Sesión 01

Diseño y conceptos básicos

Instructor:

ERICK ARÓSTEGUI

earostegui@galaxy.edu.pe

GALAXY
TRAINING

ÍNDICE

GALAXY
TRAINING

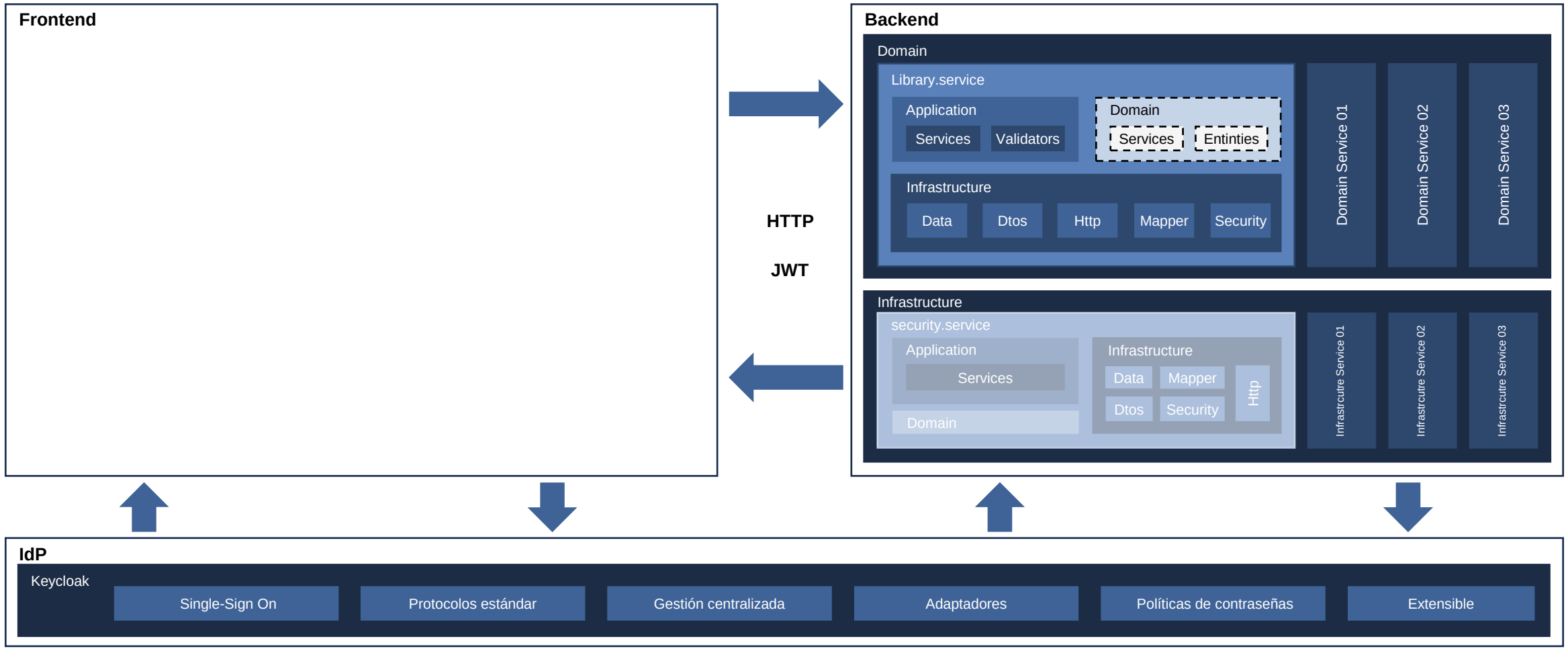
01

GALAXY
TRAINING

Diseño de la estructura del proyecto.

