

CRM for Employee Productivity Platform — System Structure

Here's a clear **system structure** breaking down the CRM into layers, components, and flows:

1. Layered Architecture

1.1 Presentation Layer (Frontend) - Technology: React.js + TailwindCSS - Components: - Dashboard - Accounts & Contacts Management - Deals/Projects (Kanban, Detail View) - Tasks & Time Entries - Reports & Analytics - Integrations & Settings - Features: - Role-based navigation (Sales, Manager, Admin) - Responsive UI (desktop & mobile) - Charts (Recharts / Chart.js)

1.2 Application Layer (Backend API) - Technology: Django REST Framework / Node.js Express - Core Services: - Auth & RBAC (JWT/OAuth) - Accounts & Contacts - Deals/Projects - Tasks/Tickets - Time Entries (from Productivity Platform) - Reporting Engine - Integrations (webhooks, external APIs) - Async Processing: - Celery (Django) / BullMQ (Node.js) for background jobs (sync, reports)

1.3 Data Layer (Databases & Storage) - PostgreSQL → structured CRM data (accounts, contacts, deals, tasks, logs) - Redis → caching, session storage, queues - Object Storage (S3/GCP/Azure Blob) → file attachments, exports - Elasticsearch (optional, Phase 2) → advanced search and filtering

1.4 Integration Layer - Connects Productivity Monitoring backend to CRM - Responsibilities: - Map employees to CRM users - Ingest time entries via API/webhooks - Sync tasks/projects both ways - Push CRM events (task created, deal updated) back to productivity system

1.5 Security Layer - Role-based access control (Admin, Manager, Sales, Support) - Data encryption (AES for sensitive data, TLS for transit) - Audit logs for all CRUD operations

2. Module Breakdown

2.1 User & Role Management - Employees & internal CRM users - Roles (Admin, Sales, Manager) - Linked to Productivity system users

2.2 Accounts & Contacts - CRUD operations for companies and people - Attach deals, tasks, notes, and files

2.3 Deals & Projects - Sales pipeline stages (Prospecting → Won/Lost) - Link to Accounts and Tasks - Track associated time entries

2.4 Tasks & Tickets - Assignable to employees - Status & priority management - Linked to Deals/Projects

2.5 Time Entries - Direct ingestion from Productivity backend - Association with deals/tasks - Productivity scoring - Review queue for unmapped entries

2.6 Reports & Analytics - Sales pipeline summary - Time-per-project/client - Revenue per deal vs. hours spent - Export to CSV/PDF

2.7 Integrations - Productivity platform (main) - Optional billing, email, and project tools - Webhooks for real-time updates

2.8 Audit & Compliance - Change logs for all important entities - Configurable data retention - Export/delete requests (GDPR compliance)

3. Data Flow Overview

Flow A: Productivity → CRM 1. Agent tracks time & activity 2. Productivity backend sends time entry to CRM API 3. CRM backend validates & stores entry 4. Time entry mapped to Deal/Task → visible in CRM UI

Flow B: CRM → Productivity 1. Manager creates a task in CRM 2. CRM backend triggers webhook to Productivity backend 3. Task synced to employee dashboard in Productivity app

Flow C: Reporting 1. CRM aggregates deals, tasks, time entries 2. Reporting engine processes & caches results 3. UI displays charts and tables (pipeline, revenue, productivity)

4. Deployment Structure

- **Frontend** → Deployed on CDN (Netlify, Vercel, S3+CloudFront)
 - **Backend API** → Containerized (Docker), hosted on AWS ECS/Kubernetes/Azure App Service
 - **Database** → PostgreSQL (AWS RDS, Azure SQL)
 - **Cache & Queue** → Redis (AWS ElastiCache)
 - **Storage** → S3 bucket or equivalent
 - **Monitoring** → Prometheus + Grafana, Logs via ELK stack
-

5. Development Phases (CRM-specific)

- **MVP** → Accounts, Contacts, Deals, Tasks, Time entries ingestion, Dashboard, Basic Reports
 - **Phase 2** → Advanced reporting, Webhooks, Integrations, Audit logs
 - **Phase 3** → Predictive analytics, Billing, Multi-currency, Search improvements
-

This structure ensures the CRM is modular, scalable, and integrates seamlessly with your Employee Productivity Monitoring Software.