

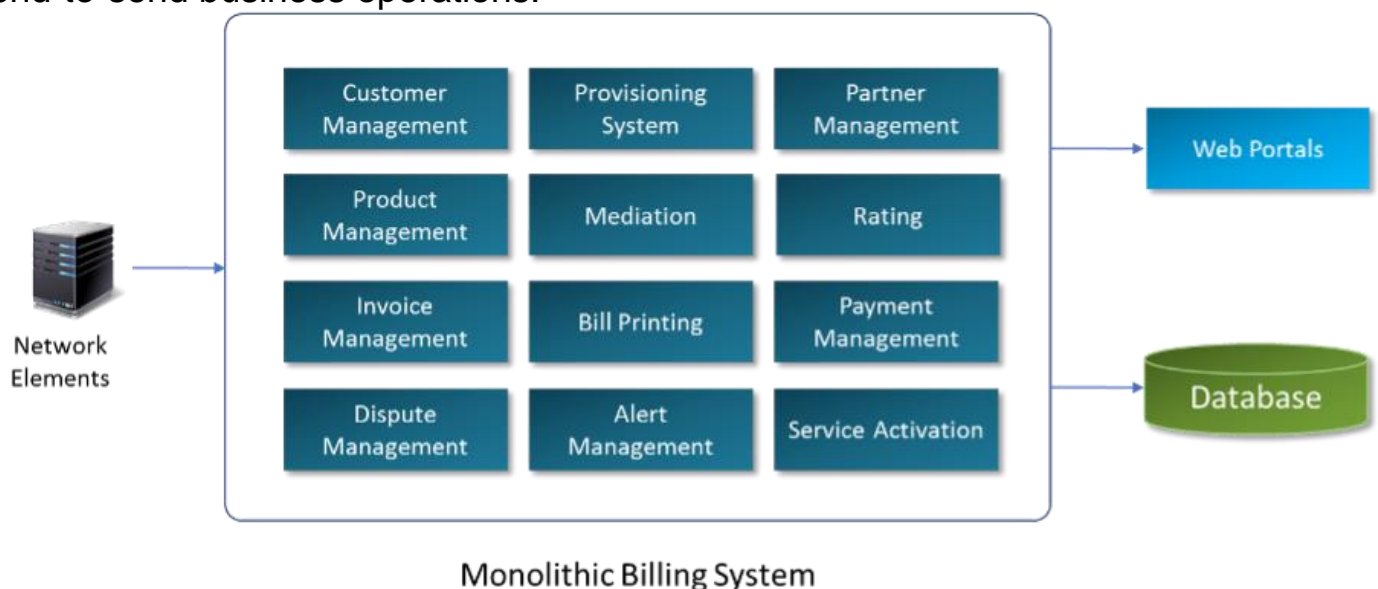
# Why Enterprises Must Transition from Monoliths to the Small-B Architecture

By Rajan Babu, Group Manager- Product Management, SunTec Business Solutions

Business processes across enterprises rely heavily on operational agility of applications and the robustness of its architecture. But when systems become large monoliths, governance becomes extremely challenging, and businesses lose agility and scalability. Further, the monolithic nature of applications makes the cloud journey almost impossible, making it difficult to get to the cloud, and use cloud native services. And even if organizations reach the cloud, achieving operational efficiency becomes a nightmare.

## Understanding Monolithic Applications

Monolithic applications are typically designed to provide multiple related functionalities; they are complex applications that encompass several tightly coupled functional modules. The enterprise applications used by banks and telecom service providers for pricing and billing are classic examples of monolithic applications. These so-called billing systems tend to provide all functionalities required to run end-to-end business operations.

