API Integration Document

V 5.0

Bangla Sahayata Kendra (BSK)

and

Departments of Government of West Bengal

By

BSK PMU - Tech Team

Bangla Sahayata Kendra

Upanna, Ground Floor, 325, Sharat Chatterjee Road, Howrah, PIN: 711102

Visit: https://bsk.wb.gov.in/ | Email: info.bsk@wb.gov.in | Call: +91 33 2214 0080

Table of Contents

1. About Bangla Sahayata Kendra (BSK)

Bangla Sahayata Kendras (BSKs) set up under State Government Memorandum No. 352-CS/2020 dated 14.10.2020, across the State 'to provide government services free of cost at the grassroot level through online mode' and also to strengthen the existing system of information dissemination about various social and development schemes. The BSKs are located in the offices of District Magistrates, Sub-Divisional Officers, Block Development Officers, Health Centres, Government Aided Libraries, Office of the SI of Schools and all Urban Local Bodies (ULBs). The Personnel & Administrative Reforms (PAR) and e-Governance Department of the Government of West Bengal is the Nodal Department coordinating BSK project. There is a Project Management Unit (PMU) at the State level looking after day-to-day functioning of BSKs. All the notified services of BSKs are provided through its online BSK portal https://bsk.wb.gov.in. The major key points

a) The grassroot level services are provided to the citizen at Bangla Sahayata Kendras (BSKs) by Data Entry Operators (DEOs) absolutely free of cost.
b) Each DEO is having unique credential (login/password) on BSK Portal and authorized to process the citizen centric services.
c) Each DEO use up to 25000/- for service-related costs incurred by the citizen through the SBI e-wallet
d) DEO may pay the online fees / bills (electricity bill etc.) of citizen through BSK Bank Account and collect the Cash or online payment from citizen and deposit to the said account.

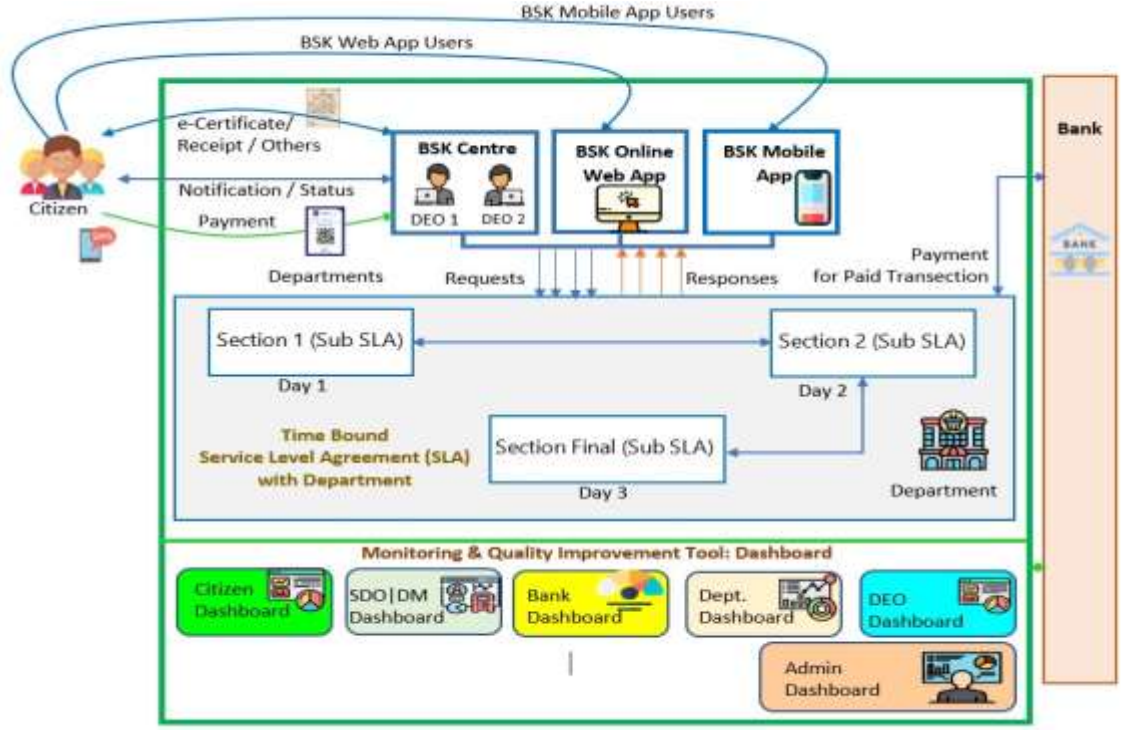

Figure 01: Present Working Model of BSK

2. Objective

The objective of the Application Programming Interface (API) integration is for handshaking between BSK and the departments for sharing of data in a secure way. BSK is a single window citizen delivery platform at the grassroots level. BSK emphases end-to-end delivery model to the citizen. BSK provides service with assistance using the resources of on behalf of the department.

3. Departments

All Departments of the government of West Bengal are mandated to use the BSK portal to provide online services to citizens of the state. The departments are developing and sharing APIs with BSK PMU tech team, onboarding the process to channelize departmental services through the BSK portal. Currently 39 departments are already working with BSK PMU to develop their APIs and use the BSK portal for delivering services at the grassroots.

4. Scope

The scope of the API Integration process is to integrate the Bangla Sahayata Kendra online portal with all the departments of the Government of West Bengal so that citizens can seamlessly get online services from (1) BSK Center, (2) BSK portal, and (3) BSK Mobile App at the doorsteps. In addition, there will be a timeline for each service integration.

1. All the departments of the Government of West Bengal can integrate with BSK Portal
2. Departmental online service(s) only be included with BSK Portal
3. BSK Portal delivers service to the citizen using an End-to-End delivery model
4. Delivery material like e-certificate / receipt / acknowledgment can be available through BSK Portal
5. Each service integration is having Service Level Agreement (SLA)
6. The citizen can get a digitally signed certificate from BSK Center / Portal.

5. Mandate of BSK

a) BSK DEO does not require any login id in Departmental portal.
b) BSKs do not store any data of citizen related to the service provided
c) There will be no Dashboard for BSK-DEO in Departmental portal
d) BSKs serve through its single window online channel for all departmental services.
e) BSK portal will deliver e-services to the citizens. BSKs can download, print documents for the citizens.
f) DEOs of BSKs cannot see the application status or citizen information without the consent of the citizen.

6. Potential Benefits

The API integration will share the data in a secure way between the BSK Portal and Departmental Portal. The API system delivers data and facilitates connectivity between devices and programs by sharing messages and **enabling interaction of data, applications, and devices**. API is also defined as an online programming interface of organizations. It allows applications to communicate with backend systems and create grounds for providing services to the ordinary citizen of the state. The new services launching information is notified through Short Message Services (SMS). The integration process has the following benefits:

a) Public Services available at the doorsteps of people
b) Citizens get complete assistance from a resource person working as Data Entry Operators at the BSK centers
c) Citizens get the assistance of resources like computers, printers, the Internet, etc. at the BSK center
d) The citizen data is entirely secure at the BSK Portal. Also, citizens provide only their basic data.

7. Challenges

The challenges from experiences:

a) The data keyed at BSK centers by the DEOs are based on verbal information from citizens.
b) Mapping of Citizen identity through Application Software using Mobile Number, Ticket Number, Beneficiary Name etc. may be an issue as those may be keyed in differently for same beneficiary in the BSK Portal & the Departmental Portals.
c) No mechanism for standardized application submission acknowledgement
d) Coordination with the department with versatile technology

8. Integration with Department

Bangla Sahayata Kendra (BSK), with 3561 Centres, reaches citizens at the grassroots level across the state of West Bengal with its services. BSK is integrating such services with BSK so that citizen-centric service reaches the people's doorsteps.

BSK Tech team has classified the departments and services into three major categories.

1. **Category A:** The departments with all services online
2. **Category B:** The departments with partial services online
3. **Category C:** The departments with no services online

The following flow diagram will identify that what to do by the department for integration:

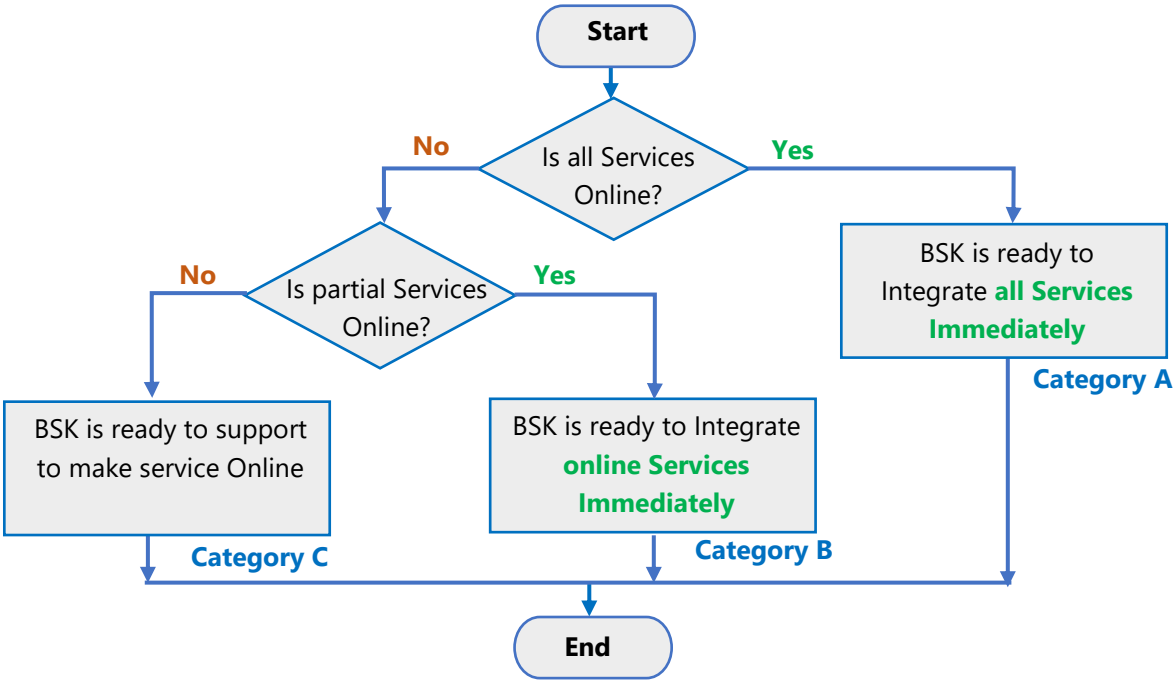

Figure 02: Department Integration Process

9. Top Level View of Application Programming Interface

The top-level view of API integration will show the objective and outcome of the API integration. Based on requirement, more API may be added.

