

Integration Guide - Data Protection Platform

Integration Guide Telkomsel Application Data Protection Platform

Prepared for



Document Information

Revision History

Date	Version	Author	Modification
27-Oct-23	1.0	Murdjoko	First release
12-Nov-23	1.1	Murdjoko	Revision
13-Dec-23	1.2	Murdjoko	<i>Update: application integration approach</i>

Distribution List

Date	Name	Title

Reference Document


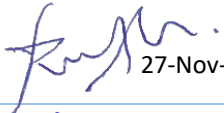

Document Name	File Name	Version	Date of issuance
Software Requirement Specification	TSEL-DPP-SRS.docx	1.0	10-Nov-23
High Level Design	TSEL-DPP-HLD.docx	1.3	10-Nov-23
Interface Agreement	TSEL-DPP-IFA.docx	1.1	10-Nov-23

Document Location

Paper copies are valid only on the day they are printed. Refer to the author if you are in any doubt about the currency of this document.

Document Approval

This document has been reviewed and approved by:

No	Name	Company	Title	Approval
1	Johnny F. Hamonangan	Accenture	Program Delivery Lead	 27-No-23
2	Rudy Lim	Accenture	Subject Matter Advisor	 27-Nov-23
3	Jack Reyner	Telkomsel	Manager Data Protection & Access Security Platform	 7 Dec 2023

Contents

1	OVERVIEW.....	1
1.1	PURPOSE.....	1
1.2	RESPONSIBILITY FOR THIS DOCUMENT	1
1.3	DOCUMENT SCOPE.....	1
2	INTEGRATION GUIDE.....	2
2.1	TIMELINE.....	2
2.2	PII DATA IN DPP.....	3
2.3	DATA PROTECTION PLATFORM (DPP).....	4
2.4	RESTFUL API.....	5
2.4.1	API Authentication.....	5
2.4.2	FPE Template.....	5
2.4.3	API Tokenize.....	6
2.4.4	API Detokenize.....	8
2.4.5	API Encryption.....	10
2.4.6	API Decryption	13
2.5	CADP.....	15
2.5.1	CADP Sample.....	18
3	INTEGRATION FLOW.....	22
4	INTEGRATION APPROACH	23
4.1	INTRODUCING NEW HTTP HEADER	ERROR! BOOKMARK NOT DEFINED.
5	DATA MIGRATION	25
6	APPENDIX.....	26

Term of Abbreviation

Abbreviation	Description	Notes
AES	Advanced Encryption Standard	Symmetric block cipher to encrypt sensitive data
API	Application programming interface	software intermediary that allows two applications to talk to each other
BDT	Batch Data & Tokenization Service	Server for tokenizing rest data and provide API Service to VTS
CTM	CipherTrust Manager	Central management point for key and policy management
CTS	CipherTrust Tokenization Server	Server for tokenizing records and managing access to tokens and clear-text data
DDC	CipherTrust Data Discovery and Clasification	Server for discovery data and classification based on template
FFX	Feistel-based encryption	Solution for Format-preserving which produces an output which matches the length of the input
FF1	FFX[Radix]	FFX Encryption Mode which is also in standards processes under ANSI X9 as X9.119 and X9.124.
FF3	FFX Encryption mode	Method for FPE by NIST (National Institute of Standards and Technology)
FPE	Format Preserving Encryption	Encrypting in such a way that the output (the ciphertext) is in the same format as the input (the plaintext)
HA	High Availability	Ability of a system to be continuously operational for a desirably long length of time
HTTP	Hypertext Transfer Protocol	Application-layer protocol for transmitting hypermedia documents,
HTTPS	Hypertext transfer protocol secure	Secure version of HTTP, which is the primary protocol used to send data between a web browser and a website
PII	Personal Identifiable Information	Any data that could potentially identify a specific individual
RSA	Rivest-Shamir-Adleman	Public-key cryptosystem that is widely used for secure data transmission.
REST API	Representational State API	Architectural style for an application program interface (API) that uses HTTP requests to access and use data

Integration Guide - Data Protection Platform

1 Overview

This document will specify the Data Protection Platform interface provided in the platform. This document will help to decide the integration requirements will be met by the proposed architecture of Data Protection Platform.

1.1 Purpose

This document identifies agreed-upon design requirements and constraints that must be satisfied by the software that provides the interfacing functions. This document is intended for use by the developers of the applications identified, and by the test organizations responsible for the testing of these applications.

1.2 Responsibility for this Document

The Project Team will be responsible for updates to this document. All parties agree that this document cannot be changed without mutual consent.

1.3 Document Scope

The objective for this solution is to protect personal data, Particularly PII (Personal Identifiable Information) within Telkomsel environment.

Integration Guide - Data Protection Platform

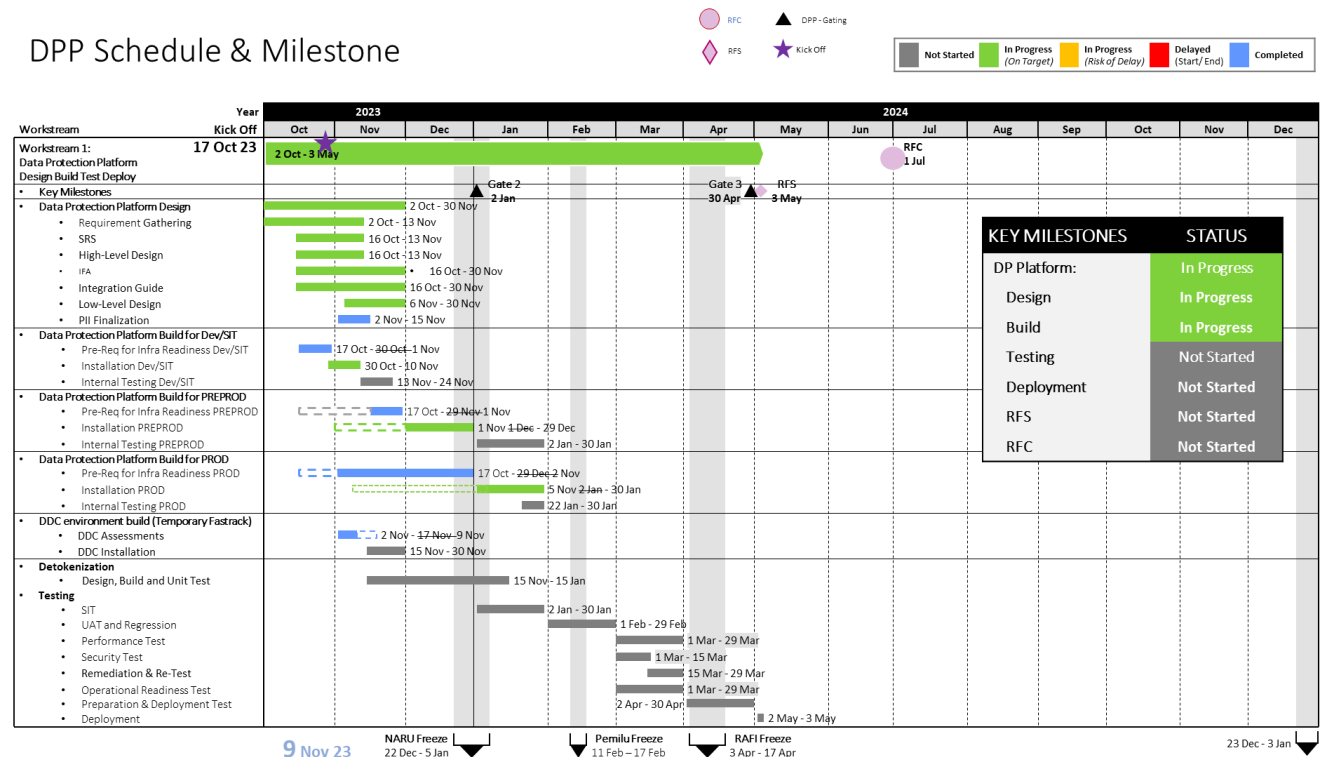
2 Integration Guide

The choice between **encryption** and **tokenization** will depend on user requirements. Following are criterias when determining what the best method to use.

