Author:	Date:	Revision:	Document no:
AK, JAS, ØBL	06.03.2023	J	RMD-010

System architecture, RoomMate

1 INTRODUCTION

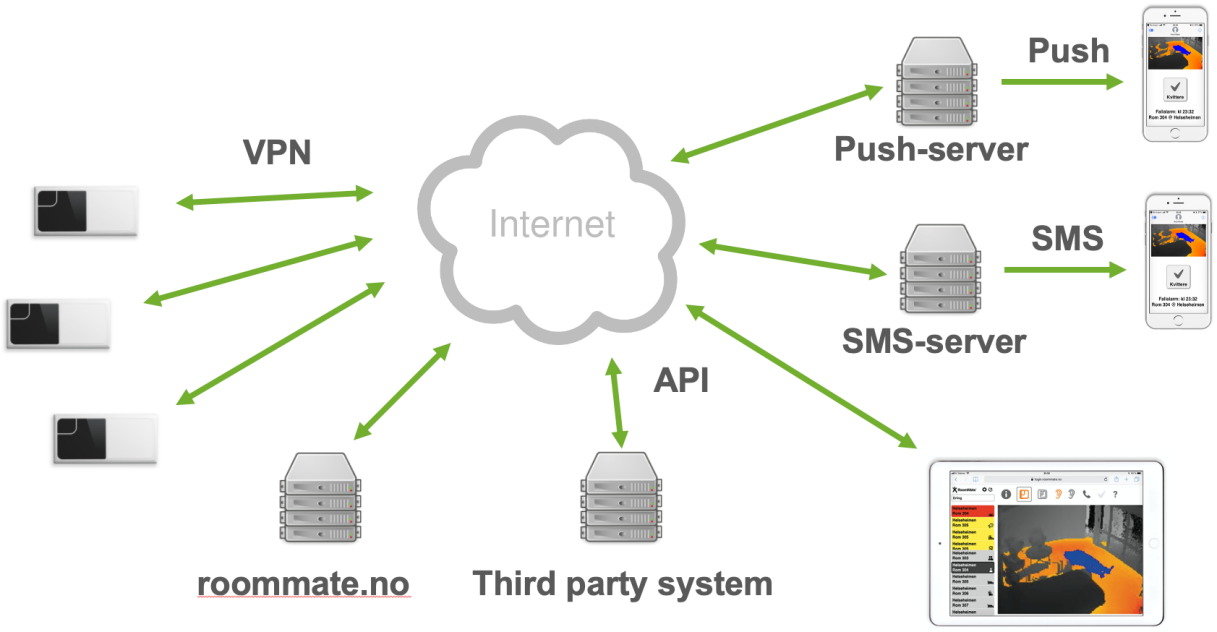
This document provides a concise overview of the system architecture and data flow in the RoomMate system.

In addition to this document, reference is made to other documentation for the RoomMate system. In particular, the following documents may be of interest:

- RMD-001, *"Installation Guide RoomMate"*
- RMD-009, *"RoomMate API"*

2 SYSTEM ARCHITECTURE

The overall system architecture is shown in the figure below



On the left are the RoomMate sensors that are deployed with the endusers. The RoomMate sensors communicate via an encrypted VPN channel with a central RoomMate server *("roommate[x].no").* This server is a physical server owned by RoomMate AS and located in a secure professional data center in Oslo. The RoomMate sensors have their own built-in computer and this is where data processing is done to detect critical situations. Information is therefore only sent to the roommate server when supervision is carried out or when an alarm or notification is to be sent.

RoomMate has several production servers that use different IP-addresses, and the users of the RoomMate system must open an IP range that allows RoomMate to balance and distribute load between them. This applies both to sensors and web clients and apps.

Alarms and alerts can be, for example, that a person falls. These alarms are sent to the user as push messages to their own RoomMate app or as SMS (illustrated at top right). The user can then also

choose to log in to the roommate[x].no server to carry out an inspection as shown on the iPad at the bottom right. Login is done with username, password and a separate code (two-factor authentication) as well as the option for IP-filtering. Code for two-factor authentication is sent via SMS or push message to the app.

If the RoomMate server is not available due to an error, the RoomMate sensors themselves can send alarms and notifications via SMS from the SMS server. Supervision is not possible if the RoomMate server is not available.

In addition to using RoomMate as a stand-alone system as described above, it is also possible to integrate the RoomMate system into an external system for alarm management. This is shown as a "Third party system" in the figure above. A separate API for this has been implemented on the RoomMate server and all communication between the third-party system and the RoomMate system goes via the RoomMate server.

2.1 App

RoomMate AS has developed a separate app for iOS and Android. The app is available for free in the Apple App store and Google Play. Using an app instead of SMS has a number of advantages:

- Messages delivered faster
- Better user interface
- Easier login for supervision
- Better control over sound notification with own sounds for notifications and alarms

The app will be used for all new installations and will eventually take over for SMS.

Notifications, alarms and code for two-factor authentication are sent as push messages via an external push server (Google Firebase). Data flow for app use is as follows:

Use the app:

1. In the administration system of RoomMate, a 10-digit code can be retrieved for each user of the system.
2. When the app is first started, it will prompt for the code.
3. The app automatically retrieves a "push token" from the push server (https).
4. The app sends code and the push token to the RoomMate server (https)
5. The RoomMate server generates a unique app ID which is sent to the app (https). The app ID is used for all further communication between the app and the RoomMate server. The App ID has a limited lifetime before it is replaced.