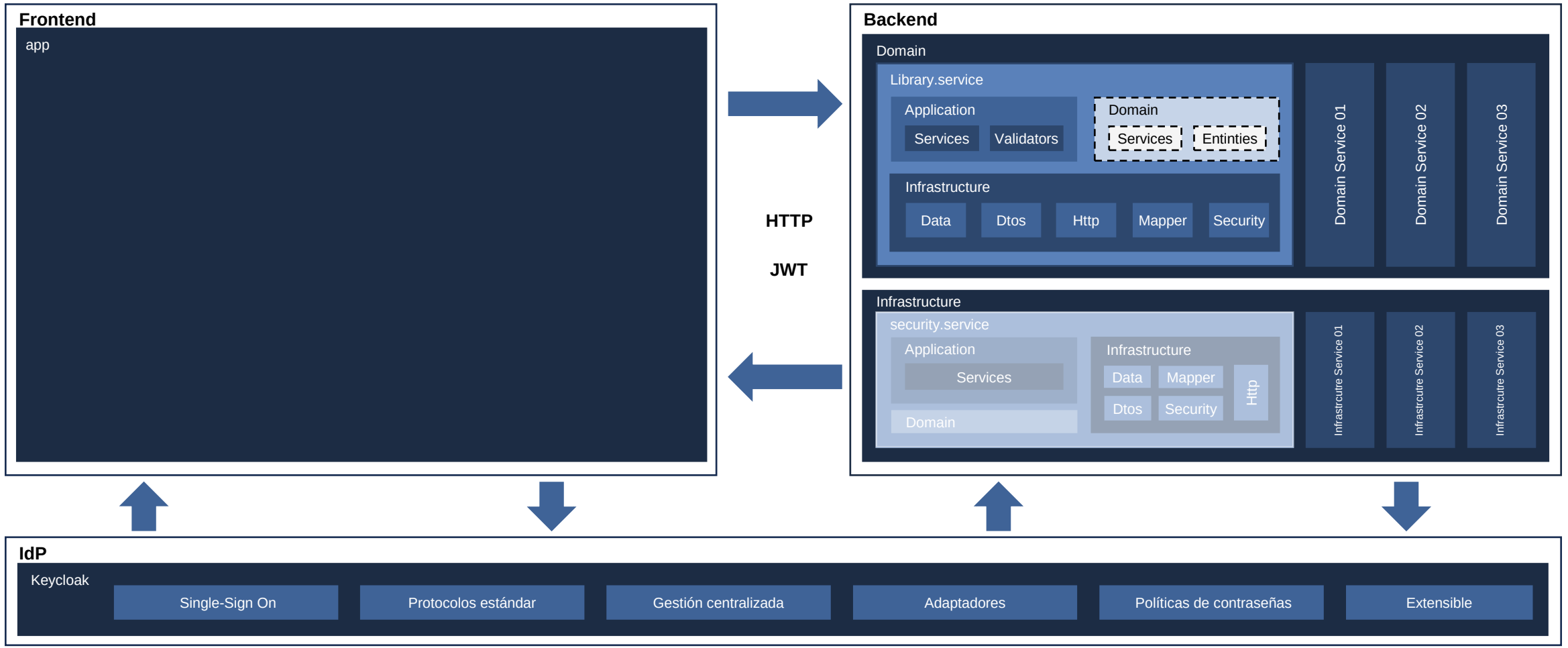
Diseño de la estructura del proyecto

Arquitectura de Aplicación



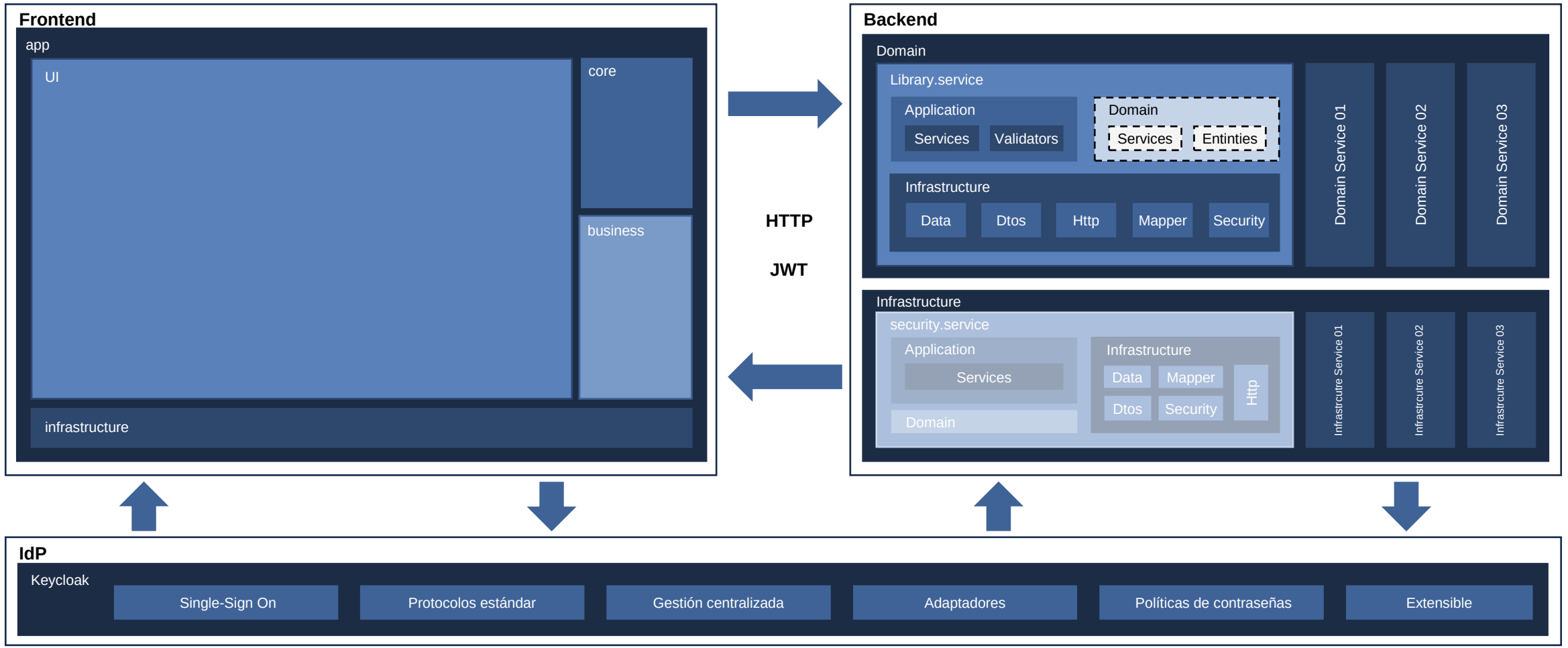
Diseño de la estructura del proyecto

Arquitectura de Aplicación



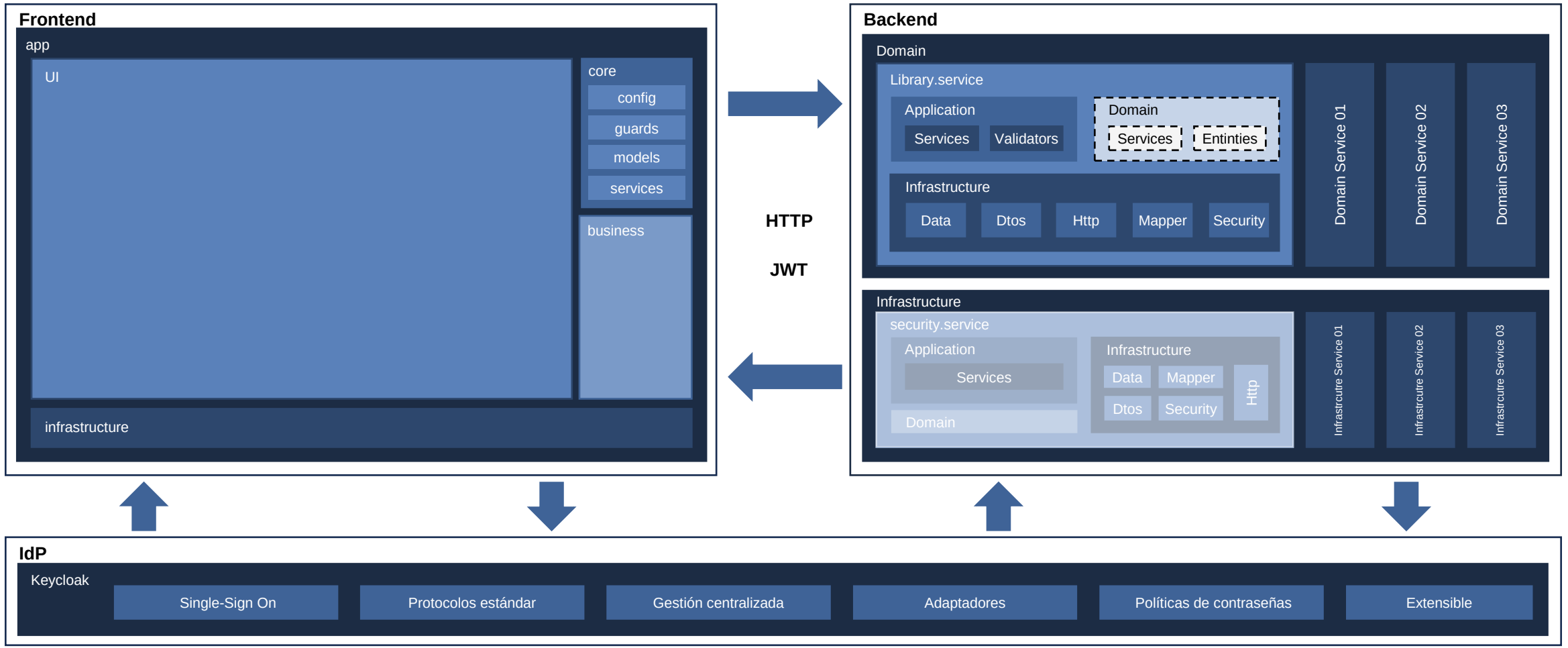
Diseño de la estructura del proyecto

Arquitectura de Aplicación



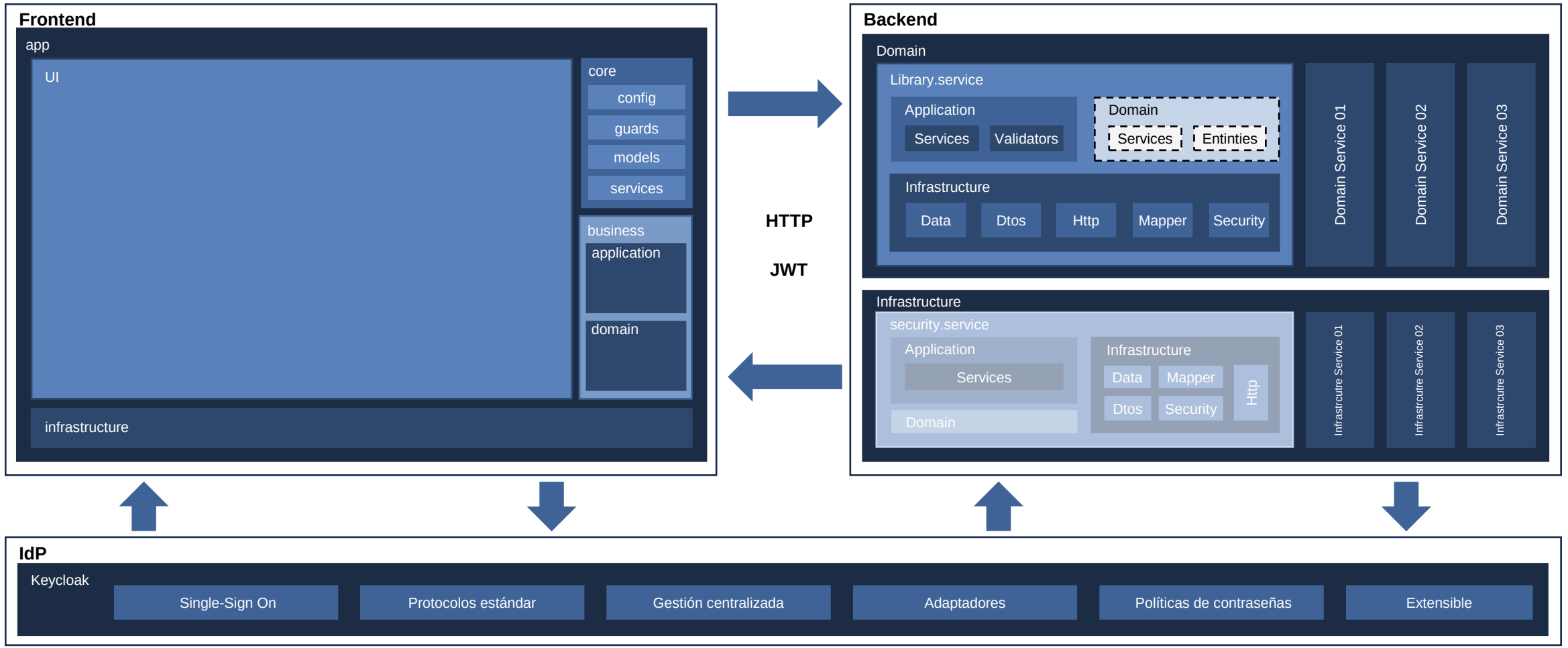
Diseño de la estructura del proyecto

Arquitectura de Aplicación



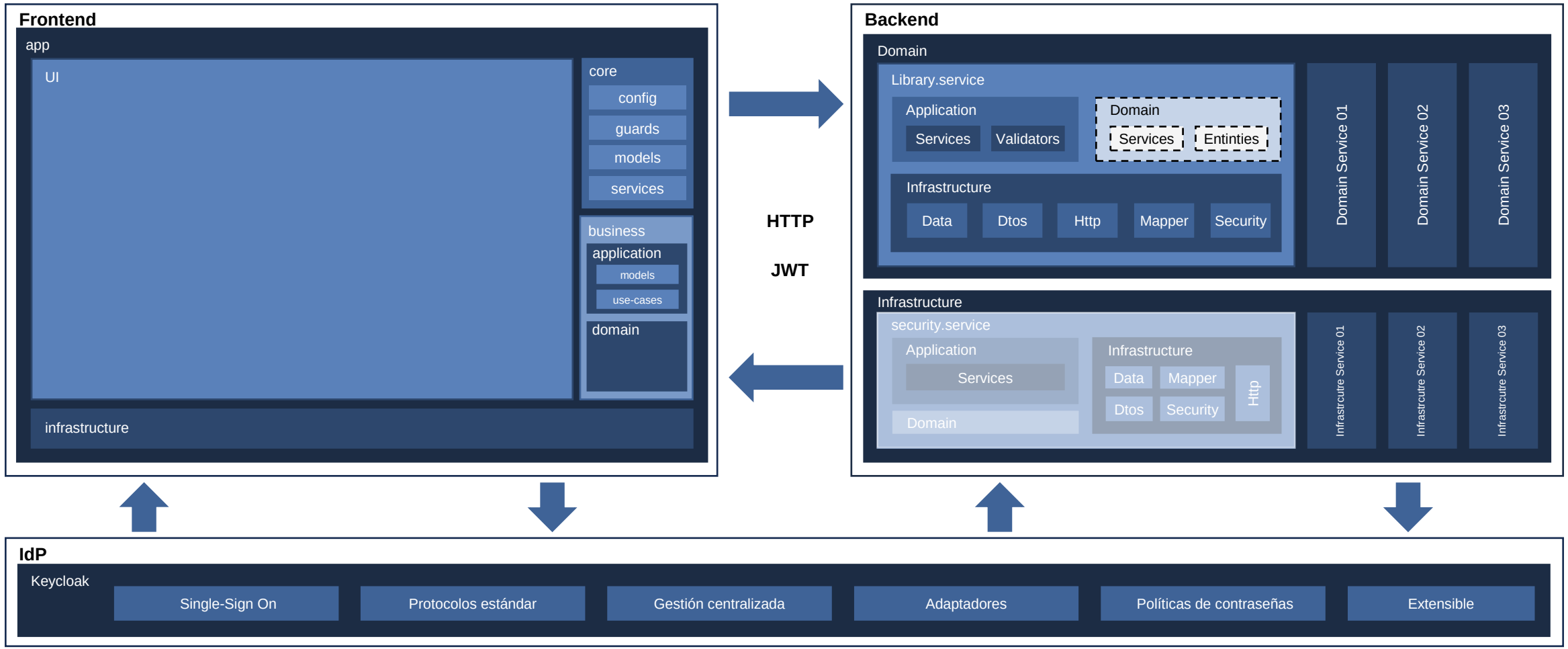
Diseño de la estructura del proyecto

Arquitectura de Aplicación



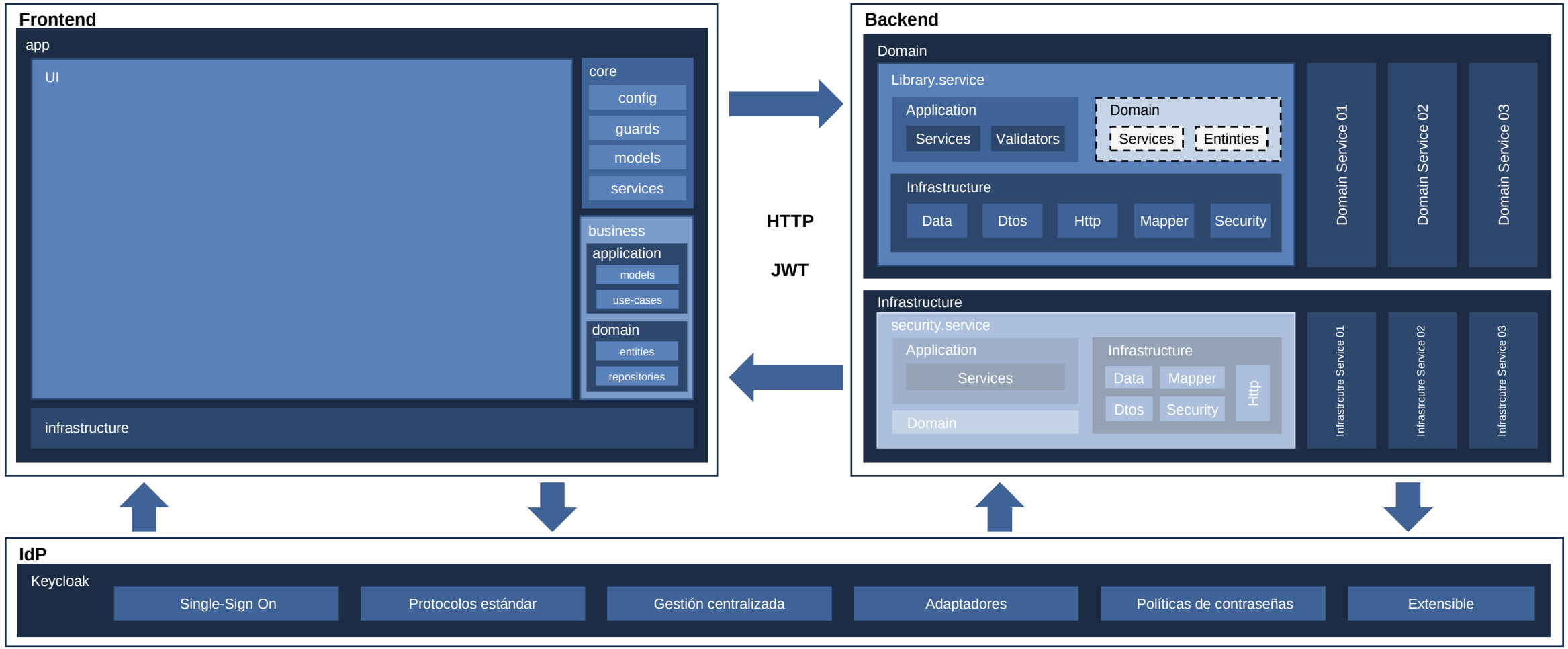
Diseño de la estructura del proyecto

Arquitectura de Aplicación



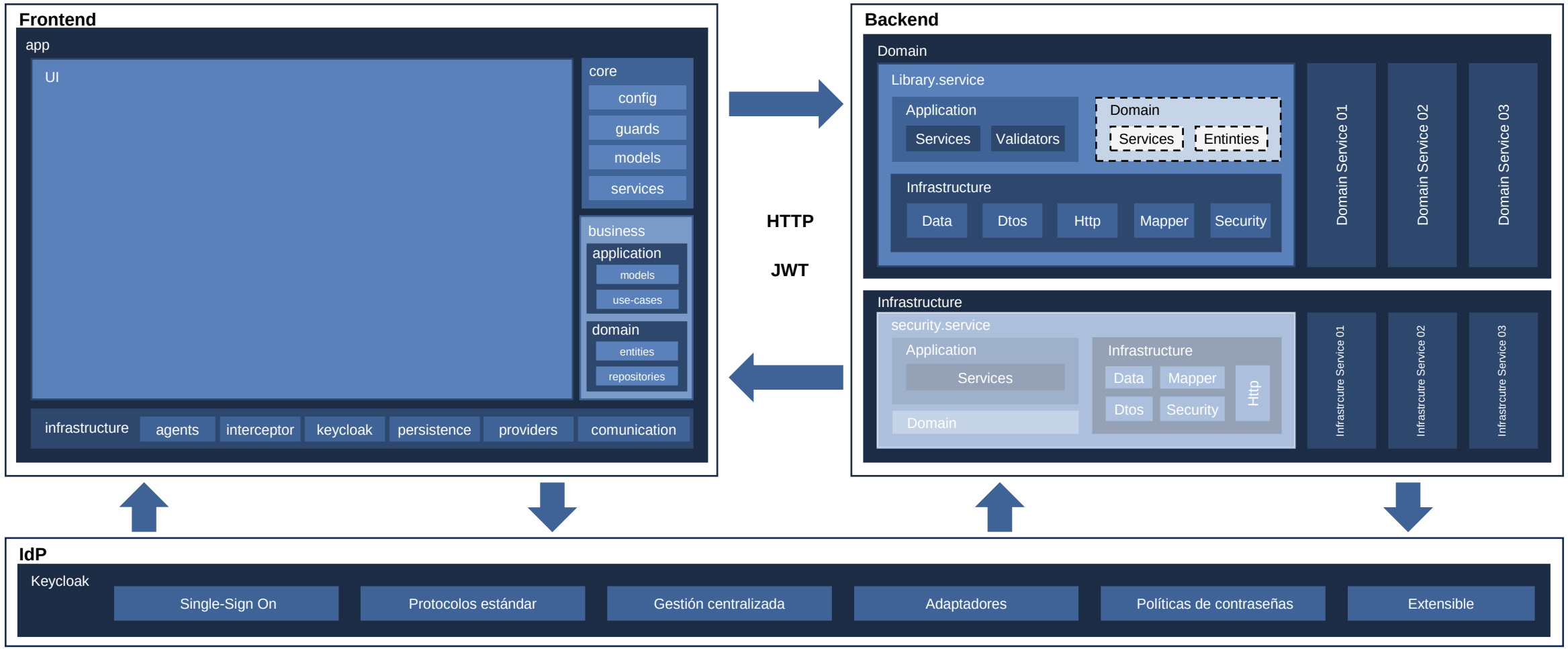
