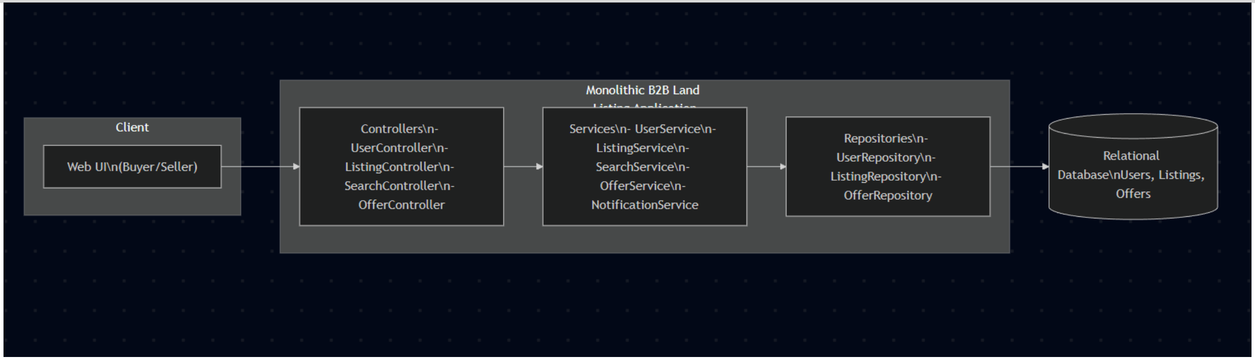
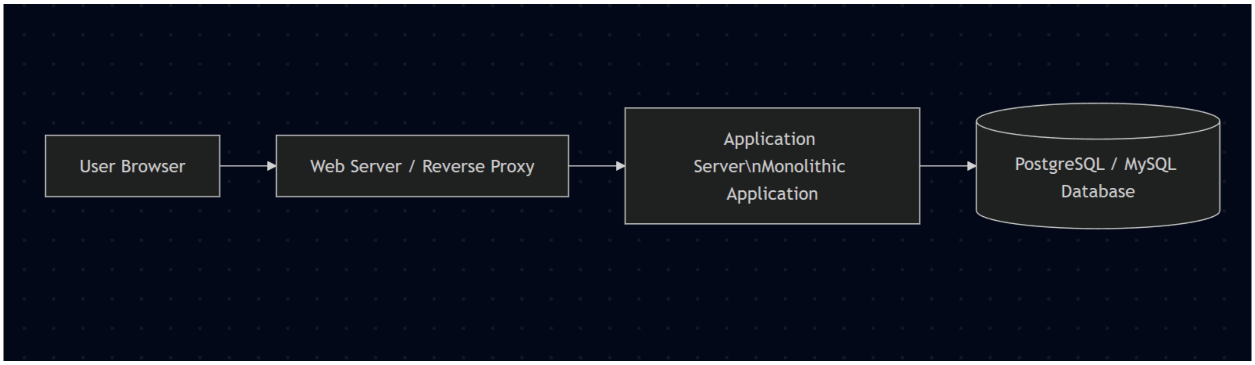
## 1) Monolithic – Component Diagram

```
flowchart LR
    subgraph Client
        UI[Web UI\n(Buyer/Seller)]
    end

    subgraph Monolith[Monolithic B2B Land Listing Application]
        Controller[Controllers\n- UserController\n- ListingController\n- SearchController\n- OfferController]
        Service[Services\n- UserService\n- ListingService\n- SearchService\n- OfferService\n- NotificationService]
        Repo[Repositories\n- UserRepository\n- ListingRepository\n- OfferRepository]
    end

    DB[(Relational Database\nUsers, Listings, Offers)]

    UI --> Controller
    Controller --> Service
    Service --> Repo
    Repo --> DB
```


Client
Web UI\n(Buyer/Seller)
Monolithic B2B Land
Controllers\n- UserController\n- ListingController\n- SearchController\n- OfferController
Services\n- UserService\n- ListingService\n- SearchService\n- OfferService\n- NotificationService
Repositories\n- UserRepository\n- ListingRepository\n- OfferRepository
Relational Database\nUsers, Listings, Offers

