

Ilustración 1: Art Attack by Martín Moreyra

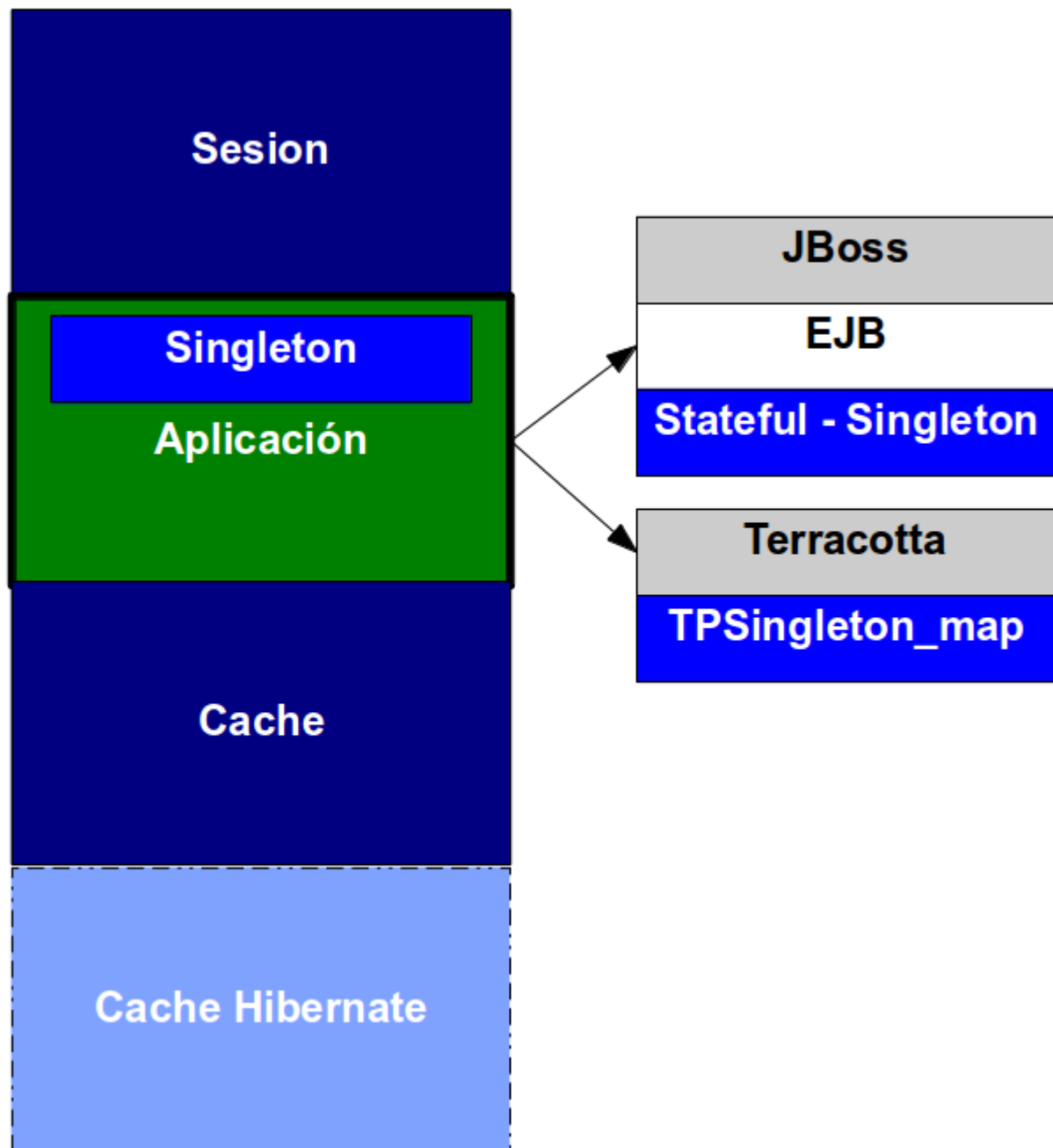


Ilustración 2: Esquema de análisis y comparación

Se estableció que a partir de las distintas configuraciones que vamos a adoptar para sendas implementaciones, se pueden definir 4 áreas de comparación