Diseño de la estructura del proyecto

Arquitectura de Aplicación



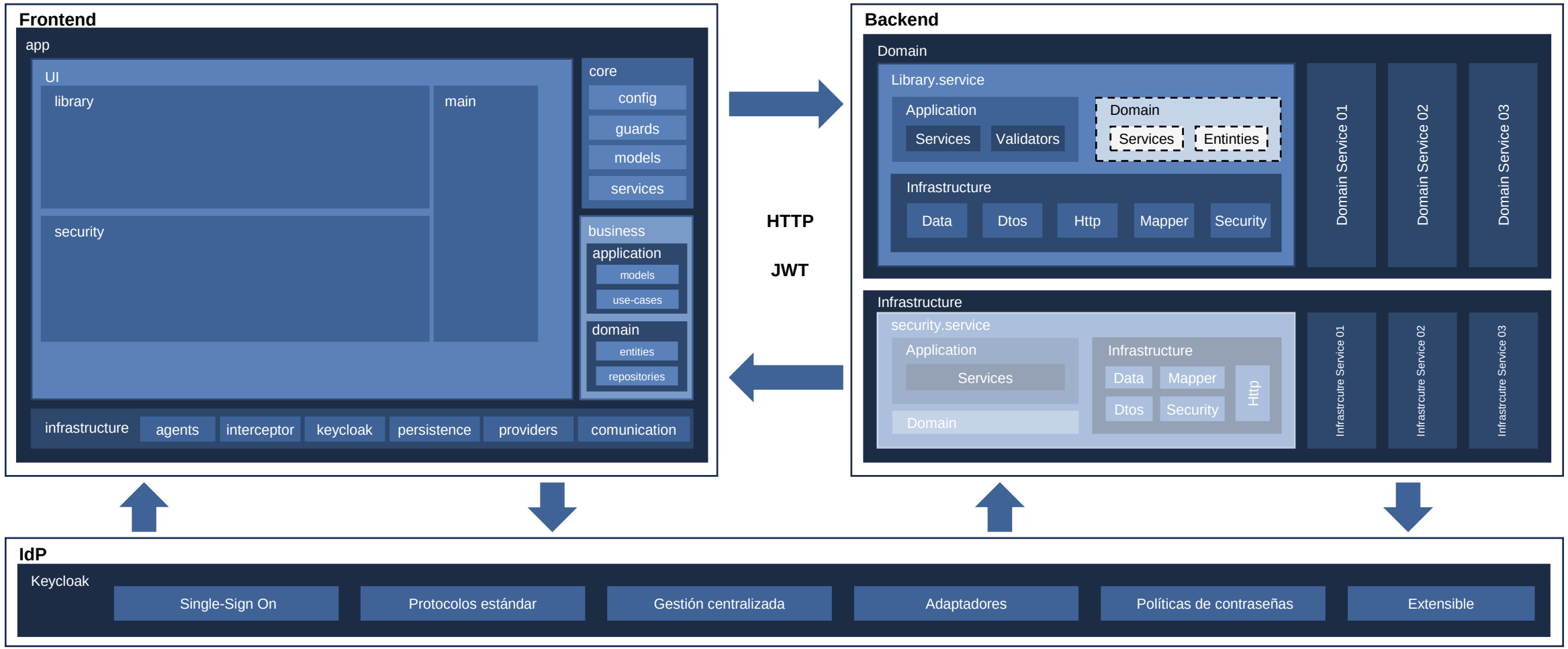
Diseño de la estructura del proyecto

Arquitectura de Aplicación



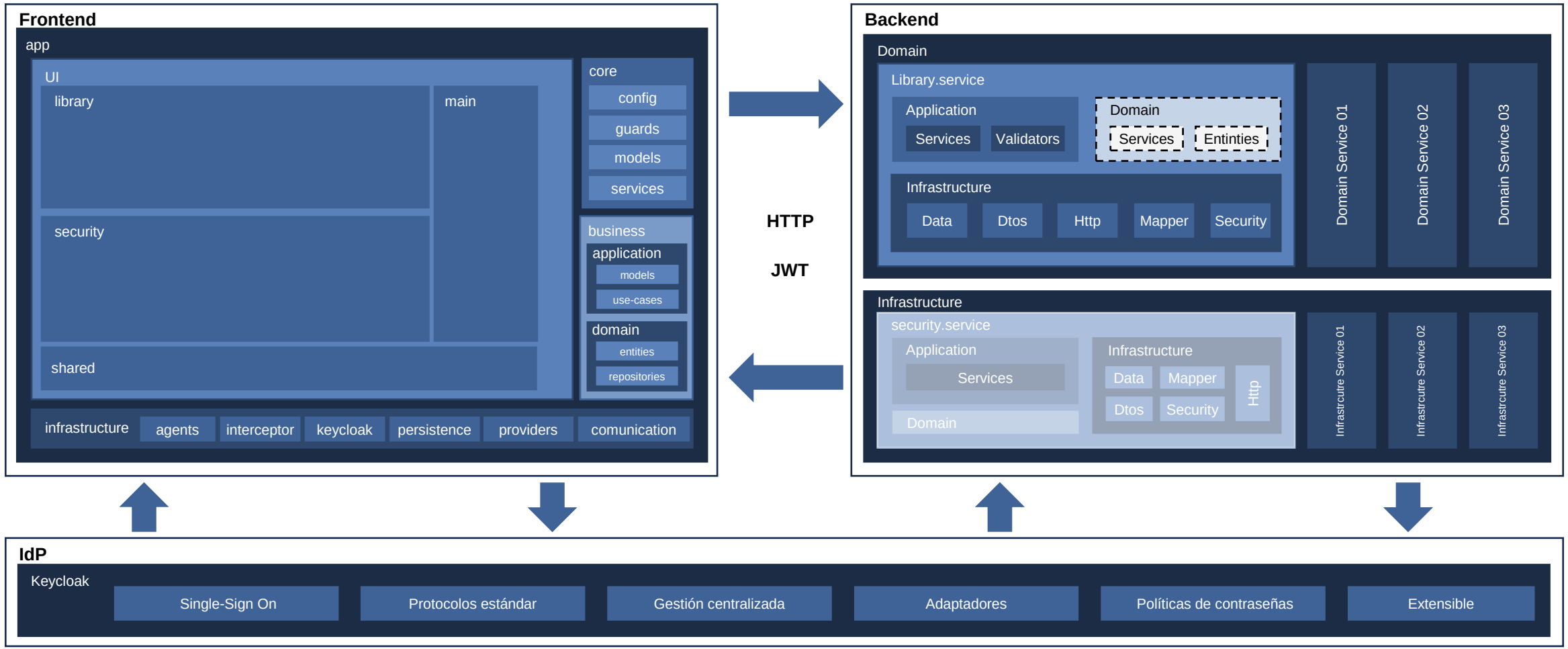
Diseño de la estructura del proyecto

Arquitectura de Aplicación



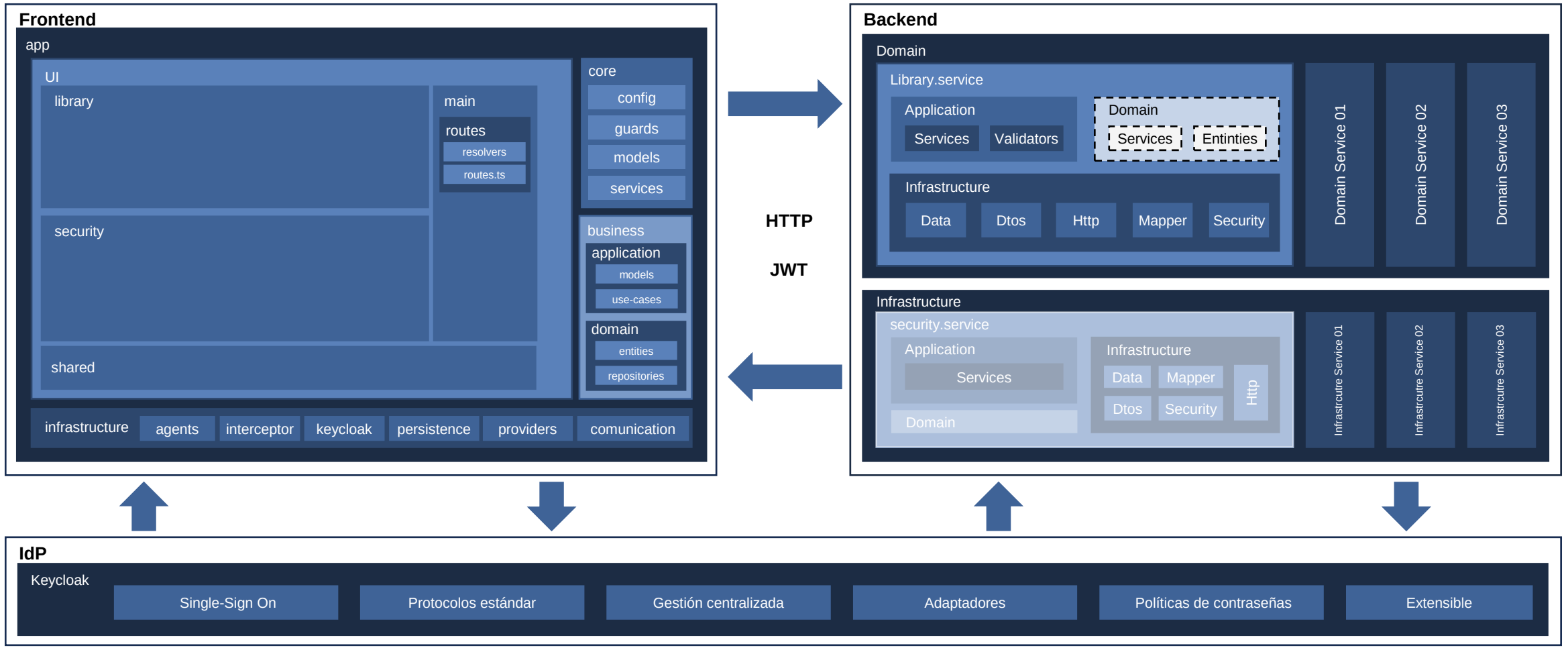
Diseño de la estructura del proyecto

Arquitectura de Aplicación



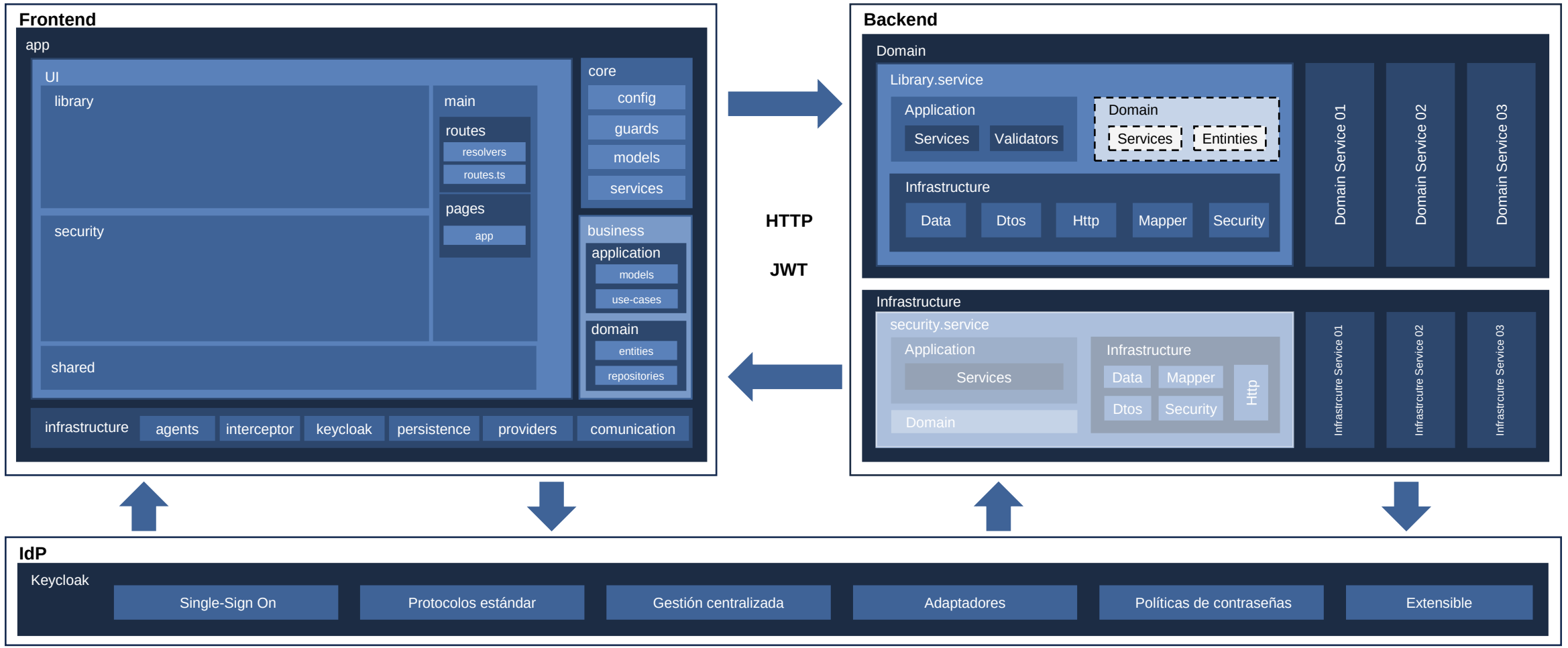
Diseño de la estructura del proyecto

Arquitectura de Aplicación



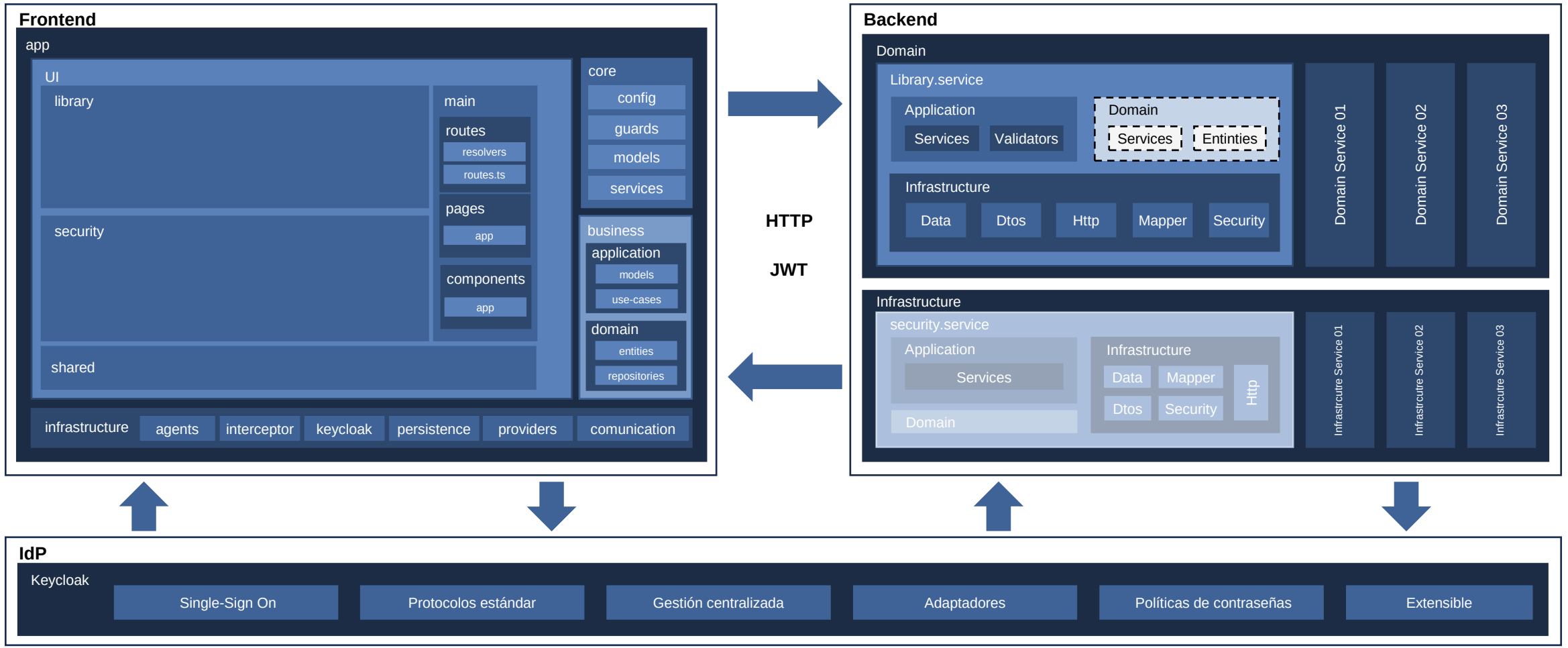
Diseño de la estructura del proyecto

Arquitectura de Aplicación



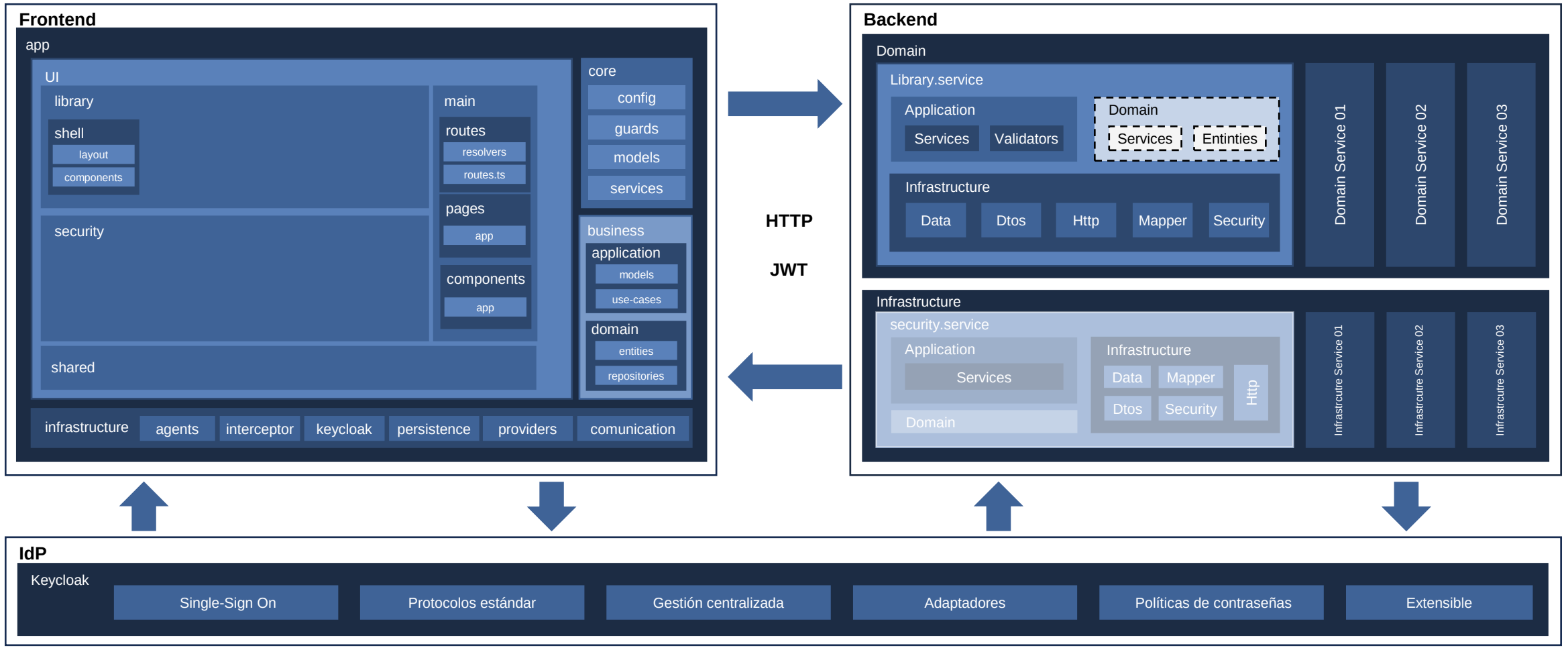
Diseño de la estructura del proyecto

