Browser User Management: Use Cases

V1.0 | Chetan Channa | 13/12/2018

Contents

[Framework & Rules for User Identification 5](#_Toc532490352)

[Various Entities & Relationship 5](#_Toc532490353)

[Assumptions Based on Which the Rules are Framed 5](#_Toc532490354)

[Various Ids in the System For Browser 6](#_Toc532490355)

[Logical View of the System 6](#_Toc532490356)

[Some changes with respect to old system 7](#_Toc532490357)

[Front end Principles & Guiding Rules 8](#_Toc532490358)

[Summary of Use Cases Covered 9](#_Toc532490359)

[Definitions 9](#_Toc532490360)

[Scenarios 9](#_Toc532490361)

[Use Cases 11](#_Toc532490362)

[Use Case 1: New User opening browser outside of partner network & doesn’t log in 11](#_Toc532490363)

[Use Case 2: New User opens browser outside of partner network & Signs Up 11](#_Toc532490364)

[Use Case 3: New User opens browser with a new partner-id & is logged out 12](#_Toc532490365)

[Use Case 4: New User opens browser with a new partner & then signs up 12](#_Toc532490366)

[Use Case 5: Repeat User comes on the same browser outside of partner network 13](#_Toc532490367)

[and doesn’t log in 13](#_Toc532490368)

[Use Case 6: Repeat User comes on the same browser and Signs Up & is outside of partner network 14](#_Toc532490369)

[Use Case 7: Repeat User comes on the same browser and Signs in & is outside of partner network 14](#_Toc532490370)

[Use Case 8: Repeat User comes on the same browser and doesn’t log in & is comes with a new partner-id 15](#_Toc532490371)

[Use Case 9: Repeat User comes on the same browser and signs up & is having a new partner-id 16](#_Toc532490372)

[Use Case 10: Repeat User comes on the same browser and signs in & is a new partner-id 17](#_Toc532490373)

[Use Case 11: Repeat User comes on the different Browser/Incognito & doesn’t authenticate & is not on partner network 18](#_Toc532490374)

[Use Case 12: Repeat User comes on the different Browser/Incognito & Signs Up & is not on partner network 19](#_Toc532490375)

[Use Case 13: Repeat User comes on the different Browser/Incognito & Signs In & is not on partner network 20](#_Toc532490376)

[Use Case 14: Repeat User comes on the different Browser/Incognito & Logged out & is on a new partner-id 22](#_Toc532490377)

[Use Case 15: Repeat User comes on the different Browser/Incognito & Signs Up & is on a new partner-id 23](#_Toc532490378)

[Use Case 16: Repeat User comes on the different Browser/Incognito & Signs In & is on a new partner-id 24](#_Toc532490379)

[Use Case 17: New User comes with an old/existing partner-id & is not logged in 26](#_Toc532490380)

[Use Case 18: New User comes with an old/existing partner-id & Signs Up 27](#_Toc532490381)

[Use Case 19: Repeat User comes on the same browser and doesn’t log in & is with existing/old partner id 29](#_Toc532490382)

[Use Case 20: Repeat User comes on the same browser and signs up & is with existing/old partner id 31](#_Toc532490383)

[Use Case 21: Repeat User comes on the same browser and signs in & is with existing/old partner id 33](#_Toc532490384)

[Use Case 22: Repeat User comes on the different Browser/Incognito & Logged out & is on old/existing partner-id 35](#_Toc532490385)

[Use Case 23: Repeat User comes on the different Browser/Incognito & signs up & is on old/existing partner-id 37](#_Toc532490386)

[Use Case 24: Repeat User comes on the different Browser/Incognito & signs in & is on old/existing partner-id 40](#_Toc532490387)

[Front End User Stories(TBD) 43](#_Toc532490388)

[Use Case 1: Multiple Partner Ids (this needs to be handled via FE) 43](#_Toc532490389)

[Use Case 2: User does a strong auth & the switch to partner network with an existing partner-id within the session 43](#_Toc532490390)

[Use Case 3 : User does a strong auth & the switch to partner network with a new partner-id within the session 43](#_Toc532490391)

[Viu-id Creation & Fetching Rules 44](#_Toc532490392)

[Backend Scenarios 45](#_Toc532490393)

