Digital Receipt API

Version 1.0

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process, we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page https://www.omg.org, under Specifications, Report a Bug/Issue.

Table of Contents

Preface

About the Object Management Group

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Meta-model); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *https://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Formal Specifications are available from this URL: *https://www.omg.org/spec*

All of OMG¨'s formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
9C Medway Road, PMB 274
Milford, MA 01757
USA

Tel: +1-781-444-0404
Fax: +1-781-444-0320

Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO/IEC standards. Please consult: http://www.iso.org

Issues

The reader is encouraged to report and technical or editing issues/problems with this specification to:
https://www.omg.org/report_issue.htm

This page intentionally left blank.

1 Scope

1.1 Objective

This document is a specification document on the standard Digital Receipt API, which has been formulated as a standard specification by the Digital Receipt Subcommittee of the .NET Retail System Council Japan as part of a contract project of the Ministry of Economy, Trade and Industry of Japan.

This specification can connect POS terminals for various business types of business of various vendors by standardizing APIs related to transmission and reception of digital receipt data between digital receipt server and POS terminal, and between digital receipt server and smartphone. The purpose is to reduce development costs and system integration costs. Another object is to make it easier to use digital receipt data by making it easier to create a smartphone application that uses digital receipt data.

1.2 Team mission

The team's mission is to create a standard API for digital receipts in today's retail environment, where transactions at the point of sale need to be completely paperless, and to facilitate the adoption of digital receipts by enabling them to be applied to modern technology environments using JSON and REST.

2 Conformance

This section intentionally left blank. Conformance is to be interpreted or asserted in relation to this specification as a whole.

3 References

3.1 Normative References

[Digital Receipt Ontology Files]

- Retail Industry Ontology (RIO) Mapping for Digital Receipt_06162023.docx
- Digital Receipt Data Items Reference List for Japanese Market.xlsx

[Digital Receipt Format and API Specification Files]

- Digital Receipt Format Specification (JSON Ver for Japanese Market)_08262023.docx
- Digital Receipt V1_API_Specification_08262023.docx

3.2 Non-normative References

[Machine Consumable]

Open API (YAML Oriented) Files

- DigitalReceiptServiceAPISpecification_V1.0.0_08262023.yaml
- Digital Receipt nodejs-server-08262023.zip

- Digital Receipt html-client-08262023.zip

Digital Receipt JSON and XML Schema Files

- DigitalReceiptV3.1.0.JSON_Schema_08262023.json
- DigitalReceiptV3.1.0_08262023.xsd

[Ancillary]

Explanation & Inventory Files

- Digital Receipt API Inventory File_08262023.doc
- Digital Receipt API V1 Explanation_08262023.pptx

[Ancillary Reference from ARTS]

Digital Receipt Related Current ARTS Files

- ARTS_Standard_XML_Digital_Receipt_Technical_Specification_V3.1.0.pdf
- ARTS Standard Digital Receipt Domain Model V3.1.0.pdf
- Best Practices for Service implementation Using ARTS Std.pdf

4 Terms and Definitions

There are no terms for which unique definitions are given in this document.

5 Symbols

5.1 Abbreviations

The following abbreviations are used throughout this specification:

API-Application Program Interface

ARTS-Association for Retail Technologies Standards

ID-Identification

JSON-JavaScript Object Notation

NRF-National Retail Federation

POS-Point of Sales

REST-Representational State Transfer

URL-Uniform Resource Locator

URI-Uniform Resource Identifier

XML-Extensible Markup Language

6 Additional Information

6.1 Acknowledgements

- The following company submitted this
  specification: Toshiba GCS
- The following organization contributed this specification Leader
- NET Retail System Council
  "Digital receipt subcommittee"
- Cooperating organization
  General Federated Open Food Service Systems Consortium
  "Digital receipt subcommittee"
  General Association Fintech
  Association "Digital receipt
  subcommittee"
  General Association XBRL Japan

6.2 Member Roster

Chairperson:

Nobuyuki Takahashi	Toshiba Tec Corporation
Masanori Sambe (Co-Chair)	Toshiba Tec Corporation

Contributor:

Daisaku Kashima	Dai Nippon Printing Co., Ltd
Yasuo Sakami	Foresight Co., Ltd.
Yoshio Yamada	FUJITSU LIMITED
Akira Matsuyoshi	Insight Corporation
Koji Sasaki	LOG NOTE Co., Ltd.
Soichi Fujii	Microsoft Japan Co., Ltd.
Akio Tajima	NCR Japan, Ltd.
Takuro Miyuki	NEC Platforms, Ltd.
Kazutoshi Ota	OK Systems
Tadashi Furuhata	Microsoft Japan Co, Ltd.
Takashi Wakatsuki	Sharp Marketing Japan Corporation
Yusuke Udagawa	SHARP CORPORATION
Takao Tamura	Transaction Media Networks Inc.
Junichi Kobayashi	Sorimachi Giken Co., Ltd
Kenichi Nagai	Star Marketing Japan Co.,Ltd.
Masaaki Morishita	STAR MICRONICS CO., LTD.
Ayako Shimazaki	GS1 Japan
Takahide Kubota	Toshiba Tec Corporation
Kenji Shimizu	Toshiba Tec Corporation
Toyohiro Yasumoto	VINX CORP.

Secretariat

Soichi Fujii	Microsoft Japan Co, Ltd.

Food Service Specification contributors:

Tadashi Furuhata	Microsoft Japan Co, Ltd.
Yoshio Ohba	Seiko Instruments Inc.
Yuu Kusama	Open Foodservice Systems Consortium (OFSC)
Kenji Oohashi	Open Foodservice Systems Consortium (OFSC)
Umetsu Noriyuki	Fujitsu Isotec Limited
Shinya Oikawa	Public university corporation Miyagi university
Kazuhiro Kobayashi	Teraoka Seiko Co., Ltd.
Masanori Murai	ASKA-T3 Co., Ltd.
Shigeru Okada	Open Foodservice Systems Consortium (OFSC)
Jun Watanabe	NEC Platforms Limited
Tsuneo Yashima	Just planning Inc.
Kiyoshi Kurokura	CITIZEN SYSTEMS JAPAN CO. LTD.
Koichi Shibata	Seiko Solutions Inc.
Ohko Fujii	MOS FOOD SERVICES, INC.
Kenichi Nagai	Star Marketing Japan Co.,Ltd.
Makoto Yoshimura	Toshiba Tec Corporation
Naonori Nagata	Toshiba Tec Corporation
Yasuo Sakami	Open Foodservice Systems Consortium (OFSC)

