Merchant Onboarding

Software Architektur Dokument

Andreas Heubeck

23. Oktober 2016

Inhaltsverzeichnis

1 Einführung und Ziele **4**
- 1.1 Aufgabenstellung . . . 4
- 1.2 Qualitätsziele . . . 4
- 1.3 Stakeholder . . . 4

2 Anforderungen **5**
- 2.1 Aktuelle Anforderungen . . . 5
  - 2.1.1 Funktionale Anforderungen . . . 5
  - 2.1.2 Nicht funktionale Anforderungen . . . 5
- 2.2 Neue Anforderungen . . . 6
  - 2.2.1 Funktionale Anforderungen . . . 6
  - 2.2.2 Nicht funktionale Anforderungen . . . 6

3 Randbedingungen **7**
- 3.1 Technische Randbedingungen . . . 7
- 3.2 Organisatorische Randbedingungen . . . 7
- 3.3 Konventionen . . . 7

4 Kontextabgrenzung **8**
- 4.1 Fachlicher Kontext . . . 8
- 4.2 Technischer Kontext . . . 8
- 4.3 Externe Schnittstellen . . . 8

5 Lösungsstrategie **9**

6 Bausteinsicht **10**

7 Laufzeitsicht **11**

8 Verteilungssicht **12**

9 Konzepte **13**
- 9.1 Fachliche Strukturen (Domäne) . . . 13
- 9.2 Typische Muster und Strukturen . . . 13
- 9.3 Ablaufsteuerung . . . 13
- 9.4 Ausnahme- und Fehlerbehandlung . . . 13
- 9.5 Batchverarbeitung . . . 14
- 9.6 Bedienoberfläche . . . 14
- 9.7 Ergonomie . . . 14
- 9.8 Geschäftsregeln . . . 14
- 9.9 Hochverfügbarkeit . . . 14

9.10 Internationalisierung 14
9.11 Kommunikation, Integration 14
9.12 Konfiguration 15
9.13 Logging, Protokollierung 15
9.14 Management und Administrierbarkeit 15
9.15 Migration 16
9.16 Parallelisierung / Nebenläufigkeit 16
9.17 Persistenz 16
9.18 Plausibilisierung und Validierung 17
9.19 Sessionbehandlung 17
9.20 Sicherheit 17
9.21 Skalierung / Clusterung 17
9.22 Transaktionsbehandlung 17
9.23 Verteilung 18

10 Entwurfsentscheidungen **19**

11 Qualitätsszenarien **20**
11.1 Bewertungsszenarien 20
11.2 Qualitätsbaum 20

12 Risiken **21**

1 Einführung und Ziele

1.1 Aufgabenstellung

Die die bestehende Merchant Onboarding Applikation soll eine neue Architektur entwickelt werden welche Continuous Deployment erlaubt.

1.2 Qualitätsziele

1.3 Stakeholder

Tabelle 1.1: Stakeholder

Name	**Beschreibung**
FA-1	todo
FA-2	todo
FA-3	todo
FA-4	todo

2 Anforderungen

2.1 Aktuelle Anforderungen

Da die Applikation bereits in Betrieb ist, sind die Anforderungen von den neuen getrennt.

2.1.1 Funktionale Anforderungen

Tabelle 2.1: Funktionale Anforderungen

ID	**Beschreibung**
FA-1	Die Applikation ermöglicht das Bestellen von Paymit für einen Geschäft.
FA-2	Es sollen mehrere Varianten auswählbar sein.
FA-3	todo
FA-4	todo

2.1.2 Nicht funktionale Anforderungen

Tabelle 2.2: Funktionale Anforderungen

ID	**Beschreibung**
FA-1	Einfach zu bedinender Bestellvorgang.
FA-2	Die Applikation muss mehrsprachig sein.
FA-3	todo
FA-4	todo

2.2 Neue Anforderungen

2.2.1 Funktionale Anforderungen

Tabelle 2.3: Funktionale Anforderungen

ID	**Beschreibung**
FA-1	Die einzelnen Teile der Applikation lassen sich separat von einander deploymen
FA-2	todo
FA-3	todo
FA-4	todo