Arquitectura de Aplicación



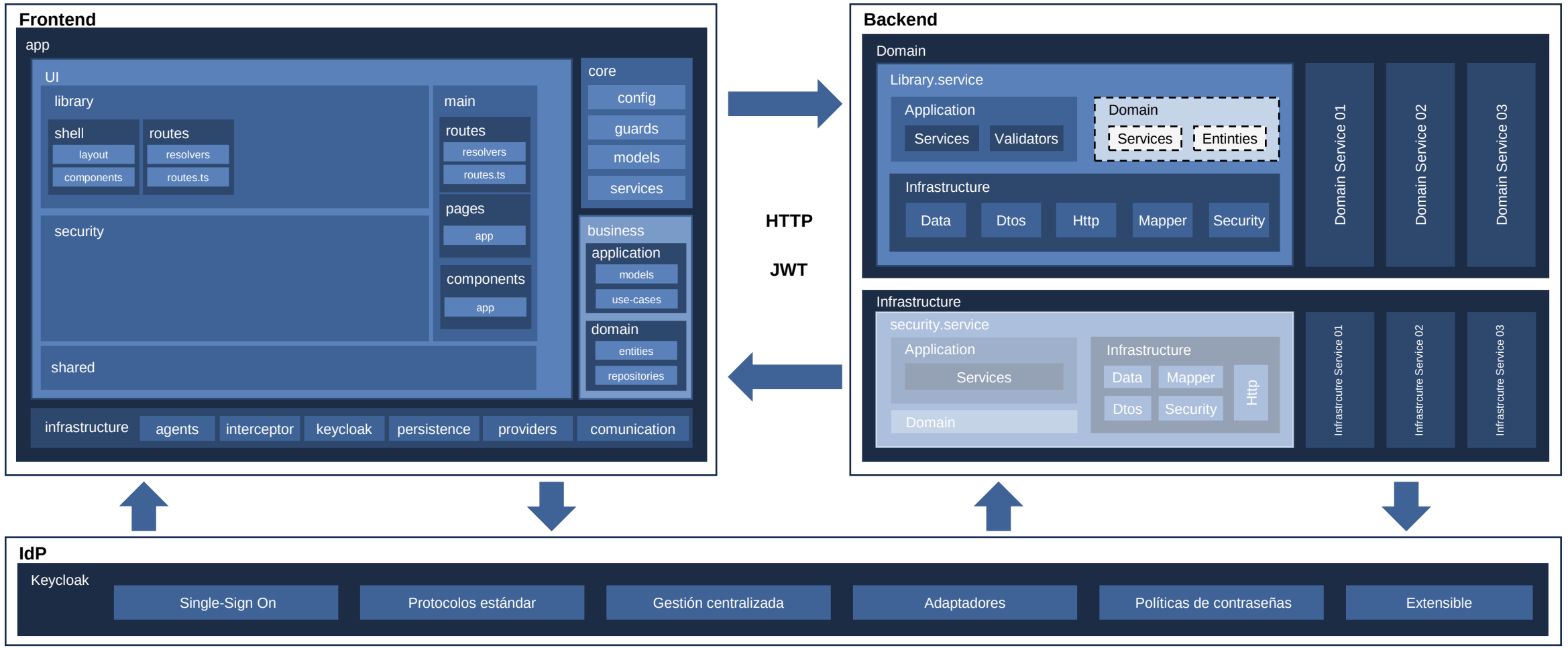
Diseño de la estructura del proyecto

Arquitectura de Aplicación



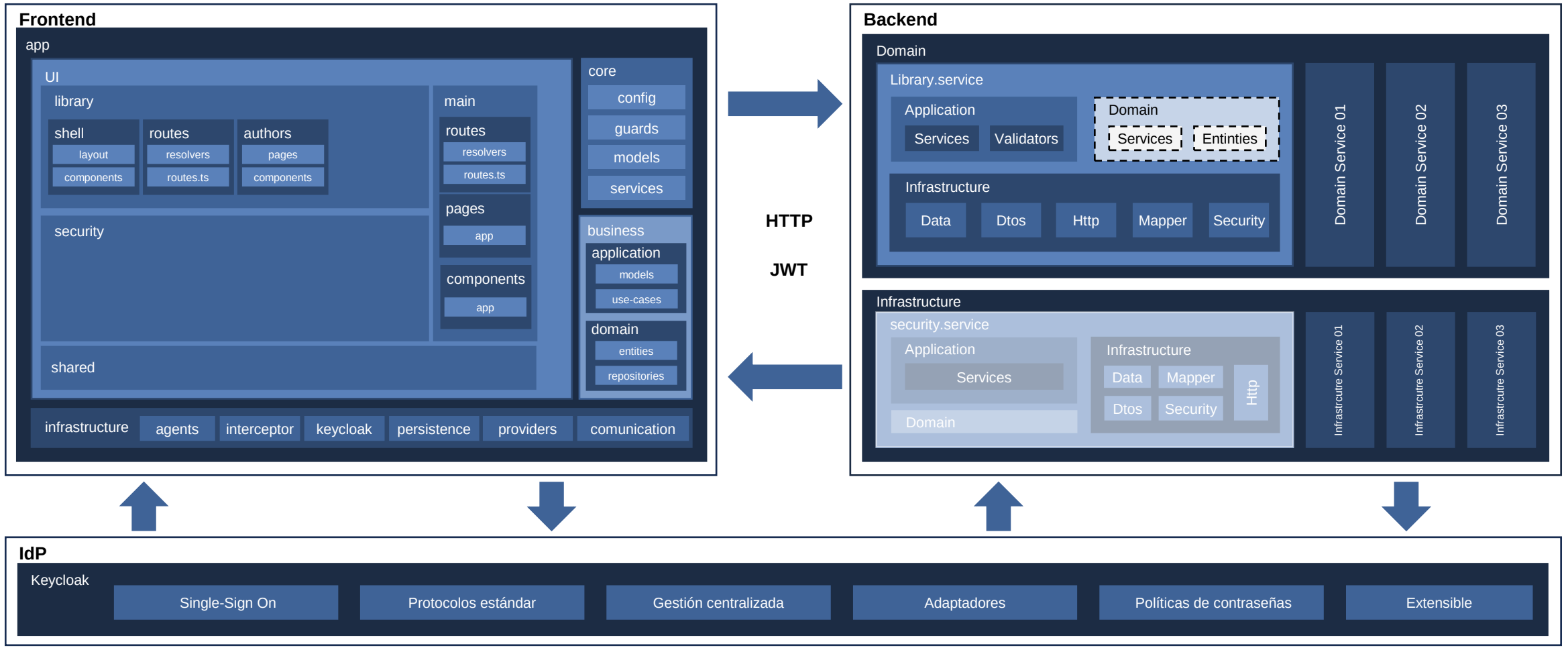
Diseño de la estructura del proyecto

Arquitectura de Aplicación



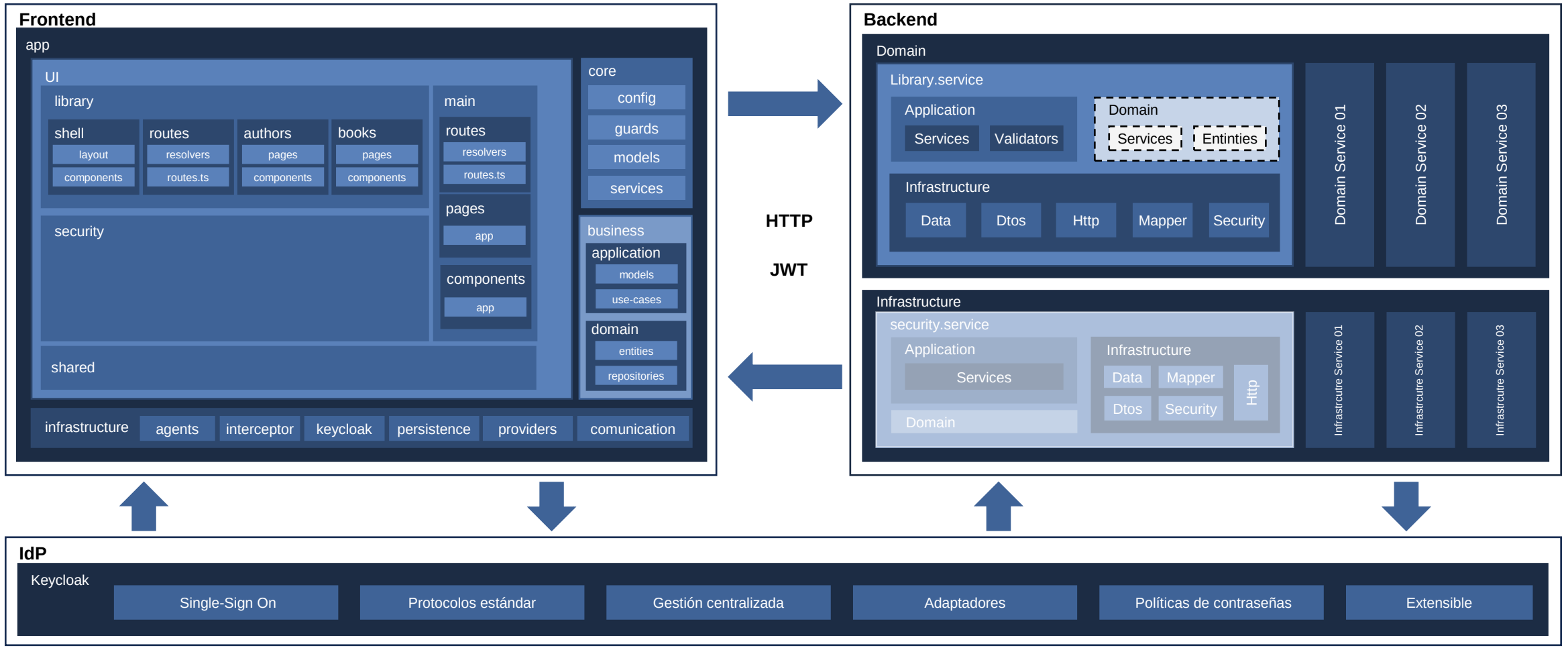
Diseño de la estructura del proyecto

Arquitectura de Aplicación



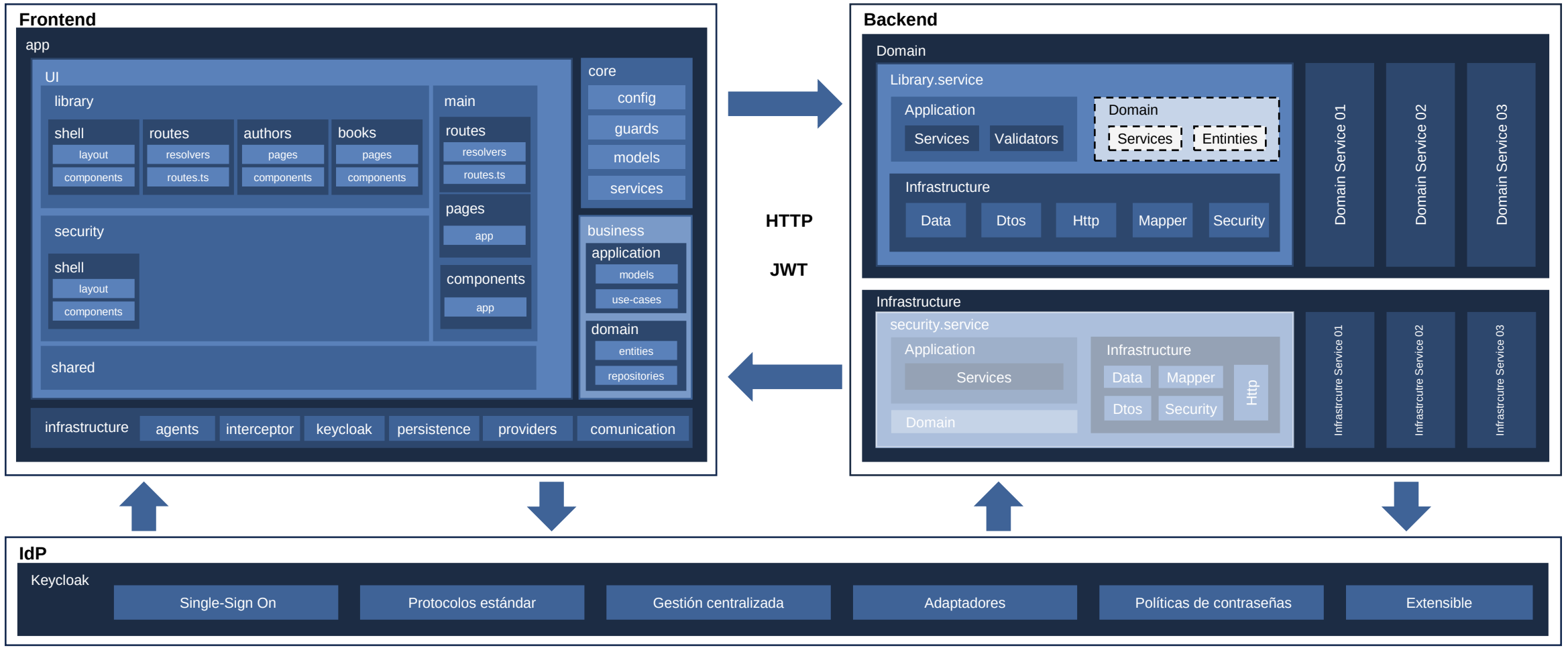
Diseño de la estructura del proyecto

Arquitectura de Aplicación



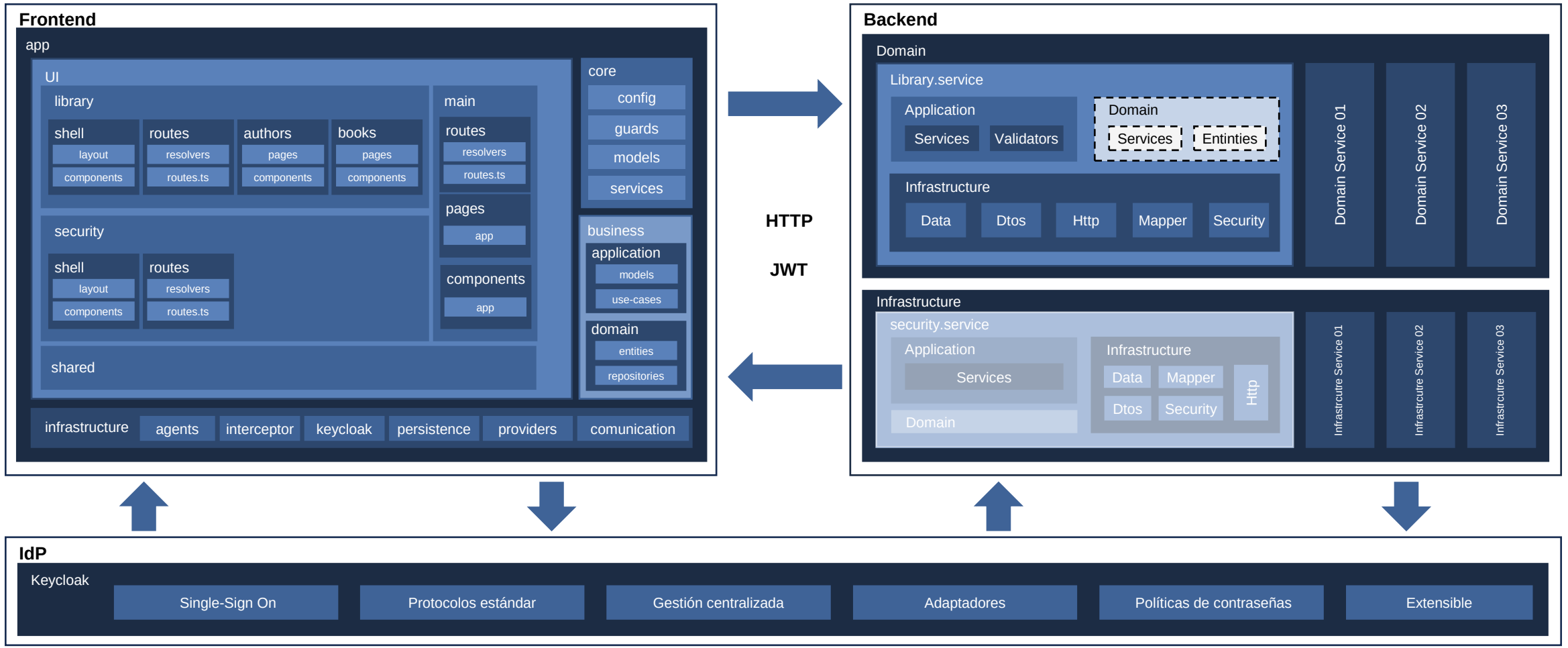
Diseño de la estructura del proyecto

Arquitectura de Aplicación



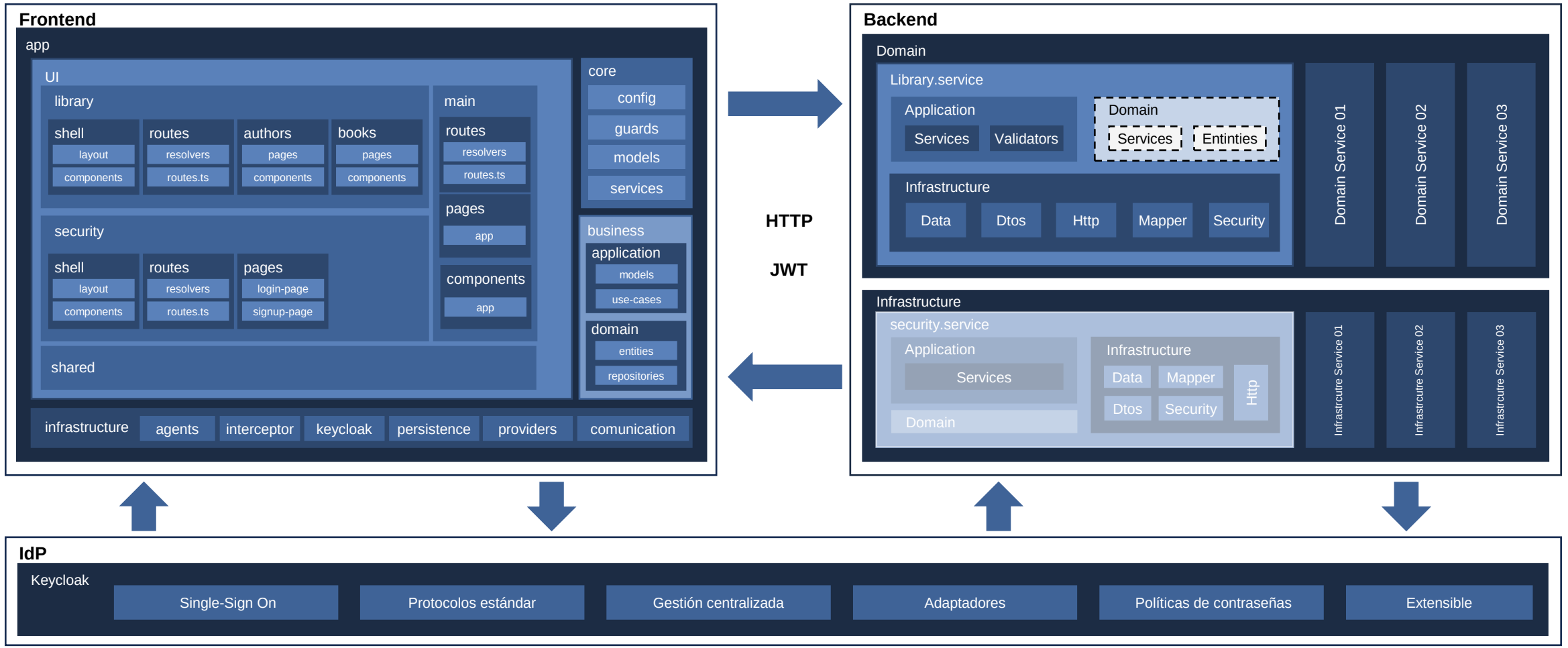
Diseño de la estructura del proyecto

Arquitectura de Aplicación



