

Professor Mihaela MUNTEAN, PhD

E-mail: mihaela.muntean@e-uvt.ro

Department of Business Information Systems

West University of Timisoara

Associate Professor Claudiu BRÂNDĂȘ, PhD

E-mail: claudiu.brandas@e-uvt.ro

Department of Business Information Systems

West University of Timisoara

Associate Professor Marian Pompiliu CRISTESCU, PhD

E-mail: marian.cristescu@ulbsibiu.ro

Department of Finance and Accounting

“Lucian Blaga” University of Sibiu

Diana MATIU

E-mail: matiu_diana@yahoo.com

SAP Romania

IMPROVING CLOUD INTEGRATION USING DESIGN SCIENCE RESEARCH

***Abstract.** Cloud integration offers new opportunities to streamline the business processes in organizations. Variate integration approaches are possible: Cloud-to-Cloud, Cloud-Integrator-Cloud or Cloud-Cloud-Cloud. Depending on the cloud service, we identify various cloud service models, namely Anything as a Service (XaaS). We propose an activity framework for deploying a Cloud-Integrator-Cloud (CIC) integration of two Software as a Service (SaaS) systems. The demarche has an achievement in the implementation of two SAP Cloud software. The software integration is subject of a SCRUM process management approach. We consider the integration framework and the integration process implementation artefacts, outputs of a software project developed in terms of the Design Science Research (DSR) process. Based on relevant literature references on DSR, we apply DSR principles to the CIC integration problem. The DSR process steps are mapped to the SCRUM sprints introduced in the integration project.*

***Keywords:** Design Science Research, Software Integration, Cloud Service Model.*

JEL Classification: L20, L21, L86, M15

1. Introduction

Design Science Research (DSR), according to the reference syllabi „Design Research in Information Systems” (see Van der Merwe) addresses two key issues: „the central role of the information system/technology artifact

and the importance of professional relevance of information system research”. Information system design is performed with respect to the business strategy and the global organizational design (Figure 1).

March and Smith (1995) identify two design processes and four design artifacts produced by Design Science Research. Whether or not the artifact is perpetuated depends on the way it is designed, on the interface with the external environment and the adaptation of one with the other, given that the implementation of the artifact within organizations causes disruptions in the business environment (see Leoz, 2018).

The artefacts can be models, frameworks, methods (see Hevner, 2004). Guidelines for design science in information systems proposed in Hevner’s research are: design as an artefact; problem relevance; design evaluation; research contribution; research rigor; design as a search process; communication of research.

According to Vaishnavi, Kuechler and Petter (2019) „Design Science Research is a set of synthetic and analytical techniques performing research in information systems”. It involves the creation of an artefact, e.g. algorithm, human/computer interface, and system design methodology or language. Any DSR approach is a process containing the following phases (Figure 2) (see Takeda, 1990).

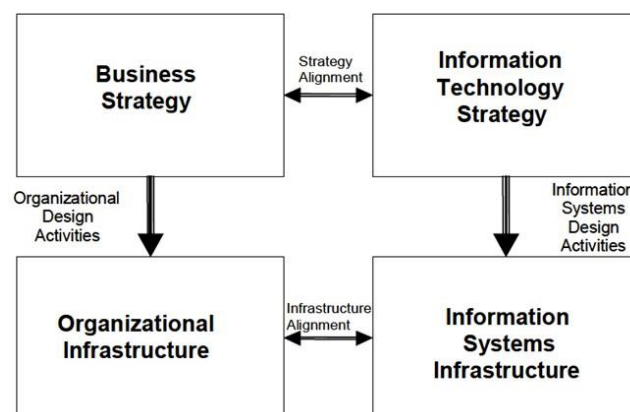


Figure 1. Organizational Design and Information System Design (Hevner 2004)

The conclusion, in form of a knowledge contribution, brings improvements in solving the problem (Beck, 2013).

In the paper “Understanding Design Research. Toward an Achievement of Balance” Owen states "knowledge is generated and accumulated through action". The general model involves undertaking something and evaluating the results obtained. According to this model, the

process is presented as "a cycle in which knowledge is used, creatively, to obtain things, they are evaluated and thus knowledge is obtained".

An innovative proposal, e.g. model, framework, architecture, methods, design principle or theory, improves the effectiveness and efficiency of the information systems/technology adopted by the business environment.

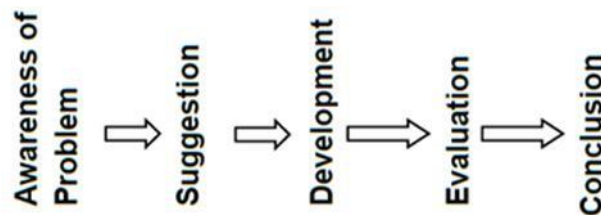


Figure 2. DSR process (Vaishnavi, 2019)

Information system, viewed as an artifact, reflects the relationships between its components (modules, interfaces, frameworks), which makes some behaviors to be supported while others are subject to constraints.

