

## Esper-Praktikum

Besprechung: 09. Juli 2019

### Ereignisproduzenten:

*Open Sky Network*: Produziert periodisch Ereignisse vom Typ *StateVector*. Jeder *StateVector* enthält Informationen zu einem aktiven Flug, z.B. dessen Position, Geschwindigkeit und Callsign ([https://en.wikipedia.org/wiki/Aviation\\_call\\_signs](https://en.wikipedia.org/wiki/Aviation_call_signs)).

Weitere Informationen finden Sie unter <https://opensky-network.org/apidoc/>.

*Open Weather Map*: Produziert Wetterereignisse für einen gegebenen Ort.

Weitere Informationen finden Sie unter <https://openweathermap.org/api>.

Bitte beachten Sie, dass Sie für diese API einen Key benötigen: <https://openweathermap.org/api>.

*Booking*: "Künstlicher" Ereignisstrom, der Buchungen von Passagieren zu aktiven Flügen enthält. Die Definition des Ereignistyp finden Sie im Paket *booking* der bereitgestellten Implementierung.

### Ereignisverbraucher:

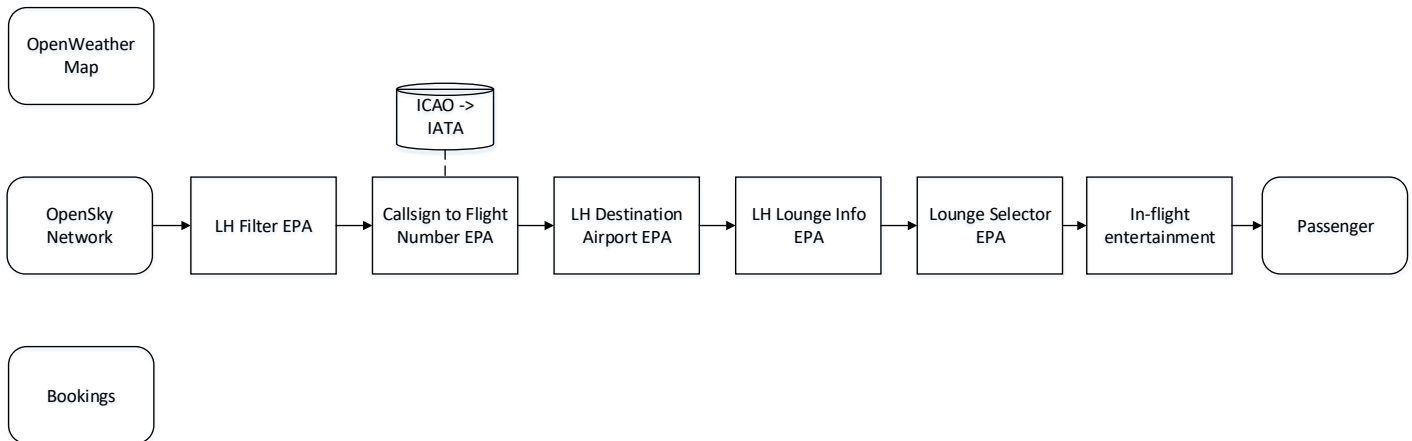
Passagier, der sich in einem aktiven Flugzeug der Airlines Lufthansa, Germanwings, Austrian oder Swiss befindet.

Dieser kann über das In-flight Entertainment (IFE) folgende Informationen einsehen:

1. ETA (estimated time of arrival) am Zielflughafen
2. Hinweis, falls das Flugzeug gerade eine bekannte Stadt überfliegt.
3. Sehenswürdigkeiten/Aktivitäten am Zielort abhängig vom Wetter. Beispielsweise wird bei Regen ein Museum vorgeschlagen.
4. Informationen zu einer Lounge am Zielflughafen (Name, Standort, Wifi- und Duschaumverfügbarkeit). Diese Information steht nur Passagieren zur Verfügung, die mindestens Premium Economy gebucht haben.

Um diese Informationen bereitzustellen, sind eine Reihe von Event Processing Agents (EPAs) notwendig. In der unteren Abbildung sehen Sie einen Ausschnitt des Event Processing Networks (EPN), das das beschriebene Szenario implementiert.

In diesem Netzwerk sind bereits alle Verbraucher und Erzeuger abgebildet sowie EPAs zur Berechnung der Loungeinformationen. Beachten Sie allerdings, dass noch nicht zwischen Passagieren mit verschiedenen Buchungsklassen unterschieden wird.



Die Anfragen, die den jeweiligen EPAs zu Grunde liegen, finden Sie in der bereitgestellten Klasse *EPN.java*.

### Aufgaben

- Geben Sie die Typen der gegebenen EPAs an.
- Erweitern Sie das EPN so, dass auch die Buchungsklasse des Passagiers für die Loungeinformationen berücksichtigt werden.
- Erweitern Sie die bereitgestellte Implementierung entsprechend.
- Erweitern Sie das EPN so, dass auch die restlichen IFE-Informationen bereitgestellt werden.
- Erweitern Sie die bereitgestellte Implementierung entsprechend.

Das IFE soll nun auch die Funktionalität haben, Passagieren das Gate ihres etwaigen Anschlussfluges mitzuteilen.

- Erweitern Sie zunächst die Klasse *Lufthansa.java*, sodass es nun auch Funktionen zur Bestimmung der Abflug- und Ankunftszeiten und der Abflug- und Ankunftsterminals/-gates gibt.
- Erweitern Sie den Booking Ereignistyp, sodass es auch ein Attribut für einen etwaigen Anschlussflug gibt.
- Senden Sie entsprechende Ereignisse an den Booking Ereignisstrom.
- Erweitern Sie das EPN, sodass Passagieren das Gate ihres etwaigen Anschlussfluges angezeigt wird.
- Erweitern Sie die bereitgestellte Implementierung entsprechend.

Schließlich soll dem IFE eine weitere Funktionalität Ihrer Wahl hinzugefügt werden. Diese soll mindestens zwei Ereignisproduzenten verwenden (Sie können bei Bedarf neue Ereignistypen definieren) und von mindestens zwei EPAs realisiert werden.

- Erweitern Sie das EPN entsprechend.
- Erweitern Sie die Implementierung entsprechend.

---

Freiwillige Zusatzaufgabe: für jede Buchung mit Anschlussflug (Connecting Flight) wird der Passagier informiert, falls aufgrund von Verspätungen der Anschluss voraussichtlich nicht erreicht wird.

m) Erweitern Sie die Implementierung entsprechend. Verwenden Sie dazu ein EPL-Pattern.

*Hinweis: Anschlussflüge sind selten, die Gefahr einen solchen zu verpassen noch seltener. Es kann also sein, dass Sie keine entsprechenden Flüge beobachten.*