

## **Building Block Definition**

### **Payments**

Developed by: Vijay Mauree - ITU, Arnold Kibuuka - ITU, Francesco Pasti, James Dailey - MIFOS, Oscar Correia - Maarifa Education, P.S Ramkumar, Khuram Farooq - World Bank. In cooperation with GIZ, ITU, DIAL, and the Government of Estonia.

# Table of Contents

<b>1 Version History</b>	<b>5</b>
<b>2 Description</b>	<b>5</b>
2.1 Financial Inclusion, Banks and Mobile Money Accounts	6
2.2 Payments Building Block Capabilities	7
2.3 Mapping Based on Level of Maturity	7
2.4 Infrastructure Requirements	8
2.5 Out of Scope Assumptions	9
<b>3 Terminology</b>	<b>10</b>
<b>4 Payment Infrastructure Deployment Scenarios</b>	<b>12</b>
4.1 Regional Switch vs Country Switch	12
4.2 Payment infrastructure scenarios	12
4.2.1 Payment Infrastructure Scenario 1	14
4.2.2 Payment Infrastructure Scenario 2-A	14
4.2.3 Payment Infrastructure Scenario 2-B	15
4.2.4 Payment infrastructure scenario 3	16
4.2.5 Payment Infrastructure Scenario 4	16
4.2.6 Payment Infrastructure Scenario 5	17
4.2.7 Payment Infrastructure Scenario 6	18
4.3 Prerequisites	18
<b>5 Key Digital Functionalities</b>	<b>19</b>
5.1 Government to Person Payment Principles	20
5.1.1 Beneficiary/Recipient-related Principles	20
5.1.2 Infrastructure and Systems-related Principles	21
5.1.3 Payment Services Provider- related Principles	22
5.2 General Key Functionalities of the Payments Building Block	22
5.3 Key Functionalities for Bulk Payment	24
5.4 Key Functionalities for Voucher Disbursement	24
<b>6 Cross-Cutting Requirements</b>	<b>25</b>
<b>7 Functional Requirements</b>	<b>25</b>
7.1 Payments Building Block Components	25
7.1.1 API Management Gateway	26
7.1.2 Payment Orchestration	27
7.1.3 Voucher Management System	27
7.1.3.1 Voucher Provisioning	27
7.1.3.2 Voucher Issuance	27

7.1.3.3 Voucher Redemption	28
7.1.4 VMS API interface	28
7.1.5 Voucher Storage	28
7.1.6 Account Lookup Directory Service (Mapper)	29
7.1.7 Payment Request Initiation	29
7.1.8 Payment Gateway	30
7.1.9 Payment Portal	30
7.1.10 Notifications Service	30
7.1.11 Reconciliation	30
7.1.12 Validation and Verification	31
7.1.13 Batch Logic and Queuing	31
7.1.14 Workflow and Scheduling	31
7.1.15 Event Log	32
7.1.16 Audit Logging	32
7.1.17 Reporting	33
7.1.18 Security layer	33
7.1.19 Data Protection	34
<b>7.2 Payments Building Block Technical Requirements</b>	<b>34</b>
<b>8 Data Structures</b>	<b>37</b>
<b>8.1 Voucher Resource Model</b>	<b>37</b>
8.1.1 Minimum Required Data	37
8.1.2 Voucher Groups	38
<b>8.2 Incoming Government Payments Resource</b>	<b>38</b>
<b>8.3 Data Elements</b>	<b>39</b>
8.3.1 API Name: Voucher APIs	39
8.3.2 API Name: Bulk Payment	40
8.3.3 API Name: Incoming Government Payment	40
<b>8.4 Account Identifiers</b>	<b>42</b>
<b>9 Service APIs</b>	<b>43</b>
<b>9.1 Incoming Payments to Government (P2G)</b>	<b>43</b>
9.1.1 Payee-Initiated Merchant Payment	44
9.1.2 Payer-Initiated Merchant Payment	44
<b>9.2 Bulk Payment APIs (Outgoing)</b>	<b>44</b>
<b>9.3 From Source Beneficiary System to Payments Building Block</b>	<b>45</b>
9.3.1 Programs	45
9.3.2 Beneficiaries	45
9.3.3 Disbursement	46
<b>9.4 From Payments Building Block to Lookup Directories (or Similar)</b>	<b>47</b>
<b>9.5 From Payments Building Block: Bulk Payment to FSPs</b>	<b>47</b>
<b>9.6 Voucher APIs (Outgoing)</b>	<b>47</b>

9.6.1 VoucherPreActivation API	48
9.6.2 VoucherActivation API	49
9.6.3 BatchVoucherActivation API	49
9.6.4 VoucherRedemption	50
9.6.5 VoucherStatus API	51
9.6.6 VoucherCancellation API	52
9.6.6.1 Response Status Codes	53
<b>10 Workflows</b>	<b>54</b>
<b>10.2 G2P Bulk Payment Workflow</b>	<b>54</b>
10.2.1 Bulk Disbursement for Unconditional Cash Transfer	54
10.2.1.1 Prerequisites	54
10.2.1.2 Description	54
10.2.1.3 Interaction with Other Building Blocks	56
10.2.1.4 Sequence Diagram	57
10.2.2 Disbursement to Beneficiary Using Mobile Money	59
10.2.2.1 Interaction with Other Building Blocks	59
<b>10.3 G2P Beneficiary Payments Using Vouchers</b>	<b>59</b>
10.3.1 Description	59
10.3.1.1 Admin Process	60
10.3.1.2 Interaction with Other Building Blocks	61
10.3.1.3 Sequence Diagram	61
Voucher Activation	61
Voucher Redemption	63
Voucher Cancellation	65
Voucher Technical Requirements	67
<b>10.4 P2G Payments</b>	<b>67</b>
10.4.1 Description	68
10.4.2 Interaction with Other Building Blocks	68
10.4.3 Sequence Diagram - P2G Payment	68
10.4.4 Sequence Diagram - P2G FSP Payment by USSD Prompt	71
10.4.5 Sequence Diagram - P2G Payment with QR Code	72
10.4.5.1 QR Code Payment Flow Use Case Example	73
<b>11 Other Resources</b>	<b>74</b>
<b>11.1 Standards</b>	<b>74</b>
<b>11.2 GovStack Resources</b>	<b>74</b>
<b>11.3 Unconditional Social Cash Transfer Resources</b>	<b>74</b>
<b>12 Key Decision Log</b>	<b>75</b>
<b>13 Future Considerations</b>	<b>79</b>

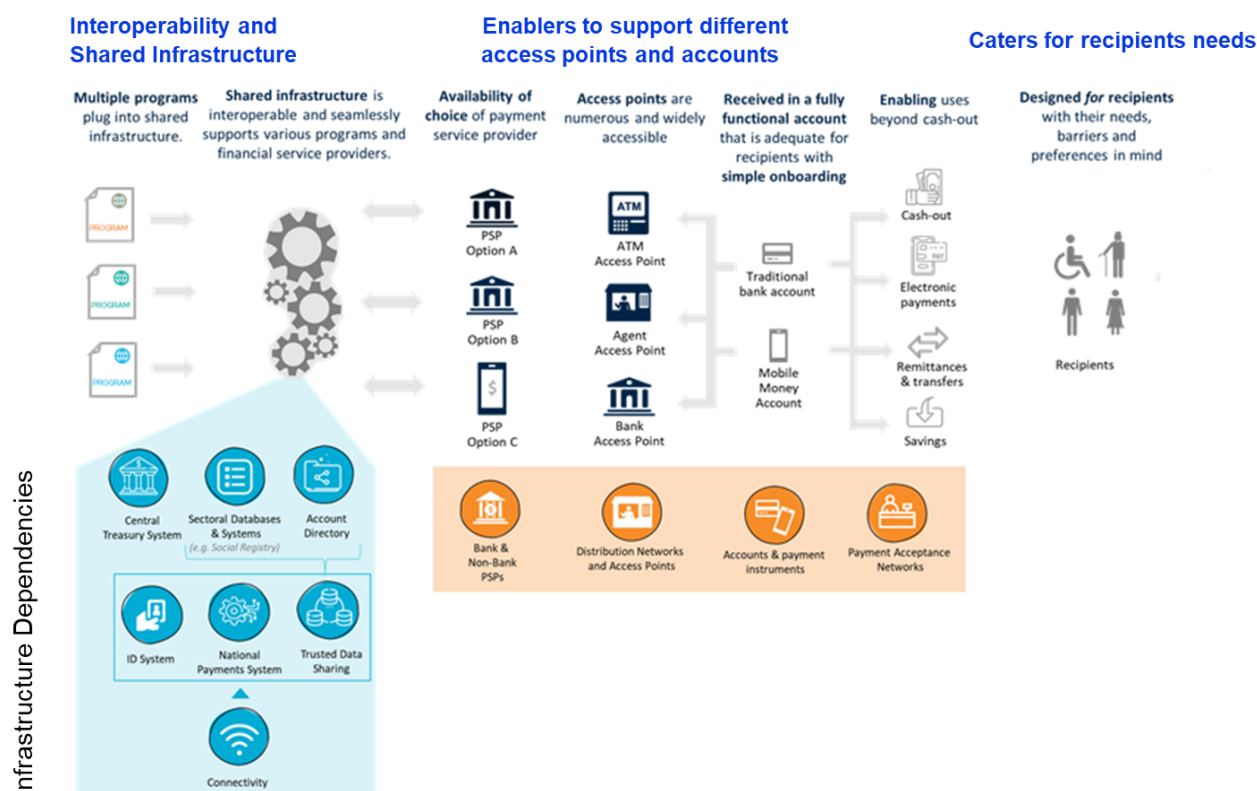
# 1 Version History

Version	Author	Comment
1.0.0	1. dq - World Bank	Initial version
1.1.0	Contributors 1. Sasa Kovacevic 2. Aare Lapõnin 3. Anita Mittal 4. Sanjay Jain - MOSIP 5. Luukas Ilves - Guardtime 6. Lesley-Ann Vaughan - MojaLoop	Applied feedback from technical review.

## 2 Description

The Payments Building Block enables digital financial payments to be tracked, evaluated, initiated, validated, processed, logged, compared and verified against budget. This ICT Building Blocks also provides interoperability with connections to the various external applications that need payment services in order to trigger transitions in their own WorkFlow. Payment services generally interface through gateways to regulated financial entities such as banks, credit facilities and insurance companies. To help users easily complete payment transactions and learn if their transaction succeeded or failed, it converts heterogeneous interface protocols, formats and user interfaces to a standard set of common interfaces and formats. It can also help in tracking costs of commodity and equipment purchases to optimize budgets.

The Payments Building Block consists of components that enable multiple use cases in a generic manner. The use cases, including Government to Person(G2P), Person to Government (P2G), Government to business (G2B) and Business to Government (B2G), The existing commercial Payments schemes are heterogenous across regions and countries, the Payments BB assumes that some components are optional when considering implementation. The payments BB covers components that can be used to deliver the key functionalities and the connections to existing systems in the market, but does not contemplate building a new payments scheme



## 2.1 Financial Inclusion, Banks and Mobile Money Accounts

Financial inclusion strategies have largely been based on growing access to regulated accounts. This was traditionally understood to be bank accounts, in recent years mobile money (MoMo) accounts have greatly increased transactional account access. These MoMo accounts are often, but not always, treated as secondary to the banking system and lack interoperability.

That said, today, with over 1.2 billion global accounts, transacting an average of over \$2 billion per day, mobile money has grown to become the leading payment platform in a growing number of emerging markets. Today, there are more mobile money accounts in Sub-Saharan Africa than bank accounts. Indeed, during the pandemic, demand for mobile money services increased among businesses, governments and new services that previously relied on cash or other payment channels.

There are currently active discussions in many markets which need integration of mobile money into existing national infrastructure. While the integration of mobile money into existing schemes and infrastructure has the potential to create value for all stakeholders in the ecosystem, many legacy payments systems may not be suitable for mobile money integration in their current state. As an example, many existing interbank switches do not support mobile money as a core feature because of legacy technical architecture. What is often missing is the essential capability of the switch to handle high-volume, low-value flows with the mobile number (Mobile Station Integrated Services Digital Network) as identifier for routing and address resolution hence the need for integration.

Recent industry developments offer solutions designed to solve many of the issues around the switching of mobile money.

Mobile money providers, regulators, infrastructure providers, and other financial system players should seek to engage on any potential interoperability initiatives from an early stage to ensure optimal payments system design for all parties. Should existing infrastructure be used, a thorough assessment should be conducted, with input from all parties, to ensure appropriate governance, commercial, and technical standards are incorporated.

Mandating interoperability within a short time frame, removes opportunities for a thoughtful and holistic analysis of commercial sustainability. Requiring immediate interoperability can also delay or hamper the uptake of interoperable transactions, and mobile money services altogether. This is a particular risk in contexts where regulators or governments impose immediate connections to a centralized hub. Mobile money providers should be encouraged to connect when their service is ready. However alternative modalities of payment such as cash-vouchers can also be considered in scenarios of low-resource regions that lack modern infrastructure and volume of mobile phone users.

Also, in countries where mobile money providers are not connected to a payment switch, aggregators can enable the flow of funds between multiple financial service providers including banks and non-banks. While aggregators may not be as efficient and scalable as switches, they can help solve interoperability issues in the short and medium term.

## 2.2 Payments Building Block Capabilities

Generally the Building Block is supportive of multiple types of payment use cases where the Government is one of the parties:

- **Government to people (G2P)**
  - Payment where the payer is a government and the payee is an individual e.g. Social benefits disbursement, government salary payments, Conditional and Unconditional Social Cash Transfers).
- **People to Government (P2G).**
  - Payment where the payer is an individual and the payee is a government (e.g. payment of taxes, payment Registration for Postpartum and Infant Care, payment of school fees, driving license, utility payments).
- **Government to Business (G2B)**
  - G2B payments include tax refunds, goods and services purchases and subsidies. Contracts payments, benefits, loans
- **Business to Government (B2G)**
  - B2G payments include paying taxes and fees.
- **Government to Government (G2G)**
  - Payments between two government entities (ie between Ministries/Departments and public sector institutions).

Government to Business payments (G2B) and Government to Government (G2G) are not covered in this specification and will be covered in the second phase..

## 2.3 Mapping Based on Level of Maturity

The DIAL use cases can be mapped to the payments building block capabilities as follows.

Capabilities	Payment Infrastructure Scenario	DIAL User journey	Use Cases	Destination Channel
G2P	Scenario 1-4 & 6	<u>UC-P-USCT-003: Payment - Unconditional Social Cash Transfer (direct payment based on family relationship)</u>  <u>Payment - Unconditional Social Cash Transfer (bank payments)</u>  Payment - Unconditional Social Cash Transfer (non-electronic/ cash payments)	a). Beneficiary payment, Incentive payment to mother.  b) Government Salary payment  c) Unconditional cash transfer	Voucher payment, Mobile Money, direct bank transfer
P2G	Scenario 1-4 & 6	Postpartum care	Payment - registration of birth (Postpartum and Infant Care)	Mobile Money.

## 2.4 Infrastructure Requirements

Modality	Infrastructure Components	Comments
Mobile Money	<ul style="list-style-type: none"> <li>Account lookup - directory service</li> <li>Payment portal</li> <li>Payment gateway (only in selected scenarios)</li> <li>Payment request initiation</li> <li>Notification service</li> </ul>	<p>Mobile money is a destination account to distribute funds to end users (in the case of G2P) or an initiation account to enable users to initiate a payment for government services (P2G) which is facilitated by private commercial entities, typically Mobile Network Operators.</p> <p>There are three ways that facilitate interoperability between a mobile money provider and a financial service institution (which could be the initiator of a payment - G2P or the receive - P2G): Through a Switch, through a third-party aggregator or through a bilateral integration. We expect one or multiple</p>



		of these components to be in place to facilitate the use cases discussed in this document but these will remain out of the payment building block.
Vouchers	Voucher Management System, Merchant Ecosystem, Merchant Registry	The Merchant Ecosystem and the Merchant Registry are both outside the scope of the Payments Building Block. The development of the Merchant Ecosystem will be driven by the relevant government institution. The Merchant Registry will be managed by another building block.

## 2.5 Out of Scope Assumptions

- Payment scheme creation is outside of the scope.
- Government to Business and Government to Government Payments are out of scope for the first phase and will be considered in the second phase.
- Identity systems are separate and outside the payments BB, with key implications for KYC/CDD in the banking system.
- Delegation of authority - Consent of the recipient for payment of G2P to be made to a third party (next of kin) is outside the scope of the payments building block and should be handled by consent management building block at the time of beneficiary registration for the G2P program.
- Consent of people eligible as beneficiaries of G2P programmes for their personal details (ie National ID and payments details) to be stored in tokenised form in the centralised mapper (where the government has to implement the mapper).
- For social benefits G2P payments, social registries are an important building block which will implement the registration, and determination of potential eligibility of citizens for one or more social programs. As such, they are a separate building block and are outside the payments BB.
- An Integrated Financial Management Information System ( IFMIS ) and a Treasury Single Account (TSA) are essential components in improving the safety and efficiency of government payment programs. The TSA, in particular, ensures effective aggregate control over government cash balances and facilitates the reconciliation between banking and account data. It is assumed that these components exist at the level of the government as they are outside the scope of the payments BB.
- Settlement systems are outside of the scope. Settlement allows the flow of money between participants and can be done on a Pre-funding basis which allows incoming transactions if the sending DFSP has already deposited sufficient liquidity with them. Alternatively settlement can be on a Clearing-base where FSPs allow incoming transactions before receiving the funds.

- Pricing. This generally revolves around processing fees (for each transaction a fixed fee is paid to the entity processing the transaction) and interchange (where one participant agrees to pay the other).
- Dispute resolution mechanisms which allow FSPs to reach consensus on a transaction's status and financial liabilities in the case of a dispute. There are two main types of dispute resolution mechanisms: the consensus option where parties must agree on a transaction's status; and the arbitration option where one party has authority over a transaction's status.
- Governance defining sets of rules on how participants make decisions.
- Development of the voucher management ecosystem that is outside the technical specification of the Payment Building Block, including but not restricted to recruitment and registration of merchants / agents for the redemption of vouchers, etc.

## 3 Terminology

Several elements of the glossary are based on the terminology defined by the [levelone project](#) and [ITU DFS Glossary](#)

### **Authentication**

The validation of user credentials for the purpose of system login and basic access.

### **Authorization**

The permission given by the Payer or entity to make a Payment.

### **Beneficiary**

non referring to a payee of a program.

### **Bulk Payments**

A Payment from a single Payer to multiple Payees, for example cash transfer programs from a government or NGO to a set of beneficiaries.

### **End User**

The customer of a digital financial services provider. the customer may be a consumer, a merchant, a government, or another form of enterprise.

### **Financial Management Information System:**

FMIS is the budget management system of the government treasury, which ensures that government (Ministries, Department, Agencies)) payments are processed within the approval budget and releases.

### **Financial Services Providers (FSPs)**

A financial services provider refers to one that is licensed by a regulatory authority to provide transaction accounts which hold customer funds and are used to make and receive payments. DFSPs have relationships with consumers, merchants, and other enterprises, and provide digital financial services to End Users.

### **Interoperability**

The ability of DFSPs participating in a payment Scheme to exchange Transactions with each other. The term may also be used when two systems interconnect.

**Mapper or Account Lookup Directory Service**

A dynamic directory matching beneficiaries' unique identifier and their account numbers enable multiple programs to direct payments to the same account and beneficiaries to switch payment service providers. The identifier can be a national ID, phone number, or other number or alias that can uniquely identify individuals across social protection and financial sector databases.

**Mobile Money (MoMo)**

A mobile money service includes transferring money and making and receiving payments using the mobile phone. The service must be available to the unbanked, e.g. people who do not have access to a formal account at a financial institution. The service must offer a network of physical transactional points which can include agents, outside of bank branches and ATMs, that make the service widely accessible. Mobile banking or payment services (such as Apple Pay and Google Wallet) that offer the mobile phone as just another channel to access a traditional banking product are not included.

**Mobile Money Services Provider**

A category of DFSPs that use mobile phones as the access method to provide transaction Accounts to End Users.

**Payee**

The recipient of funds in a payment transaction.

**Payer**

The person/organization whose account is debited in a payment transaction

**Payment**

An exchange of funds, credentials, and other necessary information to complete an obligation between End Users. A transfer is an example of a payment.

**Payment Alias/Payment address**

Alias may include phone numbers, email addresses, or other handles. They may include globally unique URLs, and may be chosen by the user. These are increasingly seen in new payment systems. Within systems, the payment alias points to an underlying payment account address, specifically and uniquely located within a payment scheme. A payment address may be different from the underlying account address, depending on the payment scheme rules but is often the same thing.

