00/00/0000 - [DRAFT] Task Management Service

Contributors

Name	Role	Company
Rob Bowman	Integration Architect	ARRT
Dan Gater	Business Analyst	Create Cadence
Dan Bovenzi	Integration Architect	ARRT

Contents

Overview

The rationale and requirements for the service are described in a TDA presentation that can be found: here
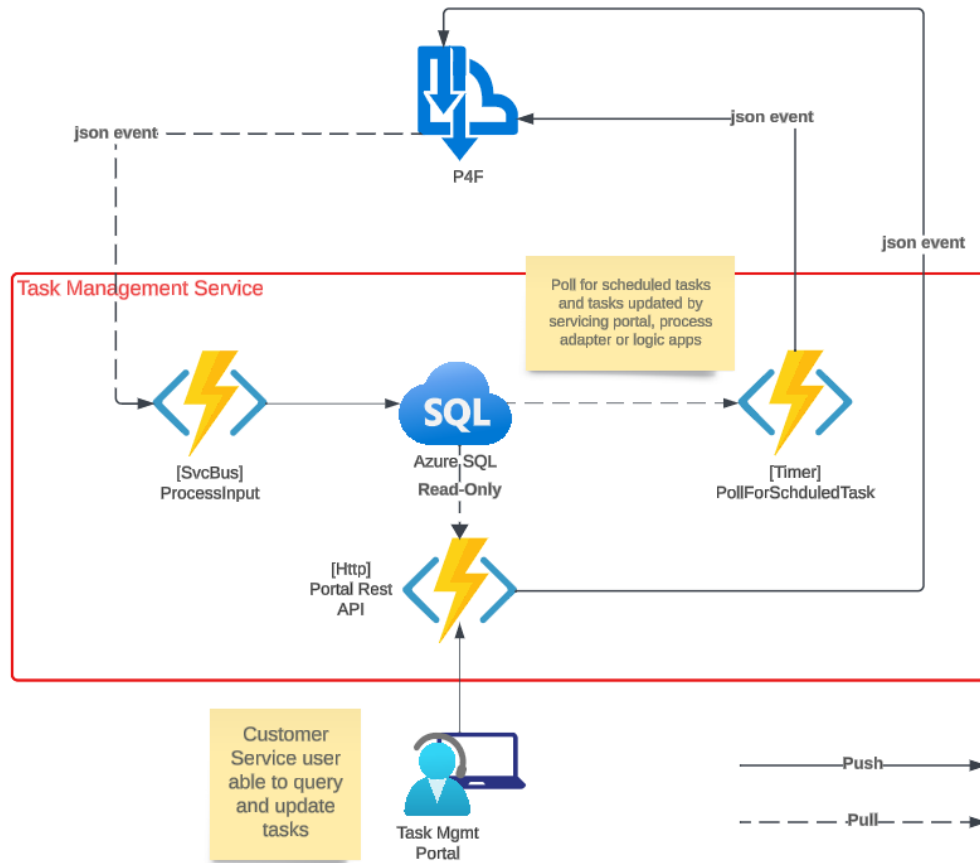
Figure 1 - Physical Design

The Task Management Service will be developed as a stand-alone microservice comprising an Azure Sql database and an Azure Function App with several Azure Functions.

Azure Function – ProcessInput

This is triggered by a message published to the *servicing-queues* service bus topic. It runs the message through a business rules engine, which determines the processing steps based on the message metadata and payload.

Sensitive Data

It is important that no sensitive data such as PII is persisted to the Azure Sql Database. Currently, there are no requirements for the ProcessInput Azure Function to subscribe to messages containing such data. If this were required in future then it would be the responsibility of ProcessInput to remove before saving event data to the database.

Azure Functions – Portal Rest API

A collection of HTTP Triggered functions to enable a customer service portal to query and manage events.

Azure Function – PollForScheduledTask

To enable the Scheduled Task Progression Type. A timer triggered function that polls the database for events with state progression that's dependent on elapsed time or an upcoming date. This would only be used where it's not possible to meet the business requirement by responding to the publication of a service bus message.

