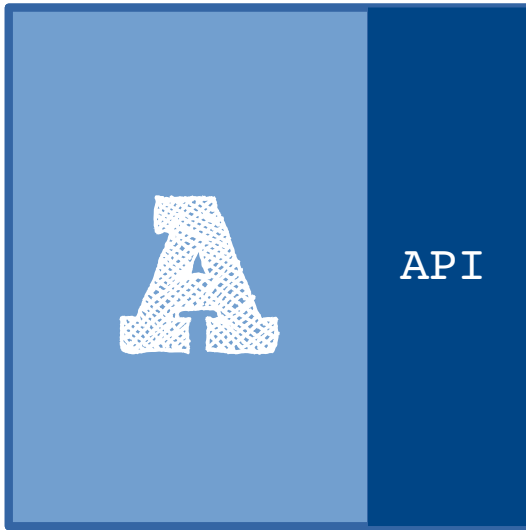


Wenn

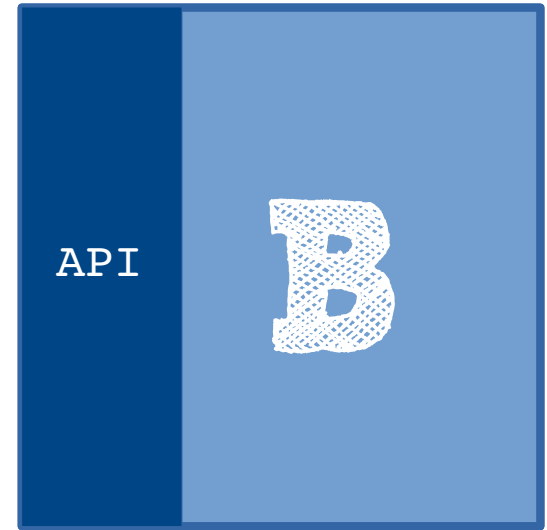
# SCHNITTSTELLEN

alt werden...

