

Instant Payment Integration at the Point of Sale

Bachelorarbeit, Studiengang Informatik
OST – Ostschweizer Fachhochschule
Campus Rapperswil-Jona

Frühlingssemester 2025

Autor
Betreuer

Jan Meier
Mirko Stocker

Inhaltsverzeichnis

1. Einleitung	3
1.1 Ausgangslage	3
1.2 Ziel der Arbeit	3
1.3 Anforderungen	3
2. Analyse, Konkrete Anforderungen	4
2.1 Instant Payment (IP)	4
2.2 Open Banking	5
2.3 Varianten: Vier Parteien Modell	5
2.4 Variantenentscheid	9
2.5 Empfängerbank	9
2.6 Unterschied zwischen Instant Payment und Twint	10
2.7 Technischer Ablauf einer Zahlung	11
2.8 Swiss QR Code	12
2.9 Statuskonzept	13
2.10 Überprüfung von NOK-Zahlungen	14
2.11 Mindestanforderungen an Open Banking Schnittstelle	14
2.12 Technologiewahl	16
2.13 Funktionale Anforderungen (FA)	17
2.14 Nicht Funktionale Anforderungen (NFA)	18
3. Lösungskonzept	19
3.1 Datenbankschema	19
3.2 C4 Diagramm	20
3.3 Codearchitektur	22
3.4 Open Banking API Schnittstelle	23
4. Umsetzung	26
4.1 Verwendete Tools und Libraries	26
4.2 Grundlagen	28
4.3 Händlerportal	29
4.4 Payment Session	29
4.5 JWT-Tokens pro Payment Session	37
4.6 Transaktionsabgleich mit Open Banking	38
4.7 Checkout Seite	39
4.8 Verschiedene Auszahlungsarten	40

4.9	Error Handling Backend	42
4.10	Testing	42
5.	Ergebnisdiskussion mit Ausblick.....	49
5.1	Befragung zu Schweizer Zahlungsmethoden	50
5.2	Wo sieht die SIX Instant Payments als Zahlungsmittel	52
5.3	Kann Twint nicht einfach auf Instant Payment umsteigen?	52
5.4	Gegenwart und Zukunft der Zahlungsmethoden im Onlinehandel	52
5.5	Optimale Voraussetzungen der Open Banking Schnittstelle	53
5.6	Problemstellungen	54
5.7	Variantenvergleich mit/ohne Zwischenkonto	58
5.8	Mögliche Weiterentwicklung – Open Banking auf Kundenseite	59
5.9	Fazit	60
6.	Glossar	61
7.	Abbildungsverzeichnis	62
8.	Tabellenverzeichnis	63
9.	Anhänge	64

Abstract

Diese Bachelorarbeit befasst sich mit der Konzeption und prototypischen Umsetzung einer Zahlungslösung, die Instant Payments (IP) und Open Banking für den Einsatz im Onlinehandel und am physischen Point of Sale nutzbar macht. Ziel war es, die technische Machbarkeit eines Systems zu demonstrieren, das IP-Zahlungen in Echtzeit validiert und damit eine sofortige Warenfreigabe ermöglicht.

Nach einer Analyse verschiedener technischer Umsetzungsoptionen wurde ein Ansatz gewählt, bei dem die Zahlungsplattform direkten Zugriff auf das Empfängerkonto hat, um Zahlungseingänge in Echtzeit zu überprüfen. Hierfür wurde ein System mit einem .NET-Backend, einem Next.JS-basierten Händlerportal und einer Angular-Checkout-Seite entwickelt. Die Datenhaltung erfolgt mittels PostgreSQL und Redis, während eine Testbankenumgebung der Innofactory AG für die Simulation von IP-Zahlungen und den Abruf von Transaktionsdaten über eine Open-Banking-Schnittstelle eingesetzt wurde.

Die entwickelte Lösung demonstriert die technische Machbarkeit einer solchen Plattform und identifiziert zudem auch bestehende Hürden, darunter Gebühren für IP-Zahlungen, die noch geringe Verbreitung von IP-versendenden Banken und Usability-Aspekte von Mobile Banking Apps bei der Erfassung von IP-Zahlungen. Interviews mit Fachpersonen des Banken- und E-Commerce-Sektors sowie eine begleitende Umfrage gaben zusätzlich Aufschluss über Marktpotenziale, Anforderungen und Akzeptanzkriterien.

Management Summary

Ausgangslage

In der Schweiz existieren zahlreiche Zahlungslösungen, die überwiegend von Drittanbietern betrieben werden. Banken müssen erhebliche finanzielle Mittel für deren Integration aufbringen und Händler zahlen pro Transaktion Gebühren an den Betreiber der Zahlungsplattform. Die Einführung von Instant Payments (IP) eröffnet das Potenzial, Zahlungsprozesse effizienter, kostengünstiger und standardisierter abzuwickeln. Ziel dieser Arbeit ist die Entwicklung einer Zahlungslösung, die auf bereits existierenden Banktechnologien wie Instant Payment und Open Banking basiert.

Vorgehen & Technologien

Zunächst wurden verschiedene Varianten für die technische Umsetzung analysiert, um den optimalen Zahlungsprozess zu identifizieren. Nach einer umfassenden Evaluation fiel die Wahl auf ein Verfahren, bei welchem die Zahlungsplattform Zugriff auf das Empfängerkonto erhält und so den Eingang von neuen Zahlungen prüfen kann. Für die Händlerseite wurde ein Portal entwickelt, das eine übersichtliche Verwaltung aller Zahlungsvorgänge ermöglicht. Kunden interagieren über eine Checkout-Seite, auf der Zahlungen einfach durchgeführt werden können. Mithilfe einer Testbankenumgebung der Innofactory AG lassen sich Bankkonten erstellen, IP-Zahlungen simulieren und Transaktionsdaten über eine Open-Banking-Schnittstelle abrufen. Technologisch setzt das entwickelte System auf ein .NET-Backend zur Abwicklung der Zahlungsprozesse, Next.js für das Händlerportal sowie Angular für die Checkout-Seite. PostgreSQL übernimmt die Datenhaltung und Redis sorgt für Caching und Messaging. Ergänzend zur technischen Umsetzung wurden Interviews mit Fachpersonen des Banken- und E-Commerce-Sektors sowie eine Umfrage durchgeführt, um Chancen, Anforderungen und Anreize für neue Zahlungslösungen zu analysieren.

Ergebnisse & Ausblick

Die entwickelte Lösung zeigt, dass eine Zahlungsplattform mithilfe der zur Verfügung stehenden Technologien IP und Open Banking aus technischer Sicht realisiert werden kann. In der entwickelten Testumgebung ist es einem Händler möglich Zahlungen via Webshop, am Point of Sale und an Offline-Kassen, wie beispielsweise in einem Hofladen, mithilfe von IP einzukassieren. Die Zahlungseingänge werden während dem Bezahlprozess in Echtzeit validiert und Waren könnten dadurch vom Händler direkt freigegeben werden. Neben der Technischen Machbarkeit gibt es jedoch noch einige Hürden, welche eine solche Zahlungsplattform aktuell unattraktiv machen. Dazu zählen unter anderem Gebühren für IP-Zahlungen, die fehlende Verbreitung von Banken, die IP-Zahlungen versenden können und die Usability von Mobile Banking Apps bei der Erfassung von IP-Zahlungen.

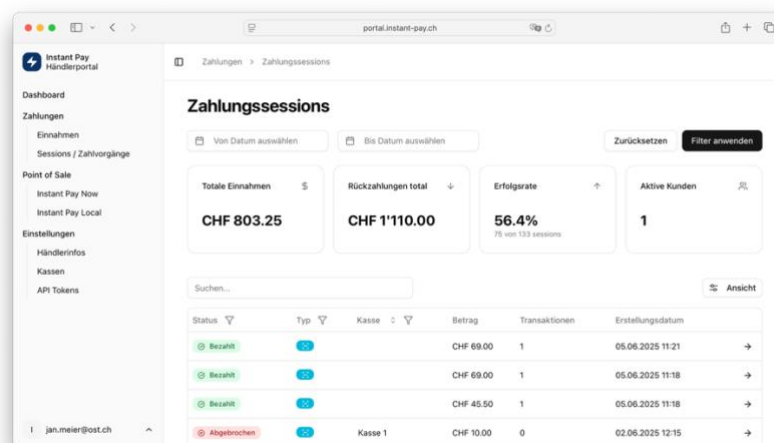


Abbildung 1 Übersicht ausgeführter Zahlungen im Händlerportal

1. Einleitung

1.1 Ausgangslage

In der Schweizer Zahlungslandschaft gibt es aktuell einige Zahlungslösungen, welche im Besitz von verschiedenen Firmen sind. Herr und Frau Schweizer bezahlen mit Bargeld, Debitkarten, Kreditkarten, Prepaid Karten, Mobile Payment Systemen (Apple Pay¹, Google Pay², Samsung Pay³, Swatch Pay⁴, Garmin Pay⁵, Fitbit Pay⁶) oder mit Twint⁷. Alle diese Lösungen haben zwei Dinge gemeinsam: keine der Lösungen ist im Besitz einer Bank und Händler müssen pro Transaktion eine Gebühr bezahlen.

1.2 Ziel der Arbeit

Ziel dieser Arbeit ist die Entwicklung einer Zahlungslösung, die unter Nutzung bestehender, bereits von Banken bereitgestellter Technologien implementiert werden soll. Die Software soll Instant Payment⁸ für den Versand von Zahlungen und Open Banking⁹ für die Abfrage des Zahlungsstatus verwenden. Für diese Software wird eine Partnerbank benötigt, welche solche Funktionalität zur Verfügung stellen kann. Falls keine Partnerbank gefunden wird, sollen Mindestanforderungen an die benötigten Schnittstellen zu Banken definiert werden und die Software soll zur Demonstration mit gemockten Schnittstellen arbeiten.

1.3 Anforderungen

An diese Arbeit bestehen folgende Anforderungen:

- Varianten evaluieren, um ein Zahlungstool zu entwickeln mithilfe der vorhandenen Technologien Instant Payment und Open Banking
- Die Entwicklung einer Software, mit welcher Zahlungen am Point of Sale (z.B. Onlineshop) durchgeführt werden können (analog zu Twint QR Code).
- Die Software soll Zahlungen zwischen einem Händler und einem Zahler durch Instant Payment ermöglichen.
- Eine Open Banking Schnittstelle soll zur Abfrage von eingehenden Zahlungen auf einem Bankkonto verwendet werden.
- Es soll eine Partnerbank gefunden werden, um das Zahlungstool in einer Sandbox-Umgebung testen zu können.
- Falls keine Partnerbank gefunden wird, wird ein Konzept ausgearbeitet, welches beschreibt, welche technischen Voraussetzungen an eine Open Banking Schnittstelle nötig sind, um einen Zahlungsdienst über Instant Payment anbieten zu können. Ein Teil der technischen Umsetzung erfolgt in diesem Fall mithilfe von Mock-Testdaten.
- Das Augenmerk soll auf die technische Umsetzung gelegt und nicht durch mögliche regulatorische Vorgaben beschränkt werden.
- Es soll (sofern verfügbar) auf bestehende Open Source Technologien gesetzt werden und keine Kopie von bestehenden Tools in Eigenregie erstellt werden.

¹ <https://www.apple.com/ch/de/apple-pay/>

² https://pay.google.com/intl/de_ch/about/

³ <https://www.samsung.com/ch/samsung-pay>

⁴ <https://www.swatch.com/de-ch/swatch-pay/how-it-works.html>

⁵ <https://www.garmin.com/de-CH/garmin-pay/>

⁶ <https://www.fitbit.com/sg/fitbit-pay>

⁷ <https://www.moneyland.ch/de/zahlungsmittel-schweizer-vergleich>

⁸ <https://www.six-group.com/de/blog/swiss-instant-payments.html>

⁹ <https://www.swissbanking.ch/de/themen/digitalisierung-innovation-cyber-security/open-banking>

2. Analyse, Konkrete Anforderungen

Für die Entwicklung eines Payment Scheme (Zahlungslösung) müssen als erstes die Teilnehmer für einen Zahlungsvorgang identifiziert werden. Es gibt den Händler, welcher eine Zahlung einfordern will, einen Zahler, welcher dem Händler einen Betrag zahlen soll, sowie die Senderbank des Zahlers und die Empfängerbank des Händlers zwischen welchen schlussendlich der Geldfluss stattfindet. Für die grafische Darstellung kann ein Vier Parteien Modell¹⁰ verwendet werden, um die einzelnen Abläufe besser darzustellen. Vier verschiedene Varianten für die Übermittlung und das Empfangen von Zahlungen werden evaluiert. Bevor auf diese eingegangen wird, müssen die Begriffe Instant Payment und Open Banking weiter erläutert werden.

2.1 Instant Payment (IP)¹¹

Seit August 2024 sind in der Schweiz Instant Payment Zahlungen (SIC 5) verfügbar, die es ermöglichen, Geldbeträge in Schweizer Franken innerhalb von maximal zehn Sekunden von einem Konto auf ein anderes zu überweisen. Diese Transaktionen können rund um die Uhr an allen Tagen des Jahres durchgeführt werden, unabhängig von Uhrzeit, Wochentag oder Feiertag. Im Hintergrund erhalten alle teilnehmenden Banken ein Verrechnungskonto für den SIC Instant Payment Service (SIC-IP Service). Der verfügbare Betrag auf diesem Konto wird für den Transfer von allen Instant Payment Zahlungen von einer Bank zu einer anderen verwendet. Der verfügbare Betrag auf dem IP-Verrechnungskonto muss von der Bank via Übertragszahlung «aufgeladen» werden. Dazu wird Geld vom SIC RTGS-Verrechnungskonto einer Bank auf das IP-Verrechnungskonto übertragen. Dies konnte bis April 2025 nur von Sonntagabend 18 Uhr bis Samstag 12 Uhr erfolgen. Seit dem April 2025 ist der RTGS-Service des SIC¹² auch am Wochenende geöffnet, um Übertragszahlungen zu ermöglichen.¹³

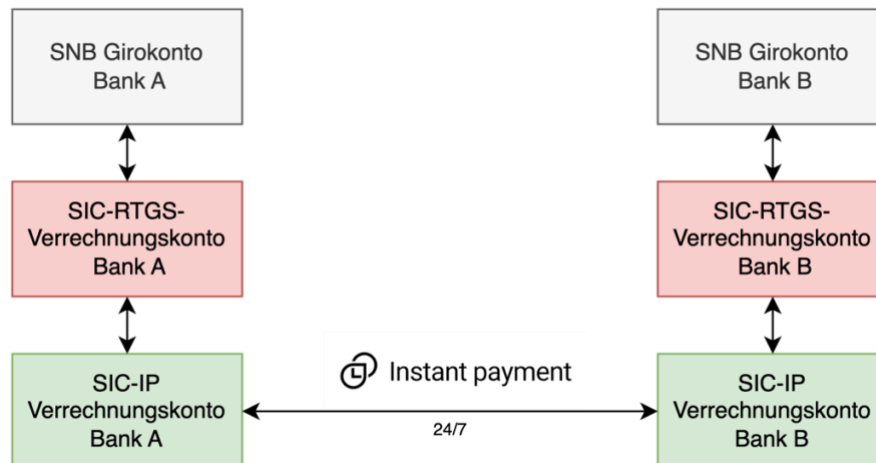


Abbildung 2 Zusammenhang SNB Girokonto und Verrechnungskonto

Überweisungen, die an einem Wochenende oder Feiertag getätigt werden, erfolgen zwar sofort (das Geld wird dem SIX-Verrechnungskonto der Empfängerbank gutgeschrieben), können aber bankindividuell unterschiedliche Valutadaten aufweisen. Das Valutadatum bezeichnet den Tag, an dem eine Bank eine Gutschrift oder Belastung auf einem Konto zinswirksam verbucht. Zinswirksam für die Bank wird ein Betrag

¹⁰ <https://www.six-group.com/dam/download/banking-services/debit-and-mobile-services/de/learning-nugget/learning-nugget-vier-parteien-modell-de.pdf>

¹¹ SIC-IP-Service-Handbuch, Version 2.1

¹² <https://www.six-group.com/de/products-services/banking-services/interbank-clearing/sic.html>

¹³ Aus persönlichem Gespräch mit Timo Pfahl (SIX Interbank Clearing)

erst, wenn er auf dem SNB-Girokonto der Bank liegt, das heisst, wenn er vom Verrechnungskonto auf das SNB-Girokonto übertragen wurde. Zum Beispiel erhält eine am Samstag eingehende Zahlung im E-Banking des Empfängers möglicherweise das Valutadatum vom darauffolgenden Montag, weil die Bank den Übertrag erst an einem Montag vorgenommen hat. Dieses Vorgehen ist von Bank zu Bank unterschiedlich¹⁴.

Bis Ende 2026 sollten alle Schweizer Finanzinstitute in der Lage sein, Instant Zahlungen zu empfangen. Für den Versand von Instant Payment Zahlungen werden die Banken nicht verpflichtet. Dies bedeutet, dass jede Bank freiwillig entscheiden kann, ob sie diesen Service Ihren Kunden anbieten will oder nicht.

2.2 Open Banking

Open Banking ermöglicht es Drittanbietern, über standardisierte Schnittstellen auf Kundendaten und Funktionalitäten von Banken zuzugreifen, natürlich nur mit ausdrücklicher Zustimmung des Kunden. Dies ermöglicht die Entwicklung von innovativen Dienstleistungen, die über traditionelle Bankangebote hinausgehen. Dazu gehören beispielsweise Finanzbuchhaltungstools wie Bexio¹⁵, Klara¹⁶ oder Infinity¹⁷.

Zu den zur Verfügung stehenden Funktionen der API gehören unter anderem das Auslesen von Kontobewegungen und erfassen von Zahlungen an Dritte. Im Vergleich zu anderen Regionen, wie der Europäischen Union mit der PSD2-Richtlinie¹⁸, gibt es in der Schweiz keinen umfassenden regulatorischen Rahmen für Open Banking. Banken können selbst entscheiden, ob sie Open Banking Schnittstellen anbieten oder nicht. Zudem sieht jede Bankeigene Open Banking Implementierung unterschiedlich aus. SIX versucht mit bLink¹⁹ eine Standardisierung in der Schweizer Open Banking Landschaft zu schaffen. Teilnehmende Banken implementieren hierzu eine Schnittstelle zu SIX. Fintech Dienstleister (in der Grafik «Service User») können über eine einheitliche Schnittstelle via SIX bLink von verschiedenen Banken z.B. Kontoinformationen auslesen, Zahlungen erstellen und mehr.

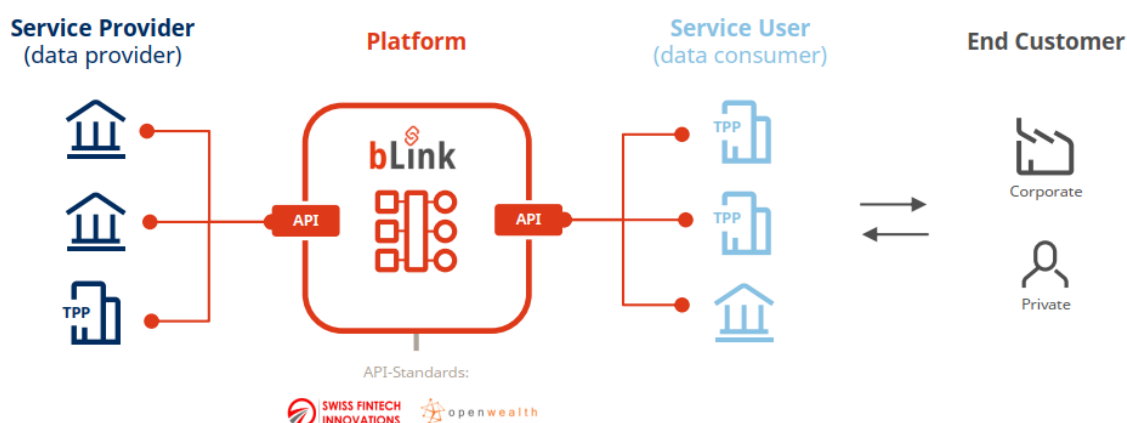


Abbildung 3 Übersichtsgrafik bLink²⁰

2.3 Varianten: Vier Parteien Modell

Die nachfolgenden Varianten beschreiben ein mögliches Vorgehen für die Nutzung der Open Banking Schnittstellen. Bei allen Varianten wird für die Zahlungsübermittlung Instant Payment verwendet.

¹⁴ Aus persönlichem Gespräch mit Julien Rösch (PPI Schweiz)

¹⁵ <https://www.bexio.com/>

¹⁶ <https://www.klara.ch/>

¹⁷ <https://www.infinity.swiss/>

¹⁸ <https://www.bundesbank.de/de/aufgaben/unbarer-zahlungsverkehr/psd2/psd2-775434>

¹⁹ <https://blink.six-group.com/de>

²⁰ <https://docs.blink.six-group.com/docs/whats-blink>

2.3.1 Variante 1: Empfänger Bank benachrichtigt Payment Scheme

In der ersten Variante wird vom Händler eine Zahlungsaufforderung (1) an den Zahler gestellt. Dies geschieht über das Payment Scheme (Zahlungsplattform). Das Payment Scheme generiert für diese Zahlung eine wie im normalen Zahlungsverkehr verwendete QR-Rechnung²¹ (1) mit einer Referenznummer. Der Zahler führt die Zahlung an den Händler aus, indem er durch das E-Banking/Mobile-Banking seiner Bank (Senderbank) die QR-Rechnung scannt und eine normale IBAN-Zahlung an die Empfängerbank des Händlers durchführt (3). Die Empfängerbank sendet dem Payment Scheme über eine Notification (4) einen Hinweis, sobald eine Zahlung auf dem Konto der Empfängerbank eingegangen ist. Das Payment Scheme prüft, ob eine eingegangene Zahlung diejenige ist, welche als Zahlungsaufforderung an den Zahler erstellt wurde (anhand der Referenznummer der QR-Rechnung). Als letzter Schritt (5) wird vom Payment Scheme eine Zahlungsbestätigung oder Ablehnung an den Händler und den Zahler gesendet. Bei dieser Variante wird vorausgesetzt, dass die Senderbank des Zahlers IP-fähig ist und eine Open Banking Schnittstelle existiert, welche Notifications über neue eingegangene Zahlungen an das Payment Scheme gesendet werden können.

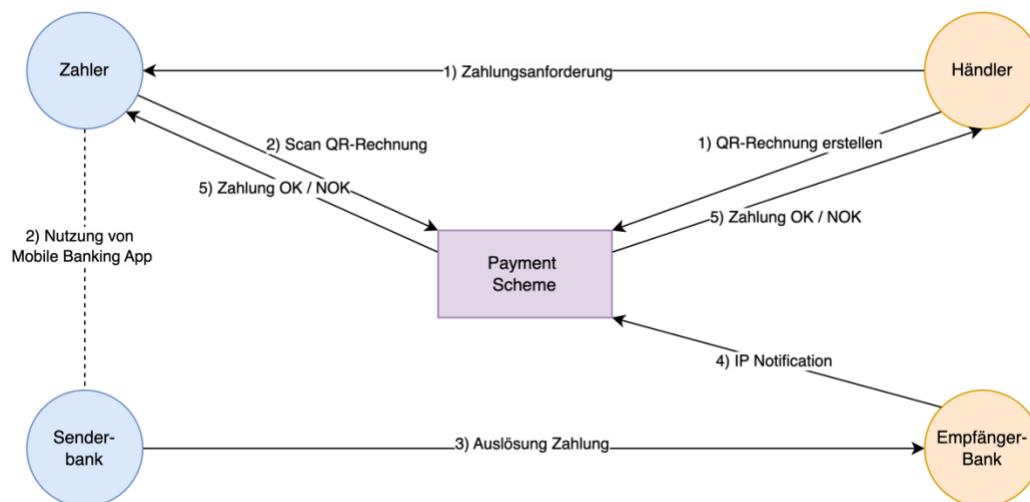


Abbildung 4 Variante Empfänger Bank benachrichtigt Payment Scheme

2.3.2 Variante 2: Payment Scheme holt sich Zahlungsinformationen ab

Die zweite Variante unterscheidet sich nur leicht von der ersten Variante. Hier wird in Schritt 4 anstelle einer Notification bei Zahlungseingängen durch die Empfängerbank eine periodische Überprüfung des Payment

²¹ <https://www.six-group.com/de/products-services/banking-services/payment-standardization/standards/qr-bill.html>

Schemes verwendet. Sofern das Payment Scheme offene Zahlungen hat, werden in einem vordefinierten Sekunden Intervall die Zahlungen auf dem Empfängerkonto abgefragt.

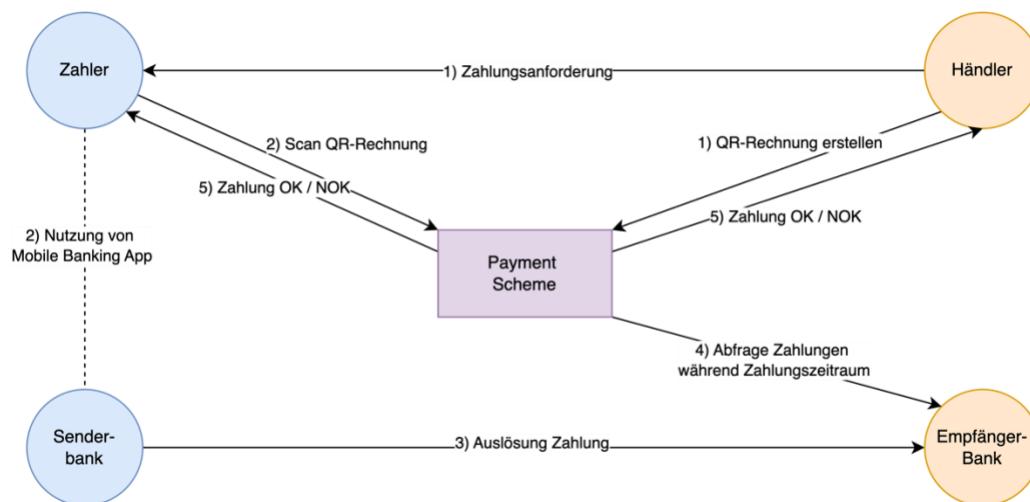


Abbildung 5 Variante Payment Scheme holt sich Zahlungsinformationen ab

2.3.3 Variante 3: Schnittstelle bei Senderbank

In dieser Variante wird der Zahlungsprozess über eine Schnittstelle zur Empfängerbank gesteuert. Nachdem ein Händler eine Zahlungsaufforderung an den Zahler gestellt und einen benutzerdefinierten QR-Code generiert hat (1), scannt der Zahler den QR-Code (2) über eine vom Payment Scheme zur Verfügung gestellte Plattform. Das Payment Scheme löst daraufhin die Zahlung im Format pain.001²² bei der Senderbank des Zahlers aus (3).

Im Anschluss erfordert die Senderbank eine Bestätigung der Zahlung durch den Zahler im E-Banking (4). Nach erfolgreicher Autorisierung initiiert das Payment Scheme die Zahlung bei der Empfängerbank des Händlers (5). Die Empfängerbank bestätigt die erfolgreiche oder nicht erfolgreiche Ausführung der Zahlung an das Payment Scheme (6). Schliesslich erhält der Zahler über das Payment Scheme (7) eine Rückmeldung über den Erfolg oder Misserfolg der Zahlung.

²² <https://zahlungsverkehr.org/internationaler-zahlungsverkehr/iso-20022-standard/pain-001>

Diese Variante setzt voraus, dass der Zahler dem Payment Scheme bekannt ist und dieser sich im Voraus beim Payment Scheme registriert hat.

