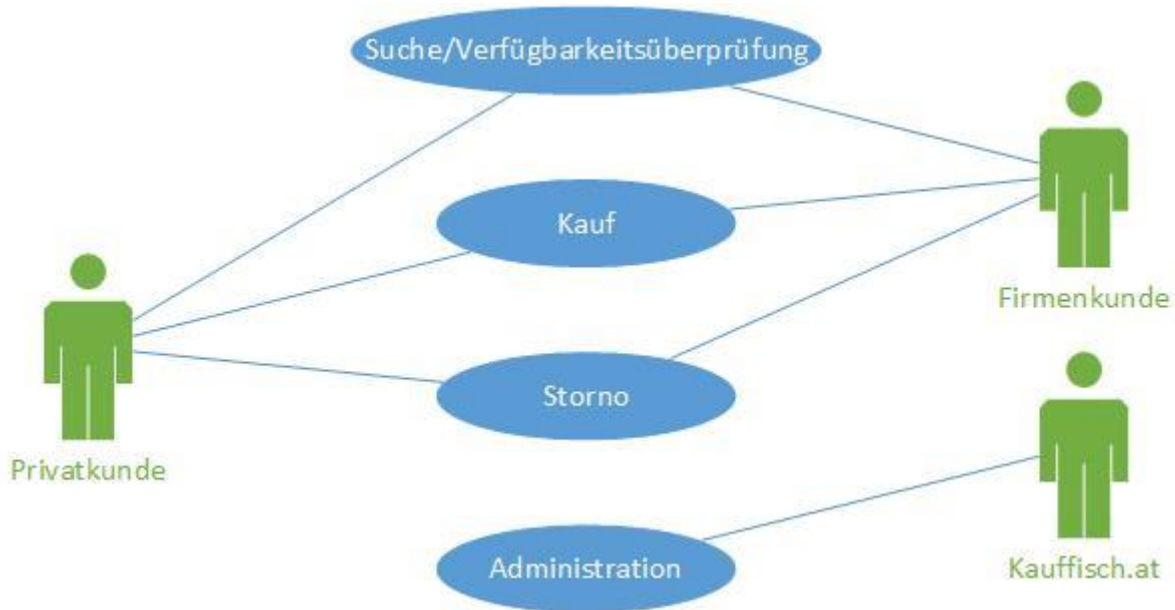


## Aufgabe 2: Konzeptuelle Modellierung des B2B/C2B Systems „Kauffisch.at“ der Webshop für Fisch und Meeresfrüchte

### A: Use Case Beschreibung (6 Punkte)



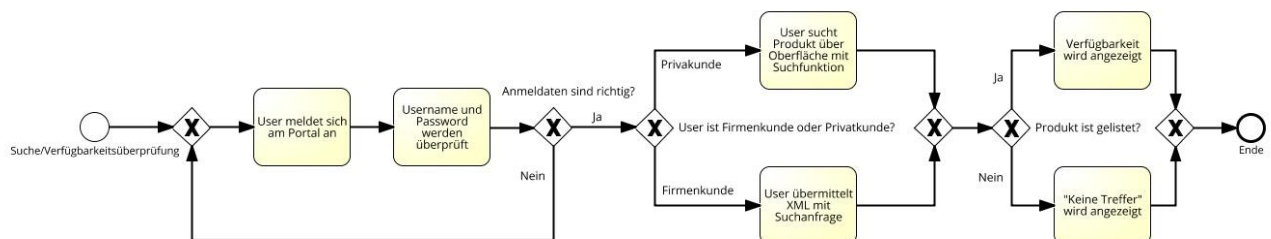
In der Abbildung kann man das Use Case Diagramm für den Webshop sehen, die 3 Akteure, die mit dem System interagieren sind der Webshop bzw. der Administrator, die Privatkunden und die Firmenkunden. Die 4 Use Cases sind Suche bzw. Verfügbarkeitsüberprüfung, Kauf, Storno und Administration.

