ANZ Feedback on Decision Proposal 306 – Updates to Banking Product and Account Detail

9 August 2023

Introduction

ANZ recognises that this decision proposal collects many change proposals with overlaps in requirements and proposed solutions. While the decision proposal has grouped them sensibly, the way in which product and account schemas are shared and inter-relate presents a challenge when attempting to design the schema to suit both purposes i.e., products being offered in the market and customer accounts.

The initial section of this response addresses some key concepts that may not have been apparent when each change proposal making up this decision proposal was considered and for which a data design was proposed. Subsequent sections provide feedback on the individual change proposals, referring to the following sections as appropriate.

Product and Account Structuring

The most significant data design challenge is in the way in which the data standards present:

1. Product and account stages (or phases) – a sequence of offered product (future) or account (current and future) stages, with each representing a current or future state. Examples include:
    a. Home loans with a fixed interest rate for a period that then revert to a variable rate. Related change proposals:
        i. #284 Product Reference Data - revert rates for fixed rate mortgages are absent (group 1, appendix 1.b)
    b. Introductory or 'honeymoon' rates, where a discount (or 'adjustment') is applied for a specified (or customer-negotiated) period, after which the rate reverts to a 'base' rate. With interest rate changes, the base rate may not be the same to which a discount was originally applied. There may be other aspects of the product or account that differ during the introductory period, in fact the interest rates may not differ.

    Related change proposals:

    - Group 2 – #567 BankingProductLendingRateV2 - Lending Rates – FIXED/INTEREST_ONLY period end date cannot be determined (appendix 2)
    - Group 2 – #569 Home Loan Revert rate and product is not available (proposed to be supported by #567)
2. Account collections – a number of accounts that can be managed as a whole e.g., aggregated views with a single balance, integrated transaction history or statements and credit limits managed at a collective 'facility' level. The components that make up the collective 'account' can be on a spectrum of tightly integrated with the industry banking and payment systems and services to those that take no part in them. For example:

a. Homogeneous accounts e.g., term deposits. Individual accounts can be managed through all the traditional channels (customer and internal channels of the data holder) The collective 'account' can be considered a 'portfolio' and individual accounts are available to the full range of industry services – particularly payment services (e.g., BECS, EFTPOS, ATM, Osko, PayTo and international payments and networks including card services). Related change proposal:
    i. Group 4 – #132 Determining Interest Rate of BankingTermDepositAccount (appendix 4)

b. Heterogeneous accounts e.g., agricultural cards and accounts for transaction, saving and lending (including lines of credit) where the collective account is used to manage credit limits in addition to providing aggregated views. The data provider manages the credit facility like those for non-agricultural businesses. Related change proposal:
    i. #62 Multiple Credit Card Account & Loan Support – Not in this decision proposal, but an excellent example (used here) that should also be incorporated. (perlboy, raised 11/12/19, remains open with no DSB feedback or other contributions)

c. Sub-accounts that are inextricably tied to the collective account and are integrated with data holder systems and channels e.g., credit card instalment plans, where a balance transfer or a specific purchase is managed under one account, including payments. Various 'plans' (including buy now, pay later (BNPL)) may also be integrated with industry services e.g., merchant terminals that allow purchases to be put on a new instalment plan at point of sale that is established on the data holder's system. Related change proposal:
    i. Group 3 - #292 Credit card balance plans and payment hierarchy: inadequate information within the CDS (appendix 3)

d. Sub-accounts that allow customers to establish their own sub-accounts with minimal or no interaction with the data holder, and transfer funds between them, with aggregated views of balance and transactions. For example: savings 'jars' or 'buckets' for holidays, home deposit, health, bills, charity. Customers may be able to transfer funds, define regular transfers, sweep amounts or balances on trigger rules including transaction or balance rounding.

An example of the need for a broad-ranging data design process is the proposal in appendix 2 for #567 BankingProductLendingRateV2 - Lending Rates – FIXED/INTEREST_ONLY period end date cannot be determined in group 2. The schema proposed is specific to loans when a more general abstraction would provide more flexibility and extensibility – in this case for sub-accounts. The three-level hierarchy with duplication of loan properties in the objects within the lendingDetail array, together with separate properties for fixedRate, variableRate and fixedCost demonstrates the problem-specific data design and the need for refactoring.

Given the impact of such a significant change to the structure of products and accounts, we propose that a new decision proposal be raised to complete the data design work and allow data holders to model their own products and accounts to ensure they are supported, and to allow data recipients to assess the impact on their data sourcing and presentation. The data design should use general data structures to further the original aims of providing for simplicity, flexibility and extensibility for data producers, consumers and DSB.

Group 1. Get Products and Get Product Detail

#283 Product Reference Data - Residential Mortgage package discounts (Mozo, appendix)

This issue suggests that there are inconsistent interpretations of the Standards in describing loan rates and discounts.

Proposal: Issue #585 has been proposed as a way to provide further consistency in this area. Issue #585 is referenced below with further detail.

Feedback: Refer to #585 Clarify Base and Adjustment Rate Types in group 1.

