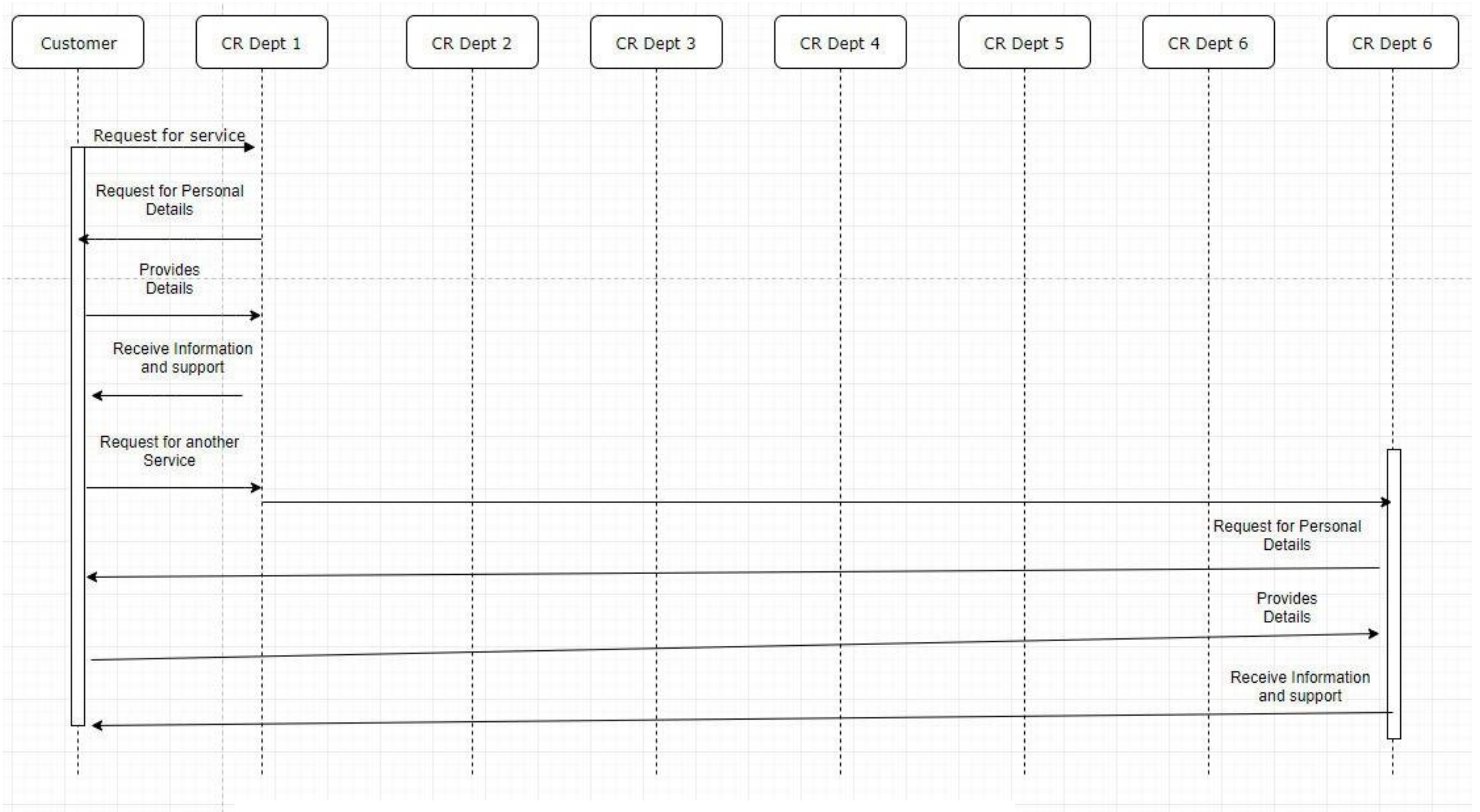
Problem Space

Problem Analysis: This case discusses about The Integrated Trading Company which has recently acquired a few companies. Currently the firm has seven different applications and for each application they have different databases to manage their client's data. The company offers 7 different services to each client and each application has its own specific database and application interface. Due to this problem, there is no connectivity within these services and if a client wants to use multiple services when on call with a customer care representative. They need to repeat all the information whenever the call is transferred from one service department to another and CR do not have data related to other services.

