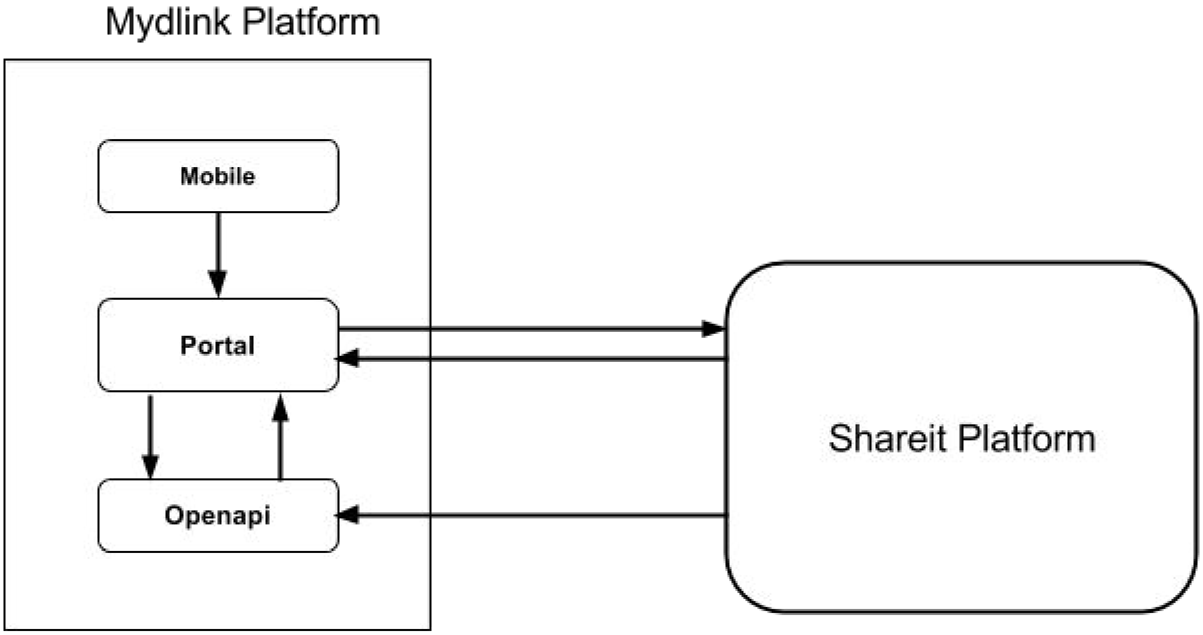
mydlink CloudNVR
Billing Specification

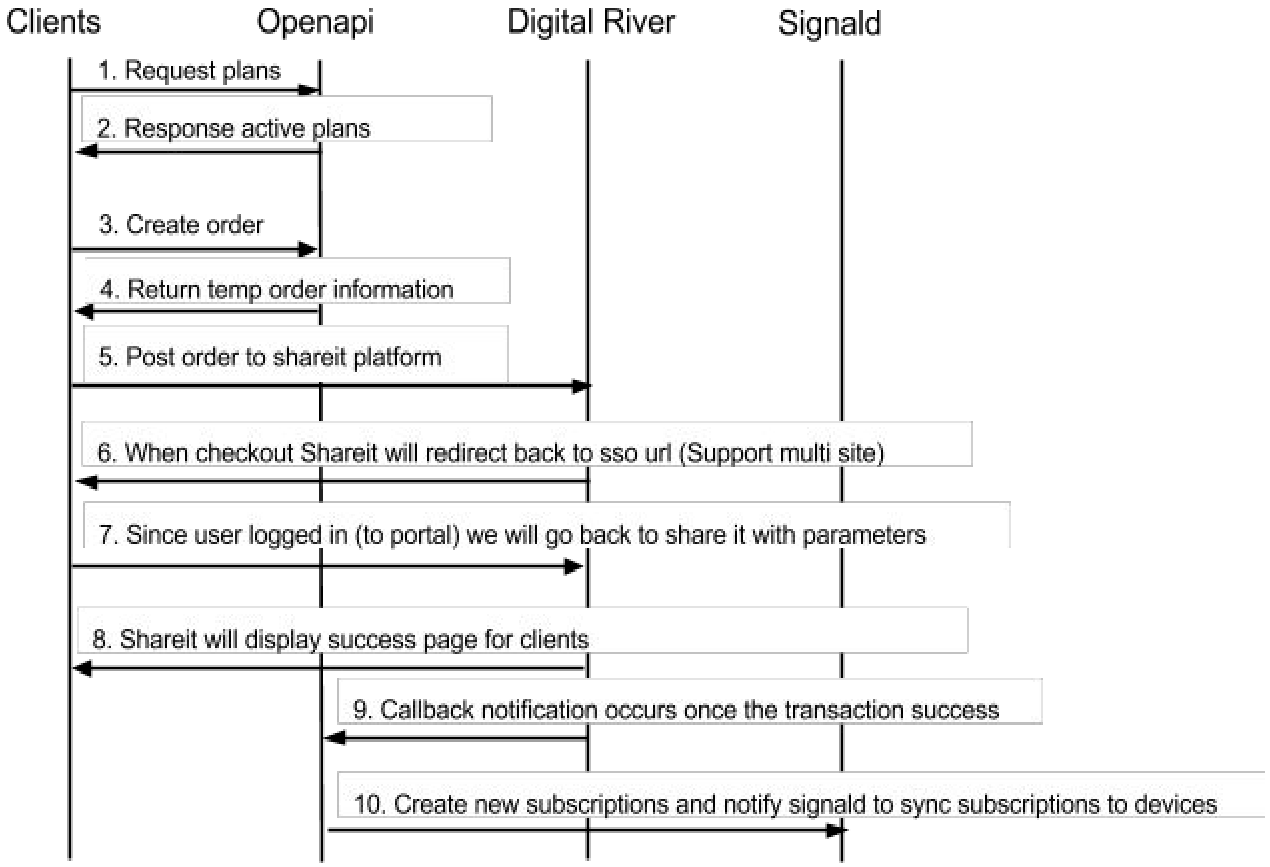
Version 1.0.0

Revise History

Versio n	Publish Date	Author	Descriptions
1.0.0	Dec. 08, 2014	Niyazi Makuloglu	Billing for CloudNVR

1. Architecture




Clients
Openapi
Digital River
Signald
1. Request plans
2. Response active plans
3. Create order
4. Return temp order information
5. Post order to shareit platform
6. When checkout Shareit will redirect back to sso url (Support multi site)
7. Since user logged in (to portal) we will go back to share it with parameters
8. Shareit 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: Portal implements cart functionality. Should support mobile.

1. Clients request available service plans from openapi. (For service plans management please refer to section 4.)
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 Share-it payment page.
6. User clicks checkout button, Share-it redirects user to our sso endpoint.
7. Since user logged in we pass user information to Share-it to prove that it's secure. (Refer to sso documentation pdf.)
8. Share-it display success page and a return link on the page. (Customizable)
9. Share-it notifies transaction results to OpenAPI callback url after a success transaction.
10. OpenAPI updates the order status & if the transaction is successful then creates subscription, changes device mode. Notify signald about the cnvr mode change. Signald syncs cnvr state with device.

2. Openapi

a. Get products

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.

```
GET /me/billing/products?access_token=[CLIENT_ACCESS_TOKEN]&lang=en HTTP/1.1
Host: api.mydlink.com
Content-Type: application/json
#  lang refers to http://10.32.38.1/index.php/I18n_language_codes
```

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"# England : GBP, Rest of Europe : EUR, Other Sites: 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"# England : GBP, Rest of Europe : EUR, Other Sites: 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"# England : GBP, Rest of Europe : EUR, Other Sites: USD
        },
        "settings": {"mode":2,"interval":"MON","space":7,"quota":"30"},
        "type": "cnvr"
      }
   ]
}
```

b. 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": [
  {"mydlink_id": "44440123", "plan": "cnvr-event-7-days"},
  {"mydlink_id": "...", "plan": "..."},
  {"mydlink_id": "...", "plan": "..."},
  {"mydlink_id": "...", "plan": "..."}
 ],
  "lang": "en" # refers to http://10.32.38.1/index.php/I18n_language_codes
}
```

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 share-it to proceed the payment.

