



# Digital Transaction Register

## Acquirer Interface Agreement

Version: 5.0

Status: Draft

## Change History

The following table shows the history of changes to this document.

| Date       | Author          | Change history  |
|------------|-----------------|---|
| 28/02/2020 | Stefano Menotti | First version Draft   |
| 28/04/2020 | Luca Somaruga   | Timestamp field format update   |
| 21/05/2020 | Debora Arena    | <p>Updates:</p> <ul style="list-style-type: none"> <li>- Chapter “Digital payment transactions covered by the service”: added constraint on enrolled HashPans.</li> <li>- Paragraph “Integration with Acquirer”: added constraint on enrolled HashPans.</li> </ul> <p>Added:</p> <ul style="list-style-type: none"> <li>- Chapter “Batch service for HashPan control”</li> </ul> <p>Update:</p> <ul style="list-style-type: none"> <li>- “Appendix 1 - File transfer mode”: the batch service will send the flows to the Platform sFTP</li> </ul> |
| 22/05/2020 | Stefano Menotti | Version exportable to Acquirer stakeholders   |
| 27/06/2020 | Debora Arena    | <p>Updates:</p> <ul style="list-style-type: none"> <li>- Standard PagoPA flow</li> <li>- Addition of the chapter “Onboarding Merchant through Acquirer”</li> </ul>  |
| 30/07/2020 | Rodolfo Viti    | <p>Update:</p> <ul style="list-style-type: none"> <li>- Standard PagoPA flow</li> <li>- Appendix 4 - Salt recovery service</li> <li>- Appendix 5 - Service for downloading HPANs registered in CentroStella</li> <li>- Appendix 6 - Acquirer Services Authentication</li> </ul> <p>Added:</p> <ul style="list-style-type: none"> <li>- Appendix 7 - Acquirer Services Authorisation</li> <li>- Appendix 8 - Environments</li> </ul>   |

|            |              |   |
|------------|--------------|---|
| 02/09/2020 | Rodolfo Viti | Update:<br>- Standard PagoPA flow (field length)  |
| 02/10/2020 | Denisa Braho | Added:<br>- “Proposed functional solution” paragraph containing the process happy flow and the relative drawing<br>Update:<br>- “circuit_type” field (circuit “00-PagoBancomat” will be managed exclusively by the Bancomat Acquirer) |
| 14/12/2020 | Denisa Braho | Update:<br>- Happy flow<br>- “circuit_type” field (circuit “03-AMEX” will be managed exclusively by Amex)   |
| 15/12/2020 | Denisa Braho | Update:<br>- Happy flow<br>- “circuit_type” field (circuit Diners will be managed exclusively by Diners)  |
| 13/04/2021 | Denisa Braho | Updated:<br>- paragraph “Standard PagoPA flow” by adding a new field in the file (parameter: PAR)<br>- updated description of the parameter “hash_pan”  |

## Document Approval

The following table lists the stakeholders with whom the document was shared and approved.

| Stakeholder (name) | Approval date | Validated processes |
|--------------------|---------------|---------------------|
|                    |               |                     |
|                    |               |                     |

## Index

**Change History**

2

**Document Approval**

**Errore. Il segnalibro non è definito.**



|  |  |
|--|--|
| <b>Index</b>   | <b>Errore. Il segnalibro non è definito.</b> |
| <b>Introduction and purpose of the document</b>                              | <b>Errore. Il segnalibro non è definito.</b> |
| Regulatory reference   | 6  |
| Privacy and data processing  | 6  |
| Introduction and scope of the initiative                                     | 6  |
| Objective:   | 6  |
| Functional solution proposed: Digital Payments Bonus                         | 7  |
| Happy Flow   | 7  |
| <b>Digital payment transactions covered by the service</b>                   | 8  |
| Information Perimeter  | 8  |
| Process for sending transactions to RTD                                      | 8  |
| <b>Acquirer Integration with PagoPA CentroStella</b>                         | 10   |
| <b>Standard PagoPA flow</b>  | 11   |
| <b>Batch service for controlling enrolled HPANs</b>                          | 16   |
| <b>Merchant Onboarding via Acquirer and saving data on the FA Platform</b>   | 19   |
| Acquirer Registration Service on API Gateway                                 | 19   |
| Show T&C   | 20   |
| Retrieving Billing Provider List   | 21   |
| T&C acceptance, sending Merchant data and saving on the FA Platform          | 22   |
| <b>Appendix 1 - PGP Public Key</b>   | <b>Errore. Il segnalibro non è definito.</b> |
| <b>Appendix 2 - File Transfer Mode</b>                                       | 25   |
| <b>Appendix 3 - sFTP SIA access manual</b>                                   | 25   |
| <b>Appendix 4 - Salt recovery service</b>                                    | 25   |
| <b>Appendix 5 - Service for downloading HPANs registered in CentroStella</b> | 27   |
| <b>Appendix 6 - Acquirer Services Authentication</b>                         | 33   |
| <b>Appendix 7- Acquirer Services Authorisation</b>                           | 36   |
| <b>Appendix 8 - Environments</b>   | 39   |
| <b>Appendix 9 - PGP Production Public Key</b>                                | 40   |

## Introduction and purpose of the document

The purpose of this document is to describe the application solution, in all its interfaces and the different



flows of incoming or outgoing events to be managed and the related data exchange methods, as well as the High Level executive architecture, with particular reference to the interfaces presented by the Acquirer subjects to PagoPa SpA systems (CentroStella).

## Regulatory reference

The service has as its regulatory reference Italian Legislative Decree 26/10/19 no. 124 converted by conversion Law of 19 December 2019, no. 157 published in the Official Journal no. 301 of 24-12-2019, and in particular in Article 21:

*Art. 21: Tax certifications and electronic payments*

*1. In Article 5 of Italian Legislative Decree no. 82 of 7 March 2005, the following are added after ((paragraph 2))-quinquies: **“2-sexies. The technological platform referred to in paragraph 2 may also be used to facilitate and automate, through electronic payments, the tax certification processes between private parties, including electronic invoicing and** the storage and transmission of daily fee data referred to in Articles 1 and 2 of Italian Legislative Decree no. 127 of 5 August 2015.*

*2-septies. By decree of the President of the Council of Ministers or of the Minister Delegate for Technological Innovation and Digitisation, in agreement with the Minister of Economy and Finance, the technical operating rules of the technological platform and the processes referred to in paragraph 2-sexies are defined”.*

## Privacy and data processing

Please refer to the document DPIA approved by the Privacy Guarantor.

## Introduction and scope of the initiative

The objective of the project is the creation of a technological infrastructure which allows to enable new use cases and services for citizens and businesses, mainly focused on the digitalisation of payments through the use of cards and payment instruments through physical POSs.

The pillar of the new infrastructure is communications with the Acquirer entities operating in Italy.

The PagoPa platform CentroStella must manage information that must comply with all the requirements of the GDPR; in particular, it must not be allowed to trace the individual transaction and recover the personal data of the payers and/or the payment in any way.

The macro components covered by the initiative are listed below:

|  |  |
|--|--|
| <b>DIGITAL TRANSACTION LOG (RTD)</b><br>Aggregates commercial transactions carried out through digital payment instruments, both by individuals and by companies through physical POSs throughout Italy. A single log that enables the creation of electronic billing, welfare and automation incentive solutions. |  |
| <b>AUTOMATIC BILLING</b><br>Relies on the Digital Transaction Log for the automatic issuance of electronic invoices in the context of a payment made by a company.   | <b>DIGITAL PAYMENTS BONUS</b><br>Relies on the Digital Transaction Log to award bonuses to citizens who make payments through digital payment instruments. |

