



## **AvaL Projekt 1 - Technisches Konzept V1.5**

## Inhalt:

<b>1 Ausgangssituation</b> .....	<b>4</b>
<b>2 Austausch und Anlage der AvaL-Vereinbarungen</b> .....	<b>5</b>
2.1 Austausch von AvaL-Vereinbarungen.....	5
2.1.1 Abbrechen eines Matchings .....	7
2.1.2 Ändern eines Matchings.....	7
2.1.3 Generierung AvaL-ID .....	8
2.2 AvaL Status - Mögliche Werte .....	8
2.3 Best Practise bei Festlegung der Leistung eines Matchings .....	8
<b>3 Vorgang zu einem Matching durchführen</b> .....	<b>9</b>
3.1 Beauftragung aufgeben (AG->AN).....	11
3.2 Technisch akzeptiert (AN->AG) [optional].....	11
3.3 Technisch abgelehnt (AN->AG) [optional].....	11
3.4 Fachlich akzeptiert (AN->AG) [optional].....	12
3.5 Fachlich abgelehnt (AN->AG) [optional].....	12
3.6 Stornierung (AN->AG; AG->AN) [optional].....	12
3.7 Avisierung (AN->AG) .....	13
3.8 Leistung erbracht (AN->AG) [optional].....	13
3.9 Rückmeldung (AN->AG).....	14
3.10 Rückmeldung mit Abweichungen (AN->AG) [alternativ].....	14
3.11 Dokument (AN->AG, AG->AN) .....	14
3.12 Abbruchstatus aufgrund unerwarteter Probleme.....	15
3.12 Pflichtattribute je Status .....	16
<b>4 Anmerkungen zum Datenmodell</b> .....	<b>17</b>
4.1 Extended Information .....	17
4.2 Datumsformat.....	17
4.3 Interne Erweiterungen .....	17
4.4 Response Timeouts bei Kommunikation zwischen den Partner – APIs .....	18
4.5 Beschränkungen von Zeichenlängen im Datenmodell .....	18
4.6 OpenAPI / Swagger Spezifikation.....	18
<b>5 HTTP Status Codes</b> .....	<b>19</b>
<b>6 Sicherheit</b> .....	<b>20</b>

## Änderungshistorie:

<b>Version</b>	<b>Änderungen</b>
2019-11 / 23.10.2019	Erstellung Hartmut Möllmann – Dr. Ing. Wandrei GmbH
2020-05 / 18.05.2020	AvaL Layout & Anpassung auf aktuellen API Stand, Christian Urban – GIPA mbH
2020-06 / 19.06.2020	Korrektur Datumsformat, Christian Urban – GIPA mbH
2020-11 / 03.11.2020	Tabellarische Form der Transaction Pflichtfelder, Christian Urban – GIPA mbH
2022-04 / 12.04.2022	Diverse Ergänzungen, Christian Urban – GIPA mbH
2022-11 / 15.11.2022	Diverse Ergänzungen & Korrekturen, Christian Urban – GIPA mbH

---

## 1 Ausgangssituation

Deutschlandweit tauschen Entsorger 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.