Criteria	Tokenization	Encryption
Working process	Replaces sensitive data with a randomly generated token value	Transforms plaintext into ciphertext using an encryption algorithm and key
Kinds of Supported Data	Structured data such as data stored in database	Structured data and unstructured data, such as image, file.
Output	Output is format and length preserving (FEP: FF1)	Output is not generally format or length preserving (e.g. AES, RSA); exception <u>FPE– Format preserving Encryption</u>

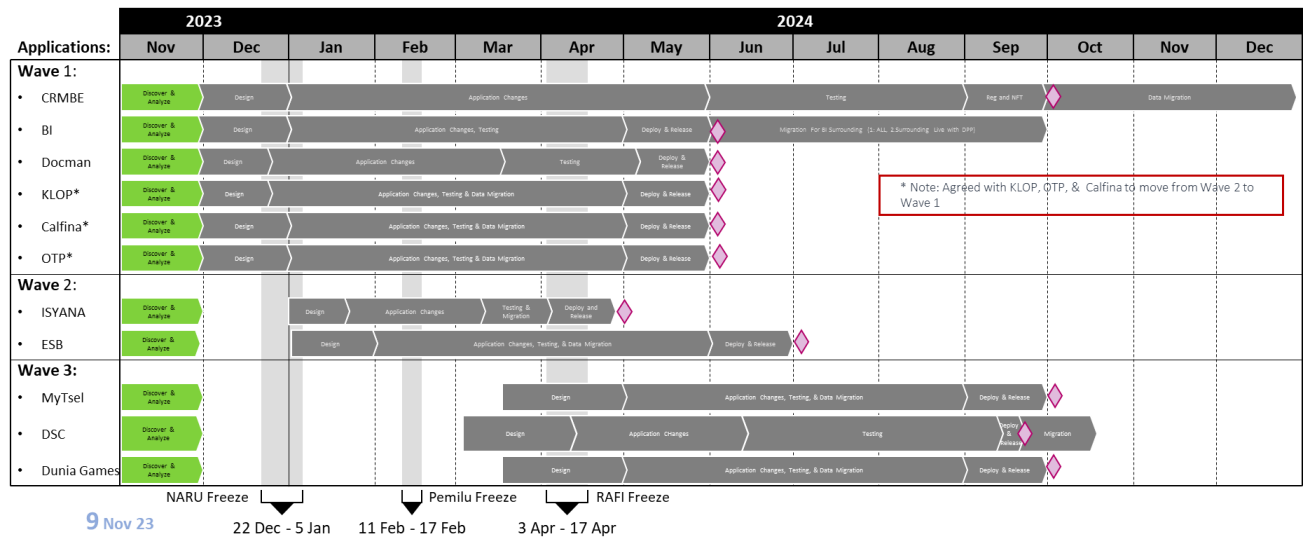
2.1 Timeline

DPP Schedule & Milestone



DPP timeline for Application

Integration Guide - Data Protection Platform



2.2 PII data in DPP

The Telkomsel's Data Governance team has delineated the specific personal data that requires protection, distinguishing it from other customer-related data that doesn't necessitate tokenization as it lacks the capacity to identify individual persons.

The table below provides a breakdown of the personal data that requires safeguarding via tokenization/encryption, as well as data that doesn't mandate protection and can be utilized in its original state.

No	Data Type	Encrypted/Tokenize	NOT to Tokenize/Encrypt
1	Name	First Name, Last Name, Mother Maiden Name, Emergency Contact	
2	Address	Address, Address in KTP, Billing Address	Floor, Residence Name, Kelurahan/Local Area, Kecamatan/District, Province, City, Country, Kode Pos/Postal Code
3	Email	Email	
4	DOB	DOB	Birthplace
5	ID	NIK, NOKK, Passport No, NPWP, KITAS, SIUPP	Marital Status, Nationality, Religion, Place of Birth, Occupation
6	Financial Info	Credit Card No, Debit Card No	Metode Pembayaran, Transaction Code, Monthly Income, Billing
7	Phone	Office Phone, Home Phone, Mobile Phone	
8	IP Address	IP Address	
9	Location		Coordinate/Longitude Latitude/GPS position, Mobile Position
10	MSISDN*	MSISDN	
11	IMSI	IMSI	
12	IMEI	IMEI	
13	Other Data	Cookie ID (as this contains ID)	Ads ID, Device Type, URL (photo URL, live tracking URL, URL Info), Browsing History,

Integration Guide - Data Protection Platform

			Payment Method, Package Purchase, Quota Usage, Telco Score
14	Custom Data	Special Case	

** Considering the pervasive operational usage of MSISDN as a customer identifier within Telkomsel's application landscape, the tokenization for MSISDN will be permitted in its original format, provided that other identifiable personal information has already undergone tokenization. Any exceptions to this rule must be thoroughly documented and approved by Telkomsel's management.*

2.3 Data Protection Platform (DPP)

List of DPP Servers:

Non-Prod

No	Hostname	IP Address	Desc
1	dppctmtbsdapp01	10.38.45.151	CipherTrust Manager for CADP
2	dppctstbsdapp01	10.38.45.153	CipherTrust Tokenization Server

Pre-Prod

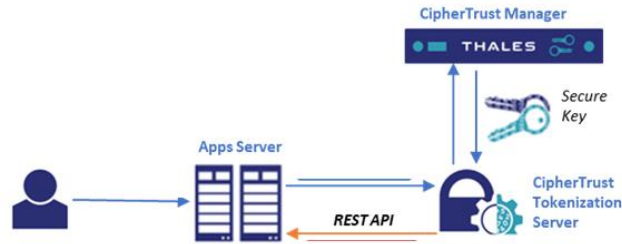
No	Hostname	IP Address	Desc
1	dppctmtbsdapp02	10.59.196.117	CipherTrust Manager for CADP
2	dppctmtbsdapp03	10.59.196.118	CipherTrust Manager for CADP
3	dppctstbsdapp02	10.59.196.123	CipherTrust Tokenization Server
4	dppctstbsdapp03	10.59.196.124	CipherTrust Tokenization Server

Production

No	Hostname	IP Services	Desc
1	dppctmtsdapp01	10.59.129.221	CipherTrust Manager for CADP
2	dppctmtsdapp02	10.59.129.222	CipherTrust Manager for CADP
3	dppctstsdapp01	10.59.129.226	CipherTrust Tokenization Server
4	dppctstsdapp02	10.59.129.227	CipherTrust Tokenization Server
5	dppctstsdapp03	10.59.129.228	CipherTrust Tokenization Server

Integration Guide - Data Protection Platform

2.4 RESTful API



DPP provides API in order application can call API POST to these endpoints.