## Objective:

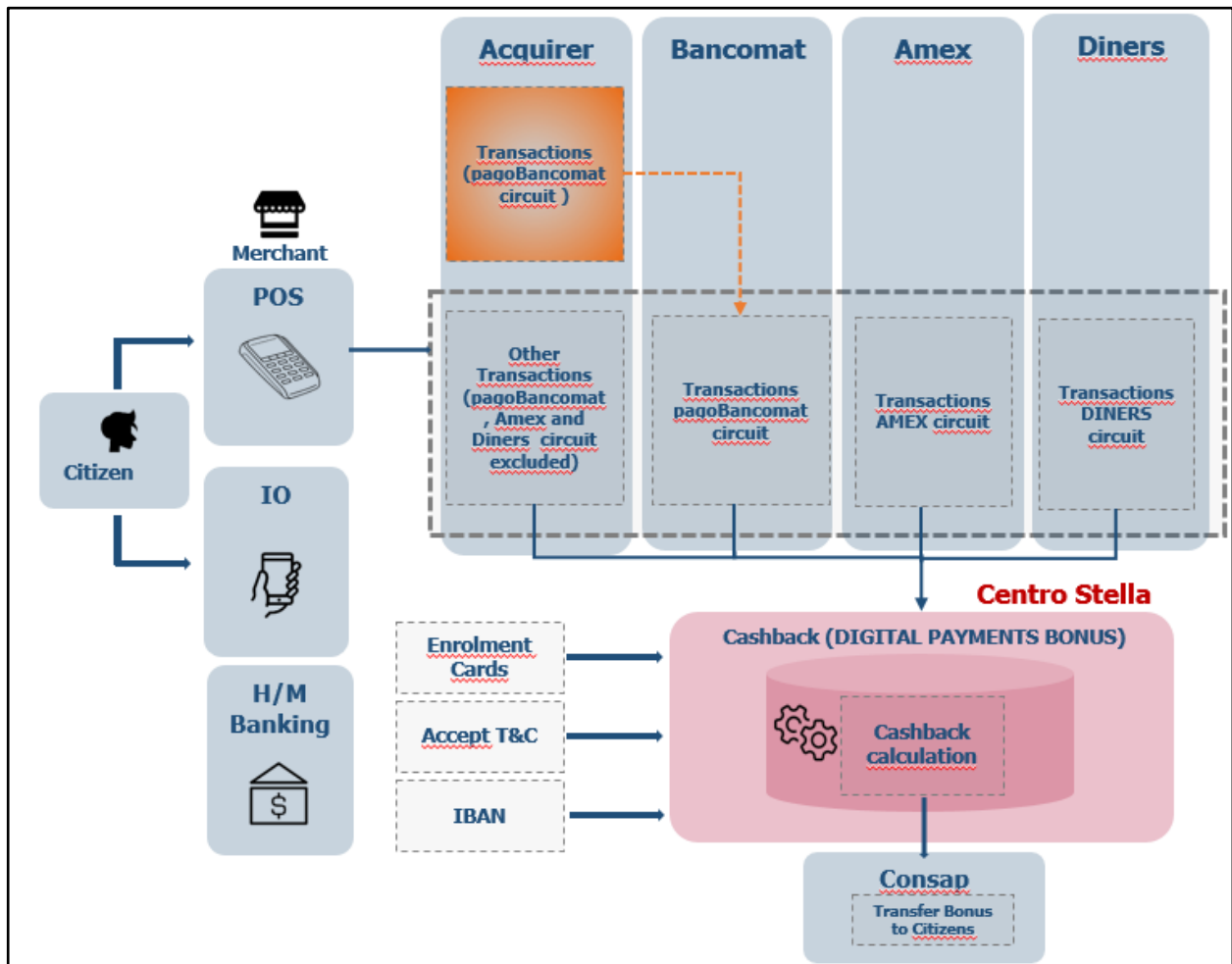
The Digital Transaction Log is therefore the enabling platform for different and future use cases that see

the role of electronic payments as central, maintaining a unique integration with Acquirer entities operating in Italy.

The main objectives of the project are therefore:

- to encourage payments by electronic means to reduce the use of cash, creating rewarding conditions and a cumulative result even if achieved with different payment instruments.
- to boost the adoption of e-billing by small operators by simplifying the exchange of information between all the actors involved in the process.

## Functional solution proposed: Digital Payments Bonus



## Happy Flow

The Happy Flow is explained below, summarising the proposed functional solution of the project:

- The Citizen securely stores their payment instrument on the platform CentroStella
- The Citizen makes a payment to a physical operator in Italy
- The Acquirer, after posting the transaction, sends the transaction data to the platform BPD
  - The Acquirers participating in the service will send to Centro Stella only the transactions related to circuits different from PagoBancomat, Amex and Diners.



- The Acquirers will send only the transactions carried out on the PagoBancomat circuit to Bancomat
- Bancomat, Amex and Diners will send directly to CentroStella all the transactions carried out on their own circuit ( including those received from the other Acquirers in case of Bancomat)
- d. CentroStella assigns points to the transaction
- e. Cashback awards are assigned based on the accumulated points
- f. The Citizen enters the IBAN of their current account through the IO App or H/M Banking to be credited with the accrued cashback.

## Digital payment transactions covered by the service

### Information Perimeter

The **digital electronic payments** to be considered are all payments made through *physical POSs in Italy*, using the following payment instruments:

- debit cards on international and ATM circuits,
- credit cards,
- prepaid cards (rechargeable cards not linked to a current account, rechargeable cards linked to a current account, rechargeable cards with account functions),
- applications related to credit transfers or other settlement systems

For a correct provision of services, and with the aim of not discriminating against any type of citizen/payment transaction, **the Acquirer must also convey the transactions including any on-us modes to PagoPA** and including any reversal operations.

The object of the service will be only transactions in *EUR currency*.

The following transactions are therefore excluded from the perimeter:

- related to cashback (cash withdrawal)
- cash advance on POS (MCC = 6010)
- cash advance on ATM (MCC = 6011)
- related to e-commerce
- Dynamic Currency Conversion (DCC)
- carried out in the territory of San Marino

It should also be noted that, according to the process approved by the Privacy Guarantor, only transactions with HashPans relating to payment instruments for which the Citizen/Buyer has enrolled in one of the services activated on the CentroStella platform of PagoPa will be able to flow to the PagoPA systems (CentroStella).

### Process for sending transactions to RTD

The process of sending transactions to the CentroStella Platform (RTD) consists of the following phases:

- a. The CentroStella Platform generates a flow containing the HPANs enrolled in the CentroStella service.
- b. The Acquirer consolidates the data relating to all transactions of interest accounted for in the last





settlement cycle with the merchant. It therefore generates a text file in csv format (with file naming and detailed layout in the paragraph "Standard PagoPA Flow") and deposits it in a folder on which the batch is being polled. The deposit of the file is the trigger that starts the batch process.

- c. The Batch installed at the Acquirers invokes the service displayed by the Platform, through which a one-shot link is generated and activated temporarily to download the flow of HPANs/HTokenPANs enrolled in the CentroStella platform services.
- d. The Batch installed at the Acquirers invokes the service displayed by the Platform, through which a one-shot link is generated and activated temporarily to download the flow of PARs for the instruments enrolled in the CentroStella platform services.
- e. The Batch calls the service presented by CentroStella to obtain the constant hashing key to add to the PAN to hash the PANs contained in the transaction flow.
- f. The Batch reads both input flows (HPAN list and transactions) and, for each line of the transaction file:
  - i. hashes the PAN in the transactions flow;
  - ii. determines whether the transaction row should be discarded or reported in the filtered output flow, matching the hashed PAN with the one obtained using the list OR if the PAR matches with the list obtained through the rest service.
- g. After the processing in the previous point, the Batch finishes writing the filtered flow in output, and deletes all the data temporary data received in input.
- h. The Batch performs PGP encryption of the output flow. "Appendix 2" shows the public key to be used for encrypting the file produced in the previous point.
- i. The Batch deposits the filtered transaction flow on CentroStella sFTP. For more information on instructions and how to access the SIA sFTP, refer to the indications in Appendix 3. It should also be noted that the Batch deposits the output file in the directory /Inbox/.

This section contains the phase to be used to obtain and send the file containing eventual TokenPANs related to enrolled instruments of which there was any indication in either the HPANs/HTokenPANs or PARs lists in the first part of the process. This phase has to be used only if the tokenized cards are in use in a system, and will be executed AFTER having produced the transactions file to send to CentroStella

- j. The discarded transaction, that contains a PAR, are used to produce another file (with the format described in the paragraph "Standard PagoPA flow – TokenPANs"), to be used as input in the next phase of the process
- k. The Batch installed at the Acquirers invokes the service displayed by the Platform, through which one-shot links is generated and activated temporarily to download the flow of Bin Range for payment instruments enrolled in CentroStella
- l. All the links obtained through the first call has to be explicitly called through GET requests, obtaining one or more compressed files containing a .csv format in which two columns are present, containing the PAN starting a Bin Range, and the second one with the PAN ending the range
- m. The Batch installed at the Acquirers invokes the service displayed by the Platform, through which one-shot links is generated and activated temporarily to download the flow of HTokenPANS for which the link to payment instruments enrolled in CentroStella is already known
- n. All the links obtained through the first call has to be explicitly called through GET requests, obtaining one or more compressed files containing a .csv format in which one column is present, containing the HTokenPANs (The TokenPAN will be hashed with the same salt used for the HPANs list)

- o. The Batch reads both input flows (Bin Range list and tokenPANS) and, for each line of the transaction file:
  - iii. Determines whether the transaction row should be discarded or reported in the filtered output flow. A token can be validated further if the circuitType of the transactions containing the TokenPAN is 02 (Mastercard), or if the PAN is contained inside one of the ranges read from the recovered files. A record has to be discarded only if neither of this conditions occur, and only after comparing the PAN with all the available ranges.
  - iv. Any record that has been validated with the Bin Range will be compared with the recovered HTokenPANs list
- p. After the processing in the previous point, the Batch finishes writing the filtered flow in output, and deletes all the data temporary data received in input.
- q. The Batch performs PGP encryption of the output flow. “Appendix 2” shows the public key to be used for encrypting the file produced in the previous point.
- r. The Batch deposits the filtered transaction flow on CentroStella sFTP. For more information on instructions and how to access the SIA sFTP, refer to the indications in Appendix 3. It should also be noted that the Batch deposits the output file in the directory /Inbox/.
- i. Finally, CentroStella deletes all the files received in input.

