Low‑Level Technical Specification: Database & Storage

### **1. PostgreSQL Schema**

* **Technology**: PostgreSQL v15+
* **Connection Pooling**: pgbouncer (min 5, max 50 connections)
* **Migrations**: Flyway (SQL‑based versioned scripts in db/migrations/)
* **Backups**:  
  + Daily pg\_dump to S3 (retention 30 days)
  + PITR via WAL archiving
* **Security**:  
  + Dedicated DB roles: app\_readwrite, app\_readonly, db\_admin
  + TLS connections enforced
  + Row‑level security for multi‑tenant data if needed

### **2. Tables & Columns**

* **users**
  + user\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + username TEXT UNIQUE NOT NULL
  + email TEXT UNIQUE NOT NULL
  + role TEXT NOT NULL CHECK(role IN ('Developer','ScrumMaster','ProductOwner','Observer'))
  + created\_at TIMESTAMPTZ NOT NULL DEFAULT now()
  + **Indexes**: (email), (username)
* **sessions**
  + session\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + title TEXT NOT NULL
  + mode TEXT NOT NULL CHECK(mode IN ('realtime','async'))
  + deadline TIMESTAMPTZ
  + created\_by UUID NOT NULL REFERENCES users(user\_id) ON DELETE CASCADE
  + created\_at TIMESTAMPTZ NOT NULL DEFAULT now()
  + **Indexes**: (created\_by), (mode)
* **stories**
  + story\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + session\_id UUID NOT NULL REFERENCES sessions(session\_id) ON DELETE CASCADE
  + title TEXT NOT NULL
  + description TEXT
  + created\_at TIMESTAMPTZ NOT NULL DEFAULT now()
  + **Indexes**: (session\_id)
* **votes**
  + vote\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + story\_id UUID NOT NULL REFERENCES stories(story\_id) ON DELETE CASCADE
  + user\_id UUID NOT NULL REFERENCES users(user\_id) ON DELETE CASCADE
  + vote\_value INT NOT NULL
  + submitted\_at TIMESTAMPTZ NOT NULL DEFAULT now()
  + **Constraints**: UNIQUE(story\_id,user\_id) to prevent duplicates
  + **Indexes**: (story\_id), (user\_id)
* **ai\_suggestions**
  + suggestion\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + story\_id UUID NOT NULL REFERENCES stories(story\_id) ON DELETE CASCADE
  + suggestion\_value INT NOT NULL
  + confidence REAL NOT NULL CHECK(confidence BETWEEN 0 AND 1)
  + model\_version TEXT NOT NULL
  + generated\_at TIMESTAMPTZ NOT NULL DEFAULT now()
  + **Indexes**: (story\_id)
* **comments**
  + comment\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + session\_id UUID NOT NULL REFERENCES sessions(session\_id) ON DELETE CASCADE
  + story\_id UUID NOT NULL REFERENCES stories(story\_id) ON DELETE CASCADE
  + user\_id UUID NOT NULL REFERENCES users(user\_id) ON DELETE CASCADE
  + content TEXT NOT NULL
  + timestamp TIMESTAMPTZ NOT NULL DEFAULT now()
  + **Indexes**: (session\_id), (story\_id), (user\_id)
* **integration\_records**
  + integration\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + tool\_name TEXT NOT NULL
  + api\_endpoint TEXT NOT NULL
  + secret\_key TEXT NOT NULL
  + last\_sync TIMESTAMPTZ
  + status TEXT
  + **Indexes**: (tool\_name)
* **audit\_logs**
  + log\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid()
  + user\_id UUID REFERENCES users(user\_id)
  + action TEXT NOT NULL
  + timestamp TIMESTAMPTZ NOT NULL DEFAULT now()
  + metadata JSONB
  + **Retention**: purge entries > 1 year old via scheduled job
  + **Indexes**: (user\_id), (action), ((metadata->>'sessionId'))