#### A.1: Use Case 1: Suche/Verfügbarkeitsüberprüfung

<b>Use Case:</b>	Suche/Verfügbarkeitsüberprüfung
<b>Use Case ID:</b>	1
<b>Actor(s):</b>	Privatkunde Firmenkunde
<b>Brief description:</b>	Privatkunde (B2C): Nach der Anmeldung im Privatkundenbereich mit Username und Passwort kann der User sein gewünschtes Produkt mit der Suchfunktion suchen oder selbst die ganze Produktliste durchlisten. Nachdem er sein Produkt gefunden hat, kann er eine Verfügbarkeitsüberprüfung durchführen. Je nach Resultat kann er sein

	<p>Produkt in den Warenkorb geben und schlussendlich den Kauf durchführen.</p> <p>Firmenkunde (B2C): Nach der Anmeldung im Firmenbereich mit Username und Passwort kann der Firmenuser ebenfalls sein gewünschtes Produkt suchen. Jedoch unterscheidet sich hier die Firmensuche, da der User eine XML mit dem gewünschten Produkt sendet. Als Response bekommt er zurück, ob dieses Produkt gelistet ist und ob es verfügbar ist. Diese Aktion wird automatisch durchgeführt.</p>
<b>Pre-conditions:</b>	Die User müssen angemeldet sein.
<b>Post-conditions:</b>	Falls ihre Suche Treffer ergaben bzw. das Produkt verfügbar ist, kann ein Kauf vollzogen werden.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>Der User meldet sich im Portal mit Username und Passwort an <ol style="list-style-type: none"> <li>B2C: Privatkundenportal</li> <li>B2B: Firmenkundenportal</li> </ol> </li> <li>Der User sucht nach seinem Produkt <ol style="list-style-type: none"> <li>B2C: Suche durch Suchfunktion</li> <li>B2B: Übermittlung von XML</li> </ol> </li> <li>Verfügbarkeitsüberprüfung wird durchgeführt <ol style="list-style-type: none"> <li>B2C: Verfügbarkeit enthalten in der Produktbeschreibung</li> <li>B2B: Verfügbarkeit enthalten in der Response der Suchanfrage</li> </ol> </li> <li>User kann Kauf tätigen</li> </ol>
<b>Extensions:</b>	Man kann den Prozess jederzeit abbrechen
<b>Priority:</b>	Hoch
<b>Performance Target:</b>	User soll Produkte finden und auf Verfügbarkeit überprüfen
<b>Issues:</b>	User findet Produkte nicht oder Produkte sind nicht verfügbar

### „Suche/Verfügbarkeitsüberprüfung“ Prozessdiagramm (BPMN 2.0)



## A.2: Use Case 2: Kauf

<b>Use Case:</b>	Kauf
<b>Use Case ID:</b>	2
<b>Actor(s):</b>	Privatkunde Firmenkunde
<b>Brief description:</b>	<p>Privatkunde (B2C): Nach dem der Privatkunde seine Produkte ausgewählt hat, muss er in seinen Warenkorb gehen und zur Auswahl der Zahlungsart voranschreiten. Nach der Bestätigung der Bestellung ist der Kauf abgeschlossen.</p> <p>Firmenkunde (B2C): Nach dem der Firmenkunde den Status seiner Produkte abgefragt hat, muss er weitere XMLs übermitteln. Jene XMLs enthalten die Bestellungen der Produkte. Eine erfolgreiche Response auf diese Requests ist die Bestätigungen für den Kauf der Güter. Diese Aktion wird ebenfalls automatisch durchgeführt.</p>
<b>Pre-conditions:</b>	Die User wählen Produkte aus.
<b>Post-conditions:</b>	Bestellungen werden durchgeführt und weiterverarbeitet.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. Der User hat Produkte ausgewählt<ol style="list-style-type: none"><li>1.1. B2C: Produkte sind im Warenkorb</li><li>1.2. B2B: Produkte sind in den Bestellungen-XMLs</li></ol></li><li>2. Der User bestätigt Kauf<ol style="list-style-type: none"><li>2.1. B2C: Zahlungsart und Daten wurden bestätigt</li><li>2.2. B2B: Übermittlung von XML</li></ol></li><li>3. Bestellung wurde bestätigt<ol style="list-style-type: none"><li>3.1. B2C: Bestätigung via Weboberfläche</li><li>3.2. B2B: Bestätigung via Webservice-Response</li></ol></li><li>4. Kauf wurde erfolgreich abgeschlossen</li></ol>
<b>Extensions:</b>	Man kann den Prozess jederzeit abbrechen
<b>Priority:</b>	Hoch
<b>Performance Target:</b>	User soll Kauf tätigen
<b>Issues:</b>	User hat Probleme bei der Bezahlung oder Bestellung

