BBPS TECHNICAL STANDARDS

Version 4.0

Date: 08.08.2019

Abstract
This document will act as a standard from a technical perspective to Bharat Bill Payment Operating Units (BBPOUs) to set up their system with respect to Bharat Bill Payment Central Unit (BBPCU).

Table of Contents

Document History

Date	Version	Description
05.12.2016	1.0	Base version of BBPS Technical Standards.
17.01.2017	2.0	▪ Modifications in BBPS Security Requirements. ▪ Modifications in BBPOU Certification Process.
04.12.2017	3.0	▪ Modifications in BBPOU System Setup.
08.08.2019	4.0	▪ Modifications across all the sections of the document.

1 BBPS Overview

Bharat Bill Payment System (BBPS) is a unified bill payment system for India. The BBPS intends to offer an interoperable and accessible bill payment services to customers through a network of agents, enabling multiple payment modes, and providing instant confirmation of payment. Bharat Bill Payments System is an integrated online platform under the umbrella of National Payments Corporation of India (NPCI) for all kinds of bill payments. This interoperable service will work through a network of agents and online modes to enable payment of bills via multiple payment modes along with instant generation of receipts of payments. To start with, the scope of BBPS will cover the bills of utility service companies. Other billers will be brought under BBPS in course of time, as and when decided by RBI.

2 About This Document

2.1 Purpose

This document will act as a standard from a technical perspective to Bharat Bill Payment Operating Units (BBPOUs) to set up their systems with respect to Bharat Bill Payment Central Unit (BBPCU), get connected to BBPCU, execute the functionalities of BBPOU and be part of the BBPS Ecosystem.

This document sets out a broad approach to designing systems that may be developed by different BBPOUs. It attempts to set general standards and create a consistent approach to the design and development of systems across the BBPS ecosystem.

2.2 Scope

The scope of this document primarily encompasses the BBPS functionalities in Production. However, it will undergo iterations over time in order to cater to the system requirements of BBPS envisaged for future enhancements in Production scenario.

The following sections cover the various considerations to be factored in while preparing the BBPOU applications.

3 BBPOU System Setup

This section will address the system setup required by the BBPOUs to get connected with BBPCU system in the Certification Environment as well as the Production Environment.

3.1 Network Requirements

1. **Network bandwidth:** Typically, a BBPOU would need adequate bandwidth infrastructure to connect and communicate with BBPCU. As a rough estimate BBPOUs expecting volumes up to 50 TPS should cater for a minimum 2 Mbps link. Usage of network bandwidth will be monitored continuously and the BBPOUs will be advised to upgrade their BW once the usage crossed threshold limits. For all the high volume players, it is strongly advised to bring in the SD-WAN solution for a better throughput and network resource optimization.

2. **Network IP and Port details:** BBPCU SPOC will provide the IP and Port details on request while establishing connection with BBPCU. The same needs to be shared by the respective BBPOUs for integrating with different environments – Sandbox, Comfort, Certification, Production & DR (Disaster Recovery) sites.
    a. All BBPOUs have to ensure that for their application there is single bi-directional IP / Port irrespective of channels, role of BBPOU. Thus for a BBPOU acting as both Customer and Biller BBPOU target IP has to be same.
    b. IP / Port combination for all environments must be mutually exclusive.

3.2 Software Requirements

1. NPCI does not mandate implementation of any specific software stack for the BBPOUs.
2. BBPOUs can deploy their own software stack, which is capable of sending and receiving signed XML messages with BBPCU.
3. BBPOUs should also have the capability to send and receive batch files from / to BBPCU.
4. Importantly, BBPOU application should ensure compliance with all requirements under BBPCU for end-to-end handling of ON-US as well as OFF-US transactions.
5. It is suggested to implement data storage and archival policy that complies with archival policy in line with banking regulations. Ideally it is recommended to keep active data backup with automatic failover.

3.3 Application Synchronization

1. The BBPCU and BBPOU systems should be synced with global NTP servers to ensure that there is hardly any difference in time between two systems.
2. To account for marginal difference, a tolerance of few seconds has been provisioned in BBPS.
3. The response timeout period between BBPCU and Biller BBPOU at system level is currently configured as 200* seconds within which Biller BBPOU has to respond back to BBPCU after sending the acknowledgement. Response timeout timer will start after the successful acknowledgement from Biller BBPOU.
4. The timeout period at Customer BBPOU end should be significantly higher (300* seconds approx.) than the timeout period maintained at BBPCU end to account for the transit of XML messages among different entities.

5. Any request or response from BBPCU to BBPOUs will be Connection timed out after 15* seconds and if the connection is established and the ACK is being awaited, then this will be Read Timed out if it exceeds 20* seconds. The BBPCU supported Connection Timeout is 15* seconds and Read Timeout is 20* seconds. The BBPOUs should respond with an ACK within Read timeout threshold which is 20* seconds.

