



DARCRUS

ENTERPRISE, EVERYWHERE.

Darcrus Whitepaper and Use Case Examples

Release Date 20 November 2016

Revised 16 December 2016

Version 1.1

<https://darcr.us>

Introduction

Darcus is introducing a new way of protecting your data, your customer's information, and ensuring your company will have the data you need when you need it with their custom dAppsheets. There are three distinct types of Darcus dAppsheet implementations, from the most private to most public:

- Private, Non-public – Confidential business data or intellectual property
- Private, Public – Authenticated usage, public verification
- Public, Public – Open for use and viewing by anyone or anything

We will give a use case example for each of the above implementations. For any of these implementations, we will utilize the underlying decentralized platforms such as Waves, NXT, Ethereum, and Bitcoin, we are able to seamlessly create copies of your database anywhere your company's Darcus node exist. Darcus and its' dAppsheet will create a transparent middleware utility to interface with the users and the blockchain.

Use Cases – Private, Non-public

For a private, non-public dAppsheet we could use NXT as the platform of choice. Realize, any of the above decentralized platforms can be used. Company XYZ needs to solve a disaster recovery item that was discovered in the latest compliance spot check. A dAppsheet is the perfect fit to meet this requirement and inherently gives additional features. A few of these features are:

- Near instant replication across all company nodes
- Near instant availability of data at all company locations
- Privacy of company data via transactional encryption
- Verifiability of data retained for future compliance audits

Imagine today's enterprise disaster recovery policy and plan. It involves backups, tape drives, hardware infrastructure, and possibly trusting a 3rd party with data and processes of their own. A company will also have Service Level Agreements and data at rest worries for assurance as the custodian of your customer's data. dAppsheet will give coverage on the above points by automatically submitting all transactions to the underlying platform to replicate these entries across the company nodes, enables remote locations to use the data, and cryptographically verifies the entry to ensure compliance. Encrypting of the transactional data is different per platform but will be available prior to that version (language compatibility) being released for public consumption.

Let's use the example of Company XYZ having a data disaster and loses all the data at Location A. Instead of trying (and praying) to get the backup tapes to work, just install the

Release Date 20 November 2016

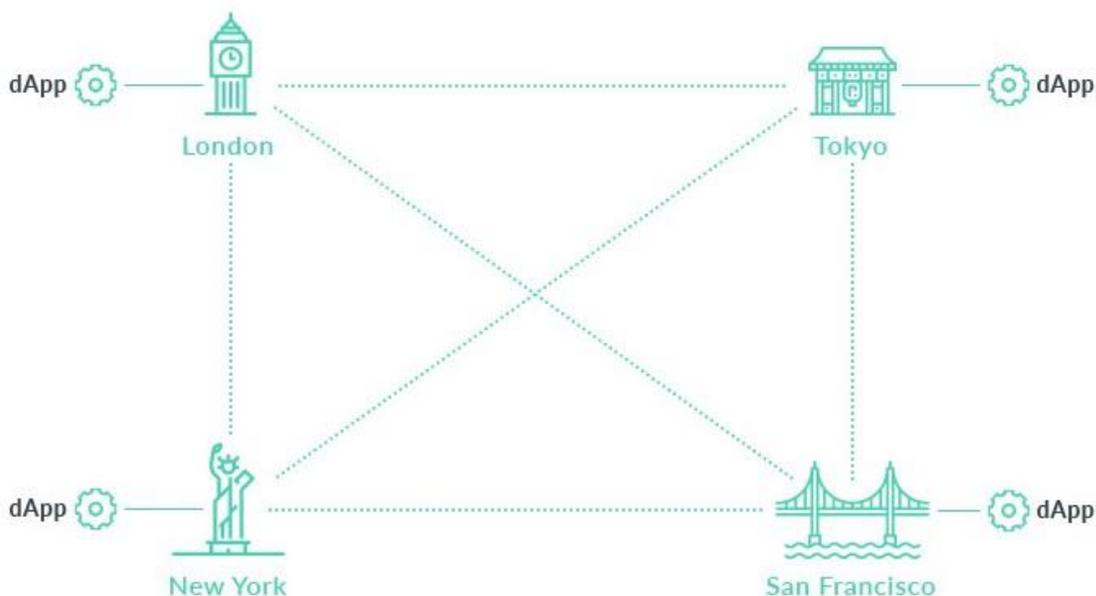
Revised 16 December 2016

Version 1.1

<https://darcr.us>

dAppsheets again. No worrying about rotating tapes, drives, securing all this data at rest, and then testing the validity of the backups. Storage consumption does not increase in any certain amount faster than a traditional database, only the management requirement goes down. The data is available at Site A as soon as the dAppsheet is synced. No intricate SQL settings or numerous patches and install guides to follow.