### „Kauf“ Prozessdiagramm (BPMN 2.0)

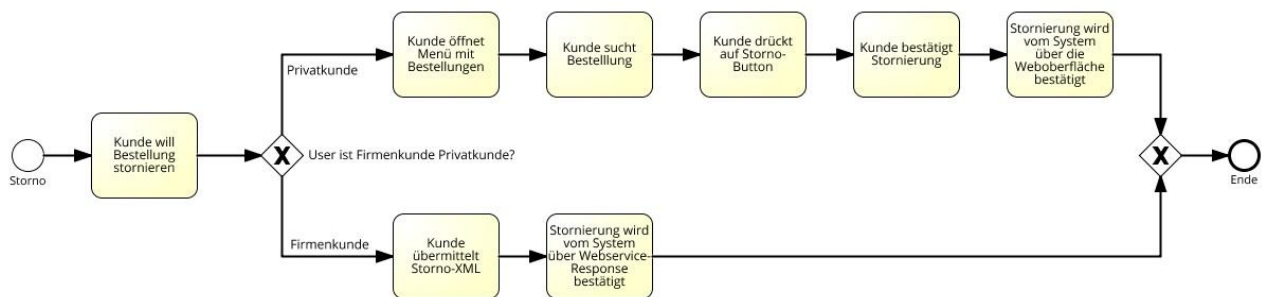


### A.3: Use Case 3: Storno

<b>Use Case:</b>	Storno
<b>Use Case ID:</b>	3
<b>Actor(s):</b>	Privatkunde Firmenkunde
<b>Brief description:</b>	<p>Privatkunde (B2C): Nach dem der Privatkunde Produkte bestellt hat, kann er gegebenenfalls seine Bestellung stornieren. Er öffnet den Menüpunkt Bestellungen und bekommt dort eine Liste seiner Bestellungen. Mit einem Klick auf einen Button wird die bestimmte Bestellung storniert.</p> <p>Firmenkunde (B2C): Nach dem der Firmenkunde seine Produkte bestellt hat, kann er bestimmte Bestellungen via XML stornieren. Hier ist es wichtig die richtige Bestellungs-ID zu übermitteln. Diese Aktion wird ebenfalls automatisch durchgeführt.</p>
<b>Pre-conditions:</b>	Der User hat bereits Produkte bestellt
<b>Post-conditions:</b>	Der User muss neue Bestellungen durchführen
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>Der User hat Bestellung ausgewählt <ol style="list-style-type: none"> <li>B2C: Bestellung in Menüpunkt suchen</li> <li>B2B: Bestellung wird durch die Bestellungs-ID dargestellt und in Storno-XML eingetragen.</li> </ol> </li> <li>Storno durchführen <ol style="list-style-type: none"> <li>B2C: Storno-Button drücken</li> <li>B2B: Übermittlung von Storno-XML</li> </ol> </li> <li>Storno wurde bestätigt <ol style="list-style-type: none"> <li>B2C: Bestätigung via Weboberfläche</li> <li>B2B: Bestätigung via Webservice-Response</li> </ol> </li> </ol>

	4. Storno wurde abgeschlossen
<b>Extensions:</b>	Man kann den Prozess jederzeit abbrechen
<b>Priority:</b>	Mittel
<b>Performance Target:</b>	User soll Bestellung stornieren
<b>Issues:</b>	Probleme beim Stornieren der Bestellung