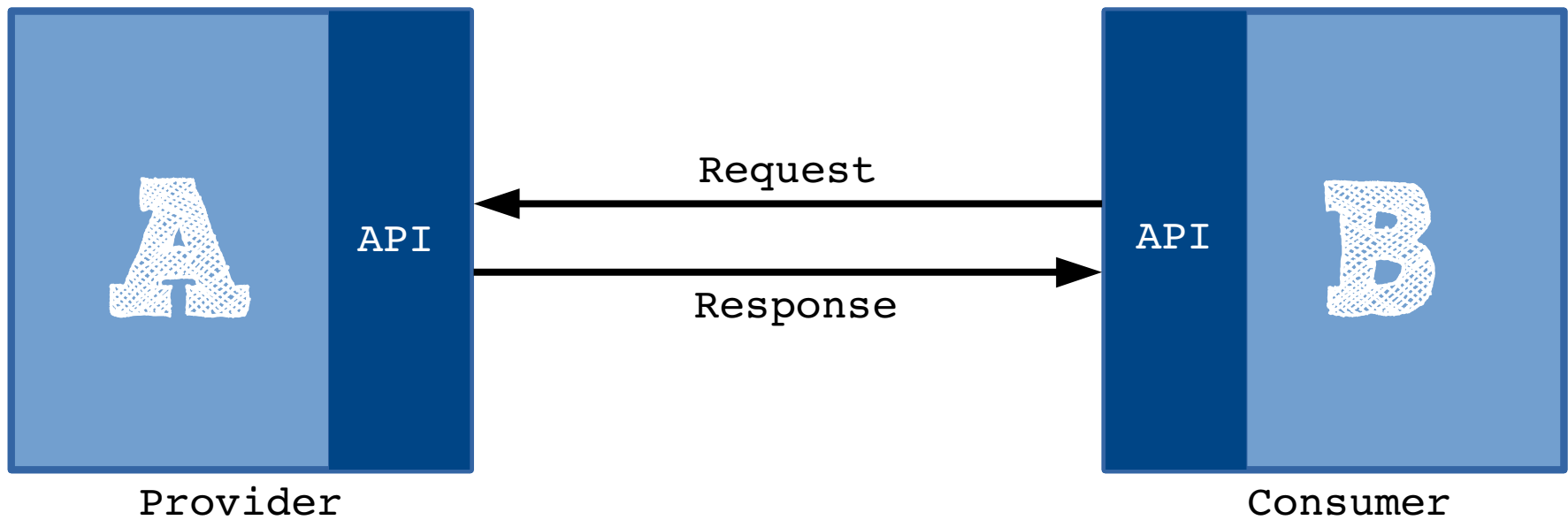


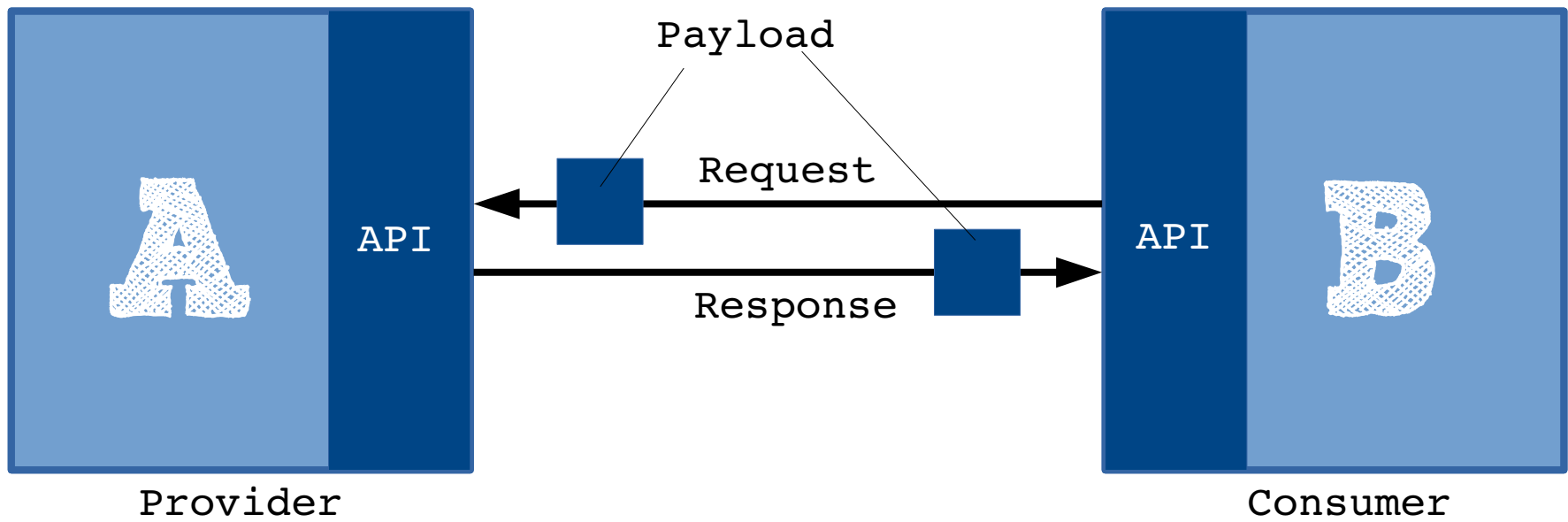


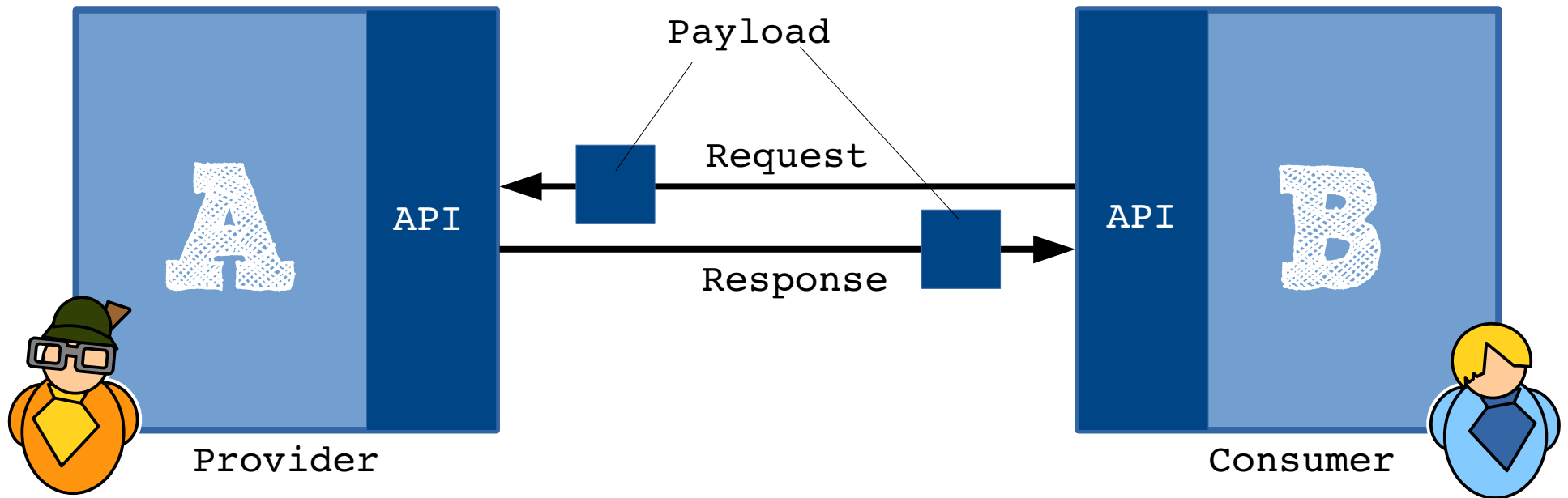
Provider

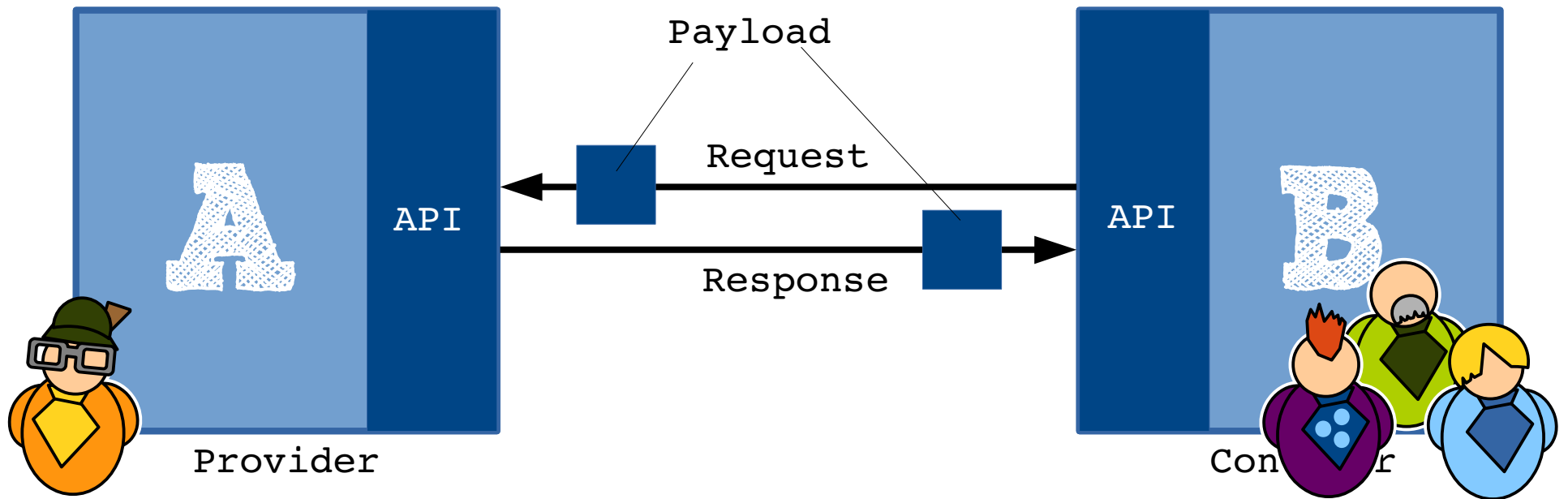


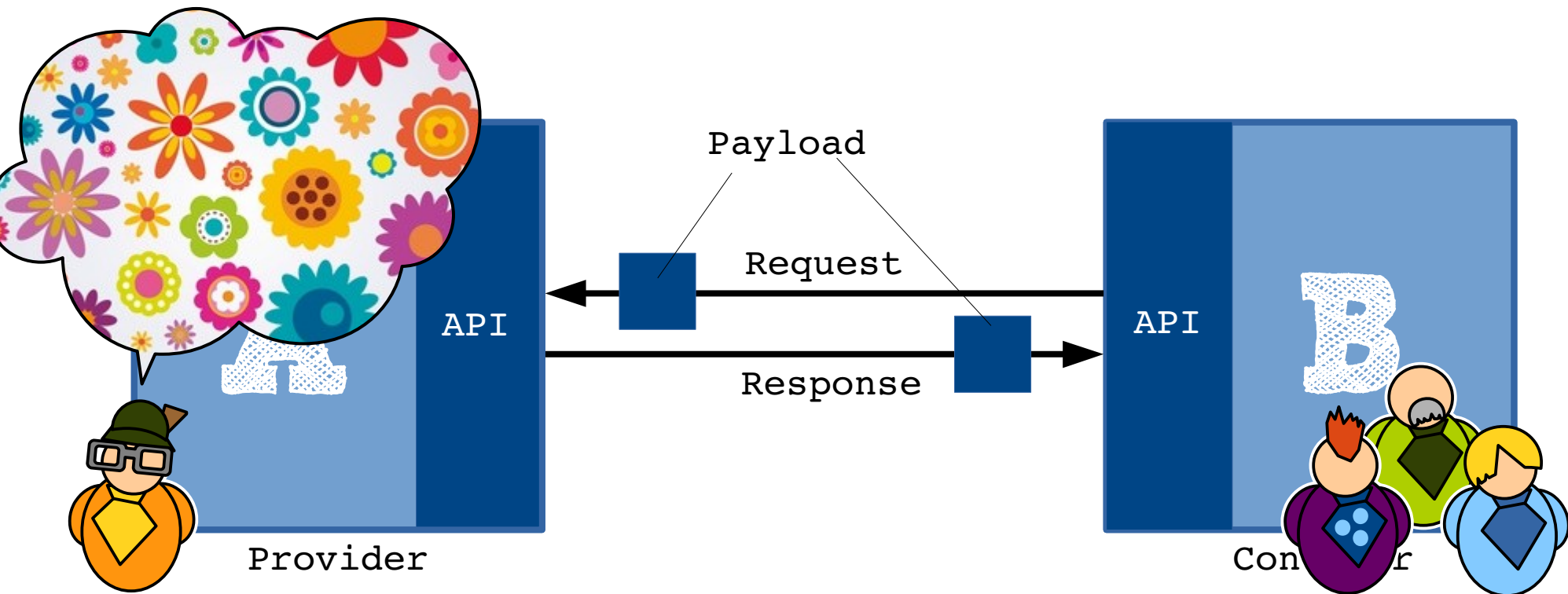
Consumer

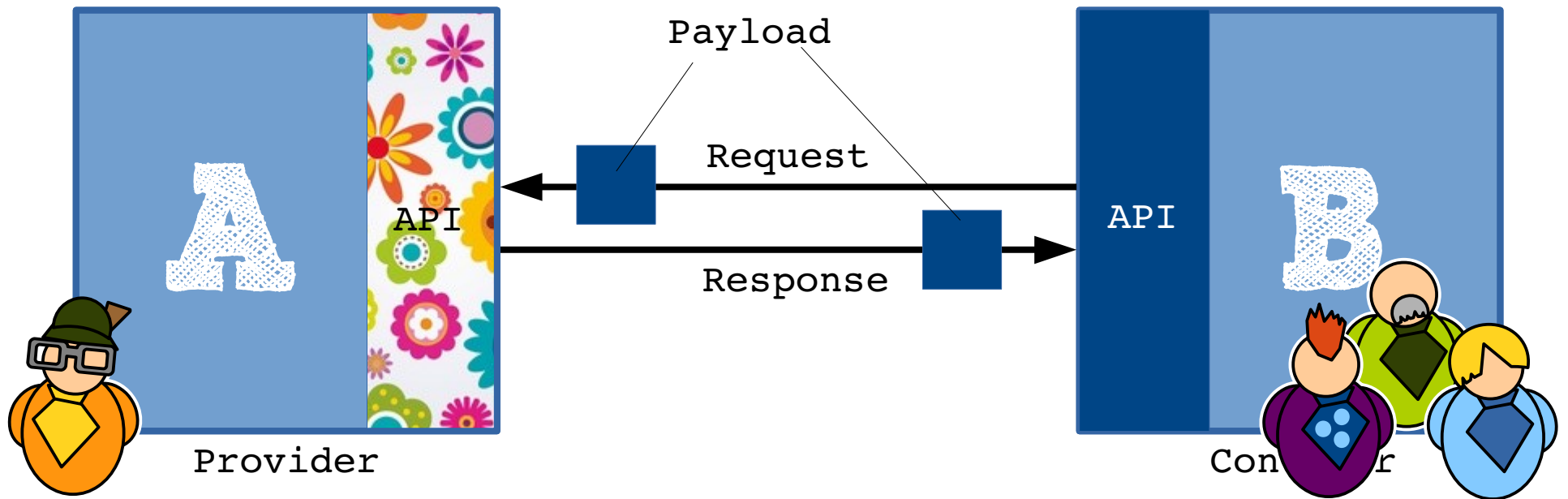




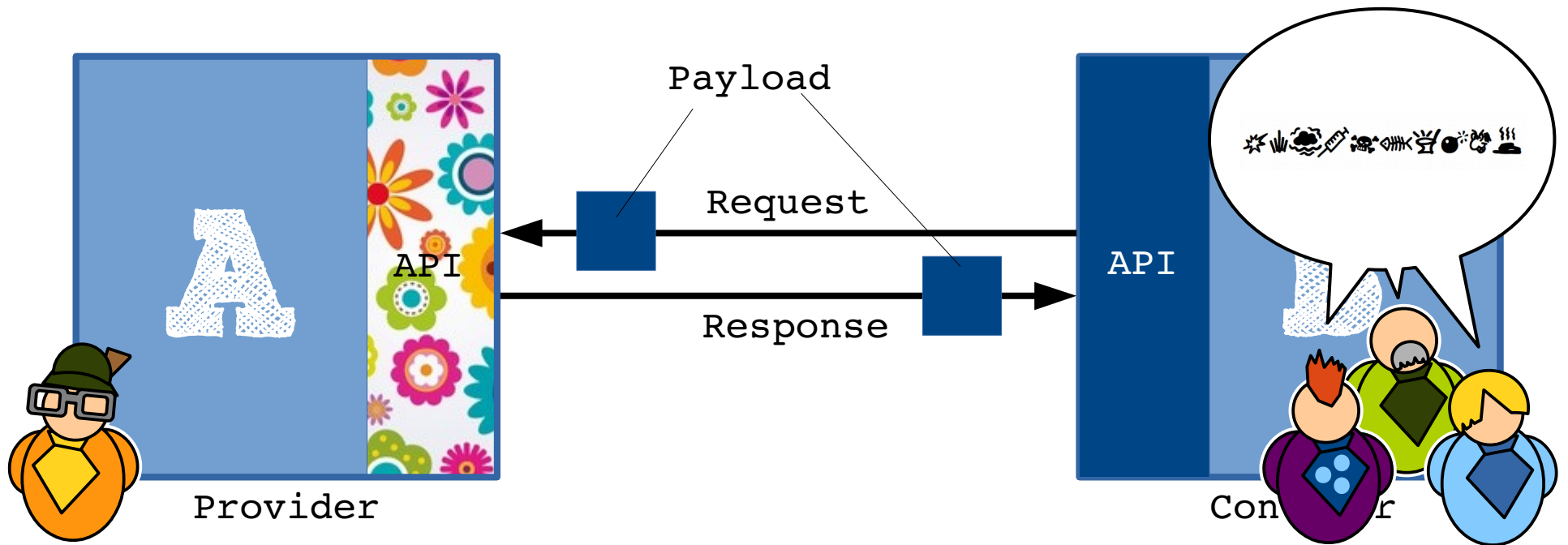


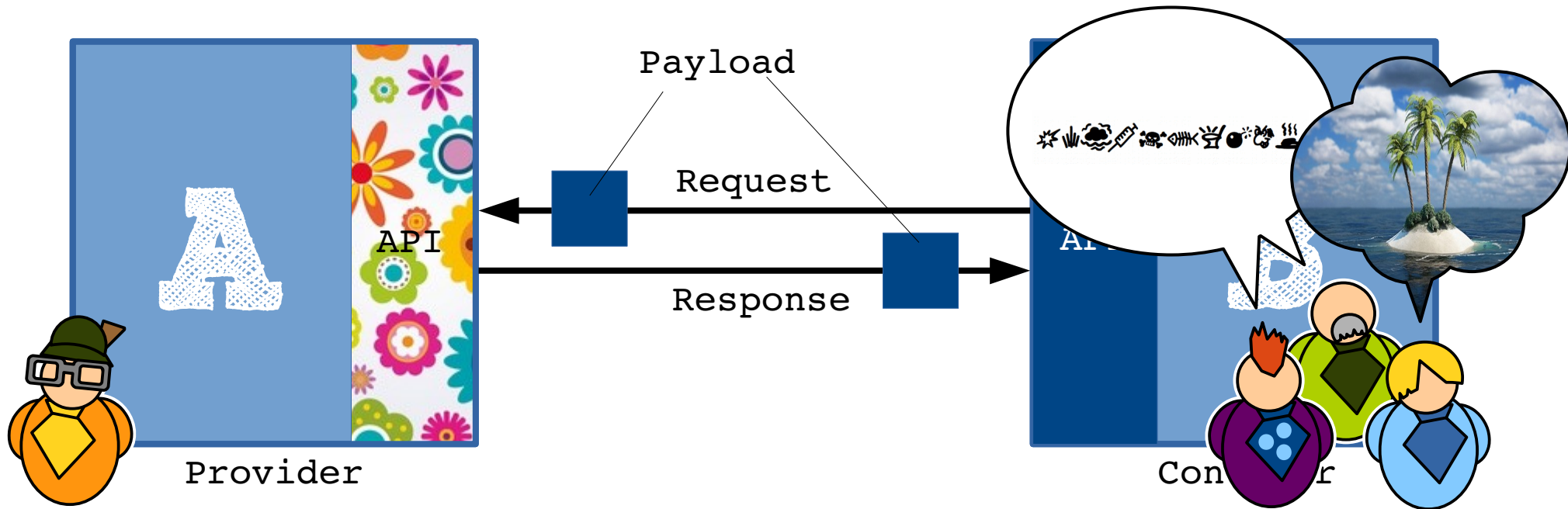


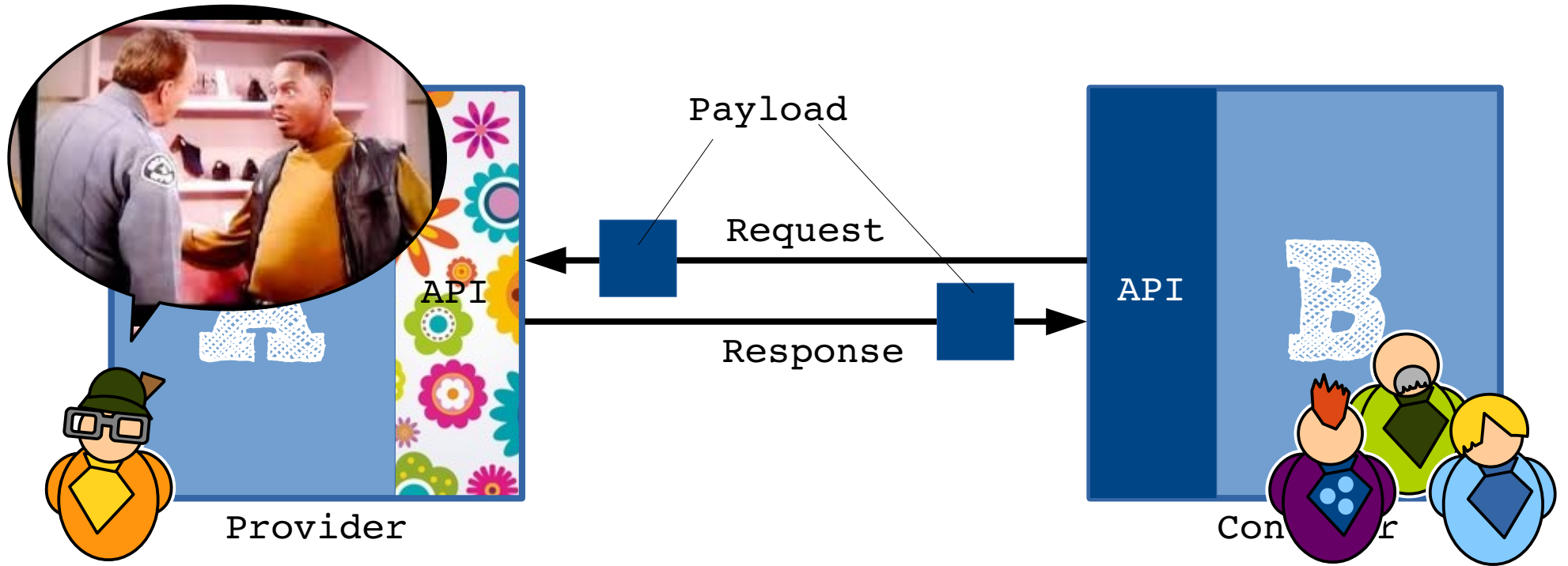


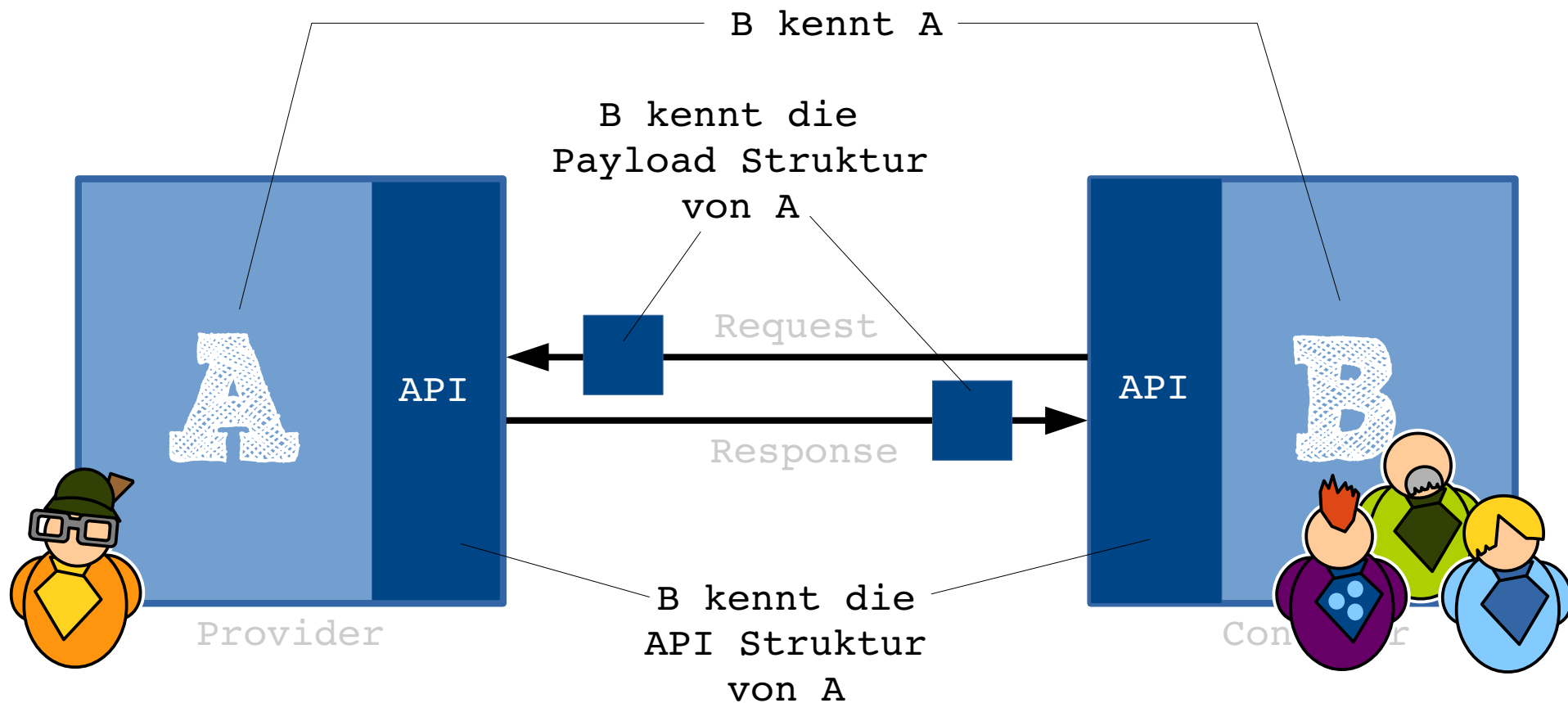






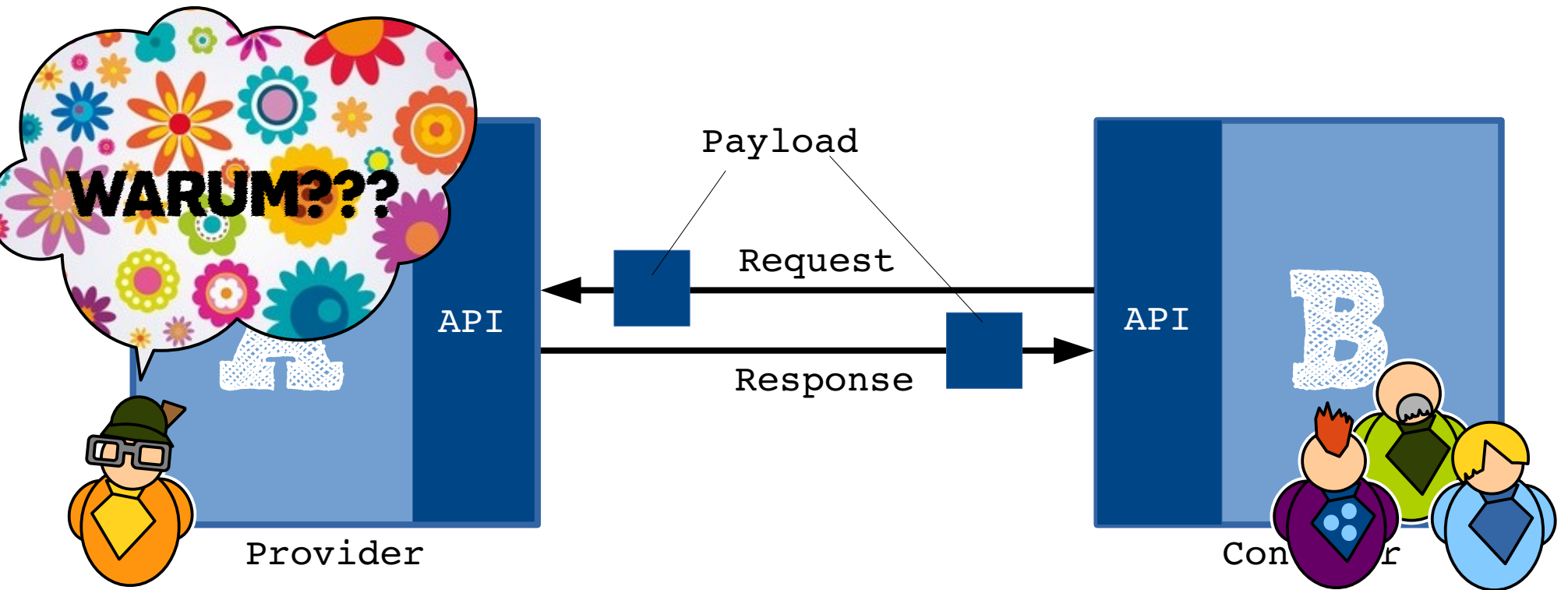


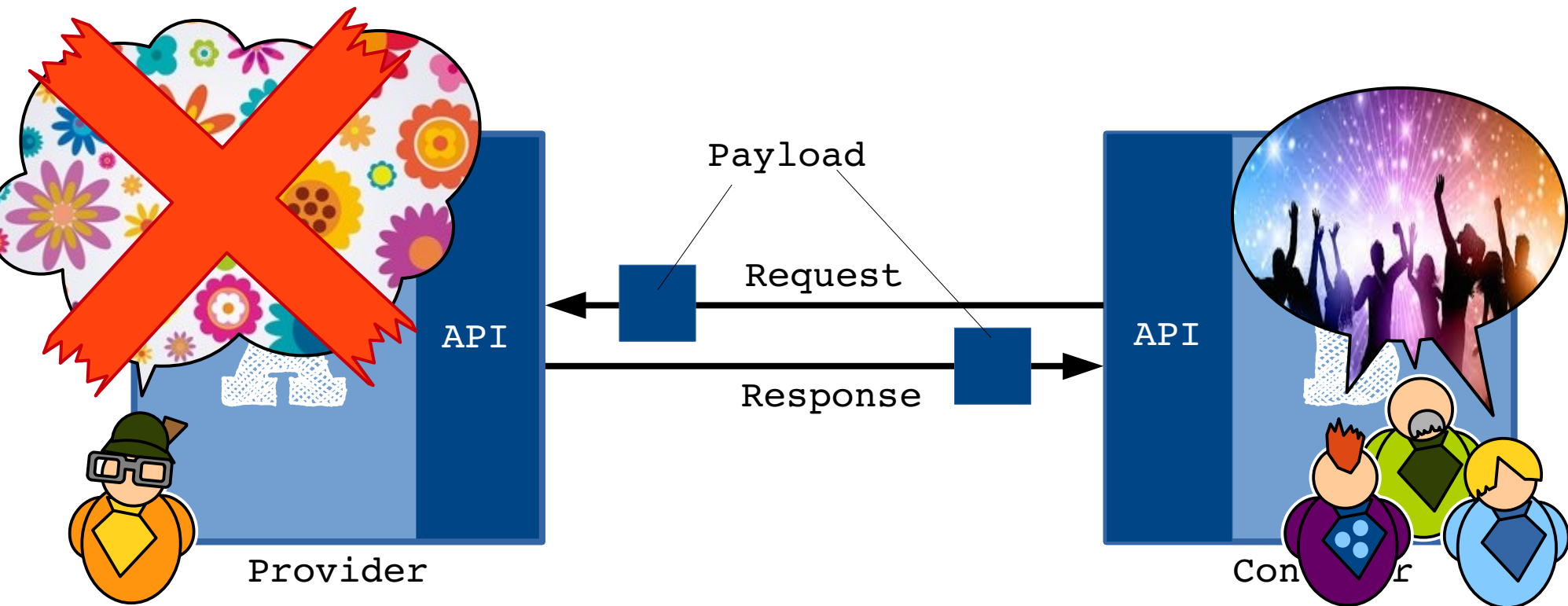