Models are sets of propositions or statements expressing relationships between constructs. Frameworks establishes real or conceptual guides to serve as support or guide. In this paper, we introduce a framework for Cloud-Integrator-Cloud (CIC) integration. Implementation aspects make referees to SAP Hybris Commerce and SAP Cloud for Customers (SAP C4C). The study case, developed as a DSR process, follows the transfer of an insurance policy from SAP Hybris Commerce to SAP C4C.

Architectures are high-level structures of systems. The proposed integration framework and the concrete implementation can determinate the architecture of a Software as a Service (SaaS) solution for integrating SAP Hybris Commerce with SAP Cloud for Customer.

In accordance with DSR principles, the process of designing, realizing and implementing information systems (IS) can be linked to the methods and techniques used in architecture and construction. Thus, an IS must be:

- functional - with a high level of reliability, without errors and defects that prevent the users to use their capabilities;
- useful - it must be developed based on the users' requirements and perfectly adapted to them;
- effective - to increase user satisfaction and give them a pleasant experience in the actual use process.

Further, design principles are core principles and concepts to guide design, while design theories are prescriptive set of statements on how to do something to achieve a certain objective (Vaishnavi, 2019).

2. Theoretical and practical perspective of cloud integration

2.1 Cloud service models

Cloud integration implies bringing together applications and systems in a unitary cloud framework. Data and cloud services are available to anyone who has access to the integrated system. The as-a- service (“aaS”) model, in the context of cloud computing, is a term that indicates a service, which is available over the Internet via the cloud (Doug, 2019). Based on the service the cloud model is offering we identify various cloud service models, namely Anything as a Service (XaaS).

Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) are the three main categories of cloud computing (Figure 3). IaaS offers access to high performant computers, networks and storage devices as needed. PaaS provides a platform for software development and offers a variety of services for development, testing and deployment. SaaS is a cloud application service and utilize the internet to deliver applications to its users (Palos-Sanches, 2017; Chody, 2018).

Cloud-computing adoption by organizations increases rapidly due to ascertained advantages (Columbus, 2018; Gil, 2019). It implies product/service innovations, process innovations, organizational innovations, and technological innovations. Cross-functional processes that are spread out across several applications need the integration of many XaaS services. Data managed by a specific XaaS services, e.g. SaaS service, is required in another XaaS/SaaS services in order to optimize the business processes.

Software as a service (SaaS)	Communications as a Service (CaaS)	Desktop as a Service (DaaS)
Platform as a service (PaaS)	Monitoring as a Service (MaaS)	Backup as a Service (BaaS)
Infrastructure as a service (IaaS)	Network as a Service (NaaS)	Hardware as a Service (HaaS)
	Disaster Recovery as a Service (DRaaS)	Metal as a Service (MAAS)
	Business Continuity as a Service (BCaaS)	Unified Communications as a service (UCaaS)
	Storage as a Service (SaaS)	Recovery as a Service (RaaS)
	SIEM as a Service (SIEMaaS)	Archive as a Service (AaaS)
	Encryption as a Service (EaaS)	FlexPod as a Service (FPaaS)
	BPM as a Service (BPMaaS)	Email as a Service (EaaS)
	VDI as a Service (VDIaaS)	Enterprise as a Service (EaaS)
	Backup as a Service mBaaS)	Process as a Service (PraaS)
		Communications Platform as a Service (CPaaS)

Figure 3. Cloud service models. Anything as a Service

(source: <https://www.orosk.com/cloud-computing/>)

For supporting Business-to-Business (B2B) and Business-to-Consumer (B2C) processes, we call for variate integration approaches: Cloud-to-Cloud, Cloud-Integrator-Cloud or Cloud-Cloud-Cloud (Cloud-Cubed) (Wall Street Journal, 2013).

CLOUD-TO-CLOUD INTEGRATION

As companies embrace more SaaS applications, the need to integrate the SaaS applications will increase. Various integration techniques use web services (Table 1) to implement the integration process (Linthicum, 1999; Kolluru, 2013). Web services are Extensible Markup Language (XML) - centered data exchange systems that use the Internet for Application-to-Application (A2A) communication and integration (Wang, 2014), (Linthicum 1999), (Muntean, 2019). A comprehensive overview upon Simple Object Access Protocol (SOAP), REST (Representational State Transfer) and ODATA (Open Data Protocol) services reflect the differences between them (Muntean, 2019).

Table 1. Web services overview (Linthicum, 1999)

Web service	Characteristics
SOAP	A service architecture; XML based; Runs on HTTP but envelopes the message; Slower than REST; Very mature, a lot of functionality; Not suitable for browser-based clients
REST	A service architecture (resource-oriented); Uses the HTTP headers to hold meta-information (although it is protocol-agnostic); Can be used with XML, JSON or whatever necessary; Usually used with JSON due to the easily parsable content; Faster than SOAP; It uses semantic media types
OData	Protocol; Based on HTTP, AtomPub and JSON; RESTful by design

An important issue in the cloud environment is to integrate and ensure the quality of transferred and processed data (Muntean, 2019). The integration framework based on the OData service proposed by Muntean and her colleagues in 2019 conducted to a Cloud-to-Cloud integration of two SAP cloud software, SAP Cloud for Customer and S/4 HANA Cloud.