## Acquirer Integration with PagoPA CentroStella

Accredited *Acquirers* (i.e., those who have entered into an agreement with PagoPA S.p.A.), at the end of the successfully concluded payment transaction, will generate the daily flow of transactions.

The latter will be filtered in order to verify that the HPANs or PARs present in the various paths correspond to those enrolled in the CentroStella Platform, before being sent to the same.

In case the acquirer manages tokenized cards, and additional daily flow will be produced from the original transaction flow, listing any tokenPAN for which the associated PAT matches the Bin Range of the instruments enrolled in CentroStella.

This controls will be guaranteed by a batch service installed on the Acquirers' systems, which PagoPa will give to the Acquirer, in the form of an *opensource* code, to be used to facilitate integration and minimise efforts. For more details see the paragraph “Batch service for HPAN control”

Given the above, it should be noted that the Acquirers will have the right to use one or more integration modes and, for each mode, one or more daily files to cover the entire set of transactions to be transmitted to CentroStella.

PagoPA SpA will ask the accredited *Acquirers*, maintaining the objective of minimising the technological effort for each Acquirer and, at the same time, ensuring *compliance* with PCI<sup>1</sup> regulations, the sending of one or more **batch flows** in the “**PagoPA standard**” format: PagoPA SpA will provide a simple implementation flow specification that includes the minimum subset of data, described in the following paragraphs of this document.

PagoPA SpA (CentroStella Platform) is responsible for managing the data in the PCI environment according to current legislation and maintaining only the minimum subset of anonymised data (**filtered for enrolled HashPans**), necessary for the operation of the Platform, eliminating any data of transactions that have not been made with payment instruments voluntarily enabled by the Citizen on the Platform.



The card data will be saved with an **irreversible cryptographic hash function**.

In this respect, the integration of the Acquirers with CentroStella is divided into the following phases:

- Standard PagoPa flow – Transactions
- Standard PagoPa flow – TokenPANs
- Batch service for enrolled HPAN/PAR and Bin Range/HTokenPAN Control

The details of the points listed above are shown in the following paragraphs.

## Standard PagoPA flow - Transactions

The following describes the details of the Standard PagoPA Flow, regarding the transaction file to be used as an input to the batch process and as the expected output to be sent in CentroStella.

The naming convention of the file is as follows:

- [service].[ABI].[filetype].[date].[time].[nnn].csv

in particular:

- service: fixed as 'CSTAR' (5 alphanumeric digits)
- ABI: Sender ABI (5 numeric digits)
- filetype: fixed as file type (6 alphanumeric digits)
- nnn: progressive file (3 numeric digits)

---

<sup>1</sup> Payment Card Industry: security certification with which all payment systems must comply.

| field         | format                | notes                                    |
|---------------|-----------------------|--|
| service       | Alphanumeric - 5 char | fixed value CSTAR                        |
| ABI           | Alphanumeric - 5 char | sender ABI code                          |
| file_type     | Alphanumeric - 6 char | type of flow sent. Fixed value TRNLOG    |
| [date].[time] | YYYYMMDD.HHMISS       | file creation timestamp                  |
| nnn           | Alphanumeric - 3 char | Progressive value of the file (e.g. 001) |

Please note that:

- The file is in .csv format, with separators “;”
- the file is encrypted with a pgp public key issued by PagoPa SpA
- The contents of the file do not include head and tail records but only detail records,



according to this layout:

Fields in the Standard PagoPA Flow.

| field                 | Type                           | Mandatory | Notes   |
|-----------------------|--------------------------------|-----------|---|
| <b>acquirer_code</b>  | Alphanumeric - max 20 char     | YES       | ABI code of Acquirer bank.  |
| <b>operation_type</b> | Alphanumeric - regexp [0-9]{2} | YES       | <p>Operation type:</p> <p>00 - payment</p> <p>01 - reversal of payment</p> <p>02 - payment with ApplePay</p> <p>03 - payment with GooglePay</p> <p>xx - future uses</p> <p>Types 02 and 03 will not always be valued by the Acquirers</p>   |
| <b>circuit_type</b>   | Alphanumeric - regexp [0-9]{2} | YES       | <p>Payment circuit:</p> <p>00 – Pagobancomat</p> <ul style="list-style-type: none"> <li>- Transactions on this circuit will be sent exclusively by the Acquirer Bancomat.</li> </ul> <p>01- Visa</p> <p>02- Mastercard</p> <p>03- Amex</p> <ul style="list-style-type: none"> <li>- Transactions on this circuit will be sent exclusively by AMEX.</li> </ul> <p>04- JCB</p> <p>05- UnionPay</p> <p>06- Diners</p> <ul style="list-style-type: none"> <li>- Transactions on this circuit will be sent exclusively by Diners.</li> </ul> <p>07- PostePay Code</p> <p>08- BancomatPay</p> <p>09- SatisPay</p> <p>10- private circuit (onus, owen)</p> <p>xx - future uses</p> |

|                  |   |     |  |
|------------------|---|-----|--|
| <b>hash_pan</b>  | Alphanumeric –<br>max 64 char   | YES | <p>Hash of the PAN of the payment instrument used.</p> <p>In the case of a non-card based circuit, it represents the unique identifier of the proprietary payment instrument, which the user can register through the IO App or touch point of the Issuer bank.</p> <p>In case the payments are done with a tokenized payment instrument, this field must contain the hash Token that uniquely identifies the payment instrument (es. Apple Pay; Google Pay etc)</p> |
| <b>date_time</b> | DateFormat<br><i>ISO8601 FORMAT</i><br>yyyy-MM- ddTHH:mm:ss.SSSXXXX X | YES | <p>Timestamp of the payment transaction carried out with the Merchant.</p> <p>Please note that the second details are not always available for all transactions. In this circumstance, the detail will be padded with all '0's</p>   |

|                        |                           |     |  |
|------------------------|---------------------------|-----|--|
| <b>id_trx_acquirer</b> | Alphanumeric max 12 char  | YES | Unique identifier of the transaction at the Acquirer level.<br><br>- can be populated with the <i>ARN</i> , or if this data is not present, with a <i>unique id</i> that allows to uniquely identify the transaction on the Acquirer side. |
| <b>id_trx_issuer</b>   | Alphanumeric max 12 char  | NO  | Authorisation code issued by the Issuer (ex: AuthCode)   |
| <b>correlation_id</b>  | Alphanumeric max 255 char | NO  | Correlation identifier between payment transaction and possible reversal.<br><br>In certain cases, the data cannot be retrieved by the Acquirer and the information in the field in question will not be sent                              |
| <b>total_amount</b>    | Numeric                   | YES | Valued in euro cents (ex: 10€ = 1000) and expressed in absolute value: the sign is assumed from the type of operation " <i>00-payment, 01-reversal</i> "   |
| <b>currency</b>        | Alphanumeric - max 3 char | NO  | Fixed value 978 = EUR. International ISO coding is used.   |
| <b>acquirer_id</b>     | Alphanumeric max 255 char | YES | Unique Acquirer ID. In the case of card transactions, it represents the homonymous value conveyed on the international circuits.   |

|   |  |     |   |
|---|--|-----|---|
|   |  |     | <ul style="list-style-type: none"> <li>- In the Pagobancomat circuit it corresponds to the field <i>codice_sia_abi</i></li> <li>- Visa/Mastercard Circuit: <i>acquirer_id</i></li> </ul> <p>In other cases the field will be valued with fixed data depending on the reference Acquirer</p>     |
| <b>merchant_id</b>                      | Alphanumeric max 255 char                      | YES | <p>Unique identifier of the physical store of the Acquirer (also known to the Merchant and used by the same to register on the Automatic Billing platform).</p> <ul style="list-style-type: none"> <li>- In the Pagobancomat circuit it can correspond to the field: <i>merchant</i></li> </ul> |
| <b>terminal_id</b>                      | Alphanumeric max 255 char                      | YES | <p>Identification of the Merchant's terminal/POS (Point of Sale).</p> <ul style="list-style-type: none"> <li>- In the Pagobancomat circuit it corresponds to the field: <i>cashier premises</i></li> <li>- Visa/Mastercard circuit: <i>terminal_id</i></li> </ul>                               |
| <b>bank_identification_number (BIN)</b> | Alphanumeric or -<br>regexp [0-9]{6}[[0-9]]{8} | YES | <p>Code containing the first 8 digits of the payment instrument.</p> <ul style="list-style-type: none"> <li>- In the Pagobancomat circuit it corresponds to the field: <i>codice_abi</i></li> </ul>   |
| <b>MCC</b>                              | Alphanumeric max 5 char                        | YES | Merchant Category Code.   |
| <b>PAR</b>                              | Alphanumeric                                   | NO  | <p>Payment Account reference</p> <p>The field must contain the information of the PAR, which can be defined as a collector capable to associate each TokenPAN with the PAN of the physical card, thanks to the unique and immutable association between PAN and PAR and Token and PAR</p>       |

## Standard PagoPA flow – TokenPANs

The following describes the details of the Standard PagoPA Flow, regarding the tokenPAN file. This file will be directly produced inside the batch process, using the transactions inside the transactions flow used as an input to the process.

