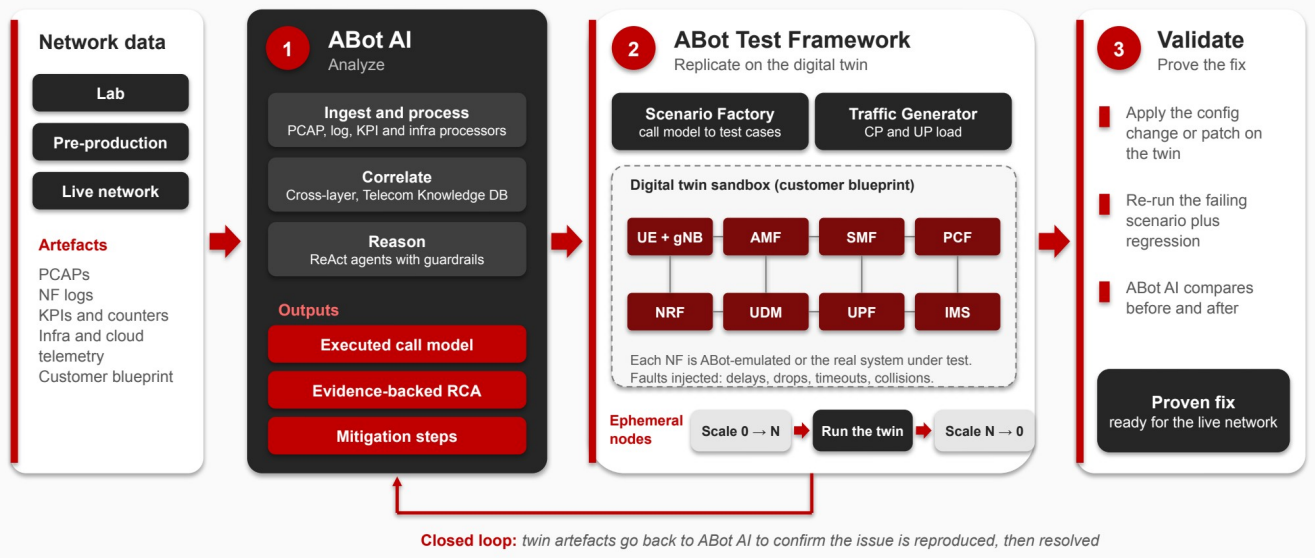
ABot 4G/5G Core Digital Twin

Find the issue in real network data, replay it in a safe sandbox, and prove the fix before it reaches live subscribers

When something breaks in a multi-vendor, cloud-native 4G/5G Core, the hardest part is usually not the fix. It is reproducing the problem somewhere safe, and then being sure the fix will not break something else. Trying changes directly on the live network is risky, and a shared lab rarely looks like the real deployment.

The ABot Core Digital Twin solves this. It is a working software copy of the operator's Core, built from the operator's own blueprint and driven by the call model recovered from real lab, pre-production or live network data. Engineers can break it, fix it and retest it as often as they need, without touching a single live subscriber.



A twin that mirrors your network and runs its real behavior

- **Mirrors the real Core.** Same network functions, interfaces, configuration and software versions as the deployment, built from the customer blueprint.
- **Runs what actually happened.** Scenarios and load come from the call model that ABot AI recovers from PCAPs, NF logs, KPIs and telemetry, not from a generic lab script.
- **Real or emulated, node by node.** Any NF can be ABot-emulated or the real system under test. Plug in a real AMF, SMF or UPF without rebuilding the lab.
- **Ephemeral and disposable.** Each twin is created for the run on K8s, EKS, OpenShift or VMs, then released. There is no fixed lab to maintain.

Analyze, replicate, validate: one closed loop

1. Analyze. ABot AI ingests network data, finds where the call flow drifted from a healthy 3GPP procedure, and hands over the executed call model along with an evidence-backed root cause.

2. Replicate. The ABot Test Framework turns that call model into executable test cases through the Scenario Factory, and the Traffic Generator adds realistic control and user plane load. Delays, drops, timeouts and collisions are injected until the field issue shows up in the sandbox.

3. Validate. Apply the config change or patch on the twin, re-run the failing scenario with regression, and let ABot AI compare before and after. What comes out is a proven fix, ready for the live network.

Where the digital twin helps

Issue replication Reproduce a fault seen in the lab, pre-prod or live network inside a safe sandbox, on demand.	**Root cause confirmation** Prove the RCA hypothesis by making the issue happen again under controlled conditions.
Mitigation validation Try a config change or patch on the twin first and check it does not break anything else.	**Upgrade and change impact** Run the real call model against a new NF release or configuration before it goes live.
Load and capacity what-ifs Recreate traffic mixes, signalling storms and peak load at realistic scale.	**Security and resilience** Simulate DDoS, procedure collisions, delays and drops, then verify the countermeasure.

Ephemeral by design: one twin per blueprint, zero idle cost

Most teams keep a fixed lab that drifts away from production and sits idle between test cycles. The twin works the other way round. ABot keeps a persistent framework, while the emulator nodes and SUT nodes exist only while a test is running. Elastic workers scale from zero for the run and back to zero afterwards. The artefacts are kept, the compute is released, and the next run starts on clean nodes. Cost follows the test, not the calendar.

1M UEs	1200 to 1500 CPS	< 20 mins	Zero idle nodes
per ABot instance	control plane, per instance	to bring elastic workers online	compute released after each run

Why ABot Digital Twin

- **Evidence, not guesswork.** Every root cause claim points to a packet, a log line or a timestamp, and the same input gives the same answer.
- **Any call flow, any failure.** ABot NFs are not bound to a fixed state machine, so corner cases, procedure collisions and DDoS are all in scope.
- **Vendor-neutral.** One platform finds the issue, reproduces it and proves the fix, whichever vendors make up the Core.
- **Proven in production.** A hyperscaler's Telco support team already uses it to replicate CSP issues and validate fixes before rollout.

The outcome: lower MTTR and safer network changes. Support teams find the issue, reproduce it and prove the fix in a sandbox before it ever touches live subscribers.

About Rebaca Technologies

Rebaca Technologies is a niche solution provider in the Telecom and OTT video streaming domain. We have delivered customized high-tech solutions and services to industry leaders since 2002, covering software development and test automation in the Video and 4G/5G/ORAN domains.

Infinity Benchmark Tower, 9th Floor Plot
G1, Block-GP,
Salt Lake City, Sector-V, Kolkata-700091

+91-33-4009-7177
marketing@rebaca.com
https://www.rebaca.com/

Rebaca