   All these timers are subject to change

3.4 Master Data Management

1. Master data will reside at BBPCU level and only authorized users (BBPOU users) of the BBPCU system will have access to data pertaining to entities associated with the particular BBPOU.
2. BBPOUs need to fetch the Biller related data from the BBPCU system at defined intervals (for example, once in a day) for newly added, modified or deleted biller details.
3. The Biller related data, once retrieved from the BBPCU system through XML messages, needs to be stored in a secure manner at the BBPOU end.
4. Based on the Biller specific parameters, the rendering of elements on user interface should take place for the Customer BBPOU.
5. A reference to logic of generating IDs in the BBPCU system is outlined in the API Specifications document on NPCI website as well as Bharat BillPay website.

Please find the below links for accessing above said websites:

a. https://www.bharatbillpay.com/library.php
b. https://www.npci.org.in/bharat-billpay-notified-documents

3.5 Data Storage and Archive

1. Transaction related data should be archived minimum for a period of at least 5 years.
2. Complaints may be raised for ON-US transactions as well. So, transaction related data one year from the date of a transaction should be made available by a BBPOU at any given time for retrieval for raising complaints anywhere in the BBPS Eco system.
3. The data stored in BBPOU databases should be secure, encrypted and in-line with the latest data security and data localization standards.

3.6 Personally Identifiable Information

1. The customer registration process and bill payment in BBPS include a lot of personally identifiable information (PII). It is, therefore, imperative that the BBPOUs comply with two key security aspects.
   a. Compliance to Information Technology Act
   b. PCI DSS Compliance
2. Customer details like mobile number passed from Customer BBPOU to BBPCU should be masked before it is forwarded from BBPCU to Biller BBPOU.
   a. Additional non-mandatory customer details like email, AADHAAR and PAN card details passed from Customer BBPOU to BBPCU have to be proper if a Customer BBPOU wants to forward the same.

3. For any Personally Identifiable information input by user and sent to BBPCU, BBPOU should have taken the consent from the end customer.

3.7 Canvas

BBPCU system will provide the BBPOUs an intranet portal (BBPOU Canvas) through which an authorized BBPOU user can on-board agent institutions, agents, conduct transaction enquiry, download settlement reports, user creation and other such functions. The BBPOU on-boarding process should be completed before an administrator profile for the respective BBPOU is created.

A BBPCU administrator will create two administrator accounts for each BBPOU which will act as parent IDs for creating other users within the BBPOU. There will be a ceiling on the number of users created for a particular BBPOU. The BBPOU users can only access the data pertaining to their own institution. The BBPOU administrator may assign / modify different profiles for the other users as per their organizational / business requirements.

4 BBPS Security Requirements

The Bharat Bill Payment System will deal with the confidential information of the customers. It is thus imperative that the communication channel between the participants is secure and information flow takes place in most secure and encrypted format. Any breach in client confidentiality or data security can have a negative impact on the reputation of BBPS.

The BBPCU solution is PII compliant and it is recommended to BBPOUs to have similar compliance. All XML messages will be signed and sensitive data, if stored, will be encrypted. There will be appropriate access control and authentication mechanism for all users of BBPS.

4.1 Data Security in Motion

BBPCU will only be able to communicate with BBPOUs and Participating Agent Institutions that are part of BBPS Eco System. There would not be any direct communication between BBPCU, agent, non-participating agent institution and biller. BBPCU will act as a server whereas BBPOUs will act as a client.

During the transaction processing, when a message is getting transmitted between BBPOU and BBPCU, the receiver of message needs assurance that the message has indeed been originated by the sender with public key of BBPCU SSL certificate and the latter should not be able to repudiate the origination of that message. This requirement is very crucial in BBPCU's secured processing environment with Private Key of BBPCU SSL certificate to obviate disputes over exchanged data.

Hence it has been decided to use the standard Digital signing to ensure the integrity. Digital signature is a cryptographic value that is calculated from the data and a secret key known only to the signer. Digital signature binds the BBPOU entity to the digital data. This binding can be independently verified by the receiving entity.

4.1.1Digital Certificate

Digital certificates will be used to ensure the trustworthiness of public facing portals and websites of BBPCU. The digital certificate will contain a unique identifier for the entity, and also include the certificate authority that verifies the information contained in the certificate, date that the certificate is valid from and the date that the certificate expires.

The payload has to be digitally signed using SHA2 with RSA (RSA 2048 bits' key) and the signature has to be embedded in the XML payload which will then be transmitted through a secure SSL channel.

A BBPOU has to obtain the following certificates for on-boarding into the BBPS Eco System:

S. No.	Requirement	Key Length	Certificate Type
1	SSL	RSA 2048	Class 3
2	Digital Signing	RSA 2048	Class 3

BBPCU may provide a provision in its portal under "Certificate Management" to upload the Certificates for signed XML exchange. At this point of time certificates will be uploaded and remain static. However, going forward BBPOUs must keep provision for dynamic certificate exchange with BBPCU in real time basis.

