mydlink Open API Specification

Part: Cloud Recorder

Version 1.0.0

Revision History

Version	Publish Date	Author	Descriptions
1.0.0	Nov. 30, 2015	Niyazi Makuloglu	Billing for CloudNVR

Cloud Recorder Service

This doc describes the APIs for clients to manage content stored in the cloud, including the events and record clips.

1. Playback APIs

1.1 Initiate playlist session

To get video playlist, client shall invoke this API first to initiate the playback session.

```
POST /me/nvr/list/initiate?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123", # mydlink number of the target device
    "start_ts": 1412560393000,# interested time point, in milliseconds
    "end_ts": 1412560500000   # optional field.
                              # - end_ts is set to start_ts + 10min (at max) if not
                              #   specified.(VOD Mode)
                              # - if current_time - start_ts < 3 * ts clip duration,
                              #   it becomes a liveview mode. (endless stream)
                              #   (Live Mode)
                              # - if end_ts is specified its Event Mode
                              # - if there are clips not available during the
                              #   period, the end_ts will be set to the first
                              #   discontinuous point.
  }
}
```

If the given information is correct and user also subscribes cnvr service on this device, mydlink platform responds session key and the actual start timestamp of the querying video stream.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "mydlink_id": "44440123",
    "session": "e69d649bb49f4a598feceba63a5ccacb",  # session for video playlist
    "start_ts": 1412560400000                        # actual start time of video, in
                                                     # milliseconds
  }
}
```

1.2 Fetch playlist

Client shall passing the specified URL, with SESSION_ID returned in 6.1 to fetch the contents of video playlist.

```
GET /me/nvr/list/video.m3u8?session=[SESSION_ID] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json
```

mydlink platform returns clips info if the given session is correct.

```
HTTP/1.1 200 OK
Content-Type: application/vnd.apple.mpegurl
Cache-Control: no-store

#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:6,
<URL of the clip>
#EXTINF:6,
<URL of the clip>
...
```

1.3 Query video availability in the specified period

The API is for client to query the clip availability in the given time frames. Note that the range of value start_ts and end_ts shall not exceed 24 hours in a request.

```
POST /me/nvr/info/timeline?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

"data": {
  "mydlink_id": "44440123",     # target device id
  "start_ts": 1412560400000,    # start time of the interested range
  "end_ts": 1412561400000       # end time of the interested range
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

"data": {
  "mydlink_id": "44440123",
  "info": [
    [ 1412560413000, 1412560908000 ], # continuous range 1, in millisecond
    [ 1412560980000, 1412561400000 ]  # continuous range 2, in millisecond
  ]
}
```

If there are no records in the given time range, the info field in the response will be an empty array.

1.4 Query preview info

The API provides functionality for client to get preview image of a device in the given timestamp. The time range of the info shall be provided, and most N records return per query. Client shall adjust the given time range for the desired records.

```
POST /me/nvr/info/preview?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123",    # mydlink number of the target device
    "start_ts": 1412560400000,   # start time of the interested range
    "end_ts": 1412561400000      # end time of the interested range
    "sort": "asc"                # "asc" or "desc" for incremental or
                                 # decremental sorting
    "type": "s"
  }
}
```

Once the request is validated, mydlink platform returns the actual time and the path of each recording.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "mydlink_id": "44440123",
    "start_ts": 1412560410000,
    "end_ts": 1412560495000,
    "has_more": false,                      # false: all records are returned
                                            # true: client should adjust
                                            #       start/end time to get
                                            #       remaining records.
    "list": [
      {
        "timestamp": 1412560410000,         # timestamp of the preview
        "path": "<URL of preview image>"    # preview image path
      },
      {
        "timestamp": 1412560495000,         # timestamp of the preview
        "path": "<URL of preview image>"    # preview image path
      }
    ]
}
```

1.5 Query snapshot
Client is able to query the snapshot of a device via timestamp. Client shall carry mydlink number of the target device, target timestamp in milliseconds, and a valid access_token.

```
GET
/me/nvr/info/snapshot?mydlink_id=[MYDLINK_NUMBER]&timestamp=[TIMESTAMP]&access_token
=[ACCESS_TOKEN]&type=[IMAGE_TYPE] HTTP/1.1
```

```
Host: api.mydlink.com
IMAGE_TYPE:
  "s" : original image
  "ss": small image
```

Once the request is validated, mydlink platform redirects client to the location of target snapshot.

```
HTTP/1.1 302 Found
Location: <TARGET LOCATION OF THE SNAPSHOT>
```

If no snapshot of the device in the specified timestamp, error code '30' will be responded.

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store

{
  "error": {
    "type": "NVR",
    "code": 30,
    "message": "No such record."
  }
}
```

1.6 Query storyboard info

The API provides functionality for client to get storyboard image of a device in the given timestamp. The time range of the info shall be provided, and most N records returned per query. (N=30). Client shall adjust the given time range for the desired records.

