**Summary**

This document highlights the project status and outlines the next steps for completing the yaaS Bitcoin Integration Proof of Concept.

**Remaining Questions**

* Do we need to use an “Application”, or will Services suffice?
* Is a yaaS integration with BitPay possible due to the 2-factor sign-in nature of BitPay, or should we switch to integration with a Bitcoin Wallet service?

**yaaS Hierarchy**

The organization and project hierarchy **should** be complete, though we may need an application for this project (none is created). The services and builder module are not complete. The high level tasks are noted as to-dos. Once the services and module are complete, they should be packaged and deployed in the existing “bitpay-package”.

**Organization – Saabor (Access from “My Account”)**

* Base Path: yaas-bitpay
* ID: 55e86ab8f36e799d4bd5fcc3
* **Test Project - Bitcoin Integration (Access from “Builder”)**
  + ID (tenant in API calls): bitcoin
  + **Dev Team – Bitcoin Int’g POC (Access from “Builder”)**
    - ID: hybrisbitcoinpoc
    - Services:
      * **merchant**
        + YaaS Client: yaas-bitpay.merchant (client in API calls)
        + Client ID: Bb1iyhEIKTPk8VJ8LmbHk7EYY1c67OL6
        + Client Secret: 01APK7xkkFasK5a1
        + Used for Merchant Management
        + To-do: Create services to get, post, put, delete merchants.
      * **bitpay-post**
        + YaaS Client: yaas-bitpay.bitpay-calls (client in API calls)
        + Client ID: 9pQtuZ4CD6w525vW9IcgjJHJezhz80OK
        + Client Secret: lYpM80wRjN6Xg85D
        + Used for Interaction with BitPay to create invoices, redirect to hosted order page, and pass return URL.
        + To-do: Create services to get pairing token, post to bitpay for invoicing, redirect to bitpay for payment, and get payment information on redirect back to order confirmation.
    - Builder Module: **bitpay-merchant**
      * Facade to manage merchant data.
      * To-do: Create angular js/jquery which leverages merchant service to manage merchant data.
      * To-do: Deploy service to Pivotal Web Service or other hosting provider.
    - Package: **bitpay-package**
      * Includes merchant and bitpay-post services
      * Includes bitpay-merchant module
      * To-do: Deploy package once services and builder module complete.

**Document Service API Call Overview**

This is a summary of API calls to get access tokens, and read from/write to the Document Services. The calls originate from the OAuth2 services in the Dev Portal on yaas.io. These calls will need to be issued from the merchant and bitpay-post services, but until this is complete, the following examples will allow one to leverage the Document Services.

1. Get API Token for write access for the merchant service.
   1. Go to <https://devportal.yaas.io/services/>.
   2. Select “OAuth2” from the left nav under “Core”.
   3. Select the yellow “API Console” button in the page header.
   4. Select “Post” under /token.
   5. On the API Console page:
      * Select “Try It” in the green rectangle.
      * In the Body Section:
      * Enter the client ID (Bb1iyhEIKTPk8VJ8LmbHk7EYY1c67OL6) and client secret (01APK7xkkFasK5a1) for the merchant service.
      * Type “client\_credentials” for “grant\_type”
      * Enter “hybris.tenant=bitcoin hybris.document\_manage” for “scope”.
      * Press the “Post” button.
   6. This should return an access token in the Body text box. Copy this access token to a text editor. The token is good for an hour.
   7. Curl command: ‘curl -i -H "Content-Type: application/x-www-form-urlencoded" -X POST -d "grant\_type=client\_credentials&scope=hybris.tenant=bitcoin hybris.document\_manage&client\_id=Bb1iyhEIKTPk8VJ8LmbHk7EYY1c67OL6&client\_secret=01APK7xkkFasK5a1" https://api.yaas.io/hybris/oauth2/b1/token’
2. Get API Token to read from the merchant service.
   1. Follow steps a-e above, accept use “hybris.tenant=bitcoin hybris.document\_view” for “scope”.
   2. Copy the access token to a text editor. The token is good for an hour.
   3. Curl command: ‘curl -i -H "Content-Type: application/x-www-form-urlencoded" -X POST -d "grant\_type=client\_credentials&scope=hybris.tenant=bitcoin hybris.document\_view&client\_id=Bb1iyhEIKTPk8VJ8LmbHk7EYY1c67OL6&client\_secret=01APK7xkkFasK5a1" https://api.yaas.io/hybris/oauth2/b1/token’
3. Write merchant data to Document Service
   1. Go to <https://devportal.yaas.io/services/>.
   2. Select “Document” from the left nav under “Core”.
   3. Select the yellow “API Console” button in the page header.
   4. Look for ‘/{tenant}/{client}/data/{type}’, then Select the “Post” button.
   5. On the API Console page:
      * Select “Try It” in the green rectangle.
      * In the Post URL, use the following:
      * For tenant, use “bitcoin”.
      * For client, use “yaas-bitpay.merchant”.
      * For type, use “merchant”.
      * In the Headers section, use the following:
      * For Authorization, use “Bearer {API Token for write access}” (generated in step 1 above).
      * In the “Body” text area field, enter the merchant data, for example:

{

"id":"666666",

"api\_key":"NtswthcMXSdZzaa3WULXgSPZWV9IQ7ed6nkeGojjY",

"url":"https://www.hybris.com/en/extend-marketplace",

"merchant\_name":"Primary Bitpay Merchant",

"description":"Primary Bitpay Merchant for Company XYZ"

}

* + - Press the “Post” button. This should write the data to the Document Service. You should see a 201 response.
  1. Curl command: ‘curl -i -X POST -d '{"id":"123456789","api\_key":"NtswthcMXSdZzaa3WULXgSPZWV9IQ7ed6nkeGojjY","url":"https://www.hybris.com/en/extend-marketplace","merchant\_name":"Curl Test Merchant","description":"Testing writing to Document Services via curl"}' -H "Content-Type: application/json" -H "Authorization: Bearer *[document\_manage access token]*" https://api.yaas.io/hybris/document/b2/bitcoin/yaas-bitpay.merchant/data/merchant’

1. Read merchant data from Document Service
   1. Follow steps a-c above.
   2. Look for ‘/{tenant}/{client}/data/{type}’, then select the “Get” button. Alternatively, select “Get” next to ‘/{tenant}/{client}/data/{type}/{dataId}’ to return a specific merchant.
   3. On the API Console page:
      * Select “Try It” in the blue rectangle.
      * In the Post URL, use the following:
      * For tenant, use “bitcoin”.
      * For client, use “yaas-bitpay.merchant”.
      * For type, use “merchant”.
      * If passing dataId, enter the ID of the merchant.
      * For Authorization in the Headers section, use “Bearer {API Token for write access}” (generated in step 2 above).
      * Press the “Get” button. This should return a 200 response and the merchant(s) data in the body text box.
   4. Curl command: ‘curl -i -X GET -H "Content-type: application/json" -H "Authorization: Bearer *[document\_view access token]*" https://api.yaas.io/hybris/dment/b2/bitcoin/yaas-bitpay.merchant/data/merchant’

**Issues with Bitpay**

BitPay seems to require 2 factor authentication to us their APIs. The BitPay documentation does not make it clear out to set up a cloud service merchant to interact with BitPay. Recommend connecting with BitPay expect to determine the proper architecture. In the meantime, we can move forward with a Bitcoin wallet integration.

**Next Steps**

Summary of anticipated next steps to complete project.

1. Get services working.
   1. Implement document service interaction.
   2. Complete mock services.
2. Change mock services to integrate to a Bitcoin wallet, as directly to Bitpay integration does not currently work.
3. Implement an exchange rate service.
4. Determine if we need an application(s).
5. Write angular JS builder module to get, push, post and delete merchant data.
6. Host builder module on Pivotal.
7. Package up and publish services.