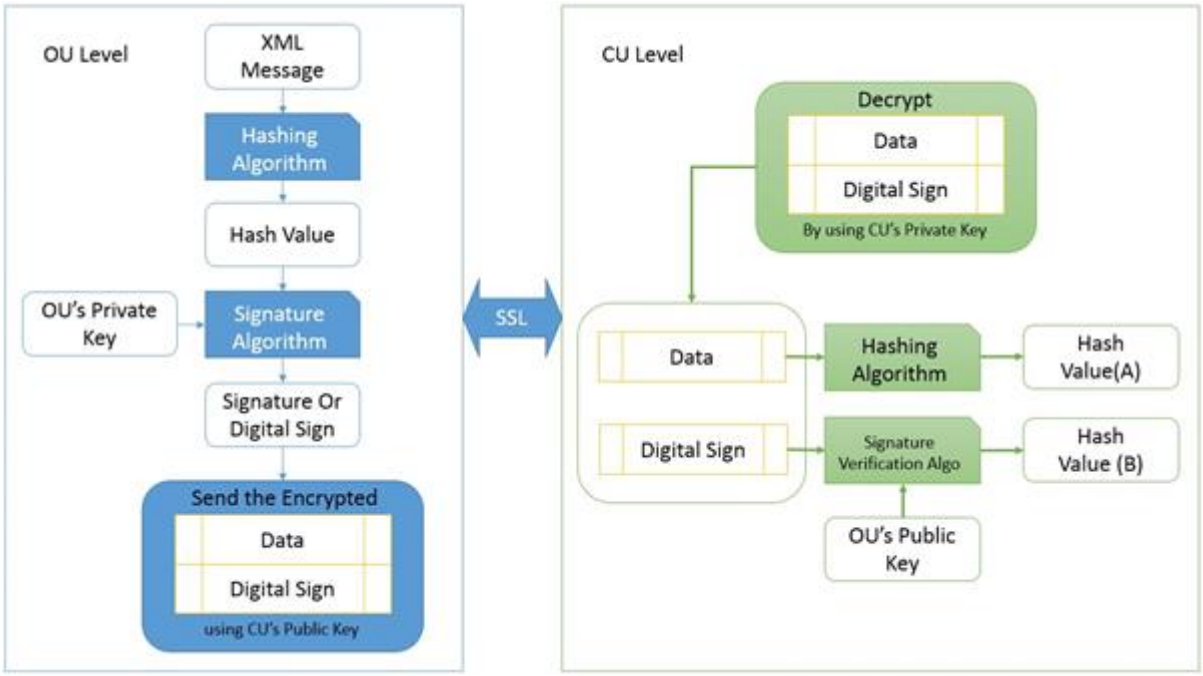
BBPOUs application should be technically capable to configure both old and new BBPCU certificates at their end during the certificate change activity so the renewal is seamless. There should not be any downtime at BBPOU level, and member OU's should be able to process transactions successfully from the new certificate once the BBPCU switches to the new BBPCU certificate.

4.1.2 Transport Layer Security

1. All REST API messages will be exchanged over TLS (v1.2), i.e. HTTPS.
2. All file exchanges will be over HTTPS.
3. All web pages will be exposed over HTTPS.
4. All batch files will be shared over HTTPS.
5. The cipher suite selected by the server from the client's cipher suites and revealed in the Server Hello message is carried out during the TLS Handshake.

4.1.3Message Security and Non-Repudiation

Digital Signature Implementation Model



The following steps indicate how a digital signature is used for validation of an entity:

1. Signer and Verifier should have its public-private key pair. The private key used for signing is referred to as the signature key and the public key as the verification key.
2. Signer feeds data to the hash function and generates hash of data.
3. Hash value and signature key (Signer's Private Key) are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the XML message (data) and then both are sent to the verifier using SSL.
4. Verifier feeds the digital signature and the verification key (Signer's public Key) into the verification algorithm. The verification algorithm gives some hash value as output.
5. Verifier also runs same hash function on received data to generate hash value.
6. For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.

Note:

1. SHA256 algorithm is used for hashing and 2-way SSL is followed using 2048-bit compression.
2. When a TLS connection is established, a handshaking, known as the TLS Handshake Protocol, occurs where a client hello (Client Hello) and a server hello (Server Hello) message are passed. First, the client sends a list of the cipher suites that it supports, in order of preference. Then the server replies with the cipher suite that it has selected from the client's list. To test which TLS ciphers a server supports, an SSL/TLS Scanner may be used.
3. There should be no \n and \r formatting applied by the signature utility in the final XML file sent.
4. TLS v1.2 is the protocol for all API message exchanges.
5. The BBPOU application should preferably select AES_128_SHA cipher suite during the SSL Handshake for receiving a request or response from BBPCU.

4.2 Data Security at Rest

1. All sensitive data (data at rest) will be encrypted and stored.
2. All private keys will be stored in HSM (FIPS compliance).
3. Public keys (certificates) will be stored in DB.

4.3 Guidelines for Password Policy

The password policy implementation on the BBPOU system should be configurable and broadly comply with the guidelines provided below:

1. Users should be able to reset and change their own passwords.
2. The system should have a provision for password expiry and password history policy should be applicable.
3. System components should enforce complex passwords – a combination of upper case, lower case, numeric and special characters.
4. The account should be locked out after a certain number of failed password attempts.
5. Locked out accounts should be enabled automatically after a cool down period.
6. A few days prior to password expiry the user should be alerted by a warning message to change the password on every login.
7. System and applications should verify the user's old password before allowing the user to set a new password.
8. Passwords should be encrypted when transmitted across any network.
9. The display and printing of passwords must be masked using asterisks, or otherwise obscured such that unauthorized parties will not be able to observe or subsequently recover them.
10. Passwords must not be logged or captured.
11. A password must not be displayed on the data entry or display device.
12. Temporary unique passwords should be assigned when creating new accounts and on user's request for passwords resets.
13. User should be forced to change password at first login.
14. When changing a password, the user should be prompted to re-enter the new password for verification.
15. On successful password reset, a password will be generated automatically through the system and the following will happen:
    - The new password will be communicated to the user through SMS and / or Email.
    - After password reset in the system, login will be considered as a first time login and the user will be asked to change password immediately after logging in.

