

Objektorientierte Analyse einer Tracking App für Nutzungsprofile von E-Scootern

im Studiengang Informatik

an der DHBW Ravensburg
Campus Friedrichshafen

zur Vorlesung
Software Engineering 2

10.12.2021

Bearbeitungszeitraum:	22.11.2021 - 10.12.2021		
Name, Matrikelnummer, Kurs:	Danhofer, David	9821366	TIM19
	Gurraj, Klaudio	2047736	TIS19
	Mayer, Tobias	8439587	TIM19
	Moya Jiménez, Eloy	6620737	TIM19
	Muhametcin, Vadim	7408773	TIS19
	von Langsdorff, Amelie	9154670	TIS19
	Welti, Andreas	4835207	TIM19
	Zedler, Nicolas	2651995	TIS19
Gutachter der Dualen Hochschule:	Prof. Dr. Andreas Judt		

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
1 Systemidee	1
2 Ablaufbeschreibung der Systemidee	3
3 Geschäftsfälle	5
4 Fachklassenmodell	8
4.1 Beziehungsdigramm	8
4.2 Fachklassendiagramm	9
4.3 Beschreibung der Fachklassen	11
5 Systemablaufmodell	12
6 Technische Machbarkeit	18
6.1 Erfassung der Ausschlusskriterien	18
6.2 Anwendung der Kriterien	19
Glossar	21
Literatur	30
A Anhang - Durchführung der Technologieentscheidung	31

Abkürzungsverzeichnis

AMQP	Advanced Message Queing Protocol
eSS	enhanced Scooter Service
MQTT	Message Queuing Telemetry Transport
OOA	Objektorientierte Analyse

Abbildungsverzeichnis

4.1	Beziehungsdiagramm mit Kardinalitäten	8
4.2	Fachklassenmodell des eSS	10
5.1	Gesamtablauf der Geschäftsfälle	12
5.2	Buchungsinformationen einer Fahrt erfassen	13
5.3	E-Scooter-Daten erfassen	13
5.4	E-Scooter-Status bereitstellen	14
5.5	Informationen der Beendigung einer Fahrt erfassen	15
5.6	Informationen einer Fahrt bereitstellen	16
5.7	Analysedaten erfassen	17
5.8	Analysedaten bereitstellen	17
6.1	Schichtenmodell zur Untergliederung verschiedener Technologien nach Ab- hängigkeit und Abstraktionsniveau	18

1 Systemidee

Im Rahmen dieser Objektorientierten Analyse (OOA) soll das Softwaresystem *enhanced Scooter Service* (eSS) konzeptioniert werden.

Ziel und Aufgabe von eSS ist es, das Anliegen des Auftraggebers, sein Angebot über die Analyse der Nutzung der E-Scooter zu verbessern, zu verwirklichen.

Entsprechend dieser Anforderung schlagen wir, die *Alpine Software Solutions*, das nachfolgend beschriebene Softwaresystem vor, welches den Prozess, die jeweilige Nutzung der E-Scooter zu erfassen, abdeckt. Das von der *Alpine Software Solutions* konzipierte System eSS ermittelt die Nutzungsdaten eines E-Scooters mittels vom Verleihsystem erfasster Daten und stellt sie anschließend einem noch nicht spezifizierten externen System, dem Analysesystem, zur Verfügung. Das Analysesystem führt eine Analyse anhand der Nutzungsdaten durch und übergibt das Ergebnis anschließend an eSS, welches dieses speichert und dem E-Scooter-Verleih bereitstellt.

eSS erfasst grundsätzlich jede Fahrt eines E-Scooters und deckt damit die gesamte Nutzung der E-Scooter ab. Um die Analyse der Nutzdaten nicht einzuschränken und das System möglichst flexibel zu gestalten, wird periodisch der Status eines E-Scooter erfasst und anschließend dem Analysesystem bereitgestellt. Sämtliche Daten sollen persistent in eSS gespeichert werden, was eSS erweiterbar und flexibel gestaltet.

Alpine Software Solutions schlägt für die Bereitstellung der Daten für Analyse- und Verleihsystem ein Konzept vor, welches einen Buffer nutzt, um diesen Systemen einen unabhängigen und einfachen Datenzugriff zu ermöglichen. Zudem macht die Nutzung eines Buffers eSS skalierbar, sodass das zu verarbeitende Datenvolumen flexibel an vorgegebene Anforderungen angepasst werden kann.

Voraussetzungen an den E-Scooter-Verleih

Es wird vorausgesetzt, dass der E-Scooter-Verleih ein bestehendes Kontingent an E-Scootern besitzt. Diese dem Kontingent bereits angehörigen, ebenso wie später hinzugefügte E-Scooter müssen wiederum die folgenden Voraussetzungen erfüllen:

Die E-Scooter benötigen Sensoren zum Ermitteln des Akkustandes, der Akkugüte und des Standorts des E-Scooters, sowie eine stets stabile Internetverbindung.

Es wird angenommen, dass dem Verleihsystem zu jedem Zeitpunkt die Daten der genannten Sensoren zur Verfügung stehen, und diese eSS nach einer Anbindung an das Verleihsystem übergeben werden können.

Zudem wird vorausgesetzt, dass der E-Scooter-Verleih bei der Buchung einer Fahrt durch einen Nutzer Kenntnis über dieses Ereignis, den Nutzer, den Zeitpunkt der Buchung sowie den gebuchten E-Scooter besitzt.

Entwicklungsstufen

Die erste Version des von eSS geplanten Systems soll die vollständige Erfassung, Speicherung und anschließende Bereitstellung der Daten wie oben beschrieben umsetzen.

Eine zweite Entwicklungsstufe kann um Zustände wie beispielsweise das Laden eines E-Scooters oder eine stattfindende Wartung ergänzt werden, um noch umfassendere und granularere Analysen durch das Analysesystem zu ermöglichen.

In einer dritten Entwicklungsstufe kann die dauerhafte und persistente Speicherung aller Daten in eSS genutzt werden, um dem Verleihsystem und dem Analysesystem Abfragen zu ermöglichen. So können beliebige Daten aufgerufen werden, um die Flexibilität der externen Systeme noch weiter zu erhöhen.

Kosten

Für die Gesamtkosten der ersten Version von eSS wird ein durchschnittlicher Stundensatz von 250€ berechnet. Dabei sind bereits der Gewinn und die anfallenden Steuern einberechnet. Der Arbeitsaufwand wird auf etwa 160 Manntage (aufgeteilt auf acht Personen) à 8 Stunden geschätzt. Somit belaufen sich die Kosten für eSS auf 320.000€.

2 Ablaufbeschreibung der Systemidee

Folgende Daten der E-Scooter werden von eSS periodisch (z. B. jede Sekunde einmal) vom Verleihsystem bezogen und gespeichert:

Der eindeutige Identifikator eines E-Scooters, der absolute Standort des E-Scooters, der absolute Zeitpunkt, der Akkustand und die Akkugüte des E-Scooters. Die Frequenz, mit der eSS die fahrtbezogenen Daten vom E-Scooter-Verleih bezieht, kann je nach Kapazität und gewünschter Granularität der Daten gewählt werden.

eSS soll so konzipiert werden, dass es unabhängig vom Datentyp der bezogenen Daten des Verleih- und Analysesystems ist. So wird die Flexibilität des Systems sichergestellt und dem Verleih- und Analysesystem Unabhängigkeit ermöglicht.

Es wird vorausgesetzt, dass das Verleihsystem die Information des Beginns einer Fahrt, den Identifikator des E-Scooters, das Nutzerprofil, den vom Nutzer gewählten Tarif und den Zeitpunkt der Buchung an eSS übermittelt. Diese Daten werden in eSS gespeichert. Beim Beenden einer Fahrt wird angenommen, dass eSS die Information, dass die Fahrt beendet wurde, vom Verleihsystem gemeinsam mit dem Zeitpunkt und dem Identifikator des E-Scooters erhält, um die Daten anschließend zu speichern.

Nach einer abgeschlossenen Fahrt stehen somit zentral und gesammelt alle Informationen zur Verfügung, die eSS für die Ermittlung aller Daten in Bezug auf eine Fahrt benötigt. eSS berechnet hierzu im Anschluss einer Fahrt die zurückgelegte Route, die zurückgelegte Distanz, die Dauer der Fahrt, die durchschnittliche Geschwindigkeit und den Akkuverbrauch des E-Scooters während der Fahrt. Diese Informationen werden gemeinsam mit dem Identifikator des E-Scooters, dem Nutzerprofil, dem Start- und Zielstandort sowie dem Start- und Endzeitpunkt der Fahrt persistent in eSS gespeichert und anschließend dem Analysesystem bereitgestellt.

Sobald eSS, wie oben beschrieben, die Daten eines E-Scooters vom Verleihsystem erhält, wird außerdem der Status eines E-Scooters ermittelt.

