Messaging Architecture for Integration of Customer Self-Services

Jonas Gann

29 September 2020

1 Context

As a result of the 2020 COVID-19 pandemic, customer self-service (CSS) technologies reached a new level of importance. Many shops were closed due to government measurements and customers had to use their web presences. CSS is a useful method to handle support for an increasing number of online customers. It enables searching for contract data, changing profile information, tracking shipment and much more without the need of human to human interaction. This reduces cost and increases customer performance [20].

2 Problem Statements

Most enterprise systems such as CSS need to be integrated into existing enterprise architectures (EA). This can be done via a business connector dedicated to providing an interface for CSS.

Resources and especially time are scarce. Integration therefore needs to be as simple and fast as possible and function with little maintenance and failure. The variety of heterogeneous landscapes and their constant development make this challenging. The integration has to interface with an EA without the need of changing it but also be able to adapt to future modifications.

The integration also has to perform according to requirements of both the EA and the CSS business connector.

As an integration does not only interface with systems but also delivers information and instructions, additional requirements relating to reliability like scalability and failure safety have to be considered.

3 Objectives

The bachelor thesis defines a messaging architecture for integration of a business connector, providing an interface for CSS, into enterprise architectures. This simplifies and speeds up integration by providing a solution reusable in many different scenarios.

A messaging approach enables deployment on different heterogeneous landscapes through loose coupling. Loose coupling allows integrated systems to not make assumptions about each other and use a send-and-forget approach, simplifying their integration.

Mechanisms increasing reliability of message communication through for example load balancing and store-and-forward are part of the messaging approach.

Through a real life example, it is validated, if the integration architecture satisfies requirements of the EA and business connector.

An operation manual presented by the thesis aims to further reduce deployment time and complexity and is used during the evaluation.

4 Planned Approaches

CSS scenarios get compiled from multiple resources in order to describe, what CSS looks like in practice. A scenario might be "New Address: User wants to change the address in his profile".

Enterprise architecture patterns (EAP) described by Perroud and Inversini [PI13] are used to abstract real life architectures. An EAP describes a common solution for reoccurring architectural problems in a generic way. It contains business processes, data objects and architecture bricks relevant for each pattern. Architecture bricks are in the context of EAPs "the smallest element that everything is built of" [PI13, Page 21]. They can e.g. be a web-server or a database. Implementations of architecture bricks are called solution bricks and can e.g. be the Apache web-server or PostgreSQL.

Business processes relevant for the CSS scenarios get selected from EAPs. A process is relevant if he is part of or related to a scenario and can be "Change Address: Login, Edit Profile, Change Address, Save Changes"

Architecture bricks and data objects contained by EAPs of selected business processes are evaluated in respect to their necessity for the process. They make up the enterprise architecture. Requirements of the EA regarding the consumption of CSS get analyzed. One requirement could be, that access to the "New Address" CSS scenario is required via a web server architecture brick. Another requirement might be, that data has to arrive in the correct order.

The **business connector** is constructed - based on CSS scenarios - as a black box of requirements regarding the access to EA systems and data objects. The black box also contains interfaces which provide functionalities for CSS scenarios. It is not included, how the business connector processes received data, triggers instructions to the EA or implements a service. The business connector could consist of the requirements: access to identities, access to identity validation, access to profile data and of the interface "changeIdentityProfileProperty"

An **integration architecture** gets defined which utilizes the selected business processes, architecture bricks and data objects in order to integrate the business connector. Purpose of the integration is to enable the EA and business connector to access each others functionalities according to their requirements. Requirements relating to the internal data and instruction delivery of the in-

tegration architecture like scalability and fail safety get considered in addition. Architecture bricks and data objects used in the final form of the integration architecture can be seen as its requirements towards an existing EA.

A **messaging approach** based on enterprise integration patterns described by Hohpe and Woolf [HW04] is used for the integration architecture. It gets documented as a component diagram containing architecture bricks, integration patterns and business connector along with communication channels and respective message layouts [HW04, cf. 16 ff.]. For each CSS scenario, relevant data and instruction flows inside the integration architecture are visualized as a sequence diagram.

Technologies which can be used for implementation get evaluated for the final form of the integration architecture. These can be different message oriented middlewares used for implementing messaging systems in general.

An **operation manual** with the purpose of guiding the requirement analysis in respect to existing architecture bricks and data objects of the EA and helping with theoretical and practical deployment of the integration architecture gets constructed. It might help finding architecture bricks required by the integration architecture inside the EA and instruct on how to implement the integration architecture in practice.

A **validation** of results is done through a real life example. For given EA, business processes, business connector and customer specific requirements, the integration architecture gets deployed (in theory) by usage of the operation manual. The quality of the integration architecture is measured by ease of deployment (e.g. used time) and satisfaction of requirements.

5 Milestones

The bachelor thesis is planned to be officially registered at the beginning of November and finished at the end of February. Parallel to the execution of described approaches, relating chapters of the bachelor thesis will be written.

1. CSS scenarios: October
2. Relevant business processes and patterns: October
3. Relevant architecture bricks and data objects: October / November
4. Business connector: October / November
5. Integration architecture: November / December
6. Operation manual: December / January
7. Validation: December / January
8. Finishing writing and presentation: February

References

[HW04] Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns*. 2004. ISBN: 0-321-20068-3.

[PI13] Thierry Perroud and Reto Inversini. *Enterprise Architecture Patterns*. 2013. ISBN: 978-3-642-37560-6.

[20] *Self-Services – gefragter denn je durch die Corona-Pandemie*. 2020. URL: `https://www.e-commerce-magazin.de/self-services-gefragter-denn-je-durch-die-corona-pandemie/`.