

A Solution for Today and Tomorrow

Feature Description

Table of Contents

1 Introduction **1**

2 Start with One Service **1**

3 Grow with New Services **2**
3.1 A Typical Path . . . 2
3.2 What Each New Service Reuses . . . 2
3.3 One Platform, Many Interfaces . . . 3

4 A Platform That Keeps Evolving **3**
4.1 New Building Blocks in Existing Tools . . . 3
4.2 Evolving Safely . . . 4
4.3 Release Lifecycle . . . 5

5 Deployment That Follows Growth **5**
5.1 Scaling . . . 5
5.2 Deployment Models . . . 5

6 Licensing That Grows with You **5**

7 Summary **6**

1 Introduction

SRE is usually purchased for a single service, and then becomes the platform on which the next services are built. The SRE Product Description summarises this path.

This document explains what makes SRE a platform that grows with its users: how new services build on existing ones, how the platform evolves without disrupting live traffic, and how deployment and licensing follow this growth.

2 Start with One Service

A first SRE project is usually driven by a clear, limited business case. It requires:

- The integration with the network: SBCs, call control platforms and other SIP equipment.
- The integration with the IT systems: provisioning from the OSS/BSS, CDRs for billing, monitoring.

- A data model reflecting the customers, numbers and services concerned.
- The service logic implementing the service.
- The operational processes: monitoring, troubleshooting, change management.

3 Grow with New Services

3.1 A Typical Path

Stage	Example services	Building blocks used
First service	SIP trunking screening, number portability, Teams Direct Routing	SIP, internal data model, REST provisioning
More routing services	New screening rules, number translations, least cost routing, emergency call handling	Additional tables and service logic on the same platform
Integration services	Routing decisions exposed to partners, webhooks to and from business applications	HTTP endpoints, ENUM, external queries
Voice services	Announcements, IVRs, call screening, click-to-call	Media services
AI services	Voice-driven menus, virtual agents, call classification	Text-to-speech, speech-to-text, LLM nodes, real-time AI agents

The order varies from customer to customer, but the principle is the same: each new service is a new service logic, and possibly a few new tables, on a platform that is already in place.

3.2 What Each New Service Reuses

Each new service builds on what is already there:

- **The network integration.** The SBCs and other equipment already send their requests to SRE.
- **The IT integration.** Provisioning, CDR collection, monitoring and alarming are already connected.

- **The data model.** Customers, numbers and services are already provisioned. A new service often only adds a few columns or tables, through a new data model version activated without service impact.
- **The service logic.** Common functions, such as customer identification, number normalisation or CDR enrichment, are built once as sub-logics and reused by every service. The logic to execute can even be selected at runtime, based on data.
- **The skills and processes.** The teams already know the editors, the simulation tools and the release process.

Compared with a dedicated solution, a new service on SRE avoids a tender, interoperability testing with the existing network, a new OSS/BSS integration and training on a new tool. The project is reduced to designing the service logic, extending the data model where needed and validating the result with the simulation tools.

3.3 One Platform, Many Interfaces

The same service logic and data are available on every interface supported by SRE:

- **SIP**: redirect, proxy and registrar modes
- **ENUM** queries
- **HTTP** endpoints, to expose APIs and receive webhooks
- **Media**: calls answered and controlled by SRE
- **WebSocket**: event streams from external platforms

A customer record created for a SIP trunking service is directly usable by an IVR, an API or an AI agent built later.

4 A Platform That Keeps Evolving

4.1 New Building Blocks in Existing Tools

SRE evolves by adding new building blocks to the existing editors, rather than new products. Recent releases illustrate this:

Release	Examples of new capabilities
SRE 4.0	SIP registration client, built-in scheduler, service logic tagging and change log, fine-grained access control, hitless and automatic rolling upgrades, automatic database switchover
SRE 4.1	Media server, SIP/SIP-I proxy, Diameter client, raw SQL queries, simulation from call traces, UUID columns and column renaming in the data model
SRE 4.2	Speech-to-text, music on hold, N-way bridging, WebSocket processor, webhook verification, MongoDB operations, runtime-resolved service logic, Kubernetes deployment

More recently, text-to-speech and AI building blocks, such as LLM queries and integration with real-time AI agents, have been added in the same way.

Each new capability appears as new nodes in the Service Logic Editor. Customers can use it in the services they already run, without changing platform, migrating data or redesigning existing service logic.

4.2 Evolving Safely

Growth only creates value if it does not put existing services at risk. SRE provides the tools to change a live platform safely:

- **Versioned service logic.** Every change is a new version that is validated before release and can be rolled back with a single click. A change log records who changed what.
- **Simulation and regression testing.** Test cases are saved as a baseline and replayed on every new version. Simulation data can be loaded from real call traces, so that new versions are tested against real traffic.
- **Versioned data model.** New data model versions are activated without service impact, with backups and history.
- **Dual database.** Large data changes and new integrations are prepared on the standby database before being switched to active.
- **Hitless upgrades.** Call Processors are upgraded one by one through an automated rolling upgrade, without call impact. The release notes document the upgrade procedure and, where applicable, the downgrade procedure.

4.3 Release Lifecycle

SRE follows a published release lifecycle. Minor releases, which bring new features, aim to remain backward-compatible with the previous minor release or provide the tools for a smooth update. Each major version benefits from an active support period of about two years, followed by reactive and extended support periods. Customers can therefore plan upgrades and benefit from new capabilities at their own pace.

Refer to the Release Management Lifecycle document for details.

5 Deployment That Follows Growth

5.1 Scaling

SRE scales horizontally. When traffic grows or new services are added, capacity is increased by adding Call Processors. Additional Call Processors can also be deployed closer to the network elements they serve, for example per region or per data centre.

5.2 Deployment Models

SRE runs on virtual machines, on dedicated hardware, in containers or on Kubernetes, and is also available as a service operated by Netaxis (SREaaS). Customers can start with the deployment model that fits their current infrastructure and move to another one as their IT strategy evolves, for example from virtual machines to Kubernetes, while keeping their service logic and data model.

6 Licensing That Grows with You

The SRE licensing model supports a step-by-step approach:

- The service provider starts with the capacity and the options that the first service requires.
- Capacity is increased as traffic grows.
- Options (such as media, HTTP, ENUM, CAC, STIR/SHAKEN or AI) are added when a new service needs them.

The service provider only pays for what it uses, and each new service is an incremental investment on top of the existing platform.

7 Summary

SRE is usually introduced to solve one problem, and often ends up delivering much more value than the initial business case anticipated:

- New services reuse the network integration, the IT integration, the data model and the skills already in place.
- The platform keeps evolving, adding new interfaces and building blocks to the existing tools.
- Changes are made safely on a live platform, with versioning, simulation, dual databases and hitless upgrades.
- Deployment and licensing grow step by step with the services.