#284 Product Reference Data - revert rates for fixed rate mortgages are absent (Mozo, appendix 1.b)

Requests inclusion of the current variable rate (or 'revert' rate) of a fixed rate home loan product.

Proposal: Include revertRate in the lendingRates array.

Feedback: We support the principle of this change proposal, but not the solution as proposed in appendix 1.b. We have proposed a more general data design for representing a sequence of product or account stages that provides the details of the rate in subsequent phases – not just for home loans, but also for those after the introductory stage of any other product. This also eliminates the use of the term 'revert', which relates to the transition between stages rather than the before or after the transition stages themselves. Refer to the Product and Account Structuring section above.

#471 Additional credit card fields (Truelayer, appendix 1.a)

Proposes including card information in the Account Detail endpoint, and similar information in Product Reference Data (PRD). Fields applicable to PRD may include the card network, card type and issuer.

Proposal: Add basic card details to the cardArt object.

Feedback: We support the principal of extending this change proposal to PRD, but suggest the following schema that better reflects the meaning and structure of the properties. This should also allow the schema to be shared with BankingAccountDetailV3.BankingCreditCardAccount (see #471 in Group 3).

```
"cardType": {
    "cardType": "CHARGE / CREDIT / DEBIT; mandatory",
    "cardScheme": "AMEX / DINERS / EFTPOS / MASTERCARD / VISA / OTHER; mandatory",
"issuer": "string; mandatory",
"cardImages": [{  // cardArt changed to cardImage to permit plural (array convention)
    "title": "string",
    "imageUri": "string",
    }]
}
```

#531 Define new Loan Repayment Type to relevant schemas (PayPalAustralia, appendix 1.b)

Proposes a new loan repayment type reflecting a fixed cost, rather than an interest rate.

Proposal: Add a FIXED_COST enum to lendingRateType in BankingProductLendingRateV2, which would make a 'rate' conditional. This would also correspond to a 'fixedCost' UType in the account loan object. A BALANCE_TRANSFER type may also be required to associate with the corresponding plan type proposed for credit cards in Account Detail.

Feedback: We do not support this proposal. We agree with PayPal Australia's original proposal that a new enum for **repaymentType** may be required to represent the up-front 'fee' that appears to comprise a repayment against the principal of the business loan and a risk premium. If so, a new PRINCIPAL enum would be required. It's not clear whether the customer-agreed margin on sales ('future receivables' in Terms) made through PayPal is credited against the principal or both principal and interest. If it's the latter, the existing PRINCIPAL_AND_INTEREST would be applicable.

Given an interest rate is not provided in the product's website (the product and fee don't appear to be in the feesAndPricingUri or termsUri links), it's not clear how this decision proposal to change a different property **lendingRateType** to add a FIXED_COST enum and make the property conditional so that the rate is not expected for FIXED_COST provides any meaningful information about the product. The fee is fully described in 'PayPal Working Capital Fixed Fee' event fee (with amount = $0), including '*Don't worry about periodic interest or hidden fees—you pay just one affordable fixed fee you know when you apply*'. Neither the repayment portion of this nor the margin on sales could be considered an interest rate, as there does not appear to be an amount to which an interest rate is applied. Refer to our recent questions and comment in #531 for additional context.

The other part of the proposal (relating to lendingRateType=BALANCE_TRANSFER) suggests that a PayPal account can have multiple business loans that credit funds to the operating account (e.g., for a different purpose after the original loan). This would fit the pattern of an instalment plan. See our response to #292 Credit card balance plans and payment hierarchy: inadequate information within the CDS (group 3). If the existing loan must be extended, this would fit the pattern of separate transaction and loan accounts with traditional draw-downs and credit limit adjustments, which could also be supported by the sub-account structure proposed in the Product and Account Structuring section above.

#585 Clarify Base and Adjustment Rate Types (DSB, appendix 2.a)

Proposes providing additional clarity around the expected use of rate types, by splitting the documentation table into groups of 'base' rate types and 'adjustment' types.

This issue also suggested a potential change to the values provided for rate adjustments, to require a negative sign to be provided with values that are to be deducted from a base rate. This would allow the base and all adjustments to be summed to establish the current 'effective' rate.

Proposal: Update the rate table documentation, including the description of the rate field to clarify that the rate value sign should correspond to the impact of the adjustment on the base rate.

Feedback: We support this change proposal. We agree with Biza in their comment in the change proposal that analysis is required on each of our products to determine whether the customer is provided with a base rate and a discount or adjustment rate. We will not provide rates only used internally for pricing.

For a summary of this group of changes in an example schema, refer to Appendix 1.

In addition to the above issues, the following are also suggested for consideration:

#285 Product Ref Data - leeway within standards makes fee comparison difficult (Mozo, appendix 1.c.2)

Suggests that event-based fee detail may be too broad to compare in a useful way.

Proposal: Is for participants to review the comments provided on issue #285 and provide feedback as to whether a change in this area is appropriate and should be considered in the same version increment.

