



AvaL

Technisches Konzept

Schnittstelle AvaL

Inhaltsverzeichnis

1 Ausgangssituation	1
2 Austausch und Anlage der AvaL-Vereinbarungen	2
2.1 Austausch von AvaL-Vereinbarungen	2
2.1.1 Abbrechen eines Matchings	4
2.1.2 Ändern eines Matchings	5
2.1.3 Generierung Aval-Id	5
2.2 AvalState - Mögliche Werte	5
2.3 Role - Mögliche Werte	6
3. Vorgang zu einem Matching durchführen	6
3.1 Beauftragung aufgeben (AG->AN)	10
3.2 Technisch akzeptiert (optional)(AN->AG)	10
3.3 Technisch abgelehnt (optional)(AN->AG)	11
3.4 Fachlich akzeptiert (optional)(AN->AG)	11
3.5 Fachlich abgelehnt (optional)(AN->AG)	12
3.6 Stornierung (optional)(AN->AG; AG->AN)	12
3.7 Avisierung (AN->AG)	12
3.8 Leistung erbracht (optional) (AN->AG)	13
3.9 Rückmeldung (AN->AG)	13
3.10 Rückmeldung mit Abweichungen (AN->AG)	14
3.11 Dokument (AN->AG, AG->AN)	14
4.1 Anmerkungen zum Möglichen Datenmodell	15
4.2 Fehlermeldungen	15
7 Dokumentenhistorie	16

1 Ausgangssituation

Entsorger deutschlandweit tauschen regelmäßig Informationen zu Entsorgungen ihrer Kunden miteinander aus, um den wachsenden Bedarf dieser bedienen zu können. Dieser Austausch erfolgt bisher per Telefon, Mail oder persönlich. Es kommt durchaus vor, dass auch der Endkunde sich für eine spontane Entsorgung nicht an seinen Vertragspartner sondern direkt an den ihm bekannten Entsorger wendet. Im Anschluss muss dieser mit dem entsprechenden Vertragspartner die Informationen für den Entsorgungsvorgang austauschen.

Damit dieser Datenaustausch einfacher und schneller vonstatten geht hat der Bund Deutscher Entsorger (BDE) eine Arbeitsgruppe beauftragt ein Konzept für eine digitale Schnittstelle zu entwerfen und testen. Die

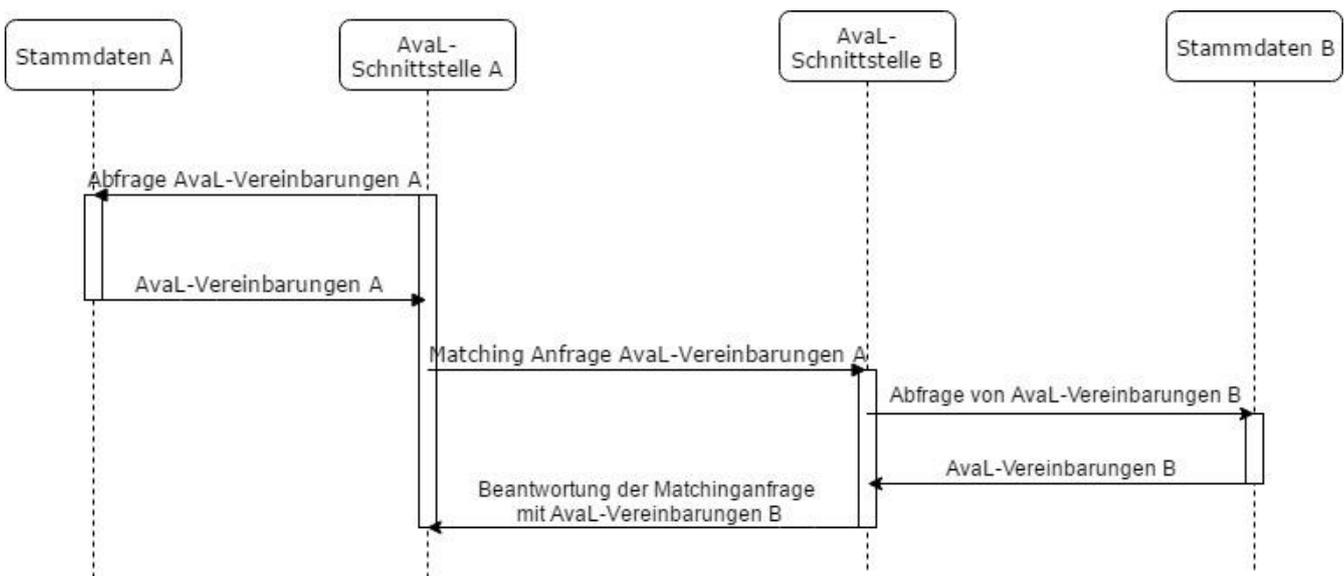
Arbeitsgruppe bestehend aus den Firmen Meinhardt, Veolia, Logex, Interseroh, Zentek, Gipa hat in einem Treffen am 7.11.2018 und 8.11.2018 einen ersten Entwurf aus fachlicher Sicht für einen Bewegungsdatenaustausch entworfen. Auf Basis dieser Ergebnisse ist der vorliegende technische Entwurf für eine Schnittstelle entstanden.

