International Research Journal of Computer Science (IRJCS)
Issue 02, Volume 6 (February 2019)

ISSN: 2393-9842
www.irjcs.com

LEGACY DATABASE SYSTEM TRANSFORMATION WITH DATABASE OPTIMIZATION IN PUBLIC SECTOR

Ravishankar Belkunde
Senior Database Administrator & Researcher
Jefferson City, Missouri, 65109.
cloudsarchitect@gmail.com;

Manuscript History
Number: **IRJCS/RS/Vol.06/Issue02/FBCS10082**
Received: 03, February 2019
Final Correction: 13, February 2019
Final Accepted: 24, February 2019
Published: February 2019
Citation: Ravishankar(2019). Legacy Database System Transformation Database Optimization in Public Sector IRJCS:: International Research Journal of Computer Science, Volume VI, 33-37.
doi://10.26562/IRJCS.2019.FBCS10082
Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India

Abstract: Most of the public-sector systems in United State of America are using legacy database systems. The study has been undertaken to investigate the impact of legacy database system in Public Sector. This paper presents research on legacy database system transformation challenges and strategies to overcome them.

Keywords: Public Sector; Legacy Database System; Database Optimization; Legacy; Data Migration; Data Transformation;

I. INTRODUCTION

The data is growing at exponential rate. The legacy systems performance is not at par to handle the data growth, it has become challenging for legacy systems to handle and process huge database. Historically 38% of the data migration projects have failed [1] it is critical to transform the legacy database systems in feasible way. In this research paper we will explore the impact and challenges of legacy systems in public sector and how to overcome those challenges.

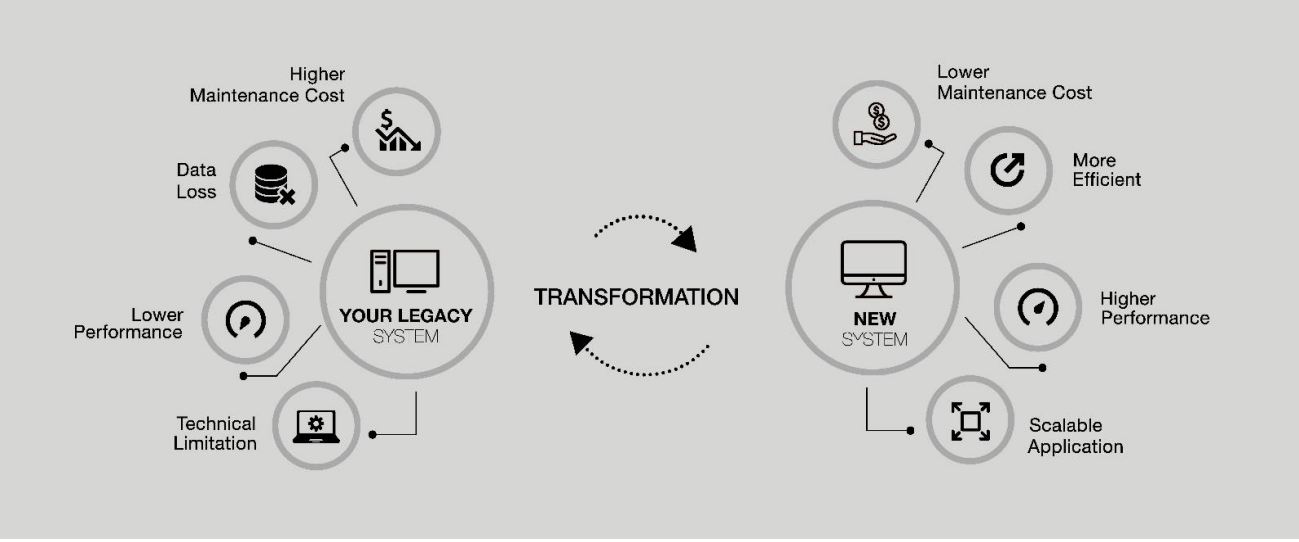

Fig 1. Legacy Database System Transformation Diagram

II. IMPACT OF LEGACY DATABASE SYSTEM IN PUBLIC SECTOR

I. **Performance Degrade**: In public sector data always grow and keep on growing, most of the applications are required to maintain public records and history of the old records. Legacy database systems are good with less data as the data volume grows systems performance starts degrading. Most of the public-sector legacy systems were not designed to handle massive data, for example to generate one-year data report is quicker but for ten years data the report generation time is doubled. With years of growing data, report generation time and system output capacity has degraded significantly.

