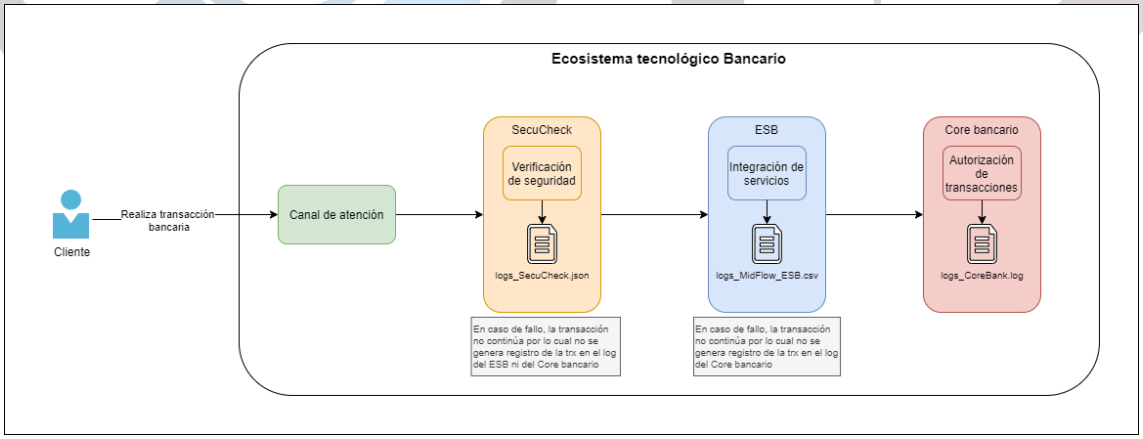
Hackathon: LogView360

Glosario:

- **ESB**: Es una plataforma de integración que permite que diferentes aplicaciones se comuniquen entre sí de forma flexible.
- **Core Bancario**: Es el sistema principal de un banco que gestiona las transacciones esenciales (transferencias, depósitos, retiros, etc) de los productos (cuentas de ahorro, corrientes, etc)
- **TimeStamp**: Es una etiqueta que indica la fecha y hora exactas en que ocurrió un evento, útil para registrar y ordenar sucesos en sistemas informáticos.
- **Latencia**: Es el tiempo que tarda un dato en viajar desde el origen hasta el destino. Menor latencia significa mayor velocidad de respuesta.
- **Dashboard**: Es una pantalla o panel que muestra visualmente información clave, como métricas o indicadores, para facilitar el monitoreo y la toma de decisiones.
- **Observabilidad**: Es la capacidad de un sistema para ser medido, monitoreado y entendido desde el exterior, usando métricas, logs y trazas para detectar y diagnosticar problemas.

Diagrama general de la situación actual:



Objetivo general:

Desarrollar una solución orientada a observabilidad que integre, procese, analice y visualice el recorrido completo de transacciones bancarias, desde su validación de seguridad inicial, su paso por el ESB y su ejecución final en el Core Bancario. La solución debe ayudar a entender el comportamiento de las transacciones, detectar inconsistencias y generar visualizaciones útiles para monitoreo y análisis.

Contexto de las Aplicaciones

Aplicación	Descripción	Archivo de log
SecuCheck	Sistema de validación de seguridad. Evalúa y aprueba o rechaza transacciones antes de enviarlas.	logs_SecuCheck.json
MidFlow ESB	Enrutador de servicios. Recibe transacciones aprobadas y las direcciona al core.	logs_MidFlow_ESB.csv
CoreBank	Sistema de administración de productos bancarios. Ejecuta la transacción financiera.	logs_CoreBank.log

Descripción detallada de los archivos de logs

Archivo: logs_SecuCheck.json

Formato: JSON
Origen: Sistema de validación de seguridad (SecuCheck)
Propósito: Verifica si una transacción es segura antes de enviarla al ESB.

Estructura:

- timestamp: Fecha y hora en que se realizó la validación.
- transaction_id: Identificador único de la transacción.
- user_id: ID del usuario que intenta ejecutar la transacción.
- ip_address: Dirección IP desde la cual se originó la transacción.
- resultado_validación: Resultado de la validación ("Aprobada" o "Rechazada").
- motivo_fallo: Motivo de rechazo si aplica (vacío si fue aprobada).
- módulo: Subsistema que originó la transacción (por ejemplo: api, web, mobile).
- verificaciones_realizadas: Lista de validaciones aplicadas (por ejemplo: token, blacklist, etc.).

Archivo: logs_MidFlow_ESB.csv

Formato: CSV
Origen: Enterprise Service Bus (MidFlow)
Propósito: Registra el intento de enrutar la transacción al core bancario.

Estructura:

- timestamp: Fecha y hora del evento (request o response).
- nivel_log: Nivel del log ("INFO" o "ERROR").
- transaction_id: ID de la transacción.
- direction: Dirección del evento ("request" o "response").
- operation: Tipo de operación ("retirar", "consignar", "transferir").
- status_code: Código de estado (por ejemplo: 200, 500).
- latency_ms: Tiempo de respuesta en milisegundos (solo se registra en response).
- user_id: ID del usuario.
- ip_address: IP del cliente.
- modulo: Subsistema que originó la transacción (api, web, mobile, etc.).

Archivo: logs_CoreBank.log

Formato: Log de texto plano
Origen: Sistema central bancario (CoreBank)
Propósito: Registra la ejecución final de una transacción financiera.

Estructura:

- Fecha y hora: Timestamp del evento.
- Nivel de log: Siempre "INFO" en este caso.
- Módulo: Módulo origen de la transacción (entre corchetes).
- Usuario e IP: ID del usuario e IP de origen.
- Detalles de la transacción: Entre paréntesis, incluye el transaction_id, tipo de transacción (transferencia, retiro, etc.), tipo de cuenta (ahorros o corriente), estado (Completada) y el valor.

Cada transacción se identifica por su transaction_id, y aparece en uno, dos o los tres sistemas, según su flujo y éxito.

Retos propuestos para los equipos

Módulo 1: Preprocesamiento de logs

- Convertir los tres formatos en una estructura homogénea.
- Unificar por transaction_id, manteniendo trazabilidad por timestamp.
- Estandarizar campos como estados, usuarios, IPs, etc.

Módulo 2: Visualización

- Mapa de flujo de cada transacción por sistema.
- Tiempos de latencia por etapa.
- Resultados de validación vs ejecución real.
- Transacciones rechazadas o fallidas y sus causas.
- Identificación de transacciones que no deberían estar en los logs (si las hay), ej: una transacción que falló en el ESB y se encuentra registro también en el Core bancario

Módulo 3: Detección de inconsistencias

- Transacciones aprobadas en SecuCheck pero no ejecutadas.
- Transacciones duplicadas o con latencias anormales.

Objetivos específicos

- Identificar y visualizar rutas de transacciones completas.
- Detectar problemas de seguridad, fallos de integración o errores del core.
- Hacer reporting visual del comportamiento por usuario, tipo de operación, módulo o IP.
- Preparar dashboards con las herramientas de visualización de datos deseadas.

Características

Duración estimada: 12 horas
Equipos: Hasta 3 personas

Descarga de archivos

- Los archivos logs estarán para la descarga de los participantes en el repositorio compartido por la organización en GitHub

Entregables esperados

- Código fuente (notebooks, scripts o app).
- Archivo consolidado con las transacciones unificadas.
- Video subido a youtube con el funcionamiento de la solución.
- Documento técnico que contenga:
    - Diseño técnico (Diagrama con componentes desplegables, integraciones entre ellos, funcionalidad de cada componente)
    - Hallazgos y decisiones tomadas.