This is hybrid model where encrypted data is transferred from BSK portal to Departmental Portal through post method and there is no requirement of response against this call. After submission of the form (may be draft / final) department will send data to BSK portal with some detail like ticket no, timestamp of form submission, form submission amount (if any) etc.

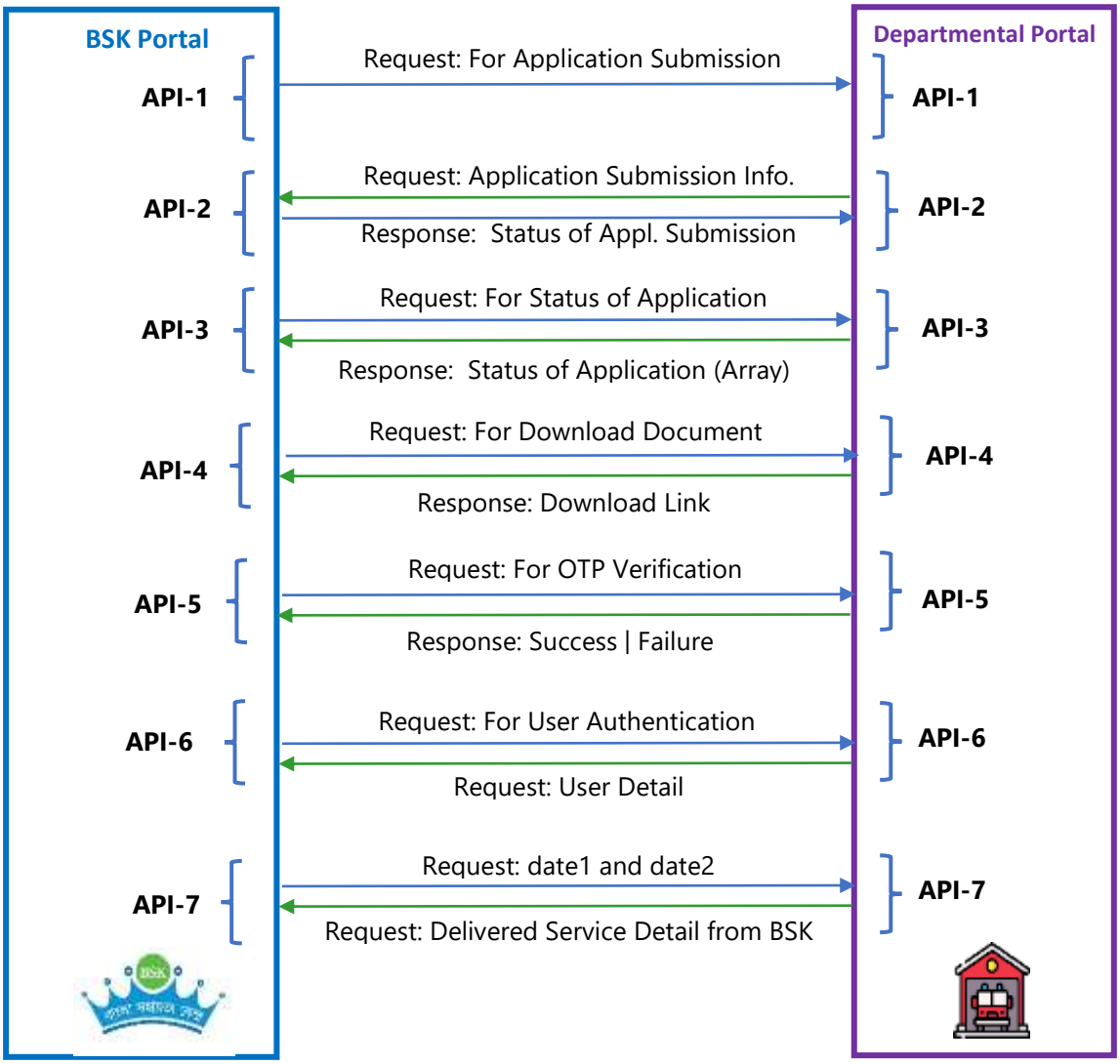

Figure 03: API Integration Overview

10. Integration Data Flow Model

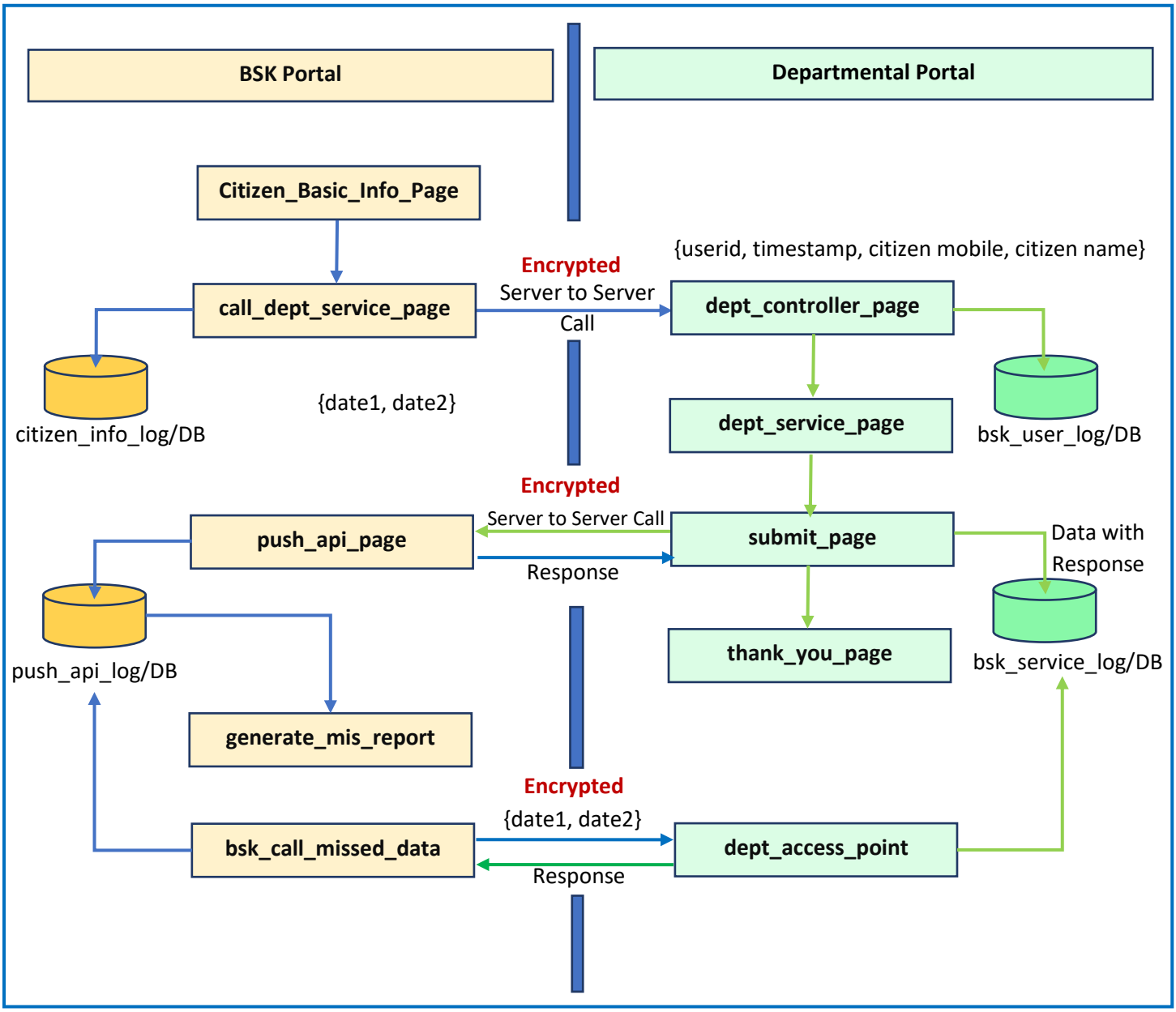

Figure 00: BSK – Department Integration Data Flow Model

Step by step process

- **Step 1:** [BSK Portal] Citizen basic information is entered (client page)
- **Step 2:** [BSK Portal] Data will be forwarded to call_dept_service_page (server page)
- **Step 3:** [BSK Portal] call_dept_service_page has three tasks as follows:
  - (a) Store the data to citizen_info_log / DB
  - (b) Encrypt the JSON data using AES-256-CBC algorithm

- (c) Server to Server call to Department Access Point using POST method

- **Step 4:** [Dept Portal] Dept portal will receive at dept_controller_page. This controller page has three tasks as follows
  - (a) Decrypt the data using AES-256-CBC algorithm in JSON format
  - (b) May Store the data into bsk_user_log/DB
  - (c) Open the Service Page (Application Form etc.)
- **Step 5:** [Dept Portal] dept_service_page will open for fill-up and after draft / final submission, it will call submit_page
- **Step 6:** [Dept Portal] submit_page has FOUR tasks as follows
  - (a) Encrypt the JSON data using AES-256-CBC algorithm
  - (b) Call BSK API (push) with the encrypted data and keep the response received from BSK Portal
  - (c) Store the data into bsk_service_log/DB with the response received from BSK
  - (d) open thank_you_page
- **Step 7:** [BSK Portal] bsk_call_missed_data will call the dept_access_data with from date (date1) and to date (date2) to retrieve the all missed out data between the dates.
- **Step 8:** [Dept Portal] dept_access_point will send the required data with encryption.
- **Step 9:** [BSK Portal] call_bsk_missed_data will receive and update the push_api_log/DB
- **Step 10:** The process ends.

11. API Descriptions

BSK Tech Team designed the API integration architecture with all Government of West Bengal departments to pull or push the data between the BSK Portal and Departmental Portal based on the requirement. This API integration process will never force to change the existing architecture of the departmental software. Instead, the API architecture will work with any architecture and any software. It is one of the most secure methods for sharing data between the BSK Portal and Departmental Portal, maintaining maximum security.

BSK follows a secure and standard process for integration across the departments. There are two values in the JSON object. The first is a secret code, '*passcode*' for authentication of the API sources**.** The department will provide the passcode. And the second one is the data value

in the form of a JWT Token. The JWT Token is secured digitally signed secure code. In a nutshell, BSK uses the following security measure:

A) **Passcode** –the secret code provided by the department
B) **Whitelisted IP** – the request from specific BSK/Department server only
C) **JWT Token** – Digitally Signed Token security

The format of the JSON Object is as follows:

```
{
  "passcode": "376423",
  "token":" eyJhbGciOiJIUzI1N5cCI6IkpXVCJ9.sdfknxcvmxv4.7474udfhrr"
}
```

BSK server will communicate to the department through POST method with multiple objectives and hence BSK Tech Team has designed multiple APIs, which are given below:

11.1. API-1: Application Initiation

API-1 is the mandatory call as it the first communication with the department. The API-1 is designed to provide the data to the department as a *request*. Department will store the data to the department database and send the application submission initial status as response.