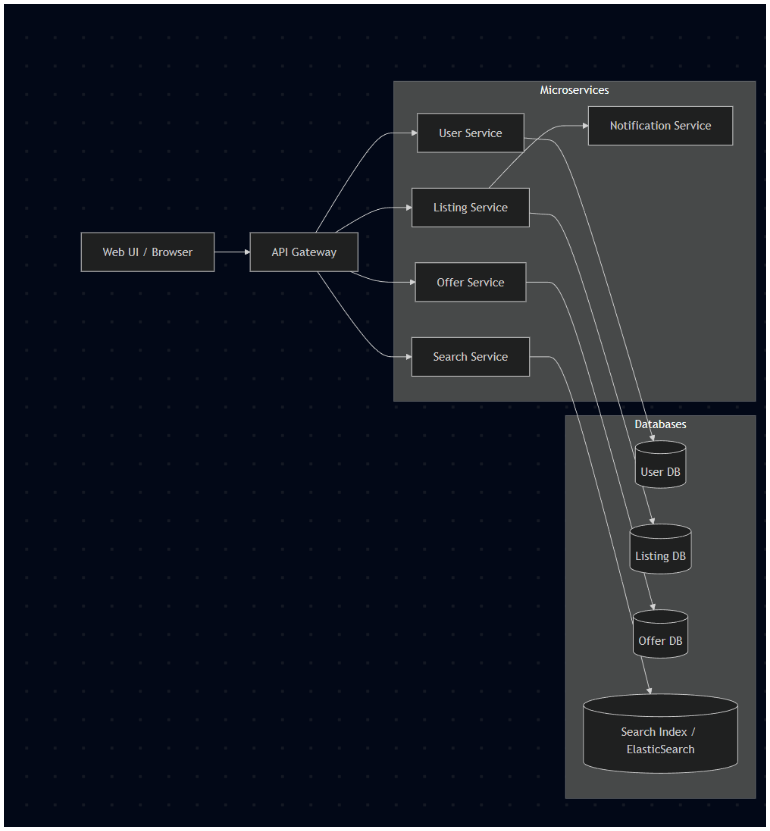
## 2) Monolithic – Deployment Diagram

```
flowchart LR
    Browser[User Browser] --> Nginx[Web Server / Reverse Proxy]

    Nginx --> AppServer[Application Server\nMonolithic Application]

    AppServer --> DB[(PostgreSQL / MySQL Database)]
```


User Browser
Web Server / Reverse Proxy
Application Server\nMonolithic Application
PostgreSQL / MySQL Database

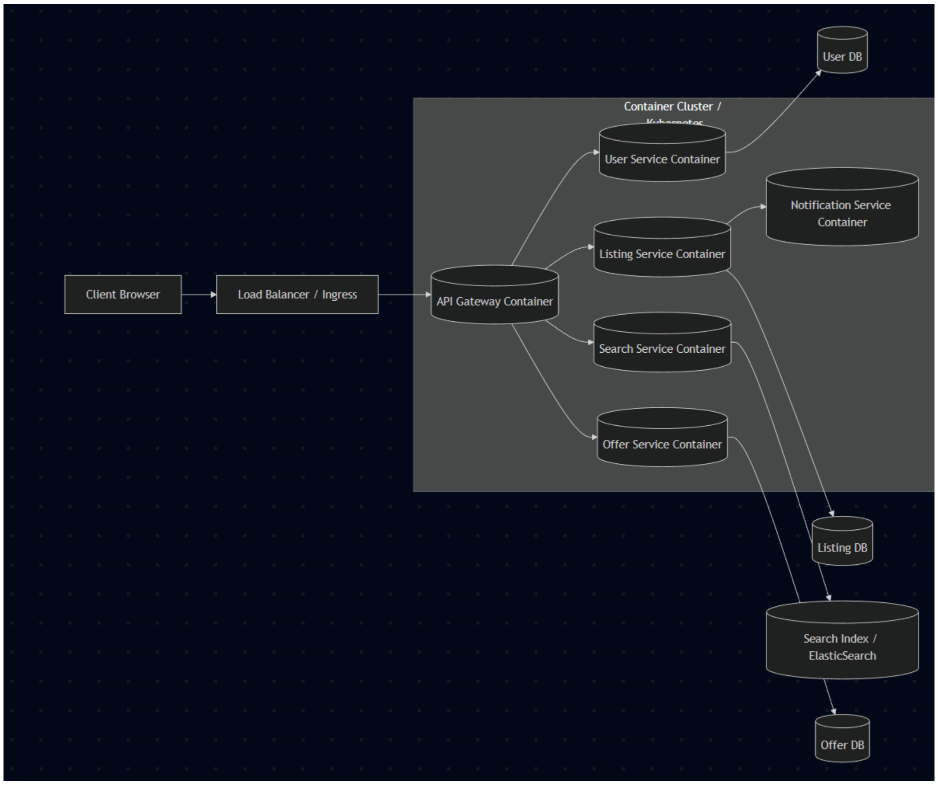
### 3) Microservices – Component Diagram

```
flowchart LR
    Client[Web UI / Browser] --> APIGW[API Gateway]

    subgraph Microservices
        UserSvc[User Service]
        ListingSvc[Listing Service]
        SearchSvc[Search Service]
        OfferSvc[Offer Service]
        NotifSvc[Notification Service]
    end

    APIGW --> UserSvc
    APIGW --> ListingSvc
    APIGW --> SearchSvc
    APIGW --> OfferSvc
    ListingSvc --> NotifSvc

    subgraph Databases
        UserDB[(User DB)]
        ListingDB[(Listing DB)]
        OfferDB[(Offer DB)]
        SearchIndex[(Search Index / ElasticSearch)]
    end

    UserSvc --> UserDB
    ListingSvc --> ListingDB
    OfferSvc --> OfferDB
    SearchSvc --> SearchIndex
```