7 Use of Digital Receipt API

7.1 Basic Flow

The basic flow is shown in the figure below about DigitalReceiptAPI. At the beginning of the basic flow, there is a flow that DigitalReceiptMember(Consumers) use the AuthenticationAPI through the applications to obtain Access Token. Next, there is a flow to get the Digital Receipt using the Access Token.
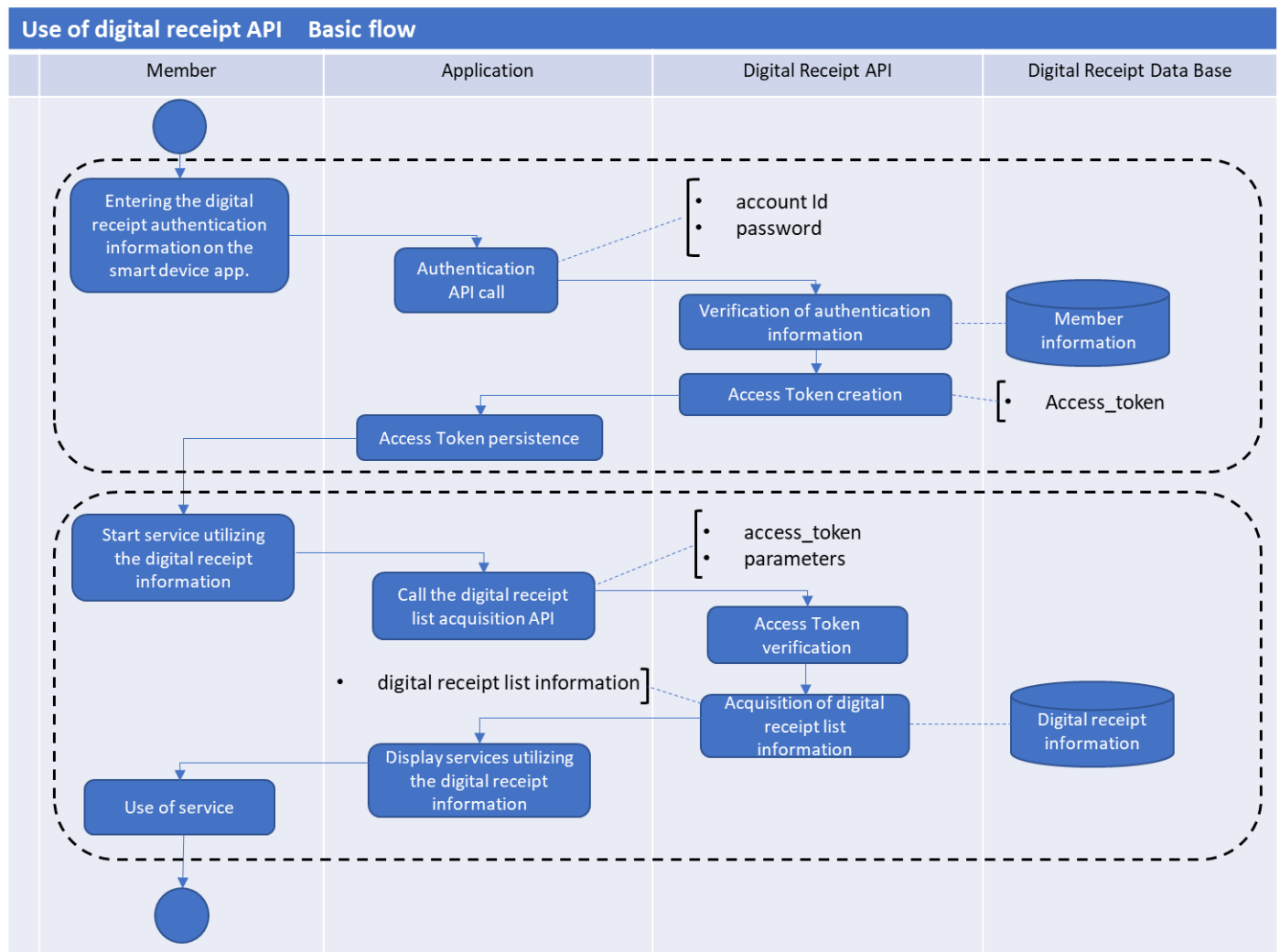
Figure 7-1: Use of Digital Receipt API Basic Flow

7.2 Exception flow

Two examples are shown below as an exception flow about Digital Receipt API.

- As a first example, the flow when authentication fails with Authentication API is shown in the figure below.

Figure 7-2: Use of Digital Receipt API Basic Flow Exception Flow No.1

- As a second example, the flow when no Access Token is set with Get Receipt List API is shown in the figure below.
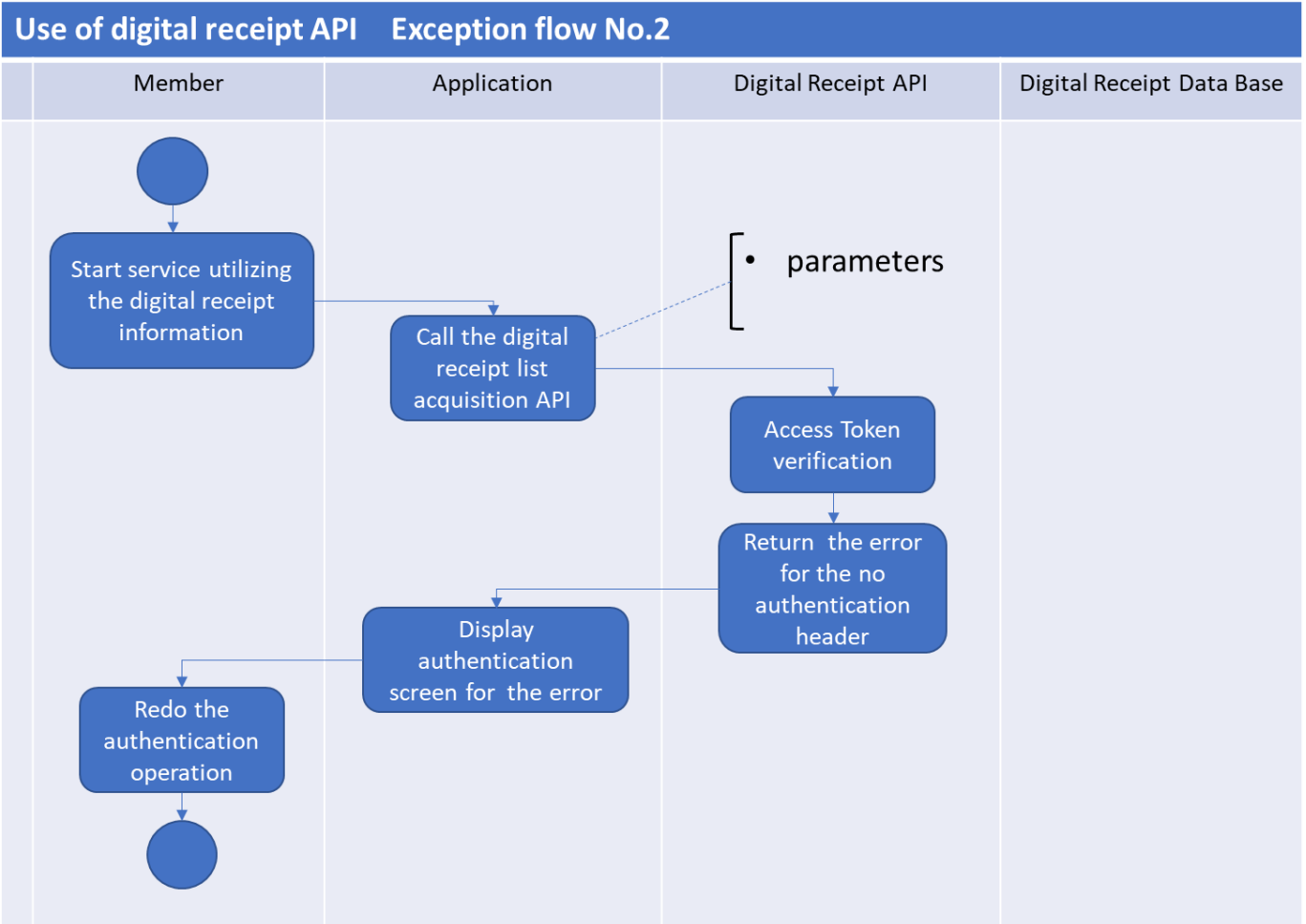
Figure 7-3: Use of Digital Receipt API Flow Exception Flow No.2

This page intentionally left blank.

8 Apply for the use of application

In order to ensure security, provide a mechanism for applications (Bookkeeping software, accounting software, health management software, etc.) to apply for Digital Receipt API usage in advance.

8.1 Authority of the Application

Applications authority are two types as below.

- Membership authority: access permission as applications of the consumer.
- Company authority: access permission as applications of the retailers etc.

8.2 Use cases

- A general use case for the Membership authority and Company authority will be introduced.

