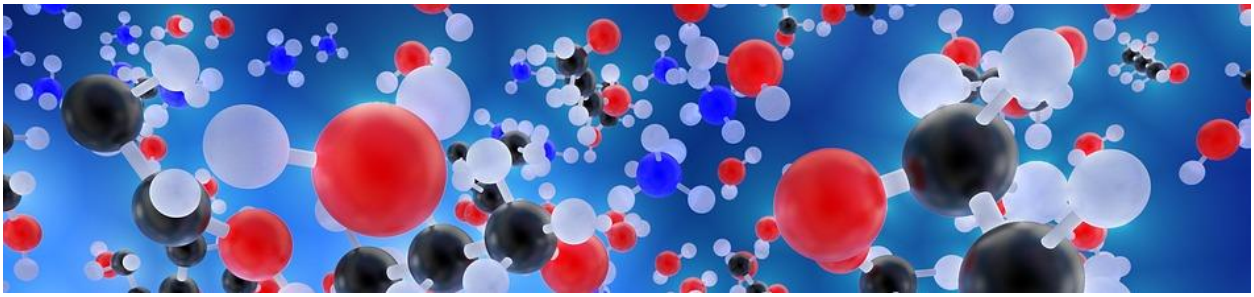
Choosing a Reporting Solution



Introduction

Reporting solutions vary depending on many factors including the size of the business (or department), security and auditability requirements, and scalability.

This article highlights common reporting patterns and examine the suitability of each. This is not an exhaustive list. Furthermore, the solutions follow "tried and true" approaches and only touches on sexier designs. There are countless permutations using newer technology and I expect to expound on this in other articles. Cloud solutions have also changed the landscape. Many of the patterns map to cloud-based platforms. Look for more on cloud-based reporting solutions in the future. In the meantime, the following addresses many of the typical patterns encountered in IT shops.

Types of Reporting Solutions

First, some terminology. Below is a list of terms to give names to solution patterns. Not everyone will agree with these terms. But they are useful for establishing the vocabulary for this article.

Built-In Application Reporting	Reports directly available within an application
Report server	server-based reports that query transaction systems directly
Operational data store	offloads transaction report queries to another system. Used for tactical reporting
Data mart	used for strategic reporting needs
Data warehouse	used for strategic reporting needs that span source systems
Data virtualization	used for reporting across systems while leaving data in place

Built-In Application Reporting

Reports that are directly available within an application. Best for a small number of users and the business has little or no IT support.

Pros

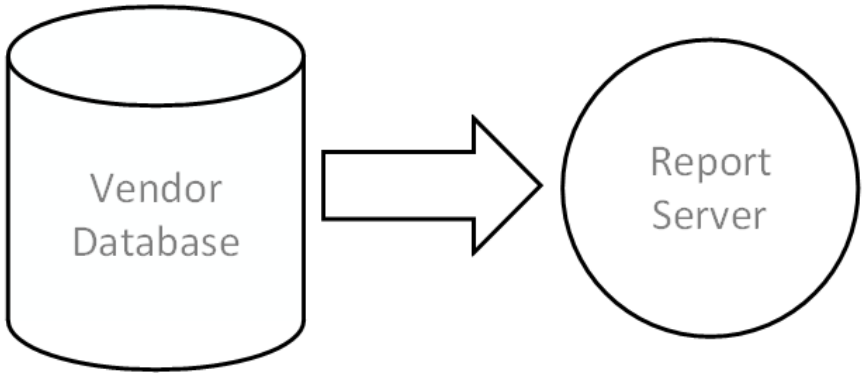
- Many business applications include reporting functionality. If the application includes reporting functionality and the number of users is small, then look seriously at building reports in the application.
- No additional software costs ate incurred if reporting is already included with the application. Note there will still be labor costs to implement the reports.

Cons

- Most built-in reporting solutions limit customizability. Usually, the user can set the header, footer, and a logo, and that's just about it. For the vast majority of situations, that's sufficient. However, this could pose a problem in some situations.
- The application itself is responsible for security. This means that it may not be controllable using a centralized mechanism like Active Directory, nor may it be able to offer row-level security, which enables the same report to present different content based on users' authorization.

Report Server

A Report Server is a central, go-to destination for organizational reporting. It contains server-based reports that query source systems directly. It can be used when the source applications use open data models and some IT support is available.



Pros

- A report server is helpful when the source applications provide little reporting capability, but their data is otherwise accessible.
- Report servers are usually integrated with central management systems like Active Directory, allowing a single point of control for allowing access to reports
- Because setting up a report requires a development, testing, and deployment process, it can enable regulatory compliance such as SOX.

