

Overview

Sensaru Cloud uses a microservice architecture. In a nutshell every service follows the Unix philosophy of "Do One Thing and Do It Well". There is a controlling service called "C1 Core". Every other microservice is connected to C1 Core. Every microservice must have the following features:

- It must be horizontal scalable.
- It must scale linearly (or close to) in terms of data throughput and the number of requests per second.
- It must implement high-availability so there is no single point of failure.
- It must be able to run in multiple data centers in parallel.

System layout

Let's start with C1 Core. C1 core is the central point within Sensaru Cloud. All requests must go through C1 Core and every client must register to it. It manages users and edge clients, performs authorization checks and publishes events:

```
erDiagram
    C1_Core
```

There can be one or more C1 Core instances spanning multiple datacenters. C1 Core does only provide the backend. The user accesses C1 Core using C1 Core UI:

```
erDiagram
    C1_Core_UI ||..|{ C1_Core : "accesses"
```

The one to many relationship in this case means that one C1 Core UI instance is associated with multiple C1 Core backend instances - with every request another instance might be accessed.

For security reasons the user credentials reside in a separate service: C1 Auth. C1 Auth is the Sensaru Cloud's authentication server (authorization is done in C1 Core). It currently supports multi-tenant capable OAuth. Like this an email address must only be unique per tenant and a user can register with multiple tenants using the same email address. C1 Auth also has an UI:

```
erDiagram
    User ||..|{ C1_Core_UI : "uses"
    User ||..|{ C1_Core_UI : "logs in"
    C1_Core_UI ||..|{ C1_Core : "accesses"
    C1_Core_UI ||..|{ C1_Core : "accesses"
    C1_Core }|..|{ C1_Core : "accesses"
```

Sensaru Cloud backend modules are connected to C1 Core:

```
erDiagram
    User ||..|{ C1_Core_UI : "uses"
    User ||..|{ C1_Core_UI : "logs in"
    C1_Core_UI ||..|{ C1_Core : "accesses"
    C1_Core_UI ||..|{ C1_Core : "accesses"
    C1_Core }|..|{ C1_Core : "accesses"
    Module_1 }|..|{ C1_Core : "connected to"
    Module_2 }|..|{ C1_Core : "connected to"
```

Modules can provide their own user interfaces. Modules can also have module clients (e. g. edge clients are connected through the module C1 Proxy).

Also an ERP system can be connected to the Sensaru Cloud, for example the C1 Wodis Connector:

At last end users access the Sensaru Cloud through a smartphone webapp. This app is also connected to C1 Core. Modules can provide content for the webapp as well (native apps use the REST API and don't need an additional app backend):

```
erDiagram
    User ||..|{ C1_Core_UI : "uses"
    User ||..|{ C1_Core_UI : "logs in"
    C1_Core_UI ||..|{ C1_Core : "accesses"
    C1_Core_UI ||..|{ C1_Core : "accesses"
    C1_Core }|..|{ C1_Core : "accesses"
    Module_1 }|..|{ C1_Core : "connected to"
    Module_1_UI }|..|{ Module_1 : "accesses"
    Module_1_app_backend }|..|{ Module_1 : "accesses"
    Module_2 }|..|{ C1_Core : "connected to"
    C1_Proxy }|..|{ C1_Core : "connected to"
    C1_Core_UI }|..|{ Module_1_UI : "integrates"
    Edge_client ||..|{ C1_Proxy : "connected to"
    C1_Wodis_Connector ||..|{ C1_Core : "synchronizes with"
    C1_Core_app_backend }|..|{ C1_Core : "connected to"
    C1_Core_app_backend }|..|{ Module_1_app_backend : "integrates"
    Smartphone_app ||..|{ C1_Core_app_backend : "accesses"
```