Below is a **complete, enterprise-grade definition of Standards, Governance, and Guardrails for Team Self-Service**, designed for **healthcare multi-product platforms** where **speed, safety, and compliance must coexist**.

This is exactly how high-maturity orgs let teams move fast **without asking permission** — while still passing audits.

Self-Service Enablement Model

Goal: Teams can provision, build, deploy, and operate safely without manual approvals
Rule: *If it's self-service, it must be standardized, governed, and guarded by automation*

1️⃣ Standards (What teams must follow)

Standards define the **approved way** to do things.
They are **documented, versioned, and enforced by tooling**.

1.1 Architecture Standards

Mandatory

- Enterprise & Solution Reference Architectures
- Approved integration patterns (API, event, batch, façade)
- No direct EHR or DB access
- Platform-first services (identity, audit, events, consent)
- ADR required for high-risk decisions

Self-service enablement

- Reference diagrams
- Templates
- Scaffolding generators

1.2 Technology Standards

Approved stacks only

- Cloud (AWS/Azure/GCP)
- Containers & Kubernetes
- API Gateway
- Streaming platform
- Lakehouse format (Delta/Iceberg)
- CI/CD tools
- Secrets manager
- Observability stack

Self-service enablement

- Golden images
- Terraform modules
- Helm charts
- Pipeline templates

1.3 Data Standards

- Data contracts (schema registry)
- Domain ownership
- Naming conventions
- Retention rules
- PHI classification
- FHIR alignment
- Certified metrics only

Self-service enablement

- Ingestion templates

- Data product templates
- DQ rule generators

1.4 Security Standards

- Zero Trust Architecture
- MFA required
- OIDC/OAuth2 only
- Encryption everywhere
- PHI tokenization
- Audit logging mandatory
- No hard-coded secrets
- JIT access for admins

Self-service enablement

- IAM role templates
- Vault integration
- Pre-built policies

2️⃣ Governance (Who decides what)

Governance is **decision clarity**, not meetings.

2.1 Decision Ownership Model

Decision Type	Owner	Mechanism
Enterprise standards	EA	ARB
Platform services	Platform team	Roadmap
Security policies	Security	Policy-as-code

Decision Type	Owner	Mechanism
Data ownership	Domain owner	Data contract
Exceptions	EA + Security	Time-boxed waiver

2.2 Governance Forums (Lightweight)

Forum	Purpose	Frequency
Architecture Working Group	Design alignment	Weekly
Platform Council	Shared priorities	Biweekly
ARB	Major decisions	Monthly
Security Review	High-risk items	On-demand
Exec Steering	Risk & investment	Quarterly

2.3 Decision Artifacts (Mandatory)

- ADRs
- Architecture diagrams
- Data contracts
- Threat models
- Runbooks

If it’s not documented, it didn’t happen.

3️⃣ Guardrails (How standards are enforced automatically)

Guardrails are **non-bypassable automated checks**.

3.1 Infrastructure Guardrails

Enforced via Terraform + OPA

- No public S3/Blob buckets
- Encryption required
- Approved regions only
- Approved instance types
- Network segmentation enforced
- Mandatory logging

3.2 Application Guardrails

Enforced via CI/CD + OPA

- Approved frameworks only
- OIDC required
- Secrets in vault
- No shared DBs
- Contract tests required
- SAST/DAST required
- Logging required

3.3 Data Guardrails

Enforced via pipelines

- Schema registered
- PHI classification required
- DQ checks pass
- Lineage captured
- Access policies enforced
- Retention applied

3.4 Security Guardrails

Enforced via Zero Trust + SIEM

- MFA required
- JIT access enforced
- Audit logs immutable
- DLP scans
- Continuous monitoring

3.5 AI / GenAI Guardrails

Enforced via AI gateway

- RAG only
- Approved models only
- Prompt versioning
- Output validation
- Human approval for clinical use
- Full audit trail

🔌 Self-Service Capabilities Provided to Teams

This is what teams can do **without tickets**:

Area	Self-Service Capability
Infra	Provision environments
CI/CD	Create pipelines
Identity	Request roles
Data	Create data products
APIs	Publish APIs

Area	Self-Service Capability
Events	Create topics
ML	Register models
GenAI	Use approved prompts
Secrets	Create secrets
Monitoring	Create dashboards

5️⃣ Exception Handling (When teams really need to break rules)

- Exception request via Jira
- Must include:
  - Business justification
  - Risk assessment
  - Mitigation
  - Expiry date
- Auto-expire after 90 days
- Must be reviewed quarterly
- Exceptions are tracked as technical debt

6️⃣ Metrics to Measure Self-Service Success

Metric	Target
% self-service provisioning	>90%
Time to provision	<30 mins
Architecture violations	↓
Security findings	0 critical

Metric	Target
Platform reuse	>70%
Shadow IT	0
Exception age	<90 days

7️⃣ Common Failure Modes & Fixes

Failure	Fix
Too many standards	Reduce & automate
Teams bypass platform	Make platform easier
Slow governance	Shift to policy-as-code
Manual approvals	Replace with templates
Compliance friction	Embed in pipelines

8️⃣ Operating Principle

Freedom inside guardrails, not freedom without responsibility

---

🎯 Interview-ready one-liner

“I enable team self-service by defining clear standards, automating guardrails with policy-as-code, and keeping governance lightweight. Teams move fast, audits pass automatically, and architects stop being bottlenecks.”
