C2C Marketplace

Architecture Design Document

<<date>>

Nikhil Gupta

This document is an Architecture Design Document for developing **C2C Marketplace**.

REVISION HISTORY

Version	Date	Author	Description
0.1	<<date>>	Nikhil Gupta	Initial document creation

1. Overview

// A1. System Definition

// C1-1. Is the defined system boundary clear?

// C1-2. Is there sufficient explanation of the system's operation and business environment as business drivers?

1.1 Introduction

A Customer-to-Customer (C2C) Marketplace is a platform that allows users to **safely buy and sell** both used and new products. At a high level, a C2C marketplace:

1. **On-boards and verifies** buyers & sellers (authentication, KYC, fraud checks, peer ratings, policy Compliance).
2. **Publishes and Discovers listings via** content moderation, catalog indexing, advanced search, and personalized recommendations.
3. **Enables Communication and Negotiation flows** between buyers & sellers (secure chat, offers/counter-offers, order initiation).
4. **Offers Moderation and Fraud Detection** across listings, messages, and payments (image/text moderation, ML-based risk scoring and dispute resolution).

1.2 System Definition

Customer (C2C) Marketplace is a platform that allows users to **safely buy and sell** both used and new products. At a high leveldasdaf akjfdsghnafdgad

The purpose of this project is to deliver a multi-region, production-grade marketplace supporting new/used goods, enabling **listing → discovery → negotiation → payment (escrow) → fulfillment → rating/dispute** with strong trust & compliance.

The figure below depicts the system boundary which shows how the system will interact with outside components and actors:

1. Iuhhgdsfaiuydshgfauyds

2. Dassd

//Decide if buyer and seller to be mentioned individually or tagged as User in general

Hifdsfhusifsfufs

FDSAF

DSA

The chatgpt verion

System boundary — “Marketplace Platform” includes:

- **Client apps & public APIs** (web/mobile, partner APIs).
- **Core services**: Identity/Auth/KYC, Listings, Search/Recommendations, Messaging/Offers, Checkout/Escrow, Fulfillment, Reviews, Disputes, Moderation, Risk/Fraud.
- **Data/ML & Observability**: event streaming, lakehouse/feature store, A/B testing, metrics/logs/traces, audit.
- **Admin/Operations**: policy & catalog governance, promotions/fees, experiments, support console.

External actors/systems:

- **Payments & Escrow**, **KYC/Identity**, **Content Moderation**, **Logistics/Carriers**, **Tax/Compliance**, **CRM/Marketing**, **Analytics/Attribution**.

Figure 1 (conceptual, analogous to your reference)

1. **Client entry**: User opens app/site → CDN/edge → API Gateway → Marketplace Platform.
2. **Primary interactions**: Buyers/Sellers act only through clients—browse/search, list/manage, message/offer, pay, confirm delivery, rate, raise disputes.
3. **Operational integrations**: External providers exchange events/webhooks; operators configure policies, promos, and rollouts in an admin console.
   (Maps to the boot ROM → bootloader → kernel chain, but with clients → platform → external providers.)

DSF

FDSA

1.3 Business Context & Drivers

2 ~~Our marketplace serves buyers and sellers across regions, devices, and network conditions. Inventory is user-generated and long-tail, with uneven titles, images, and metadata; demand is spiky and intent can fade quickly. Shoppers expect instant, relevant answers and smooth handoffs from search to chat to checkout.~~

3 ~~The peer-to-peer nature also introduces real risks — counterfeits, scams, off-platform payments—that must be contained without adding friction. Success therefore hinges on surfacing the right items fast, keeping interactions responsive, and building enough trust for people to complete the deal. These realities lead directly to the following business drivers:~~

4 Our marketplace connects buyers and sellers across many regions, devices, and internet speeds. Since listings are created by users and often cover niche items, the titles, photos, and details are inconsistent. Also, demand can rise suddenly, and shopper interest drops quickly.

Shoppers expect instant, relevant results and a seamless handoff from search to chat to checkout.

5 The peer-to-peer model also creates real risks, including counterfeits, scams, and attempts to pay off the platform. We must reduce these risks without adding any friction. Our success depends on showing the right items quickly, keeping every interaction responsive, and building enough trust for people to complete their purchase. These realities lead to the following business drivers.