As the BankingProductFee schema is shared with Account Detail and would also result in a change to that endpoint, a further consideration may be whether the Fee structure should remain common between PRD - where more flexibility in describing them may be required, and Account Detail - where certain values may be known.

Feedback: While we support this proposal in principle, a revised classification scheme is not provided. As experienced by those involved in the initial draft of the data standards where product categories were enumerated and described, it's difficult to produce classification schemes and gain consensus on them. Rather than further overload the existing BankingProductFee.feeType, considerations should be given to a separate property that enumerates the more specific purpose/origin of the fee (as provided as examples in the comment). The introduction of VARIABLE demonstrated how overloading an enumerated property can cause information loss. Any existing fee providing a useful fee category was lost when the feeType was set to VARIABLE. This includes temporal information (e.g., UPFRONT, PERIODIC, EVENT, EXIT) and broader purpose/origin categories for which the list in the comment provide an additional level of detail e.g., DEPOST, WITHDRAWAL, PAYMENT, PURCHASE and TRANSACTION.

We suggest this change proposal be progressed separately from this decision proposal to allow participants to contribute to a proposal for a fee classification scheme.

#387 PRD - Constraint types (Mekaal, 1.c.1)

Analysis of this issue indicated that there may be a need, or benefit in the Standards to describe operational limitations of products, which are distinct from constraints that apply to the origination of a product.

Proposal: Is for participants to review the comments provided on #387 and provide feedback as to whether a change in this area is appropriate and should be considered in the same version increment.

Feedback: As per our comment in the change proposal, our preferred options are:

1. Product option 2 (new featureType of FUNDS_AVAILABLE_AFTER)
2. Product option (new constraintType=OTHER) - for flexibility - in line with other type enumerations in the product schema
3. Account option 2 (reflect funds held in availableBalance) - to avoid a significant schema change

#588 Add structured fields for rate applicability (DSB, appendix 1.c.3)

Analysis of rates conducted as part MI14 found that free-text fields have been used to describe rates that are only available to new customers, or for new accounts.

There are currently no fields designated to convey this type of information with predefined values that would make selection and comparison more effective.

Proposal: Is for participants to review the options provided on #588 and provide feedback as to whether a change in this area is appropriate and should be considered in the same version increment.

Feedback: We support the principal of this proposal but propose a combination of options 2 and 3. The applicabilityConditions property was an array in drafts before v1.0 – as indicated by the name. This is an opportunity to correct that with:

```
"depositRates": [{
    "depositRateType": "BONUS",
    "rate": "string",
    "calculationFrequency": "string",
    "applicationFrequency": "string",
    "applicabilityConditions": [{
         "rateApplicabilityType": "NEW_ACCOUNTS / NEW_CUSTOMERS; optional; default: any",
         "additionalValue": "string; optional",
         "additionalInfo": "string",
         "additionalInfoUri": "string"
    }]
    "tiers": [{
        "name": "string",
        "unitOfMeasure": "DAY",
        "minimumValue": 0,
        "maximumValue": 0,
        "rateApplicationMethod": "PER_TIER",
        "applicabilityConditions": [{
            "rateApplicabilityType": "NEW_ACCOUNTS / NEW_CUSTOMERS; optional; default: any",
            "additionalValue": "string; optional",
            "additionalInfo": "string",
            "additionalInfoUri": "string"
        }],
        "additionalInfo": "string",
        "additionalInfoUri": "string"
    }]
```

This removes the redundancy inherent in option 2 (i.e., a single object can simply be an object in an array) and provides the flexibility of option 3 while supporting specialisation of all attributes at the subordinate level i.e., in specific tiers. We've used that inheritance and specialisation pattern throughout the standards when there's a natural hierarchy.

Group 2. Get Account Detail – loan UType

#531 Define new Loan Repayment Type to relevant schemas (PayPalAustralia, appendix 1.b)

Proposes a new loan repaymentType reflecting a negotiated or accepted fee or rental cost, rather than a fixed or variable interest rate.

Proposal: To add a FIXED_COST enum to lendingRateType in BankingProductLendingRateV2 in PRD, which would allow the 'rate' field to be conditional. This value would then correspond to a new fixedCost UType in the account loan object.

Feedback: We do not support this change, for the reasons outlined in Group 1 – Get Products and Get Product Detail.

#566 Optionality of critical fields is facilitating data quality issues across Data Holder implementations (Adatree)

Suggests that fields currently specified as 'optional' (specifically lendingRate, lendingRates) be made conditional.

Proposal: Issue #567 proposes a way to provide consistency in this area. Issue #567 is referenced below with further detail.

Feedback: Refer to our response to #537 (group 2). We agree with the comment made by perlboy in the change proposal that rate structures for accounts may not be digitally available for all products.

#567 BankingProductLendingRateV2 - Lending Rates – FIXED/INTEREST_ONLY period end date cannot be determined (Sherlok, appendices 2b, 5)

Requests the start and end dates of a fixed rate period, and when an INTEREST_ONLY repaymentType period will end, and –

- feedback suggested that a 'revert' start date may not be necessary if all end dates are provided,
- Standard Variable Rate (SVR) to always be included where applicable,

- proposed raising a new change request (which has become this Decision Proposal) to include the requested details in the loan object. The loan object currently includes the repaymentType field but not loanPurpose.

