International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

Bridging On-Premises and Cloud a Framework for HANA Smart Data Integration in Multi-Stream Data Lake Architectures

Sivasankararao Kavuri

Data Migration Lead, USA

ABSTRACT: Enterprises running SAP HANA on premises face a persistent architectural question as they build cloud-based analytics platforms: how to combine core transactional and master data held in HANA with the broader, multi-source data lake environments increasingly hosted in public cloud infrastructure. SAP HANA Smart Data Integration (SDI) provides a native mechanism for federating, replicating, and transforming data across this boundary, but its effective use in a multi-stream data lake context requires a deliberate architectural framework rather than an ad hoc set of point connections.

This article presents such a framework. It defines the core SDI architecture, describes a multi-stream data lake model spanning batch, change data capture, and near real-time streams, and proposes a layered reference architecture for bridging on-premises HANA systems with cloud data lake zones. It covers adapter strategy, data flow design patterns, flowgraph transformation practices, performance and sizing guidance, governance and lineage considerations, security architecture, monitoring, implementation methodology, cost considerations, and common risk factors.

The framework is illustrated with simple reference tables covering adapter types, data flow pattern comparisons, sizing benchmarks, and illustrative cost ranges. It concludes with best practices and an outlook on how hybrid on-premises to cloud data integration patterns were continuing to mature at the time of writing in early 2021.

KEYWORDS: SAP HANA, Smart Data Integration, Multi-Stream Data Lake, Hybrid Cloud, On-Premises Integration, Data Architecture, Real-Time Data Processing

I. INTRODUCTION

1.1 The On-Premises to Cloud Data Challenge

Most enterprises that have run SAP HANA for several years have accumulated a substantial on-premises data estate, including transactional tables, calculation views, and years of historical operational data. At the same time, analytics and data science initiatives increasingly depend on cloud-hosted data lake environments that combine this operational data with external, unstructured, and streaming sources that were never intended to live inside a transactional database.

Bridging these two environments raises several recurring questions: which data should be virtualized through federation rather than physically copied, which data should be replicated in near real time through change data capture, how data lake zones should be organized to receive both batch and streaming input from HANA alongside other sources, and how governance, lineage, and security controls should be maintained consistently across the boundary. SAP HANA Smart Data Integration was designed specifically to address the technical mechanics of this bridge, but a technical capability alone does not constitute an architecture.

1.2 Purpose and Scope

This article's purpose is to provide data architects and platform engineering teams with a structured, reference-level framework for using SDI as the integration layer between on-premises HANA systems and cloud-based, multi-stream data lake architectures. The scope covers SDI's data provisioning and adapter framework, flowgraph-based transformation, and the architectural patterns for combining federation, replication, and batch extraction. Detailed vendor-specific cloud storage service configuration is referenced only at a conceptual level, since implementation specifics vary by cloud provider and evolve independently of the SDI framework itself.

1.3 Relationship to Broader Hybrid Data Strategy

The bridging framework described in this article sits inside a larger hybrid data strategy that most enterprises pursue in some form once a meaningful cloud footprint exists alongside on-premises systems. That broader strategy typically also

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

covers cloud-native application data, third-party software-as-a-service data, and partner-provided data feeds, none of which flow through SDI. The scope of this article deliberately narrows to the on-premises HANA to cloud lake boundary, since that boundary carries a distinct set of technical mechanics, mainly through SDI, that the broader multi-source hybrid strategy does not need to address uniformly across every source type. In practice, the data domains addressed by this framework typically represent a minority of total source count in a mature multi-stream lake, but a disproportionately high share of business-critical transactional volume, since HANA-hosted systems commonly include order management, financial posting, and inventory data. This imbalance is one reason the framework treats governance and reconciliation with particular emphasis relative to lower-criticality external feeds.

II. OVERVIEW OF SAP HANA SMART DATA INTEGRATION

SAP HANA Smart Data Integration extends the HANA platform with a data provisioning framework capable of real-time and batch data movement, federation through virtual tables, and in-flight transformation through flowgraphs. It is built around three core components: the Data Provisioning Server running inside HANA, a set of adapters that connect to specific source or target systems, and, for remote or non-HANA-hosted adapters, a Data Provisioning Agent that runs outside the database and communicates back to the Data Provisioning Server.

