1. TECHNICAL DEBT LANDSCAPE

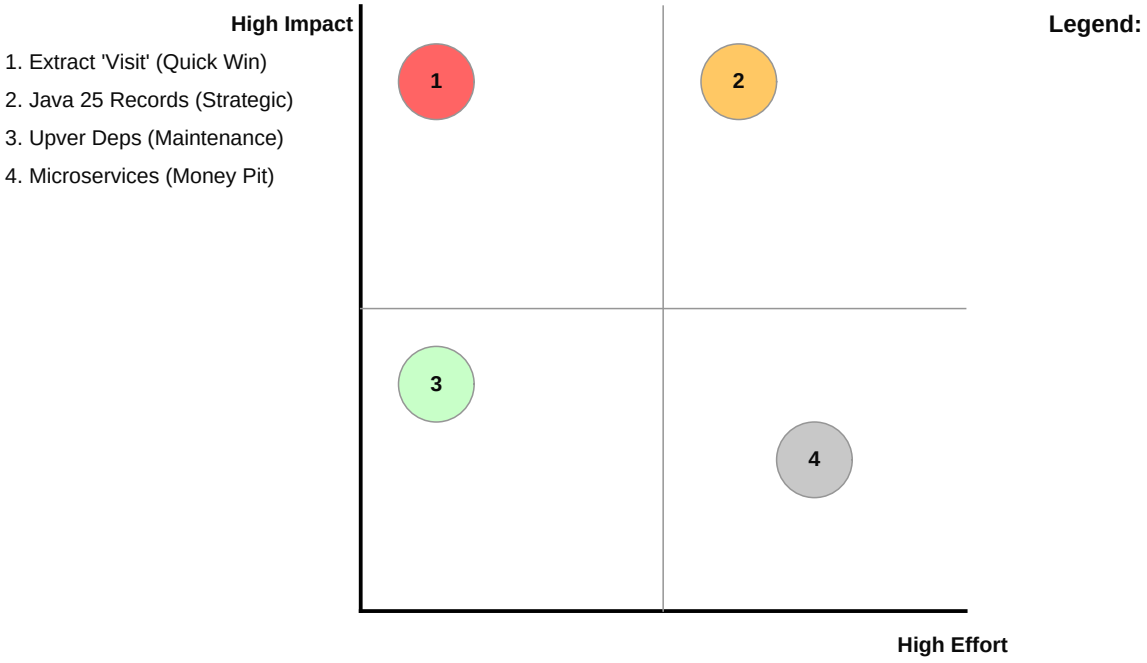
Report Date:	February 10, 2026	**Reviewers:**	P. Pineda & C. Martinez
Target State:	Java 25 Distributed System	**Risk Status:**	**CRITICAL (Blocking Scalability)**

1.1. THE RECOVERY STRATEGY

The Spring PetClinic repository, while functional, suffers from 'Architectural Entropy'. Our audit identified tightly coupled domains and an outdated concurrency model. This Backlog Recovery Plan prioritizes tasks using the **ICE Score** (Impact, Confidence, Ease) to enable a safe migration to Java 25.

Technical Debt Quadrant (Prioritization Matrix)

We mapped the identified issues based on Business Impact vs. Remediation Effort.



STAFF ENGINEER DECISION

We will focus on Quadrant 1 (Quick Wins) and Quadrant 2 (Strategic). We will NOT pursue full Microservices (Quadrant 4) immediately, as the overhead outweighs the benefit at this stage. Instead, we adopt a 'Modular Monolith' approach.

2. ARCHITECTURE DEBT: THE COUPLING CRISIS

The most critical item in the backlog is the domain coupling found in `org.springframework.samples.petclinic.owner`. Currently, `Visit` is treated as a sub-property of `Pet`, making it impossible to scale the Scheduling service independently.

2.1. Refactoring Backlog: Decoupling 'Visits'

Objective: Move `Visit` from a nested entity to a Root Aggregate.

