

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik Event-gesteuerte Architektur im RESTful-API Kontext

1. Projektarbeit

Im Rahmen der Prüfung zum Bachelor of Science (B. Sc.)

4. September 2023

VerfasserIn:	Jona Rumberg
Kurs:	WWI22B5
Dualer Partner:	SAP SE, Walldorf
Betreuer der Ausbildungsfirma:	Steven Rösinger
Wissenschaftlicher BetreuerIn:	Prof. Dr. Thomas Freytag
Abgabedatum:	4. September 2023

Selbstständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende 1. Projektarbeit mit dem Thema:

Event-gesteuerte Architektur im RESTful-API Kontext

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, 31. August 2020, _____

Jona Rumberg

Inhaltsverzeichnis

Abkürzungsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation und Problemstellung

Die wohl umfassendste Entwicklung in der globalen **IT! (IT!)**-Branche ist heutzutage wohl die in Richtung Cloud. Um Geschäftsprozesse in diesem Umfeld umfassend abzubilden ist es immer wieder nötig verschiedene Dienste miteinander zu integrieren und so ist die Entwicklung sogenannter APIs zu einem Kernbestandteil der modernen Entwicklungstätigkeit geworden.

Ein weit verbreitetes Design Pattern ist dabei die sogenannte **REST! (REST!) API! (API!)**. In der Migration von ehemals sequenziell ablaufenden Prozessen kann es jedoch zu Hürden kommen, da eine **REST! API!** von Grund auf stateless angelegt sein sollte. Hier kommt Event basiertes Design ins Spiel, das dieses Problem lösen könnte.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist die Untersuchung vom Zusammenspiel des eventbasierten Ansatzes und dem **REST!ful** design pattern, wobei speziellen Wert auf die Betrachtung der Robustheit und Fehlerresillienz des entstehenden Systems gelegt werden soll. Zudem soll ein solches System im Kontext einer HR-Anwendung beispielhaft umgesetzt werden.

1.3 Betrieblicher Kontext

Die SAP SE ist ein deutsches Softwareunternehmen, das seit 1972 Unternehmenssoftware entwickelt. Heute beschäftigt es rund 105000 Mitarbeiter und hat Standorte weltweit. Die **ERP! (ERP!)** Systeme der Firma haben in der Geschäftswelt entscheidenden, branchenübergreifenden Einfluss. SAP bietet hierbei Möglichkeit, durch umfassende Funktionen und eine einheitliche, integrierte Datenbasis, Geschäftsprozesse zu überblicken, dieses digital abzuwickeln und zu automatisieren.¹

¹sapse_was.

1.4 Abgrenzung

1.5 Vorgehensweise und Aufbau der Arbeit

2 Theoretischer Hintergrund

2.1 EDA! (EDA!)

Fragen an diesen Abschnitt

- Was ist **EDA!**?
- Was ist ein Event?
- Warum ist **EDA!** relevant?

Ereignisorientierung als Architekturansatz

Zuerst einmal handelt es sich bei **EDA!**² um ein Konzept der Prozessmodellierung. Im Gegensatz zur gewöhnlichen Ablauf-orientierten Modellierung werden die Prozesse nicht als aufeinanderfolgende Schritte, sondern als Reaktionen auf Zustände konzeptioniert. Daraus resultiert, dass nicht mehr die prozedurale Abhandlung von Arbeitsschritten die zentrale Aufgabe in der Anwendungssystem-Entwicklung darstellt, sondern die Reaktion auf Ereignisse. Im Mittelpunkt von Architekturentscheidungen steht die Frage: "Was passiert, wenn dieses Ereignis eintritt?" und nicht mehr: "Welche Schritte müssen zur Erfüllung dieser Anforderung gegangen werden?". Was daraus resultiert, ist eine Architektur, die schon mit Beginn der Konzeption wesentlich agiler und robuster ist, da von Anfang an mit der Annahme gearbeitet wird, dass prinzipiell zu jedem Zeitpunkt jedes Ereignis eintreten kann.³ Ein Definitionsversuch für **EDA!** könnte also wie folgt lauten: Event-Driven Architecture bezeichnet einen Modellierungsansatz für ein verteiltes, asynchrones System, das verschiedene Komponenten durch eine zentrale Verarbeitung von Events verbindet.⁴

²Da sich bis jetzt keine allgemeingültige deutsche Übersetzung der Fachterminologie durchgesetzt hat, sollen in dieser Arbeit die englischen Begrifflichkeiten verwendet werden.