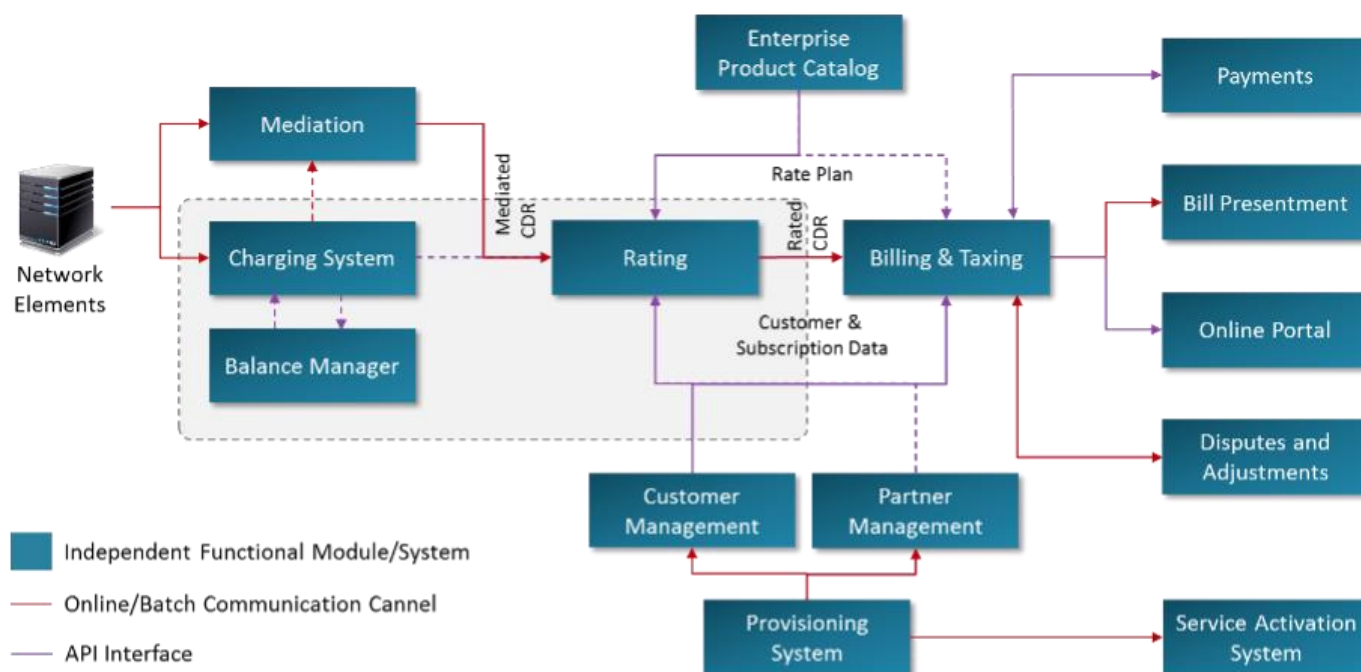


Monolithic applications consist of many cohesive modules, and it is easy to develop, test, deploy and scale at the initial stages of business. However, the ease of managing monolithic application comes with some obvious drawbacks, such as:

1. Large code base: Monolithic applications tend to have a huge code base and making a small change in a function requires compiling and testing the entire application, making it less scalable in the long run.
2. Long deployment time: Making even a small change must go through a rigorous release and deployment cycle, which increases deployment time.
3. Lack of options to scale by functional units: Horizontal scaling can be applied to a monolithic system only in its entirety. Scaling just to the affected module is not feasible.
4. Locked into a technology stack: Monoliths may result in a suboptimal solution since there are limitations in terms of adopting the most suitable technology for a particular functionality.

## Why Opting for the Small-B System Can Be More Efficient

Small-B typically breaks down the monolithic billing system into small independent systems that interact with each other to achieve end-to-end functionalities. In a Small-B architecture, each component of the ecosystem will have a limited set of functionalities that can be integrated with other systems through online or offline channels. For instance, the billing system will only do the bill preparation using the inputs it receives from the supporting systems.



Small-B Architecture

The modular and service-oriented approach of the Small-B architecture offers numerous advantages. The independent nature of the services allows each service to adopt technologies suitable for its functional and non-functional behaviour. Each service can be independently deployed, scaled, and managed, without impacting

other components in the ecosystem. Each service has its own data store based on the data it is handling, which ensures the data is distributed and manageable.

The services interact with each other using an online or offline communication channel. The online communication can be done using Restful APIs, while data streams can be used for offline or near-real-time communication.

A well thought and designed architecture is crucial to the success of the Small-B architecture. The selection of the subsystems, technologies, data storage, communication channels, the design of availability and recoverability are all critical components.

In a monolithic system, service providers depend on the billing systems to make any changes to the customer experience, and it is always costly and time consuming. In case of Small B, the customer experience can be designed upfront, and the best subsystem can be chosen to provide the customer experience. Alternatively, the customer experience can be built using the services provided by the subsystems. Such an architecture enables service providers to own their customer experience rather than leaving it to the billing system. Better customer experience ensures customer stickiness, which has a greater impact on the revenue and bottom line.



