10 crucial components of a data integration plan

With real-word examples

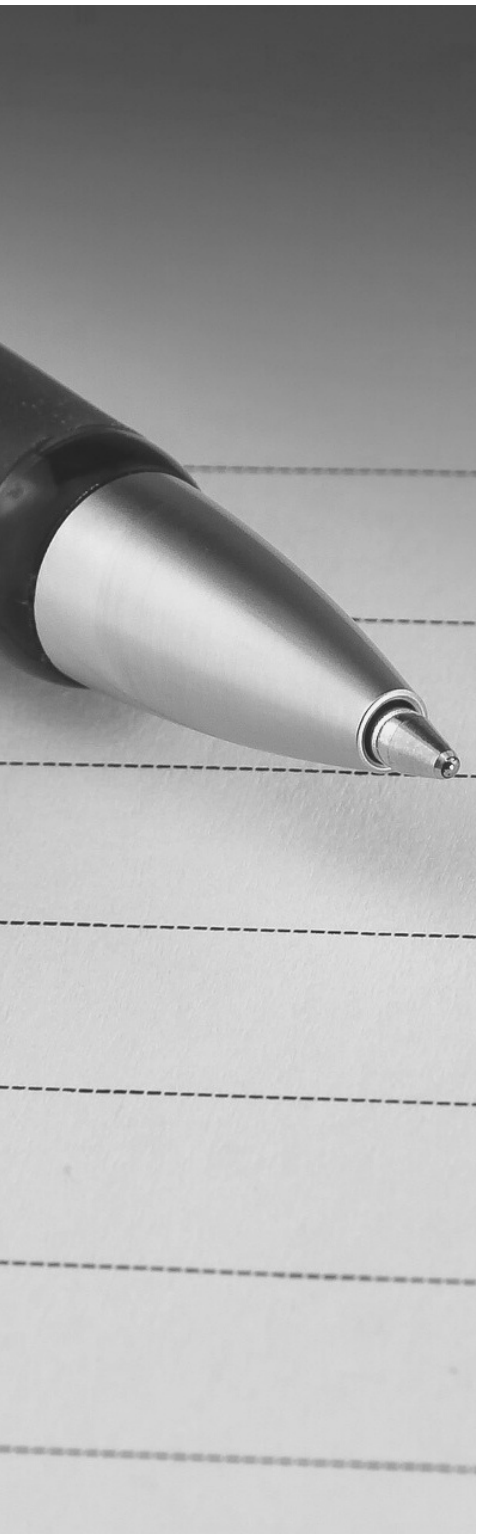
Introduction

Thank you for downloading this guide, which contains real-world examples to help you successfully integrate data from mutliple systems.

If you're carrying out the project in-house, this guide suggests ten practical things to do to create an effective methodology and plan. If you're briefing an external consultancy, the examples in this guide will help you compare different service providers and thoroughly check their proposals.

The guide is based on our years of work in data integration. We integrate, cleanse and consolidate data for organisations worldwide. The processes in this guide form the proven plan that our consultants follow to deliver integrated data successfully to our clients.

Click here to subscribe to our email newsletter list and be the first to receive our future publications.



1. Clear project objectives

A set of clear objectives for the project ensures that it delivers all the requirements of its stakeholders, on time and to budget. Objectives are also a tool for measuring the project's success. Consider setting objectives for both the integration project itself and for the ongoing integration.

The project's objectives should be aligned with the business's goals as much as possible. They should also be realistic in terms of budget, resource availability and timescales. For instance, objectives can be used to make the best use of staff resources by specifying a reduced set of data for integration during the first phase.

Real-world example:

Our data integration system is called DataHub. A bespoke version of the system is created for each client, but each project's objectives almost always include deadlines and milestones for the number of dealers connected to the system.

Once a DataHub system is implemented for a client, our objectives include measures such as the amount of data moved daily, number of successful data uploads and the proportion of errors. Here is an example from a current DataHub system collecting dealership data for an automotive manufacturer:

- 99+% of dealers uploading daily
- New dealers added in one working day
- 80% of dealers have no downtime
- Changes to dealers' systems are dealt with in one working day.