[Client Passes Device Id only 45](#_Toc532490394)

[No entry found in backend (Use Case#1) 45](#_Toc532490395)

[One entry found in backend not mapped to any other ids (Use Case#5) 45](#_Toc532490396)

[Entries found in backend but all of them mapped to some other ids (partner or strong auth) 45](#_Toc532490397)

[Multiple entry found for Device id not attached to any other identifier (partner or strong auth) 45](#_Toc532490398)

[Client Passes Device Id & one Strong Auth Id 45](#_Toc532490399)

[Exact match found with the given Device id & Strong Auth id (Use Case#7) 45](#_Toc532490400)

[Device id not found but one entry found with strong auth (Use Case#13) 45](#_Toc532490401)

[Device id not found but multiple entries found with same strong auth 45](#_Toc532490402)

[Device id found but strong auth not found (Use Case#6) 45](#_Toc532490403)

[Device id not found & strong auth not found (Use Case#12) 45](#_Toc532490404)

[Multiple Exact match found with the given Device id & Strong Auth id 45](#_Toc532490405)

[Client Passes Device Id & MULTIPLE Strong Auth Id 46](#_Toc532490406)

[Client Passes Device Id & one Partner id 46](#_Toc532490407)

[Exact match found with the given Device id & partner id 46](#_Toc532490408)

[One entry found with partner id (not attached to strong auth) 46](#_Toc532490409)

[One entry found with partner id but attached to strong auth 46](#_Toc532490410)

[Partner-id found but multiple entries found with strong auth 46](#_Toc532490411)

[Device id found but partner id not found 46](#_Toc532490412)

[Device id not found & partner id not found 46](#_Toc532490413)

[Multiple Exact match found with the given Device id & Strong Auth id 46](#_Toc532490414)

[Client Passes Device Id & MULTIPLE Strong Auth Id 46](#_Toc532490415)

[Client Passes Device Id & Multiple Partner Ids 47](#_Toc532490416)

[Client Passes Device Id & Partner Id & Strong Auth ID 47](#_Toc532490417)

[Exact match found 47](#_Toc532490418)

[Exact match not found 47](#_Toc532490419)

# Framework & Rules for User Identification

1. Viu-id will be used in all situations for reporting AUs & no other logic is needed.

# Various Entities & Relationship

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Entity** | **Relationship** | **Comment** |
| User | Device | 1:n | One user can have many devices |
| Device | Browser | 1:n | One device can have multiple browsers |
| User | Partner | 1:n | A user can have multiple partners (at the same time) |
| User | Strong Auth | 1:n | One user can do multiple strong auths |
| Strong Auth | Viu id | 1:1 | We will map one strong auth to one Viu id which means that Viu id is an identifier for strong auth. If the user is using different strong auths, that for us means he wants to be identified as a different user |
| Partner | Strong Auth | n:1 | One strong auth can be linked to multiple partners |

From browser specific point of view, we can’t identify devices, but we can identify browsers. So the first relationship is ‘immaterial’ from browser point of view. Each instance of one type of browser is what will be identified.

# Assumptions Based on Which the Rules are Framed

* All devices are personal devices used by one user unless otherwise indicated by different email ids/authentications
* Devices are not shared by multiple users a lot. For desktop 10% of the devices had multiple email ids but for mobile site the number is less than 1%.
* In order of strength for identifying a user: Strong auth > Partner id > Device id. It means the following
  + One email id even if it is mapped to multiple partner-ids & devices is one user.
  + In absence of email, one partner-id across multiple browsers/devices is one user.
* Users change their devices more often than phone numbers.
* The only way to identify users 100% accurately is if we force them to authenticate & they use only one email-id. Since this is not what we can enforce, there would always be some leakage & one actual user may be mapped to multiple-ids (and we will never know that).

