KubeCon CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024


KubeCon
CloudNativeCon
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
AI_dev
Open Source GenAI & ML Summit
China 2024
xRegistry – Looking Beyond CloudEvents
Leo Li - Red Hat

👋 Hello!

Leo Li

Software Engineering Intern @ Red Hat

Knative Eventing Maintainer

@Leo6Leo

Agenda

CloudEvents(**CE**): Intro & Community updates

1. Understand CE: analogy of delivery package
2. Updates from the community
3. What is the next step/direction for CloudEvents?

xRegistry: Event Discovery

1. Understand current problems: Checkpoints analogy
2. How xRegistry resolves the current problem?
3. xRegistry work in practice
4. xRegistry CLI

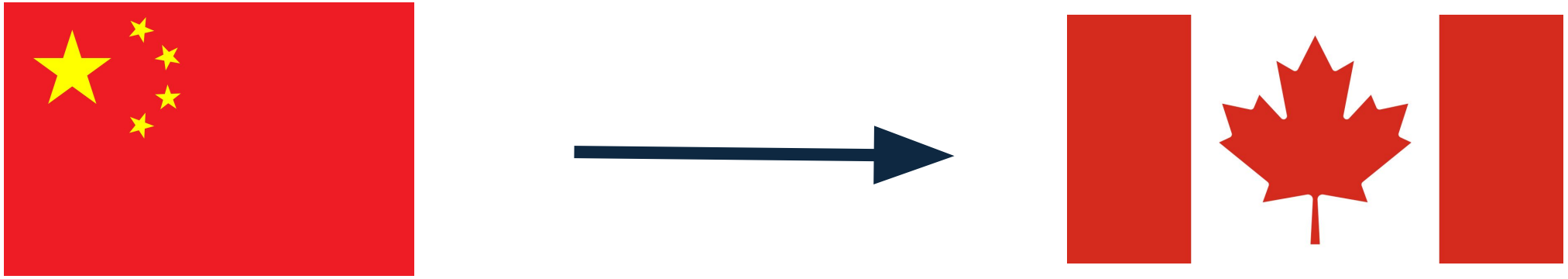
Q & A

CloudEvents

A specification for **describing event metadata** in a common way

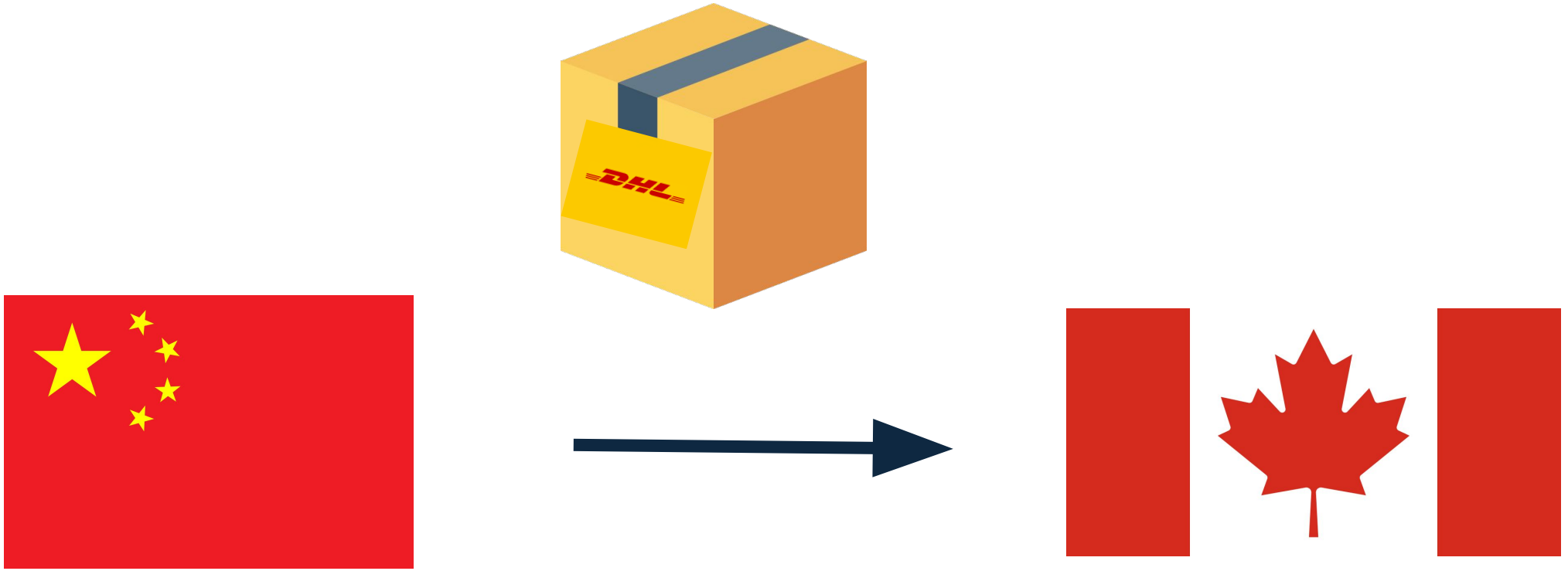
CloudEvents Example

Within each country, there are different courier services

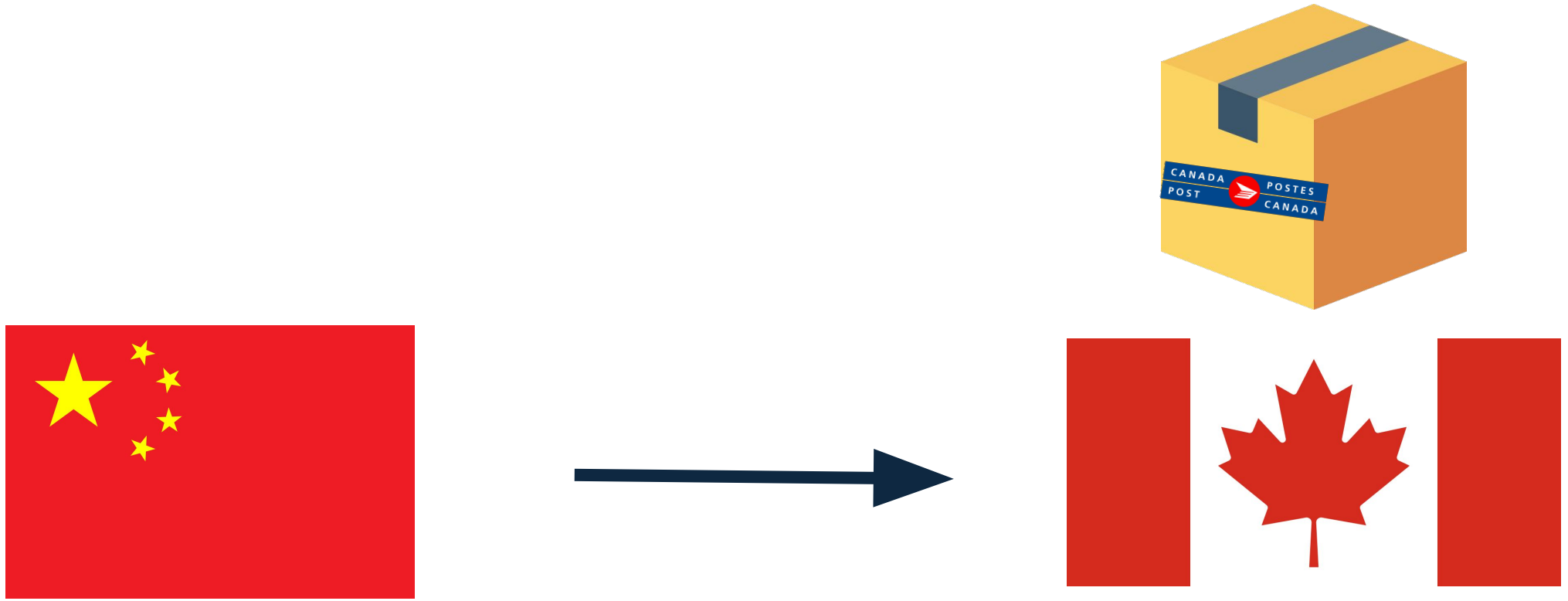


Send a package from China to Canada

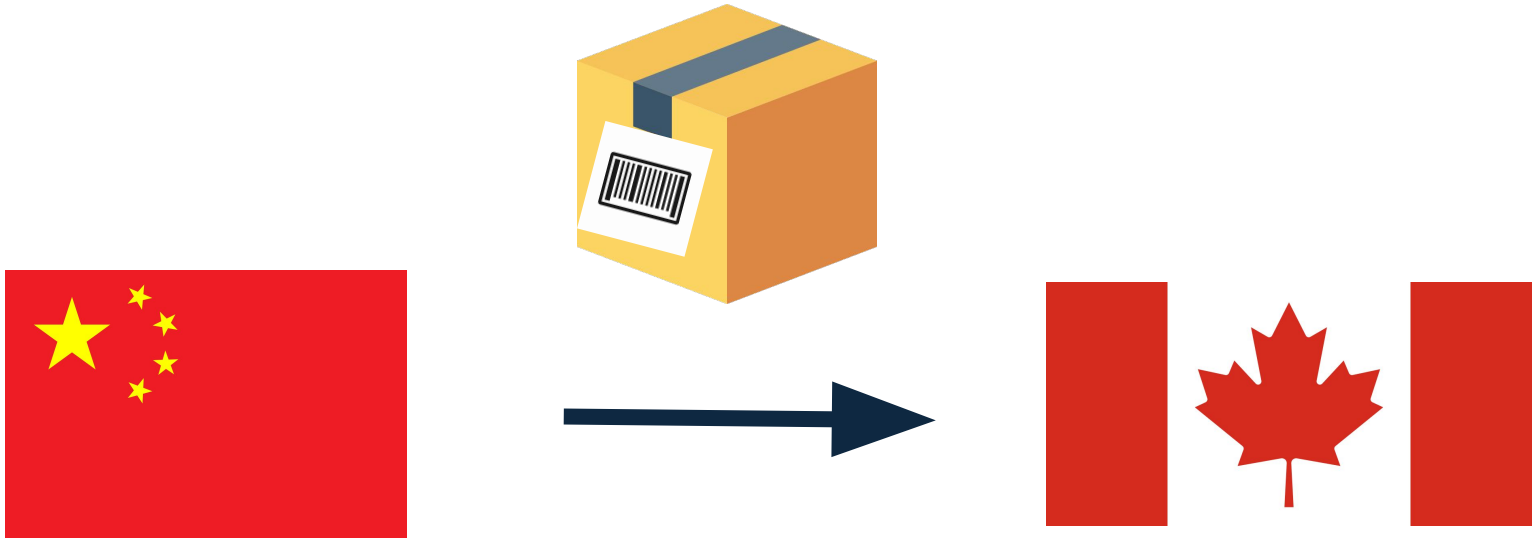
Within China, the package has the **Chinese** Courier label