#### „Storno“ Prozessdiagramm (BPMN 2.0)

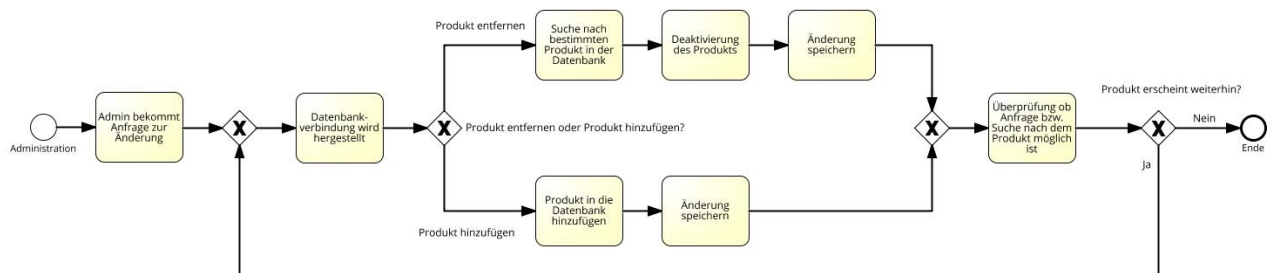


#### A.4: Use Case 4: Administration

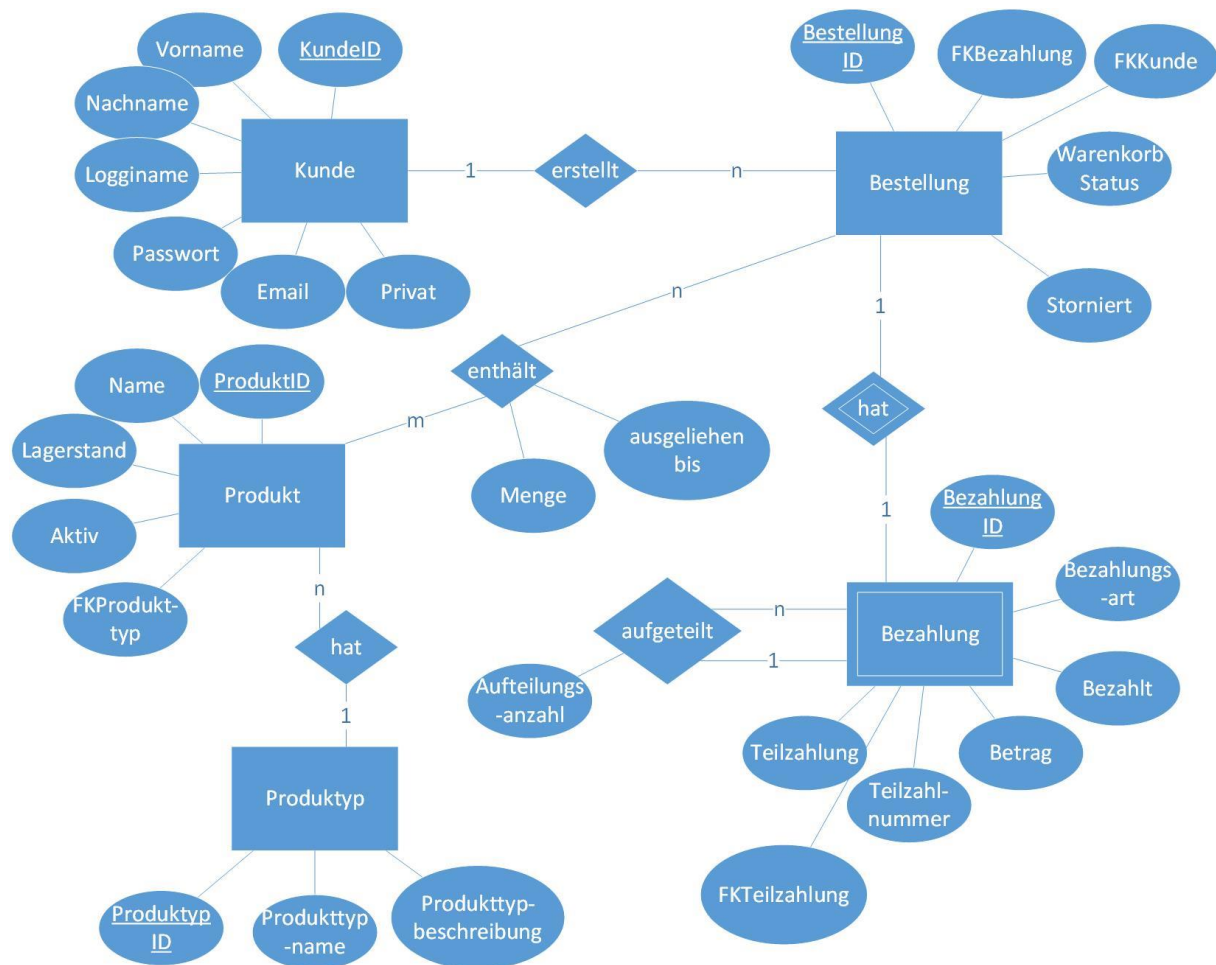
<b>Use Case:</b>	Administration
<b>Use Case ID:</b>	4
<b>Actor(s):</b>	Administrator
<b>Brief description:</b>	Der Administrator ist dazu da bestimmte Administrortätigkeiten wie z.B.: Wartung des Webshops, Wartung des Services, Entfernen oder Hinzufügen von Produkte, Wartung der Datenbank etc. durchzuführen. Alle diese Tätigkeiten sind in dieser einen Aktivität „Administration“ zusammengefasst.
<b>Pre-conditions:</b>	Daten müssen umgestellt oder angelegt werden. Z.B.: Produkt wird nicht mehr gelistet und muss aus dem Angebot geholt werden oder neues Produkt wird angeboten und muss in die Datenbank eingetragen werden