5 API Messages

The BBPCU system has been designed keeping in mind easy integration with prospective BBPOUs. The messages have been designed in native XML so that it is can be easily integrated. The XML structure and tag names have been kept to ensure relevance to specific bill payment transactions and have been verified and vetted by key players and external experts. Please refer to our website for latest version of online specifications and XSDs.

The following lists down the considerations pertaining to online API messages:

1. All BBPOUs need to send heartbeat requests to the BBPCU at regular intervals (3 minutes) to ensure that they are available in BBPS.
2. Bill fetch request and response has its own Message ID (msgId), payment request and response has its own Message ID and similarly validation request and response has its own Message ID.
3. Reference ID (refId) binds together the bill fetch/validation and payment messages. In a scenario where bill fetch/validation is mandatory before the payment is made, the refId will validate the information from bill fetch/validation against payment request. In a scenario where the bill fetch/validation is not mandatory, ref ID will still be generated at payment leg.
4. Transaction ID (txnReferenceId) is generated only at the time of initiation of payment request and this is the ID which will be communicated to the customer for subsequent references. The first 4 characters of a txnReferenceId is the Originating Entity Id, from which the transaction is initiated.
5. Bill fetch response message will be signed by the Biller BBPOU and this signed message (Biller Response and Additional Information parameters) should be added in the Payment Request by Customer BBPOU.
6. Biller will only get information that is relevant for posting in their records.
7. List of Error Codes along with the Response Codes and Compliance Response Codes maintained by BBPCU should be adhered to by the BBPOUs. These have been shared with BBPOUs as part of the on-boarding process. The same will be available as an attachment in the BBPS API Specifications.
8. Every leg of the online message should be responded with an acknowledgment (synchronous message) from the receiver of the message. However, if the sender of the message fails to receive this acknowledgement then the failure needs to be handled as per the CU Decline Scenarios shared with the BBPOUs.
9. In fetch and payment response messages the compliance response code and reason (complianceRespCd) are the attributes which will be populated as per the Decline Scenarios.
10. There will be no case sensitivity at any point in the messages where alphabetical characters are involved, e.g. both "ABCDE1234F" and "abcde1234f" will be accepted for a customer's PAN.
11. All monetary values will be expressed in paise, i.e. a value of Rs 100 will be represented as "10000".
12. Currency codes will be represented through a Code set that follows the ISO 4217 convention for representation of currencies. For instance, Indian Rupee (INR) is represented by the standard three digit code '356'.
13. All date and time values follow the ISODateTime standard YYYY-MM-DDThh:mm:ss+/-hh:mm, where ' T ' represents a separator between date and time.
14. Timestamp of each message should be at the time of creating the message (e.g. Reversal message). Timestamp of each message should be as follows:
    - **Header timestamp** - At the time of creating the message by each entity and thus will constantly change.
    - **Txn Tag timestamp** - Constant and should be same throughout the message.
    - **Analytics timestamp** - Constant and should be same throughout the message.
15. The BBPOUs have to strictly adhere to the BBPCU published XML message construct / standards and have to comply where mandatory, conditional or optional elements are required along with the data type and format.
16. Payment Information parameters for payment must be passed on alternatively by the Customer BBPOU:
    - CardNum|AuthCode: value="<<Last 4 digits or masked card number>|<Card>"
      e.g. <Tag name="CardNum|AuthCode" value="8625|Card"/>
    - IFSC|AccountNo: value="<PG reference number>|<PG>"
      e.g. <Tag name="IFSC|AccountNo" value="SRAN0000341|PG"/>

17. A reversal is initiated only for a payment message and the different failure scenarios, both for fetch or payment have been listed down in the CU Decline Scenarios shared with BBPOUs as part of the on-boarding process. The same will be available in BBPS API Specification document.
18. The receiver of a Request and Response message (apart from a Diagnostic Request and Response) should mandatorily provide an acknowledgement message to sender for completing that particular leg of the transaction cycle. This acknowledgement message is a synchronous message from the receiver to the sender.
19. Acceptance of different payment modes at various initiating channels are outlined in the API Specifications document on NPCI website.
20. The Customer BBPOU needs to ensure that a payment is initiated for a Biller supporting Fetch and Pay only if the response code in the Bill Fetch Response is '000' (denoting a successful fetch response). The Biller BBPOU also needs to verify the same before processing the Bill Payment Request.
21. The Customer BBPOU needs to ensure that the Biller Response block passed on as part of the Bill Fetch Response by the Biller BBPOU is not tampered when being embedded in the Bill Payment Request. The Biller BBPOU also needs to verify the same before processing the Bill Payment Request.