Der Status eines E-Scooters beinhaltet den absoluten Zeitpunkt, den eindeutigen Identifikator des E-Scooters, den absoluten Standort des E-Scooters, die Geschwindigkeit des E-Scooters, sowie den Akkustand und die Akkugüte des E-Scooters. Außerdem beinhaltet der Status des E-Scooters die Information, ob ein E-Scooter aktuell verliehen ist, und, falls ja, das Nutzerprofil des aktuellen Nutzers.

Der Status des E-Scooters wird ebenfalls persistent in eSS gespeichert und dem Analysesystem bereitgestellt.

Das externe Analysesystem führt auf Basis der von eSS zur Verfügung gestellten Informationen Analysen durch, die anschließend eSS bereitgestellt werden. eSS erfasst diese Daten und speichert sie persistent, um sie wiederum dem Verleihsystem bereitzustellen.

3 Geschäftsfälle

Name	E-Scooter-Daten erfassen
Anwendungsfall-Typ	Geschäftsfall
Kurzbeschreibung	Der eindeutige Identifikator, der absolute Standort, der absolute Zeitpunkt, der Akkustand und die Akkugüte werden vom Verleihsystem bereitgestellt und von eSS erfasst und gespeichert.
Auslöser	periodisch
Ergebnis	Die Daten eines E-Scooters wurden erfasst und persistent gespeichert.
Akteure	Verleihsystem, eSS

Name	Buchungsinformationen einer Fahrt erfassen
Anwendungsfall-Typ	Geschäftsfall
Kurzbeschreibung	Sobald eine Fahrt über das Verleihsystem gebucht wurde, wird diese Information inklusive dem Zeitpunkt der Buchung, dem eindeutigen Identifikator des E-Scooters und dem Nutzerprofil übergeben, von eSS erfasst und gespeichert.
Auslöser	Buchung einer Fahrt
Ergebnis	Die Buchungsinformationen einer Fahrt wurden erfasst und persistent gespeichert.
Akteure	eSS, Verleihsystem

Name	Informationen der Beendigung einer Fahrt erfassen
Anwendungsfall-Typ	Geschäftsfall
Kurzbeschreibung	Sobald eine Fahrt über das Verleihsystem beendet wurde, wird diese Information inklusive dem Zeitpunkt der Beendigung und dem eindeutigen Identifikator des E-Scooters übergeben, von eSS erfasst und gespeichert.
Auslöser	Beendigung einer Fahrt
Ergebnis	Die Informationen der Beendigung einer Fahrt wurden erfasst und persistent gespeichert.
Akteure	eSS, Verleihsystem

Name	Status eines E-Scooters bereitstellen
Anwendungsfall-Typ	Geschäftsfall
Kurzbeschreibung	Der Status eines E-Scooters wird ermittelt, persistent gespeichert und dem Analysesystem bereitgestellt. Der Status beinhaltet den eindeutigen Identifikator des E-Scooters, den absoluten Standort, den absoluten Zeitpunkt, den Akkustand, die Akkugüte, die Geschwindigkeit, die Information, ob ein Scooter aktuell verliehen ist, und, sofern verliehen, das Nutzerprofil des aktuellen Nutzers.
Auslöser	periodisch
Ergebnis	Status des E-Scooters wurde ermittelt, persistent gespeichert und dem Analysesystem bereitgestellt.
Akteure	eSS, Analysesystem

Name	Informationen einer Fahrt bereitstellen
Anwendungsfall-Typ	Geschäftsfall
Kurzbeschreibung	Die Informationen einer Fahrt werden ermittelt, gespeichert und dem Analysesystem bereitgestellt. Diese fahrtenbezogenen Informationen beinhalten den Identifikator des E-Scooters, die zurückgelegte Route, die zurückgelegte Distanz, die Dauer, die durchschnittliche Geschwindigkeit, den Akkuverbrauch, das Nutzerprofil, den Start- und Zielstandort und den Start- und Endzeitpunkt einer Fahrt.
Auslöser	Fahrt wurde beendet
Ergebnis	Daten der Fahrt wurden erfasst und dem Analysesystem bereitgestellt.
Akteure	eSS, Analysesystem

Name	Analysedaten erfassen
Anwendungsfall-Typ	Geschäftsfall
Kurzbeschreibung	Die Analysedaten der Nutzung der E-Scooter wird vom Analysesystem bereitgestellt, von eSS erfasst und persistent gespeichert.
Auslöser	Analysedaten wurden bereitgestellt
Ergebnis	Das Ergebnis der Analyse der Nutzungsdaten wird erfasst und persistent gespeichert.
Akteure	eSS, Analysesystem

Name	Analysedaten bereitstellen
Anwendungsfall-Typ	Geschäftsfall
Kurzbeschreibung	Die Analysedaten des externen Analysesystems werden dem Verleihsystem bereitgestellt.
Auslöser	Analysedaten wurden erfasst.
Ergebnis	Die Analysedaten wurden dem Verleihsystem bereitgestellt.
Akteure	eSS, Verleihsystem

4 Fachklassenmodell

4.1 Beziehungendiagramm

Die folgende Abbildung zeigt die Beziehungen zwischen den für die Softwarearchitektur erforderlichen Klassen und ihre dazugehörigen Kardinalitäten. Mit Hilfe des Konzepts kann die spätere Umsetzung erleichtert und auf Korrektheit überprüft werden.

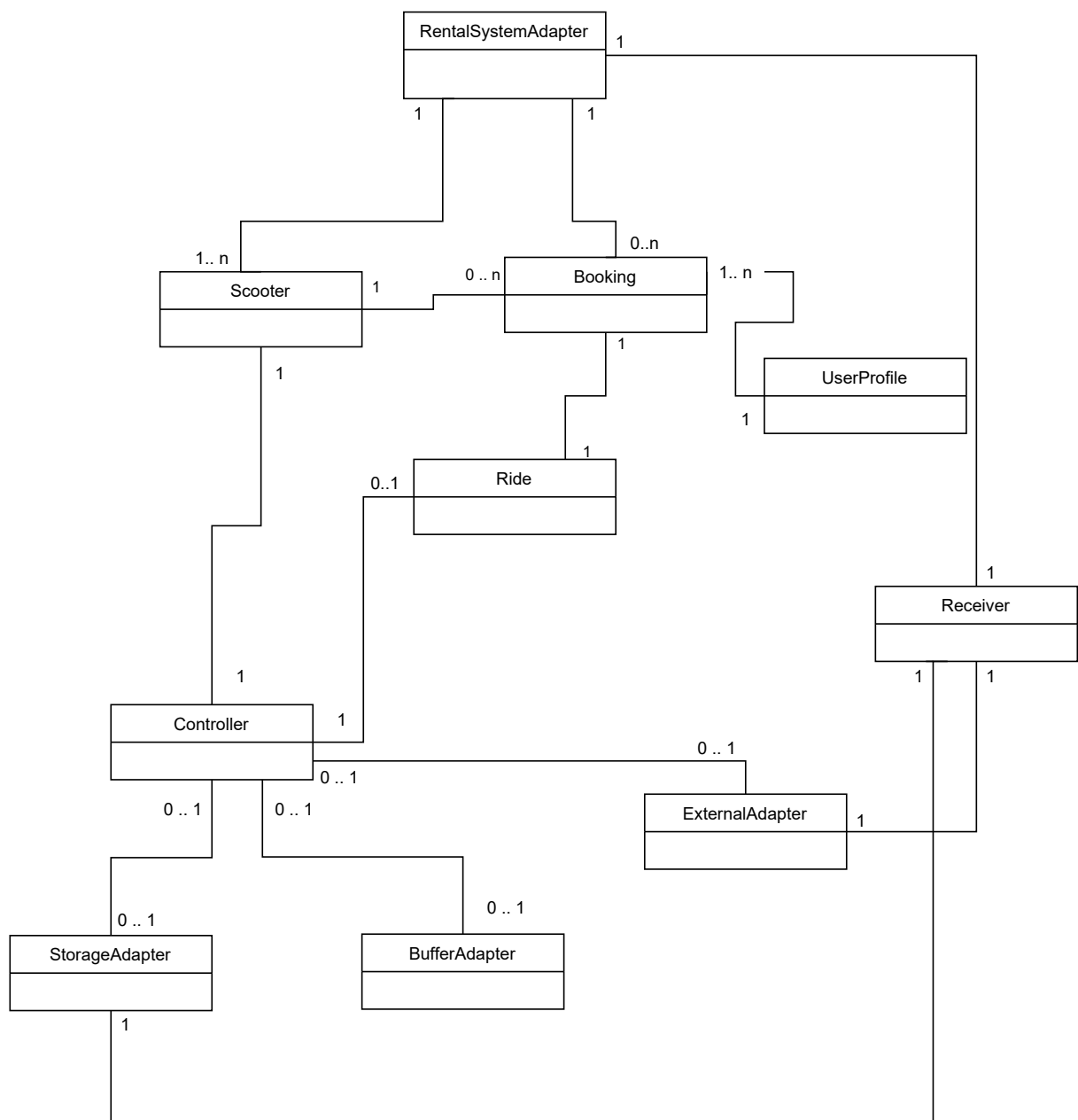


Abb. 4.1: Beziehungendiagramm mit Kardinalitäten

4.2 Fachklassendiagramm

Das folgende Fachklassendiagramm zeigt die benötigten Klassen mit ihren Attributen und Operationen für die Umsetzung des eSS. Bei der Modellierung wurden Design Patterns berücksichtigt, um eine Entkopplung der Komponenten zu verwirklichen. Durch die Entkopplung soll eine bessere Wartbarkeit, Erweiterbarkeit und die Qualität der Software sichergestellt werden.

