OPEN GEOSPATIAL CONSORTIUM
FAA SWIM Discovery Service Workshop
Summary Meeting Minutes
September 9, 2020

The Open Geospatial Consortium (OGC) and the Federal Aviation Administration (FAA) hosted the SWIM Discovery Service (SDS) Technical Review Workshop virtually via GoToMeeting on September 9, 2020. This 3-hour community workshop solicited feedback from the OGC technical community on the FAA's SDS Implementation Specification. Potential participants were asked to provide an initial review of the SDS specification and provide comments via the Comments and Resolutions form (attached). Here are the video recording and GoToMeeting Transcript (https://transcripts.gotomeeting.com/#/s/dd4ddd290c3683744a34c8903278543c5e9ab3dcb09942e297ed2c52bcf86e50).

Presenters
Mark Kaplun, FAA
Nadine Alameh, OGC
Charles Chen, Skymantics

Attendees

Peter Vretanos – Cubewerx
Pedro Fernandez – Eurocontrol
Stephane Fellah – Image Matters
Josef Jahn - Frequentis
Ricardo Silva – DECEA
Alex Nguyen – Concepts Beyond
Wen Zhu – NIRA
Oliver Krueger – DFS
Carol Uri (FAA contractor)
Harry Newett – Noblis
Henry Chan – Hong Kong CAD
Caroline Inglefield – Noblis
Jiseok Kang – KAC
John Fort – Frequentis

Peter Vretanos – Cubewerx
Pedro Fernandez – Eurocontrol
Stephane Fellah – Image Matters
Josef Jahn - Frequentis
Ricardo Silva – DECEA
Alex Nguyen – Concepts Beyond
Wen Zhu – NIRA
Oliver Krueger – DFS
Carol Uri (FAA contractor)
Harry Newett – Noblis
Henry Chan – Hong Kong CAD
Caroline Inglefield – Noblis
Jonathan Fath – OGC
(Offline) Clemens Portele – interactive instruments

Introduction
Workshop coordinator, Charles Chen, and OGC CEO, Nadine Alameh, jointly kicked off the meeting with brief opening remarks, which were followed by introduction of the OGC members in attendance and guest organizations. Nadine described the OGC organization and the purpose of the workshop. Charles introduced the FAA governance lead, Mark Kaplun, who kicked off the technical presentation.

Background
Mark Kaplun provided an overview and background of the FAA's SDS Implementation Specification. He gave a historical background on SWIM and a comparison of Discovery Service vs. Registry. He presented the Objectives, Design Principles, Architectural Vision, and the Specification content headings.

First Questions Pause
Because the SDS specification has a dual purpose (implementing a single instance of an SDS, and interaction among SDS instances), what would be your preferred documentation approach? Two documents instead of one? What kind of documents?

Josef Jahn from Frequentis commented (1) that from his perspective, the SDS is addressing a principal issue when dealing with service registries, which is either synchronization or the discoverability of services across different service registry. As a user of a service registry, he would prefer to have as few different specifications as possible and expects the discoverability to become part of the API at the interface of the service registry itself.

Mark responded that the FAA currently has this capability now. The application is designed for a user and follows organization specifics, approach, innovation specific engineering, and business practices. A user can create an account in the registry and search for some specific service. If you want to compare this information from another registry, you have to create another account, login into another registry, and collect information from there. This information is semantically and structurally not consistent. Which means then as a user, you will have to take whatever you can scrape from both systems. What we are trying to achieve is a single consolidated response. Also, when you say API for registry, to me it means building some kind of service which is used as a registry back-end or data source. This is one solution. Based on our trials, we now have an operational registry interchange model with the FAA registry. We realized that the cleaner solution is to compose services using a software agent to exchange information between services.

Peter Vretanos from CubeWerx says that from implemented implementation point of view, two documents may be preferable. Going down the path of writing specifications that encapsulate a lot of information as the old OGC web service specifications (e.g. WFS, OGC Catalog, etc.) and the success was questionable because the documents ended up being too big and implementers didn't want to have to slog through all of that information. So now we're adopting the more modular approach, which tends to indicate that no, well defined functional units or implementation units are defined in individual specifications, and then you know referenced where necessary. From an implementer point of view, we found that having smaller more focused documents is preferable.

Mark Kaplun: Do you have any idea of the point, or it's still too early to say what kind of documents you would suggest for this multiple document?

