Kanban

To do | 4	In progress | 1	Done | 2
Diagrama de componentes C4	Diagrama de contenedores C4	Creación del miro de trabajo
Diagrama de código C4	+	Diagrama de contexto C4
Diagrama de clases		+
Diagrama modelo de datos		
+		

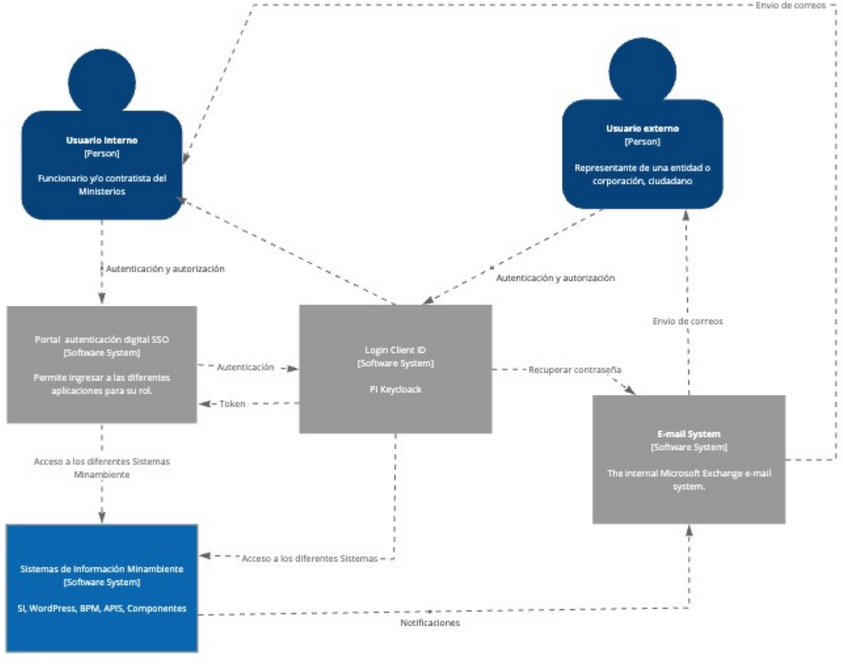


Diagrama de contexto ecosistema Minambiente

Contiene la arquitectura de referencia con BPM

Última fecha de modificado: 2024-04-08

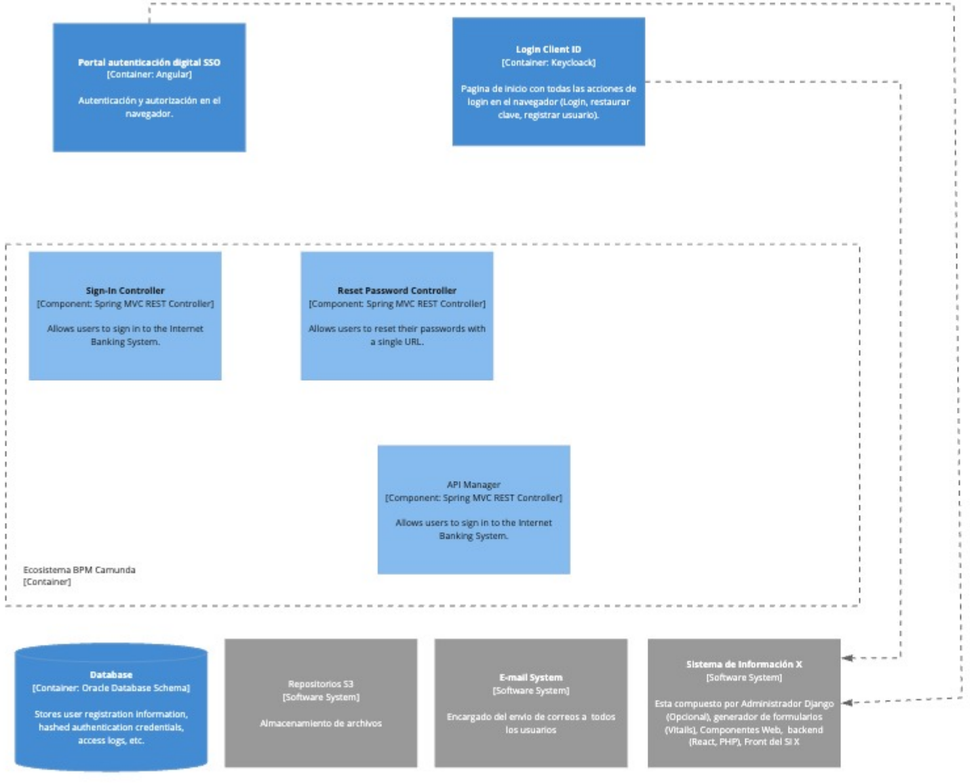


Diagrama de componentes Aplicaciones, APIS, Servicios

Contiene la arquitectura de referencia con BPM

Última fecha de modificado: 2024-04-08

Welcome to the C4 Architecture Model Miro template!

