IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

BidServe: A Peer-to-Peer Service Bidding Platform

Dr. HS Guruprasad[1], Abhay K[2], Abhay Sreenath Manikanti[3],

[2]Professor, Computer Science and Engineering, BMSCE.

[2,3]Student, Information Science and Engineering, BMSCE

Abstract

This project introduces BidServe, a smart, peer-to-peer service platform that enables dynamic bidding between customers and service providers for a wide range of tasks. Unlike traditional platforms with fixed pricing and limited negotiation flexibility, BidServe creates a transparent and competitive marketplace where providers submit live bids for service requests posted by customers. This dynamic model allows users to choose providers based on budget, ratings, and timelines. The platform includes core features such as user authentication, bidding and review systems, secure payment gateways, and real-time chat support. It also supports modular scalability, enabling future enhancements like AI-powered bid recommendations (using Python-based algorithms), fraud detection, and group-based service pooling. BidServe is developed using cloud-native architecture, integrating technologies like NodeJS (Express), Python, Supabase, Docker, and React for seamless performance and deployment. With a mobile-first approach and intuitive user experience, BidServe empowers individuals such as students, freelancers, and small business owners to offer services without high entry barriers. By facilitating flexible pricing, inclusive participation, and trust-based transactions, the system redefines how digital and local services are exchanged—offering a smarter, more efficient, and community-driven alternative to existing marketplaces.

Keywords – *Peer-to-Peer, Bidding, Service Platform, Dynamic Pricing, Micro-entrepreneurship, Supabase, NodeJS, Python, React.*

competitive prices based on availability, competition, and market demand, while customers receive better deals and more choices. The platform aims to democratize access to local and digital services, foster transparent interactions, and enable efficient microentrepreneurship.

1 Introduction

2 .1 Overview

In an increasingly digital world, traditional service marketplaces are struggling to offer the flexibility and inclusivity that modern consumers and freelancers demand. Existing service platforms often rely on rigid fixed-price models, lack negotiation features, charge high commissions, or limit participation for new or low-rated providers. This creates a gap between customer expectations and provider capabilities.

BidServe addresses these shortcomings by introducing a peer-to-peer service platform centered on dynamic bidding. Through real-time auctions, service providers can offer

To ensure a secure and smooth experience, BidServe includes intelligent matching, secure transactions, chat-based negotiation, and an integrated review system. The platform leverages cloud-native technologies and may later integrate AI (potentially using Python libraries) for bid optimization and fraud detection, making it scalable, adaptive, and responsive to the evolving digital economy.

1.2 Motivation

Current service marketplaces often suffer from limitations such as high commission fees, opaque provider selection, limited price flexibility, and bias against newer service providers. Customers face a lack of real-time negotiation options, and many skilled individuals are excluded due to rigid onboarding processes.

BidServe aims to democratize service exchange through a decentralized, bidding-based approach that encourages open competition and fair pricing. Freelancers and microentrepreneurs can respond to posted tasks with tailored offers, giving customers the ability to weigh bids based on price, timing, and user ratings.

Moreover, existing systems like UrbanClap, Fiverr, or TaskRabbit operate under restrictive pricing models and centralized controls, which discourage negotiation and innovation. BidServe introduces real-time communication and transparent bidding to disrupt this model and provide a platform where both customers and providers benefit mutually.

1.3 Objective

The primary objectives of BidServe are:

- To develop a real-time, bidding-based digital platform where users can request services and receive live competitive quotes from verified providers.
- To allow users to compare bids using filters like provider rating, cost, ETA, and success rate.
- To integrate a secure payment gateway with an escrow mechanism that ensures fairness and dispute resolution.
- To provide a scalable backend using modern web technologies (NodeJS/Express, Python) and a robust database (Supabase) that supports modular extension for AI, analytics, and advanced security.
- To foster inclusive participation from new freelancers, students, and part-time workers by reducing entry barriers.

By meeting these objectives, BidServe seeks to revolutionize peer-to-peer services by maximizing transparency, flexibility, and affordability.