II. **Makes Security Worse, Not Better**: The security vulnerabilities can't be address with limited fixes and patching, with new ever-growing security threats applications need continuous updated to meet the latest threats, public-sector legacy systems by their nature struggle with this because of their age. They may not be as easy to fix due to the large, inflexible nature of older database systems and outdated code. In public sector even if there is fix, the patch is typically delayed as it is much more difficult for developers to create legacy fix and far lower on the priority list. These systems are most vulnerable to security threats.

III. **Compatibility Issues**: There are many applications in public sector who uses old formats. These legacy systems typically support file and data formats up to a point and then those formats evolve beyond what the legacy system can handle in short span of time or so. Thus, these systems are stuck using older formats which their vendors, partners or public may not be able to use. Integration of these systems with the latest systems becomes difficult and expensive.

IV. **Not Cost Effective to Manage**: It appears like legacy systems should be less expensive, but over time that becomes less true. Support and updates for old systems is usually much more expensive than support for the current model because it takes a lot more work for developers and system administrators to offer continued maintenance and updates.

V. **Proprietary Limitations**: The public sector proprietary systems tend to be very large, clunky where legal rights are ensured by copyrights, which makes them difficult to change and customize where as the new systems are far more flexible and made of smaller parts that are easier to adopt on as need basis.

VI. **Compliance Issues**: Compliances are utmost import in public sector. New IT standards and compliance norms are introduced with advancements of the database and software systems. The legacy systems were not designed to meet the new IT standards and compliances. Thus the legacy systems are not compliant with latest compliances.

III. LEGACY DATABASE SYSTEM TRANSFORMATION CHALLENGES

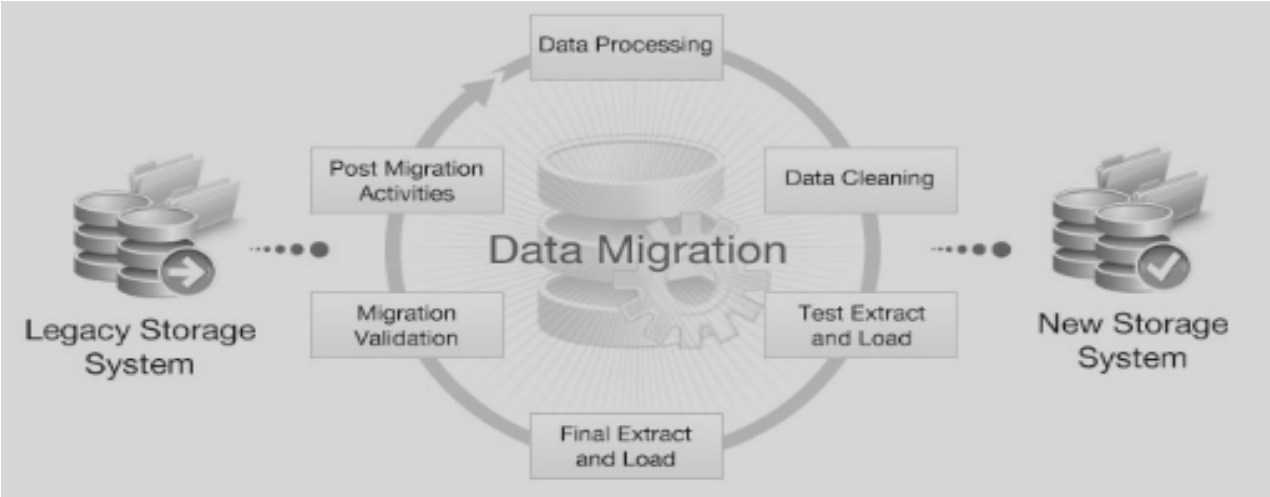

Fig 2. Data Transformation (Migration) Overview Diagram

I. **Expensive Process**: The public sector agencies are very tight on budget, it is tedious process to get fund approval for major system upgrade. The legacy system users are settled with their existing system creating their own shortcuts and processes in or around the system usually oppose any change. The more money spent on the legacy system the harder it can be to convince higher authorities to spend further. The transformation involves new licenses, hardware, implementation, development, data migration and training its expensive process, the future cost savings may take some time to materialize.