The naming convention of the file is as follows:

- [service].[ABI].[filetype].[date].[time].[nnn].csv

in particular:

- service: fixed as 'TKM' (3 alphanumeric digits)
- ABI: Sender ABI (5 numeric digits)
- filetype: fixed as file type (6 alphanumeric digits)
- nnn: progressive file (3 numeric digits)

<sup>1</sup> Payment Card Industry: security certification with which all payment systems must comply.

| field         | format                | notes                                    |
|---------------|-----------------------|--|
| service       | Alphanumeric - 5 char | fixed value CSTAR                        |
| ABI           | Alphanumeric - 5 char | sender ABI code                          |
| file_type     | Alphanumeric - 6 char | type of flow sent. Fixed value TKNLST    |
| [date].[time] | YYYYMMDD.HHMISS       | file creation timestamp                  |
| nnn           | Alphanumeric - 3 char | Progressive value of the file (e.g. 001) |

Please note that:

- The file is in .csv format, with separators “;”
- the file is encrypted with a pgp public key issued by PagoPa SpA
- The contents of the file do not include head and tail records but only detail records, according to this layout:

Fields in the Standard PagoPA Flow.



| field               | Type                              | Mandatory | Notes   |
|---------------------|-----------------------------------|-----------|---|
| <b>TokenPAN</b>     | Alphanumeric                      | YES       | PAN for a tokenized payment instrument (not hashed)   |
| <b>circuit_type</b> | Alphanumeric - regexp<br>[0-9]{2} | YES       | Payment circuit:<br>00 – Pagobancomat<br>- Transactions on this circuit will be sent exclusively by the Acquirer Bancomat.<br>01- Visa<br>02- Mastercard<br>03- Amex<br>- Transactions on this circuit will be sent exclusively by AMEX.<br>04- JCB<br>05- UnionPay<br>06- Diners<br>- Transactions on this circuit will be sent exclusively by Diners.<br>07- PostePay Code<br>08- BancomatPay<br>09- SatisPay<br>10- private circuit (onus, owen)<br>xx - future uses |
| <b>PAR</b>          | Alphanumeric                      | YES       | Value associating the Tokenized card with the physical card.<br><br>The first four digits are used to match the Bin Range of enrolled instruments for the CentroStella services   |

Batch service for controlling enrolled HPANs/PARs and



## Bin Range/HTokenPANs

The service will be developed by CentroStella and installed at the accredited Acquirers. However, it should be noted that the maintenance of the service, and any modifications thereof, are the responsibility of the Acquirer. PagoPa will provide the source code in opensource logic, published in public repositories to facilitate integration and minimise efforts.

### Operating Mode

The artefact consists of an executable jar produced with *spring-boot*, therefore all the project dependencies are contained within the jar along with the classes that contain its business logic.

In this way the artefact is completely autonomous and usable on any device that has a JVM.

The installation and execution of the batch requires:

- Java 1.8+
- *Batch-transaction-filter.jar* artefact

With regard to the parameters and execution commands, please refer to the indications in the README file in the public repository accessible via the link: <https://github.com/pagopa/rtd-ms-transaction-filter/blob/master/README.md>

### Minimum requirements

Below are the minimum requirements for the execution of the batch described above:

Software:

- JVM 1.8+

Hardware:

- CPU:
  - Architecture: x86\_64
  - CPU op-mode(s): 32-bit, 64-bit
  - CPU(s): 4
  - CPU MHz: 2992.966
- RAM: 6 GB
- HD: Depending on the size of the transaction file. To the previous file must be added the size of the file containing the pan hashes which is around 300 MB (in pgp format).

### Execution status management

The batch service manages the files used in cases where a blocking error occurs or when the execution is completed successfully.

The behaviour of the various steps of the service will be affected by the configuration of the property *deleteLocal*, which requires possible deletion at the end of the execution of all processed files, if active.

If otherwise configured, the appropriate behaviour is archiving the files processed in the flow, both for the file containing the list of PANs, and for that of transactions, for any management alongside the same.

In case of successful execution, the pan file will be removed, along with all temporary files generated before the final output file is sent. The transaction file obtained will be stored in a dedicated directory.

It will also be possible to configure, for generic errors in processing individual file records, a margin of tolerance with respect to the number of rows for which an error was found, using the property *skipLimit*.

The file will be processed without exceeding the configured threshold value, a success will be reported conditioned on the presence of some errors, which can possibly be managed on the sides.

## Merchant Onboarding via Acquirer and saving data on the FA Platform (in review)

In order to be able to Onboard the Merchant to Automatic Billing services, the Acquirer's systems will invoke the APIs displayed by the CentroStella platform to census the Merchant and will display a service to communicate the Merchant's data to the Platform.

Subsequently, the FA system will save this information in the internal database.

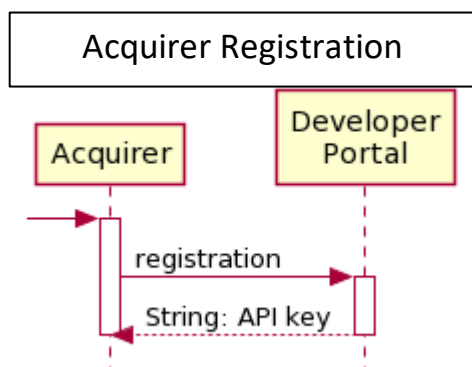
Having said the above, the following APIs are expected to be presented to Acquirers:

- Service subscription T&C display [showT&C]
- Acceptance of T&C service subscription [acceptT&C]
- Retrieving Billing provider list
- Sending Merchant data and saving on the FA Platform

The following paragraphs describe the services and parameters necessary for correct integration with the Automatic Billing service.

### Acquirer Registration Service on API Gateway

The "Acquirer Registration" service allows the Acquirer to register on the API Gateway portal and obtain the key used for authentication on CentroStella.



For more details, refer to [Appendix 7](#).



## Show T&C

**Path:** fa/tc/html

**Method:** GET

### Path Parameters

No parameters envisaged

### Query Parameters

No parameters envisaged

### Request Header

| Field        | Type   | Mandatory | Description   |
|--------------|--------|-----------|---|
| x-request-id | String | NO        | Request ID, unique identifier determined by initialisation (UUID) |

### Request Body

No parameters envisaged

### Response Code

HTTP Response Code 200

### Response Header

| Field        | Type   | Mandatory | Description   |
|--------------|--------|-----------|---|
| x-request-id | String | NO        | Request ID, unique identifier determined by caller or system (UUID) |

### Response Body

The service responds with HTML containing the Automatic Billing Terms and Conditions.

### HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code     | Description    |
|--------------------|----------------|----------------|
| 404                | FILE_NOT_FOUND | file not found |
| 500                | GENERIC_ERROR  | generic error  |



## Retrieving Billing Provider List

Through the service in question, it will be possible to retrieve the list of Providers integrated with the FA platform during the Onboarding phase of a Merchant through the Acquirer and select the Provider indicated by the Merchant, i.e., the Provider with which the Merchant signed the contract.

**Path:** /fa/provider/list

**Method:** GET

### Path Parameters

No parameters envisaged

### Query Parameters

No parameters envisaged

### Request Body

No parameters envisaged

### Response Code

HTTP Response Code 200

### Response Header

| Field        | Type   | Mandatory | Description   |
|--------------|--------|-----------|---|
| x-request-id | String | NO        | Request ID, unique identifier determined by caller or system (UUID) |

### Response Body

| field        | format       | Mandatory | Description   |
|--------------|--------------|-----------|---|
| providerList | Alphanumeric | YES       | List of providers participating in the service:<br>1. providerID<br>2. providerDesc |

### HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code      | Description   |
|--------------------|-----------------|---------------|
| 401                | TOKEN_NOT_VALID | invalid token |



|     |               |                               |
|-----|---------------|-------------------------------|
| 500 | GENERIC_ERROR | error retrieving user profile |
|-----|---------------|-------------------------------|

## T&C acceptance, sending Merchant data and saving on the FA Platform

**Path:** /fa/onboarding/merchant/{vatNumber}

**Method:** PUT

### Path Parameters

| Field     | Format       | Description                |
|-----------|--------------|----------------------------|
| vatNumber | Alphanumeric | VAT number of the Merchant |

### Query Parameters

No parameters envisaged

### Request Header

| Field         | Type   | Mandatory | Description   |
|---------------|--------|-----------|---|
| x-request-id  | String | NO        | Request ID, unique identifier determined by initialisation (UUID) |
| Authorisation | String | YES       | Bearer <token> (JWT format)                                       |