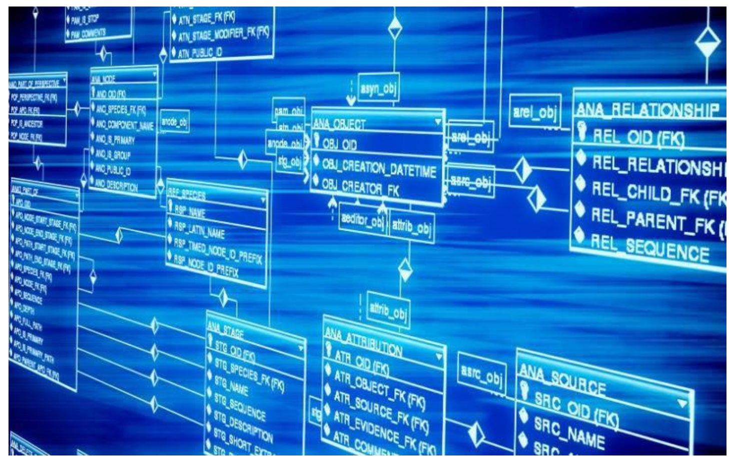
What is the reason for the problem?	**The company hasn't integrated the 7 applications and their databases. The customers need to repeat all the details every time call is transferred to another service department.**
How is the Problem caused?	**Separate database for each service. No synchronization. CR representative is experienced only in their specific department.**
Why is the Problem caused?	**Multiple Mergers and acquisitions which lead to expansion of business and the company did not think much about the software end. Each service has different business needs and resources hence takes time and skill.**

Current System's Sequence Diagram


Customer
CR Dept 1
CR Dept 2
CR Dept 3
CR Dept 4
CR Dept 5
CR Dept 6
CR Dept 6
Request for service
Request for Personal Details
Provides Details
Receive Information and support
Request for another Service
Request for Personal Details
Provides Details
Receive Information and support

Goals and Objectives

- To integrate the 7 applications and their databases
- To improve customer support and increase customer satisfaction
- To get rid of redundancy in the database and have a better Application which supports easy increment and decrement of modules



