
CIplus
Band 3/2018

Reference Models for the Standardization and Automation of Data Warehouse Architecture including SAP Solutions

Regys Mene, Hartmut Westenberger, Hrvoje Husic

Technology
Arts Sciences
TH Köln

Reference Models for the Standardization and Automation of Data Warehouse Architecture including SAP Solutions

Authors: Regys Mene, Hartmut Westenberger, Hrvoje Husic

1 Abstract

Architectural approaches are considered to simplify the generation of re-usable building blocks in the field of data warehousing. While SAP's Layer Scalable Architecture (LSA) offers a reference model for creating data warehousing infrastructure based on SAP software, extended reference models are needed to guide the integration of SAP and non-SAP tools. Therefore, SAP's LSA is compared to the Data Warehouse Architectural Reference Model (DWARM), which aims to cover the classical data warehouse topologies.

2 Introduction

Growing information needs of fast changing business models require a stable but scalable and flexible data warehouse (DWH) infrastructure. Many companies started their DWH architecture development either with a centralized data repository and data marts built as described by Bill Inmon (Inmon, 2005) or with a dimensional data model as described by Ralph Kimball (Kimball, 2008). Since then, the infrastructures have evolved evolutionarily and have become complex and hybrid based on various architectural patterns. Here, reference models can help to frame the DWH strategy.

Szwed, Komnata and Dymek published an ontology called DWARM describing classical DWH architecture styles (Szwed, Komnata, & Dymek, 2015). They propose a layer-based reference model with data containers like a temporary staging area, a central data repository and data marts and data containers like extraction, loading and data mart feeding (see Figure 1). DWARM uses taxonomies to organize the core components of the DWH infrastructure reference model. A container class is considered as an abstraction of components able to store data. A process class is seen as an abstraction of the data processing tasks. These two parent classes are used to build up two taxonomies. Both taxonomies are related because data containers are used as sources or targets for the data processing units.

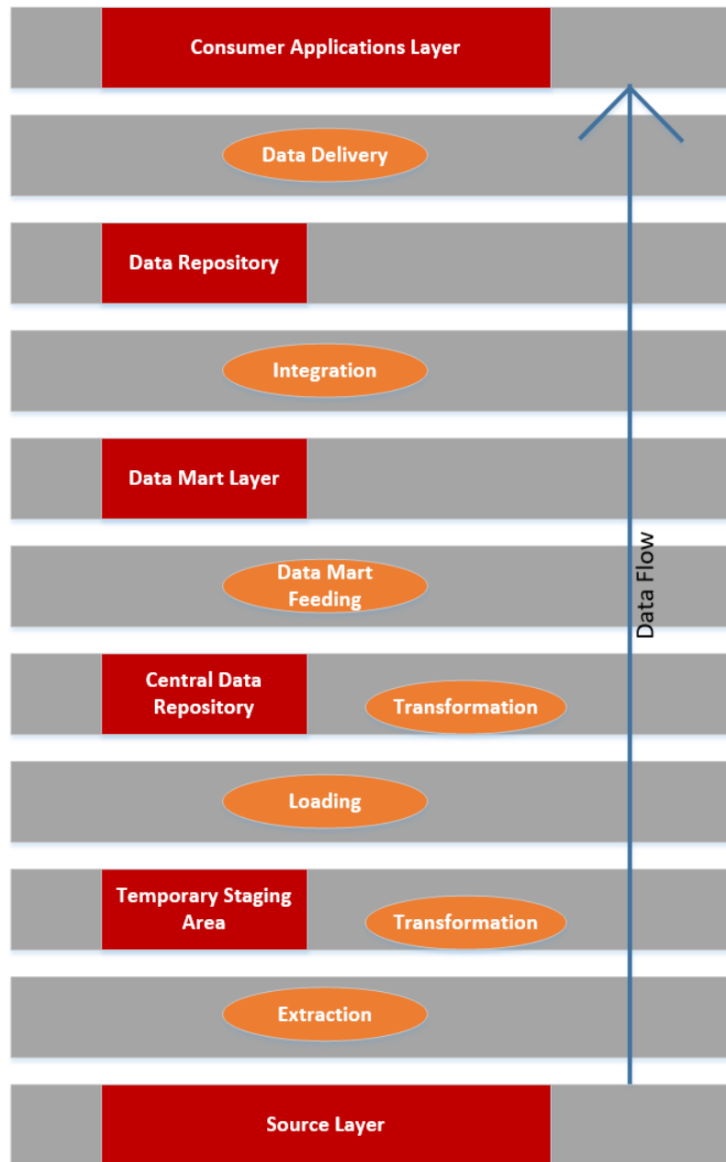


Figure 1: Layers (grey), data containers (rectangles) and processes (ellipses) building the core elements of DWARM (see (Szwed, Komnata, & Dymek, 2015))