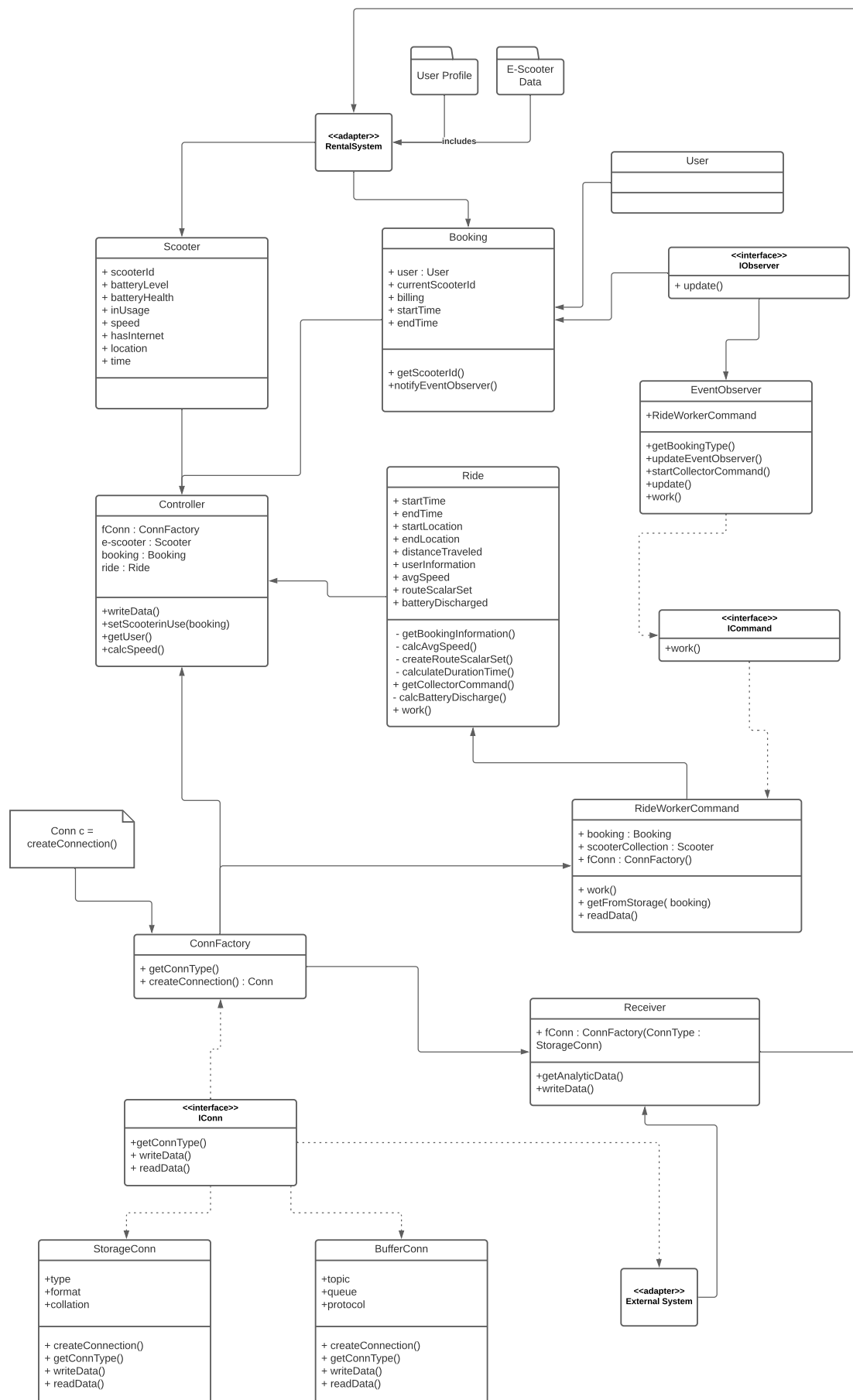


Abb. 4.2: Fachklassenmodell des eSS

4.3 Beschreibung der Fachklassen

Adapter: RentalSystem

Der RentalSystem-Adapter stellt eSS Informationen über die E-Scooter und Nutzerprofile zur Verfügung für die Erstellung der „Scooter“- und „Booking“-Objekte.

Klasse: Booking

Die Klasse Booking enthält Informationen über den Tarif, Start- und Endzeitpunkt einer Buchung, sowie das Nutzerprofil.

Klasse: Scooter

Die Klasse Scooter enthält die bereitgestellten E-Scooter-Daten als Attribute. Zu den Attributen gehören unter anderem folgende E-Scooter-Daten: Identifikator, Akkustand, Akkugüte, Geschwindigkeit und Standort.

Klasse: Ride

Die Klasse Ride wird bei Beendigung einer Fahrt erstellt und repräsentiert ein Objekt für eine Fahrt. Das Objekt enthält Attribute wie den Start, das Ziel, die anhand der Wegpunkte berechnete Route, die durchschnittliche Geschwindigkeit und den Akkuverbrauch.

Klasse: Controller

Die Klasse Controller verwaltet die Objekte Scooter, Booking, Ride und ConnFactory. Es enthält die Logik für das Speichern und Bereitstellen der Objekte.

Klasse: ConnFactory

Die Klasse ConnFactory ist für die Initiierung der Verbindung zum Zielsystem zuständig. Es ist eine Abstrahierung um die passende Verbindung zum Zielsystem zu erhalten.

Klasse: Receiver

Die Klasse Receiver erhält Daten aus dem Analysesystem, speichert die Analysedaten mit Hilfe von ConnFactory und stellt sie dem Adapter des Verleihsystems (RentalSystem) zur Verfügung.

Adapter: External System

Der Adapter stellt die Kommunikation von eSS zum Analysesystem her.

5 Systemablaufmodell

Abbildung 5.1 stellt den Ablauf aller Geschäftsfälle dar. Hierbei werden insbesondere die Beziehungen zwischen den Geschäftsfällen und den beteiligten Systemen aufgezeigt. Innerhalb des Systems gibt es auf der Ebene der Geschäftsfälle keine fachlichen Ausnahmen, die berücksichtigt werden könnten. Daher zeigt das Diagramm gleichzeitig den vollständigen Ablauf, als auch den Standardablauf des Systems.