**Quick-Response (QR) Code**

A method of encoding and visualization of data, which are machine-readable. There are multiple QR models.

**Routing**

The process by which a route to a payment address is determined. The mechanism of sending payment information across different systems. This is sometimes used interchangeably with addressing.

**Transaction**

A reference to the entirety of the exchange, including a Payment but may also include information between the Payer's DFSP and the Payee's DFSP.

**Treasury Single Account**

A TSA is a unified structure of government bank accounts that gives a consolidated view of government cash resources. Based on the principle of unity of cash and the unity of treasury, a TSA is a bank account or a set of linked accounts through which the government transacts all its receipts and payments. The principle of unity follows from the fungibility of all cash irrespective of its end use. While

it is necessary to distinguish individual cash transactions for control and reporting purposes, this purpose is achieved through the accounting system and not by holding/depositing cash in transaction specific bank accounts. This enables the treasury to delink management of cash from control at a transaction level<sup>1</sup>.

#### **Voucher**

A token that entitles the holder to a discount or that may be exchanged for goods or services

#### **Voucher Group**

A voucher group is a characteristic of a voucher that restricts the function or use case that a voucher can be used for. It can also be used a dimension for reporting on

## **4 Payment Infrastructure Deployment Scenarios**

### **4.1 Regional Switch vs Country Switch**

Domestic payment switches have specific economic characteristics and may not reach adequate economies of scale. Whilst costs can be passed onto customers, without sufficiently low charges, volumes may remain perennially low. Domestic switches facing this scenario thus remain under-utilized, costly, and under-capitalized. In recent years, the concept of Regional payment switches has gained some traction with new efforts realized or underway across Southern African Development Community (SADC), West African Monetary and Economic Union (UEMOA), East African Community (EAC), and Economic and Monetary Community of Central Africa (CEMAC).

Regional switches (which may serve as a common resource for multiple countries) often rely on pre-existing arrangements between Central Banks, regional Commercial Banks, or a combination thereof. When multiple currencies are in play, a reference currency or a basket of currencies must be chosen for regional settlement accounts to be viable. Settlement arrangements are vital to the viability of a regional switch. A Regional switch may also integrate financial transactions from countries that already have domestic payment switches (like a switch of switches).

Regional switches may take on any and all use cases, or may be restricted to specific use cases, for example Large Value Transfer or B2B trade. Because regional trade drives economic activity, regional switches often go hand in hand with portability of business and consumer credentials (i.e. relating to Customer Due Diligence), trade policy, and movement of labor.

### **4.2 Payment infrastructure scenarios**

The Payments building block (BB) sits between the government account systems (i.e. at the Ministry of Finance or Central Bank) and the public or private switching available in the market. As such the PaymentsBB is assumed in each, but not shown. The primary focus in describing these scenarios is on the relationship between payment schemes, banking institutions, and government institutions. This section describes *what is available* in-country.

<sup>1</sup> Source: [Treasury Single Account: Concept, Design, and Implementation Issues](#); by Sailendra Pattanayak and Israel Fainboim; IMF Working Paper

Infrastructure Scenarios	Suggested Approach
<b>Payment Infrastructure Scenario 1</b> Regional Switch (or a switch of switches) connects financial service providers including banks and non-banks in different countries in a region.	Government should leverage the regional payments switch (scheme) to enable all use-cases. This may also imply important developments of a standardized Customer Due Diligence (CDD). Note that while a State owned bank is shown in the diagram below, the option exists for the Government to designate a specific Commercial bank, or to have the funds flow directly from the Ministry of Finance (Treasury) assuming such Treasury payment functionality can be supported by the Regional Switch.
<b>Payment Infrastructure Scenario 2</b> A payment switch connecting financial service providers such as banks to non-banks including mobile money providers is in place and actively working.	Government should leverage the existing switch to enable payments use-cases to reach mobile money providers. There may be a State Owned Bank (2-A) or not (2-B).
<b>Payment Infrastructure Scenario 3</b> A payment switch is in place in the country or is in the process to be deployed but non-banks, including mobile providers are not connected to it.	Government should leverage third-party aggregators or bilateral integrations to enable payment use-cases to reach mobile money providers.
<b>Payment Infrastructure Scenario 4</b> A payment switch is not place or in the process of being deployed	Government should leverage third-party aggregators or bilateral connections to enable payment use-cases to reach mobile money providers
<b>Payment Infrastructure Scenario 5</b> Government makes G2P payments using Central Bank Digital Currency (CBDC) .	The Central Bank distributes the funds for the CBDC accounts via regulated entities (e.g. Payment Service Providers).
<b>Payment Infrastructure Scenario 6</b> Citizens have CBDC accounts directly with the Central Bank or other accounts with other payment service providers	The Central Bank sets up limited accounts for each person, providing a base level of access and functionality. Such accounts may be denominated in country currency (fiat) or CBDCs. This scenario places more responsibility on government structures and is not often considered (outside of Postal Banks and the like).

Under each of these scenarios, there are different setups possible that relate the key roles played by the Programmatic Ministries (e.g. Ministry of Education or Pension Program), the Ministry of Finance, the Central Bank and the existing Payment Switch.

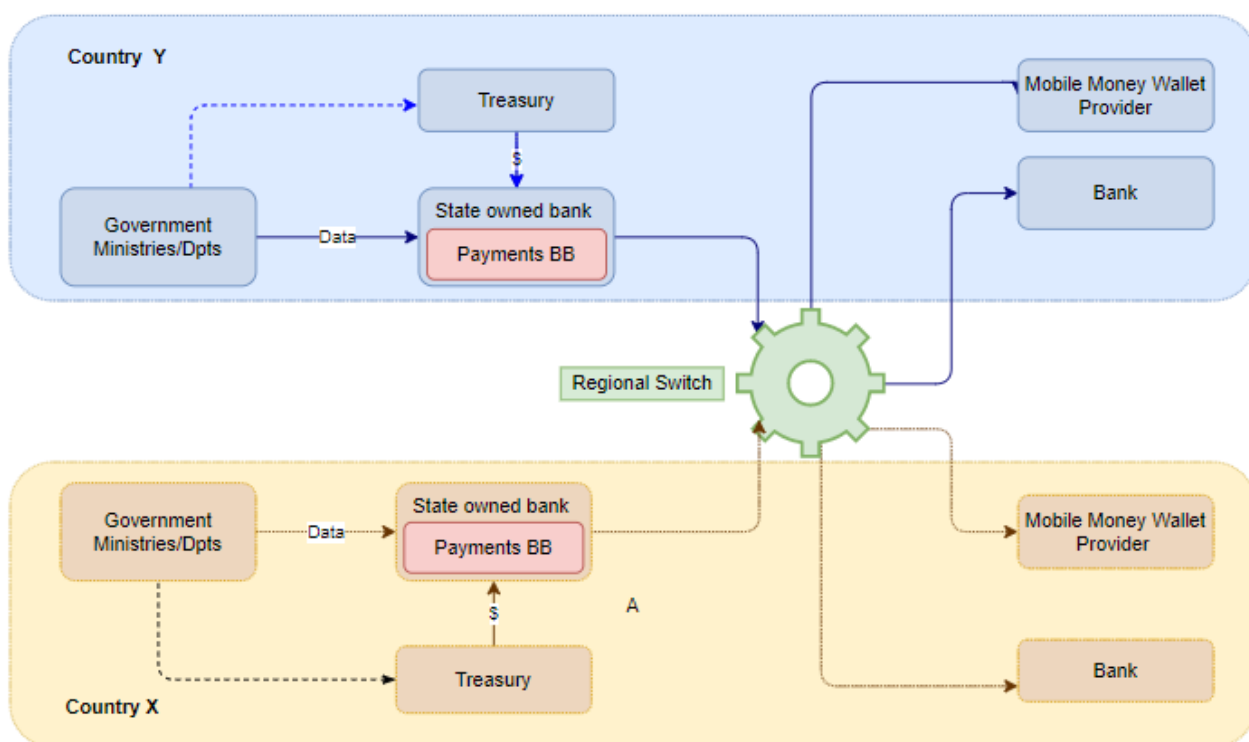
The flow of funds and the flow of data are separate but equally important and they have a number of interactions.

### 4.2.1 Payment Infrastructure Scenario 1

Regional Switch (or a switch of switches) connects financial service providers including banks and non-banks.

#### Scenario 1

Regional Switch (or a switch of switches) connects financial service providers including banks and non-banks.



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

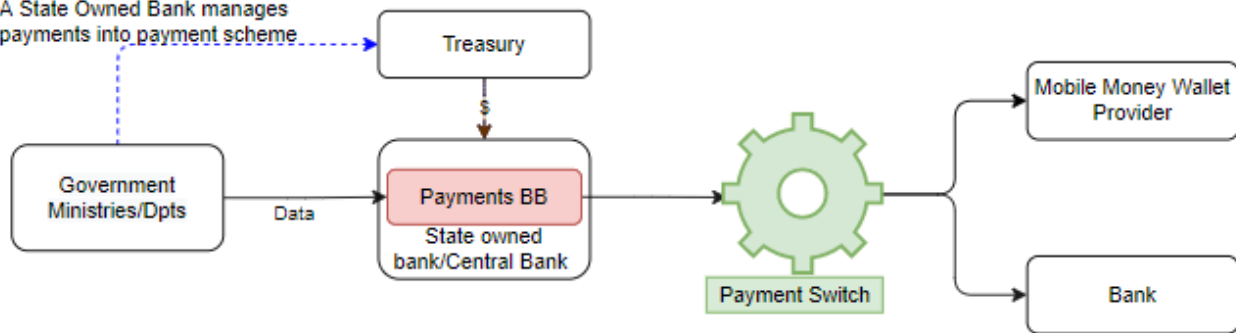
### 4.2.2 Payment Infrastructure Scenario 2-A

Scenario 2-A and Scenario 2-B cover the concept of separating flows of data from accounting flows.

The Central Bank receives the funds transfer advice from the Treasury and the data from the Ministry of Health and does the steps of breaking it into bulk, based on rules of capacity of source and destination systems. Payments are then routed "on-us" or to the external financial providers "off-us" via a payment switch or similar mechanism.

## Scenario 2A

A State Owned Bank manages payments into payment scheme



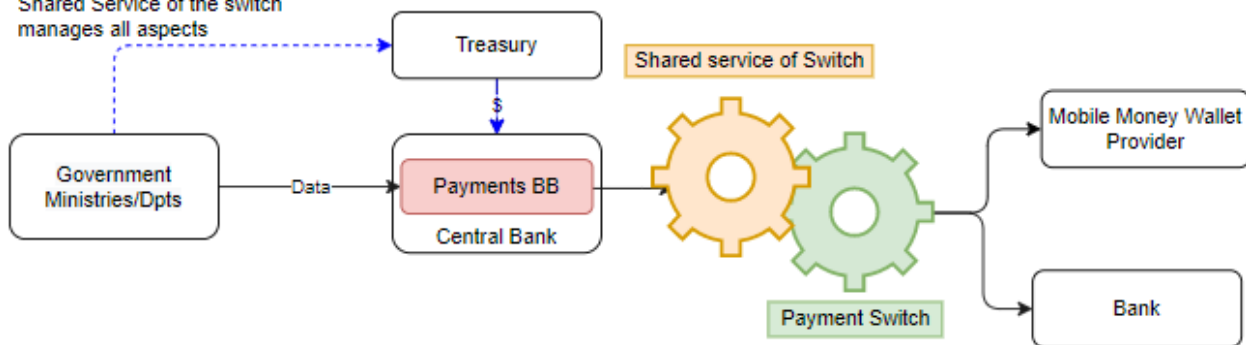
[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

### 4.2.3 Payment Infrastructure Scenario 2-B

The Central Bank plays a key role as a participant in the Payment scheme and routes the payment details and funds via a shared service of the payment switch. Same logic of break-bulk applies and with all transactions effectively "off-us".

## Scenario 2B

Shared Service of the switch manages all aspects



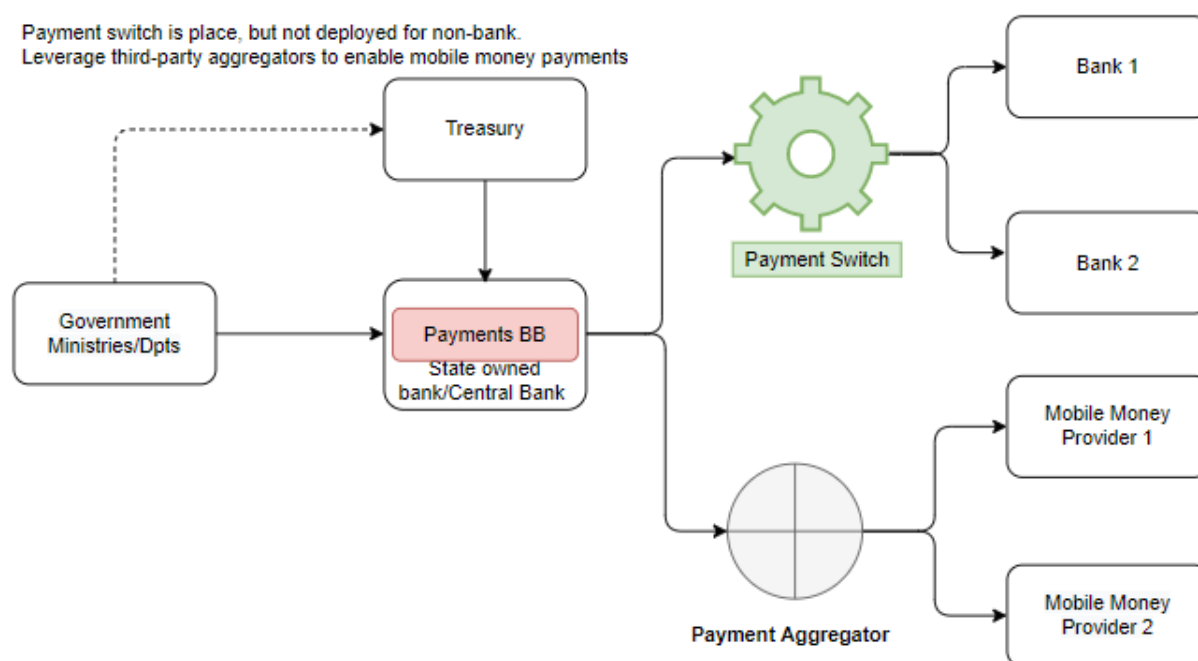
[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

## 4.2.4 Payment infrastructure scenario 3

Government should leverage third-party aggregators to enable payment use-cases to reach mobile money providers

### Scenario 3

Payment switch is place, but not deployed for non-bank.  
Leverage third-party aggregators to enable mobile money payments



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

## 4.2.5 Payment Infrastructure Scenario 4

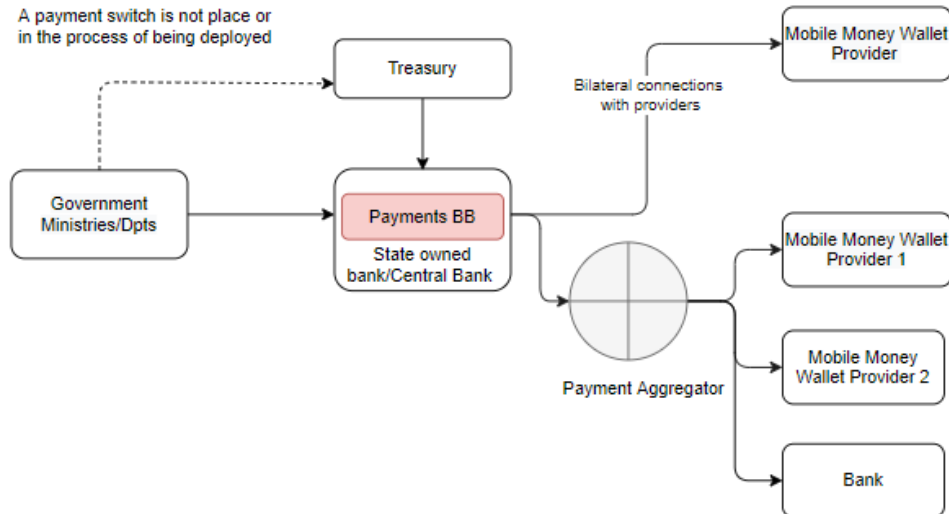
A payment switch is not in place or in the process of being deployed. Government should leverage third-party aggregators or bilateral connections to enable payment use-cases to reach mobile money providers. In the bilateral connection scenario, each mobile money provider in the country would connect to the government portal through APIs, enabling a seamless transfer of funds to end users (in the case of G2P payments) or the receiving of funds from end users (in the case of P2G payments) and



facilitate reporting, reconciliation and settlement processes.

#### Scenario 4

A payment switch is not place or in the process of being deployed



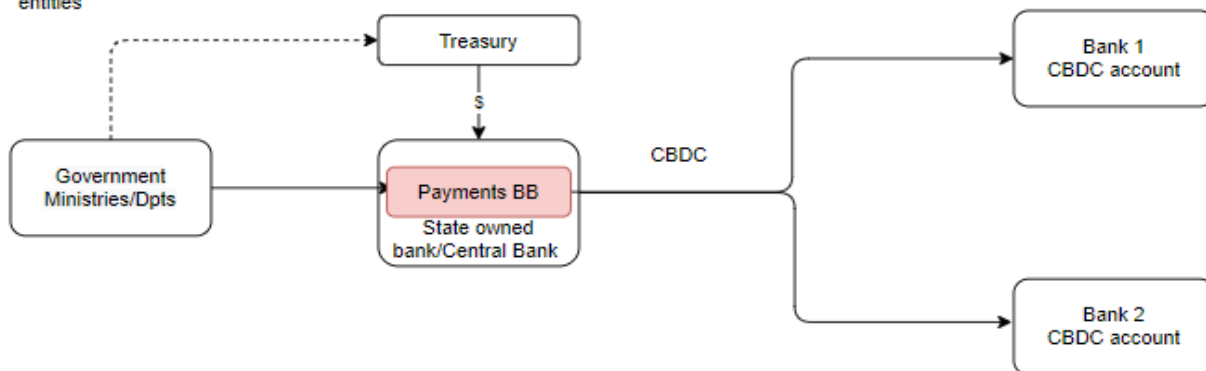
[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

## 4.2.6 Payment Infrastructure Scenario 5

The Central Bank mints digital currency and distributes funds via regulated entities (e.g. Payment Service Providers), and in the case of G2P sends funds to destination accounts known to the Ministry of Finance. Account information is unitary and unique for all beneficiaries (e.g. citizens) but the Central Bank does not "hold" account balances on an individual basis.

#### Scenario 5

Central Bank mints digital currency and distributes funds via regulated entities



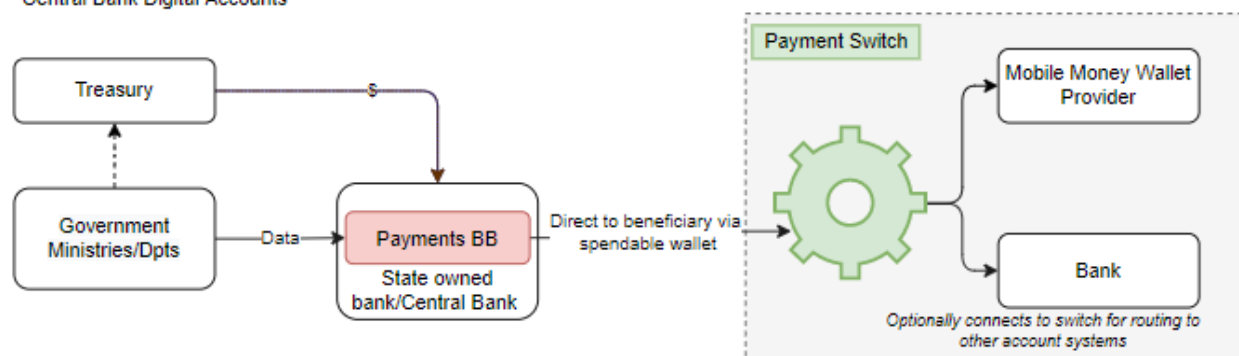
[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

## 4.2.7 Payment Infrastructure Scenario 6

The Central Bank directly provides Digital Accounts for all people (Citizens) and Treasury routes funds to the relevant accounts (uniquely and unitarily held by people).

### Scenario 6

#### Central Bank Digital Accounts



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.drawio.net)

- For this payment use case the most common scenarios currently in place are 1 to 4, scenario 5 and 6 could be envisaged later.
- In Scenario 2B it is assumed that the Payment Switch is run by the Central Bank.
- In Scenario 2A and 3 it is assumed that the Payment Switch is an independent part outside the Central Bank.

