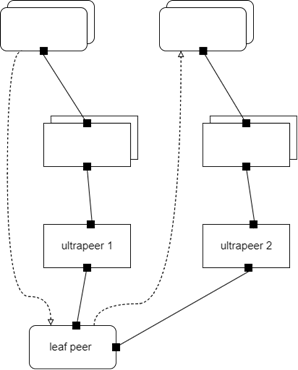
1. Sistema de archivos distribuido

Debido a que cada nodo debe poder funcionar tanto en modo cliente como servidor, el patrón arquitectónico más adecuado para el problema es el peer to peer. Esta arquitectura es útil para descentralizar la información, que es buscada en todos los nodos. Es decir, para poder compartirla entre dos o más usuarios de manera directa. Una de sus ventajas es la escalabilidad ya que puede crecer horizontalmente de forma indefinida a nivel de usuarios. Además proporciona robustez al sistema.

Una posible alternativa para integrar el sistema es el patrón call & return. La ventaja es que proporciona un conjunto de servicios que pueden ser invocados por otros componentes.

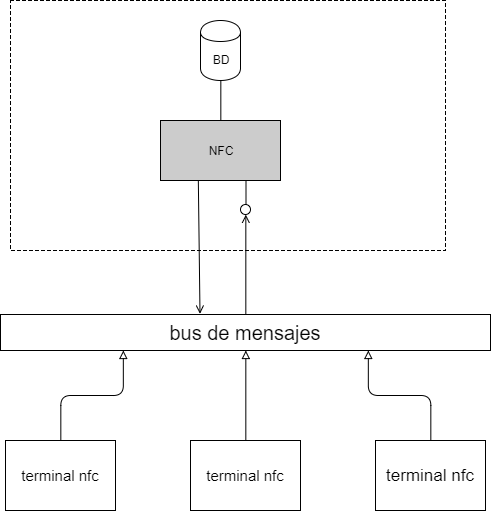


Otra alternativa podría ser el patrón bus de mensajes. Las ventajas de este patrón son desacoplamiento, confiabilidad y eficiencia ya que tiene alta frecuencia de envío de mensajes. Como desventajas se tiene programación compleja, mayor costo de desarrollo para la implementación y posible latencia en el envío de mensajes.

1. NFC

Como la solución debe ofrecer flexibilidad para integrar diferentes fabricantes de terminales, que posiblemente puedan usar tecnologías diversas, se recomienda utilizar un estilo SOA, ya que permite interoperabilidad entre las mismas.

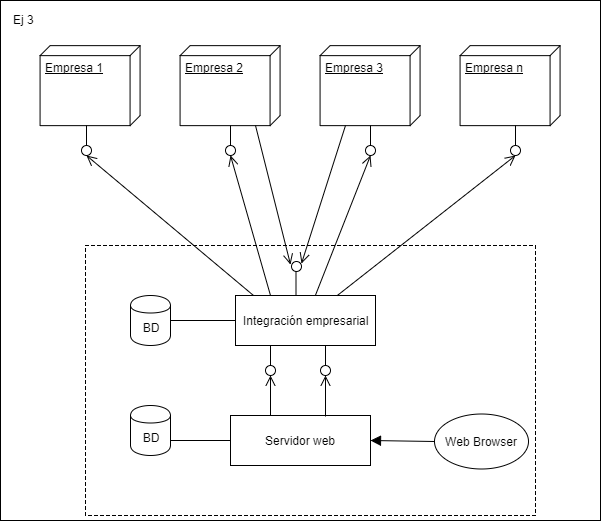
Este estilo se puede integrar con un bus de mensajes, que satisfaría el hecho de necesitar alta frecuencia de operaciones, soportar múltiples formatos de representación de los datos y permitir el desacoplamiento entre terminales. Además, como es un patrón asincrónico, no realiza bloqueos para esperar a las respuestas y es posible el envío de mensajes de manera publish subscribe.



