**Can you explain the project architecture?**

Let me give a brief about Paypal and my project.

Paypal does business on online payment system. User can register their information; add their credit card, debit card, bank accounts for payment on any merchant site. Some of the popular merchant site offering paypal payments are ebay, Walmart, macys, microsoft

There are different project modules associated like Authentication, Authorization, Risks, Security, Fraud Management, Payment/Checkout, User Account Management, Merchant Account Management, Financial Instrument Management.

I was working in project FIMS i.e Financial Instrument Management System. This is a backend service built on JAX-RS web service. Through this service, registered user can perform CRUD operations on Credit Card, Debit Card, Gift Card, Bank Account and other various payment instruments.

Front-End for our service is User Account Management.

The initial project is built on legacy c++ code base.

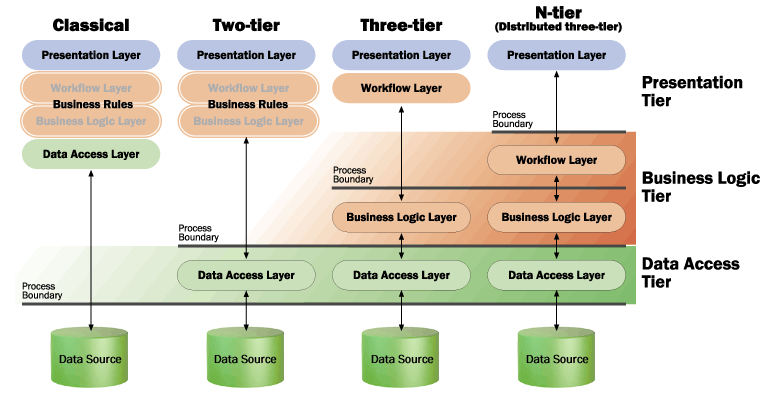
We wrote Java code i.e JAX-RS services for all those codes and new clients are directly pointing to the Java-services.

We also started migrating the existing C++ traffic to Java slowly using Throttling logic.

Coming back to architecture…

The project follows n-tier architecture. Presentation Tier on 1 system, Business Logic on 1 system, Database on 1 system, Messaging Communication on 1 system, Different Third party applications, Dependency services all deployed in different system.

Components vs services.



The server backend is built as REST services using Spring-Webmvc and Rest-Easy.

Once the request is received in Application Container there are multiple filters configured for Fraud, Security, Decryption, Audit-Logging etc…

Then web.xml maps the request to DispatcherServlet.

Then request is mapped to appropriate controller.

We have a JSON Schema validator as a first step in controller, which will validate the required fields and data type.

Then JSON is mapped into appropriate POJO using “Jackson-databind” which is based on fasterxml.

Once the validation is success, the controller will verify “Authentication & Authorization” information of the user through OAUTH client service. OAuth confirms whether user has the scope to access particular service. Ex: Admin can view/update the account details of user but cannot add a card by himself. Otherwise could have his card and can transfer money from user bank account to admin account.

The different layers of our project are Controllers🡪Transformers🡪Façade🡪Service🡪DAO or Integration Layer

JSON🡪Controller🡪DTO

DTO🡪Transformers🡪BO

Then the request is handled to the facade layer. The transaction will begin on facade Layer. The façade layer calls 1 or more service to complete the operation. The service layer will call Repository layer for CRUD operation or Integration Layer for calling remote service. The exception will be propagated back to the service layer. The service layer converts the exception to appropriate business exception.

The Logging is done through distributed asynchronous call.