Szwed and his co-authors point out how their reference model covers generic classical topologies like Kimball or Inmon. For example, the left model of Figure 2 reflects Inmon's hub and spoke approach in a DWARM-based notation. Due to its generic character DWARM can describe hybrid variants, too. Limitations arise with the assumption of the order of the layers. For example, the Data Vault architecture style substitutes the ETL-process by ELT and the business rule transformation are applied later in the data manipulation. Asif proposes an extension of the DWARM approach to cover Data Vault (Asif, 2017). The Central Data Repository should be separated in sub-layers Raw Data Vault and Business Vault as designed by Linstedt and Olschimke (Linstedt & Olschimke, 2016). The

transformation process has to be divided in a syntactical step assigned to the staging area and a syntactical step (business rules) shifted to the central data repository layer (see right part of Figure 2).

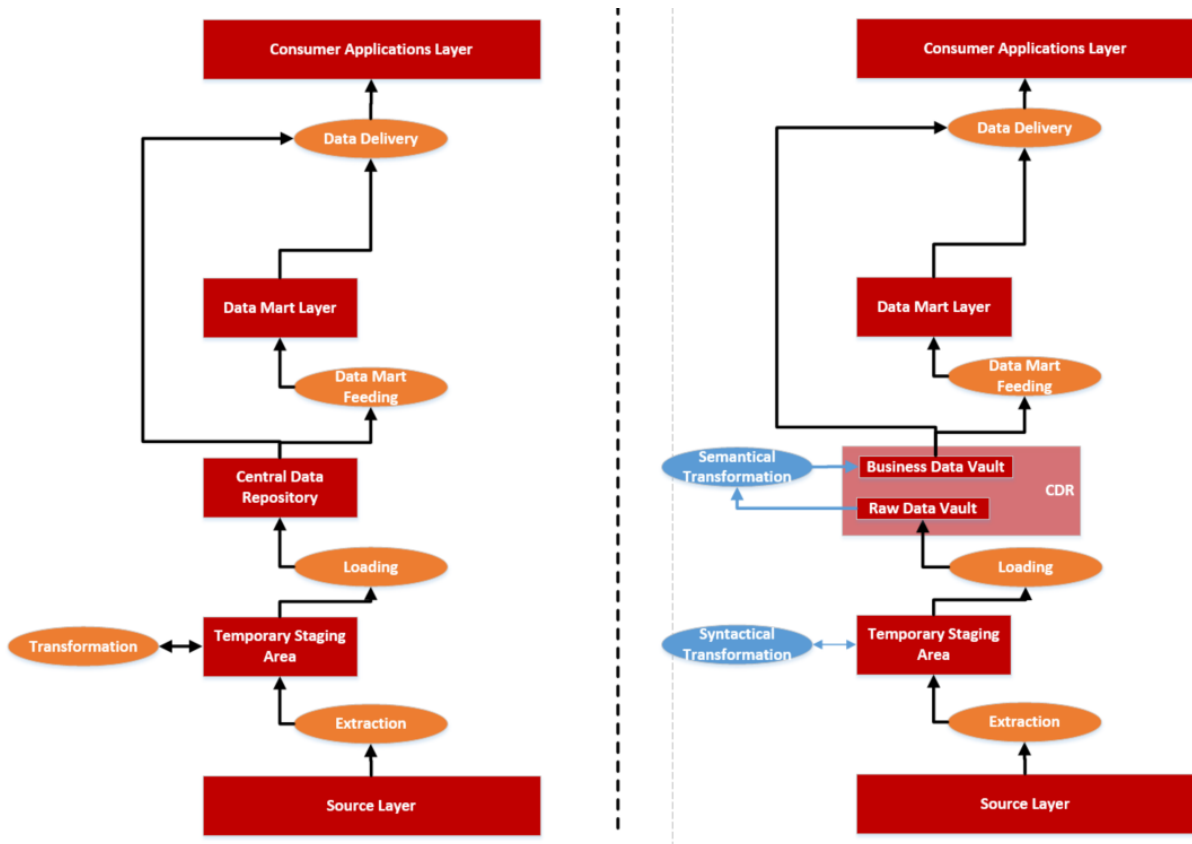


Figure 2: Inmon's DWH Reference Architecture Model (left) and with Data Vault extension (right) according to the DWARM-concept (see (Szwed, Komnata, & Dymek, 2015))