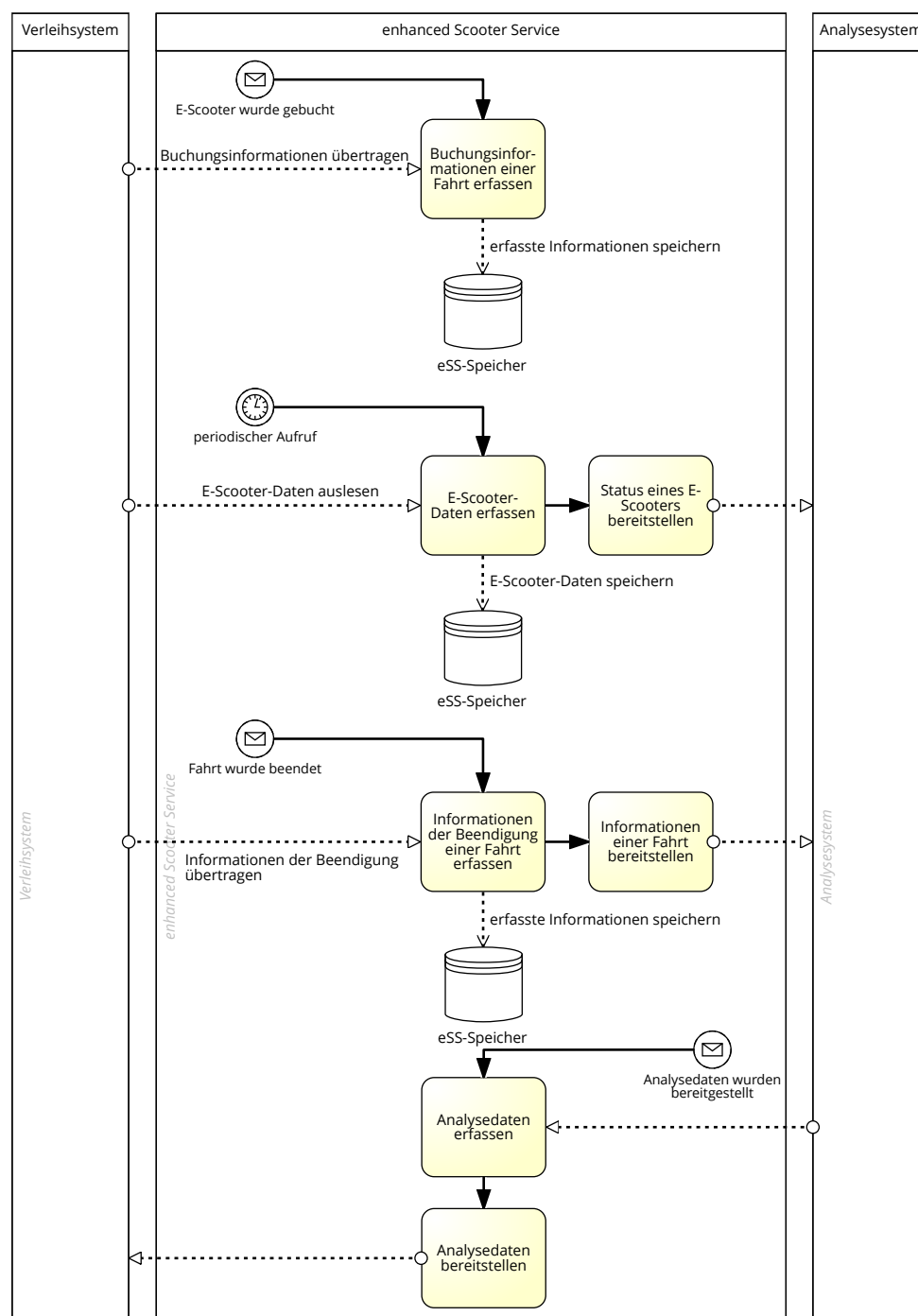


Abb. 5.1: Gesamtablauf der Geschäftsfälle

Die nachfolgenden sieben Diagramme 5.2 bis 5.8 zeigen die Systemablaufmodelle der einzelnen Geschäftsfälle.

