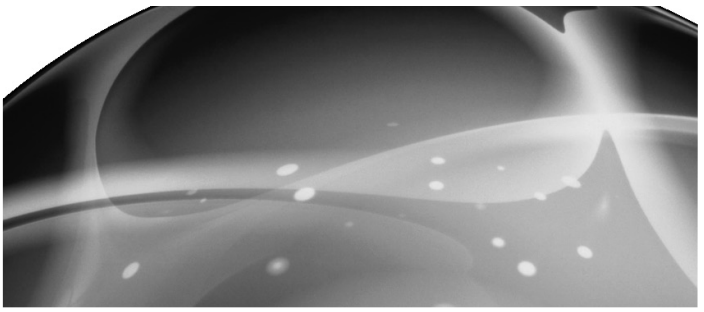
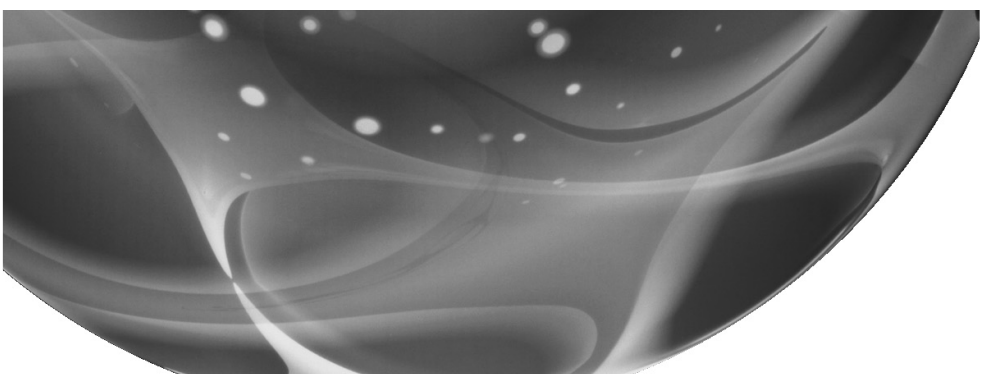


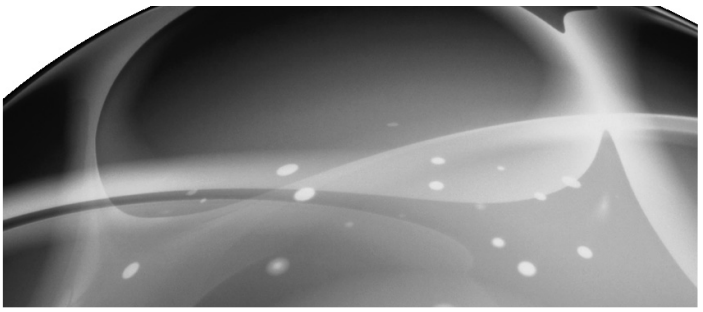
PART IV

Implementing Customer Data Integration for the Enterprise



Up to this point, our discussion has been focused on issues and concerns surrounding Master Data Management and Customer Data Integration. We took a close look at the business and technology drivers for MDM and CDI, discussed architecture considerations and design concerns, and offered an architectural approach to addressing some of the key challenges presented by the MDM-CDI goal of integrating data across the enterprise, and especially the challenges related to data security and visibility.

This part of the book deals with the practical aspects of implementing MDM-CDI solutions as complex, multidisciplinary, enterprise-wide projects or programs. The complexity and multidisciplinary nature of MDM-CDI systems is such that implementing these solutions requires an approach that may be even broader than a single project, and should be managed as an initiative-level, enterprise-wide program that consists of multiple projects, phases, steps, tasks, and activities that are organized into a cohesive set of concurrent but interdependent work streams. The terms "project," "program," and "initiative" are often defined according to the rules, structures, processes, and standards of an individual organization, and are often treated as synonyms. Therefore, to avoid confusion related to the appropriate use of these terms, we will use the familiar term "project" when discussing implementation concerns of MDM-CDI systems.



CHAPTER 11

Project Initiation

IN THIS CHAPTER

Implementation Begins

Let us assume at this point that the business drivers, objectives, and value propositions for the MDM-CDI project have been established and agreed upon within the organization. This agreement should result in senior management's issuing marching orders to the information technology organization to proceed with the project. Senior management wants to know how the project will be organized and planned—major milestones and releases. Senior management will also want to know how much this project would cost to develop and deploy. What should the IT organization do to translate business objectives into IT vision, strategy, actionable road map, and resource-loaded project plans, in order to successfully implement the project? Senior management will want to see the end-state vision across all business and technology domains, and will ask a number of key questions such as how the currently ongoing projects should be modified, aligned, and prioritized in the context of the MDM-CDI project. These and other critical questions are discussed in this chapter.