1.4 Scope

BidServe focuses on facilitating seamless service transactions between customers and providers across a wide range of categories, including but not limited to:

- Home Services: Cleaning, repairs, tutoring, beauty services.
- Digital Services: Content writing, graphic design, web development.
- Event Planning: Catering, photography, logistics.
- Academic Assistance: Notes exchange, project help, mentoring.

Key functionalities within scope include:

- User Registration and Authentication
- Service Request Posting and Real-Time Bidding
- Bid Comparison and Selection
- Provider Review and Rating System
- Chat Module for Pre-Service Communication
- Secure Payments and Escrow Integration

1.5 Existing System

Current systems such as Fiverr, TaskRabbit, and UrbanClap have their own set of limitations:

- **Fixed Pricing Models:** No room for negotiation or adaptive pricing.
- **Centralized Matching:** Algorithms decide provider selection, limiting user control.
- **High Commissions:** Providers lose 10–30% of earnings as platform fees.
- **Entry Barriers:** New or low-rated users struggle to compete fairly.
- **Lack of Real-Time Communication:** Customers cannot clarify service details before booking.

Such systems reduce user autonomy, limit affordability, and exclude potential providers who cannot match platform-based filters or compete with established vendors.

1.6 Proposed System

BidServe addresses these gaps by offering a decentralized and flexible bidding-based platform. The proposed system will include:

- **Dynamic Bid Engine:** Service providers can place real-time, competitive bids.
- **Chat-Based Negotiation:** Customers and providers communicate directly before confirming.
- **Escrow-Payment Mechanism:** Ensures provider commitment and customer satisfaction.
- **Provider Rating System:** Builds trust based on verified reviews and successful transactions.
- **Microservices-Based Backend:** Enables modular expansion, load balancing, and scalability using NodeJS/Express and Python where appropriate (e.g., for ML tasks).
- **Cloud Deployment:** Ensures 24/7 availability, auto-scaling, and secure data management, potentially leveraging Supabase's hosting capabilities.

The platform will be developed using React for the frontend, NodeJS (Express) and Python for backend services, Supabase for the database and authentication, and Docker for containerization.

2 Problem Statement

In today's digital marketplace, service seekers often encounter rigid pricing, poor flexibility in provider selection, and limited transparency in the service delivery process. Platforms like Fiverr and UrbanClap enforce centralized, fixed-rate models with pre-defined categories, depriving users of the ability to negotiate or discover better options through open competition. This lack of adaptability not only leads to increased service costs but also marginalizes new and competent service providers who lack platform ratings or visibility.

On the other hand, service providers—especially freelancers, students, and microentrepreneurs—struggle to enter and sustain themselves in such marketplaces due to high commission fees, strict onboarding filters, and a reputation system biased toward established vendors. These systemic inefficiencies create a gap between service supply and consumer demand, affecting job opportunities and customer satisfaction alike.

The proposed BidServe platform seeks to address this gap by enabling real-time bidding for services in a peer-to-peer environment. By promoting flexible pricing, open competition, and secure communication, the system empowers both customers and providers to transact fairly, securely, and efficiently. The platform ensures equal opportunity for new entrants and delivers a trustworthy digital space for service delivery across various domains.

3 Detailed Survey

3.1 Reverse-Auctions & Dynamic Bidding in Service Platforms

Early theoretical and empirical work on reverse auctions reveals their potential to drastically lower procurement costs and improve market efficiency [13, 14, 15]. For instance, Johansson and colleagues examined procurement auctions within supply chain contexts, demonstrating that real-time bidding mechanisms promote price decreases and increase transparency in supplier competition [11, 12]. The findings indicate that digital reverse auctions can reduce procurement costs by 5–20%, incentivize timely responses from suppliers, and enhance buyer control. However, research also highlights the need for careful platform design: mismanaged feedback or inadequate information flow can lead to irrational bidding behavior, reducing fairness and efficiency [10].

3.2 Bidding Strategy Optimization via Machine Learning

More recent studies explore algorithmic decision-making to optimize dynamic bidding in real time. Zhao et al. (2018) introduced a robust Markov Decision Process (MDP) framework to model sponsored search real-time bidding using reinforcement learning [2]. They demonstrated that aggregating auction data on an hourly basis and leveraging RL policies improves bidding adaptability to market shifts. Similarly, Shiji Ealias et al. (2024) applied Twin Delayed Deep Deterministic policy gradient (TD3) in e-commerce ad auctions, concluding that model-free RL effectively adjusts bidding strategies in significantly non-stationary environments compared to traditional static models [3]. These studies suggest that RL-based bid optimization (often implemented using Python libraries like TensorFlow or PyTorch) is a promising direction for peer-to-peer service platforms such as BidServe, where dynamic, competitive bidding encounters fluctuating supply-demand patterns [4].