## 4.3 Prerequisites

- The Payments Building Block assumes that all identification, registration, and enrollment logic by GovStack components are done externally before using this building block and those elements are presentable if required for compliance with regulated payments and banking systems within a jurisdiction.
- Further, the Payment System or Scheme in a country *may* require that participating payor or payee entities, whether health clinics, ministries, or individuals *must* have been registered with a regulated banking or non-banking entity - i.e. for accounts - as understood in that context prior to use of the payments BB.
- In the context of CBDC or Central Bank Accounts for individuals, the payment system block would assume that the regulatory conditions must have been verified for compliance and that payor and payee entities *must* have been properly registered according to the rules in that scheme.
- The Payments Building Block assumes the conditions of statutory and operational requirements around accounts (i.e. KYC/AML/CFT) *must* have been completed by an outside

system, which are themselves capable of communicating that status in appropriate timeframes.

- The Payments Building Block assumes that capabilities of underlying infrastructure must enable transactions to meet a predefined limit to turn-around-time in real time response (e.g. < 500ms response times) when called for in immediate or instant payment systems and for queuing systems when required for management of high throughput and/or asynchronous systems.
- During operation, the Payment Block should have the ability to validate against a set of external systems for status of account, account address and routing information, confirmation of payment, and various error conditions of accounts and specific payments.
- During operation in certain modes, the Payment Block would be validating against authorizations/releases or allotments provided by the treasury systems for source of funds, and designated third party providers would perform net debit cap management prior to processing the payments. These validations SHOULD be done within the system against the available allotments/funds.
- Normally, verification of eligibility of beneficiaries for the service should be done in another block. If eligibility for a standing instruction is part of the design within this BB, then the responsibility lies with the Enrollment or Payroll system to send such standing instructions and to cancel or revoke such standing instructions.
- Budget availability must be checked before voucher creation is requested. This could be done on a bulk or individual voucher basis. For efficiency, it would be recommended to create vouchers in bulk and then only activate/preactivate them as needed.
- Calculations of payments may depend on several attributes laid down by a specific program such as Unconditional Social Cash Transfer (USCT) or Postpartum Healthcare benefits. Based on these attributes and rules the statement of payments to be disbursed must be generated in respective applications and issued to the payments BB.
- It is understood that payment systems in the market, either provided by a public entity, a quasi public entity, or a purely commercial player, are required for the functioning of this building block. That is, systemically significant systems play an important role in both origination of payments and termination of payments as well as payment clearing and settlement. To be explicit, settlement (gross or net) must be handled externally to this BB. In the absence of appropriate payment rails, this building block would require additional features and functionality. Such systems MUST provide the following functionality to this use case:
  - Rules by which transactions are initiated, halted, reversed, examined.
  - Destination system for funds and related channels by which the Recipient accesses the funds.
  - Source system for funds and related channels for the individual or institution.

## 5 Key Digital Functionalities

The payments building block provides functionalities to cater for the following payment types:

- Government to person (G2P) payments such as:
  - periodic bulk payments such as salary payments,

- unconditional social cash transfer: Cash payments provided to financially disadvantaged or vulnerable people or households without requiring anything in return. The unconditional cash transfer should support different modes/channels for payments: bank payments, voucher payments, mobile money payments and other electronic channels.
- conditional cash transfer: Cash payments provided to financially disadvantaged or vulnerable people or households aimed at changing behaviors. The payments are conditional depending upon the recipient's actions.
- Person to Government (P2G) payments: support payments for government services e.g payment of government fees, driving licenses, utility payments.

The following requirements below would be implemented by the payments BB:

- Cater for the distribution of social service transfers electronically and cash safely to end users (in the case where the user does not want to use electronic payments, it is recommended that vouchers are used).
- Creates eVouchers to disburse financial subsidies in a controlled and safe manner.
- Responds to payers to provide information about the status of the payments (e.g confirmation of payment, insufficient balance or a mismatch in credentials, transfer failure, etc.). based on status in the backend applications.
- Posts status of transactions with traceability information into transaction logs.
- Tracks due payment and sent payment notifications through alerting mechanisms, along with associated information.
- Receives triggers for payment collection, posts the amount with relevant disclaimers to payer and obtains payer approval.
- Securely posts the approval, user ID and associated payment information to appropriate backend (eg mobile, debit/credit card, Internet banking entities) of relevant financial applications from banks, employers, insurance; awaits transfer confirmation from those applications.
- Searches and provides a logged information-based query of other applications.
- Able to handle operations in remote and inaccessible locations.

## 5.1 Government to Person Payment Principles

G2P payments architecture should strive to achieve the following principles:

### 5.1.1 Beneficiary/Recipient-related Principles

- Beneficiaries can receive their payment through a fully functional account that allows them to save and make payments using an associated payment instrument with general acceptance.
- They can choose the payment service provider and payment instrument through which they receive their payments based on their informed choice; they can use the same account for multiple G2P payments, make P2G payments and easily switch if desired.

- Onboarding to their payment method is low cost and easy with account opening requirements that are available to recipients and with no opening fees; a payment instrument is provided to the recipient at no cost.
- The Integrated Financial Management Information System (IFMIS) is used to process government payments against the budgetary allocations. It supports (1) the Treasury Single Account (TSA) that aggregates all incoming government receipts and disburses all government payments and (2) budget management to ensure budget compliance, tracking, and reporting. The Integrated Financial Management Information System (IFMIS) provides the Ministry of Finance with a unified view of the government's budget execution.
- The payment instrument is easy-to-use and generally accepted and the overall costs for using it do not result in increased costs for the beneficiary in comparison to other forms of payment.
- Beneficiaries are well informed, their accounts/data is protected and have access to redress. They know when, where, and how much payments will be made and understand how their payment method works, its costs, how to use their payment instrument and how to access their payment. They can access and know how to access effective grievance redressal mechanisms, their funds are secured, and their data privacy is ensured.
- Beneficiaries can easily access their funds. They are able to cash-out their funds at any time at a wide range of conveniently close financial access points, at a reasonable and clearly communicated withdrawal fee or at no cost.
- Beneficiaries are included regardless of gender, race or other immutable characteristic, through at least one of the payment methods used; Gender gaps are considered in the design.

### 5.1.2 Infrastructure and Systems-related Principles

- The leveraged infrastructures and systems are shared across G2P programs and payment streams, as well as other use cases, avoiding the implementation of systems to exclusively deliver G2P payments. They are scalable and have cyber security arrangements in place.
- Common ID provides access to multiple programs; the national ID system has high coverage and quality; it allows government agencies and payment service providers to validate recipients' identities; it enables data sharing across government agencies.
- Payment processing systems unlike social protection management information systems, pension and payroll systems will not include gender disaggregated data.
- The payments systems are interoperable to maximize the potential of the available infrastructure to recipients; interoperability arrangements exist for Integrated Financial Management Information System (IFMIS), Treasury Single Account (TSA), banks and non-banks covering most channels and instruments.
- There is no manual intervention on the disbursement process and the entire payments flow is Straight-Through-Processing, including reconciliation of payments. Payments are made without delays.

### 5.1.3 Payment Services Provider- related Principles

- Payment service providers have a strong and long-term, predictable business case or incentives to deliver payments and provide choice.
- Large variety of bank and non-bank financial institutions operating in a competitive market are used to deliver G2P payments.
- Agents of bank and non-bank financial institutions are accessible to G2P payment recipients.

## 5.2 General Key Functionalities of the Payments Building Block

In general the functions below describe the activities/actions that are performed in the payments building block:

The payment BB architecture should:

- Allow any government program to channel payments through a shared payments infrastructure to accounts at multiple providers. This enables citizens to choose their preferred financial services provider. It enables citizens to switch providers if their circumstances change or they discover a better service.
- Connect securely and interact with registry, identity, and other important building blocks through the information mediator building block.
- Payments processing and orchestration: Handle the processing of the transactions before moving to the payment. This would include the following:
  - Check that the information provided by the user is correct and meets the requirements for the service being requested.
  - Get the confirmation of the **payee** on the details submitted or looked up via external id-account mapping.
  - Verify the destination account for the beneficiary to be in good standing - e.g. the bank has not noted a suspicious account, or suspended the account for another reason.
  - Return information from external systems on fees to be charged by providers.
  - Handle automated bulk transactions (e.g. payroll, social benefits disbursement)
- Send automated requests to the FSP/bank to effect payment to the person in the case of unconditional social cash transfers.
- Keep records of transactions history for audit purposes
- Auditability refers to the ability of the Payment's Building Block to have its controls evaluated for effectiveness, efficiency and security. Looking at auditability from an effectiveness standpoint, transactions must be capable of being traced to ensure that all transactions (100%) initiated within the systems have been completed, irrespective of whether the transaction was successful or not. This also requires the Payment Building Block to degrade gracefully under conditions of high traffic or node failure so that such conditions are captured in the audit trail. From an effectiveness point of view, auditors must

be able to trace the flow of transactions through the system in a logical flow and be able to determine the duration of each part of a transaction to determine if the optimal route is being followed or if there are bottlenecks that need to be addressed. From a security standpoint, external auditors should be able identify users who access the system and what actions they may perform in the system. In addition, the auditability function will also require the system controls to be clearly defined and access to evidence of their effectiveness either through prevention or escalation and an appreciation of the residual risk. Lastly, the system will need to be able to show compliance to local regulations and standards which will depend on the country in which it is implemented. If at some point PII or payment data is stored the system should show compliance to global best practices such as PCI-DSS v3 (or the prevailing standard at the time).

- Schedule and aggregate payments to individuals (scheduling, tracking and triggering will be done in scheduling BB if it receives a schedule of payment via an API request, any specific attributes accessible by the scheduler in addition to date/time to access and verify before triggering).
- Payment confirmation: Send receipts, notifications and acknowledgement of receipts in a secure manner such that payers and other external applications (e.g other building blocks) receive confirmation of payment made . It should also include the safekeeping of a record of the receipt that was sent.
- Event logs and audit trails: All transactions processed and their outcomes should be logged and stored securely for monitoring and audit purposes.
- Validation/Verification - The system shall validate data structures and content (i.e the list information) to ensure that it is not missing key pieces of information. Verification of the destination account is a best practice, whereby the system shall dynamically confirm that the beneficiary's account exists, is in good status, and if not, triggers a different process.
- Batch logic/ Queue - The system shall break down the bulk payment into "batches" that can be handled efficiently, or creates a queue mechanism whereby the list is sequentially processed. The logic to reduce the flow of payment initiated (called "throttling") is contemplated here.
- Scheduling - The timing of when such batches are sent is important for many reasons, including programmatic ones. Programs may seek to stagger payments or to combine with other payments to the same groups of individuals.
- Control logic - The system shall coordinate or orchestrate the validation, verification, batching, scheduling using a set of rules. Such rules also include options for kicking back to the operator for review or resubmission in certain error cases.
- Reconciliation and Reports Dashboard - The system shall be capable of generating reconciliation of accounts, based on the successful sending of funds into accounts. Reports shall show this information and additional status of systems and payment types and timings.
- Fees Calculation - Fees to be paid by the Government should be quoted back to the system and managed in a centralized way. Fee schedules can also be managed in this component

- i.e. pre-negotiated amounts that shall also leave the government source of funds to the various providers involved.

- Should support use of digital channels to/from end-users, and all regulated account systems with payment signaling (e.g. QR codes) and notifications by the FSP enablers.
- The system should have the ability to scale on transaction performance (transaction latency, reliability, resilience, graceful service degradation).
- MUST support digital tools for queries management, including beneficiary queries including complaints about the payment services.

## 5.3 Key Functionalities for Bulk Payment

Bulk payments are used to accomplish G2P beneficiary payments (eg, payroll, pension plans, and other social protection programs). Bulk payments have to be routed through the account holding the funds. (This could be either at the government Treasury or a commercial bank).

The funds should be released by the Ministry or payroll master and go through appropriate approvals.

## 5.4 Key Functionalities for Voucher Disbursement

The voucher component of the Payment Building block should support the provisioning, issuance, activation and redemption of vouchers.

- The **provisioning** process will involve ensuring that adequate funds have been allocated to the voucher.
- The **issuance** process will allow vouchers to be issued but not usable until they are in the hands of the beneficiary.
- The **activation** process will make the voucher active and hence usable for certain predefined use cases and for a certain period of time.
- **Redemption:** When the vouchers are appropriately used, the beneficiary receives the appropriate benefit (cash, product or service) from a third party (either an agent or a merchant - e.g. a school).

In terms of the reporting, this is expected to be a standard part of the voucher management system which would be capable of showing the vouchers in their different states as well as the aggregate quantity "stock". Such reports would trigger, either automatically or manually, requests for "re-stocking" of vouchers. Detailed reporting requirements are out of the scope.

The authentication of the beneficiary at the point of redemption, will lie with the calling block which would likely check the Registry building block to authenticate the user. Other flows could involve the voucher management server storing the beneficiary details but this would appear to go against the principle of avoiding duplication. This may also need consideration of consent in cases where delegation applies.



## 6 Cross-Cutting Requirements

This section will highlight important requirements or describe any additional cross-cutting requirements that apply to this building block.

The table below summarizes the cross cutting requirements.

Requirement	Relevant Building Block	Type (Must/Should/May)
Reporting	Requesting building block	Must
Compliance with applicable regulations and laws	All	Must
Merchant and beneficiary registration	Registry	Must
Account lookup and directory services to resolve accounts	All	Must

The payment building block must meet the security requirements described in the Security building block (click [here](#) for more details).

The payment building block must meet the requirements described in the Architecture building block (click [here](#) for more details).

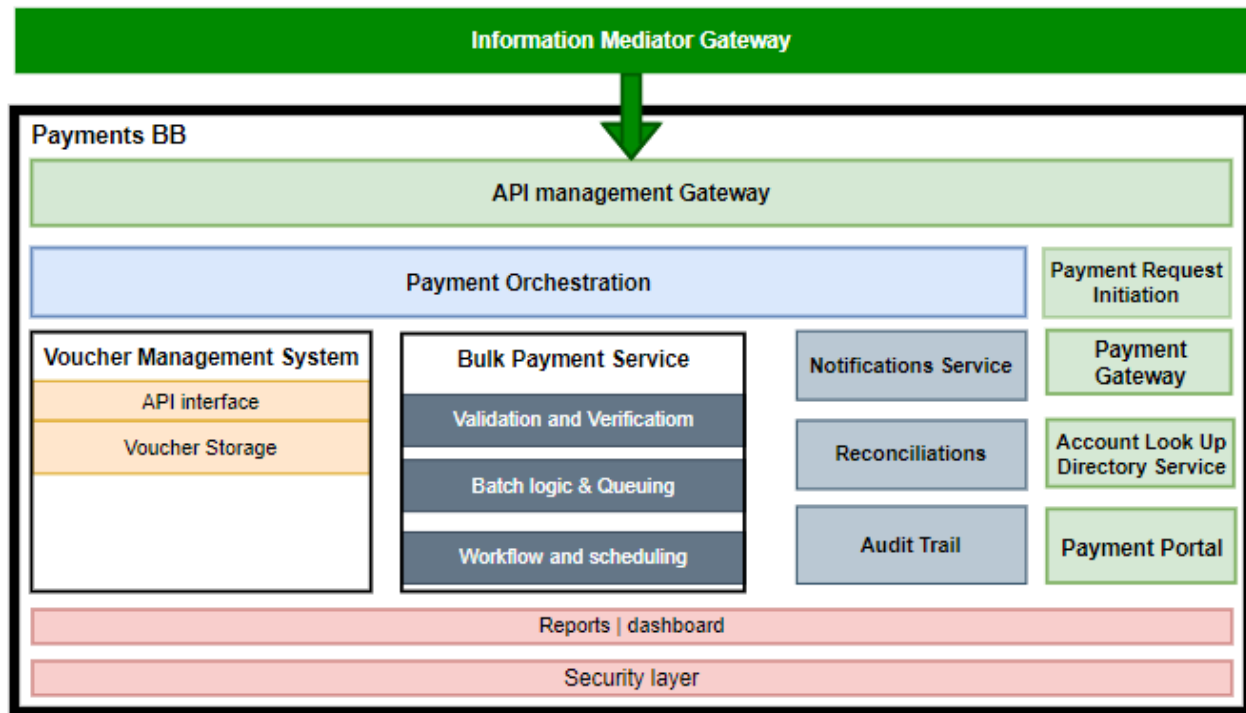
The payments BB should meet the mechanisms for consuming and publishing APIs as described in the Information mediator BB. (click [here](#) for more details).

## 7 Functional Requirements

This section lists the technical capabilities of the payments building block.

### 7.1 Payments Building Block Components

The following components are needed to achieve the technical functionalities of the payments building block.



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

### 7.1.1 API Management Gateway

Handles all the API messaging calls and API access control verification from other BBs to the Payment BB and vice versa as well as within the Payment BB. All requests from other BBs first go through the API gateway. The gateway then routes requests to the appropriate application/service. The API Management gateway will:

- Use Identity and access management for authentication.
- Perform input validation checks to prevent oversized message attacks, SQL injection attacks as well as JSON and XML threats.
- Require authentication for all API users.
- Manage access quotas and throttling.
- Log all API calls..
- Allow API providers to limit the rate of consumption for all API users.
- Transform backend error messages into standardized messages so that all error messages look similar; this also eliminates exposing the backend code structure.

## 7.1.2 Payment Orchestration

Payments orchestration provides for end-to-end workflow across different sub-subsystems, enables asynchronous processing, and covers different payment types, use cases, account systems, and channels.

The orchestration relates different sub-components or microservices whereby it:

- Explicitly defines and model workflows that span multiple microservices.
- Provides detailed visibility into how a workflow is performing and identifying where there are issues.
- Ensure that within the a defined workflow, that all workflow instances are completed according to plan—even when there are issues along the way.

Payments orchestration is used to configure the payments building block functionalities to be exercised during a specific workflow.

## 7.1.3 Voucher Management System

The voucher management system is responsible for provisioning, storage, issuance, activation, redemption, validation, suspension, unsuspension, purging and reporting on vouchers.

### 7.1.3.1 Voucher Provisioning

This process traditionally involves the generation of group vouchers against some authorized value (budget release or allotment).

- Each voucher should have a unique secret voucher number, unique identification number (voucher serial number) indicating its position in an inventory of issued vouchers, and is linked to a fixed amount of value in a particular currency. Vouchers will also be associated with a voucher group during provisioning. Voucher serial numbers will be unique across currencies should there be vouchers of multiple currencies.
- The vouchers should be created with an expiry date and MUST be stored securely. All vouchers will be expected to have the same duration of expiry and this expiry period will be from the moment the voucher is activated.
- Alternative design options include dynamically creating a voucher at transaction time and creating variable amounts, but this increases complexity and requires tighter operational controls

### 7.1.3.2 Voucher Issuance

This process involves the Registration BB (or any other BB for that matter) requesting a voucher number from the voucher management system through an API.

Once confirmation is received that the voucher has been released it flags the voucher as activated. Design decisions include making this step as optional. Having an additional step increases security by ensuring that the voucher is not used until it is in the hands of the beneficiary.

Once a voucher has been issued by the calling building block (registration bb) it will be presented to the beneficiary in a form that is determined by the calling Building Block (refer to voucher workflow). The format of the final voucher presentation will be determined by the function of the calling Building

Block. At a minimum this presentation should have the voucher number as well as the voucher serial number. It could also have the details of the beneficiary. The details of the beneficiary on the voucher presentation will also help the merchant authenticate the beneficiary at the point of redemption.

### 7.1.3.3 Voucher Redemption

In the redemption process, the merchant will authenticate the beneficiary and use a predefined technology (USSD, Mobile App, web browser) to extract the voucher number and call a redemption API through the relevant calling Building Block. The calling Building Block may also validate the beneficiary details if so required. The Building Block will also be able to validate the merchant and determine the voucher group which the merchant belongs to. Lastly, the calling Building Block will invoke the Payment Building Block Redeem API, through the Payment Building Block API Management gateway, to validate the voucher and if valid to redeem it.

Once the voucher is validated, the Voucher Management System should invoke an API on the Payment Gateway to effect the payment in the merchant or agent wallet or bank account (depending on what was set up at merchant / agent registration). The payment gateway/switch will debit a prefunded account / wallet and credit the merchant / agent account / wallet. The successful execution will result in the voucher being flagged as consumed/used.

## 7.1.4 VMS API interface

The details of the VMS APIs are described in the Service API section.

