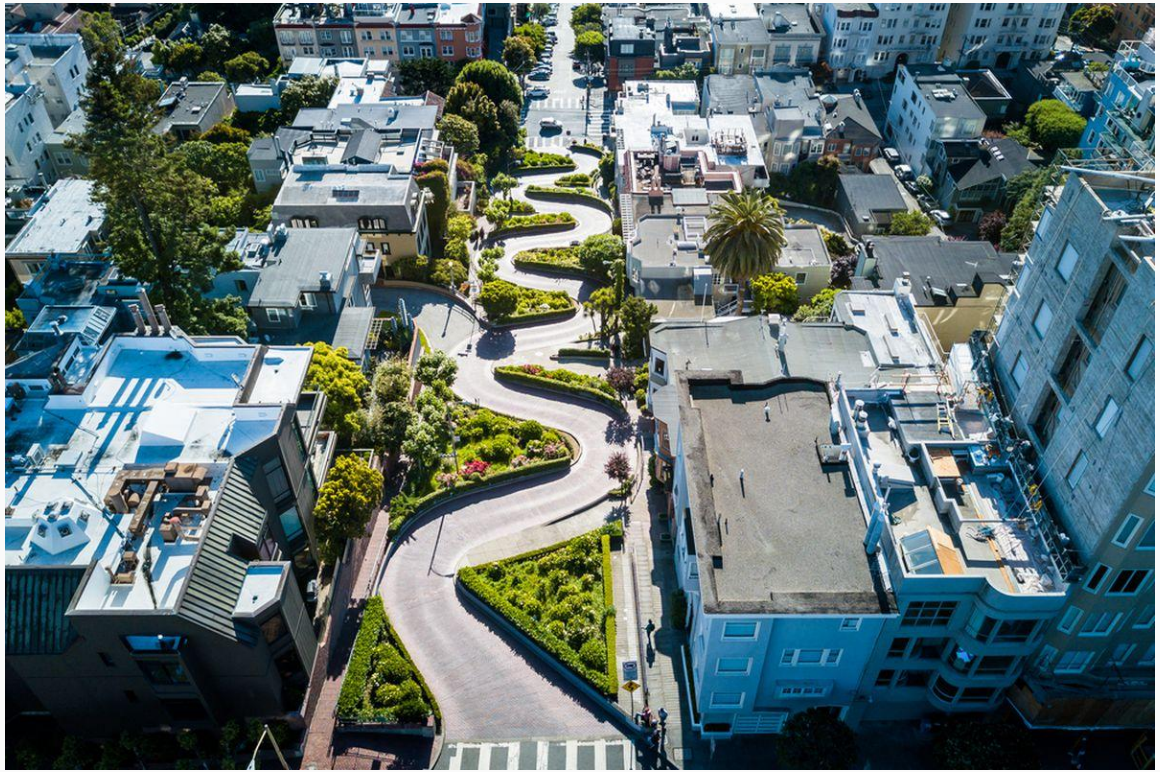
Beam's High-level Components

- **Portability API**. Beam was created with a vision to be portable across runners and languages. The Portability API is a work-in-progress to enable this. It's a set of APIs (duh) to enable communication between a runner, and an SDK; and a set of utilities written in Java for runners to support it.
- **Java SDK**. The Java SDK is the most complete, and has support for the most features in the most runners [capability matrix].
    - Java is by far the largest language in the Beam codebase, and it's the base for the portability API.
- **Python SDK**. The Python SDK was the second language that Beam supported, and has partial support on the Portability and Legacy stacks. Check `sdks/python` to learn more.
- **Go SDK** is the most recent addition. This SDK was written from the beginning to work on the Portability API, but its support is limited. Check out `sdks/go` to learn more.
- **Beam SQL**. A new exciting development was the addition of a series of SQL transforms for Batch and Streaming pipelines. The code lives in `sdks/java/extensions/sql` [documentation].
- **Many more.** These are the more active parts of Beam, but not the only ones. Other interesting sections are the runners, the build system in Gradle/Groovy, the website, etc.

Beam Roadmap

We want to build new things too!

See: https://beam.apache.org/roadmap



Not always the straightest path

More ways to contribute

Bug reports - we take them seriously!