CLOUD-INTEGRATOR-CLOUD INTEGRATION

The integration platform (Figure 4) receives data from one cloud/SaaS application, convert it into the format required by the receiver

cloud/SaaS application, and send the data by invoking the web services exposed by the receiving application (Wall Street Journal, 2013).



Figure 4. Cloud-Integrator-Cloud integration

Due to the extended integration capabilities, the platform integrates data and processes in Application-to-Application (A2A) and Business-to-Business scenarios. SAP Cloud Platform Integration (SAP CPI) is representative for our demarche. It is a cloud-based integration platform for integrating cloud applications with other SAP and non-SAP cloud and on premise solutions. SAP CPI supports SOAP over HTTP and HTTPS protocols. The connectivity adapters allow the implementation of the integration process via web services, e.g. OData service (SAP Help Portal).

CLOUD-CLOUD-CLOUD INTEGRATION

This type of integration is similar to the cloud-integrator-cloud approach, except that the integration system is also a cloud-based application, e.g. Oracle Integration Cloud Service or others Integration Platform as Service (iPaaS) solutions.

2.2 Cloud-Integrator-Cloud integration

The integration platform sustains the integration process, all the necessary tasks to transfer the data send to the receiver application (Figure 5).

Activity A4 defines the main integration flow. Additionally, security aspects are settled in A5 and error handling is introduced in A6. Finally, the data is copied into the receiver application according to the previously defined sender-receiver mapping (activity A7).

Digital businesses are highly integrated and innovative. Integration platforms are commonly used for integrating different applications and services. In a Cloud-Integrator-Cloud (CIC) integration framework, the integrator receives the data from one cloud application, convert it into the format required by the other application and send it by invoking web services (Mittal, 2017).

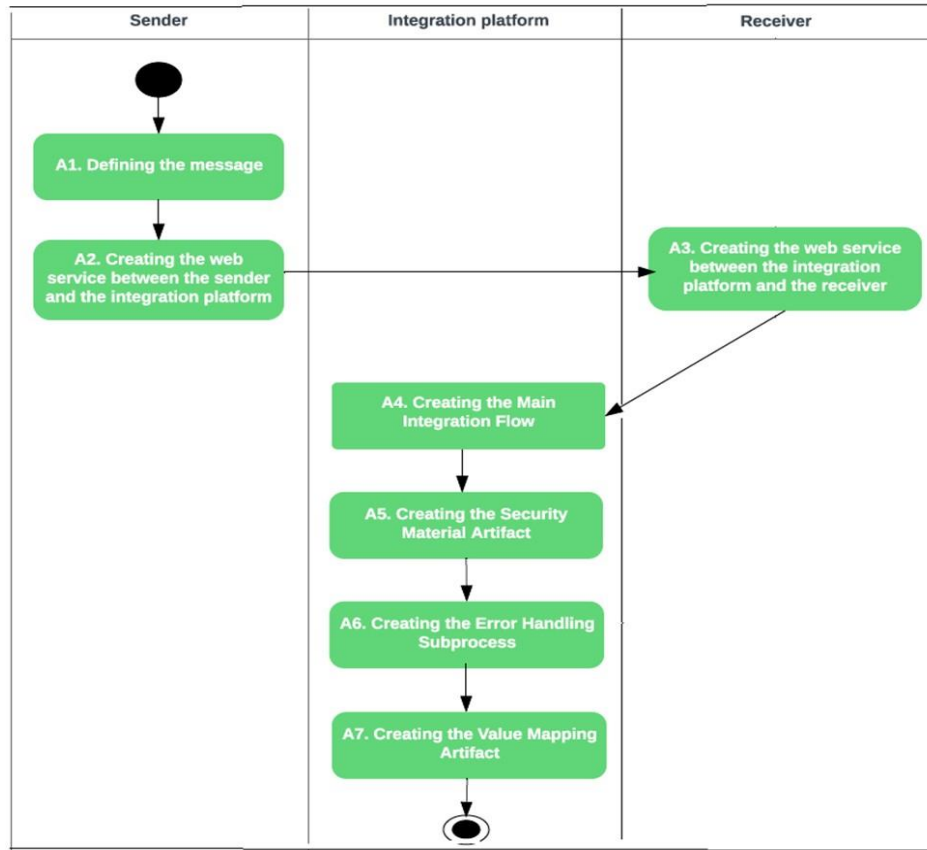


Figure 5. Framework for Cloud-Integrator-Cloud integration (CIC integration)

2.3 CIC integration in the cloud environment

SAP Cloud Platform Integration (SAP CPI) suite allows the integration of processes, data and devices by using different methods for consuming cloud services, open Application Programming Interfaces (APIs) and tools (Techtarget,2018), (Vaishnavi,2004).

