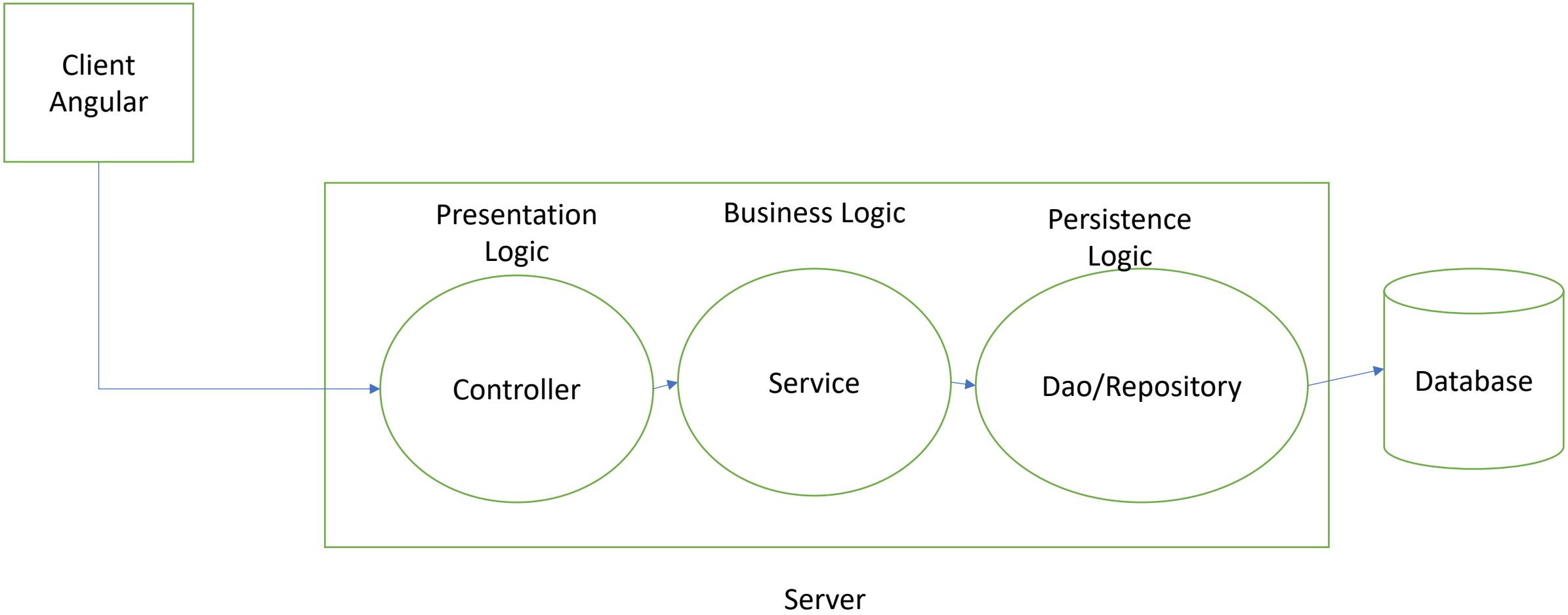
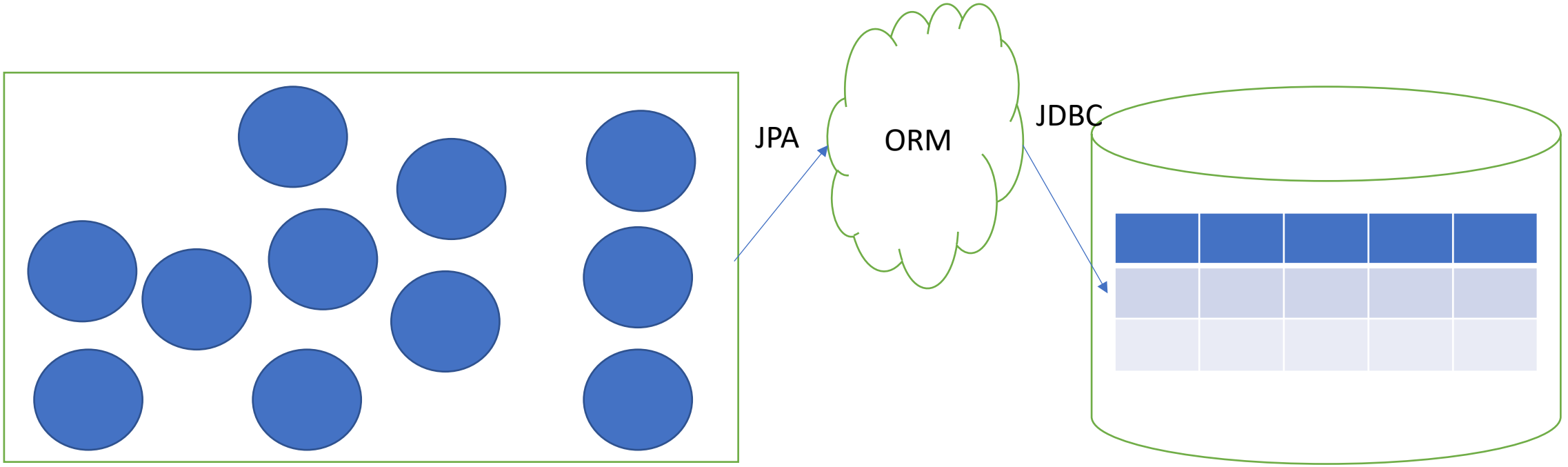
Roughwork