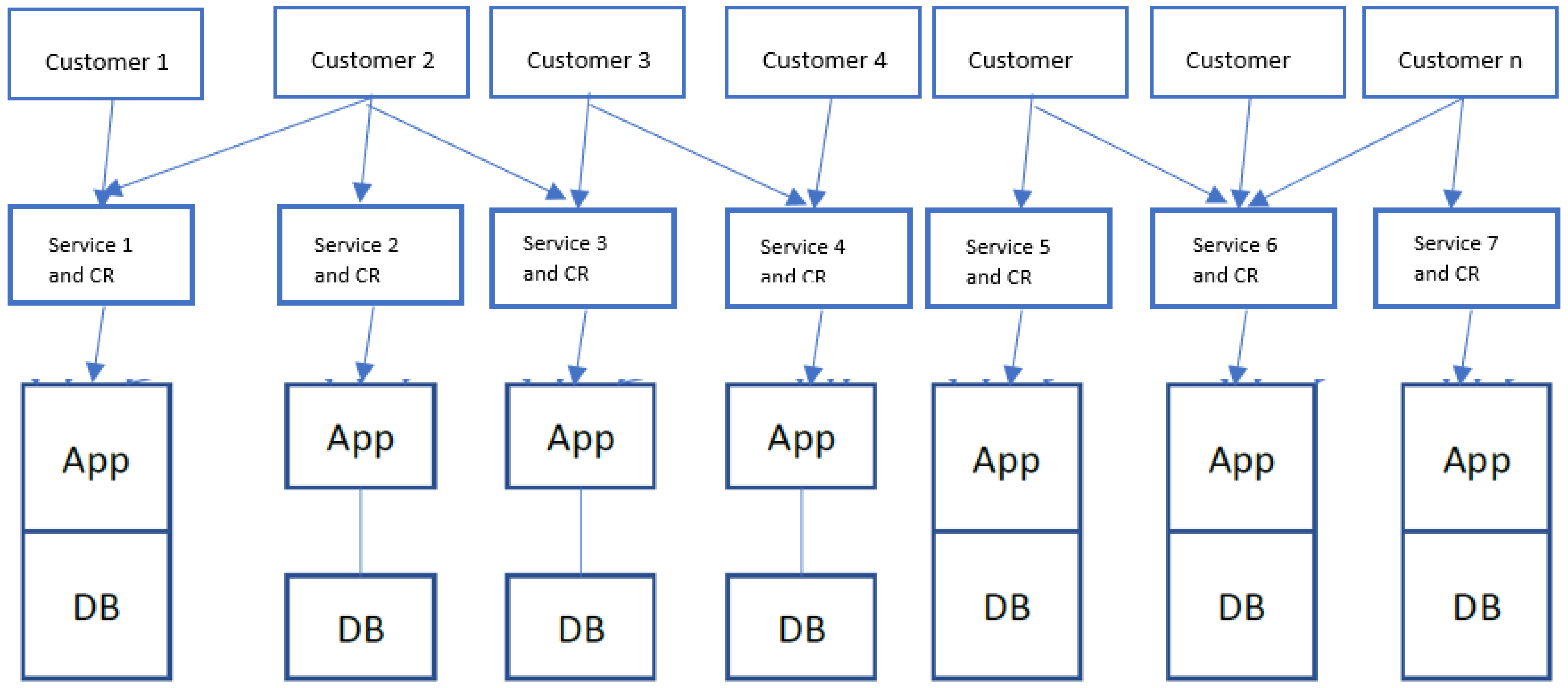
Constraints

- To offer a seamless integration service across all the 7 applications which have different application based code and systems
- To get rid of redundnacy and inconsistency in database due to different database formats

Challenges

- To integrate the databases without a data warehouse
- To keep the legacy systems intact without affecting them during this process
- To integrate various type of file systems and application together
- To complete the project in less time to keep up with customer needs

Current System



Customer's are frustrated since they need to repeat details and databases are not in synchronization