2.1 Architecture Components

- Data Provisioning Server: the in-database service that manages adapters, remote sources, and flowgraph execution.
- Data Provisioning Agent: a lightweight, separately installed component that hosts adapters requiring connectivity outside the HANA host, commonly used for on-premises source systems.
- Adapters: source- or target-specific connectors, for example for relational databases, file systems, message queues, or cloud storage services.
- Remote Source: a configured connection instance pointing to a specific source or target system through a given adapter.
- Virtual Table: a HANA-side object that exposes a remote source table for federated query without physically copying data.
- Flowgraph: a visually designed data transformation and movement pipeline executed by the Data Provisioning Server.

2.2 Real-Time Versus Batch Replication Capabilities

SDI supports three principal modes of data movement. Federation allows a HANA query to reach a remote source directly through a virtual table without copying data at all. Batch replication periodically extracts and loads data on a scheduled basis, suitable for sources where near real-time freshness is not required. Real-time replication uses change data capture, commonly log-based for relational sources, to propagate inserts, updates, and deletes continuously with a latency typically measured in seconds rather than the minutes or hours associated with batch cycles.

2.3 Supported Source and Target Category Overview

Category	Representative Examples	Typical Mode
Relational databases	SAP ASE, Oracle, Microsoft SQL Server, IBM DB2	Batch or real-time change data capture
SAP application sources	SAP ECC, SAP S/4HANA (via ABAP adapter)	Batch or real-time change data capture
File-based sources	Delimited files, structured file shares	Batch
Message-based sources	Message queue and event-based feeds	Real-time
Cloud storage targets	Object storage services, cloud data warehouses	Batch or micro-batch
OData and web service sources	REST and OData-exposed services	Batch

Table.1. Representative SDI-supported source and target categories and their typical data movement mode

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

2.4 Licensing and Enablement Considerations

SDI is licensed as a capability within the broader SAP HANA platform, and enablement typically requires confirming that the relevant adapter entitlements are included in the organization's existing HANA licensing agreement before implementation begins. Adapter availability and supported versions have historically evolved alongside HANA platform revisions, so implementation teams should confirm compatibility between the target HANA revision, the Data Provisioning Agent version, and the specific adapters required for a given source, rather than assuming uniform support across all combinations.

- Confirm SDI and required adapter entitlements are included in the current HANA licensing agreement.
- Validate Data Provisioning Agent version compatibility against the target HANA platform revision.
- Confirm adapter-specific prerequisites, such as source database driver versions, before implementation.
- Plan for periodic agent and adapter version upgrades aligned to the broader HANA platform maintenance cycle.

III. MULTI-STREAM DATA LAKE ARCHITECTURE CONCEPTS

A multi-stream data lake architecture ingests data from several independent pipelines, each with its own latency and structural characteristics, into a common storage and processing environment. Combining HANA-sourced data with other streams requires the lake architecture to accommodate differing arrival patterns without forcing every source into a single ingestion model.

3.1 Defining Multi-Stream Ingestion

In this context, a stream refers to a distinct ingestion pipeline with a defined source, cadence, and structural format, rather than strictly a real-time event stream in the message-queue sense. A typical enterprise lake supporting a HANA-integrated architecture combines at least three stream types: change data capture streams from transactional systems, scheduled batch extracts from reporting or master data sources, and externally sourced streams such as partner feeds or unstructured content.

3.2 Lake Zone Structure

Zone	Purpose	Typical Retention
Raw / Landing	Immutable capture of source data in its native or near-native format	Extended, often indefinite for audit purposes
Curated / Conformed	Cleansed, deduplicated, and structurally conformed data ready for modeling	Aligned to business retention policy
Consumption / Serving	Aggregated or modeled data optimized for reporting and analytics access	Rolling window aligned to reporting needs

Table.2. Standard lake zone structure used to organize data arriving from multiple streams, including HANA-sourced feeds.

4.3 Comparing Batch, Change Data Capture, and Streaming Inputs

Attribute	Batch Extract	Change Data Capture	Event Streaming
Typical latency	Hours to a day	Seconds to minutes	Sub-second to seconds
Source system impact	Low, scheduled window	Low, log-based capture	Depends on producer design
Common HANA use case	Master data, reference tables	Transactional tables, order and financial data	External event feeds, sensor or clickstream data
Lake landing pattern	Scheduled file drop	Continuous micro-batch append	Continuous append or windowed aggregation

Table.3. Comparison of batch, change data capture, and event streaming ingestion characteristics within a multi-stream lake

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

IV. REFERENCE FRAMEWORK FOR BRIDGING ON-PREMISES AND CLOUD

The reference framework organizes the bridge between on-premises HANA and the cloud data lake into four layers: connectivity, provisioning, transformation, and consumption. Each layer has a distinct set of design decisions and, in a well-governed architecture, a distinct owner.

