WHITE PAPER ON INTEGRATING G2PCONNECT & GOVSTACK SPECIFICATION PLATFORMS

Abstract

This document explores the harmonization of specifications between G2PConnect and GovStack, focusing on integrating the Scheme Management/Social Benefit Program Management (aligned with G2P-Connect Specifications) with the Payment Building Block from GovStack. This harmonization aims to facilitate seamless interoperability for processing payments and conducting disbursements across different system architectures. By bridging these platforms, the initiative seeks to enhance the efficiency and effectiveness of service delivery in social benefit programs and financial transactions.

Drafted by: MIFOS

Version 1.0
05/01/2024

Document Purpose

The primary objective of this white paper is to articulate the integration strategy and practical steps necessary for harmonizing the G2PConnect and GovStack specifications. By focusing on the integration of G2PConnect's Scheme Management with GovStack's Payment Building Block, this document aims to provide a clear, actionable framework for achieving seamless interoperability and enhanced efficiency in government-to-person payment processing and social benefit program management. Targeting technical and functional bodies, implementation partners, and entities planning to adopt these specifications, this white paper serves as a comprehensive guide and strategic roadmap for robust and user-centric integration.

Executive Summary

This white paper provides an in-depth analysis and strategic framework for harmonizing the G2PConnect and GovStack specifications, focusing on integrating the Scheme Management/Social Benefit Program Management of G2PConnect with GovStack's Payment Building Block. The document is structured into several key sections, each addressing different aspects of this harmonization:

- **Introduction:** This section outlines the broader context of government-to-person (G2P) payments, the roles of G2PConnect and GovStack, and the importance of their harmonization for enhancing service delivery in social benefit programs and financial transactions.
- **G2P Connect Landscape:** It examines the ecosystem of G2P Connect, detailing its components like Scheme Management, Payment and Settlement systems, and how they contribute to unified service delivery.
- **Objectives of G2P Connect:** This part focuses on the specific aims of G2P Connect, highlighting its commitment to seamless integration and standardization across various solutions in G2P payment processing.
- **G2P Connect Interface Specifications:** A comprehensive look at the specifications of G2P Connect is provided, emphasizing its robust framework for digital identity and financial services.
- **GovStack Landscape:** This section delves into the GovStack framework, illustrating its approach to creating a common platform for various e-government services.
- **The Payments Building Block:** An exploration of the core modules of the GovStack Specification Payment Building Block and its pivotal role in processing G2P payments.
- **Harmonization Approach:** The document details the strategic approach for integrating the two platforms, focusing on principles like adaptive integration, data retention, and handling information gaps.
- **Implementation Considerations for the Adapter:** It discusses the technicalities of implementing an adapter pattern for bulk payments, ensuring seamless communication between the two systems.

- **Risk Assessment and Mitigation Strategies:** This section addresses potential risks involved in the harmonization process and proposes strategies to mitigate them.

This white paper not only presents a clear pathway for integrating G2PConnect and GovStack but also underscores the benefits and challenges of this endeavor. It serves as a comprehensive guide for stakeholders, ensuring an understanding of the processes and objectives involved in achieving a cohesive and efficient system for social benefit program management and financial transactions.

Table of Contents

Introduction

G2P is broadly defined as the class of payments made by government entities (such as national or local governments) to individual citizens. These payments can take many forms, including social welfare benefits (such as pensions, unemployment benefits, or subsidies), compensation for public sector employment, or other forms of assistance. The goal of G2P payments is typically to provide support to individuals who need it and to stimulate economic activity in each region or country. The process of making these payments can be complex and requires effective systems for identification, payment processing, and fund disbursement.

[1]G2PConnect is an open-source effort to enable government-to-person digital payments built through interoperable standards and design blueprints. G2P Connect enables policy makers across various departments to 'talk' to each other without revealing any sensitive information. This means that:

- Every department is allowed to own its own information. They simply standardize the information using a common identifier for each individual and link their database through APIs.
- Other departments that require information to plan regarding a transfer can specify their eligibility criteria (which can freely change with the times) as they do currently.
- All systems are standardized to share telemetry information for policy makers to observe key performance metrics and to make informed decisions.

G2P Connect principles are meant to ensure that autonomy, freedom of choice and change, and simplification of the process is ensured at each step of the government to person payment process.

[1] https://g2pconnect.cdpi.dev/g2p-connect/readme

G2P Connect Landscape

The G2P Connect ecosystem is designed to unify and streamline the delivery of government services to individuals. It highlights the interconnected nature of various components such as Scheme Management, Payment and Settlement, and Banking systems, which work together to ensure efficient service provision. The ecosystem comprises of 11 individual components.

1. Foundational Digital ID System
2. Trusted Data Sharing & Digital Credentialing Infrastructure
3. Civil & Other Federated Registries
4. ID-Account Mapper
5. Social Program & Beneficiary Management
6. Payment & Settlement Switch
7. Bank/Mobile-wallet System
8. Last Mile Cash-In/Cash-Out System
9. Unified View for Policy Makers
10. Unified View for Beneficiaries
11. Banking/Wallet Interface Systems

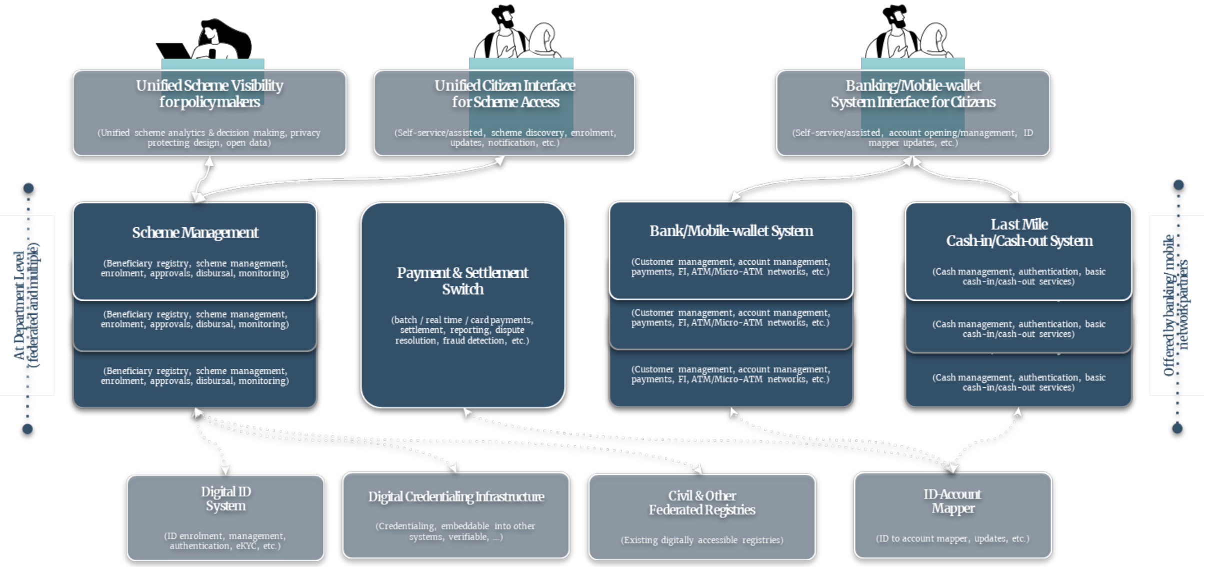

Unified Scheme Visibility for policymakers
(Unified scheme analytics & decision making, privacy protecting design, open data)
Unified Citizen Interface for Scheme Access
(Self-service/assisted, scheme discovery, enrolment, updates, notification, etc.)
Banking/Mobile-wallet System Interface for Citizens
(Self-service/assisted, account opening/management, ID mapper updates, etc.)
At Department Level (federated and multiple)
Scheme Management
(Beneficiary registry, scheme management, enrolment, approvals, disbursal, monitoring)
(Beneficiary registry, scheme management, enrolment, approvals, disbursal, monitoring)
(Beneficiary registry, scheme management, enrolment, approvals, disbursal, monitoring)
Payment & Settlement Switch
(batch / real time / card payments, settlement, reporting, dispute resolution, fraud detection, etc.)
Bank/Mobile-wallet System
(Customer management, account management, payments, FI, ATM/Micro-ATM networks, etc.)
(Customer management, account management, payments, FI, ATM/Micro-ATM networks, etc.)
(Customer management, account management, payments, FI, ATM/Micro-ATM networks, etc.)
Last Mile Cash-in/Cash-out System
(Cash management, authentication, basic cash-in/cash-out services)
(Cash management, authentication, basic cash-in/cash-out services)
(Cash management, authentication, basic cash-in/cash-out services)
Offered by banking/ mobile network partners
Digital ID System
(ID enrolment, management, authentication, eKYC, etc.)
Digital Credentialing Infrastructure
(Credentialing, embeddable into other systems, verifiable, ...)
Civil & Other Federated Registries
(Existing digitally accessible registries)
ID-Account Mapper
(ID to account mapper, updates, etc.)

Figure 1: Components of G2P Connect Solution Blueprint

The architecture supports a unified scheme visibility for policymakers, offering analytics and decision-making tools while protecting privacy. Citizens are provided with a consolidated interface for scheme access and wallet system interactions. Essential support systems like Digital ID, trusted data sharing, civil registries, and account mapping underpin the framework, ensuring reliable and secure operations. This cohesive structure is aimed at enhancing user experience for both policymakers and citizens, optimizing the management and disbursement of social benefits.

Objectives Of G2P Connect[2]

The following are key objectives of G2P Connect Specifications:

1. Focus on G2P Connect solution blueprint - enable seamless integration between solutions relevant to end to end G2P payment scope keeping policy maker and beneficiary at the center.
2. Flexible to accommodate existing standards where applicable, e.g., OAuth2, OpenID Connect, W3C Verifiable Credentialing.
3. Standardize message envelope to support **harmonized** integrations across various solutions with key features:
    - Transport layer agnostic communication
    - Async based processing with retry/polling support.
    - Offline processing capability enabling store & forward.
    - Batch processing
    - Support plug n play for other payloads e.g. OpenID / country specific custom data or verifiable credentials
    - Support for digitally signed and encrypted payloads
4. Enable integration between existing proprietary, DPI/DPGs or country specific custom in-house solutions.
5. Focus is on standardizing core interfaces and NOT on implementation.
6. Each interface acts as a connector between solutions and allows country implementations to realize various use cases.

[2] https://g2pconnect.cdpi.dev/protocol/overview

G2P Connect Interface Specifications [3]

G2P Connect specifications offer a robust framework for digital identity and financial services, integrating various standards and protocols. The Identity component utilizes civil and federated registries with access via HTML and YAML, aligning with standards such as W3C VC Data Model. Verifiable Credentials are issued through OIDC and W3C VC-API, with wallets conforming to W3C Universal Wallet and Open Wallet Foundation guidelines. Beneficiary Management automates the registration, assessment, and enrollment processes. Program Management dictates eligibility and disbursement rules, while Financial Service Providers ensure fund liquidity, secure holds/releases, and voucher management. Furthermore, the system supports last-mile cash-in/cash-out activities, agent, and device management, and incorporates a comprehensive security model with strong authorization protocols. This detailed structure ensures a secure, efficient, and scalable G2P service ecosystem.

