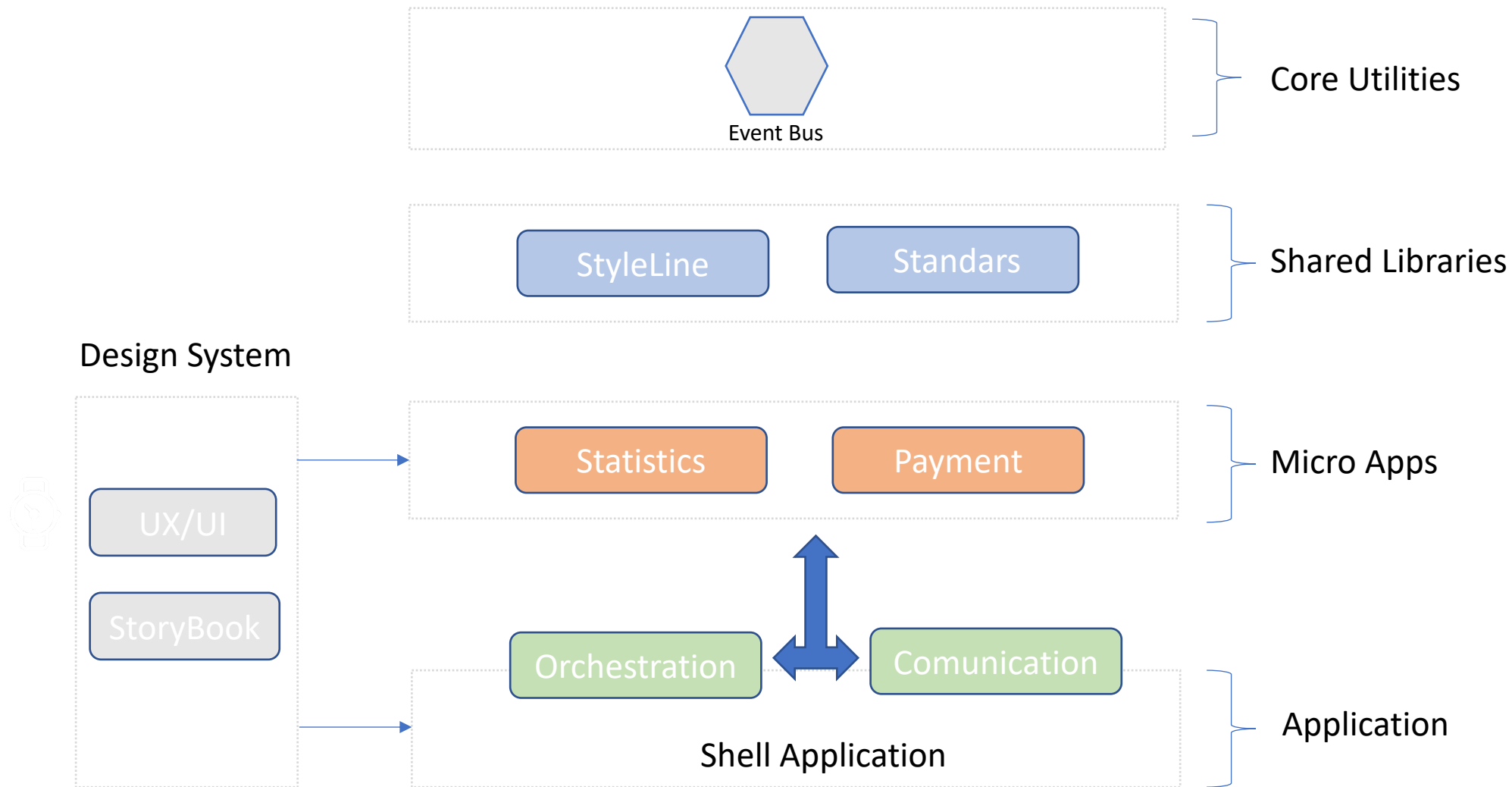


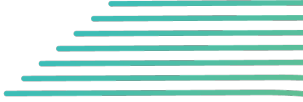
Propuesta Arquitectura FrontEnd

En el presente documento se explica la visión y los conceptos aplicados a una arquitectura Frontend desacoplada.