- ✓ **Step 1:** When the citizen visits a BSK for service, BSK Portal captures citizen's basic data. BSK Portal then generates a service log with a unique identification which comprises of userId and ticketNo. The userId is the operator mobile no and ticketNo is the timestamp of the service. BSK Portal stores userId and ticketNo along with citizen basic data into the BSK database.
- ✓ **Step 2:** userId is the 10-digit mobile number of DEO (user) and ticketNo is the timestamp (YYYYMMDDHHmmSS) of the service entry in the BSK Portal. Jointly userId and ticketNo is unique.

- ✓ **Step 3:** BSK portal transfers service detail and citizen basic data to the departmental portal as a *request* of API-1 in format of JSON Web Token (JWT). The department has to keep the ticketNo and userId along with citizen service-related information in their database.

- ✓ **Step 4:** The citizen will be redirected to departmental specific service page for application fill up. After redirecting to the application / service page, department

will return the initial status of the application as *response* of the API-1. The task of API-1 completes.

The process flow is given below:

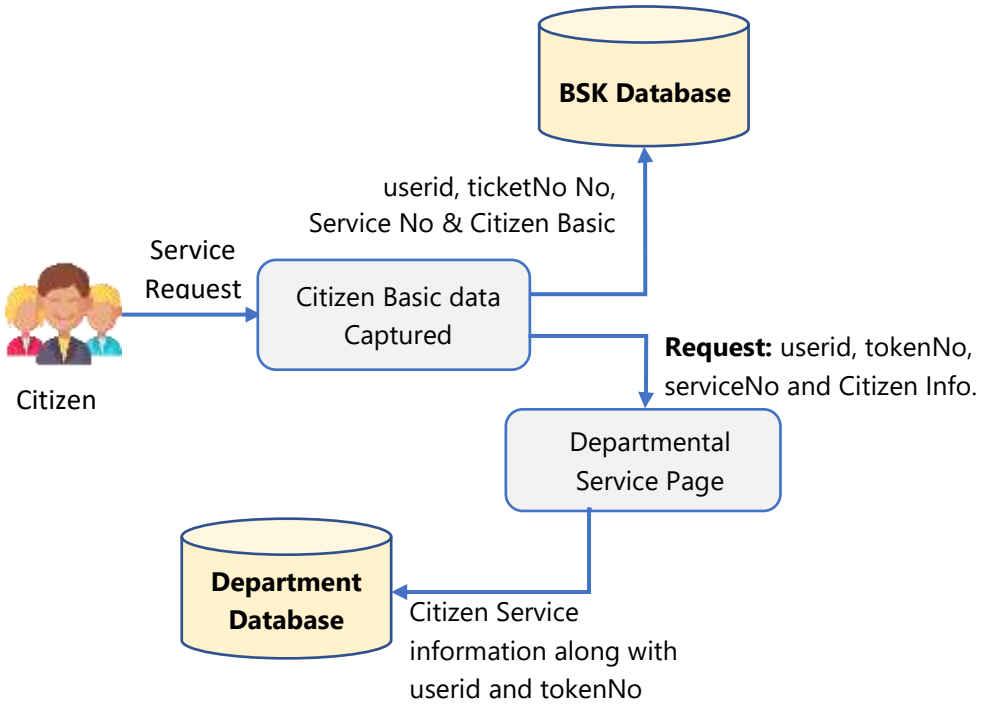

Figure 04: Process Flow of API-1

API-1			
Use Case	The API-1 is called to push the data of user and citizen to the department database as the service has initiated.		
HTTP Request	POST	URL will be provided by the department	
Request Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
ticketNo	Number	-	Timestamp value of the request
serviceCode	String	-	Service code of the department
name	String	-	Name of the Citizen
mobile	String	-	Mobile no of the Citizen
email	String	-	Email of the Citizen

gender	String	-	Gender of the Citizen
age	Number	-	Age of the Citizen

Response Body

Attribute	Type	Value	Description
userId	Number	-	10 digit mobile no of user (DEO)
ticketNo	Number	-	Timestamp value of the request
appSubTime	Number	-	Initial submission timestamp
message	String	"Initiated"	Application Initiation response
applicationStatus	Number	1	The value for application initiation
statusCode	Number	200	Success
		400	Bad Request

Example Request


```
{
    "userid": 9054233544,
    "ticketNo": 20220811130723,
    "serviceCode": "AMD/003",
    "name": "Arindam Ray",
    "mobile": "9350778824",
    "email": "arindam.ray@cmail.com",
    "gender": "male",
    "age": 48
}
```


Example Response

11.2. API-2: Draft | Final Submission of Application

The API-2 is initiated when the application is submitted either in the draft or final version. Once the form is submitted (draft or final) then the API-2 is called by the department for transferring the data to BSK Server. The data will be sent as request and BSK Portal will send the response to that request.

- ✓ **Step 1:** The API-2 is called once the application submitted successfully (draft or final). The transaction information like transaction detail, payment detail (if any) will be transferred to the BSK by the department. And BSK will update the flag of application status (**applicationStatus**) in the BSK portal.

✓ **Step 2:** After successful submission of the application, department will generate a unique application number and communicate to citizen through SMS / Email / WhatsApp along with acknowledgement receipt (this is required to get the application status in future). And hence the task of API-2 completes.

The process flow is given below:

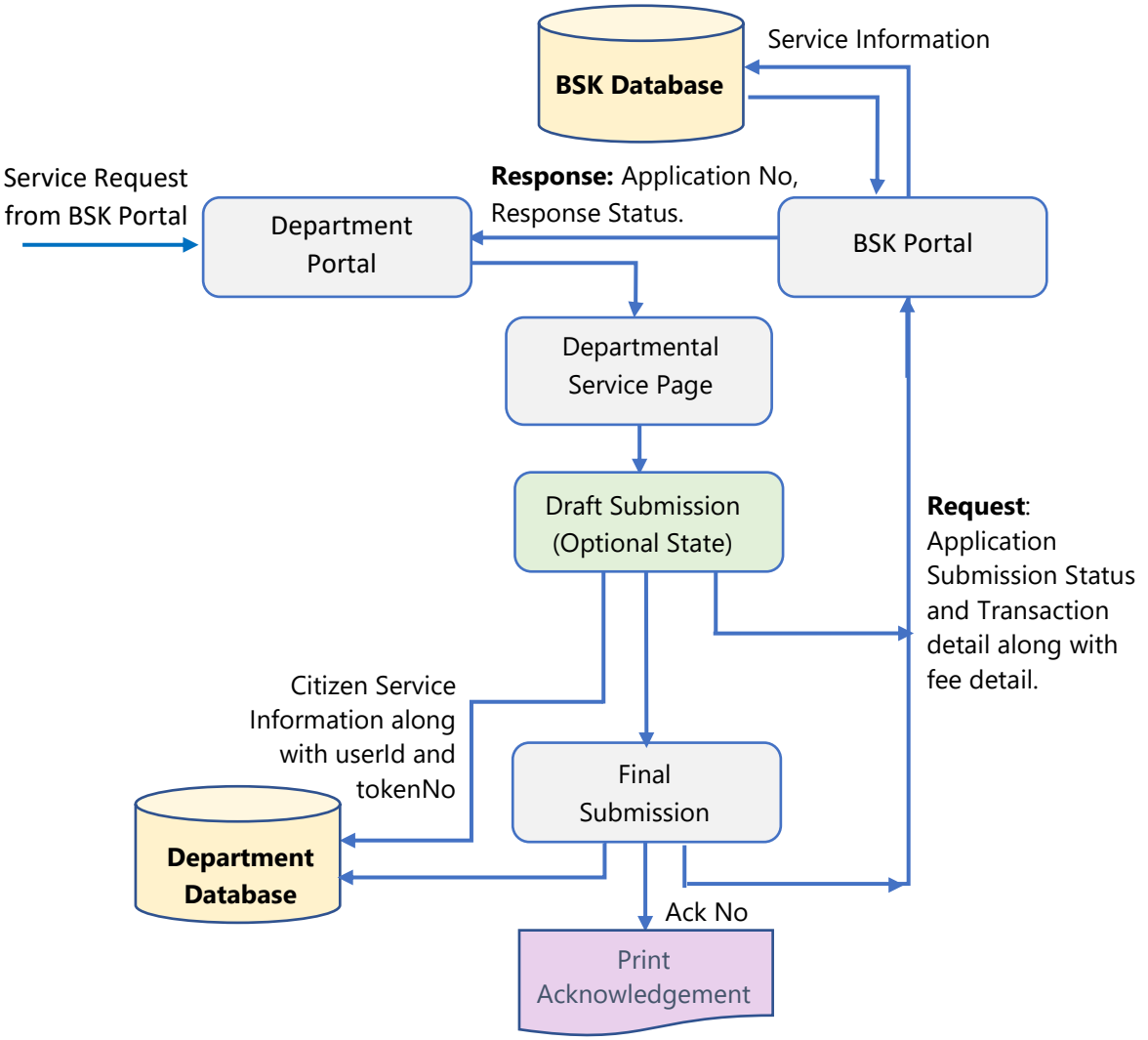

Figure 05: Process Flow of API-2

API-2			
Use Case	API-2 is called by the department once the application is submitted as draft or final submission along with transaction detail.		
HTTP Request	POST	URL will be provided by the department	
Request Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
ticketNo	Number	-	Timestamp value of the request
serviceCode	String	-	Service code of the department

appNo	String	-	Application No of the department
appSubTime	Number	-	Application Submission Time
deptPayRefNo	String	-	Payment Reference No
transNo	String	-	Transaction Number
bankRefNo	String	-	Bank Reference Number
paidAmt	Number	-	Transaction Amount in rupees
message	String	-	"Draft Submitted" | "Final Submitted"
applicationStatus	Number	2	For Draft submission of the application
	Number	3	For Final submission of the application
statusCode	Number	200	Success
	Number	400	Bad Request
Response Body			
Attribute	**Type**	**Value**	**Description**
appNo	String	-	Application No of the department
responseStatus	String	-	"Success" | "Failure"
statusCode	Number	200	Successfully received
	Number	400	Bad Request

Example Request

```
{
    "userid": 9054233544,
    "ticketNo": 20220811050723,
    "serviceCode": "AMD/003",
    "appNo" : "123456",
    " appSubTime " : 20220811052025,
    "deptPayRefNo" : "23424423424",
    "transNo" : "1234353435",
    "bankRefNo" : "4653443656",
    "paidAmt" : 200.00,
    "message": "Draft Submitted",
    "applicationStatus": 2,
    "statusCode": 200
}
```

Example Response

```
{
    "appNo" : "123456",
    "responseStatus": "Success",
    "statusCode": 200
}
```

11.3. API-3: Search Status of the Application

The API-3 is called for finding the status of the application. Each department maintains the service level agreement (SLA) for each service. So, during the service, citizen may search the status of the application and that can be shown through this API-3.