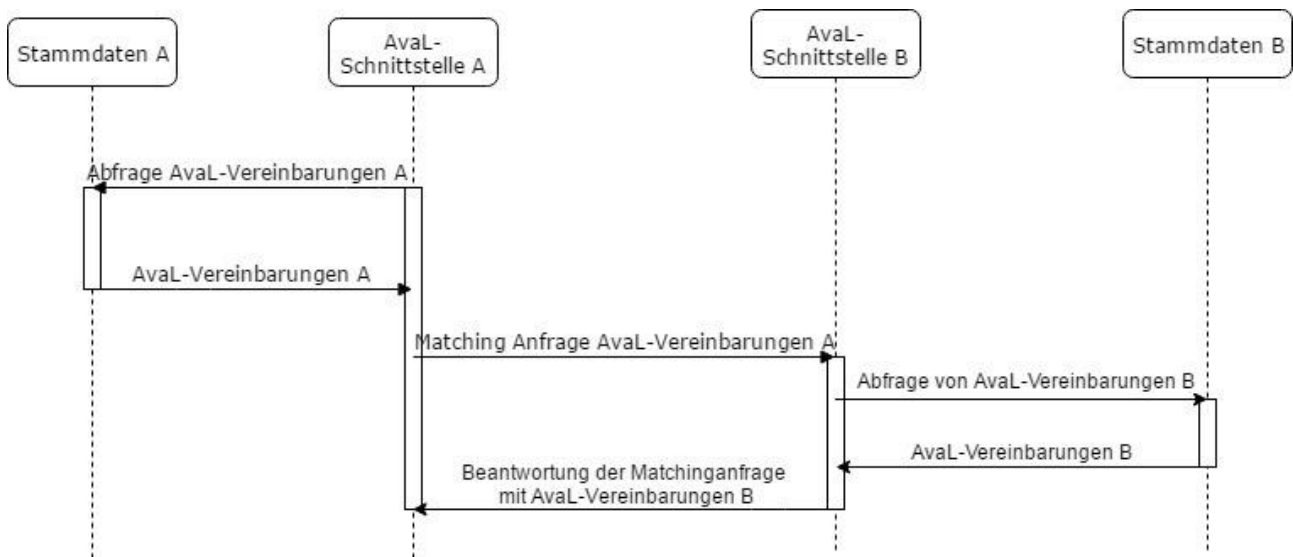
Damit dieser Datenaustausch einfacher und schneller vonstattengeht, hat der Bund Deutscher Entsorger (BDE) eine Arbeitsgruppe beauftragt ein Konzept für eine digitale Schnittstelle zu entwerfen und zu testen. Die Arbeitsgruppe bestehend aus den Firmen Meinhardt, Veolia, Logex, Interseroh, Zentek, GIPA hat in einem Treffen am 07.11.2018 und 08.11.2018 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 AvaL-ID (**AvaL - Austausch von auftragsbezogenen Leistungsdaten**) sein, die zur 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 und Leistungsadresse 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/avalmatchings/

{
  "state": 1,
  "id": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "client": {
    "contractPartner": "MaxMustermann GmbH",
    "containerType": "",
    "material": "Papier",
    "serviceType": "Austausch",
    "performancePlace": "Musterstraße 43, 22041 Hamburg",
    "materialUnit": "",
    "serviceUnit": "S",
    "cycleInformation": false,
    "cycleRhythm": 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 (Auftraggeber und Auftragnehmer). Der AvaL-Status dient der Erkennung des Zustands des Matchings. Mögliche Zustände werden später aufgelistet.

Die AvaL-ID 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 erleichtern. 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 AvaL-ID "043fb274-21da-482a-96ef-ed7e666fdf01" geschickt wurde:

```
HTTP-PATCH:
http://www.aval-interface-A.org/avalmatchings/043fb274-21da-482a-96ef-ed7e666fdf01

{
  "state": 2,
  "id": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "client": {
    "contractPartner": "M. Mustermann GmbH",
    "containerType": "",
    "material": "Papier",
    "serviceType": "Austausch",
    "performancePlace": "Musterstraße 43, 22041 Hamburg",
    "materialUnit": "",
    "serviceUnit": "S",
    "cycleInformation": false,
    "cycleRhythm": null
  },
  "supplier": {
    "contractPartner": "Max Musterman GmbH",
    "containerType": "UL",
    "material": "PPK",
    "serviceType": "Tausch",
    "performancePlace": "Muster Strasse 43 Hamburg",
    "materialUnit": "",
    "serviceUnit": "S",
    "cycleInformation": false,
    "cycleRhythm": "2x 14-tägig"
  }
}
```

Die Antwort des Matchingpartners besteht nun aus einem Update (HTTP-PATCH) auf die bekannte AvaL-ID. 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 im oberen Beispiel auch Informationsunterschiede deutlich gemacht. Der Client (Auftraggeber) speichert den Kunden als "M. Mustermann GmbH", der Supplier (Auftragnehmer) als "Max Mustermann GmbH". 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.

```
HTTP-PATCH:
http://www.aval-interface-A.org/avalmatchings/043fb274-21da-482a-96ef-ed7e666fdf01

{
  "state": 3,
  "id": "043fb274-21da-482a-96ef-ed7e666fdf01"
}
```

### 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 eine Anpassung vortäuschen, 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. Um eine eindeutige AvaL-ID technisch zu generieren ist eine 40-stellige UUID nach dem [RFC 4122](#) zu erzeugen. 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.