4.1 Reference Architecture Layers

- Connectivity layer: network path between the on-premises Data Provisioning Agent and cloud-hosted targets, including firewall, proxy, and encryption configuration.
- Provisioning layer: the SDI remote sources, adapters, and replication tasks that define what data moves and by which mode.
- Transformation layer: SDI flowgraphs and, where applicable, downstream cloud-native transformation jobs that shape data as it lands in the curated zone.
- Consumption layer: the modeled, aggregated data structures exposed to reporting, analytics, and data science consumers.

4.2 Connectivity and Network Considerations

The Data Provisioning Agent typically runs within the on-premises network, adjacent to the HANA system or the source systems it connects to, and initiates outbound connections toward the cloud environment. This outbound-initiated pattern generally simplifies firewall configuration relative to an inbound connection model, since it avoids opening inbound ports on the on-premises network perimeter. Bandwidth planning should account for the peak volume of the largest concurrent replication task, not only the aggregate average, since change data capture bursts can occur during batch processing windows on the source system.

4.3 Security and Data Governance Considerations

Every layer of the framework should carry its own security control rather than relying on network perimeter security alone. Connectivity should be encrypted in transit end to end. Provisioning configuration should follow least-privilege access for the technical accounts used by remote sources. Transformation logic should apply data classification and masking rules before data reaches the curated zone, and consumption layer access should be governed through role-based entitlements aligned to the sensitivity of the underlying data.

4.4 Illustrative Reference Architecture Walkthrough

The table below maps each of the four framework layers introduced in Section 5.1 to representative technology choices and the primary control owned at that layer, illustrating how the framework translates into a concrete implementation.

Layer	Representative Technology	Primary Control Owned
Connectivity	Data Provisioning Agent, VPN or dedicated network link, TLS encryption	Network path security and availability
Provisioning	SDI remote sources, adapters, replication tasks	Data movement mode and source access scope
Transformation	SDI flowgraphs, downstream cloud-native transformation jobs	Data shaping, masking, and classification
Consumption	Cloud data warehouse or lakehouse query layer, reporting semantic models	Role-based access to modeled data

Table.4. Illustrative mapping of framework layers to representative technology and primary ownership.

This mapping is intentionally technology-general, since the specific cloud storage or warehouse service selected varies by organization and evolves independently of the SDI-side provisioning and transformation layers. The framework's value lies in keeping the four layers, and their respective owners, distinct even as the underlying technology choices change over time.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

V. SDI ADAPTER STRATEGY

5.1 Adapter Selection Criteria

Criterion	Consideration
Latency requirement	Determines whether change data capture or scheduled batch is appropriate
Source system load sensitivity	Log-based capture minimizes load; query-based polling increases source load
Data volume and change rate	High change-rate tables favor change data capture over full batch reload
Target compatibility	Cloud storage and warehouse targets vary in supported adapter versions
Operational maturity of the adapter	Long-established adapters generally carry lower implementation risk

Table.5. Selection criteria for choosing an SDI adapter and replication mode for a given source.

5.2 Real-Time Change Data Capture Configuration

Log-based change data capture reads database transaction logs rather than querying source tables directly, which minimizes load on the source system and captures inserts, updates, and deletes with low latency. Configuration typically requires enabling supplemental logging or an equivalent capture mechanism on the source database, defining the initial load strategy for existing data, and configuring the ongoing capture task to append changes to the target continuously.

- Confirm source database supports log-based capture and that required logging is enabled before initial load.
- Perform the initial full load during a low-activity window to minimize contention with production traffic.
- Validate that the initial load and the ongoing change capture are reconciled so no transactions are missed at the handover point.
- Monitor replication latency continuously rather than only at initial go-live.

5.3 Cloud Storage and Warehouse Targets

Cloud object storage and cloud data warehouse targets are typically written to through a combination of an SDI file or database adapter and a landing convention that partitions data by source table and load date. This partitioning convention should be defined once, at the framework level, so that downstream curated-zone transformation jobs can rely on a consistent landing structure regardless of which source system produced the data.

