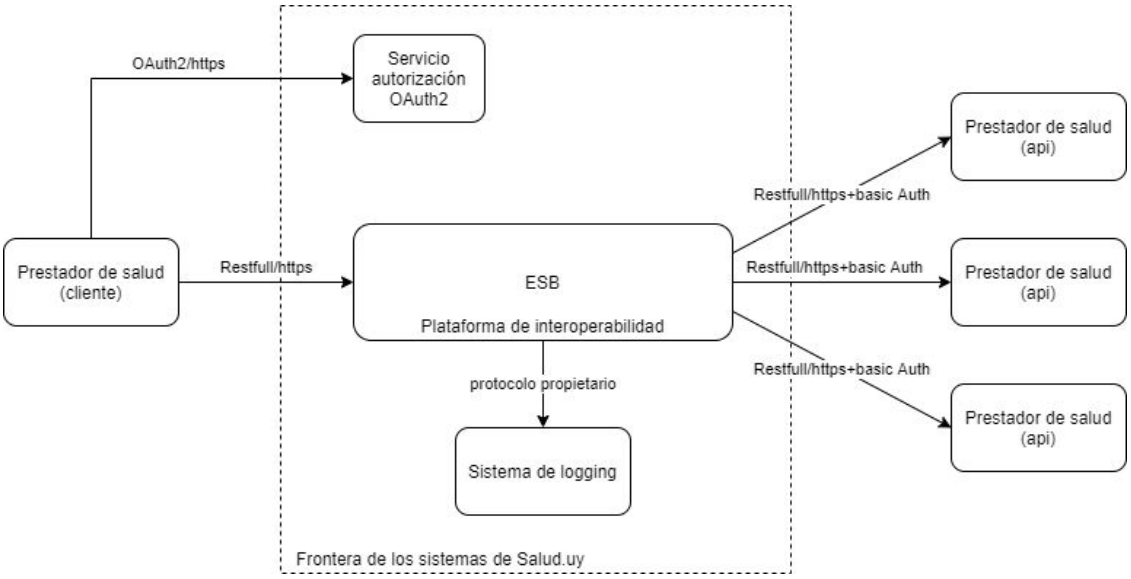
Facultad de Ingeniería – Instituto de Computación
Introducción al Middleware
Solución Evaluación Escrita – 27 de Noviembre de 2019

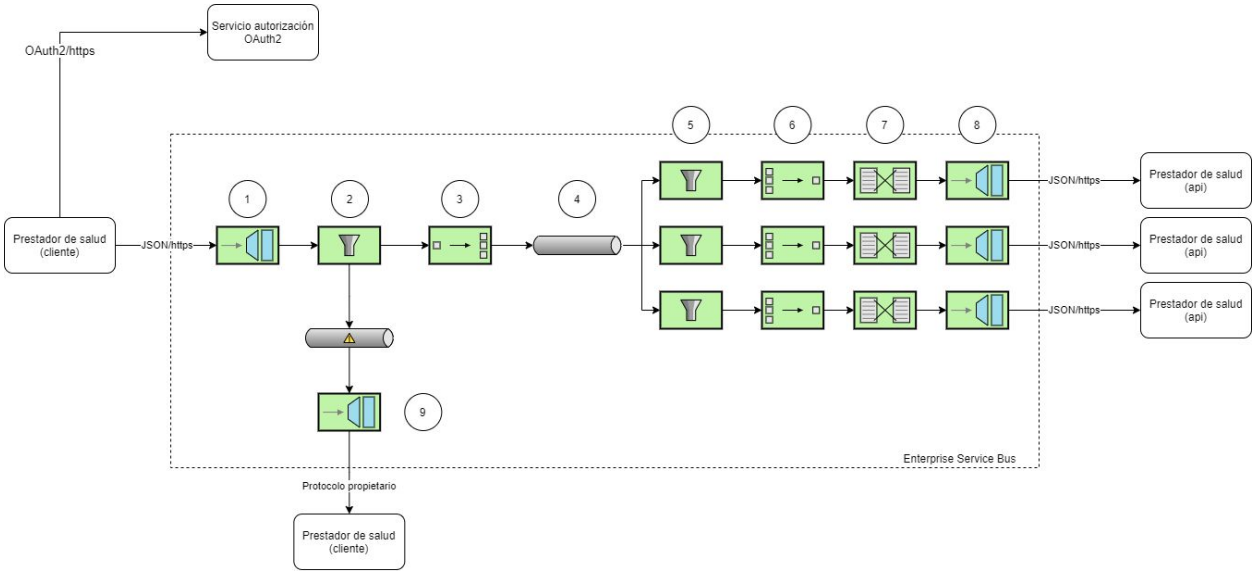
Preguntas (60 puntos)
Ver teórico.

