# **JuaJobs API Design Document**

## **Platform Overview**

JuaJobs connects African skilled workers (plumbers, electricians, tutors) with clients via a mobile-first platform.

**Key Features**

* **Job Posting and Application**
* **Mobile Money and bank payments (**Rwanda—first)
* **Offline Sync** for low-connectivity areas
* **Multi-Currency Support (**RWF, KES, USD**)**
* **Localized Content (English,** French, Swahili)

## **1. API Architect**

**JuaJobs** connects skilled workers with clients across Africa via a mobile-first platform.

### **1.1 Platform Users**

**Workers**: Skilled professionals (e.g., plumbers, tutors).  
**Clients**: Individuals or businesses posting jobs.  
**Admins**: Handle disputes, verify profiles, and ensure platform integrity.

**Platform Features:**

* Job Posting & Applications
* Mobile Money & Bank Payments (Rwanda-focused)
* Offline Sync for low-connectivity areas
* Multi-Currency (RWF, KES, USD) & Localized Content

### **1.2 Resource Modeling**

#### **Core Resources**

**Users**

* id (UUID)
* phone\_number (string, e.g., +250...) *(Required for SMS auth)*
* role (enum: worker, client, admin)
* locale (string: en\_RW, fr\_RW)

**Jobs**

* budget (integer in *cents*) → 1000 RWF = 100000
* payment\_methods (array): ["mobile\_money", "bank\_transfer"]
* location (object):



**Payment**

* payment\_method (enum: mobile\_money, bank\_transfer)
* mobile\_money\_provider (string): "MTN Mobile Money," "Airtel Money"
* bank\_details (object): { "account\_number": "123456", "bank\_code": "BK001" }
* escrow\_enabled (boolean): *(Holds funds until job completion)*