One of the key management challenges of MDM-CDI projects is the need to define the project's success criteria. When asked informally, IT executives may provide very different answers to the question of how to decide if the project is successful, including answers such as "Have better quality data," "Improve application functionality," "Achieve regulatory compliance faster," and "Keep the end users happy." In practice, MDM-CDI projects should follow a capabilities road map with a sound release strategy that defines what will be implemented and deployed, and when. A key component of this road map is clearly defined short-term and long-term success criteria that need to be understood and agreed upon by all stakeholders.

Of course, the tasks of defining the road map and success criteria would be so much easier to accomplish if there were a single strategy and solution architecture that works for all MDM-CDI implementations. Unfortunately, that is not the case. As we mentioned earlier, MDM-CDI projects address multiple, often diverse business requirements, involve a broad spectrum of technical disciplines, and can be extremely complex.

At a high level all MDM-CDI projects are quite similar. For example, all CDI projects have common goals of delivering an authoritative system of record for customer data that includes a complete, 360-degree view of customer data including the totality of the relationships the customer has with the organization. At this high level, project management and all key stakeholders are enthusiastic and are in "violent" agreement about the capabilities and opportunities offered by a CDI system.

The devil is in the details. And there are *many* details to consider. At the beginning and in the course of an MDM-CDI project, the wide variety of questions, issues, and dependencies may appear so overwhelming that the initiative's stakeholders feel like the fishermen in the movie "Perfect Storm." The characters of the film were practically

Key Senior Management Concerns

- How will the equity value and market capitalization of the company change as a result of the MDM-CDI project?
- What is the first set of applications and business functions that would become the early adopters and beneficiaries of the MDM-CDI project?
- What is the phase-by-phase and total project cost/ROI?
- How will the new business processes be different?
- What will it take to accomplish the transition to the new business processes?
- What additional skills, both business and technology, will the organization's staff members have to acquire and how?
- Will the organizational structure be affected, and to what extent?
- What legacy systems and functions will be affected and how?
- How will the project be organized and planned?
- What are the major milestones and releases?
- Does the technology organization have adequate knowledge, resources, and understanding of industry best practices in order to translate business objectives into IT vision, strategy, actionable road map, and project plans, and successfully implement the project?
- What are the investment and delivery risks and mitigation strategies?
- What is the end-state vision that would impact business and technology domains?
- How should the current in-flight initiatives be modified, aligned, and prioritized in light of starting a CDI project?
- What are the success criteria for each phase and for the project overall?

helpless before the fury of the ocean. The situation may seem similarly unmanageable for the participants of some MDM-CDI projects. Many large MDM-CDI projects failed with dire consequences for the company and people who worked on the project. We will discuss risks and reasons for project failure in Chapter 18.

In short, due to the variety of conditions and the complexity of the MDM-CDI projects, it is difficult to define a single one-size-fits-all set of recommendations. Experience shows that an effective working approach to complex and diverse problems like the one presented by MDM-CDI is to define a comprehensive solution framework that is designed around sound problem-solving architecture principles including separation of concerns, layered architecture, federation, and service orientation (please see Part II for more details on the architecture framework and design principles). Such a framework allows us to use industry best practices for particular areas of concern, and to break down the problem domain into smaller and more manageable pieces. At a more granular level the tasks and decision-making points are much more common and manageable across MDM-CDI projects. We will follow this approach and break down the CDI problem domain into work streams and components that support and are supported by what we define as the CDI "ecosystem." The areas of concerns and key components that constitute a CDI "ecosystem" are shown in Figure 11-1. We discuss most of them throughout this book in more detail.