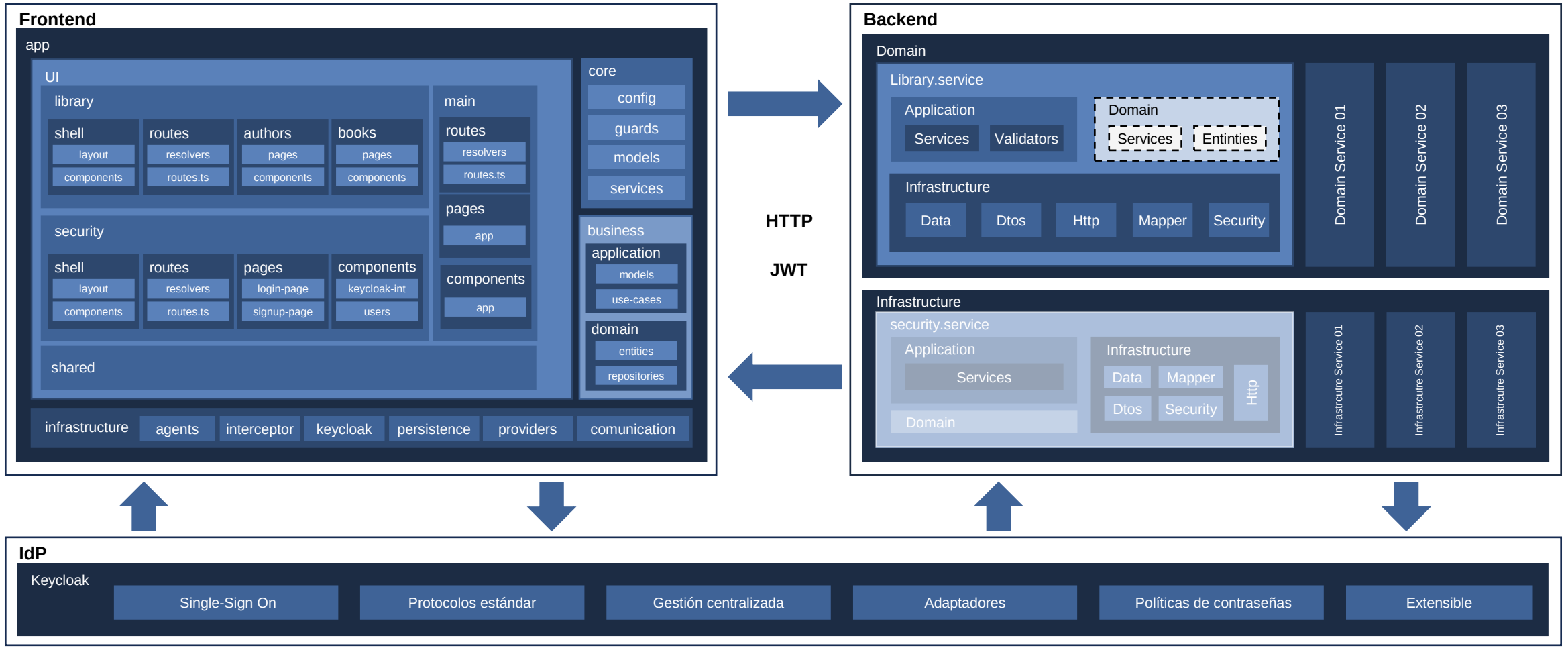
Diseño de la estructura del proyecto

Arquitectura de Aplicación



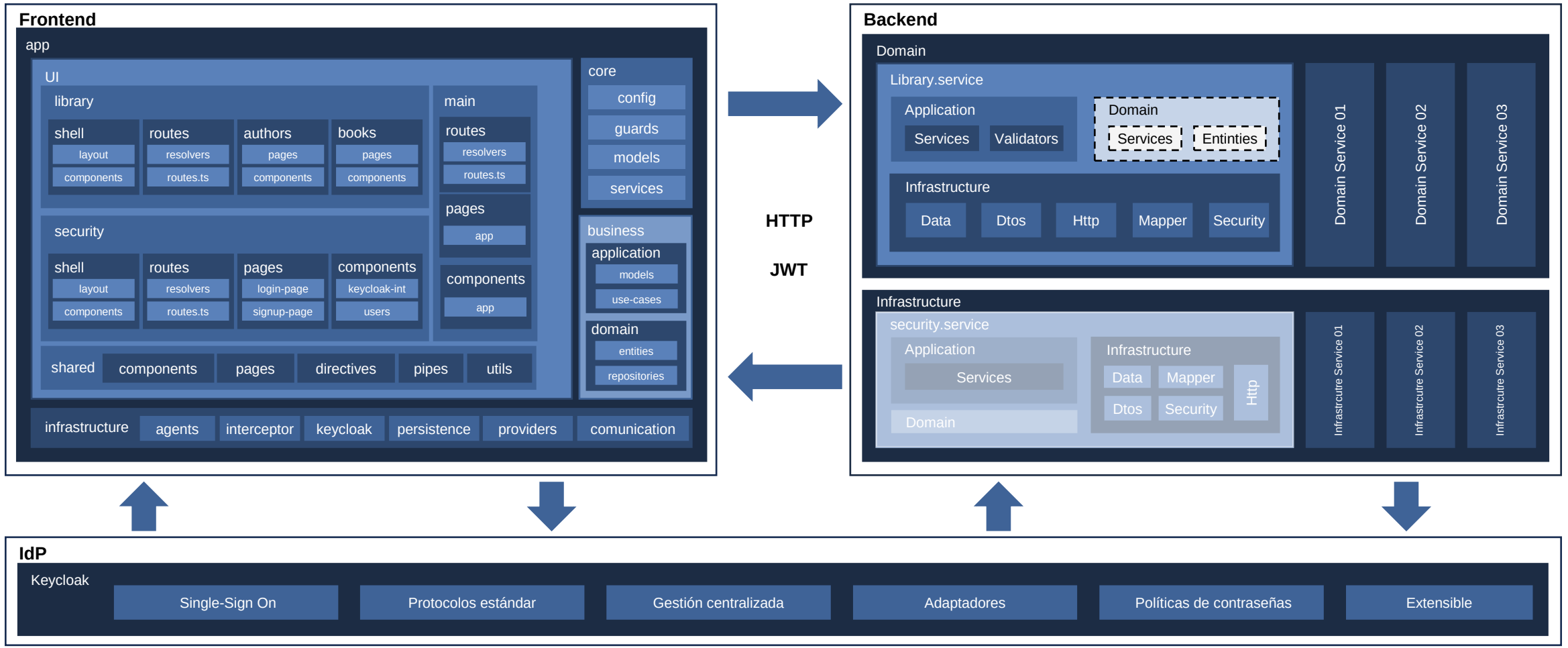
Diseño de la estructura del proyecto

Arquitectura de Aplicación



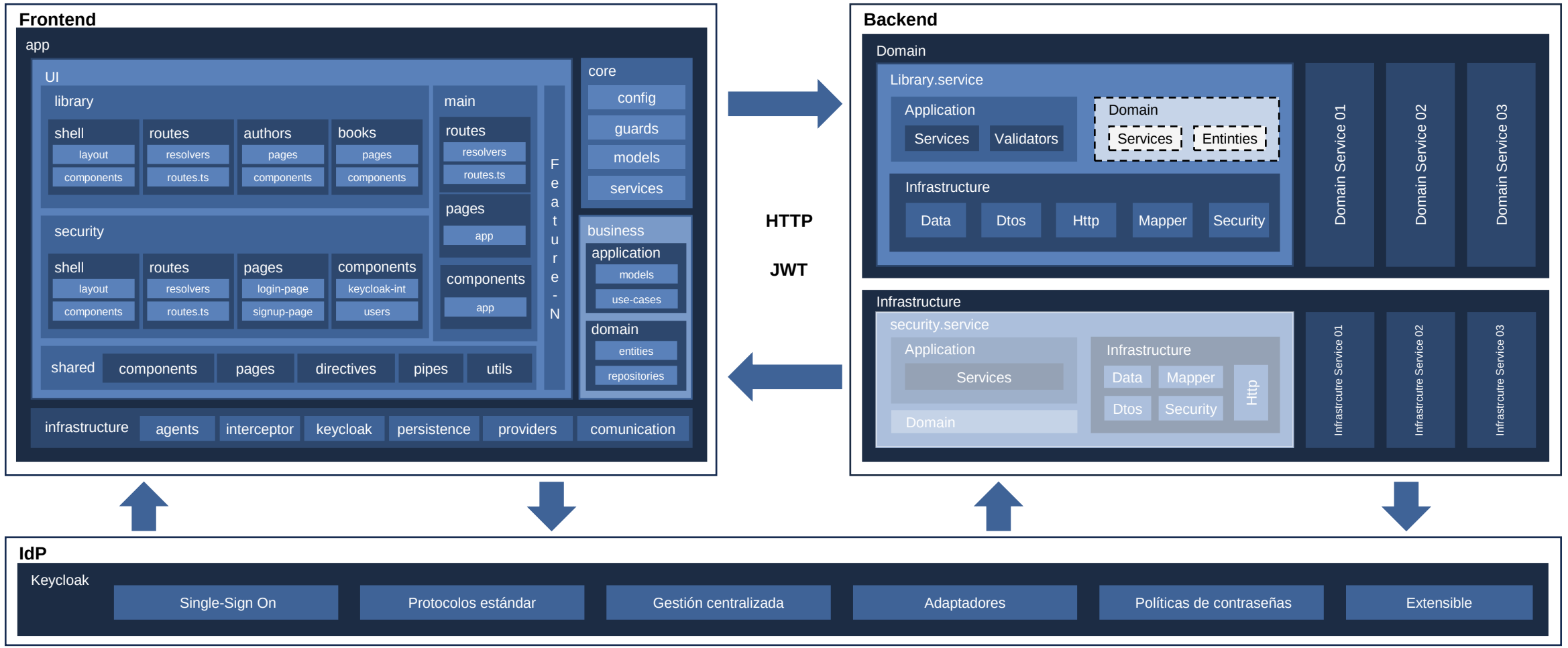
Diseño de la estructura del proyecto

Arquitectura de Aplicación



Diseño de la estructura del proyecto

Arquitectura de Aplicación



02

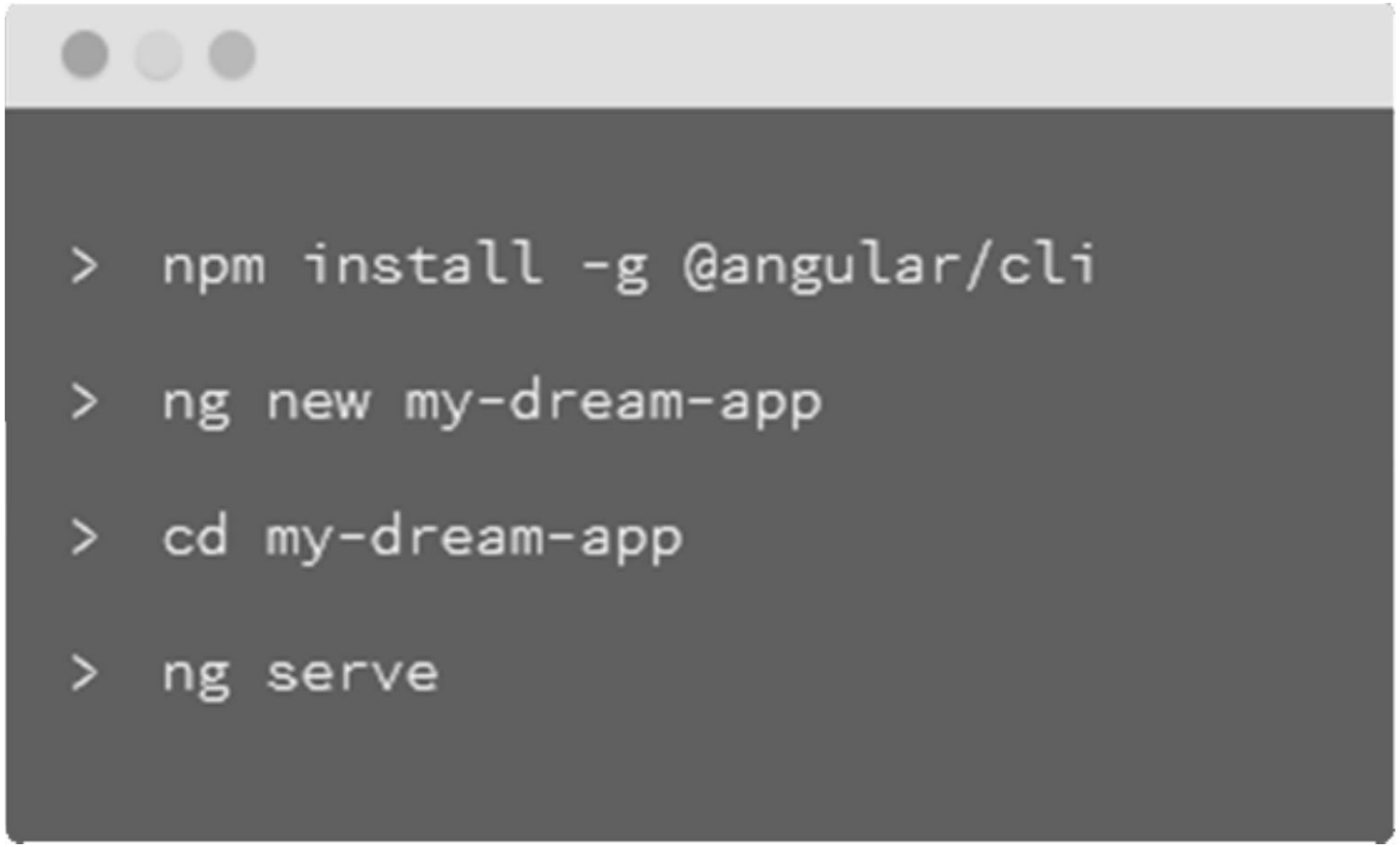
Creación del Proyecto.

GALAXY
TRAINING

DEMO

GALAXY
TRAINING

Creación del Proyecto



03

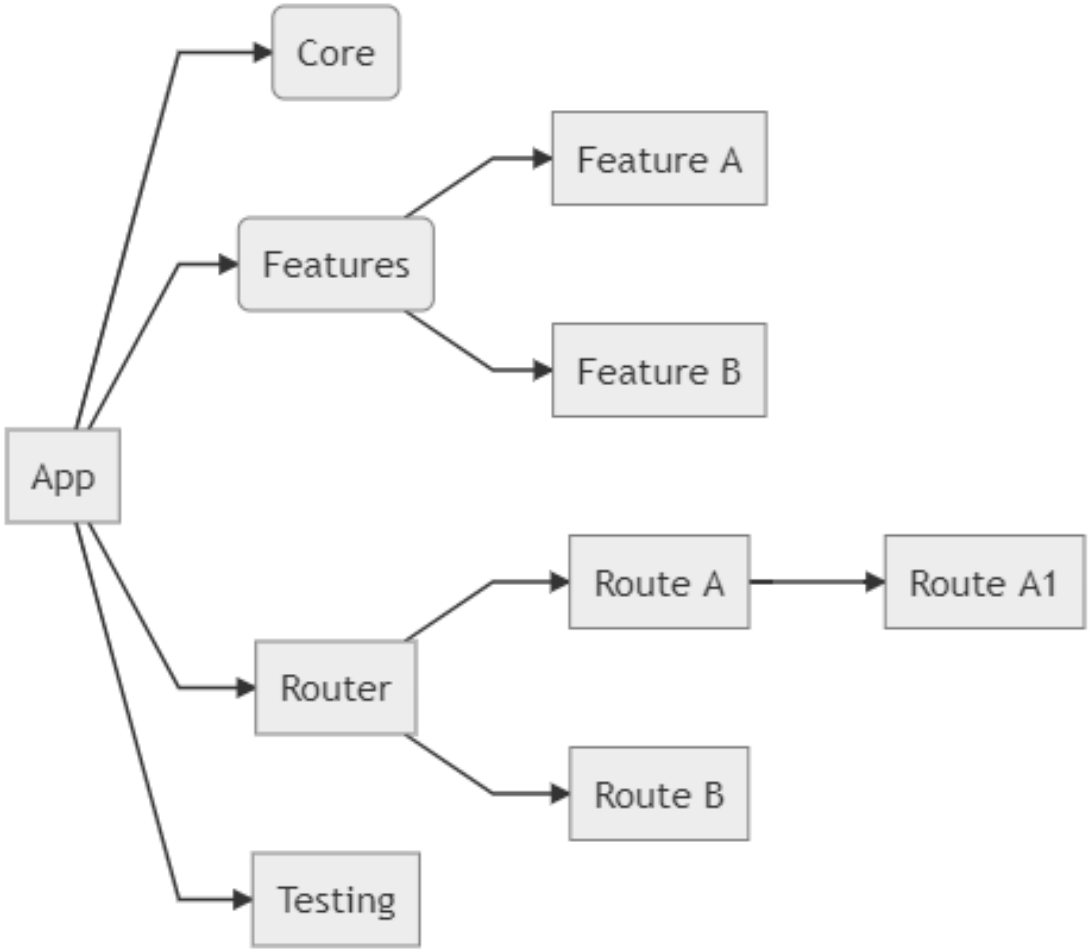
GALAXY
TRAINING

Configuración del proyecto.

DEMO

GALAXY
TRAINING

Configuración del proyecto

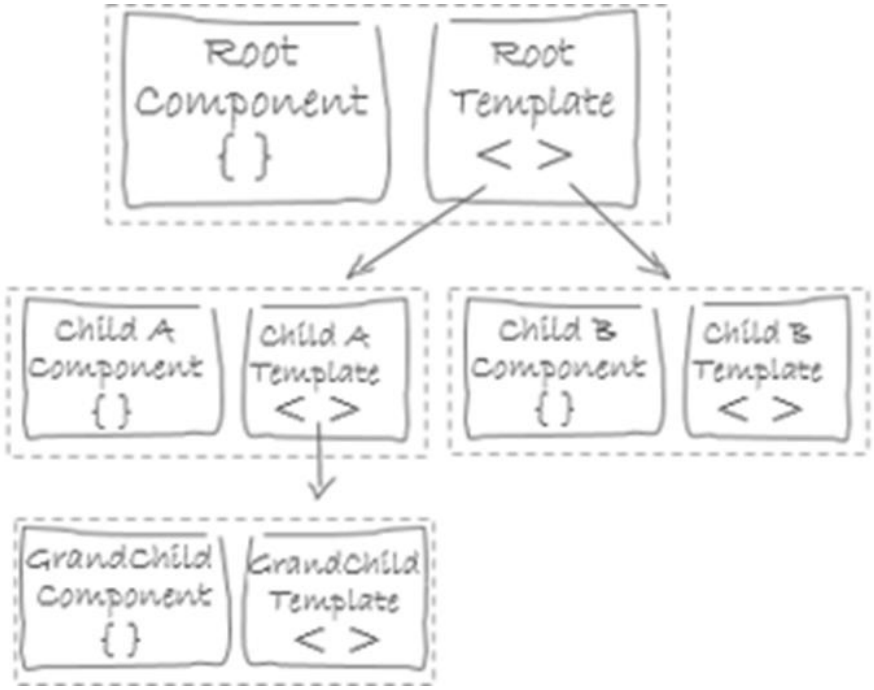