## 2.2 AvaL Status - Mögliche Werte

AvaL-State	AvaL-State-Nummer	Beschreibung
initiated	1	Standardwert bei Neuanlage. 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 Best Practise bei Festlegung der Leistung eines Matchings

### Fragestellung:

Werden mehrere Behälter mit gleicher Fraktion an ein und demselben Standort in einem Matching abgebildet oder erhält jeder einzelne Behälter sein eigenes individuelles Matching?

### Idee:

Die Anzahl der Transporte ergeben die Anzahl der Matchings. Sind es zwei Transporte dann werden zwei AvaL Matching IDs vergeben. Ist nur ein Transport notwendig, dann wird nur eine AvaL Matching ID mit Behältermenge X definiert.

### Beispiele:

- 5 Umleerer mit PPK an einem Standort → eine AvaL ID
- 3 Abrollpressen an einem Standort → 3 AvaL IDs



## 3 Vorgang zu einem Matching durchführen

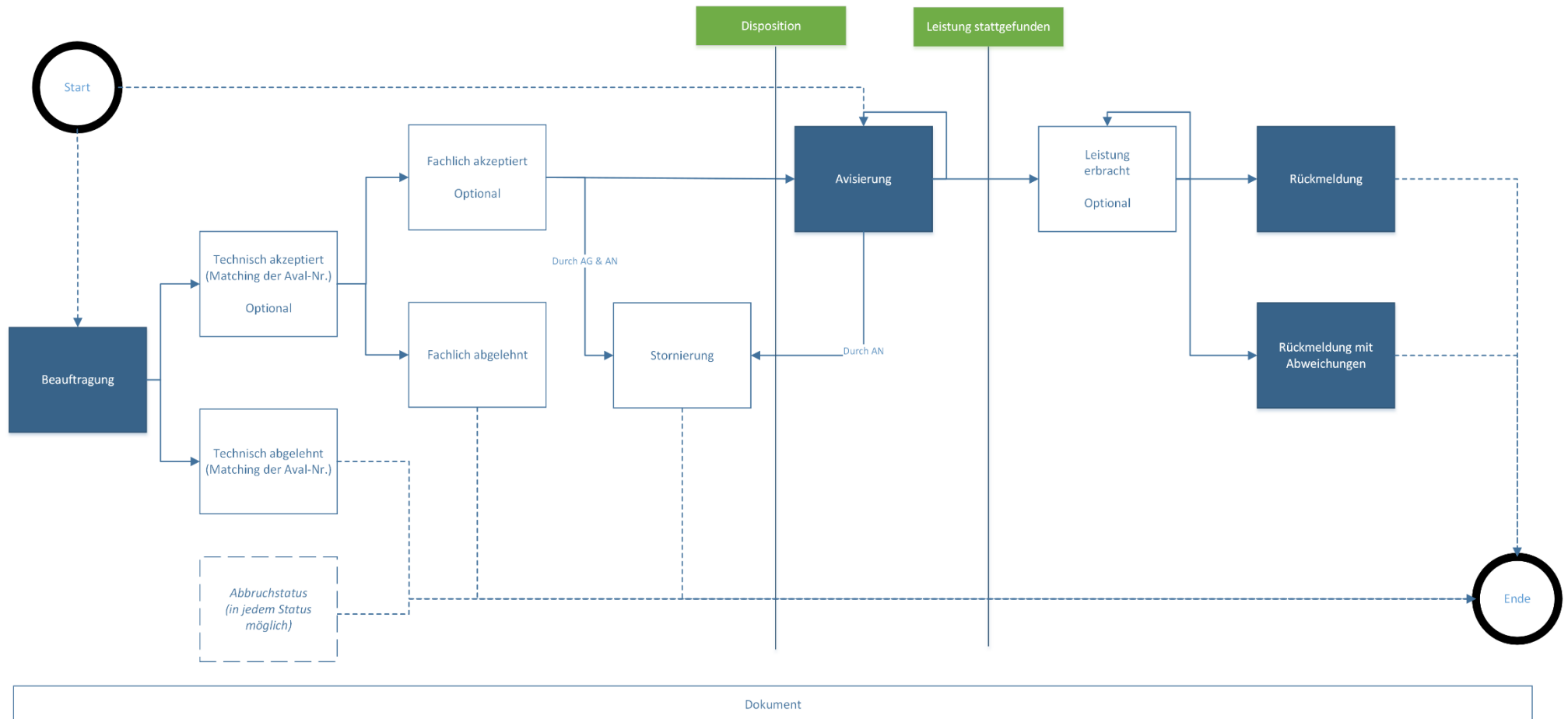
Nachdem das Matching zwischen beiden Parteien besteht kann jeder von beiden im Anschluss einen Vorgang auslösen.