# Various Ids in the System For Browser

* **Device-id** – This is a misnomer but what it means is that the browser will drop a persistent cookie with an identifier in a browser. Till the user deletes this cookie or goes incognito or uses a different browser, this device-id would remain same & will not be impacted by any other flow in the system. This device id must be uniquely generated if not already present on the browser.
* **Partner-id** – This is an external id received by the product by way of partnership. Generally, this id is a unique identifier for the user on partner eco-system. Product will not generate this id. For e.g. MSISDN received via HE or user-id received via backend in Myanmarnet is a partner-id. We can have multiple Viu-ids for same partner-id
* **Viu-id** – This is an identifier backend generates depending upon the information received by front-end. The ONLY rule here we need to follow is that for one strong-auth, we will have only one Viu-id. For e.g. for email id [abc@xyz.com](mailto:abc@xyz.com), we can’t have more than one Viu-id mapped at backend. This is something we need to enforce. What this also means is that by design, we can’t link two strong auths.

# Logical View of the System

1. We would never be able to enforce that the user to have only one Viu-id except that for each strong auth we can map one Viu id.
2. Front end will pass whatever information is has for the user to backend.
3. Backend will process all this information & return one Viu-id based on pre-defined rules.
4. This Viu id would be used for user information.
5. Front end at anytime can send updated information & Viu-id might change depending on the rule.

A close up of a map

Description automatically generated

# Some changes with respect to old system

1. There would be one new functionality via which front-end will pass all the available identifiers & backend would return Viu-id, status (Conflict or No Conflict) & Call to action (Authenticate user via email ids).
2. Since we do shadow-login on partner-id to fetch privileges & account, this is kept out of scope of current doc since shadow login is no necessary to provide privilege.
3. For us login/authentication will mean the same & it has to be explicitly done by the user. Front-end will decide if & when to authenticate the user.
4. Some rules related to linking
   1. Two Viu-ids mapped to different strong auths will never be linked
   2. If a Viu-id is mapped to partner-id, it would not be linked to strong auth
   3. A sign-up (new strong auth) will create a new Viu-id except for the cases where the viu-id in context is not mapped to any other strong auth or partner-id

# Front end Principles & Guiding Rules

1. If user requests to log in via strong auth, we always log the user in.
2. If user has multiple strong auths attached, always ask the user which one to use.
3. User should always know what accounts he or she has.
4. User should always be able to de-activate himself or herself if he/she is mapped to multiple accounts.
5. Always Inform the user in case ids switched & privileges he/she might gain or lose.

# Summary of Use Cases Covered

## Definitions

1. New – A new user is a user who’s coming onto our system the first time (irrespective of our internal systems can identify or not).
2. Repeat – A user who has visited us before time (irrespective of our internal systems can identify or not).
3. Device -id – If the persistent cookie with the identifier dropped by browser is present or not.
4. Partner-id – If the front-end was able to fetch the partner-id
   1. Not Available
   2. Old/Existing – User has visited Viu with that partner-id
   3. New – User has come for the first time using this partner-id.
5. Auth Status
   1. Logged Out – User is not signed in & user is not authenticated
   2. Signs up – User authenticates with this id for the first time & we don’t have info on this strong auth at backend
   3. Signs in – user authenticates using an already used/available strong auth in the sytem

## Scenarios

User cases are defined basis various combination of new/repeat, device-id existing or new, partner id available or not or new, user’s authentication status. The number of mathematically possible use cases is 36 & out of these 24, 12 are not logically possible & ‘strike-throughed’ below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Type of User** | **Device Id** | **Partner Id** | **Strong Auth Log in Status** |
| 1 | New | N | Not Available | Logged Out |
| 2 | New | N | Not Available | Signs Up |
| ~~N/A~~ | ~~New~~ | ~~N~~ | ~~Not Available~~ | ~~Signs In~~ |
| 3 | New | N | New | Logged Out |
| 4 | New | N | New | Signs Up |
| ~~N/A~~ | ~~New~~ | ~~N~~ | ~~New~~ | ~~Signs In~~ |
| 5 | Repeat | Y | Not Available | Logged Out |
| 6 | Repeat | Y | Not Available | Signs Up |
| 7 | Repeat | Y | Not Available | Signs In |
| 8 | Repeat | Y | New | Logged Out |
| 9 | x | Y | New | Signs Up |
| 10 | Repeat | Y | New | Signs In |
| 11 | Repeat | N | Not Available | Logged Out |
| 12 | Repeat | N | Not Available | Signs Up |
| 13 | Repeat | N | Not Available | Signs In |
| 14 | Repeat | N | New | Logged Out |
| 15 | Repeat | N | New | Signs Up |
| 16 | Repeat | N | New | Signs In |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~Not Available~~ | ~~Logged Out~~ |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~Not Available~~ | ~~Signs Up~~ |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~Not Available~~ | ~~Signs In~~ |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~New~~ | ~~Logged Out~~ |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~New~~ | ~~Signs Up~~ |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~New~~ | ~~Signs In~~ |
| 17 | New | N | Old/Existing | Logged Out |
| 18 | New | N | Old/Existing | Signs Up |
| 19 | Repeat | Y | Old/Existing | Logged Out |
| 20 | Repeat | Y | Old/Existing | Signs Up |
| 21 | Repeat | Y | Old/Existing | Signs In |
| 22 | Repeat | N | Old/Existing | Logged Out |
| 23 | Repeat | N | Old/Existing | Signs Up |
| 24 | Repeat | N | Old/Existing | Signs In |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~Old/Existing~~ | ~~Logged Out~~ |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~Old/Existing~~ | ~~Signs Up~~ |
| ~~N/A~~ | ~~New~~ | ~~Y~~ | ~~Old/Existing~~ | ~~Signs In~~ |
| ~~N/A~~ | ~~New~~ | ~~N~~ | ~~Old/Existing~~ | ~~Signs In~~ |