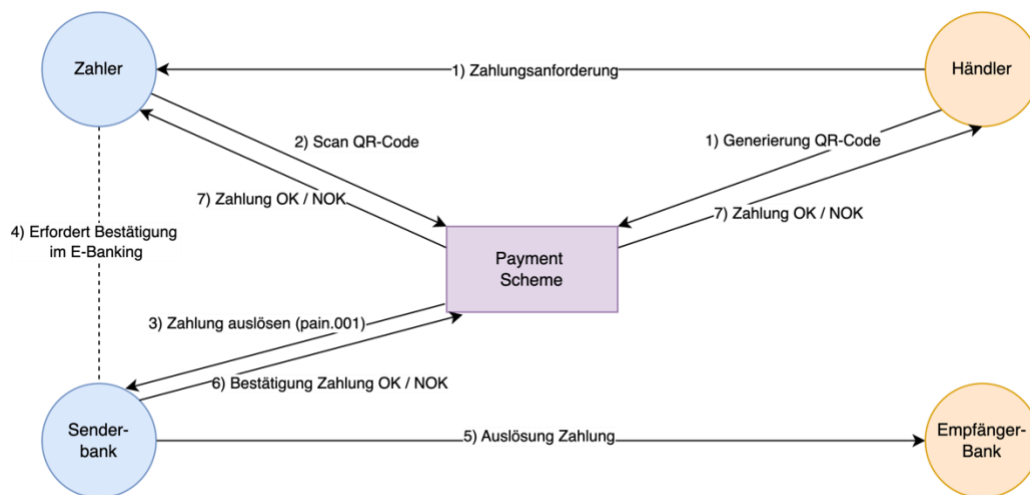


Abbildung 6 Variante Schnittstelle bei Senderbank

2.3.4 Variante 4: Instant Payment Bridge^{23 24}

In dieser Variante initiiert der Händler eine Zahlung, indem er eine Zahlungsanforderung erstellt und einen QR-Code generiert (1). Der Zahler scannt den QR-Code (2) über eine vom Payment Scheme zur Verfügung gestellte Plattform, woraufhin die Senderbank eine Zahlungsanweisung erhält (3). Die Senderbank sendet die Kundenzahlung an die Empfängerbank. Auf der Instant Payment Plattform von SIX sitzt eine zusätzliche Komponente, die Instant Payment Bridge. Die Instant Payment Bridge übernimmt die Weiterleitung der Kundenzahlung an die Empfängerbank (6). Die Sender- und die Empfängerbank erhalten eine Verrechnungsbestätigung (7). Schliesslich erhält sowohl der Zahler als auch der Händler eine Rückmeldung vom Payment Scheme über den Erfolg oder Misserfolg der Zahlung (8).

²³ <https://www.six-group.com/en/products-services/banking-services/billing-and-payments/instant-payments-bridge.html>

²⁴ Aus persönlichem Gespräch mit Julien Rösch (PPI Schweiz)

Diese Variante kann aktuell nicht umgesetzt werden, da die Instant Payment Bridge erst ein Konzeptvorschlag der SIX ist. Die früheste Implementierung der Instant Payment Bridge (sofern sie dann wirklich implementiert wird) wäre im Jahr 2027.

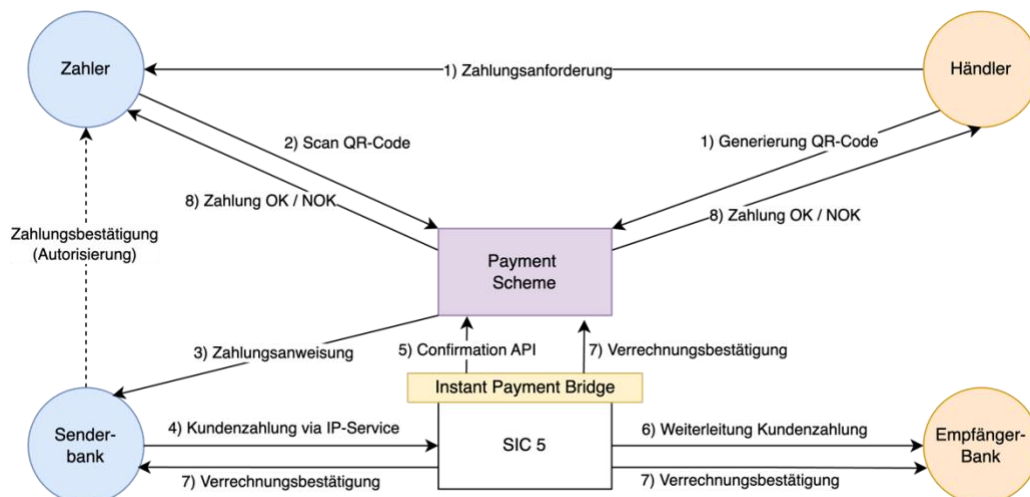


Abbildung 7 Variante Instant Payment Bridge

2.4 Variantenentscheid

Die Variante 4 fällt als erstes weg, da noch nicht klar ist, ob die Instant Payment Bridge überhaupt eingeführt wird. Aktuell laufen Konsultationsverfahren bei den Banken, ob diese so eine Funktion implementieren würden oder nicht. Frühestens könnte die Instant Payment Bridge im Jahr 2027 eingeführt werden und müsste zusätzlich von jeder Bank einzeln implementiert werden. Variante 3 wird ebenfalls verworfen, da bei dieser die zahlende Partei dazu gezwungen wäre eine Bankverbindung mit entsprechend kompatibler Open Banking API zu haben. Nutzer müssten sich zuvor beim Payment Scheme registrieren und bei den meisten Banken werden Zahlungen nicht direkt ausgeführt, sondern müssen zusätzlich im E-Banking validiert werden. Dies würde die Benutzerfreundlichkeit des Tools massiv beeinträchtigen.

Als letztes bleiben die Varianten 1 und 2, welche beide einen ähnlichen Ansatz verfolgen. Die zahlende Partei muss keine zusätzlichen Apps oder Konten erstellen. Es genügt ein Zugriff auf eine E-Banking App und die Funktion eine QR-Rechnung scannen zu können. Variante 1 wäre die geeignetere Variante, da das Payment Scheme aktiv über neue Zahlungen von der Bank informiert wird. Da aktuell keine solche offen verfügbare Schnittstelle in der Schweizer Bankenlandschaft existiert²⁵, wird diese Variante ebenfalls verworfen. Übrig bleibt Variante 2 wo Zahlungen über ein Pull-verfahren in einem Intervall abgefragt werden.

2.5 Empfängerbank

Bei der Wahl einer Empfängerbank stellt sich die Frage auf welches Bankkonto werden die Zahlungen gesendet. Einerseits könnte die Hausbank des Händlers als Empfängerbank fungieren. Hier wäre die Voraussetzung, dass die Empfängerbank eine Open Banking Schnittstelle anbietet und die Schnittstelle zum Payment Scheme unterstützt. Falls dies nicht so ist, müsste der Händler ein Konto bei einer kompatiblen Empfängerbank eröffnen.

Eine weitere Möglichkeit ist, dass das Konto der Empfängerbank im Besitz des Betreibers des Payment Scheme ist. Dies würde bedeuten, dass alle Zahlungen von Zahlern zuerst auf ein Zwischenkonto eingehen und dann direkt oder zu einem bestimmten Zeitpunkt an den Händler weitergeleitet werden. Diese

²⁵ Aus persönlichem Gespräch mit Julien Rösch (PPI)

Möglichkeit bietet den Vorteil, dass der Händler jede Bankverbindung für die Benützung des Payment Scheme verwenden könnte und dass der Betreiber des Payment Scheme direkt allfällige Gebühren abziehen könnte. Der einzige Nachteil dieser Variante ist vermutlich der rechtliche Aspekt, wodurch viele Verträge für die Regelung des Besitzes der Gelder aufgesetzt werden müssten.

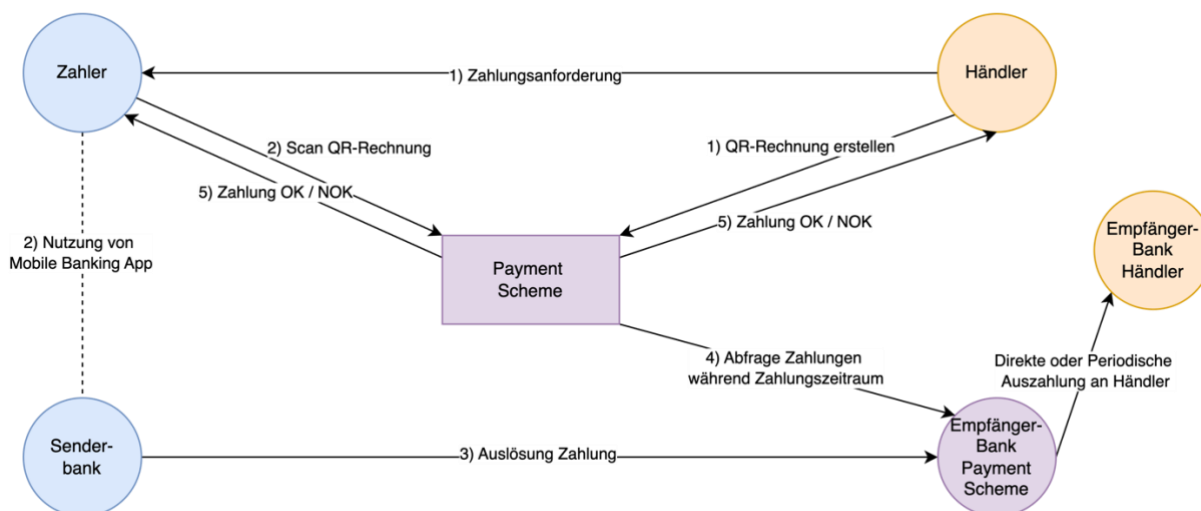


Abbildung 8 Vier Parteien Modell mit zwei Empfängerbanken

Diese Arbeit beleuchtet beide Varianten und die erstellte Software unterstützt sowohl Transaktionen mit und ohne Zwischenkonto. Anhand von den gewonnenen Erkenntnissen werden im [Ausblick](#) weitere Empfehlungen erläutert.

2.6 Unterschied zwischen Instant Payment und Twint^{26 27 28}

Das Erste, was bei einem Vergleich von Twint und Instant Payment beachtet werden muss, ist das Instant Payment allein noch keine Zahlungsplattform bildet. Erst das Zusammenspiel zusätzlicher Komponenten machen aus Instant Payment eine Zahlungsplattform, welche unter der Haube Instant Payment verwendet. Twinten können Personen in der Schweizer bereits seit 2016. Für Nutzer der Plattform wirkt es seit dem Start von Twint so, als würden Zahlungen direkt von einem Konto auf das andere übermittelt. In Wirklichkeit geschehen diese Zahlungen jedoch versetzt. Zum Zeitpunkt des Aufbaus der Twint Plattform konnten Zahlungen nur innerhalb derselben Bank Instant abgewickelt werden. Instant Zahlungen zu anderen Banken waren damals nicht möglich. Zur Lösung dieses Problems unterhält Twint bei den Banken ein «Poolaccount». Wenn eine Person 1 mit Konto bei Bank A Geld an Person 2 mit Konto bei Bank B sendet, wird der Betrag bei Person 1 dem Konto abgebucht und auf den Poolaccount der Bank A überwiesen. Die Bank B überweist der Person 2 dann den Betrag vom Poolaccount auf sein Konto. Über Nacht werden die entsprechenden Beträge zwischen den Banken überwiesen, um die Poolaccount auszugleichen. Diese

²⁶ <https://twint.ch>

²⁷ Enthält Informationen aus persönlichem Gespräch mit René Hägeli (Twint)

²⁸ <https://www.netzwoche.ch/news/2024-07-11/das-sagt-der-twint-ceo-zur-einfuehrung-von-instant-payment>

Ausgleichszahlungen werden über dasselbe System wie bei normalen Banküberweisungen (RTGS) abgewickelt.

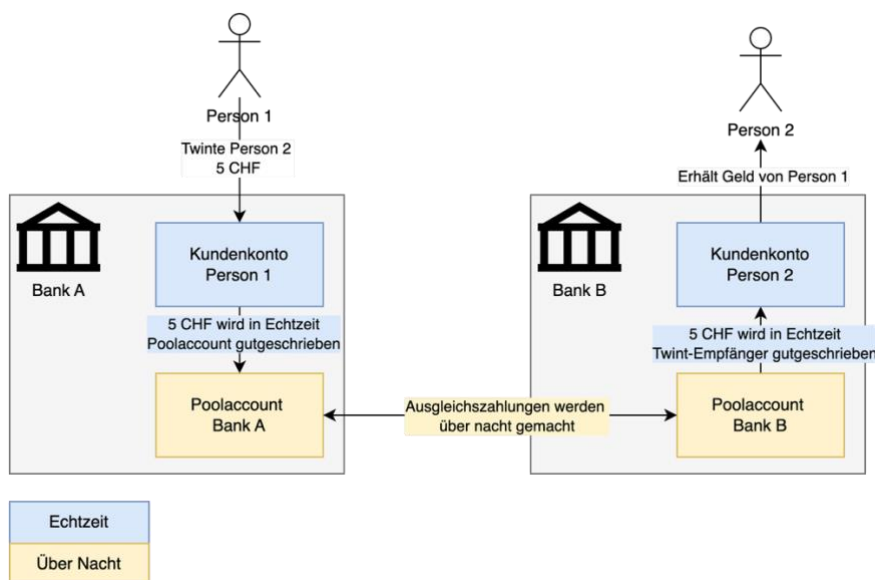


Abbildung 9 Ablauf Twint Überweisung zwischen zwei Personen mit unterschiedlichen Banken

2.7 Technischer Ablauf einer Zahlung

Die entwickelte Applikation in dieser Arbeit beschränkt sich im ersten Schritt auf Zahlungen im Onlinehandel und wird bei genügend Kapazität erweitert durch Zahlungen am physischen «Point of Sale²⁹». Für Onlinezahlungen orientiert sich der Workflow einer Zahlung an dem Zahlungsprozess von Stripe³⁰. Stripe ist einer der führenden Online-Zahlungsanbieter im globalen Markt. Mit Stripe können Zahlungen mit fast allen Zahlungsmethoden getätigt werden und der digitale Zahlprozess durch eine API-Schnittstelle in jede Applikation eingebunden werden. Das Ziel einer einfachen Einbindung in z.B. einen Onlineshop soll auch in dieser Arbeit verfolgt werden. Nachfolgend eine Übersicht eines Prozesses von einer Kundenzahlung in einem Onlineshop.

In der Applikation des Händlers initiiert der Kunde den Bezahlprozess mit der Zahlungsmethode «Instant Pay». Die Händlerapplikation erstellt eine Zahlungssession über einen API Request an das Payment Scheme backend. Dieses generiert eine Zahlungs-URL und eine Referenznummer. Über die Referenznummer hat der Händler die Möglichkeit eine API-Abfrage über den Status der Zahlungssession zu machen. Die Händlerapplikation leitet den Benutzer über den Zahlungslink weiter zur Bezahlseite (Payment Scheme Frontend). Der Benutzer, welcher die Zahlung durchführt, befindet sich jetzt in der Applikation des Payment Scheme und nicht mehr im Onlineshop des Händlers. Der Bezahler scannt die vom Payment Scheme zur Verfügung gestellte QR-Rechnung und führt die Zahlung in der E-Banking App der Bank des Benutzers aus. Im gleichen Schritt startet das Payment Scheme die periodische Prüfung der Zahlung im Hintergrund mithilfe der Schnittstelle zur Empfängerbank. Falls die Zahlung erfolgreich validiert wurde, wird der Benutzer vom Payment Scheme Frontend an die Bestätigungsseite des Händlers weitergeleitet. Sollte die Zahlung fehlgeschlagen sein (NOK-Zahlung) wird der Benutzer an eine Fehlerseite des Händlers

²⁹ <https://wirtschaftslexikon.gabler.de/definition/point-sale-pos-46867>

³⁰ <https://stripe.com/de-ch>

integriert werden. Die Applikation soll auf der Checkout Seite für den Bezahlprozess jeweils nur den Swiss QR Code aus der QR-Rechnung anzeigen.

Empfangsschein	Zahlteil	Konto / Zahlbar an								
Konto / Zahlbar an CH64 3196 1000 0044 2155 7 Max Muster & Söhne Musterstrasse 123 8000 Seldwyla Referenz 00 00082 07791 22585 74212 86694 Zahlbar durch Simon Muster Musterstrasse 1 8000 Seldwyla <table> <tr> <th>Währung</th> <th>Betrag</th> </tr> <tr> <td>CHF</td> <td>50.00</td> </tr> </table>	Währung	Betrag	CHF	50.00	 <table> <tr> <th>Währung</th> <th>Betrag</th> </tr> <tr> <td>CHF</td> <td>50.00</td> </tr> </table>	Währung	Betrag	CHF	50.00	CH64 3196 1000 0044 2155 7 Max Muster & Söhne Musterstrasse 123 8000 Seldwyla Referenz 00 00082 07791 22585 74212 86694 Zusätzliche Informationen Bezahlung der Reise Zahlbar durch Simon Muster Musterstrasse 1 8000 Seldwyla
Währung	Betrag									
CHF	50.00									
Währung	Betrag									
CHF	50.00									

Annahmestelle

Abbildung 11 Beispiel QR-Rechnung inkl. Swiss QR Code

Ein Punkt, welcher bisher nicht durch den Swiss QR-Code beeinflusst werden konnte, ist der Zahlungstyp. Es wird nicht vorgegeben, ob die gescannte Zahlung der E-Banking App eine Instant Payment Zahlung oder eine normale RTGS-Zahlung³² (Zahlung an Bankwerktagen) sein muss. Dies wird (sofern die Bank überhaupt Instant Payment Zahlungsverband unterstützt) vom Benutzer oder automatisch festgelegt. Falls in dem Bezahlprozess eine RTGS-Zahlung versendet wird, würde die Zahlung erst am nächsten Bankwerktag beim Empfänger ankommen.

2.9 Statuskonzept

Um die verschiedenen Zustände eines Bezahlvorgangs zu kategorisieren, werden die nachfolgenden Status verwendet. Ein Bezahlvorgang kann nur einen Status haben.

- **Offen:** Zahlungssession wurde erstellt und Benutzer kann auf die Checkout Seite mithilfe der generierten Session URL umgeleitet werden. Die Zahlungssession muss innerhalb der nächsten 5 Minuten initialisiert werden, ansonsten läuft sie ab.
- **Initialisiert:** Der Benutzer hat die Zahlungsseite geöffnet. Ab diesem Zeitpunkt hat muss die Zahlung innerhalb von 10 Minuten ausgeführt werden.
- **Bezahlt:** Die Zahlung ist innerhalb der Zahlungsfrist eingetroffen und hat keinen NOK-Status (siehe [Überprüfung von NOK Zahlungen](#)).
- **Abgebrochen:** Für die Zahlungssession ist ein NOK Status Ereignis eingetroffen (siehe [Überprüfung von NOK Zahlungen](#)). Neben dem Status wird ein Grund in Form eines Satzes für den Abbruch angegeben.

³² <https://www.six-group.com/de/products-services/banking-services/payment-standardization/downloads-faq/glossary.html#scrollTo=SIC>

Nachfolgend werden die verschiedenen Abhängigkeiten der Status durch ein Zustandsdiagramm dargestellt.

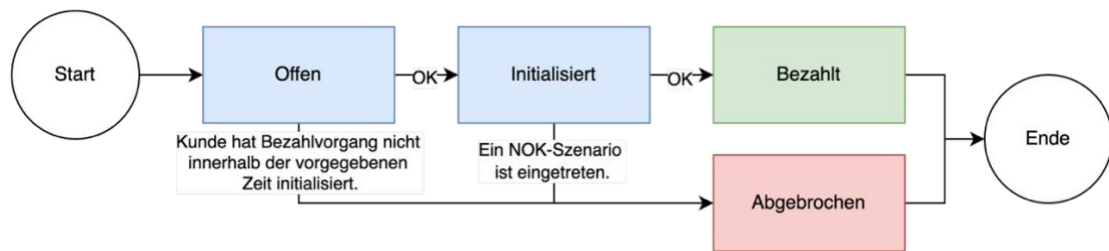


Abbildung 12 Zustandsdiagramm Statuskonzept

2.10 Überprüfung von NOK-Zahlungen

Da der Prozess von einer Benutzereingabe in der E-Banking App des Nutzers abhängig ist, müssen verschiedene Szenarien abgefangen werden können.

- Es wurde zu wenig Geld vom Zahler überwiesen.
- Es wurde zu viel Geld vom Zahler überwiesen.
- Es wurde innerhalb des Timeouts kein Geld vom Zahler überwiesen.
- Es wurde nach dem Timeout Geld vom Zahler überwiesen.
- Der Betrag wurde in einer Fremdwährung überwiesen.
- Der Zahler bricht die Zahlung über das Payment Scheme Frontend ab.
- Es wurde ein Betrag auf das Empfängerkonto überwiesen, welcher nicht erwartet wurde.
- Es wurden, während eine Zahlungssession im Status «offen» war, mehrere Zahlungen gesendet.

In all diesen Fällen soll die Zahlung den Status «abgebrochen» erhalten und falls ein Betrag überwiesen wurde, dieser vollständig an den Zahler zurücküberwiesen werden.

Werden für eine Zahlungssession mehrere Zahlungen empfangen, und die erste Zahlung wurde bereits verbucht und als "bezahlt" markiert, werden alle nachfolgenden Zahlungen direkt an den Absender zurückerstattet.

2.11 Mindestanforderungen an Open Banking Schnittstelle

Da es keinen Schweizweiten Standard für Open Banking Schnittstellen gibt und daher jede Bank unterschiedliche Schnittstellen implementiert hat, wird hier eine Beschreibung von Mindestanforderungen an die Open Banking Schnittstelle gestellt.

Die Bank muss in der Lage sein zwei API-Requests anzubieten. Der erste und wichtigste Request ist die Abfrage von aktuellen Kontobewegungen auf einem Bankkonto. Es müssen nicht alle Transaktionen abgefragt werden können, lediglich diejenigen der letzten 30 Minuten. Die Transaktionen müssen für eine ordentliche Überprüfung folgende Informationen beinhalten:

- Eindeutige Transaktionsnummer der Bank (ID)
- Referenznummer (sofern vom Zahler angegeben)
- Betrag
- Währung
- Name des Bezahlers
- IBAN des Bezahlers
- PLZ des Bezahlers
- Ort des Bezahlers

Zum Abgleich der Transaktion werden Transaktionsnummer, Referenznummer, Betrag und Währung herangezogen. Für eventuelle Rückzahlungen an den Zahler, falls dessen Zahlung nicht erfolgreich ist (NOK-Szenario), sind Name, IBAN, Postleitzahl und Wohnort des Zahlers erforderlich.

Der zweite Request, welcher von der Schnittstelle benötigt wird, ist die Möglichkeit Geld an eine IBAN zu versenden. Diese Funktion wird benötigt, um im Falle von NOK-Zahlungen den Betrag an den Zahler zurückzusenden und für Auszahlungen an den Händler. Für eine Anfrage an die Schnittstelle sollen jeweils die folgenden Felder angegeben werden:

- Name
- IBAN
- Postleitzahl
- Wohnort

Neben den auszutauschenden Datenfeldern muss die Open Banking Schnittstelle Echtzeitdaten liefern können. Dies bedeutet, dass wenn in der Bank eine Instant Payment Zahlung eintrifft und verarbeitet wird (in der Grafik Musterbank B), diese innerhalb von Sekunden auch über die Open Banking Schnittstelle abgefragt werden kann.

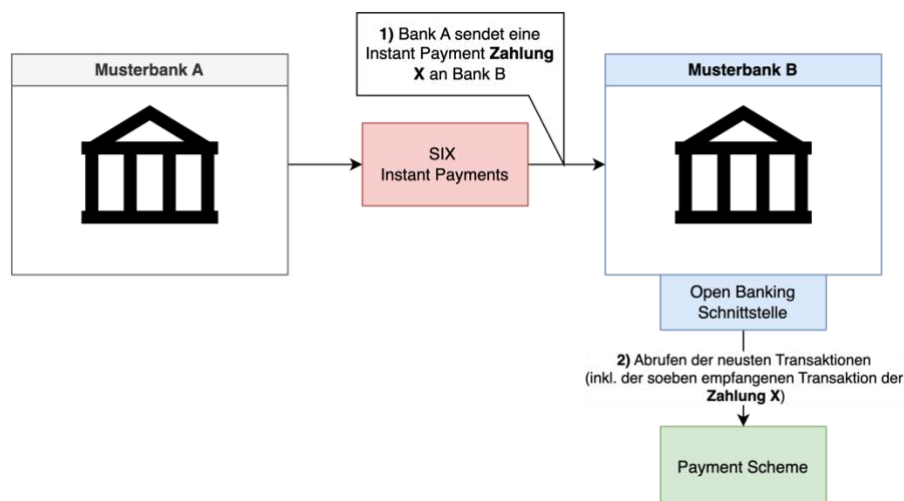


Abbildung 13 Einfache Darstellung Zusammenspiel Instant Payment und Open Banking

Um ein Prototyp end-to-end zu implementieren und testen zu können wird für diese Arbeit ein Testbankensystem benötigt. Idealerweise soll dies in Partnerschaft mit einer Bank geschehen. Nach der Anfrage einiger Banken wurde mit der Innofactory AG der ideale Partner gefunden. Die Innofactory AG³³ entwickelt Innovationen in der Bankenwelt und setzt diese gemeinsam mit ihren Partnern um. Die Trägerschaft der Innofactory AG bildet einerseits die Hypothekbank Lenzburg AG³⁴ sowie die Berner Kantonalbank AG³⁵. Ein Produkt, welches die Innofactory anbietet, ist «Imagine Banking³⁶», eine Lern- und Entwicklungsplattform, die Entwicklern, Fintechs und Bildungsinstitutionen ermöglicht, Innovationen in einer realen Bankenumgebung zu testen und zu optimieren. Mithilfe der «Imagine Banking» Plattform wird das Payment Scheme dieser Arbeit umgesetzt.

³³ <https://www.innofactory.ch/>

³⁴ <https://www.hbl.ch/de/ueber-uns/>

³⁵ <https://www.bekb.ch/de/die-bekb>

³⁶ <https://www.innofactory.ch/angebot/>

2.12 Technologiewahl

Die Applikation besteht aus drei Teilen. Der erste Teil stellt eine Oberfläche für einen Händler dar. Durch diese erhält der Händler eine Übersicht aller Transaktionen und kann Zahlungssessions verwalten. Der zweite Teil ist die Bezahlseite, welche eine QR-Rechnung anzeigt, die in einem Zahlprozess vom Zahler gescannt werden kann, um die Zahlung auszuführen. Der dritte Teil ist das Herzstück des Systems, nämlich das Backend und die Datenbank. Das Backend enthält alle Daten aller Transaktionen und führt Abgleiche mit der Open Banking Schnittstelle durch. Zudem kann das Backend über eine API-Schnittstelle Zahlungssessions erstellen.

Für die Umsetzung dieser drei Teile gibt es fast unendlich viele Varianten von Technologien, welche verwendet werden könnten. Eine Voraussetzung für das System ist eine hohe Skalierbarkeit und die Möglichkeit der Verarbeitung von vielen parallelaufenden Zahlungssessions. Die Anforderung an das Frontend sind einfache Erweiterbarkeit und Kompatibilität mit einer simplen und für den Benutzer einfach zu verstehenden UI Library. Im folgenden Abschnitt wird auf zwei Varianten eingegangen und es werden die jeweiligen Vor- und Nachteile beschrieben.