There can be a variety of business use cases (Sharma, 2018), but all of them start from the business and customer perspective: replication of customer profile from SAP Hybris Commerce to SAP C4C; directly access of a customer from SAP Hybris Commerce to SAP C4C in order to create a service ticket; directly access from SAP C4C to SAP Hybris Commerce and deliver support to the customer (Pillai, 2015).

Integration projects in the cloud imply various agile approaches, like Crystal methodologies, Extreme Programming, Feature-Driven Programming,

Kanban and SCRUM (Hart, 2010).

An agile approach in the SAP environment implies the following steps (Hevner,2015): 1 - to understand what agile means and what the chosen methodology, e.g. SCRUM; 2 - to identify the stakeholders, e.g. exec management, business owners, development and testing teams, for implying them into the project; 3 – to build a business case, e.g. for a CIC integration with SAP CPI; 4 – to “start small” meaning to start with one team; 5 – to reorganize teams and establish how they are structured and communicate; in SAP projects usually work many small, multifunctional teams; in SCRUM approaches a business representative is the product owner/representative; 6 – to create a prioritized backlog; 7 – to organize the sprint meetings; 8 – to choose some tools to support the agile development process, e.g. Jira, Rally.

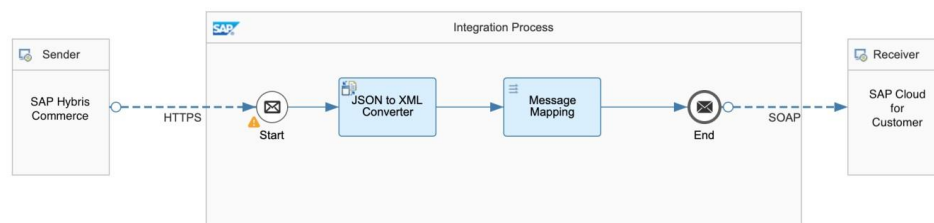


Figure 6. CIC integration with SAP CPI (SAP, 2019)

Based on the integration framework introduced in paragraph II B (Figure 5), the backlog for implementing the considered CIC integration is the following (Figure 7).

Activities A1-A7, strictly related to technology, imply directly the sender, receiver and the integration platform. Additionally, two activities, A0 and A8, complete the development approach (Figure 7).

The deployment of A0 takes into consideration the IT players, but is not executed by neither the sender, receiver nor the integration platform. Activity A8 takes place independently from SAP CPI, SAP Hybris Commerce and SAP Cloud for Customer.

According to SCRUM methodology, the nine activities have story points assigned directly proportional to their complexity. Analyzing the overall project, we propose five sprints for deploying the integration of SAP Hybris Commerce with SAP C4C.

Improving Cloud Integration Using Design Science Research

Sprint No	Activity ID	Activity	Task ID	Task	Story Points
0	A0	Awareness of the Problem and Tentative Design			15
1	A1	Defining the message	T11	Sprint Planning	13
			T12	Defining the body of the message	
			T13	Defining the headers of the message	
	A2	Creating the web service between SAP Hybris Commerce and SAP CPI	T21	Defining the operation of the web service	7
			T22	Configuring the authentication method in SAP CPI	
			T23	Configuring the integration flow endpoint	
2	A3	Creating the web service between SAP CPI and SAP C4C	T31	Creating the Communication System	15
			T32	Configuring the Communication Scenario	
			T33	Creating and configuring the Communication Arrangement	
			T34	Configuring the authentication method in SAP C4C	
			T35	Defining the endpoint	
3	A4	Creating the Main Integration Flow	T41	Defining and configuring the Sender System	20
			T42	Configuring the Adapter of the Sender System	
			T43	Configuring the Converter	
			T44	Defining the Content Modifier for the message headers	
			T45	Configuring the Routers of the message	
			T46	Creating the Message Mappings	
			T47	Defining and configuring the Receiver System	
			T48	Configuring the Adapter of the Receiver System	
4	A5	Creating the Security Material Artifact			3
	A6	Creating the Error Handling Subprocess	T61	Defining the Content Modifier for message headers	8
			T62	Catching the error message	
			T63	Converting and modifying the error message	
	A7	Creating the Value Mapping Artifact			4
	A8	Disseminate the results			5

Figure 7. CIC integration. Backlog

3. Research strategy. Applying the DSR process to CIC integration

The scientific approach of the methods and techniques of research design in the field of IS, implies the division of the research effort in two directions: on the one hand, the focus is on the set of activities that configure a process, e.g. the cloud integration process, and on the other hand the focus should be on the product, viewed as an artifact, a result of the process, e.g. the integration scenario. The DSR process uses past knowledge in its running and can contribute to enrich the knowledge base in a given area. The proposed integration scenario, its development and implementation represents a best practice artefact that can be further used for similar projects in the application domain (Figure 8).