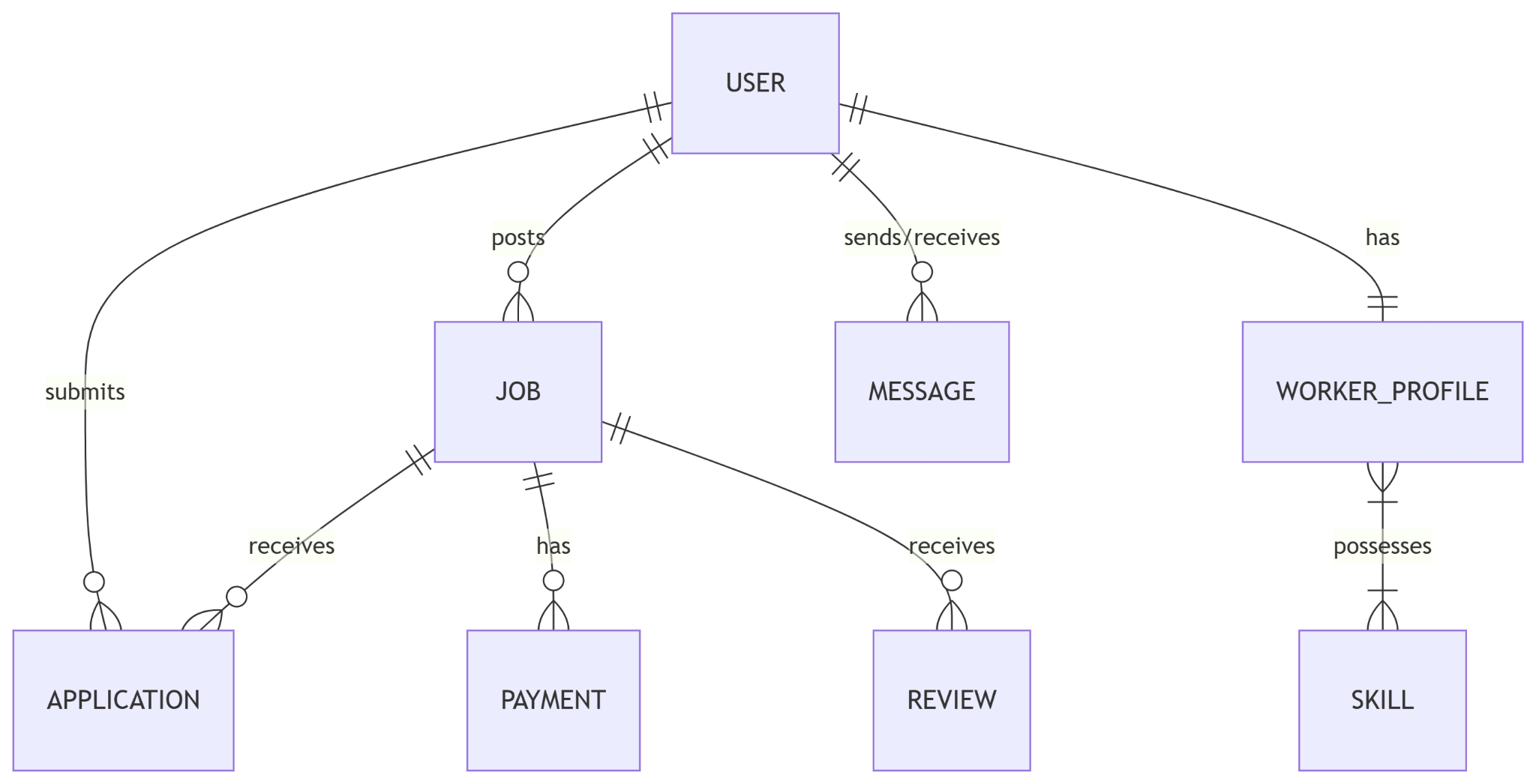
#### **Core Platform Actions**

* Clients can post jobs
* Workers can browse & apply
* Post-job reviews by clients & workers
* Profile & application management
* Messaging for negotiation or clarification

### **1.3 Entity-Relationship Diagram (ERD)**

**Resource Relationships**

* **User → JobPostings**: One-to-many
* **User → WorkerProfile**: One-to-one
* **JobPosting → Applications**: One-to-many
* **Application → WorkerProfile**: Many-to-one
* **WorkerProfile ↔ Skills**: Many-to-many
* **User → Messages**: One-to-many (sent/received)
* **JobPosting → Reviews**: One-to-many
* **JobPosting → Disputes**: One-to-many



## **2. Endpoint Design**

**Base URL:** /api/v1/  
**Conventions:** RESTful, plural nouns, kebab-case, nested as needed

### **Key Endpoints**

| **Resource** | **Endpoint** | **Method** | **Description** |
| --- | --- | --- | --- |
| Jobs | /api/v1/jobs?location=kigali | GET | Filter by location |
| Jobs | /api/v1/jobs | POST | Create job (client-only) |
| Payment | /api/v1/jobs/{id}/payments | POST | Pay via mobile money/bank |
| Users | /auth/otp/send | POST | Send SMS OTP |

### **Query Parameters**

* **Pagination**: /jobs?page=1&limit=20
* **Geo-Search**: /jobs?lat=-1.9403&lng=30.0644&radius=10km
* **Offline Sync**: /jobs/offline?last\_sync=<timestamp>

### **Additional Resources**

**Messages**: Enables real-time communication to clarify jobs. Crucial for trust-building.  
**Payments**: Secure transactions through mobile money or banks. Ensures accountability.  
**Skills**: Used for accurate worker matching (e.g., “Electrician”, “Solar Experience”).

## **3. API Documentation (OpenAPI Snippet)**

*This section contains the OpenAPI YAML or JSON specification snippet (to be included separately).*

## **4. Security & Compliance**

### **Authentication**

**SMS OTP Flow:**

* POST /auth/otp/send → Sends OTP (6-digit) to phone\_number
* POST /auth/otp/verify → Returns JWT token (valid 24h)

### **Authorization**

* **Workers**: Can GET /jobs, POST /applications
* **Clients**: Can POST /jobs, DELETE /jobs/{id}
* **Admin:** Can PATCH / profile, DELETE/ ban user {id}/verify

