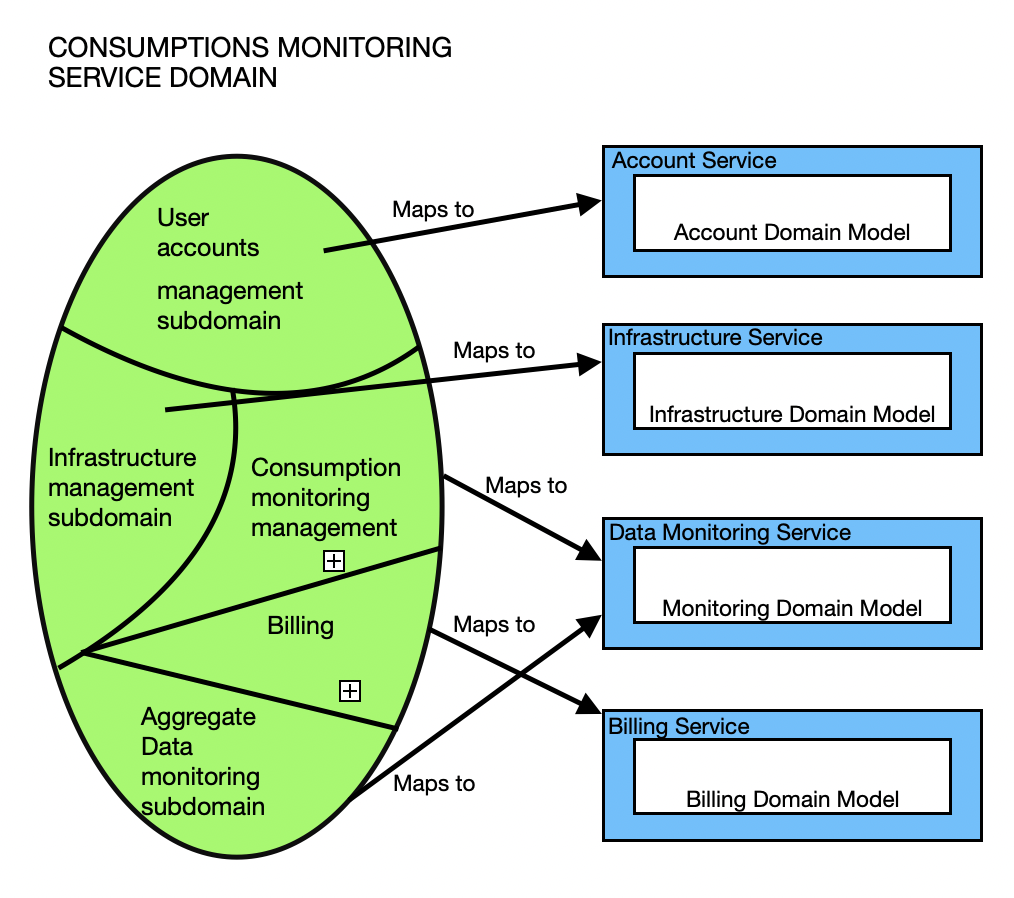
**Software Architecture description**

This doc contains a description of the Software Architecture of the proposed solution.

The software architecture is designed upon the bounded contexts and the context map described in related docs.

