



Product Description

SRE 4

Table of Contents

1	Introduction	1
2	Scalable Design	3
2.1	High Level Architecture	3
2.2	SRE Building Blocks	4
3	Unlimited Routing Capabilities	5
3.1	Service Logic	5
3.2	Data Model	6
3.3	SIP Services	7
4	Operational Excellence	8
4.1	Deployment	8
4.1.1	On-Premise Deployment	8
4.1.2	As a Service	9
4.2	Management	9
4.2.1	Role-Based Access	10
4.2.2	Dual Database	10
5	Multiple Use Cases	11
5.1	SIP Trunking Offering	11
5.2	Outbound Routing - Use Case 1	11
5.3	Outbound Routing - Use Case 2	12
5.4	Inbound Routing	12
5.5	Microsoft Teams Direct Routing	12
5.6	Centralised Routing in a Carrier Network	13
6	Simple Licensing Model	14

1 Introduction

Carrier networks are more complex than ever, combining a multitude of networks elements of different origins. The routing logic is then often spread over this equipment due to (mainly) historical reasons. This makes the global routing configuration, the related troubleshooting or other maintenance activities extremely complicated. It can also slow down the development of new features and services.

Netaxis Solutions has developed the Session Routing Engine (SRE) to drastically simplify this routing

challenge. SRE is a centralised, carrier-grade and multi-protocol routing platform for voice and other type of sessions.

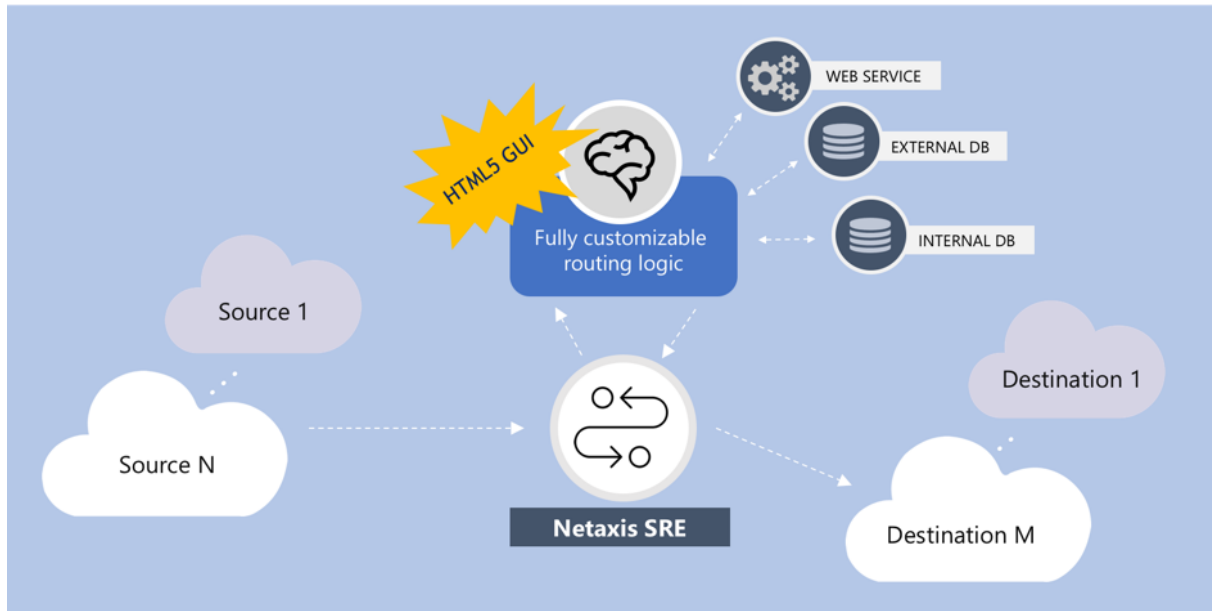


Figure 1: SRE: Centralised routing engine

The platform features in a nutshell:

1. The SRE centralizes routing decisions in a VoIP (or other SIP based) network
2. The platform is multi-protocol and support triggers via SIP, ENUM and HTTP
3. The distributed architecture assures local and geo redundancy while scaling up to thousands of CAPS
4. It provides the service provider a GUI to create, test and release new routing logics. Several nodes are available to analyze, query and manipulate
5. This flexibility is extended to the implementation of the supporting data model. External data sources can complement this internal database. They are queried via ENUM, HTTP (Restful API calls)
6. The SRE is a software only solution. It runs on a standard virtualized hardware or in a container environment. It can also be consumed as a SaaS product.