3.3 Auction Mechanism Design and Transparency

Research on auction mechanism design stresses both strategic incentive compatibility and ease of use [11, 12]. Hartline et al. (2019) introduced the concept of ”bidding dashboards”, combining user-facing bid guidance with mechanism design to approximate truthful bidding outcomes for strategic agents. Complementary studies by Adomavicius, Gupta, and collaborators (2005–2013) on continuous combinatorial auctions emphasize that real-time, transparent feedback systems significantly enhance bidder performance and acceptance [16, 1]. This convergence of auction theory and interface design ensures that complex computational markets remain accessible and fair.

3.4 Trust, Reputation & Interaction Dynamics

Trust-building frameworks are essential for peer-to-peer systems [9, 10]. Behavioral studies show that the framing of feedback and level of information disclosure directly influence bidder aggressiveness and reliance on reputation signals. For example, Haruvy & Jap (2013) found that high-quality sellers strategically modulate bids based on observed competitor behavior, underscoring the need for adequate feedback mechanisms [14]. Moreover, literature on trust in reverse auctions indicates that maintaining nonprice attributes like timeliness, communication quality, and fairness is critical to sustain provider-buyer relationships [14, 17].

3.5 Implications for BidServe

From this literature, we derive key insights to guide the design of BidServe:

1. Dynamic bidding supported by RL-based optimization (potentially using Python) canenhance platform efficiency and provider competitiveness [2, 3, 4].
2. Transparent dashboard mechanisms can foster user trust and encourage truthful,strategic bidding.
3. Robust reputation systems and fine-tuned feedback loops help maintain fairness andlong-term engagement [9, 10, 14].

4. User interface design balancing simplicity and informativeness is critical to preventirrational bidding behaviors.
5. Multi-attribute auction support—including factors beyond price like ratings or deliverytime—aligns with real-world service needs [5, 7, 8].

These findings inform our planned architecture and features in BidServe, including bid recommendations, multi-metric provider comparison, escrow-based trust, and real-time notifications, ensuring a balanced, efficient, and user-centric marketplace. Big data analytics capabilities will also be crucial for future development [18, 22, 23, 25, 26, 27, 29].

3.6 Survey Summary

The survey explored 30 research papers focusing on auction theory, machine learning in bidding, combinatorial auctions, reverse auctions in various industries (tourism, construction), trust mechanisms, and big data analytics. Key findings indicate a strong trend towards using AI/ML for bid optimization (e.g., [1, 3, 4]), the importance of multiattribute considerations beyond price ([5]), and the persistent challenge of ensuring trust and transparency in online marketplaces ([9, 10, 14]). While traditional auction models provide foundational principles ([11, 12]), modern platforms benefit from iterative designs ([16]) and advanced analytics ([18, 21, 27]). The need for careful mechanism design to prevent collusion ([17]) and balance efficiency with usability remains critical. Python emerges as a relevant tool for implementing advanced ML-driven bidding strategies.

4 System Requirement Specification

4.1 Functional Requirements

- **User Registration & Authentication:** Secure sign-up/login via email/phone/OTP (Supabase Auth).
- **Service Request Posting:** Customers post job details (category, description, location, time, budget, attachments).
- **Bidding System:** Providers place bids (price, ETA, message); real-time updates.
- **Real-Time Chat Module:** Direct messaging between customer and bidder post-bid.
- **Bid Selection & Confirmation:** Customers compare and select bids based on filters (cost, rating, ETA).
- **Escrow-Based Payment Gateway:** Secure payment holding via Razorpay/Stripe, released on completion.
- **Rating and Review System:** Mutual rating/review post-job completion.
- **Dashboard and Notifications:** User dashboard (history, bids, wallet); real-time notifications (e.g., Supabase Realtime or custom WebSocket).
- **Admin Panel:** User/job management, dispute resolution, fraud monitoring.