2. An optimal infrastructure

Start by reviewing all the systems involved with the data. Include the systems involved in extraction and every system that uses the consolidated output, not forgetting those in the cloud. Check that there is adequate connectivity between the systems, including data connectors, SFTP ports or APIs. Also identify any manual processes currently in use, and check whether or not these can be automated to free staff from repetitive tasks. In addition, identify any legacy systems currently holding the data and consider whether or not they can be updated to more modern systems.

Any necessary hardware also needs to be reviewed. For non-cloud projects, consider what hardware will be required for development and for the live system itself.

Real-world example:

For non-cloud-based versions of DataHub, we usually set up and host four servers. These are development, test, live and failover. The first three servers maintain a clean separation of the development, test and live environments. This structure allows the development of the system to continue after launch, without impacting the daily processing of the data by the live environment.

Within this set-up, all servers hold the same configuration and system structure. Once we are ready for release, all software is versioned within our version control system and released to the test server for validation. For future releases, for instance for updates or change requests, the system is reviewed using files generated on the test server to ensure that it meets the defined requirements. Once the system updates on the test server are accepted, the system is released to the live server. This process enables all updates and change requests to be signed off before going live.

On a daily basis, the live server content will be copied to the failover server. The failover server will therefore contain an identical copy of the configuration and data, running in the live environment. In the unlikely scenario that a critical issue arises with the live server, we would switch to the disaster recovery failover server. This allows the daily processing of data to continue.

3. A defined data structure

It's vital to gain a complete understanding of the data that will be integrated. But if resource is limited, focus on the data that is needed for the business's goals. Rather than over-complicating the project by trying to collect the maximum data right from the start, ensure that you have an extensible infrastructure that can accommodate further data streams in the future.

List all the data sources and identify how they can be accessed. Access methods could include: direct from database; file uploads e.g. Excel, CSV or XML; API; or push notifications.

Invest in profiling the data to ensure that it meets requirements and that any issues are identified. This is always a good use of budget as it will identify issues, gaps and risks early in the implementation. A data mapping process should be used to match the fields in the source dataset with those in the final output. Check the format of the data to identify any inconsistencies in fields such as dates, monetary values, names, addresses, telephone numbers, locations or product codes. Also check the dataset for any gaps. Our consultants create a mapping specification spreadsheet to share the results of the mapping process with stakeholders.

Where data is supplied by a third party, ensure that the specification of file contents is defined and signed off early. If the data is already available within your organisation, ensure that sample data is available for testing early on in the project.

Real-world example:

The standard DataHub implementation involves the extraction and receipt of data from a client's systems in a variety of forms. Our consultants therefore use the process above to determine whether the data will be received using our specialist automated extraction applications, or in the form of data files from the source, such as a retail location, or a mix of the two.

The data is usually extracted in terms of what we call data feeds. This gives a categorisation to the data. As an example, typical feeds we extract within the automotive industry include: parts; invoices; new and used car sales; and labour.

4. A defined data flow

Data flow is concerned with how, when and where the final data feed is delivered. Reporting requirements should also be considered at this stage. Determine how frequently the data should be transferred, or whether there is a need for real-time access to data.

Map out the current data flow, if there is one, then determine the most efficient data flow for the project.