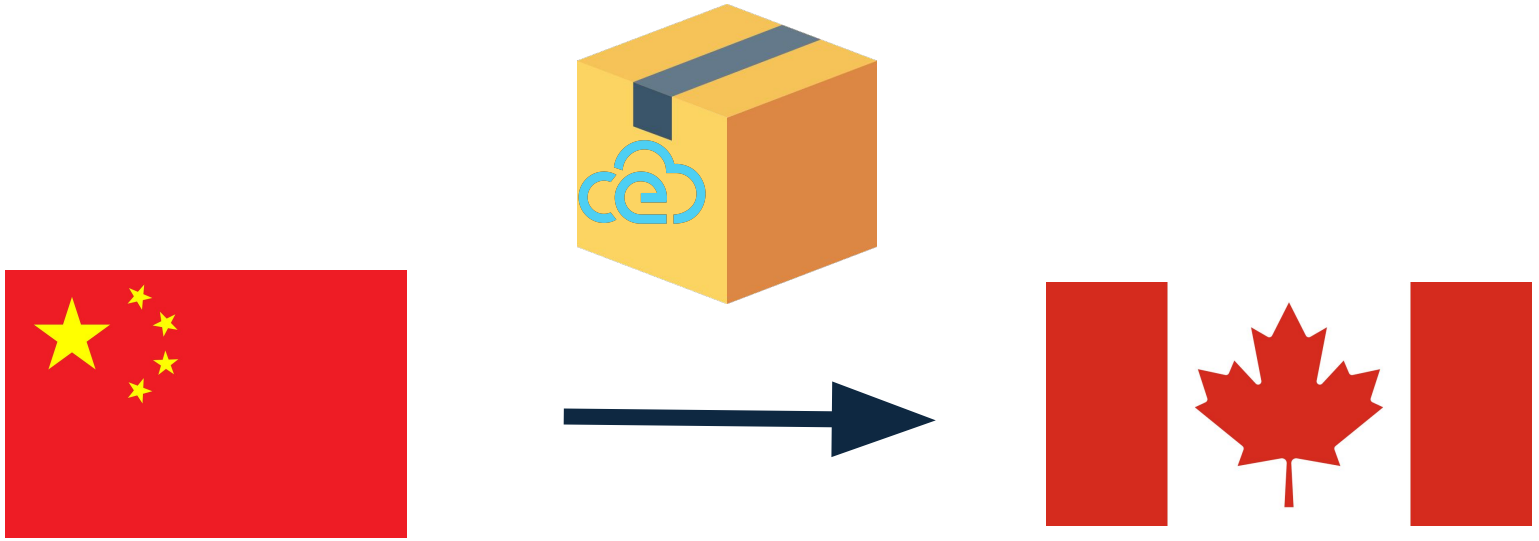
When leaving China, the package will have **international courier label**



When arriving Canada, the package will have the **Canadian Courier label**



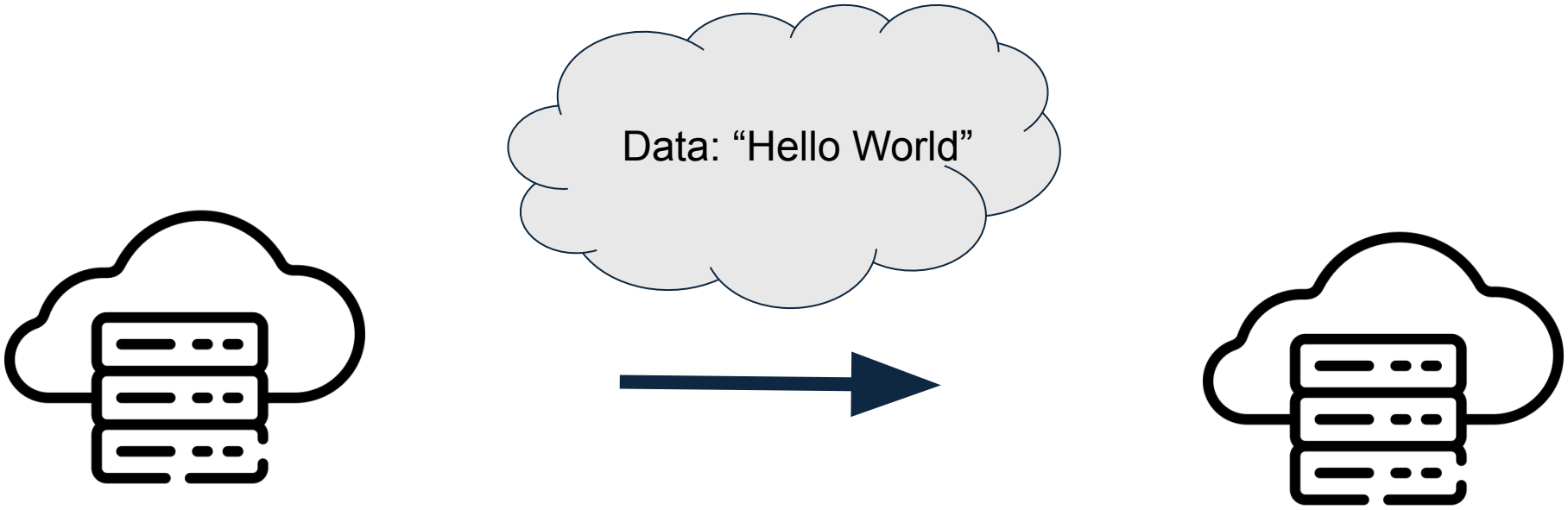
If there is a **universal label** that all couriers can understand,
No need to change label every time.



That’s why we name CloudEvents as that

magic universal label at the beginning.