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

{
  "data":{
    "url":
"https://secure.shareit.com/shareit/checkout.html?sessionid=2810962245&random=173f1ca
73fc6ba4a6637f9ba379b58ac"
  }
}
```

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

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

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" ]
}
```

d. Create free trial

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": ["44440123", "...", "..."]
}
```

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

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

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

e. 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":  [  "44440123", "44440124", "44440125", ... ],
 "lang": "en"  # refers to http://10.32.38.1/index.php/I18n_language_codes
}
```

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

f. 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"}
}
```

g. 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.
```

h. 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.
```

i. Change plan

Step 1. 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 http://10.32.38.1/index.php/I18n_language_codes
}
```

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"# 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": -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"# 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": 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"# England : GBP, Rest of Europe : EUR, Other Sites: USD
          }
        }
    ]
  }
}
```

Step 2. Change plan request :
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 http://10.32.38.1/index.php/I18n_language_codes
}
```

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 share-it to proceed the payment.

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

{
  "data":{
    "url": "[SHAREIT_CART_URL]" #
  }
}
```

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

j. 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 http://10.32.38.1/index.php/I18n_language_codes
}
```

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":
     [ {
        "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
}
```

3. Database Structure

- SKU Table

```
CREATE TABLE sku
(
  sid serial NOT NULL,
  external_code character varying(128), # ex. 300671324
  code character varying(128) NOT NULL, # ex. cnvr-event-7-days-monthly
  vendor_id integer NOT NULL, # ex. 1 for mydlink or 2 for  komfy
  prices text NOT NULL, # ex.{'USD': 9.99, 'EUR': 5.99}
  description text NOT NULL, # ex. 7 days cloud storage for event base
  settings text NOT NULL, # ex. {"mode": 1, "space": 7, "quota": 30}
  state integer DEFAULT 0, # ex. 1: on,  0: off
  type character varying(12), # ex. cnvr( the first part of sku code)
  CONSTRAINT sku_pkey PRIMARY KEY (sid),
  CONSTRAINT sku_code_unique UNIQUE (code),
  CONSTRAINT sku_external_code_unique UNIQUE (external_code)
)
```

- Vendor Table

```
CREATE TABLE vendor
(
  vid serial NOT NULL,
  code character(10) NOT NULL, # ex. mydlink or komfy
  CONSTRAINT vendor_pkey PRIMARY KEY (vid)
)
```

- Subscription Table

```
CREATE TABLE subscription
(
  sid bigserial NOT NULL,
  uid integer NOT NULL, # ex. 375330
  did character(8) NOT NULL, # ex. 26625352
  sku integer NOT NULL,  # ex. 1
  state integer NOT NULL, # ex. 0: expired, 1: active, 2: cancel
  type integer NOT NULL, # ex. 0: given internally, 1: purchase, 2:free trial
  recurring_period integer, # ex. 2 .  0 when created
  rebilling_price character(10)  , # ex. price for next rebilling
  created_at bigint NOT NULL DEFAULT 0, # ex. 1432642609
  updated_at bigint NOT NULL DEFAULT 0, # ex. 1432642609
  cancelled_at bigint NOT NULL DEFAULT 0, # ex. 1432643509
 expires_at bigint NOT NULL DEFAULT 0, # ex. 1433247581. if 0, the subscription won't expire
depend on order_item.payment_completed_at.
  content_valid_thru bigint, # ex. 1433247581
   change_flag boolean default false, # ex. if true, and recurring_period=0 , can not refund or
change plan
  CONSTRAINT subscription_pkey PRIMARY KEY (sid),
  CONSTRAINT "subscription_FK_1" FOREIGN KEY (sku)
      REFERENCES sku (sid) MATCH SIMPLE
      ON UPDATE NO ACTION ON DELETE NO ACTION
)

-- ALTER TABLE subscription RENAME COLUMN retry_count TO type;
-- ALTER TABLE subscription ADD COLUMN updated_at bigint;
-- ALTER TABLE subscription ADD COLUMN cancelled_at bigint;
-- ALTER TABLE subscription ADD COLUMN order_item_id bigint;
-- ALTER TABLE subscription ADD COLUMN recurring_period integer;
```

- Order Table