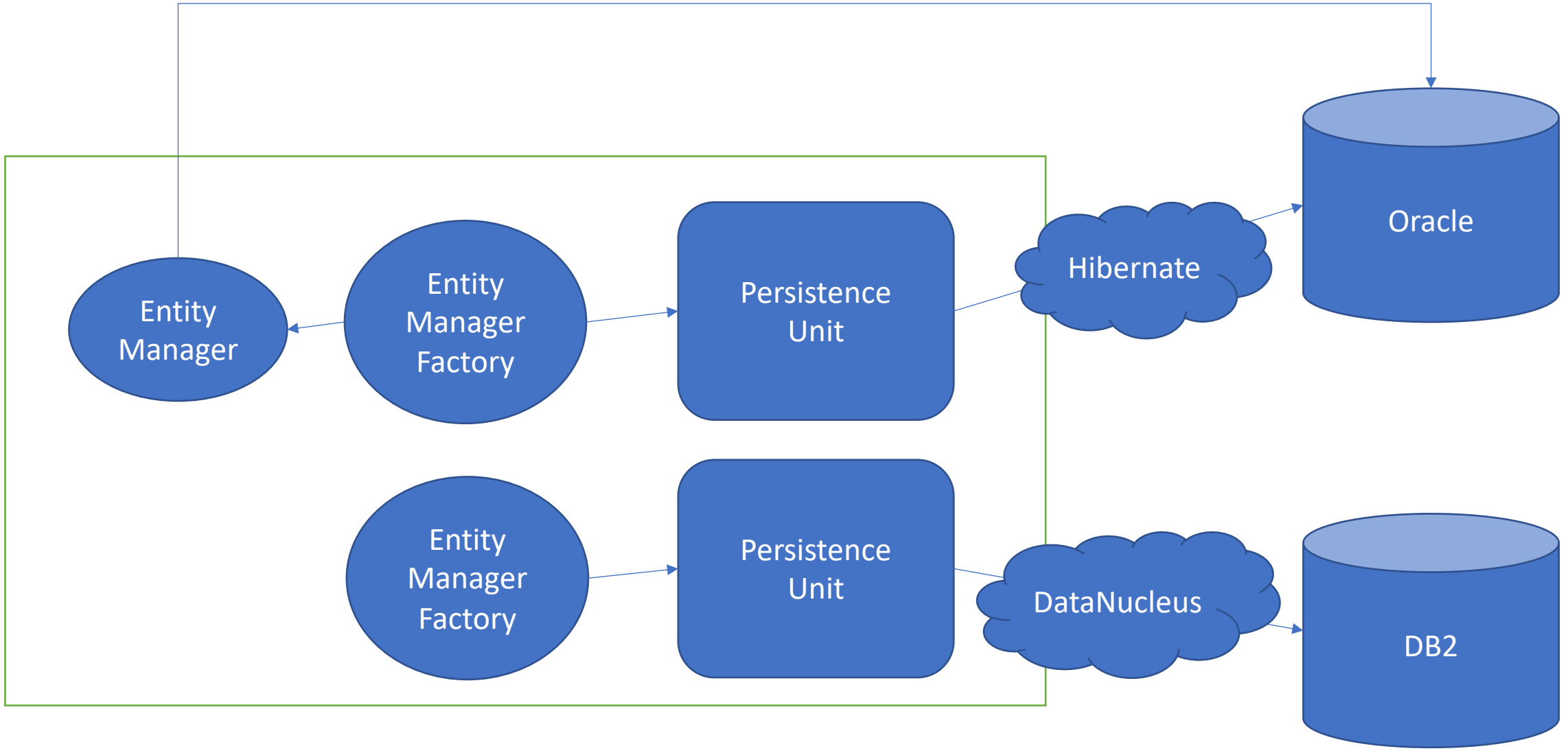
Common layers in an enterprise application