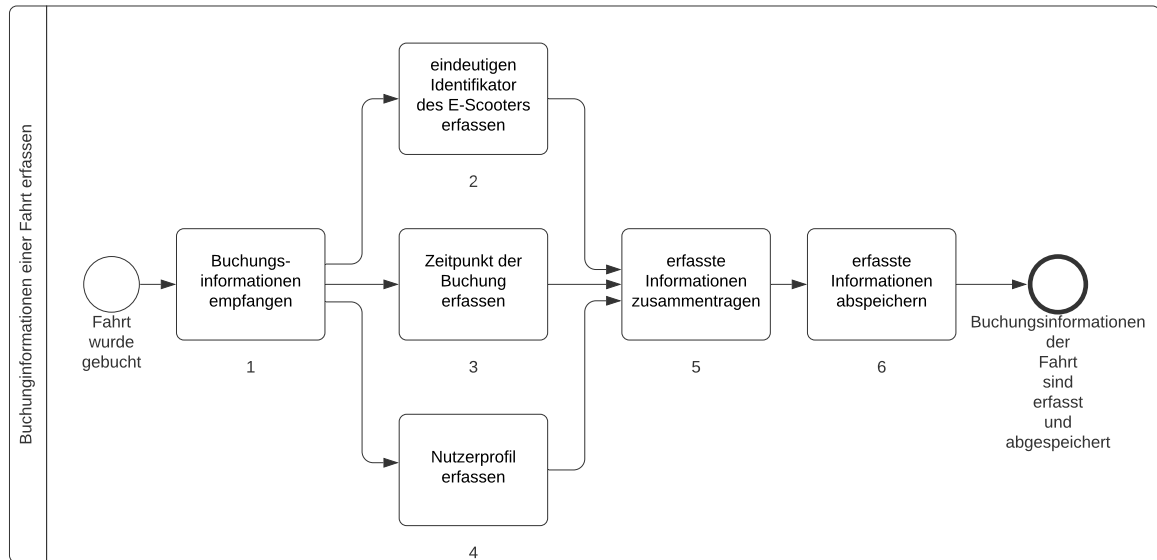


Abb. 5.2: Buchungsinformationen einer Fahrt erfassen

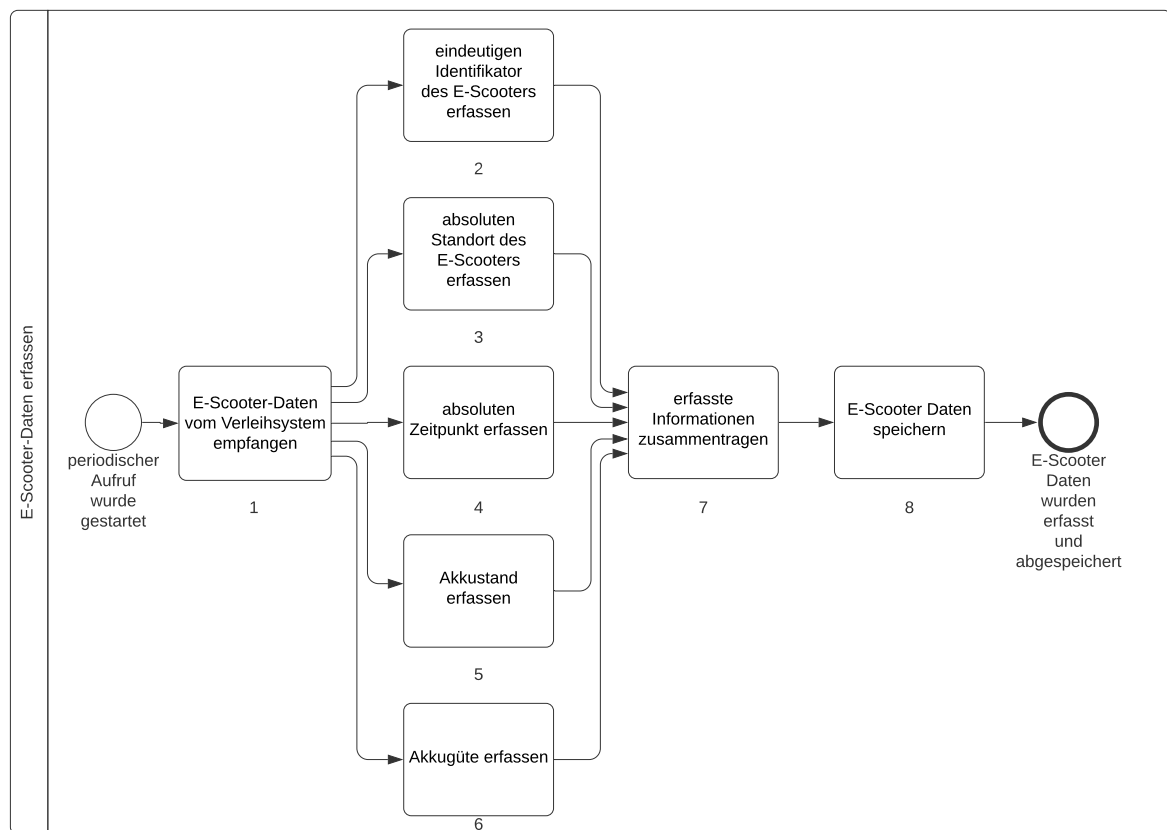


Abb. 5.3: E-Scooter-Daten erfassen

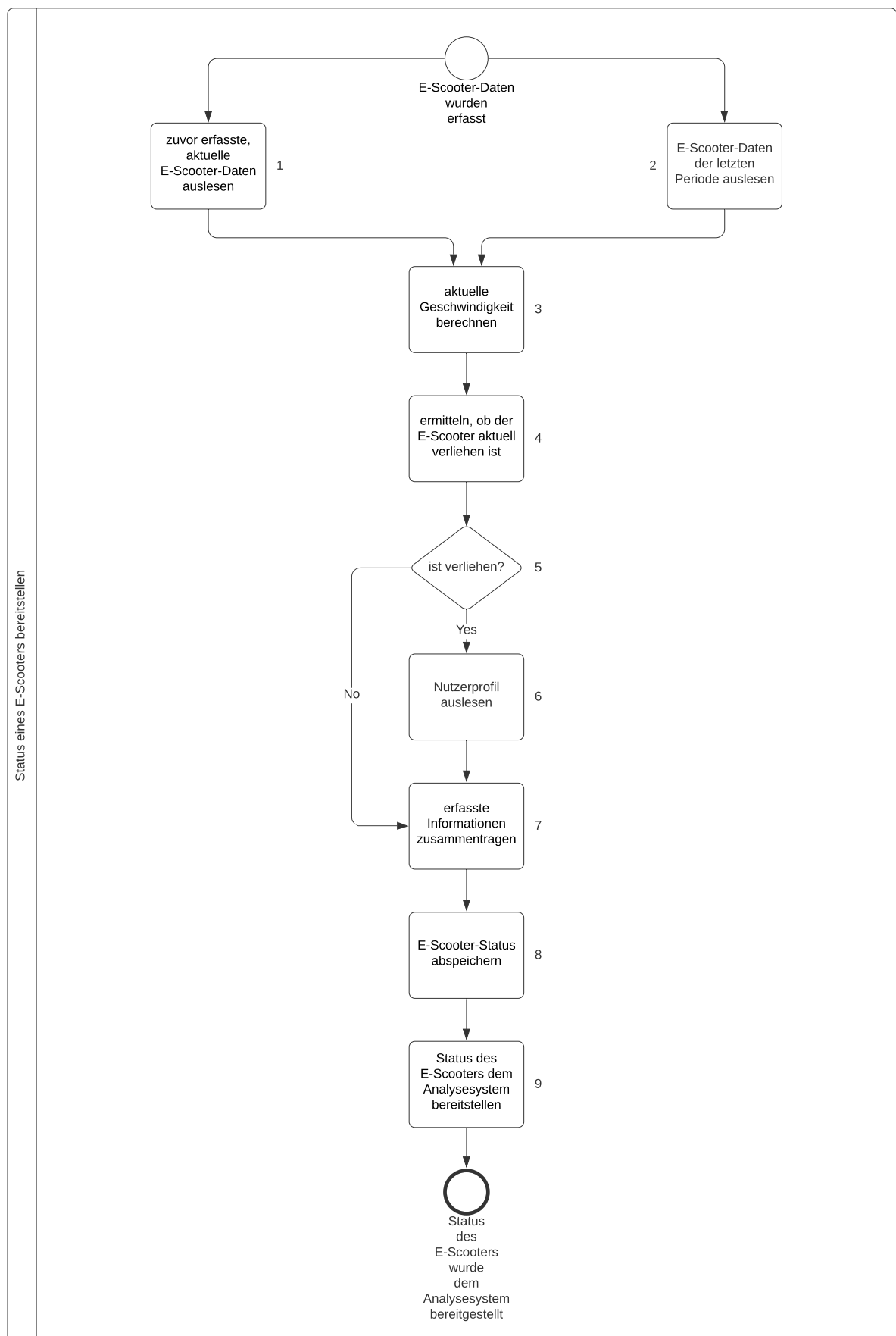


Abb. 5.4: E-Scooter-Status bereitstellen

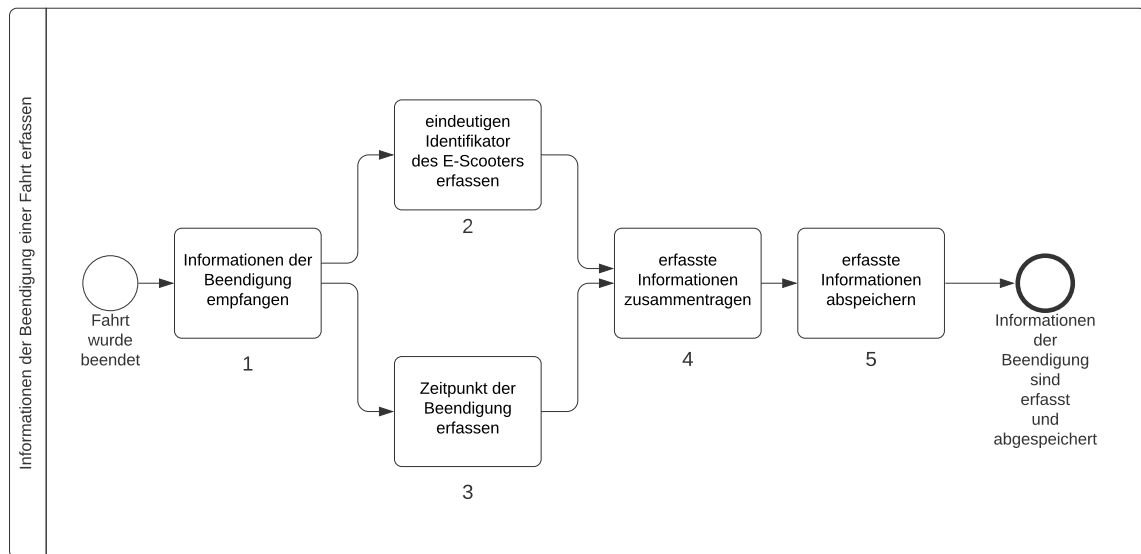
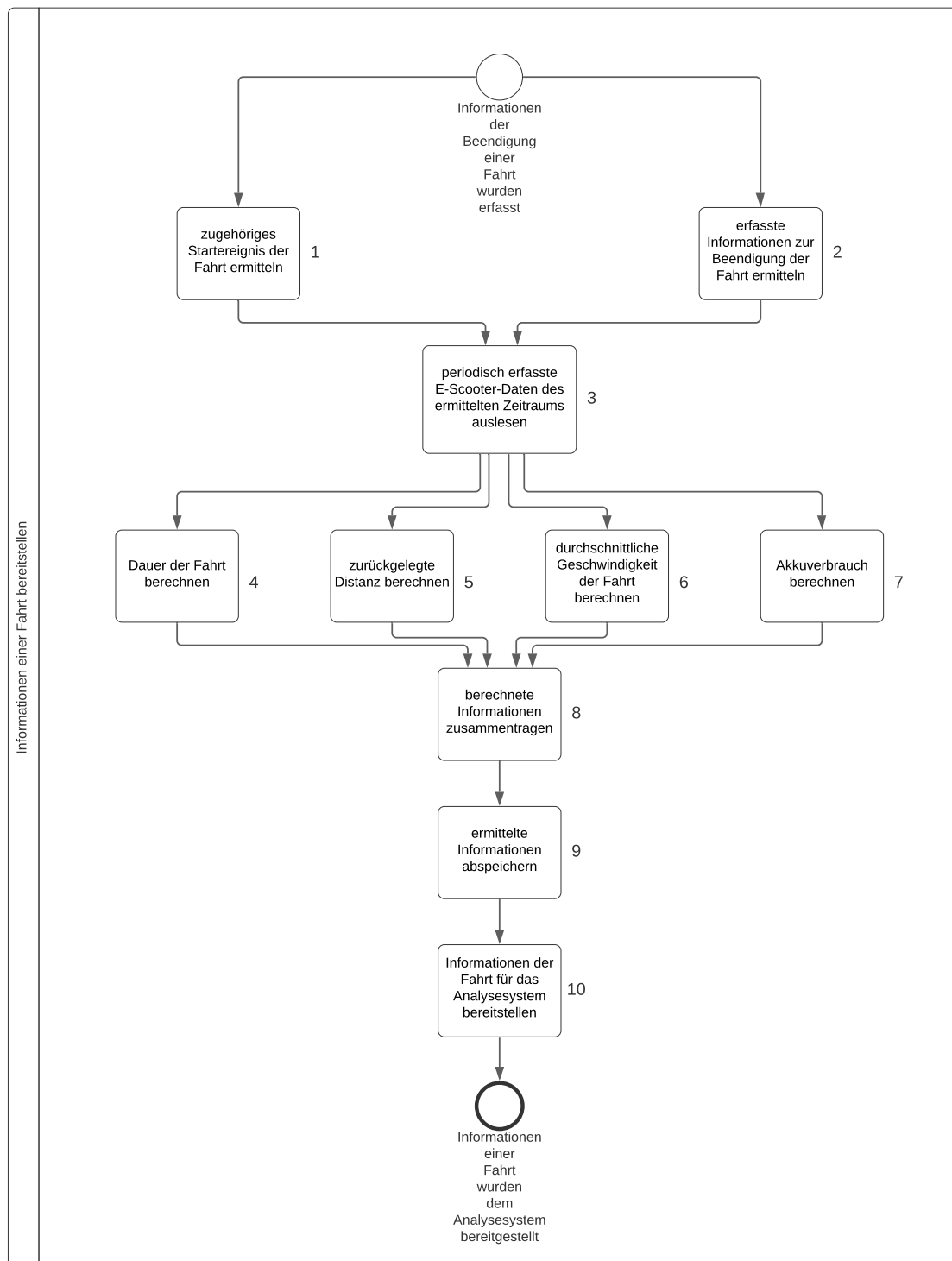