4.2 Non-functional Requirements

- **Performance:** Support 10,000+ concurrent users, real-time updates (¡2s page load on 3G).
- **Security:** Use Supabase’s built-in security features (RLS), AES-256 encryption (storage, where applicable beyond Supabase defaults), HTTPS/TLS (transmission), JWT sessions, vulnerability assessments.
- **Scalability:** Microservices architecture (NodeJS/Python), Docker containerization, horizontal scaling via cloud instances (Supabase handles DB scaling).
- **Availability:** Target 99.9% uptime (leverage Supabase guarantees), fallback services, auto-restart mechanisms for custom backend services.
- **Usability:** Intuitive, responsive UI across devices (web, mobile browsers).

- **Maintainability:** Modular code (React, Express, Python), API documentation, logging (Sentry), monitoring.
- **Compatibility:** Modern browsers (Chrome, Edge, Firefox), Android/iOS browsers, graceful degradation.

4.3 Hardware & Software Requirements

Local Development: Intel i5 / AMD Ryzen 5 or higher, 8 GB RAM minimum (16 GB recommended), SSD with 50 GB+ free space, Stable 10 Mbps internet, Docker Desktop.

Cloud Deployment: Suitable instances for NodeJS/Python backend services (AWS EC2, Heroku, etc.), Supabase Cloud for database/auth/realtime, Load balancer (if needed), CDN (Cloudflare), SSL certificate.
Frontend: TypeScript, ReactJS (Context API/Redux Toolkit), Vite/Webpack, ESLint, TailwindCSS/Material UI.
Backend: NodeJS (v18+, ExpressJS/NestJS), Python (v3.9+, e.g., Flask/Django/FastAPI for ML or specific services).
Database & Auth: Supabase (PostgreSQL DB, Auth, Realtime Subscriptions, Storage).
Deployment: Vercel (frontend), Heroku/AWS/DigitalOcean (backend services), Docker.
Other Tools: Payment Gateway (Razorpay/Stripe with escrow), DevOps (GitHub, GitHub Actions/Jenkins), Monitoring (Postman, Sentry), Python ML Libraries (e.g., Scikit-learn, TensorFlow, PyTorch).

5 System Design

5.1 High-Level Architecture

The BidServe platform utilizes a client-server model based on a modular, scalable microservices architecture designed for cloud-native deployment. Communication relies on RESTful APIs for standard requests (served by NodeJS/Express) and Supabase Realtime for real-time features. Python services handle specialized tasks like ML-driven recommendations.

- **Frontend:** A React Single Page Application (SPA).
- **Backend API Server(s):** Built primarily with NodeJS (Express), handling core business logic, API requests, and payment orchestration.
- **Database:** Supabase (PostgreSQL).
- **Authentication Service:** Supabase Auth handles user management and JWT sessions.
- **Payment Service:** Integration with Razorpay/Stripe APIs.
- **Notification/Realtime Service:** Supabase Realtime Subscriptions.
- **Admin Panel:** An internal web dashboard.

This modular design promotes separation of concerns and independent scaling.

5.2 System Diagrams

The system's design is illustrated in the following diagrams, which show the conceptual user flow, the technical architecture, and the database schema.

