Real Estate Finder Platform - Network Architecture Documentation

Table of Contents

Assumptions

Traffic and Usage Patterns

1. **Traffic Distribution**: 60% traffic from US region, 40% from EU region
2. **Peak Hours**: Platform experiences 3x normal traffic during peak hours (6 PM - 10 PM local time)
3. **Search-to-Bid Ratio**: 100:1 (100 searches for every 1 bid placed)
4. **Chat Activity**: 30% of users engage in chat functionality
5. **Session Duration**: Average user session is 15 minutes
6. **API Calls**: Each user session generates approximately 50 API calls to 3rd party services

Data and Storage

7. **Database Size**: Initial database size of 500 GB, growing at 100 GB/month
8. **Image Storage**: Average property has 20 images (5 MB each), stored in S3
9. **Cache Hit Ratio**: 80% cache hit rate for frequently accessed property data
10. **Search Index**: OpenSearch index size approximately 30% of total property database

Availability and Performance

11. **SLA Requirements**: 99.95% uptime requirement
12. **Response Time**: <200ms for search queries, <500ms for page loads
13. **Failover Time**: Maximum 2 minutes for regional failover
14. **Data Replication**: Asynchronous replication between regions (5-minute lag acceptable)

Development and Security

15. **Developer Access**: 20-50 developers accessing the network via VPN

16. **Deployment Frequency**: 3-5 deployments per day using CI/CD pipeline
17. **Third-Party APIs**: Integration with 10+ external services (payment, mapping, verification)
18. **Compliance**: GDPR compliance required for EU region, data residency enforced
19. **DDoS Protection**: Expecting potential attacks up to 10 Gbps
20. **Encryption**: All data encrypted at rest and in transit (TLS 1.3)

Project Details

The Real Estate Finder Platform is a comprehensive web-based application designed to connect property buyers, sellers, and renters across multiple countries. The platform enables users to search for properties using advanced filters, participate in real-time bidding, communicate through integrated chat functionality, and access property data aggregated from various third-party services across different geographical locations.

The platform operates in a multi-region architecture spanning US-EAST-1 and EU-WEST-1 AWS regions, each with two availability zones to ensure high availability and disaster recovery capabilities. The system integrates with numerous external APIs to fetch property listings, pricing data, neighborhood information, and verification services from different countries and cities. Developers maintain continuous access to the infrastructure through secure VPN connections for ongoing development, testing, and deployment activities.

The architecture supports multiple user personas including property searchers, buyers/sellers, real estate agents, and administrative staff. Core functionalities include property search with filtering, image galleries, real-time bidding systems, WebSocket-based chat, user authentication, payment processing, and analytics dashboards. The platform is designed to scale horizontally to accommodate fluctuating user loads while maintaining low latency and high availability.

Architecture Decisions

Multi-Region Active-Active Deployment: We implemented an active-active architecture across US-EAST-1 and EU-WEST-1 regions to reduce latency for users in different geographical locations and provide disaster recovery capabilities. Route 53 uses geolocation-based routing to direct users to their nearest region, ensuring optimal response times.

Five-Tier Subnet Architecture: The network is segmented into five distinct tiers - Public, Application, Data, Integration, and Developer subnets. This separation provides security through isolation, allows for granular access control via security groups, and enables independent scaling of each tier based on specific resource requirements.

Multi-AZ Deployment: Each region utilizes two availability zones with synchronized deployments. This design ensures that if one AZ experiences failure, the other can handle 100%

of the traffic without service disruption. Load balancers automatically distribute traffic and perform health checks to route requests only to healthy instances.

Hybrid Database Strategy: We selected RDS PostgreSQL for transactional data requiring ACID compliance, ElastiCache Redis for session management and frequently accessed data caching, and OpenSearch for full-text property search capabilities. This combination optimizes for different data access patterns and query types.

Serverless Integration Layer: Lambda functions handle third-party API integration to provide automatic scaling without provisioning servers, reduce costs during low-traffic periods, and isolate external service failures from core application logic. SQS and SNS provide reliable message queuing and event notification.

Auto Scaling with ECS/EKS: Containerized application deployment using ECS/EKS provides consistent environments across development and production, enables rapid deployment rollbacks, and allows for efficient resource utilization through container orchestration and automatic scaling based on CPU and memory metrics.

Reasoning

The multi-region architecture addresses the global nature of real estate transactions where latency directly impacts user experience. Property searches require sub-second response times, which cannot be achieved with a single-region deployment serving international users. By deploying in both US and EU regions, we reduce network latency by 60-80% for users in those geographical areas while providing natural disaster recovery through regional redundancy.