Problema (40 puntos)
En alto nivel, la solución se compone de:

- Un servicio de autorización OAuth2 encargado de emitir los tokens de seguridad JWT a las aplicaciones de tipo Prestadores de salud. Utiliza como protocolo de transporte http con TLS/SSL y utiliza http basic authentication para autenticar a cada prestador de Salud.
- La plataforma de interoperabilidad de Salud.uy implementada con un ESB, la cual provee una Restfull para recibir las entradas de las historias clínicas de los prestadores de salud. Esta API requiere de un token JWT emitido por el OAuth2 server de Salud.uy. Se comunica con las APIs Restfull de los prestadores de salud para notificarles las entradas recibidas. La comunicación con esta API es vía http con TLS y se utiliza como mecanismo de autenticación http basic authorization. Se comunica con el sistema de logging para la notificación de mensajes con formato no válido, utilizando un protocolo propietario.
- Prestadores de salud con rol cliente encargado de enviar las entradas en la historia clínica de un paciente. Existirá un componente de este tipo por cada prestador de salud que notifica entradas de historia clínica. Esta aplicación se comunica con el ESB vía una API Restfull utilizando http con TLS como protocolo de comunicación. A su vez, se comunica con el servidor OAuth2 para solicitar tokens JWT y lo hace vía http con TLS.
- Prestador de salud con rol API encargado de recibir las entradas desde la plataforma de interoperabilidad. Expone una API Restfull con http, TLS y basic authorization.
- Sistema de logging: encargado de recibir los mensajes no válidos recibidos por la plataforma.



A continuación se describe en detalle el diseño interno de la integración para despachar un trámite.



Componentes:

1. Gateway: Se implementa el patrón ESB de conectividad Gateway. El Gateway se encarga de verificar que todos los pedidos hacia las APIs Restfull venga con un token JWT firmado por el servicio de autorización OAuth2.
2. Filtro: encargado de validar que los mensajes vengan con el formato de entrada solicitado. En caso contrario manda el pedido a un invalid message channel
3. Splitter: Encargado de dividir el mensaje. Se genera un nuevo mensaje por cada entrada de historia clínica del mensaje original.
4. Publish/Subscribe Channel: canal de tipo P&S encargado de notificar a cada uno de los prestadores de salud. Existirá un suscriptor por cada prestador de salud integrado a la Plataforma.
5. Filtro: encargado de filtrar aquellas entradas de historia clínica generado por el propio prestador de salud.
6. Aggregator: encargado de agrupar las entradas en un único mensaje con el número de entrada definido por el prestador de salud. Ejemplo, 100.
7. Transformador: encargado de transformar el mensaje al formato requerido por el prestador de salud.
8. Conector de API Rest: encargado de comunicarse con la API Restfull del prestador de salud. Posee la configuración para la autenticación con basic authentication.
9. Conector con sistema de log: Encargado de tomar los mensajes del invalid message channel y enviarlos al sistema de logging

Variantes:

- En lugar del patrón Gateway se podría haber utilizado un endpoint Restfull con OAuth2 para que haga la misma validación.
- En lugar de utilizar un canal Publish/Subscribe se podría haber utilizado un Recipient List. El uso de este componente evita el filtro luego del P&S al incluir en el Recipient List la condición de no enviar si el dueño de la entrada es el prestador destino.