6 Clearing and Settlement

BBPS clearing and settlement system is a three party module consisting of BBPCOU (Customer BBPOU), BBPCU and BBPBOU (Biller BBPOU). BBPCU clearing and settlement is operational throughout the year. It extracts or collects the authorized transactions from the core application server (transaction server) transmitted by the BBPOUs and based on that, processes the transactions in clearing and settlement module and prepares a settlement file which will be uploaded into the Canvas for the BBPOUs to download.

6.1 Settlement File Transfer Mechanism

The BBPCU system sends the following files to BBPOUs as part of every settlement cycle, which typically happens multiple times on every working day:

1. **RAW File** – Will be generated for OU & Participating AI if transactions available for Settlement (XML format)
2. **Daily Settlement Report** – Will be generated for OU & Participating AI (CSV & PDF formats)
3. **Daily GST Report** – Will be generated for OU (CSV & PDF formats)
4. **Participating AI wise Aggregated Daily Report** – Will be generated for OU (CSV & PDF formats)

The BBPCU also shares Monthly reports once a month with the BBPOUs:

1. **Account Ledger** – Will be generated for OU (CSV & PDF formats)
2. **Direct AI wise Aggregated Monthly Report** – Will be generated for OU (CSV & PDF formats)
3. **Monthly GST Report** – Will be generated for OU (CSV & PDF formats)
4. **Monthly Invoice Report** – Will be generated for OU (CSV & PDF formats)

All files are securely shared over HTTPS. To be able to receive these files, BBPOU systems must expose the below APIs. Assuming a BBPOU name as OU12, and base URL of BBPOU system as https://a.b.c.d/context

URL	Files Received	MIME Type	Method
https://a.b.c.d/context/csv	**Settlement Report:** Example: 07001OU122016080400.csv	multipart/form-data	POST

	Account Ledger: Example: 85201607OU122016080400.csv		
https://a.b.c.d/context/pdf	**Settlement Report:** Example: 07001OU122016080400.pdf **Account Ledger:** Example: 85201607OU122016080400.pdf		
https://a.b.c.d/context/txt	**Raw Data File:** Example: 00001OU132016080400.xml This file contains all transactions for the settlement cycle in XML format, and the XML is signed		

All APIs are expected to return an OK (200) status on successfully receiving a file.

Note: The example mentioned in the file transfer process mentions a nomenclature **https://a.b.c.d/context** (context is just indicative) and this in its entirety is assumed to be the endpoint URL. This is the same endpoint URL which BBPOUs provide for transmission of online messages but in the form **https://<IP_Address>:<Port_No>**. And it is the same location where the BBPOUs have deployed their respective BBPOU application.

7 BBPOU Certification Process

There are three environments / stages that BBPOU should come across to complete the BBPS certification process namely:

- **Sandbox Test environment**: To verify the necessary testing requirements
- **Comfort Environment**: To test and verify the possible structural / logical and compliance parameters
- **Certification Environment**: To certify before integrating into the production line

BBPS Sandbox environment is a testing environment for BBPOUs. The Sandbox environment will help BBPOUs to check their developed application and test their systems readiness at their end. Sandbox supports the Message Dumps provided as part of the Sandbox Test Cases. It is a mini replica of the BBPCU production system which will use a self-signed certificate. Simulated test data is used for Sandbox Testing, however, it is not recommended to perform any load testing in such an environment. BBPS will provide the IP and Port Details to establish a handshake and enable the connectivity.

On completion of the Sandbox round, the BBPOUs proceed to the Comfort round where they get to execute a set of Comfort Test Cases. On successful completion of the Comfort round, the BBPOUs will be eligible for the Certification round (a.k.a. UAT Round) where they get to execute the test cases from their production ready system and Designated Front end channels. However, during the Certification round these test cases need to be executed at a stretch by the BBPOUs during a defined time slot, with a first time right criteria. A BBPOU will only be certified on successful execution of online test cases and validation of the respective offline reports and frontend channels.

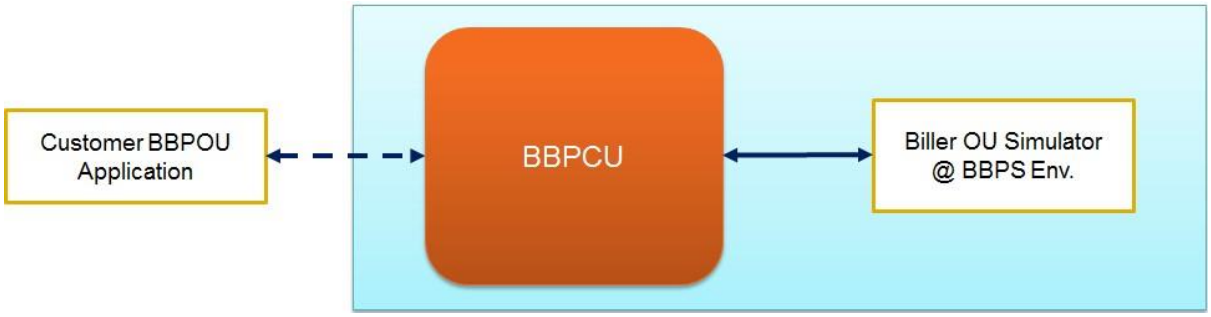
7.1 Transaction Flow

7.1.1Sending Transactions from Customer BBPOU

Customer BBPOU can send the transaction requests from its own system to BBPS test environment.