- **Tech-Stack 1: TypeScript Frontend + backend mit Next.JS³⁷**
 - Vorteile:
 - Gesamte Codelogik in einem Projekt und derselben Sprache
 - Dokumentation und Beispiele sowie Support durch Community
 - Zahlreiche UI Libraries mit fertigen React Komponenten
 - Grosse Auswahl an npm Packages für diverse Use-Cases (Authentifizierung, UI, Util-Libraries)
 - Nachteile:
 - Horizontale Skalierung von Next.JS Applikationen bringt ein paar Schwierigkeiten mit, wenn die Applikation nicht auf Vercel³⁸ Deployed wird (z.B. Probleme mit distributed caching³⁹).
 - Next.JS Backends laufen auf NodeJS in einem einzigen Thread⁴⁰, was die Performance bei hohem Traffic stark beeinflussen könnte. Zudem wird das Backend bei hohem Traffic durch SSR⁴¹ der einzelnen Seiten ausgebremst.
- **Tech-Stack 2: Next.js (Händler Frontend), Angular⁴² (Checkout Frontend) und .NET⁴³ Backend**
 - Vorteile:
 - .NET backend bietet hohe Leistung und Skalierbarkeit.
 - .NET Ökosystem bietet Vielzahl an Libraries und Frameworks
 - Trennung von Verantwortlichkeiten zwischen Frontend und Backend.
 - Next.JS: Zahlreiche UI Libraries mit fertigen React Komponenten.
 - Next.JS kann für Server spezifische Tasks vom Händler erweitert werden (z.B. Generierung eines Reports)

³⁷ <https://nextjs.org/>

³⁸ <https://vercel.com/>

³⁹ <https://nextjs.org/docs/app/building-your-application/caching>

⁴⁰ <https://codewithpawan.medium.com/understanding-the-single-threaded-nature-of-node-js-62ec55687d1d>

⁴¹ <https://nextjs.org/docs/app/building-your-application/rendering>

⁴² <https://angular.dev/>

⁴³ <https://dotnet.microsoft.com/en-us/>

- Angular Checkout Frontend: Die Bezahlseite ist die am häufigsten aufgerufene Seite im System und dadurch das mit Angular alle UI-Sourcen auf Frontend sind (SPA), wird für generieren des UI keine Serverressourcen verwendet.
- Einfache Client und Server Side Validierung von Benutzereingaben.
- Erstellung von API Routes und Generierung von DTOs mithilfe von OpenApi⁴⁴.
- Nachteile:
 - Höherer Konfigurationsaufwand für die Integration von Frontend und Backend.
 - Es wird Expertise in verschiedenen Tech-Stacks benötigt.
 - Komplexere Bereitstellung (mehrere Deployments).

Die Wahl des Tech-Stacks fällt auf Next.js für das Händler-Frontend, Angular für die Checkout Seite und .NET für das Backend. Diese Kombination adressiert die Anforderungen des Projekts, insbesondere in Bezug auf Performance und Skalierbarkeit. Eine reine Next.js Applikation wird trotz Vorteile einer einheitlichen Programmiersprache aus Performance Gründen nicht in Betracht gezogen. Andere Programmiersprachen für das Backend, wie beispielsweise Java, werden nicht in die Evaluation miteinbezogen, da bei den gewählten Technologien das meiste Knowhow besteht und im Falle von Java .NET aus Sicht des Entwicklers durch „syntactic sugar“ und anderen Vorteilen eine höhere Produktivität verspricht. Eine Möglichkeit mit .NET wäre die Nutzung von Blazor⁴⁵. Diese Variante wird ebenfalls nicht in Betracht gezogen, da Blazor eine verhältnismässig beschränkte Unterstützung durch externe Libraries und wenig Community Support bietet.

2.13 Funktionale Anforderungen (FA)

ID	Beschreibung	Kategorie
FA-1	Händler muss sich registrieren und ein Konto erstellen können.	Benutzerverwaltung
FA-2	Händler muss seine Bankkontodetails für Auszahlung hinterlegen können.	Benutzerverwaltung
FA-3	Händler muss über die API-Zahlungsaufforderungen erstellen können.	Zahlungen
FA-4	Applikation muss eine QR-Rechnung generieren können, welche alle notwendigen Zahlungsinformationen für eine Zahlung enthält.	Zahlungen
FA-5	Zahler muss den QR-Code mit einer E-Banking App scannen können.	Zahlungen
FA-6	Die Applikation muss eine Schnittstelle zur Empfängerbank haben, um Zahlungen zu empfangen.	Zahlungen
FA-7	Die Applikation muss regelmässig überprüfen, ob Zahlungen auf dem Empfängerkonto eingegangen sind.	Zahlungen
FA-8	Die Applikation muss den Status der Zahlung aktualisieren, sobald sie empfangen wurde.	Zahlungen
FA-9	Nachdem die Zahlung eingegangen ist, muss der Zahler auf der Zahlseite zur Erfolgsseite des Händlers weitergeleitet werden.	Zahlungen
FA-10	Die Applikation muss eine Historie aller Zahlungen für Händler und Zahler führen. Diese können nach Datum gefiltert werden.	Reporting
FA-11	Die Applikation muss eine API anbieten, über die Händler den Status der Zahlungssession abrufen können.	Zahlungen

⁴⁴ <https://www.openapis.org/>

⁴⁵ <https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor>

FA-12	Die Applikation muss dem Empfänger das gesammelte Guthaben aller ihm zugeordneten Zahlungen überweisen können (Initiierung über Open Banking Schnittstelle).	Zahlungen
FA-13	Der Händler muss dem Zahler Beträge manuell zurückerstatten können.	Zahlungen
FA-14	Falls ein NOK-Fall für eine Zahlung eintritt, wird automatisiert eine Rückzahlung erstellt (sofern eine Zahlung eingegangen ist).	Zahlungen

Tabelle 1 Funktionale Anforderungen

2.14 Nicht Funktionale Anforderungen (NFA)

ID	NFA-1
Anforderung	Gleichzeitige Zahlungssessions
Beschreibung	Das System muss in der Lage sein, mindestens 10 gleichzeitige Zahlungssessions zu verarbeiten.
Messung	Anzahl der gleichzeitig laufenden Zahlungssessions, die das System korrekt verarbeiten kann.
Verifizierung	Alle Zahlungssessions werden erfolgreich erstellt und der Abgleichprozess wird für alle 10 Zahlungen ausgeführt.

Tabelle 2 Nicht Funktionale Anforderung 1

ID	NFA-2
Anforderung	Gerätekompatibilität der Bezahlseite
Beschreibung	Die Applikation muss auf verschiedenen Geräten (Desktop, Mobile) und Browsern funktionieren.
Messung	Die Bezahlseite wird im Zahlungsprozess auf dem Desktop in Google Chrome, Edge, Firefox und Safari sowie Mobile auf Google Chrome und Safari geöffnet.
Verifizierung	Auf jedem Browser wird die Seite erfolgreich geladen und die QR-Rechnung angezeigt.

Tabelle 3 Nicht Funktionale Anforderung 2

ID	NFA-3
Anforderung	Skalierbarkeit
Beschreibung	Die Applikation soll problemlos hoch und runterskaliert werden können, ohne Datenverlust oder Serviceunterbruch.
Messung	Von einer Instanz auf 10 Instanzen hochskalieren und gleichzeitig 10 Zahlungssessions simulieren.
Verifizierung	Zahlungssessions erfolgreich erstellt und die Benutzer wurden auf die Bezahlseite weitergeleitet.

Tabelle 4 Nicht Funktionale Anforderung 3

ID	NFA-4
Anforderung	Portabilität
Beschreibung	Die Applikation muss auf verschiedenen Cloud-Umgebungen lauffähig sein.
Messung	Laufen lassen der Applikation in einer Docker Umgebung sowie einer Kubernetes Umgebung.
Verifizierung	In allen Umgebungen wird die Applikation erfolgreich gestartet, kann Händlerbenutzer erstellen und Zahlungssessions erstellen.

Tabelle 5 Nicht Funktionale Anforderung 4

ID	NFA-5
Anforderung	Error handling backend API → Standardisierte Error Messages

Beschreibung	In einem Fehlerfall gibt das Backend standardisierte Fehlermeldungen zurück (RFC-7807 ⁴⁶).
Messung	Einfügen einer Exception an einer beliebigen Stelle und Request mithilfe von Postman durchführen.
Verifizierung	Response muss Content-Type «application/problem+json» Haben und im JSON-Body die properties «title» und «type».

Tabelle 6 Nicht Funktionale Anforderung 5

3. Lösungskonzept

Im nachfolgenden Kapitel wird die Architektur mithilfe von Diagrammen dargestellt und beschrieben.

3.1 Datenbankschema

Das Datenbankschema besteht aus neun Tabellen, wobei die Tabelle _EFMigrationHistory nur zur nachverfolgen von Schemaanpassungen verwendet wird. Die restlichen Tabellen Users, Merchants, ApiTokens, PaymentSessions, PaymentTransaction, PaymentTransactionCreditors, PaymentTransactionDebitors und Cashboxes enthalten für die Applikation relevante Informationen. Für Händler wird ein Eintrag in der Tabelle «Users» mit der Rolle «Merchant» erstellt. Ein Händleruser kann genau einem «Merchant» zugeordnet werden. Entweder kann ein User einem bestehenden Merchant zugewiesen werden oder ein neuer Merchant für einen User wird erstellt. Ein Merchant hat keine oder mehrere ApiTokens, mit welchen z.B. ein Onlineshop über die REST-API zum Backend eine «PaymentSession» erstellen kann. Eine PaymentSession beschreibt einen gesamten Bezahlvorgang für einen Käufer. Eine PaymentSession sollte in der Regel jeweils eine PaymentTransaction zugeordnet sein. Eine PaymentTransaction entspricht einer Banktransaktion, welche über die Open Banking Schnittstelle zur Bank importiert wird und enthält Informationen über den Debitor und Kreditor (PaymentTransactionDebitor, PaymentTransactionCreditor). Eine PaymentTransaction wird anhand einer Referenznummer (PaymentReference) mit einer Payment Session abgeglichen. Diese Referenznummer entspricht der Referenznummer auf einer QR-Rechnung, wodurch eingegangene Zahlungen auf einem Bankkonto eindeutig zu einer Payment Session zugeordnet werden können. Da es sein kann, dass ein Zahler mehrere Transaktionen mit derselben Referenznummer machen, besteht eine eins zu N Beziehung zwischen PaymentSessions und PaymentTransactions. Eine PaymentTransaction enthält zudem eine Informationen zu Zahler und eine Information zu Empfänger (PaymentTransactionDebitor, PaymentTransactionCreditor, 1 zu 1 Beziehung wird in ERD mit 1 zu N abgebildet, wird jedoch mit Constraint auf Property

⁴⁶ <https://datatracker.ietf.org/doc/html/rfc7807>

PaymentTransactionId sichergestellt). Cashboxes entsprechen verschiedenen Kassen eines Merchants, welche einer PaymentSession optional zugeordnet werden können.

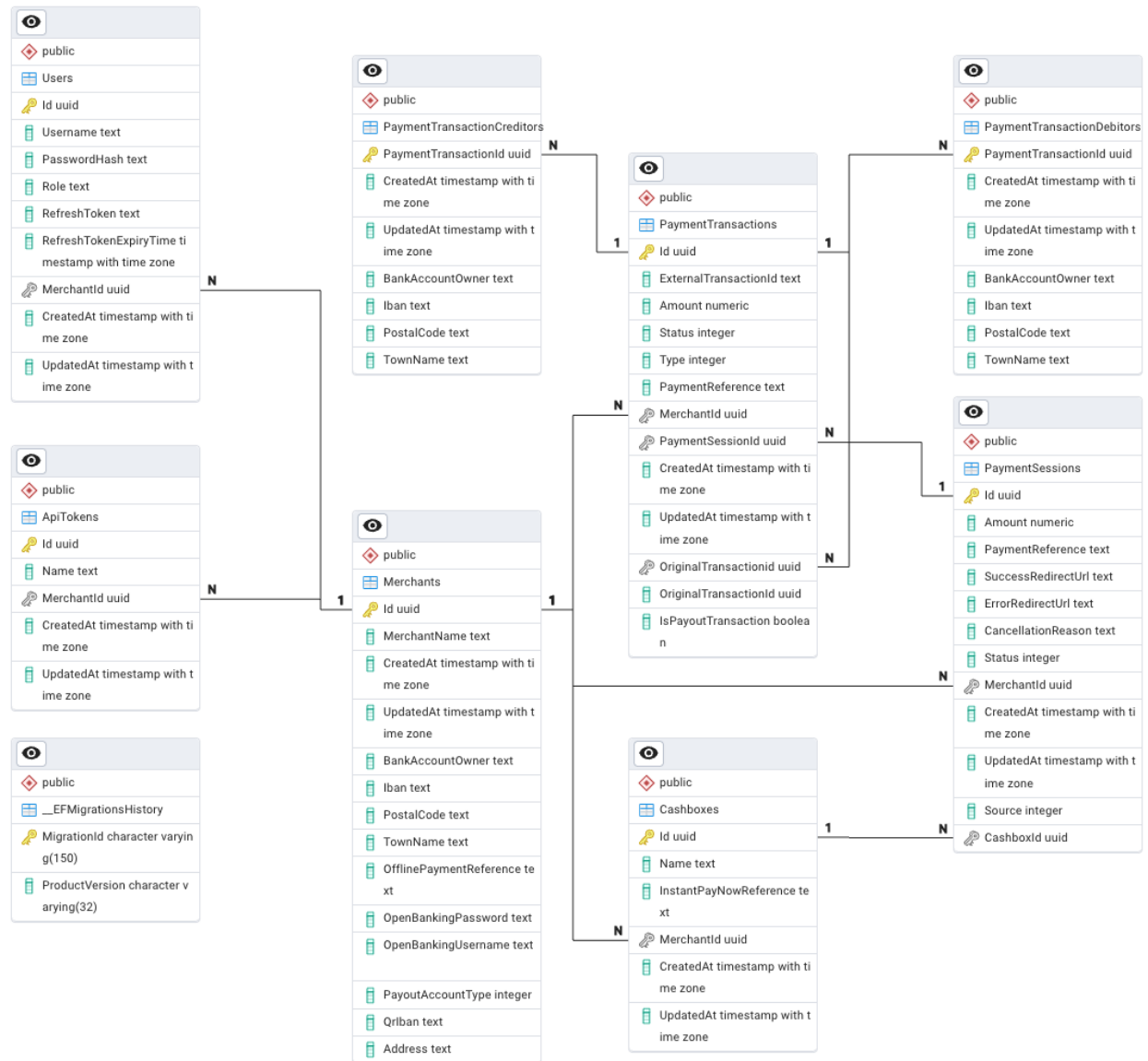


Abbildung 14 ERD Schema Instant Pay

3.2 C4 Diagramm

Nachfolgend wird das Zusammenspiel der einzelnen Komponenten durch Level 1 und Level 2 in einem C4 Diagramm dargestellt.

3.2.1 Level 1: Kontext

Auf dem ersten Level des C4-Diagramms befinden sich zwei Akteure und zwei externe Systeme. Der Erste Akteur ist der Händler, welcher sich im System registrieren kann und eine Übersicht über alle Zahlungssessions hat. Zudem hat er die Möglichkeit seinen Webshop einzubinden, sodass dieser Zahlungssessions über eine REST API Schnittstelle mithilfe eines API Keys erstellen kann. Der zweite Akteur, der Bezahler, kauft im Webshop ein und wird am Ende für die Bezahlung an eine Checkout Seite auf die Zahlungsplattform (Instant Pay) weitergeleitet. Damit Instant Pay verifizieren kann, ob eine Zahlung

eingegangen ist, überprüft die Applikation in einem Intervall alle eingegangenen Zahlungen auf dem Empfängerkonto über eine Open Banking Schnittstelle.

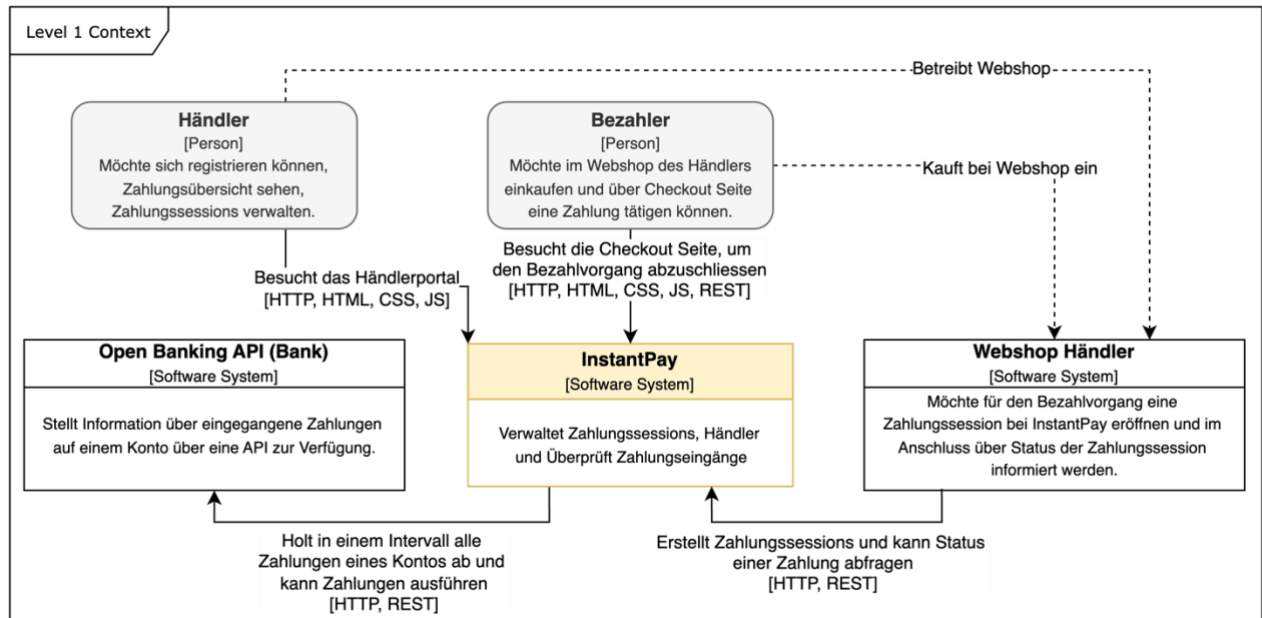


Abbildung 15 C4 Diagramm Level 1

3.2.2 Level 2: Containers

Das C4-Diagramm auf Level 2 visualisiert die detaillierte Container-Architektur des Systems. Das System „Instant Pay“ umfasst zwei Haupt-Frontends. Das erste ist das Händler Portal, realisiert mit Next.js, dass Händlern eine Benutzerschnittstelle zur Verwaltung ihrer Konten, zur manuellen Erstellung von Zahlungssessions und zur Generierung von API-Schlüsseln bietet. Das zweite Frontend ist die Checkout Seite, realisiert mit Angular, welches den Bezahlern die zur Bezahlung notwendige QR-Rechnung anzeigt. Das Backend besteht aus dem Backend, realisiert mit .NET, dass die Kernlogik für die Erstellung, Verwaltung und Abwicklung von Zahlungen bereitstellt, sowie die Logik für die Zahlungsabwicklung und die Kommunikation mit der Open Banking API der Bank enthält. Die Datenhaltung erfolgt über eine Postgres Datenbank, die Informationen wie Händlerprofile und Zahlungssessions speichert, sowie Redis, dass für die

Benachrichtigung aller Backend-Pods über neue Transaktionen mittels eines Pub/Sub-Mechanismus und Caching verwendet wird.

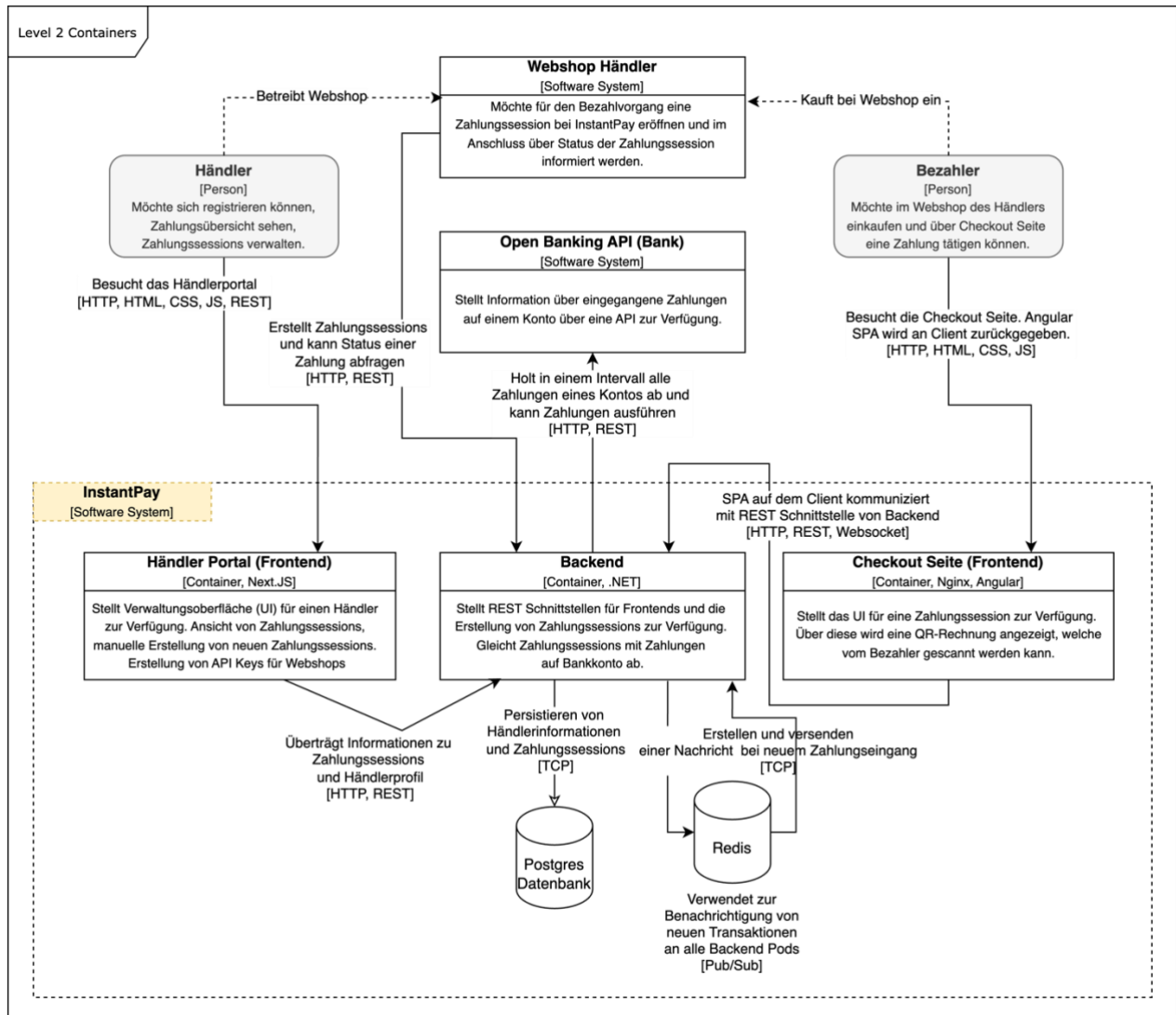


Abbildung 16 C4 Diagramm Level 2

3.3 Codearchitektur

Der Code von Instant Pay ist modular aufgebaut. Das Backend ist in Controller, Services und Models unterteilt. Im Backend sind drei «boundaries» definiert: die API-Grenze für die REST API zwischen Backend und dem Next.JS Frontend (Händlerportal) sowie dem Angular Frontend (Checkout Seite), die Adapter-Grenze welche für das Abholen von Transaktionsdaten sowie der Ausführung von Zahlungen an die Open Banking API Schnittstelle und dem Versenden von Transaktionsbenachrichtigungen über Redis verwendet wird und die DB-Grenze zur Datenbank.

Das Frontend ist komplett unabhängig vom Backend und könnte jederzeit durch eine andere Technologie (z.B. React oder Vue) ersetzt werden. Da das .NET-Backend eine Open API Spezifikation zur Verfügung stellt, kann für jegliche Technologie eine einfache Anbindung an das Backend implementiert werden.

Da das Backend Entity Framework Core⁴⁷ verwendet, ist die Postgres Datenbank komplett vom Backend entkoppelt. Bei Bedarf könnte die Datenbank mit MySQL⁴⁸ oder einer NoSQL Datenbank wie MongoDB⁴⁹ ersetzt werden. Damit dies möglich ist, muss der verwendete EF-Core Datenbankprovider ersetzt werden⁵⁰. Der Code für die Datenbankabfragen kann grösstenteils unverändert für verschiedene Datenbanksysteme verwendet werden.

Services befinden sich im .Net Backend im Projekt „InstantPayBackend.Business“, Code zu dem Adapter im Projekt „InstantPayBackend.Adapter“, API-Controller in „InstantPayBackend“ und Datenbankspezifischer Code im Projekt „InstantPayBackend.Database“. Die Models und DTOs finden sich im Projekt „InstantPayBackend.Model“.

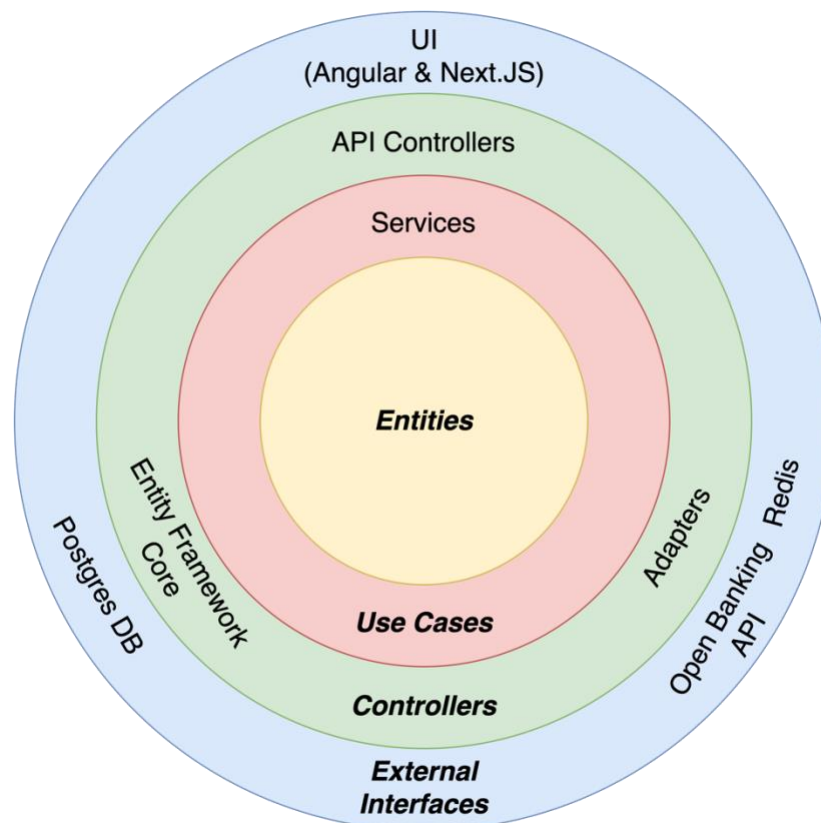


Abbildung 17 Diagram Clean Code

3.4 Open Banking API Schnittstelle

Die API der Innofactory stellt vier Requests zur Verfügung, welche für den Abgleich von Transaktionen und das Versenden von Geld verwendet werden. Die Adapter Schnittstelle zur Open Banking API ist so aufgebaut, sodass sie einfach durch eine Open Banking Schnittstelle einer anderen Bank ersetzt werden kann. Dies wird mithilfe von Interfaces realisiert. In diesem Kapitel wird die konkrete Implementierung zur Innofactory Open Banking Schnittstelle beleuchtet. Für die Schnittstelle ist eine Open API Dokumentation⁵¹ vorhanden. Alle Requests zur Schnittstelle erfolgen über: <https://inofwebdevapiopenpv.azurewebsites.net>. Da es sich hier um eine Testumgebung handelt, wurden gewisse Validierungsmechanismen ausgeschaltet

⁴⁷ <https://learn.microsoft.com/en-us/ef/core/>

⁴⁸ <https://www.mysql.com/>

⁴⁹ <https://www.mongodb.com/>

⁵⁰ <https://learn.microsoft.com/en-us/ef/core/providers/?tabs=dotnet-core-cli>

⁵¹ <https://inofwebdevapiopenpv.azurewebsites.net/swagger/index.html>

wie beispielsweise die Validierung der IBAN, Korrektheit der Referenznummer sowie das zwingende Befüllen von Pflichtfeldern.