- ✓ **Step 1:** The API-3 is required for finding the status of the application. Citizen can check the status of the application through BSK portal. It retrieves the data from the departmental portal by providing the Application No as *request* in API-3 which will return the application status in an array as the *response*.

- ✓ **Step 2:** The response is an array of data of all the phases (Sub-SLA). The array of data is required to show in citizen dashboard.

- ✓ **Step 3:** After receiving the data as the response, the task of API-3 completes.

The prototype of the interface is as follows:

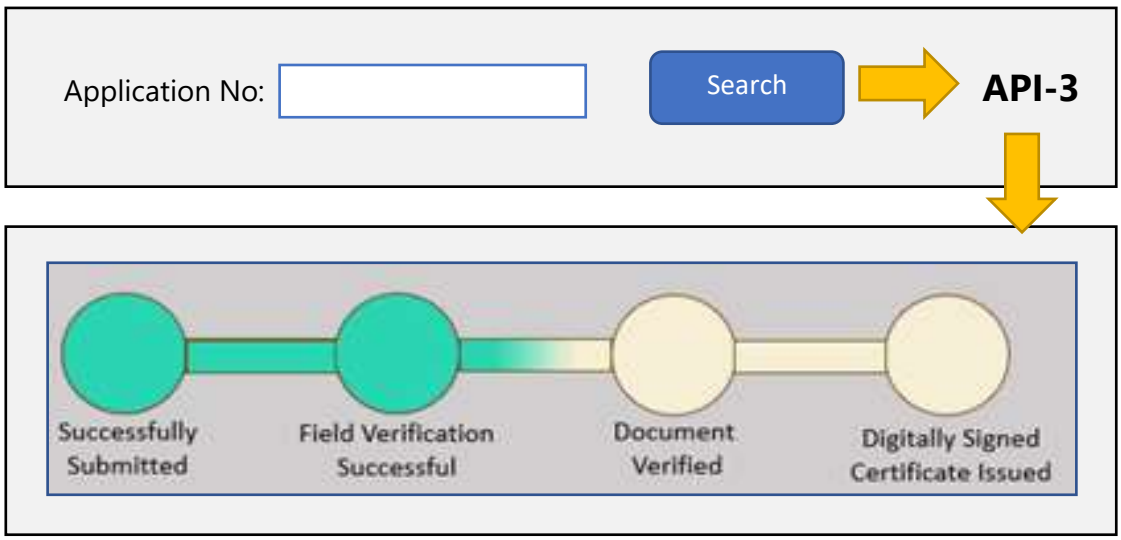

Figure 06: Interface of Status of the Application

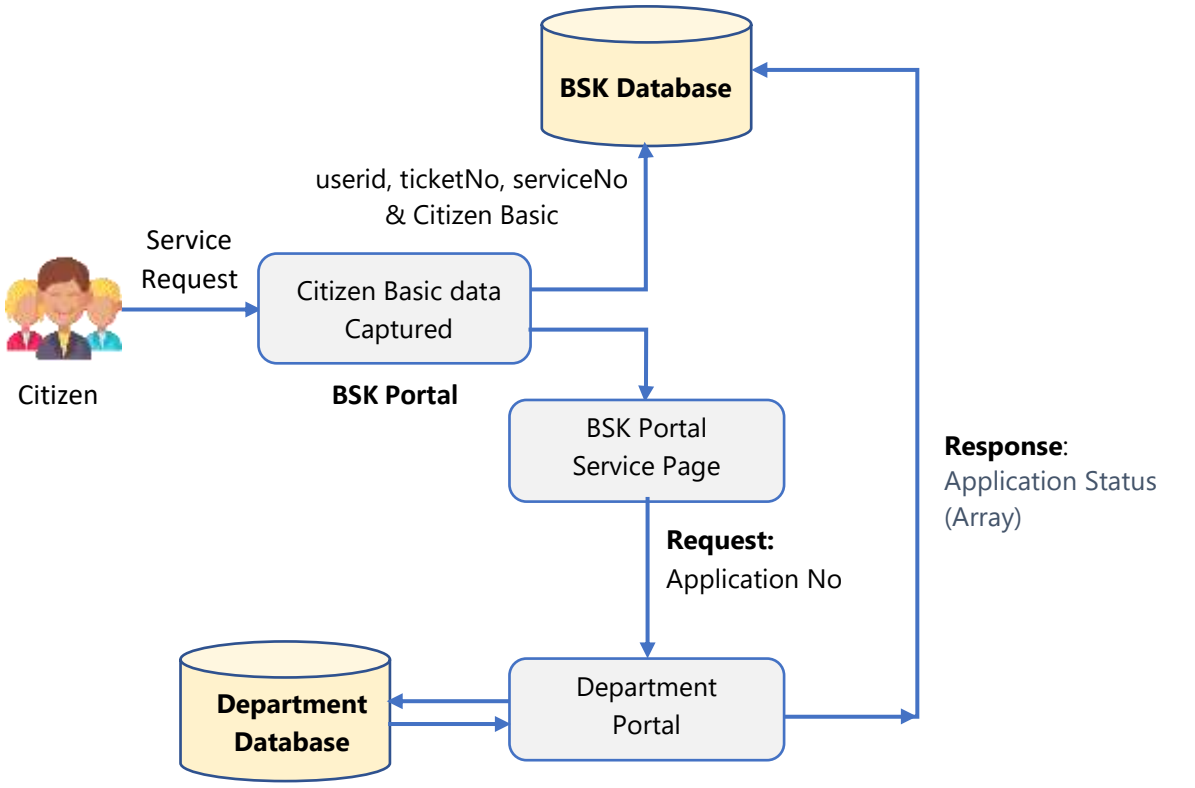

Figure 07: Process Flow of API-3

API-3			
Use Case	API-3 is called for getting the intermediate status of the application. Against the application number it returns the stagewise detail in the form of array and that will be shown in the citizen dashboard.		
HTTP Request	POST	URL will be provided by the department	
Request Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
ticketNo	Number	-	Timestamp value of the request
serviceCode	String	-	Service code of the department. Dept. will provide the exact code. Example: AMD/003
applicationNo	String	-	Application number provided by Department
Response Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
ticketNo	Number	-	Timestamp value of the request
ServiceCode	String	-	Service code of the department.
serviceName	String	-	Service name
applicationStage	Array	-	Array of object of Response Time and Message

applicationStatus	Number	4	For Search the Application status only
statusCode	Number	200	Success

Example Request


```
{
  "userId": 9054233544,
  "tokenNo": 20220811050723,
  "serviceCode": "AMD/003",
  "applicationNo": "AP7373633/003"
}
```


Example Response


```
{
  "userId": 9054233544,
  "tokenNo": 20220811050723,
  "applicationNo": "AP7373633/003"
  "serviceName": "Seed Certificate",
  "ServiceCode": "AMD/003",
   "applicationStage":[
    {
      "responseTime": 20220710052220,
      "message": "Successfully Submitted"
    },
    {
      "responseTime": 20220712022000,
      "message": "Field Verification Successful"
    },
    {
      "responseTime": 20220713140500,
      "message": "Document Verified"
    }
    {
      "responseTime": 20220714100000,
      "message": "Certificate Issued"
    }
   ],
  "applicationStatus": 4,
  "statusCode": 200
  }
```


11.4. API-4: Download Document (Certificate or others)

The API-4 is required to download the document. The document means certificate, receipt, acknowledgement, duplicate document etc. The API-4 will send the application certificate no and pull the document, if available otherwise a suitable message will be shown. The request and response parameters are show below:

- ✓ **Step 1:** The API-4 is for verify and download the document. Citizen can verify or download document from BSK portal by receiving data/link from the departmental portal. BSK portal will provide the Document No as *request* of API-4. The department will return the link of the Certificate/document as the *response*.

✓ **Step 2:** If the certificate is not available then it will send the relevant message in the response and API-4 completes.

The prototype of the interface is as follows:

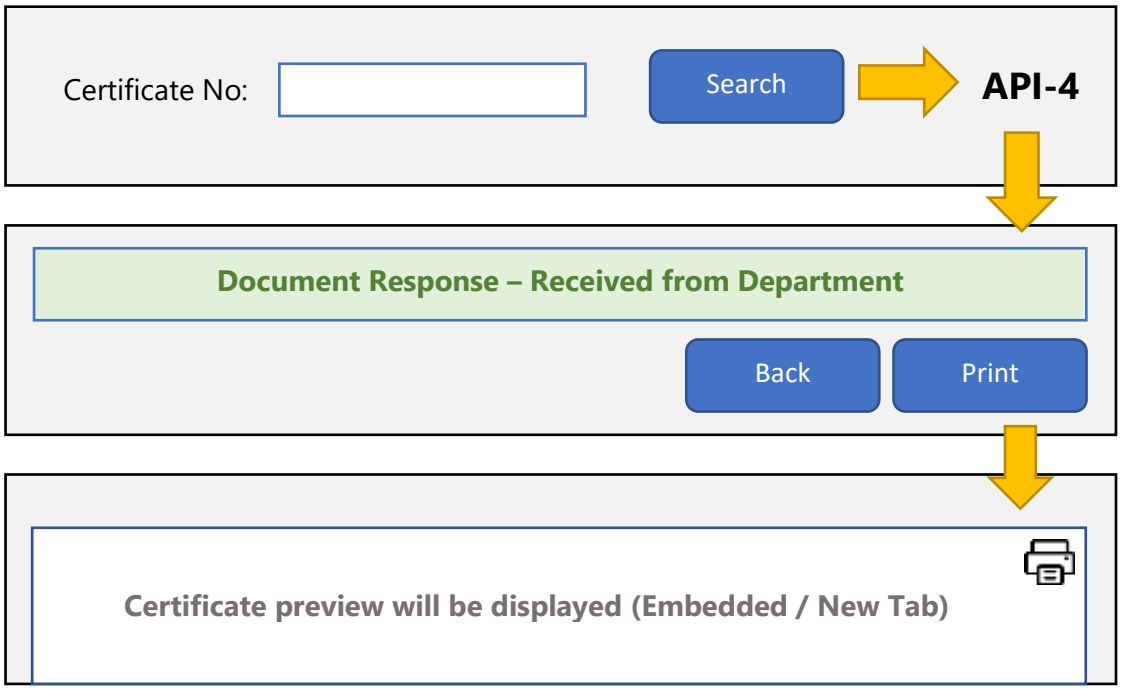

Figure 08: Interface of Download Document

Process follow of Download Certificate / Document is shown below:

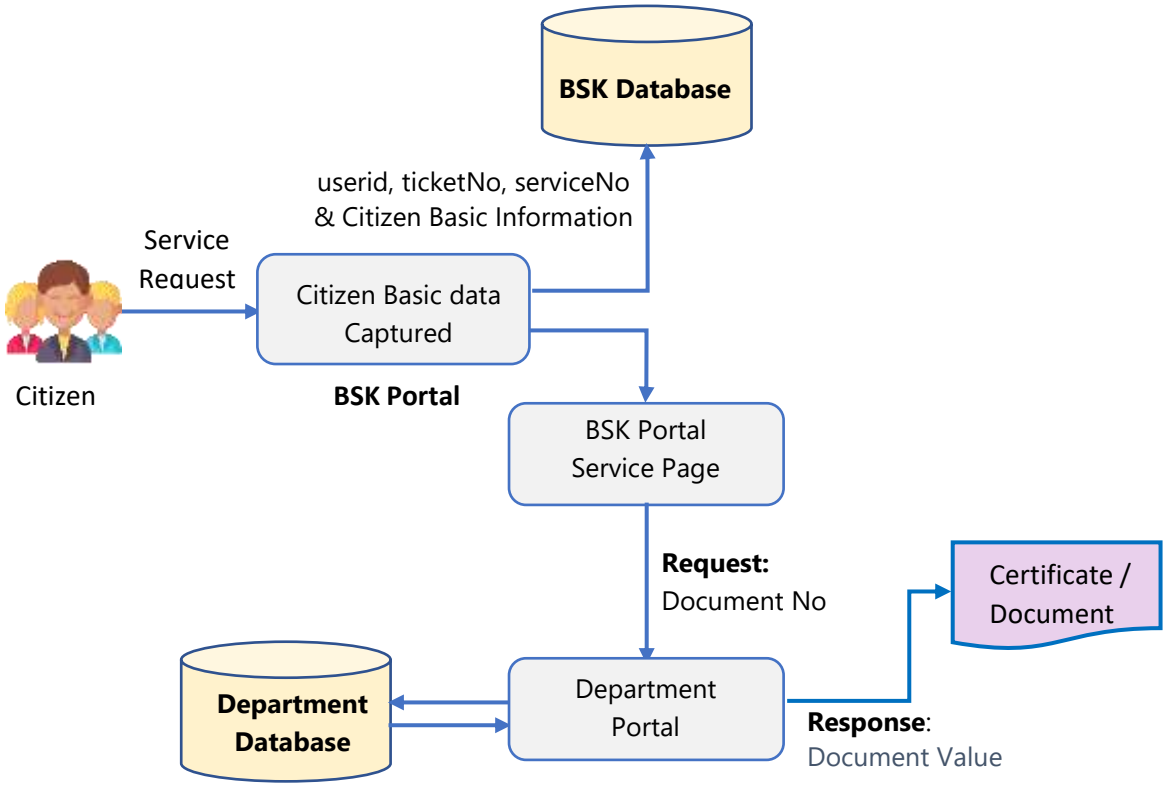

Figure 09: Process Flow of API-4

API-4			
Use Case	API-4 is called to download the document provided by the department. The document may be received as link (URL) or base64 format or image etc. The API-4 can handle all of the formats but preferable is link (URL).		
HTTP Request	POST	URL will be provided by the department	
Request Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
ticketNo	Number	-	Timestamp value of the request
serviceCode	String	-	Service code of the department.
documentNo	String	-	Document number
Response Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
ticketNo	Number	-	Timestamp value of the request
serviceCode	String	-	Service code of the department.
serviceName	String	-	Service name
documentType	Number	1	For link (URL)
		2	For base64Code
		3	For Image
		4	For other type
documentValue	String	-	Value of the document
applicationStatus	Number	5	For Download Document
statusCode	Number	200	Success
		400	Bad Request

Example Request

```
{
  "userId": 9054233544,
  "tokenNo": 20220811050723,
  "serviceCode": "AMD/003",
  "applicationNo": "AP7373633/003"
}
```

Example Response

```
{
  "userId": 9054233544,
  "tokenNo": 20220811050723,
  "serviceCode": "AMD/003",
  "serviceName": "Birth Certificate",
  "documentType": 1,
  "documentValue": "<URL>",
  "applicationStatus": 5,
  "statusCode": 200
}
```

11.5. API-5: OTP Verification for Issuance of Document

The API-5 is used for One Time Password (OTP) verification. The document may be download after confirmation OTP from citizen. That can be done through BSK Portal. The OTP will be generated by the department only. To complete the process of OTP verification two APIs is needed.

The request and response parameters are show below:

- ✓ **Step 1:** The API-5 is required only if OTP verification is required. This API completes the process by calling twice **API-5A** and **API-5B** sequentially.
- ✓ **Step 2:** The API-5A will send the document number along with OTPId (Unique number for OTP sender identification), generated by BSK portal, as the *request*. The department will send the OTP to the citizen and confirmation to BSK portal as *response*.
- ✓ **Step-3:** The API-5B will send the OTP number entered by the citizen along with the earlier OTPId (which was sent in API-5A) for department validation. The department will send response with document download / show status information and API-5 completes.

The prototype of the interface is as follows:

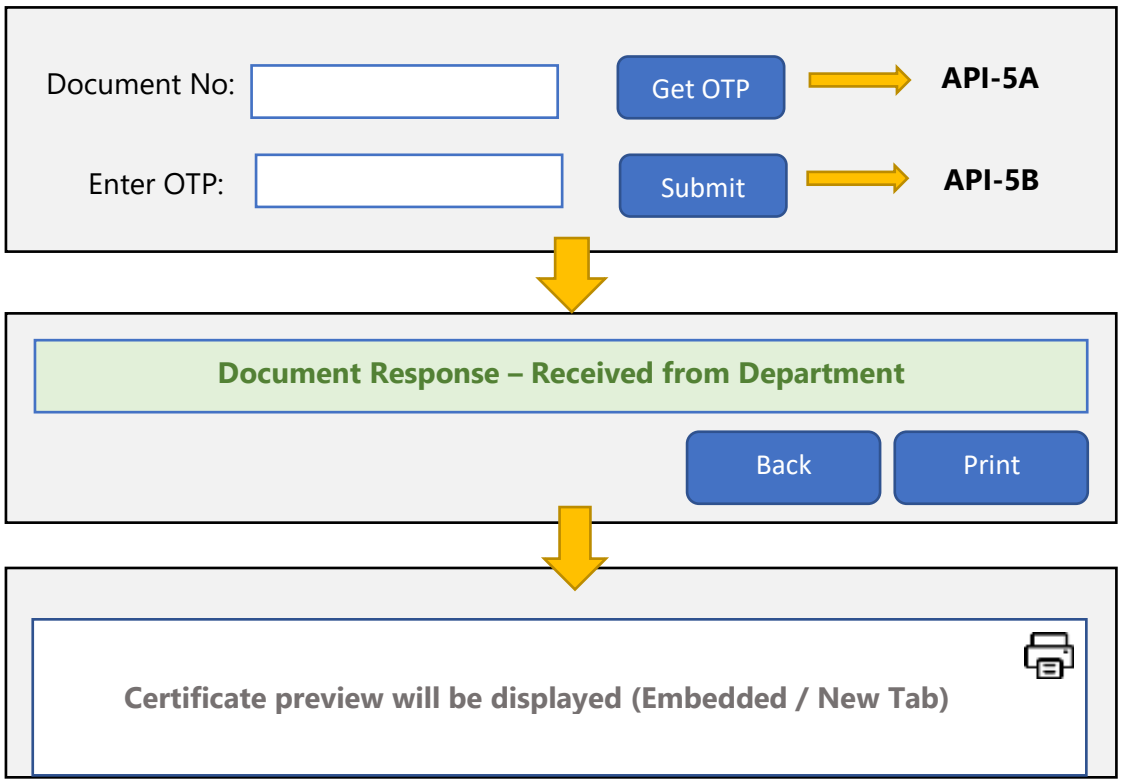

Figure 10: Interface of OTP Verification

The process flow of the OTP verification is given below:

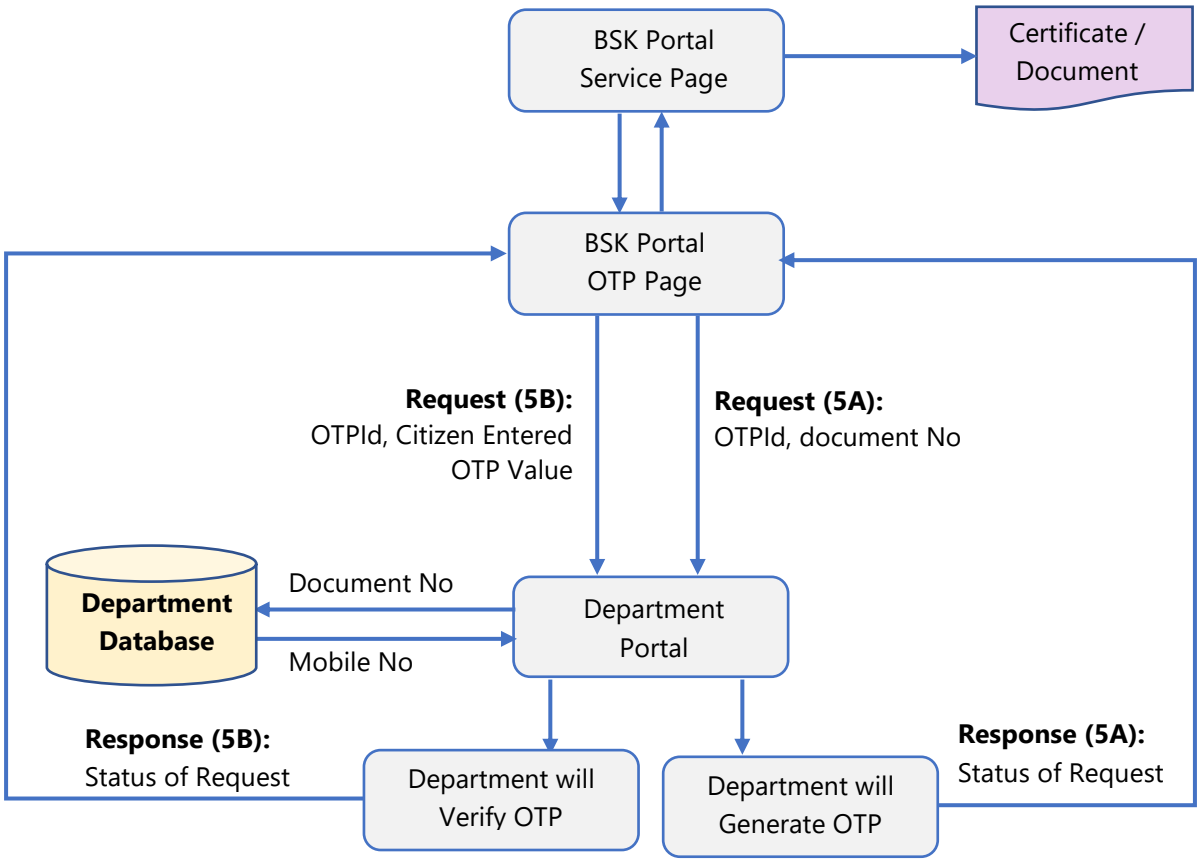

Figure 11: Process Flow of API-5A & API-5B

API-5A			
Use Case	API-5A is called for sending the document no and request to generate OTP for citizen confirmation of service. A unique number is generated for identification of sender.		
HTTP Request	POST	URL will be provided by the department	
Request Body			
Attribute	**Type**	**Value**	**Description**
otpId	Number	-	Unique Id number
documentNo	String	-	Document number
Response Body			
Attribute	**Type**	**Value**	**Description**
otpId	Number	-	Unique Id number
otpGenStatus	Number	1	OTP sent successfully
		2	OTP sent unsuccessful
statusCode	Number	200	Success
		400	Bad Request