3 Comparison of SAP's LSA to DWARM

SAP offers with the Layered Scalable Architecture (LSA) a DWH architecture reference model, too. Based on this reference model SAP's Business Warehouse solution promises a high degree of standardization and automation of the DWH/BI processes, while not losing the focus on the scalability and flexibility of the DWH/BI processes, see (SAP AG, n.d.- A), (SAP AG, n.d. - B). In a big picture, LSA is composed of two main areas: the *Enterprise Data Warehouse* layer and *Architected Data Mart* layer (see Figure 3.) Additionally, each of these two layers contains further sub-layers. In the Data Acquisition Layer, which is the entry point of the data in the SAP BW environment, the source data is stored in the so-called Persistent Staging Area (PSA). Data cleansing and conforming transformations are performed on it, before data are loaded in the target objects of the Data Propagation layer, Corporate Memory or even Operational Data Stores. The LSA's Data Acquisition layer together with the Quality and Harmonization layer and their data loading process would align to the DWARM's extraction and loading process. Furthermore, the LSA's PSA provides the same function as the DWARM's Temporary Staging Area (TSA).

After the data loading process is executed in SAP BW, data is loaded from PSA to the Data Propagation layer. In fact, the main purpose of the Data Propagation layer is to integrate data that comes from multiple sources into the so-called DWH. Therefore, the Data Propagation layer is actually the DWH layer of the LSA architecture and is comparable to DWARM's Central Data Repository layer, as both layers serve for the consolidation of the data uploaded to the DWH. On the other hand, before loading data to the Data Propagation layer, data could be also loaded in the Corporate Memory layer, which may keep the history of data extracted to the DWH for a longer period (Palekar, Patel, & Shiralkar, 2015). At the first glance a comparable element is not specified in DWARM. It can be argued that DWARM's Central Data Repository layer can cover LSA's Corporate Memory layer. However, a counter-argument would be that DWARM's Central Data Repository Container is seen as the single source of truth for the DWH while the LSA's Corporate Memory layer is just a reproduction of the staging area's (PSA/TSA) storing data persistently for a longer period of time. Nevertheless, depending

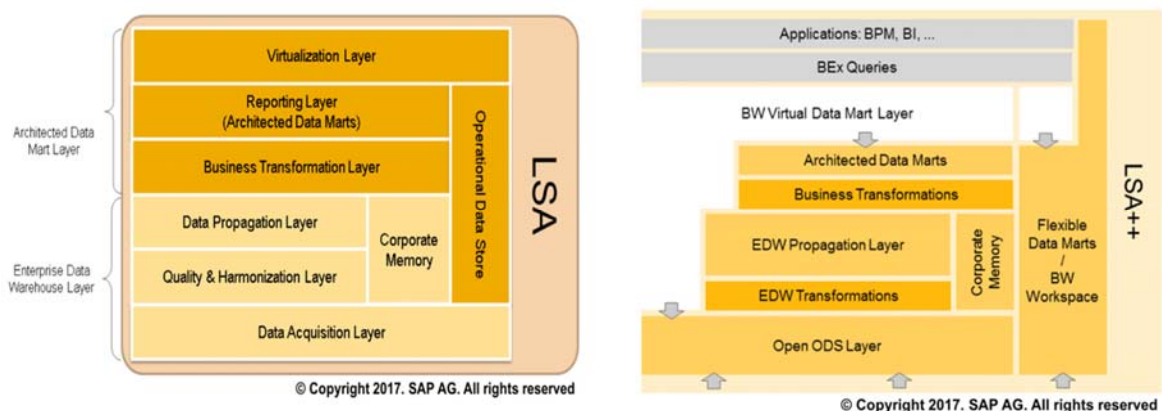


Figure 3: SAP BW Layer Scalable Architecture (LSA and LSA++) (SAP Online Library, 2017)

on the interpretation of this argument, there might exist the possibility to extend the DWARM with an additional Corporate Memory layer.