**Small Inter-connected  
Systems**

Individual systems can be independently managed, which makes the maintenance easy



**Easy Upgrade and  
Replacement**

The individual systems can be upgraded or replaced without affecting the surrounding systems



**Fast Technology  
Adoption**

Since each component is independent, they can adopt different technologies based on their behavior

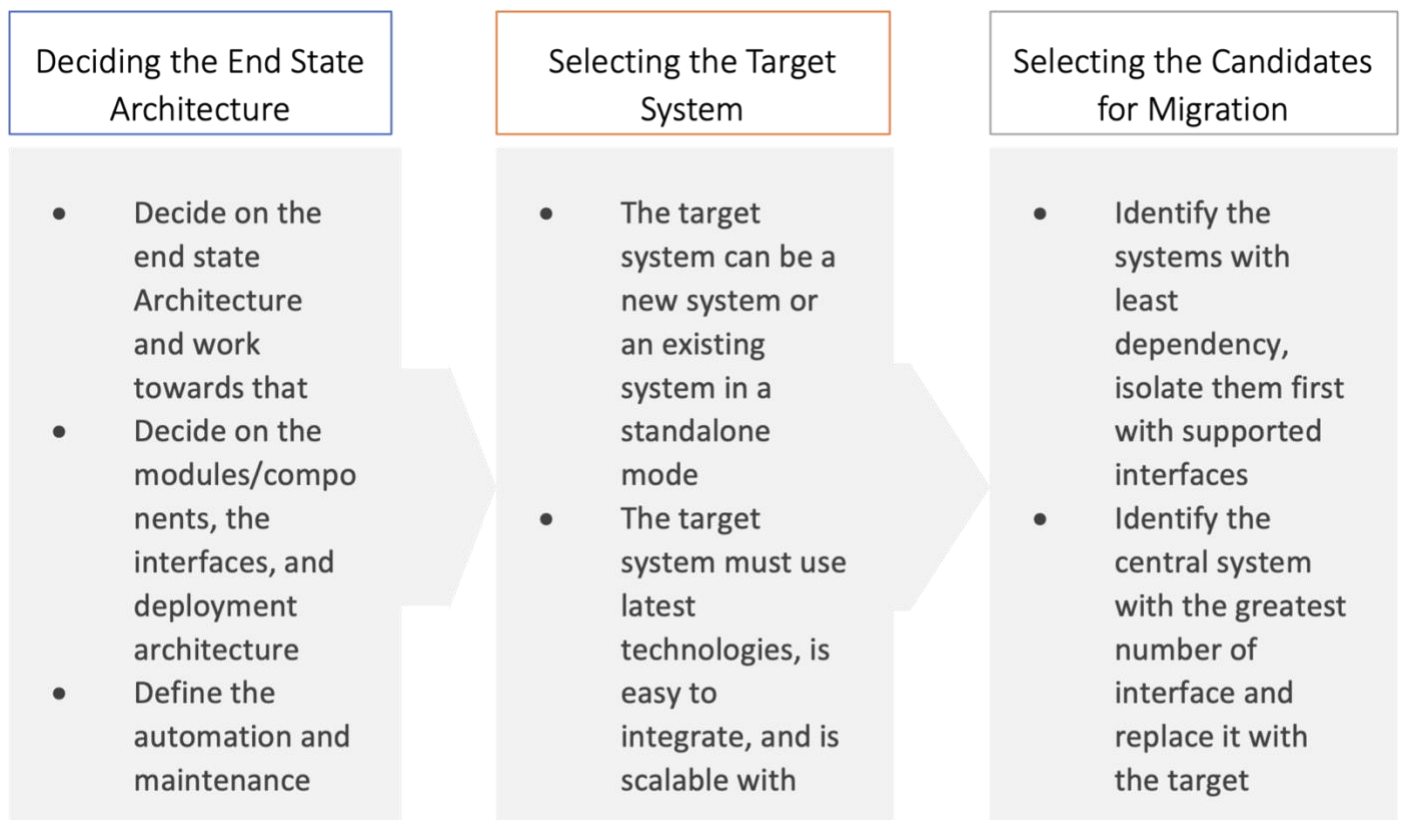


**Distributed Data  
Management**

Since each system manages its data, we can go for distributed heterogeneous data stores with HA and DR

## **The Journey to Small-B**

The journey from a big monolithic billing system to a Small-B architecture must be a step-by-step process, where each step takes it closer to the end state, by making at least one functional component independent and interfaced to the rest of the ecosystem. It is better to avoid a big-bang approach since the chances of failure are high.



The key is to decouple the system by identifying independent subsystems and their integration points. These subsystems can further be divided into functional modules, depending on the level of abstraction required. Each of these subsystems or modules can be replaced independently, making sure that the integration points are intact.

In the billing domain, the billing module is the central system that interacts with most of the other subsystems. To replace the billing system, make sure that the new system can integrate with the other subsystems without or with minimum changes to the existing systems.

### *Rajan Babu*

Rajan Babu C leads the product management practice in SunTec for Digital Communication Systems.

He is a dynamic, result oriented IT professional with over 18 years of professional experience in Product Management, Product Design, Delivery Management, Program Management, Project Management, Solution Design, and Product Consulting.

He has a track record of delivering business and technology solutions to multiple clients with key focus on client satisfaction. Trusted to carry out major initiatives, he reliably manages multiple projects and vital systems.