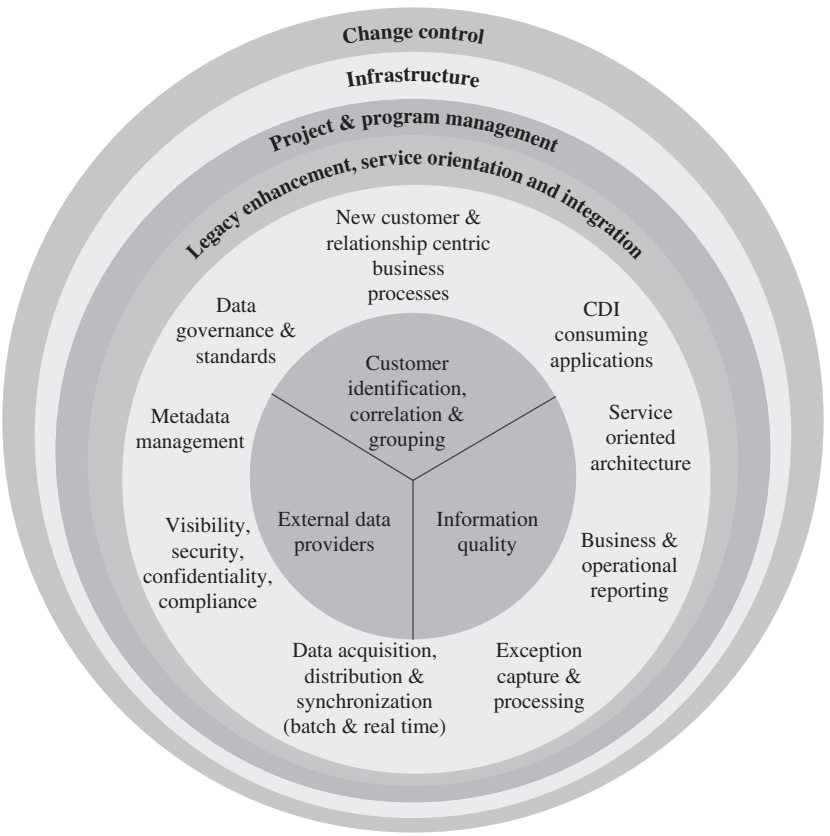


Figure 11-1 *CDI "ecosystem"— high-level areas of the solution*

The CDI "ecosystem" is a layered construct that includes business processes and technical domains of change. The core CDI functional area includes:

- Customer identification, matching, correlation, hierarchies, relationships, and grouping
- Information quality
- External data providers

The layer immediately surrounding the CDI core is a key part of the CDI "ecosystem" even though it includes some components that support both CDI and non-CDI environments. We discuss various aspects of these components throughout the book.

As we continue to "peel" the layers of the CDI "ecosystem," we can see the components and services that do not necessarily represent "pure" CDI functionality, but are affected and/or required by the CDI solution to function. For example, the legacy layer demonstrates these dual properties—it is usually the source of CDI data and is a key consumer of new functionality enabled by the CDI platform.

Similarly, the outer layers of the CDI "ecosystem" cover Infrastructure, Project/Program Management, and Change Control. These areas of concern are vital for any successful CDI implementation. Indeed, it is hard to imagine a project of MDM-CDI magnitude that can be successful without considering infrastructure issues or providing adequate program management and change control.

CDI-related areas of the "ecosystem" contain components that have to be acquired or built. Thus, the CDI "ecosystem" also provides a framework for "buy vs. build" decisions. These decisions are not easy to make, especially considering that many vendor products in the CDI space overlap, and the resulting market focus and positioning of many CDI vendors continues to change. For example, ETL and data synchronization vendors are moving into the Information Quality space, and Information Quality vendors are extending their capabilities towards CDI Data Hubs. The discussion of the vendor landscape and their product features can be found in Chapter 17.

The complexity of the MDM-CDI problem space requires participation of multiple stakeholders. This in turn, creates a formidable socialization problem. The idea of bringing all people onto the same page on all issues can easily paralyze any initiative. While consensus building is a good strategy, we need to remember that both unlimited democracy and military-style decision making can cause large initiatives to fail. Best practices suggest to set up a small leadership group that can successfully combine principles of strong management control and decision making with principles of sharing information and keeping all participants on common ground in key areas.

Workshop Agenda

- State project goals and senior management concerns.
- Define and agree on the end-state of the solution and what it means from both business and technology points of view.
- Discuss the types of partners and vendors that will be required to achieve the objectives.
- Define and socialize the project's organizational structure and project governance.
- Discuss a high-level road map.
- Determine project success criteria.
- Analyze cost benefits of the project.
- Discuss the timeline, content, and the deliverables of the project on a phase-by-phase basis.
- Build consensus between business and technology teams as the decision on "What should the first release look like?" is made.

Of course, every time you want diverse groups such as business and technology teams to agree, you have to overcome the challenge of each group looking at the problem from its own perspective. Using the comprehensive solution framework described throughout this book should help the reader to manage the discussions about the project and to make decisions that each project team can be comfortable with.

It is a good idea to start an MDM-CDI project with a well-structured workshop where business vision and technology approach will be discussed. Two or three days spent as a team can be very beneficial to jump-start the project by getting high-level agreement on a number of key issues.

Using a multiphased approach to a CDI project is clearly a good strategy. The outcome of the first phase (first release of the CDI solution) should be thought of as a trade-off between what the business ultimately needs and what is achievable in a single release. A practical rule of thumb is that each phase should not exceed six to eight months and should deliver tangible, business-recognizable benefits. Credibility of the project will be at stake if a year has gone by and no changes have been implemented. Enterprise-level planning should be in place to ensure successful cross-departmental delivery.

When a CDI project is initiated, IT group should have access to a business-sponsored document that defines business case by business function and line of business. This business requirements document should include the following:

- Formulation of business problems as they relate to MDM-CDI
- Definition of the business scope including articulations of the new business processes
- Strategic business vision and objectives
- Business drivers and priorities
- ROI estimation for MDM-CDI implementation

Considering the complexity and potential breadth of the impact a CDI solution may have on the organization, defining the scope is one of the key factors that determines the actual or perceived success or failure of the CDI project.