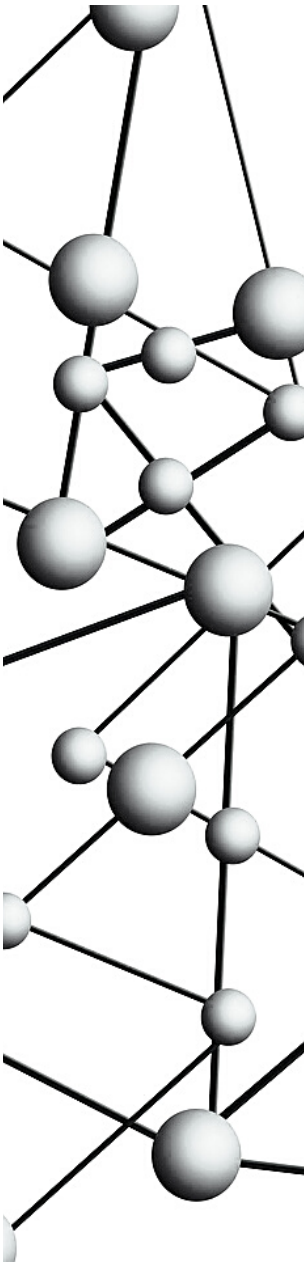
Real-world example:

Data is transferred into DataHub using our Transformation Manager software. DataHub then processes the data and carries out data quality and validation checks in accordance with each client's defined business rules and mappings. Invalid data is rejected and reported to the client or our support team, along with any source failing to upload their feed to DataHub. Valid data is then sent to the client's systems, whether on-premise or in the cloud.

A typical DataHub system usually follows the data flow outlined below:

1. Data is extracted from each source, such as a retail location, after the close of each day from 6pm GMT onwards, or received in file format up until a close point of midnight GMT. Data is retrieved using one of our standard extraction methods:
   a) Direct access to the source's system and extraction via our application
   b) Our application on the source's server sends a flat file containing extracted data from the source's system
   c) We provide a sftp server for the client's software vendor to send flat files containing the data extracted from the data source
2. Data is processed daily from midnight onward by our system
3. Data is populated into the client's system (such as an Azure SQL database) during the early hours of the morning, prior to a close point of 7am GMT.

This data flow structure and timings are held in a project spreadsheet. This spreadsheet is versioned and kept within source control. The document is agreed with all stakeholders to ensure clarity on data flow and other components of the system.



5.Proven software

Cost, functionality and staff expertise in a particular application are key factors in choosing software for any project. However, for a data integration project, specific issues include:

- The software should make mapping the data straightforward, preferably with a drag-and-drop interface
- The software should connect with the source and target systems, preferably with pre-built connectors and with minimal coding
- The software should scale down or up according to the quantity of data involved in the project
- The software must be compliant with all government security and data protection regulations, including the UK GDPR.

Real-world example:

Our framework of proven software and methodology is used to implement our data integration service. It involves two applications:

The DataHub system is used to manage the flow of data from the network to the client. Following the aggregation and validation of the data, along with application of business rules/mapping rules, automated reports will be generated prior to sending the data to the client's systems.

Our Transformation Manager software applies the agreed business rules and mappings to the data. It manages the transformation of the source data into the DataHub system, through validation and out to the final system.

6. A robust data quality plan

A successful data integration will need to include the following steps during data processing:

- Deduplication
- Data cleansing
- Data enhancement
- Data validation.

The rules behind each step will need to be defined, and the information held in a central document which is versioned and kept within source control.

Testing is an important stage in ensuring data quality. Thorough testing includes: unit tests, which ensure each component of the system is working correctly; system tests, which check the entire system; and integration tests, which test components of the system in specific groups. User acceptance testing is also important in any new project; test data can be sent to users such as outlets or manufacturers for validation.

When processing data feeds, the raw source and the expected results can be incorporated into regression tests. The tests are usually performed in the test environment prior to making a release to the live environment. This allows the continuing development of the system during and after the pilot phase, minimising any distribution to the live system.

During the daily processing of data, automatic data quality reports are extremely useful. The reports can be configured to be sent as a single email to a list of end users or a support team.

Real-world example:

Our DataHub system includes standard rules applied by default to the data to ensure its baseline validity. As an example, for our automotive clients these rules include initial data clean-up:

- Validate VIN structure
- Validate email structure
- Validate address structure
- Validate customer name structure
- Remove duplicate records.