2.2.2 Nicht funktionale Anforderungen

Tabelle 2.4: Nicht funktionale Anforderungen

ID	**Beschreibung**
FA-1	todo
FA-2	todo
FA-3	todo
FA-4	todo

3 Randbedingungen

3.1 Technische Randbedingungen

Docker, Java 8, Spring, AngularJS, Typescript

3.2 Organisatorische Randbedingungen

PCI DSS

3.3 Konventionen

4 Kontextabgrenzung

4.1 Fachlicher Kontext

4.2 Technischer Kontext

4.3 Externe Schnittstellen

5 Lösungsstrategie

6 Bausteinsicht

7 Laufzeitsicht

8 Verteilungssicht

9 Konzepte

9.1 Fachliche Strukturen (Domäne)

Fachliche Modelle, Domänenmodelle, Business-Modelle – sie alle beschreiben Strukturen der reinen Fachlichkeit, also ohne Bezug zur Implementierungs- oder Lösungstechnologie. Oftmals tauchen Teile solcher fachlichen Modelle an vielen Stellen in der Architektur, insbesondere der Bausteinsicht, wieder auf. Das "Domain-Driven-Designöder die uralte ëssentielle Systemanalyse"können Ihnen hierbei helfen.

Mit Absicht stellen wir diesen Abschnitt an den Anfang der übergreifenden Konzepte.

9.2 Typische Muster und Strukturen

Oftmals tauchen einige typische Lösungsstrukturen oder Grundmuster an mehren Stellen der Architektur auf. Beispiele dafür sind die Abhängigkeiten zwischen Persistenzschicht, Applikation sowie die Anbindung grafischer Oberflächen an die Fach- oder Domänenobjekte. Solche wiederkehrenden Strukturen beschreiben Sie möglichst nur ein einziges Mal, um Redundanzen zu vermeiden. Dieser Abschnitt erfüllt genau diesen Zweck.

9.3 Ablaufsteuerung

Ablaufsteuerung von IT-Systemen bezieht sich sowohl auf die an der (grafischen) Oberfläche sichtbaren Abläufe als auch auf die Steuerung der Hintergrundaktivitäten. Zur Ablaufsteuerung gehört daher unter anderem die Steuerung der Benutzungsoberfläche, die Workflow- oder Geschäftsprozessteuerung sowie Steuerung von Batchabläufen.

9.4 Ausnahme- und Fehlerbehandlung

Wie werden Programmfehler und Ausnahmen systematisch und konsistent behandelt? Wie kann das System nach einem Fehler wieder in einen konsistenten Zustand gelangen? Geschieht dies automatisch oder ist manueller Eingriff erforderlich? Dieser Aspekt hat mit Logging, Protokollierung und Tracing zu tun. Welche Art Ausnahmen und Fehler behandelt ihr System? Welche Art Ausnahmen werden an welche Außenschnittstelle weitergeleitet und welche Ausnahmen behandelt das System komplett intern? Wie nutzen Sie die Exception-Handling Mechanismen ihrer Programmiersprache? Verwenden Sie checked- oder unchecked-Exceptions?

9.5 Batchverarbeitung

Batchverarbeitung sequentielle Verarbeitung einer i.d.R. vorab festgelegten Menge an Daten oder Aufgaben.

9.6 Bedienoberfläche

IT-Systeme, die von (menschlichen) Benutzern interaktiv genutzt werden, benötigen eine Benutzungsoberfläche. Das können sowohl grafische als auch textuelle Oberflächen sein.

9.7 Ergonomie

Ergonomie von IT-Systemen bedeutet die Verbesserung (Optimierung) deren Benutzbarkeit aufgrund objektiver und subjektiver Faktoren. Im wesentlichen zählen zu ergonomischen Faktoren die Benutzungsoberfläche, die Reaktivität (gefühlte Performance) sowie die Verfügbarkeit und Robustheit eines Systems.

9.8 Geschäftsregeln

Wie behandeln Sie Geschäftslogik oder Geschäftsregeln? Implementieren die beteiligten Fachklassen ihre Logik selbst, oder liegt die Logik in der Verantwortung einer zentralen Komponente? Setzen Sie eine Regelmaschine (rule-engine) zur Interpretation von Geschäftsregeln ein (Produktionsregelsysteme, forward- oder backward-chaining)?

9.9 Hochverfügbarkeit

Wie erreichen Sie hohe Verfügbarkeit des Systems? Legen Sie Teile redundant aus? Verteilen Sie das System auf unterschiedliche Rechner oder Rechenzentren? Betreiben Sie Standby-Systeme? Könnte in Zusammenhang zu Clusterung stehen.

