

Messaging in practice



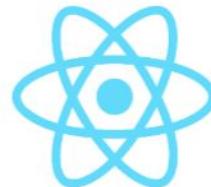
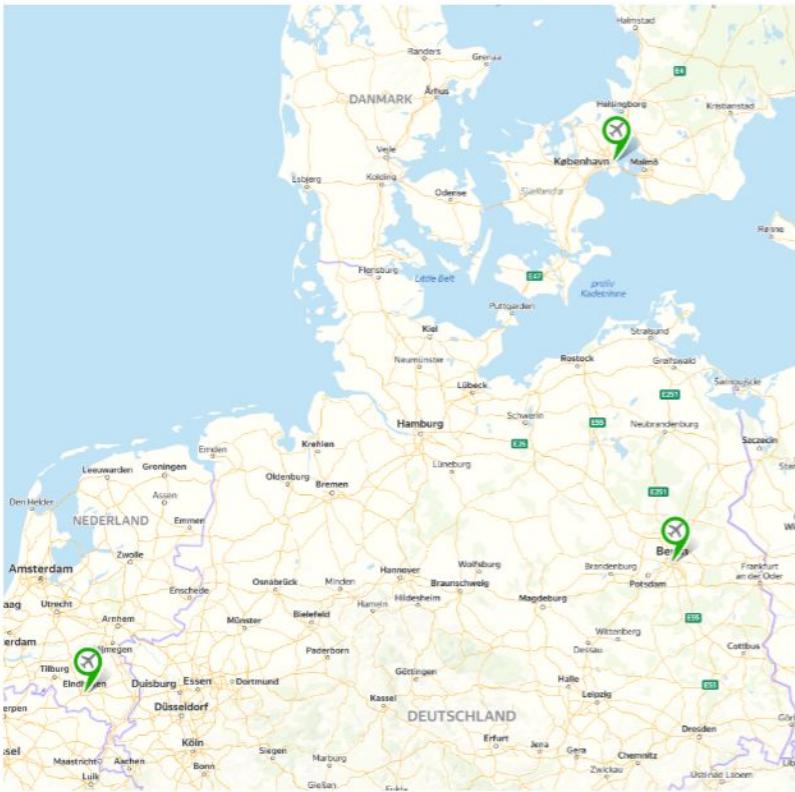
Daniil Doronkin

Senior Software Engineer at Templafy

<https://www.linkedin.com/in/ddydeveloper/>

telegram: paranamix2

email: ddydeveloper@gmail.com



<https://wwwtemplafycom/>

<https://www.linkedincom/companytemplafy>

<https://infotemplafycom/blog>

Templafy 

We are hiring



Senior Software Engineer
.NET - Desktop Application
Templafy
Eindhoven, North Brabant,

Be an early applicant

3 days ago



Full-stack .NET Engineer -
Integrations
Templafy
Eindhoven, North Brabant,

Be an early applicant

1 week ago

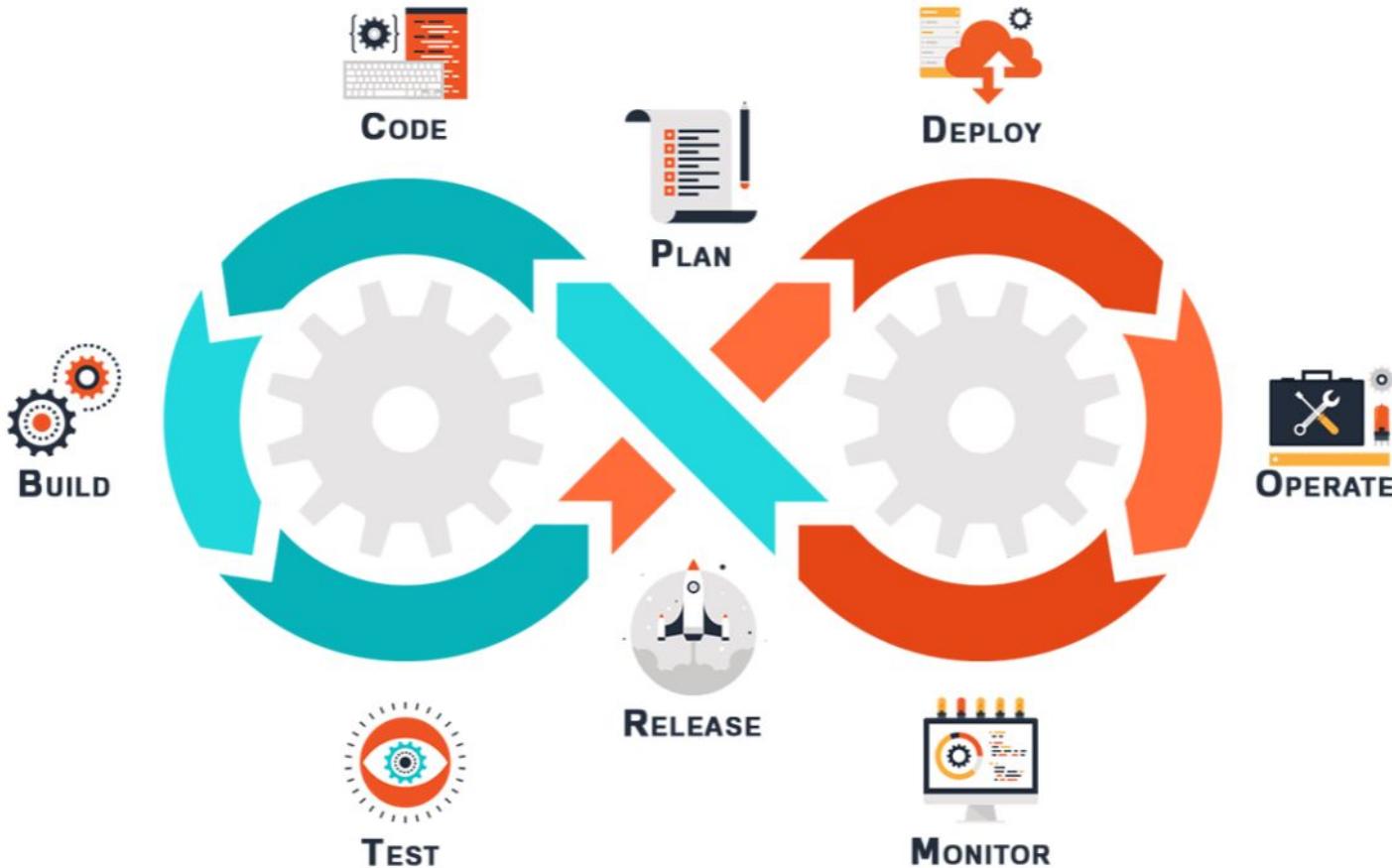


Senior Software Engineer
.NET
Templafy
Eindhoven, North Brabant,

Be an early applicant

2 weeks ago

<https://www.linkedin.com/company/templafy/jobs/>



Meetup

Meetup

Cheap development

Meetup

Cheap development

Less dependencies

Meetup

Cheap development

Less dependencies

Entire end-to-end tests

Meetup

Cheap development

Less dependencies

Entire end-to-end tests

Easy deployment

Meetup

~~Cheap development~~

~~Less dependencies~~

~~Entire end-to-end tests~~

~~Easy deployment~~

Speaker

Trainer

Producer

Host

Audience

Meetup

Speaker applied

Speaker prepared

Sponsor confirmed

Meetup announced

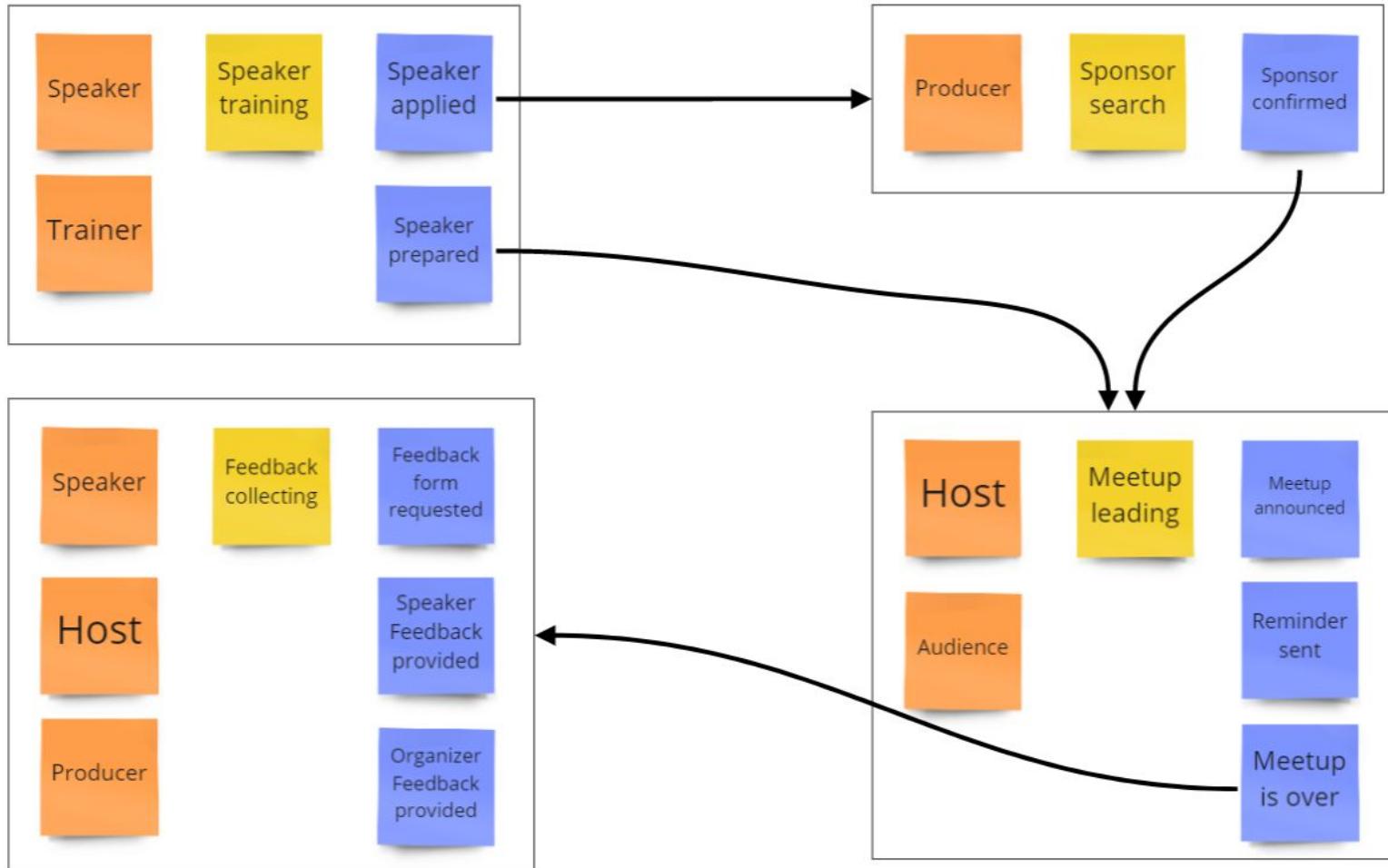
Reminder sent

Meetup is over

Feedback form requested

Speaker Feedback provided

Organizer Feedback provided



Meetup

Speaker
training

Sponsor
search

Feedback
collecting

Meetup
leading

Speaker
training

Sponsor
search

Less code

Feedback
collecting

Meetup
leading

Speaker
training

Sponsor
search

Less code
Scaling

Feedback
collecting

Meetup
leading

Speaker
training

Sponsor
search

Less code

Scaling

Tests

Feedback
collecting

Meetup
leading

Speaker
training

Sponsor
search

Less code

Scaling

Tests

Feedback
collecting

Meetup
leading

Integration & Delivery

Speaker
training

Sponsor
search

Less code

Scaling

Tests

Feedback
collecting

Meetup
leading

Integration & Delivery

Tech stack

Speaker
training

Sponsor
search

Infrastructure

Feedback
collecting

Meetup
leading

Speaker
training

Sponsor
search

Infrastructure

Boilerplate

Feedback
collecting

Meetup
leading

Speaker
training

Sponsor
search

Infrastructure

Boilerplate

Data consistency (BASE vs ACID)