# Use Cases

## Use Case 1: New User opening browser outside of partner network & doesn’t log in

1. Browser creates a device id.
2. Backend generates a Viu-id associated with this device id.
3. This Viu-id is used to identify the user.
4. Ideally this device id remains on user’s browser forever or if user deletes it via clearing the cache we should be able to re-generate the same
5. This is how data should look.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |

## Use Case 2: New User opens browser outside of partner network & Signs Up

1. Browser creates a device id.
2. Backend generates a Viu-id associated with this device id.
3. User does a strong auth with a new strong auth id.
4. Device id remains the same.
5. So Device Id & strong auth id maps to the same Viu Id
6. User is identified via Viu id
7. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |

## Use Case 3: New User opens browser with a new partner-id & is logged out

1. Browser creates a device id which is unique per browser per user.
2. Since user is on partner network, we get partner id.
3. Device is mapped to partner id.
4. Backend generates Viu id which is mapped to both device id & partner id
5. User is identified via Viu id
6. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |

## Use Case 4: New User opens browser with a new partner & then signs up

1. Front end drops device id on the browser
2. Since user is on partner network, we get partner id.
3. Device id is mapped to partner id.
4. Backend generates Viu id which is mapped to both device id & partner id
5. User logs in via strong auth which is new
6. A new Viu id is mapped to strong auth is created
7. New entry is created in the system with all the 3 ids mapped to strong auth Viu-id
8. Use Viu Id For Identification & Reporting DAU
9. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |

## Use Case 5: Repeat User comes on the same browser outside of partner network

## and doesn’t log in

1. There is no other identifier available apart from Device id.
2. Product makes a call for the backend & gets Viu-Id just mapped to this device-id
3. Use the last accesses Viu-id to identify the user.
4. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |

## Use Case 6: Repeat User comes on the same browser and Signs Up & is outside of partner network

1. Device id is already present on the device.
2. There is no other identifier available.
3. User Signs Up
4. Existing Viu-id is mapped to strong auth.
5. Use this Viu-id

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |

## Use Case 7: Repeat User comes on the same browser and Signs in & is outside of partner network

1. Device id is already present on the device
2. User signs in
3. Product fetches the Viu id associated with the strong auth based on rules.
4. We identify & report the user via the viu id.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |

## Use Case 8: Repeat User comes on the same browser and doesn’t log in & is comes with a new partner-id

1. Device Id & partner id is available
2. Since the request is coming on same device via a different partner, we assume that this is the same user (as per our guiding principle)
3. Fetch the last accessed Viu-id basis this device-id & create a new entry
4. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |

## Use Case 9: Repeat User comes on the same browser and signs up & is having a new partner-id

1. Device Id & partner id are available
2. Link the partner-id to device-id & use the last accessed Viu-id
3. Since a new partner-id is received
   1. We check if for this device id, any Viu-id is available which is mapped to strong auth
      1. If yes, we force the user to authenticate
      2. If user doesn’t authenticate
         1. we check if we have any Viu-id for this device-id which is not mapped to strong auth and return that if available
         2. If not, we create a new entry & new id
4. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |

