

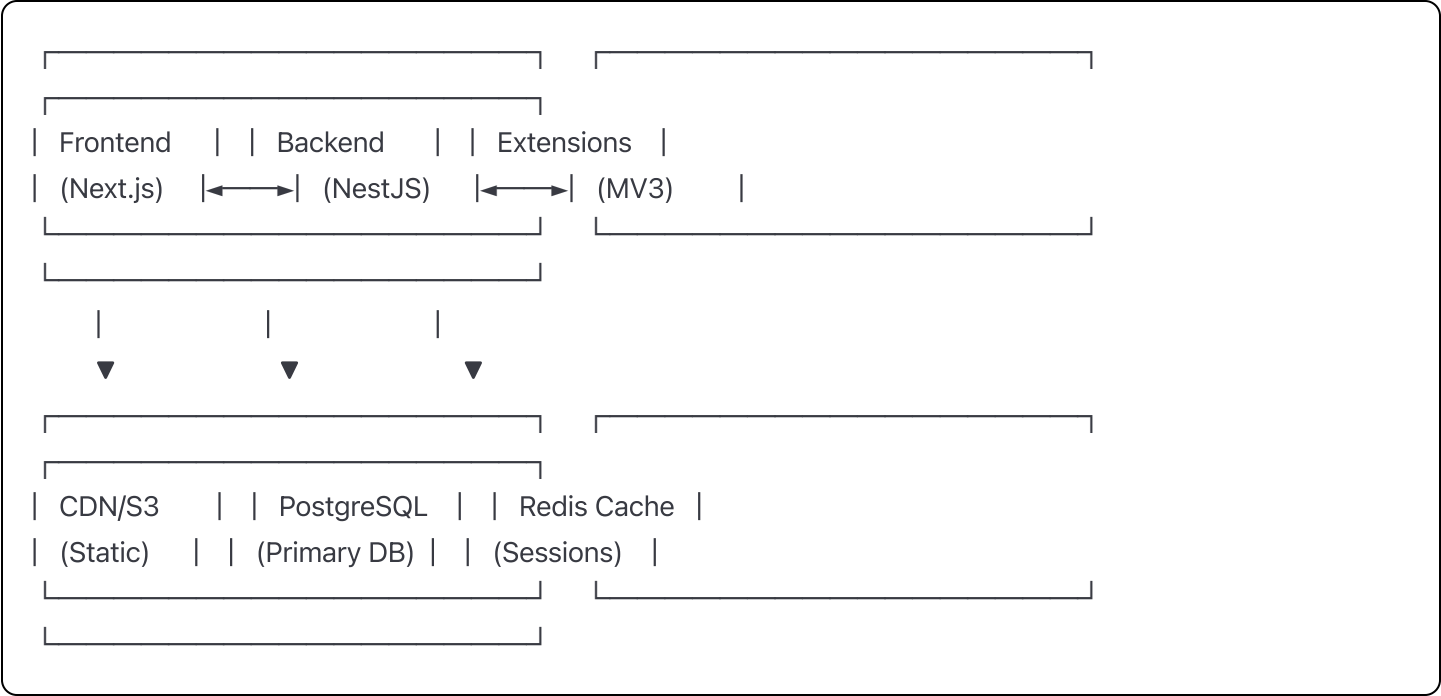
OverseeNOI Technical Specification

Executive Summary

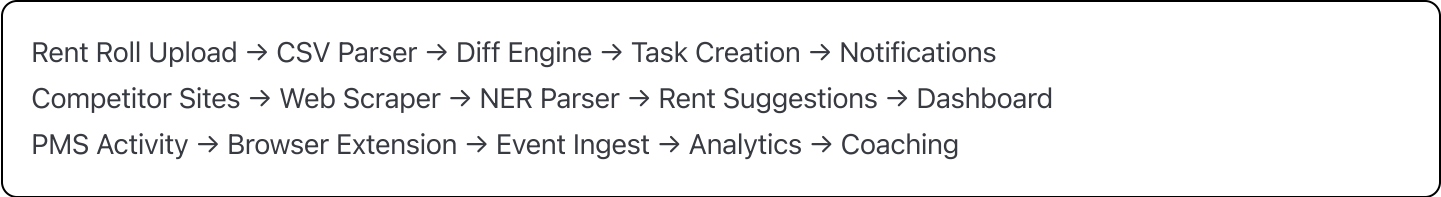
OverseeNOI is a next-generation SaaS platform bridging Asset Management and Property Management with real-time operational intelligence, communication, and oversight capabilities.