JSON Events

Events published to the service bus have the following key value pairs

Key Name	Value Description	Example
eventid	GUID unique to the published message	515fc487-dabe-43f8-88ec-d4614fa48c03
eventtype	Type of business event that triggered the message publication. Event types are listed in this wiki page	CancellationInstructionEvent
correlationid	GUID to correlate between initiating system and integration services e.g. file adapter to payments engine to process adapter to this Task Management service	abb4d962-517c-4d2f-9a99-5c2f5eec8fd7
datetimecreated		2023-11-03T12:03:42.935087Z
eventdata	dataitems object containing business data	"field": "accountexternalid", "value": "PFA000001002"
dataitems	Array of field / value	
eventmetadata	dataitems object containing data about the message	"field": "sourcesystem", "value": "Payments.Engine"

Example Event Message

```
 1  {
 2    "eventid": "515fc487-dabe-43f8-88ec-d4614fa48c03",
 3    "eventtype": "CancellationInstructionEvent",
 4    "correlationid": "abb4d962-517c-4d2f-9a99-5c2f5eec8fd7",
 5    "datetimecreated": "2023-11-03T12:03:42.935087Z",
 6    "eventdata": {
 7      "dataitems": [
 8        {
 9          "field": "accountexternalid",
10          "value": "ACC-0000001095"
11        },
12        {
13          "field": "ddireference",
14          "value": "DDI-0000001162"
15        },
16        {
17          "field": "reason",
18          "value": "AccountClosed"
19        },
20        {
21          "field": "description",
22          "value": "Account Closed."
23        }
24      ]
25    },
26    "eventmetadata": {
27      "dataitems": [
```
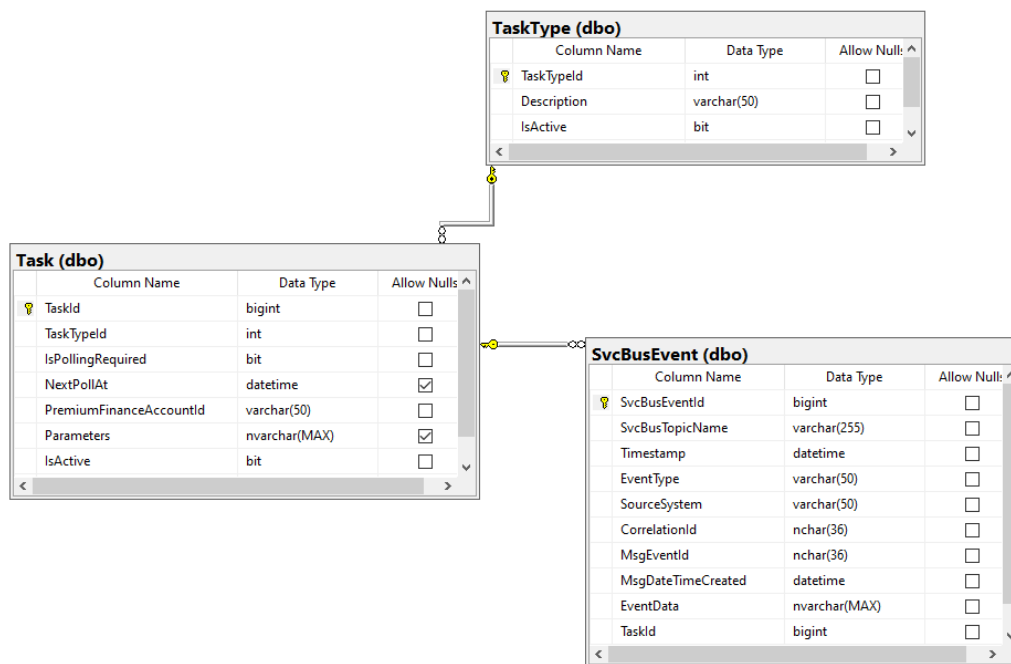
```
28          {
29              "field": "sourcesystem",
30              "value": "Payments.Engine"
31          },
32          {
33              "field": "fieldlist",
34              "value": "accountexternalid|ddireference|reason|description"
35          }
36      ]
37  }
38 }
39
```

Database Design

The Sql Server database design is shown below:

Storing JSON Data

The column *SvcBusEvent.EventData* will be used to store the content of the *eventdata* property of the JSON message that the *ProcessInput* Azure Function consumes from the service bus topic. Azure Sql provides extensive functionality when working with JSON documents, such as:

- use any T-SQL function and query clause to query JSON documents
- ability to index fields within the JSON documents

This functionality is described at: Microsoft documentation

Task Table

Stores tasks to be processed or that have been processed. Note **one** Task may be updated by **many** Service Bus Events, so the Task table has a **one-to-many** relationship to SvcBusEvent table.