We then jointly agree rules with each client such as:

- Determine customer relationship to vehicle:
  Customer is a person or a company
  Customer is a person and is the owner of the car
  Customer is a person and is a company car driver
  Customer is a company who is the registered keeper of the car
- Determine service or repair types using job code\text matching e.g. routine service, other service, MOT, repair. bodywork
- Determine invoice sales types e.g. warranty, internal, external
- Identity match items e.g. use VIN to check if vehicle is seen at different sites.

External address checking and cleansing services with a programmatic interface are also often included.

Our Transformation Manager software validates the data feed structures, and key fields are checked against predefined rules (e.g. next service date is in the future and VIN contains 17 characters). Data from new dealers, and in particular new systems, are subjected to a rigorous set of tests.

Automatic data quality reports show the data availability for all live dealers, recording any failure of dealers or software providers to upload data to our system. The report also records when a dealer's data has been missing but has now been recovered. In addition, a report also assigns a level of severity to any data issues.

To complete the reporting, an automated report is usually configured to show whether or not the DataHub output has reached its destination. The format of the output transfer report varies depending on the method of transferring the data and the requirements of the client.

7. Compliance with local data protection regulations

Governments across the world are implementing increasingly strict data protection laws. One example is the GDPR, which is the European Union's General Data Protection Regulation. (It has been incorporated into UK legislation as the UK GDPR.) This regulation is designed to protect an individual's personal data. To comply with the GDPR's principles, any project involving customer data will need to ensure that the data is:

- Processed lawfully, fairly and in a transparent manner
- Collected for specified, explicit and legitimate purposes
- Adequate, relevant and limited to what is necessary in relation to the project's purpose
- Accurate and kept up-to-date
- Kept for no longer than necessary
- Processed in a manner that ensures appropriate security.

As data management specialists, we take compliance with the GDPR very seriously. In most of our data integration projects we are classified as a 'Data Processor', while each client is a 'Data Controller'.

If you're working on a project for stakeholders within your business, you are likely to be classified as a Data Controller. You will be responsible for determining the purposes and means of processing personal data. Furthermore, you must be able to demonstrate compliance with the principles of the GDPR listed above.

(Please be aware that this information is provided for your help and guidance and does not constitute legal advice.)

Real-world example:

As a Data Processor, we audit every project we carry out for our clients to:

- Identify which of the data we manage for our clients is classed as personal data and whether it is classed as sensitive personal data
- Ensure that all data is transmitted to and from our servers according to the GDPR's security requirements
- Ensure that all data is processed securely and appropriately
- Keep records of the processing activities we undertake
- Assist the Data Controller in allowing data subjects to exercise their rights under the GDPR
- Assist the Data Controller in meeting its GDPR obligations in relation to the security of processing, the notification of personal data breaches and data protection impact assessments.

For further information, download our GDPR checklist for data management projects.

8. A solid deployment plan

A successful deployment of the new system will usually include some or all of the following components.

A project champion

An effective champion will ensure that the integration has a sufficiently high profile within the organisation. He or she can ensure that adequate resources are available and that the relevant leaders buy into the project.

Defined stakeholders

Be sure to define all the departments within the organisation who use the data or systems involved in the project. It is likely that these departments will need to be involved in the design and implementation of the project.

Technical experts

Ensure that experienced technical people are available throughout the implementation. Knowledge of the systems and applications being used for the integration are both important.

Buy-in from the network

Build trust and confidence in the project by involving the people holding the source data at each relevant stage. Communicate the project's benefits to them and, if relevant, promise to only integrate the data that they have agreed to (or are contractually obligated to share).

Future-proofing

It is important to document the integration thoroughly to avoid dependence on the expertise of one or two individuals. It's also important that changes or extensions to the data can be made easily.

Quick wins

Build in short-term gains to show value to the business after the investment in the integration project has been made. Many data management projects reveal their value over time, so build in some quick wins to satisfy business objectives. Examples include the savings from closing legacy systems, an improved user experience leading to productivity gains, or new KPIs that highlight business issues.