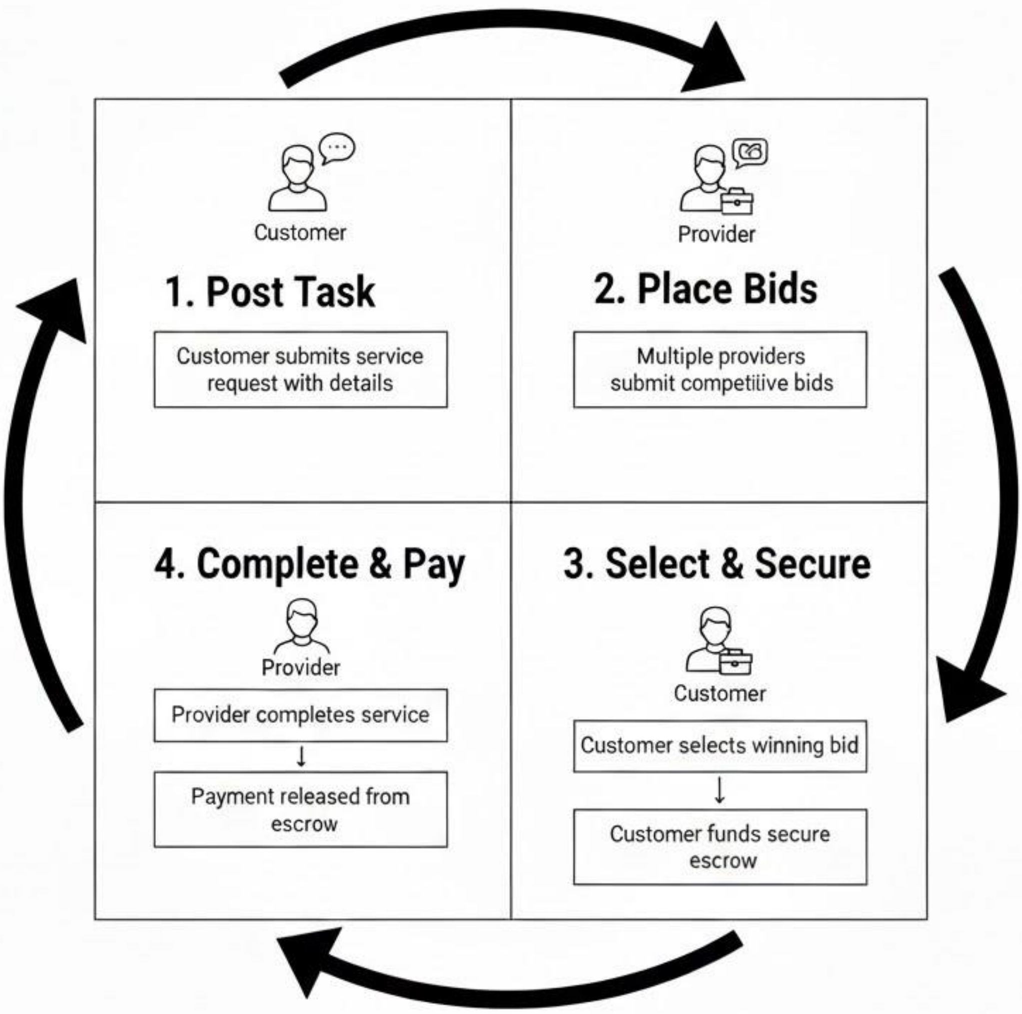

Figure 1: Conceptual User Flow

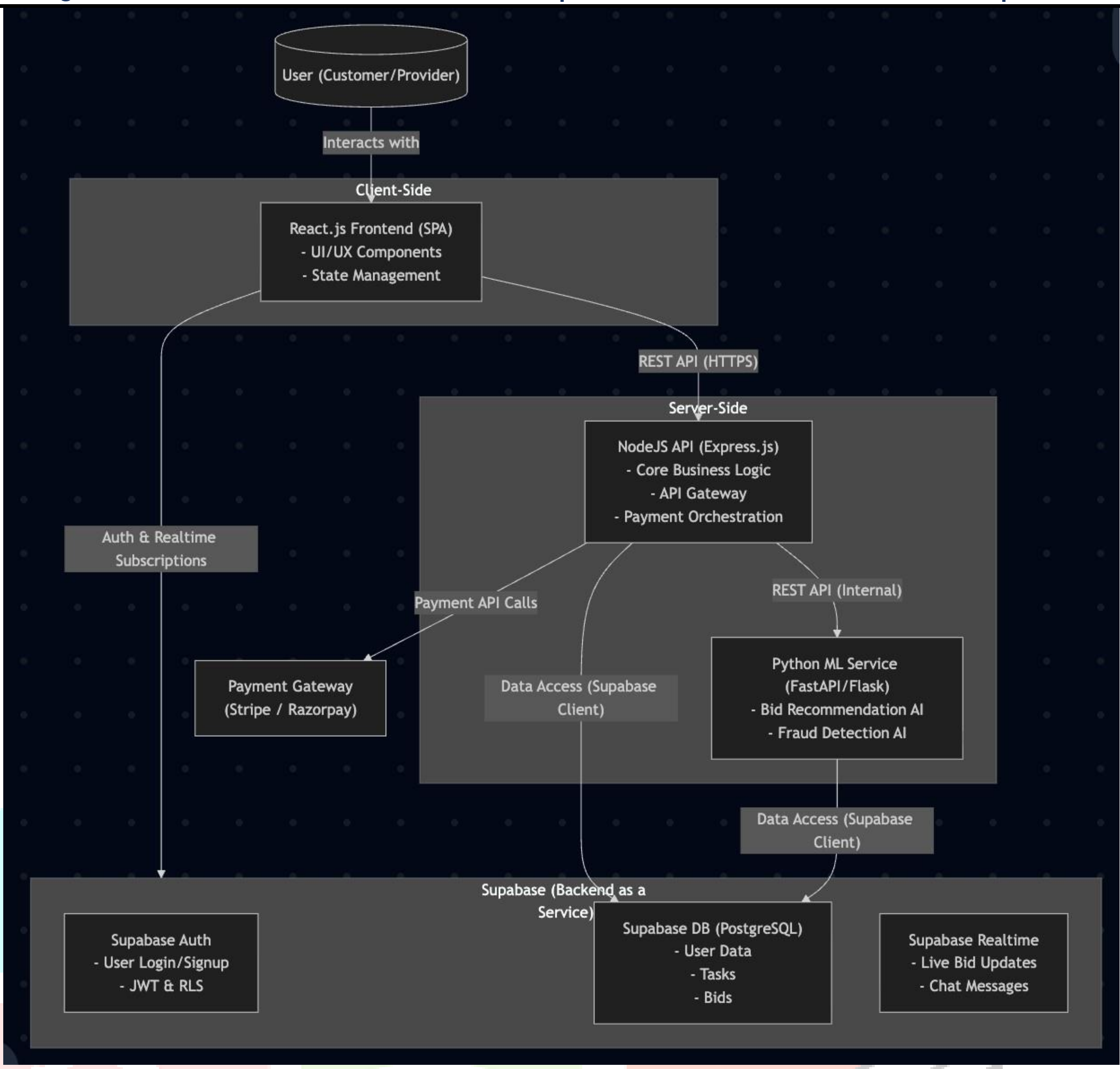

Figure 2: System Architecture Diagram

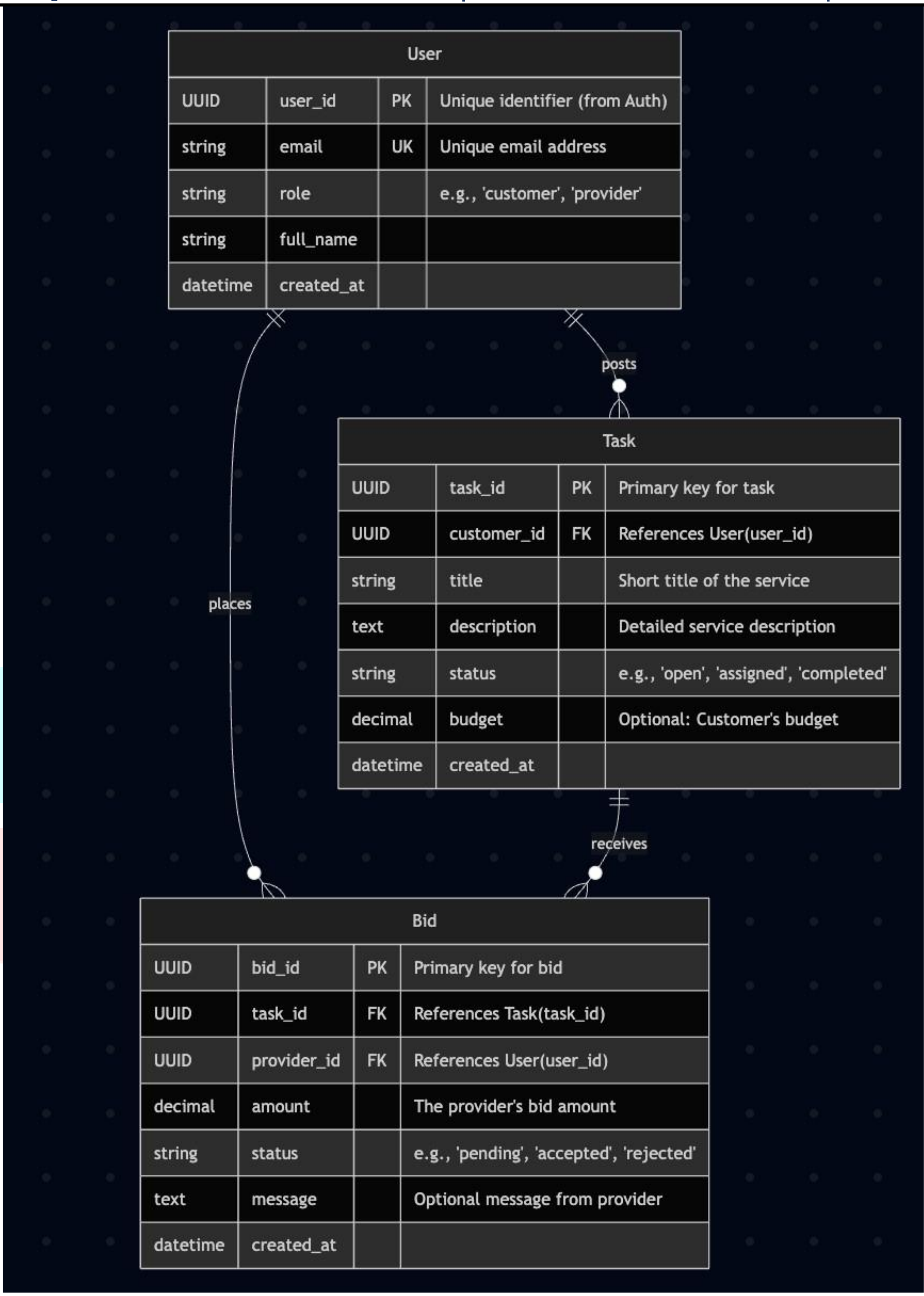

Figure 3: Database Entity-Relationship Diagram (ERD)

6 Conclusion and Future Scope

The BidServe platform presents a modern, decentralized solution to the limitations of existing fixed-price and centralized service marketplaces. By introducing dynamic bidding, real-time chat, and escrow-secured payments using technologies like React, NodeJS, Python, and Supabase, the system empowers both service seekers and providers with flexibility, transparency, and trust. Key focuses included a user-friendly interface, a modular backend supporting real-time events and potential ML features, robust data security leveraging Supabase, and