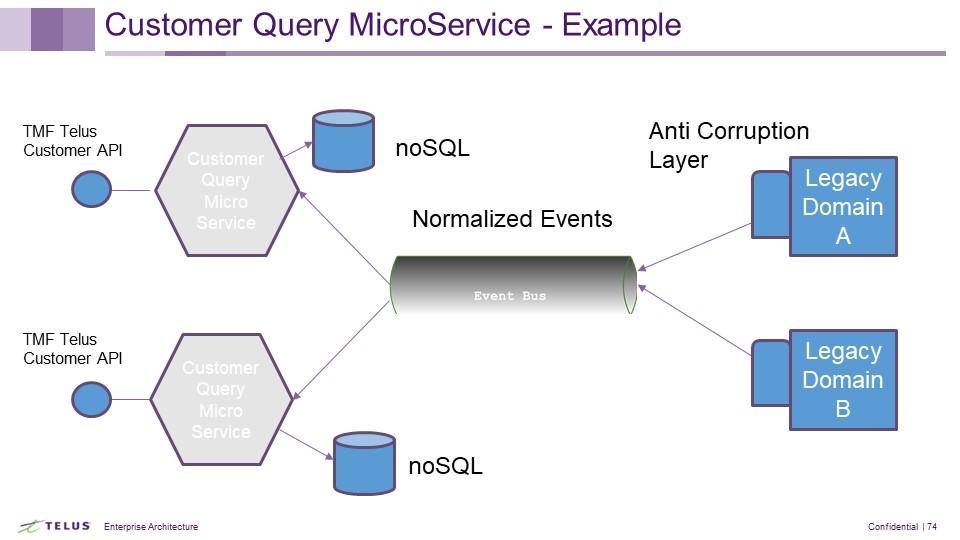
|  |  |
| --- | --- |
|  | Reference Best Practices and Design Best Practices for design and development can be found in the API Best Practices for Design and API Best Practices for Development documents. API Reference Architecture Use Cases Two domains were used to show the application of the API Reference Architecture principles and integration patterns to a specific TELUS use case. The information for each can be found in the API documentation portal.   * [Appointment Domain](https://drive.google.com/drive/folders/1OKoP6jvq_lrVPHbU4MQ7zAW0Tg8lgTZr) * [Communications Domain](https://drive.google.com/open?id=1KrMStzwOpITNAS0S4i-AE4pVVQqtxOlU)  Applying the API Reference Architecture Principles and Patterns The following describes the steps to apply the principles and patterns of the reference architecture to transform the customer management.   * Create Microservices to expose Customer Data using the industry standard interface specification   + Use the industry standard Open Data Model for customer   + Apply the industry standard Open Model Extension Pattern to include TELUS specific attributes   + Provides for an easily consumable API by developers and architects * Segregate Query and Update functions into different Microservices * Expose normalized events aligned with industry standard event model on the event bus * Create master and slave microservices and integrate them via Events published and consumed on the Event bus |

## Migration Steps

* Define the uniformed customer shema
* Define the structure of the normalized event
* Create uniform customer query functions for each customer data source for example
  + Uniform Customer Query Function – Customer Query Microservice – Customer DB Source 1
  + Uniform Customer Query Function – Customer Query Microservice – Customer DB Source 2
* Create the customer event function and event microservices
  + Uniform Customer Event Function – Customer Event Microservice – Customer DB Source 2
  + Uniform Customer Event Function – Customer Event Microservice – Customer DB Source 2
* Create the Customer Management Microservices
  + Single Source of Truth – Customer Management MicroServices
  + Segregate Query and Update
* Customer Management Microservice (DB) published to the event Bus
  + Events are subscribed to by
    - MS Customer DB Domain 1
    - MS Customer DB Domain 2