The five-tier subnet design implements defense-in-depth security principles. Public subnets contain only internet-facing components (load balancers, NAT gateways, bastion hosts), while application and data tiers reside in private subnets with no direct internet access. This architecture minimizes attack surfaces and ensures that even if the public tier is compromised, attackers cannot directly access databases or application servers. The integration tier's isolation prevents third-party API failures or security vulnerabilities from affecting core platform functionality.

Our database strategy recognizes that different data types have different performance requirements. Property listings require complex searches with filters (OpenSearch), user sessions need microsecond access times (Redis), while transaction data requires strong consistency guarantees (PostgreSQL). Attempting to use a single database for all purposes would result in either performance bottlenecks or data consistency issues. The three-database approach optimizes each data access pattern independently.

The serverless integration layer using Lambda functions provides cost efficiency for the variable workload of third-party API calls. Real estate data updates occur sporadically throughout the day, making continuously running servers wasteful. Lambda's per-invocation pricing model

means we only pay for actual API integration work, reducing costs by approximately 70% compared to dedicated EC2 instances while providing automatic scaling to handle traffic spikes.

Auto-scaling capabilities are critical for handling the real estate market's inherent traffic variability. Property viewings spike during weekends and evenings, while weekday mornings see minimal traffic. Without auto-scaling, we would need to provision for peak capacity 24/7, wasting resources during low-traffic periods. Our auto-scaling configuration maintains 2-4 baseline instances and scales up to 20+ instances during peak hours, optimizing costs while ensuring performance.

Networking Components Used

Global Layer

- **Amazon Route 53**: DNS service providing geolocation-based routing to direct users to their nearest region, health checks for automatic failover between regions, and low-latency DNS resolution with 100% uptime SLA
- **Amazon CloudFront**: Content Delivery Network caching static assets (images, CSS, JavaScript) at 400+ edge locations worldwide, reducing origin server load by 80% and improving page load times by 50-70%
- **AWS Shield Standard & Advanced**: DDoS protection defending against network and application layer attacks, automatic traffic analysis, and 24/7 response team for Advanced tier protecting against volumetric attacks up to 10+ Gbps

Regional Components

Public Subnet (10.0.1.0/24, 10.1.1.0/24)

- **Internet Gateway**: Enables bidirectional internet connectivity for public subnet resources, allowing load balancers to receive incoming traffic and NAT gateways to forward outbound traffic
- **NAT Gateway**: Provides secure outbound internet access for private subnet resources (application servers, Lambda functions) to reach external APIs and services while preventing inbound connections
- **Application Load Balancer (ALB)**: Layer 7 load balancing with SSL/TLS termination, path-based routing to different services (search, bidding, chat), health checks, and sticky sessions for WebSocket connections
- **Bastion Host**: Secure jump server for administrative SSH access to private subnet instances, with MFA authentication and session logging for audit compliance

Application Tier (10.0.10.0/24, 10.1.10.0/24)

- **EC2 Auto Scaling Groups**: Automatically adjusts compute capacity (2-20 instances) based on CPU utilization and request counts, ensuring availability during traffic spikes while minimizing costs during low-traffic periods
- **ECS/EKS Clusters**: Container orchestration managing microservices deployment, providing service discovery, automatic container replacement on failure, and rolling updates with zero downtime
- **API Gateway**: RESTful API management with request throttling, API key management, request/response transformation, and integration with Lambda functions for serverless endpoints
- **Elastic Network Interfaces**: Attach multiple private IP addresses to instances, enable hot-swap capability for maintaining IP addresses during instance replacement, and support enhanced networking for higher bandwidth

Data Tier (10.0.20.0/24, 10.1.20.0/24)

- **RDS PostgreSQL Multi-AZ**: Primary-standby database configuration with automatic failover (60-120 seconds), automated backups (35-day retention), read replicas for reporting queries, and point-in-time recovery
- **ElastiCache Redis**: In-memory cache storing user sessions, frequently accessed property data, and search results, providing sub-millisecond latency and reducing database load by 70%
- **Amazon OpenSearch**: Distributed search and analytics engine indexing property data with full-text search, geo-spatial queries for location-based searches, and faceted filtering for advanced property searches

Integration Tier (10.0.30.0/24, 10.1.30.0/24)

- **AWS Lambda**: Serverless functions processing third-party API calls, image processing, data validation, and scheduled tasks without managing servers, with automatic scaling to handle 1,000+ concurrent executions
- **Amazon SQS**: Message queue decoupling services, buffering API requests during traffic spikes, ensuring reliable delivery with dead-letter queues for failed messages, and supporting delayed message processing
- **Amazon SNS**: Pub-sub messaging for event notifications, sending alerts to multiple subscribers (email, SMS, webhooks), and triggering Lambda functions based on property updates or bid events
- **Amazon EventBridge**: Event bus routing events between services, scheduling recurring tasks, and integrating with SaaS applications through partner integrations