```
CREATE TABLE orders
(
  oid bigserial NOT NULL,
  user_id integer NOT NULL, # ex. 375330
  type integer NOT NULL, # ex. 1: order, 2: refund, 3: change plan, 4: cancel
  vendor_id integer NOT NULL, # ex. 1: mydlink, 2:comfy
  mydlink_oid character varying(128) NOT NULL, # ex. MO-000-33920240002 (TBD)
  purchase_id character varying(128), # ex. 485956915
  original_pid character varying(128), # ex. 485956915
  total_price numeric(10,2) NOT NULL DEFAULT 0.00, # ex. 99.99
  total_discount numeric(10,2) NOT NULL DEFAULT 0.00,  # ex. 19.99
  vat_percentage numeric(3,2) DEFAULT 0.00, # ex. 19.00
  invoice_no character varying(128), # ex. e5-DE-2015-00000189605
  currency character varying(3), # ex. USD
```

```
 customer_currency character varying(3), # ex. TWD
 initial_content text, # ex. { "info" :[
                                     {"mydlink_id": "44440123", "plan": "cnvr-event-7-days"},
                                     {"mydlink_id": "...", "plan": "..."},
                                     {"mydlink_id": "...", "plan": "..."},
                                     {"mydlink_id": "...", "plan": "..."}
                                    ]}
  notification_content text, # received from notification callbacks
  status integer NOT NULL, # ex. 0  (0: pending, 1: success, 2: failed)
  created_at bigint NOT NULL, # ex. 1432642609
  updated_at bigint NOT NULL, # ex. 1432642609
  purchased_at bigint, # ex. 1432642609
  payment_completed_at bigint, # ex. 1432642609
 CONSTRAINT orders_pkey PRIMARY KEY (oid),
 CONSTRAINT orders_mydlink_oid_unique UNIQUE (mydlink_oid),
 CONSTRAINT orders_purchase_id_unique UNIQUE (purchase_id)
)  WITH (
 OIDS=FALSE
);
ALTER TABLE orders
 OWNER TO mydlink;

CREATE INDEX orders_original_pid_idx
 ON orders
 USING btree
 (original_pid);
```

- Order item Table

```
CREATE TABLE order_items
(
 oiid bigserial NOT NULL,
 uid integer NOT NULL, # ex. 375330
 did character varying(8) NOT NULL, # ex. 44441234
 type integer NOT NULL, # ex. 1: order, 2: refund, 3: change plan, 4: cancel
 order_id bigint NOT NULL, # ex. 12333
 running_no character varying(128), # ex. 3
 purchase_id character varying(128), # ex. 485956915
 original_pid character varying(128), # ex. 485956915
 item_id character varying(128), # ex. 485956915
 sku_id integer NOT NULL, # ex. 485956915
 subscription_id bigint NOT NULL, # ex. 485956915
 price numeric(5,2) NOT NULL, # ex. 99.99
```

```
  discount numeric(5,2) NOT NULL DEFAULT 0.00, # ex. 19.99
  handling_fee numeric(5,2) DEFAULT 0.00, # ex. 1.99
  vat_percentage numeric(3,2) NOT NULL DEFAULT 0.00, # ex. 19.00
  currency character varying(3),
  customer_currency character varying(3),
  status integer NOT NULL, # ex. 0  (0: pending, 1: success, 2: failed)
  notes character varying(255), # ex. “user request cancel from shareit”
  next_rebilling_at bigint, # ex. 1432642609
  created_at bigint NOT NULL, # ex. 1432642609
  updated_at bigint NOT NULL, # ex. 1432642609
  CONSTRAINT order_item_pkey PRIMARY KEY (oiid)
)