1. BBPCOU (Customer BBPOU) system sends the API transaction request to BBPCU.

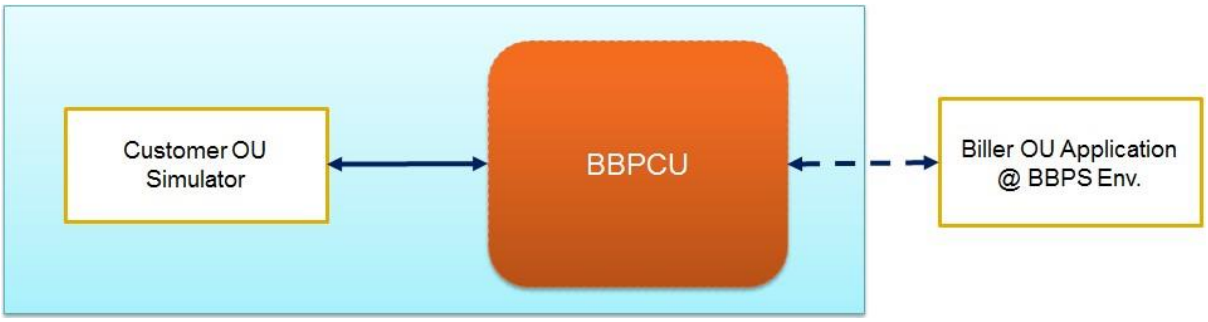
2. BBPCU routes the transaction request to BBPBOU (Biller BBPOU) simulator installed at BBPCU test environment.
3. BBPBOU (Biller BBPOU) Simulator responds back to BBPCU.
4. BBPCU forwards the response to BBPCOU (Customer BBPOU) system to complete the transaction life cycle.



7.1.2Sending Transactions from Biller BBPOU

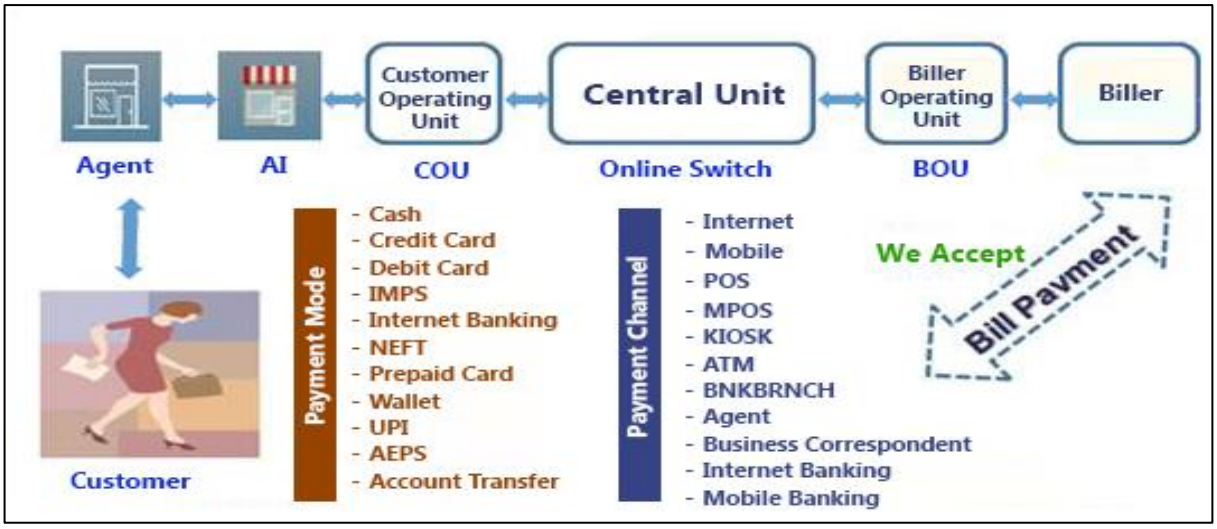
BBPBOU (Biller BBPOU) can send the transaction responses from its own system to BBPCU test environment, mimics' as if it comes from a biller.

1. BBPCOU (Customer BBPOU) Simulator at BBPCU test environment sends the pre-defined API transaction request to BBPCU.
2. BBPCU routes the transaction request to BBPBOU (Biller BBPOU) system.
3. BBPBO (Biller BBPOU) system responds back to BBPCU.
4. BBPCU forwards the response to BBPCOU (Customer BBPOU) simulator to complete the transaction life cycle.



