

Framework for a Symmetric Integration Approach

Mihaela Muntean ^{1,*} , Claudiu Brândaş ¹  and Tanita Cîrstea ²

¹ Business Information Systems Department, Faculty of Economics and Business Administration, West University of Timisoara, 30023 Timisoara, Romania; claudiu.brandas@e-uvvt.ro

² SAP Romania, 300011 Timisoara, Romania; tanita.cirstea@gmail.com

* Correspondence: mihaela.muntean@e-uvvt.ro; Tel.: +40-742-089-058

Received: 26 December 2018; Accepted: 5 February 2019; Published: 14 February 2019



Abstract: An Application-to-Application integration framework in the cloud environment is proposed. The methodological demarche is developed using a data symmetry approach. Implementation aspects of integration considered the Open Data Protocol (OData) service as an integrator. An important issue in the cloud environment is to integrate and ensure the quality of transferred and processed data. An efficient way of ensuring the completeness and integrity of data transferred between different applications and systems is the symmetry of data integration. With these considerations, the integration of SAP Hybris Cloud for Customer with S/4 HANA Cloud was implemented.

Keywords: application integration; data integration; data symmetry; Open Data Protocol (OData) service

1. Introduction

Ensuring the quality of data integration in cloud-based hybrid systems and applications is an important issue both in business and in data science.

Integration frameworks are focused on data integration or application integration. Data integration implies the transfer, replication, and transformation of data from one place to another (from one application to another)—ignoring application or business logic. Application integration is defined as linking systems together at the application/business logic level. According to Tibbetts (2011) “neither method of integration is superior or inferior to the other—it simply depends on the need” [1]. Application integration in the business environment has grounded the widely used Enterprise Application Integration (EAI) paradigm, which is sustained by dedicated systems, e.g., Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Supply Chain Management (SCM) systems, or other enterprise-integrated applications. However, various integration projects in organizations nowadays employ both types of integration frameworks.

According to Linthicum (1999), EAI conducts the strategic utilization of company data and technology for greater efficiency and profit [2]. Four types of integration are identified: Data-level, application interface-level, method-level, and user interface-level—the last three together are supported by the business model level [2]. The EAI architecture has been transposed into the cloud and is supported by cloud computing infrastructure and services [3,4]. Cloud applications integration is facilitated by various classes of Application Programming Interfaces (APIs), predefined or developed, such as application, data service, or data set APIs.

Cloud services provide access to server infrastructure and resources that are managed by a provider, e.g., Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). According to Deloitte practitioners [5], to integrate cloud applications, the following integration approaches are applicable: Cloud-to-Cloud; Cloud-Integrator-Cloud; and Cloud-Cloud-Cloud (Cloud-Cubed) integration.

A primary issue in integration processes is the level of data quality. Corrales (2018) synthesized the diverse approaches regarding data quality issues [6]. An integration approach based on symmetry ensures the completeness and integrity of data transferred between different applications and systems. Weyl (1983) presented the importance of symmetry in science and other domains [7]. Murtagh (2009) underlined the role of symmetry in Data Analysis and Data Mining [8]. In the context of the widespread use of cloud-based hybrid applications and the importance of symmetry in the quality of the data integration process, we considered it necessary to develop a framework for the symmetric integration of applications.

2. Framework for a Symmetric Integration Approach

The research problem of our study is the development and implementation of a framework for the symmetric integration of applications and systems in cloud environments in order to ensure the completeness and integrity of data in the integration process. Our research is quantitative and experimental. For development and validation, we used SAP Hybris Cloud for Customer (C4C), S/4 HANA Cloud (S/4), and Open Data Protocol (OData) service.

Various approaches reference application integration (AI) topics [1–3]. Relevant for our demarche is the following consideration regarding AI, which is defined as a process of “keeping redundant copies of data (in independently designed applications) consistent, or enabling end-users to access data and functionality from independently designed applications on a single user interface” [9]. In Figure 1, two independently designed applications are involved in an integration process. Data are retrieved from Application 1 and stored in Application 2 in a corresponding format. Making an analogy with a symmetry model using (d) as its symmetry axis, the “copy of data” object is symmetric to the “data source” object.