Abb. 5.5: Informationen der Beendigung einer Fahrt erfassen

**Abb. 5.6:** Informationen einer Fahrt bereitstellen

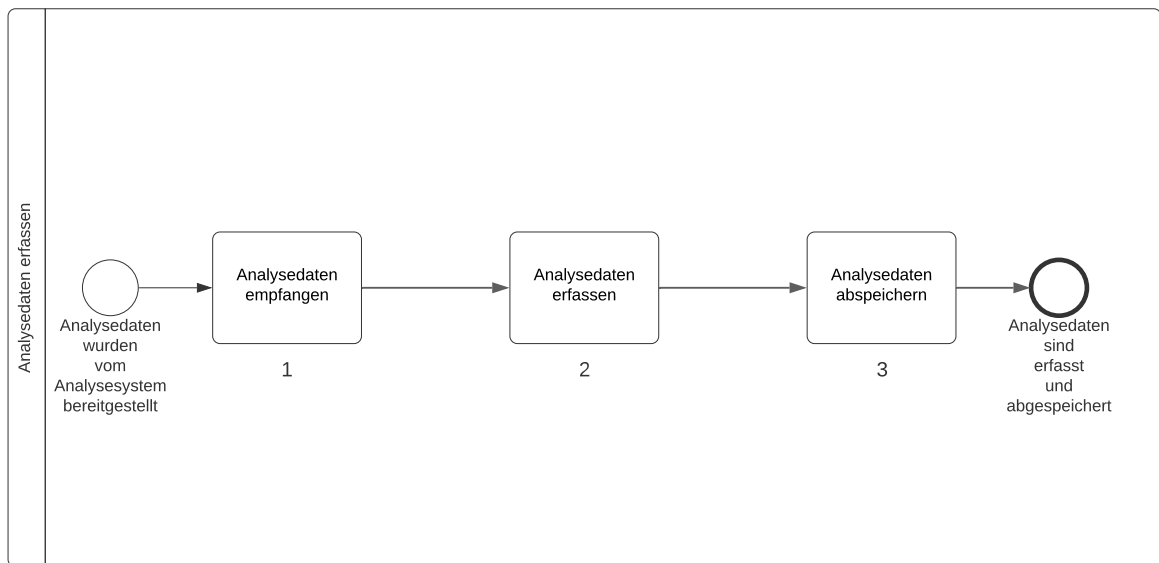


Abb. 5.7: Analysedaten erfassen

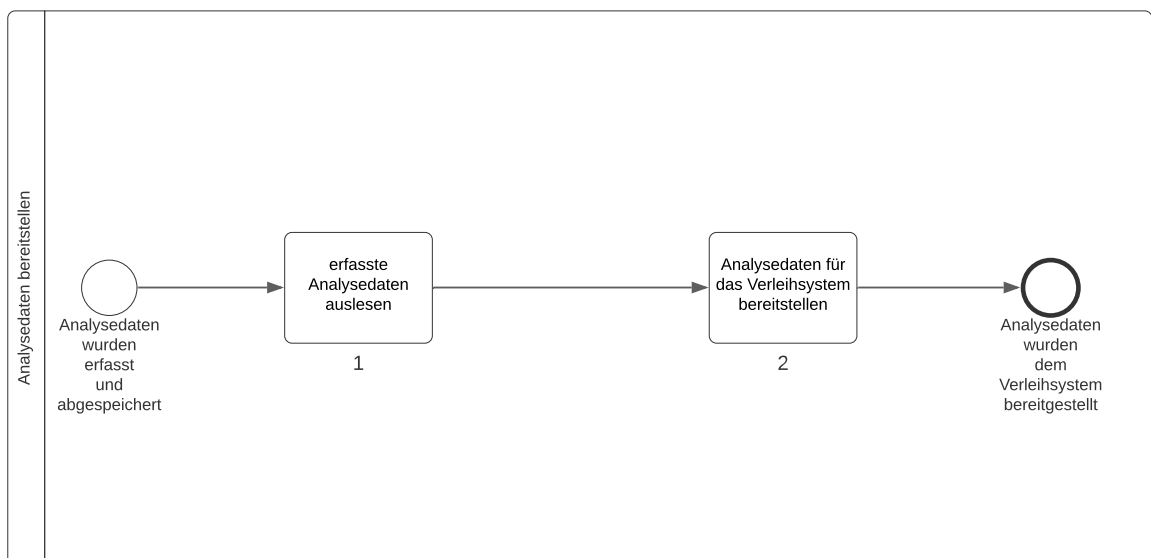


Abb. 5.8: Analysedaten bereitstellen

6 Technische Machbarkeit

Die technische Machbarkeit unseres Angebots soll nachfolgend anhand aller relevanten Parameter nachgewiesen werden. Um die Analyse der technischen Machbarkeit zu strukturieren, werden zwei Verfahren angewandt.

Zum einen sollen zur Vereinfachung der Analyse vorab verschiedene Ausschlusskriterien definiert werden, die den Raum der in Frage kommenden Technologien einschränken. Diese Ausschlusskriterien werden von den fünf wesentlichen Einflussphären Entwickler/-Zulieferer, Unternehmen, Markt, Zeit und dem rechtlichen Rahmen abgeleitet.

Zum anderen wird der Raum an technischen Komponenten gemäß Abb. 6.1 entlang der Abhängigkeitsbeziehung der Komponenten und deren Abstraktionsniveau strukturiert. Dadurch kann sukzessive die Anzahl an relevanten Technologien limitiert werden und die relevanten Ausschlusskriterien können systematisch an den jeweiligen Entscheidungspunkten der Architektur berücksichtigt werden.

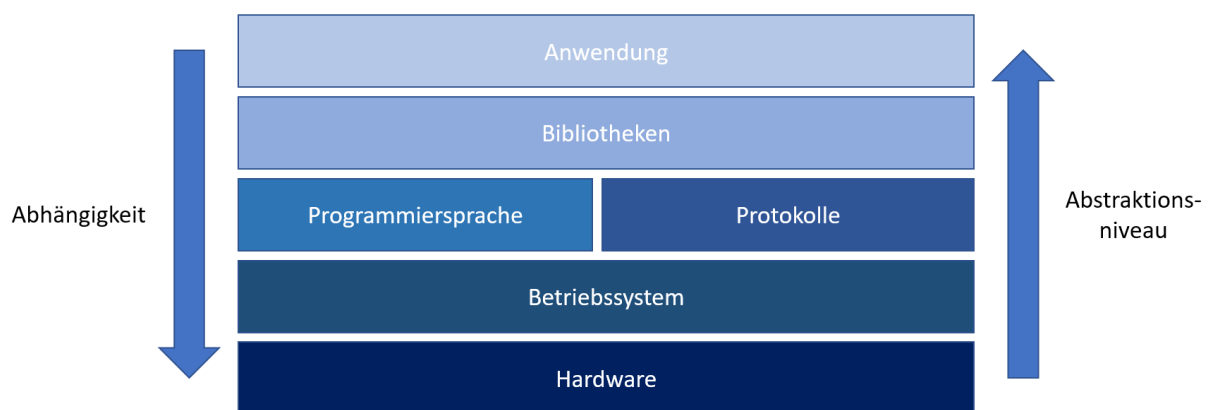


Abb. 6.1: Schichtenmodell zur Untergliederung verschiedener Technologien nach Abhängigkeit und Abstraktionsniveau

6.1 Erfassung der Ausschlusskriterien

Nachfolgend sollen alle Ausschlusskriterien definiert werden, die sich aus den Interessen der fünf Einflussphären Entwickler/Zulieferer, Unternehmen, Markt, Zeit und rechtlicher Rahmen ableiten lassen.

Das erste Ausschlusskriterium ist eine **weite Verbreitung** der anzuwendenden Technologie. Dies begründet sich dadurch, dass weiter verbreitete Technologien in der Regel von mehr Unternehmen unterstützt werden und so die Abhängigkeit von einzelnen Marktteilnehmern minimiert werden kann.

Die Technologie muss einen **hohen Reifegrad** aufweisen. Dadurch kann die hohe Aussagekraft der hier durchgeführten Analyse gestützt werden und verhindert werden, dass Neuanpassungen in der Umsetzung erforderlich werden. Die Reife kann durch die Berücksichtigung der Kriterien Alter, Nutzerzahlen und Nutzerfeedback angenähert werden.

Die Technologie soll an sich **unabhängig von den Entscheidungen einiger weniger Entscheidungsträger** sein. Dies ist erforderlich, um sicherzugehen, dass der Technologiestack nicht von den unberechenbaren Aktionen einzelner Marktteilnehmer abhängt. Im Allgemeinen sind deswegen Open-Source-Technologien, die durch Communities mit vielen Entscheidungsträgern entwickelt werden, proprietären Technologien, die häufig von einzelnen Unternehmen entwickelt werden, vorzuziehen.

Wichtig ist außerdem, dass Technologien im gewählten Stack **untereinander möglichst unabhängig** sind, um die Austauschbarkeit und Unabhängigkeit der Komponenten untereinander maximal zu gewährleisten. Dadurch kann die Austauschbarkeit der einzelnen Komponenten im Bedarfsfall maximiert und der Anpassbedarf des Systems minimiert werden.

Auf die Software ist **geltendes Recht** anzuwenden. Dies bedeutet u. a., dass die Lizenzen eine kommerzielle Nutzung erlauben müssen und der geltende Datenschutz zu gewährleisten ist. Wichtig ist daher, dass alle Ebenen, die direkt mit der Verarbeitung von personenbezogenen Daten befasst sind, die Möglichkeit der Löschung vorsehen. Das sind alle Technologien, die den Layern drei (Protokolle) bis fünf (Anwendung) zugerechnet werden.

Letztlich muss sichergestellt werden, dass für die verwendeten Technologien eine **Leistungsfähigkeit** angenommen werden kann, die einer realistischen Schätzung der Belastung standhält.

6.2 Anwendung der Kriterien

Die erarbeiteten Kriterien wurden auf die Schichten des Schichtenmodells in Abb. 6.1 systematisch angewandt. Mit jeder Stufe werden dabei die Möglichkeiten zur Implementierung der Anwendung eingeschränkt und präzisiert.

Somit konnte es gelingen einen Pool verschiedener, exemplarischer zur Verfügung stehender Technologie zu einer einzigen Technologieempfehlung zu reduzieren.

Die detaillierte Vorgehensweise findet sich im Anhang wieder.

