Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-SIT

Seidlgasse 22 / 9, 1030 Wien
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

Inffeldgasse 16a, 8010 Graz
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

ZUGRIFFSKONTROLLE VON FINANZ-APIS

Version 1.0 vom 14.06.2018
Bojan Suzic – bojan.suzic@a-sit.at
Johannes Feichtner – johannes.feichtner@a-sit.at
Peter Aufner – peter.aufner@a-sit.at

Mit der Richtlinie (EU) 2015/2366 des Europäischen Parlaments und Rates vom 25. November 2015 wurden neue Rahmenbedingungen für eine Harmonisierung von Zahlungsdiensten im europäischen Binnenmarkt beschlossen. Als neue Interaktionsmodelle wird die Bereitstellung von APIs von Banken erwartet, so dass Finanzdaten und Dienste breiter konsumiert und ausgetauscht werden können. Ziel dieses Projektes ist es, exemplarisch bekannte Schnittstellen zu analysieren und potentiell sicherheitsrelevante Aspekte zu beleuchten. Konkret werden die vier APIs NextGenPSD2, STET, SBAS und Priora untersucht.

Inhaltsverzeichnis

1. Einführung

Daten gelten bereits für mehrere regionale und internationale Behörden als wichtige Triebkraft für Innovation und Wettbewerbsfähigkeit. So wurden in den letzten Jahren zahlreiche wissenschaftliche, organisatorische sowie juristische Initiativen mit dem Ziel gesetzt, die Verwendung, Integration und Distribution von Daten im breiteren Sinn in allen wirtschaftlichen Prozessen zu unterstützen. Eine der jüngsten Unternehmungen in dieser Richtung ist auch durch die Mitteilung „Aufbau eines gemeinsamen europäischen Datenraums“ der Europäischen Kommission festgelegt [1], die Maßnahmen zur effizienten Nutzung von Daten im gesamten europäischen Wirtschaftsraum vorsieht.

Die Notwendigkeit der Modernisierung von Finanzunternehmen ist in den letzten Jahren durch diverse Faktoren evident geworden. Unter dem Schirm des Finanzsektors werden täglich zahlreiche, für die gesamte Wirtschaft und Gesellschaft wichtige Aktivitäten ausgeführt. Dennoch existieren viele Hürden, die eine Anwendung neuester Technologien unter Berücksichtigung geschäftlicher Entwicklungen potenziell verlangsamen. Als wichtige Maßnahme in Richtung der Abschaffung solcher Hürden und zum Aufbau neuer Märkte wurde die PSD2-Richtlinie erarbeitet [2].

Dieses Projekt analysiert wesentliche technische Aspekte dieser neuen Regelung. Insbesondere betrachten wir den Aufbau vorgeschlagener Standardisierungen von Finanz-APIs mit besonderem Augenmerk auf die Sicherheitsaspekte solcher technischen Regelungen. Web-APIs stellen einen wichtigen Baustein des Internets dar, da sie einen systemunabhängigen Austausch von Daten zwischen mehreren Akteuren ermöglichen.

Diese Studie ist folgendermaßen aufgebaut: Im zweiten Kapitel beschreiben wir wichtige, durch die PSD2-Richtlinie neu eingeführte, technischen Aspekte. Dabei werden insbesondere neue Rollen für Teilnehmer am Finanzmarkt sowie wichtige Prozesse berücksichtigt. Das dritte Kapitel befasst sich mit dem Aufbau von vier relevanten Lösungen für PSD2-kompatible APIs. Dabei werden die einzelnen Operationen dargestellt, sowie die Struktur von APIs und zugehöriger Datensätze analysiert. Auf dieser Darstellung baut die nachfolgende Analyse von autorisierungsrelevanten Mechanismen und deren praktischen Anwendungen auf. Diese Aspekte sind auch für die Freigabe von Daten und API-Operationen von größter Relevanz, da sie die zugrundeliegende Zugriffsteuerung sowie deren Eigenschaften wesentlich beeinflussen. Diese Studie schließt mit einer Zusammenfassung, die die gewonnenen Erkenntnisse aufschlüsselt.

2. Technische Aspekte der PSD2-Richtlinie

Mit der Richtlinie (EU) 2015/2366 des Europäischen Parlaments und Rates vom 25. November 2015 wurden neue Rahmenbedingungen für eine Harmonisierung von Zahlungsdiensten im europäischen Binnenmarkt beschlossen. Es war vorgesehen, dass Mitgliedsstaaten die auch als „Payment Service Directive 2“ (PSD2) bekannte Richtlinie bis 13. Jänner 2018 in nationales Recht umgesetzt haben. In Österreich erfolgt eine Umsetzung durch Änderung des Zahlungsdienstegesetz (ZaDiG)[1].

Zu den wesentlichen Neuerungen von PSD2 gehört die an bestehende Zahlungsdienstleister gerichtete Verpflichtung, Schnittstellen bereitzustellen, um unter gewissen Voraussetzungen auch Dritten Zugriff auf sensible Kontodaten zu ermöglichen. Das Einverständnis von KundInnen vorausgesetzt, können Dienstleister fortan sowohl auf Kontoinformationen (z.B. Saldo, Transaktionshistorie der letzten 90 Tage, etc.) zugreifen, als auch Zahlungsvorgänge auslösen. Zahlungsdienstleistern wird auferlegt, „*die Vertraulichkeit und die Integrität persönlicher Sicherheitsmerkmale zu schützen*“ (Art. 97, RL 2015/2366), indem angemessene Sicherheitsvorkehrungen in Form „starker Authentifizierungsverfahren“ (Art. 4, Nr. 30, RL 2015/2366) bereitgestellt werden müssen.

Für die technische Ausgestaltung der Authentifizierung und Kommunikation über Schnittstellen von Zahlungsdienstleistern wurde vorgesehen, dass die „European Banking Authority“ (EBA) in Zusammenarbeit mit ausgewählten Akteuren die technischen Regulierungsstandards („Regulatory Technical Standards“, RTS) ausarbeitet (Art. 98, RL 2015/2366). Zusätzliche Ausführungen zur

[1] https://www.parlament.gv.at/PAKT/VHG/XXV/ME/ME_00332/index.shtml

PSD2-Richtlinie werden darüber hinaus im Rahmen von „Implementing Technical Standards“ (ITS) und weiteren „Guidelines“ festgelegt[2][3].

Im Kontext dieses Projekts ist insbesondere der technische Regulierungsstandard betreffend „starke Kundenauthentifizierung und sichere Kommunikation“ gem. Art. 98 (1), RL 2015/2366 relevant. Am 27.11.2017 wurde von der EU-Kommission die finale Version des RTS präsentiert[4] und wartet zum gegenwärtigen Zeitpunkt auf eine Veröffentlichung im Amtsblatt der Europäischen Union[5]. 18 Monate später – voraussichtlich also in der zweiten Jahreshälfte 2019 – erhalten die Vorgaben des RTS verbindlichen Rechtscharakter.

In den nachfolgenden Abschnitten wird zunächst ein Überblick über die von der PSD2-Richtlinie neu definierten Akteure und Prozesse gegeben. Anschließend werden die Sicherheitsanforderungen an Schnittstellen gem. Art 98 der Richtlinie und dem RTS auf ihre praktische Umsetzbarkeit hin beleuchtet. Darauf aufbauend werden schließlich konkrete technische Gestaltungsmöglichkeiten der Schnittstellen im Kontext sicherheitsrelevanter Eigenschaften präsentiert.

