The BW Layered Scalable Architecture (LSA)
An Introduction

Juergen Haupt, SAP NetWeaver RIG BI EMEA
KHNC March, 11th 2009
1. What is the LSA?
2. SAP LSA Overview
   - LSA Building Blocks
     - Layers
     - Domains
   - LSA Implementation Reference
   - LSA Operations Reference
3. Lifecycle of the Customer LSA

Note: All slides are taken from the Workshop PDEBW1
‘Blueprinting an Enterprise Data Warehouse – The BW Layered, Scalable Architecture (LSA)’
SAP introduces the term **LSA – Layered, Scalable Architecture** in order to describe the design of service-level oriented, scalable, best practice BW architectures founded on accepted EDW principles*.

The LSA serves as a reference architecture to design transparent, complete, comprehensive customer DWH architectures (Customer LSA).

The Customer LSA describes corporate standards to build BI applications in a performant, maintainable, flexible manner.

* As introduced in Bill Inmon’s Corporate Information Factory (CIF)
SAP LSA and Customer Adoption

SAP LSA: The BW Reference Architecture

Customer LSA Standards - Handbook

BI Project Design

Step 1: Design

Step 2: Apply

© SAP 2009 / KHNC- BW Layered, Scalable Architecture (LSA) Intro, Juergen Haupt /200903/ Page 4
1. What is the LSA?

2. SAP LSA Overview
   1. LSA Building Blocks
      1. Layers
      2. Domains
   2. LSA Implementation Reference
   3. LSA Operations Reference

2. Lifecycle of the Customer LSA
SAP LSA: The Reference Architecture

**LSA Building Blocks Reference**
- Describes core structures & definitions
- Building Blocks
  - Layer
    - Data Quality
    - Data Integration
    - ETL
  - Domains
    - Data Model
    - Landscape
    - Storage
    - Ownership
    - Development concept

**LSA Implementation Reference**
- Describes design standards to build BI applications founded on building blocks
- Data Staging/Management for transactional & master data
  - Persistent Objects
  - Flows - scheduled/recovery
  - Transformation
  - Programming (Abap)
  - Organization (Process Ch.)
- Meta Data Management
  - Naming Conventions
  - Organization (InfoAreas)

**LSA Operations Reference**
- Describes Support Scenarios
- Life Cycles
  - Information
  - Meta Object
  - LSA
- Administration
  - Data Base
  - Housekeeping
  - Monitoring
- Transport
- Security

© SAP 2009 / KHNC- BW Layered, Scalable Architecture (LSA) Intro, Juergen Haupt /200903/ Page 6
From LSA Reference to Customer LSA

- LSA Building Blocks Reference
- LSA Implementation Reference
- LSA Operations Reference

- Customer-LSA Building Blocks
- Customer-LSA Implementation Standards
- Customer-LSA Operations Standards

Core structures & definitions

Design standards to build BI applications founded on building blocks

Support Scenarios
The LSA is a reference architecture.
A concrete customer LSA will always be unique

As background and targets of each customer differ, the preferred services differ and thus the customer LSA (layers & domains)

Customers differ with respect to:
- Priority of services
  - Painful experiences -> Control & Influence (outsourcing)
- State of integration (master data, source systems (processes))
- Skills
- Overall governance, sponsorship, commitment
- Industries
- Their starting point
  - Enterprise Application Rollout Driven BW EDW
  - Heterogeneous Source Driven BW EDW
    - All in one EDW for local and group reporting
    - Classic EDW for group reporting
1. What is the LSA?
2. SAP LSA Overview
   - LSA Building Blocks
     - Layers
     - Domains
   - LSA Implementation Reference
   - LSA Operations Reference
3. Lifecycle of the Customer LSA
SAP LSA: The Reference Architecture

LSA Building Blocks Reference

Describes core structures & definitions

- Building Blocks
  - Layer
    - Data Quality
    - Data Integration
    - ETL
  - Domains
    - Data Model
    - Landscape
    - Storage
    - Ownership
    - Development concept

LSA Implementation Reference

Describes design standards to build BI applications founded on building blocks

- Data Staging/Management for transactional & master data
  - Persistent Objects
  - Flows - scheduled/recovery
  - Transformation
  - Programming (Abap)
  - Organization (Process Ch.)