For testing purposes, access for application will be pointed to Non-Prod, and Pre-Prod server (**dppctstbsdapp01, dppctstbsdapp02/dppctstbsdapp03**) with specific **client certificate**.

No	Endpoint	Method	Endpoint description	Authentication
1	/vts/rest/v2.0/tokenize	POST	Perform Tokenization	Required
2	/vts/rest/v2.0/detokenize	POST	Perform Detokenization	Required
3	/vts/crypto/v1/encrypt	POST	Perform Encryption	Required
4	/vts/crypto/v1/decrypt	POST	Perform Decryption	Required

2.4.1 API Authentication

DPP provides client certificates for API authentication to endpoints. There are 2 (two) client certificates provided :

- Client certificate (*client-cert.pem*) that signed by Telkomsel CoreCA
- Private key to enable (*client-key.pem*) authentication

2.4.2 FPE Template

Template for Tokenization/Detokenization

No	Tokenization Template	Tokenization Group	Tokenization Key	Format	Character Set Name	Keep Left	Keep Right
1	TT_Name	TG_Name	TT_Name	FF1	Alphanumeric	0	0
2	TT_Address	TG_Address	TT_Address	FF1	Alphanumeric	0	0
3	TT_DOB	TG_DOB	TT_DOB	FF1	All Digits	0	0
4	TT_Email	TG_Email	TT_Email	FF1	Alphanumeric	0	0
5	TT_ID_NIK_NOK	TG_ID	TT_ID	FF1	All Digits	0	0
6	TT_ID_OTHERS	TG_ID	TT_ID	FF1	Alphanumeric	0	0
7	TT_IP_V4	TG_IP	TT_IP	FF1	All Digits	0	0
8	TT_IP_V6	TG_IP	TT_IP	FF1	Alphanumeric	0	0
9	TT_Phone	TG_Phone	TT_Phone	FF1	All Digits	0	0
10	TT_MSISDN	TG_MSISDN	TT_MSISDN	FF1	All Digits	5	0

Integration Guide - Data Protection Platform

11	TT_IMEI	TG_IMEI	TT_IMEI	FF1	All Digits	0	0
12	TT_IMSI	TG_IMSI	TT_IMSI	FF1	All Digits	0	0
13	TT_Finance	TG_Finance	TT_Finance	FF1	All Digits	0	0
14	TT_Cookie_Id	TG_Cookie_Id	TT_Cookie_Id	FF1	Alphanumeric	0	0
15	TT_Message_Id	TG_Message_Id	TT_Message_Id	FF1	Alphanumeric	0	0
16	TT_Digits	TG_Digits	TT_Digits	FF1	Digits	0	0
17	TT_Alphanumeric	TG_Aphanumeric	TT_Aphanumeric	FF1	Alphanumeric	0	0
18	TT_Other	TG_Other	TT_Other	FF1	Alphanumeric	0	0
19	TT_NoTokenized	TG_NoTokenized	TT_Other	Special	case not Tokenized	0	0

2.4.3 API Tokenize

Source data:

```
{
  "tokengroup": "TG_ID", "tokentemplate": "TT_ID", "data": "CRMB0001",
  "tokengroup": "TG_Name", "tokentemplate": "TT_Name", "data": "Sanjaya Pratama Putra",
  "tokengroup": "TG_Phone", "tokentemplate": "TT_Phone", "data": "+629564456456",
  "tokengroup": "TG_Email", "tokentemplate": "TT_Email", "data": "sanjayapratama@telkomsel.co.id",
  "tokengroup": "TG_Other", "tokentemplate": "TT_Other", "data": "#343545356343453#",
  "tokengroup": "TG_ID", "tokentemplate": "TT_ID", "data": "CRMB0002",
  "tokengroup": "TG_Name", "tokentemplate": "TT_Name", "data": "Raja Putra Perdana",
  "tokengroup": "TG_Phone", "tokentemplate": "TT_Phone", "data": "086234534732",
  "tokengroup": "TG_Email", "tokentemplate": "TT_Email", "data": "rajaperda@telkomsel.co.id",
  "tokengroup": "TG_Other", "tokentemplate": "TT_Other", "data": "#34652645642355#"
}
```

Expected response:

```
{
  {
    "token": "CRMB0001",
    "status": "Succeed"
  },
  {
    "token": "J7YLC7V wnHoEAD xNjaX",
    "status": "Succeed"
  },
  {
    "token": "+738876546497",
    "status": "Succeed"
  },
  {
    "token": "IB20GZOv0EYxCO@xnVphPtko.kG.H6",
    "status": "Succeed"
  },
  {
    "token": "#959029461301042#",
    "status": "Succeed"
  },
  {
    "token": "CRMB0002",
    "status": "Succeed"
  },
  {
  }
```

Integration Guide - Data Protection Platform

```

    "token": "sDsH x6YNq xyKdD0k",
    "status": "Succeed"
  },
  {
    "token": "871541105056",
    "status": "Succeed"
  },
  {
    "token": "tKZjaiTAZ@AFxLhkSKB.fq.7J",
    "status": "Succeed"
  },
  {
    "token": "#72801269830316#",
    "status": "Succeed"
  }
}

```

Using curl command

`curl --tlsv1.2 -k --key client.pem --cert client.cer -X POST -d'<data>' https://dppctstbsdapp02/vts/rest/v2.0/tokenize/`

```

test - Notepad
File Edit Format View Help
[{"tokengroup": "TG_ID", "tokentemplate": "TT_ID", "data": "CRMB0001"},
{"tokengroup": "TG_Name", "tokentemplate": "TT_Name", "data": "Sanjaya Pratama Putra"},
{"tokengroup": "TG_Phone", "tokentemplate": "TT_Phone", "data": "+629564456456"},
{"tokengroup": "TG_Email", "tokentemplate": "TT_Email", "data": "sanjayapratama@telkomsel.co.id"},
{"tokengroup": "TG_Other", "tokentemplate": "TT_Other", "data": "#343545356343453#"},
{"tokengroup": "TG_ID", "tokentemplate": "TT_ID", "data": "CRMB0002"},
{"tokengroup": "TG_Name", "tokentemplate": "TT_Name", "data": "Raja Putra Perdana"},
{"tokengroup": "TG_Phone", "tokentemplate": "TT_Phone", "data": "086234534732"},
{"tokengroup": "TG_Email", "tokentemplate": "TT_Email", "data": "rajaperda@telkomsel.co.id"},
{"tokengroup": "TG_Other", "tokentemplate": "TT_Other", "data": "#34652645642355#"}]

22/11/2023 16:36:32 /home/mobaxterm/Desktop/Token curl -k -X POST -u 'dppSvcCRM:T3lkomsel@123' --data-binary @te
st.txt https://10.11.8.228/vts/rest/v2.0/tokenize
[{"token": "CRMB0001", "status": "Succeed"}, {"token": "J7YLC7V wnHoEAD xNjaX", "status": "Succeed"}, {"token": "+738876546497", "status":
"Succeed"}, {"token": "IB20GZ0v0EYxCO@xnVphPtko.kG.H6", "status": "Succeed"}, {"token": "#959029461301042#", "status": "Succeed"}, {"toke
n": "CRMB0002", "status": "Succeed"}, {"token": "sDsH x6YNq xyKdD0k", "status": "Succeed"}, {"token": "871541105056", "status": "Succeed"},
{"token": "tKZjaiTAZ@AFxLhkSKB.fq.7J", "status": "Succeed"}, {"token": "#72801269830316#", "status": "Succeed"}]

