API Testing for Payment Service Directive2 and Open Banking

Rares Coste and Liviu Miclea

***Abstract*—In this paper a solution is described for API Testing of the Payment Service Directive 2 [abbreviation PSD2] which was adopted within EU legislations. The directive PSD2 is a general requirement from European Banking Authority for the banking industry in European Union to provide programming interfaces APIs. The solution proposed is targeting types of software testing in order to validate the Payment Service Directive within a digital banking system. It comes with great opportunities in technology to develop and test applications which need to be regulated by the PSD2. Therefor in this paper an approach will be described and especially results for API testing run against a 3rd party provider [abbreviation TPP] are available. The paper is focusing on customer benefit and cost-efficient adaptation for the PSD2.**

***Index Terms*—Payment service directive2, API, API testing, open banking, API management.**

I. Introduction

A. Payment Service Directive No.2 EU 2015/2366

Also known as PSD2 has been the subject of consultation and debate within the European Commission since 2013. The most debated and impactful parts of the PSD2 are related to the provisions on strong customer authentication for online payments and on the introduction of new 'payment initiation and account information services', operated by third party providers.

Key elements of PSD2 include opening access across the industry to payment processing services, as well as to the customer accounts held by banks. It recognizes a market demand for Payment Service Providers (PSPs) granting third parties access to their online Payment Services in a regulated and secure way also named as Third-Party Payment (TPP) service [as described by the Open Banking glossary] [1], [2]. Also, the 'Access to Accounts' (XS2A) rule, will force banks to facilitate access via API to their customer accounts and provide account information to third party apps if the account holder(client) allows/agrees it.

Account Servicing Payment Service Providers [ASPSPs as per the Open Banking glossary] for banks/financial institutions must provide account information to third party providers (TPPs) as payment initiation service providers (PISPs) or account Information service providers (AISPs) in a secured and regulated form[1]-[4]. This information which includes transaction data, credit transfer initiations, balance data, fund checks, identity verification can only be used with the consent of the customer and only by the 3rd party that has been given consent, and only for the specific purpose consented. The industry consensus is the information will need to be provided through APIs.

PSD2 aims to secure e-payments and expand the financial services.

B. Open Banking Project

The Open Banking is the process which brings together the Financial service resources and Open Data resources and enables a path for Financial Services Providers to develop new methodologies of delivering Financial Services to a customer.

Open Banking is based on the use of API access to the data pools, to the infrastructure of the regulatory compliant and to other financial services resources [1]-[4].

If a bank entity will join the Open Banking Model it will consume and provide APIs also it will be a new Financial Service Provider.

In 2015 Open Banking Working Group [annotation to be used OBWG] was established in order to explore opening of bank data in the UK could be beneficial for consumers and how these benefits could be achieved. The consolidated report recommended the creation of an Open Banking Working Group [annotation to be used OBWG] using an open application programming interface API. The following common API standard allows:

- Open access to open data – allowing anyone, from TPPs to individual customers, to access publicly-available data as pricing and product information.
- Controlled access to shared data –granting regulated TPPs access to customer-account transaction data with customer consent [3].

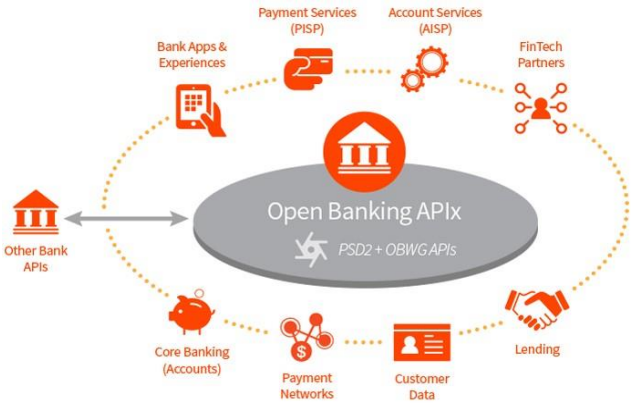

Fig. 1. Open bank standard.

PSD2 allows access to customer transactional data for specific institutions (Fig. 2), which must be regulated.

Manuscript received September 25, 2018; revised December 23, 2018.