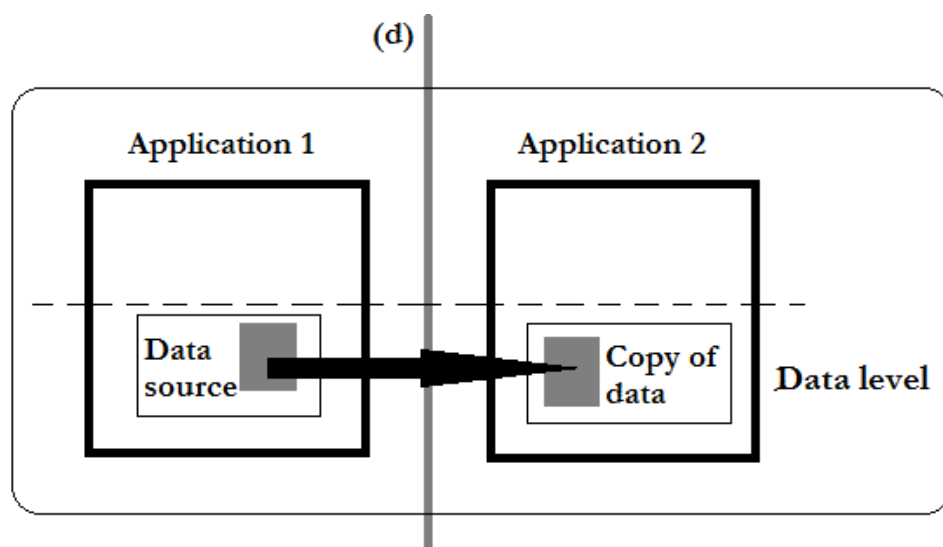


Figure 1. Application integration.

In an EAI framework, ERP systems are put together with CRM systems, SCM systems, or other Enterprise Applications (EA) systems. The integration approach presented in Figure 1 is applicable to the EAI context and is valid for both data structures and Business Objects (BO) [10]. A collection of technologies and services support the integration of the systems, building a “middleware integration framework” [10].

The increasing use of cloud technologies creates integration challenges for many companies. A scenario for integrating SAP Hybris Cloud for Customer (C4C) and S/4HANA Cloud (S/4) is the subject of this article (Figure 2).

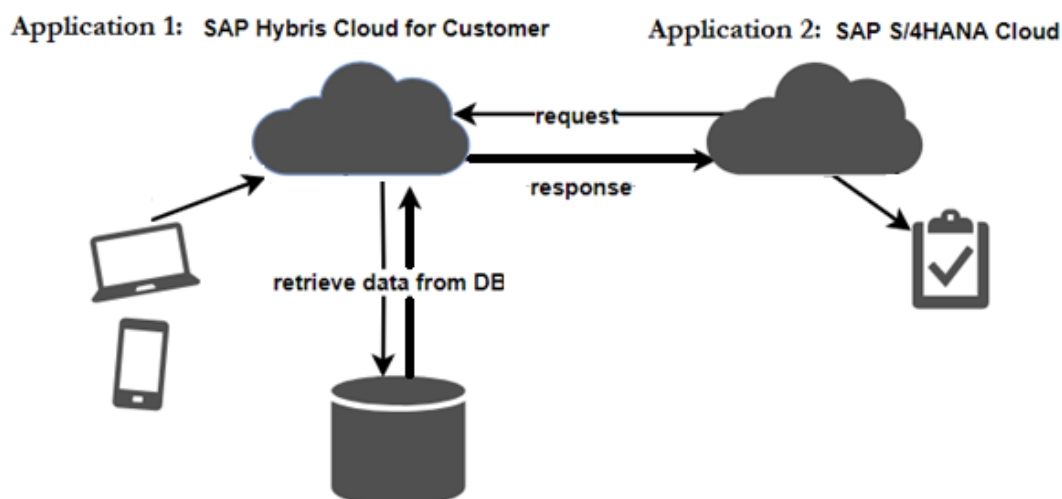


Figure 2. Integration framework for SAP Hybris Cloud for Customer (C4C)–S/4HANA Cloud (S/4) [7].