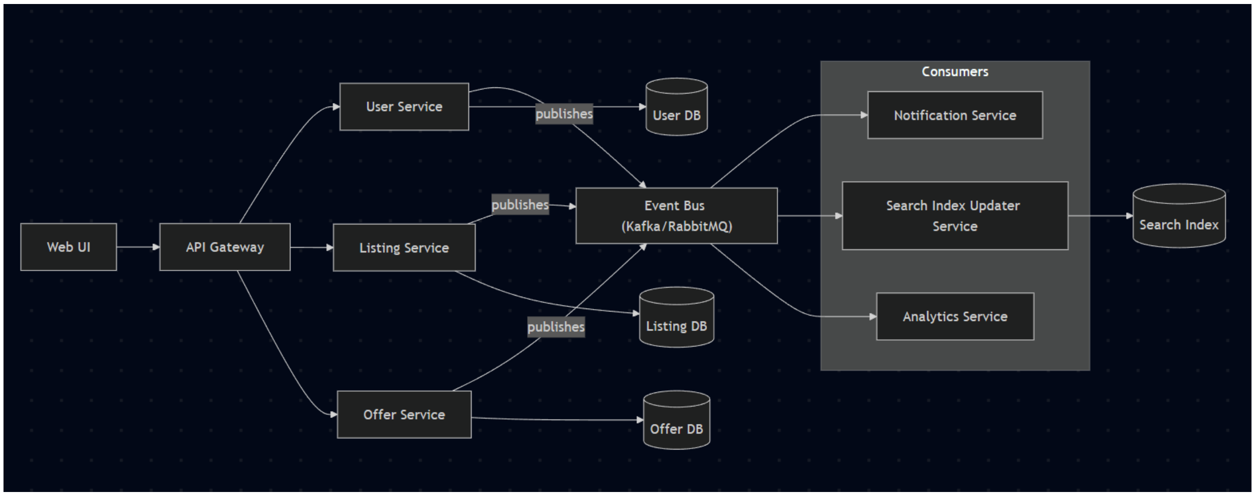
## 4) Microservices – Deployment Diagram

```
flowchart LR
    Browser[Client Browser] --> LB[Load Balancer / Ingress]

    LB --> APIGW[(API Gateway Container)]

    subgraph Cluster[Container Cluster / Kubernetes]
        APIGW
        UserPod[(User Service Container)]
        ListingPod[(Listing Service Container)]
        SearchPod[(Search Service Container)]
        OfferPod[(Offer Service Container)]
        NotifPod[(Notification Service Container)]
    end

    APIGW --> UserPod
    APIGW --> ListingPod
    APIGW --> SearchPod
    APIGW --> OfferPod

    ListingPod --> NotifPod

    UserPod --> UserDB[(User DB)]
    ListingPod --> ListingDB[(Listing DB)]
    OfferPod --> OfferDB[(Offer DB)]
    SearchPod --> SearchIndex[(Search Index / ElasticSearch)]
```