2 Scalable Design

2.1 High Level Architecture

The client – e.g. the SBC or any other call control equipment - asks the SRE which routing policy should be applied. The SRE receives the request, applies the service logic to provide the client with the routing information. Based on this answer, the client routes the call to the correct destination. Different types of interfaces are exposed: SIP, ENUM and HTTP. Specifically for SIP, the SRE can act as a proxy. The SRE receives a SIP request from another network equipment. It applies the service logic and proxies the request to the correct destination, also providing its next hop any required information useful to reach the final destination.

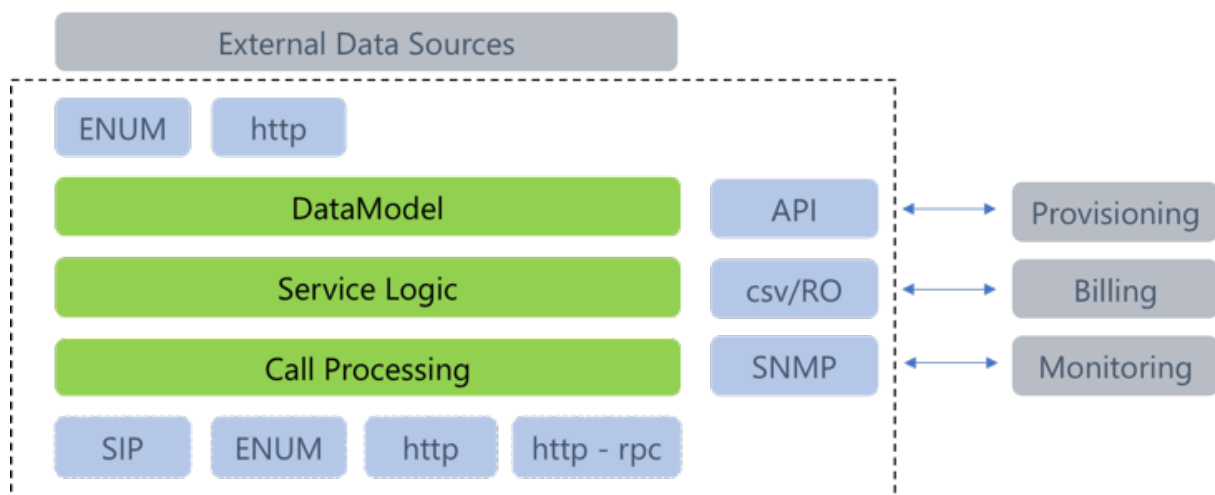


Figure 2: High level architecture

The service logic is defined via a GUI. The execution of the service logic is supported by the data stored in an internal database. External data sources are queried via DNS/ENUM, Diameter, HTTP JSON, HTTP XML. LDAP (Active Directory) queries are possible as well to support use cases for enterprise customers.

The system supports the integration in the service provider BSS and OSS IT systems in different ways: - The internal database of the SRE comes with a set of APIs to provision the data. GET/PUT/POST and DELETE operations are supported. This possibility comes on top of the manual provisioning or bulk provisioning via CSV files - The SRE creates Call Detail Records (CDR) for billing and reporting purposes. The call information includes called and calling party, call duration..., the CDRs comes with a default format including 50 fields. They can be customized by adding or modifying fields via the service logic, using information from signalling as well as from the data model. The Ro-interface was also added to support on-line billing - The SRE generates SNMP for performance and alarm monitoring

2.2 SRE Building Blocks

The SRE has two types of building blocks.

The *Element Manager* (typically one active and one standby), is providing the GUI, the master DB with API's to manipulate the data, the system configuration, CDRs logs and stats.

Several *Call Processors* process the actual routing requests from the call control, based on a local copy of the master DB. They keep track of the peer SIP agents' availability by probing via OPTIONS messages. Logs, call tracing information and stats are sent to the Element Managers.

