**Account Transfer API Services:**

**Description** :

Has simple REST service with some very basic functionality - to add and read an account and funds transfer between accounts.

**New Task/Service:**

add functionality for a transfer of money between accounts. Transfers should be specified by providing:

* accountFrom id
* accountTo id
* amount to transfer between accounts

**Implementation** :

Multiple concurrent read , insert & update(transfer) operation should be possible.

**Future Possible Enhancement:**

To make application production ready following features can be added.

It provides Fault Tolerant + Highly Secure + Elastic Load Balancing … application

**API Documentation:**

Spring Rest API with Swagger – Creating documentation

1)This applications rest endpoint/controller can be enhanced using swaggers **API End Point Documentation**

using annotation : @Api, @ApiOperation, @ApiParam , @ApiResponse

2)Can have **Model Documentation**:

Application domain model & request objects & response objects can enhanced

Using : @ApiModel, @ApiModelProperty ...

-------------------------------------------------------------------------------------------------------------------------------

**Logback Logging : Splunk logging**

Application logging can be enhanced using logback for

**ConsoleAppender**: console logging

**Splunk Appender** logging : by sending logging event to Splunk event collector endpoint & can be used for Log analytics & Reporting.

& Real-Time Enterprise Log Management to Search, Diagnose and Report

**Log Data Masking:** Mask sensitive data in logs with logback

-------------------------------------------------------------------------------------------------------------------------------

**Spring Cloud /Netflix cloud support:**

These service can be put on cloud & integrated with Discovery services like Eureka

using : @EnableEurekaServer, @EnableEurekaClient

**Centralized Configuration:**

A config server can be setup and this account service can consumes the configuration on startup and

then refreshes the configuration without restarting the client.

**Security :**

OAuth security can be implemented for Rest Api services.

**Include CorelationId:**

Implementing Correlation ids in this application for Distributed Tracing

Example : using application request filters

<Filter>

|

CorrelationHeaderFilter {

1. get CorelationId header attribute from Http request header
2. if not present , create a new one & set in request header

}

**Fraudnet/Fraud Analytics integration:**

Users machine details & user profile can be captured in important calls and send info to Fraudnet/Fraud Analytics

systems , so that the descision can be used in important payments Apis.

**Storing Api State in Db:**

for each microservices Api the state information can be stored in NoSQL DB ring cluster to provide

fail safe auto scaling of APIs.

**Load Balancer (with Proxy Services)**

Integration with Zuul procxy filter for having JVM-based router and server-side load balancer

to have Facade level Authentication, Static Response handling, traffic management, Monitoring

**Deployment on Cloud Infra :**

1) Deploy the API on to AWS cloud infra possibly in **VPC** in **SubNet** with **SecurityGroup** configured + Network ACL policy

configured

2) Protection with **WAF layer** : protecting API Service with Web Access FireWall will protect Api from

various attcks like distributed denial of services, cross site scripting,

**Integration with LDAP/Active Directory :**

For End user Authentication & Authorization or SSO.

**Security Layers & Infra components Diagram:**

Financial Services

OAUTH2

API Gateway /Proxy Services

AWS WAF

Network Firewall

Account Transfer API

Audit/logging

DB

Fraud Detection

LDAP,Active Dir

Splunk Analytics