Enfoque de Arquitectura

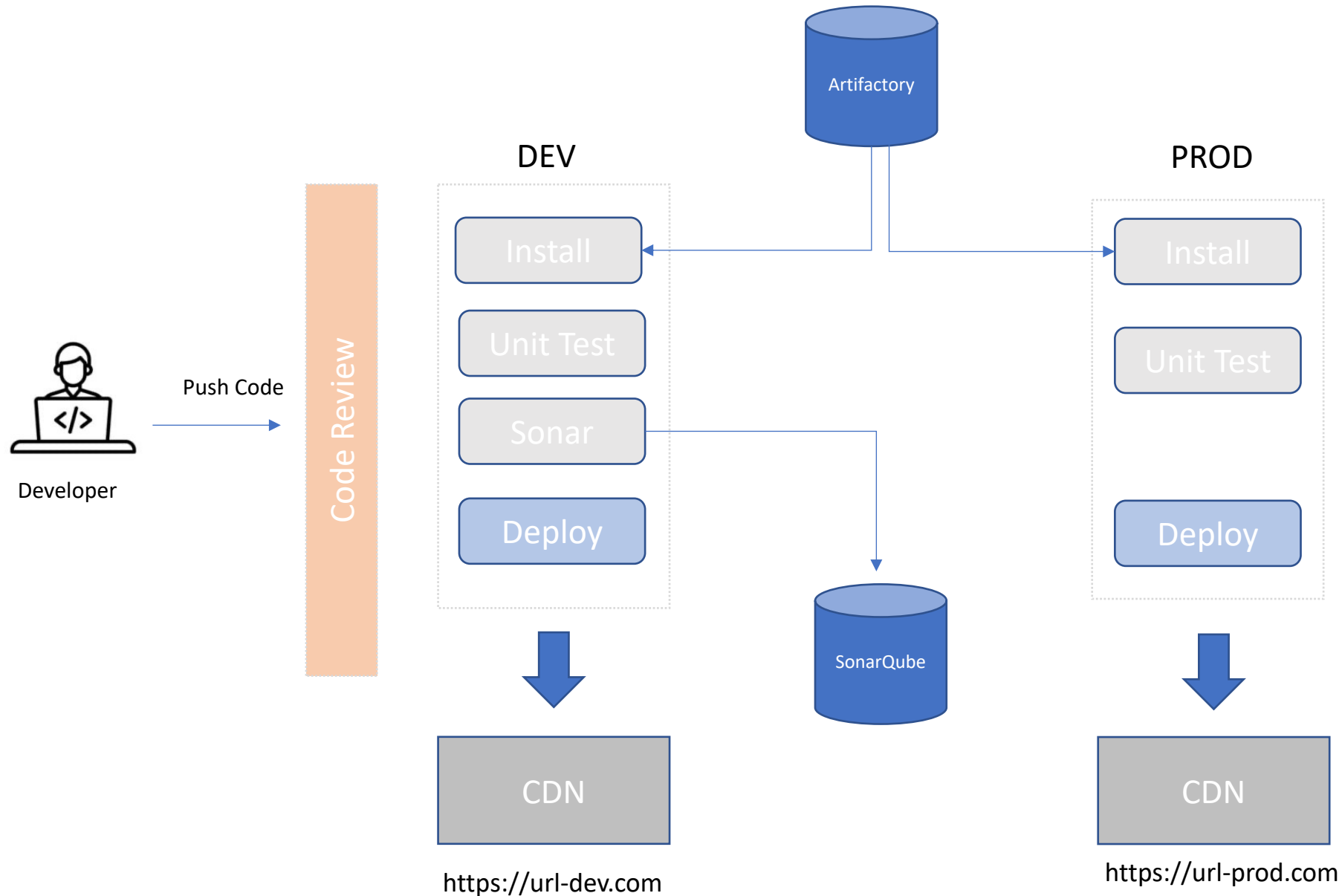


Enfoque de Arquitectura

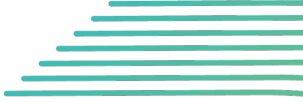


- ✓ **Core Utilities:** Es la capa que contiene artefactos con funcionalidades cross en todo desarrollo frontend, disponibiliza modulos como el EventBus que permite la comunicación entre aplicaciones.
- ✓ **Shared Libraries:** Se encuentran las definiciones de estandares como Linter y StyleLine, reglas de husky que seran aplicadas en los proyectos.
- ✓ **Micro Apps:** Representan aplicaciones desacopladas e independientes que encapsulan una funcionalidad de negocio específica, éstas cuentan con su propio repositorio y equipo de desarrollo.
- ✓ **Shell Application:** Es la aplicación que se encarga de la orquestación e integración de las MicroApps, en ocasiones establece comunicación con las MicroApps por intermedio del EventBus.
- ✓ **Design System:** Es la guia de estilo que esta representada en una librería de componentes visuales siendo estos últimos disponibilizados en un StoryBook, de esta forma los desarrolladores cuentan con documentación y ejemplos de uso.

Estrategia CI/CD



Estrategia CI/CD



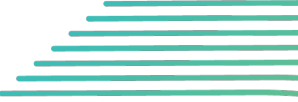
Es necesario contar con un flujo de CI/CD que agilice el proceso de delivery de software de calidad, en este sentido es indispensable cumplir con estándares y/o buenas practicas que garanticen tal objetivo.

- ✓ Disponibilizar un Artefactory donde se almacenaran las librerias frontend comunes por multiples proyectos.
- ✓ Habilitar SonarQube para evaluar constantemente la salud del software desarrollado.
- ✓ Definir estándares de desarrollo y reglas de antipatrones al equipo de desarrollo, todo nuevo desarrollo debe ser validado automaticamente antes de que éste sea subido al repositorio (previo al push).
- ✓ Establecer un gitflow que este acorde a las necesidades del proyecto.
- ✓ Implementar Code Review para garantizar y fomentar el cumplimiento de buenas prácticas.
- ✓ Contar con un ambiente preproducción para escenarios de hotfix.

Descripción de etapas:

- ✓ **Install:** Etapa donde se instalan las dependencias (privadas y públicas) necesarias para el correcto funcionamiento del proyecto.
- ✓ **Test:** Etapa donde se ejecutaran las pruebas unitarias para garantizar un minimo % de coverage establecido.
- ✓ **Deploy:** Etapa que disponibilizara las aplicaciones en un CDN.

Estrategia de testing



En un ecosistema de aplicaciones desacopladas la implementación de pruebas unitarias resulta mas acotado, esto facilita el cumplimiento de dicha práctica para los desarrolladores.

En las pruebas E2E de una aplicación Shell o microApps es factible usar el framework Cypress el cual ofrece las siguientes características:

- ✓ Compatible para la integración en procesos CI/CD.
- ✓ Permite capturar videos.
- ✓ Ofrece un dashboard en la nube con el histórico de las pruebas ejecutadas.

Arquitectura Frontend

El uso de arquitecturas orientadas a microApps, ofrece los siguientes beneficios:

Despliegues Independientes, Testing Segmentado, Equipos Autónomos, Actualización Incremental, Convivencia Legacy Apps