2.1. Akteure und Prozesse

Entsprechend der durch die PSD2-Richtlinie neu geschaffenen Rahmenbedingungen werden für die Erbringung und Nutzung von Zahlungsdiensten neue Begrifflichkeiten (Art. 4, RL 2015/2366) eingeführt, die die wesentlichen Akteure gemäß ihren Verantwortlichkeiten benennen:

- **Kontoführender Zahlungsdienstleister**
  Account Servicing Payment Service Provider – ASPSP

  „*Ein Zahlungsdienstleister, der für einen Zahler ein Zahlungskonto bereitstellt und führt*“ (Art. 4, Nr. 17, RL 2015/2366). In der Praxis sind kontoführende Zahlungsdienstleister somit in erster Linie traditionelle Finanzinstitutionen, über die KundInnen auch bisher bereits Konten verwalten und Zahlungen abwickeln konnten.

- **Zahlungsdienstnutzer[6]**
  Payment Service User – PSU

  „Eine natürliche oder juristische Person, die einen Zahlungsdienst als Zahler oder Zahlungsempfänger oder in beiden Eigenschaften in Anspruch nimmt“ (Art 4, Nr. 10, RL 2015/2366).

- **Zahlungsauslösedienstleister**
  Payment Initiation Service Provider – PISP

  „*Ein Zahlungsdienstleister, der gewerbliche Tätigkeiten nach Anhang 1 Nummer 7 ausübt*“ (Art. 4, Nr. 18, RL 2015/2366). In dieser mit der PSD2-Richtlinie neu geschaffenen Rolle wird es tertiären Dienstleistern ermöglicht, die im Anhang 1, Nr. 7 genannten Zahlungsdienste gegenüber KundInnen anzubieten. ZahlerInnen haben nach Art. 66 der PSD2-Richtlinie das Recht, einen Zahlungsauslösedienst für Zahlungen zu nutzen, sofern er sich bei jeder Zahlung gegenüber dem ASPSP identifiziert und auf sichere Weise mit ihm kommuniziert. Der Zahlungsauslösedienstleister steht bei Zahlungen somit zwischen ZahlerInnen und ASPSP.

---

[2] https://www.eba.europa.eu/documents/10180/87703/EBA+Mandates+PSD2.pdf/5c2493a4-ef26-4434-8338-736895bd423f
[3] https://www.eba.europa.eu/regulation-and-policy/payment-services-and-electronic-money/-/activity-list/MgjX6aveTl7v/more
[4] http://europa.eu/rapid/press-release_MEMO-17-4961_en.htm
[5] http://eur-lex.europa.eu/legal-content/DE/ALL/?uri=pi_com:C(2017)7782
[6] In direkter Übernahme des Wortlauts aus der PSD2-Richtlinie werden als Definition daraus die Begrifflichkeiten ausnahmsweise nicht geschlechtsneutral transformiert.

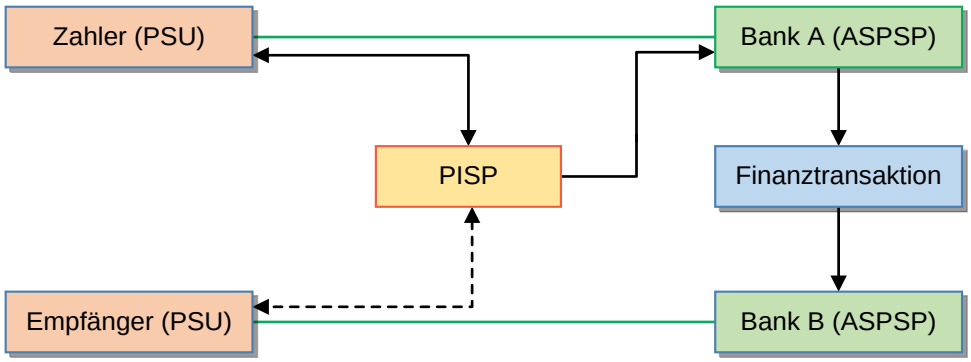

Abbildung 1. Interaktion mit PISP[6]

Abbildung 1 illustriert die Abfolge von Finanztransaktionen unter Zuhilfenahme eines Zahlungsauslösedienstleisters. Grün hinterlegte Verbindungen zwischen PSU und ASPSP erinnern an das klassische Modell eines Zahlungsablaufes, das in rechtsvertraglicher Hinsicht auch weiterhin beibehalten wird. Der Ablauf eines Zahlungsvorganges mithilfe eines PISP ist wie folgt:

1. Ein Zahlungsdienstnutzer[6] beauftragt ausdrücklich einen PISP, eine Finanztransaktion an eine gewisse EmpfängerIn durchzuführen. Der PISP darf hierfür vom PSU nicht mehr Daten verlangen, als für das Erbringen des Auslösedienstes erforderlich, weiters darf er sie nur für den vorgesehenen Zweck verwenden und speichern, und Daten des Zahlungsvorganges (z.B. EmpfängerIn oder Betrag) nicht ändern.
2. Für die Durchführung einer Zahlung ermächtigt, interagiert der PISP mit einem kontoführenden Zahlungsdienstleister (ASPSP), wo das Vermögen der/des ZahlerIn verwaltet wird. Der PISP muss sich hierfür gegenüber dem ASPSP identifizieren und mit ihm auf sichere Weise kommunizieren.
3. Der ASPSP ist nach Eingang des Zahlungsauftrags angewiesen, dem PISP über einen sicheren Kommunikationskanal Details zur tatsächlichen Auslösung des Zahlungsvorganges mitzuteilen. Im Zuge der tatsächlichen Finanztransaktion findet eine Überweisung von ASPSP zu ASPSP statt, bei der der PISP nicht involviert ist.
4. Wurde der PISP vom ASPSP über die erfolgreich durchgeführte Transaktion informiert, kann er, das explizite Einverständnis einer zahlenden Person vorausgesetzt, der empfangenden Person Informationen über die zahlende Person bzw. die Transaktion mitteilen.

- **Kontoinformationsdienstleister**
  Account Information Service Provider – AISP

  „Ein Zahlungsdienstleister, der gewerbliche Tätigkeiten nach Anhang 1 Nummer 8 ausübt“ (Art. 4, Nr. 19, RL 2015/2366). Nach Art. 67 der PSD2-Richtlinie haben ZahlerInnen das Recht, Kontoinformationsdienstleister zu verwenden, die rein *lesend* auf Zahlungsvorgänge in Zahlungskonten zuzugreifen. Im Einklang mit den angeführten Einschränkungen zur Verwendung können AISP die Daten nutzen, um ihrerseits Dienstleistungen gegenüber ZahlerInnen anzubieten. Vorstellbar ist beispielsweise, dass AISP im Auftrag von ZahlerInnen auf Daten (Saldi, Transaktionshistorie, etc.) bei mehreren ASPSP zugreifen und das Ergebnis dann den NutzerInnen in aufbereiteter Form präsentiert wird.

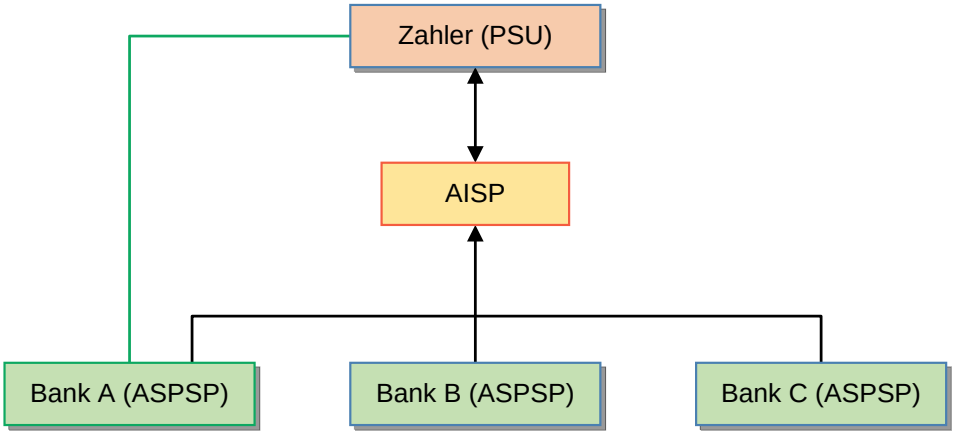

Abbildung 2. Interaktion mit AISP

In Abbildung 2 werden Interaktionsabhängigkeiten des Kontoinformationsdienstleisters in Bezug auf PSU und eine oder mehrere ASPSP dargestellt. Wesentlich ist dabei, dass nur der PSU mit jedem ASPSP in einer rechtsvertraglichen Beziehung steht. Der AISP selbst unterhält keine vertragliche Beziehung zum ASPSP und verwaltet oder verfügt zu keiner Zeit über das Finanzvermögen von ZahlerInnen. Der konkrete Ablauf eines lesenden Zugriffs auf Zahlungskonten geschieht wie folgt:

1. Ein Zahlungsdienstnutzer[6] ermächtigt einen AISP ausdrücklich, auf Zahlungsvorgänge beim ASPSP zuzugreifen. Der AISP muss dabei sicherstellen, dass die Kommunikation mit dem PSU auf sichere Weise passiert und Daten nur für den Zweck des Kontoinformationsdienstes verarbeitet und gespeichert werden.
2. Der AISP fordert von einem oder mehreren ASPSP Daten an, die Auskunft über Zahlungsvorgänge geben. Im Einklang mit den Datenschutzvorschriften darf es sich dabei um keine sensiblen Daten handeln.
3. ASPSP müssen mit AISP auf sichere Weise kommunizieren.

Für die Interaktion mit PISP und AISP sieht Art. 97 der PSD2-Richtlinie vor, dass ZahlerInnen sich über „starke Authentifizierungsmechanismen“ identifizieren müssen. Wie eingangs bereits erwähnt, müssen Zahlungsdienstleister darüber hinaus sicherstellen, dass Vertraulichkeit und Integrität von Sicherheitsmerkmalen gewahrt werden. Analog dazu ist vorgesehen, dass auch ASPSP gegenüber Zahlungsdienstleistern dieselben Authentifizierungsverfahren anbieten, die jene für PSU bereitstellen (Art. 97, Nr. 5, RL 2015/2366).

Art. 98 der RL 2015/2366 delegiert die Ausarbeitung technischer Regulierungsstandards für die sichere Authentifizierung und Kommunikation an die EBA. Durch die Festlegung von Erfordernissen, Ausnahmen und Anforderungen an offene Sicherheitsstandards für die Kommunikation zwischen Zahlungsdienstleistern (PISP, AISP), ASPSP und Zahlungsdienstnutzern[6] soll (neben weiteren Motiven) ein angemessenes Sicherheitsniveau für die Authentifizierung, Meldung und die Weitergabe von Informationen geschaffen werden.

2.2. Authentifizierung & Kommunikation nach PSD2

Die technische Ausgestaltung der Maßnahmen für sichere Kommunikation zwischen den Akteuren zu Vertraulichkeit, Integrität und „starke Authentifizierung" findet sich im technischen Regulierungsstandard „*RTS on strong customer authentication and common and secure communication under Directive 2015/2366 (PSD2)*"[7]. Die darin formulierten Festlegungen haben unmittelbare Relevanz für die Gestaltung von PSD2-konformen Schnittstellen und werden daher in einer Zusammenfassung der wesentlichen Aspekte nachfolgend vorgestellt.

2.2.1. Authentifizierungsmechanismus

Die PSD2-Richtlinie schreibt Zahlungsdienstleistern vor, „starke Kundenauthentifizierung" zu verlangen, „*wenn der Zahler a) online auf sein Zahlungskonto zugreift, b) einen elektronischen Zahlungsvorgang auslöst, c) über einen Fernzugang eine Handlung vornimmt, die das Risiko eines Betrugs im Zahlungsverkehr oder anderen Missbrauchs birgt*" (Art. 97, RL 2015/2366). Die daraus abgeleiteten technischen Maßnahmen für Schnittstellen können wie folgt zusammengefasst werden:

- *Allgemeine Anforderungen an die Authentifizierung:* Zahlungsdienstleister müssen Transaktionsüberwachungsmechanismen einsetzen, um nicht autorisierte oder betrügerische Zahlungsvorgänge zu erkennen. Die eingesetzten Mechanismen müssen dabei sicherstellen, dass gewisse risikobasierte Faktoren (z.B. „*Anzeichen für eine Malware-Infektion bei einer Sitzung während des Authentifizierungsverfahrens*" nach Art. 2, Nr. 2d, RTS) hinreichend einbezogen werden.
  Praktisch bedeutet das, dass sämtliche Zugriffe auf PSD2 APIs nicht nur mitsamt aller gesendeten und empfangenen Parameter protokolliert, sondern darüber hinaus Maßnahmen zur Erkennung von anomaler Verwendung gesetzt werden müssen. Bei vergleichbaren Motiven wird hierfür auf „Intrusion Detection Systeme" zurückgegriffen, die für den jeweiligen Einsatzzweck adaptiert werden müssen.
- *Authentifizierungscode:* „Starke Kundenauthentifizierung" muss über Zweifaktorauthentifizierung, im Zuge derer ein einmal verwendbarer, fälschungssicherer Authentifizierungscode generiert wird, stattfinden. Nach spätestens fünf Minuten ohne BenutzerInnen-Interaktion soll der Code seine Gültigkeit verlieren (Art. 4, RTS).
- *Dynamische Verknüpfung*: Der jeweilige Authentifizierungscode ist an einen gewissen Zahlungsbetrag und ZahlungsempfängerIn, die der zahlenden Person angezeigt werden, gebunden. Eine Änderung von Betrag oder EmpfängerIn würde auch die Integrität des Codes verletzen bzw. ihn ungültig machen (Art. 5, RTS).
- *Anforderungen an Authentifizierungsfaktoren:*
  Gem. Art. 9, RTS muss bei der Mehrfaktorauthentifizierung sichergestellt werden, dass die eingesetzten Faktoren in voneinander getrennten, von Dritten nicht veränderbaren Ausführungsumgebungen operieren. Darunter fallen folgende Faktoren:
  - *Wissen:* Bei wissensbasierten Faktoren müssen Maßnahmen getroffen werden, um ihre Vertraulichkeit maximal zu schützen (Art. 6, RTS). Wird in der Praxis hierfür ein Passwort verwendet, könnten zur Risikominimierung beispielsweise Vorgaben zu Länge und beinhaltenden Zeichen gemacht werden.
  - *Besitz:* Wird auf den Faktor Besitz zurückgegriffen, wäre die notwendige Vertraulichkeit verletzt, wenn eine Replikation möglich wäre (Art. 7, RTS).
  - *Inhärenz:* Faktoren, die direkt mit ZahlerInnen assoziiert sind, wie etwa biometrische Methoden (Fingerabdruck, Iris-Scan, Stimmerkennung), müssen hinreichend fälschungssicher sein (Art. 8, RTS).
- Vertraulichkeit und Integrität persönlicher Sicherheitsmerkmale: Vom PSU während der Authentifizierung eingegebene persönliche Sicherheitsmerkmale müssen durch Verschlüsselung seitens des Zahlungsdienstleisters geschützt werden. Weder der gewählte Schlüssel, noch die personalisierten Sicherheitsmerkmale dürfen dabei jemals im Klartext abgespeichert werden (Art. 22, RTS). Inwieweit sich dieser Punkt in der Praxis auch auf die Ausgestaltung von PSD2-konformen Schnittstellen erstreckt, hängt davon ab, an welcher Stelle des Prozesses ein Zahlungsdienstleister die Verschlüsselung einsetzt.