Peter Vretanos: If you're defining a service API, then logically related resources and behaviors should be described in in one document. If we take a look at the current sort of trend in the OGC, we tend to define a core specification like we did for OGC API features and processes, which define the basic, minimal functionality that's required to satisfy the requirements of the surface. And then, there are additional modules are parts that define more extended or related capabilities, but that aren't absolutely necessary for implementing the core functionality. So, that requires that you do a really detailed analysis of what is absolutely core and discard everything else. We went through that exercise with the WFS, where it too had a lot of functionality. We actually went through the process of completing WFS 3.0 from WFS 2.0, and it had even more functionality and got even bigger. So that's when we kind of put on the brakes, we threw away what we did for WFS 3.0 and started again. We started from the point of view: If I was implementing this feature service, what the absolute core functionality that satisfies a certain set of requirements? Then after that, whatever other related capabilities or extensions you need to put into other sections, that’s sort of straightforward. That’s the approach that we took, and so far, it seems to be working well. All of the other OGC resource specifications are following the same model.

Ricardo Silva from the Institute of Airspace Control – Brazil: On the topic of web services, there appear to be many different ways to do the same thing. We are looking for interoperability. OGC could lead us, and we could follow your ideas to work together. The SDS specification was really useful and helpful for me because I had a lot of questions. For example, we could use REST or other technology and XML for the logical exchange. You showed us a way to doing things, and we see a benefit in doing things the same way to standardize the development of the global exchange environment.

Mark Kaplun: Based on Ricardo’s comment, after the SDS is implemented it should be some specific specification or documentation, for instance, of the service. This is correct, it exists and it’s something that we have in FAA and in the Korean Airport Corporation, building services to this specification, and, instances of that service and its own documentation of its of service description, set of requirements, etc. We'll talk about this at the very end of this workshop when we talk about current implementations.

Alex Nguyen from Concepts Beyond: There are lots of dependencies among those two things. While having separate docs would help developers, having a single doc would ensure consistency and reduce coordination in ensuring consistency.

Behavior Model
Mark provided an overview of the Behavior Model, Actors, Assumptions, and Use Cases 1, 2, & 3.

Second Questions Pause
Do these use cases sufficiently describe the behavior consistently with identified objectives? Will you suggest more use cases? Less? Others?

Stephane Fellah from Image Matters asks, how do you handle service failures?

Mark Kaplun: The methodologies that we follow for the use case scenarios, it's always a perfect day scenario. We didn't include in the scope of this model, possible failures. One example might be: I expect the Discovery Service to provide me with list of preferences, but then Discovery Services, has not implemented a list of preferences. So, instead of my request fails, and I never receive a response. This is an opportunity to find out what this service can do. What is required for authentication?

Ricardo Silva: Where can I find more specifications and instances about SWIM?

Mark Kaplun: Some information is available via swim.aero. Please send an email and we will try to send you information. FAA has a page for standards and specification such as versioning and for machine processable documentation. There are also several SWIM specifications that Eurocontrol has released. There are several working papers from APAC.

Ricardo Silva: It would be very useful to have all these specifications available about how to develop and implement all the registry, the services, the WSDLs, UDDI, and the software necessary. I have my own ideas about how to do so, and so does my team. This kind of workshop is very useful to work together and get all the information together.

Alex Nguyen: What about automatically publishing new changes/discoveries? Other than a request/reply model (or in addition to), Can new changes be published to others? In a request/reply model or other services, how would a registry publish any new information it gets to others?

Mark Kaplun: SDS is a service. Like any service, if there are changes to any data, it's not affecting in any way the service interface or service behavior, then there is no publication at this point. It's not publish-subscribe model, and not event driven. It's a request/response model which is stateless, and every time you ask about the specific state of a specific service, that's what you will receive.

Peter Vretanos: I think Alex was indicating that you, you have a pub slash sub model as a method for disseminating changes, so when a new surface gets added, it gets published and the services that have subscribed will get a notification.

Alex Nguyen: Right, so otherwise, the person inquiring would have to constantly poll the service to find out when changes are made. So you have to continue to ask what changes have happened versus the other model which will inform about changes.

Mark Kaplun: Yes, that's correct. It's up to the user to determine how often to request this information to refresh. It's not like a weather service that changes every hour. It's pretty static information that changes very infrequently.

Information Model
Mark Kaplun presents details on the information model, structure, and UML diagrams of the Discovery Service Information, Indexed list of services, and Service Description Conceptual Model (SDCM).

Third Questions Pause
Is there any other information that the Discovery Service should provide? Could you suggest any other categories that should be used for "filtering"?