System **B** muss ziemlich viel von System

**A** kennen, damit die Kommunikation funktioniert!





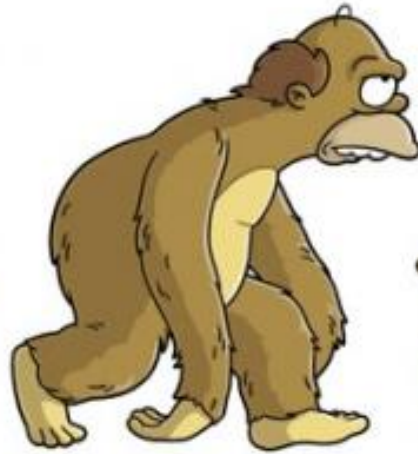
# EVOLUTION



MONKIUS EATALOTIS



CHIMPUS IMBECILUS



APEIS STUPIDIUS



NEANDERSLOB



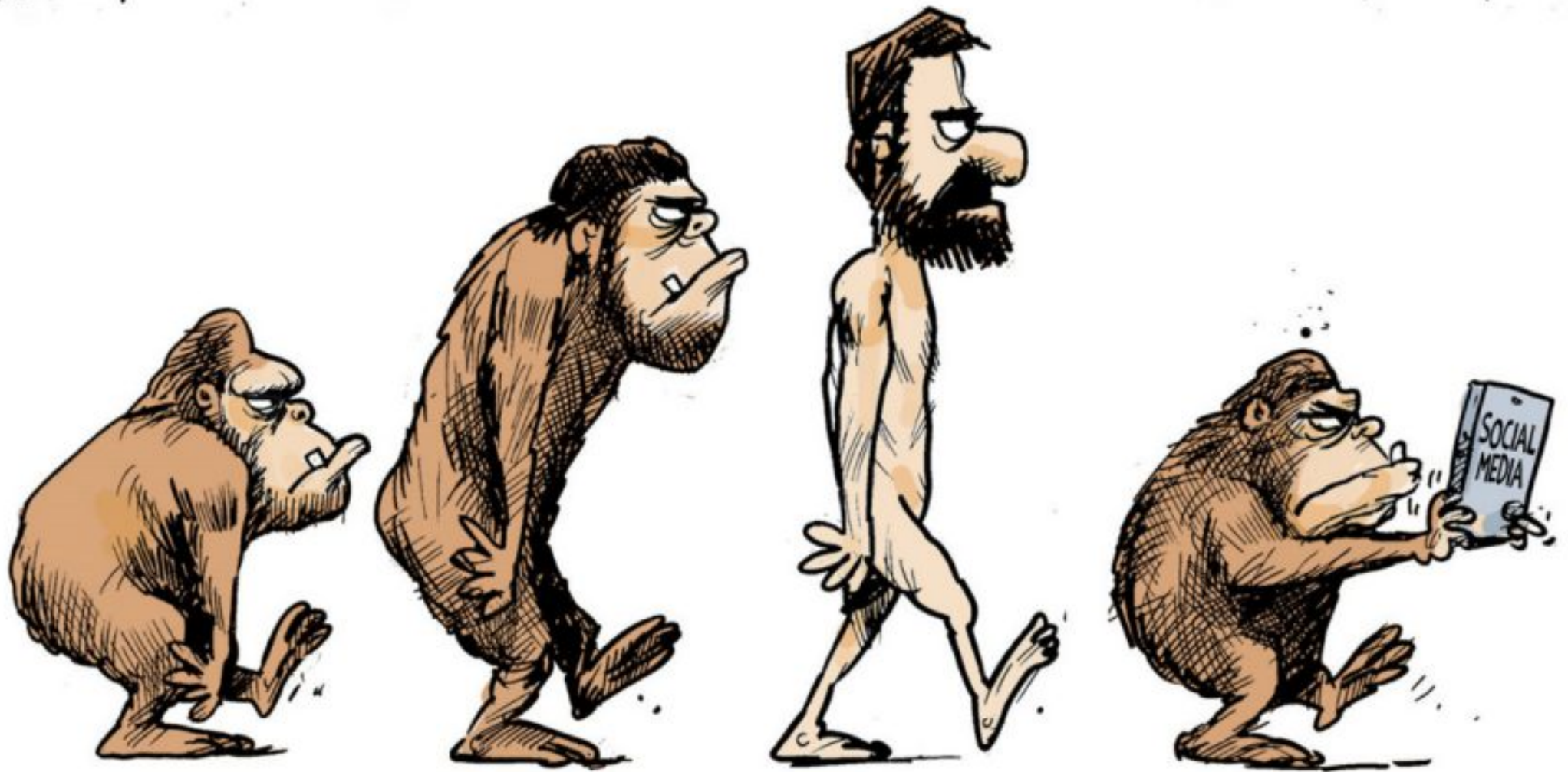
HOMERSAPIEN

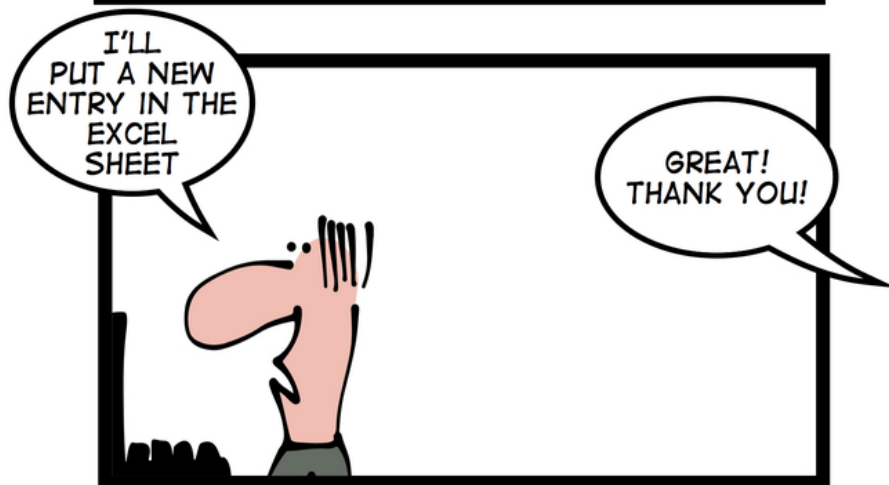
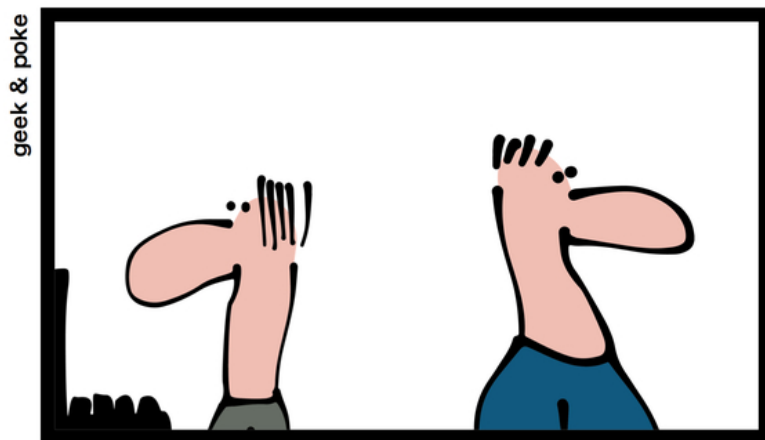
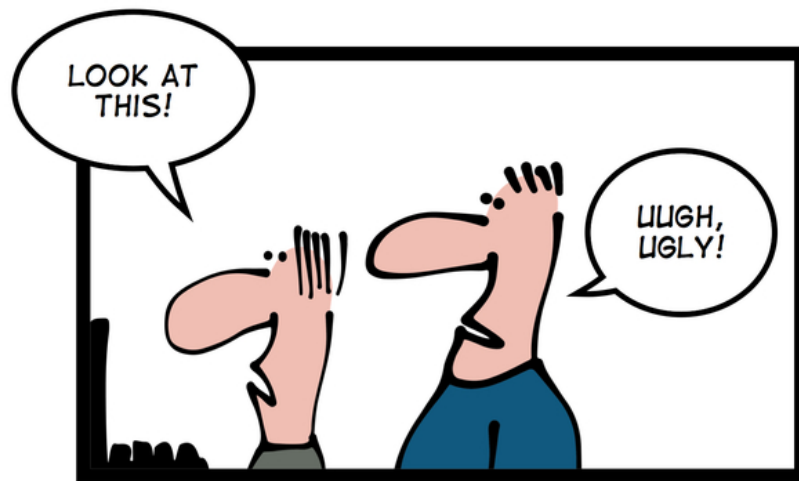
HOMERSAPIEN

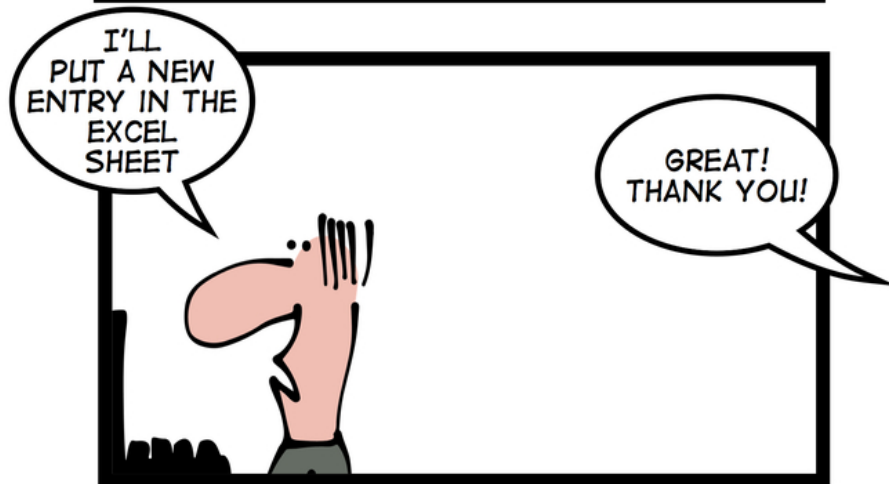
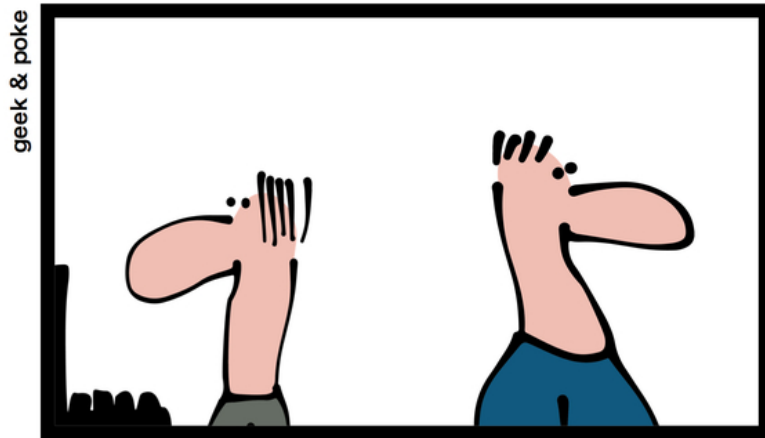
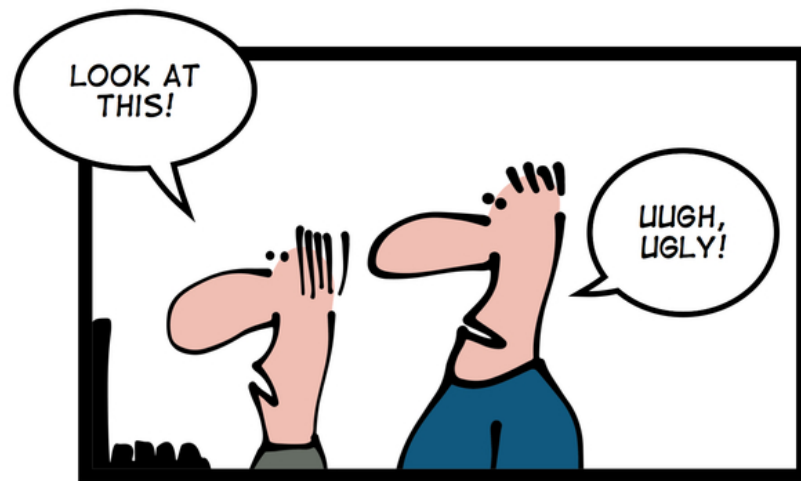