[7] http://eur-lex.europa.eu/legal-content/DE/ALL/?uri=pi_com:C(2017)7782

2.2.2. Sichere Kommunikation

Neben Maßnahmen zur Authentifizierung spezifiziert der RTS auch explizite technische Anforderungen an Schnittstellen für Zahlungsdienstleister:

- *Anforderungen an die Identifizierung:* Zahlungsdienstleister haben sicherzustellen, dass „*die Risiken einer Fehlleitung der Kommunikation an Unbefugte wirksam eingedämmt wird*“ (Art. 28, RTS).Gängige technische Ansätze hierfür wären beispielsweise die Verwendung von DNSSEC (RFC 4033[8]) und „HTTP Public Key Pinning“ (RFC 7469[9]). Beiden Ansätzen ist gemein, dass sie die Identität des Ziels mithilfe einer kryptographisch eindeutigen Angabe möglichst präzisieren. Dies ermöglicht dem Gerät bzw. einer Anwendung auf der Seite des Zahlungsdienstnutzers, eine Fehlleitung zu erkennen.
- *Rückverfolgbarkeit:* Die Kommunikation von Zahlungsdienstleistern mit jeglicher anderen Entität wird als Sitzung gesehen, die sich durch eine eindeutige Kennung ausweist (Art. 29, RTS). Praktisch impliziert dies, dass Schnittstellen eine Verwaltung von Sitzungen (Session Management) benötigen, um Protokollen eine eindeutige Kennung zuweisen zu können.
- *Allgemeine Anforderungen an Zugangsschnittstellen:*
  Kontoführende Zahlungsdienstleister (ASPSP) müssen für PISP und AISP Schnittstellen mit folgenden Eigenschaften bereitstellen:
  - *Sichere Kommunikation:* Im konkreten Kontext bezieht sich die Sicherheit der Kommunikation auf die vertrauliche Auslösung von Zahlungen bzw. Anforderung von Informationen (Art. 30, Nr.1, RTS). Sollte eine auf Webtechnologien basierende Schnittstelle zum Einsatz kommen, wäre somit der Einsatz von TLS unabdinglich.
  - *Aufforderung zur Authentifizierung:* Die Schnittstelle des ASPSP kann den Zahlungsdienstleister anweisen, die Authentifizierung zu starten, während die Kommunikationssitzung zwischen ASPSP, Zahlungsdienstleister (AISP oder PISP) und PSU unter Wahrung von Integrität und Vertraulichkeit aufrecht gehalten wird.
  - *Standardkonformität:* Die vom ASPSP verwendeten Schnittstellen müssen gegenwärtigen internationalen oder europäischen Standards entsprechen und dokumentiert sein (Art. 30, Nr. 5, RTS). Es müssen darüber hinaus Testumgebungen bereitgestellt werden, die den gesamten Prozessfluss abbilden. Ein Austausch sensibler Informationen (Zugriff auf echte Kundendaten) darf jedoch nicht stattfinden.
- *Anforderungen an dedizierte Schnittstellen:* Wenn ASPSP eigene Schnittstellen zur Authentifizierung und sicheren Kommunikation bereitstellen, müssen sie sicherstellen, dass die Schnittstellen verfügbar, leistungsfähig und wirksam sind (Art. 32, RTS). In der Praxis impliziert dies die Notwendigkeit für eine permanente Überwachung und Ausfallsicherheit.
- *Zertifikate:* AISP und PISP müssen sich gegenüber ASPSP mithilfe qualifizierter Zertifikate für elektronische Siegel (Art. 30 (3), eIDAS-VO[10]) oder für Website-Authentifizierung (Art. 30 (39), eIDAS-VO) identifizieren (Art. 34, RTS). Im verwendeten Zertifikat muss dabei als Registriernummer des jeweiligen AISP oder PISP, die Rolle des Zahlungsdienstleisters, sowie der Name der für den Zahlungsdienstleister zuständigen Behörde eingetragen sein.
- *Sicherheit von Kommunikationssitzungen:* In Ergänzung der Anforderungen an sichere Kommunikation in Art. 30, Nr. 1, RTS, spezifiziert Art. 35, RTS die Notwendigkeit des Einsatzes weithin als sicher geltender Verschlüsselungstechnologien während einer Kommunikationssitzung. Im Praxisbezug erinnern die Ausführungen zu den Anforderungen an die durch TLS möglichen Sicherheitseigenschaften. Der Einsatz einer als sicher geltenden „Cipher suite“ bei TLS-Verbindungen ist eine Voraussetzung dafür, dass Vertraulichkeit, Authentizität und Integrität von ausgetauschten persönlichen Sicherheits-merkmalen gewahrt werden kann.
- *Datenaustausch:* ASPSP sind angehalten, in ihren Schnittstellen keine Informationen vorzuenthalten, die einem PSU auch bei direkter Kommunikation mit dem ASPSP gezeigt werden würden (Art. 36, RTS). Damit ein PISP weiß, ob eine Finanztransaktion möglich ist, muss der ASPSP auf Verlangen eine Antwort in Form eines einfachen „Ja“ oder „Nein“ bereitstellen.

[8] https://tools.ietf.org/html/rfc4033
[9] https://tools.ietf.org/html/rfc7469
[10] „eIDAS“-Verordnung (EU), Nr. 910/2014 des Europäischen Parlaments und Rates

3. Analyse von APIs

In diesem Kapitel stellen wir eine Übersicht von Funktionen bzw. durch die APIs zur Verfügung gestellten Daten vor. Es wird jede in Bezug auf PSD2 relevante Rolle gesondert betrachtet. Hier ist die AISP-Rolle besonders relevant, da ihr Funktionsumfang groß ist und insgesamt ausführlichere Informationen über Kontobewegungen an Dritte freigegeben werden. Im Vergleich dazu dient die PISP-Rolle primär zur Initialisierung einer Zahlung und gibt somit wenige Informationen an Dritte weiter. Das Gleiche gilt auch für die PISP-Rolle, die hauptsächlich zur Deckungsprüfung im Rahmen der Anwendung von Zahlungsinstrumenten dient.

Für die nachfolgende Betrachtung werden vier Schnittstellen miteinander verglichen, die die von PSD2 vorgesehen Rollen abbilden. Hinter den angeführten APIs stehen Initiativen unterschiedlicher Länder und Interessensgruppen:

- NextGenPSD2 [3]: In einem Zusammenschluss der Deutschen Kreditwirtschaft mit Unternehmen für kartengestützten Zahlungsverkehr, hat die Berlin Group eine Schnittstellendefinition erarbeitet.
- STET [4]: Als Anbieter von Zahlungssystemlösungen für Banken wurde eine weitere Spezifikation vom französischen Unternehmen STET vorgelegt.
- SBAS [5]: Eine weitere Entwicklung einer PSD2-entsprechenden API ist der „Slovak Banking API Standard“ (SBAS), der von einer Gemeinschaft slowakischer Banken entworfen wurde.
- Priora[11]: Die vierte Umsetzung einer Schnittstellen, die die Rollen entsprechend der PSD2-Richtlinie berücksichtigt, kommt vom kanadischen FinTech-Unternehmen SaltEdge Inc.

3.1. AISP-Funktionen

Die unter der AISP-Rolle vorgesehene Reihe von Funktionalitäten ermöglicht grundsätzlich den Abruf von Kontodaten bei einem kontoführenden Zahlungsdienstleister (ASPSP). Für diesen Zweck werden mehreren Aktivitäten (Prozessabläufe), abhängig von der API, vorgesehen. Tabelle 1 zeigt eine Übersicht von Aktivitäten, die durch API-Aufrufe von AISP an ASPSP realisiert werden können. Der grundsätzliche Unterschied zwischen APIs besteht in der Granularität bzw. dem Umfang von retournierten Datenstrukturen. In allen Fällen sind die Operationen als RESTful API Endpunkte definiert. Zusätzlich zu anderen APIs, definiert NextGenPSD2 die Hyperlinks, die einem Klienten zusätzliche, kontext-basierte Informationen über weiterführende Aufrufe zur Verfügung stellen. Dies ist besonders für komplexe Operationen relevant, die mehrere API-Aufrufe für den Abschluss eines Vorgangs benötigen.