In an OO application, data is represented in the form of Objects
Whereas in a Database, data is represented in the form of rows and columns (tuple)



ORM bridges the gap between the OO and RDBMS by providing a convenient API for interaction



EntityManager and EntityManagerFactory object

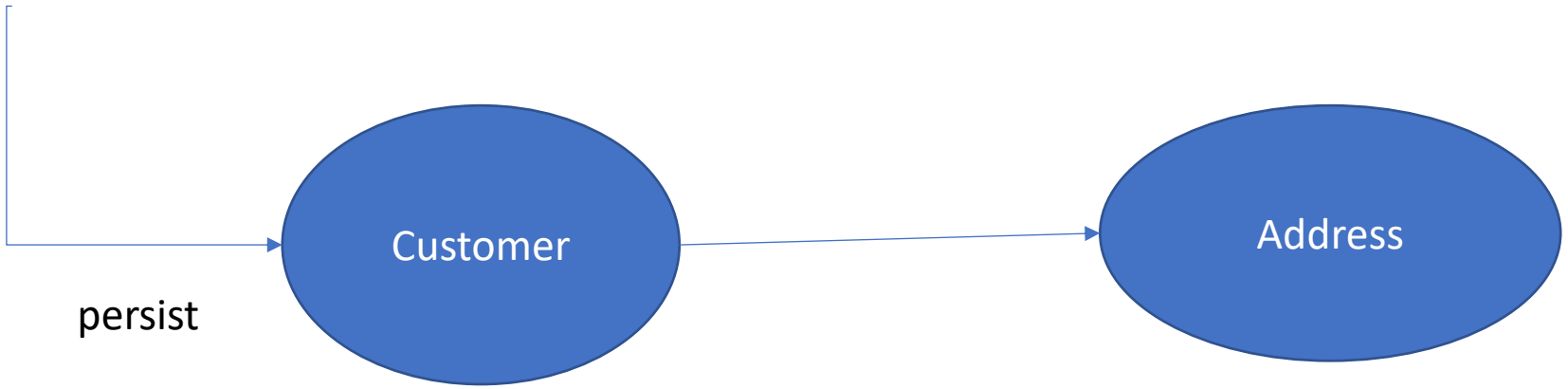
One to One Association

Cust_id (pk)	Name	Email	Addr_id (fk)
111	Majrul	…	100

Associations:
One to one
One to many / many to one
Many to many

Addr_id (pk)	City	Pincode	State
100			
200			
300			

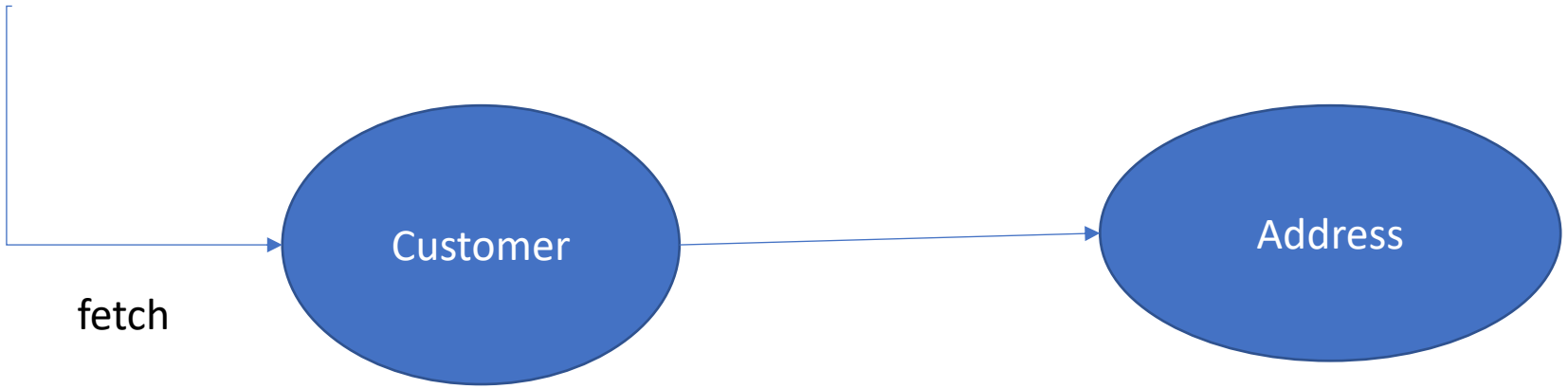
Cascading



In the above example, we will need to use cascade if we want the address to get stored automatically when we store the customer in the database

Whatever action we perform on one end of the association, should that be applied to the other end is decided using cascade property in Hibernate/JPA

Fetching



If we are fetching one end of the association, should Hibernate fetch the other end or not is decided using lazy/eager loading

For one ended association (one to one, many to one), default is eager loading
For many ended association(one to many, many to many), default is lazy loading

Album_id (pk)	Name	Release_date	Copyright
100	Hits of 2021	10-Dec-21	Sony
200	Hits of 2020	15-Dec-20	Amazon

One to Many association

Many to One association

Song_id (pk)	Title	Artist	Duration	Album_id (fk)
111	BAD HABITS	ED SHEERAN	4.00	100