```
POST /me/nvr/info/storyboard?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123",    # mydlink number of the target device
    "start_ts": 1412560400000,   # start time of the interested range
    "end_ts": 1412561400000,     # end time of the interested range
    "sort": "asc"                # "asc" or "desc" for incremental or
                                 # decremental sorting
  }
}
```

Once the request is validated, mydlink platform returns the actual time and the path of each recording.

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
Cache-Control: no-store

{
  "data":{
    "mydlink_id": "44440123",
    "start_ts": 1412560410000,
    "end_ts": 1412560495000,
    "has_more": false,                     # false: all records are returned
                                           # true: client should adjust start/end
                                           # time to get remaining records.

    "list": [
      {
        "start_ts": 1412560410000,         # start timestamp of the storyboard
        "end_ts": 1412560410000,           # end timestamp of the storyboard
        "path": "<URL of storyboard image>", # storyboard image path
        "blank_list": "[1,2,3,23,30]",     # array of index of blank image, index
                                           # start from 0
        "attribute": "5,10,400,300,6"      # #1: Number of units in horizontal,
                                           # #2: Number of units in vertical,
                                           # #3: pixels in width a unit
                                           # #4: pixels in height a unit
                                           # #5: interval (seconds) between units
      },
      {
        "start_ts": 1412560410000,         # start timestamp of the storyboard
        "end_ts": 1412560410000,           # end timestamp of the storyboard
        "path": "<URL of storyboard image>", # storyboard image path
        "blank_list": "[1,2,3,23,30]",     # array of index of blank image, index
                                           # start from 0
        "attribute": "5,10,400,300,6"
      }


}
```

2. Clip Management

2.1 Save clip

Client is able to save clip with the given time range persistently. This kind of clips will be moved to separate place and won't expire with subscribed plan.

```
POST /me/nvr/clip/create?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123",                # mydlink number of the target device
    "name": "funny doggy",                    # name of the bookmark
    "range": [ 1412560393000, 1412560500000 ], # time range of the bookmark (ms)
    "type": 0,                                # 0 normal , 1 timelapse
    "timelapse": {                            # shall include when type is 1
      "length": 2                             # 0: 30s, 1: 60s, 2: 90s, 3: 120s
    }
  }
}
```

Success returns if the given fields are correct.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "mydlink_id": "44440123",
    "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # unique id of the bookmark
    "name": "funny doggy",                    # name of the bookmark
    "range": [ 1412560393000, 1412560500000 ] # time range of the bookmark (ms)
  }
}
```

2.2 Delete saved clips

Client can delete clips of the specified id to release sapce.

```
POST /me/nvr/clip/delete?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "list": [
      "abaac316a8ae46c69c1ba35af1b4fa61",  # clip id
      "1b123acd0703454c829941ee48adde82",
      "9f7c8875bf544d2f85a2e2fefe507b11",
      "a40b5d49be514b4fbb997b907b1390fd",
      "139b1055221b4e1d8247ffbc3b346378"
    ]
  }
}
```

Success returns if the given fields are correct.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "result": "success"
  }
}
```

2.3 Query clip list

The API returns all created clips of the specified devices. If no device are specified, all clips of the user shall be responded.

```
POST /me/nvr/clip/list?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": [ "44440123" ],        # mydlink number of the target device(s).
    "start_ts": 1421914355691,           # search period (from)
    "end_ts": 1421914355691,             # search period (to)
    "page": 1                            # start from 1, default 1
  }
}
```

Success returns if the request is correct.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data": {
    has_more": true                              # if has more records
    "list": [
      {
        "mydlink_id": "44440123",                # mydlink number
        "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # unique id of the clip
        "range": [ 1412560393000, 1412560500000 ],# time range of the clip (ms)
        "duration": 107,                         # clip length in seconds
        "name": "funny doggy",                   # name of the clip
        "type": 0,                               # type of the clip to be made,
                                                 # refer IV.2.5 of type definition
        "status": 0,                             # clip generate status:
                                                 #  0: in progress
                                                 #  1: done
                                                 # -1: generate failed
        "path": "",                              # URL of generated clip
        "preview": "",                           # URL of clip preview image
        "watched": true,                          # has watched the clip,
```

```
      "filesize": 0                              # filesize in kb
    },
    {
      "mydlink_id": "44440123",                  # mydlink number
      "id": "1b123acd0703454c829941ee48adde82", # unique id of the clip
      "range": [ 1412560000000, 1412560015000 ],# time range of the clip (ms)
      "duration": 15,                            # clip length in seconds
      "name": "baby smile",                      # name of the clip
      "type": 0,
      "status": 0,
      "path": "",
      "preview": "",                             # URL of clip preview image
      "watched": false,                           # not watch this clip yet
      "filesize": 0                              # filesize in kb
    }
  ]
  }
}
```

2.4 Update clip attributes

The API returns all update clip attributes of the given clip id of the user.

```
POST /me/nvr/clip/update?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # unique id of the bookmark
    "attributes": {
      "name": "funny catty",                 # name to be changing
      "watched": true                        # view status to be changed
    }
  }
}
```

Success returns if the request is correct.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "result": "success"
  }
}
```

3. Event API

3.1 Query events of a device

Client can query events of a device via this API. For a query, the time range of the events shall be provided, and most N records will be returned per query. Client shall adjust the given time range for the desired event records.

```
POST /me/device/events?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123"    # mydlink number of the target device
    "start_ts": 1404050675000,  # start time in milliseconds
    "end_ts": 1404057875000     # end time in milliseconds
    "sort": "asc"               # "asc" or "desc" for incremental or
                                # decremental sorting
  }
}
```

mydlink platform returns all events of the device within the specified time range.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data": {
    "mydlink_id": "44440123",              # mydlink number of the target device
    "start_ts": 1404050675000,
    "end_ts": 1404057875000,
    "has_more": false,                     # false: all records are returned
                                           # true: client should adjust
                                           #       start/end time to get
                                           #       remaining records.

