

Eiffel introduction



Kristofer Hallén
CI tools architect
Ericsson

Software development at scale



1000s

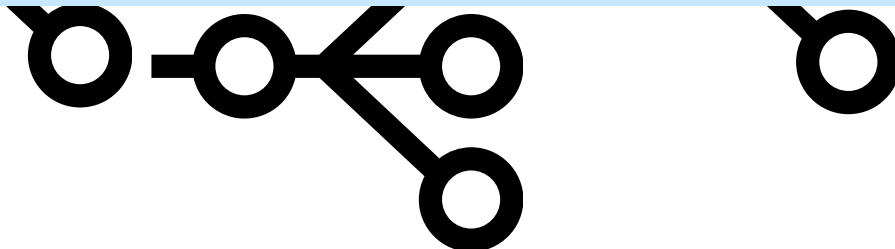
10s -> 100s

Few/year ->
1000s /year



How do we scale? Automate.

How do we automate? Align tools and process or find an interoperability solution



What is Eiffel?



- A protocol for machine to machine communication. Focused on CI/CD use cases.

Why should I use it?

- Decouple tools and processes – still control the complete CI/CD pipelines and know what is going on.

The Eiffel protocol

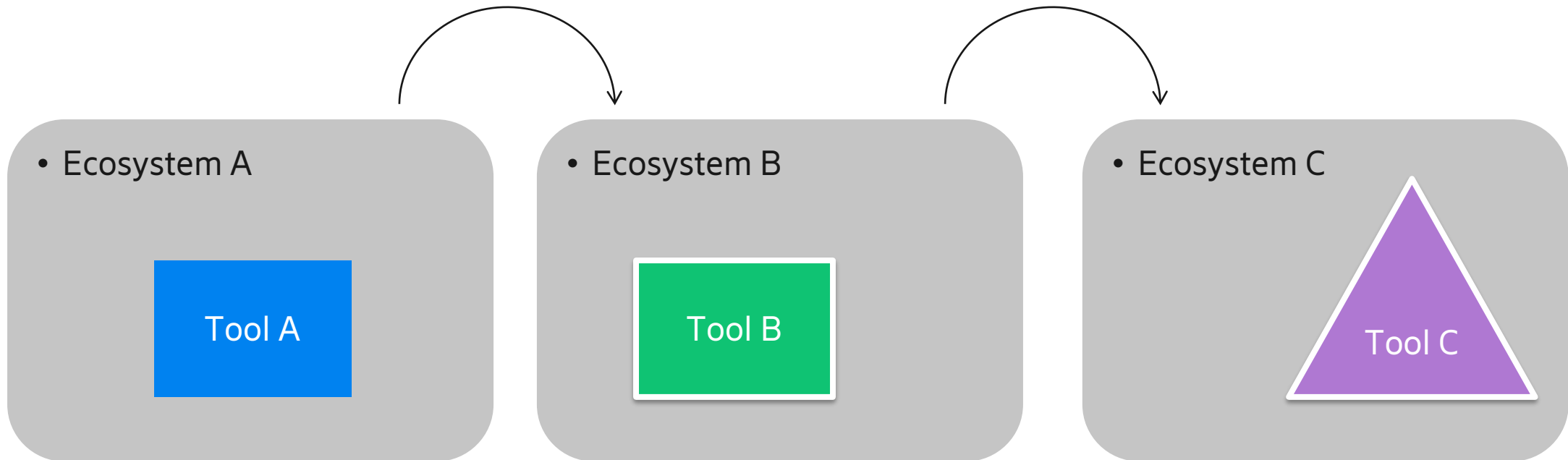


- Started in Ericsson 2012
- Open source 2016
 - <https://github.com/eiffel-community/eiffel>
- Technical committee
 - Ericsson
 - Axis Communications
 - Volvo
- In the Continuous Delivery foundation landscape
 - <https://landscape.cd.foundation/>