```

When using postman, upload 2 client certificates from menu Setting → Certificates

Integration Guide - Data Protection Platform

POST https://10.11.8.228/vts/rest/v2.0/tokenize Send

Params Auth Headers (9) **Body** Pre-req. Tests Settings

raw Text

```

1 {"tokengroup":"TG_ID","tokentemplate":"TT_ID",
  "data":"CRMB0001"},
2 {"tokengroup":"TG_Name","tokentemplate":"TT_Name",
  "data":"Sanjaya Pratama Putra"},
3 {"tokengroup":"TG_Phone","tokentemplate":"TT_Phone","data":"+629564456456"},
4 {"tokengroup":"TG_Email","tokentemplate":"TT_Email",
  "data":"sanjayapratama@telkomsel.co.id"},
5 {"tokengroup":"TG_Other","tokentemplate":"TT_Other",
  "data":"#343545356343453#"},
6 {"tokengroup":"TG_ID","tokentemplate":"TT_ID",
  "data":"CRMB0002"},
7 {"tokengroup":"TG_Name","tokentemplate":"TT_Name","data":"Raja
  Putra Perdana"},
8 {"tokengroup":"TG_Phone","tokentemplate":"TT_Phone",
  "data":"086234534732"},
9 {"tokengroup":"TG_Email","tokentemplate":"TT_Email",
  "data":"rajaperda@telkomsel.co.id"},
10 {"tokengroup":"TG_Other","tokentemplate":"TT_Other",
  "data":"#34652645642355#"}
11

```

Pretty Raw Preview Visualize JSON Save Response

```

1 {
2   "token": "CRMB0001",
3   "status": "Succeed"
4 },
5 {
6   "token": "J7YLC7V wnHoEAD xNjaX",
7   "status": "Succeed"
8 },
9 {
10  "token": "+738876546497",
11  "status": "Succeed"
12 },
13 {
14  "token": "IB20GZ0v0EYxC0@xnVphPtko.kG.H6",
15  "status": "Succeed"
16 },
17 {
18  "token": "#959029461301042#",
19  "status": "Succeed"
20 },
21 {

```

POST https://10.11.8.228/vts/rest/v2.0/tokenize Send

Params Auth Headers (9) **Body** Pre-req. Tests Settings

raw Text

```

1 {"tokengroup":"TG_ID","tokentemplate":"TT_ID",
  "data":"CRMB0001"},
2 {"tokengroup":"TG_Name","tokentemplate":"TT_Name",
  "data":"Sanjaya Pratama Putra"},
3 {"tokengroup":"TG_Phone","tokentemplate":"TT_Phone","data":"+629564456456"},
4 {"tokengroup":"TG_Email","tokentemplate":"TT_Email",
  "data":"sanjayapratama@telkomsel.co.id"},
5 {"tokengroup":"TG_Other","tokentemplate":"TT_Other",
  "data":"#343545356343453#"},
6 {"tokengroup":"TG_ID","tokentemplate":"TT_ID",
  "data":"CRMB0002"},
7 {"tokengroup":"TG_Name","tokentemplate":"TT_Name","data":"Raja
  Putra Perdana"},
8 {"tokengroup":"TG_Phone","tokentemplate":"TT_Phone",
  "data":"086234534732"},
9 {"tokengroup":"TG_Email","tokentemplate":"TT_Email",
  "data":"rajaperda@telkomsel.co.id"},
10 {"tokengroup":"TG_Other","tokentemplate":"TT_Other",
  "data":"#34652645642355#"}
11

```

Pretty Raw Preview Visualize JSON Save Response

```

20 },
21 {
22   "token": "CRMB0002",
23   "status": "Succeed"
24 },
25 {
26   "token": "sDsH x6YNq xyKdD0k",
27   "status": "Succeed"
28 },
29 {
30   "token": "871541105056",
31   "status": "Succeed"
32 },
33 {
34   "token": "tKZjaiTAZ@AFxLhkSKB.fq.7J",
35   "status": "Succeed"
36 },
37 {
38   "token": "#72801269830316#",
39   "status": "Succeed"
40 }

```

2.4.4 API Detokenize

Data:

```

{"tokengroup":"TG_ID","tokentemplate":"TT_ID","token":"CRMB0001"},
{"tokengroup":"TG_Name","tokentemplate":"TT_Name","token":"J7YLC7V wnHoEAD xNjaX"},
{"tokengroup":"TG_Phone","tokentemplate":"TT_Phone","token":"+738876546497"},
{"tokengroup":"TG_Email","tokentemplate":"TT_Email","token":"IB20GZ0v0EYxC0@xnVphPtko.kG.H6"},
{"tokengroup":"TG_Other","tokentemplate":"TT_Other","token":"#959029461301042#"},
{"tokengroup":"TG_ID","tokentemplate":"TT_ID","token":"CRMB0002"},
{"tokengroup":"TG_Name","tokentemplate":"TT_Name","token":"sDsH x6YNq xyKdD0k"},
{"tokengroup":"TG_Phone","tokentemplate":"TT_Phone","token":"871541105056"},
{"tokengroup":"TG_Email","tokentemplate":"TT_Email","token":"tKZjaiTAZ@AFxLhkSKB.fq.7J"},
{"tokengroup":"TG_Other","tokentemplate":"TT_Other","token":"#72801269830316#"}

```

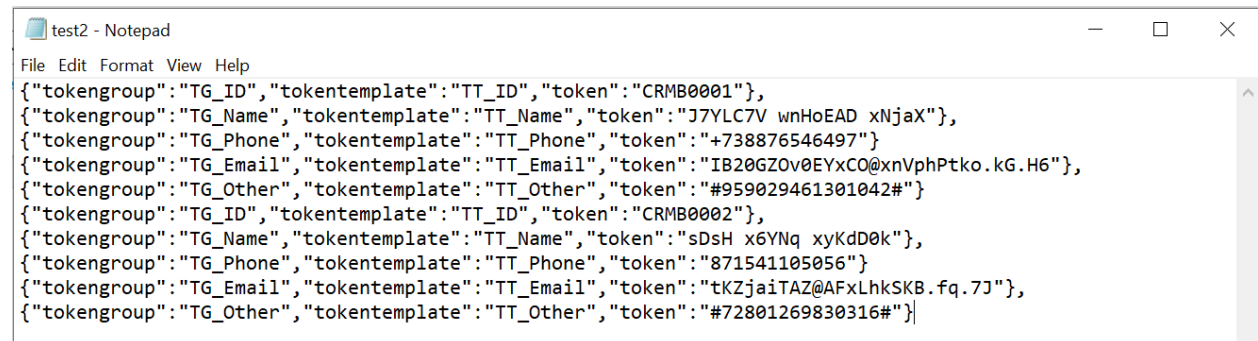
Integration Guide - Data Protection Platform