Column Name	Description

TaskId	Primary key. Auto incrementing integer (Identity Column)
TaskTypeId	Foreign key to TaskType table, ensure only tasks of expected types are stored
IsPollingRequired	Boolean used to influence if record is to be selected by *PollForChange* Azure Function
NextPollAt	DateTime value used with IsPollingRequired value
PremiumFinanceAccountId	Mastered in CRM
Parameters	Stored as JSON, these are values from EventData of child SvcBusEvent records that may influence processing of the Task e.g. AutoApproved=True

TaskType Table – Static Data Seeded by DevOps

This ensures only expected task types are stored in the Task table. A foreign key relationship exists between the TaskType and Task tables.

Note: an alternative implementation would be to use a check constraint on a TaskType column of the Task table

Column Name	Description
TaskTypeId	Primary key. Auto incrementing integer (Identity Column)
Description	Simple description of the task type

Seed Data Values

TaskTypeId	Description
1	ManualDueDiligence
2	Cancellation
3	UnsignedAgreement
4	CustomerRefund
5	PayerDeceased
6	WaiveFee
7	Forbearance
8	MTAPending

SvcBusEvent

Stores details of JSON event consumed from a service bus topic by *PollForChange* Azure Function

Column Name	Description	Example
SvcBusEventId	Primary key. Auto incrementing integer (Identity Column)	1

SvcBusTopicName	The name of the topic from which the message was consumed. May be useful for troubleshooting	P4F
Timestamp	Datetime record created in YYYY-MM-DDThh:mm:ss UTC	2023-12-13T17:04:34
EventType	Taken from "eventtype" property of JSON event	CancellationInstructionEvent
SourceSystem	Taken from "eventmetadata" property of JSON event	Payments.Engine
CorrelationId	Taken from "correlationid" property of JSON event	abb4d962-517c-4d2f-9a99-5c2f5eec8fd7
MsgEventId	Taken from "eventid" property of JSON event. May be useful for troubleshooting	515fc487-dabe-43f8-88ec-d4614fa48c03
MsgDateTimeCreated	Taken from "datetimecreated" property of JSON event	2023-11-03T12:03:42.935087Z
EventData	JSON object taken from "eventdata" property of JSON event	see example
TaskId	Foreign key to Task table – the task to which this event message relates	1

Processing

Task Progression Types

There are three different ways in which the status of a task may be progressed:

- Business Event e.g. Stripe payment clears arrears to enable MTA
- Scheduled e.g. Attempt process in 3 days
- Manual e.g. Manually complete due diligence

Note: we've not yet identified a user story that definitely requires the "Scheduled" progression type. It's possible there may be some in relation to the fact definitive acknowledgment of BACS collection events is not provided, so we assume collection two working days beyond the request date. The *PollForScheduledTask* Azure Function will only be required if such a user story is defined.

Business Rules Processing

The Task Management service will make use of Microsoft's Rule Engine nuget package to determine processing to be executed depending on the consumed service bus message. This is consistent with our Process Adapter and Payments Engine services.

Details of the Rule Engine package can be found at: https://github.com/microsoft/RulesEngine

On receipt of a service bus message, the ProcessInput Azure Function will query the database to build an object containing all information about a task that may influence business rules of how it should be processed. For example, it may find an MTA Task for which event data has been received with an "AutoComplete" field having a value of "True".