In order to support the operational reporting needs of the different managerial levels, data from the Data Acquisition layer can be also loaded directly to the Operational Data Store layer. This is similar to the Data Propagation layer. Therefore, particular transformations can be applied on it and make it ready to fulfill the daily decision-making processes. Here, a difference from the Reporting layer is that data have a shorter time span and serves rather to answer the real-time reporting requirements of the end users. In addition to that, another difference is that the objects on this layer do not have a star schema structure as they have in the Reporting (Data Mart) layer, but a flat table structure. This means that with big amounts of data the reporting performance can decline. Figure 4 visualizes the mentioned extensions of DWARM highlighted on blue .

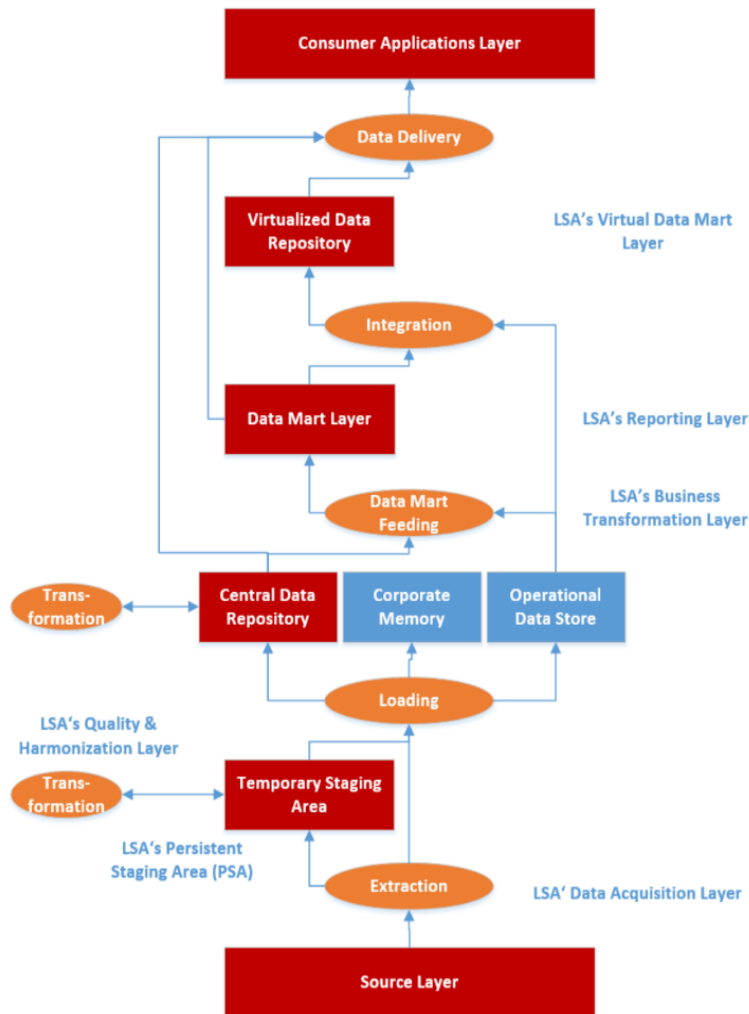


Figure 4: Mapping of LSA on an extended DWARM reference architecture

Going back to the LSA, further logical aggregations can be performed on the detailed data in the Data Propagation layer. These aggregated data is loaded in the reporting layer, where it can be further accessed by queries. Therefore, data on this layer is aggregated and stored persistently in different SAP BW objects (e.g. InfoCubes etc.), which consist of multidimensional (MD) structures. Seen from the DWARM layers perspective, the LSA's Business Transformation and Reporting layers comprise of the same functionality with the DWARM's Data Mart Feeding process and Data Mart container.

At last, the virtualization layer supports the union/join of different reporting layer's objects into a virtual one. Thus, the only difference of the virtualization layer with the reporting one is that data is accessed on the runtime virtually without being stored persistently. When considering this layer from the DWARM layers perspective, one can realize that it performs the same functionality of the DWARM's data integration processes and Federated Data Repository container.

Finally, DWARM contains the two additional layers Data Delivery and Customer Applications, which are not included in the LSA reference architecture's figure above, but are actually equivalent to the required SAP BW queries (known as BEx Queries), which are then needed to expose the data to different BI/BO applications. Nevertheless, when placing against the SAP BW's LSA and LSA++ architectures against each others, besides the structural differences that they have with each other, in overall, the difference in contexts between LSA and LSA++ are not that big. However, it is worth mentioning that the LSA++ is the type of reference architecture that aligns best to the agile DWH modeling. Therefore, with its upgraded methods of DWH modeling and its enhanced objects, it is now easier to rapidly build new data flows and at the same time, offer better data loading and reporting performance.