3.4.1 Authentifizierung an der Schnittstelle

Um Daten mit der Schnittstelle auszutauschen, wird auf eine Authentifizierung mittels JWT⁵² Token gesetzt. Um einen solchen zu erhalten, muss über den folgenden Endpunkt ein Request erfolgen.

Endpunkt: /api/OpenBankingPayment/login

Methode: POST

Body:

```
1 {  
2   "ebankingContractNumber": "string",  
3   "password": "string"  
4 }
```

Abbildung 18 Open Banking Schnittstelle Request Authentifizierung

Die Antwort des Requests ist ein JWT-Token.

3.4.2 Transaktionen abrufen

Damit Transaktionen abgerufen werden können, stehen zwei Endpunkte zur Verfügung. Beim ersten Endpunkt kann eine Liste aller Transaktionen abgefragt werden.

Endpunkt: /api/OpenBankingPayment/transactions

Methode: GET

Response:

```
1 [  
2   {  
3     id: 'aed055c7-b790-4ef7-9c4e-ce7f3fac4f84',  
4     debtorAccount: 'CH1189144384113162191',  
5     amount: 20,  
6     date: '2025-04-09T12:08:07.0471124Z',  
7     type: 'Debit',  
8     status: 'ausgeführt',  
9     creditorAccount: 'CH2189144857391529527',  
10  }  
11 ]
```

Abbildung 19 Open Banking Schnittstelle Response Transaktionen

Die Empfangenen Daten der Schnittstelle beinhalten lediglich Informationen zur Sender- und Empfänger-IBAN, dem Transaktionsstatus, dem Betrag, dem Transaktionsdatum und dem Typ. Um weitere Informationen zur Transaktion zu erhalten, wie beispielsweise Empfänger und Senderadresse sowie der Referenznummer, welche für den Abgleich der Zahlungen essenziell ist, muss pro Transaktion ein weiterer Request an die Schnittstelle erfolgen.

Endpunkt: /api/OpenBankingPayment/paymentrequest?TransactionID=transaktionsId

Methode: GET

⁵² <https://jwt.io/>

Response:

```

1 {
2   "paymentRequests": [
3     {
4       "debtor": {
5         "account": {
6           "type": "",
7           "identifier": "",
8           "iban": "CH1189144384113162191"
9         }
10      },
11      "creditor": {
12        "account": {
13          "type": "",
14          "identifier": "",
15          "iban": "CH2189144857391529527"
16        },
17        "structuredAddress": {
18          "name": "Payment Solutions by Jan",
19          "country": "CH",
20          "townName": "Rapperswil-Jona",
21          "postCode": "8640",
22          "streetName": "Oberseestrasse 10",
23          "buildingNumber": ""
24        }
25      },
26      "ultimateDebtor": {
27        "structuredAddress": {
28          "name": "Max Mustermann",
29          "country": "CH",
30          "townName": "Appenzell",
31          "postCode": "9050",
32          "streetName": "Industriestrasse 12",
33          "buildingNumber": ""
34        }
35      },
36      "ultimateCreditor": {
37        "structuredAddress": {
38          "name": "",
39          "country": "",
40          "townName": "",
41          "postCode": "",
42          "streetName": "",
43          "buildingNumber": ""
44        }
45      },
46      "amount": {
47        "amountValue": "20",
48        "currency": "CHF"
49      },
50      "purposeCode": "",
51      "executionDate": "",
52      "remittanceReference": {
53        "type": "QRR",
54        "referenceNo": "202504091207391638419026569"
55      },
56      "endToEndId": "",
57      "initiatingParty": {
58        "id": ""
59      },
60      "remittanceInformation": "",
61      "transaction_ID": "aed055c7-b790-4ef7-9c4e-ce7f"
62    }
63  ]
64 }

```

Abbildung 20 Open Banking Schnittstelle Response Transaktion

Die Antwort des Requests enthält alle Informationen, welche zur Zahlung vorhanden sind. Von hier können Adressdaten von Sender und Empfänger (Debtor und Creditor) sowie die Referenznummer für die jeweiligen Transaktionen ermittelt werden. Alle Felder aus der Response, welche leer sind, werden nicht von der Schnittstelle benötigt. Sie werden jedoch aufgrund des DTO-Schemas der Open Banking Schnittstelle mitgeliefert (als leeren String).

3.4.3 Rückzahlungen erstellen

Für Rückzahlungen von fehlerhaften Zahlungen sowie der Auszahlung an einen Händler wird eine Schnittstelle für das Erstellen von Instant Payment Zahlungen benötigt. Für dies wird folgender Endpunkt verwendet. Um eine Zahlung erfassen zu können, welche später auch im Payment Scheme System gültig ist, benötigt es lediglich die gelb markierten Felder. Dies restlichen Felder müssen an die Open Banking Schnittstelle geliefert werden, können jedoch einen leeren String als Wert enthalten.

Endpunkt: /api/OpenBankingPayment/create-instantpayment

Methode: POST

Request:

```

1 {
2   "debtor": {
3     "account": {
4       "type": "string",
5       "identifier": "string",
6       "iban": "string"
7     }
8   },
9   "creditor": {
10    "account": {
11      "type": "string",
12      "identifier": "string",
13      "iban": "string"
14    },
15    "structuredAddress": {
16      "name": "string",
17      "country": "string",
18      "townName": "string",
19      "postCode": "string",
20      "streetName": "string",
21      "buildingNumber": "string"
22    }
23  },
24  "ultimateDebtor": {
25    "structuredAddress": {
26      "name": "string",
27      "country": "string",
28      "townName": "string",
29      "postCode": "string",
30      "streetName": "string",
31      "buildingNumber": "string"
32    }
33  },
34  "ultimateCreditor": {
35    "structuredAddress": {
36      "name": "string",
37      "country": "string",
38      "townName": "string",
39      "postCode": "string",
40      "streetName": "string",
41      "buildingNumber": "string"
42    }
43  },
44  "amount": {
45    "amountValue": "string",
46    "currency": "string"
47  },
48  "purposeCode": "string",
49  "executionDate": "string",
50  "remittanceReference": {
51    "type": "string",
52    "referenceNo": "string"
53  },
54  "endToEndId": "string",
55  "initiatingParty": {
56    "id": "string"
57  },
58  "remittanceInformation": "string",
59  "transaction_ID": "string"
60 }

```

Abbildung 21 Open Banking Schnittstelle Request Zahlungserfassung

Als Antwort wird die ID der erstellten Transaktion sowie deren aktuellen Status zurückgegeben. Der Status einer Transaktion ist initial „Eingereicht“ und wechselt nach ein paar Sekunden zu „ausgeführt“, sofern die Zahlung ausgeführt werden konnte. Im Falle eines Fehlers erhält die Zahlung den Status „fehlerhaft“. Die aktualisierten Status müssen nach ein paar Sekunden über `/api/OpenBankingPayment/transactions` oder `/api/OpenBankingPayment/paymentrequest` abgerufen werden.

4. Umsetzung

In diesem Kapitel wird auf die Implementierung der verschiedenen Features der Zahlungsplattform eingegangen.

4.1 Verwendete Tools und Libraries

Für die Implementierung der Software wurden verschiedene Technologien, Tools und Libraries verwendet. Im nachfolgenden Abschnitt eine Übersicht der relevanten Tools mit jeweiliger Beschreibung. Verwendet

wurden .NET⁵³, Angular⁵⁴, Automapper⁵⁵, Chat GPT⁵⁶, Entity Framework Core⁵⁷, Git⁵⁸, GitHub⁵⁹, GitHub Actions⁶⁰, GitHub Copilot⁶¹, Jest⁶², Hangfire⁶³, Lucide-react⁶⁴, MSTest⁶⁵, Next.js⁶⁶, Open API⁶⁷, pdf-lib⁶⁸, pdfkit⁶⁹, Postgres⁷⁰, Redis⁷¹, Scalar.AspNetCore⁷², Shadcn/ui⁷³, SonarAnalyzer.CSharp⁷⁴, SpartanUI⁷⁵, swissqrbill⁷⁶, Tailwind CSS⁷⁷, QuickStack⁷⁸, zod⁷⁹

Tool	Beschreibung
.NET	Eine von Microsoft entwickelte Open-Source-Entwicklungsplattform zur Erstellung verschiedenster Anwendungen.
Angular	Ein von Google entwickeltes TypeScript-basiertes Frontend-Framework zur Erstellung von Webanwendungen mit komponentenbasierter Architektur.
Automapper	Eine .NET-Bibliothek, die hilft, Objekte zwischen verschiedenen Schichten (z. B. DTOs und Domänenobjekte) automatisch zu mappen.
Chat GPT	Ein KI-Chatbot, der mit dem Nutzer über textbasierte Nachrichten und Bilder ähnlich wie mit einem Menschen kommunizieren kann.
Entity Framework Core	Ein ORM (Object-Relational Mapper) für .NET, der die Interaktion mit Datenbanken über .NET-Klassen erlaubt und SQL automatisch generiert.
Git	Ein Versionskontrollsystem, das es Entwicklern unter anderem ermöglicht, Änderungen im Quellcode nachzuverfolgen.
GitHub	Eine Plattform für Versionskontrolle und kollaborative Softwareentwicklung basierend auf Git, mit zusätzlichen CI/CD Features.
GitHub Actions	CI/CD-Tool, mit dem sich automatisierte Workflows direkt in GitHub erstellen und ausführen lassen.
GitHub Copilot	Ein KI-basierter Codeassistent, der auf maschinellem Lernen basiert und Entwicklern Codevorschläge in Echtzeit macht.
Jest	JavaScript Testing Framework, welches für eine breite Palette von Frontend-Frameworks verwendet werden kann.

⁵³ <https://dotnet.microsoft.com/en-us/>

⁵⁴ <https://angular.dev/>

⁵⁵ <https://automapper.org/>

⁵⁶ <https://openai.com/de-DE/chatgpt/overview/>

⁵⁷ <https://learn.microsoft.com/en-us/ef/core/>

⁵⁸ <https://git-scm.com/>

⁵⁹ <https://github.com>

⁶⁰ <https://github.com/features/actions>

⁶¹ <https://github.com/features/copilot>

⁶² <https://jestjs.io/>

⁶³ <https://www.hangfire.io/>

⁶⁴ <https://lucide.dev/>

⁶⁵ <https://learn.microsoft.com/de-de/dotnet/core/testing/unit-testing-mstest-intro>

⁶⁶ <https://nextjs.org/>

⁶⁷ <https://www.openapis.org/>

⁶⁸ <https://pdf-lib.js.org/>

⁶⁹ <https://pdfkit.org/>

⁷⁰ <https://www.postgresql.org/>

⁷¹ <https://redis.io/>

⁷² <https://www.nuget.org/packages/Scalar.AspNetCore/>

⁷³ <https://ui.shadcn.com/>

⁷⁴ <https://www.nuget.org/packages/sonaranalyzer.csharp/>

⁷⁵ <https://spartan.ng/>

⁷⁶ <https://github.com/schoero/swissqrbill>

⁷⁷ <https://tailwindcss.com/>

⁷⁸ <https://quickstack.dev>

⁷⁹ <https://zod.dev/>

Hangfire	Eine .NET-Bibliothek zur Hintergrundverarbeitung von Aufgaben (z. B. geplante oder wiederkehrende Cron Jobs).
Lucide-react	Eine Sammlung von Open-Source-Icons für React-Anwendungen.
MSTest	Ein von Microsoft bereitgestelltes Framework für das Schreiben und Ausführen von Unit- und Integrationstests in .NET-Anwendungen.
Next.JS	Ein React-basiertes Web-Framework, das serverseitiges Rendering und statische Webseiten-Generierung unterstützt. Es vereinfacht die Entwicklung von Webanwendungen mit Funktionen wie Seitenrouting und API-Handling. Next.js verwendet standardmässig weitere externe Libraries, welche bei der Installation automatisch mitinstalliert werden. Diese werden hier nicht mitaufgelistet.
Open API	Eine standardisierte Spezifikation zur Beschreibung von RESTful APIs, die maschinenlesbare Dokumentation und Codegenerierung ermöglicht.
pdf-lib	Eine JavaScript-Bibliothek zur Manipulation und Erstellung von PDF-Dateien im Browser oder in Node.js.
pdfkit	Eine Node.js-Bibliothek zur Erstellung von PDF-Dokumenten (Server-Side only).
Postgres	Ein objektrelationales Open-Source-Datenbankmanagementsystem, bekannt für Stabilität, Erweiterbarkeit und SQL-Konformität.
Redis	Eine In-Memory-Datenstruktur-Datenbank, die als Cache, Message Broker oder temporäre Datenspeicherung verwendet wird.
Scalar.AspNetCore	Eine Middleware-Bibliothek zur einfachen Erstellung und Dokumentation von APIs mit OpenAPI-Spezifikation in ASP.NET Core.
Shadcn/ui	Eine auf Tailwind CSS basierende UI-Komponentenbibliothek, die anpassbare und einfach zu verwendende Komponenten für moderne React Web-Interfaces bereitstellt.
SonarAnalyzer.CSharp	Ein statisches Codeanalyse-Tool zur Qualitäts- und Sicherheitsüberprüfung von C#-Code.
SpartanUI	Eine UI-Komponentenbibliothek, welche shadcn/ui Komponenten für Angular verfügbar macht.
swissqrbill	Eine JavaScript-Bibliothek zur Generierung von QR-Rechnungen gemäss dem Schweizer Standard (Swiss QR Code).
Tailwind CSS	Ein CSS-Framework, das es ermöglicht, mit Hilfe von Klassen direkt im HTML individualisierte Layouts zu gestalten.
QuickStack	Open-Source Container-PaaS, mit dem Apps via Git oder Docker Image auf eigenen Servern betrieben werden können.
zod	Eine TypeScript-First-Bibliothek zur Validierung und Typisierung von Daten.

Tabelle 7 Verwendete Tools und Technologien

4.2 Grundlagen

Um zu verstehen, wie die Software aufgebaut ist, müssen vorerst ein paar Begriffe genauer erläutert werden.

Instant Pay: Der Name der entwickelten Software lautet «Instant Pay». Im weiteren Verlauf des Berichts werden einzelne Funktionen mit dem Präfix «Instant Pay XXX» erwähnt.

Payment Session: Eine Payment Session bezeichnet eine erwartete Zahlung mit einem zugewiesenen Betrag. Im Erfolgsfall hat eine Payment Session genau eine Transaktion. Beispielsweise will ein Kunde in einem Onlineshop eine Zahlung aufgeben. Der Onlineshop erstellt über eine REST-API eine Payment Session von 5.50 CHF und der Kunde erstellt eine Instant Payment Zahlung von 5.50 CHF. In diesem Fall wäre die Payment Session erfolgreich bezahlt. Eine Payment Session kann dann mehrere Transaktionen beinhalten, wenn ein Betrag zurückerstattet wurde. Dies kann der Fall sein, wenn beispielsweise die zahlende Person einen anderen Betrag als erwartet überweist, mehrere Transaktionen überweist oder der Händler eine manuelle Rückerstattung erfasst.

Kasse: Mithilfe von Kassen können Payment Sessions kategorisiert werden. Die Verknüpfung von Kassen mit einer Payment Session ist optional. Das bedeutet, dass Payment Sessions auch ohne Zuweisung einer Kasse erstellt werden können (z.B. via REST-API).

Handlerportal: Das Händlerportal ist eine Oberfläche auf welchem Händler alle Payment Sessions und Transaktionen einsehen können sowie Kassen, API-Keys und Profilinformationen bearbeiten können.

4.3 Händlerportal

Das Händlerportal ist der Dreh und Angelpunkt für Händler. Händler erhalten hier eine Übersicht über alle Payment Sessions, können Exporte erstellen, mithilfe von Instant Pay Now einkassieren, API-Tokens für die Anbindung von Webshops erstellen und QR-Codes für Instant Pay Local erstellen. Das Händlerportal bildet eine Self-Service-Plattform wodurch der Händler alle Aspekte rund ums Einkassieren mit Instant Payment selbst erledigen kann. Das Portal ist unter <https://portal.instant-pay.ch> verfügbar.

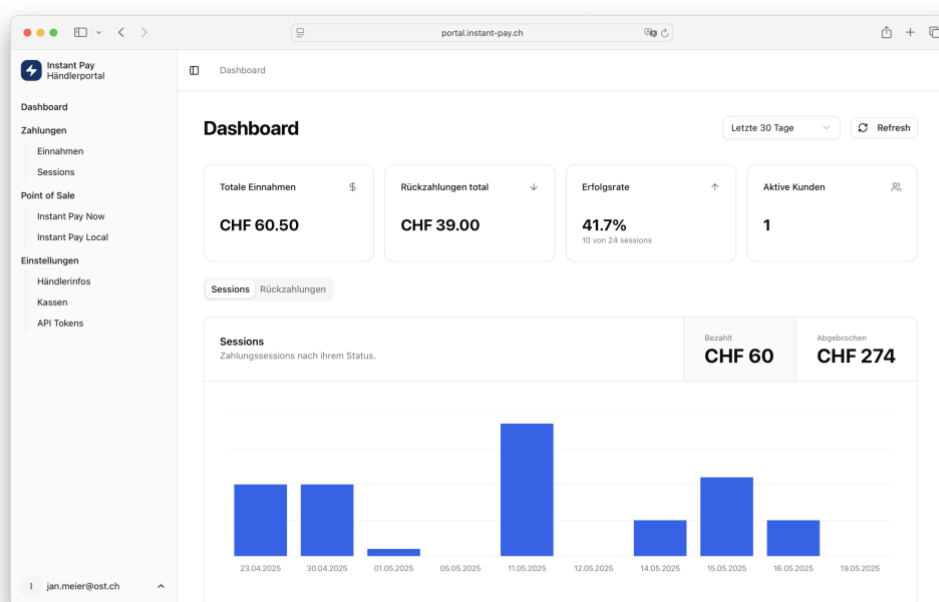


Abbildung 22 Screenshot Händlerportal Dashboard

4.4 Payment Session

Payment Sessions können aus drei verschiedenen Quellen erstellt werden: Direkt durch das Händlerportal (Instant Pay Now), durch einen QR-Sticker (Instant Pay Local) oder durch die REST-API (Instant Pay Web).

4.4.1 Instant Pay Local

Bei Instant Pay Local wird die Transaktion vor der Payment Session erstellt. Instant Pay Local ist geeignet für Einsatzgebiete, bei welchen keine Kassen (und je nach dem kein Personal) eingesetzt wird. Die zahlende

Person scannt mit der E-Banking App der Hausbank einen aufliegenden QR-Rechnung Code und kann anschliessend den Betrag selbst definieren und die Zahlung ausführen.



Abbildung 23 Beispiel eines QR-Stickers von Instant Pay Local

Die Zahlung wird innerhalb von fünf Minuten durch einen Abgleich mit der Open Banking Schnittstelle der Bank ins System importiert. Erkennbar sind Instant Pay Local Zahlungen an dem Präfix «0611» in der Referenznummer.

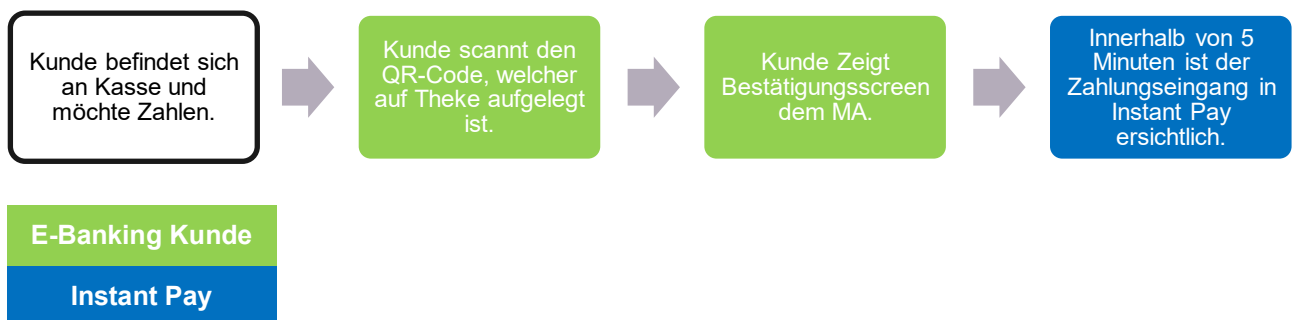


Abbildung 24 Bezahlprozess Instant Pay Local

4.4.2 Instant Pay Now

Instant Pay Now kann an bedienten Kassen eingesetzt werden. Im Gegensatz zu Instant Pay Local kann hier direkt überprüft werden, ob die Zahlung mit dem korrekten Betrag der zahlenden Person eingegangen ist oder nicht.

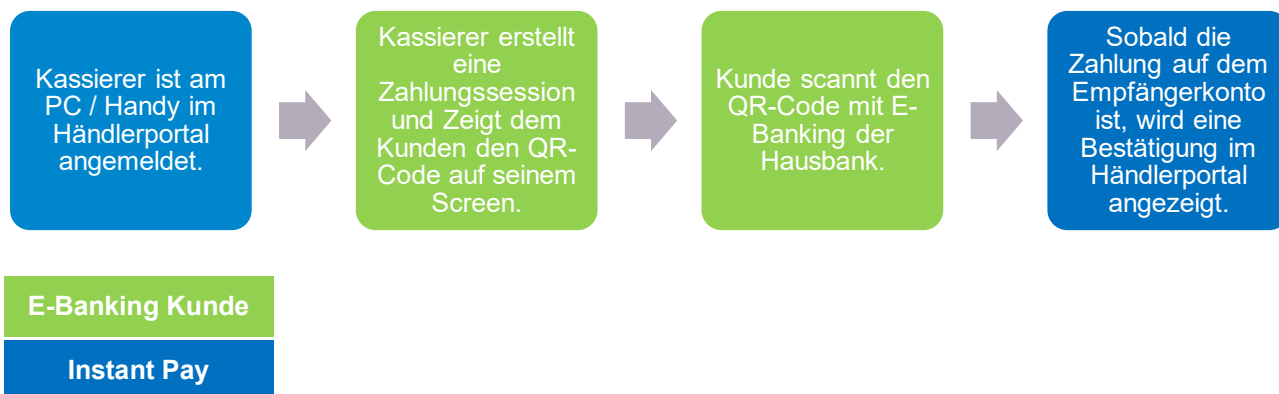
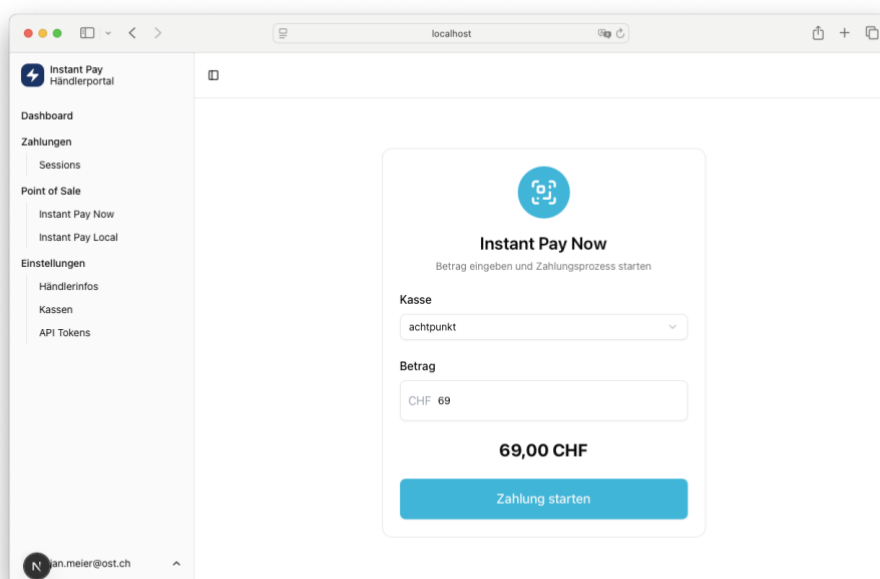


Abbildung 25 Übersicht Bezahlprozess Instant Pay Now

Mitarbeiter am Point of Sale können entweder via Smartphone oder PC einkassieren. Die Person, welche einkassiert, ist im Händlerportal angemeldet und kann über die Instant Pay Now Schaltfläche einen Betrag und optional eine Kasse wählen. Im Anschluss muss «Zahlung starten» gewählt werden.



The screenshot shows a web browser window with the URL 'localhost'. The page is titled 'Instant Pay Händlerportal'. On the left, there is a sidebar menu with the following items: Dashboard, Zahlungen (with a sub-item Sessions), Point of Sale (with sub-items Instant Pay Now and Instant Pay Local), and Einstellungen (with sub-items Händlerinfos, Kassen, and API Tokens). The main content area is titled 'Instant Pay Now' and contains the instruction 'Betrag eingeben und Zahlungsprozess starten'. It features a dropdown menu for 'Kasse' with 'achtpunkt' selected, a text input field for 'Betrag' containing 'CHF 69', and a large blue button labeled 'Zahlung starten'. The amount '69,00 CHF' is displayed below the input field. At the bottom left of the browser window, the user's email 'jan.meier@ost.ch' is visible.

Abbildung 26 Instant Pay Now Eingabemaske

Sobald der Zahlprozess startet, wird auf dem Bildschirm die Checkout-Seite angezeigt, welche von der zahlenden Person mit der E-Banking App gescannt werden kann.

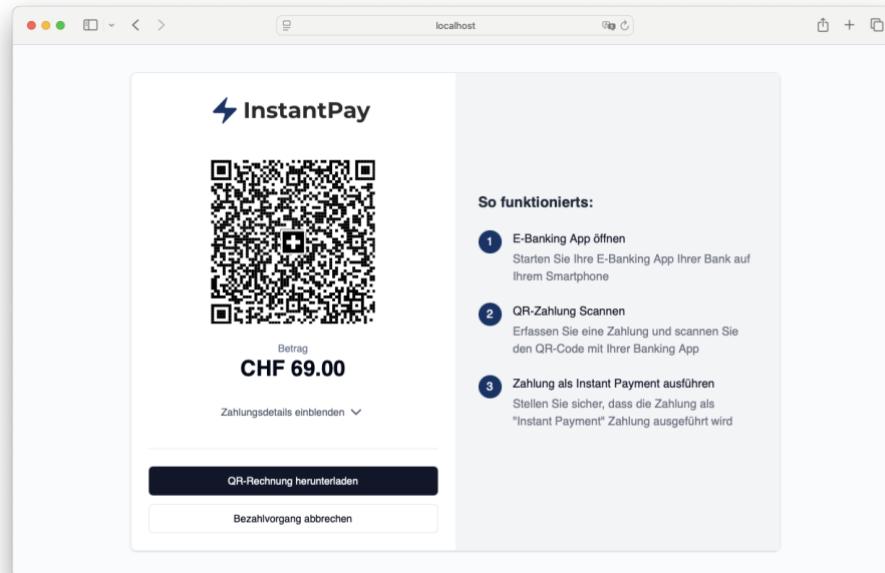


Abbildung 27 Instant Pay Checkout Seite

Nach erfolgreichem Zahlungseingang wird auf dem Bildschirm des Händlers angezeigt, ob die Zahlung erfolgreich war oder nicht. Falls die Zahlung nicht erfolgreich abgeschlossen werden konnte, wird zusätzlich ein Grund angegeben.