Due to the increased use of SAP cloud solutions and the necessity to integrate these systems in order to support business processes that span multiple organizational areas, the research was oriented on SAP cloud integration application domain.

After investigating the application domain from theoretical and practical point of view and evaluating some ongoing integration projects, we have considered SAP Hybris Commerce and SAP Cloud Customer the pillars of our integration framework.

According to the DSR process, the integration process was modeled, each stage/step is further explicitly outlined in the presentation of the research.

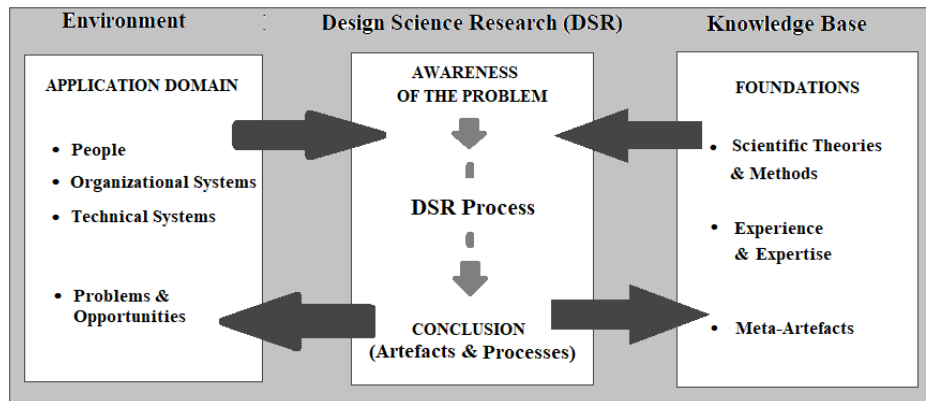


Figure 8. The DSR process and its connectivity with the environment and the knowledge base (Kapinga, Montero, Mbise, 2019)

AWARENESS OF THE PROBLEM

In our paper, the starting point, in this stage, is the detailed analysis of the two systems (SAP Hybris Commerce and SAP C4C), which are the object of the integration process, and the integration platform (SAP Cloud Platform Integration).

The CIC integration of the two systems implies the transfer of an insurance policy from SAP Hybris Commerce to SAP C4C. The integration process is part of the following business process:

- (1) Step 1: A business consultant fills in a form with the client personal data. The client assist the consultant during this step.
- (2) Step 2: The insurance policy, containing the client data, is generated.
- (3) Step 3: The client gives his consent and signs the insurance policy.
- (4) Step 4: The consultant activates the policy generation button and the data is automatically replicated into SAP Cloud for Customer software.
- (5) Step 5: SAP Cloud for Customer maintains the client data and the insurance policy in pdf format.

SUGGESTION

The suggestion is a creative stage in which the aim is to obtain a new functionality, of the integration framework, based on a new or existing configuration in which new or existing applications and services appear. The objective of this phase consists in a proposal for the integration process in order to support the business process (Figure 9).

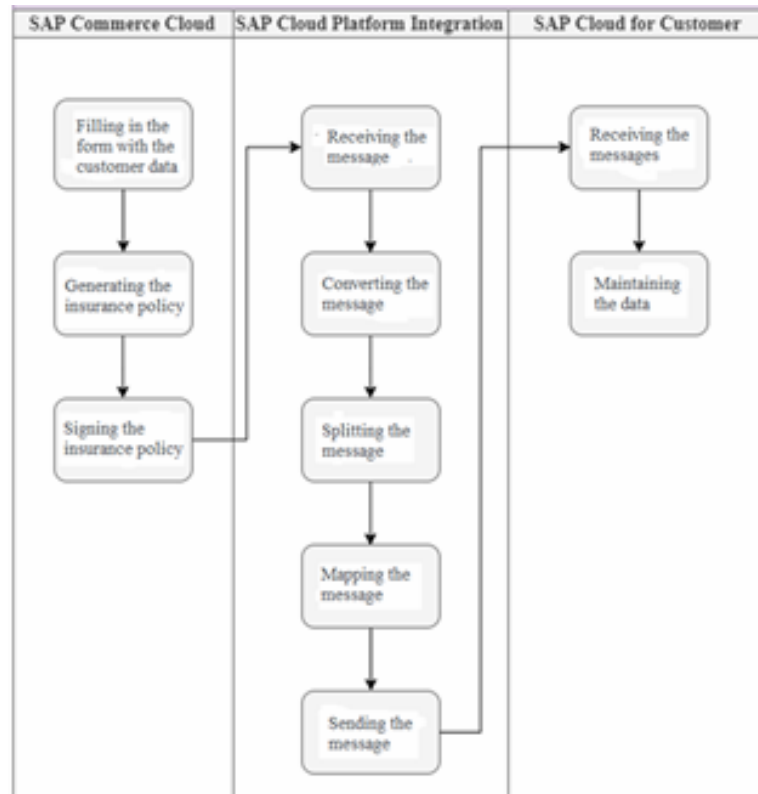


Figure 9. Integration scenario

SAP CPI represents the suggested integration alternative, despite a directly Cloud-to-Cloud variant.