Best practices in integration methods point out three alternatives: Application-to-X users (A2X), using a SOAP-based web service in synchronous execution; Application-to-Application (A2A), using a SOAP-based web service in asynchronous execution; and Open Data Protocol using a REST-based web service in synchronous execution [11–13]. The OData service, metaphorically named the “SQL of the web”, allows the creation and consumption of queryable and interoperable RESTful APIs in a simple and standard way. The service enables the client, e.g., Application 2, to publish and manipulate the resource identified by URIs and defined in a data model using simple HTTP messages. In an OData-based approach, Application 1 is the OData service producer and will expose its service along with metadata that contain the semantics for consumption. The OData service exploits common formats like XML and JSON for communication. Application 2 understands the OData service using generic tools and can combine information from multiple data sources.

The scenario for integrating C4C with S/4 includes the actions seen in Table 1, on both the OData producer (C4C application) and consumer (S/4 application) sides.

Table 1. C4C–S/4 integration activities.

Application	Action ID	Action	Result
C4C	A1	Identifying the Business Object (BO), that will represent the data source for the OData service	BO (data source)
C4C	A2	Creating the OData service for reading the BO	OData service
C4C	A3	Activating the OData service	URL
C4C	A4	Testing the OData service	Displaying OData service answer
S/4	A5	Consuming the OData service	BO in S/4 (symmetric copy of BO)

A time report from the C4C system is transferred and stored in the S/4 system. A Time Report BO is a collection of time entries, which can be defined for a date or a date range. Employees can record time for different activities/tasks in which they are involved, e.g., work, travel, administration, vacation [14,15]. An employee can also have many time reports. Time recording enables managers to keep track of which employees are doing day-to-day work, which in turn helps to improve productivity and customer satisfaction, reduce cost, and enable employees to be more productive by making it easier for them to record their time.

3. Implementation Aspects of SAP Hybris Cloud for Customer and S/4 HANA Cloud Integration

Action A1. Identifying the Business Object that Will Represent the Data Source for the OData Service

C4C software maintains the time reports for the employees. Operations like creating, updating, and deleting time reports; retrieving time entries; retrieving employee data; and grouping time entries for an employee for a specified time are possible.

Time reports are defined based on a standard Time Report BO [14,15]. Therefore, they are instances of the BO. The report header contains information, e.g., the Universally Unique Identifier (UUID) of the time report, the reference to the standard BO, and the employee's user ID. A time report can record time for a certain ticket or independent of any ticket. Solving a ticket implies the completion of different tasks by the employee to whom it was assigned. The time spent performing the necessary tasks is recorded in time entries. A time entry has the following attributes: Time type (work, travel, administration, vacation, etc.), ticket ID, start time, end time, break start, break end, duration, and time zone (Table 2). After creating the time report as an instance of the Time Report BO, it has to be approved by the reporting line manager. Once the time is recorded, it has to be billed. Ticket IDs are integrated with S/4 for billing processes. During the proposed integration process, each time entry is copied into the S/4 system and becomes a time item of the ticket for billing processes.

Table 2. Details of a time report: 2018/CW23 example.

Time Type	Date	Ticket ID	Start Time	End Time	Break Start	Break End	Duration
Work	06/04/2018	6227-T1028	9:00 AM	5:30 PM	12:00 AM	12:30 AM	8 h
Work	06/05/2018	6227-T1028	9:00 AM	5:30 PM	1:00 PM	1:30 PM	8 h
Work	06/06/2018	6227-T1028	9:00 AM	5:30 PM	12:30 AM	1:00 PM	8 h
Work	06/07/2018	6227-T1028	9:00 AM	5:30 PM	12:30 AM	1:00 PM	8 h
Work	06/08/2018	6227-T1028	9:00 AM	5:30 PM	12:00 AM	12:30 AM	8 h

A time report covering week 23, from 4 June 2018 to 10 June 2018, for employee 'tanita business' created in C4C can be seen in Table 2 and is transferred into the S/4 system for further processing. The replication of the 2018/CW23 time report into S/4 is an example of data symmetry within the integration approach.

Action A2. Creating the OData Service for Reading the BO

The OData service contains the metadata of the entities exposed by the service, the relationships between these entities, and system and custom query options for retrieving data [11]. The service can be created in an assisted way. The appropriate selection can then be made in the OData editor window according to requirements.

Action A3. Activating the OData Service

After creating the OData service for time report 2018/CW23, the service is activated and its URL is accessible. The metadata structure can be checked by using the service URI [14].