CloudEvents Example

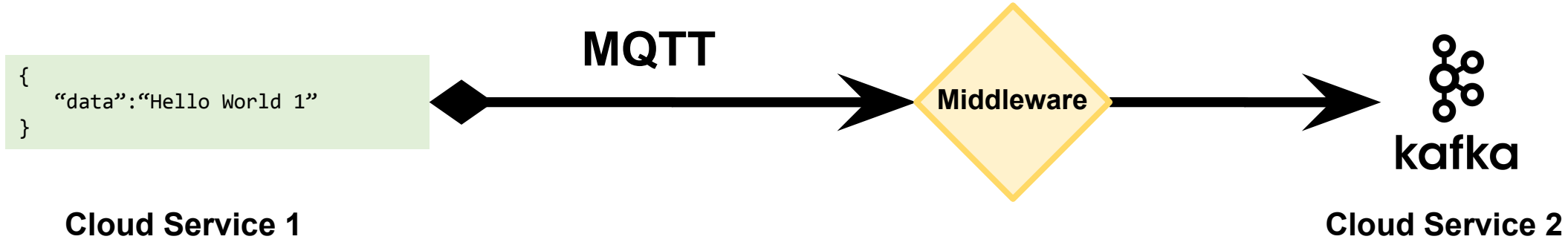


Similarly, events move **between cloud services**

Events Are Everywhere

Different Cloud Services may have protocols

Events Are Everywhere



Events Are Everywhere

```
{
   “data”:“I love HK!”
}
```

MQTT

Cloud Service 1

```
{
   “data”:“I’m at KubeCon China!”
}
```

HTTP

Cloud Service 2

Cloud Service 3

```
{
   “data”:“Hello World!”
}
```

NATS

Varying:
Formats
Schemas
Business Logic

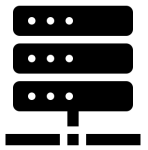
Middleware

Smart middleware (selective routing) is costly

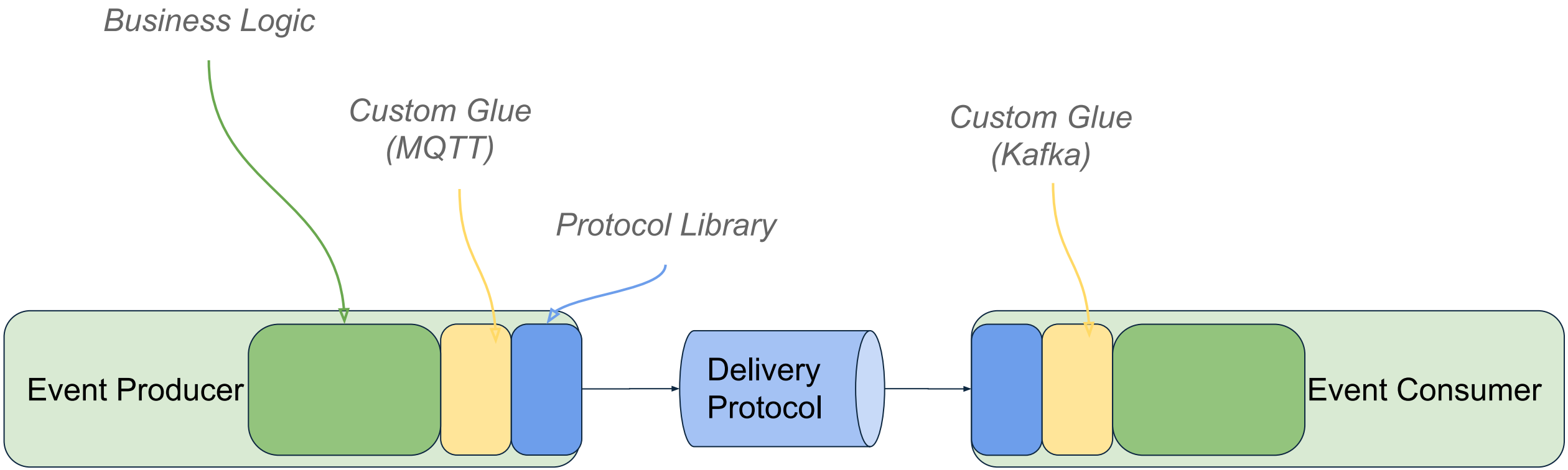
kafka

Cloud Service 4

Cloud Service 5

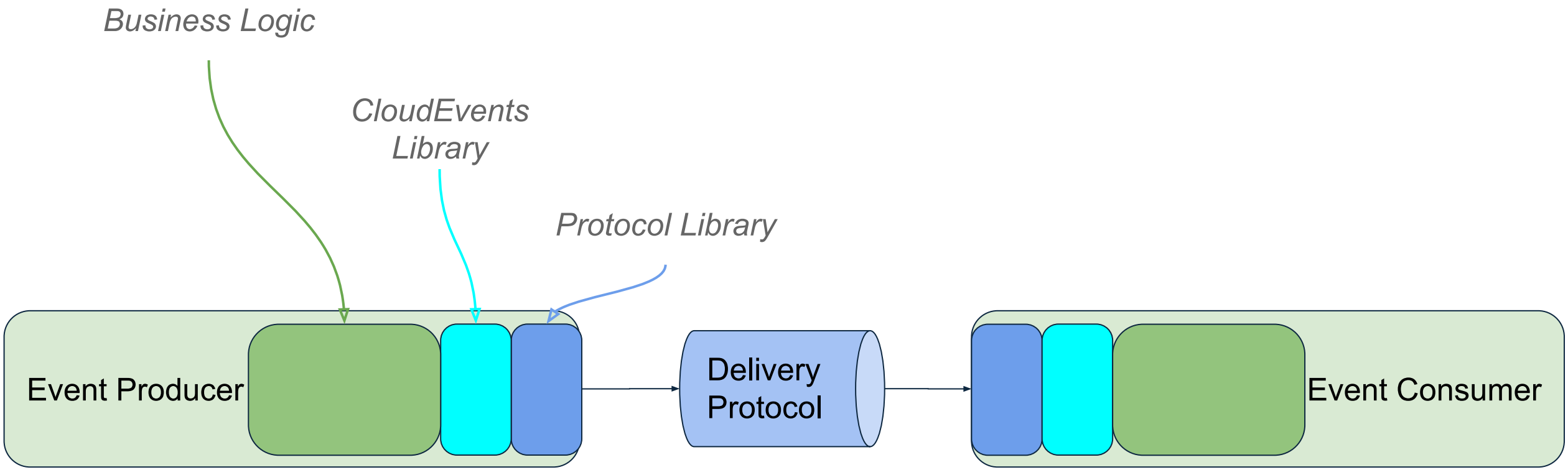


Before CloudEvents



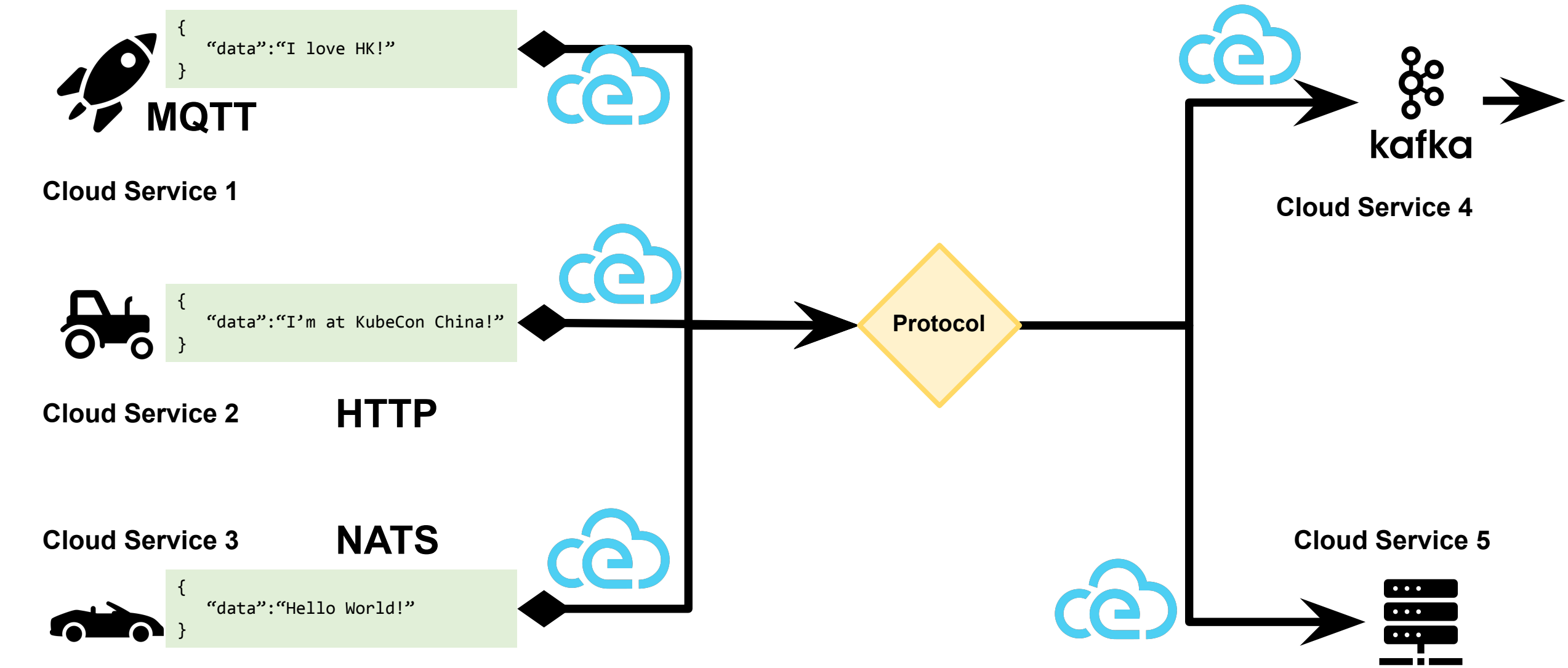
Source: Thinking Cloud Native, CloudEvents Future - Scott Nichols, Chainguard, Inc. - KubeCon EU 2022

After CloudEvents



Source: Thinking Cloud Native, CloudEvents Future - Scott Nichols, Chainguard, Inc. - KubeCon EU 2022

Events Are Everywhere

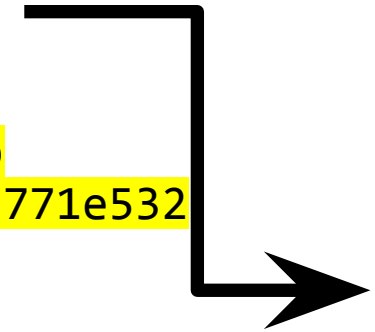


Source with some modification: CloudEvents - Don't call us, we'll call you- Klaus Deissner, SAP - KubeCon EU 2024

CloudEvents in Action

HTTP - Binary

```
POST /events HTTP/1.0
Host: example.com
Content-Type: application/json
ce-specversion: 1.0
ce-type: com.bigco.newItem
ce-source: http://bigco.com/repo
ce-id: 610b6dd4-c85d-417b-b58f-3771e532

{
  "action": "newItem",
  "itemID": "93"
}
```



HTTP - Structured

```
POST /events HTTP/1.0
Host: example.com
Content-Type: application/cloudevents+json

{
  "specversion": "1.0",
  "type": "com.bigco.newItem",
  "source": "http://bigco.com/repo",
  "id": "610b6dd4-c85d-417b-b58f-3771e532",
  "datacontenttype": "application/json",
  "data": {
    "action": "newItem",
    "itemID": "93"
  }
}
```

Source with some highlight: CloudEvents - Don’t call us, we’ll call you- Klaus Deissner, SAP - KubeCon EU 2024

CloudEvents - Binary

When events are flowing between services

HTTP - Binary

```
POST /events HTTP/1.0
Host: example.com
Content-Type: application/json
ce-specversion: 1.0
ce-type: com.bigco.newItem
ce-source: http://bigco.com/repo
ce-id: 610b6dd4-c85d-417b-b58f-3771e532

{
  "action": "newItem",
  "itemID": "93"
}
```