Wir empfehlen für die Umsetzung unseres Konzepts die Entwicklung einer Applikation

in *Python 3.8*, die für die Erfüllung der Speicheranforderungen auf den freien *pyODBC*-Driver zur Verknüpfung mit einer relationalen *PostgreSQL*-Datenbank und das *AMQP*-Protokoll zur Kommunikation mit einer *ActiveMQ Artemis*-Message Queue zurückgreift. Die Applikation sollte für einen mit *Ubuntu Server 20.04*¹ betriebenen Rechner, der mit dem Befehlssatz der *x86-64*-Architektur kompatibel ist, entwickelt werden.

¹bgl. GetUbuntuServer

Glossar

Akkugüte

Beschreibung	Die Akkugüte beschreibt die maximale Kapazität, mit der ein Akku aufgeladen werden kann, d.h. der Energie die in dem Akku gespeichert werden kann. Da die Spannung eines Akkus über seinen Lebenszyklus nur geringfügig schwankt, entspricht die Akkugüte direkt dem Produkt aus Dauer und Stromfluss, den der Akku leisten kann. Die maximale Kapazität lässt dabei in der Regel mit der Anzahl der durchgeführten Ladezyklen eines Akkus nach.
Synonyme	Akkukapazität
Abgrenzung	Die Akkugüte ist von derjenigen Kapazität zu unterscheiden, die ein Akku theoretisch zu Beginn (nach der Herstellung) aufweist. Insbesondere ist die Akkugüte auch von der Leistung des Akkus abzugrenzen, die bemisst, in welchem Zeitraum die gespeicherte Energie abgegeben werden kann.
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 Erweiterung

Akkustand

Beschreibung	Der Akkustand ist ein Kennwert für den Ladezustand von Akkus. Der Akkustand kennzeichnet die noch verfügbare Kapazität eines Akkus im Verhältnis zum Nominalwert.
Synonyme	-
Abgrenzung	
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt

Akkuverbrauch

Beschreibung	Der Akkuverbrauch beschreibt die Differenz des Akkustands zu Beginn und zum Ende der Fahrt.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt

Analysesystem

Beschreibung	Das Analysesystem beschreibt das noch nicht spezifizierte externe System, welches anhand der ihm von eSS übergebenen Nutzungsdaten eine Analyse durchführt.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Von Langsdorff Amelie
Status	Entwurf
Historie	08.12.2021 erstellt

Buchung

Beschreibung	Eine Buchung beschreibt den Zeitpunkt, an dem ein E-Scooter freigeschaltet wurde.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 06.12.2021

Dauer

Beschreibung	Die Dauer beschreibt die Zeitspanne bzw. den Zeitraum einer Fahrt, d. h. die absolute Differenz zwischen dem Start- und dem Endzeitpunkt einer Fahrt.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 Erweiterung

Distanz

Beschreibung	Die Distanz beschreibt die angenäherte Länge der zurückgelegten Route und somit die Summe der Abstände aller Wegpunkte einer Route.
Synonyme	zurückgelegte Strecke
Abgrenzung	Die Distanz beschreibt nicht den absoluten Abstand (Luftlinie) zwischen Start und Ziel.
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 kleine Änderung

E-Scooter-Verleih

Beschreibung	Ein E-Scooter-Verleih betreibt die Vermietung von E-Scootern und stellt im Sinne der OOA den Auftraggeber dar. Dieser Verleih betreibt ein bestehendes Verleihsystem, welches um eSS erweitert werden soll.
Synonyme	Verleih der E-Scooter
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt

E-Scooter

Beschreibung	Ein E-Scooter ist ein elektrischer Tretroller, der eine Lenkstange besitzt und auf sechs bis höchstens 20 km/h beschleunigt werden kann. Die Leistung des E-Scooters darf 500 Watt nicht überschreiten. Zudem muss ein E-Scooter verkehrssicher sein (z. B. Front- und Rücklicht, funktionierende Bremsen, etc.). Die Verwendung ist nur auf Straßen und Radwegen erlaubt.
Synonyme	Scooter
Abgrenzung	Ein E-Roller hingegen ist ein elektrischer Motorroller mit den selben Rechten und Pflichten eines Motorrollers.
Einschränkungen	keine
Ansprechpartner	Wolti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 kleine Änderungen

externes System

Beschreibung	Als externe Systeme werden all diejenigen Systeme beschrieben, welche nicht Teil von eSS sind, aber mit diesem interagieren.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Wolti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 Änderungsanfrage; 03.12.2021 überarbeitet

Fahrt

Beschreibung	<p>Eine Fahrt ist diejenige Sequenz von Ereignissen zu allen Zeitpunkten zwischen dem Start- und dem Endzeitpunkt einer Fahrt.</p> <p>Eine Fahrt beginnt, sobald ein Nutzer einen E-Scooter gebucht bzw. freigeschalten hat. Der Nutzer kann während einer Fahrt den E-Scooter frei nutzen und damit fahren. Die Fahrt endet, wenn der Nutzer keinen Zugriff mehr auf den E-Scooter hat. Dies kann z.B. durch eine Rückgabe des E-Scooters oder durch den Ablauf der gebuchten Zeit geschehen.</p>
Synonyme	-
Abgrenzung	Die Geschwindigkeit gibt nicht die Richtung an, in die sich ein Objekt bewegt.
Einschränkungen	keine
Ansprechpartner	Wolti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 Erweiterung

Geschwindigkeit

Beschreibung	<p>Die Geschwindigkeit ist ein Maß für die Änderung der Position eines Körpers unter einem betrachteten Zeitraum.</p>
Synonyme	Schnelligkeit
Abgrenzung	<p>Die Geschwindigkeit gibt nicht die Richtung an, in die sich ein Objekt bewegt.</p> <p>Die Geschwindigkeit ist ebenfalls eine zeitabhängige Größe und gibt keinen Aufschluss über die Änderung der Position zu einem beliebigen Zeitpunkt.</p>
Einschränkungen	keine
Ansprechpartner	Wolti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 Überarbeitung

Identifikator

Beschreibung	Ein Identifikator beschreibt ein (künstliches) Merkmal, welches eindeutig mit einer bestimmten Identität verknüpft werden kann. Damit ist eine eindeutige Identifizierung des Objekts oder einer Person möglich. Mathematisch entspricht diese Definition der Identifikatoren einer bijektiven Funktion über der Menge der zu identifizierenden Komponenten \mathbb{K} und der Menge der verwendeten Identifikatoren \mathbb{I} .
---------------------	--

$$k, k' \in \mathbb{K}, f(k) = f(k') \rightarrow k = k'$$

$$\forall i \in \mathbb{I} \exists k \in \mathbb{K} f(k) = i$$

Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 eindeutige Beschreibung hinzugefügt

Nutzer

Beschreibung	Ein Nutzer beschreibt eine Person, welches ein Produkt nutzt und mit diesem interagiert.
Synonyme	Benutzer, Anwender, Bediener
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	02.12.2021 erstellt

Nutzerprofil

Beschreibung	Unter einem Nutzerprofil wird eine Zusammenfassung von benutzerdefinierten Einstellungen und Angaben eines Nutzers beschrieben.
Synonyme	Profil
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt

Route

Beschreibung	Eine Route ist beschrieben durch eine Sequenz von absoluten Wegpunkten $(s_i)_{i \in [k]}$, $k \in \mathbb{N}$, die periodisch in Zeit erfasst werden. Die Wegpunkte beschreiben somit näherungsweise den zurückgelegten Weg zwischen Start und Ziel einer Fahrt.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 Überarbeitet; 04.12.2021 Δt als relevantes Kriterium hinzugefügt

Standort

Beschreibung	Der Standort ist die absolute Position auf der Erdoberfläche unter Angabe von mindestens 3 Koordinaten.
Synonyme	Standpunkt
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Vadim Muhametcin
Status	Entwurf
Historie	08.12.2021 erstellt; 09.12.2021 Änderung; 10.12.2021 Überarbeitung

Start

Beschreibung	Der Start beschreibt denjenigen absoluten Standort, zu dem sich der Scooter zum Startzeitpunkt befindet, d. h. an dem eine Fahrt beginnt.
Synonyme	Startstandort
Abgrenzung	Der Start beschreibt nicht den Zeitpunkt, zu dem eine Fahrt beginnt.
Einschränkungen	keine
Ansprechpartner	Wolti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021, Erweiterung

Verleihsystem

Beschreibung	Das Verleihsystem ist das IT-System des E-Scooter-Verleihs, mit welchem mindestens die Buchungen von E-Scootern, die E-Scooter selbst sowie die Nutzerprofile verwaltet werden.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Von Langsdorff Amelie
Status	Entwurf
Historie	08.12.2021 erstellt

Wegpunkt

Beschreibung	Ein Wegpunkt beschreibt einen eindeutigen, absoluten Standort eines E-Scooters zu einem bestimmten Zeitpunkt während einer Fahrt.
Synonyme	-
Abgrenzung	Ein Wegpunkt ist zu unterscheiden von dem Standort eines E-Scooters während sich der E-Scooter in keiner Fahrt befindet.
Einschränkungen	keine
Ansprechpartner	Wolti Andreas
Status	Entwurf
Historie	05.12.2021 erstellt