Scope Definition

When we discuss CDI projects, we should realize that the scope of these projects is a multidimensional matter. The most important dimensions are shown in Figure 11-2 and discussed in this section.

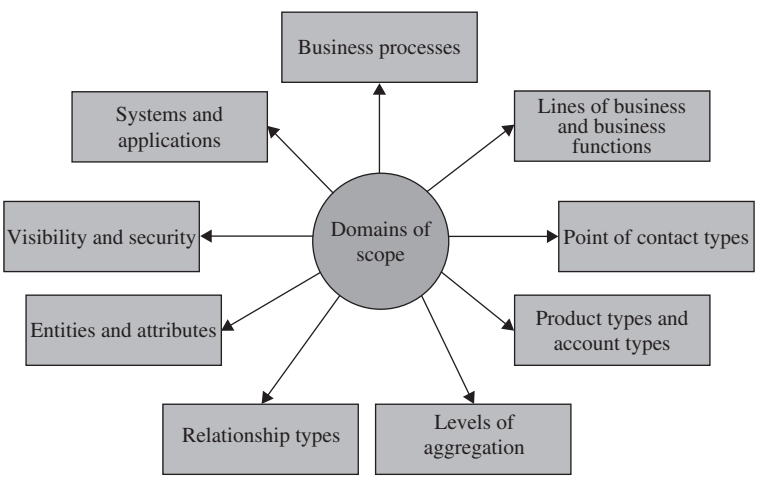


Figure 11-2 *Domains of Scope*

Business Processes

It is critically important to understand what business processes need to be improved. Without a clear understanding of what business processes will be improved and how, the entire effort may turn into a costly and most likely useless endeavor. Starting from the business objectives, drivers, and value propositions, the technology group should work with the business team to document the current state of the business processes that need to be improved. There are many methodologies on how to define and document business processes. Rational Unified Process (RUP) is one of the better-known methodologies used for this purpose. RUP uses a concept of use cases as the core of business process analysis and definition.

Once the current state of the business processes and their weaknesses are defined and documented, the target-state business processes need to be determined. There are many techniques and methodologies on business process improvement and re-engineering. For example, see *Business Process Management: Practical Guidelines to Successful Implementations* by John Jeston and Johan Nelis (Butterworth-Heinemann, 2006).

What level of granularity in the business process definition is sufficient? If you need to describe the process of lawn mowing you can do it at a high level, e.g., get the lawn mower, turn it on, mow the lawn, clean the lawn mower, and put it back. Alternatively, you can describe this process with more granularity and show decision points, e.g. what happens if you do not have enough gas, or what you do if the device breaks, or how do you trim corners that cannot be reached by the lawn mower. The required granularity of the business process should be at the level of detail that is sufficient to define the logical data model of the CDI solution. This brings us to an important point. The data modelers should work closely with the business process development team to provide input and pose questions driving the depth of the business process coverage.

Lines of Business and Functions

One of the outcomes of the creation and analysis of the business requirements for the project is the determination of new and impacted legacy business processes. At the same time, the requirements should specify the needs of key lines of business and business functions that drive the change. At that point, the requirements analysis should determine whether there are other lines of business that can benefit from the CDI-driven change. The perspective of these other lines of business is important for the enterprise to understand that additional benefits can be realized from implementing the CDI project. These additional requirements and benefits may strengthen the CDI business case and its value proposition.

In addition, a comprehensive view presented by all lines of business and functions early in the project life cycle can help understand constraints that otherwise would

be revealed at later phases of the project with additional risks and costs always accompanying late changes in project planning and execution.

An understanding of all impacted lines of business should also help in building a project team that adequately represents all interested parties.

Customer Touch Points, Product Types, and Account Types

Modern enterprises frequently have multiple channels that support various customer touch points. For instance, customers interact with a hotel chain by phone, personally in the hotel lobby, online over e-mail and the Internet, by mail, etc. The same channels may provide additional touch points for customers who participate in hotel membership clubs or are hotel credit card holders. Similar variety in the touch points exists in the financial services and other industries.

Analyzing the channel and touch-point requirements helps bring into focus additional perspectives and questions that can impact the scope of the CDI project. Specifically, the customer data presented at different touch points may vary significantly, and as the result, may impact the identification and matching process, data visibility, and security approaches. We will discuss customer identification and matching in more detail in Chapter 12. Visibility and security are discussed in depth in Part III of the book.

A typical enterprise normally offers and serves many product types to its customers. In the financial services industry, for example, products are often linked to or represented by account types such as Wealth Management Account, Cash Management Account, 401K Retirement Account, etc. Other industries also have a strong emphasis on products. For example, a telecommunications company can offer and provide local, long distance, and international phone service, DSL, wireless connectivity services, satellite or cable TV services, etc. They may also offer their private label credit card and other financial instruments.