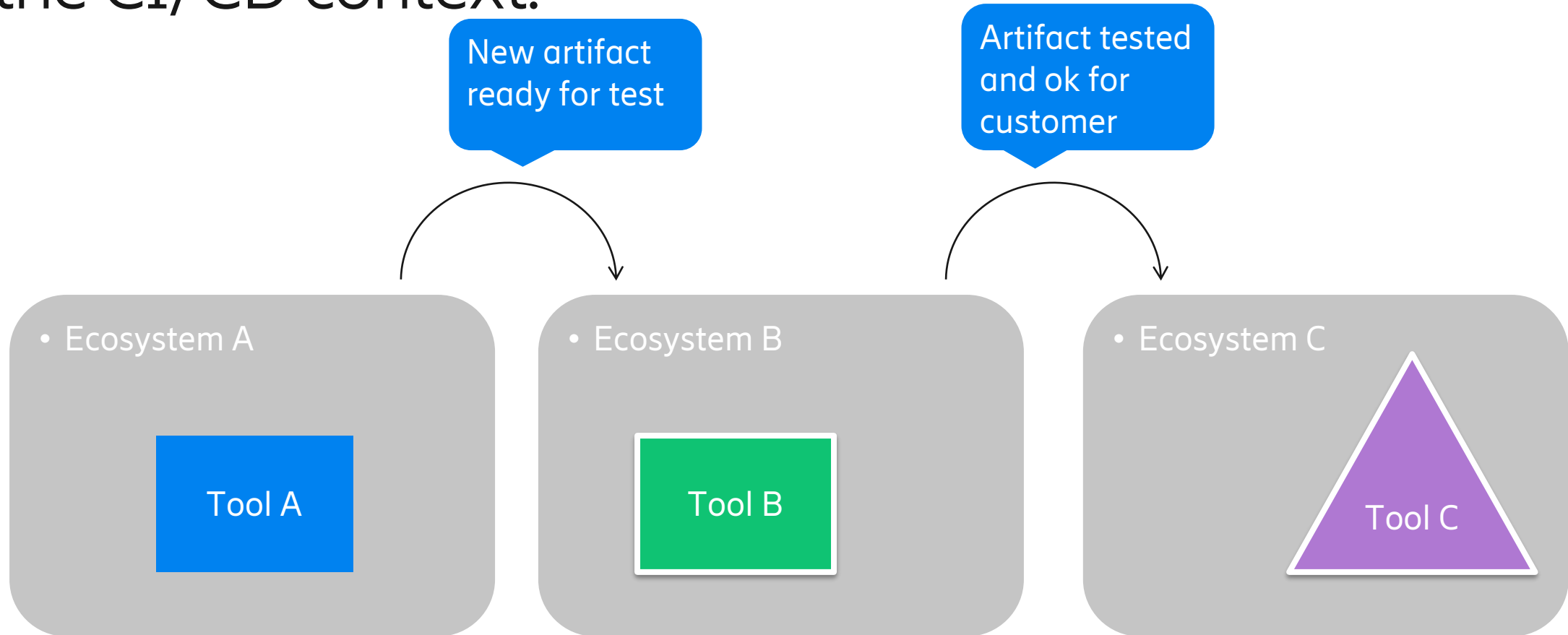
Large scale software development going into CI/CD/DevOps needs interoperability



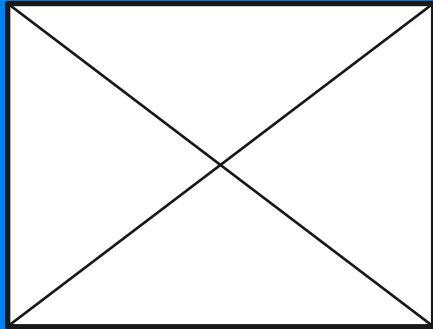
- Process
- Ad hoc automation setups



Provide a standardized M2M language for tools in the CI/CD context.



Eiffel protocol basics



The Eiffel protocol defines events.

- EiffelArtifactCreatedEvent
- EiffelActivityStartedEvent
- EiffelTestCaseStartedEvent

There are different event types.

```
{
  "meta": {
    "type":
    "EiffelArtifactCreatedEvent",
  },
  "data": {
    "identity": "pkg:maven/artifact-
name@2.1.7",
    "fileInformation": [
      {
        "name": "debug/launch",
        "tags": ["debug", "launcher"]
      }
    ],
  },
}
```

Json

The protocol – event types



Artifact related –
source code,
artifacts, issues,
compositions.

Activity related –
triggered, started,
stopped doing
something.

Test related –
test suites, test
cases.

