
KubeCon
CloudNativeCon
North America 2024

KubeCon | CloudNativeCon

North America 2024

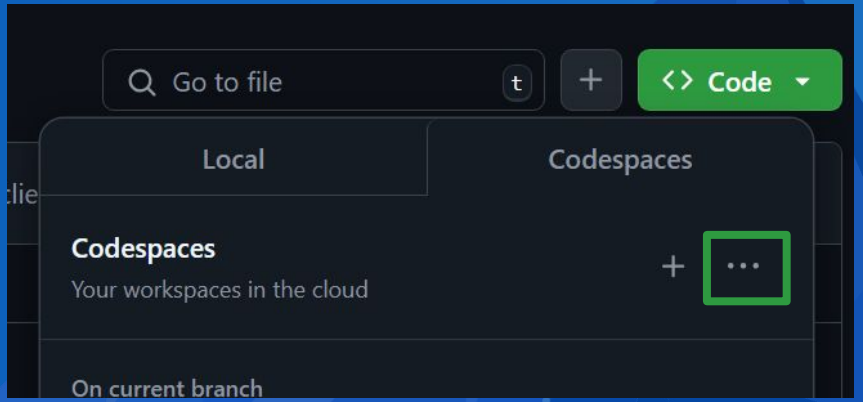
Sidecar-less Service Mesh: Let's Work Together on Istio

Lin Sun & Mitch Connors

Before we get started...

github.com/istio/istio

```
make build
```



Agenda

- Service Mesh and Istio Overview
- Istio Infrastructure Overview and Workshop
- Istio Control Plane Overview and Workshop
- Istio.io Overview and Workshop

What is a Service Mesh?

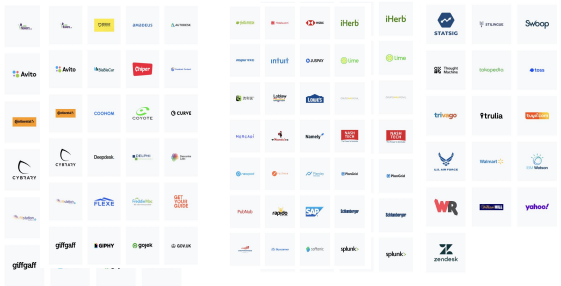
A service mesh is a **programmable** framework that allows you to **observe, secure and connect** microservices.

Istio - The Leading Service Mesh

Istio Launched — 2017

envoy Data Plane Enhancements — 2019-20

1000s Production Customers — 2019-2022



CNCF — 2022

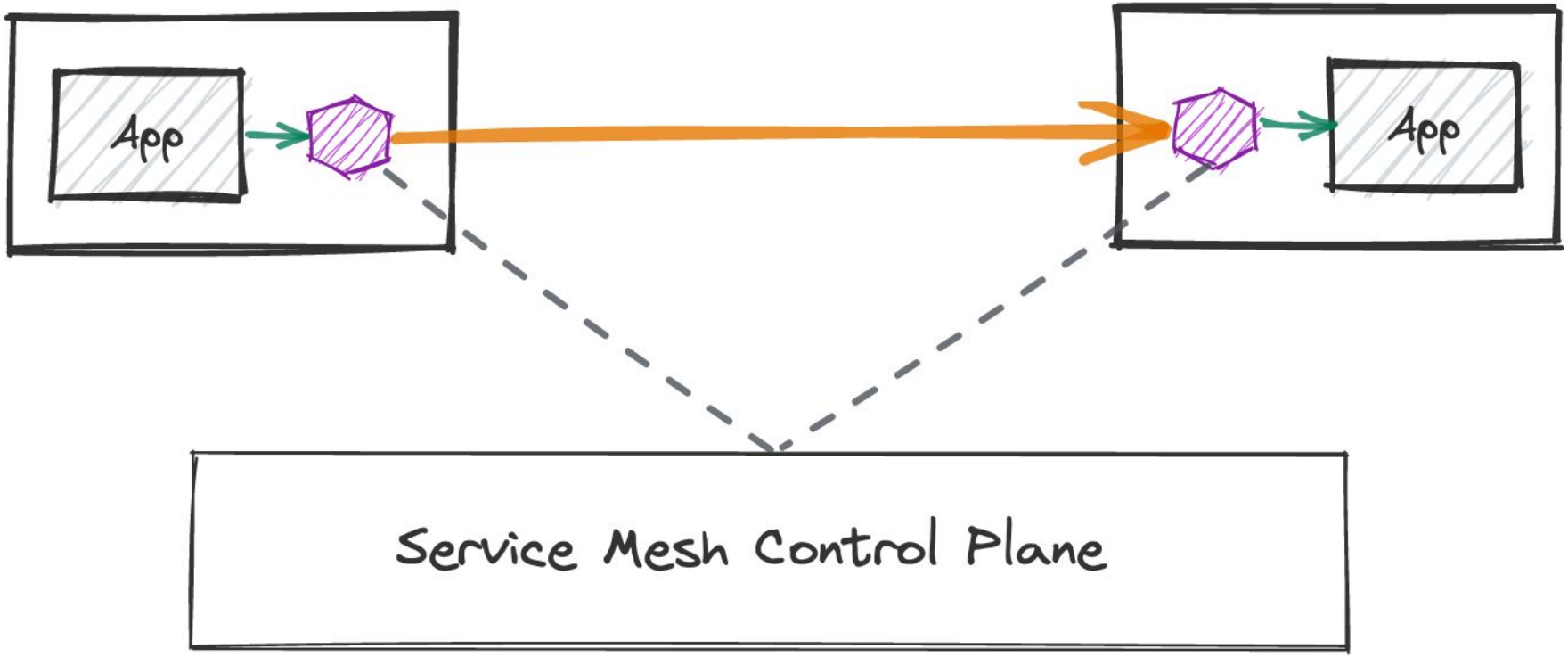
ambient mesh Ambient Mesh Launched — 2022

Ambient Mesh -> GA — 2024

10+ New Community Releases

~ 1000 Community Contributors

Service Mesh Architecture



Challenges With Sidecars

- Operation complexity & transparency
- Incremental adoption
- Overprovision resources

Introducing Ambient Mesh

A new dataplane mode for Istio without sidecars.