# EVOLUTION 2.0

GREG  
PERRY

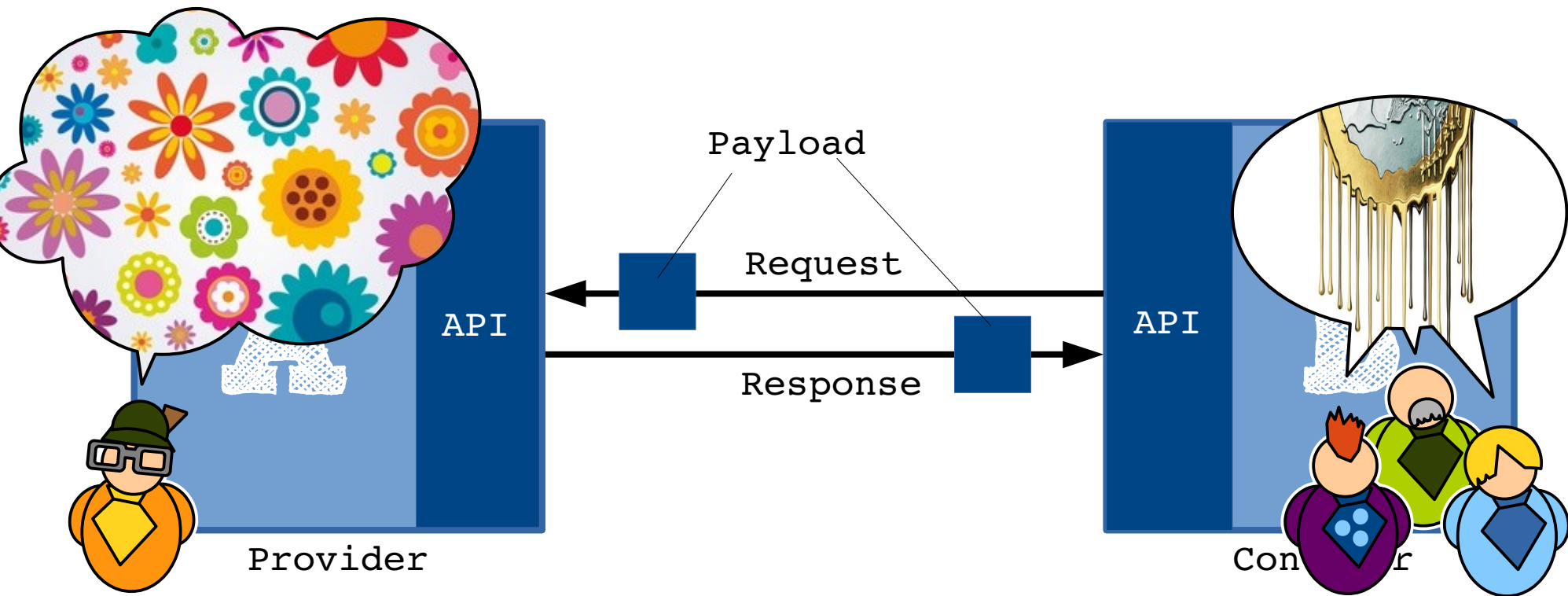


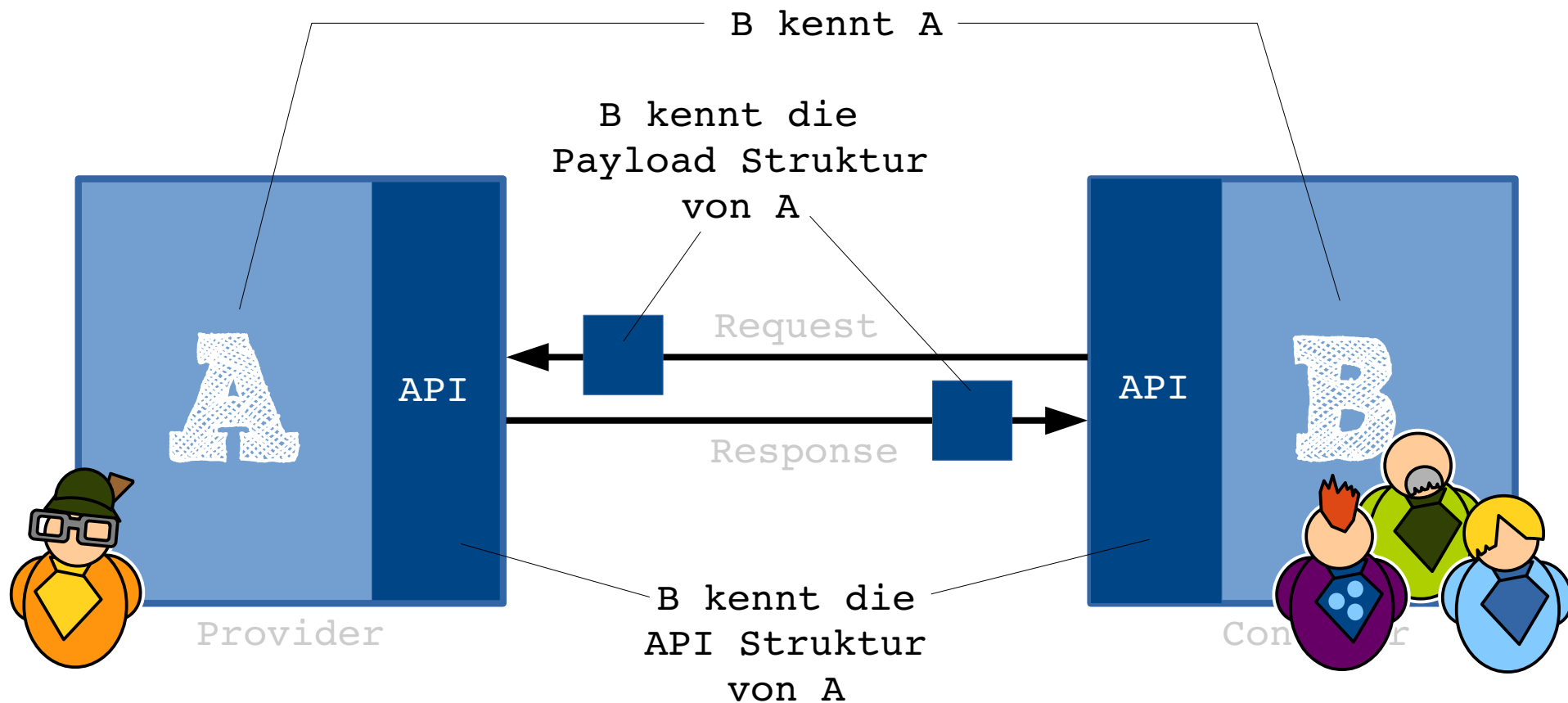


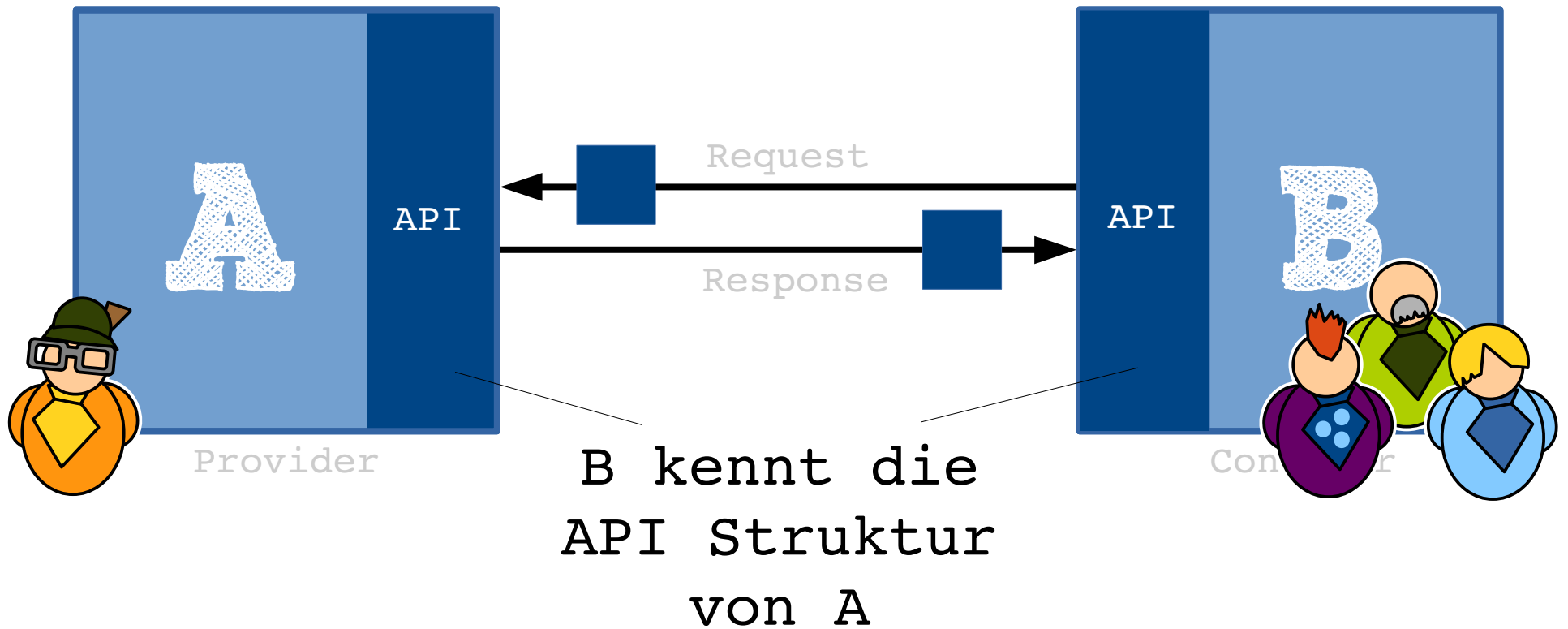


technicalDebts.xls

# TECHNICAL DEPT









# RESTBUCKS, THE SAMPLE



<http://codebetter.com/glennblock/2011/06/06/rest-in-practice-the-book-restbucks-the-sample>

<https://github.com/olivergierke/spring-restbucks>

<https://speakerdeck.com/olivergierke>

<http://olivergierke.de/2016/04/benefits-of-hypermedia/>



[\*\*https://RESTBUCKS.IO/V1/CREATE\\_ORDER\*\*](https://RESTBUCKS.IO/V1/CREATE_ORDER)

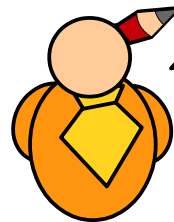




[\*\*https://RESTBUCKS.IO/V2/CREATE\\_ORDER\*\*](https://RESTBUCKS.IO/V2/CREATE_ORDER)



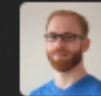
<https://RESTBUCKS.IO/V2/ORDER/{ID}>



POST = create order  
PATCH = update order  
DELETE = delete order  
GET = read order

Speaker Deck

Published on Apr 19, 2016



Oliver Gierke  
70 Presentations

★ Star this Talk 5 Stars

Published in Programming

Stats 6,187 Views

## Share

Twitter, Facebook

</> Embed

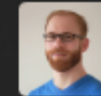
Direct Link

Download PDF

## Domain-Driven Design & REST by Oliver Gierke

Published April 19, 2016 in Programming

Method	URI	Action	Step
POST	/orders	Create new order	1
POST/PATCH	/orders/{id}	Update the order <small>(only if "payment expected")</small>	2
DELETE	/orders/{id}	Cancel order <small>(only if "payment expected")</small>	3
PUT	/orders/{id}/payment	Pay order <small>(only if "payment expected")</small>	4
Barista preparing the order			
GET	/orders/{id}	Poll order state	5
GET	/orders/{id}/receipt	Access receipt	
DELETE	/orders/{id}/receipt	Conclude the order process	6



Oliver Gierke

70 Presentations

★ Star this Talk 5 Stars

Published in Programming

Stats 6,187 Views

## Share

Twitter, Facebook

&lt;/&gt; Embed

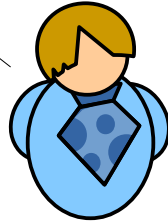
Direct Link

Download PDF

# Domain-Driven Design & REST by Oliver Gierke

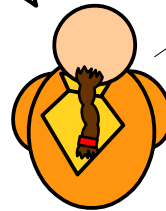
Published April 19, 2016 in Programming

Kunde



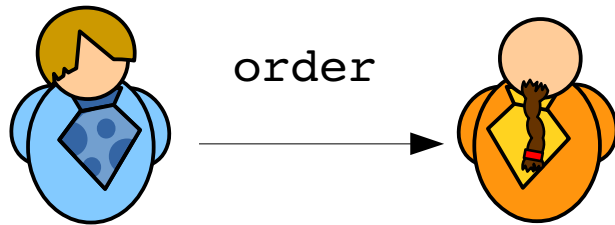
Hi

Hallole

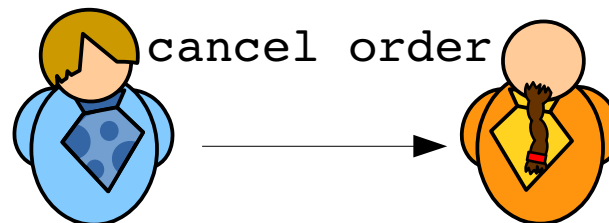
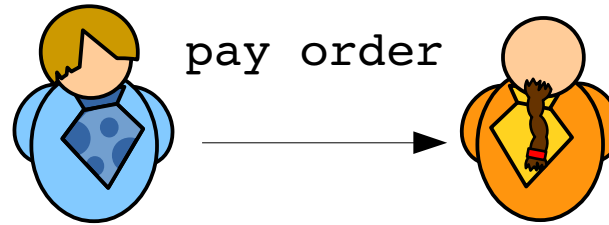
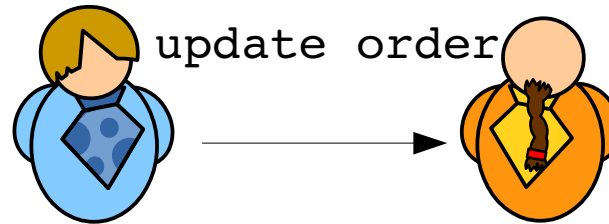


Barista

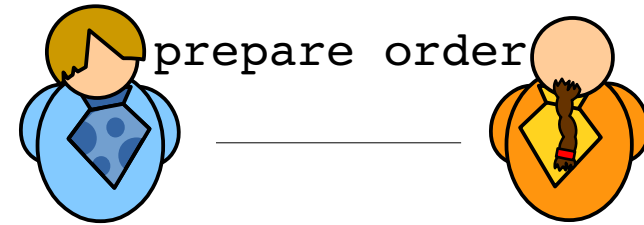
# SCHRITT 1



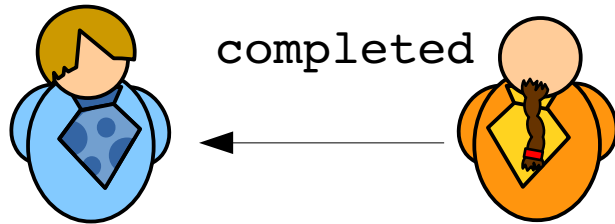
# SCHRITT 2



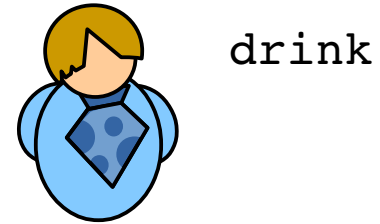
# SCHRITT 3



# SCHRITT 4



# SCHRITT 5



# HATEOAS

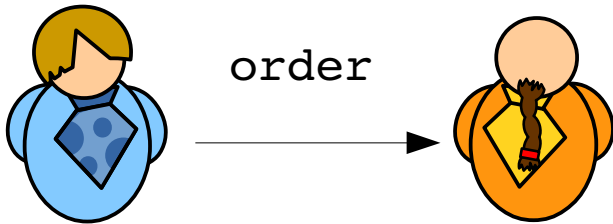
Hypermedia  
As  
The  
Engine  
Of  
Application  
State



# SCHRITT 1

## Request

## Response



POST: RESTBUCKS.io/v2/order

self

GET: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

update\_order

POST/PATCH: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

cancel\_order

DELETE: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

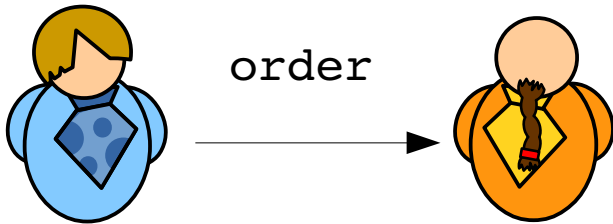
pay\_order

PUT: RESTBUCKS.io/v2/order/1/payment  
RESTBUCKS.io/v2/order/{id}/payment

# SCHRITT 1

## Request

## Response



POST: RESTBUCKS.io/v2/order

### self

GET: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

### update\_order

POST/PATCH: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

### cancel\_order

DELETE: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

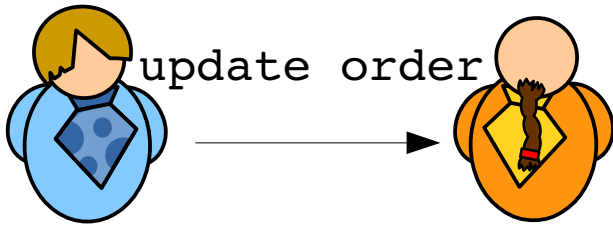
### pay\_order

PUT: RESTBUCKS.io/v2/order/1/payment  
RESTBUCKS.io/v2/order/{id}/payment

# SCHRITT 2

Request

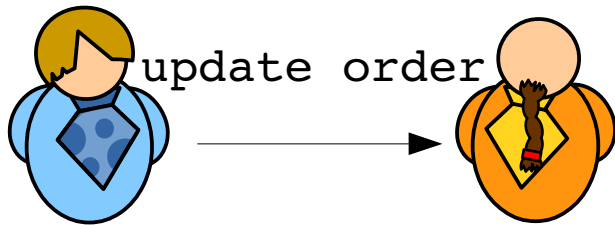
Response



PATCH: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

# SCHRITT 2

## Request



PATCH: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

## Response

### self

GET: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

### update\_order

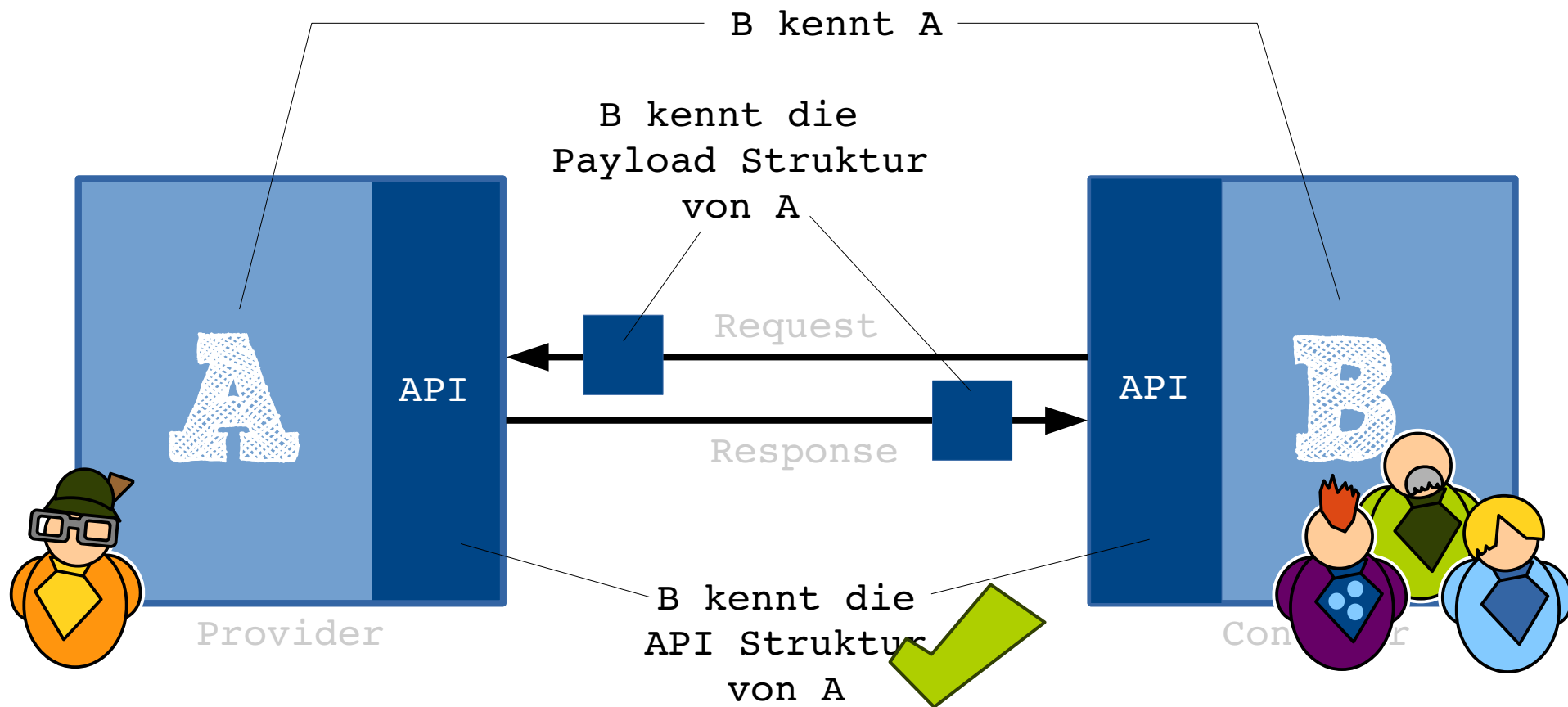
POST/PATCH: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