Expected respons

```
{
  "data": "CRMB0001",
  "status": "Succeed"
},
{
  "data": "Sanjaya Pratama Putra",
  "status": "Succeed"
},
{
  "data": "+629564456456",
  "status": "Succeed"
},
{
  "data": "sanjayapratama@telkomsel.co.id",
  "status": "Succeed"
},
{
  "data": "#343545356343453#",
  "status": "Succeed"
},
{
  "data": "CRMB0002",
  "status": "Succeed"
},
{
  "data": "Raja Putra Perdana",
  "status": "Succeed"
},
{
  "data": "086234534732",
  "status": "Succeed"
},
{
  "data": "rajaperda@telkomsel.co.id",
  "status": "Succeed"
},
{
  "data": "#34652645642355#",
  "status": "Succeed"
}
}
```

Using Curl command

`curl --tlsv1.2 -k --key client.pem --cert client.cer -X POST -d'<data>' https:// dppctstbsdapp02/vts/rest/v2.0/detokenize/`



The screenshot shows a Notepad window titled 'test2 - Notepad' with a JSON response. The JSON contains an array of objects, each with 'tokengroup', 'tokentemplate', and 'token' fields. The 'token' values are the same as the 'data' values in the 'Expected respons' section.

```
{
  "tokengroup": "TG_ID", "tokentemplate": "TT_ID", "token": "CRMB0001",
  "tokengroup": "TG_Name", "tokentemplate": "TT_Name", "token": "J7YLC7V wnHoEAD xNjaX"},
  "tokengroup": "TG_Phone", "tokentemplate": "TT_Phone", "token": "+738876546497"},
  "tokengroup": "TG_Email", "tokentemplate": "TT_Email", "token": "IB20GZOv0EYxCO@xnVphPtko.kG.H6"},
  "tokengroup": "TG_Other", "tokentemplate": "TT_Other", "token": "#959029461301042#"},
  "tokengroup": "TG_ID", "tokentemplate": "TT_ID", "token": "CRMB0002",
  "tokengroup": "TG_Name", "tokentemplate": "TT_Name", "token": "sDsH x6YNq xyKdD0k"},
  "tokengroup": "TG_Phone", "tokentemplate": "TT_Phone", "token": "871541105056"},
  "tokengroup": "TG_Email", "tokentemplate": "TT_Email", "token": "tKZjaiTAZ@AFxLhksKB.fq.7J"},
  "tokengroup": "TG_Other", "tokentemplate": "TT_Other", "token": "#72801269830316#"
}
```

Integration Guide - Data Protection Platform

```
22/11/2023 16:38.06 /home/mobaxterm/Desktop/Token curl -k -X POST -u 'dppSvcCRM:T3lkomsel@123' --data-binary @te
st2.txt https://10.11.8.228/vts/rest/v2.0/detokenize
{"data": "CRMB0001", "status": "Succeed"}, {"data": "Sanjaya Pratama Putra", "status": "Succeed"}, {"data": "+629564456456", "status": "Succ
eed"}, {"data": "sanjayapratama@telkomsel.co.id", "status": "Succeed"}, {"data": "#343545356343453#", "status": "Succeed"}, {"data": "CR
B0002", "status": "Succeed"}, {"data": "Raja Putra Perdana", "status": "Succeed"}, {"data": "086234534732", "status": "Succeed"}, {"data":
rajaperda@telkomsel.co.id", "status": "Succeed"}, {"data": "#34652645642355#", "status": "Succeed"}
```

When using postman, upload 2 client certificates from menu Setting → Certificates

The first screenshot shows a POST request to `https://10.11.8.228/vts/rest/v2.0/detokenize` with a raw JSON body. The response is a 200 OK status with a JSON body containing 11 items, each with a token group, template, token, and status.

The second screenshot shows the same POST request, but the response is a 200 OK status with a JSON body containing 40 items, each with a token group, template, token, and status.

2.4.5 API Encryption

Key used for Encryption

No	Tokenization Key
1	TT_Name
2	TT_Address
3	TT_DOB
4	TT_Email

Integration Guide - Data Protection Platform

5	TT_ID
6	TT_ID
7	TT_IP
8	TT_IP
9	TT_Phone
10	TT_MSISDN
11	TT_IMEI
12	TT_IMSI
13	TT_Finance
14	TT_Cookie_Id
15	TT_Message_Id
16	TT_Digits
17	TT_Aphanumeric
18	TT_Other

Source data:

Simply enter your data then push the encode button.

testing dppapps1

Action:

1. Encode source text/file to base64
Text: *testing dppapps1*
Encode base64: *dGVzdGluZyBkcHBhcHBzMQ==*
2. Using Algorithm: **A256CBC** with key id **TK_Other**
3. Result: *BGhrVjz1SJyg6KFawilecQ==*

```
{
  "plaintext": "dGVzdGluZyBkcHBhcHBzMQ==",
  "alg": "A256CBC",
  "kid": "TK_Other",
  "params": {
    "iv": "bWF0YW5uYXNpMWJhc2FuZw=="
  }
}
```

Response

```
{
  "ciphertext": "BGhrVjz1SJyg6KFawilecQ==",
  "tag": ""
}
```

Curl

Integration Guide - Data Protection Platform

```
curl --tlsv1.2 -k --key client.key --cert client.crt -X POST 'https://ipaddress/vts/crypto/v1/encrypt' -H 'content-type: application/json' -d '{"plaintext": "<plaintext>", "alg": "<algorithm>", "kid": "<keyID>", "params": { "iv": "<ivID>" } }'
```

sample

```
curl --tlsv1.2 -k --key ./client.key --cert ./client.crt -X POST 'https://dppctstbsdapp02/vts/crypto/v1/encrypt' -H 'content-type: application/json' -d '{
```

```
  "plaintext": "testing dppapps1,
  "alg": "A256CBC",
  "kid": "TK_Other",
  "params": {
    "iv": "bWF0YW5uYXNpMWJhc2FuZw=="
  }
}'
```



```
24/11/2023 11:36:13 /drives/c/Users/22345357/Downloads curl -k -u 'dppSvcDuGam:T3lkomse1@123' -X POST 'https://10.38.45.153/vts/crypto/v1/encrypt' -H 'content-type: application/json' --data @encryp.yaml {"ciphertext": "BGhrVjz1SJyg6KFawilecQ=", "tag": ""}
```

Using Postman



Integration Guide - Data Protection Platform

2.4.6 API Decryption

Key used for Decrypt must be same when doing encrypt.

Action:

1. Source text/file encrypted base64
Encrypted base64: *BGhrVjz1SJyg6KFawilecQ==*
2. Using Algorithm: **A256CBC** with key id **TK_Other**
Response: *dGVzdGluZyBkcHBhcHBzMQ==*
3. Decode base64
Source: *dGVzdGluZyBkcHBhcHBzMQ==*
Result: *testing dppapps1*

```
{
  "ciphertext": "BGhrVjz1SJyg6KFawilecQ==",
  "alg": "A256CBC",
  "kid": "TK_Other",
  "params": {
    "iv": "bWF0YW5uYXNpMWJhc2FuZw=="
  }
}
```

Response

```
{
  "plaintext": "dGVzdGluZyBkcHBhcHBzMQ=="
}
```

Using curl command

curl --tlsv1.2 -k --key client.key --cert client.crt -X POST 'https://ipaddress/vts/crypto/v1/decrypt' -H 'content-type: application/json' -d '{"ciphertex": "<ciphertext>", "alg": "<algorithm>", "kid": "<keyID>", "params": { "iv": "<ivID>" } }'

sample

curl --tlsv1.2 -k --key ./client.key --cert ./client.crt -X POST 'https://dppctstbsdapp02/vts/crypto/v1/decrypt' -H 'content-type: application/json' -d '

```
{
  "ciphertex": "dGVzdGluZyBkcHBhcHBzMQ==",
  "alg": "A256CBC",
  "kid": "TK_Other",
  "params": {
    "iv": "bWF0YW5uYXNpMWJhc2FuZw=="
  }
}
```

Integration Guide - Data Protection Platform



Using Postman



Decode base64 to text

Decode from Base64 format

Simply enter your data then push the decode button.

dGVzdGluZyBkcHBhcHBzMQ==

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

< DECODE > Decodes your data into the area below.

testing dppapps1

Integration Guide - Data Protection Platform

Note:

When encrypted base64 is directly to decrypted will have different value from the original.