222	GOOD 4 U	OLIVIA RODRIGO	3.30	100
333	DRIVERS LICENSE	OLIVIA RODRIGO	4.20	100
444	Memories	Maroon 5	4.00	200
555	The Box	Roddy Ricch	2.45	200

Possible tables from a project point of view

Album_id (pk)	Name	Release_ date	Copyright
100	Hits of 2021	10-Dec-21	Sony
200	Hits of 2020	15-Dec-20	Amazon

Artist_id	Name	Gender		
1				
2				
3				

Song_id (pk)	Title	Duration	Album_id (fk)

Song_id	Artist_id
111	1
111	2
222	1

Cust_id	Name	Email	Date of Registration

Some random example

Prod_id	Name	Price	Quantity

Order_id	Total	Order_Date	Prod_Id	Cust_Id

LineItem_id	Qty	Product_id	Order_Id	

One to Many association

Account

Account_id	Name	Type	Balance
12345	John	Savings	5000
98765	Smith	Savings	10000

One more example on one to many association

TransactionDetail

Many to One association

Transaction_Id	Type	Date_And_Time	Amount	Account_id
111	Withdraw	10-Jun	1000	12345
222	Withdraw	11-Jun	5000	98765
333	Deposit	10-Jun	5000	12345
444	Withdraw	12-Jun	2000	98765
555	Deposit	11-Jun	1000	12345

Cust_id	Name	Email	City
1	John		
2	Smith		
3	Jack		

Subs_id	Type	Amount	Duration
1	Movie Rental	1500	12
2	Book Rental	500	12
3			

Cust_id	Subs_id
1	1
1	2
2	2
2	1

Many to Many Association

user_id	Name	Email	City
1	John		
2	Smith		
3	Jack		

Subs_id	Type	Amount	Duration
1	Movie Rental	1500	12
2	Book Rental	500	12
3			

user_id (pk+fk)	Subs_id (pk+fk)	subs_date
1	1	10-May-2021
1	2	15-June-2022
2	2	
2	1	