The C4 model is an approach to architectural diagramming that uses four levels of abstractions:

- **Context**: the highest level view of a software system, a starting point showing the system in the context of the world
- **Container**: the applications and data stores that make up the software system
- **Component**: a grouping of related functionality encapsulated behind a well-defined interface within a Container
- **Code**: the implementation details of a Component, as shown in UML class diagrams, entity relationship diagrams, etc. Code diagrams are optional and typically auto-generated by development tooling. An example is not included here.

Instructions for Use of this Miro template

1. Start by adapting the Context diagram to your software system: add actors (persona, devices, etc.), external systems, and the name and description of your software system. Label relationships with their intent and technology/protocol.
2. Adapt the Container diagram to your software system: add containers for applications, data stores, and other subsystems within the system block. Be sure to re-use actors and external systems from the Context diagram.
3. Copy and adapt the Component diagram for each container, at least those that require further explanation. Be sure to re-use dependent containers and external systems from the Container diagram.

Source

The C4 Model was created by Simon Brown, as documented at https://c4model.com/.

License

The C4 Model is licensed under a Creative Commons Attribution 4.0 International License.

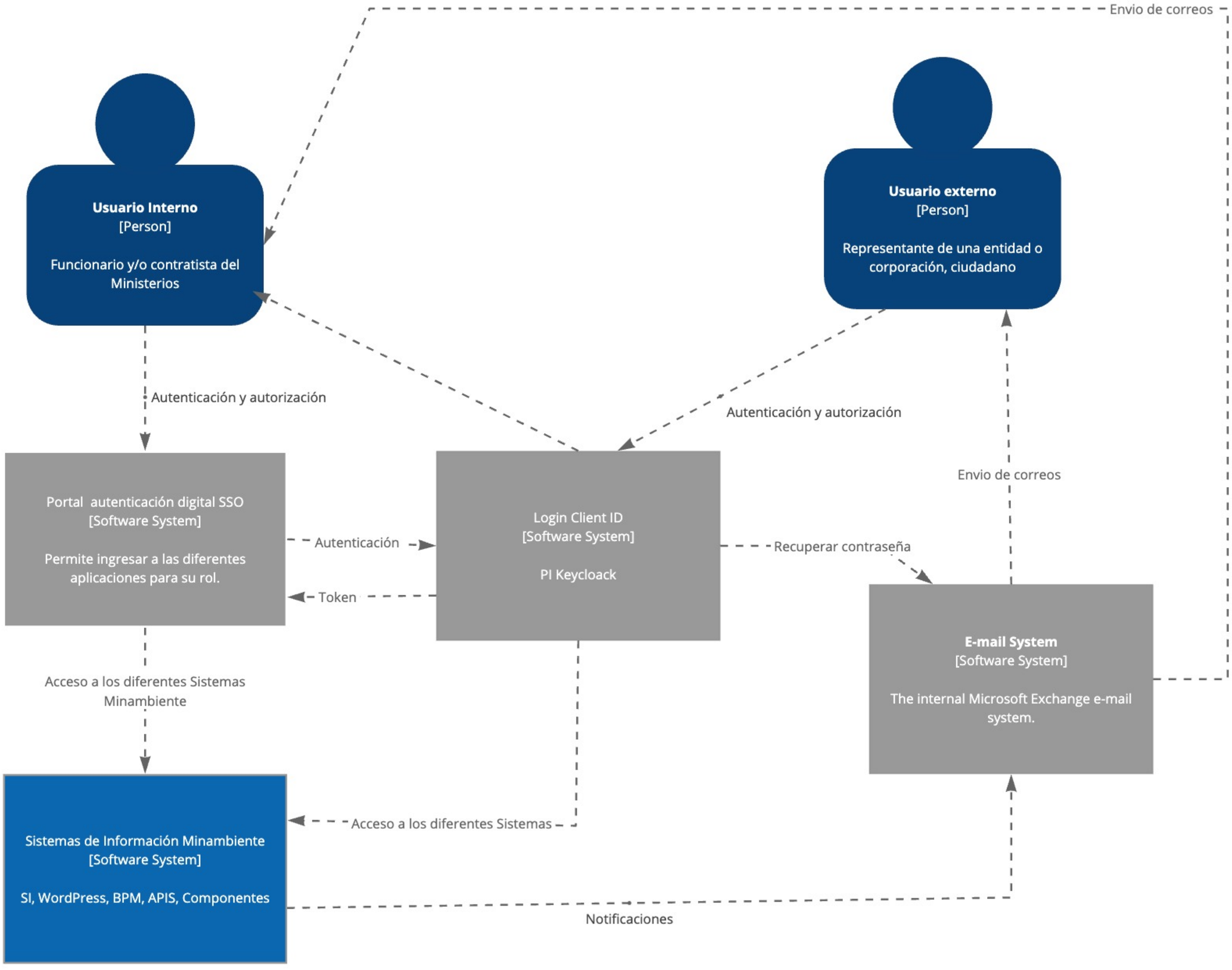
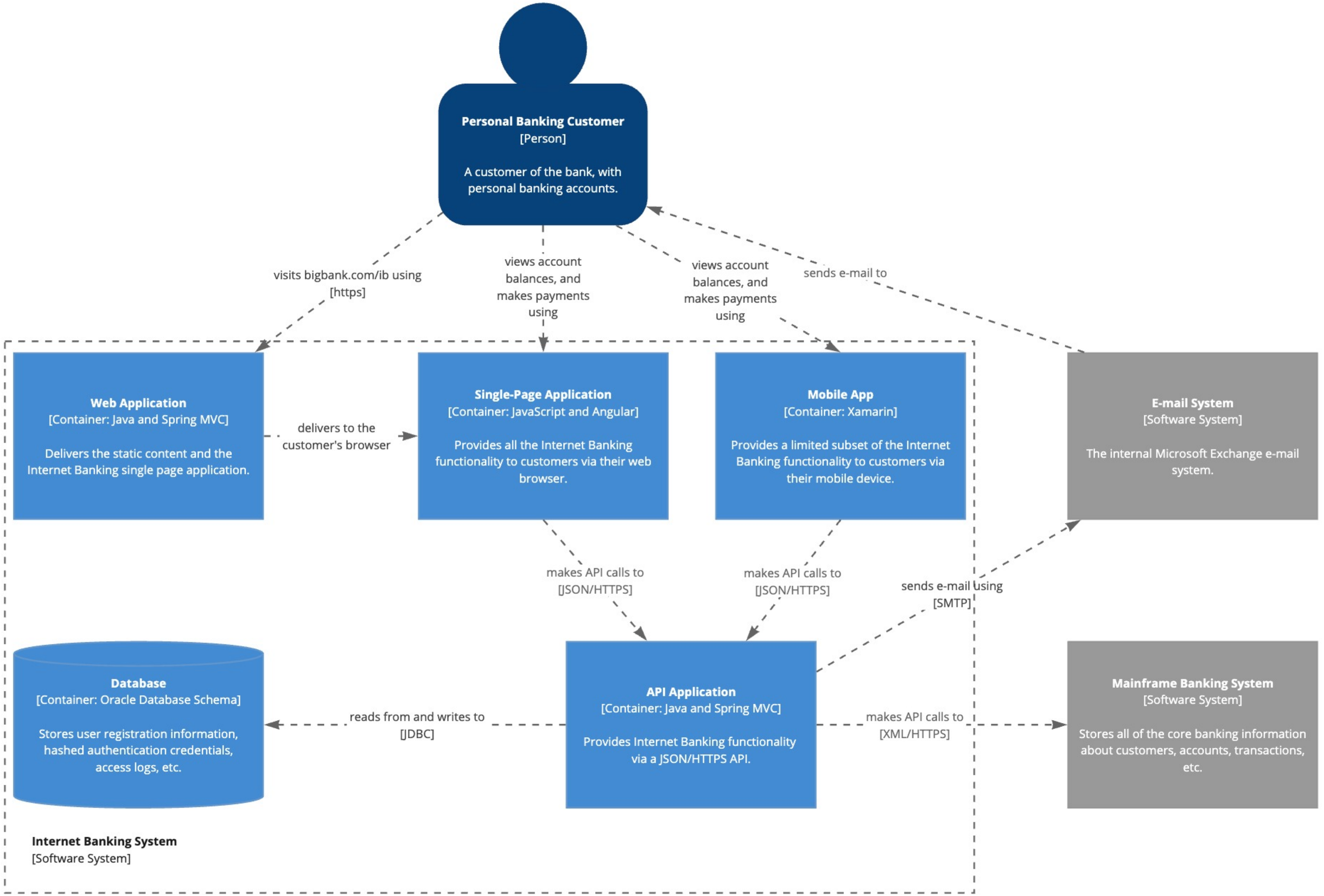


Diagrama de contexto ecosistema Minambiente

Contiene la arquitectura de referencia con BPM

Última fecha de modificado: 2024-04-08



Container Diagram for Internet Banking System

The container diagram for the Internet Banking System

Last modified: 2022-02-10

Portal autenticación digital SSO
[Container: Angular]

Autenticación y autorización en el navegador.