Decode from Base64 format

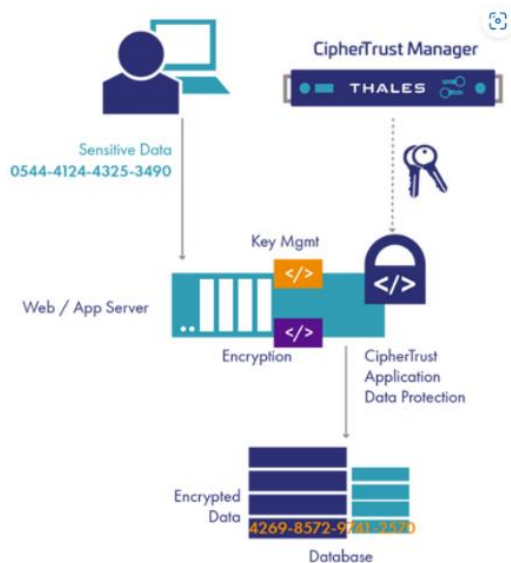
BGhrVjz1SJyg6KFawilecQ==

UTF-8 Source character set.

< **DECODE** > Decodes your data into the area below.

hkV<HZZ)^q

2.5 CADP



Deploy CADP for Java using installer

1. Download the provider from our customer support site. The software adheres to the following naming convention:

Product Name - Product Version - File Format
CADP_for_JAVA_v8.14.1.000.zip

2. Extract the file using any standard archive utility.
Extracting the archive creates the following directory structure:

Integration Guide - Data Protection Platform

```
+-----documentation
|           +---javadoc
|           +---vaultless_tokenization
+-----lib
|   +---cseg
|   +---ext
|   +---utilities
|   +---webservices
```

For information on directory structure and its content, refer to Directory Structure

3. Navigate to the directory where the CADP for Java is extracted.
4. Navigate to the CADP_for_JAVA/lib/ext directory and run the following command to install CADP for Java.

```
java -jar CADP_for_JAVA_Installer-8.14.1.000.jar
```

The license agreement is displayed on the console.

5. Agree to the license terms. The installer verifies the Java version.
6. Specify the installation location for CADP for Java.
 - For Java 8, enter Yes to install CADP for Java in the default directory that is <JAVA_HOME>\lib\ext. Enter No to select a different location.
 - For Java 10 and higher versions, the installer will prompt for the installation directory.

The installer displays the following properties that are required to connect to the Key Manager.

- NAE_IP.1
- NAE_Port
- Log_File

7. Enter Yes to update these properties. Enter No to continue with the existing values. The installation setup is complete; now, you need to perform Post Installation Steps.

Here are CADP directory structure.

Integration Guide - Data Protection Platform

filename	Description
commons-codec-1.15.jar	Apache Commons reusable Java components. Apache Commons Codec (TM) software provides implementations of common encoders and decoders such as Base64, Hex, and others.
commons-lang3-3.12.0.jar	Apache Commons reusable Java components. Lang provides methods for manipulation of core classes in the standard Java library, utilities for the java.lang API, and helps with building methods, such as hashCode, toString and equals.
commons-collections4-4.4.jar	Apache Commons reusable Java components. Extends or augments the Java Collections Framework.
gson-2.10.1.jar	A Java library to convert JSON to Java objects and vice versa.
guava-31.1-jre.jar	Guava is a suite of core and expanded libraries that include utility classes, Google's collections, I/O classes, and much more. This library is used for key caching.
failureaccess-1.0.1.jar	Contains com.google.common.util.concurrent.internal.InternalFutureFailureAccess and InternalFutures classes.
CADP_for_JAVA.properties	CADP for Java's configuration file.
CADP_for_JAVA-8.14.1.000.jar	The Java components of the CADP for Java (Ingrian) Provider. Required for all installations.
CADP_for_JAVA_Installer-8.14.1.000.jar	Automates the CADP for Java installation by placing the CADP for Java jar files at default Java path or user provided path and setting basic Key Manager configuration parameters such as IP, Port, and Log path.
cryptodatautility.jar	CADP for Java utility used to decrypt a string without specifying the keyName and algorithm.
license.rtf	CADP for Java license file.
NOTICE.txt	CADP for Java's open source notice.
log4j-core-2.19.0.jar	The Apache Log4j ImplementationLogging application.
log4j-api-2.19.0.jar	Provides the interface that applications should code to and provides the adapter components required for implementers to create a logging implementation.
bcprov-jdk15to18-1.71.jar	Required for AES/GCM and SEED algorithms when symmetric cache is enabled.
bcpkix-jdk15on-1.70.jar	Bouncy castle jar required for format conversion of EC keys.
bcutil-jdk15on-1.70.jar	Bouncy Castle Java APIs required for ASN.1 extension and utility APIs to support bcpkix and bctls.
SafeNetVaultlessTokenization.properties	Contains the parameters used for tokenization. Required for all installations.
SafeNetVaultlessTokenization-8.10.0.000.jar	Contains the Java components required for tokenization. Required for all installations.
unicode.properties	Contains the parameters required to tokenize Unicode characters.
migration.properties	Contains the parameters used to set up the bulk migration feature.

Integration Guide - Data Protection Platform

filename	Description
detokenization.properties	Contains the parameters used to set up the bulk detokenization feature.

Important notes

1. In case of Java 10, the `java.se.ee` module needs to be added as a JVM argument.
2. In case of Java 11 or higher versions, download and add the path of the following jar files in the classpath:
 - `activation-1.1.1`
 - `jaxb-api-2.3.1`
 - `jaxb-core-2.3.0.1`
 - `jaxb-impl-2.3.1`
3. For Java 8, the following jar files are OSGI compliant:
 - `cryptodatautility.jar`
 - `CADP_for_JAVA-8.14.1.000.jar`

These jar files access some non-public APIs, user needs to enable the `org.osgi.framework.bootdelegation=sun.,com.sun.` property in the OSGI framework to provide access to these APIs.

4. User can configure an external logger to capture logs instead of the default logging jar files `log4j-core-2.19.0.jar` and `log4j-api-2.19.0.jar`.