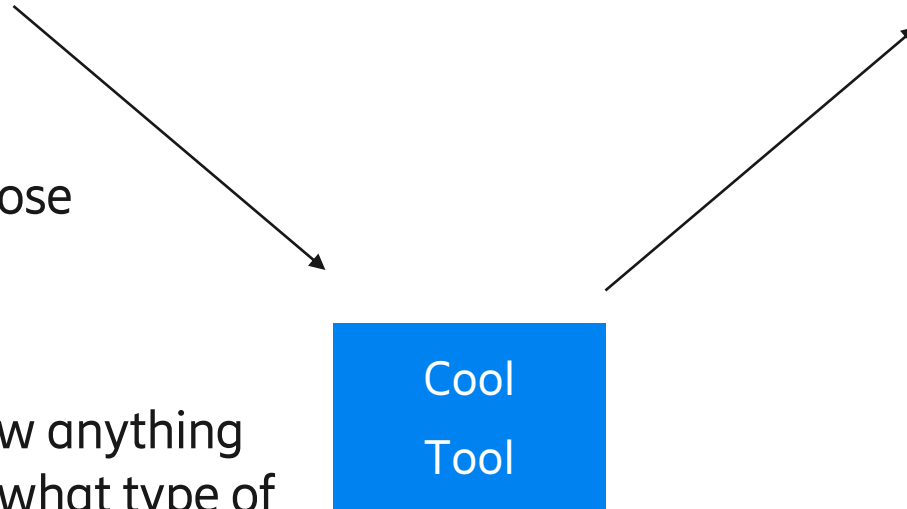
Using Eiffel events - decoupling



Receive events.

Decide what you are interested in and get those events delivered.

You do not need to know anything about the sender. Only what type of information that you want.

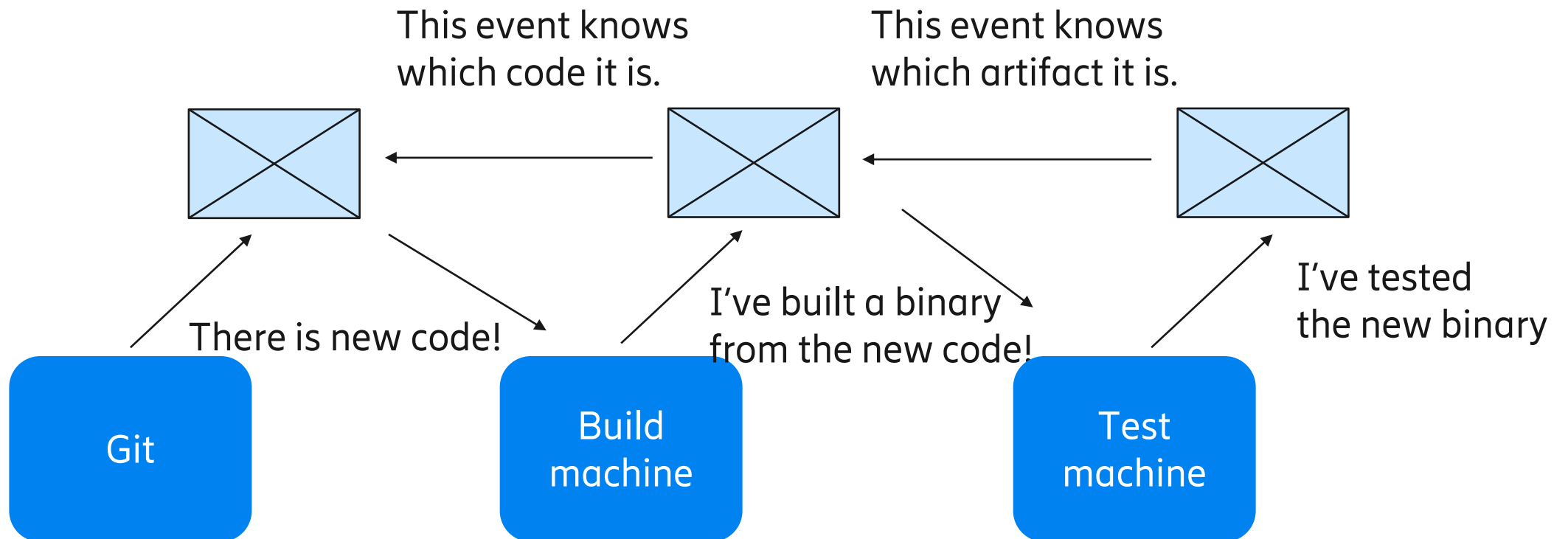


Send events.