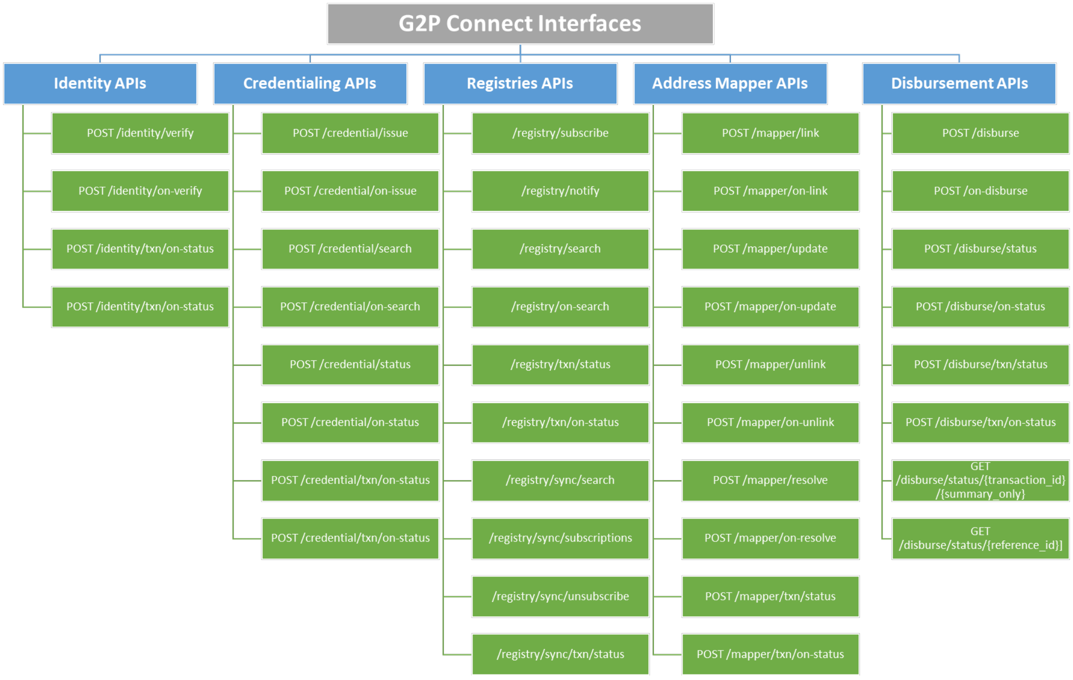


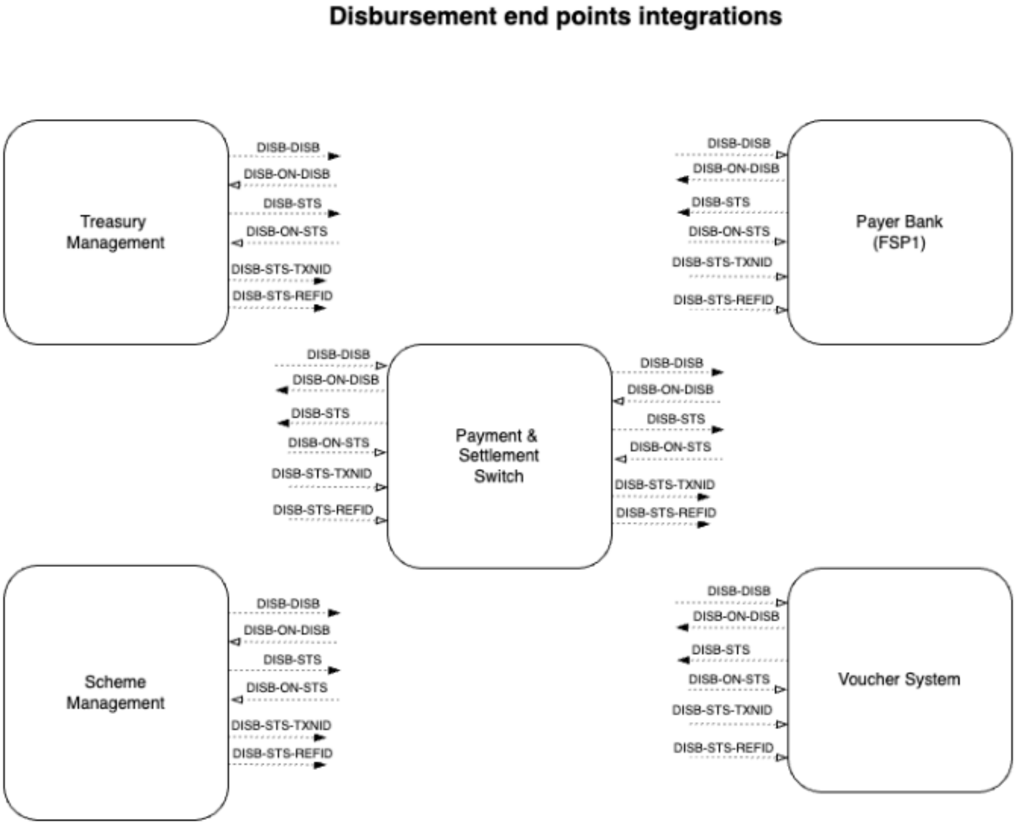
Figure 2 – Logical Representation of G2P Connect Specifications

This is an exhaustive list of all interfaces part of G2P Connect specifications as of January 2024, though these specifications are subject to change from time to time or addition/reduction in interfaces is possible, it gives a broad overview of the range of services involved in either executing or supporting a G2P payment.

Some of the points listed above are also used in the Bulk Disbursement process. The integration scheme shown in the diagram on the proceeding page highlights interaction between various platforms that come together to achieve bulk disbursement end to end. Notably, in a G2P Connect Setting, the following 5 entities are part of the payments process:

[3] https://g2p-connect.github.io/specs/release/index.html

1. **Treasury Management:** This entity/platform encapsulates the budgeting and allocation processes, in a real-world application this is where the Treasury Single Account would lie, or this could also be performed by a Commercial Financial Service Provider, they key requirement is that the disbursing agency has funds held with this function. Disbursal interfaces may be invoked from two distinct services, and the Treasury Management function is one of them.
2. **Scheme Management:** This entity/platform encapsulates the beneficiary and social welfare program management processes. Usually, this is the entity/platform responsible for collecting, maintaining, and providing eligibility and beneficiary data. This is the other function from where disbursal interfaces may be invoked.
3. **Payment & Settlement Switch:** This is the platform/entity which provides the interoperability layer to all Financial Service Providers.
4. **Payer Bank (FSP):** This is the entity integrated to the Payment and Settlement Switch or to Treasury Management/Scheme Management directly for the purpose of taking instructions.
5. **Voucher System:** This platform and its APIs don't currently exist and are hypothetical, but this component is to cater to the requirement where vouchers may be issued.



- DISB-DISB : /disburse/
- DISB-ON-DISB. : /on-disburse
- DISB-STS. : /disburse/status
- DISB-ON-STS. : /disburse/on-status
- DISB-STS-TXNID : /disburse/status/{transaction_id}/ {summary_only}
- DISB-STS-REFID : /disburse/status/{reference_id}

Figure 3 –G2P Connect Interface Schematic

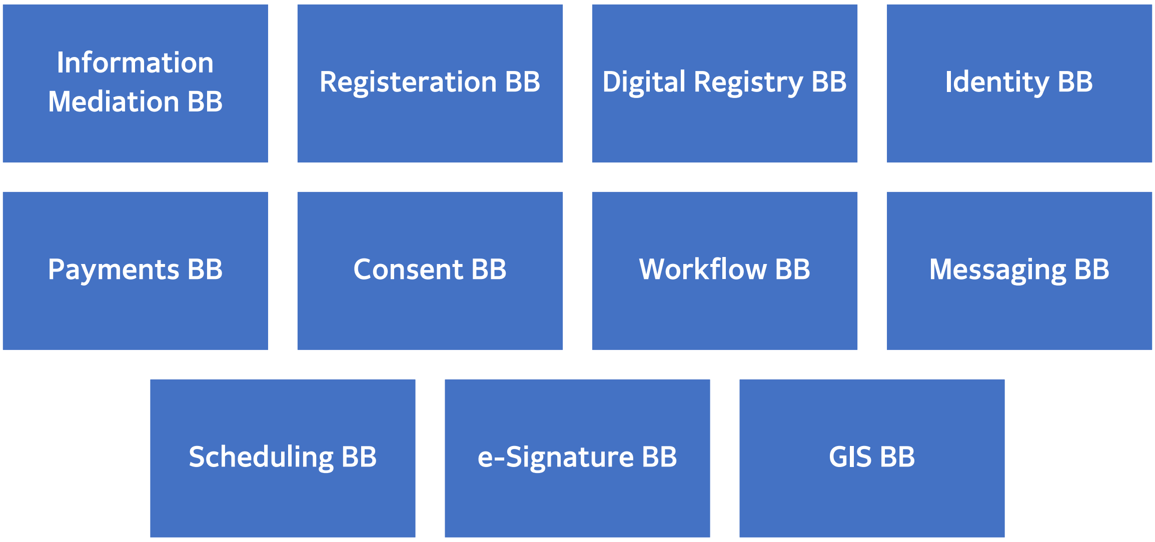
GovStack Landscape

There is an existing logical approach practiced by leading e-gov to create a common shared platform to deliver various government services digitally. A holistic (Whole-of-Government) digital platform that can be used by any government agency across sectors to build new e-government services without the need to redesign, test, and operate the underlying systems and infrastructure themselves every time. Instead of creating unique and disparate solutions, use a common reusable stack of Building Blocks to form the core platform engine and contextualize various e-government services on top.

Building blocks[4] (BBs) are software modules that can be deployed and combined in a standardized manner. Each building block can work independently, but they can be combined to do much more.

Building blocks are composable, interoperable software modules that can be used across a variety of use cases. They are standards-based, preferably open-source, and designed for scale.

Each building block exposes a set of services in the form of REST APIs that can be consumed by other building blocks or applications.



Information Mediation BB

Registeration BB

Digital Registry BB

Identity BB

Payments BB

Consent BB

Workflow BB

Messaging BB

Scheduling BB

e-Signature BB

GIS BB

[4] https://govstack.gitbook.io/implementation-playbook/govstack-implementation-playbook/building-block-approach

The Payments Building Block

The GovStack Specification Payment Building Block consists of four different modules, two of these are core modules that would be part of every implementation regardless of jurisdiction, these are components that are fundamentally required for processing of G2P Payments. This is the Account Mapper and the Payments Hub. Apart from these there are two supporting components that would be required where needed. The Voucher Engine would be required where the Payments Building Block is expected to support issuance and redemption of vouchers as a mode of payment offline. An Oracle would be required to provide Alias Lookup services if a jurisdiction wishes to enable pay by alias but does not have existing alias management infrastructure.