## Use Case 10: Repeat User comes on the same browser and signs in & is a new partner-id

1. Depending upon the condition, a new viu-id or the partner-id would get linked to existing Viu-id.
2. Since user authenticates, Viu id mapped to strong auth will be in context
3. A new entry would be created with the strong auth Viu-id mapped to device-id & partner-id
4. Use this Viu-id to identify & report.
5. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |

## Use Case 11: Repeat User comes on the different Browser/Incognito & doesn’t authenticate & is not on partner network

1. Device Id is not available, hence browser drops a new device-id
2. There is no way of knowing that user came before as well.
3. New Viu-id is created mapped to this device-id
4. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |

## Use Case 12: Repeat User comes on the different Browser/Incognito & Signs Up & is not on partner network

1. Device Id is not available; hence browser drops a new device-id.
2. User then does a strong auth & this is a new strong auth.
3. Create a new Viu id associated with strong auth.
4. Link Device Id to this strong auth & Viu id
5. Use this Viu-id to identify the user.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |

## Use Case 13: Repeat User comes on the different Browser/Incognito & Signs In & is not on partner network

1. Device Id is not available; hence browser drops a new device-id.
2. User then does a strong auth & this is a not a new strong auth.
3. Based on rules, fetch the Viu id associated with strong auth.
4. Link the new Device Id to this strong auth & Viu id.
5. Since the user is authenticated, use this Viu-id to identify the user.
6. This is how the data looks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |

## Use Case 14: Repeat User comes on the different Browser/Incognito & Logged out & is on a new partner-id

1. Device Id is not available; hence browser drops a new device-id.
2. We then fetch partner-id based on pre-defined process like HE.
3. Since the partner is new, this partner-id would get linked to the Device-id & Viu-is would be same
4. Use this Viu-id to identify the user.
5. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |

## Use Case 15: Repeat User comes on the different Browser/Incognito & Signs Up & is on a new partner-id

1. Device Id is not available; hence browser drops a new device-id & links the partner-id to this device-id & Viu-id
2. User then signs up & is a new user.
3. Since this is a new strong auth & the in-context Viu-id is not linked to any strong-auth, we link this strong-auth to the existing Viu-id
4. A new Entry is created with device-id, partner id & strong auth id
5. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |

## Use Case 16: Repeat User comes on the different Browser/Incognito & Signs In & is on a new partner-id

1. Device Id is not available; hence browser drops a new device-id.
2. We then fetch partner-id based on pre-defined process like HE.
3. Based on rules, fetch the Viu id associated with the partner-id if available else create new.
4. User then signs in & is not a new user. Browser sends device id, partner id & strong auth id to backend
5. Backend returns Viu-id associated with strong auth
6. Backend creates a new entry in the system with all the ids associated
7. Use this Viu-id to identify the user.
8. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |

## Use Case 17: New User comes with an old/existing partner-id & is not logged in

1. Browser passes device & partner-id
2. We fetch the Viu-id as per the rules for the partner-id.
3. If there is a strong auth associated with the partner-id, ask the user to authenticate
4. If the user authenticates, use the Viu id as per authentication.
5. Even if not authenticated, we use the last accessed Viu-id in case of multiple Viu-ids
6. User is identified via Viu-id
7. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |

## Use Case 18: New User comes with an old/existing partner-id & Signs Up

1. Front end drops device id on the browser
2. Since user is on partner network, we get partner id.
3. User logs in via strong auth which is new
4. Existing Viu-id is mapped to this strong auth
5. Use Viu Id For Identification & Reporting DAU
6. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 |  | V11 | 18 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 | S10 | V11 | 18 | Since there is no strong auth mapped to partner, we map the same Viu-id |

## Use Case 19: Repeat User comes on the same browser and doesn’t log in & is with existing/old partner id

1. Device Id & partner id is available
2. We fetch the Viu-id as per the rules for the partner-id.
3. If there is a strong auth associated with the partner-id, ask the user to authenticate
4. If the user authenticates, use the Viu id as per authentication.
5. Even if not authenticated, we use the last accessed Viu-id in case of multiple Viu-ids
6. User is identified via Viu-id
7. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 |  | V11 | 18 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 | S10 | V11 | 18 | Since there is no strong auth mapped to partner, we map the same Viu-id |
| D8 | P6 |  | V11 | 19 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |

## Use Case 20: Repeat User comes on the same browser and signs up & is with existing/old partner id

1. Device Id & partner id is available
2. We fetch the Viu-id as per the rules for the partner-id.
3. If there is a strong auth associated with the partner-id, ask the user to authenticate.
4. If the user authenticates, use the Viu id as per authentication.
5. Even if not authenticated, we use the last accessed Viu-id in case of multiple Viu-ids
6. User is identified via Viu-id
7. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 |  | V11 | 18 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 | S10 | V11 | 18 | Since there is no strong auth mapped to partner, we map the same Viu-id |
| D8 | P6 |  | V11 | 19 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 |  | V9 | 20 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 | S7 | V16 | 20 | We create a new Viu-id since P8 is already mapped to strong auth (s5) |

## Use Case 21: Repeat User comes on the same browser and signs in & is with existing/old partner id

1. Device Id & partner id is available
2. We fetch the Viu-id as per the rules for the partner-id.
3. If there is a strong auth associated with the partner-id, ask the user to authenticate
4. If the user authenticates, use the Viu id as per authentication.
5. Even if not authenticated, we use the last accessed Viu-id in case of multiple Viu-ids
6. User is identified via Viu-id
7. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 |  | V11 | 18 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 | S10 | V11 | 18 | Since there is no strong auth mapped to partner, we map the same Viu-id |
| D8 | P6 |  | V11 | 19 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 |  | V9 | 20 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 | S7 | V16 | 20 | We create a new Viu-id since P8 is already mapped to strong auth (s5) |
| D12 | P6 |  | V11 | 21 | Create a new id, if the user doesn't authenticate |
| D12 | P6 | S5 | V9 | 21 | Fetch the viu-id mapped to strong auth |

## Use Case 22: Repeat User comes on the different Browser/Incognito & Logged out & is on old/existing partner-id

1. Device Id is not available; hence browser drops a new device-id.
2. Device Id & partner id are now available.
3. We fetch the Viu-id as per the rules for the partner-id.
4. If there is a strong auth associated with the partner-id, ask the user to authenticate
5. If the user authenticates, use the Viu id as per authentication.
6. Even if not authenticated, we use the last accessed Viu-id in case of multiple Viu-ids
7. Create a new entry in the system with new device-id & partner-id mapped to this Viu-id
8. User is identified via Viu-id
9. This is how the data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 |  | V11 | 18 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 | S10 | V11 | 18 | Since there is no strong auth mapped to partner, we map the same Viu-id |
| D8 | P6 |  | V11 | 19 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 |  | V9 | 20 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 | S7 | V16 | 20 | We create a new Viu-id since P8 is already mapped to strong auth (s5) |
| D12 | P6 |  | V11 | 21 | Create a new id, if the user doesn't authenticate |
| D12 | P6 | S5 | V9 | 21 | Fetch the viu-id mapped to strong auth |
| D13 | P6 |  | V11 | 22 |  |

## Use Case 23: Repeat User comes on the different Browser/Incognito & signs up & is on old/existing partner-id

1. Device Id is not available; hence browser drops a new device-id.
2. Device Id & partner id are now available.
3. We fetch the Viu-id as per the rules for the partner-id. we use the last accessed Viu-id in case of multiple Viu-ids
4. If there is a strong auth associated with the partner-id, ask the user to authenticate
5. User then signs up & is a new user.
6. Check
   1. If the Viu-id in context is mapped to any strong auth
      1. If yes, we create a new Viu-id
      2. A new Entry is created with device-id, partner id & strong auth id
   2. If Viu-id in context is not mapped to any strong auth
      1. This strong-auth is linked to this Viu-id
7. This is how data should look