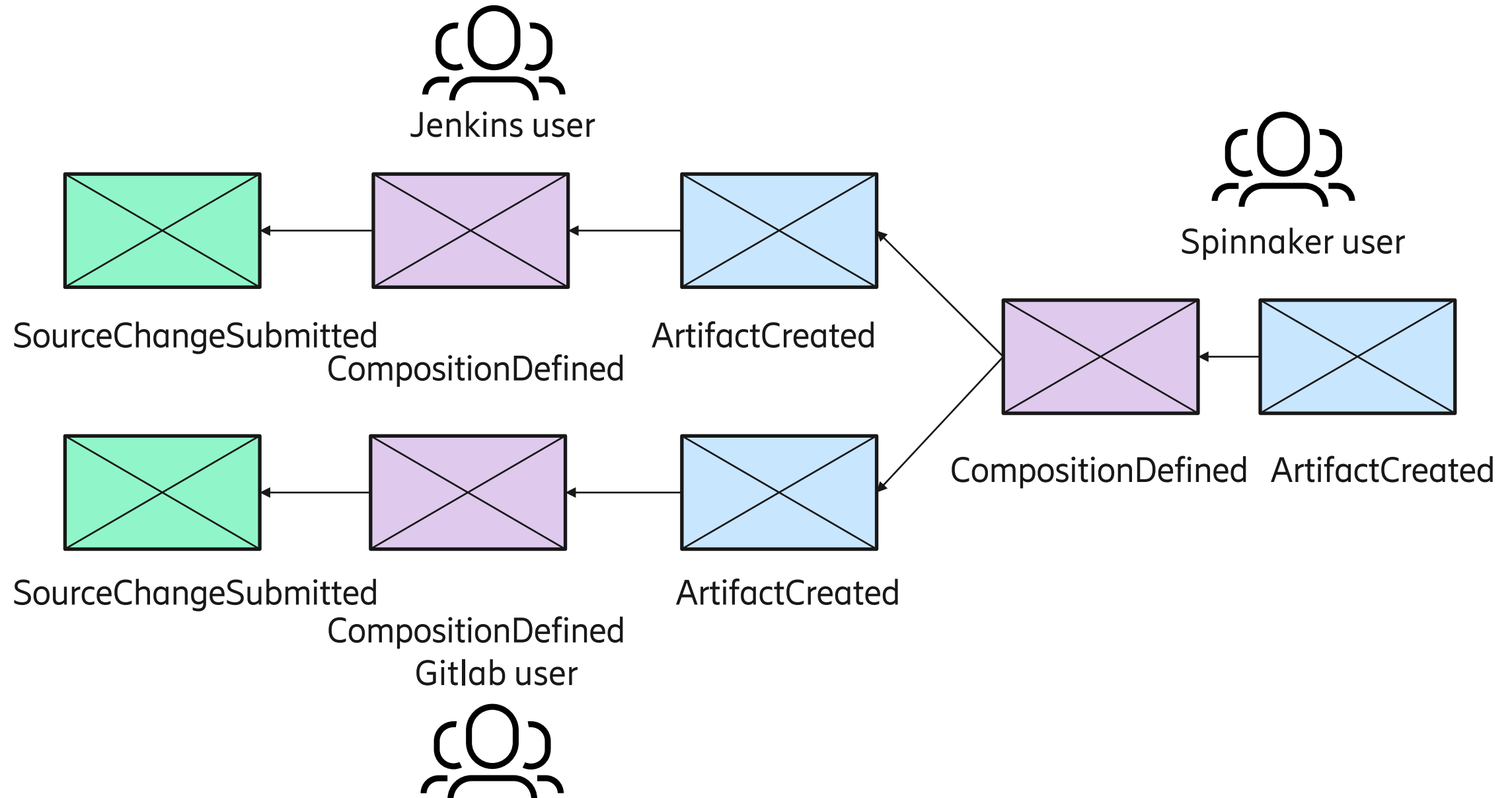
Tell the world about what you are doing and the results of that.

Broadcast. You do not need to know anything about the receivers.