Metadata aids with filtering, to deliver the event correctly

Event payload is the event's content

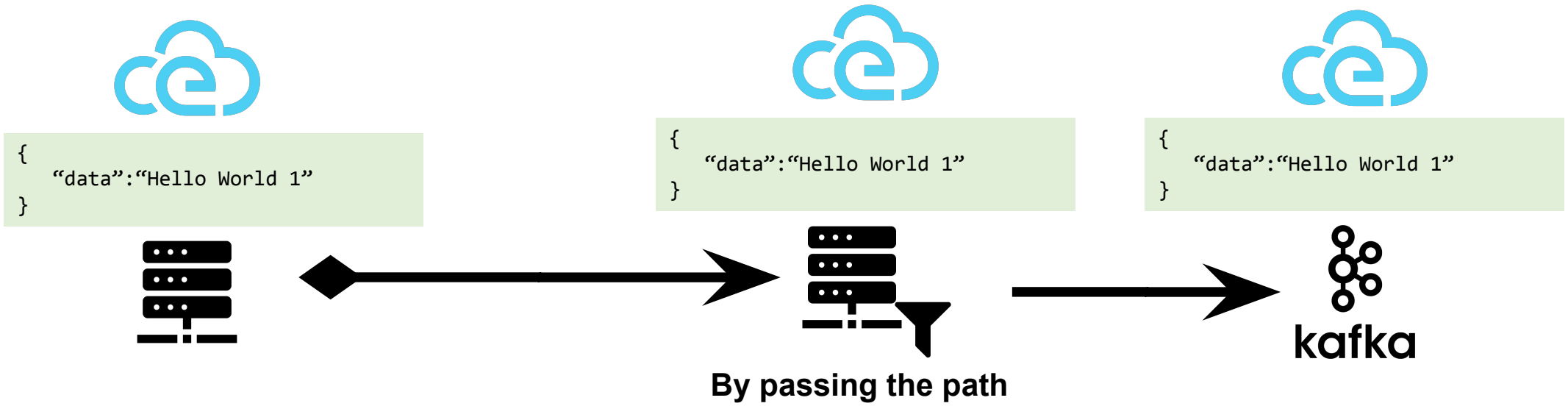
When events are flowing between services

HTTP - Binary

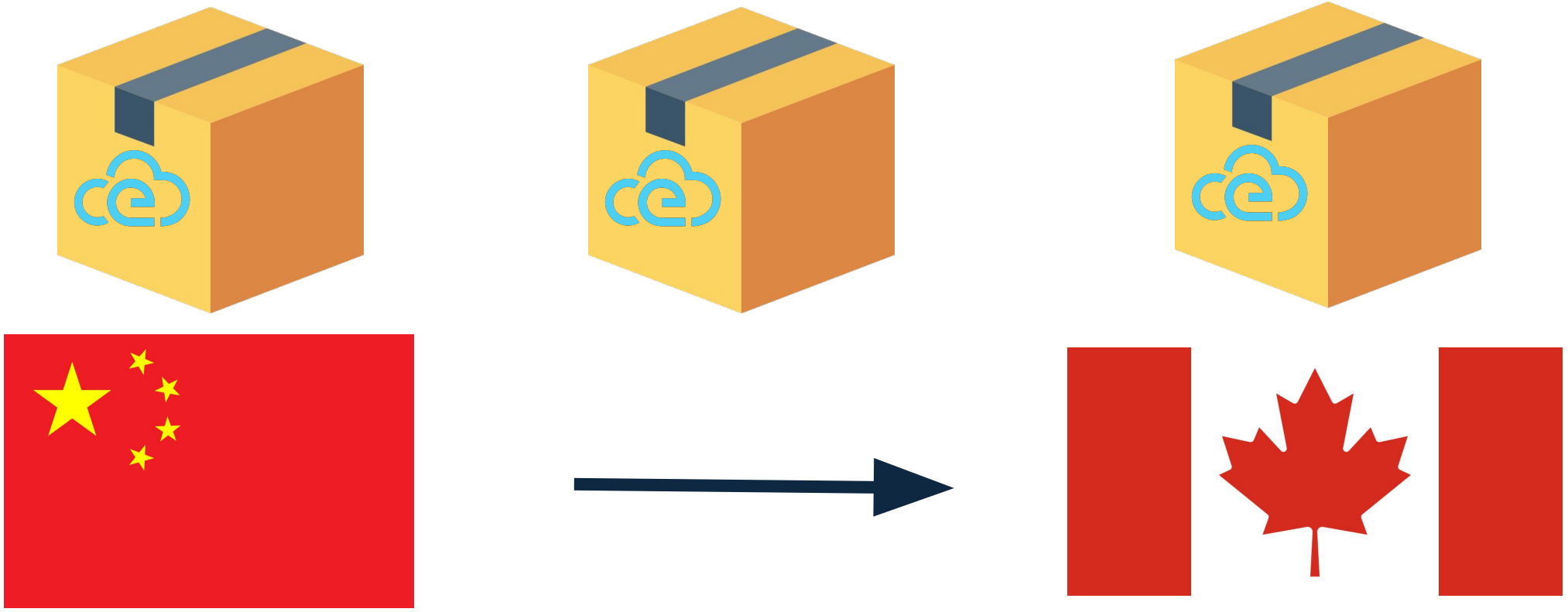
```
POST /events HTTP/1.0
Host: example.com
Content-Type: application/json
ce-specversion: 1.0
ce-type: com.bigco.newItem
ce-source: http://bigco.com/repo
ce-id: 610b6dd4-c85d-417b-b58f-3771e532

{
  "data":“Hello World 1”
}
```

Metadata aids with filtering, to deliver the event correctly

Services along the way **don’t need** to understand the payload to pass the events along.



CloudEvents



CloudEvents is that magic universal label

CloudEvents Graduated!

Started
Late 2017

Graduated
January 25th, 2024

Learn more at cloudevent.io

Community Updates

July 15th, 2024

CESQL V1.0 spec Released!

Along with GoLang & Java SDK

Filtering CloudEvents: an SQL-like language that provides a simple and consistent way for middleware and event processors to interact with CloudEvents.

https://bit.ly/**cesql-v1**

Community Updates

Feb. 9th, 2024

AWS EventBridge now supports CloudEvents

https://bit.ly/**awscloudevents**

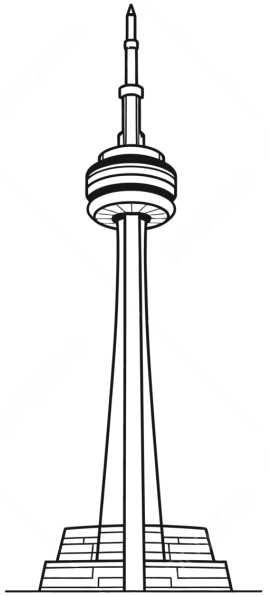
Community Updates

At **KubeCon CloudNativeCon 2019 in San Diego**

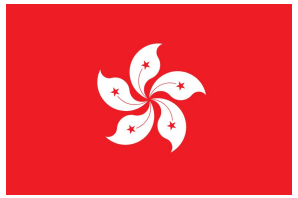
⇒ Decision to **work on event discovery and subscriptions next**

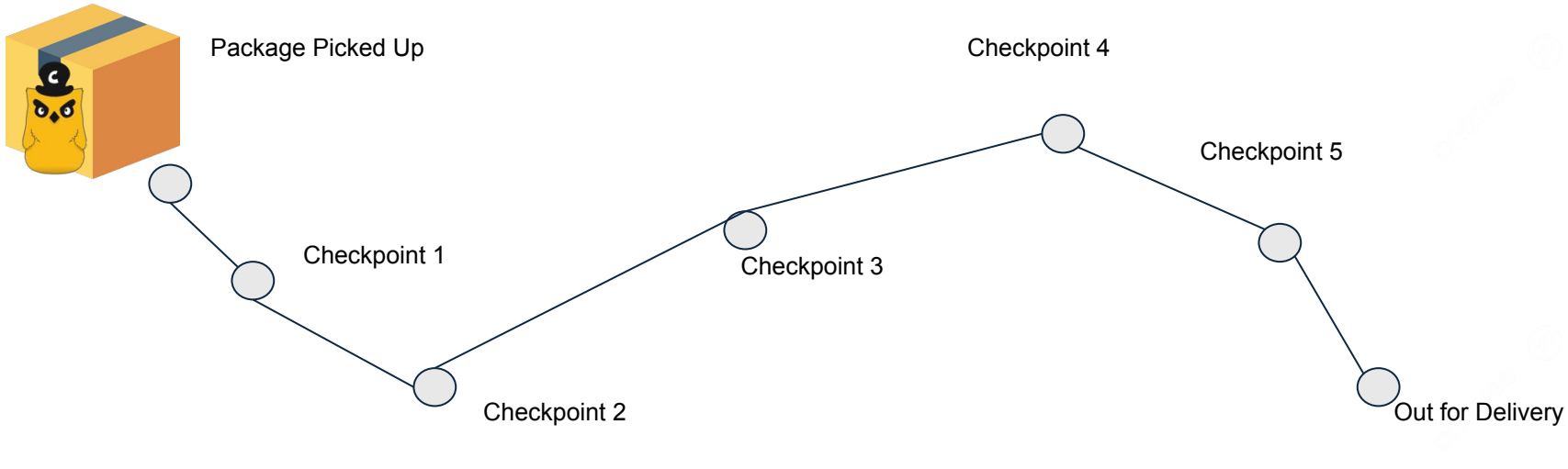
XREGISTRY

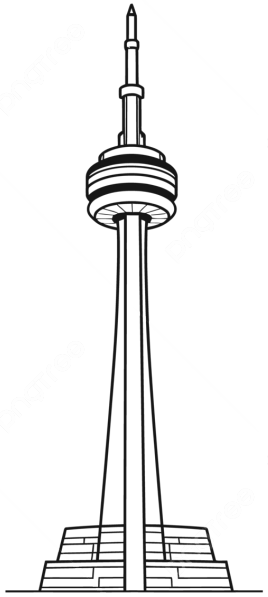
1. Understand current problems: Package and Checkpoints analogy
2. How xRegistry resolves the current problem?
3. xRegistry work in practice
4. xRegistry CLI



Toronto store submitted an order to the factory in HK







Packages need to pass different checkpoints/scan points

When package get Picked up, out for delivery, at checkpoint, notification will be sent