    "list": [
      {
        "id": 100,                         # event id
        "timestamp": 1404050685849,        # event time
        "source": "<UID of sensor>",       # uid of event source (optional)
        "name": "<name of sensor>",        # name of event source (optional)
        "policy": "<name of policy>"       # name of policy (optional)
      },
      {
        "id": 101,                         # event id
        "timestamp": 1404050708578,        # event time
        "source": "<UID of sensor>",       # uid of event source
        "name": "<name of sensor>",        # name of event source
        "policy": "<name of policy>"       # name of policy
      }
    ]
  }
}
```

3.2 Query events of a user

Client can query all device events of the specified account via this API. For a query, the time range of the events shall be provided, and most N records will be returned per query. (N=30). Client shall adjust the given time range for the desired event records.

```
POST /me/user/events?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "start_ts": 1404050675000,  # start time in milliseconds
    "end_ts": 1404057875000     # end time in milliseconds
    "sort": "asc"               # "asc" or "desc" for incremental or
                                # decremental sorting
  }
}
```

mydlink platform returns all events of the user within the specified time range.
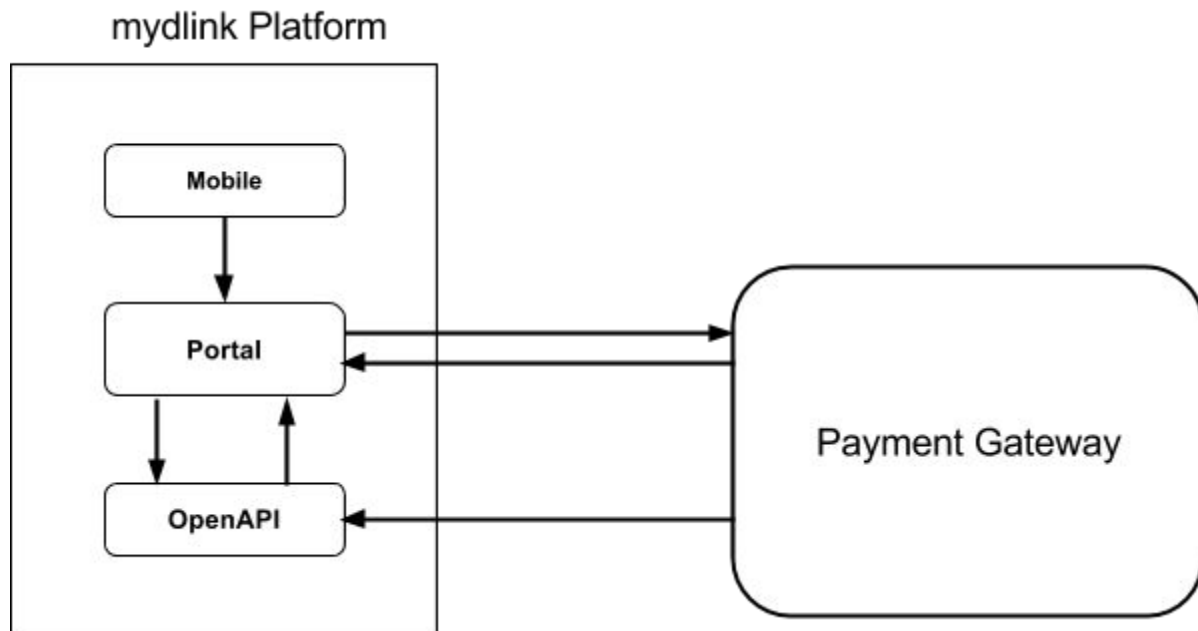
```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data": {
    "start_ts": 1404050675000,
    "end_ts": 1404057875000,
    "has_more": false,                      # false: all records are returned
                                            # true: client should adjust
                                            #       start/end time to get
                                            #       remaining records.

