

Request-to-Resolution Process Blueprint

A practical operating model for reliable Salesforce automation across Flow, Apex and Agentforce.



DEMOCO

Fictional B2B SaaS scenario

Illustrative architecture / synthetic data

One governed path. Clear ownership.

The blueprint aligns process design, automation boundaries and operational controls.

DESIGN OBJECTIVE

Predictable by design

Create one predictable path from request intake to resolution while keeping business policy configurable, complex routing testable and AI actions controlled.

01

Case

system of record

03

Layers

Flow / Apex / Agent

04

Types

request scenarios

EXPLICIT NON-GOALS

- No autonomous high-risk approval
- No production or client data
- No performance claims

STATE

Single source of truth

Case owns the request state, ownership and lifecycle.

ORCHESTRATE

Flow owns the process

Visible business steps stay modular and fault-aware.

DECIDE

Apex owns complex routing

A bulk-safe service returns results without updating Case.

ASSIST

Agent actions stay constrained

Guidance first; record writes require confirmation.

EVIDENCE

Decisions stay observable

Correlation IDs and audit events support review.

Where reliability breaks down

An illustrative scenario used to define the demonstrator - not a statement about a real company.

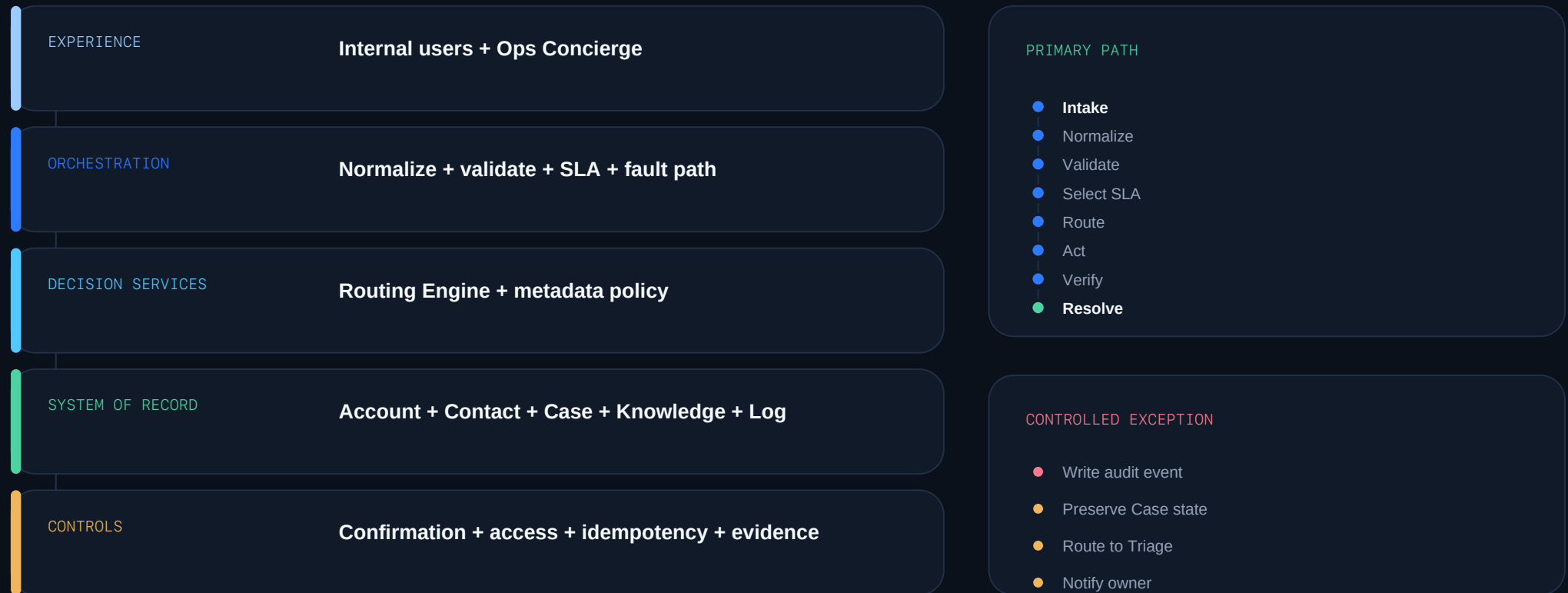


Friction map

STAGE	FRICTION	CONSEQUENCE	SIGNAL TO CAPTURE
Intake	Context varies by channel	Rework before routing	Missing field reason
Triage	Rules live in team memory	Inconsistent assignment	Routing decision
Ownership	Capacity is not explicit	Silent fallback	Queue and fallback code
Policy	Guidance is scattered	Different answers	Knowledge source used
Escalation	Follow-up is manual	Late or duplicate action	Due time and reason
Handoff	Context is rewritten	Detail loss	Handoff summary