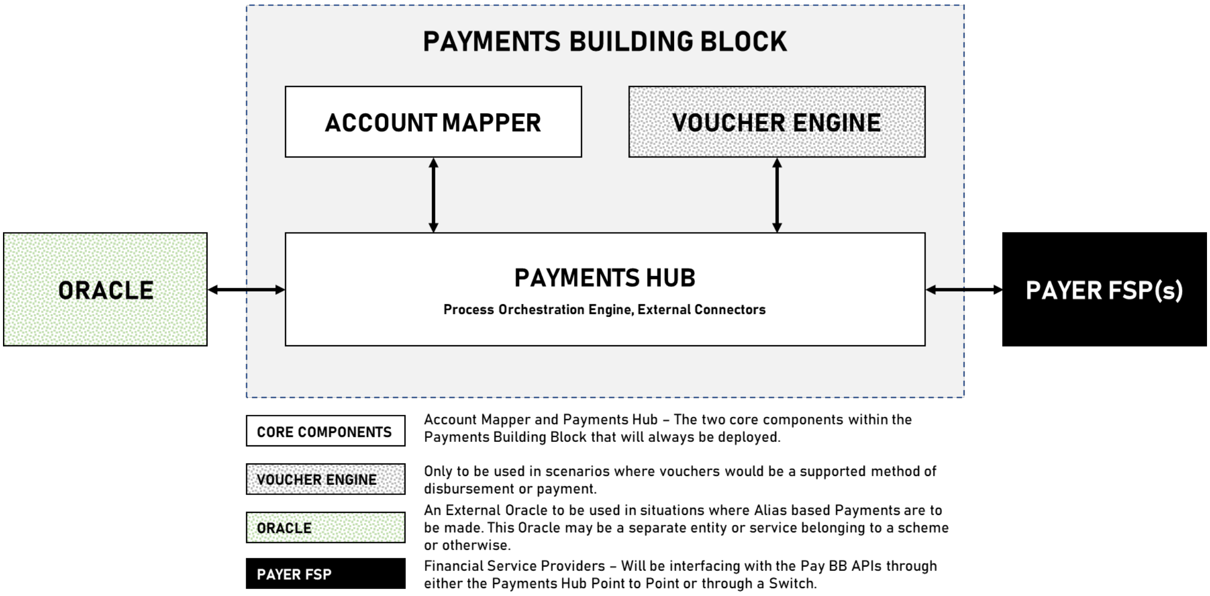

PAYMENTS BUILDING BLOCK
ACCOUNT MAPPER
VOUCHER ENGINE
ORACLE
PAYMENTS HUB
Process Orchestration Engine, External Connectors
PAYER FSP(s)
CORE COMPONENTS
Account Mapper and Payments Hub – The two core components within the Payments Building Block that will always be deployed.
VOUCHER ENGINE
Only to be used in scenarios where vouchers would be a supported method of disbursement or payment.
ORACLE
An External Oracle to be used in situations where Alias based Payments are to be made. This Oracle may be a separate entity or service belonging to a scheme or otherwise.
PAYER FSP
Financial Service Providers – Will be interfacing with the Pay BB APIs through either the Payments Hub Point to Point or through a Switch.

Figure 4 – Logical Components within the GovStack Specification Payments Building Block

1. **Payment Hub** – This is the core processing engine that will perform payment orchestration and operations such as bulking/de-bulking of instructions, this is also the component that will integrate with Financial Institutions of Schemes/Switches where needed.

2. **Account Mapper** – The account mapper service identifies the FSP, and exact destination address where the merchant/agent/payee's account that is used to route payouts to beneficiaries or settlement payments to agents/merchants.

3. **Voucher Engine** – This is the core voucher engine, the voucher component of the Payment Building Block facilitates the provisioning, issuance, activation, and redemption of vouchers. During the voucher provisioning process, sufficient funds must be allocated to the voucher. Vouchers can be issued but are not usable until they are in the hands of the beneficiary. The activation process makes the voucher active and usable for specific use cases and a limited time. When the vouchers are appropriately used, beneficiaries

receive benefits in the form of cash, products, or services from a third party, such as an agent or merchant.

4. **Oracle** – This is the Alias Lookup Service that is to be used for payments (if available in a jurisdiction) to be made without having to exchange sensitive payment information such as bank account numbers. Instead, beneficiaries can use an alias, which is a unique identifier that represents the user's payment information. This is optional because some jurisdictions may have Alias Lookup Service while others may not. If an ALS is not present but desired, the Mojaloop Oracle could be used but its governance and operations will be completely outside the scope of the Payments Building Block and is usually done by a scheme.

GovStack Payment BB in the Larger Scheme of Things

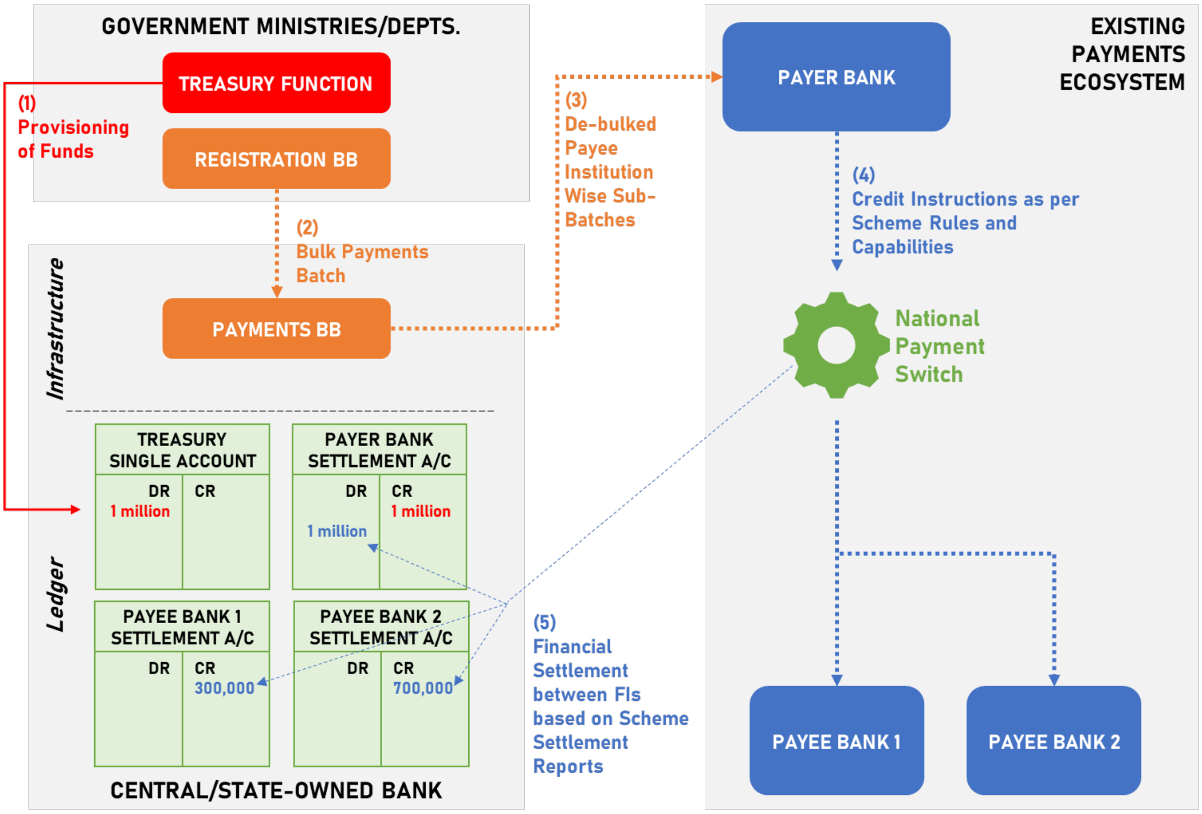

GOVERNMENT MINISTRIES/DEPTS.
TREASURY FUNCTION
REGISTRATION BB
(1) Provisioning of Funds
(2) Bulk Payments Batch
PAYMENTS BB
Infrastructure
Ledger
TREASURY SINGLE ACCOUNT
DR
CR
1 million
PAYER BANK SETTLEMENT A/C
DR
CR
1 million
1 million
PAYEE BANK 1 SETTLEMENT A/C
DR
CR
300,000
PAYEE BANK 2 SETTLEMENT A/C
DR
CR
700,000
CENTRAL/STATE-OWNED BANK
(3) De-bulked Payee Institution Wise Sub-Batches
(5) Financial Settlement between FIs based on Scheme Settlement Reports
EXISTING PAYMENTS ECOSYSTEM
PAYER BANK
(4) Credit Instructions as per Scheme Rules and Capabilities
National Payment Switch
PAYEE BANK 1
PAYEE BANK 2

Figure 5 – Interaction between various different components in a GovStack Specification Pay-BB Implementation

The diagram depicts the financial workflow within a government payment ecosystem. It illustrates the flow of funds from government ministries, detailing the provisioning of funds, registration, and payment processes. Bulk payments are batched at the Payments Building Block (BB), then proceed to payer banks and through the national payment switch, following which they are disbursed to payee banks. The ledger shows the transactions recorded at a central/state-owned bank. The process ensures that funds are distributed as per scheme rules and financial settlements between financial institutions are based on scheme settlement reports.