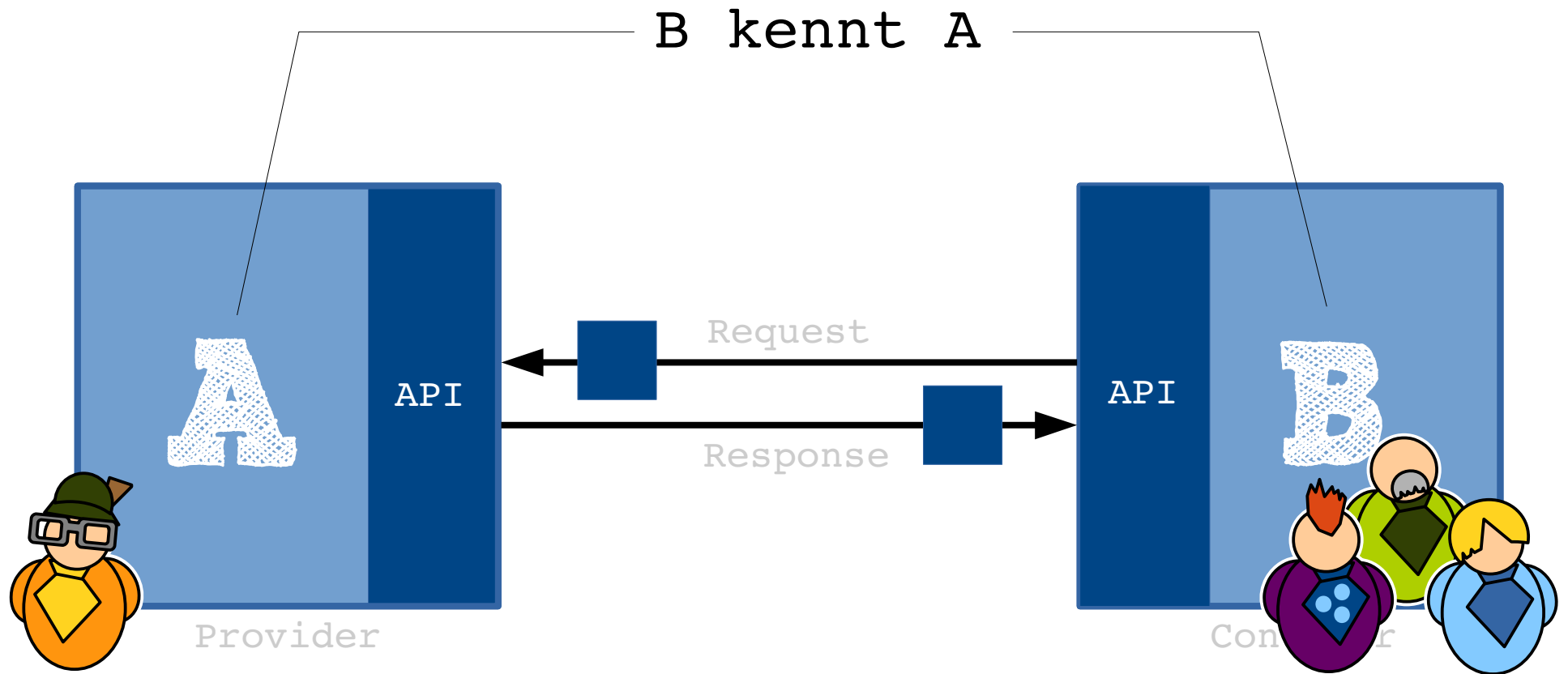
### cancel\_order

DELETE: RESTBUCKS.io/v2/order/1  
RESTBUCKS.io/v2/order/{id}

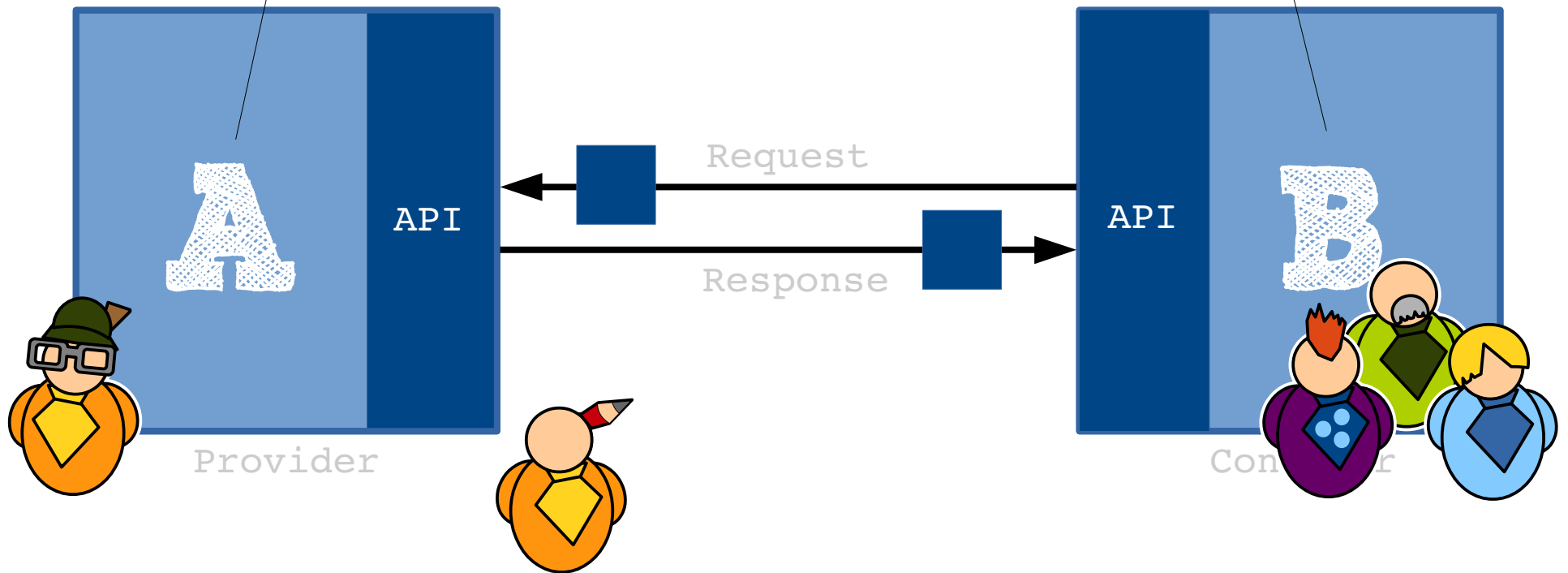
### pay\_order

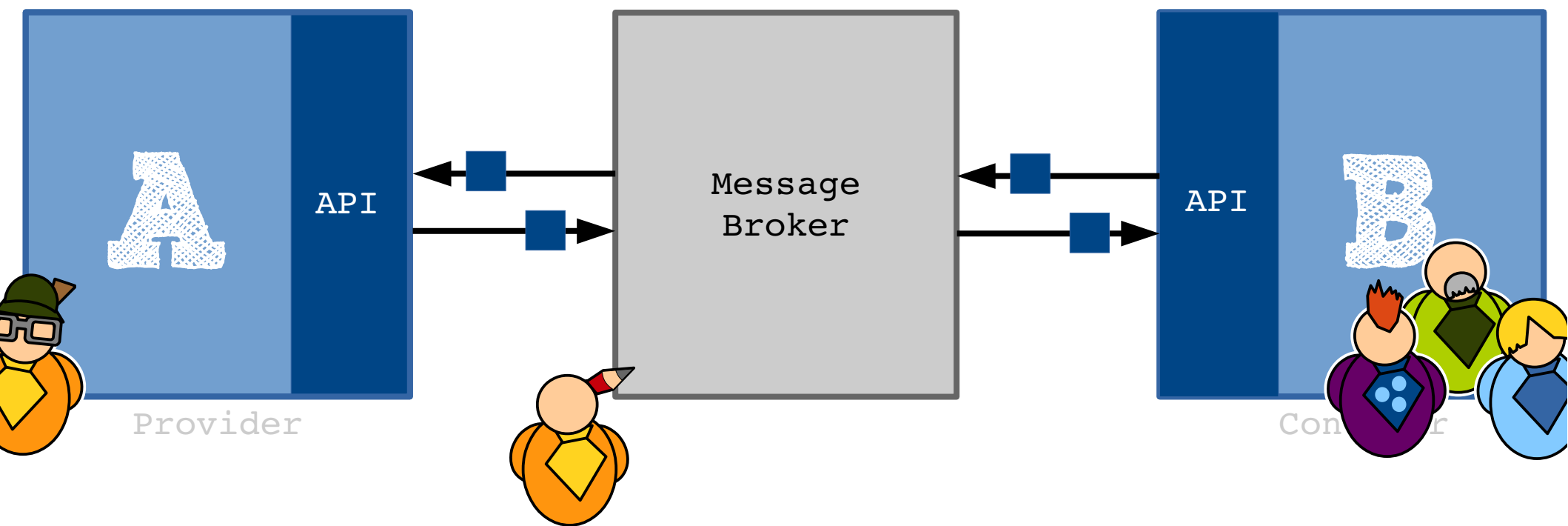
PUT: RESTBUCKS.io/v2/order/1/payment  
RESTBUCKS.io/v2/order/{id}/payment



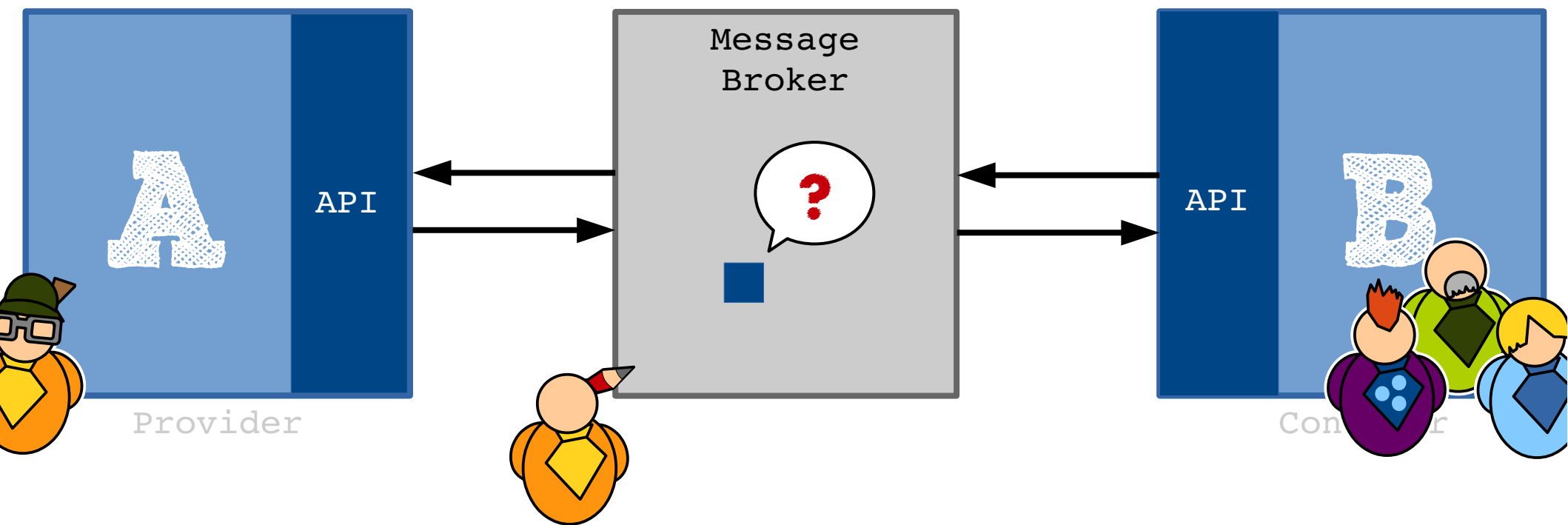


B kennt A

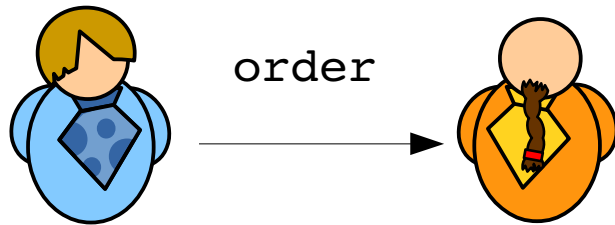




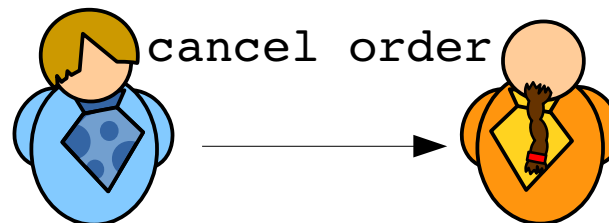
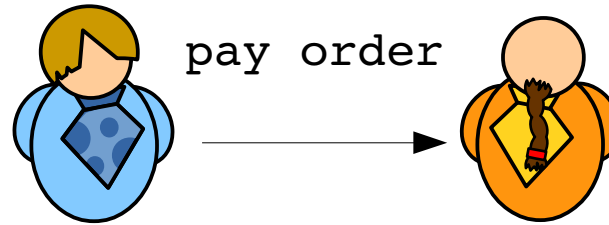
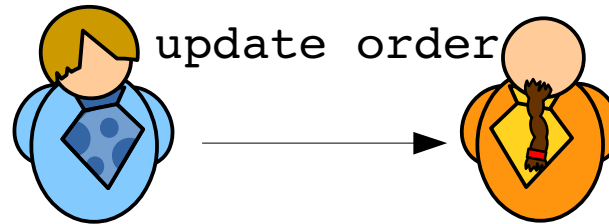