Operation	NextGenPSD2 [3]	STET [4]	SBAS [5]	Priora[12]
Liste der erreichbaren Konten			○	
Abruf der Kontodetails einer Liste zugänglicher Konten	●	●		●
Abruf des Saldos eines bestimmten Kontos	●	●		
Abruf der Details für ein bestimmtes Konto			●	
Ermitteln von Transaktionsinformationen für ein bestimmtes Konto	●	●	●	●

Tabelle 1: Übersicht von definierten Operationen per API
● Muss implementiert werden ○ Optional unterstützt

In den nächsten Kapiteln betrachten wir jede dieser Operationen separat. Besonderes Augenmerk legen wir dabei auf die in API-Antworten gegebenen Daten, da alle Aktivitäten als passive Operationen[13] vorgesehen wurden.

[11] https://priora.saltedge.com/docs#tpp
[12] https://priora.saltedge.com/docs#tpp
[13] Diese Art von HTTP-Operationen dient hauptsächlich zum Abruf von Daten. Es werden daher keine Daten aktualisiert, geändert oder gelöscht.

3.1.1. Liste der erreichbaren Konten

Dieser Endpunkt bietet die vereinfachten Informationen über alle für ein AISP zugreifbaren Konten eines PSUs an. Diese Funktionalität ist nur bei SBAS als optionaler Endpunkt vorgesehen.

Der vorliegende Endpunkt gibt nur die minimalen Daten über verfügbare Konten frei. Daten, wie das Kontensaldo, sollen per gesonderten Aufrufen ermittelt werden. So sieht der Vorgang von SBAS vor, dass die AISPs zuerst eine Abfrage über verfügbare Konten stellen, und dann die Informationen spezifisch für jedes Konto im Rahmen von gesonderten Interaktionen abfragen.

Daten	SBAS
IBAN	●
Währung	●
Konto Name (von/für BenutzerIn)	●
Produkttyp	○
Kontotyp (laut ISO20002[14])	○
BIC	●
Zeitstempel	●
Zustimmung für die Konten (Rollen)	●

Tabelle 2: Liste der erreichbaren Konten – Datenübersicht
● Muss unterstützt werden ○ Optionales Feld (ASPSP abhängig)[15]

3.1.2. Abruf der Kontodetails einer Liste zugänglicher Konten

Anders als SBAS ermöglichen andere APIs konsolidierte Abfragen der erweiterten Datensätze von allen unterstützten Konten. Der gegebene Datensatz umfasst typischerweise zusätzliche Konten- und Saldendetails. Wenn es um eine Zahlungskarte geht, bietet dieser Endpunkt typischerweise auch Daten über verbundene Konten.

Daten	NextGenPSD2	STET	Priora
Identifikator (vom ASPSP)	○	●	○
Identifikator (Service-spezifisch)			●
IBAN	○[16]		●
BBAN	○		○
PAN	○		
Verbundenes Konto		○	
Masked-PAN	○		
MSISDN	○		
Währung	●	●	●
Konto Name (von/für BenutzerIn)	○	●	●
Produkttyp	○		
Kontotyp	○	●	●
BIC	○		○
Salden	○	○	●
Weiterführende Links	○	○	
Zusätzliche Details		○	○
Zahlungskonto			●
Zeitstempel			●
Anwendung des Kontos[17]		○	

[14] ISO 20022 Universal financial industry message scheme - External Code Sets, verfügbar über https://www.iso20022.org/external_code_list.page
[15] Diese Notation gilt auch bei weiteren Übersichten
[16] Einer von mehreren möglichen Identifikatoren (IBAN, BBAN, PAN) muss einbezogen werden
[17] Organisatorisch oder privat

Status des Benutzers[18]		○	
Letzte Bewegung		○	

Tabelle 3: Kontodetails von zugänglicher Konten – Datenübersicht

NextGenPSD2 und STET APIs setzen auf externe Standardisierungsinitiativen für die Bezeichnung von oft verwendeten und ausgetauschten Informationsfeldern. So werden etwa die im Rahmen von ISO 20022 definierten Datenstrukturen bei beiden APIs eingesetzt. Im Gegensatz dazu setzt Priora auf individuell definierte Bezeichnungen bzw. Formate, was die Interoperabilität, sowie die Integration mit anderen Umgebungen potentiell einschränken kann.

Bezüglich Anpassungsmöglichkeiten bei Anfragen bietet NextGenPSD2 die höchste Flexibilität. Priora unterstützt wenige optionale Parameter zur individuellen Gestaltung von API-Antworten.

3.1.3. Abruf des Saldos eines bestimmten Kontos

Zusätzlich zur Abfrage von Kontodetails bieten zwei APIs die Möglichkeit Salden bei bestimmten Konten gesondert aufzulisten. Beide API-Spezifikationen unterscheiden zwischen mehreren Saldenarten und beruhen dabei auf in ISO 20022[19] definierten Bezeichnungen. In Anbetracht des Umfangs unterstützter Kategorien von Saldenarten bieten die APIs unterschiedliche Detailgrade an. Diese gliedern STET und NextGenPSD2 in vier bzw. fünf Kategorien.

Daten	NextGenPSD2	STET
Name (für die Saldo-Position)		●
Wert	●	●
Währung		○
Status		●
Letzte Bewegung		○
Konto Name (von/für BenutzerIn)		●
Saldenart	●	●
Zeitstempel des Berichts		○
Zeitstempel letzter Bewegung	○	

Tabelle 4: Details einer Saldoübersicht

3.1.4. Abruf der Details für ein bestimmtes Konto

Slovak Banking API und Priora stellen die Kontogrunddaten sowie die Saldodaten nur durch einen Endpunkt zur Verfügung. Die restlichen APIs ermöglichen getrennte Saldo- und Kontoinformations-aufrufe. Die folgende Tabelle zeigt die Übersicht von Daten, die die SBAS API für ein bestimmtes Konto liefert.

Daten	SBAS
Konto Name (von/für BenutzerIn)	●
Währung	●
Produkttyp	○
Kontotyp (laut ISO20002[20])	○
Zeitstempel	●
Saldenart	●
Wert	●
Währung (für Position)	●
Statusindikator	●

Tabelle 5: Datenübersicht eines Kontos

[18] Kontoinhaber, Mitinhaber oder Anwalt
[19] ISO 20022 Universal financial industry message scheme – Balance Type Code
[20] ISO 20022 Universal financial industry message scheme - External Code Sets

3.1.5. Ermitteln von Transaktionsinformationen für ein bestimmtes Konto

Eine der wichtigsten Funktionalitäten von ASPSPs ist die Freigabe von Kontobewegungen eines Kontos an Drittparteien. Eine rechtsvertragliche Vereinbarung mit KontoinhaberInnen vorausgesetzt, können AISPs, basierend auf diesen Daten, entsprechende Schritte zur Aufbereitung und Verarbeitung vornehmen. Diese Möglichkeit wird von allen analysierten APIs in unterschiedlichem Ausmaß, wie in Tabelle 6 dargestellt, implementiert.

Daten	NextGenPSD2	STET	SBAS	Priora
Identifikator (Transaktion)	○	○	○	●
Identifikator (Händler)	○		○	●
Identifikator (Mandate)	○		●	
Identifikator (Benutzerkonto)				●
Gläubiger-Identifikationsnummer	○		○	
Buchungsablauf		●	●	
Buchungsdatum	○	●	●	
Valuta	○		●	●
Wert	●	●	●	●
Währung		○	●	●
Name des Gläubigers	○		○	
Gläubigerkonto	○		○	
Debitorenkonto	○		○	
Name des Debitors	○		○	
Buchungsstatus		●	●	
Stornotransaktion			○	
Zahlungsinformationen	○	●	○	●
Zweck	○		○	
Transaktionscode	○		○	
Gebührenreferenz			○	●
Zahlungsmittelreferenz			○	
Währungsumrechnungskurs			○	
Name des Handelspartners			○	
Kreditoren Bankleitzahl			○	
Debitoren Bankleitzahl			○	
Zeitstempel des Berichts				●

Tabelle 6: Übersicht von Kontobewegungen

NextGenPSD2 und SBAS bieten die am besten entwickelten und detailliertesten Datenbestände hinsichtlich Darstellungsgranularität und Konfigurationsmöglichkeiten. Die Granularität der Darstellung sowie der Umfang freigegebener Daten sind in Anfragen anpassbar. Priora unterstützt nur die statische und beschränkte Darstellung von Kontobewegungsdaten.