Feedback
collecting

Meetup
leading

Speaker
training

Sponsor
search

Feedback
collecting

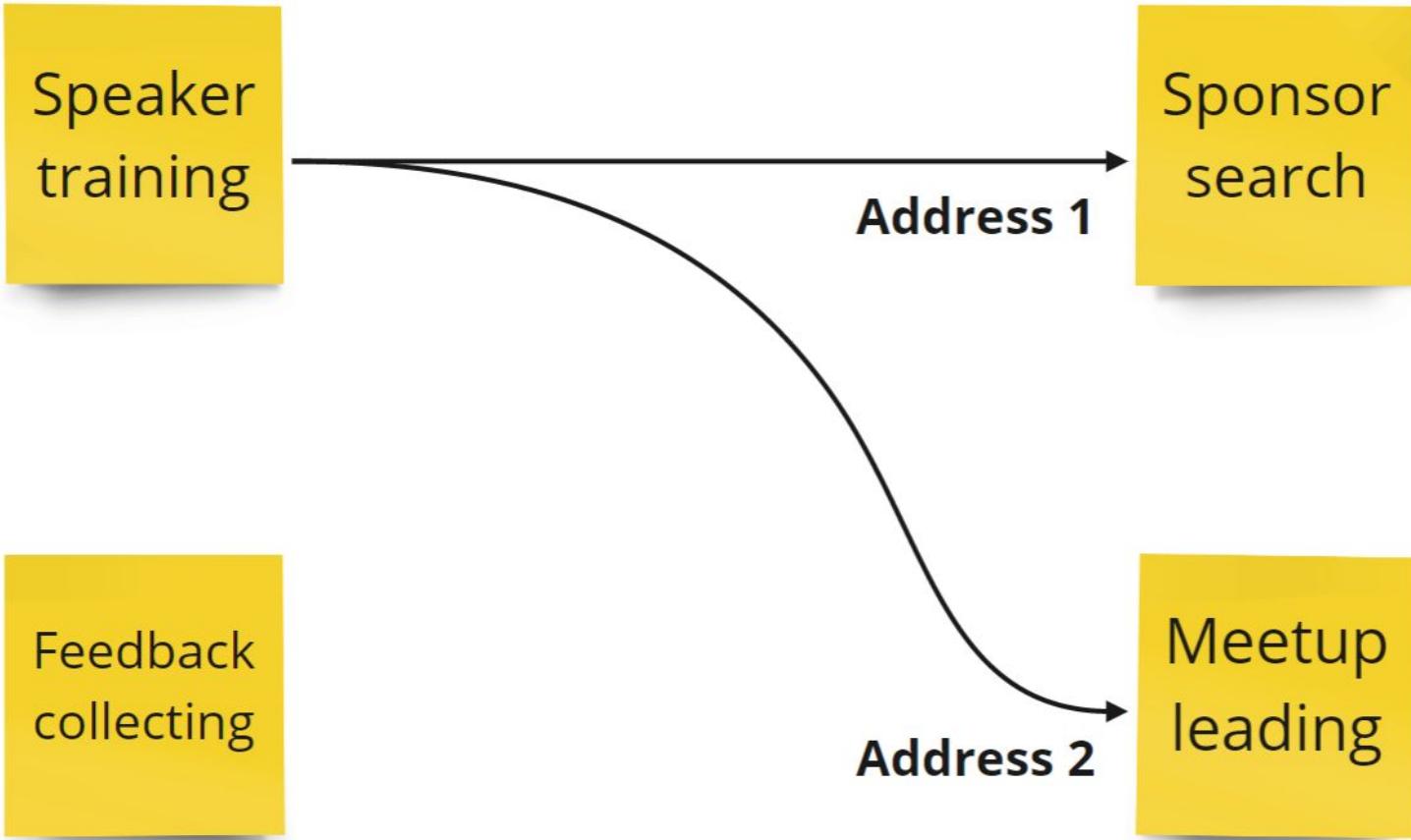
Meetup
leading

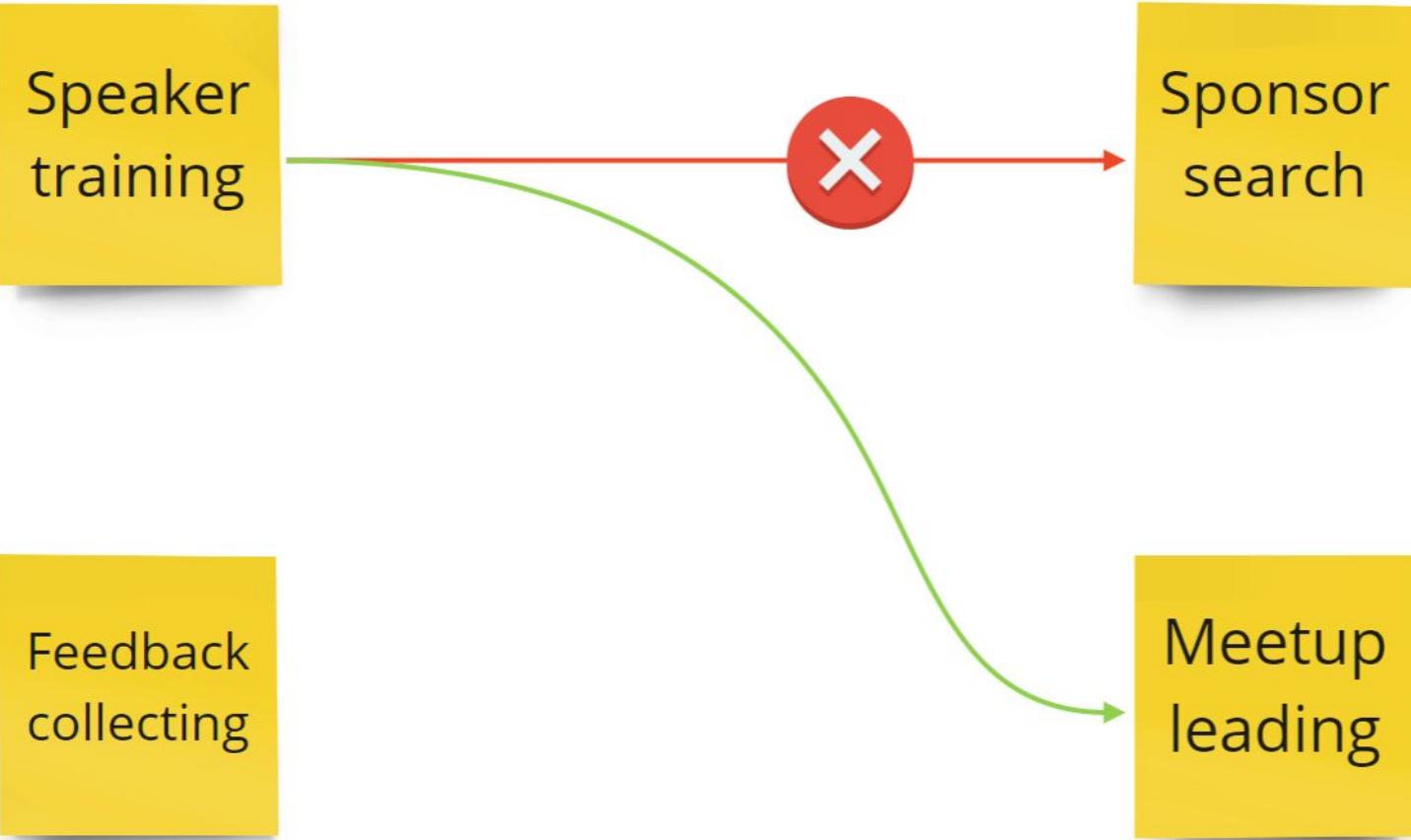
Infrastructure

Boilerplate

Data consistency (BASE vs ACID)