VI. DATA FLOW DESIGN PATTERNS

Three data flow patterns cover the great majority of practical bridging scenarios between HANA and a multi-stream data lake, and most architectures use a combination of all three across different data domains.

6.1 Pattern 1: Direct Federation via Virtual Tables

Federation exposes HANA data to lake-side consumers, or exposes lake-side data to HANA, without physically copying it, through a virtual table over a remote source. This pattern suits low-volume, infrequently accessed reference data where the overhead of maintaining a replicated copy is not justified by the access pattern.

6.2 Pattern 2: Real-Time Replication to the Data Lake

Real-time replication continuously propagates transactional changes from HANA into the raw landing zone of the data lake, typically using log-based change data capture. This pattern suits high-value transactional domains, such as order or financial data, where downstream analytics require near current data.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

6.3 Pattern 3: Batch Extract with Flowgraph Transformation

Batch extraction, combined with an SDI flowgraph that applies filtering, joins, or aggregation before landing data in the lake, suits master data and reference domains where daily or intraday freshness is sufficient and where some transformation is more efficiently performed close to the source.

6.4 Pattern Comparison

Attribute	Federation	Real-Time Replication	Batch with Flowgraph
Data freshness	Always current at query time	Near current, seconds to minutes	Aligned to batch schedule
Source system load	Per query, can be significant at scale	Low, log-based	Moderate, scheduled window
Best-fit data domain	Low-volume reference data	High-value transactional data	Master and reference data with transformation needs
Lake storage footprint	Minimal, no physical copy	Continuous append growth	Scheduled incremental growth

Table.6. Comparison of the three principal SDI-based data flow patterns for bridging HANA and the data lake.

6.5 Selecting a Pattern by Data Domain

The table below applies the comparison in Table 5 to four representative data domains commonly found in a HANA-sourced multi-stream lake, illustrating how pattern selection follows directly from the freshness, volume, and sensitivity classification introduced in Section 13.

Data Domain	Recommended Pattern	Rationale
Financial posting transactions	Real-time replication	High business value, requires near current visibility
Customer or vendor master data	Batch with flowgraph	Low change rate, benefits from conformance transformation
Reference and lookup tables	Federation	Low volume, infrequently accessed, avoids replication overhead
Inventory and order transactions	Real-time replication	High change rate, direct operational analytics dependency

Table.7. Illustrative pattern selection by representative data domain

VII. DATA TRANSFORMATION AND FLOWGRAPH DESIGN

7.1 Flowgraph Components

- Source and target nodes representing the remote sources and destinations participating in the flow.
- Projection and filter nodes that select and restrict columns and rows in transit.
- Join and union nodes that combine multiple sources within a single flowgraph.
- Aggregation nodes that summarize data before it lands in the target.
- Data quality and cleansing transforms, where SAP HANA Smart Data Quality capabilities are also licensed.

7.2 Transformation Best Practices

- Keep flowgraphs focused on a single logical data domain rather than combining unrelated sources in one flow.
- Push filtering as early as possible in the flowgraph to minimize the volume of data carried through later stages.
- Document the business logic embedded in each transformation node, since flowgraph logic is otherwise opaque to downstream consumers.
- Version-control flowgraph exports alongside other data platform configuration artifacts.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

- Separate cleansing and structural transformation from business-rule transformation where practical, to simplify future maintenance.

VIII. PERFORMANCE AND SCALABILITY CONSIDERATIONS

8.1 Sizing Guidance

Component	Illustrative Sizing Driver	Illustrative Range
Data Provisioning Agent host	Concurrent replication task count	4 to 8 vCPU, 16 to 32 GB memory
Network bandwidth (on-premises to cloud)	Peak concurrent change data capture volume	100 Mbps to 1 Gbps dedicated
HANA memory allocation for SDI workload	Number and complexity of concurrent flowgraphs	5 to 15 percent of total HANA memory
Initial load batch window	Volume of historical data for first load	Sized to complete within a defined maintenance window

Table.8. Illustrative sizing drivers and ranges for core SDI components. Actual sizing should be validated against workload-specific testing.