II. **Time Consuming**: The transformation involves multiple steps, from detail analysis of business needs to comparing different solutions and from hardware capacity planning to after implementation training. These initial steps are very crucial and can't be rushed as they define core framework of new database system, after funds approval public sector look for major change in short time span this could be a major bottleneck.

III. **Unknown Dependencies**: In public sector most of the applications share data and there are multiple small applications dependent of each other. It becomes difficult to identify those dependencies the easiest way to identify them is by tuning down these systems temporarily to see any alerts or complaints come in. If other systems interact seldomly like once in month or quarterly, identifying those dependencies becomes more challenging.

IV. **Legacy Code**: The legacy database systems have complex code, and lacks developer's familiarity it becomes challenging to new developers to understand the code. Most of the legacy systems lack proper documentation of the architecture and change requests. Over the period many changes occur to the overall software and database making it challenging to maintain and develop new system without breaking it.

V. **Service Disruption**: There will be downtime during the database transformation process. Depending on the complexity of the legacy system, database size and interdependencies the downtime could vary this may impact public sector critical processes, it becomes challenging to carefully consider the timing and scope of the change to keep the downtime to minimum.

VI. **Integration Issues**: The new system integration with the existing to be upgraded legacy system is one of the biggest challenges. In public sector it may happen they get fund approval to transform one system but not the other which are dependent on each other. The challenge is to transform the system by keeping backward compatibility and future system integration.

VII. **Data Integrity and Corruption**: While legacy database system transformation the format and content of data may differ, the new system may not use the data the way legacy system used. During the data migration process this may lead to data corruption. Identifying the new system structure and transforming the data is critical and challenging. New fields and data interdependencies exists in new system while migrating data redundant and unwanted data may get migrated, especially if legacy system have multiple input steams of data flow arising data integrity issues.

IV. LEGACY DATABASE SYSTEM TRANSFORMATION STRATEGIES

I. **Existing Resource Optimal Usage**: If old hardware is in good state and have exceeding processing capability hardware cost can be cut by reusing old hardware, most new database systems are compatible with the old hardware. While defining the new system it's good to consider usage of the latest version of existing evolved technologies than going for completely new technologies this will reduce the cost of hiring new resources. Existing staff can easily be trained and transit to use and support new system.

II. **Return on Investment**: During transformation it's critical to invest in open standards-based technology, which allows maximum ability to port existing assets as well as ensures future extensibility and portability of future software. When the cost of maintaining legacy system and its throughput outweighs the actual cost of refactoring or rewriting the applications with modern architectures on next-gen systems, it's time to pull the plug, most of the time transformation benefits are underestimated. The software licensing, hardware infrastructure and integration with other systems becomes less costly and easier. Application performance is enhanced which deliver large productivity gains and flexibility to use more productive development tools and large skills pool.

III. **Detail Data Analysis & Validation**: A good understanding and study of legacy system data and system structure is needed for smooth quick data transformation. Thorough detail data analysis should be done at the earliest possible stage to uncover hidden data in obscure places, because often there aren't specific fields to hold all elements of the data or users may not be aware of the purpose of the available fields this will result in incomplete, inaccurate and outdated data transferred during the transformation, often discovered very late sometimes even after project completion causing data integration issues. Sample, subsets or complete data set validation should be put in place, early data validation transformation with actual data would avoid data corruption. In the final stage comprehensive data evaluation should be performed by defining test cases and collaborating it with real time users and subject matter experts to identify the inconsistencies and incompatibilities between the transformed data and the parameterized new system. Following diagram gives overview of this process.