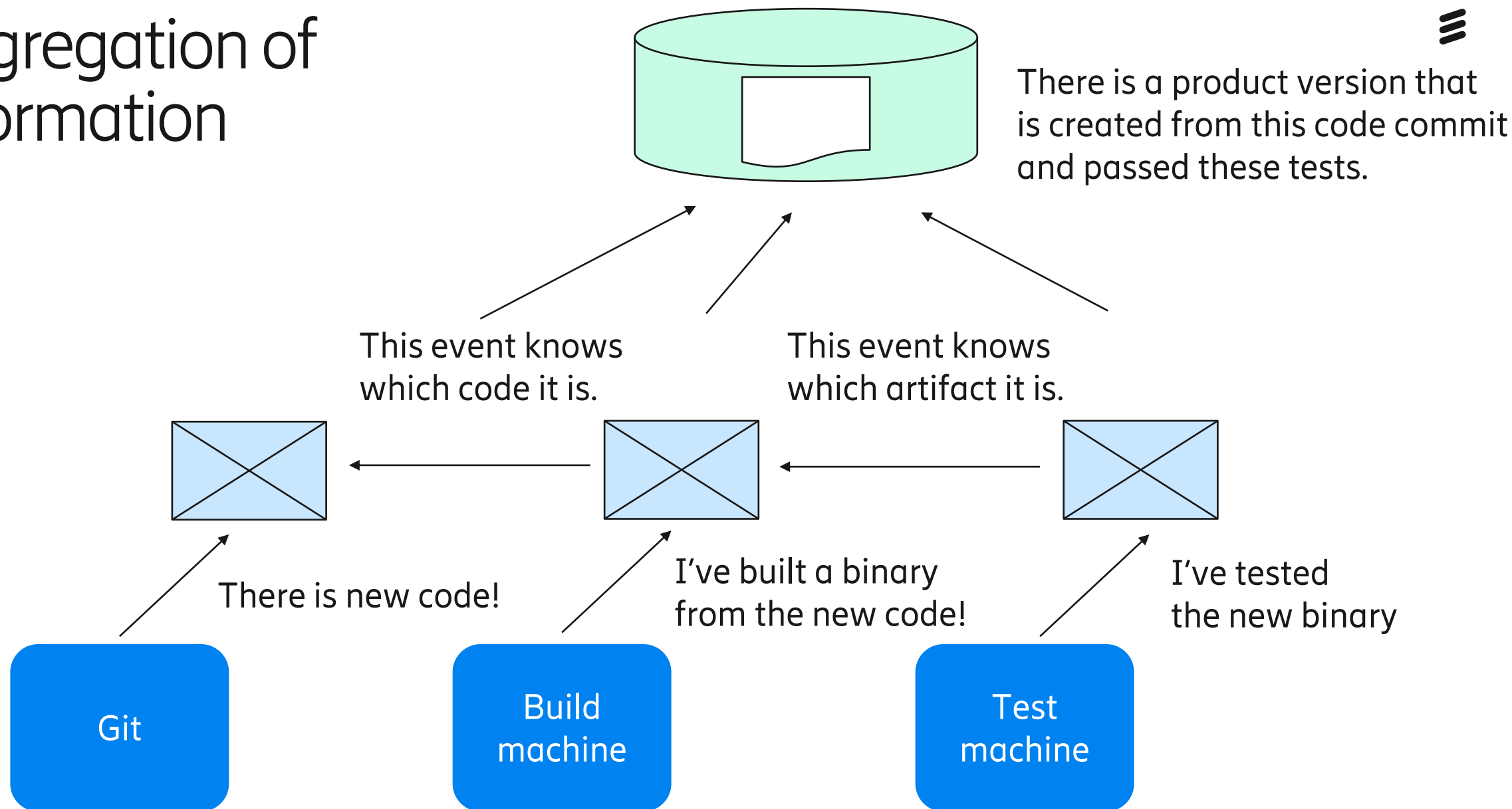
Simple example



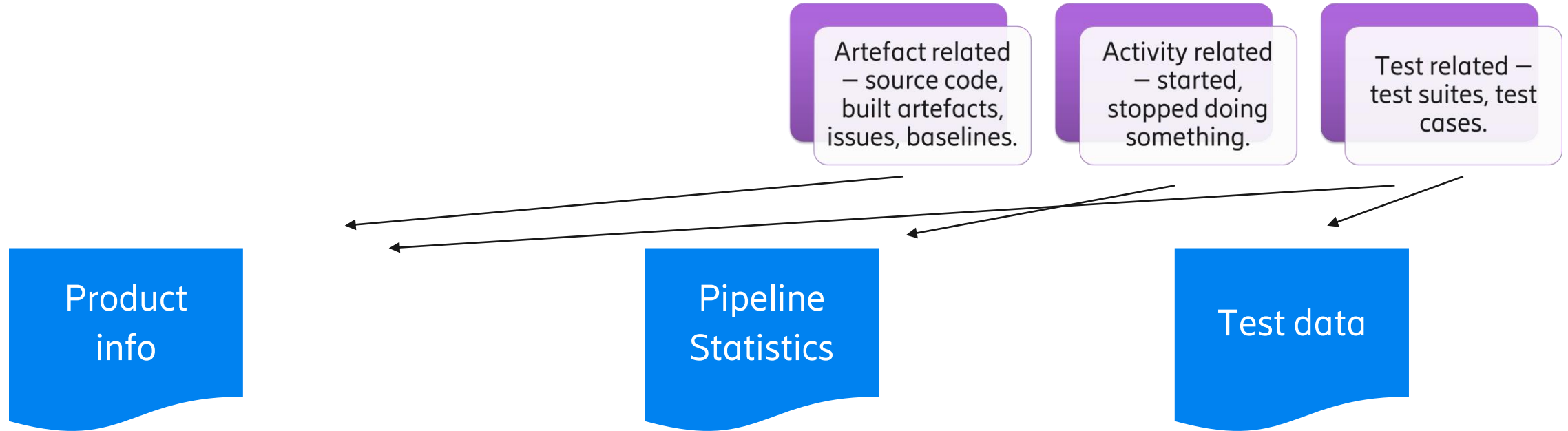
Bigger scale example



Aggregation of information

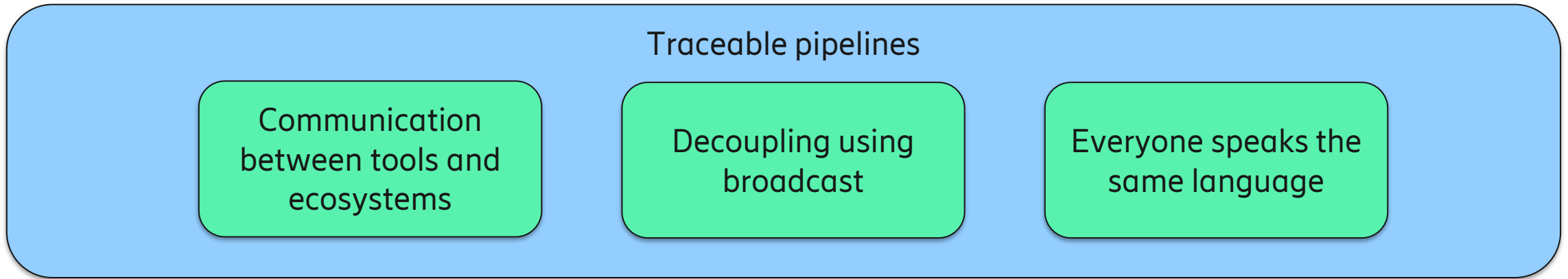


That's not all!

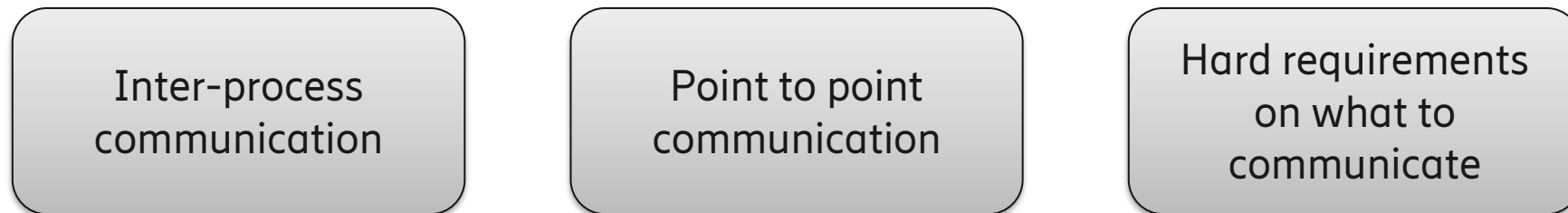


- There is a product version that is created from this code and passed these tests.
- The previous version was...
- The commit difference between those version is...
- We run X builds every day.
- The frequency of test failure is Y.
- Our time from commit to customer is Z.
- We run X test cases every day.
- The full test coverage of product version Y is...
- Test case Z is flaky

