**Tech Architecture, Components & Features**

**Project Name :** Maya Exchange

**Architecture:**

1. Frontend: User interface (Web, Mobile)
2. Backend: API, Services (Microservices architecture)
3. Database: Relational databases (e.g., MySQL) for storing user data, transactions, and order books
4. Cryptocurrency Integration: APIs for cryptocurrency exchanges, wallets, and payment processors
5. Banking Integration: APIs for banking services, UPI, and payment gateways
6. Security: OpenSSL, SSL/TLS, 2FA, JWS, Oauth Token, Encryption, access controls

**Components:**

**1. User Management :**

* KYC/AML checks
* User profiles
* Account management

**2. Crypto Management :**

* Wallet Integration
* Exchange Integration
* Trading engine

**3. Banking Management :**

* Upi & IMPS & Individual Bank interface integration
* Payment gateway integration
* Fiat currency management

**4. Transaction Management :**

* Order books
* Trade matching
* Settlement

**5. Cross-Border Transactions Management :**

* International payment processing
* Currency conversion
* Compliance with regulations

**Local Currency to Crypto Feature (Same Scenario With Local Currency to crypto) :**

**1. User Interface :**

* + Select recipient (User 2)
  + Choose local currency (INR)
  + Select desired cryptocurrency or local currency

**2. Backend Processing :**

* + Verify user balance and KYC/AML status
  + Convert local currency to cryptocurrency (or other local currency)
  + Execute trade on cryptocurrency exchange or OTC desk
  + Update user balances and transaction history

**Cloud Services Integration :**

**AWS:**

* + Amazon EC2 (scalable infrastructure)
  + Amazon RDS (database management)
  + Amazons3 (storage)
  + Amazon CloudWatch (monitoring)

**Advanced Features**

1. **Artificial Intelligence (AI)**: Implement AI-powered trading bots and market analysis tools
2. **Machine Learning (ML)**: Integrate ML algorithms for predictive modeling and risk management
3. **Blockchain Analytics**: Utilize blockchain analytics tools for transaction tracking and security
4. **Compliance**: Implement automated compliance checks for regulatory requirements

**Deployment Strategy**

1. **Containerization**: Use Podman for containerization
2. **Orchestration**: Utilize Kubernetees for container orchestration
3. **Continuous Integration/Continuous Deployment (CI/CD)**: Implement CI/CD pipelines using GitLab CI/CD

**Security Measures**

1. **Encryption**: Implement end-to-end encryption for data protection
2. **Access Controls**: Enforce strict access controls and role-based permissions
3. **Two-Factor Authentication**: Require 2FA for all user accounts
4. **Regular Security Audits**: Conduct regular security audits and penetration testing

**Scalability**

1. **Horizontal Scaling**: Scale infrastructure horizontally using cloud services
2. **Load Balancing**: Implement load balancing for efficient traffic distribution
3. **Caching**: Utilize caching mechanisms for improved performance

**Database Combination**

**Traditional Databases**

1. **MongoDB**: Store customer profiles, order history, and market data.
2. **MySQL**: Manage transactional workloads, user accounts, and banking information.
3. **PostgreSQL**: Store complex relational data, trade history, and order books.

**AWS Databases**

1. **Amazon RDS (MySQL/PostgreSQL)**: Relational database for ACID-compliant transactions.
2. **Amazon DynamoDB**: High-performance key-value store for real-time market data.
3. **Amazon Redshift**: Data warehousing and analytics for sales data and customer behavior.

**Transaction Database**

1. **Apache Cassandra**: Distributed, scalable database for transaction history and trade matching engine.

**Integration Strategies**

1. **Data Replication**: Amazon Data Migration Service (DMS) for replicating data between traditional and AWS databases.
2. **API Gateway**: AWS API Gateway for creating a unified API layer accessing data from multiple databases.
3. **Serverless Functions**: AWS Lambda for interacting with databases and performing tasks like data processing and notifications.

**Data Flow**

1. User data → MySQL
2. Customer profiles → MongoDB
3. Market data → MongoDB and Amazon DynamoDB
4. Transaction data → Apache Cassandra
5. Trade history → PostgreSQL
6. Analytics data → Amazon Redshift

**Key Considerations**

1. **Data Consistency**: Ensure data consistency across databases using replication and validation mechanisms.
2. **Performance**: Optimize performance through indexing, query optimization, and caching.
3. **Cost**: Monitor costs associated with each database and AWS service.
4. **Management**: Establish clear management procedures for backups, security, scaling, and updates.

**Benefits**

1. **Flexibility**: Leverage strengths of each database for specific use cases.
2. **Scalability**: Handle high traffic and large data volumes.
3. **Performance**: Optimize query performance and data processing.
4. **Cost-Effective**: Reduce costs by choosing the right database for each use case.

**Components**

1. User Management
2. Crypto Management
3. Banking Management
4. Transaction Management
5. Cross-Border Transactions Management
6. Analytics and Reporting

**Features**

1. Local Currency to Crypto
2. P2P (Crypto Currency)
3. Crypto to Local Currency
4. P2P ( Local Currency)
5. P2P (Local Currency To Other International Country Currency)
6. P2P (Other International Country Currency To Local Currency)
7. Real-time Market Data
8. Advanced Analytics and Reporting
9. Secure Authentication and Authorization

**Transaction Database (Apache Cassandra) Schema**

1. transactions table: transaction\_id, user\_id, currency, amount, timestamp
2. order\_books table: order\_id, user\_id, currency, amount, price
3. trade\_history table: trade\_id, user\_id, currency, amount, price, timestamp

**Tech Stack:**

1. **Frontend:** HTML, CSS, React.tsx, tsx, java, kotlin
2. **Backend & Security and encryption:** C#, Perl, Ruby, Python
3. **Encryption algorithms:** CRYSTALS-Kyber, FALCON, Bcrypt
4. **Hashing Algorithms:** SHA-256, Skein, Grøstl, Whirlpool, Streebog
5. **Database:** MongoDB, MySQL, PostgreSQL, Amazon RDS, Amazon DynamoDB, Amazon Redshift, Apache Cassandra
6. **Integration:** Amazon DMS, AWS RDBS, AWS API Gateway, AWS Lambda
7. **Cryptocurrency Integration:** APIs (cryptoapi.io, tradinview), Web3.js, own liquidity pool
8. **Banking Integration APIs & SDKs:** debit, credit, rupay, visa, master card, amex, UPI SDKs
9. **Security**: OpenSSL, SSL/TLS, 2FA, JWS, Oauth Token, Encryption, access controls
10. **AWS:** Amazon EC2, Amazon RDS, Amazon S3, Amazon CloudWatch
11. **Orchestration & Containerization & Deployment:** Podman& Kubernetes
12. **(CI/CD):** GitLab CI/CD & Jenkins