As it can be seen in Figure 3 SAP BW has added further components and seem to have made the LSA++ a more advanced version of its initial LSA reference architecture. Moreover, from the implementation perspective, LSA++ uses enhanced SAP BW objects that have better loading and accessing performance.

Particularly, when jumping to the Open Operational Data Store (ODS) layer, it performs the same functions as the LSA's Data Acquisition and ODS layer. As such, the OpenODSView object, typically used in this layer, is a virtual object that does not store any data persistently. However, it may also be used to generate persistent Data Sources to stage data physically in SAP BW on HANA (SAP AG, n.d.- A). On the other hand, it can extract data virtually from the SAP HANA DB and bring it in the SAP BW on HANA environment. From there, this virtually extracted data can be then combined with persistent InfoProviders in the SAP BW using CompositeProviders, which can be latter then used for reporting. Furthermore, it is also possible to build queries for reporting directly on the OpenODSView object. This added functionality in LSA++ reference architecture enables then the real-time reporting of the data directly at the Open ODS layer without any data latency. Thus, the initial functionality of LSA's ODS layer is now fulfilled by the LSA++'s Open ODS layer and its OpenODSView object. At last, this layer provides the necessary capabilities to share the SAP BW on HANA InfoProviders as views in the HANA DB environment and vice-versa, HANA DB views into SAP BW on HANA (Haupt, 2012). In this way, users have the opportunity to work with their DWH models using the hard-coded SQL queries in HANA DB or the provided SAP BW on HANA interfaces.

Following the upcoming data flow modelling, when comparing further LSA++ layers to the ones of LSA, not so many differences are identified among them. Up to the Architected Data Mart layer (corresponding to the Reporting layer in LSA) everything is almost the same. On the other hand, the main differences here reside in the enhanced SAP BW on HANA objects used in the SAP BW on HANA (LSA++). As such, the ADSO is considered as a universal object that can be used in multiple layers such as the Corporate Memory, Data Propagation layer etc., but with different settings in its definitions for each layer. Furthermore, in the LSA++'s BW Virtual Data Mart layer, the CompositeProvider object substitutes the functionalities of the previously used objects (e.g. MultiProvider etc.) for the Virtualization layer in the LSA. Moreover, an additional improvement of the LSA++ here is that the CompositeProvider is able to virtually integrate and report on SAP BW InfoProviders from any layer of the LSA++, without losing the reporting performance, as it would normally happen in the SAP BW on any DB (LSA) using MultiProvider objects. At last, the LSA++'s Flexible Data Marts/BW Workspace layer, signifies the agile capabilities offered by LSA++. Thus, data from the actual SAP BW models, can be further exposed and extended using local or departmental data to fulfill a department's agile and ad-hoc reporting needs (Peleshuk, 2015).

After having a deep overview of the DWARM and LSA/LSA++ reference architectures it is of interest to evaluate the degree of standardization and automation provided by SAP BW towards generating different DWH architectural models. Table 1 provides an overall comparison between DWARM and LSA/LSA++ reference architectures.

Generally speaking, the LSA and LSA++ reference architectures have a high similarity with the DWARM one. However, the LSA particularly has two more layers (Corporate Memory and Operational Data Store) compared to the DWARM. On the other hand, the LSA++'s Flexible Data Mart and BW Workspaces layer is not appended to the DWARM layers, as it is too specific in comparison to the overall purpose of DWARM itself. Nevertheless, regarding the above-suggested extensions in the DWARM layers, the authors of it also admit the fact that their reference architecture model is mainly provided on the generic level and has place for further customization with regard to specific DWH architectures (Szwed, Komnata, & Dymek, 2015). On the other hand, an argument for the extension of DWARM with regard to the LSA and LSA++ layers may follow from the fact that the actual purpose of LSA and LSA++ is to fit at best the capabilities supported by the SAP BW. Therefore, in comparison to DWARM, one could see the SAP BW's LSA and LSA++ reference architectures in a level down in granularity.