Architecture Overview

System Components



Data Flow Architecture



Core Modules

1. Communication & Task Management

- **Channel Hierarchy:** Portfolio → Property → Channel → Task Card → Thread
- **Real-time:** WebSocket connections for instant updates
- **Voice-to-Task:** Speech recognition with task auto-creation
- **File Attachments:** S3 storage with virus scanning

2. Rent Roll Intelligence

- **Import Engine:** Multi-format parser (CSV/XLS/PDF)
- **Diff Algorithm:** Day-over-day comparison with anomaly detection
- **Auto-Task Creation:** Delinquency alerts, variance investigation
- **Trend Analysis:** Historical patterns and predictions

3. Competitor Intelligence

- **Web Scraping:** Configurable selector packs per site
- **NER Calculation:** Net Effective Rent with concession analysis
- **Market Positioning:** \$/SF analysis and rent recommendations
- **Automated Alerts:** Price change notifications

4. Activity Monitoring ("Binoculars")

- **Browser Extension:** MV3 compliant with minimal data capture
- **Workflow Analytics:** Time tracking and efficiency metrics
- **Private Coaching:** Individual performance insights
- **Manager Dashboard:** Aggregate team performance

5. AI Assistant

- **Context Awareness:** Property data, conversations, and tasks
- **Image Recognition:** Property condition assessment
- **Document Analysis:** Lease audit and compliance checking
- **Predictive Insights:** Budget alerts and maintenance scheduling

Data Models

Core Entities

typescript

```
interface Company {  
  id: string;  
  name: string;  
  settings: CompanySettings;  
  created_at: Date;  
}
```

```
interface Portfolio {  
  id: string;  
  company_id: string;  
  name: string;  
  properties: Property[];  
}
```

```
interface Property {  
  id: string;  
  portfolio_id: string;  
  name: string;  
  address: string;  
  unit_count: number;  
  property_type: string;  
}
```

```
interface User {  
  id: string;  
  email: string;  
  display_name: string;  
  company_id: string;  
  roles: UserRole[];  
}
```

```
interface Channel {  
  id: string;  
  property_id: string;  
  key: string; // 'leasing', 'maintenance', 'ar', etc.  
  name: string;  
  visibility_roles: string[];  
  messages: Message[];  
  tasks: Task[];  
}
```

```
interface Task {  
  id: string;
```

```
channel_id: string;
title: string;
description: string;
assignee_id: string;
status: TaskStatus;
priority: Priority;
due_at: Date;
sla_at: Date;
tags: string[];
created_by: string;
created_at: Date;
thread: Message[];
}
```

```
interface RentRollSnapshot {
  id: string;
  property_id: string;
  date: Date;
  units: Unit[];
  aggregates: RentRollAggregates;
}
```

```
interface Unit {
  unit_id: string;
  unit_label: string;
  building?: string;
  floorplan: string;
  bedrooms: number;
  bathrooms: number;
  sqft: number;
  tenant_id_hash?: string;
  tenant_name_masked?: string;
  lease_start?: Date;
  lease_end?: Date;
  move_in?: Date;
  move_out?: Date;
  market_rent: number;
  actual_rent: number;
  other_charges: number;
  concessions: number;
  payments_mtd: number;
  balance: number;
  status: UnitStatus;
  delinquency_bucket: DelinquencyBucket;
```

```
notes?: string;  
}
```

RBAC Schema

typescript

```
interface UserRole {  
  user_id: string;  
  property_id?: string; // null = company-wide  
  role: Role;  
  scope: AccessScope;  
}  
  
enum Role {  
  VP = 'vp',  
  DIRECTOR = 'director',  
  ASSET_MANAGER = 'asset_manager',  
  SENIOR_ANALYST = 'senior_analyst',  
  CAPEX_PM = 'capex_pm',  
  REGIONAL_PM = 'regional_pm',  
  PROPERTY_MANAGER = 'property_manager',  
  ASSISTANT_PM_AR = 'assistant_pm_ar',  
  LEASING_MANAGER = 'leasing_manager',  
  LEASING_AGENT = 'leasing_agent',  
  MAINTENANCE_SUPER = 'maintenance_super',  
  MAINTENANCE_TECH = 'maintenance_tech',  
  VENDOR = 'vendor'  
}  
  
interface AccessScope {  
  properties: string[];  
  channels: string[];  
  permissions: Permission[];  
}
```

API Specifications

GraphQL Schema

graphql

```

type Query {
  channels(propertyId: ID!): [Channel!]!
  tasks(channelId: ID!, filters: TaskFilters): [Task!]!
  rentRollDiff(propertyId: ID!, date: Date!): RentRollDiff
  compSnapshot(propertyId: ID!, date: Date!): CompSnapshot
  properties: [Property!]!
  activityAnalytics(propertyId: ID!, timeRange: TimeRange!): ActivityAnalytics
}

type Mutation {
  createTask(input: CreateTaskInput!): Task!
  updateTask(id: ID!, input: UpdateTaskInput!): Task!
  completeTask(id: ID!): Task!
  sendMessage(channelId: ID!, content: String!, attachments: [String!]): Message!
  uploadRentRoll(propertyId: ID!, file: Upload!): RentRollSnapshot!
}

type Subscription {
  channelUpdates(channelId: ID!): ChannelUpdate!
  taskUpdates(taskId: ID!): TaskUpdate!
  notifications: Notification!
}

