Garderos Configuration Server

Introduction to auto-configuration

The management philosophy of Garderos for large-scale projects which have routers located at tens of thousands of remote sites, is to use auto-configuration. Auto-configuration means that after being switched on and booting for the first time, each router must fetch its own configuration from a configuration server, which can be any web server or the Garderos Configuration Server (GCS). After the initial boot-up and successful auto-configuration, the router must thereafter regularly query the GCS for any updates to its configuration and/or firmware version. In this way, a multi-site network of tens of thousands of routers can be managed with a minimum of operational effort. This type of efficient manageability means that the cyber-security of the network can be kept very high, by ensuring that all of the routers have up-to-date configurations and firmware version.

Garderos routers use an HTTP-API for auto-configuration. Using HTTPS, each router downloads a centrally stored configuration file (see Fig. 1)and configures itself. Thereafter, the router will periodically check for any changes to its configuration or for new firmware within a configurable period of time. This configurable interval is contained in the configuration.

The Garderos Configuration Server supports your large-scale roll out in an easy and efficient way. When operating a network with many routers, most configuration parameters like the addresses of NTP servers, SNMP servers and which interfaces to activate will be the same on all devices, while a few parameters, like the IP addresses on the router's LAN interfaces and maybe IPsec tunnel policies depend on the router's location. The Garderos Configuration Server dynamically creates configuration files from templates, replacing macros by values taken from a configuration database.

The Garderos Configuration Server consists of up to 3 components, which can be run in Tomcat Java Application Servers:

- The configuration servlet creates the configuration files for the Garderos routers and stores access data in the database.
- The administration servlet is a management interface used to manipulate the routers in the Garderos Configuration Server's database and to view router status and statistics.
- An optional logging servlet can collect and store data sent from the routers by HTTPS requests.

While the Garderos Configuration Server is implemented as platform-independent Java servlets, the typical setup is a Linux server (e.g. Ubuntu Server) with Tomcat application server.

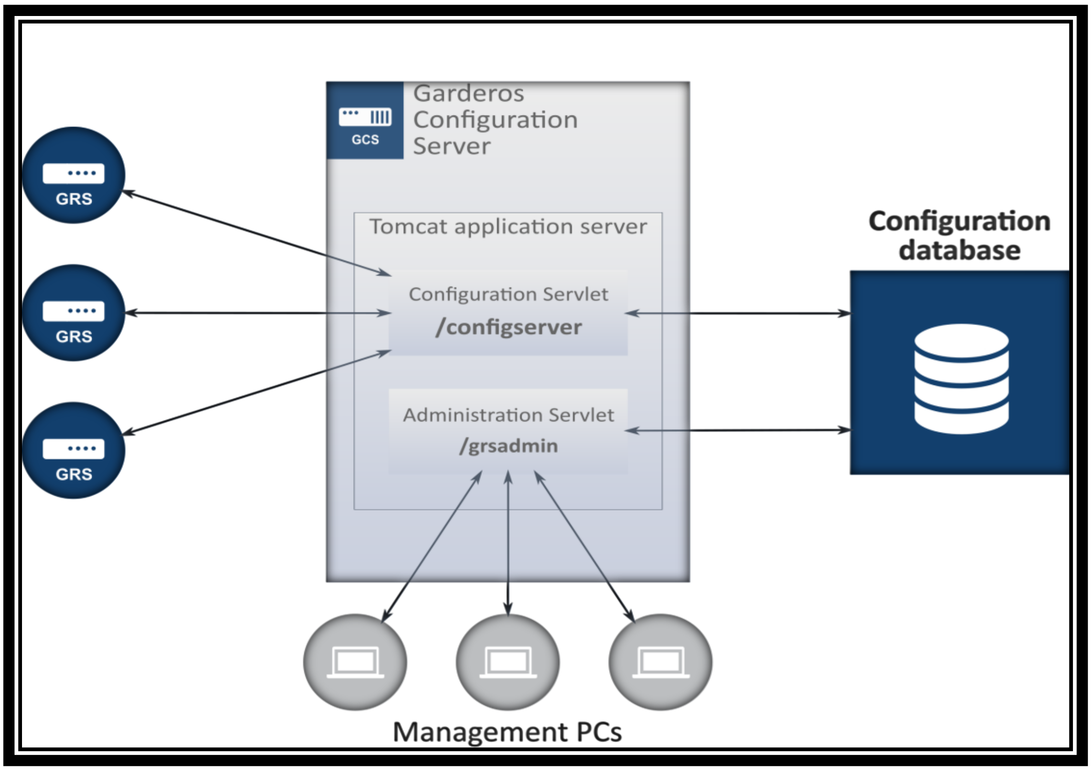
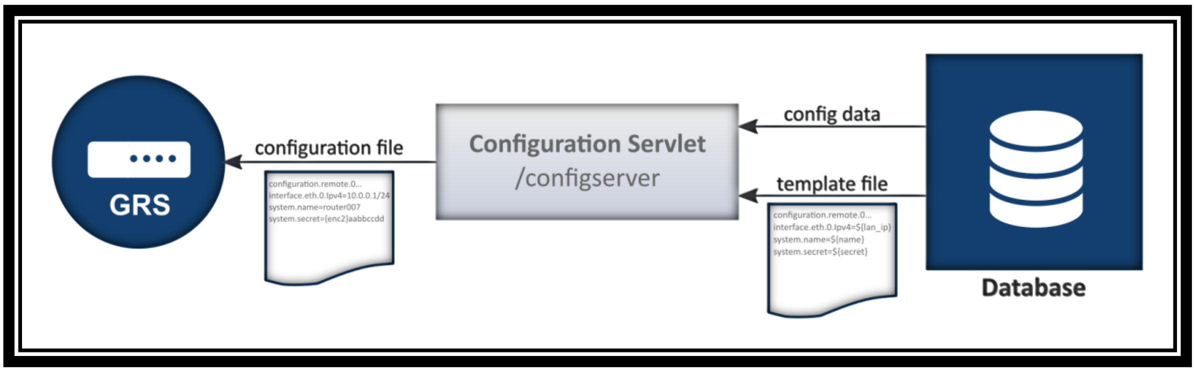