The VMS functionality can be accessed by four external APIs:

- **Voucher preactivate:** An API call to pre-activate the voucher and get a voucher number, a voucher serial number and expiry date.
- **Voucher activate:** An API call to activate the vouchers by serial number.
- **Voucher validity:** An API call to check the validity of a voucher by serial number.
- **Voucher redeem:** An API to redeem the voucher by voucher number

The API interface should provide a minimum of five internal API calls.

- Voucher preactivation: An API to pre-activate the voucher and get a voucher number, a voucher serial number and expiry date
- Voucher activation: A call to activate the voucher, by serial number
- Voucher validity:
  - A call to check the validity of the voucher by serial number
  - A call to check the validity of the voucher by voucher number and group
- Voucher consumption: A call to consume the voucher by voucher number

## 7.1.5 Voucher Storage

The voucher management server shall have a storage subsystem to store the vouchers.

- Vouchers must be stored in a secure but high performance data storage (READ access will marginally exceed WRITE access).
- The storage of the voucher number will require encryption of data at rest and in motion (unless the channel is encrypted).
- Logs generated should NEVER contain the voucher numbers.
- All access to the voucher database MUST be logged.
- Segregation of duty must be done with respect to privileged access to the database and key management of the encryption of the voucher. (Level of key management may need to be determined).

### 7.1.6 Account Lookup Directory Service (Mapper)

The account lookup directory service identifies the FSP where the merchant/agent/payee's account is located.

The account lookup directory service or mapper simplifies the payment routing and is an important component to avoid storing the payment information of the user in the social registry system and preserve the privacy and confidentiality of sensitive information pertaining to the beneficiary.

The account lookup directory service provides a directory that maps the beneficiary's unique identifier (which matches the record in the social registry system) to the transaction account where the beneficiary wishes to receive their G2P payment. allowing the government to address payments to a specific individual. The identifier can be a national ID, phone number, or other number or alias that can uniquely identify individuals across social protection and financial sector databases. The information will be kept in a tokenised form in the account lookup directory service.

In the case where there is a national payment switch, the account lookup directory service will be maintained by the FSPs. In the scenario, where there is no payment switch, the government would need to manage the account lookup directory service and provide a mechanism for linking it to the FSPs.

### 7.1.7 Payment Request Initiation

This request could come from two sources: external or internal. An external source could be another GovStack Building Block (e.g. the Registration BB or Social Benefits Registry BB or Payroll ). Either source must be appropriately authenticated and authorized to initiate the request. The initiation could be synchronous (typically for a single payment) or asynchronous (typically for batch payments). The request should contain at a minimum: the payer identifier, the payee identifier, the amount, the currency, the policy, and the initiating source's unique transaction ID. In the case of the internal payment request it should also contain an ID provided by the payment orchestration module.

Certain processes in the transaction flow might require proof of intent from the user, for example, entering the PIN/Password or pressing an 'accept' key to initiate the payment process. Such events and their outcomes should be recorded for audit trail purposes. However, the PINs and passwords should not be stored in logs or if they have to, PINs and passwords should be hashed out.

### 7.1.8 Payment Gateway

Payment gateway allows different (digital) financial service providers (FSPs) to:

- interconnect and exchange information.
- initiate and receive transactions.
- Accept or reject transactions and debit or credit end user accounts.

### 7.1.9 Payment Portal

The payment portal will:

- Enable government payers (departments/ministries/public sector bodies) to make G2P payments through one or multiple FSP/payment service providers.
- Focus on managing, sending, and reconciling payments.
- Connect government agencies to sending and recipient institutions.
- Coordinates sending / receiving of payment requests, approvals.
- Provide reports across G2P programs, FSPs and beneficiaries.
- Track payments status and payments history.
- Register FSP who are entitled to process payments for G2P payments.
- Provide a restricted access to authorized FSP to connect to the portal, to process the payments for beneficiaries who are their customers.
- Automate reconciliation.
- Provide data analytics on payments processed and their status, whether successful or not.

### 7.1.10 Notifications Service

Support different events related to triggering specific actions on payment outcomes such as issuing receipt upon successful payment, automating payments in case of bulk transactions, passing information to other building blocks as necessary and handling of exceptional cases for instance user/system errors amongst others.

All notification events shall have a timestamp associated with it and kept as part of the audit log.

### 7.1.11 Reconciliation

This should happen at two levels: internally and externally.

- The internal reconciliation will occur between the different sub-blocks within the Payments BB. In order to achieve internal reconciliation the internal payment request initiator should issue a unique ID that will be referenced by subsequent Payment BB sub-blocks in all future calls related to the particular payment. This will allow end to end tracing of the transaction within the

Payment BB.

- The external reconciliation is more complex as it involves a calling BB which is outside the Payment BB (such as the Registration BB) and a third party (e.g. DFS). Ideally, there needs to be a sequence of IDs that can identify a transaction from start to finish.
- Cross-cutting Prerequisites for reconciliation:
  - The nodes that are under the control of the GovStack should be time synched.
  - IDs should be unique, if possible contain the timestamp, and should not rollover across short ranges.
- Transactions are expected to be irrevocable. Transaction reversals are subject to local regulations. In some countries, a transaction is revoked/clawed back if the beneficiary has not withdrawn the payment within a certain time period. Good practice, from a financial inclusion perspective, is to not claw back. Beneficiary could use this money as savings when he/she needs it. The system should allow both configurations.

### 7.1.12 Validation and Verification

Batch files go through a final check to be clean of defects and inconsistencies, to check with external systems as necessary:

- Low level validation of data types and data completeness.
- Verification of lookup of accounts to ensure that the account information matches the destination system expectation.
- Check for inconsistencies.
- Auto correct items as possible - consistency logic can be applied to fill in missing formatting if required by recipient banks and telecoms.
- Errors are kicked back to program level or to an internal data review process.

### 7.1.13 Batch Logic and Queuing

- Prepares the batch breakdown on the basis of rulesets governing which FSPs shall receive which payments and other considerations.
- Combines payments with other payments to the same beneficiary.
- At high volumes, batches are queued for processing.
- Detects batch failure rates.

### 7.1.14 Workflow and Scheduling

- In relation to batch logic, Payments are scheduled against the availability of systems, throughput limitations, and rules set by programs.

- Regular and repeat payments are scheduled.
- Batches may be given prioritization in the queue.
- Essential control logic may be included here specific to the individual batch sending and resending.
- Availability of funds in different budget accounts may be incorporated into this process.
- Additional workflow checks as required, including resending of failed transactions.

### 7.1.15 Event Log

Each component of the payment block should be capable of producing both application and transaction logs. This is important to ensure that the system can be adequately monitored and troubleshooting can be performed efficiently and effectively.

- Application or event logs will capture events that each component performs and should contain at least the following information:
  - application / user ID
  - event date and time
  - terminal identity (name and / or IP address)
  - event related information (message or code)
  - event success or failure
- The components should also generate transaction logs which capture at least the following information:
  - transaction date and time
  - transaction source
  - transaction destination
  - supplementary data
  - transaction status (success, failed, in progress)
- The event logs and the transaction logs should NOT capture any sensitive data such as voucher numbers, passwords, etc.
- There should be an individual transaction record for every payment transaction. For example, if a batch payment process is executed there should be a transaction record for each individual transaction and a separate event log for the entire batch.

### 7.1.16 Audit Logging

Audit trails are required to provide assurance on the integrity of the requests received and actions taken on these requests. An audit trail is a chronological record of system activities that is sufficient to enable the reconstruction and examination of the sequence of environments and activities surrounding or leading to an operation, procedure, or event in a transaction from inception to final results. The audit trail shall comply with the following requirements:

- Must be automatically captured by the computer system whenever a payment request is created, modified or deleted.



- Must be stored in a secure manner and must not be editable by any user, including privileged users. A common approach is to copy / redirect logging to a separate logging server.
- Each audit trail entry must be time stamped according to a controlled clock which cannot be altered. The time should either be based on central server time or a local time, so long as it is clear in which time zone the entry was performed.
- Each audit trail entry must be traceable, i.e. attributable to the individual responsible for the direct data input. Updates made to data records must not obscure previous values and where required by regulation the reason for changing the data must also be recorded.
- The audit trail must be retained as long as the electronic record is legally required to be stored.
- The audit trail must be available for internal review/audit..
- The auditing system must be self sufficient, i.e., for auditing/regulatory purposes the information stored by a BC must be enough.
- Audit entries cannot be changed, ie, the audit BC persisted store should be immutable, should only support append and not changes or deletes.
- Audit entries, should have a pair of fields to store encrypted data and the encryption key id - these are provided by the submitting BCs.
- Querying capabilities:
  - Auditor wants to see all activity in the last X days.
  - Auditor wants to see all activity of action type Y, or from a certain BC.
  - Auditor should be able to request the decryption of encrypted data to an operator.
- Access to the audit store must be securable, so whatever tech is chosen must implement access control mechanisms (ideally that can connect to our IAM provider/connector).

### 7.1.17 Reporting

- The data store will be write-only from the core service and should be read-only by external components.
- The data model on the reporting data store can be different from the internal operational data models that the switch uses.
- The component provided by the switch will be translating internal events and internal data models to the external data store models - This component can be replaced.

### 7.1.18 Security layer

Protects the system at the transport and application levels. It provides the necessary APIs for encrypting the data in transit and when stored as well as the digital signatures used for API

authentication. The digital signatures are based on public key cryptography using X.509 digital certificates.

At the transport layer:

- All communication between building blocks must be TLS-secured using *client authentication*, Transport Layer Security protocol (TLS) 1.2 and above should be used to protect the confidentiality and integrity of the data in transit.
- Strong authentication for parties involved in the transactions should be supported.
- Confidentiality of personal information related to the transaction – Information on account data, transaction data, payment credentials and users'/payee' personal profiles must never be disclosed to any unauthorized party.
- Non-repudiation of transactions by parties involved.
- Acknowledgement receipt - This will result in creating a trusted communication path for all transactions between each party, be they end users, telecommunication companies, merchants or banks.
- The messages concerning the payment transaction shall be authenticated.

### 7.1.19 Data Protection

Use of a hardware security module (HSM) or equivalent to provide cryptographic keys for critical functions such as encryption, decryption and authentication for the use of applications, identities and databases.

## 7.2 Payments Building Block Technical Requirements

Requirement	Type (Must/Should/May)
Secure API exposure: All APIs exposed via secure socket connections (HTTPS)	MUST
Client application authorization tokens: Client applications must send authorization tokens in the authorization header of the request to authenticate users and the API Management Gateway will verify whether the token is valid.	MUST
Transaction receipting: For each disbursement made by the system, a receipt should be issued to the recipient of the funds containing information about the transaction id, transaction date and time, reason for the payment, details of the payor, and the system should store all receipts issued for easy reference and reconciliation.	MUST
Transaction status querying capability	MUST
Display details of the transaction to the payer	MUST

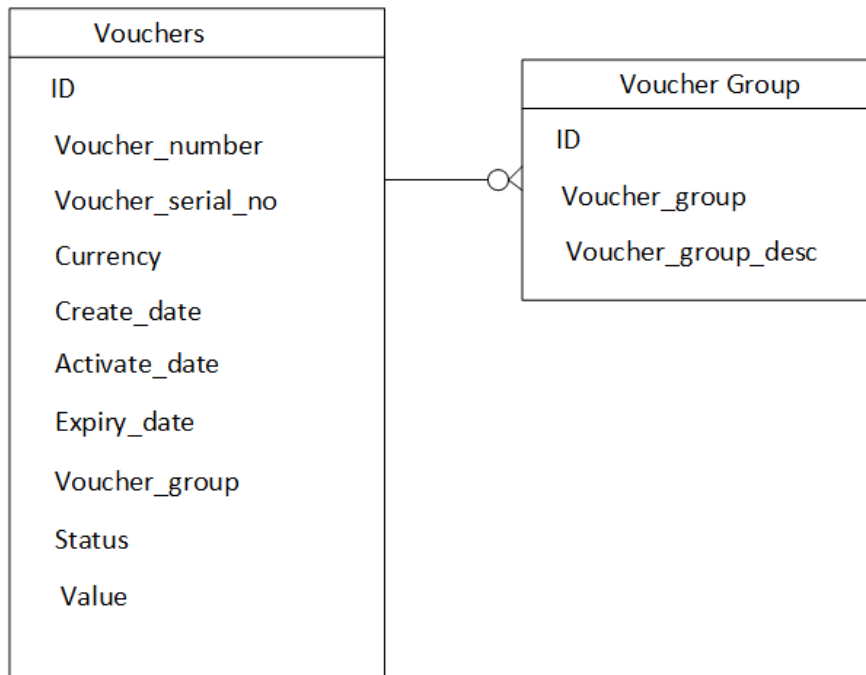
Requirement	Type (Must/Should/ May)
Transaction pre approval: Prior to processing bulk payment and batch payment transactions, the transaction must be authorized and approved in the system.	MUST
Pre-processing validation of data (well formed data and completeness checks) prior to disbursement.	MUST
Ability to schedule bulk payments	MUST
Support for batch payments	MUST
Ability to lookup payment addresses	SHOULD
Error handling and reconciliation	MUST
Transaction status logging	MUST
Triggering of payments	MUST
<b>Voucher Management</b>	
Batch generation of vouchers	MUST
Voucher uniqueness and randomness: Voucher pin must be unique and unpredictable	MUST
Secure voucher data storage	MUST
High availability of storage	MUST
Expose APIs that can be invoked by voucher serial number for purposes of querying a voucher, suspending or unsuspending a voucher	MUST
Support an API that can be invoked to redeem a voucher using the voucher number.	MUST
Support an API that can invoke payment gateway	MUST
<b>Mobile Payments</b>	
Real time debiting / crediting of mobile money accounts	SHOULD
Regular balance reconciliation with disbursement Agency	MUST
Documented process for partners to dispute transaction records	MUST
Target account identifier or lookup of payment address	MUST
Ability to retrieve details of completed transactions in the batch	MUST

Requirement	Type (Must/Should/ May)
<b>Bulk Payments</b>	
Ability to securely receive bulk payment requests as a single HTTPS request containing data for multiple transactions. The transaction data (with payment instructions data) is passed format and will be compatible with ISO 20022.	MUST
The number of transactions that can be included in a single batch is limited by the size of the file upload and the processing time. If the number of transactions in the file exceeds the file size and could impact the performance of the system, the batch should be split into multiple batch requests.	MUST
Batch files should be verified for any errors and validated as per business rules and regulations before it is accepted as a valid bulk payment file.	MUST
Batch files containing duplicate payments will not be processed and an error will be generated.	MUST
The bulk payment process has to be explicitly triggered by an authorized user. All requests to the Bulk Payment Application API must be authorized and digitally signed by the person initiating the bulk payment request.	MUST
The batch file for bulk payments should contain the beneficiary ID token, amount to be paid. The payment information is not included in the batch file for security and privacy but resolved by the verification and validation component of the bulk payment service by invoking the Account Lookup Directory Service (ALDS/ALS)..	MUST
The Bulk Payment Application API shall inspect the batch disbursement file and split transactions into bank payments and non-bank payments (e.g. Mobile money) in separate payment files before initiating the call to the Payment Gateway.	MUST
The status of the bulk payment transaction can be obtained from the event log. The payment status code indicates the status of a single payment transaction and will be according to ISO 20022 Payment status codes table.	MUST

## 8 Data Structures

### 8.1 Voucher Resource Model

The voucher resource model is shown below.



#### 8.1.1 Minimum Required Data

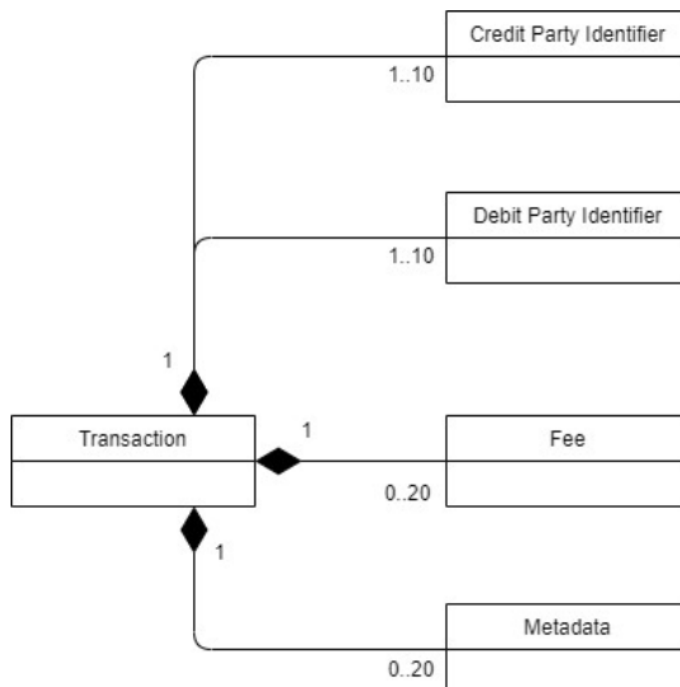
Data element	Description	Type	Required
Vouchers ID	Unique voucher identifier	Int64	yes
Voucher_Number	Secret voucher number	Varchar	Yes
Voucher_serial_no	Unique voucher identifier for external parties	Varchar	Yes
Currency	Voucher currency	Varchar	Yes
Create_date	Date when the voucher was created	Date	Yes
Activate_date	Date when the voucher was activated	Date	Yes
Expiry_date	Date when voucher will expire	Date	Yes
Voucher_group	Voucher group	Varchar	Yes

Status	Status of the voucher (e.g. ACTIVATED, SUSPENDED, CONSUMED, etc.)	Varchar	Yes
Value	Value of the Voucher	Double	Yes

### 8.1.2 Voucher Groups

Data element	Description	Type	Required
Vouchers ID	Unique voucher group identifier	Int64	Yes
Voucher_group	Voucher group short code	Varchar	Yes
Voucher_group_desc	Voucher description	Varchar	Yes

## 8.2 Incoming Government Payments Resource



## 8.3 Data Elements

### 8.3.1 API Name: Voucher APIs

Name	Description	Type	Required	Standard	Notes
voucher_amount	Denomination of the voucher required	Integer	Y		Input
voucher_currency	The currency of the voucher	String	Y	ISO 4217	Input
voucher_group	The group of the voucher	String	Y		Input
voucher_number	The voucher number (PIN). This is the secret number of the voucher.	Integer: 64-bit	Y		Output
voucher_serial_number	The voucher serial number.	Integer: 64-bit	Y		Output
expiry_date	The expiry date of the voucher.	String: date-time	Y		Output
status	The status of the process: SUCCESSFUL or FAILED	String	Y		Output
Merchant_group	The group of the merchant is captured in the registry.	String	Y		Input
Gov_Stack_BB	Calling GOV Stack Building Block	String	Y		Input
merchant_bank_details	Merchant / agent payment details	String	Y		Input
merchant_name	Merchant name	String	Y		Input
Override	Override	Boolean	Y		Input
Result_Status	Result of the process	String	Y		Output

### 8.3.2 API Name: Bulk Payment

Name	Description	Type	Required	Standard	Notes
Program_name	Program that beneficiary is participating in	String	Y		Input
Program_category	Program Category	Integer	Y		Input
Beneficiary_name	First and last ...	String	Y		Input
Identity_token	Internal to government systems	Integer: 64-bit	Y		Input
Identity_info	Other identity information as required	String?	Y		Input
Set recurring	type of payment	Boolean	Y		Input
Start Date	start date	Date			Input
End Date	end date	Date			Input
Batch_ID		Integer			Output
Batch_note		String			Output

### 8.3.3 API Name: Incoming Government Payment

Name	Description	Type	Required	Standard	Notes
Program_name	Program that beneficiary is participating in	String	Y		
Program_category	Program Category	Integer	Y		
transactionReference	Unique reference for the transaction. This is returned in the response by the API provider.	string	y		
requestingOrganisation TransactionReference	A reference provided by the requesting organisation that is to	string	y		



Name	Description	Type	Required	Standard	Notes
	be associated with the transaction				
creditParty	A series of key/value pairs that enable the credit party to be identified. Keys include MSISDN and Wallet Identifier. creditParty must be supplied if debitParty is omitted. If debitParty is supplied, then creditParty is optional	array			
debitParty	A collection of key/value pairs that enable the debit party to be identified. Keys include MSISDN and Wallet Identifier.	array			
transactionStatus	Indicates the status of the transaction as stored by the API provider.	string			
amount	string	The transaction amount.	y		
currency	string	Currency of the transaction amount	y	ISO 4217	

## 8.4 Account Identifiers

The Account Identifier enumeration lists all possible means to identify a target account. Identifiers can be combined if necessary, to provide a unique identifier for the target beneficiary account.