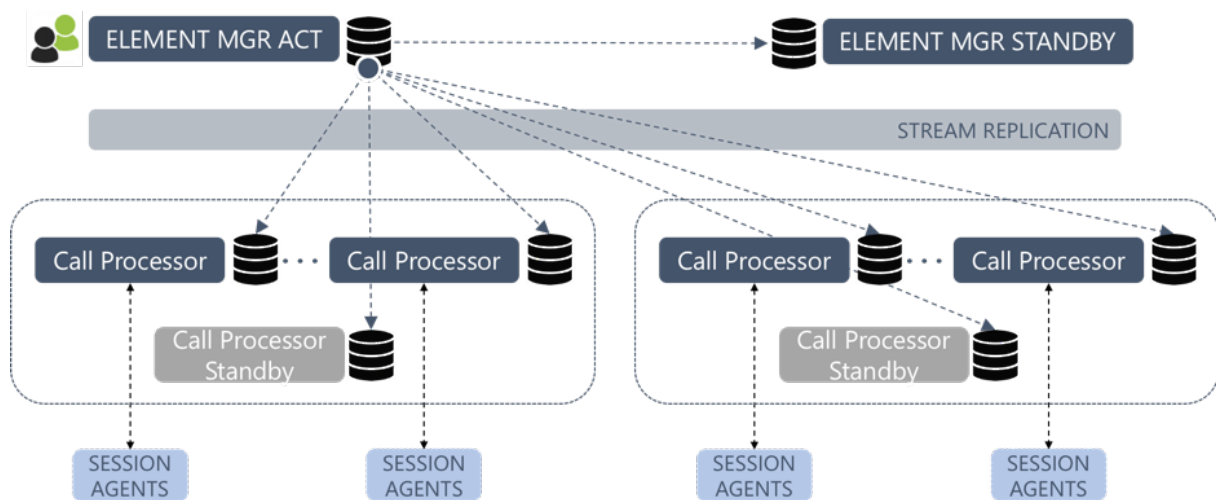


Figure 3: SRE building blocks

The CPs are self-contained call processing instances. The number of call processors depends on the capacity needs (expressed in Call Attempts per Second or CAPS), redundancy and the size of the supporting virtual machines. This allows the SRE to scale horizontally into the thousands of CAPS. The system recovers from an infrastructure failure without call impact. Typically, the platform is designed in a N+1 mode in two datacentres and as such is protected against local infrastructure failure or even the outage of a complete datacentre (geo-redundancy). Further decentralisation is possible either to reduce the cost of redundancy or to increase the proximity with the call control, e.g. in the case of a SIP trunking scenario.

3 Unlimited Routing Capabilities

3.1 Service Logic

The service provider creates the service logic in a graphical way via the Service Logic Editor. The creation of the logic is done by linking various nodes, each of them performing operations such as:

- Analysing and manipulating SIP messages
- Combining information, stored in variables, from the original request with data from the internal DB tables and/or external systems
- Ultimately deciding when to relay the call, to block or crankback instead

