Protocol Bindings Additional protocol-specific information you can add to AsyncAPI document Binding - Channel Binding - Operation # ... (redacted for brevity) # ... (redacted for brevity) channels: operations : userSignup: userSignedUp : address : user_signedup action : receive channel: messages: \$ref : '#/channels/userSignedUp' userSignedUp : \$ref : '#/components/messages/userSignedUp' bindings : bindings: mqtt: kafka: qos : 2 bindingVersion : '0.5.0' retain : true bindingVersion: 0.2.0 partitions : 10 replicas : 2 topicConfiguration : cleanup.policy : ["delete", "compact"] Binding - Server Binding - Message # ... (redacted for brevity) # ... (redacted for brevity) servers : components: production: messages: bindings: userSignedUp : payload: pulsar: \$ref : '#/components/schemas/userSignedUp' tenant: contoso bindings: bindingVersion: '0.1.0' amqp: contentEncoding : gzip messageType : 'user.signup' bindingVersion: 0.3.0

Multi Format Schema

• By default you can use AsyncAPI Schema that is a superset of JSON Schema Draft 7. Multi format allows you to use other formats, like OpenAPI Schema, Apache Avro, Protobuf and others.

```
Avro Example
                              Default - AsyncAPI Schema
                                                                                  # ... (redacted for brevity)
 # ... (redacted for brevity)
                                                                                   components:
 components:
                                                                                    schemas:
   schemas:
                                                                                      userSignedUp :
     userSignedUp :
                                                                                        schemaFormat : 'application/vnd.apache.avro;version=1.9.0'
       type : object
                                                                                        schema:
       properties:
                                                                                         type: record
         userId :
                                                                                         name: UserSignedUp
           type : integer
                                                                                         namespace: com.company
           description: This property describes the id of the user
                                                                                         doc: User sign-up information
         userEmail:
                                                                                         fields:
           type : string
           description : This property describes the email of the user
                                                                                            - name: userId
                                                                                            type: int
                                                                                            - name: userEmail
                                                                                             type: string
                                                                                                              Extensibility
                             Protobuf Example
                                                                                           • With x- prefix you can add additional fields to AsyncAPI
# ... (redacted for brevity)
```

Reusability

message UserSignedUp {

required int32 user_id = 1;

optional string user_email = 2;

components:

schemas:

userSignedUp :

schema : |

• You can reuse all parts of AsyncAPI, within one document or by pointing to external resources, separate files or servers that for example can host schemas.

```
# ... (redacted for brevity)
components:
  messages:
   UserSignedUp :
     payload:
       $ref: '#/components/schemas/UserSignedUp'
  schemas:
   UserSignedUp :
      $ref: http://localhost:8080/apis/registry/v2/groups/my-group/artifacts/UserSignedUp
```

Single document

schemaFormat: 'application/vnd.google.protobuf;version=3'

```
# ... (redacted for brevity)
components:
  messages:
    UserSignedUp :
      payload:
       $ref : '#/components/schemas/UserSignedUp'
  schemas:
    UserSignedUp :
      type : object
      properties :
        displayName :
         type : string
          description : Name of the user
        email:
         type : string
         format : email
description : Email of the user
```

To a file

asyncapi: 3.0.0

version: 0.1.0

title: Cool Example

```
# ... (redacted for brevity)
components:
  messages:
   UserSignedUp :
     payload:
       $ref: '#/components/schemas/UserSignedUp'
  schemas:
    UserSignedUp :
     $ref : './userSchema.json'
```

documents to fulfil your specific use cases

x-linkedin: 'https://www.linkedin.com/company/asyncapi'