Communications





Service A
response time

Service B
response
time

Service C
response time

Service D
response time

100ms

200ms

300ms

400ms

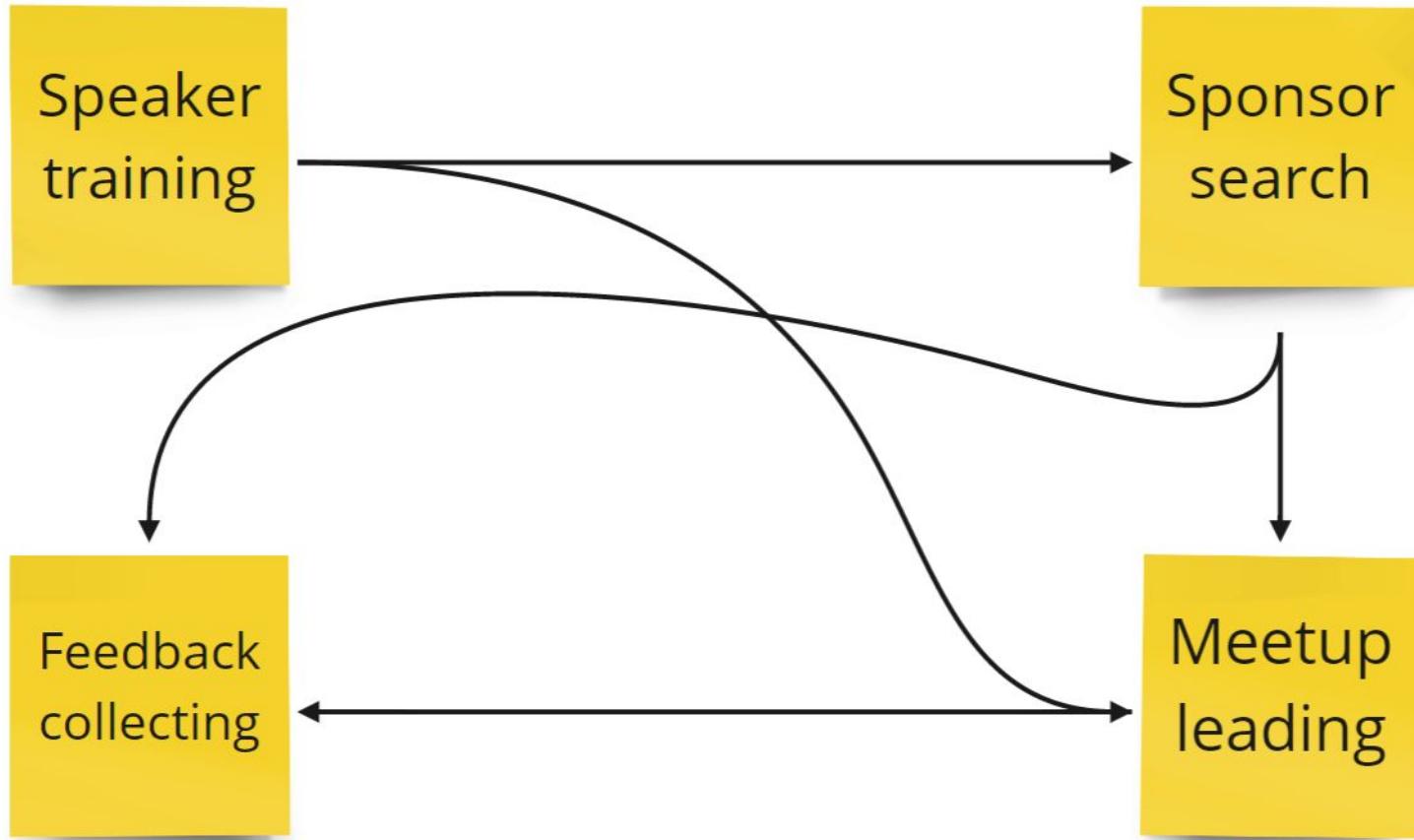
500ms

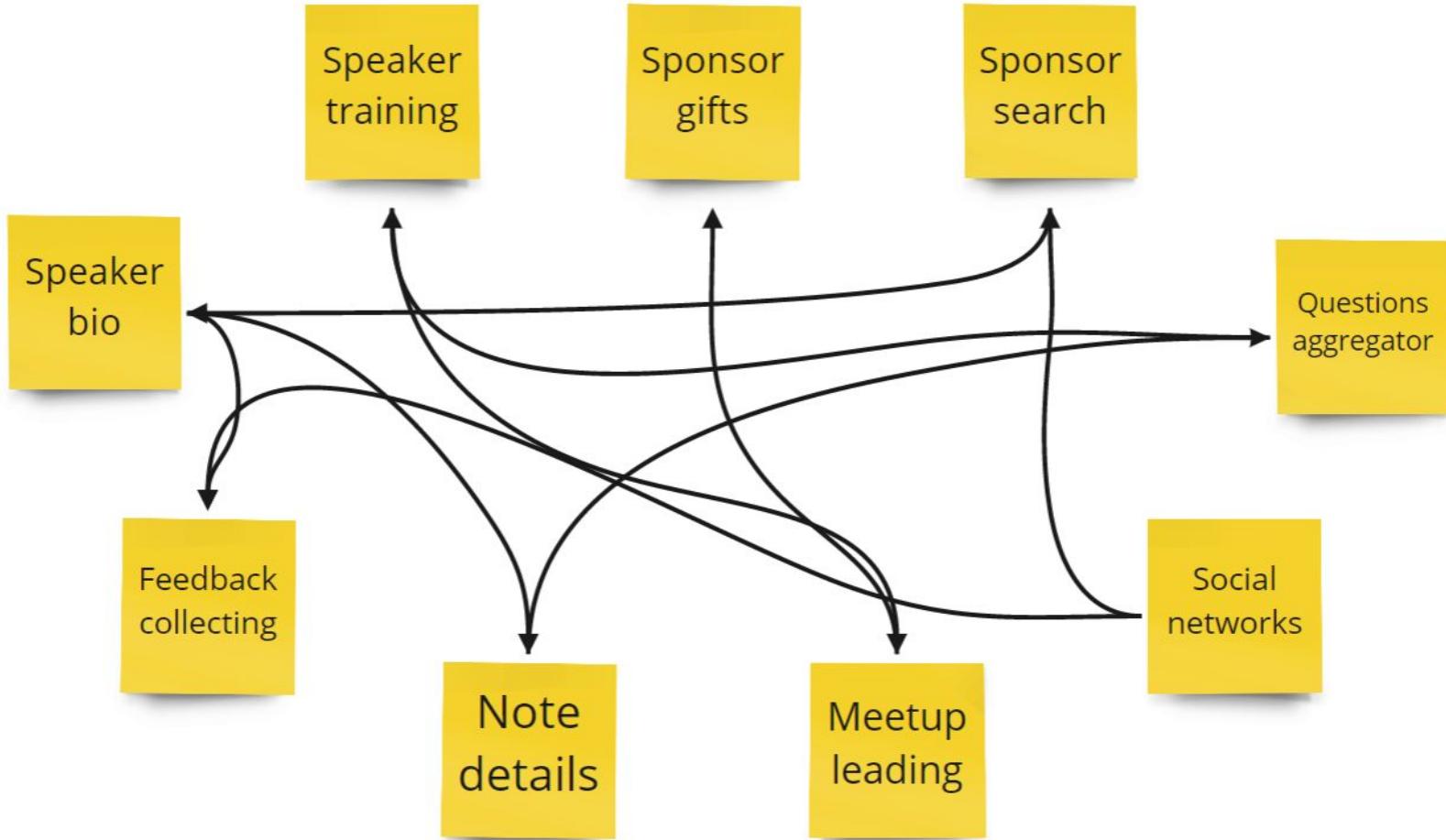
600ms

700ms

800ms







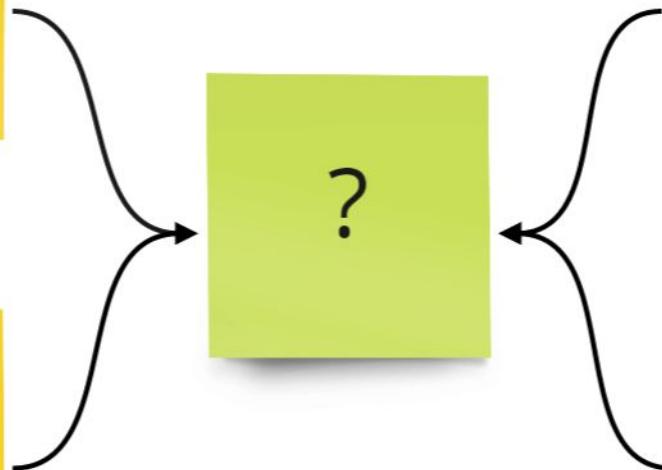
Speaker
training

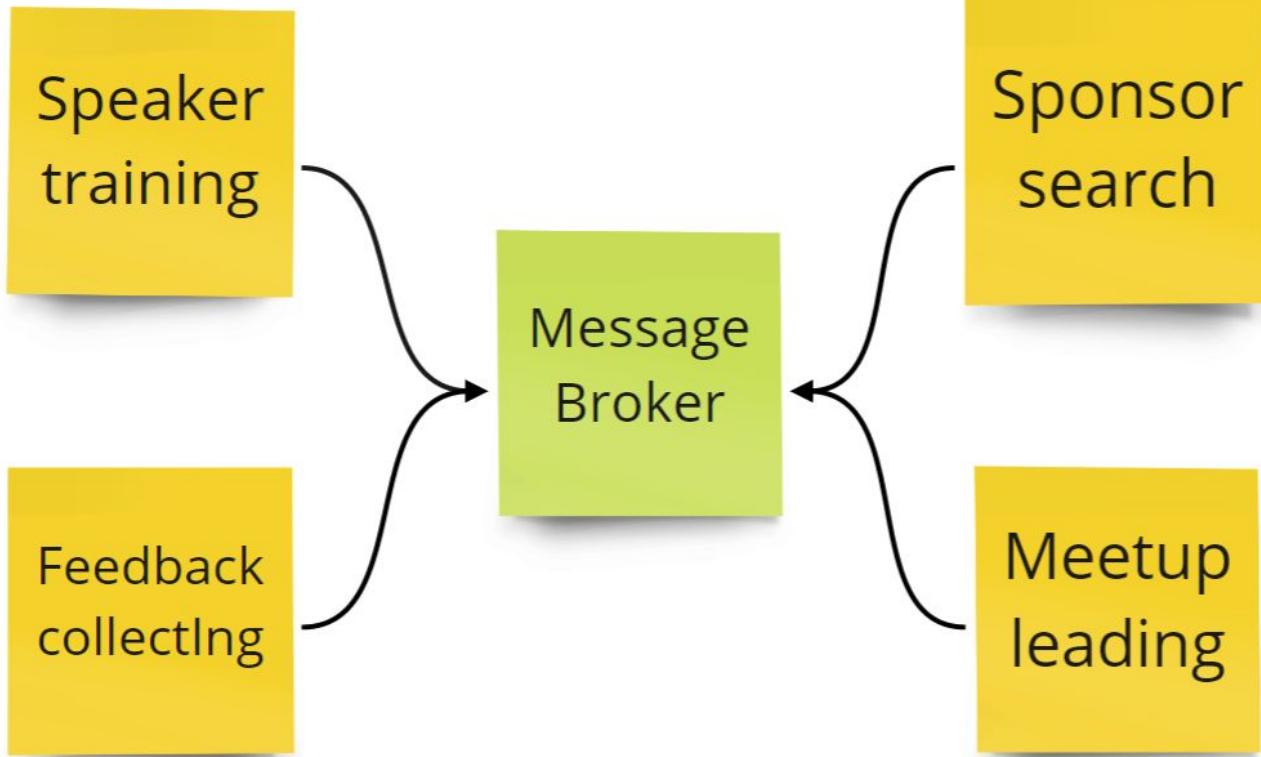
Feedback
collecting

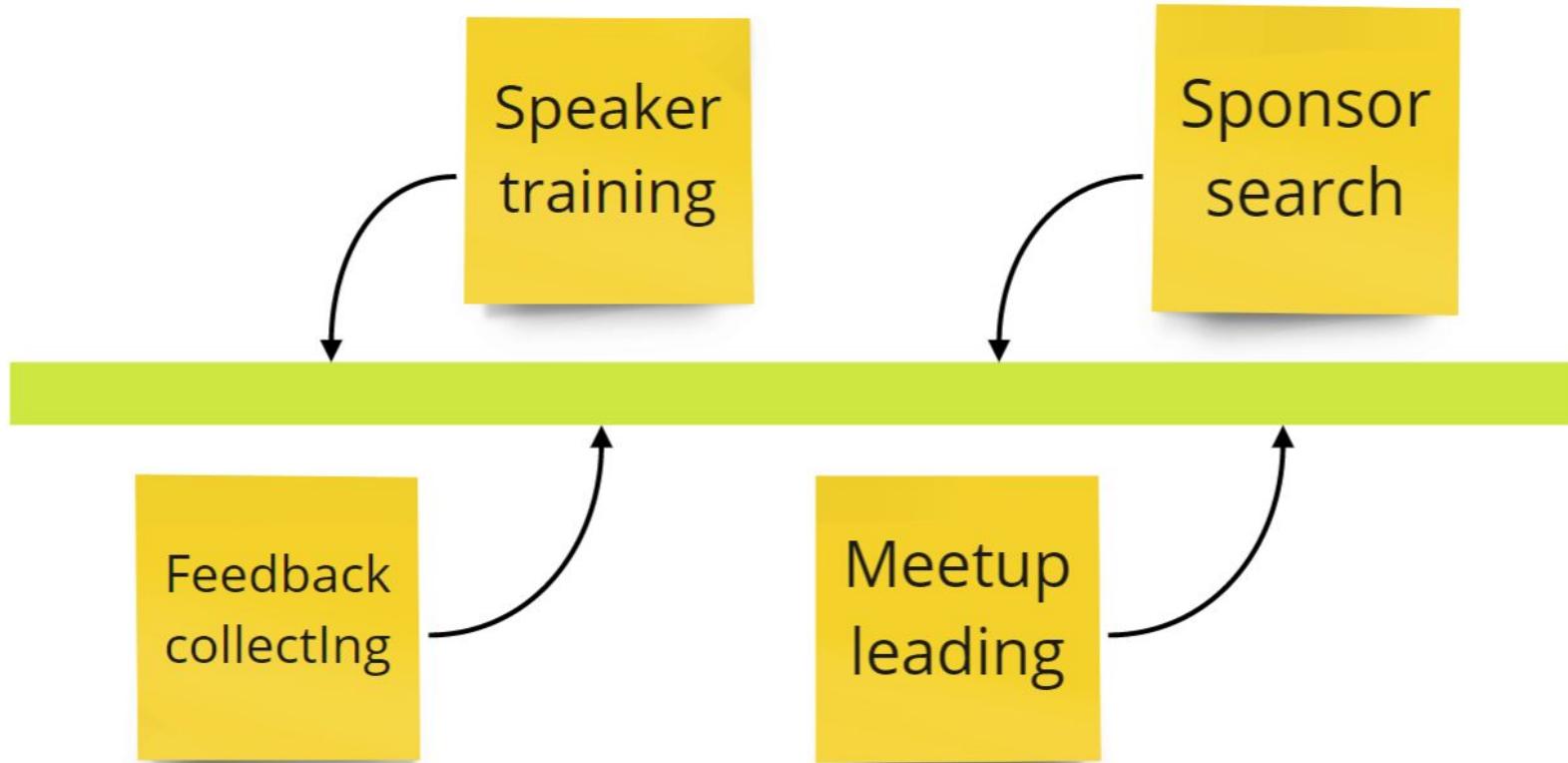
Sponsor
search

Meetup
leading

?







What is message
broker?