Delivery by Amazon

Tracking ID: ·

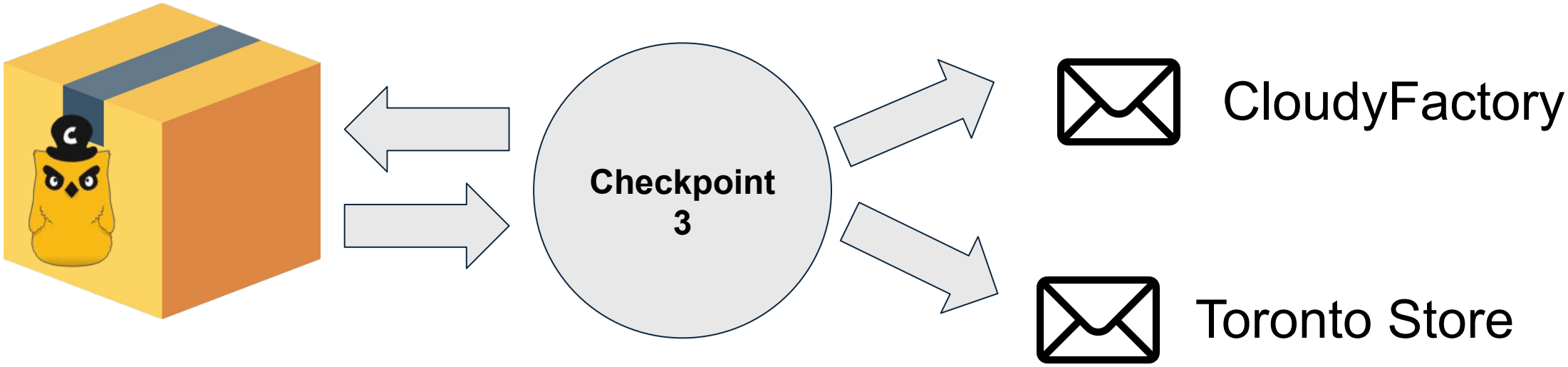
Tuesday, 30 July

5:17 PM	Package delivered near the front door or porch. *CA*
9:54 AM	Package is out for delivery. *Etobicoke, CA*
4:13 AM	Package being processed at carrier facility. *Etobicoke, CA*
3:19 AM	Package arrived at a carrier facility. *Etobicoke, CA*
	Carrier picked up the package.

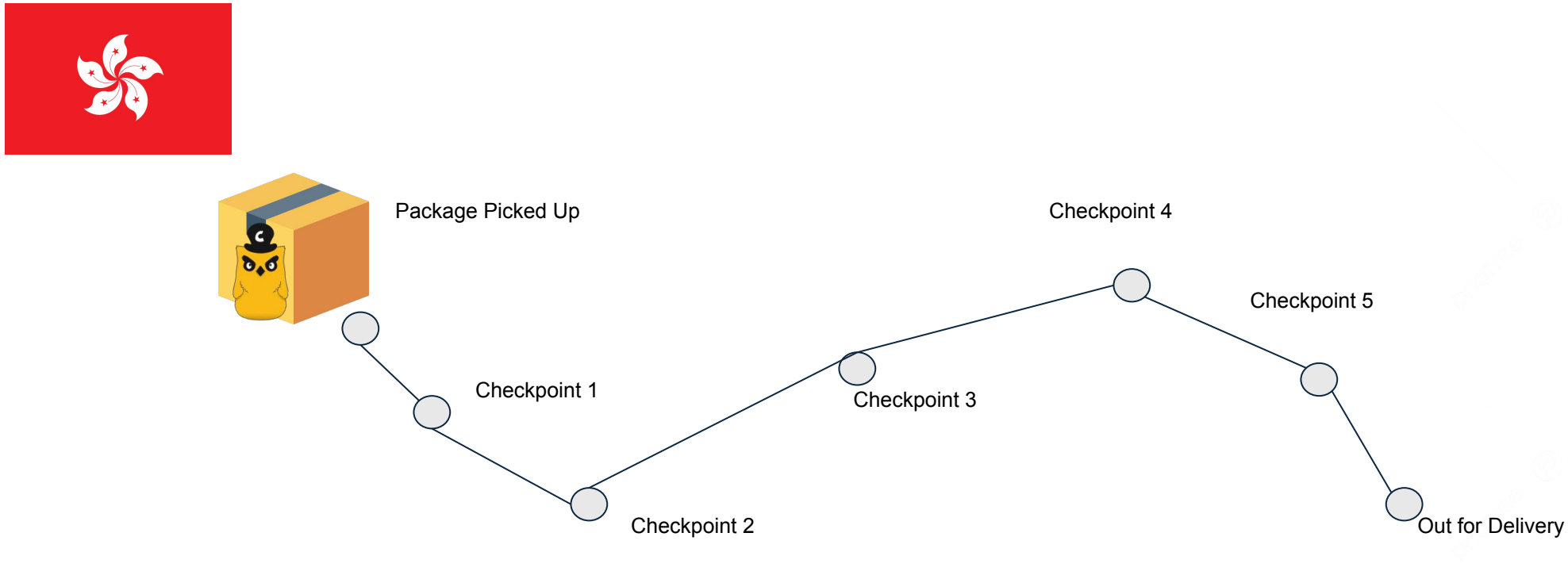
×

顺丰速运

- **已签收** 05–11 17:22
  派送成功
- **派送中** 05–11 16:10
  快件交给 ，正在派送途中（联系电话: 顺丰已开启“安全呼叫”保护您的电话隐私,请放心接听！）（主单总件
- **运输中** 05–11 15:59
  快件到达 【北京大兴宝善街店】
- 05–11 15:12
  快件在【北京马驹桥中转场】完成分拣，准备发往 【北京大兴宝善街店】
- 05–11 13:34
  快件到达 【北京马驹桥中转场】



When package **arrives/leave at checkpoint**, notification will be sent

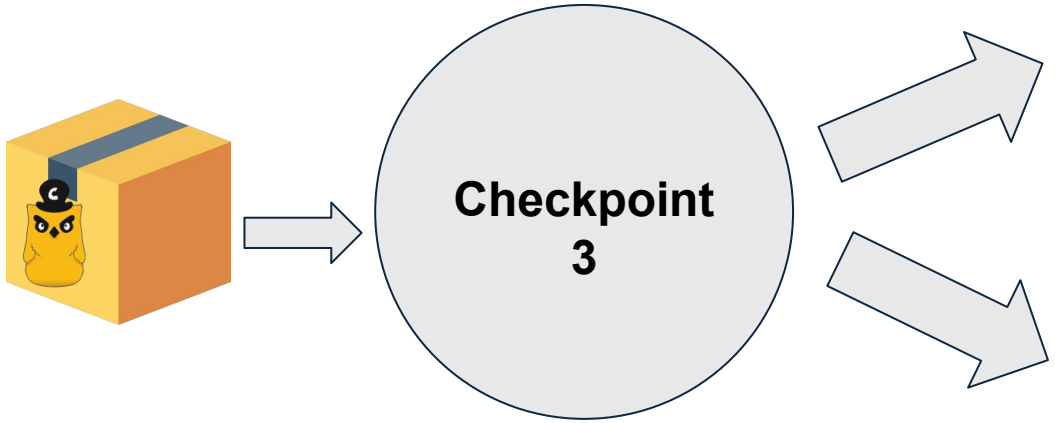


Each checkpoint operates **independently** in different part of the world.

How can each checkpoints let the factory know the CloudyExpress package has arrived at their location?

What need to be sent and where to send?

Where need to be sent?



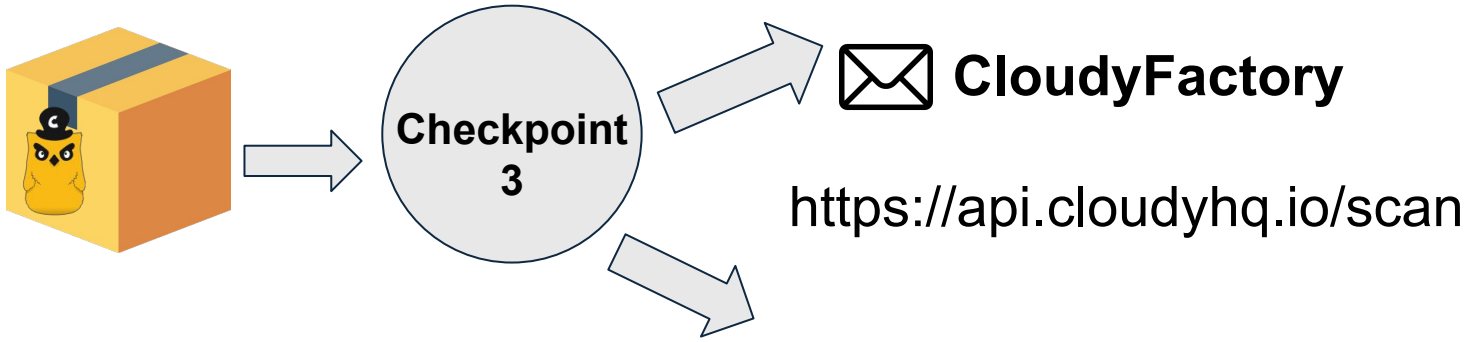
CloudyFactory

https://api.cloudyhq.io/scan

Toronto Store

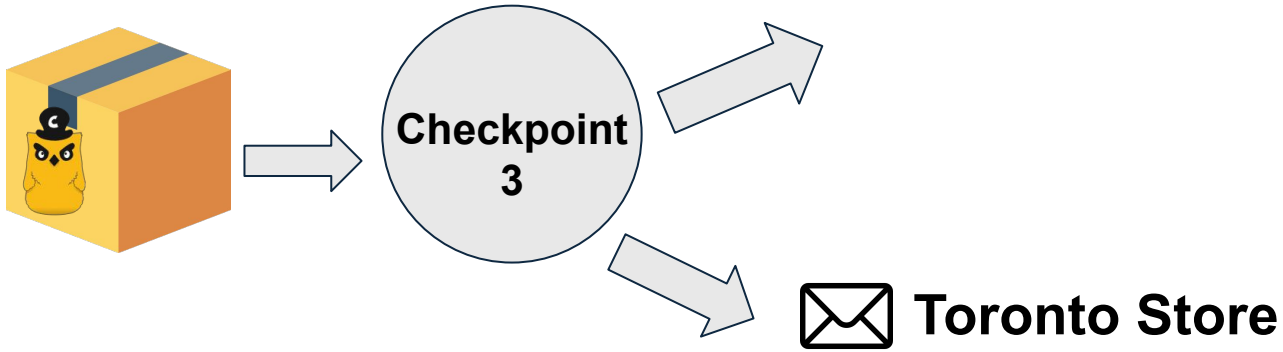
https://api.cloudyhq.io/notify-store

What need to be sent?