8.2.1 Get a receipt in a membership application

- Membership authority can obtain Digital Receipt.
- Both the Digital Receipts registered with the membership authority and the company authority can be obtained (pictured below).
- However, when using the Receipt Image acquisition API , only Digital Receipts registered with Company authority can be obtained.
- As the receipt Image file, html, jpg, pdf, bmp, tif and gif type files are selectable.
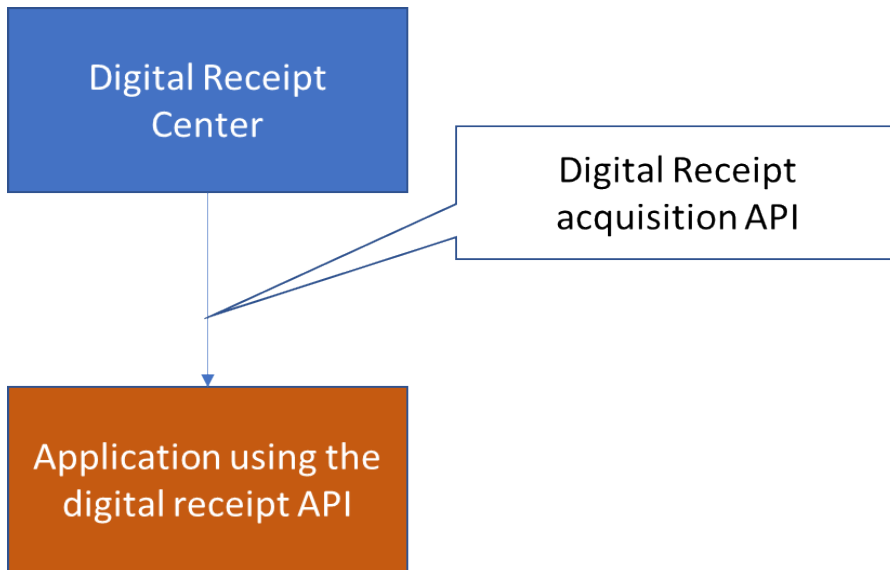
Figure 8-1: Relationship between Digital Receipt Center and App. using the Digital Receipt API

8.2.2 Register Digital Receipt by Membership authority

- Membership authority can register Digital Receipts.

- Digital Receipt registered from Membership authority will be recorded as receipts with unknown issuer sources in the Digital Receipt Center. Because it does not have the Company authority of the receipt issuing company.

- As a use case for registering Digital Receipt with Membership authority, Registration from other services not cooperating with the Digital Receipt Center, registration of the receipt independently created by the household account book software, etc. (below)
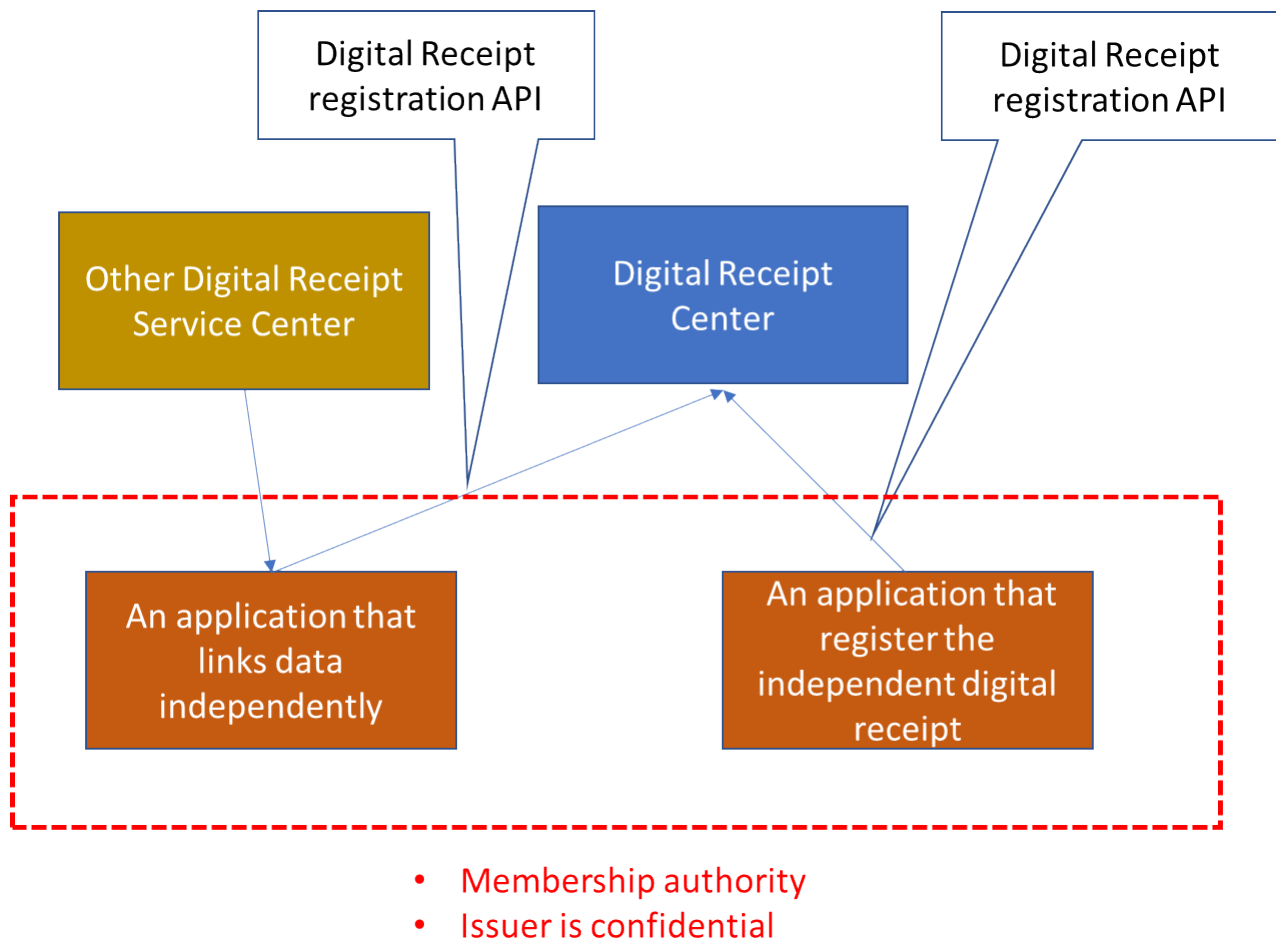
Figure 8-2: Digital Receipt Registration by Membership Authority

8.2.3   Register receipt of application with company authority

- Company authority applications can only register receipts.

- The receipt will be registered as a receipt of the trusted publisher with the Digital Receipt Center.

- As a use case to use company authority, register receipt information with the Digital Receipt Center from other services or POS described as below.

- In addition, company authority application is not allowed to obtain receipts.

- When both of "Membership Authority" and "Company Authority" application register the receipt that have a same ID, the receipt of the "Company Authority" is valid.

- In addition, when register the receipt multiple times with the same ID and same authority, the last receipt is valid.
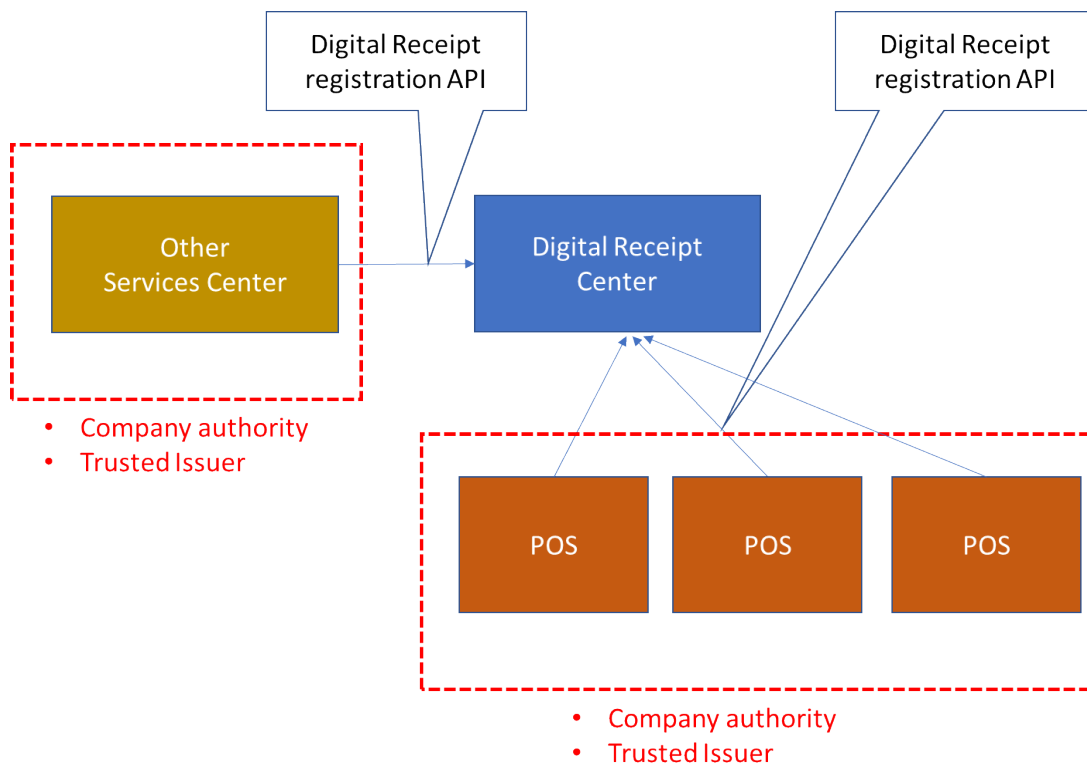
Figure 8-3: Digital Receipt Registration App. with Company Authority

8.3 Membership authority application overview

- The application of the membership authority is able to obtain the receipt of the authorized member and issue the receipt of unknown publisher.