AMQP

Open binary protocol

AMQP

Open binary protocol

0.9.1 AMQ Broker model:

exchange, binding, queue

AMQP

Open binary protocol

0.9.1 AMQ Broker model:

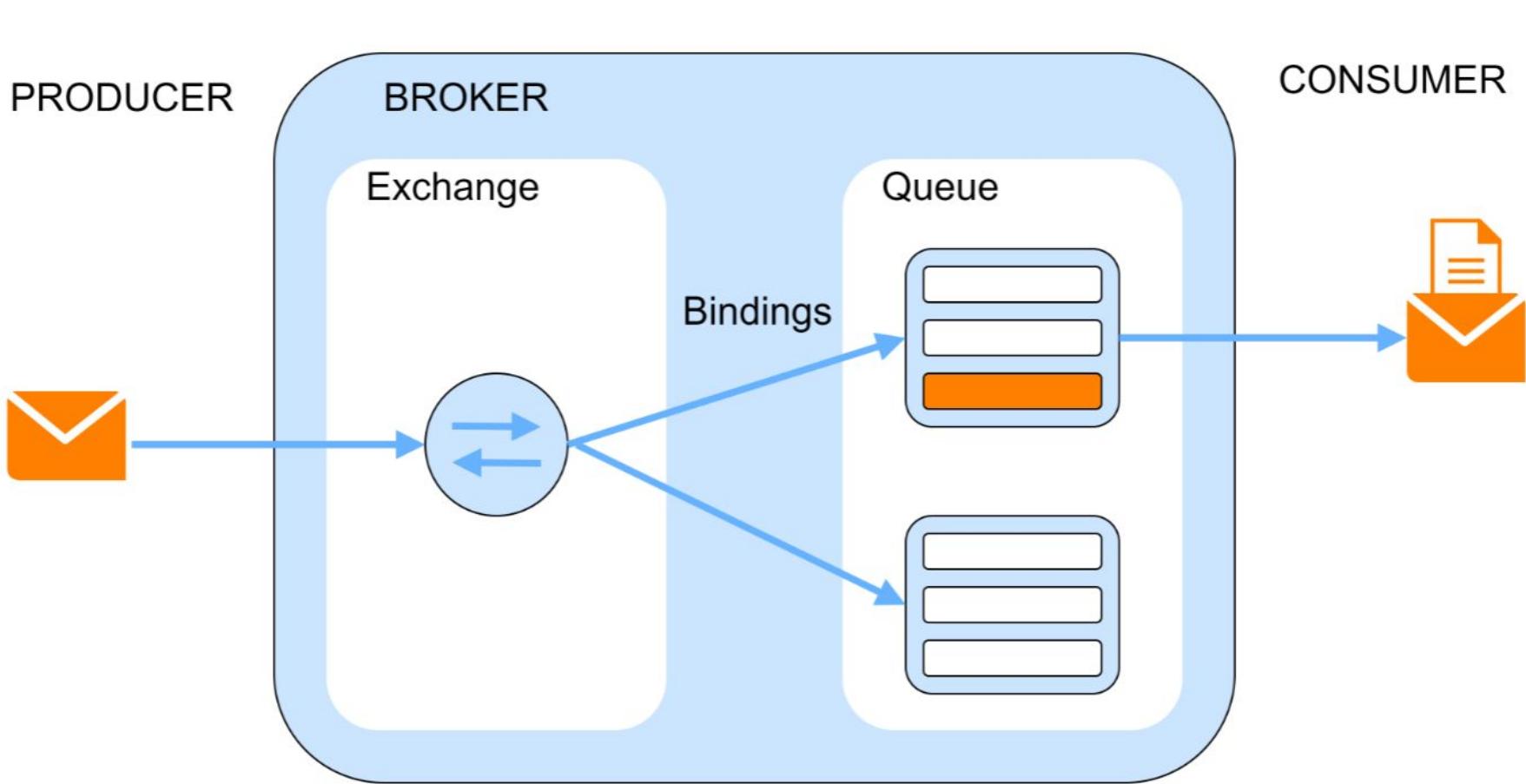
exchange, binding, queue

1.0 completely different from 0.9.1

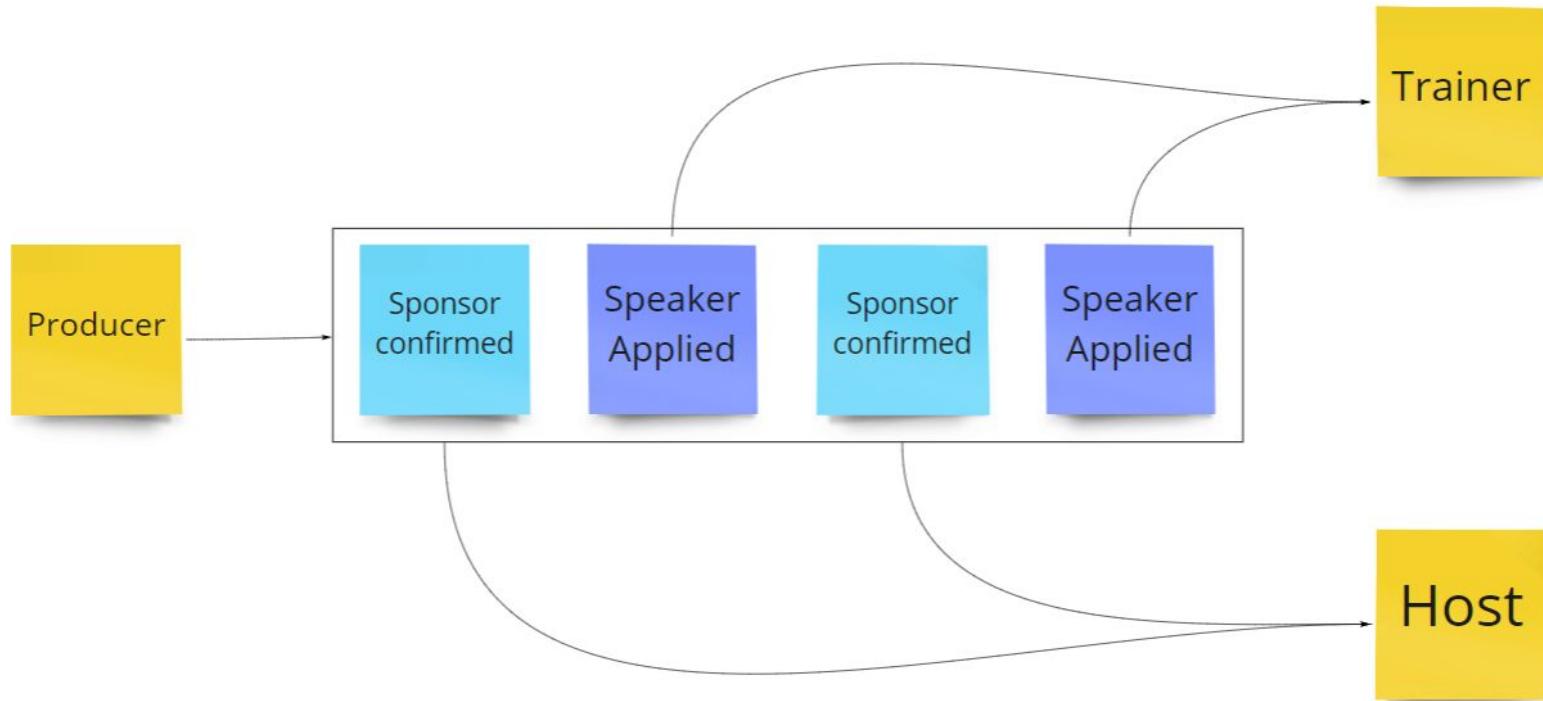
PRODUCER

BROKER

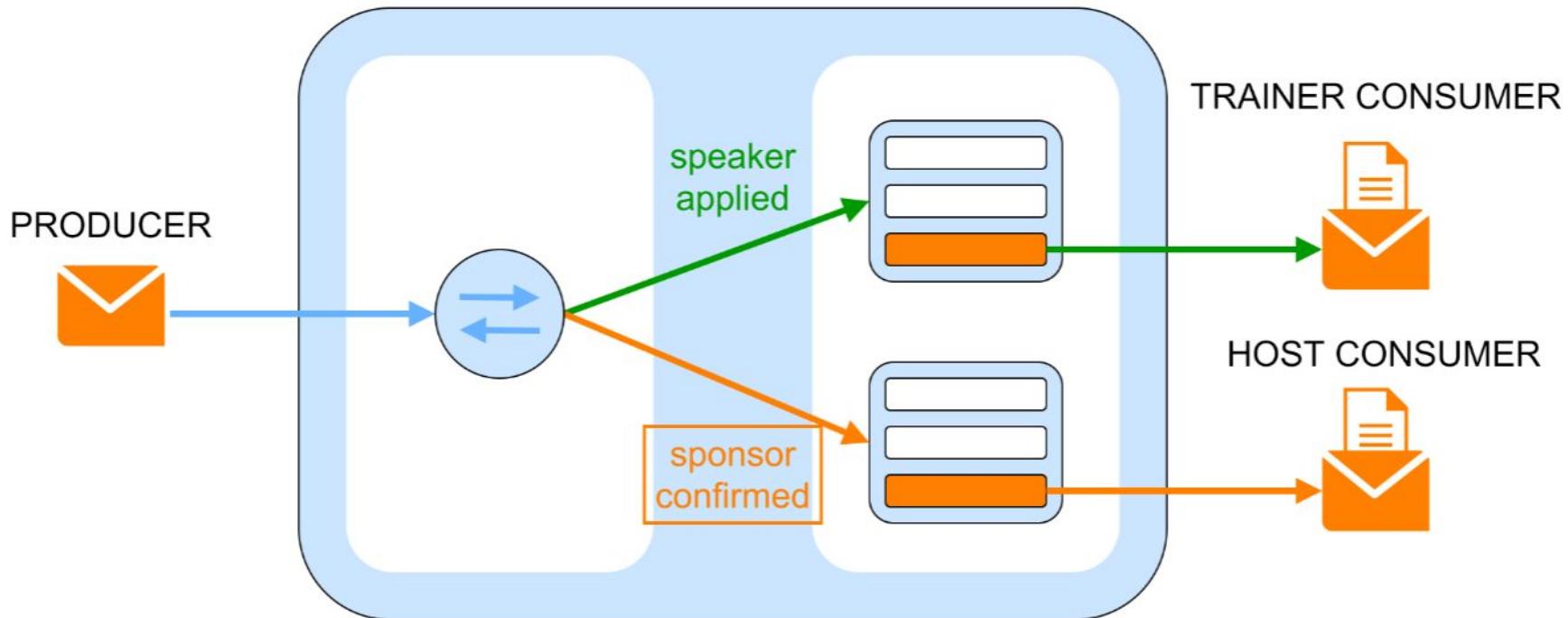
CONSUMER

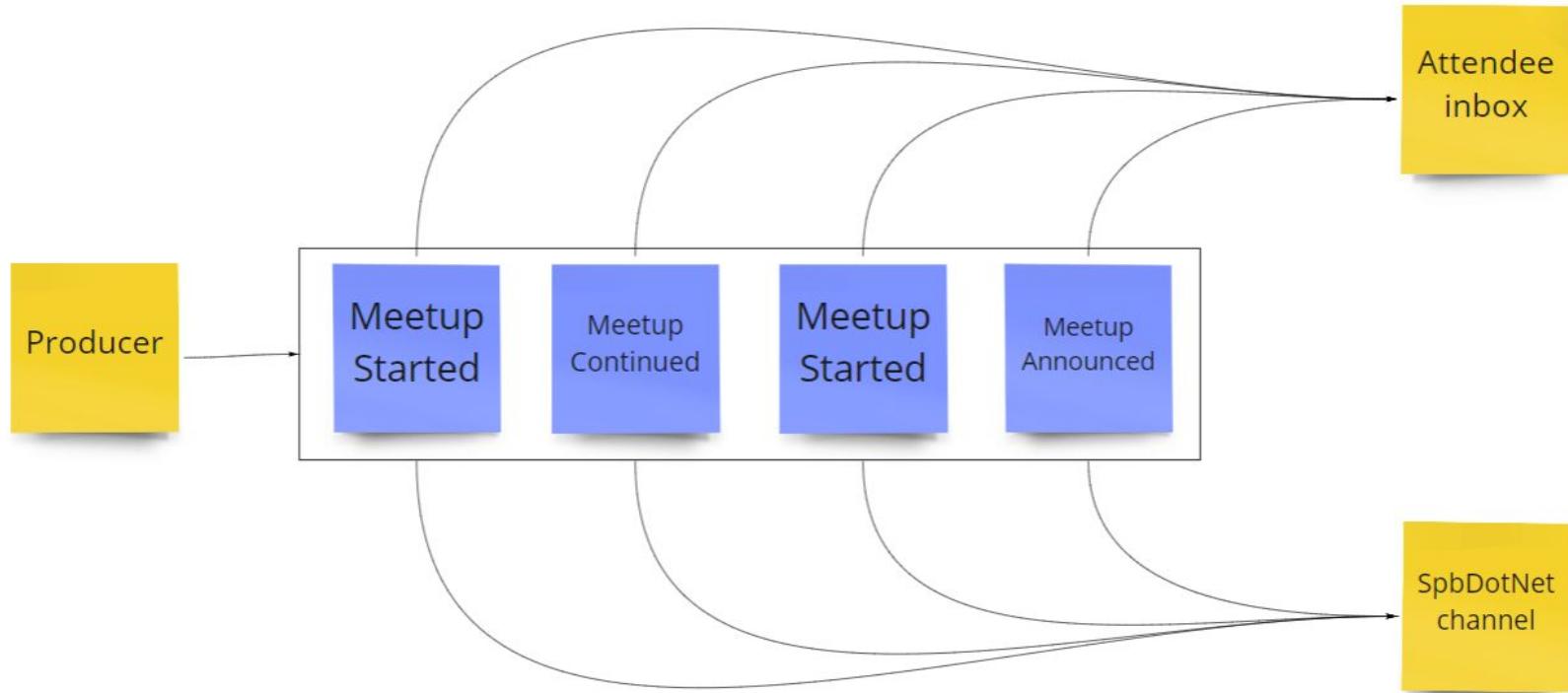


How can we use
messaging?