The authors are with Automation Department of Technical University of Cluj-Napoca Cluj-Napoca, Cluj, Romania (e-mail: rares.coste@aut.utcluj.ro, Liviu.Miclea@aut.utcluj.ro).

DOI: 10.7763/IJMO.2019.V9.675

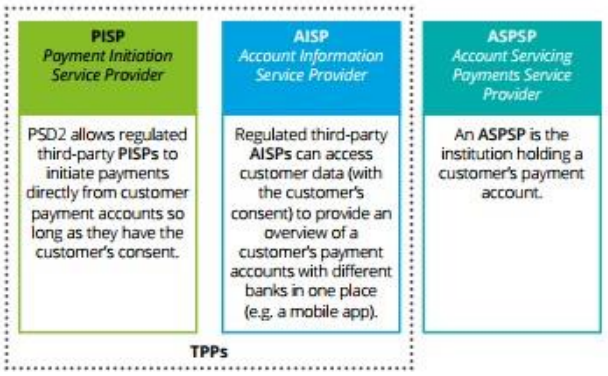

Fig. 2. PSD2 permissions.

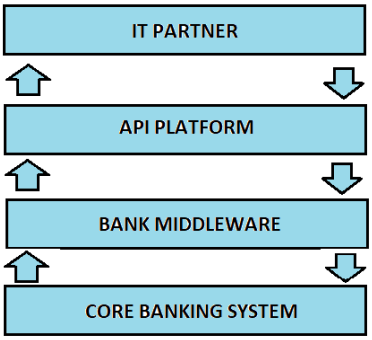

Fig. 3. API platform.

PSD2 and the Open Banking Standard are impacting the existing core and legacy systems for all payments actors [2].

The Open Banking Working Group report2 recommend the frameworks for:

- An open data API for market information and relevant open data.
- An open API for data that is shared (including customer data). This permits a business or person to consent to a Third-Party Provider to access *account level* data stored with their bank.

The framework provides recommendations on the design and delivery of the open API, including:

- API standards (specifications for design, development and maintenance of the APIs) and data standards (rules by which data are described and recorded).
- Developer resources.
- Security standards and policies, which protect customer data.

Therefor an open API needs to be implemented with all the mentioned frameworks and security in. Open data sets will also be made available via the API. The report suggests that work to be done on a phased basis.

II. Open API for PSD2

An API is an application programming interface. An API, works to connect an application to the web and to other APIs. The API is the brain of the connected world. It is a set of tools/protocols/standards and code.

The use of APIs is fundamental to the concept of Open Banking and PSD2. The requests for services and products which can deliver multichannel customers and provide relationships to these customers need a great development on the Open API sector.

API Platform. The API Platform's main function is to publish and secure APIs. The Platform is described in our case as a layer that communicates with bank middleware as shown Fig. 3.

The solution implemented for the banking system that is under test on this paper was to build their own API Platform.

The API Platforms bring a solution which is characterize by a range of capabilities, only a few will include a portal for developer teams. In our case a portal was not implemented until the testing solution started. Nevertheless, the portals are the starting point for developer teams which use this API Solution. The portals provide documentation, help in creating drafts for the API, examples in interacting with the API. Example is Deutsche Bank which has an Open Portal for developers [9].

A. API Economy

The API economy is an enabler for turning a business or organization into a platform." We live in an API economy, a set of business models and channels based on secure access of functionality and exchange of data" [5].

APIs are changing personal banking – as well the financial space with quicker processes, integrations and partnerships with Third Party Providers [6].

API management is the practice an organization implements to manage the APIs they expose. This is done internally or externally so it makes sure that the APIs are consumable, secure, and available to consumers in conditions agreed in the terms of use of APIs. The features API management should provide are the following:

- provide a place for organizations to catalog/document their APIs, incorporate metadata , provide description of the API , taxonomy of the types of API , runtime capabilities This catalog should also contain the state of the API and the currently supported versions;
- provide means to work on the catalog, exposing the APIs to internal and/or external developer communities and enforce security controls, consumption entitlements.
- API management should also provide the ability to transform the inputs and outputs accordingly, exposing a standardized form to the API consumers;
- API management should be the system of record for API utilization, updating the catalog with information regarding actual runtime behavior and characteristics of a given API. This information can include the number of API keys registered, average and peak requests per second.