Code	Short Description	Type	Description
accountcategory	Account Category	string	Can be used to identify the sources of funds category where there are multiple accounts (wallets) held against an account holder.
bankaccountno	Bank Account Number	string	Financial institution account number that is typically known by the account holder.
accountrank	Account Rank	string	Is used to identify the rank of the source of funds where there are multiple accounts (wallets) held against an account holder.
identityalias	Identity Alias	string	An alias for the identity, e.g. short code for an agent till.
iban	IBAN	string	Internationally agreed system of identifying bank accounts across national borders to facilitate the communication and processing of cross border transactions. Can contain up to 34 alphanumeric characters.
accountid	Account Holder Identity	string	Identifier for the account holder.
msisdn	MSISDN	string	Must contain between 6 and 15 consecutive digits First character can contain a '+' or digit Can contain spaces.
swiftbic	SWIFTBIC	string	A bank identifier code (BIC) is a unique identifier for a specific financial institution. A BIC is composed of a 4-character bank code, a 2-character country code, a 2-character location code and an optional 3-character branch code. BICs are used by financial institutions for letters of credit, payments and securities transactions and other business messages between banks. Please refer to <a href="#">ISO 9362</a> for further information.

Code	Short Description	Type	Description
sortcode	Bank Sort Code	string	Sort code to identify the financial institution holding the account.
organisationid	Organisation Account Identifier	string	Used to identify the organisation for which a payment is to be made.
username	Username	string	Used to identify target account via an associated username.
walletid	Wallet Identifier	string	A means to identify a mobile money wallet, particularly where multiple wallets can be held against an MSISDN. typically used in conjunction with MSISDN or identity alias to identify a particular wallet.
linkref	Link Reference	string	A means to uniquely identify an account via an account to account link. E.g. wallet account link to bank account.
consumerno	Consumer Number	String	Identifies the consumer associated with the account.
serviceprovider	Service Provider	String	Provides a reference for a Service Provider.
storeid	Store ID	String	Identifies the transacting store / retail outlet.
bankname	Bank Name	String	Name of the bank.

## 9 Service APIs

### 9.1 Incoming Payments to Government (P2G)

The implementation will be such that a "Request to Pay" APIs is exposed and the Gov't Ministry (entity) is treated as a type of *Biller*. Refer to the [GSMA API](#)

### 9.1.1 Payee-Initiated Merchant Payment

The government entity initiates the request to the FSP and will be credited when the payer approves the request. This API covers the use case where the mother pays for registration payment.

```
{
  "amount": "200.00",
  "id": "2",
  "debitParty": [
    {
      "key": "accountid",
      "value": "2999"
    }
  ],
  "creditParty": [
    {
      "key": "accountid",
      "value": "2999"
    }
  ],
  "currency": "RWF"
}
```

### 9.1.2 Payer-Initiated Merchant Payment

The payer initiates the request and will be debited upon successful completion of the request.

```
{
  "amount": "200.00",
  "debitParty": [
    {
      "key": "accountid",
      "value": "2999"
    }
  ],
  "creditParty": [
    {
      "key": "accountid",
      "value": "2999"
    }
  ],
  "currency": "RWF"
}
```

## 9.2 Bulk Payment APIs (Outgoing)

There are APIs:

- that connect the Payments Building Block to the Source of Payee (Beneficiary system)
- for sending bulk payments through the gateway to the FSPs.

- APIs for doing lookup of identity and maps to valid bank or wallet accounts. As noted previously, third party providers, depending on the topography of the payments landscape in the country may bring additional APIs to connect to the FSPs. Those are out of scope.
- APIs for querying the payments building block for information about a batch job, payments made under a specific program over time, and specific payment enquiries for a specific date, beneficiary, or any combination.

## 9.3 From Source Beneficiary System to Payments Building Block

### 9.3.1 Programs

(noun, meaning a program that sends funds to beneficiaries)

```
POST/program
    Create Program
GET/programs
    Get all Programs
GET/program/{program_id}
    Get Program by id
```

### 9.3.2 Beneficiaries

(noun, meaning a payee of a program)

```
GET/beneficiaries
    Get Beneficiaries (list of beneficiaries)

GET/beneficiary/{beneficiary_id}
    Get Beneficiary by id
"beneficiaries": [
  {
    "id": 9,
    "firstname": "Mitta",
    "lastname": "Agarwal",
    "email": "",
    "mobile": "",
    "active": true,
    "activity_ids": [],
    "activity_state": "",
    "activity_summary": "",
    "bank_account_id": {
      details here
    },
    "identity_data_kyc": {
      "passport_id": "",
```

```

        "national_id": "",
        "ssn": ""
    },
    "identities": {},
}
},
*note that we don't assume KYC is handled by GovStack

PUT/beneficiary/{beneficiary_id}/kyc
    Update Beneficiary KYC

PUT/beneficiary/{beneficiary_id}/Provider
    Update Beneficiary Financial Service Provider

POST/enroll-into-program
    Enroll into a Program

POST/de-enroll-from-program
    De-enroll beneficiary from a program

```

### 9.3.3 Disbursement

(verb, relating to sending funds in a batch)

```

GET/batches
    Get all batches

POST/batches
    Create Scheduled Batch
    "params": {
        "name": "Regular Benefits Batch",
        "program_id": 1,
        "date_start": {},
        "date_end": {},
        "active": true,
        "state": "draft",
        "note": "Note 3",
        "approved-by": "person, office",
        "approval-tracking": "tracking acct num",
        "source": "Source account"
    }

GET/batch/{transaction_batch_id}
    Get batch details

POST/map-beneficiaries
    Mapping Beneficiaries (relates batch to beneficiary)
    beneficiaries": [
        {

```

```

    "batch_id": 16,
    "bank_account_id": 2,
    "beneficiary_id": 2,
    "amount": 100,
    "currency_id": 2,
    "date_start": {},
    "date_end": {},
    "note": "Disbursement for specific purpose payment"
  },

```

```

POST/transaction/{transaction_batch_id}
  Create transaction for batch

```

```

GET/transaction-status/{transaction_batch_id}
  Transaction Status

```

see this: ([https://app.swaggerhub.com/apis/rrkas/open-g\\_2\\_p\\_erp/1.0#/](https://app.swaggerhub.com/apis/rrkas/open-g_2_p_erp/1.0#/))

## 9.4 From Payments Building Block to Lookup Directories (or Similar)

[Account-Lookup Service · GitBook \(mojaloop.io\)](#)

## 9.5 From Payments Building Block: Bulk Payment to FSPs

```

GET /batch
  Batch Summary
GET /batch/detail
  Batch Details

```

```

POST/transfer...

```

see [https://app.swaggerhub.com/apis/myapi943/payment-hub\\_ap\\_is/1.0#/](https://app.swaggerhub.com/apis/myapi943/payment-hub_ap_is/1.0#/)

## 9.6 Voucher APIs (Outgoing)

The first API call (pre-activation) is a request for a voucher of a specific value in a specific currency. The API call may also include a voucher group indicating that the voucher is to be used for a specific purpose. The voucher management server will respond with a voucher number - typically a 16-digit code, a voucher serial number and the expiry date. The voucher would be marked in a pre-activated state.

The second API call (activation) is a request to activate a pre-activated voucher. This call would send the voucher number to the Payment Building Block to have the voucher activated.

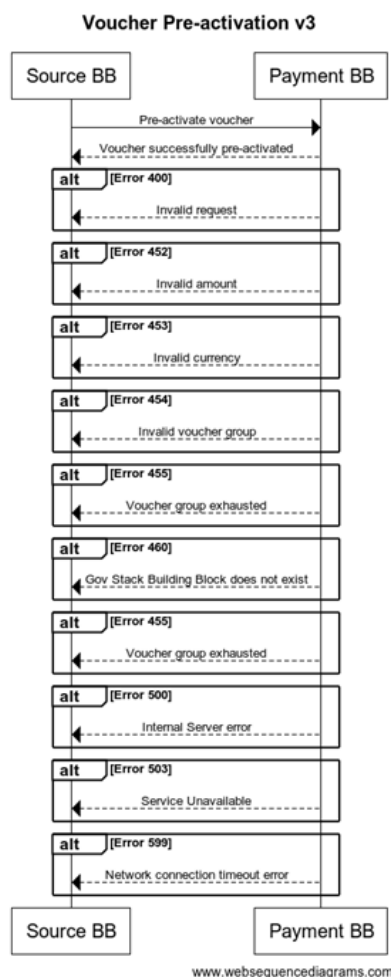
The third API call (redemption) sends the serial number, the voucher number and the merchant payment details to the Payment Building Block. If the voucher details are valid, the merchant is credited and the voucher is consumed,

A fourth API allows for batch activation of vouchers through an encrypted file. The source file would contain details on the amount, the currency and the voucher group while the encrypted response file would contain the voucher serial number, the voucher number and the expiry date.

A last set of APIs are available for checking the status of a voucher as well as canceling a voucher.

## 9.6.1 VoucherPreActivation API

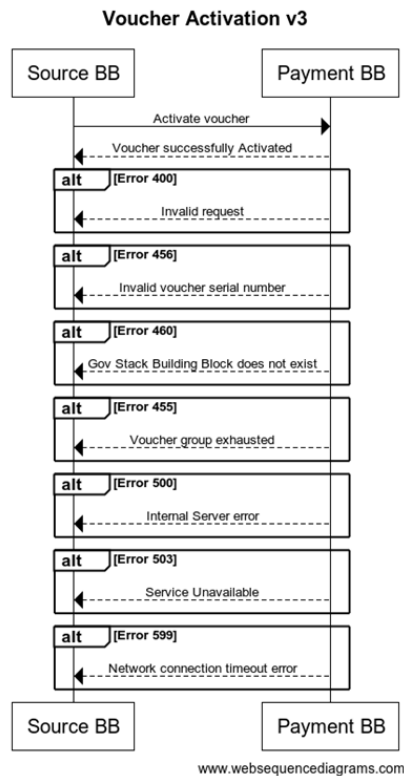
The VoucherPreActivation API is used by non-Payment Building Blocks in the GovStack Framework to request for a voucher to be used. This call reserves the voucher (for a period, which is to be implemented). This API requests a single voucher from the voucher server that can be used for a future redemption process. The caller sends an amount, a voucher group (depending on the use case), the currency and the name of the calling GovStack Building Block. If the API call is successful, the Payment Building Block will respond with a voucher number, a voucher serial number and an expiry date.





## 9.6.2 VoucherActivation API

The VoucherActivation API is used by non-Payment Building Blocks in the GovStack Framework to activate a pre-activated voucher. This second function call is intended to ensure that the voucher is only activated when it is disbursed. This API requests for the activation of a voucher when the caller sends the voucher number to be activated. If the API call is successful, the activation is confirmed, and the voucher can now be used by the beneficiary.



## 9.6.3 BatchVoucherActivation API

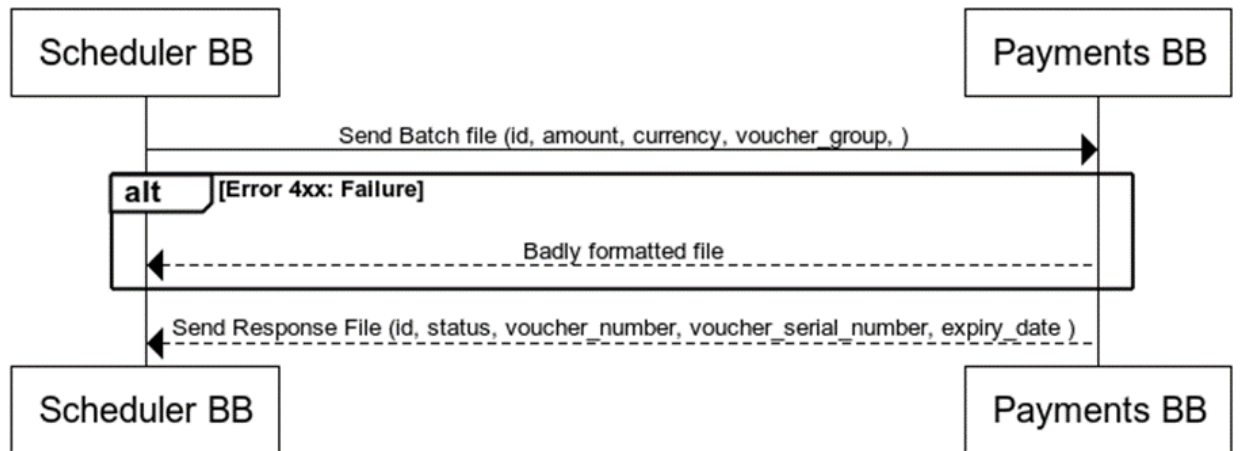
The BatchVoucherActivation API is used by a calling BB to activate vouchers in bulk. This may be used for bulk social cash transfers where the recipients receive benefits by vouchers. The calling BB is responsible for generating the beneficiary file as well as dispensing of the vouchers. The Payment BB is responsible for generating and redeeming the vouchers codes. Both BBs will have had to have exchanged encryption keys at the implementation phase.

The file provided by the calling building block (typically the scheduler building block) will typically contain a unique identifier, an amount of the voucher required, the currency of the voucher and voucher group. While the file format may vary, the recommended file format is an encrypted json file.

If the file is properly formatted, the Payment BB will respond with a file that contains the unique ID that was sent with, the status, the voucher number, the voucher serial number and the expiry date of the voucher. The response file will also be an encrypted file to ensure the appropriate security of the voucher number.

The calling building block will dispense the vouchers as needed using an appropriate delivery mechanism. The calling BB will also be responsible for any verification of the beneficiary during the redemption process. The description of the dispensing of the vouchers once the calling block has received it is outside the scope of the Payments BB.

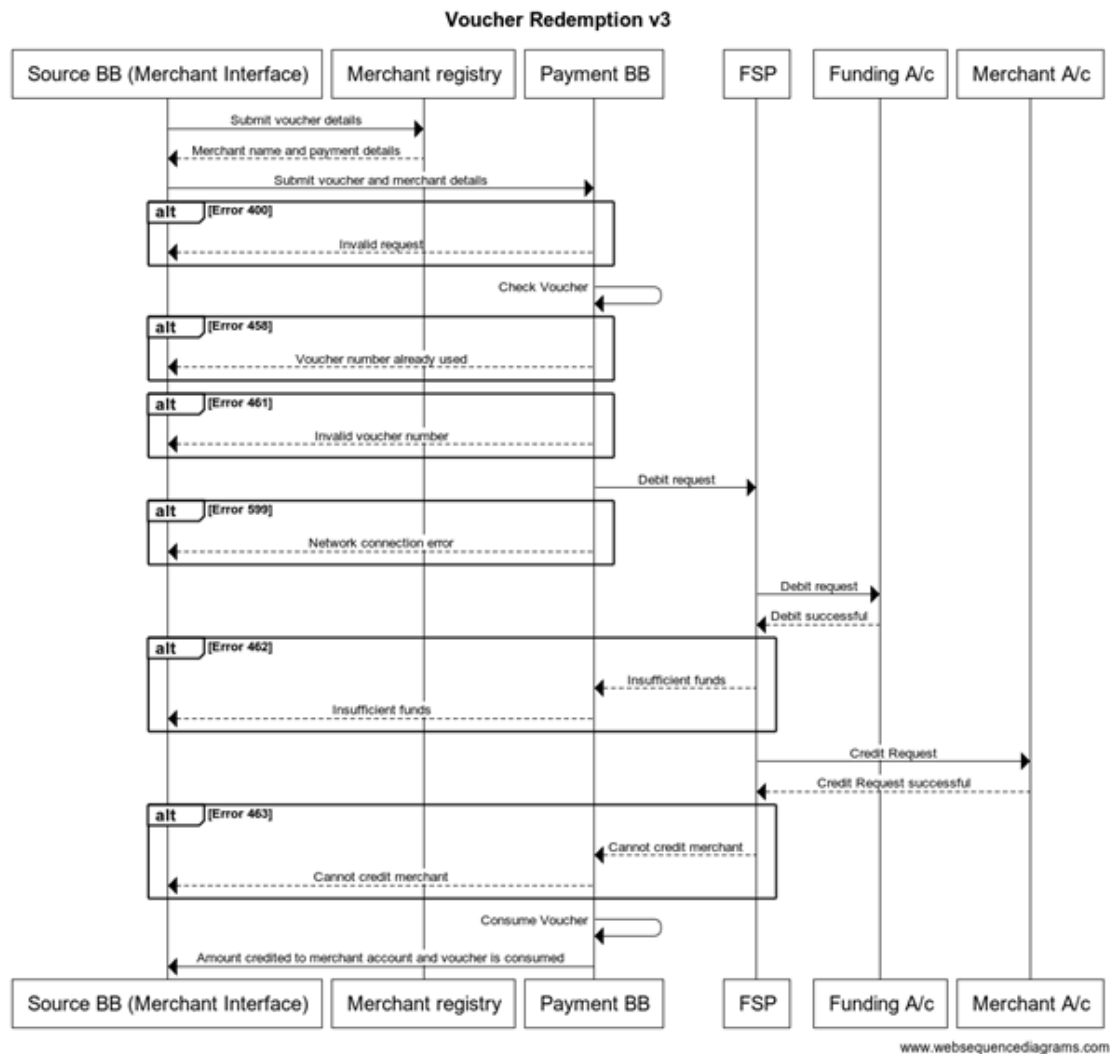
### Voucher Batch Payment (Asynch)



www.websequencediagrams.com

## 9.6.4 VoucherRedemption

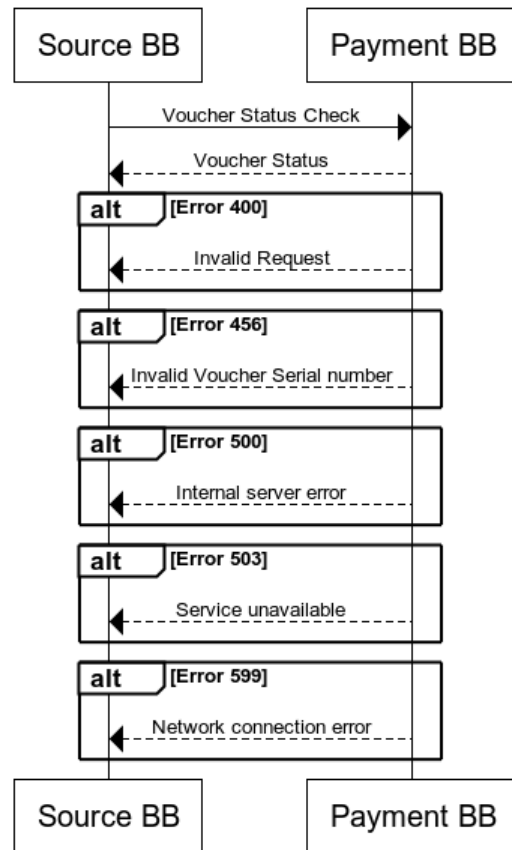
The VoucherRedemption API is used by non-Payment Building Blocks in the GovStack Framework to redeem a voucher. The calling Building Block will capture the voucher number and the voucher serial number from the merchant point. The external Building Block will also acquire the merchant name and payment details from the merchant registry. The calling Building Block will then send the voucher number, the voucher serial number, the merchant's name and payment details. The Payment Building Block will verify that the voucher has been activated and has not been used or blocked or cancelled. If so, the Payment Building Block will then send a payment request to the funding agency / FSP. The Payment Gateway of the Payments Building Block will facilitate the debit of the funding account, and the credit of the merchant as well as handle any intermediary fees. Once the payment has been successfully done the Payment Building Block will mark the voucher as consumed and notify the merchant (and beneficiary if possible).



## 9.6.5 VoucherStatus API

The VoucherStatus API is used by non-Payment Building Blocks in the GovStack Framework to check the status of a voucher. The calling Building Block will capture the voucher number and send it to the Payments Building Block to determine the status of a voucher. The Payments Building will respond with one of the statuses of Pre-Activated, Activated, Suspended, Blocked or Not Existing.

