

Below is a **detailed, production-grade Application Architecture** for the same **healthcare multi-product initiative**, written at **Principal / Lead Architect depth**.

This shows *exactly how individual systems are designed, built, secured, deployed, and operated* inside your Solution Architecture.

---

## Application Architecture (Detailed)

**Scope:** Individual application or microservice within a product line

**Examples:** Televisit Service, Care Plan Service, Claims Intake Service, Provider Directory Service

---

### 1 Application Architecture Goals

- Ensure **safety, reliability, and compliance**
  - Enable **independent deployment**
  - Minimize blast radius
  - Enforce **platform standards**
  - Enable **observability and operability**
  - Support **audits and regulatory evidence**
- 

### 2 Application Context (C4 – Level 1)

Each application lives inside:

User / System

↓

API Gateway (platform)

↓

Application Service

↓

Platform Services

↓

Integration Facade

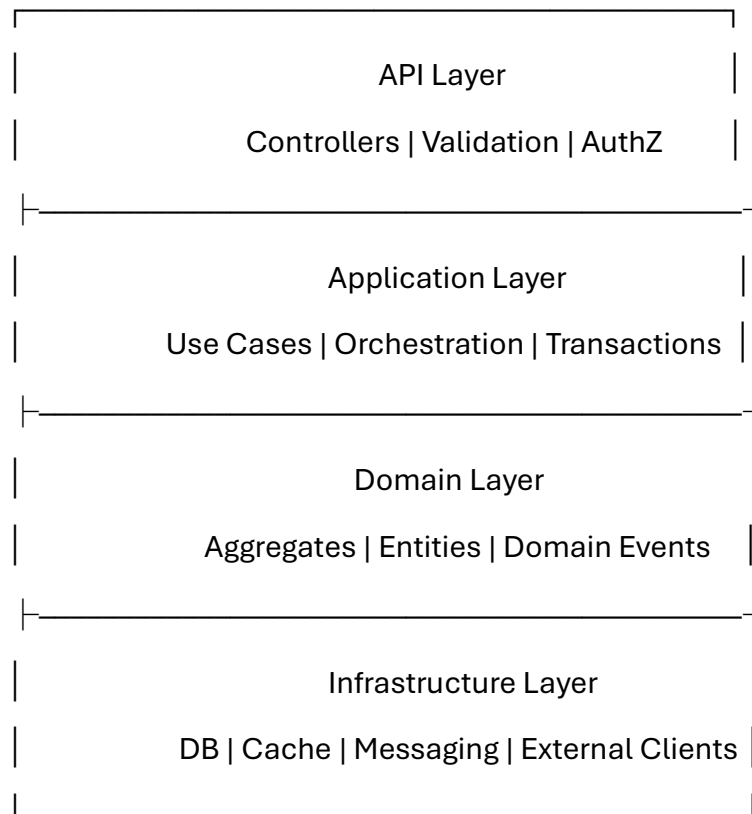
↓

**Rule:**

👉 No application talks directly to EHR or identity systems

---

**3 Internal Application Structure (C4 – Level 3)**



**Pattern:** Clean Architecture / Hexagonal Architecture

---

## Domain Design (DDD)

Each application:

- Owns **one bounded context**
- Has **its own data store**
- Emits domain events
- Does not share databases

Example:

CarePlanContext

- CarePlanAggregate
  - Goal
  - Intervention
  - DomainEvents
- 

## API Architecture

### REST / GraphQL

- External access only
- Versioned
- OpenAPI mandatory
- Idempotent writes
- Pagination enforced

### Event APIs

- AsyncAPI documented
- Schema registry enforced
- Backward compatible only

### Internal APIs

- gRPC allowed for internal calls
- No tight coupling

---

## 6 Data Architecture

### Datastores

- One DB per service
- PHI encrypted
- Logical & physical separation

### Patterns

- CQRS for high read/write split
- Event sourcing (where appropriate)
- Read replicas for analytics

### Retention

- Policy-driven retention
- Right-to-forget enforcement
- Immutable audit logs

---

## 7 Security Architecture (HIPAA-grade)

Area	Implementation
------	----------------

AuthN	Platform IAM (OIDC/OAuth2)
-------	----------------------------

AuthZ	RBAC + ABAC
-------	-------------

Secrets	Vault / KMS
---------	-------------

Encryption	TLS 1.2+, AES-256
------------	-------------------

Audit	Immutable audit logs
-------	----------------------

Logging	PHI redaction
---------	---------------

Access	Zero trust
--------	------------

Testing	SAST/DAST
---------	-----------

### Rule:

👉 If it handles PHI, it must produce an audit trail

---

## 8 Resilience & Reliability

### Patterns

- Circuit breaker
- Retry with backoff
- Bulkheads
- Timeout budgets
- Graceful degradation
- Feature flags

### Healthcare requirement

Manual fallback must exist for critical workflows

---

## 9 Observability

- Structured logging (trace ID)
  - Metrics (RED + USE)
  - Distributed tracing
  - Health probes
  - Synthetic monitoring
- 

## 10 Deployment Architecture

### Environments

Dev → Test → UAT → Prod

### Infrastructure

- Kubernetes / ECS
- Blue-green or canary
- IaC (Terraform)
- Immutable images

- Multi-AZ deployment

## **DR**

- Cross-region backup
  - Automated restore tests
- 

## **1 1 CI/CD Architecture**

Pipeline includes:

- Build
- Unit test
- Contract test
- Security scan
- Image scan
- IaC validation
- Deploy
- Post-deploy verification

**No pipeline = no production**

---

## **1 2 Configuration Management**

- Externalized config
  - Feature flags
  - Environment isolation
  - No secrets in code
-

### **1 3 Integration with Platform Services**

Mandatory integrations:

- Identity
- Consent
- Audit
- Event bus
- API gateway
- Notification
- Workflow

Applications cannot bypass platform services

---

### **1 4 Operational Runbooks**

Each app must have:

- Startup & shutdown
  - Incident response
  - Data correction
  - Rollback
  - DR recovery
  - Contact matrix
- 

### **1 5 Governance & Compliance Evidence**

Each app must produce:

- Architecture diagram
- ADRs
- Data flow diagram
- Threat model

- Test evidence
- Release checklist
- Audit logs

---

## **1 6 Common Failure Modes (Healthcare-specific)**

Failure	Prevention
PHI leakage	DLP + audit
Incorrect patient match	Platform identity
Schema breakage	Contract testing
Outage impact	Graceful degradation
Vendor dependency	Façade pattern
Overcoupling	No shared DBs

---

## **1 7 Why this Application Architecture Works**

- Supports **safe independent deployments**
- Enforces **compliance by default**
- Reduces blast radius
- Enables audits
- Scales with product lines
- Aligns with enterprise & solution architecture

---

## **Interview-ready closing**

“My application architecture enforces clean boundaries, security, and observability by default, so teams can deploy independently without creating risk. In healthcare, this is the only way to scale safely.”

---