CREATE INDEX order_items_original_pid_idx
  ON order_items
  USING btree
  (original_pid);
```

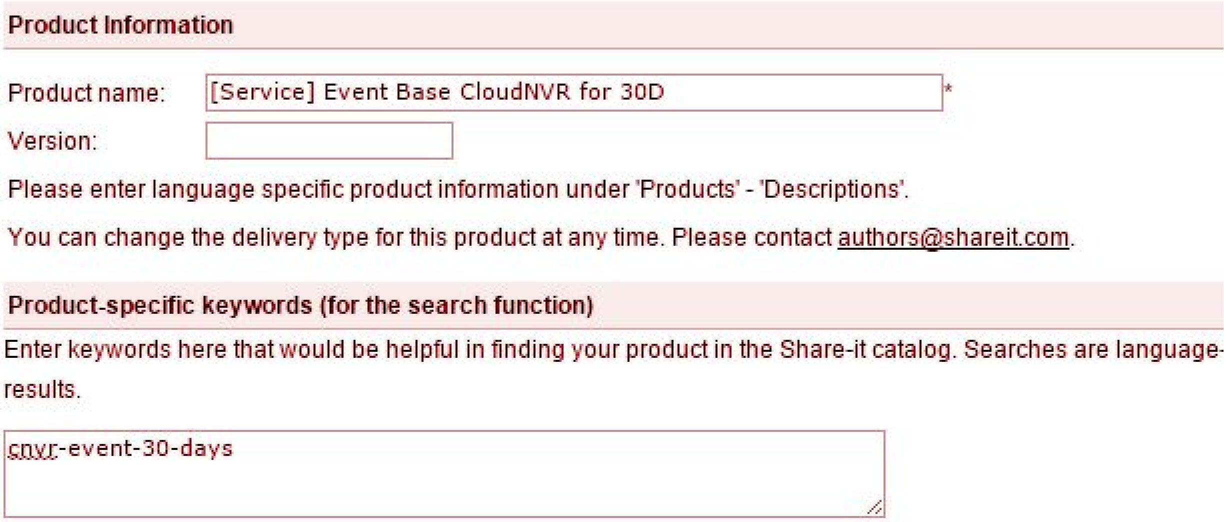
4. Product management

Share-it control panel is the master for product management. Mydlink platform will maintain a copy of product in backend. Products will be cached for better performance. Below workflow should apply for product management.

a. Creating products on share-it control panel

To create product, one have to have access to shareit control panel. Things to pay attention

- Fill in the product code to use in Mydlink platform into the keywords field. When Mydlink platform sync products from share-it, it will automatically parse this value to create non existing products.



Product Information

Product name: [Service] Event Base CloudNVR for 30D *

Version:

Please enter language specific product information under 'Products' - 'Descriptions'.

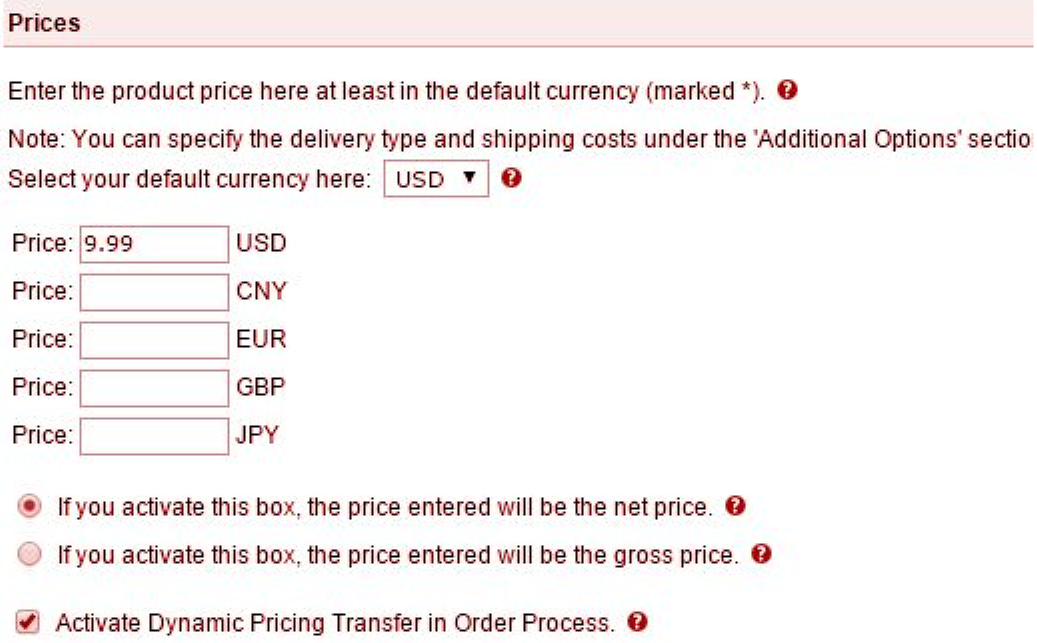
You can change the delivery type for this product at any time. Please contact authors@shareit.com.

Product-specific keywords (for the search function)

Enter keywords here that would be helpful in finding your product in the Share-it catalog. Searches are language-results.

cnvr-event-30-days

- Activate dynamic prices for upgrade/downgrade compatibility.



Prices

Enter the product price here at least in the default currency (marked *).

Note: You can specify the delivery type and shipping costs under the 'Additional Options' sectio

Select your default currency here: USD ▼

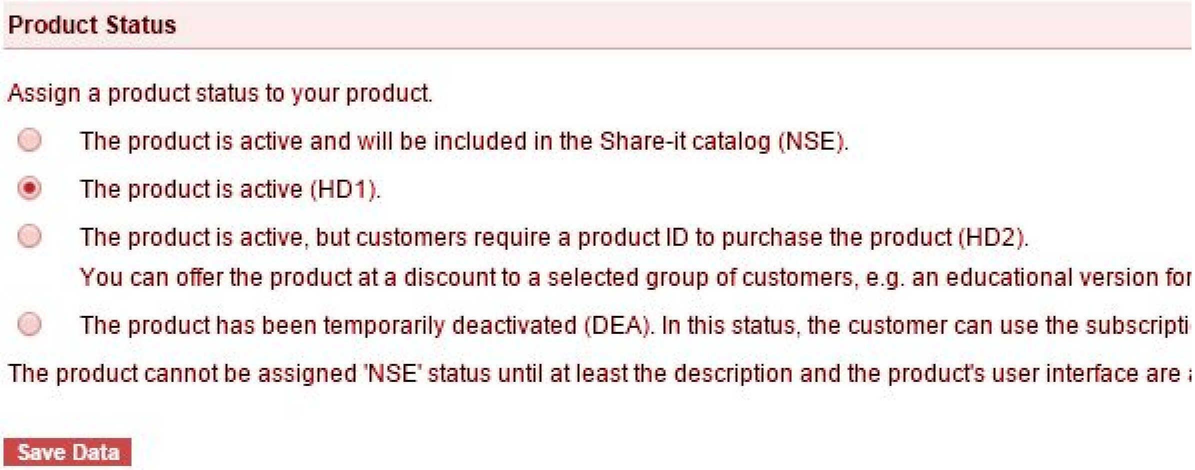
Price:	9.99	USD
Price:		CNY
Price:		EUR
Price:		GBP
Price:		JPY

● If you activate this box, the price entered will be the net price.