Proposal: To change the loan UType to an array and include a loanCosts array inside each loan object to provide a clear association between a loan and rates and remove the dependency on the lendingRates schema shared with PRD.

This revised structure would also provide the flexibility for sub-loans suggested as a comment in issue #292.

Feedback: We do not support this change, as outlined in the Product and Account Structuring section relating to product and account stages. The schema proposed in appendix 2 is specific to loans when a more general abstraction would provide more flexibility and extensibility. The addition of a three-level hierarchy with duplication of loan properties in the objects within the lendingDetail array, together with separate properties for fixedRate, variableRate and fixedCost demonstrates the problem-specific data design and the need for refactoring to make it more general. Presumably the three properties share one schema and a subset of those properties are provided for readability.

#569 Home Loan Revert rate and product is not available

Requests details about the 'post expiry' product and 'revert' rate details when applicable.

Feedback suggested that details about both the current and revert 'Discount' rates applicable to an instantated account are important.

Proposal: To implement the proposal above for issue #567 and indicated in Appendix 2.

Feedback: Refer to our response to #567.

#585 Clarify Base and Adjustment Rate Types

This change has been described in Group 1 related to PRD. The change is repeated in this section as it also applies to the description of rates associated with instantiated accounts.

Feedback: We do not support this change, as outlined in the Product and Account Structuring section relating to sub-accounts. The schema proposed in appendix 2 is specific to loans when a more general abstraction would provide more flexibility and extensibility. The addition of a three-level hierarchy with duplication of loan properties in the objects within the lendingDetail array, together with separate properties for fixedRate, variableRate and fixedCost demonstrates the problem-specific data design and the need for refactoring to make it more general. Presumably the three properties share one schema, and a subset of those properties are provided for readability.

For a summary of this group of changes in an example schema, refer to Appendix 2.

In addition to the above issues, the following is also suggested for consideration:

#316 Update description of features[].isActivated to remove default (ANZ, appendix 2.b)

Proposes removing the default 'true' value of isActivated, to allow for an unknown state.

Proposal: Is for participants to review issue #316 and provide feedback as to whether a change in this area is appropriate and should be considered in the same version increment.

Feedback: We support this proposal (as the proposer) and appreciate the contribution of Great Southern Bank.

Group 3. Get Account Detail – creditCard UType

#292 Credit card balance plans and payment hierarchy: inadequate information within the CDS (Stay or Go, appendix 3)

Suggests that detail for common credit card repayment plans is missing, and –

- that lendingRateType could be extended to include BALANCE_TRANSFER and INSTALMENT_PLAN types,
- repayment plan hierarchy may be required,
- detail to indicate how balance transfers may affect interest-free days may be required,
- detail about historical and current 'interest-free days' may be required.

Proposal: For plan detail to be added to the Get Account Detail creditCard UType, with BALANCE_TRANSFER and INSTALMENT_PLAN options, and –

- include a repayment hierarchy field,
- provide detail indicating how balance transfers affect interest-free days,
- include an array to represent interest-free periods,
- include plan rate detail inside the creditCard object to provide a clear association between a plan and a rate to remove the dependency on the lendingRates schema shared with PRD.

Feedback: Refer to the Product and Account Structuring section, specifically relating to sub-accounts. Sub-accounts for a defined amount with a separate repayment schedule and amounts (agreed or calculated) should be considered 'instalment plans'. The purpose of the sub-account (e.g., a balance transfer, purchase, cash advance). Credit card reward schemes are also considered 'plans', so a second major category of 'instalment plan' is more appropriate than an additional four as proposed in appendix 3. An additional category and creditCard UType may be necessary when credit card providers begin to compete with buy now, pay later (BNPL) and pay day lending products and services.

Appendix 3 (Get Account Detail – Additional card plan detail and an array for instalment plans) demonstrates that purchasePlan, cashAdvancePlan, balanceTransferPlan and instalmentPlans should be refactored into a more generic ‘creditCardplan’ schema with UTypes

We agree with CBA in #291 – Credit card loyalty program data: significant gaps and lack of structure (subsumed by #292) that ‘*due to complexity arising from the multitude of loyalty program constructs within the banking sector, and the additional complexity that will be introduced as new sectors, many of which also have loyalty programs, are designated*’. This aligns with our view that the proposal in the ‘Product and Account Structuring’ section (that encompasses this change proposal - #292) should be the subject of a separate decision proposal due to the data design work required (including participants modelling their own products and accounts to ensure they are supported) and the impact of such a significant change to the structure of products and accounts.

#471 Additional credit card fields (Truelayer, appendix 3)

Proposes including the card network, issuer, card type, name on card, valid from/to dates, credit limit and last statement date.

Proposal: For a smaller set of instantiated card details to be added to the Get Account Detail creditCard UType object.