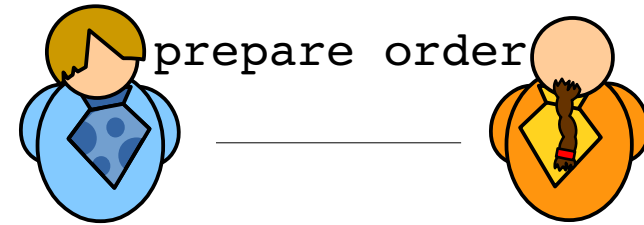
# SCHRITT 1



# SCHRITT 2



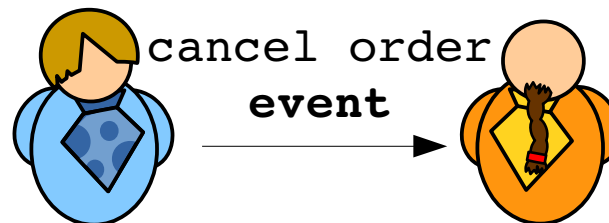
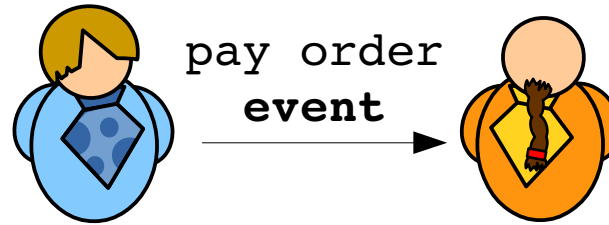
# SCHRITT 3



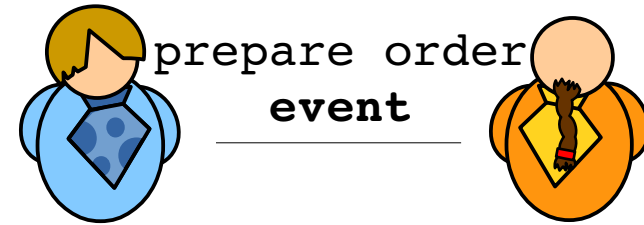
# SCHRITT 1



# SCHRITT 2

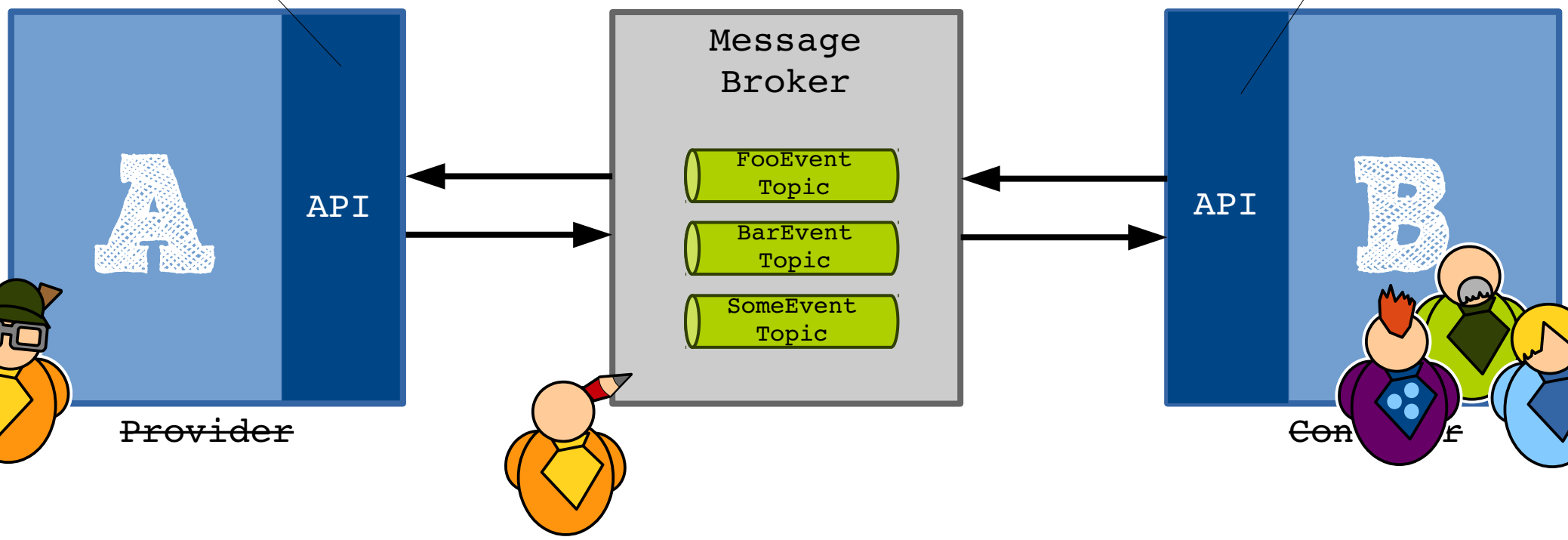


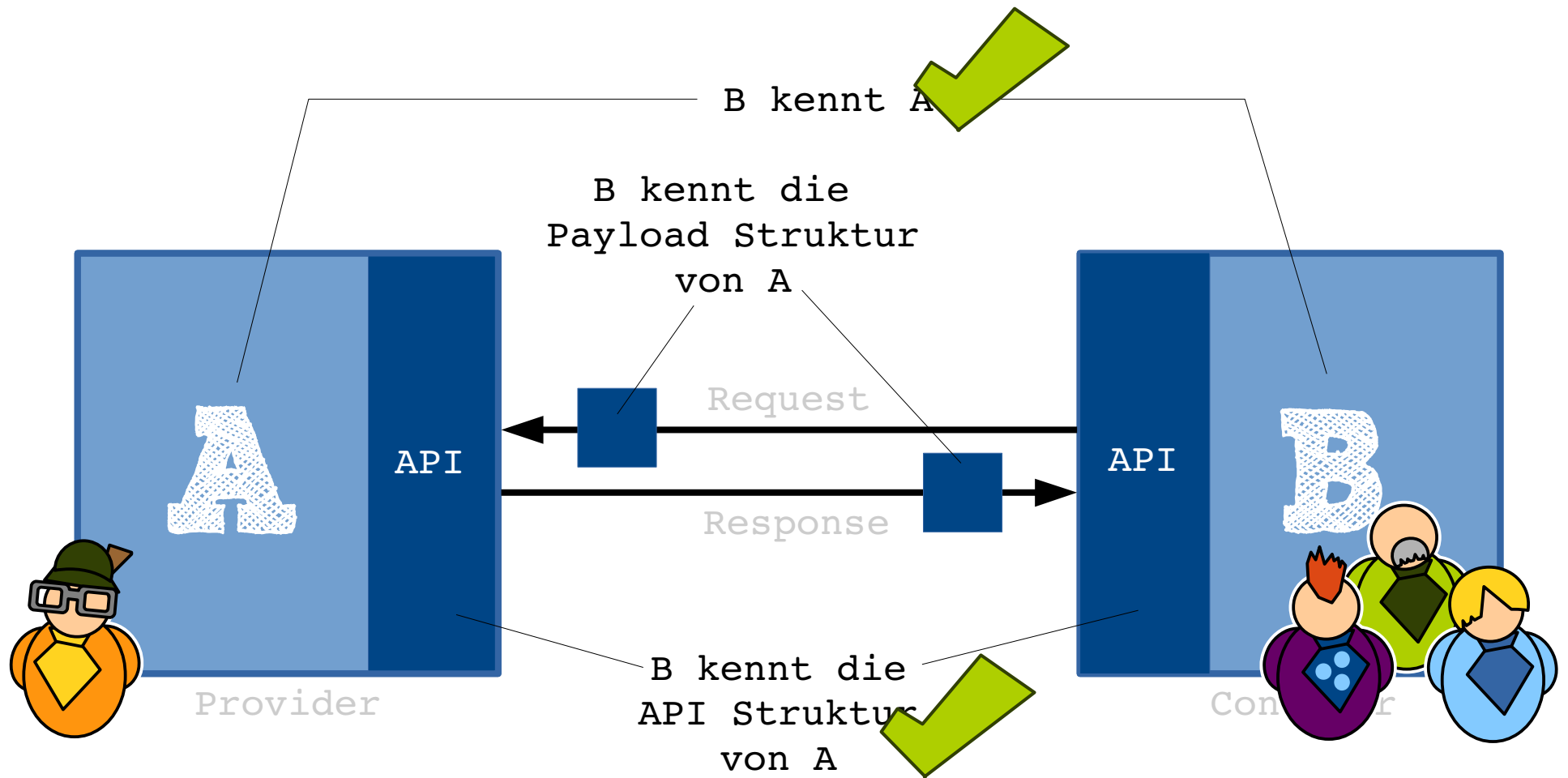
# SCHRITT 3



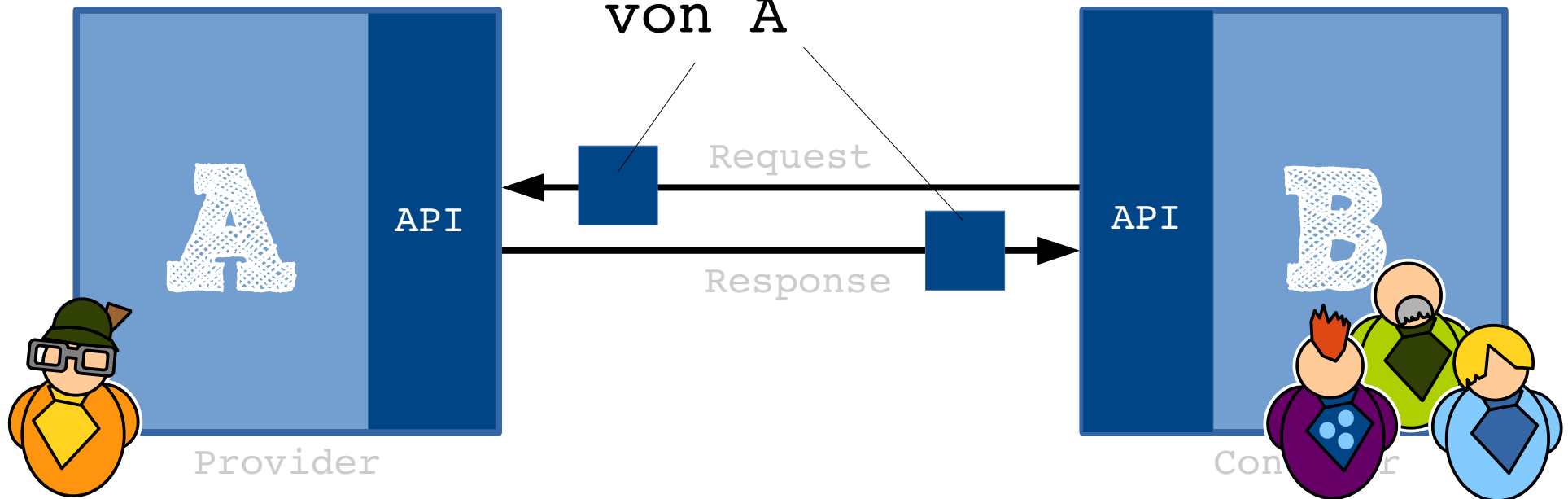
FooEvent-Listener  
SomeEvent-Listener

BarEvent-Listener  
SomeEvent-Listener





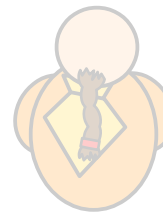
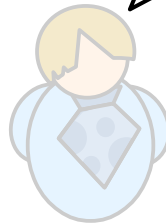
B kennt die  
Payload Struktur  
von A