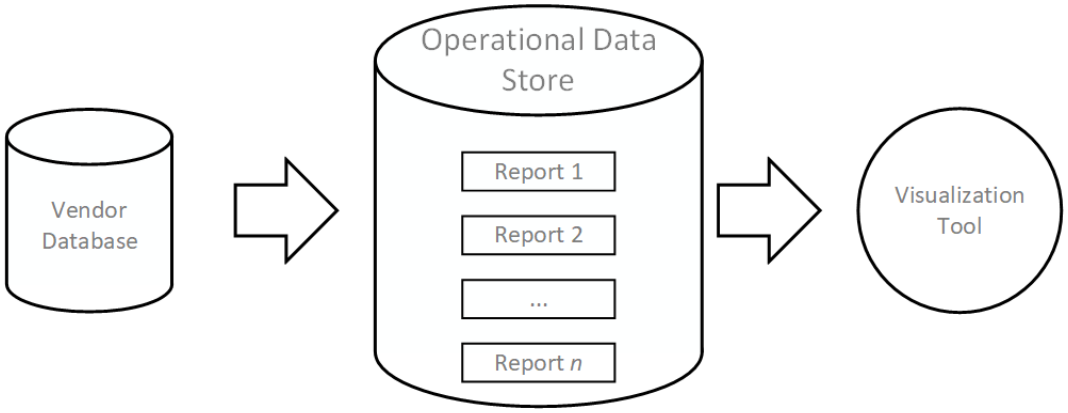
Cons

- Reporting servers require a bit of infrastructure. In particular, a central server with source systems that are sophisticated enough to enable another application to access their data. This may exclude some small desktop business applications. With infrastructure comes the staff to maintain it at well.
- Reports are coupled directly to the source system's structure. If a vendor changes its data model between versions, the dependent reports will break. This coupling therefore increases the complexity of upgrading source systems.

Operational Data Store

An Operational Data Store (ODS) offloads transaction report queries to another system. It insulates reports from the vagaries of source systems. An ODS is indicated when source applications has a challenging data model, or if there is a need for historical reporting that goes beyond what the source applications can provide.

An ODS system also works when users outside the department must access the source systems' data. Externalizing data from the source reduces application license cost.



Pros

- ODS systems can keep data secure
- An ODS design is useful when queries are too complex too run interactively and/or results need to be persisted
- An ODS can capture data snapshots over time
- A centralized ODS server lends itself to a deployment process which can comply with regulatory requirements
- An ODS data model insulates data consumers from vendor-specific details, concentrating dependencies in one place

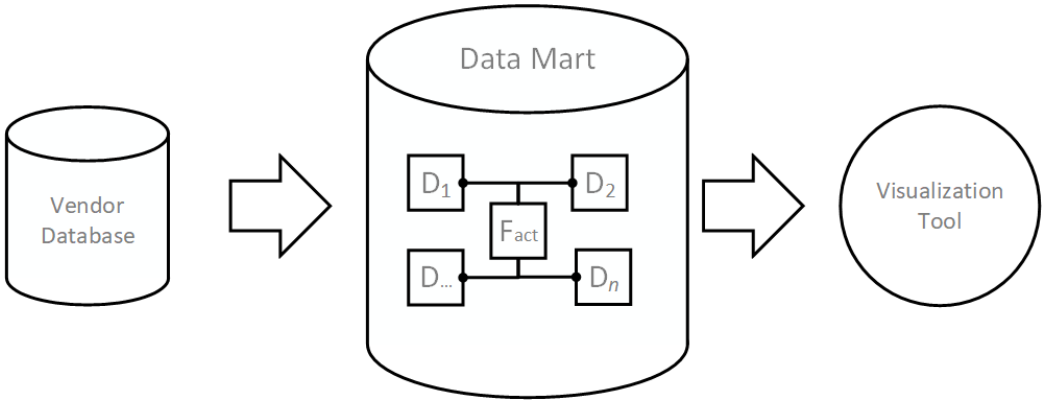
Cons

- An ODS costs to own. Plan on dedicating a person to maintain it.
- A process has to be created to transfer data from the vendor database to the ODS
- historical snapshots consume a lot of storage
- Requires a UI/Visualization tool (which could be perceived as a positive)

Data Mart

Data marts are used for historical reporting needs and are optimized for storing lots of data. They tend to belong to a single department or line of business. Data mart designs usually follow standard design patterns based on Kimball's or Inmon's approach. Dear, reader, apologies up front as I am biased toward Kimball's dimensional approach. However, any responsibly designed solution that addresses your requirements will work. The below example follows a Kimball pattern.

The data mart's vendor-neutral data model design is useful when source system data requires significant transformation to be consumed. Data marts also provide scaffolding to support sophisticated visualization and security requirements. If there is need for historical snapshots, data marts are a natural fit. Following a regular design pattern allows for efficient storage of large data sets.