Zusätzlich zum Konzept für den Bewegungsdatenaustausch wird in diesem Dokument ein Austausch von Rahmeninformationen für die Durchführung von Entsorgungen beschrieben. Ergebnis dieses Austauschs soll die sogenannte AVA-L-ID (AvaL Austausch von auftragsbezogenen Leistungsdaten) sein, die zu Identifizierung von Leistungsorten, Stoffen und weiteren Informationen dienen soll.

2 Austausch und Anlage der AvaL-Vereinbarungen

Für die Entsorgung von Stoffen beim Kunden braucht ein Entsorger eine Reihe von Informationen. Zum Beispiel müssen Behältertyp, Stoff, Dienstleistungsart, Leistungsadresse uvm. abgestimmt sein. Dies sind unveränderliche Informationen die bei jeder Durchführung identisch sein müssen, die sogenannten AvaL-Vereinbarungen. Neben diesen gibt es von Durchführung zu Durchführung veränderliche Informationen wie Menge und Ausführungstag.

Wird die Entsorgung von einem anderen Entsorger als dem direkten Vertragspartner des Kunden durchgeführt ist eine Weitergabe der fixen Informationen notwendig. Sobald diese feststehen kann der Entsorger beliebig oft auf Anfrage oder selbstständig (bei fixem Rhythmus) die Entsorgung durchführen. Im Folgenden wird ein Prozess für den Austausch der fixen Informationen entworfen.



2.1 Austausch von AvaL-Vereinbarungen

Die AvaL-Vereinbarungen bestehen aus einer Reihe von Fachlichen Informationen und sollten im jeweiligen Entsorger-System zusätzlich "technisch" eindeutig definierbar sein. Hierfür soll die sogenannte AvaL-ID dienen. Je nach Aufbau des Entsorger-Systems müssen dazu mehrere Daten gebündelt werden oder es kann direkt ein eindeutiger Schlüssel verwendet werden.

Ein Aval-Matching soll wie folgt aufgebaut werden (erste Nachricht von System A an System B)

```
HTTP-Post: http://www.aval-interface-B.org/avalmatching/

{
  "avalMatching": {
    "avalState": 1,
    "avalId": "veolia4711A0815B",

    "client": {
      "contractPartner": "MaxMustermann GmbH",
      "containerType": "",
      "material": "Papier",
      "serviceType": "Austausch",
      "performancePlace": "Musterstraße 43, 22041 Hamburg",
      "materialUnit": "",
      "serviceUnit": "S",
      "cycleInformation": false,
      "cycleRythm": null
    },
    "supplier": {}
  }
}
```

Das angegebene Schema und die Beispieldaten beschreiben einen Aval-Status, eine Aval-ID, Role und eine Menge an Info-Feldern. Die Role dient der Unterscheidung der zwei Partner für ein Matching. Es gibt einen Client und einen Supplier (Auftragnehmer und Auftraggeber). Der Aval-Status dient der Erkennung des Zusatands des Matchings. Mögliche Zuständer werden später aufgelistet. Die AvalID ist die eindeutige Referenz im System desjenigen der sie geschickt hat und dient in Zukunft als Erkennungsmerkmal des Matchings. Die Zusammensetzung und Entstehung der AVAL-ID wird in einem späteren Kapitel beschrieben. Für die Eröffnung eines Matchings sollten so viele Felder wie möglich mitgeliefert werden, um das Matching zu elreichern. Da jedoch jeder Entsorger seine Systeme anders aufbaut, kann nicht gewährleistet werden, dass immer alle Felder geschickt werden.