<b>Post-conditions:</b>	Umstellung oder Wartung wird upgedated
<b>Main Success Scenario:</b>	<p>Löschen eines Produkts:</p> <ol style="list-style-type: none"> <li>1. Administrator erstellt Verbindung zu Datenbank</li> <li>2. Sucht bestimmtes Produkt in der Datenbank</li> <li>3. Deaktiviert das Produkt damit es auf der Seite bzw. vom Service nicht mehr angeboten wird</li> <li>4. Anfrage bzw. Suche nach dem Produkt ist nicht mehr möglich</li> </ol> <p>Hinzufügen eines Produkts:</p> <ol style="list-style-type: none"> <li>1. Administrator erstellt Verbindung zu Datenbank</li> <li>2. Hinzufügen des Produkts in die Datenbank</li> <li>3. Anfrage bzw. Suche nach dem Produkt ist nun möglich</li> </ol>
<b>Extensions:</b>	Ablauf der Prozesse kann erweitert oder verändert werden
<b>Priority:</b>	Hoch
<b>Performance Target:</b>	System pflegen
<b>Issues:</b>	Probleme mit dem System oder bei der Dateneingabe

### „Administration“ Prozessdiagramm (BPMN 2.0)



## B: Datenmodell (3 Punkte)



## C: Definition der EDI Sprache für B2B Use Case Prozesse (5 Punkte)

Wie oben bereits beschrieben gibt es 3 B2B Use Cases:

1. GET Methode: „Suche/Verfügbarkeitsüberprüfung“
2. POST Methode: „Kauf“
3. PUT Methode: „Storno“

Für alle Cases wurde ein passendes XML-Schema entworfen damit die XML-Requests validiert werden können. Durch die Schnittstelle können Firmenkunde ihre Requests dem System übermitteln, wie bereits erwähnt werden die UseCases danach automatisch abgeschlossen.

### C.1 GET Methode: „Suche/Verfügbarkeitsüberprüfung“

XML-Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<suche kundeid="709">
  <produktid>342</produktid>
</suche>
```

XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="suche">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="produktid" type="xs:int"/>
      </xs:sequence>
      <xs:attribute name="kundeid" type="xs:int"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## C.2 POST Methode: „Kauf“

XML-Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<kauf kundeid="709">
  <produktid>342</produktid>
  <menge>100</menge>
  <!-- optional: Fall wenn Produkt ausgeliehen wird -->
  <ausleihtage>60</ausleihtage>
</kauf>
```

XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="kauf">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="produktid" type="xs:int"/>
        <xs:element name="menge" type="xs:int"/>
        <xs:element name="ausleihtage" type="xs:int" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="kundeid" type="xs:int"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## C.3 PUT Methode: „Storno“