```

REST Endpoints

```

typescript

// Ingestion APIs
POST /api/ingest/rentroll
POST /api/ingest/activity
POST /api/ingest/comps

// File Management
POST /api/files/upload
GET /api/files/:id

// Webhook endpoints
POST /api/webhooks/accounting
POST /api/webhooks/pms-sync

```

Security & Compliance

Authentication & Authorization

- **SSO Integration:** SAML 2.0, OIDC support
- **MFA:** TOTP, WebAuthn support
- **Passwordless:** Magic links, biometric authentication
- **Row-Level Security:** PostgreSQL RLS implementation

Data Protection

- **Encryption:** AES-256 at rest, TLS 1.3 in transit
- **PII Handling:** Field-level encryption, masking by default
- **Data Retention:** Configurable per-tenant policies
- **Audit Logging:** Immutable, tamper-evident logs

Browser Extension Privacy

- **Opt-in Consent:** Clear user consent flow
- **Data Minimization:** Only whitelisted PII fields
- **Local Processing:** Client-side hashing and filtering
- **Admin Controls:** Company-wide kill switches

Implementation Roadmap

Phase 1: Core MVP (8-12 weeks)

1. **Authentication & RBAC** (2 weeks)
2. **Channel & Task System** (3 weeks)
3. **Rent Roll Diff Engine** (2 weeks)
4. **Basic Notifications** (1 week)
5. **File Upload & Management** (2 weeks)

Phase 2: Intelligence Layer (6-8 weeks)

1. **Competitor Scraping** (3 weeks)
2. **Browser Extension** (3 weeks)
3. **AI Assistant Foundation** (2 weeks)

Phase 3: Advanced Features (8-10 weeks)

1. **Advanced Analytics** (3 weeks)

2. **Vendor Management** (2 weeks)
3. **AP Automation** (3 weeks)
4. **Mobile Applications** (4 weeks)

Technology Stack

Frontend

- **Framework:** Next.js 14 with App Router
- **Language:** TypeScript 5.0+
- **Styling:** Tailwind CSS 3.0
- **State Management:** Zustand + React Query
- **Real-time:** Socket.io client
- **Testing:** Jest + Playwright

Backend

- **Framework:** NestJS with Fastify
- **Language:** TypeScript 5.0+
- **Database:** PostgreSQL 15+ with Prisma ORM
- **Cache:** Redis 7.0
- **Search:** OpenSearch
- **Queue:** BullMQ with Redis
- **Testing:** Jest + Supertest

Infrastructure

- **Cloud:** AWS (multi-region)
- **Compute:** ECS Fargate
- **Database:** RDS PostgreSQL
- **Storage:** S3 with CloudFront
- **Monitoring:** OpenTelemetry + DataDog
- **CI/CD:** GitHub Actions + Terraform

Browser Extension

- **Manifest:** v3

- **Language:** TypeScript
- **Build:** Webpack 5
- **Storage:** chrome.storage.local
- **Permissions:** Minimal scope

Performance Targets

Response Times

- **GraphQL Queries:** p95 < 200ms
- **REST APIs:** p95 < 150ms
- **Real-time Updates:** < 100ms latency
- **File Uploads:** 10MB in < 30s

Scalability

- **Concurrent Users:** 10,000+
- **Properties:** 100,000+
- **Tasks:** 1M+ active
- **Messages:** 10M+ daily

Availability

- **Uptime:** 99.9% SLA
- **RTO:** < 4 hours
- **RPO:** < 15 minutes
- **Multi-region:** Active-passive

Success Metrics

User Engagement

- **Daily Active Users:** 80%+ of licenses
- **Task Completion Rate:** 95%+
- **Response Time:** < 2 hours average
- **User Satisfaction:** NPS > 50

Business Impact

- **Operational Efficiency:** 30% faster task resolution

- **Communication Quality:** 50% reduction in email volume
- **Data Accuracy:** 99%+ rent roll accuracy
- **Cost Savings:** 20% reduction in operational overhead

Deployment Architecture

Development Environment

```
yaml
# docker-compose.yml
services:
  app:
    build: ./frontend
    ports: ["3000:3000"]
  api:
    build: ./backend
    ports: ["4000:4000"]
  db:
    image: postgres:15
    ports: ["5432:5432"]
  redis:
    image: redis:7
    ports: ["6379:6379"]
```

Production Environment

```
hcl
```

```
# Terraform AWS Infrastructure
```

```
module "vpc" {  
  source = "./modules/vpc"  
  cidr   = "10.0.0.0/16"  
}  
  
module "ecs" {  
  source = "./modules/ecs"  
  vpc_id = module.vpc.vpc_id  
}  
  
module "rds" {  
  source = "./modules/rds"  
  vpc_id = module.vpc.vpc_id  
}
```

This specification provides the foundation for building **OverseeNOI** as a production-ready, scalable SaaS platform. Each component is designed with enterprise-grade requirements in mind while maintaining the simplicity and user experience that will drive adoption.