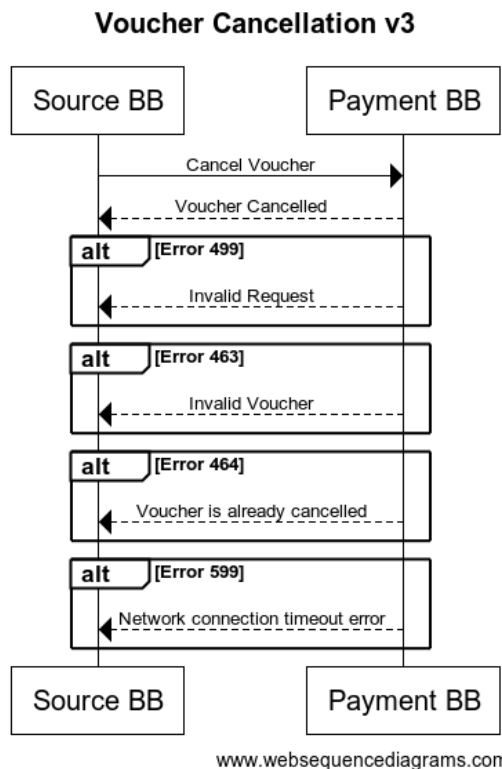
### Voucher Status Check v3



www.websequencediagrams.com

## 9.6.6 VoucherCancellation API

The VoucherCancellation API is used by non-Payment Building Blocks in the GovStack Framework to cancel a voucher. The calling Building Block will capture the voucher number and send it to the Payments Building Block to cancel the voucher. The Payments Building Block will respond with a status of Cancelled, Already Cancelled or Not existing. The VoucherCancellation will override the Blocked status and render the voucher permanently unusable.



### 9.6.6.1 Response Status Codes

Status codes in the 200 range imply that the request completed normally, the 300 range indicate that the request must be present to a different location. Status codes in the 400 or 500 ranges imply that there was an error executing the request.

#### Used Response Codes

Response codes	Description
200 (ok)	Sent when the request completed successfully.
202 (Accepted)	The request has been accepted for processing, but the processing has not been completed (pending).
401 (Unauthorised)	The request needs authentication.
403 (Forbidden)	The request was denied, and will be denied also in the future.
404 (Not Found)	The resource specified in the URI was not found.
500 (Internal Server Error)	The server encountered a general error during execution.

# 10 Workflows

The workflow provides a detailed view of how the payments BB will interact with other building blocks to support common use cases.

## 10.2 G2P Bulk Payment Workflow

### 10.2.1 Bulk Disbursement for Unconditional Cash Transfer

#### 10.2.1.1 Prerequisites

Before bulk payments are made the following are prerequisites:

Funding requirements

- The funding requirements must operate within the budget/ceiling.
- The number of funding accounts and the life cycle processes will vary depending on the payment infrastructure scenarios.

Bulk payments file

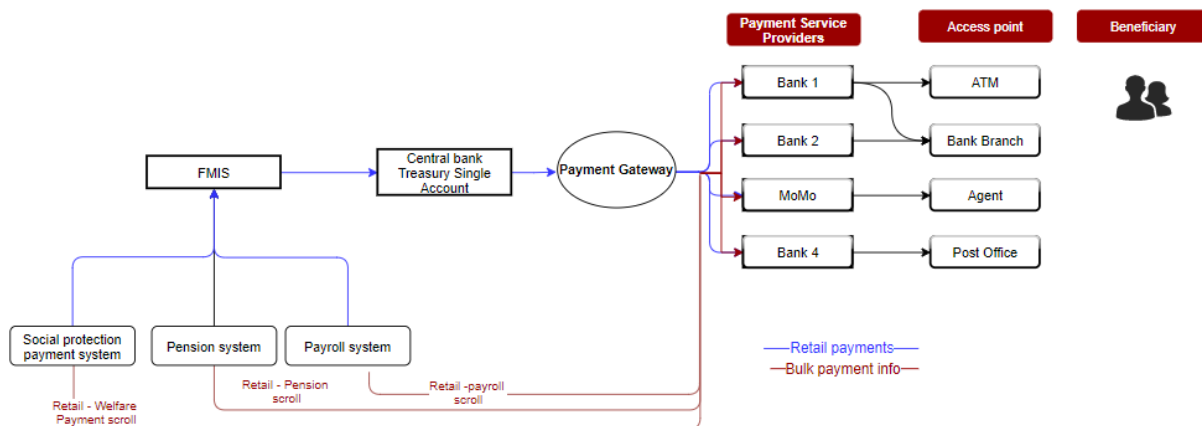
- For the salary payments use case, this would be generated by the payroll system.
- For the unconditional social cash transfer use case, this would be generated by a process that would be triggered as per a pre agreed frequency and the information about the payments to be made would be extracted from the registry (database containing the information about beneficiaries for a particular government social cash transfer programme). The process generating this payments file is outside the payments building block.

#### 10.2.1.2 Description

There are three options for the disbursement,

- Manual process for Govt/Dept to send the retail payment details for each FSP (ie either by email or other means). This would be the case where there is a lack of interoperability among FSPs.
- Upload the batch disbursement file in the payment web portal for each FSP to retrieve in the case of a centralised Account Lookup Directory Service.
- Automate the disbursement process through the decentralised Account Lookup Directory Service and the payment gateway.

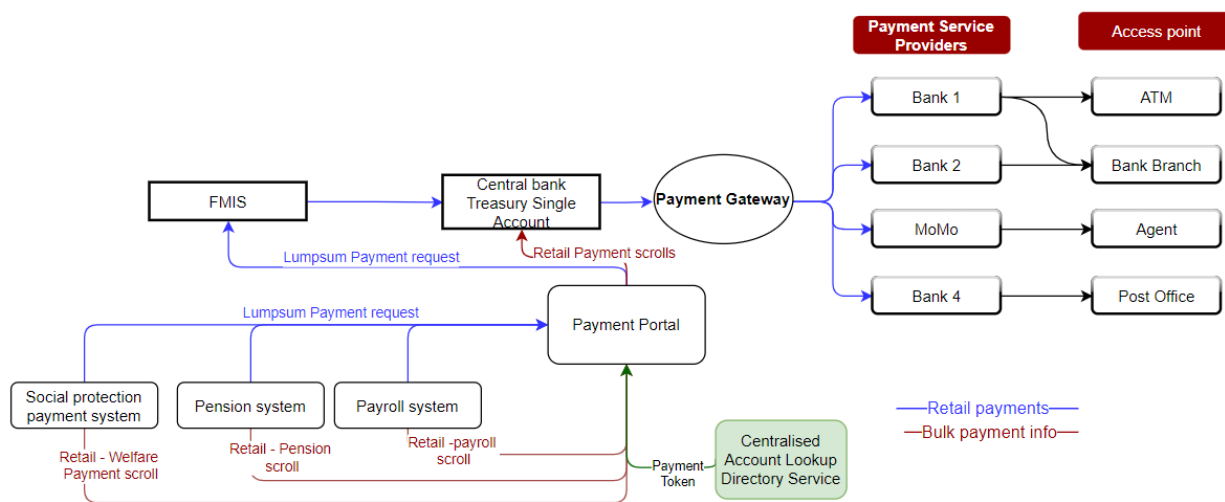
**Option 1:** Retail Payment Information is Sent Securely to Each Payment Service Provider for Disbursement



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

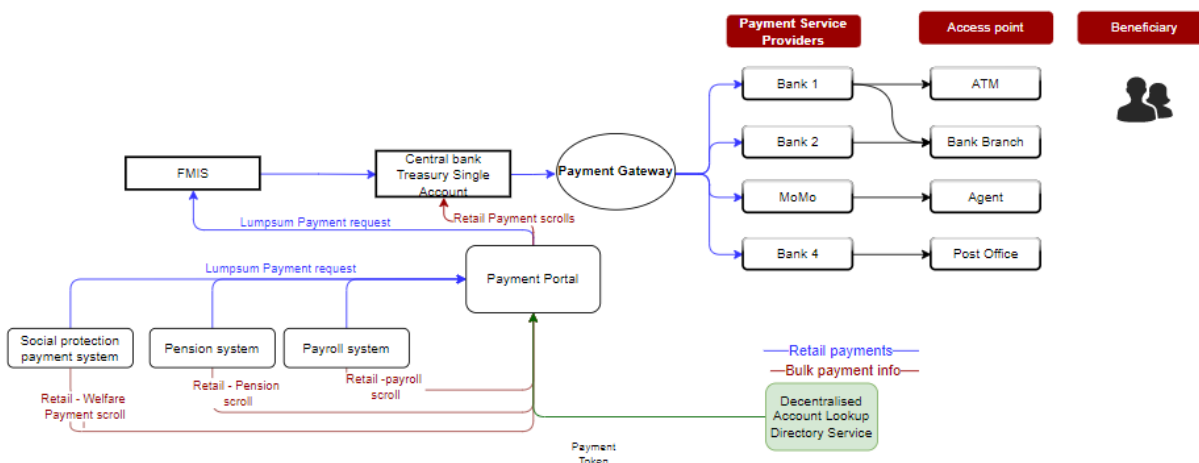
**Option 2:** Retail payments are accessed via a payment web portal by the Payment Service Provider the advice/electronic fund transfer request, or paper-based check is issued from the Financial Management Information System to the TSA-holding Bank.

Based on this advice, the Bank disburses funds from the Trusted Single Account (TSA) into the digital payments system of the FSP which transfers the corresponding funds to the recipient's account. For the retail payment scrolls, where each agency is responsible for running the payment system - payroll, social welfare payments, etc - the payment details are not stored in those systems. Instead, the beneficiary's payment token is retrieved from the Centralised Account Lookup Directory Service and kept in the government payment portal. The payment lists are only shared with the program account holder institution/FSP, PSP, via the government payment portal. The FSP/PSP can log in on the government web portal to access the directory for payments that the FSP needs to effect for each G2P Program.



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

**Option 3:** Similar to option 2 but the disbursement process to the beneficiary is routed automatically through the payment gateway using the payment token retrieved from the decentralised account lookup directory service.



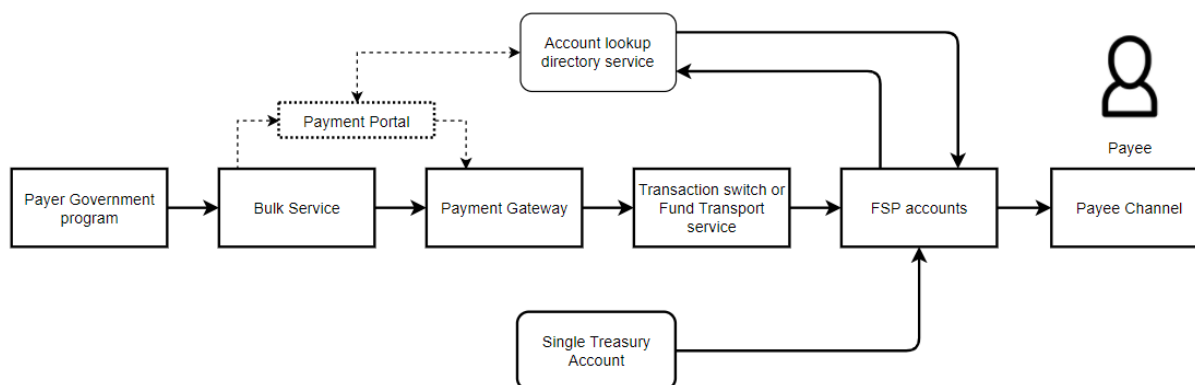
[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

Bulk payments require the functionality of a Payments Gateway and the functionality therein. It also assumes a separate mechanism by which the recipient account address is determined. In the figure below the "account lookup directory service" functionality maps the concept of identity to the payment alias and from there to the FSP (financial service provider) routing address. (i.e. wallet address or account address)

### 10.2.1.3 Interaction with Other Building Blocks

This workflow requires interaction with the registries building Blocks

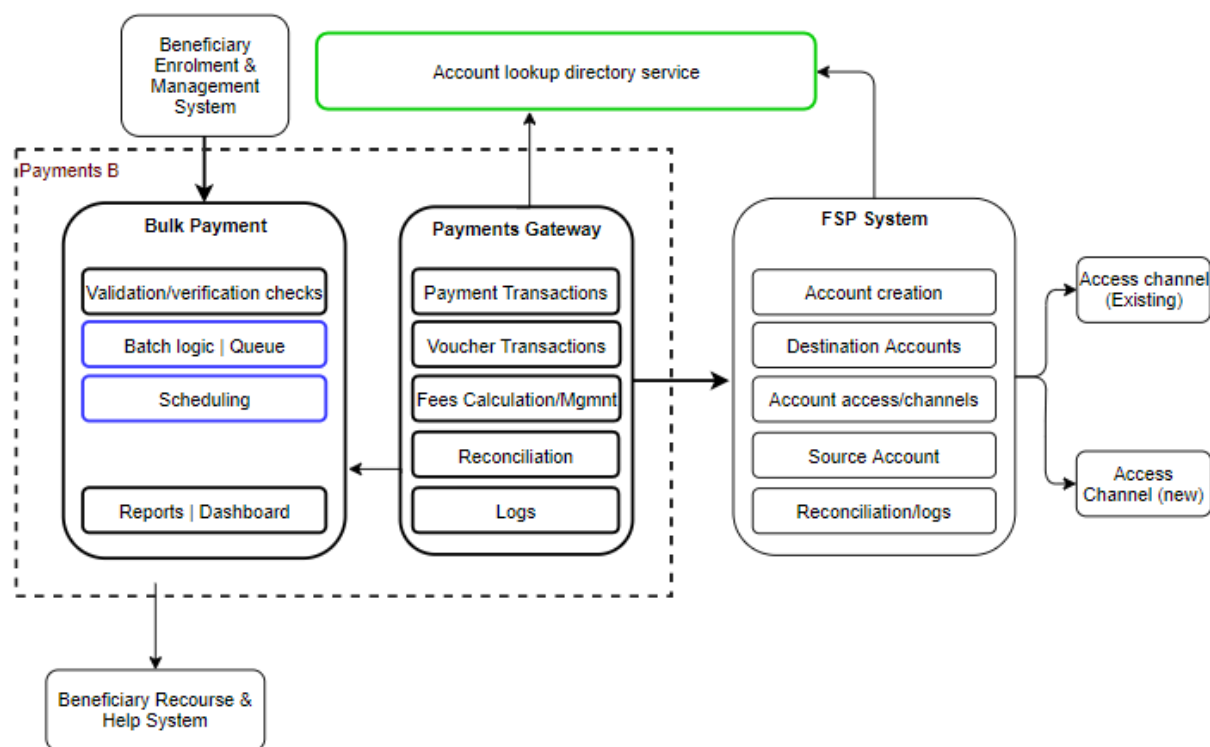
At a high level, the payment components used for bulk payments are shown in the figure below.



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)



Figure: Key digital requirements of Bulk Payments expressed as a high level block diagram. Dotted line for required components.



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.net)

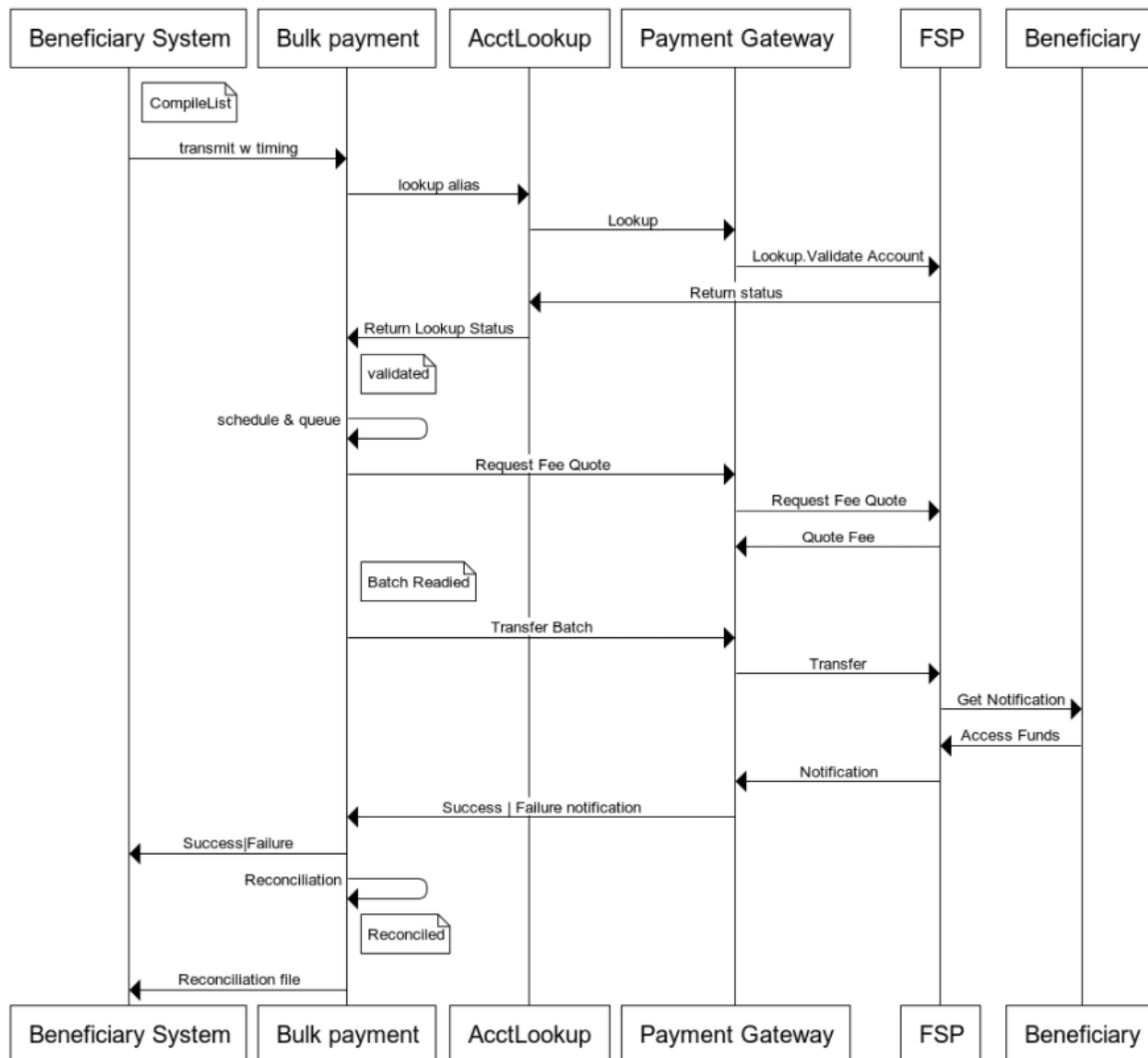
#### 10.2.1.4 Sequence Diagram

The sequence diagram shows the flow of data between building blocks for bulk payments workflow.

- The Beneficiary system (registry BB) transmits the compiled list with boundary conditions for payment timings for each G2P programme (i.e. staggered, each week, standing, etc.).
- Bulk payment services validate data structures and content in compiled lists.
- Bulk payment service uses the Account Lookup (Directory) Service (ALS), a type of Discovery Service that is more protective of account information and privacy. The ALS is used to establish the destination FSP, and the payment alias is then provided by the destination FSP. These functions are used when the account address is not specified in advance. The discovery service can also be used to verify whether the account address information provided by the beneficiary system is valid (this would be in the event the payment information is provided by the beneficiary system to the payments BB directly).
- FSP validates account exists and provides status of account.
- Bulk payment service prepares the batch breakdown on the basis of rulesets governing which FSPs shall receive which payments, combining payments with other payments to the same beneficiary, etc.

- In the case of salary payments, there is a single entry in the payment instruction file sent to the bulk payments service.
- FSP returns a quote on the fees to be charged.
- Bulk payment service sends the batch to each FSP with payment instructions.
- FSP notifies the end beneficiary who then requests the payment via a channel (Merchant POS, Agent, mobile banking, wallet account feature, bank transfer, voucher).
- Beneficiary is paid and the success is communicated back to the Beneficiary system (as well as error codes).
- Bulk payment service system does reconciliation on accounts paid/not paid and communicates that back to the beneficiary system as well.

#### Bulk Payment and Payment Gateway within the bulk payment use cases



<https://www.websequencediagrams.com/#open=768631>

## 10.2.2 Disbursement to Beneficiary Using Mobile Money

In order to facilitate the transfer of funds from the disbursement organisation (the payer) to the mobile money provider, the mobile money provider would need to be connected to the payment gateway / switch. Should this connection not be in place, the disbursement could be facilitated by a third-party aggregator or there would need to be a bilateral connection between the payer's FSP and the Mobile Money Provider.

### 10.2.2.1 Interaction with Other Building Blocks

This workflow requires interaction with the *registry* Building Blocks

The disbursement organisation (payer) gives instruction to its FSP to process a bulk disbursement to a number of mobile money recipients (Healthcare workers).

The payer's FSP forwards the bulk payment instructions to the Bulk payment service which validates the list and queries the ID directory service to determine the recipients mobile money providers. The ID directory returns the list of the recipients providers to the BB which would then create separate batches for each mobile money provider.

The bulk payments service sends the batch and payment instructions to the payer's FSP which processes the transfer through the gateway which forwards the payment to the correct Mobile Money Provider.