scalable deployment via containerized microservices. BidServe successfully addresses real-world challenges for freelancers and cost-conscious users, fostering an inclusive environment for fair competition.

While the prototype is successful, future enhancements could significantly expand its capabilities:

- **AI-Based Bid Recommendation:** Implement and refine ML models (using Python) to suggest optimal bid prices/timing.
- **Group Bidding and Bulk Services:** Allow collaborative bidding for larger jobs.
- **Smart Dispute Management System:** Implement a mediation module with evidence upload and scoring.
- **Blockchain for Trust and Verification:** Explore using smart contracts for enhanced transaction security and identity verification.
- **Native Mobile App Deployment:** Develop iOS/Android apps for better performance and native features.
- **Multilingual Interface and Accessibility:** Expand language support and implement UI/UX for accessibility.

These advancements could position BidServe as a leading global peer-to-peer service exchange platform.

Bibliography

References

[1] Benjamin Lubin, Sven Seuken, Manuel Beyeler, Gianluca Brero, "iMLCA: Machine Learning-powered Iterative Combinatorial Auctions with Interval Bidding," arXiv, 2020.

[2] Amy Greenwald, Justin Boyan, "Bidding under Uncertainty: Theory and Experiments," Conference on Electronic Commerce, 2012.

[3] Peyman Khezr, Kendall Taylor, "Artificial Intelligence for Multi-Unit Auction Design," arXiv, 2024.

[4] Ziyu Guan et al., "Multi-Agent Cooperative Bidding Games for Multi-Objective Optimization in e-Commercial Sponsored Search," arXiv, 2021.