Action A4. Testing the OData Service

The created OData service is tested through HTTP requests launched on the C4C test console (Figure 3). The service response can be viewed in different formats, e.g., Form, XML, or JSON [15].

Entry Details		Test		
Request				
Method:		GET	https://my327692.crm.ondemand.com/sap/c4c/odata/cust/v1/timereports_odata/TimeReportCollection?\$filter=ID eq '22'	
Response				
Data	XML	JSON	Headers	
ID		22		
ETag		2018-06-12T21:33:19...		
ObjectID		00163E10F3201EE89...		
EmployeeUUID		00163E27-EAAD-1EE...		
StartDate		2018-06-04T00:00:00		
Status		1		
TotalDuration		PT8H30M		
ReportName		2018/CW23		
EndDate		2018-06-08T00:00:00		

Figure 3. Testing the OData service: GET method.

Security issues are also tested and service performance, e.g., response time, is analyzed.

Action A5. Consuming the OData Service

Several tasks are necessary to implement action A5, as seen in Table 3. In order to respect the application integration framework presented in Figure 1, task T3 was proposed. It was designed based on several sub-tasks introduced in Table 3 and is implemented through ABAP programming.

Table 3. Activity A5. Tasks for consuming the OData service.

Application	Task ID	Task
S/4	T1	Establishing endpoint for the time report into the S/4 system (fields in S/4: timereport-id, timereport-reportname, timereport-startdate, timereport-enddate, timereport-totalduration)
S/4	T2	Creating communication system, defining communication scenario, and establishing communication arrangement Effective consuming of the OData service:
		<ul style="list-style-type: none"> - T31: Check if outbound service exists in the communication scenario; - T32: Create HTTP client to access the outbound service; - T33: Create the service request by specifying the request method and header;
S/4	T3	<ul style="list-style-type: none"> - T34: Send the request to C4C and get a response; - T35: Parse the response; - T36: Map response fields coming from C4C to endpoint fields in S/4; - T37: Display the time report in S/4.

A communication system was created in S/4 HANA [14,16] to integrate SAP Hybris Cloud for Customer and S/4 HANA Cloud, a process for “keeping copies of data (in independently designed applications) consistent”, as mentioned at the beginning of Section 3. In the communication scenario, the OData service URL was specified, as seen in Figure 4. Further integration aspects, e.g., authentication methods, were established by the communication arrangement [16].

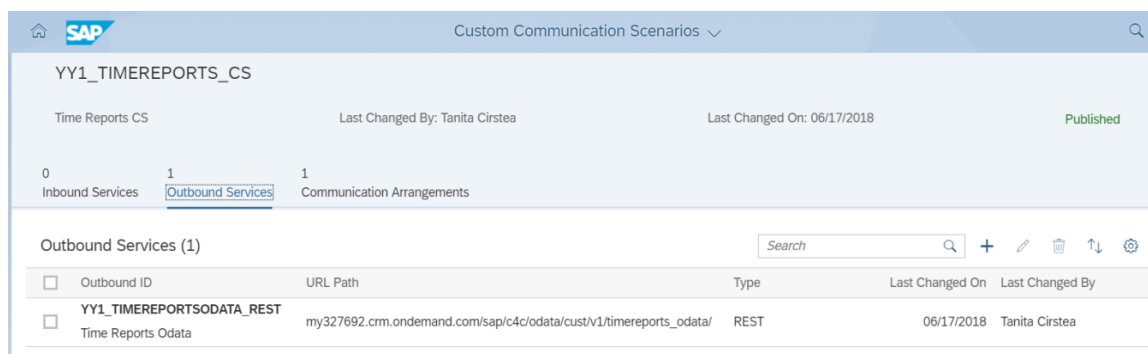


Figure 4. Communication scenario.

After receiving the response from the OData service (task T34), in order to obtain a “symmetric” copy of the time report in the S/4 system, the tasks T35–T37 were proposed and developed.

Implementation aspects, including coding, of the effective consuming of the OData service (activity A5, task T3) proposed by the authors are presented in Tables 4 and 5.

Table 4. Activity A5, task T3. Implementation aspects.