- Meta Data Management
  - Naming Conventions
  - Organization (InfoAreas)

LSA Operations Reference

Describes Support Scenarios

- Life Cycles
  - Information
  - Meta Object
  - LSA
- Administration
  - Data Base
  - Housekeeping
  - Monitoring
- Transport
- Security

© SAP 2009 / KHNC- BW Layered, Scalable Architecture (LSA) Intro, Juergen Haupt /200903/ Page 10
LSA Reference Layers
Quick Intro

- Reporting & Analysis ready
  - Often Near Real Time Reporting
  - Granular, operational like

- Apply business logic

- Digestible data - ready to consume for BI applications
  - Harmonized view on data
  - Application neutral
  - Corporate owned
  - Granular
  - Business Key

- Create harmonized view
  - Guarantee quality

- Accepts extracted data 1:1
  - Temporary

- Source system like service level
  - Long term, granular
  - Comprehensive, complete
  - Master the unknown
**Detailing LSA Reference Layers**

**Acquisition to Propagator Layer**

Propagators

This flow describes daily, weekly, monthly recurring staging of data to feed finally the BI application layers.

The staging creates data in Propagator DSOs, which are easy to digest for BI applications on top.

Easy to digest means standards like:

- **Additive delta** i.e. data can be directly processed into InfoCubes
- **Data are integrated** if the BI applications ask for integrated data
- **Data are local** if the BI applications ask for local, not harmonized data
- **Data have no flavor** with respect to specific business rule transformations but offer additional data with respect to the loaded data, which are commonly or frequently needed by the BI applications

- Manageable portions of data to fulfill Report Availability, Recovery, Administration SLAs (→ Domains)

- ......
The core service of a Propagation Layer is to offer digestible data to applications.

Digestible may mean:

- harmonized data in the broadest sense (for more details s. Chapter on Data Model)
  1. integrating data: common semantics, common values
  2. smoothing data: common semantics, technically unified values (e.g. compounding)

- trimmed to fit DataSources and Data persistency's to
  - Reduce data complexity for applications
    1. Extending DataSources by looking up information, which applications frequently ask for. Note: introduced parent-child relationship must be stable otherwise realignment issues!
    2. Merging different but highly related DataSources and store data in a single propagator, if applications always or frequently request them together (e.g. HR InfoTypes, avoiding extractor enhancements)
  - Provide sound data portions for better support of application services (availability etc)
    3. Collecting data from the same (or similar) DataSource but from different source systems to less or a single source system independent propagator (s) (⇒ LSA domains)
    4. Splitting data from a DataSources into multiple persistency’s with identical structure (⇒ LSA domains)
Detailing LSA Reference Layers

Content of EDW Layers

- **DataSource**
  - **no history**
  - Comprehensive: all fields

- **DataSource**
  - Complete history
    - ‘booking history’
    - Source system SLA
  - Complex Harmonization:
    - common semantics
    - mapping
    - content consolidation
    - Harmonized fields + add fields

- **DataSource**
  - Limited history
    - Business key
  - Less comprehensive + add fields

- **DataSource**
  - Complete history
    - Business key

Acquisition Layer

Layer Moldings

Corporate Memory Layer

Harmonization:
- compounding
- concatenation
- or 1:1

© SAP 2009 / KHNC- BW Layered, Scalable Architecture (LSA) Intro, Juergen Haupt /200903/ Page 14
Detailing LSA Reference Layers
Propagation Layer & Trimming Data

1. Extend
Add data
Data Source A

2. Merge
‘Data Source A & B’
Propagator
Data Source A
Data Source B

3. Collect
‘Data Source A’
Propagator
Data Source A
Source 1
Data Source A
Source 2

4. Split
‘Data Source A’
Propagator 1
‘Data Source A’
Propagator 2
Data Source A

Note on Collecting and Splitting Data Sources:
- This is very close related to LSA Domains!
- Both may not be applied without regarding volume of data!
# LSA Reference Layers