04

Componentes y servicios

GALAXY
TRAINING

Componentes



- **Un Template** es un HTML que indica la estructura visual
- **Una clase**, formada por por propiedades, métodos, atributos y funciones que definen el comportamiento del componente.
- **Metadata,** que sirve para indicar un nombre mediante el cual referenciar al componente desde nuestro HTML.

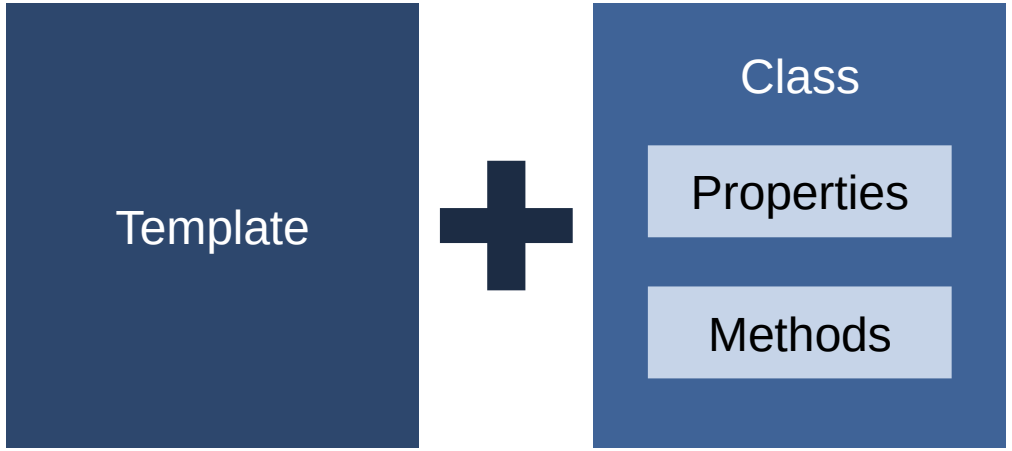
Componentes

Componentes

Template

- View layout
- Usa HTML
- Incluye binding y directivas

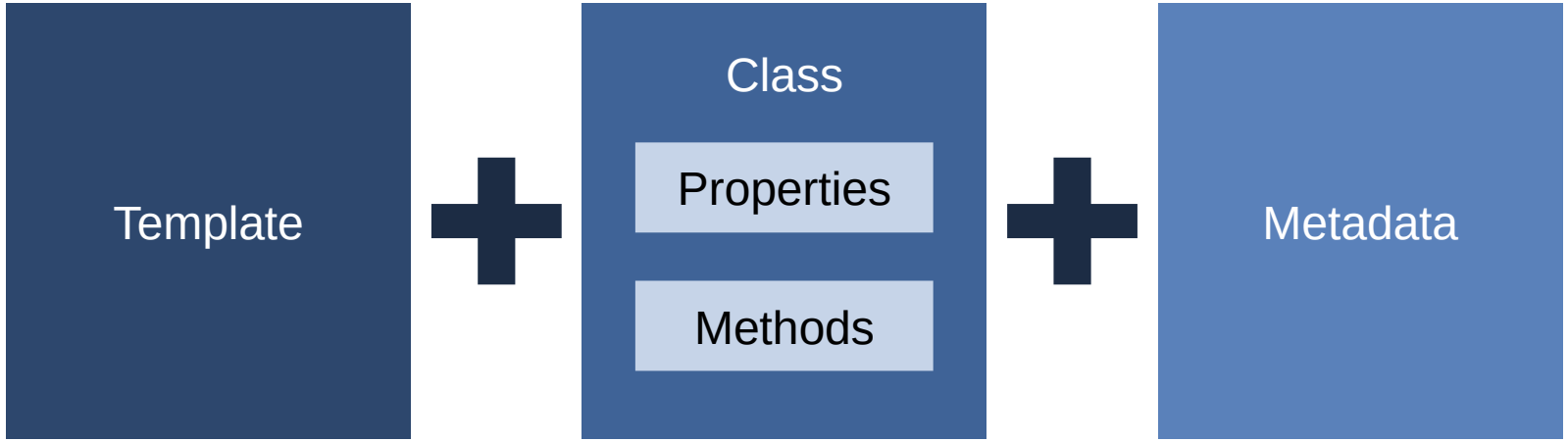
Componentes



- View layout
- Usa HTML
- Incluye binding y directivas

- Código de soporte a la vista
- Usa TypeScript
- Propiedades: data
- Metodos: logica

