Messaging Architecture for Integration of Customer Self Service Systems Version 0.2

Jonas Gann

28 September 2020

1 Context

As a result of the 2020 pandemic, customer self-service (CSS) technologies reached a new level of importance [20]. Many shops were closed due to government measurements. Customers therefore had to use respective web presences. CSS is a useful method to handle support for a high number of customers.

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 CSS.

Resources and especially time are scarce. Integration therefore needs to be as simple and fast as possible and function with little support, failure and modification over time. 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 its modifications. Of course, the integration also has to perform according to requirements of both the EA and the CSS business connector, which the bachelor thesis evaluates. As an integration does not only interface with systems but also delivers information and instructions, additional requirements relating to reliability e.g. scalability, failure safety have to be considered.

3 Objectives and Planned Approaches

In order to simplify integration, the bachelor thesis defines a messaging architecture for integration of a CSS connector into enterprise architectures. The integration architecture simplifies integration by providing a integration solution reusable in many different scenarios. It uses a messaging approach to be applicable to heterogeneous landscapes through loosely coupling. Loose coupling allows integrated systems to not make assumptions about each other and use a send-and-forget approach, simplifying their integration. Messaging systems can also reliably handle communication through e.g load balancing and store-and-forward mechanisms. If the integration system satisfies requirements of the EA and business connector will be validated on a real life example. An operation manual presented by the thesis aims to further reduce deployment time and complexity and is used in the evaluation.

The following describes the planned approaches:

In order to describe, what CSS is in practice, the bachelor thesis compiles CSS scenarios from multiple resources. A scenario might e.g. be: *New Address: User has a new home address and wants to change it in his profile.*

Enterprise architecture patterns (EAP) described by [PI13] are used as model for 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.

For each CSS scenario, relevant business processes of the EAPs get selected. A pattern might already provide business processes which are part of or related to a CSS scenario. This can e.g. be: *Change Address: Login, Edit Profile, Change Address, Save Changes*

EAPs, which contain the selected business processes, are evaluated in respect to relevant architecture bricks and data objects, which will make up the enterprise architecture. The requirements of the EA regarding the consumption of CSS services will be analyzed. One requirement could e.g. be, that access to the *New Address* CSS scenario is required via a web server architecture brick.

CSS scenarios will be used to construct the business connector as a black box of requirements regarding access to EA systems and data objects. The black box will also contain interfaces, which provide functionalities for CSS scenarios. It will not be included, how the business connector processes the received data, triggers instructions to the EA or implements a service. The business connector could e.g. 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 like e.g. scalability and fail safety will be considered. Architecture bricks and data objects used in the final form of the integration architecture can be seen as its requirements towards the existing EA.

The integration architecture takes a message based approach by using enterprise integration patterns described by [HW04]. The integration architecture 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.

With the integration architecture finished, technologies which can be used for implementation get evaluated. These can e.g. be different message oriented middlewares used for implementing messaging systems in general.

An operation manual guides the requirement analysis in respect to existing architecture bricks and data objects of the EA and helps with theoretical and practical deployment of the integration architecture. It might e.g. help finding architecture bricks required by the integration architecture inside the EA and instruct on how to implement the integration architecture in practice.

The results of the bachelor thesis are validated through application on customer examples. For given EA, business processes and customer specific requirements, the integration architecture will be 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 customer requirements.

4 Related Work

- Architecture Patterns described in the book by [PI13]
- Integration Patterns described in the book by [HW04]

5 Milestones

1. Compilation CSS scenarios: October
2. Relevant business processes and patterns: October / November
3. Relevant architecture bricks and data objects: October / November
4. Integration architecture: November / December
5. Operation manual: December / January
6. Validation: December / January

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/.