Many to many can be visualized as one to many style association as well

True many to many association doesn’t allows additional columns in the the link table

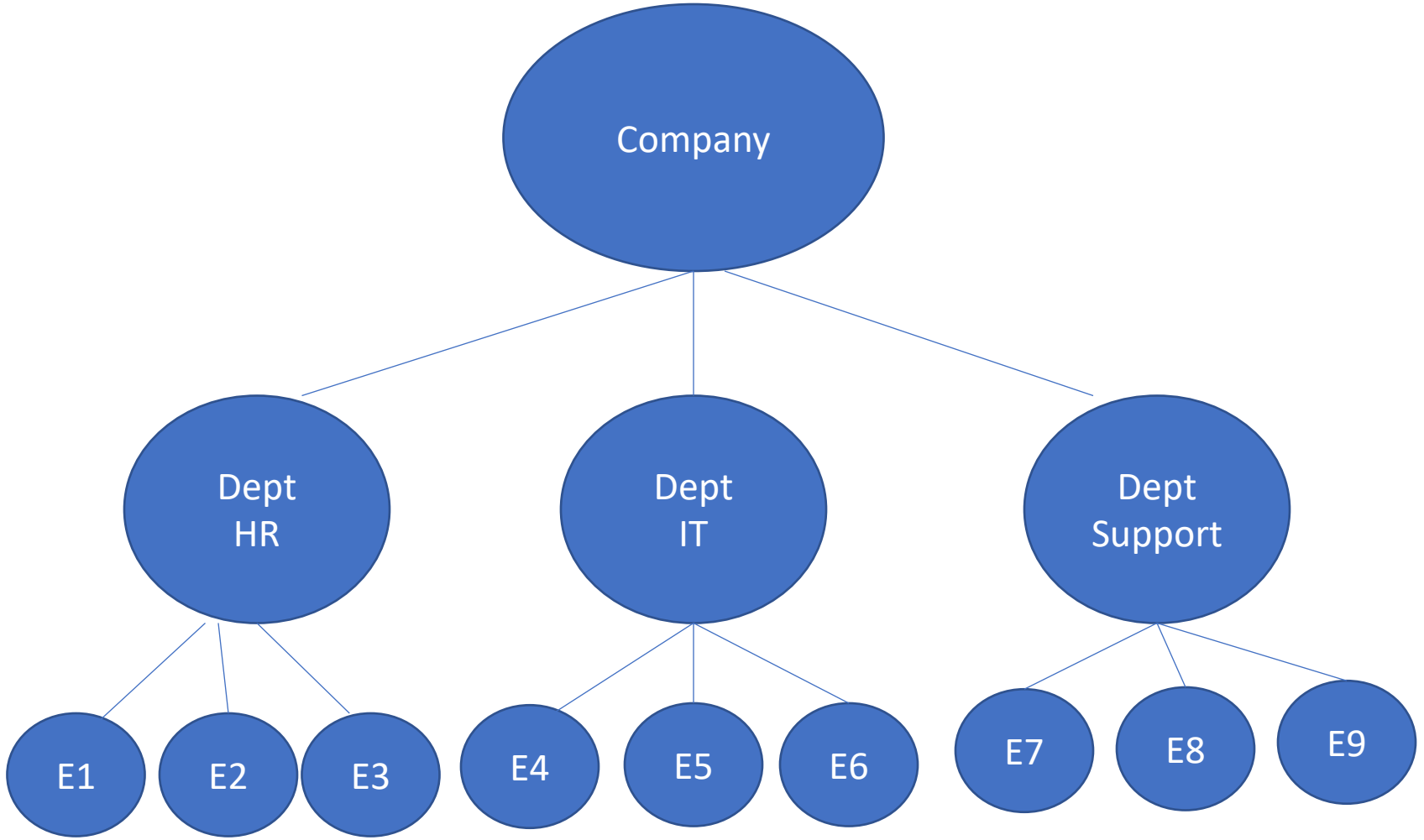
Embeddable example

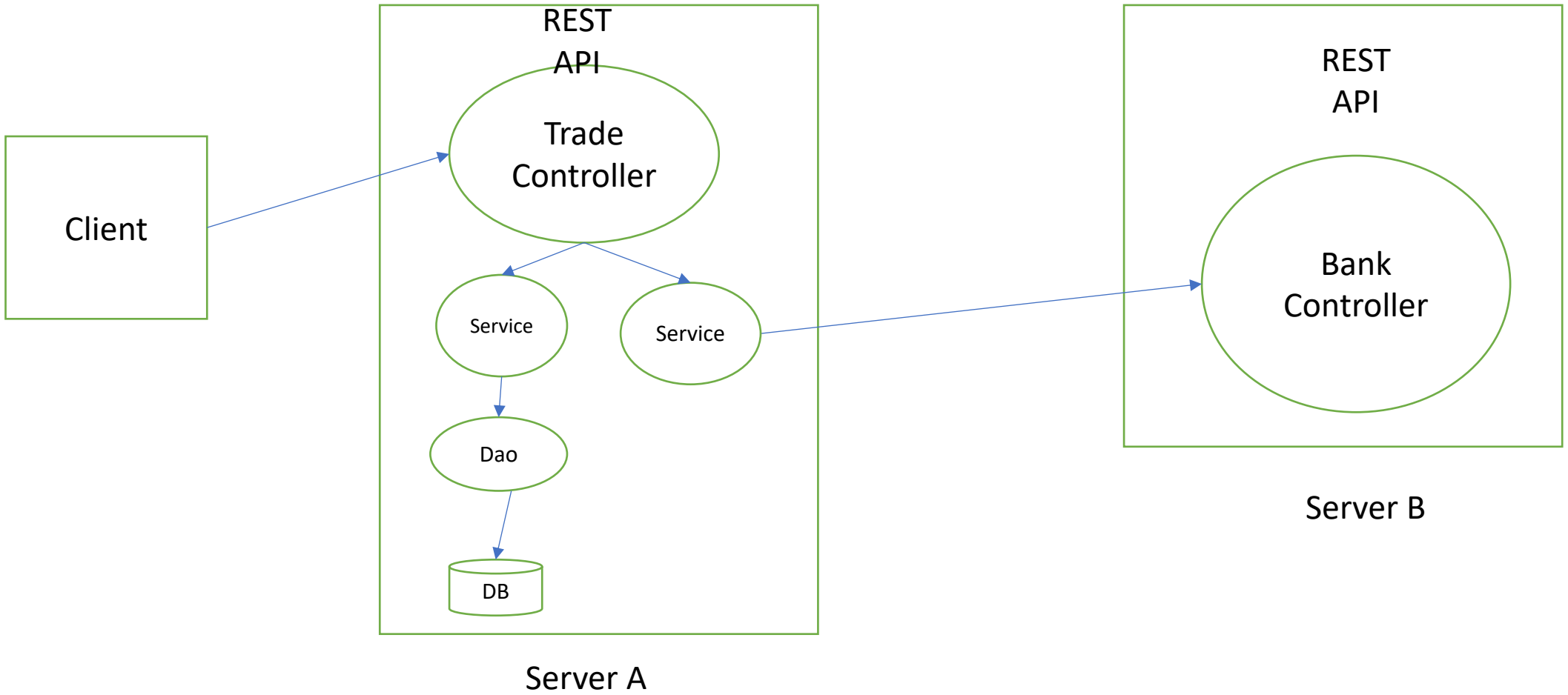
Cust_id	Name	Email	Dob	City	Pincode	State
1	John	…	..	Mumbai	400001	MH

