Requirements Traceability Matrix (Sample Scenario)
This matrix ensures traceability from business goal to implementation of the new feature Pet Insurance for
the mobile app Lucky
.
BR — Business Requirement
JS — Job Story / User Requirement
FR — Functional Requirement
NFR — Non-Functional Requirement
ARCH— Architectural constraints
INT — Integration points
Artifacts — Supporting design & analysis artifacts

Traceability flow: BR → JS → FR → NFR → ARCH → INT → Artifacts

BR ID	Business Requirement	Job Story / User Requirement
BR-01	Implement Pet Insurance to quickly enter the growing pet insurance market and increase revenue	**JS-01:** As an owner of an active and curious pet, I want to purchase insurance for my pet so that in case of injury or illness I can quickly get professional help

FR ID	Functional Requirement
FR-01	User can view insurance details
FR-02	User can add insurance to the shopping cart
FR-03	User can pay for insurance

NFR ID	Non-Functional Requirement
NFR-01	Service response time ≤ 10 seconds
NFR-02	Throughput: ~5,000 interested users/day, ~100 purchases/day (~4–5 req/min)

NFR-03	Integration with Shopping Cart via Kafka
NFR-04	Support for internationalization (Russian + English)
NFR-05	Operations are secure and synchronous

ARCH ID	Architectural Constraint
ARCH-01	New Insurance microservice using MongoDB for storing insurance descriptions
ARCH-02	Access via API Gateway, Service Registry, Load Balancer
ARCH-03	REST-style API, JSON data format
ARCH-04	No new integrations with external systems (except existing)

INT ID	Integration / Notes
INT-01	No additional load on Shipping (insurance requires no delivery)
INT-02	CRM reads shopping cart via existing service
INT-03	External systems remain unchanged: PAY-PAY, SHIP-SHIP, CRM, Amazon S3

Artifacts
API Model
C4 Diagram
OpenAPI Specification (YAML)