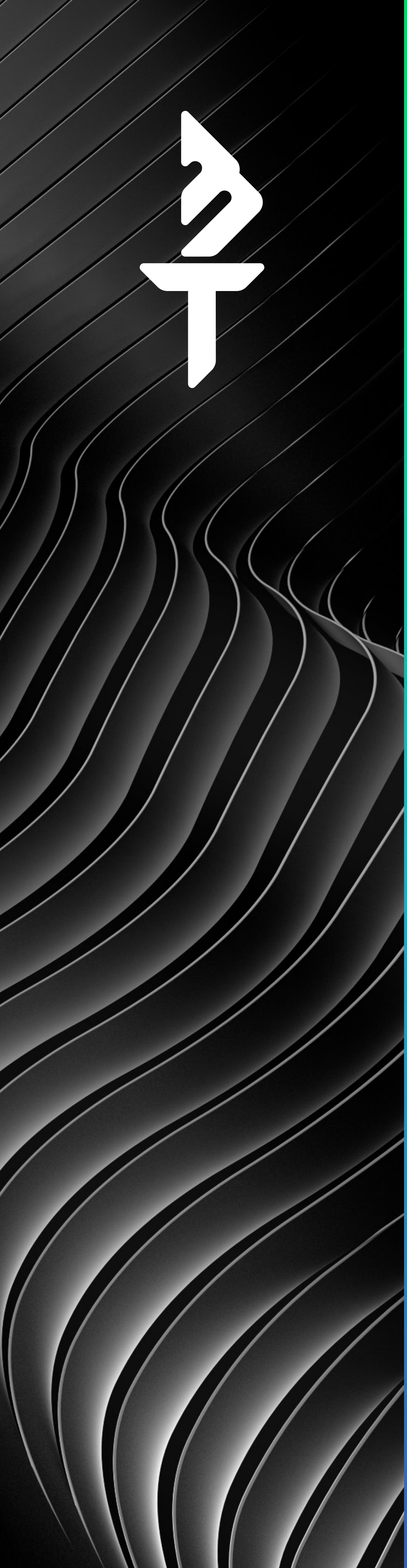


White Paper

Data Governance Maturity

Is it time for a Master Data Management solution?

By Mark Boltri

MACEDON
TECHNOLOGIES

Table of Contents

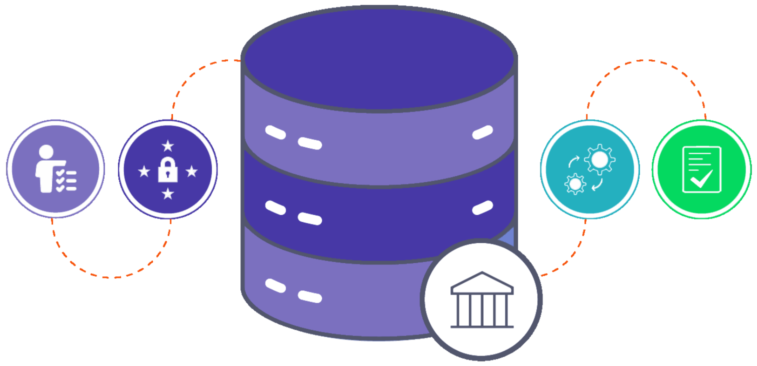
Introduction

As digital transformation becomes more central to the success strategy of modern businesses, one of the most significant by-products is a proliferation of data. Processes that were previously completed by filling out a pen-and-paper form or entering data into a spreadsheet are now being done in shared systems. As a result, all the data associated with those processes is visible to the organization, often for the first time. This influx of data presents many opportunities, but also a key challenge – how does the organization ensure that the data is consistent, accessible, and manageable, even as the volume of data increases exponentially? This is where Master Data Management, or MDM, comes in.

MDM is, simply, the idea that the critical data of an organization can be easily maintained even as the business continues to scale up. When organizations have complex application ecosystems, they can often face challenges with data management. The same data might be stored in multiple different systems, causing errors due to inconsistencies. Such dispersion of data makes consistent reporting a challenge, and it becomes harder to protect sensitive data like PII (personally-identifiable information).

To address these challenges, the ideal situation is that data management strategies have been put into place from Day One, but the reality is that most organizations are far into their digital transformation journey before these problems start to become unmanageable. Fortunately, there are many techniques that exist to standardize and centralize the data to fit into the MDM paradigm. I will explore some of those techniques later, but first, I would like to dive deeper into some of the use cases that are a good fit for MDM.

