# This is the cleaned list of the chats made at the RDA GEDE DO Web-Workshop at 22.5.2019

I took all non-content chat contributions.

Peter

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): Where/how are types defined?

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): this sounds a lot like <https://swagger.io/docs/specification/basic-structure/>

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): How do the default operations compare with: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): <https://www.w3.org/TR/ldp/> shows a structured way to use such collection operations using HTTP

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): how are the indirect references  to the larger audio recording bytes included? Are they also DOIs?

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): The DOBES approach to metadata for their "bundles" (collections) doesn't seem to be compliant with the recommendations of the RDA Research Data Colections WG. This is of course no problem in itself, but how important will it be for future DOIP-implementations that this type of (global) recommandations are followed?

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): <https://doi.org/10.15497/RDA00022>

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): During the work with the RDA RDC WG, ePIC people were looking at implementing the group's collection metadata schema and/or layers of operations on top to allow operations on the collection (search, access all members etc.). I'm not sure that this would have involved using CORDRA or not, but in any case the recommendations (linked above) are a good read for anyone wanting to implement machine-actionable DO collections...

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): DO for types is similar to URIs for XML name spaces, JSON Schema's $id HTTP URIs, and RDF/OWL use of ?HTTP URI namespaces. The important extra bit here is the "type registry" so that they are also Findable (and thus can reduce re-inventing types)

[Martin Schultz](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20141%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%209240588%3c/onRightClick%3e%22%3e%3c/A%3e): Regarding the discussion on Peter's slides, I am wondering if we really get an advantage with DOIP services compared to standard web processing services. Imagine a web processing service, which offers 1000 or more operations (for example all kinds of slicing and trimming of geodata plus operations for averaging and other aggregations across any dimension). If this services goes out of operation or changes its location, you need to update one URL in a service registry. With DOIP you would need to update 1000 service endpoints. The danger is thus that we get more garbage.

Peter: @Martin: this is exactly the point. DOBES would have to make an effort, what is the gain. We need to work this out in more detail.

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): +1 - but I hope the DOIP is not meant as a replacement for SOAP etc. for "doing anything", but mainly for data access in which cas ethe operations should mainly stay general or domain-oriented

Peter: In EUDAT we had enormous problems with different types of repositories. Where is the GOLDEN inerface standard?

Peter: NO - for sure DOIP is not a replacement for SOAP. It could be channeled via SOAP or REST whatever you like.

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): that is what SOAP said.. <https://en.wikipedia.org/wiki/SOAP#Transport_methods>

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): We will need a couple of very well-documented "demonstrators" that data managers of larger projects can investigate in detail - both to understand advantages & disadvantages compared with the current solutions, and to estimate costs & resources required to put the DOIP layer in place.

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): I would focus on \*mapping\* to the other protocols rather than wrapping - kind of like in the LDP spec which just says how you should use HTTP for that particular purpose. Then wrapping is only needed when something is unnatural in that protocol.

[Mark van de Sanden](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20131%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208585228%3c/onRightClick%3e%22%3e%3c/A%3e): I see advantages to have a standardised API for repository technologies, but this API should be supported by the Technologies and Technologie providers. It should not be that the repositories owners should develop these interfaces to the technologies. This will cost to must effort.

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): <https://swordapp.github.io/swordv3/swordv3.html> is more like that now compared to v2

[Martin Schultz](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20141%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%209240588%3c/onRightClick%3e%22%3e%3c/A%3e): It's just that at some point you are going to hit the anti-interoperability wall, no matter what you do. I like the way of thinking in DOIs and DOIPs, but I am sceptical that this concept can be implemented, or rather that any implementation provides real advantages over current ways fo doing things. Yet, we need to continue discussing and standardizing. At some point, critical masses will help.

[Mark van de Sanden](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20131%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208585228%3c/onRightClick%3e%22%3e%3c/A%3e): How does DOIP compare to SWORD, which has the same aim? SWORD is supported by some of the repository  technologies

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): I would say if you are just dealing with files and file collections then SWORD is the right abstraction level

[Mark van de Sanden](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20131%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208585228%3c/onRightClick%3e%22%3e%3c/A%3e): What are the added benefits of DOIP? within the metadata data collections and files is or can be referenced via PIDs.