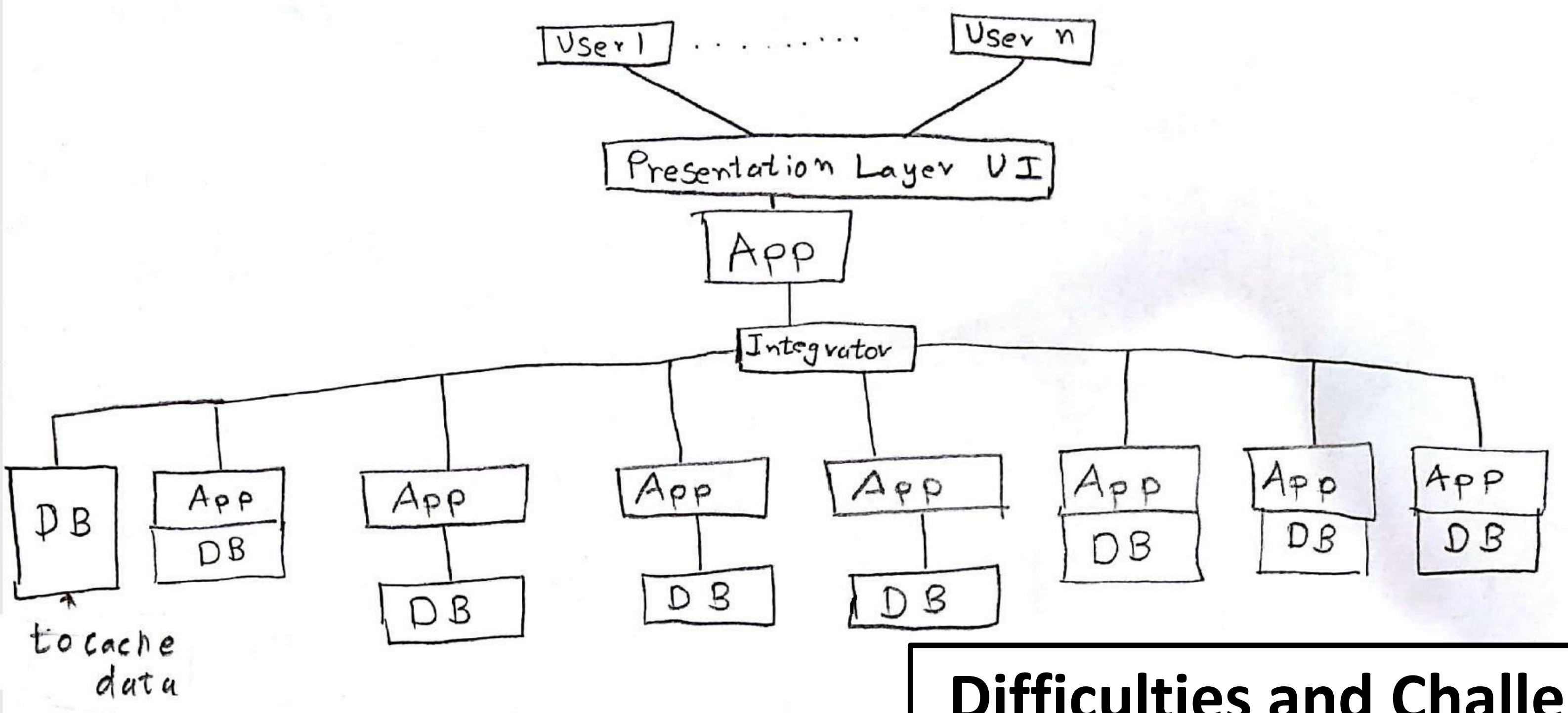
Observations:

- **The 7 services have completely different application and database**
- **These 7 applications have Legacy code and must stay intact. Each Service Database is controlled by its application that involves large amounts of code. Nearly 100,000 lines of code of JAVA/COBOL etc. These systems are either loosely or tightly coupled with their database.**
- **A survey on Customer Usage says 40% use 1 service, 30% use 2 services and the rest 30% use 3 or more services.**
- **Coding:**
- **The database gets Less than 12 updates a year for customer information. Typically, customers that use more than one service will not do so within the same call.**

Solution Space

This Slide covers point 1,2,5,6

Solution 1:

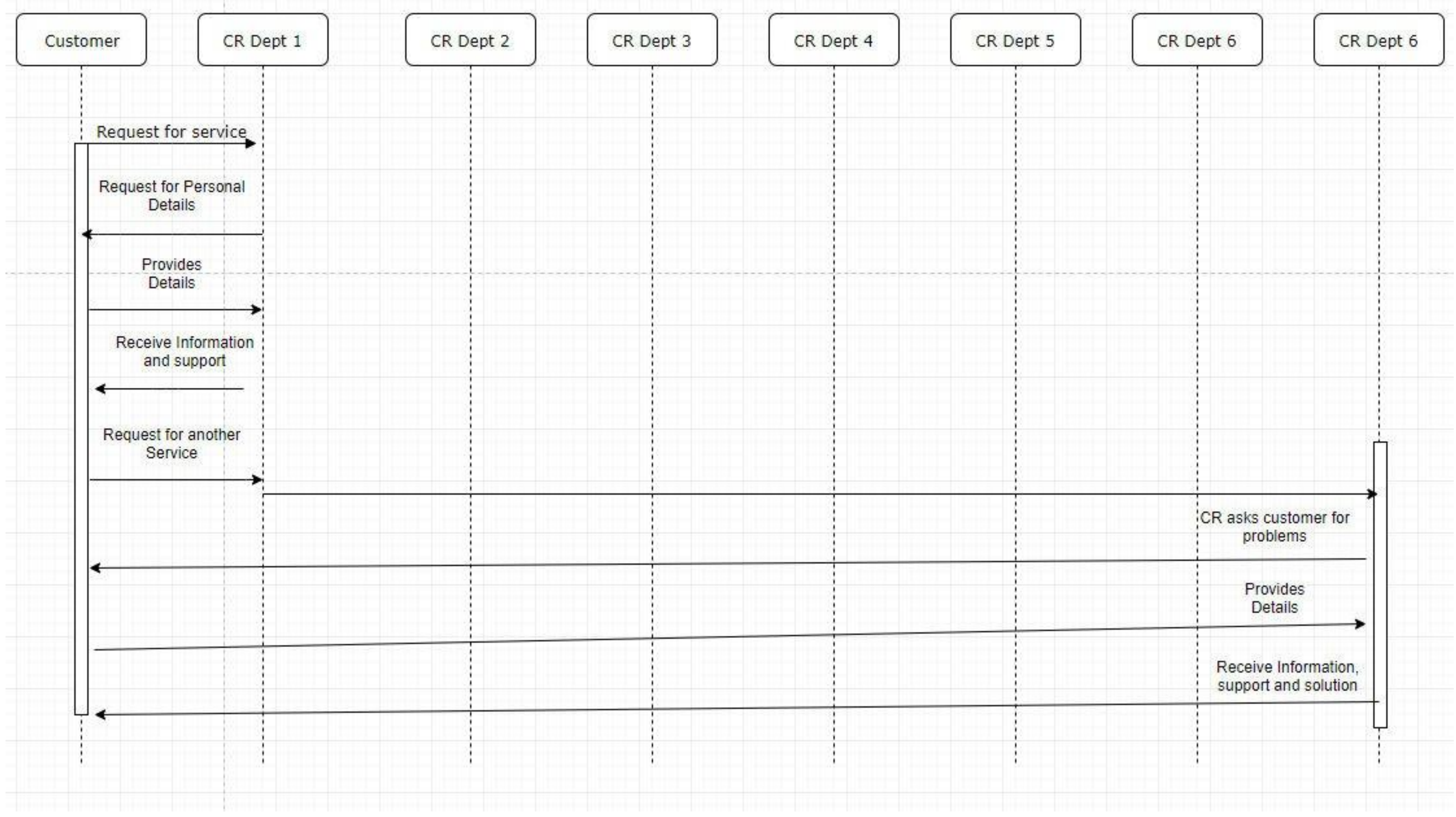


Explanation:

- **This System will consist of a hybrid language mechanism which will be interacting with all the 7 applications on their user interface level and add an extra code to access the database and to merge the database into a joined table and store it into the extra database.**
- **The integrator will also have a User Interface layer above it to let Customer Service Representatives access any service and will be easily able to pull out details from any service.**
- **The cached database has a joined database of all the applications,, hence employees from any department can make changes to customers of any service.**

Difficulties and Challenges:

- **It's extremely difficult to bring this system to work on a live environment since it involves the integration layer to understand multiple languages and various types of databases.**
- **If the integrator system fails at any point of time all the services would be shut down. At such a point of time the firm would have to start to use the legacy systems to work with.**
- **It is extremely difficult to get access to closely coupled systems since, getting direct access to the database would change the legacy systems. In such cases we need to add modules to these systems to create connectivity to the integrator system. This modules would be like creating an API.**
- **The frontend application will need an additional database to maintain a copied database to be a backup when integrator fails**