Task ID	Description	Code
T31	Check if outbound service exists in the communication scenario (Figure 4).	<pre>CHECK cl_ble_http_client => is_service_available(communication_scenario = 'YY1_TIMEREPORTS_CS' outbound_service = 'YY1_TIMEREPORTS_CS_REST') = abap_true.</pre>
T32	Create HTTP client to access the outbound service.	<pre>DATA(lo_client) = cl_ble_http_client => create (communication_scenario = 'YY1_TIMEREPORTS_CS' outbound_service = 'YY1_TIMEREPORTS_CS_REST').</pre>
Similar to the testing approach of the Odata service (Action A4), the GET method and the JSON format for viewing the answer were chosen (Figure 3).		
T33	Create the service request by specifying the request method and header.	<pre>DATA(request) = cl_ble_client_request => create() -> set_method('GET') -> set_content_type('application/json') -> set_header_parameter(name = 'Accept' value = 'Application/json').</pre>
T34	Send the request to C4C and get a response.	<pre>TRY . DATA(response) = lo_client->send(request). DATA(lv_body) = response->get_body(). CATCH cx_ble_http_exception INTO DATA(lx_Response). DATA(error_message) = 'Connection to C4C failed with HTTP errorcode ' + lx_Response->status_code. EXIT. ENDTRY.</pre>

Table 5. Activity A5, tasks T35–T37. Implementation aspects.

Task/Description	Code
	DATA: id TYPE string, reportname TYPE string, startdate TYPE string, enddate TYPE string, totalduration_v TYPE string, lv_rest TYPE string, lv_pair TYPE string, lv_key TYPE string, lv_value TYPE string.
T35: Parse the response	
Transformations of the response string stored in <i>lv_body</i> :	
<ul style="list-style-type: none"> - All occurrences of substring "" are eliminated; - Iteratively, each "line" is identified (the different "lines" are separated by commas) and stored in <i>lv_pair</i>; the next "lines" remain in <i>lv_body</i>; - In <i>lv_pair</i>, we have <i>lv_key</i> (ID, ReportName, StartDate, EndDate, TotalDuration) and <i>lv_value</i>; they are separated by substring ':'; - All occurrences of substring '{' and '}' are eliminated; - For each <i>lv_key</i>, the <i>lv_value</i> is extracted and stored into a corresponding variable. 	<pre> REPLACE ALL OCCURRENCES OF SUBSTRING "" IN lv_body WITH ". SPLIT lv_body AT ':' INTO lv_pair lv_body. WHILE NOT (lv_body EQ lv_pair). SPLIT lv_pair AT ':' INTO lv_key lv_value. REPLACE ALL OCCURRENCES OF SUBSTRING '{' IN lv_value WITH ". REPLACE ALL OCCURRENCES OF SUBSTRING '}' IN lv_value WITH ". CONDENSE lv_value. IF lv_key EQ 'ID'. id = lv_value. IF lv_key EQ 'ReportName'. reportname = lv_value. IF lv_key EQ 'StartDate'. startdate = lv_value. IF lv_key EQ 'EndDate'. enddate = lv_value. IF lv_key EQ 'TotalDuration'. totalduration_v = lv_value. SPLIT lv_body AT ':' INTO lv_pair lv_body. ENDWHILE. </pre>
T36: Map response fields coming from C4C to endpoint fields in S/4	<pre> timereport-id = id. timereport-reportname = reportname. timereport-startdate = startdate. timereport-enddate = enddate. timereport-totalduration = totalduration_v. </pre>
T37: Display the time report in S/4 HANA system	

The OData service response in JSON format, as seen in Figure 5, is the subject of a parsing process (task T35) in order to identify the data that are copied into the symmetric data fields (task T36) of the S/4 system. The parsing algorithm describes a direct parse of the JSON string and transforms the data into an internal table of key-value pairs (Table 5). The implementation is based on an iterative WHILE-ENDWHILE approach.

The response string is stored in *lv_body* and is applied to the following transformations.

The time report can be displayed, and is involved in further processing according to the business flows.

In conclusion, the integration approach was conducted in terms of a data symmetry approach as seen in Figure 1, and was implemented by OData service as seen in Figure 2, a parsing process of the service response, and the mapping of the response fields coming from SAP C4C to endpoint fields in S/4 (Tables 1, 3 and 5, Figure 6).