As industries define their specific portfolios of products and services, this view is also important to adequately define the scope and priorities of the planned MDM-CDI effort. This understanding can help bring a valuable perspective from the groups of existing or new stakeholders of the CDI project.

Levels of Aggregation and Relationship Types

This dimension of scope defines how the data should be aggregated. Typically data aggregation is an area discussed within data warehousing projects. If a data warehouse has already been built, the CDI project scope should answer the question of whether the CDI Data Hub will feed the data warehouse in the future and how the existing processes will be impacted. If the data warehouse is not available yet, we do not recommend mixing the MDM-CDI Data Hub project and a data warehousing effort, even though interdependencies between the two efforts should be well understood.

Even though creation of multiple data aggregation layers is not the primary focus of the CDI Data Hub, we should consider a data aggregation view that is directly associated with CDI Data Hubs. This particular data aggregation view is also known as a "single version of truth" for customer data. We defined this new, additional view in Parts I and II of the book. Indeed a typical enterprise does not want to get rid of all of its customer records even though some of them may exist in multiple versions. The discussion on customer data aggregation may reveal an enterprise's intent to preserve and maintain the existing redundant customer records along with the new single version of truth for customer data. Since by definition, a CDI platform integrates all available data about the customer into the authoritative system of record, this single version of truth represents an aggregated data view. We discuss CDI data aggregation in more detail in Chapter 12.

A discussion of relationship types that have to be supported and managed by the CDI platform is another important dimension of scope. CDI and customer relationships are discussed in Chapter 13.

Entities and Attributes

As the CDI project is initiated, an initial logical/canonical data model of the integrated solution should include all entities, key attributes, and other attributes (to the extent possible at this early project stage) required by the integrated solution regardless of which systems these data elements reside in at present. The canonical data model, described in Chapter 8, defines the entities in scope, and the relationships between the entities and data attributes in scope no matter where they physically reside.

Clearly, to enable proper CDI functionality, the data attributes used for customer identification and matching should be included in the model. However, some of the data attributes and entities may not be available in any of the existing systems at all. Such a logical data model provides the organization with a technique to abstract their analysis from the complexities of the existing data structures and develop a desired consolidated data model that represents business vision correctly.

It is not always easy to conceptualize the enterprise vision and abstract it from the organizational realities. Therefore, we highly recommend finding the right external partners specializing in logical data modeling, preferably with deep expertise in an appropriate subject area domain (i.e., customer, product, subscriber, etc.). Some domain-specific data models are published by their owners or vendors. As an example, please see *The Data Model Resource Book,* Vol. 1, *A Library of Universal Data Models for All Enterprises*, and Vol. 2, *A Library of Data Models for Specific Industries* by Len Silverston (Wiley, 2001).

If you feel that the data models recommended by your partners do not entirely fit your organizational needs, which is not uncommon, your organization would still benefit from the experience of data modelers who built industry-specific models. It is also very useful for project direction to acquire a clear understanding of why the

industry model does not fit your organizational business model. Whether you buy a data model from an external source or decide to develop it internally, discussions about the choices you need to consider in developing and deploying the data model for the CDI platform will enable the organization to establish a logical data model that defines the scope of the solution from the data attributes' perspective.

There are other considerations that drive the scope of the canonical model. In addition to the initial scope, the team should determine the scope of the incremental data model changes as the CDI platform evolves from one release to the next.

Systems and Applications in Scope

Systems and applications are another important dimension of scope. Specifically, in a typical Customer Data Hub the data is sourced from multiple systems. When the Customer Data Hub is in production, how will the current systems be affected? Some applications and systems may have to be phased out, which is a significant scope issue that also determines the end state of the solution and work in the legacy system areas that must be planned.

Alternatively, existing legacy systems may have to coexist with the Data Hub. The discussion of what such coexistence means will lead us to the topic of the next section about the Customer Data Hub solution architecture.

Customer Data Hub Solution Architecture

As the process of the project scope definition reaches a point where the team gains a consensus about entities, data attributes, products/account types, and lines of business, the CDI project can move into the next phase to decide upon architectural choices for the CDI Customer Data Hub. CDI products and solutions known as Customer Data Hubs are designed to support data structures, functions, and services that enable rationalization, integration, and delivery of customer data. A conceptual CDI Data Hub architecture, described in detail in Part II of the book, recognizes a number of options that can be used to solve customer data integration problems in the context of the business requirements of a given enterprise. Let's review these architecture options as they have been defined by the industry research firm Gartner Group. The follow-on section in this chapter provides an analysis of the architecture styles and offers some insights into and variations of the architecture options, which are based on the authors' practical experience implementing CDI solutions.