MDM Use Cases



Shared Data Entities

Let me start with a very common problem: different applications have different versions of the same data. As an example, consider a business that uses separate applications for CRM (customer relationship management) and invoicing. Both applications are going to need access to data about customers, although the CRM system will have sales-oriented data while the invoicing system will be more focused on financial information. As a result, these applications may end up storing different fields related to the same customers, or you may end up with a customer that exists within one application but not another.

Why is this a bad thing? Consider what happens when data becomes inconsistent between applications. To continue the above example, suppose that one of the customers of our hypothetical business relocated their headquarters and therefore have a new mailing address. The sales department is in regular contact with the customer, so they updated the CRM system with the new address. However, the finance department does not have frequent customer contact, so the invoicing system does not have the new address. As a result, invoices get sent to the wrong location, a mistake which may not be realized until weeks later. The net result is reduced cash flow for the business and potentially an annoyed customer.

If instead, the business had a robust MDM solution in place, there would be a single record that is shared across applications. Therefore, when the sales department updates the customer address, the finance department would see the change right away and take the appropriate actions to ensure the invoice gets sent to the correct location.

Consistent Structure and Formatting

Centralizing data in this way is a big benefit of MDM, but it is not the only one. Even if data is centralized, it can still suffer from the problem of poor standardization. For example, the customer record from the earlier example may have a field for "state". If different applications are not following the same standards, you may end up with customers for whom the value of that field is "West Virginia", "WV", and "W. Va."

In each case, it may be obvious to a user what state is being referred to, but it becomes a problem when trying to report on the data. If a report exists aggregating data by the value of the "state" field, and a manager wanted to see sales in each state, they would have to manually tally up all of the variations of every state.

MDM tackles this problem by not only centralizing the data, as we saw in the first example, but also by standardizing it. In the "state" example, the team responsible for implementing MDM would likely attempt to normalize that field by using a shared reference table for states. Instead of storing "West Virginia" or "WV", an ID corresponding to West Virginia would be used; this ID would be consistent across applications. In this case, the hypothetical manager's report would be able to aggregate data across consistent categories, without the need for manual reconciliation.

Safeguarding Regulated Data

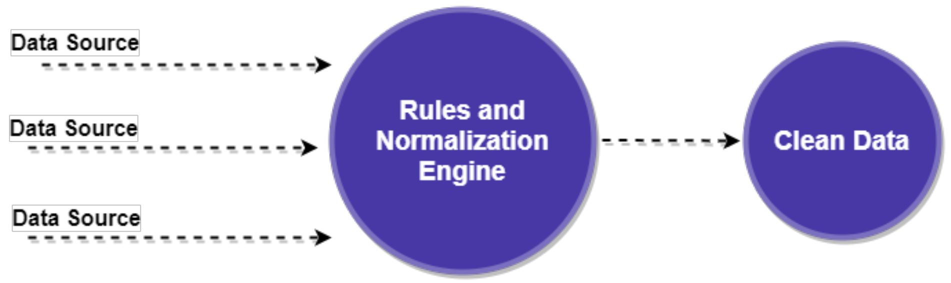
The last use case I would like to examine is protection of sensitive data. Let me go back to the example of the CRM and invoicing systems. Suppose the business using these systems is a hedge fund, and their customers are investors. Such an organization likely stores information that is critical to protect - bank account information, income data, possibly even social security numbers.

It is critical that this information is safeguarded, but in my example, such information is spread across two different systems, and both systems will require an implementation of the proper security standards. For example, if the organization is subject to Europe's General Data Protection Regulation (GDPR), they are required to encrypt users' personal data, and there is likely a cost associated with such encryption. In the example, the organization would have to bear this cost twice, once each for the CRM and invoicing systems. Now imagine a large organization with dozens or even hundreds of different systems, and the hypothetical costs balloon out of proportion

The solution to the first problem - centralization - also yields benefits here as well. By keeping shared data in the same location, businesses can not only ensure that common records are consistent, but also keep the costs of compliance down.

In addition to the cost component, centralization also reduces security-related risks. How? Centralizing the data's location reduces the number of potential attack vectors. For example, an organization may have sensitive data in multiple different databases, which may exist on a range of servers, each with their own administrator. In such a scenario, an attacker only has to get credentials for one of those systems to compromise the organization's data. When the data is spread out in this manner, it becomes a lot harder to consistently enforce things like password complexity, session idle time, and database user access levels. Conversely, centralizing the data makes it much easier to administer the necessary security standards. Instead of protecting dozens of homesteads, businesses can focus on defending a single fortress.