8.2 Illustrative Throughput Benchmarks

Replication Mode	Illustrative Throughput	Typical Constraint
Log-based change data capture	5,000 to 20,000 changed rows per second	Network bandwidth and agent CPU
Batch full load	1 to 5 million rows per hour	Source query performance and target write throughput
Flowgraph with in-flight transformation	30 to 60 percent of raw replication throughput	Transformation complexity and node count

Table.9. Illustrative throughput benchmarks by replication mode, compiled from published implementation guidance current as of early 2021.

8.3 Scalability Patterns for Growing Source Counts

As the number of registered remote sources and active flowgraphs grows, a single Data Provisioning Agent installation eventually becomes a scaling constraint, both in terms of host resource capacity and in terms of operational blast radius, since a single agent outage can interrupt every source it hosts simultaneously. Programs anticipating growth beyond an initial pilot scope should plan for a multi-agent topology from the outset, rather than scaling a single agent host indefinitely.

- Group remote sources across multiple agents by business domain or criticality tier rather than by arbitrary assignment.
- Reserve a dedicated agent, or an isolated resource pool, for the highest-criticality real-time replication tasks.
- Monitor per-agent resource utilization independently rather than relying on an aggregate fleet-level view alone.
- Revisit agent topology at each major milestone when source count grows beyond an agreed threshold, for example every 20 additional remote sources.

IX. GOVERNANCE, METADATA, AND LINEAGE

9.1 Metadata Cataloging

Every remote source, virtual table, and flowgraph registered in SDI represents a metadata asset that should be captured in an enterprise data catalog, not only in the HANA-side configuration repository. This ensures that data lake consumers can discover the origin and structure of HANA-sourced data without needing direct access to the source HANA system.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

9.2 Data Lineage Tracking

Lineage should be tracked at three levels: source-to-landing lineage capturing which remote source and replication task produced a given raw-zone data set, landing-to-curated lineage capturing which flowgraph or downstream transformation job produced a curated data set, and curated-to-consumption lineage capturing which reporting or analytics model consumed a given curated data set. Capturing all three levels allows an impact analysis to trace a reporting anomaly back to a specific source system change.

9.3 Data Quality Monitoring

Governance is incomplete without an explicit data quality dimension layered on top of lineage and cataloging. Where SAP HANA Smart Data Quality capabilities are licensed alongside SDI, quality rules such as completeness, uniqueness, and referential validity checks can be embedded directly into flowgraphs. Where those capabilities are not licensed, equivalent checks should be implemented as explicit validation steps in the curated-zone transformation layer rather than omitted.

Quality Dimension	Representative Check	Illustrative Target
Completeness	Percentage of expected records present versus source count	99.5 percent or higher
Uniqueness	Duplicate key rate in the curated zone	Below 0.1 percent
Referential validity	Foreign key match rate against reference tables	99 percent or higher
Timeliness	Replication or batch landing time versus service level	Within defined latency target

Table.10. Core data quality dimensions monitored in the curated zone, with illustrative targets.

X. SECURITY ARCHITECTURE

10.1 Authentication and Encryption

- Use dedicated technical accounts with least-privilege access for every SDI remote source connection.
- Enforce encrypted connections for all agent-to-server and agent-to-cloud traffic.
- Rotate technical account credentials on a defined schedule aligned to enterprise security policy.
- Restrict Data Provisioning Agent host access to a limited set of platform administrators.

10.2 Data Masking and Anonymization

Sensitive fields identified during data classification should be masked or tokenized in the transformation layer before data reaches the curated zone, rather than relying on consumption-layer access control alone. This reduces the exposure surface for sensitive data that may otherwise persist in the raw landing zone in its original form.

XI. MONITORING AND OPERATIONS

11.1 Core Monitoring Metrics

Metric	Purpose	Illustrative Alert Threshold
Replication latency	Detect change data capture falling behind source activity	Above 5 minutes sustained
Agent host CPU and memory utilization	Detect resource contention on the Data Provisioning Agent	Above 80 percent sustained
Flowgraph execution failure rate	Detect transformation job reliability issues	Any failure on a scheduled flowgraph
Row count reconciliation variance	Detect data loss or duplication between source and target	Any variance beyond an agreed tolerance

Table.11. Core monitoring metrics for an SDI-based bridging architecture, with illustrative alert thresholds.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

11.2 Alerting and Incident Response

Alerting should distinguish between a transient latency spike, which may self-resolve within minutes, and a sustained replication failure, which requires operational intervention. A tiered alerting policy, similar in spirit to the escalation structures used in test and defect management contexts, helps avoid alert fatigue while still surfacing genuine incidents promptly.