The Mobile Money provider would then credit the beneficiaries accounts who would receive a notification, confirming the amount has been credited to their accounts. Upon payment success, the Mobile Money provider would notify the payer's FSP of the payment completion

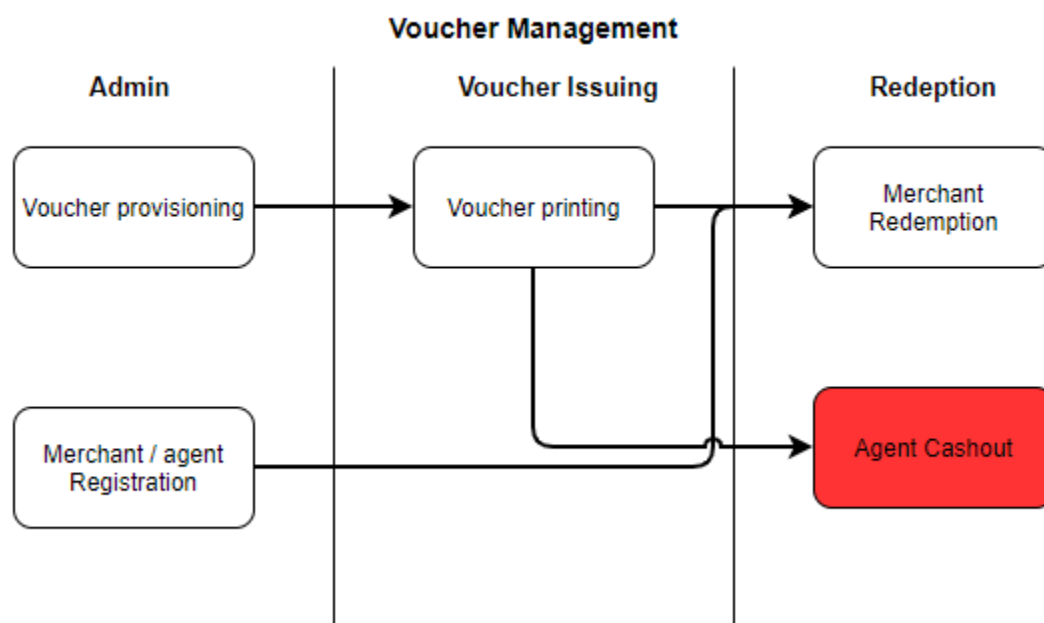
## 10.3 G2P Beneficiary Payments Using Vouchers

### 10.3.1 Description

The Voucher Management System supports at least three workflows for:

- a) administration (voucher provisioning and merchant registration),
- b) voucher issuing (pre-activation and activation) and
- c) redemption use cases as shown in the diagram below. These use cases and the relationship between each one of them is shown and further described below.

The use cases are described in the section below:



[Payments building block diagrams.drawio - diagrams.net](https://diagrams.drawio.net)

### 10.3.1.1 Admin Process

These processes are usually done prior to the issuance of the voucher to ensure a smooth flow at the point of issuance.

The Admin processes for the Voucher Management Server cover the lifecycle of the vouchers and are typically performed by a privileged user (this may be done through User Interface or an API). These processes include but are not limited to creating voucher groups, provisioning vouchers, suspending vouchers, unsuspending vouchers, validating vouchers, key management and purging of used vouchers.

- **Voucher Provisioning**  
This function will be done by an administrator (privileged user). It will typically be triggered by the deposit of funds in a funding account, thus the source account. Voucher provisioning creates a conditional right to funds, and an inventory of issued-vouchers. Other processes related to this are voucher inventory management, voucher suspension and voucher purging.
- **Merchant Registration**  
In order for efficient redemption of vouchers merchants **MUST** be registered in advance to create a network of trusted providers. This registration is assumed to be managed by the Registration BB. The account verification of the merchant can be done at registration or during redemption subject to a configuration

During the registration process, merchants **MAY** also be assigned to different voucher groups depending on the required function that has been implemented. For example, there may be a voucher group for schools. This implies that vouchers of this type can only be redeemed at schools. This also requires that the use flow system at which the voucher is issued is aware of these voucher groups and is able to send the appropriate request to the Voucher Management System..

- **Agent Registration**

In markets where cash outs are being used, it is expected that the Registration BB will register agents in a similar way.

- **Voucher Groups**  
Depending on the requirement it should also be possible to set up multiple voucher groups. Vouchers in the same voucher group will have similar characteristics and are labeled with a specific voucher group name. During voucher provisioning, a voucher can be created and attached to a single voucher group.. When a voucher is requested (using the pre-activation API) it is expected that the voucher group will be one of the parameters set.
- **Voucher Issuing**  
Voucher issuing is triggered by the registrations building block which will determine whether the conditions of issuance have been met. The calling block will determine the denomination and voucher group of the voucher to be issued. The voucher number and the voucher serial number that is issued can be presented to the beneficiary in multiple ways including but not restricted to encoding in the form of QR codes, bar code, printed voucher or even SMS. This is outside the scope of the Payment BB. It is expected that Building Blocks through which the voucher is redeemed will also be able to decode the voucher

### **10.3.1.2 Interaction with Other Building Blocks**

This workflow requires interaction with the Registration and Merchant registry Blocks

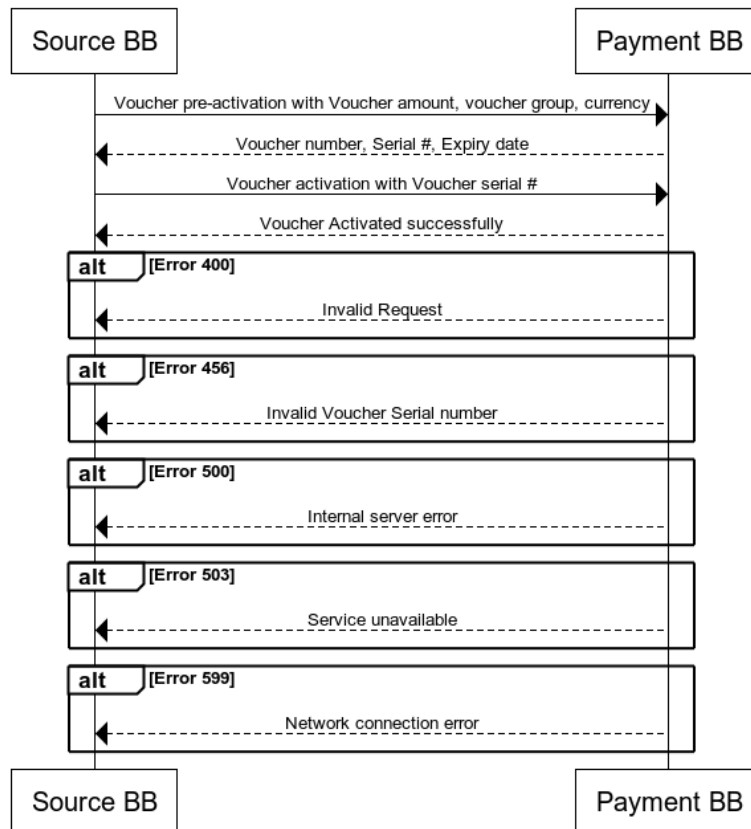
### **10.3.1.3 Sequence Diagram**

The sequence diagram shows the flow of data between building blocks for this workflow

### **Voucher Activation**

The voucher activation flow is shown in the diagram below.

## Voucher Activation



www.websequencediagrams.com

Link to edit: <https://drive.google.com/file/d/1F11u50rrWluGcnOOJjdhQHS-5if00l2B/view?usp=sharing>

### Flow Description:

- An external Building Block may invoke the Payment BB API gateway to pre-activate API on the Voucher Management Server with the amount of the voucher, the voucher currency and the voucher group. The calling block may optionally send a comment. The comment will be stored by the voucher server. The voucher group will indicate that it is looking for a voucher from a specific voucher group. This API call will be made through the payment orchestrator.
- The voucher group is typically used for the conditional social transfer (e.g. for school fee payment). If any voucher can be used for any purpose, then all vouchers should be created with a generic voucher group (e.g. "GENERAL-PURPOSE").
- The API returns to get a voucher number, the voucher serial number and its expiry date. At this point the voucher will be flagged Pre-Activated.
- The calling Building Block may render the voucher as a QR code, as a barcode or even an SMS text. It is recommended that the voucher should include supplementary data of the recipient. It is also recommended that this data should also be printed in human readable form so that the recipient can verify the data on the voucher. This data can also be verified at the point of redemption.

- Once the calling API successfully prints / issues the QR code, the voucher can then be activated using the activation API. It is assumed that there will not be a substantial delay between pre-activation and activation to necessitate the need for multiple expiry periods.

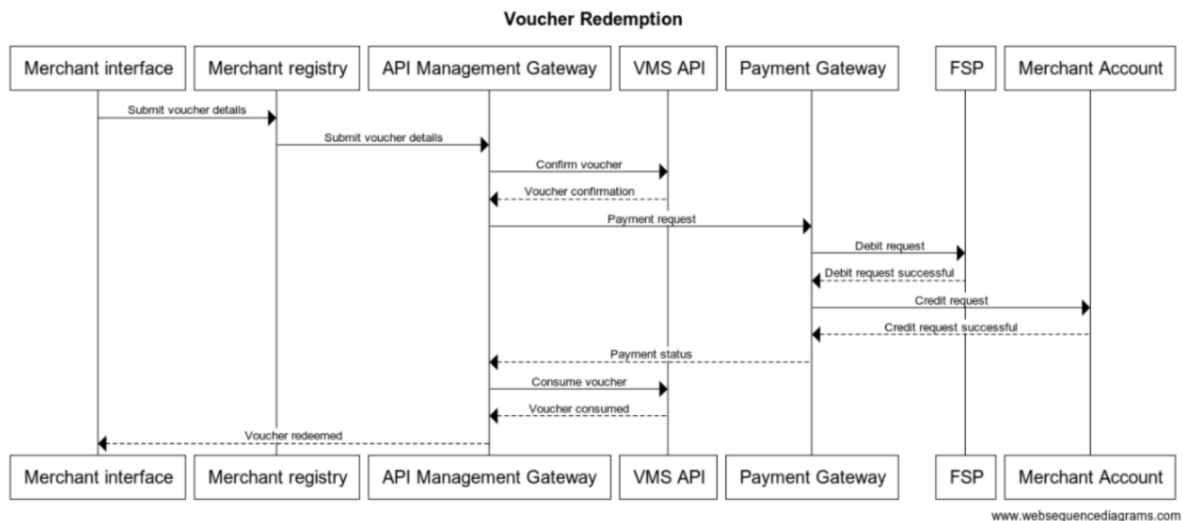
Alternative: the voucher could be activated immediately on being requested. This could be controlled at the Payment Orchestrator.

- Preconditions
  - The calling registration block will determine that all the necessary preconditions are met.
  - The most critical precondition is the prefunding which will be done manually at voucher provisioning stage. (Other options that could be prefunding triggered by activation or at redemption which are not considered because they are considerably more complex and create additional failure points in the activation of the vouchers.)
- Assumptions / Queries
  - It is assumed that only one currency will be used.
  - A zero-trust architecture.
  - The transaction is not reversible.
- Post-Condition
 

The calling BB may invoke another payment BB API e.g. initiating an incentive payment for the agent.

## Voucher Redemption

The voucher redemption is shown in the diagram below.



Link to edit: <https://drive.google.com/file/d/1F11u5orrWluGcnOoJdhQHS-5ifool2B/view?usp=sharing>  
<https://app.diagrams.net/#G1F11u5orrWluGcnOoJdhQHS-5ifool2B>

Flow Description:

- At redemption the beneficiary shows the voucher to the merchant or agent.

- The merchant / agent will scan and interpret the medium in which the voucher is presented (this could be a QR code or a barcode or an SMS or even a printed number).
- Voucher redemption Validation: The details presented MAY allow the merchant / agent to authenticate/validate the owner of the voucher.
- The agent will then initiate the redemption process which will call the Payment BB API Management Gateway.
- The API Management Gateway will validate the voucher and initiate the payment through the payment gateway.
- If specific voucher groups have been set up, voucher usage may be restricted to specific merchants. An override parameter should also be provided that will allow a Voucher of any voucher group to be redeemed at any merchant.
- Once the payment to the merchant / agent is successful the Payment gateway will inform the API Management Gateway.
- The API Management Gateway will then instruct the VMS API to flag the voucher as consumed.

Alternatives:

- Payments could be made through a switch in which case there would be no need for prefunding accounts in each financial institution.

Preconditions:

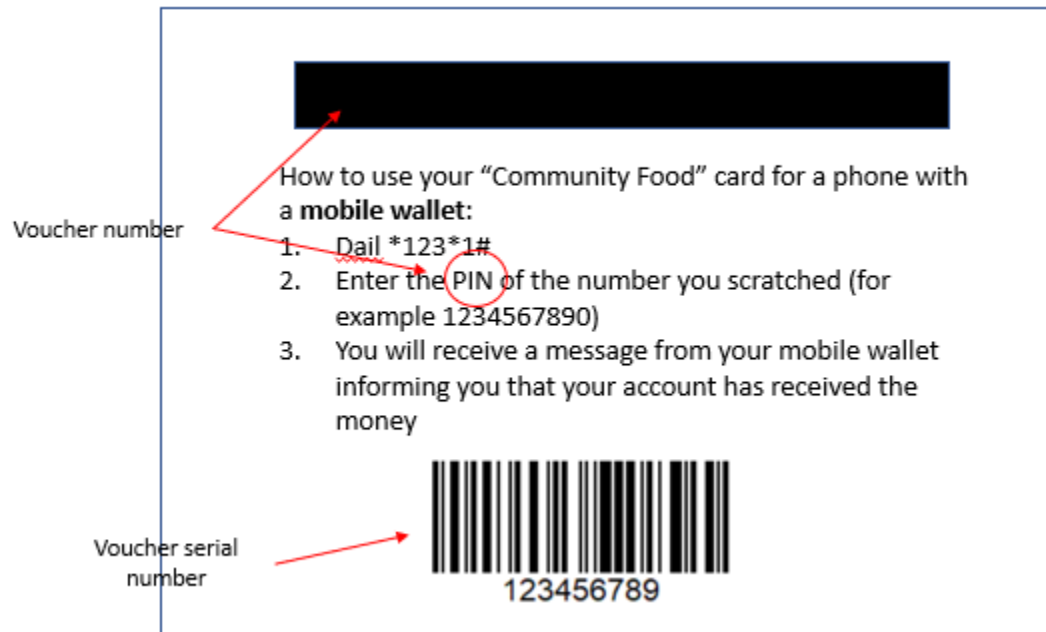
- The merchant must have been registered possibly by the Registration BB with their preferred payment method.
- The merchant must have a mechanism (e.g. a mobile APP, USSD app or SMS option) to verify the beneficiary prior to redemption.

Assumptions:

- The app for reading the QR code and redemption are not part of the scope of the Payment BB.
- The entire voucher amount is consumed. Partial redemption will not be allowed.
- The amount that is paid to the merchant or agent is debited from a prefunded account.



## Sample Redemption



Source: *Kenya Government and WFP Disburse Relief Cash to Drought Stricken Families accessed 05-Sep-21*

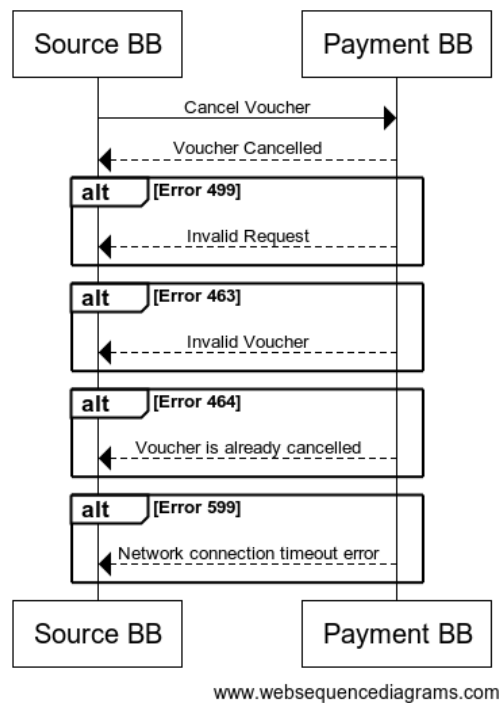
### Note:

In the case of a physical voucher the voucher number or the secret number is hidden behind some material that must be scratched away to see the number. The voucher number is also commonly known as the PIN.

## Voucher Cancellation

The voucher cancellation flow is shown in the diagram below.

### Voucher Cancellation



#### Flow Description:

- The calling Building Block will request the API Management Gateway to cancel a Voucher.
- The API Management Gateway will invoke the VMS API to cancel the Voucher.
- The VMS API interface will check if the Voucher is valid and then cancel the Voucher.
- The VMS API interface will confirm that the Voucher has been canceled.

#### Alternatives:

- If the Voucher does not exist, the VMS API will respond that the Voucher does not exist.
- If the Voucher is already consumed, the VMS API will respond that the Voucher is already consumed.
- If the Voucher is already canceled the VMS API will respond that the Voucher is already canceled.
- If the Voucher is suspended the VMS API will respond that the Voucher is suspended.

## Voucher Technical Requirements

Requirement	Type (Must/Should/May)
<b>Voucher Provisioning</b>	
High volume generation of vouchers	MUST
Voucher numbers must be unique and not predictable	MUST
Logs must not capture voucher numbers	MUST
<b>Voucher Storage (will this be in a separate BB)</b>	
Secure storage	MUST
High Availability	MUST
Issuance	
API invoked to get voucher number and serial number	MUST
<b>Redemption</b>	
API to redeem voucher	MUST
API to invoke payment gateway	MUST

- All calls from external parties (e.g. Registration BB) to the voucher management system will be initiated through the API management gateway.
- The payment orchestration module may direct transitions between the various functions.
- The discovery service could be called by other building blocks to determine where bank accounts / wallets sit.
- The only function that speaks to the DFS is the payment gateway. Any function that needs to speak to the DFS goes through the Payment gateway.
- The Payment gateway may need to speak to a Number Portability provider.
- Audit trails (transaction logs) capture each event as it happens and are to be used for queries, analysis and reconciliation.
- Event logs will capture specific events that happen at each node.

## 10.4 P2G Payments

A P2G payment is a payment made by a person to the government or a government agency in exchange for a service provided.

### 10.4.1 Description

The following P2G payments using mobile applications are considered.

- Government initiated - request to pay
- Mobile payments
  - With QR codes
  - with reference ID (payment transaction ID)
- Banking channels
  - with reference ID

### 10.4.2 Interaction with Other Building Blocks

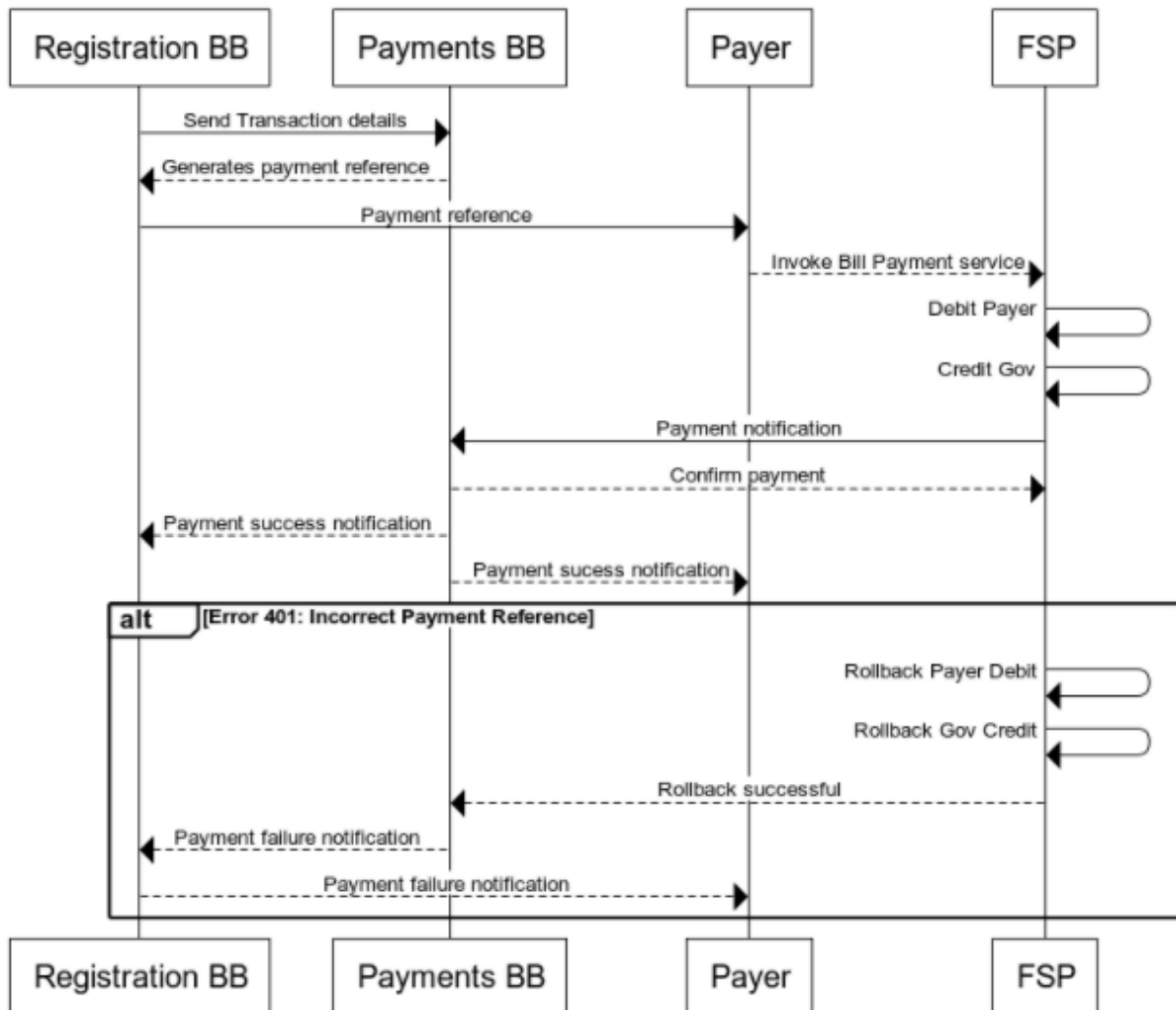
This workflow requires interaction with the messaging BB and the registration building block

### 10.4.3 Sequence Diagram - P2G Payment

General Notes:

- It is assumed that all APIs used are real (sync or async).