    "list": [
      {
        "mydlink_id": "30038291",           # mydlink number of the device
        "id": 100,                          # event id
        "timestamp": 1404050685849,         # event time
        "source": "<UID of sensor>",        # uid of event source
        "name": "<name of sensor>",         # name of event source
        "policy": "<name of policy>"        # name of policy
      },
      {
        "mydlink_id": "30038291",           # mydlink number of the device
        "id": 101,                          # event id
        "timestamp": 1404050708578,         # event time
        "source": "<UID of sensor>",        # uid of event source
        "name": "<name of sensor>",         # name of event source
        "policy": "<name of policy>"        # name of policy
      }
    ]
  }
}
```
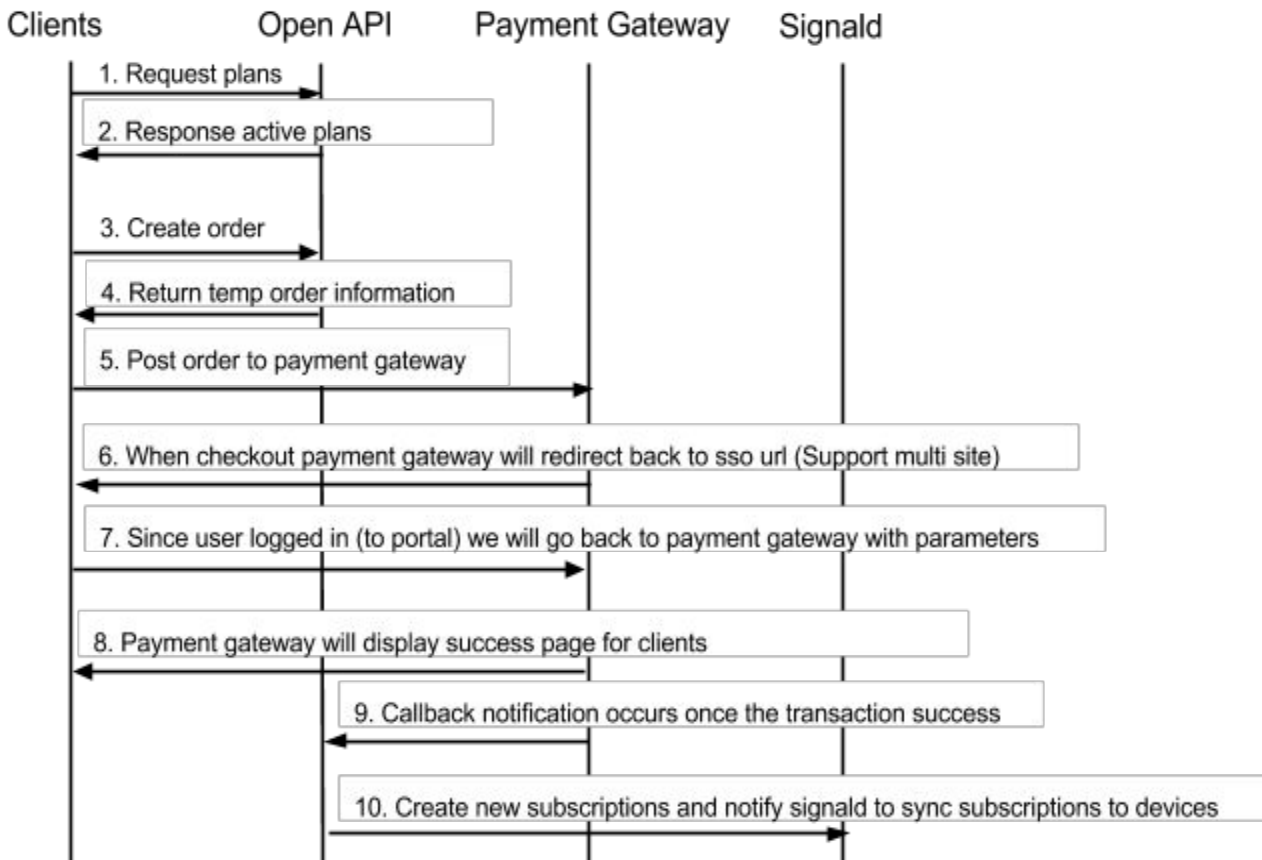
4. Billing API

A. The Architecture & Flows

● **Purchase Flow**

Clients   Open API   Payment Gateway   Signald

1. Request plans

2. Response active plans

3. Create order

4. Return temp order information

5. Post order to payment gateway

6. When checkout payment gateway will redirect back to sso url (Support multi site)

7. Since user logged in (to portal) we will go back to payment gateway with parameters

8. Payment gateway will display success page for clients

9. Callback notification occurs once the transaction success

10. Create new subscriptions and notify signald to sync subscriptions to devices

Note: The shopping cart and logic is implemented in web approach. Both Desktop and mobile browsers environment shall be supported.

1. Clients request available service plans from openapi.
2. OpenAPI returns available service plans.
3. Clients request openapi for creating a new temp order with device & service plan combinations.
4. OpenAPI generates a new order on Mydlink platform with posted devices & service plans and return generated temporary Mydlink order id.
5. Client follows the generated url to payment gateway page.
6. User clicks checkout button, payment gateway redirects user to our SSO endpoint.
7. Since user logged in, we pass user information to payment gateway to prove that it's secure. (Refer to SSO documentation pdf.)
8. Payment gateway display success page and a return link on the page. (Customizable)
9. Payment gateway notifies transaction results to OpenAPI callback url after a success transaction.
10. OpenAPI updates the order status. If the transaction is successful, OpenAPI shall sync

the device plan and device shall start the recording function to the cloud storage.

- **Free trial flow**
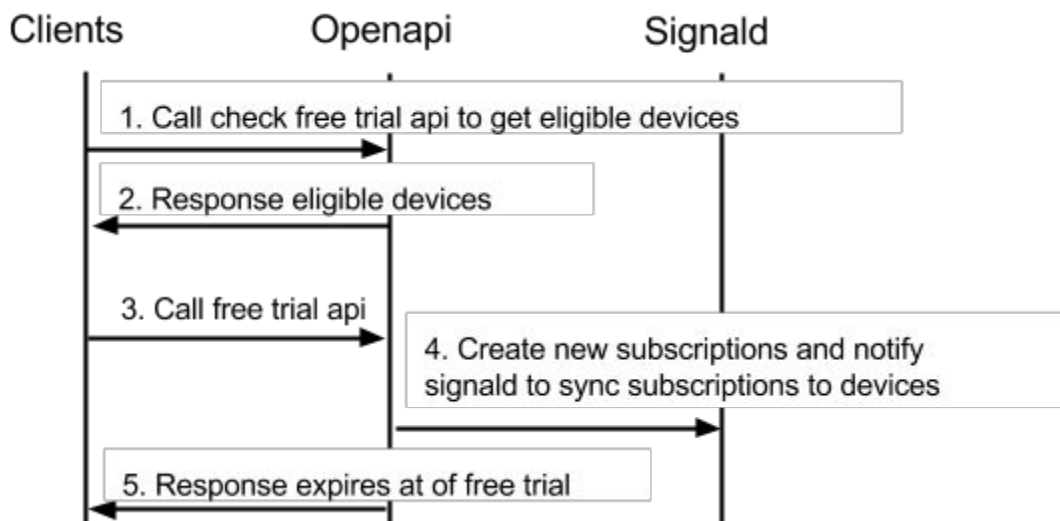


1. Clients call 4.3 api to get free trial eligible devices.
2. Openapi returns eligible devices.
3. Clients request 4.4 api for creating a free trial for devices.
4. If all the devices client request for free trial is eligible openapi will create free trial subscription and notify signald about the cnvr mode change.
5. Openapi will response expires at for free trial period.

- **Refund flow**
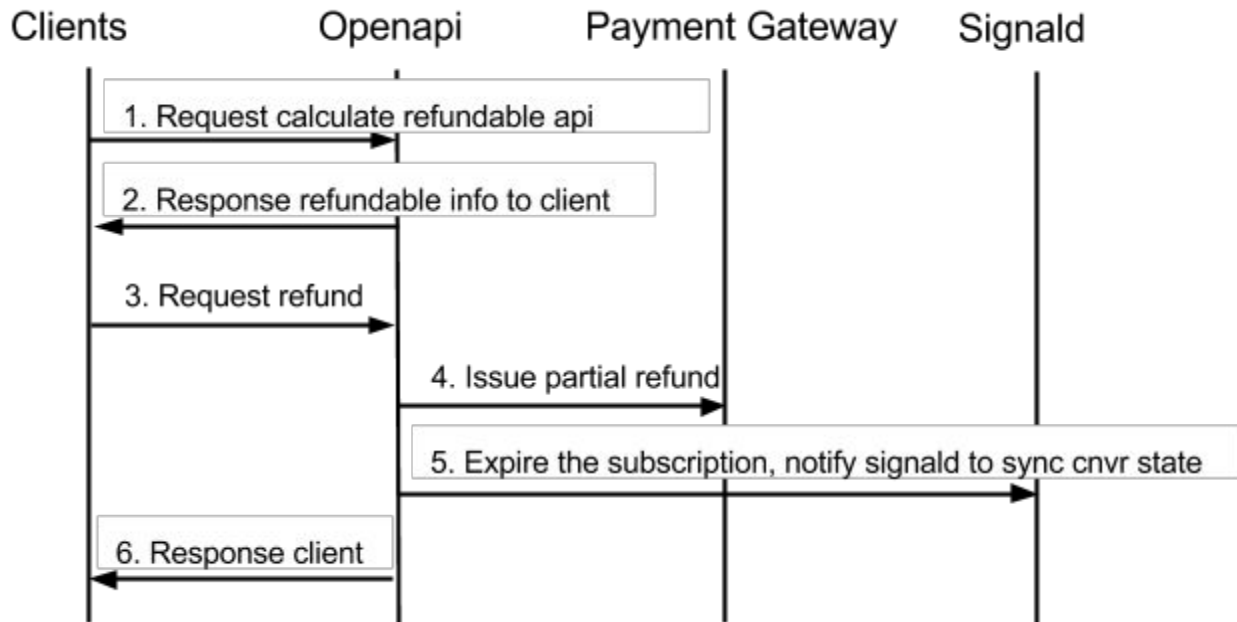
1. Clients call 4.7 api to get calculate refundable info.
2. Openapi returns calculated refundable values for the subscription.
3. Clients request 4.8 api to issue a refund.
4. Openapi request a partial refund to payment gateway.
5. Openapi expire the subscription and notify signald that cnvr state change.
6. Openapi response client if the request was success or failed.

- **Change plan flow**
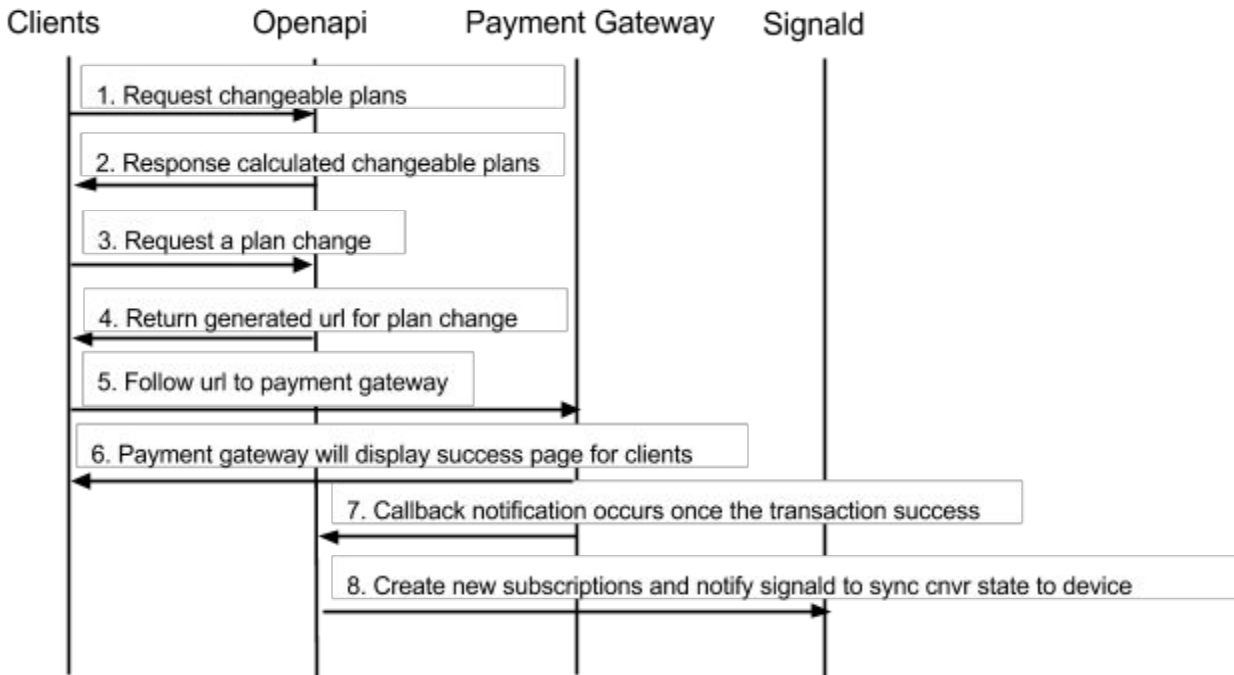
1. Clients call 4.9 api to get list of changeable plans.
2. Openapi calculates and response changeable plans for the given subscription.
3. Clients request 4.10 api to issue a plan change.
4. Openapi response a generated url for plan change.
5. Client follows the url to reach payment gateway for plan change.
6. If user complete payment flow payment gateway will display success page.
7. Callback notification for plan change will be sent from payment gateway to openapi
8. Openapi will handle plan change (expire the previous subscription, create a new subscription and order history records, notify signald about cnvr state change)

B. The Billing APIs

4.1 Get Products List
Client can get available list of cloud products in mydlink platform via the following interface with the client level access_token. Client can carry language for specific language. English will be returned as default. (refer to appendix II for language definition)