Example Request	Example Response
{ "otpId":393747483, "documentNo": "AP7364733/273" }	{ "otpId": 393747483, "otpGenStatus": 1, "statusCode": 200 }

API-5B			
Use Case	API-5B is called for sending the OTP value by the citizen for verification. If OTP verified then the control forwarded for service.		
HTTP Request	POST	URL will be provided by the department	
Request Body			
Attribute	**Type**	**Value**	**Description**
otpId	Number	-	Unique Id number
userTypedOtp	Number	-	OTP Value entered by the citizen
Response Body			
Attribute	**Type**	**Value**	**Description**
otpId	Number	-	Unique Id number
ServiceCode	String	-	Service code of the department. Dept. will provide the exact code. Example: AMD/001
serviceName	String	-	Name of the Service
documentType	Number	1	For link (URL)
		2	For base64Code
		3	For Image
		4	For another format
documentValue	String	-	Value of the document
statusCode	Number	200	Success
		400	Bad Request

Example Request	Example Response
{ "otpId": 393747483, "userTypedOtp": 8374 }	{ "otpId": 393747483, "serviceCode":"AMD/003", "serviceName": "Seed Certificate", "documentType: 1, "documentValue": "<URL>/Base64 Code", "statusCode": 200 }

11.6. API-6: User Authentication

The API-6 is required to authenticate a BSK user. Through the userId (DEO) the department may get the user detail for their verification and others.

The request and response parameters are show below:

- ✓ **Step 1:** The API-6 is required to get the user detail. The API-6 will verify the BSK Data Entry Operator's (DEO) UserId, department code, department access code for the authentication. BSK has large number of operators having variable status at any point of time like 'Active / Inactive / Left '. So, if require, at any point of time it can be verified.

- ✓ **Step 2:** The API will be called by department with userId as request along with the department accessCode provided by BSK Team and BSK portal reply the status of the user along with other information of user as response and the process of API-6 completes.

The process flow of user Authentication is as follows:

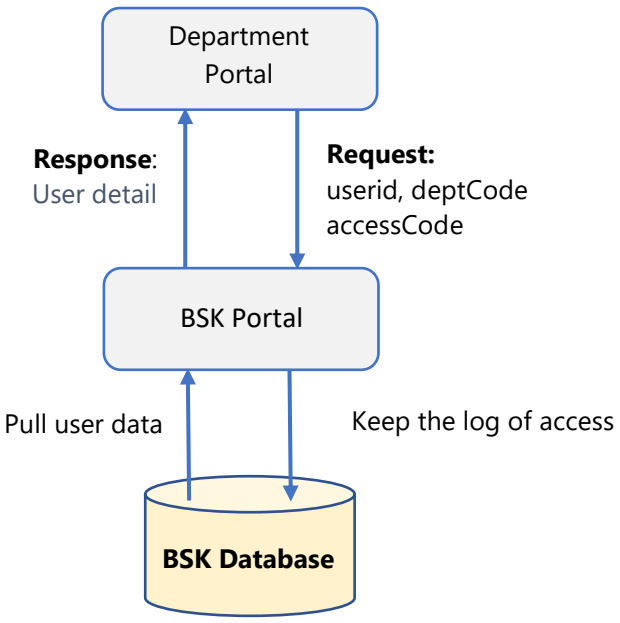

Figure 12: Process Flow of API-6

API-6		
Use Case	API-6 is called for getting the user detail. There are department code and access code verification for getting the detail of user (DEO).	
HTTP Request	POST	URL will be provided by the department

Request Body			
Attribute	**Type**	**Value**	**Description**
userId	Number	-	Userid of the user (DEO)
deptCode	String	-	Department Code
accessCode	Number	-	Access Code to the department will be provided by BSK-Tech Team
Response Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
userName	String	-	Name of user (DEO)
userEmail	Number	-	Registered email id
bskCode	String	-	BSK Code of the associated BSK
bskName	String	-	BSK Name of the associated BSK
Gp	String	-	Gram Panchayat of the associated BSK
block	String	-	Block Name of the associated BSK
subDivision	String	-	Sub Division of the associated BSK
District	String	-	District of the associated BSK
userStatus	Number	1	Active User
		2	In active User
userType	String	"DEO"	Type of the user like DEO
statusCode	Number	200	Success
		400	Bad Request

Example Request

```
{
   "userId": 9054233544,
   "deptCode": "AMD"
   "accessCode":7364443
}
```

Example Response

```
{
   "userId": 9054233544,
   "userName":"Charan Das",
   "userEmail": "charan.das@gmail.com",
   "bskCode":"B/L/384",
   "bskName":"BDO Office Balagarh",
   "gp":"Guptipara",
   "block":"Balagarh",
   "subdivision":"Chandannagar",
   "district":"Hooghly",
   "userStatus": 1,
   "userType": "DEO",
   "statusCode": 200
}
```

11.7. API-7: Pull data from Department

The API-7 is required to pull the missed data from department. BSK will request to the department with two parameters from date and to date. Department will reply all the data received from BSK portal within the dates in JSON Array format only.

The request and response parameters are show below:

- ✓ **Step 1:** The API-7 is required to get the service detail which were served to the department during the start and end date. The all information will be encrypted standard only. It will be server to server calling.

- ✓ **Step 2:** The API will be called by BSK Portal with From_Date and To_Date. Department will validate the request and response the data as mentioned in the table and the process of API-7 completes.

The process flow of user Authentication is as follows:

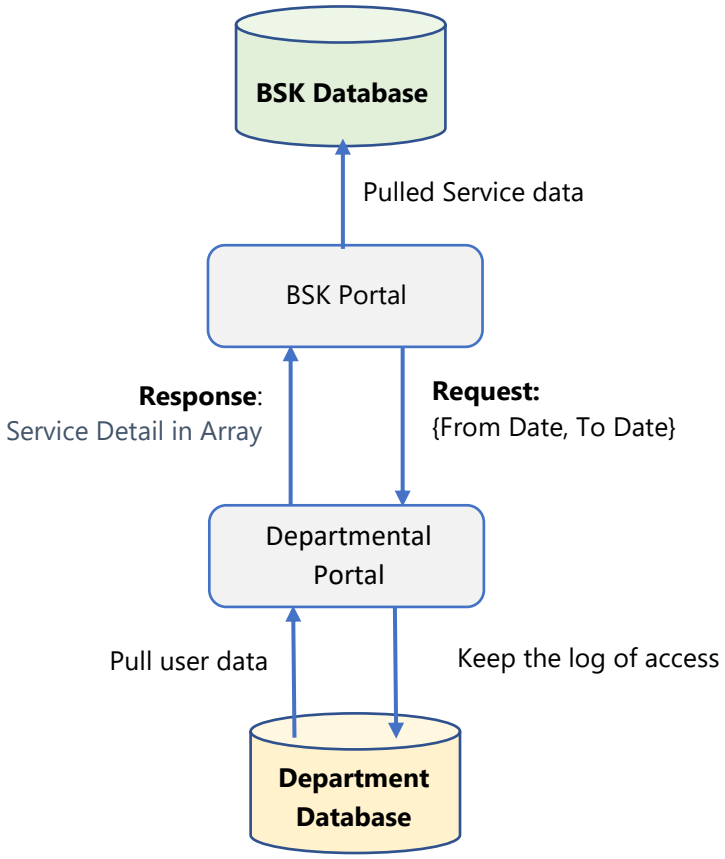

Figure 13: Process Flow of API-7

API-7			
Use Case	API-7 is called by the BSK once the application is submitted as draft or final submission along with transaction detail. All the data must be in JSON Array format only.		
HTTP Request	POST	URL will be provided by the department	
Request Body			
Attribute	**Type**	**Value**	**Description**
From Date	Date	-	From Date (YYYY-MM-DD)
To Date	Date	-	To Date (YYYY-MM-DD)
Response Body			
Attribute	**Type**	**Value**	**Description**
userid	Number	-	10-digit mobile number of user (DEO)
ticketNo	Number	-	Timestamp value of the request
serviceCode	String	-	Service code of the department
appNo	String	-	Application No of the department
appSubTime	Number	-	Application Submission Time
deptPayRefNo	String	-	Payment Reference No
transNo	String	-	Transaction Number
bankRefNo	String	-	Bank Reference Number
paidAmt	Number	-	Transaction Amount in rupees
message	String	-	"Draft Submitted" | "Final Submitted"
applicationStatus	Number	2	For Draft submission of the application
	Number	3	For Final submission of the application
statusCode	Number	200	Success
	Number	400	Bad Request