Feedback: We support this change proposal, provided the cardScheme and cardType properties are described in terms of accounts. Since many cards may be supported by the account, the properties in BankingCreditCardAccount must apply to all the cards. In this respect we agree with Westpac in their feedback on this decision proposal.

Refer to our response to #531 in Group 1, which suggests a shared schema with a different structure. There appears to be no reason why cardImages should not be applied to accounts, since there’s no direct relationship to a product in PRD. Presumably multiple images with different resolutions or aspect ratios could be provided in the same way as for PRD products.

```
"creditCard": {
    "cardType": {
        "cardType": "CHARGE / CREDIT / DEBIT; mandatory",
        "cardScheme": "AMEX / DINERS / EFTPOS / MASTERCARD / VISA / OTHER; mandatory",
        "issuer": "string; mandatory",
        "cardImages": [{  // cardArt changed to cardImage to permit plural (array convention)
            "title": "string",
            "imageUri": "string",
            }]
        }
    }
```

For a summary of this group of changes in an example schema, refer to Appendix 3.

Group 4. Get Account Detail – termDeposit UType

#132 Determining Interest Rate of BankingTermDepositAccount (perlboy, appendix 4)

Outlines complexity in determining rates where an account has multiple term deposits.

Proposal: To include rate details inside the termDeposit object to provide a clear association between a deposit and a rate to remove the dependency on the depositRates schema shared with PRD.

Feedback: We do not support this change proposal. Refer to the Product and Account Structuring section, specifically relating to sub-accounts.

While we recognise that termDeposit was changed

For a summary of this group of changes in an example schema, refer to Appendix 4.

Group 5. Get Account Detail – other account types

As the changes described in Group 2, 3 and 4 propose to remove the dependency on the depositRates and lendingRates fields shared with the PRD structure, an additional generic 'otherAccountType' specificAccountUType would need to be defined to provide the equivalent fields for association with other account types.

Proposal: To include a generic account UType and remove the current rate fields that should no longer be required:

- depositRate
- depositRates
- lendingRate
- lendingRates

Feedback: We support this proposal as it relates to the addition of a 'otherAccountType ' specificAccountUType enumeration. We do not support the other data structures in the proposal that have been inherited from proposals in groups 2, 3, and 4, some of which are discussed in the Product and Account Structuring section relating to product and account stages. Examples include depositDetail, lendingDetail, rateEndDate and revertProductId.

For a summary of this group of changes in an example schema, refer to Appendix 5.

Current recommendation

- It is recommended to implement the full set of changes to PRD outlined in **Group 1**, with the option to include the additional issues suggested for consideration.

  Notwithstanding other maintenance or urgent changes, at the date of this proposal, it is expected that these changes would result in the **Get Products** endpoint incrementing to **v4** and **Get Product Detail** incrementing to **v5**.

An example of these changes is provided in **Appendix 1**. It is proposed that these changes be applicable from **Obligation Milestone Y24 #1 11/03/2024**.

- As the changes associated with the **Get Account Detail** endpoint in **Group 2, 3, 4 and 5** are related to the same schema (BankingAccountDetailV3), it is recommended to implement these in a single endpoint version change.

  Notwithstanding other maintenance or urgent changes, it is expected that these changes would result in the **Get Account Detail** endpoint incrementing to **v4**.

  It is acknowledged that this change covers multiple account types and could be delivered as a series of changes over multiple versions if that would be preferable to participants.
  A downside of that approach is that it may result in multiple concurrent versions being defined over a longer period of change.

  The final state of this series of changes would allow for the removal of the dependency on the rate fields shared with PRD. Moving these fields into the respective account UType structures will allow more specific account detail to be defined in future without impact to the PRD rates structures, which are currently tailored to provide product variant search and selection.

  If completed as a single version update, it is proposed that these changes be applicable from **Obligation Milestone Y24 #3 15/07/2024**.

Appendices (original, retained only for cross-referencing)

Appendix 1:

The example schemas below indicate the proposed changes in **bold green text**. Comments in *red*.

1.a. Get Products – Additional card fields (#471)

```
    "cardArt": [{
        "title": "string",
        "imageUri": "string",
        "cardScheme": "AMEX / DINERS / EFTPOS / MASTERCARD / VISA / OTHER; mandatory",
        "issuer": "string; mandatory",
        "cardType": "CHARGE / CREDIT / DEBIT; mandatory"
    }],
```

1.b. Get Product Detail – Additional lending rate types and revert rate field (#284, #531)

```
"lendingRates": [{
    "lendingRateType": "... / VARIABLE / FIXED / ... / FIXED_COST / BALANCE_TRANSFER;
mandatory",
    "rate": "string; conditional: Mandatory unless the lendingRateType “FIXED_COST” is
supplied", Documentation
to specify a negative value for a discount lendingRateType
    "comparisonRate": "string",
    "revertRate": "RateString; optional", The revert rate applicable after the respective rate
expires, for example; INTRODUCTORY, FIXED, or an INTEREST_ONLY type
    "calculationFrequency": "string",
    "applicationFrequency": "string",
    "interestPaymentDue": "IN_ADVANCE",
    "repaymentType":
"INTEREST_ONLY",     "loanPurpose":
"INVESTMENT",
    ...
```

The example schemas below indicate the additional changes for consideration in **bold orange text**.