Ein Mensch muss im Anschluss an diese Anfrage anhand der vorliegenden Informationen die passenden Informationen in seinem ERP-System finden. Er weiß durch die Client/Supplier Informationen um welchen Vertrag es geht und in welchem Verhältnis (Auftraggeber/Auftragnehmer) er zu dem Partner steht. Hat er die passenden Informationen gefunden kann über die Schnittstelle folgende Antwort geschickt werden:

Es wird im folgenden davon Ausgegangen, dass zuvor die Client-Nachricht mit AvalID "veolia4711A0815B" geschickt wurde:

```
HTTP-PATCH: http://www.aval-interface-A.org/avalmatching/veolia4711A0815B

{
  "avalMatching": {
    "avalState": 2,
    "avalId": "veolia4711A0815B",
    "client": {
```

```
    "contractPartner": "M. Mustermann GmbH",
    "containerType": "",
    "material": "Papier",
    "serviceType": "Austausch",
    "performancePlace": "Musterstraße 43, 22041 Hamburg",
    "materialUnit": "",
    "serviceUnit": "S",
    "cycleInformation": false,
    "cycleRythm": null
  },
  "supplier": {
    "contractPartner": "Max Musterman GmbH",
    "containerType": "UL",
    "material": "PPK",
    "serviceType": "Tausch",
    "performancePlace": "Muster Strasse 43 Hamburg",
    "materialUnit": "",
    "serviceUnit": "S",
    "cycleInformation": false,
    "cycleRythm": "2x 14-tägig"
  }
}
```

Die Antwort des Matchingpartners besteht nun aus einem Update (http-put) auf die bekannte Avalld. Im Objekt wird der avalState als "matched" mitgeliefert, um das Matching zu bestätigen. Diesmal wird als Role Supplier geliefert, um die fachlichen Informationen des Partners anzubieten. Hierbei sind in obere Beispiel auch Informationsunterschiede deutlich gemacht. Der Client (Auftraggeber) speichert den Kunden als "Max Mustermann", der Supplier (Auftragnehmer) als "Max Mustermann". Aus fachlicher Sicht sind dies wahrscheinlich dieselben Kunden. Für das Matchingsystem sollten diese Unterschiede egal sein. Es verwendet die Aval-Id als Verknüpfung.

Wie bereits erwähnt, könnte die Antwort auch Client als Role enthalten, wenn Client und Supplier tauschen.

2.1.1 Abbrechen eines Matchings

Wenn ein Partner ein Matching beenden möchte, kann er einfach ein Put mit dem Cancelled Status an den anderen schicken. Wenn die Partner bereits ihre Avallds getauscht haben, ist ein Update des Status sowohl auf die initialAvalld als auch die matchedAvalld möglich.

Hier zwei Beispiele:

```
HTTP-PATCH:
http://www.aval-interface-A.org/avalmatching/veolia4711A0815B

{
  "avalMatching": {
    "avalState": 3,
    "avalId": "veolia4711A0815B"
  }
}
```

2.1.2 Ändern eines Matchings

Eine Änderung soll der Vereinfachung nach nicht möglich sein. Wenn sich Details ändern, muss das Matching abgebrochen und im Anschluss neu abgestimmt werden. Eine Software-Implementierung kann natürlich dem Anwender ein Anpassung vortauschen, wenn im Untergrund zuerst ein Abbruch und im Anschluss neu angefragt wird.

2.1.3 Generierung Aval-Id

Die Generierung der Aval-Id obliegt grundsätzlich dem Partner der ein Matching zuerst vorschlägt. Er muss gewährleisten, dass diese ID in seinem System eindeutig für ein Matching mit dem angestrebten Partner-System ist. Sollte per Zufall ein anderer Teilnehmer die gleiche Avalld bilden ist die Eindeutigkeit dadurch gesichert, dass die Systeme wissen zwischen welchen Eigen- und Fremd-System die Beziehung besteht:

- Avalld: 0815 Client Veolia, Supplier Meinhard
- Avalld: 0815 Client Veolia, Supplier Zentek
- Avalld: 0815 Client Meinhard, Supplier Zentek

Eine eindeutigen Aval-ID technisch zu generieren ist eine 40-stellige UUID nach dem [RFC 4122](#) durchzuführen. Dabei wird davon ausgegangen, dass der Partner die Eindeutigkeit des Schlüssels vor der Annahme der ID aus seinem System heraus testet. Damit wird die Wahrscheinlichkeit weiter erhöht, die ID eindeutig zu halten.

