

Xero API Invoice Creation

1. Business Logic Summary

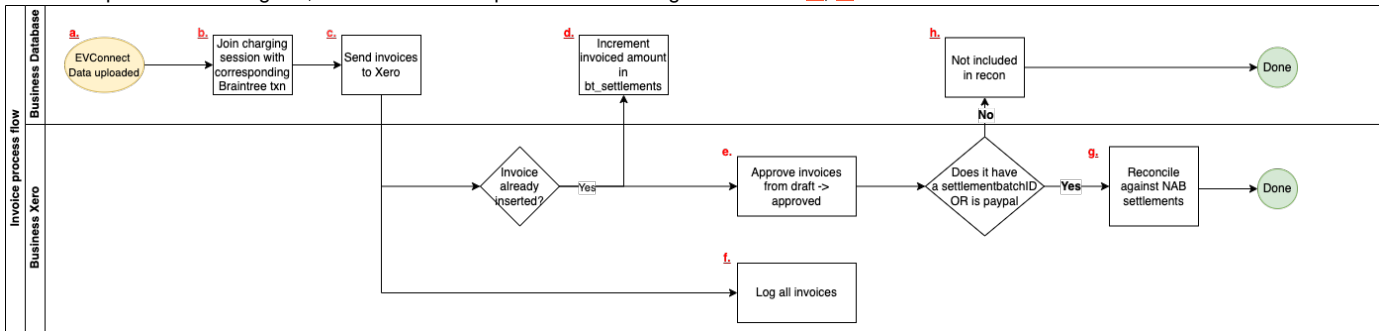
The primary function of this service is to facilitate a hands off reconciliation process for charging sessions (all sessions that exist in EV Connect) and their associated payments in braintree.

The process will ensure that a Xero invoice is created for EVERY EV Connect transaction and that have a corresponding transaction ID in Braintree that it is associated with that transaction ID.

At a high level, this is what the process flow looks like.

1.1 Process flow

Below is a process flow diagram, and a table that expands on each stage as labelled a., b. etc...



Decision tree for business logic in the service

Item	Description
a. EVConnect data uploaded	<ul style="list-style-type: none">EVConnect data is uploaded on a daily basis from evconnect.The invoicing process runs every day, with a lookback window of between 2 and 4 days
b. Compare EV Connect session data to Braintree data	<ul style="list-style-type: none">A view in mongoDB has been created that links the transaction id in the braintree transaction object to the transaction id in the EVConnect charging session
b.1 Financial transactions have a Xero invoice created	<ul style="list-style-type: none">Data that we need to invoice with the correct tracking options in Xero sourced from EV Connect and BraintreeOnly transactions that are in a settled state or transaction declined state (i.e finalised in braintree) are included in the process. It is up to the business (us) to recharge customers that have incurred a declined transaction. Settling transactions will be processed the next time the process runs as they have not yet been finalised in braintree.At the end of the month ALL Braintree txns will exist in Xero.<ul style="list-style-type: none">Any EV Connect session data not included in Braintree is processed as an NON financial transaction and follows the NON financial transaction data rules
b.2 Non financial transaction have a Xero invoice created	<ul style="list-style-type: none">All EV Connect transactions that do not exist in Braintree will have a Xero invoice created with the following rules<ul style="list-style-type: none">Transaction ID: <i>a1ff7b70-f69f-4aaf-b214-bc243ee55873 (Session Id from EVC)</i>Reference: 09-Jan-22 - [{"Driver Group Type}" OR "Missing Braintree"]Description: Whatever relevant info for that txn typeAt the end of the month all non-financial txns will exist in Xero
c. Send Invoices to Xero	<ul style="list-style-type: none">All the transactions that lie within the daterange of the EVConnect session upload
d. Increment invoiced amount in bt_settlements	<ul style="list-style-type: none">When xero successfully inserts an invoice we find the payment type and the batch date and increment the invoiced amount for reporting purposesThis is done to ensure that all Braintree settlements are associated with transactions. This will allow for easy matching in Xero

e. Approve invoices from draft to approved	<ul style="list-style-type: none"> At this stage, this is where the company Accountant checks through the transactions to make sure the data quality is good - and subsequently approves the transactions to be reconciled
f. Log all invoices	<ul style="list-style-type: none"> We log all invoices that were attempted to be pushed into Xero - whether the attempt was successful or not. This is used to send log files per run of the service
g. Reconcile against NAB settlements	<ul style="list-style-type: none"> At this stage, all invoices needed for reconciliation should be in Xero. See part 1.6 for more details on this.
h. End of month reconciliation report created	<ul style="list-style-type: none"> To ensure that all records from EV Connect and Braintree have been processed a report will be generated to demonstrate that ALL EV Connect and Braintree transactions have a Xero record created

1.2 Frequency

The entire process is triggered by the weekly upload of EVConnect sessions to our database, upon which the service will fetch the sessions corresponding transactions (if available).