1.c. Get Product Detail – Additional Feature types, fee types, and rate applicability (#285, #387, #588)

1.c.1. Features (#387 PRD - Constraint types)

```
"features": [{
    "featureType": "... / DIGITAL_BANKING / ... / FUNDS_AVAILABLE_AFTER / MAX_BALANCE /
MAX_LIMIT / MAX_TXNS /
MIN_BALANCE / MIN_LIMIT",
    "additionalValue": "string",
    "additionalInfo": "string",
    "additionalInfoUri": "string"
}],
```

1.c.2. Fees (#285 Product Ref Data - leeway within standards makes fee comparison difficult)

```
"fees": [{
    "name": "string",
    "feeType": "... / PURCHASE / EVENT / ... / EVENT_ATM / EVENT_CARD / EVENT_CORRESPONDENCE",

"amount":
"string",
...
```

1.c.3. Deposit Rates (#588 Add structured fields for rate applicability)

```
"depositRates": [{
    "depositRateType": "BONUS",
    "rate": "string",
    "calculationFrequency": "string",
    "applicationFrequency": "string",
    "tiers": [{
        "name": "string",
        "unitOfMeasure": "DAY",
        "minimumValue": 0,
        "maximumValue": 0,
        "rateApplicationMethod": "PER_TIER",
        "applicabilityConditions": [{
            "rateApplicabilityType": "NEW_ACCOUNTS / NEW_CUSTOMERS; optional; default: any",
            "additionalValue": "string; optional",
            "additionalInfo": "string",
            "additionalInfoUri": "string"
        }],
        "additionalInfo": "string",
        "additionalInfoUri": "string"

}
]
,
.
.
.
```

Appendix 2:

The example schemas below indicate the proposed changes in **bold green text**.
Comments in *red*. Existing fields that have been relocated are indicated in normal green text.

2.a. Get Account Detail – Sub-loan array with rates, adjustments and revert rate specified with each, plus new feature type:

- **#569 Home Loan Revert rate and product is not available**
- **#585 Clarify Base and Adjustment Rate Types**

```
{
    ...
    "productCategory": "RESIDENTIAL_MORTGAGES",
    ...
    "specificAccountUType": "loan",
    "loan": [{ array; conditional
```

```
        "subAccountId": "ID permanence string; mandatory", May enable a reference to a sub-
balance
        "originalStartDate": "string",
        "originalLoanAmount": "string",
        "originalLoanCurrency": "string",
        "loanEndDate": "string",
        "loanOpenStatus": "OPEN / CLOSED; mandatory", Status of this (sub-)account
        "nextInstalmentDate": "string",
        "minInstalmentAmount": "string",
        "minInstalmentCurrency": "string",
        "maxRedraw": "string",
        "maxRedrawCurrency":
"string",         "minRedraw":
"string",
        "minRedrawCurrency": "string",
        "offsetAccountEnabled": true,
        "offsetAccountIds": [
            "string"
        ],
        "lendingDetail": [{ array; mandatory
            "loanCostType": "CURRENT / FUTURE / REVERT; mandatory", Only one CURRENT and one
REVERT type expected, may be multiple FUTURE
            "loanPurpose": "INVESTMENT / OWNER_OCCUPIED; optional; default: any",
            "repaymentType": "INTEREST_ONLY / PRINCIPAL_AND_INTEREST; optional; default:
PRINCIPAL_AND_INTEREST",
            "rateEndDate": "DateTimeString, optional",
            "revertProductId": "string; optional",
            "repaymentUType": "fixedRate / variableRate / fixedCost; mandatory",
            "fixedRate": { object; conditional
                "fixedPeriod": "ExternalRef ISO 8601 Durations; mandatory",
                "standardRate": "RateString; mandatory",
                "effectiveRate": "RateString; mandatory",
                "calculationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "applicationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "interestPaymentDue": "IN_ADVANCE / IN_ARREARS; optional",
                "repaymentFrequency": "ExternalRef ISO 8601 Durations; optional",
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            },
            "variableRate": { object; conditional
                "type": "FLOATING / MARKET_LINKED / VARIABLE; mandatory",
                "standardRate": "RateString; mandatory",
                "effectiveRate": "RateString; mandatory",
                "calculationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "applicationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "interestPaymentDue": "IN_ADVANCE / IN_ARREARS; optional",
                "repaymentFrequency": "ExternalRef ISO 8601 Durations; optional",
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            },
            "fixedCost": { object; conditional
                "amount": "AmountString; mandatory",
                "currency": "CurrencyString; optional; default: AUD",
                "interestPaymentDue": "IN_ADVANCE / IN_ARREARS; optional",
                "repaymentFrequency": "ExternalRef ISO 8601 Durations; optional",
"additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            },
            "adjustments": [{ array; optional
                "type": "BUNDLE_DISCOUNT_FIXED / BUNDLE_DISCOUNT_VARIABLE / DISCOUNT /
INTRODUCTORY / PENALTY /
LVR_THRESHOLD_DISCOUNT; mandatory",
                "amount": "AmountString; optional",
                "currency": "CurrencyString; optional",
                "rate": "RateString; optional",
                "adjustmentPeriod": "ExternalRef ISO 8601 Durations; optional"
                "adjustmentEndDate": "DateTimeString; optional"
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            }]
        }]
    }],
    "features": [{
        "featureType": "... / INSURANCE / ... / LVR_THRESHOLD_DISCOUNT; mandatory",
        "additionalValue": "0.15",
        "additionalInfo": "Rate discount of 0.25% when your LVR reaches a 15% threshold.",
        "additionalInfoUri": "string",
        "isActivated": false
    }]
}
```