Zeitpunkt

Beschreibung	Ein Zeitpunkt im Allgemeinen ist zunächst der kleinste Abschnitt, unter dem ein zeitlicher Ablauf betrachtet wird. In der Praxis wird die Größe dieses Abschnitts in der Regel durch die Fähigkeit der verwendeten Messtechnik beschränkt. Je nach betrachteter Sequenz, können einzelne Zeitpunkte durch das mit Ihnen verknüpfte Ereignis oder ihre Position in der Sequenz als hervorgehoben betrachtet werden. Bei der Betrachtung einer Fahrt z. B. beschreibt der Startzeitpunkt den ersten Zeitabschnitt dieser Fahrt, der Endzeitpunkt hingegen den letzten Zeitabschnitt der Fahrt.
Synonyme	-
Abgrenzung	-
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 komplette Überarbeitung der Definition; 03.12.2021 kleine Änderung

Ziel

Beschreibung	Das Ziel beschreibt denjenigen absoluten Standort, zu dem sich der Scooter zum Endzeitpunkt befindet, d. h. an dem eine Fahrt endet.
Synonyme	Zielstandort
Abgrenzung	Das Ziel beschreibt nicht den Zeitpunkt, zu dem eine Fahrt beendet wird.
Einschränkungen	keine
Ansprechpartner	Welti Andreas
Status	Entwurf
Historie	01.12.2021 erstellt; 03.12.2021 Erweiterung

Literatur

- [ActiveMQVsRabbitMQa] : *ActiveMQ vs RabbitMQ*.
<https://www.openlogic.com/blog/activemq-vs-rabbitmq>
- [ActiveMQVsRabbitMQ] : *ActiveMQ vs RabbitMQ / What Are the Differences?*
<https://stackshare.io/stackups/activemq-vs-rabbitmq>
- [GetUbuntuServer] : *Get Ubuntu Server / Download*. <https://ubuntu.com/download/server>
- [GitHub 2021] GITHUB: *Github Language Stats*. 2021. – URL https://madnight.github.io/github/#/pull_requests/2021/3. – Zugriffsdatum: 2021-12-09
- [GlobalStats 2021] GLOBALSTATS: *Operating system market share worldwide*. 2021. – URL <https://gs.statcounter.com/os-market-share>
- [Kerr 2010] KERR, Justin: *AMD Loses Market Share as Mobile CPU Sales Outsell Desktop for the First Time*. 2010
- [Lithmee 2019] LITHMEE: *What Is the Difference Between ActiveMQ and RabbitMQ*. <https://pediaa.com/what-is-the-difference-between-activemq-and-rabbitmq/>. Juni 2019
- [Oracle 2021] ORACLE: *Oracle Java SE Support Roadmap*. 2021. – URL <https://www.oracle.com/java/technologies/java-se-support-roadmap.html>. – Zugriffsdatum: 2021-12-09
- [Python 2021] PYTHON: *Download Python*. 2021. – URL <https://www.python.org/downloads/>. – Zugriffsdatum: 2021-12-09
- [V8 2021] V8: *V8 JavaScript engine*. 2021. – URL <https://v8.dev/>. – Zugriffsdatum: 2021-12-09

A Anhang - Durchführung der Technologieentscheidung

Die Auswirkung der Leistungsfähigkeit von Hardware auf die Qualität und Leistung der Anwendung ist geringfügig, weswegen auf die weite Verbreitung von x86-64-, IA-32- und ARM64-Prozessoren verwiesen wird.² Sie sollen eine mögliche Auswahl für die darauf aufbauenden Schichten bilden.

Schließt man mobile Betriebssysteme aus, sind die meistverbreiteten Betriebssysteme Windows, macOS und Linux in verschiedenen Distributionen. Das Betriebssystem sollte für die Auswahl von Hardware verfügbar sein. Für x86-64-Architekturen sind die anwendbaren Betriebssysteme Windows, macOS und Linux. Auf ARM64-Architekturen läuft auch das Betriebssystem Windows und es existieren entsprechende Linuxdistributionen. Für IA-32-Architekturen wird keine Version von macOS angeboten. Darüber hinaus ist macOS ein proprietäres Betriebssystem.³

Deswegen werden als Möglichkeiten zur Implementierung der Anwendung nur noch aktuelle Windowsversionen und Linuxdistributionen betrachtet.

Auf Programmiersprachen lassen sich die meisten der Kriterien anwenden. Sie wurden exemplarisch auf drei Programmiersprachen angewandt. Bereits ausgeschlossene Technologien und nicht anwendbare Kriterien werden in der folgenden Tabelle nicht erwähnt.

	Java	Python	JavaScript
Unabhängigkeit	proprietär	PSF Lizenz (GPL-kompatibel)	BSD
End of Life	September 2026 (Java 17) ⁴	Oktober 2024 (Python 3.8) ⁵	jährliche Updates von ECMAScript
Verbreitung (GitHub Q3/2021) ⁶	12,977%	15,943%	17,955%
Betriebssysteme	Windows, macOS, Linux	Windows, macOS, Linux, FreeBSD, Android	Windows, macOS, Linux (V8) ⁷

Tab. A.2: Vergleich von Programmiersprachen hinsichtlich Ausschlusskriterien

Alle drei Sprachen sind mit den ausgewählten Betriebssystemen kompatibel und insofern

²vgl. Kerr (2010)

³vgl. GlobalStats (2021)

⁴vgl. Oracle (2021)

⁵vgl. Python (2021)

⁶vgl. GitHub (2021)

⁷vgl. V8 (2021)

von diesen unabhängig.

Mit einer Nutzung von 12,977% auf GitHub und einem garantierten Support bis September 2026 ist Java zwar als besonders ausgereift zu betrachten, allerdings ist die Sprache stark abhängig von der Oracle Corporation, die die Sprache entwickelt.

Auch Python ist sehr weit verbreitet und die Version 3.8 wird bis Oktober 2024 durch den Lizenzinhaber unterstützt. Die Sprache wird den Entwicklungen des Markts entsprechend oft neu angepasst und ist von den Entscheidungen Einzelner unabhängig (PSF).

JavaScript ist die aus den drei am weitesten verbreitete Programmiersprache und erfährt jährliche Updates durch die Ecma International. Die Sprache wird über die Ecma International durch verschiedene Interessengruppen beeinflusst, aber ist freizügig lizenziert (BSD). Besonders geeignet sind also Python und JavaScript.

Um Daten langfristig persistent zu speichern wird heutzutage primär von Datenbanken Gebrauch gemacht. Das gilt unabhängig davon, ob es sich um Scooter- oder Analysedaten handelt. Zugleich werden Daten, die in einem Internet-of-Things-Kontext (wie bei Scootern gegeben) übermittelt werden, häufig als Stream bereitgestellt und auch konsumiert. Es wäre also ideal das System so zu gestalten, dass diese beiden Möglichkeiten Daten zu konsumieren und zu persistieren auch für das künftige Analysesystem beibehalten werden.

Weil es sich um ein Datenbankmodell mit hohem Reifegrad handelt, wird eine relationale Datenbank für die Technologieauswahl vorgeschlagen. ODBC ist eine Open-Source-Schnittstelle für relationale Datenbanken und kann damit verwendet werden.

Um Daten als Stream zu empfangen und bereitzustellen, sollen deswegen die beiden Protokolle Message Queuing Telemetry Transport (MQTT) und Advanced Message Queuing Protocol (AMQP) verglichen werden, die grundsätzlich beide für diesen Zweck erdacht wurden.

Als Protokolle, die seit mehr als einem Jahrzehnt weit verbreiteten Einsatz erfahren und von einem pluralen Konsortium als Open-Source-Spezifikation entwickelt werden, erfüllen MQTT und AMQP beide einige der relevanten Kriterien. Allerdings ist AMQP mit Verschlüsselung vereinbar und bietet zudem die Möglichkeit die Funktionalitäten zu erweitern. Dies ist hinsichtlich rechtlicher Kriterien und Flexibilität ein großer Vorzug. Ersteres entfällt demnach für die weitere Betrachtung.

Die Wahl des Protokolls schränkt die in Frage kommenden Bibliotheken und Software stark ein. Exemplarisch werden weiterhin die beiden Open-Source-Projekte RabbitMQ und Apache ActiveMQ verglichen.

Beide Systeme unterstützen AMQP out-of-the-box und erfüllen die Kriterien hinsichtlich Verbreitung und Reifegrad. Allerdings unterstützt ActiveMQ SSL, und damit die im Da-

tenschutz relevante Verschlüsselung, RabbitMQ hingegen zunächst nicht. Des weiteren ist ActiveMQ in Java, RabbitMQ in Erlang geschrieben. Die bibliothek eigenen Abhängigkeiten sprechen also für die Wahl von ActiveMQ.⁸

⁸vgl. Lithmee (2019), ActiveMQVsRabbitMQ, ActiveMQVsRabbitMQa