

Conference Call Notes 25 October

Summary

This week we went through the slide deck that Mike presented to the Open Financial Data Group the previous week, by way of a summary of where we are. We also focused in on the question “What kind of thing can be posted to a distributed Ledger?” The answer is, any of the kinds of thing that be stored in a database. AS with any solution, performance considerations should be kept in mind. In any case the DL is not a “ledger” in the sense of being postings of monetary amounts.

Detailed Notes

Posting to a Distributed ledger

Q: What is the kind of thing that can be posted to a DL?

A: Any data.

For some, a record that goes on the ledger is just a sequence of bytes.

So it can be anything.

Anything that can go in a database can go in a DL.

It is simply a database where you can't delete or update records you can only append records. And you can't query.

If the records you append to the ledger are data that takes th form of transferring of currency from one party to another then it si a monetary amount.

So it is not a ledger in the financial sense.

Q: Does it have datatype and if so what?

- there are containers within containers. Consider Smart Contracts. These are containerized, fully encapsulated code. That code can bejust about aything. Can incude what are loosely called oracles. These are coming out fo the serverless architecture world.

Once you get past the vocabulary, SCs = microservices and Oracles = serverless applications.

Relating this to databases:

If you look at databases that implement the SQL language then the data follows that. If Mondo B then it's JSON, if some other kind of database, it's binary.

So in DL in general, all of these arrangements are possible.

In BitCoin a very specific datatype goes on th ledger. highly structured, per BitCin requirements.

Ethereum has its own, sparate design decisions about the kind of data.

The ones intended as general purpose DLs, have general purpose kinds of data.

So, if we look at one specific architecture, e.g. Ethereum, it will have its own defined answer to these questions.

Q: What do we typically see?

With few exceptions (BitCoin most DLT systems are defined to be general purpose. So most of these will support general purpose data.

So for instance you can post for example a hash code, a whole doc in binary (but likely that performance limits will apply and there may be an upper limit to record size); you can post percentage amounts, currency amounts and so on.

What is true of DLTs is that whatever you post lots of copies will be made.

So it's a good idea not to consider posting binaries although in principle you could. Hence the motivation to put a hash to the thing.

Also keep in mind that a DLT is less efficient than databases, so always consider what really needs to be decentralized and what does not.

Then:

Design considerations for relating in and out of the ledger.

Example you can post a URI that points to wherever.

Smart Contracts

Smart contracts - would you post the whole of this to the DL or just the structural stuff?

Think of a SC as like a Stored procedure.

SC would determine whether a new record can be allowed on. Can also calculate, using as input stuff that's already been posted on the ledger.

So the SC does those 2 things.

The SC is executed on every copy that is in the decentralized system. Only works if it gives the exact same answer at each node.

The SC goes on to the BC since all the nodes need to execute the code. There are engineering questions around whether the code is executed within the SC or somehow else.

Legal contracts are not the same as the Smart Contract.

Clarification: given "it only works given the same information"; the SC can be built with event handlers, such that they can react to verifiable outside events.

The stuff outside of the dotted line as it were, is outside the Smart Contract. For instance when you have an agreed source for a stock price. That itself gets posted to the blockchain so once all the nodes have

verified that it's ok to post that price, it can be used by the calculations.

There is some variability in terms of how contracts have been written e.g. the Dow contract.
Mechanisms for validation, verification, mechanisms to treat certain events, including recursive

If the contract is written between 2 counterparties and they agree to all the code, then that contract operates almost autonomously at that point.

Contract in SC sense = Stored Procedure

See Gideon Greenspan blog post "Beware the impossible Smart Contract"

Comments on slides

See slide deck - summarises where we were at

Comment on the IR Swap Process Model

The calculation itself DOES happen on the DL.

Adjust the diagram.

SC is Stored Procedure so it happens inside the database of the server itself.

It would be unnecessary to do a calc and then have the DL validate it, since it would do the same calc

Reading e.g. LIBOR happens off-DL

External actions e.g. Execute Trade, happens off-DL

Actions

ACTION: split our model into:

- the business process as we know it - what happens on and off DL, which can be done in more than one way even on one DL.

Chat Log

Link to main wiki: <http://www.omgwiki.org/OMG-FDTF/doku.php>

Link to the DLT Page: 1:11 PM: <http://www.omgwiki.org/OMG-FDTF/doku.php?id=blockchain-wg-page>

Comments

Anthony B. Coates (DB) (to Everyone): 2:05 PM: We might need to be careful about trying to remove

players from the processes, given that regulators are adding players, e.g. clearers for OTC trades.

From:

<https://www.omgwiki.org/OMG-FDTF/> - **Financial Services DTF wiki**

Permanent link:

<https://www.omgwiki.org/OMG-FDTF/doku.php?id=notes20161025>

Last update: **2016/11/01 10:14**