1.3 Inputs

```

{
  "_id": {
    "$oid": "62b389dd4c114483a185d424"
  },
  "sessionId": "56f8c03c-855f-4a95-a0e1-5971492dd7fb",
  "chargeBoxId": "veefil-62100370",
  "chargeDuration": 36.044716666666666,
  "chargeSessionAmount": 10.84,
  "connectedTime": "2022-04-30 07:44:13",
  "connectorDurationMin": 36.044716666666666,
  "connector_id": "25646311-3885-4cff-b466-cb2dc667ca10",
  "disconnectedTime": "2022-04-30 08:20:16",
  "discountRatePercentage": 0,
  "driverId": "f40c8ca1-3126-4c92-b4ba-2b1ee762490a",
  "evseId": "25646311-3885-4cff-b466-cb2dc667ca10",
  "idTag": "SQ3P57QQXZI1IMIXX4MI",
  "location_name": "Coles Schofields - MS042",
  "netAmount": 10.84,
  "networkId": "evie",
  "organizationId": "delce940-8602-492a-bc62-5dcb01706a4a",
  "portId": "627c9a39-25a0-4b80-8658-7ea710450200",
  "salesTaxAmount": 0,
  "sessionEnd": {
    "$date": {
      "$numberLong": "1651306816000"
    }
  },
  "sessionFeeAmount": 0,
  "sessionFeeMinimum": 0,
  "sessionFeePercentage": 0,
  "sessionStart": {
    "$date": {
      "$numberLong": "1651304653000"
    }
  },
  "sessionStatus": "PAID",
  "siteId": "b693c279-c0ce-469e-8db3-6f0720fdd7ac",
  "stationName": "Coles Schofields - MS042A",
  "totalTransactedAmountDriver": 10.84,
  "transactionId": "lgs8b63j",
  "energyProvided": 27092,
  "driverGroup": null,
  "planId": null,
  "socEnd": 99,
  "socStart": 0
}

```

Example of a session in evc-sessions

```
{
  "_id": {
    "$oid": "62b389dd4c114483a185d424"
  },
  "session_id": "56f8c03c-855f-4a95-a0e1-5971492dd7fb",
  "xero_id": "56f8c03c-855f-4a95-a0e1-5971492dd7fb",
  "amount_net": 10.84,
  "charge_box_id": "veefil-62100370",
  "charge_session_amount": 10.84,
  "amount_transacted": 10.84,
  "total_txn_amount": 10.84,
  "power_provided_wh": 27092,
  "driverId": "f40c8ca1-3126-4c92-b4ba-2b1ee762490a",
  "session_status": "PAID",
  "location_name": "Coles Schofields - MS042",
  "discount": 0,
  "station_name": "Coles Schofields - MS042A",
  "transaction_id": "lgs8b63j",
  "connected_time": {
    "$date": {
      "$numberLong": "1651304653000"
    }
  },
  "disconnected_time": {
    "$date": {
      "$numberLong": "1651306816000"
    }
  },
  "charge_duration_min": 36.044716666666666,
  "connected_duration_min": 36.044716666666666,
  "connector_id": "25646311-3885-4cff-b466-cb2dc667ca10",
  "isInvoiced": false,
  "driver_group": null,
  "station_id": "MS042",
  "station_type": "Metro",
  "organisation_id": "delce940-8602-492a-bc62-5dcb01706a4a",
  "account_code": 220,
  "settlementBatchId": "2022-04-30_evienetworksAUD_gn4jyjex",
  "dateTimeBatchDate": {
    "$date": {
      "$numberLong": "1651276800000"
    }
  },
  "status": "settled",
  "type": "sale",
  "cardType": "Visa",
  "id": "lgs8b63j",
  "dateTimeCreatedUTC": {
    "$date": {
```

```

    "$numberLong": "1651306819000"
  },
  "networkUser": "Geoff.Brady@goevie.com.au",
  "amount": 10.84,
  "date": {
    "$date": {
      "$numberLong": "1651304653000"
    }
  },
  "dateCreated": {
    "$date": {
      "$numberLong": "1651304653000"
    }
  },
  "planId": null
}

```

Example of a joined document in evc-sessions_financial

1.4 Business rules

1.4.1 Invoicing Requirements

Business Requirement	Reason	How it is achieved	Exceptions
Invoice is <u>Tax Inclusive</u>	Invoice amounts in braintree are tax inclusive - so no tax is needed to be added on top	xero-node-invoice-service/services/buildQuery.ts - Line 156	N/A
Invoice is inserted as a <u>DRAFT</u>	So invoices can be deleted before approval if needed	xero-node-invoice-service/services/buildQuery.ts - Line 164	N/A
Invoice is mapped to the correct tracking categories	So finance can track revenue by Site and Staff Group/Team Each site has a group Highway, Metro, CVGA, Actew assigned. It also has a account code and particular site ie 200, HS008	https://github.com/goevie/xero-node-invoice-service/blob/main/functions/findTrackingOptionID.ts	N/A
Invoices are loaded on a daily	To retry failed transactions as quickly as possible	EVC sends us a file via sftp every day	N/A