7.1.3Sending Transactions from Biller BBPOU

A pictorial representation of the end-to-end transaction flow in a typical Bill payment cycle:



7.2 Pre-Requisites

Following are the pre-requisites for using the BBPS Sandbox environment:

1. BBPOUs should have a clear understanding of the XML constructs and the different combination of parameters in an online message, e.g., mandatory, conditional and optional elements, convenience fee calculation, amount block variations, etc.
2. BBPOUs have to develop BBPCU compatible system as per the BBPS API specifications and compliance parameters. They are required to validate the field values in their application in line with BBPS API Specifications document. Ideally there should be no rejection of XML messages received by BBPCU from BBPOUs owing to structural validation failures at BBPCU.
3. Communication will be over HTTPS using NPCINET. NPCI will expose the SSL endpoint for communication with BBPCU. BBPOUs need to give information about their SSL endpoint.
4. NPCI will provide 2048 bit RSA SSL Certificates to BBPOUs for connecting to the BBPCU sandbox environment. Similarly, BBPOUs need to provide their 2048 bit RSA SSL certificate to BBPCU. This is required for SSL communication between a BBPOU and BBPCU.
5. All messages will be digitally signed and BBPOUs need to provide their certificate to BBPCU prior to message communication. The certificate should be based on RSA 2048 bit keys.
6. BBPOUs are required to share their NPCINET or Public static IP details which will be whitelisted at BBPCU end. Similarly, BBPOUs are requested to enable connection at their end by whitelisting the IPs shared by BBPCU.
7. All the BBPOUs must provide a single bidirectional IP and port number for both incoming and outgoing messages. Routing the BBPS traffic through Two separate IP's will not be entertained.
8. BBPOUs must provide their priority (as Customer BBPOU or Biller BBPOU or BOTH) well in advance for a smooth certification process, since each entity requires different certification approach.
9. There is a separate test case coverage and execution and certification process for Customer BBPOU / Biller BBPOU, but it can occur in parallel.
10. BBPOUs and BBPCU need to share the Signer and SSL certificates only in .cer or .crt format with at least 1-year validity.
11. While a self-signed certificate (with a validity of at least a year) is acceptable during the certification environment, a CA signed certificate must be ensured in production environments.
12. BBPOUs need to implement the Heart-beat API to make successful communication with BBPCU.

7.3 Considerations

1. The test cases for Sandbox round will be shared with the BBPOUs after whitelisting of the respective IPs and exchange of Signer and SSL certificates.
2. Proper message dumps should be shared with the BBPCU Certification team by the BBPOUs before commencing the sandbox or comfort testing in order to ensure that their application is capable of generating the desired XML messages.
3. Front end channel availability and activation should be done for the payment channels the BBPOUs have opted for and have been duly certified. The front end screenshots need to be shared by the BBPOUs with the BBPS team.
4. Similarly, the test cases for Comfort round will be shared with the BBPOUs after successful completion of the Sandbox round, whitelisting of the respective IPs and exchange of Signer and SSL certificates.
5. In order to expedite the Certification round, the test cases need to be executed in batches instead of running them one after the other. The BBPOU application should have the capability of firing test cases in bulk.
6. The Production BBPOU IP / Port, MDM and Certificates should be shared well in advance during the comfort round to ensure a smooth integration to the production environment.
7. The test execution results and the log files shared by the BBPOUs during the certification process will be used by the certification team to validate the execution of the test cases and also serve as an audit trail.
8. After successful completion of the Certification round the BBPOU configuration parameters need to be shared in a desired format for populating in production environment.
9. Any major changes in BBPOU or BBPCU application requires re-certification for the respective BBPOU. This exercise is carried out to ensure stability of the BBPS Eco System. The BBPOU needs to reach out to BBPCU teams to clarify whenever a change induced in their application needs to undergo re-certification or not.
10. All REST messages will be exchanged over TLS (v1.2), i.e. HTTPS and to test which TLS ciphers a server supports, an SSL / TLS Scanner may be used. The BBPOU application should preferably select TLS_RSA_WITH_AES_128_GCM_SHA256 cipher suite during the SSL Handshake for receiving a request or response from BBPCU.
11. There should be no '\n' and '\r' formatting applied by the signature utility in the final XML file sent.
12. Apart from sharing IP and port details, BBPOUs may also provide their 'context root', where the BBPOU application may be deployed for easier application maintenance.
13. The last 3 digits of the mobile number should be same as Test Case ID for the comfort and certification fetch and payment test cases. This is only applicable during the certification process and not in a production environment.
14. BBPOU applications should strictly follow the BBPCU message standards and any deviation from it will result in decline at CU level.
15. BBPOUs should be flexible and adaptable in order to simulate negative, reversal or exceptional scenarios for a better test coverage.

8 BBPOU Scheduled Activities and Downtime

The Bharat Bill Payment System (BBPS) is an on-line application designed to independently support the Customer and Biller Functions of the Bill inquiry and Bill payment process. BBPS interfaces with BBPCOU and BBPBOU networks in a real-time update environment. This means the system is in contact with banking and non-banking BBPOUs virtually 24 hours a day, 7 days a week. This provides:

- Guaranteed transaction delivery
- Immediate response to bill information inquiries
- Immediate processing of bill payment transactions
- Support for both Customer and Biller processing

As the BBPS eco-system allows users to pay their bills and billers to accept the payments round the clock, all participants must strive to ensure maximum uptime for the BBPS application from their respective Operating Units (BBPOUs). In case if there are any scheduled activities, BBPOUs must have a Business Continuity Plan (BCP) in place to ensure that the business continues in an uninterrupted manner. In case there is a scheduled activity resulting in system downtime, such a planned downtime should be as minimal as possible. All such exceptions should be planned well in advance (minimum 2 ~ 4 weeks) and executed in a manner so that the impact is as minimal as possible for all the participating members and bill paying customers and entities. All such events should be informed well in advance and approved by the BBPCU, prior to the actual scheduled events. Such approved downtimes need to be informed to all the BBPS participants, including agents and billers, well in advance to plan accordingly.