Developer Tier (10.0.40.0/24, 10.1.40.0/24)

- **VPN Gateway**: IPsec VPN connection providing encrypted tunnel for developer access from corporate networks, supporting 1.25 Gbps throughput and multiple VPN connections for redundancy

- **AWS CodePipeline**: CI/CD automation orchestrating build, test, and deployment stages, integrating with GitHub/GitLab, and enabling blue-green deployments with automatic rollback on failure
- **AWS CodeBuild**: Managed build service compiling source code, running tests, and producing deployment artifacts in isolated build environments with configurable compute resources

Cross-Region Components

- **S3 Cross-Region Replication**: Automatically replicates property images and documents between regions with 15-minute replication time, providing data redundancy and enabling low-latency access from either region
- **AWS Transit Gateway**: Hub-and-spoke network architecture connecting VPCs within and across regions, simplifying network topology, and enabling centralized routing policies with 50 Gbps bandwidth per connection
- **AWS Global Accelerator**: Uses AWS global network to route traffic to optimal regional endpoints, providing static anycast IP addresses, automatic regional failover, and improved performance with 60% faster routing compared to internet paths
- **VPC Peering**: Direct network connection between VPCs in different regions for private database replication, file synchronization, and cross-region service communication without traversing the internet

Security & Monitoring

- **AWS WAF**: Web Application Firewall protecting against SQL injection, XSS attacks, and bot traffic with managed rule sets and custom rules, blocking malicious requests before they reach application servers
- **Security Groups**: Stateful firewall rules controlling inbound/outbound traffic at instance level, implementing least-privilege access with separate groups for load balancers, application servers, and databases
- **Amazon CloudWatch**: Centralized logging and metrics collection, custom dashboards for infrastructure monitoring, alerting on anomalies, and log aggregation from all services with 15-month retention
- **AWS CloudTrail**: Audit logging of all API calls across AWS services, compliance reporting, security analysis, and forensic investigation capabilities with immutable log storage in S3
- **AWS Secrets Manager**: Secure storage and automatic rotation of database credentials, API keys, and encryption keys, with encryption at rest and fine-grained IAM access control
- **AWS KMS**: Key Management Service for encryption key generation, management, and rotation, supporting envelope encryption and integration with all AWS services for data encryption

Cost Estimation

Methodology

- All costs are estimated for AWS US-EAST-1 region (EU-WEST-1 will have similar costs)
- Pricing based on November 2025 AWS pricing (subject to change)
- Costs include data transfer, storage, and compute but exclude domain registration and SSL certificates
- Assumes 30-day month for monthly calculations
- Concurrent users: simultaneous active users; Monthly users: unique users per month

Cost Breakdown by User Load

Table 1: Concurrent Users Cost Estimation (USD/month)

Component	100 Concurrent	10,000 Concurrent	100,000 Concurrent
Compute (EC2 + ECS/EKS)	$150 (2 × t3.medium)	$2,400 (8 × c5.2xlarge)	$18,000 (60 × c5.4xlarge)
Load Balancers (ALB)	$25 (1 ALB)	$75 (3 ALB)	$200 (6 ALB)
RDS PostgreSQL	$180 (db.t3.large Multi-AZ)	$1,200 (db.r5.2xlarge Multi-AZ)	$8,500 (db.r5.12xlarge Multi-AZ)
ElastiCache Redis	$85 (cache.t3.medium)	$680 (cache.r5.2xlarge)	$4,800 (cache.r5.12xlarge)
OpenSearch	$120 (2 × t3.medium.search)	$980 (3 × r5.2xlarge.search)	$7,200 (6 × r5.4xlarge.search)
Lambda Functions	$20 (1M invocations)	$180 (10M invocations)	$1,500 (100M invocations)
API Gateway	$15 (1M requests)	$140 (10M requests)	$1,200 (100M requests)
S3 Storage + Transfer	$50 (500 GB storage)	$400 (5 TB storage)	$3,500 (50 TB storage)
CloudFront CDN	$35 (1 TB transfer)	$280 (10 TB transfer)	$2,400 (100 TB transfer)
NAT Gateway	$45 (2 NAT × 1 TB)	$180 (2 NAT × 5 TB)	$800 (4 NAT × 20 TB)
VPN Gateway	$72 (1 VPN connection)	$144 (2 VPN connections)	$216 (3 VPN connections)
SQS + SNS	$8 (1M messages)	$65 (10M messages)	$520 (100M messages)
CloudWatch Logs	$15 (10 GB logs)	$95 (100 GB logs)	$750 (1 TB logs)
Route 53	$10 (hosted zone + queries)	$25 (hosted zone + queries)	$80 (hosted zone + queries)
Security (WAF, Shield)	$25 (WAF only)	$125 (WAF + rules)	$3,250 (Shield Advanced)
Secrets Manager + KMS	$12 (20 secrets)	$30 (50 secrets)	$80 (200 secrets)

