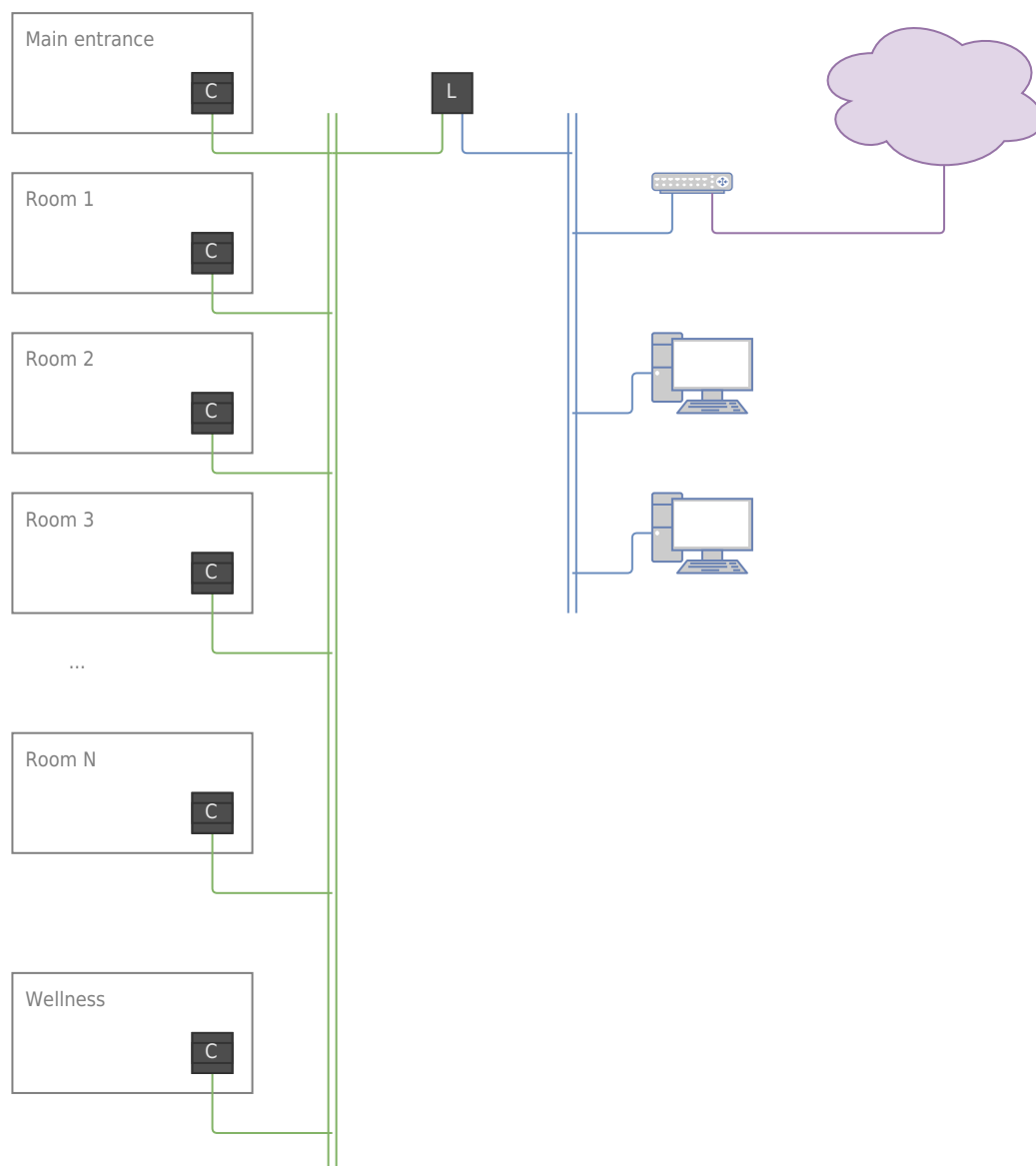


# Access

Controllers [C] manage access and other automation functions in hotel rooms and other spaces. Typically, there is one controller per room/space.

The Linker [L] (a dedicated mini Linux server) allows the management of access codes and ensures connection to the outside world through a standardized API.



Each access point has its own unique identification (AP\_ID), assigned by the system integrator during system setup.

All access codes are entered into the controllers via the Linker. Guest codes are entered into room controllers, while staff and general codes are entered into the common access controller. Each controller can hold up to 100 access codes. Every code has a validity period (date, time from - to) and access rights for all necessary access points in the system.

<b>Code</b>	32-bit signed (-2147483648..2147483647)
<b>Label</b>	text (can be empty)
<b>Type</b>	1 = Guest 2 = Staff 3 = Security
<b>Valid from</b>	UTC time rounded to 10 min or unlimited
<b>Valid to</b>	UTC time rounded to 10 min or unlimited
<b>Access rights</b>	up to 100 × (AP_ID, pass); pass = 1..254 or unlimited)

The controller reads a code on the connected code reader. Many kind of code readers are supported:

- QR code reader
- RFID code reader (EM 125kHz)
- Mifare code reader
- PIN keypad
- Some readers allows combination (QR + RFID)

If the scanned code is found in the local list, controller verifies whether the code has access rights (valid time and permission for the access point). If valid, the controller unlocks the door and decreases the remaining allowed accesses for codes with restricted entries.

If the code is not on the local list, the controller sends an access request over the local network. The controller holding the code checks the permission for entry, and if allowed, it sends a positive response. The requesting controller then unlocks the door and sends a request to decrement the remaining access count to the controller with the code in its list.

Codes that expire or have no remaining permissions are automatically deleted from the controller.

## Example

Hotel with 3 rooms, main entrance and wellness.

Access point	AP_ID	access_code_controller_id
Main entrance	01	room_01_access
Room 1	10	room_02_access
Room 2	11	room_03_access
Room 3	12	room_04_access
Wellness	70	room_05_access

### Code for "Guest 1"

- code = 1234
- code type = 1 (guest)
- from 03.08.2024 15:00 (SI time zone: +2)
- to 10.08.2024 10:00 (SI time zone: +2)
- for main door (unlimited passes)
- for Room 2 (unlimited passes) ⇒ to controller for Room 2
- 10 accesses to wellness

Can be added with POST to **/api/v1/room/room\_03\_access/code** with following data in JSON format

```
{
  "code": 1234,
  "label": "Guest 1",
  "code_type": 1,
  "valid_from": "2024-08-03T13:00:00.000Z",
  "valid_to": "2024-08-10T08:00:00.000Z",
  "access": [
    {
      "access_point": 1,
      "remaining_passes": null
    },
    {
      "access_point": 11,
      "remaining_passes": null
    },
    {
      "access_point": 70,
      "remaining_passes": 10
    }
  ]
}
```

## Code for "Staff 1"

- to controller for Main entrance
- code = 3141
- code type = 2 (staff)
- from = unlimited
- to = unlimited
- for main door (unlimited passes)
- for all Rooms (unlimited passes)
- for wellness (unlimited passes)

Can be added with POST to **/api/v1/room/room\_01\_access/code** with following data in JSON format

```
{
  "code": 3141,
  "label": "Staff 1",
  "code_type": 2,
  "valid_from": null,
  "valid_to": null,
  "access": [
    {
      "access_point": 1,
      "remaining_passes": null
    },
    {
      "access_point": 10,

```

```
    "remaining_passes": null
  },
  {
    "access_point": 11,
    "remaining_passes": null
  },
  {
    "access_point": 12,
    "remaining_passes": null
  },
  {
    "access_point": 70,
    "remaining_passes": 10
  }
]
}
```