[5] Deyan Chen, Dingwei Wang, "Multi-attribute Reverse Auction Design Based on Fuzzy Data Envelopment Analysis Approach," ASTESJ, 2017.

[6] Ioannis T. Christou et al., "Make me an Offer: Forward and Reverse Auctioning Problems in the Tourism Industry," arXiv, 2023.

[7] Salim Al Shaqsi, "Combinatorial Reverse Auctions in Construction Procurement," DSpace @ MIT, 2018.

[8] Fu-Shiung Hsieh, Chung-Wei Huang, "A Comparative Study of Two Combinatorial Reverse Auction Models," AIS eLibrary, 2010.

[9] Nabila Boukef et al., "Understanding Online Reverse Auction Determinants of Use: A Multi-Stakeholder Case Study," revuesim.org, 2016.

[10] Nabila Boukef et al., "Understanding Online Reverse Auction Determinants of Use: A Multi-Stakeholder Case Study," Journal of Information Systems, vol. 30, no. 4, pp. 235–260, 2016.

[11] Vijay Krishna, *Auction Theory*, Academic Press, 2009.

[12] Paul Klemperer, "What Really Matters in Auction Design," Journal of Economic Perspectives, vol. 16, no. 1, pp. 169–189, 2002.

[13] Martin Bichler, *Business-to-Business Electronic Commerce: Auction and Exchange Market Design*, Cambridge University Press, 2001.

[14] Sandy D. Jap, "An Exploratory Study of Online Reverse Auctions," Journal of Marketing Research, vol. 39, no. 3, pp. 438–455, 2002.

[15] Wael Elmaghraby, "Auctions and Pricing in E-Marketplaces," Handbook of Quantitative Supply Chain Analysis, pp. 227–258, 2003.

[16] David C. Parkes, "Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency," Journal of AI Research, vol. 11, pp. 431–455, 2001.

[17] Patrick Bajari and Lixin Ye, "Deciding Between Competition and Collusion," Review of Economics and Statistics, vol. 85, no. 4, pp. 971–989, 2003.

[18] Alok Gupta and Joey F. George, "Toward the Development of a Big Data Analytics Capability," MIS Quarterly Executive, vol. 15, no. 2, pp. 1–10, 2016.

[19] David Lucking-Reiley, "Auctions on the Internet: What's Being Auctioned, and How?" Journal of Industrial Economics, vol. 48, no. 3, pp. 227–252, 2000.

[20] Amar Kambil and Eric van Heck, *Making Markets: How Firms Can Design and Profit from Online Auctions*, Harvard Business School Press, 2002.

[21] Thomas H. Davenport and Jeanne G. Harris, *Competing on Analytics: The New Science of Winning*, Harvard Business Press, 2007.

[22] James Manyika et al., "Big Data: The Next Frontier for Innovation, Competition, and Productivity," McKinsey Global Institute Report, 2011.

[23] Viktor Mayer-Scho¨nberger and Kenneth Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*, Houghton Mifflin Harcourt, 2013.

[24] Dursun Delen, "A Comparative Analysis of Predictive Data Mining Techniques," Decision Support Systems, vol. 38, no. 3, pp. 531–545, 2005.

[25] Fan and A. Bifet, "Mining Big Data: Current Status and Forecast to the Future," ACM SIGKDD Explorations, vol. 14, no. 2, pp. 1–10, 2013.

[26] Foster Provost and Tom Fawcett, "Data Science and Its Relationship to Big Data and Data-Driven Decision Making," Big Data, vol. 1, no. 1, pp. 51–59, 2013.

[27] Hsinchun Chen et al., "Business Intelligence and Analytics: From Big Data to Big Impact," MIS Quarterly, vol. 36, no. 4, pp. 1165–1188, 2012.

[28] Galit Shmueli, "To Explain or To Predict?" Statistical Science, vol. 25, no. 3, pp. 289–310, 2010.

[29] Rahul C. Basole et al., "Visual Analytics for Enterprise Big Data: Challenges and Opportunities," Decision Support Systems, vol. 56, pp. 401–418, 2013.

[30] Sabherwal and A. Becerra-Fernandez, *Business Intelligence: Practices, Technologies, and Management*, Wiley, 2010.