1. Integración de sistemas empresariales

Como el requerimiento principal del sistema es la interoperabilidad entre diversas tecnologías se aplica la arquitectura orientada a servicios (SOA). La principal ventaja es la ya mencionada, porque es un estilo distribuido que toma a las distintas aplicaciones o sistemas como servicios independientes que pueden invocarse entre sí a través de interfaces. En este caso se aplica para las distintas empresas cuyos sistemas deben integrarse. Ya que todos cuentan con acceso a internet a través de HTTP, el conector ideal es call-return, cada aplicación se puede comunicar con un servidor central a través de servicios tanto REST como SOAP, ambos sincrónicos. La desventaja de este modelo de integración es que se tiene un cuello de botella en el servidor web, ya que si llega a fallar termina siendo el único punto de falla.

Por otro lado, podría usarse un bus de mensajes dado a la necesidad de, en algunos casos, realizar conversiones en el soporte de representación de datos, y favorecer la comunicación entre los distintos nodos, pero como desventajas puede llegar a tener latencia y es mucho más costosa su implementación.



1. Tráfico Aéreo

El enunciado está haciendo referencia al estilo arquitectónico cliente-servidor ya que la consola de control de tráfico aéreo debe recuperar los datos de forma sincrónica.

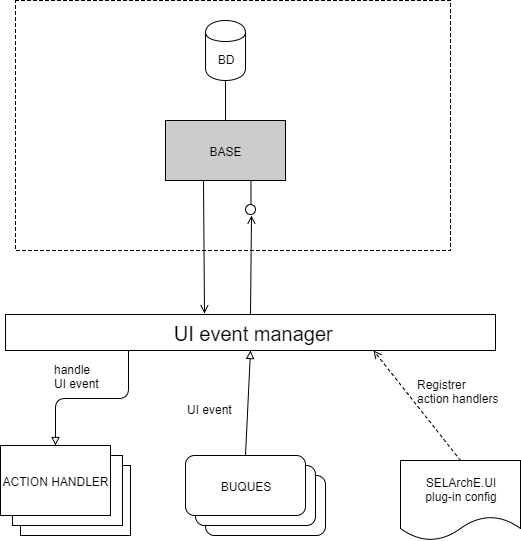
El patrón que puede dar una buena integración a este tipo de sistema es el call&return. De esta forma cuando el cliente realiza una petición (call), se bloquea hasta que recibe una respuesta (return), para garantizar consistencia en los datos, dado que el problema es crítico.

Además, debido a la criticidad del sistema, a través de este patrón, se garantiza la posibilidad de gestionar errores y proveer seguridad en el acceso al servidor.

1. Marina mercante

Es crítica la integridad en el sistema tanto desde la base a un buque como viceversa, por eso se utiliza el patrón de arquitectura centrado en datos, el cual proporciona facilidad para tal efecto. Un DBMS se encarga de mantener dicha integridad en los informes y mediciones. Además, una ventaja de este estilo es que proporciona escalabilidad en el sentido de que si se requiere mantener mediciones con un nuevo buque simplemente debe conectarse a la base.

Se utilizará el patrón de integración de eventos porque cada vez que desde la base se necesite hacer un informe meteorológico a todos los buques que se encuentren en la misma región, lo puede enviar de la forma publish/subscribe.



1. FinTech

En esta solución las fuentes tienen que estar aisladas entre sí porque en algunos casos los servicios de información son personalizados, es decir, pueden estar diseñados en distintas tecnologías. Por eso, el patrón arquitectónico que se adapta es el de microservicios. Al dividir una aplicación monolítica en pequeños servicios aislados, se permite elegir la tecnología que dará soporte a cada uno, teniendo su propio proceso. La ventaja es que es una versión simplificada del estilo SOA, en consecuencia, menos costosa.

Una alternativa de integración podría ser el patrón bus de mensajes, facilitando de esta forma la comunicación entre los distintos sistemas de inversores, soporte de envío de mensajes con alta frecuencia, transferencia inmediata y con la posibilidad de utilizar diferentes formatos de mensajes.