DEVELOPMENT

The constructions and the specific approach of the CIC reveal the vocabulary and symbols necessary to define the specific SAP problems and to find the appropriate solutions to the integration process. All these aspects have a major impact at this stage, especially on the way the tasks are designed and the problems to be solved are dealt with by the members of the integration team.

The tentative design of the integration process (Figure 9) is further developed according to the general CIC framework and the project backlog is established (Figure 10). After establishing all activities (A1 – A7) and the individual tasks (T11 – T63), sprint planning is necessary.

This means to define the deliveries for each sprint and how that is achieved. Sprint planning implies the collaboration with the whole SCRUM team (Figure 10).

Figure 10. Sprint planning

The message contains the client data, assurance policy data and a pdf attachment with the policy. In the receiver, the data is stored in the client table, policy table and in the attachments table. Therefore, two communications scenario establish the transfer: one for the client data and another for the policy data and the .pdf file.

The basic characteristic of this stage, according to the basic principles of DSR is the critical approach of the activities undertaken. Sprints are evaluated, as IT artifacts, in the business environment, in an organization, within which the requirements and criteria based on which they will be evaluated are established. The basic characteristic of this stage, according to the basic principles of DSR is the critical approach of the activities undertaken. Sprints are evaluated, as IT artifacts, in the business environment, in an organization, within which the requirements and criteria based on which they will be

evaluated are established.

This stage results, in many cases, in identifying the weaknesses of the sprint (viewed as an artifact) and implicitly improving it through refining and reevaluation processes. The sprint evaluation process involves the prior definition of the corresponding values and the collection and analysis of the appropriate data from the development stage. Sprints can be evaluated in terms of functionality, completeness, coherence, accuracy, performance, reliability, usefulness, how they fit with the organization where the development and implementation process takes place, to which other features can be added.

In this process, the aim is to accumulate the necessary and sufficient knowledge regarding the different evaluation techniques, how the questions are asked and their correctness, the method of choosing the most suitable technique, which will give confidence in the results obtained from the evaluation, as a proof of the quality and performance obtained in the sprint design process.

According to SCRUM methodology, each sprint is an iteratively mini-project. Beyond the final evaluation, a proper passed evaluation of each sprint output determinates the moving to the next sprint.

The final test, after sprint 4, implies the sending of different messages, including wrong messages for testing the error handling process. Evaluation means to compare the message (data) received by SAP C4C with the message (data) send by SAP Hybris Commerce.

SAP CPI allows you to view the message details, like properties, content, header, format and message pass (Figure 12).

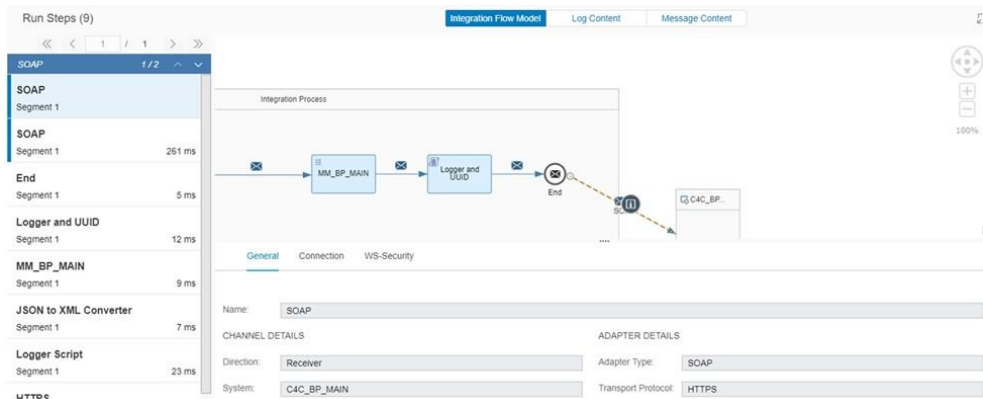


Figure 12. Evaluation. Message pass

CONCLUSION

SAP CPI is a mature integration platform, open and flexible with different methods for consuming cloud services, open Application Programming Interfaces (APIs) and software developer kits.

The DSR process based on an agile SCRUM demarche describes a framework for successful CIC integration processes. Exemplified on a study case in the SAP environment, the approach is applicable to any CIC integration.

4. Discussion

Currently, companies must be able to sustain, monitor and control all kind of business processes that are distributed across different locally installed and/or cloud-based systems. Nowadays, Enterprise Application Integration models in large business environments, including partners, customers and suppliers imply the integration of different Enterprise Resource Planning solutions with other Enterprise systems.