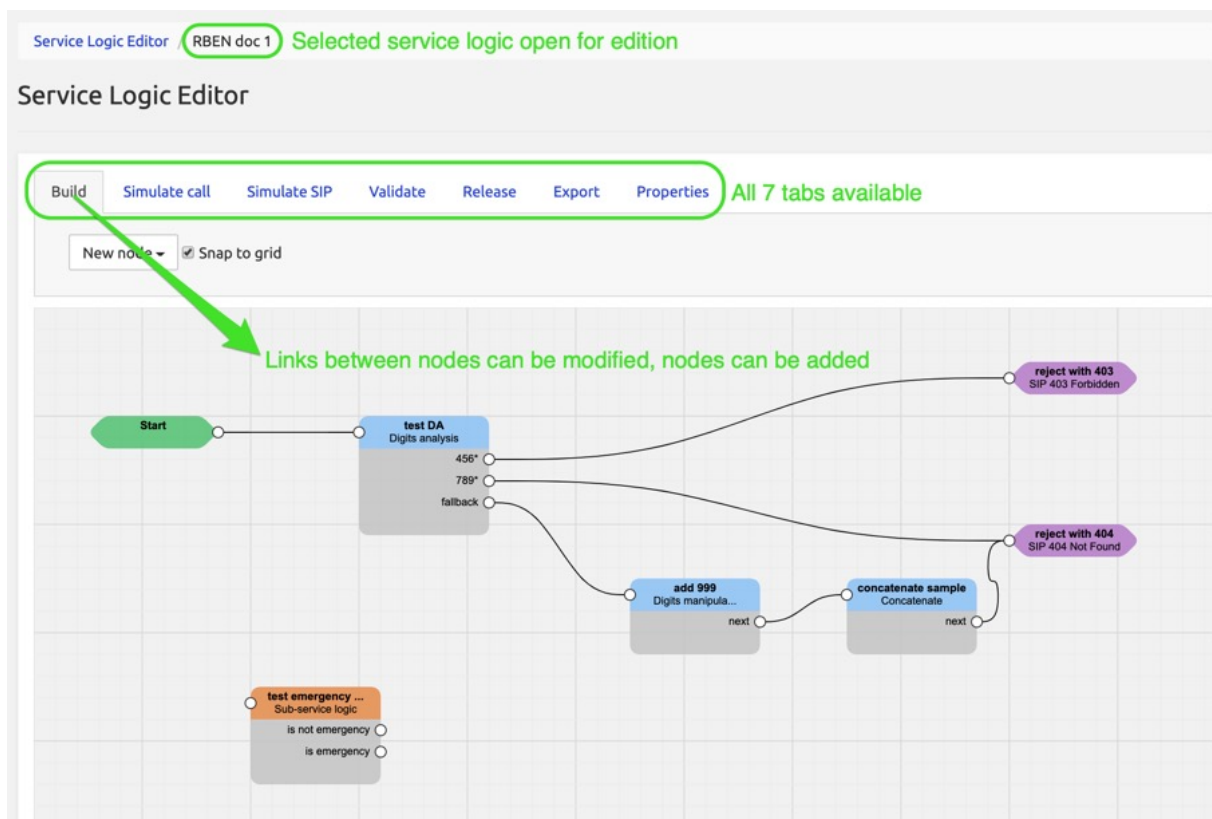


Figure 4: Service Logic Editor

The service provider thus defines a service logic combining e.g. call blocking for blacklisted customers or incorrect incoming SIP headers, emergency call handling, least cost routing, number translation, codec-based routing etc. Some use cases are illustrated further in this document. The service logic can be applied on any SIP method (e.g. INVITE, NOTIFY, ...) as well as SIP provisional responses.

The ease of interfacing with existing IT systems through various protocols ensures that the SRE can seamlessly adapt to the current service provider infrastructure, data, and business practices, rather

than requiring the infrastructure to adapt to it. This flexibility empowers service providers to leverage their existing systems and practices while integrating new technologies and capabilities.

The service provider validates the new or adapted service logic in the GUI. The built-in simulation tools remove the need for any long and tedious real-call testing. A simulation timeline shows the status of variables at each step in the service logic and the time needed to execute the actual step. Test cases are saved as a baseline for future validations. The service logic is released in production after a successful validation. Rolling back is always possible with a simple click.

3.2 Data Model

The service logic execution is supported by a data model that can be fully customized and managed by the service provider. This is done via the Data Model Editor. This function allows defining tables, their different columns and their relationship.

Information like customer ID and names, phone numbers, trunk groups, number portability prefixes are examples of data that the service provider typically stores in the SRE. Fields can be of the type of text, integer, float, boolean. They can be nullable or assume a default value. Along with the fields definition you can add constraints enforced at DB level. For each table, you can enable .csv import and export for batch provisioning.