11.3 Sample Operational Dashboard Extract

The extract below illustrates the level of detail appropriate for a daily operational dashboard reviewed by the platform team, combining the metrics defined in Table 8 with a simple status flag.

Remote Source	Replication Mode	Current Latency	Status
Order Management (HANA)	Real-time CDC	42 seconds	Normal
Financial Postings (HANA)	Real-time CDC	3 minutes 10 seconds	Watch
Vendor Master (HANA)	Batch, daily	Completed on schedule	Normal
Inventory Transactions (HANA)	Real-time CDC	58 seconds	Normal
Partner Feed (External)	Batch, hourly	Completed on schedule	Normal

Table.12. Sample daily operational dashboard extract for an SDI-based bridging pipeline.

XII. IMPLEMENTATION METHODOLOGY

The following sequence reflects a practical, project-tested order of operations for implementing an SDI-based bridge between an on-premises HANA system and a cloud multi-stream data lake.

1. Inventory candidate source tables and classify each by required freshness, volume, and sensitivity.
2. Select the data flow pattern, federation, real-time replication, or batch with flowgraph, for each classified data domain.
3. Provision and size the Data Provisioning Agent host and confirm network connectivity to the cloud target.
4. Configure remote sources and adapters for each in-scope source system.
5. Enable change data capture logging on source databases selected for real-time replication.
6. Design and test flowgraphs for domains requiring in-flight transformation.
7. Execute an initial full load during a defined low-activity window and reconcile against the source.
8. Enable ongoing replication and validate latency against the target service level.
9. Register all remote sources, flowgraphs, and lineage metadata in the enterprise data catalog.
10. Establish monitoring dashboards and alerting thresholds before promoting the pipeline to production use.
11. Conduct a structured cutover review confirming reconciliation, security controls, and monitoring are all active.

12.1 Illustrative Phase Timeline

The table below provides an illustrative duration range for each methodology phase in a mid-size implementation covering approximately 15 to 25 remote sources across a mix of the three data flow patterns described in Section 7.

Phase	Illustrative Duration
Domain inventory and classification	2 to 3 weeks
Agent provisioning and connectivity setup	1 to 2 weeks
Adapter and remote source configuration	3 to 5 weeks
Flowgraph design and testing	3 to 6 weeks
Initial load, reconciliation, and cutover	2 to 4 weeks
Monitoring and catalog registration	1 to 2 weeks, overlapping prior phases

Table.13. Illustrative phase timeline for a mid-size SDI bridging implementation

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

XIII. RISK FACTORS AND MITIGATION

Risk Factor	Potential Impact	Mitigation
Undersized Data Provisioning Agent host	Replication latency and task failures under peak load	Size against peak concurrent volume, not average
Missing reconciliation between initial load and change capture	Silent data loss at cutover	Formal reconciliation step before enabling ongoing replication
Unclassified sensitive data reaching the raw zone unmasked	Compliance and privacy exposure	Classify and mask in the transformation layer before landing
Flowgraph sprawl without documentation	Unmaintainable transformation logic over time	Enforce documentation and version control standards
Network bandwidth contention with other on-premises to cloud traffic	Unpredictable replication latency	Dedicated or prioritized bandwidth allocation for replication traffic

Table.14. Common risk factors in SDI-based bridging architectures, with corresponding mitigations.

XIV. COST CONSIDERATIONS

Cost drivers for an SDI-based bridging architecture fall into four categories: licensing, infrastructure, network, and operational effort. The table below summarizes illustrative relative cost weight across these categories for a mid-size implementation.

Cost Category	Primary Driver	Illustrative Relative Weight
SDI and HANA licensing	Data volume and adapter entitlements	Moderate to high, one-time and ongoing
Data Provisioning Agent infrastructure	Host sizing and redundancy requirements	Low to moderate, ongoing
Network bandwidth	Peak replication volume and cloud data egress	Moderate, ongoing
Operational and maintenance effort	Number of active flowgraphs and remote sources	Moderate to high, ongoing

Table.15. Illustrative relative cost weight by category for a mid-size SDI bridging implementation.