Im oberen Beispiel sind alle Matchings eindeutig zu identifizieren obwohl immer die Avalld 0815 gewählt wurde. Da bei jedem Matching die Systeme bekannt sind.

Wichtig:

Dadurch liegt es in der Verantwortung der implementierten Schnittstelle bei einer Nachricht an <http://www.aval-interface-A.org/avalmatching/0815> zu berücksichtigen von welchem System die Nachricht kommt. Schickt Meinhard eine Nachricht an Veolia, muss Veolia den Veolia-Meinhard-Datensatz aktualisieren und nicht den Veolia-Zentek-Datensatz.

Es darf auch ein Matching von unterschiedlichen Systemen mit gleicher Aval-Id nicht einfach abgelehnt werden. So lange die teilnehmenden System sich unterscheiden kann die Avalld mehrfach auftreten.

2.2 AvalState - Mögliche Werte

Aval-State	Aval-State-Nummer	Beschreibung
initiated	1	Standardwert bei neuer Anlage. Es muss seitens des Partners ein Matching erfolgen.
matched	2	Aktives zugeordnetes Matching. Es können Aufträge zu diesem ausgetauscht werden.
cancelled	3	Matching wurde abgebrochen. Dies kann passieren, wenn der Partner kein Matching bei sich finden kann oder ein Matching beendet werden soll.

2.3 Role - Mögliche Werte

AvaL-State	AvaL-State-Nummer	Beschreibung
client	1	Auftragnehmer
supplier	2	Auftraggeber

3. Vorgang zu einem Matching durchführen

Nachdem das Matching zwischen beiden Parteien besteht kann jeder von beiden im Anschluss einen Vorgang auslösen.

Restaufrufe werden mit den Parametern, Version (Aktuelle Version der Schnittstellen), avalmatching (AvaL-ID) und transactions (jeweilig gültige Transaktion) durchgeführt. Wenn der Parameter transactions weggelassen wird, liefert ein GET eine Liste aller transactions.

Der GET-Aufruf mit transactions-ID liefert den vollständigen, aktuell gültigen Datensatz mit allen Feldern.

Die jeweils möglichen REST-Aufrufe sind in jedem Unterpunkt aufgeführt (PUT, PATCH).

Dazu dienen die Daten von folgendem Objekt:

