



## **PolishAPI**

# **Recommendation on the application of the PolishAPI Standard for Providing Payment Initiation Services (PIS) and Account Information Services (AIS) – Extension of Address Data**

*Prepared by the PolishAPI Working Group*

16 June 2025

## Contents

|                                                                                                                          |   |
|--------------------------------------------------------------------------------------------------------------------------|---|
| 1. Problem description .....                                                                                             | 2 |
| 2. Proposed Solution.....                                                                                                | 2 |
| 3. Additional Enhancements: Optional Fields for Increased Interoperability and Data Consistency in Payment Systems ..... | 4 |
| 4. Rationale.....                                                                                                        | 5 |
| 5. Attachments .....                                                                                                     | 6 |

### 1. Problem description

In connection with the migration of subsequent payment systems to the ISO20022 standard, there has been a demand for utilizing an expanded set of data within payment messages. Specifically, this pertains to a new, structured format of address data. Previously used data was limited to 140 characters, broken into four lines of 35 characters each, and included both the name and address of the transaction party. Additionally, this information was unstructured, and correct interpretation depended solely on market practice defining the meaning of each line, which was not always reliably effective.

Under the new ISO20022-based XML message standards, this data is significantly expanded and structured. The original standard allows for up to 140 characters for the name and a total of 575 characters for the address, divided into fourteen elements. While adapting the PolishAPI standard, it is essential to remember that some payment systems have not yet migrated; therefore, support for the current format must be maintained.

Polish banks, in conjunction with the consolidation of T2/T2S and the migration of SWIFT, have begun implementing changes in their interfaces to fully or partially utilize the new structure. Due to the ongoing migration of SORBNET2 to SORBNET3 and the planned migration of Elixir to XML, clients increasingly encounter new interfaces. However, the scope of information permitted during payment initiation and account information retrieval within the PolishAPI standard remains unchanged, limited to 140 characters for transaction parties.

This situation could, as a consequence, increase the risk of allegations of discrimination against third-party providers, as the current API standard is more restrictive than the bank's native interfaces.

### 2. Proposed Solution

To align the capabilities of PolishAPI with those of native interfaces, the proposed approach involves extending the PIS (Payment Initiation Service) and AIS (Account Information Service) standards

regarding transaction parties. The guiding principle is minimal changes and maximum backward compatibility.

The changes involve expanding the classes RecipientPIS and SenderPIS within PIS services, as well as SenderRecipient in AIS services, by adding a new attribute nameAddressStructured. This attribute, in turn, should reference a new class NameAddressStructured. Additionally, within RecipientPIS and SenderPIS, the nameAddress attribute will become optional. The revised bank documentation will specify whether support for nameAddress or nameAddressStructured is mandatory.

Furthermore, in consultation with the sector, additional attributes have been introduced: ultimateSender, ultimateRecipient, and ultimateParty, referencing the class UltimateParty, as well as the identifiers attribute within the Identifiers class. The addition of these attributes relates to extending native interfaces with corresponding elements; however, support by some providers may be partial. In such cases, internal documentation should indicate that these elements will be ignored in supported services.

The NameAddressStructured class consists of a mandatory name attribute (string, max. 140 characters), an optional addressStructured attribute referencing the AddressStructured class, and, following market consultations, an optional adrLine attribute (up to 140 characters). The AddressStructured class will represent a complete structured address conforming to the ISO20022 standard. The adrLine attribute is intended to support the presentation of unstructured or hybrid address data, and can also be used within PIS services to input addresses in hybrid form, provided the provider supports this address type.

The UltimateParty class is invoked from the relevant transaction party element—SenderPIS and RecipientPIS for PIS services, or sender or recipient attributes pointing to SenderRecipient in AIS services. This class uses nameAddressStructured to invoke NameAddressStructured and identifiers to invoke Identifiers. Due to the structure of UltimateDebtor and UltimateCreditor in ISO messages, data for these entities cannot be entered in unstructured form. Thus, this class does not reference nameAddress attributes, which remain accessible via the sender and recipient objects.