Component	100 Concurrent	10,000 Concurrent	100,000 Concurrent
Data Transfer (inter-region)	$20 (200 GB)	$180 (2 TB)	$1,500 (20 TB)
Backup & Snapshots	$30 (automated backups)	$150 (automated backups)	$800 (automated backups)
Transit Gateway	$50 (attachments + data)	$120 (attachments + data)	$450 (attachments + data)
Global Accelerator	$38 (2 accelerators)	$76 (2 accelerators)	$152 (4 accelerators)
CloudTrail	$8 (event logging)	$15 (event logging)	$35 (event logging)
TOTAL PER REGION	**$1,017**	**$7,544**	**$55,933**
TOTAL (2 REGIONS)	**$2,034**	**$15,088**	**$111,866**

Table 2: Monthly Users Cost Estimation (USD/month)

Component	100K Monthly	1M Monthly	10M Monthly	100M Monthly
Compute (EC2 + ECS/EKS)	$300	$1,200	$8,500	$42,000
Load Balancers (ALB)	$50	$150	$400	$800
RDS PostgreSQL	$360	$720	$3,600	$15,000
ElastiCache Redis	$170	$340	$2,400	$9,600
OpenSearch	$240	$480	$3,600	$14,400
Lambda Functions	$40	$120	$800	$6,000
API Gateway	$30	$90	$600	$4,500
S3 Storage + Transfer	$100	$300	$2,000	$12,000
CloudFront CDN	$70	$210	$1,400	$8,400
NAT Gateway	$90	$180	$600	$2,400
VPN Gateway	$72	$144	$216	$288
SQS + SNS	$16	$48	$320	$2,400
CloudWatch Logs	$30	$60	$400	$2,400
Route 53	$10	$20	$60	$200
Security (WAF, Shield)	$50	$100	$500	$4,000
Secrets Manager + KMS	$24	$36	$60	$120
Data Transfer (inter-region)	$40	$120	$800	$4,800
Backup & Snapshots	$60	$120	$600	$2,400
Transit Gateway	$100	$150	$300	$900
Global Accelerator	$76	$114	$152	$304
CloudTrail	$8	$12	$25	$60
TOTAL PER REGION	**$1,946**	**$4,614**	**$27,333**	**$132,972**

Component	100K Monthly	1M Monthly	10M Monthly	100M Monthly
TOTAL (2 REGIONS)	**$3,892**	**$9,228**	**$54,666**	**$265,944**

Key Cost Drivers Analysis

1. **Compute Resources (30-35%)**: EC2/ECS instances represent the largest cost component, scaling linearly with user load and requiring right-sizing based on actual utilization patterns.
2. **Database Services (25-30%)**: RDS, ElastiCache, and OpenSearch combined form the second-largest expense, with costs driven by instance sizes and storage requirements.
3. **Data Transfer (15-20%)**: Cross-region replication, CloudFront CDN, and NAT Gateway data transfer costs increase significantly with user growth and content consumption.
4. **Serverless Components (10-15%)**: Lambda, API Gateway, and SQS provide cost efficiency at low scales but become significant expenses at high volumes.
5. **Security & Monitoring (5-10%)**: WAF, Shield Advanced (at large scale), CloudWatch, and CloudTrail provide essential security and observability.

Cost Optimization Recommendations

1. **Reserved Instances**: Purchase 1-year or 3-year reserved instances for baseline capacity (40-60% cost savings on compute)
2. **Savings Plans**: Commit to consistent compute usage for flexible savings across EC2, Lambda, and Fargate
3. **S3 Intelligent-Tiering**: Automatically move infrequently accessed data to cheaper storage tiers
4. **CloudFront Cache Optimization**: Increase cache TTL to reduce origin requests and data transfer costs
5. **Right-Sizing**: Regular review of instance utilization to downsize over-provisioned resources
6. **Spot Instances**: Use spot instances for non-critical workloads (development, testing, batch processing)
7. **Lambda Memory Optimization**: Tune Lambda function memory allocation to balance execution time and cost
8. **Data Transfer Optimization**: Use VPC endpoints to avoid NAT Gateway charges for AWS service access