REST-Aufrufe werden mit den Parametern, Version (Aktuelle Version der Schnittstellen), Matching (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 (POST, PATCH).

Dazu dienen die Daten von folgendem Objekt:

```
{
  "state": 1,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20",
  "operationPeriod": {
    "start" : "2020-01-01T00:00:00Z",
    "end": "2020-01-02T00:00:00Z",
  },
  "fullfillmentTimestamp": "2020-01-01T00:00:00Z",
  "plannedFullfillmentPeriod" : {
    "start": "2020-01-01T00:00:00Z",
    "end": "2020-01-02T00:00:00Z"
  },
  "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": "AV.1.104",
  "variationNotes": "",
  "alternateInitialAvalId": "",
  "alternateMatchedAvalId": ""
}
```

Ein Vorgang läuft nach folgendem Diagramm ab:



## 3.1 Beauftragung aufgeben (AG->AN)

Ein 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-POST:
http://www.aval-interface-B.org/v1/avalmatchings/043fb274-21da-482a-96ef-ed7e666fdf01/avaltransactions

{
  "state": 1,
  "avalId": "v043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20",
  "operationPeriod": {
    "start": "2020-01-01T00:00:00Z",
    "end": "2020-01-02T00:00:00Z",
  },
  "serviceAmount": 1
}
```

Wichtig sind für die Beauftragung die Pflichtfelder, sowie der Einsatzzeitraum (*OperationPeriod*). Die Transaction-ID muss für dieses Mapping eindeutig sein. Hierfür wird ebenfalls eine UUID (siehe AvaL-ID) erzeugt. Nach Anlage in beiden Systemen kann über die Transaction-ID in Zusammenhang mit einer AvaL-ID ein Vorgang eindeutig identifiziert werden.

## 3.2 Technisch akzeptiert (AN->AG) [optional]

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 und sollten, falls möglich, mitgeliefert werden.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/043fb274-21da-482a-96ef-ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b2

{
  "state": 2,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20"
}
```

## 3.3 Technisch abgelehnt (AN->AG) [optional]

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/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

{
  "state": 3,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20"
}
```

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 (AN->AG) [optional]

Neben technische 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/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

{
  "state": 4,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20"
}
```

### 3.5 Fachlich abgelehnt (AN->AG) [optional]

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

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

{
  "state": 5,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20"
  "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 (AN->AG; AG->AN) [optional]

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

```
HTTP-PATCH:
http://www.aval-interface-B.org/v1/avalmatchings/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b2

{
  "state": 6,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20"
  "cancellationReason": "Keine freien Kapazitäten."
}
```

## 3.7 Avisierung (AN->AG)

Die Avisierung ist die Einplanung einer Transaktion zu einem geplanten Zeitraum. Nach der Avisierung darf eine Transaction nur noch durch den Auftragnehmer 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/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

HTTP-POST:
http://www.aval-interface-A.org/v1/avalmatchings/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions

{
  "state": 7,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20",
  "plannedFulfillmentPeriod" : {
    "start": "2020-01-01T00:00:00Z",
    "end": "2020-01-01T00:00:00Z"
  }
}
```

## 3.8 Leistung erbracht (AN->AG) [optional]

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/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

{
  "state": 8,
  "avalId": "veolia4711A0815B",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b201",
  "fulfillmentTimestamp": "2020-01-01T00:00:00Z"
}
```

## 3.9 Rückmeldung (AN->AG)

Der Status **Rückmeldung** ist ein obligatorischer Status. Hier muss angegeben werden, wann die Transaktion tatsächlich durchgeführt wurde. Davon abgesehen sind keine Informationen verpflichtend. Abhängig von der erbrachten Leistung sollten Attribute *serviceAmount*, *containerAmount*, *materialAmount*, *isUnderMeasureThreshold*, *measureNoteNumber* usw. verwendet und belegt werden.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

{
  "state": 9,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20",
  "fulfillmentTimestamp": "2020-01-01T00:00:00Z",
  "serviceAmount": 8,
  "isUnderMeasureThreshold": false,
  "serviceNoteNumber": "123aa",
  "governmentalAssetNumber": "123a",
  "governmentalCarrierNumber": "456b",
}
```

## 3.10 Rückmeldung mit Abweichungen (AN->AG) [alternativ]

Der Status **Rückmeldung mit Abweichung** ist ein alternativer Status zur **Rückmeldung**. Er soll verwendet werden, wenn die tatsächliche Leistungserbringung von der vertraglich vereinbarten Leistung abweicht.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/043fb274-21da-482a-96ef-
ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

{
  "state": 10,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01",
  "id": "e1b2a2ab-2f21-4c85-a118-8eb76e347b20",
  "fulfillmentTimestamp": "2020-01-01T00:00:00Z",
  "complaintReason": "AV.1.104",
}
```

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

Dokumente können jederzeit in beide Richtungen verschickt werden. Dokumente werden per HTTP Multipart Request übermittelt. Der Request enthält dabei das Dokument selbst sowie zusätzliche fachliche Attribute wie beispielsweise die Dokumenten-ID oder den ursprünglichen Dateinamen.

HTTP-POST (am Beispiel von curl):

```
Curl -X POST "http://www.aval-interface-B.org/v1/avalmatchings/043fb274-21da-482a-96ef-ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20/avaldocuments"
-H "accept: application/json"
-H "Content-Type: multipart/form-data"
-F "documentId=9dace61f-640b-46c7-8b34-7ca8c69975a7"
-F "filename=Wiegescchein"
-F "fileType=2"
-F "fileFormat=PDF"
-F "file=@Wiegescchein.pdf;type=application/pdf"
```

## 3.12 Abbruchstatus aufgrund unerwarteter Probleme

In besonderen Ausnahmesituationen kann es notwendig werden, den Austausch von Informationen zu einer Transaction komplett zu unterbinden bzw. abzurechnen. Hierfür ist der Status „-1“ zu verwenden. Wurde dieser gesendet und mit dem Partner ausgetauscht ist keinerlei weiterer Austausch für die Transaction mehr möglich. Der Abbruchstatus kann zu jeder Zeit gesendet werden.

```
HTTP-PATCH:
http://www.aval-interface-A.org/v1/avalmatchings/043fb274-21da-482a-96ef-ed7e666fdf01/avaltransactions/e1b2a2ab-2f21-4c85-a118-8eb76e347b20

{
  "state": -1,
  "avalId": "043fb274-21da-482a-96ef-ed7e666fdf01"
}
```

## 3.12 Pflichtattribute je Status

Die nachfolgende Matrix zeigt die Pflichtattribute abhängig vom Status der übermittelten Transaction auf. Ist ein Attribut bei einem Status verpflichtend, so enthält die entsprechende Zelle ein „X“.