```
GET /me/billing/products?access_token=[CLIENT_ACCESS_TOKEN]&lang=en HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json
```

Note : Products will be filtered according to client id.

If the given information is correct openapi will return the list of active products in system.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":
   [
     {
       "code": "cnvr-event-7-days-monthly",
       "name": "[Monthly] 7 days cloud storage for event base",
       "price": {
         "value": 4.99,
         "currency": "USD"                    # GBP, EUR, and USD
       },
       "settings": {"mode":2,"interval":"YEA","space":7,"quota":"30"},
       "type": "cnvr"
     },
     {
       "code": "cnvr-event-30-days-monthly",
       "name": "[Monthly] 30 days cloud storage for event base",
       "price": {
         "value": 4.99,
         "currency": "USD"                    # GBP, EUR, and USD
       },
       "settings": {"mode":1,"interval":"MON","space":7,"quota":"30"},
       "type": "cnvr"
     },
     {
       "code": "cnvr-continuous-30-days-monthly",
       "name": "[Monthly] 30 days cloud storage for continuous base",
       "price": {
         "value": 4.99,
         "currency": "USD"                    # GBP, EUR, and USD
       },
       "settings": {"mode":2,"interval":"MON","space":7,"quota":"30"},
       "type": "cnvr"
     }
   ]
}
```