Fig. 1: Basic architecture

The servlets are implemented in a modular way and can both run on the same application server or on different servers. Redundancy is possible by connecting several Configuration Servers to the database. Database redundancy is supported by standard MySQL replication mechanisms.

Configuration Servlet

The routers requesting a configuration file send a unique identifier (usually their name) and a hash of their secret to the Configuration Servlet. The Configuration Servlet will look up the identifier in the database and check the router’s secret. If a matching router configuration is found and the secret is valid, the Configuration Servlet picks the correct template and fills the macros with the corresponding values from the database (see Fig. 2)



```
configuration.remote.0...
interface.eth.0.Ipv4=10.0.0.1/24
system.name=router007
system.secret={enc2}aabbccdd
```

```
configuration.remote.0...
interface.eth.0.Ipv4=${lan_ip}
system.name=${name}
system.secret=${secret}
```

Fig. 2: Illustration of configuration servlet

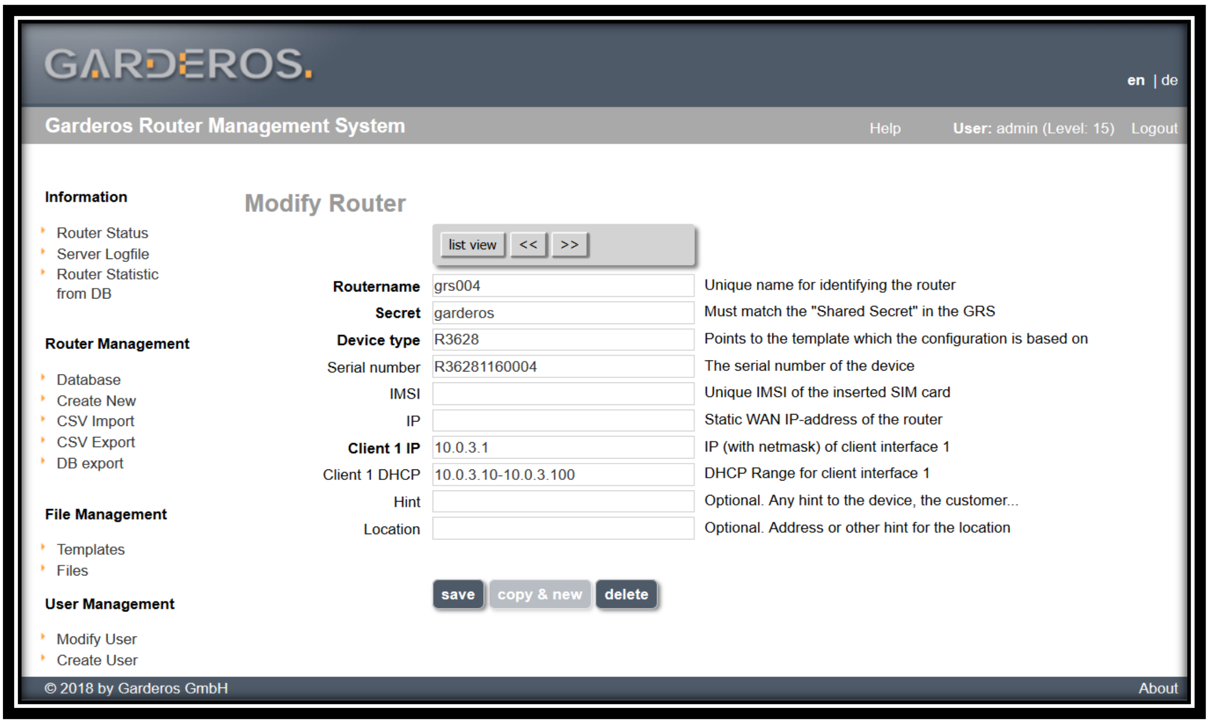
While the database is delivered with a default database schema, the schema can be enhanced and any required value can be stored in the database for later use in the configuration templates.

Apart from the router name as standard identifier, the routers can also be identified by serial number, MAC address, public IP, IMSI or SIM ID of the inserted SIM card (3G/4G-Routers only).

Administration Servlet

In many cases routers and router locations are already managed by an existing inventory management system or asset management system. In this case the Garderos Configuration Server is operated without the Administration Servlet and integrated with an existing database e.g. which the customer might already have as part of their IT infrastructure.

In a so-called greenfield scenario the Administration Servlet allows the adding, changing and removal of router configurations from the Garderos Configuration Server's database using a web-based GUI (see Fig. 3), thus requiring almost no integration effort.



Routername	grs004	Unique name for identifying the router
Secret	garderos	Must match the "Shared Secret" in the GRS
Device type	R3628	Points to the template which the configuration is based on
Serial number	R36281160004	The serial number of the device
IMSI		Unique IMSI of the inserted SIM card
IP		Static WAN IP-address of the router
Client 1 IP	10.0.3.1	IP (with netmask) of client interface 1
Client 1 DHCP	10.0.3.10-10.0.3.100	DHCP Range for client interface 1
Hint		Optional. Any hint to the device, the customer...
Location		Optional. Address or other hint for the location

Fig. 3: Screenshot of the GCS web-based GUI

Integration and Configuration

Initial setup of the Garderos Configuration Server on an existing application server is fast and easy. Because not all customers who want to use the Garderos Configuration Server already have an application server available, or do not want to run the Garderos Configuration Server on their existing machines, Garderos offers all services required to get the Garderos Configuration Server up and running: (see Fig. 4)

- Requirement analysis
- Network planning
- Installation
- Configuration and setup of the initial configuration files
- Support

