**WHAT**

**System Architecture:**

* Microservice Architecture

Microservice gives flexibility for complexity of internal processes, and has many benefits such as stability improvements, fault isolation, different programming language, simple to deploy, reusable across different area of business, ability to experiment, improved data security and team optimization.

Besides price engine system, the service that must be present can be supplier management, purchase order management, payment management and goods handover management service.

* Database

PostgreSQL can be used for storing transactional data, but for handling unstructured and scalability, NoSQL database such as MongoDB is more suitable. Even we can use bigdata such as Cassandra and Hbase.

* Authentication and Authorization security

JWT based authentication service with role-based access control can be used for authorization. For enhanced input validation, signature for each request can be used.

* External Integrations

APIs for integrations with supplier system and buyer systems or even competitor market price.

* Machine Learning and AI

AI is the futures

**Infrastructure:**

* Cloud infrastructure

Utilize cloud platform like AWS for scalability and flexibility. Cloud infrastructure has advance scalability such as autoscaling system to handle varying workload demands and it has load balancing configurations also.

* Containerization

Docker containers can be used for packaging microservices and Kubernetes for container orchestration and management.

* Repository and Deployment

For developer team, repository such as GitLab can be used for project development. Deployment automation can be achieved by CI/CD tools

* Ansible Configuration

Ansible can be used for automate the management of remote systems and controls, emilite repetition and simplify workflows, manage and maintain system configuration, and perform zero-downtime rolling updates.

* Monitoring and Logging

Implementing monitoring tools like Prometheus or datadog for tracking system performance, new relic for application monitoring. Centralize logging using ELK stack for troubleshooting and analysis.

* Security Measures

Implement network security groups, firewall, and encryption to protect data. Regular security audits and vulnerability assessments also implement ISO certifications for IT governance.

**WHO**

**Internal Team:**

* Lead Engineer

Experienced engineer that responsible for overall technical direction and architecture design

* Backend Developers

Skilled developers for implementing microservice and database management

* Frontend Developers

Skilled developer for user interface application such as website and mobile applications for customer acquisition and retention

* Site Reliability Engineer

Responsible for infrastructure setup, deployment automation, and monitoring.

* QA Engineer

Ensure software quality through testing and validation

* Security Specialist

Focuses on implementing security measures and conducting audits.

**Vendor and Freelancer:**

* Vendor Support

On-demand access to specialized skills or additional resources as needed

* Freelancers

Contract-based support for specific tasks or projects, such as UI/UX design or niche development skills.

**HOW**

**Recruitment Process:**

* Target Hiring

Focus on hiring top-tier talent with expertise in relevant domains

* Networking and Referrals

Leverage professional networks and industry connections for recruitment

* Technical Assessments

Conduct rigorous technical interviews and coding assessments to evaluate candidates’ skill

* Competitive Compensation

Offer competitive salaries and benefits to attract top talent

**Engineering Process:**

* Agile Methodology

Adopt agile practices for iterative development and flexibility

* Continuous Integration/Continuous Deployment

Implement CI/CD pipeline for automated testing and deployment

* Coding Standard

Follow coding standard for any programming language that used.

* Code Reviews

Enforce code review processes to ensure code quality and knowledge sharing

* Regular Retrospectives

Hold regular retrospectives to identity areas for improvement and optimize team performance

* Cost Considerations

Balance the need for high-quality engineering practices with cost-effectiveness. Invest in tools and processes that provide long-term benefits and efficiency gains.