Customer
CR Dept 1
CR Dept 2
CR Dept 3
CR Dept 4
CR Dept 5
CR Dept 6
CR Dept 6
Request for service
Request for Personal Details
Provides Details
Receive Information and support
Request for another Service
CR asks customer for problems
Provides Details
Receive Information, support and solution

Procedure:

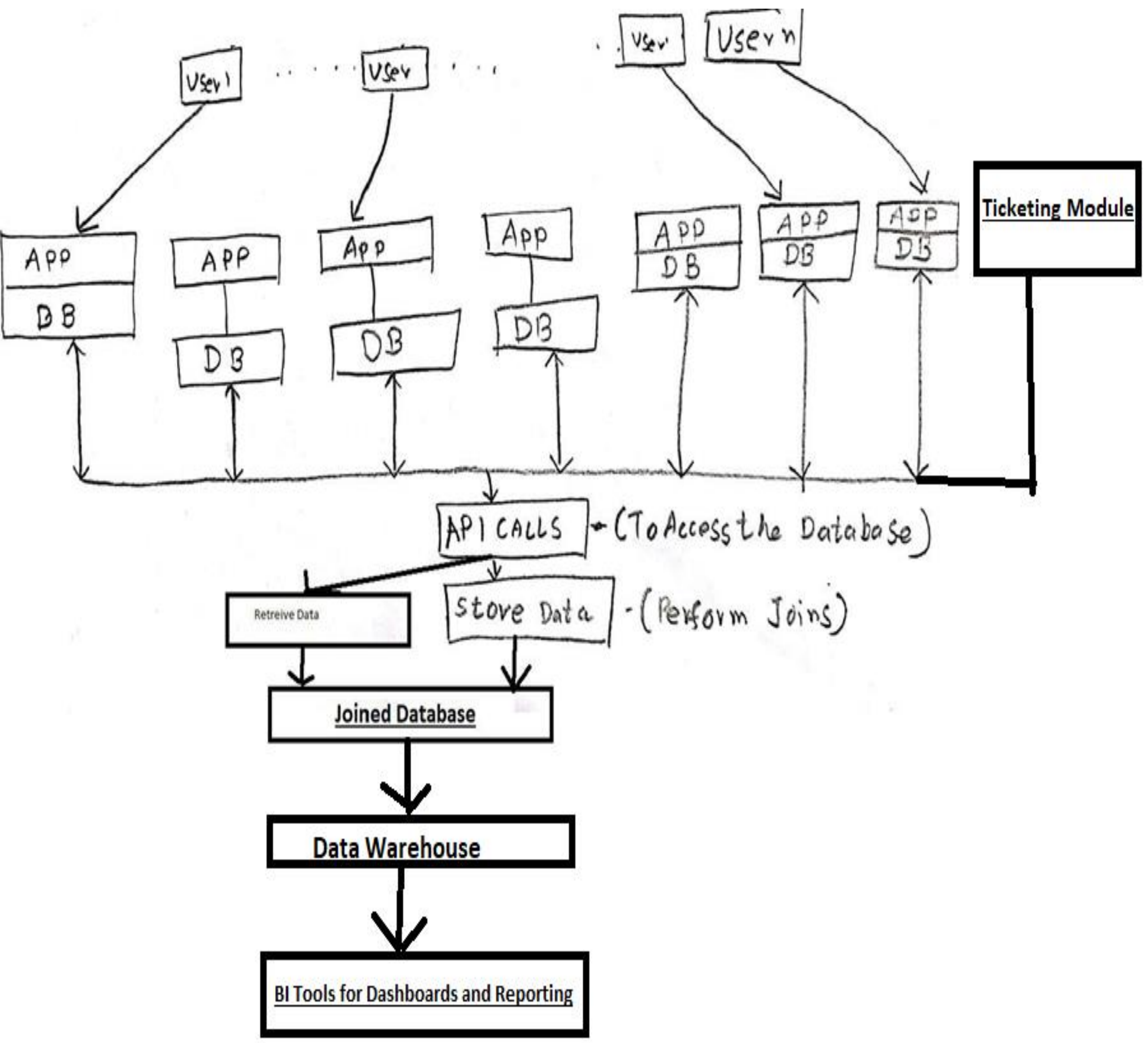
- **Customer calls the Customer Care for Service.**
- **The Customer Care representative asks for Personal Details.**
- **The Customer Provides the credentials/details and**
- **issue/problems.**
- **The Customer receives the information related to the concerned service. (if it's a change its pushed to all the applications since the cached database has a map of all the connected applications and its customers hence they do not need to contact other departments for changes).**
- **The customer requests for another Service**
- **Call is forwarded to another Service Department**
- **Customer Representative asks customer for problems (since everything is in the same application)**
- **The customer provides details**
- **The customer representative provides information, support and solution.**