Real-world example:

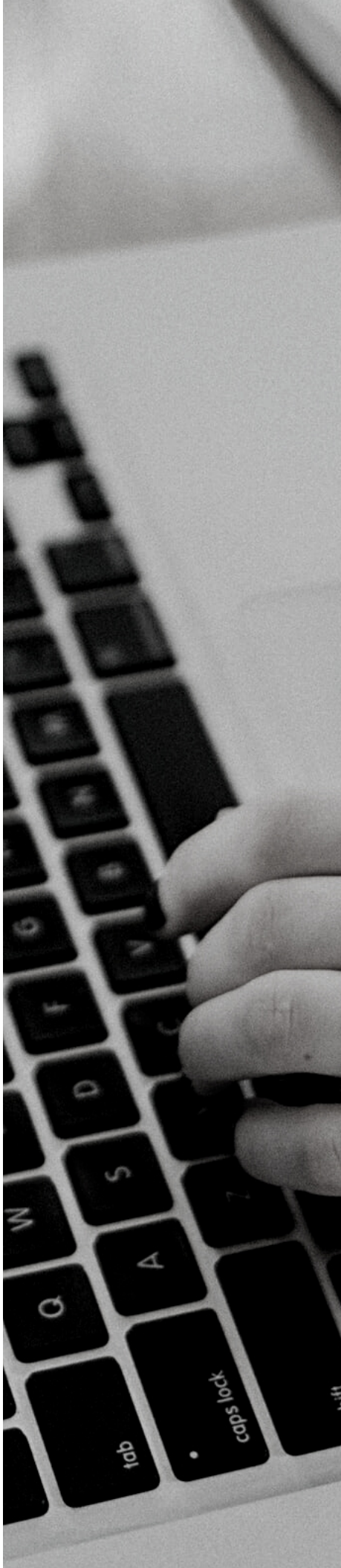
Deployment of our financial data integration projects usually involves a pilot. With the help of the client, we identify a small set of lenders (such as credit unions) who will be open and proactive in terms of their cooperation with the project. The client then notifies the chosen lenders of the data collection requirement and introduce us as the company carrying out this task.

For direct access extracts, we will provide the lender with a document guiding them through the short graphical setup required for the extract to occur. Where the lender has no technical capability, our support team can connect to the lender's system using a remote connection and assist with the setup.

A typical mechanism for ensuring lender buy-in is to provide an easy-to-access portal for the lender to see the data that has been shared. There are also often options to share data quality metrics so that the lender can identify areas for potential improvement.

After the pilot, a typical rollout will start with the system which covers the largest percentage of lenders, and then works down the list by percentage.

One future-proofing step that we recommend, if resources allow, is to specify a wider range of data fields than might be required initially. This is particularly important when extracts can only be obtained via financial software providers themselves. This will minimise the need to request additional fields in the future. However, this isn't an issue with systems where we have direct access, as we can make changes to the collection ourselves and this will be automatically deployed to the lenders.



9. A recovery and back-up policy

We recommend that all projects include a set schedule for back-ups. These ensure that data can be recovered in the unlikely event of data corruption, unauthorised access or system outage.