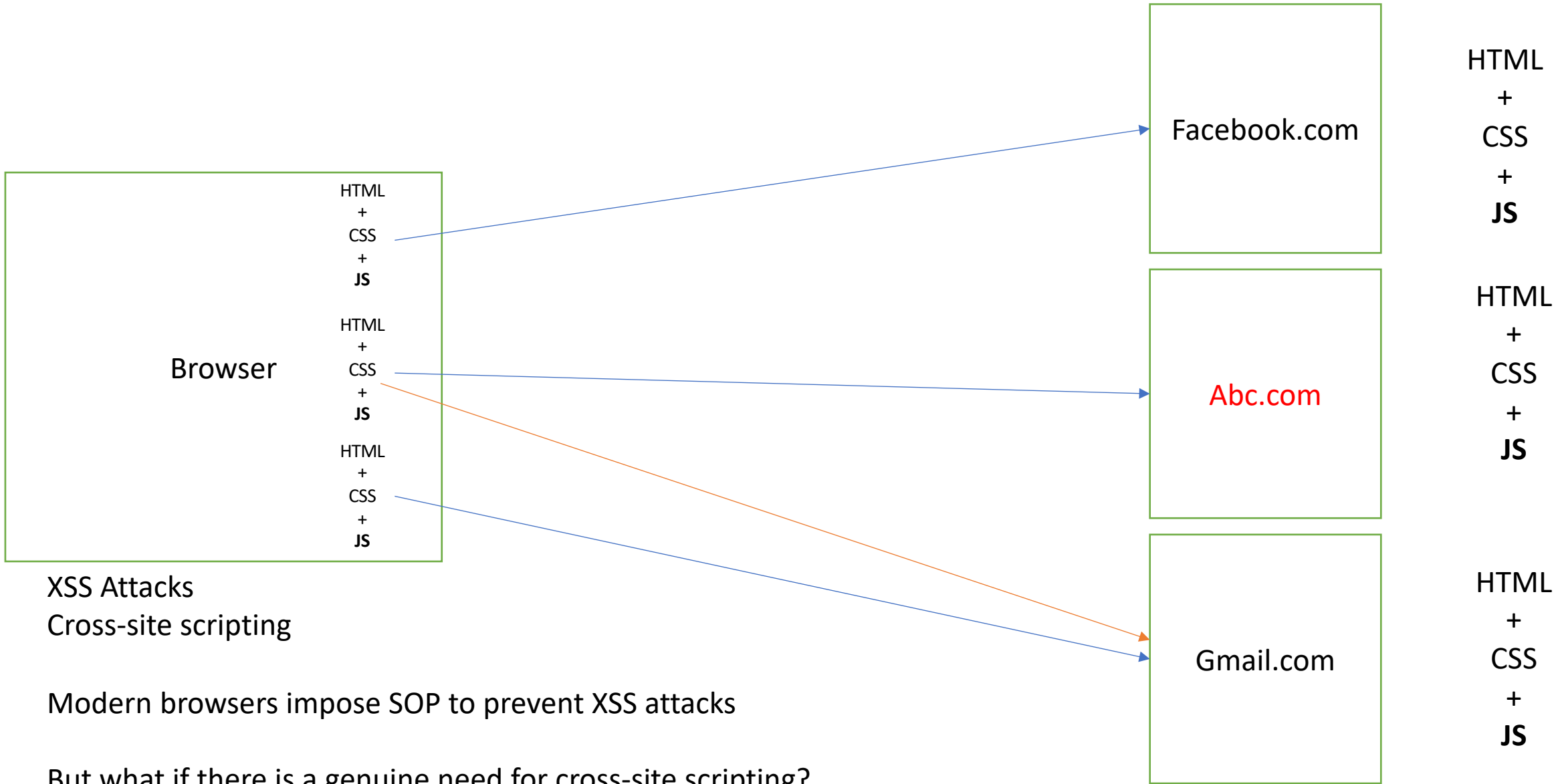
The above table contains two parts:

1. Customer Details
2. Address Details

While designing the Entity, we may not write a single class, rather we will use embeddable annotation