1.4.2 Data Mapping requirements

Business Requirement	Reason	How it is achieved	Exceptions
Charging session transaction_id must equal the braintree id	So charging sessions can be mapped to transactions	the evc-sessions_financial view in mongoDB	If transaction ID is missing from the charging session
Objects in evc-sessions_financial must map to a station object in stations	So we can extract the tracking info for Xero	Use charging_session.siteId to join the charging session with the locations_api_w_acc_code1 view in mongoDB	If charging session is missing, station will not be found

1.4.3 Simplified transformation pipeline applied to braintree_txns2 to create the evc-sessions_financial view

NOTE: omitted some steps for the sake of simplicity

```
// join by transaction id
```

```

[ {
  $lookup: {
    from: 'braintree_txns2',
    localField: 'transactionId',
    foreignField: 'id',
    pipeline: [
      {
        $limit: 1
      }
    ],
    as: 'payment'
  }
},
// join by siteId
{
  $lookup: {
    from: 'locations_api_w_acc_code1',
    localField: 'siteId',
    foreignField: 'externalId',
    pipeline: [
      {
        $limit: 1
      }
    ],
    as: 'station'
  }
},
// get InsertResult
{
  $lookup: {
    from: 'sessions_xero_mapping',
    localField: 'sessionId',
    foreignField: 'session_id',
    as: 'xero_res',
    pipeline: [
      {
        $limit: 1
      }
    ]
  }
}, {
  $addFields: {
    station: {
      $arrayElemAt: [
        '$station',
        0
      ]
    },
    payment: {
      $arrayElemAt: [

```

```

        '$payment',
        0
    ]
},
xero_res: {
    $arrayElemAt: [
        '$xero_res',
        0
    ]
}
},
// mapping
{
    $project: {
        deprecated_session_id: '$oldSessionId.oldSessionId',
        session_id: '$sessionId',
        xero_id: '$sessionId',
        amount_net: '$netAmount',
        charge_box_id: '$chargeBoxId',
        charge_session_amount: '$chargeSessionAmount',
        amount_transacted: '$chargeSessionAmount',
        total_txn_amount: '$totalTransactedAmountDriver',
        power_provided_wh: '$energyProvided',
        driverId: '$driverId',
        session_status: '$sessionStatus',
        location_name: '$location_name',
        discount: '$discountRatePercentage',
        station_name: '$stationName',
        transaction_id: '$transactionId',
        connected_time: '$sessionStart',
        disconnected_time: '$sessionEnd',
        charge_duration_min: '$chargeDuration',
        connected_duration_min: '$connectorDurationMin',
        connector_id: '$connector_id',
        plan: '$plan',
        xero_res: '$xero_res',
        isInvoiced: {
            $ifNull: [
                false,
                false
            ]
        },
        driver_group: '$driverGroup',
        station_id: '$station.id',
        station_type: '$station.group',
        organisation_id: '$station.organizationId',
        account_code: '$station.account_code',
        settlementBatchId: '$payment.settlementBatchId',
        dateTimeBatchDate: '$payment.dateTimeBatchDate',
    }
}

```

```

    status: '$payment.status',
    type: '$payment.type',
    cardType: '$payment.cardType',
    id: '$payment.id',
    dateTimeCreatedUTC: '$payment.dateTimeCreatedUTC',
    networkUser: '$payment.networkUser',
    amount: '$payment.amount',
    date: '$sessionStart',
    dateCreated: '$sessionStart',
    planId: '$planId'
  }
}]

```

1.5 Output

```

{
  "_id": {
    "$oid": "62b389dd4c114483a185d424"
  },
  "session_id": "56f8c03c-855f-4a95-a0e1-5971492dd7fb",
  "xero_id": "56f8c03c-855f-4a95-a0e1-5971492dd7fb",
  "amount_net": 10.84,
  "charge_box_id": "veefil-62100370",
  "charge_session_amount": 10.84,
  "amount_transacted": 10.84,
  "total_txn_amount": 10.84,
  "power_provided_wh": 27092,
  "driverId": "f40c8ca1-3126-4c92-b4ba-2b1ee762490a",
  "session_status": "PAID",
  "location_name": "Coles Schofields - MS042",
  "discount": 0,
  "station_name": "Coles Schofields - MS042A",
  "transaction_id": "lgs8b63j",
  "connected_time": {
    "$date": {
      "$numberLong": "1651304653000"
    }
  },
  "disconnected_time": {
    "$date": {
      "$numberLong": "1651306816000"
    }
  },
  "charge_duration_min": 36.044716666666666,
  "connected_duration_min": 36.044716666666666,
  "connector_id": "25646311-3885-4cff-b466-cb2dc667ca10",
  "isInvoiced": false,