Techniques to Facilitate an MDM Solution



Now that I have reviewed some of the use cases that MDM is appropriate for, I will take a look at some of the techniques that make such a solution attainable: centralization, duplicate elimination, and normalization. Each of these topics are complex enough to warrant their own paper, so I will stay high-level in my discussion.

It is also worth noting that these techniques tie in closely with the process of data migration. A more detailed look at the data migration process can be found within a our Data Migration white paper.

Centralization

As alluded to in the use cases, one of the key tenets of MDM is centralization of data. This usually, but not always, means storing all key data on the same system, e.g., a single relational database. Less frequently, you may have some subset of data that ends up on one system, and another subset that ends up on another system, although even in such cases, there should still be an effort to ensure that each individual data entity has a single source of truth.

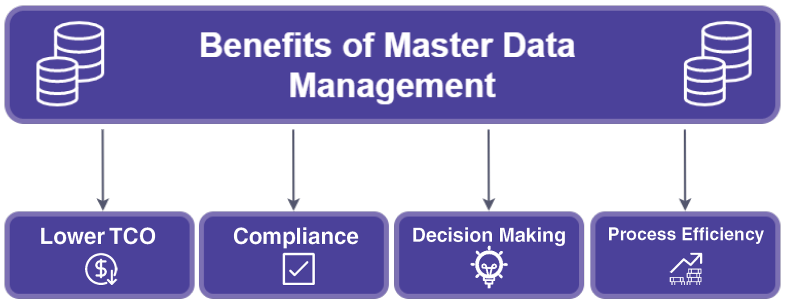
In either case, the key is consolidation. That is, identifying data entities that are the same between systems and combining them. In the example of customer data, this means having a single storage location for the “Customer” record, even though multiple systems use that data.

Duplicate Elimination

Centralizing data is a great first step, but often, one of the by-products of such an effort is that a lot of duplicate entries end up in the consolidated data locations. For instance, in our customer example, we might be consolidating the data by identifying records where the customer name matches. However, if a customer’s name is “ABC Company” in one system, and “ABC Company, Inc” in another system, we will end up with a duplicate, which has many of the same problems as keeping the same data in separate systems.

Eliminating these duplicates can range from relatively simple to extremely complex. There are typically two phases - identifying the duplicates and merging them, both of which have their own challenges. An MDM implementation team has to weigh the time and cost considerations of implementing an automated solution vs manually correcting the data; most solutions will involve some combination of both.

Normalization



Hand-in-hand with duplicate elimination is normalization, which ensures that data can be referenced in a consistent way. This ties into the “West Virginia" example I used earlier - in that example, common reference data was factored out to a separate location. This can also apply at the field level; for example, a name field (name: "John Doe") might be broken out into separate fields (firstName: "John", lastName: "Doe"). Normalization becomes especially important as data is consolidated, because different systems may use different formats for the same data, and/or the same data field may exist in the separate data sources with a different field name or set of properties.

Conclusion

To summarize, I have reviewed the concept of master data management as a whole, examined some use cases that are appropriate for it, and explained several techniques that make it attainable. When put into practice, an effective MDM strategy serves to ease the burden of managing the proliferation of data, which can at times seem overwhelming. Some of the benefits I discussed include:

- **Reduction of Data Inconsistencies:** Ensuring a unified record for key data entities enables different applications within the IT landscape to speak the same language and avoid costly mistakes caused by inconsistencies.
- **More Valuable Reporting:** Standardizing the way the data is represented and removing duplicates helps guarantee consistency in reporting, providing the key insights needed to make crucial business decisions.
- **Easier Safeguarding of Protected Data:** Centralizing the location where data is housed significantly reduces the exposure for businesses that are worried about security and compliance and serves as a selling point to potential customers.

Viewed holistically, this variety of strategies makes the task of managing enterprise data into a realistic vision and can be a strong differentiator over the competition. With an effective data management approach in hand, organizations can become primed for success in a business landscape where data is everything.

About the Author

Mark Boltri joined Macedon in 2014 and serves as a technical advisor and escalation point in his role as an Appian Enterprise Architect. He previously operated in a more management-focused role as Portfolio Manager, and this experience gives him a unique, business-focused lens through which he is able to analyze technical challenges and design high return-on-investment solutions.

Contact: (571) 526-4281
info@macedontechnologies.com