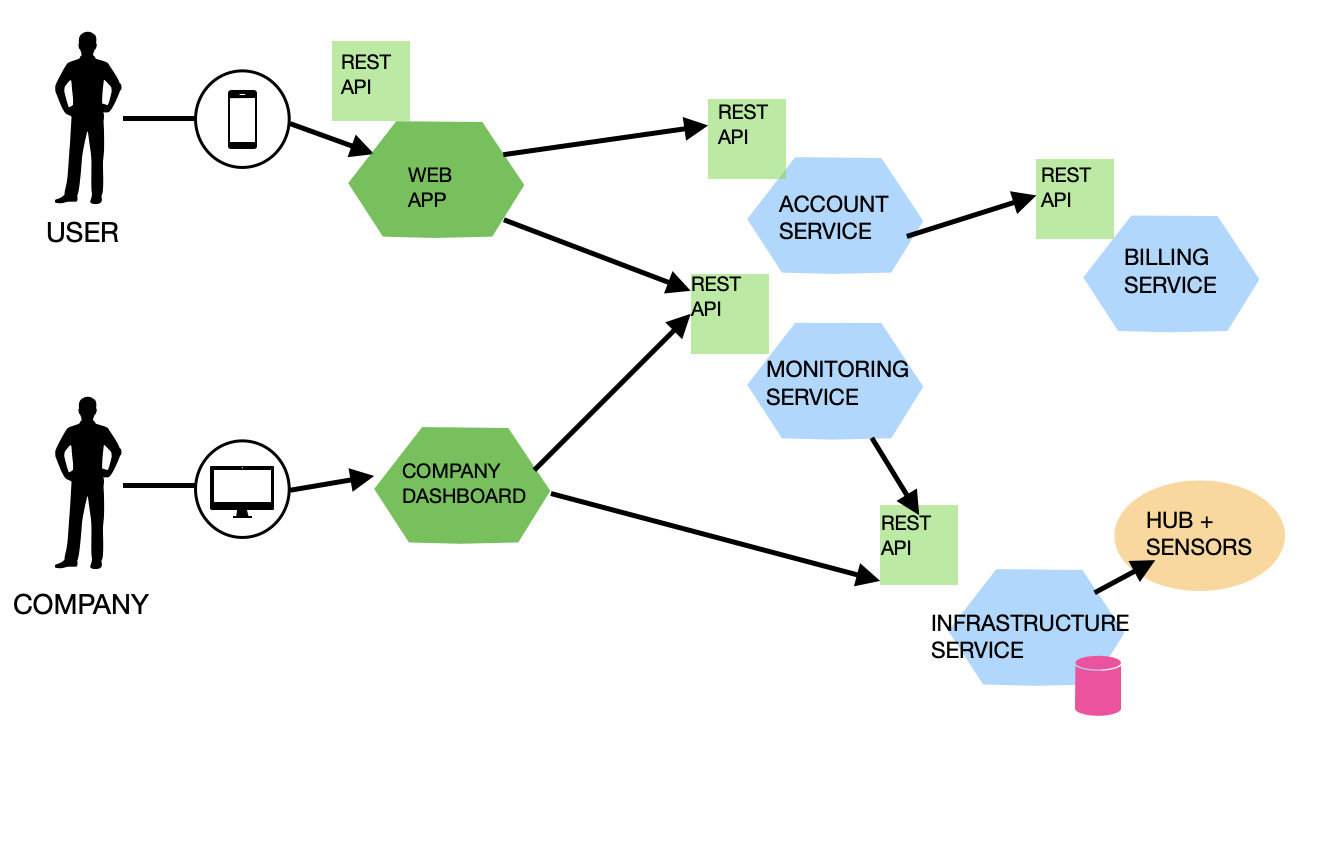


As is possible to see from the above representation that map Subdomains in Microservices, our Sw is composed by four different microservices corresponding to the Bounded Contexts previously identified:

* **Account Service**
* **Infrastructure Service**
* **Monitoring Service**
* **Billing Service**

Once we identify the microservices of our Software we can move to the representation of the Sw Architecture. In order to do so, we make use of three main kinds of software components:

* Backend microservices
* Frontend applications
* Embedded system components



BACKEND MICROSERVICES

The proposed architecture is based on four main backend microservices, directly mapping the identified bounded contexts and all designed upon the Ports & Adapters / Hexagonal pattern. Some mentions are done also about the domain model and the APIs.

* **Account Service** 
  + this microservice concerns the management of the accounts of the monitoring service
    - keeping track of the current accounts
    - registering a new user (account)
    - get information about an existing account
  + Domain model
    - account – entity
    - accounts – aggregate
  + The domain model is mapped into a RESTful architecture
    - account and accounts are represented by resources
  + API - RESTful model
    - To register a new user
    - To get all the accounts
* **Infrastructure Service** 
  + This microservice concerns the management of the infrastructure in charge of collecting and gathering the data to monitor.
  + Domain Model
    - Sensor dt - entity
    - Sensors - aggregate
    - Hub dt - entity
* The domain model is mapped into a RESTful architecture
  + each sensor is represented as a REST resource
* API
  + to register a new sensor
  + to lookup for an existing sensor
  + to get to sensor state (DT)
  + to sync sensor DT with its physical twin
    - to notify events to the main hub
* **Monitoring Service** 
  + this microservice concerns the management of the monitoring service
    - keeping track of the current level of consumptions
    - Filtering, selecting and visualizing data
    - Monitoring level of expenses
  + Domain model
    - account – entity
    - accounts – aggregate
  + The domain model is mapped into a RESTful architecture
  + API
    - to get general information about overall consumption of the user
    - to get information on the specific utilities (gas, electricity, water)
* **Billing Service**
  + this microservice concerns the management of the billing service
    - Producing billings.
    - Visualizing billings.
    - Paying billings.
  + Domain model
    - billing – entity
    - billings – aggregate

* + API
    - to get all billings

FRONT-END APPLICATIONS

* **User app**
  + To register.
  + To visualize ongoing and past consumption data.
  + To monitor billing situations.
* **Company dashboard**
  + To get aggregate data about users' consumptions.
  + To monitor single users' consumption.
  + To register a new sensor.
  + To monitor the state of the sensors.

EMBEDDED SYSTEM COMPONENTS

* **Hub**
  + The Hub is a processor that runs software able to do ETL operations on data.