1. Which checkpoint it is?
2. What is the package id?
3. When it is scanned?
4. Type: is it a arriving or leaving package?

What need to be sent?

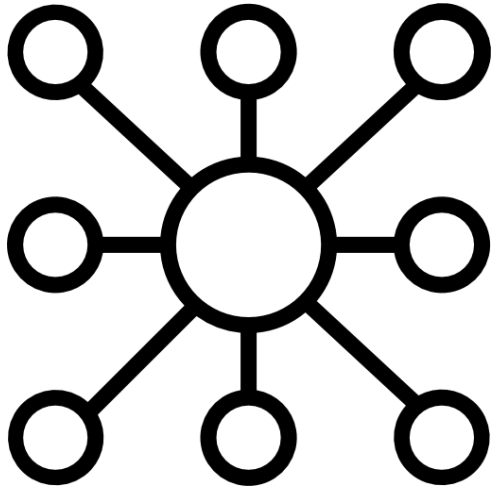


https://api.cloudyhq.io/notify-store

1. Which checkpoint it is?
2. What is the package id?
3. When it is scanned?
4. Type: is it a arriving or leaving package?
5. Where is the checkpoint?
6. Who scanned it?

How can we know?

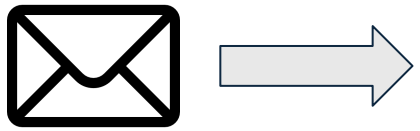
- **What** should be contained in the event?
- **Where** should the event being sent to?
- What **types** of event are there?
  - Factory event or customer-facing event?



We need a central information hub, that can maintain the information

Each checkpoint can just retrieve the info from there

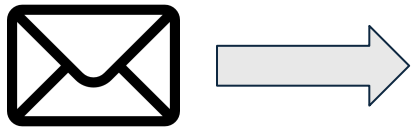
CloudyFactory



https://api.cloudyhq.io/scan

Event Type:
CloudyHQ.Factory.Arrive
CloudyHQ.Factory.Depart

Toronto Store



https://api.cloudyhq.io/notify-store

Event Type:
CloudyHQ.Store.Arrive
CloudyHQ.Store.Depart

What is Discovery?

What events are used in a specific context?

⇒ ***Event definition registry***

What do they look like?

⇒ ***Schema registry***

Where can I consume or produce them?