### Request Body

| Field       | Format       | Mandatory | Description  |
|-------------|--------------|-----------|--|
| timestampTC | Timestamp    | YES       | T&C timestamp acceptance. FORMAT ISO8601 yyyy-MM-ddTHH:mm:ss.SSSXXXXX                  |
| token       | Numeric      | YES       | authentication token between the portal and FA Platform                                |
| fiscalCode  | Alphanumeric | NO        | id of the natural person associated with the merchant that corresponds to the tax code |



|                           |              |     |   |
|---------------------------|--------------|-----|---|
| <b>acquirerMerchantId</b> | Alphanumeric | YES | Merchant ID separated from the Acquirer   |
| <b>providerID</b>         | Alphanumeric | YES | Unique ID of the Billing Provider where the Merchant has integrated with the platform |

## Response Code

HTTP Response Code 200

## Response Header

| Field               | Type   | Mandatory | Description   |
|---------------------|--------|-----------|---|
| <b>x-request-id</b> | String | NO        | Request ID, unique identifier determined by caller or system (UUID) |

## Response Body

| field            | format       | Mandatory | Description                |
|------------------|--------------|-----------|----------------------------|
| <b>vatNumber</b> | Alphanumeric | YES       | VAT number of the Merchant |

## HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code      | Description                    |
|--------------------|-----------------|--------------------------------|
| <b>400</b>         | INVALID_DATE    | incorrect date value or format |
| <b>401</b>         | TOKEN_NOT_VALID | invalid token                  |
| <b>500</b>         | GENERIC_ERROR   | error retrieving user profile  |
| <b>400</b>         | INVALID_PIVA    | Invalid VAT ID                 |

# Appendix 1 – Tokenized cards design review



The details regarding the integration of tokenized cards inside the process are available inside the design review document, available at:

<https://pagopa.atlassian.net/wiki/spaces/CEN/pages/71338906/Carre+Tokenizzate+-+Design+Review>

Note: The design review documentation is currently In Italian only

## Appendix 2 - PGP Public Key

For any problems related to the use of the public key and for the issuance of specifications and/or updates relating to the public key to be used to encrypt the file, you must contact the responsible office delegated by PagoPa (ref. SIA OPE Innovative Payments - sistemisti\_bigdata@sia.eu)

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: GnuPG v2.0.22 (GNU/Linux)

```
mQENBF6QNPABACAC3R3mV17UnvyiBIHssvXmYIhgS8dMDnqkwTNTw+7qt4cASzlwd
uaX4MvItwtYRt5oMMFKdAjVmDJrVZu0xpdoklet/LJX/3NhZTsJNnP/vckNc2QOt
NhfcJ5rsBoNTCUL25VJicM5KQeqCGIPF6gcSKVGkvTwjgRctIL85ua7syDM9pU6
3PhTz8mpN3PTnzNT0PPK3GxMg7N15BcHrNb7gA/SiNZpuBZ4BaEIIOCIhAhHE+5j
E1v8mWQiiRXohJUH3+R7nkU96rKbxk8/pN5Ey/SS2r/jb+xoJvh/knCSHNndY72q
DdnEj6/hqXwk4axx3RmhiNi3ywY1tpMKHSFtABEBAAAG0HnJ0ZGJhdGNoVEkgPHJ0
ZGJhdGNoVElAc2lhLmV1PokBPwQTAQIAKQIbAwcLCQgHAwIBBhUIAgkKCwQWAgMB
Ah4BAheABQJexNWjBQkNkwezAAoJEOYoxTAgG4FpxZ4H/AkE2IzuIHE8pnVpP3p2
JtmE78k/O8VC33jfoE9sDyIuYuFEi8CZqAp1BA+B8i0dv6/ccP1SfXs79QdyFyfU
JtjcrXgwbVmiiIkHkt38/5oSiSzlc/OOEY AuRvEthZFeXfDHS+/UIJ2BuTpmwNf
+pG4gAEjTRnzve3+TimUZV1MeNwMl21Jzk7romiHHGs6zMA97NcFxb/gbDk3AF/H
uplUoSgUWlwiYx3D3TyAfNwMzBSe8fj/gWRlXpGYfG+Ckgul02u6N3ZL/ntFvUMGP
d/ydHLAJR4SHSpMabJtyVrEmoRbIaPDINWykeMDx1VDW7sLFuFo8aLKzRwoEPahW
/T+JATkEEwECACMFAl6QNPACGwMHCwkIBwMCAQYVVCALJCsEFgIDAQIeAQIXgAAK
CRDmKMUwIBuBaS4vB/41MttoQoMNxKlqC78Qq9flpMZNohNdyO1P4rPex5mnMsMi
wQQ6hIFZTjdPismMZOX/bxWwF61Jd12tbkUBPnblsCMpasWIEy0RMCnIwJJonFqn
VflziHXCSHGfN25Sdel1tFEZ8iG1yP7eMG7relNfoH3RF9wIySUu6h45Bj4jdc
mfERQikWa4oOmoCVO350yF0fLY8Dt+jLRv9IBnFoyYWuCEIZ0knMI5yBbs+MI7
yGo5bl4xD+LcYSklmhQ7C42Bh/1VDWXAAGX7EC1s0QT2wDuEIG+tdi+odMe/DWP5
i31SvPCTWZ7y3wQMChsS1PTPcwithzCLGIkkoee5vuQENBF7E3H4BCADEwPaEMNsJ
28jQKJvxeqautkXtjaSx8UJDWgZP+mUTQe/DAohqFXcnOUI5l+E+KfC4DMpOY3g
waPMrw6tUBB7Ee5V4Ym5yALiQxK+fzi+ImHn9dqsng48LLx6Q1S9I8xsui+yxZo0
ifG36coQOYI2ATp9DPwTOdBRm8NCgJzc1VXMUqUxmmJ9Zl7sevUvFeLVURXnMIwe
UbFsGwJH3XX2vM3qJBMPKq0QqxZg7AsnVftxgStgaVZRbNg0A2lItZHpcZu12tz8
xMZyJ1z3GJHnWGM+sbZy/o19psTfthJLVzqtLYU5X82+YLn9WTGJ4VYPsOX7BQl
iMLQTVwA/AthABEBAAAGJASUEGAECFAA8FAI7E3H4CGwwFCQ1eZwAACgkQ5ijFMCAB
gWmWJQf+MjXBwb8GSwp/ILglGF1XqKTL057Z/VjmuPpOJ3Y/bIB/wgXgt4KXlSbM
YIiHrhJSHK64+DPA6OZD0ZQPwGOLk+VDfW6T2iEDtbOS1QHBBHkwyysNr9jn9mmo8
yM+xEguUoYcCnn+NdkH+zvJgDHUORNZ0OwOIOWR5yeLRePTLMgG673Cp+MoWePaY
FWM+hcdZDKwvU9Hzb5Laq7rXNGhdehPcZTHX+SvhjdOuVoKX/PbLa/9Hm+9F0vE
kVT7HK68ya8KZOJ3lmWzdsD9wVeQWRcYijTT7CeeGBqil3JN4+2jbw0/PLalQBew
v5HOUCTpJORE/SpdV6BcBy1dgtNtQ==
```

=b61E

-----END PGP PUBLIC KEY BLOCK-----



## Appendix 3 - File Transfer Mode

The CentroStella Platform provides an sFTP server on a public network, where the batch service installed on the systems of each Acquirer can deposit the files subject to the service in a specific folder.

The details of the Secure File Gateway (SFG) public Internet IPs are as follows:

| Env  | IP             | Port  | Protocol | User       | Auth Type                      | Upload Dir | Details        |
|------|----------------|-------|----------|------------|--------------------------------|------------|----------------|
| UAT  | 193.203.229.79 | 20022 | SFTP     | “ABI user” | Key Auth (RSA – min. 2048 bit) | /Inbox/    | SFG – Internet |
| PROD | 185.91.56.144  | 8022  | SFTP     | “ABI user” | Key Auth (RSA – min. 2048 bit) | /Inbox/    | SFG – Internet |

Each subject accesses with authentication modes to be defined, through unique keys. For configuration and any problems related to sFTP access, please refer to the following contact person delegated by PagoPa:

➤ **MFT Specialist** Mauro Cauli OPE  
 SIA S.P.A.  
 Managed File Transfer  
 Via Gonin, 36 - 20147 Milan, Italy P. +39 02.6084.4301  
 M. +39 335.13.30.882

Alternatively, if the Acquirer already has active transmission channels that guarantee the same security standards with PagoPA's technological partner, these channels may be used, subject to the development of the batch service in question.

## Appendix 4 - sFTP SIA access manual

[FTP access to SIA Spa systems on the Internet – v.1.0.pdf](#)

## Appendix 5 - Salt recovery service

CentroStella PagoPA (internal Payment Manager component) provides a REST service



for the recovery of the SALT to be chained to the original pan before hashing.

For details on Authentication and Authorisation, refer to [Appendix 12](#) and [Appendix 13](#).

