



# An event based approach for CI/CD pipelines

What challenges are there in the communication between different tools in CI/CD ecosystems and how can they be mitigated?



Emelie Pettersson

Fredrik Fristedt





# Agenda

- What is a pipeline?
- What communication challenges exists between CI/CD tools?
- How has Ericsson tackled the challenges?
- How Axis started using events in CI/CD
- Q&A



# What is a pipeline?



# What is a pipeline?

”Kärt barn har många namn”

```
stages:
```

- build
- test
- publish

```
job 1:
```

```
  stage: build  
  script: build artifacts
```

```
job 2:
```

```
  stage: test  
  script: run tests
```

```
job 3:
```

```
  stage: test  
  script: run more tests
```

```
job 4:
```

```
  stage: publish  
  script: publish artifact somewhere
```

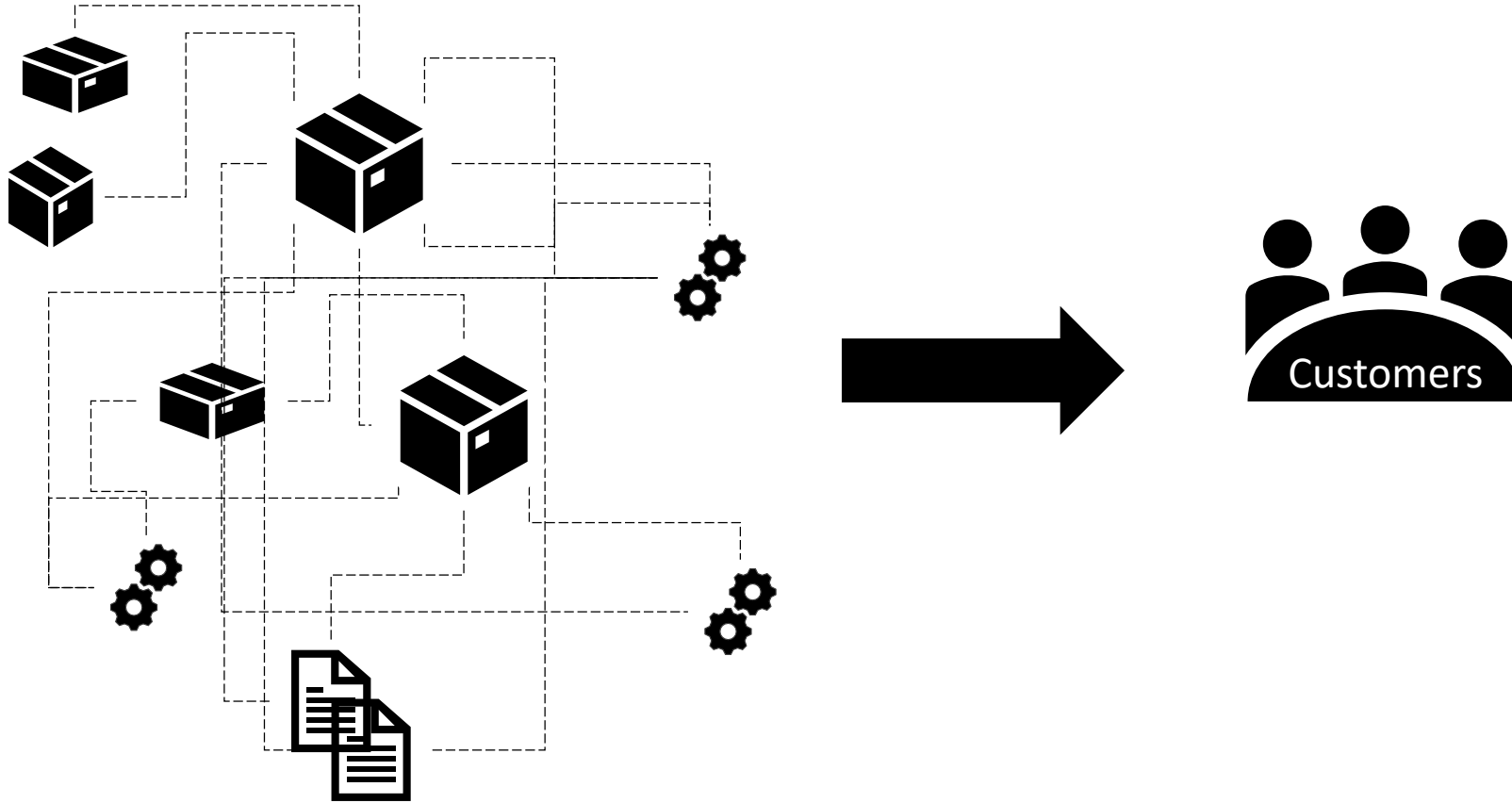
Reference: <https://docs.gitlab.com/ee/ci/yaml/>