Data Hub Architecture Styles

A well-known industry research firm, the Gartner Group, has defined the following architecture styles for typical Data Hub implementations.

Registry Hub

The Registry-style Data Hub uses a metadata repository, or a stand-alone Data Attribute directory that points to the locations of the data attributes using specialized Data Hub services called Attribute Locator service and the Metadata service (see Chapter 5 for more details on the Data Hub services). Figure 11-3 illustrates the way the Registry-style Hub operates. For instance, the metadata repository should store the rules about the retrieval of the "best" customer name, the "best" (i.e., authoritative) source for account type, etc. The rules can be complex enough to take into account multiple conditions and scenarios. The Data Hub of this style stores only key identifiers and links them to the fragments of master data in source systems. In addition, the Registry-style Data Hub supports data transformations necessary to achieve semantic consistency and reconciliation (please refer to Chapter 14 for additional discussion on semantic reconciliation). The Registry-style Data Hub provides a real-time reference by dynamically assembling an integrated but read-only customer view from the source systems.

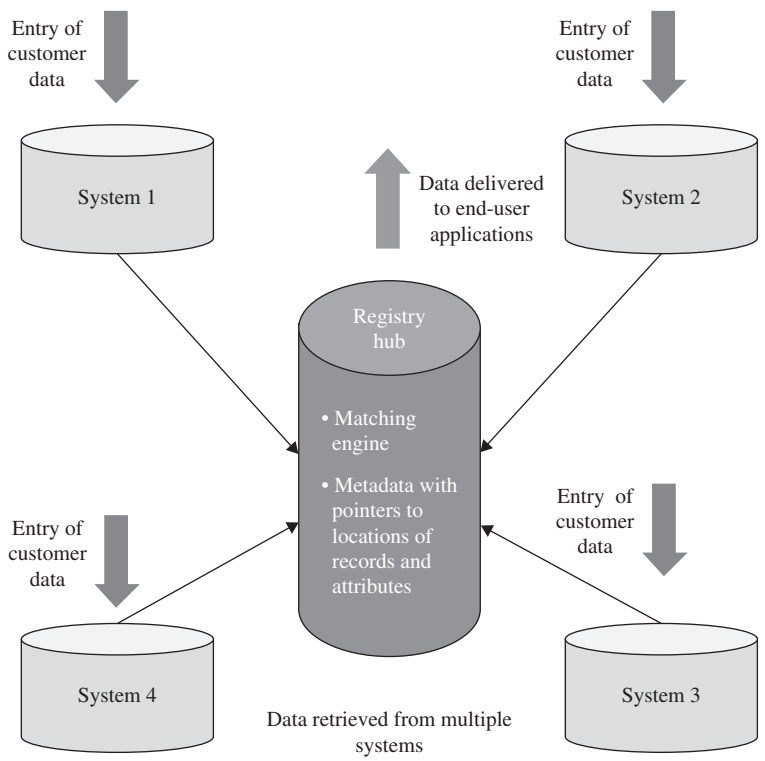


Figure 11-3 *Registry Hub*

The Registry Hub is the right choice if the company's strategy is to preserve the existing systems and invest their development funds to fix and enhance legacy systems over time. Considerations for and against a Registry-style Data Hub are shown in the following list.

Pros:

- The lowest-cost customer data integration solution.
- The data flow changes are limited and implementation risks are minimized.
- Only limited data reconciliation between the legacy and the new data Hub systems is required.

Cons:

- If the Data Hub has to support complex business rules including data survivorship, the data access (query) performance of the Hub can be an issue. The term "data survivorship" refers to the rules defining how to assemble a single record from two or more records with overlapping attributes that may contain conflicting values. In this case the attributes "compete" for survivorship to resolve the conflicts.
- Query performance represents an even bigger concern when multiple systems must be accessed to retrieve the data.

Coexistence Hub

The Coexistence Hub architecture style of the Data Hub (see Figure 11-4) physically stores some master data along with referencing some other data in the source systems. This Data Hub style is not used to directly originate transactions, but is updated from the source systems that initiate transactions. The source systems serve as the systems of record. The Data Hub is used as a central reference point for customer data. In addition to the customer data, the Data Hub can also store relationship data, customer groups, etc. with specifics dependent on the industry and organizational needs.

Identity Hub

A slight variation of the Registry-style Hub design that enables primarily matching and linking services is known as the *Identity Hub*. It is similar to the Registry Hub in that it stores pointers to data attributes that reside in external systems, but it is designed to store and manage only those data attributes that are required to perform record matching, linking, and party identification.

