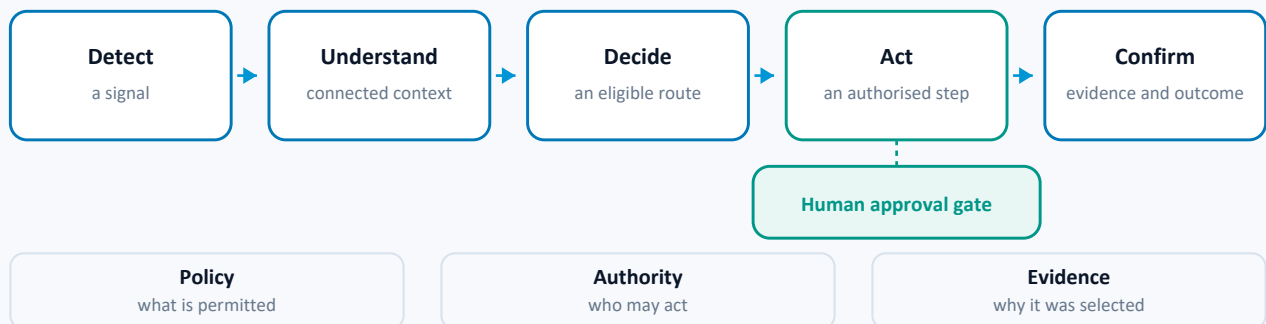


Pick a loop. Improve an outcome.

An operating model for agentic BSS — and how to choose the first one.

Most agentic AI proposals in telecom start from the technology and work outwards. This one starts from an outcome and works inwards, because that is the only end of the problem a business case can be written from.



One loop: a governed path from a signal to a measured outcome. Human approval is part of the path, not an exception to it.

In short

Telecom does not lack data. It lacks connected understanding — and that gap is where cost, delay and leakage live.

A loop closes the gap for one outcome at a time. It detects a signal, connects the telecom and customer context around it, selects a route that policy permits, progresses an authorised step, and records what happened. It is narrow on purpose: a loop that does one thing under clear authority can be evaluated, governed and trusted in a way that a general-purpose assistant cannot.

The decision this paper is meant to support is not whether to adopt a platform. It is which single outcome you would most like to improve first, because that choice determines everything that follows.

The problem is meaning, not data

The facts already exist. They sit in billing, ordering, product, service and CRM systems that work exactly as designed. What does not exist, in any one place, is the relationship between them.

What it looks like day to day	
Fragmented facts	Customer and operational data sit across systems that were never meant to answer a single question together.
Disconnected meaning	Products, contracts, policies and events are not consistently related to one another.
Manual decisions	A person reconstructs the situation before anyone can act — every time, from the beginning.
Stalled outcomes	Queues and hand-offs turn a decidable case into an ageing one.

“The gap is between knowing what happened and knowing what to do next.”

This is why reporting has not solved it. A dashboard describes the first half of that sentence accurately and has nothing to say about the second. A loop is the opposite: it exists only to get to the next action, and to show its working.

What a loop actually is

Five stages, one outcome, and three rails running underneath. The stages are unremarkable — that is the point, because each one is inspectable.

- **Detect** — recognise an event, request or exception worth acting on.
- **Understand** — connect the telecom and customer context that bears on it.
- **Decide** — select a route that is eligible under the operator's own rules.
- **Act** — progress an authorised step, or stop at an approval gate.

- **Confirm** — record the evidence and measure the outcome.

The three rails

Policy decides what is permitted, applied before an action is offered rather than after it is taken. **Authority** decides who or what may act: high-impact steps such as disconnection, large credits, write-offs, payment-plan exceptions and invoice holds stop for a person by design. **Evidence** records why a particular action was selected — the facts retrieved, the rule applied and the reason code — so the decision can be reviewed long after the moment has passed.

“Not another report. A governed path from signal to action.”

Choosing the first loop

Loops are grouped by the outcome they improve, not by the department that owns them. The examples below are a portfolio of possibilities rather than a standard catalogue, and no single loop defines the product — which is precisely why the first one should be chosen against your priorities rather than ours.

Outcome	Example loops	What you would measure
Protect revenue	Collections · Billing disputes · Revenue assurance · Contract compliance	Cash · Leakage · Dispute ageing
Reduce cost to serve	Care resolution · Order fallout · Case investigation · Process exceptions	Effort · Resolution · Backlog
Accelerate revenue	Quote-to-order · Offer eligibility · Renewals · Product changes	Conversion · Cycle time · Time to bill

Other loops span product, partner, enterprise customer and service operations. The pattern is the same in each case; what changes is the signal at the front and the measure at the end.

What “governed” has to mean to be worth anything

Governance that arrives as an audit log after the fact is not governance. Two things make the difference in practice, and both are visible in a demonstration rather than promised in a document.

1. The answer is structured the same way every time

Every response separates the facts actually retrieved from the assessment drawn from them, the recommendation from the action, and records an audit line for both. A reviewer always knows where to look, and a recommendation is never mistaken for something that has already happened.

2. Missing data is stated, not filled in

If a connector or dataset is unavailable, the loop says what it could not retrieve and shows the next query required. That is a stronger position than a fluent answer, and it is the single most useful thing to watch for when you are being shown any agentic system — including this one.

“That is stronger than pretending the answer exists.”

How you would know it worked

Measures are agreed with the operator against a baseline you already trust, before anything is built. For a collections loop that might be cash collected, avoidable disconnections, arrangement completion and case ageing. For revenue assurance it is leakage identified before invoice and the proportion corrected in cycle.

Two disciplines matter more than the choice of metric. Impact is scoped from source data rather than extrapolated, so a sample is never turned into an unsupported total. And commercial effect is reported only where it is present in the source systems — a loop that cannot see margin does not estimate it.

Where to start

Start with one loop, chosen for an outcome you already report on and already care about. Prove it against your own baseline. The telecom understanding, the governance rails and the integration work it requires are the same foundation the next loop inherits, which is what makes the second one faster than the first.

One dependency to raise early

The architecture assumes a Microsoft agent platform: Exosphere supplies the telecom meaning and the governed execution, and Microsoft supplies the surfaces and the agent runtime — Microsoft 365 Copilot and Teams where people work, Copilot Studio and Microsoft Foundry where agents are built and run. Your systems of record are untouched either way. Where that platform decision is already made this is an advantage; where it is not, it belongs in the first conversation rather than the integration phase. The companion technical briefing sets out the boundary in detail.

“Start with one loop. Build a reusable foundation.”

The question we would ask first is the one this paper ends on: **which outcome should your first loop improve — protecting revenue, reducing cost to serve, or accelerating revenue?**

Exos Systems is a TM Forum member and aligns to ODA, and operates to ISO/IEC 27001:2022. These describe the context we work in and do not imply endorsement by those bodies.

The loops named in this paper are examples of possible outcomes, not a standard catalogue and not a claim that any one use case dominates. Demonstrations run on synthetic accounts; any figures shown illustrate the mechanism and are not a performance claim.

Companion paper: *Standards-grounded agents — TM Forum, MCP and governed execution*, for architecture and integration teams.