Below is the API detail:

**Path:** /rtd/payment-instrument-manager/salt

**Method:** GET

#### Path Parameters

No parameters envisaged

#### Query Parameters

No parameters envisaged

#### Request Header

| Field                     | Type         | Mandatory | Description                                 |
|---------------------------|--------------|-----------|---|
| Ocp-Apim-Subscription-Key | Alphanumeric | YES       | Subscription key associated with the issuer |

#### Request Body

No parameters envisaged

#### Response Code

HTTP Response Code 200

#### Response Header

No parameters envisaged

#### Response Body

The service responds with the salt to be used during hashing.

#### HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code    | Description   |
|--------------------|---------------|---------------|
| 500                | GENERIC_ERROR | generic error |



## Appendix 6 - Service for downloading HPANs registered in CentroStella

After appropriate verification of the presence of the file, which is generated daily by a batch process, the HPAN download service provides the possibility of downloading the file containing the pan hash. On the first call, the service redirects (http 302) to the download url.

The downloaded file will be in csv format (the estimated size for a file containing 10 million HPANs is about 300 MB).

The file is produced daily and is available from 2:00 am. Specifically, the file of day T contains all the payment instruments participating in CentroStella services registered by 23:59:59 on T-1.

For details on Authentication and Authorisation, refer to [Appendix 12](#) and [Appendix 13](#).

Below is the API detail:

**Path:** /rtd/payment-instrument-manager/hashed-pans

**Method:** GET

### Path Parameters

No parameters envisaged

### Query Parameters

No parameters envisaged

### Request Header

| Field                     | Type         | Mandatory | Description                                 |
|---------------------------|--------------|-----------|---|
| Ocp-Apim-Subscription-Key | Alphanumeric | YES       | Subscription key associated with the issuer |

### Request Body

No parameters envisaged

### Response Code

HTTP Response Code 302 (FOUND).

### Response Header

| Field | Type | Mandatory | Description |
|-------|------|-----------|-------------|
|-------|------|-----------|-------------|

|                         |        |    |   |
|-------------------------|--------|----|---|
| <b>x-request-id</b>     | String | NO | Request ID, unique identifier determined by caller or system (UUID)       |
| <b>last-modified</b>    | String | NO | Timestamp indicating the last datetime in which the file has been changed |
| <b>x-ms-meta-sha256</b> | String | NO | SHA-256 checksum of the compressed file. To be used for validation        |

## Response Body

The service responds with a redirect to the link to download the csv file containing the hash of the PANs registered for the BPD and FA program.

## HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code           | Description          |
|--------------------|----------------------|----------------------|
| <b>404</b>         | FILE_NOT_FOUND       | file not found       |
| <b>500</b>         | GENERIC_ERROR        | generic error        |
| <b>403</b>         | AUTHENTICATION_ERROR | Authentication error |
| <b>401</b>         | AUTHORIZATION_ERROR  | Authorization error  |

# Appendix 7 - Service for downloading PAR of payment instruments registered in CentroStella

After appropriate verification of the presence of the file, which is generated daily by a batch process, the PAR download service provides the possibility of downloading the file containing the PARs of payment instruments enrolled with tokenized cards. On the first call, the service redirects (http 302) to the download url. The downloaded file will be in csv format, and will be received in a compressed .zip format.

The file is produced daily and is available from 2:00 am. Specifically, the file of day T contains all the payment instruments participating in CentroStella services registered by 23:59:59 on T-1.

For details on Authentication and Authorisation, refer to [Appendix 12](#) and [Appendix 13](#).

Below is the API detail:

**Path:** /rtd/payment-instrument-manager/par-list



**Method:** GET

### Path Parameters

No parameters envisaged

### Query Parameters

No parameters envisaged

### Request Header

| Field                     | Type         | Mandatory | Description                                 |
|---------------------------|--------------|-----------|---|
| Ocp-Apim-Subscription-Key | Alphanumeric | YES       | Subscription key associated with the issuer |

### Request Body

No parameters envisaged

### Response Code

HTTP Response Code 302 (FOUND).

### Response Header

| Field            | Type   | Mandatory | Description   |
|------------------|--------|-----------|---|
| x-request-id     | String | NO        | Request ID, unique identifier determined by caller or system (UUID)       |
| last-modified    | String | NO        | Timestamp indicating the last datetime in which the file has been changed |
| x-ms-meta-sha256 | String | NO        | SHA-256 checksum of the compressed file. To be used for validation        |

### Response Body

The service responds with a redirect to the link to download the csv file containing the PAR of the PANs/TokenPANs registered for the BPD and FA program, having an associated PAR.

### HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code     | Description    |
|--------------------|----------------|----------------|
| 404                | FILE_NOT_FOUND | file not found |



|     |                      |                      |
|-----|----------------------|----------------------|
| 500 | GENERIC_ERROR        | generic error        |
| 403 | AUTHENTICATION_ERROR | Authentication error |
| 401 | AUTHORIZATION_ERROR  | Authorization error  |

## Appendix 8 - Service for downloading Bin Range for valid TokenPANs to be used in CentroStella

After appropriate verification of the presence of the file, which is generated daily by a batch process, the Bin Range download service provides the possibility of downloading a list of files containing the Bin Ranges for valid PANs to be used in the context of payment instruments enrolled with tokenized cards. The service has a response containing a JSON body with a list of links to be called with separate GET requests to retrieve the file retrievable using the links. The downloaded file will be in csv format, and will be received in a compressed .zip format.

The file is produced daily and is available from 2:00 am. Specifically, the file of day T contains all the payment instruments participating in CentroStella services registered by 23:59:59 on T-1.

For details on Authentication and Authorisation, refer to [Appendix 12](#) and [Appendix 13](#).

For details on the services swagger refer to [Appendix 16](#).

Below is the API detail:

**Path:** /tkm/acquirerm/binrange/link (Note: in UAT the base URI is /tkm/uat/acquirerm....)

**Method:** GET

### Path Parameters

No parameters envisaged

### Query Parameters

No parameters envisaged

### Request Header

| Field | Type | Mandatory | Description |
|-------|------|-----------|-------------|
|-------|------|-----------|-------------|



|                                  |              |     |   |
|----------------------------------|--------------|-----|---|
| <b>Ocp-Apim-Subscription-Key</b> | Alphanumeric | YES | Subscription key associated with the issuer |
|----------------------------------|--------------|-----|---|

### Request Body

No parameters envisaged

### Response Body

| field          | format             | Mandatory | Description   |
|----------------|--------------------|-----------|---|
| fileLinks      | List<Alphanumeric> | YES       | List of temporary links to download the bin range partial files                             |
| numberOfFiles  | Numeric            | YES       | Number of partial files to download   |
| availableUntil | Timestamp          | YES       | Timestamp to indicate the ending date of validity for the temporary links                   |
| generationDate | Timestamp          | YES       | Datetime in which the files have been created. Used for validation inside the batch process |

### Response Code

200 (OK).

### Response Header

| Field             | Type   | Mandatory | Description   |
|-------------------|--------|-----------|---|
| <b>Request-Id</b> | String | NO        | Request ID, unique identifier determined by caller or system (UUID) |

### HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code           | Description          |
|--------------------|----------------------|----------------------|
| <b>404</b>         | FILE_NOT_FOUND       | file not found       |
| <b>500</b>         | GENERIC_ERROR        | generic error        |
| <b>403</b>         | AUTHENTICATION_ERROR | Authentication error |



|     |                     |                     |
|-----|---------------------|---------------------|
| 401 | AUTHORIZATION_ERROR | Authorization error |
|-----|---------------------|---------------------|

## Appendix 9 - Service for downloading HTokenPAN for valid TokenPANs to be used in CentroStella

After appropriate verification of the presence of the file, which is generated daily by a batch process, the HTokenPANs download service provides the possibility of downloading a list of files containing the Bin Ranges for valid PANs to be used in the context of payment instruments enrolled with tokenized cards. The service has a response containing a JSON body with a list of links to be called with separate GET requests to retrieve the file retrievable using the links. The downloaded file will be in csv format, and will be received in a compressed .zip format.

The file is produced daily and is available from 2:00 am. Specifically, the file of day T contains all the payment instruments participating in CentroStella services registered by 23:59:59 on T-1.

For details on Authentication and Authorisation, refer to [Appendix 12](#) and [Appendix 13](#).

For details on the services swagger refer to [Appendix 16](#)

Below is the API detail:

**Path:** /tkm/acquirerm/htoken/known/link (Note: in UAT the base URI is /tkm/uat/acquirerm....)

**Method:** GET

### Path Parameters

No parameters envisaged

### Query Parameters

No parameters envisaged

### Request Header

| Field                     | Type         | Mandatory | Description                                 |
|---------------------------|--------------|-----------|---|
| Ocp-Apim-Subscription-Key | Alphanumeric | YES       | Subscription key associated with the issuer |

### Request Body

No parameters envisaged





## Response Body

| field          | format             | Mandatory | Description   |
|----------------|--------------------|-----------|---|
| fileLinks      | List<Alphanumeric> | YES       | List of temporary links to download the bin range partial files                             |
| numberOfFiles  | Numeric            | YES       | Number of partial files to download   |
| availableUntil | Timestamp          | YES       | Timestamp to indicate the ending date of validity for the temporary links                   |
| generationDate | Timestamp          | YES       | Datetime in which the files have been created. Used for validation inside the batch process |

## Response Code

200 (OK).