## P2G Payment



[Link to Edit Diagram](#)

### Flow Description:

- Upon registration for a government service, the registration building block sends transaction details to the payment building block which creates and returns a unique payment reference for the payer of the service.
- The registration building block sends the unique reference number to the payer.
- After entering their account the payer would invoke a bill payment to a selected service (in this case paying for the registration service). The payer would need to enter the payment reference which would prompt the retrieval of the payment details from the registration building block in real time. If the payment details are correct, the payer is prompted to enter his / her mobile

money pin to authorize the payment.

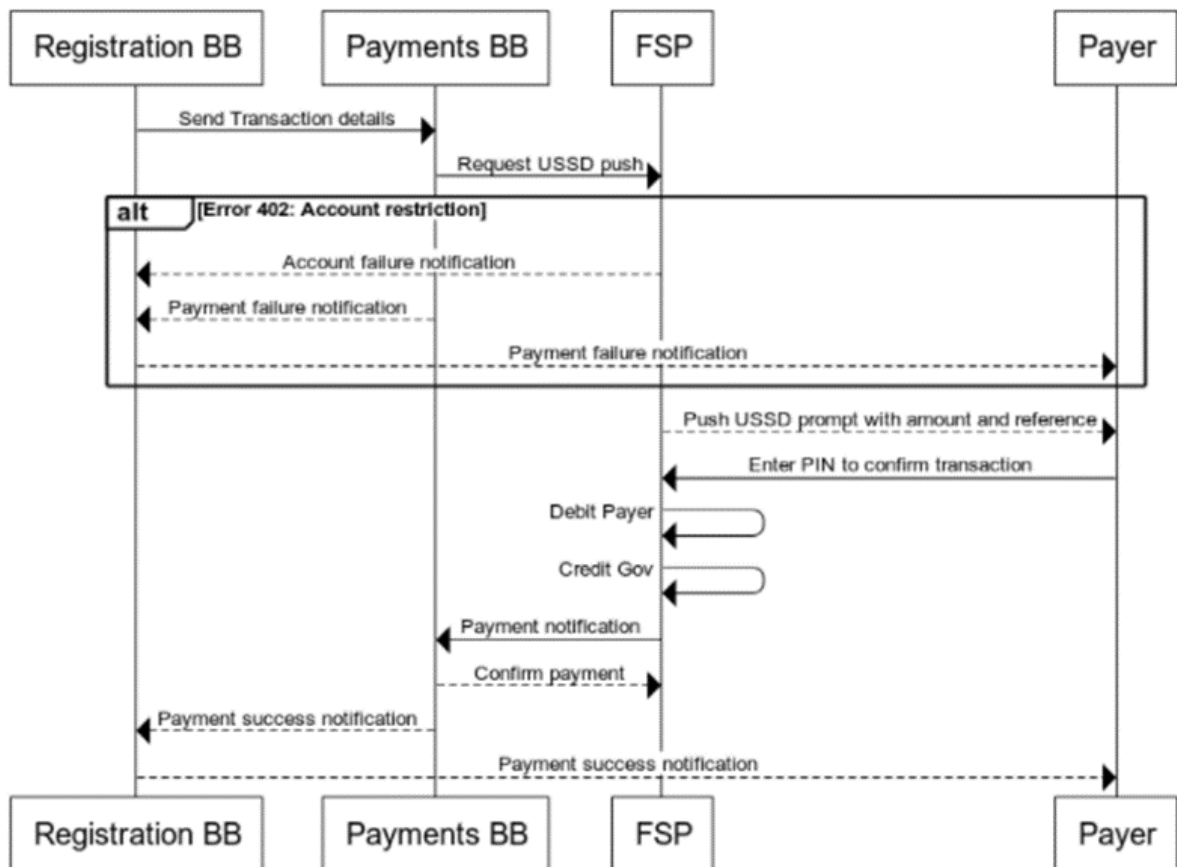
- Upon authorisation, the payer's account is debited and the government's connected account is credited in real time. The FSP, in this case the mobile money provider would then notify the success status to the payment building block.
- For this type of payment, as the payment reference is needed to validate the payment, this can be done by any mobile money account holder.

Notes:

The above model requires that the payer must provide two pieces of information through their USSD, STK or mobile app:

- A merchant ID. Typically the government would be given a special merchant ID. The payer should be able to select which government service he/she would be paying through the mobile money interface.
- A reference ID: this is unique and time bound for each transaction.
- As the bill payment is invoked by inputting the reference number which prompts the retrieval of the payments details in real time from the registration building block, a failed transaction could be triggered by a session time-out or a wrong PIN. In both cases the payer would have to re-initiate the transaction.
- In the P2G payment in the flow above, the government holds an account with the FSP which would collect the payments on the government behalf and transfer it to the single treasury account on a defined timeline (i.e. daily) in an aggregated way. For reconciliation purposes, the registration BB would need to notify the government of a successful / unsuccessful payment.

### 10.4.4 Sequence Diagram - P2G FSP Payment by USSD Prompt



www.websequencediagrams.com

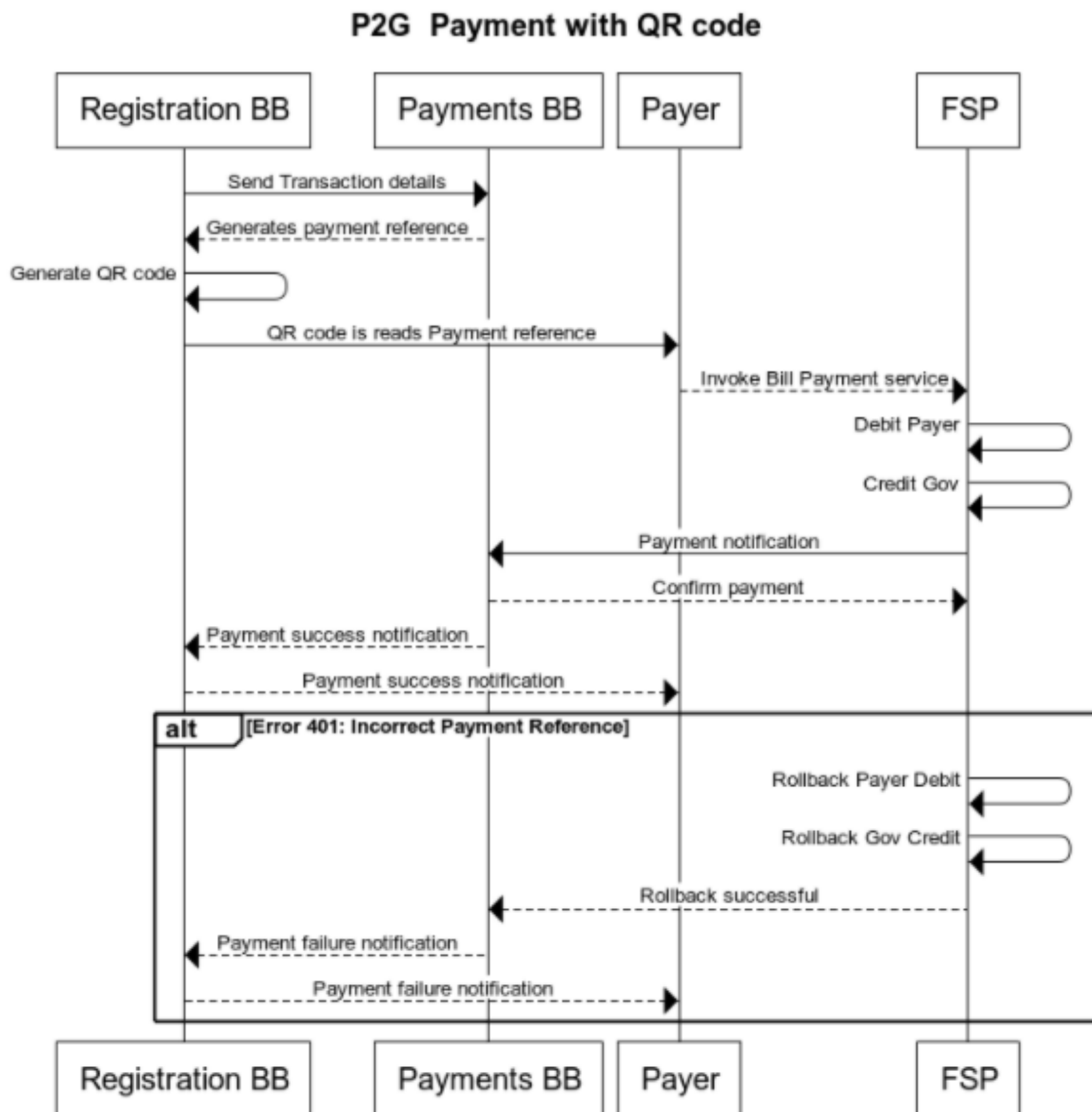
#### Flow Description:

- Upon registration for a government service, the registration building sends transaction details to the payment building block which sends a payment request to the payer through his / hers mobile money provider (USSD prompt).
- The payer will see a request coming for the specific service requested with payment details, prompting him / her to authorize the payment by entering the pin code.
- Upon authorisation, the payer's account is debited and the government's connected account is credited in real time.
- The FSP, in this case the mobile money provider would then notify the success status to the payment building block.
- This payment differs from the previous P2G mobile money payment as the payment is not initiated by the payer by invoking a bill payment but is initiated by a merchant (in this case the government agency providing the service). Therefore the payer is requested to pay for the service immediately.

Notes:

- There can be no error in the above as both the merchant number and the payment reference are pushed to the phone.
- The only risk here is a timeout on the USSD or the user keying in the wrong PIN which may require the transaction to be reinitiated.

### 10.4.5 Sequence Diagram - P2G Payment with QR Code





Notes:

- There is small room for error in reading a QR code (unless the payer scans the wrong code).
- All mobile operator and banking apps need to be able to read the reference in the same way for this to work meaning that a standardised / interoperable QR code needs to be in place at a country level. While this is in place in some Asian countries (i.e. Indonesia, Sri Lanka) where there is a widespread adoption of QR codes, In Africa the uptake of QR codes is significantly lower and standardisation is typically not in place at a country level.

#### **10.4.5.1 QR Code Payment Flow Use Case Example**

1. During registration, the registration BB will generate transaction details including amount to be paid by the payer and transaction ID and send to the payments building block.
2. The payments building block will use the transaction details to initiate a request to pay to the FSP.
3. The QR payment widget displayed to the payer will have the following different attributes
  - a. payment\_entity\_id
  - b. amount
  - c. currency
  - d. transaction\_id
4. The payer scans the QR code to approve/reject the payment.
5. The FSP sends a notification of the status of the transaction to the payments BB and the payer.
6. Transaction status sent to the registration building block on completion of payments.
7. The messaging building block sends a transaction confirmation message to the payer.



# 11 Other Resources

## 11.1 Standards

The following standards are applicable to data structures in the Payments Building Block:

- All dates should follow ISO 8601.
- The transaction data is passed in a format that is compatible with ISO 20022.
- The voucher number generation should be aligned to 18031:2011 or NIST SP 800A (Recommendation for Random Number Generation Using Deterministic Random Bit Generators), B (Recommendation for Entropy Sources used for Random Bit Generation) and C (Recommendation for Random Bit Generator (RBG) Constructions).
- Merchant IDs must conform to the Universally unique identifier (UUID).
- A Merchant Presented Mode dynamic QR code would be recommended (EMV QRCPS v1.1 2020).

## 11.2 GovStack Resources

- <https://discourse.govstack.global/t/about-the-payments-category/62>
- [Information Mediator Building Block Definition](#)
- [Registration Building Block Definition](#)
- [Architecture Blueprint and non-functional requirements](#)
- [Security Building Block Definition](#)
-  Elevator pitches and visual aids - Architecture working group
- [Digital Registries Building Block Definition](#)
-  BB - Existing Database Connector

## 11.3 Unconditional Social Cash Transfer Resources

- [UC-P-USCT-001: Payment - Unconditional Social Cash Transfer \(bank payments\)](#)
- [UC-P-USCT-002: Payment - Unconditional Social Cash Transfer \(non-electronic/cash payments\)](#)
- [UC-P-USCT-003: Payment - Unconditional Social Cash Transfer \(direct payment based on family relationship\)](#)
- [UC-P-USCT-004: Payment - Unconditional Social Cash Transfer \(Mobile Money payments\)](#)

## 12 Key Decision Log

- "Discovery service" also known as the "Account lookup service" in the payments BB components.
- Direct bilateral connections between the state owned bank and the Financial services provider vs all connections going through a switch.
- The Debit and credit entries for the treasury single account are out of scope of this document
- The user interface for payments management will be managed through the payments portal.
- Payments, reconciliations and settlements out of scope of the building block.
- Sequence flow diagrams to the FSP were removed -
  - G2P payments using Mobile Money - individual Disbursement Flow were removed - 24/Nov/2021
- 01/02/2022 All security requirements removed from the specification of the payments BB, security requirements are described in the security BB.
- 01/02/2022 A requirement to have a hardware security module was removed, data security is handled in the security BB
- 07/02/2022 Account mapper and the account lookup service to be discussed in detail in phase 2 along with considerations for use of federated IDs with considerations for user privacy.

Comments and corresponding replies from the payments BB are shown in the table below

No.	Comments/Feedback	Suggested Action/Reason
1	Payments BB has core requirements called out. However, the Voucher mgmt system can be made more generalised.	One powerful feature that facilitates generic characteristics in the VMS in the specification is the use of voucher groups. These groups can restrict the use of vouchers to certain functions, geographies and merchants. One extension we could do is to add a custom field to the Voucher Management System API so that the calling Building Block could submit data regarding how the voucher is to be used and by whom. However, as per the use cases the voucher initiation and redemption are triggered by other Building Blocks so this data is actually collected by the calling Building Block. In terms of the final delivery of the voucher this is in control of the calling Building block

2	Payments from other BB perspective (and from citizen perspective) is to integrate to the payment service to avail a govt service	Unclear comment - payments is handling all government related payments and disbursements.
3	It may be a good idea to separate Payment BB into two logical parts - a) as payment integrator b) as core building block to support payments at country scale.	The payments building block is focusing on enabling the digitization of the different types of payment modes for government services.
4	Generally, we have a paying side (Payer) and a receiving side (Payee)? Why diversification (P2G, B2G, etc.)?	The payment processes are different for each of the payment types B2G and G2B, P2G etc.
5	Section 3. What if the user of the service wants or for some reason paid physically / in the traditional way and not electronically?	In a G2P, the beneficiary "user" can be paid with a physical voucher if they don't want to receive electronic payments. A voucher can be cashed out.
6	Section 3. What exactly does that mean? Is it validation against the budget of the organization?  In public administration it is better to use terms like a budget or allotment or something like that (see IMF GFSM, for example). The term "operating cost" might be confusing	Yes, it means validated against the budget before payment is made.  correction in section 3 to allotment instead of budget.
7	Generally we have a paying side and a receiving side? Why diversification (P2G, B2G, etc)?	The payment processes are different for each of the payment types B2G and G2B, P2G etc.
8	Should include also G2G as governmental institutions are paying taxes as well	These will probably be considered in the next phase of the payments BB G2G are out of scope in this version
9	3.1 Introduction to financial inclusion, banks and mobile money accounts:  Does it have to be in spec? I suggest specification contains only normative information	
10	Should include G2G as well	G2G will be addressed in the second phase
11	Our sense is that this BB is focussed on developing countries - more developed countries will have Direct banking arrangements e.g, taxation authorities, social welfare systems. It is likely we will adopt a very light approach for more incidental digital financial transactions	To clarify more on the above please note we are not focusing on only emerging economies. We are have considered different levels of maturity from developing and developed countries i.e from paper vouchers to central bank

12	what does this mean for PBB.. it may not be advisable to share the gender information for payment info based on minimalism principle. The payments building block should ideally be blind to this.	PBB - payments building block, this was corrected.
13	which information and how would the payments BB know the requirements of different services which use this shared payments BB? this requirement can be made more specific maybe?	
14	Section 5.2 what does this entail? interaction with the payee to verify account details? could there be privacy or security concerns? What does this entail? interaction with the payee to verify account details? could there be privacy or security concerns?	There is no interaction with the payee involved in this, the Account Lookup Services (ALS) in the payments BB will be called to verify the payment information provided by the payee before the payment is made.
15	Section 5.4: any requirements on accounting or reporting of vouchers for different processes listed (provisioning, issuance etc.) .. reconciliation of funds based on redemption? What about the authentication of the beneficiary at the time of activation/redemption ?	These are all good and valid questions. 1) In terms of the reporting, this is expected to be a standard part of the voucher management system which would be capable of showing the vouchers in their different states as well as the aggregate quantity "stock". Such reports would trigger, either automatically or manually, requests for "re-stocking" of vouchers. As of this time my understanding is that the detailed reporting requirements are out of the scope of this document. 2) Reconciliation of funds and vouchers is a key part of the ecosystem. However, the funding of accounts and the management of funds is, to my understanding, out of the scope of this document. 3) The authentication of the beneficiary at the point of redemption, as per my understanding, will lie with the calling block which would likely check the Registry building block to authenticate the user. Other flows could involve the voucher management server storing the beneficiary details but this would appear to go against the principle of avoiding duplication. This may also need consideration of consent in cases where delegation applies.

16	7.1.2 Payment orchestration in the second wave of BB-s there is a Workflow BB. Probably this part can be delegated to the Workflow BB	The payments BB orchestrator handles the microservices interactions within the BB as there are several components for bulk processing, voucher management, Account lookup service.  This is an internal orchestration of workflow in the payments BB. it might embed an appropriate workflow component to realise this orchestration
17	Voucher management:  I did not see support for redemption in the Registration BB. There is only a voucher issuing process. How exactly redemption will work? who will withdraw actual funds from the Treasury account and when?	where the authentication of the beneficiary is required, the merchant authenticates the beneficiary through the registration bb.
18	Account lookup directory service  What is the interface for the users or payment providers to update this mapper e.g. update their account number /bank name/FSP/or even personal identification number ? Who all can access this ?	The user cannot update the mapper directly. When the user makes changes to their account, (e.g. switches from one FSP to another FSP), it is the FSP who will update the mapper.
19	In case of payment of taxes P2G and B2G there is a need to reconcile individual ID/company ID and tax type or even document (in case of Customs clearing). How such reconciliation is supported?	Out of scope of the BB
20	if this would be a request to collect a tax due then I do not see a data field for automatic reconciliation of this payment with outstanding liability in the tax administration accounting system.	Tax administration and accounting are out of the current scope for the payments BB.
21	what kind reference ID it is? issued by whom?	This payment transaction ID issued by the registry
22	in case tax and customs duties payment, I guess, there are no Registration BB but some other BB or how?	This will be considered in future scope

## 13 Future Considerations

- G2G payments
- B2G, G2B payments
- APIs for Payment receipting and payment status