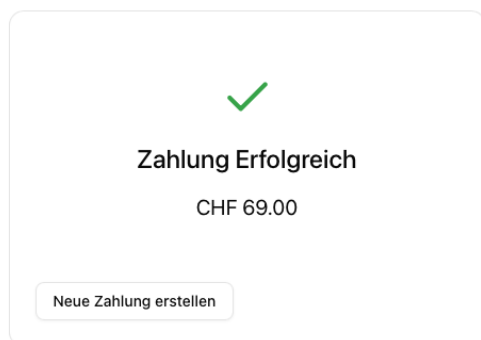


Abbildung 28 Bestätigungsseite Zahlung Instant Pay Now



Abbildung 29 Fehlgeschlagene Zahlung Instant Pay Now

4.4.3 Instant Pay Web

Instant Pay Web wird für Zahlungen in Onlineshops verwendet. Ein Webshop kann über eine REST-API zum Instant Pay Backend eine Payment Session erstellen und deren Status abfragen. Für den Kunden des Webshops sieht der Bezahlprozess wie folgt aus:

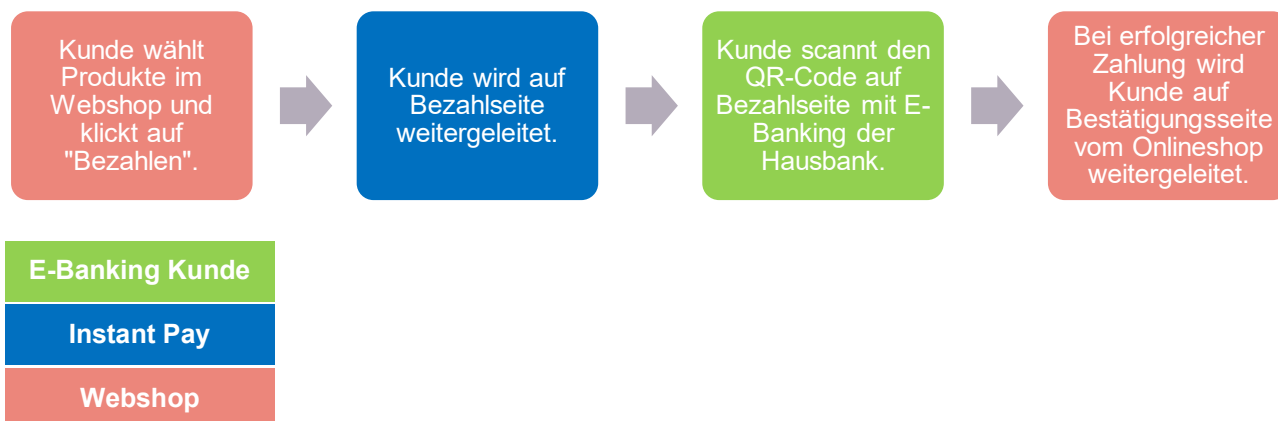


Abbildung 30 Bezahlprozess Instant Pay Web

Für den Bezahler ist der Bezahlprozess nahtlos integriert und dieser bekommt nichts von dem Wechsel zwischen den verschiedenen Systemen mit. Von der Bestellseite des Webshops wird der Kunde direkt auf

die Instant Pay Checkout Seite weitergeleitet (Schritt 2) und bei erfolgreichem Bezahlvorgang wird der Kunde auf die Bestätigungsseite des Webshops weitergeleitet (Schritt 3).

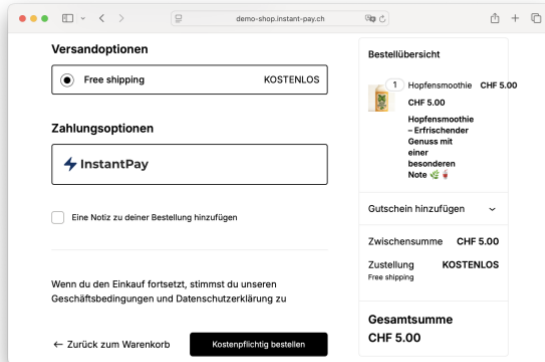


Abbildung 31 Instant Pay Web Bezahlprozessschritt 1 (Webshop)

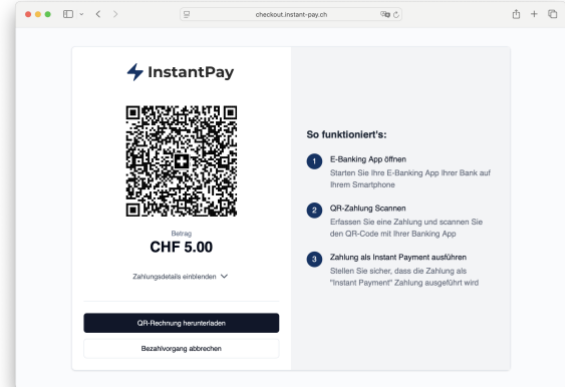


Abbildung 32 Instant Pay Web Bezahlprozessschritt 2 (Instant Pay)

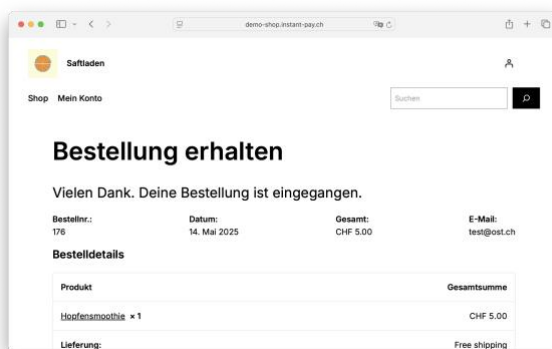


Abbildung 33 Instant Pay Web Bezahlprozessschritt 3 (Webshop)

4.4.4 Schnittstelle Instant Pay Web

Für die Kommunikation zwischen dem Webshop und dem Instant Pay Backend werden die folgenden zwei Schnittstellen verwendet. Die erste Schnittstelle wird für das Erstellen einer Payment Session angesprochen. Neben dem Betrag übergibt der Webshop Instant Pay eine URL für die Bestätigungsseite (bei erfolgreicher Zahlung) und eine URL einer Fehlerseite (falls die Zahlung fehlschlägt). Als Antwort erhält der Webshop eine URL, auf welche er den Bezahler für den Bezahlprozess weiterleiten muss.

Endpunkt: /api/ PaymentSession/create

Methode: POST

Body:

```
1 {  
2   "amount": 49.95,  
3   "successRedirectUrl": "https://example.com/success",  
4   "errorRedirectUrl": "https://example.com/error"  
5 }
```

Abbildung 34 Request erstellen Payment Session über REST API

Response:

```
1 {  
2   "paymentSessionId": "d290f1ee-6c54-4b01-90e6-d701748f0851",  
3   "paymentSessionUrl": "https://checkout.instant-pay.ch/session?id=1234"  
4 }
```

Abbildung 35 Response erstellen Payment Session über REST API

Sobald die Zahlung abgeschlossen wurde, kann der Webshop anhand der paymentSessionId den aktuellen Status der Payment Session ermitteln.

Endpunkt: /api/ PaymentSession/{paymentSessionId}

Methode: GET

Response:

```

1 {
2   "id": "0196cf0e-9c5a-7e8e-b240-e60666c24e0c",
3   "paymentReference": "202505141349237313842791150",
4   "amount": 69,
5   "successRedirectUrl": "http://localhost:3000/dashboard/point-of-sale/payment-success?sid=0196cf0e-9c5a-7e8e-b240-e60666c24e0c",
6   "errorRedirectUrl": "http://localhost:3000/dashboard/point-of-sale/payment-cancel?sid=0196cf0e-9c5a-7e8e-b240-e60666c24e0c",
7   "cancellationReason": null,
8   "cashboxId": null,
9   "source": 2,
10  "status": 3,
11  "merchantId": "0195d81c-e610-76d8-8750-6b2e0fd44721",
12  "paymentTransactions": [
13    {
14      "id": "10dc5ac6-5d08-4ae0-8e40-465b35b5a32e",
15      "externalTransactionId": "3548d2fb-6b93-4b45-b15f-ad0e00693797",
16      "amount": 69,
17      "status": 3,
18      "type": 0,
19      "paymentReference": "202505141349237313842791150",
20      "paymentSessionId": "0196cf0e-9c5a-7e8e-b240-e60666c24e0c",
21      "debtor": {
22        "paymentTransactionId": "10dc5ac6-5d08-4ae0-8e40-465b35b5a32e",
23        "bankAccountOwner": "Max Mustermann",
24        "iban": "CH1189144384113162191",
25        "postalCode": "9050",
26        "townName": "Appenzell",
27        "createdAt": "2025-05-14T13:52:48.259841Z",
28        "updatedAt": "2025-05-14T13:52:48.259841Z"
29      },
30      "creditor": {
31        "paymentTransactionId": "10dc5ac6-5d08-4ae0-8e40-465b35b5a32e",
32        "bankAccountOwner": "Sepp Moserr",
33        "iban": "CH4889144496154856756",
34        "postalCode": "9050",
35        "townName": "Appenzell",
36        "createdAt": "2025-05-14T13:52:48.259841Z",
37        "updatedAt": "2025-05-14T13:52:48.259841Z"
38      },
39      "originalTransactionId": null,
40      "refundTransaction": null,
41      "createdAt": "2025-05-14T13:52:48.259841Z",
42      "updatedAt": "2025-05-14T13:52:48.259842Z"
43    }
44  ],
45  "createdAt": "2025-05-14T13:49:23.418486Z",
46  "updatedAt": "2025-05-14T13:52:48.259842Z"
47 }

```

Abbildung 36 Response Status Payment Session REST API

Für die Verbindung mit dem Instant Pay Backend muss der Webshop für jeden Request an die REST API einen JWT-Token im Authentication Header mitgeben. REST API Tokens können im Händlerportal ausgestellt und bei Kompromittierung invalidiert werden. Der JWT-Token wird nach der Erstellung im Händlerportal nur einmal angezeigt und von Instant Pay nicht persistiert. Anhand des Claims «role» und dem «nameidentifier» im Payload des JWT-Tokens kann überprüft werden, ob der JWT-Token noch aktiv ist und zu welchem Händler der Token zugewiesen ist. REST API Tokens haben jeweils die Rolle «API_TOKEN» und als «nameidentifier» die ID des Datenbankeintrags des API-Tokens in der Tabelle «ApiToken». In der

Darstellung der verschiedenen API-Calls zwischen dem Händlerportal oder Webshop, der Instant Pay Checkout Seite und dem Instant Pay Backend.

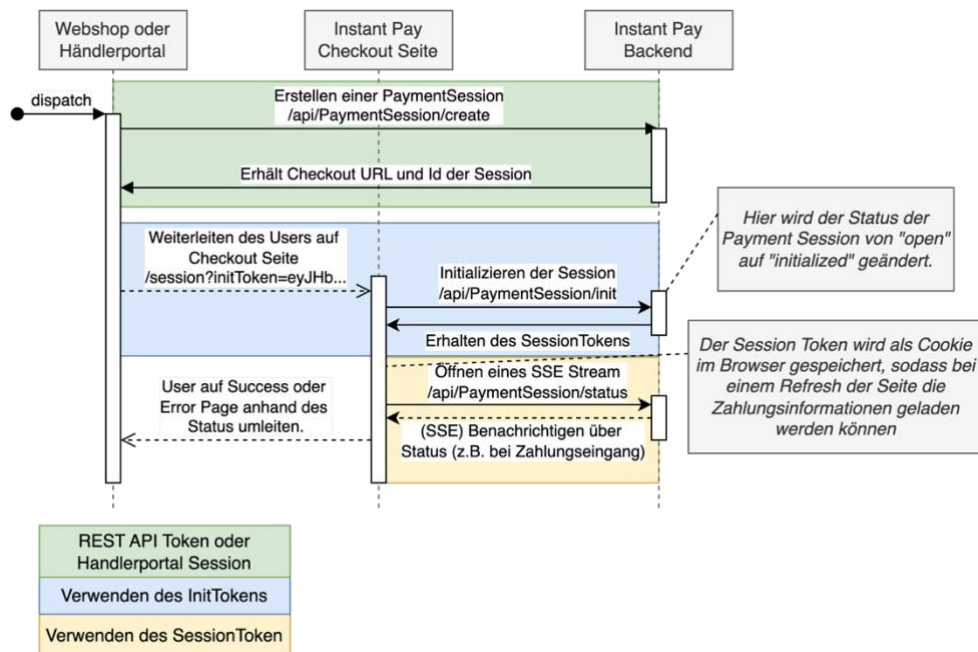


Abbildung 38 Sequenzdiagramm JWT Token während Zahlungssession

4.6 Transaktionsabgleich mit Open Banking

Da die Open Banking Schnittstelle das Instant Pay Backend nicht direkt benachrichtigen kann, müssen die aktuellen Transaktionen jeweils bei der Instant Payment Schnittstelle abgefragt werden (polling⁸⁰). Für die Implementierung der regelmässigen Abfrage von Transaktionsinformationen wurden drei Cron Jobs⁸¹ implementiert.

- **RunOutgoingPaymentSyncJob** | alle 10 Minuten: Sendet ausgehende Zahlungen an die Open Banking Schnittstelle, welche den Status «created» haben (wurden erstellt, jedoch noch nicht an die Open Banking Schnittstelle gesendet. Im Anschluss wird der Status von gesendeten Zahlungen mit dem Status der Transaktion in der Instant Pay Datenbank abgeglichen.
- **RunLocalPaymentSynchronizationJob** | alle 5 Minuten: Lädt alle Transaktionen von der Open Banking Schnittstelle, welche eine Instant Pay Local Zahlung sind (Referenznummer mit Präfix 0611) und gleicht Sie mit den Transaktionen in der Instant Pay Datenbank ab.
- **RunPaymentSynchronizationJob** | alle 15 Sekunden: Lädt alle Transaktionen (ausgenommen Instant Pay Local Transaktionen) von der Open Banking Schnittstelle und gleicht sie mit den Transaktionen in der Instant Pay Datenbank ab. Der Request zur Open Banking Schnittstelle wird nur ausgeführt, wenn aktuell offene Instant Pay Now oder Instant Pay Web Payment Sessions bestehen. Er wird zudem alle zehn Minuten ausgeführt, um unerwartete Zahlungseingänge auf das Empfängerkonto abzufangen und für diese Rückzahlungen zu erstellen.

⁸⁰ [https://de.wikipedia.org/wiki/Polling_\(Informatik\)](https://de.wikipedia.org/wiki/Polling_(Informatik))

⁸¹ <https://de.wikipedia.org/wiki/Cron>

Damit bei mehreren Backend Instanzen die Cron Jobs nicht doppelt ausgeführt werden, setzt das Backend auf die Hangfire⁸² Library. Hangfire plant und verwaltet Cron Jobs in einem separaten Datenbankschema der Postgres Datenbank und stellt durch Datenbank Locks sicher, dass ein Cron Job nur auf einer Backend Instanz ausgeführt wird.

4.7 Checkout Seite

Die Checkout Seite von Instant Pay hat zwei zentrale Aufgaben. Als erstes soll dem Benutzer ein Swiss QR-Code angezeigt werden. Mit diesem soll der Benutzer die Zahlung ausführen können, indem er diesen mit der E-Banking App scannt und die Zahlung ausführt. Die Zweite zentrale Aufgabe der Checkout Seite ist das Umleiten des Benutzers auf die Bestätigungsseite (z.B. die eines Onlineshops) oder im Falle einer fehlgeschlagenen Transaktion die Weiterleitung auf eine externe Fehlerseite (z.B. die eines Onlineshops).

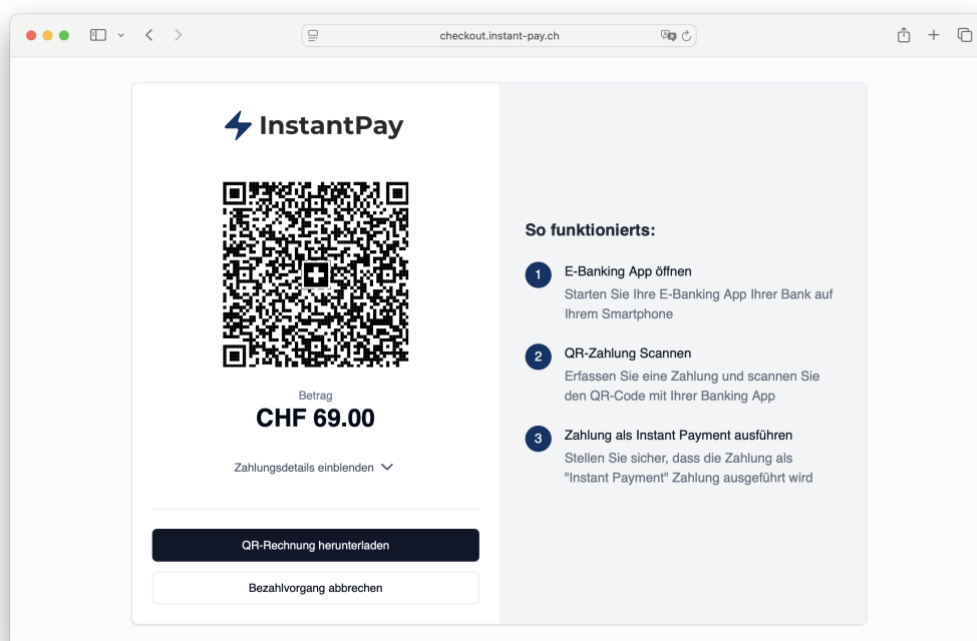


Abbildung 39 Screenshot Instant Pay Checkout Seite

Neben den zwei zentralen Aspekten spielt die Performance der Seite ebenfalls eine Rolle. Um eine optimale Performance zu gewährleisten, werden zwischen dem Backend und der Bezahlseite nur JSON-Daten ausgetauscht. Der QR-Code sowie das PDF der QR-Rechnung wird auf der Clientseite generiert, damit keine unnötigen Serverressourcen für die Generierung von Bildern und PDFs verschwendet wird.

Sobald die Seite geladen ist, wird ein Event-Stream für Server-Sent-Events (SSE)⁸³ zum Backend geöffnet. Über diesen benachrichtigt das Backend die Checkout Seite über Statusänderungen der Payment Session (erfolgreich oder abgebrochen). Die an die Checkoutseite gesendeten Events werden durch den Cron Job «RunPaymentSynchronizationJob» erstellt, sobald eine eingehende Zahlung mit der Open Banking Schnittstelle passend zur Payment Session der Checkout Seite erkannt wurde.

⁸² <https://www.hangfire.io>

⁸³ https://developer.mozilla.org/de/docs/Web/API/Server-sent_events/Using_server-sent_events

Auf einem Smartphone wird dem Benutzer auf der Checkout Seite der Button «In Mobile-Banking öffnen» angezeigt. Mit Klick auf diesen wird ein PDF einer QR-Rechnung (der gesamte Bezahlteil) erstellt und über die share()⁸⁴ Methode der native «Teilen-Dialog» des Smartphones geöffnet. Einige Banken Apps bieten die Option an, ein PDF direkt in der Mobile Banking App zu öffnen und direkt eine Zahlung zu erfassen, ohne dass der QR-Code abgescannt werden muss. So auch im untenstehenden Beispiel die Banking App «Yuh»⁸⁵. Durch Klick auf «Eine QR-Rechnung bezahlen» (orange markiert) wird die Yuh App geöffnet und die Bezahlmaske mit den Daten der Instant Payment Zahlung vorabgefüllt.

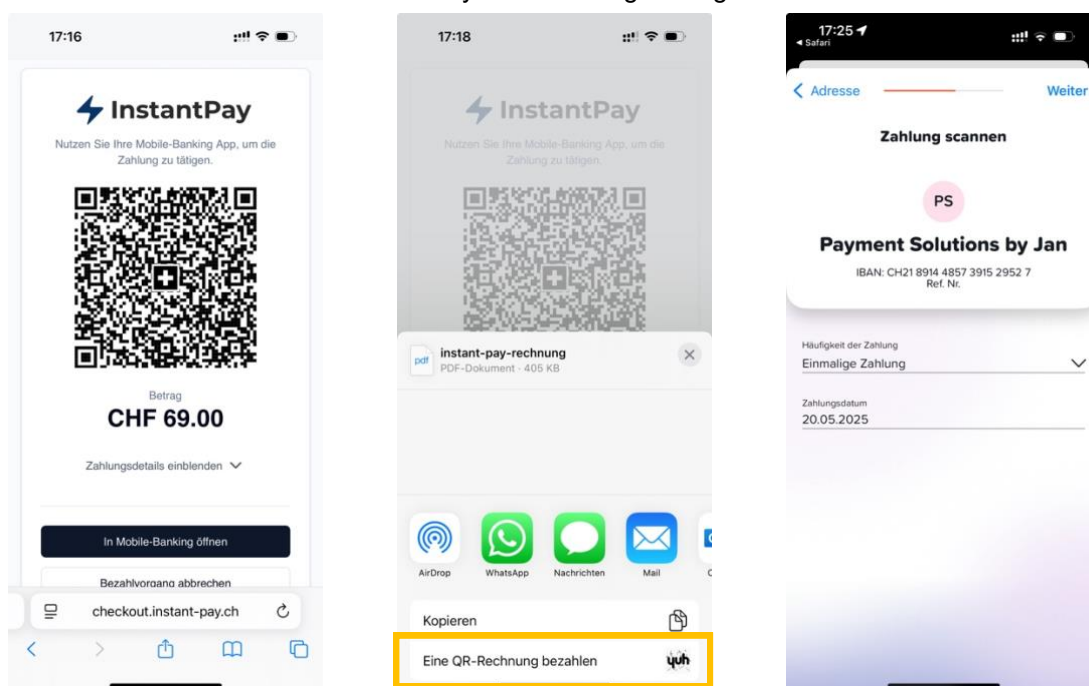


Abbildung 40 Instant Pay Zahlung in Mobile Banking importieren

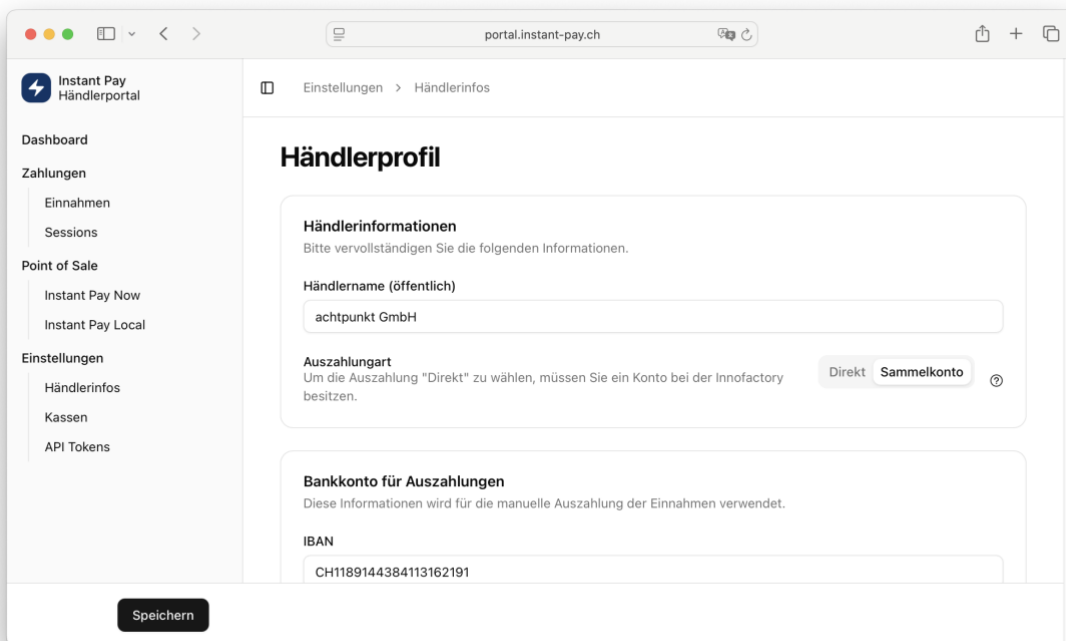
4.8 Verschiedene Auszahlungsarten

Wie im Variantenentscheid beschrieben, kann Instant Pay Zahlungen direkt auf das Empfängerkonto des Händlers senden (nachfolgend «direkte Auszahlung») oder mit einem Zwischenkonto (nachfolgend «Variante Sammelkonto») arbeiten, von welchem aus manuell Auszahlungen an das Händlerkonto gemacht

⁸⁴ <https://developer.mozilla.org/de/docs/Web/API/Navigator/share>

⁸⁵ <https://www.yuh.com/de/>

werden können. Ein Händler kann sich jeweils nur für eine Art der Auszahlung entscheiden. Im Händlerportal unter «Händlerinfos» kann dies festgelegt werden.



portal.instant-pay.ch

Instant Pay
Händlerportal

Einstellungen > Händlerinfos

Händlerprofil

Händlerinformationen
Bitte vervollständigen Sie die folgenden Informationen.

Händlername (öffentlich)
achtpunkt GmbH

Auszahlungsart
Um die Auszahlung "Direkt" zu wählen, müssen Sie ein Konto bei der Innofactory besitzen.

Direkt **Sammelkonto** ⓘ

Bankkonto für Auszahlungen
Diese Informationen wird für die manuelle Auszahlung der Einnahmen verwendet.

IBAN
CH1189144384113162191

Speichern

Abbildung 41 Händlerinfos Auszahlungsarten

Falls die Auszahlungsart «direkte Auszahlung» gewählt ist, müssen die Anmeldedaten des Innofactory Kontos des Händlers angegeben werden. In Kombination mit einer echten Open-Banking-Schnittstelle ist es erforderlich, dass der Händler in seinem E-Banking eine entsprechende Freigabe erteilt, damit Instant Pay auf die Schnittstelle zugreifen darf.

Im Falle der Auszahlungsart «Sammelkonto» würde das Konto mit der IBAN CH2189144857391529527 als Empfängerkonto für Zahlungen verwendet werden. Die Anmeldedaten für die Interaktion mit der Open Banking Schnittstelle werden über Environment Variablen dem Backend zur Verfügung gestellt. Der Händler kann eine Auszahlung auf das Bankkonto des Händlers starten indem unter dem Menüpunkt «Einnahmen»

eine Auszahlung erstellt wird. Im Hintergrund wird eine ausgehende Transaktion auf dem Sammelkonto erstellt und eine Bankzahlung über die Open Banking Schnittstelle erstellt.

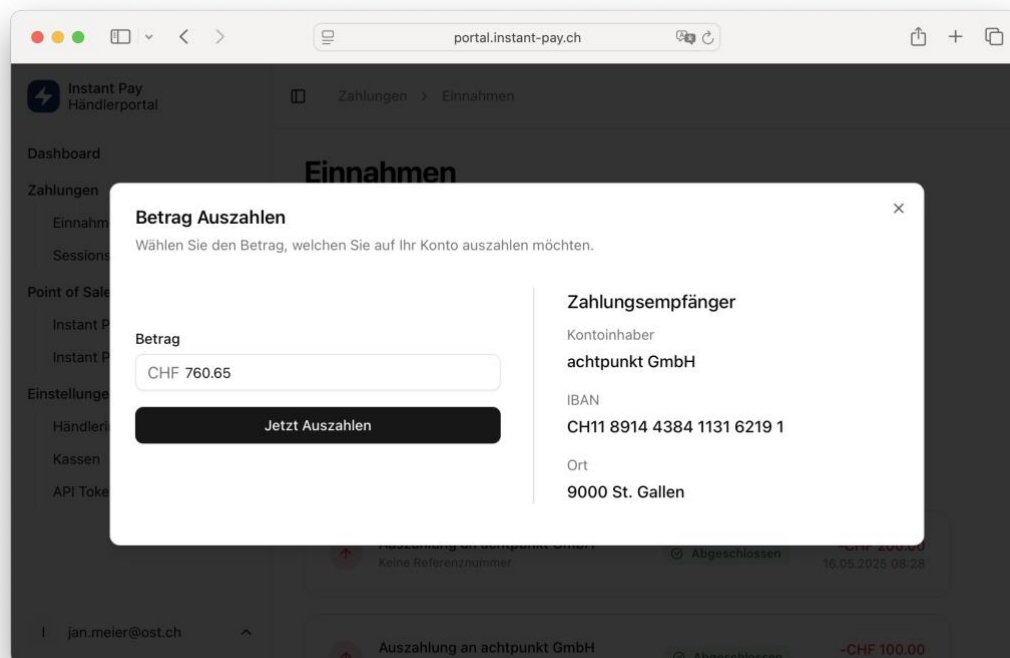


Abbildung 42 Overlay Sammelkonto Auszahlung

In den automatisierten Cron-Jobs zum Abgleich mit der Open-Banking-Schnittstelle werden die Transaktionen aller Händler auf dem Sammelkonto gleichzeitig verarbeitet. Der Abgleich der Transaktionen der Händlerkonten mit «direkter Auszahlung» erfolgt nacheinander.