4.2 Initiate Order
To create an order, client shall invoke this API first to initiate the mydlink order id.

```
POST /me/billing/initiate?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json
```

```
{
  "data": {
    "cart":[
      { "mydlink_id": "44440123", "plan": "cnvr-event-7-days" },
      { "mydlink_id": "...", "plan": "..." },
      { "mydlink_id": "...", "plan": "..." }
    ],
    "lang": "en" # refers to appendix II
  }
}
```

If the given information is correct openapi will create a pending order id and url to follow to client. Clients should follow this url to reach payment gateway to proceed the payment.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "url": "https://<host of payment gateway>/<payment uri>?<session info>"
  }
}
```

Note: New request to payment gateway cart will replace the previous one.

4.3 Check Devices Eligible to Free Trial

To check which devices under the account is eligible for a free trial, client should call below api endpoint.

```
POST /me/billing/checktrial?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": [
      "44440123",  # If device array given openapi will check for specified
                   # devices. if empty openapi will check all the devices in
                   # user's device list.
      "44440124"
    ]
  }
}
```

Openapi will return free trial eligible devices to the clients.

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
Cache-Control: no-store

{
  "data": [ "44440123" ]
}
```

4.4 Create Free Trial Subscription

To create a free trial subscription, can only add once for the combination of user_id and device_id

```
POST /me/billing/trial?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id":[
      "44440123", "...", "..."
    ]
  }
}
```

If the given information is correct openapi will return the expire timestamp of the free trial period in seconds.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "expires_at": 1443744589 # timestamp in second
  }
}
```

4.5 Query Subscription

To access user subscriptions client should send a request to the below endpoint with a valid user access token.

```
POST /me/billing/subscription/list?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": [  "44440123", "44440124", "44440125", ... ],
    "lang": "en"  # refers to appendix II
  }
```

```
}
```

If the given information is correct openapi will return the list of subscriptions to client.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
{
    "data": {
        "44440123": [
            {
                "name": "[Monthly] 30 days cloud storage for event base",
                "plan": "cnvr-event-30-days",
                "state": 0,
                "start_date": 1432642609,
                "expire_date": 1433247581,
                "cancel_date": 0,
                "rebilling_price": "12.99USD",
                "settings": {
                    "mode": 1,
                    "interval": "MON",
                    "space": 7,
                    "quota": "30"
                }
            },
            {
                "name": "[Monthly] 30 days cloud storage for event base",
                "plan": "cnvr-event-7-days",
                "state": 1,
                "start_date": 1432642609,
                "expire_date": 1433247581,
                "cancel_date": 1432643509,
                "rebilling_price": "12.99USD",
                "settings": {
                    "mode": 1,
                    "interval": "MON",
                    "space": 7,
                    "quota": "30"
                }
            }
        ],
        "44440124": [
            {
                "name": "[Monthly] 30 days cloud storage for event base",
                "plan": "cnvr-event-30-days",
                "state": 0,
                "start_date": 1432642609,
                "expire_date": 1433247581,
                "cancel_date": 0,
                "rebilling_price": "12.99USD",
```

```
                    "settings": {
                        "mode": 1,
                        "interval": "MON",
                        "space": 7,
                        "quota": "30"
                    }
                },
                {
                    "name": "[Monthly] 30 days cloud storage for event base",
                    "plan": "cnvr-event-7-days",
                    "state": 1,
                    "start_date": 1432642609,
                    "expire_date": 1433247581,
                    "cancel_date": 1432643509,
                    "rebilling_price": "12.99USD",
                    "settings": {
                        "mode": 1,
                        "interval": "MON",
                        "space": 7,
                        "quota": "30"
                    }
                }
            ]
        }
    }
}
```