The Identifiers class contains two mandatory attributes—idType and idValue—and an optional issuer attribute. Its purpose is to enable providing LEI or other equivalent identifiers, in compliance with Regulation 1113/2023. All attributes are designed to be optional since their submission is conditional. Internal documentation should specify formatting constraints per identifier type. The class initially supports two identifier types: LEI and TXID, as agreed in the sector.

Adding this new class within the SenderRecipient attribute of AIS services allows reading information in both existing and new formats without modifying the current NameAddress class. To minimize

changes, it is recommended that name always contain the transaction party's name, with full address information (name and address) being presented as before for transactions with limited data scope. For the expanded data scope, all permissible address lines in the NameAddress class will be supplemented with the transaction party's name, while the structured address elements will be returned within NameAddressStructured.

In cases of hybrid address formats, the provider will decide whether to concatenate and truncate the unstructured address data (up to 140 characters) into NameAddress, or to reserve this element solely for the name, leaving all address components—both structured and unstructured—in NameAddressStructured. For third-party institutions desiring, where applicable, data solely in structured form, the transaction party's name will also be duplicated within the name attribute of NameAddressStructured. This approach guarantees support for key data elements even if third-party providers do not implement changes on their side.

The proposed YAML configuration file, which includes the service structure and data model for AIS services, is provided as an attachment to this document.

### 3. Additional Enhancements: Optional Fields for Increased Interoperability and Data Consistency in Payment Systems

Furthermore, to maintain consistency with SORBNET 3, the proposal introduces a new class Purpose, with two attributes: Code and Proprietary. This addition is included in the classes TransferDataBase (for PIS services) and TransactionInfo (for AIS services), as this allows mapping almost all services. It should be emphasized that this solution is optional and primarily aimed at standardizing and streamlining the reporting of transaction motives between banking system parties and regulatory authorities.

Additionally, an optional EndToEndId field has been added for domestic transfers within SORBNET, SEPA transfers, and SWIFT transactions. The goal is to increase the compliance of PolishAPI with international systems such as SEPA and SWIFT, where EndToEndId plays a vital role in transaction identification. Support for this reference by SORBNET has also begun, justifying its inclusion in the standard.

This new field has been incorporated in a way that preserves existing implementations and ensures full backward compatibility. Its support will depend on the specific payment type and internal preferences of individual banks.

This change has been implemented within the base classes TransferDataBase and TransactionInfo, enabling broad coverage across PIS and AIS services without requiring new classes or modifications to

specialized transactions, such as ZUS, tax transfers, or card transactions, which typically do not support this reference. This approach allows for consistent and efficient standard extension with minimal disturbance to existing structures.

These modifications are proposed for the two most common standard versions in the market: version 3.0 and version 2.1.1.

## 4. Rationale

Due to the urgent need to address the issues described above, the proposed changes focus on implementing the necessary minimum modifications to mitigate risks of discrimination allegations. Thanks to limited changes and the principle of maximum backward compatibility, these updates are minimal and require only a three-month notification period for providers.

The risks related to discrimination claims materialize with the rollout of SORBNET 3. Its deployment will generate extensive changes in native bank interfaces for domestic payment instructions, which will significantly expose the disparity between native interfaces and the API, thus increasing existing risks.

It is also worth noting that implementing small, incremental adjustments now within PolishAPI will enable proper functionality of PIS and AIS services in the future when unstructured addresses are no longer accepted. These modifications are planned to fully support the use of structured addresses without requiring technological changes in the standard. Only documentation updates will be necessary to prohibit unstructured elements in specific or all payment types, depending on the migration status of individual systems in the Polish market.

## 5. Attachments

1. Zmiany\_PolishAPI-ver3\_0.yaml
2. Zmiany\_PolishAPI-ver2\_1\_1.yaml