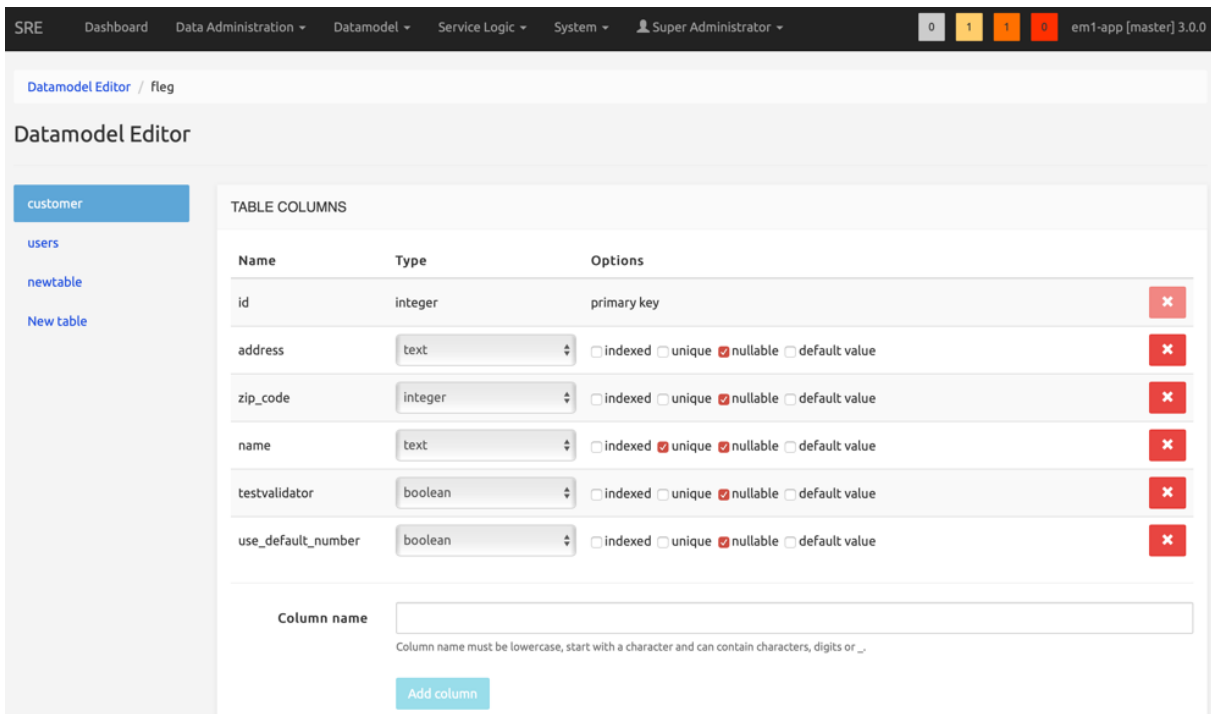


Figure 5: Data model editor

This process is not static. When a data model version is created and then activated, the GUI as well as other provisioning interfaces - being the REST API and bulk provisioning - are adapted to provide the needed provisioning interfaces automatically. Every time a Data Model definition is saved through the editor, a new version of the Data Model is available to be activated (without impact on the service). Older versions are stored to allow for roll-back operations.

While most equipment forces the service provider to adapt its provisioning model to the configuration-based nature of the voice equipment, the SRE can be adapted to match the existing service provider provisioning model. This flexibility allows the transition from configuration-based routing to data-driven routing, where provisioning modifications are immediately reflected in the applied routing.

3.3 SIP Services

Call Admission Control (CAC) determines whether to accept or reject a new call, ensuring quality of service, resource assessment, and policy enforcement. It uses identity information extracted from the protocol or data provisioned in the internal DB as criteria to monitor active calls and manage thresholds effectively.

The SIP registrar functionality enables the handling of the registration of SIP devices, storing dynamic device location information in an internal location database. This process ensures that devices are properly registered and can be easily located within the network, facilitating efficient call routing and user mobility.

Conversely, surrogate registration allows the SRE to function as a SIP registration client for other equipment, registering on behalf of users or devices. This is particularly beneficial when users lack the capability to register themselves, when the topology changes dynamically, or under other constraints. In this role, the SRE acts as an intermediary node, concealing the potentially dynamic topology of end-users behind itself.

STIR/SHAKEN is a framework of standards and protocols designed to combat caller ID spoofing and enhance the security of telephone networks. For originating service providers, an authentication service can be built using the SRE. This service generates a proper Identity header and adds it to the outbound SIP message. Similarly, for terminating service providers, a verification service can be implemented. The SRE extracts the Identity header from the received INVITE message and validates it. All necessary building blocks are available to handle certificates, public/private keys, and interface with external repositories.

4 Operational Excellence

4.1 Deployment

4.1.1 On-Premise Deployment

SRE is a software only solution. It runs on virtual machines or on a container-based environment. Netaxis supports installations on dedicated COTS hardware as well. Dimensioning of the underlying infrastructure will be done with the Netaxis staff based on traffic expectations.