Bulk Payment Process in GovStack Specifications

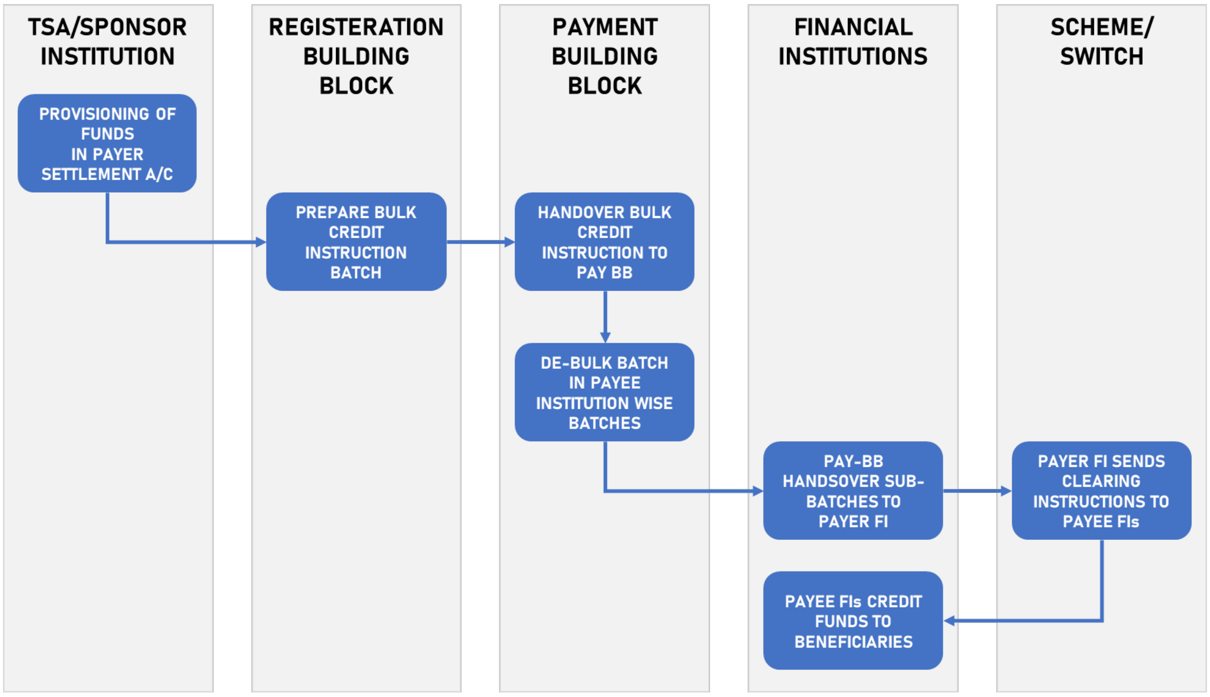

TSA/SPONSOR INSTITUTION
REGISTERATION BUILDING BLOCK
PAYMENT BUILDING BLOCK
FINANCIAL INSTITUTIONS
SCHEME/ SWITCH
PROVISIONING OF FUNDS IN PAYER SETTLEMENT A/C
PREPARE BULK CREDIT INSTRUCTION BATCH
HANDOVER BULK CREDIT INSTRUCTION TO PAY BB
DE-BULK BATCH IN PAYEE INSTITUTION WISE BATCHES
PAY-BB HANDSOVER SUB-BATCHES TO PAYER FI
PAYER FI SENDS CLEARING INSTRUCTIONS TO PAYEE FIs
PAYEE FIs CREDIT FUNDS TO BENEFICIARIES

Figure 6 – Core Functions in the Bulk Disbursement Process

Provisioning of Funds	The government ministry/agency/organization ensures the provisioning of funds which may be either through the Treasury Single Account at the Central Bank or at a State-Owned Bank, or through a sponsored account at a Commercial Financial Institution. This is assumed to be a manual process where the TSA/Sponsoring FI is notified by the Ministry of their intent to initiate a bulk transfer so that sufficient provisioning of funds can be ensured.
Preparation of Bulk Credit Batch	The government ministry/agency/organization manages their beneficiary list in the registration building block, finalized beneficiary lists are created and disbursement schedules is maintained within the Registration Building Block.
Send Bulk Instructions to Payments Building Block	The Registration BB calls the Payments BB API to handover a bulked credit instruction batch.
De-bulk Payment Batch	The Payment BB de-bulks the large payments batch payee institution wise.
Send Sub-batches to Payer FSP	The Payment BB sends the multiple de-bulked sub-batches to the Payer FSP.
Clearing Instructions	Payer FI sends clearing instructions to the Payee FIs through the prevalent scheme/switch ecosystem.
Post funds to beneficiaries	Payee FIs post funds to beneficiary accounts/mobile wallets.

Relationship between the G2P Payment Scheme and the Payer FSP/Treasury Function

A basic assumption is that multiple G2P Payment Schemes must be able to register on different Source Building Blocks, the Registration Building Block for example is one such Source Building Block in GovStack terminology, in the case of G2P Connect, Scheme Management plays this same role.

Within each Source Building Block, every program that must initiate payments needs to be registered in the Payments Building Block. In the later parts of this document, specifically in the APIs we have referred to this as "RegisteringInstitutionID". Meaning a unique identified which must be configured in the Payments Building Block and be used to identify source of Payments when Payment interfaces are invoked. In the proposed structure.

- Each G2P Payment Scheme can have a relationship with 1 or more Payer FSPs.
- Each G2P Payment Scheme can have one or more Budget Accounts from which it will do G2P Disbursement.

Registering Inst. ID	Program ID	Program Name	Payer FSP ID	Budget Account	Default
GOV001	P001	Unconditional Cash Transfer	FSP001	LU28 00194006447500003	Y
GOV001	P002	Flood Relief Support	FSP001	UK28 00123206447512312	N
GOV002	P001	Education Scholarship	FSP002	LU28 00178341681732621	Y
GOV003	P001	Maternity Benefits	FSP003	PK28 12576344231577535	Y
GOV003	P002	Special Benefits	FSP004	LU28 00194006447500003	Y

Using the above configuration table, when Registering Institution Gov001 wishes to make payments under the Unconditional Cash Transfer Program, it will invoke the Bulk Payments and provides both the Program ID as well as the Registering Institution ID in the header of the interface. This is used to determine which budget account is to be debited when these payments are processed.

GovStack Bulk Payment Process

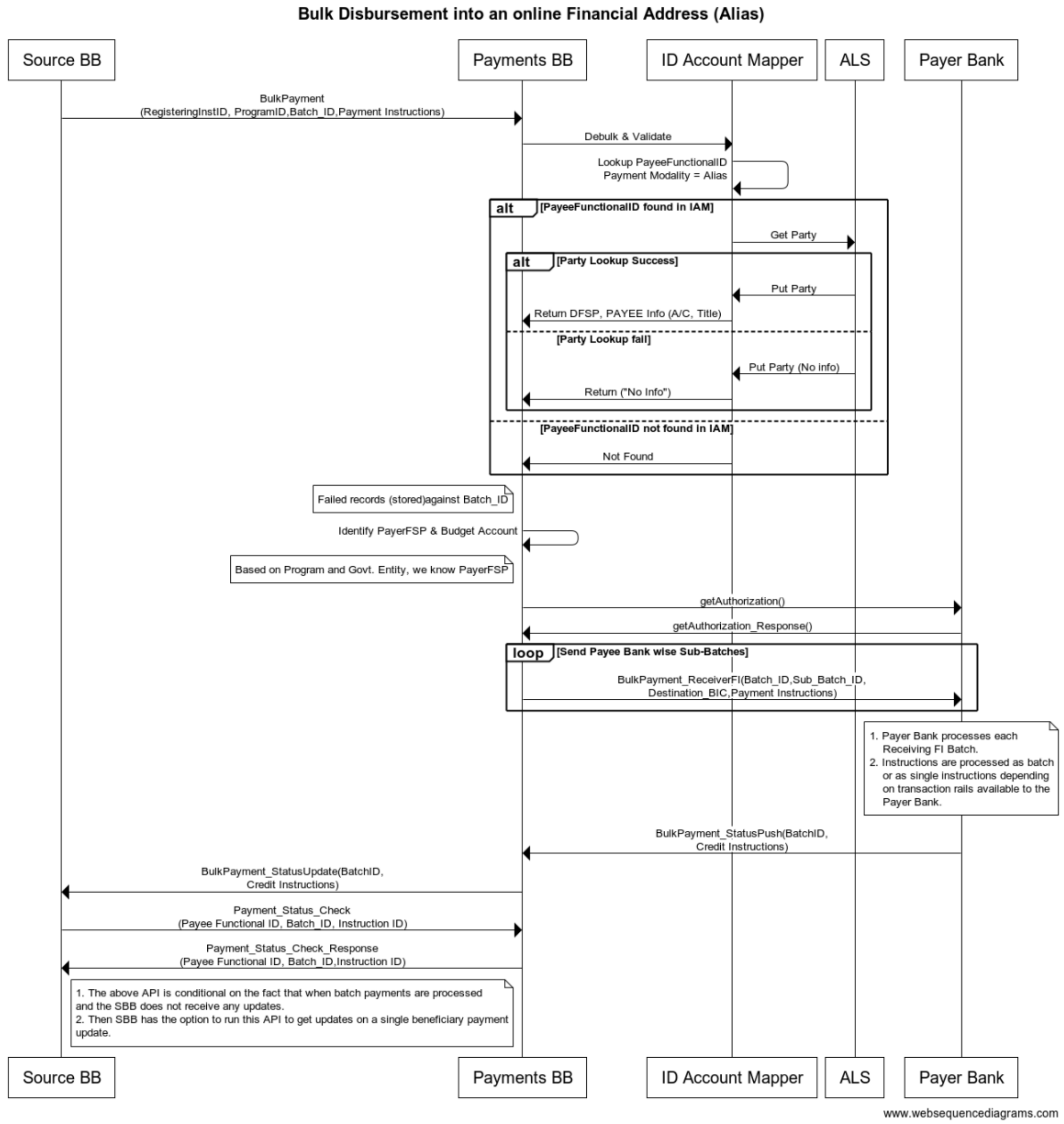

Bulk Disbursement into an online Financial Address (Alias)
Source BB
Payments BB
ID Account Mapper
ALS
Payer Bank
BulkPayment
(RegisteringInstID, ProgramID,Batch_ID,Payment Instructions)
Debulk & Validate
Lookup PayeeFunctionalID
Payment Modality = Alias
alt
[PayeeFunctionalID found in IAM]
Get Party
alt
[Party Lookup Success]
Put Party
Return DFSP, PAYEE Info (A/C, Title)
[Party Lookup fail]
Put Party (No info)
Return ("No Info")
[PayeeFunctionalID not found in IAM]
Not Found
Failed records (stored)against Batch_ID
Identify PayerFSP & Budget Account
Based on Program and Govt. Entity, we know PayerFSP
getAuthorization()
getAuthorization_Response()
loop
[Send Payee Bank wise Sub-Batches]
BulkPayment_ReceiverFI(Batch_ID,Sub_Batch_ID,
Destination_BIC,Payment Instructions)
1. Payer Bank processes each Receiving FI Batch.
2. Instructions are processed as batch or as single instructions depending on transaction rails available to the Payer Bank.
BulkPayment_StatusPush(BatchID,
Credit Instructions)
BulkPayment_StatusUpdate(BatchID,
Credit Instructions)
Payment_Status_Check
(Payee Functional ID, Batch_ID, Instruction ID)
Payment_Status_Check_Response
(Payee Functional ID, Batch_ID,Instruction ID)
1. The above API is conditional on the fact that when batch payments are processed and the SBB does not receive any updates.
2. Then SBB has the option to run this API to get updates on a single beneficiary payment update.
Source BB
Payments BB
ID Account Mapper
ALS
Payer Bank
www.websequencediagrams.com

Figure 7 – Bulk Disbursement API Specification Flows

The diagram outlines the bulk disbursement process to online financial addresses (aliases) in a financial system. It begins with the Source Building Block (BB) providing payment instructions to the Payments BB, which debulks and validates the details, interfacing with an ID Account Mapper to match payee information. Depending on the result, it sends payment batches to the payer bank. The bank processes each transaction, while the Payments BB updates the payment status. The process is designed for efficiency and traceability, ensuring that funds reach the correct recipients.

Harmonization Approach

The harmonization approach aims to create a seamless interface between G2PConnect's Scheme Management and GovStack' s Payment Building Block. This integration facilitates the processing of payments and disbursements by aligning the Source Building Block's operations, which are based on G2P-Connect specifications, with GovStack' s payment systems. The rationale behind this is to leverage the strengths of both platforms, ensuring a robust, interoperable framework that can handle the complexities of social benefit program management while maintaining efficient, secure, and transparent financial transactions. This approach is set to establish a unified operational standard, enhancing service delivery and scalability across different program implementations.