What communication  
challenges exists  
between CI/CD tools?

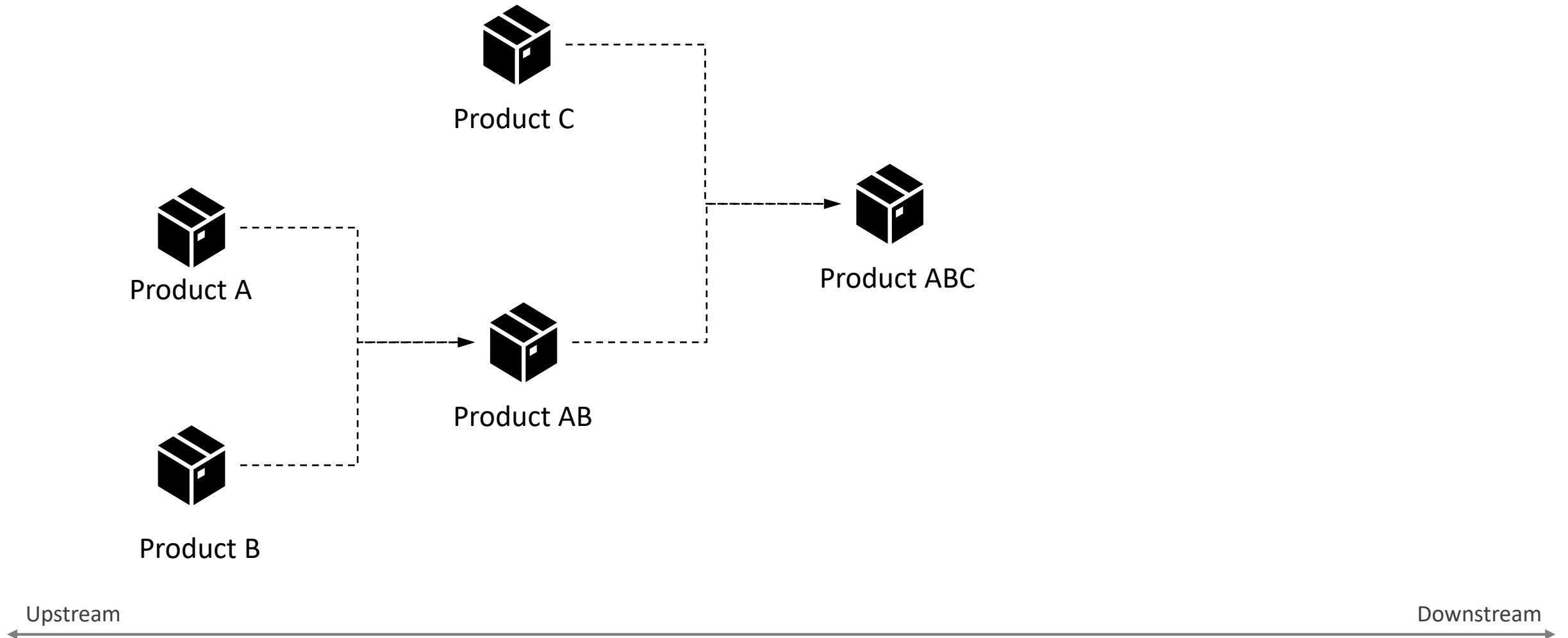


# Why does Ericsson care about CI/CD?

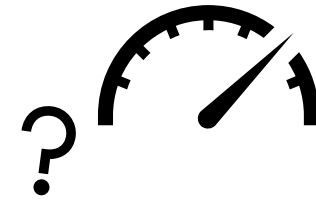
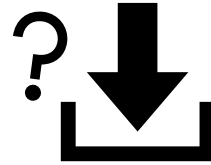
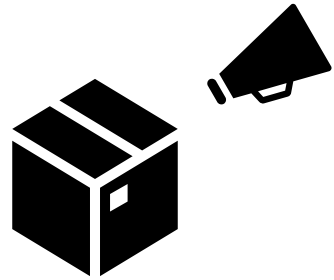




# What are connected pipelines?



# What communication challenges has Ericsson seen?



Upstream

Downstream





How has Ericsson  
tackled the  
challenges?



How has Ericsson tackled the challenges?





# How can we communicate about new artifacts available for others to integrate?



Upstream

Downstream