Eiffel events give us



...but not



The protocol – the principles



- Events inform consumers about
 - what the producer is doing
 - what is happening to it
 - or a result that is achieved.
- Events are tool agnostic.
- Event shall not duplicate data.
- Events are broadcasted, never sent to a recipient.
- It is up to consumers to subscribe and decide whether events are of interest or not.

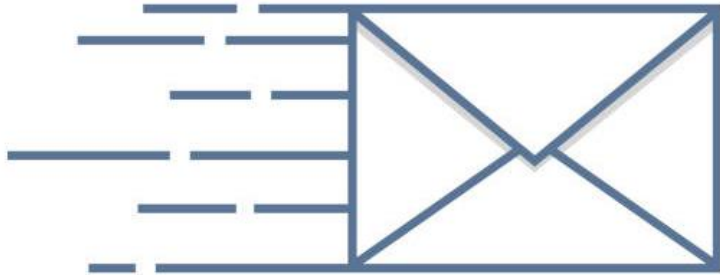
A protocol is not enough



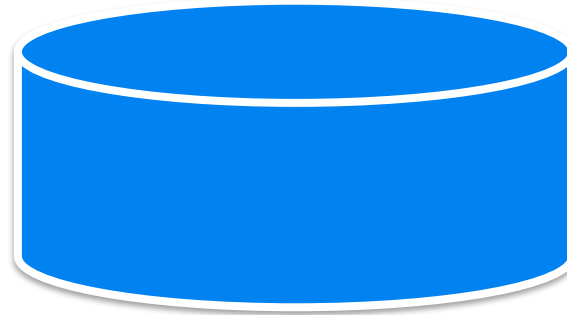
The supporting tools



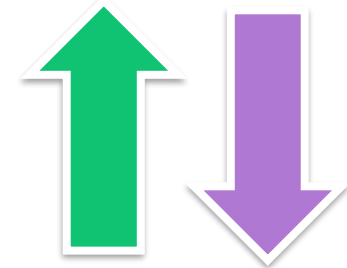
Message bus



Event aggregation

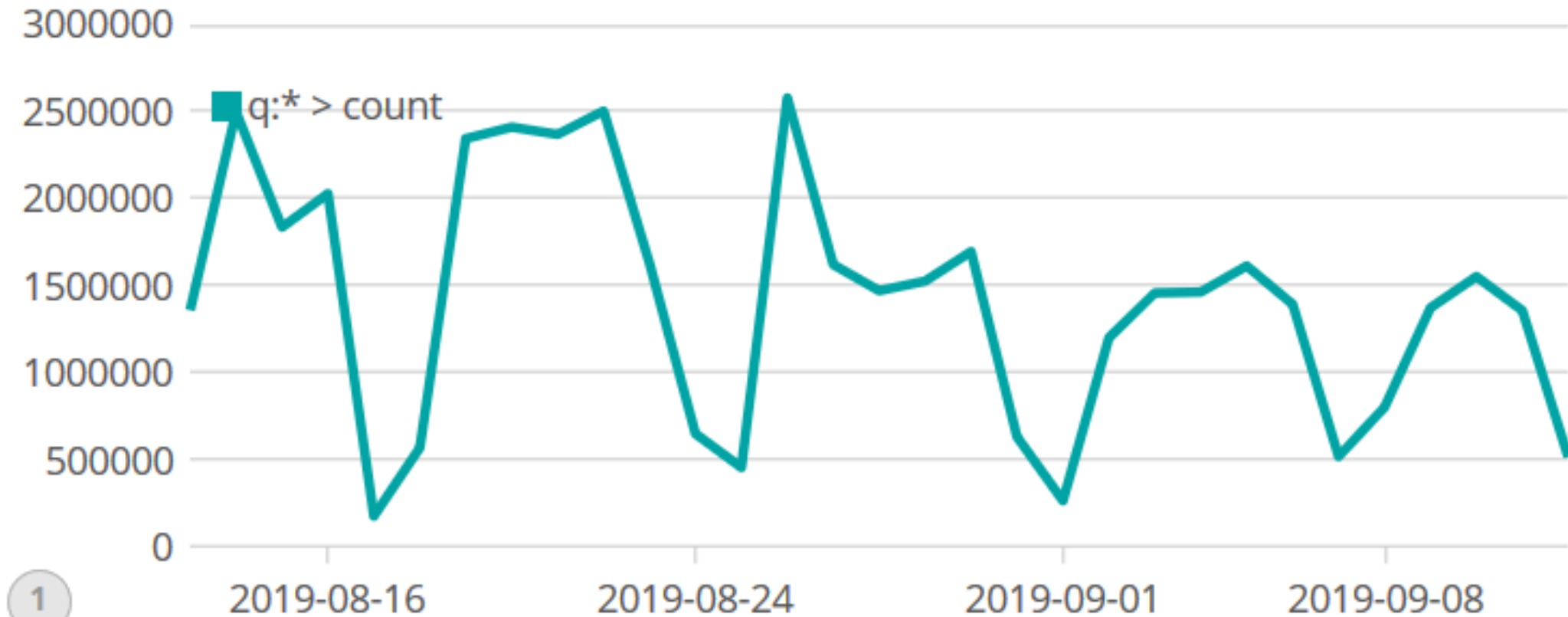


Event sender/receiver



Learn about this in the summit

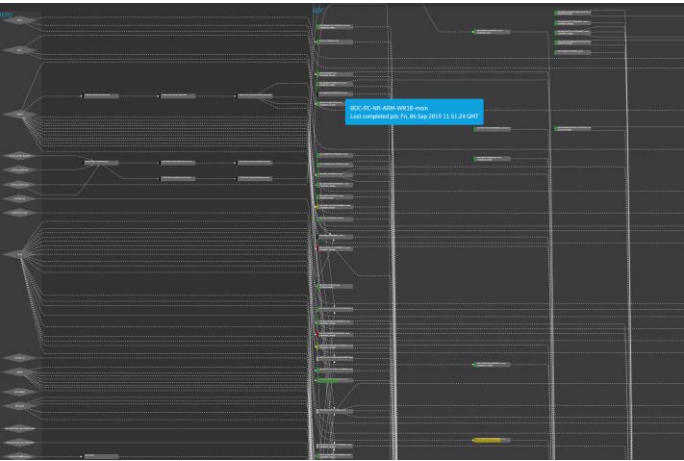
An example from Ericsson. Events per day.



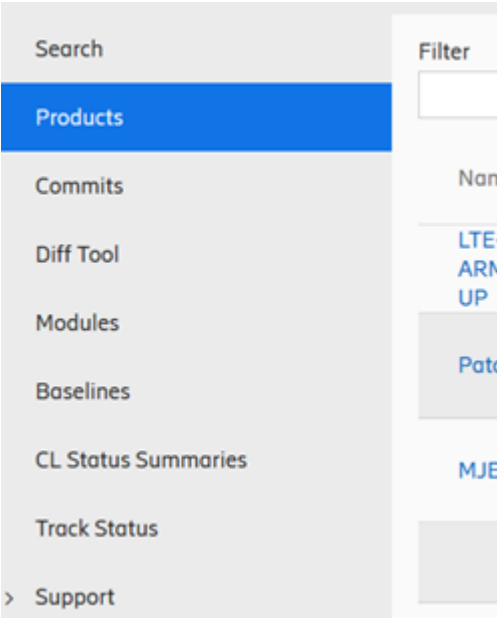
Where Ericsson uses Eiffel event data



Activity overview



Product information



Pipeline statistics



Test results



Learnings



01

Don't do everything at once. Eiffel can be applied on top of existing toolsets.

02

Don't duplicate functionality that is already available in a tool (e.g. Internal pipeline triggering)

What's next



When we started	Where we are going	What does that mean for the Eiffel protocol
Pipelines in CI/CD a new concept.	Pipeline functionality built into every decent tool. But still a local context.	Less focus on triggering every activity.
Events used but not in CI/CD.	Events becoming mainstream in CI/CD.	Tools support events out of the box.
CI focused	Convergence – development, integration, deployment and operations.	Find the interaction points. Continued collaboration with tool teams and communities.