The example schemas below indicate the additional changes for consideration in **bold orange text**.

2.b. Get Account Detail – Additional state for the isActivated field to allow, true, false, and indeterminate (#316 Update description of features[].isActivated to remove default)

```
"features": [{
    "featureType": "INSURANCE",
    "additionalValue": "string",
    "additionalInfo": "string",
    "additionalInfoUri": "string",
    "isActivated": true / false / null; boolean; optional; default: indeterminate (ie. null)
}]
```

Appendix 3:

The example schemas below indicate the proposed changes in **bold green text**.
Comments in *red*. Existing fields that have been relocated are indicated in normal
green text.

3. Get Account Detail – Additional card plan detail and an array for instalment plans:

- #292 Credit card balance plans and payment hierarchy: inadequate information within the CDS
- #471 Additional credit card fields

```
{
    ...
    "productCategory": "CRED_AND_CHRG_CARDS",
    ...
    "specificAccountUType": "creditCard",
    "creditCard": {
        "cardScheme": "AMEX / DINERS / EFTPOS / MASTERCARD / VISA; mandatory",
        "cardType": "CREDIT / CHARGE / DEBIT; mandatory",
        "issuer": "string; mandatory",
        "cardLimit": "AmountString; optional",
        "repaymentHierarchy": ["subAccountId", "subAccountId", ...], [string]; mandatory First
sub-account/plan paid off first
        "purchasePlan": { object; optional
            "planName": "string; optional",
            "subAccountId": "ID permanence string; mandatory", May enable a reference to a
sub-balance
            "atPlanExpiryBalanceTransfersTo": "subAccountId",
            "planCreationDate": "DateTimeString; optional",
            "planPeriod": "ExternalRef ISO 8601 Duration; optional",
            "planEndDate": "DateTimeString; optional",
            "planStandardRate": "RateString; mandatory",
            "planEffectiveRate": "RateString; mandatory",
            "minPaymentAmount": "AmountString; optional",
            "paymentDueAmount": "AmountString; optional",
            "paymentCurrency": "CurrencyString; optional; default: AUD",
            "paymentDueDate": "DateTimeString; optional",
            "additionalInfo": "string; optional",
            "additionalInfoUri": "URIString; optional",
            "interestFreePeriods": [{ ;optional array
                "from": "DateTimeString; mandatory",
                "to": "DateTimeString; mandatory"
            }],
            "adjustments": [{ array; optional
                "type": "BUNDLE_DISCOUNT_FIXED / BUNDLE_DISCOUNT_VARIABLE / DISCOUNT /
INTRODUCTORY / PENALTY /
LVR_THRESHOLD_DISCOUNT; mandatory",
                "amount": "AmountString; optional",
                "currency": "CurrencyString; optional; default: AUD",
                "rate": "RateString; optional",
                "adjustmentPeriod": "ExternalRef ISO 8601 Duration; optional"
                "adjustmentEndDate": "DateTimeString; optional"
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            }],
            "planFeatures": [{ array; optional
                "planFeatureType": "INTEREST_FREE / BALANCE_TRANSFER_ENDS_INTEREST_FREE /
INSTALMENTS; mandatory",
                "additionalValue": "string; optional",
                "period": "ExternalRef ISO 8601 Duration; optional",
                "endDate": "DateTimeString; optional"
            }]
        },
        "cashAdvancePlan": { object; optional
            ... ('plan' schema as per purchasePlan)
        },
        "balanceTransferPlan": { object; optional
            ... ('plan' schema as per purchasePlan)
        },
        "instalmentPlans": [{ array; optional
            ... ('plan' schema as per purchasePlan)
        }]
    }...
```

Appendix 4:

The example schemas below indicate the proposed changes in **bold green text**. Comments in *red*. Existing fields that have been relocated are indicated in normal green text.