Due to the evolution of cloud computing technologies, the variety of SaaS solutions and other XaaS services, and the advantages of the cloud systems, more and more businesses are moving to the cloud. The widespread business processes imply real-time communication, data accessibility from any location. Therefore, cloud integration approaches are actual research priorities. Any successful implementation enriches the best practices portfolio. With these general considerations, we have introduced three approaches: Cloud-to-Cloud, Cloud-Integrator-Cloud and Cloud-Cloud-Cloud integration. The proposed CIC integration framework in Figure 5 can be applied to any concrete integration project in a similar way as the conducted implementation in the SAP cloud environment.

DSR in information systems has as an objective the creation of an artefact to solve real-life, business problems (Hart, 2010), (Hevner, 2015). According to Hart, the artefact represents a proper solution of a “heretofore unsolved and important business problem”.

References on DSR identified in literature (see Hevner, 2004) consist of a combination of general design and development phases, like identification, design, development and testing. Vaishnavi et al. introduced in 2004 a general DSR process model (Figure 2). Other DSR process models are subject of a synthetic analysis in Van der Merwe’s paper that is referenced at the end. Each process phase has an established output (Table 2). The symbiotic approach of DSR with SCRUM methodology conducts to an agile integration process (Table 3).

Table 2. DSR process model

PHASE	OUTPUT
Awareness of the problem	Approach proposal: STEP 1 - STEP 5 for CIC integration
Suggestion	Tentative design: Integration scenario (Figure 5)
Development	Artefact: Integration framework (Figure 1); Integration process modeled by SAP CPI (Figure 7)
Evaluation	Performance measures: Evaluation of the message pass (Figure 8)
Conclusion	Results: Integration implementation - data transfer from sender (SAP Hybris Commerce) to receiver (SAP C4C). Communication of the results

Table 3. DSR – SCRUM

DSR STEP	SCRUM - ACTIVITY	SCRUM - SPRINT
STEP 1. Awareness of the Problem	A0 Awareness of the problem and Tentative Design	Sprint 0
STEP 2. Suggestion		
STEP 3. Development	A1 Defining the message	Sprint 1
	A2 Creating the web service between SAP Hybris Commerce and SAP CPI	
	A3 Creating the web service between SAP CPI and SAP C4C	Sprint 2
	A4 Creating the Main Integration Flow	Sprint 3
	A5 Creating the Security Material Artefact	Sprint 4
	A6 Creating the Error Handling Sub-process	
	A7 Creating the Value Mapping Artefact	
STEP 4. Evaluation*		
STEP 5. Conclusion	A8 Communicating the result	Sprint 4

*evaluation after each sprint and a final evaluation

Design science research is a problem-solving methodology for developing socio-technical artefacts. The DSR process steps facilitate the integration in the cloud for any CIC integration problem:

- Awareness of the problem: the identified problem is refined. A CIC integration of two SAP Cloud software will be of interest to more than one entity (company).
- Suggestion: the “artefact design” is proposed. Starting from a specific problem (integration of SAP Hybris Commerce with SAP C4C) the integration approach is generalized in order to be relevant to more entities.
- Development: a practice-based design grounds the development of the integration problem. The SCRUM paradigm brings innovation to the artefact, in fact the integration framework and further the integration process.
- Evaluation: a rigorous evaluation of the artefact is essential, e.g. completeness, consistency, accuracy.
- Conclusion: the problem and its importance, the artefact, its utility and novelty, the rigor of its design and its effectiveness are subject of the communication.

4. Conclusion

Integration projects of cloud applications manage to streamline the business processes in organizations. The involved systems, SaaS services are integrated. Data introduced in one system is used in the other one.

Integration approaches are based on best practices in the field. Beyond these, research initiatives are necessary to improve the integration framework and to establish general applicable principles. Design Science Research is used in IT/information system projects to solve real-time problems. DSR is an iterative methodology consisting of five steps: awareness of the problem, suggestion, development, evaluation and conclusion. Design science theories and research practices bring improvement to the integration context.

SCRUM methodology, an agile project management methodology, delivers the outcomes “after the most valuable work has been completed” by implying all stakeholders. The DSR-SCRUM framework proposed for CIC integration can be extrapolated, with adaption to the context, to software projects in general.

REFERENCES

- [1] Chody, J. (2018). *IaaS, PaaS and SaaS: The Cloud Comparison Guide*, [available at: <https://community.hpe.com/t5/Transforming-IT/IaaS-PaaS-and-SaaS-The-Cloud-Comparison-Guide-2018-Update/ba-p/7017543#Xd1l2ugzbIV>];