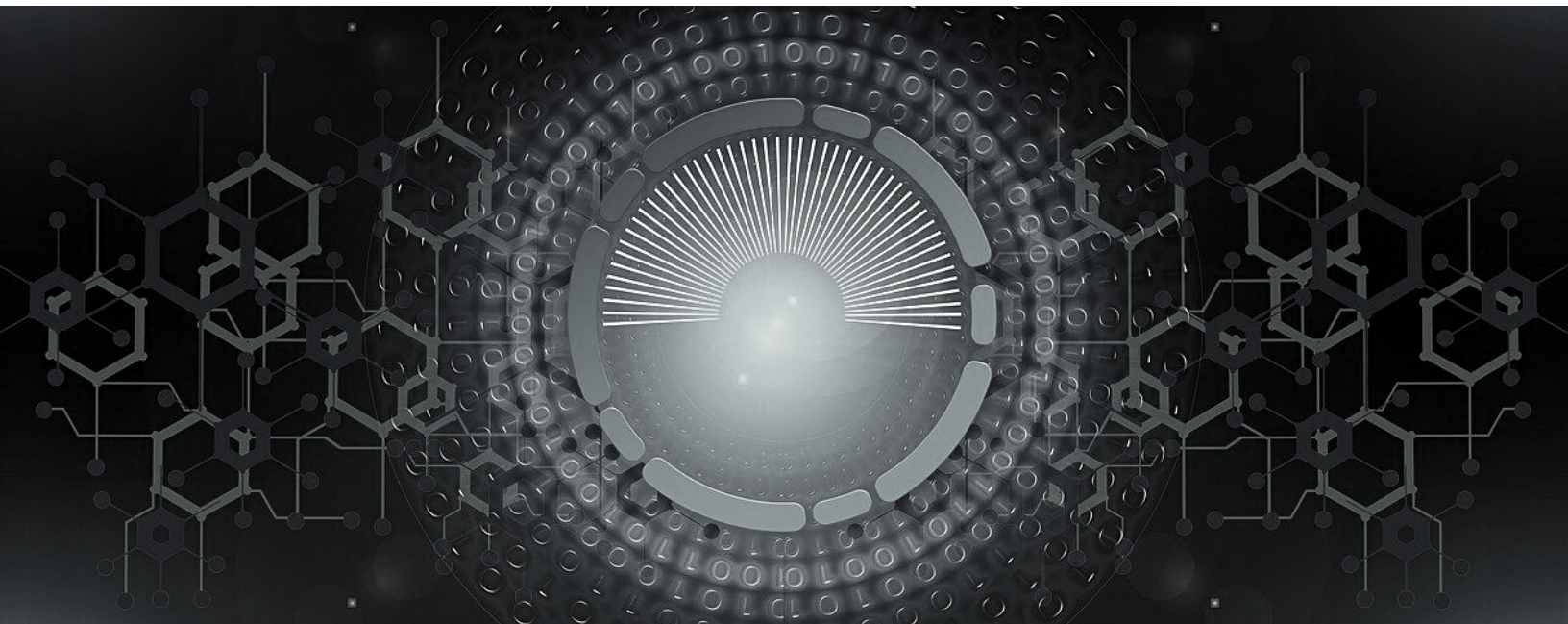
Each stage of the data integration should also include a plan for recovery. Consider how you can, if you need to, roll back changes during the process. Options include using a staging area or moving data in batches to ensure that it can be safely recovered.

Real-world example:

Every time a data extract is loaded into the central DataHub database, the original source file is backed up securely on our server. Daily backups also take place of the DataHub system itself (configuration database, source code and continuous integration tasks). Any cloud databases are also backed up.

Automated scripts run on a daily basis to delete files in the backup location whose timestamp is older than one year. Similarly, scripts are triggered in the database to delete records older than the agreed storage period.

A typical recovery time objective of 24 hours will be delivered by our failover system. This failover system mirrors the live system, and the switchover to the failover machine is carried out in less than 24 hours.