2.5.1 CADP Sample

Sample of Java API for tokenization and detokenization using CADP for Java

```
String naeUser="vuser";
char[] naePswd="xxxxxxx".toCharArray();
String keyName="token_key";
TokenServiceVaultless ts=new TokenServiceVaultless(naeUser, naePswd, keyName);
TokenSpec spec=new TokenSpec();
spec.setFormat(ts.FIRST_SIX_TOKEN);
spec.setClearTextSensitive(false);
spec.setNonIdempotentTokens(false);
spec.setGroupID(1);
String token=ts.tokenize("3646436545758756", spec);
```

Integration Guide - Data Protection Platform

```
System.out.println(token);
String value=ts.detokenize(token/*as returned above*/, spec);
System.out.println(value);
ts.closeService();
```

Java API sample demonstrating tokenization and detokenization of Unicode characters

```
String naeUser="vuser";
char[] naePswd="xxxxxxx".toCharArray();
String keyName="token_key";
AlgoSpec algospec=new AlgoSpec();
algospec.setVersion(1);
algospec.setUnicode(AlgoSpec.UNICODE_HIRAGANA);
TokenServiceVaultless ts=new
TokenServiceVaultless(naeUser,naePswd, keyName,algospec);
TokenSpec spec=new TokenSpec();
spec.setFormat(ts.TOKEN_ALL);
spec.setGroupID(1);
// HIRAGANA Charset
String token=ts.tokenize("\u3042\u3064\u3076\u3089\u3075\u3090\u305F";, spec );
System.out.println("Tokenized Value: "+ token);
String detoken_value=ts.detokenize(token, spec);
System.out.println("Detokenized Value: "+ detoken_value);
ts.closeService();
```

Sample Encrypting and Decrypting Large Files

To handle files that are larger than the payload limit of 5 MB, use the following script to automatically break down a large file into a chain of smaller files for encryption, and to join these files and decrypt them

```
#!/bin/bash
URL=http://localhost:8000/crypto/v1
KEY=cc01
IV="MTAwMDIwMDAzMDAwNDAwMA=="
CHUNKSIZE=256
FILE="/etc/hosts"
ALG="A256CTR"
BASE=`basename $FILE`
OIV=$IV
if [ ! -e ./${BASE} ]
then
cp $FILE .
fi
rm -f ${BASE}-* *-${BASE}-*
split -b $CHUNKSIZE ./${BASE} ${BASE}-
NOMORE=0
echo "===== Encryption ====="
for i in ${BASE}-*
do
if [ $NOMORE -eq 1 ]
```


Integration Guide - Data Protection Platform

```

        then
            break
        fi
    PTEXT=`cat $i | base64 -w 0`
    PAYLOAD=$(jq -n -r \
        --arg kid $KEY \
        --arg iv $IV \
        --arg pt "$PTEXT" \
        --arg alg $ALG \
        '{ plaintext: $pt, alg: $alg, kid: $kid, params: {iv: $iv} }'
    )
    RESULT=`echo $PAYLOAD | curl -Ss -X POST -k -d @- -H "Accept: application/json"
    $URL/encrypt`
    NEWIV=`echo $RESULT | jq -r '.params.iv'`
    CTEXT=`echo $RESULT | jq -r '.ciphertext'`
    if [ -z "$NEWIV" ] || [ "$NEWIV" == "null" ]
    then
        NOMORE=1
    fi
    IV=$NEWIV
    echo -n $CTEXT | base64 -d -w0 > ${i}.enc
done
cat ${BASE}/*.enc > ./${BASE}.enc
split -b $CHUNKSIZE ./${BASE}.enc enc-${BASE}-
IV=$OIV
NOMORE=0
echo "===== Decryption ====="
for i in enc-${BASE}-*
do
    if [ $NOMORE -eq 1 ]
    then
        break
    fi
    CTEXT=`cat $i | base64 -w0`
    PAYLOAD=$(jq -n -r \
        --arg kid $KEY \
        --arg iv $IV \
        --arg ct "$CTEXT" \
        --arg alg $ALG \
        '{ ciphertext: $ct, alg: $alg, kid: $kid, params: {iv: $iv} }'
    )
    RESULT=`echo $PAYLOAD | curl -Ss -X POST -k -d @- -H "Accept: application/json"
    $URL/decrypt`
    NEWIV=`echo $RESULT | jq -r '.params.iv'`
    PTEXT=`echo $RESULT | jq -r '.plaintext'`
    if [ -z "$NEWIV" ] || [ "$NEWIV" == "null" ]
    then
        NOMORE=1
    fi
    IV=$NEWIV
    echo -n $PTEXT | base64 -d -w0 > ${i}.dec

```

Integration Guide - Data Protection Platform

```
done
cat enc-${BASE}/*.dec > ${BASE}.dec
diff ${BASE}.dec ${BASE}
echo diff status = $?
```

The large file is broken into chunks of size whose modulo 16 equals 0 (that is, into chunk sizes that are multiples of 16 bytes).

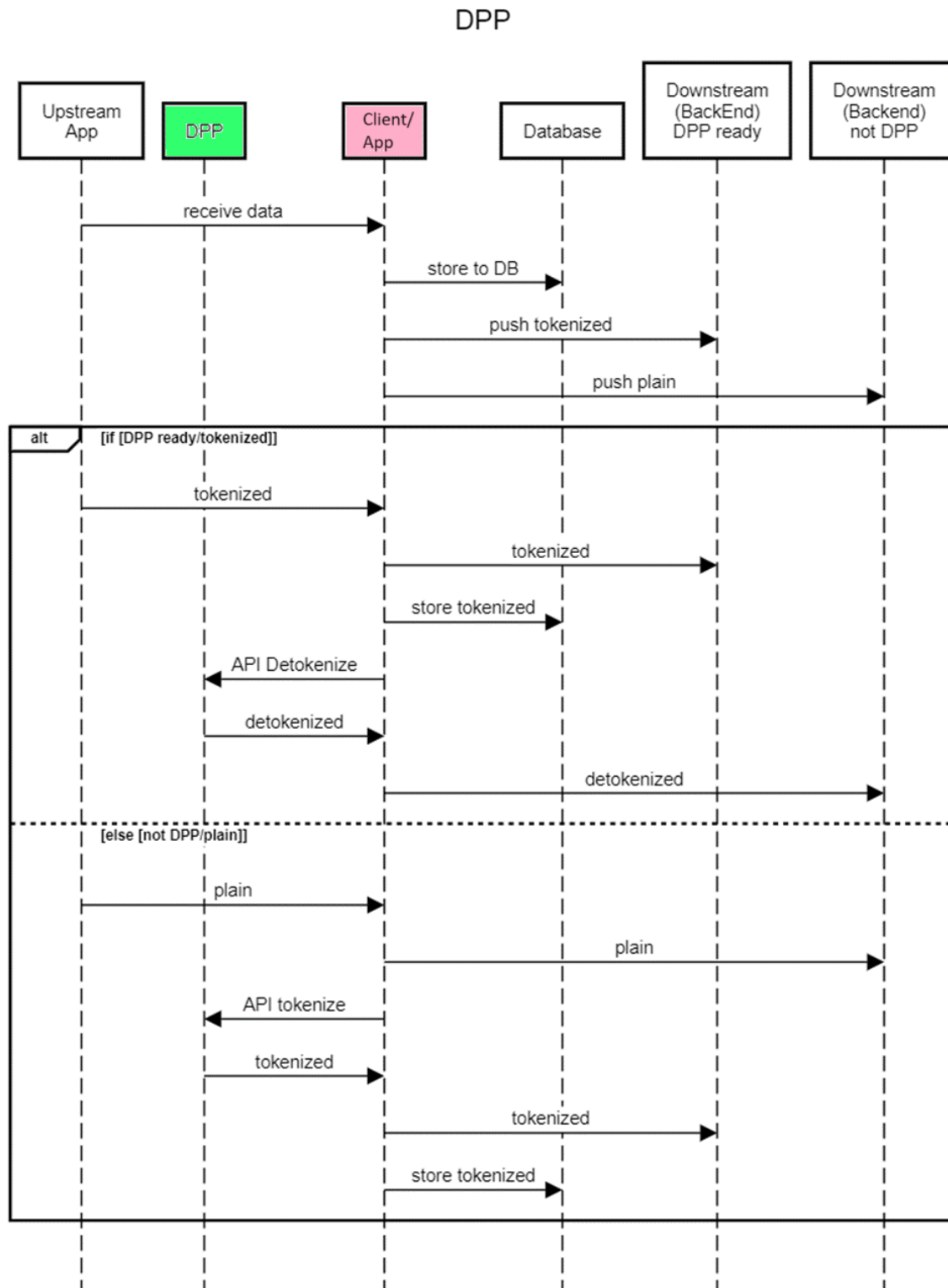
When submitting the chunks to the encryption endpoint, the response includes an IV that is used to encrypt the next chunk. If the payload length is not 16-byte aligned, the response does not return the next IV, assuming that chaining is complete.