3.2. PISP-Funktionen

Der PISP-Funktionsumfang konzentriert sich auf die Steuerung von Zahlungsanforderungen. Dazu zählen insbesondere die Erstellung, der Abruf sowie die Bestätigung oder Stornierung von bereits initiierten Zahlungen. Die Übersicht der Anwendungsfälle untersuchter APIs findet sich in Tabelle 7.

Bei näherer Analyse der angeführten Anwendungsfälle zeigen sich wesentliche Unterschiede zwischen den APIs. So verwenden STET, NextGenPSD2 und SBAS ISO-20022 XML-basierte Strukturen für die Beschreibung von Zahlungsdaten bei der Zahlungsinitiierung. Konkret integriert STET *pain.013 – CreditorPaymentActivationRequest* [6] die Zahlungsanforderung jener Nachricht, die typischerweise von einem Gläubiger verwendet wird, um die Bewegung von Geldmitteln von einem Schuldnerkonto zu einem Gläubiger zu beantragen. Die erste Fassung dieser Struktur wurde bereits im Jahr 2010 seitens Payments SEG akzeptiert. Die aktuelle Fassung *pain.013.001.06* wurde im Jahr 2017 veröffentlicht.

Operation	NextGenPSD2	STET	SBAS	Priora
Zahlungsinitiierung im Auftrag eines Händlers	●	●	●	●
Abruf einer bereits gebuchten Zahlungsanforderung		●		●
Abruf des Status einer Zahlungsanforderung		●	●	
Bestätigung einer Zahlungsanforderung		●	●	●
Zahlungsinitiierung für Massenzahlungen und Mehrfachzahlungen	●			
Initiierung einer zukünftigen Zahlung	●			
Initiierung von Daueraufträgen für wiederkehrende/periodische Zahlungen	●			
e-Commerce-Zahlungsinitialisierung[21]	●	●	●	
Stornierung der Zahlung				●

Tabelle 7: Zusammenfassung von PISP-Anwendungsfällen

Die Strukturierung von Parametern in HTTP-Anfragen zweier weiterer APIs - SBAS und NextGenPSD2 – basiert auf *pain.001.001.03 – CustomerCreditTransferInitiation* Nachrichten [7]. Diese Nachricht wird typischerweise an den Zahlungsdienst eines Schuldners gesendet, um die Bewegung vom Konto des Schuldners zu einem Gläubiger zu beantragen. Die aktuelle Fassung dieser Struktur (*pain.001.001.08*) wurde im Jahr 2018 veröffentlicht. Im Gegensatz zu anderen Lösungen definiert Priora ein eigenes JSON-Format.

Einige fortgeschrittene Funktionen, wie z.B. Massenzahlungen, zukünftige Zahlungen, sowie die Festlegung von periodischen Zahlungen definiert nur NextGenPSD2. Eine schrittweise oder workflow-basierte Manipulation von Zahlungsanforderungen ist grundsätzlich bei den anderen drei APIs vorgesehen. Die Möglichkeit einer Zahlungsstornierung ist jedoch nur bei Priora vorgesehen.

3.3. PIISP-Funktionalitäten

Die letzte unter PSD2 definierte Funktion ermöglicht die Deckungsprüfung eines bestimmten Zahlungskontos. Solche Abläufe finden hauptsächlich bei PIISP bzw. Zahlungsdienstleistern und den Ausgebern von Zahlungsinstrumenten Anwendung. Letztere nutzen typischerweise den API-Aufruf, um die Verwendung eines Zahlungsinstruments im Rahmen einer Transaktion erlauben oder ablehnen zu können. Für die Transaktion entscheidend ist dabei die Kontendeckung.

Unter der Prüfung der Kontodeckung wird ein Aufruf (Aktivität) verstanden, der die Verfügbarkeit eines bestimmten Betrags auf dem Konto des PSU prüft. In allen analysierten Fällen wird dieser Aufruf durch einen booleschen Wert beantwortet, der nur die Information über die Verfügbarkeit des angefragten Finanzmittels signalisiert. Jedoch unterscheiden sich die APIs durch die Menge der im Aufruf übergebenen Daten. Tabelle 8 zeigt eine Übersicht von diesen Daten für jede der analysierten APIs.

Basierend auf dem in der Tabelle angeführten Vergleich, gehören zu den wichtigsten Informationen für diesen Aufruf der Kontoidentifikator (bei ASPSP, intern sichtbar) beziehungsweise IBAN des Kontos (extern sichtbar). APIs von SBAS und NextGenPSD2 können zusätzlich dazu Daten über die Drittpartei (Händler) sowie die Kartennummer (verbundenes Zahlungsinstrument) übertragen. Die Nutzbarkeit dieser Daten sowie deren Zweck für eine Kontodeckungsprüfung sind derzeit unklar.

Operation	NextGenPSD2	STET	SBAS	Priora
Kontodeckung	●	●	●	●
Daten				
Identifikator (generiert von PIISP)			●	
Identifikator (Drittpartei bzw. Händler)			○	
Kontoidentifikator (bei ASPSP)		●		
Typ des Kontoidentifikators		●		

[21] Unter eCommerce Zahlungen werden primär die sofortigen, nicht umkehrbaren Zahlungen verstanden.

MCC-Kode (von Händler)			○	
Datum und Uhrzeit			○	
IBAN	●		●	
Betrag	●	●	●	●
Währung		●	●	●
Name von Drittpartei/Händler	○		○	
Anschrift und Staat der Drittpartei			○	
Kartennummer	○		○	
Name einer KarteninhaberIn			○	

Tabelle 8: Übersicht von Aktivitäten und Daten vorgesehen für die PIISP-Rolle in PSD2 APIs

3.4. Autorisierungsmechanismen in APIs

Die für verschiedene Rollen zur Verfügung gestellten Operationen müssen vorab von KontoinhaberInnen genehmigt werden. Dies soll durch Anwendung entsprechender Mechanismen zur Autorisierung erfolgen, was in den Spezifikationsdokumenten für jeden Standardisierungsansatz bzw. API näher geregelt ist.

Alle analysierten APIs nutzen OAuth 2 als ein potentiell anwendbares Framework für die Autorisierung von Kontozugängen. SBAS, STET und Priora betrachten dabei OAuth 2 als primäre oder einzige Möglichkeit der Autorisierung. NextGenPSD2 jedoch definiert OAuth 2 als eine mögliche, alternative Variante, und zwar nur für die AISP und PISP Rollen.

↓Rolle \ API→	**SBAS**	**NextGenPSD2**	**STET**	**Priora**
AISP	○	○	○	○
PIISP	□	-	○	○
PISP	○	○	□	○

Tabelle 9: Übersicht von OAuth 2 Grant-Varianten und TPP-Rollen in APIs

○ Authorization code grant flow □ Client credentials grant flow

OAuth 2 sieht mehrere Grant Flow Typen vor, um die verschiedenen Konfigurationen bzw. anwendungsorientierten Anforderungen optimal abdecken zu können. Grant-Flow definiert die Prozesse sowie die konkreten Schritte, die eine Zustimmungserklärung des Ressourceninhabers und die Übergabe von *Access Tokens* ermöglichen. Für die Anwendung mit NextGenPSD2 und Priora sind zwei Grant-Typen von Relevanz: *Authorization Code Grant (ACG)* und *Client Credentials Code Grant (CCG).* Dabei steht der erste Grant-Typ für die Autorisierung einer Aktivität mit drei Akteuren (PSU, TPP[22], ASPSP), die einem PSU im Rahmen eines Online-Prozesses ermöglicht, den Zugang eines TPP auf sein bei einem ASPSP eröffnetes Konto zu bewilligen.