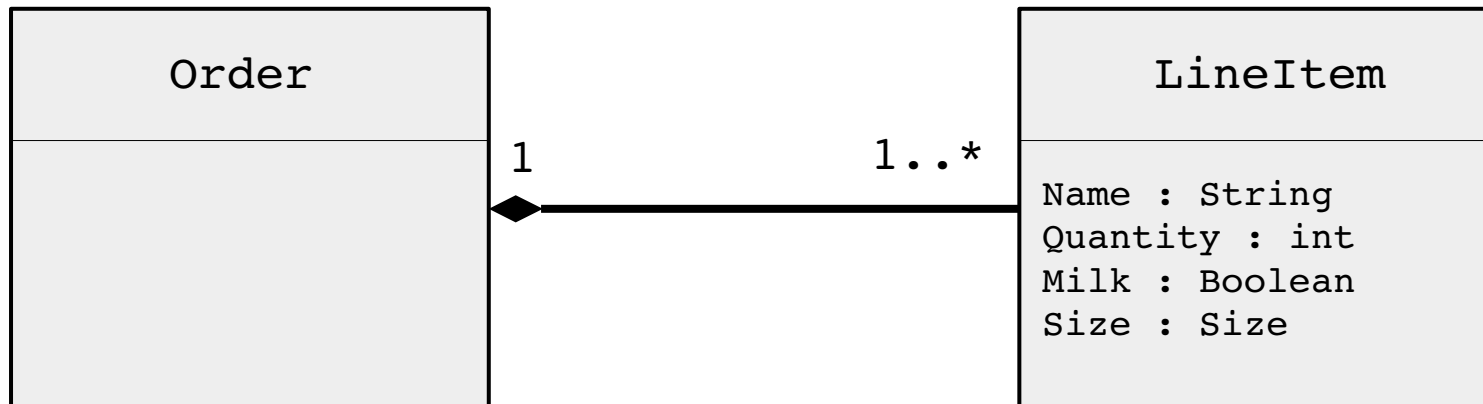


Order

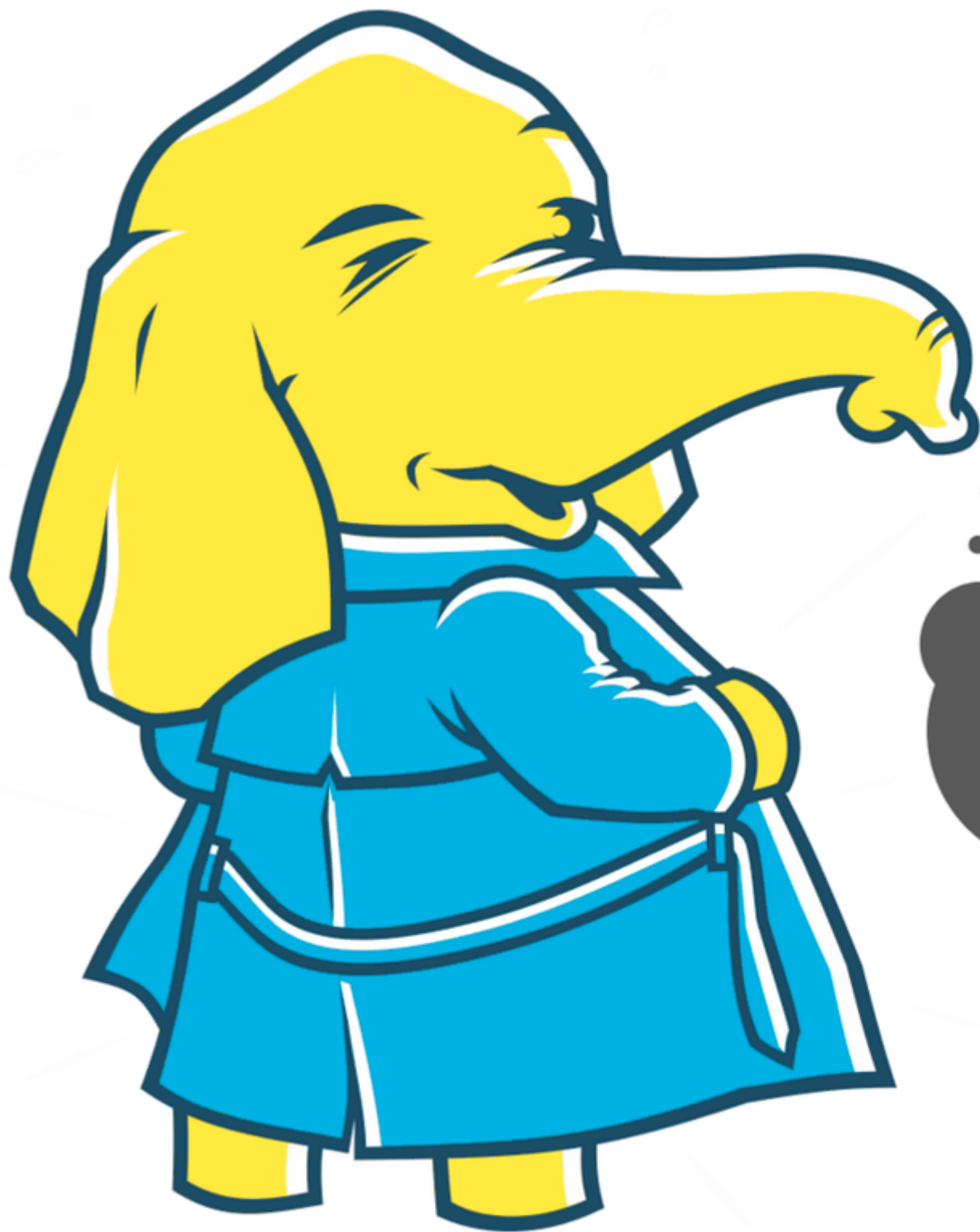
LineItem



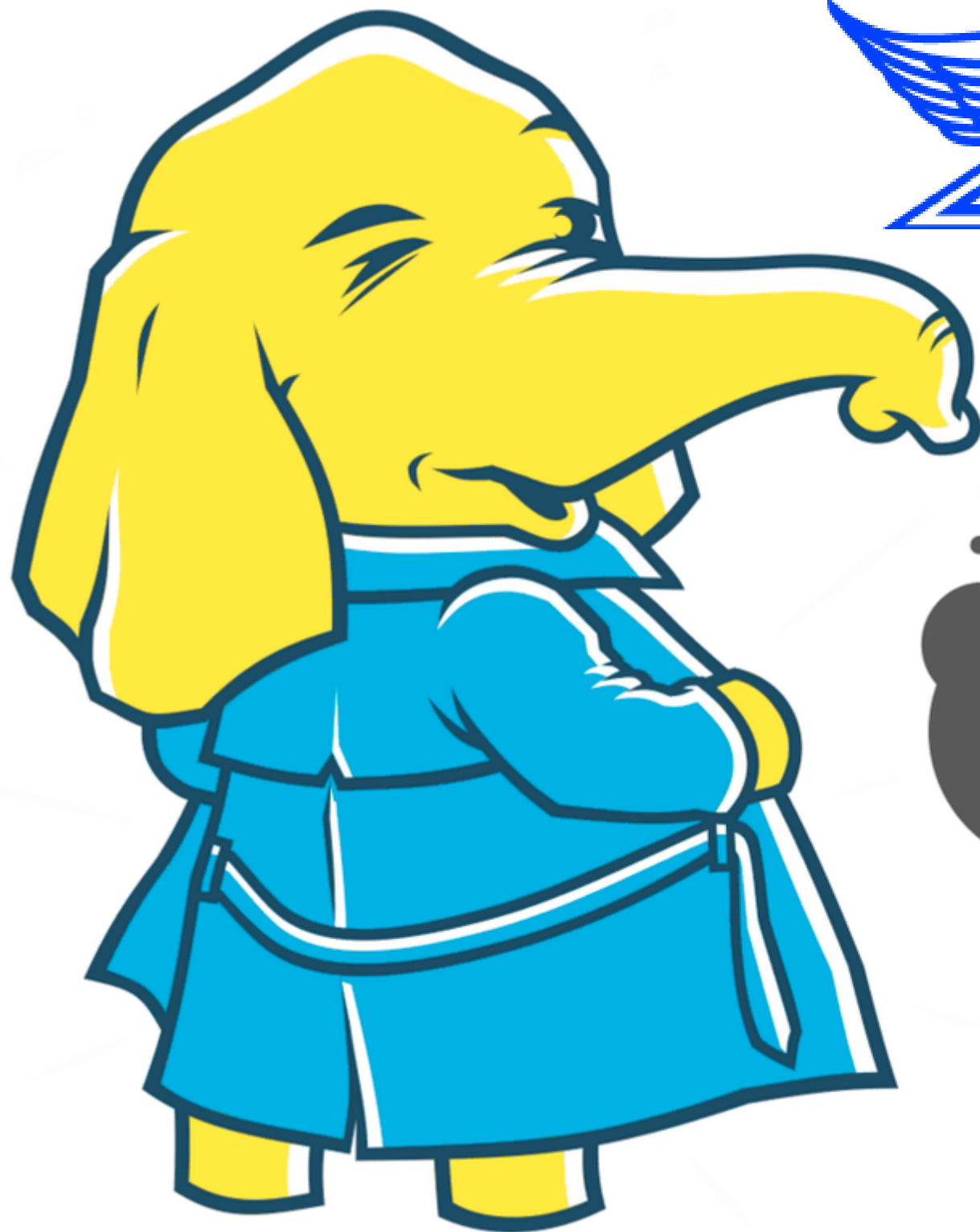




# SCHEMA EVOLUTION

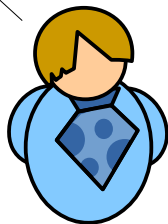


MY HADOOP  
IS **BIGGER**  
THAN YOURS...

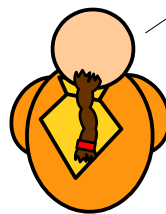


MY HADOOP  
IS **BIGGER**  
THAN YOURS...

Writer

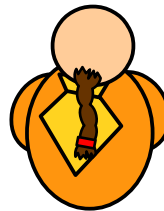
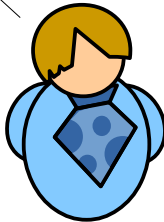


Reader

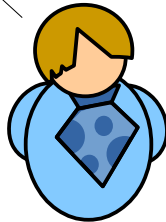
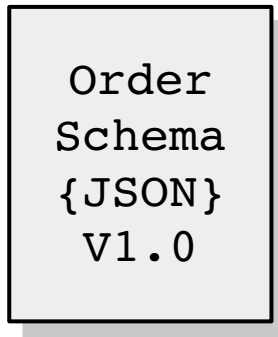


Writer

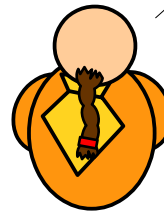
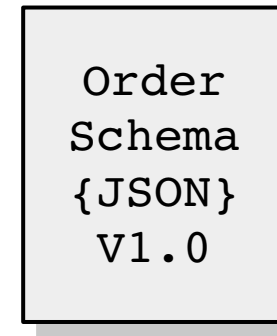
Order  
Schema  
{JSON}  
V1.0



Writer



Reader



Version 1.0

```
[
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "LineItem",
    "fields": [
      {"name": "name", "type": "string",
        "default": "coffee"},
      {"name": "quantity", "type": "int",
        "default": "1"},
      {"name": "milk", "type": "boolean",
        "default": "false"}
    ]
  },
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "Order",
    "fields": [
      {"name": "item", "type": "LineItem"}
    ]
  }
]
```



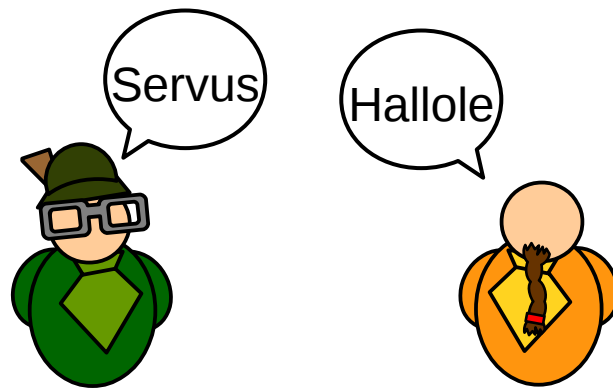
```
[
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "LineItem",
    "fields": [
      {"name": "name", "type": "string",
        "default": "coffee"},
      {"name": "quantity", "type": "int",
        "default": "1"},
      {"name": "milk", "type": "boolean",
        "default": "false"}
    ]
  },
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "Order",
    "fields": [
      {"name": "item", "type": "LineItem"}
    ]
  }
]
```

LineItem Type

Version 1.0

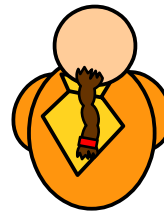
```
[
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "LineItem",
    "fields": [
      {"name": "name", "type": "string",
        "default": "coffee"},
      {"name": "quantity", "type": "int",
        "default": "1"},
      {"name": "milk", "type": "boolean",
        "default": "false"}
    ]
  },
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "Order",
    "fields": [
      {"name": "item", "type": "LineItem"}
    ]
  }
]
```

Order Type



Writer

Order  
Schema  
{JSON}  
**v2.0**



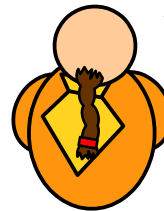
Writer

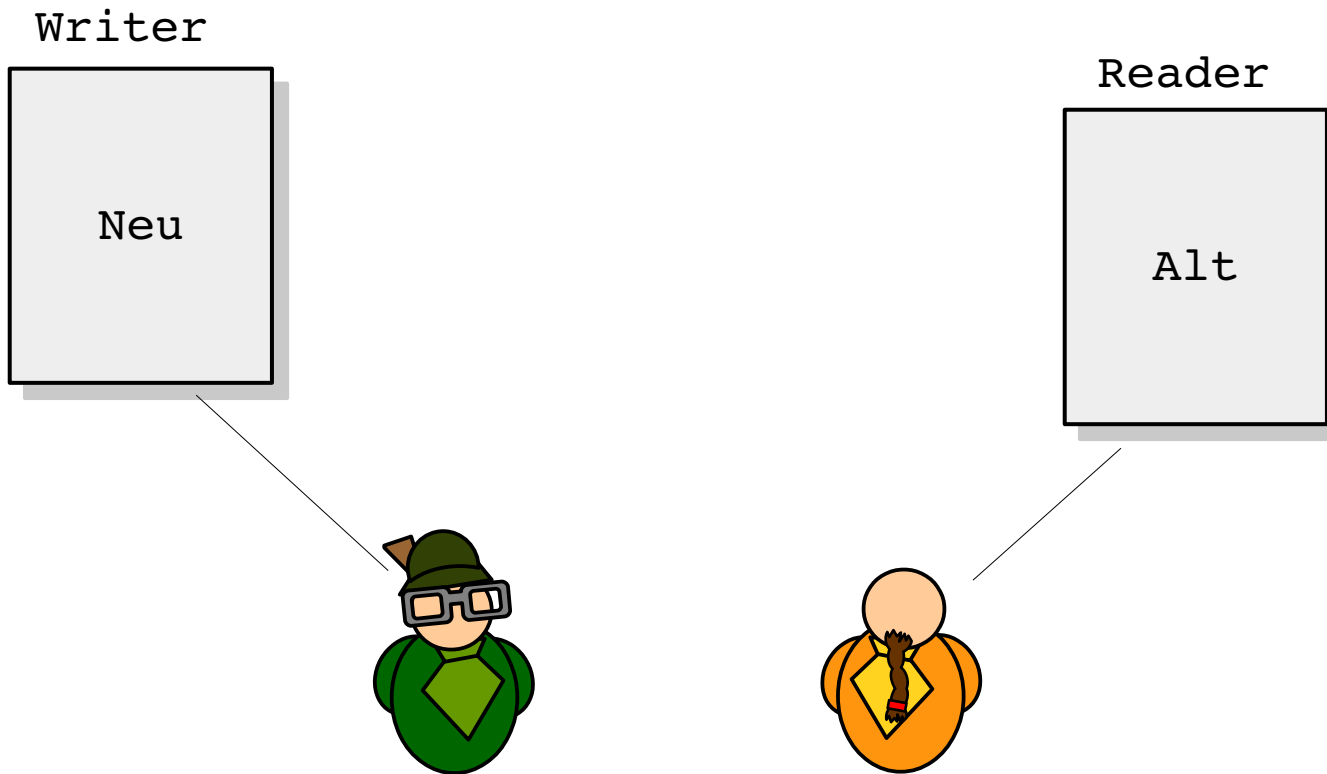
Order  
Schema  
{JSON}  
**V2.0**



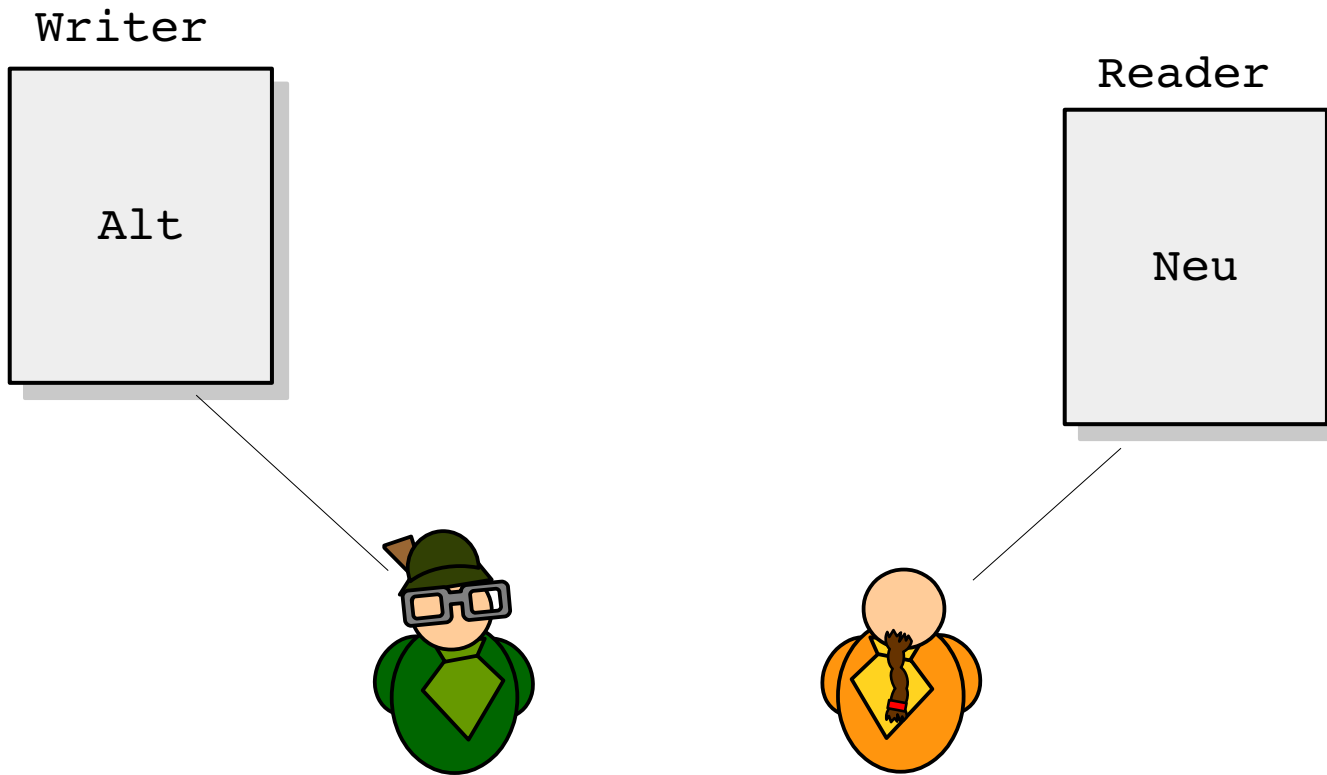
Reader

Order  
Schema  
{JSON}  
**V1.0**





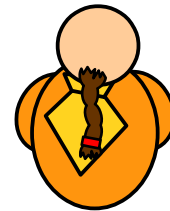
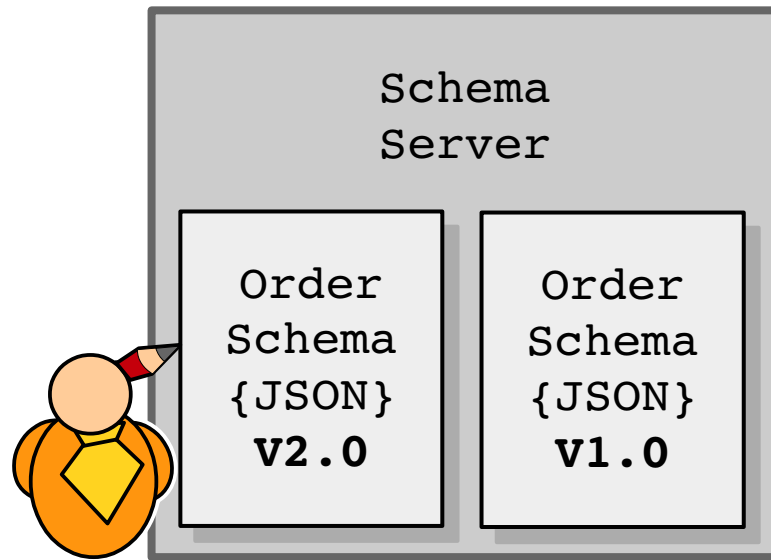
Vorwärtskompatibilität