Example Request	Example Response
{ "formDate" : "2022-12-10", "toDate": "2022-12-22", }	[{ "userid": 9054233544, "ticketNo": 20220811050723, "serviceCode": "AMD/003", "appNo" : "123456", " appSubTime " : 20220811052025, "deptPayRefNo" : "23424423424", "transNo" : "194858535", "bankRefNo" : "46535687", "paidAmt" : 2500.00, "message": "Draft Submitted",

```
    "applicationStatus": 2,
    "statusCode": 200
},
{
    "userid": 9054233476,
    "ticketNo": 20220811050744,
    "serviceCode": "AMD/003",
    "appNo" : "765456",
    " appSubTime " :
    20220811052025,
    "deptPayRefNo" : "23424423424",
    "transNo" : "1234353435",
    "bankRefNo" : "4653443656",
    "paidAmt" : 0.00,
    "message": "Final Submitted",
    "applicationStatus": 2,
    "statusCode": 200
},
{
    "userid": 9854233744,
    "ticketNo": 20220811050723,
    "serviceCode": "AMD/003",
    "appNo" : "123456",
    " appSubTime " :
    20220811052025,
    "deptPayRefNo" : "23424423424",
    "transNo" : "1234353435",
    "bankRefNo" : "4653443656",
    "paidAmt" : 200.00,
    "message": "Draft Submitted",
    "applicationStatus": 2,
    "statusCode": 200
}
]
```


There are some parameters whose value is transmitted in request and response. These values are very important for BSK Portal to generate Management Information Service (MIS) Report. The parameters and corresponding values are at a glance

Parameter: applicationStatus	
Value	**Description**
0	Default value
1	Application Initiated
2	Draft Submission
3	Final Submission
4	Search Application only
5	Download Application

Parameter: documentType	
Value	**Description**
0	Default Value
1	URL Link
2	base64Code
3	Image type
4	Another format 1
5	Another format 2
6	Another format 3

12. Endpoint | URL | IP Address

An endpoint or Uniform Resource Locator (URL) or IP Address is one end of a communication channel. When an API interacts with another system, the touchpoints of the communication are considered endpoints. For APIs, an endpoint can include a URL of a server or service. Each endpoint is the location from which APIs can access the resources they need to carry out their function. APIs work using 'requests' and 'responses.' When an API requests information from a web application or web server, it will receive a response. The place that APIs send requests and where the resource lives, is called an endpoint.

Examples:

https://example.com/another/endpoint
https://example.com/some/other/endpoint
https://example.com/login
https://example.com/accounts

Note: If the department changes the departments endpoint, it is requested to communicate BSK Tech Team to update the APIs before the physical change of endpoints.

13. Integration Time Frame

The process of the API integration must be completed in a stipulated time. Just as the first API integration is completed by completing all FOUR steps. BSK Tech Team proposes a timeline to complete all the stages of API Integration for seamless andedn to end delivery of services. If the department is having number of services, then the stipulated time will be considered accordingly.

SL	API Name	Stipulated Time	Step 1	Step 2	Step 3	Step 4
1	API-1	15 Days	Implement in Staging server	Testing	Implement in Production Server	Final Testing
2	API-2	7 Days				
3	API-3	7 Days				
4	API-4	7 Days				
5	API-5	7 Days				
6	API-6	7 Days				

14. Definition

14.1. REST API

A REST API (also known as RESTful API) is a **RE**presentational **S**tate **T**ransfer **A**pplication **P**rogramming **I**nterface (REST API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. It's also a way for an organization to share resources and information while maintaining security, control, and authentication—determining who gets access to what. When a client request is made via an API, it transfers a representation of the state of the resource to the endpoint. This information, or representation, is delivered in JSON format. This is applicable to web and mobile also.

14.2. JSON Web Token (JWT)

JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. In its compact form, JSON Web Tokens consist of three parts separated by dots (.), which are:

- Header (aaaaaa)
- Payload (bbbbb)
- Signature (cccc)

Therefore, a JWT typically looks like the following.

aaaaaa.bbbbb.cccc

Header: The header typically consists of two parts: the type of the token, which is JWT, and the signing algorithm being used, such as HMAC SHA256. alg=>algorithm, typ=>the media type, cty=> the contain type.

Example:

```
{
  "alg": "HS256",
```

```
  "typ": "JWT"
}
```

Then, this JSON is Base64Url encoded to form the first part of the JWT.

Payload: The second part of the token is the payload, which contains the claims. Claims are statements about an entity (typically, the user) and additional data. There are three types of claims: registered, public, and private claims.

Example:

```
{
  "userid": "9350778825",
   “serviceno”:”275”,
  "name": "Mahit Sen",
  "admin": false
}
```

Signature: To create the signature part one must take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign it. HMAC using SHA-256, called HS256 in the JWA spec. if you want to use the HMAC SHA256 algorithm, the signature will be created in the following way:

Example:

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

The output is three Base64-URL strings separated by dots that can be easily passed in HTML and HTTP environments, while being more compact when compared to XML-based standards such as SAML.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

14.3. IP Whitelisting Process

A whitelist, allow list, or pass list is a mechanism which explicitly allows some identified entities to access a particular privilege, service, mobility, or recognition. Allowing only preapproved individuals to access your network can lower the chances that you might encounter a virus, malware or another cyber-

attack. It can also help you share any sensitive information your business might have with only those you trust.

BSK server has the firewall security and the data receive from other departmental server through API will be restricted unless the public IP address has pre-allowed to send data. The pre-allowed implies whitelisting of IP address. If the department has dynamic public IP, then the DNS (Domain Name System) need to be whitelisted.

14.4. Service Code

As per the notification of Chief Secretary, Notification No: 908-CS/(61)/2022 dated 05-07-2022 the all departments have assigned a 3-character code for uniformity across the state. Department may user this code along with 3-digit service sequence no and form a unique service code.

Department Code	Service Sequence No
AMD	001

Sample Service Code: **AMD001**

SL No	Dept. Code	Department Name
1	AMD	Agricultural Marketing Department
2	AGD	Agriculture Department
3	ARD	Animal Resources Development Department
4	BCW	Backward Classes Welfare Department
5	CMO	Chief Minister's Office
6	CSO	Chief Secretary Office
7	COA	Consumer Affairs Department
8	COD	Co-Operation Department
9	CAD	Correctional Administration Department
10	DMC	Disaster Management and Civil Defence Department
11	ENV	Environment Department
12	FIN	Finance Department
13	FES	Fire & Emergency Services Department
SL No	**Dept. Code**	**Department Name**
14	FIS	Fisheries Department
15	FSD	Food & Supplies Department
16	FPH	Food Processing Ind. and Horticulture Department
17	FOR	Forests Department
18	GSD	Governor Secretariat Department
19	HFW	Health & Family Welfare Department

20	HED	Higher Education Department
21	HHA	Home and Hill Affairs Department
22	HOU	Housing Department
23	ICE	Industry Commerce and Enterprises Department
24	ICA	Information & Cultural Affairs Department
25	ITE	Information Technology & Electronics Department
26	IWD	Irrigation & Waterways Department
27	JUD	Judicial Department
28	LAB	Labour Department
29	LND	Land & Land Reforms and Refugee Relief & Rehabilitation Department
30	LAW	Law Department
31	LAS	Legislative Assembly Secretariat Department
32	MEL	Mass Education Extn. & Library Services Department
33	MSM	Micro, Small & Medium Enterprises and Textiles Department
34	MAM	Minority Affairs & Madrasah Education Department
35	NCE	Non-Conventional Energy Sources Department
36	NBD	North Bengal Development Department
37	PRD	Panchayats & Rural Development Department
38	PAD	Parliamentary Affairs Department
39	PUA	Paschimanchal Unnayan Affairs Department
40	PAR	Personnel & Administrative Reforms Department
41	PSP	Planning, Statistics and Programme Monitoring Department
42	POD	Power Department
43	PEI	Public Enterprises and Industrial Reconstruction Department
44	PHE	Public Health Engineering Department
45	PWD	Public Works Department
46	SED	School Education Department
47	STB	Science & Technology and Bio-Technology Department
48	SHE	Self-Help Group & Self-Employment Department
49	SAD	Sunderban Affairs Department
50	TET	Technical Education, Training & Skill Development Department
51	TOU	Tourism Department
52	TRA	Transport Department
53	TDD	Tribal Development Department
54	UDM	Urban Development and Municipal Affairs Department
55	WRI	Water Resources Investigation & Development Department
56	WCD	Women & Child Development and Social Welfare Department
57	YSS	Youth Services and Sports Department

15. Encryption / Decryption Algorithm

Encryption is the process of scrambling or enciphering data so it can be read only by someone with the means to return it to its original state. It is a crucial feature of a safe and trustworthy Internet. Encryption is important when we need to find out whether

data has been tampered with (data integrity), to increase people's confidence that they are communicating with the people they think are communicating with (authentication) and to be sure that messages were sent and received (non-repudiation).

BSK uses Advanced Encryption Standard–256 bits-Cipher Block Chaining (AES-256-CBC) technology for maximizing the security. "AES-256-CBC" is a 256-bit AES encryption refers to the process of concealing plaintext data using the AES algorithm and an AES key length of 256 bits. In addition, 256 bits is the largest AES key length size, as well as its most mathematically complex. It is also the most difficult to crack. AES 256-bit encryption uses 14 transformation rounds to convert plaintext into ciphertext and, because it's nearly impossible to break, is approved by the National Security Agency (NSA) to protect both secret and top-secret government information. The Cipher Block Chaining (CBC) mode is a typical block cipher mode of operation using block cipher algorithm. In this version, we provide Advanced Encryption Standard (AES) processing ability, the cipher key length for 256 bits. AES is a symmetric encryption algorithm because it uses one key to encrypt and decrypt information, whereas its counterpart, asymmetric encryption, uses a public key and a private key.