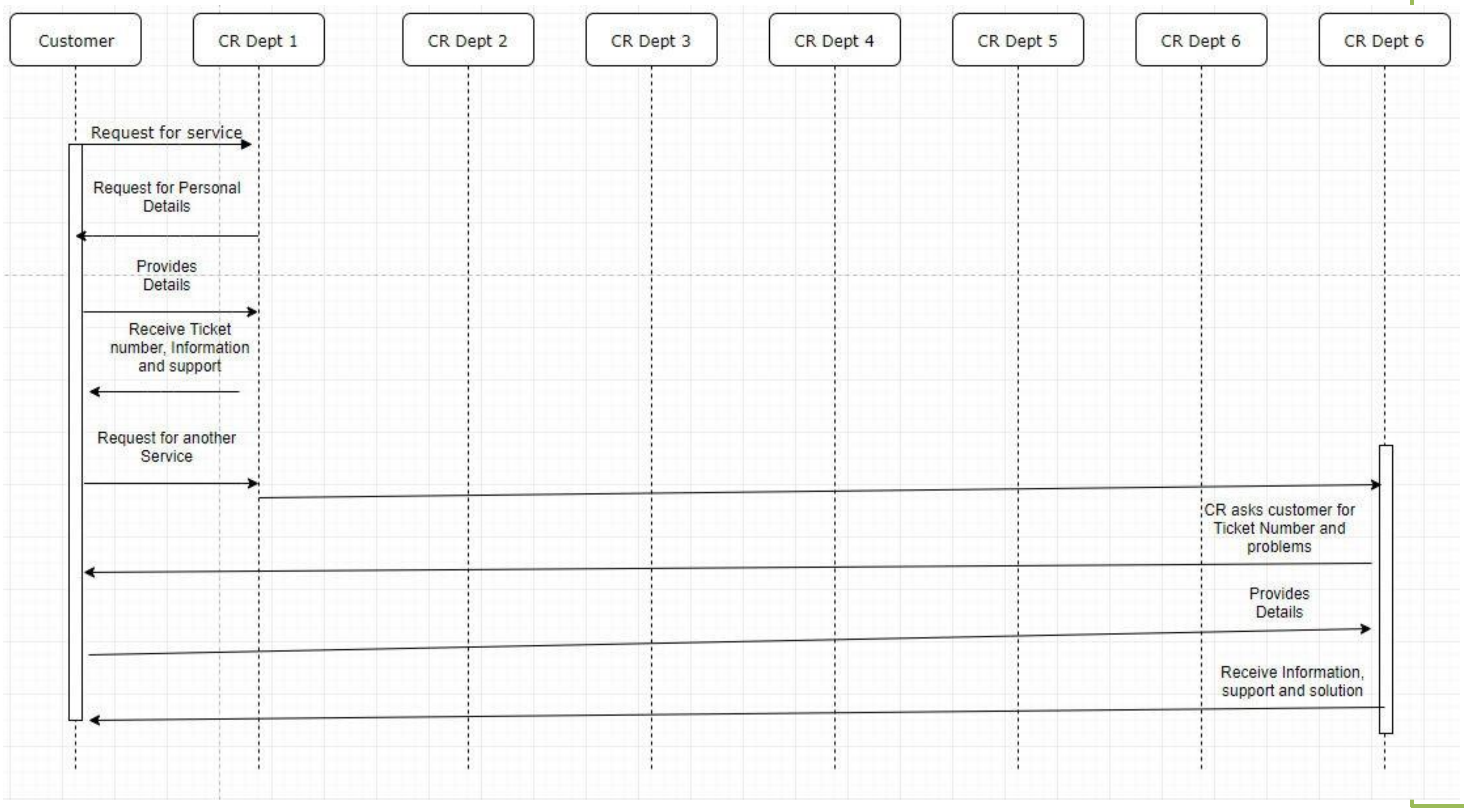
Solution Space

This Slide covers point 2,3,4,5,6,7

Solution 2:

Preferred Solution




Customer
CR Dept 1
CR Dept 2
CR Dept 3
CR Dept 4
CR Dept 5
CR Dept 6
CR Dept 6
Request for service
Request for Personal Details
Provides Details
Receive Ticket number, Information and support
Request for another Service
CR asks customer for Ticket Number and problems
Provides Details
Receive Information, support and solution

Explanation:

- **This System will make API level calls to access the various types of database, and perform join operations to integrate all the data into a single database.**
- **The 7 Services will still have the same application interface but the database that they would be accessing would be the master database that is generated by joining the 7 databases.**
- **Any change created in the master database by one application can be seen having a ripple effect over every database concerning to the specific customer.**
- **The customers can have a ticket number to forward all their description of issues to be out in over there to forward the issues and send the information to the next department for further details.**
- **If the customers just want to provide any name change then that change in one application can be seen for all the connected applications that they use since the system now has a connected master table. The Master Table can be further taken up by ETL tools to put all the data into a data warehouse and use BI tools to help create dashboards, Reports. This can also help the firm in creating a single bill that contains details from all the different applications used.**

  Since, all the applications are connected to a single master database, the employees from any department can view all the customers of Integrated Trading.

Solution 2:

Solution Space

Procedure:

- Customer calls the Customer Care for Service.
- The Customer Care representative asks for Personal Details.