Worked Examples

Business Event – Arrears Cleared

Data Flow
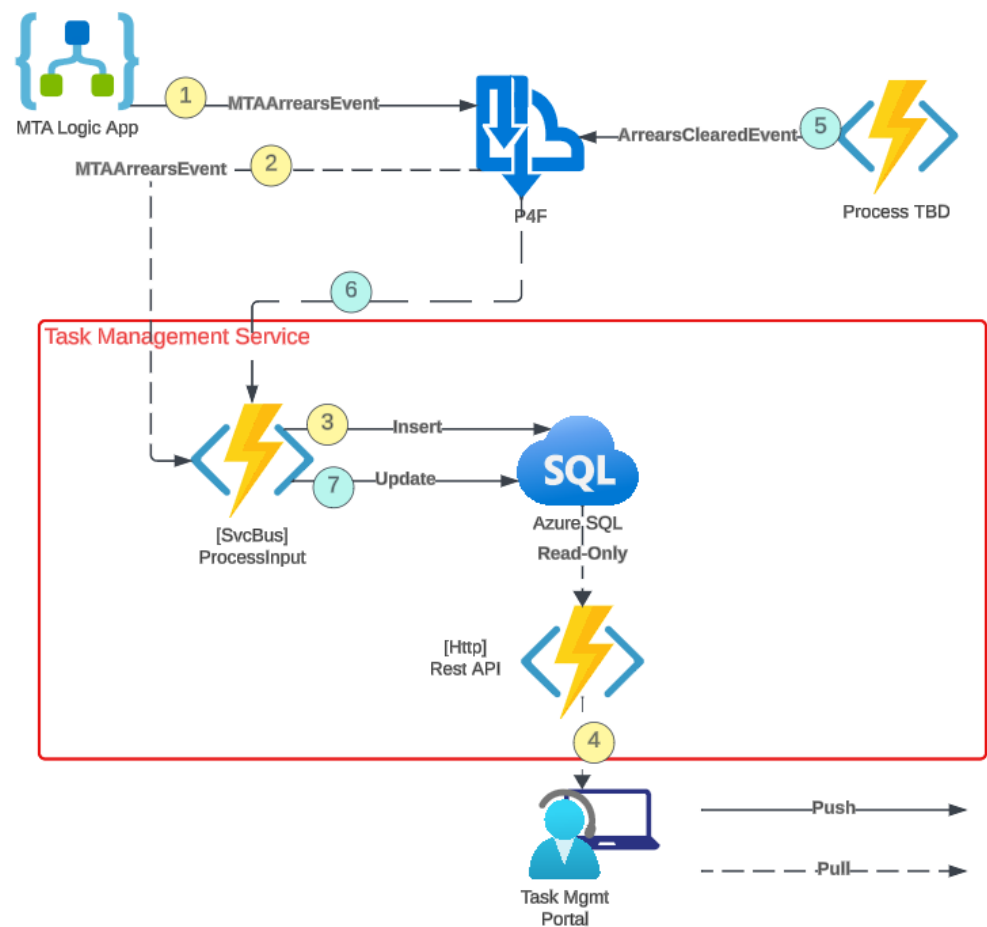
Flow Number	Description
1	During MTA processing, the account is found to be in arrears and so an event is published to the P4F service bus topic. Sample MTAArrearsEvent
2	The ProcessInput function subscribes to the MTAArrearsEvent.
3	It executes the Task Management Rules Policy and matches MTAArrears-Rule01. It uses the output from the rule to INSERT into the Task table. It inserts details of the event into the SvcBusEvent table, along with the TaskId of the previously inserted Task.
4	At this point, the customer services team are able to view the active task through the portal
5	A process runs periodically to determine accounts that have previously moved out of arrears. How it manages this is to be determined, possibly from thought machine events directly or querying the ODS

The ProcessInput function subscribes to the StripeChargeEvent.

It executes the Task Management Rules Policy and matches StripeCharge-Rule01.

It uses the output from the rule to UPDATE a single record in the Task table.

It inserts details of the event into the SvcBusEvent table, along with the TaskId of the previously updated Task.

⑦	

Service Bus Subscription

SourceSystem= 'LA.MTM' and EventType = 'MTAArrearsEvent'

Business Rules Processing

Policy Workflow Name: Task Management Policy

Rule Name: MTAArrears-Rule01

Rule Matching		
Field	**Value**	**Condition**
eventtype	MTAArrearsEvent	AND
sourcesystem	LA.MTA	
Rule Output Object		
Property	Value	Notes
TaskType	MTA Arrears	
IsPollingRequired	False	
NextPollAt	Null	
PremiumFinanceAccountId	event.accountexternalid	
IsActive	True	

Rule Name: StripeCharge-Rule01

Rule Matching		
Field	**Value**	**Condition**
eventtype	StripeChargeEvent	AND
sourcesystem	Adapter.Stripe	
Rule Output Object		
Property	Value	Notes
TaskType	MTAArrears	
IsPollingRequired	False	
NextPollAt	Null	
PremiumFinanceAccountId	event.accountexternalid	
IsActive	False	