```



```

"driver_group": null,
"station_id": "MS042",
"station_type": "Metro",
"organisation_id": "delce940-8602-492a-bc62-5dcb01706a4a",
"account_code": 220,
"settlementBatchId": "2022-04-30_evienetworksAUD_gn4jyjex",
"dateTimeBatchDate": {
  "$date": {
    "$numberLong": "1651276800000"
  }
},
"status": "settled",
"type": "sale",
"cardType": "Visa",
"id": "lgs8b63j",
"dateTimeCreatedUTC": {
  "$date": {
    "$numberLong": "1651306819000"
  }
},
"networkUser": "Geoff.Brady@goevie.com.au",
"amount": 10.84,
"date": {
  "$date": {
    "$numberLong": "1651304653000"
  }
},
"dateCreated": {
  "$date": {
    "$numberLong": "1651304653000"
  }
},
"planId": null
}

```

Typical Transaction

```
{
  "_id": {
    "$oid": "62b389dd4c114483a185d49a"
  },
  "session_id": "49cbcfff-55e4-4efc-adb9-27f5fac49077",
  "xero_id": "49cbcfff-55e4-4efc-adb9-27f5fac49077",
  "amount_net": 0,
  "charge_box_id": "veefil-62100144",
  "charge_session_amount": 8.2,
  "amount_transacted": 8.2,
  "total_txn_amount": 0,
  "power_provided_wh": 20503,
  "driverId": "237bbb06-764f-4fa0-9d3c-88ea874fba5c",
  "session_status": "PAID",
  "location_name": "CTR Kyneton CV020A",
  "discount": 100,
  "station_name": "CTR Kyneton CV020A",
  "transaction_id": "00000000",
  "connected_time": {
    "$date": {
      "$numberLong": "1635027711000"
    }
  },
  "disconnected_time": {
    "$date": {
      "$numberLong": "1635030226000"
    }
  },
  "charge_duration_min": 41.906666666666666,
  "connected_duration_min": 41.906666666666666,
  "connector_id": "79157f89-3bd0-4514-919e-1241f9f27c15",
  "isInvoiced": false,
  "driver_group": "CVGA Free Rating",
  "station_id": "CV020",
  "station_type": "CVGA",
  "organisation_id": "847eac07-29c4-4890-8dae-022c670ca3ed",
  "account_code": 593,
  "date": {
    "$date": {
      "$numberLong": "1635027711000"
    }
  },
  "dateCreated": {
    "$date": {
      "$numberLong": "1635027711000"
    }
  },
  "planId": "9654ba83-f3d0-4092-a28a-47d9dc5c67e9"
}
```

Driver Group Transaction

Draft

Preview Email Print PDF Invoice Options

To: [rob@intheblack.c...](#) x Date: 12 Jan 2021 Due Date: 12 Jan 2021 Invoice #: pxj9rkba Reference: 2021-01-12 Visa Branding: Standard

AUD Australian Dollar Amounts are: Tax Inclusive

Item	Description	Qty	Unit Price	Disc %	Account	Tax Rate	Evie Sites	Staff ...	Amount AUD
	sale - settled - Visa	1.00	9.10		220 - Metro	GST on Income	MB018	Metro	9.10

Add a new line

Subtotal: 9.10
Includes GST 10.00%: 0.83

TOTAL 9.10

Save Approve Cancel

1.6 Final reconciliation

The primary objective of this final reconciliation process is to demonstrate that every transaction in EV Connect has been loaded into Xero. This is demonstrated in the process below.

1. Search the date as YYYY-MM-DD in the "Search by name or reference tab" like below:

Options Match Discuss

1 Sep 2021 [More details](#) Spent Received 242.03

Find & select matching transactions below

1. Find & select matching transactions ? ☐ Show Spent Items ☒ Show AUD items only

Search by name or reference Search by amount

2021-09-01 Go Clear search

Date	Name ↓	Reference	Spent	Received
<input checked="" type="checkbox"/> 1 Sep 2021	paige.loadsman97@hotmail.com	Opfbd5p7 2021-09-01 MasterCa...		6.17 AUD
<input checked="" type="checkbox"/> 1 Sep 2021	picsbystuart@gmail.com	hpbeq7z1 2021-09-01 MasterCa...		11.23 AUD
<input checked="" type="checkbox"/> 1 Sep 2021	amandaw@aeutas.org.au	hcz0jsx 2021-09-01 Visa		15.63 AUD
<input checked="" type="checkbox"/> 1 Sep 2021	OERNZQEVCY@evconnect.com	ezwpgddm 2021-09-01 apple_pa...		5.62 AUD
<input checked="" type="checkbox"/> 1 Sep 2021	mgascoyne@neato.com.au	ccw182s7 2021-09-01 MasterCa...		11.12 AUD
<input checked="" type="checkbox"/> Select all on this page				Showing 1 - 5 of 5

- a. Then click "select all on this page"
- b. All in order, the selected amount should match the settled amount in NAB

1.7 Invoice tracking

I track the invoiced amount sent to Xero in the business database as seen below:

```

settlement_date: "2021-09-20"
expected_amount: 189.87
apple_pay_mastercard: 39.27
apple_pay_visa: 23.02
mastercard: 62.8
paypal: 47.7
visa: 64.78
other_instrument: 0
invoiced_paypal: 47.7
invoiced_amount: 189.87