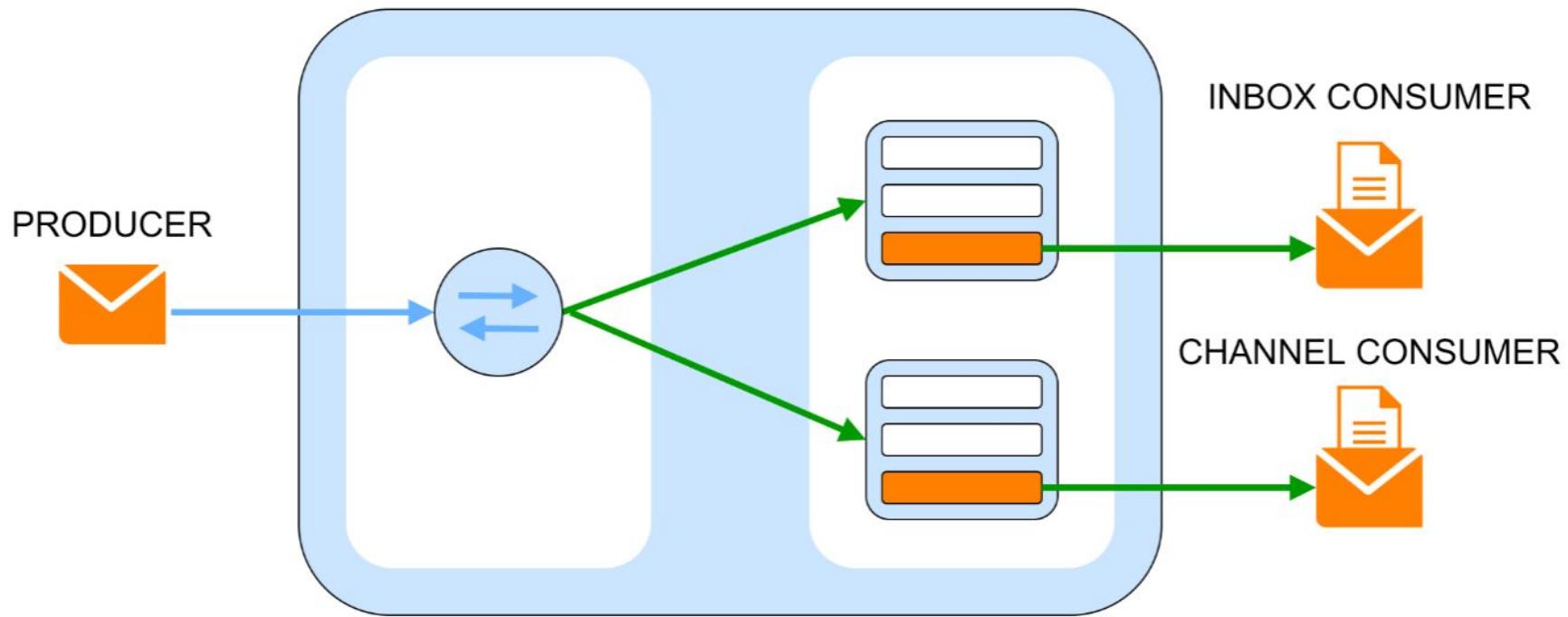


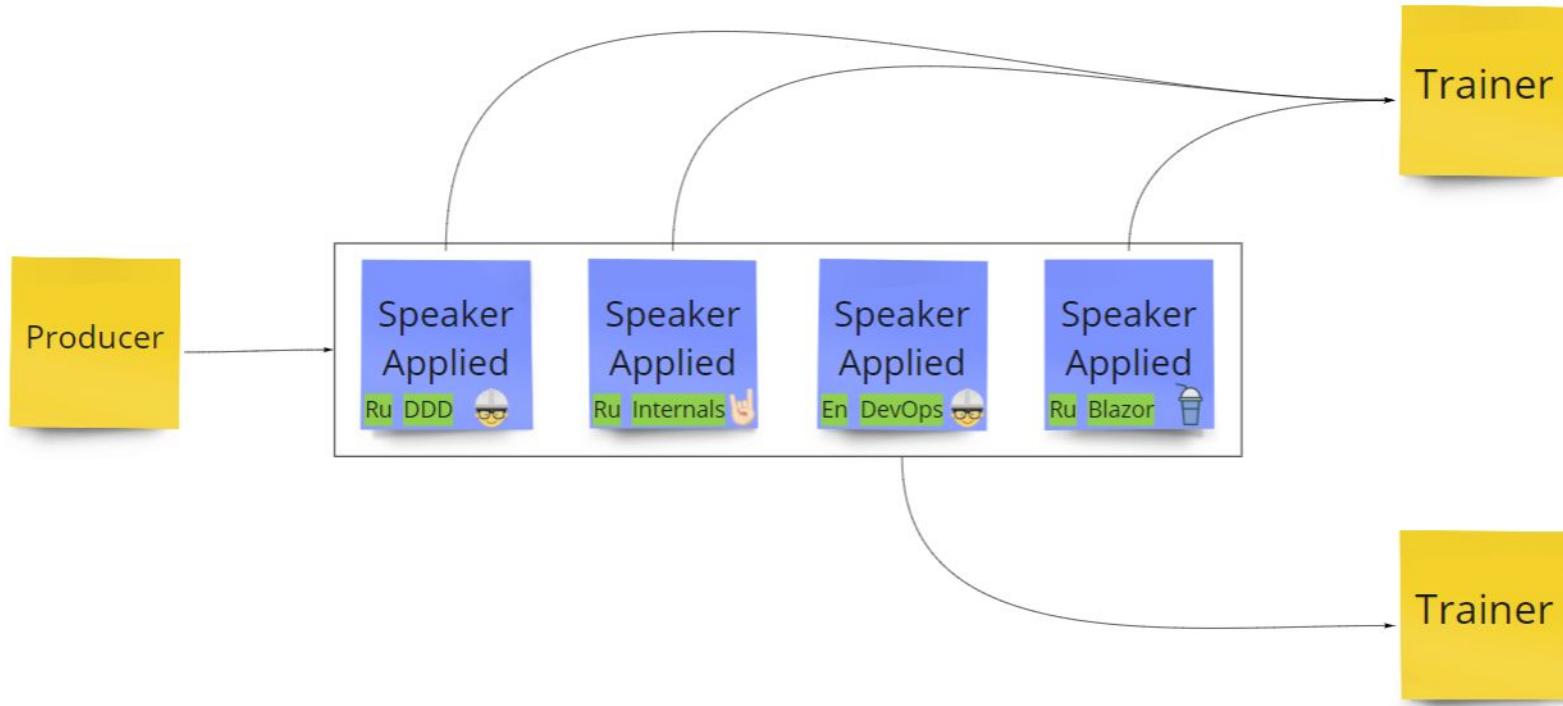
Routing pattern



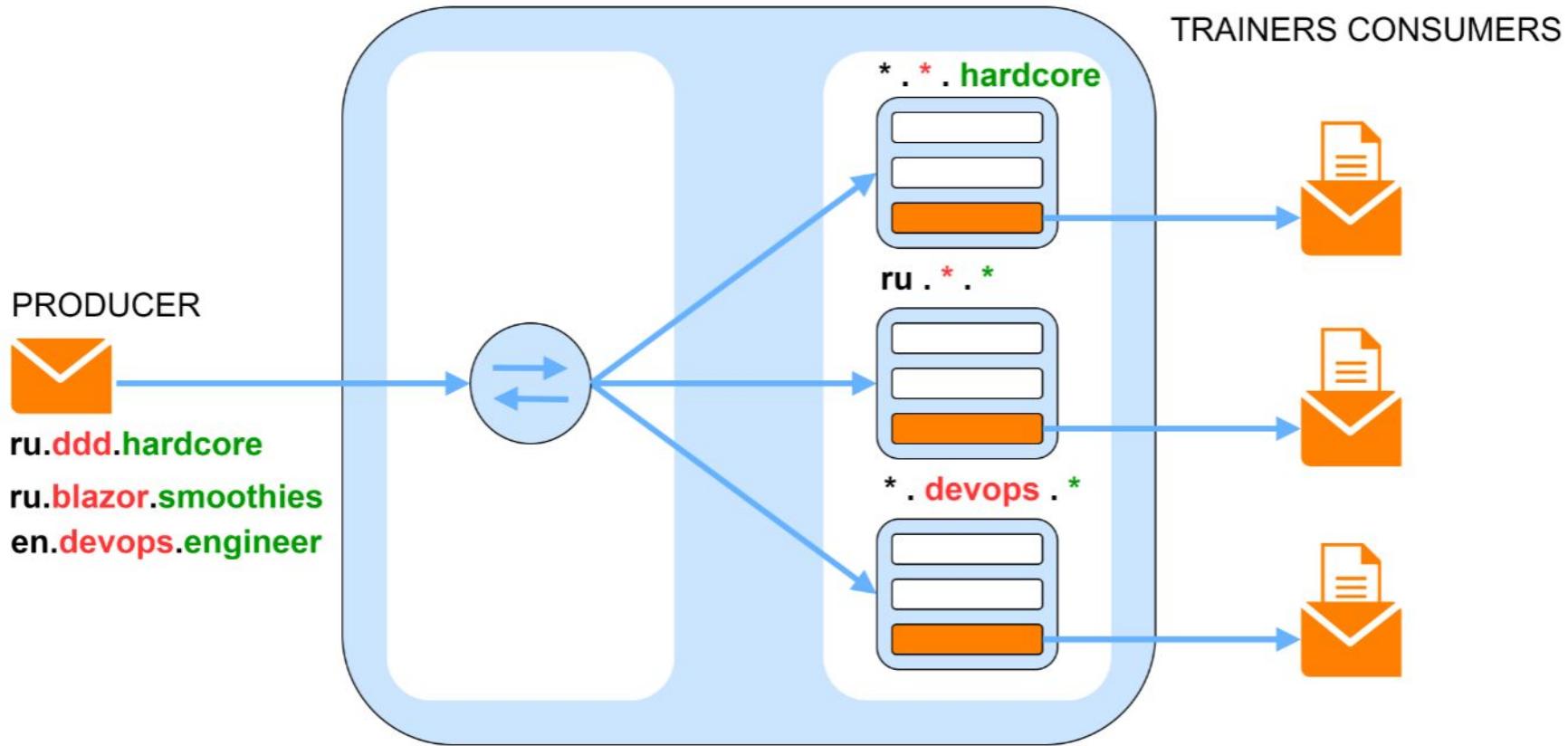


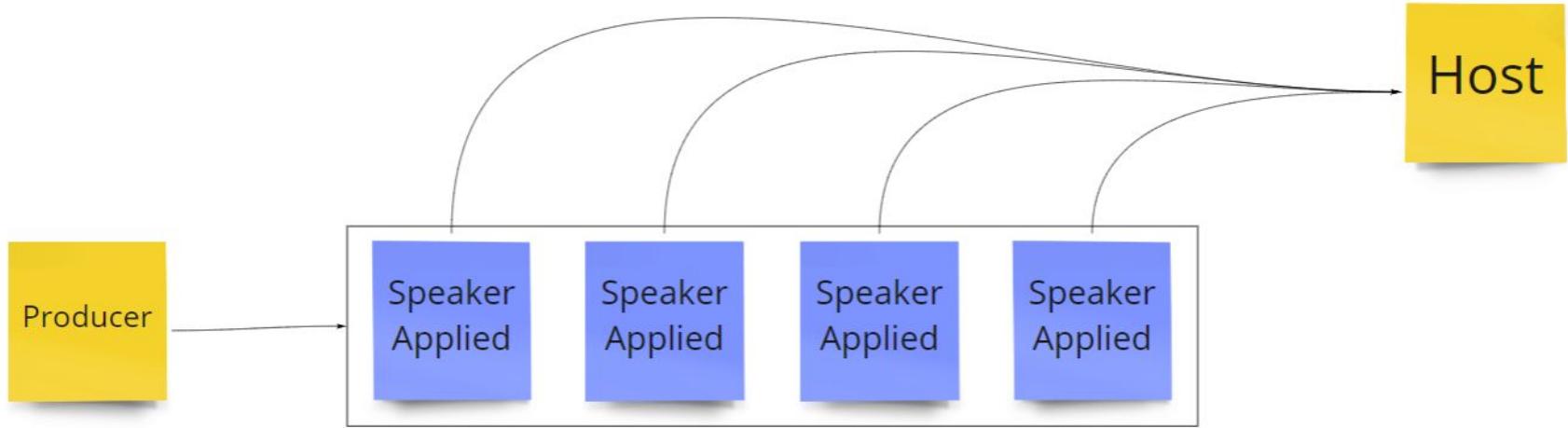
Publish/subscribe pattern

