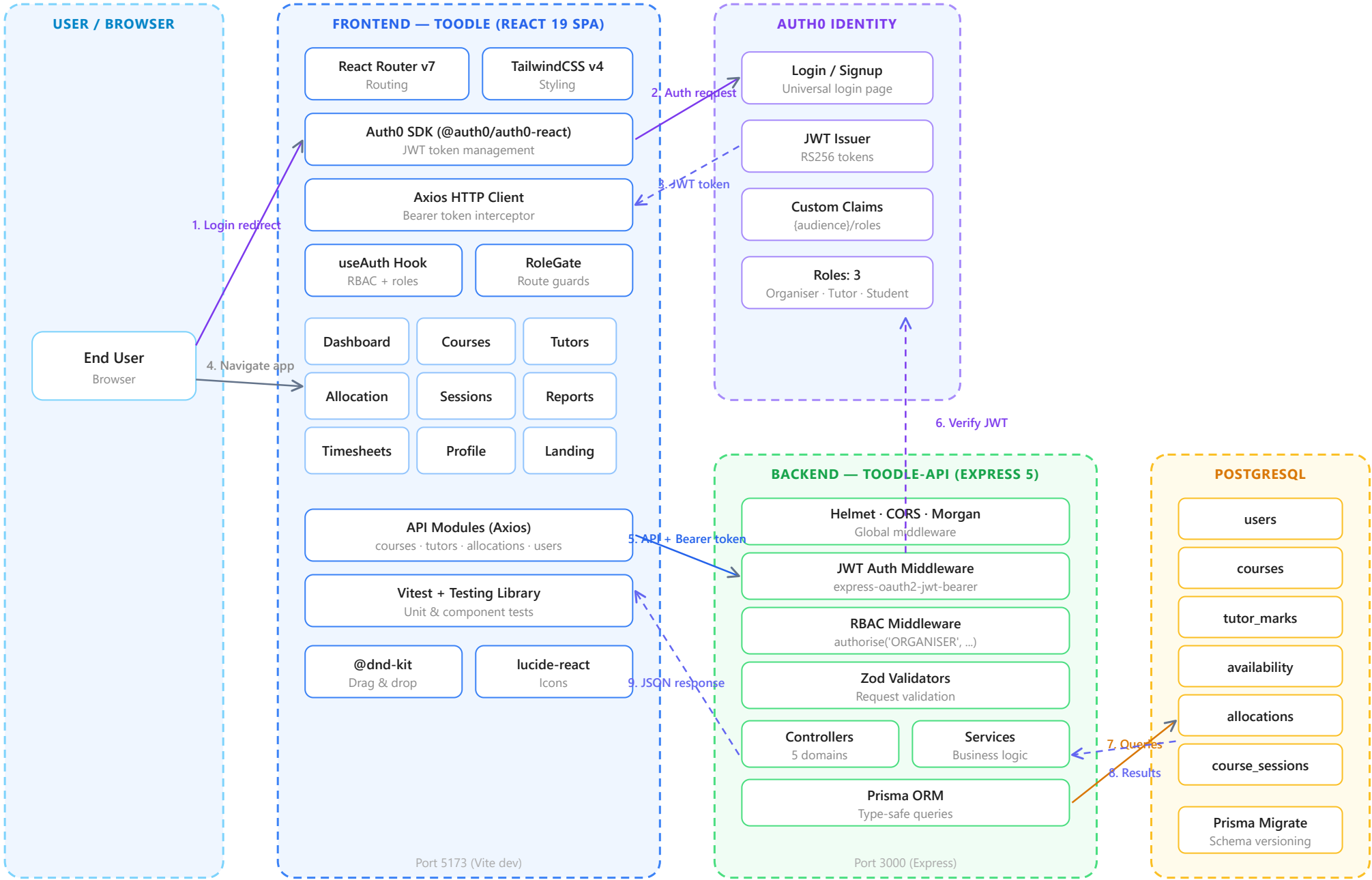


Toodle — System Architecture

Tutor Management System · React 19 + Express 5 + PostgreSQL · Auth0 RBAC



Frontend Stack

- React 19 + JSX
- Vite 6.2 build tool
- React Router v7
- Auth0 SDK (auth0-react)
- Axios (JWT interceptor)
- TailwindCSS v4
- @dnd-kit (drag & drop)
- lucide-react (icons)
- Vitest + Testing Library

Backend Stack

- Node.js + Express 5
- Prisma ORM 5.22
- PostgreSQL database
- express-oauth2-jwt-bearer
- Helmet + CORS + Morgan
- Zod validation
- RBAC middleware
- 5 domain controllers/services
- Vitest + Supertest

Auth & Security

- Auth0 universal login
- RS256 JWT tokens
- Custom claim: {audience}/roles
- 3 roles: Organiser, Tutor, Student
- Frontend: RoleGate + useAuth

Backend: authenticate + authorise

CORS: origin-restricted

Helmet security headers

Dev mode: mock role switcher

Request Flow

1. User authenticates via Auth0 → receives JWT with custom role claims.
2. React app stores token via Auth0 SDK; Axios interceptor attaches Bearer token to every API call.
3. Express backend validates JWT via express-oauth2-jwt-bearer, then RBAC middleware checks {audience}/roles claim.
4. Controller delegates to Service → Prisma executes type-safe queries against PostgreSQL.
5. JSON response flows back through the chain to the React frontend.