

**For the context:**

The usual approach for distributed variables is to handle them with a database. This means the causality and the message order is important and thus a consensus algorithm is needed.

A NEW approach is to use CRDT and peer-to-peer. With this approach, causality is automatically handled by metadatas and the message order has no impact, thus no consensus algorithm is required. This means no need for database server and much less cumbersome algorithms.

By the way, this solution is already adopted by some big companies such as Riot Games and TomTom.

**So globally how does it work?**

CRDT is for conflict free data type. It's a datastructure with values and metadatas. From time to time, nodes will send it to their peers.

The important notion here, is that it is CONVERGENT. Information is diffused everywhere and the message order is comutative thus every node eventually end up with the same state.

**Now let's talk about Lasp:**

Its an Erlang solution for distributed applications using CRDT but it's fully automated and thus the developer has very little control.

So my job is to allow more control via an API to visualize convergence time but also to modify convergence time. And finally, to use these tools to test different cases.

**Finally, one last notion I want to present to you is the relation between update speed and convergence speed.**

As you can imagine it is useless to have a super fast convergent system if we update the value only once per minute. In the opposite, if we update the value frequently but the system is very slow to convergent, it means high delays. Thus the relation between these two notions is also something I am trying to explore.