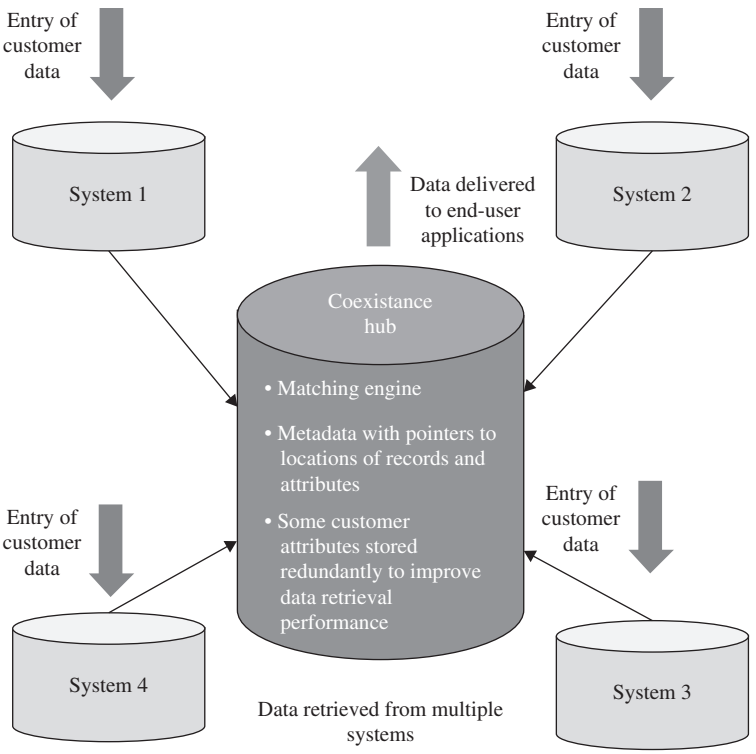


Figure 11-4 *Coexistence Hub*

The Coexistence Hub bridges some gaps in the existing systems. It is the right choice if the company's strategy is to partially preserve the existing systems and decommission at least some of the legacy systems. The Coexistence Hub style is sometimes used as a step towards a Transaction Hub. Its advantages and disadvantages are summarized in the following list.

Pros:

- The Coexistence Data Hub solution cost is relatively low.
- Data flow is limited to one-directional synchronization.
- Data retrieval performance issues are resolved by storing certain data attributes in the Data Hub. The complexity of data transformation is moved to ETL.

Cons:

- ETL transformations that are required to maintain Data Hub content can be fairly complex.
- Since the Coexistence-style Data Hub assumes some data redundancy, its design should provide for synchronization and reconciliation of data changes between the source systems and the Data Hub.

Transaction Hub

This Data Hub style physically stores the customer data and is used as the authoritative system of record for customer data, as shown in Figure 11-5. This style of Data Hub supports services that apply data access transactions directly to the Hub and generate messages or batch files that publish the results of the transactions to the external systems. The Transaction Hub is the right choice when the organization does not intend to invest additional money and resources in the source systems for the data domains where the Data Hub must become the master. In this case, a prudent approach is to prepare to support significant data flow changes in the

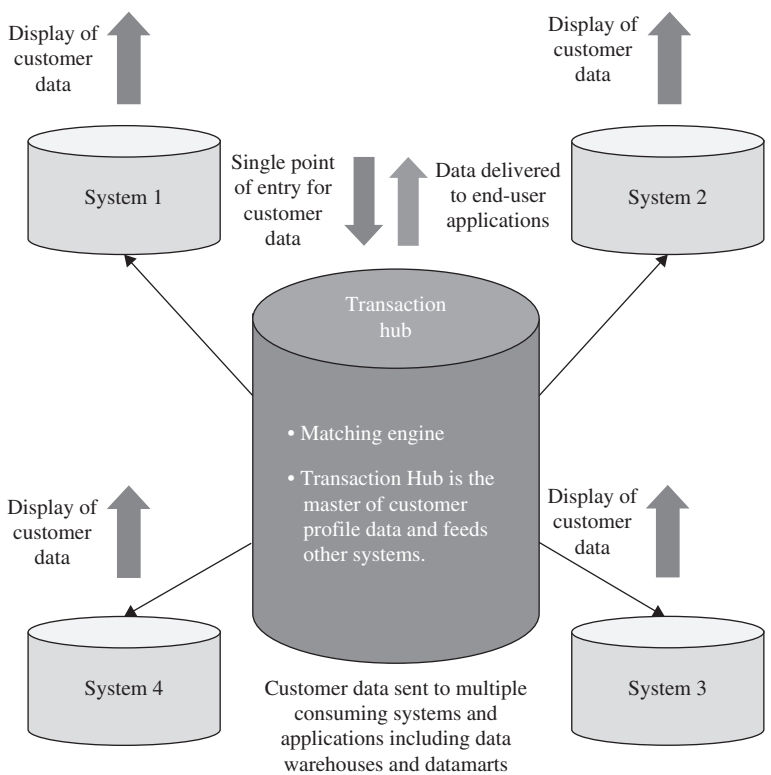


Figure 11-5 *Transaction Hub*

existing system structure including decommissioning of some of the legacy systems. Transaction hub advantages and disadvantages are shown in the following list.