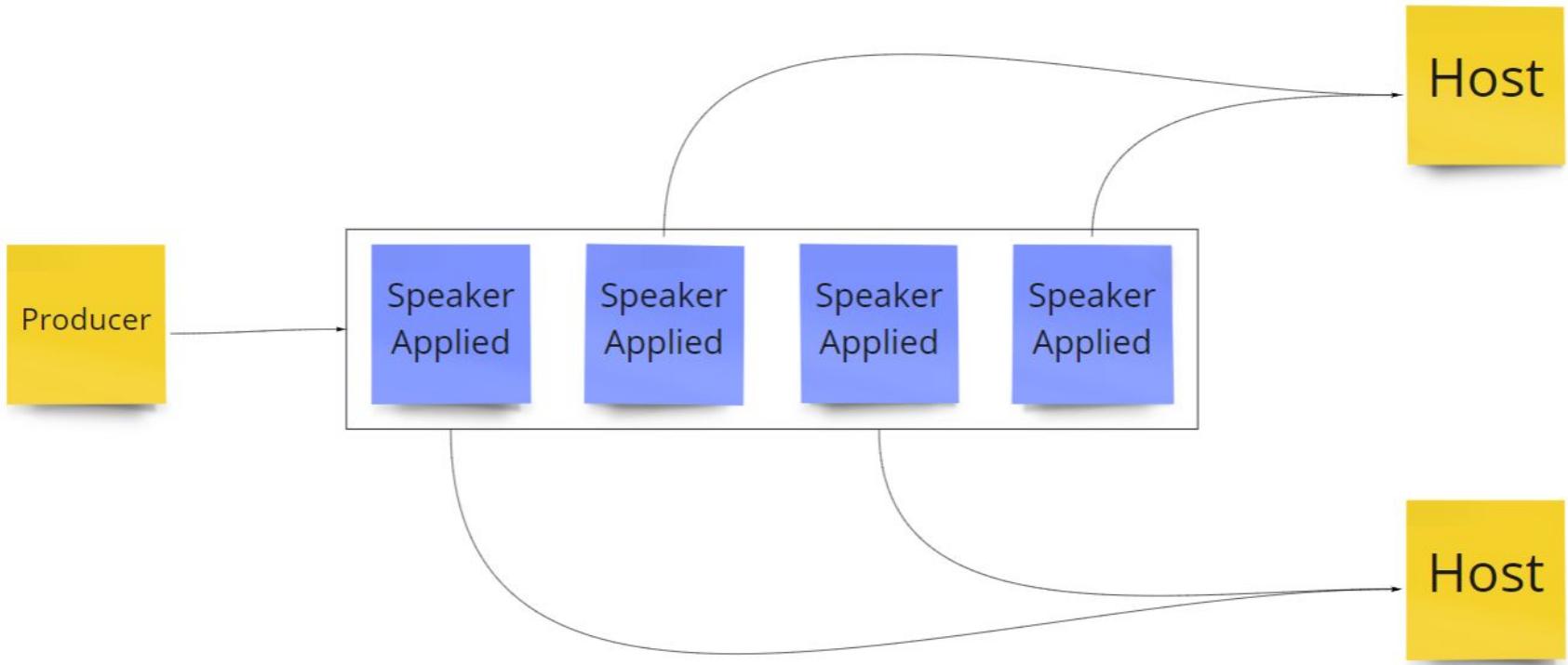




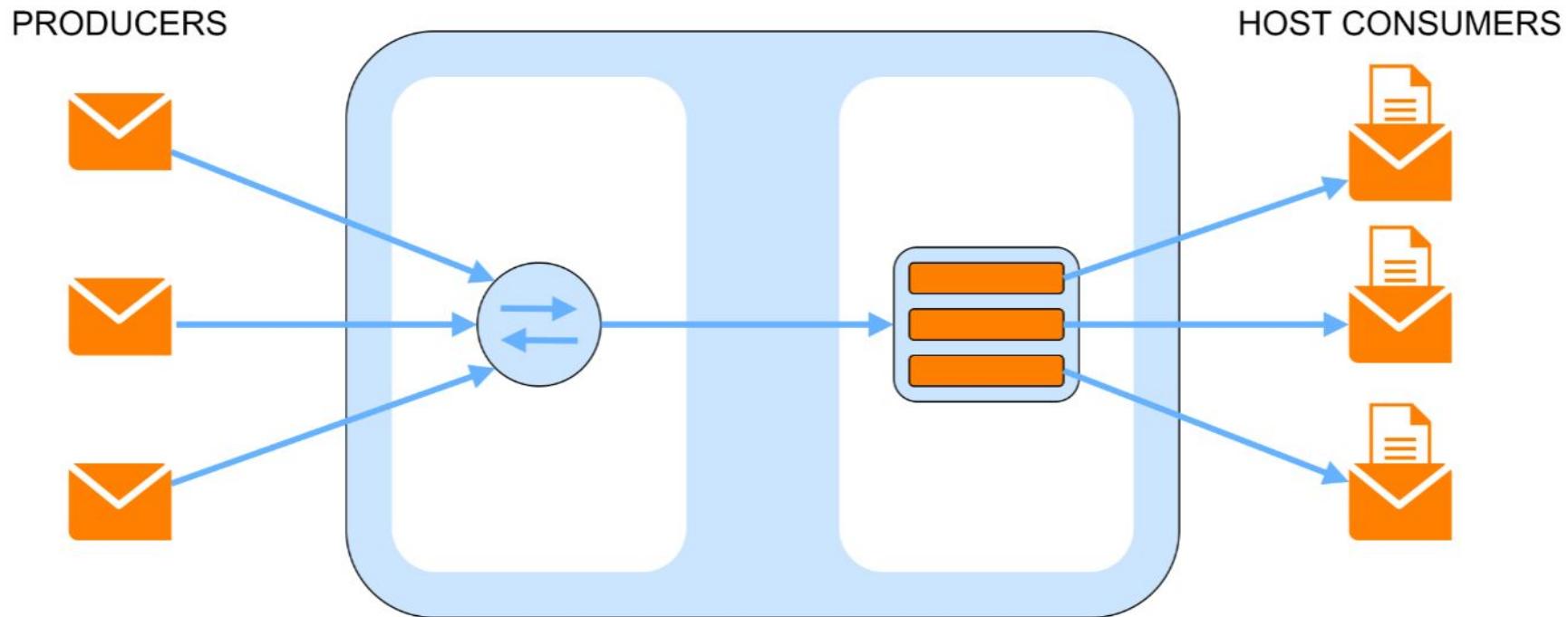
Topics pattern







Competing consumers

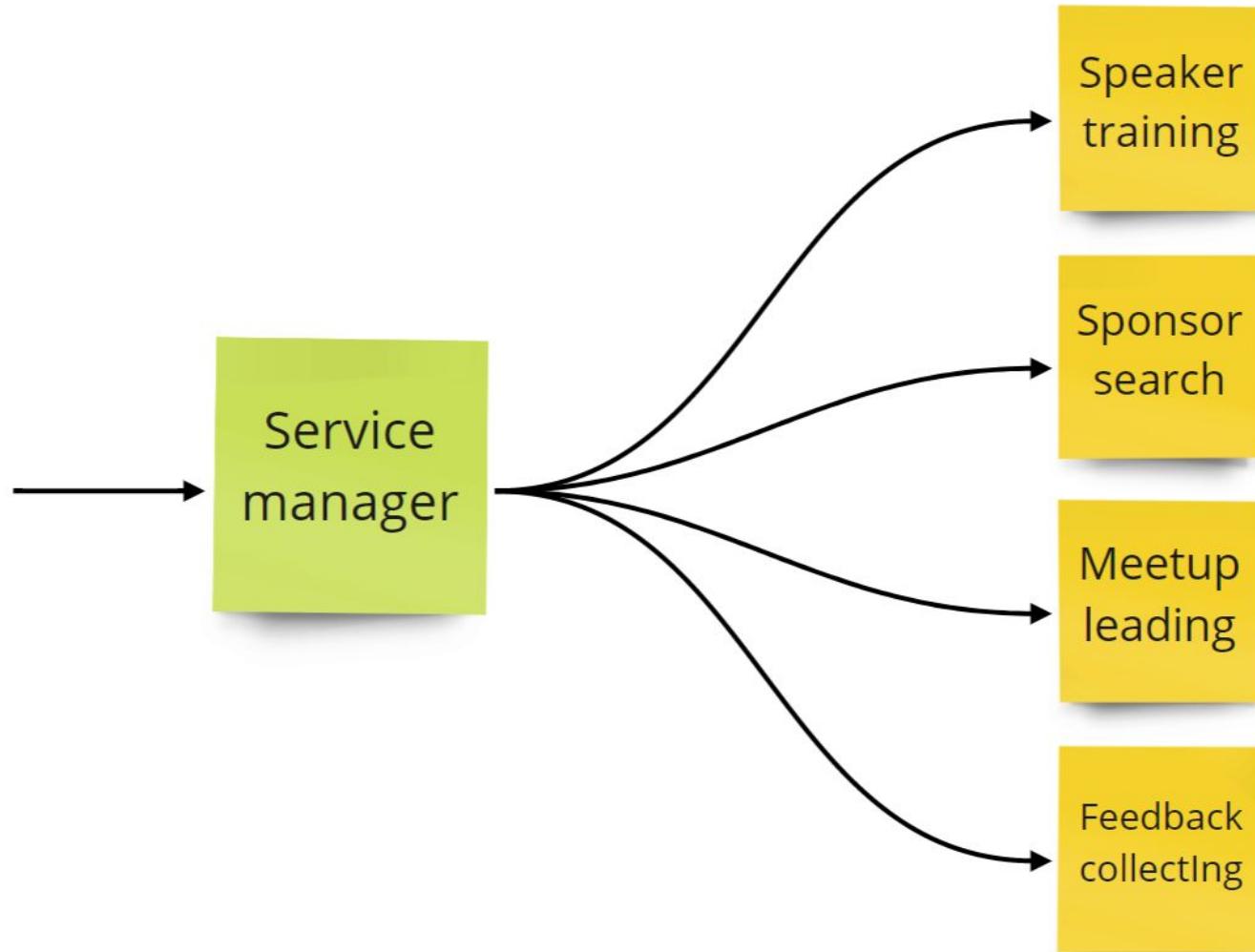


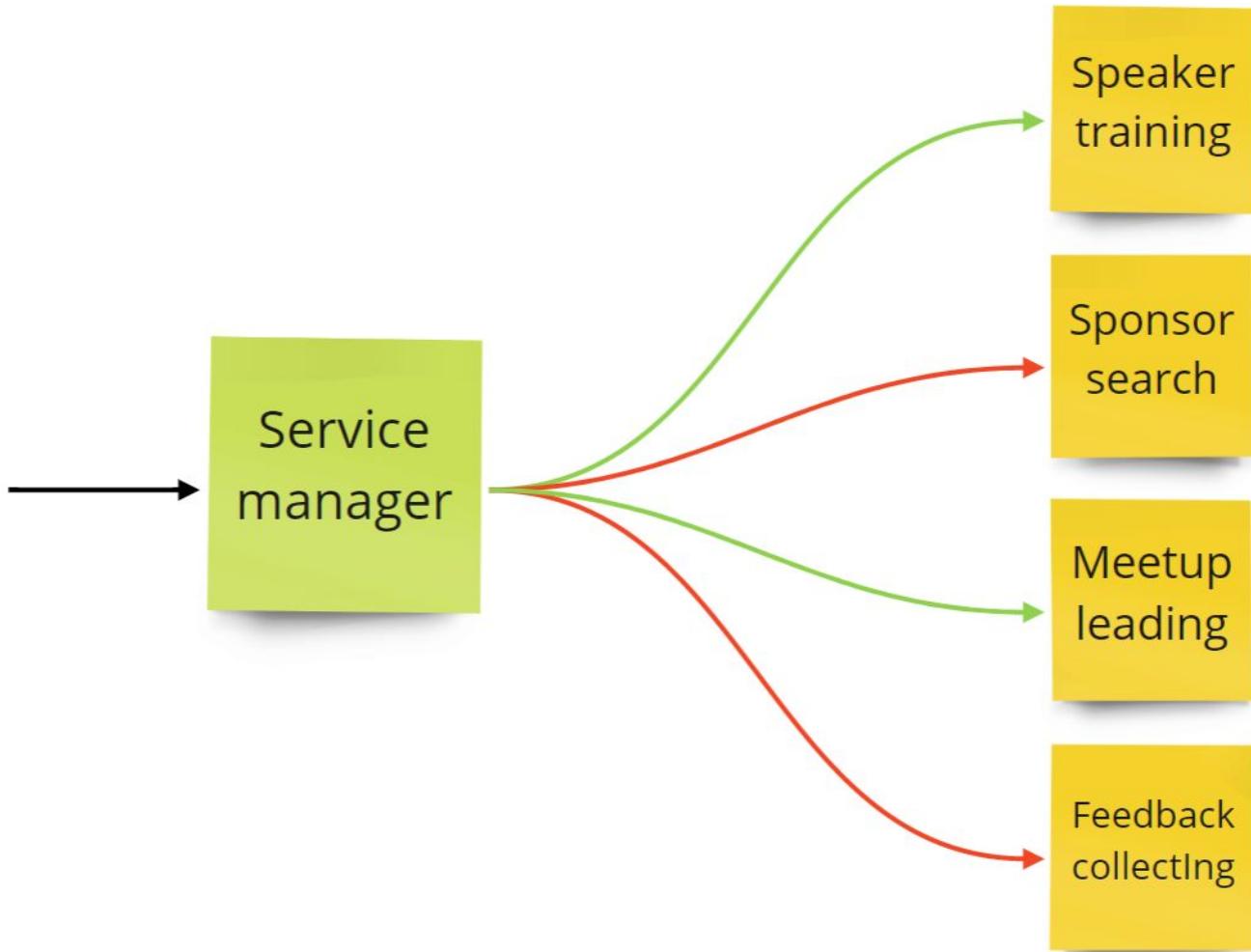
Why do we need messaging?

