Da TM12 er skrevet i Java er det allerede designet til at eksekvere i et virtuelt miljø – Java VM’en. Det er naturligvis muligt at der findes en Cloud udbyder der giver mulighed for er PaaS hvor man simpelthen får stillet en Java VM til rådighed. Der findes microcontrollere der er designet specifikt til at eksekvere Java byte code (eller et udsnit af den), så hvorfor ikke en sky. Hvis vi derimod kikker på en anden leverandør kan der nemt være tale om et helt andet API, og applikationen vil derfor skulle redesignes til dette API. Jo mere moduleret, med logiske lag og høj lav coupling og høj cohesion, arkitekturen til TM12 er designet jo nemmere vil det være.

Google App Engine anvender Java, Python eller Go. For Python skriver google følgende:

* a complete Java 6 runtime environment in a secure sandbox environment
* based on common Java web technology standards, including servlets and WARs, JDO and JPA, java.net, JavaMail and JCache
* a [plugin for the Eclipse IDE](https://developers.google.com/appengine/docs/java/tools/eclipse) makes project creation, testing and deployment a snap
* supports other languages that compile to the JVM or use JVM-based interpreters, such as JRuby, JavaScript (Rhino), and Scala

Baseret på dette skulle det ikke være en større opgave at portere TM12 til Google App Engine – I hvert fald ikke in-memory versionen. Mongodb er ikke Java, og vil derfor ikke kunne eksekvere i Google App Engine. Det er muligt at eksekvere mongodb udenfor google app engine og bare tilgå den derfra. Hvis databasen også skal ind i under Google App Engine skal der anvendes App Engine's data repository i stedet for mongodb.

Den bedste måde at finde ud af det på er dog at prøve det. Først deployede vi et Hello World projekt til Google App Engine, og det virkede fint. Vi konstaterede at Google AppEngine også anvender Jetty, så det gør det nemmere.

API’et skal dog overholdet, og det kræver et entry point modul, ligesom at der skal anvendes App Engine's data repository – dette vil dog ”bare” være en endnu et persisteringsmodul på linje med in-memory og mongodb. Endelig skal resources filerne lægges ind under web.xml, som er Googles konfigurationsfil.

For sjov prøvede vi at lægge source koden ind under vores Hello World projekt, og det kan compile og eksekvere uden problemer på localhost (uden jetty.jar og jetty-util.jar, da de er med i Google API’et), hvis man ændre entry point. Dette er dog ikke muligt i en deployet version, der skal entry point være korrekt.

Et forsigtigt estimat vil være at en der kender både TM12 og Google App Engine API vil kunne gøre det på et par timer til den første hul-igennem test, og derefter måske 3-4 timer til oprydning.