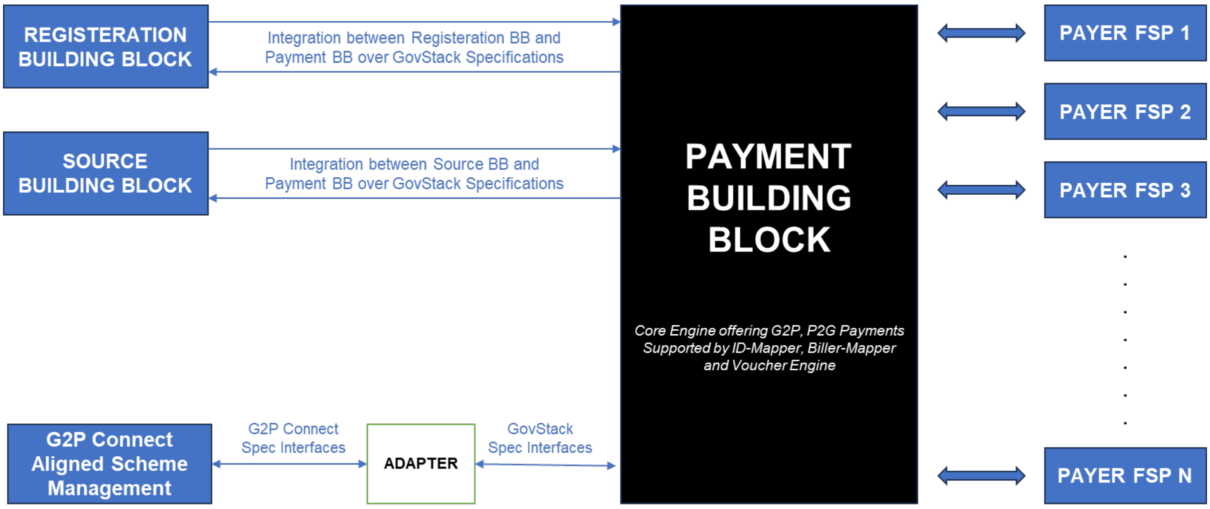

REGISTERATION BUILDING BLOCK
Integration between Registeration BB and Payment BB over GovStack Specifications
SOURCE BUILDING BLOCK
Integration between Source BB and Payment BB over GovStack Specifications
PAYMENT BUILDING BLOCK
Core Engine offering G2P, P2G Payments Supported by ID-Mapper, Biller-Mapper and Voucher Engine
PAYER FSP 1
PAYER FSP 2
PAYER FSP 3
PAYER FSP N
G2P Connect Aligned Scheme Management
G2P Connect Spec Interfaces
ADAPTER
GovStack Spec Interfaces

Figure 8 – Overview of Interoperation between GovStack Pay-BB and G2P Connect Scheme Management

Principles for Harmonization

In the proposed harmonization framework, it is suggested that the integration of G2P Connect and GovStack is guided by a set of principles. These principles are designed to anchor the technical strategy, justifying the importance of each, and illustrating their role in facilitating a coherent and effective union between the two systems. The focus is on non-disruptive adaptation, maintaining data coherence, and effectively bridging the informational divides that stand between the current specifications of both platforms.

Principle 1: Adaptive Integration without Core Changes

Adopting an Adapter ensures GovStack specification APIs remain unaltered, maintaining system integrity. This principle is vital for preserving existing functionalities while enabling G2P Connect interactions, offering a sustainable solution for legacy systems.

Principle 2: Data Retention for Additional Information

The Adapter will have data retention capabilities to manage extra information required by G2P Connect. This principle supports enhanced data flows necessary for complete transactions, ensuring no loss of critical payment information during processing.

Principle 3: Handling Information Gaps with Static Solutions

Where G2P Connect requires, information absent in GovStack, the Adapter will provide static data or highlight gaps for resolution. This principle ensures continuity in information exchange, facilitating a collaborative approach to resolving specification mismatches.

Implementation Considerations for the Adapter

In the context of Bulk Payments, implementing the adapter pattern involves creating an interface that bridges the gap between G2P Connect's Bulk Payment specifications and GovStack' s Payment APIs. The adapter pattern is crucial here because it allows the two systems, which have different mechanisms for handling bulk payments, to communicate without direct interaction. By defining a 'target' interface based on GovStack' s expectations and an 'adapter' that translates G2P Connect's payment instructions into this target format, bulk payment transactions can be processed seamlessly. This method retains the integrity of the existing systems while facilitating necessary interactions for bulk payment processing. The adapter thus serves to translate and manage payment instructions, ensuring compatibility and maintaining the robustness of the bulk payment operations.

What is the adapter pattern in the context of Bulk Payments?

The adapter pattern is a design solution that enables disparate Bulk Payment interfaces between G2P Connect and GovStack to interact. It involves establishing a new interface (the target) that conforms to the client's expectations within GovStack' s environment, while internally translating and mapping these expectations to the G2P Connect system through an intermediary (the adapter).

Why use the adapter pattern for Bulk Payments?

Using the adapter pattern for Bulk Payments is essential for integrating systems with divergent payment processing protocols. It allows for the preservation of G2P Connect's and GovStack's core functionalities, while still enabling the smooth processing of bulk payment transactions. This is crucial when direct changes to either system's codebase are impractical or potentially disruptive.

How to implement the adapter pattern for Bulk Payments?

To implement the adapter pattern in Bulk Payments, define a target interface for GovStack that represents the expected Bulk Payment processing methods. Then, create an adapter that embodies this target interface, internally converting G2P Connect Bulk Payment calls into a form acceptable by GovStack. The adapter handles all necessary translations, ensuring that Bulk Payments initiated by G2P Connect can be successfully executed by GovStack' s Payment Building Block without direct modification to either system's existing interfaces.

Placement of the Adapter

Delving into the implementation considerations for the adapter pattern, it is important to explore various strategic options for its placement. We examined four distinct approaches:

Each option presents unique advantages and implications for the integration process, crucial for determining the most effective and efficient path towards harmonizing G2PConnect and GovStack systems.

1. **Embedded within G2PConnect Scheme Management**: The adapter is integrated directly into G2PConnect's scheme management, allowing it to translate data to GovStack' s format from within the G2PConnect environment.

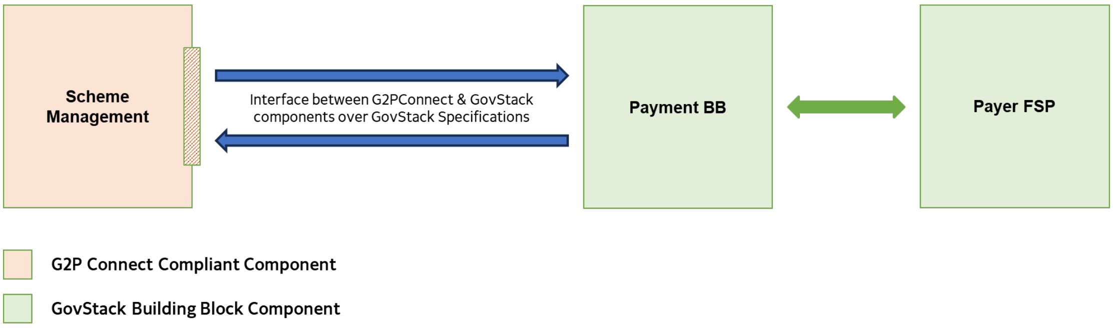

Scheme Management
Interface between G2PConnect & GovStack components over GovStack Specifications
Payment BB
Payer FSP
G2P Connect Compliant Component
GovStack Building Block Component

Figure 9: Adapter built into G2P Connect Specs

The diagram showcases the integration of the adapter within G2PConnect's Scheme Management framework. This positioning allows for the direct translation of Scheme Management data into the format required by GovStack's Payment Building Block. It simplifies the data translation process, ensuring that information relayed from Scheme Management is immediately comprehensible to GovStack, thereby optimizing the efficiency of the payment disbursement process within the G2PConnect environment.

2. **Embedded within GovStack Payment Building Block**: This approach involves embedding the adapter within GovStack' s payment building block, enabling it to interpret and process data coming from G2PConnect.

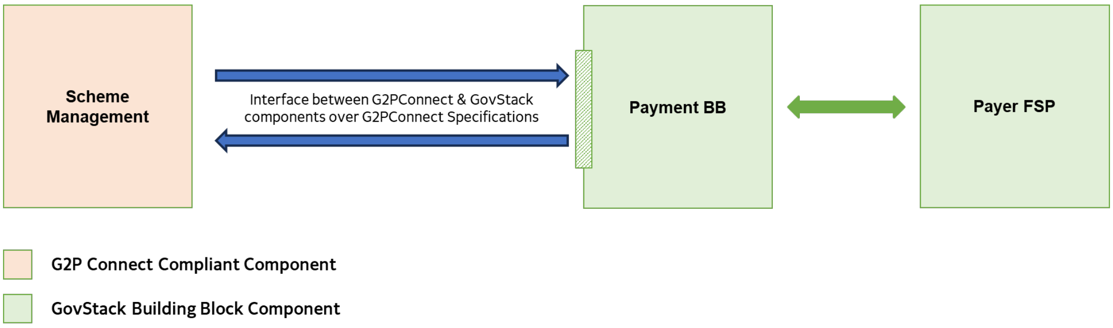

Scheme Management
Interface between G2PConnect & GovStack components over G2PConnect Specifications
Payment BB
Payer FSP
G2P Connect Compliant Component
GovStack Building Block Component

Figure 10: Adapter built into GovStack Specs

In this integration approach, the adapter is incorporated directly into GovStack's Payment Building Block. This configuration enables GovStack to natively understand and interact with data formatted according to G2PConnect specifications. By embedding the adapter, it becomes an intrinsic part of GovStack's processing flow, converting data on-the-fly, and streamlining the communication pipeline between Scheme Management and the Payment BB, ultimately ensuring that transactions are executed smoothly and without the need for external translation mechanisms.

3. **Standalone Middleware Component**: Here, the adapter functions as an independent middleware, separate from both G2PConnect and GovStack, facilitating communication between the two.

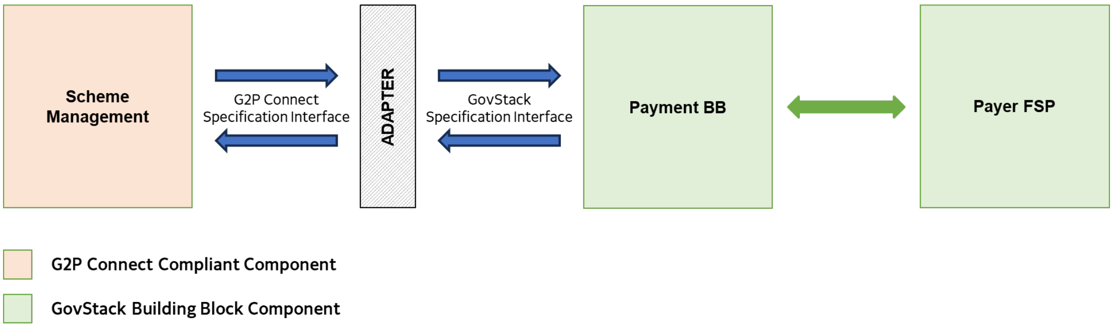

Scheme Management
G2P Connect Specification Interface
ADAPTER
GovStack Specification Interface
Payment BB
Payer FSP
G2P Connect Compliant Component
GovStack Building Block Component

Figure 11: Standalone Adapter

This middleware layer is crucial for translating and relaying messages and data formats between the two systems, ensuring that they can understand and respond to each other's requests without any direct modifications. It acts as a neutral translator, allowing Scheme Management to send and receive information to and from Payment BB, with the adapter ensuring the fidelity and compatibility of the communicated data.

