



PSD2 meets ISO 20022

How will service providers and banks implement ISO 20022 in the APIs being created for PSD2?

Introduction

With less than a year to go until the EU's Revised Payment Services Directive (PSD2) is transposed into domestic law, banks are working now on their implementation projects. Most articles on this subject focus on API connectivity and the governance of this connection. However, one of the few prescriptive elements of the Directive's technical standards mandates the use of ISO 20022 to define the data. We believe that this creates serious challenges for the diverse range of financial institutions and industry groups that are involved.

The background to PSD2

The PSD2 Directive aims to stimulate new entrants into the provision of innovative banking services by opening up the data silos of the traditional banks. Two specific types of third party service are identified: payments initiation services (PIS) and account information services (AIS) – the latter being the provision of aggregated account information.

The key point is that under the rules, banks will be obliged to provide licensed third parties with secure access to bank customers' accounts. In practice the banks will provide this access via open APIs.

The basic idea of open APIs in banking is not new. Over the last few years a growing list of banks - including Crédit Agricole, Standard Chartered, Citi, CapitalOne, Saxo Bank, HSBC and digital banks such as Monzo and Starling - have launched open APIs. These banks aim to attract fintechs and other

front-end service providers to write apps that take advantage of the bank data made available via the API.

So far, each bank has created its own API without any reference to any other bank. A provider of, say, an aggregated accounts display would thus have to use a different interface for each bank where an account was held. This led some to hope that regulators would provide or enable a move towards industry-wide standard APIs.

There have indeed been some moves in this direction, primarily in the UK. For example the UK's Open Banking Working Group (OBWG) produced its first report 'The Open Banking Standard' in February 2016. More recently the Competition and Markets Authority (CMA) made the development of standardised open APIs the principal 'remedy' in its report on the UK's retail banking industry, and tasked the biggest banks with setting up an Implementation Entity to carry on and complete the OBWG's work. The final delivery date was deliberately chosen to be 13 Jan 2018, the date of application of PSD2.

Meanwhile pan-European payments initiative the Berlin Group is working on a detailed 'Access to Account Framework' with data model (at conceptual, logical and physical data levels) and associated messaging, aiming to publish this in the autumn of 2017.

However, the regulatory technical standards (RTS) for PSD2, which are now being finalised by the European Banking Authority (EBA), do *not* in themselves include any overarching requirement for standardisation of the API (or 'dedicated interface' to use its language), apart from a statement that the data structures should be based on ISO 20022. The banks are free to design their own PSD2-compliant APIs, provided only that they meet the various security requirements, and provide free technical documentation to any service provider which wants to use them.

The challenges of ISO 20022

API developer Intive, responding to the EBA's draft RTS, have pointed out that ISO 20022 does not fit well with the mainstream of API development. No open banking APIs currently exposed by the banks directly use ISO 20022 messages or elements. Most of existing interfaces are RESTful, and as such use the transport protocol to convey semantic value, while ISO 20022 makes no provision for that. Also, most APIs use JSON for syntax of the requests and responses, but ISO 20022 currently does not define JSON syntax, and is usually implemented in XML.

Over and above these technical issues, a basic problem with ISO 20022 is that it allows for any amount of variation and development. Multiple 'variant' forms of a given message, such as the Payment Instruction, can exist at the same time. On top of this, different implementations may restrict the message – prohibiting the use of certain optional fields for example – in different ways.

The outcome is thus likely to be that the banks will create a diverse range of APIs, each of which uses a different flavour of ISO 20022 to structure its data. This will create a major challenge for the service providers who will need to call the APIs of multiple banks.

A separate issue for the banks' API designers is that of retrieving data from one or more legacy systems and restructuring it for publishing via the API, or in the case of payments initiation, a transformation in the other direction. The process might involve multiple in-house systems and require a range of different types of communication; from database updates to message exchanges.

The solution

Trace Financial's Transformer adds value in all of these areas. Transformer offers banks and service providers the tools they need to work with banking APIs including full support for RESTful services, JSON and ISO 20022 based data structures.

Transformer uniquely allows analysts or developers to address all aspects including mapping, validation, enrichment, testing and maintenance without resorting to coding or scripting.

Transformer dramatically shortens the time taken to build solutions involving complex data structures. The analyst using Transformer creates solutions directly in the intuitive Design-Time GUI. There is no coding stage, even when the required data transformation is highly complex. In this way projects remain clearly articulated and easy to maintain, and removing the old-fashioned 'spec handover' from analyst to developer eliminates a major source of risk and delay. Testing is fully integrated at all stages.

Once configurations are complete and tested they are deployed into any runtime architecture that suits the client. A Transformer configuration can be deployed into the multitude of technical infrastructures found at a client site including Java, .NET, RESTful web services, etc.

Transformer is a truly strategic solution to all the challenges and delivers one best-of-breed solution for any data standard; for deployment in any technical infrastructure; for integration in any version control system and automated testing environment; and can be used by analysts or developers without any need to resort to coding or scripting. Transformer is an essential tool for coping with these evolving and complex demands.