

# Table of Contents

- 1. Overview: Building SaaS with Quantum ..... 1
  - 1.1. What Quantum Solves ..... 1
  - 1.2. Core Building Blocks ..... 1
  - 1.3. When to Use Quantum ..... 1
  - 1.4. Next Steps ..... 1

# 1. Overview: Building SaaS with Quantum

Quantum accelerates building multi-tenant SaaS platforms by providing secure-by-default foundations that scale from prototype to production.

## 1.1. What Quantum Solves

**Multi-tenant isolation with controlled sharing** - Each tenant's data is isolated by default - Selective sharing via policy rules (e.g., shared catalogs, partner directories) - Flexible tenancy models: one-database-per-tenant or shared-database-with-filtering

**Consistent, secure APIs** - Single BaseResource pattern for all CRUD operations - Automatic DataDomain filtering ensures users see only authorized data - Built-in query language works across all endpoints

**Policy-driven authorization** - Human-readable rules define "who can do what" - Rules contribute data filters (not just allow/deny) - Changes take effect without code deployment

**Developer productivity** - Minimal boilerplate: extend BaseModel and BaseResource - Live reload in development mode - Automatic OpenAPI documentation - Built-in CSV import/export for all entities

## 1.2. Core Building Blocks

**Models:** Extend BaseModel for automatic multi-tenancy, validation, and audit trails

**Repositories:** Extend MorphiaRepo for MongoDB operations with security filtering

**Resources:** Extend BaseResource for consistent REST APIs with minimal code

**Security:** Pluggable authentication (JWT, OIDC) with declarative permission rules

**Query Language:** Single syntax for filtering across APIs, permissions, and reports

## 1.3. When to Use Quantum

□ **Good fit:** - Multi-tenant B2B SaaS platforms - Applications requiring flexible data sharing between organizations - Teams wanting consistent REST APIs without boilerplate - Projects needing audit trails and compliance features

□ **Consider alternatives:** - Single-tenant applications - High-frequency trading or real-time systems - Applications with complex, non-relational data models - Teams preferring GraphQL over REST

## 1.4. Next Steps

- **New to Quantum?** Start with [Getting Started](#)
- **Learning concepts?** Read [Multi-Tenancy Models](#)
- **See it in action?** Try the [Supply Chain Tutorial](#)