14.1 Illustrative Cost Comparison: Point-to-Point Versus Framework-Based Bridging

A common alternative to the framework presented in this article is an ad hoc set of point-to-point integrations built independently for each new source as the need arises. The comparison below illustrates why the incremental cost curve of that approach tends to diverge unfavorably from a framework-based approach as source count grows.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

Source Count	Point-to-Point Illustrative Relative Cost	Framework-Based Illustrative Relative Cost
5 sources	1.0x baseline	1.3x baseline (framework setup overhead)
15 sources	2.8x baseline	1.8x baseline
30 sources	5.6x baseline	2.4x baseline

Table 16. Illustrative relative cost comparison between point-to-point and framework-based bridging as source count grows.

The framework-based approach carries a higher relative cost at very small scale, reflecting the upfront investment in the four-layer structure defined in Section 5, but the incremental cost of each additional source declines significantly once the framework is established, since connectivity, governance, and monitoring capabilities are shared across sources rather than rebuilt for each one.

XV. INDUSTRY ADOPTION PATTERNS

The patterns below are drawn from anonymized, generalized implementation experience. No specific organization is identified.

15.1 Manufacturing and Supply Chain Pattern

Manufacturing organizations frequently prioritize real-time replication of order and inventory transactional data into the lake, combined with batch extraction of master data such as material and vendor records, reflecting the differing volatility of these two data domains.

15.2 Retail and Consumer Pattern

Retail organizations commonly combine real-time replication of point-of-sale and order data with externally sourced streaming data, such as clickstream or partner feed data, landing both in a shared raw zone before conforming them together in the curated zone.

15.3 Financial Services Pattern

Financial services organizations tend to apply the most stringent masking and lineage controls in the transformation layer, given regulatory sensitivity, and frequently restrict real-time replication to a narrower set of pre-approved transactional domains rather than broadly enabling change data capture across all source tables.

15.4 Cross-Industry Summary

Despite differing regulatory and operational contexts, the three patterns above share a common underlying logic: real-time replication is reserved for the transactional domains with the highest business value and volatility, batch extraction with transformation handles master and reference data, and governance controls, particularly masking, tighten in proportion to regulatory sensitivity rather than applying uniformly across all industries.

Industry Pattern	Dominant Data Flow Pattern	Governance Emphasis
Manufacturing and supply chain	Real-time replication for orders and inventory	Standard, aligned to Section 11
Retail and consumer	Real-time replication combined with external streaming	Standard, with added focus on external feed validation
Financial services	Restricted real-time replication, stronger masking	Elevated, per Section 11.2

Table.17. Cross-industry summary of dominant data flow pattern and governance emphasis by sector.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

XVI. COMMON PITFALLS

- Treating every source table as a candidate for real-time replication rather than classifying by actual freshness need.
- Underestimating Data Provisioning Agent sizing based on average rather than peak concurrent volume.
- Allowing flowgraph transformation logic to accumulate without documentation or version control.
- Omitting formal reconciliation between the initial load and the start of ongoing change data capture.
- Applying data masking only at the consumption layer instead of in the transformation layer.
- Neglecting to register SDI metadata in the enterprise data catalog, leaving lineage undiscoverable to lake consumers.

XVII. BEST PRACTICES AND RECOMMENDATIONS

- Classify every candidate data domain by freshness, volume, and sensitivity before selecting a data flow pattern.
- Default to log-based change data capture for high-value transactional domains rather than frequent batch reloads.
- Size the Data Provisioning Agent against peak concurrent replication volume, and revisit sizing as scope grows.
- Apply data classification and masking in the transformation layer, not only at the point of consumption.
- Maintain a consistent landing zone partitioning convention across all sources feeding the data lake.
- Register every remote source, flowgraph, and lineage relationship in the enterprise data catalog.
- Establish monitoring and alerting thresholds before promoting any pipeline to production use.
- Reconcile initial load against source counts before enabling ongoing replication for any table.
- Review flowgraph documentation and version control compliance on a regular cadence, not only at initial build.

XVIII. FUTURE OUTLOOK

As of early 2021, enterprise data platform strategies were increasingly converging on a hybrid model in which core transactional systems, including on-premises HANA, remain in place while analytics and data science workloads expand into cloud-hosted lake and lakehouse architectures. This trend places growing emphasis on integration frameworks capable of supporting multiple data movement modes consistently, rather than relying on a single replication technique for all data domains. Continued investment in metadata cataloging, automated lineage capture, and consistent governance across the on-premises to cloud boundary is expected to remain a priority as multi-stream lake architectures grow in scale and source diversity.