## Data Propagation Layer Flier

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>potential sources</td>
<td>Data Acquisition Layer, Harmonization Layer, Corporate Memory</td>
</tr>
<tr>
<td>potential targets</td>
<td>Business Transformation Layer, Reporting Layer</td>
</tr>
<tr>
<td>reusability</td>
<td>Yes</td>
</tr>
<tr>
<td>transformations</td>
<td>Additional, stable fields to increase (re-)useability &amp; accessibility (e.g. currency translation). No application-specific rules!</td>
</tr>
<tr>
<td>granularity</td>
<td>single records, granularity defined by DataSource business-key</td>
</tr>
<tr>
<td>content</td>
<td>- DataSource specific</td>
</tr>
<tr>
<td></td>
<td>- as comprehensive as possible, if propagator is expecting volatile requierments</td>
</tr>
<tr>
<td></td>
<td>- Merge of different DataSources to reduce complexity</td>
</tr>
<tr>
<td>history</td>
<td>Minimum history defined by requirements of target-applications/ dependent from Corporate Memory existence</td>
</tr>
<tr>
<td>main services</td>
<td>- ‘Single Point of Truth’ for BI applications (Business Transf. &amp; Reporting Layers)</td>
</tr>
<tr>
<td></td>
<td>- Provide digestible (additive delta, content, performance) data for BI applications</td>
</tr>
<tr>
<td>store &amp; deploy</td>
<td>- ‘normal’ DSO in overwrite</td>
</tr>
<tr>
<td></td>
<td>- semantical/ logical partitioned for large scale DWH/ time-zone support</td>
</tr>
<tr>
<td>update</td>
<td>driven by BI application requirements (report availability)</td>
</tr>
<tr>
<td>housekeeping</td>
<td>Regular delete of DSO change-log content</td>
</tr>
</tbody>
</table>
1. What is the LSA?
2. SAP LSA Overview
   - LSA Building Blocks
     - Layers
     - Domains
   - LSA Implementation Reference
   - LSA Operations Reference
3. Lifecycle of the Customer LSA
SAP LSA: The Reference Architecture

LSA Building Blocks Reference

Describes core structures & definitions

Building Blocks
- Layer
  - Data Quality
  - Data Integration
  - ETL
- Domains
  - Data Model
  - Landscape
  - Storage
  - Ownership
  - Development concept

LSA Implementation Reference

Describes design standards to build BI applications founded on building blocks

Data Staging/Management for transactional & master data
- Persistent Objects
- Flows - scheduled/recovery
- Transformation
- Programming (Abap)
- Organization (Process Ch.)

Meta Data Management
- Naming Conventions
- Organization (InfoAreas)

LSA Operations Reference

Describes Support Scenarios

Life Cycles
- Information
- Meta Object
- LSA

Administration
- Data Base
- Housekeeping
- Monitoring

Transport

Security
## LSA Domains: Background – The Broad View of BW on EDW

<table>
<thead>
<tr>
<th></th>
<th>Traditional EDW</th>
<th>SAP BW EDW</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Companies</strong></td>
<td>- operational world fragmentation</td>
<td>- Increasing operational world harmonization</td>
</tr>
<tr>
<td><strong>situation</strong></td>
<td>- no view across the business</td>
<td>- Yet limited view across the business</td>
</tr>
<tr>
<td><strong>Scope-Areas</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>**Cross-/</td>
<td>- In scope</td>
<td>- In scope</td>
</tr>
<tr>
<td>Corporate BI</td>
<td>- little information freshness (monthly)</td>
<td>- high information freshness (daily)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- high flexibility</td>
</tr>
<tr>
<td>**Local-/</td>
<td>- Not in scope</td>
<td>- In scope</td>
</tr>
<tr>
<td>depart. BI</td>
<td></td>
<td>- standard reporting with local adoptions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- flexible roll out</td>
</tr>
<tr>
<td><strong>Operational</strong></td>
<td>- Not in scope</td>
<td>- Partly in scope</td>
</tr>
<tr>
<td><strong>BI</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Evaluation</strong></td>
<td>- Long time for ROI</td>
<td>- Incremental approach to EDW</td>
</tr>
<tr>
<td></td>
<td>- High Latency (e.g. monthly)</td>
<td>- Immediate ROI (local BI)</td>
</tr>
<tr>
<td></td>
<td>- High costs</td>
<td>- Std. DWH Latency (day) to Low (RDA)</td>
</tr>
<tr>
<td></td>
<td>- Low synergies for local/departmental BI</td>
<td>- Standardization lowers costs</td>
</tr>
<tr>
<td></td>
<td>- Overall acceptance problems</td>
<td>- High synergies for all BI flavors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Increasing acceptance over time</td>
</tr>
</tbody>
</table>
The goals of an EDW providing cross-organizational insights are generally accepted.