B. Application Solution

In the solution that was tested through this paper the following supported flows related were treated :

- TPP management tool which casts the following testing actions:
- Create new TPP.
- Check TPP status.
- Change TPP status.
- BLOCK TPP.
- Consents – which casts the following testing actions:

- Create Consent.
- Block/Unblock/Renew Consents.
- Mock data.

In the below image an example of API Documentation in open source Swagger UI is presented. The list of operations under test is present.

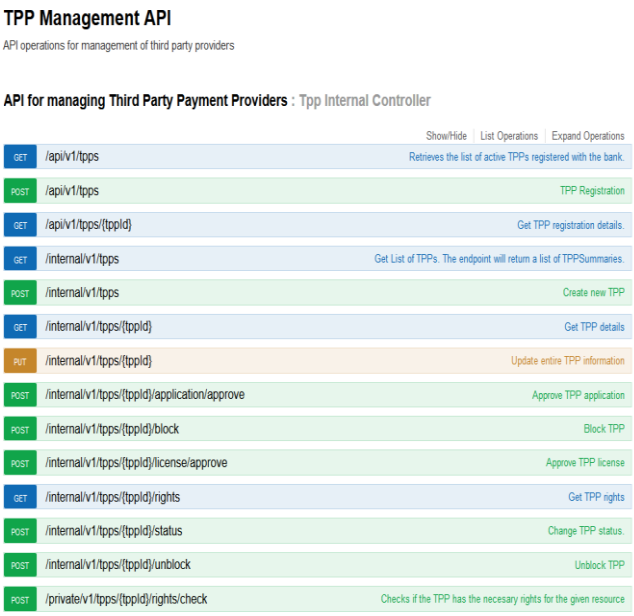

Fig. 3. API TPPs in swagger UI.

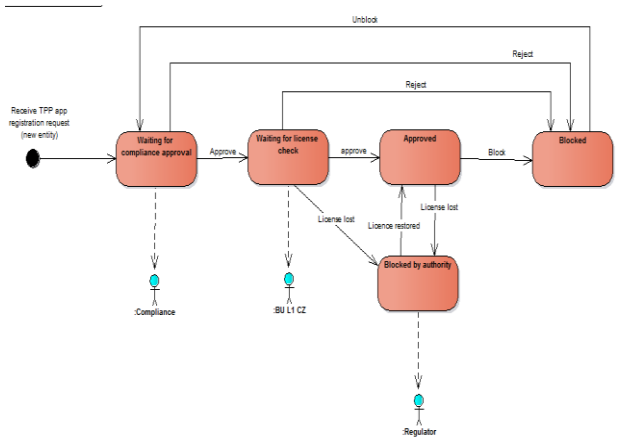

Fig. 4. TPPs state diagram.

Following operations should be allowed on the TPPs:

- checkTppRights, cacheable
- getTPPDetails
- registerTPP
- approve TPP
- updateTPP
- notify
- block TPP
- checkTppRedirectUrl

The TPP will be able to do following operations on its own:

- register with the bank.
- update the contact information.

To initialize and keep up to date the list of TPPs the Central Bank provides a list of currently registered and approved TPPs with their statuses. Our component should be able to synchronize the database information with the list provided by the Central Bank.

Except for the Central bank list also the Bank Employee will be able to modify/update the list of TPPs and their statuses. The Employee will be able to:

- getTPPList
- getTPPDetails
- approve TPP
- updateTPP
- blockTPP

The State Diagram for TPPs can be inspected bellow in Fig. 4 and it provides the statuses and the allowed transactions that were implemented.

III. Simulation and Results

The changes brought by PSD2 which are under test and analyze in the paper are as it follows:

- Third-party payment initiation (XS2A) – PSD2 directive regulates the payment initiation service providers [named PISPs). PISPs allow user to start an online payment to a e-merchant or beneficiary directly from the payer's bank account via the online portal of the PISP. This is to be another solution in using the card payments in online transactions.
- Third-party account access (XS2A) – PSD2 will regulate account information service providers [named AISPs]. These services are an aggregator of customer payment account information allowing users to log in one portal to view their account transaction history on payments and balances.