```

settlement_date : The settlement date in NAB

expected_amount : Amount settled in NAB - amount invoiced though match this number

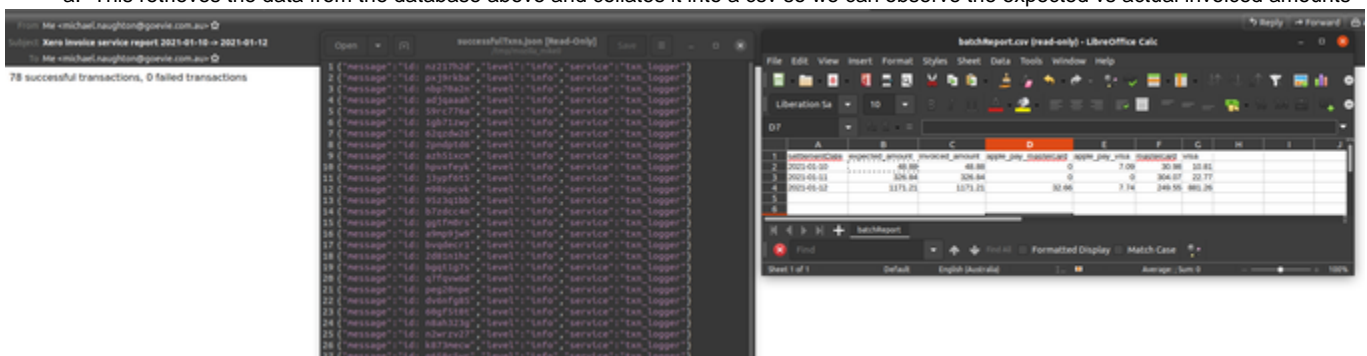
invoiced_amount : The sum of payments sent to Xero that are included in the NAB recon

invoiced_paypal : The sum of payments sent to Xero that are paypal on that date

1.8 Reporting system

The current reporting email structure sends an email with 3 files:

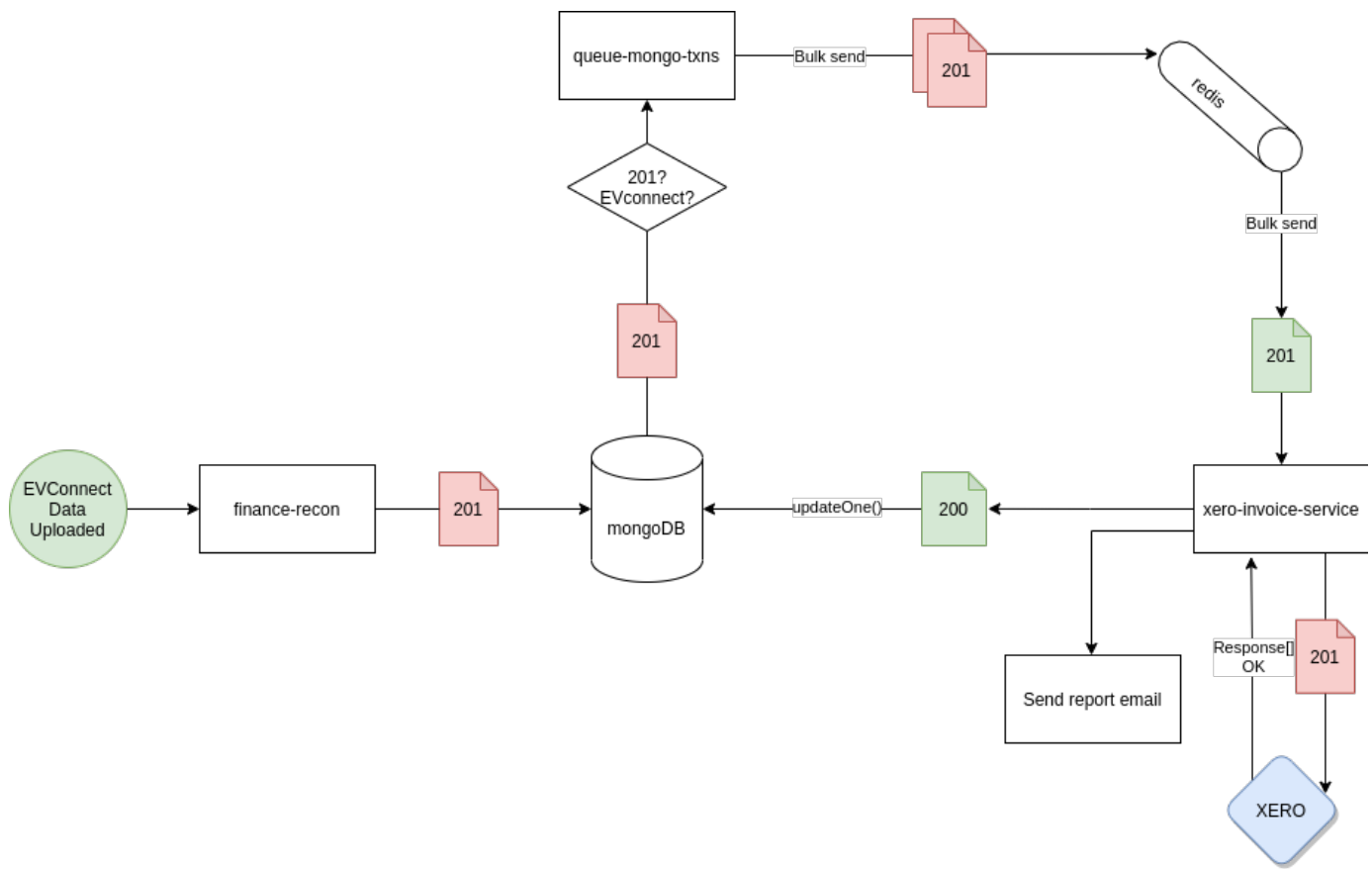
1. successfulTxns.json
 - a. This file is a log of all transactions sent to Xero successfully
2. failedTxns.json
 - a. A log of transactions that did not make it to Xero and the supposed reason
3. batchReport.csv
 - a. This retrieves the data from the database above and collates it into a csv so we can observe the expected vs actual invoiced amounts