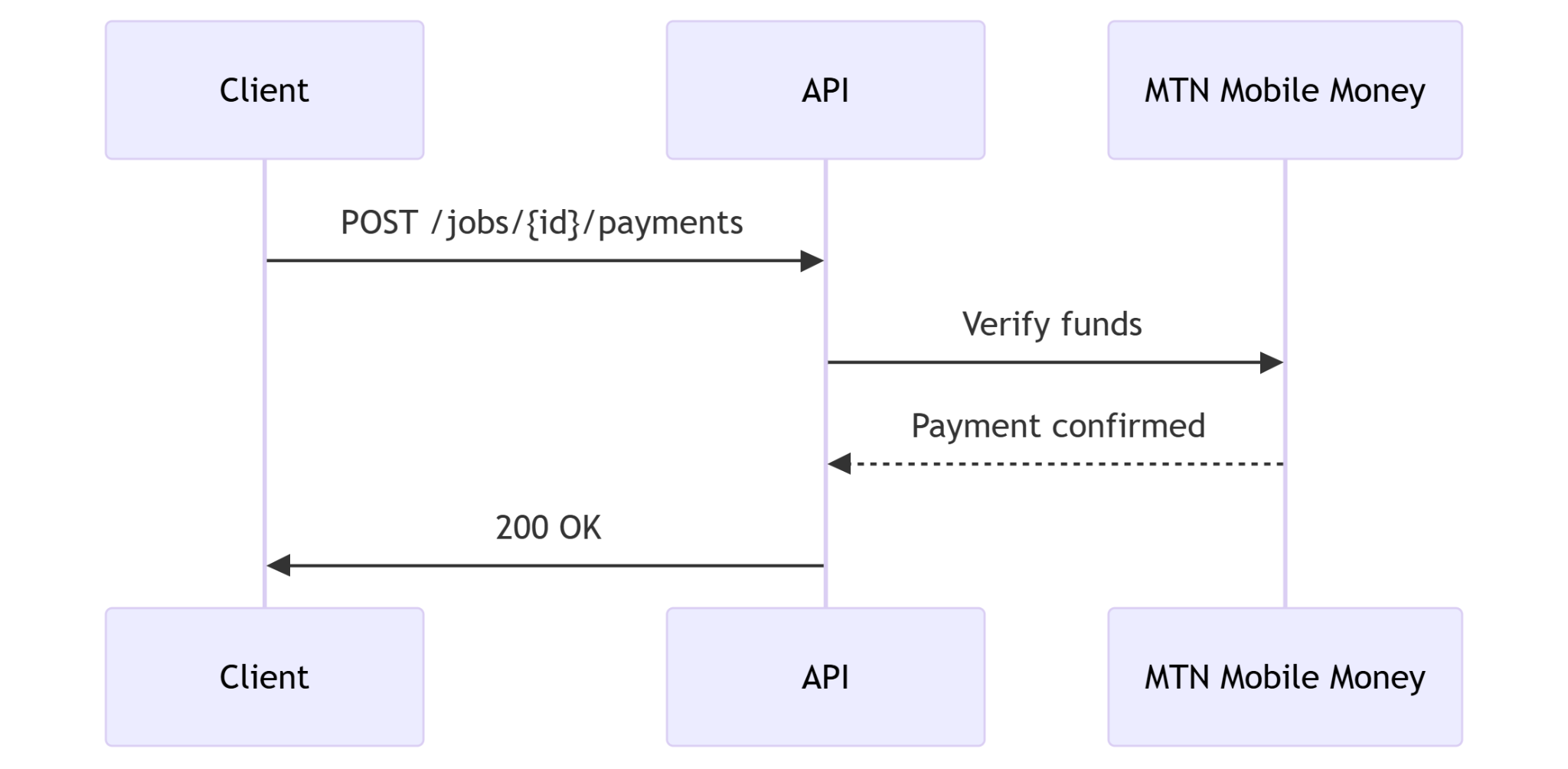
### **4.1 Market-Specific Adaptations**

**Rwanda-Focused Features**

* Mobile Money: MTN & Airtel supported
* Currency: All amounts in **cents** (e.g., 100000 = 1000 RWF)
* Localization:
  + Dates: ISO 8601 in Rwanda timezone (e.g., 2025-05-23T12:00:00+02:00)
  + Language: via Accept-Language: fr\_RW

### **Diagrams**

* **Payment Flow Diagram**



### **Job Lifecycle States**

[draft] → [posted] → [in\_progress] → [completed] → [paid]

### 

### **Error Handling**

**Standard Format:**

{

"error": {

"code": "RESOURCE\_NOT\_FOUND",

"message": "The requested resource does not exist.",

"status": 404

}

}



**Common Error Codes**

* 400: VALIDATION\_ERROR
* 401: UNAUTHORIZED
* 403: FORBIDDEN
* 404: RESOURCE\_NOT\_FOUND
* 409: CONFLICT
* 500: INTERNAL\_SERVER\_ERROR