4. Get Account Detail – Rate detail contained within each sub-term deposit in the array (#132 Determining Interest Rate of BankingTermDepositAccount)

```
{
    ...
    "productCategory": "TERM_DEPOSITS",
    ...
    "specificAccountUType": "termDeposit",
    "termDeposit": [{
            "lodgementDate": "string",
            "maturityDate": "string",
            "maturityAmount": "string",
            "maturityCurrency": "string",
            "maturityInstructions": "HOLD_ON_MATURITY"
        },
        "depositDetail": { object; mandatory
            "subAccountId": "ID permanence string; mandatory", May enable a reference to a
sub-balance
            "depositRateType": "FIXED / FLOATING / MARKET_LINKED / VARIABLE; mandatory",
            "standardRate": "RateString; mandatory",
            "effectiveRate": "RateString; mandatory",
            "calculationFrequency": "ExternalRef ISO 8601 Durations; optional",
            "applicationFrequency": "ExternalRef ISO 8601 Durations; optional",
            "additionalValue": "string; optional",
            "additionalInfo": "string; optional",
            "additionalInfoUri": "URIString; optional",
            "adjustments": [{ array; optional
                "type": "BONUS / BUNDLE_BONUS / INTRODUCTORY; mandatory",
                "amount": "AmountString; optional",
                "currency": "CurrencyString; optional; default: AUD",
                "rate": "RateString; optional",
                "adjustmentPeriod": "ExternalRef ISO 8601 Durations; optional"
                "adjustmentEndDate": "DateString; optional"
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            }]
        },

]
}
```

Appendix 5:

The example schemas below indicate the proposed changes in **bold green text**. Comments in *red*.

5. Get Account Detail

```
{
    ...
    "productCategory": "TRANS_AND_SAVINGS_ACCOUNTS",
    ...
    "specificAccountUType": "otherAccountType",
    "otherAccountType": { object; conditional
        "depositDetail": { object; optional
            "depositRateType": "FIXED / FLOATING / MARKET_LINKED / VARIABLE; mandatory",
            "standardRate": "RateString; mandatory",
            "effectiveRate": "RateString; mandatory",
            "calculationFrequency": "ExternalRef ISO 8601 Durations; optional",
            "applicationFrequency": "ExternalRef ISO 8601 Durations; optional",
            "additionalValue": "string; optional",
            "additionalInfo": "string; optional",
            "additionalInfoUri": "URIString; optional",
            "adjustments": [{ ;optional
                "type": "BONUS / BUNDLE_BONUS / INTRODUCTORY; mandatory",
                "amount": "AmountString; optional",
                "currency": "CurrencyString; optional",
                "rate": "RateString; optional",
                "adjustmentPeriod": "ExternalRef ISO 8601 Durations; optional",
                "adjustmentEndDate": "DateString; optional",
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            }]
        },
        "lendingDetail": [{ array; optional
            "loanCostType": "CURRENT / FUTURE / REVERT; mandatory",
            "loanPurpose": "INVESTMENT / OWNER_OCCUPIED; optional; default: any",
            "repaymentType": "INTEREST_ONLY / PRINCIPAL_AND_INTEREST; optional; default:
PRINCIPAL_AND_INTEREST",
            "rateEndDate": "DateString; optional",
            "revertProductId": "string; optional", Reference to the post-expiry productId.
Expected to be an on-market product in PRD
            "repaymentUType": "fixedRate / variableRate / fixedCost; mandatory",
            "fixedRate": { object; conditional
                "fixedPeriod": "ExternalRef ISO 8601 Durations; mandatory",
                "standardRate": "RateString; mandatory",
                "effectiveRate": "RateString; mandatory",
                "calculationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "applicationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "interestPaymentDue": "IN_ADVANCE / IN_ARREARS; optional",
                "repaymentFrequency": "ExternalRef ISO 8601 Durations; optional",
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            },
            "variableRate": { object; conditional
                "type": "FLOATING / MARKET_LINKED / VARIABLE; mandatory",
                "standardRate": "RateString; mandatory",
                "effectiveRate": "RateString; mandatory",
                "calculationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "applicationFrequency": "ExternalRef ISO 8601 Durations; optional",
                "interestPaymentDue": "IN_ADVANCE / IN_ARREARS; optional",
                "repaymentFrequency": "ExternalRef ISO 8601 Durations; optional",
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            },
            "fixedCost": { object; conditional
                "amount": "AmountString; mandatory",
                "currency": "CurrencyString; optional; default: AUD",
                "interestPaymentDue": "IN_ADVANCE / IN_ARREARS; optional",
                "repaymentFrequency": "ExternalRef ISO 8601 Durations; optional",
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            },
            "adjustments": [{ array; optional
                "type": "BUNDLE_DISCOUNT_FIXED / BUNDLE_DISCOUNT_VARIABLE / DISCOUNT /
INTRODUCTORY / PENALTY /
LVR_THRESHOLD_DISCOUNT; mandatory",
                "amount": "AmountString; optional",
                "currency": "CurrencyString; optional",
                "rate": "RateString; optional",
                "adjustmentPeriod": "ExternalRef ISO 8601 Durations; optional"
                "adjustmentEndDate": "DateTimeString; optional"
                "additionalInfo": "string; optional",
                "additionalInfoUri": "URIString; optional"
            }]
        }]
    }
}
```