Testing is a part of an API management solution. Tests were elaborated and done in order to evaluate the PSD2 implementation. Tests were conducted against the operations that a TPP can achieve. As test tool Swagger was used. HP ALM was used a test management tools – key central place to hold the test cases composed for the PSD2 functionality

TABLE I: Error Codes / Description

Error code	Http Code	Description
TP_100	400	Mandatory field is missing in the request.
TP_101	400	To ID not found
TP_102	401	Client is not unauthorized to execute the request.
TP_103	403	Access to the requested resource is not allowed or is not possible for the user.
TP_106	403	The client is not unauthorized to execute the request because of the TPP status.
TP_107	403	Invalid Tpp status change.
TP_108	500	Server error.
TP_109	400	The url TPP id is different from the une provided in the request body.
TP_110	400	Tpp ID alreay exists
TP_111	400	Page number must not be less than zero
TPP_112	400	Page size must not be less than one
TPP_113	400	Invalid page number

Validation of Swagger and TPPs was done by investigating the results in a front-end side which is not displayed in this paper. In Swagger each API operation against TPP returns a response code which is mapped to

All error codes were validated and tested.

Create new TPP Flow tested

In order to add a new TPP, performed the following steps.

Step1: In Swagger "Create new TPP" was completed by using the following code in the body:

{ "email": "string",
"id": "tppid6",
"name": "test tpp",
"redirectUris": [
"string"
], "roles":
["PSP_AI", "PSP_AS", "PSP_IC", "PSP_PI"]
}

Step1.1: Validation of Create TPP action is done with click on "Try it out" button. If is NOT OK error will be displayed.[10]

Result = TPP status is "PENDING_APPROVAL".

Step 2. Approval of TPP application action is done by click on "Approve TPP application" button.

A tppId is added and validated.

Result = TPP status is "PENDING_LICENSE".

Step 3. Approval of TPP License action is made by click on "Approve TPP license"

A tppId is added and click on "Try it out" is made

Result = TPP status is "APPROVED".

As extra check the TPP status was viewed by the action "Get List of TPPs" and checked Response Body.The result was a entire list of available TPP with their current status displayed.

Step 4.Change TPP status

This was done from "Change TPP status"action.

Following transitions are defined in the bellow table.

TABLE II: TRANSITIONS OF TPPS

From Status	To Status
PENDING_APPROVAL	PENDING_LICENSE
PENDING_APPROVAL	BLOCKED
PENDING_APPROVAL	BLACKLISTED
PENDING_LICENSE	APPROVED
PENDING_LICENSE	BLOCKED
PENDING_LICENSE	BLACKLISTED
APPROVED	BLOCKED
APPROVED	BLACKLISTED
APPROVED	BLOCKED_BY_AUTHORITY
BLOCKED	BLACKLISTED
BLOCKED	PENDING_APPROVAL
BLOCKED_BY_AUTHORITY	APPROVED
BLOCKED_BY_AUTHORITY	BLACKLISTED
BLACKLISTED	PENDING_APPROVAL
BLOCK TPP	---

Create Consent Flow results

A Consent for TPP is synonymous with a right which is being added to that TPP. Customer is able to grant access to his accounts for the TPP applications. In summary Customer should be able to allow / disallow access to his bank services for the TPP application.

CIS TPP Enrollment happens via Contact Form. The option to enroll CIS is shown to "Logged in" customers only. After Customer submits the Contact Form with CIS enrollment request, the new consent is created for this customer.

The below figure is presenting the requests that are available to be executed/ tested for a Consent

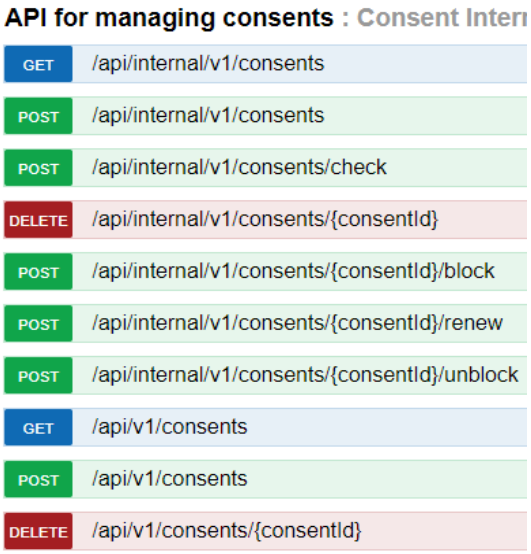

Fig. 5. API consent in swagger UI.