9.10 Internationalisierung

Unterstützung für den Einsatz von Systemen in unterschiedlichen Ländern, Anpassung der Systeme an länderspezifische Merkmale. Bei der Internationalisierung (aufgrund der 18 Buchstaben zwischen I und n des englischen Internationalisation auch i18n genannt) geht es neben der Übersetzung von Aus- oder EIngabetexten auch um verwendete Zeichensätze, Orientierung von Schriften am Bildschirm und andere (äußerliche) Aspekte.

9.11 Kommunikation, Integration

Kommunikation: Übertragung von Daten zwischen System-Komponenten. Bezieht sich auf Kommunikation innerhalb eines Prozesses oder Adressraumes, zwischen un-

terschiedlichen Prozessen oder auch zwischen unterschiedlichen Rechnersystemen. Integration: Einbindung bestehender Systeme (in einen neuen Kontext). Auch bekannt als: (Legacy) Wrapper, Gateway, Enterprise Application Integration (EAI).

9.12 Konfiguration

Die Flexibilität von IT-Systemem wird unter anderem durch ihre Konfigurierbarkeit beeinflusst, die Möglichkeit, manche Entscheidungen hinsichtlich der Systemnutzung erst spät zu treffen. Konfiguration kann zu folgenden Zeitpunkten erfolgen: Während der Programmierung: Dabei werden beispielsweise Server-, Datei- oder Verzeichnisnamen direkt ("hart") in den Programmcode aufgenommen. Während des Deployments oder der Installation: Hier werden Konfigurationsinformationen für eine bestimmte Installation angegeben, etwa der Installationspfad. Beim Systemstart: Hier werden Informationen vor oder beim Programmstart dynamisch gelesen. Während des Programmablaufs: Konfigurationsinformation wird zur Programmlaufzeit erfragt oder gelesen.

9.13 Logging, Protokollierung

Es gibt zwei Ausprägungen der Protokollierung, das Logging und das Tracing . Bei beiden werden Funktions- oder Methodenaufrufe in das Programm aufgenommen, die zur Laufzeit über den Status des Programms Auskunft geben. In der Praxis gibt es zwischen Logging und Tracing allerdings sehr wohl Unterschiede:

- Logging kann fachliche oder technische Protokollierung sein, oder eine beliebige Kombination von beidem.
- Fachliche Protokolle werden gewöhnlich anwenderspezifisch aufbereitet und übersetzt. Sie dienen Endbenutzern, Administratoren oder Betreibern von Softwaresystemen und liefern Informationen über die vom Programm abgewickelten Geschäftsprozesse.
- Technische Protokolle sind Informationen für Betreiber oder Entwickler. Sie dienen der Fehlersuche sowie der Systemoptimierung.
- Tracing soll Debugging -Information für Entwickler oder Supportmitarbeiter liefern. Es dient primär zur Fehlersuche und -analyse.

9.14 Management und Administrierbarkeit

Größere IT-Systeme laufen häufig in kontrollierten Ablaufumgebungen (Rechenzentren) unter der Kontrolle von Operatoren oder Administratoren ab. Diese Stakeholder benötigen einerseits spezifische Informationen über den Zustand der Programme zur Laufzeit, andererseits auch spezielle Eingriffs- oder Konfigurationsmöglichkeiten.

9.15 Migration

Für manche Systeme gibt es existierende Altsysteme, die durch die neuen Systeme abgelöst werden sollen. Denken Sie als Architekt rechtzeitig auch an alle organisatorischen und technischen Aspekte, die zur Einführung oder Migration der Architektur beachtet werden müssen. Beispiele:

- Konzept, Vorgehensweise oder Werkzeuge zur Datenübernahme und initialen Befüllung mit Daten
- Konzept zur Systemeinführung oder zeitweiliger Parallelbetrieb von Alt- und Neusystem

Müssen Sie bestehende Daten migrieren? Wie führen Sie die benötigten syntaktischen oder semantischern Transformationen durch?

9.16 Parallelisierung / Nebenläufigkeit

Programme können in parallelen Prozessen oder Threads ablaufen - was die Notwendigkeit von Synchronisationspunkten mit sich bringt. Die Grundlagen dieses Aspekten legt die Parallelverarbeitung. Für die Architektur und Implementierung nebenläufiger Systeme sind viele technische Detailaspekte zu berücksichtigen (Adressräume, Arten von Synchronisationsmechanismen (Guards, Wächter, Semaphore), Prozesse und Threads, Parallelität im Betriebssystem, Parallelität in virtuellen Maschinen und andere).