³**EDA2010.**

⁴**CLOUD2021.**

Technische Grundkonzepte der EDA!

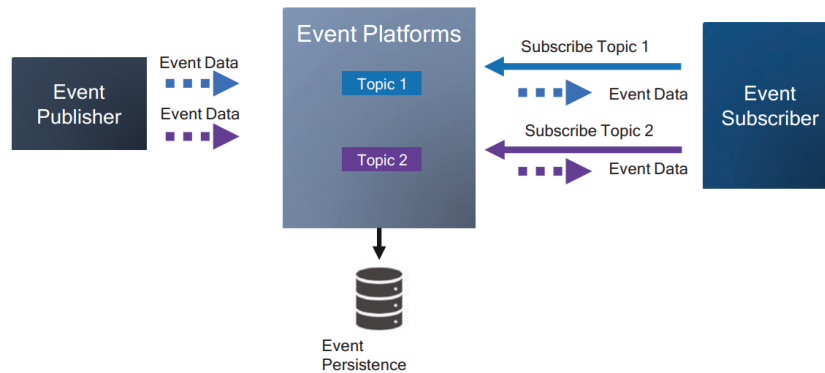


Abbildung 1: Komponenten der EDA!⁵

Die wichtigste Komponente eines durch **EDA!** modellierten Systems ist die zentrale Plattform zur Verarbeitung der Ereignisse in der Mitte der Architektur. Sie stellt die Infrastruktur bereit, um Events anzunehmen und diese weiterzugeben. Um einen Mehrwert aus dem System zu ziehen, muss sie darüber hinaus in der Lage sein, einen Kontext um Events herzustellen, d.h. sie in Verbindung mit anderen Ereignissen zu setzen, Ereignisse auf höheren Abstraktionsebenen zu erstellen und Ereignisse gegebenenfalls zu konsolidieren. Man spricht bei diesem Prozess von **CEP!** (**CEP!**).

Weitere Komponenten des **EDA!** sind Publisher und Subscriber.⁶ Sie sind explizit von außen an das System herangeschaltet, d.h. sie haben keine Kenntnis voneinander und können auch auf völlig unterschiedlichen Plattformen basieren. Das bringt den Vorteil, dass ein durch **EDA!** modelliertes System inhärent modular aufgebaut ist und so zum einen weniger anfällig für Totalausfälle ist, da die Komponenten unabhängig sind, und zum anderen prädestiniert für Integrationsvorhaben ist. Zu diesen grundlegenden Komponenten können im Zuge des **CEP!** noch einige weitere Konzepte hinzukommen. Die Abbildung zeigt beispielsweise eine Datenbank auf der Ereignisse persistent abgelegt werden können und die Einteilung von Ereignissen in Klassen, sogenannte Topics, die die Handhabung von verschiedenen Ereignisarten über ein System ermöglichen.

⁵[CLOUD2021]

⁶Für diese Komponenten finden sich in der Fachliteratur verschiedene Bezeichnungen. Außer Publisher und Subscriber findet man noch Producer und Receiver oder Producer und Listener.

Ereignisse

Zu klären bleibt die grundlegende Frage, was nun ein Ereignis ist. Die heute gängige Definition eines Ereignis im Kontext von **EDA!** ist, dass ein Ereignis eine "signifikante Änderung des Zustands" ist.⁷

2.2 RESTful API!

2.3 Technologie im Anwendungsbeispiel

2.4 Forschungsmethodik

2.5 Zusammenfassung des theoretischen Teils

⁷EDA2006.

3 Anwendung in der Praxis

3.1 Analyse des bestehenden Systems

3.2 Mögliche Anwendung der theoretischen Erkenntnisse

3.3 Implementierung eines Prototyps

4 Diskussion der Ergebnisse

4.1 Bewertung des Prototyps

4.2 Beurteilung von EDA und REST

4.3 Chancen der Technologie im betriebswirtschaftlichen Kontext

5 Resümee

5.1 Zusammenfassung der wichtigsten Ergebnisse

5.2 Handlungsempfehlung

5.3 Kritische Reflexion der Arbeit und Ausblick

Quellenverzeichnis

Anhang

1. Digitale Version der Arbeit
2. Interviews
 - 2.1. Expertmann 2018