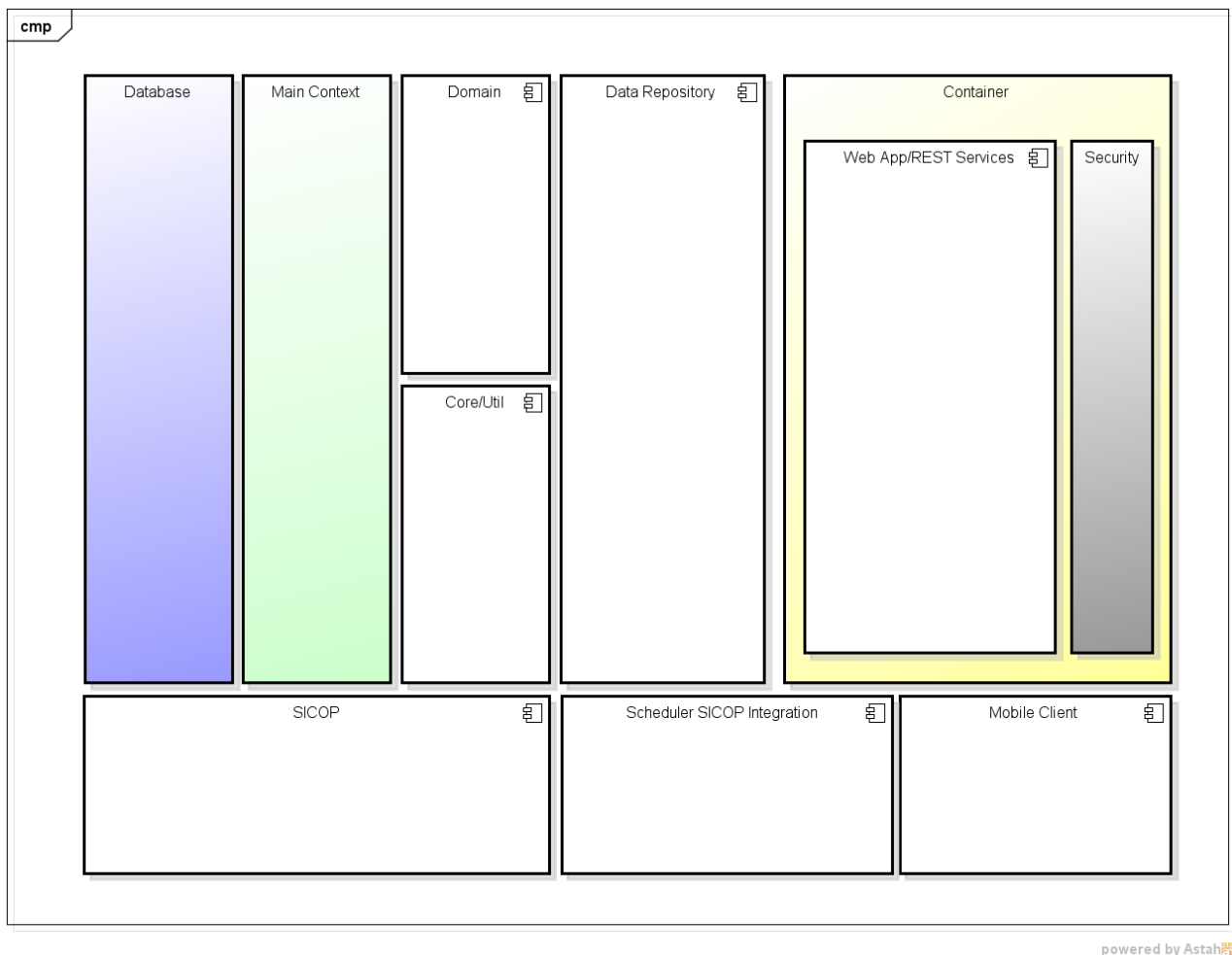


# Esboço de Arquitetura de Componentes/Módulos



## Database

Base de dados da parte servidora, independente de SGBD.

## Main Context

Contexto da aplicação servidora. Responsável por IoC e demais configurações básicas para execução da aplicação servidora.

## Domain

Classes de domínio, como entidades JPA e DTOs.

## Core Util

Funcionalidade utilitárias para uso da aplicação, caso faça-se necessário.

## Data Repository

Repositórios para manipulação dos dados. Usado **Spring Data JPA**. O contexto do dado estaria configurado aqui, para melhor organização.

## Container

### Web App/REST Services

Contexto WEB com os serviços REST (usando **Spring MVC**) e, possivelmente, cliente WEB. No cliente WEB, poderia ser usado JSP com JQuery e Twitter Bootstrap.

### Security

Configuração da segurança da aplicação, tanto serviços quanto cliente WEB. Usado **Spring Security**.

## SICOP

Legado.

## Scheduler SICOP Integration

Scheduler de integração com o legado. Uso do **Spring Integration** para integração com este legado. Tal scheduler pode estar em um contexto standalone ou no web, a depender dos requisitos não funcionais.

## Mobile Client

Aplicativo móvel com as funcionalidades previstas para CRUD de dados. Por facilidade, o **Spring Android** poderia ser usado, por já possuir REST templates e um Auth Module que facilitariam o desenvolvimento da integração com os REST services existentes.

## Observações

Caso **Spring** seja mesmo adotado, o **Spring Integration Test** é indicado para os testes, tanto de unidade quanto de integração. Com o uso do **Spring Integration Test**, os contextos **Spring** são disponibilizados, o que facilita a escrita dos testes.