4.9 Error Handling Backend

Um sicherzustellen, dass auch in einem Fehlerfall dem aufrufenden System eine standardisierte Antwort gegeben werden kann, verwendet das Instant Pay Backend neben Statuscodes den RFC 7807 Standard (Problem Details for HTTP APIs). Dieser Standard regelt, dass in einem Fehlerfall standardisierte Antworten im Body einer Response zurückgegeben werden. Solche standardisierten Fehlermeldungen können anhand des Content-Type «application/problem+json» erkannt werden. Ein Beispiel einer solchen Antwort sieht wie folgt aus:

```
1 HTTP/1.1 403 Forbidden
2 Content-Type: application/problem+json
3
4 {
5   "type": "https://api.instant-pay.ch/api/PaymentSession/asdf",
6   "title": "You are not permitted for this action."
7 }
```

Abbildung 43 Beispiel application/problem+json

4.10 Testing

Um die korrekte Funktion und das Verständnis aller Instant Payment-Funktionen zu gewährleisten, werden manuelle Tests, automatisierte Tests und Usability-Tests durchgeführt.

4.10.1 Manuelle Tests

Nachfolgend sind alle Testfälle aufgelistet mit jeweiliger Beschreibung, den auszuführenden Schritten und dem erwarteten Resultat. Bei einem Testdurchlauf muss die Reihenfolge eingehalten werden. Als E-Banking wird die Web-App unter <https://scanner.instant-pay.ch> bezeichnet, mit welcher Zahlungen analog zu einer echten E-Banking App gescannt und ausgeführt werden können.

Nr.	Beschreibung	Schritte	Erwartetes Resultat
1	Registrieren eines neuen Händlers	1. Navigiere zur Login Seite der Applikation. 2. Gib test2@ost.ch und OST.1234 als Benutzernamen und Passwort ein.	Das Benutzerkonto wurde erstellt und der Benutzer wurde auf die Einstellungsseite des Händlerportals weitergeleitet, um die Details des Händlers abzufüllen.
2	Eingabe der Händlerinfos	1. Navigiere zu den Händlerinfos in den Einstellungen 2. Fülle die folgenden Daten im Formular ab - Händlername: achtpunkt GmbH - IBAN: CH1189144384113162191 - Auszahlungsart: Sammelkonto - Inhaber Bankkonto: achtpunkt GmbH - Adresse: Oberer Graben 8 - Postleitzahl: 9000 - Ort: St. Gallen 3. Speichere die Eingaben	Die Angaben wurden korrekt gespeichert und werden bei einem Refresh der Seite wieder abgefüllt.
3	Erstellen einer Instant Pay Now Payment Session	1. Navigiere zu «Instant Pay Now» 2. Lasse das Kassenfeld auf «Keine Kasse ausgewählt» 3. Gib den Betrag 10 CHF ein	Die Zahlungssession wurde erstellt und auf dem Bildschirm wird die Checkout Seite mit 10 CHF angezeigt.
4	Bezahlen einer Instant Pay Now Payment Session	1. Erstelle durch Testfall 3 eine Payment Session 2. Die QR-Rechnung wird mithilfe der E-Banking App gescannt (10 CHF)	Der Benutzer wird innerhalb von 30 Sekunden auf die Bestätigungsseite im Händlerportal weitergeleitet.
5	Anzeigen der Payment Session Details	1. Navigiere zu «Sessions» 2. Klicke auf die Payment Session mit 10 CHF	Die Detailseite der Transaktionen wird mit dem Betrag 10 CHF angezeigt. Es ist nur eine Transaktion mit dem Status «Abgeschlossen» aufgelistet.
6	Download eines Zahlungsbelegs	1. Klicke auf der Detailseite der Payment Session auf «Zahlungsbeleg» 2. Öffne die heruntergeladene Datei	Es wird ein PDF mit der Überschrift «Zahlungsbeleg» angezeigt. Die Transaktion sowie Zahlungsreferenz und erwarteter Betrag stimmen im PDF mit der Detailseite der Payment Session überein.
7	Herunterladen einer QR-Rechnung auf der Checkout Seite	1. Erstelle durch Testfall 3 eine Payment Session auf einem PC 2. Klicke auf «QR-Rechnung herunterladen» 3. Öffne die heruntergeladene Datei	Ein PDF wird angezeigt mit einem vollständigen Zahlteil einer QR-Rechnung. Der Betrag beläuft sich auf 10 CHF.
8	Abbrechen einer Checkout Seite	1. Erstelle durch Testfall 3 eine Payment Session	Es wird eine Fehlerseite im Händlerportal angezeigt. Als

		2. Klicke auf «Bezahlvorgang abbrechen»	Ursache ist «Zahlungssession wurde durch den Benutzer abgebrochen.» angegeben.
9	Einnahmen auszahlen	1. Navigiere zu «Einnahmen» 2. Klicke auf «Auszahlen» 3. Warte 10 Minuten	Es wurde eine neue Auszahlung an achtpunkt GmbH erstellt und nach 10 Minuten wechselt der Status zu «abgeschlossen».
10	Anmeldung mit bestehendem Account (Direct Payout)	1. Melde dich ab (sofern noch angemeldet) 2. Logge dich erneut ein mit folgenden Daten: a. Benutzername: jan.meier2@ost.ch b. Passwort: nuprob-2xitmi-carSeb 3. Verifiziere, dass unter den Händlerinfos alle Felder abgefüllt sind und die Auszahlungsart auf «Direkt» gesetzt ist.	Das Login ist erfolgreich und unter dem Menüpunkt «Zahlungen» wird nur das Untermenü «Sessions» angezeigt.
11	Kasse erstellen	1. Navigiere zu «Kassen» 2. Erstelle eine neue Kasse mit einem beliebigen Namen	Die Kasse wird in der Kassenübersicht angezeigt
12	Instant Pay Local Zahlung ausführen	1. Navigiere zu «Instant Pay Local» 2. Wähle die zuvor erstellte Kasse 3. Scanne den QR-Code mit der E-Banking App 4. Gib den Betrag von 5 CHF ein 5. Warte 5 Minuten 6. Navigiere zu «Sessions»	Es wird eine neue Session angezeigt mit dem Typ «Instant Pay Local» und dem Betrag 5 CHF.
13	Erfolgreiche Instant Pay Web Zahlung	1. Navigiere zu https://demo-shop.instant-pay.ch/ 2. Lege mindestens ein Produkt in den Warenkorb und navigiere zum Warenkorb 3. Wähle «Weiter zum Bezahlen» 4. Gib als E-Mail, Vorname, Nachname, Adresse fiktive Angaben an. 5. Stell sicher, dass Instant Pay als Zahlungsmethode gewählt ist. 6. Klicke auf «Kostenpflichtig bestellen» 7. Scanne mit dem E-Banking die QR-Rechnung	Der Benutzer wird 30 Sekunden nach der Zahlung auf die Bestätigungsseite des Webshops weitergeleitet.
14	Fehlgeschlagene Instant Pay Web Zahlung	8. Navigiere zu https://demo-shop.instant-pay.ch/ 9. Lege mindestens ein Produkt in den Warenkorb und navigiere zum Warenkorb 10. Wähle «Weiter zum Bezahlen» 11. Gib als E-Mail, Vorname, Nachname, Adresse fiktive Angaben an. 12. Stell sicher, dass Instant Pay als Zahlungsmethode gewählt ist.	Der Benutzer wird nach Ablauf des Payment Session Zeitlimits auf die Fehlerseite der Seite mit den Bestellinformationen zurück umgeleitet.

		13. Klicke auf «Kostenpflichtig bestellen» 14. Warte 12 Minuten	
--	--	--	--

Tabelle 8 Manuelle Testfälle

4.10.2 Testresultate manuelle Tests

Nachfolgend die Testresultate aller manuellen Tests nach dem ersten Durchlauf.

Nr.	Resultat	Erwartetes Resultat	Kommentar
1	OK	Das Benutzerkonto wurde erstellt und der Benutzer wurde auf die Einstellungsseite des Händlerportals weitergeleitet, um die Details des Händlers abzufüllen.	-
2	OK	Die Angaben wurden korrekt gespeichert und werden bei einem Refresh der Seite wieder abgefüllt.	-
3	OK	Die Zahlungssession wurde erstellt und auf dem Bildschirm wird die Checkout Seite mit 10 CHF angezeigt.	-
4	OK	Der Benutzer wird innerhalb von 30 Sekunden auf die Bestätigungsseite im Händlerportal weitergeleitet.	-
5	OK	Die Detailseite der Transaktionen wird mit dem Betrag 10 CHF angezeigt. Es ist nur eine Transaktion mit dem Status «Abgeschlossen» aufgelistet.	-
6	OK	Es wird ein PDF mit der Überschrift «Zahlungsbeleg» angezeigt. Die Transaktion sowie Zahlungsreferenz und erwarteter Betrag stimmen im PDF mit der Detailseite der Payment Session überein.	-
7	OK	Ein PDF wird angezeigt mit einem vollständigen Zahlteil einer QR-Rechnung. Der Betrag beläuft sich auf 10 CHF.	-
8	OK	Es wird eine Fehlerseite im Händlerportal angezeigt. Als Ursache ist «Zahlungssession wurde durch den Benutzer abgebrochen.» angegeben.	-
9	OK	Es wurde eine neue Auszahlung an achtpunkt GmbH erstellt und nach 10 Minuten wechselt der Status zu «abgeschlossen».	-
10	OK	Das Login ist erfolgreich und unter dem Menüpunkt «Zahlungen» wird nur das Untermenü «Sessions» angezeigt.	-
11	OK	Die Kasse wird in der Kassenübersicht angezeigt	-
12	OK	Es wird eine neue Session angezeigt mit dem Typ «Instant Pay Local» und dem Betrag 5 CHF.	-
13	OK	Der Benutzer wird 30 Sekunden nach der Zahlung auf die Bestätigungsseite des Webshops weitergeleitet.	-
14	OK	Der Benutzer wird nach Ablauf des Payment Session Zeitlimits auf die Fehlerseite der Seite mit den Bestellinformationen zurück umgeleitet.	-

Tabelle 9 Testresultate manuelle Testfälle

4.10.3 Automatisierte Tests

Damit unter anderem aktueller Code nach Erweiterungen einfacher getestet werden kann, wurden automatisierte Tests erstellt. Unit-Tests werden jeweils nach einem Commit und Push ins Repository durch GitHub Actions ausgeführt.

Da die Hauptaufgabe vom Händlerportal das Senden, Empfangen und Anzeigen von Daten ist, wurden nur die Util-Klassen getestet, welche eine Frontend-Spezifische Aufgabe ausführen. Für die Frontend Tests wurde Jest verwendet.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	97.33	94.35	100	97.33	
domain.utils.ts	100	100	100	100	
form.utils.ts	100	100	100	100	
format.utils.ts	100	91.66	100	100	39,56
react-node.utils.ts	100	100	100	100	
swiss-qr-bill.utils.ts	100	100	100	100	
toast.utils.ts	100	100	100	100	
token.utils.ts	76.74	50	100	76.74	...3-15,25-27,34-35
zod.utils.ts	98.95	97.91	100	98.95	95
zustand.states.ts	100	100	100	100	

Test Suites: 9 passed, 9 total
 Tests: 88 passed, 88 total
 Snapshots: 0 total
 Time: 2.207 s
 Ran all test suites.

Abbildung 44 Screenshot Test-Coverage Händlerportal

Da die Checkout Seite mehr Logik enthält als das Händlerportal (z.B. Generierung von Swiss QR Code und QR-Rechnung) werden hier neben den Util-Klassen einige Teile von Service-Klassen getestet. Die UI-Komponenten von spartan-ui kommen teilweise mit bereits implementierten Testmethoden.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	67.59	91.42	46.66	67.59	
app/checkout-page/services	32.88	83.33	26.66	32.88	
checkout-session.service.ts	30.76	75	25	30.76	22-34, 37-48, 52-67, 76-77, 80-95, 101-105, 108-142
qr-bill.service.ts	34.22	100	28.57	34.22	61-95, 106-132, 135-138, 141-184, 187-224
app/common/decorators	83.33	92.85	85.71	83.33	
catch.decorator.ts	100	83.33	100	100	15
loader.decorator.ts	75.92	100	80	75.92	42-54
app/common/services	55.47	83.33	45.45	55.47	
alert.service.ts	43.13	66.66	50	43.13	15-24, 30-48
api.service.ts	50.81	100	25	50.81	17-22, 25-46, 49-50
common.service.ts	100	100	100	100	
navigation.service.ts	66.66	100	50	66.66	12-17
app/common/ui/ui-button-helm/src	100	100	100	100	
index.ts	100	100	100	100	
app/common/ui/ui-button-helm/src/lib	92.68	100	0	92.68	
hlm-button.directive.ts	100	100	100	100	
hlm-button.token.ts	72.72	100	0	72.72	16-18, 20-22
app/common/ui/ui-dialog-helm/src	100	100	100	100	
index.ts	100	100	100	100	
app/common/ui/ui-dialog-helm/src/lib	94.04	100	0	94.04	
hlm-dialog-close.directive.ts	100	100	100	100	
hlm-dialog-content.component.ts	100	100	100	100	
hlm-dialog-description.directive.ts	100	100	100	100	
hlm-dialog-footer.component.ts	100	100	100	100	
hlm-dialog-header.component.ts	100	100	100	100	
hlm-dialog-overlay.directive.ts	100	100	100	100	
hlm-dialog-title.directive.ts	100	100	100	100	
hlm-dialog.component.ts	100	100	100	100	
hlm-dialog.service.ts	58.82	100	0	58.82	21-34
app/common/ui/ui-icon-helm/src	100	100	100	100	
index.ts	100	100	100	100	
app/common/ui/ui-icon-helm/src/lib	89.28	100	0	89.28	
hlm-icon.directive.ts	100	100	100	100	
hlm-icon.token.ts	70	100	0	70	14-16, 18-20
app/common/utills	100	100	100	100	
responsive.utills.ts	100	100	100	100	
token.utills.ts	100	100	100	100	
url.utills.ts	100	100	100	100	
app/global-alert	77.77	100	50	77.77	
global-alert.component.html	100	100	100	100	
global-alert.component.ts	70.37	100	0	70.37	20-27
environments	100	100	100	100	
environment.ts	100	100	100	100	

Test Suites: 11 passed, 11 total
 Tests: 21 passed, 21 total
 Snapshots: 0 total
 Time: 6.295 s
 Ran all test suites.

Abbildung 45 Screenshot Test-Coverage Checkout Seite

Das Backend-System verwendet MStest als Testframework. Util-Klassen und einige Business/Service-Klassen werden mithilfe einer In-Memory-Datenbank vom Entity Framework getestet. Hier gilt zu beachten, dass die Code Coverage den generierten Open API Client Code für die Open Banking Schnittstelle, Modelklassen sowie die Migrationsdateien von Entity Framework in die Statistik einbezieht, was ca. 6000 der 9133 nicht abgedeckten Blöcke ausmacht.

Hierarchie	Abgedeckt (Blöcke)	Nicht abgedeckt (Blöcke)	Abgedeckt (Zeilen)	Teilweise abgedeckt (Zeiler)	Nicht abgedeckt (Zeilen)
juan_JANMEIER75BB_2025-06-11.16_13_58.coverage	6961	9133	3996	32	11859
instantpaybackend.model.test.dll	9	0	6	0	0
instantpaybackend.business.dll	1128	1209	424	9	535
instantpaybackend.adapter.dll	17	1456	12	0	1061
instantpaybackend.adapter.test.dll	40	2	28	2	0
instantpaybackend.database.dll	95	3	42	1	2
instantpaybackend.utills.test.dll	743	49	568	4	40
instantpaybackend.dll	12	5800	5	0	9741
instantpaybackend.model.dll	167	259	173	0	261
instantpaybackend.utills.dll	877	331	460	9	207
instantpaybackend.test.dll	38	2	36	2	0
instantpaybackend.business.test.dll	3835	22	2242	5	12

Abbildung 46 Screenshot Test-Coverage Backend

4.10.4 Usability Test

Mit dem Usability Test soll aufgezeigt werden, ob Instant Pay von einem Händler so benutzt wird, sodass er Einnahmen erzielen kann und von einem Käufer so benutzt wird, dass er Produkte kaufen kann.

Der Aufbau des Usability Tests ist so gestaltet, dass den Benutzern ein Skript mit ein paar Input Infos (z.B: Login) und einem Ziel vorgelegt wird. Es gibt verschiedene Aufgaben, welche so in Instant Pay umgesetzt werden müssen. Die Testpersonen benötigen einen Laptop und erhalten ein Smartphone mit der E-Banking App für die Ausführung von Zahlungen.

Die Benutzer werden während des Tests von dem Entwickler beobachtet, der sich dadurch Notizen über mögliche Verbesserungen machen kann. Am Ende werden die User noch kurz persönlich über die Usability von Instant Pay befragt.

Nr.	Aufgabe	Bemerkungen Tester
1	- Im ersten Schritt soll ein Benutzerkonto im Händlerportal erstellt werden. Dafür sind die folgenden Informationen vorgegeben: - Händlername: achtpunkt GmbH - IBAN: CH1189144384113162191 - Auszahlungsart: Sammelkonto - Inhaber Bankkonto: achtpunkt GmbH - Adresse: Oberer Graben 8 - Postleitzahl: 9000 - Ort: St. Gallen - Zahlungen sollen über ein Sammelkonto gemacht werden.	
2	Erstelle zwei Kassen: «Kasse 1» und «Kasse 2»	
3	Versuche mit Instant Payment Now eine Zahlung zu tätigen. Du sollst 10 CHF einkassieren und die Zahlung mit der E-Banking App ausführen.	
4	Erstelle einen QR-Sticker (Instant Pay Local) für die Kasse 1 und Kasse 2 und lade diesen herunter.	
5	Erstelle eine Rückerstattung für die zuvor erstellte Zahlung.	
6	Erstelle einen neuen API Token und kopiere ihn in das Feld «Bemerkungen Tester»	

Tabelle 10 Usability Test

Aus den Usability Tests geht hervor, dass das Händlerportal grundsätzlich verständlich aufgebaut ist. Missverständnisse gab es lediglich beim Statuskonzept von Payment Sessions und der Button für die Rückzahlung wurde nicht auf Anhieb gefunden. Für das Statuskonzept wurden zusätzliche Beschreibungen auf der Detailseite einer Zahlungssession eingefügt, damit verständlicher ist, welcher Status zur Payment Session und welcher Status zu einer Transaktion gehört. Der Button für die Rückzahlungen wurde nach den Usability Tests oben rechts neben den Button «Zahlungsbeleg» gesetzt.

4.10.5 Auswertung FR und NFR

Alle funktionalen und nicht funktionalen Anforderungen wurden erfolgreich umgesetzt, was den Proof of Concept für ein solches System anhand der zu Beginn gestellten Anforderungen erfolgreich macht.

5. Ergebnisdiskussion mit Ausblick

In diesem Abschnitt werden die Ergebnisse der Arbeit sowie mögliche Verbesserungsvorschläge aufgezeigt. Neben der Entwicklung des Proof of Concept für eine Zahlungsplattform mit Instant Payment wurden Gespräche mit PPI Schweiz, der Innofactory, der Zürcher Kantonalbank, der Luzerner Kantonalbank, Twint, SIX, Digitec Galaxus und Brack geführt, um herauszufinden, wie diese Firmen Instant Payment aktuell nutzen und ob sie eine Chance für Instant Payment als Zahlungssystem sehen. Ein herzliches Dankeschön geht an folgende Personen:

- **Julien Rösch**
Consultant
PPI Schweiz GmbH
- **Mark Chardonens**
CEO, Innovationsdesigner
Innofactory AG
- **Fabian Widmer**
DevOps Engineer Systemverbund Instant Payment
Zürcher Kantonalbank
- **Timo Pfahl**
Head SIC IT
SIX Interbank Clearing AG
- **Markus Ruggiero**
Leiter Produktmanagement Basisleistungen und Zahlen
Luzerner Kantonalbank AG
- **Stephan Haller**
Software Engineering Manager
Digitec Galaxus AG
- **Brack.Alltron AG**
Medienstelle
- **René Hägeli**
Head of Scheme Management
Twint AG

5.1 Befragung zu Schweizer Zahlungsmethoden

Um ein aktuelles Bild über die Akzeptanz verschiedener Zahlungsmethoden der Kunden in der Schweiz zu erhalten, wurden 350 Personen zwischen 18 und 67 Jahren befragt. Die Verteilung der befragten Personen sieht wie folgt aus:

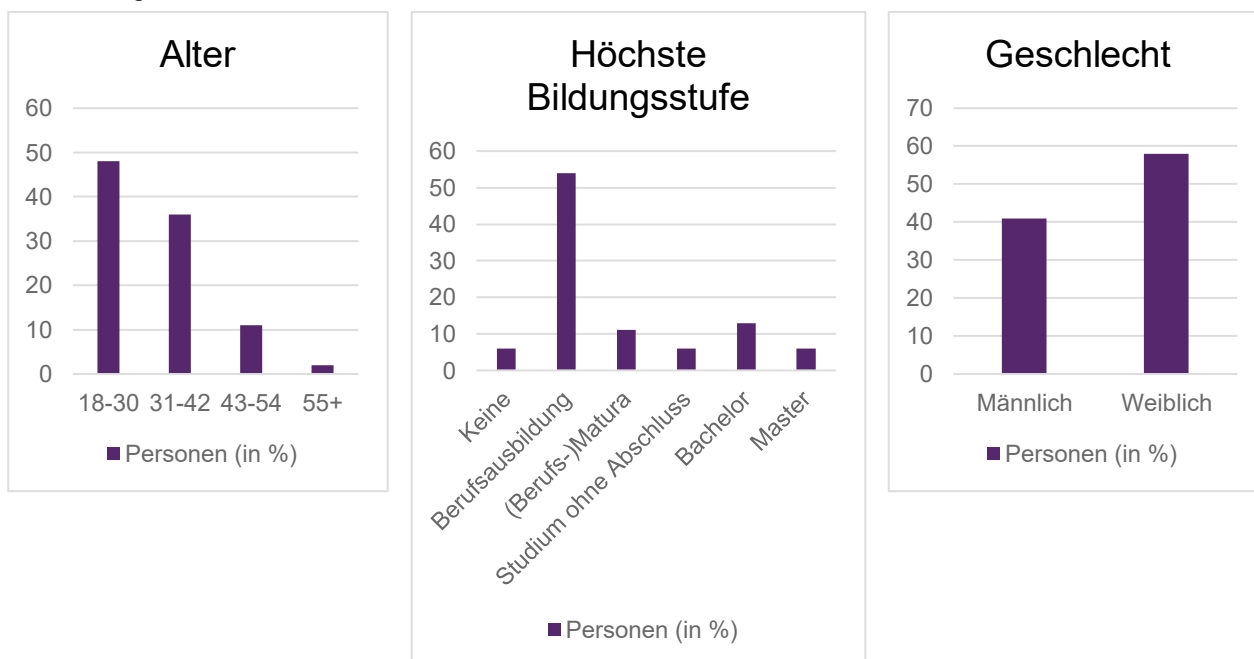


Abbildung 47 Verteilung der Personen für Befragung Schweizer Zahlungsmethoden

Die Karte ist die beliebteste Zahlungsmethode in Geschäften, knapp gefolgt von Twint, Barzahlung und Mobile Payments (Apple Pay / Google Pay). Unter 30-Jährige bevorzugen die Zahlung mit Twint vor einer Kartenzahlung, während die über 30-Jährigen eher auf die Kartenzahlung setzen. Es zeigt sich auch, dass unter 30-Jährige eher Mobile Payments als Bargeld benutzen. Bei den über 30-Jährigen ist dies genau umgekehrt.

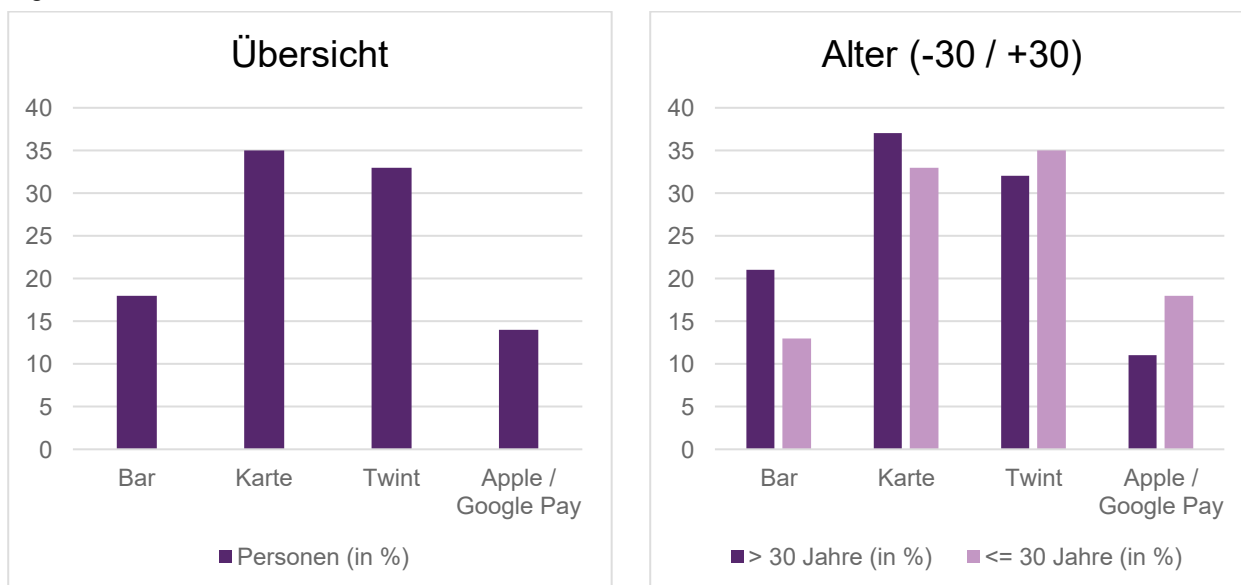


Abbildung 48 Charts beliebteste Zahlungsmethoden in Geschäften

In Onlineshops ist Twint die beliebteste Zahlungsmethode. Bei den unter 30-Jährigen ist sie noch etwas beliebter als bei den über 30-Jährigen. Auf dem zweiten Platz ist die Kartenzahlung, gefolgt von Kauf auf Rechnung, Klarna⁸⁶, Mobile Payments und der Banküberweisung (Vorauskasse).