But what happens with the BI needs of all the organizational units (e.g. markets) and what’s about overall BI/ DWH consolidation?

The originally limited view of EDW promoters is one of the main reasons for missing acceptance of the EDW investments.

A lot of SAP customers thinking of a BW EDW have a broader scope:
Yes, of course we urgently want this cross-corporate BI but on the other hand side we urgently need a consolidation of DWHousing & BI for all their organizational units to support the daily business based on corporate best practice.

Nobody can afford (from business & cost point of view) on the long run that all organizational units build their own BI & DWH reality.

Why not consolidate and standardize BI & DWHhousing ‘local’ BI requirements having EDW principles in mind and building incrementally the foundation for cross-corporate BI, the BW EDW, but having immediate ROI standardizing ‘local’ BI?

The LSA addresses an evolutionary EDW approach introducing Data Domains to support ‘local’ BI services without neglecting the broader EDW picture.

Note: of course we find also the ‘traditional’ EDW approach using BW!
Domains mean structuring / modelling of data within the layers
Transparent, disjoint structuring of transactional data using stable criteria.
Target is the support of:

- Independency / autonomy of organizations
- 24x7, time-zones
- Scalability / performance/ low latency (parallel vers. sequential)
- Challenging recovery window
- Embedding into RDBMS
- Implementation & operational robustness

Advantages

- Transparency & Flexibility
  - Development, Maintenance
  - Administration, Operations
- Scalability & Robustness
  - Application
  - Load / Query Performance
  - Database Integration
A BW EDW replaces a bunch of existing BWs and/or legacy DWHs (BI Consolidation) spread across the organization. To enable comparable services like we had in a distributed, multiple DWH instance world *(yes, there are some nice things)* we introduce Data Domains in a BW EDW that divide the transactional data but use identical meta data.

**Using Domains in a BW EDW stands for manageability & flexibility**

**Domains allow SLAs in a BW EDW like in a distributed BW world**
II) BW EDW Data Domains Supporting Enterprise ERP Rollout

A single BW EDW shall offer standard reporting & analytics for all organizational units in a global ERP rollout.

To enable comparable services like we had in a distributed, multiple BW instance world we introduce Data Domains in a BW EDW that divide the transactional data but use identical meta data.

Using Domains in a BW EDW stands for manageability & flexibility.
Domains allow SLAs in a BW EDW like in a distributed BW world.
III) BW EDW Data Domains to Divide Data by Sources-‘Quality’

- Main Domain: main ERP
  - stable, controlled

- Remote Domain:
  - remote ERP 1
  - remote ERP 2
  - less stable, no control

- Using Domains in a BW EDW stands for manageability & flexibility
- Domains allow SLAs in a BW EDW like in a distributed BW world
Summary: Domains makes always sense keeping large BW EDWs manageable & flexible

- Using Domains in a BW EDW stands for manageability & flexibility
- Domains allow SLAs in a BW EDW like in a distributed BW world
LSA Domains distribute the transactional data for the entire BW EDW or certain areas (flows) in a disjunctive manner. The meta data definitions of domains are common.
LSA Data Domains and Layers
Properties of LSA Domains

As a rule of thumb:

- Domains organize data by a **general criteria** driven by reporting, BI and manageability requirements, i.e. domains often **differ from operational data organization**.
- Domains are **disjoint** from a transactional data point of view – harmonized master data is shared.
- Domains use **identical meta data definitions**.
- **Cross-views** are achieved through MultiProviders or dedicated persistent InfoProviders.
- Domains should be **stable**.
- Domains should be **consistent** across all layers (across all flows).
- Domains should be **general** for all layers, exceptions:
  - The Acquisition Layer inherits the structuring from source systems / extractors.
  - Domains do not apply on the Corporate Memory.
From Reference LSA to Customer LSA:
- Individual Domains
1. What is the LSA?