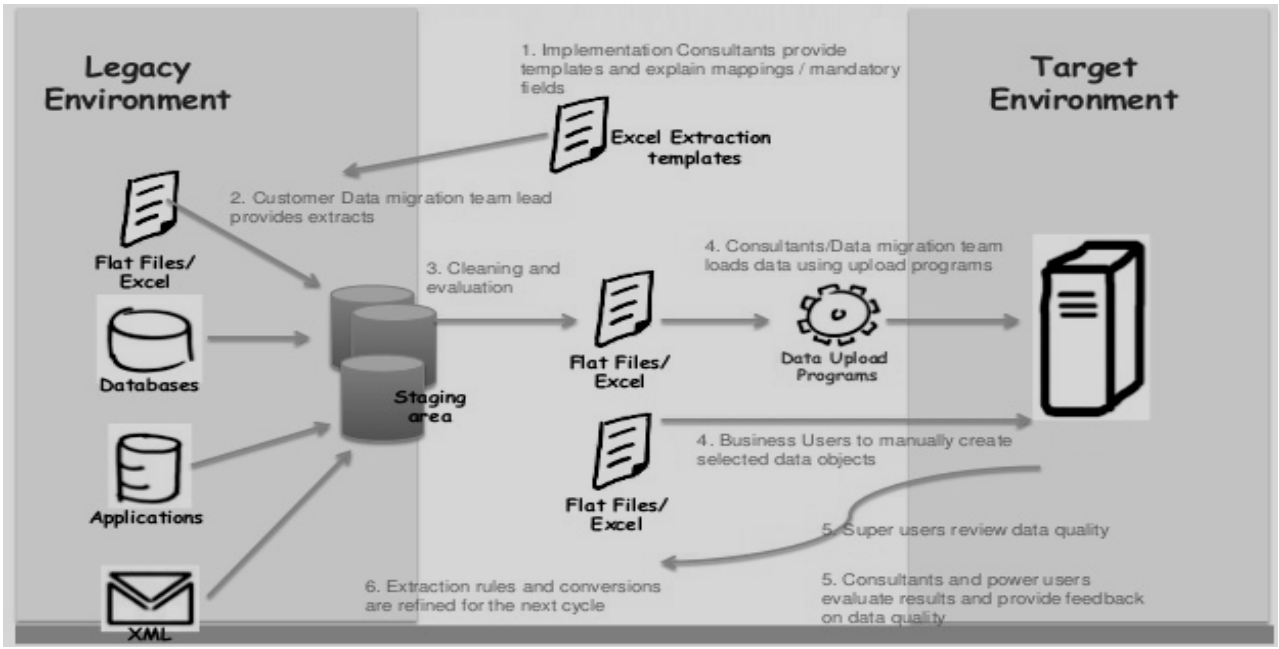

Fig 3. Detail Data Analysis, Validation, Transformation SAP Legacy System Diagram

IV. Downtime Avoidance: The database transformation system can come with significant downtime leading to temporarily lost revenue. During the system transformation process downtime could be avoided by releasing the new system in phases and by using data migration tools, advanced database high availability and replication techniques. For example we could use Golden Gate replication technology to give zero downtime during transformation. Golden Gate enables homogenous/heterogenous database live data replication from one database to another. Following figure depicts the architecture and topology of the golden gate implementation.