- Session: por ahora tenemos solamente el id del usuario, si se nos ocurre qué otra cosa se puede sumar, mejor. Por ejemplo, se podría sumar el nodo que atiende primero a el usuario, el “origen”; la session debería ser clusterizable sin problema alguno tanto por Jboss como por Terracotta
- Aplicación: el nivel de aplicación es el más sensible. No hay una comparación, digamos, “directa”. Lo que sí podemos comparar, es el mecanismo y sus resultados. A nivel aplicación, el Jboss permite sin problema alguno clusterizar el Singleton con acceso a nivel contenedor. Terracotta por su lado no permite esto; un escape para esta a priori traba, es trabajar con el concepto de servidor de Terracotta. Un servidor de terracotta, es un nodo del esquema extendido y completo de clusterización, en que solamente se aloja configuración

(xml) que determina qué es lo que se clusteriza; lo que se clusteriza se aloja en nodos denominados clientes de Terracotta. Por ende, lo que se puede hacer, es configurar el deployment de toda la aplicación, sobre Terracotta, para que se clusterice el map que soporta la sincronización de textos; tal vez se podría usar la PC en que estará la base de datos, total la latencia de red es despreciable. Mal o bien estamos hablando de clusterizar una entidad static que encima no forma parte de la sesión; se mantiene cierto optimismo al respecto, pero igualmente hay que verlo en acción. Supongamos por el momento que esto es viable; de todas maneras si no lo fuera, se podría hacer alguna manganeta para subirlo a sesión. Pero bueno, apostemos a que sale perfecto de una

- Cache: no hablamos nada muy particular al respecto
- Cache de Hibernate: inicialmente se supuso que posiblemente, habría que desactivar el caché de nivel 2 de hibernate tanto en Jboss como en Terracotta, en caso de que en Terracotta (se sobreentiende que en Jboss tanto por la potencia de este como por ser Hibernate “el framework de persistencia de la casa“, no hay problema alguno con la persistencia) se detectaran falencias al intentar sincronizar en el esquema de clusterización el cache más global de la persistencia, pero posteriormente se comprobó que se tiene acceso sin problemas al caché de Hibernate usando Terracotta, mediante un módulo finalmente gratuito de Terracotta (ehcache-hibernate), cuya referencia estaba “algo disimulada” en la documentación de Terracotta

Todo esto implica que entonces, se debe modificar el código de la implementación de Jboss para que haya un solo EJB, el Singleton, de modo tal de que todo el código, menos el Singleton, pueda ser usado sin problema en ambas implementaciones. El Singleton, es lo que debe ser trabajado tanto para Jboss como para Terracotta. Para Jboss queda tal como está, para Terracotta hay que efectuar las modificaciones pertinentes para que sea una clase más con un map static, no soporta ninguna sofisticación ni asociada con EB ni con ninguna otra tecnología “supra-servlet container”

Se cree que atendiendo a todas estas modificaciones, se puede tener un escenario de comparación adecuado a los objetivos del TP, tratando de manera especial al singleton en cada caso