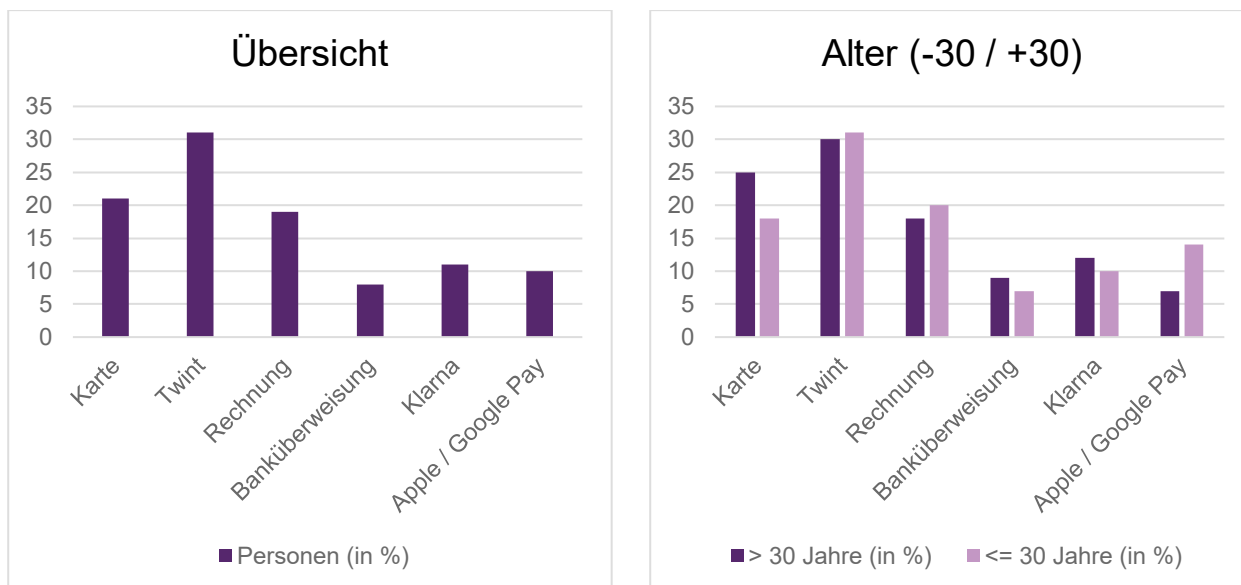


Abbildung 49 Charts beliebteste Zahlungsmethoden in Onlineshops

Neben der Befragung zu aktuellen Zahlungsmethoden, wurden die Personen zudem über Instant Payment aufgeklärt und über Instant Payment als neues Zahlungsmittel in Onlineshops und Geschäften befragt. Aus der Befragung geht hervor, dass viele Personen noch nicht wissen, was Instant Payment ist. Gerade mal 29% der Befragten haben bereits von Instant Payment gehört. Dafür zeigt sich ab, dass 68% der befragten Personen offen für neue Zahlungsmethoden wären. Hier sind interessanterweise die über 30-Jährigen mit 73% etwas offener als die unter 30-Jährigen mit 53%.

Am meisten Zuspruch erhält Instant Payment als Zahlungsmethode in Onlineshops, Hofläden, Restaurants und Lebensmittelgeschäften. Personen mit höherem Einkommen würden eher in einem Hofladen mit Instant Payment zahlen, hingegen würden Personen mit niedrigerem Einkommen Instant Payments eher für Onlinezahlungen nutzen. Eine wichtige Rolle für die Wahl der Zahlungsmethode spielt zudem auch der Preis, welchen die Händler Zahlen. 83% würden in einem Hofladen Instant Payment vor Twint als Zahlungsmittel bevorzugen, wenn die Gebühren von Instant Payment geringer wären als jene von Twint.

Damit Kunden Instant Payment anstelle einer bisher bekannten Zahlungsmethode wählen würden, sind Kriterien wie Sicherheit, Kostenfreiheit, Benutzerfreundlichkeit und Geschwindigkeit entscheidend. Bei rund 10% der befragten Personen müssten zusätzliche Anreize geschaffen werden, wie beispielsweise Cashback Angebote, Rabatte oder andere Bonussysteme.

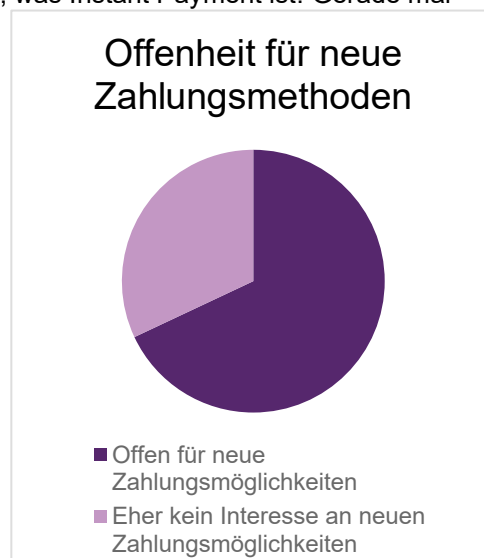


Abbildung 50 Chart Offenheit für neue Zahlungsmethoden

⁸⁶ <https://www.klarna.com/ch/>

5.2 Wo sieht die SIX Instant Payments als Zahlungsmittel ⁸⁷

Wie bereits im Kapitel [Varianten](#) beschrieben, hat die SIX ein Konsultationsverfahren für die Instant Payment Bridge (IPB) im August 2024 durchgeführt. Ziel dieser Initiative ist es, Instant Payments für Payment Schemes wie Visa, Mastercard oder Twint zugänglicher zu machen. Anstelle von proprietären Infrastrukturen und langwierigen Settlement-Prozessen sollen zukünftig IP-Transaktionen über das SIC-System in Echtzeit abgewickelt werden. Dies reduziert nicht nur die Latenzzeiten, sondern minimiert auch das Kontrahentenrisiko, da die Überweisung in wenigen Sekunden final und unwiderruflich in Zentralbankgeld abgeschlossen wird.

Wie aus dem Interview mit Timo Pfahl, Head SIC IT bei SIX Interbank Clearing AG, hervorgeht, liegt der Fokus der IPB nicht auf der Konkurrenz zu bestehenden Lösungen wie Twint, sondern auf der Schaffung einer effizienteren Abwicklungsinfrastruktur. Die IPB soll es Schemes ermöglichen, Zahlungen über einheitliche, standardisierte Schnittstellen direkt über das SIC IP-System abzuwickeln, ohne für jede Bank eine individuelle Integration vornehmen zu müssen.

Der Konsultations-Report⁸⁸ zeigt, dass die Marktrückmeldungen die IPB grundsätzlich begrüßen, jedoch noch viele technische, rechtliche und regulatorische Fragen offen sind. Besonders hervorgehoben wurde der Bedarf nach einer zentralen Bestätigungs-API für Transaktionsstatus, klar definierten Zugangsregeln und einem transparenten Preismodell. Die Integration in bestehende Bankprozesse soll dabei möglichst ohne Beeinträchtigung der Sicherheit erfolgen. Langfristig verfolgt die SIX mit der IPB das Ziel, die Fragmentierung im Schweizer Zahlungsverkehr zu reduzieren und eine gemeinsame, moderne Infrastruktur zu etablieren, über die Zahlungen unabhängig vom gewählten Scheme abgewickelt werden können.

5.3 Kann Twint nicht einfach auf Instant Payment umsteigen?⁸⁹

Ein unmittelbarer Umstieg von Twint auf Instant Payment ist aus mehreren Gründen nicht praktikabel. Im Rahmen der Swiss Payment Vision⁹⁰ evaluierte Twint eingehend die Möglichkeit, Instant Payment für den Versand von Zahlungen zu nutzen. Dabei zeigte sich, dass der technische und organisatorische Aufwand erheblich wäre: Jede Bank müsste individuell auf das neue System migriert werden, was eine Übergangsphase mit parallelen Infrastrukturen erforderlich machen würde. Twint müsste in dieser Zeit sowohl Zahlungen über das bestehende Poolaccount-System als auch über Instant Payment abwickeln, ein komplexes Hybridmodell. Aktuell sieht Twint den sinnvolleren Einsatz von Instant Payment im Bereich des interbanklichen Settlements: Anstelle der heutigen einmal täglichen Ausgleichszahlungen zwischen den Poolkonten der Banken soll künftig eine mehrfache Abwicklung pro Tag, auch an Wochenenden, ermöglicht werden. Ziel ist es, hohe Ausgleichszahlungen (wie sie etwa nach verlängerten Feiertagen entstehen) zu vermeiden und das Liquiditätsmanagement effizienter zu gestalten.

5.4 Gegenwart und Zukunft der Zahlungsmethoden im Onlinehandel ⁹¹

Die Zahlungslandschaft im Schweizer Onlinehandel befindet sich im Wandel. Laut Aussagen von Brack.Alltron AG⁹², dem Betreiber des Onlineshops Brack⁹³, bleibt der Kauf auf Rechnung weiterhin die beliebteste Zahlungsmethode bei ihrer Kundschaft, allerdings verzeichnet der Händler seit der Covid-Pandemie einen deutlichen Anstieg bei Sofortzahlungsmethoden, insbesondere Twint. Unterschiede im

⁸⁷ Aus persönlichem Gespräch mit Timo Pfahl (SIX Interbank Clearing)

⁸⁸ <https://www.six-group.com/dam/download/banking-services/billing-and-payments/instant-payments-bridge/consultation-report-ipb-en.pdf>

⁸⁹ Aus persönlichem Gespräch mit René Hägeli (Twint)

⁹⁰ https://www.snb.ch/public/publication/de/www-snb-ch/publications/communication/speeches/2023/ref_20230330_amrtmo/0/ref_20230330_amrtmo.de.pdf

⁹¹ Aus schriftlichem Interview mit Brack.Alltron AG

⁹² <https://brackalltron.ch/>

⁹³ <https://www.brack.ch/unternehmen>

Zahlungsverhalten sind zwischen B2C- und B2B-Kundschaft feststellbar, wobei B2C-Kunden häufiger Sofortzahlungen bevorzugen als B2B-Kunden, welche weiterhin überwiegend auf Rechnung bezahlen.

Neue Zahlungsmethoden wie Instant Payment können insbesondere dann interessant sein, wenn damit sowohl für den Onlinehändler als auch für die Kundschaft Kosten gesenkt werden können. Besonders im Bereich der Vorauszahlungen bietet es Potenzial, da Aufträge dadurch schneller freigegeben werden könnten. Der Entscheidungsprozess zur Einführung neuer Zahlungsmethoden basiert bei Brack primär auf den Kriterien des internen Aufwands hinsichtlich technischer Implementierung, operativer Abwicklung und Benutzerfreundlichkeit sowie des Kosten-Nutzen-Verhältnisses. Erst wenn diese Punkte in einem sinnvollen Rahmen liegen, wird die Umsetzung im Detail geprüft, je nach Fall auch mit Tests und gestaffeltem Rollout. Die Offenheit gegenüber neuen Zahlungsmethoden im Onlinehandel ist somit stark von der Praktikabilität und Wirtschaftlichkeit abhängig, wobei der Fokus klar auf etablierten sowie unmittelbar vorteilhaften Lösungen liegt.

Die Digitec Galaxus AG⁹⁴ bietet Teams beispielsweise die Möglichkeit, aktuelle Themen im Rahmen eines Hackathons aufzugreifen und umzusetzen. Dies erlaubt es dem Unternehmen, neue Technologien und Konzepte wie Instant Payments auf ihre Machbarkeit zu prüfen und dadurch vielfältige Ideen zu erkunden und frühzeitig potenziellen Mehrwert zu identifizieren. Auch bei der Auswahl neuer Zahlungsmethoden in den Onlineshops von Digitec Galaxus erfolgt eine fundierte Analyse hinsichtlich Kosten und Nutzen, bevor eine Einführung überhaupt in Betracht gezogen wird. Besonders interessant: Die bevorzugten Zahlungsmethoden der Kundschaft unterscheiden sich je nach Land deutlich⁹⁵.

Ein Blick in die Zukunft zeigt, dass Brack.Alltron AG durchaus Potenzial in biometrischen Verfahren für den E-Commerce der Zukunft sieht. Ein Beispiel dafür ist die Identifikation durch Scannen von Venenmustern und des Blutflusses in der Hand mithilfe von Infrarottechnologie.

5.5 Optimale Voraussetzungen der Open Banking Schnittstelle

In dieser Arbeit wird für die Abfrage der aktuellen Transaktionsdaten ein Polling verfahren verwendet. Als erstes wird eine Liste mit Transaktionen geladen (ID, Sender/Empfänger IBAN und Betrag) und für jede Transaktion einzeln die Transaktionsdetails (Senderinformationen, Empfängerinformationen, Adressdaten). Für die Abfrage von 10 Transaktionen werden somit 11 API-Calls an die Open Banking Schnittstelle gemacht und das alle 15 Sekunden. Ein weiterer Nachteil der aktuellen Implementierung ist, dass der Zahler teilweise etwas länger warten muss, bis er von der Checkout Seite bis zur Bestätigungsseite weitergeleitet wird. Dies ist vor allem dem Intervall von 15 Sekunden geschuldet, da es je nach Zeitpunkt der Zahlung bis zu 25 Sekunden dauern kann (10 Sekunden für Überweisung und 15 Sekunden bis zum nächsten Polling) bis der Zahler auf dem Bildschirm eine Bestätigung seiner Zahlung sieht.

Um diese zwei Punkte zu adressieren, würde eine optimale Open Banking Schnittstelle das Instant Pay Backend über neue Zahlungen benachrichtigen. Anstelle eines Pollings wird für jede eingegangene Zahlung auf dem Bankkonto eine Notification mit allen Informationen zu einer Transaktion (siehe [Mindestanforderung an Open Banking Schnittstelle](#)) an Instant Pay übermittelt. Einige Banken benutzen ein solches Vorgehen bereits intern, um eingehende Transaktionen direkt an das E-Banking System weiterzuleiten, damit die Transaktion dem Benutzer direkt nach Erhalt im Online- oder Mobile-Banking angezeigt wird⁹⁶. Diese

⁹⁴ <https://www.digitec.ch/de/wiki/528>

⁹⁵ Aus persönlichem Gespräch mit Stephan Haller (Digitec Galaxus)

⁹⁶ Aus persönlichem Gespräch mit Julien Rösch (PPI Schweiz)

Variante wurde bereits zu Beginn der Arbeit vorgeschlagen, jedoch aufgrund fehlender Machbarkeit auf Seite der Open Banking Schnittstelle verworfen.

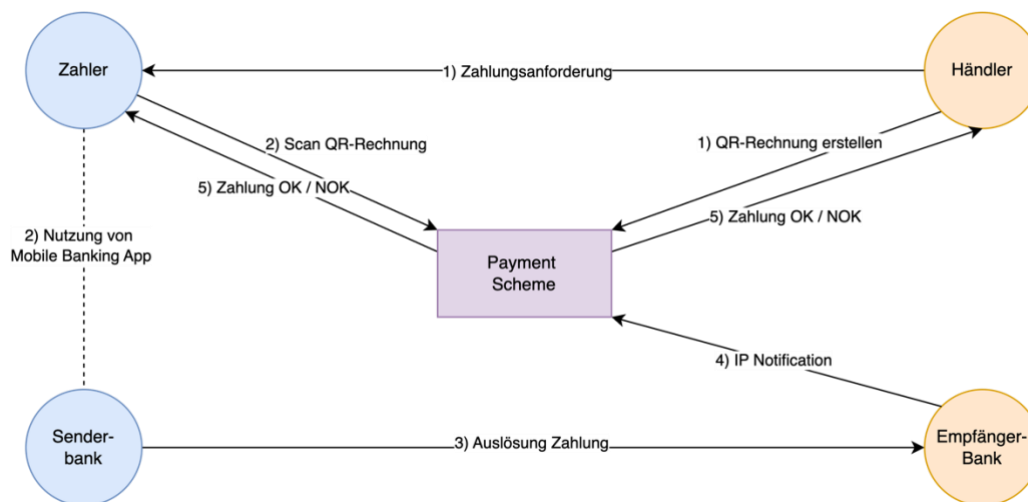


Abbildung 51 Variante Empfängerbank benachrichtigt Payment Scheme

In der realen Welt gäbe es zudem weitere Herausforderungen, welche angegangen werden müssten. Es kann je nach Bank unterschiedlich sein, wie schnell eine eingehende Instant Payment Zahlung über eine Open Banking Schnittstelle zur Verfügung steht. Der Kunde kann sofort über das Geld verfügen, jedoch könnte es ein paar Minuten bis Stunden dauern, bis die eingegangene Transaktion über Open Banking Schnittstellen abgefragt werden kann⁹⁷. Sollte es beim Abfragen des Zahlungseinganges mit Instant Pay zu solchen Verzögerungen kommen, könnte ein Zahlungssystem mit Instant Payment und Open Banking nicht realisiert werden.

5.6 Problemstellungen

Neben den bereits aufgezeigten Voraussetzungen für eine optimale Open Banking Schnittstelle gibt es noch weitere Hürden, welche in folgendem Abschnitt beschrieben werden.

5.6.1 Keine Pflicht für Versenden von Zahlungen

Alle Banken sind bis 2026 dazu verpflichtet, Instant Payment Zahlungen empfangen zu können. Rund 60 Banken können seit August 2024 Instant Payments empfangen. Dies deckt rund 95% der Kundenzahlungen in der Schweiz ab⁹⁸. Die Schweizerische Nationalbank und SIX schreiben jedoch keine Verpflichtung vor, Instant Payments versenden zu können. Daher können Banken selbst entscheiden, ob sie ihren Kunden Instant Payment Zahlungen anbieten möchten. Banken beobachten den Markt derzeit und entscheiden anhand der Nachfrage nach Instant Payments, ob und wie schnell sie ihren Kunden diese Zahlungsart anbieten⁹⁹.

Stand Mai 2025 bieten nicht einmal ein Dutzend Banken den Versand von Instant Zahlungen an. Nachfolgend eine Auflistung aller Banken, welche Instant Zahlung versenden können inklusive der

⁹⁷ Aus Interviews und persönlichen Gesprächen mit Fabian Widmer (Zürcher Kantonalbank), Julien Rösch (PPI Schweiz) und Markus Ruggiero (Luzerner Kantonalbank)

⁹⁸ <https://www.six-group.com/de/products-services/banking-services/billing-and-payments/instant-payments.html#:~:text=Seit%20dem%2020,Zahlungen%20empfangen%20und%20sekunde nschnell%20verarbeiten>

⁹⁹ Aus Interview mit Makrus Ruggiero (Luzerner Kantonalbank)

Gebühren für Privatkunden¹⁰⁰. Der Empfang von Instant Zahlungen ist bei allen Banken für Privatkunden kostenlos.

Bank	Gebühren (Privatkunden, Versand)
Hypothekarbank Lenzburg	CHF 0.00 (kostenlos) ¹⁰¹
Berner Kantonalbank	CHF 0.00 (kostenlos) ¹⁰²
Spar- und Leihkasse Gürbetal	CHF 0.10 pro Überweisung (Gratis mit Privatkonto Premium) ¹⁰³
Bank CIC	CHF 1.50 pro Überweisung ¹⁰⁴
Raiffeisen	CHF 2.00 pro Überweisung (12 pro Jahr gratis für Mitglieder) ¹⁰⁵
Zürcher Kantonalbank	CHF 2.00 pro Überweisung ¹⁰⁶
St. Galler Kantonalbank	CHF 2.00 pro Überweisung ¹⁰⁷
Aargauische Kantonalbank	CHF 2.00 pro Überweisung ¹⁰⁸
Banca dello Stato del Cantone Ticino	CHF 5.00 pro Überweisung ¹⁰⁹
UBS	CHF 5.00 pro Überweisung ¹¹⁰

Tabelle 11 Transaktionsgebühren Versand Instant Payment

Die nachfolgenden Banken haben die Einführung für den Versand von Instant Payment mit einem konkreteren Zeitpunkt geplant¹¹¹. Für diese sind jedoch stand Mai 2025 keine Informationen über Gebühren bekannt. Es gibt weitere Banken, welche die Einführung für den Versand von Instant Payment Zahlungen angekündigt haben, jedoch bisher kein konkretes Datum genannt haben.

Bank	Geplante Einführung von IP-Versand
LLB	2025
Swissquote	2025
Valiant Bank AG	2025 ¹¹²
Schwyzer Kantonalbank	2025 ¹¹³
Bank BSU	2026

¹⁰⁰ <https://www.moneyland.ch/de/instant-payment-echtzeit-ueberweisung-schweiz>

¹⁰¹ <https://www.hbl.ch/de/private/zahlen/zahlungsverkehr/instant-zahlungen/>

¹⁰² <https://www.bekb.ch/die-bekb/blog/aktuelles/instantpayment>

¹⁰³ <https://www.slguerbetal.ch/unternehmen/zahlen/zahlungsverkehr/digital>

¹⁰⁴ <https://www.cic.ch/dam/jcr:1acd4520-3340-46ec-83bf-a433ea65aafa/fees-and-conditions-for-private-individuals.pdf>

¹⁰⁵ <https://www.raiffeisen.ch/rch/de/privatkunden/konten-und-bezahlen/instant-zahlung.html>

¹⁰⁶ <https://www.zkb.ch/de/private/konten-karten/zahlungsverkehr/instant-zahlung.html>

¹⁰⁷ <https://www.sgkb.ch/de/e-banking/leistungsumfang/instant-zahlung>

¹⁰⁸ <https://www.akb.ch/private/bezahlen/zahlungsverkehr/instant-zahlung>

¹⁰⁹ <https://www.bancastato.ch/prodotti-e-prezzi/Pagamenti-istantanei.html>

¹¹⁰ <https://www.ubs.com/ch/de/services/payments/pay-invoices/instant-payment.html>

¹¹¹ <https://www.moneyland.ch/de/instant-payment-echtzeit-ueberweisung-schweiz>

¹¹² <https://www.luzernerzeitung.ch/wirtschaft/zentralschweiz/instant-payment-jetzt-kommt-die-ruckzuck-zahlung-doch-zentralschweizer-banken-zoegern-ld.2657273>

¹¹³ <https://www.luzernerzeitung.ch/wirtschaft/zentralschweiz/instant-payment-jetzt-kommt-die-ruckzuck-zahlung-doch-zentralschweizer-banken-zoegern-ld.2657273>

5.6.2 Gebühren für Instant Zahlungen

Wie die Umfrage zu Schweizer Zahlungsmethoden zeigt, ist für die Akzeptanz von Instant Payment Zahlungen neben Sicherheit und Einfachheit auch Gebührenfreiheit entscheidend.

Der aktuelle Markt spielt dabei eine Rolle. Banken beobachten den Markt und erstellen dadurch ihre Strategien. Solange kein etablierter Mitbewerber kostenfreie Instant Zahlungen anbietet, hat eine neu einsteigende Bank keinen Anreiz, dies zu tun. Banken müssen immense Investitionen in die Infrastruktur stecken, um den Versand von Instant Payments überhaupt zu realisieren.

Ein weiterer Faktor, wie Markus Ruggiero von der Luzerner Kantonalbank betont, ist das Sicherheitsgefühl und das Vertrauen, das eine Bank schafft. Denn jeder, der Instant Payments versenden kann, musste eine vollständige KYC-Prüfung (Know Your Customer¹¹⁵) durchlaufen und wird laufend geprüft. Prozesse wie Geldwäschereikontrolle, Betrugsschutz, etc. laufen kontinuierlich im Hintergrund ab und bleiben für Nutzer meist unsichtbar. Dem Kunden wird mittels eines solchen Accounts die Nutzung oder Anwendung solcher Dienstleistungen ermöglicht. Je nach Kundenwunsch bzw. Segment wird ihm die gesamte Infrastruktur bis hin zur Filiale und/oder ein persönlicher Berater bereitgestellt, argumentiert Ruggiero. Somit sind Gebühren für Instant Payments durchaus gerechtfertigt, da sie den Aufwand für ein sicheres, gesetzeskonformes und vertrauenswürdige Zahlungssystem widerspiegeln, insbesondere in einem Umfeld, in dem nicht alle Akteure mit guten Absichten handeln. Daher kann auch nicht nur die einzelne Transaktion beurteilt werden. Es gibt im Gesamten vor und auch nach einer Zahlung viele Aspekte, die einbezogen werden müssen.¹¹⁶

Auf der anderen Seite kann argumentiert werden, dass Banken für die Integration von Kartenzahlungen sowie Apple und Google Pay ebenfalls viel Aufwand hatten. Für Twint beispielsweise, müssen Banken eine Lizenzgebühr zahlen¹¹⁷. Diese Gebühren werden den Kunden nicht in Rechnung gestellt. Es stellt sich nun die Frage: Wieso sollten Kunden dann für ein Zahlungsmittel wie Instant Payment bezahlen, welches von den Banken selbst betrieben wird?

Um das Problem der Gebührenpflicht zu lösen, könnten individuelle Vereinbarungen mit Banken ausgehandelt werden. Im Gegenzug für die Befreiung von Gebühren auf Zahlungen im Zusammenhang mit einer Plattform wie Instant Pay, würden die Banken einen Teil der Gebühren erhalten, die der Händler zahlen muss. Dies wäre in der Schweiz eine Chance für Banken, um an einer Zahlungsmethode mitverdienen zu können. Eine Gebühr würde mit dieser Variante jedoch nur vermeintlich abgeschafft werden. Es würde, wie bei den meisten heutigen Zahlungsmitteln, die Händler treffen, welche die Gebühren wiederum durch höhere Preise auf die Kunden abwälzen.

Die EU hatte ähnliche Probleme mit den Gebühren von Instant Zahlungen. Um die Akzeptanz von Echtzeitüberweisungen in der EU zu fördern, wurde durch die EU-Verordnung 024/886¹¹⁸ festgehalten, dass eine für Instant Zahlungen erhobene Gebühr nicht höher sein darf als die Gebühr für herkömmliche SEPA-Überweisungen. So eine Regelung könnte auch in der Schweiz Instant Payments für Kunden attraktiver machen.

Für Banken kostet eine Instant Payment Überweisung im SIC gleich viel, wie eine normale Kundenzahlung im klassischen SIC RTGS. Der Durchschnittspreis der SIX Interbank Clearing AG liegt bei knapp über einem Rappen für Kundenzahlungen im SIC¹¹⁹.

¹¹⁴ Aus persönlichem Gespräch mit Markus Ruggiero (Luzerner Kantonalbank)

¹¹⁵ https://de.wikipedia.org/wiki/Know_your_customer

¹¹⁶ Aus persönlichem Gespräch mit Markus Ruggiero (Luzerner Kantonalbank)

¹¹⁷ <https://www.netzwoche.ch/news/2024-07-11/das-sagt-der-twint-ceo-zur-einfuehrung-von-instant-payment>

¹¹⁸ <https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32024R0886>

¹¹⁹ Aus persönlichem Gespräch mit Timo Pfahl (SIX Interbank Clearing)

5.6.3 Swiss QR Code

Aktuell kann die E-Banking App des Kunden anhand des Swiss QR Codes (oder auch QR-Rechnung) nicht ermitteln, ob für eine gescannte Zahlung eine Instant Payment Zahlung oder eine normale RTGS-Zahlung erstellt werden soll. Dies bedeutet, dass der Benutzer aktiv bei der Erfassung der Zahlung festlegen muss, dass es sich um eine Instant Payment Zahlung handelt. Die Lösung für dieses Problem wäre eine Anpassung des Standards für QR-Rechnungen¹²⁰. Im Hintergrund des Swiss QR Code werden die Zahlungsinformationen auf mehrere Zeilen aufgeteilt. Es ist fix definiert auf welcher Zeile welche Information stehen muss. Hier könnte der QR-Code so erweitert werden, sodass eine Zahlungsform angegeben werden kann. Beispielsweise enthält ein Swiss QR Code die Informationen, dass die Zahlung als Instant Payment ausgeführt werden muss. Falls die Bank des Zahlers das Senden von Instant Payment nicht unterstützt, kann die Zahlung nicht ausgeführt werden. Alternativ könnte der Swiss QR Code so erweitert werden, dass eine empfohlene Zahlungsform mitgegeben werden kann (Instant Payment oder RTGS). In dieser Variante könnte der Zahler auch mit einer Bank, welche kein Instant Payment unterstützt die Zahlung als normale Banküberweisung ausführen.

Aktuell gibt es bei der SIX Interbank Clearing AG keine geplanten Vorhaben, um eine Angabe des Zahlungstyps im Swiss QR Code zu integrieren¹²¹.