4. **Standalone Translation Service**: The adapter operates as an independent service, focusing solely on translating and returning adapted payloads between G2PConnect and GovStack, without being part of either's internal structure.

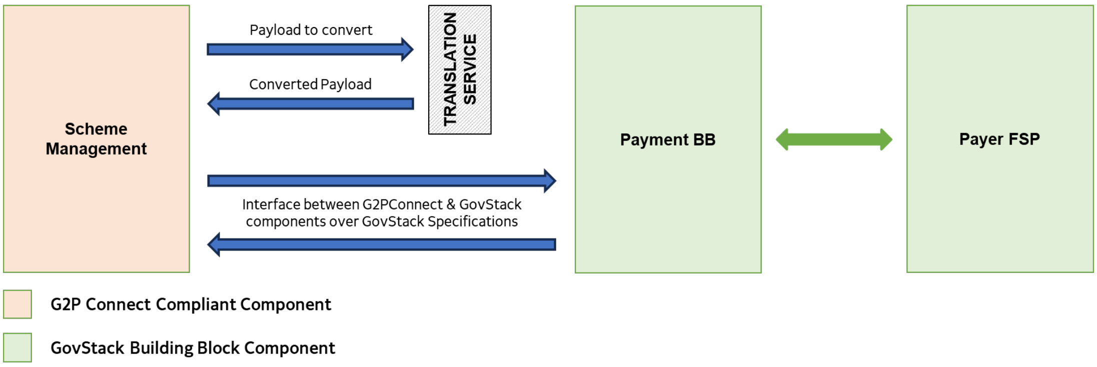

Scheme Management
Payload to convert
Converted Payload
TRANSLATION SERVICE
Interface between G2PConnect & GovStack components over GovStack Specifications
Payment BB
Payer FSP
G2P Connect Compliant Component
GovStack Building Block Component

Figure 12: Standalone Translation Service

The translation service acts as a mediator to ensure smooth communication between G2PConnect's Scheme Management and GovStack' s Payment Building Block (Pay-BB). When initiating a disbursement, Scheme Management sends a request to the translation service, which converts the G2PConnect payload into a GovStack-compliant format. This adapted payload is then used to invoke the Pay-BB API. Upon receiving a response, the translation service reconverts the GovStack payload back into the G2PConnect format, allowing Scheme Management to process and understand the transaction outcome. This bidirectional translation ensures that both systems operate within their own specifications while maintaining a coherent transaction flow.

Preferred Approach

We prefer the Standalone Middleware Component approach for placing the adapter due to its independent operation, enhancing the interaction between G2PConnect and GovStack. This method brings modularity, allowing both systems to evolve separately with minimal dependency. It introduces flexibility, permitting updates or replacements without modifying the core systems. Additionally, it simplifies scalability, as the middleware can be scaled without impacting the connected systems. Lastly, it streamlines maintainability, as any issues within the middleware can be resolved independently of G2PConnect or GovStack, ensuring a resilient integration framework.

Identifying Gaps and Key Harmonization Areas

GovStack and G2PConnect share many similarities, they both focus on facilitating government-to-person services and the use of APIs for interoperability in public service delivery. Both aim to enhance efficiency and transparency in government transactions. Differences lie in their approach: G2PConnect is specific to payment and social benefit schemes, with a specialized architecture for financial transactions. GovStack has a broader scope with building blocks for various digital government services. Their implementations and specifications are tailored to their unique goals, with G2PConnect emphasizing financial disbursements and GovStack providing a more general digital infrastructure.

Disbursement APIs

#	Type	Endpoint	Description of Interface	GovStack Equivalent Interface
1	Async	POST /disburse	Social protection platform initiating a G2P disbursement. Initiate payment through disbursement instructions	BulkPayment() POST /{{BulkHostName}}/batchtransactions?type=raw
2	Async	POST /on-disburse	Disbursement initiating systems receive disbursement status info through callback end points. Disburse response through callback	BulkPayment_StatusPush() to provide Batch Summary details. • Case 1 : As a call back of Batch Payment. • X-Callback-URL registered in the request GET BatchDetails API to fetch transaction level status of each batch and append that in the response back to on-disburse
3	Sync	/disburse/sync/disburse	Initiate payment through disbursement instructions through sync call	GovStack only has the Async interface in this regard, and by principle we do not see any value in implementing bulk payments through synchronous interfaces. Our approach therefore is to stick with an Async interface on the GovStack side of the integration and manage sync-async orchestration on the wrapper layer. BulkPayment() POST /{{BulkHostName}}/batchtransactions?type=raw

Batch/Transaction Status Check APIs

#	Type	Endpoint	Description of Interface	GovStack Equivalent Interface
4	Async	POST /disburse/txn/status	Status check of previous disbursement transactions using transaction_id and/or reference_id(s)	Payment_Status_Check Request() Synchronous interface in GovStack specs. G2P Connect Specs are Async whereas our specs are Sync.
5	Async	POST /disburse/txn/on-status	Disbursement status to Social Protection, Treasury Systems.	None – Since the Payment_Status_Check Request() in GovStack specs is Sync in this case, there is no comparative endpoint for this interface.
6	Sync	/disburse/sync/search	Fetch disbursement processing status though sync call	There is no comparative endpoint for this interface in GovStack specifications.
7	Sync	/disburse/sync/txn/status	Sync status check of disburse Async APIs	GET BatchDetails API to fetch transaction level status of each batch.

Other APIs

#	Type	Endpoint	Description of Interface	GovStack Equivalent Interface
8	Async	POST /disburse/search	Search disbursement by searchable attributes	There is no comparative endpoint for this interface in GovStack specifications.
9	Async	POST /disburse/on-search	Disbursement search results through callback	There is no comparative endpoint for this interface in GovStack specifications.

Sample Comparison on Disbursal API

G2P Connect Specification			GovStack Pay-BB Specification		
POST /disburse			POST /{{BulkHostName}}/batchtransactions?type=raw		
Field	**Description**	**Y/N**	**Field**	**Description**	**Y/N**
reference_id	Unique reference_id set by txn initiating system for each request in a batch.	Y	requestId	Individual ID for each instruction in the Credit Batch.	Y
payer_fa	Payer's Financial Address.	Y	debitParty. value	Payer's Financial Address.	Y
payee_fa	Payee's Financial Address.	Y	creditParty. value	Payee's Financial Address.	Y
amount	Describes amount in decimal value format.	Y	amount	Amount to be credited.	Y
scheduled_ timestamp	If not present then immediate disbursement else schedule as per scheduled date-time.	N	Not Supported		
payer_name	Payer account name for easy reference.	N	Not Supported		
payee_name	beneficiary name.	N	Not Supported		
note	Note to payer/payee to reflect on statements.	N	Not Supported		
purpose	Payment disbursement purpose e.g Salary, Farmer Relief 2023.	N	description Text	Description/Narration of Payment.	Y
instruction	Additional systems instruction for interfacing systems.	N	Not Supported		
currency_co de	Currency code allocated as per ISO 4217 format.	N	currency	Transaction Currency Code.	Y
locale	indicates language code. G2P Connect supports country codes as per ISO 639.3 standard.	N	Not Supported		
Missing in G2P Connect Specs/Design →			paymentMode	Modality of the payment.	Y

API Definitions and Message Flows

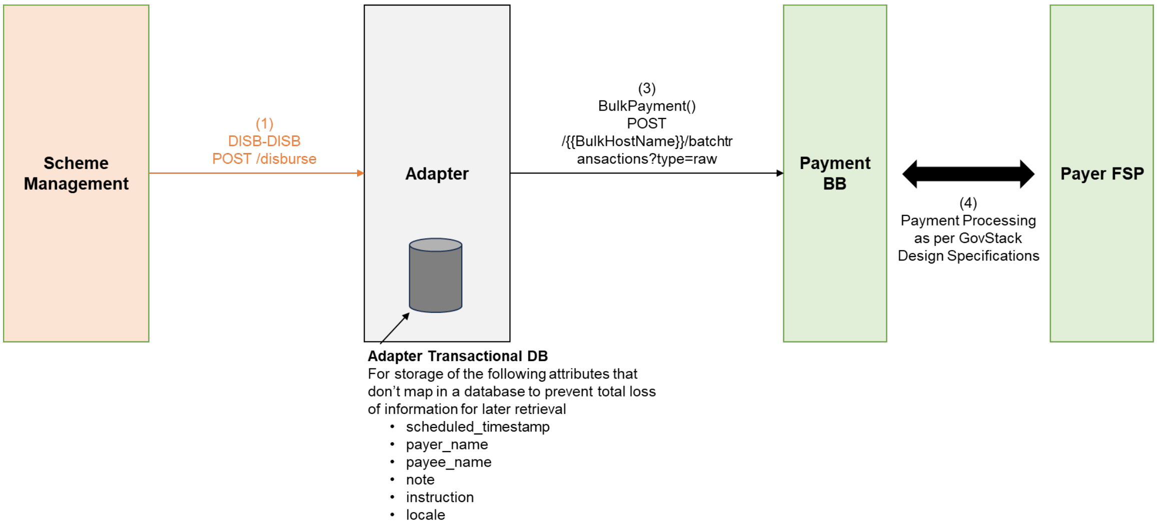
1. Call to initiate a Bulk Disbursement

G2PConnect Interface Type: Asynchronous

G2PConnect Endpoint: POST /disburse

GovStack Interface Type: Asynchronous

GovStack Endpoint: POST/{{BulkHostName}}/batchtransactions?type=raw



- **Transaction Initiation:** Scheme Management sends a disbursement request to the Adapter, signaling the commencement of a financial operation.
- **Disbursement Execution:** The Adapter translates the request into the GovStack Payment BB's format and forwards it for processing. Payment BB, upon receiving the request, proceeds with the transaction through the Payer FSP.
- **Acknowledgment of Receipt:** In response, the Payer FSP issues an asynchronous HTTP response, acknowledging the receipt of the message and confirming that the request is being processed.

  The diagram does not depict the transaction's completion or the response back to Scheme Management after the transaction processing, as this stage focuses solely on initiating and acknowledging the disbursement request. The actual completion of the transaction and final confirmation would occur outside of this immediate process flow.