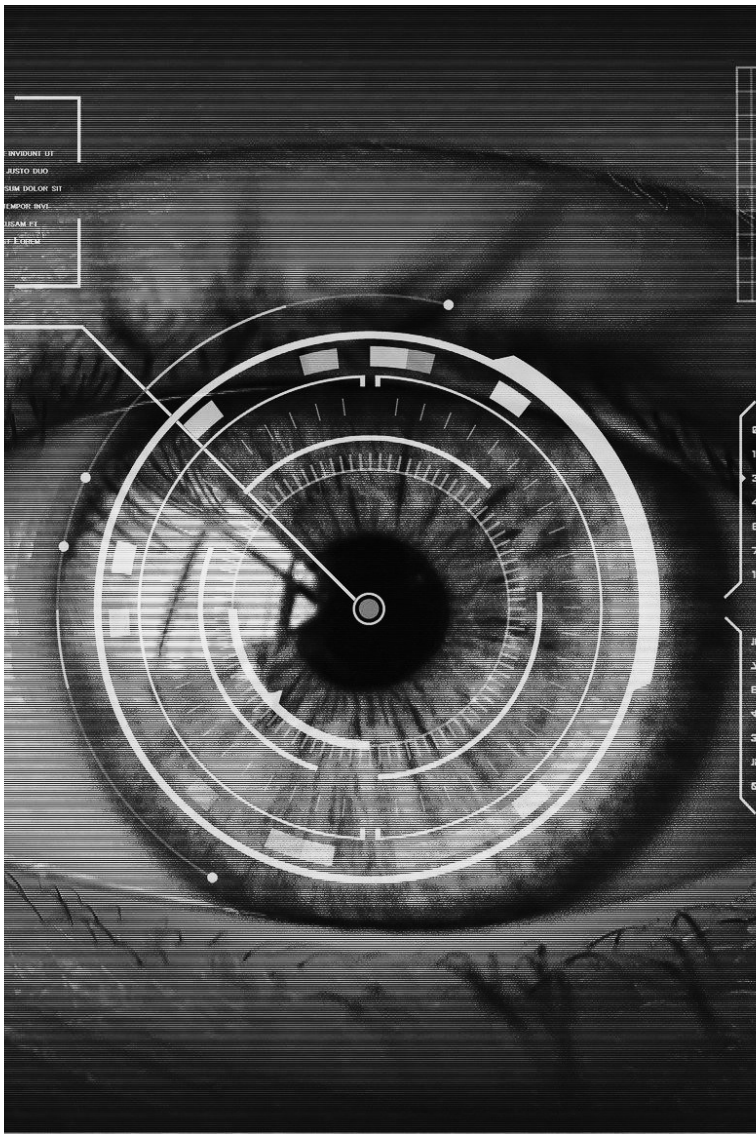


10. Robust security

Security has to be at the forefront of any data management project. All sensitive information, particularly customer data, should have detailed levels of security in place.

Before you start the project, check exactly which security protocols are currently in place: who is allowed access to the data, how and when. Then you can begin to identify any gaps.

Legal obligations should be thoroughly checked. As mentioned in the previous section on the GDPR, statutory measures covering data breaches and data protection are now in place in most countries. These often outline the levels of security that have to be in place, as well as stipulating operating procedures to keep data secure.



Draw up data security plans early on and embed them in the data integration plan. Areas to consider include:

- How to ensure secure data transfer
- How to create secure server access
- How to ensure secure data access
- Whether or not to increase the number of permissions required to access data
- Clearance and vetting of personnel, including outside consultants and partners
- The training or information sessions required by staff working on the project
- Vetting of the software that will be used to migrate the data
- Protocols for the use of email and portable storage devices.

Real-world example:

All our projects are developed with privacy considerations by design and as a default. Customers whose DataHub systems are hosted on a server have their own physical server and dedicated public IP address, hosted in a Tier 2 data centre. This category of data centre offers 24/7 monitoring, backup power, gas suppression, dual zone fire detection, dark fibre capability and multiple connections.

The systems we run are deployed from a base ESXi virtual machine that is continually kept up-to-date with security patches and operating system updates, and which is fully tested. We use the latest version of Debian 64bit Linux, hardened to improve security. All software is installed from only the official Debian repositories, or our own software built from secure source repositories and deployed via our change management processes. Using a common virtual machine configuration across projects means we can respond rapidly and efficiently to address security vulnerabilities.

All communication to and from every system is encrypted. Administrative work on the virtual machine is conducted via ssh using 2048-bit private and public keys. All web traffic uses https with 2048-bit certificates from Thawte. Data submission to each system is also carried out via encrypted sftp, scp or https connections.

Two-factor authentication is used by all staff who have access to customer systems. This can also be the default for a customer's users if required.

Further information

Click here to subscribe to our email newsletter list and be the first to receive our future publications on data integration.

Click here to find out more about our data integration services and software.