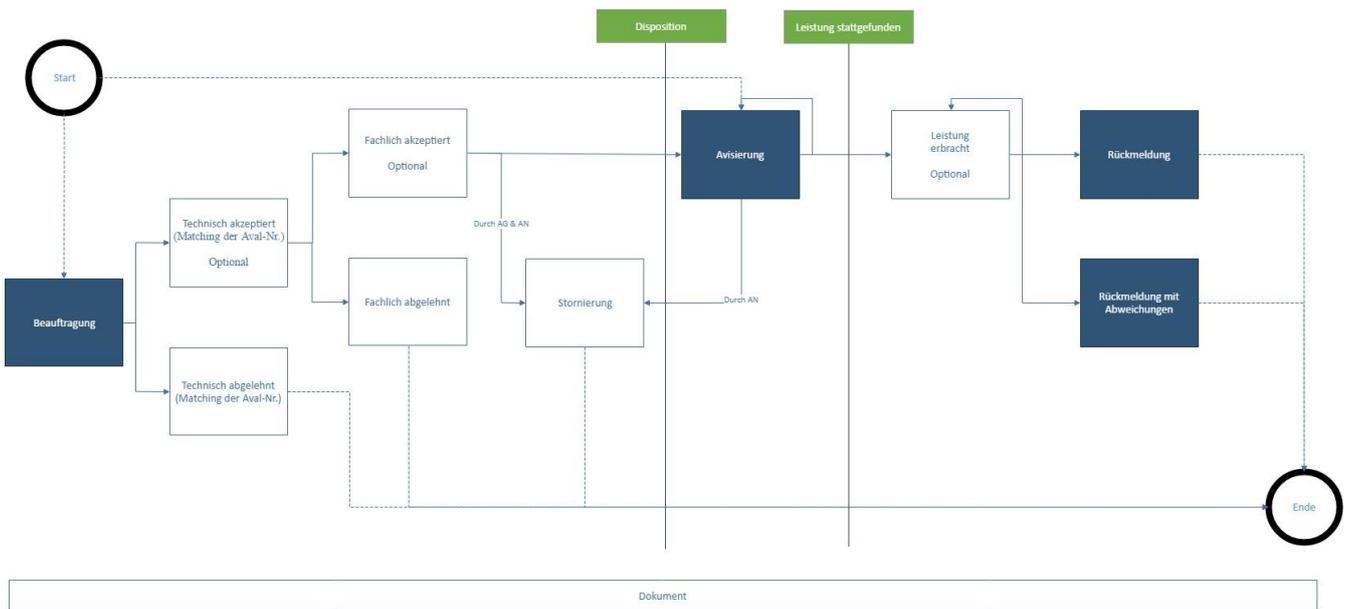
```
{
  "avalTransaction": {
    "state": 1,
    "avalId": "veolia4711A0815B",
    "transactionId": "AG_3HJKGDAKSGF7",
    "messageTimestamp": "01-01-02018T00:00:00Z",
    "operationStartTimestamp": "01-01-02018T00:00:00Z",
    "operationEndTimestamp": "01-01-02018T00:00:00Z",
    "fullfillmentTimestamp": "01-01-02018T00:00:00Z",
    "plannedFullfillmentStartTimestamp": "01-01-02018T00:00:00Z",
    "plannedFullfillmenteEndTimestamp": "01-01-02018T00:00:00Z",
    "serviceAmount": 1,
    "containerAmount": 1,
    "materialAmount": 1.5,
    "isUnderMeasureThreshold": false,
    "serviceNoteNumber": "DL-4711",
    "measureNoteNumber": "WS-0815",
    "governmentalAssetNumber": "",
    "governmentalCarrierNumber": "",
    "OrderNumberSupplier": "123456",
    "OrderNumberClient": "123A456",
```

```
"logisticComments": "Bitte hinten rum fahren",
"cancellationReason": "Tor verschlossen",
"complaintReason": 200,
"variationNotes": "",
"alternateInitialAvalId": "",
"alternateMatchedAvalId": "",
"file": "",
"fileName": "",
"fileFormat": "",
"fileType": ""
}
}
{
"$schema": "http://json-schema.org/draft-06/schema#",
"$ref": "#/definitions/AvalTransaction",
"definitions": {
  "AvalTransaction": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "avalTransaction": {
        "$ref": "#/definitions/AvalTransactionClass"
      }
    },
    "required": [
      "avalTransaction"
    ],
    "title": "AvalTransaction"
  },
  "AvalTransactionClass": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "state": {
        "type": "integer"
      },
      "avalId": {
        "type": "string"
      },
      "transactionID": {
        "type": "string"
      },
      "messageTimestamp": {
        "type": "string"
      }
    }
  }
}
```

```
"operationStartTimestamp": {
  "type": "string"
},
"operationEndTimestamp": {
  "type": "string"
},
"fullfillmentTimestamp": {
  "type": "string"
},
"plannedFullfillmentStartTimestamp": {
  "type": "string"
},
"plannedFullfillmenteEndTimestamp": {
  "type": "string"
},
"serviceAmount": {
  "type": "integer"
},
"containerAmount": {
  "type": "integer"
},
"materialAmount": {
  "type": "number"
},
"isUnderMeasureThreshold": {
  "type": "boolean"
},
"serviceNoteNumber": {
  "type": "string"
},
"measureNoteNumber": {
  "type": "string"
},
"governmentalAssetNumber": {
  "type": "string"
},
"governmentalCarrierNumber": {
  "type": "string"
},
"OrderNumberSupplier": {
  "type": "string"
},
"OrderNumberClient": {
  "type": "string"
},
"logisticComments": {
```

```
        "type": "string"
      },
      "cancellationReason": {
        "type": "string"
      },
      "complaintReason": {
        "type": "integer"
      },
      "variationNotes": {
        "type": "string"
      },
      "alternateInitialAvalId": {
        "type": "string"
      },
      "alternatematchedAvalId": {
        "type": "string"
      },
      "file": {
        "type": "string"
      },
      "fileName": {
        "type": "string"
      },
      "fileFormat": {
        "type": "string"
      },
      "fileType": {
        "type": "string"
      }
    },
    "required": [
      "avalId",
      "transactionID",
      "messageDate"
    ],
    "title": "AvalTransaction"
  }
}
```

Ein Vorgang läuft nach folgendem Diagramm ab:



3.1 Beauftragung aufgeben (AG->AN)

Eine Vorgang kann mit zwei Status beginnen. Im Regelfall fängt er mit dem Status 1 (Beauftragung) an. Für eine Beauftragung muss mindestens folgendes JSON geschickt werden:

```

HTTP-PUT:
http://www.aval-interface-B.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 1,
    "avalId": "veolia4711A0815B",
    "transactionId": "AG_1",
    "messageTimestamp": "01-01-02018T00:00:00Z",
    "operationStartTimestamp": "01-01-02018T00:00:00Z",
    "operationEndTimestamp": "01-01-02018T00:00:00Z",
    "serviceAmount": 1
  }
}
    
```

Wichtig sind für die Beauftragung die Pflichtfelder, sowie der Einsatzzeitraum (OperationTimeStart/End). Die transactionID muss für dieses Mapping eindeutig sein. Dies wird gewährleistet indem beide Partner eine TransactionID hochzählen und mit AG (Auftraggeber) oder AN (Auftragnehmer) für den Vorgang schicken. Nach Anlage in beiden Systemem kann über die TransactionID in Zusammenhang mit einer Avalld ein Vorgang eindeutig identifiziert werden.

3.2 Technisch akzeptiert (optional)(AN->AG)

Je nach implementation der Schnittstelle kann die Verarbeitung der Beauftragung synchron oder asynchron erfolgen. Im asynchronen Fall weiß der Sender nicht ob das Fremdsystem die Nachricht akzeptiert. Hier kann Optional das Fremdsystem eine Statusmeldung schicken, um die erfolgreiche Annahme zu quittieren. Die Antwort kann natürlich um optionale Felder (den OrderNumbers von AG/AN) angereichert werden. Diese

Informationen helfen einem Anwender sich über den Zustand des Vorgangs zu Informieren uns sollten mitgeliefert werden, falls möglich.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 2,
    "avalId": "veolia4711A0815B",
    "transactionId": "AG_1",
    "messageTimestamp": "01-01-02018T00:00:00Z"
  }
}
```

3.3 Technisch abgelehnt (optional)(AN->AG)

Je nach implementation der Schnittstelle kann die Verarbeitung der Beauftragung synchron oder asynchron erfolgen. Im asynchronen Fall weiß der Sender nicht ob das Fremdsystem die Nachricht akzeptiert. Im Fehlerfall kann mit einer Nachricht vom Status 3 eine technische Ablehnung erfolgen.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 3,
    "avalId": "veolia4711A0815B",
    "transactionID": "AG_1",
    "messageTimestamp": "01-01-02018T00:00:00Z"
  }
}
```

Sollte nach einem Endstatus eine weitere Nachricht auf eine Aval-Transaktion erfolgen (Ausgenommen hiervon sind Dokumente), gibt es eine Fehlermeldung. Details im Abschnitt Fehlermeldungen.

3.4 Fachlich akzeptiert (optional)(AN->AG)

Neben technischen Meldung kann es auch zu einer Fachlichen Rückmeldung kommen. Eine fachliche Akzeptanz wird durch Status 4 gekennzeichnet.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 4,
    "avalId": "veolia4711A0815B",
    "transactionID": "AG_1",
    "messageTimestamp": "01-01-02018T00:00:00Z"
  }
}
```

3.5 Fachlich abgelehnt (optional)(AN->AG)

Neben technischen Meldung kann es auch zu einer fachlichen Rückmeldung kommen. Eine fachliche Ablehnung wird durch Status 5 gekennzeichnet. Hier wäre eine fachliche Antwort mit cancellation Reason wünschenswert:

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 5,
    "avalId": "veolia4711A0815B",
    "transactionID": "AG_1",
    "messageTimestamp": "01-01-02018T00:00:00Z",
    "cancellationReason": "Alle Fahrzeuge defekt."
  }
}
```

Sollte nach einem Endstatus eine weitere Nachricht auf eine Aval-Transaktion erfolgen (Ausgenommen hiervon sind Dokumente), gibt es eine Fehlermeldung. Details im Abschnitt Fehlermeldungen.