ID	Task Description	Est. Effort
ARCH-01	Create package `org.springframework.samples.petclinic.visit`	1 SP
ARCH-02	Refactor `Visit` entity: Replace `Pet` object with `petId` (Integer)	3 SP
ARCH-03	Create `VisitRepository` independent of `OwnerRepository`	2 SP
ARCH-04	Update `VisitController` to use `VisitService` (New Interface)	5 SP

2.2. Database Schema Impact

The SQL schema must also be updated to reflect this loose coupling. We need to break the Foreign Key constraint rigidity.

```
-- BACKLOG ITEM: DB-05
ALTER TABLE visits DROP FOREIGN KEY fk_visits_pets;
ALTER TABLE visits ADD COLUMN pet_id INTEGER NOT NULL;
CREATE INDEX idx_visit_pet_id ON visits(pet_id);
-- Note: Logic integrity will be handled by the Service Layer, not DB constraints.
```

3. JAVA 25 MODERNIZATION BACKLOG

Once the architecture is stabilized, we will apply Java 25 features to reduce memory footprint and improve concurrency. This is the 'Strategic' phase of the recovery.

3.1. Implementation Tasks (Java 25 Preview)

Ticket	Priority	Task
J25-01	CRITICAL	Enable `--enable-preview` in Maven Compiler Plugin
J25-02	HIGH	Replace Lombok @Data DTOs with Java `records`
J25-03	HIGH	Enable Virtual Threads (`spring.threads.virtual.enabled=true`)
J25-04	MEDIUM	Refactor `PetType` Logic using Pattern Matching for Switch

3.2. Performance Impact Projection

Switching to Virtual Threads (Project Loom) will allow the Tomcat server to handle thousands of concurrent requests without blocking OS threads. This replaces the need for reactive programming (WebFlux) in this scenario.

4. INFRASTRUCTURE & QA RECOVERY

The backlog also addresses the 'Rot' in the CI/CD pipeline and testing strategy. Current tests are slow and rely on H2 exclusively. We need Docker parity.

4.1. DevOps Backlog

ID	Action Item	Priority
OPS-01	Containerize App using Cloud Native Buildpacks (Paketo)	High
OPS-02	Add OpenTelemetry Agent for Tracing	High
OPS-03	Create GitHub Actions Workflow for Java 25 Matrix	Medium
OPS-04	Integrate Snyk for Vulnerability Scanning	Medium

4.2. Testing Strategy Overhaul

We will introduce **Testcontainers** to replace H2 in integration tests. This ensures that tests run against the real production database engine (MySQL/Postgres).

```
@Testcontainers
class VisitControllerTests {
    @Container
    static MySQLContainer<?> mysql = new MySQLContainer<>("mysql:8.0");

    @DynamicPropertySource
    static void props(DynamicPropertyRegistry registry) {
        registry.add("spring.datasource.url", mysql::getJdbcUrl);
    }
}
```

5. EXECUTION ROADMAP & SIGN-OFF

This plan is designed to be executed in 4 Sprints (2 Weeks each). The Staff Engineering team (Pineda & Martinez) will oversee the architectural integrity.

5.1. Sprint Schedule

Timeline	Theme	Deliverables
Sprint 1	Decoupling	Extract `Visit` package. Break FK constraints.
Sprint 2	Modernization	Apply Java 25 Records to DTOs. Enable Virtual Threads.
Sprint 3	Resilience	Add OpenTelemetry. Implement Testcontainers.
Sprint 4	Cleanup	Remove legacy XML configs. Final Load Testing.

DEFINITION OF DONE (DoD)

1. Code compiles with JDK 25.
2. All tests pass in CI (GitHub Actions).
3. No circular dependencies in packages (ArchUnit verified).
4. Startup time is under 3.5 seconds.

PLAN APPROVAL

Pablo Pineda
Lead Staff Engineer
Approved for Execution

Christian Martinez
Lead Staff Engineer
Approved for Execution