To help us to quickly fix the bug, be sure to include as much information with your report as possible such as your platform, version numbers, error logs, configuration, etc.

To submit a bug report, first make sure the bug hasn't been reported before, fixed in a newer version of the software, or fixed in the current development version. Then file a report.

Here is Beam's Jira for tracking issues.

If you have the knowledge to supply a patch that fixes the issue, please do so!

Code reviews

Apache Beam follows Review-then-Commit (RTC) policy for code changes.

For code reviews, ***either*** the reviewer ***or*** the author must be a committer.

We think that reviews are not just a way to make code better or spread knowledge, they are a *place where community is built*. Letting non-committers review code by committers gives them another opening to the Beam community.

Read this blog written by the Apache Beam PMC chair on their approach to community building.

Merging code

Always get to LGTM in Committer Guide

Once a contributor approves a Pull Request, a committer is able to go ahead and merge it.

- It is good practice to squash long sets of commits into fewer meaningful commits.
- If there is a JIRA issue to track your change, please tag the commits with it
- Committers should try to merge changes with a Merge commit. This allows to jump to the Pull Request when reviewing the history.

You are also welcome to check out this Code Velocity Dashboard with information about Pull Requests and reviewers.

Dealing with test failures

Beam has a few basic principles to address breakages in critical tests (pre and post-commit). These are:

- Rollback first
- A failing or flaky test is a critical issue (P1)
- Flaky tests must be fixed or removed
- Post-commit fixes should come with pre-commit tests
- It’s good to be explicit about downstream project failures

Code changes summary

1

Submit the PR

Any contributor can submit PR. It is good to tag them with a JIRA number, if there is one.

2

Review the PR

In Beam, the convention is get a reviewer by mentioning them with @username.

3

Look for LGTM

- Reviewers may ask you to make changes to your PR. If it's large, the may even ask you to provide a design document first.
- Iterate on the code until you get LGTM.

4

Merge

- Once a reviewer approves your PR, a committer will be able to merge your change.
- It is good policy to squash your changes.

After a pull request is merged, it's good practice to check if the JIRA issue should be resolved.

Beam Testing

https://cwiki.apache.org/confluence/display/BEAM/Contribution+Testing+Guide

Beam is a large project, with many code bases. Testing is organized in pre-commit tests that run on every Pull Request, and post-commit tests that run continuously on a schedule.

- The main tool to Build and Test Beam is Gradle
- For non-JVM languages, Gradle calls language-specific utilities

For a deep dive into Beam testing, check out this wiki page.

Mikhail Gryzykhin made a great dashboard to monitor post-commit test status. These should be useful to check the state of the code, and whether your changes are to blame for a test failure in a PR.

Technical/Design Docs

https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=95653903

Apache and Beam work by encouraging healthy discussion related to new features, and product vision.

When proposing significant changes / new features, it is important to have a design review.

- In Beam, the convention is:
  - For small designs, it is enough to have a discussion on the dev@ mailing list,
  - For larger designs, a detailed design document in Google docs can be shared to dev@, along with a short paragraph describing the proposal [example].

Design docs are important documentation artifacts. They help the community get a sense of your proposal; and will help users and contributors in the future as they try to understand the project better.

- To find an index of existing design documents, please check the wiki page and Beam website.
  - Also, please try and add any documents that you write :)

Code of Conduct

A reminder that once you become Apache Beam community member, you must adhere to the code!

If you use open source software in your company or as a part of your job, remember that dev@ user@ mailing lists are public, and you must not refer to internal tests, documentation in public forums.

If you contribute to Apache projects, your contribution earn merit to you as an individual. You represent yourself, not the company you work for.

For more info..

- Apache Beam website
- How to get involved with Apache
- Apache Code of Conduct
- Why Licensing matters
- The Apache Way: https://s.apache.org/apache-way-for-everyone

Thank you

This slide deck was originally developed by Aizhamal Nurmamat kyzy <aizhamal@apache.org>

Contributors: Pablo Estrada, Austin Bennett

Q & A

- Other suggestions or questions?
- As suggested in prior slides, use dev@ and user@ !

Doing: Steps

- Subscribe to list(s)
- Signup For Jira
- GitHub