Pros:

- This is a comprehensive solution that can be used to phase out obsolete legacy systems.
- This architecture style of the Data Hub allows organizations to achieve at least one of the major CDI goals—the creation of an accurate, timely, and complete system of record that maintains just-in-time data accuracy and integrity.

Cons:

- This architecture style of the Data Hub results in the highest complexity of ETL implementation.
- This style also requires complex real-time synchronization and reconciliation. Therefore, it usually demands the highest cost and implementation risk. The complexity of data synchronization is discussed in Chapter 15.

The taxonomy of these architecture styles illustrates an interesting pattern: the richness of the Data Hub functionality and the complexity and the associated risks of CDI implementations increase as the Hub stores and manages more and more customer data. Therefore, CDI projects need to carefully evaluate the benefits and risks associated with each approach and decide on the appropriate road map for implementing a CDI solution in a phased, risk-managed fashion.

Phased Implementation of Customer Data Hub

Even though an ultimate goal of the CDI project may be to develop a Transaction Hub solution, in order to achieve this goal and to manage the risks and impact of implementing a Transaction Hub, a CDI project typically begins with a "slim" Data Hub implementation. The plan should be to evolve the Data Hub by increasing the number of data attributes for which it acts as a master. As the Data Hub data scope grows, so does the value that the Data Hub provides to the organization. From the project management point of view, this evolutionary change should be organized into well-defined project phases. Of course, it is clear that this chapter discusses a phase of the CDI project called the Initiation phase.

Using this approach as a guide, we recommend a somewhat different categorization of the Customer Data Hub styles that are more aligned with phased implementation as shown in Figure 11-6.

Artifacts That Should Be Produced in the Project Initiation Phase

Typical artifacts that are to be produced at the end of the Project Initiation phase are shown in the following list:

Phase 1: Hub "Slave"	Phase 2: Hub "Slave" enhanced	Phase 3: Hub "Master"
Data governance • Reference data • Data cleansing in source • Data cleansing during Transformations Buy vs build and client hub vendor selection • Data-model-specific • Data-model-agnostic Customer identification (matching) & data quality vendor selection • Deterministic match • Probabilistic match Key generation cross-reference loading and synchronizing the hub Visibility & security at the record level limited to view only	Additional attributes, LOBs and systems Attribute level visibility & security solution All account types, party types, domestic vs international Enhanced customer identification capabilities based on improved data quality and stewardship Comprehensive reference data translations Visibility & security at the attribute level Legacy system/functionality phase out road map	Direct updates against the hub Inverse data flows to support hub master scenario for selected fields Customer identification enhanced by end-user input • Merge • Split • Data enrichment Comprehensive visibility & security implementation Legacy system/functionality phase-out Comprehensive transactional semantics Comprehensive visibility & security solution including support for direct data changes in the hub

Figure 11-6 *Data Hub phased implementation*

- Business process analysis (current state)
- Requirements for business process improvement and re-engineering (desired target state of the business processes)
- Incremental business process changes by release
  - Incremental benefits by business function and line of business
  - State of the solution architecture by release
- Conceptual and logical data model of the integrated solution and how it ties back to the business processes
- Scope and priority definitions in terms of data attributes, products/account types, and lines of business
- Solution architecture and the architecture road map indicating how the architecture evolves with the implementation releases
- Vendor product evaluation criteria, buy vs. build decision, and tool recommendation/selection for the key areas of CDI Data Hub functionality

Project Work Streams

In the beginning of this chapter we mentioned that an effective methodology to managing complex projects such as a CDI Data Hub is to use a phased approach and organize the work and resources into a number of interconnected and interdependent

work streams. The following work streams typically represent the body of work that needs to be planned and executed.

- Customer identification
- Customer/account groups and relationships
- Data governance, standards, quality, and compliance
- Data architecture
- Metadata and related services including record locator and attribute locator metadata and services
- Initial data load
- Inbound data processing (batch and real-time)
- Outbound data processing (batch and real-time)
- Changes to legacy systems and applications
- Visibility and security
- Exception processing
- Infrastructure
- Data Hub applications
- Reporting requirements of a stratified user community
- Testing
- Release management
- Deployment
- Training
- Project management

To sum up, if you are planning to embark on an MDM-CDI effort, this list can be used as a guide to build a detailed project plan including appropriate resources and the project team composition. Each of these work streams should have clearly defined deliverables that have to be aligned at the entire project level in order to produce a cohesive and comprehensive solution. Although these work streams define different interdependent efforts that prevent them from being executed in a totally parallel fashion, many of these work streams can be structured so that their dependence on each other is minimized and the overall project timeline is optimized to parallelize as much work as possible. We will cover the areas addressed by these work streams in the chapters that follow.