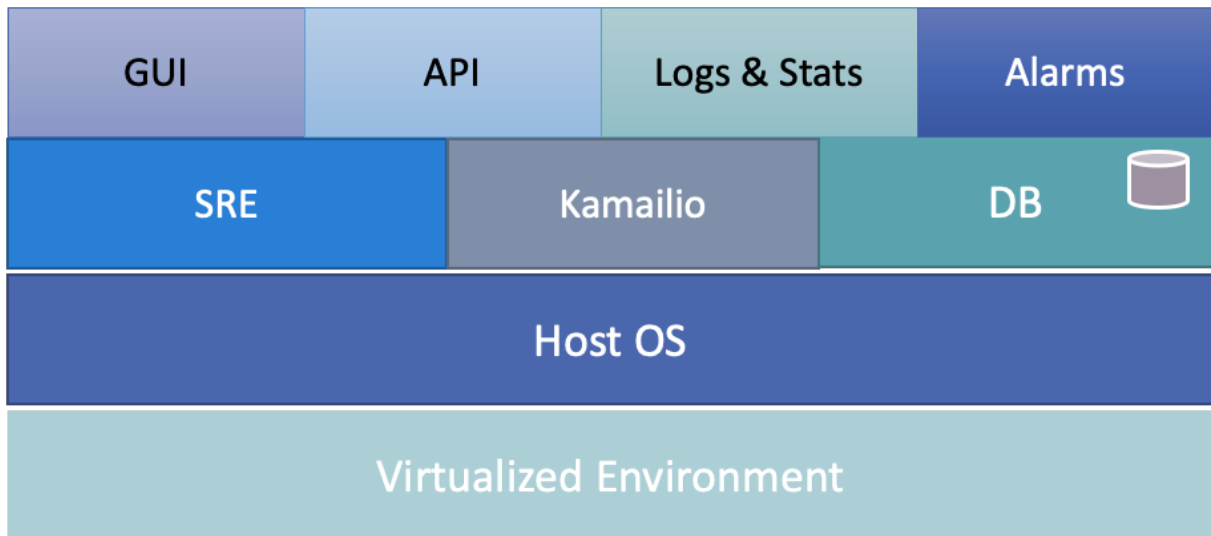


Figure 6: Virtual machine deployment

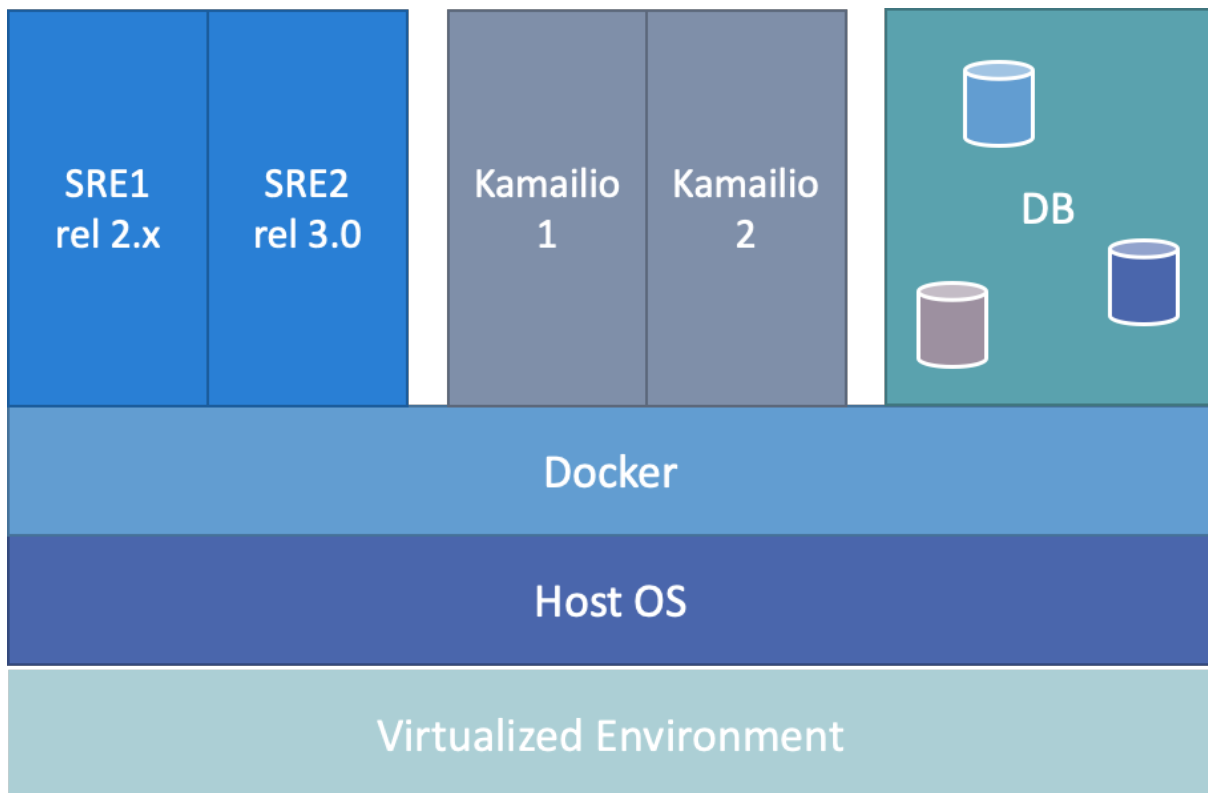


Figure 7: Container deployment

4.1.2 As a Service

Service providers and Enterprises can also subscribe to the SREaaS product operated by Netaxis. This platform can host new customers without the need for them to deploy any software. This translates into a capex reduction and a quicker time to market. In addition, it removes the needs, for the Service Provider or Enterprise, to maintain and secure the lower layers (virtualization, OS, etc.) since Netaxis will take care of those.

The onboarding includes the setup of a SIP trunk connection to the cloud front-end, the design and setup of a dedicated data model and service logic (done by the customer or with the support of Netaxis). The will access a dedicated GUI where Data Model and Service Logic can be edited, exactly in the same way as in an on-prem deployment.

4.2 Management

The SRE offers an administrator portal. On top of the service logic and data model editor explained above, the GUI offers a status service and statistics dashboard, alarming and extensive logging capabil-

ities. Furthermore, it allows for in-GUI call simulation which dramatically reduces the time to find the point in the Service Logic where the issue may show up.

The statistics framework provides in-depth insights into service logic performance, system usage, and the health of underlying subsystems. This allows the service provider to monitor platform performance in real-time, ensuring optimal operation and swift identification of any issues.

The logging and tracing capabilities enable the capture of network calls, facilitating the identification of potential issues directly from the GUI. Activation occurs based on predefined criteria, enabling call-per-call activation without overloading the system. This streamlined approach ensures efficient troubleshooting and maintenance while minimizing system strain.