- The receipt registered in the application of the membership authority cannot be used as public documents as the receipt Image file with time stamp.

- As the receipt Image file, html, jpg, pdf, bmp, tif and gif type files can be selected.

8.3.1 Application information obtaining

- In order to ensure security, provide ID (application ID) and application authentication function.
- Use Application ID to identify application uniquely.
- Application authentication function uses secret strings to identify the application.

8.3.2 Access token obtaining

- When membership authority application use API, need to retrieve access token using authentication API of membership authority then use API with the access token.

8.3.3　Request example

- Authentication API example is shown below.

Table 8-1: Authentication API Example

Method	POST			
URI	/members/auth			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
applicationId	String	20	✔	Uniquely identify an application ID.
applicationSecret	String	40	✔	A secret string to identify the application.
accountId	String	1-256	✔	The phone number or email address to log in to the Digital Receipt system.
password	String	1-256	✔	The password used to log in to the Digital Receipt system.
Body Example				

```
{
  "applicationId": {APPLICATION_ID}, "applicationSecret":
  {APPLICATION_SECRET}, "accountId":
  "name@example.jp",
  "password": "password"
}
```

Need to set application ID and application secret that is published / managed by System Management Company to "APPLICATION_ID" and "APPLICATION_SECRET".

8.4 Company authority application overview

- A company authority application can register a receipt for an authorized company.
- The receipt registered in the application of the company authority can be used as public documents as the receipt Image file with time stamp.
- As the receipt Image file, html, jpg, pdf, bmp, tif and gif type files can be selected.

8.4.1　Application information obtaining

- Application ID is used for uniquely identifies the application. You cannot change the ID once it is registered.
- The application secret is a secret string that you use to identify your application.

8.4.2　Access token obtaining

- When company authority application use API, need to retrieve access token using authentication API of company authority then use API with the access token.

8.4.3　Request example

- Authentication API an example is shown below.

Table 8-2: Authentication API Example

Method	POST			
URI	/companies/auth			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
applicationId	String	20	✔	Uniquely identify an application ID.
applicationSecret	String	40	✔	A secret string to identify the application.
companyCode	String	13	✔	Company code.
password	String	1-256	✔	The password that is issued by the Digital Receipt system.
Body Example				

{
 "applicationId": {APPLICATION_ID}, "applicationSecret":
 {APPLICATION_SECRET}, "companyCode": "0000000000001",
 "password": "password"
 }				

Need to set application ID and application secret that is published / managed by System Management Company to "APPLICATION_ID" and "APPLICATION_SECRET"

9 API list

9.1 Digital Receipt API list

- A list of Digital Receipt APIs is shown below.

Table 9-1: A List of Digital Receipt APIs

API Name	URI	Method	Overview
Membership Authentication API *1	/members/auth	POST	It authenticates the member and issues an authentication token.
Company Authentication API *1	/companies/auth	POST	It authenticates the company and issues an authentication token.
Get Receipt List API *1	/receipts	GET	Get the receipt list information for the authenticated member. Support the Pagination functions.
Get Receipt Detail API *1	/receipts/{receiptId}	GET	Get the receipt detail information for the authorized member's specified receiptId.
Get Receipt Print String API *1	/receipts/{receiptId}/ digitalReceiptPrintD ata	GET	Get the receipt print string for the authorized member's specified receiptId.
Get Receipt image API *1	/receipts/{receiptId}/ image	GET	Get the receipt of image file for the authorized member's specified receiptId
Receipt Registration API *2	/receipts	POST	Register receipt information.

*1 URI example: https://exp-openapi.sampledomain.jp/v1 Write it so as to follow after.

*2 URI example: https://exp-public-receiver.sampledomain.jp/srr Write it so as to follow after.

Special
Notes None

This page intentionally left blank.

10 Membership API

- It is an API on the premise that it is used by consumer used Digital Receipt application.

10.1 Authentication API

- Digital Receipt member certification.

- If the authentication succeeds, issue the access token of the receipt API.

- Arbitrary Token technology is available for access token.

10.1.1 Request

Table 10-1: Authentication of Membership API Request

Method	POST
URI	/members/auth
Headers	Content-Type: application/json

Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
applicationId	String	20	✔	The ID that uniquely identifies the application.
applicationSecret	String	40	✔	A secret string to identify the application.
accountId	String	1-256	✔	This is the phone number or e-mail address you use to log in to the Digital Receipt system.
password	String	1-256	✔	This is the password used to log in to the Digital Receipt system.

Body Example

{
  "applicationId": {APPLICATION_ID}, "applicationSecret":
  {APPLICATION_SECRET}, "accountId":
  "name@example.jp",
  "password": "password"
}

10.1.2 Response

Success

Table 10-2: Authentication of Membership API Success Response

Status Code	200			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
access_token	String	Undefined	✔	This is token.
Body Example				

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY
    3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2c
    Bab30RMHrHDcEfxjoYZgeFONFh7HgQ"
}
```

Error

Table 10-3: Authentication of Membership API Error Response

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
error	Object	Undefined	✔	Error object.
message	String	Undefined	✔	Error message.
type	String	Undefined	✔	Error type.
Body Example				