XIX. CONCLUSION

Bridging on-premises HANA systems with cloud-based, multi-stream data lake architectures is not solely a matter of selecting the right adapter or replication mode. It requires a layered framework that addresses connectivity, provisioning, transformation, and consumption consistently, with governance, security, and monitoring built into each layer rather than added afterward.

The framework presented in this article, spanning adapter strategy, data flow design patterns, flowgraph transformation practices, performance sizing, governance, and security architecture, gives data platform teams a structured starting point for extending SAP HANA Smart Data Integration beyond point-to-point replication into a coherent, multi-stream bridging architecture

REFERENCES

1. Few, S. (2013). Information Dashboard Design: Displaying Data for At-a-Glance Monitoring (2nd ed.). Analytics Press.
2. Forrester Research. (2019). The Forrester Wave: Data Lakes, Q2 2019. Forrester Research, Inc.
3. Gartner, Inc. (2019). How to Build an Effective Data Lake Strategy. Gartner Research.
4. Gartner, Inc. (2020). Market Guide for Data Integration Tools. Gartner Research.
5. Heilman, R., Sanyal, J., Ravi, R., & Yang, J. (2017). SAP HANA Smart Data Integration and Smart Data Quality. SAP PRESS.
6. IDC. (2020). Worldwide Data Integration and Intelligence Software Market Forecast, 2020 to 2024. IDC.
7. ISO/IEC. (2017). ISO/IEC 38505-1:2017, Governance of Data. International Organization for Standardization.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

8. Inmon, B. (2016). Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump. Technics Publications.
9. Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.). Wiley.
10. Loshin, D. (2010). The Practitioner's Guide to Data Quality Improvement. Morgan Kaufmann.
11. SAP SE. (2018). SAP HANA Data Provisioning Guide. SAP Help Portal.
12. SAP SE. (2019). SAP HANA Smart Data Integration and Smart Data Quality: Administration Guide. SAP Help Portal.

Appendix A: Glossary of Key Terms

Term	Definition
SDI	SAP HANA Smart Data Integration, a data provisioning and transformation framework.
Data Provisioning Agent	A separately installed component hosting adapters requiring connectivity outside the HANA host.
Virtual Table	A HANA object exposing a remote source table for federated query without physical replication.
Flowgraph	A visually designed data transformation and movement pipeline executed by SDI.
Change Data Capture (CDC)	A technique for continuously propagating inserts, updates, and deletes from a source system.
Data Lake Zone	A logical storage area within a data lake, such as raw, curated, or consumption.
Data Lineage	A traceable record of the origin and transformation history of a given data set.
Data Masking	A technique for obscuring or tokenizing sensitive data fields to reduce exposure risk.

Table.18. Glossary of key terms referenced throughout this article.

Appendix B: Sample SDI Implementation Checklist

- Source data domains inventoried and classified by freshness, volume, and sensitivity.
- Data flow pattern selected for each classified domain, with rationale documented.
- Data Provisioning Agent host sized and provisioned, with network connectivity confirmed.
- Remote sources and adapters configured and tested against a non-production target.
- Change data capture logging enabled on source databases selected for real-time replication.
- Flowgraphs designed, documented, and version-controlled for domains requiring transformation.
- Initial load executed and reconciled against source counts before ongoing replication is enabled.
- Data classification and masking rules applied in the transformation layer for sensitive fields.
- Remote sources, flowgraphs, and lineage relationships registered in the enterprise data catalog.
- Monitoring dashboards and alert thresholds established prior to production cutover.
- Cutover review completed, confirming reconciliation, security controls, and monitoring are active.

Appendix C: Sample Flowgraph Node Reference

The reference table below summarizes the flowgraph node types introduced in Section 8.1, together with a brief usage note for each.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 3, Issue 2, March-April 2021 |

DOI:10.15662/IJEETR.2021.0302007

Node Type	Usage Note
Source	Reads from a configured remote source or virtual table
Target	Writes to a configured remote target, such as a cloud storage or warehouse destination
Projection	Selects and renames a subset of columns
Filter	Restricts rows based on a defined condition
Join	Combines rows from two or more inputs based on a join condition
Union	Appends rows from two or more inputs with a compatible structure
Aggregation	Summarizes rows using grouping and aggregate functions
Data Quality Transform	Applies completeness, uniqueness, or validity checks, where licensed

Table.19.Sample flowgraph node type reference