Sep 7, 2022 | By John Howard Levine - Solo.io, Justin Pettit

Fast, Secure, and Simple: Istio's Ambient Mode Reaches General Availability in v1.24

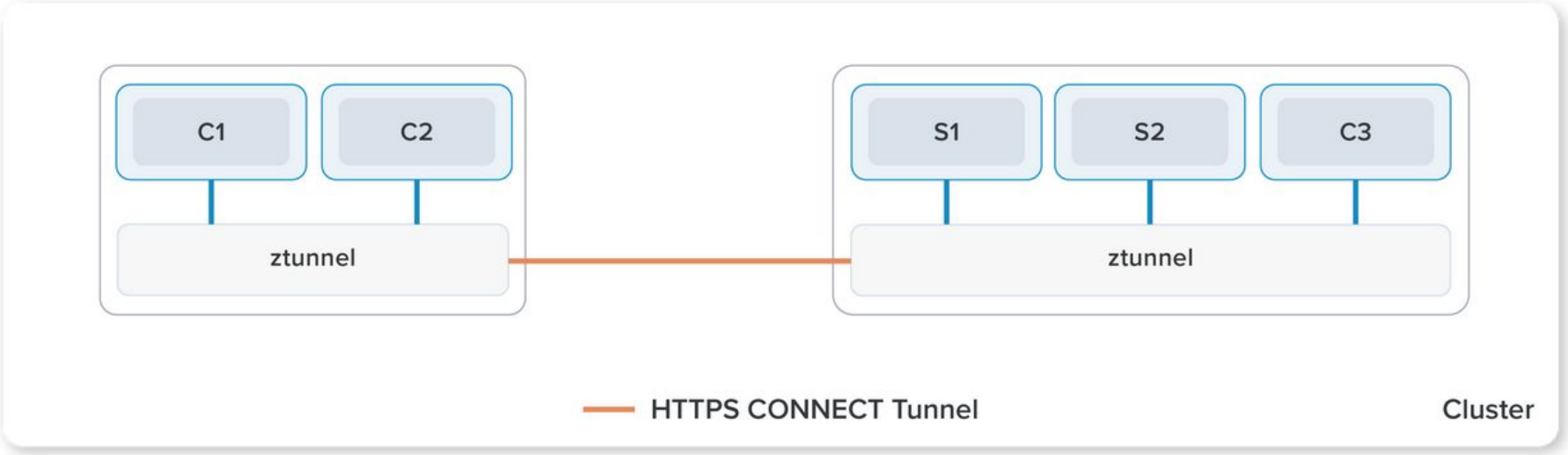
Our latest release signals ambient mode – service mesh without sidecars – is ready for everyone.

Nov 7, 2024 | By Lin Sun - Solo.io, for the Istio Steering and Technical Oversight Committees

We are proud to announce that Istio's ambient data plane mode has reached General Availability, with the ztunnel, waypoints and APIs being marked as Stable by the Istio TOC. This marks the final stage in Istio's feature phase progression, signaling that ambient mode is fully ready for broad production usage.

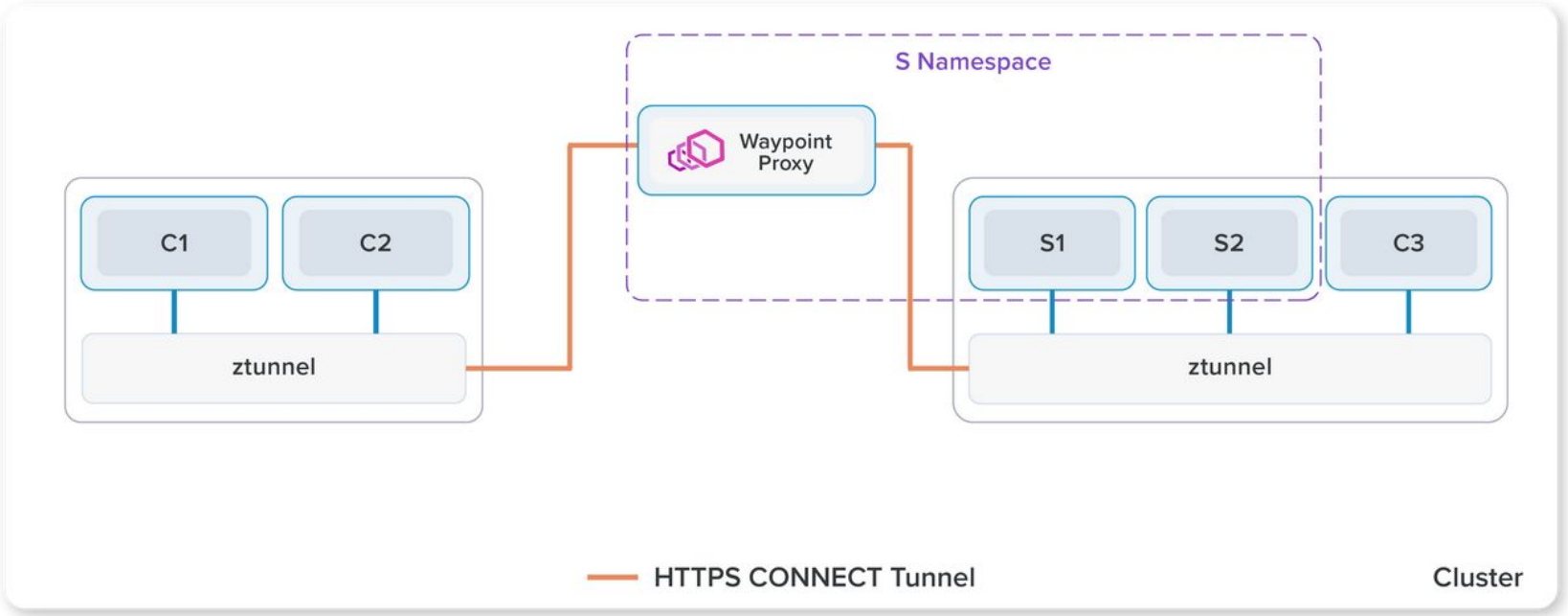
Ambient mesh — and its reference implementation with Istio's ambient mode — was announced in September 2022. Since then, our community has put in 26 months of hard work and collaboration, with contributions from Solo.io, Google, Microsoft, Intel, Aviatrix, Huawei, IBM, Red Hat, and many others. Stable status in 1.24 indicates the features of ambient mode are now fully ready for broad production workloads. This is a huge milestone for Istio, bringing Istio to production readiness without sidecars, and offering users a choice.