Componentes



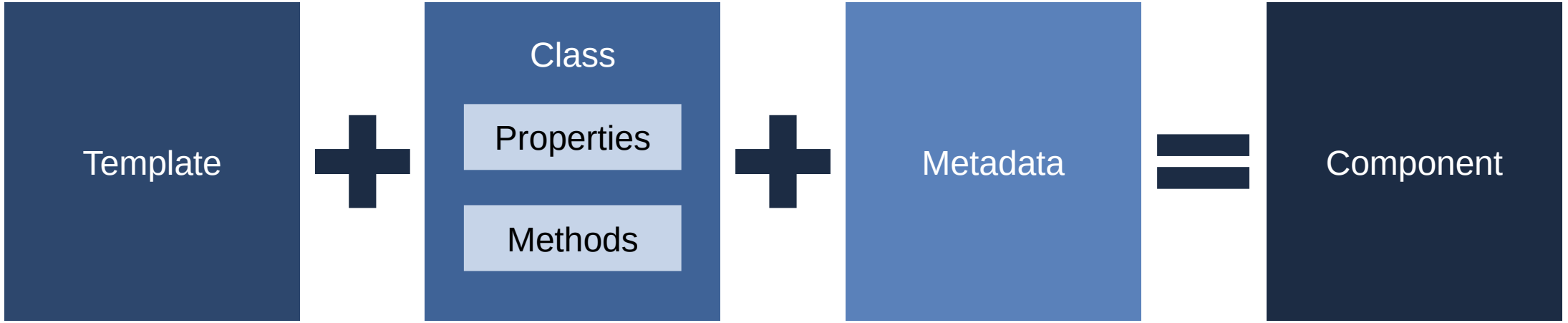
- View layout
- Usa HTML
- Incluye binding y directivas

- Código de soporte a la vista
- Usa TypeScript
- Propiedades: data
- Metodos: logica

- Información adicional para la ejecución de Angular
- Definido por decoradores

GALAXY TRAINING

Componentes



- View layout
- Usa HTML
- Incluye binding y directivas

- Código de soporte a la vista
- Usa TypeScript
- Propiedades: data
- Metodos: logica

- Información adicional para la ejecución de Angular
- Definido por decoradores

Componentes

GALAXY TRAINING

Componentes

app.component

Componentes

app.component

```
import { Component } from '@angular/core';
```

Import

GALAXY TRAINING

Componentes

app.component

```
import { Component } from '@angular/core';

@Component({
    selector: 'pm-root',
    template: `
    <div><h1>{{pageTitle}}</h1>
        <div>My First Component</div>
    </div>
    `
})
```

Import

Metadata & Template

GALAXY TRAINING

Componentes

app.component

```
import { Component } from '@angular/core';

@Component({
    selector: 'pm-root',
    template: `
    <div><h1>{{pageTitle}}</h1>
        <div>My First Component</div>
    </div>
    `
})
export class AppComponent {
 pageTitle: string = 'Acme Product Management';
}
```