## Response Header

| Field      | Type   | Mandatory | Description   |
|------------|--------|-----------|---|
| Request-Id | String | NO        | Request ID, unique identifier determined by caller or system (UUID) |

## HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code           | Description          |
|--------------------|----------------------|----------------------|
| 404                | FILE_NOT_FOUND       | file not found       |
| 500                | GENERIC_ERROR        | generic error        |
| 403                | AUTHENTICATION_ERROR | Authentication error |
| 401                | AUTHORIZATION_ERROR  | Authorization error  |

# Appendix 11 - Service for downloading Bin



## Range/HTokenPANs partial file (Temporary link)

The files to be obtained through the link in response for either Appendix 9 or Appendix 10 will be downloaded through links using this structure. In this case the TKM will not provide a 302 response with the initial call, therefore all the links within the list have to be directly called with explicit GET requests

The file is produced daily and is available from 2:00 am. Specifically, the file of day T contains all the payment instruments participating in CentroStella services registered by 23:59:59 on T-1.

For details on Authentication and Authorisation, refer to [Appendix 12](#) and [Appendix 13](#).

For details on the services swagger refer to [Appendix 16](#)

Below is the API detail:

**Path:** /tkm/acquirerm/download/file/{id}/{value} (Note: in UAT the base URI is /tkm/uat/acquirerm....)

**Method:** GET

### Path Parameters

No parameters envisaged

### Query Parameters

No parameters envisaged

### Request Header

| Field                     | Type         | Mandatory | Description                                 |
|---------------------------|--------------|-----------|---|
| Ocp-Apim-Subscription-Key | Alphanumeric | YES       | Subscription key associated with the issuer |

### Request Body

No parameters envisaged

### Response Body

Compressed file containing the request .csv file. Depending on the original method invoked it contains either a list of Bin Ranges, or a list of HTokenPANs

### Response Code

200 (OK).

### Response Header

| Field           | Type   | Mandatory | Description   |
|-----------------|--------|-----------|---|
| Request-Id      | String | NO        | Request ID, unique identifier determined by caller or system (UUID) |
| Checksum-Sha256 | String | NO        | SHA-256 checksum of the compressed file. To be used for validation  |

## HTTP Error Codes

Below is the list of error messages and the associated response codes

| HTTP Response Code | Error code           | Description          |
|--------------------|----------------------|----------------------|
| 404                | FILE_NOT_FOUND       | file not found       |
| 500                | GENERIC_ERROR        | generic error        |
| 403                | AUTHENTICATION_ERROR | Authentication error |
| 401                | AUTHORIZATION_ERROR  | Authorization error  |

## Appendix 12 - Acquirer Services Authentication

The interactions for Acquirer batch services use a mutual authentication mechanism on TLS 1.2 protocol, through the exchange of public certificates, issued by a CA (certifying authority), used for verification by both actors with respect to the keys in their possession. For this mechanism to be applicable, the following is therefore necessary:

- the Client must be configured to send requests over TLS 1.2 protocol, indicating a store containing the chain of certificates necessary to verify the reliability of the server on which the request is made; in addition, a store containing at least the private and public key with which the client authenticates with the machine contacted.
- the API must be configured to accept requests over TLS 1.2 protocol, it must be configured to use a collection of keys on which to apply certificate verification, it must be configured to provide a public certificate, used by the Client for the authentication of the machine to which the request is directed.

To generate the Certificate Signed Request it is necessary to use the [client-certificate.cnf](#) configuration template (suitably reconfigured with the information of the specific Acquirer). The command to invoke for generating the csr and its private key (using OpenSSL) is as follows:



```
openssl req -new -config client-certificate.cnf -keyout client-certificate.key -out client-certificate.csr
```

To enable the authentication process, certificates related to CAs in ".cer" format must be provided to the API publisher (since they must contain only the public key, the password is not mandatory, otherwise it must also be provided).

Client certificates must be provided to the Publisher API in ".pfx" format (containing only the public key), together with the relative password. The command to invoke for generating pfx from the client certificate (using OpenSSL) is as follows:

```
openssl pkcs12 -export -in client-certificate-signed.pem -nokeys -out public-cert.pfx
```

**N.B:** for tests in the SIT environment, the client certificate can be self-signed, and must be provided to the API publisher in ".pfx" format, while for higher environments it must be signed by the PagoPA internal CA, and it is not necessary to share it with the API Publisher. Consequently, the file containing the CA's public key should only be provided by the Acquirers in the SIT environment. In higher environments the PagoPA CA certificate will already be preconfigured. If it is necessary to obtain a certificate with a signature valid for environments above SIT, send the .csr to be signed to [security@pagopa.it](mailto:security@pagopa.it).

The APIs will be presented and configured to enable the mutual authentication process based on a given certificate. In the case of services used by Acquirers, a dedicated policy is introduced to allow the authentication process through multiple certificates, to allow the use of certificates for the Acquirers.

## Appendix 13 - Acquirer Services Authorization

Issuer system developers who need to use the published APIs must include a valid subscription key in HTTP requests when making calls to those APIs. Otherwise, the calls are immediately rejected by the API Management gateway and, as a result, are not forwarded to back-end services.

To obtain a subscription key for API access, a subscription is required. A subscription is essentially a container for a pair of subscription keys. Developers who need to use published APIs can obtain the subscriptions in two ways (depending on how they were configured):

- with the approval of API publishers;
- without the need for API publisher approval.

API publishers can also create subscriptions directly for API consumers.

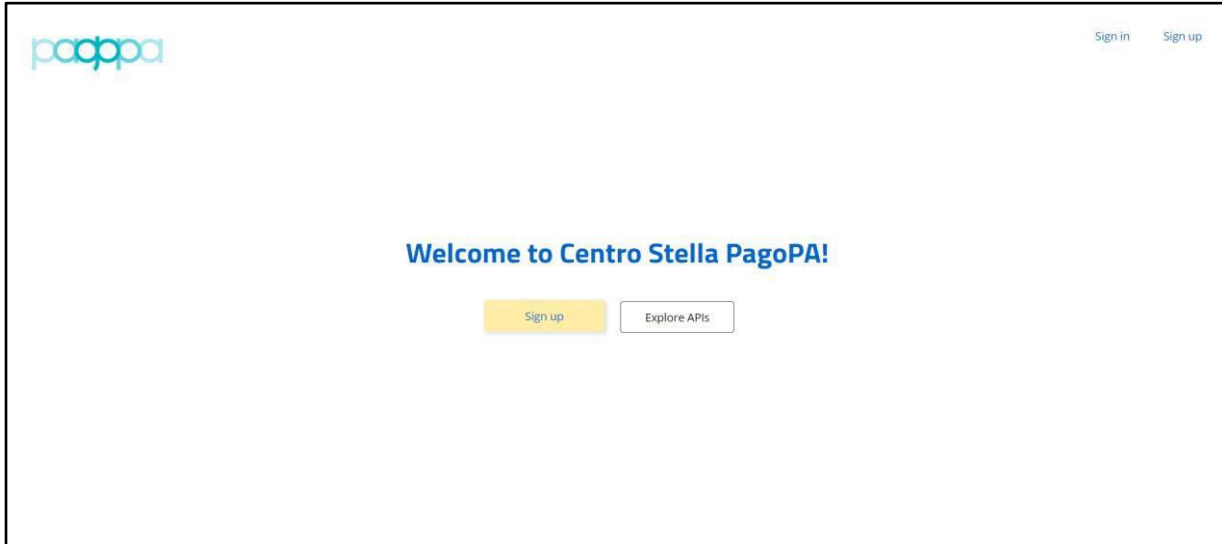


After subscribing, the client can invoke the services (for which it has subscribed) by entering the field **Ocp-Apim-Subscription-Key** as the parameter of the request header. The value of the field must match the code



obtained after registering with the developer portal Azure.