5.6.4 Verschiedene Zahlungsflows in Mobile Banking Apps

Mobile Banking Apps gibt es viele. Einige Banken kaufen ganze Lösungen bei externen Anbietern ein während andere Mobile Banking Apps und die zugrundeliegende Infrastruktur in House entwickeln. Dies führt zwangsläufig zu einer breiten Mischung von diversen Banking Apps und jede davon sieht anders aus, bietet eine andere Benutzerführung und andere Features. Für eine Plattform wie Instant Pay ist es entscheidend, dass die Zahlungsabwicklung von Instant Zahlungen so schnell und einfach wie möglich geht. Das bedeutet, der Benutzer muss sich schnell anmelden können, einen Swiss QR Code scannen können und Instant Payments ausführen können. Weitere Features wie das Importieren von QR-Rechnungen (siehe Kapitel [Checkout Seite](#)), oder ein Ladescreen und eine Bestätigungsseite in der Mobile Banking App, nachdem die Zahlung ausgeführt wurde, wären aus Sicht der Usability für den Zahler von grosser Bedeutung.

Dies sind Themen und Wünsche, die sich kaum beeinflussen lassen. Damit der Zahler trotzdem für jede Mobile-Banking-App eine Wegleitung hat, könnte auf der Checkout-Seite von Instant Pay der Benutzer auswählen, bei welcher Bank er ein Konto hat. Anhand dieser Information werden verschiedene Anleitungen angezeigt.

5.6.5 QR-Rechnungen auf dem Smartphone scannen

Wenn ein Benutzer auf einem Smartphone in einem Onlineshop eine Zahlung mit Instant Pay machen will und im Webshop dann auf die Checkout Seite von Instant Pay umgeleitet wird, müsste er den Swiss QR Code auf seinem Smartphonescreen mit seiner Smartphonekamera scannen. Einige Mobile Banking Apps bieten aktuell die Möglichkeit an, QR-Rechnung zu importieren und daraus direkt eine Zahlung zu erfassen. Instant Pay kann dieses Feature bereits nutzen (siehe [Checkout Seite](#)). Die einzige Problematik hier ist, wenn die Mobile Banking App den Import von QR-Rechnungen nicht unterstützt, kann der Zahler die Instant Pay Zahlung auf dem Smartphone nicht ausführen, ohne den QR-Code auf einem zweiten Gerät oder von einem ausgedruckten Papier zu scannen.

Ein Lösungsansatz für dieses Problem wäre ein Button auf der Checkout Seite, welcher direkt die Mobile Banking App über ein standardisiertes Verfahren öffnet und die Zahlungsdaten im Hintergrund vorabfüllt. Für die technische Implementierung könnten Deep Links verwendet werden (bei Android App Links¹²², bei IOS

¹²⁰ <https://www.six-group.com/de/products-services/banking-services/payment-standardization/standards/gr-bill.html>

¹²¹ Aus persönlichem Gespräch mit Timo Pfahl (SIX Interbank Clearing)

¹²² <https://developer.android.com/training/app-links?hl=de>

Universal Links¹²³). Voraussetzung hierfür wäre ein (idealerweise) einheitlicher Standard, um QR-Rechnungen über Deeplinks in Mobile Banking Apps zu öffnen, welche alle Banken in ihren Apps implementieren.

5.6.5.1 Limit von 20'000 für IP-Zahlungen^{124 125}

Aktuell gilt in der Schweiz ein Transaktionslimit von 20'000 CHF für Instant Payments. Das dürfte die meisten Alltagszahlungen abdecken. Im B2B Bereich könnte jedoch diese Limitierung Einschränkungen aufweisen. Dieses Limit wurde ursprünglich von regulatorischen Überlegungen geprägt, etwa zur Risikokontrolle und schrittweisen Einführung der neuen Zahlungsinfrastruktur. Die Schweiz hat sich hierbei am Vorgehen der EU bei der Einführung von Instant Payments für Euro orientiert. In der EU wurde das Limit im Laufe der Zeit sukzessive höher, mit dem Ziel, es perspektivisch ganz aufzuheben. Auch in der Schweiz ist eine Anhebung des Limits vorgesehen, insbesondere wenn die Marktteilnehmer mit dem System vertrauter werden und regulatorisch-technische Sicherheit gegeben ist. Eine Erhöhung würde weitere Use Cases ermöglichen, etwa im Corporate-Umfeld (z. B. hochpreisigen B2B-Transaktionen).

5.7 Variantenvergleich mit/ohne Zwischenkonto

Wie im Kapitel [Variantenentscheid](#) beschrieben ist, kann die Software sowohl mit Zwischenkontos sowie auch mit der direkten Auszahlung auf das Händlerkonto umgehen.

5.7.1 Variante mit Zwischenkonto

Die Variante mit Zwischenkonto bietet dem Händler den Vorteil, dass er unabhängig von seiner jeweiligen Hausbank Zahlungen empfangen kann. Durch ein zentrales Zwischenkonto, das vom Betreiber des Zahlungssystems verwaltet wird, werden sämtliche eingehenden Zahlungen zunächst gesammelt. Dies ermöglicht eine unkomplizierte Abwicklung, bei der Gebühren direkt vom Betreiber einbehalten und Auszahlungen an Händler gesammelt durchgeführt werden können. Bei dieser Variante sind die Zahlungen zwar nicht «instant» auf dem Händlerkonto jedoch können Sie ebenfalls mit einem Klick «instant» ausgezahlt werden. Eine mögliche Erweiterung für dieses Problem wäre bei jeder eingehenden Zahlung die direkte Weiterleitung an das Händlerkonto. Diese Variante birgt jedoch auch Herausforderungen: rechtliche und regulatorische Anforderungen entstehen vermutlich, da die Eigentumsverhältnisse des Geldes auf dem Zwischenkonto zu regeln sind. Hinzu kommt, dass Händler dem Betreiber des Systems Vertrauen entgegenbringen müssen, da dieser zeitweise im Besitz der eingegangenen Gelder ist. Ein weiterer Nachteil ist der Zahlungsempfänger beim Zahlvorgang durch den Kunden. Dem Kunden wird als Empfänger immer das Zwischenkonto mit dem Inhaber «Instant Pay» angezeigt. Dies könnte Kunden verwirren, wenn Sie bei «Onlineshop A» einkaufen und eine Zahlung an «Instant Pay» machen müssen.

5.7.2 Variante ohne Zwischenkonto

In der Variante ohne Zwischenkonto fließen Zahlungen direkt auf das individuelle Konto des Händlers. Dadurch entfallen rechtliche Komplexitäten, die bei der Verwaltung eines gemeinsamen Zwischenkontos auftreten können und Händler behalten jederzeit die direkte Kontrolle über die eingehenden Gelder. Diese Variante setzt jedoch voraus, dass Händler Konten bei Banken unterhalten, die kompatible Open-Banking-Schnittstellen bereitstellen, was eine viel grössere Hürde für einen Händler darstellt. Zudem könnte das Gebührenmanagement und die Zahlungsabwicklung für das Payment Scheme aufwendiger sein, da jeweils eine individuelle Integration mit unterschiedlichen Banken erforderlich ist und Gebühren von den Händlern nachträglich eingefordert werden müssen. Ein weiterer Nachteil ist, dass der Händler direkten Zugriff auf dieses Konto hat und Gelder einzahlen und abheben kann. Falls der Händler Geld auf dieses Konto einzahlt, wird es aktuell automatisch von Instant Pay wieder zurückgezahlt, da Instant Pay diese Zahlung nicht

¹²³ <https://developer.apple.com/documentation/xcode/supporting-universal-links-in-your-app>

¹²⁴ Aus persönlichem Gespräch mit Timo Pfahl (SIX Interbank Clearing)

¹²⁵ <https://de.wikipedia.org/wiki/Echtzeit%C3%BCberweisung#:~:text=Seit%20dem%2021.,weniger%20als%203%20Sekunden%20ausgef%C3%BChrt.>

erwartet hat. Wenn man diesen Mechanismus abschaltet, könnten unwillentliche oder doppelte Zahlungen der Kunden nicht automatisch von Instant Pay zurückgesendet werden. Das Bankkonto könnte mit dieser Variante somit nicht als normales Firmenkonto genutzt werden, sondern müsste dediziert nur für die Verwendung mit Instant Pay eröffnet und genutzt werden.

5.7.3 Zusätzliche Variante mit individuellen Händlerkonten ohne direkten Zugriff

Eine weitere Variante wäre, dass Instant Pay mithilfe einer Partnerbank individuelle Konten für jeden Händler eröffnet. Diese Konten wären dabei rechtlich eindeutig den Händlern zugeordnet, wodurch sämtliche Eigentumsverhältnisse klar geregelt sind. Allerdings hätten die Händler keinen direkten Zugriff auf diese Konten, wodurch sichergestellt wird, dass keine eigenständigen Transaktionen durch Händler stattfinden und somit keine Konflikte bei der Verarbeitung von Zahlungen durch Instant Pay auftreten. Stattdessen erfolgt die Verwaltung der Gelder ausschliesslich durch Instant Pay. Um über die Gelder verfügen zu können, müssen Händler jeweils manuell oder in einem automatisierten Intervall eine Auszahlung über das Instant-Pay-System auf ihr eigenes externes Bankkonto veranlassen. Dieses Modell verbindet somit die Vorteile klarer rechtlicher Verhältnisse mit einer zentralisierten Verwaltung, die ähnliche Abläufe wie die Variante mit Zwischenkonto bietet, jedoch den administrativen Aufwand für rechtliche Regelungen reduziert. Dennoch bleibt ein gewisses Vertrauen in den Betreiber von Instant Pay notwendig, da dieser die Verwaltung der Konten und der Auszahlungen übernimmt. Zudem könnten je nach Vereinbarung mit einer Partnerbank Gebühren für die Kontoführung anfallen.

5.7.4 Vergleich Zwischenkonto mit den Twint Poolaccounts

Wie im Kapitel [Unterschied zwischen Instant Payment und Twint](#) beschrieben, benötigt Twint für Zahlungen zwischen verschiedenen Banken Poolaccounts. Auf diesen Poolaccounts wird das Geld jeweils bis zur täglichen Ausgleichzahlung zwischen den Banken deponiert. Die wirtschaftlichen Berechtigten des Geldes auf diesen Poolaccounts sind die jeweiligen Personen (oder im P2M Fall die Firmen) die als Empfänger vom Sender ausgewählt wurden. Bei der Autorisierung einer Zahlung werden unwiderrufliche Zahlungsversprechen¹²⁶ ausgesprochen. Damit geht der Besitz, die wirtschaftliche Berechtigung, an den Empfänger über.¹²⁷ Dieser Ansatz könnte ebenfalls für das Zwischenkonto von Instant Pay verwendet werden.

5.8 Mögliche Weiterentwicklung – Open Banking auf Kundenseite

Instant Pay setzt zum jetzigen Zeitpunkt Open Banking nur auf der Empfängerseite ein. Die Plattform könnte so erweitert werden, dass die Open-Banking-Schnittstelle ebenfalls die Bankkonten der Kundschaft anspricht. Konkret bedeutet das, dass Instant Pay weiterhin eine Überprüfung auf der Empfängerseite durchführen kann und neu auch auf das Konto des Zahlenden zugreifen darf, etwa zur Erfassung von Zahlungen oder zur Abfrage des Kontostandes. Mit dieser Erweiterung ergeben sich zahlreiche neue Use-Cases:

- **Peer-to-Peer-Zahlungen (P2P):** Nutzer könnten sich direkt Geld untereinander senden, ähnlich wie bei Twint oder Revolut¹²⁸. Dies würde Instant Pay zu einer universellen Bezahlplattform machen, unabhängig davon, ob ein Händler involviert ist.
- **Reservierungen auf Konten:** Besonders im Bereich der Vorautorisierung von Zahlungen eröffnen sich neue Möglichkeiten. Beispielsweise an Tankstellen oder bei Autovermietungen. Bevor eine Zahlung ausgelöst wird, kann ein Betrag vorab reserviert werden, der später bei erfolgtem Kauf freigegeben oder angepasst wird. So lassen sich Szenarien wie „Karte autorisieren → tanken → final abrechnen“ umsetzen.

¹²⁶ <https://www.ubs.com/ch/de/services/guide/mortgages-and-financing/articles/irrevocable-promises-of-payment.html>

¹²⁷ Aus persönlichem Gespräch mit René Hägeli (Twint)

¹²⁸ <https://www.revolut.com/de-CH/send-and-receive/>

- **Wiederkehrende Zahlungen (Abos):** Über die Open-Banking Schnittstelle zum Kundenkonto könnten wiederkehrende Zahlungen direkt vom Bankkonto ausgelöst werden, ohne dass jedes Mal eine manuelle Freigabe notwendig ist. Diese Funktion könnte auch mit eBill oder einem Lastschriftverfahren verglichen werden.

Bei dieser Erweiterung würden sich weitere regulatorische und technische Fragen stellen. Zudem müssten in diesen Szenarien nicht nur eine Bank, sondern alle Banken mitziehen, damit eine breite Verfügbarkeit der Zahlungslösung sichergestellt werden kann. Hier würde sich vor allem die Frage stellen, wie sieht die Interaktion mit dem Benutzer aus. Werden diese Funktionalitäten in die E-Bankings der Schweizer Banken integriert oder muss sich jeder separat für einen externen Dienst registrieren, gibt es eine eigene App dafür oder nicht, etc.

5.9 Fazit

Ziel dieser Arbeit war es, eine Zahlungslösung zu entwickeln, die auf den bestehenden Banktechnologien Instant Payment und Open Banking basiert, um Zahlungsprozesse effizienter und standardisierter zu gestalten. Mit der Umsetzung einer Plattform bestehend aus einem Händlerportal und einer Checkout-Seite konnte die technische Machbarkeit eines solchen Systems demonstriert werden. Die entwickelte Lösung ermöglicht die Abwicklung von Instant-Payment-Transaktionen sowohl im Onlinehandel als auch am physischen Point of Sale sowie in Offline-Szenarien, wie einem Hofladen. Der Zahlungseingang wird dabei in Echtzeit validiert, wodurch eine sofortige Freigabe der Ware durch den Händler möglich ist.

Trotz der erfolgreichen technischen Umsetzung zeigt die Arbeit auch zentrale Herausforderungen auf, die einer breiten Einführung von Instant Payments in der Schweiz aktuell im Weg stehen. Besonders relevant ist die ungleiche Verfügbarkeit: Während der Empfang von Instant Payments bis Ende 2026 für alle Schweizer Banken verpflichtend ist, bleibt der Versand optional. Nur wenige Banken bieten aktuell den Versand an, oftmals verbunden mit zusätzlichen Gebühren. Dies wirkt sich negativ auf die Attraktivität für Endkunden aus und erschwert eine flächendeckende Marktdurchdringung.

Instant Payment birgt jedoch im Zusammenspiel mit weiteren Technologien wie Open Banking ein erhebliches Potenzial für innovative Zahlungslösungen. Derzeit steckt die Nutzung solcher Technologien durch externe Akteure, insbesondere Fintechs, noch in den Kinderschuhen. Damit Instant Payment zu einem festen Bestandteil des alltäglichen Zahlungsverkehrs wird, sind weitere technische, regulatorische und wirtschaftliche Entwicklungen notwendig. Dies umfasst nicht nur die Reifung und externe Verfügbarkeit der technischen Schnittstellen, sondern auch eine klarere regulatorische Unterstützung sowie ein aktives Engagement der Banken, den Versand von Instant Payments breiter und zu attraktiven Konditionen verfügbar zu machen.

Letztlich wird das volle Potenzial von Instant Payments nur dann ausgeschöpft, wenn alle relevanten Marktteilnehmer (Banken, Fintechs, Regulatoren und Händler) gemeinsam an einem Strang ziehen. Nur so lässt sich eine moderne, zukunftsfähige Zahlungsinfrastruktur in der Schweiz etablieren.

6. Glossar

Begriff	Beschreibung
.NET	Ein von Microsoft entwickeltes Framework zur Erstellung verschiedener Anwendungen, hier primär für das Backend-System genutzt.
Angular	Ein von Google entwickeltes TypeScript-basiertes Frontend-Framework, hier für die Checkout-Seite verwendet.
API (Application Programming Interface)	Eine Schnittstelle, die es verschiedenen Softwarekomponenten ermöglicht, miteinander zu kommunizieren.
Automapper	Eine .NET-Bibliothek, die das automatische Übertragen von Daten zwischen verschiedenen Objektmodellen (z.B. DTOs und Domänenobjekte) erleichtert.
bLink	Eine von SIX betriebene Plattform zur Standardisierung von Open Banking Schnittstellen in der Schweiz.
C4-Modell	Eine Methode zur Visualisierung von Softwarearchitekturen auf verschiedenen Abstraktionsebenen (Context, Containers, Components, Code).
Checkout Seite	Die Webseite im Bezahlprozess, auf der der Kunde die Zahlungsinformationen (z.B. QR-Code) sieht und die Zahlung auslöst.
Cron Job	Zeitgesteuerte Aufgaben, die automatisch im Hintergrund ausgeführt werden (z.B. Synchronisation von Zahlungsdaten).
DTO (Data Transfer Object)	Objekte, die dazu dienen, Daten zwischen Schichten oder Komponenten einer Anwendung zu transportieren.
EF-Core (Entity Framework Core)	Ein von Microsoft entwickeltes ORM-Framework (Object-Relational Mapper) für .NET, das die Interaktion mit Datenbanken vereinfacht.
ERD (Entity Relationship Diagram)	Ein Diagramm, das die Struktur einer Datenbank darstellt, indem es Entitäten und ihre Beziehungen zueinander visualisiert.
Händlerportal	Eine Weboberfläche, über die Händler ihre Zahlungstransaktionen verwalten, Einstellungen vornehmen und Berichte einsehen können.
Hangfire	Eine .NET-Bibliothek zur einfachen Erstellung, Ausführung und Verwaltung von Hintergrundaufgaben (wie Cron Jobs).
Imagine Banking	Eine Lern- und Entwicklungsplattform der Innofactory AG, die eine reale Bankenumgebung für Tests und Innovationen simuliert.
Innofactory AG	Ein Schweizer Unternehmen, das Innovationen im Bankensektor entwickelt und die "Imagine Banking" Plattform anbietet.
Instant Payment (IP)	Echtzeitzahlungen, bei denen Geld innerhalb von Sekunden (in der Schweiz max. 10 Sekunden) vom Sender zum Empfänger transferiert wird.
Instant Payment Bridge (IPB)	Ein Konzeptvorschlag der SIX, um Instant Payments für verschiedene Zahlungsschemes über eine standardisierte Schnittstelle zugänglich zu machen.
JWT (JSON Web Token)	Ein kompakter, URL-sicherer Standard zur Darstellung von Claims, die zwischen zwei Parteien übertragen werden, oft zur Authentifizierung/Autorisierung.
KYC (Know Your Customer)	Regulatorische Anforderung für Finanzinstitute, die Identität ihrer Kunden zu überprüfen, um Geldwäsche und Betrug vorzubeugen.
Next.JS	Ein React-basiertes Frontend-Framework, das u.a. serverseitiges Rendering ermöglicht, hier für das Händlerportal verwendet.
NOK (Nicht OK)	Bezeichnet im Kontext von Zahlungen einen fehlgeschlagenen oder nicht erfolgreichen Vorgang.
Node.js	Eine serverseitige JavaScript-Laufzeitumgebung.
Open Banking	Ein Konzept, bei dem Banken Drittanbietern über standardisierte APIs (mit Zustimmung des Kunden) Zugriff auf Kundendaten und -funktionen gewähren.
OpenAPI	Eine Spezifikation zur Beschreibung, Erstellung, Dokumentation und Nutzung von RESTful APIs.
P2M	Zahlungen, welche von einer Privatperson an einen Händler gehen.

P2P (Peer-to-Peer) Zahlungen	Direkte Geldtransfers zwischen Privatpersonen, oft über mobile Apps oder Online-Plattformen.
pain.001	Ein ISO 20022 Nachrichtenformat, das für die Initiierung von Kredittransfers (Zahlungsaufträgen) im Bankwesen verwendet wird.
Payment Scheme	Ein Regelwerk und eine Infrastruktur, die festlegen, wie Zahlungen zwischen Teilnehmern (z.B. Banken, Händlern, Kunden) abgewickelt werden.
Payment Session	Ein einzelner Bezahlvorgang, der eine erwartete Zahlung mit einem bestimmten Betrag und Status repräsentiert.
Polling	Ein Verfahren, bei dem ein System periodisch eine andere Ressource (z.B. eine API) abfragt, um auf neue Daten oder Statusänderungen zu prüfen.
PostgreSQL	Ein objektrelationales Open-Source-Datenbankmanagementsystem.
QR-Rechnung	Ein Schweizer Standard für Rechnungen mit einem QR-Code, der alle zahlungsrelevanten Informationen enthält und das Scannen vereinfacht.
Redis	Eine In-Memory-Datenbank, die oft als Cache, Message Broker oder für temporäre Datenspeicherung verwendet wird.
REST API	Eine API, die den Prinzipien des Representational State Transfer (REST) folgt, typischerweise über HTTP kommuniziert.
RTGS (Real-Time Gross Settlement)	Ein System zur Abwicklung von Großbetragszahlungen in Echtzeit und auf Bruttobasis (jede Transaktion einzeln). SIC ist das Schweizer RTGS-System.
Server-Sent Events (SSE)	Eine Technologie, die es einem Webserver ermöglicht, automatisch Aktualisierungen an einen Client über eine HTTP-Verbindung zu senden.
Shadcn/ui	Eine Sammlung wiederverwendbarer UI-Komponenten, die mit Tailwind CSS erstellt wurden, verfügbar für React und (via SpartanUi) für Angular.
SIC (Swiss Interbank Clearing)	Das Schweizer Zahlungssystem für den Interbanken-Zahlungsverkehr, betrieben von SIX. SIC5 ist die Version für Instant Payments.
SIX Group	Ein Schweizer Unternehmen, das Finanzmarktinfrasturktur betreibt, einschliesslich Zahlungsverkehrsdiensten wie SIC und bLink.
SPA (Single Page Application)	Eine Webanwendung, die nur eine einzige HTML-Seite lädt und Inhalte dynamisch aktualisiert, anstatt neue Seiten vom Server zu laden.
SSR (Server-Side Rendering)	Eine Technik, bei der Webseiten auf dem Server gerendert und als fertiges HTML an den Client gesendet werden.
Swiss QR Code	Der spezifische QR-Code-Standard, der auf Schweizer QR-Rechnungen verwendet wird.
Tailwind CSS	Ein Utility-First CSS-Framework zur schnellen Erstellung von Benutzeroberflächen.
Twint	Ein mobiles Bezahlungssystem in der Schweiz.
Vercel	Eine Cloud-Plattform für das Deployment und Hosting von Frontend-Anwendungen, insbesondere für Next.JS-Projekte.
Vier Parteien Modell	Ein Modell im Zahlungsverkehr, das typischerweise Kunde, Händler, Kundenbank (Issuer) und Händlerbank (Acquirer) umfasst.
Yuh	Eine Schweizer Mobile-Banking-App.
zod	Eine TypeScript-First-Bibliothek zur Schema-Deklaration und -Validierung.

Tabelle 13 Glossar

7. Abbildungsverzeichnis

Abbildung 1 Übersicht ausgeführter Zahlungen im Händlerportal	2
Abbildung 2 Zusammenhang SNB Girokonto und Verrechnungskonto	4
Abbildung 3 Übersichtsgrafik bLink	5
Abbildung 4 Variante Empfänger Bank benachrichtigt Payment Scheme	6
Abbildung 5 Variante Payment Scheme holt sich Zahlungsinformationen ab	7
Abbildung 6 Variante Schnittstelle bei Senderbank	8
Abbildung 7 Variante Instant Payment Bridge	9

Abbildung 8 Vier Parteien Modell mit zwei Empfängerbanken	10
Abbildung 9 Ablauf Twint Überweisung zwischen zwei Personen mit unterschiedlichen Banken	11
Abbildung 10 Ablauf einer Zahlung	12
Abbildung 11 Beispiel QR-Rechnung inkl. Swiss QR Code	13
Abbildung 12 Zustandsdiagramm Statuskonzept	14
Abbildung 13 Einfache Darstellung Zusammenspiel Instant Payment und Open Banking	15
Abbildung 14 ERD Schema Instant Pay	20
Abbildung 15 C4 Diagramm Level 1	21
Abbildung 16 C4 Diagramm Level 2	22
Abbildung 17 Diagram Clean Code	23
Abbildung 18 Open Banking Schnittstelle Request Authentifizierung	24
Abbildung 19 Open Banking Schnittstelle Response Transaktionen	24
Abbildung 20 Open Banking Schnittstelle Response Transaktion	25
Abbildung 21 Open Banking Schnittstelle Request Zahlungserfassung	26
Abbildung 22 Screenshot Händlerportal Dashboard	29
Abbildung 23 Beispiel eines QR-Stickers von Instant Pay Local	30
Abbildung 24 Bezahlprozess Instant Pay Local	30
Abbildung 25 Übersicht Bezahlprozess Instant Pay Now	31
Abbildung 26 Instant Pay Now Eingabemaske	31
Abbildung 27 Instant Pay Checkout Seite	32
Abbildung 28 Bestätigungsseite Zahlung Instant Pay Now	33
Abbildung 29 Fehlgeschlagene Zahlung Instant Pay Now	33
Abbildung 30 Bezahlprozess Instant Pay Web	33
Abbildung 31 Instant Pay Web Bezahlprozessschritt 1 (Webshop)	34
Abbildung 32 Instant Pay Web Bezahlprozessschritt 2 (Instant Pay)	34
Abbildung 33 Instant Pay Web Bezahlprozessschritt 3 (Webshop)	34
Abbildung 34 Request erstellen Payment Session über REST API	35
Abbildung 35 Response erstellen Payment Session über REST API	35
Abbildung 36 Response Status Payment Session REST API	36
Abbildung 37 Beispiel eines JWT ApiTokens (Screenshot jwt.io)	37
Abbildung 38 Sequenzdiagramm JWT Token während Zahlungssession	38
Abbildung 39 Screenshot Instant Pay Checkout Seite	39
Abbildung 40 Instant Pay Zahlung in Mobile Banking importieren	40
Abbildung 41 Händlerinfos Auszahlungsarten	41
Abbildung 42 Overlay Sammelkonto Auszahlung	42
Abbildung 43 Beispiel application/problem+json	42
Abbildung 44 Screenshot Test-Coverage Händlerportal	46
Abbildung 45 Screenshot Test-Coverage Checkout Seite	47
Abbildung 46 Screenshot Test-Coverage Backend	47
Abbildung 47 Verteilung der Personen für Befragung Schweizer Zahlungsmethoden	50
Abbildung 48 Charts beliebteste Zahlungsmethoden in Geschäften	50
Abbildung 49 Charts beliebteste Zahlungsmethoden in Onlineshops	51
Abbildung 50 Chart Offenheit für neue Zahlungsmethoden	51
Abbildung 51 Variante Empfängerbank benachrichtigt Payment Scheme	54

8. Tabellenverzeichnis

Tabelle 1 Funktionale Anforderungen	18
Tabelle 2 Nicht Funktionale Anforderung 1	18
Tabelle 3 Nicht Funktionale Anforderung 2	18
Tabelle 4 Nicht Funktionale Anforderung 3	18

Tabelle 5 Nicht Funktionale Anforderung 4.....	18
Tabelle 6 Nicht Funktionale Anforderung 5.....	19
Tabelle 7 Verwendete Tools und Technologien.....	28
Tabelle 8 Manuelle Testfälle	45
Tabelle 9 Testresultate manuelle Testfälle	45
Tabelle 13 Usability Test.....	48
Tabelle 14 Transaktionsgebühren Versand Instant Payment.....	55
Tabelle 15 Geplante Einführung IP-Versand für weitere Banken	56
Tabelle 16 Glossar	62

9. Anhänge

Eigenständigkeitserklärung

Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selbst und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit der Betreuerin / dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe,
- dass ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum:

Rapperswil, 13.06.2025

Name, Unterschrift:

Jan Meier

OST – Ostschweizer Fachhochschule, Rapperswil-Jona