Secure Overlay Layer



Ambient mesh uses a shared, per-node ztunnel to provide a zero-trust secure overlay

Layer 7 Processing Layer

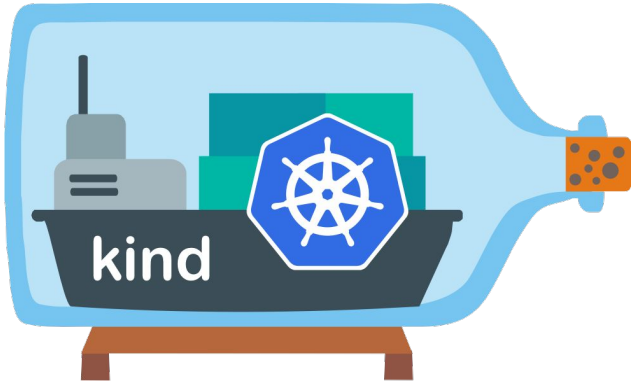


Development Workflow

Toolchain

GitHub Codespaces
github.com/istio/istio

Ready-to-use development environment, no need to download dependencies…



Run Kubernetes locally (in Docker)

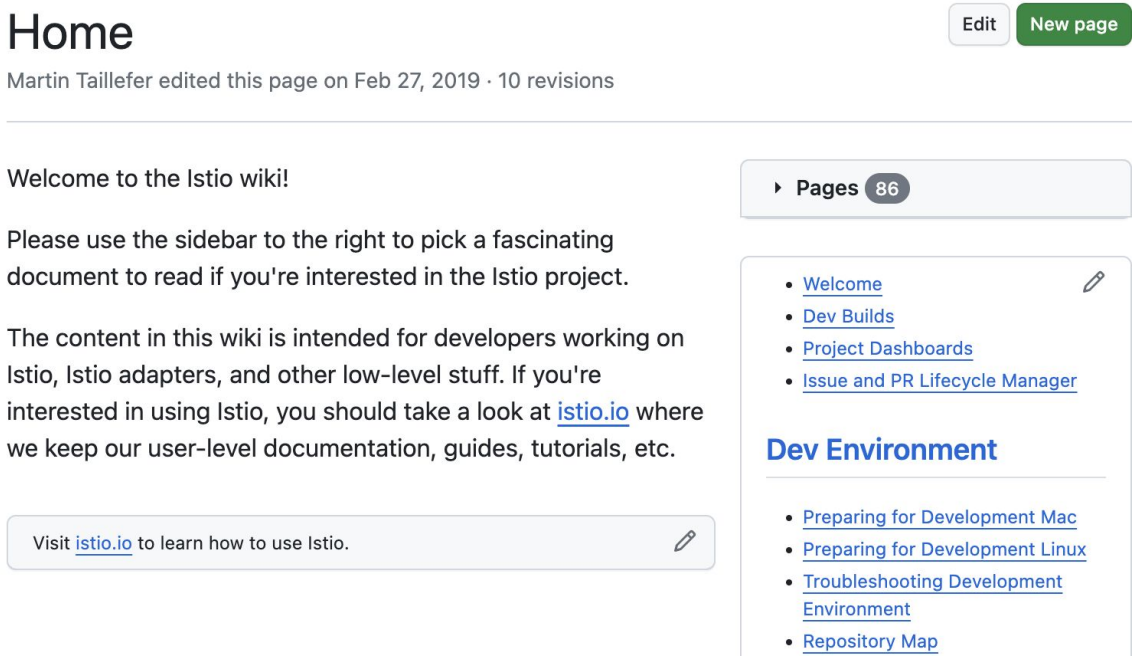
Getting Help

github.com/istio/istio/wiki

slack.istio.io

#contributors

#ambient-dev



Building Istio - Local builds


```
# This defines the docker hub to use when running integration tests and building docker images
# eg: HUB="docker.io/istio", HUB="gcr.io/istio-testing"
export HUB="docker.io/$USER"

# This defines the docker tag to use when running integration tests and
# building docker images to be your user id. You may also set this variable
# this to any other legitimate docker tag.
export TAG=$USER
```

- To build docker images: ***make docker***
  - Can do specific images: ***make docker.proxyv2***
- To push docker images: ***make docker.push //required for integ tests***
  - ***docker login -u $USER***
- Starting Istio:
  - Create a Kubernetes cluster using kind
  - ***istioctl install --set profile=demo --set hub=$HUB --set tag=$TAG***
    - *Check for the istioctl location, for example:./out/linux_amd64/istioctl*

Running the Tests

- Unit Tests
  - ***make test***
  - ***go test ./pilot/pkg/networking/core/ -v***
  - vscode Go Test Integration

- Integration tests:
  - ***go test -tags=integ ./tests/integration/pilot/cni/...***
    - Runs in cluster from kubeconfig
  - ***./prow/integ-suite-kind.sh test.integration.pilot.kube***
    - Sets up kind cluster, runs there
    - ***export INTEGRATION_TEST_FLAGS="-run TestTraffic"***
    - (Optional) Add `--skip-cleanup` to inspect the cluster after tests run (or fail!)