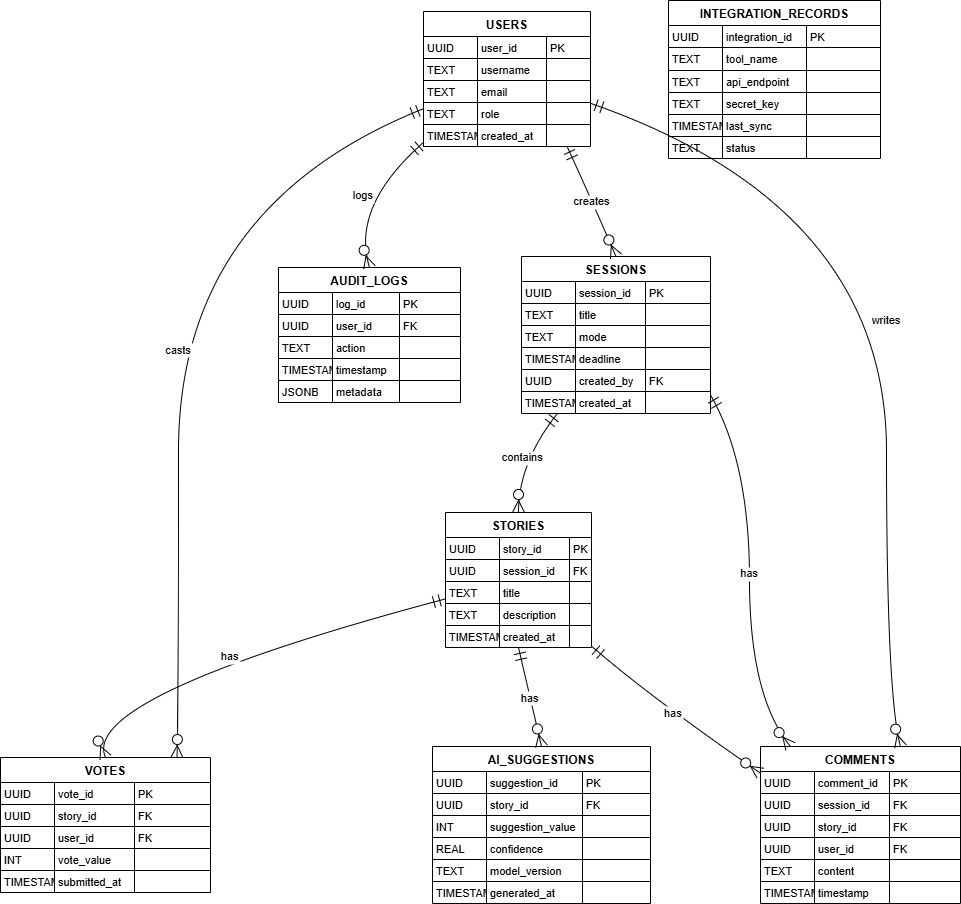
### **3. Indexing & Partitioning**

* **Foreign Key Indexes** on all REFERENCES columns for join performance.
* **Partial Index** for active sessions:  
    
   sql  
  CopyEdit  
  CREATE INDEX idx\_sessions\_active ON sessions(deadline)
* WHERE mode='async' AND deadline > now();
* **Partitioning Strategy** (optional):  
  + Partition votes by month if volume > 10 million rows/year.
  + Use PostgreSQL declarative partitioning on submitted\_at.

### **4. Redis Caching & Pub/Sub**

* **Technology**: Redis 7.x, single primary + replica
* **Cache Keys**
  + session:{sessionId}:state → JSON of current vote tallies (TTL 1 h)
  + session:{sessionId}:users → active participant list (TTL 1 h)
* **Pub/Sub Channels**
  + session:{sessionId}:events → broadcast real‑time events (vote, reveal)
* **Locking**
  + Use SETNX session:{sessionId}:lock (TTL 5 s) to coordinate summary generation

### **5. ERD Diagram (Mermaid.js)**



### **6. Data Retention & Archiving**

* **Audit Logs**: scheduled job deletes entries > 1 year
* **Session History**: optionally archive old sessions to a separate schema archive.sessions\_\*
* **Backups**: daily full dumps + hourly WAL for PITR

### **7. Disaster Recovery**

* **Point‑in‑Time Restore** from WAL
* **Cross‑Region/Cloud Replica** for high availability (optional)
* **DR Drills**: quarterly restore tests