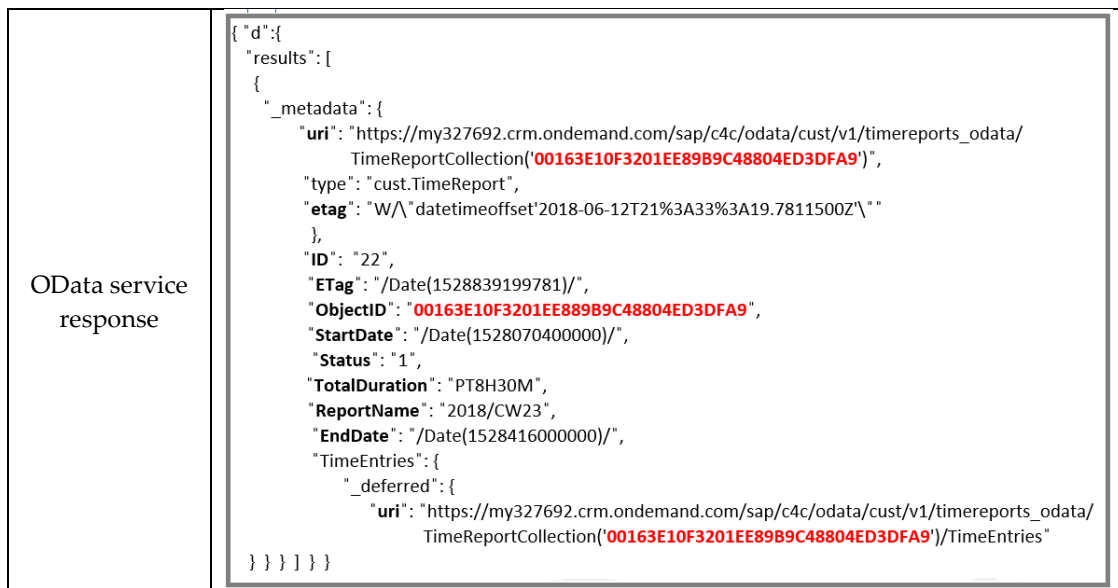


Figure 5. OData service response in JSON format.