Company
Dept HR
Dept IT
Dept Support
E1
E2
E3
E4
E5
E6
E7
E8
E9

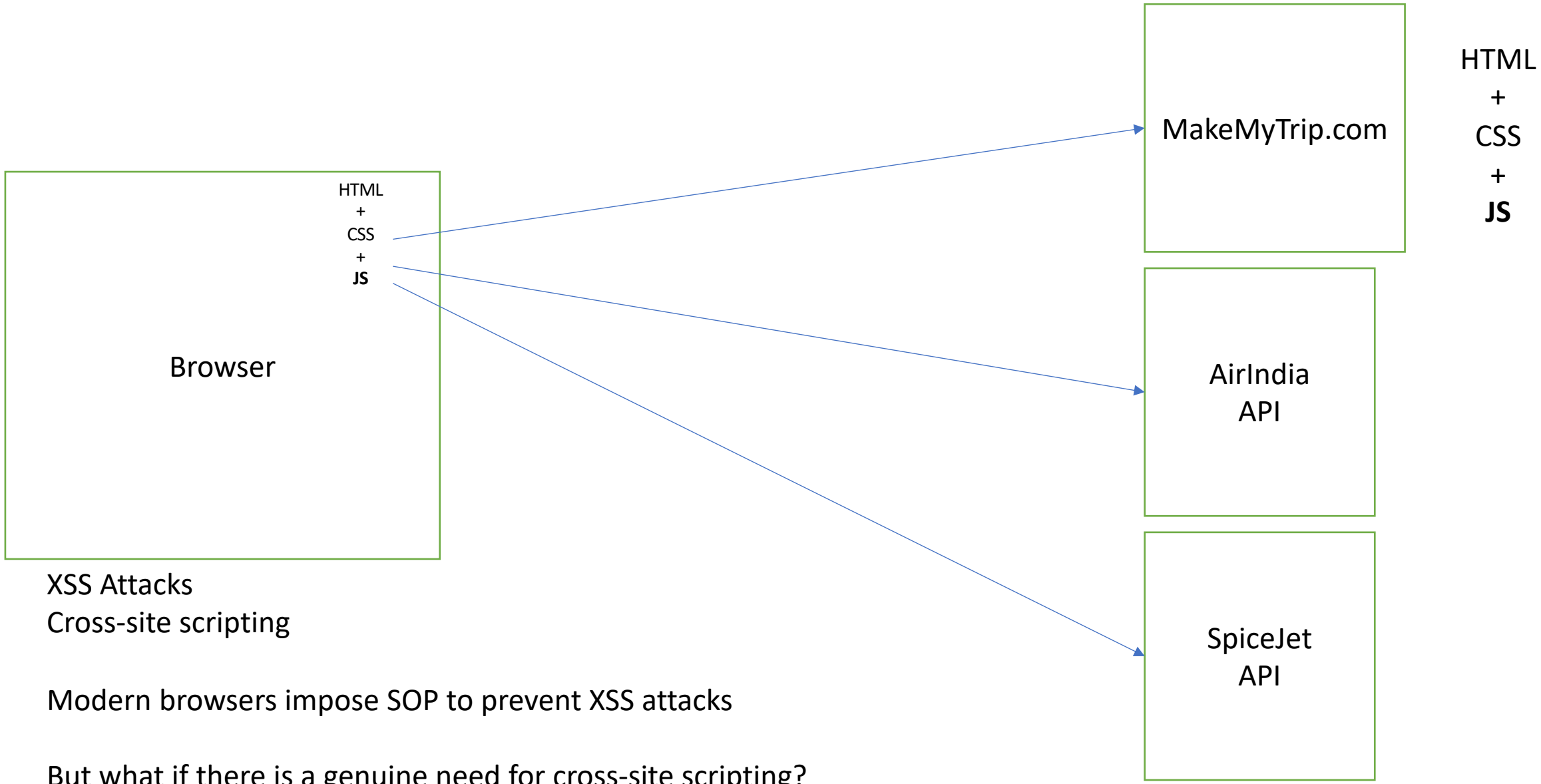

Client
REST
API
Trade
Controller
Service
Service
Dao
DB
Server A
REST
API
Bank
Controller
Server B



XSS Attacks
Cross-site scripting

Modern browsers impose SOP to prevent XSS attacks

But what if there is a genuine need for cross-site scripting?
Solution is CORS



XSS Attacks
Cross-site scripting

Modern browsers impose SOP to prevent XSS attacks

But what if there is a genuine need for cross-site scripting?
Solution is CORS