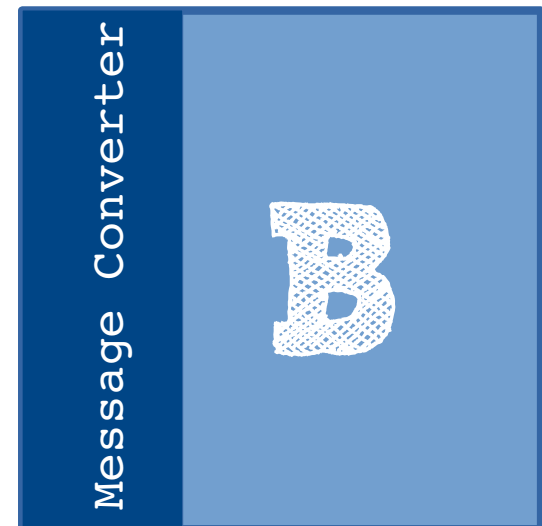
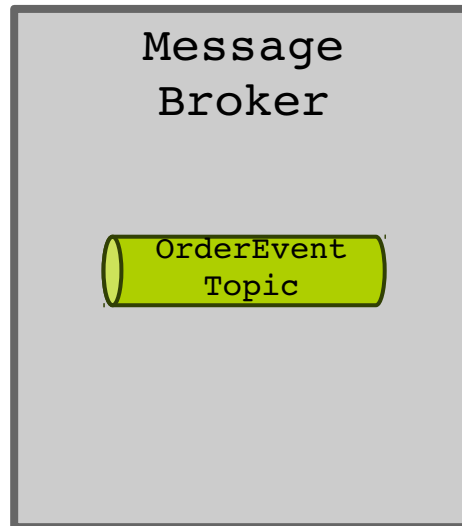
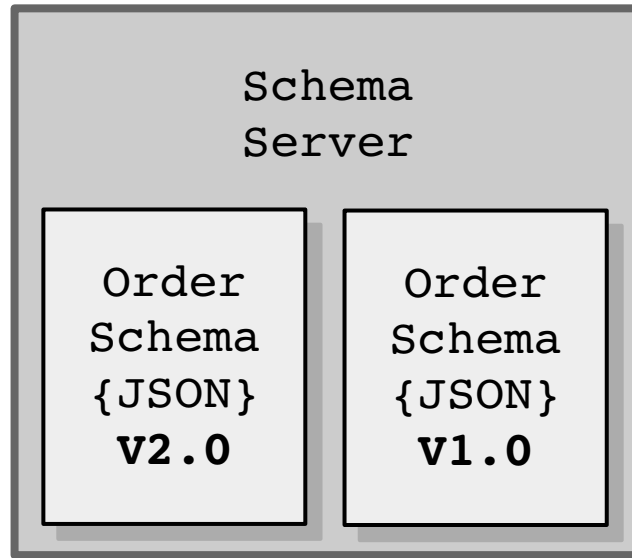


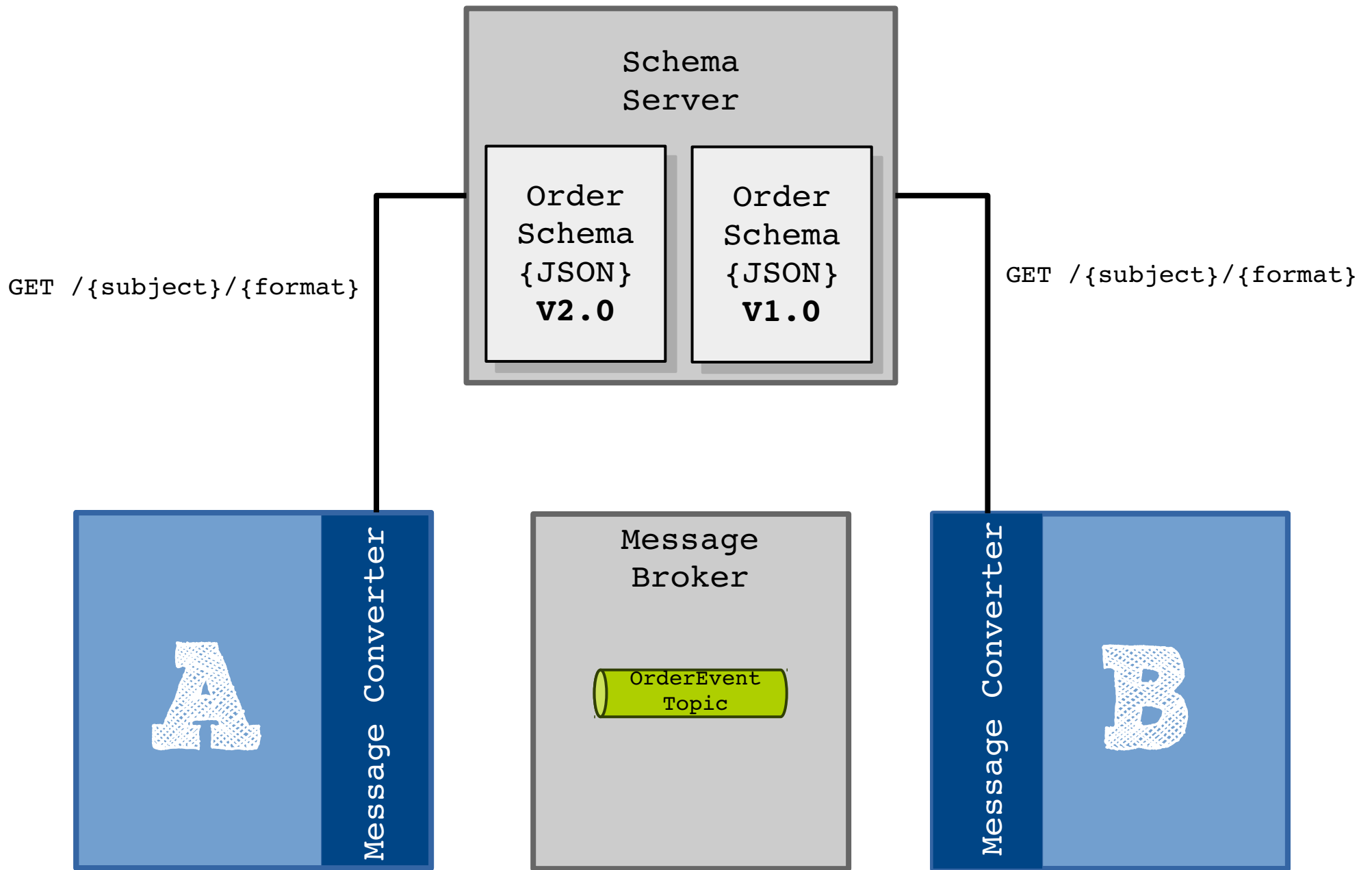
Rückwärtskompatibilität

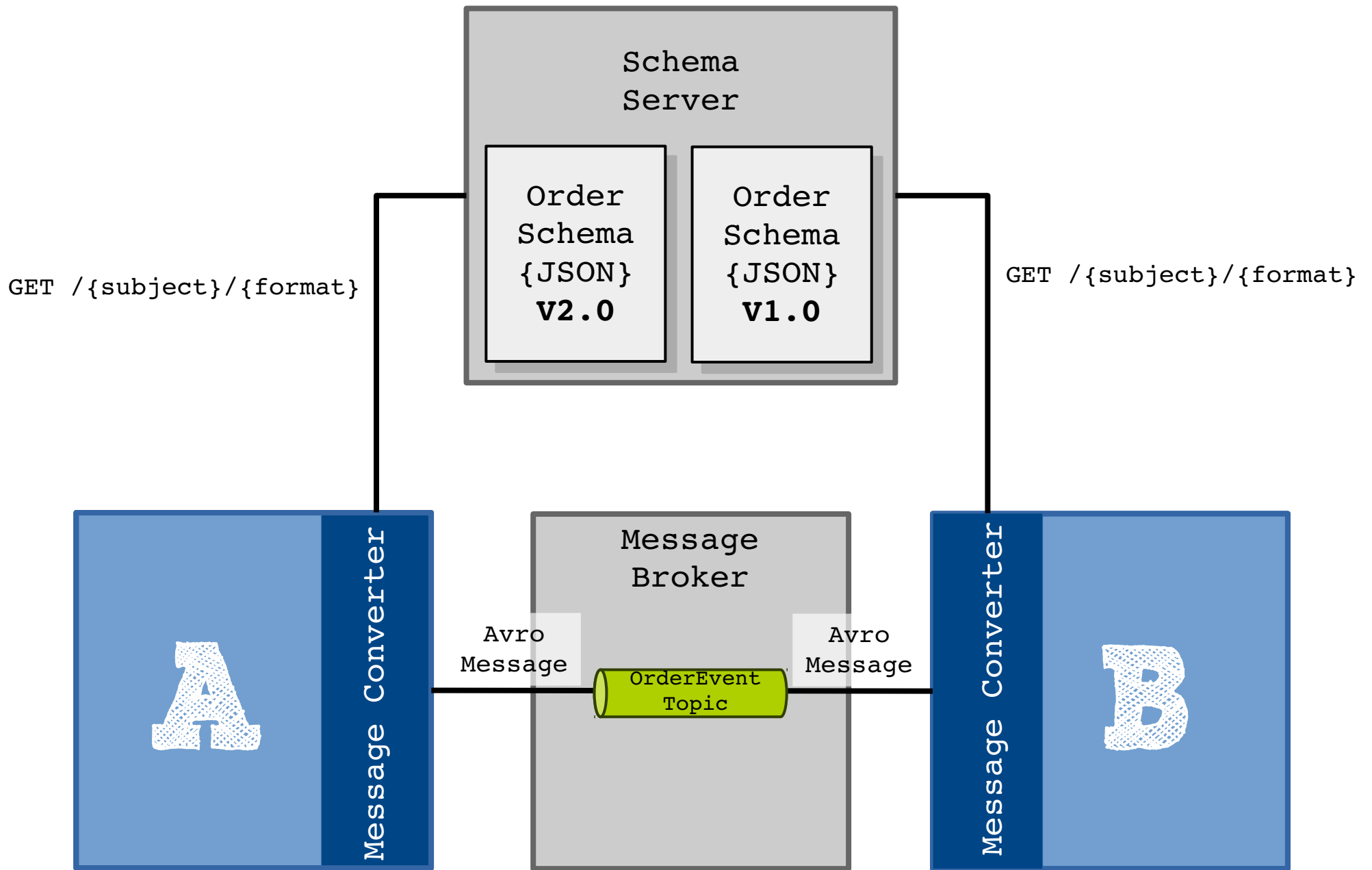
```
[
  {
    "namespace": "de.muenchen.jfs17",
    "type": "enum",
    "name": "Sizes",
    "symbols": ["LARGE", "MIDDLE", "SMALL"]
  },
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "RechnungsPosten",
    "aliases": ["LineItem"],
    "fields": [
      {"name": "name", "type": "string",
        "default": "coffee"},
      {"name": "quantity", "type": "int",
        "default": "1"},
      {"name": "milk", "type": "boolean",
        "default": "false"},
      {"name": "size", "type": "Sizes",
        "default": "MIDDLE"}
    ]
  },
  {
    "namespace": "de.muenchen.jfs17",
    "type": "record",
    "name": "Bestellung",
    "aliases": ["Order"],
    "fields": [
      {"name": "item", "type": "RechnungsPosten"}
    ]
  }
]
```

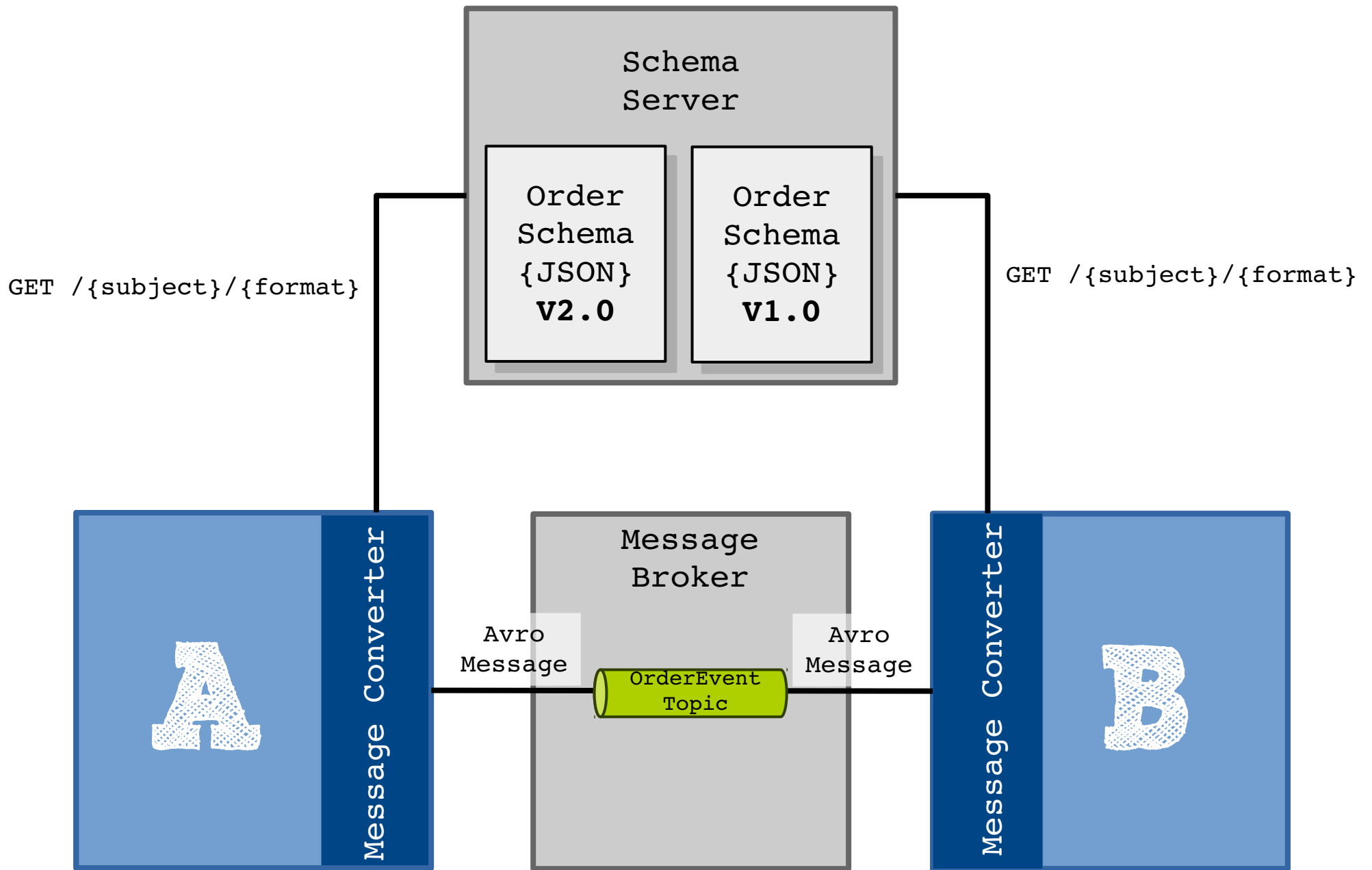












<https://cloud.spring.io/spring-cloud-stream/>

O'REILLY®

# Designing Data-Intensive Applications

THE BIG IDEAS BEHIND RELIABLE, SCALABLE,  
AND MAINTAINABLE SYSTEMS



Martin Kleppmann



# DEMO

[https://github.com/xdoo/schema\\_evolution\\_demo](https://github.com/xdoo/schema_evolution_demo)

Wenn Schnittstelle x

Sicher | <https://speakerdeck.com/xdoo/wenn-schnittstellen-alt-werden-dot-dot-dot-jfs-2017>

Speaker Deck Search... Browse Upload Claus Straube

Presentation updated successfully.

Speaker Deck Published on Jul 4, 2017



Wenn  
**SCHNITTSTELLEN**  
alt werden...

claus.straube@muenchen.de

Claus Straube  
3 Presentations

★ Star this Talk 0 Stars

Published in Programming

Stats 0 Views

Edit this presentation

Share

Twitter, Facebook

Embed

Direct Link

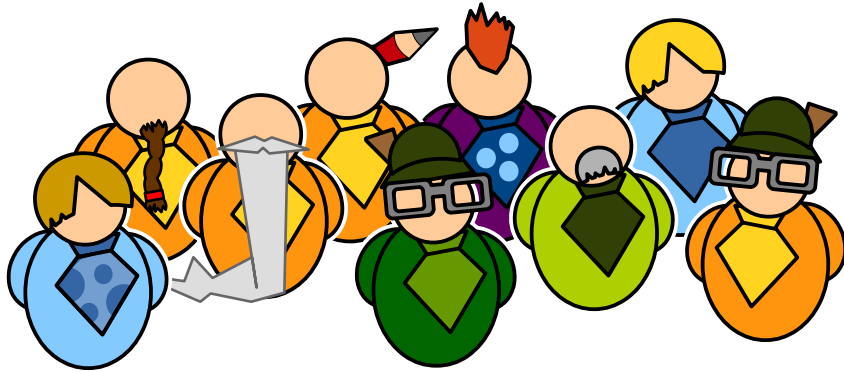
Download PDF

Wenn Schnittstellen alt werden... (JFS 2017)  
by Claus Straube  
Published July 4, 2017 in Programming

Folien zum Vortrag auf dem Java Forum Stuttgart 2017

<https://speakerdeck.com/xdoo>





**HERZLICHEN DANK  
FÜR DIE  
AUFMERKSAMHEIT !**

Claus Straube

Landeshauptstadt München  
it@M

Agnes-Pockels-Bogen 21  
80992 München

[claus.straube@muenchen.de](mailto:claus.straube@muenchen.de)



@clausstraube