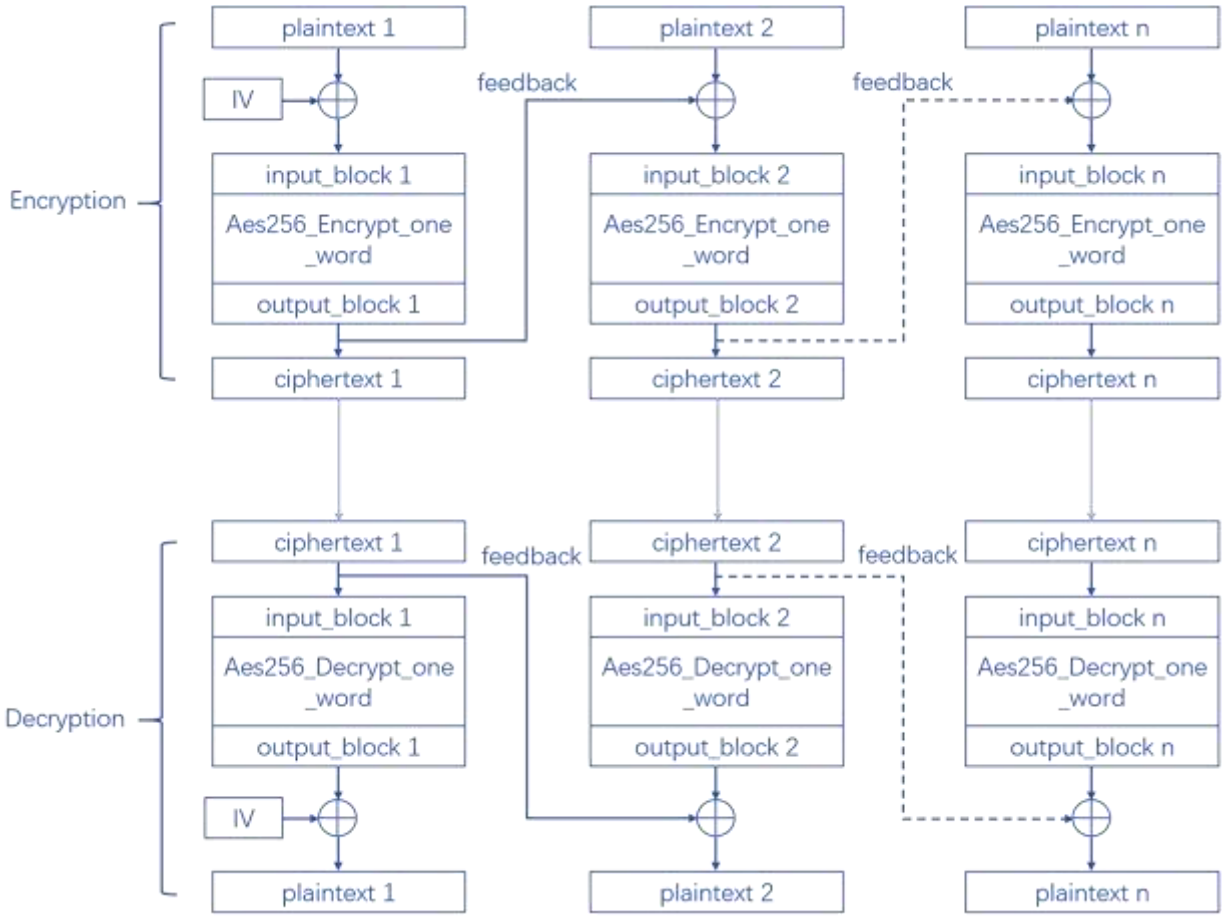

Figure 14: The algorithm flow chart

Encryption Key: Advanced Encryption Standard (AES) keys are symmetric keys that can be three different key lengths, BSK uses 256 bits. AES is the encryption standard that is recognized and recommended. The 256-bit keys are the longest allowed by AES. BSK Tech Team will provide the decryption key to decrypt the data. It is required to encrypt the data before sending to network.

Decryption Key: Advanced Encryption Standard (AES) keys are symmetric keys that can be three different key lengths, BSK uses 256 bits. AES is the encryption standard that is recognized and recommended. The 256-bit keys are the longest allowed by AES. BSK Tech Team will provide the decryption key to decrypt the data. It is required to decrypt the data before further use.

Initialization Vector: An initialization vector (IV) is an arbitrary number that can be used with a secret key for data encryption to foil cyber-attacks. This number, also called a nonce (number used once), is employed only one time in any session to prevent unauthorized decryption of the message by a suspicious or malicious actor. BSK Tech Team will provide that value which will be required to decrypt the code.

Algorithm: Use the algorithm as "AES-256-CBC" or "aes-256-cbc" depending upon the language is used in the department portal.

Data: The data will be forwarded from BSK portal only in the JSON format. The variable length data will be transferred. Based on the requirement of the department, BSK portal will send accordingly.

Step by Step process of Encryption

Step 1: keep the data ready in JSON format

Step 2: Get the following data from BSK Tech Team

a) Encryption Key
b) Initialization Vector

Step 3: call the function for encryption algorithm "AES-256-CBC" and encrypt. Functions in different language is given for the ready reference.

Step 4: Send the encrypted data in POST method

Step 5: The process of encryption ends.

Step by Step process of Decryption

Step 1: keep the encrypted data ready

Step 2: Get the following data from BSK Tech Team

a) Decryption Key

b) Initialization Vector

Step 3: call the function for decryption algorithm "AES-256-CBC" and decrypt. Functions in different language is given for the ready reference.

Step 4: After decryption, the data will be in JSON format

Step 5: The process of decryption ends.

BSK Provides Department Landing Page for Testing

https://bsk.wb.gov.in/landingfrombsk

Source Code is available for ready reference in multiple languages (Node | Java | PHP)

Prior to run the following code, it is required to keep the two encryption and decryption values. The sample value is given here for test purpose. For live, each department has different values. BSK Tech Team will provide the encryption / decryption code.

For the following code, the encryption decryption test keys are as follows:

a) Encryption Key: **IP16TDW0L5AQB41V6S6J8QLTPLRXBV2W**
b) Decryption Key: **IP16TDW0L5AQB41V6S6J8QLTPLRXBV2W**
c) Initialization Vector: **V0ZMZO6WZ45KY2PL**

a) Node

Encryption:

```
let text =
'{"userid":"8527419636","ticketno":"20221202153322882","citizenmobile":"9963258963","citizenname":"Anupam Ghosh"}'

const key   = "IP16TDW0L5AQB41V6S6J8QLTPLRXBV2W";
const iv    = "V0ZMZO6WZ45KY2PL";

let cipher = crypto.createCipheriv('aes-256-cbc', Buffer.from(key), iv);
let encrypted = cipher.update(text);
encrypted = Buffer.concat([encrypted, cipher.final()]);
let encryptedData = encrypted.toString('base64');
```

OUTPUT:

```
m4p3CxxkyshZk6hQdSi509Uu3Gn8i4MXIz/uqgSl40FfH0Hv7M/ThZj3YnAiVLdmuQ4Dd
```

GEPVLYtOihNsC2c59tkK97qagxo9+rX6hWJem/AilzVNBTsN206liJKiAaoFAKfmaxFAFo/UmGTrVExKI428RouVxqyechhX0OyIDg=

POST REQUEST:
Department URL: https://example.com/landingPage
Accepted Method: POST
Parameters:
agId: BSK
encData: encryptedData

Decryption:

```
const key   = "IP16TDW0L5AQB41V6S6J8QLTPLRXBV2W";
const iv    = "V0ZMZO6WZ45KY2PL";

let encryptedText = Buffer.from(encryptedData, 'base64');
let decipher = crypto.createDecipheriv('aes-256-cbc', Buffer.from(key), iv);
let decrypted = decipher.update(encryptedText);
decrypted = Buffer.concat([decrypted, decipher.final()]);
let decryptedData =  decrypted.toString();
```

OUTPUT:
{"userid":"8527419636","ticketno":"20221202153322882","citizenmobile":"9963258963","citizenname":"Anupam Ghosh"}

b) Java

```
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64; //commons-codec-1.15.jar need to add in class path

public class App {

   private static final String key = "IP16TDW0L5AQB41V6S6J8QLTPLRXBV2W";
   private static final String initVector = "V0ZMZO6WZ45KY2PL";

// Encryption
   public static String encrypt(String value) {
      try {
         IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
         SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
         Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
         cipher.init(Cipher.ENCRYPT_MODE, skeySpec, iv);
         byte[] encrypted = cipher.doFinal(value.getBytes());
```

```
            return Base64.encodeBase64String(encrypted);
              } catch (Exception ex) {
                 ex.printStackTrace();
              }
           return null;
    }

// Decryption
    public static String decrypt(String encrypted) {
        try {
            IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
            SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, skeySpec, iv);
            byte[] original = cipher.doFinal(Base64.decodeBase64(encrypted));
            return new String(original);
             } catch (Exception ex) {
                ex.printStackTrace();
         }
        return null;
    }

public static void main(String[] args) {
        String originalString = "{'userid':'8527419636','ticketno':'20221202153322882',
                                        'citizenmobile': '9963258963','citizenname':'Anupam
Ghosh'}";
        System.out.println("Original String to encrypt - " + originalString);
        String encryptedString = encrypt(originalString);
        System.out.println("Encrypted String - " + encryptedString);
        String decryptedString = decrypt(encryptedString);
        System.out.println("After decryption - " + decryptedString);
    }
}
```

c) PHP

```
<?php
   echo "<p>Encryption - Decryption ACE-256-CBC</p>\n";

    // Encryption
    $data='{"userid":"8527419636","ticketno":"20221202153322882","citizenmobile":"996
    3258963","citizenname":"Anupam Ghosh"}';
```

```
    echo "Before Encryption : " . $data . "<br>";

    $encryptionKey = "IP16TDW0L5AQB41V6S6J8QLTPLRXBV2W";
    $initializationVector = "V0ZMZO6WZ45KY2PL";

    $options = 0;
    $encryptedData = openssl_encrypt($data, 'aes-256-cbc', $encryptionKey, $options,
        $initializationVector);

    echo "Encrypted Data: ". $encryptedData."<br>";


// Decryption


    $decryptionKey = "IP16TDW0L5AQB41V6S6J8QLTPLRXBV2W";
    $initializationVector = "V0ZMZO6WZ45KY2PL";

    $decryptedData = openssl_decrypt($encryptedData, 'aes-256-cbc', $decryptionKey,
        $options, $initializationVector);

    echo "After Decryption: ".$decryptedData;
?>
```

16. Point of Contacts

For Departments to Onboard the BSK portal:

Saadia Azim
Chief Operating Officer
Bangla Sahayata Kendra, PMU
+91 9830047512
coo.bsk@wb.gov.in

For technical matters regarding API Integration:

Dr. Arindam Ray
Chief Technology Officer (CTO)
Bangla Sahayata Kendra, PMU
+ 91 93507 78825
cto.bsk@wb.gov.in

Manojit Boral
Senior Software Personnel
Bangla Sahayata Kendra, PMU
+91 94339 33723
manojitbaral.bskpmu@gmail.com

Anupam Ghosh
Senior Software Personnel
Bangla Sahayata Kendra, PMU
+91 99326 12608
anupamghosh.bskpmu@gmail.com

17. Version Information

There are major changes of API Integration concepts over the previous versions. BSK-Tech Team had already integrated with number of departments with previous version of API documents which will be gradually updated with new architecture. The version information is given below:

SL	Version No	Updation	Description
1	V 1.0	October 2020	Initial Version
2	V 2.0	January 2021	
3	V 3.1	March 2022	
4	V 3.2	July 2022	
5	V 4.0	August 2022	JWT Token – Digitally Signed Token security
6	V 5.0	December 2022	Hybrid Model of API Integration. Modified: API-1 Added: Encryption-decryption algorithm, Data flow model, API-7

18. Conclusion

The API Integration process is the data flow between two parties with all possible securities against Unauthorised Access, Denial-of-Service (DoS) attack and followed the best practices. As the process develops and the scope for new APIs increases, BSK PMU may add new APIs per the system's requirements.