Pros

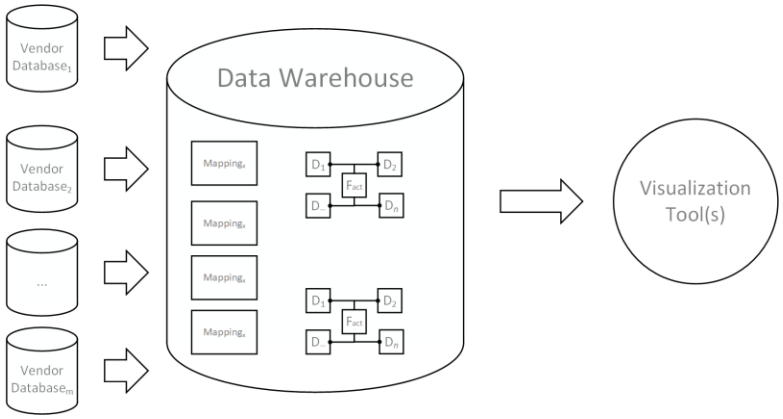
- Historical Snapshots- data marts are usually designed to capture data on a daily or monthly basis and sized accordingly.
- Security- using commercial off-the-shelf databases can provide robust security
- Deployment process-the infrastructure requires specialized skills. in a sox world, specialized skills can promote separation of duties
- Vendor Independent- getting data outside the source system into something you can control relieves some vendor dependency
- Efficient storage- standard data mart design can focus on just what's needed, minimizing redundancy by normalizing data structures

Cons

- Setting up a data mart is labor intensive
- Data mapping from source system data model can be complex
- Querying data can be challenging since data is restructured for efficiency
- A data mart alone only addresses storage concerns; a reporting or visualization tool is needed to make the data consumable

Data Warehouse

Data warehouses are data marts in which the scope spans multiple departments and source systems. They are used for strategic reporting needs that cut across the organization. Data warehouses are a fit when multiple source systems are involved and users specifically need to see data from those source system combined into a single view.



Pros

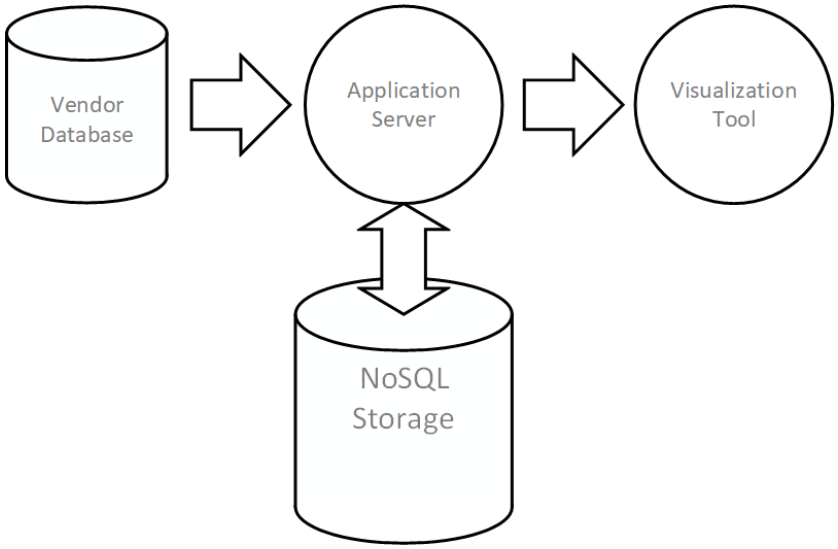
- Can combine sources across the enterprise
- Serve as a single source of truth for an organization

Cons

- Need for common master data or canonical data model
- Complex data model
- Sensitive to data quality issues
- Requires cross-department governance concerning its use and maintenance
- Challenging to overcome multi-department inertia

Data Lake

A data lake is a large capacity system with an emphasis on flexibility and processing power. Data lakes support large amounts of differing types of data, streaming data, or data in which the structure is likely to change over time. Primary drivers are storage, flexibility, and analytics.



Pros

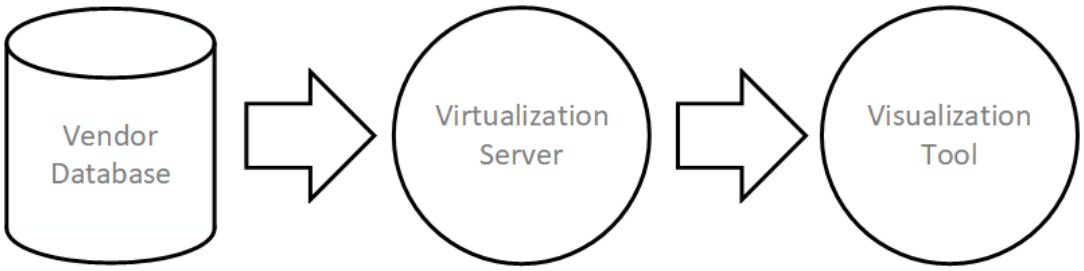
- Ingestion of data is efficient
- Tools can query across large amounts of data efficiently
- Many cloud providers provide data lake platforms as a service

Cons

- Platforms are eventually-consistent nature rather than ACID
- Tooling can be a bit complicated as there are so many players in the space

Data Virtualization

Used for reporting across systems while leaving data in place.
Data is needed from multiple systems, but historical snapshots are not a concern



Pros

- Usually fast to implement
- No data modeling of source system, all modeling done as queries

Cons

- Performance can be a concern if queries are complex
- No snapshot capability

Conclusion

When considering a reporting solution, choose the simplest solution that still works. Invest in building for what you know. Don't build for what might happen. Instead, build for the requirements that have been voiced by the stakeholders.