Die Verwendung des zweiten Grant-Typ (*CCG*) schließt BenutzerInnen (PSU) im Autorisierungsprozess nicht ein. So sind bei CCG nur zwei Parteien berücksichtigt, nämlich TPP und ASPSP. Daher ist anzunehmen, dass die konkrete Autorisierung beim CCG in Rahmen von sog. *Out-of-Band-Verfahren*[23] erfolgen soll. Mögliche Alternativen in diesem Fall wären *ex ante* und *ex post* Autorisierung. Die ex-ante Autorisierung ist vom PSU vorzeitig beim ASPSP einzutragen und für alle weiteren Zugriffe gültig. Andererseits kann der PSU bei ex-post Autorisierung die Aktivitäten erst nachdem die entsprechende Anfrage beim ASPSP seitens TPP gestellt und angenommen wurde, erlauben. Abhängig von der Implementierung kann diese Autorisierung auch für alle weiteren Zugriffe gelten.

[22] TPP – Third Party Payment Provider ist eine abstrakte Entität, die mehreren Subjekten entspricht, unter anderem auch AISP, PISP oder PIISP.

[23] In diesem Kontext bezeichnet Out-of-Band die Autorisierung, die durch einen getrennten Kanal bzw. getrenntes Medium erfolgt und somit nicht durch die Standardisierung fest definiert wird. Freie Implementierungen ohne konkrete Richtlinien können unter Umständen zu Inkompatibilitäten und weiteren Hindernissen im Sinne eines *Informationssilos* führen.

Prinzipiell berücksichtigen alle analysierten APIs die Verwendung von *Authorization Code Grant (ACG)* und *Client Credentials Code Grant (CCG).* Tabelle 9 zeigt eine Aufstellung der Verwendung dieser zwei Grants unter verschiedenen TPP-Rollen in den jeweiligen APIs.

Das Spezifikationsdokument von SBAS, Kapitel 4.4.1 [5] sieht vor, dass ASPSPs im Fall von AISP- und PISP-Rollen selbst feststellen können, welche zwei OAuth 2 Grant Flows sie unterstützen bzw. anwenden wollen. Die Dokumentation in Kapitel 5 bzw. 6 [5] sieht für die Verwendung bei beiden Rollen nur den *Authorization Code Grant* vor*.* Für die PIISP-Rolle definiert die SBAS-Spezifikation nur den *Client Credentials Code Grant*. Die Zustimmung des PSUs soll dabei extern und getrennt erfolgen (Kapitel 7 in [5]), wobei die konkreten Prozesse und Methoden vom ASPSP selbst definiert werden müssen. In Bezug auf PIISP-Datenzugriffe sieht NextGenPSD2 kein OAuth 2-Verfahren vor. Es bleibt beim Einsatz dieser API somit nur die Möglichkeit einer expliziten *ex ante* Zustimmungserklärung, die durch ein *Out-of-Band*-Verfahren außerhalb der standardisierten Ansätze zu erteilen ist. Die restlichen APIs stellen das ACG-basierte Verfahren für die Online-Autorisierung zu Verfügung. Die Zustimmung von BenutzerInnen soll im Rahmen der Registrierung der PSU-TPP Vereinbarung erfolgen und die weiteren Zugriffe des TPPs regeln.

Schließlich verwenden alle APIs bis auf STET ACG-basierte Workflows für die Erteilung von Autorisierungen von PISP-Funktionen. In diesem Fall erfolgt die Autorisierung außerhalb von OAuth 2-Prozessen durch die Anwendung von SCA[24] im Rahmen der ASPSP Infrastruktur. Die gesamte Zahlung wird als eine mehrstufige Transaktion betrachtet, die zuerst gespeichert, genehmigt und dann ausgeführt werden soll. Nach der Zahlungsinitialisierung wird der PSU an den ASPSP weitergeleitet, um die angeforderte Zahlung zu bestätigen. Die STET API prüft im Hintergrund den Status solcher Zahlungsanforderung beim ASPSP und führt die Zahlung aus, sofern sie vom PSU genehmigt wird. Dafür reicht wiederum auch CCG aus, da die Statusprüfung, sowie die Ausführung der Zahlung getrennt im Hintergrund mittels Pooling erfolgen.

3.5. Zugriffssteuerung in APIs

Nachfolgend werden die wichtigsten Aspekte der Zugriffskontrolle und ihre Anwendung erörtert.

3.5.1. Technische Basis zur Darstellung und Bearbeitung von Autorisierungen

Die Autorisierung mittels OAuth 2 in APIs beruht auf *Bearer Access Tokens*, deren Umfang bzw. erlaubte Operationen durch die im Rahmen von Autorisierungsverfahren zugewiesenen *Scopes* [8] beschrieben werden. Die Scopes selbst sind als einfache und atomische Datenstrukturen vorgesehen, deren Bedeutung implizit und unstrukturiert durch die dienst-spezifische Dokumentation definiert wird.

Generell sind Scopes als statische Strukturen definiert, die auch mehrere Operationen (implizit) umfassen können [9]. Dieser Umstand spiegelt sich auch in der technischen Umsetzung wider. Erstens ist der Umfang von zugrundeliegenden Autorisierungseigenschaften einseitig durch den Dienstanbieter festgestellt und geregelt. Das Interesse, sowie die Anforderungen von BenutzerInnen, werden in diesem Prozess nicht berücksichtigt. Die einsetzbaren sicherheitsrelevanten Schutzfunktionen für Daten sind nur in begrenztem Ausmaß konfigurierbar und können lediglich vom Dienstanbieter festgelegt werden. Die damit verbundenen Vorteile für Anbieter resultieren in einer vergleichsweise einfachen sowie vom Aufwand her günstigen Implementierung sicherheitsrelevanter Funktionalität, die im Wettbewerb unter Umständen auch weitere wirtschaftliche und wettbewerbsrelevanten Vorteile nach sich ziehen könnte.

Ein Scope beschreibt Autorisierungen und dabei gewährte Berechtigungen in losem, unstrukturiertem Text. Die Kopplung an den Dienst bzw. das System des Dienstanbieters geschieht individuell ohne Rückgriffe auf etablierte Praktiken. Dies erschwert die maschinelle Lesbarkeit von Scopes, das automatisierte Auslesen von Informationen aus dieser Darstellungen sowie das Referenzieren von Autorisierungen und Ressourcen. Diese Hürde ist besonders bei der Anwendung in heterogenen Umgebungen bemerkbar, indem Interoperabilität zwischen verschiedenen Systemen wesentlich erschwert wird. Dies wiederum wirkt sich potentiell negativ auf die Umsetzung von

[24] Strong Customer Authentication

Sicherheitsrichtlinien aus, die die Integration der Daten in fremden Umgebungen betrifft. Eine detaillierte Auswertung von weiteren Auswirkungen der Designentscheidungen hinter Scopes ist in [9] genauer dargestellt. Eine weitere Studie [10] beschäftigt sich mit der praktischen Implementierung von Scopes bei diversen Online-Diensten. Dabei wurde eine generelle Tendenz zur inkonsistenten und sicherheitswidrigen Anwendung von Scopes bei einer großen Anzahl von Dienstanbietern festgestellt (es handelt sich um eine Beobachtung zu in Studie [10] analysierten Lösungen, was nicht bedeutet, dass dies auch in PSD2 APIs ähnlich zu erwarten ist).