4. We also have a DOMO report which tracks all variances across a variable date range here: <https://goevie-com-au.domo.com/kpis/details/1188227432>

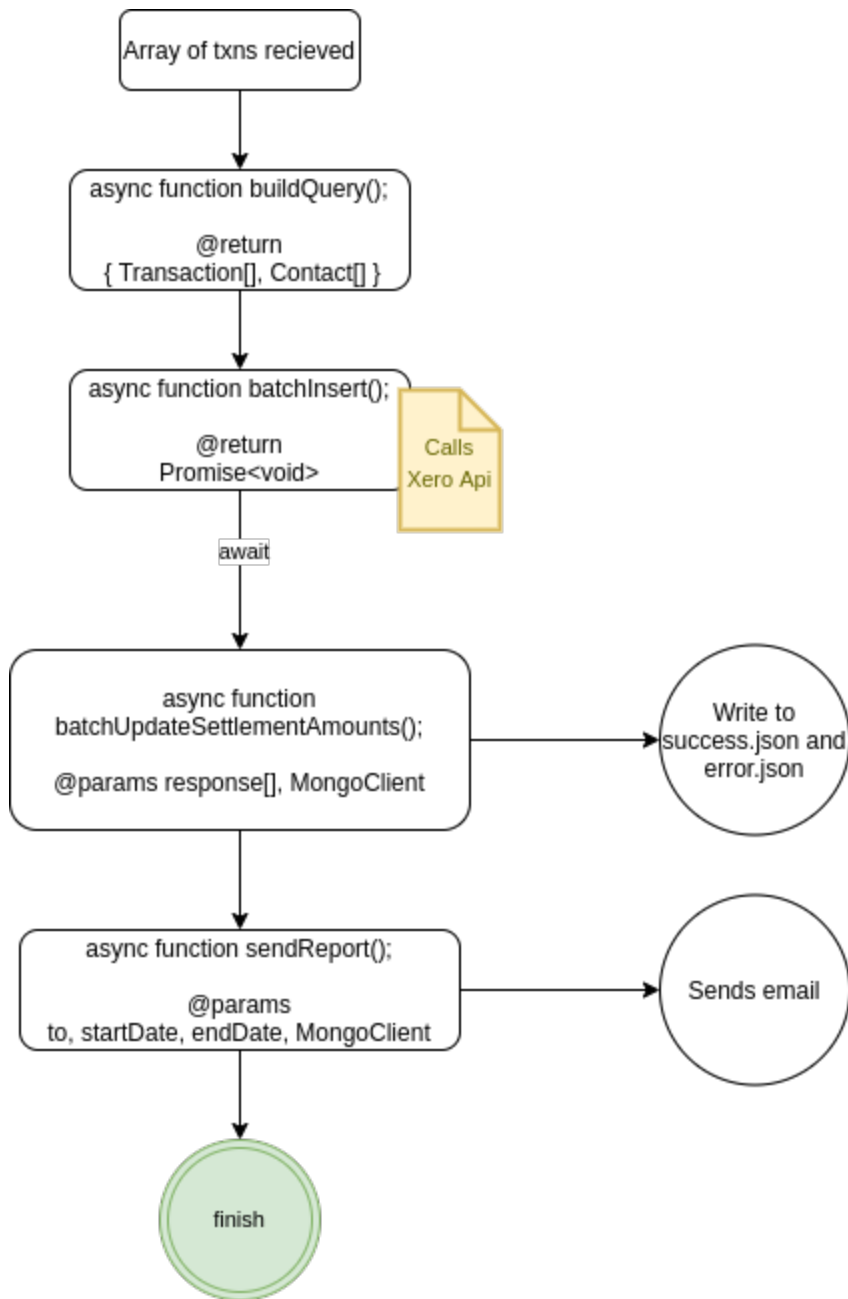
2. Technical Overview

2.1 Current Software Architecture



Architecture Diagram for the flow of transactions through the system

2.2 Process flow



Process flow of the data through the service

2.3 What happens when an invoice is sent to Xero

Read more about the Xero API here

- <https://developer.xero.com/documentation/api/accounting/overview/>

It is important to note that Xero requires a unique Contact Name.