1. **Speed & Responsiveness:** Keep search and checkout interactions fast; ensure new listings index quickly.
2. **Relevance & Recommendations:** Return highly relevant results and personalized recommendations to boost discovery and intent.
3. **Trust & Safety:** Protect buyers/sellers with strong fraud prevention, content moderation, secure payments, and fair disputes.
4. **Conversion:** Maximize search-to-purchase completion by removing friction across discovery, negotiation, checkout, and fulfillment.

2. Requirements

2.1. Functional Requirements

// A2. Functional Requirement Specification

// C2-1. Is there sufficient functional requirement specification to affect the system's architecture?

// C2-2. Is the relationship between use cases clear?

// C2-3. Is the division of use cases explicit?

Nikhil Checkpoints: Pending : Common Use case Diagram

Search ; recc; chat /enquiries ; transaction(Seq & Status – >payment) ; notification ; product listing ; landing page → product details.

UC_01	Product Listing
Description	Seller can Create/Update or Delete a product listing.
Actor	User (Seller)
Pre-condition	Seller is signed-in (with minimum KYC done). / Seller Is authorized
Post-condition	New Listing is created in the system and changes are propagated to Search & recommendation Modules.
Basic Flow	1. Seller uploads images and enters item details like Category, Title, Description, Price, Location etc. 2. The system runs content moderation checks to check the filled data. 3. Seller validates the filled data and publishes the listing.
Additional Flow	1. The system auto saves drafts regularly and lets the seller restore the latest version after interruptions. 2. If policies are violated, the system explains the issue and offers fix options or an appeal to review.

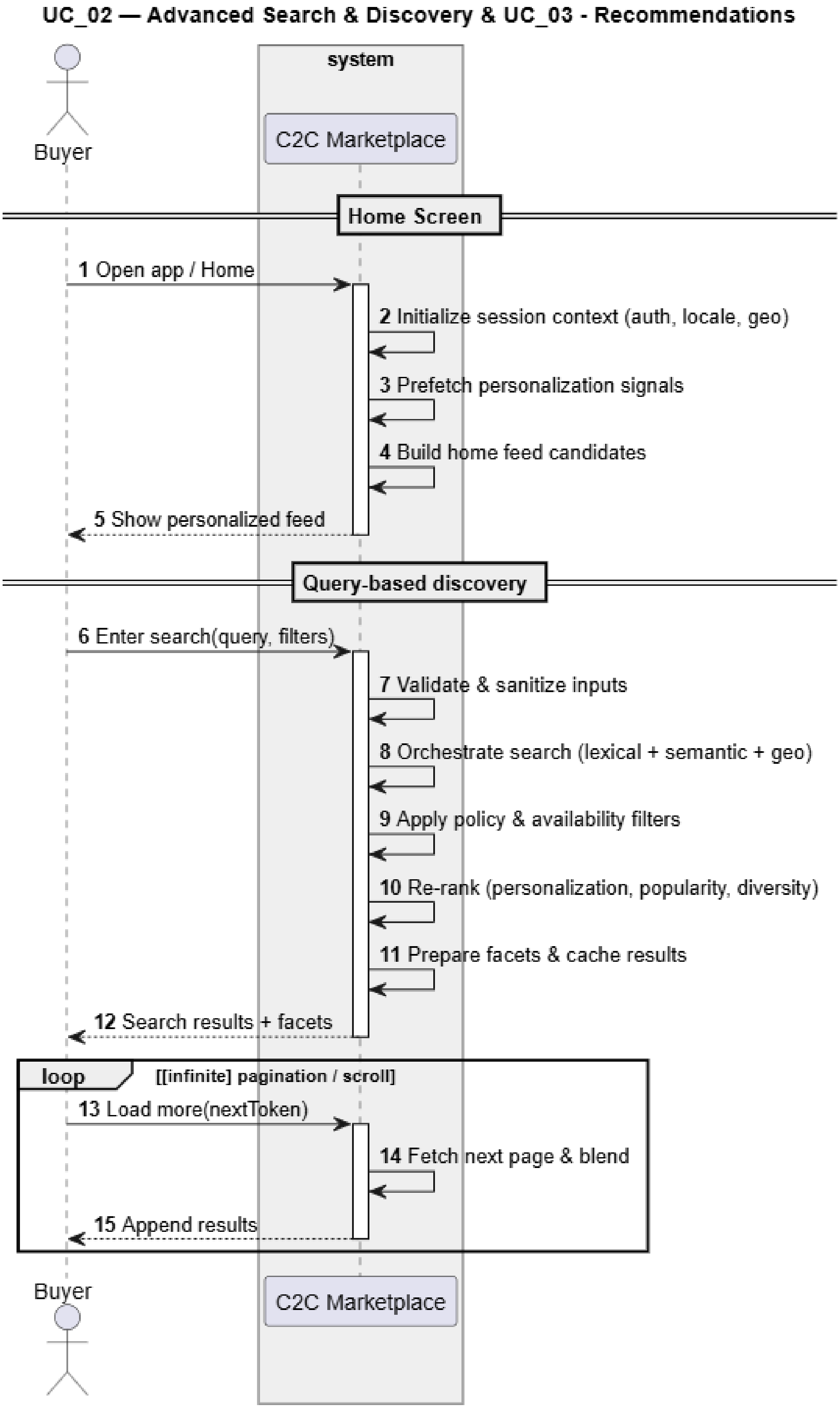
UC_02	Advanced Search
Description	Advanced Item Search & Discovery — blended semantic/keyword/geo search with fast, relevant results.
Actor	User (Buyer)
Pre-condition	1. Listings are already created and are indexed 2. User Locale & location are available
Post-condition	1. Results returned with categorical breakdowns 2. Search/engagement events logged.
Basic Flow	1. Buyer enters query or opens home feed. 2. System executes lexical + vector (semantic) + geo search. 3. Apply policy/availability filters; de-duplicate. 4. Re-rank with personalization and popularity signals. 5. Return paginated results and aggregates; log telemetry.
Additional Flow	No results → relax filters, semantic expansion, similar items. Saved search → subscribe and notify on new matches. Degraded network → fallback to cached/trending near user.