3.6 Stornierung (optional)(AN->AG; AG->AN)

Eine Storno kann bis zur Avisierung beidseitig erfolgen. Sie unterscheidet sich eigentlich nicht von der fachlichen Ablehnung

```
HTTP-PATCH:
http://www.aval-interface-B.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 6,
    "avalId": "veolia4711A0815B",
    "transactionID": "AG_1",
    "messageTimestamp": "01-01-02018T00:00:00Z",
    "cancellationReason": "Alle Fahrzeuge defekt."
  }
}
```

3.7 Avisierung (AN->AG)

Die Avisierung ist die Einplanung einer Transaktion zu einem geplanten Zeitraum. Nach der Avisierung darf eine Transaction nicht mehr storniert werden. Bei Problemen sind Rückmelde-Status vorgesehen. Die Avisierung darf nur vom Auftragnehmer an den Auftraggeber gesendet werden.

Die Avisierung ist neben der Beauftragung auch der zweite Einstiegspunkt. Es kann vorkommen, dass eine Avisierung direkt durch den Auftragnehmer erzeugt wird.

```
HTTP-PATCH:
http://www.aval-interface-B.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
HTTP-PUT:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AN_1
{
```

```
"avalTransaction": {
  "state": 7,
  "avalId": "veolia4711A0815B",
  "transactionID": "AG_1",
  "messageDate": "01-01-02018T00:00:00Z",
  "plannedFullfillmenteStartTimestamp": "01-01-02018T00:00:00Z",
  "plannedFullfillmenteEndTimestamp": "01-01-02018T00:00:00Z"
}
```

3.8 Leistung erbracht (optional) (AN->AG)

Die Info Leistung erbracht ist ein optionaler Status. Hier kann mitgegeben werden wann die Transaktion tatsächlich durchgeführt wurde. Davon abgesehen sind keine Informationen verpflichtend. Der Empfänger kann dieses Datum bereits anzeigen, aber bekommt später noch eine extra Rückmeldung.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 8,
    "avalId": "veolia4711A0815B",
    "transactionID": "AG_1",
    "messageDate": "01-01-02018T00:00:00Z",
    "fullfillmentTimesstamp": "01-01-02018T00:00:00Z"
  }
}
```

3.9 Rückmeldung (AN->AG)

Die Info Leistung erbracht ist ein optionaler Status. Hier kann mitgegeben werden wann die Transaktion tatsächlich durchgeführt wurde. Davon abgesehen sind keine Informationen verpflichtend. Der Empfänger kann dieses Datum bereits anzeigen, aber bekommt später noch eine extra Rückmeldung.

```
HTTP-PUT:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1
{
  "avalTransaction": {
    "state": 9,
    "avalId": "veolia4711A0815B",
    "transactionID": "AG_1",
    "messageTimestamp": "01-01-02018T00:00:00Z",
    "fullfillmentTimestamp": "01-01-02018T00:00:00Z",
  }
}
```

```
"serviceAmount": 8,  
"isUnderMeasureThreshold": false,  
"serviceNoteNumber": "123aa",  
"governmentalAssetNumber": "123a",  
"governmentalCarrierNumber": "456b",  
}  
}
```

3.10 Rückmeldung mit Abweichungen (AN->AG)

Die Info Leistung erbracht ist ein optionaler Status. Hier kann mitgegeben werden wann die Transaktion tatsächlich durchgeführt wurde. Davon abgesehen sind keine Informationen verpflichtend. Der Empfänger kann dieses Datum bereits anzeigen, aber bekommt später noch eine extra Rückmeldung.

```
HTTP-PATCH:  
http://www.aval-interface-A.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1  
{  
  "avalTransaction": {  
    "state": 10,  
    "avalId": "veolia4711A0815B",  
    "transactionID": "AG_1",  
    "messageTimestamp": "01-01-02018T00:00:00Z",  
    "fullfillmentTimestamp": "01-01-02018T00:00:00Z",  
    "complaintReason": 300,  
  }  
}
```

3.11 Dokument (AN->AG, AG->AN)

Dokumente können jederzeit in beide Richtungen verschickt werden. Dafür sind lediglich die normalen Pflichtfelder sowie die Dokumentklasse, fachliche Dokumentklasse und das Dokument als Base64-String notwendig. Es passiert in dem Sinne kein "Statuswechsel". Der Status dient der Info, dass die Nachricht eine Datei ist.