Event Samples

MTAArrearsEvent

note: this event is not currently in use

```
1  {
2    "eventid": "515fc487-dabe-43f8-88ec-d4614fa48c03",
3    "eventtype": "MTAArrearsEvent",
4    "correlationid": "abb4d962-517c-4d2f-9a99-5c2f5eec8fd7",
5    "datetimecreated": "2023-11-03T12:03:42.935087Z",
6    "eventdata": {
7      "dataitems": [
8        {
9          "field": "accountexternalid",
10         "value": "PFA000001001"
11       },
12       {
13         "field": "arrearsammount",
14         "value": "147.00"
15       }
16     ]
17   },
18   "eventmetadata": {
19     "dataitems": [
20       {
21         "field": "sourcesystem",
22         "value": "LA.MTA"
23       },
24       {
25         "field": "fieldlist",
26         "value": "accountexternalid|arrearsammount"
27       }
28     ]
29   }
30 }
31
```

ArrearsClearedEvent

Note: this event is specified [here](#)

```
1  {
2    "eventid": "515fc487-dabe-43f8-88ec-d4614fa48c03",
3    "eventtype": "ArrearsClearedEvent",
4    "correlationid": "abb4d962-517c-4d2f-9a99-5c2f5eec8fd7",
5    "datetimecreated": "2023-11-03T12:03:42.935087Z",
6    "eventdata": {
7      "dataitems": [
8        {
9          "field": "accountexternalid",
10         "value": "PFA000001001"
11       }
12     ]
13   },
14   "eventmetadata": {
15     "dataitems": [
16       {
17         "field": "sourcesystem",
```
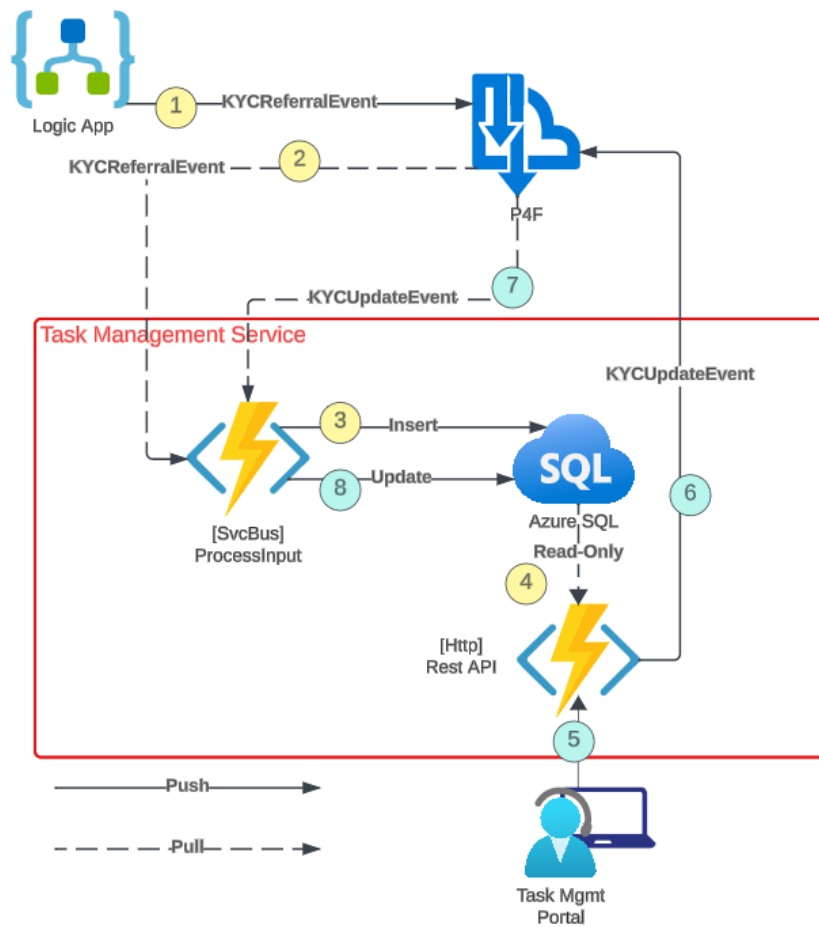
```
18          "value": "Default.Cycle" #TBD
19        },
20        {
21          "field": "fieldlist",
22          "value": "accountexternalid"
23        }
24      ]
25    }
26 }
27
28
```

Business Event - Manual Due Diligence

Data Flow

Flow Number	Description
1	During new-business processing, the logic app publishes a KYCReferralEvent to the P4F service bus topic.
2	The ProcessInput function subscribes to the KYCReferralEvent.
3	It executes the Task Management Rules Policy and matches <policy name>.