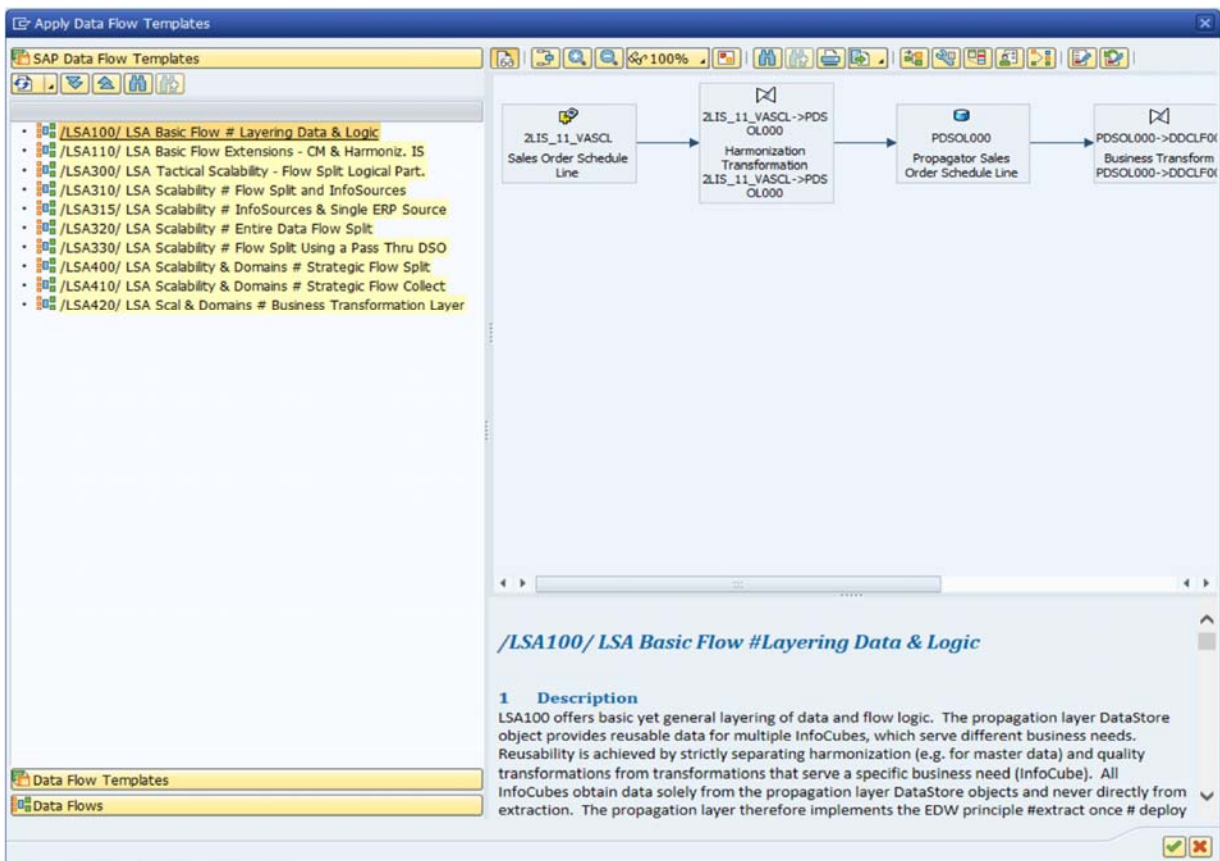
Nonetheless, the following suggested layers to be extended to DWARM are not only found in the LSA and LSA++, but may also be useful in the cases of the above-mentioned DWH architectures. As such, in their book about their DWH approach, Kimball & Caserta (2004) identify the functionality of the LSA's and LSA++'s Corporate Memory layer with the ones of long-term archiving of the staged data. Indeed, this is also the purpose of the LSA and LSA++'s Corporate Memory layer. Thus, the authors suggest that staged data should be archived in a long-term repository (in this case, in the Corporate Memory layer) up to the moment that it is sure that it will not be needed anymore (Kimball & Caserta, 2004, S. 8).

