

## Summary of existing TS implementations

Name	Technology	+	-	Additional Notes
Jini (or Apache River)	Java	Highly configurable	Heavy	<i>Needs specific JVM version</i>
<b>Blitz Project</b> (based on Apache River)	Java	Dashboard leasing <i>extends River Transactions</i>	Heavy Necessitates a JVM everywhere	
<b>Polyester</b> (based on LightTS)	Java	One thread for each Task => light and fast	Designed to run on one application only (not client-server)	
<b>Terrastore</b>	Java	Different APIs (HTTP and Java) Made for clustering <u>HTTP protocol</u>	Documents oriented	<i>Based on the Master-Slave scheme</i>
SemiSpace	Java	Not tested	Not tested	<i>Requires Maven installed on the computer</i>
Rinda (based on the Linda model)	Ruby	Integrated by default in Ruby 1.8	No real documentation	
Heterocera	RoR	Close to the spimes concept <u>HTTP protocol</u>	Needs to have a rails environment on the server	
PyLinda	Python	<u>Multiple TS</u>	Depreciated	<i>Requires the sudo rights to test</i>

## Legend

- **Tested at King's**
- Important feature