```
{
  "error": {
    "message": "Error message.", "type":
    "ErrorType"
  }
}
```

Error content

Table 10-4: Authentication of Membership API Error Content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	MemberAuthError	Account ID or password is incorrect.	Error in member authentication information.
403	AccountLockOutError	Account is locked out.	Account lockout due to continuous failure of membership authentication.
401	ApplicationAuthError	Application ID or application secret is incorrect.	Error in application authentication information.

Special Notes
None

This page intentionally left blank.

11 Company API

- It is an API based on the assumption that it is used by company applications (POS etc.).

11.1 Authentication API

- The company authenticates.
- If the authentication succeeds, issue an authentication token of the receipt API. Arbitrary Token technology is available for access token.

11.1.1 Request

Table 11-1: Authentication of Company API Request

Method	POST
URI	/companies/auth
Headers	Content-Type: application/json
Body	

Property Name	Type	Size	Required	Remarks
applicationId	String	20	✔	Uniquely identify an application ID.
applicationSecret	String	40	✔	A secret string to identify the application.
companyCode	String	13	✔	The company codes.
password	String	1-256	✔	The password that is issued by the Digital Receipt system.

Body Example

{
  "applicationId": {APPLICATION_ID}, "applicationSecret":
  {APPLICATION_SECRET}, "companyCode":
  "0000000000001",
  "password": "password"
}

11.1.2 Response

Success

Table 11-2: Authentication of Company API Success Response

Status Code	200			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
access_token	String	Undefined	✔	This is token.
Body Example				

{
 	
 "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY 3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2c Bab30RMHrHDcEfxjoYZgeFONFh7HgQ"
 	
 }				

Error

Table 11-3: Authentication of Company API Error Response

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
error	Object	Undefined	✔	Error object.
message	String	Undefined	✔	Error message.
type	String	Undefined	✔	Error type.
Body Example				

{
 "error": {
   "message": "Error message.", "type":
   "ErrorType"
 }
 }				

Error content

Table 11-4: Authentication of Company API Error Content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	CompanyAuthError	Company code or password is incorrect.	Error in company authentication information
401	ApplicationAuthError	Application ID or application secret is incorrect.	Error in application authentication information

This page intentionally left blank.

12 Receipt API

- It is the API concerning acquisition of Digital Receipt information.

12.1 API to get receipt list

- We will get Digital receipt list information of members who have been authenticated by Token.

- Query parameters can be used as a narrowing condition of list information.

- Pagination is supported.

- When using the pagination function, the Link header provides link information to the first page, the previous page, the next page, and the last page.

12.1.1 Request

Table 12-1: Digital Receipt Info. Acquisition API Request

Method	GET
URI	/receipts
Headers	Authorization: Bearer {Token}
Query Parameters	

Parameter name	Type	Size	Required	Remarks
dateTimeFrom	DateTime	Indefinite	N/A	Specify the transaction date and time after the specified date and time. Corresponds to "ReceiptDateTime" in standard DigitalReceipt format and it is ISO8601 formatted datetime value.
dateTimeTo	DateTime	Indefinite	N/A	Specify the transaction date and time before the specified date and time Corresponds to "ReceiptDateTime" in standard Digital Receipt format and it is ISO8601 formatted datetime value.

offset	Integer	Undefined	N/A	Specify the offset number of the list of receipts (the list of receipts specified by the dateTimeFrom and dateTimeTo parameters) to be retrieved. Generally, offset=0 is specified for the first time.
				If omitted, the pagination function is not supported.
limit	Integer	Undefined	N/A	Specifies the number of receipts recorded transaction after the offset parameter.
storeNumber	String	Indefinite	N/A	Specified the storeNumber parameter as "BusinessUnits" of ""UnitID":"#Value"".
companyCode	String	13	N/A	Specified the companyCode parameter as "OrganizationHierarchies" of "ID".
customerId	String	Indefinite	N/A	Specified the customer parameter as "CustomerID".

Note: companyCode and storeNumber are strongly recommend assigning and available as the digital receipt mandatory data.	

Query Parameters Example
/receipts?dateTimeFrom=2018-02-01T00:00:00+09:00&dateTimeTo=2018-02-28T23:59:59+09:00&offset=0&limit=10&companyCode=0000000000001&storeNumber=34567745778&customerId=9900000000010001

12.1.2 Response

<u>Success</u>

Table 12-2: Digital Receipt Info. Acquisition API Success Response

Status Code	200
Body	
It is an object with an array of receipt information in standard format. For the items, see the example below and the ARTS Digital Receipt Specification v3.1.0.	
Body Example	

{
  "DigitalReceiptIndex": [
    {
      "DigitalReceipt": {
        "MajorVersion": 3,
        "MinorVersion": 1,
        "FixVersion": 0,
        "ReceiptID": { "#value": "2018010110000000000001000001000000000010001"
    },

```
    "Transactions": [

      {

        "TypeCode": "SaleTransaction",
#       storeNumber information is "BusinessUnits" of ""UnitID": "#value""
        "BusinessUnits": [{

                "UnitID": {

                        "#value": "1234567890123",

          }

        }

      ],
#       companyCode information is "OrganizationHierarchies" of "ID"
        "OrganizationHierarchies": [{

                "ID": "1234567890123",

          }

      ],

      "WorkstationID": { "#value": "0001" },

      "ReceiptDateTime": { "#value": "2018-01-01T10:00:00+09:00" },

      "ReceiptNumber": { "#value": "1234" }, "RetailTransactions": [

        {

          "Totals": [

            {

              "#value": 1000,

              "TotalType": "TransactionGrandAmount"

            },

            {

              "#value": 80,

              "TotalType": "TransactionTaxAmount"

            }
```

Error

Table 12-3: Digital Receipt Info. Acquisition API Error Response

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
error	Object	Undefined	✔	Error object.
message	String	Undefined	✔	Error message.
type	String	Undefined	✔	Error type.
Body Example				

```
{
  "error": {
    "message": "Error message.", "type":
    "ErrorType"
  }
}
```

Error content

Table 12-4: Digital Receipt Info. Acquisition API Error Content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
400	RequestParametersInvalidError	Request parameters are invalid or missing.	Invalid request parameter
400	LimitIncorrectError	The specified limit value is incorrect.	The specified limit value is too large or is specified below 0. The upper limit is implementation dependent.
401	TokenMissingError	Token is required.	There is no Authorization header or Bearer scheme.
401	TokenIncorrectError	Token is incorrect.	Token is illegal.

401	TokenExpiredError	Token is expired.	Token Expired.
404	OffsetOutOfRangeE rror	The specified offset value is out of range.	No data exists at the position specified by the offset value.

Special Notes
None

12.2 Receipt Detail Acquisition API

- This allows the application to get receipt details specified by "receiptId" which is certified by Token.

- The "receiptId" means "DigitalReceipt.ReceiptID" of Receipt list acquisition API. This ID can identify a receipt. It should be issued by the system automatically. Receipt details include company information, store information, cashier number, receipt number, transaction date and time, purchase information, store logo image, and promotional bitmap image.

12.2.1 Request

Table 12-5: Digital Receipt Detail Acquisition API Request

Method	GET
Sites	/receipts/{receiptId}
Headers	Authorization: Bearer {Token}

12.2.2 Response

Normal

Table 12-6: Digital Receipt Detail Acquisition API Normal Response

Status Code	200
Headers	Content-Type: application/json
Body	
Receipt information as a standard format. For items, refer to the following examples and ARTS Digital Receipt Specification v3.1.0.	
Body Example	

```
{
 "DigitalReceipt": {
  "MajorVersion": 3,
  "MinorVersion": 1,
  "FixVersion": 0,
  "Transactions": [
```

```json
{
  "TypeCode": "SaleTransaction",
  "BusinessUnits": [
    {
      "UnitID": {
        "Name": "Branch Name",
        "#value": "1234567890123",
        "TypeCode": "RetailStore"
      }
    }
  ],
  "OrganizationHierarchies": [
    {
      "Level": "OperatingCompany",
      "ID": "1234567890123",
      "#value": "Company Name"
    }
  ],
  "Logo": {
    "LogoFormat": "JPG",
    "FileName": "https://example.jp/logo.jpg"
  },
  "WorkstationID": { "#value": "0001" },
  "ReceiptDateTime": { "#value": "2018-01-01T10:00:00+09:00" },
  "ReceiptNumber": { "#value": "1234" },
  "RetailTransactions": [
    {
      "LineItems": [
        {
          "Sale": {
            "ItemIDs": [
              {
                "Name": "Product Name",
                "#value": "4900000000010"
              }
            ],
            "MerchandiseHierarchies": [
              {
                "ID": "J1",
                "Level": "Category"
              }
            ],
            "ActualSalesUnitPrice":{"#value": 1000},
            "Quantities": [ { "#value": 1 } ]
          },
          "SequenceNumbers": [ 1 ]
        },
```

```json
      {
        "Discount": {
          "SequenceNumber": 1,
          "Amount": {
            "Action": "Subtract",
            "#value": 100
          }
        },
        "SequenceNumbers": [ 2 ]
      }
    ],
    "Totals": [
      {
        "TotalType": "TransactionGrandAmount",
        "#value": 900
      },
      {
        "TotalType": "TransactionTaxAmount",
        "#value": 72
      }
    ],
    "Customers": [
      {
        "CustomerID": "9900000000010001"
      }
    ]
  }
    ]
    }
  }
  ]
  }
}
```

Error

Table 12-7: Digital Receipt Detail Acquisition API Error Response

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property name	**Type**	**Size**	**Required**	**Remarks**
error	Object	Undefined	✔	Error object.
message	String	Undefined	✔	Error message.
type	String	Undefined	✔	Error type.
Body Example				