- [2] Columbus, L. (2018), *State of Enterprise Cloud Computing*, [available at: <https://www.forbes.com/sites/louiscolumbus/2018/08/30/state-of-enterprise-cloud-computing-2018/>];
- [3] Doug, M. F. (2019), *What is XaaS*, [available at: <https://stonebridgeinnovations.com/what-is-xaas/>];
- [4] Gil, D., Johnsson, M., Mora, H., Szymanski, J. (2019), *Review of the Complexity of Managing Big Data of the Internet of Things*; Complexity – Special Issue on Advances in Architectures, Big Data, and Machine Learning Techniques for Complex Internet of Things Systems, <https://doi.org/10.1155/2019/4592902>;
- [5] Owen, C. (1997), *Understanding Design Research. Toward an Achievement of Balance*; *Journal of the Japanese Society for the Science of Design* 5(2): 36–45;
- [6] Pillai, K. (2015), *SAP C4C and Hybris Commerce (Integration B2C Use Cases)*, available at: <https://sapcrm7.wordpress.com/2015/11/01/sap-c4c-and-hybris-commerce-integration-use-cases/>;
- [7] Sharma, P. (2018), *SAP C4C and SAP Commerce – Integration Highlights*, available at: <https://blogs.sap.com/2018/12/02/sap-c4c-and-sap-commerce-integration-highlights/>;
- [8] Beck, R., Weber, S. Gregory, R.W. (2013), *Theory-Generating Design Science Research*; Hawaii International Conference on System Sciences (HICSS), available at: https://www.researchgate.net/publication/251237390_Theory-Generating_Design_Science_Research;
- [9] Linthicum, D.S. (1999), *Enterprise Application Integration*; Addison-Wesley Professional: Boston, MA, USA, ISBN 0201615835;
- [10] Hart, D.N., S. D. Gregor, S.D. – editors. (2010), *Information Systems Foundation: The Role of Design Science*, E – Press, ISBN: 9781921666353 (eBook);
- [11] Hevner, A.R., S. Chatterjee, S. (2015), *Design Science Research in Information Systems*, in: Association for Information Systems. Reference Syllabi, ed.: J. vom Brocke, Eduglopedia.org, available at: http://eduglopedia.org/reference-yllabus/AIS_Reference_Syllabus_Design_Science_Research_in_IS.pdf;
- [12] Hevner, A.R. et. all (2014), *Design Science in Information Systems Research*, *MIS Quarterly*, vol. 28, No. 1;
- [13] Kapinga, A.F., Montero, C.S., Mbise, E.R. (2019), *Mobile Marketing Application for Entrepreneurship Development: Code sign with Women Entrepreneurs in Iringa, Tanzania*, available at: <https://onlinelibrary.wiley.com/doi/full/10.1002/isd2.12073>;
- [14] Kolluru, N.V.S., N. Mantha, N. (2013), *Cloud Integration — Strategy to Connect Applications to Cloud*; Annual IEEE India Conference (INDICON), available at: <https://ieeexplore.ieee.org/document/6726096>;

-
- [15] Leoz, G., S. Petter, S. (2018), *Considering the Social Impacts of Artefacts in Information Systems Design Science Research*; *European Journal of Information Systems* 27(2): 154170;
 - [16] March, S.T., G. Smith, G. (1995), *Design and Natural Science Research on Information Technology*; *Decision Support Systems*, 15(4):251-266;
 - [17] Mittal, S., Negi, N., Chauhan, R. (2017), *Integration of Edge Computing with Cloud Computing*; International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), Publisher: IEEE,1-6;
 - [18] Muntean, M., Brandas, C., Cirstea, T. (2019), *Framework for a Symmetric Integration Approach*; *Symmetry*, 11(2), 2019, available at: <https://www.mdpi.com/2073-8994/11/2/224>;
 - [19] Takeda, H., Veerkamp, P., Tomiyama, T., Yoshikawam, H. (1990), *Modelling Design Processes*. *AI Magazine* Winter: 37–48;
 - [20] Palos-Sanches, P. R. (2017), *Cloud Computing (SaaS) Adoption as a Strategic Technology: Results of an Empirical Study*; *Mobile Information Systems*, <https://doi.org/10.1155/2017/2536040>;
 - [21] Uldall-Espersen, T., Frøkjær, E., Blanford, A. (2007), *Increasing the Impact of Usability Work in Software Development*; Proceedings of the Conference on Human Factor in Computing Systems (CHI2007), San Hose, California, USA, available at: https://www.researchgate.net/publication/221514531_Increasing_the_impact_of_usability_work_in_software_development;
 - [22] Van der Merwe, A., Gerber, A., Schmuts, H. (2017), *Guidelines for Conducting Design Science Research in Information Systems*, available at: http://www.altavandermerwe.com/resources/Pre_print%20Draft.pdf;
 - [23] Vaishnavi, V., Kuechler, B., Petter, S. (2019), *Design Science Research in Information Systems*, available at: <http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf>;
 - [24] Vaishnavi, V., B. Kuechler, B., S. Petter, S. (2004), *Design Science Research in Information Systems*, (created in 2004 and updated until 2015 by Vaishnavi, V. and Kuechler, W.); last updated (by Vaishnavi, V. and Petter, S.), June 30, 2019. URL: <http://www.desrist.org/design-research-in-information-systems/>;
 - [25] Wang, S.H., Chen, D.A. (2014), *Cloud Framework for Enterprise Application Integration on SOA*; Proceedings of the International Conference on Software Intelligence Technologies and Applications & International Conference of the International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things, Hsinchu, Taiwan.