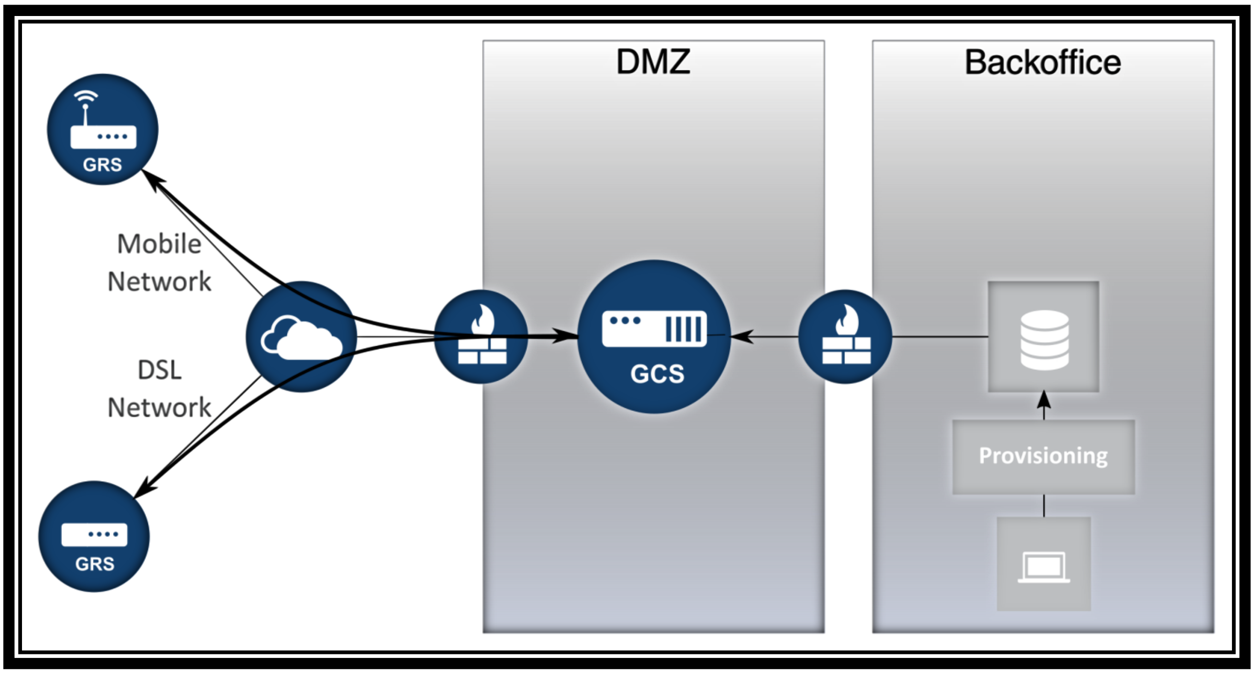


Fig. 4: Typical network setup

Security Considerations

Configuration data of the routers should be well protected to prevent unauthorized access to your network. State-of-the-art security requires asymmetrical encryption mechanisms to protect data exchanged via the Internet. The Garderos Configuration Server supports configuration file download by HTTPS with 2-way (client and server certificate) authentication and certificate revocation lists based on OCSP.

Like any other web-based service, the Garderos Configuration Server should be protected by a firewall in addition to the previously mentioned security measures.

Garderos takes your security considerations into account when setting up a Configuration Server in your network.

System Requirements

Number of routers per server	10,000 *)
Supported OS	Ubuntu Server 22.04 (recommended), Ubuntu Server 20.04, CentOS 7/8
Application server (depending on OS)	Apache Tomcat 9 plus corresponding Java
DBMS	MySQL or MariaDB
Minimum Space on HD	100 GB
Minimum RAM	8 GB

*) The number of supported routers depends on the functions used. Functions can be switched on and off in the servlet configuration and functions can be used for part of the routers only.

Features

Router recognition based on name, MAC, serial number, IP, IMSI or SIM ID
Garderos router authentication by: - Hashed secret - User-Agent - Certificate
Dynamic creation of configuration files
HTTPS secured data transfer
Web based GUI - Role based administrator rights - Administrator authentication in local database or by RADIUS
Easily customizable administration pages
Router monitoring: - View active routers - View dynamic router configuration files - Collect and show router statistics *)
Supports all GRS versions in a mixed setup
CSV import & export
Mass rollout of firmware updates
Certificate file and script distribution
Integration with syslog server possible
Redundancy and load balancing
Client and server certificates
OCSP support

*) Limited to up to 2000 database entries per day, higher numbers can have a significant performance impact. SNMP management system for higher numbers recommended (e.g. PRTG).