Table 1: Mapping between LSA, LSA++ and DWARM layers and SAP BW objects

DWARM Layers	LSA Layers	LSA++ Layers	SAP BW objects assigned to the respective DWARM and LSA/LSA++ layers
1. Source	Source Systems (outside SAP BW environment)	Source Systems (outside SAP BW environment)	SourceSystem (SAP ERP, DB, FlatFile, Webservices etc.)
2. Extraction*	Data Acquisition Layer, Quality & Harmonization Transformations	Open Operational Data Store Layer, EDW Transformations	DataSource, InfoSource, Transformation, InfoPackage (SAP BW on any DB + HANA); OpenODSView (SAP BW on HANA) etc.
3. Temporary Staging Area (TSA)	Persistent Staging Area (PSA)	Persistent Staging Area (PSA) optional, Operational Data Provisioning (ODP)	DataSource (SAP BW on any DB + HANA); ODP DataSource (SAP BW on any HANA);
4. Loading*	Data Transfer Process (DTP)	Data Transfer Process (DTP)	Data Transfer Process (DTP) (SAP BW on any DB + HANA)
5. Central Data Repository	Data Propagation Layer	EDW Data Propagation Layer	DataStore (DSO) (SAP BW on any DB); ADSO (SAP BW on HANA)
6. Corporate Memory (Extended Layer)	Corporate Memory Layer	Corporate Memory Layer	Write-Optimized DSO (WO-DSO) (SAP BW on any DB); ADSO (SAP BW on HANA)
7. Operational Data Store (Extended layers)	Operational Data Store Layer	Open Operational Data Store Layer	Real-time Data Acquisition (RDA) DSO, VirtualProvider etc. (SAP BW on any DB); OpenODSView (SAP BW on HANA) etc.
8. Data Mart Feeding*	Data Transfer Process (DTP)	Data Transfer Process (DTP)	Data Transfer Process (DTP) (SAP BW on any DB + HANA)
9. Data Mart	Reporting Layer	Architected Data Mart Layer	InfoCube, semantically partitioned Object (SAP BW on any DB); Advanced InfoCubes, ADSO (SAP BW on HANA) etc.
10. Data Integration*	-	-	-
11. Federated Data Repository	Virtualization Layer	BW Virtual Data Mart Layer	InfoSet (SAP BW on any DB); Multiprovider (SAP BW on any DB + HANA); CompositeProvider (SAP BW on HANA) etc.
12. Data Delivery*	BEx Queries Layer	BEx Queries Layer	BEx Query (SAP BW on any DB + HANA);
13. Customer Applications	SAP BW/ Business Objects Applications	SAP BW/ Business Objects Applications	SAP BW/ Business Objects Applications
* The layers that are marked with "*" in the DWARM are actually processes and therefore they cannot be specifically found in the LSA and LSA++ reference architectures. In the above table, for them are assigned the related processes of SAP BW based on the reference architectures of LSA and LSA++.			

In addition to that, the authors state that “[...] It is almost always less of a headache to read data back in from permanent media than it is to reprocess data through the ETL system at a later time.” (Kimball & Caserta, 2004, S. 8). Moreover, in difference from the DWARM’s Temporary Staging Area (TSA) layer, in the Corporate Memory layer, the long-term data persistency does not come with those performance costs, as it would be in the case of storing long-term data in the temporary staging area. The reason for this resides in the fact that the more data is stored in the TSA/PSA for a long-term perspective, the higher is the negative influence in the performance of ETL processes in the DWH. For this reason, adding a Corporate Memory layer to DWARM and its derived DWH architectures may resolve such issues.

Regarding the extension of the ODS layer in the DWARM, although the ODS layer is found in different DWH architectures, Kimball and Caserta suggest that nowadays, the modern DWH-s already provide possibilities for real-time reporting, which are equivalent to the ones offered by the ODS (Kimball & Caserta, 2004, S. 20). Therefore, the ODS as a separate layer in a DWH architecture is seen as a redundancy from the above-mentioned authors. Moreover, the authors suggest the usage of the real-time data marts attached to static data marts to substitute the ODS layer (Kimball & Caserta, 2004, S. 425-428).



© Copyright 2017. SAP AG. All rights reserved

Figure 5: SAP BW - LSA Data Flow Templates (SAP System Screenshot, 2017)

Although this point might make sense when considering Kimball's DWH Bus Architecture; when considering Inmon's or Centralized DWH architecture, there is still the need for an ODS layer, as the main goal of the DWH itself is not to report directly on it (as it would be the case with the ODS), but integrate and provide atomic data to the upcoming data marts layer.

Moreover, in order to have the best performance in loading data and avoid data redundancy, the DWH structure should be in highly normalized form. As such, when having ODS shrunk into the DWH structure, the data accessing and query performance for the near-real time operational reporting would drastically decline. Therefore, similarly to the other DWARM layers, depending on the chosen type of DWH architecture (whenever a Central Data Repository container is used), the ODS layer may be appended to the derived DWH architecture models from DWARM.

In this context, similarly to DWARM, the LSA/LSA++ reference architecture models support different DWH modelling scenarios to be based on them for their further customization.

As such, to standardize and automatize the generation of different high-level/specific DWH architectural models, SAP BW offers several best-practice pre-defined data flow templates, which are actually based on the LSA layers. These templates can be further used and extended for any user-specific modeling scenario. On the other hand, the availability and usage of the BI Content in SAP BW, enforces the standardization and automation of several data-warehousing aspects.