For decryption, the same rules apply. The first call must use the same IV as the call used for encrypting. The ciphertext chunk size can be a different size from the original plaintext encryption request, as long as the chunks are 16-byte aligned.

For example a 1030 byte file can be encrypted in 8 chunks of 128 bytes each and a last one of 6 bytes. The same file once encrypted can be split in 3 chunks of 320 bytes and a last one of 70 bytes.

3 Integration Flow

Bellow are application that integrated into DPP with flow process as shown in diagram below.



4 Integration Approach

Following diagram will show about high level application integration approach. There are 3 major use case application flow as shown in following diagram.

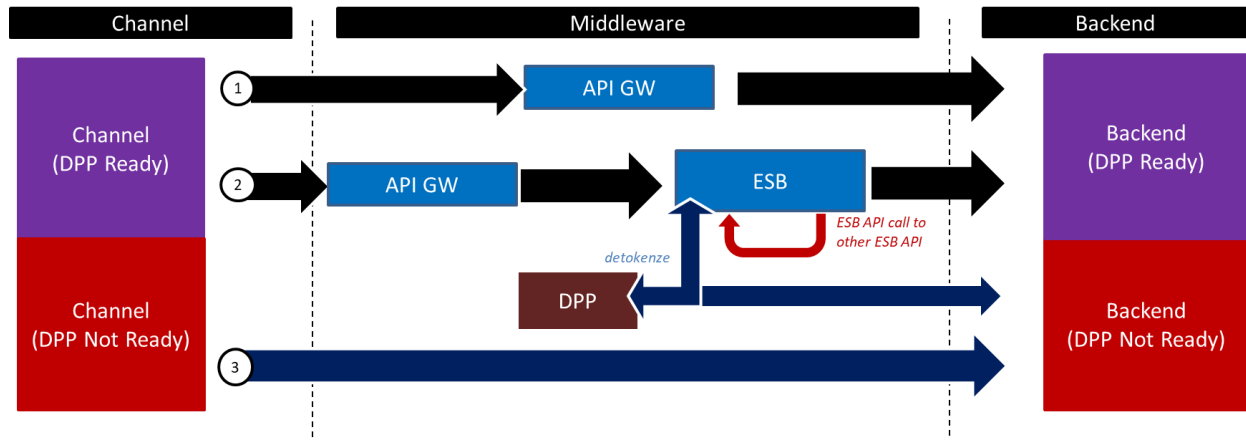


Figure 1 high level application integration approach

Telkomsel have determined the approach for implementing data tokenization and encryption and have also finalized the sequence for onboarding applications onto the DPP platform. The implementation approach is to tokenize or encrypt the personal sensitive data promptly after provided by the customer personal. Subsequently, this tokenized or encrypted data will be:

- securely stored in a persistent data repository, and
- transmitted to downstream applications in a tokenized or encrypted format.

This approach necessitates a precise implementation sequence, as the downstream applications must first be capable of receiving data in tokenized or encrypted formats before the channel applications can transmit such data to them. In the absence of a well-defined sequence, channel applications may need to make multiple changes at different time based on the varying readiness of downstream applications.

4.1 Introducing new HTTP Header

HTTP header (T-Status) will be used to indicate if payload contain tokenized data or plain data.

T-Status: plain|secure;enc=<category>

HTTP Header	Payload
T-Status not present	Plain
T-Status: payload=plain	Plain
T-Status: payload=secure;enc=name,address	Sample: Name and address data type are tokenized Refer to PII data type for that parameter

Integration Guide - Data Protection Platform

Telkomsel downstream API's invocation are currently being validated and sanitized by either (1) API GW or (2) by both API GW and ESB.

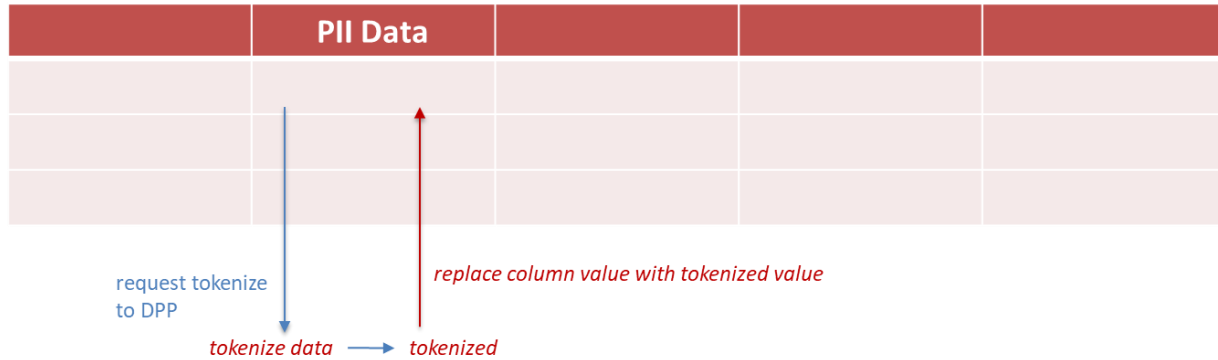
Following Application State Action based on HTTP Header

No	Channel	HTTP Header	Payload	API GW	ESB	Backend
1	DPP Not Ready	T-Status not present	Plain	Forward data	Forward data to backend	DPP Not Ready
					Forward data to backend	DPP Ready (<i>perform Tokenize</i>)
2	DPP Ready	T-Status: payload =plain	Plain		Forward data to backend	DPP Not Ready
					Forward data to backend	DPP Ready (<i>perform Tokenize</i>)
3	DPP Ready	T-Status: payload =secure; enc =name,address	Name and address data type are tokenized (refer to data type table for other data type)	Forward header	Detokenize before send to backend	DPP Not Ready
					Forward data to backend	DPP Ready

If Channel or any upstream application could not use HTTP header, it can use another reference attribute, such as **intRef** attribute according to the specification given by channel. Utilizing this attribute to distinguish between tokenized and plain data within the submitted API.

5 Data Migration

DPP proposes to application developer providing flag to each of row to indicate data is in plain or tokenized value.



Introduce channelID or header when receiving data from Upstream to indicate that incoming data is in plain or tokenized.

6 Appendix