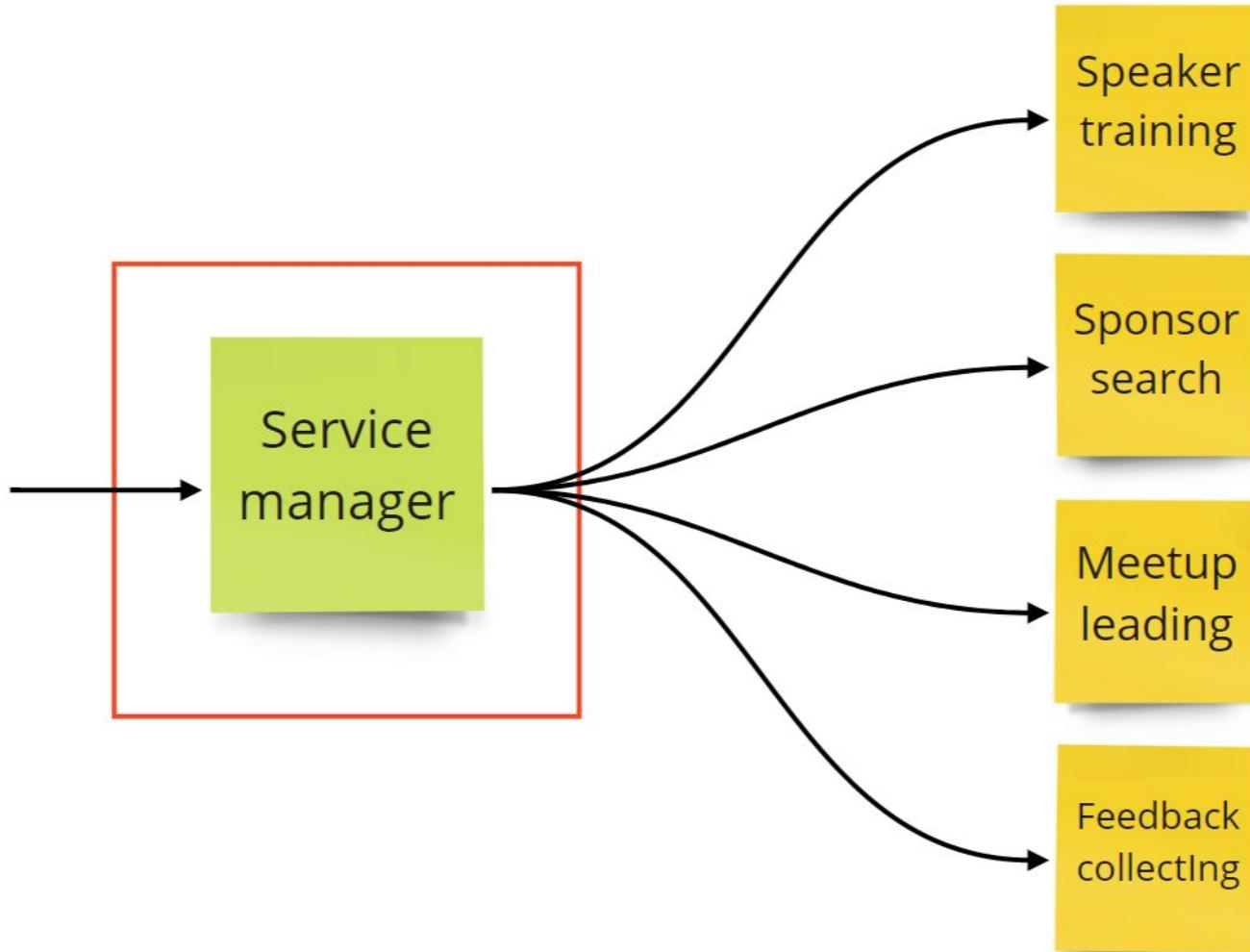
- Decoupling

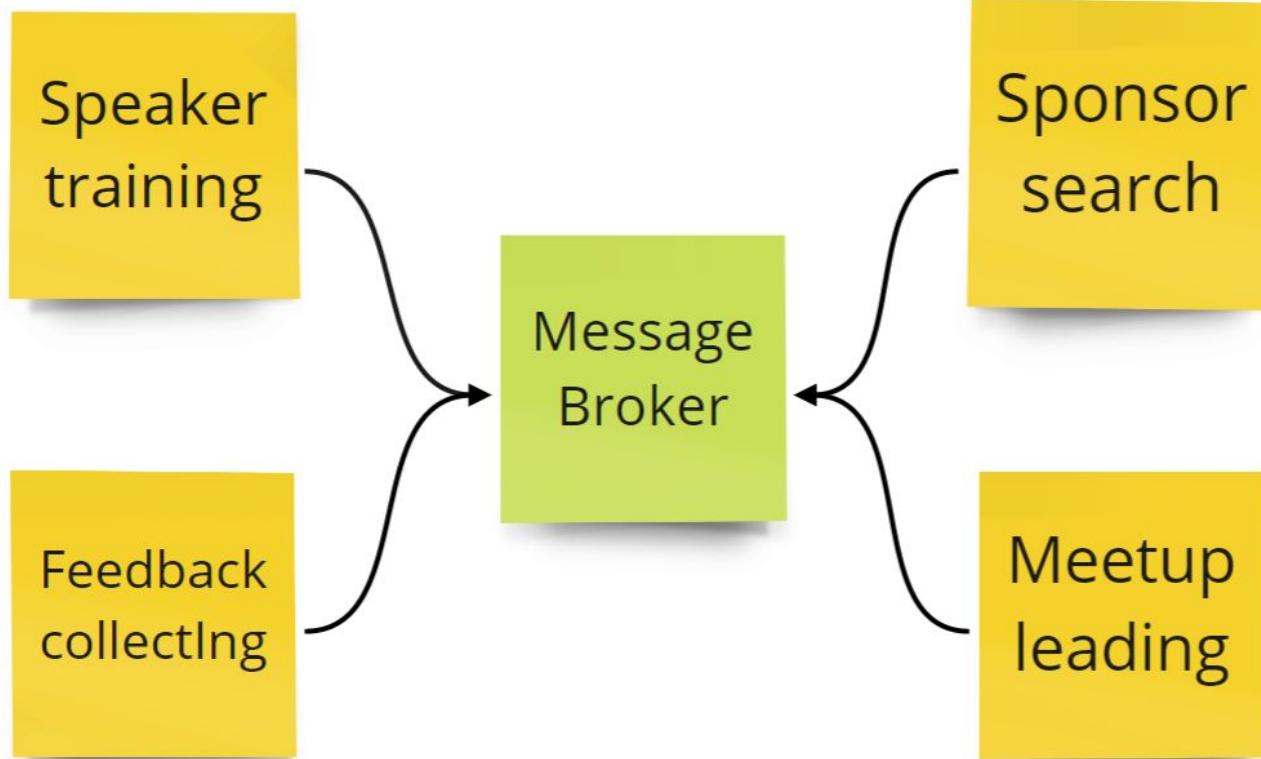
What are the
benefits?

Async vs Sync









Async vs Sync

- Async nature of messaging

Async vs Sync

- Async nature of messaging
- RPC

Async vs Sync

- Async nature of messaging
- RPC
- Scatter-Gather

Async vs Sync

- Async nature of messaging
- RPC
- Scatter-Gather
- Maybe sync?

Why do we need messaging?

- Decoupling
- Async communications

Why do we need messaging?

- Decoupling
- Async communications
- Provider specific benefits

RabbitMQ

- Management UI
- Plug-ins
- Great docs
- Widely supported
- Docker image

Azure Service Bus

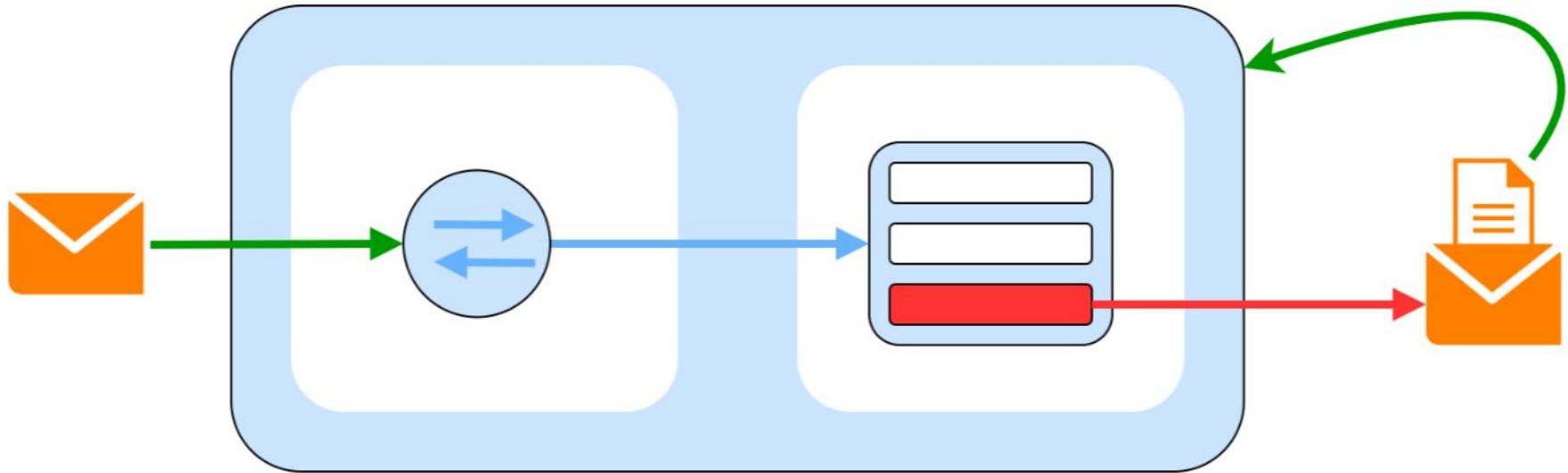
- PaaS
- Monitoring
- Scalability
- Sessions
- Duplicates
- Scheduling

Why do we need messaging?

- Decoupling
- Async communications
- Provider specific benefits
- Serverless apps

Why do we need messaging?

- Decoupling
- Async communications
- Provider specific benefits
- Serverless apps
- **Reliability**



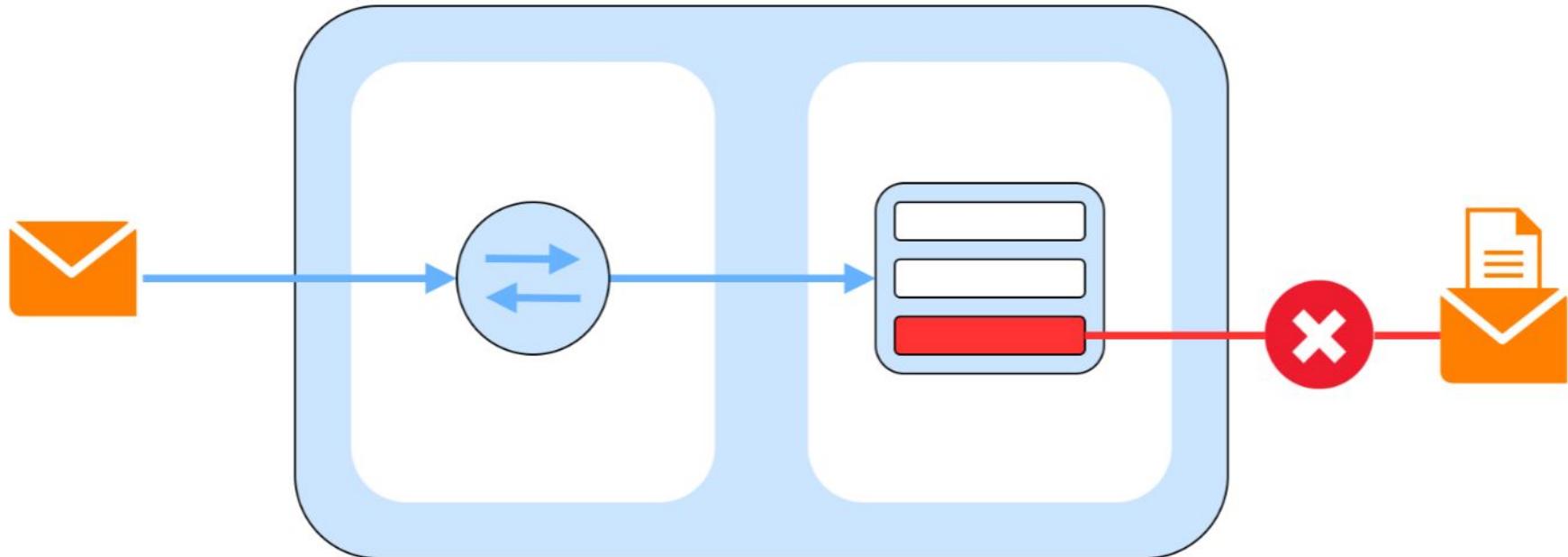
Publisher confirms

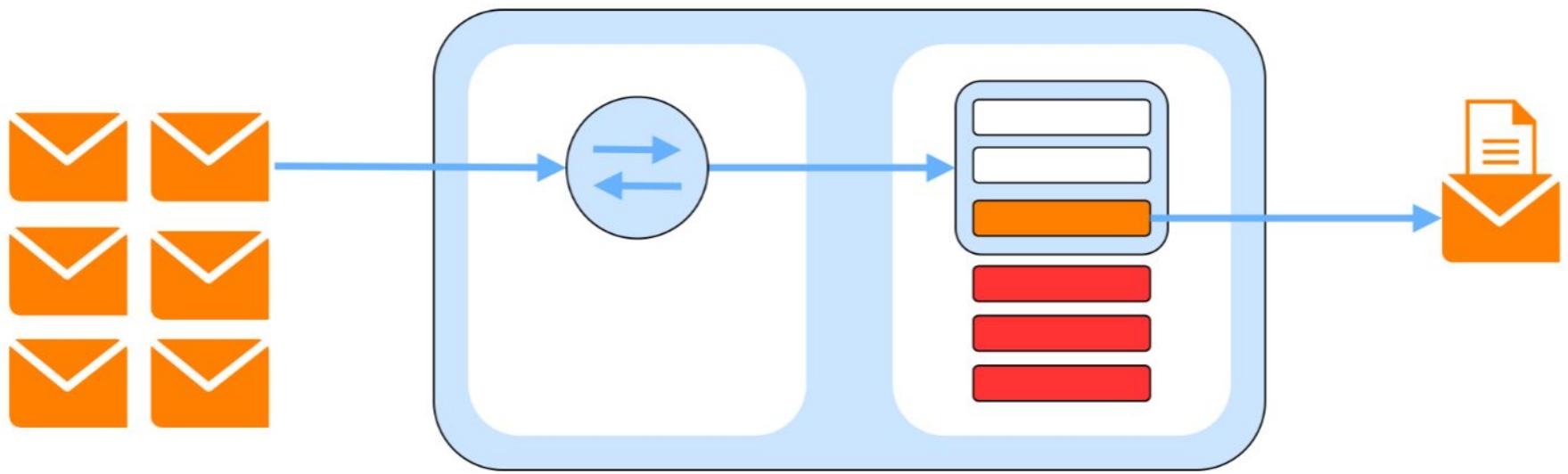
- Broker sends ack/nack
- Routable messages mapped
- Unroutable messages confirmed