UC02 – can user change his location.

Wishlist features.

UC_03	Personalized Recommendations
Description	Proactive, context-aware recommendations dynamically that will be shown across home feed, similar or related items (on Product Display Page & Cart) adapting to page context, recent behavior, location, and availability that boost discovery and purchase intent.
Actor	User (Buyer)
Pre-condition	
Post-condition	Personalized items shown; recs impressions/clicks logged for learning.
Basic Flow	Scenario A: 1. Buyer Lands Buyer lands on a surface that supports recommendations (home, PDP, cart, inbox/push). 2. System assembles candidates (similar items, trending near buyer, recently viewed, and complementary items). 3. Apply eligibility/policy filters (availability, region/category rules, blocked

	sellers, deduplication). 4. Rank candidates using session intent, user/item similarity, diversity, price/condition fit, and distance (for local). 5. Render cards with brief explanations (e.g., “Because you viewed X”) and CTAs (View, Save, Make Offer, Buy Now). 6. Log impressions/clicks/add-to-cart/offer events and update session features for future requests.
Additional Flow	1. Cold start: Use trending, editorial collections, and location-aware bestsellers. 2. Saved signals: Trigger saved-search reminders and price-drop alerts when inventory changes. 3. Latency guardrails: Degrade gracefully to cached/trending if ranking exceeds budget; suppress widget if empty. 4. Trust alignment: Exclude prohibited/risky items per policy; respect user blocks and privacy/consent settings. 5. Experimentation: A/B variants with guardrails (no harm to speed/checkout); frequency caps and diversity constraints.