Step 1. In order to create a Consent we use the POST method and create a consent resource at the direct bank connection regarding the access to the accounts which are specified in the request

Step 2. Response validated is Status 201 – Created. The x-api-header must be encoded base64. ClientID is to be filled and encoded [5]-[8].

Step 3. X-user-agent-info encoding base64 is done for the following parameters:

{ "userAgent": "userAgent1",
"ipAddress": "ipAddress1",
"geoLocation": "geoLocation1",
"acceptLanguage": "acceptLanguage1"
}

Step 4. On the application side we check that for the client ID and validity that was added the Consents are present in a correct state.

For the TPP and Consent testing all the error codes were treated. As well in Swagger the validation was extended to the body request and negative results were made available to the development team. The list of attributes were validated and their type from description was taken in account(if it is a String, Resource or Amount). Mandatory fields were validated against the presented Error codes [11].

IV. CONCLUSIONS

This paper draws a systematic approach for PSD2 testing and validation. It describes the PSD2 directive and the flows that were under validation for a Financial Institution.

During the tests acknowledgement was made that TPPs and Consents are an interesting part to interact and the flows that were defined for these areas were impressive. There

was a big amount of tests created in HP ALM and run.

The validation and the results were started from Swagger and they were inspected in the Customers account which made all the result part quite challenging and interesting. As well as validating the test part brought clear results to the developers which fixed all the errors flagged by the tests.

Test Data which was created during the API Testing with Swagger will be used in the future plans in automated tests. For the automated tests the suggestion made is to use SoapUI as a test tool. Another path to follow is the elaboration of test suites which will be run and maintain at each build.

The purposes of this paper was to give hints on the flows that need to be covered for PSD2 directive testing. The future plans are to elaborate Automation test which will cover all the developed areas for PSD2 and they ca be wide used a solution for testing this directive in various Financial Institutions.

REFERENCES

[1] Website Glossary. [Online]. Available: https://www.openbanking.org.uk/about-us/glossary

[2] Want to know how industry leaders are harnessing risk. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/cz/Documents/financial -services/cz-open-banking-and-psd2.pdf

[3] C. Kankanamge, "Web services testing with SoapUI EMV integrated circuit card specifications for payment systems, common payment application specification," Packt Publishing Ltd, ISBN 978-1-84951-566-5.

[4] Electronic Money Directive - Directive 2009/110/EC of the European Parliament and of the Council of 16 September 2009 on the taking up, pursuit and prudential supervision on the business of electronic money institutions amending Directives 2005/60/EC and 2006/48/EC and repealing Directive 2000/46/EC.

[5] L. Miclea, in *Proc. 2010 IEEE International Conference on Automation, Quality and Testing Robotics*, Editura IEEE, 2010.

[6] *The API Economy Disruption and the Business of APIs 2015 - 2016 Nordic APIs*.

[7] D. S. Leaman, *NVLAP Cryptographic and Security Testing*.

[8] Data Elements' supports the new card message standards defined in ISO (International Organization for Standardization) (ISO 20022).

[9] Developer. [Online]. Available: https://developer.db.com/#/apidocumentation/apiauthorizationguide/api authguideflows

[10] Linda G. Hayes, *The Automated Testing Handbook*, Editura Software Testing Institute, 1995, ISBN: 0970746504.

[11] *Modernizing and Managing Enterprise Applications in the API Economy, International Business Machines Corporation (IBM)*.

Rares Coste is a PhD student in the Automation Department at the Technical University of Cluj-Napoca. His research interests include software testing, finance software testing, performance software testing, security software testing, digital banking and performance management. Coste works as a test manager in a fin-tech company named Nexttech International from Cluj-Napoca and also as a test manager within Consorsbank Germany.

Liviu Miclea is a full professor the Automation Department at the Technical University of Cluj-Napoca. He is also the dean of the same faculty. He is the author or co-author of 17 books, 40 research works and more than 180 scientific publications. His research interests include: dependability, cyber-physical-systems, agent systems. Miclea is a senior member of IEEE and is regular the general chairman of the bi annual IEEE-CS-TTTC-AQTR .