Login Client ID
[Container: Keycloack]

Pagina de inicio con todas las acciones de login en el navegador (Login, restaurar clave, registrar usuario).

Sign-In Controller
[Component: Spring MVC REST Controller]

Allows users to sign in to the Internet Banking System.

Reset Password Controller
[Component: Spring MVC REST Controller]

Allows users to reset their passwords with a single URL.

API Manager
[Component: Spring MVC REST Controller]

Allows users to sign in to the Internet Banking System.

Ecosistema BPM Camunda
[Container]

Database
[Container: Oracle Database Schema]

Stores user registration information, hashed authentication credentials, access logs, etc.

Repositorios S3
[Software System]

Almacenamiento de archivos

E-mail System
[Software System]

Encargado del envio de correos a todos los usuarios

Sistema de Información X
[Software System]

Esta compuesto por Administrador Django (Opcional), generador de formularios (Vitails), Componentes Web, backend (React, PHP), Front del SI X

Diagrama de componentes Aplicaciones, APIS, Servicios
Contiene la arquitectura de referencia con BPM
Última fecha de modificado: 2024-04-08

1. hu-otic-gestion-solicitudes-001 - Inicio de sesión

Usuarios Externos entran por el flujo normal de convocatoria (Autoridades ambientales, ciudadando etc)
Usuarios Internos entra por el portal unico (Funcionarios y contratistas de Minambiente)
Componente de autenticación Keycloack
Base de datos keycloack postgres

2. hu-otic-gestion-solicitudes-002 - Pantalla principal

API de autorización (Roles y permisos)
Front Portar único de entrada x aplicaciones (Usuarios internos), incluimos por en un fututo proceso como una vista adicional???
Nodo aplicacion (VITAL, RENARE, Convocatorias etc)

3. hu-otic-gestion-solicitudes-003 - Menú gestión de información

Cada sistema debe incorporar un botón volver a portal, no se agrega ningun componente adicional

4. hu-otic-gestion-solicitudes-004 - Menú - Mis tareas

Componente de auditoria
Componente de historico
Componente de errores
https://github.com/camunda/docker-camunda-bpm-platform/blob/next/test/docker-compose.yml
Mysql, Postgres, repo de librerias, tomcat y apache
Elastic
Base de datos de Camunda
Base de datos de l metamodelo de BPM Minambiente
Componente de visualización del proceso???

1. Asignar tarea a un rol especifico
2. Listar Roles asociados a un proceso
3. Listar tareas sin finalizar con filtrado por proceso, rol, usuario, fecha desde y hasta, paginado, radicado
4. Listar tareas finalizadas con filtrado por proceso, rol, usuario, fecha desde y hasta, paginado, radicado
5. Consultar estado tarea (Los posibles estados los define el proceso)
6. Cambiar estado tarea (Para cambiar el estado debe mostrarse un formulario propio de la aplicación Vital, Renare, Convocarorias
7. Ver en que lugar me encuentro del procesos, es posible representarlo gráficamente con un servicio y un componente de visualización?
8. Visualizar histórico del proceso

Portal autenticación digital
[Container: Angular]

Portal SSO de inicio con todas las acciones de autenticación y autorización en el navegador.

Login Client ID
[Container: Keycloack]

Pagina de inicio con todas las acciones de login en el navegador (Login, restaurar clave, registrar usuario).

makes API calls to
[JSON/HTTPS]

makes API calls to
[JSON/HTTPS]

makes API calls to
[JSON/HTTPS]

makes API calls to
[JSON/HTTPS]

makes API calls to
[JSON/HTTPS]

API Application
[Container]

Sign-In Controller
[Component: Spring MVC REST Controller]

Allows users to sign in to the Internet Banking System.

Reset Password Controller
[Component: Spring MVC REST Controller]

Allows users to reset their passwords with a single URL.

Accounts Summary Controller
[Component: Spring MVC REST Controller]

Provides customers with a summary of their bank accounts.

uses

uses

uses

uses

Security Component
[Component: Spring Bean]

Provides functionality related to signing in, changing passwords, etc.

E-mail Component
[Component: Spring Bean]

Sends e-mail to users.

Mainframe Banking System Facade
[Component: Spring Bean]

A facade onto the mainframe banking system.

reads from and
writes to
[JDBC]

sends e-mail using
[SMTP]

makes API calls to
[XML/HTTPS]

Database
[Container: Oracle Database Schema]

Stores user registration information, hashed authentication credentials, access logs, etc.

E-mail System
[Software System]

The internal Microsoft Exchange e-mail system.

Mainframe Banking System
[Software System]

Stores all of the core banking information about customers, accounts, transactions, etc.

Component Diagram for Internet Banking System - API Application

The compoent diagram for the API Application

Last modified: 2022-02-10