UC_02 — Advanced Search & Discovery & UC_03 - Recommendations
system
Buyer
C2C Marketplace
Home Screen
1 Open app / Home
2 Initialize session context (auth, locale, geo)
3 Prefetch personalization signals
4 Build home feed candidates
5 Show personalized feed
Query-based discovery
6 Enter search(query, filters)
7 Validate & sanitize inputs
8 Orchestrate search (lexical + semantic + geo)
9 Apply policy & availability filters
10 Re-rank (personalization, popularity, diversity)
11 Prepare facets & cache results
12 Search results + facets
loop
[[infinite] pagination / scroll]
13 Load more(nextToken)
14 Fetch next page & blend
15 Append results
Buyer
C2C Marketplace

UC_04	Secure Chat Messenger
Description	Buyer–Seller Messaging & Offers — secure chat and structured offers/counteroffers.
Actor	Buyer, Seller; Risk/Moderation (supporting)
Pre-condition	Valid accounts; listing exists; messaging permitted per policy.
Post-condition	1. Messages/offers stored and audited. 2. Accepted offer creates an order draft.
Basic Flow	1. Buyer opens chat on a listing and sends a message 2. Buyer or seller sends an offer with price 3. Counter/accept/decline handled; acceptance creates order draft. 4. Notifications sent to the counterparty.
Additional Flow	1. Safety filters block off-platform payment solicitations/scams. 2. Report/block user; rate limits applied for spam/abuse.

UC_05	User Notifications
Description	
Actor	
Pre-condition	
Post-condition	
Basic Flow	
Additional Flow	

UC_06	Checkout & Escrow
Description	
Actor	
Pre-condition	
Post-condition	

Basic Flow	
Additional Flow	

UC_07	Fraud & Risk
Description	
Actor	
Pre-condition	
Post-condition	
Basic Flow	
Additional Flow	

2.2. Non-functional Requirements

// A5. Quality Requirement Specification

// C5-2. Is the specification of quality requirements appropriate?

// C5-3. Is quality requirement measurable?

// C5-4. Is the allowance of non-functional requirement clear?

NFR_01		
Description		
Environment		
Stimulus		
Response		
Measure		
Allowance		

2.3. Quality Attributes

// A5. Quality Requirement Specification

// C5-2. Is the specification of quality requirements appropriate?

// C5-3. Is quality requirement measurable?

QA_01		
Description		
Environment		
Stimulus		
Response		
Measure		

3. Architecture

// A8. Architecture Documentation

// C8-1. Is allocation of processes, etc. appropriate? (deployment)

// C8-2. Is grouping appropriate in terms of components? (component & connector)

// C8-3. Is the description of the system architecture appropriate?

4. Modules

// A9. Module Specification

// C9-1. Is component specification sufficient to develop?

// C9-2. Is grouping appropriate in terms of module?

// C9-3. Is it appropriate to design dependencies between modules?

// C9-4. Is the work assignment appropriate?

Appendix

A. Domain Model

// A3. Domain Model Design

// C3-1. Is domain model sufficiently sub-divided?

// C3-2. Does domain model reflect architecture decisions?

B. Quality Scenarios

// A4. Quality Scenario Elicitation

// C4-1. Is there sufficient scenario elicitation affecting the architecture?

// C4-2. Is there sufficient review of the quality related to performance?

// C4-3. Is there sufficient review of the quality related to modifiability?

C. Quality Scenario Analysis

// A5. Quality Requirement Specification

// C5-1. Is quality scenario analysis appropriate? (evidence)

D. Candidate Architectures

// A6. Candidate Architecture Design

// C6-1. Are quality analysis and solution candidate appropriate?

// C6-2. Are performance analysis and solution candidate appropriate?

// C6-3. Are modifiability analysis and solution candidate appropriate?

E. Candidate Architecture Evaluation

// A7. Architecture Design

// C7-1. Is comparison analysis of colliding candidates appropriate? (evidence)

// C7-2. Is there sufficient complement of the selected candidate?

F. Final Architecture

// A7. Architecture Design

// C7-3. Is there right integration into the final architecture?

// C7-4. Is there appropriate risk management of the final architecture?

G. Architecture Evaluation(ATAM)

// A10. Architecture Evaluation

// C10-1. Are there sufficient quality scenarios evaluating architecture?

// C10-2. Are there sufficient architectural decisions identified?

// C10-3. Is the analysis of design decisions appropriate? (evidence)

// C10-4. Are the mitigation plans to the risk factors appropriate?