XML-Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<storno kundeid="709">
  <bestellungid>342</bestellungid>
</storno>
```

XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="storno">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="bestellungid" type="xs:int"/>
      </xs:sequence>
      <xs:attribute name="kundeid" type="xs:int"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

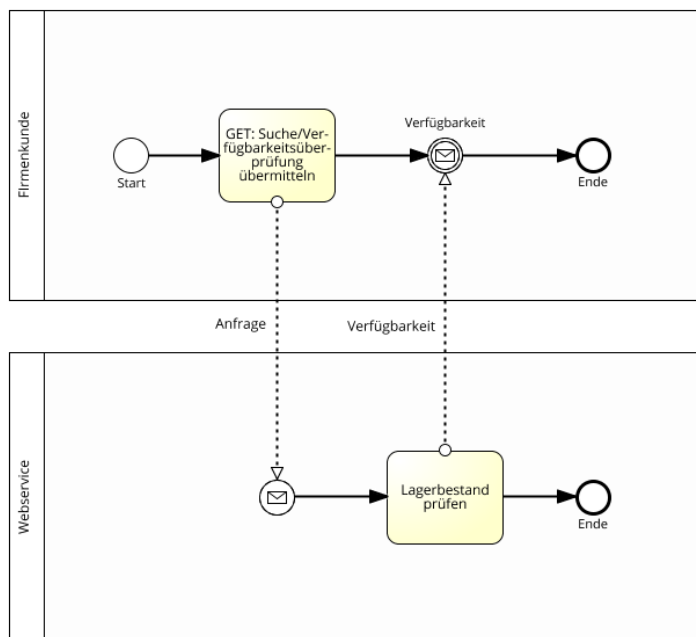


## D: B2B Schnittstellenbeschreibung (6 Punkte)

Die XML-Requests sollen über eine Rest-Schnittstelle verschickt werden, welche das Lesen, Speichern und Modifizieren von Daten leicht macht. Die Firmenkunden können auf der Webseite ihre XML-Requests übermitteln, das System liest diese ein und ordnet zu um welche Art von Request es sich handelt. Danach werden die Daten verarbeitet und die Datenbank geschrieben. Obwohl es sich um 3 verschiedene Request Methoden handelt, sind die Prozesse nicht sehr verschieden.

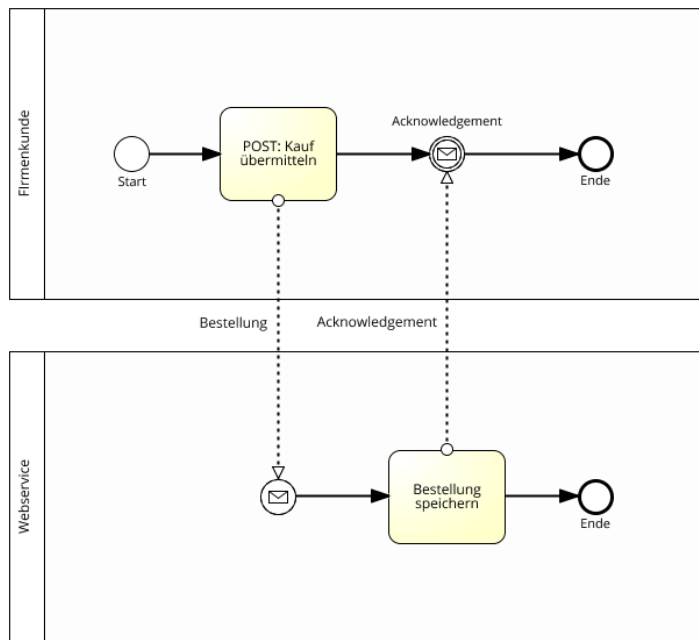
### D.1 GET Methode: „Suche/Verfügbarkeitsüberprüfung“ – Diagramm

Der Firmenkunde versendet die Verfügbarkeitsanfrage über XML mit einem GET, danach verarbeitet das System die Anfrage und antwortet dem Firmenkunden mit der Verfügbarkeit für das bestimmte Produkt. Die Antwort enthält Lagerbestand und ob das Produkt noch gelistet ist und bei erfolgloser Suche, dass es dieses Produkt nicht gibt.



### D.2 POST Methode: „Kauf“ – Diagramm

Der Firmenkunde versendet die Bestellung über XML mit einem POST, danach verarbeitet das System die Anfrage und speichert die neue Bestellung in die Datenbank. Schlussendlich antwortet das System dem Firmenkunden mit einer positiven oder negativen Nachricht.



### D.3 PUT Methode: „Storno“ – Diagramm

Der Firmenkunde versendet das Storno einer Bestellung über XML mit einem Put, danach verarbeitet das System die Anfrage und modifiziert den alten Datensatz in der Datenbank. Schlussendlich antwortet das System dem Firmenkunden mit einer positiven oder negativen Nachricht.