9.17 Persistenz

Persistenz (Dauerhaftigkeit, Beständigkeit) bedeutet, Daten aus dem (flüchtigen) Hauptspeicher auf ein beständiges Medium (und wieder zurück) zu bringen. Einige der Daten, die ein Software-System bearbeitet, müssen dauerhaft auf einem Speichermedium gespeichert oder von solchen Medien gelesen werden:

- Flüchtige Speichermedien (Hauptspeicher oder Cache) sind technisch nicht für dauerhafte Speicherung ausgelegt. Daten gehen verloren, wenn die entsprechende Hardware ausgeschaltet oder heruntergefahren wird.
- Die Menge der von kommerziellen Software-Systemen bearbeiteten Daten übersteigt üblicherweise die Kapazität des Hauptspeichers.
- Auf Festplatten, optischen Speichermedien oder Bändern sind oftmals große Mengen von Unternehmensdaten vorhanden, die eine beträchtliche Investition darstellen.

Persistenz ist ein technisch bedingtes Thema und trägt nichts zur eigentlichen Fachlichkeit eines Systems bei. Dennoch müssen Sie sich als Architekt mit dem Thema auseinander setzen, denn ein erheblicher Teil aller Software-Systeme benötigt einen effizienten Zugriff auf persistent gespeicherte Daten. Hierzu gehören praktisch sämtliche kommerziellen und viele technischen Systeme. Eingebettete Systeme (embedded systems) gehorchen jedoch oft anderen Regeln hinsichtlich ihrer Datenverwaltung.

9.18 Plausibilisierung und Validierung

Wo und wie plausibilisieren und validieren Sie (Eingabe-)daten, etwa Benutzereingaben?

9.19 Sessionbehandlung

Eine Session, auch genannt Sitzung, bezeichnet eine stehende Verbindung eines Clients mit einem Server. Den Zustand dieser Sitzung gilt es zu erhalten, was insbesondere bei der Nutzung zustandsloser Protokolle (etwa HTTP) wichtige Bedeutung hat. Sessionbehandlung stellt für Intra- und Internetsysteme eine kritische Herausforderung dar und beeinflusst häufig die Performance eines Systems.

9.20 Sicherheit

Die Sicherheit von IT-Systemen befasst sich mit Mechanismen zur Gewährleistung von Datensicherheit und Datenschutz sowie Verhinderung von Datenmissbrauch. Typische Fragestellungen sind:

- Wie können Daten auf dem Transport (beispielsweise über offene Netze wie das Internet) vor Missbrauch geschützt werden?
- Wie können Kommunikationspartner sich gegenseitig vertrauen?
- Wie können sich Kommunikationspartner eindeutig erkennen und vor falschen Kommunikationspartner schützen?
- Wie können Kommunikationspartner die Herkunft von Daten für sich beanspruchen (oder die Echtheit von Daten bestätigen)?

Das Thema IT-Sicherheit hat häufig Berührung zu juristischen Aspekten, teilweise sogar zu internationalem Recht.

9.21 Skalierung / Clusterung

Wie gestalten Sie Ihr System „wachstumsfähig“, so daß auch bei steigender Last oder steigenden Benutzerzahlen die Antwortzeiten und/oder Durchsatz erhalten bleiben?

9.22 Transaktionsbehandlung

Transaktionen sind Arbeitsschritte oder Abläufe, die entweder alle gemeinsam oder garnicht durchgeführt werden. Der Begriff stammt aus den Datenbanken - wichtiges Stichwort hier sind ACID-Transaktionen (atomar, consistent, isolated, durable). Im Bereich von NoSQL-Datenbanken gelten andere Kriterien.

9.23 Verteilung

Verteilung: Entwurf von Software-Systemen, deren Bestandteile auf unterschiedlichen und eventuell physikalisch getrennten Rechnersystemen ablaufen. Zur Verteilung gehören Dinge wie der Aufruf entfernter Methoden (remote procedure call, RPC), die Übertragung von Daten oder Dokumenten an verteilte Kommunikationspartner, die Wahl passender Interaktionsstile oder Nachrichtenaustauschmuster (etwa: synchron / asynchron, publish- subsribe, peer-to- peer).

10 Entwurfsentscheidungen

11 Qualitätsszenarien

11.1 Bewertungsszenarien

11.2 Qualitätsbaum

12 Risiken

Beispieleintrag für ein Glossarverweis: Architekturdokumentation.