Blockchain PSIG Call Notes

*29 October 2020*

# Attendees

* Dennis Schouten
* Peter Willemsen
* Frederic de Vaulx
* Mike Bennett
* Char Wales (Jackrabbitt, 265992)
* Robert Stavros (Jackrabbit, 569992)
* Rencher, Robert J
* Rob Nehmer

# Agenda

* Disposable SSIDs RFI sign-off
* LETS RFP
	+ SKALY introduction
	+ LETS RFP actions and next steps

# Meeting Notes

## RFI Sign-off Comments

### Section 2.1

CW has comments in 2.1 (errors spotted)

* One: one during the call
* Two: MB Still not happy with 2nd edit during the call
* 3rd one…
	+ There are other parts also fragmentary

### Other Comments

Anything else on the RFI from this week’s review?

* Nothing yet

CW still to go through the rest – send comments to MB via email.

Recall the 4 week rule doesn’t apply to this RFI. So take our time to get it right. Focus for 9 Nov 4 Week Deadline is on the LETS RFP.

### Section 3.1

MB: Section 3.1 was re-written in this pass. Would appreciate some eyes on this.

### RFI Review Outcome

When we are happy that the summary reads well it should be OK to submit.

Do another offline pass on this and sign off next week.

MB: Distribute Fri / Mon.

## LETS RFP

### SKALY Introduction

#### Dennis Schouten - Co-founder

Founded 2018. 3 founders. Formed to identify people standing on a weighing scale, using DLT.

Grant from IOTA Foundation for the TNO project (NL research institute, gov funded). Exploring Private Health data.

What are called ‘Private Trains’ where users of their own medical data can give their consent to different orgs, e.g. hospital, insurance company, physician. Data stays with the user not stored at a centralized server. GDPR considerations. Gives a global overview of your own personal health.

SKALY – from person on scales with ML, to something that makes sure something works on a protocol. Started with MAM (now Streams) from IOTA. Limitations with MAM Protocol. Does not do everything we need to do, e.g.:

* Send multiple data to another org

Developed Freighter – open source protocol, available on GitHub. Free to use.

Developing the platform, that will be used with TNO and on medical devices. In development but the 2nd layer protocol itself is complete.

Legal protections:

* Consumer health data NOT protected by law (except in India)
	+ E.g. fitbit, weight on a scale
* Medical data (patient data) IS protected by law

Bottleneck: with TNO, need a standard to be able to use medical data in the 2nd layer protocol. Hence conversations with MB on the IOTA Standards plans and timelines.

Need certification (standard) before we can go live with real data at TNO.

#### Pete Willemsen - Co-founder, CTO

Started with MAM. Issues for our use case. Used MAM v1 and later MAM v2.

Main problem (also in Streams): have to keep track of your own channel. When you send a message you have to update your channel. Was still the case in initial specs of Streams.

Also design of channels is quite complex. Did not suit us with healthcare data. When you download a msg from MAM channel, have to get msg, wait for it to be downloaded, then get the next. Link from each msg to the next. If a delay, can’t get the next data. MAM solves by making multiple channels, but still have the same issue.

Needed something where you don’t need to wait for each message in turn.

Have a channel with 1 key that never changes. Users never need to know they are using a decentralized protocol. Same as any other protocol. Same for Admins – just see data as if from a server.

Freighter works like BitTorrent – can download multiple bits. Can also use multiple IOTA Nodes, e.g. get 4 nodes to do the work. Makes it easier for us and the user.

In MAM, if I deleted the Skaly app, the channel was gone, needed to do something to recover it (had to back up the channel – to a central server, so not decentralized).

MAM: were not able to give users a set-up similar to any other app e.g. username, password – but without a central server.

Q: One per user?

* Wait to see the demo.

There are 2 versions of Freighter

Simplest channel – any other things like encryption you get by extensions

Real time downloading also a separate extension.

Creates a random key.

### Demo / Walkthrough

IOTA /MAM: Snapshots – data discovered from the Tangle. If you are posting some area of the channel and people not aware you are posting there, people can’t get your messages.

In Freighter, special thing in the protocol makes sure it is snapshot resistant. Essential for where users interact with it.

Freighter is not a competitor to MAM but aimed at a different area. Streams – machines focus (IoT). Freighter more end user friendly.

Have to copy paste channel address to someone else.

Freighter doesn’t use encryption. Uses game theory to add small checksum bytes. This makes every user of the channel the owner. Others can send messages to the same channel. This limit is solved with an extension called Private Channels.

Private channel – generate a key pair. Like a super secure email address. Message will be encrypted and sent to the owner of the address – only the owner can decrypt.

If they reply then I know it is the same user who is behind that address. This makes the system anonymous until you know the person behind the address.

Channel Address = the public Key. This Is LIKE a Public Key but it is actually an IOTA Address.

Channel key not transmitted over the internet or the channel; never sent over channel in plain form. You can contact anyone over a secure channel without having had to meet them yet.

