# Enterprise Inventory

Se implementa como decisión de arquitectura el patrón Enterprise Inventory ya que la presentación de forma independiente de servicios legados representa un riesgo si se tienes implementaciones inconsistentes independientes, por tal razón estas implementaciones de exponen de forma estandarizada dentro de un inventario de servicios obteniendo composición y abstracción de servicios promoviendo la interoperabilidad entre ellos.

# Data Format Transformation

Se implementa como decisión de arquitectura el patrón Data Format Transformation, debido a la necesidad de homologar la respuesta y las peticiones de los orígenes de datos externo proveniente de endpoints expuestos por los proveedores de facturación. Este patrón ofrece como beneficio perdida de acoplamiento e interoperabilidad de tecnologías al tener contratos estandarizados y la transformación de formatos.

# Intermediate Routing

Se implementa como decisión de arquitectura el patrón intermediate Routing debido a la necesidad de crear logia de enrutamiento y despacho debido a que nuevos enpoints pueden ser incluidos y se debe poder configurar este nuevo endpoint, o en otros casos es posible que endpoints existentes cambien de ubicación y genere la necesidad de configurar nuevamente tu dirección de consumo, permitiendo una forma rápida adaptación a los cambiantes requisitos empresariales que puedan surgir en los convenios.

# Legacy Wrapper

Se implementa como decisión de arquitectura el patrón Legacy wrapper debido a la necesidad de desacoplar al consumidor del servicio con los servicios expuestos por los convenios, permitiendo estandarizar por medio de un componente de transformación, los contratos con los consumidores, esto permite la abstracción completa de características heredadas de los convenios, y proporciona la libertad de evolucionar o sustituir los servicios expuestos por algún convenio por otro con el mínimo impacto en los consumidores de servicios existentes.

# Metadata Centralization

Se implementa como decisión de arquitectura el patrón metada Centralization debido al riesgo constante de construir funcionalidad ya existente y a la necesidad de centralizar la definición de la metada de los servicios en un medio formal de registro de servicios para facilitar su descubrimiento.