Writing Integration Tests

Framework abstracts:

- Installing Istio
- Deploying test apps
- Multi-cluster concerns
- Sending traffic between pods
- Applying and cleaning up config

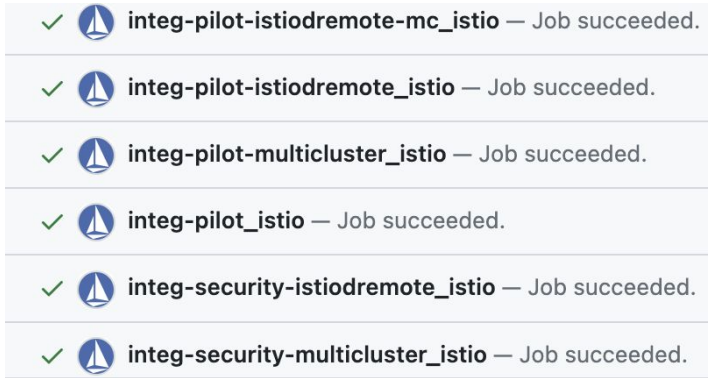
```
func TestMain(m *testing.M) {  Steven Landow +2
    framework.
func TestReachable(t *testing.T) {  Steven Landow *
    framework.NewTest(t).Run(func(t framework.TestContext) {

        t.NewSubTestf( format: "with dr").Run(func(t framework.TestContext) {
            t.ConfigIstio().YAML( ns: "ns",  yamlText...: `...`)
            echotest.New(t, apps).
                Run(func(t framework.TestContext, from echo.Instance, to echo.Target) {
                    from.CallOrFail(t, echo.CallOptions{To: to})
                })
        })

        // more tests with different config
    })
}
                    }).
                    Build()
                return err
            })
}
```

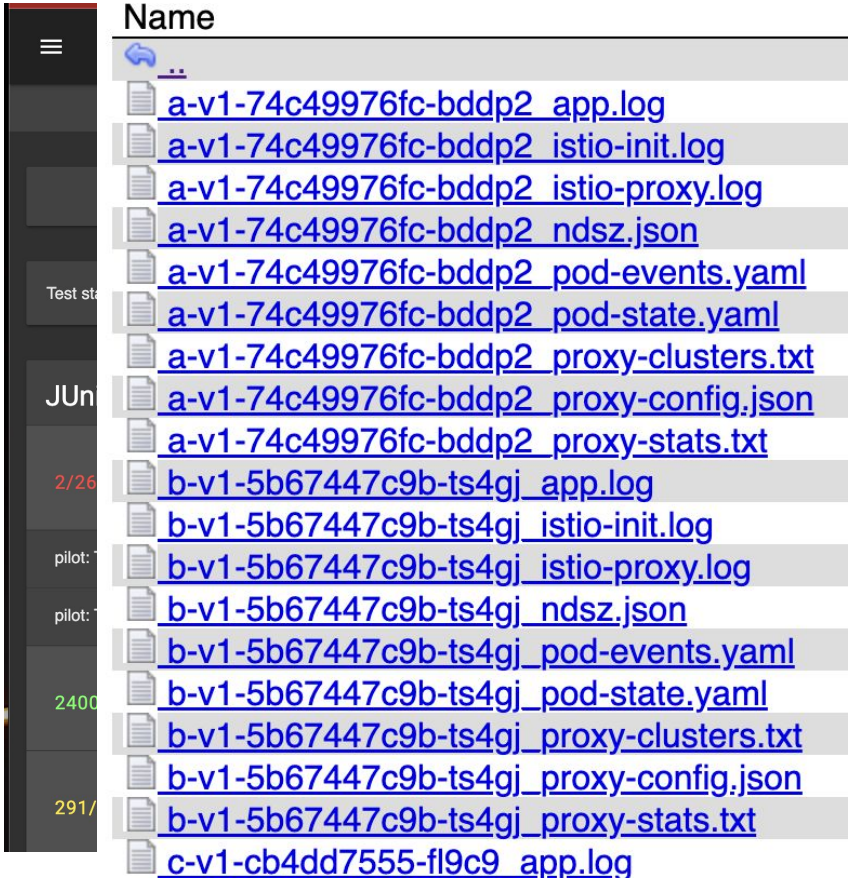
Submitting a PR

- Make changes in your fork, not in origin
  - Fork istio at github.com/istio/istio
  - **`git remote add personal https://github.com/yourhandle/istio`**
- Create a local branch
  - **`git checkout -b my-branch`**
- Keeping your fork in sync
  - **`git fetch origin master`**
  - **`git rebase origin/master`**
  - **`git push -f personal`**
- Committing changes to your fork
  - When you're happy with the changes, you can commit them to your repo:
    - **`git add .`**
    - **`git commit –a`** ***OR*** **`git commit –amend`** (This will allow you to add/edit a commit message for your commit)
    - **`git push –f personal <local branch name>`** (e.g. git push –f origin my-branch)
- Create the PR via the web from your branch

Test Automation - CI Tools

- In CI we use Prow
  - https://prow.istio.io/
- Test run logs are available here
- Artifacts link has valuable dumps of:
  - Cluster state
  - Pod logs
  - Proxy config





Project Components

Components

Control Plane:

- XDS Server
- Validating Webhook
- Mutating Webhook (Injection)
- Ingress Controller
- Gateway Controller

Data Plane:

- Istio Agent
- Envoy Proxy
- iptables
- zTunnel

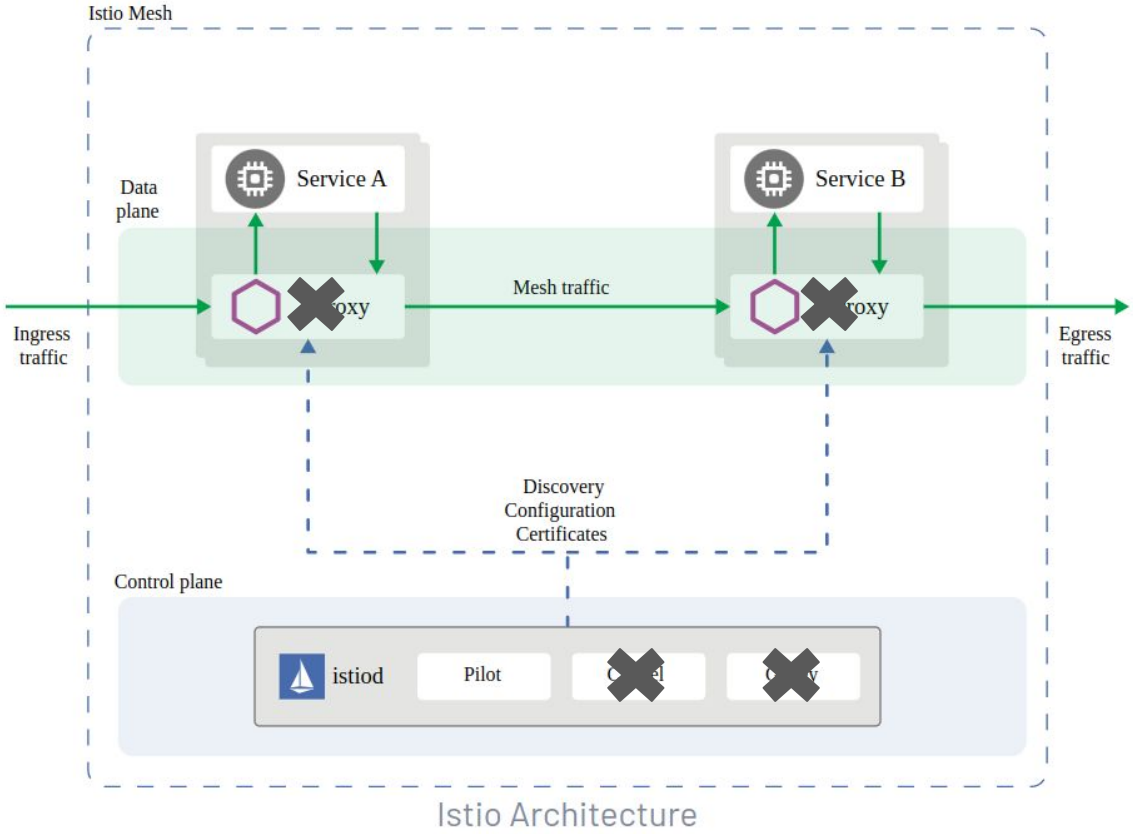
CNI:

- CNI Daemon

CLI:

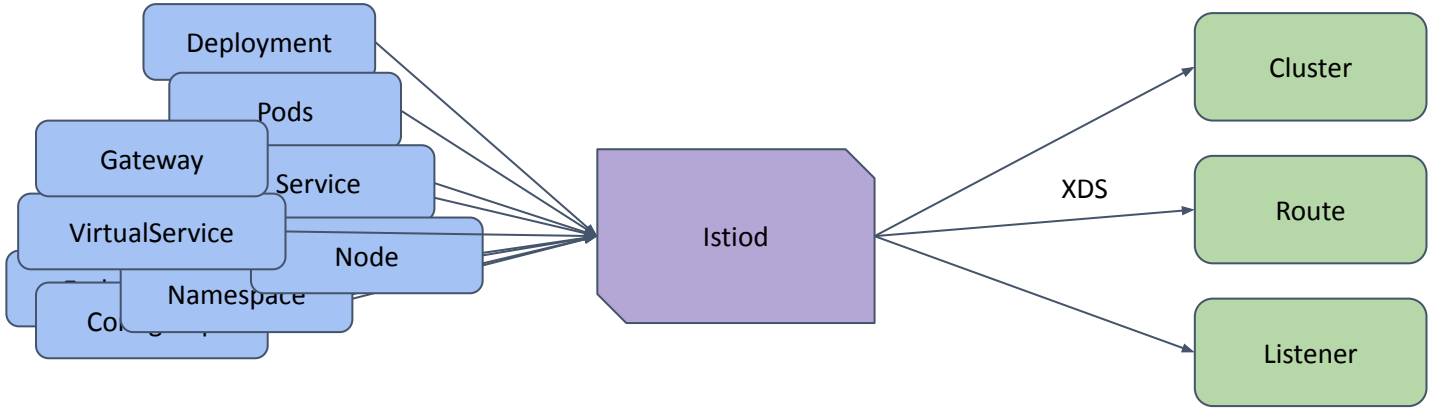
- istioctl

Control Plane Internals - pilot



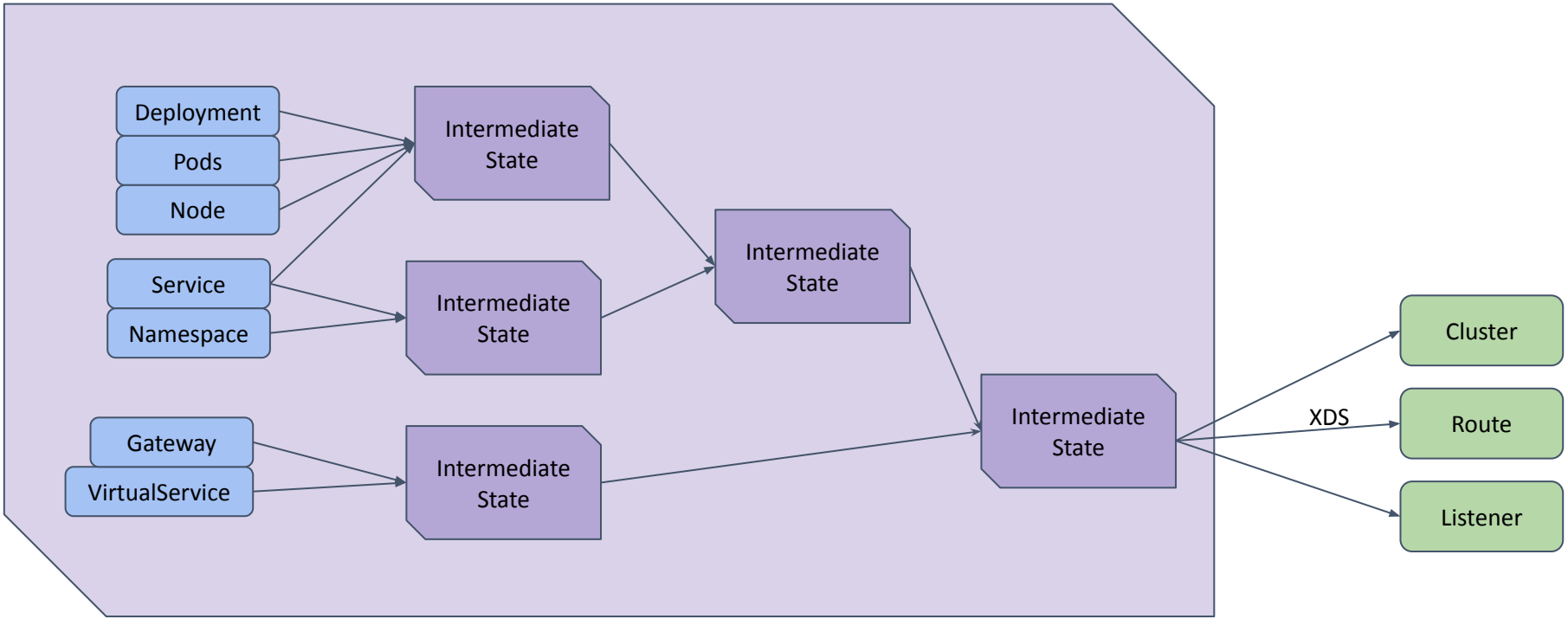
Istio Architecture

Control Plane Internals - pilot



1. Read a bunch of objects
2. Translate them into envoy (or zTunnel!) config

Control Plane Internals - pilot



... with a bunch of intermediate layers and machinery

istio.io website

Documentation

Learn how to deploy, use, and operate Istio.

Concepts
Learn about the different parts of the Istio system and the abstractions it uses.

Setup
Instructions for installing the Istio control plane on Kubernetes.

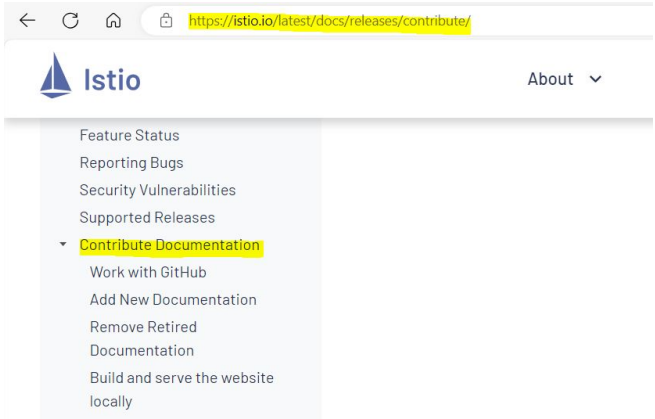
Tasks
How to do single specific targeted activities with the Istio system.

Examples
A variety of fully working example uses for Istio that you can experiment with.

Operations
Concepts, tools, and techniques to deploy and manage an Istio mesh.

Releases
Information relating to Istio releases.

Reference
Detailed authoritative reference material such as command-line options, configuration options, and API calling parameters.



In addition to the above documentation links, please consider the following resources:

- Frequently Asked Questions
- Glossary
- Documentation Archive, which contains snapshots of the documentation for prior releases.

Where do I look for easy issues to fix?

- Look for issues in istio.io repo with the following labels:
  - community/help wanted,
  - community/good first issue,
  - community/intern-help-wanted
- While reviewing the docs on istio.io, if you come across any bugs, you can create a PR (pull request) to fix it. This is the easiest way to get started.
- You can also contribute to the automated tests suite for docs on istio.io. You can find more information about this here.

Contribute to Documentation

Work with GitHub
Shows you how to use GitHub to contribute to the Istio documentation.

Add New Documentation
Details how to contribute new documentation to Istio.

Remove Retired Documentation
Details how to contribute retired documentation to Istio.

Build and serve the website locally
Explains how to locally build, test, serve, and preview the website.

Front matter
Explains the front matter used in our documentation and the fields available.

Documentation Review Process
Shows you how changes to the Istio documentation and website are reviewed and approved.

Add Code Blocks
Explains how to include code in your documentation.

Use Shortcodes
Explains the shortcodes available and how to use them.

Follow Formatting Standards
Explains the standard markup used to format Istio documentation.

Style Guide
Explains the style conventions used in the Istio documentation.

Terminology Standards
Explains the terminology standards used in the Istio documentation.

Diagram Creation Guidelines
Provides assets and instructions to create diagrams for the Istio documentation.

Refer: [Istio / Contribute Documentation]

Add new documentation

- Identify the audience and intended use for the information.
- Choose the type of content you wish to contribute.
- Choose a title.
- Write your contribution following our documentation contribution guides.
- Submit your contribution to our GitHub repository.
  - Fork the Istio documentation repository. `git clone https://github.com/istio/istio.io.git `
  - Create a branch for your changes. `git checkout -b first-contribution`
  - Add commits to that branch.
  - Open a PR to share your contribution.
- Alternatively, anyone with a GitHub account who signs the CLA can contribute a quick edit to any page on the Istio website directly when you access `preliminary.istio.io`. Refer Istio / Work with GitHub
- Follow our review process until your contribution is merged.

Build and serve the website locally

To guarantee the tests you run locally use the same versions as the tests running on the Istio Continuous Integration (CI), we provide a Docker image with all the tools needed, including our site generator: Hugo.