Peter Vretanos: Do you anticipate that filtering would be required on every field in the information model? Would the concept of "queryables" be useful?

Mark Kaplun: No, we have about 40 or 50 fields, not more. We hope that this is something that can be addressed through semantics. We have through SDCM an ontology equivalent. For now, we select three criteria which are most commons, but this is relative. Hopefully, these are common to every implementer of SWIM.

Peter Vretanos: I don't know if, if this idea is useful to you, but one kind of one emerging idea is this idea of queryables where there's a resource or an endpoint that you can query that will tell you here are the properties that you can build filters on.

Mark Kaplun: Yes, that sounds interesting, similar to WFS where we can query anything. Maybe for the future.

Peter Vretanos: If the information is static, as you said previously, and there's not that much of it, in terms of the number of services, what kind of filtering are you anticipating by geographic location?

Mark Kaplun: That's a good question (by geographic location). We filter information requests such as filter requests to go to a certain discovery service. So, if you are to say, request form discovery services by city (for references), all services are probably are going to be USA geographically. The same is probably true for a Europe, Asia, etc. Maybe for this reason, we haven't gone this direction. It's a useful attribute, but I'm not sure about filtering. We have to think about this some more.

Josef Jahn: Geographic filtering is quite important, especially for services that might not be tied to ATM references, like sensor coverage restricted services, UTM, etc.

Mark Kaplun: I agree, and we've been looking to, as the same model, we've been thinking about updating this model in one item as a geographical extension. We should have some kind of taxonomy of filter and a predefined set of values which we don't have now. We would like to add this to our To-Do's. Perhaps OGC has something here?

Peter Vretanos: I was not specifically thinking about Geographic filtering (although I can see it being of prime importance). I was thinking more generally about the fact that when you have a large pool of properties describing a service it is likely that only a smaller subset of properties would be used to actually query or filter the data. The "querables" idea is simply a means to allow a service to advertise which properties can be used to query the information that the service offer. We in OGC use this is features and records (i.e. catalogue). Here is an example ... https://dev.api.weather.gc.ca/msc-wis-dcpc/collections/discovery-metadata/queryables?f=json

Resource Model
Mark Kaplun presented the Resource Model.

Fourth Questions Pause
Does the SDS's resource model accurately represent the OpenAPI structure? Any examples of verbalizing a resource model that you can share?

Stephane Fellah: If the services are supposed to be described semantically, we should have a least one machine understandable format such as JSON-LD, TTL, RDF, etc. https://www.w3.org/TR/vocab-dcat-2/

Mark Kaplun: All services for all information unions that I showed you before have JSON schemas associated for every information element. Is that sufficient? Do you suggest more?

Stephane Fellah: The schema: it's not really a semantic format, because it's not an ontology which defines the core constraint on the model. There is a way to reach JSON output to linked data by using JSON LD that implied that you define a JSON LD context that's grounded to an ontology. It's very important that it's done semantically or the machine will not be able to understand the model and this will have to be hardcode the semantic into the code based on the schema. To be able to link to other resource, like datasets, maps and layers, you want to use a common framework, and use a data centric approach or you will not be able to enable integration.

Mark Kaplun: Common Semantic, unambiguous, semantic machine processable will be critical for success for an effort like this.

Stephane Fellah: Especially, you have defined already, some taxonomies for the service interface, shows categories defined in SKOS, and you ought to be able to link to his vocabulary using a data format. You have an XML schema referring to a URI in linked data, but it's difficult to do reasoning, because you're using a different model for presenting information. So, having a unified way to do that using a linked data framework to facilitate the integration. I will advise you look at DCAT for example, to introduce data services linking to data sets, and you may want to extend the data service with your model that you have defined in the spec, by expanding it. I'm referring to the DCAT 2 model published by the W3C. The OGC Testbed 16 spec is aligned with this.

Peter Vretanos: Wouldn't pretty much any OGC API specification be considered a verbalization of a resource model? All the OGC API specifications normatively include an open API description of the resources, but the specification itself also has narrative text, going into more detail about each resource and the requirements of behavior on each resource. The two have to go parallel. We don't make Open API mandatory. It's a conformance class, but we use it to describe the resource models for features in OGC API records and coverages, and all the other specifications.

Mark Kaplun: Probably. There is a lot of OGC documentation. We will take a look.