# How can we visualize pipelines from changes in source code to customer deployment?



## Eiffel events:

- 1 ArtifactCreatedEvent
- 2 TestCaseTriggeredEvent
- 3 TestCaseStartedEvent
- 4 TestCaseFinishedEvent
- 5 ArtifactPublishedEvent

```
stages:
```

- build
- test
- publish

```
job 1:
```

```
  stage: build  
  script: build artifacts
```

```
job 2:
```

```
  stage: test  
  script: run tests
```

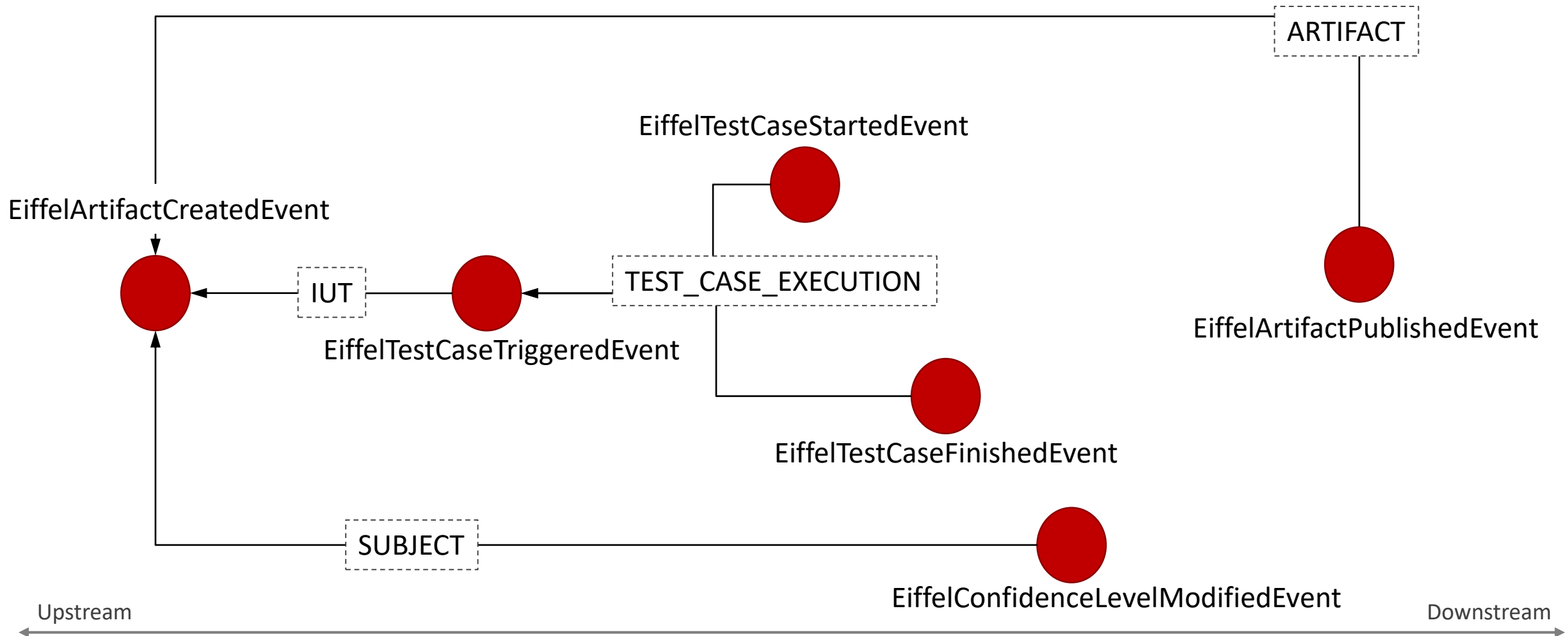
```
job 3:
```

```
  stage: test  
  script: run more tests
```

```
job 4:
```

```
  stage: publish  
  script: publish artifact somewhere
```