Import

Metadata & Template

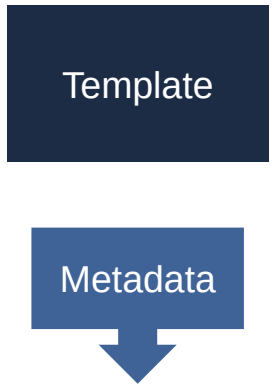
Class

Esquema de ejecución de componentes

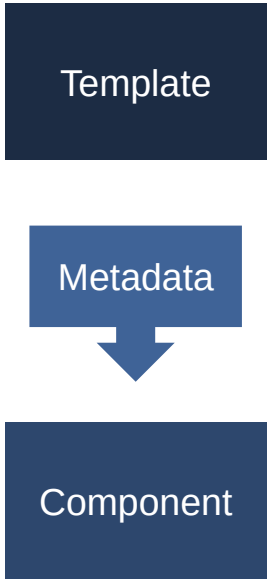
Esquema de ejecución de componentes

Template

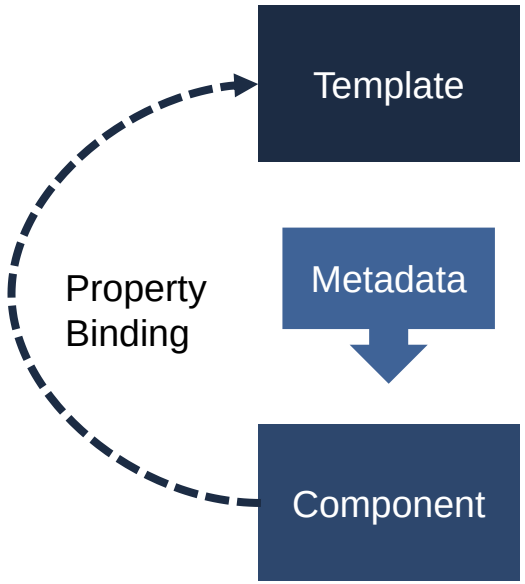
Esquema de ejecución de componentes



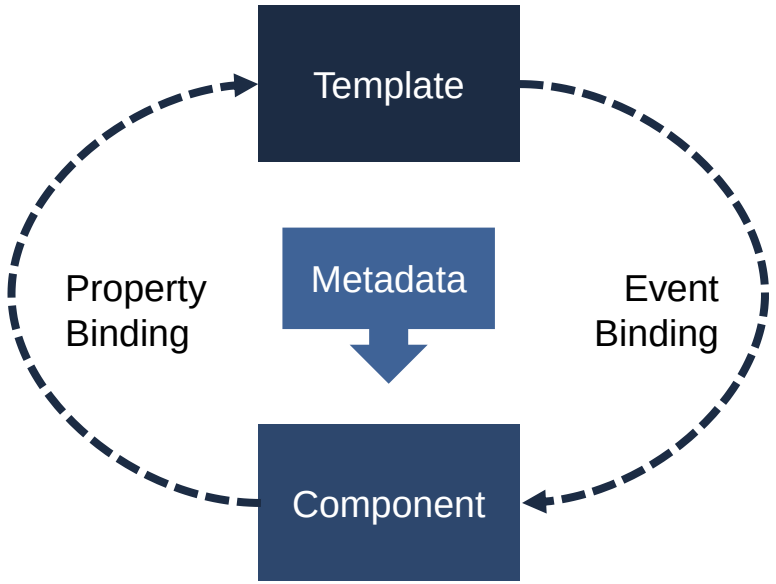
Esquema de ejecución de componentes



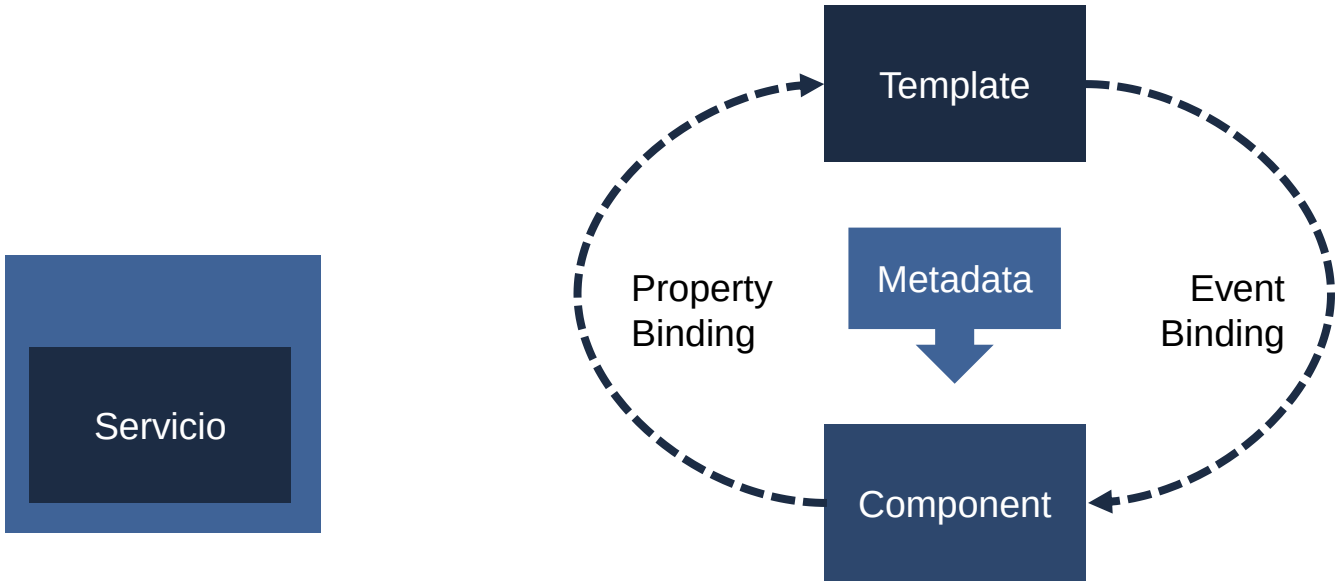
Esquema de ejecución de componentes



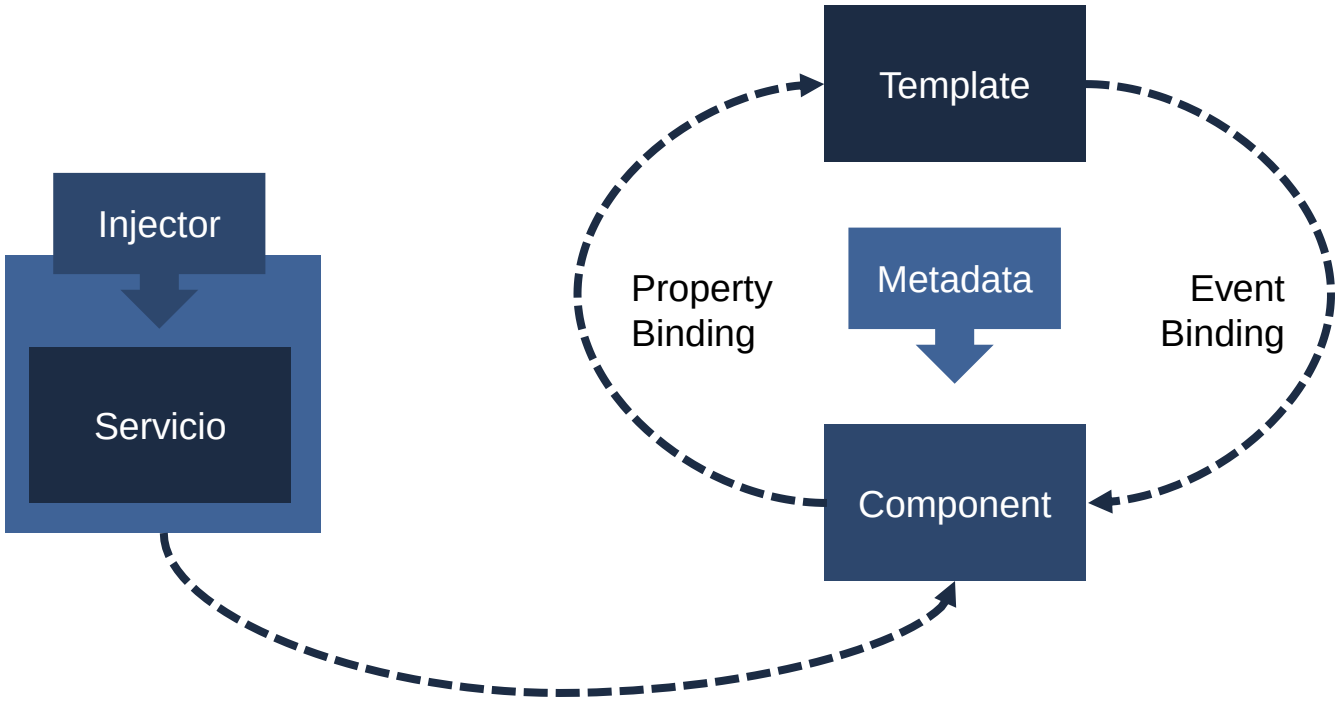
Esquema de ejecución de componentes



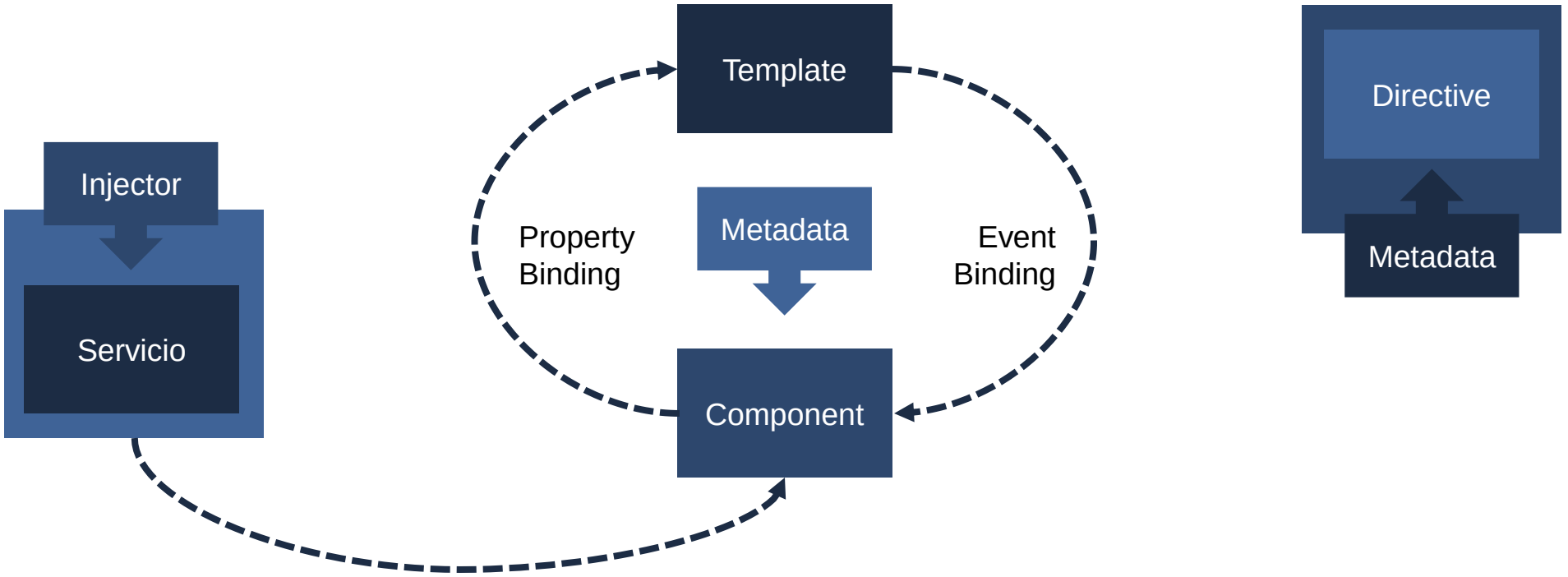
Esquema de ejecución de componentes



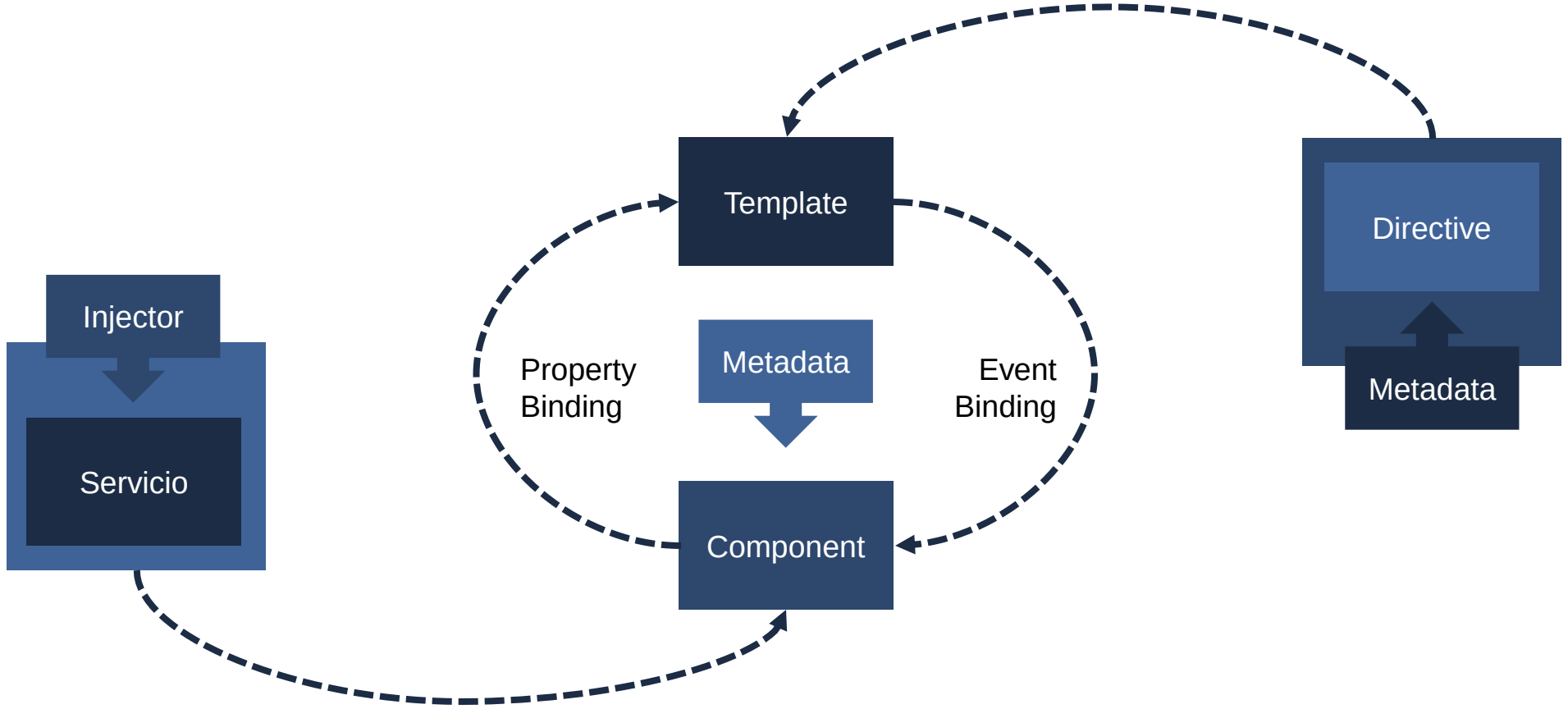
Esquema de ejecución de componentes



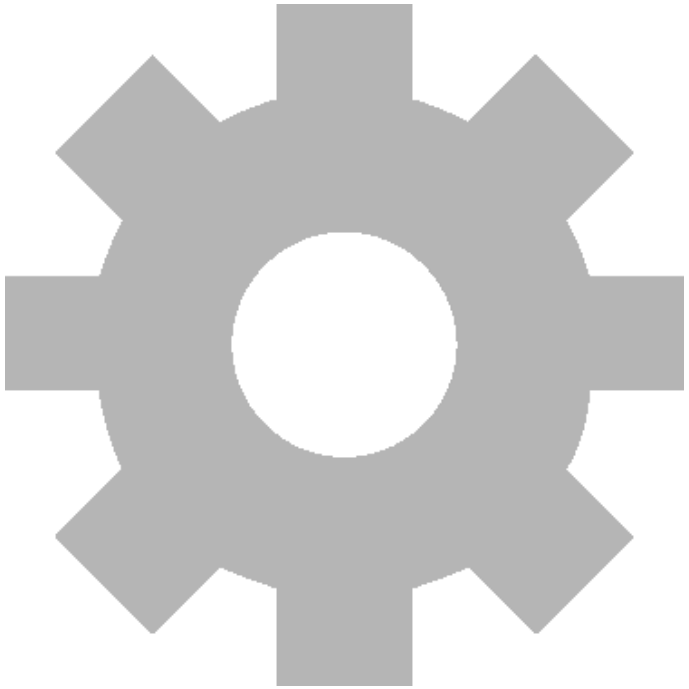
Esquema de ejecución de componentes



Esquema de ejecución de componentes



Servicios



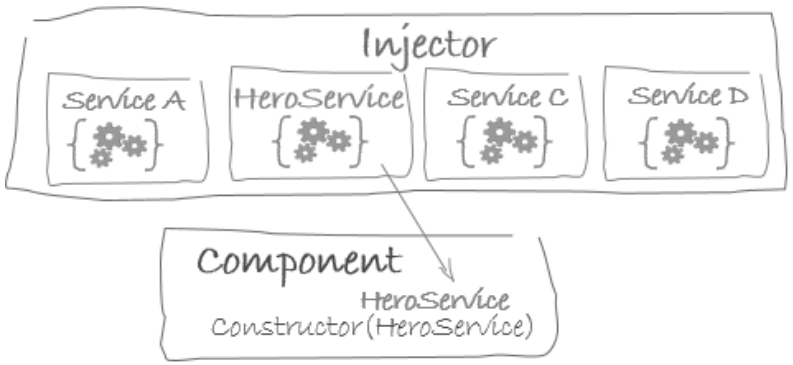
Una clase enfocada en un propósito.

- Pieza reutilizable de funcionalidad compartida entre componentes
- Responsable de una única pieza discreta de funcionalidad
- Capaz de ser entregado cuando y donde se necesita

Se utiliza para características que:

- Son independientes de cualquier componente en particular.
- Proporcionar datos o lógica compartidos entre componentes
- Encapsular interacciones externas

Servicios



Para los datos o la **lógica que no están asociados a una vista específica** y que desea compartir entre componentes, cree una clase de tipo servicio.

Una definición de clase de tipo servicio es inmediatamente precedida por el decorador. El decorador proporciona los metadatos que permiten que otros proveedores se inyecten como dependencias en la clase.

@Injectable()

Servicios

"Limite la **lógica en un componente solo a la requerida** para la vista. **Toda otra lógica debería delegarse a los servicios** ".

Angular Style Guide
https://angular.io/guide/styleguide#delegate-complex-component-logic-to-services

Servicios

Considere crear un servicio si ... ".

- La vista no requiere la funcionalidad necesaria
- Debe compartir la lógica o las reglas de negocio

  entre los componentes.
- Necesita compartir datos entre los componentes.

05

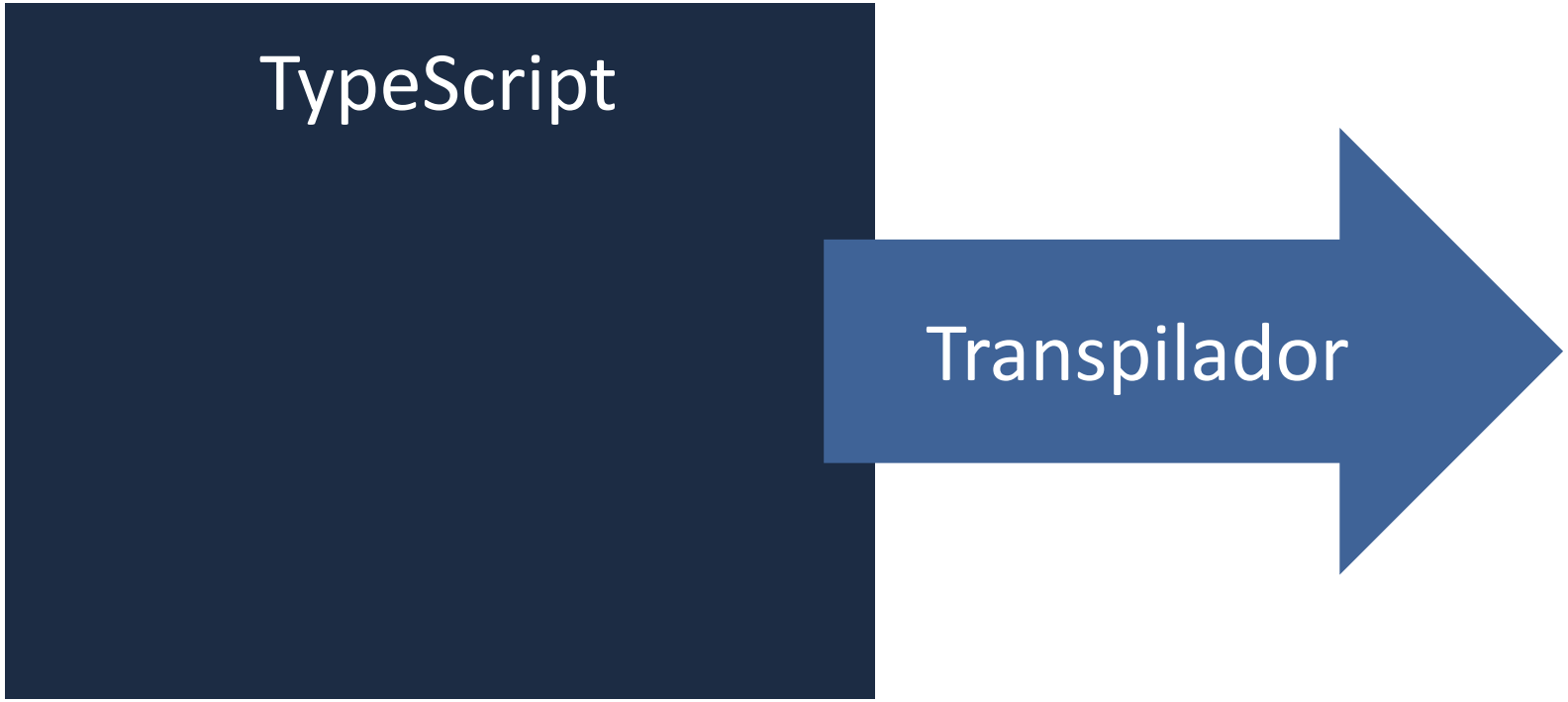
Modelos usando TypeScript

TypeScript, Definición

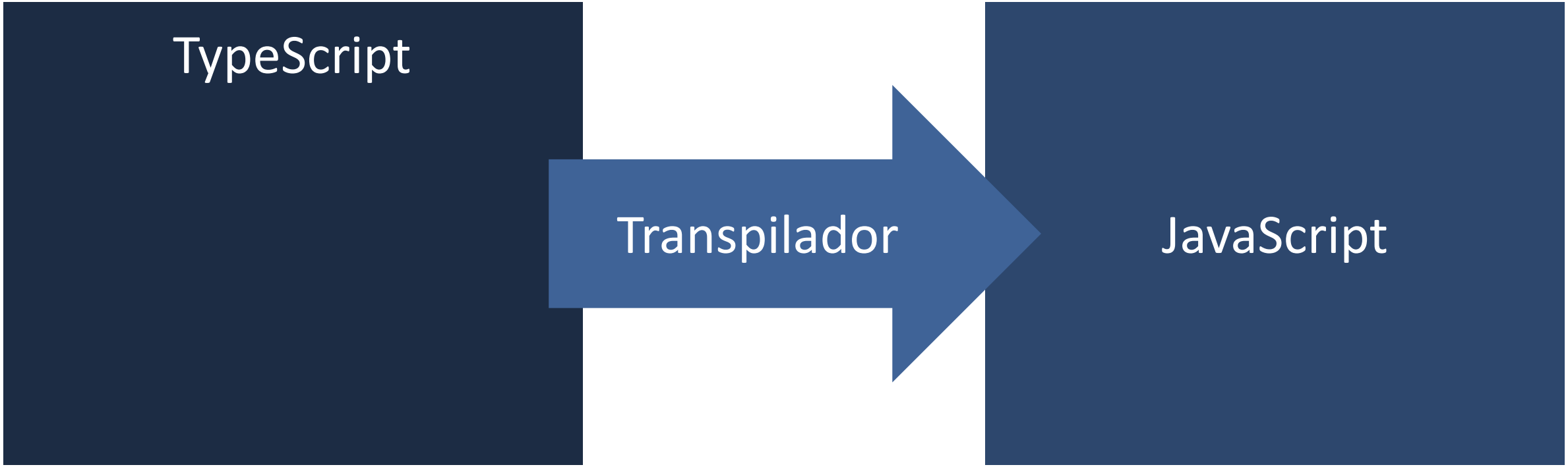
TypeScript, Definición

TypeScript

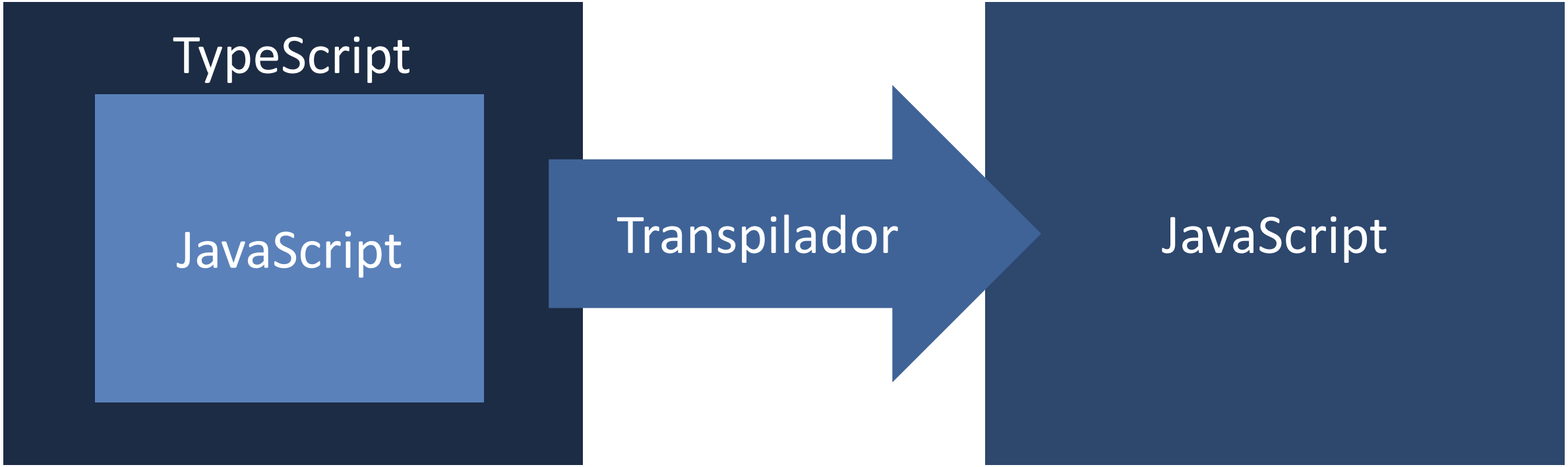
TypeScript, Definición



TypeScript, Definición



TypeScript, Definición



Clases, interfaces y herencia

DEMO

GALAXY
TRAINING

GRACIAS

POR SU PREFERENCIA

GALAXY
TRAINING