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): If you use PIDs and manage your repos well over HTTP then you don't need DOIP. (or you could say, it is implementing DOIP already)

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): Yes, Mark's question is very pertinent. System architects and data managers will only "take the leap" to implement DOIP if they see an immediate benefit and/or it becomes what "everybody else" in the relevant scientific domain are doing. (For example because the use of DOIP becomes a strong FAIRness assessment metrics criterium, or is required by service catalogs such as the European Open Science Cloud marketplace.

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): that is the "stick" approach - there should also be a "carrot" motivation

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): @Stian: well, sort of - but the traditional REST API approach doesn't require you to register the operations and other details of e.g. your repository etc with a Type Registry (which gives them unique PIDs that can be used for unambiguous referencing.) That's one of the good things that DOIP brings to the table IMHO.

[Mark van de Sanden](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20131%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208585228%3c/onRightClick%3e%22%3e%3c/A%3e): I am also in favour of the carrot approach. There should be a buying from the technology providers and developers. They should develop the DOIP API on top of their technologies. This makes it easier for repository owners to enable the API.

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): To clarify: my post from 17:38 was in response to Stian's post from 17:33 ;-)

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): Exactly - so I think something similar to <https://www.re3data.org/> with data types  (e.g. JSON Schema) and OpenAPI protocols would do that well. Then a subset of those would be marked 'golden' by say EOSC for certain  repositories (e.g. how datacite schema is re-used by lots of repos)

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): Carrots are good, but I'm afraid that sticks will also be needed to get the ball rolling to quickly create a critical mass of adopters... (Sorry for mixing metaphors...)

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): say for a map type there would be a corresponding OpenAPI method set.  It is this glue ("what can I do?") that is not easy to find today.

[Aida Turrini](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20140%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%209175052%3c/onRightClick%3e%22%3e%3c/A%3e):I have a question as non-technique person. Is the blockchain structure compatible with the presented architectures?

[Christophe Blanchi (DONA Foundation)](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20115%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207536652%3c/onRightClick%3e%22%3e%3c/A%3e): Blockchain can be used to demonstrate that the state of a DO is still the same as that added to the chain.

[Maggie Hellstrom](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20116%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207602188%3c/onRightClick%3e%22%3e%3c/A%3e): Going back to my earlier comment about demonstrators: I think it would be great to have something like a Docker image with \_all\_ necessary "server side" components built in. That would make it really easy for data center people to instantiate it locally and play around with it - and then to build on the code to implement aspects of their own specific (data) services. Does this exist for e.g. the SEADTrain case?

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): <http://swordapp.org/>

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): v2 is commonly implemented, v3 is work in progress

[Martin Schultz](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20141%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%209240588%3c/onRightClick%3e%22%3e%3c/A%3e): Compare to very basic CRUD: these are abstract operations with defined behaviour, which have been implemented in different APIs/protocols. With DOIP we seem to be mixing the abstract definition and the API, which might lead to trouble as Mark is arguing. Perhaps one can add one more layer of abstract operations to CRUD (things that everyone does, similar to operators in programmign languages, e.g. add, subtract, ...) - but I doubt very much that "yet another standard" (thanks!) helps.

[Rob](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20118%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%207733260%3c/onRightClick%3e%22%3e%3c/A%3e): <https://github.com/rpidproject/puppet>

[Stian Soiland-Reyes](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20133%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208716300%3c/onRightClick%3e%22%3e%3c/A%3e): focus on best practice using existing technology, and the added value of a registry for types/protocols. Don't re-invent another Semantic Web without Web or without Semantic.

[Mark van de Sanden](res://\\G2MResource_en.dll/%3cA%20HREF=%22%3conLeftClick%3eeCMD_SetChatTo%20131%3c/onLeftClick%3e%3conRightClick%3eeCMD_DoAttendeeContextMenu%208585228%3c/onRightClick%3e%22%3e%3c/A%3e): Peter, and the presenters, thank you for organising this workshop. It was very interesting. I have to leave.