```
{
  "error": {
    "message": "Error message.", "type":
    "ErrorType"
  }
}
```

Error content

Table 12-8: Digital Receipt Detail Acquisition API Error Content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	TokenMissingError	Token is required.	Authorization header does not exist. Bearer scheme does not exist.
401	TokenIncorrectError	Token is incorrect.	Token is illegal
401	TokenExpiredError	Token is expired.	Token Expired
404	ReceiptNotExistError	Receipt does not exist.	The specified receipt does not exist.

12.3 Receipt print character acquisition API

- This API get the printing character string of the Digital Receipt, which is specified by the "receiptId" of the members, authenticated by Token.
  The "receiptId" indicates DigitalReceipt.ReceiptID, which is included in the response of receipt list acquisition API.

- This ID is an ID to specify the receipt uniquely, and it is assumed to be issued automatically by the system. It does not include a standard print command which is sent to the printer from a standard POS (for example, Code which is called Escape sequence etc.).

12.3.1 Request

Table 12-9: Digital Receipt Printer Character Acquisition API Request

Method	GET
URI	/receipts/{receiptId}/digitalReceiptPrintData
Headers	Authorization: Bearer {Token}

12.3.2 Response

Success

Table 12-10: Digital Receipt Printer Character Acquisition API Success Response

Status Code		200
Headers		Content-Type: application/json
	Body	
	Receipt printing character string, store logo, promotional images. Please refer ARTS Digital Receipt Specification v3.1.0 and the example below for more information.	
	Body Example	

```
{
  "DigitalReceipt": {
    "MajorVersion": 3,
    "MinorVersion": 1,
    "FixVersion":
    0,
    "Transactions":
    [
      {
        "ReceiptImages": [
          {
            "ReceiptLines":
            [ "{receipt
            text}",
            "{receipt text}",
            "{receipt text}",
            "{receipt text}",
            "{receipt text}",
            "{receipt text}",
            "{receipt
            text}",
            "{receipt
            text}"
            ]
          }
        ],
        "Logo": {
          "LogoFormat":
          "JPG",
          "FileName": "https://example.jp/logo.jpg"
        },
        "Advertising": {
          "AdvertisingID":
          "1",
          "ImageURIs": [ "https://example.jp/image2.jpg" ]
        }
      }
    ]
  }
}
```

Error

Table 12-11: Digital Receipt Print Character Acquisition API Error Response

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
error	Object	Undefined	✔	Error object.

message	String	Undefined	✔	Error message.
type	String	Undefined	✔	Error type.

Body Example

```
{
  "error": {
    "message": "Error message.", "type": "ErrorType"
  }
}
```

Error content

Table 12-12: Digital Receipt Print Character Acquisition API Error Content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	TokenMissingError	Token is required.	Authorization Header does not exist. Bearer scheme does not exist.
401	TokenIncorrectError	Token is incorrect.	Token is illegal.
401	TokenExpiredError	Token is expired.	Token Expired.
404	ReceiptNotExistError	Receipt does not exist.	The specified receipt does not exist

Special Notes
None

12.4 Receipt Image acquisition API

- This API gets the time stamp and the Image file with e-signature of the Digital Receipt, which is specified by the receiptId of the members, authenticated by Token.
- The "receiptId" indicates "DigitalReceipt.ReceiptID" which is included in the response of Receipt list acquisition API. This ID is an ID to specify the receipt uniquely, and it is assumed to be issued automatically by the system.

12.4.1 Request

Table 12-13: Digital Receipt Image Acquisition API Request

Method	GET				
URI	/receipts/{receiptId}/image				
Headers	Authorization: Bearer				
Query Parameters					
Parameter name	**Type**	**Size**	**Required**	**Remarks**	
type	String	Indefinite	Yes	As the image file query parameter, html, jpg, pdf, bmp, tif and gif files are selectable.	
Query Parameter Example					
/receipts/201801011000000000000100000100000000010001/image?type=pdf					

12.4.2 Response

Success

Table 12-14: Digital Receipt Image Acquisition API Success Response

Status Code	200
Headers	Content-Type: application/image
Body	
PDF file will be downloaded.	

Error

Table 12-15: Digital Receipt Image Acquisition API Error Response

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
error	Object	Undefined	✔	Error object.
message	String	Undefined	✔	Error message.
type	String	Undefined	✔	Error type.

Body Example

{
  "error": {
    "message": "Error message.", "type":
    "ErrorType"
  }
}

Error content

Table 12-16: Digital Receipt Image Acquisition API Error Content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	TokenMissingError	Token is required.	No Authorization header Bearer scheme does not exist.
401	TokenIncorrectError	Token is incorrect.	Token is illegal
401	TokenExpiredError	Token is expired.	Token Expired
404	ReceiptNotExistError	Receipt does not exist.	The specified receipt does not exist.
404	ReceiptUnauthenticationError	Receipt is not trusted.	The publisher of the specified receipt is unknown.

Special Notes
None

12.5 Receipt registration API

- Register the Digital Receipt.

12.5.1 Request

Table 12-17: Digital Receipt Registration API Request

Method	POST
URI	/receipts
Headers	Content-Type: application/json Authorization: Bearer {Token}
Body	
Standard Digital Receipt format in JSON.	

Body Example

```
{
  "DigitalReceipt": {
    "MajorVersion": 3,
    "MinorVersion": 1,
    "FixVersion": 0,
    "Transactions": [
      {
        "TypeCode": "SaleTransaction",
        "BusinessUnits": [
          {
            "UnitID": {
              "Name": "Store Name",
              "TypeCode": "RetailStore",
              "#value": "0000000000001"
            }
          }
        ],
        "OrganizationHierarchies": [
          {
            "#value": "Company Name", "Id":
            "0000000000001",
            "Level": "OperatingCompany"
          }
        ],
        "Logo": { "LogoFormat":
          "BMP",
          "FileName": "logo.bmp"
        },
        "WorkstationID": { "#value": "0001" },
        "ReceiptDateTime": { "#value": "2018-02-16T10:20:30+09:00" },
        "ReceiptImages": [
          {
            "ReceiptLines": [
              "{receipt text}",
              "{receipt text}",
```

```json
            "{receipt text}",
            "{receipt text}",
            "{receipt text}",
            "{receipt text}",
            "{receipt
            text}",
            "{receipt
            text}"
          ]
        }
     ],
     "ReceiptNumber": { "#value": "0001" },
     "RetailTransactions": [
       {
         "LineItems": [
           {
             "Sale": {
               "ItemIDs": [
                 {
                   "Name": "Product Name1",
                   "#value": "4900000000001"
                 }
               ],
               "MerchandiseHierarchies": [
                 {
                   "ID": "J01",
                   "Level": "Class",
                   "Name": "Category name",
                   "Type": "Retailer"
                 }
               ],
               "ActualSalesUnitPrice": { "#value": 1000 },
               "ExtendedAmount": { "#value": 2000 }, "Quantities": [ {
               "#value": 2 } ]
             },
             "SequenceNumbers": [ 1 ]

           },
           {
             "Sale": {
               "ItemIDs": [

                 {
                   "Name": "Product Name2",
                   "#value": "4900000000002"
                 }
               ],
               "MerchandiseHierarchies": [
                 {
                   "ID": "J02",
                   "Level": "Class",
               "Name": "Category name", "Type": "Retailer"
                 }
```

```
        ],
        "ActualSalesUnitPrice": { "#value": 100 }, "ExtendedAmount": {
        "#value": 100 }, "Quantities": [ { "#value": 1 } ]
        },
        "SequenceNumbers": [ 2 ]
      }
    ],
    "Totals": [
      {
        "TotalType": "TransactionGrossAmount", "#value": 2100
      },
      {
        "TotalType": "TransactionTaxAmount", "#value": 168
      }
    ]
   }
  }
 ],
 "Customers": [
   {
     "CustomerID": "1"
   }
  ]
 }
 ]
 }
}
```

12.5.2 Response

Success

Table 12-18: Digital Receipt Registration API Success Response

Status Code	200
Headers	Content-Type: text/plain

Error

Table 12-19: Digital Receipt Registration API Error Response

Status Code	400 / 401			
Headers	Content-Type: application/json			
Body				
Property Name	**Type**	**Size**	**Required**	**Remarks**
error	Object	Undefined	✔	Error object.
message	String	Undefined	✔	Error message.

type	String	Undefined	✔	Error type.

Body Example