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 |  | V11 | 18 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 | S10 | V11 | 18 | Since there is no strong auth mapped to partner, we map the same Viu-id |
| D8 | P6 |  | V11 | 19 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 |  | V9 | 20 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 | S7 | V16 | 20 | We create a new Viu-id since P8 is already mapped to strong auth (s5) |
| D12 | P6 |  | V11 | 21 | Create a new id, if the user doesn't authenticate |
| D12 | P6 | S5 | V9 | 21 | Fetch the viu-id mapped to strong auth |
| D13 | P6 |  | V11 | 22 |  |
| D14 | P6 |  | V11 | 23 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D14 | P6 | S8 | V17 | 23 | Since P6 is already mapped to strong auth, we create a new Viu-id |

## Use Case 24: Repeat User comes on the different Browser/Incognito & signs in & is on old/existing partner-id

1. Device Id is not available; hence browser drops a new device-id.
2. Device Id & partner id are now available.
3. We fetch the Viu-id as per the rules for the partner-id. we use the last accessed Viu-id in case of multiple Viu-ids
4. If there is a strong auth associated with the partner-id, ask the user to authenticate
5. User then signs in & is a new user.
6. A new Entry is created with device-id, partner id & strong auth id which is mapped to Viu-id based on strong auth

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Device** | **Partner** | **Auth** | **Viu id** | **Use Case #** | **Comment** |
| D1 |  |  | V1 | 1 | New User Comes |
| D2 |  |  | V2 | 2 | Before Signup |
| D2 |  | S1 | V2 | 2 | After Signup. Since V2 is not mapped to any auth or partner, it is linked |
| D3 | P1 |  | V3 | 3 | New Viu-id mapped to partner id is created |
| D4 | P2 |  | V4 | 4 | Before Signup |
| D4 | P2 | S2 | V4 | 4 | Since no strong auth is mapped to V4 & user did a signup we keep the same auth |
| D1 |  |  | V1 | 5 |  |
| D1 |  |  | V1 | 6 |  |
| D1 |  | S3 | V1 | 6 | Strong auth is now linked to existing Viu-id |
| D2 |  | S1 | V2 | 7 |  |
| D3 | P3 |  | V3 | 8 | Fetch the last accessed Viu-id as per the device-id |
| D3 | P4 |  | V3 | 9 | Link the partner-id to device-id & use the last accessed same Viu-id |
| D3 | P4 | S4 | V5 | 9 | Strong auth gets linked on same viu id |
| D1 | P5 |  | V1 | 9 | Last accessed for D1 was V1, hence we map the partner-id |
| D1 | P5 | S40 | V20 | 9 | User doesn’t authenticate |
| D3 | P4 | S40 | V3 | 9 | Link the stong auth |
| D2 | P5 |  | V2 | 10 | Fetch the last accessed Viu-id as per the device-id |
| D2 | P5 | S2 | V4 | 10 | Strong auth id is now mapped to all other ids |
| D5 |  |  | V8 | 11 |  |
| D6 |  |  | V9 | 12 | Since this is a new device id, we will have no way of knowing that this is a repeat user |
| D6 |  | S5 | V9 | 12 | Strong auth gets linked on same viu id |
| D7 |  |  | V10 | 13 |  |
| D7 |  | S5 | V9 | 13 | Viu-id mapped to strong auth is returned |
| D8 | P6 |  | V11 | 14 | New Viu-id is created |
| D9 | P7 |  | V12 | 15 | New Viu-id is created |
| D9 | P7 | S6 | V12 | 15 | Strong auth gets linked on same viu id |
| D10 | P8 |  | V14 | 16 | New Viu-id is created |
| D10 | P8 | S5 | V9 | 16 | Viu-id mapped to strong auth is returned |
| D11 | P6 |  | V11 | 17 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 |  | V11 | 18 | Since there is no strong auth mapped to partner-id, we return the same Viu-id mapped to partner-id. If there had been a strong auth mapped we would have forced user to authenticate (but still use the same Viu-id) |
| D12 | P6 | S10 | V11 | 18 | Since there is no strong auth mapped to partner, we map the same Viu-id |
| D8 | P6 |  | V11 | 19 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 |  | V9 | 20 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D8 | P8 | S7 | V16 | 20 | We create a new Viu-id since P8 is already mapped to strong auth (s5) |
| D12 | P6 |  | V11 | 21 | Create a new id, if the user doesn't authenticate |
| D12 | P6 | S5 | V9 | 21 | Fetch the viu-id mapped to strong auth |
| D13 | P6 |  | V11 | 22 |  |
| D14 | P6 |  | V11 | 23 | Since there is strong auth mapped to the user, we ask the user to authenticate. If the user doesn't authenticate, we still assume it is the same user. |
| D14 | P6 | S8 | V17 | 23 | Since P6 is already mapped to strong auth, we create a new Viu-id |
| D15 | P6 |  | V11 | 24 |  |
| D15 | P6 | S5 | V9 | 24 |  |