It is due to this reason that we are using customer email to as the contact name in Xero.

Once an invoice **is** successfully created in Xero we:

1. Log the successful transactions in reports/success.json
2. Update the statusCode of the invoice to 200 (completed) in the collection `braintree_txns2`
 - a. `queue-txns-service` now does **not** pull this transaction again

If a transaction **is not** successfully created in Xero we:

1. Log the unsuccessful transactions in reports/error.json
2. Continue

3. Service Description

3.1 finance-recon

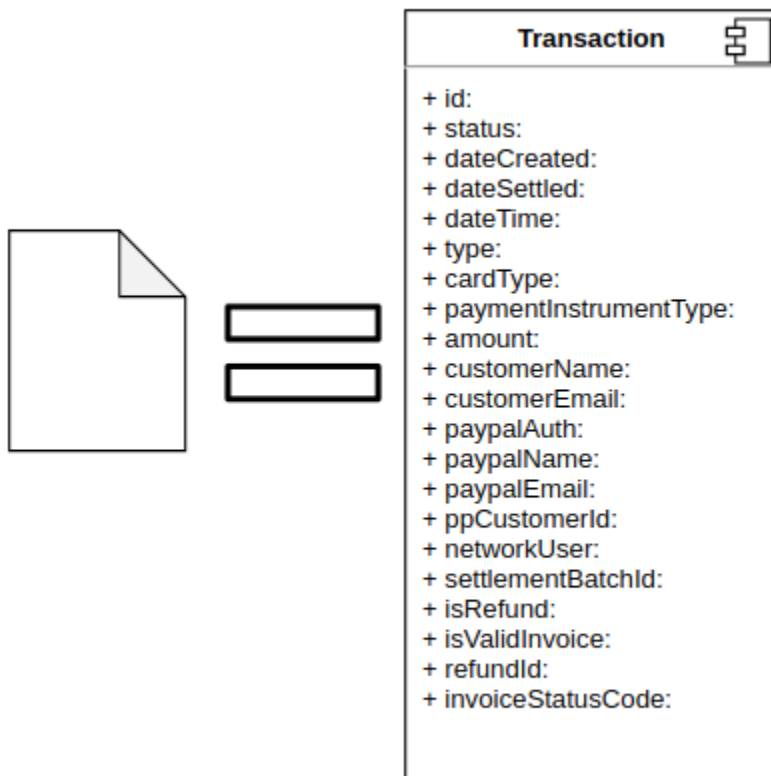
Sends braintree data to mongoDB when EVConnect data is uploaded into the smtp folder in our server. When in mongo, each unique transaction is matched with its corresponding charging session and station.



This service has one endpoint

1. <http://finance-recon:3000/jobs/braintreeToMongo?startDate=YYYY-MM-DD&endDate=YYYY-MM-DD>

Transactions from braintree initially are in the form as seen below.




BT Transaction document schema

When the join occurs in the mongo view **bt_txns_final** we can extract the data necessary for the invoice creation.

3.2 queue-txns-service

This service pulls data from **bt_txns_final2** and sends them to a worker queue to be picked up by the xero invoice service.

 This service has one endpoint:

1. <http://queue-txns-service:3000/jobs/mongoTxnsToRedis?startDate=YYYY-MM-DD&endDate=YYYY-MM-DD>

Sender	Recipients
jobs/mongoTxnsToRedis on node-red:8882	<ul style="list-style-type: none">• xero-invoice-service<ul style="list-style-type: none">• job.name === "invoice-txns"• Consumers:<ul style="list-style-type: none">• buildAndInsert() in index.ts

The transactions are sent in the form below:

```
interface Transaction {
  // required
  readonly transactionId: string;
  readonly customerName: string;
  readonly paymentType: string;
  readonly batchDate: string;
  customerEmail: string;
  amount: number;
  accountCode: number;
  dateCreated: Date;
  description: string;
  siteName: string;
  stationType: string;
}
```

One message is sent to the queue, comprised of the following:

```
/**
 * Creates a Transaction[] to be sent to the queue
 *
 * @export
 * @param {*} txnArray
 * @param {string} startDate
 * @param {string} endDate
 */

await queue.add(
  "invoice-txns",
  { invoices, startDate, endDate },
  { removeOnComplete: 2000, }
);
```


This service uses a bullmq worker to take messages off the queue sent by **queue-txns-service**. It is also responsible for sending report emails to a chosen recipient.

```
if (job.name === "invoice-txns") {  
  try {  
    await buildAndInsert(  
      job.data.invoices,  
      job.data.startDate,  
      job.data.endDate,  
      client,  
      xero  
    );  
  } catch (err) {  
    logger.error(`${JSON.stringify(err, null, 2)}`);  
  }  
}
```