A controlled Request-to-Resolution spine

Each layer has a clear job; exceptions use the same observable recovery path.



ARCHITECTURE PRINCIPLE

Automation must fail into a visible, owned state - never into silence.

Choose the tool by responsibility

The simplest tool is the one that keeps the decision testable, observable and owned.

DECISION	DEFAULT TOOL	WHY	BOUNDARY
Normalize Case fields	FLOW	Declarative and visible	Before-save only
Orchestrate lifecycle	FLOW	Readable business sequence	Keep decisions modular
Resolve routing	APEX	Deterministic and bulk-testable	Returns result; no Case DML
Store policy	METADATA	Deployable configuration	Validate missing config
Provide guidance	AGENTFORCE	Conversational access to knowledge	No record write
Create or escalate	AGENT + FLOW	Reuses governed automation	Confirmation required
High-risk judgment	HUMAN	Accountability and context	Agent must escalate
Ambiguous exception	DO NOT AUTO	Reliability value is unclear	Track frequency first

DESIGN RULE

Moving logic to Apex is not an upgrade by itself. Leaving complexity in one Flow is not simplicity.

Make failure safe and reviewable

Guardrails apply before, during and after an automated decision.



FAILURE MODE	SYSTEM RESPONSE	HUMAN VISIBILITY
Missing input	NEEDS_INPUT; no owner change	Missing field list
No active rule	FALLBACK to Triage	Fallback reason
Capacity reached	Next rule, then fallback	Decision code
Missing fallback	Controlled config error	Fault + owner alert
Resolved before SLA	Exit without escalation	Optional audit
High-risk intent	Stop autonomous path	Compliance handoff
Injection / data breach	Refuse action	Security audit event

Build the governed spine before the agent

Order reflects dependency and risk reduction, not measured return on investment.

NOW

Governed spine

DEPENDENCY STAGE 01

- 01 Define Case state and intake fields
- 02 Normalize, validate and correlate
- 03 Build Routing Engine contract
- 04 Add fault log and Triage fallback
- 05 Test exact match and batch paths

FOUNDATION

NEXT

Operational resilience

DEPENDENCY STAGE 02

- 01 Add scheduled SLA re-check
- 02 Define queue capacity policy
- 03 Publish synthetic Knowledge
- 04 Expose outcomes and fallbacks

RESILIENCE

LATER

Constrained agent pilot

DEPENDENCY STAGE 03

- 01 Release guidance-only topics
- 02 Add confirmed Case actions
- 03 Run adversarial prompt tests
- 04 Review audits before expansion

CONTROLLED SCALE

Evidence before expansion

Readiness is demonstrated through controls and tests - metrics are defined only after deployment.

READINESS GATES

- ✓ Process states and ownership documented
- ✓ Routing rules have an accountable owner
- ✓ Happy, fallback and fault paths tested
- ✓ Seed is synthetic and idempotent
- ✓ Agent writes require confirmation
- ✓ Logs exclude secrets and production PII
- ✓ Rollback and Triage paths exercised

METRICS TO ESTABLISH

- | | |
|-----|--|
| M01 | Intake completeness |
| M02 | Routing outcome distribution |
| M03 | Fallback / NEEDS_INPUT rate |
| M04 | Reassignment count |
| M05 | SLA escalation review |
| M06 | Agent confirmation / refusal / handoff |

Definitions only. Baselines and targets require a real, isolated environment.

RECOMMENDED VALIDATION SEQUENCE

Developer org

Automated tests

Business walkthrough

Constrained pilot

Control review

Staged expansion

Salesforce, Agentforce and Apex are trademarks of Salesforce, Inc. This sample is independent and is not endorsed by or affiliated with Salesforce, Inc.