2. **Status of Disbursement against Request Raised in POST/Disburse Endpoint**

G2PConnect Interface Type: Asynchronous

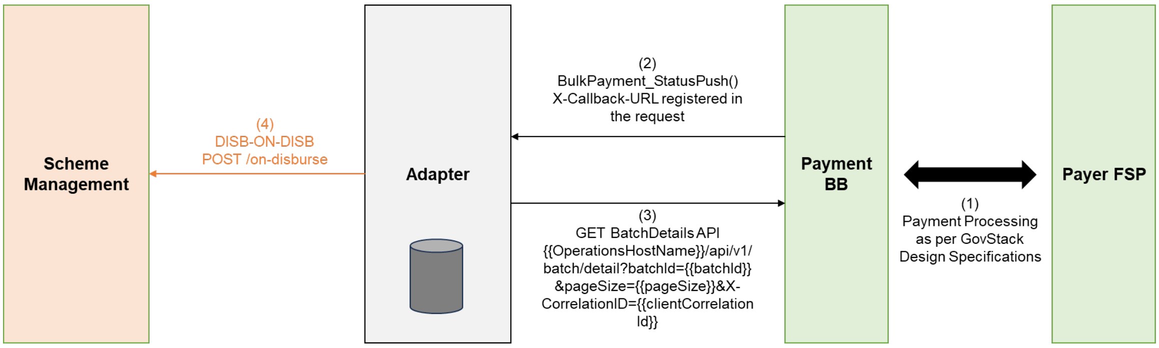
G2PConnect Endpoint: POST /on-disburse

GovStack Interface Type: Asynchronous

GovStack Endpoint: 2 Endpoints to be orchestrated to prepare response as per G2PConnect POST /on-disburse Expectations.

- BulkPayment_StatusPush() to provide Batch Summary details as a call back of Batch Payment.
- GET BatchDetails API to fetch transaction level status of each batch and append that in the response back to on-disburse.

Following the bulk disbursement request initiation, the process continues as the Payment Building Block (Pay-BB) concludes the payment transactions with the Payer FSP. Upon completion, Pay-BB invokes the Adapter with a status push, delivering a summary of processed payments. Subsequently, the Adapter engages the Get BatchDetails API to aggregate a detailed report of all transactions, including both processed and pending. This report preparation by the Adapter involves meticulous data management, ensuring that the payload for the POST /on-disburse call to Scheme Management is comprehensive, containing both the summary and the specifics of each transaction within the batch. This multi-step workflow ensures that Scheme Management receives a complete overview of the disbursement outcome, allowing for accurate record-keeping and reconciliation.


Scheme Management
(4)
DISB-ON-DISB
POST /on-disburse
Adapter
(2)
BulkPayment_StatusPush()
X-Callback-URL registered in the request
(3)
GET BatchDetails API
{{OperationsHostName}}/api/v1/batch/detail?batchId={{batchId}}&pageSize={{pageSize}}&X-CorrelationID={{clientCorrelationId}}
Payment BB
(1)
Payment Processing as per GovStack Design Specifications
Payer FSP

- **Payment Processing Summary:** The Pay-BB concludes transactions with the Payer FSP and initiates a status push to the Adapter, providing a summary of the processed payments.
- **Detailed Transaction Retrieval:** The Adapter then invokes the Get BatchDetails API to compile detailed information on all transactions, both processed and unprocessed.
- **Data Management for Scheme Management:** The Adapter undertakes significant data management tasks to construct the detailed payment status, which is necessary for the Scheme Management API call.
- **Final Disbursement Report:** With the comprehensive payment status report prepared, the Adapter is set to execute the POST /on-disburse endpoint, delivering both summary and individual transaction details back to Scheme Management.

3. Get Updated Status of Disbursement against Request Raised in POST/Disburse Endpoint

G2PConnect Interface Type: Asynchronous

G2PConnect Endpoint: POST /disburse/txn/status

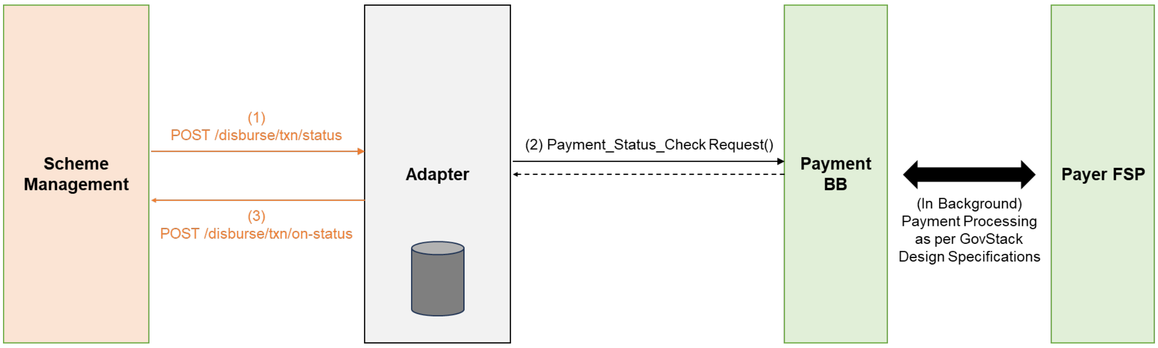
G2PConnect Interface Type: Asynchronous

G2PConnect Endpoint: POST /disburse/txn/on-status

GovStack Interface Type: Synchronous

GovStack Endpoint: Payment_Status_Check Request()

The process depicted below involves the Adapter acting as an intermediary for status verification of disbursed transactions. Scheme Management queries the Adapter to check the status of a specific batch or transaction using G2PConnect's asynchronous API. The Adapter then synchronously calls GovStack's Payment Status Check Request. Once the payment status is retrieved, the Adapter utilizes a callback to provide the response to G2PConnect, thereby completing the status verification loop. This ensures the Scheme Management system is kept updated on the disbursement status in accordance with GovStack's protocols.


Scheme Management
(1)
POST /disburse/txn/status
(3)
POST /disburse/txn/on-status
Adapter
(2) Payment_Status_Check Request()
Payment BB
(In Background) Payment Processing as per GovStack Design Specifications
Payer FSP

- **Initiation of Status Check:** Scheme Management initiates the process with a status inquiry for a disbursed batch or individual transaction.
- **Adapter's Synchronous Operation:** The Adapter receives the inquiry and performs a synchronous call to GovStack's Payment BB using the Payment Status Check Request.
- **Response Handling and Callback Execution:** Upon obtaining the payment status, the Adapter responds to Scheme Management via G2PConnect's callback mechanism, delivering the transaction's current state.
- **Completion of Status Inquiry Loop:** This sequence ensures that Scheme Management receives timely updates on the disbursement status, reflecting the latest data from GovStack's systems.

Data Retention in the Adapter

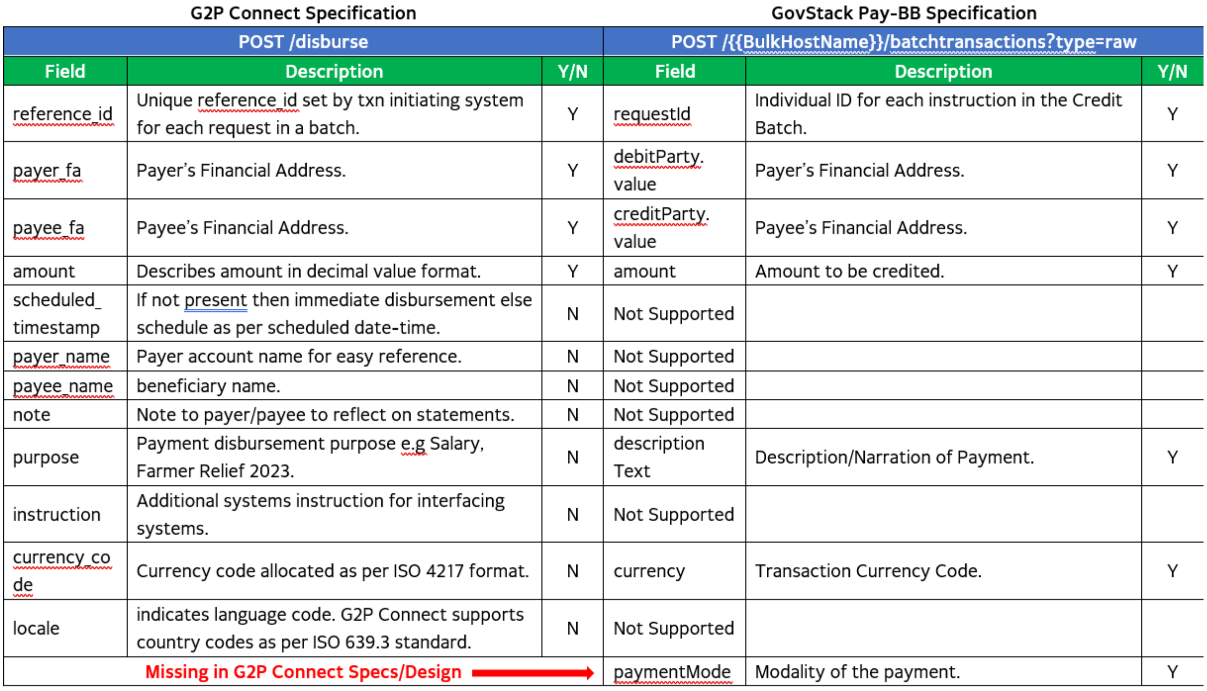
Data retention within the adapter context refers to the systematic storage of transactional information that the adapter processes between G2PConnect and GovStack. It includes storing necessary details temporarily for the successful completion of transactions and facilitating asynchronous communications.

The Need for Data Retention

Retention is crucial to ensure the reliability of the transaction process, enabling the adapter to manage asynchronous requests and responses effectively, and to provide a mechanism for transaction traceability and recovery in case of interruptions.

As an example, In the context of the Disbursal API, data retention in the adapter is critical for fields that are present in the G2PConnect specification but not supported by GovStack. For instance, the 'payer_name' and 'purpose' fields are vital for G2PConnect's processing but are not used by GovStack. The adapter must store these fields temporarily so that when G2PConnect expects this information in the response, the adapter can replenish it from its data retention layer, ensuring continuity and integrity of the information for Scheme Management.

Conversely, the adapter may also need to populate certain fields to craft a GovStack-compliant call based on data not provided by G2PConnect, demonstrating the adapter's role in data retention and translation between the two systems.