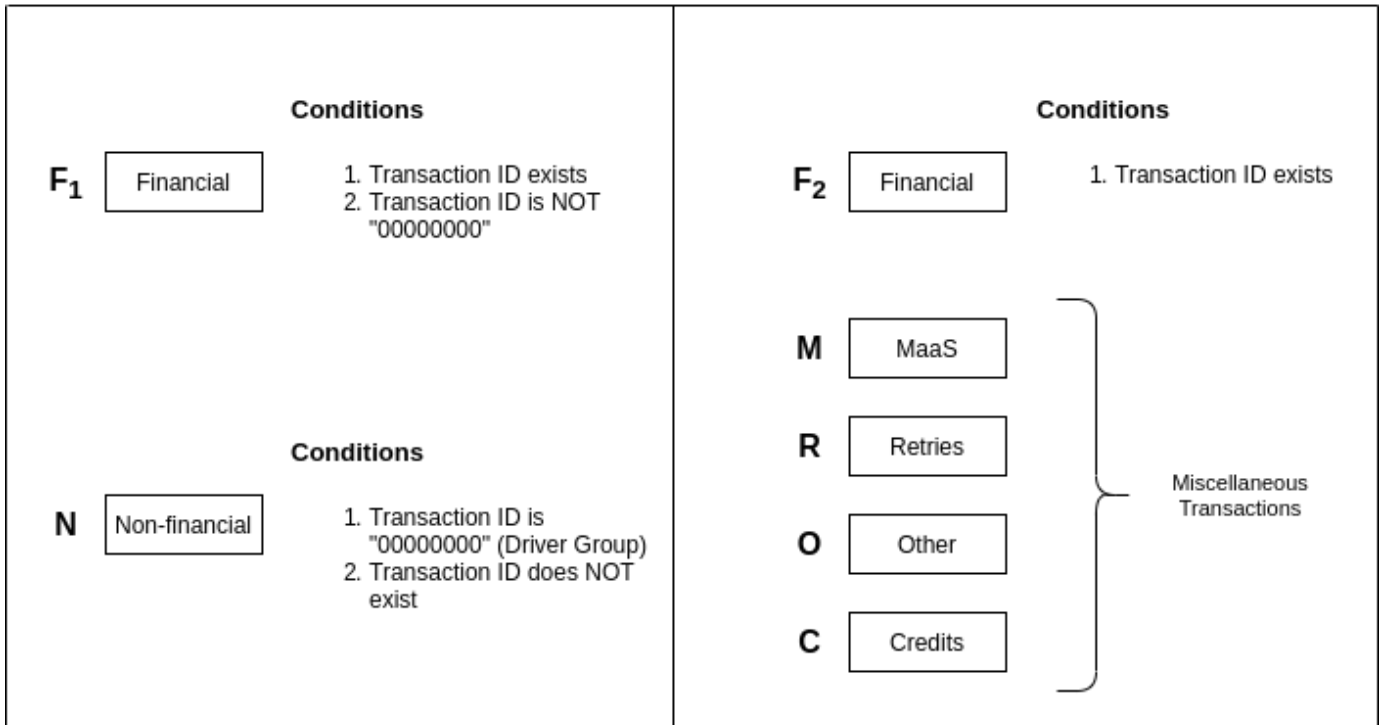
Sender	Recipients
buildAndInsert() in index.ts	Xero service account

EV Connect

Stored in mongo collection
sessions new

Braintree

Stored in mongo collection
bt txns final



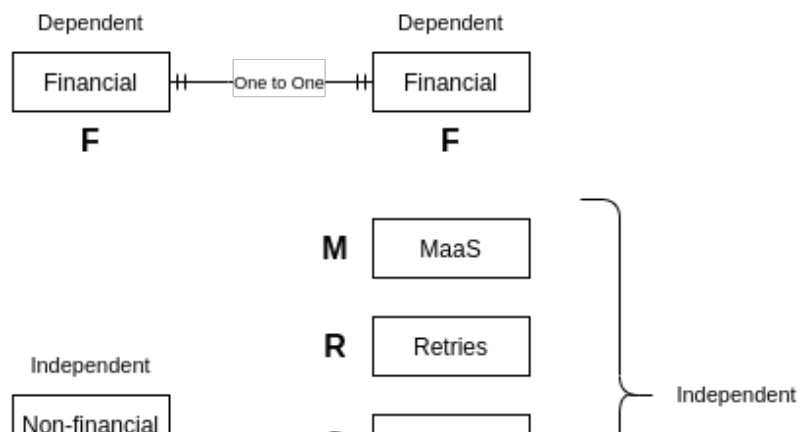
Xero

$$F_1 = F_2 = F$$

$$F + N + M + R + O + C = X$$

Relationships

All things in order, these relationships should be fulfilled



N

O

Other

C

Credits