3.5.2. Anwendung in APIs

Die analysierten APIs verbinden die Scopes hauptsächlich durch die von XS2A vorgesehenen Rollen. Typischerweise werden dabei Scopes wie „*AISP*“ oder „*PISP*“ eingesetzt, die alle verfügbaren Operationen für jede dazugehörige Rolle umfassen. So unterstützt der Scope „AISP“ alle für AISPs notwendige Operationen und gibt alle dazugehörigen Ressourcen frei. Das unter diesem Scope erteilte Access Token ermöglicht somit den Aufruf von allen AISP-Funktionen und den Zugriff auf alle darunter spezifizierten Daten ohne weitere Einschränkungen. Dies zeigt sich beispielsweise in den in Kapitel 3.1 dargestellten Daten: ein typischer „*AISP*“ Scope kann nicht die genaue Spezifizierung von einzelnen ausführbaren Operationen sowie zugriffbaren Felder unterstützen[25].

Eine feinere Definition von Rollen und Autorisierungen wird bei NextGenPSD2 umgesetzt. In dieser API referenzieren die Scopes nicht nur die tatsächliche Rolle, sondern auch die konkrete Ressource bzw. erteilte Zustimmung. Die entsprechende Struktur wird durch das Format „*PIS:<ZahlungsID>*“ bzw. „*AIS:<ZustimmungsID>*“ repräsentiert. Diese Zustimmung ist durch eine komplexe Datenstruktur dargestellt und kann mehrere, für den Zugriff relevante, Eigenschaften beinhalten. So sind beispielweise mehrere Operationen enthalten, die auf die Gültigkeitsdauer sowie die Auftrittshäufigkeit konkreter Daten verweisen können. Zwei Beispiele dazu sind in den folgenden Abbildungen dargestellt:

```
{"access":
  {"available-accounts": "all-accounts"},
  "recurringIndicator": "false",
  "validUntil": "2018-08-06",
  "frequencyPerDay": "1"
}
```

Abbildung 3: Zustimmung, Zugriff auf alle Konten

```
{"access":
  {"balances": [], "transactions": []},
  "recurringIndicator": "true",
  "validUntil": "2018-11-01",
  "frequencyPerDay": "4"
}
```

Abbildung 4: Zustimmung, partielle Zugriff auf Ressourcen

Abbildung 3 zeigt die Struktur zur Beschreibung einer Zustimmung für einen einmaligen Zugriff. Diese Zustimmung erlaubt den Zugriff auf die Daten aller verfügbaren Konten, längstens bis zum 6. Juni 2018. In diesem Beispiel wird somit eine konkrete Operation ohne weitere Einschränkungen im Bezug auf konkrete Daten und weiteren Funktionalitäten erlaubt. In Abbildung 4 wird ein Entwurf einer anderen Zustimmung, die mehrfache Zugriffe auf einzelne konkrete Operationen über bestimmten Konten erlaubt, illustriert. Diese Datenstruktur ist im Rahmen einer Zustimmungserklärung bei einem ASPSP zu ergänzen, und zwar entsprechend den Wünschen des PSUs. Im Vergleich mit dem vorigem Beispiel weist diese Struktur eine höhere Granularität auf, da sie mehrere zur AISP-Rolle zugehörige Operationen referenziert und für jede Operationen weitere Einschränkungen erlaubt.

[25] Beispiele dafür wären die Einschränkungen in Bezug auf die Zeit, Anzahl von Zugriffen, Klienten oder Netzwerken sowie die Anordnung von umformenden Operationen wie Data Masking oder Verschlüsselung.

Im Vergleich mit anderen Ansätzen ermöglicht dieser eine flexiblere und feinere Definition von Autorisierungen. Der Zugriff kann daher beispielweise auf ein bestimmtes Konto, die genaue Operation sowie durch die konkrete Gültigkeitsdauer und der genauen Anzahl von täglichen Aufrufen beschränkt werden. Die Struktur dieser Zustimmung ist jedoch fix vorgegeben und kann die weitere Gliederung auf die Datenfelder, sowie die Anwendung von dynamischen und umformenden Operationen, nicht darstellen. Im Vergleich mit anderen Lösungen hebt sich NextGenPSD2 also durch eine besonders detaillierte Zugriffssteuerung hervor, die auch auf einzelne Ressourcen angewendet werden kann.

4. Zusammenfassung

Durch die neue PSD2-Richtlinie werden Prozesse, wie der Datenaustausch zwischen Finanzakteuren und die Zahlungsinitiierung, unter Betrachtung jüngster technologischer und geschäftlicher Entwicklungen neu gefasst. In dieser Studie wurden die technischen Aspekte von Änderungen analysiert, die durch die PSD2-Richtlinie eingeführt wurden. Besonderes Augenmerk legten wir daher auf die vier Finanz-APIs, die als PSD2-konforme Lösungen gelten. Dabei wurde auch die Darstellung von repräsentativen Zahlungstransaktionen in bestehenden eBanking-Oberflächen einbezogen. Die dabei untersuchten Daten dienen als gemeinsame Grundlage und werden auch per APIs an Dritte, im Rahmen von PSD2-Vorgängen, weitergegeben.

Im Zuge der Studie hat sich gezeigt, dass alle untersuchten APIs auf dem REST-Standard aufbauen und das etablierte OAuth2-Framework für die Verwaltung von Autorisierungen einsetzen. Anhand mehrerer Beispiele, sowie der detaillierteren Struktur und durch APIs freigegebenen Datenbeständen, wurden diese APIs untersucht, sowie die ihnen zugrundeliegenden technologischen Ansätze zur Autorisierung bzw. Sharing-Zustimmung inspiziert.

Unsere Erkenntnisse zeigen auf, dass die in den untersuchten APIs gewählten Lösung zur Verwaltung von Autorisierungen eine Steuerung oft nur auf grober Ebene vorsehen. Bestehende Lösungen, und insbesondere die Realisierung über OAuth2 erfüllen aber die Anforderungen entsprechend der PSD2-Richtlinie. Wie durch eine andere Studie belegt, wäre eine feinere Integration mit Zugriffsbeschränkung auf einzelne Ressourcen sinnvoll [10]. Als Einstiegspunkt für einen Ansatz zur Ermächtigung von BenutzerInnen und zur genaueren Kontrolle des Datenaustauschs kann die in [9] eingeführte Lösung in Erwägung gezogen werden.

Referenzen

[1] European Commision, „Aufbau eines gemeinsamen europäischen Datenraums," 2018.

[2] „RICHTLINIE (EU) 2015/2366 DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 25. November 2015 über Zahlungsdienste im Binnenmarkt," 2015.

[3] The Berlin Group, „NextGenPSD2 XS2A Framework - Operational Rules v1.0," 2018.

[4] R. Herve, „STET PSD2 API v.1.2.3," 2017.

[5] Slovak banking association, „Slovak Banking API Standard v1.0," 2017.

[6] ISO, „ISO 20022 - Creditor Payment Activation Request," 2010.

[7] ISO, „ISO 20022 - Payments - Maintenance 2009," 2009.

[8] D. Hardt, „RFC 6749 The OAuth 2. 0 Authorization Framework. Internet Engineering Task Force (IETF)," 2012.

[9] B. Suzic, A. Reiter und A. Marsalek, „Structuring the Scope: Enabling adaptive and multilateral authorization management," in *2017 IEEE Conference on Communications and Network Security (CNS)*, 2017.

[10] B. Suzic, B. Prünster und D. Ziegler, „On The Structure and Authorization Management of RESTful Web Services," in *Proceedings of The 33rd ACM/SIGAPP Symposium on Applied Computing (ACM SAC)*, 2018.

[11] The Berlin Group, „NextGenPSD2 XS2A Framework - Implementation Guidelines v1.0," 2018.

[12] SaltEdge, „Priora Client API,“ 2018. [Online]. Available: https://priora.saltedge.com/docs.