Below are the steps necessary to register to test the behaviour of the services;  
Access the dev address dedicated to developers (see appendix 8)

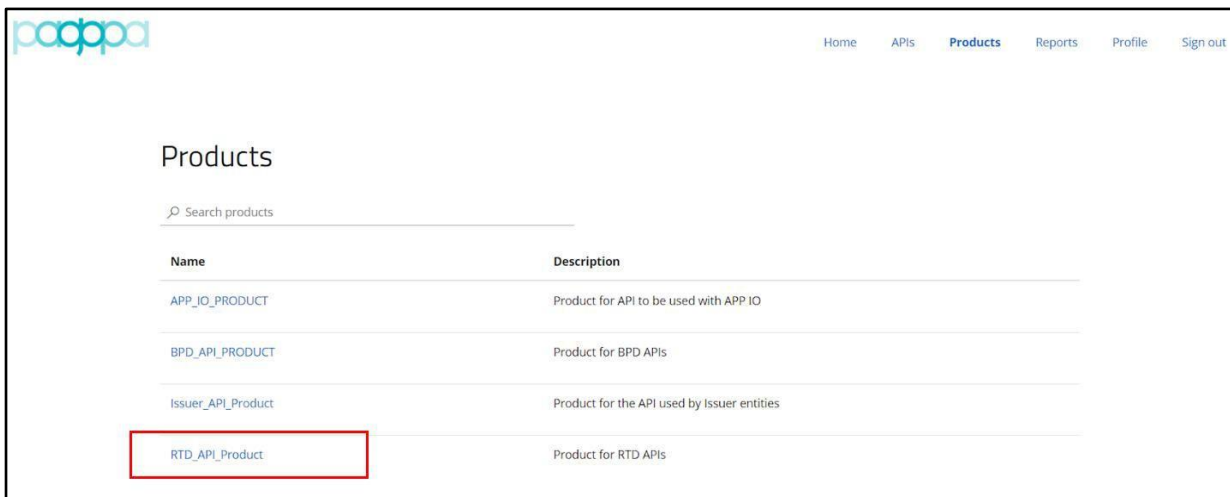


1. After clicking on the yellow button, you will be directed to the registration page where the credentials for the account configuration must be entered

2. After completing the credential entry process, we will receive via email the necessary configurations to complete the verification via a link.
3. After clicking on the link contained in the email, you will be redirected to the login page where you will have to authenticate with the created user. To create the subscription and its keys you must select the "Products" option.



4. At this stage, you must select the subscription type RTD\_API\_PRODUCT to access the services displayed for the Acquirers



5. Enter a name and select the Subscribe option.

Starter

Starter

Subscribers will be able to run 5 calls/minute up to a maximum of 100 calls/week.

Your subscriptions

| Name    | Status               |
|---------|----------------------|
| starter | <div>Subscribe</div> |

APIs in the product

Search APIs

| Name      | Description |
|-----------|-------------|
| _Echo API |             |

6. The outcome of the subscription will be visible under “Profile” in the menu.

Home

APIs

Products

Reports

Profile

Sign out

User profile

Account details

Email

First name

Last name

Registration date

Change name

Change password

Close account

Subscriptions

| Subscription details  | Product                              | State                                | Action  |
|---|--------------------------------------|--------------------------------------|---|
| <div>Name</div> <div>Started on</div> <div>04/15/2020</div> <div>Primary key</div> <div>Secondary key</div> | <div>Starter</div> <div>Rename</div> | <div>Starter</div> <div>Active</div> | <div>Cancel</div> <div>Show   Regenerate</div> <div>Show   Regenerate</div> |

## Appendix 14 - Environments

| Environment | IP address | API URL Gateway | Developer Portal URL |
|-------------|------------|-----------------|----------------------|
|-------------|------------|-----------------|----------------------|

|      |               |   |   |
|------|---------------|---|---|
| SIT  | 104.40.204.96 | <a href="https://bpd-dev.azure-api.net">https://bpd-dev.azure-api.net</a> | <a href="https://bpd-dev.developer.azure-api.net">https://bpd-dev.developer.azure-api.net</a> |
| UAT  | 20.54.178.216 | <a href="https://test.cstar.pagopa.it">https://test.cstar.pagopa.it</a>   | <a href="https://developer-test.cstar.pagopa.it">https://developer-test.cstar.pagopa.it</a>   |
| PROD | 51.137.18.218 | <a href="https://prod.cstar.pagopa.it">https://prod.cstar.pagopa.it</a>   | <a href="https://developer.cstar.pagopa.it">https://developer.cstar.pagopa.it</a>             |

## Appendix 15 - PGP Production Public Key

For any problems related to the use of the public key and for the issuance of specifications and/or updates relating to the public key to be used to encrypt the file, you must contact the responsible office delegated by PagoPa (ref. SIA OPE Innovative Payments - [sistemisti\\_bigdata@sia.eu](mailto:sistemisti_bigdata@sia.eu))

-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: GnuPG v2.0.22 (GNU/Linux)

```
mQENBF+IZlWBCADQ4KTeUitBNzQScZMdJEP7GLmqvU0GCN7rrWG6qdK7OvzQt/q
b2ImtqvildGii59GfsgyM0Rlpr5mkvPOv7no1KY5xW7r0c0ylBhKtX+xeFVbL
Ozu3YhqjSycinunTH8zblSSTIE6p/+SzCyV231XwP0yjHvUskCe1dnDM+ceJxQc
jQ46sBQ4vXJtA3TkaTwyBMrf0TW84mecdDtT2b4zt0FcQtnpG0B8ZanFO+cqH
d9Z6qLhTcHQM8C8ISKijKagPrWJLPfGuHzRh0YDIXSJp2alqirfuHVDctN1dtqcV
JhP39qf6WokJULEV8jFWRR2QPkLzAw5gcQ/ABEBAAAG0HnJ0ZGJjaHBwMDEgPHU0
ZGJjaHBwMDFAc2lhLmVlP0okBOQQTAAQIAIwUCX4hmXAIbAwcLCQgHawIBBhUIAgkK
CwQWAgMBAAAh4BAheAAAOJEFpOgVpChNKWddIIAJfx2Oh5zym2PLk8M40DGvcaVvyD
ZLsatwCCXknmD9xxOk7f3084nSvYfdXS8+UXFZvO4FZolVzPMpZC/ODIDWOF4MC
4F7wMSGaia5Wf6Tp+KOGInDzaWzBlu9K5YspHG/ee5h0ioMBTcwq1Q0cK1Bbn35j
t8uBX1iD47Ea/F0F3A1ME5OkT+JC8t3RYN4DuTwEnhUL6TeVoVKziOWrvO1HyQKD
4BCeB/aw1lihl8e3CLc0QNQqrH4CmiJDxjDv/Ozf9IdRqIXg34rtpdrgZnzC8Uo
ubyzq8dwkbl1ohErKpcxBSPHEh2bVER5zbleu5gv6lbyzbglj6ugUr9V9G5AQOE
X4hm3gEIALoyR2/IQfDSVvkiP+HfVHdv2Vp1z98C+BhflPuQ2kknBUXZaYMr5IB/
NYycls+QFbxHtkA50qOAGzY/gjLv/+Yr9J0Wu5xkZ1HCNwlpNMQwKwT8nh1vYo
EslweSZVRhDeCgwHDDwdo/Xa/xIzdAqMyxoS8yJVCwN1IiyvNOgbwQUXFHe0BR
AIBKKwibQ8iDNi+9NU2gzZyiLh6WsAAQLVolF3cS9sjhWcl+OB0Nsffj+oqkIb2
/wgpDwtUHcltCKGCB5HfueXCYNABs+O/TO4W5w42Up952s5tm9Mvaz8xbZW2ipi4
lyjeaPwm39aV/qTqkhwwk3YpQH4BCUAEQEAAYkCPgQYAQIACQUX4hm3glbAgEp
CRBAToFaQoTSlSbdlAQZABgUCX4hm3gAKCRAe9FqPx6cAANZAB/0eU9DXf6+s
Q6FeaXhDjC1BP2qLtrED1hmlJ8lrbAtk4AGyeTDhI5f1XiOuEqy9ATR9LOu4ap
97/3rItSx6J49gJeTCUyYhBAEclRXHoU44PnyxYj1xn/kQqSd6iq/xtUDqSjrpax
q42x/ISUmBhNZTa1JFPNWE4aedph9eiWIN/gT8/9m0cA/ZXfZuOKbq8EeTTJiSsd
Jq5blTitdbclREc781L2gCR2KHbunLXNSFYBAco/Pn6DR8PPmvKIPsKDVMMHLO3u2
h/nrMmUMZ8t9u4TxMsX/V6DwdPVnmv8YWOFQdQGGMOlpJfEArp4j9+yPq7w9yMBD
uZcQyI6V91aQz8IH/RoYJOeVbMXbjRwtwcFIJIECH7LuGmVatq9x1AppFnevxFb8
bZhlTBgiPlChVrK4yttePP3/mv6ZzWSMhVDO9vVZuTzS1X9uAkk9rSBUnAFg0uj5
jF2sNw5x4UcP3qWifY0DH5XEJhl0XfcAs/olGlkJaK6PkiephcX5QlprKzUZBO
W0/XN4R2IPVUdqIUdoBQd+7WzMoqfH6/30L+zw3mXVmaxgh0YLCdDYZapkeZ/Fn
nHTVjst7UUr0y/JwRCHL7Jc7R2lppkcRkhMolbNF4JwktGog4wDDbdBDS3wrg
wRHxmjkRkZFgInR9pwteFVmh+jVgZu4Lx5GBBQc=
=fKwP
```

-----END PGP PUBLIC KEY BLOCK-----

## Appendix 16 – Token Manager Swagger

The swagger containing the description for the Token Manager REST services, used to retrieve the bin-range and httokenpans service is:

[https://github.com/pagopa/rtd-ms-transaction-filter/blob/master/ops\\_resources/swagger/acquirer-manager.yml](https://github.com/pagopa/rtd-ms-transaction-filter/blob/master/ops_resources/swagger/acquirer-manager.yml)