The CDR post-processing engine facilitates additional processing on completed CDRs, including augmenting call metadata with data from external systems or rating CDRs based on their duration and characteristics, supporting various business needs.

4.2.1 Role-Based Access

SRE users are created with a username and password for local authentication and are assigned a role. As an alternative, users can be authenticated against an Active Directory as well. Roles are defined into profiles granting access rights (Read-Edit-None). Depending on the access rights granted to the role a user has been assigned, some items might not be visible for that user. The fine-grained access rights system ensures that the users can only access parts of the service logic and data provisioning that their role grants them access to.

The auditing capabilities allow the service provider to monitor all actions taken on the system, ensuring compliance and enhancing security. This detailed tracking helps identify and address potential issues, supports regulatory adherence, and provides a clear audit trail for accountability.

4.2.2 Dual Database

SRE natively comes with a dual version of its internal databases hosting the data (numbers, trunk info, ...). When it comes to provisioning of massive quantity of data, or testing of a new provisioning platform using APIs, the standby version of the DB can help avoiding any impact on the data set supporting the running service logic.

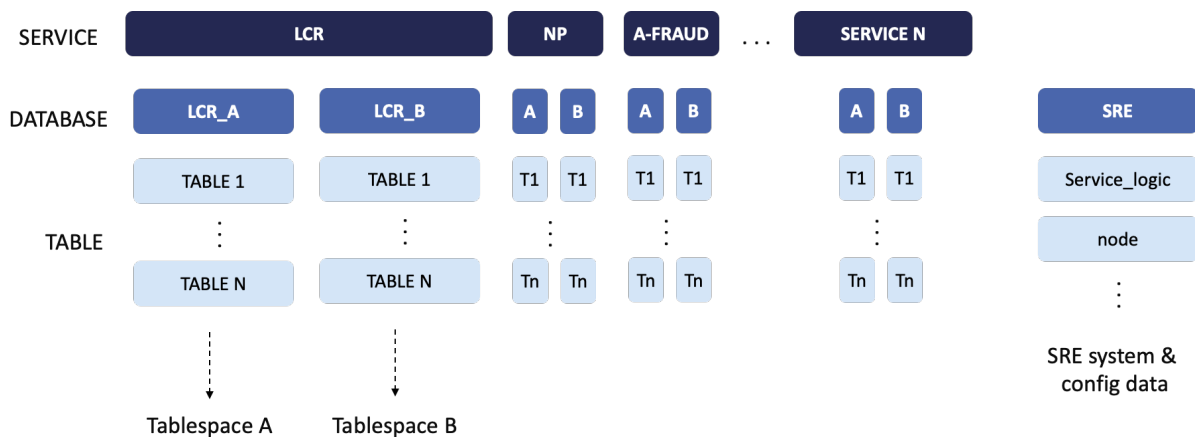


Figure 8: Dual database

5 Multiple Use Cases

Here follows a few use cases from some live SRE customers. Those examples are just illustrative and intended to show SRE's flexibility. We are happy to hear about your own routing challenges.

5.1 SIP Trunking Offering

In many cases, SIP trunking is offered purely based on class 4 routing between SBC's. Neither the inbound nor the outbound traffic is handled by a class 5 application. Call screening is not in place – which opens the doors to fraud. All provisioning actions, as well as the moves and changes, are done manually. A solution is to connect the SRE on top of the existing SBCs, hence moving the routing intelligence to SRE. For each of the incoming or outgoing calls, the SBC will send an INVITE to the SRE. Several actions such as screening of trunks and CLI, SIP header manipulations, call admission control, crankback can be performed. As such the SRE takes the role of a light SIP trunking application server: - Block the call if the CLI of the outgoing call if the number does not belong to the range attributed to the customer, when the number of the calls exceeds the maximum CAC-value for that customers or when the customer is marked a 'bad payer' - Change the From and PAI when needed - Choose an alternative trunk group when the primary trunk group fails or is congested

5.2 Outbound Routing - Use Case 1

A service provider is relying on one partner to route all outbound traffic to the other licensed service providers in the same country. The local regulator requires that the service provider performs a number

portability check and adds a routing prefix before the call is actually routed to the partner. The service provider maintains a table of the internal SRE DB in synchronization with the central number portability database of the regulator, mapping the phone number with the service provider. Another table is available to map the service provider with the routing prefix and the SIP domain. In case of an outbound call, the interconnect SBC will trigger the SRE and will send a 302 Redirect message with Contact: sip:<RoutingPrefix><NSN>@olo_x.com (the format is configurable in all its parts).