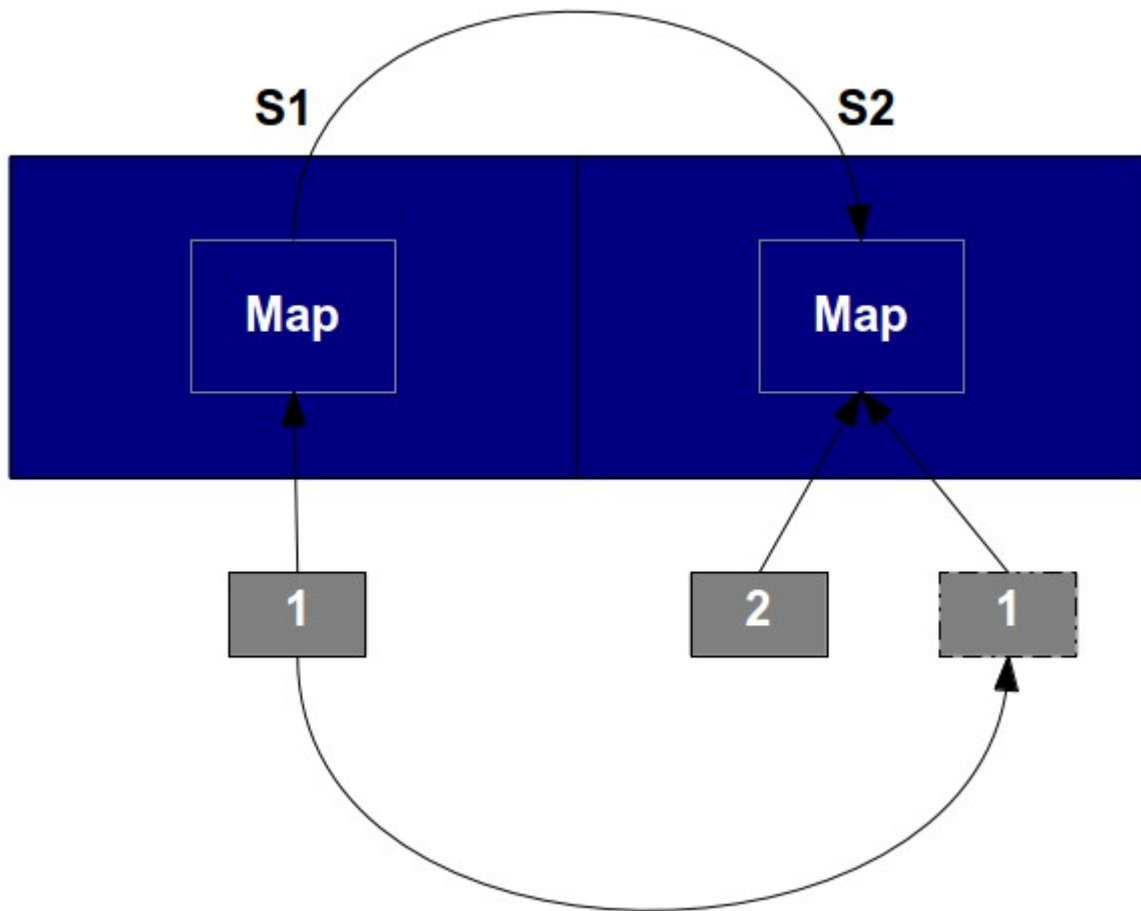


Ilustración 3: ¿Qué hará Terracotta al clusterizar un static?

Al intentar imaginar qué haría Terracotta al clusterizar una entidad static, llegamos a plantearnos que posiblemente, si no lo hiciera de manera correcta, intentaría copiar en un segundo cliente, la sesión entera del primer cliente, y de todas maneras ni así estaríamos tan seguros de lo que realmente está haciendo