It uses the output from the rule to INSERT into the Task table.

It inserts details of the event into the SvcBusEvent table, along with the TaskId of the previously inserted Task.

At this point, the customer services team are able to view the active task through the task management portal

The task management portal user runs a manual process to determine KYC outcome. They request to update the task via the portal. The portal calls the Task Management Service's Rest API Azure Function

The Azure Function publishes a KYCUpdateEvent to the P4F service bus topic.

It executes the Task Management Rules Policy and matches <rule goes here>.

It uses the output from the rule to UPDATE a single record in the Task table.

It inserts details of the event into the SvcBusEvent table, along with the TaskId of the previously updated Task

Service Bus Subscription

SourceSystem= 'LA.KYC' and EventType = 'KYCReferralEvent'

Business Rules Processing

Policy Workflow Name: Task Management Policy

Rule Name: KYCReferral-Rule01

Rule Matching		
Field	**Value**	**Condition**
eventtype	KYCReferralEvent	AND
sourcesystem	LA.KYC	
Rule Output Object		
Property	Value	Notes
TaskType	Manual Due Diligence	
IsPollingRequired	False	
NextPollAt	Null	
PremiumFinanceAccountId	event.accountexternalid	
IsActive	True	

Rule Matching		
Field	**Value**	**Condition**
eventtype	KYCUpdateEvent	AND
sourcesystem	TaskManagement.Portal	
Rule Output Object		
Property	Value	Notes
TaskType	Manual Due Diligence	Reviewer in EventData
IsPollingRequired	False	
NextPollAt	Null	
PremiumFinanceAccountId	event.accountexternalid	
IsActive	False	

Event Samples

KYCReferralEvent

```
1   {
2     "eventid": "515fc487-dabe-43f8-88ec-d4614fa48c03",
3     "eventtype": "KYCReferralEvent",
4     "correlationid": "abb4d962-517c-4d2f-9a99-5c2f5eec8fd7",
5     "datetimecreated": "2023-11-03T12:03:42.935087Z",
6     "eventdata": {
7       "dataitems": [
8         {
9           "field": "accountexternalid",
10          "value": "PFA000001001"
11        },
12        {
13          "field": "customerid",
14          "value": "de6212dd-3890-ee11-be36-6045bd0f5176"
15        },
16        {
17          "field": "autoapproval",
18          "value": "true"
19        }
20      ]
21    },
22    "eventmetadata": {
23      "dataitems": [
24        {
25          "field": "sourcesystem",
26          "value": "LA.KYC"
27        },
28        {
29          "field": "fieldlist",
30          "value": "accountexternalid|customerid|autoapproval"
31        }
32      ]
33    }
34  }
```

KYCUpdateEvent

```
1   {
2     "eventid": "515fc487-dabe-43f8-88ec-d4614fa48c03",
3     "eventtype": "KYCUpdateEvent",
4     "correlationid": "abb4d962-517c-4d2f-9a99-5c2f5eec8fd7",
5     "datetimecreated": "2023-11-03T12:03:42.935087Z",
6     "eventdata": {
7       "dataitems": [
8         {
9           "field": "accountexternalid",
10          "value": "PFA000001001"
11        },
12        {
13          "field": "customerid",
14          "value": "de6212dd-3890-ee11-be36-6045bd0f5176"
15        },
16        {
17          "field": "status",
18          "value": "passed"
```

```
19        },
20        {
21          "field": "reviewer",
22          "value": "rob.bowman"premfina.com"
23        }
24      ]
25    },
26    "eventmetadata": {
27      "dataitems": [
28        {
29          "field": "sourcesystem",
30          "value": "TaskManagement.Portal"
31        },
32        {
33          "field": "fieldlist",
34          "value": "accountexternalid|customerid|status|reviewer"
35        }
36      ]
37    }
38  }
```