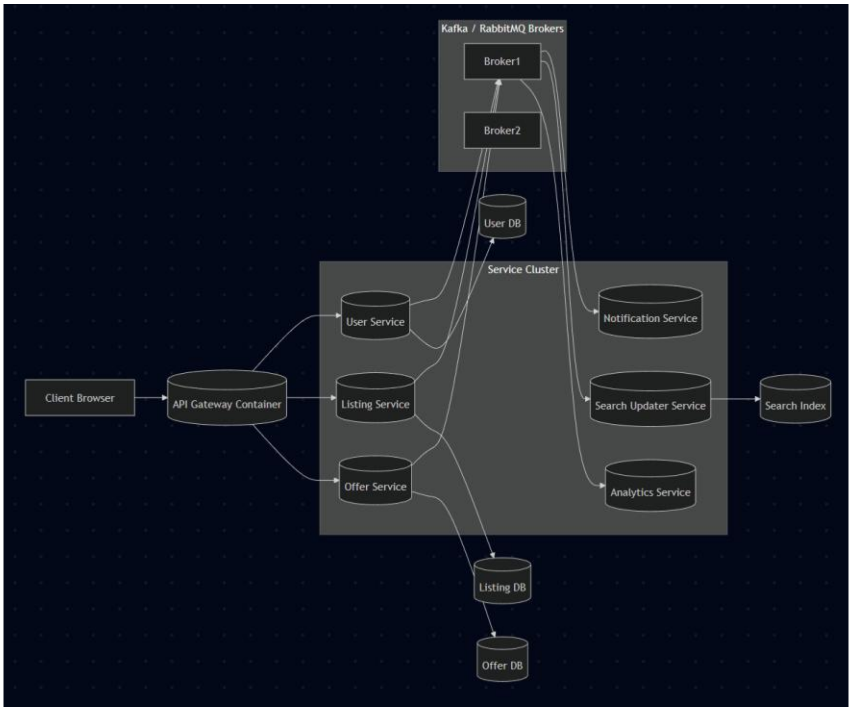
## 5) Event-Driven – Component Diagram

```
flowchart LR
    Client[Web UI] --> APIGW[API Gateway]

    APIGW --> UserSvc[User Service]
    APIGW --> ListingSvc[Listing Service]
    APIGW --> OfferSvc[Offer Service]

    subgraph Broker["Event Bus (Kafka/RabbitMQ)"]
    end

    ListingSvc -- publishes --> Broker
    OfferSvc -- publishes --> Broker
    UserSvc -- publishes --> Broker

    subgraph Consumers
        NotifSvc[Notification Service]
        SearchUpdater[Search Index Updater Service]
        AnalyticsSvc[Analytics Service]
    end

    Broker --> NotifSvc
    Broker --> SearchUpdater
    Broker --> AnalyticsSvc

    ListingSvc --> ListingDB[(Listing DB)]
    OfferSvc --> OfferDB[(Offer DB)]
    UserSvc --> UserDB[(User DB)]
    SearchUpdater --> SearchIndex[(Search Index)]
```



## 6) Event-Driven – Deployment Diagram

```
flowchart LR
    Browser[Client Browser] --> APIGW[(API Gateway Container)]

    subgraph Services[Service Cluster]
        UserSvc[(User Service)]
        ListingSvc[(Listing Service)]
        OfferSvc[(Offer Service)]
        NotifSvc[(Notification Service)]
        SearchUpd[(Search Updater Service)]
        AnalyticsSvc[(Analytics Service)]
    end

    APIGW --> UserSvc
    APIGW --> ListingSvc
    APIGW --> OfferSvc

    subgraph EventBus[Kafka / RabbitMQ Brokers]
        Broker1
        Broker2
    end

    ListingSvc --> Broker1
    OfferSvc --> Broker1
    UserSvc --> Broker1

    Broker1 --> NotifSvc
    Broker1 --> SearchUpd
    Broker1 --> AnalyticsSvc

    UserSvc --> UserDB[(User DB)]
    ListingSvc --> ListingDB[(Listing DB)]
    OfferSvc --> OfferDB[(Offer DB)]
    SearchUpd --> SearchIndex[(Search Index)]
```



https://mermaid.live