- Obtain a shell using the pre-built docker image mentioned above
  - go to the root of your fork of istio/istio.io and run `make shell`
- Preview your changes
  - To preview your changes to the site, go to the root of your fork of istio/istio.io and run `make serve`
- Test your changes
  - HTML proofing: ensures all links are valid along with other checks.
  - Spell check: ensures content is spelled correctly.
  - Markdown Style check: ensures the markup used complies with our Markdown style rules.
  - To test your changes to the site, go to the root of your fork of istio/istio.io and run `make lint`

Diagram Creation Guidelines

To create your diagrams, follow these steps:

1. Refer to the guide and copy-paste from it as needed.
2. Connect the shapes with the appropriate style of line.
3. Label the shapes and lines with descriptive yet short text.
4. Add a legend for any labels that apply multiple times.
5. Contribute your diagram to our documentation.

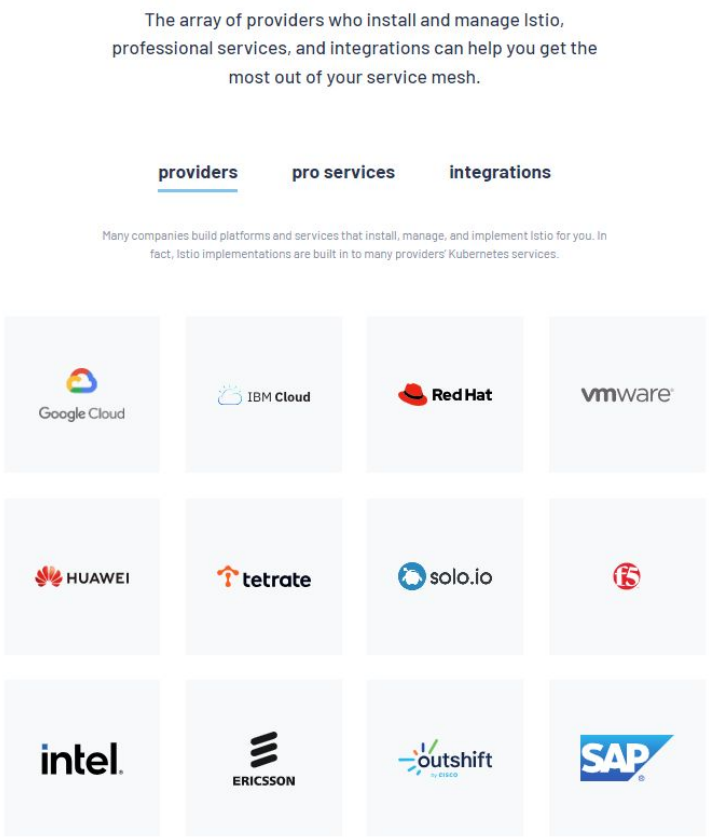
If you create the diagram in Google Drawings, follow these steps:

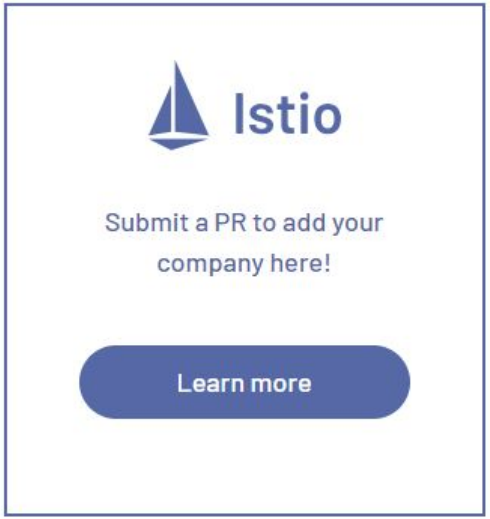
1. Put your diagram in our shared drive.
2. When the diagram is complete, export it as SVG and include the SVG file in your PR.
3. Leave a comment in the Markdown file containing the diagram with the URL to the Google Drawings file.

Testing document content

- Doc tests confirm that the example, task, and other documents, which contain instructions in the form of bash commands and expected output, are working as documented.
- Refer [istio.io/tests/README.md at master · istio/istio.io (github.com)]
- To write an istio.io test, follow these steps:
  - In the metadata at the top of the index.md file to be tested, change the field `test: no` to `test: yes`
  - Run **`make snips`** to generate the bash script. After the command completes, you should see a new file, snips.sh, next to the index.md file that you modified in the previous step.
  - Run **`make lint-fast`** to check for script errors.
  - Create a test bash script named test.sh next to the snips.sh you have just generated.
  - Your bash script will consist of a series of test steps that call the commands in your generated snips.sh file.
  - Include test setup and cleanup, and include required snips and verify functions as required
- Run the doc test using **`make doc.test`** or its variants as required