Attribut	Zielstatus										
	Beauftragung aufgegeben (1)	Technisch akzeptiert (2)	Technisch abgelehnt (3)	Fachlich akzeptiert (4)	Fachlich abgelehnt (5)	Stornierung (6)	Avisierung (7)	Leistung erbracht (8)	Rückmeldung (9)	Rückmeldung mit Abweichung (10)	Abbruchstatus (-1)
<i>id</i>	X	X	X	X	X	X	X	X	X	X	X
<i>state</i>	X	X	X	X	X	X	X	X	X	X	X
<i>operationPeriod</i>	X										
<i>serviceAmount</i>	X								X		
<i>plannedFulfillmentPeriod</i>							X				
<i>fulfillmentTimestamp</i>								X	X	X	
<i>complaintReason</i>										X	

**Ergänzung:** Insbesondere die Felder, die sich auf eine bestimmte Art von Leistungen beziehen, lassen sich nicht als Pflichtfelder ausprägen. Wäre beispielweise die Angabe eines Gewichts verpflichtend, ließen sich keine reine Behälteraufstellung oder Dienstleistungen à la Straßenreinigung und Co. darstellen. Die Anzahl von Containern wäre für Saugwageneinsätze auch eher untypisch.



## 4 Anmerkungen zum Datenmodell

### 4.1 Extended Information

Das AvaL Datenmodell wurde auf Basis von in der Branche üblichen Informationen konstruiert. Es ist davon auszugehen, dass AvaL Partner darüber hinaus weitere Informationen austauschen müssen. Diese Informationen können über das Attribut *ExtendedInformation* sowohl für Matchings als auch für Transactions übermittelt werden. Das Attribut ist optional.

Die *ExtendedInformation* bestehen aus einem Dictionary / Map vom Typ *AdditionalContent*. Der Key der einzelnen Einträge kann frei gewählt werden, er muss aber einmalig sein. Ein zusätzlicher Inhalt besteht immer aus einer Typisierung und dem zugehörigen Inhalt. Die Typisierung muss zwischen den AvaL Partner abgestimmt werden, andernfalls ist eine maschinelle Verarbeitung des *AdditionalContent* nicht möglich.

```

{
  "extendedInformation": {
    "container1": {
      "type": "Behälternummer",
      "content": "XY-1001",
    },
    "container2": {
      "type": "Behälternummer",
      "content": "XY-7878",
    },
    "customerContainer": {
      "type": "Behälternummer",
      "content": "FB-9999",
    }
  }
}

```

### 4.2 Datumsformat

Daten, die über die API übertragen werden, sind gemäß [RFC 3334 Section 5.6](#) zu kodieren. Der zu verwendende Subtyp ist "date-time".

Beispiele für gültige Zeitstempel:

- „2020-01-01T00:00:00Z“
- „2020-01-01T00:00:00+01:00“

### 4.3 Interne Erweiterungen

Es hat sich als praxistauglich erwiesen, bestimmte Informationen innerhalb der API vorzuhalten bzw. persistiert, diese aber nicht über die API zu veröffentlichen. Diese Informationen dienen beispielsweise der Protokollierung von eingegangenen Anfragen oder der Zuordnung der Daten zu unterschiedlichen AvaL Teilnehmern.

Zu speichernde Information	AvaL-Objekt(e)	Erläuterung
----------------------------	----------------	-------------

An API angemeldeter System-User	Matching + Transaction	Es sollte abgespeichert werden, welcher System-User ein Objekt an das System schickt, damit intern zugeordnet werden kann, welches Fremdsystem die Daten ausgetauscht hat.
Client / Supplier	Matching	Anlage eines Vorgangs sollte abgespeichert werden ob das eigene System Client oder Supplier ist.
Message Timestamp	Matching + Transaction	Protokollierung wann Daten an die API übermittelt wurden.

## 4.4 Response Timeouts bei Kommunikation zwischen den Partner – APIs

AvaL erlaubt, dass die Verbreitung der eingehenden Nachrichten (Matchings / Transactions) synchron oder asynchron erfolgen kann. Dies wirkt sich direkt auf das Kommunikationsverhalten per HTTP / HTTPS Protokoll aus. Nach der Übermittlung einer Nachricht, wartet der Sender auf die Antwort des Empfängers, um anhand des HTTP Status Codes (siehe Abschnitt 5 HTTP Status Codes) auswerten zu können, ob der Request erfolgreich verarbeitet wurde. Abhängig von der Art der Verarbeitung auf der Empfängerseite erfolgt diese Antwort unterschiedlich schnell.