- The Customer Provides the credentials/details and issue/problems or changes in details/personal information.
- The Customer receives the information related to the concerned service, Ticket Number. If the call is for change in name/something similar then they do not need to contact any other department again for the same. If they have any other specific questions related to another department, then the customer would ask for call transfer to another department.
- The customer requests for another Service
- Call is forwarded to another Service Department
- Customer Representative asks customer for Ticket Number, problems
- The customer provides details
- The customer representative provides information, support and solution.

Continued……….

Preferred Solution:

I would prefer Solution 2 Since it's a very flexible and scalable application in all. If at all the company plans to have a few more acquisitions soon, then this system will be able to integrate those services as new modules.

Advantages:

- This is the fastest and cheapest solution.
- Changes made through 1 System for one customer can be seen across each and every database since all changes are made to the main database and this is reflected in every application. Hence, the customer doesn't need to call the customer care multiple times for changes. The triggers make changes to all related databases
- The bills and reports for all the services used by the customer can be generated in one bill.
- Customers related to all departments can be seen through any application since all the applications use the master database.
- The system in modular currently, any new application that needs to be added just needs to be made compatible to the master database and any module can be added or removed from the system.
- The system supports Data Warehouse and Business Intelligence tools for reporting, dashboarding.
- The data keeps getting mapped from master database to applications database using Triggers and API calls