```
{
  "error": {
    "message": "Error message.", "type":
    "ErrorType"
  }
}
```

Error content

Table 12-20: Digital Receipt Registration API Error Content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
400	ReceiptDataError	Digital Receipt is invalid data.	The data in the Digital Receipt is incorrect.
401	TokenMissingError	Token is required.	Authorization header not included. Bearer scheme not included.
401	TokenIncorrectError	Token is incorrect.	Token is incorrect.
401	TokenExpiredError	Token is expired.	Token is Expired.
401	MemberAuthError	Account ID or password is incorrect.	Incorrect membership credentials
401	CompanyAuthError	Company code or password is incorrect.	Incorrect company credentials

Special Notes
None

13 Additional References

ARTS Digital Receipt Specification v3.1.0
https://www.omg.org/cgi-bin/doc?retail/2018-04-01

Best Practices for Services Implementation Using ARTS Standards
https://www.omg.org/cgi-bin/doc?retail/2019-02-05

Digital Receipt Data Items Reference List for Japanese Market
https://www.omg.org/cgi-bin/doc?dtc/24-02-24

Digital Receipt Data Format Specification JSON Version for Japanese Market
https://www.omg.org/spec/DRAPI

 Digital Receipt RFP
https://www.omg.org/cgi-bin/doc.cgi?retail/2019-3-3

RFC 7519 - JSON Web Token (JWT)
https://tools.ietf.org/html/rfc7519

RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
https://tools.ietf.org/html/rfc6750

RFC 7235 - Hypertext Transfer Protocol (HTTP/1.1): Authentication
 https://tools.ietf.org/html/rfc7235

This page intentionally left blank.

14 Unresolved issues

There are no unresolved issues as of this publication.

This page intentionally left blank.

15 Relationship between Digital Receipt API Related Documents

OMG has two existing standards related to digital receipts.

The first set international standards for digital receipts "ARTS Digital Receipt Specification v 3.1.0".

The other specifies the method of XML-JSON conversion using the ARTS standard "Best Practices for Services Implementation Using ARTS Standard".

This specification adds the following three documents.

*Digital Receipt API specification V 1.0 (this book)

*Digital Receipt Data Items Reference List for Japanese Market (References)

*Digital Receipt Data Format Specification JSON Version for Japanese Market (References)