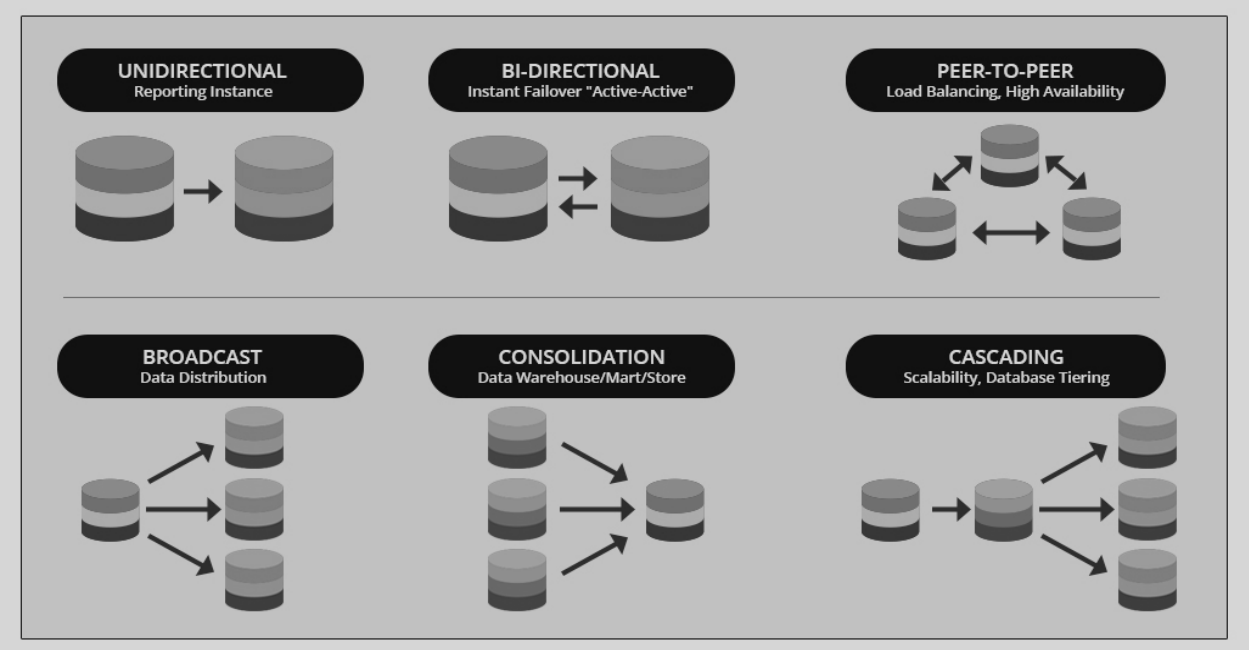

Fig 4. GoldenGate Topology

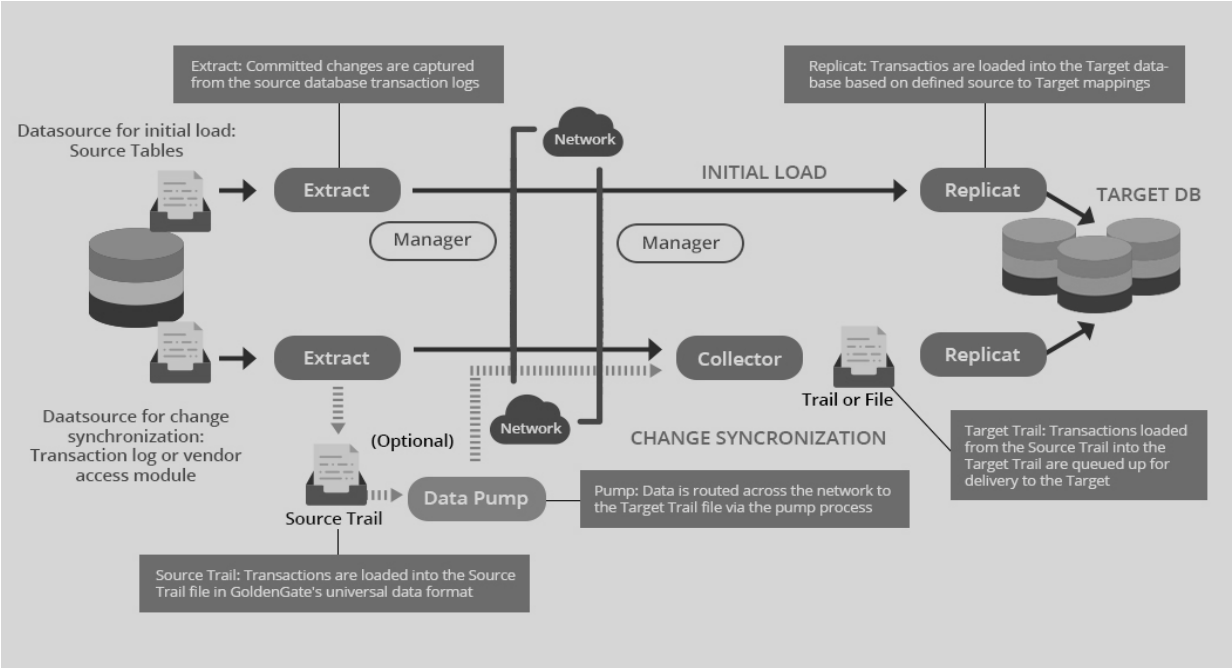

Fig 5. Golden Gate Architecture used for Database Replication during data transformation

V. CONCLUSIONS

In this paper we can conclude that legacy systems have huge negative impact in terms of cost and productivity in public sector and this legacy database system transformation with database optimization is a challenges process that could be overcome by using appropriate database transformation strategies.

REFERENCES

1. Data Migration – 2011, A White Paper by Bloor Research, Author : Philip Howard, Publish date : December 2011
2. Jason Williamson Oracle Information Integration, Migration, and Consolidation Publisher: Packt Publishing Release Date: September 2011.