The nodes for Company XYZ are fully closed from the public. Nodes are only permitted to connect to each other, and use SSL to replicate data between each node. There is no public viewing of the ledger without authentication, authorization, and accounting.

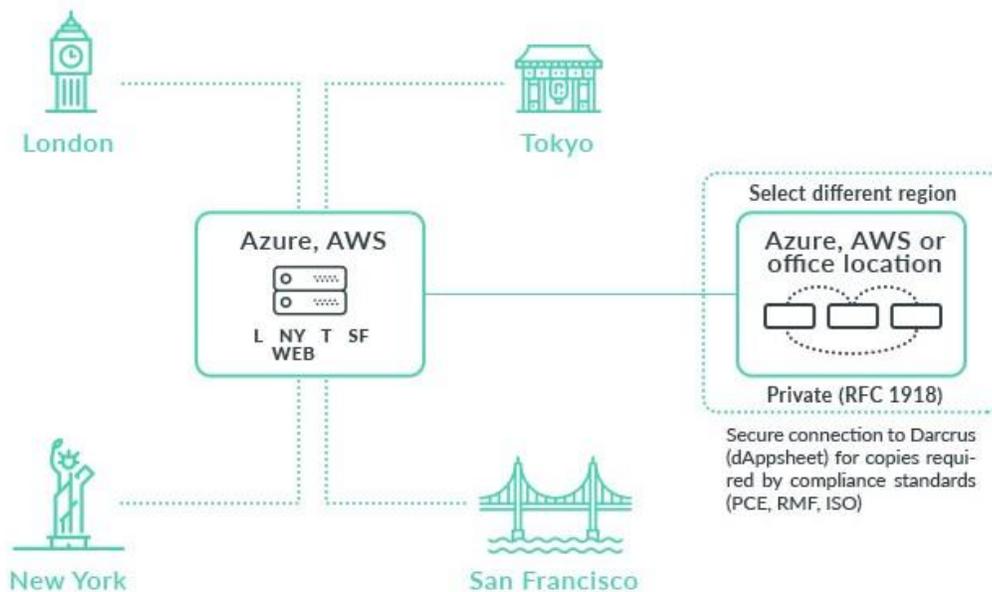


Use Cases – Private, Public

For a Private, public dAppsheets we could again use NXT as the platform of choice. As only the permissioning and viewability of the ledger changes in each example, so we will not revisit the feature sets each time.

A company that would want to use this type of dAppsheet could be an online gambling or dice site. This type of permissioned (you need an account to use, but anyone can make an account) ledger is available for public viewing and scrutiny. A customer for a site would sign up and their data is stored in the dAppsheet. While this customer places bets, the provably fair betting hash (PFBH) is transacted into the dAppsheet. When the bet is actually placed by clicking the 'Place Bet' button, the PFBH, the TXID of the PFBH TX, and the bet placed are all pushed to a verification TX. The outcome of this would be the

result for the bet, also being written to the dAppsheet. These transactions (or ‘saves’) would make provably fair bets even easier to expose to the customers of the site. Customer data is encrypted and anonymized while still providing the public proof the site operator is acting in full faith. No 3rd party trust is necessary for provable verification of bets.



From this graphic, imagine the bettors being in each location with the betting site having provable backups either on-site or in another cloud instance. No customer data or bets will be lost.

Use Case – Public, Public

For a Public, public dAppsheet, again all the same standard conditions apply. For this example, imagine a township that requires building permits for adding on a deck or room to your house. You simply go to the permit registration site, enter the data required (GUI built by our consultants), and publicly publish the permit data for the township to have. There is no paper copy to have on hand, or ‘I had it but I lost it’ to contend with. This example could look just like the private/public node’s diagram.

What does this mean?

All of the replication, protection, encryption data is always present in each dAppsheets implementation. We are aiming to greatly reduce the risks associated with provable data, encryption of customer data, and disaster recovery for businesses. I have seen backup procedures fail or get overlooked because today's IT staffs are over worked and underappreciated. Deploying dAppsheets in an environment cuts down on the manual human interactions for mundane but *important* tasks, taking care of customer and company data! Along with providing consultation services for deploying a full dAppsheets suite, there will be open source templates and examples for each platform we support. We welcome the community feedback to build onto the capability and extend the reach of dAppsheets.