○ If you activate this box, the price entered will be the gross price.

☑ Activate Dynamic Pricing Transfer in Order Process.

- One can enable/disable products with below radio buttons. Once user disable a product should trigger sync to Mydlink platform. Please refer to 4.b for the ways sync**.**



Product Status

Assign a product status to your product.

○ The product is active and will be included in the Share-it catalog (NSE).

● The product is active (HD1).

○ The product is active, but customers require a product ID to purchase the product (HD2). You can offer the product at a discount to a selected group of customers, e.g. an educational version for

○ The product has been temporarily deactivated (DEA). In this status, the customer can use the subscripti

The product cannot be assigned 'NSE' status until at least the description and the product's user interface are

b. Syncing products to Mylink platform

- **Initiative way**

Can be triggered with a button in Mydlink platform control panel. (TBD)

- **Passive way**

A running cron job will sync active products to Mydlink platform periodically. (TBD)

5. Handling notifications from Digital River

Please check XML_2.4_and_JSON_Schema_EN.pdf for more information.

6. 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.

<table>
<tr>
<td>500
Internal Server Error</td>
<td>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.</td>
</tr>
<tr>
<td>503
Service Unavailable</td>
<td>

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.

</td>
</tr>
</table>

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	Shareit error	Api call to shareit 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.