# Front End User Stories(TBD)

## Use Case 1: Multiple Partner Ids (this needs to be handled via FE)

For browser this case needs to be handled by front-end. The only use case I can think of is that user is on dual sim phone and he has a cookie with MSISDN1 but he is on data-network with MSISDN2. For such use cases, the front-end will decide & pass only one partner-id to backend.

1. Check if MSISDN/Partner-id is received in HE, then send only this partner-id. No need to look at cookie

2. If MSISDN is not there in HE, then check cookie

\* If only one partner-id present in cookie, send this partner-id

\* If multiple partner-id(s) are present in cookies - This is TBD

## Use Case 2: User does a strong auth & the switch to partner network with an existing partner-id within the session

## Use Case 3 : User does a strong auth & the switch to partner network with a new partner-id within the session

# Viu-id Creation & Fetching Rules



# Backend Scenarios

## Client Passes Device Id only

### No entry found in backend (Use Case#1)

Create a new Viu-id & send it

### One entry found in backend not mapped to any other ids (Use Case#5)

Return this Viu-id.

### Entries found in backend but all of them mapped to some other ids (partner or strong auth)

Create a new Viu-id mapped to just this Device-id & send it.

### Multiple entry found for Device id not attached to any other identifier (partner or strong auth)

Return the last created (latest) Viu-id

## Client Passes Device Id & one Strong Auth Id

### Exact match found with the given Device id & Strong Auth id (Use Case#7)

Return the fetched Viu-id

### Device id not found but one entry found with strong auth (Use Case#13)

Return the Viu-id attached to strong auth

### Device id not found but multiple entries found with same strong auth

Return the Viu-id attached to strong auth

### Device id found but strong auth not found (Use Case#6)

Create new Viu-id attached to strong auth & return it

### Device id not found & strong auth not found (Use Case#12)

Create a new entry in the system & new Viu id

Return this new Viu id

### Multiple Exact match found with the given Device id & Strong Auth id

This is an error scenario

Return the latest entry & since strong auth is same, similar Viu id would be returned

## Client Passes Device Id & MULTIPLE Strong Auth Id

This is not a possible scenario. Front end will pass only one strong auth whichever user is requesting.

## Client Passes Device Id & one Partner id

### Exact match found with the given Device id & partner id

Return the fetched Viu-id

### One entry found with partner id (not attached to strong auth)

Create a new entry in the system with device-id, partner-id & Viu-id mapped to this partner-id

### One entry found with partner id but attached to strong auth

Create a new entry in the system with device-id, partner-id & a new Viu-id

We can’t use strong auth Viu id here as the user hasn’t authenticated

### Partner-id found but multiple entries found with strong auth

This is a conflict

Ask front end to authenticate the user

If front-end doesn’t authenticate, create a new entry in the system with device-id, partner-id & a new Viu-id

### Device id found but partner id not found

Create a new entry and return the new Viu-id

### Device id not found & partner id not found

Create a new entry in the system & new Viu id

Return this new Viu id

### Multiple Exact match found with the given Device id & Strong Auth id

This is an error scenario

Return the latest entry & since strong auth is same, similar Viu id would be returned

## Client Passes Device Id & MULTIPLE Strong Auth Id

This is not a possible scenario. Front end will pass only one strong auth whichever user is requesting.

## Client Passes Device Id & Multiple Partner Ids

In the rule system, we should provide preference at country, device, product & partner level

System according to preference checks if an exact match is found, and should return that

Else based on preference pick the partner id & execute rules for one partner id

This will happen more on app & can’t see this use case for browser. Browser will only pass one partner id

## Client Passes Device Id & Partner Id & Strong Auth ID

### Exact match found

Return this Viu-id

### Exact match not found

Create a new entry in the system mapping all the ids & return the Viu-id mapped to strong auth