Interface Requirements
Mark Kaplun presented the Interface requirements, Operations, and Security Context. Wen Zhu (NIRA) presented the approach to Security Requirements for the SDS.

Fifth Questions Pause
How are Security and Trust requirements specified in your domain? What security protocols and standards are you using?

No Questions or Comments

Presentation on Current Implementation of FAA & Korean Airport Corporation SDS instance

Charles Chen provided some closing statements before Parking Lot questions review.

Parking Lot questions review

Regarding Questions Pause 1

Stephane Fellah: End users are mostly focused on discovering and consuming data, which are provided by services. Services and datasets are the Ying and Yang of information systems. I would not separate the services from datasets in different registries because they are interlinked. What is needed is a semantic registry that links services, datasets , organizations, maps, layers, etc. using a unified framework based on open linked data standards. Central focus is the dataset. What's important is getting the right metadata. Concern is the model is getting into too much detail such as how to describe the message and operations. What should be focused is what standard (e.g. Swagger OpenAPI) for the service type, function of services, etc.) Advise to push on more semantics and linked data representation.

I think the end user is really looking for information, not as a service, but as the first thing is the information you're looking for. And of course, you get these data from services with a rest API.
So, I think the central focus is really the data set, and then the service is kind of an ulterior thing. Of course, both are important. What's important is getting the right metadata for the search and discovery for both of them. I'm a little concerned that the model for the service scanning is getting too much into detail. Really, what we're looking for is: I have a service following these standards. Do we have to describe the detail of operation and the messages, you know, in all the gory detail? That's not very useful. I think what's important is what type of service you are looking at and what standard API description standard that we're using, like, Swagger or Open API, that, just enough. I don't think we need to describe in detail in the registry, but just having a reference to a standard URI or an Open API document, will be enough. What's important is really the service type, the function of the services, and that's about it when you search and discover.

My other comment is about the fact that we described all our data centric using XML schema or JSON schema. My experience is if you want to integrate everything semantically, you have to define ontology. It would be much easier

if the data were directly you encoded in the linked data using data centric like JSON or XML. Because that will require us to we write code to convert the data schema to ontology. It would be much easier to skip this step by providing directly linked data representation of the service. It would be easier to extend the API with a machine understandable format with JSON-LD or RDF. You will avoid to do this step of manual conversion from schema to ontology. I invite you to consider pushing more of a semantic on the semantic front, and DCAT is a kind of a central role in the community to enable the integration. You just make the integration much easier when you have linked data representation.

Mark Kaplun: For us it's important that we not only strive for what we like or what's the best possible solution, but also consider the user and what the user is ready for. As good as this ontology and semantic approach is for the client for this kind of information, the implementations are rather limited in number, and there is even less from the consumer that we feel is ready for this. It will take some time to educate our potential consumers to come up with manageable solutions in our immediate business plan. If we use something like REST HTTP services, there is something we can provide now. This is really important for us to make this a useful services and address existing demand. However, at the same time it's important to continue to work on semantically enabled registry, etc. I think we're doing this. We presented some of this today. We are sponsoring OGC to make an environment to continue this work in OGC Testbed. This topic can be addressed perhaps in a future workshop.

Stephane Fellah: Are we solving the right problem? You want to solve a problem for specific users. It's important you define the problem well. Does this solve a question for a specific user? Is the system useful today? Does it do what they need it to do? That's the critical question. I'm not sure if this is going to solve the problem. It's just another schema out there and another silo that's pushing the system.

Clemens Portele (interactive instruments) and Oliver Krueger provided comments offline via the Comments and Resolutions Matrix (attached).

A discussion about Open API and its applicability to the SDS architecture. Peter Vretanos describes that Open API is one solution which can provide a way to describe the SDS service interface. Mark Kaplun explains that WSDL and WADL are other options, but OpenAPI seems to check all the boxes. An assessment for usefulness of OpenAPI needs to be conducted.

Final comments regard FAA and KAC integrations to do testing of these interfaces and future use cases for bi-directional testing. Mark Kaplun openly invites more participants to join in the testing to add more peers to validate the interfaces. He says that they are looking for a permanent home for this information to be hosted such as OGC. Once a location has been identified, all participants will be notified. This specification is version 1.0.0, so this is brand new, and ideas, comments, questions, critiques are welcome. Please send comments and resolutions matrix to us.