In der Praxis hat sich hier ein Response Timeout von 30 Sekunden als praktikabel erwiesen. Ein unbegrenzter Response Timeout ist dringend zu vermeiden, er führt im schlimmsten Fall zur Blockade der gesamten Kommunikation.

## 4.5 Beschränkungen von Zeichenlängen im Datenmodell

Jedes an AvaL angeschlossene ERP System besitzt aufgrund der verwendeten Datenbanktechnologie und Programmiersprache unterschiedlichste Beschränkung hinsichtlich der maximalen Anzahl von Zeichenketten. Der AvaL Standard gibt keinerlei Beschränkungen vor. Es wird empfohlen, dass Requests nur in Ausnahmefälle aufgrund von zu langen Zeichenketten als ungültig abgelehnt werden. Daher ist jeder Implementierer dafür verantwortlich, dass zu lange Zeichenketten sinnvoll gekürzt werden, bevor diese im angeschlossenen System gespeichert werden.

## 4.6 OpenAPI / Swagger Spezifikation

Die im Abschnitt 2, 3 und 4 beschriebenen Rahmenbedingungen und Details sind in einer sogenannten OpenAPI Spezifikation (vormals bekannt als Swagger) zusammengefasst. Das Open Source Projekt *OpenAPI Specification* ist ein Standard zur Beschreibung von RESTful Web APIs und wird von vielen großen Softwareunternehmen wie Microsoft, IBM und Co. angewendet und unterstützt.

Sie finden die aktuelle Version der OpenAPI Spezifikation im Github Bereich der *AvaL Standard Organization* unter <https://github.com/Entwicklung-AvaL-Standard/aval-api-spec> oder auf der [www.avalstandard.de](http://www.avalstandard.de) Homepage. Sie gilt als Grundlage für die hier vorliegende Dokumentation.

Information zur *OpenAPI Specification* finden Sie unter <https://www.openapis.org/>.

## 5 HTTP Status Codes

Die AvaL API verwendet zur Beantwortung von HTTP Request die im HTTP Standard spezifizierten Response Status Codes ([RFC 7231 - Section 6](#)). Die vollständige Liste der zu implementierenden Codes ist der „AvaL Projekt 1 – OpenAPI Spezifikation“ zu entnehmen. Die AvaL API kann im Fehlerfall eine detaillierte Nachricht im Response Body übermitteln, dies ist jedoch nicht verpflichtend.

Nachfolgend einige Beispiele für mögliche Status Codes:

Fehlerszenario	HTTP-Code
Matching erfolgreich übermittelt (POST)	200
Matching wurde aufgrund fehlender Attribute oder ungültiger Zustände (AvaL State) nicht akzeptiert (POST)	400
Update eines Matching wegen unbekannter AvaL-ID fehlgeschlagen (PATCH)	404
Transaktion im Endzustand, keine Aktualisierung möglich (PATCH)	405
Matching konnte nicht übermittelt werden, weil die AvaL-ID bereits verwendet wird (POST)	409
Matching konnte aufgrund eines unerwarteten Fehlers nicht übertragen werden (POST)	500

## 6 Sicherheit

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

Dazu wird das offene Protokoll OAuth2 (siehe [RFC 6749](#)) zur Autorisierung verwendet. Das Protokoll ermöglicht die Autorisierung von Zugriffen an der Schnittstelle ohne dabei Benutzer – Passwort – Kombinationen zu übermitteln. 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 bzw. 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 demnach 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.

AvaL Standard wird zukünftig einen zentralen OAuth2 Service zur Verfügung stellen. Jeder AvaL Teilnehmer muss technisch in der Lage sein mit dem zentralen Server zu kommunizieren. Abweichend davon dürfen zwei AvaL Teilnehmer miteinander vereinbaren, dass sie eigene OAuth2 Server verwenden. Diese Abweichung beruht auf beiderseitigem Einverständnis.

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