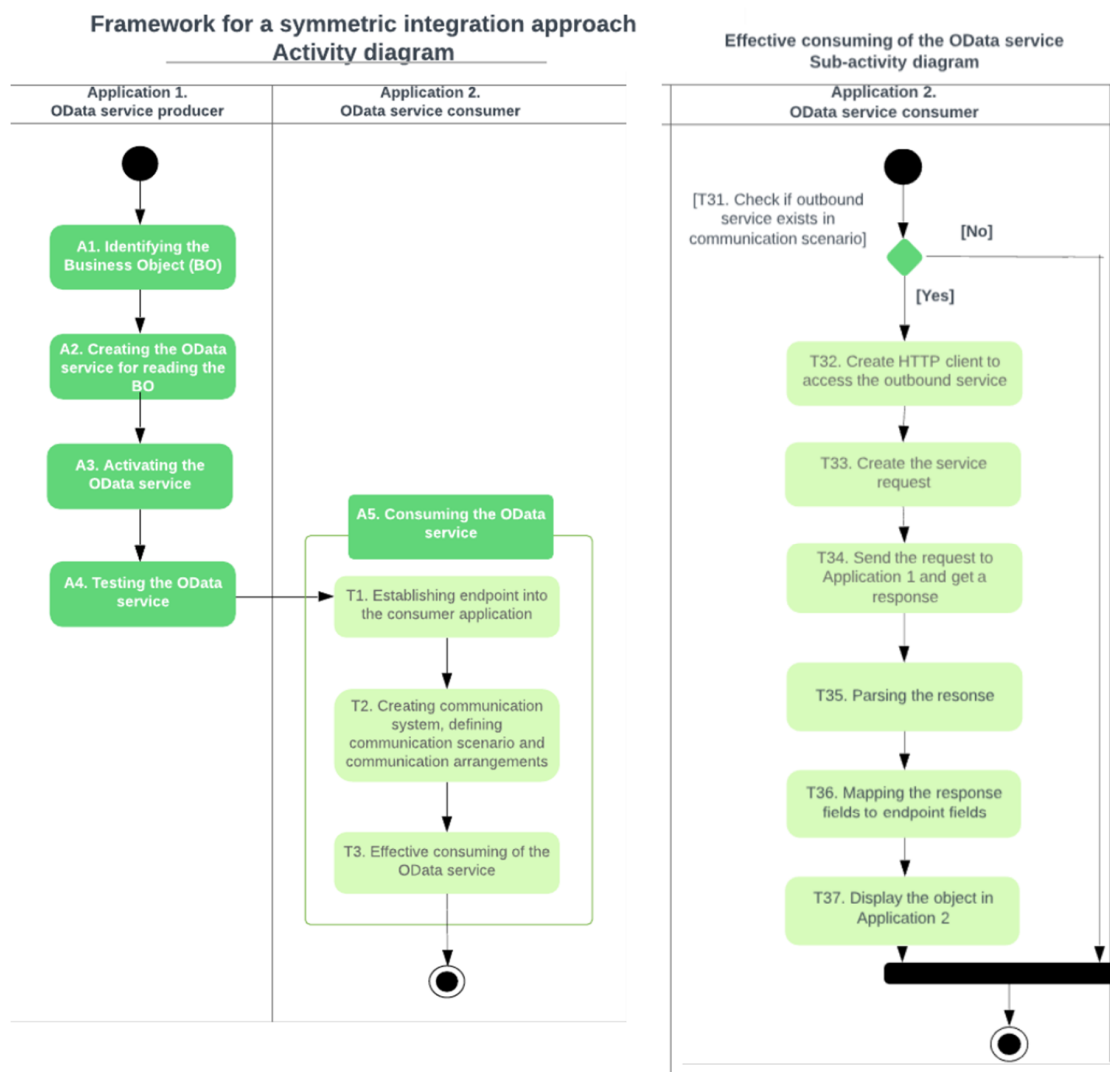


Figure 6. Framework for a symmetric integration approach.

4. Discussions and Conclusions

All forms of cloud integration, e.g., cloud-to-cloud, cloud-to-on-premises integration, or a combination of both, employ data and application integration in order to address different business components.

In all cases, the integration framework includes practices, tools, and technologies used by an enterprise to connect applications, systems, and data. The approach can be developed with or without a cloud integration platform.

Business-to-Business (B2B) enterprises use an integration platform as a service (iPaaS) in order to develop integration projects. The two systems C4C and S/4, which are the subject of our investigation, can be integrated via SAP Cloud Platform Integration (SAP CPI). It connects cloud applications with other SAP and non-SAP cloud and on-premises applications. Different scenarios can be developed, e.g., Application-to-Application (A2A) integration, or customized access to the SAP CPI with public OData APIs [17].

As an alternative, we propose an approach without an integration platform that can be applied to any A2A integration (Figure 6). A methodological framework for the integration of two independent applications is proposed. The scenario presented in Figure 1 indicates a symmetric integration approach: Data is retrieved from one application and stored (copied) in the other application. As mentioned in Section 2, the “copy of data” object is symmetric to the “data source” object. One application is an OData service producer, the other a service consumer (Figure 6).

For the consumption of the OData service (activity A5), three tasks are necessary (T1–T3). Furthermore, task T3 was designed as a sequence of seven sub-tasks (Figure 6). After sending the request to Application 1 and getting the service response (sub-task T34), parsing the response (sub-task T35) followed by mapping the response fields to the endpoint fields in Application 2 (T35) is essential to obtain a symmetric integration approach.

Mainly, based on the OData service [18], the integration framework implies the following phases: A1—Identifying the Business Object, which will represent the data source for the OData service; A2—Creating the OData service for reading the BO; A3—Activating the OData service; A4—Testing the OData service; A5—Consuming the OData service.

Today’s business environment requires application integration in order to support all kinds of business processes and value chains. The business data are processed through all these systems. This has to be done “in the best possible way and in the least possible time to come up with the best possible outcome” [19]. Therefore, all data that has to be transferred from one system to another is identified. “The OData ecosystem” is continually growing, and diverse OData producers and consumers might be the pillars of the integration framework [20]. The symmetry of data has a decisive impact on the quality of the integration processes.