No.	Reviewer/ Organization	Section/ Paragraph	Comment	Resolution (leave blank)
1	Josef Jahn/ Frequentis	Questions Pause 1 – Slide 13	Prefer to be part of the API of the Service Registry. Prefer to have as few specifications as possible to make it part of the service registry itself.	FAA has trialed a service and there is a disconnect between the registry and the data source documents. API for registry – some kind of service registry with data source.
2	Peter Vretanos/ CubeWerx	Questions Pause 1 – Slide 13	Prefer SDS to be two documents instead of one. We are not adopting a more modular approach. Well-defined implementation units are defined in individual specifications. Implementations are using reference specifications. Smaller more focused documents. Logical and resource behaviors should be specified. OGC API Features and OGC API Processes. Additional modules or parts for extended and related capabilities. Analyze what is core and discard everything else. WFS2 had more functionality. WFS3 has even more. Threw that out and went back to core and defined that in OGC API Features, and everything else are other functionalities.	Agree. We recognize it's a problem for OGC standard to try to cover everything.
3	Ricardo Silva/ DECEA	Questions Pause 1 – Slide 13	Once the SDS is a service, I understand that a unique document is suitable.	
4	Stephane Fellah/ Image Matters	Questions Pause 1 – Slide 13	End users are mostly focused on discovering and consuming data, which are provided by services. Services and datasets are the Ying and Yang of information systems. I would not separate the services from datasets in different registries because they are interlinked. What is needed is a semantic registry that links services, datasets , organizations, maps, layers, etc. using a unified framework based on open linked data standards. Central focus is the dataset. What's important is getting the right metadata. Concern is the model is getting into too much detail such as how to describe the message and operations. What should be focused is what standard (e.g. Swagger OpenAPI) for the service type, function of services, etc.) Advise to push on more semantics and linked data representation.	Another thing to think about is what is the end user ready for? FAA SWIM Consumer readiness means educating the consumer to come up with a manageable solution that fits into the current business plan. REST and HTTP services are useable now. We need to address existing demand, but also continue to work on semantic enablement. We're doing that today with OGC Testbed 16. Perhaps a separate workshop specific for semantic enablement.

No.	Reviewer/ Organization	Section/ Paragraph	Comment	Resolution (leave blank)
5	Pedro Fernandez/ Eurocontrol	Questions Pause 1 – Slide 13	Agree for a split of 2 documents	
6	Alex Nguyen/ Concepts Beyond	Questions Pause 1 – Slide 13	There are lots of dependencies among those two things. While having separate docs would help developers, having a single doc would ensure consistency and reduce coordination in ensuring consistency.	
7	Stephane Fellah/ Image Matters	Questions Pause 2 – Slide 21	How do you handle service failures?	Current use cases are positive cases and do not address failures. Instead of a failure,
8	Oliver Krüger, DFS	Questions Pause 2 – Slide 21	I have provided some findings in the comment sheet concerning potential contradictions between the text and the figures. Will those be addressed today or is that for upcoming meetings?	We will address these offline.
9	Ricardo Silva/ DECEA	Questions Pause 2 – Slide 21	Where can I find more specifications and instances about SWIM?	Some information is available via swim.aero. Please send an email and we will try to send you information.
10	Alex Nguyen/ Concepts Beyond	Questions Pause 2 – Slide 21	What about automatically publishing new changes/discoveries? Other than a request/reply model (or in addition to), Can new changes be published to others?	At this moment, there is no event-driven pub/sub model. It is purely request/response based on stateless model.
11	Ricardo Silva/ DECEA		Are anyone at the audience researching the use of web semantic or any reference ontology for SWIM	Charles: Yes, there is currently some work being conducted in OGC Testbed 16 - Aviation thread for the use of semantic ontology. The results will be published sometime at the end of this year after the testbed is concluded. If you would like sooner information, consider joining the OGC membership and becoming an "Observer" of the Testbed and participating in the discussions actively. Please email Scott Serich for more information regarding joining the OGC Testbed.
12	Peter Vretanos/ CubeWerx	Questions Pause 3 – Slide 29	Do you anticipate that filtering would be required on every field in the information model? Would the concept of "queryables" be useful?	No, we have a semantic ontology that we plan to use. We select 3 most common criteria. Yes, queryables could be useful, but we're not ready for that yet. Perhaps in the future.
13	Josef Jahn/ Frequentis	Questions Pause 3 – Slide 29	Geographic filtering is quite important, especially for services that might not be tied to ATM references, like sensor coverage restricted services, UTM, etc.	Agreed. FAA is looking at updating the Service Description Conceptual Model (SDCM) to accommodate. Service taxonomy development is also on our to-do list. Please share any references for this kind of information.