4 Conclusion

To sum up, the usage of LSA and LSA++ to generate different DWH architectural models can be considered as an accelerated method for standardizing and automatizing (when using LSA Data Flow Templates) the generation of different DWH architectures. A reference model like DWARM helps to make SAP's architectural approach transparent and to embed SAP's solution building blocks into a heterogenous BI landscape. In this way, unnecessary costs needed to redundantly develop DWH architecture model concepts that are already once implemented, is avoided. This leads to higher agility in the overall project management and reduced TCO-s of the DWH/BI projects.

5 Disclaimer

This publication contains references to the products of SAP AG.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

SAP AG is neither the author nor the publisher of this publication and is not responsible for its content. SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

6 References

Asif, M. (2017). *Assessment of Data Vault Efficiency for Industrialization of Data Warehousing and Business Intelligence*. Cologne: Cologne University of Applied Sciences.

Haupt, J. (2012, August). *SAP HANA as Driver of EDW Evolution: LSA++ (Layered Scalable Architecture) for BW on SAP HANA*. Retrieved from SAP AG:
<https://assets.cdn.sap.com/sapcom/docs/2012/08/3273bcde-517c-0010-82c7-eda71af511fa.pdf>

Inmon, W. H. (2005). *Building the data warehouse* (4th ed ed.). Indianapolis, Ind: Wiley.

Kimball, R. (2008). *The data warehouse lifecycle toolkit* (2nd ed ed.). Indianapolis, IN: Wiley Pub.

Kimball, R., & Caserta, J. (2004). *The data warehouse ETL toolkit*. Indianapolis Ind. u.a.: Wiley.

Linstedt, D., & Olschimke, M. (2016). *Building a scalable data warehouse with data vault 2.0*. Amsterdam, Netherlands: Morgan Kaufmann.

Palekar, A., Patel, B., & Shiralkar, S. W. (2015). *SAP BW 7.4 - practical guide* (3rd edition ed.). Rheinwerk Publishing, Inc.

Peleshuk, S. (2015, January 19). *A Few Cool Features in BW 7.4 on HANA that Make a Difference*. Retrieved from BIPortal.ORG: <https://www.biportal.org/single-post/2015/01/19/A-Few-Cool-Features-in-BW-74-on-HANA-that-Make-a-Difference>

SAP AG. (n.d. - B). *SAP Documentation*. Retrieved from Layer Architecture of Data Warehouse (LSA): https://help.sap.com/saphelp_nw73/helpdata/en/22/ad6fad96b547c199adb588107e6412/frame-set.htm last called 20.01.2018

SAP AG. (n.d.- A). *SAP Documentation*. Retrieved from Layer Architecture of a Data Warehouses with SAP HANA Database (LSA++): https://help.sap.com/saphelp_nw73/helpdata/en/22/ad6fad96b547c199adb588107e6412/frame-set.htm last called 20.01.2018

Szwed, P., Komnata, W., & Dymek, D. (2015). DWARM: An Ontology of Data Warehouse Architecture Reference Model. In *Kozielski, Mrozek et al. (Ed.) 2015 – Beyond Databases, Architectures and Structures* (Vol. 521, pp. 222–232).

Kontakt/Impressum

Diese Veröffentlichungen erscheinen im Rahmen der Schriftenreihe "Ciplus". Alle Veröffentlichungen dieser Reihe können unter

<https://cos.bibl.th-koeln.de/home>
abgerufen werden.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Datum der Veröffentlichung: 23.07.2018

Herausgeber / Editorship

Prof. Dr. Thomas Bartz-Beielstein,
Prof. Dr. Wolfgang Konen,
Prof. Dr. Boris Naujoks,
Prof. Dr. Horst Stenzel
Institute of Computer Science,
Faculty of Computer Science and Engineering Science,
TH Köln,
Steinmüllerallee 1,
51643 Gummersbach
url: www.ciplus-research.de

Schriftleitung und Ansprechpartner/ Contact editor's office

Prof. Dr. Thomas Bartz-Beielstein,
Institute of Computer Science,
Faculty of Computer Science and Engineering Science,
TH Köln,
Steinmüllerallee 1, 51643 Gummersbach
phone: +49 2261 8196 6391
url: <http://www.spotseven.de>
eMail: thomas.bartz-beielstein@th-koeln.de

ISSN (online) 2194-2870

**Technology
Arts Sciences
TH Köln**