2. SAP LSA Overview
   - LSA Building Blocks
     - Layers
     - Domains
   - LSA Implementation Reference
   - LSA Operations Reference

3. Lifecycle of the Customer LSA
SAP LSA: The Reference Architecture

LSA Building Blocks Reference

- Describes core structures & definitions

LSA Implementation Reference

- Describes design standards to build BI applications founded on building blocks

LSA Operations Reference

- Describes Support Scenarios

Building Blocks
- Layer
  - Data Quality
  - Data Integration
  - ETL
- Domains
- Data Model
- Landscape
- Storage
- Ownership
- Development concept

Data Staging/Management for transactional & master data
- Persistent Objects
- Flows - scheduled/recovery
- Transformation
- Programming (Abap)
- Organization (Process Ch.)

Meta Data Management
- Naming Conventions
- Organization (InfoAreas)

Life Cycles
- Information
- Meta Object
- LSA

Administration
- Data Base
- Housekeeping
- Monitoring

Transport

Security
An EDW means managing high volumes in all areas: data store size, data loads, querying and this under often extremely challenging conditions: 24x7.

The customer LSA building blocks provide the basic framework (layers & domains) to reconcile the competing services even under extreme conditions.

The customer LSA implementation blueprint has then the task to provide standards that translate services like flexibility, throughput, robustness, completeness, comprehensiveness... into BW functionality.
Standardize data management as much as possible regardless the origin of data

- Observe 80:20 rule – first provide guidelines for core BI application requirements
- Implementations standards are developed incrementally
- Exceptions to implementation guidelines must be approved
- The more exceptions the less robustness and the higher TCO
- The bigger the expected EDW (meta data) will become the more generic the implementation must be

Anticipate growth – implementation standards must be able to manage growth

Avoid serialization of data processing – parallelize data flows

- Strategic – follow customer domain concept (general logical/ semantical partitioned implementation)
- Strategic + Tactical – expand domain concept by tactical parallelization to meet individual application requirements
- Tactical – no general domains chosen – use parallelization to meet individual application requirements
- Branch out services – observe core services an put other services aside the main data flow

Advertise & Train the idea of Customer LSA and implementation guidelines
LSA Implementation Reference
Trimming Data: Merge DataSources (Example)

© SAP 2009 / KHNC- BW Layered, Scalable Architecture (LSA) Intro, Juergen Haupt /200903/ Page 35
Filling Domains
Flow Splitting Implementation: Data Unification

- The Pass Thru DSO based split
- The early PSA-based split

Unification of data
- data management
- administration
1. What is the LSA?
2. SAP LSA Overview
   - LSA Building Blocks
     - Layers
     - Domains
   - LSA Implementation Reference
   - **LSA Operations Reference**
3. Lifecycle of the Customer LSA
Data Warehouse Performance Optimization

Recommendation Customer LSA & Housekeeping:
Use and adapt existing best practices (SAP or own) to standardize operations
LSA Operations
Potential Content Topics

Load Performance Guidelines
- InfoCubes
  - Efficient load of data into aggregates
- InfoProvider properties
- Line item & high cardenality
- Inventory
- Multi-dimensional clustering
- Compression
- Number range buffering
- DataStore Objects
  - Run time parameters
- Master data
  - Reorganization
  - Delete
- DTPs
- Flat File
- ABAP-Programming

Information Lifecycle Management
- Data archiving processes (DAPs)
- archiving data
- archiving of request data
- NLS

Housekeeping
- Delete requests of PSA
- Delete Change Log data
- Selektive delete löschen
- Delete master data & texts

Transports

Security
```

Load Balancing
- For Data Warehouse processes
- For load processes into BW
- For BW background processes
- For DataStore-Object processes

Optimization data storage

Tools for run time analysis of BW processes
1. What is the LSA?
2. SAP LSA Overview
   - LSA Building Blocks
     - Layers
     - Domains
   - LSA Implementation Reference
   - LSA Operations Reference
3. Lifecycle of the Customer LSA
Life Cycle of The Customer LSA

Step 1: Design Customer LSA

Step 2: Apply Customer LSA

Step 3: Perfect Customer LSA

Step 4: Update Customer LSA

SAP LSA: The Reference Architecture

Customer LSA: Standards - Handbook

BI Project Design

© SAP 2009 / KHNC- BW Layered, Scalable Architecture (LSA) Intro, Juergen Haupt /200903/ Page 41
Thank you!