

# Lecture 14 - What is distributed consensus

---

All the different blockchain systems are really about Distributed Consensus.

The method for getting it different depending on how you go about it. The biggest distributed consensus system is Googles Spanner database based on F1 and BigTable.

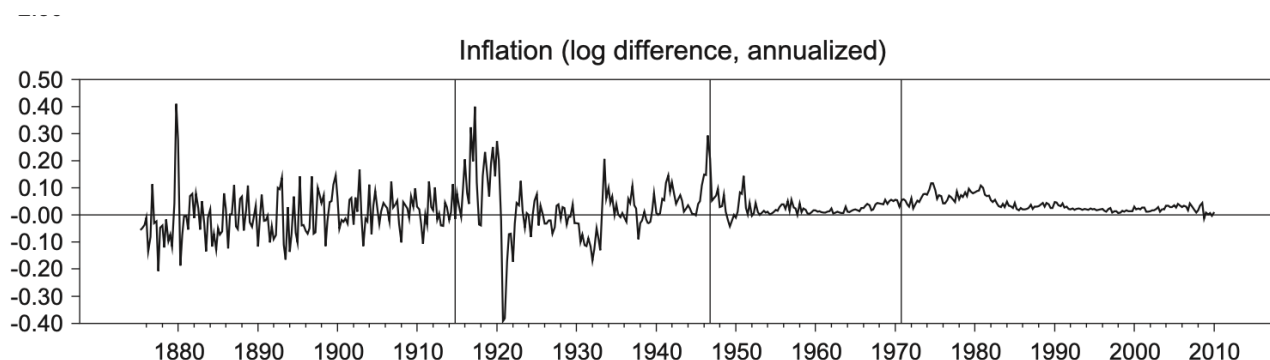
Let's talk a little bit about F1.

First world scale database.

Uses Atomic Clocks for Consistency. With blockchain we get a slower system - F1 is millisecond consistency - but it is not decentralized.

Benefits to decentralized is that there is nobody "controlling" the system. With the US dollar there is the FED that manipulates the supply and the value of the dollar. There are many bad examples of a central bank manipulating a currency - in the 1970's Argentina. More recently in Venezuela.

The US FED has done a good job.



(from White Paper)

You can see a similar kind of a graph when looking at commodity prices. Since the 1930 most commodity prices have been much more stable. The value of the dollar has been reasonably stable.

So... Not having a "FED" in a cryptocurrency means that there is nobody controlling the available supply and the price is left to the market to determine.

Back to consensus...

What is distributed consensus.

## Byzantine Generals Problem

---

This was originally described by L. Lamport [4] in 1982. It is the underlying problem and solution that is behind blockchain and cryptocurrency.

Basically you have multiple independent systems sending messages. Some of the systems can like. You want to get to a consensus where a set of data is agreed upon. This was originally developed for systems that control airplanes and is still heavily used in flight-controls. Real time systems like cars, planes and others use this today.

Lamport's "discovery" was that you can not assume that a system is honest. Systems can lie. They can receive accurate data and then send inaccurate data to others. They don't just fail and stay kited. They can fail and produce garbage data.

The real question is how do you assure that multiple systems or people that are at different locations agree on a "world state" before taking some action.

This is based on the Paper "Practical Byzantine Fault Tolerance" by Castro [5] in 1999:

1. Commander sends the value to every lieutenant
2. Every lieutenant
  - if it receives a new value  $v$ , broadcast (prepare,  $v$ )
  - if it receives  $2f+1$  (prepare,  $v$ ), broadcast (commit,  $v$ )
  - if it receives  $2f+1$  (commit,  $v$ ), broadcast (committed,  $v$ )
  - if it receivers  $f+1$  (committed,  $v$ ), broadcast (committed,  $v$ )
3. Ensure agreement
4. Ensure liveness under an loyal commander

## Clock solution to Distributed Consensus


---

This is L. Lamport's other contribution to distributed fault tolerance. A provable consensus system using accurate clocks. I think his paper on this was in 1978.

The commercial version of F1 [1] is "spanner" the worlds first global scale database. You can use it by using Google Compute Cloud database. It is the database under gmail, Google docs, and doubleclick - the advertising platform. This means that all of use it many times a day. 74% of the web has google analytics on it - feeding data - billions of rows a day into it.

At the hart of this is the Google TrueTime [2] API - you can use it also. True time is a microsecond accurate time clock in the API that is globally accurate. By globally I mean that the data center in Dublin Ireland will give you the same results as Portland Oregon. To do this Google used atomic clocks in its data centers and adjusted for the curvature of space time (General Relativity). It then went to substantial effort to get all of the clocks to agree.

This is only the 2nd application of Einstein's General Relativity in the world. GPS is the 1st.

 Google Atomic Clock In TrueTime API

When you change data a "leader" is picked. This is the system that will make the change. This is done if the changes is within a single block. The leader changes the block then distributes the block to all the systems that have a replica of the block.

The leader is picked based on who responds the fastest (Remember the clocks).

If the change spans more than 1 block then a coordinator for a 2-phase commit is picked in the same way. The coordinator manages the 2-phase over the set of single block commits.

In all cases the process chooses reliability over speed.

## References

---

1. Google's F1 Database Paper: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41344.pdf>
2. Google Spanner Database: <https://levelup.gitconnected.com/how-google-spanner-assigns-commit-timestamps-the-secret-sauce-of-its-strong-consistency-8bc143614f26>
3. TrueTime System: <https://medium.com/searce/internals-of-google-cloud-spanner-5927e4b83b36>
4. Byzantine Fault Tolerance: <https://people.eecs.berkeley.edu/~luca/cs174/byzantine.pdf>
5. Practical Byzantine Fault Tolerance: <http://pmg.csail.mit.edu/papers/osdi99.pdf>
6. Is the FED doing a good job: <https://www.fte.org/wp-content/uploads/WhiteSelginLastrapesFed1.pdf>