The existing "ARTS Digital Receipt Specification v 3.1.0" defines a set of use case scenarios that use digital receipts and defines the data format of digital receipts as XML schemas. However, the service APIs for realizing the use case scenario are undefined.

This specification proposes APIs for the digital receipt service to implement the use case scenario defined in "ARTS Digital Receipt Specification v 3.1.0" and complements it.
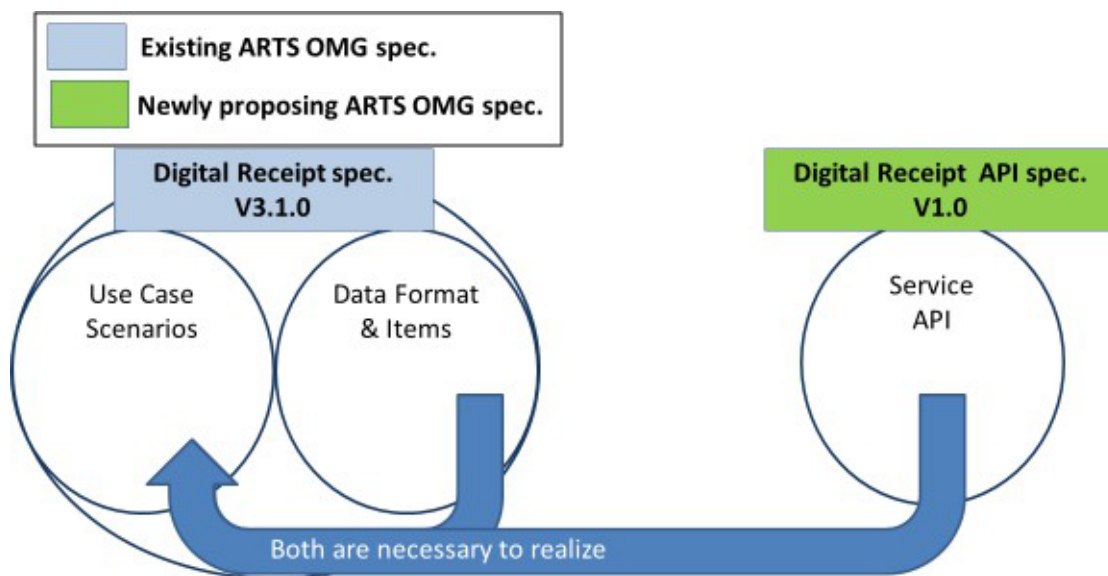

Figure 15-1: Relationship between Digital Receipt v3.1.0 and Digital Receipt API

Although XML is used as the data format in "ARTS Digital Receipt Specification v 3.1.0", as mentioned in "Digital Receipt API v 1.0 Request For Proposal", the RESTful interface and JSON data

format have recently become de facto standards for communication interfaces and data formats between cloud services and clients (Smartphone), so this specification proposes APIs using the RESTful interface and JSON data format.
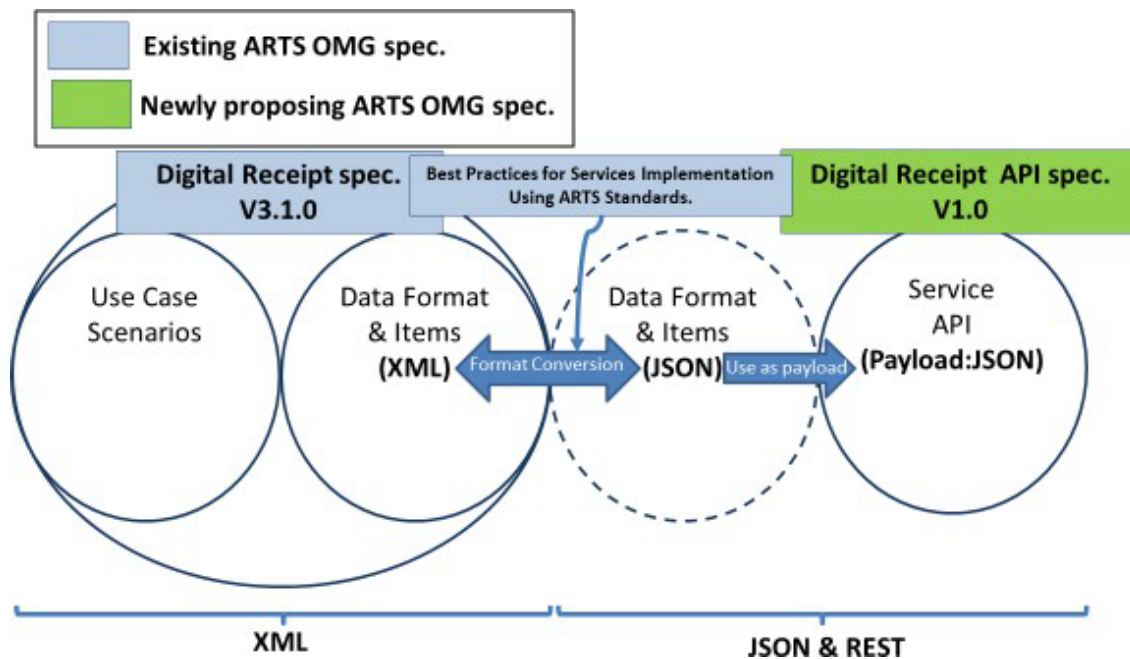

Figure 15-2: XML and Json & Rest relationship in Digital Receipt v3.1.0 and Digital Receipt API

The JSON data format used by the APIs proposed in this specification and the XML data format defined in "ARTS Digital Receipt Specification v 3.1.0" are best interconverted by "Best Practices for Services Implementation Using ARTS Standard".

The "ARTS Digital Receipt Specification v 3.1.0" data format covers international data items. The reference file "Digital Receipt Data Items Reference List for Japanese Market" is the material which extracted the required items in the Japanese market.

Similarly, "Digital Receipt Format Specification JSON Version for Japanese Market" is a document that converts the "ARTS Digital Receipt Specification v 3.1.0" data format into JSON using the "Best Practices for Services Implementation Using ARTS Standard" method, and extracts required items for the Japanese market.
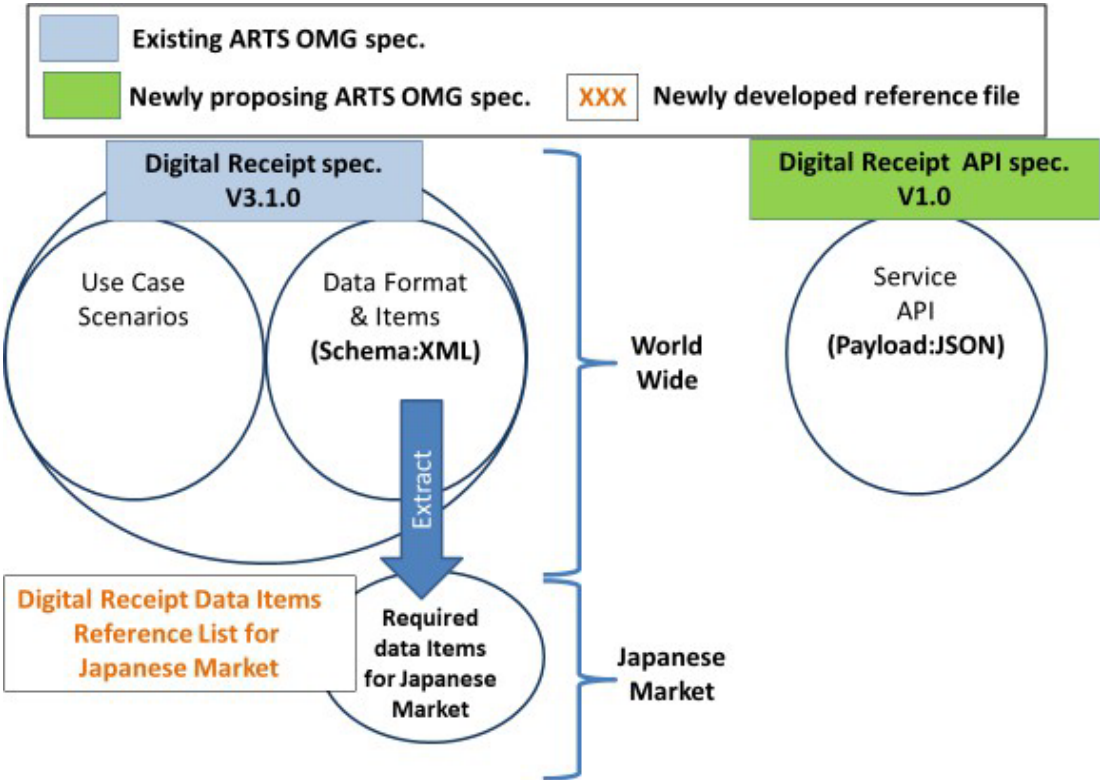
The overall relationship is shown below.

Figure 15-3: Explanation of Newly Developed Reference List for Japanese Market from the current Digital Receipt related specifications
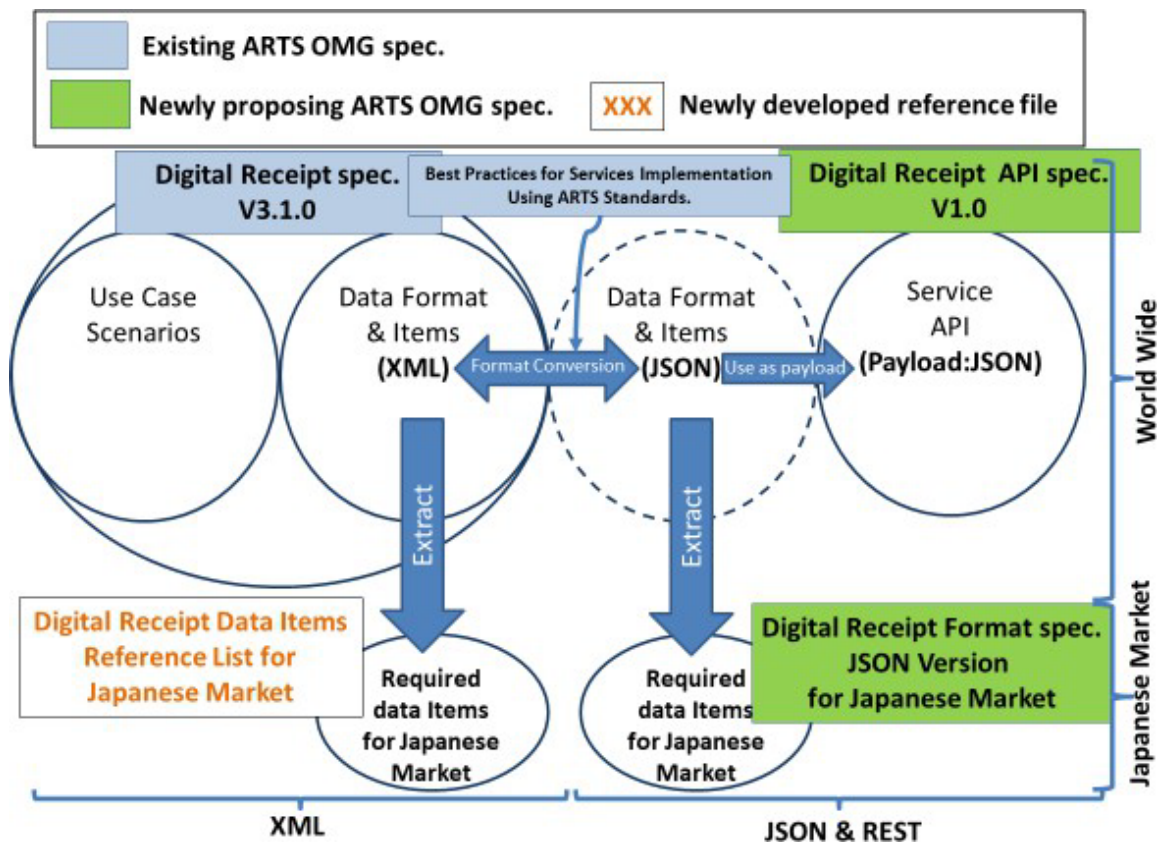
Figure 15-4: Overall relationship of Digital Receipt Related Specifications

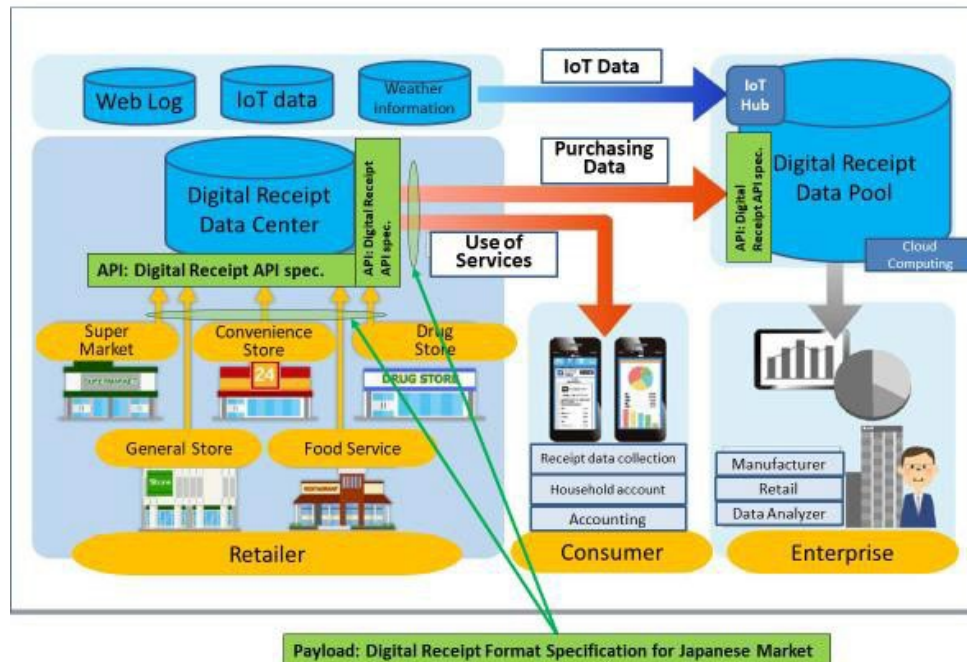For reference, the data flow of digital receipts in the Japanese market is shown below.

Figure 15-5: Digital Receipt Data Flow in the Japanese Market