In Healthcare you can hard code a private key address to an app. Person can send a handshake to the public key and start sending messages.

Handshake has 2 steps:

(Note that most comparable systems require >2 steps and participants must both be on line)

For this, make a private channel. You only need to be online once for it to work. Can then go offline. I share the address, someone sends me messages, I can go online later and can get the responses. This is with no intermediary, only IOT Nodes.

Another feature, for Healthcare, since everything is an extension, including encryption, you can share parts of your channel without disclosing the contents.

IOTA Mainnet disposes of transactions (messages) after 3 months. This is called the Snapshot. After that you lose messages.

Freighter – 2nd layer called Switchyard – people can offer their store for a low fee (overflow). You can get them to store your data in an IOTA Permanode but they can’t read the contents. Then you share the part of the Encryption Key that only tells where the messages will be but not what they contain. Can store or download these once they come in but cannot read. This is fully trustless – a ‘Need to Know’ basis. This uses different software. The User and admin don’t notice having those keys – the key relates to a key chain.

That’s the main parts.

#### History

Freighter was 2017 hackathon, before SKALY. It was optimized for use on IOTA Nodes.

#### Questions

Q: The Extensions – do these represent standard conformance point that anyone can write code to conform to?

A: Yes.

Only needed simplest form of channels. Accessible e.g. via a search function. The idea for an extension system arose out of that.

Also has a grant from the IF – parallel development w Streams. Learning.

#### Comments

NS: A lot of this work is already part of DDS.

We should talk to Erik Hendricks of Adlink on this. Suggest we talk to them on the LETS RFP Response. Also talk to Mat Grubis of GE Healthcare.

(addresses posted in chat log but not include in these formal minutes)

### DDS

MB: Can Nick (and others?) explain What DDS is?

**Answers:**

DDS is a real time publish subscribe system paradigm that allows messages to flow back and forth between publishers and subscribers with any number of publishers and subscribers, dynamically. You also create a data model that controls how it works (you publish the data model). Model can be simple e.g. txt message, or more detailed with levels of records. It is secure and defines levels of security - Including multi-level security, so every part of a message can be secured separately.

It is also governed by Quality of Service parameters, such as durability. So if message is stored, set the Durability flag. Can also define whether messages have to come in in the same order or not. A QoS parameter can be ‘show me latest’. Another can be ‘just show me updates’.

Other QoS – I can define frequency with which I publish updates, or that I publish updates when data changes by a given parameter.

So DDS defines both time-series based and event-driven.

DDS defines the notion of a ‘Topic’ (like our Channel here and in Streams).

Can also define something like geographical parameters (and / or something called Donuts) for e.g. when a message falls into or outside of a given ‘donut’. This allows for the creation of filters based on various parameters.

You can request what’s called a ‘Slug Trail’ e.g. ‘give me the last 10 messages’.

The things described here can all be built into a DDS Service.

### Introductions

Need to introduce DDS Foundation people to the SKALY folks

**Action:** MB forward SKALY emails to Nick.

#### Comparisons – Freighter and IOTA Streams / MAM

MAM v Freighter is like Ferrari v truck.

There are also something like Containers!

## LETS RFP potential changes.

Conformance points for the various extensions.

Long term persistency – relate to the DDS Durability QoS parameter.

QoS stuff for next layer (IOTA Layer1).

Should the LETS RFP Response describe the kinds of thing that need to be afforded to a LETS-conformant solution, by the next layer below it? This seems likely. Perhaps we should add something to the LETS RFP to ensure that the response includes some treatment for the layer that sits below it (Layer 1 DLT protocols, eg IOTA Tangle; Ethereum and others if they have the relevant affordances). We would then want to ensure that RFP responses reference DDS QOS parameters appropriately (material to be added in the Evaluation Criteria RFP section).

## Next week

SKALY can come again, assist with update on this LETS RFP

Char: How much to ask for in the RFP versus what detail is only in the response.

e.g. when we did DDS – had the idea of creating levels of compliance. E.g. basic pub / sub and various Conformance Profiles, e.g. kinds of QoS level.

Clarify if she means:

* Minimum that everyone has to do in the app?
* Or the minimum that an RFP Response (the spec) has to include?

Here CW is referring to the minimum the Specification (response) needs to say to conform with the RFP. This is the material in Clause 6.5 (mandatory features) that define what items must be included. There is also a Clause 6.6 (non mandatory features).

Separately there is the matter of what is the minimum that an application needs to do to conform with the Specification (that is, the RFP Response, which will become a standard). This talks to the matter of conformance points (Clause 2 of any OMG specification) and the use of what we call ‘Conformance Profiles’, which are coherent combinations of conformance points.

MB: On this, it should be clear that the RFP will include as mandatory items, the need to define both mandatory and optional conformance points in the RFP Response specification.

CW: See DDS 1.0 Spec, in response to a given RFP.

Can also talk with NS offline between now and then, on the DDS stuff.