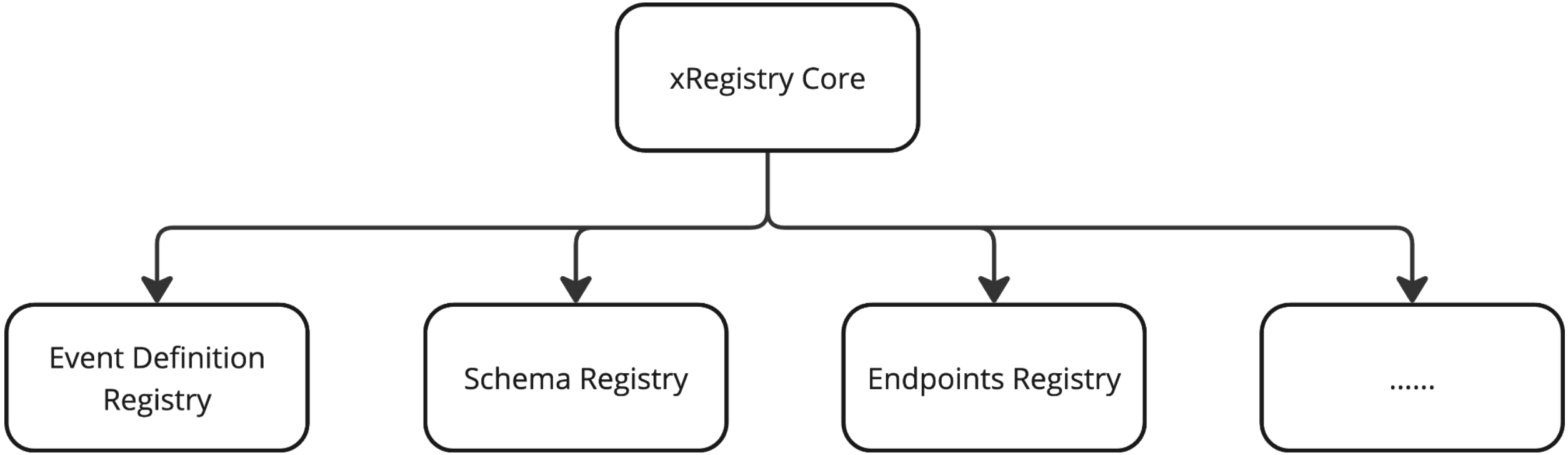
⇒ ***Endpoint registry***

Source: Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

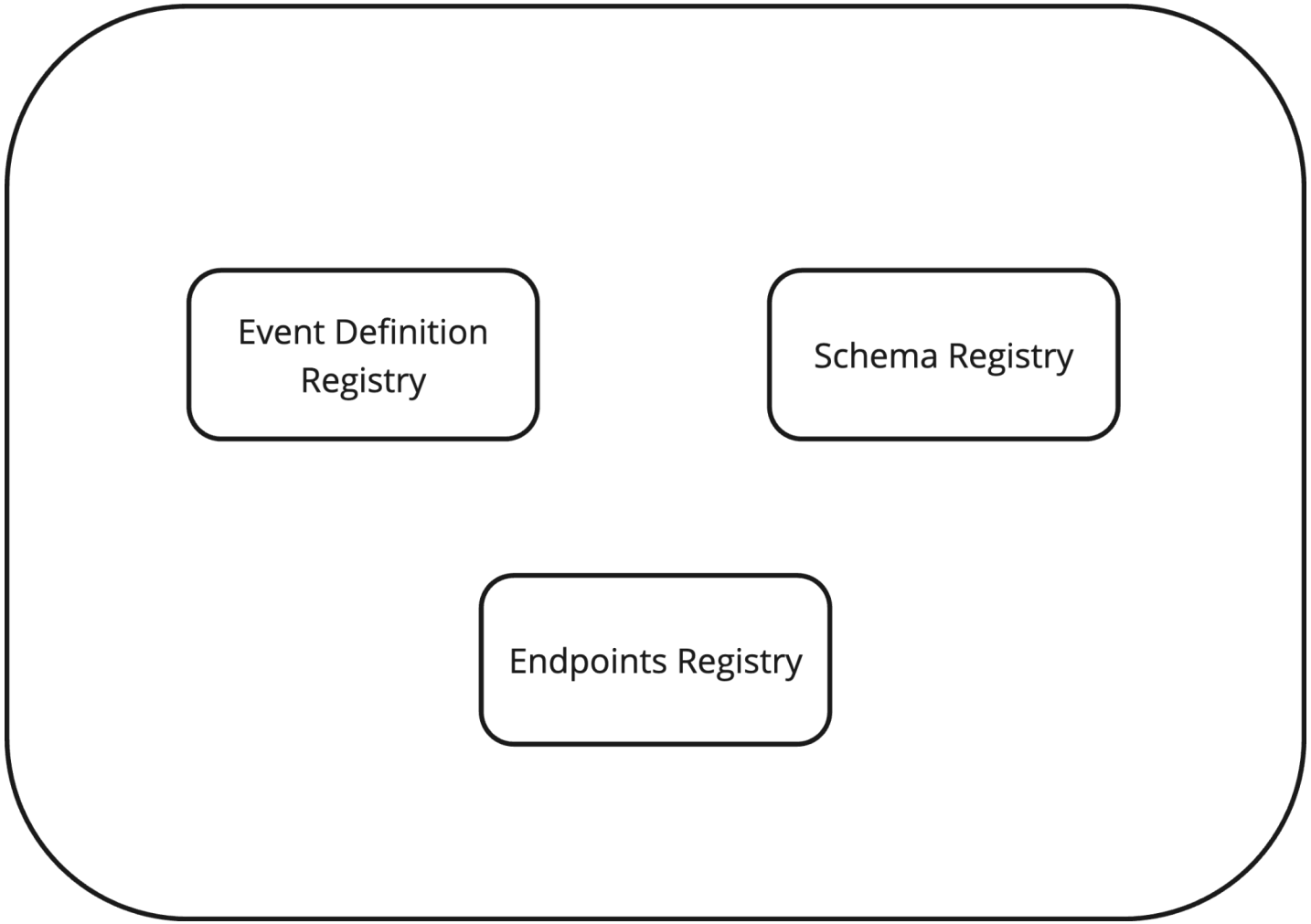
Here introducing xregistry

xREGISTRY

xREGISTRY

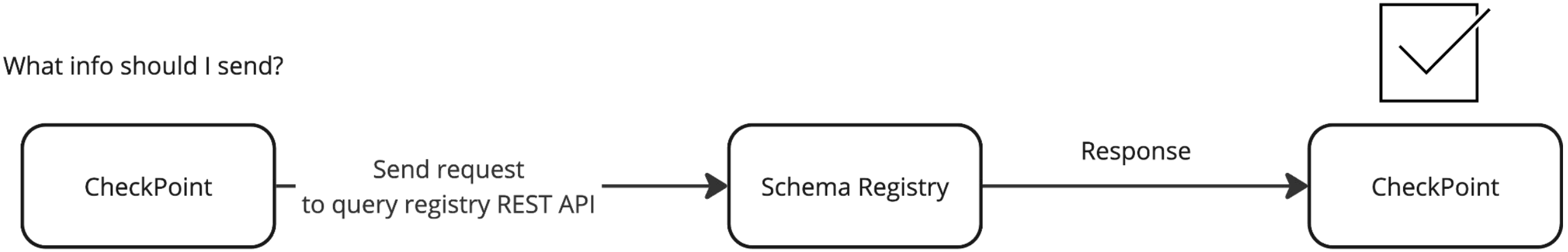


CloudyHQ - Registry System



xREGISTRY

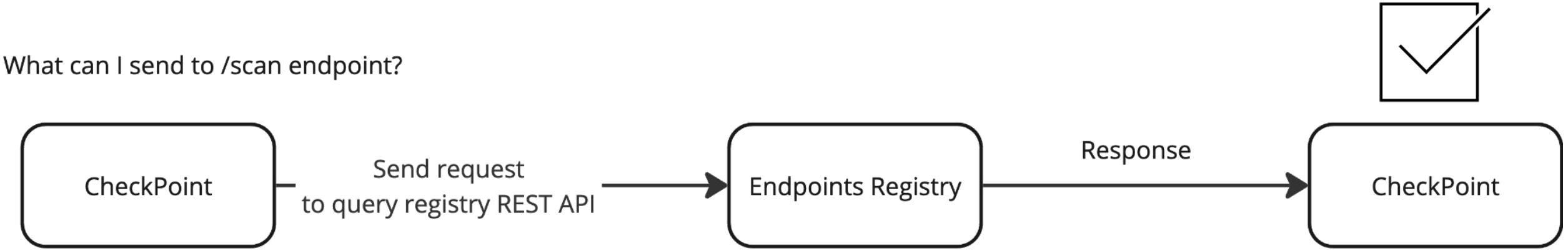
What info should I send?



You need to send package id, timestamp....

xREGISTRY

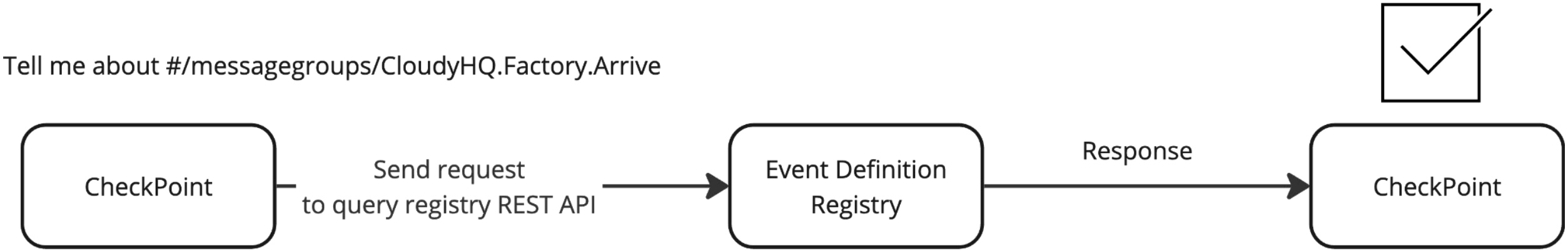
What can I send to /scan endpoint?



You can:

- Send these types of event
  - #/messagegroups/CloudyHQ.Factory.Arrive
  - #/messagegroups/CloudyHQ.Factory.Depart
- Format should be CloudEvents
- Full URL is: https://api.cloudyhq.io/scan
- Protocol should be HTTP

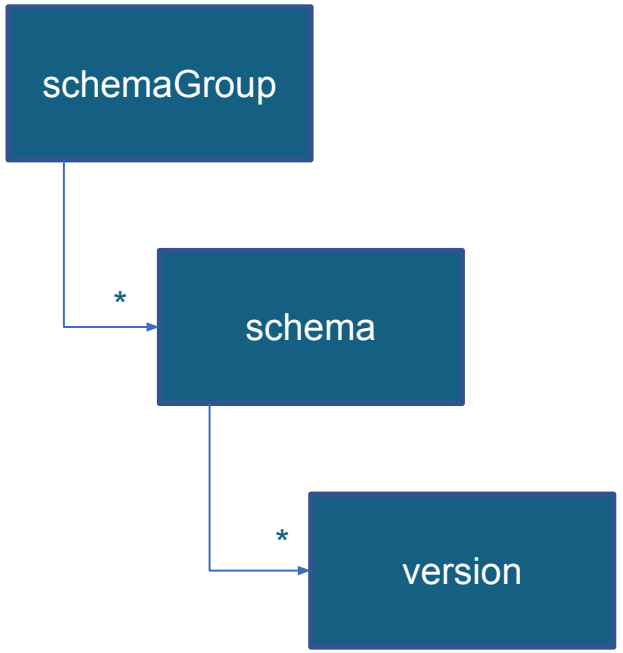
xREGISTRY



Sure:

- Schema URL is:
  #/schemagroups/CloudyHQ.Factory.Arrive/schemas/ArriveData
- Format should be JSONSchema/draft-07
- Event type should be: CloudyHQ.Factory.Arrive
- ...

Schema registry

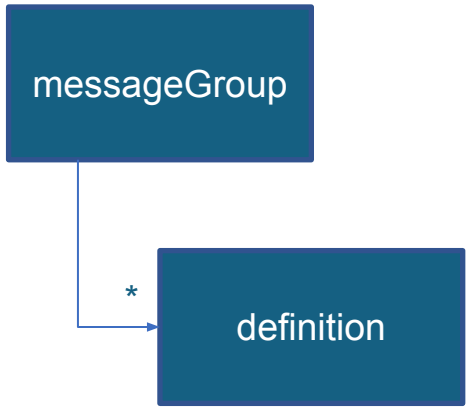


- Collect schema information for payloads in groups
- Store multiple versions of a schema
- Schema definitions are widely used beyond messages and events
- Independent from a specific schema definition language

```
"Schemagroups": {
 "CloudyHQ.Factory": {
     "name": "CloudyHQ Factory Events",
     …
     "schemas": {
      "CloudyHQ.Factory.Arrive": {
       …
       "versions": {
        "1": {
         "id": "1",
         "format": "CloudEvents/1.0",
        …
        }
       }
}
```

Source: modified based on Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

Event definitions



Groups of event definitions

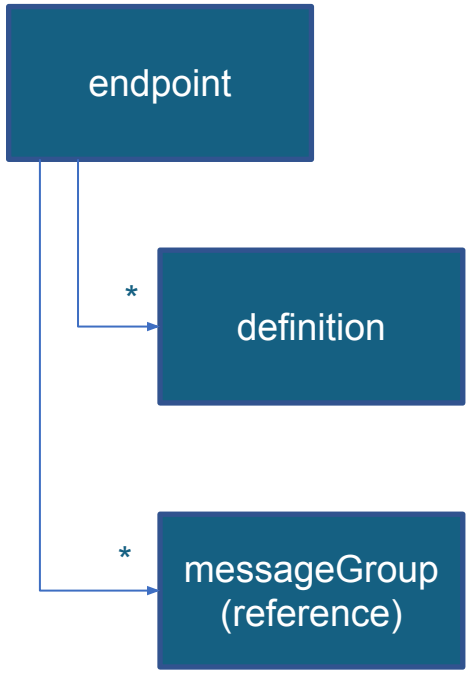
Other message formats possible:

- AMQP
- MQTT
- Protobuf
- HTTP
- …

```
“messageGroup”:{
  "id": "CloudyHQ.Factory",
  "description": "CloudyHQ Factory Events",
  "format": "CloudEvents/1.0",
  "metadata": {
    "attributes": {…}
  },
  "schemaurl":"#/schemagroups/CloudyHQ/schemas/PackageTrackingData/versions/1”
}}
```

Source: Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

Endpoints




```
"CloudyHQ.Factory.Http": {
  "id": "CloudyHQ.Factory.Http",
  "usage": "producer",
  "config": {
    "protocol": "HTTP",
    "strict": false,
    "endpoints": [
      "https://api.cloudyhq.io/scan"
    ]
  },
  "channel":"my/topic",
  "messageGroups": [
    "http://localhost:8080/reg-Messages/CloudyHQ.Factory"
  ],
  "format": "CloudEvents/1.0"
}
```

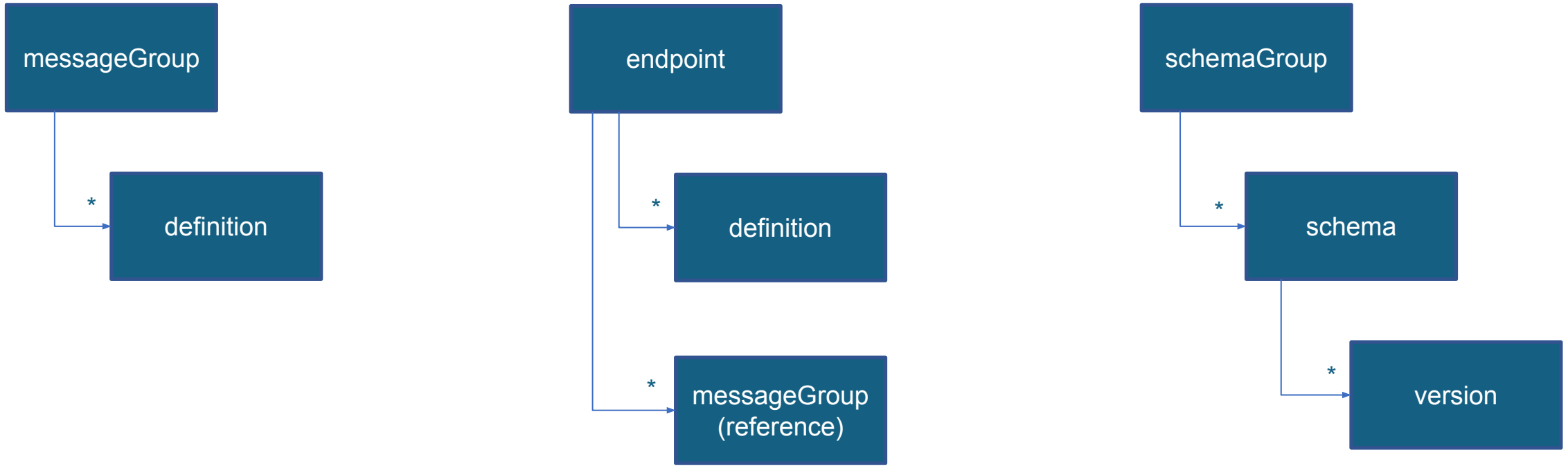
Superset of **`messageGroup`**

Additional attributes

- **`usage`**: producer | consumer | subscriber
- **`channel`**: Correlate multiple endpoints of the same channel
- **`messageGroups`**: References to existing definition groups
- **`config:`** Connectivity configuration

