### Aufgabe 2 b) Analyse des Use-Case „Ticket Buchung“

Dieser Abschnitt umfasst das Feststellen der NFAs für den Use-Case „Ticket Buchung“, Kommunikationsarten für Systeme, welche Kommunikationsart für das geforderte System am sinnvollsten ist und wie sich die ausgewählte Kommunikationsart auf die NFAs auswirkt. Um die NFAs für diesen Use-Case bestimmen zu können, wird zunächst betrachtet, welche Komponenten benötigt werden und wie überhaupt wie eine „Ticket Buchung“ ablauft.

Durch die Anforderungen an das System, ist das klassische kaufen eines Tickets nicht mehr notwendig, da das System eine bzw. mehrere Fahrten in einem bestimmten Zeitraum zusammenrechnet und somit eine optimale Abrechnung für den Zuggast bereitstellt. Es steht dem Zuggast jedoch frei, eine Reservierung über das Internet oder einen Bahnschalter zu tätigen. Mit diesen beiden Fällen ist der eigentliche Use-Case gemeint, da hier eine vorab Buchung stattfindet.

Die beiden Fälle unterscheiden sich jedoch nicht wirklich. Nur der Einstiegspunkt für den Prozess ist anders. Bei einem Fall nutzt der Zuggast die Möglichkeit über die Internetseite der Bahn, eine Fahrt zu reservieren, was eine Buchung der Strecke automatisch beinhaltet. Dies findet also auf einem externen Gerät statt, welches nicht der Bahn gehört. Da die Internetseite jedoch auf Hardware der Bahn gehostet ist, wird der Prozess jedoch Bahn intern angestoßen. Ähnlich findet der Prozess statt, wenn der Zuggast eine Reservierung über einen Bahnschalter tätigt. Dieser wird auch von der Bahn selbst verwaltet und ist somit intern. Daraus folgt das der Prozess immer auf interner Seite angestoßen wird.

Nachdem der Zuggast also eine Reservierung/Buchung vorgenommen hat, wird das Ticket-Buchungssystem angesteuert. Dort wird dann das Abrechnungssystem angesteuert, damit auch sichergestellt wird, dass das gebuchte Ticket bezahlt wird, da eine weitere Bearbeitung des Prozesses sonst keinen Sinn machen würde. Nachdem sichergestellt worden ist, dass das Ticket bezahlt ist, wird ggfs. das Sitzplatz-Reservierungssystem darüber informiert, dass es in der Sitzplatzverwaltung eine neue Reservierung hinterlegen muss. Zum gleichen Zeitpunkt kann jedoch dem Zuggast bereits ein Feedback gegeben werden, dass die Abbuchung erfolgreich war und sein Ticket im System hinterlegt ist.

Platz für NFA

Nachdem wir im oberen Abschnitt eine Analyse durchgeführt haben, welche NFAs wichtig für unseren Use-Case sind, schauen wir uns nun an, wie sich das System unter den jeweiligen Kommunikationsarten verhalten würde. Zum besseren Verständnis, folgt zunächst aber eine Erläuterung zu den drei Kommunikationsarten synchron, asynchron und caching.



Abbildung Unterschied zwischen Synchroner und Asynchroner Kommunikation

Anhand von Abbildung 1 kann man gut den unterschied von synchroner und asynchroner Kommunikation erklären. Synchrone und asynchrone Kommunikation funktionieren so, dass ein Prozess (Prozess A) beispielweise einen anderen Prozess (Prozess B) anstößt. Bis hier hin unterscheiden sich synchron und asynchron noch nicht. Der entscheide unterschied folgt nun, denn synchrone Kommunikation unterbricht Prozess A und wartet bis Prozess B abgeschlossen ist und gegebenenfalls einen Rückgabewert liefert. Da bedeutet also, sollte Prozess B einen längeren Zeitraum benötigen, so wartet Prozess A auch so lange. Daraus ergibt sich dann folgende Laufzeit:

Bei der asynchronen Kommunikation hingegen, wird Prozess A nicht unterbrochen und läuft weiter durch. Also:

Wann sollte man nun eine synchrone und wann eine asynchrone Kommunikation verwenden? Grundsätzlich dient als grober Leitfaden, ob sich beide Prozesse im selben System befinden. Ist dies nicht der Fall, sollte man asynchrone Kommunikation wählen, weil man nicht abschätzen kann, wie viel Zeit der Prozess des Fremdsystem benötigt und somit das eigene System unterbrochen wird. Ist im Gegenzug der andere Prozess im selben System (schließt Subsysteme mit ein), kann man im Normalfall abschätzen wie lange der Prozess dauert. Meist wird hier dann auf eine synchrone Kommunikation gesetzt, wenn man weiß, dass der Prozess keine längere Zeit benötigt und der aufrufende Prozess diese Zeit warten kann.

Als weitere Kommunikationsart gibt es noch Caching, welches so funktioniert, dass die Daten beim ersten Aufruf initial in einen temporären Speicher (Cache) geladen werden. Das hat zur Folge, dass die Daten nicht immer neu geladen werden müssen, sondern aus dem Cache geladen werden können. Nach einer gewissen Zeit verfallen dann diese Daten, da man sonst nicht garantieren kann, dass diese immer auf dem aktuellen Stand sind. Natürlich kann man diese Art der Kommunikation nur nutzen, wenn auf Daten zugegriffen werden, die sich selten ändern, da man sonst die Daten über keinen längeren Zeitraum (>= 30 Min.) zwischen speichern könnte, ohne dass das andere System die Änderungen mit bekommt. Nutzbare Daten wären beispielsweise die Profildaten, da sich diese zwar ändern können, dieser Fall jedoch eher selten vorkommt bzw. der Zeitraum zwischen zwei Änderungen relativ groß ist (>=1 Tag).

Nachdem wir nun geklärt haben, wie die drei Kommunikationsarten funktionieren, schauen wir nun, wie die Umsetzung in dem Bahn 2.0 System aussehen würde.