1. Planned downtime should be avoided by all means during the normal business hours, and such planned downtime should not exceed 2 hours.
2. Scheduled downtimes may be planned during 2nd or 4th Saturdays in between 02:00 to 04:00 AM.
3. According to the standards, all BBPOUs must ensure an uptime (systems) of at least 98.5% on a monthly basis.
4. BBPOU Systems should be designed in such a manner, that a planned downtime of COU application of a BBPOU should not affect their respective BOU business and operations in any manner and vice versa.
5. BBPOU Technical, IT and Network teams should be made available during the entire course of such a scheduled downtime for verifications prior to commencement of the scheduled window and after completion of the scheduled window.
6. If any BBPOU is acting as a Technical Service Provider (TSP) for other BBPOUs, the BBPOU's (BBPCOU / BBPBOU) application downtime should not affect / impact the uptime of any of the other serviced BBPOUs.

9 Biller Types in BBPS

The following table outlines the various Biller integration scenarios in BBPS. These fields are passed on as Biller MDM Response parameters.

S. No.	Type	Accepts Ad-hoc	Fetch Requirement	Support Validation	QuickPay value in Pay Request	Transaction
1	ONLINE	T	OPTIONAL	-	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).
			OPTIONAL	-	▪ No	▪ Payment against fetch can also be done for any amount.
2	ONLINE	T	NOT_SUPPORTED	OPTIONAL	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).
			NOT_SUPPORTED	OPTIONAL	▪ No	▪ Payment against validation can also be done for any amount.
3	ONLINE	T	NOT_SUPPORTED	MANDATORY	▪ No	▪ Payment against validation can also be done for any amount.
4	ONLINE	T	NOT_SUPPORTED	NOT_SUPPORTED	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).
5	ONLINE	T	MANDATORY	-	▪ No	▪ Ad-hoc Payment cannot be done. Payment against fetch can be done for any amount.
6	ONLINE	F	MANDATORY	-	▪ No	▪ Ad-hoc Payment cannot be done. EXACT, EXACT_UP, EXACT_DOWN can be paid against fetched bill as per configuration.
7	OFFLINE A	T	OPTIONAL	-	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).
			OPTIONAL	-	▪ No	▪ Payment against fetch can also be done for any amount.
8	OFFLINE A	T	NOT_SUPPORTED	OPTIONAL	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).
			NOT_SUPPORTED	OPTIONAL	▪ No	▪ Payment against validation can also be done for any amount.
9	OFFLINE A	T	NOT_SUPPORTED	MANDATORY	▪ No	▪ Payment against validation can also be done for any amount.
10	OFFLINE A	T	NOT_SUPPORTED	NOT_SUPPORTED	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).
11	OFFLINE A	T	MANDATORY	-	▪ No	▪ Ad-hoc Payment cannot be done. Payment against fetch can be done for any amount.
12	OFFLINE A	F	MANDATORY	-	▪ No	▪ Ad-hoc Payment cannot be done. EXACT, EXACT_UP, EXACT_DOWN can be paid against fetched bill as per configuration.
13	OFFLINE B	T	NOT_SUPPORTED	OPTIONAL	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).
			NOT_SUPPORTED	OPTIONAL	▪ No	▪ Payment against validation can also be done for any amount.
14	OFFLINE B	T	NOT_SUPPORTED	MANDATORY	▪ No	▪ Payment against validation can also be done for any amount.
15	OFFLINE B	T	NOT_SUPPORTED	NOT_SUPPORTED	▪ Yes	▪ Ad-hoc Payment can be done (for any amount).

10 List of Abbreviations

Abbreviation	Full Form
API	Application Program Interface
BBPCU	Bharat Bill Payment Central Unit
BBPOU	Bharat Bill Payment Operating Unit
BBPCOU	Bharat Bill Payment Customer Operating Unit
BBPBOU	Bharat Bill Payment Biller Operating Unit
BBPS	Bharat Bill Payment System
CA	Certificate Authority
CSV	Comma Separated Values
DB	Database
DMS	Dispute Management System
FIPS	Federal Information Processing Standards
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
ID	Identity
IP	Internet Protocol
ISO	International Organization for Standardization
Mbps	Mega Bits Per Second
MDM	Master Data Management
NPCI	National Payments Corporation of India
NTP	Network Time Protocol
PAN	Permanent Account Number
PCI-DSS	Payment Card Industry Data Security Standard
PDF	Portable Document Format
PII	Personal Identification Information
POST	A common HTTP Request Method
RBI	Reserve Bank of India
REST	Representational State Transfer
RSA	Rivest, Shamir, and Adelman
SHA2	Secure Hash Algorithm 2
SMS	Short Message Service
SPOC	Single Point of Contact
SSL	Secure Sockets Layer
TCP / IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
TPS	Transactions Per Second
TSP	Technology Service Provider
URL	Uniform Resource Locator
XML	Extensible Markup Language
XSD	XML Schema Definition