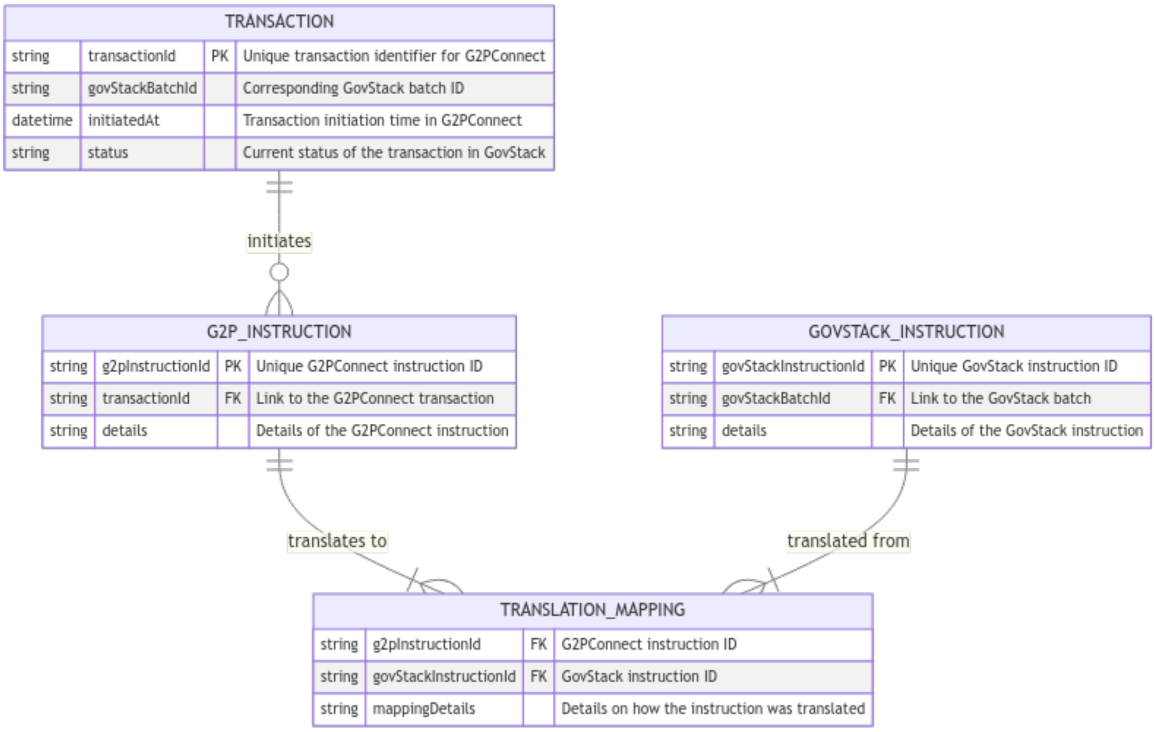
G2P Connect Specification			GovStack Pay-BB Specification		
POST /disburse			POST /{{BulkHostName}}/batchtransactions?type=raw		
Field	Description	Y/N	Field	Description	Y/N
reference_id	Unique reference_id set by txn initiating system for each request in a batch.	Y	requestId	Individual ID for each instruction in the Credit Batch.	Y
payer_fa	Payer's Financial Address.	Y	debitParty.value	Payer's Financial Address.	Y
payee_fa	Payee's Financial Address.	Y	creditParty.value	Payee's Financial Address.	Y
amount	Describes amount in decimal value format.	Y	amount	Amount to be credited.	Y
scheduled_timestamp	If not present then immediate disbursement else schedule as per scheduled date-time.	N	Not Supported		
payer_name	Payer account name for easy reference.	N	Not Supported		
payee_name	beneficiary name.	N	Not Supported		
note	Note to payer/payee to reflect on statements.	N	Not Supported		
purpose	Payment disbursement purpose e.g Salary, Farmer Relief 2023.	N	description Text	Description/Narration of Payment.	Y
instruction	Additional systems instruction for interfacing systems.	N	Not Supported		
currency_code	Currency code allocated as per ISO 4217 format.	N	currency	Transaction Currency Code.	Y
locale	indicates language code. G2P Connect supports country codes as per ISO 639.3 standard.	N	Not Supported		
Missing in G2P Connect Specs/Design →			paymentMode	Modality of the payment.	Y

- **Field Compatibility and Data Retention:** Data retention within the adapter addresses the discrepancy between G2PConnect's and GovStack's data fields. It ensures that transaction-specific information required by G2PConnect but not utilized by GovStack is not lost in translation.
- **Adapter's Role in Data Preservation:** The adapter acts as a repository, preserving data such as 'payer_name' and 'purpose' during the transaction lifecycle, which are critical for G2PConnect's internal processes and reporting requirements.

- **Data Enrichment for Compliance:** For GovStack compliance, the adapter enriches outgoing data by injecting necessary fields that are not provided by G2PConnect, such as 'paymentMode', ensuring that GovStack's Payment BB can process the request appropriately.
- **Example of Data Retention Utilization:** Using the fields 'payer_name' and 'purpose' as an example, the adapter's data retention layer temporarily holds these values so that when a response is crafted back to G2PConnect, these fields are populated as expected, maintaining data consistency and integrity.

Adapter Data ERD

In the ERD below, the prefix **G2P_** denotes elements specific to G2PConnect, making it clear which parts of the data model pertain to which system. This will help maintain separate logical domains within the same data storage strategy and ensure data integrity when interacting between the two specifications.



TRANSACTION

string	transactionId	PK	Unique transaction identifier for G2PConnect
string	govStackBatchId		Corresponding GovStack batch ID
datetime	initiatedAt		Transaction initiation time in G2PConnect
string	status		Current status of the transaction in GovStack

G2P_INSTRUCTION

string	g2pInstructionId	PK	Unique G2PConnect instruction ID
string	transactionId	FK	Link to the G2PConnect transaction
string	details		Details of the G2PConnect instruction

GOVSTACK_INSTRUCTION

string	govStackInstructionId	PK	Unique GovStack instruction ID
string	govStackBatchId	FK	Link to the GovStack batch
string	details		Details of the GovStack instruction

TRANSLATION_MAPPING

string	g2pInstructionId	FK	G2PConnect instruction ID
string	govStackInstructionId	FK	GovStack instruction ID
string	mappingDetails		Details on how the instruction was translated

In this ERD, **TRANSACTION** represents a batch of instructions from G2PConnect. **G2P_INSTRUCTION** represents individual instructions within a transaction. **GOVSTACK_INSTRUCTION** is the GovStack equivalent of a G2PConnect instruction. **TRANSLATION_MAPPING** is the associative entity that maintains the relationship between an individual G2PConnect instruction and its GovStack counterpart, including details of the translation for follow-up status checks.

This structure ensures that each instruction's journey from G2PConnect to GovStack and back can be tracked, providing a clear audit trail for the entire transaction lifecycle.

Mermaid Code for ERD for future changes:

```
erDiagram
  TRANSACTION {
    string transactionId PK "Unique transaction identifier for G2PConnect"
    string govStackBatchId "Corresponding GovStack batch ID"
    datetime initiatedAt "Transaction initiation time in G2PConnect"
    string status "Current status of the transaction in GovStack"
  }
  G2P_INSTRUCTION {
    string g2pInstructionId PK "Unique G2PConnect instruction ID"
    string transactionId FK "Link to the G2PConnect transaction"
    string details "Details of the G2PConnect instruction"
  }
  GOVSTACK_INSTRUCTION {
    string govStackInstructionId PK "Unique GovStack instruction ID"
    string govStackBatchId FK "Link to the GovStack batch"
    string details "Details of the GovStack instruction"
  }
  TRANSLATION_MAPPING {
    string g2pInstructionId FK "G2PConnect instruction ID"
    string govStackInstructionId FK "GovStack instruction ID"
    string mappingDetails "Details on how the instruction was translated"
  }

  TRANSACTION ||--o{ G2P_INSTRUCTION : "initiates"
  G2P_INSTRUCTION ||--|{ TRANSLATION_MAPPING : "translates to"
  GOVSTACK_INSTRUCTION ||--|{ TRANSLATION_MAPPING : "translated from"
```

Limitations of Data Retention in the Adapter

While data retention within the adapter serves critical operational functions, it is important to recognize its limitations to prevent misconceptions about its capabilities and purpose.

1. **Temporal Nature of Storage**: The adapter is not designed for long-term data storage. Data is retained only as long as necessary to facilitate ongoing transactions and to provide an audit trail for a limited period.
2. **Scope of Data**: The adapter's data retention is transaction-centric and does not encompass broader data warehousing functionalities such as extensive historical analysis or data mining.
3. **Database Functionality**: It does not replace the functions of a full-fledged database system. Complex queries, indexing, and relationship management that you would expect in a transactional database are outside the scope of this data retention layer.
4. **Data Volume**: The storage capacity is not infinite; it is sized for optimal performance and not for accumulating data indefinitely.

5. **Compliance and Legal Constraints**: Data retention is subject to compliance with data privacy laws and regulations, which may limit the duration and nature of the data stored.
6. **Performance Considerations**: Retaining excessive amounts of data could impact the adapter's performance, potentially leading to slower response times and increased processing overhead.
7. **Security Risks**: Prolonged data retention can pose security risks; the more data held, the greater the potential impact of a data breach.

By acknowledging these limitations, the design of the adapter can be strategically tailored to balance data retention needs with system performance and compliance requirements.

Risk Assessment and Mitigation Strategies

1. **Data Discrepancies**:
   - **Nature**: Occurs if the adapter incorrectly translates data between G2PConnect and GovStack.
   - **Impact**: Could lead to erroneous transaction processing or data mismatch.
   - **Mitigation**: Implement rigorous testing protocols for data translation logic.
2. **Unauthorized Access**:
   - **Nature**: Risk of security breaches compromising sensitive financial information.
   - **Impact**: High, potentially leading to data leaks and loss of public trust.
   - **Mitigation**: Strengthen security measures, including encryption and access controls.
3. **API Misalignments**:
   - **Nature**: Incompatibilities due to changes in G2PConnect or GovStack APIs.
   - **Impact**: Could disrupt transaction processing.
   - **Mitigation**: Regular updates and compatibility checks for APIs.
4. **Performance Bottlenecks**:
   - **Nature**: Adapter unable to handle high volumes of transactions efficiently.
   - **Impact**: Slow response times, potential system crashes.
   - **Mitigation**: Scale resources and optimize code for high performance.
5. **Incorrect Data Translation**:
   - **Nature**: Adapter fails to accurately translate data fields.
   - **Impact**: Incorrect payments or transaction failures.
   - **Mitigation**: Establish a robust validation and verification system.
6. **System Failures**:
   - **Nature**: Breakdowns in the adapter due to software or hardware issues.
   - **Impact**: Transaction disruptions, potential data loss.

- **Mitigation**: Implement redundancy and regular system maintenance.

7. **Compliance Violations**:
    - **Nature**: Failure to adhere to data protection laws and financial regulations.
    - **Impact**: Legal repercussions, fines.
    - **Mitigation**: Regular compliance audits and staff training on regulatory requirements.
8. **Data Retention Overload**:
    - **Nature**: Excessive data retention leading to storage and management challenges.
    - **Impact**: Increased operational costs, potential system slowdowns.
    - **Mitigation**: Define clear data retention policies and automate data lifecycle management.

Risk Prioritization and Response Planning: After identifying and analyzing these risks, prioritize them based on their potential impact and likelihood. Develop detailed response plans for each high-priority risk, outlining specific actions, responsible parties, and recovery procedures.

Continuous Monitoring and Review: Establish mechanisms for ongoing risk monitoring and regular reviews of the risk management strategy to adapt to new threats and changes in the system environment.