# How can we visualize pipelines from changes in source code to customer deployment?





# Why Eiffel?

Eiffel provides one language across different platforms.

Eiffel events contain links, enabling traceability.





# How Axis started using events in CI/CD



# Let's establish a time-line









First recorded  
presence of the  
Eiffel protocol  
at Axis

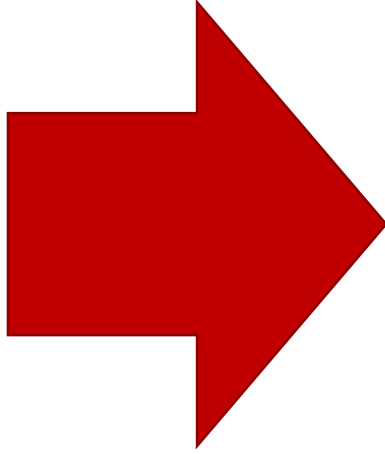
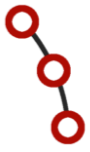
- First heard of it externally 2016
- Development of first service in 2017
- Driven through Research
- Mostly engineering curiosity and enthusiasm



Axis releases  
eiffel-  
broadcaster-  
plugin for  
jenkins on  
GitHub

- Eiffel-Jenkins-Broadcaster was published on GitHub early 2019
- Had been active internally during 2018
- Was anyone listening?







Axis starts  
prototyping an  
Eiffel based  
test system

- Jenkins was talking, no one was listening
- Testing parts of CI-pipeline were getting old
- Started from scratch with Eiffel as a main requirement
- An excellent talker, but of course also a great listener



Axis releases  
eiffel-pythonlib  
and eiffel-  
graphql-api on  
GitHub

- Community contributions
- Eiffel is important for AXIS
- Axis more engaged in the Eiffel Community
- Open backlogs



Multiple  
deployed  
services  
producing and  
consuming  
Eiffel events

- Eiffel is now a thing at Axis
- Dedicated Eiffel team
- Independent teams, working on multiple services
- Fully event driven pipeline
- Even more community contributions



# Thank you!

**Emelie Pettersson**

[linkedin.com/in/emelie-pettersson-53aaa5136/](https://www.linkedin.com/in/emelie-pettersson-53aaa5136/)  
[emelie.pettersson@ericsson.com](mailto:emelie.pettersson@ericsson.com)

**Fredrik Fristedt**

[linkedin.com/in/fredrik-fristedt-1a9b611/](https://www.linkedin.com/in/fredrik-fristedt-1a9b611/)  
[fredrik.fristedt@axis.com](mailto:fredrik.fristedt@axis.com)

Learn more about Eiffel and join our community on GitHub: <https://github.com/eiffel-community>

You can ask questions in our [Google group](#) or join us on [Slack](#).

Learn more on our [YouTube channel](#).





# Questions?