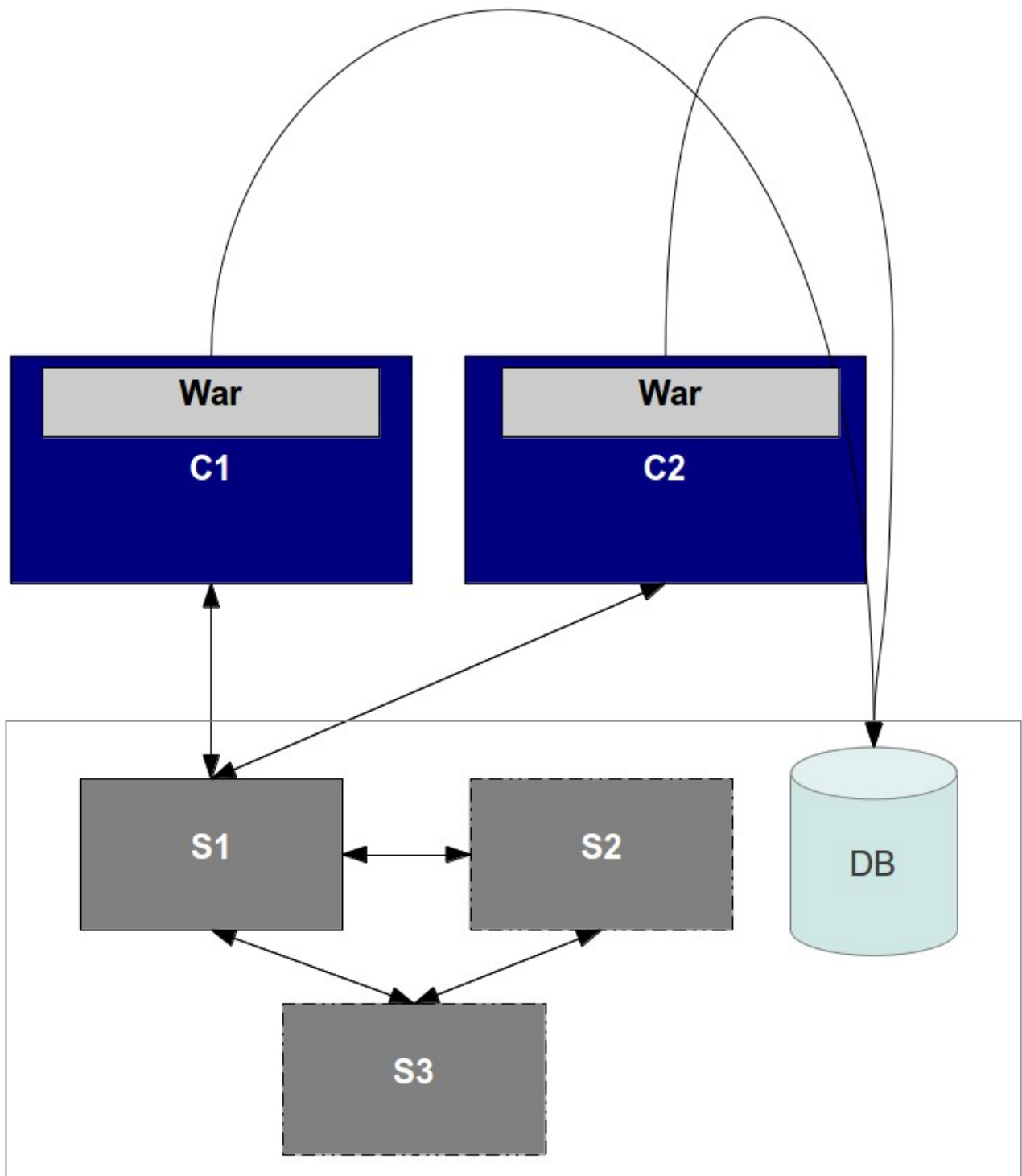


Ilustración 4: ServerArray de Terracotta

Finalmente se dilucidó qué significa ServerArray; hace referencia a la posibilidad que ofrece Terracotta, de soportar múltiples servidores (= xml) de Terracotta, para obviamente servir a N clientes de Terracotta; podría servirnos, quien sabe

HACER

- Diego B.
 - Modificar aplicación en Jboss tal como se explicó anteriormente
 - Probarla, para verificar que sigue levantando el cluster y que sigue andando
 - Sentar las bases, a partir del código de Jboss, del código de Terracotta
 - Vemos
- Martín
 - Seguir investigando Terracotta
 - Vemos