Notes

state: 0 = expired, 1 = active

mode: 1 = event mode, 2 = continuous mode, 11 = free mode

interval: MON = monthly , YEA= yearly, WEE= weekly

space: recording period of the plan in days

quota: clip saving quota in minute

4.6 Cancel Subscription (stop charging me)

Clients can send a request to below endpoint to cancel active subscriptions to not to charge the users after the end of their subscription period for specified devices.

```
POST /me/billing/subscription/cancel?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123",
    "plan": "cnvr-event-30-days"
  }
}
```

NOTE: Cancel active subscription since there will be only one active subscription at a time for a device.

4.7 Calculate Refundable

Clients can send a request to below endpoint to get refundable amount of money for active subscriptions of specified devices.

```
POST /me/billing/subscription/refundable?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json


{
  "data": {
    "mydlink_id": "44440123",
    "plan": "cnvr-event-30-days"
  }
}
```

If the given information is correct openapi will return the refundable info to client.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
{
  "data": {
    "refund_fee":12.4,
    "rest_fee":14.1,
    "handling_fee":1.7,
    "currency":"USD"
  }
}
```

4.8 Refund Subscription

Clients can send a request to below endpoint to refund active subscriptions for specified devices. Refund will be calculated by the usage of the plan. Unused part of the plan will be refunded with a charge of handling fee.

```
POST /me/billing/subscription/refund?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123",
    "plan": "cnvr-event-30-days"
  }
}
```

NOTE: System will issue refund for active subscriptions behind the scenes. Might take some time to take effect.

4.9 Get Changeable Plans

Client should make a query to openapi to get upgradable/downgradable plans for users current plan. Openapi will calculate the price respectively and return all the plans available for the action

```
POST /me/billing/subscription/changeable?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json


{
  "data": {
    "mydlink_id": "44440123",
    "plan": "cnvr-event-30-days",
    "lang": "en"                              # refers to appendix II
  }
}
```

If the given information is correct openapi will return the list of products in system.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "upgrade": [
      {
        "code": "cnvr-event-7-days-monthly",
        "name": "[Monthly] 7 days cloud storage for event base",
        "prices": {
          "value": 13,                 # Final price for upgrade
          "original_price": 15.99,     # Original price
          "discount_fee": 2.99,        # Amount to be discounted
          "currency": "USD"            # England : GBP,
                                       # Rest of Europe : EUR,
                                       # Other Sites: USD
        }
      },
      {
        "code": "cnvr-event-30-days-monthly",
        "name": "[Monthly] 30 days cloud storage for event base",
        "prices": {
          "value": 13, # Final price for upgrade
          "original_price": 15.99,     # Original price
          "discount_fee": 2.99,        # Amount to be discounted
          "currency": "USD"            # England : GBP,
                                       # Rest of Europe : EUR,
                                       # Other Sites: USD
        }
      },
```

```
      {
        "code": "cnvr-continuous-30-days-monthly",
        "name": "[Monthly] 30 days cloud storage for continuous base",
        "prices": {
          "value": 13, # Final price for upgrade
          "original_price": 15.99,      # Original price
          "discount_fee": 2.99,         # Amount to be discounted
          "currency": "USD"             # England : GBP,
                                        # Rest of Europe : EUR,
                                        # Other Sites: USD
        }
      }
    ],
    "downgrade": [
      {
        "code": "cnvr-event-7-days-monthly",
        "name": "[Monthly] 7 days cloud storage for event base",
        "prices": {
          "value": -1.99,               # final price for downgrade, if negative
                                        # will be refunded. If positive user have
                                        # to pay this amount.
          "original_price": 15.99, # Original price
          "handling_fee": 0.99,
          "currency": "USD"             # GBP, EUR, and USD
        }
      },
      {
        "code": "cnvr-event-30-days-monthly",
        "name": "[Monthly] 30 days cloud storage for event base",
        "prices": {
          "value": -1.99,               # final price for downgrade, if negative
                                        # will be refunded. If positive user have
                                        # to pay this amount.
          "original_price": 15.99, # Original price
          "handling_fee": 0.99,
          "currency": "USD"             # GBP, EUR, and USD
        }
      },
      {
        "code": "cnvr-continuous-30-days-monthly",
        "name": "[Monthly] 30 days cloud storage for continuous base",
        "prices": {
          "value": 1.99,                # final price for downgrade, if negative
                                        # will be refunded. If positive user have
                                        # to pay this amount.
          "original_price": 15.99, # Original price
          "handling_fee": 0.99,
          "currency": "USD"             # GBP, EUR, and USD
          }
        }
```

```
        ]
    }
}
```

4.10 Change Plan

To change plan, client shall invoke this API first to initiate the temp mydlink order id.

```
POST /me/billing/changeplan?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123",
    "current_plan": "cnvr-event-30-days",
    "new_plan": "cnvr-event-7-days",
    "lang": "en"                             # refers to appendix II
  }
}
```

If the given information is correct openapi will create a pending order id and url to follow to client. Clients should follow this url to reach payment gateway to proceed the payment.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "data":{
    "url": "[PAYMENT_CART_URL]"
  }
}
```

Note: New request to payment gateway cart will replace the previous one.

4.11 Get Order History

To access user order history, client should send a request to the below endpoint with a valid user access token.