Author Contributions: The main contributions to this article were: M.M.—conceptualization, methodology, writing and validation; C.B.—conceptualization, investigation and validation; T.C.—investigation and writing.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tibbetts, H. Integration on the Edge: Data Explosion & Next-Generation Integration. 2011. Available online: <http://www.ebizq.net/blogs/integrationedge/2011/12/application-integration-in-the-cloud---finally-here.php> (accessed on 2 August 2018).
2. Linthicum, D.S. *Enterprise Application Integration*; Addison-Wesley Professional: Boston, MA, USA, 1999; ISBN 0201615835.
3. Thoopurani, G. Rethinking B2B Business: Enterprise Application Integration (EAI) in the Cloud. 2014. Available online: <https://www.ibm.com/developerworks/cloud/library/cl-softlayer-sterling-trs/cl-softlayer-sterling-trs-pdf.pdf> (accessed on 3 August 2018).

4. Wang, S.H.; Chen, D. A cloud framework for Enterprise Application Integration on SOA. In Proceedings of the International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things, Hsinchu, Taiwan, 4–6 December 2014. [CrossRef]
5. Deloitte Consulting LLP. 3 Ways to Integrate Cloud Applications. 2013. Available online: <https://deloitte.wsj.com/cio/2013/04/16/3-ways-to-integrate-cloud-apps/> (accessed on 15 July 2018).
6. Corrales, D.C.; Corrales, J.C.; Ledezma, A. How to Address the Data Quality Issues in Regression Models: A Guided Process for Data Cleaning. *Symmetry* **2018**, *10*, 99. [CrossRef]
7. Weyl, H. *Symmetry*; Princeton University Press: Princeton, NJ, USA, 1983.
8. Murtagh, F. Symmetry in data mining and analysis: A unifying view based on hierarchy. *Proc. Steklov Inst. Math.* **2009**, *265*, 177–198. [CrossRef]
9. Gartner. Application Integration. 2018. Available online: <https://www.gartner.com/it-glossary/application-integration> (accessed on 9 May 2018).
10. Holland, K. An Introduction to Service Integration and Management and ITIL. 2015. Available online: <https://www.axelos.com/case-studies-and-white-papers/introduction-to-service-integration-management> (accessed on 20 November 2018).
11. Ciularu, H. SAP Open Data Protocol. Available online: <http://www.saponline.ro/sap-open-data-protocol/> (accessed on 23 November 2018).
12. Wang, J. OData Service Backend Implementation in C4C, CRM and S4Hana. 2017. Available online: <https://blogs.sap.com/2017/08/07/odata-service-backend-implementation-in-c4c-crm-and-s4-hana/> (accessed on 20 April 2018).
13. Chua, E. Comparing C4C Integration Methods (A2A, A2X, ODATA) for High Volume Scenarios. 2016. Available online: <https://blogs.sap.com/2016/07/14/comparing-c4c-integration-methods-a2a-a2x-odata-for-high-volume-scenarios/> (accessed on 13 May 2018).
14. Tutorialspoint. SAP C4C Tutorial. Available online: https://www.tutorialspoint.com/sap_c4c/index.htm (accessed on 30 April 2018).
15. SAP. SAP Cloud for Customer Administration Guide (Public Document). 2015. Available online: <https://vdocuments.mx/sap-c4c-admin-guide-2015-05.html> (accessed on 30 April 2018).
16. Tutorialspoint. SAP C4C. Available online: https://www.tutorialspoint.com/sap_c4c/sap_c4c_tutorial.pdf (accessed on 17 June 2018).
17. SAP SE. SAP Cloud Platform Integration Onboarding Guide. 2018. Available online: https://help.sap.com/doc/fc3b7258441440f189cbaa050452780e/Cloud/en-US/onboarding_pdf.pdf (accessed on 20 December 2018).
18. OASIS Standards. Available online: <https://www.oasis-open.org/standards#odatav4.0> (accessed on 24 January 2018).
19. Integrating Business Applications. Available online: <https://www.appseconnect.com/integrating-business-applications-5-key-points-to-consider/> (accessed on 20 January 2019).
20. Pizzo, M.; Handl, R.; Zurmuehl, M. “OData Version 4.0 Part 1: Protocol” OASIS Standard. Available online: <http://docs.oasis-open.org/odata/odata/v4.0/os/part1-protocol/odata-v4.0-os-part1-protocol.pdf> (accessed on 10 December 2017).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).