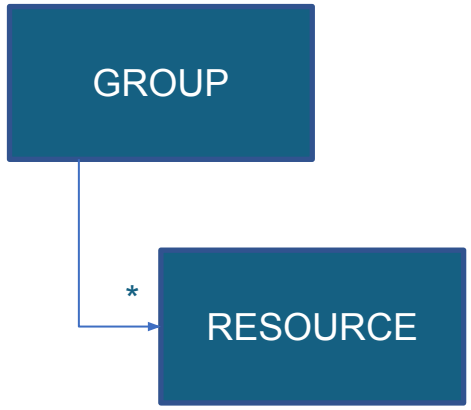
Source: Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

What do these have in common?



Source: Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

A Registry for Groups of Resources



API Access

```
/model                                  # Manage the registry model
/                                       # Show all Groups
/GROUPs                                 # Manage a Group Type
/GROUPs/gID                             # Manage a Group
/GROUPs/gID/RESOURCEs                   # Manage a Resource Type
/GROUPs/gID/RESOURCEs/rID               # Manage the latest Resource version
/GROUPs/gID/RESOURCEs/rID?meta          # Metadata about the latest Resource version
/GROUPs/gID/RESOURCEs/rID/versions      # Show version strings for a Resource
/GROUPs/gID/RESOURCEs/rID/versions/vID          # Manage a Resource version
/GROUPs/gID/RESOURCEs/rID/versions/vID$meta     # Metadata about a version
```

Source: Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

Start Small...

All in one `.cereg` file

Managed with your source code

Inline everything

- Schema inside event definition
- Event definition inside endpoint

With infrastructure support

- Just schemas and event definitions
- Let infrastructure create endpoint definition

Source: Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

...Or Do Something More Advanced

Host a central registry in your organization

Define events and schemas for re-use

- Governance
- Versioning

Interoperability with other organizations

- Departments inside one company
- Vendors
- Customers

Federated discovery

Source: Introducing CloudEvents Discovery - Clemens Vasters & Klaus Deissner - KubeCon EU 2023

xRegistry has the tool to make developers life easier

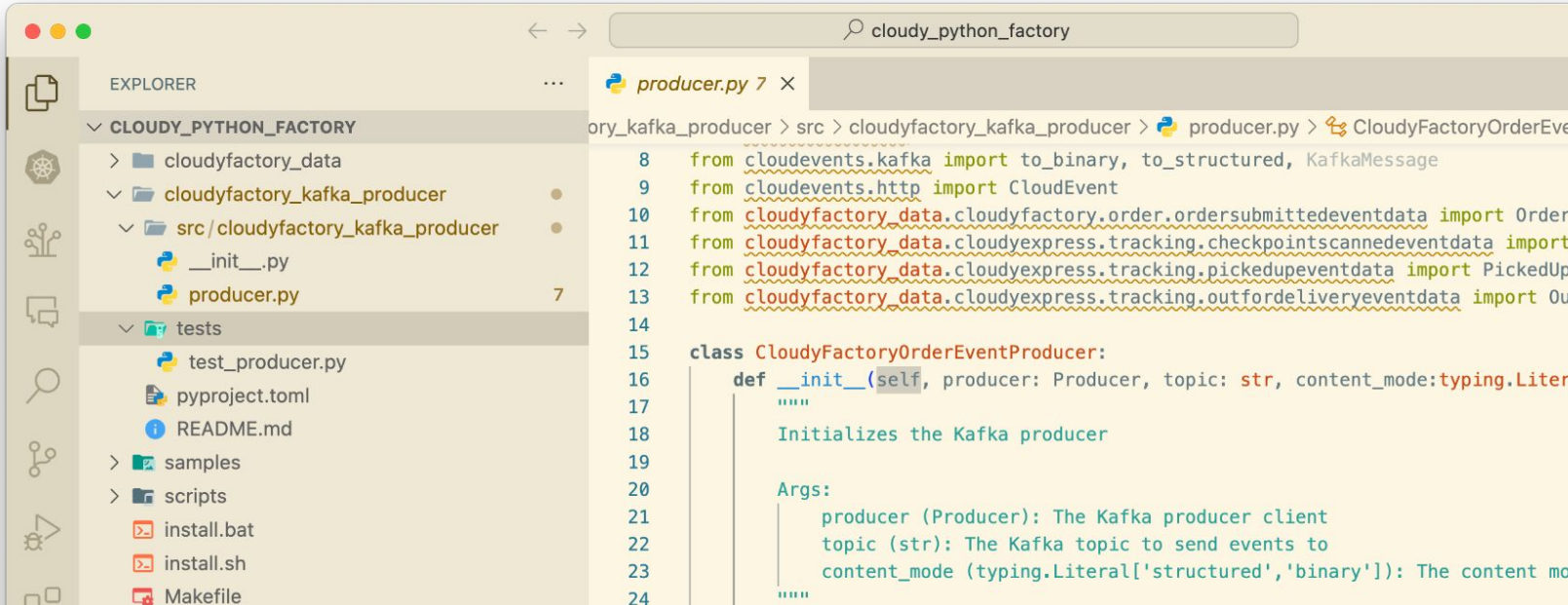
Automatically **generate code artifacts**
from xRegistry definitions, especially message catalogs.

xRegistry CLI

xRegistry has the tool to make developers life easier

Automatically **generate code artifacts**
from xRegistry definitions, especially message catalogs.

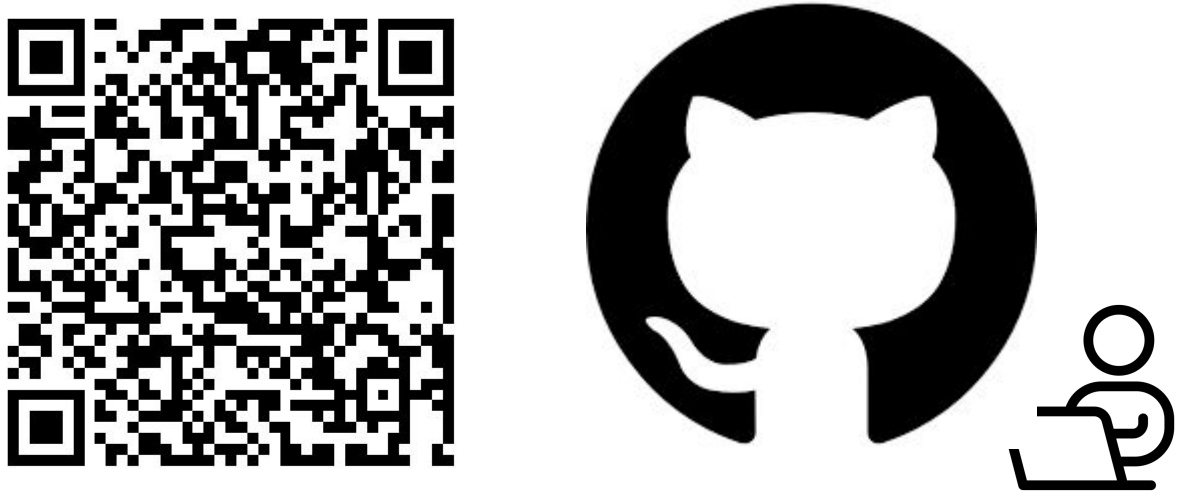
```
xregistry generate
--language=openapi
--style=producer
--projectname=MyProjectProducer
--definitions=definitions.json
--output=MyProjectProducer
```



CLI Video Demo

https://drive.google.com/file/d/1sPDkJ3RzrylorBVsyPR77mXnadQIlPIV/view?usp=sharing

Try yourself!



See how xRegistry is implemented in CloudyFactory

https://github.com/Leo6Leo/xreg-github/pull/1/files

Agenda

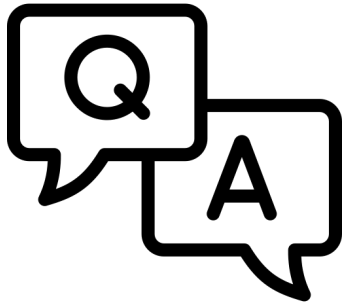
CloudEvents(**CE**): Intro & Community updates

1. Understand CE: analogy of delivery package
2. Updates from the community
3. What is the next step/direction for CloudEvents?

xRegistry: Event Discovery

1. Understand current problems: Checkpoints analogy
2. How xRegistry resolves the current problem?
3. xRegistry work in practice
4. xRegistry CLI

Q & A



Q&A

You are welcome to ask questions now!

Stay Connected

#cloudevents

#xregistry

Thank you

KubeCon

CloudNativeCon

China 2024

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

多谢！