```
POST /me/billing/orders?access_token=[ACCESS_TOKEN] HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json

{
  "data": {
    "mydlink_id": "44440123",
    "purchase_id": "4792838291",
    "page":1,                      # mydlink_id and purchase_id are optional.
                                   # if given they will be searched with and
```

```
                                        # condition. if not given will query all
                                        # orders of the user. page starts from 1,
                                        # default is 1 if not given
    "lang": "en"                        # refers to appendix II
  }
}
```

If the given information is correct openapi will return the list of orders to client.

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
{
  "data": {
    "orders": [
        {
            "purchase_id": "4792838291",
            "original_pid": "",
            "mydlink_id": "44443321",
            "name": "[Monthly] 7 days cloud storage for event base",
            "type": 1,
            "status": 1,
            "price":1.9,
            "discount":14.99,
            "handling_fee":2.4,
            "vat_percentage":0,
            "currency": "USD",
            "created_at": 1432642609,
            "valid_thru": 143342609,
            "notes":""
        },
        {
            "purchase_id": "",
            "original_pid": "4792838291",
            "mydlink_id": "44443321",
            "name": "[Monthly] 7 days cloud storage for event base",
            "type": 4,
            "status": 1,
            "price":0,
            "discount":0,
            "handling_fee":0,
            "vat_percentage":0,
            "currency": "USD",
            "created_at": 1432642609,
            "valid_thru": 0,
            "notes":""
        },
        {
            "purchase_id": "4792838292",
            "original_pid": "",
```

```
                "mydlink_id": "44443321",
                "name": "[Monthly] 7 days cloud storage for event base",
                "type": 3,
                "status": 1,
                "price":1.9,
                "discount":14.99,
                "handling_fee":2.4,
                "vat_percentage":0,
                "currency": "USD",
                "created_at": 1432642609,
                "valid_thru": 143342609,
                "notes":"upgrade"
            },
            {
                "purchase_id": "4792838291",
                "original_pid": "",
                "did": "44443321",
                "name": "[Monthly] 7 days cloud storage for event base",
                "type": 1,
                "price":1.9,
                "discount":14.99,
                "handling_fee":2.4,
                "vat_percentage":0,
                "currency": "USD",
                "created_at": 1432642609,
                "valid_thru": 143342609,
                "notes":""
            }
        ],
        "total":8,
        "page":1,
        "has_more":false
    }
}
```

Appendix I - Handling of HTTP Error Codes

The client of Open API shall be able to handle the HTTP error code well. The listing error codes must be handled:

Status Code	Client Actions
301 Move Permanently	Client shall save the target location in local cache and always send request to the cached one.
302 Found	Client shall save the target location in local cache till next login action.
400 Bad Request	The request is malformed. May be caused by wrong AppID, signature, username/password combination, or missing/invalid access token.
404 Not Found	The requested resource doesn't exist in OpenAPI. The client SHALL stop to request this resource.
500 Internal Server Error	Client shall retry the request for 3 times with certain frequency in each specified scope. If fail, the client shall stop the auto retry, unless triggered by user.

503
Service Unavailable
	Client shall display the message carried in the response and stop to send any request to the service. The server may return the error code when it's overloading or under maintenance period. Client should send any request to the service unless user trigger it manually.
	
	The response message of 503 status code shall be:
	
	``
	HTTP/1.1 503 Service Temporarily Unavailable
	Content-Type: application/json
	Retry-After: Fri, 1 May 2013 23:59:59 GMT
	
	{
	  "data": {
	    "type": "maintenance",
	    "info": [
	      { "lang": "en",
	        "msg": "mydlink website will undergo scheduled
	maintenance on May 14 from 12:00AM to 4:00AM." },
	      { "lang": "zh_TW",
	        "msg": "mydlink網站將會於5/14 12:00AM至4:00AM進行維護
	" }
	    ]
	  }
	}
	``
	
	The client shall handle the 'Retry-After' tag and be able to display the message carried in 'msg' tag by user's language. If client receives 503 response, the client should display the carried message and stop to provide any function which related to the service to user.

The table below lists the code id definitions for 400 Errors

Code ID	Message	Description
10	Error validating this request.	Parameter(s) carried with this request is missing or wrong.
14	Access token invalid.	The carried access token is invalid.
16	Field format invalid.	The field value to be updated is invalid
17	Session id invalid.	The session id for binding the given device is invalid.
18	Invalid ownership.	The user doesn't have the permission to access the specified device.
19	Offline.	The device client going to manage is offline at this moment.
30	No such record	No record found for the given information.
31	No privilege	The carried client_id has no privilege for this call
52	Internal error	DB access error.
53	Internal error	Failed to access external service (e.g., sqs, s3)
87	Payment gateway error	Api call to payment gateway api endpoints failed
88	Already subscribed	One or more of the given device has an ongoing active subscription
89	Action locked	Once a device changed plan, in the first period of new plan change plan and refund actions are locked.

Appendix II - Language Code Definitions

Code	Language
en	English
fr	French
ru	Russian
es	Spanish
pt	Portuguese(Portugal)
pt_BR	Portuguese(Brazil)
ja	Japanese
zh_TW	Traditional Chinese
zh_CN	Simplified Chinese
ko	Korean
cs	Czech
da	Danish
de	German
el	Greek
hr	Croatian
hu	Hungarian
it	Italian
nl	Dutch
no	Norwegian
pl	Polish
ro	Romanian
sl	Slovenian
sv	Swedish
fi	Finnish