

Messaging Architecture for Integration of Customer Self Service Systems

Version 0.1

Jonas Gann

25 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 has to be as simple and fast as possible. Simple means e.g. that existing systems do not need to be modified for integration, that integration systems do not differ fundamentally between heterogenous landscapes or, that support for deployment of the integration solution is available.

Of course, besides being simple and fast, integration also has to satisfy requirements of the connector and provide its CSS functionalities towards the EA. Typical integration challenges have to be addressed: Networks are slow and unreliable, applications have different interfacing requirements or are not built to be interfaced at all and possible changes of applications have to be considered.

3 Objectives and Planned Approaches

The bachelor thesis defines a messaging architecture for integration of CSS systems into existing enterprise architectures.

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 EAPs get selected. Patterns might already provide business processes which are part of or related to CSS scenarios. 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.

An integration architecture gets defined, which utilizes the selected CSS scenarios, business processes, architecture bricks and data objects in order to enable the EA and business connector to access each others functionalities. The connector is a system providing an interface to functionalities with which the selected CSS scenarios can be realized. Requirements of the EA and business connector will be analyzed for the selected CSS scenarios, architecture bricks and data objects.

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 connector is a piece of software providing an interface to services with which CSS scenarios can be realized in existing EAs. The connector has requirements towards the integration system in regard to access to the EA. The concrete requirements will be analyzed for previously selected CSS scenarios, architecture bricks and data objects. As the purpose of a connector is integration, it is assumed, that the connector can satisfy any requirements of an integration system.

The integration architecture takes a message based approach by using enterprise integration patterns described by [HW04]. This approach aims to solve the previously described challenges in the "Problem Statements" section.

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 business process, relevant data and instruction flows inside the integration architecture are visualized as a sequence diagram.

An operation manual, amongst other things, 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.

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 Realted Work

- Business Connecor: The business connector of IDAS providing CSS functionalities
- 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 – gefragt denn je durch die Corona-Pandemie*. 2020. URL: <https://www.e-commerce-magazin.de/self-services-gefragter-denn-je-durch-die-corona-pandemie/>.