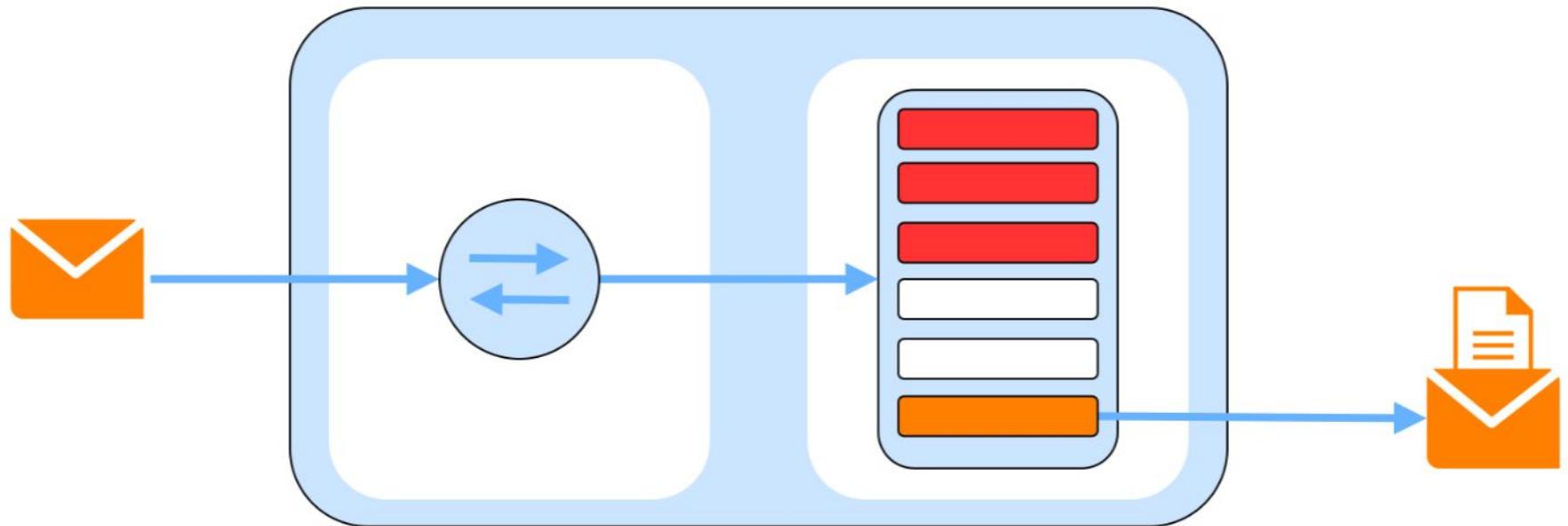
Consumer acknowledgement

- Consumers sends ack/nack
- Manual and automatic modes
- Auto-requeue when connection failed

Could something go
wrong?

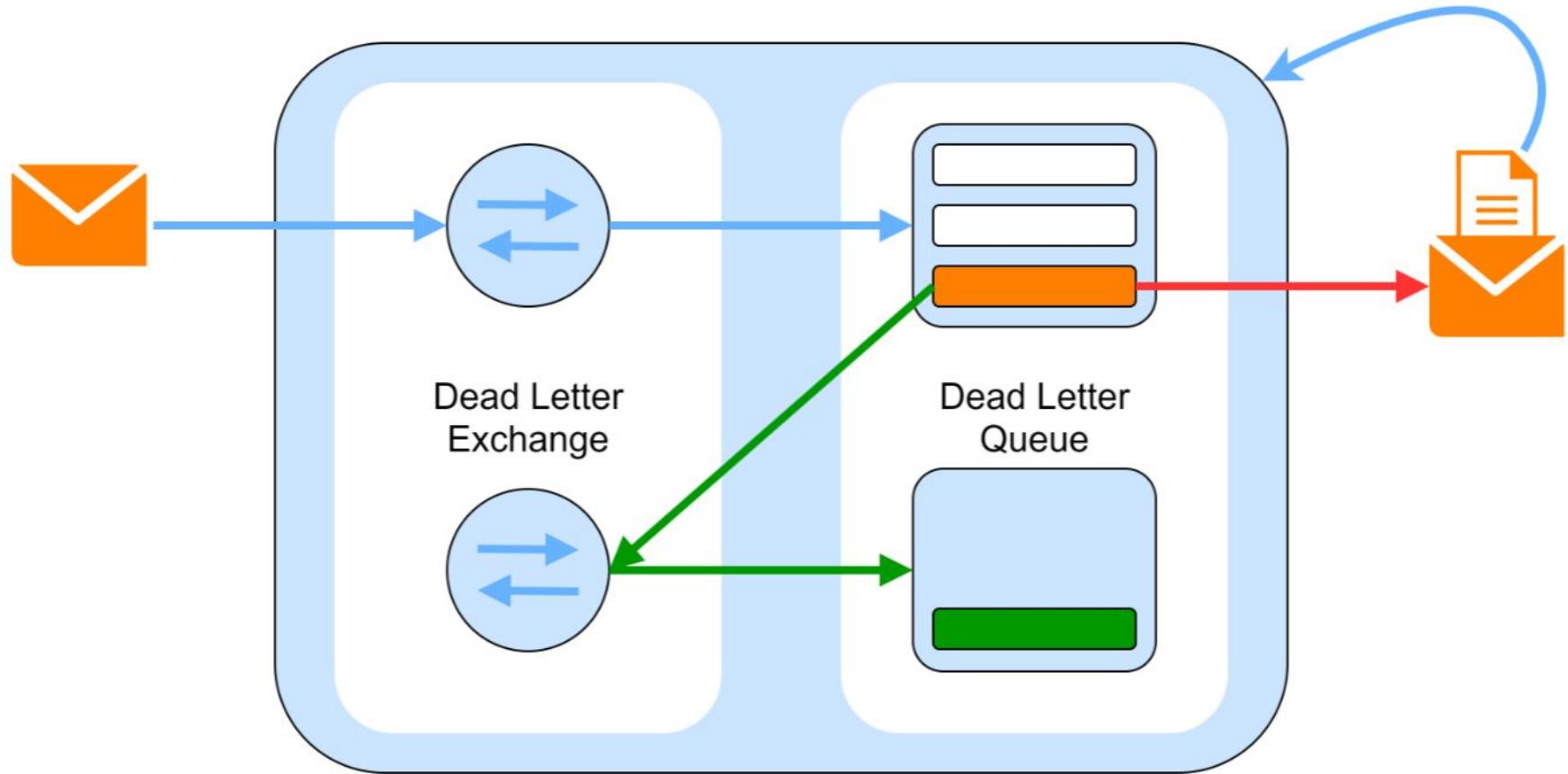






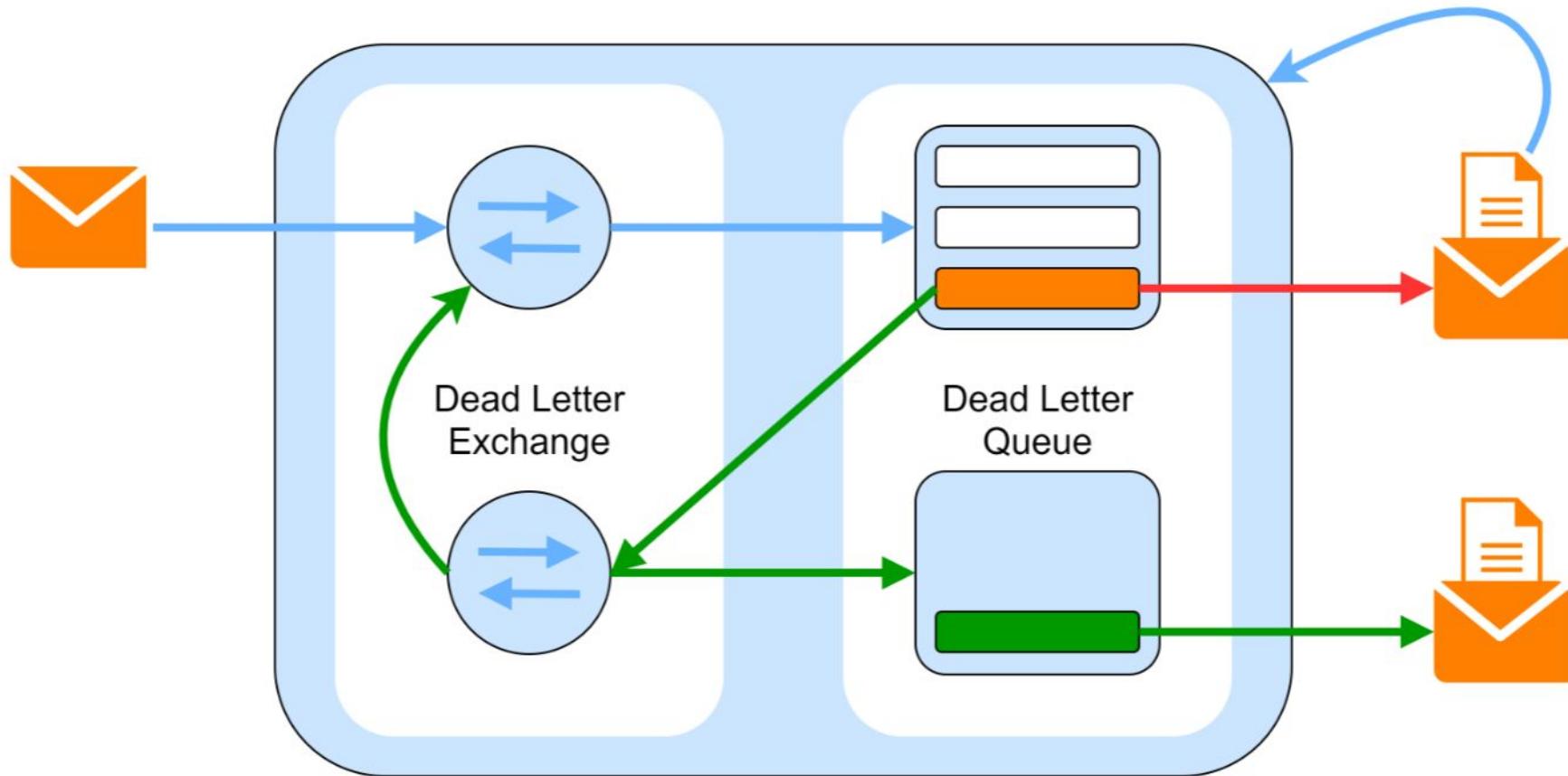
Reasons

- Message rejected with no requeue
- Queue size exceeded
- Message expired due to TTL



x-death

- Origin queue
- Reason
- Dead Letter Exchange
- Origin routing key



Detect the problem

RabbitMQ

- [prometheus](#)
- [health checks](#)
- [alerts](#)

Azure Service Bus

- [monitor](#)
- [service health](#)
- [alerts](#)

.net API

Reasons

- API

```
// Native RabbitMQ publish code
var factory = new ConnectionFactory {HostName = "localhost"};
using var conn = factory.CreateConnection();
using var channel = conn.CreateModel();

channel.QueueDeclare(QueueName, true, false, false, null);

var body = Encoding.UTF8.GetBytes(message);
channel.BasicPublish(string.Empty, QueueName, properties, body);

// EasyNetQ publish code
var bus = RabbitHutch.CreateBus("host=localhost");
bus.Publish(message);
```

Reasons

- API
- Advanced patterns

Reasons

- API
- Advanced patterns
- Unit tests tools

Reasons

- API
- Advanced patterns
- Unit tests tools
- Monitoring support

Reasons

- API
- Advanced patterns
- Unit tests tools
- Monitoring support
- Error handling strategy

Reasons

- API
- Advanced patterns and features
- Unit tests tools
- Monitoring support
- Error handling strategy
- **Transport Liquidity**

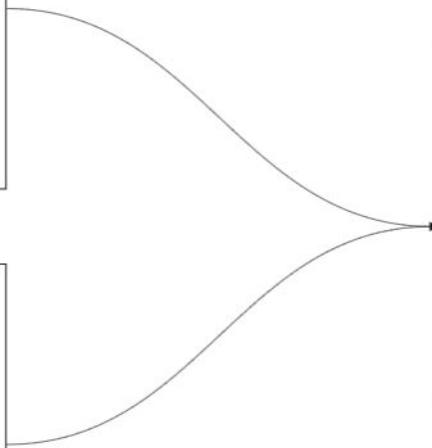
Development



Production



Message broker



.net API

- MassTransit
- NServiceBus
- EasyNetQ
- Rebus
- ...

Rebus

- Easy to configure
- Single queue per app
- Wide range of transports

Rebus Transport

- RabbitMQ
- MSMQ
- Azure Service Bus
- In Memory
- [MS SQL, PostgreSQL, MySQL](#)

Recipe

✓ Make sure you need it

Recipe

- ✓ Make sure you need it
- ✓ Choose your option

Recipe

- ✓ Make sure you need it
- ✓ Choose your option
- ✓ Implement error handling strategy

Recipe

- ✓ Make sure you need it
- ✓ Choose your option
- ✓ Implement error handling strategy
- ✓ Configure monitoring

Recipe

- ✓ Make sure you need it
- ✓ Choose your option
- ✓ Implement error handling strategy
- ✓ Configure monitoring
- ✓ Production checklist

Sources



Udi Dahan

[Distributed Systems Design Fundamentals](#)

[NServiceBus](#)

Sources



RabbitMQ

[Documentation](#)

[Get Started](#)

Sources



Jeffrey Richter

[Messaging with queue](#)

Sources

- [Enterprise messaging patterns](#)
- [ZeroMQ brokerless model](#)
- [Rebus](#)
- [Samples](#)



Daniil Doronkin

Messaging in practice

Thank you for attention!