Istio Ecosystem

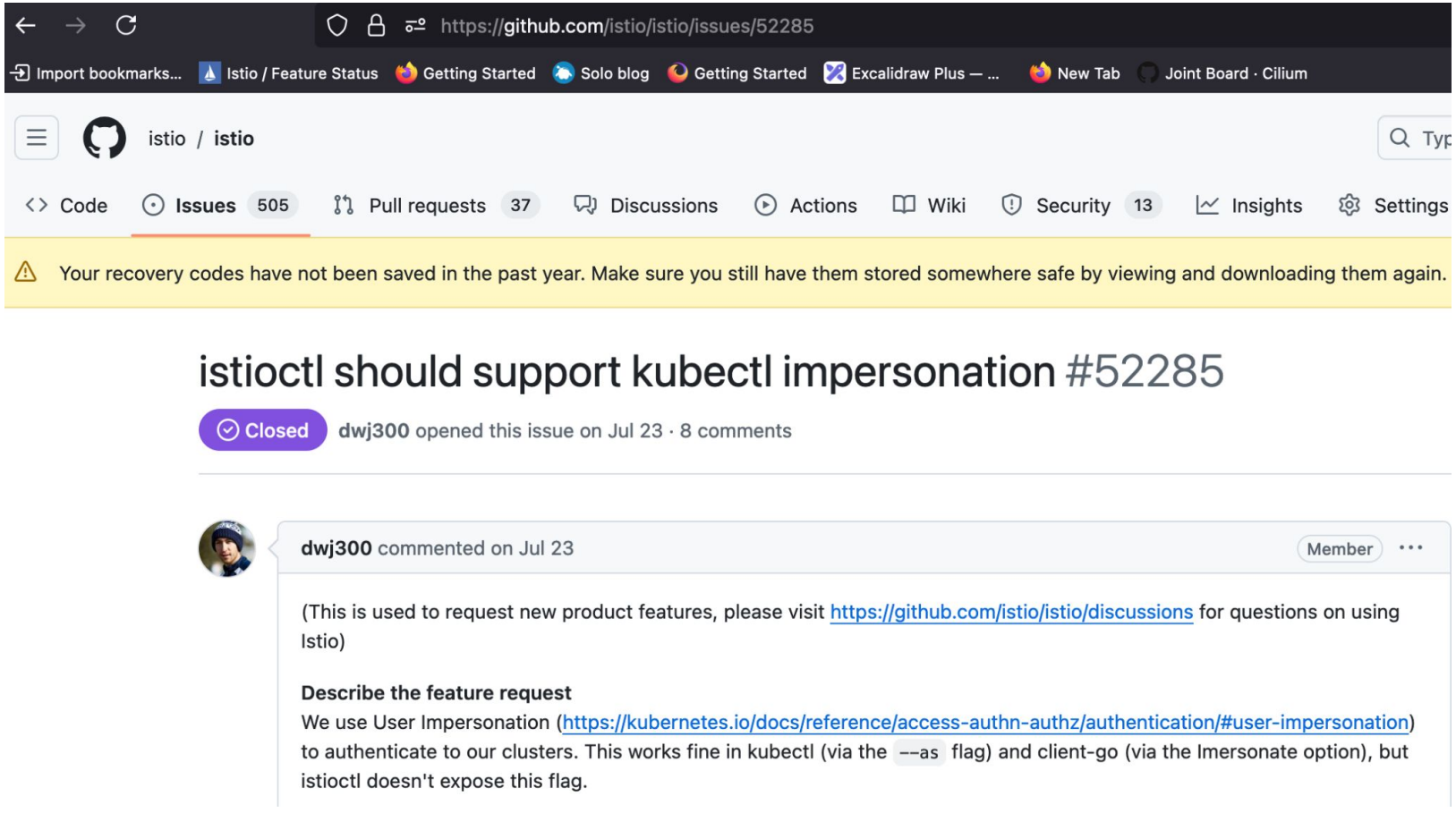




Add your company logo at : https://github.com/istio/istio.io/blob/master/data/companies.yml

Your company will be listed at Istio / Ecosystem

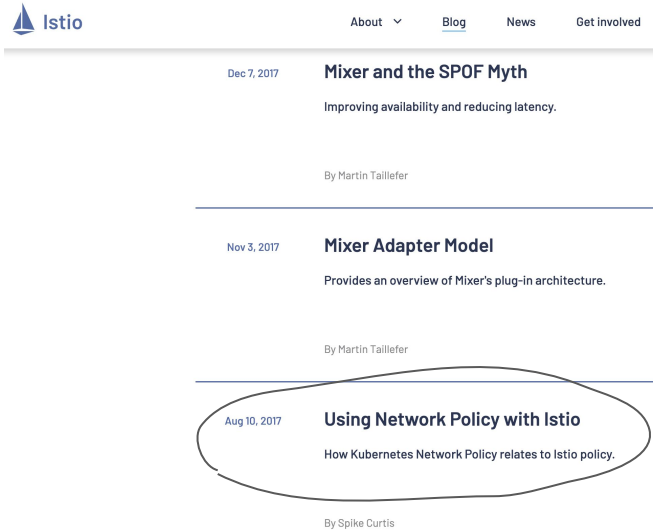
A small task to get started!



A small task to get started!

- *CHECK_EXTERNAL_LINKS=true make lint*
  - *[404] https://envoyproxy.github.io/envoy/*
  - *[0] https://gateway-api.org/*
  - *[404] https://www.tetrate.io/blog/istio-has-applied-to-join-the-cncf/*
  - *[404] https://kubernetes-sigs.github.io/service-apis/*
  - *[404] https://www.kiali.io/documentation/getting-started*
- Perform quick edit or push a PR the normal way

https://preliminary.istio.io/latest/blog/2017/0.1-using-network-policy/



Wrap Up

Get more information

- Working Group Weekly Meetings: https://github.com/istio/community/blob/master/WORKING-GROUPS.md
- Community, Code of Conduct and Processes: https://github.com/istio/community
- Wiki pages: https://github.com/istio/istio/wiki
  - Dev Environment, Writing Code, Test, etc.
- Istio Slack: https://istio.slack.com/
- Istio Google Drive (Design docs, etc): https://drive.google.com/drive/folders/0ADmbrU7ueGOUUk9PVA

Community and Project Structure

- Roles
  - Collaborator - Casual contributor to the project.
  - Member - Regular active contributor having merged at least 1 PR.
  - Maintainer - Experienced maintainer that can approve contributions from other members.
  - Workgroup Lead - Approves proposals and sets priority for their functional area.
  - Release Manager - Shepherd a relapse to GA. Appointed by ToC.
- Committees
  - Technical Oversight Committee - Responsible for cross-cutting product and design decisions.
  - Steering Committee - Oversees administrative aspects of the project, allowing the ToC to maintain a technical focus.

Grow your reach

- Ask - Engage with other Istio users/members on discuss and Istio slack
- Join - Join the Istio community and help contribute
- Engage - Regularly attend and participate in WG meetings, be responsive to requests, help review PRs
- Contribute - Help contribute to Istio by fixing/triaging issues, enhancing docs and implementing new features

Reference: https://istio.io/latest/get-involved/