```
HTTP-PUT:  
http://www.aval-interface-  
B.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1/avaldocuments/DAG_1  
HTTP-PATCH:  
http://www.aval-interface-  
B.org/v1/avalmatchings/veolia4711A0815B/avaltransactions/AG_1/avaldocuments/DAG_1  
{  
  "avalDocument": {  
    "avalId": "veolia4711A0815B",  
    "transactionId": "AG_1",  
    "documentId": "DAG_1",  
  }  
}
```

```

"messageTimestamp": "01-01-02018T00:00:00Z",
"file": "123sad123sae1esae1...",
"fileName": "bild.jpg",
"fileFormat": "jpg",
"fileType": 7
}
}

```

4.1 Anmerkungen zum Möglichen Datenmodell

Damit das obere Prinzip funktionieren kann gibt es eine Reihe von Nebenbedingungen die abgespeichert werden sollten.

Zu Speichernder Wert	Aval-Objekt	Erläuterung
An API angemeldeter System-User	Matching+Durchführung	Es sollte abgespeichert werden welcher System-User ein Objekt an das System schickt, damit intern zugeordnet werden kann welches Fremdsystem die Daten ausgetauscht hat. Eine Aval-Id kann nur je Fremdsystem eindeutig sein.
Client	Matching	Anlage eines Vorgangs sollte abgespeichert werden ob das eigene System Client oder Supplier ist.
Supplier	Matching	Anlage eines Vorgangs sollte abgespeichert werden ob das eigene System Client oder Supplier ist.

4.2 Fehlermeldungen

Im Austausch kann auch eine Menge schief gehen folgendes Szenarien sollen als technische Fehler bereits definiert sein:

Fehlerszenario	HTTP-Code	Beispiel
ID der Resource (in der URL) und Id im Objekt stimmen nicht überein.		URL: .../avalID/111 JSON-Value: "avalID": "123"
Transaktion im Endstatus, keine		

Aktualisierung möglich.		

5 Sicherheit

Damit die AvaL Schnittstelle vor dem Zugriff durch Unbefugte geschützt ist soll die Schnittstelle mit einem der gängigen Standards zu Absicherung von Web APIs ausgestattet werden.

Weiterhin sollen Aufrufe der API nur verschlüsselt (SSL) erfolgen, da hier unternehmenskritische und personenbezogene Daten ausgetauscht werden.

Als Empfehlung wird das offene Protokoll OAuth2 für die Autorisierung benannt. Das Protokoll ermöglicht die Autorisierung von Zugriffen an der Schnittstelle ohne Übermittlung von Benutzer / Passwort Kombinationen. Stattdessen werden sogenannte Access Token an die AvaL Schnittstelle übergeben. Diese werden durch die Schnittstelle selbst auf Gültigkeit überprüft. Erst dann werden die gewünschten Daten geliefert und entgegen genommen. Das Verfahren ist zustandslos. Dies bedeutet, dass bei jedem Aufruf der AvaL API das Access Token mitgeliefert werden muss. Die AvaL API muss also kein Session-Handling für die Autorisierung implementieren. Die Access Token müssen durch den API Client im Vorfeld von einem OAuth2 Server abgerufen werden. Access Token haben üblicherweise eine begrenzte Gültigkeit und müssen daher nach Ablauf erneut angefordert werden. In der Zwischenzeit kann der API Client das Token cachen und wiederverwenden. Zusätzlich sollte der Inhalt des Access Token festgelegt werden. Hierfür wird das Json Web Token kurz JWT vorgeschlagen. In dieser Kombination beschreibt OAuth2 wie ein Access Token für die Autorisierung verwendet werden soll und JWT legt den Inhalt des Tokens fest.

Vorteile des Verfahrens:

- Standardisiert (OAuth2 - RFC 6749, JWT - RFC 7519)
- Vorhandene Unterstützung in gängigen Frameworks und Bibliotheken
- Der OpenAPI Standard enthält einen Security Scheme welches OAuth2 unterstützt (siehe hier)
- AvaL API muss keine Benutzerverwaltung implementieren, dies übernimmt der OAuth2 Service
- Optional besteht die Möglichkeit einen zentralisierten OAuth2 Service für alle AvaL APIs zu bereitzustellen
- OAuth2 Server lassen sich, für den Fall das man den Server nicht selbst bereitstellen möchte, bei diversen Cloud Anbietern (bsp. Amazon Cognito) mieten

7 Dokumentenhistorie

Erstellungs-/ Änderungsdatum:	Erstellt / Geändert von:	Änderungen:

Prüfdatum:	Freigabedatum:
Geprüft von: [Vorname Nachname] Unterschrift	Freigegeben von: [Vorname Nachname] Unterschrift