Send notification or alarm to the app:

1. The RoomMate server sends a push message via the push server (https). The push message contains texts describing the event as well as information about the notification sound.
2. When the app is opened, it retrieves information about outstanding notifications and alarms from the RoomMate server (https). This information includes, among other things, a link corresponding to the one communicated via SMS.
3. If the RoomMate server sees that the lifetime of the App ID has expired or is about to expire, it will first update to a new App ID. This happens as a push message to make sure it reaches the correct recipient.

The app also contains a button for direct login into the system. When using this, the two-factor code will be handled in the background without user intervention. This is because the system already knows that the user has the phone in question in hand.

3 NETWORK REQUIREMENTS

From and including December 2022, new customers or customers who migrate from roommate.no or do not have any sensors from before must be connected to roommate4.no.

As of 2023 RoomMate are using the following domains for the servers:
roommate.no, roommate2.no, roommate3.no, roommate4.no, roommate5.no, roommate6.no, roommate7.no, roommate8.no, roommate9.no

The entire range 82.199.16.0/24 belongs to RoomMate and for the network openings it is desirable that RoomMate's IP addresses are used where they are provided. If for some reason hostnames restrictions are needed, then as a minimum the domain names **roommate.no** and **roommate[*].no** with all of its sub domains needs to be allowed.

[*] = an integer for the actual server a customer is installed on. Customers connected to, for example, roommate3.no who are to have new sensors or more locations continue on the roommate3.no server etc.

RoomMate sensors have the following network traffic:

Destination	Dest. Port	Protocol	Purpose
82.199.16.0/24 Subdomain(s): vpn	1194	OpenVPN over UDP	Primary VPN tunnel for all communication with the roommate server.
82.199.16.0/24 Subdomain(s): vpn	443	VPN over TCP	Secondary VPN tunnel for all communication with the roommate server.
82.199.16.0/24	80	HTTP	Monitoring if port 80 is open. (Not required for operation)
Local NTP server provided by DHCP And/Or ntp.ubuntu.com 0.no.pool.ntp.org 1.no.pool.ntp.org 2.no.pool.ntp.org 0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org	123	NTP	Clock synchronization
Local DNS server provided by DHCP And/Or 8.8.8.8 8.8.4.4 208.67.222.222 208.67.220.220	53	DNS over TCP & UDP	Domain resolving
ipv4.connman.net	80	HTTP	To know if it has access to internet
smtp.rmate.no	587	SMTP	To send email (only required for email)
sveve.no www.sveve.no	443	HTTPS	To send SMS (only required for SMS)

It might be problematic connections if the sensors are installed in a subnet or would try to reach a local NTP, DNS or DHCP server in the following ranges depending on the RoomMate server.

Server	IP
roommate2.no	10.11.0.0/16
roommate3.no	10.12.0.0/16
roommate4.no	10.14.0.0/16
roommate5.no	10.15.0.0/16

If the IP addresses makes a conflict with existing infrastructure, please inform support before installation.

A sensor needs at least 100 kB/s bandwidth in both directions for operation and maintenance. During updates a sensor could download more than 100MB of data, which should not be blocked or dropped.

The RoomMate app are using the phone's web browser and the traffic from the RoomMate app and a browser on a computer would be the following network traffic.

Destination	Dest. Port	Protocol	Purpose
82.199.16.0/24 Subdomain(s): app	80 443	HTTP HTTPS	RoomMate app
82.199.16.0/24 Subdomain(s): login, www	80 443	HTTP HTTPS	Web browser, login
82.199.16.0/24 Subdomain(s): voip	3478	SIP over UDP	Web browser, SIP calls for one-way and two-way speech
82.199.16.0/24 Subdomain(s): voip	50000-53999	RTPS over UDP	Web browser, one-way and two-way speech stream
82.199.16.0/24 Subdomain(s): turn	5349	TCP & UDP	Web browser, Traversal using relays around NAT
82.199.16.0/24 Subdomain(s): voip	443	WSS (WebSocket)	Web browser, web socket for one-way and two-way speech
82.199.16.0/24	443	HTTPS	Web browser, RoomMate portal

To receive notifications from the RoomMate app, then the notification services form Apple and/or Google needs to work. The following network flow could be found regarding their services:

Destination	Dest. Port	Purpose
17.0.0.0/8	443 2197 5223	Apple Push Notification Service *Additional information about push messages for iOS can be found here:* *https://support.apple.com/en-us/HT203609*
mtalk.google.com mtalk4.google.com mtalk-staging.google.com mtalk-dev.google.com alt1-mtalk.google.com alt2-mtalk.google.com alt3-mtalk.google.com alt4-mtalk.google.com alt5-mtalk.google.com alt6-mtalk.google.com alt7-mtalk.google.com alt8-mtalk.google.com android.apis.google.com device-provisioning.googleapis.com firebaseinstallations.googleapis.com	443 5228* 5229* 5230*	Firebase Cloud Messaging for push notifications ** "If your network implements Network Address Translation (NAT) or Stateful Packet Inspection (SPI), implement a 30 minute or larger timeout for our connections over ports 5228-5230. This enables us to provide reliable connectivity while reducing the battery consumption of your users' mobile devices."* Additional information about push messages for Android can be found here: *https://firebase.google.com/docs/cloud-messaging/concept-options#messaging-ports-and-your-firewall*