5.3 Outbound Routing - Use Case 2

A service provider is relying on one partner to route all outbound traffic to the other licensed service providers in the same county. The transit partner performs a number portability check as part of the service. The service provider wants however to save on the transit cost by interconnecting directly with 3 main olo's while keeping the transit partner for the rest of the traffic.

As in the previous case, the service provider maintains a table of the internal SRE dB in synchronization with the central number portability database of the regulator, mapping the phone numbers with the service providers. For each of the outbound calls, the interconnect SBC will trigger the SRE and will send a 302 redirect message pointing to the right direction – either the 3 main olo's directly connected to the service provider, or the transit service provider.

5.4 Inbound Routing

A service-provider-1 has 3 call control platforms delivering Class5 services. Today, the inbound call routing to the right platform is performed via the SBC itself. It requires the definition of local routing tables on the SBC. service-provider-1 wants to partner with a new service-provider-2. service-provider-2 will bring in a new call control platform. Inbound and outbound call of both service providers will go through the same SBC.

However, service-provider-1 does not want to grant service-provider-2 access to its SBC to define its own LRT. Instead, the SBC will be connected to a SRE. Dedicated local routing tables will be defined for each service provider. Using the role-based access feature of the SRE, the view of service-provider-2 is limited to their own tables only. This assures that inbound traffic as well as the traffic between both service providers is handled safely.

5.5 Microsoft Teams Direct Routing

Many service providers do have a Microsoft Teams Direct Routing offering. The phone system of the customer's Teams tenant is connected to a SIP trunk via SBCs in the data centers of the service provider.

This allows PSTN to Teams calls to be established. On top of the basic direct routing, the service provider enriches the service with e.g. a fax service, voice recording, some kind of fixed mobile convergence or a fallback to mobile should Microsoft Teams connection fail.

Each customer requires a specific routing configuration on the SBC. To do so, a manual SBC provisioning is feasible as long as the number of customers remains limited. In order to scale this offering, some automation is required. The SRE offers a way to abstract the complex routing logic from the SBC thanks to its predefined service logic and the presence of a datamodel based on customized information such as customers, users, number and services (instead of routing based on static concepts like trunkgroups and routing prefix). This abstraction is certainly needed in case the direct provisioning routing is to be automated.

5.6 Centralised Routing in a Carrier Network

For historical reasons, one Carrier runs different TDM/VoIP platforms, all of them configured and patched over time to (try and) follow a consistent business logic when it comes to call routing. The Carrier decides to centralize the routing logic to a more modern and dynamic platform in order to solve its network fragmentation and allow for future-proof services, while ensuring the classic Least Cost Routing logics.

The SRE is aware of the Carrier's Voice Products (Platinum, Gold, Premium, OTT, ...) as they are actually implemented in the platform. Least Cost Routing can be complemented with QoS based routing and, when a trunk is discarded due to bad quality or due to open issues, it can be easily removed from routing in a single point of the network.

Customized alarms based on traffic thresholds can support the upselling process. Barring is available and can be combined with different logics and conditions, the same way as other call flows are, thus not based anymore on a static feature in an old-fashioned switch. New business demands can be easily designed, tested and implemented by Engineering and Operations teams network-wide or applied to portions of the network, without impacts on the running call flows and with improved Time-to-Market.

The available REST APIs on SRE are now used to provision the platform, hence reducing the OSS/BSS complexity and troubleshooting needs. Finally, running the platform on a virtualized environment allows the Carrier to more easily scale by adding new VMs, thus reducing the time for network growth, with automatic scaling up/down in the product radar.

6 Simple Licensing Model

SRE is licensed in a simple and transparent way. One SRE license allows the SRE to manage one CAPS of traffic. The basic SRE license gives access to most SRE features with a few optional features requiring specific add-on licenses. Those are described below.

Reference	License name	Description	Type
SRE-LIC-BASIC	SRE Basic Licenses (SIP)	Basic SRE license required to run SRE	Requests/sec
SRE-LIC-ENUM-DNS-CLIENT	SRE Option ENUM/DNS Client	Required to perform ENUM or DNS requests	On/Off
SRE-LIC-CAC	SRE Option CAC	Required to perform Call Admission control	On/Off
SRE-LIC-FLEX-ACCOUNTING	SRE Option Flex Accounting	Required to customize the default CDR	On/Off
SRE-LIC-LDAP-AD	SRE Option LDAP/AD	Required to connect to a LDAP / AD directory	On/Off
SRE-LIC-HTTP	SRE HTTP LICENSE	Required for the SRE to be able to process HTTP requests	Requests/sec
SRE-LIC-ENUM-SERVER	SRE ENUM SERVER LICENSE	Required for the SRE to be able to process ENUM requests	Requests/sec