No.	Reviewer/ Organization	Section/ Paragraph	Comment	Resolution (leave blank)
14	Peter Vretanos/ CubeWerx	Break	Just so that it is captured in the video ... I was not specifically thinking about Geographic filtering (although I can see it being of prime importance). I was thinking more generally about the fact that when you have a large pool of properties describing a service it is likely that only a smaller subset of properties would be used to actually query or filter the data. The "querables" idea is simply a means to allow a service to advertise which properties can be used to query the information that the service offer. We in OGC use this is features and records (i.e. catalogue). Here is an example ... https://dev.api.weather.gc.ca/msc-wis-dcpc/collections/discovery-metadata/queryables?f=json	
15	Stephane Fellah/ Image Matters	Questions Pause 4 – Slide 34	If the services are supposed to be described semantically, we should have a least one machine understandable format such as JSON-LD, TTL, RDF. https://www.w3.org/TR/vocab-dcat-2/	
16	Peter Vretanos/ CubeWerx	Questions Pause 4 – Slide 34	Wouldn't pretty much any OGC API specification be considered a verbalization of a resource model?	Yes, probably.
17	Clemens Portele/ interactive instruments	Email comments	The use of OpenAPI does seem to add little to the architecture since the spec is mainly about describing services/APIs with their own SWIM-specific metamodel for services/APIs. Since you mentioned OGC API in the original email: As far as I understand it, I do not see a real relationship with the OGC API developments. Different motivations and design decisions.	There was a discussion about Open API and its applicability to the SDS architecture. Peter Vretanos describes that Open API is one solution which can provide a way to describe the SDS service interface. Mark Kaplun explains that WSDL and WADL are other options, but OpenAPI seems to check all the boxes. An assessment for usefulness of OpenAPI needs to be conducted.
18	Clemens Portele/ interactive instruments	Email comments	Some parts look inconsistent (e.g., an XML response is specified by a JSON schema?), so I assume this has not been implemented and tested with multiple implementations. If that is the case, I would recommend to do that before moving forward.	Yes, we have XML and JSON schemas. We've tested with Korea Airport Corporation who has 3 iterations. Also, there is a plan for bi-directional testing. Also expand from two-peers to multiple peers.
19	O. Krueger, DFS	2.1	Contradiction between Assumptions C) and 2.1.1 Use Case 1 "the user sends a request to Y. Y responds with…", the text also contradicts the figure, where all communication goes via X.	*Change text in the scenario according to the assumptions and the figure no 2*

No.	Reviewer/ Organization	Section/ Paragraph	Comment	Resolution (leave blank)
20	O. Krueger, DFS	2.1.2	Why does the user need to know the network address of Y as a precondition for Use case 02? All communication should be done via X.	Consider deleting this part of the precondition.
21	O. Krueger, DFS	General	The “hierarchy” of peers is limited to two from the user’s point of view in this concept. I guess that is intentionally, because in principle all connected peers could provide all information via X to the user. If there are reasons not to go that way, those should be stated as constraint for the implementation of the distribution service.	
22	Oliver Krueger, DFS	General	Maybe a chapter “Architectural Constraints” should be considered.	
23	Oliver Krueger, DFS	2.1.4	Figure 5 is not in line with the text in the scenario, the behavioral diagram gives the impression that everything goes via X.	Align text and figure
24	Oliver Krueger, DFS	2.2.1.1.	Figure 6 : the UML model should include all attributes mentioned in the text underneath, otherwise it is confusing. When implementing such things, for interoperability reasons one should be as explicit as possible.	Add missing attributes to the classes.
25	Oliver Krueger, DFS	2.2.1.1.	Remark: if security mechanisms like authentication are in place, it can be very cumbersome for the single user to communicate with several peers. That can be a reason why it should at least be possible to ask one peer to gather information about all others. A server to server authentication is anyway to be established. This presumes that the servers trust each other. This leads to some questions: - Do we need a transversal user access policy to be applied to a registry which wants to participate as a peer? - This is all about “read only”, so maybe not a big problem, but I know from the European registry, that for un-authorized users, not all service information is visible.	
26	Oliver Krueger, DFS	2.2.1.2.	Figure 9: the peer.json should not reside at FAA but maybe at OGC.	
27	Oliver Krueger, DFS	2.2.2.1	Service information should be aligned with the ICAO Service Overview.	Align with the ICAO Service Overview and also mention somewhere this as a reference